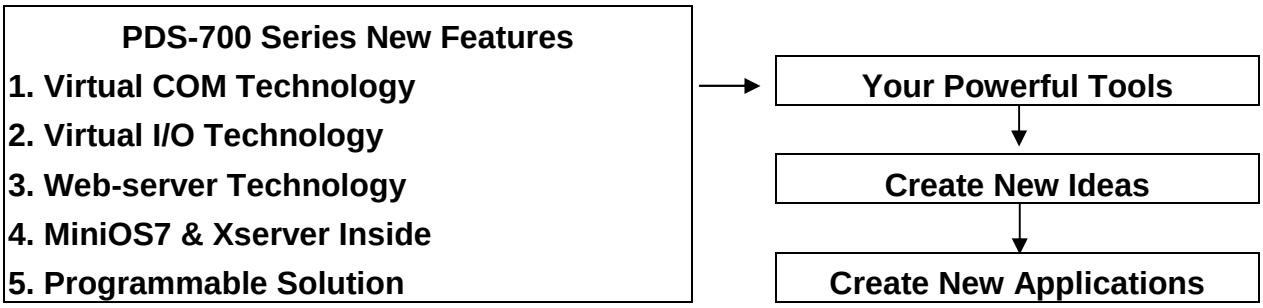
PDS-700 Series

User's 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 furnished by ICP DAS is believed to be accurate and reliable. However, no responsibility is assumed by ICP DAS for its use, nor for any infringements of patents or other rights of third parties resulting from its use.

Copyright

Trademark

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

Table of Contents

Packing List

The package includes the following items:

- One PDS-700 series hardware module
- One printed quick start guide
- One software utility CD
- One download cable, CA-0910

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-700\Readme.htm
CD:\NAPDOS\PDS\PDS-700\Document\Readme.htm

VxComm Driver (Virtual COM)
CD:\NAPDOS\7188e\TCP\VxComm\driver(pc)

Firmware
CD:\NAPDOS\PDS\PDS-700\VxComm\Server(PDS)

MiniOS7
CD:\NAPDOS\PDS\PDS-700\OS_image

1. Introduction

The PDS-700 series is a series of **Programmable Serial-to-Ethernet Device Servers** designed to meet the most common requirements of Internet/Ethernet applications. Besides the advantage of easy to use on remote control your serial devices through Ethernet network, the PDS-700 comes with a powerful and reliable **Xserver** programming structure for you to design your robust Ethernet applications in one day.

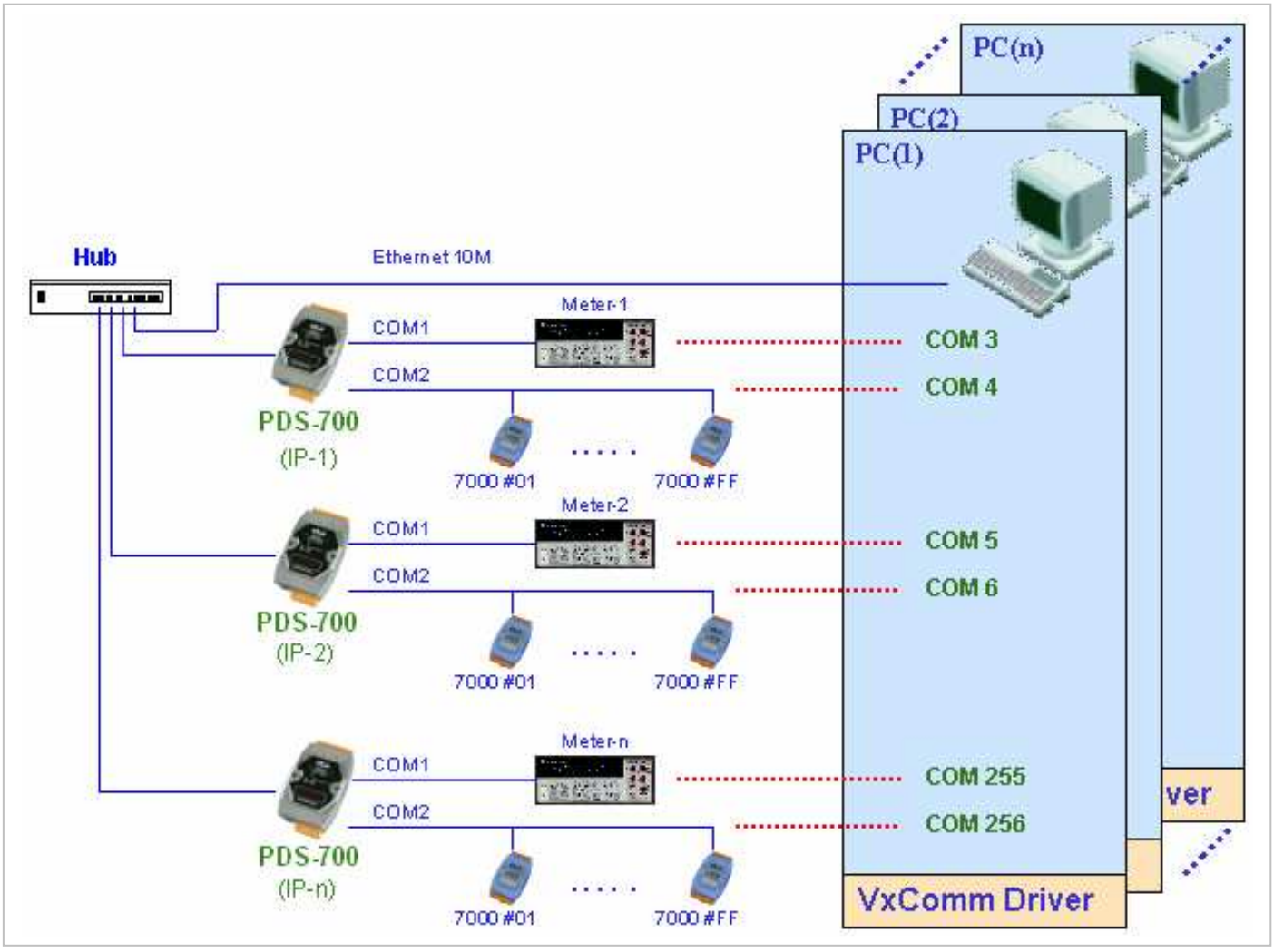
1.1 Why Ethernet Solutions?

Nowadays 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. Ethernet can link office automation and industrial control networks, access remote systems and share data and information between multiverdor machines; it also provides a cost-effective solution for your industrial control network.

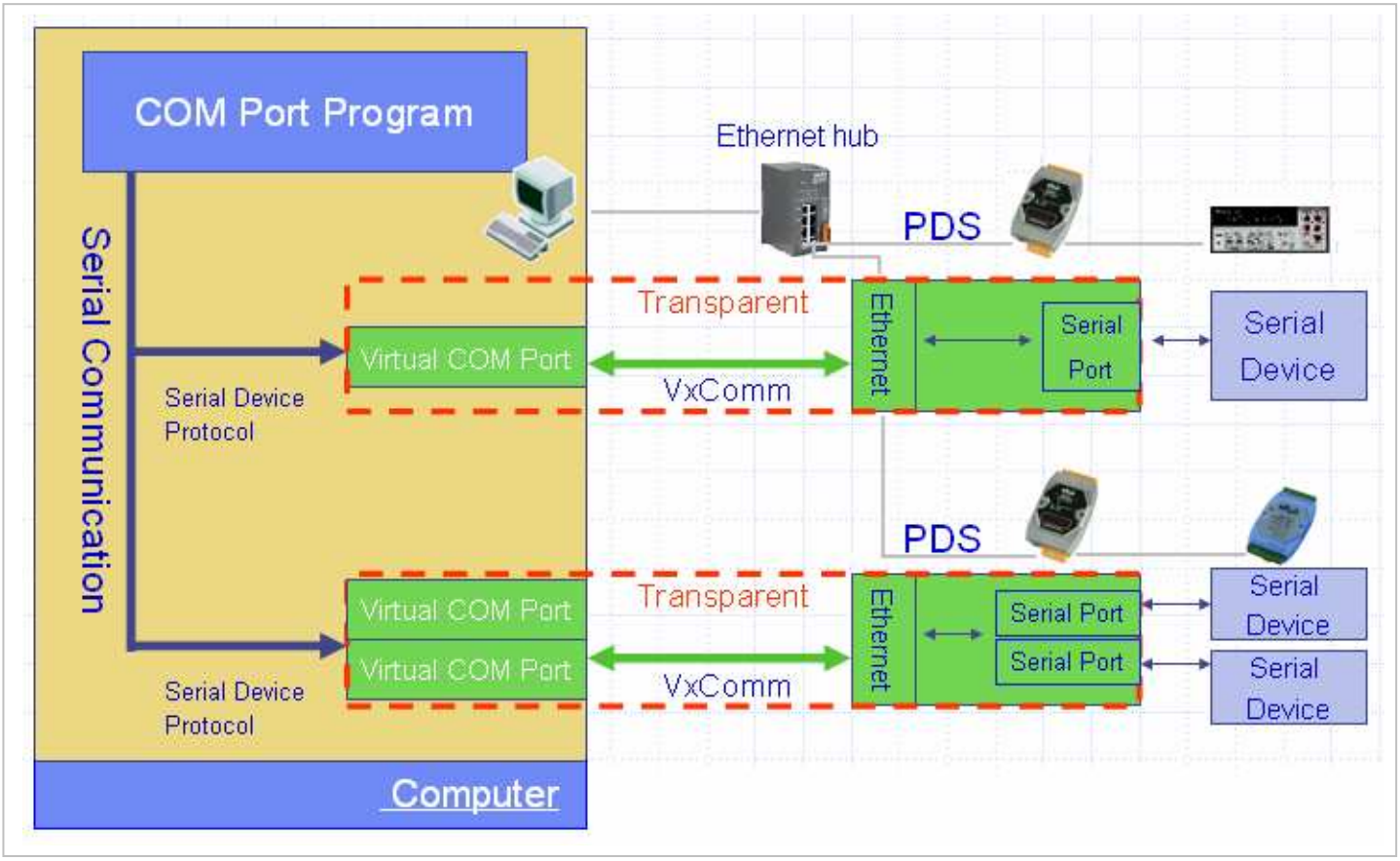
1.2 Why VxComm Technology?

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

Therefore, the new technology **VxComm** is developed to **virtualize the COM ports of the PDS-700 to become COM 3/4/5…/256 of the central computer.** The VxComm driver saves your time to access your serial devices through Ethernet without reprogramming PC's original COM port software.

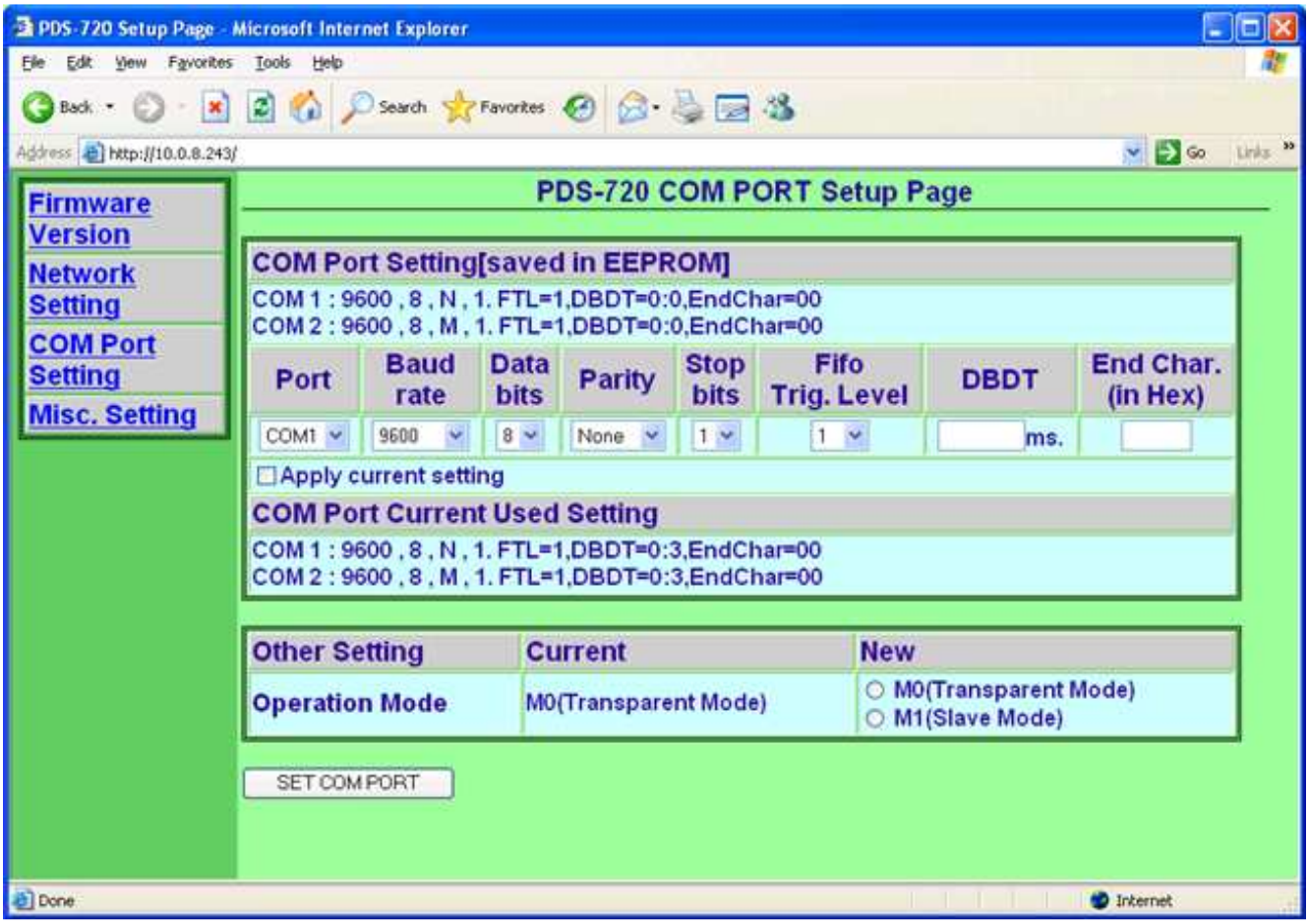


The VxComm driver handles all the detail of Ethernet TCP/IP program technique; your COM port program will access your serial devices through Ethernet as through COM port by the help of PDS-700 and VxComm technology.



1.3 Why Web Server Technology?

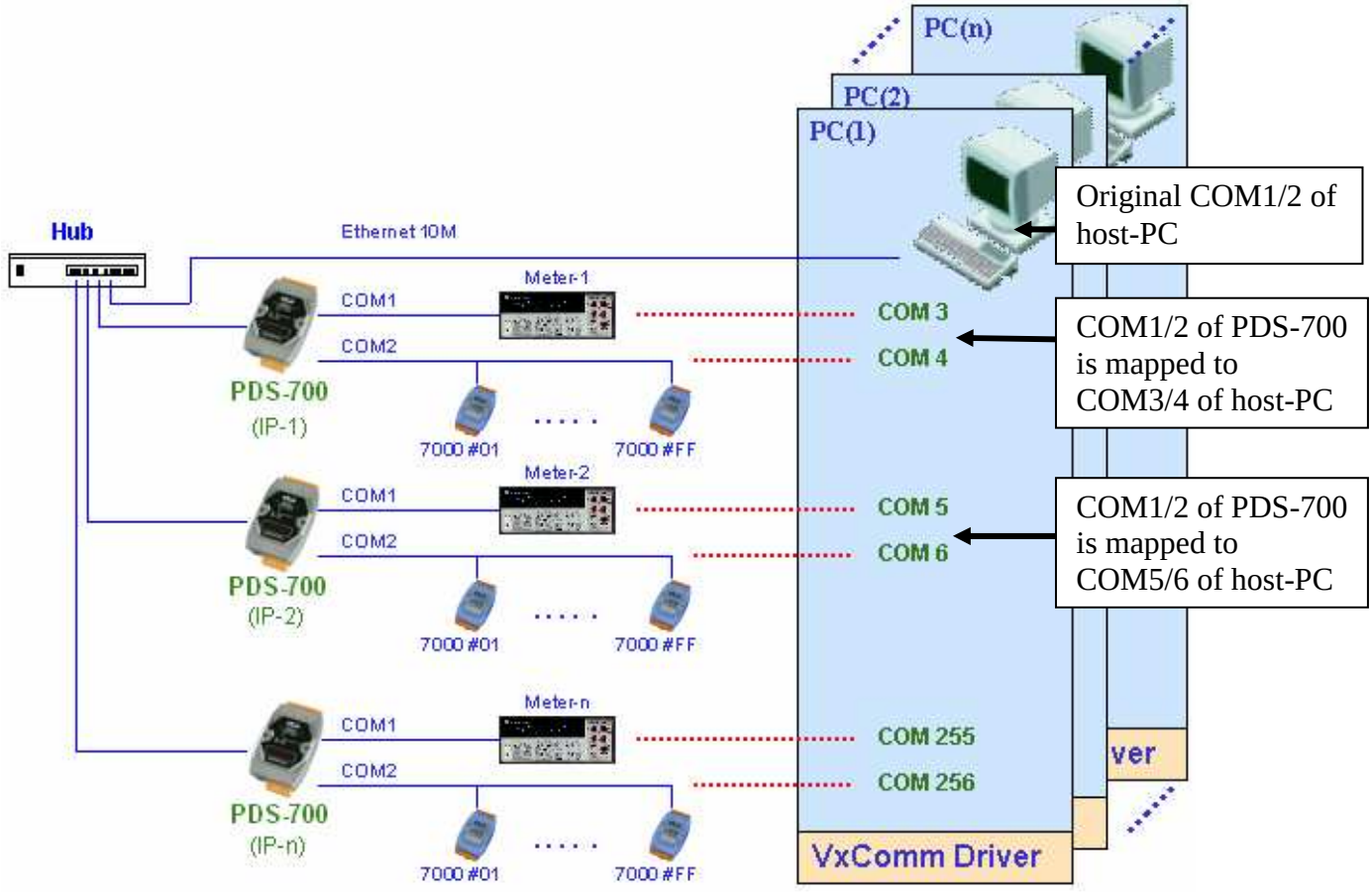
By the Web Server technology, users can set the configuration of PDS-700 with standard browsers (IE, FireFox, Mozilla, etc.). It is very comfortable for users to check the configuration of PDS-700 through Ethernet without installing external software tools; it also reduces user's learning anxiety.



2. Typical Applications of PDS-700

2.1 RS-232/485/422 Devices Networking
--- Using Virtual COM Technology

The PDS-700 series is designed for linking RS-232/485/422 devices to Ethernet network. With the help of VxComm utility, the built-in COM port of PDS-700 can be virtualized to standard COM port of host-PC as follows:



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

2.2 Ethernet I/O Applications

The PDS series provide 2 types of Ethernet I/O solutions as follows:
1. Link to the i-7000 series modules
2. Built-in DIO (if the module support DIO function)

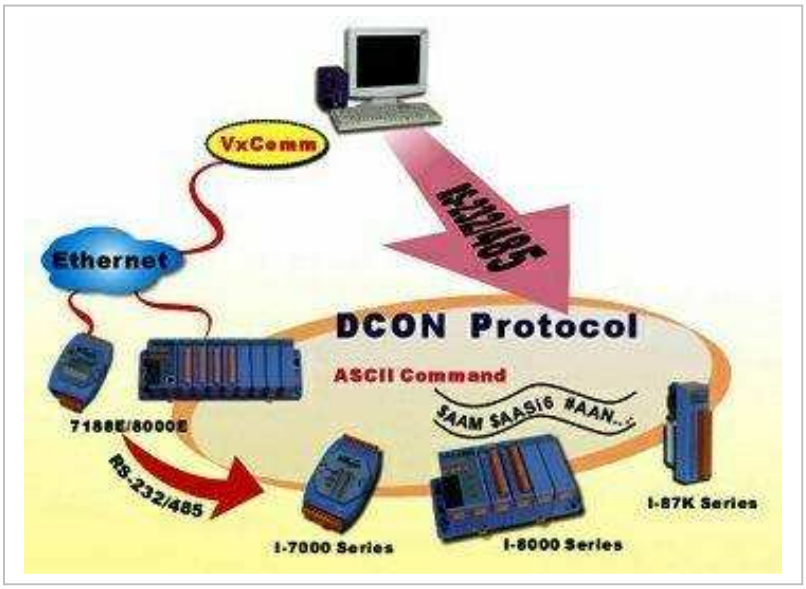
Link to the i-7000 series modules

I-7000 series provide a variety of I/O operations such as D/I, D/O, A/D, D/A, Counter and Frequency Measurement and so on. I-7000 series are originally designed for the RS-485 network, so the COM2 of PDS-700 can be used to link to i-7000 modules.

Using VxComm technology, programs support serial devices in PC can be upgraded from RS-485 network to Ethernet without any program modification. Refer to Sec 2.1 for more information.

Built-in DIO

DCON protocol is a request/ reply communication protocol; it defines a simple ASCII format protocol, like $AAN, $AASi6, #AAN, .., etc. for accessing the PDS-700 and I-7000/8000/87K series I/O modules .



The DCON protocol command sets for PDS-700 are introduced in Sec. 7 for accessing the built-in I/O through the COM port virtualized by Port I/O of PDS-700 in VxComm Utility.

2.3 Link i-7000 Series Modules to Ethernet

I-7000 family is originally designed for RS-485 network; they are very robust and work well under the harsh environments of industry.

The PDS-700 upgrades i-7000 modules to Ethernet solutions. Linking i-7000 modules to Ethernet combines the advantages of RS-485 and Ethernet and enlarges the 485 applications to whole the world.

The VxComm approach provides a MS-COMM-compatible interface. Therefore, the old programs may work without any modification. However, it is also limited by MS-COMM as follows:

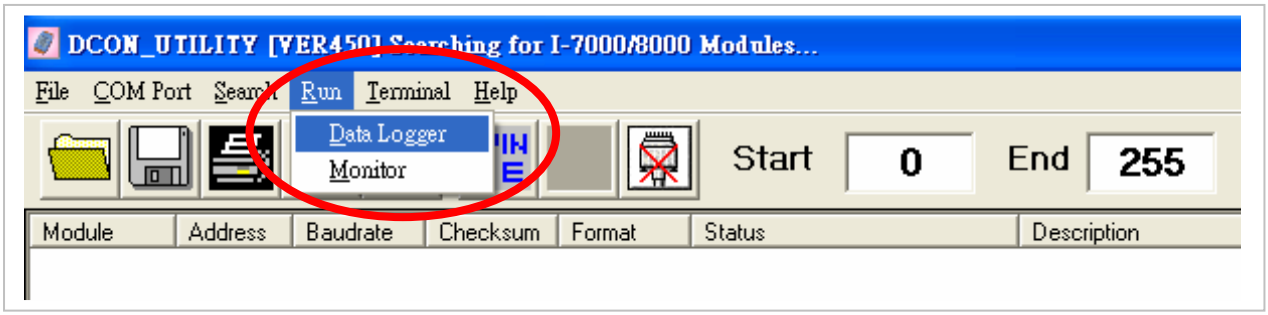
Step 1: Send command to module 1
Step 2: Read response from module 1

Step 3: Send command to module 2
Step 4: Read response from module 2
……
Step N: Send command to module M
Step N+1: Read response from module M
Step N+2: Compute results
Step N+3: Go to step 1 for next loop

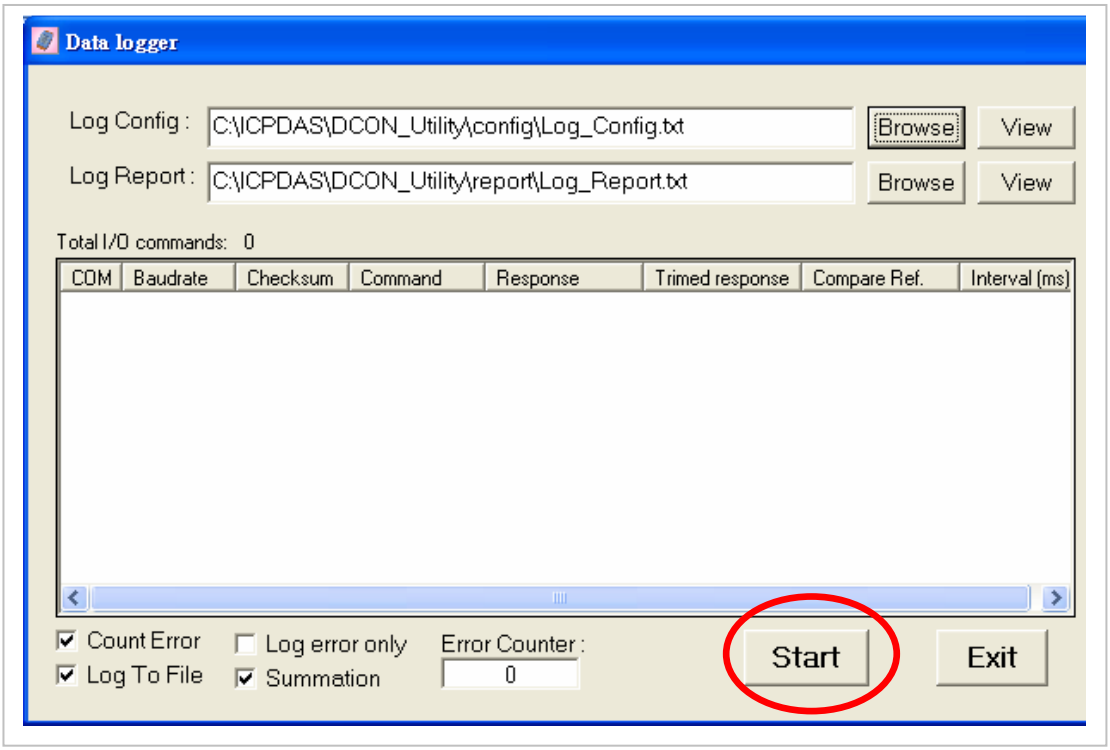
2.4 Configurable Ethernet Data Logger

Using the VxComm driver, PDS-700+7000 modules can be virtualized to become COM port+7000 modules on host-PC, and then the Data Logger in DCON Utility could be used to allocate data from Ethernet. Users can analyze the signal data coming from Ethernet network by Excel without writing any programs

1: DCON utility supports log function as follows:



2: Users can configure the system connection as follows and press "Start" button to start log data.



3: Use Excel to read the log data as follows:

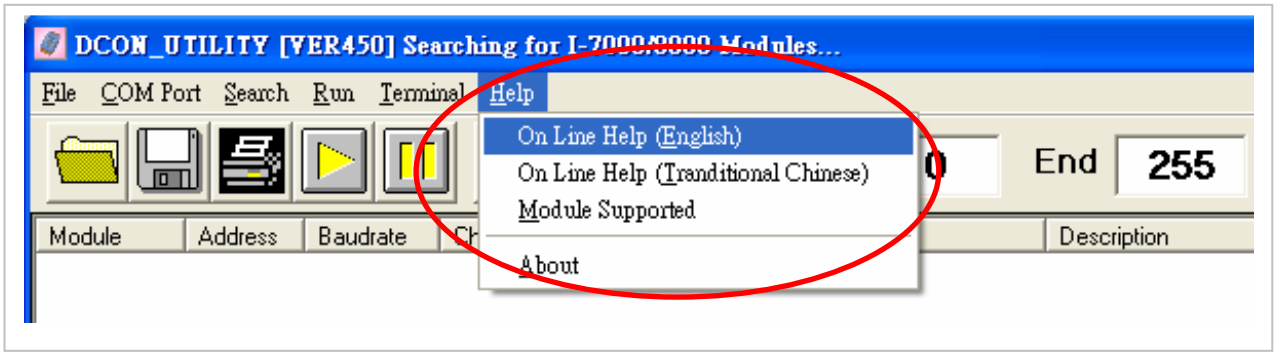
Microsoft Excel - report.log

File Edit View Insert Format Tools Data Window Help

A1 = Start log at

	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		

With the help of **VxComm technology**, the **log functions of 7000 utility** and **Excel**, users can analyze the signal data coming from Ethernet network without writing any programs. Refer to the on-line help of the DCON utility for more information about log function as follows:



3. Hardware information

3.1 Features

- CPU: 80186-80MHz
- SRAM: 512K bytes
- Flash ROM: 512K bytes
- Built-in EEPROM (16K B)
- Built-in COM port: COM1=RS-232, COM2=RS-485
- Built-in watchdog timer for harsh environments
- Built-in power protection circuit
- Built-in RS-485 network protection circuit for RS-485 port
- Built-in self-tuner ASIC controller on RS-485 port
- Program download from PC
- Built-in 5-digit LED display interface (only for D version)
- Ethernet Port: 10/100M Base-TX
- Built-in OS: MiniOS7 of ICP DAS
- ODM wanted

3.2 Specifications

System
- Module name: PDS-700 series
- CPU: 80186-80MHz
- SRAM: 512K bytes
- FLASH ROM: 512K bytes, Erase unit is one sector (64K bytes); 100,000 erase/write cycles.
- COM port: COM1=RS-232, COM2=RS-485
- Program download from COM1

EEPROM
- 16K bytes
- Data retention: 40 years
- 1,000,000 erase/write cycles

Flash Memory
- 512K bytes
- Erase unit is one sector(64K bytes)
- 100,000 erase/write cycles

COM1
- RS-232: TXD,RXD,RTS,CTS,GND; Non-isolation
- Communication speed: 115200 max.
- Program download port

COM2
- RS-485: Data+, Data-, self-tuner ASIC inside; Non-isolation
- Communication speed: 115200 max.

