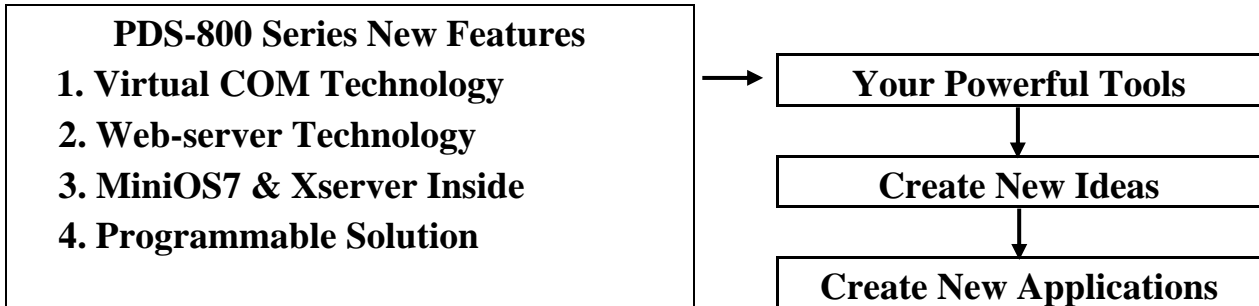

PDS-811/PDS-821 Series

User Manual



Warranty

All products manufactured by ICP DAS are under warranty regarding defective materials for a period of one year, starting from the date of delivery to the original purchaser.

Warning

ICP DAS assumes no liability for damages resulting from the use of this product. ICP DAS reserves the right to change this manual at any time without notice. The information published 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.

Copyright

Copyright 2009 by ICP DAS. All rights are reserved.

Trademark

The names used for identification only may be registered trademarks of their respective companies.

Table of Contents

PACKING LIST	5
MORE INFORMATION:	5
1. INTRODUCTION	6
1.1 WHY ETHERNET SOLUTIONS?	6
1.2 WHY VxCOMM TECHNOLOGY?	7
1.3 WHY WEB SERVER TECHNOLOGY?.....	9
2. TYPICAL APPLICATIONS FOR THE PDS-800.....	10
2.1 RS-232/485/422 DEVICE NETWORKING	10
2.2 ETHERNET I/O APPLICATIONS.....	11
2.3 LINKING I-7000 SERIES MODULES TO AN ETHERNET NETWORK	12
2.4 CONFIGURABLE ETHERNET DATA LOGGER.....	12
3. HARDWARE INFORMATION.....	15
3.1 FEATURES	15
3.2 APPLICATIONS	15
3.3 SPECIFICATIONS	16
3.4 FRONT VIEW	17
3.5 PIN ASSIGNMENT	17
3.6 DIMENSIONS	18
3.7 PDS-800 SELECTION GUIDE	19
3.8 OPTIONAL SERIAL MODULES PIN ASSIGNMENTS	20
3.8.1 I-8112iW Pin Assignments.....	20
3.8.2 I-8114W Pin Assignments.....	22
3.8.3 I-8114iW Pin Assignments.....	24
3.8.4 I-8142iW Pin Assignments.....	26
3.8.5 I-8144W Pin Assignments.....	28
4. WIRING NOTE FOR RS-232 AND RS-422/485 DEVICES.....	30
4.1 RS-232 WIRE CONNECTIONS	30
4.2 RS-485 WIRE CONNECTIONS	31
4.3 RS-422 WIRE CONNECTIONS	32
5. SETTING UP THE PDS-800 CONTROLLER.....	33
6. CONFIGURATION WITH WEB BROWSER.....	42

6.1	CONNECTING TO THE PDS-800 CONTROLLER.....	42
6.2	NETWORK SETTINGS	43
6.3	IP FILTER SETTING.....	46
6.4	COM PORT SETTINGS	47
6.5	MISCELLANEOUS SETTINGS.....	53
7	CONSOLE / TELNET COMMANDS LIST	55
7.1	OPERATION FLOWCHART	55
7.2	INIT/RUN MODE SETTING.....	56
7.3	COMPARISON TABLE (INIT/RUN/CONSOLE MODES).....	56
7.4	COMMAND LIST.....	57
7.4.1	<i>IPFILTER</i>	58
7.4.2	<i>IPCONF</i>	60
7.4.3	<i>SOCKET</i>	61
7.4.4	<i>COM</i>	62
7.4.5	<i>Broadcast</i>	63
7.4.6	<i>SystemTimeout (ms)</i>	64
7.4.7	<i>SocketTimeout (ms)</i>	65
7.4.8	<i>M</i>	66
7.4.9	<i>EchoCmdNo</i>	67
7.4.10	<i>EndChar</i>	68
7.4.11	<i>IP</i>	69
7.4.12	<i>MASK</i>	69
7.4.13	<i>GATEWAY</i>	70
7.4.14	<i>MAC</i>	70
7.4.15	<i>NAME</i>	71
7.4.16	<i>ALIAS</i>	71
7.4.17	<i>DHCP</i>	72
7.4.18	<i>UDP</i>	73
7.4.19	<i>VER</i>	74
7.4.20	<i>SAVE</i>	74
7.4.21	<i>LOAD</i>	75
7.4.22	<i>CONFIG</i>	77
7.4.23	<i>RESET</i>	78
7.4.24	<i>QUIT</i>	78
APPENDIX A:	LINKING TO A DEVELOPMENT PC.....	79
APPENDIX B:	GLOSSARY.....	83

1.	ETHERNET	83
2.	INTERNET	83
3.	TCP/IP	83
4.	TCP (TRANSMISSION CONTROL PROTOCOL)	83
5.	UDP (USER DATAGRAM PROTOCOL)	83
6.	GATEWAY	83
7.	IP (INTERNET PROTOCOL) ADDRESS	84
8.	MAC (MEDIA ACCESS CONTROL) ADDRESS	84
9.	SUBNET MASK	84
10.	ARP (ADDRESS RESOLUTION PROTOCOL)	84
11.	RARP (REVERSE ADDRESS RESOLUTION PROTOCOL)	85
12.	ICMP (INTERNET CONTROL MESSAGES PROTOCOL)	85
13.	PING	85
14.	PACKET	85
15.	SOCKET	85
16.	CLIENTS AND SERVERS	86
17.	FIRMWARE	86
	FAQ	87
1.	HOW TO ACCESS THE REMOTE PDS-800 THAT PLACED BEHIND AN NAT OR FIREWALL?	87
2.	HOW TO OPEN A VIRTUAL COM PORT THAT LARGER THAN "COM 10" BY CALLING CREATEFILE() WIN32 API? ...	89
3.	DOES VxCOMM DRIVER (PC) SUPPORT AUTO-RECONNECTION AFTER FIXING A NETWORK BREAK?	89
4.	WHY DOESN'T THE VxCOMM DRIVER (PC) RECEIVE DATA FROM THE 7188E/8000E/PDS-700/DS-700/PDS-800 MODULE?	90
5.	DOES THE TRANSMISSION SPEED BECOME FASTER WHEN THE SERIAL DEVICE WORKING WITH SERIAL TO ETHERNET DEVICE SERVERS?	91
6.	WHY DOES THE 7188E/8000E/PDS-700/DS-700/PDS-800 MODULE FAIL ON A (PUBLIC) INTERNET CONNECTION? ...	93
7.	CAN I USE THE SETCOMMSTATE () API TO CHANGES THE BAUD RATE/DATA FORMAT SETTINGS OF A VIRTUAL COM PORT?	95
8.	HOW MANY PCs CAN BE CONNECTED TO A SINGLE PDS-800 DEVICE?	95
9.	CAN I SEARCH OR CONNECT TO PDS-800 WHEN MY PC'S IP ADDRESS IS NOT IN THE IP FILTER LIST OF PDS-800? HOW CAN I SOLVE IT?	97

Packing List

The package includes the following items:

- One PDS-800 series main control unit
- One printed Quick Start Guide
- One software utility CD
- One download cable, CA-0910 (only for PDS-800 controller)

Note:

If any of these items are missed or damaged, contact the local distributors for more information. Save the shipping materials and cartons in case you want to ship in the future.

More Information:

Documentations

CD:\NAPDOS\PDS\PDS-800\Readme.htm

CD:\NAPDOS\PDS\PDS-800\Document\Readme.htm

VxComm Driver (Virtual COM)

CD: \NAPDOS\Driver\VxComm_Driver

Firmware

CD:\NAPDOS\PDS\PDS-800\VxComm\Server(PDS)

MiniOS7

CD: \NAPDOS\PDS\PDS-800\OS_image

1. Introduction

PDS-800 is a series of **Programmable Serial-to-Ethernet Device Servers**. They designed to meet the most common requirements of Internet/Ethernet applications, and enables users to remotely control your serial devices through an Ethernet network. PDS-800 comes with a powerful and reliable **Xserver** programming structure that allows you to quickly design robust Ethernet applications.

1.1 Why Ethernet Solutions?

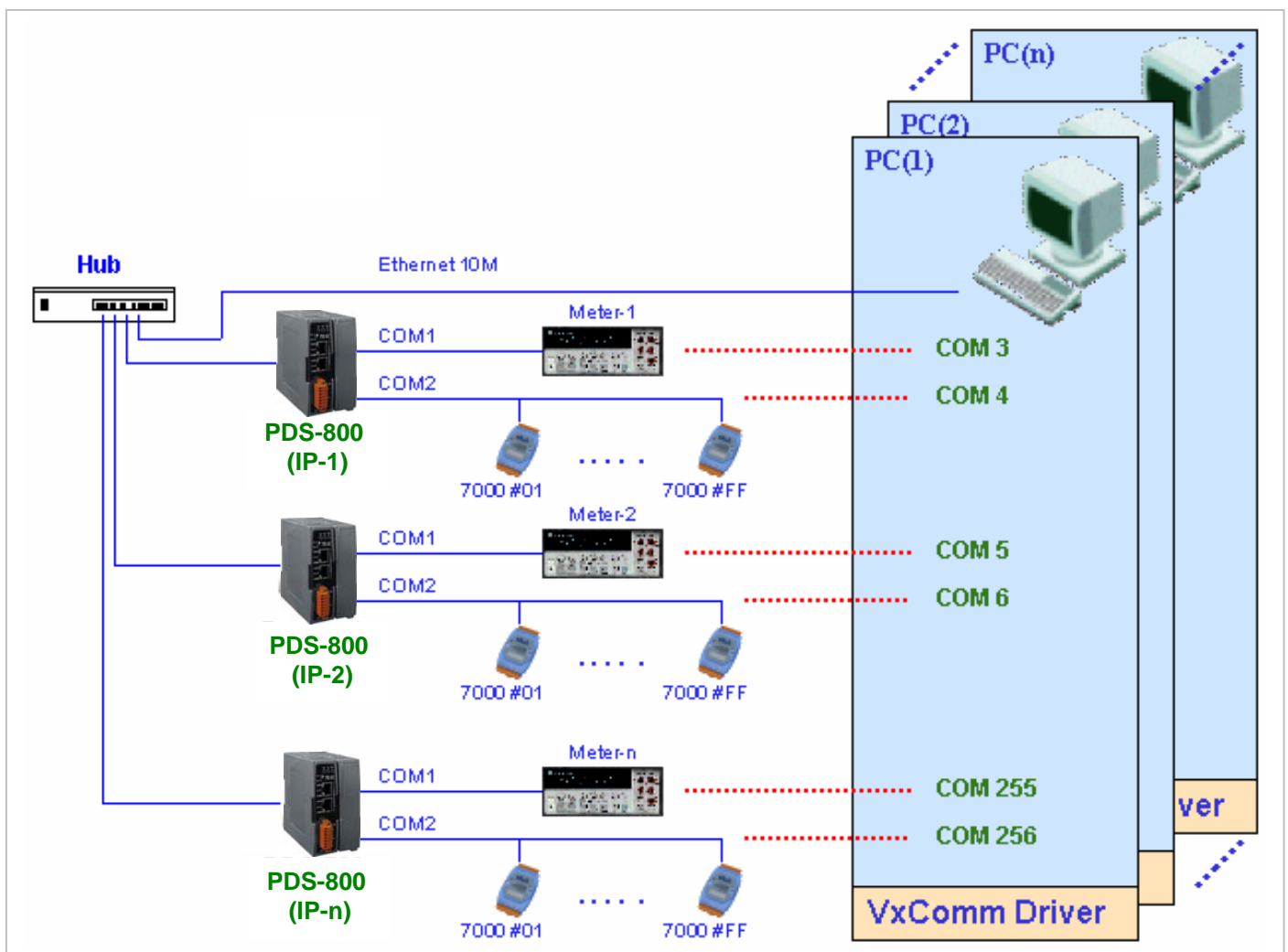
Nowadays, the Ethernet protocol has become the de-facto standard for local area networks. Via the Internet, connectivity is occurring everywhere, from home appliances, to vending machines, to testing equipment, to UPS ...etc. An Ethernet network can link office automation and industrial control networks, access remote systems and share data and information between multivendor machines; it also provides a cost-effective solution for industrial control networks.



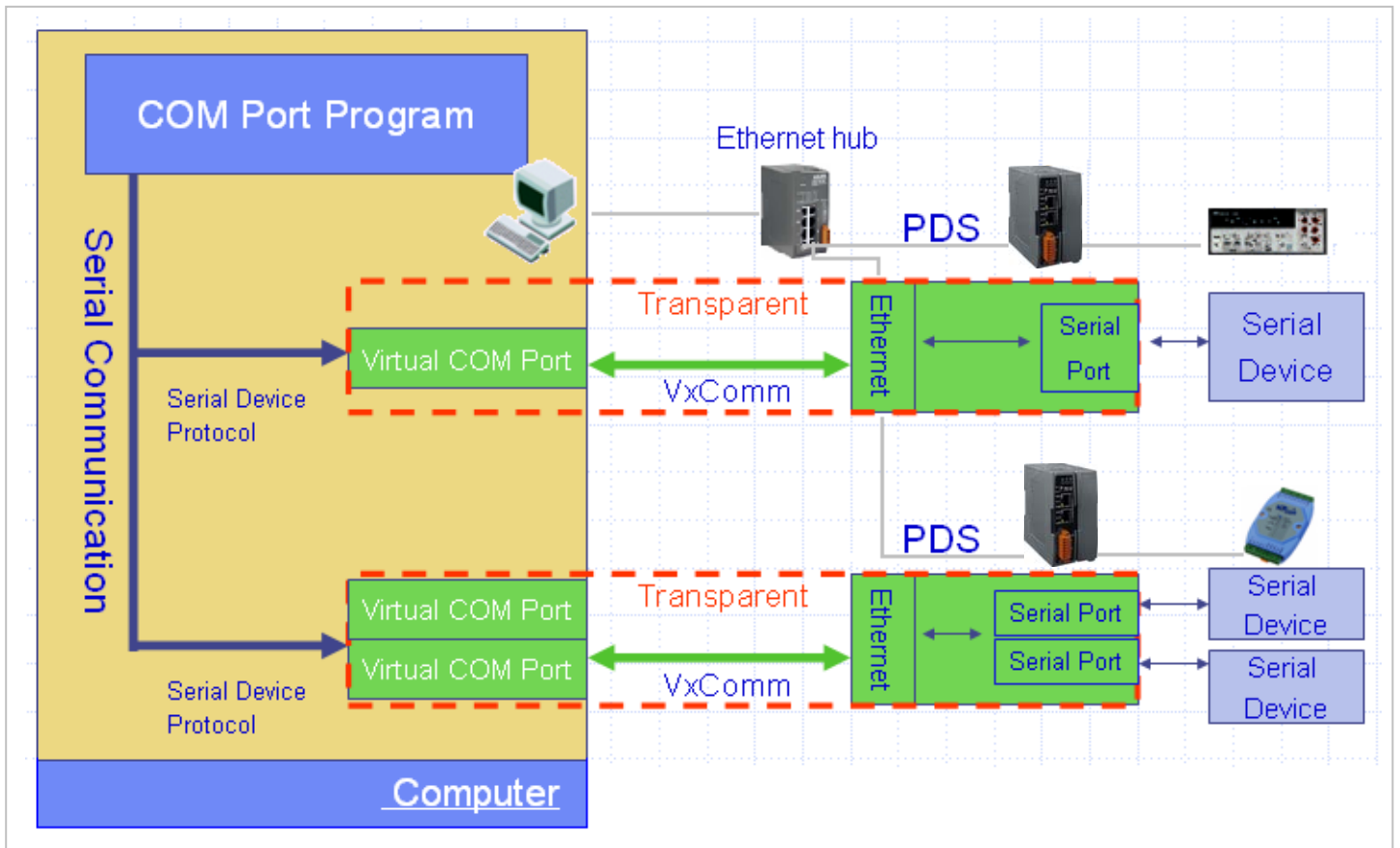
1.2 Why VxComm Technology?

In general, writing a TCP/IP program is more difficult than a COM port program, or the COM port communication system was built many years ago.

As a result, a new technology, **VxComm** was developed to **virtualizes the COM ports of the PDS-800 to allow up to 1 ~ 256 COM Ports to be used on the central computer.** The VxComm driver saves time when accessing serial devices through the Ethernet without the need for reprogramming the COM port software on the PC.

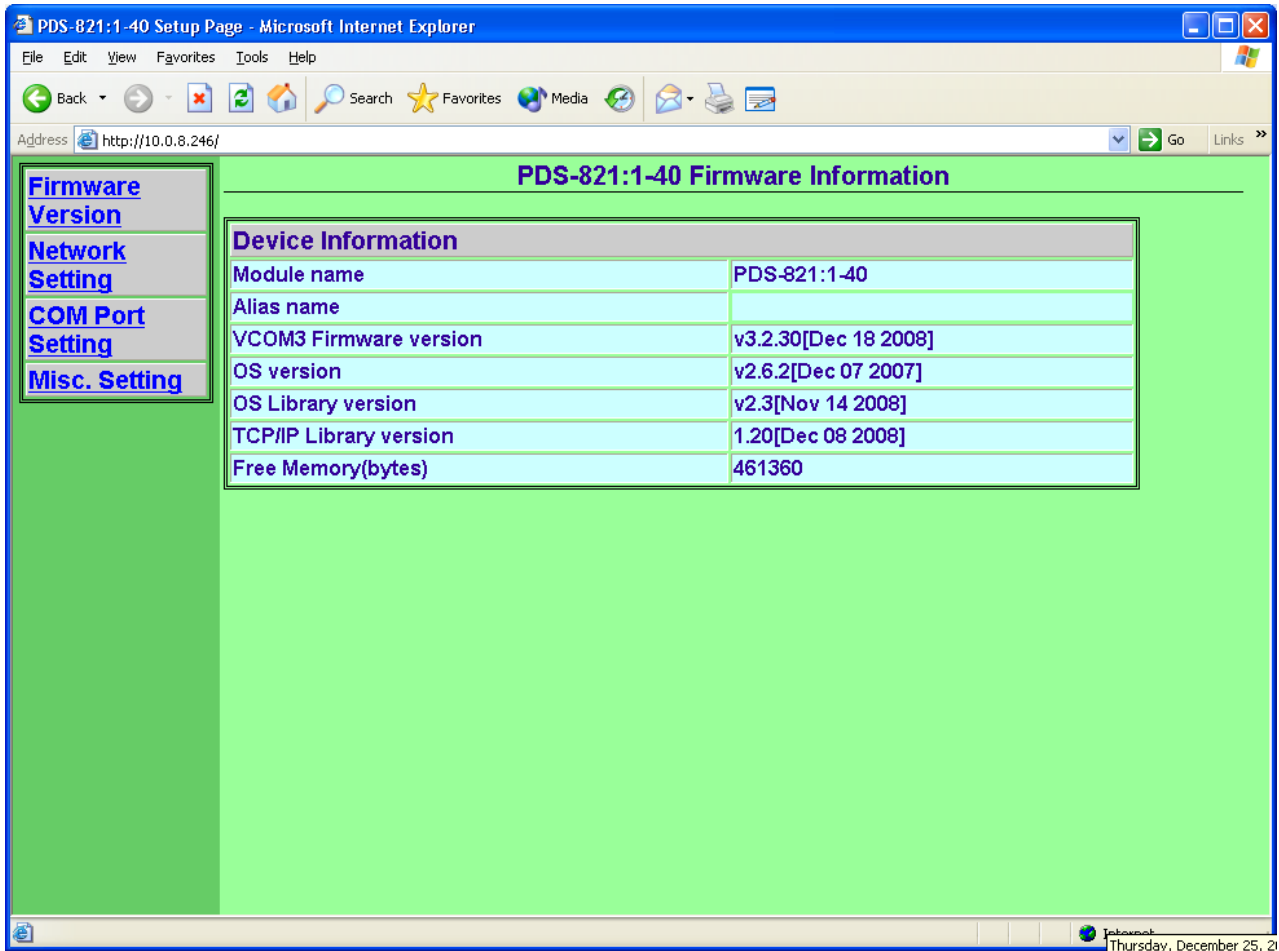


The VxComm driver controls all the details of the Ethernet TCP/IP programming technique; your COM port program will be able to access your serial devices through Ethernet in the same way as through COM port with the assistance of PDS-800 and VxComm technology.



1.3 Why Web Server Technology?

Web server technology enables configuration of the PDS-800 via a standard web browser interface, e.g. Internet Explorer, FireFox or Mozilla, etc. This means that it is easy to check the configuration of the PDS-800 via an Ethernet network without needing to install any other software tools; thereby reducing the user's learning curve.

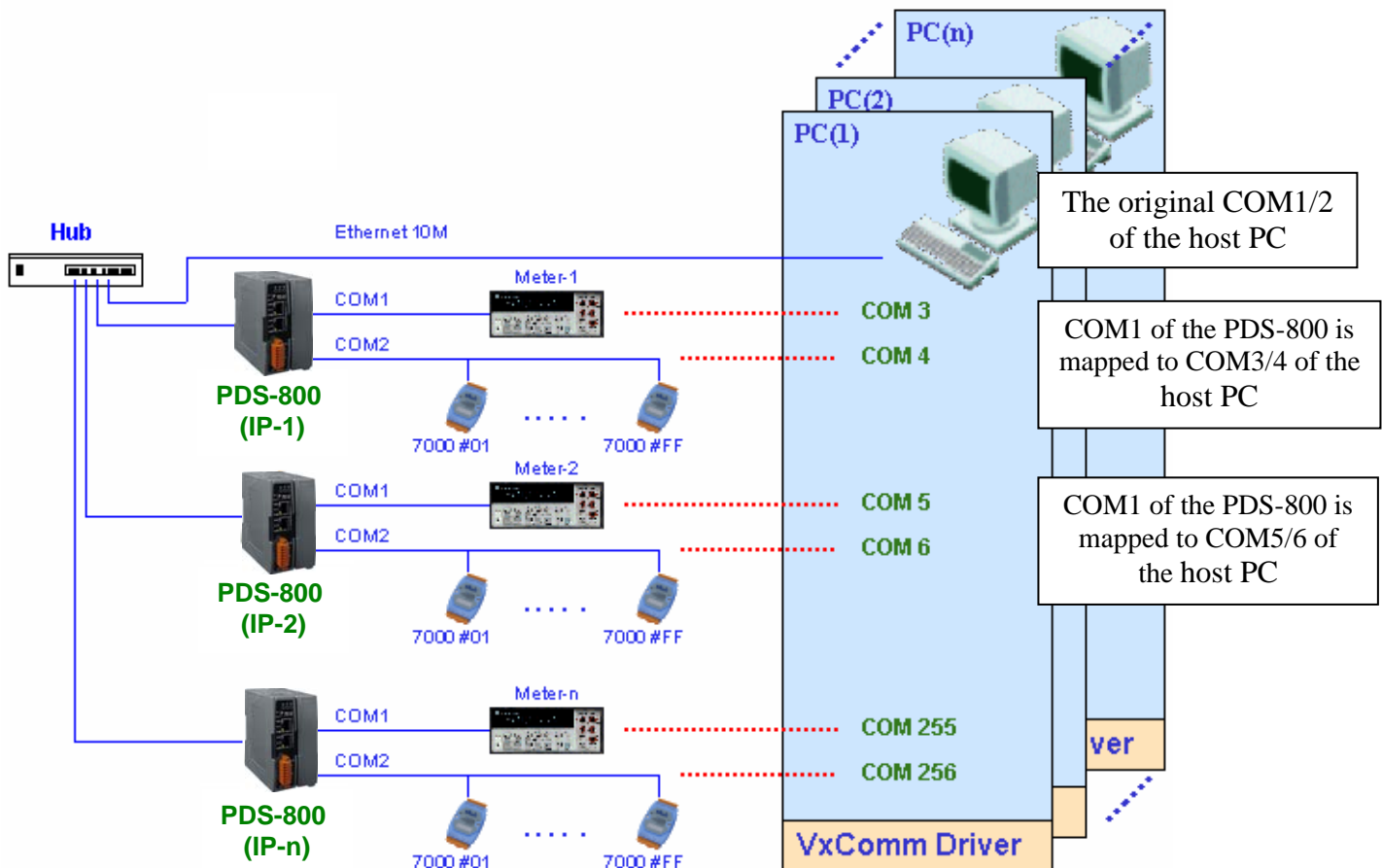


2. Typical Applications for the PDS-800

2.1 RS-232/485/422 Device Networking

--- Using Virtual COM Technology ---

The PDS-800 series is designed to link RS-232/485/422 devices to an Ethernet network. The VxComm utility allows the built-in PDS-800 COM Port to be virtualized to a standard COM Port of the host PC as shown below:



In the configuration above, Meter-1 is virtualized to link to COM3 of the host PC. Therefore a program original designed for the MS-COMM standard can access the meter without any modification.

2.2 Ethernet I/O Applications

The PDS-800 series provides of Ethernet I/O solutions:
Linking to I-7000 series modules

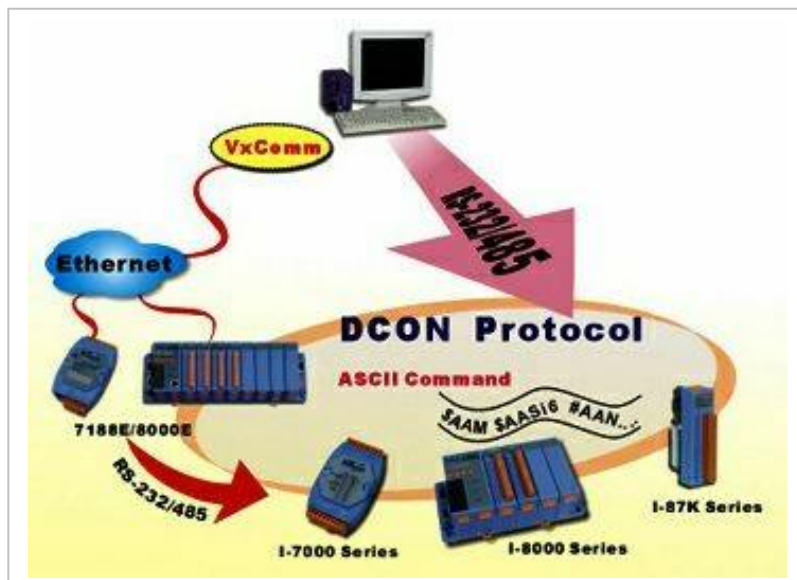
Linking to I-7000 series modules

The I-7000 series provides a variety of I/O operations, such as D/I, D/O, A/D, D/A, Counter and Frequency Measurement, etc. The I-7000 series was originally designed to be used with RS-485 networks, so RS-485 of COM on the PDS-800 can be used to link to I-7000 series modules.

By using VxComm technology, programs that on the host PC support serial devices can be upgraded from a RS-485 network to an Ethernet network without requiring any modifications to the program. Refer to Sec. 2.1 for more information.

DCON Protocol

The DCON protocol is a request /reply communication. Protocol that is defined using a simple ASCII format, such as \$AAN, \$AASi6, #AAN, etc. and is used to access I-7000/8000/87k series I/O modules.



2.3 Linking I-7000 Series Modules to an Ethernet

Network

The I-7000 family was originally designed for use with an RS-485 network. They are very robust and work well under the harsh industrial environments.

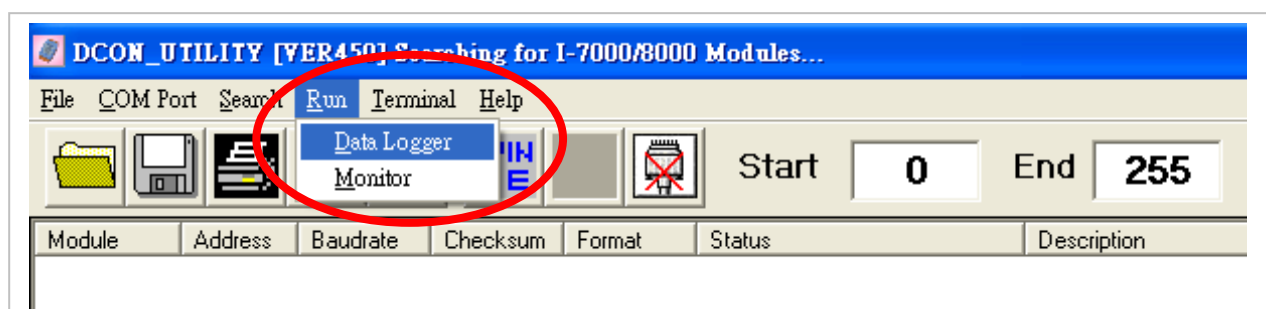
The PDS-800 enables I-7000 modules to be upgraded to an Ethernet solution. Linking I-7000 modules to an Ethernet combines the advantages of both RS-485 and Ethernet solutions and expands RS-485 applications to the whole world.

The VxComm approach provides an MS-COMM compatible interface. Therefore, previously developed programs should still function without the need for any modifications.

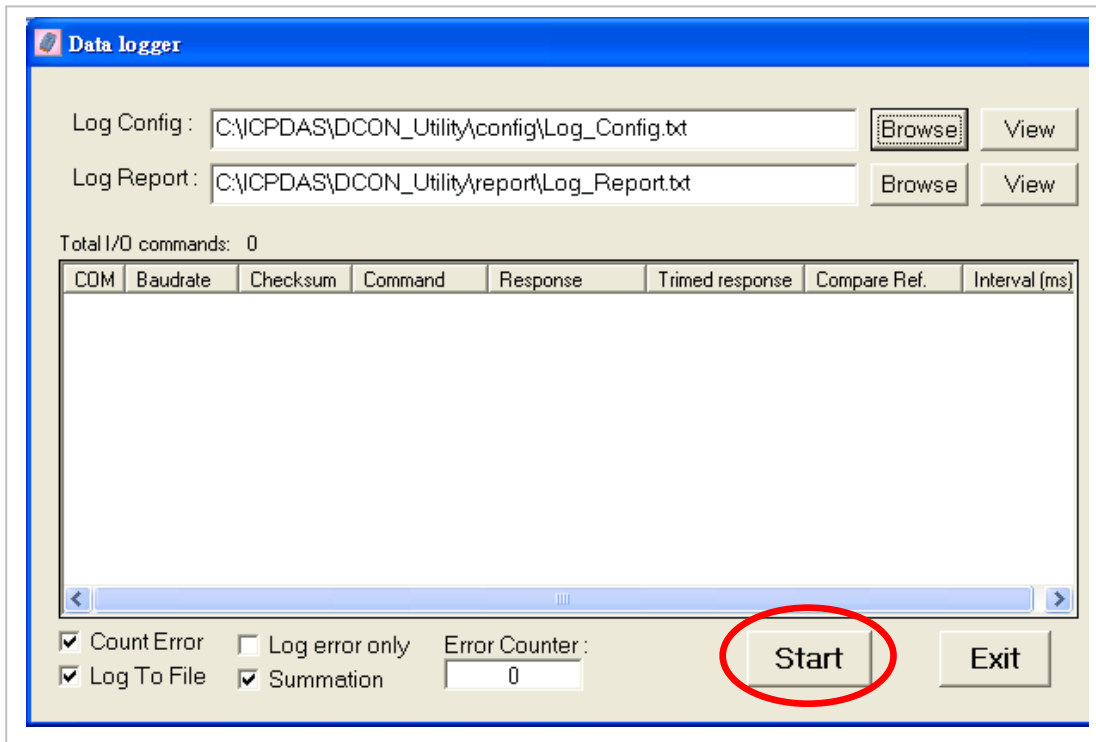
2.4 Configurable Ethernet Data Logger

Using the VxComm driver, PDS-800 + 7000 modules can be virtualized to become COM Port + 7000 modules located on the host-PC, and then the Data Logger in the DCON Utility can be used to access data of I-7000 from the Ethernet. Signal data originating from the I-7000 modules can be analyzed using MS-Excel without the need to write any custom programs

1: The DCON utility includes a log function, as show below:



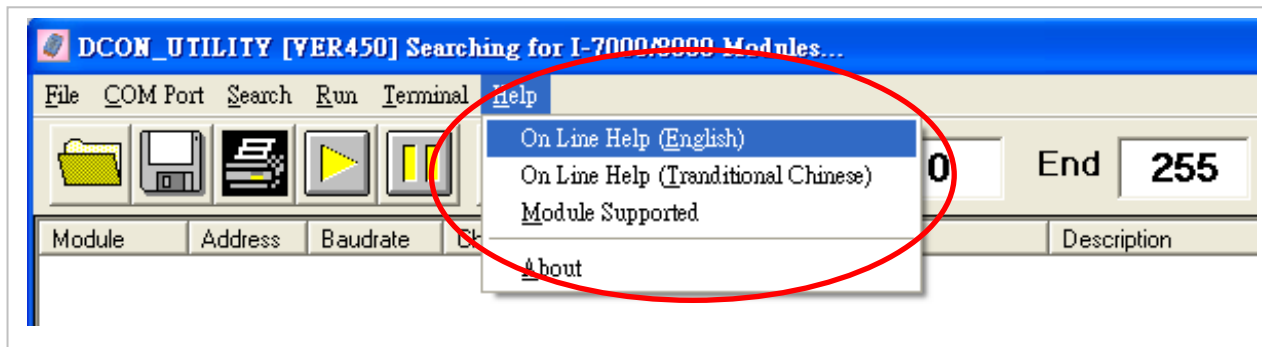
2: Configure the system connection as shown below and click the “Start” button to begin logging data.



3: Open the log file in Excel to read the log data as shown in the example below:

	A	B	C	D	E	F	G	H	I	J
1	Start log at	11/26/01	#####							
2	14:36:1:0	2	9600	0	#010	>+000.00	>+000.62	1000		
3	14:36:2:40	2	9600	0	#010	>+000.00	>+000.65	1000		
4	14:36:3:30	2	9600	0	#010	>+000.00	>+000.65	1000		
5	14:36:4:20	2	9600	0	#010	>+000.00	>+000.60	1000		
6	14:36:5:10	2	9600	0	#010	>+000.00	>+000.66	1000		
7	14:36:6:0	2	9600	0	#010	>+000.00	>+000.66	1000		
8	14:36:7:40	2	9600	0	#010	>+000.00	>+000.66	1000		
9	14:36:8:30	2	9600	0	#010	>+000.00	>+000.71	1000		
10	14:36:9:20	2	9600	0	#010	>+000.00	>+000.69	1000		
11	14:36:10:1	2	9600	0	#010	>+000.00	>+000.67	1000		
12	14:36:11:0	2	9600	0	#010	>+000.00	>+000.71	1000		
13	14:36:12:4	2	9600	0	#010	>+000.00	>+000.65	1000		
14	14:36:13:3	2	9600	0	#010	>+000.00	>+000.72	1000		
15	14:36:14:2	2	9600	0	#010	>+000.00	>+000.66	1000		
16	14:36:15:1	2	9600	0	#010	>+000.00	>+000.60	1000		
17	14:36:16:0	2	9600	0	#010	>+000.00	>+000.70	1000		

By using the I-7000 DCON utility and MS Excel in conjunction with the **VxComm technology**, the signal data of I-7000 modules from the Ethernet network can be analyzed without the need to write custom programs. For more information about the **log function** refer to the online help feature (English and Traditional Chinese) of the DCON utility.



3. Hardware information

3.1 Features

- Networkable serial devices
- “Virtual COM” extend PC COM ports
- VxComm Driver for Windows NT 4.0, 2000/XP/2003 and Vista32.
- Programmable Internet/Ethernet controller
- Watchdog timer suitable for use in harsh environments
- 2-Port 10/100 Base-TX Ethernet Switch (Auto-negotiating, auto MDI/MDI-X, LED indicator)
- Power reverse polarity protection circuit
- 3-Wire RS-232 console port
- RS-232 TxD/RxD LED indicators
- System status LED indicator
- ESD protection and frame ground design
- RoHS compliant with no halogen
- Built-in high performance MiniOS7 operating system.

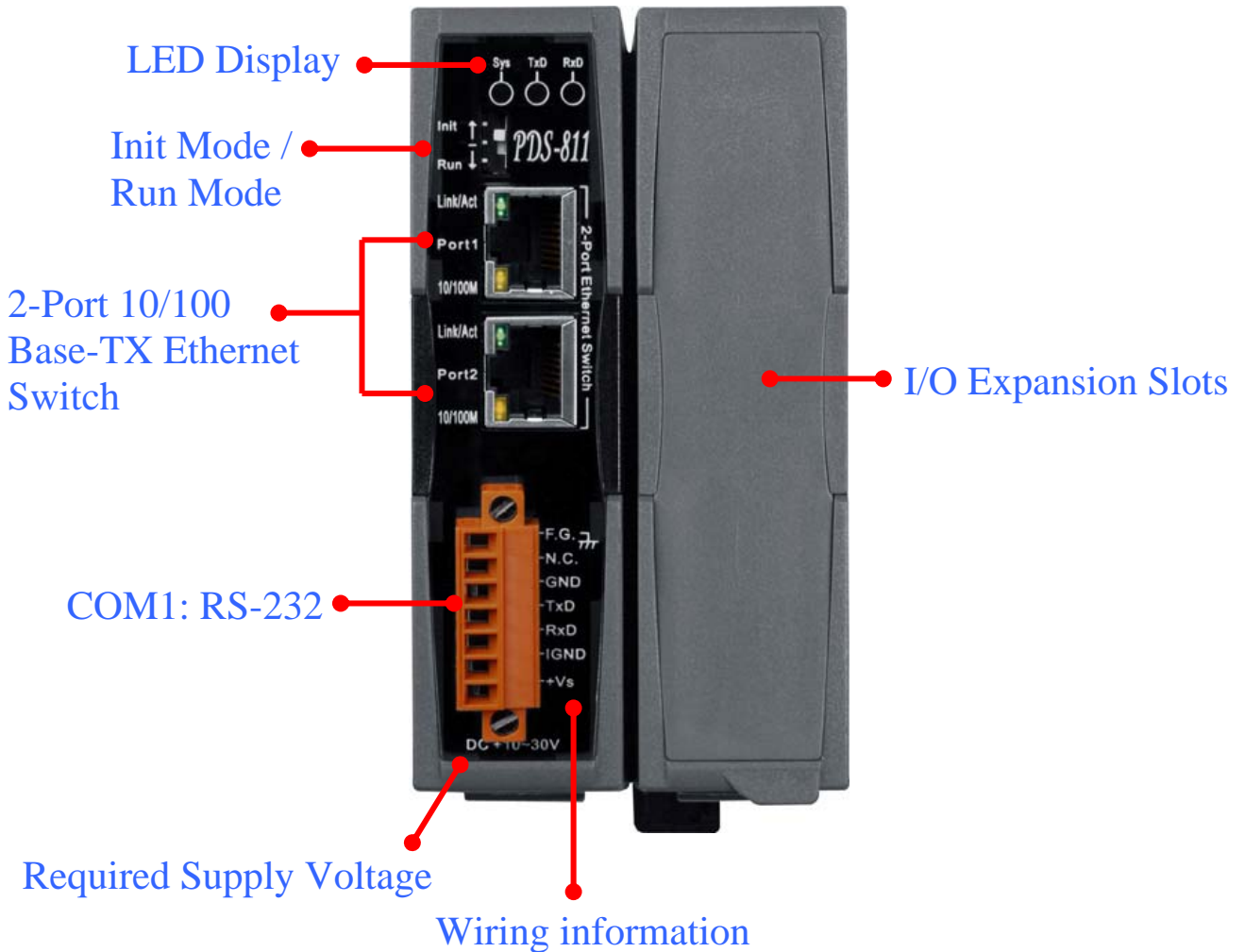
3.2 Applications

- Factory Automation
- Building Automation
- Home Automation

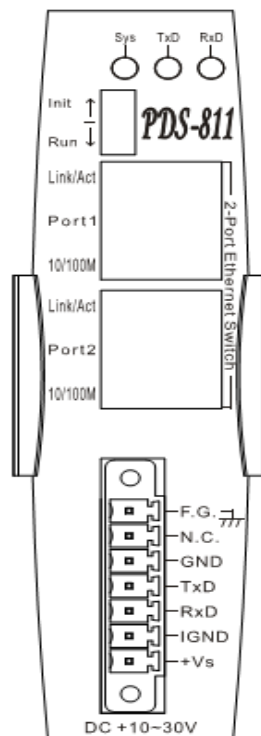
3.3 Specifications

CPU	
CPU	80186, 80 MHz or compatible
SRAM	512 KB
Flash Memory	512 KB
EEPROM	16 KB
Built-in Watchdog Timer	Yes
I/O Expansion Slots	PDS-811: 1 Slot PDS-821: 2 Slots
Communication Interface	
COM1 (Console)	RS-232 (TXD, RXD, GND)
Ethernet	2-Port 10/100 Base-TX Ethernet Switch (Auto-negotiating, Auto_MDIX, LED indicator)
COM Port Formats	
Data bit	7, 8
Parity	Even, Odd, None
Stop bit	1
LED Display	
TxD/RxD LED Indicators	Yes (for COM1 console port)
System LED Indicator	Yes
Operating Environment	
Operating Temperature	-25 °C ~ 75 °C
Storage Temperature	-40 °C ~ 80 °C
Power	
ESD Protection	Yes (with Frame Ground)
Protection	Power reverse polarity protection
Required Supply Voltage	10 ~ 30 V _{DC} (non-regulated)
Power consumption	2.0 W
Dimensions (W x H x D)	
PDS-811	64 mm x 132 mm x 110 mm
PDS-821	95 mm x 132 mm x 110 mm

3.4 Front View



3.5 Pin Assignment

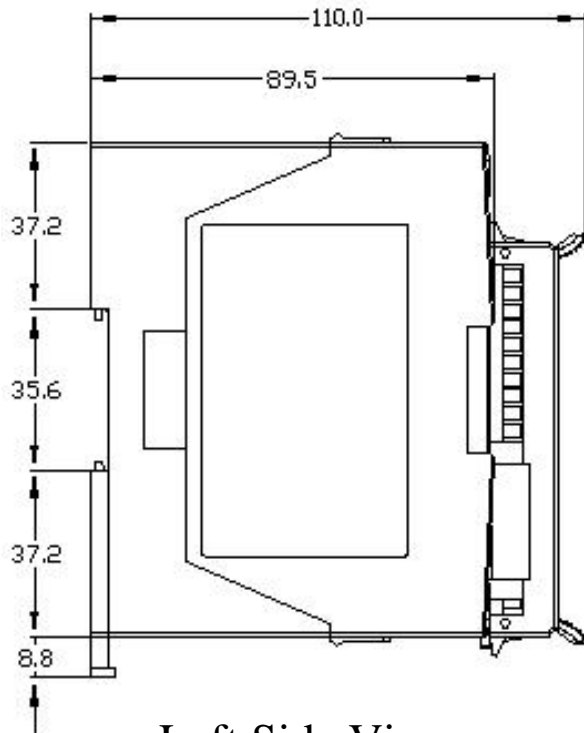


Note:

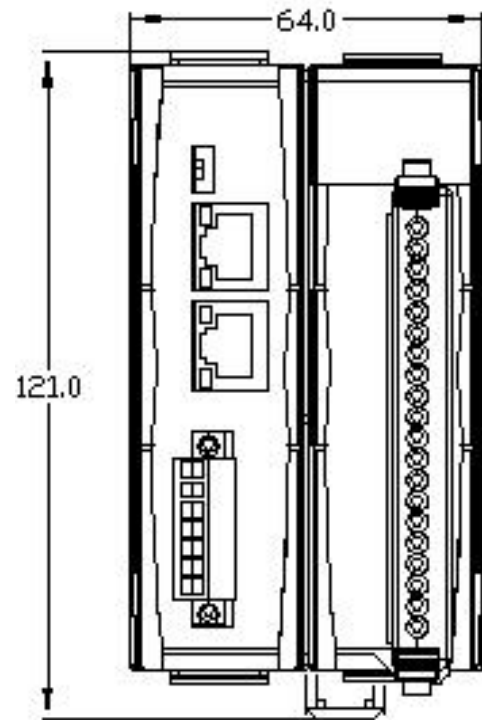
- F.G. : Frame Ground
- N.C. : Not Connected
- GND : Signal Ground
- TxD : COM1 TxD pin (RS-232)
- RxD : COM1 RxD pin (RS-232)
- IGND : Power Ground
- +Vs : Power supply +10 ~ +30 V_{DC}

3.6 Dimensions

● PDS-811

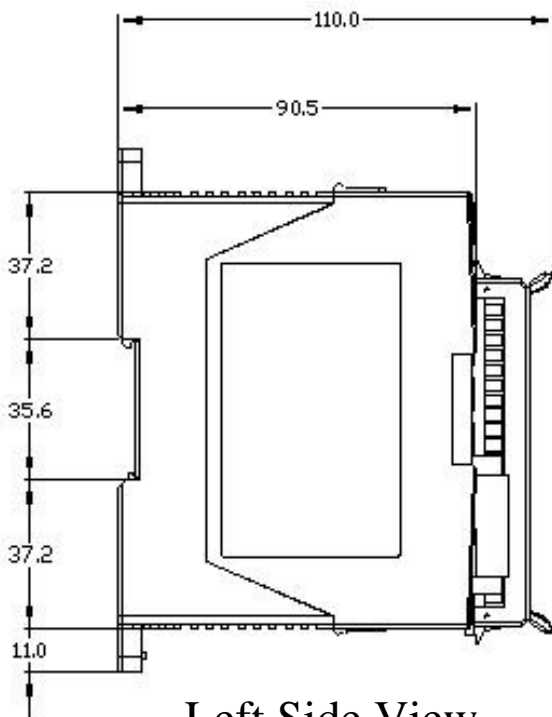


Left Side View

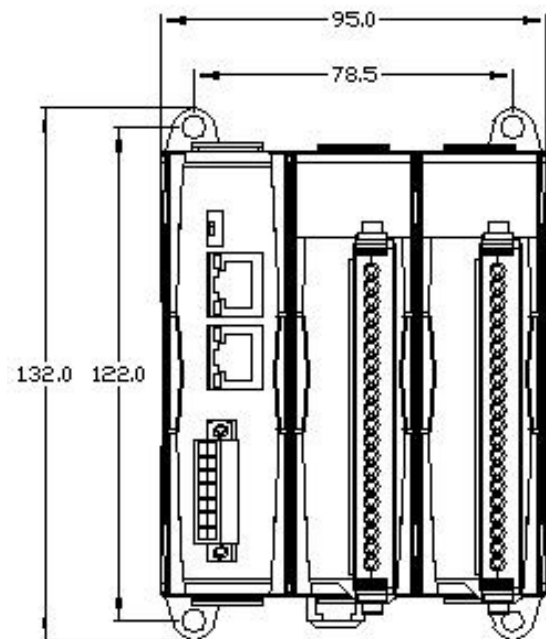


Front View

● PDS-821



Left Side View



Front View

3.7 PDS-800 Selection Guide

Model	PDS-811	PDS-821
Slots	1	2
CPU	80186 (80 MHz)	80186 (80 MHz)
RAM/Flash Disk	512 KB/ 512 KB	512 KB/ 512 KB
Ethernet	2-Port Ethernet Switch	2-Port Ethernet Switch
Operating System	MiniOS7	MiniOS7
Console Port	3-Wire RS-232	3-Wire RS-232
(Optional) Max. Serial Ports	4	8

Optional Serial Modules:

Module	I-8112iW	I-8114W	I-8114iW	I-8142iW	I-8144iW
Interface	9-Wire RS-232	9-Wire RS-232	5-Wire RS-232	4-Wire RS-422 2-Wire RS-485	4-Wire RS-422 2-Wire RS-485
Ports	2	4	4	2	4
FIFO	128 Bytes	128 Bytes	128 Bytes	128 Bytes	128 Bytes
Isolation	3000 V	-	3000 V	3000 V	3000 V
Self-tuner	-	-	-	Yes	Yes
Connector	DB-9	DB-37	DB-37	Terminal Block	Terminal Block

2-wire RS-485: Data+, Data- with Self-Tuner inside

4-wire RS-422: TxD+, TxD-, RxD+, RxD-

3-wire RS-232: RxD, TxD, GND

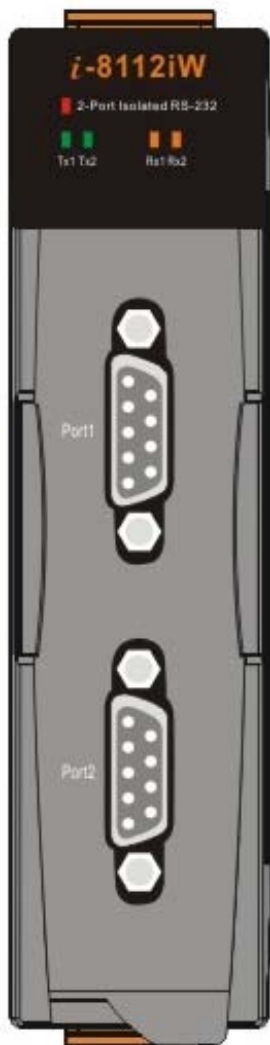
5-wire RS-232: RxD, TxD, CTS, RTS, GND

9-wire RS-232: RxD, TxD, CTS, RTS, DSR, DTR, DCD, RI, GND

3.8 Optional Serial Modules Pin Assignments

3.8.1 I-8112iW Pin Assignments

2-Port Isolated RS-232 Module



Pin Assignment Name	Terminal No.	Pin Assignment Name
GND1	05	
DTR1	04	RI1
TxD1	03	CTS1
RxD1	02	RTS1
DCD1	01	DSR1

DB-9 Male Connector(Port1)

Pin Assignment Name	Terminal No.	Pin Assignment Name
GND2	05	
DTR2	04	RI2
TxD2	03	CTS2
RxD2	02	RTS2
DCD2	01	DSR2

DB-9 Male Connector(Port2)

DB-9 Male Connector (Port 1)

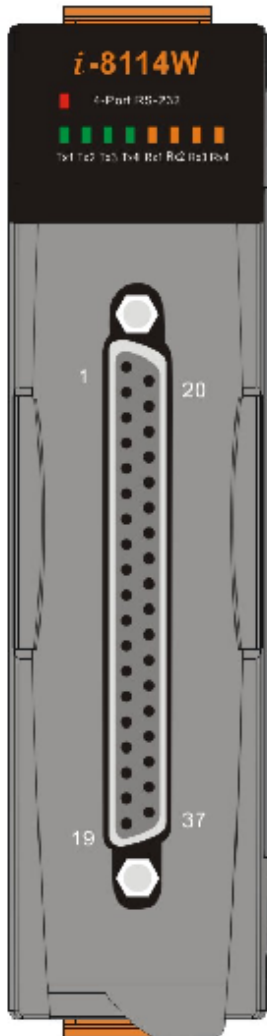
Pin	Name	Description
01	DCD1	Port1 DCD pin (RS-232)
02	RxD1	Port1 RxD pin (RS-232)
03	TxD1	Port1 TxD pin (RS-232)
04	DTR1	Port1 DTR pin (RS-232)
05	GND1	Power's Ground (Port1 GND)
06	DSR1	Port1 DSR pin (RS-232)
07	RTS1	Port1 RTS pin (RS-232)
08	CTS1	Port1 CTS pin (RS-232)
09	RI 1	Port1 RI pin (RS-232)

DB-9 Male connector (Port 2)

Pin	Name	Description
01	DCD2	Port2 DCD pin (RS-232)
02	RxD2	Port2 RxD pin (RS-232)
03	TxD2	Port2 TxD pin (RS-232)
04	DTR2	Port2 DTR pin (RS-232)
05	GND2	Power's Ground (Port2 GND)
06	DSR2	Port2 DSR pin (RS-232)
07	RTS2	Port2 RTS pin (RS-232)
08	CTS2	Port2 CTS pin (RS-232)
09	RI 2	Port2 RI pin (RS-232)

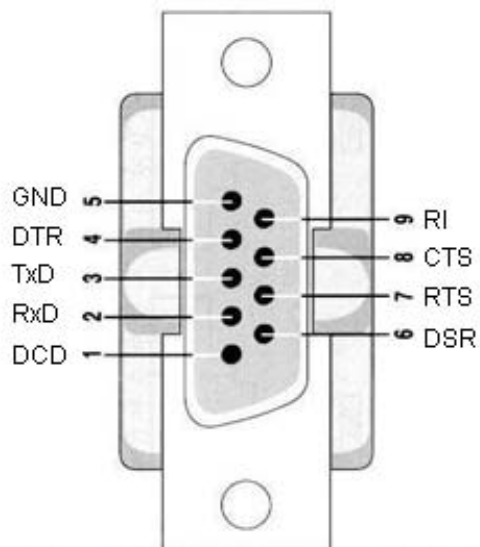
3.8.2 I-8114W Pin Assignments

4-Port RS-232 Module



Pin Assignment Name	Terminal No.	Pin Assignment Name
N.C.	01	
DCD3	02	RI3
GND	03	DTR3
CTS3	04	DSR3
RxD3	05	RTS3
RI4	06	TxD3
DTR4	07	DCD4
DSR4	08	GND
RTS4	09	CTS4
TxD4	10	RxD4
DCD2	11	RI2
GND	12	DTR2
CTS2	13	DSR2
RxD2	14	RTS2
RI1	15	TxD2
DTR1	16	DCD1
DSR1	17	GND
RTS1	18	CTS1
TxD1	19	RxD1

37-Pin Female D-Sub Connector(Port1~Port4)



DB-37 to Male DB-9 Connector_RS232

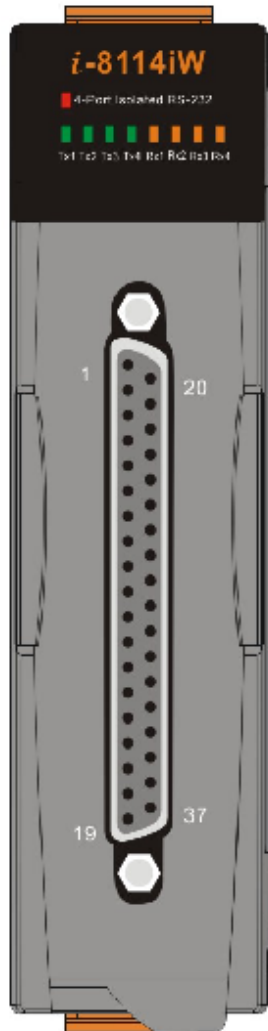
37-Pin Female D-Sub Connector for Port1~Port4

Pin	Name	Description	Pin	Name	Description
01	N.C.	Not Connected	20	RI3	Port3 RI pin (RS-232)
02	DCD3	Port3 DCD pin (RS-232)	21	DTR3	Port3 DTR pin (RS-232)
03	GND	Power's Ground	22	DSR3	Port3 DSR pin (RS-232)
04	CTS3	Port3 CTS pin (RS-232)	23	RTS3	Port3 RTS pin (RS-232)
05	RxD3	Port3 RxD pin (RS-232)	24	TxD3	Port3 TxD pin (RS-232)
06	RI4	Port4 RI pin (RS-232)	25	DCD4	Port4 DCD pin (RS-232)
07	DTR4	Port4 DTR pin (RS-232)	26	GND	Power's Ground
08	DSR4	Port4 DSR pin (RS-232)	27	CTS4	Port4 CTS pin (RS-232)
09	RTS4	Port4 RTS pin (RS-232)	28	RxD4	Port4 RxD pin (RS-232)
10	TxD4	Port4 TxD pin (RS-232)	29	RI2	Port2 RI pin (RS-232)
11	DCD2	Port2 DCD pin (RS-232)	30	DTR2	Port2 DTR pin (RS-232)
12	GND	Power's Ground	31	DSR2	Port2 DSR pin (RS-232)
13	CTS2	Port2 CTS pin (RS-232)	32	RTS2	Port2 RTS pin (RS-232)
14	RxD2	Port2 RxD pin (RS-232)	33	TxD2	Port2 TxD pin (RS-232)
15	RI1	Port1 RI pin (RS-232)	34	DCD1	Port1 DCD pin (RS-232)
16	DTR1	Port1 DTR pin (RS-232)	35	GND	Power's Ground
17	DSR1	Port1 DSR pin (RS-232)	36	CTS1	Port1 CTS pin (RS-232)
18	RTS1	Port1 RTS pin (RS-232)	37	RxD1	Port1 RxD pin (RS-232)
19	TxD1	Port1 TxD pin (RS-232)			

DB-37 to Male Db-9 Connector for Port1~Port4

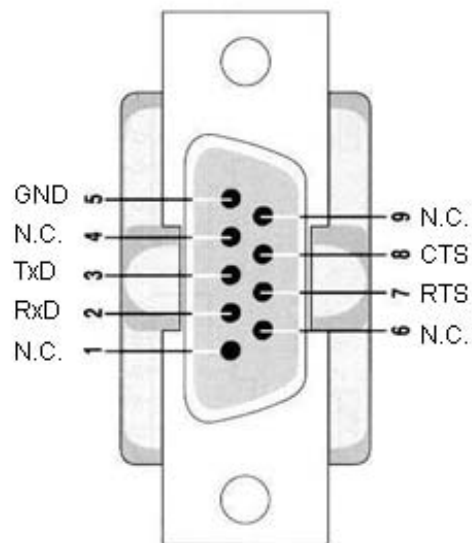
Pin	Name	Description
01	DCD	Port1~Port4 DCD pin (RS-232)
02	RxD	Port1~Port4 RxD pin (RS-232)
03	TxD	Port1~Port4 TxD pin (RS-232)
04	DTR	Port1~Port4 DTR pin (RS-232)
05	GND	Port1~Port4 Power's Ground
06	DSR	Port1~Port4 DSR pin (RS-232)
07	RTS	Port1~Port4 RTS pin (RS-232)
08	CTS	Port1~Port4 CTS pin (RS-232)
09	RI	Port1~Port4 RI pin (RS-232)

3.8.3 I-8114iW Pin Assignments 4-Port RS-232 Module



Pin Assignment Name	Terminal No.	Pin Assignment Name
N.C.	01	
N.C.	02	20
GND3	03	21
CTS3	04	22
RxD3	05	23
N.C.	06	24
N.C.	07	25
N.C.	08	26
RTS4	09	27
TxD4	10	28
N.C.	11	29
GND2	12	30
CTS2	13	31
RxD2	14	32
N.C.	15	33
N.C.	16	34
N.C.	17	35
RTS1	18	36
TxD1	19	37

37-Pin Female D-Sub Connector(Port1~Port4)



DB-37 to Male DB-9 Connector_RS232

37-Pin Female D-Sub Connector for Port1~Port4

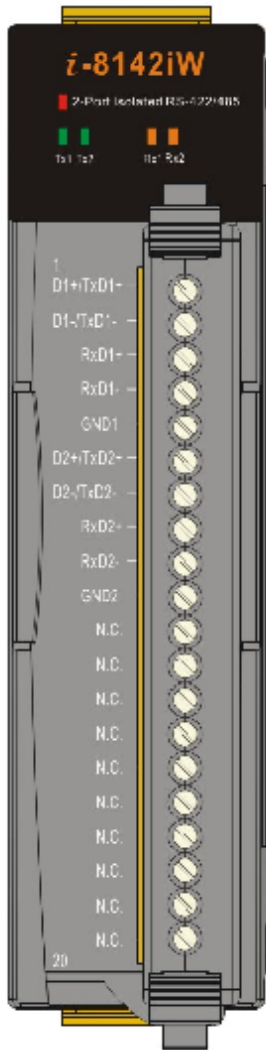
Pin	Name	Description	Pin	Name	Description
01	N.C.	Not Connected	20	N.C.	Not Connected
02	N.C.	Not Connected	21	N.C.	Not Connected
03	GND3	Power's Ground (Port3 GND)	22	N.C.	Not Connected
04	CTS3	Port3 CTS pin (RS-232)	23	RTS3	Port3 RTS pin (RS-232)
05	RxD3	Port3 RxD pin (RS-232)	24	TxD3	Port3 TxD pin (RS-232)
06	N.C.	Not Connected	25	N.C.	Not Connected
07	N.C.	Not Connected	26	GND4	Power's Ground (Port4 GND)
08	N.C.	Not Connected	27	CTS4	Port4 CTS pin (RS-232)
09	RTS4	Port4 RTS pin (RS-232)	28	RxD4	Port4 RxD pin (RS-232)
10	TxD4	Port4 TxD pin (RS-232)	29	N.C.	Not Connected
11	N.C.	Not Connected	30	N.C.	Not Connected
12	GND2	Power's Ground (Port2 GND)	31	N.C.	Not Connected
13	CTS2	Port2 CTS pin (RS-232)	32	RTS2	Port2 RTS pin (RS-232)
14	RxD2	Port2 RxD pin (RS-232)	33	TxD2	Port2 TxD pin (RS-232)
15	N.C.	Not Connected	34	N.C.	Not Connected
16	N.C.	Not Connected	35	GND1	Power's Ground (Port1 GND)
17	N.C.	Not Connected	36	CTS1	Port1 CTS pin (RS-232)
18	RTS1	Port1 RTS pin (RS-232)	37	RxD1	Port1 RxD pin (RS-232)
19	TxD1	Port1 TxD pin (RS-232)			

DB-37 to Male Db-9 Connector for Port1~Port4

Pin	Name	Description
01	N.C.	Not Connected
02	RxD	Port1~Port4 RxD pin (RS-232)
03	TxD	Port1~Port4 TxD pin (RS-232)
04	N.C.	Not Connected
05	GND	Port1~Port4 Power's Ground
06	N.C.	Not Connected
07	RTS	Port1~Port4 RTS pin (RS-232)
08	CTS	Port1~Port4 CTS pin (RS-232)
09	N.C.	Not Connected

3.8.4 I-8142iW Pin Assignments

2-Port Isolated RS-422/485 Module

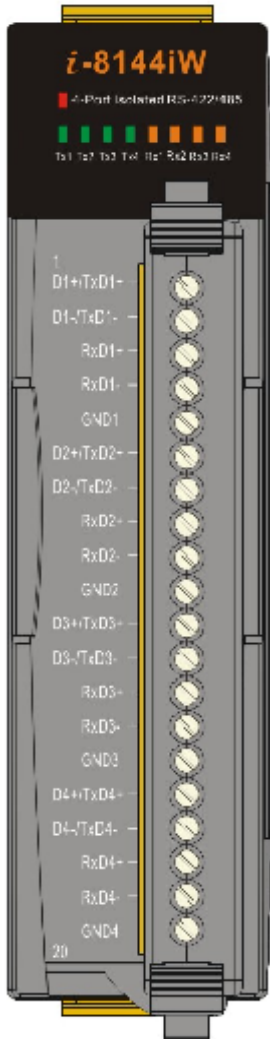


Terminal No.	Pin Assignment Name
01	D1+/TxD1+
02	D1-/TxD1-
03	RxD1+
04	RxD1-
05	GND1
06	D2+/TxD2+
07	D2-/TxD2-
08	RxD2+
09	RxD2-
10	GND2
11	N.C.
12	N.C.
13	N.C.
14	N.C.
15	N.C.
16	N.C.
17	N.C.
18	N.C.
19	N.C.
20	N.C.