Display
- 7-segment LED: 5-digit (for D version)

Ethernet
- 10/100 Base-TX
- Auto-negotiating, Auto_MDIX, LED indicator

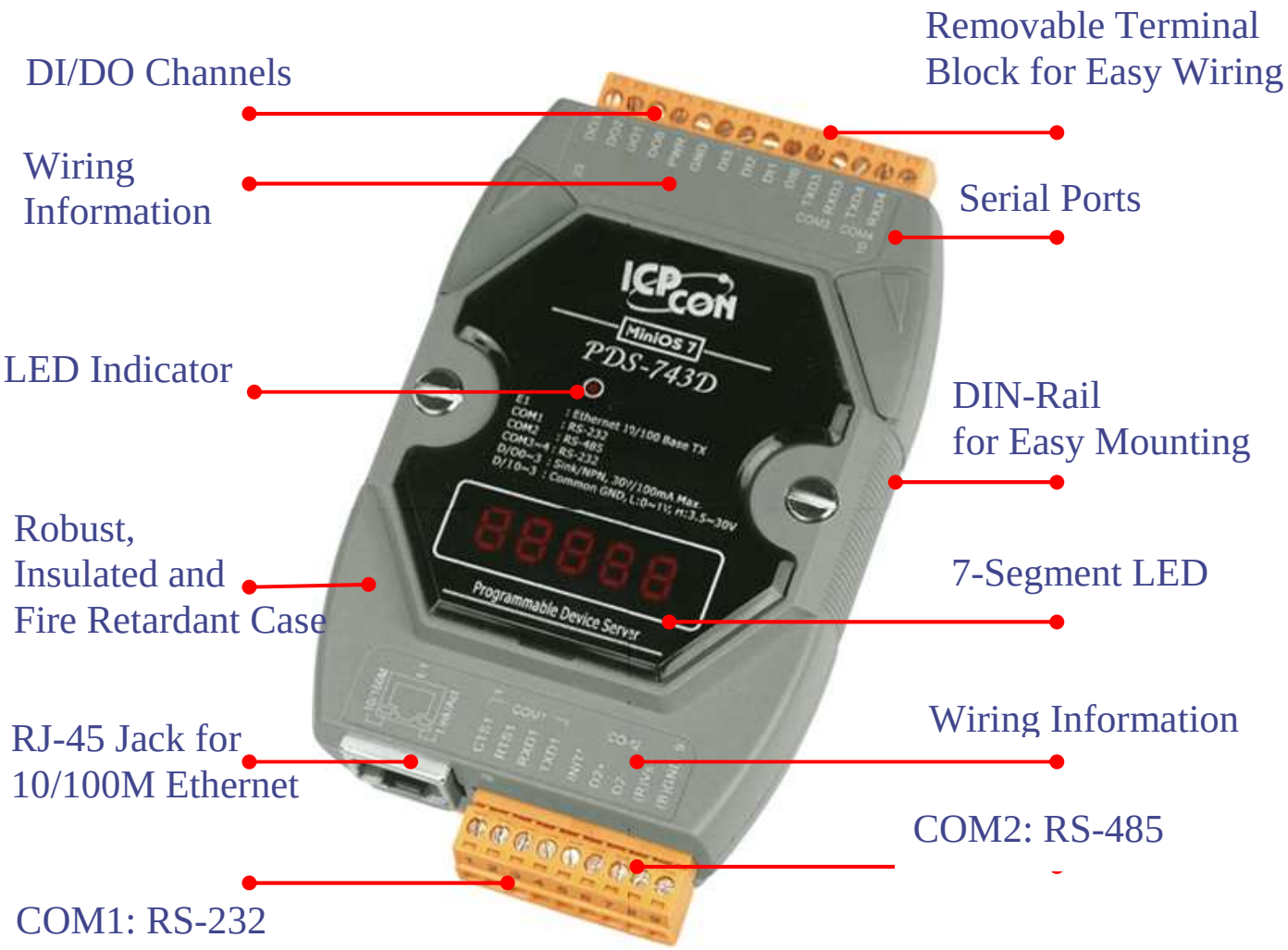
Power

- Power requirements: 10 to 30VDC(non-regulated)
- Power consumption: 2.0W for PDS-700 series
  2.7W for PDS-700D series

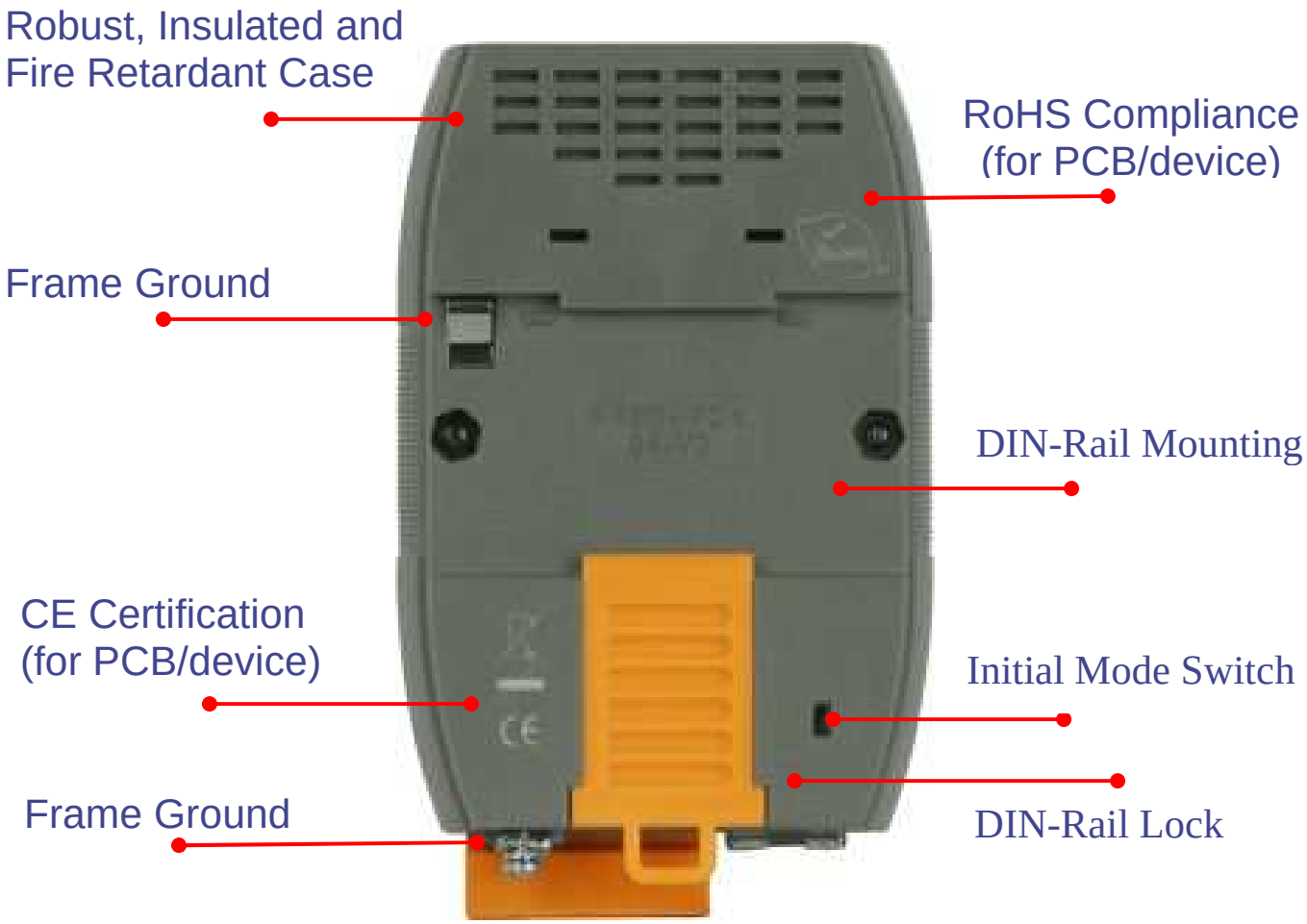
D/I/O:

- D/I: 3.5V ~ 30V max.
- D/O: 100mA, 30V max.

PDS-700 Front View



PDS-700 Back View



3.3 PDS-700 Selection Guide

Model	DI/DO	COM1	COM2	COM3	COM4	COM5	COM6	COM7	COM8
PDS-720 PDS-720D	-	5-wire RS-**232**	2-wire RS-**485**	-	-	-	-	-	-
PDS-721 PDS-721D	6/7	5-wire RS-**232**	2-wire RS-**485**	-	-	-	-	-	-
PDS-732 PDS-732D	4/4	5-wire RS-**232**	2-wire RS-**485**	5-wire RS-**232**	-	-	-	-	-
PDS-734 PDS-734D	4/4	5-wire RS-**232**	2-wire RS-**485**	4-wire RS-**422**	-	-	-	-	-
PDS-742 PDS-742D	-	5-wire RS-**232**	2-wire RS-**485**	5-wire RS-**232**	9-wire RS-**232**	-	-	-	-
PDS-743 PDS-743D	4/4	5-wire RS-**232**	2-wire RS-**485**	3-wire RS-**232**	3-wire RS-**232**	-	-	-	-
PDS-752 PDS-552D	-	5-wire RS-**232**	2-wire RS-**485**	5-wire RS-**232**	5-wire RS-**232**	5-wire RS-**232**	-	-	-
PDS-755 PDS-755D	-	5-wire RS-**232**	2-wire RS-**485**	2-wire RS-**485**	2-wire RS-**485**	2-wire RS-**485**	-	-	-
PDS-762 PDS-762D	1/2	5-wire RS-**232**	2-wire RS-**485**	3-wire RS-**232**	3-wire RS-**232**	3-wire RS-**232**	3-wire RS-**232**	-	-
PDS-782 PDS-782D	-	5-wire RS-**232**	2-wire RS-**485**	3-wire RS-**232**	3-wire RS-**232**	3-wire RS-**232**	3-wire RS-**232**	3-wire RS-**232**	3-wire RS-**232**
PDS-782-25 PDS-782D-25		5-wire RS-**232**	2-wire RS-**485**	3-wire RS-**232**	3-wire RS-**232**	3-wire RS-**232**	3-wire RS-**232**	3-wire RS-**232**	3-wire RS-**232**

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

3.4 Pin Assignments

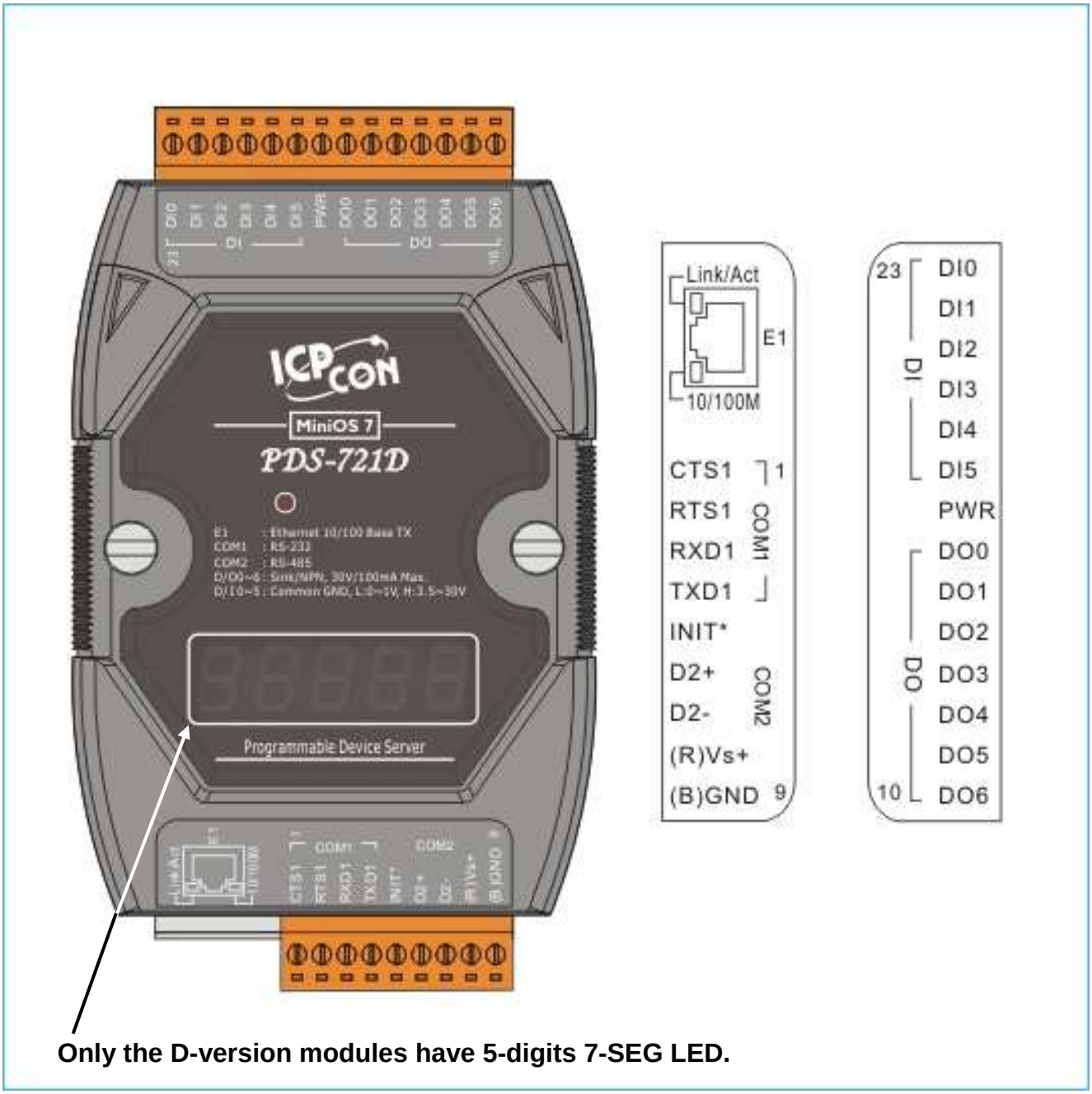
PDS-720/720D Pin Assignments

Only the D-version modules have 5-digits 7-SEG LED.

Pin	Name	Description
1	CTS1	CTS pin of COM1 (RS-232)
2	RTS1	RTS pin of COM1 (RS-232)
3	RXD1	RXD pin of COM1 (RS-232)
4	TXD1	TXD pin of COM1 (RS-232)
5	INIT*	Initial pin (for enable/disable AUTOEXEC.BAT)
6	D2+	Data+ pin of COM2 (RS-485)
7	D2-	Data- pin of COM2 (RS-485)
8	VS+	V+ of power supply (+10 to +30VDC unregulated)
9	GND	GND of power supply (GND of COM1)

E1: 10/100M Base TX

PDS-721/721D Pin Assignments

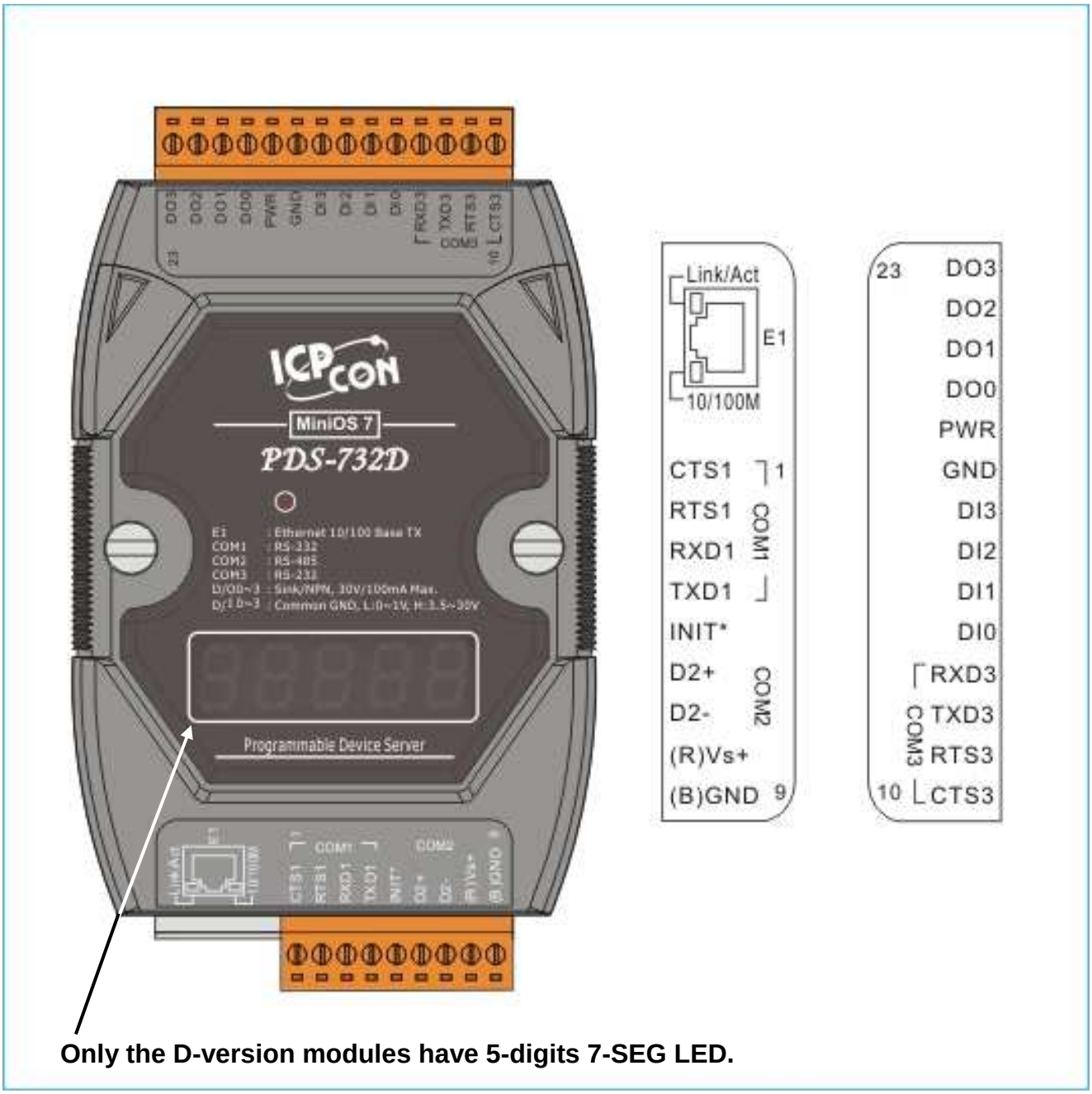


Only the D-version modules have 5-digits 7-SEG LED.

Pin	Name	Description
1	CTS1	CTS pin of COM1 (RS-232)
2	RTS1	RTS pin of COM1 (RS-232)
3	RXD1	RXD pin of COM1(RS-232)
4	TXD1	TXD pin of COM1(RS-232)
5	INIT*	Initial pin (for enable/disable AUTOEXEC.BAT)
6	D2+	Data+ pin of COM2 (RS-485)
7	D2-	Data- pin of COM2 (RS-485)
8	VS+	V+ of power supply (+10 to +30VDC unregulated)
9	GND	GND of power supply (GND of COM1)
10	DO6	Digital Output, 100 mA, 30V max., channel 6
11	DO5	Digital Output, 100 mA, 30V max., channel 5
12	DO4	Digital Output, 100 mA, 30V max., channel 4
13	DO3	Digital Output, 100 mA, 30V max., channel 3
14	DO2	Digital Output, 100 mA, 30V max., channel 2
15	DO1	Digital Output, 100 mA, 30V max., channel 1
16	DO0	Digital Output, 100 mA, 30V max., channel 0
17	PWR	Power Input for Digital Output
18	DI5	Digital Input, 3.5V ~ 30V, channel 5
19	DI4	Digital Input, 3.5V ~ 30V, channel 4
20	DI3	Digital Input, 3.5V ~ 30V, channel 3
21	DI2	Digital Input, 3.5V ~ 30V, channel 2
22	DI1	Digital Input, 3.5V ~ 30V, channel 1
23	DI0	Digital Input, 3.5V ~ 30V, channel 0

E1: 10/100M Base TX

PDS-732/732D Pin Assignments

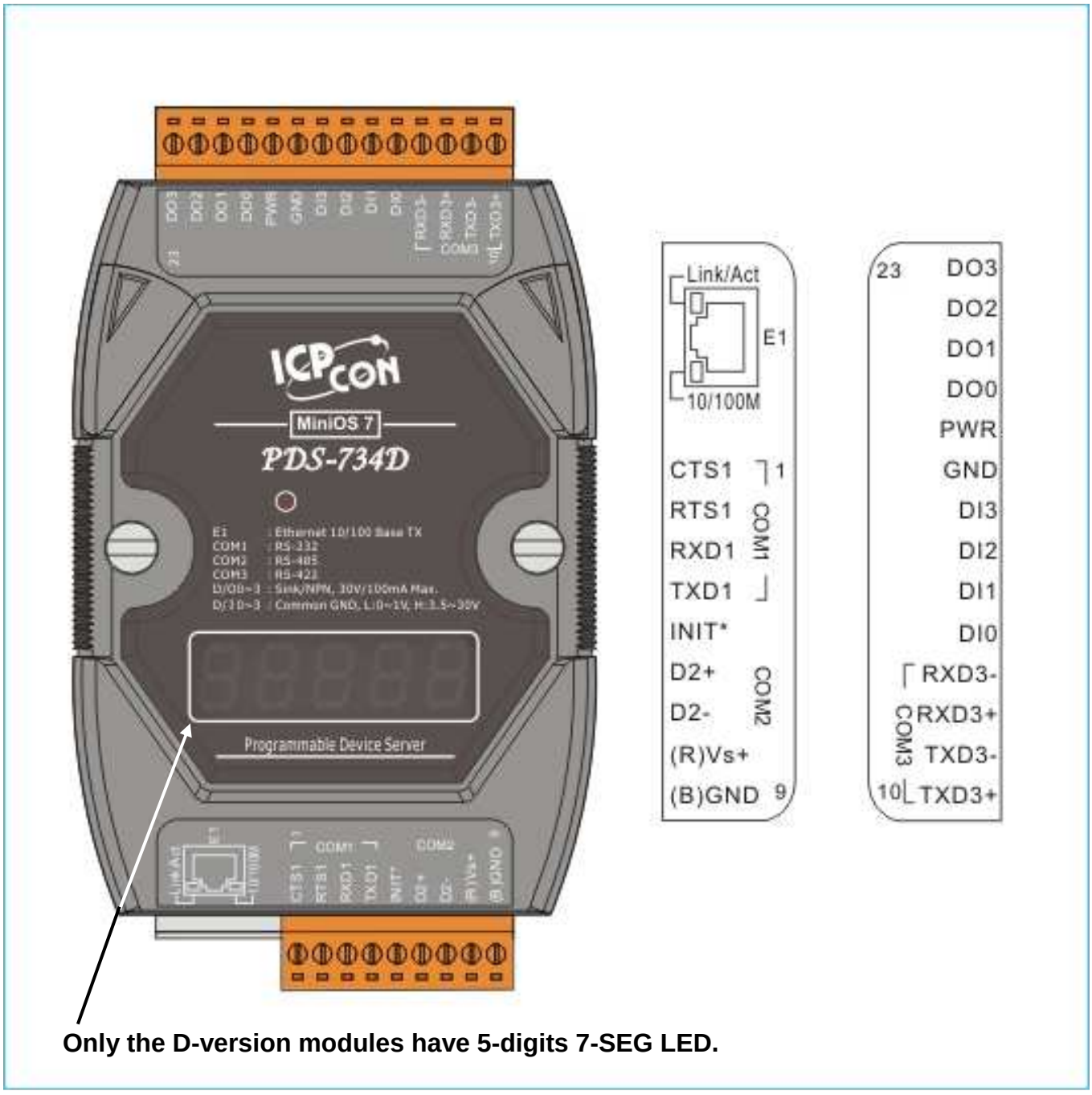


Only the D-version modules have 5-digits 7-SEG LED.

Pin	Name	Description
1	CTS1	CTS pin of COM1 (RS-232)
2	RTS1	RTS pin of COM1 (RS-232)
3	RXD1	RXD pin of COM1 (RS-232)
4	TXD1	TXD pin of COM1 (RS-232)
5	INIT*	Initial pin (for enable/disable AUTOEXEC.BAT)
6	D2+	Data+ pin of COM2 (RS-485)
7	D2-	Data- pin of COM2 (RS-485)
8	VS+	V+ of power supply (+10 to +30VDC unregulated)
9	GND	GND of power supply (GND of COM1)
10	CTS3	CTS pin of COM3 (RS-232)
11	RTS3	RTS pin of COM3 (RS-232)
12	TXD3	TXD pin of COM3 (RS-232)
13	RXD3	RXD pin of COM3 (RS-232)
14	DI0	Digital Input, 3.5V ~ 30V, channel 0
15	DI1	Digital Input, 3.5V ~ 30V, channel 1
16	DI2	Digital Input, 3.5V ~ 30V, channel 2
17	DI3	Digital Input, 3.5V ~ 30V, channel 3
18	GND	GND of Digital Output (GND of COM3)
19	PWR	Power Input for Digital Output
20	DO0	Digital Output, 100 mA, 30V max., channel 0
21	DO1	Digital Output, 100 mA, 30V max., channel 1
22	DO2	Digital Output, 100 mA, 30V max., channel 2
23	DO3	Digital Output, 100 mA, 30V max., channel 3

E1: 10/100M Base TX

PDS-734/734D Pin Assignments

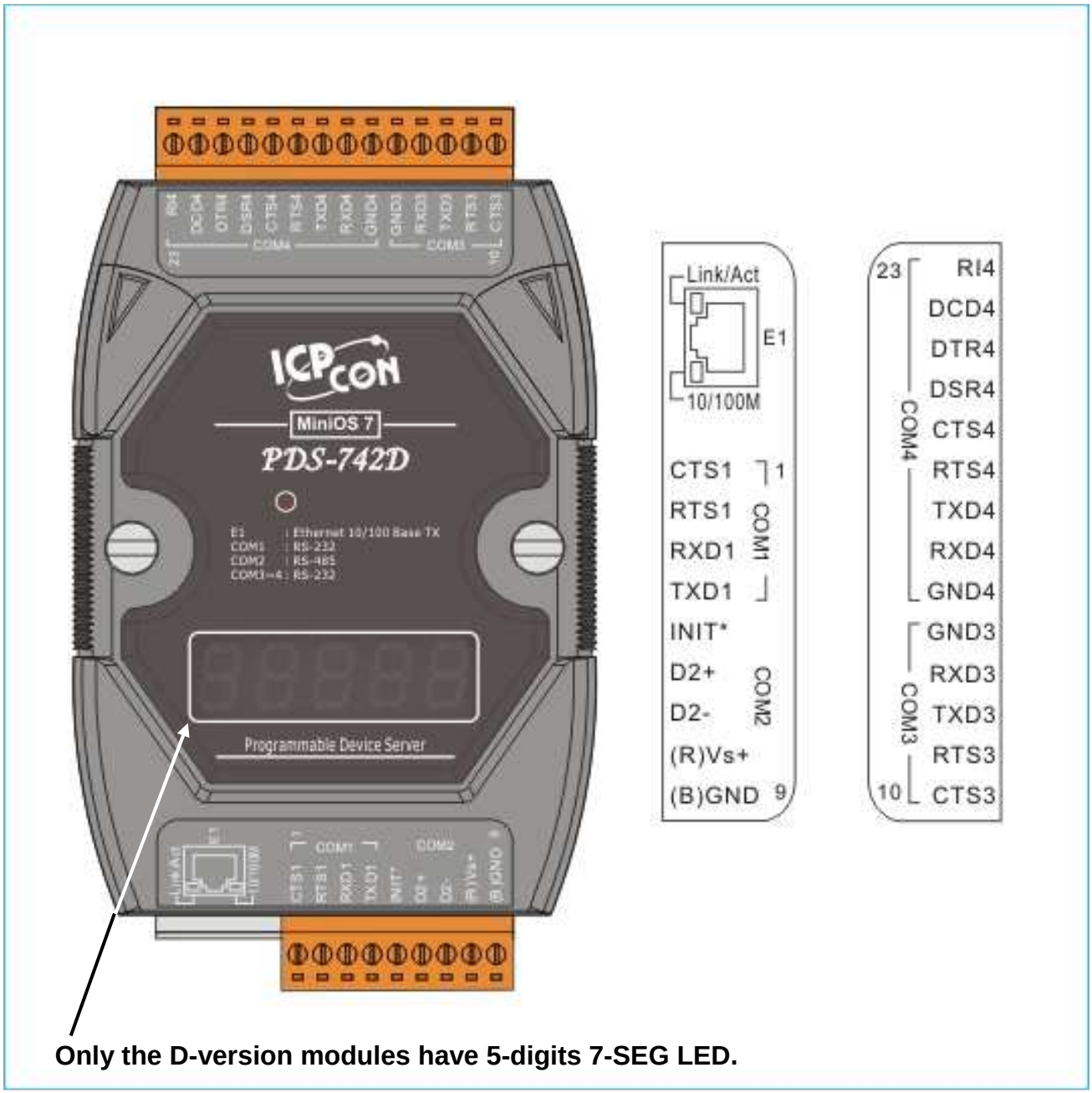


Only the D-version modules have 5-digits 7-SEG LED.

Pin	Name	Description
1	CTS1	CTS pin of COM1 (RS-232)
2	RTS1	RTS pin of COM1 (RS-232)
3	RXD1	RXD pin of COM1 (RS-232)
4	TXD1	TXD pin of COM1 (RS-232)
5	INIT*	Initial pin (for enable/disable AUTOEXEC.BAT)
6	D2+	Data+ pin of COM2 (RS-485)
7	D2-	Data- pin of COM2 (RS-485)
8	VS+	V+ of power supply (+10 to +30VDC unregulated)
9	GND	GND of power supply (GND of COM1)
10	TXD3+	TXD+ pin of COM3 (RS-422/RS-485)
11	TXD3-	TXD- pin of COM3 (RS-422/RS-485)
12	RXD3+	RXD+ pin of COM3 (RS-422)
13	RXD3-	RXD- pin of COM3 (RS-422)
14	DI0	Digital Input, 3.5V ~ 30V, channel 0
15	DI1	Digital Input, 3.5V ~ 30V, channel 1
16	DI2	Digital Input, 3.5V ~ 30V, channel 2
17	DI3	Digital Input, 3.5V ~ 30V, channel 3
18	GND	GND of Digital Output
19	PWR	Power Input for Digital Output
20	DO0	Digital Output, 100 mA, 30V max., channel 0
21	DO1	Digital Output, 100 mA, 30V max., channel 1
22	DO2	Digital Output, 100 mA, 30V max., channel 2
23	DO3	Digital Output, 100 mA, 30V max., channel 3

E1: 10/100M Base TX

PDS-742/742D Pin Assignments

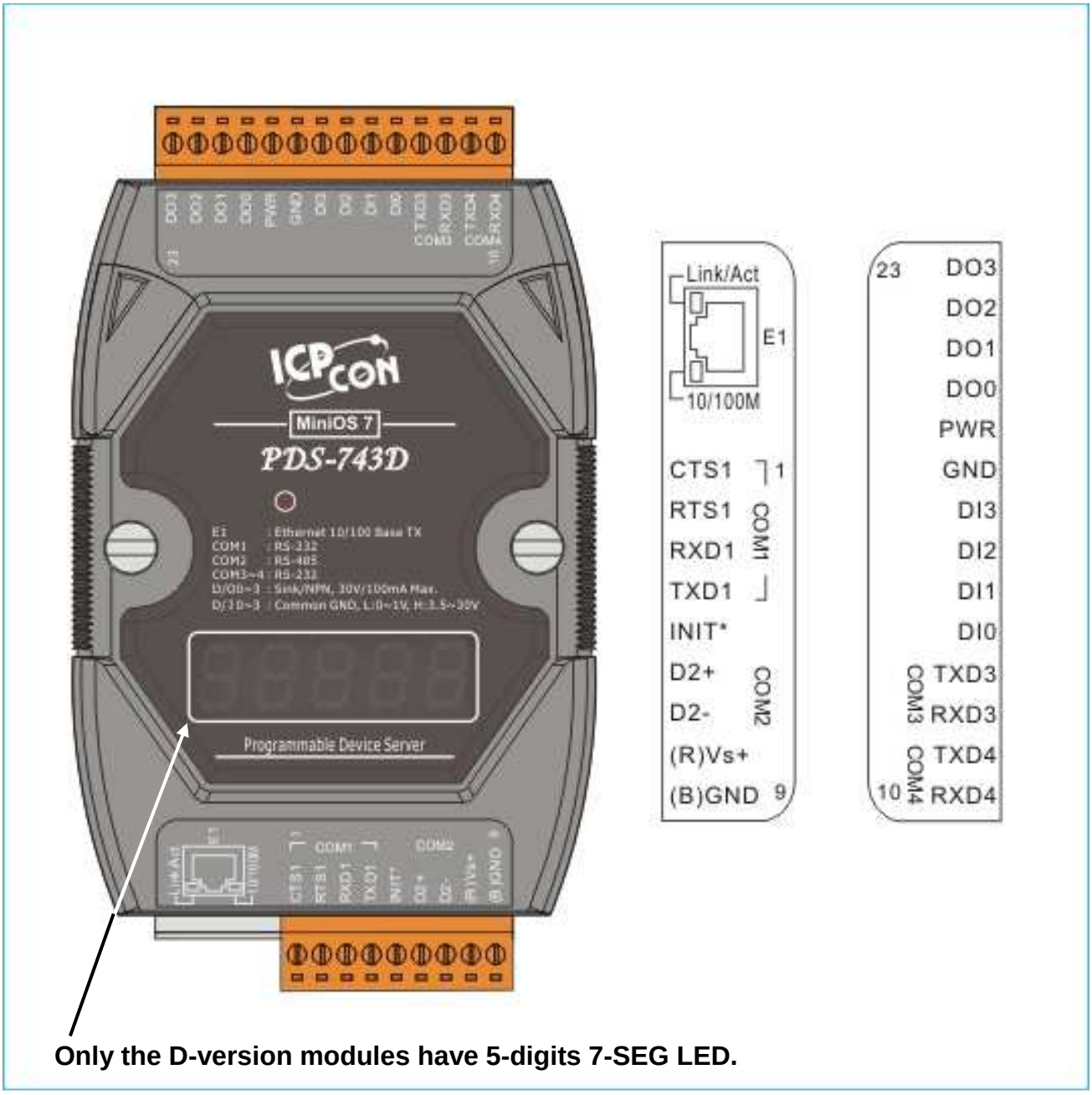


Only the D-version modules have 5-digits 7-SEG LED.