Pin	Name	Description
01	D1+/TxD1+	Port1 Data+ pin (RS-485)/TxD+ pin (RS-422)
02	D1-/TxD1-	Port1 Data- pin (RS-485)/TxD- pin (RS-422)
03	RxD1+	Port1 RxD+ pin (RS-422)
04	RxD1-	Port1 RxD- pin (RS-422)
05	GND1	Power's Ground (Port1 GND)
06	D2+/TxD2+	Port2 Data+ pin (RS-485)/TxD+ pin (RS-422)
07	D2-/TxD2-	Port2 Data+ pin (RS-485)/TxD+ pin (RS-422)
08	RxD2+	Port2 RxD+ pin (RS-422)
09	RxD2-	Port2 RxD+ pin (RS-422)
10	GND2	Power's Ground (Port2 GND)
11	N.C.	Not Connected
12	N.C.	Not Connected
13	N.C.	Not Connected
14	N.C.	Not Connected
15	N.C.	Not Connected
16	N.C.	Not Connected
17	N.C.	Not Connected
18	N.C.	Not Connected
19	N.C.	Not Connected
20	N.C.	Not Connected

3.8.5 I-8144W Pin Assignments

4-Port Isolated RS-422/485 Module

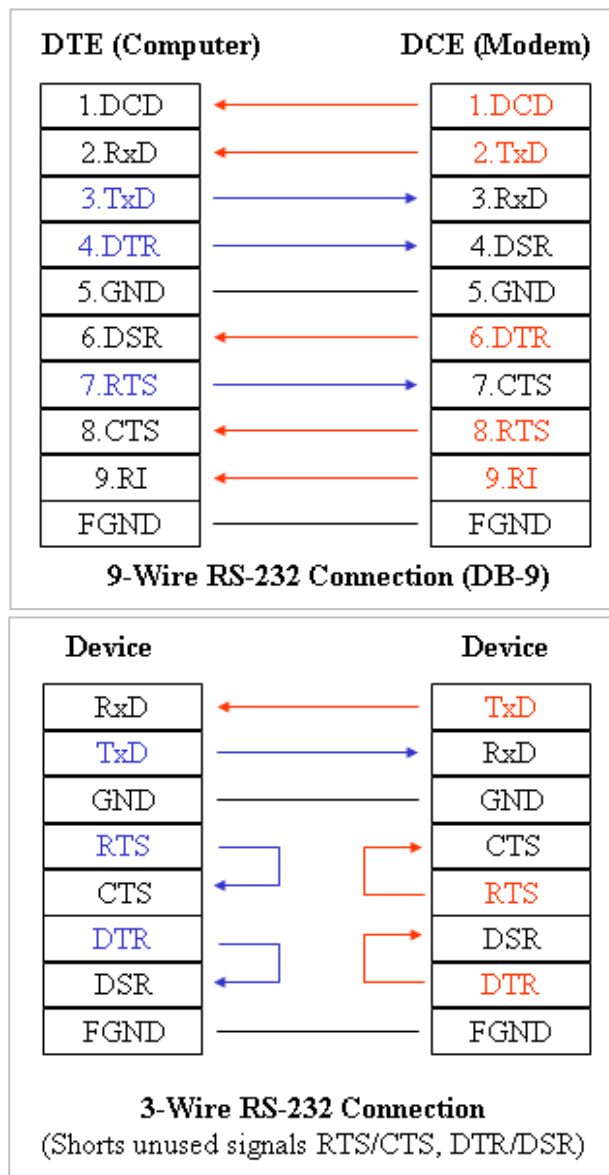


Terminal No.	Pin Assignment Name
01	D1+/TxD1+
02	D1-/TxD1-
03	RxD1+
04	RxD1-
05	GND1
06	D2+/TxD2+
07	D2-/TxD2-
08	RxD2+
09	RxD2-
10	GND2
11	D3+/TxD3+
12	D3-/TxD3-
13	RxD3+
14	RxD3-
15	GND3
16	D4+/TxD4+
17	D4-/TxD4-
18	RxD4+
19	RxD4-
20	GND4

Pin	Name	Description
01	D1+/TxD1+	Port1 Data+ pin (RS-485)/TxD+ pin (RS-422)
02	D1-/TxD1-	Port1 Data- pin (RS-485)/TxD- pin (RS-422)
03	RxD1+	Port1 RxD+ pin (RS-422)
04	RxD1-	Port1 RxD- pin (RS-422)
05	GND1	Power's Ground (Port1 GND)
06	D2+/TxD2+	Port2 Data+ pin (RS-485)/TxD+ pin (RS-422)
07	D2-/TxD2-	Port2 Data+ pin (RS-485)/TxD+ pin (RS-422)
08	RxD2+	Port2 RxD+ pin (RS-422)
09	RxD2-	Port2 RxD+ pin (RS-422)
10	GND2	Power's Ground (Port2 GND)
11	D3+/TxD3+	Port3 Data+ pin (RS-485)/TxD+ pin (RS-422)
12	D3-/TxD3-	Port3 Data+ pin (RS-485)/TxD+ pin (RS-422)
13	RxD3+	Port3 RxD+ pin (RS-422)
14	RxD3-	Port3 RxD+ pin (RS-422)
15	GND3	Power's Ground (Port3 GND)
16	D4+/TxD4+	Port4 Data+ pin (RS-485)/TxD+ pin (RS-422)
17	D4-/TxD4-	Port4 Data+ pin (RS-485)/TxD+ pin (RS-422)
18	RxD4+	Port4 RxD+ pin (RS-422)
19	RxD4-	Port4 RxD+ pin (RS-422)
20	GND4	Power's Ground (Port4 GND)

4 Wiring Note for RS-232 and RS-422/485 Devices

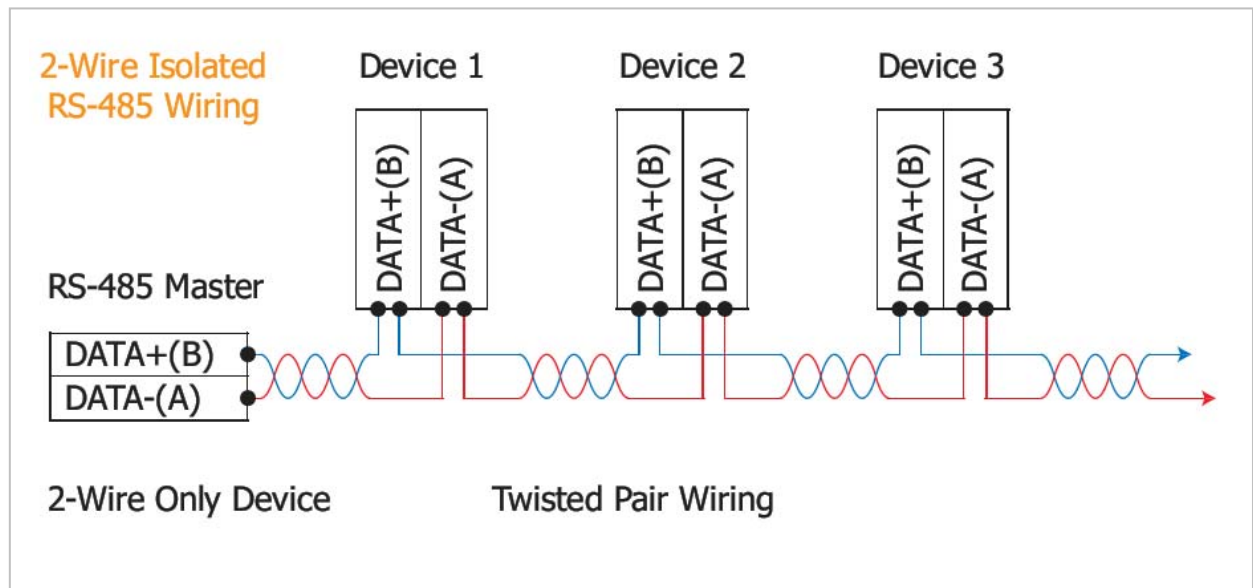
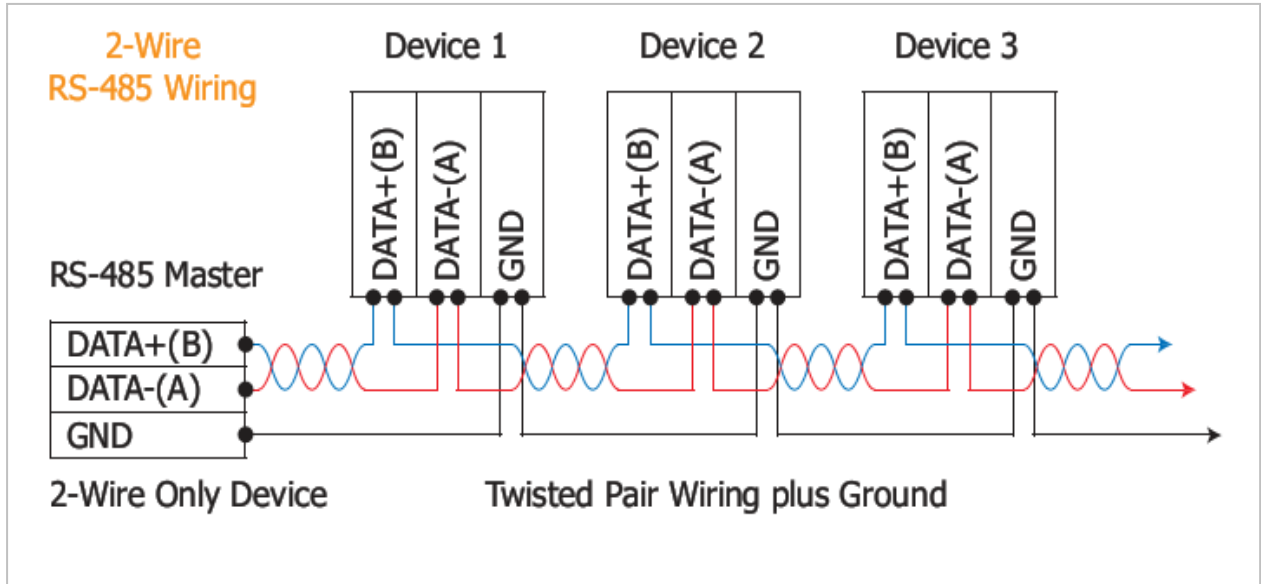
4.1 RS-232 Wire Connections



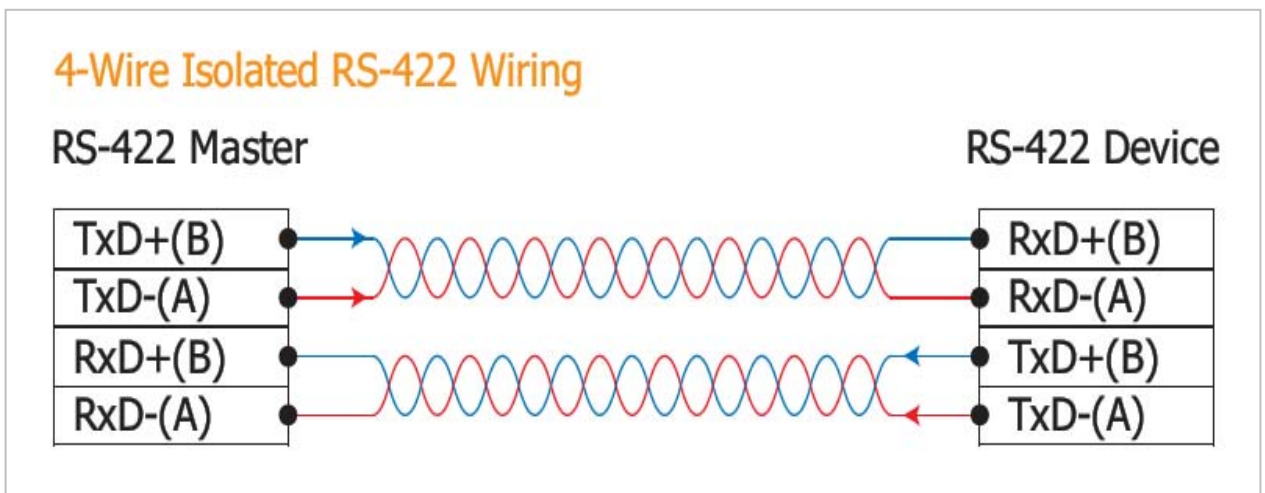
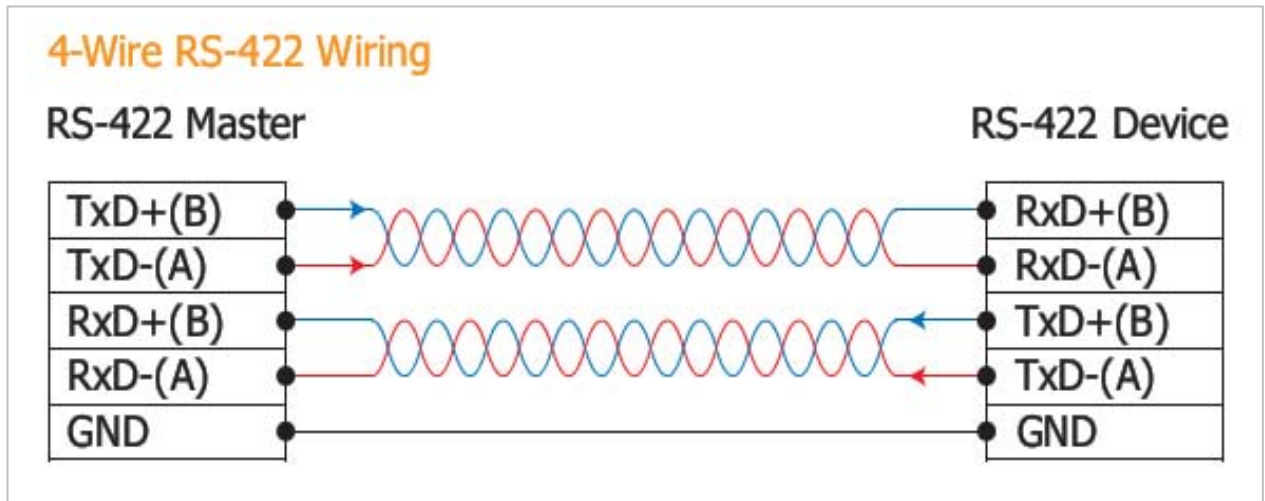
Note:

1. For 3-Wire RS-232 connections, it is recommended to short unused signals such as RTS/CTS and DTR/DSR, since some system may still check the CTS and DSR status.
2. FGND is the frame ground that soldered to DB9 metal-shield.

4.2 RS-485 Wire Connections



4.3 RS-422 Wire Connections



Note:

1. For **non-isolated** RS-422/485 ports, you should connect all signal grounds of RS-422/485 devices together. This reduces common-mode voltage between devices.
2. For **isolated** RS-422/485 ports, you should ****not**** connect all signal grounds or power grounds of RS-422/485 devices together. This ensures truly isolation between devices.

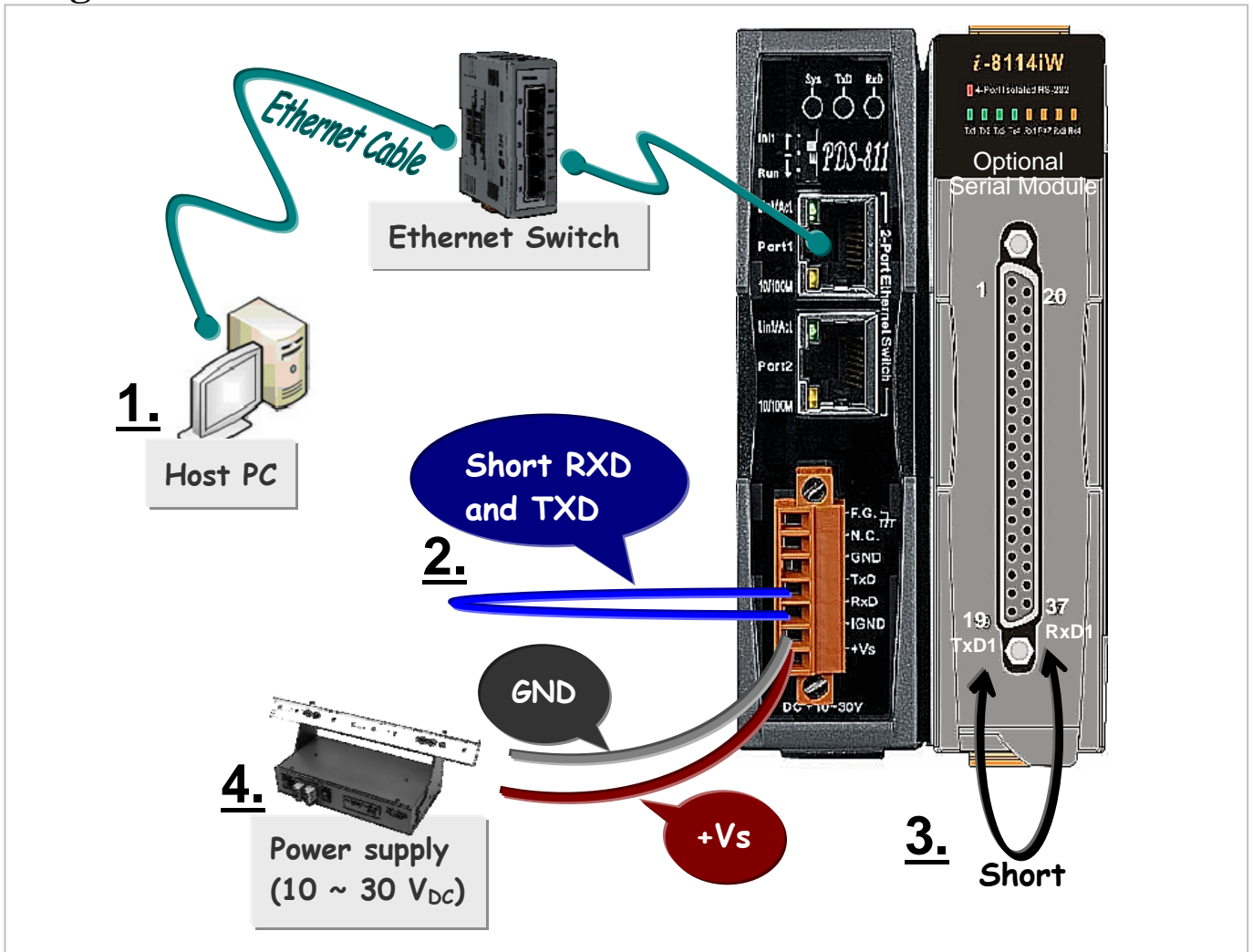
5 Setting up the PDS-800 controller

Step 1: Connect the PDS-800 controller to the Ethernet Network

Before connecting the PDS-800 controller to an Ethernet network, the following items are needed: (*For example: PDS-811*)

1. Inserts a serial module (optional) in PDS-800 controller. (*For example: I-8114iW*)
2. Power Supply: 10 ~ 30 V_{DC}
(*e.g.: DP-665: http://www.icpdas.com/products/Accessories/power_supply/power_list.htm*)
3. Hub
(*e.g.: NS-205: http://www.icpdas.com/products/Switch/industrial/industrial_list.htm*)
4. The network settings in the PC are correctly configured and the Ethernet connection is functioning normally.
5. Disable or correctly configure the Windows firewall and any Anti-Virus software firewall first or else the “**Search Servers**” function in the VxComm Utility may not work. (Contact your System Administrator for more details of how to do this.)
6. Connect the PDS-800 controller to the Ethernet as shown on the following page and switch on the power.
7. Make sure the System LED indicator is flashing.
8. Install VxComm Utility on your PC
The software is located at :
CD: \NAPDOS\Driver\VxComm_Driver
http://ftp.icpdas.com/pub/cd/8000cd/napdos/driver/vxcomm_driver/

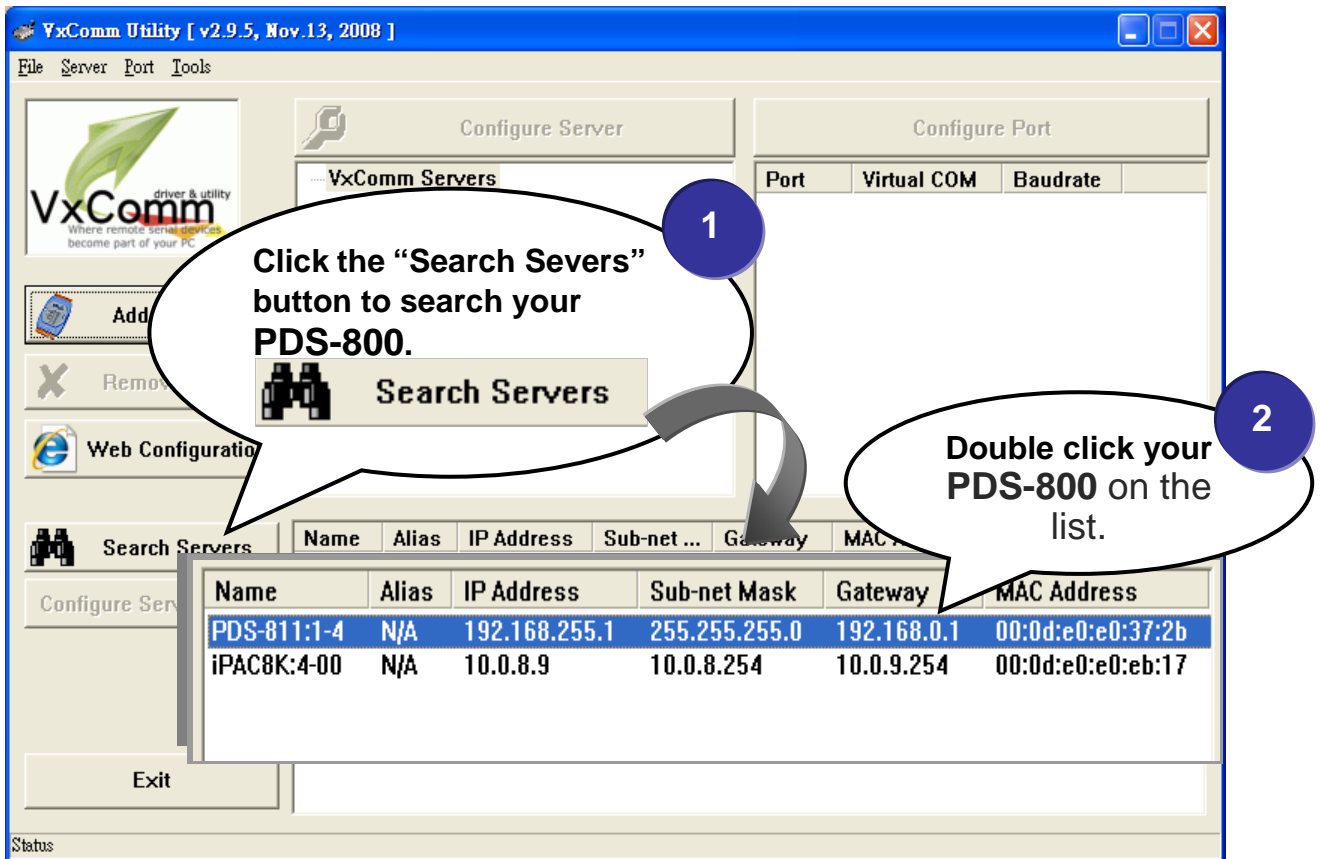
Diagram:



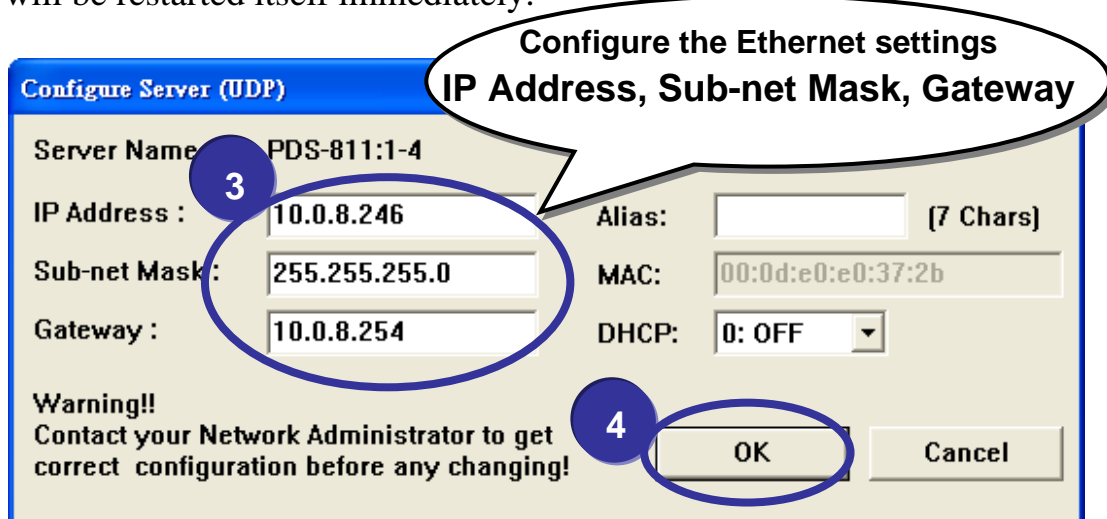
1. Connect both the PDS-800 controller and your computer to the same sub network or the same Ethernet Switch.
2. Short the RXD and TXD pins of the PDS-800 controller for self-test.
3. Short the TxD1 (Pin 19) and RxD1 (Pin 37) pins of the 4-Port I-8114iW module for example of self-test.
4. Supply 24 V_{DC} (10 ~ 30 V_{DC}) power to the PDS-800 controller.

Step 2: Search the PDS-800 controller on the Ethernet network

1. Execute the VxComm Utility and then search your PDS-800 controller.
2. Double click the name of the PDS-800 to open the configuration settings dialog box.

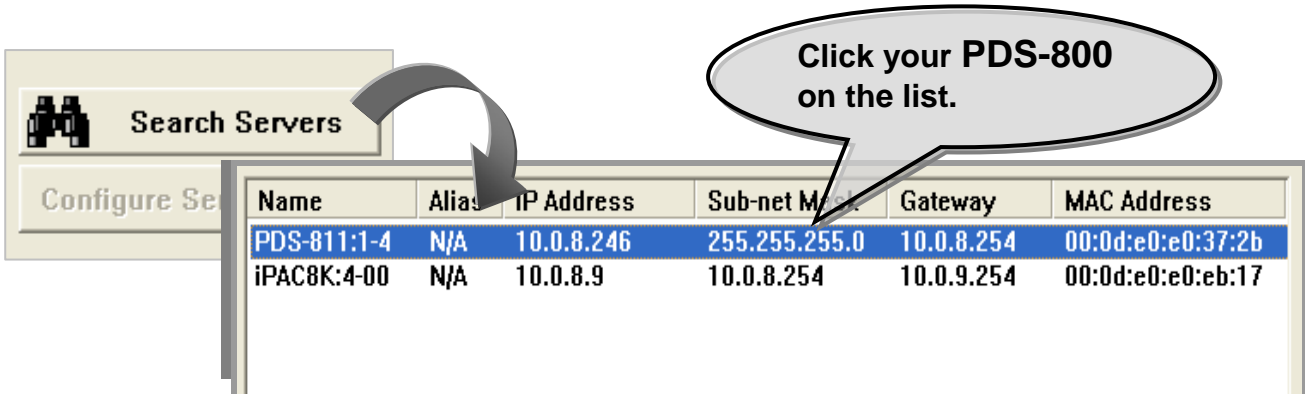


3. Contact your Network Administrator to obtain a correct network configuration (such as IP/Mask/Gateway). Enter the network settings and then click **“OK”**. The PDS-800 controller will be restarted itself immediately.