Pin	Name	Description
1	CTS1	CTS pin of COM1 (RS-232)
2	RTS1	RTS pin of COM1 (RS-232)
3	RXD1	RXD pin of COM1 (RS-232)
4	TXD1	TXD pin of COM1 (RS-232)
5	INIT*	Initial pin(for enable/disable AUTOEXEC.BAT)
6	D2+	Data+ pin of COM2 (RS-485)
7	D2-	Data- pin of COM2 (RS-485)
8	VS+	V+ of power supply (+10 to +30VDC unregulated)
9	GND	GND of power supply (GND of COM1)
10	CTS3	CTS pin of COM3 (RS-232)
11	RTS3	RTS pin of COM3 (RS-232)
12	TXD3	TXD pin of COM3 (RS-232)
13	RXD3	RXD pin of COM3 (RS-232)
14	GND3	GND pin of COM3 (RS-232)
15	GND4	GND pin of COM4 (RS-232)
16	RXD4	RXD pin of COM4 (RS-232)
17	TXD4	TXD pin of COM4 (RS-232)
18	RTS4	RTS pin of COM4 (RS-232)
19	CTS4	CTS pin of COM4 (RS-232)
20	DSR4	DSR pin of COM4 (RS-232)
21	DTR4	DTR pin of COM4 (RS-232)
22	DCD4	DCD pin of COM4 (RS-232)
23	RI4	RI pin of COM4 (RS-232)

E1: 10/100M Base TX

PDS-743/743D Pin Assignments

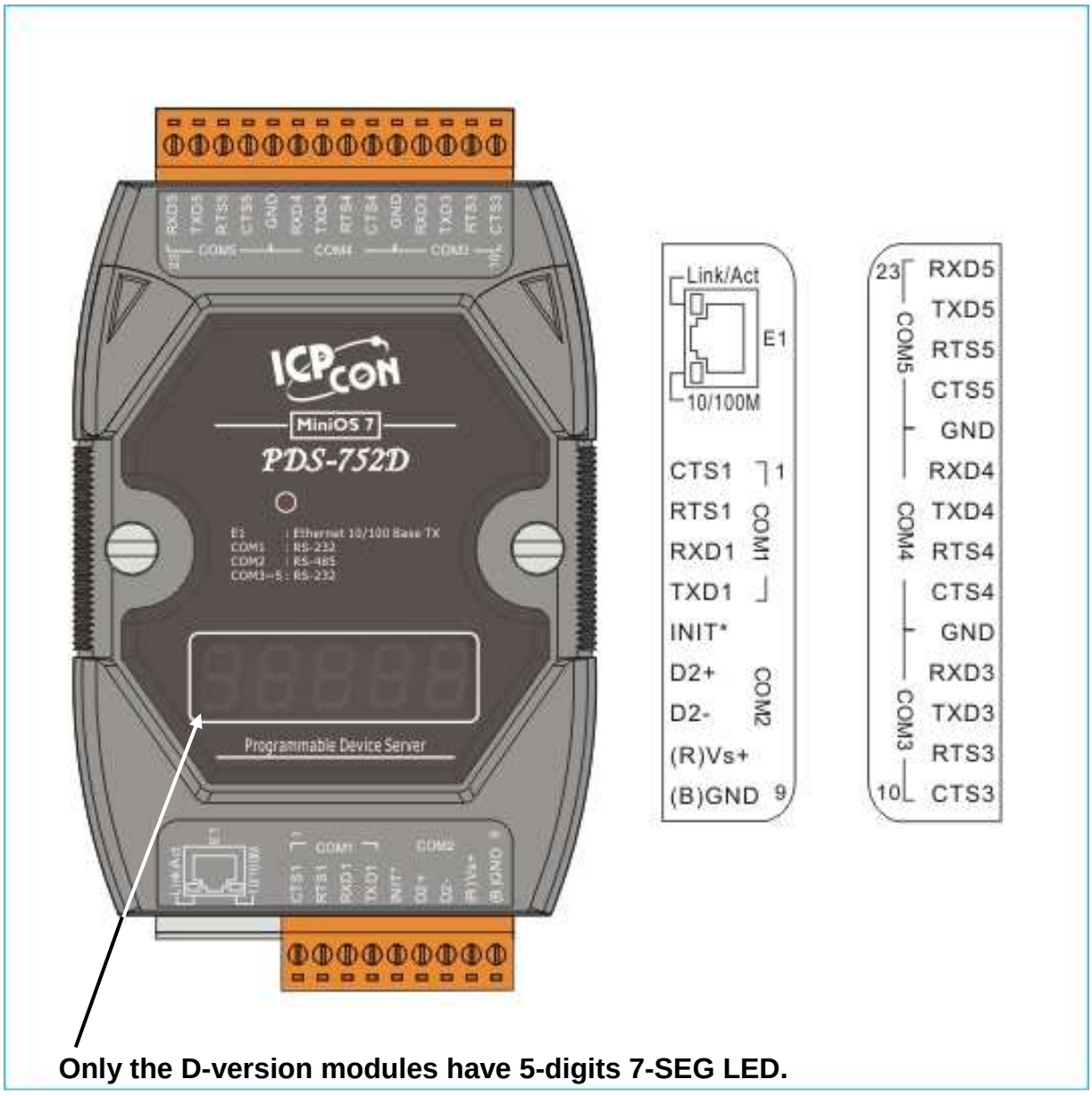


Only the D-version modules have 5-digits 7-SEG LED.

Pin	Name	Description
1	CTS1	CTS pin of COM1 (RS-232)
2	RTS1	RTS pin of COM1 (RS-232)
3	RXD1	RXD pin of COM1 (RS-232)
4	TXD1	TXD pin of COM1 (RS-232)
5	INIT*	Initial pin (for enable/disable AUTOEXEC.BAT)
6	D2+	Data+ pin of COM2 (RS-485)
7	D2-	Data- pin of COM2 (RS-485)
8	VS+	V+ of power supply (+10 to +30VDC unregulated)
9	GND	GND of power supply (GND of COM1)
10	RXD4	RXD pin of COM4 (RS-232)
11	TXD4	TXD pin of COM4 (RS-232)
12	RXD3	RXD pin of COM3 (RS-232)
13	TXD3	TXD pin of COM3 (RS-232)
14	DI0	Digital Input, 3.5V ~ 30V, channel 0
15	DI1	Digital Input, 3.5V ~ 30V, channel 1
16	DI2	Digital Input, 3.5V ~ 30V, channel 2
17	DI3	Digital Input, 3.5V ~ 30V, channel 3
18	GND	GND of Digital Output (GND of COM3/ COM4)
19	PWR	Power Input for Digital Output
20	DO0	Digital Output, 100 mA, 30V max., channel 0
21	DO1	Digital Output, 100 mA, 30V max., channel 1
22	DO2	Digital Output, 100 mA, 30V max., channel 2
23	DO3	Digital Output, 100 mA, 30V max., channel 3

E1: 10/100M Base TX

PDS-752/752D Pin Assignments

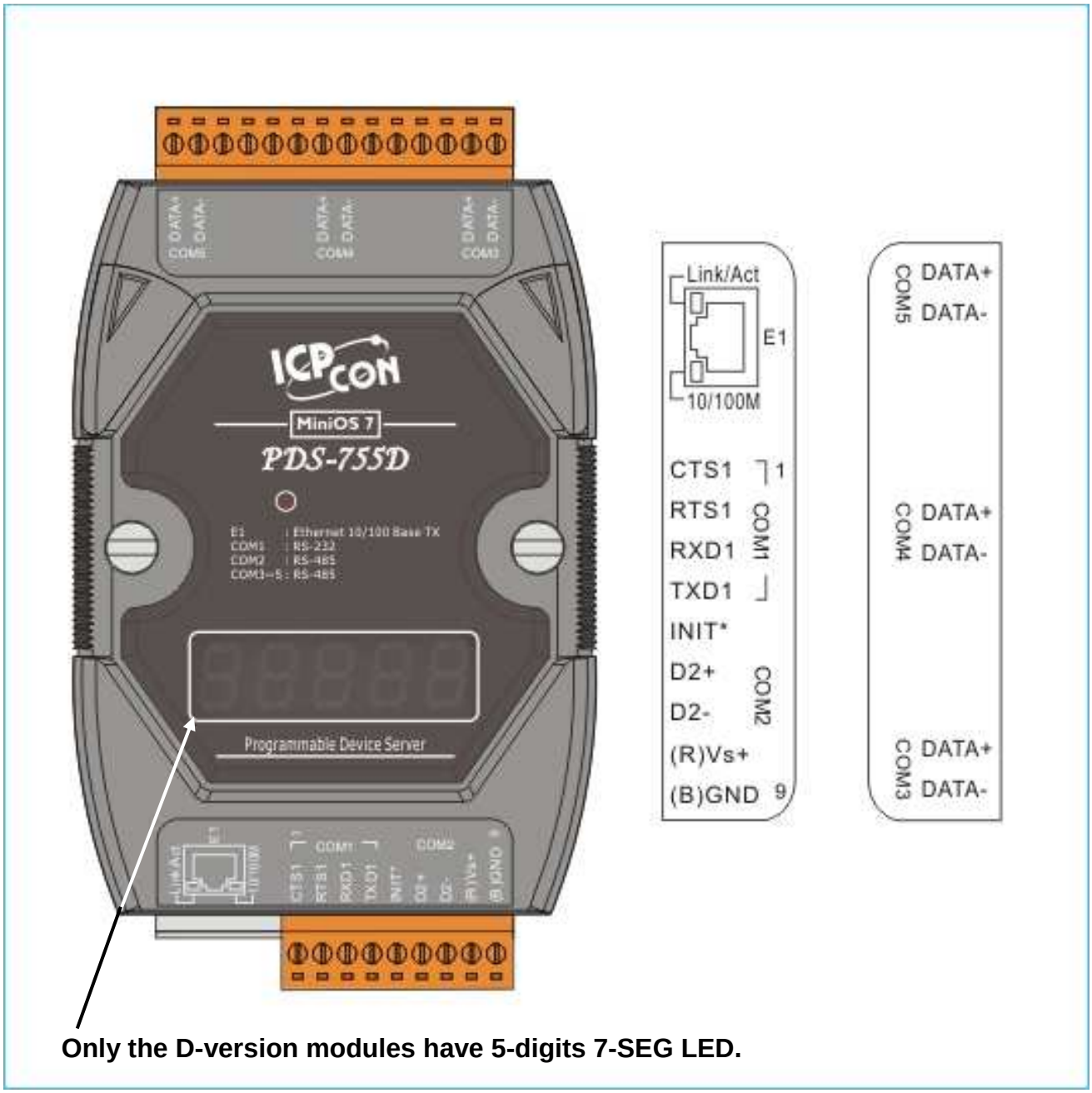


Only the D-version modules have 5-digits 7-SEG LED.

Pin	Name	Description
1	CTS1	CTS pin of COM1 (RS-232)
2	RTS1	RTS pin of COM1 (RS-232)
3	RXD1	RXD pin of COM1(RS-232)
4	TXD1	TXD pin of COM1(RS-232)
5	INIT*	Initial pin (for enable/disable AUTOEXEC.BAT)
6	D2+	Data+ pin of COM2 (RS-485)
7	D2-	Data- pin of COM2 (RS-485)
8	VS+	V+ of power supply (+10 to +30VDC unregulated)
9	GND	GND of power supply (GND of COM1)
10	CTS3	CTS pin of COM3 (RS-232)
11	RTS3	RTS pin of COM3 (RS-232)
12	TXD3	TXD pin of COM3 (RS-232)
13	RXD3	RXD pin of COM3 (RS-232)
14	GND	GND pin of COM3/ COM4 (RS-232)
15	CTS4	CTS pin of COM4 (RS-232)
16	RTS4	RTS pin of COM4 (RS-232)
17	TXD4	TXD pin of COM4 (RS-232)
18	RXD4	RXD pin of COM4 (RS-232)
19	GND	GND pin of COM4/ COM5 (RS-232)
20	CTS5	CTS pin of COM5 (RS-232)
21	RTS5	RTS pin of COM5 (RS-232)
22	TXD5	TXD pin of COM5 (RS-232)
23	RXD5	RXD pin of COM5 (RS-232)

E1: 10/100M Base TX

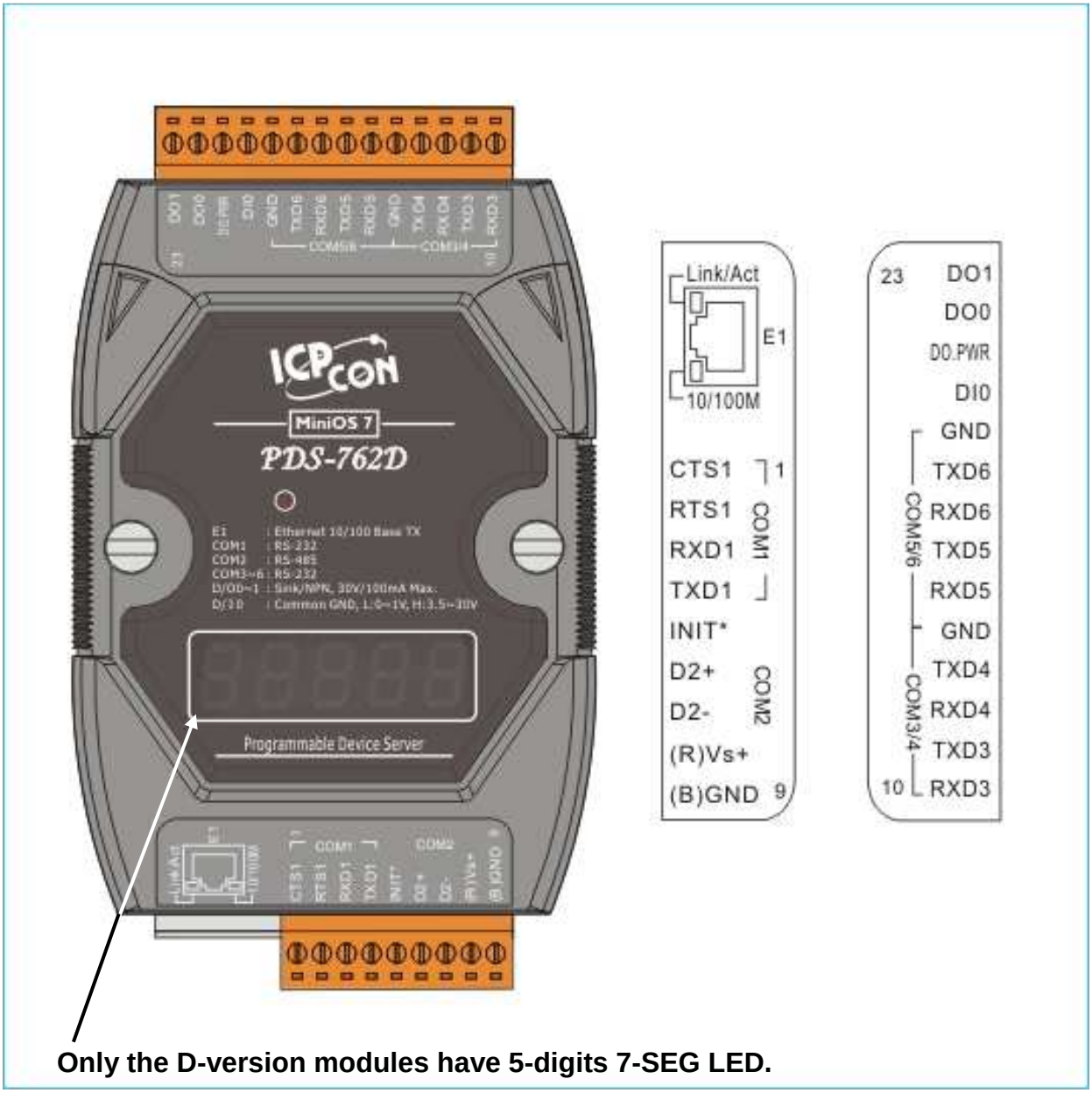
PDS-755/755D Pin Assignments



Pin	Name	Description
1	CTS1	CTS pin of COM1 (RS-232)
2	RTS1	RTS pin of COM1 (RS-232)
3	RXD1	RXD pin of COM1(RS-232)
4	TXD1	TXD pin of COM1(RS-232)
5	INIT*	Initial pin (for enable/disable AUTOEXEC.BAT)
6	D2+	Data+ pin of COM2 (RS-485)
7	D2-	Data- pin of COM2 (RS-485)
8	VS+	V+ of power supply (+10 to +30VDC unregulated)
9	GND	GND of power supply (GND of COM1)
10	D3-	Data- pin of COM3 (RS-485)
11	D3+	Data+ pin of COM3 (RS-485)
12	-	N.C.
13	-	N.C.
14	-	N.C.
15	-	N.C.
16	D4-	Data- pin of COM4 (RS-485)
17	D4+	Data+ pin of COM4 (RS-485)
18	-	N.C.
19	-	N.C.
20	-	N.C.
21	-	N.C.
22	D5-	Data- pin of COM5 (RS-485)
23	D5+	Data+ pin of COM5 (RS-485)

E1: 10/100M Base TX

PDS-762/762D Pin Assignments

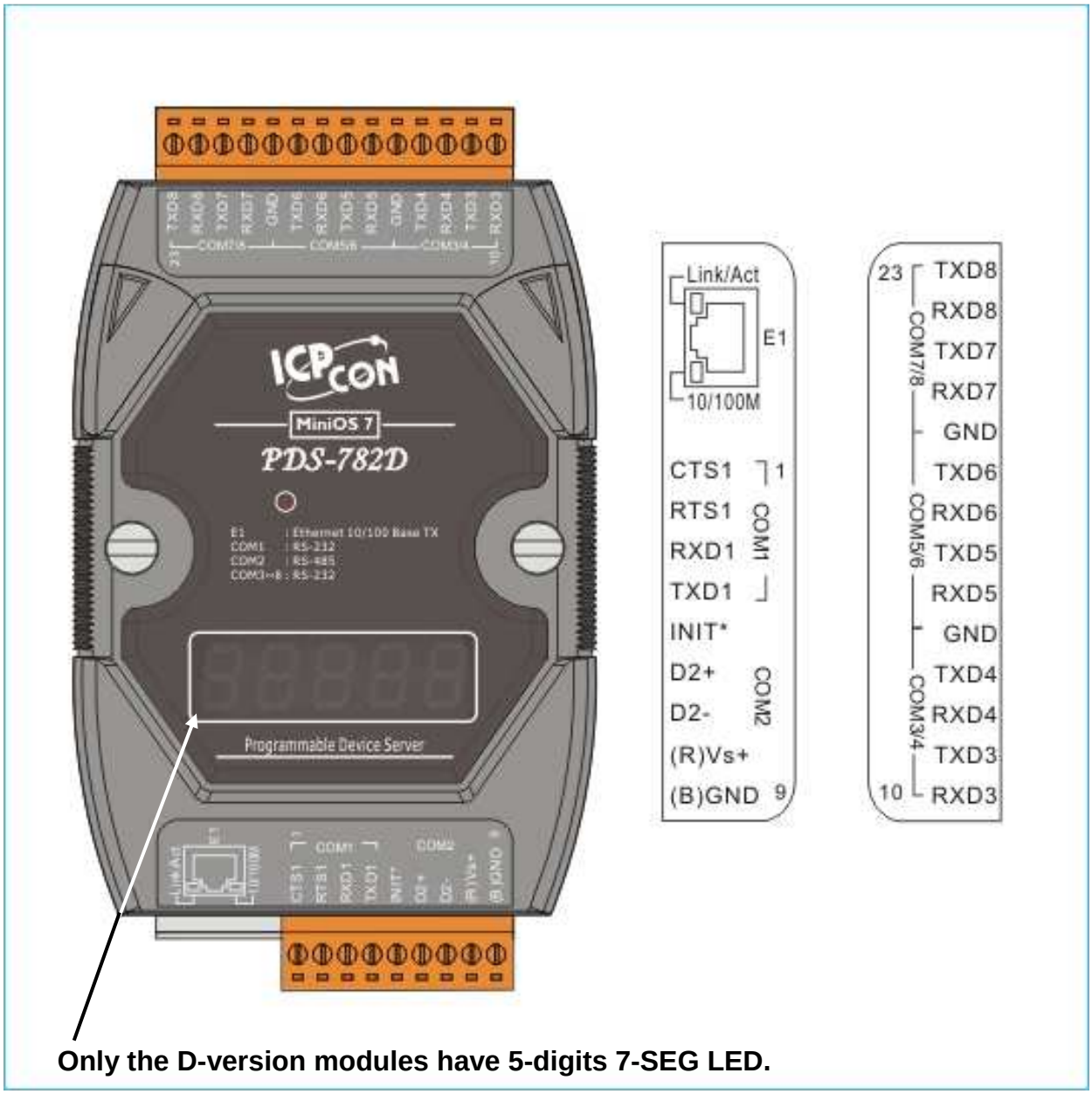


Only the D-version modules have 5-digits 7-SEG LED.

Pin	Name	Description
1	CTS1	CTS pin of COM1 (RS-232)
2	RTS1	RTS pin of COM1 (RS-232)
3	RXD1	RXD pin of COM1 (RS-232)
4	TXD1	TXD pin of COM1 (RS-232)
5	INIT*	Initial pin (for enable/disable AUTOEXEC.BAT)
6	D2+	Data+ pin of COM2 (RS-485)
7	D2-	Data- pin of COM2 (RS-485)
8	VS+	V+ of power supply (+10 to +30VDC unregulated)
9	GND	GND of power supply (GND of COM1)
10	RXD3	RXD pin of COM3 (RS-232)
11	TXD3	TXD pin of COM3 (RS-232)
12	RXD4	RXD pin of COM4 (RS-232)
13	TXD4	TXD pin of COM4 (RS-232)
14	GND	GND of COM3/ COM4
15	RXD5	RXD pin of COM5 (RS-232)
16	TXD5	TXD pin of COM5 (RS-232)
17	RXD6	RXD pin of COM6 (RS-232)
18	TXD6	TXD pin of COM6 (RS-232)
19	GND	GND of COM5/ COM6
20	DI0	Digital Input, 3.5V ~ 30V, channel 0
21	DO.PWR	Power Input for Digital Output
22	DO0	Digital Output, 100 mA, 30V max., channel 0
23	DO1	Digital Output, 100 mA, 30V max., channel 1

E1: 10/100M Base TX

PDS-782/782D Pin Assignments



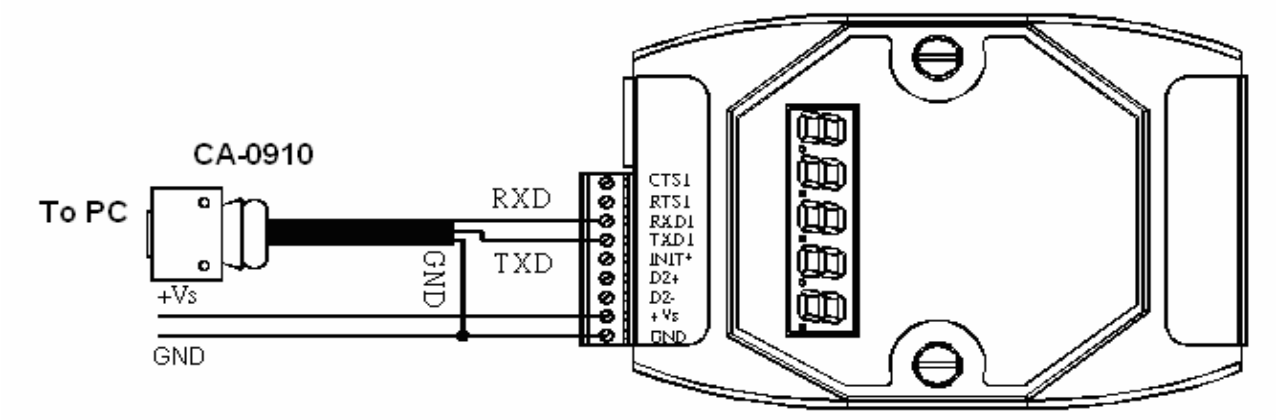
Only the D-version modules have 5-digits 7-SEG LED.

Pin	Name	Description
1	CTS1	CTS pin of COM1 (RS-232)
2	RTS1	RTS pin of COM1 (RS-232)
3	RXD1	RXD pin of COM1 (RS-232)
4	TXD1	TXD pin of COM1 (RS-232)
5	INIT*	Initial pin (for enable/disable AUTOEXEC.BAT)
6	D2+	Data+ pin of COM2 (RS-485)
7	D2-	Data- pin of COM2 (RS-485)
8	VS+	V+ of power supply (+10 to +30VDC unregulated)
9	GND	GND of power supply (GND of COM1)
10	RXD3	RXD pin of COM3 (RS-232)
11	TXD3	TXD pin of COM3 (RS-232)
12	RXD4	RXD pin of COM4 (RS-232)
13	TXD4	TXD pin of COM4 (RS-232)
14	GND	GND of COM3/ COM4/ COM5/ COM6
15	RXD5	RXD pin of COM5 (RS-232)
16	TXD5	TXD pin of COM5 (RS-232)
17	RXD6	RXD pin of COM6 (RS-232)
18	TXD6	TXD pin of COM6 (RS-232)
19	GND	GND of COM5/ COM6/ COM7/ COM8
20	RXD7	RXD pin of COM7 (RS-232)
21	TXD7	TXD pin of COM7 (RS-232)
22	RXD8	RXD pin of COM8 (RS-232)
23	TXD8	TXD pin of COM8 (RS-232)

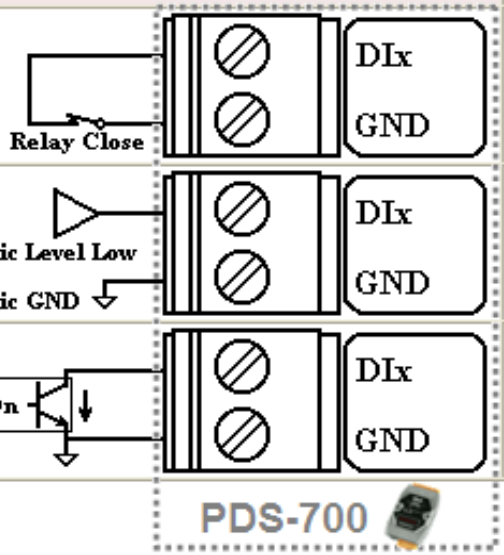
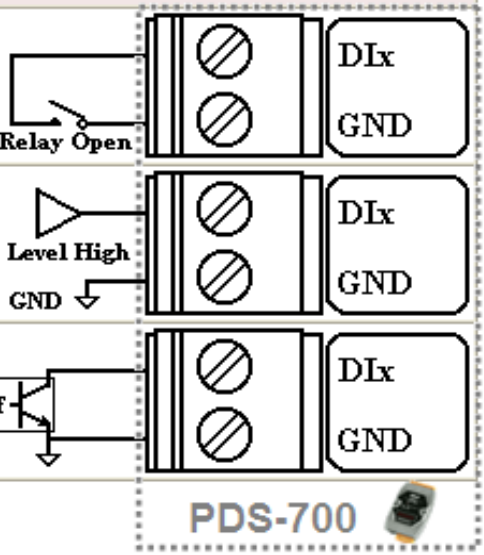
E1: 10/100M Base TX

3.5 Wire Connection

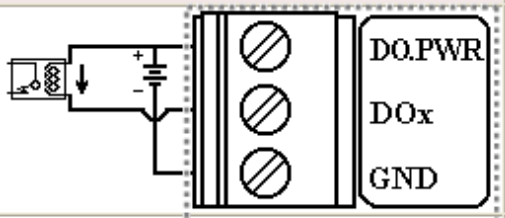
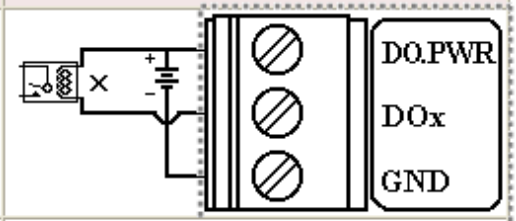
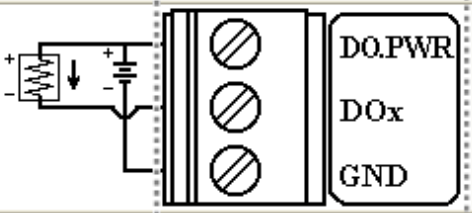
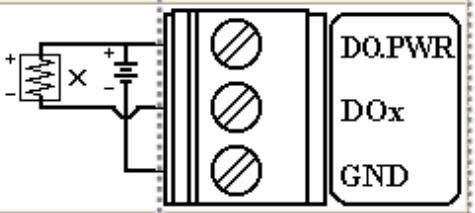
COM1 : The GND-signal of COM1 is shared with pin-9, GND



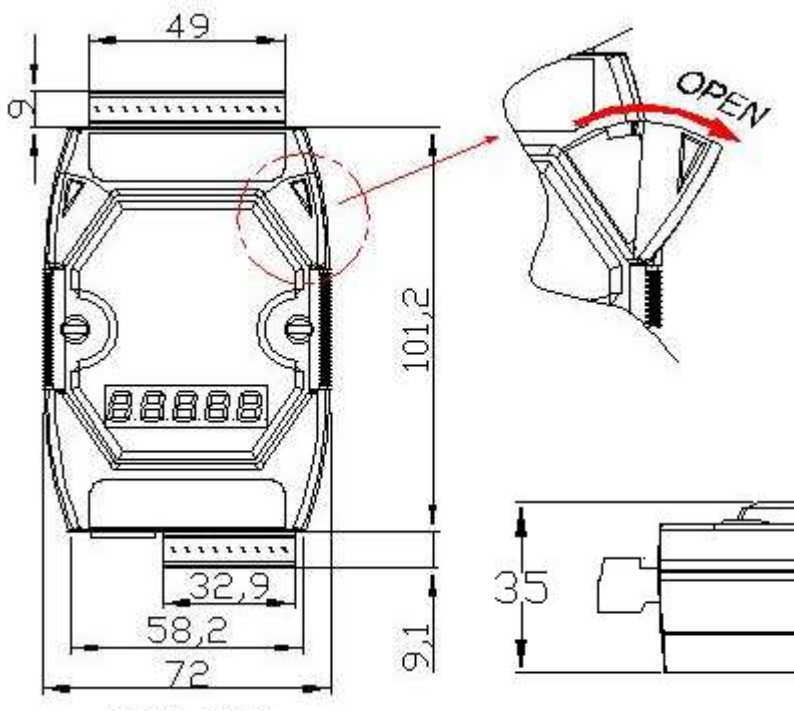
Digital Input

Input Type	DI value as 0	DI value as 1
Relay Contact		
TTL/CMOS Logic		
Open Collector		

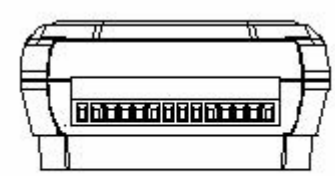
Digital Output

Input Type	ON State DO command as 1	OFF State DO command as 0
Drive Relay		
Resistance Load		
	PDS-700	PDS-700

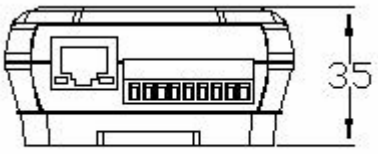
3.6 Dimensions and Mounting



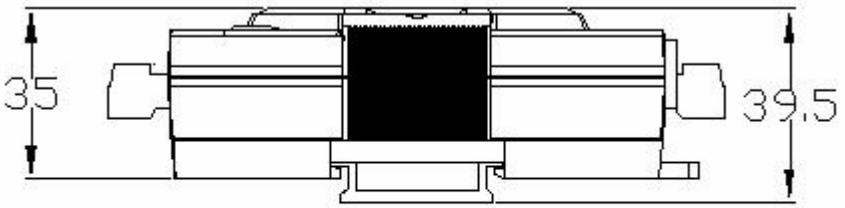
Front View



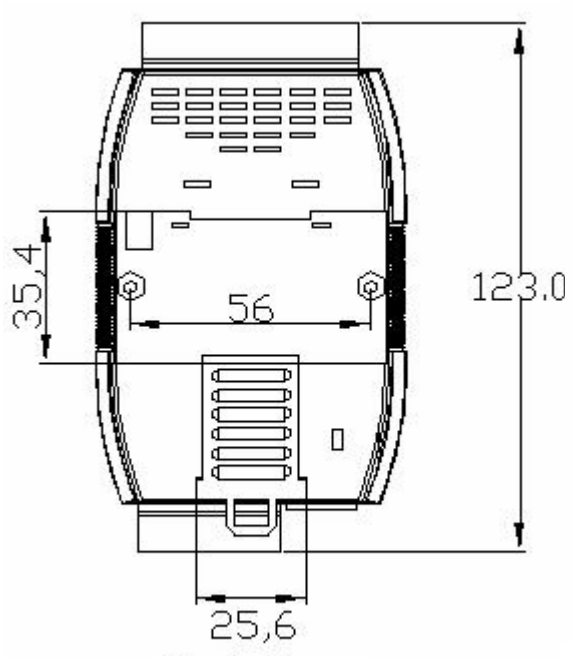
Top View



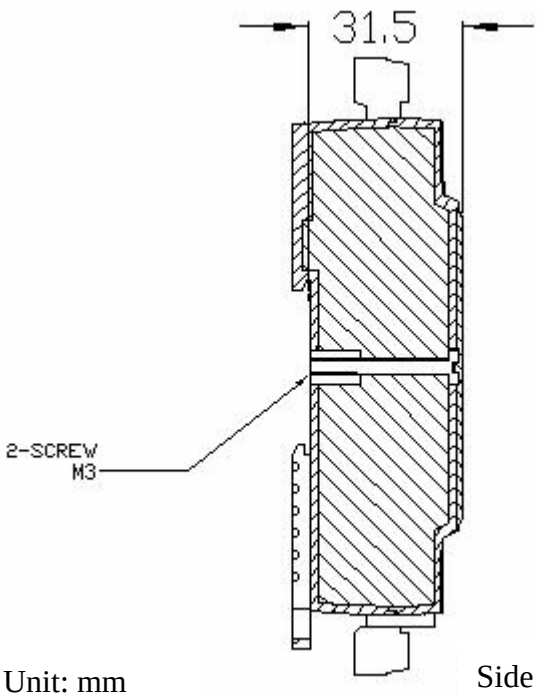
Bottom View



Din-Rail Mounting Bracket

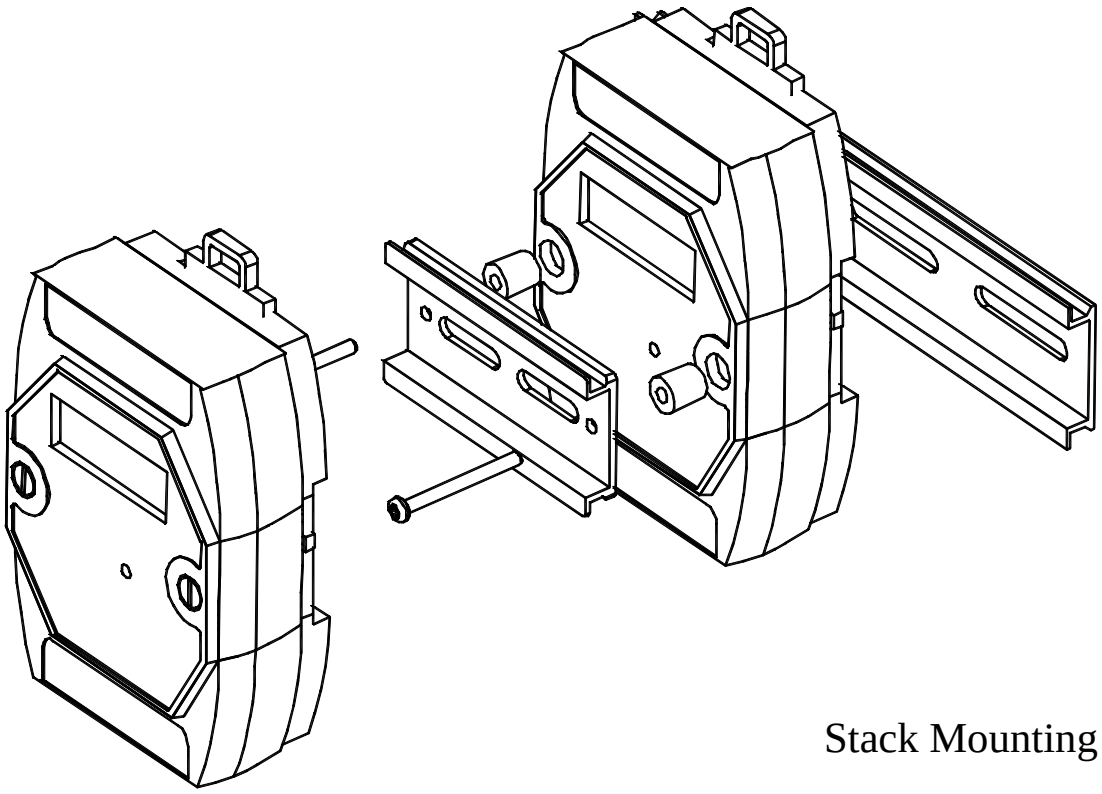


Back View

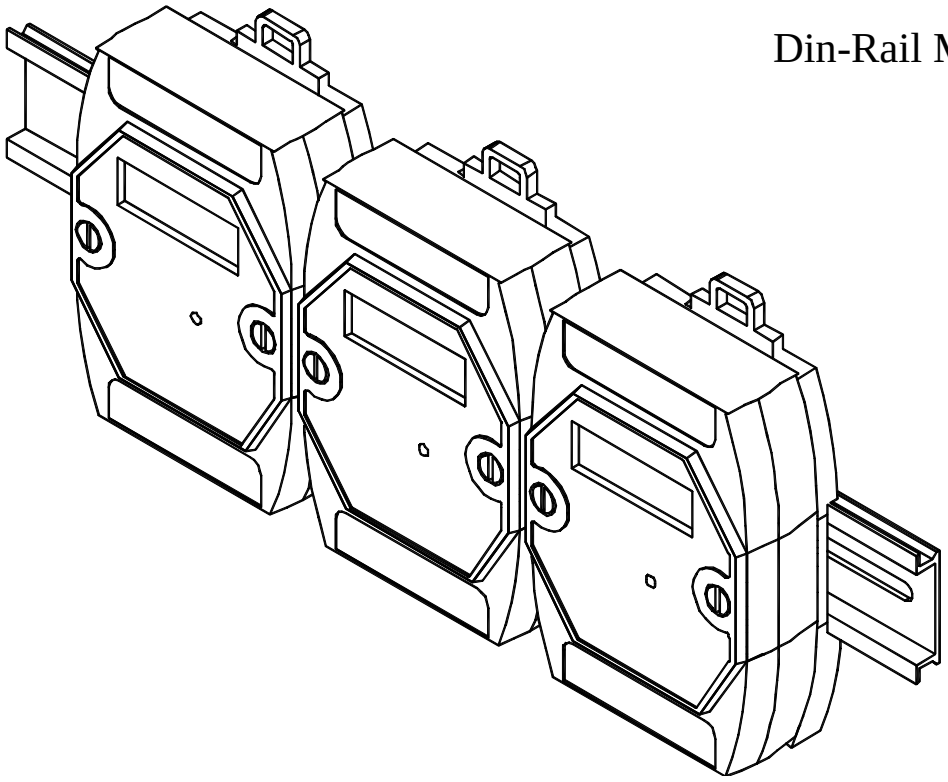


Unit: mm

Side View

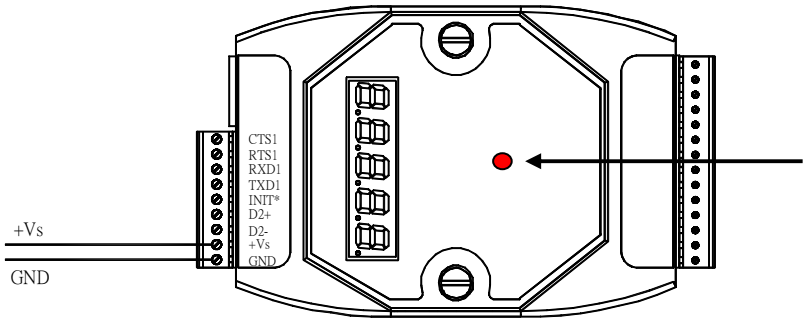


Stack Mounting



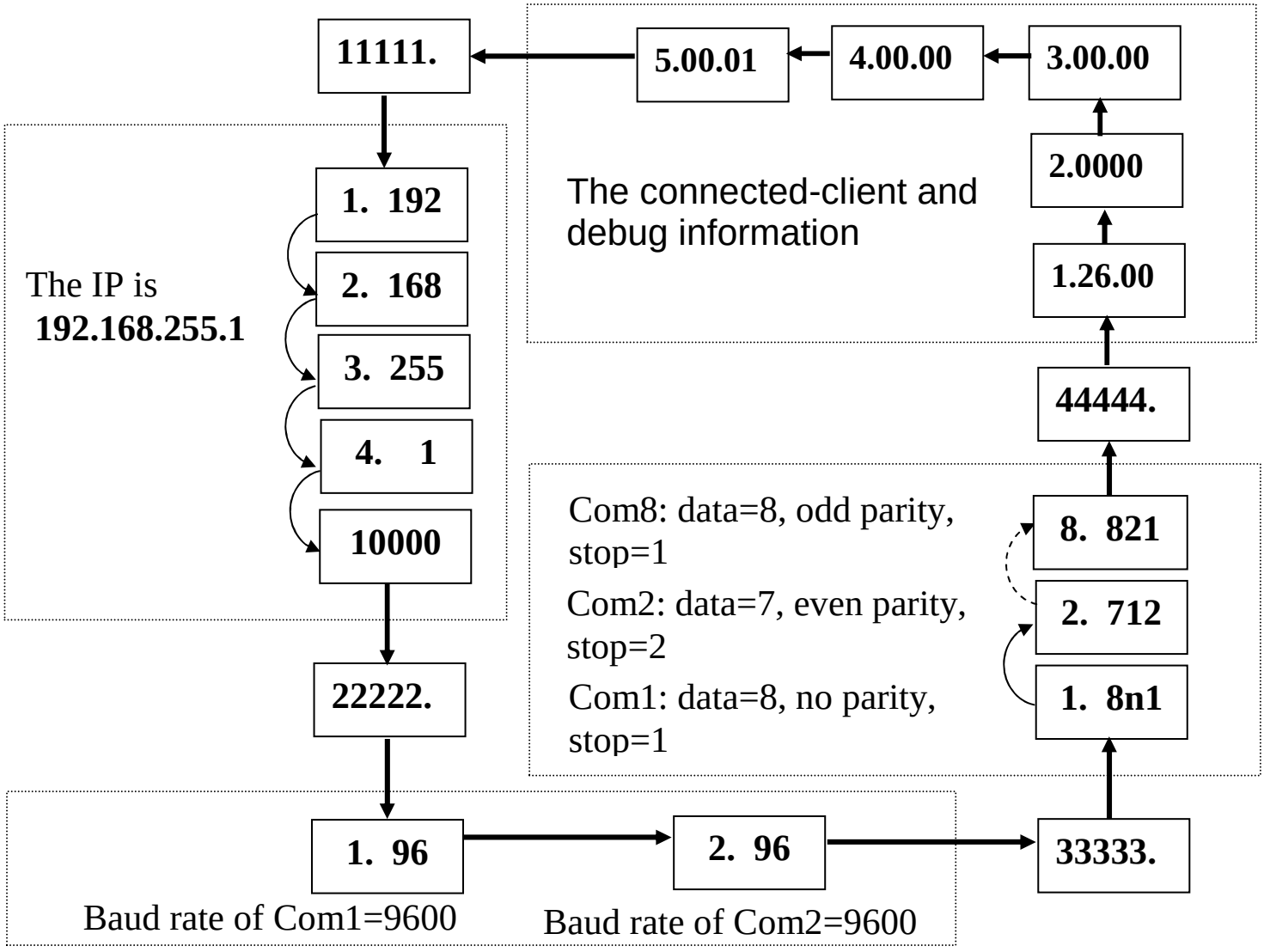
Din-Rail Mounting

3.7 Diagnostics of the PDS-700 series



Indicator LED
VxComm running: **On/Off**
Xserver running: **On/Off**

Step 1: Apply power (+Vs, GND) to the PDS-700, the power could be from +30V to +10V.
Step 2: Checking the 5-digits of the 7-SEG LED will show as follows.
Note: Only the D-version modules have 5-digits 7-SEG LED.



The Important information related to the PDS-700 can be classified as follows:

- Group-ID 11111 :IP information for the PDS-700
- Group-ID 22222: The Baud Rate of all COM ports
- Group-ID 33333: COM port configuration
- Group-ID 44444: Connected-client and debug information of this PDS-700

The format of the PDS-700 IP-information is as follows:

- 5-digit LED Group-ID: 11111
- LED-1: indicator, which can be either 1,2,3 or 4
- LED-2~5: IP address
- TCP command port (Default=10000)
- DHCP setting: disabled(0)/ enabled(1)

The LED will initially show Group-ID first, and then show its IP address as indicated in the above diagram. If the user changes the IP address, the value displayed will change immediately. The default shipping IP = 192.168.255.1 and the LED-display sequence is shown in the above diagram.

The COM port Baud Rate format is follows:

- 5-digit LED Group-ID : 22222
- LED-1: COM port number
- LED-2~5: The Baud rate determined as (Baud Rate/100)

LED-1 displays the COM port number, with LED-2~5 showing its Baud Rate. The Baud Rate = (value shown by LED-2~5) * 100. Therefore, a displayed value of 1.96 means that the Baud Rate of COM1=9600bps; a displayed value of 2.1152 means that the Baud Rate of COM2=115200bps. All PDS-700 COM port Baud Rates will be shown in sequence.

The COM ports configurations are as follows:

- 5-digit LED Group ID: 33333
- LED-1: COM port number
- LED-3: data bit: 5 , 6 , 7 or 8
- LED-4: parity bit, n=no parity, E=Even parity, o=Odd parity
  ⎍ = Mark parity, S = space parity
- LED-5: stop bit: 1 or 2

The connected-client and debug information is as follows:

- 5-digit LED Group ID : 44444
- LED-1 will display 1, 2, 3, 4, 5 and module name in sequence.
- When LED-1 is 1, LED-2/3 indicates the number of free sockets available on (default is 26 for PDS-700), and LED-4/5 shows the number of sockets being used by clients (default is 0) , e.g. 12600
- When LED-1 is 2, LED-2~5 indicates how many times the PDS-700 has been reset, e.g. 20002 (The PDS-700 is reset for 2 times)
- While LED-1 is 3, the information indicates that how many Ethernet packets enter into the PDS-700 at present.
- While LED01 is 4, the information indicates that the status of internal Flag used to allow the Ethernet packets can be send is 0 or 1.
- When LED01 is 5, the information indicates that the reset number of Ethernet chip.
- Module name : dS.7xx

When the PDS-700 is first powered-up or just been reset, the reset state=1. If any one client connects to this PDS-700, the reset-state will be changed to 0, free-sockets will be decreased and used-sockets will be increased. If the free-sockets number is reduced to 0, then no extra clients can link to this PDS-700. **The default number of free-sockets is 26 for the PDS-700. Therefore, the server (VxComm firmware or Xserver firmware) allows 26 connections link to one PDS-700. Each client program occupies at least 2 connections for a serial port, one for data and another one for command.**

If the 5-digit LEDs do not shown as above, you can do the following steps:

- Power off
- Connect INIT* to VS+

- Power on and double check

Step 3: There is a red indicator-LED in the PDS-700 as follows:

- VxComm is running: **On/Off**
- Xserver is running: **On/Off**

The default shipping of PDS-700 will be Xserver or VxComm inside, so the red indicator-LED of PDS-700 will be ON 0.5 second then OFF 0.5 second periodically.

If the LED is always ON, you can do the following steps:

- Power off
- Connect INIT* to VS+
- Power on and double check

Step 4: Power off.

4. Setup the PDS-700

Step 1: Connect the PDS-700 to Ethernet

Before connecting the PDS-700 to Ethernet, you need to prepare:

1. Power Supply：+10V ~ +30V/DC (EX：DP-665:
   http://www.icpdas.com/products/Accessories/power_supply/powerlist.htm)
2. Hub (EX：NS-205
   http://www.icpdas.com/products/Switch/industrial/industrial_list.htm)

3. The PC has available network settings and is working well on Ethernet.

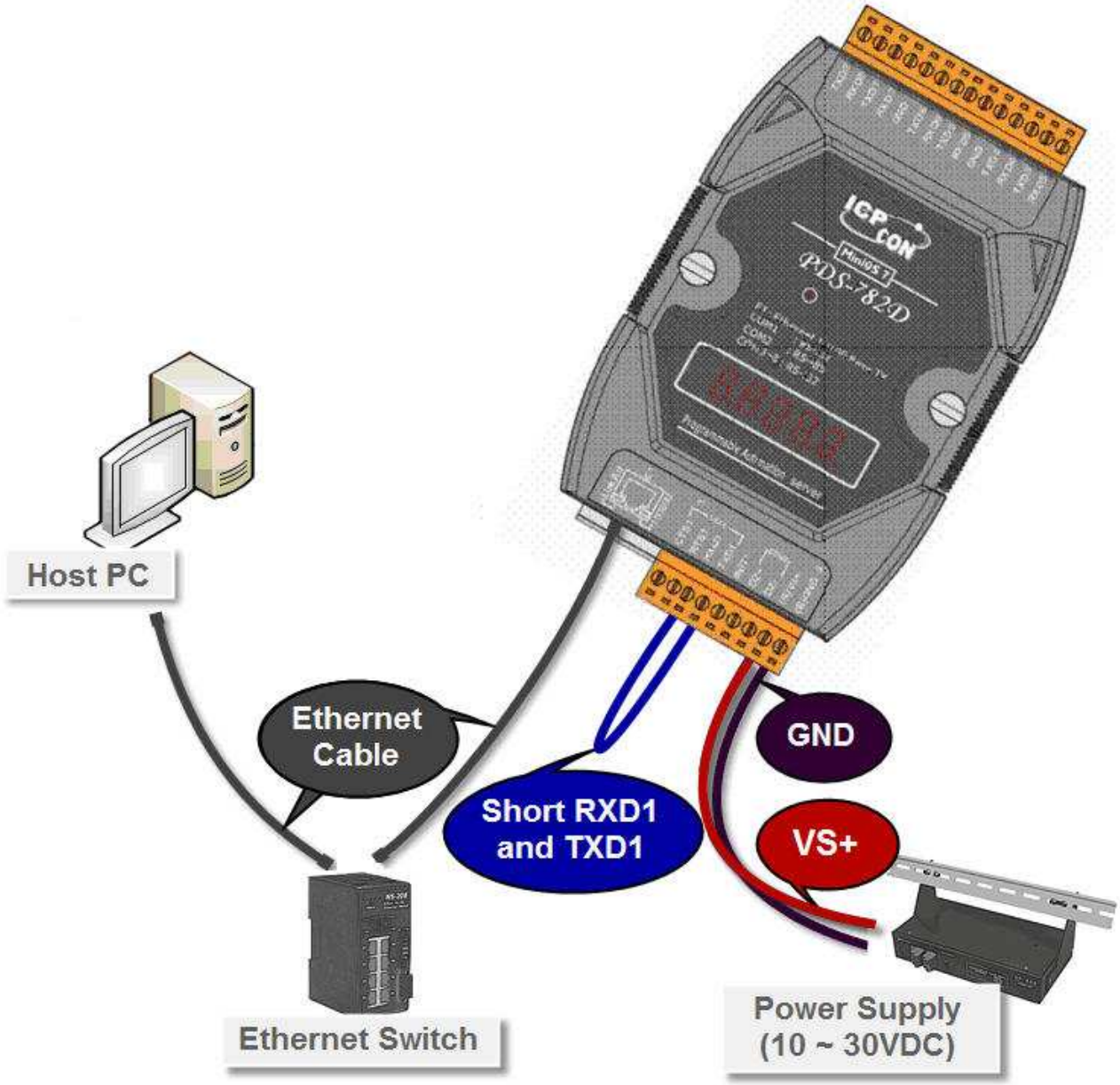
4. Disable or well configure your Windows firewall and Anti-Virus firewall first, else the "**Search Servers**" on VxComm Utility may not work. (Please contact with your System Administrator)

5. Connect the PDS-700 to Ethernet as next page and power-on it.

6. Make sure the Indicator LED is flashing. (ON 1 second and then OFF 1 second periodically.)

   If your PDS-700 module is a D-version modules, the 5-digits 7-SEG LED will show the system information described in sec.3.7

7. Install VxComm Utility on your PC
   The software is located at：
   CD:\Napdos\7188e\tcp\vxcomm\driver(pc)\
   http://ftp.icpdas.com/pub/cd/8000cd/napdos/7188e/tcp/vxcomm/driver(pc)/



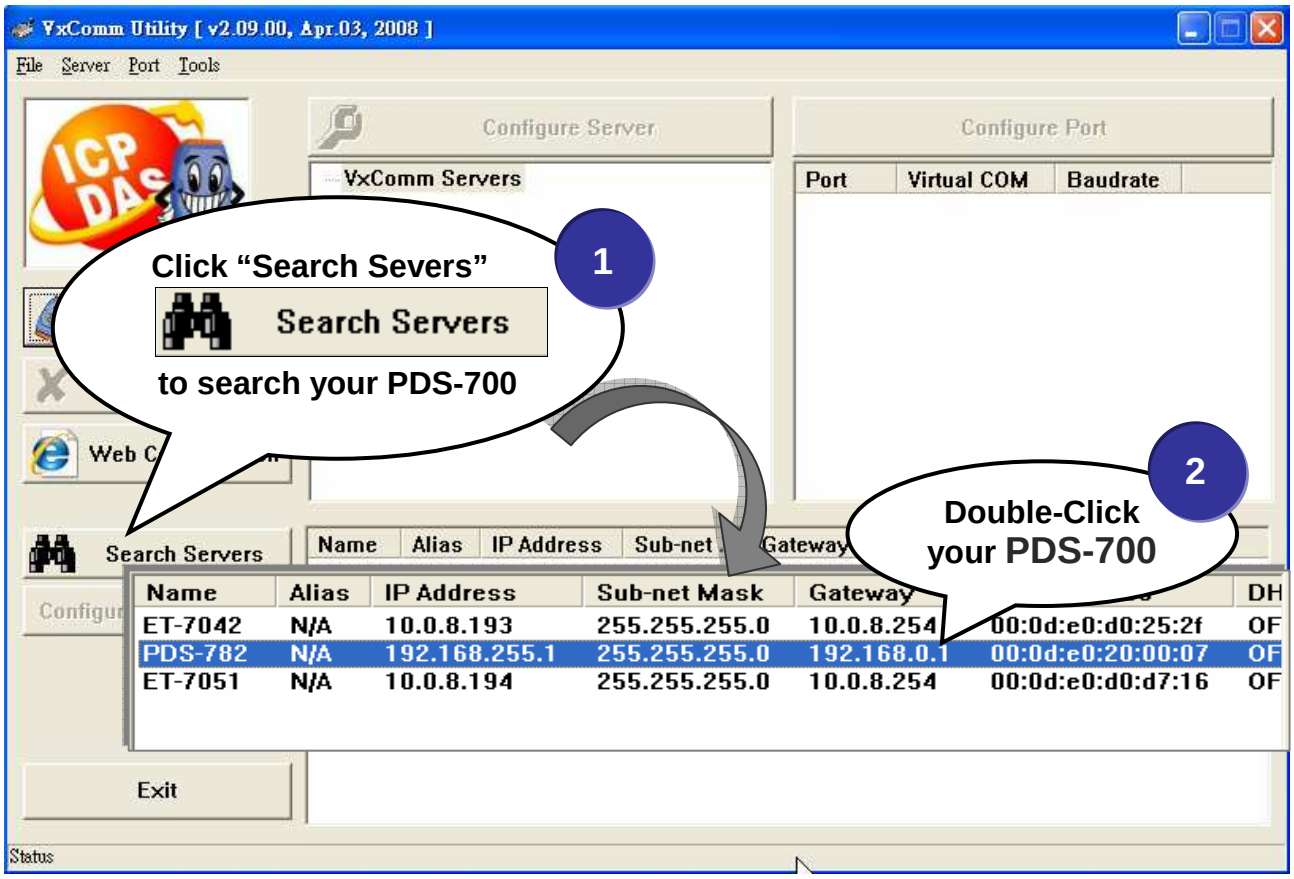
Connect the PDS-700 and your computer to the same sub network or same Ethernet Switch.

Short the RXD1 and TXD1 pins of PDS-700 for self-test.

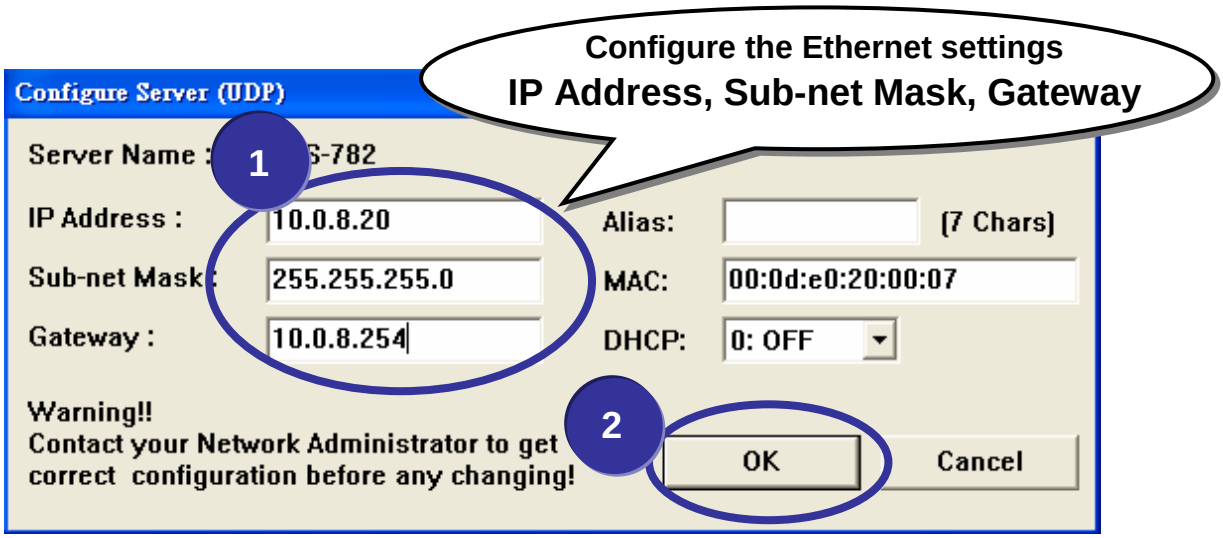
Provide 24VDC (10 ~ 30VDC) power to PDS-700.

Step2: Search the PDS-700 on Ethernet

1. Run VxComm Utility and then search your PDS-700.
2. Double-Click your PDS-700 to configure the Ethernet settings.

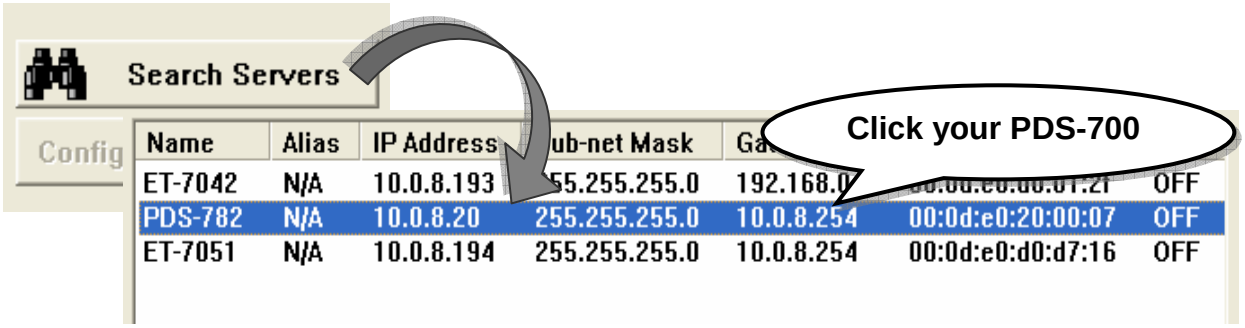


3. Contact your Network Administrator to get correct network configuration (such as IP/Mask/Gateway); Set the network settings and then click "OK". The PDS-700 will restart it-self immediately.