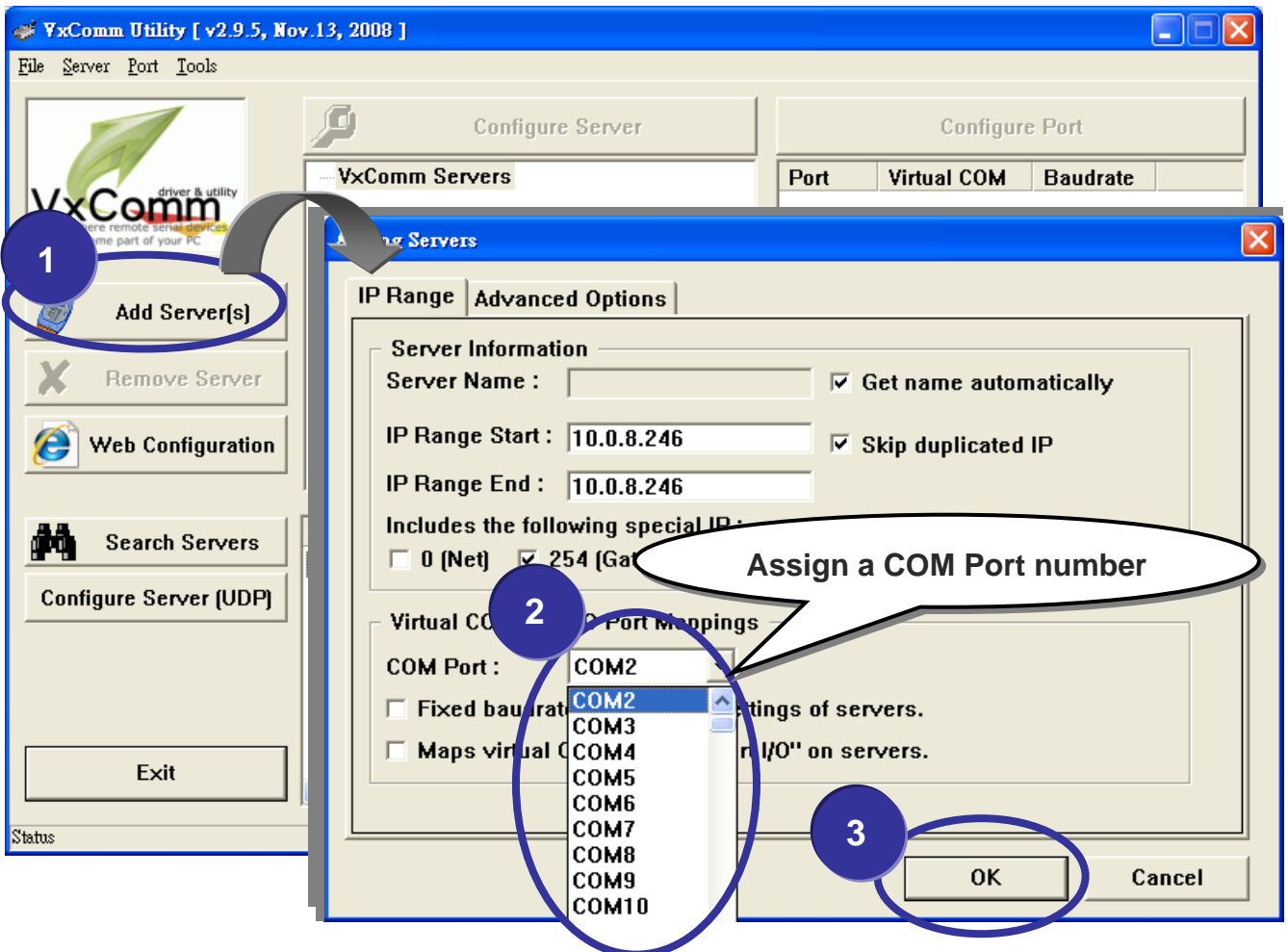


Step 3: Configuring Virtual COM Ports

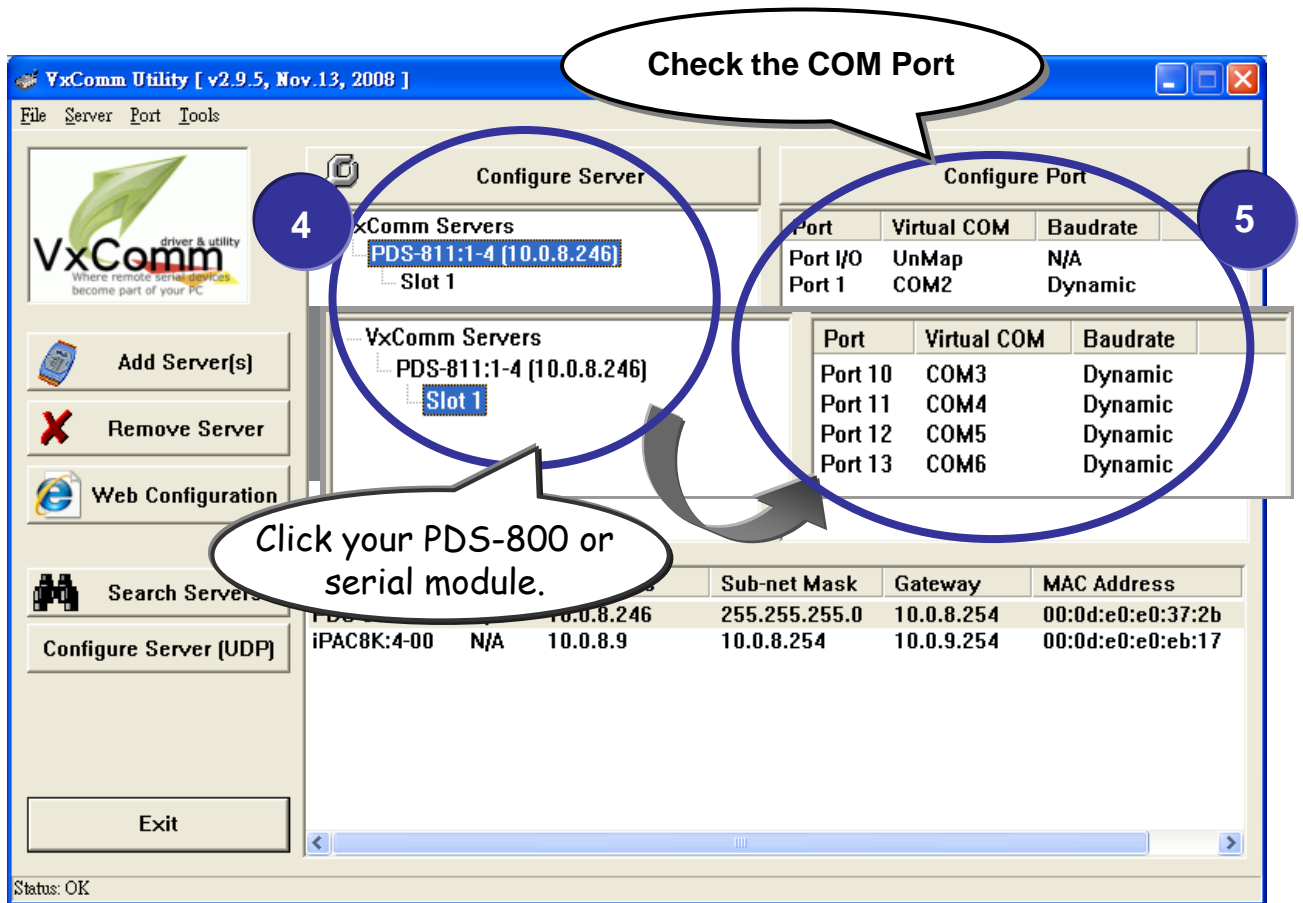
1. Click the **"Search Servers"** button again to search PDS-800 for ensuring the new configuration of PDS-800 is working well. Then click your PDS-800 on the list to select it.



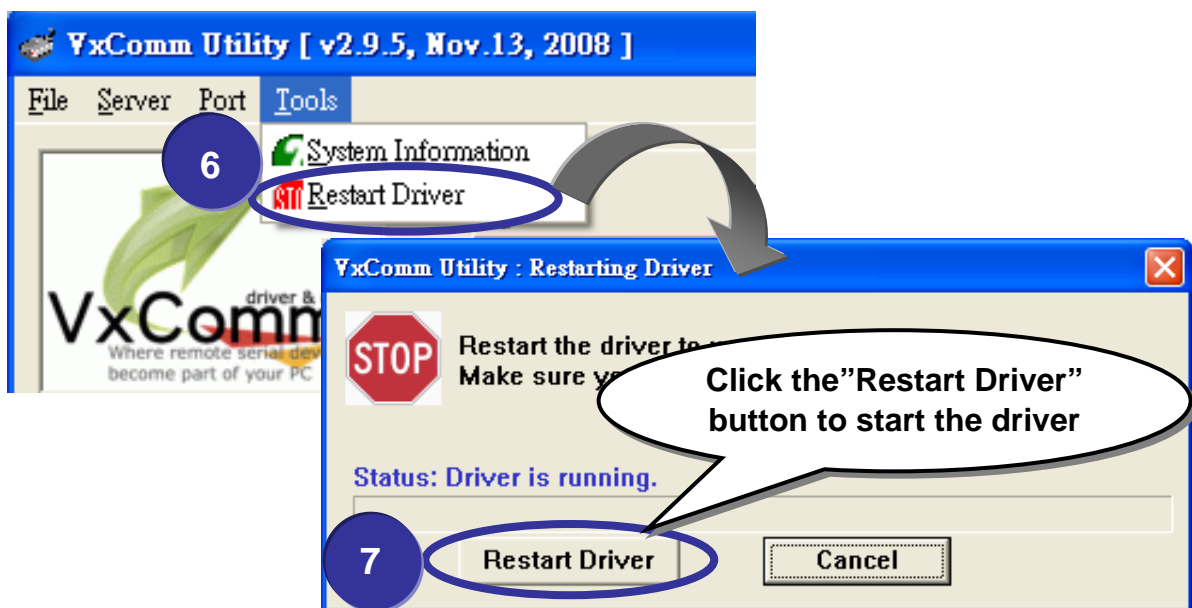
2. Click the **"Add Server(s)"** button. Assign a COM Port number and click "OK" to save your settings.



- Click on PDS-800 name or slot that your module plugged in , and then check the Virtual COM port numbers on the PC.

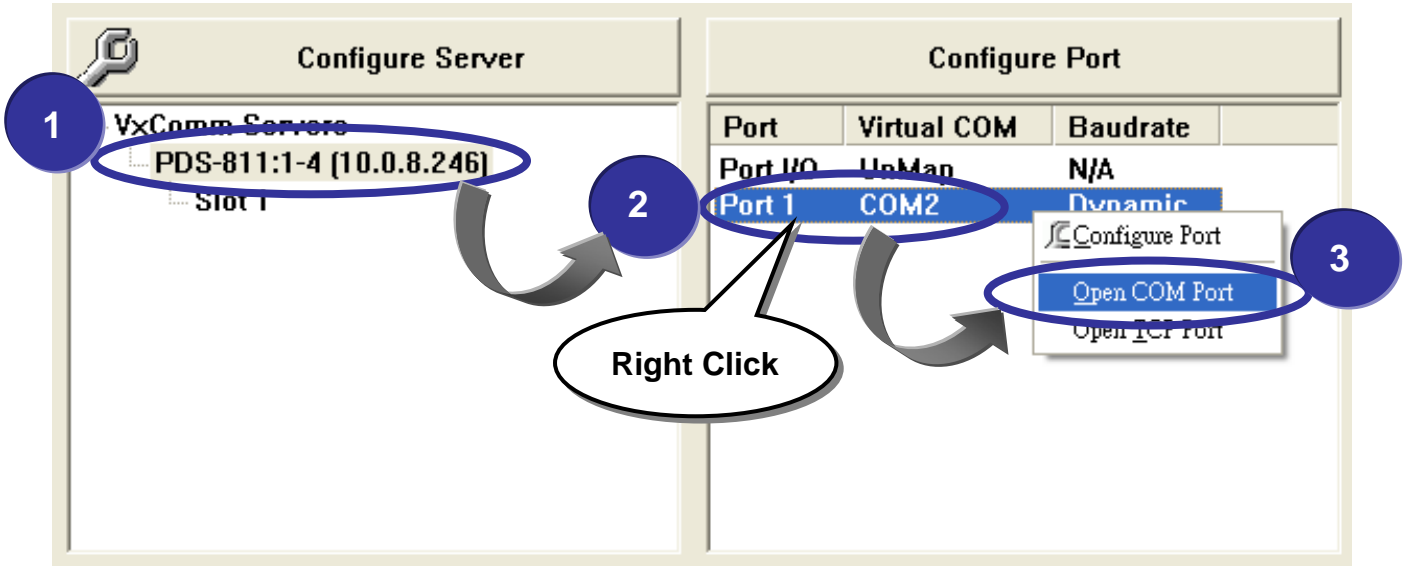


- Click “Tools” >> “Restart Driver”, and then click the “Restart Driver” button to start the driver.

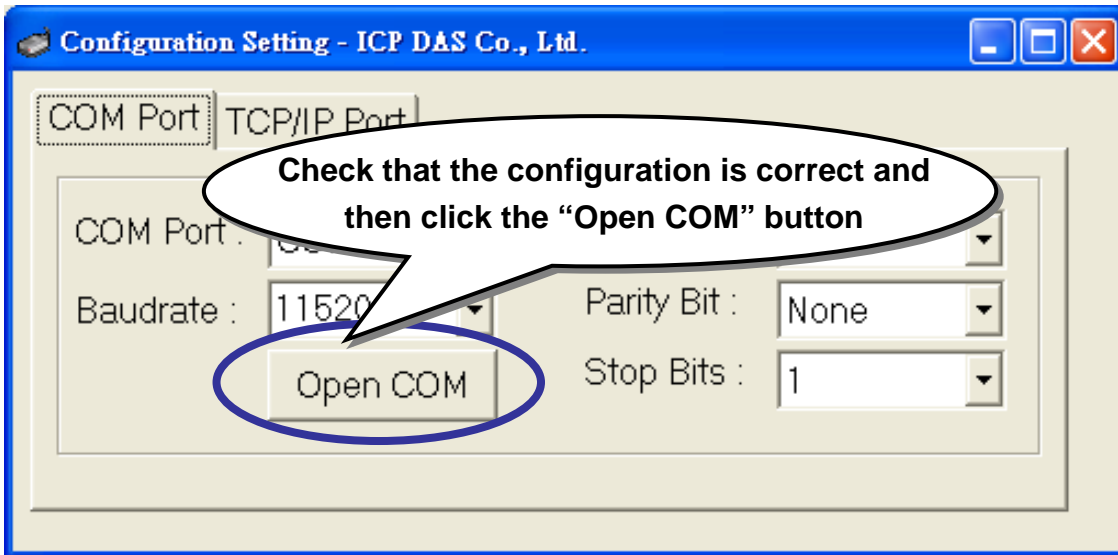


Step4: Testing your PDS-800

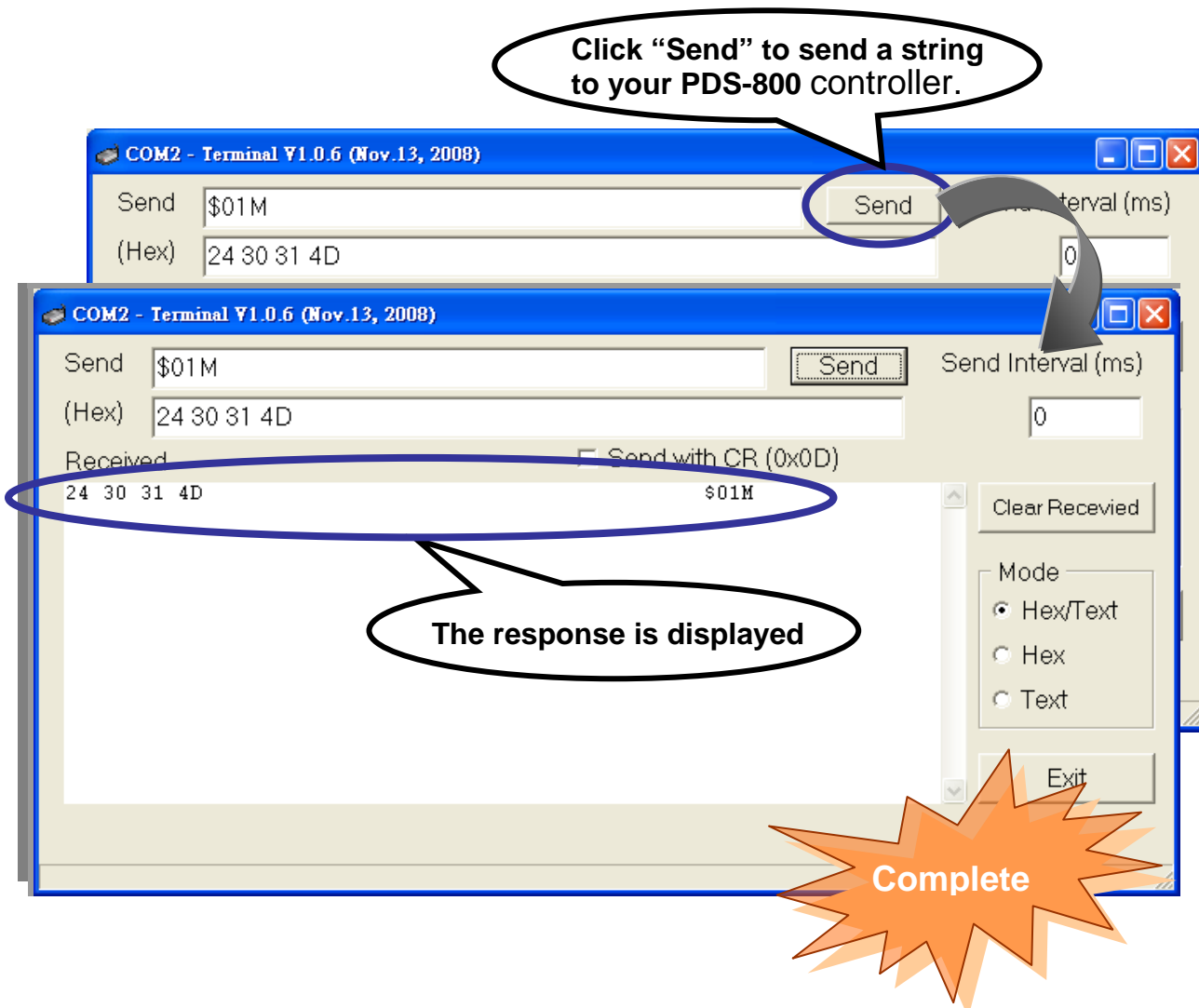
1. Connect the “RXD” and the “TXD” of the PDS-800 controller, as shown in the diagram in **Step1**.
2. Right click Port 1 and then choose the “**Open COM Port**” option.



3. Check that the configuration of the COM Port is correct and then click the “**Open COM**” button.



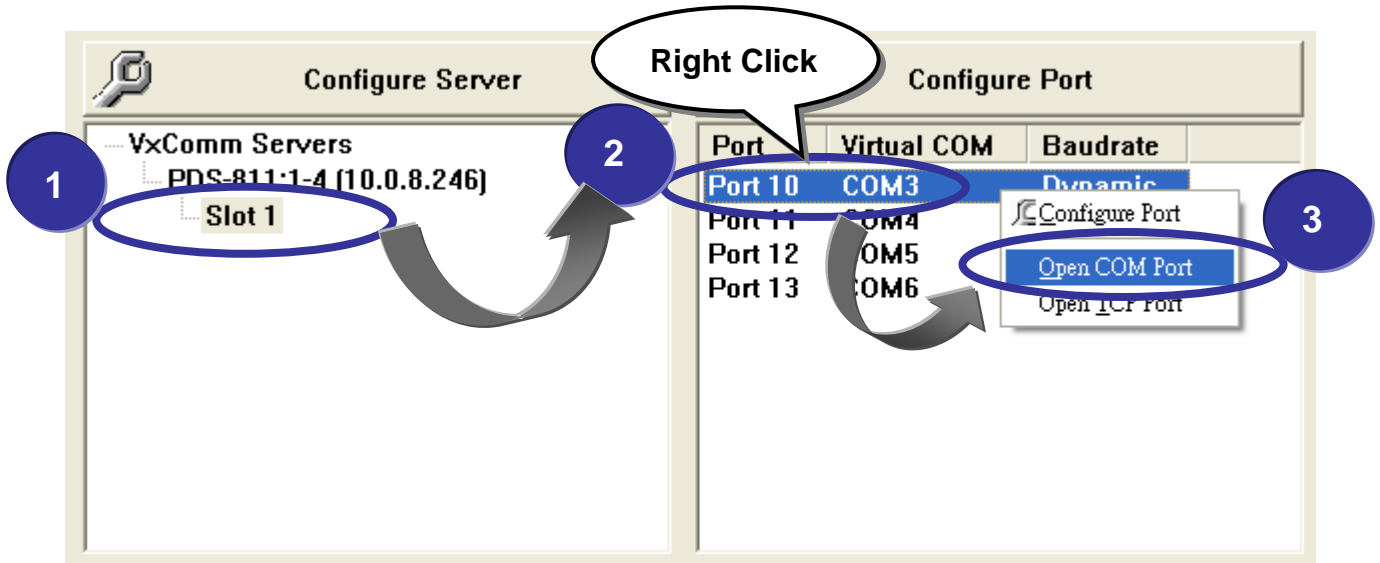
4. Type a string in the send field then click the “send” button. If a response is received, it will be displayed in the received field.



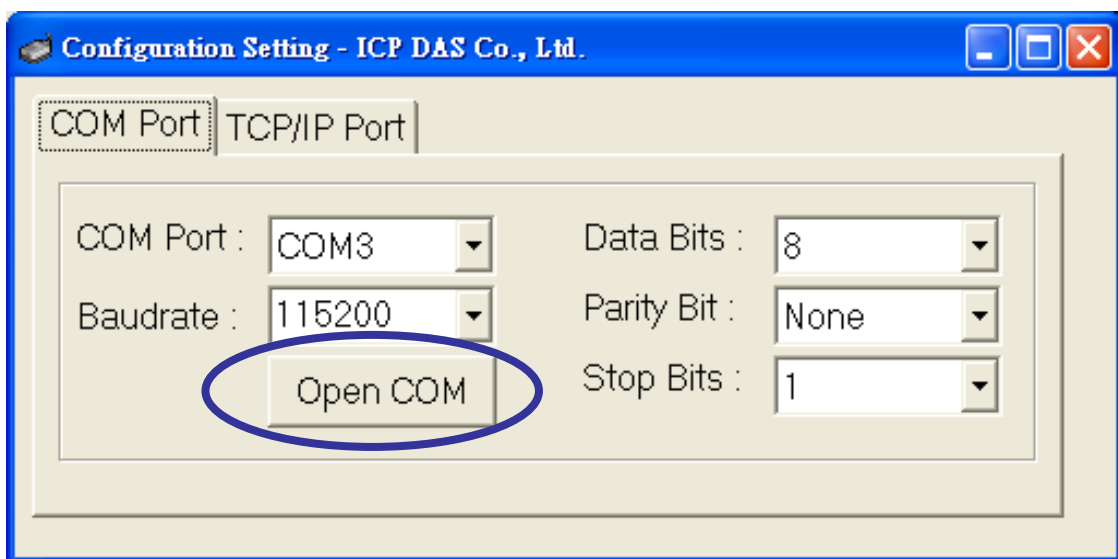
5. If the test is successful, other COM port programs will be able to work directly by using the Virtual COM Port.

Step5: Testing your serial modules (Optional)

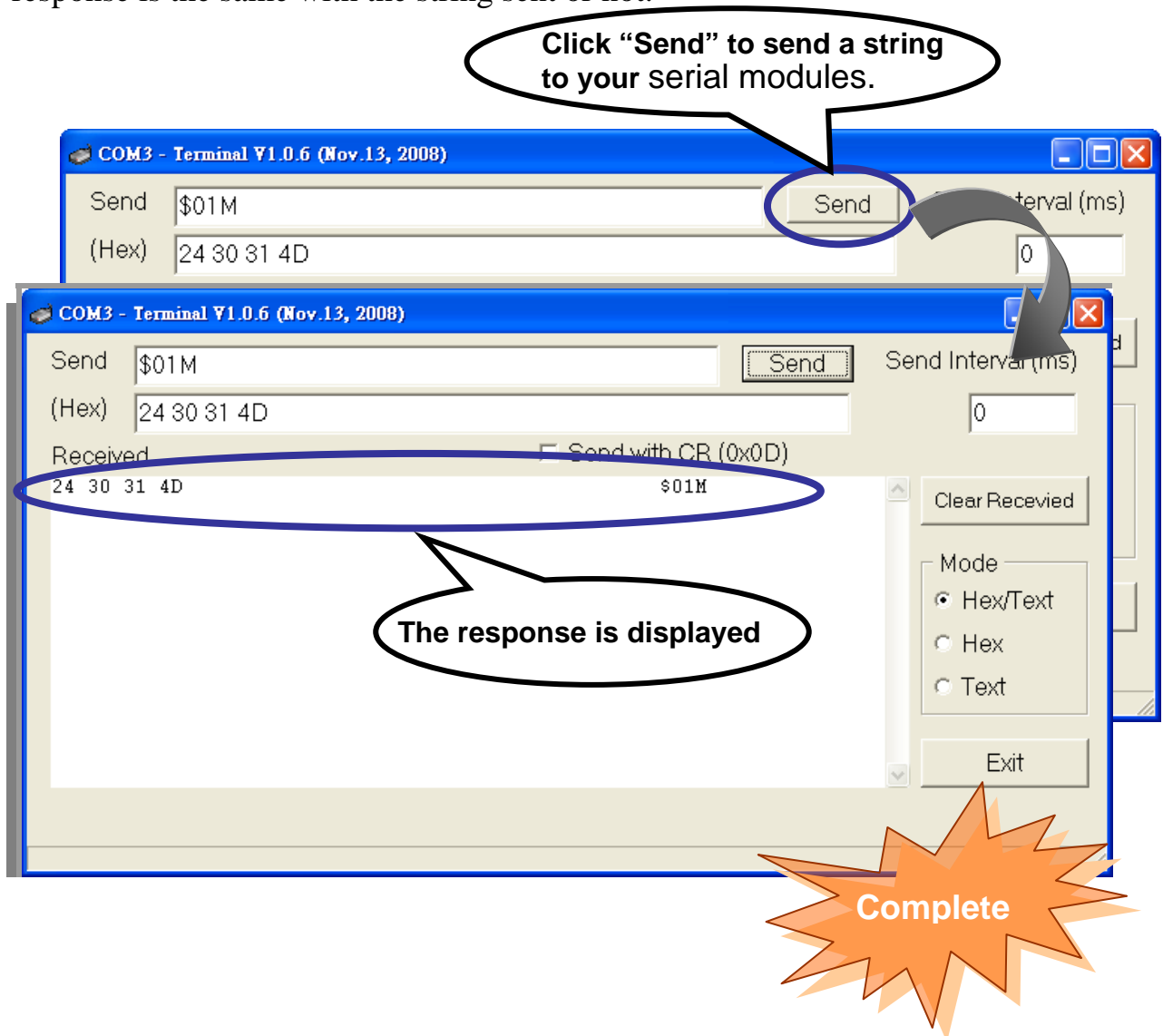
1. Connect the “RXD1” and the “TXD1” of the serial modules (optional), as shown in the diagram in **Step1**. (For example: I-8114iW)
2. Click “**Slot 1**” and right click Port 10 and then choose the “**Open COM Port**” option.



3. Check that the configuration of the COM Port is correct and then click the “**Open COM**” button.



4. Type a string in the send field then click the “send” button. Check whether the response is the same with the string sent or not.



5. If the test is successful, other COM port programs will be able to work directly by using the Virtual COM Port.
6. While using RS-485 modules (*Ex:I-8144iW*), you should wire the “Data1+” with “Data 2+” signals, and wire the “Data1-” with “Data2-” signals for self-test. Then open the first two COM ports, send data to one and receive data from the other.

6 Configuration with Web Browser

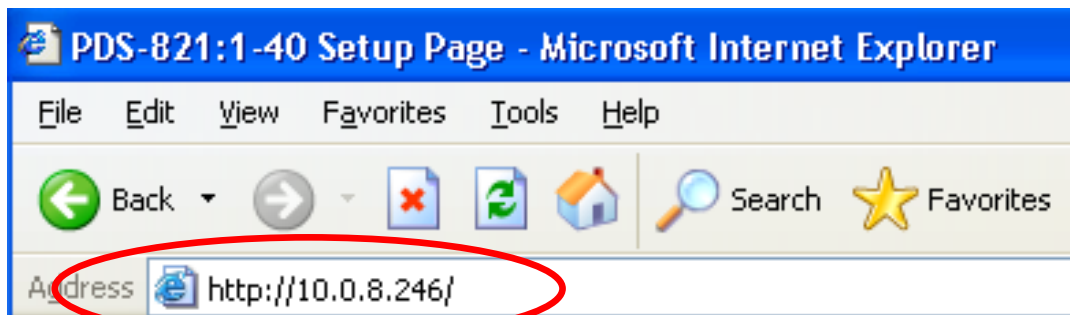
Once the PDS-800 controller has been correctly configured and is networking normally, the configuration details can be retrieved or amended using either the VxComm Utility or a standard web browser, such as IE, FireFox, or Mozilla, etc.

6.1 Connecting to the PDS-800 controller

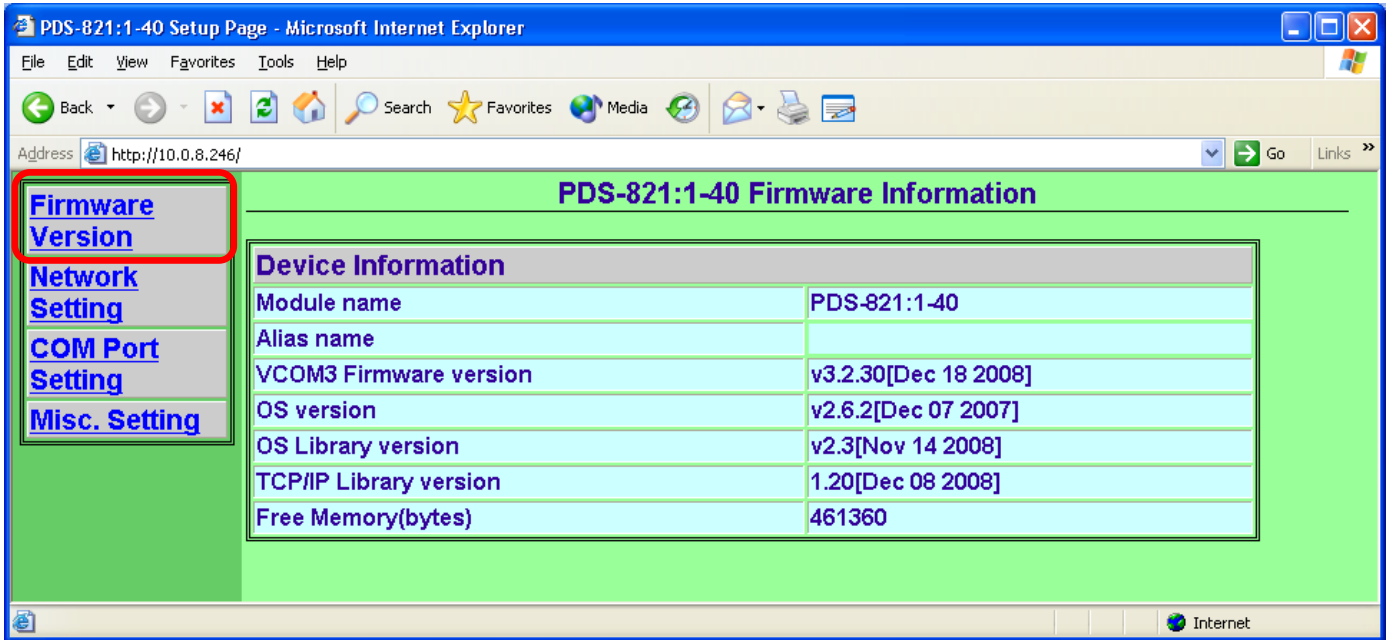


If the COM port program is operating with a PDS-800, changing the configuration will cause a program error.

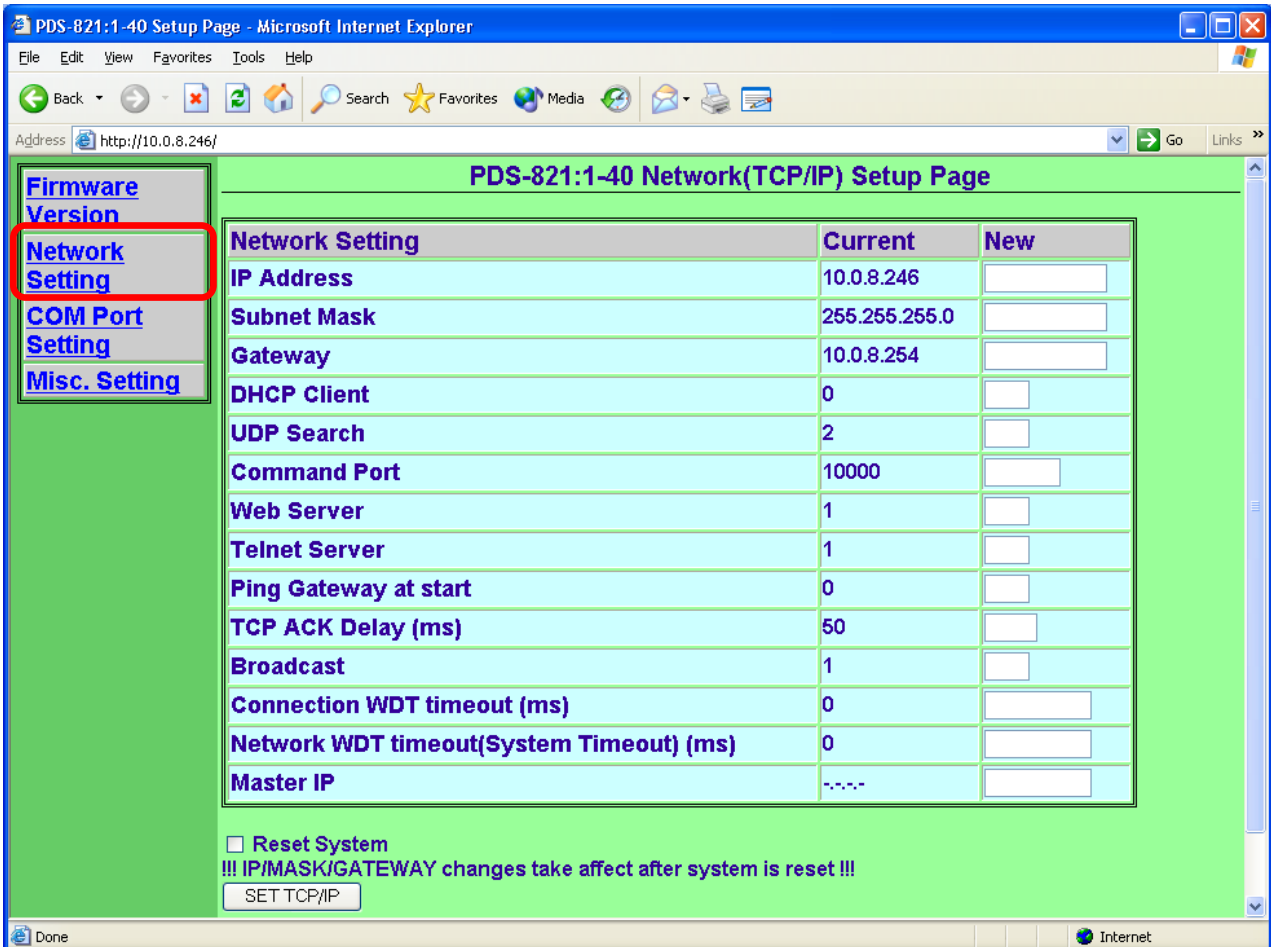
Enter the IP address of the PDS-800 controller in the Address field and press “Enter” to connect to the PDS-800 controller.