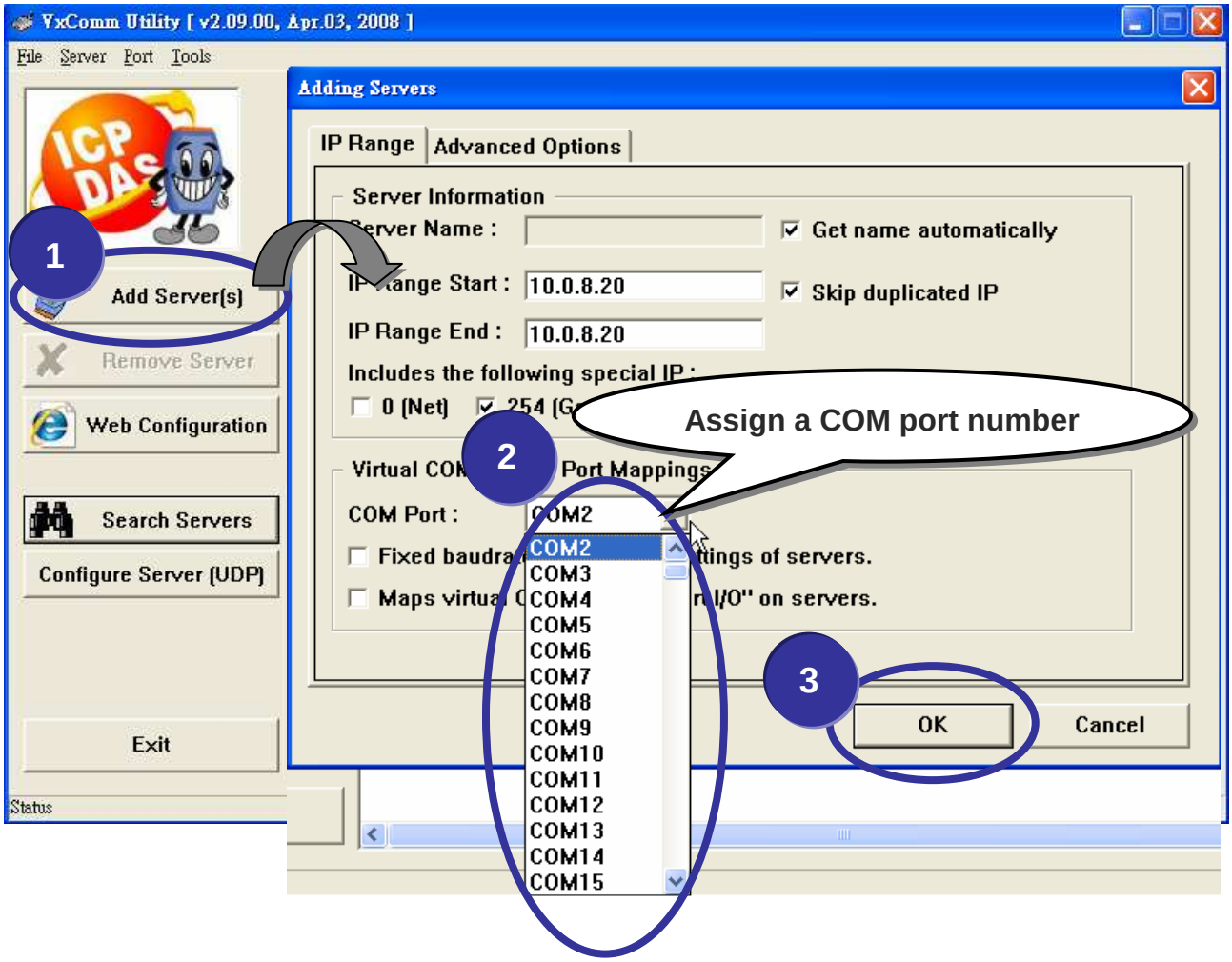
Step3: Configuring virtual COM ports

1. Again search out your PDS-700 to make sure the new IP/Mask/Gateway are set in, then click your PDS-700 to select it.

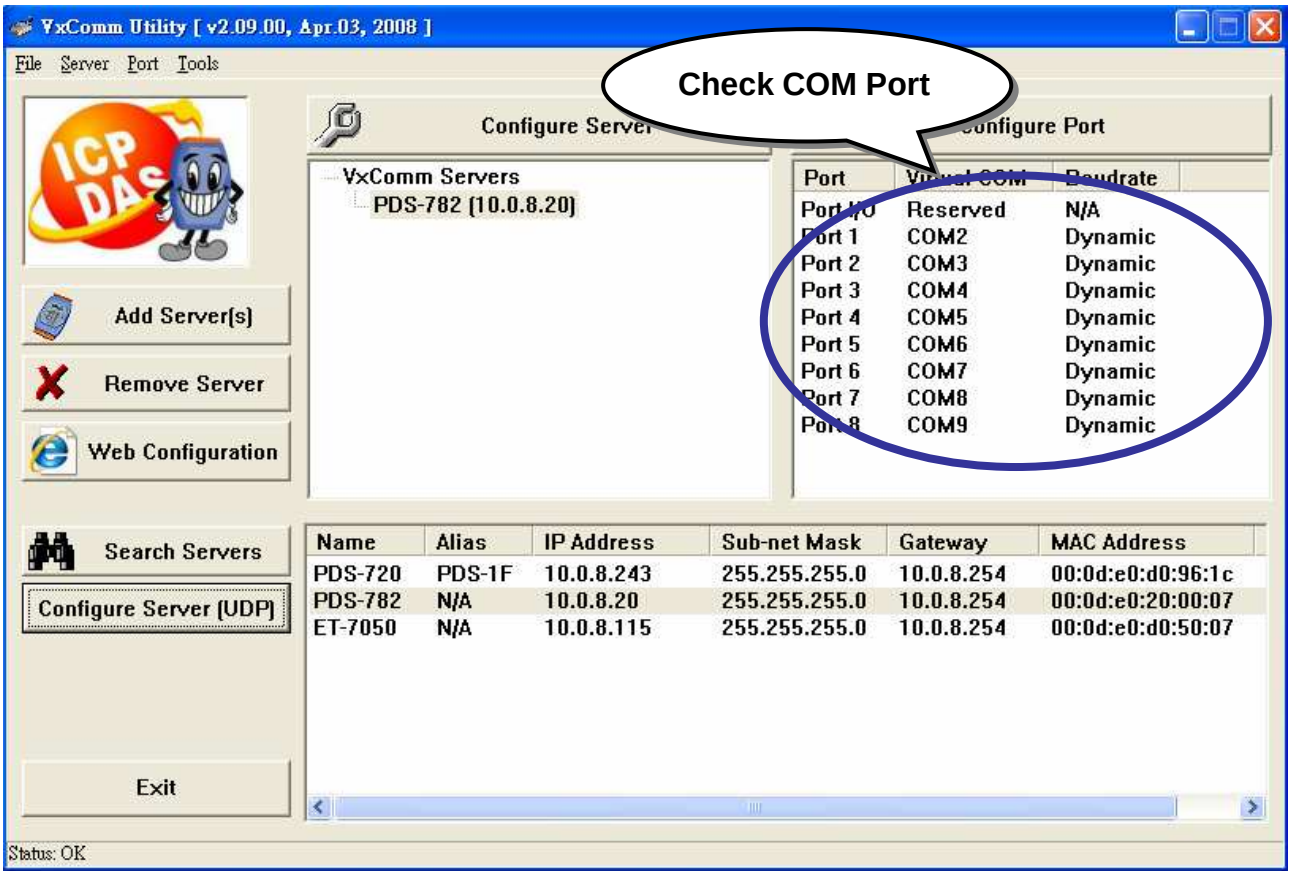


2, Click 

, assign a COM port number and then click "OK".



3. Check the COM port numbers virtualized on PC.

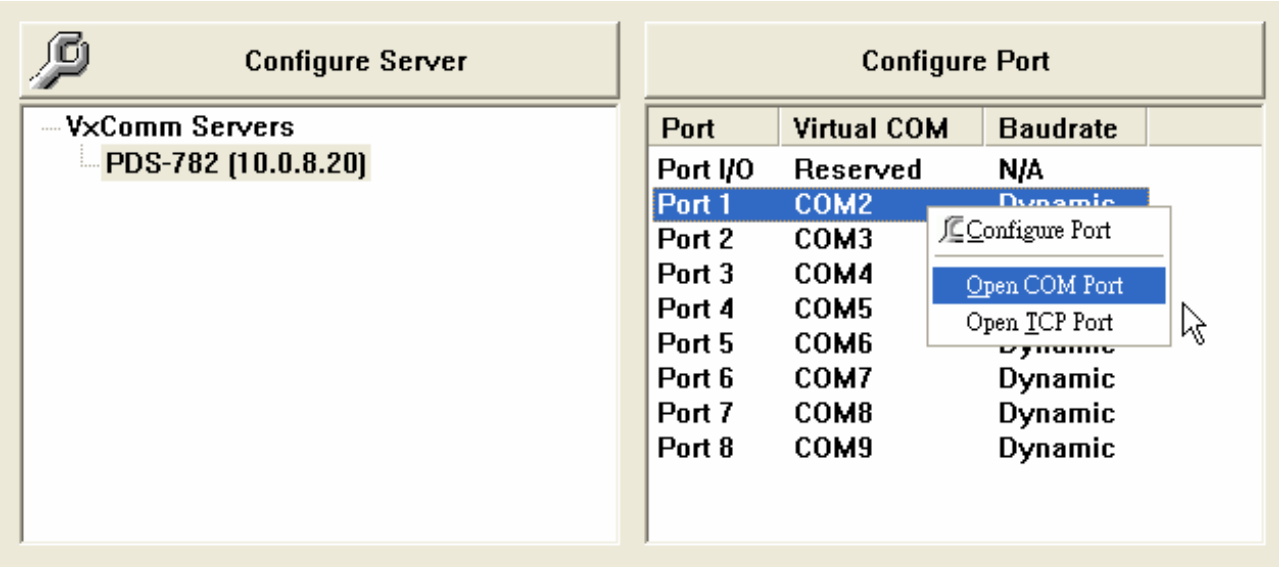


4. Click "**Tools** -> **Restart Driver**" and then click "**Restart Driver**" to start the driver.

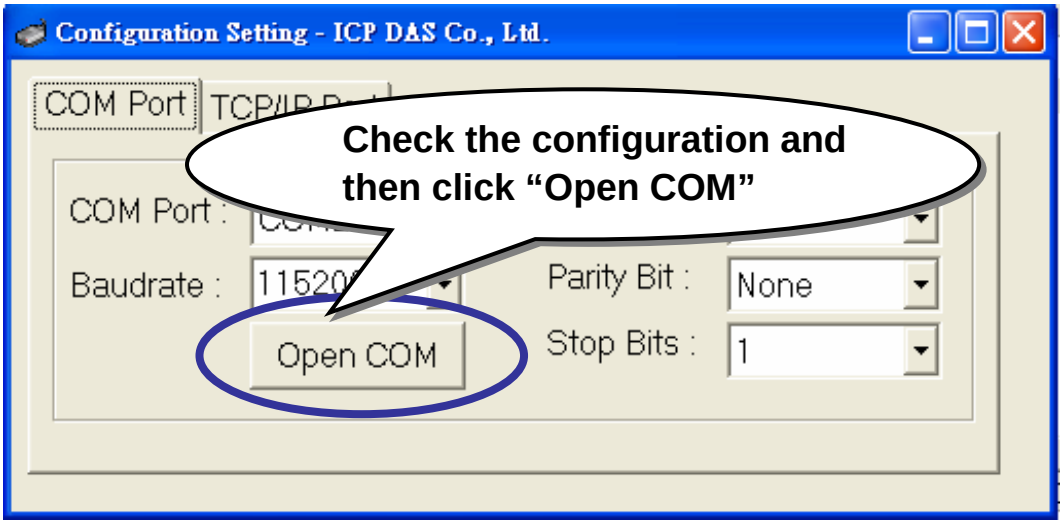
Step4: Testing your PDS-700

1. Wire “RXD1” and “TXD1” of the PDS-700, shown as picture on Step 1.

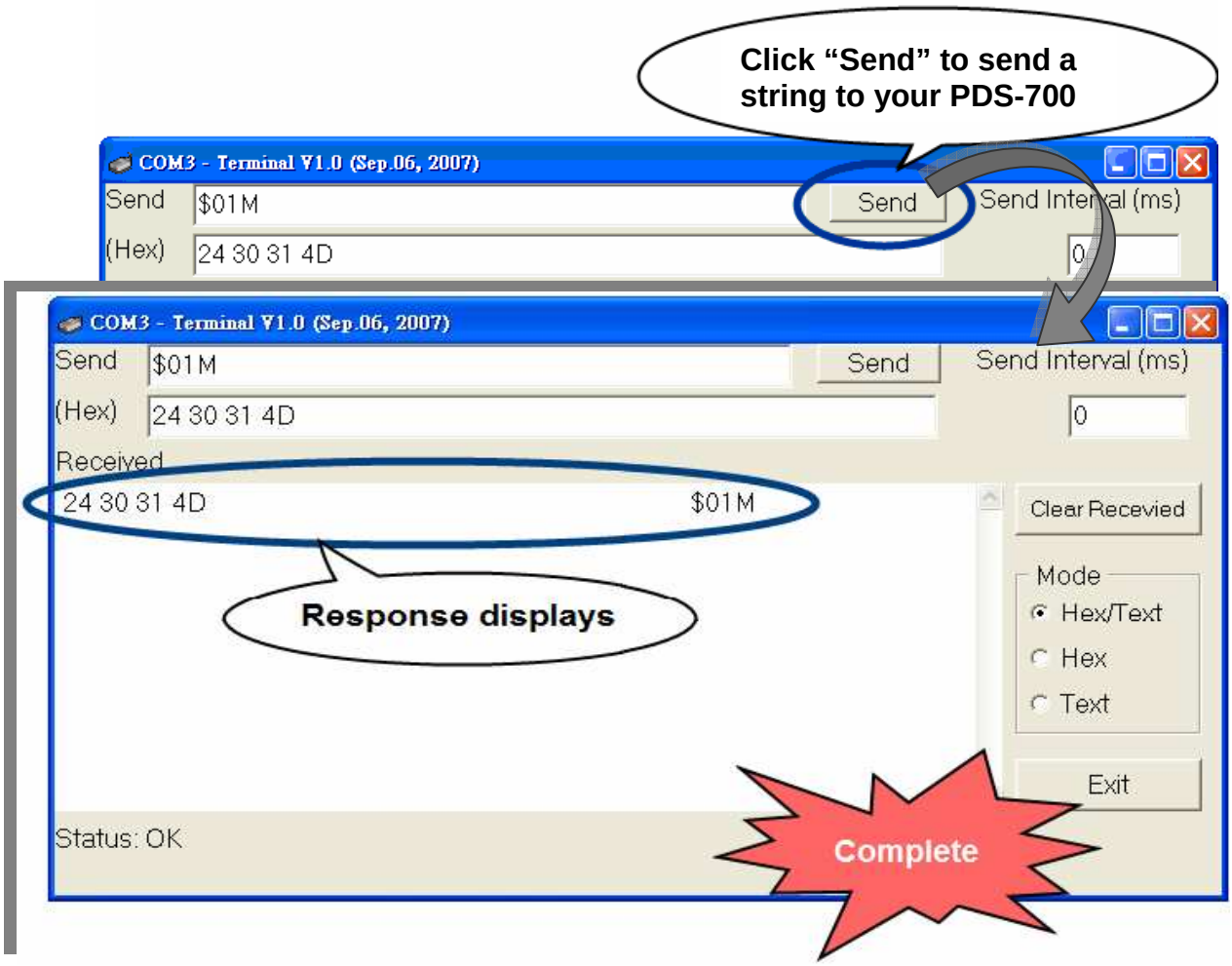
2. Right-Click the Port1 and then choose “**Open COM Port**”.



3. Check the COM Port’s configuration and then click “**Open COM**”.



4. Send a string and get the response = string sent.



5. After the test successful, your COM port program can work direct by setting the Virtual COM Port number.

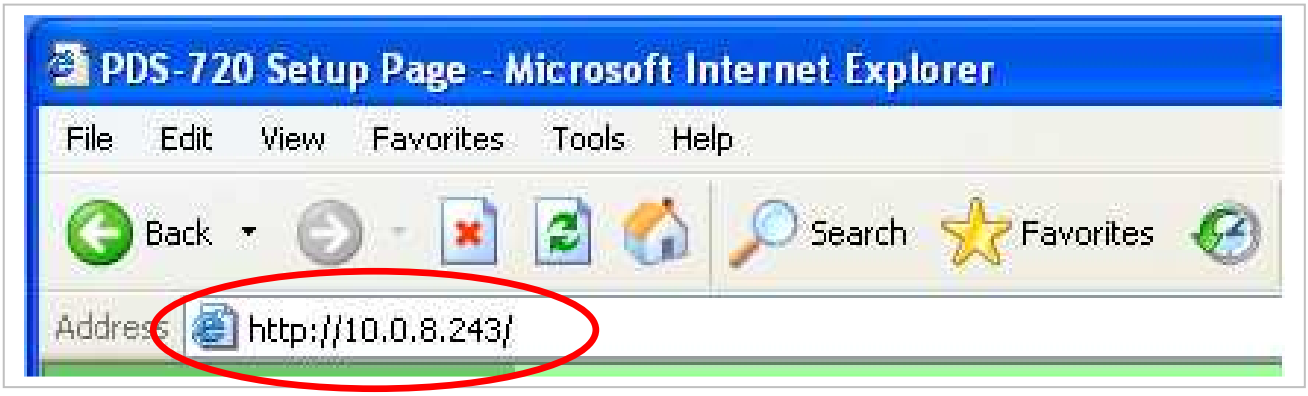
5. Configuration with Web Browser

Now the PDS-700 is setup complete and working well with your program based on COM port. If you need to get or change the configuration of the PDS-700, besides using VxComm Utility, you can also use standard Web browsers (IE, FireFox, Mozilla, etc.) to set the new configuration into the PDS-700.

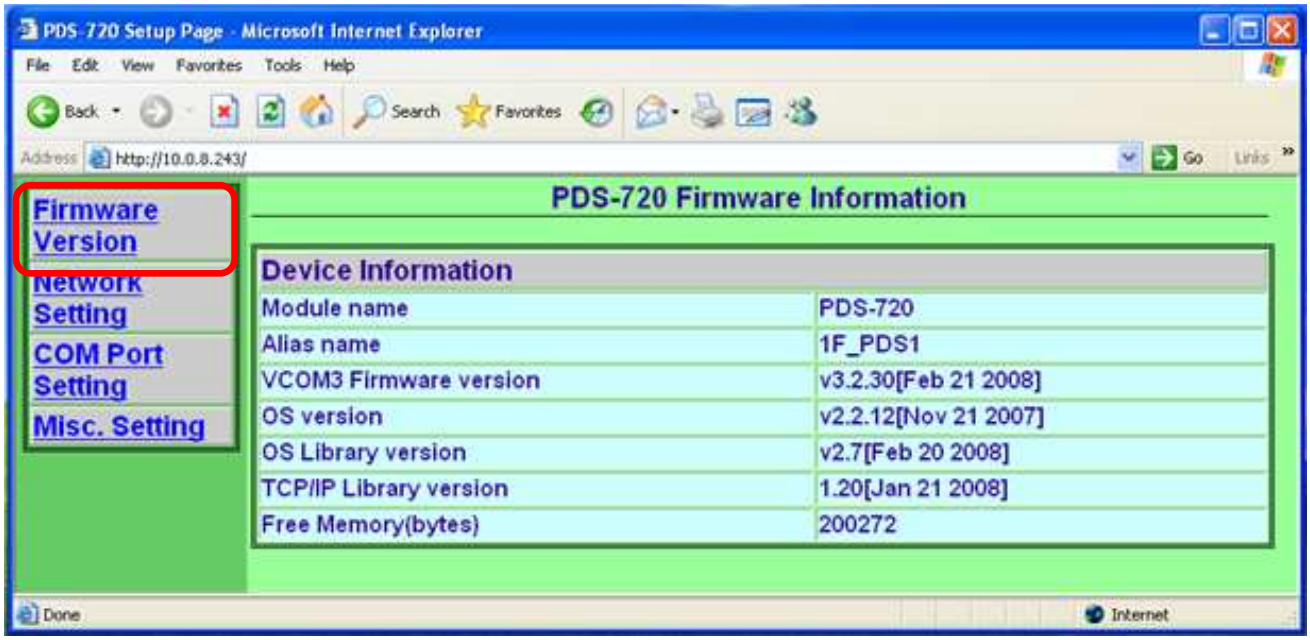
If the program based on COM port is running with PDS-700, change the configuration of the PDS-700 will cause the program error.

5.1 Linking with the PDS-700

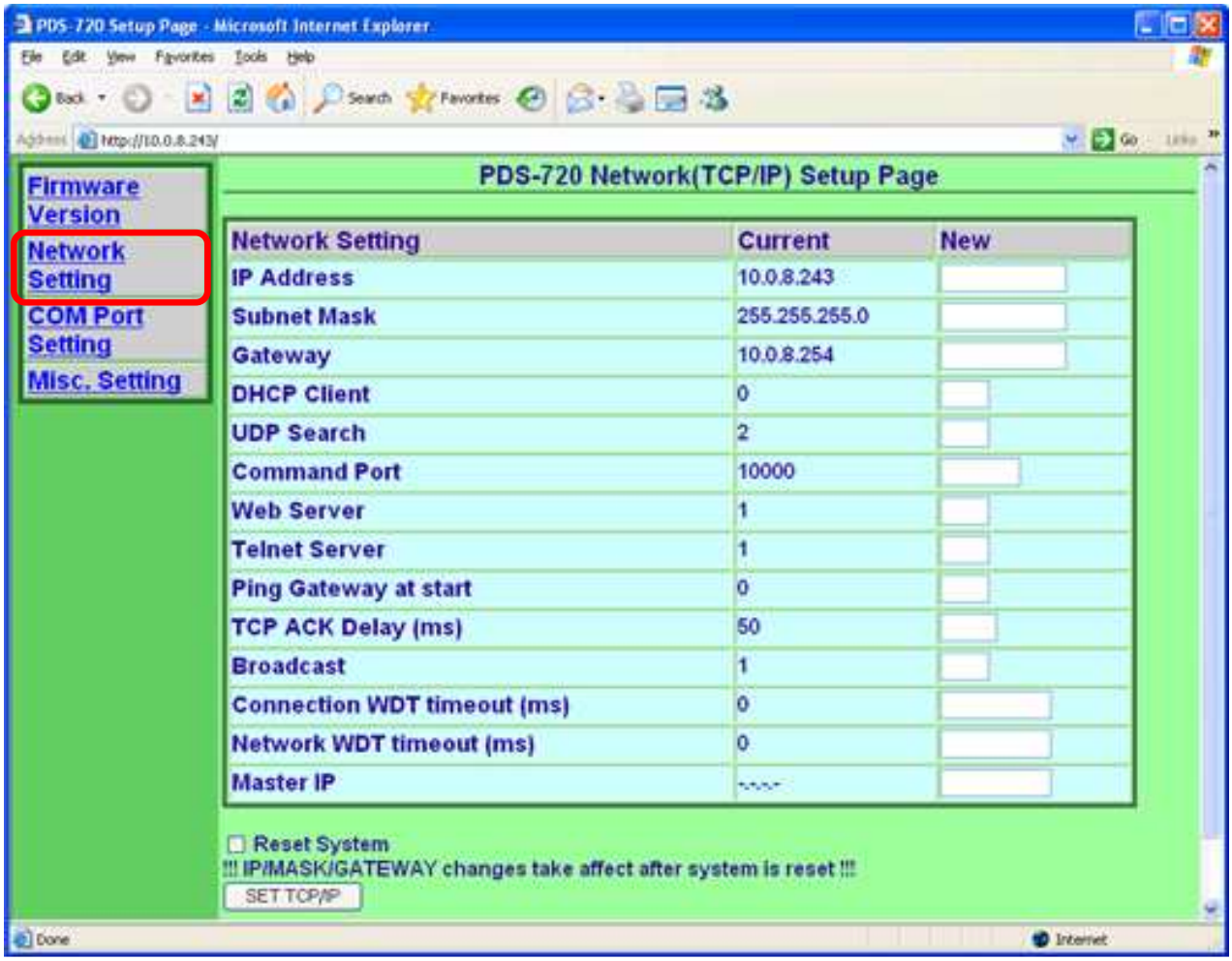
Key in the IP of the PDS-700 in the Address line and press ”Enter” key to link with the PDS-700.



When the browser connects with the PDS-700, Firmware Information page is the first page.



5.2 Network Setting



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

  The most important network settings, they should correspond to the LAN definition. If the three settings do not meet the LAN definition, the PDS-700 cannot work; if they are changed into a working PDS-700, the running program based on COM port will lose virtual COM port linking and get error.

- **DHCP Client** : 0 = disable, 1 = enable
  Keep the setting = 0, DHCP client disabled.

- **UDP Search**: 0 = disable, 1 = always enable.
  2 = enable UDP search till any client connected. (Default = 2)

  Keep the setting = 2 can reduce the PDS loading. If UDP search = 0, your PDS-700 can not be searched by VxComm Utility again.

- **Command Port**
  Keep the command port = 10000.

- **Web Server**
- **Telnet Server**
  0 = disable, 1 = enable

- **Ping Gateway at start**: 0 = disable, 1 = enable.
  If the setting is 1 (enable), the PDS-700 will send ping packet to gateway when power-on. It is for informing the gateway that a PDS-700 (itself) joins into the network.

- **TCP ACK Delay (ms)**, default = 50.
  If the PDS-700 has no data to respond to the PC client's request, the PDS-700 will delay time of "TCP ACK Delay" setting and then send an ACK. In a busy network, a shorter delay time is recommended to try to get a better efficiency.

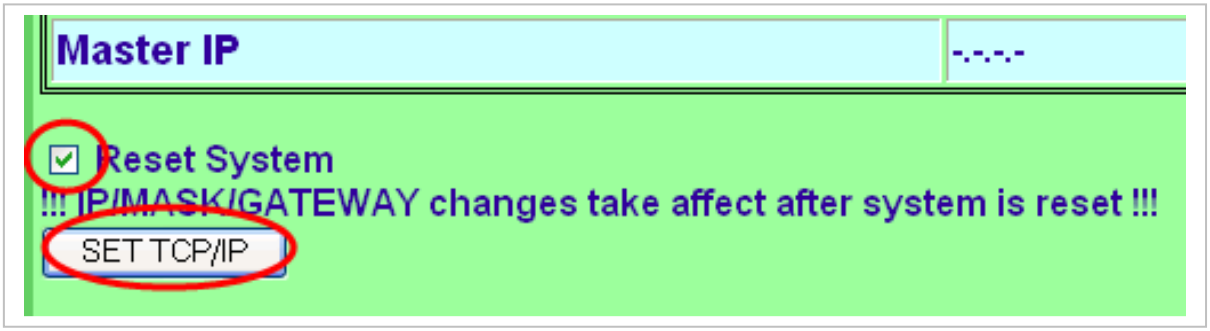
- **Broadcast**
  1 = receive UDP broadcast packets
  0 = reject UDP broadcast packets

- **Connection WDT timeout (ms)**: default = 0 (disable), min. = 10000.
  When the PDS-700 does not receive data from a client PC over the period of “Connection WDT timeout” setting, the PDS-700 will close the connection between the client and itself.

- **Network WDT timeout (ms)**: default = 0 (disable), min. = 30000.
  When the PDS-700 does not receive data from all the clients over the period of “Network WDT timeout” setting, the PDS-700 will reset itself.

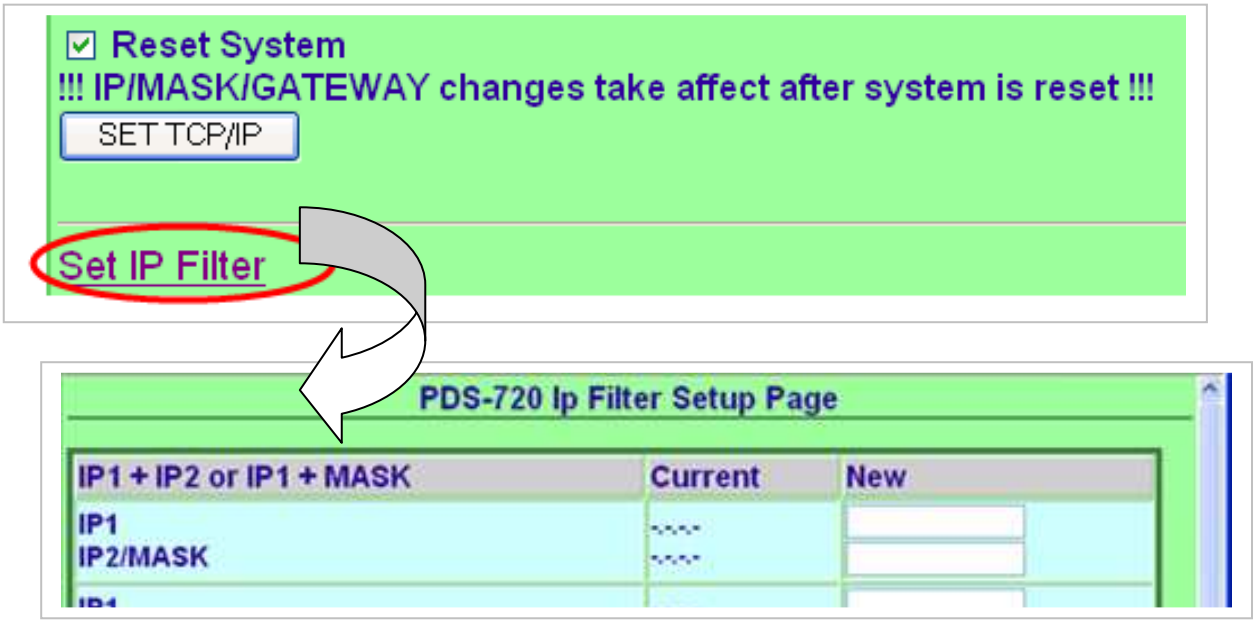
- **Master IP**: default = empty (disable).
  If Master IP is set, only the client of Master IP owner can change the COM port configuration. It is to prevent the COM configuration changed by other clients.

After fill in the new configurations, press “**Set TCP/IP**” button to set the new settings into the PDS-700. If the Reset System is checked, the PDS-700 will reset after the setting operation complete, otherwise all the settings will be valid after next power on.



5.3 IP filter setting

The IP filter setting limits which client PC can link with the PDS-700 by IP. When one or more IP are set in the table, only the client PC which’s IP is in range of setting could link with the PDS-700, other PC’s linking request will be rejected

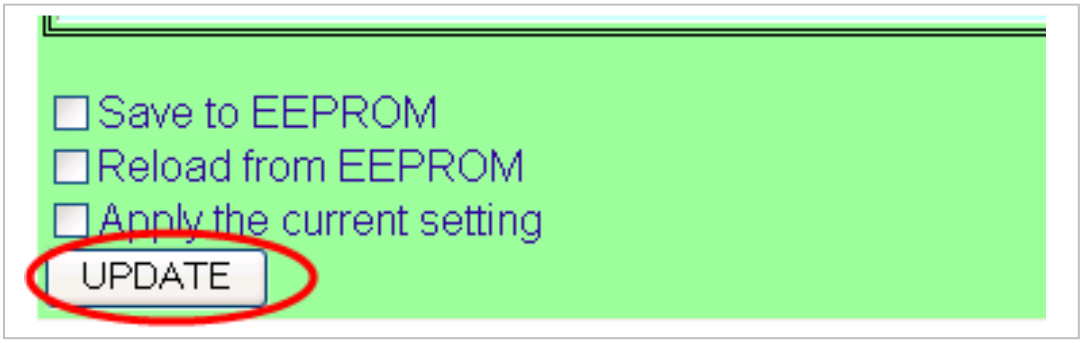


- **Set IP1 only**: only a client which's IP is set in the table can connect with the PDS-700
- **Set IP1 + IP2**: set a range of IP as a starting and ending address. The setting allows a client which's IP is in the range connecting with the PDS-700.
- **Set IP1+Mask**: set the IP filter range = (IP1 & Mask) + 0 ~ (IP1 & Mask) + (~Mask). Only the client which's IP is in the range can connect with the PDS-700. For instance:

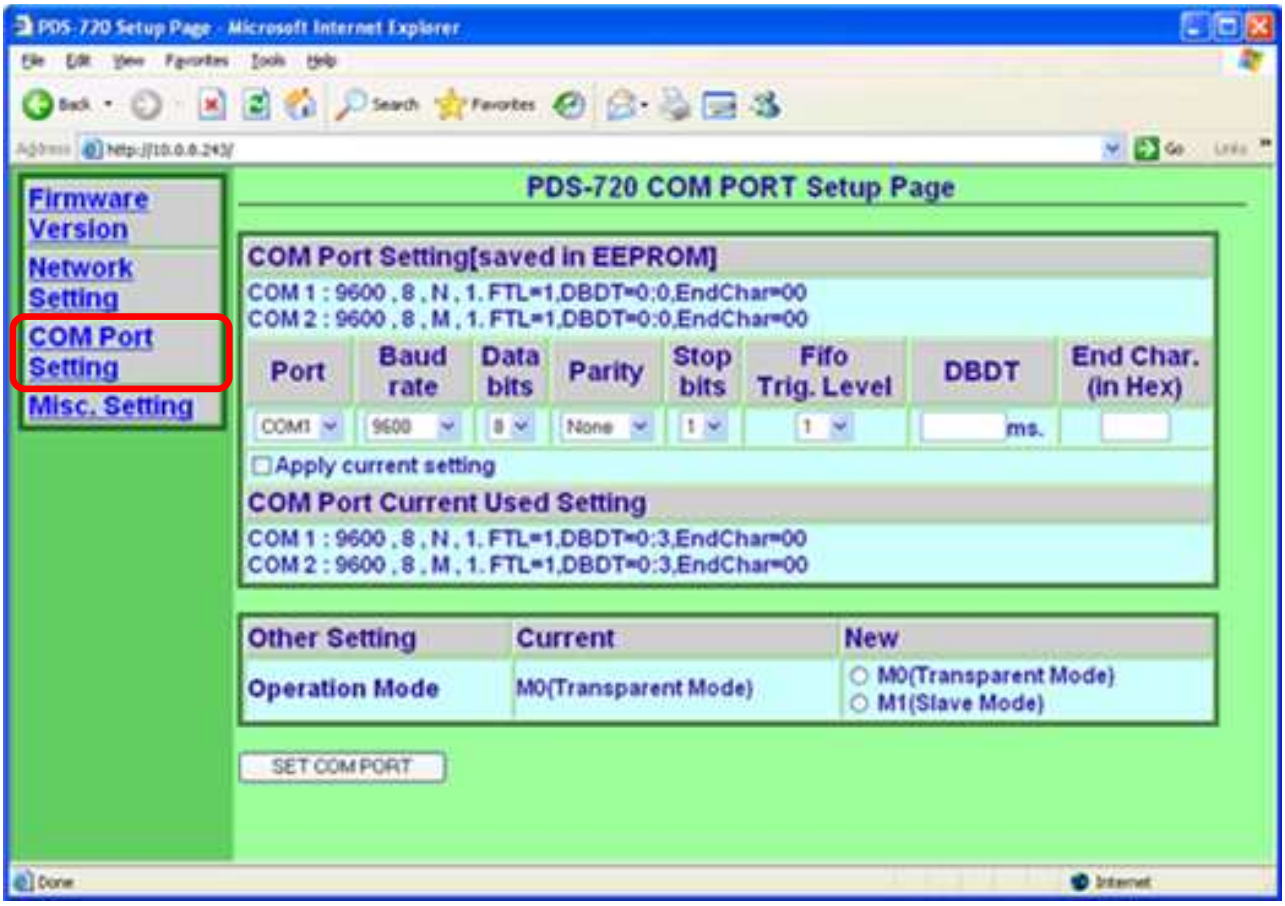
  IP1=10.0.9.5, mask=255.255.255.0
  IP1 & MASK = 10.0.9.0, ~mask = 0.0.0.255

  It allows a client which's IP is in the range of 10.0.9.0 ~ 10.0.9.255 connecting with the PDS-700.

- Move to bottom of the IP filter setup page and press the "**Update**" button to validate the settings.



5.4 COM Port Setting



- The COM Port Setting list saved in PDS-700

COM Port Setting[saved in EEPROM]

COM 1 : 9600 , 8 , N , 1. FTL=1,DBDT=0:0,EndChar=00
COM 2 : 9600 , 8 , M , 1. FTL=1,DBDT=0:0,EndChar=00

- The COM Port Current Used Setting list

COM Port Current Used Setting

COM 1 : 9600 , 8 , N , 1. FTL=1,DBDT=0:3,EndChar=00
COM 2 : 9600 , 8 , M , 1. FTL=1,DBDT=0:3,EndChar=00

- The COM Port Setting area

If "**Set COM Port**" button is pressed to set the new COM port setting without checking "**Apply current setting**", the new setting will be saved into the PDS-700 only, the new setting will be valid after next power on. If "**Apply current setting**" checked when "**Set COM Port**" button is pressed, the new setting will be valid immediately.

- **Port** : COM port number on PDS-700

- **Baud rate, Data bits, Parity**
- **Stops bits, End Character**:
  The configuration settings should meet to your serial device used.

- **Fifo Trig. Level**: FIFO trigger level
  It is to set the number of characters that when COM received each time, the PDS will move the data from COM port FIFO into PDS. If your data transferred is huge and by fast transfer speed (115200 bps), set a smaller number is helpful for preventing data lost.

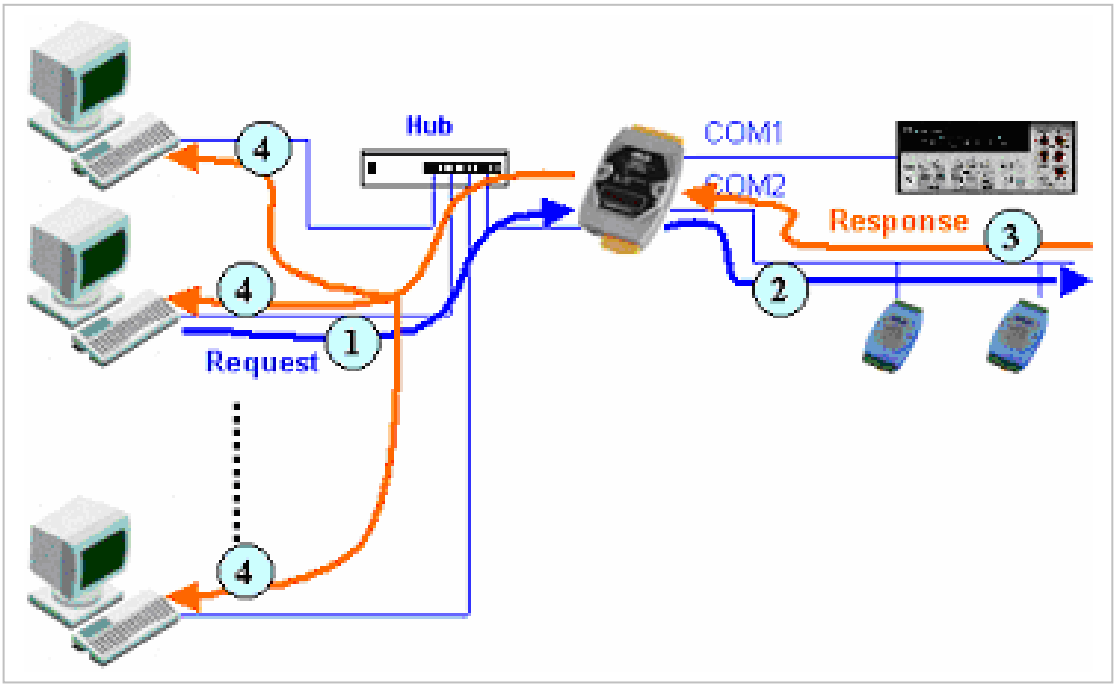
- **DBDT(ms)**: Data buffer delay timeout
  When the COM port does not receive data from devices connected over the period of DBDT setting, the PDS-700 will determine that the data transfer is over and return to process next tasks.

- **Other setting**: M0 mode

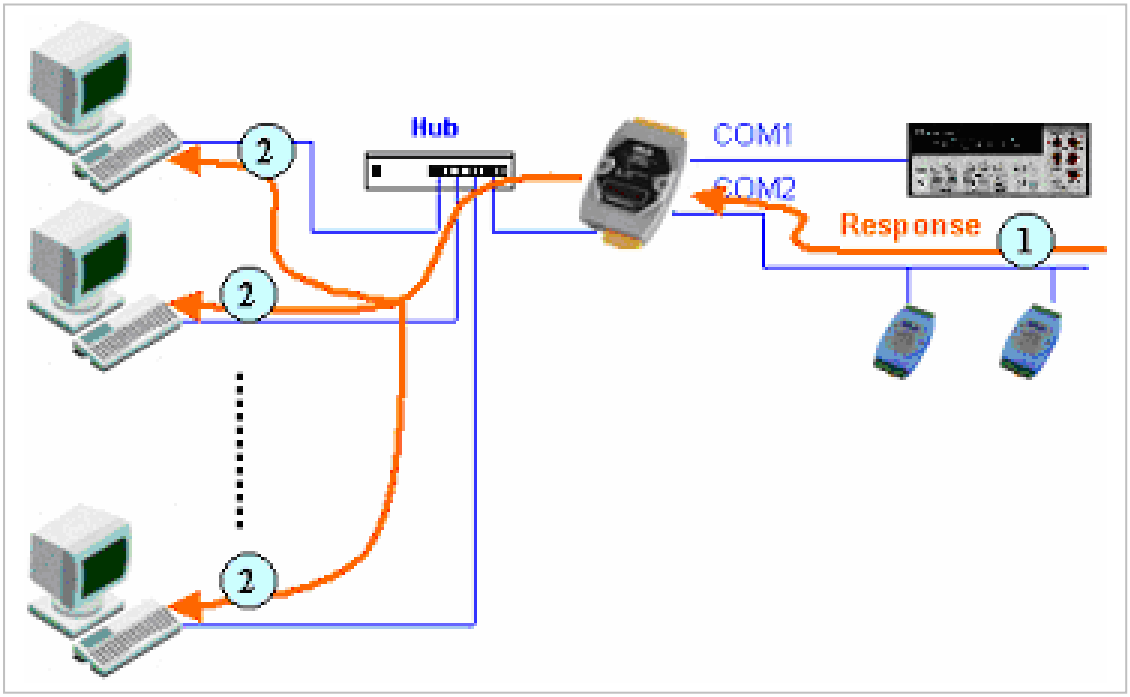
Other Setting	Current	New
Operation Mode	M0(Transparent Mode)	○ M0(Transparent Mode) ○ M1(Slave Mode)

M0: Transparent Mode (Multi-echo mode)

Condition 1: One client sends a request to PDS-700 to access devices. The PDS-700 echoes data from devices to every client connected.

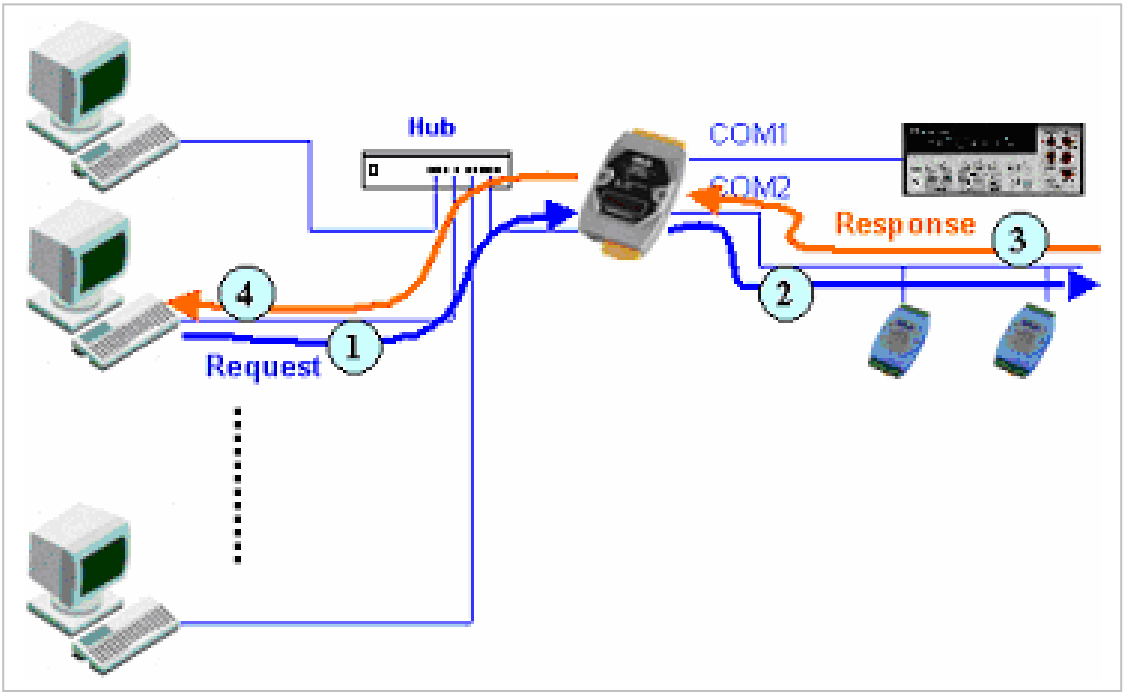


Condition 2: No clients send a request to the PDS-700. The PDS-700 echoes data from devices to every client connected.

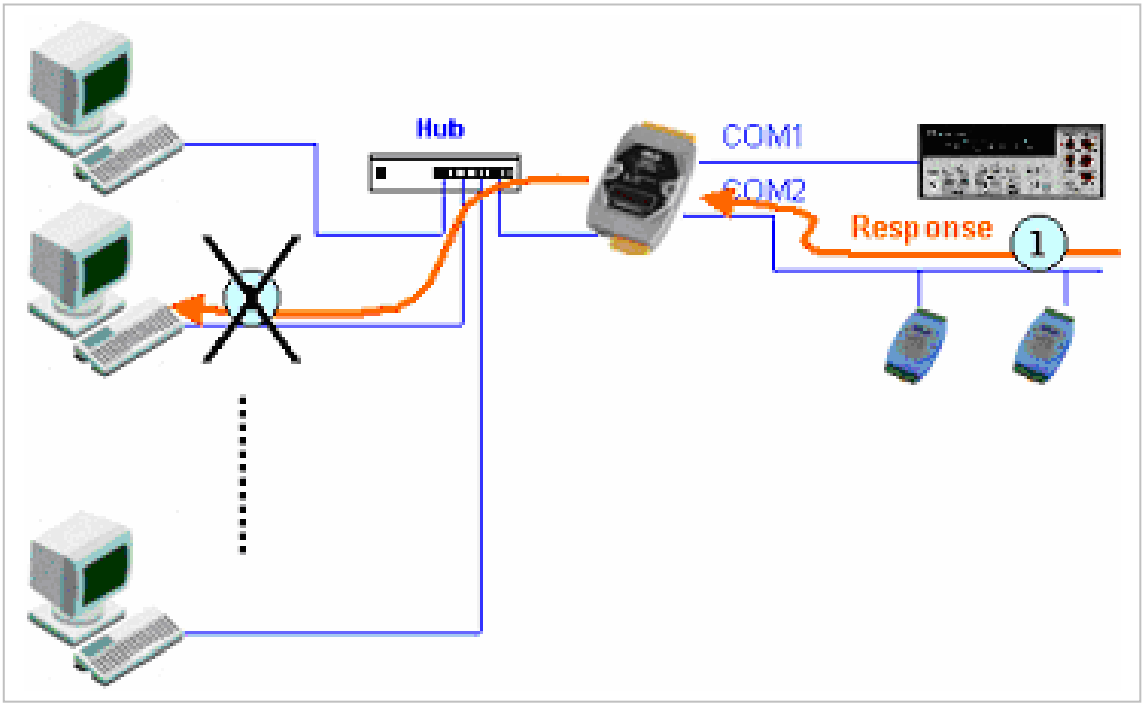


M1: Slave Mode (Single-echo mode)

Condition 1: One client sends a request to PDS-700 to access devices. The PDS-700 echoes data from devices to the client that requested the service.

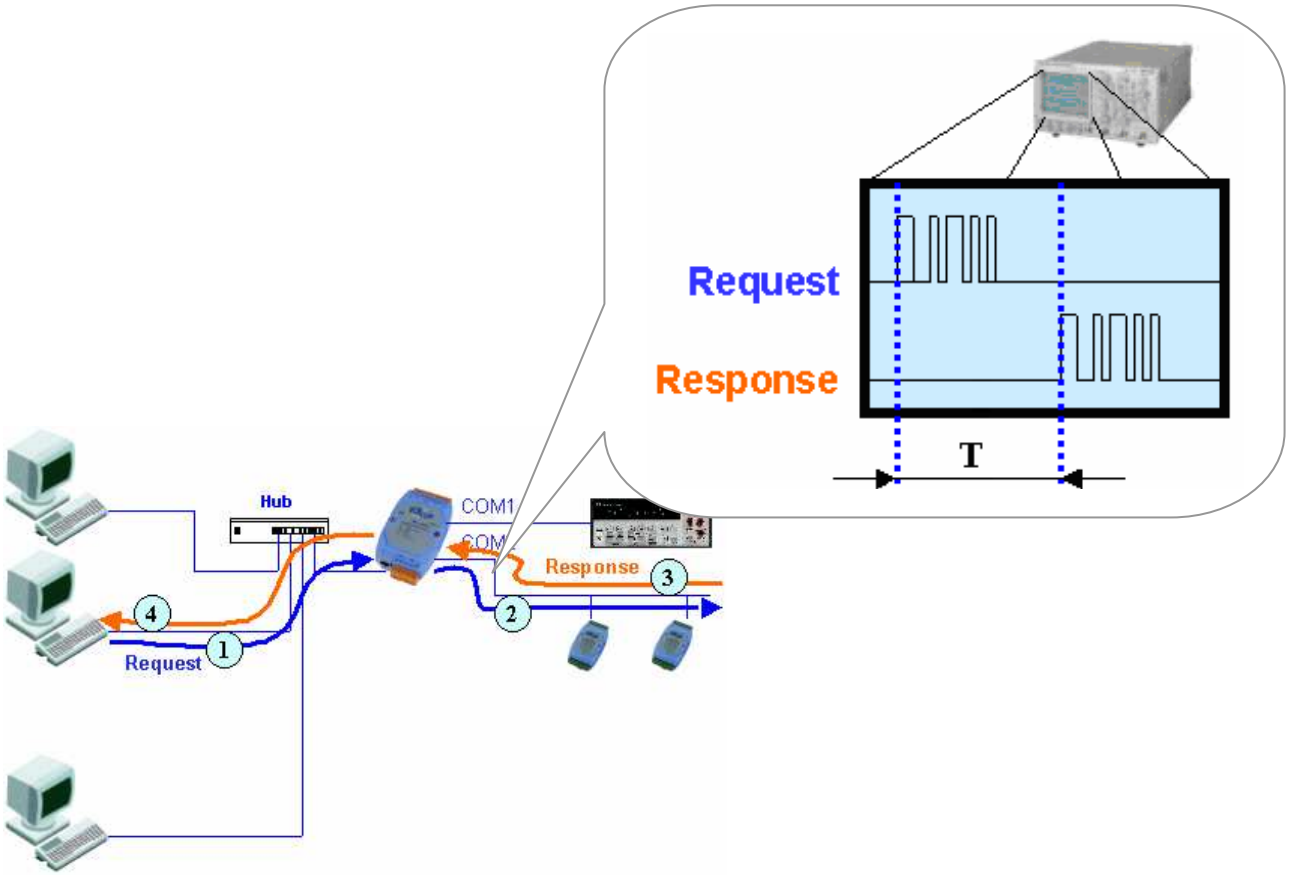


Condition 2: No clients send any request to PDS-700. The PDS-700 doesn't echo data from devices to any client.

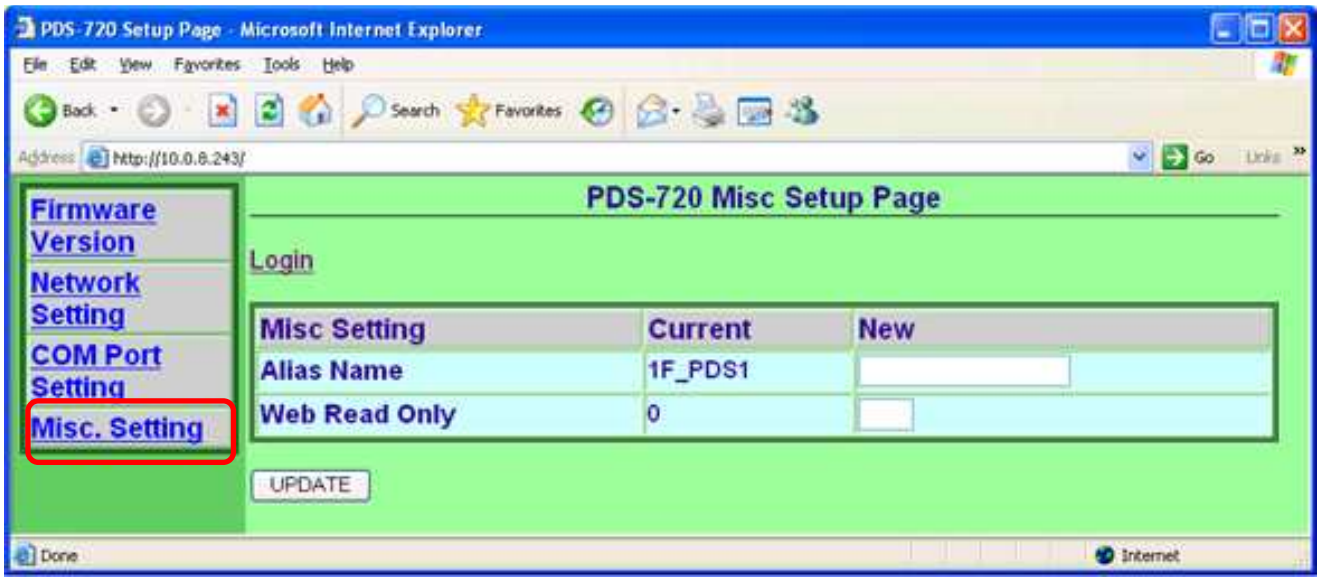


In M1, the salve mode timeout setting is for setting the waiting time after last character of request sent to device. If the device does not response in the setting time, the PDS-700 will return a timeout error and process next request.

Other Setting	Current	New
Operation Mode	M1(Slave Mode)	○ M0(Transparent Mode) ○ M1(Slave Mode)
Slave Mode Timeout(ms)	100	



5.5 MISC setting

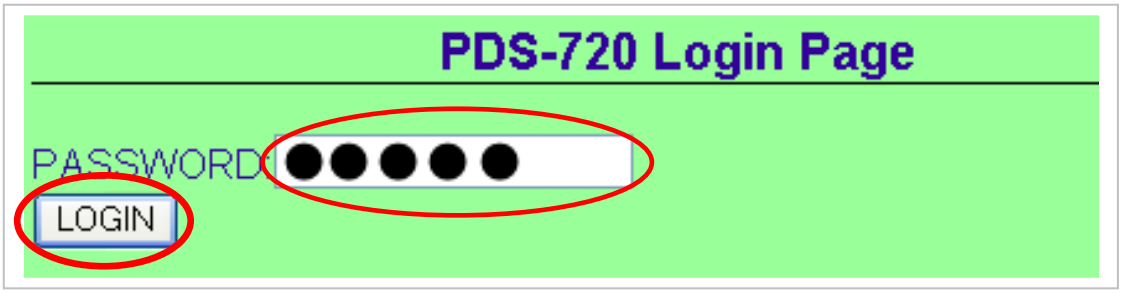


- **Alias Name**: give the PDS-700 an alias name.
- **Web Read Only**: 0 = disable, 1 = enable

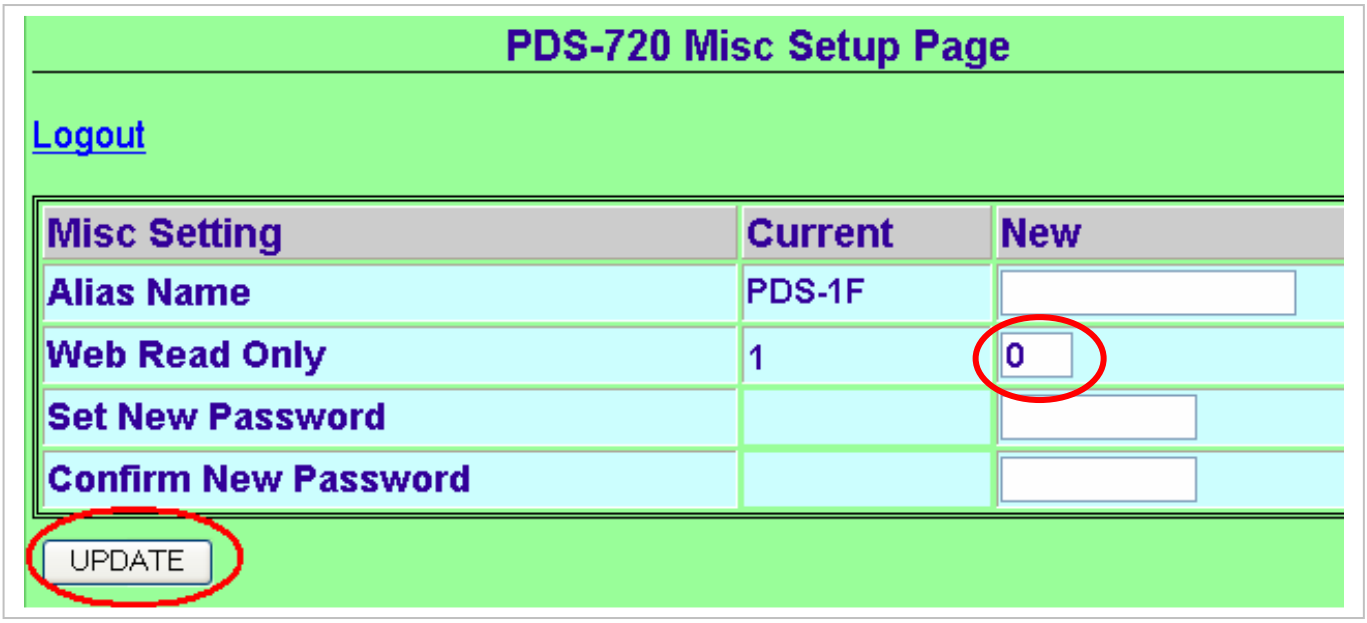
*While the "**Web Read Only**" is set to 1, enabled, the web server can **not** set new configuration into the PDS-700 anymore. To disable "**Web Read Only**" property, please refer to the following paragraph.*

- **Login**: for disabling the "***Web Read Only***" *property* or setting new password

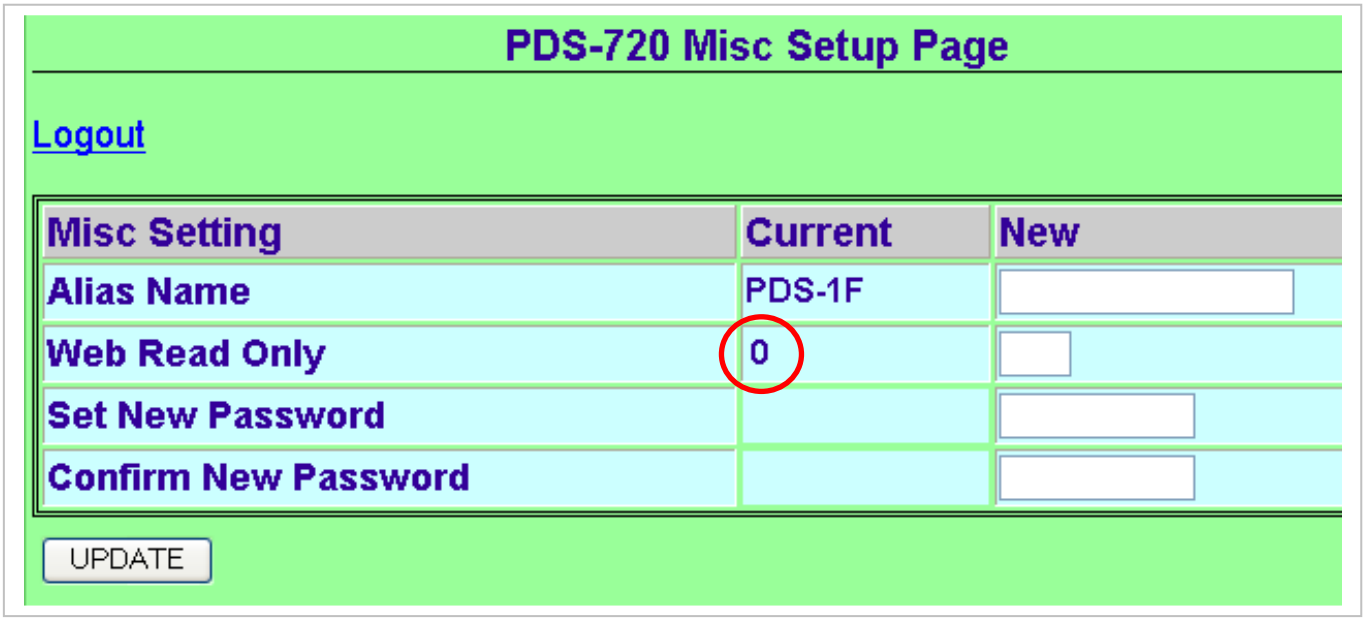
1. Enter the password and press "**LOGIN**" button to go into the setting page



2. Set the “**Web Read Only**” New = 0 and press “**UPDATE**” button



3. Check the “**Web Read Only**” current = 0 and then click “**Logout**” to finish the operation



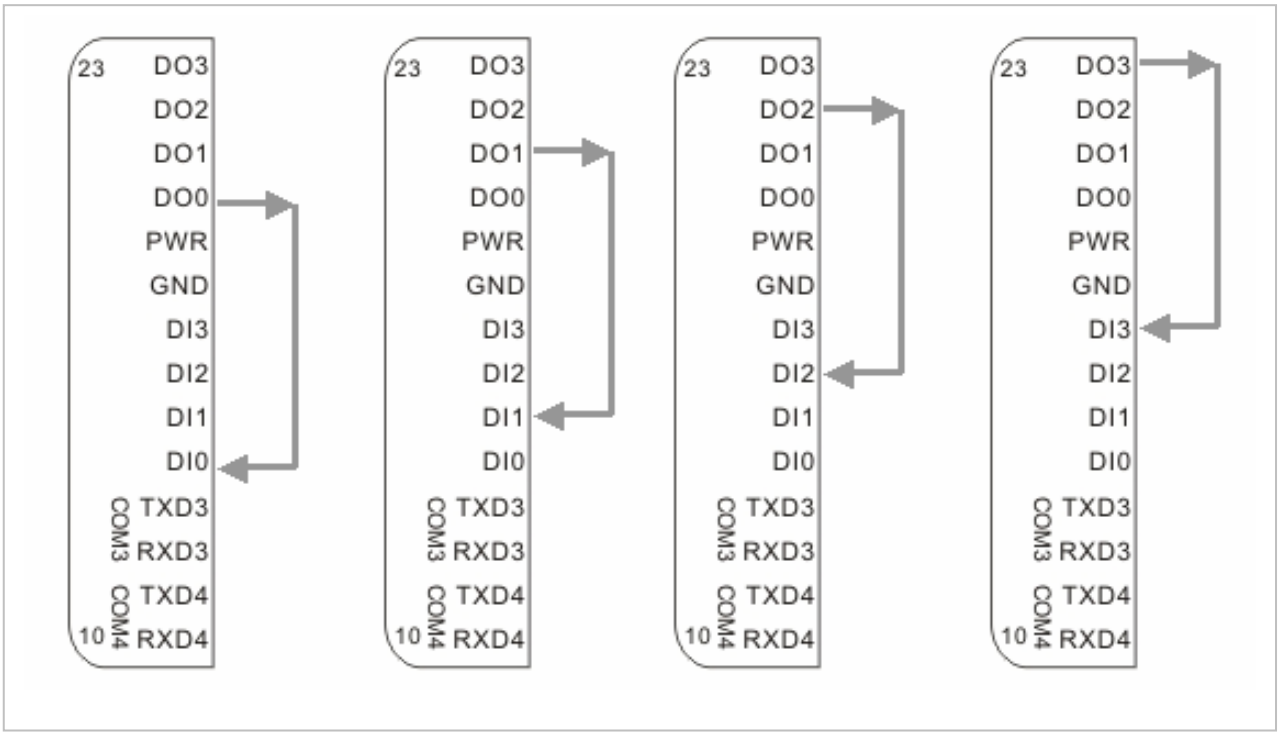
6. Virtual I/O

The PDS-700 provides digital I/O lines, such as PDS-721/721D, PDS-732/732D, PDS-734/734D, PDS-743/743D, PDS-762/762D. The DI is 0~30VDC wide range digital Input, while the DO is 30V/100mA (max.), current sink, open collector digital output. Users can use these digital I/O lines to control relay, actuator, switch… etc.

6.1 Test the virtual I/O

1. The PDS-700 has been connected to Ethernet, finished the configuration setup and passed the Virtual COM test as described in chapter 4.
2. Power-on the PDS-700.
3. Wire the DO(n) to DI(n).