The browser shows firmware information of PDS-800 when connected to it.



6.2 Network Settings



- ✦ **IP Address**
- ✦ **Subnet Mask**
- ✦ **Gateway**

The above three items are the most important network settings and should always correspond to the LAN definition. If they do not match, the PDS-800 controller will not operate correctly. If the settings are changed while the module is operating, any links to Virtual COM Port based applications currently in use will be lost and an error will occur.

- ✦ **DHCP Client** : 0 = disabled, 1 = enabled

It is recommended that the DHCP Client setting is kept as disabled, and using static network settings. This ensures your PDS-800 always using a fixed IP address, and you don't need to configure the virtual COM mappings again and again.

- ✦ **UDP Search**: 0 = disabled, 1 = always enabled.

2 = enable the UDP Search function until another client is connected. (Default = 2)

By keeping the UDP search setting as 2, the PDS loading will be reduced. The VxComm Utility will not be able to search for this module until this module's clients are all disconnected.

- ✦ **Command Port**

The default Command Port is 10000.

- ✦ **Web Server**

- ✦ **Telnet Server**

0 = disabled, 1 = enabled

- ✦ **Ping Gateway at start**: 0 = disabled, 1 = enabled.

If the setting is 1 (enabled), the PDS-800 controller will send a ping packet to the gateway during the power-on stage. It is used to inform the gateway that a PDS-800 (itself) has joined the network.

- ✦ **TCP ACK Delay (ms)**, default = 50.

PDS-800 does not want to send an empty ACK followed by a TCP data packet 1 ms later, every time. So it delays a little (TCP ACK Delay), and then can combine the ACK and data packet into one. This efficiency reduces the number of packets and reduces network loadings.

☎ **Broadcast**

1 = receive UDP broadcast packets

0 = reject UDP broadcast packets

☎ **Connection WDT timeout (ms):** default = 0 (disabled), min. = 10000.

If the PDS-800 controller does not receive any data from a client PC within the period of the “Connection WDT timeout”, the module will close the connection to the client.

☎ **Network WDT timeout (ms):** 0 = disabled, min. = 30000.

If the PDS-800 controller does not receive any data from any of the clients within the period of the “Network WDT timeout”, the module will reboot itself. The default setting should be 300000 ms (= 300 seconds).

This setting is the same as “SystemTimeout” setting (unit: ms) on Console/Telnet command, and is the same as “/STxxx” in command line parameter (unit: seconds).

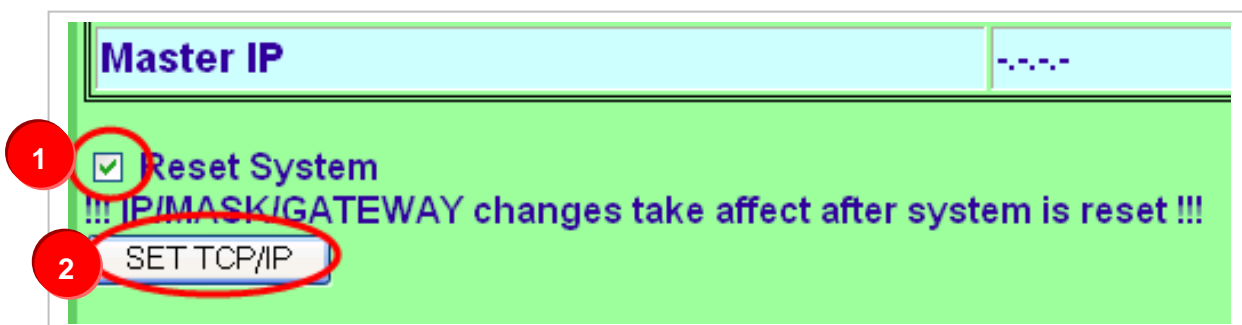
When user uses “config=RESET” Console/Telnet command to clear the EEPROM, the “Network WDT timeout” (SystemTimeout, /ST) setting will also be cleared to 0.

Users have to configure this setting again by “SystemTimeout” Console/Telnet command.

☎ **Master IP:** default = empty (disabled).

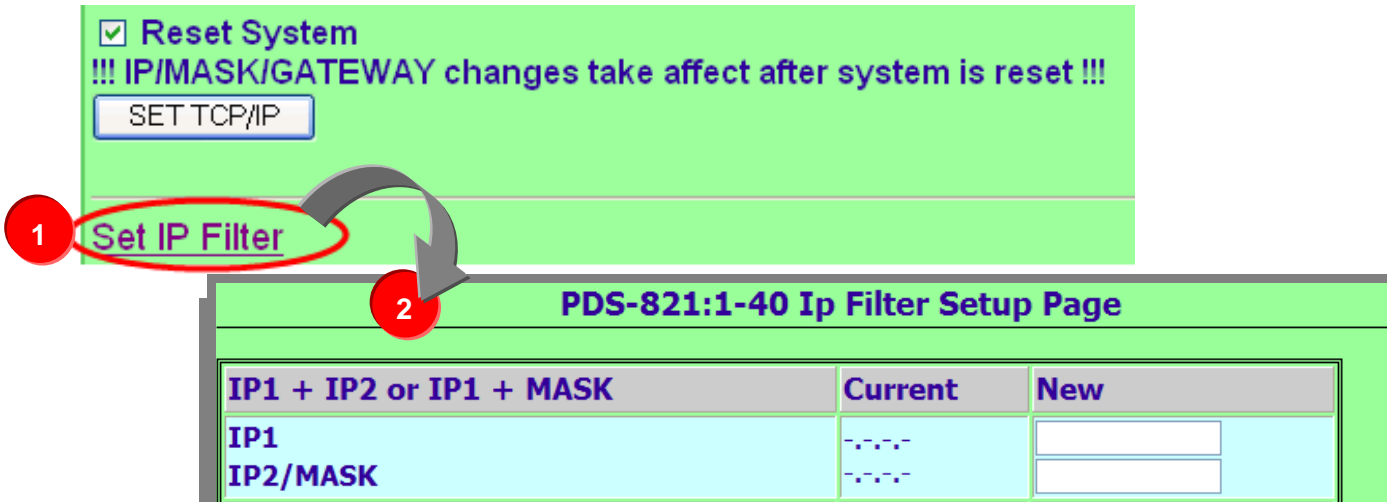
If the Master IP is set, only the client using Master IP can change the COM Port configuration. It is to prevent the COM Port configuration changed by other clients.

After setting the new configuration, click the “**Set TCP/IP**” button to save the new settings to the PDS-800 controller. If the “Reset System” option is checked, the PDS-800 controller will reboot itself after the saving operation is complete, otherwise the original settings will still be valid until the next power-on.



6.3 IP filter setting

The IP filter setting limits which client PCs are able to link to the PDS-800 controller via specific IP addresses. When one or more IP addresses are set in the filter table, only client PCs where the IP address is included in the range listed of the filter table will be able to connect to the PDS-800 controller. Any requests from other PCs will be rejected.



- ⊕ **Set IP1 only:** only clients who's IP address is included in the filter table are able to connect to the PDS-800 controller.
- ⊕ **Set IP1 + IP2:** set a range of IP address as a starting and ending point. The setting allows clients who's IP address is included in the range are able to connect to the PDS-800 controller.
- ⊕ **Set IP1+Mask:** set the IP filter range as:
 $(IP1 \& Mask) + 0 \sim (IP1 \& Mask) + (\sim Mask)$.

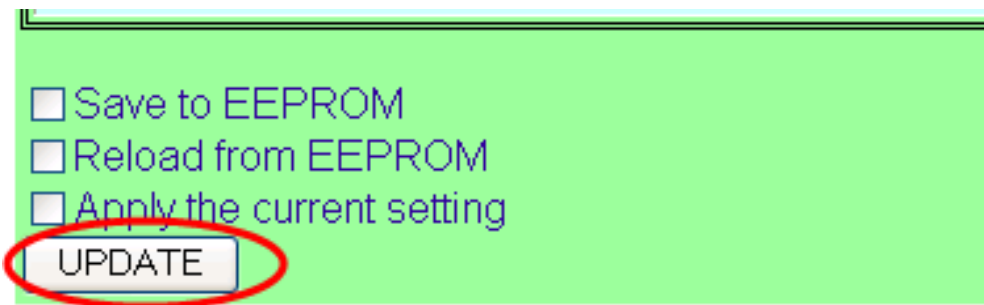
Only clients who's IP address is included in the range are able to connect to the PDS-800 controller. For instance:

IP1 = 10.0.9.5, mask = 255.255.255.0

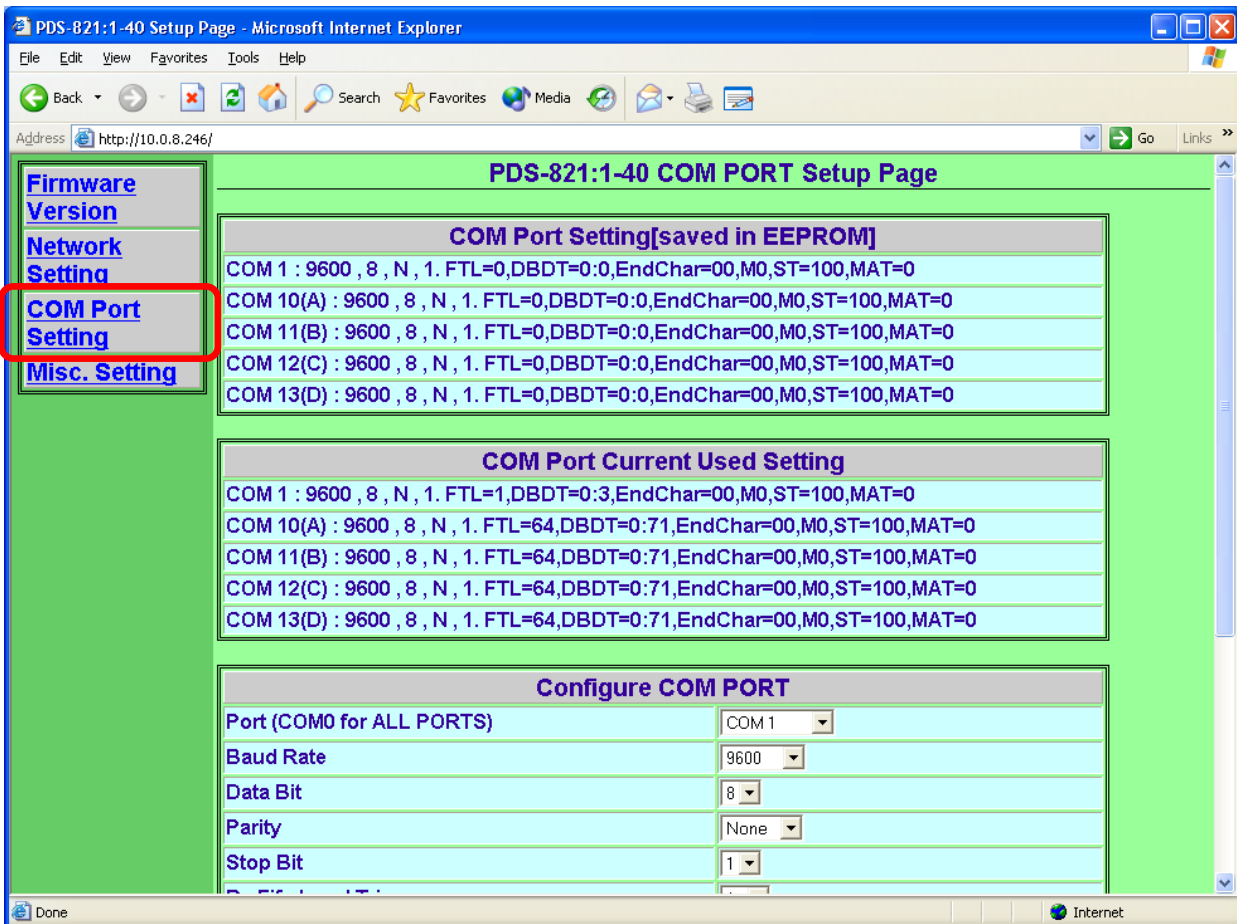
IP1 & MASK = 10.0.9.0, ~mask = 0.0.0.255

This allows clients who's IP address is included in the range of 10.0.9.0 ~ 10.0.9.255 are able to connect to the PDS-800 controller.

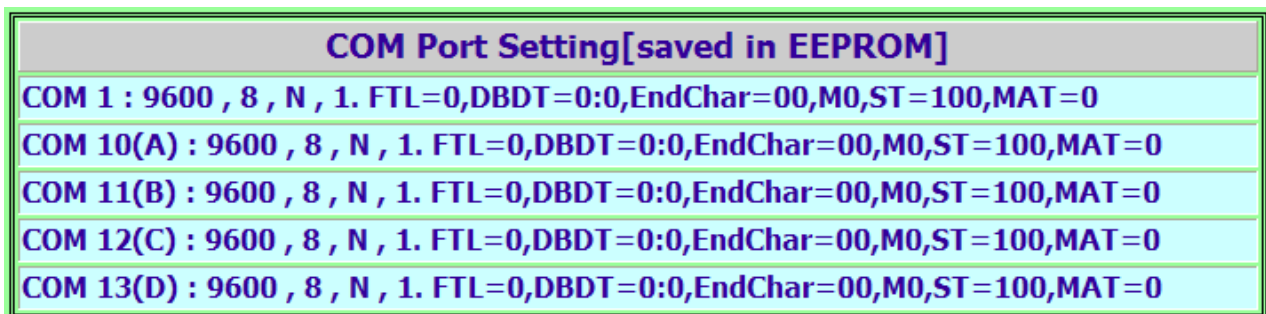
- Click the **“Update”** button to validate the settings.



6.4 COM Port Settings



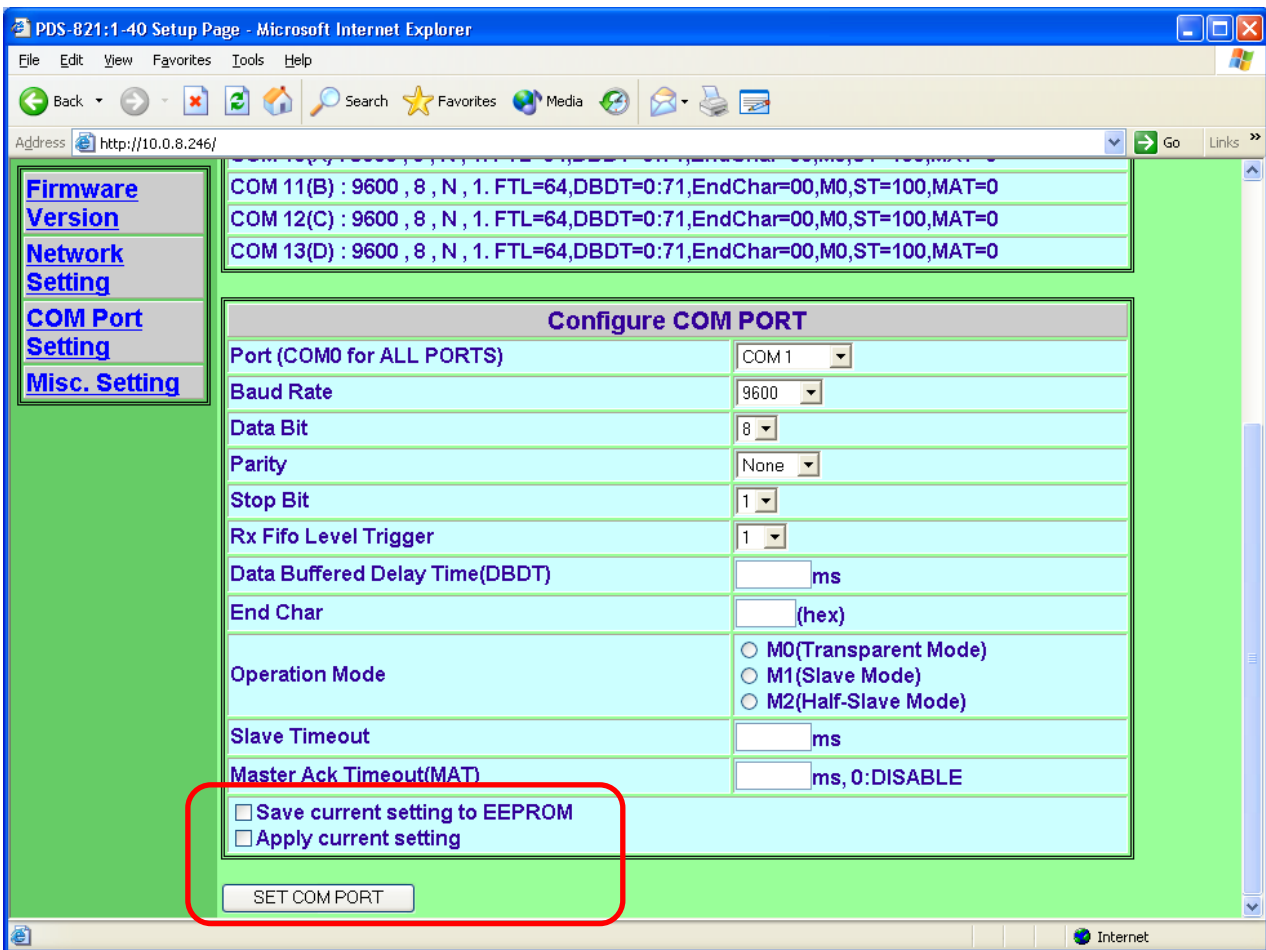
- The COM Port Settings list is saved in the EEPROM on the PDS-800 controller.



✚ The Currently Used COM Port Settings list

COM Port Current Used Setting
COM 1 : 115200 , 8 , N , 1. FTL=1,DBDT=0:2,EndChar=00,M0,ST=100,MAT=0
COM 10(A) : 115200 , 8 , N , 1. FTL=64,DBDT=0:7,EndChar=00,M0,ST=100,MAT=0
COM 11(B) : 9600 , 8 , N , 1. FTL=64,DBDT=0:71,EndChar=00,M0,ST=100,MAT=0
COM 12(C) : 9600 , 8 , N , 1. FTL=64,DBDT=0:71,EndChar=00,M0,ST=100,MAT=0
COM 13(D) : 9600 , 8 , N , 1. FTL=64,DBDT=0:71,EndChar=00,M0,ST=100,MAT=0

✚ The COM Port Settings area



Note: If the “**Set COM Port**” button is clicked without checking “**Save current setting to EEPROM**” and “**Apply current setting**”, option the new settings will be saved to the PDS-800 only and the new settings will be valid after the next power-on.

If the “**Save current setting to EEPROM**” and “**Apply current setting**” checked when the “**Set COM Port**” button is clicked, the new settings will be valid immediately.

⌘ **Port (COM0 for ALL PORTS):** The COM Port number on the PDS-800 controller.

⌘ **Baud Rate, Data Bits, Parity**

⌘ **Stops Bits, End Char:**

The configuration settings should match the serial device used.

⌘ **Rx Fifo Level Trigger:**

This option is used to set the number of characters that the COM Port can receive at once time, the PDS will move the data from the COM Port FIFO to the PDS. If the amount of data transferred is large and uses a transfer speed (115200 bps), setting a smaller value is helpful in preventing data loss.

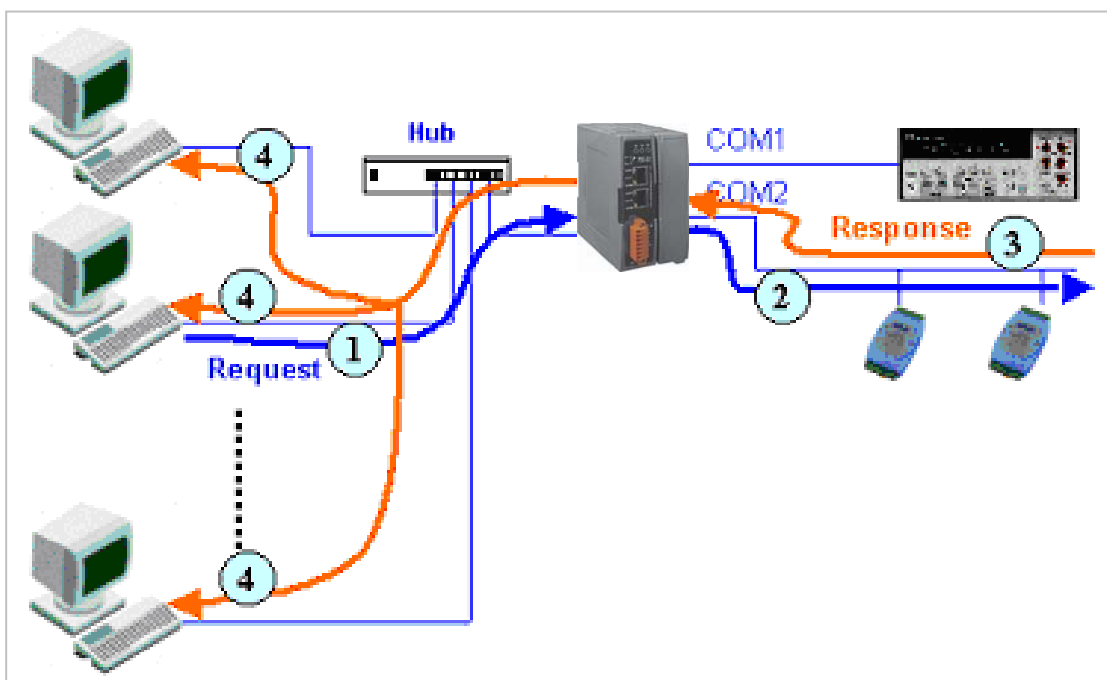
⌘ **Data Buffered Delay Time (DBDT) :**

When the COM port does not receive data from devices connected over the period of DBDT setting, the PDS-800 will determine that the data transfer is over and return to process next tasks.

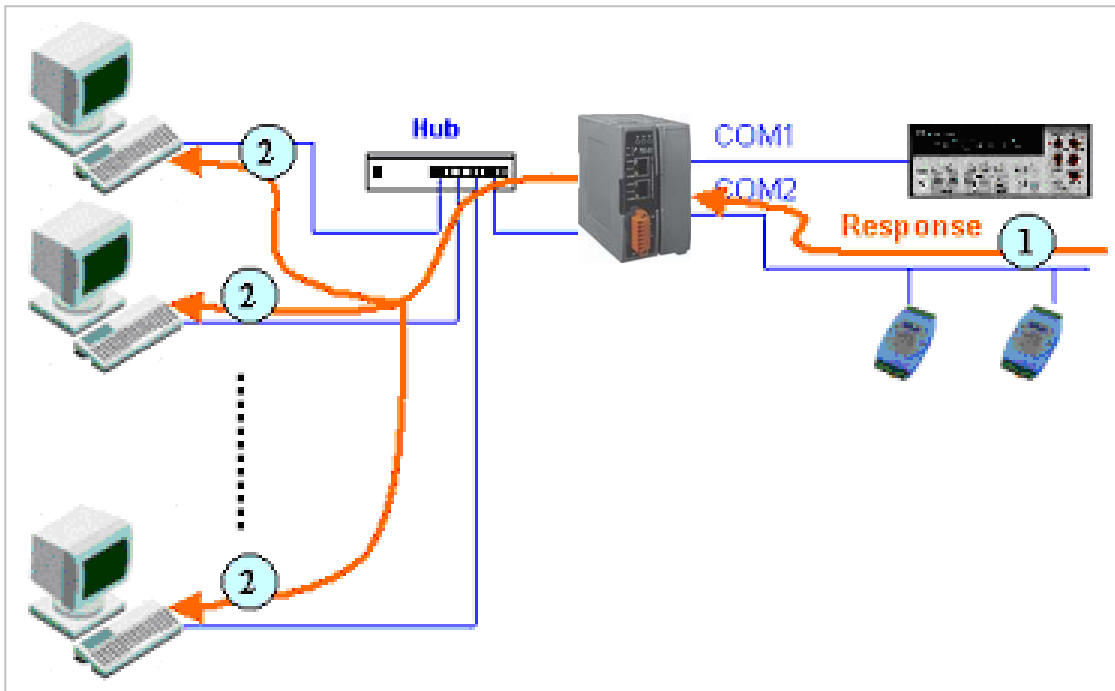
⌘ **Operation Mode: M0, M1, M2**

M0: Transparent Mode

Condition 1: One client sends a request to the PDS-800 controller to access each device.
The PDS-800 controller echoes the data from each device to each connected client.

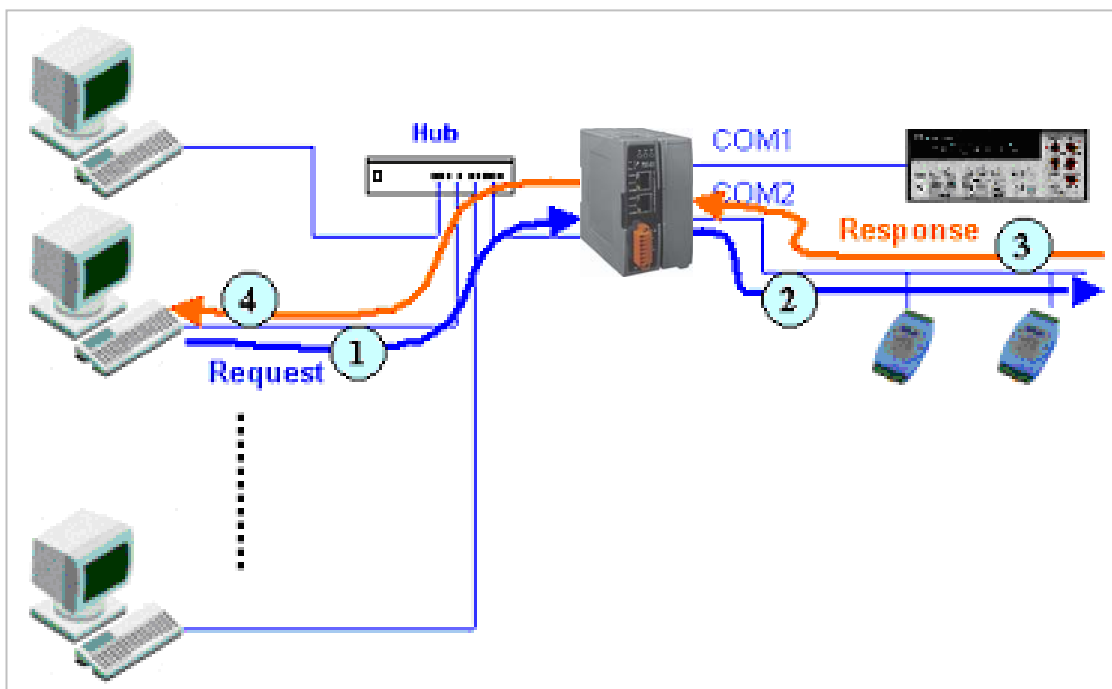


Condition 2: No clients send any requests to the PDS-800 controller. The PDS-800 controller echoes data from the devices to each connected client.

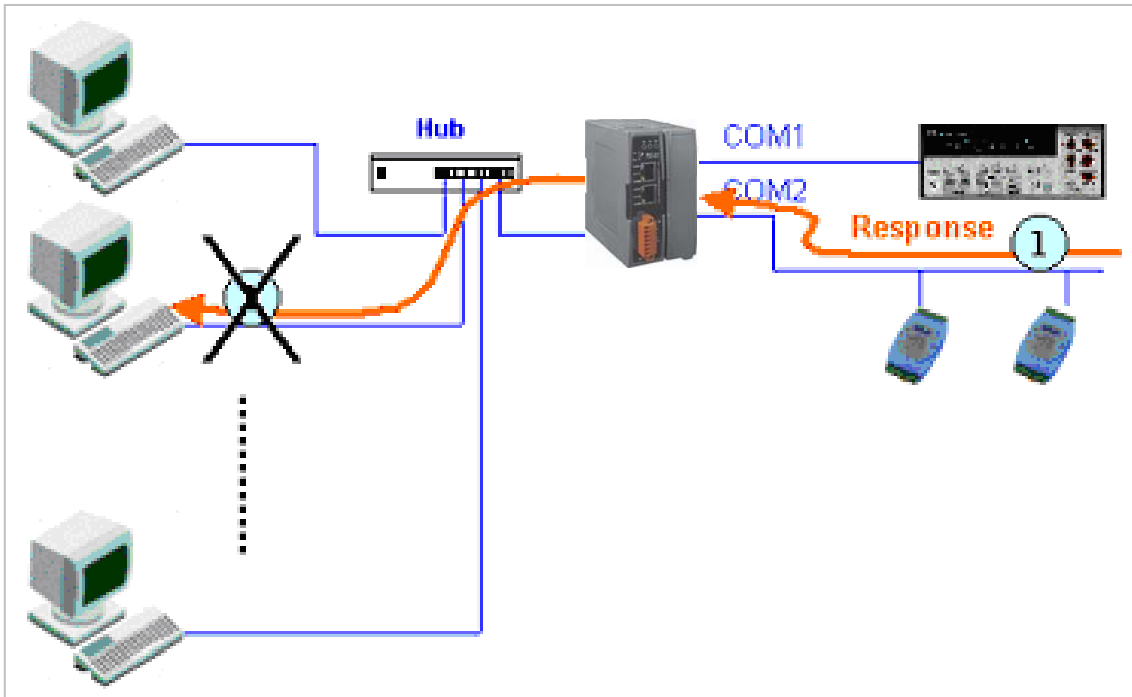


M1: Slave Mode (Single-echo mode)

Condition 1: One client sends a request to the PDS-800 controller to access the other devices. The PDS-800 controller echoes data from the devices to the client that requested the service.



Condition 2: No clients send any requests to the PDS-800 controller. The PDS-800 controller doesn't echo any data from the devices to any client.



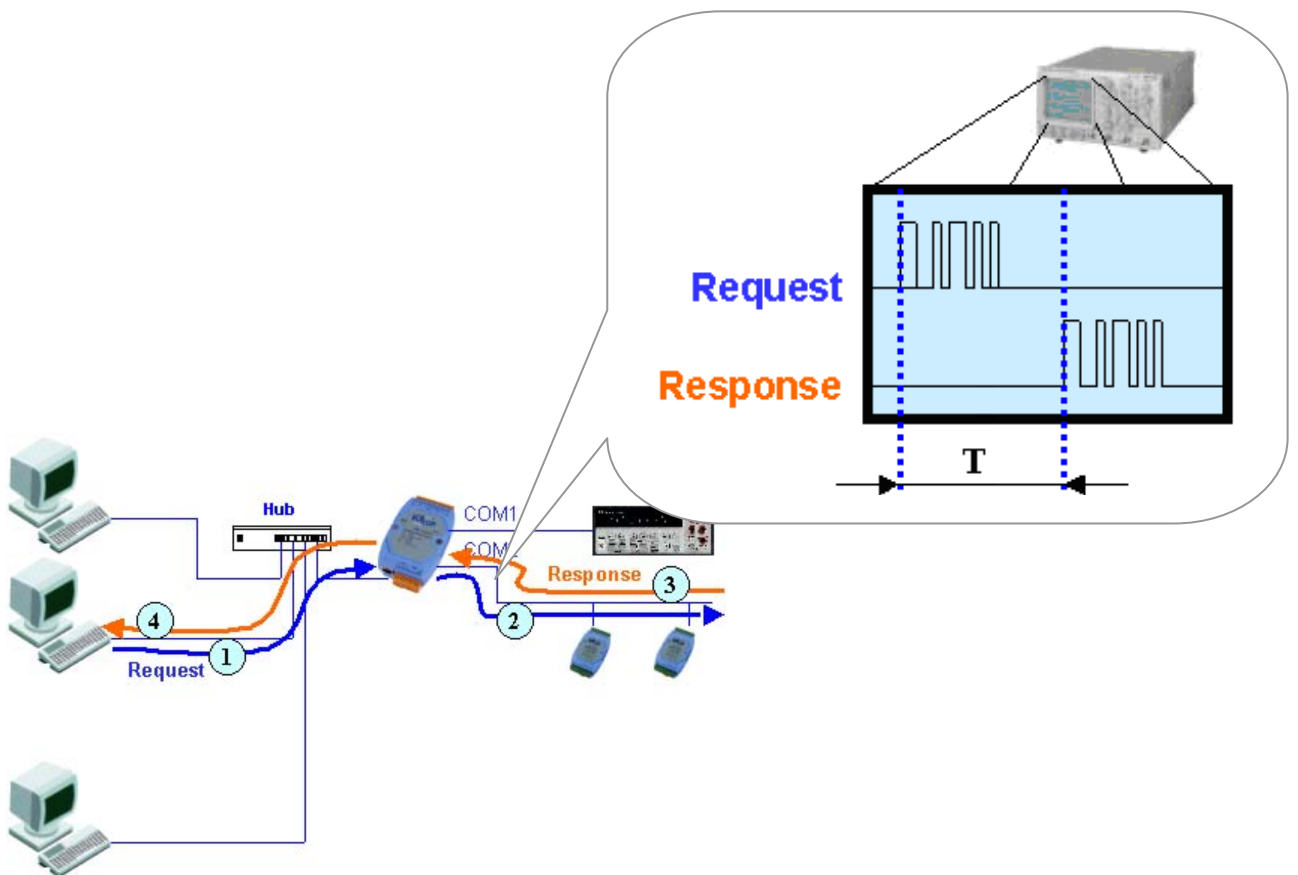
M2: Half-Slave Mode (Is situated between M0 and the M1 mode)

In M2 mode, if only then single connection is the same with the M0 mode, this connection has the right of use, even if will come from COM Port of message also to give the TCP port on own initiative. (Refer to P53. **M0 Mode** declaration)

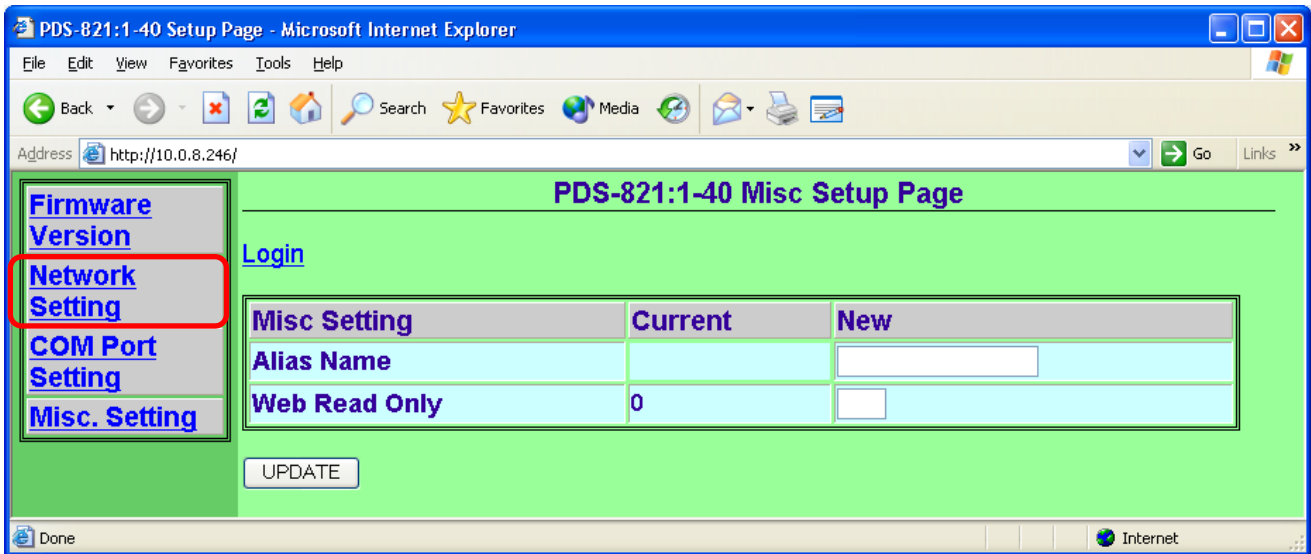
When has multiple connections is equally operates with the M1 mode, uses COM Port by way in turn, but COM Port right of use will retain the TCP connection which will issue a command for last. (Refer to P54. **M1 Mode** declaration)

In M1, the slave mode timeout setting is use to set the waiting time after last character of the request sent to the device. If the device does not respond within the timeout value, the PDS-800 controller will return a timeout error and process next request.

Operation Mode	<input type="radio"/> M0(Transparent Mode) <input checked="" type="radio"/> M1(Slave Mode) <input type="radio"/> M2(Half-Slave Mode)
Slave Timeout	<input type="text"/> ms



6.5 Miscellaneous settings

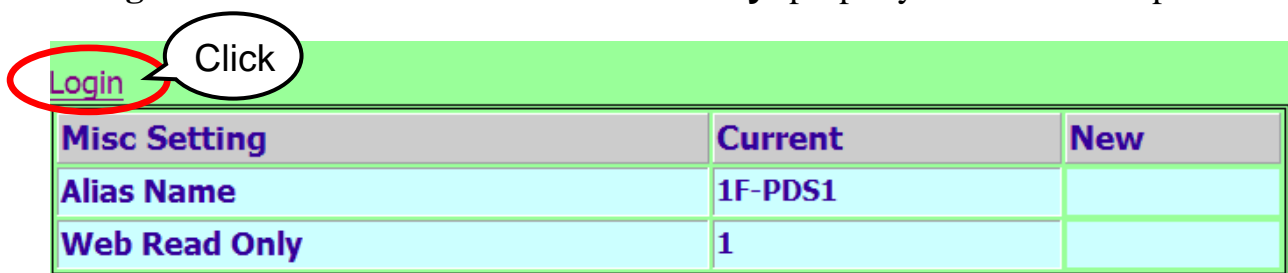


- ⊕ **Alias Name:** allocates an alias to the PDS-800 controller.
- ⊕ **Web Read Only:** 0 = disabled, 1 = enabled

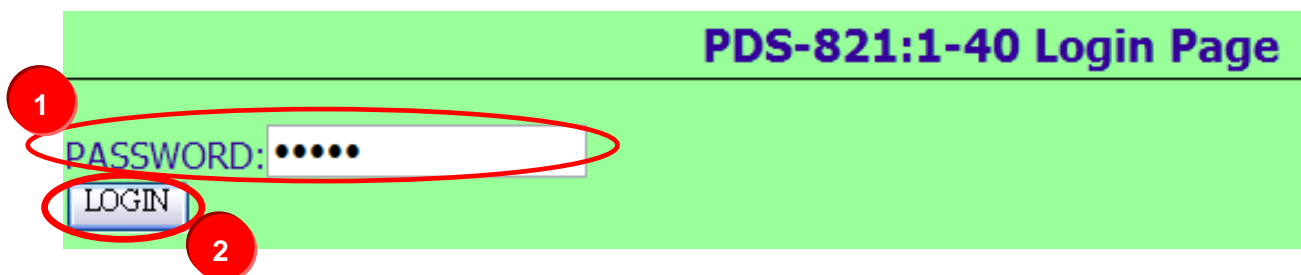


*If the “Web Read Only” property is set to 1, enabled, the web server will **not** be able to save any new configurations to the PDS-800 module. To disable the “Web Read Only” property, refer to the information below.*

- ⊕ **Login:** used to disable the “Web Read Only” property or to set a new password.



1. Enter the password (default is **admin**) and click the “**LOGIN**” button to proceed to the settings page.



3. Set the new “**Web Read Only**” properly = 0 and click the “**UPDATE**” button.

PDS-821:1-40 Misc Setup Page

[Logout](#)

Misc Setting	Current	New
Alias Name	1F-PDS1	<input type="text"/>
Web Read Only	1	<input type="text" value="0"/>
Set New Password		<input type="text"/>
Confirm New Password		<input type="text"/>

4. Check that the current the “**Web Read Only**” = 0 and then click “**Logout**” to complete the operation.

PDS-821:1-40 Misc Setup Page

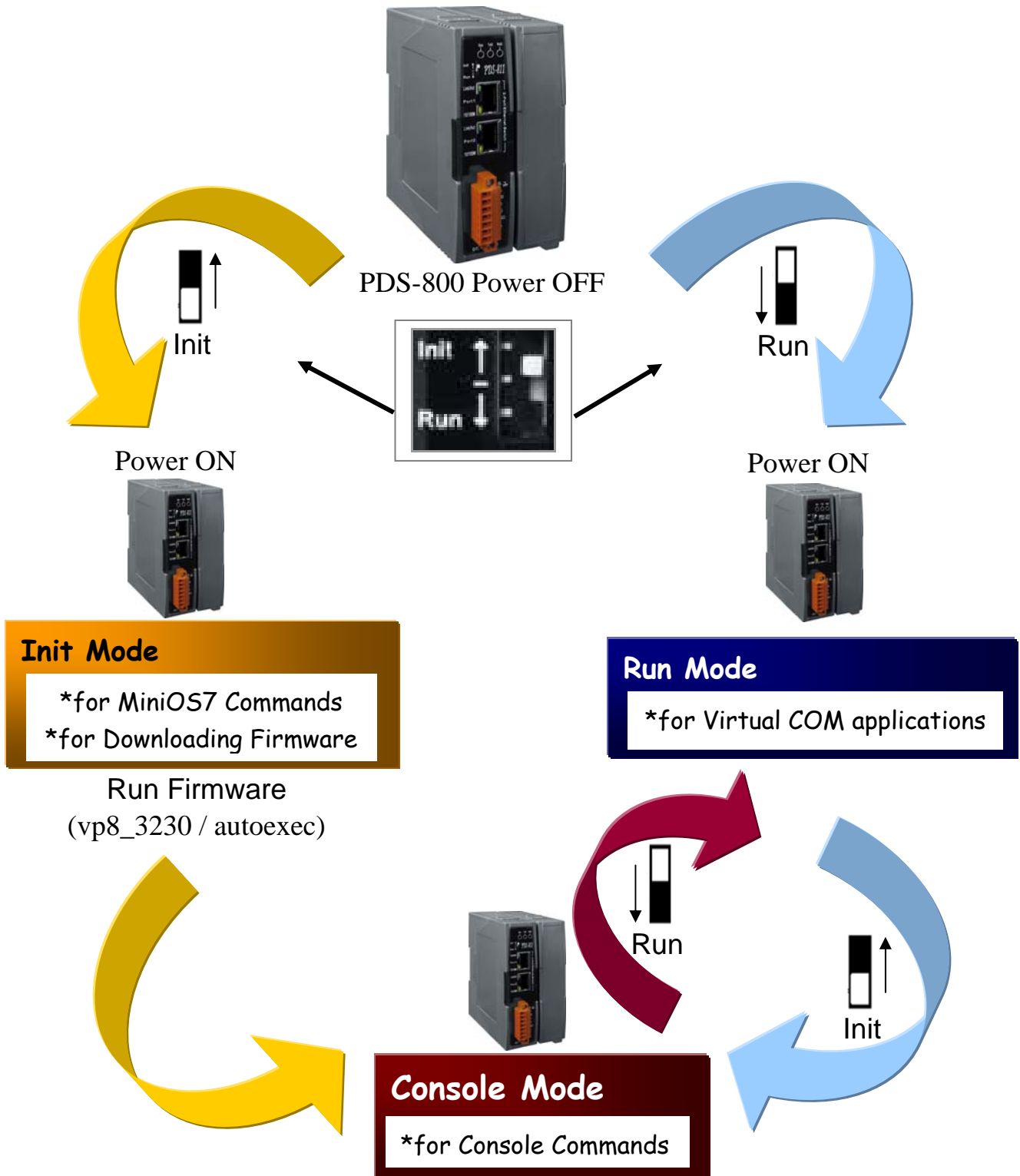
[Logout](#)

Misc Setting	Current	New
Alias Name	1F-PDS1	<input type="text"/>
Web Read Only	0	<input type="text"/>
Set New Password		<input type="text"/>
Confirm New Password		<input type="text"/>

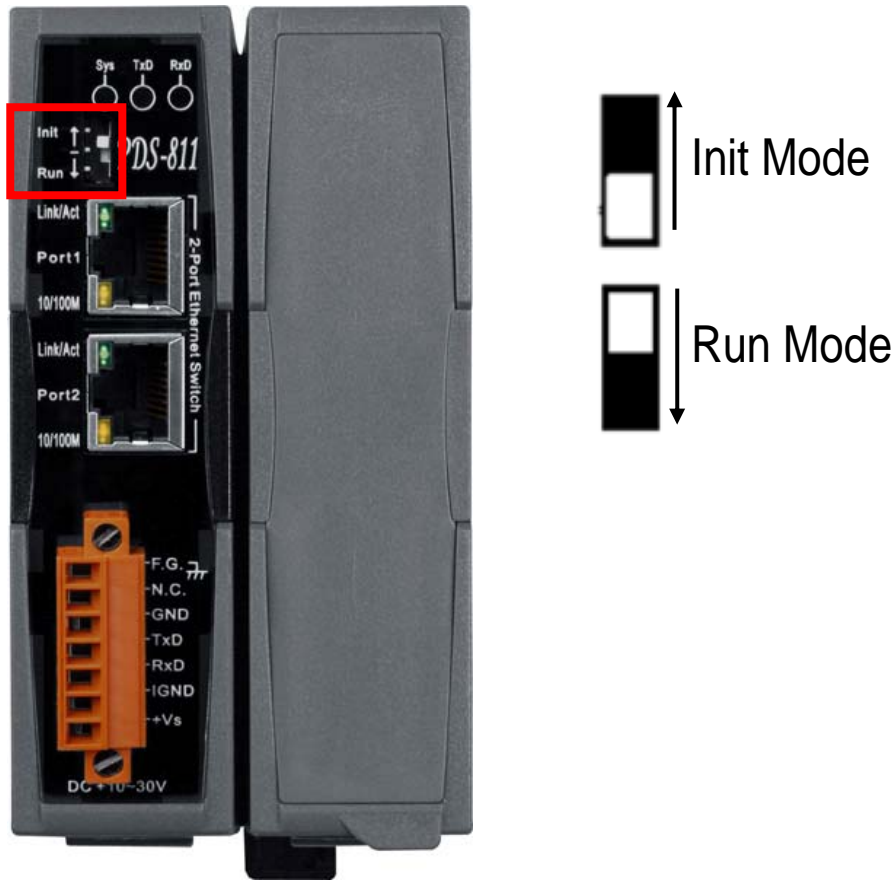
4. User can restore PDS-800 password to default value “admin” by using “**config=RESET**” console command (refer to section Console/Telnet Commands List). This command sets most configurations of PDS-800 to factory setting. It requires rebooting the PDS-800 for loading new configuration (includes default password).

7 Console / Telnet Commands List

7.1 Operation Flowchart



7.2 Init/Run Mode Setting



7.3 Comparison Table (Init/Run/Console Modes)

Mode	Firmware	VCOM Commands	Telnet Commands	Console Commands
Init	<i>Stop</i>	<i>No</i>	<i>No</i>	<i>No</i>
	Init Mode is used to upgrade firmware and accepts MiniOS7 commands (from PDS.COM1) only.			
Run	<i>Running</i>	<i>Yes</i>	<i>Yes</i>	<i>No</i>
	Run Mode is used for Virtual COM applications, and accepts Virtual COM commands (TCP port 10000) and Telnet commands (TCP port 23).			
Console	<i>Running</i>	<i>Yes</i>	<i>Yes</i>	<i>Yes</i>
	Console Mode is used to configure the Virtual COM. PDS.COM1 is the console port that accepts console commands while other ports are still working with Virtual COM applications.			

7.4 Command List

No.	Command	Description
1	<u>IPFILTER</u>	Retrieves/Sets the IP addresses that are allowed to access the PDS.
2	<u>IPCONF</u>	Queries the network configuration. (IP/Mask/Gateway/MAC addresses).
3	<u>SOCKET</u>	Lists all the status of sockets (Listen/Not Used Yet) together with the type of each socket (TCP Server: Port No./UDP/Unused).
4	<u>COM</u>	Queries or sets the configuration of the COM Ports (Baud Rate/Parity /Stop Bits)
5	<u>Broadcast</u>	Queries or sets the Broadcast parameter, which determines whether or not the module can receive Broadcast packets.
6	<u>SystemTimeout</u>	If PDS-800 has no network communications during the SystemTimeout period, the PDS-800 will reboot it-self automatically.
7	<u>SocketTimeout</u>	If there is no data send/receive on the connection during the SocketTimeout period, the PDS-800 will close the connection automatically.
8	<u>M</u>	Gets/Sets the echo mode. /M0: Transparent Mode, Multi-Echo, Data-Shared. /M1: Slave Mode, Single-Echo, None-Shared.
9	<u>EchoCmdNo</u>	Queries or sets the EchoCmdNo parameter that enables or disables adding Command Number before response.
10	<u>EndChar</u>	Sets a character that determines the end of a response string.
11	<u>IP</u>	Queries or sets the IP address.
12	<u>MASK</u>	Queries or sets the subnet Mask value.
13	<u>GATEWAY</u>	Queries or sets the Gateway address.
14	<u>MAC</u>	Queries the MAC address.
15	<u>NAME</u>	Queries the module name.
16	<u>ALIAS</u>	Sets the alias for a PDS-800.
17	<u>DHCP</u>	Enables/Disables the DHCP client.
18	<u>UDP</u>	Sets whether to reply to a UDP search command.
19	<u>VER</u>	Queries the version information
20	<u>SAVE</u>	Determines whether or not backup copies of the "autoexec.bat" and "vcom.ini" files are saved when using the "load" command.
21	<u>LOAD</u>	Loads file to the built-in flash disk on PDS-800. It should be used to update firmware only.
22	<u>CONFIG</u>	Restores the factory default settings.
23	<u>RESET</u>	Reboots the PDS-800 controller .
24	<u>QUIT</u>	Exits the running firmware.

7.4.1 IPFILTER

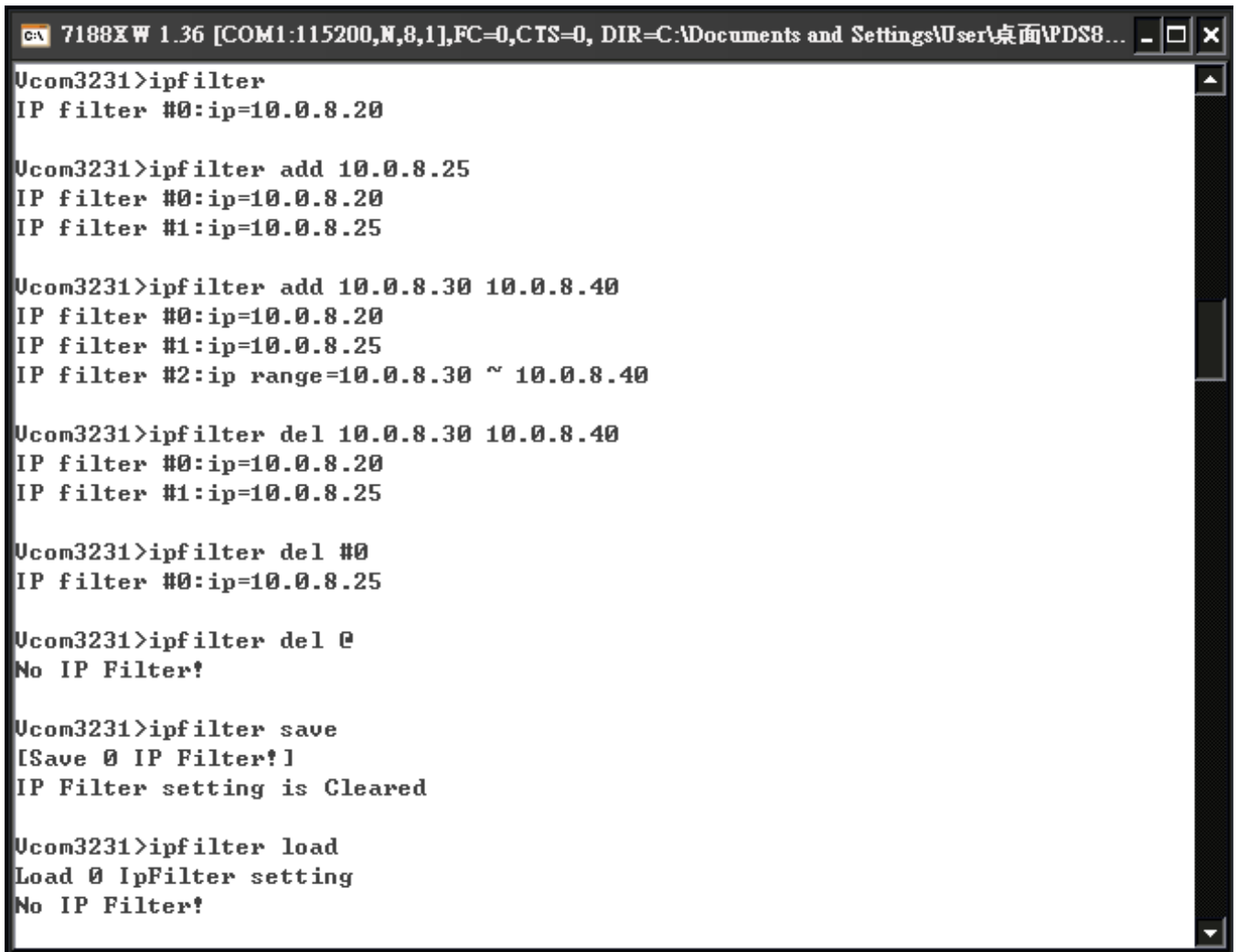
Description: This command is used to query or edit IP filter table. The IP filter table restricts the access of packets based on the IP header. If one or more IP addresses are saved into the IP filter table, only clients whose IP is specified in the IP filter table can access the PDS-800.

Effect: Immediate

Command	Arguments	Description
ipfilter		Queries the IP filter table.
ipfilter	ADD ip1 ADD ip1 ip2	Adds an IP address to the IP filter table. Adds a range of IP addresses (ip1 ~ ip2) to the IP filter table.
ipfilter	DEL ip1 DEL ip1 ip2	Deletes an IP address (ip1) from the IP filter table. Deletes a range of IP addresses (ip1 ~ ip2) from the IP filter table.
ipfilter	DEL #n	Deletes item “n” from the IP filter table.
ipfilter	DEL @	Deletes all items from the IP filter table.
ipfilter	SAVE	Saves the IP filter table to the EEPROM. If the IP filter table is empty, the data in EEPROM will be cleared.
ipfilter	LOAD	Loads the IP filter table from the EEPROM.

- ※ The IP filter table is loaded automatically when the PDS-800 is booted.
- ※ Use the “ipfilter save” command to save a new IP filter table to the EEPROM.

Example:



```
7188XW 1.36 [COM1:115200,N,8,1],FC=0,CTS=0, DIR=C:\Documents and Settings\User\桌面\PDS8...
Ucom3231>ipfilter
IP filter #0:ip=10.0.8.20

Ucom3231>ipfilter add 10.0.8.25
IP filter #0:ip=10.0.8.20
IP filter #1:ip=10.0.8.25

Ucom3231>ipfilter add 10.0.8.30 10.0.8.40
IP filter #0:ip=10.0.8.20
IP filter #1:ip=10.0.8.25
IP filter #2:ip range=10.0.8.30 ~ 10.0.8.40

Ucom3231>ipfilter del 10.0.8.30 10.0.8.40
IP filter #0:ip=10.0.8.20
IP filter #1:ip=10.0.8.25

Ucom3231>ipfilter del #0
IP filter #0:ip=10.0.8.25

Ucom3231>ipfilter del @
No IP Filter!

Ucom3231>ipfilter save
[Save 0 IP Filter!]
IP Filter setting is Cleared

Ucom3231>ipfilter load
Load 0 IpFilter setting
No IP Filter!
```

7.4.2 IPCONF

Description: This command is used to display the network configuration information, such as IP/Mask/Gateway/MAC addresses and the status of DHCP/ACK_Delay/Free Memory/Socket status.

Effect: Immediate

Command	Argument	Description
ipconf		Queries the network configuration.

Example:



```
7188X W 1.36 [COM1:115200,M,8,1],FC=0,CTS=0, DIR=C:\Documents and Settings\User\桌面\PDS8...
Ucom3231>ipconf
IP=10.0.8.246
MASK=255.255.255.0
GATEWAY=10.0.8.254
MAC=00:0D:E0:E0:37:2B
DHCP=0
ACK_Delay=50
Free Memory=440848 bytes
Socket number=32,Free socket number=23
```

7.4.3 SOCKET

Description: This command lists the status of all sockets (Listen/Not Used Yet) together with the type of each socket (TCP Server: Port No./UDP/Unused)

If stat = 1, the socket is used. If stat = 0, the socket is not yet used.

Take Effect: Immediately

Command	Argument	Description
socket		Lists the status of all sockets.

Example:

```

Ucom3231>socket
[00=16:LISTEN],stat=1 , [01=16:LISTEN],stat=1
[02=16:LISTEN],stat=1 , [03=16:LISTEN],stat=1
[04=16:LISTEN],stat=1 , [05=16:LISTEN],stat=1
[06=16:LISTEN],stat=1 , [07=16:LISTEN],stat=1
[08=00:NOT_USED_YET],stat=0 , [09=00:NOT_USED_YET],stat=0
[10=00:NOT_USED_YET],stat=0 , [11=00:NOT_USED_YET],stat=0
[12=00:NOT_USED_YET],stat=0 , [13=00:NOT_USED_YET],stat=0
[14=00:NOT_USED_YET],stat=0 , [15=00:NOT_USED_YET],stat=0
[16=00:NOT_USED_YET],stat=0 , [17=00:NOT_USED_YET],stat=0
[18=00:NOT_USED_YET],stat=0 , [19=00:NOT_USED_YET],stat=0
[20=00:NOT_USED_YET],stat=0 , [21=00:NOT_USED_YET],stat=0
[22=00:NOT_USED_YET],stat=0 , [23=00:NOT_USED_YET],stat=0
[24=00:NOT_USED_YET],stat=0 , [25=00:NOT_USED_YET],stat=0
[26=00:NOT_USED_YET],stat=0 , [27=00:NOT_USED_YET],stat=0
[28=00:NOT_USED_YET],stat=0 , [29=00:NOT_USED_YET],stat=0
[30=00:NOT_USED_YET],stat=0 , [31=00:NOT_USED_YET],stat=0

Socket Type:
[00]:TCP Server:10001 , [01]:TCP Server:10010
[02]:TCP Server:10011 , [03]:TCP Server:10012
[04]:TCP Server:10013 , [05]:TCP Server:10000
[06]:TCP Server:23 , [07]:TCP Server:80
[08]:UDP , [09]:UnUsed
[10]:UnUsed , [11]:UnUsed
[12]:UnUsed , [13]:UnUsed
[14]:UnUsed , [15]:UnUsed
[16]:UnUsed , [17]:UnUsed
[18]:UnUsed , [19]:UnUsed
[20]:UnUsed , [21]:UnUsed
[22]:UnUsed , [23]:UnUsed
[24]:UnUsed , [25]:UnUsed
[26]:UnUsed , [27]:UnUsed
[28]:UnUsed , [29]:UnUsed
[30]:UnUsed , [31]:UnUsed
  
```

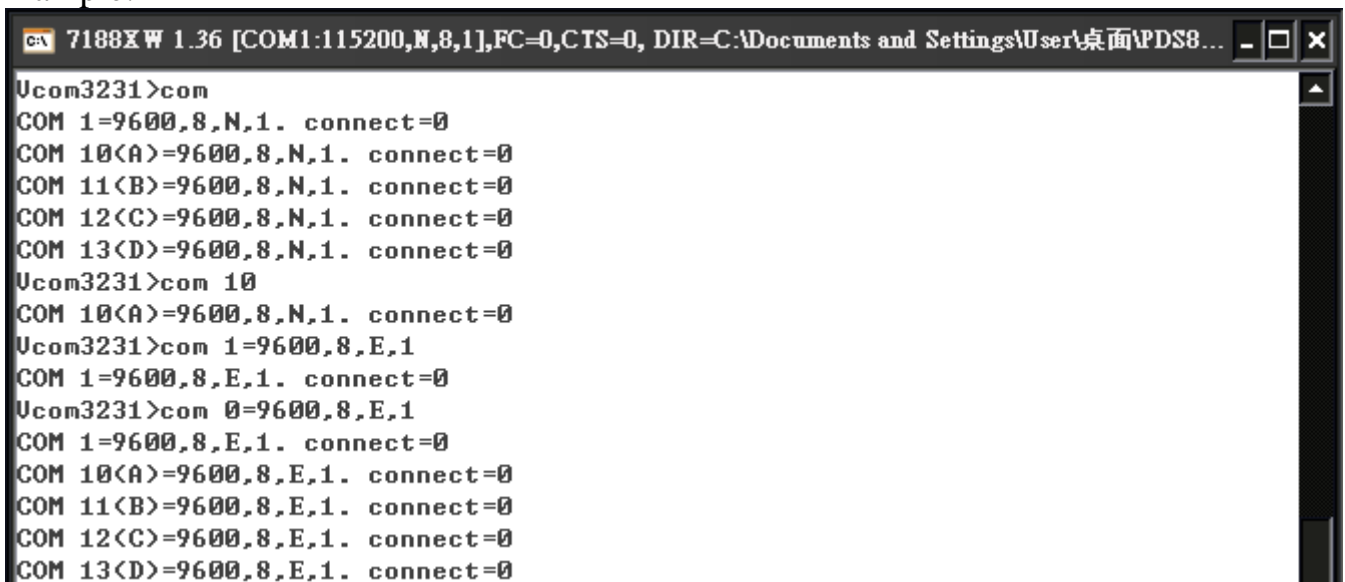
7.4.4 COM

Description: This command queries or sets the configuration of the COM Ports (BaudRate/DataBits/Parity/StopBits).

Effect: Immediate

Command	Arguments	Description
com		Queries the configuration of all COM Ports.
com	n	Queries configuration of COM Port “n”. If n = 0, the configuration of all COM Ports will be listed in the same way as using the command “com” above.
com	N = BaudRate, DataBits, Parity, StopBit(s)	Sets the configuration of COM Port “n”. If n = 0, the settings will be valid for all Com Ports on the PDS-800.