   For instance: the PDS-734 with 4-port DI/DO

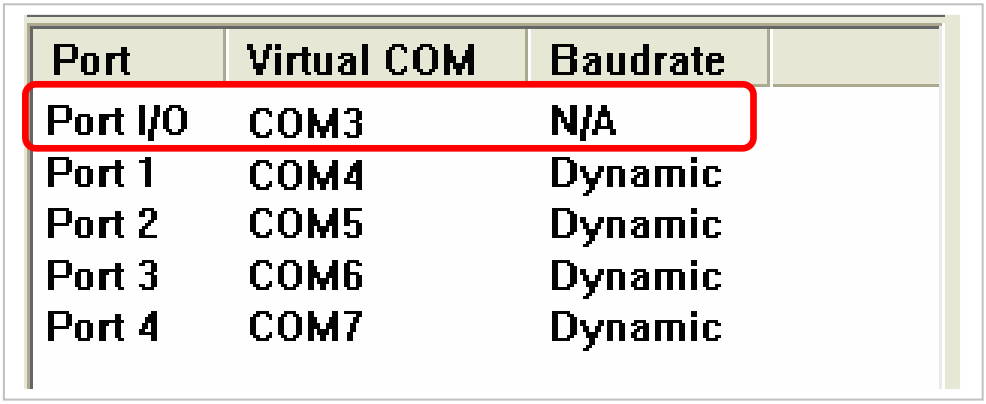


Connect the DO0 to DI0, DO1 to DI1, DO2 to DI2, DO3 to DI3.

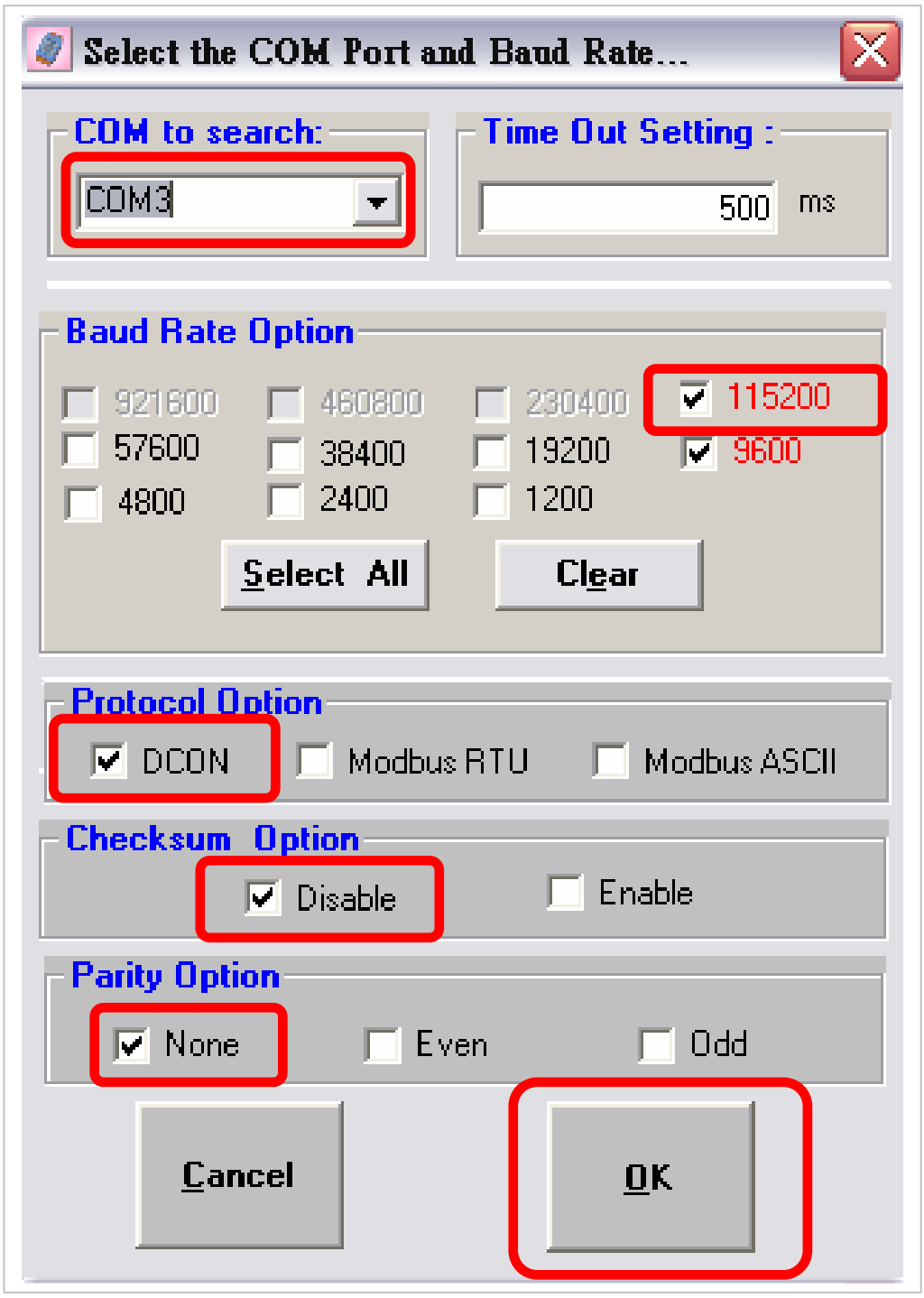
4. Install DCON Utility v4.5.0 (or later).
   DCON Utility is located at
   CD:\Napdos\driver\dcon_utility
   http://ftp.icpdas.com/pub/cd/8000cd/napdos/driver/dcon_utility/setup/

5. Run DCON Utility, click “COM Port” on the toolbar

6. Check the COM port number virtualized by Port I/O in VxComm Utility. (refer to chapter 4)

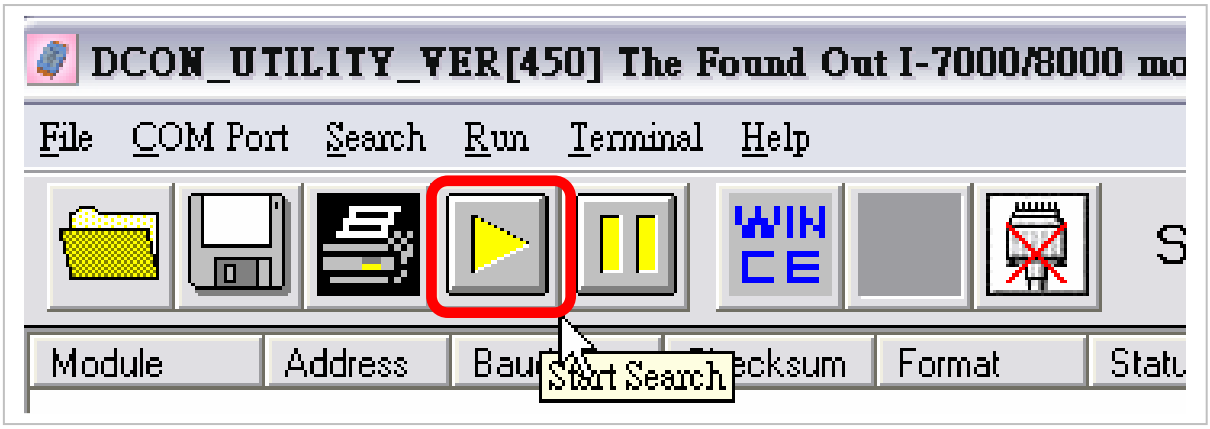


7. Select the COM port number that the port I/O virtualized. Check 115200 (baudrate), DCON (protocol), checksum disable, none parity, and press “OK” button.

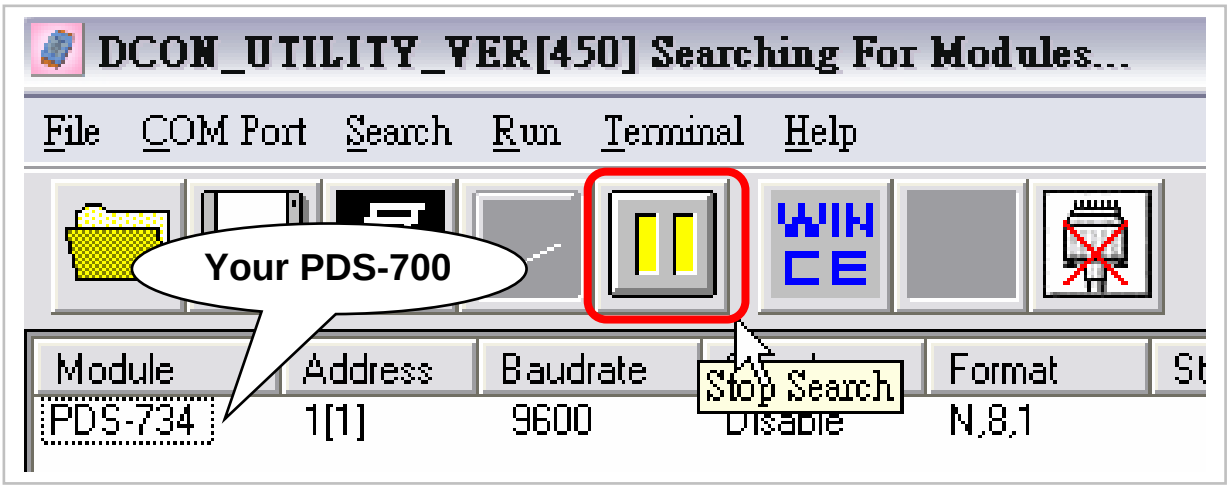


*If your PDS-700 does not equip digital I/O lines, the DCON Utility will return “**Open COM error!**”.*

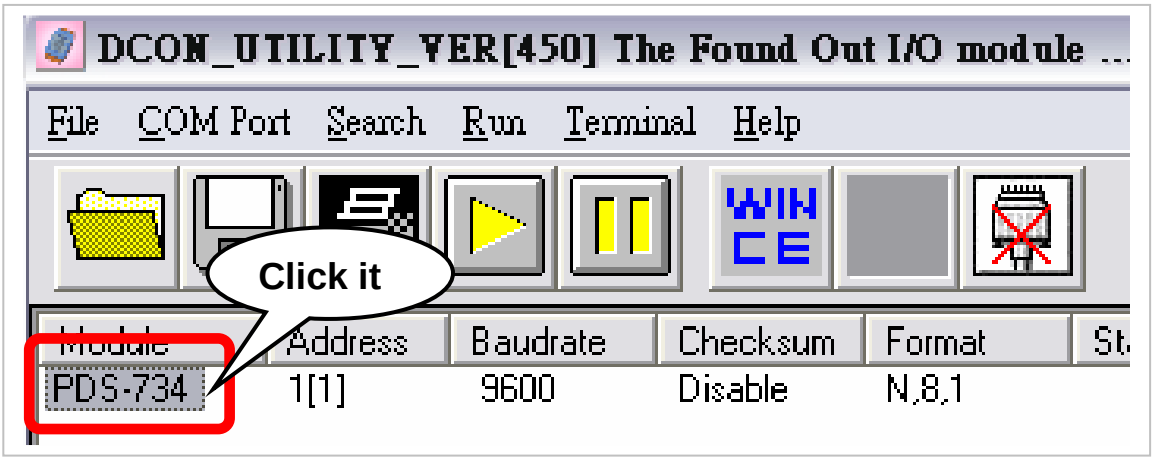
8. Click  to start searching the PDS-700



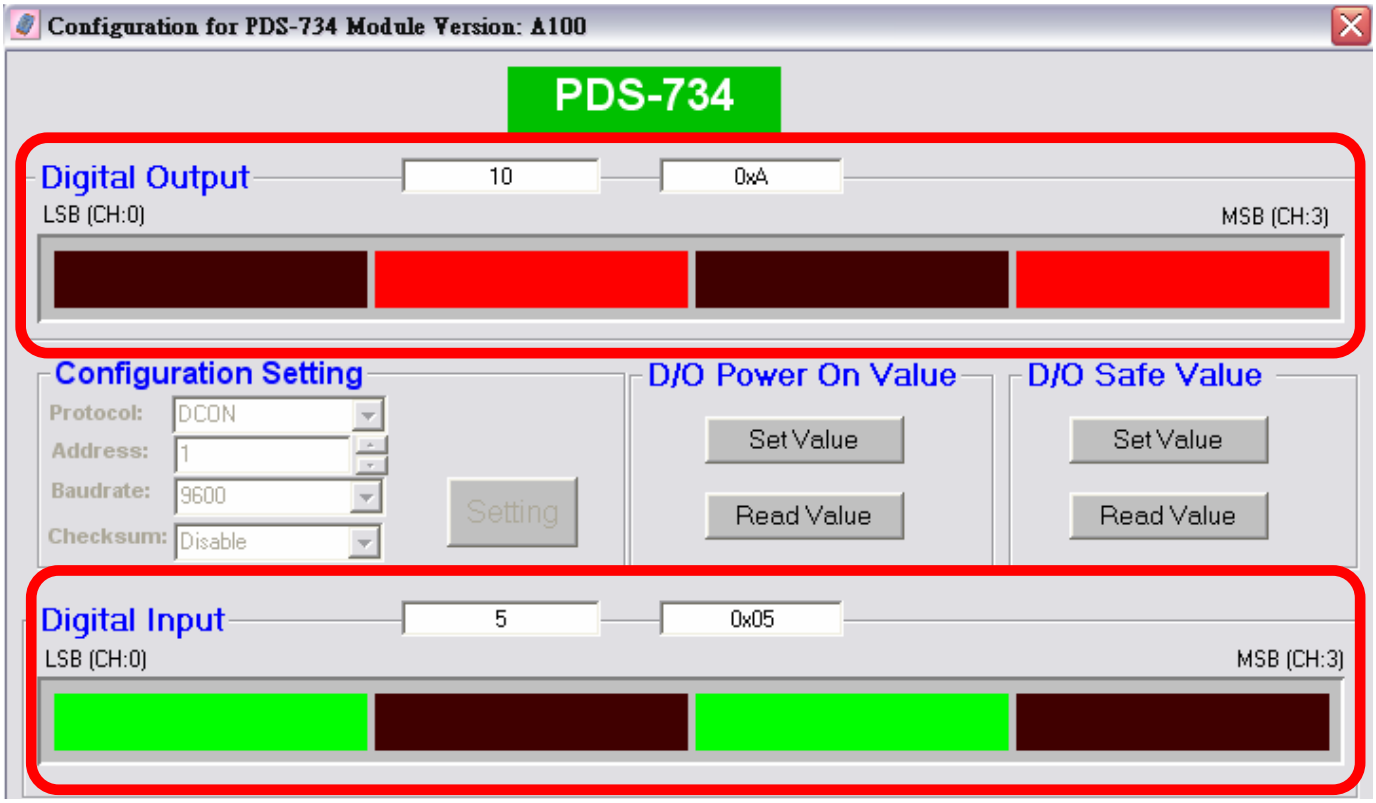
9. When the PDS-700 is searched and showed in the DCON Utility, click  to stop the search



10. Click your PDS-700.



11. Click the “**Digital Output**” icon to change the DO high/low status.



Since all DI lines are connected to DO lines, the DI read will be 0 when the DO sends a high state, whereas the DI read will be 1.

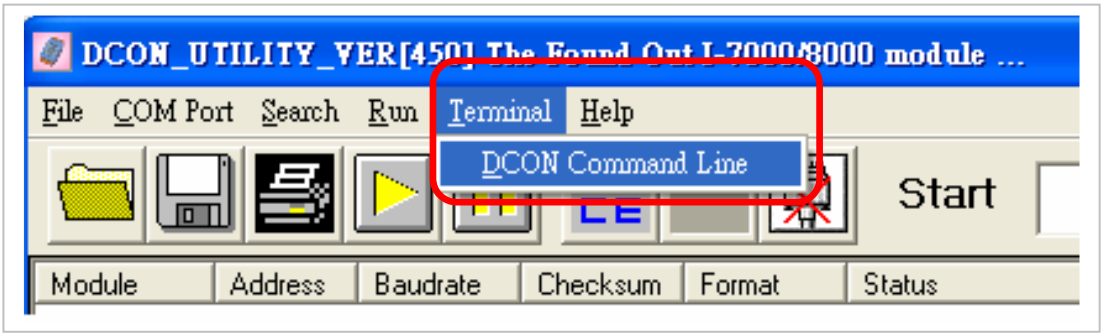
6.2 Virtual I/O commands test

DCON protocol is a request/ reply communication protocol; it defines a simple ASCII format protocol, like $AAN, $AASi6, #AAN, .., etc. for accessing the PDS-700 and I-7000/8000/87K series I/O modules .

The Virtual I/O command sets are parts of DCON protocol for accessing the PDS-700 digital I/O lines from the virtualized COM port that I/O port mapped. Only the PDS-700 which equips digital I/O lines will reply the DCON request.

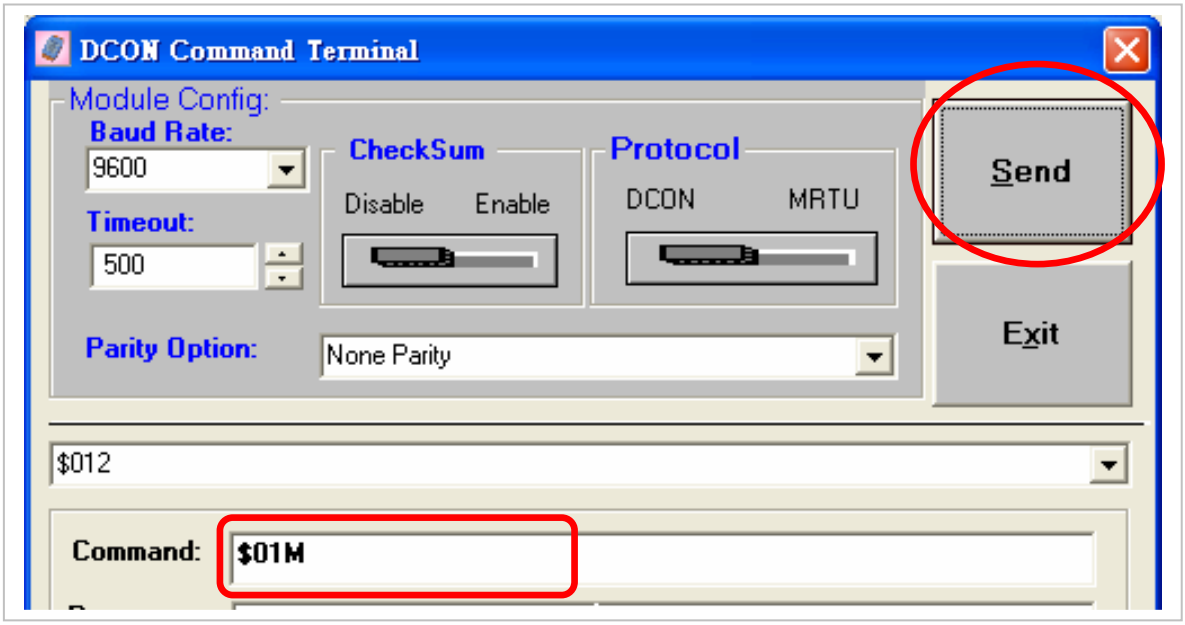
You can test the virtual I/O commands by DCON Utility:
(The DCON command sets are introduced in chapter 7)

1. Click “**Terminal**” >> “**DCON Command Line**”

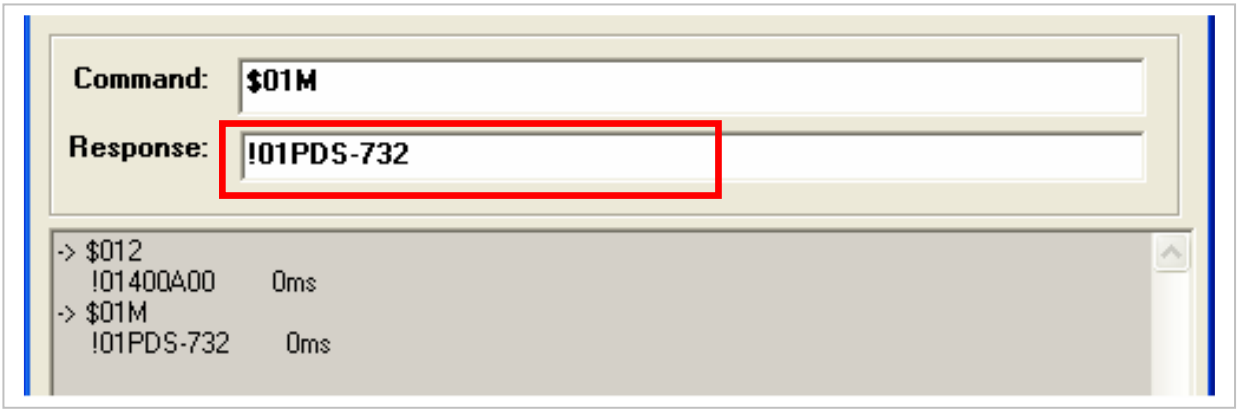


2. Type the virtual I/O command in the command column and press “**Send**” button to send the command

   For example, $01M is for reading the module name.



3. Get the response from the PDS-700 successful.



6.3 Program on PC client

General DCON Application Programming Interface kit is a set of DLL (lib) functions designed for running on Windows 98/2000/XP to access remote I/O modules such as PDS-700, i-7000, i-8000 and i-87k series.

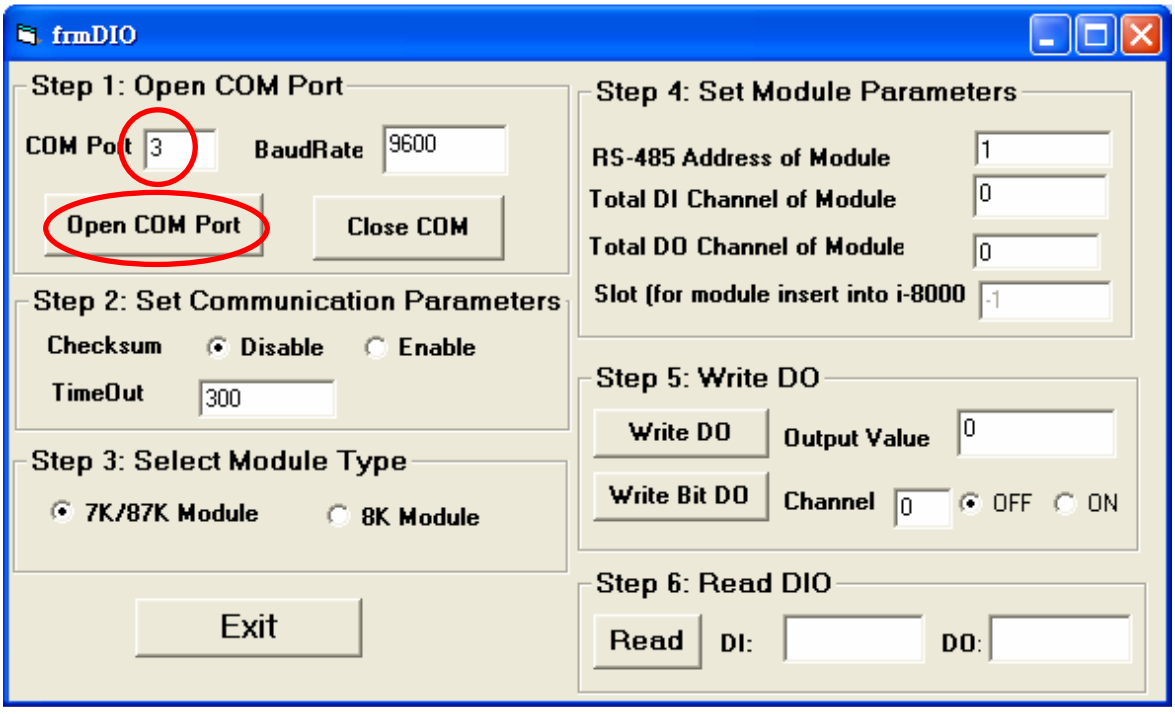
The General DCON API kit is located at:

- CD:\ napdos\ driver\ dcon_dll_new\
- ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/driver/dcon_dll_new/

The General DCON API kit provides VC and VB driver, VB demos and a document named "**dcon_fun_user_manual.pdf**". Only the dio demo in dcon_dll_new\demo\vb6 supports PDS-700 series. The following steps could test the general DCON API kit with the dio demo program.

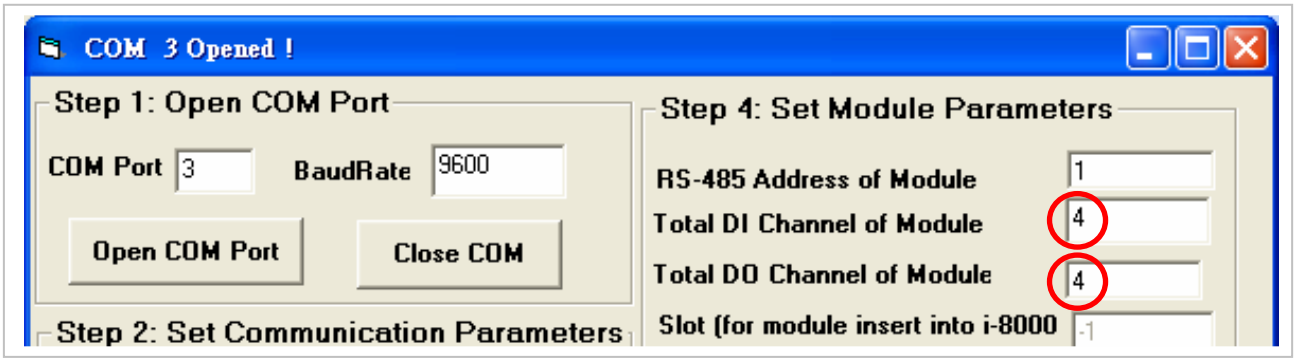
To run the dio demo, your PC must have VB6 installed.

1. Double click the "**prjdio.vbp**" to open the dio project.
2. Run the demo.
3. Set the COM port number virtualized by I/O port of the PDS-700 and click "**Open COM Port**" button.

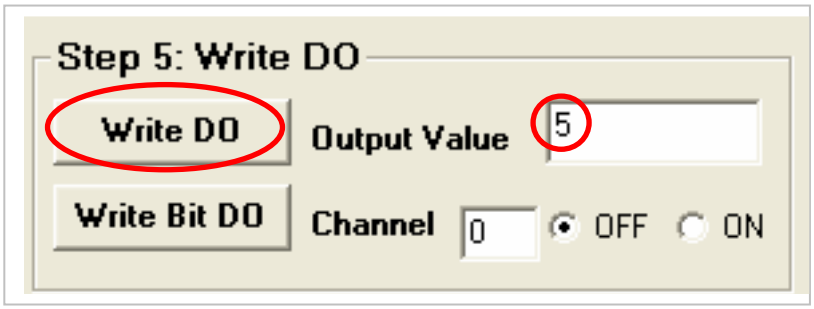


Response “**COM n Opened!**” on the title bar

4. Set the DI and DO total channel number on your PDS-700. For instance, the PDS-732 equips 4 DI channels and 4 DO channels.



5. Set the Output value and then Press “**Write DO**” button to send the data out.



6. Press “**Read**” button to get the DI data and DO read back.

7. Press “**Exit**” button to exit the program

To access the I/O lines on the PDS-700, the functions in General DCON API kit will be used:

Categorization	Dll and lib used	Call condition
Starting function: **Open_Com()**	Uart.dll Uart.lib	Call once when program starts.
I/O functions **DCON_Write_DO()** **DCON_Write_DO_Bit()** **DCON_Read_DIO()**	dcon_pc.dll dcon_pc.lib	Call the I/O functions for requirements
Communication **Send_Receive_Cmd()**	Uart.dll Uart.lib	Call the communication functions for requirements
Ending function **Close_Com()**	Uart.dll Uart.lib	Call once before program exit.

```
// DO program demo on PC client
void CManual1Dlg::OnOpen_Com()
{
     Open_Com(3,115200,8,0,1);
       //ComPort: 3, Baudrate:115200, DataBit:8, ParityBit: 0, StopBit: 1
}

void CManual1Dlg::OnClose_Com()
{
    Close_Com(3);  }

void CManual1Dlg::OnDigital_Out()
{
  iRet=DCON_Write_DO(3,1,-1,4,iDO_value,0,100);
     //Com port: 3, Address: 1, Slot: -1, total channel count:4, DO data,
     //Checksum: disable, Timeout: 100(ms)
}
```

Open_Com()

Description: open com port specified.
Syntax: Open_Com(unsigned char cPort,
DWORD dwBaudrate,
char cData,
char cParity,
char cStop);

Return:
0 : no error.
Others : error codes.

Parameters:
cPort: Com port number (1 ~ 255)
dwBaudrate: communication baudrate
cData: data bit, (8 for PDS-700)
cParity: 0 = none parity
cStop: 0 = 1 stop bit.

Close_Com ()

Description: close com port specified.
Syntax: Close_Com(unsigned char cPort)

Return:
0 : no error.
Others : error codes.

Parameters:
cPort: Com port number (1 ~ 255)

Send_Receive_Cmd ()

Description: send DCON command string and receive the response.
Syntax: Send_Receive_Cmd(unsigned char cPort,
char szCmd[],
char szResult[],
WORD wTimeOut,
WORD wChksum,
WORD *wT)

Return:

0 :no error.

Others : error codes.

Parameters:

cPort: Com port number (1 ~ 255)

szCmd[]: send string, 1024 Bytes maximum, without zero (0x0D) character.

szResult[]:result string recevied, 1024 Bytes maximum, with one zero or 0x0D terminal character.

wTimeOut: timeout for receiving result string. Unit: ms

wChksum: 0 ==> add one 0x0D byte to the end of the szCmd
<>0 ==> add two check sum bytes and one 0x0D byte to the end of the szCmd

*__wT__: return a reference number for identify the performance.

DCON_Write_DO ()

Description: send group digital output data to the PDS-700.
Syntax: DCON_Write_DO(unsigned char cComPort,
short iAddress,
short iSlot,
short iDO_TotalCh,
unsigned long lDO_Value,
short iCheckSum,
short iTimeOut);

Return:

0 : no error.

Others : error codes.

Parameters:

cComPort: COM port number, 1 to 255
iAddress: module address,**1** for the PDS-700
iSlot: **–1** for the PDS-700
iDO_TotalCh: total DO channel count on the PDS-700.
lDO_Value: digital output data
iCheckSum: 0: disable or 1: enable
iTimeout: time out setting, normal=100, unit: ms

DCON_Write_DO_Bit ()

Description: send bit digital output data to the PDS-700.
Syntax: DCON_Write_DO_Bit(unsigned char cComPort,
short iAddress,
short iSlot,
short iChannel,
short iDO_TotalCh,
short iBitValue,
short iCheckSum,
short iTimeOut);

Return:

0 : no error.
Others : error codes.

Parameters:

cComPort: COM port number, 1 to 255
iAddress: module address, **1** for the PDS-700
iSlot: **–1** for the PDS-700
iChannel: the digital output channel No.
iDO_TotalCh: total DO channel count on the PDS-700.
iBitValue: bit digital output data, 0 = off, 1 = on.
iCheckSum: 0: disable or 1: enable
iTimeout: time out setting, normal=100, unit: ms

DCON_Read_DIO ()

Description: read DO and DI lines status.
Syntax: DCON_Read_DIO(unsigned char cComPort,
short iAddress,
short iSlot,
short iDI_TotalCh,
short iDO_TotalCh,
short iCheckSum,
short iTimeOut,
unsigned long* lDI_Value,
unsigned long* lDO_Value,
char* cDI_BitValue,
char* cDO_BitValue);

Return:

0 : no error.
Others : error codes.

Input Parameter:

cComPort: COM port number, 1 to 255
iAddress: module address, **1** for the PDS-700
iSlot: **–1** for the PDS-700
iDI_TotalCh: total DI channel count on the PDS-700.
iDO_TotalCh: total DO channel count on the PDS-700.
iCheckSum: 0: disable or 1: enable
iTimeout: time out setting, normal=100, unit: ms
iDI_Value: read digital input data
iDO_Value: read digital output data
cDI_BitValue: read digital input data, Boolean array format.
cDO_BitValue: read digital output data, Boolean array format.

7. Virtual I/O Command

Command Format: **(Leading)(Address)(Command)[CHK](cr)**
Response Format: **(Leading)(Address)(Data)[CHK](cr)**

(Address): 01 (2-character)
[CHK]: 2-character checksum
(cr): end of command character, character return (0x0D)

Calculate Checksum:
1. Calculate ASCII sum of all characters of command (or response) string except the character return (cr).
2. Mask the sum of string with 0ffh

Example
Command string: $012(cr)
Sum of string = '$' + '0' + '1' + '2' = 24h+30h+31h+32h = B7h
The checksum is B7h, and **[CHK]** ="B7".
Command string with checksum: $012B7(cr)

Response string: !01300600(cr)
Sum of string = '!' + '0' + '1' + '3' + '0' + '0' + '6' + '0' + '0' = 21h + 30h + 31h + 33h + 30h + 30h + 36h + 30h + 30h = 1ABh
The checksum is ABh, and **[CHK]** ="AB".
Response string with checksum: !01300600AB(cr)

General Command Sets			
Command	Response	Description	Section
$AA5	!AAS	Read Reset Status	Sec.5.1
$AA6	!AA(Data)	Read Digital I/O Status	Sec.5.2
$AAC	!AA	Clear Latched Digital Input	Sec.5.3
$AACn	!AA	Clear Digital Input Count	Sec.5.4
$AAGCN	>AA(Data)	Get I/O Channel Count	Sec.5.5
$AALs	!(Data)	Read Latched DI	Sec.5.6
$AAF	!AA(Data)	Read Firmware Version	Sec.5.7
$AAM	!AA(Data)	Read Module Name	Sec.5.8
@AA	>(Data)	Read Digital Input/Output Status	Sec.5.9
@AA(Data)	>	Set Digital Output	Sec.5.10
#AAn	!AA(Data)	Read DI counter	Sec.5.11
#AA00dd	>	Set Multi-channel Output	Sec.5.12
#AA1n0s	>	Set Single Channel Output	Sec.5.13

Host Watchdog Command Sets			
~**	No Reponse	Host OK	Sec.5.14
~AA0	!AASS	Read Module Status	Sec.5.15
~AA1	!AA	Reset Module Status	Sec.5.16
~AA2	!AAeff	Read Host Watchdog Timeout Value	Sec.5.17
~AA3eff	!AA	Set Host Watchdog Timeout Value	Sec.5.18
~AA4P	!AA(Data)	Read Power-on Value of D/O	Sec.5.19
~AA4S	!AA(Data)	Read Safe Value of D/O	Sec.5.20
~AA5P	!AA	Set Power-on Value of D/O	Sec.5.21
~AA5S	!AA	Set Safe Value of D/O	Sec.5.22

7.1 $AA5

Description: Read Reset Status

Syntax: $AA5[CHK](cr)

$ delimiter character

AA address of the module (01 only for PDS-700)

5 command for reading status

Response: Valid Command: **!AAs[CHK](cr)**

Invalid Command: **?AA[CHK](cr)**

Syntax error or communication error may get no response.

! delimiter for valid command

? delimiter for invalid command

AA address of the module (01 only for PDS-700)

s reset status

1 = the module had been reset, and the status is reset to 0 after the $AA5 command used.

0 = the module has not ever reset

Example:

Command: $015 Response: !011

Read reset status, return first read

Command: $015 Response: !011

Read reset status, return no reset occurred

Related Topics:

Sec Reset Status

7.2 $AA6

Description: Read Digital I/O Status

Syntax: $AA6[CHK](cr)

$ delimiter character

AA address of the module (01 only for PDS-700)

6 command for reading digital I/O status

Response: Valid Command: **!ddff00[CHK](cr)**

Invalid Command: **?AA[CHK](cr)**

Syntax error or communication error may get no response.

! delimiter for valid command

? delimiter for invalid command

AA address of the module (01 only for PDS-700)

dd the current digital output status

ff the digital input status

Example:

Command: $016 Response: !0F0000

Read digital input/output status, return F000;
digital output channel 3 to 0 are on status,
digital input status are all off status.

Related Command:

Sec.5.9 @AA

7.3 $AAC

Description: Clear Latched Digital Input

Syntax: $AAC[CHK](cr)

$ delimiter character

AA address of the module (01 only for PDS-700)

C command for clearing latched digital inputs

Response: Valid Command: **!AA[CHK](cr)**

Invalid Command: **?AA[CHK](cr)**

Syntax error or communication error may get no response.

! delimiter for valid command

? delimiter for invalid command

AA address of the module (01 only for PDS-700)

Example:

Command: $01L0 Response: !FFFF00

Read latch-low data, return FFFF.

Command: $01C Response: !01

Clear latched digital inputs, return success.

Command: $01L0 Response: !000000

Read latch-low data, return 0000.

Related Command:

Sec.5.6 $AALs

7.4 $AACn

Description: Clear Digital Input Counter

Syntax: $AACn[CHK](cr)

$ delimiter character

AA address of the module (01 only for PDS-700)

C command for clearing digital input count

n the digital input channel number

Response: Valid Command: **!AA[CHK](cr)**

Invalid Command: **?AA[CHK](cr)**

Syntax error or communication error may get no response.

! delimiter for valid command

? delimiter for invalid command

AA address of the module (01 only for PDS-700)

Example:

Command: #010 Response: !0100123

Read counter value on digital input channel 0, return 123.

Command: $01C0 Response: !01

Clear counter value on digital input channel 0, return success.

Command: #010 Response: !0100123

Read counter value on digital input channel 0, return 0.

Related Command:

Sec.5.11 #AAn

7.5 $AAGCN

Description: Read Digital Input/Output Channel Count

Syntax: $AAGCN[CHK](cr)

$ delimiter character

AA address of the module (01 only for PDS-700)

GCN command for reading digital input/output channel count

Response: Valid Command: **>DINxxDONxx[CHK](cr)**

Valid Command: **>DONxx[CHK](cr)** (DO only)

Invalid Command: **?AA[CHK](cr)**

Syntax error or communication error may get no response.

> delimiter for valid command

? delimiter for invalid command

AA address of the module (01 only for PDS-700)

DINxx DIN: digital input channel, xx: the total channel count

DONxx DON: digital output channel, xx: the total channel count

Example:

Command: $01GCN Response: >DIN01DON02

Read the total I/O channel count on the module, return module equipped with 1 digital input channel and 2 digital output channels.

7.6 $AALs

Description: Read Latched Digital Input

Syntax: $AALs[CHK](cr)

$ delimiter character

AA address of the module (01 only for PDS-700)

L command for reading latched digital input data

s 1 = read latch-high data, 0 = read latch-low data

Response: Valid Command: **!(Data)[CHK](cr)**

Invalid Command: **?AA[CHK](cr)**

Syntax error or communication error may get no response.

! delimiter for valid command

? delimiter for invalid command

AA address of the module (01 only for PDS-700)

(Data) read data

1 = the input channel is latched

0 = the input channel is not latched

Example:

Command: $01L1 Response: !FF0000

Read latch-high data, return FF00, channel 7 to channel 0 are all latched. (Module with DI channel count between 5 to 8)

Command: $01L1 Response: !F00000

Read latch-high data, return FF00, channel 3 to channel 0 are all latched. (Module with DI channel count between 1to 4)

Related Command:

Sec.5.3 $AAC

7.7 $AAF

Description: Read Firmware Version

Syntax: $AAF[CHK](cr)

$ delimiter character

AA address of the module (01 only for PDS-700)

F command for reading firmware version

Response: Valid Command: **!AA(Data)[CHK](cr)**

Invalid Command: **?AA[CHK](cr)**

Syntax error or communication error may get no response.

! delimiter for valid command

? delimiter for invalid command

AA address of the module (01 only for PDS-700)

(Data) firmware version

Example:

Command: $01F Response: !01A1.00

Read firmware version, return version A1.00

7.8 $AAM

Description: Read Module Name

Syntax: $AAM[CHK](cr)

$ delimiter character

AA address of the module (01 only for PDS-700)

M command for reading module name

Response: Valid Command: **!AA(Data)[CHK](cr)**

Invalid Command: **?AA[CHK](cr)**

Syntax error or communication error may get no response.

! delimiter for valid command

? delimiter for invalid command

AA address of the module (01 only for PDS-700)

(Data) name of module

Example:

Command: $01M Response: !01PDS-721

Read module name, return name PDS-721

7.9 @AA

Description: Read Digital Input/Output Status

Syntax: @AA[CHK](cr)

@ delimiter character

AA address of the module (01 only for PDS-700)

Response: Valid Command: **>(Data)[CHK](cr)**

Invalid Command: **?AA[CHK](cr)**

Syntax error or communication error may get no response.

> delimiter for valid command

? delimiter for invalid command

AA address of the module (01 only for PDS-700)

(Data) read DIO status

Example:

Command: @01 Response: >050F

Read DIO status, return 050F. The first two bytes are DO status, 05 means that channel0 and channel2 are 1, other channels are 0. The last two bytes are DIs status, 0F means that all the 4 channels read are 1.

7.10 @AA(Data)

Description: Set Digital Output

Syntax: @AA(Data)[CHK](cr)

@ delimiter character

AA address of the module (01 only for PDS-700)

(Data) output value

(Data) is one character for output channel less than 4

For PDS-762/762D, from 0 to 3

For PDS-732/732D/734/734D/743/743D, from 0 to F

(Data) is two characters for output channel less than 8

For PDS-721/721D, from 00 to 7F

Response: Valid Command: **>[CHK](cr)**

Invalid Command: **?[CHK](cr)**

Ignore Command: **![CHK](cr)**

Syntax error or communication error may get no response.

> delimiter for valid command

? delimiter for invalid command

! delimiter for ignore command. The module is in Host Watchdog Timeout Mode, and the output is set to safe value

Example:

Command: @013 Response: >

Output value 3, return success

Command: @011F Response: !

Output value 1F, return the module is in host watchdog timeout and the output command is ignored

7.11 #AAn

Description: Read Digital Input Counter from Channel n

Syntax: #AAn[CHK](cr)

delimiter character

AA address of the module (01 only for PDS-700)

n the digital channel number (from 0)

Response: Valid Command: **!AA(Data)[CHK](cr)**

Invalid Command: **?AA[CHK](cr)**

Syntax error or communication error may get no response.

! delimiter for valid command

? delimiter for invalid command

AA address of the module (01 only for PDS-700)

(Data) digital input counter value in decimal, from 00000 to 65535

Example:

Command: #012 Response: !0100103

Read digital input counter of channel 2, return value 103

Command: #013 Response: ?01

Read digital input counter of channel 3, return the channel is not available

Related Command:

Sec.5.4 $AACn

7.12 #AA00dd

Description: Set Multi-channel Output

Syntax: #AA00dd[CHK](cr)

delimiter character

AA address of the module (01 only for PDS-700)

00 command for setting multi-channel output

dd output value

Response: Valid Command: **>[CHK](cr)**

Invalid Command: **?[CHK](cr)**

Ignored Command: **![CHK](cr)**

Syntax error or communication error may get no response.

> delimiter for valid command

? delimiter for invalid command

! delimiter for ignore command. The module is in Host Watchdog Timeout Mode, and the output is set to safe value

Example:

Command: #01000F Response: >

Set digital output channel 3 to 0 are on, return success.

Command: #010005 Response: !

Set digital output channel 0 and 2 are on, return the module is in host watchdog timeout mode, and the output is set to save value.

Related Commands:

Sec5.10 @AA(Data), Sec.5.15 ~AA0, Sec.5.16 ~AA1

7.13 #AA1ndd

Description: Set Single Channel Output

Syntax: #AA1ndd[CHK](cr)

#	delimiter character
AA	address of the module (01 only for PDS-700)
1n	command for setting single channel output n is the digital output channel number
dd	00: set the digital output channel off 01: set the digital output channel on

Response: Valid Command: **>[CHK](cr)**

Invalid Command: **?[CHK](cr)**

Ignored Command: **![CHK](cr)**

Syntax error or communication error may get no response.

>	delimiter for valid command
?	delimiter for invalid command
!	delimiter for ignore command. The module is in Host Watchdog Timeout Mode, and the output is set to safe value

Example:

Command: #011201 Response: >

Set digital output channel 2 on, return success.

Related Commands:

Sec5.10 @AA(Data), Sec.5.15 ~AA0, Sec.5.16 ~AA1

7.14 ~**

Description: Host OK

Host sends this command to all modules for sending the information “Host OK”

Syntax: ~**[CHK](cr)

~ delimiter character

** command for all modules

Response: No response

Example:

Command: ~** No response

Related Commands:

Sec.5.15 ~AA0, Sec.5.16 ~AA1, Sec.5.17 ~AA2, Sec5.18 ~AA3eff, Sec.5.19 ~AA4P, Sec.5.20 ~AA4S, Sec.5.21 ~AA5P, Sec.5.22 ~AA5S

7.15 ~AA0

Description: Read Module Status

Syntax: ~AA0[CHK](cr)

~ delimiter character

AA address of the module (01 only for PDS-700)

0 command for reading module status

Response: Valid Command: **!AASS[CHK](cr)**

Invalid Command: **?AA[CHK](cr)**

Syntax error or communication error may get no response.

! delimiter for valid command

? delimiter for invalid command

AA address of the module (01 only for PDS-700)

SS module status

00 = host watchdog timeout status is clear

04 = host watchdog timeout status is set. The status will store into EEPROM and only may be set by the command ~AA1

Example:

Sec 5.18 ~AA3eff example

Related Commands:

Sec.5.15 ~AA0, Sec.5.16 ~AA1, Sec.5.17 ~AA2, Sec5.18 ~AA3eff, Sec.5.19 ~AA4P, Sec.5.20 ~AA4S, Sec.5.21 ~AA5P, Sec.5.22 ~AA5S

7.16 ~AA1

Description: Reset Module Status

Syntax: ~AA1[CHK](cr)

~ delimiter character

AA address of the module (01 only for PDS-700)

1 command for resetting module status

Response: Valid Command: **!AA[CHK](cr)**

Invalid Command: **?AA[CHK](cr)**

Syntax error or communication error may get no response.

! delimiter for valid command

? delimiter for invalid command

AA address of the module (01 only for PDS-700)

Example:

Sec 5.18 ~AA3eff example

Related Commands:

Sec.5.15 ~AA0, Sec.5.16 ~AA1, Sec.5.17 ~AA2, Sec5.18 ~AA3eff, Sec.5.19 ~AA4P, Sec.5.20 ~AA4S, Sec.5.21 ~AA5P, Sec.5.22 ~AA5S

7.17 ~AA2

Description: Read Host Watchdog Timeout Value

Syntax: ~AA2[CHK](cr)

~ delimiter character

AA address of the module (01 only for PDS-700)

2 command for reading host watchdog timeout value

Response: Valid Command: **!AAeff[CHK](cr)**

Invalid Command: **?AA[CHK](cr)**

Syntax error or communication error may get no response.

! delimiter for valid command

? delimiter for invalid command

AA address of the module (01 only for PDS-700)

e host watchdog enable status, 1 = Enable, 0 = Disable

ff timeout value in Hex format, the unit is 0.1 second,

01 = 0.1 second

FF = 25.5 second

Example:

Sec 5.18 ~AA3eff example

Related Commands:

Sec.5.15 ~AA0, Sec.5.16 ~AA1, Sec.5.17 ~AA2, Sec5.18 ~AA3eff, Sec.5.19 ~AA4P, Sec.5.20 ~AA4S, Sec.5.21 ~AA5P, Sec.5.22 ~AA5S

7.18 ~AA3eff

Description: Set Host Watchdog Timeout Value

Syntax: ~AA3eff[CHK](cr)

~ delimiter character

AA address of the module (01 only for PDS-700)

3 command for setting host watchdog timeout value

e 1 = Enable / 0 = Disable host watchdog

ff timeout value, from 01 to FF, the unit is 0.1 second

Response: Valid Command: **!AA[CHK](cr)**

Invalid Command: **?AA[CHK](cr)**

Syntax error or communication error may get no response.

! delimiter for valid command

? delimiter for invalid command

AA address of the module (01 only for PDS-700)

Example:

Command: ~010 Response: !0100

Read module status, return host watchdog timeout status is clear.

Command: ~013164 Response: !01

Set host watchdog timeout value 10.0 seconds and enable host watchdog, return success.

Command: ~012 Response: !01164

Read host watchdog timeout value, return host watchdog timeout value is 10.0 senonds, and the host watchdog is enabled.

Command: ~** No response

Reset the host watchdog timer.

Wait more than 10 seconds and don't send command ~**, the LED of module will start to flash. The LED indicates the host watchdog timeout status is set.

Command: ~010 Response: !0104

Read module status, return host watchdog timeout status is set.

Command: ~012 Response: !01064

Read host watchdog timeout value, return host watchdog timeout value is 10.0 senonds, and the host watchdog is disabled.

Command: ~011 Response: !01

Reset host watchdog timeout status, return success.
And the LED stops flash.

Command: ~010 Response: !0100

Read module status, return host watchdog timeout status is clear.

Related Commands:

Sec.5.15 ~AA0, Sec.5.16 ~AA1, Sec.5.17 ~AA2, Sec5.18 ~AA3eff, Sec.5.19 ~AA4P, Sec.5.20 ~AA4S, Sec.5.21 ~AA5P, Sec.5.22 ~AA5S

7.19 ~AA4P

Description: Read Power-on Value of DO

Syntax: ~AA4P[CHK](cr)

~ delimiter character

AA address of the module (01 only for PDS-700)

4P command for reading power-on value of DO

Response: Valid Command: **!AA(Data)[CHK](cr)**

Invalid Command: **?AA[CHK](cr)**

Syntax error or communication error may get no response.

! delimiter for valid command

? delimiter for invalid command

AA address of the module (01 only for PDS-700)

(Data) power-on value

Example:

Command: ~014P Response: !01000F

Read power-on value, return power-on value 0F

Related Command:

Sec.5.21 ~AA5P

7.20 ~AA4S

Description: Read Safe Value of DO

Syntax: ~AA4S[CHK](cr)

~ delimiter character

AA address of the module (01 only for PDS-700)

4S command for reading safe value of DO

Response: Valid Command: **!AA(Data)[CHK](cr)**

Invalid Command: **?AA[CHK](cr)**

Syntax error or communication error may get no response.

! delimiter for valid command

? delimiter for invalid command

AA address of the module (01 only for PDS-700)

(Data) power-on value

Example:

Command: ~014S Response: !01000F

Read safe value, return power-on value 0F

Related Command:

Sec.5.22 ~AA5S

7.21 ~AA5P

Description: Set Power-on Value of DO

Syntax: ~AA5P[CHK](cr)

~ delimiter character

AA address of the module (01 only for PDS-700)

5P command for setting power-on value of DO

Response: Valid Command: **!AA[CHK](cr)**

Invalid Command: **?AA[CHK](cr)**

Syntax error or communication error may get no response.

! delimiter for valid command

? delimiter for invalid command

AA address of the module (01 only for PDS-700)

Example:

Command: @0103 Response: >

Output value 03, return success

Command: ~015P Response: !01

Set the current output status as power-on value, return success.

Related Command:

Sec.5.19 ~AA4P

7.22 ~AA5S

Description: Set Safe Value of DO

Syntax: ~AA5S[CHK](cr)

~ delimiter character

AA address of the module (01 only for PDS-700)

5S command for setting safe value of DO

Response: Valid Command: **!AA[CHK](cr)**

Invalid Command: **?AA[CHK](cr)**

Syntax error or communication error may get no response.

! delimiter for valid command

? delimiter for invalid command

AA address of the module (01 only for PDS-700)

Example:

Command: @013 Response: >

Output value 03, return success

Command: ~015S Response: !01

Set the current output status as safe value, return success.

Related Command:

Sec.5.20 ~AA4S

7.23 Application Note

Module Status

PowerOn Reset or Module Watchdog Reset will let all output go to PowerOn Value. And the module may accept the host's command to change the output value.

Host Watchdog Timeout will let all output go to Safe Value. The module's status (read by command ~AA0) will be 04, and the output command will be ignored.

Dual Watchdog Operation

Dual Watchdog = Module Watchdog + Host Watchdog

The Module Watchdog is a hardware reset circuit to monitor the module's operating status. While working in harsh or noisy environment, the module may be shutdown by the external signal. The circuit may let the module to work continues and never halt.

The Host Watchdog is a software function to monitor the host's operating status. Its purpose is to prevent the network from communication problem or host halt. When the timeout interval expired, the module will turn all outputs to predefined Safe Value. This can prevent the controlled target from unexpected situation.

The PDS-700 series with Dual Watchdog may let the control system more reliable and stable.

Reset Status

The Reset Status is set while the module power on or reset by Module Watchdog, and is cleared while the command read Reset Status ($AA5) applied. This is useful for user to check the module's working status. When the Reset Status is set means the module is reset and the output may be changed to the PowerOn Value. When the Reset Status is clear means the module is not reset, and the output is not changed.

Digital Output

The module's output status has 3 different situations:

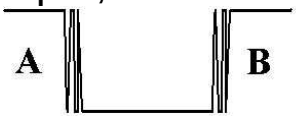
<1> Safe Value: If the host watchdog timeout status is set, the output is set to Safe value. While the module receive the output command, like @AA(Data) or #AABBDD, the module will ignore the command and return '!', and will not change the output to the output command value. The host watchdog timeout status is set and store into EEPROM while the host watchdog timeout interval expired, and only can be cleared by command ~AA1.
If user wants to change the output, he needs to clear the host watchdog timeout status firstly, and send output command to change the output into desired value.

<2> PowerOn Value: While the module reset, and the host watchdog timeout status is clear, the module's output is set to predefined PowerOn Value.

<3> Output command value: If the host watchdog timeout status is clear, and user issue a digital output command, like @AA (Data) or #AABBDD, to module for changing the output value. The module will response success (receive >).

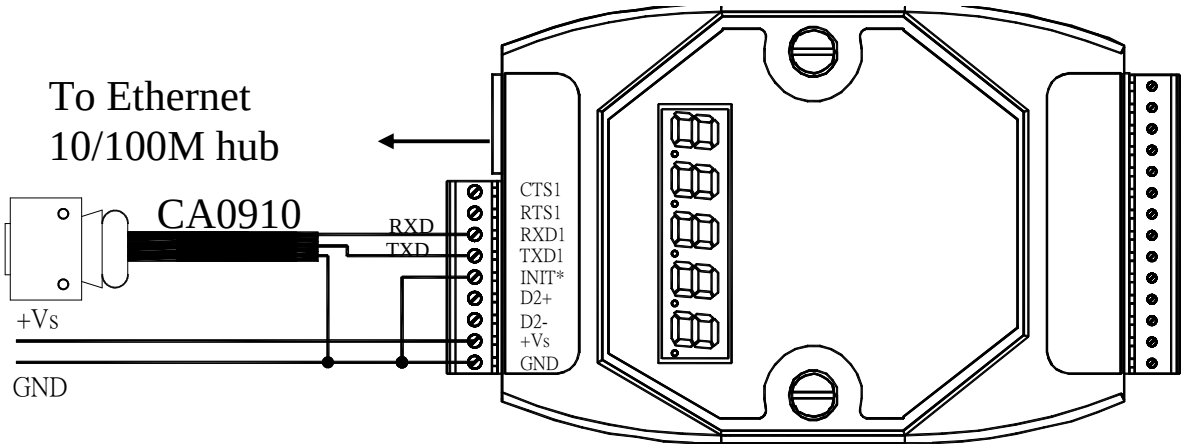
Latch Digital Input

For example, while the user connects the key switch to digital input channel and want to read the key stoke. The key input is a pulse digital input, and user will lost the



strike. While reading by command $AA6 in A and B position, the response is that no key stroke and he will lose the key stroke information. Respect, the read latch-low digital input command $AAL0 will solve this problem. When issue $AAL0 command in A and B position, the response denote that there is a low pulse between A and B position for a key stroke.

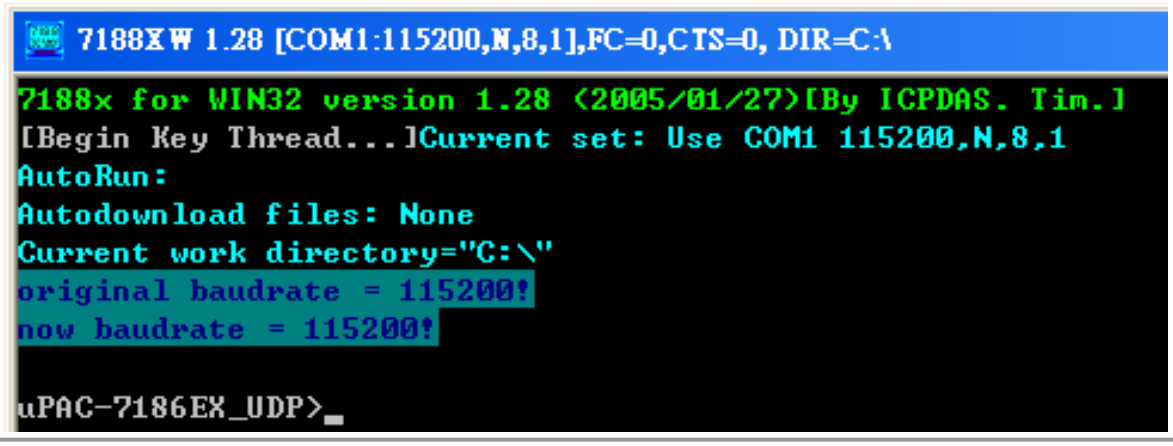
Appendix A: Linking to Program-development PC



Step 1: Connect download-cable, CA0910, to the PDS-700 as above diagram and COM1 (or COM2) of program-development PC.
Step 2: Connect INIT*-pin to GND-pin as shown in the above diagram.
Step 3: Unzip “7188XW_yyyymmdd.zip” on PC, the file is located at CD:\Napdos\MiniOS7\utility.
Step 4: Apply power (+Vs, GND) to the PDS-700, +Vs can be anywhere from +30V to +10V.
Step 5: Checking the 5-digits of the 7-SEG LED will continuously show as follows:

Hour.Minute.Second

Note: Only with display version modules have 5-digits of 7-SEG LED.
Step 6: Execute 7188XW.EXE /C#, and change baud rate to 115200, N81. “/C#” is the COM port of the program-development PC.
Step 7: Press [Enter] twice in the program-development PC as follows:



Step 8: Read configuration of the PDS-700 as follows:

```
uPAC-7186EX_UDP>ip
IP=10.0.8.20
uPAC-7186EX_UDP>mask
MASK=255.255.255.0
uPAC-7186EX_UDP>gateway
Gateway=10.0.8.254
uPAC-7186EX_UDP>mac
Ethernet Address = 00:0d:e0:20:00:07
uPAC-7186EX_UDP>setcom 1
Current set is: 9600,8,0,1
```

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

Note: You can change the configuration of the PDS-700 as follows:

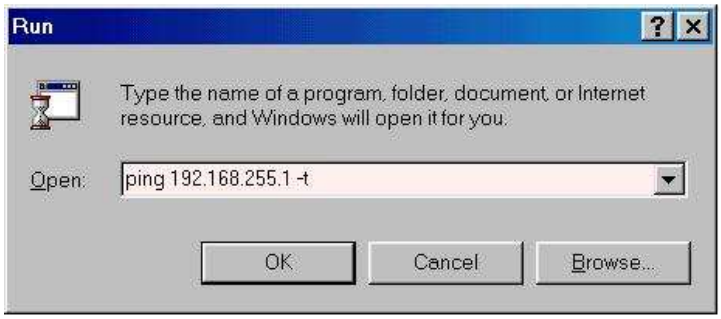
```
uPAC-7186EX_UDP>ip 192.168.41.1
Set IP=192.168.41.1
[ReadBack]IP=192.168.41.1
uPAC-7186EX_UDP>mask 255.255.255.0
Set MASK=255.255.255.0
[ReadBack]MASK=255.255.255.0
uPAC-7186EX_UDP>gateway 192.168.41.4
Set GATEWAY=192.168.41.4
[ReadBack]Gateway=192.168.41.4
uPAC-7186EX_UDP>setcom 1 115200,n,8,1
Current set is: 9600,8,0,1
Set to: 115200,8,0,1 [checksum:CC]
```

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

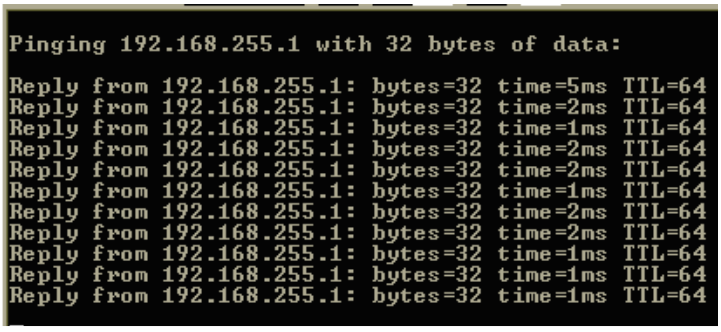
Parameters of **"setcom**" are as follows:

port: 1-8
baud: 2-921600
databit:
7, 8: for COM1 and COM2
5,6,7,8: for COM3 ~ COM8
parity:
N, n: None parity
E, e: Even parity
O, o: Odd parity
M, m: Mark, parity=1
S, s: Space, parity=0
stopbit:
1: for COM1, COM2
1, 2: for COM3 ~ COM8

Step 9: Disconnect INIT*-pin form GND-pin.
Step 10: Power off then power on.
Step 11: Execute **ping 192.168.255.1 –t** as follows:



Execute Ping 192.168.255.1 in PC client.



Ping results must be smooth & continuous

Note:

- **192.168.255.1** is the default IP of the PDS-700. Users can change IP (step 8).
- If the PC cannot ping the PDS-700 successfully, please refer to step 8 to change the configuration of the PDS-700. (the mask and gateway of the PDS-700 and PC both need meet the network definition.
- The MAC address of the PDS-700 should be unique in the same network. Refer to step 8 for changing the MAC address of the PDS-700.
- Every MAC address of the PDS-700 is unique in the default shipping.

In general, if host-PC can ping the PDS-700 smoothly and continuously, all other software and drivers for the PDS-700 will work fine. Therefore, users should make sure the program-development PC can ping the PDS-700 smoothly before any further testing.

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 technology today. Ethernet is a best-effort delivery system that uses CSMA/CD technology. It recognizes hosts using 48-bit MAC address.

2. Internet:

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

3. TCP/IP:

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

4. TCP (Transmission Control Protocol):

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

5. UDP (User Datagram Protocol):

UDP provides a much simpler service to the application layer. It just sends packets of data from one host to the other. But there is no guarantee that the packets reach the destination host.

6. Gateway:

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

7. IP (Internet Protocol) address:

Every interface on an Internet must have a unique IP address (also called an Internet address). These addresses are 32-bit numbers. They are normally written as four decimal 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 Ethernet interface hardware. Thus, no two hardware interfaces have the same MAC address.

9. Subnet Mask:

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

10. ARP (Address Resolution Protocol):

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

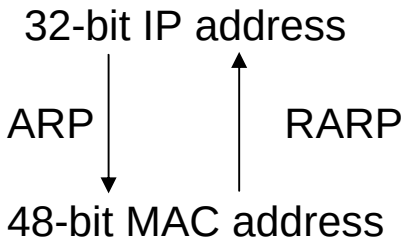
by physical networks using whatever MAC address scheme the hardware supplies.

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

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

11. RARP (Reverse Address Resolution Protocol):

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



12. ICMP (Internet Control Messages Protocol):

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

13. Ping:

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

14. Packet:

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

15. Socket:

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

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

16. Clients and Servers:

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

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

17. Firmware:

Alterable programs in the semi permanent storage, e.g., ROM, EEPROM, or Flash memory.