Example:



```
C:\ 7188X W 1.36 [COM1:115200,N,8,1],FC=0,CTS=0, DIR=C:\Documents and Settings\User\桌面\PDS8...
Ucom3231>com
COM 1=9600,8,N,1. connect=0
COM 10(A)=9600,8,N,1. connect=0
COM 11(B)=9600,8,N,1. connect=0
COM 12(C)=9600,8,N,1. connect=0
COM 13(D)=9600,8,N,1. connect=0
Ucom3231>com 10
COM 10(A)=9600,8,N,1. connect=0
Ucom3231>com 1=9600,8,E,1
COM 1=9600,8,E,1. connect=0
Ucom3231>com 0=9600,8,E,1
COM 1=9600,8,E,1. connect=0
COM 10(A)=9600,8,E,1. connect=0
COM 11(B)=9600,8,E,1. connect=0
COM 12(C)=9600,8,E,1. connect=0
COM 13(D)=9600,8,E,1. connect=0
```

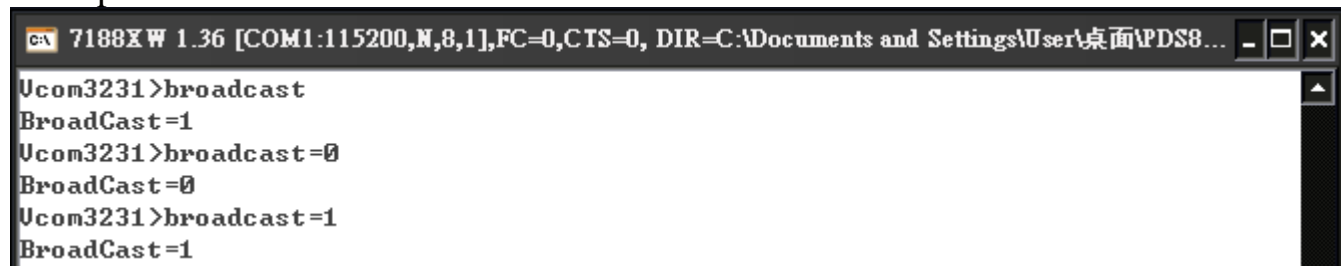
7.4.5 Broadcast

Description: This command is used to Enable/Disable listening broadcast packets on PDS-800.

Effect: Immediate

Command	Arguments	Description
Broadcast		Queries the Broadcast settings.
Broadcast	= 1	Sets Broadcast = 1. The system is able to receive broadcast packets.
Broadcast	= 0	Sets Broadcast = 0. The system will ignore broadcast packets.

Example:



```
7188X W 1.36 [COM1:115200,N,8,1],FC=0,CTS=0, DIR=C:\Documents and Settings\User\桌面\PDS8...  
Ucom3231>broadcast  
BroadCast=1  
Ucom3231>broadcast=0  
BroadCast=0  
Ucom3231>broadcast=1  
BroadCast=1
```

7.4.6 SystemTimeout (ms)

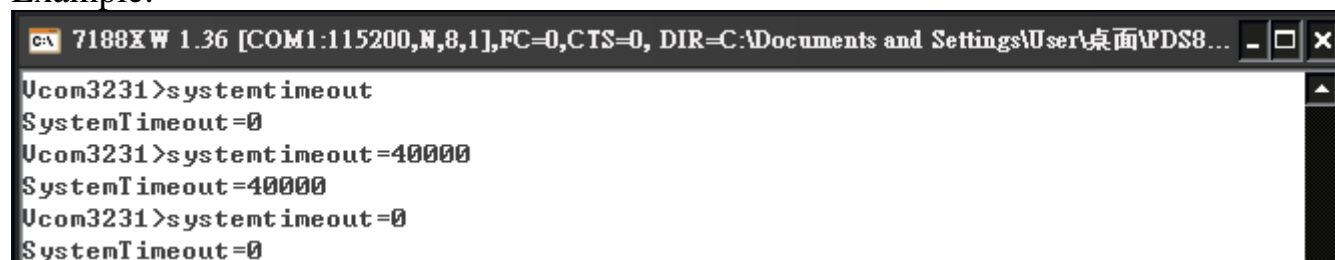
Description: This command queries or sets the system timeout value.

If the SystemTimeout is greater than zero, and the PDS-800 does not receive any packets from any client for longer than the SystemTimeout period, the PDS-800 will reboot itself.

Effect: Immediate

Command	Arguments	Description
SystemTimeout		Queries the SystemTimeout settings.
SystemTimeout	= nnnnn	Sets the SystemTimeout. (Unit : ms) Default factory setting is 300000 ms (= 300 seconds = 5 minutes), min. value is 30000 ms (= 30 seconds)

Example:



```
7188X W 1.36 [COM1:115200,N,8,1],FC=0,CTS=0, DIR=C:\Documents and Settings\User\桌面\PDS8...
Ucom3231>systemtimeout
SystemTimeout=0
Ucom3231>systemtimeout=400000
SystemTimeout=400000
Ucom3231>systemtimeout=0
SystemTimeout=0
```

7.4.7 SocketTimeout (ms)

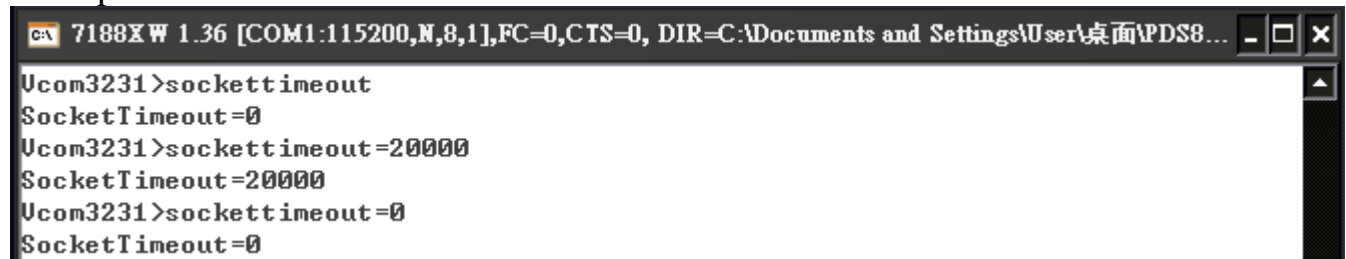
Description: This command is used to query or set the SocketTimeout parameter.

If the SocketTimeout is greater than zero, and the PDS-800 does not receive any data from a client PC for longer than the SocketTimeout period, the PDS-800 will close the socket connection between itself and the client PC.

Effect: Immediate

Command	Arguments	Description
SocketTimeout		Queries the SocketTimeout settings.
SocketTimeout	= nnnnn	Sets the SocketTimeout. (Unit : ms) default = 0 (disable), min = 10000

Example:



```
7188X W 1.36 [COM1:115200,N,8,1],FC=0,CTS=0, DIR=C:\Documents and Settings\User\桌面\PDS8...
Ucom3231>sockettimeout
SocketTimeout=0
Ucom3231>sockettimeout=200000
SocketTimeout=200000
Ucom3231>sockettimeout=0
SocketTimeout=0
```

7.4.8 M

Description: This command is used to query or set the echo mode.

Effect: Immediate

Command	Arguments	Description
M		Queries the echo mode settings.
M	= 0	Sets the multi-echo mode to enable. When set to multi-echo mode, the PDS-800 echoes data from a device to all clients that are connected.
M	= 1	Sets the single-echo mode to enable. When set to single-echo mode, the PDS-800 echoes data from a device to the client that requested the service.
M	= 2	Sets the half-slave mode to enable. (Is situated between M0 and the M1 mode) When set to half-slave mode, if is the single connection is equally operates with the M0 mode or has multiple connections is equally operates with the M1 mode.

Example:



```
c:\ 7188x W 1.36 [COM1:115200,N,8,1],FC=0,CTS=0, DIR=C:\Documents and Settings\User\桌面\PDS8... - □ ×
Ucom3231 >m
M=0
Ucom3231 >m=1
M=1
Ucom3231 >m=2
M=2
Ucom3231 >m=0
M=0
```

7.4.9 EchoCmdNo

Description: This command is used to query or set the EchoCmdNo parameter.

The EchoCmdNo parameter is used to set whether the PDS-800 prefixes the Virtual COM command to the corresponding response. (Virtual COM commands are used to configure a PDS-800 through TCP port 10000)

Effect: Immediate

Command	Arguments	Description
EchoCmdNo		Queries the EchoCmdNo settings.
EchoCmdNo	= 0	If EchoCmdNo = 0, a Virtual COM command number will not be prefixed to the corresponding response.
EchoCmdNo	= 1	If EchoCmdNo = 1, a Virtual COM command number will be prefixed to the corresponding response.

Example:

```
C:\> 7188X W 1.36 [COM1:115200,N,8,1],FC=0,CTS=0, DIR=C:\Documents and Settings\User\桌面\PDS8...
Ucom3231>echocmdno
EchoCmdNo=0
Ucom3231>echocmdno=1
EchoCmdNo=1
Ucom3231>echocmdno=0
EchoCmdNo=0
```

EchoCmdNo = 0

Send Command	
Send	13
Response	10.0.8.254

EchoCmdNo = 1

Send Command	
Send	13
Response	1310.0.8.254

7.4.10 EndChar

Description: This command is used to query or set the EndChar parameter.

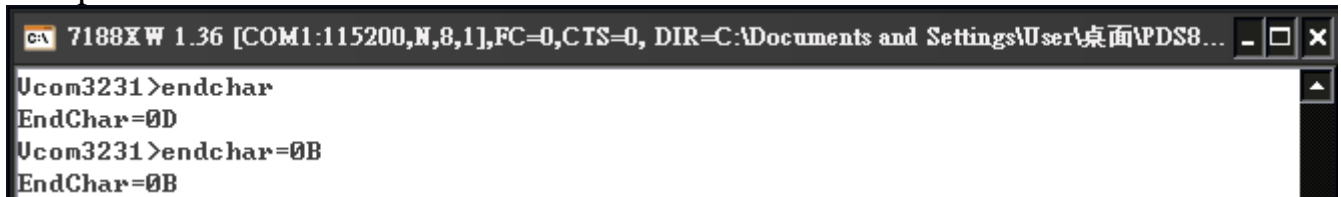
PDS-800 sends out the response string from serial port to TCP client immediately when it received a char on the response string that matching the EndChar.

Set EndChar = 00 to disable the EndChar feature.

Effect: Immediate

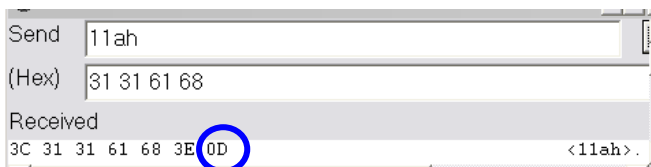
Command	Arguments	Description
Endchar		Queries the endchar setting.
Endchar	= HH	Sets the endchar.

Example:

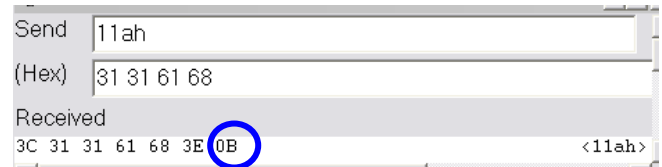


```
7188X W 1.36 [COM1:115200,N,8,1],FC=0,CTS=0, DIR=C:\Documents and Settings\User\桌面\WDS8...
Ucom3231>endchar
EndChar=0D
Ucom3231>endchar=0B
EndChar=0B
```

EndChar = 0D



EndChar = 0B



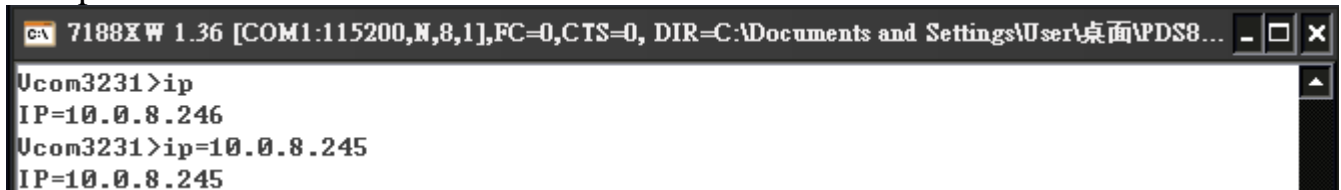
7.4.11 IP

Description: This command is used to query or set the IP address.

Effect: After the next reboot.

Command	Arguments	Description
IP		Queries the IP address.
IP	= xxx.xxx.xxx.xxx	Sets the IP address.

Example:



```
C:\ 7188X W 1.36 [COM1:115200,N,8,1],FC=0,CTS=0, DIR=C:\Documents and Settings\User\桌面\PDS8... - □ ×
Ucom3231>ip
IP=10.0.8.246
Ucom3231>ip=10.0.8.245
IP=10.0.8.245
```

7.4.12 MASK

Description: This command is used to query or set the subnet Mask value.

Effect: After the next reboot.

Command	Arguments	Description
mask		Queries the subnet Mask value.
mask	= xxx.xxx.xxx.xxx	Sets the subnet Mask value.

Example:



```
C:\ 7188X W 1.36 [COM1:115200,N,8,1],FC=0,CTS=0, DIR=C:\Documents and Settings\User\桌面\PDS8... - □ ×
Ucom3231>mask
MASK=255.255.255.0
Ucom3231>mask=255.255.255.254
MASK=255.255.255.254
```

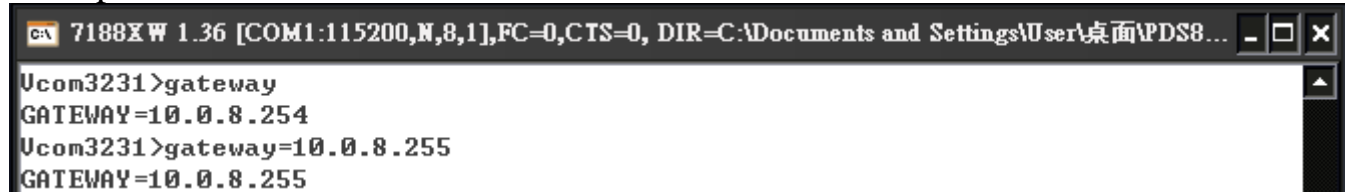
7.4.13 GATEWAY

Description: This command is used to query or set the outgoing Gateway address of the subnet.

Effect: After the next reboot.

Command	Arguments	Description
Gateway		Queries the Gateway address.
Gateway	= xxx.xxx.xxx.xxx	Sets the Gateway address

Example:



```
7188X W 1.36 [COM1:115200,N,8,1],FC=0,CTS=0, DIR=C:\Documents and Settings\User\桌面\PDS8...
Ucom3231>gateway
GATEWAY=10.0.8.254
Ucom3231>gateway=10.0.8.255
GATEWAY=10.0.8.255
```

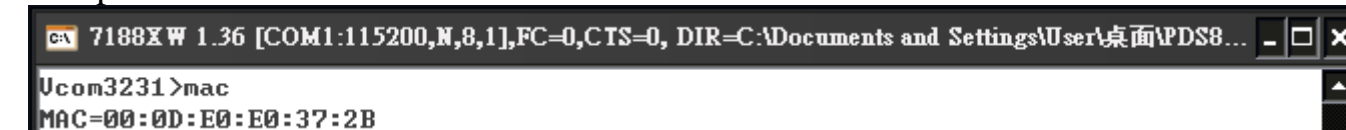
7.4.14 MAC

Description: This command is used to query the MAC address.

Effect: Setting the address is not allowed.

Command	Arguments	Description
Mac		Queries the MAC address.(Setting the address is not allowed)

Example:



```
7188X W 1.36 [COM1:115200,N,8,1],FC=0,CTS=0, DIR=C:\Documents and Settings\User\桌面\PDS8...
Ucom3231>mac
MAC=00:0D:E0:E0:37:2B
```

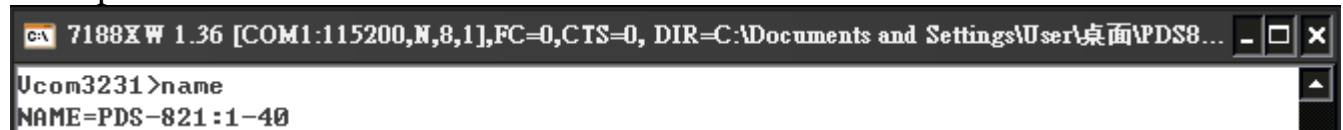
7.4.15 NAME

Description: This command is used to query the name of a PDS-800 controller.

Effect: Setting the name is not allowed.

Command	Arguments	Description
name		Queries the name of a PDS-800 controller.

Example:



```
C:\ 7188X W 1.36 [COM1:115200,N,8,1],FC=0,CTS=0, DIR=C:\Documents and Settings\User\桌面\PDS8...
Ucom3231>name
NAME=PDS-821:1-40
```

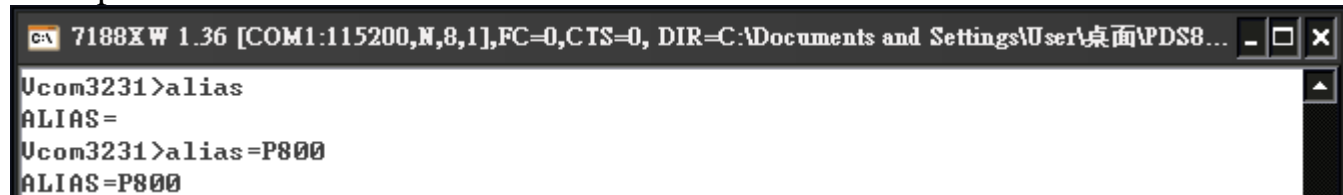
7.4.16 ALIAS

Description: This command is used to query or set the alias of a PDS-800 controller.
The maximum character length of PDS-800 alias name is 16 bytes.

Effect: Immediate

Command	Arguments	Description
alias		Queries the alias.
alias	= xxxx	Sets the alias of a PDS-800 controller to “xxxx”.

Example:



```
C:\ 7188X W 1.36 [COM1:115200,N,8,1],FC=0,CTS=0, DIR=C:\Documents and Settings\User\桌面\PDS8...
Ucom3231>alias
ALIAS=
Ucom3231>alias=P800
ALIAS=P800
```

7.4.17 DHCP

Description: This command is used to set the DHCP client to either enabled or disabled.

DHCP function will get a dynamic IP address setting for PDS-800 automatically. Thus it's recommended to disable DHCP function and use a static IP address setting. This prevents you to configure virtual COM mappings again and again.

Effect: Immediate

Command	Arguments	Description
DHCP	= 0	Disables the DHCP client.
DHCP	= 1	Enables the DHCP client.

Example:



```
7188X W 1.36 [COM1:115200,N,8,1],FC=0,CTS=0, DIR=C:\Documents and Settings\User\桌面\PDS8...
Ucom3231>dhcp
DHCP=0
Ucom3231>dhcp=1
DHCP=1
Ucom3231>dhcp=0
DHCP=0
```

7.4.18 UDP

Description: This command is used to configure the UDP Search function.

UDP is used to set the action mode for when a PDS-800 controller receives a UDP search command.

Effect: Immediate

Command	Arguments	Description
UDP	= 0	Rejects UDP search commands. The PDS-800 will not reply to the UDP search command, and can not be searched again.
UDP	= 1	Replies to UDP search commands. The PDS-800 controller will reply to the UDP search command, and can be searched.
UDP	= 2 (Default)	Replies to UDP search commands till a client is connected.

Example:



```
7188 W 1.36 [COM1:115200,N,8,1],FC=0,CTS=0, DIR=C:\Documents and Settings\User\桌面\PDS8...
Ucom3231>udp
UDP=2
Ucom3231>udp=0
UDP=0
Ucom3231>udp=1
UDP=1
Ucom3231>udp=2
UDP=2
```

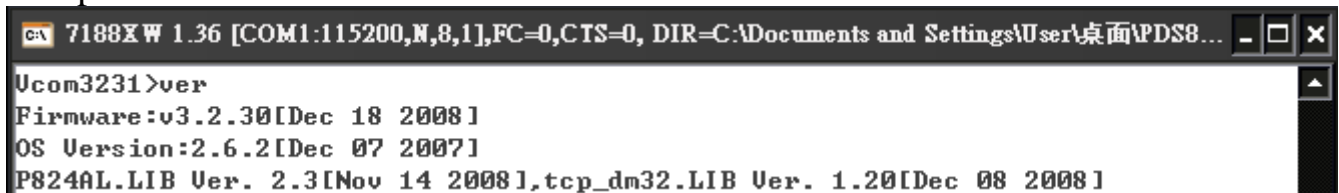
7.4.19 VER

Description: This command is used to query the version information for a PDS-800 controller.

Effect: Setting the version information is not allowed.

Command	Argument	Description
VER		Queries the version information.

Example:



```
Ucom3231>ver
Firmware:v3.2.30[Dec 18 2008]
OS Version:2.6.2[Dec 07 2007]
P824AL.LIB Ver. 2.3[Nov 14 2008],tcp_dm32.LIB Ver. 1.20[Dec 08 2008]
```

7.4.20 SAVE

Description: This command is used to set the PDS-800 controller to backup or no-backup "autoexec.bat" and "vcom.ini" files when using "load" command.

Effect: Immediate

Command	Arguments	Description
save	= 1	When the "load" command is used, backup copies of the "autoexec.bat" and "vcom.ini" files will be saved.
save	= 0 (Default)	When the "load" command is used, backup copies of the "autoexec.bat" and "vcom.ini" files will NOT be saved.

Example: see images [21-1] and [21-2] below.

7.4.21 LOAD

Description: This command is used to load file to the built-in flash disk on PDS-800 controller. It should be used to update firmware only.

Effect: Immediate

Command	Argument	Description
load		The command is coordinated with the MiniOS7 “load” command and can be used to renew the “vcom3230.exe,” “vcom.ini” or “autoexec.bat” file(s).

※”Load” is not a Telnet command.

Example:

[21-1]

```

C:\> 7188X W 1.36 [COM1:115200,N,8,1],FC=0,CTS=0, DIR=C:\Documents and Settings\User\桌面\PDS8...
P824A_UDP>dir

 0>vp8_3230.exe 12/18/2008 15:52:17 76421 [12A85]8002:0000-92AA:0005
 1>autoexec.bat 12/17/2008 15:02:06      8 [00008]92AC:0005-92AC:000D
Total File number is 2 Free space=382227 bytes
P824A_UDP>autoexec.bat

Ucom3231>save=0
Backup .ini file=0

Ucom3231>load
Press ALT_E to download file!
Input filename:vp8_3230.exe
Load file:vp8_3230.exe [crc=3036,0000]
Send file info. total 299 blocks
Block 299
Transfer time is: 9.859000 seconds

Ucom3231>quit

P824A_UDP>dir

 0>vp8_3230.exe 12/18/2008 15:52:17 76421 [12A85]8002:0000-92AA:0005
 1>autoexec.bat 12/17/2008 15:02:06      8 [00008]92AC:0005-92AC:000D
 2>vp8_3230.exe 12/18/2008 15:52:17 76421 [12A85]92AE:000D-A557:0002
Total File number is 3 Free space=305774 bytes
P824A_UDP>
  
```

※When save = 0, the system doesn't back up the "autoexec.bat" and "vcom.ini" files to memory, and doesn't clear the flash disk. It only loads the file that is selected specified.

[21-2]

```
C:\ 7188X W 1.36 [COM1:115200,N,8,1],FC=0,CTS=0, DIR=C:\Documents and Settings\User\桌面\PDS8...
P824A_UDP>dir
 0>vp8_3230.exe 12/18/2008 15:52:17 76421 [12A85 18002:0000-92AA:0005
 1>autoexec.bat 12/17/2008 15:02:06      8 [00008 192AC:0005-92AC:000D
Total File number is 2 Free space=382227 bytes
P824A_UDP>autoexec.bat

Ucom3231>save=1
Backup .ini file=1

Ucom3231>load
save file:autoexec.bat
Press ALT_E to download file!
Input filename:vp8_3230.exe
Load file:vp8_3230.exe [crc=3036,0000]
Send file info. total 299 blocks
Block 299
Transfer time is: 9.875000 seconds

Ucom3231>quit
Command not supported !

P824A_UDP>dir
 0>autoexec.bat 12/17/2008 15:02:06      8 [00008 18002:0000-8002:0008
 1>vp8_3230.exe 12/18/2008 15:52:17 76421 [12A85 18004:0008-92AC:000D
Total File number is 2 Free space=382227 bytes
P824A_UDP>
```

Save=1

Ucom3231>save=1
Backup .ini file=1

Load vp8_3230.exe

Ucom3231>load
save file:autoexec.bat
Press ALT_E to download file!
Input filename:vp8_3230.exe
Load file:vp8_3230.exe [crc=3036,0000]
Send file info. total 299 blocks
Block 299
Transfer time is: 9.875000 seconds

Ucom3231>quit
Command not supported !

P824A_UDP>dir
 0>autoexec.bat 12/17/2008 15:02:06 8 [00008 18002:0000-8002:0008
 1>vp8_3230.exe 12/18/2008 15:52:17 76421 [12A85 18004:0008-92AC:000D
Total File number is 2 Free space=382227 bytes
P824A_UDP>

※ When save = 1, system will back up the "autoexec.bat" and "vcom.ini" files to memory first, clear all files in the flash disk, and then load the "autoexec.bat" and "vcom.ini" files from memory and run the "load" command to load the specified file(s).

7.4.22 CONFIG

Description: This command is used to clear the settings in the EEPROM.

Effect: Immediate

Command	Argument	Description
config	= RESET	<p>Clears the settings in the EEPROM.</p> <p>After reboot, the firmware will use the new (default) settings on EEPROM.</p> <p><i>Note:</i> ("RESET" MUST be in capital letters.)</p>

※When “Config=RESET” is used, the Password, Alias and IPFILTER settings will also be cleared, but the IP/MASK/GATEWAY addresses will not.

※The SystemTimeout setting is also cleared to 0 by the "config=RESET" command, you have to configure the SystemTimeout value again. The default factory setting of SystemTimeout value should be 300000ms (= 300 seconds).

Example:



```
7188X W 1.36 [COM1:115200,N,8,1],FC=0,CTS=0, DIR=C:\Documents and Settings\User\桌面\PDS8...
Ucom3231>config=RESET
Ucom3231>_
```

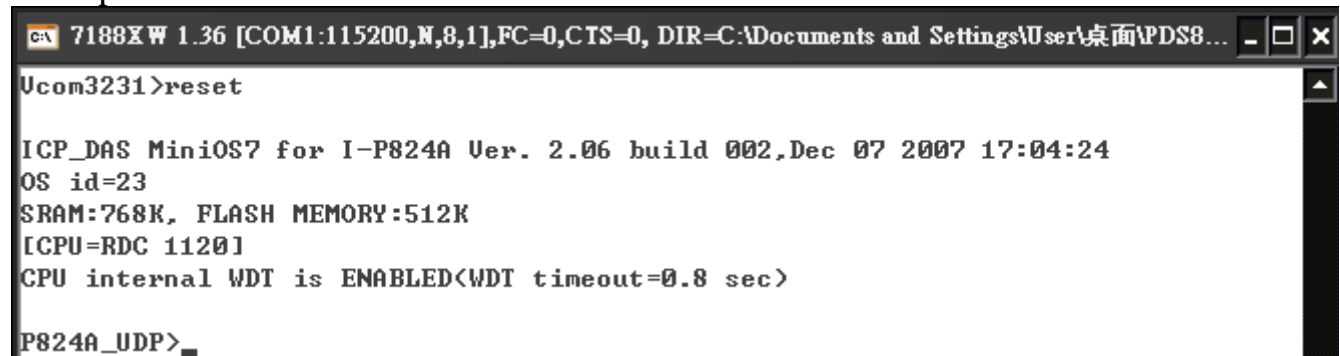
7.4.23 RESET

Description: This command is used to reboot the PDS-800 controller.

Effect: Immediate

Command	Argument	Description
reset		Reboot the PDS-800 controller .

Example:



```
7188X W 1.36 [COM1:115200,N,8,1],FC=0,CTS=0, DIR=C:\Documents and Settings\User\桌面\PDS8...
Ucom3231>reset

ICP_DAS MiniOS7 for I-P824A Ver. 2.06 build 002,Dec 07 2007 17:04:24
OS id=23
SRAM:768K, FLASH MEMORY:512K
[CPU=RDC 1120]
CPU internal WDT is ENABLED(WDT timeout=0.8 sec)

P824A_UDP>_
```

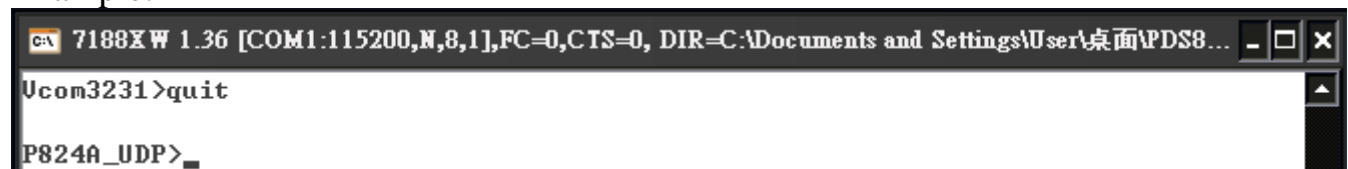
7.4.24 QUIT

Description: This command is used to stop and quit the firmware of PDS-800 controller.

Effect: Immediate

Command	Argument	Description
quit		Quits the firmware.

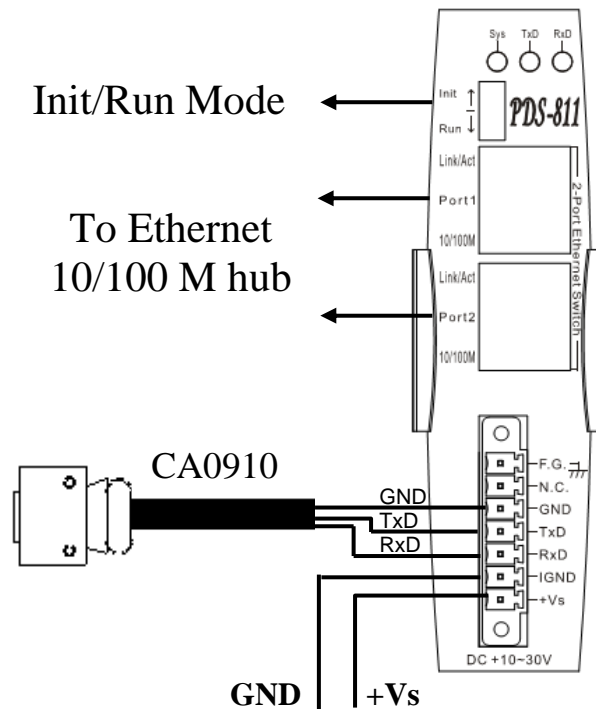
Example:




```
7188X W 1.36 [COM1:115200,N,8,1],FC=0,CTS=0, DIR=C:\Documents and Settings\User\桌面\PDS8...
Ucom3231>quit

P824A_UDP>_
```

Appendix A: Linking to a Development PC



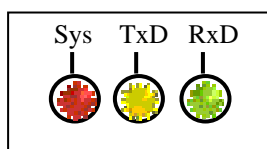
Step 1: Connect the download-cable, CA0910, between the PDS-800 controller and COM 1 of the development PC as per above the diagram.

Step 2: Regulate for “**Init Mode**” , as shown in the above diagram.

Step 3: Unzip the “7188XW_yyyymmdd.zip” file on the PC. The file is located in the CD:\Napdos\MiniOS7\utility folder.

Step 4: Apply power (+Vs, GND) to the PDS-800 controller. +Vs can be anywhere from +30 ~ +10 V.

Step 5: Check that the LED Display is continuously showing the following information:



Note: Only display versions of PDS-800 controller have a System, TxD and RxD LED Display.

Step 6: Execute 7188XW.EXE/C#, and change the Baud Rate to 115200 bps, N81. “/C#” is the COM Port of the development PC.

Step 7: Press [← Enter] twice on the development PC:

```
7188XW 1.36 [COM1:115200,N,8,1],FC=0,CTS=0, DIR=C:\n
7188x for WIN32 version 1.36 <2008/02/26>[By ICP DAS. Tim Tsai.]
[Begin Key Thread...]Current set: Use COM1 115200,N,8,1
AutoRun:
Autodownload files: None
Current work directory="C:\n"
original baudrate = 115200!
now baudrate = 115200!

P824A_UDP>_
```

Step 8: Read the configuration of the PDS-800 (The following is a default value):

```
P824A_UDP>ip
IP=192.168.255.1
P824A_UDP>mask
MASK=255.255.0.0
P824A_UDP>gateway
Gateway=192.168.0.1
P824A_UDP>mac
Ethernet Address = 00:0d:e0:e0:37:2b
P824A_UDP>setcom 1
Current set is: 9600,8,0,1
```

- Read configuration command
- ip
 - mask
 - gateway
 - mac
 - setcom port


Note: The configuration of the PDS-800 as follows can be changed:
(Please setting for with your PC same network region **IP/Mask/Gateway**. *This example: 10.0.8.246/255.255.255.0/10.0.8.255*)

```
P824A_UDP>ip 10.0.8.246
Set IP=10.0.8.246
[ReadBack]IP=10.0.8.246
P824A_UDP>mask 255.255.255.0
Set MASK=255.255.255.0
[ReadBack]MASK=255.255.255.0
P824A_UDP>gateway 10.0.8.255
Set GATEWAY=10.0.8.255
[ReadBack]Gateway=10.0.8.255
P824A_UDP>setcom 1 115200,n,8,1
Current set is: 9600,8,0,1
Set to: 115200,8,0,1 [checksum:CC]
```

- Settings configuration command
- ip [new ip]
 - mask [new mask]
 - gateway [new gateway]
 - mac [new mac]
 - setcom port
[baud][data_bit][parity][stop_bit]

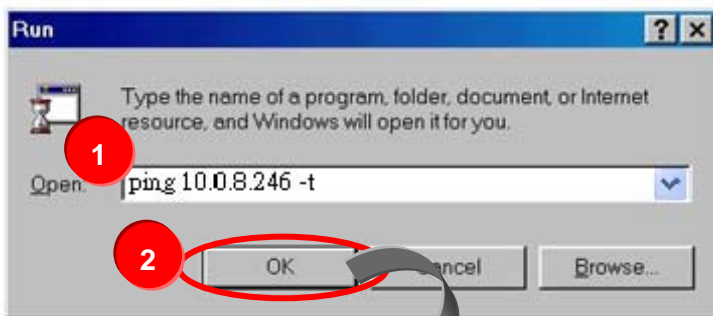
“setcom” parameters are as follows:

Port	1
Baud	2 - 921600
Data Bit	7, 8: for COM 1
Parity	N, n : None parity E, e : Even parity O, o : Odd parity
Stop Bit	1: for COM 1

Step 9: Regulate for “Run Mode”. 

Step 10: Power-off the module then power on again.

Step 11: Execute ping you setting IP (Example: **ping 10.0.8.246 -t**) using a run command as follows:



Execute **Ping 10.0.8.246 -t** on the PC client.

```
Pinging 10.0.8.246 with 32 bytes of data:
Reply from 10.0.8.246: bytes=32 time=1ms TTL=64
Reply from 10.0.8.246: bytes=32 time<1ms TTL=64
Reply from 10.0.8.246: bytes=32 time<1ms TTL=64
Reply from 10.0.8.246: bytes=32 time<1ms TTL=64
Reply from 10.0.8.246: bytes=32 time<1ms TTL=64
Reply from 10.0.8.246: bytes=32 time<1ms TTL=64
Reply from 10.0.8.246: bytes=32 time<1ms TTL=64
Reply from 10.0.8.246: bytes=32 time<1ms TTL=64
Reply from 10.0.8.246: bytes=32 time<1ms TTL=64
Reply from 10.0.8.246: bytes=32 time<1ms TTL=64
```

The ping results should be smooth and continuous

Note:

- **192.168.255.1** is the default IP of the PDS-800 controller. The IP address can be changed using the instructions in step 8.
- If the PDS-800 cannot be successfully pinged from the PC, refer to step 8 to change the configuration of the PDS-800 controller. (The mask and gateway addresses of PDS-800 controller and the PC should make the network definition.)
- The MAC address of the PDS-800 controller should be unique on the same network. Refer to step 8 for details of how to change the MAC address of the PDS-800 controller.
- Each PDS-800 MAC address is unique in the default shipping.

In general, if the host PC can ping the PDS-800 controller smoothly and continuously, all other software and drivers for the PDS-800 controller will operate correctly. Therefore, users should ensure that the development PC is able to ping the PDS-800 controller smoothly before any further testing is carried out.

Appendix B: Glossary

1. Ethernet

The term Ethernet generally refers to a standard published in 1982 by Digital Equipment Corp., Intel Corp. and Xerox Corp. Ethernet is the most popular physical layer local area network (LAN) 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, the Internet is 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. The Internet recognizes hosts using 32-bit IP address.

3. TCP/IP

The transmission Control Protocol (TCP) and the 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 networks.

4. TCP (Transmission Control Protocol)

TCP provides a reliable flow of data between two hosts. It is associated with tasks 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 will reach the destination host.

6. Gateway

Computers that interconnect two networks and pass packets from one to the other are called Internet Gateways or Internet Routers. Gateways route packets that are based on the destination network, not on the 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 numbers, 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 integer 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 the Ethernet interface hardware. Thus, no two hardware interfaces have the same MAC address.

9. Subnet Mask

Subnet mask is often simply called the 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, or (2) on a different network. If (1), the packet will be delivered directly; otherwise if, will be delivered via gateways or routers.

10. ARP (Address Resolution Protocol)

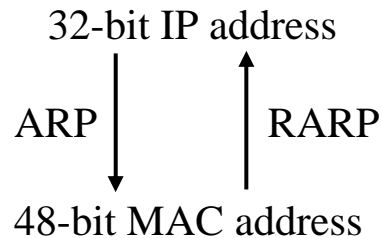
Consider two machines A and B that share a physical network. Each has an assigned IP address IP_A and IP_B , and a MAC address the MAC_A and MAC_B . The goal is to devise low-level software that hides MAC addresses and allows higher-level programs to work only with the IP addresses. Ultimately, however, communication must be carried out by the 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 are both attached, but A only has the Internet address for B, IP_B . The question arises: how does A map that address to the MAC address for B, MAC_B ?

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

11. RARP (Reverse Address Resolution Protocol)

RARP provides a method of dynamically mapping 48-bit MAC address to the corresponding 32-bit IP address.



12. ICMP (Internet Control Messages Protocol)

No system works correctly all the time. ICMP provides a method of communicating 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 allows a host to know what is 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 a host cannot be pinged, you won't be able to use Telnet or FTP to connect to the host. Conversely, if Telnet or FTP cannot be used to connect to a host, Ping is often the starting point to determine what the problem is.

14. Packet

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

15. Socket

Each TCP segment contains the source and destination port number that can be used 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 categorize 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 programs consist of conventional application program develop tools. Each time a client program is executed; it contacts a server, sends a request and waits for 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.

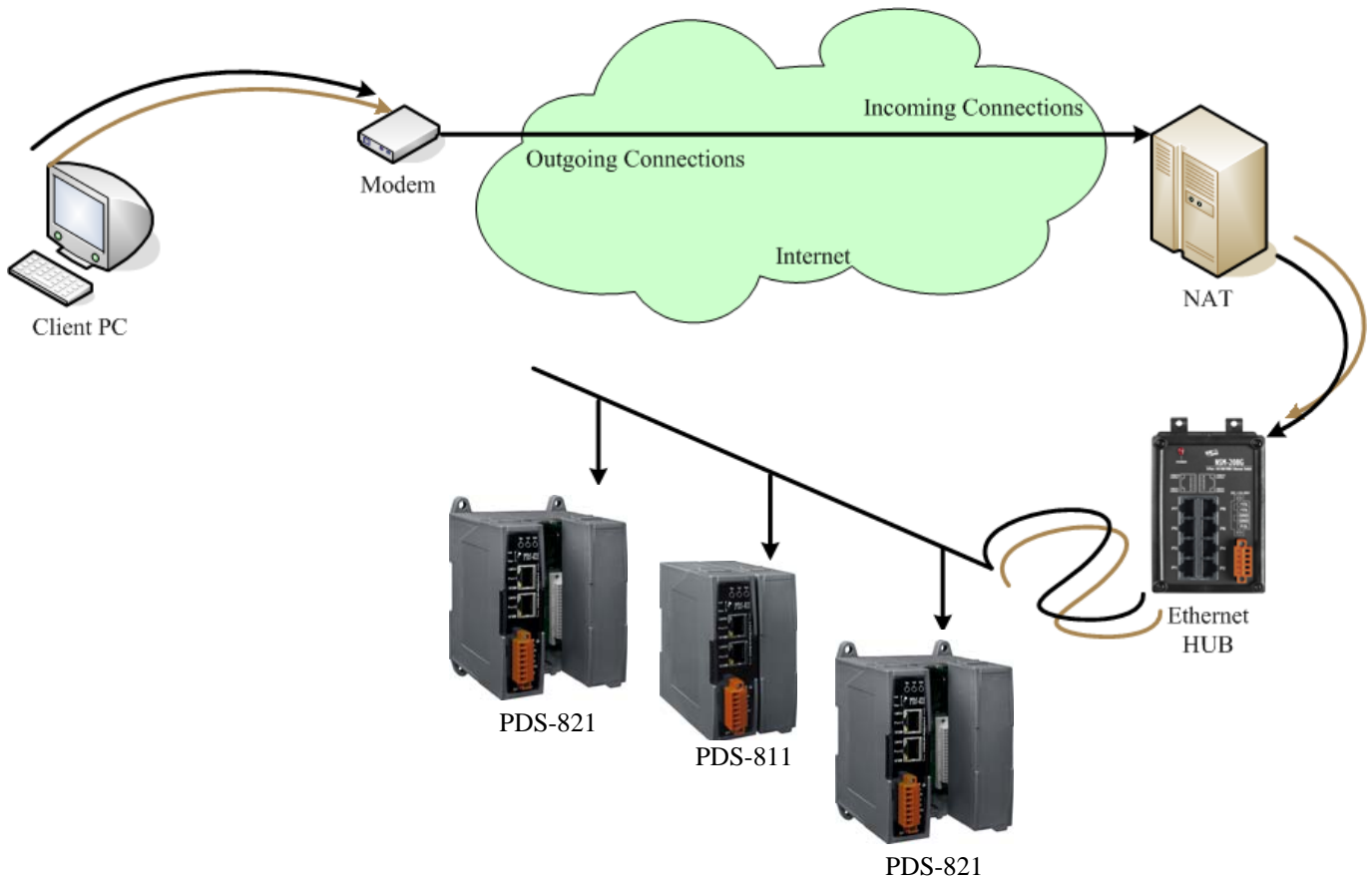
By comparison, a server is any program that waits for incoming requests from a client program. The server receives a request from a client, performs the necessary computation and returns the result to the client.

17. Firmware

Firmware is an alterable program located or stored in the semi-permanent storage area, e.g., ROM, EEPROM, or Flash memory.

FAQ

1. How to access the remote PDS-800 that placed behind an NAT or firewall?



The remote site must have a NAT (or a router supports NAT) server. NAT stands for Network Address Translator.

By using (configuring) the NAT server, NAT can forward (bypass) all specified TCP port connection to specified PDS-800 devices.

For example:

- *PDS-800 #1 with 4-Port serial module * 2 pcs*
TCP ports **10000~10001** of NAT maps to **Command and COM1** of PDS-800 #1
TCP ports **10010~10017** of NAT maps to **COM1 ~ COM8** of serial modules on PDS-800 #1
- *PDS-800 #2 with 4-Port serial module * 2 pcs*
TCP ports **10020~10021** of NAT maps to **Command and COM1** of PDS-800 #2

TCP ports **10030~10037** of NAT maps to **COM1 ~ COM8** of serial modules on PDS-800 #2

Please note, if your NAT (router) built-in a firewall feature, you have to configure the NAT to allow incoming TCP port connections.

For example:

PDS-800 #1:

TCP port includes 10000 ~ 10001 and 10010 ~ 10017 of NAT

PDS-800 #2:

TCP port includes 10020 ~ 10021 and 10030 ~ 10037 of NAT

In the VxComm Utility, you have to connect to the PDS-800 by using NAT's address and NAT's TCP ports instead of PDS-800's setting.

For example:

To add PDS-800 #1, it's IP: Port should be NAT: 10000.

To add PDS-800 #2, it's IP: Port should be NAT: 10020.

2. How to open a virtual COM port that larger than "COM 10" by calling CreateFile() Win32 API?

If you want to open "COM 11", the correct way to call the CreateFile() is as follows:

```
CreateFile(  
    "\\.\COM11",    // address of name of the communications device  
    fdwAccess,     // access (read-write) mode  
    0,            // share mode  
    NULL,         // address of security descriptor  
    OPEN_EXISTING, // how to create  
    0,           // file attributes  
    NULL         // handle of file with attributes to copy  
);
```

Notes:

1. This syntax also works for ports COM 1 through COM 10. See more... MS [Q115831](#).
2. Maximum COM port number for VxComm Driver is COM 256.
3. Valid COM port number for MSCOMM.OCX is between 1 to 16. Please refer to [MSComm.CommPort](#).
4. The "\\.\\" prefix must be add to the COM port name (device name) when it is larger than "COM 10". But please note that the "\" character is a special escape symbol in C\C++ language, thus you have to use "\\.\\" prefix in C\C++ language.

3. Does VxComm Driver (PC) support auto-reconnection after fixing a network break?

Yes, the [VxComm Driver](#) (PC) supports the auto-reconnection mechanism in version 2.00 and above. The [VxComm Utility](#) allows the user to set the Keep-Alive Time (ms) and Connection-Broken Time (ms) in the server options.

For more details, refer to the "[Adding a 7188E/8000E/PDS-700/DS-700/PDS-800 server and configuring the VxComm Driver](#)" section of the [VxComm Driver/Utility User Manual](#).

4. Why doesn't the VxComm Driver (PC) receive data from the 7188E/8000E/PDS-700/DS-700/PDS-800 module?

Make sure that the [7188E/8000E/PDS-700/DS-700/PDS-800](#) module is operating in mode 0 (/M0). [7188E/8000E/PDS-700/DS-700/PDS-800](#) module has the following two communication modes:

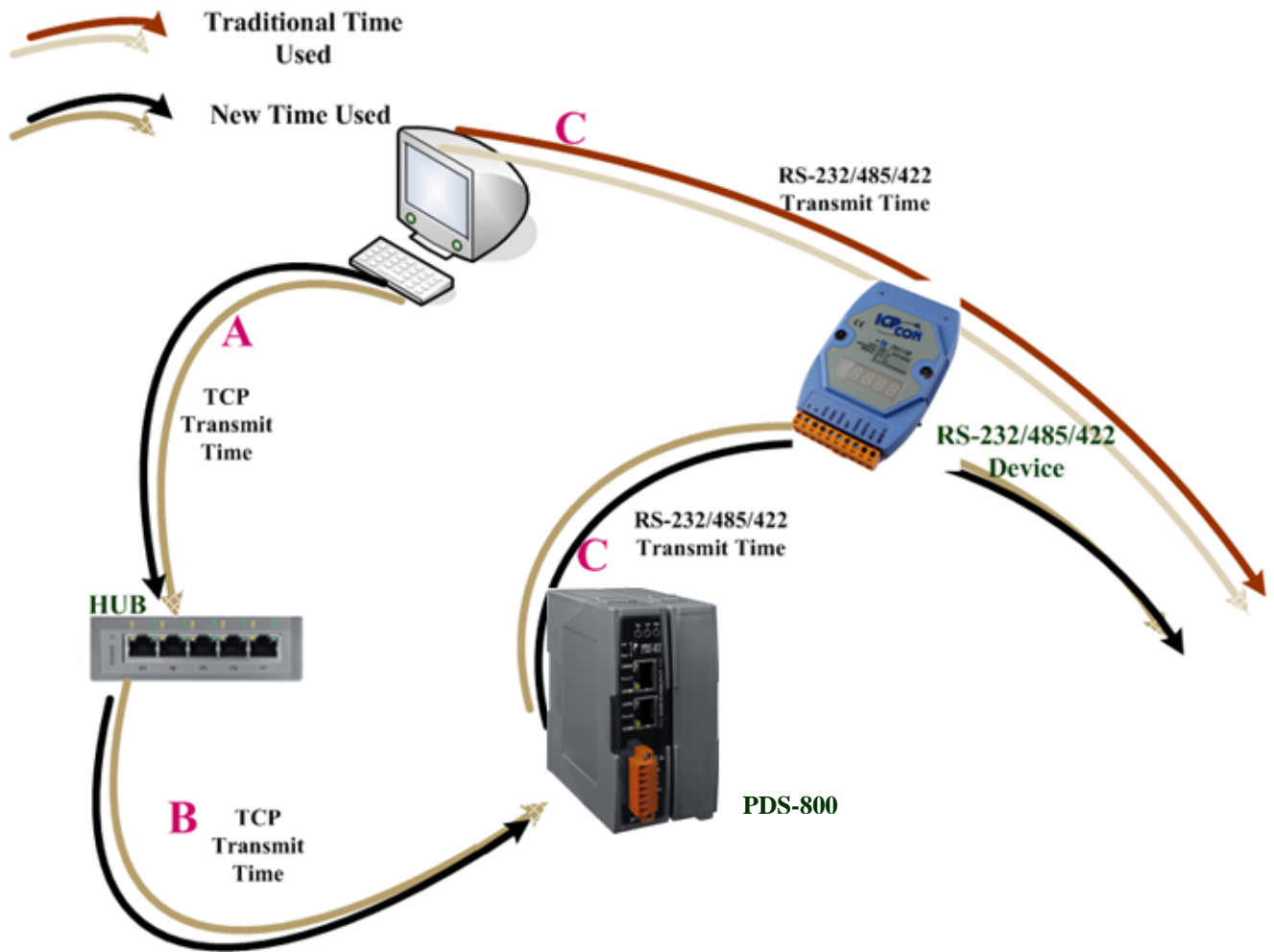
/M0	Transparent Mode (Multi-echo, shared). In this mode, data is echoed from the COM Ports of the 7188E/8000E/PDS-700/DS-700/PDS-800 module to each client that is connected to the 7188E/8000E/PDS-700/DS-700/PDS-800 module.	
/M1	Slave Mode (Single-echo, Non-Shared). In this mode, data is echoed from the COM Ports of the 7188E/8000E/PDS-700/DS-700/PDS-800 module to the specific client that requested the service.	Version 2.6.12 and above
/M2	Half-slave mode to enable. (Is situated between M0 and the M1 mode) In this mode, if is the single connection the same as M0 mode or has multiple connections the same as M1 mode.	Version 2.6.12 and above

In /M1 mode, if the client does not send a request to the COM port of the [7188E/8000E/PDS-700/DS-700/PDS-800](#) module, then the module won't return any data to it. For more information, please refer 6.4 “[COM Port Settings](#)” section.

Other reasons causing the problem may be: **incorrect wiring, power supply problems IP conflicts, MAC conflicts, an incorrect subnet mask or an invalid IP address.** For more details, refer to the "[Diagnostics and Troubleshooting](#)" section of the VxComm Driver/Utility User Manual.

5. Does the transmission speed become faster when the serial device working with Serial to Ethernet device servers?

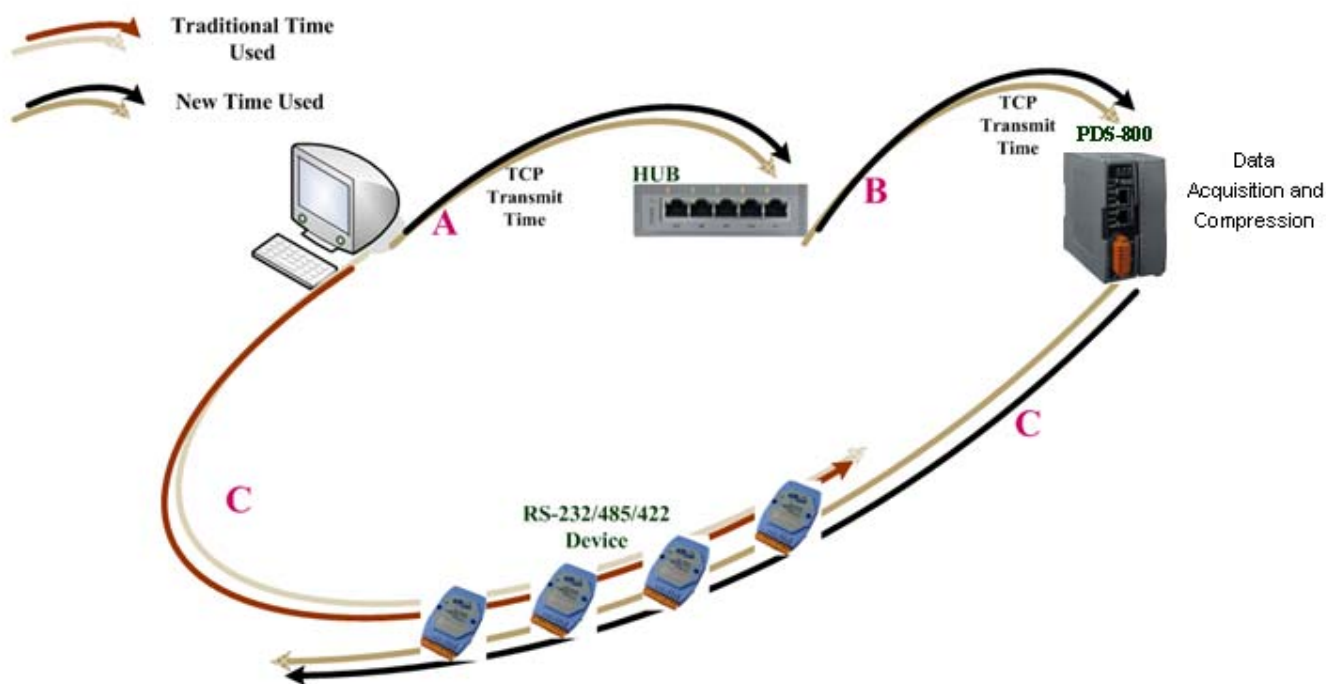
The speed depends on the applications. For transparent applications, it includes Ethernet latency in transmission and may get slower. But you can improve the communication speed by increasing the baud rate since you placing the device server more close to serial device and reduce the communication distance. The higher baud rate should be able to be used in short cable (distance) without communication problem.



Traditional time used	RS-232/485/422 transmit time (C)
New time used	Internet/Ethernet transmit time + RS-232/485/422 transmit time (A+B+C)

(All TCP packets need an extra ACK packet to commit the transmit action. This also causes a little additional delay in communication).

For **Xserver** applications, it can become faster. User can write their own Xserver applications to acquire data automatically, and then compress and transmit this large amount data at one time. Your application can reach high performance by pre-acquire data before asking by client and then response immediately.



Traditional time used	RS-232/485/422 transmit time ($C * n$ modules)
New time used	Internet/Ethernet transmit time ($A + B + C$)

6. Why does the 7188E/8000E/PDS-700/DS-700/PDS-800 module fail on a (public) Internet connection?

The default IP address of the 7188E/8000E/PDS-700/DS-700/PDS-800 module is 192.168.255.1, which can be only used on a private Internet connection. A private network packet will not be routed via a (public) Internet connection, which is the reason why the 7188E/8000E/PDS-700/DS-700/PDS-800 module failed on the Internet.

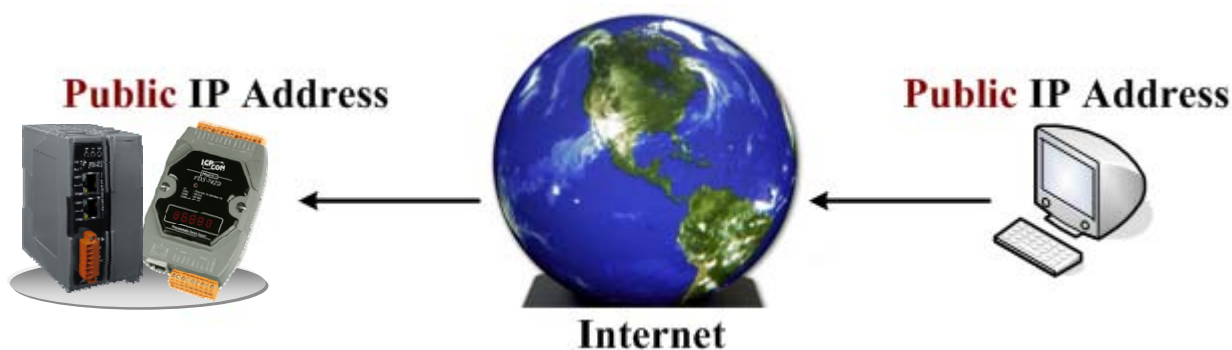
The IANA has reserved three address spaces for private internets (RFC 1918).

10.0.0.0 - 10.255.255.255 (10/8 prefix)

172.16.0.0 - 172.31.255.255 (172.16/12 prefix)

192.168.0.0 - 192.168.255.255 (192.168/16 prefix)

The 7188E/8000E/PDS-700/DS-700/PDS-800 module can operate on the Internet using a legal public IP address. This address can be obtained from your ISP or network administrator.



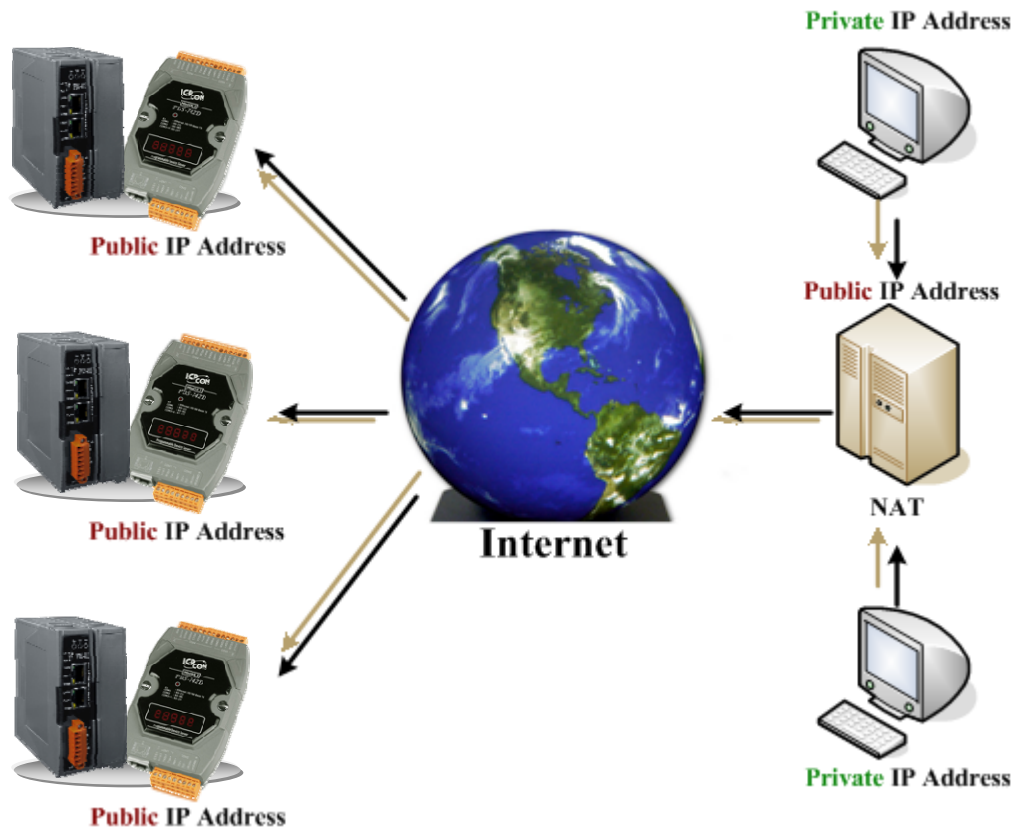
Note:

IANA: Internet Assigned Numbers Authority

RFC: Request for Comments

ISP: Internet Service Providers

A private internet client may communicate with a public Internet server (7188E/8000E/PDS-700/DS-700/PDS-800 modules) only if the NAT service for the client is available.



Note:

NAT: Network Address Translator

7. Can I use the SetCommState () API to changes the Baud Rate/data format settings of a virtual COM port?

Yes. In a Win32 environment, the CreateFile() API should be called to open the COM Port(s) and then the SetCommState() API can be used to configure the settings.

Third-party tools may provide an OpenCom() function for accessing a COM port. In actuality, the CreateFile() and SetCommState() APIs must be used to implement these kinds of functions.

8. How many PCs can be connected to a single PDS-800 device?

This depends on how many serial ports are available on the PDS-800 controller and how many serial ports which can be connected to each PC of PDS-800.

The PDS-800 controller has 32 sockets in total includes some reserved listening sockets. The PDS-800 controller provides a single command port for configuring the data (serial) ports. Thus, no matter how many data (serial) ports on the PDS-800 are used, one more socket connection is needed for the command port in order to configure them.

Data (Serial) Ports	Listening Sockets	Available Sockets	Max. PCs when using all data ports	Max. PCs when using 1 data port
PDS-811 Serial Modules				
2	6	$32 - 6 = 26$	$25/7 = 3$	$25/2 = 12$
4	8	$32 - 8 = 24$	$23/9 = 2$	$23/2 = 11$
PDS-821 Serial Modules				
4	8	$32 - 8 = 24$	$23/9 = 2$	$23/2 = 11$
6	10	$32 - 10 = 22$	$21/11 = 1$	$21/2 = 10$
8	12	$32 - 12 = 20$	$19/13 = 1$	$19/2 = 9$

Notes:

1. CMD Port = Command Port (TCP port 10000). The CMD Port is used to configure the data ports (TCP port 10010 ~ 10017) of a PDS-800 controller, such as BaudRate, and data format, etc.
2. The data port (TCP port 10010 ~ 10017, which are mapped to serial ports 10 ~ 17 of PDS-800), is only used to send/receive data.
3. The Listening Sockets (for PDS-800 controller) = Number of Data ports + 1 CMD port + Web + Telnet + UDP Search.
4. The number of Available Sockets (for PDS-800 controller) = max. (32) sockets - Listening sockets.
5. The maximum number of PCs when using all data ports of PDS = Available sockets / (data ports + 1 command port).
6. The maximum number of PCs when using 1 data port of PDS = Available sockets / (1 data port + 1 command port).
7. The web uses the TCP port 80. (It can be disabled.)
8. The telnet uses the TCP port 23. (It can be disabled)
9. The UDP search function will occupy one socket.
 - UDP = 0 → Doesn't support UDP search
 - UDP = 1 → Support UDP search and always occupy one socket
 - UDP = 2 → Support UDP search but while has the connection in that UDP search will be stopped.

9. Can I search or connect to PDS-800 when my PC's IP address is not in the IP filter list of PDS-800? How can I solve it?

- A. No, you cannot search or connect to PDS-800 when the PC's IP address is not in the IP filter list of PDS-800. You can solve the problem by:
1. Add your PC's IP address into the IP filter table of PDS-800 by using console command "IPFILTER" in section 7.4.1.
 2. Or change your PC's IP address to one of the IP addresses listed in IP filter table.
 3. Or disable the IP filter function of the PDS-800 by using console command. Refer to section 7.4.1.
 4. Or clear all configuration setting on PDS-800 by using "config=RESET" command on section 7.4.22. It also clears the IP filter table, password, alias... setting. You have to reboot the PDS-800 for loading new configuration.