I-7521/I-7522/I-7522A/I-7523/I-7524 /I-7527 User’s Manual

Warranty

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

Warning

ICP DAS assume no liability for damages 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.

World Wide Web Home Page and FTP Site

To access ICPDAS home page go to http://www.icpdas.com

To download document and software go to http://www.icpdas.com/download/752n.htm or ftp://ftp.icpdas.com.tw/pub/cd/8000cd/napdos/752n/

Technical Support

Questions and requests can be sent via E-mail to service@icpdas.com

TABLE OF COMTENTS

Chapter 1 Introduction

Nowadays, a great number of RS-232 devices for both automation and information transfer are being used in industrial applications, and linking these devices is very important in the modern situation. The devices are usually located at a distance from the Host PC, meaning that linking via multiple serial cards is inefficient. ICPDAS I-752N series products have been developed to provide an effective link between multiple RS-232 devices via a single RS-485 network. This network protocol offers stability, reliability and simple cabling while delivering a low–cost, easy-to-maintain product.

1.1. Overview

Addressable RS-232 Converter

Most RS-232 devices don’t support individual device addressing. To overcome this limitation, ICPDAS I-752N series modules assign a unique address to any RS-232 device installed on an RS-485 network. When the Host PC sends a command to the RS-485 network a device address can be attached to the command. The destination I-752N module will then remove the address field and pass the remaining commands to the destination RS-232 device. Responses from the local RS-232 devices will be returned to the Host PC via the I-752N module.

Responses from RS-232 devices can be addressable

ICPADS I-752N series modules can prefix a response from an RS-232 device with a unique address and then pass the response to the RS-485 network. The Host PC can then identify which RS-232 device the response comes from.

Master-type Addressable RS-232 Converter

ICPDAS I-752N products are unique. In that they are Master-type converters, while most other converters are Slave-type, which are helpless without a Host PC. In real industrial applications, many users are not satisfied with Slave-type converters as they cannot be adapted to individual requirement. The powerful I-752N series analyzes the DI / DO of local RS-232 devices without the need for a Host PC. Refer to Applications 7~9 in Sec.6.2 for more information.

Onboard 1Kb Queue buffer

I-752N series modules are equipped while a 1Kb queue buffer for its local RS-232 device. All input data can be stored in the queue buffer until the Host PC has time to read it. These features allow the Host PC to be linked to thousands of RS-232 devices without any loss of data. (Refer to Sec.5.3.30)

Onboard DI signal trigger

I-752N series modules are equipped with 1/2/5 DI channels for sensor interfacing. These DI channels are linked to a photo sensor/switch to act as a signal. They also can be used as general purpose DI. I-752N series modules provide ODM and applicatin demos for a user to modify them to read and analyze these DI signals without the need for a Host PC.

Onboard DO channels for emergency control

I-752N series modules equipped with 1/3/5 DO channels for emergency control. The DO channels can directly drive either relay or an LED, and can be used to control the local devices in the event of an emergency. I-752N series modules provide ODM and applicatin demos for a user to modify them to control these DO channels without the need for a Host PC.

3000V isolation on the RS-485 side

COM2 of the I-7521/I-7522/I-7523 modules is an isolated RS-485 port with 3000V isolation, which will protect the local RS-232 devices from transient noises coming from the RS-485 network.

Self-Tuner ASIC inside

The interned I-752N Self-Tuner ASIC for the RS-485 port can auto detect and control the send/receive direction of the RS-485 network, meaning that there is no need for application programs to be concerned about direction control of the RS-485 network.

A Wide range available for selection

	RS-232	RS-485	RS-232/ RS-485	RS-422/ RS-485	DI	DO
I-7521(D)		1	1		2	3
I-7522(D)	1	1	1		2	1
I-7523(D)	2	1	1		1	0
I-7522A(D)		1	1	1	5	5
I-7524(D)	3	1	1		1	1
I-7527(D)	6	1	1		1	1

Note: I-752N means any one of seven kinds of modules above.

Order Information

◆ Order item choice

Order item	Module name	O.S
I-7521(D)	Intelligent communication controller (1 channel of RS-232)	MiniOS7
I-7522(D)	Intelligent communication controller (2 channel of RS-232)	MiniOS7
I-7523(D)	Intelligent communication controller (3 channel of RS-232)	MiniOS7
I-7522A(D)	Intelligent communication controller (1 channel of RS-232 and 1 channel Of RS-422/485)	MiniOS7
I-7524(D)	Intelligent communication controller	MiniOS7

	(4 channel of RS-232)	
I-7527(D)	Intelligent communication controller (7 channel of RS-232)	MiniOS7
I-7XXX	I-7XXX without 7-SEG display	MiniOS7
I-7XXXD	I-7XXX with 7-SEG display	MiniOS7

◆Package List

The package includes the following item:

- One I-7521/I-7522/I-7522A/I-7523/I-7524/I-7527 User's Manual (this manual)
- One release note(Depend on situation)
- One software utility disk or CD
- One download cable → CA0910F for I-7521, I-7522, I-7523
  → CA0910 for I-7522A, I-7524, I-7527

Note:

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

Release Note (Important):

It is recommended to read the release note first. All important information will be given in release note as follows:

- **Influential modification about software or hardware**
- **The path that points to I-752N document and software changes in the companion CD**
- **others……….**

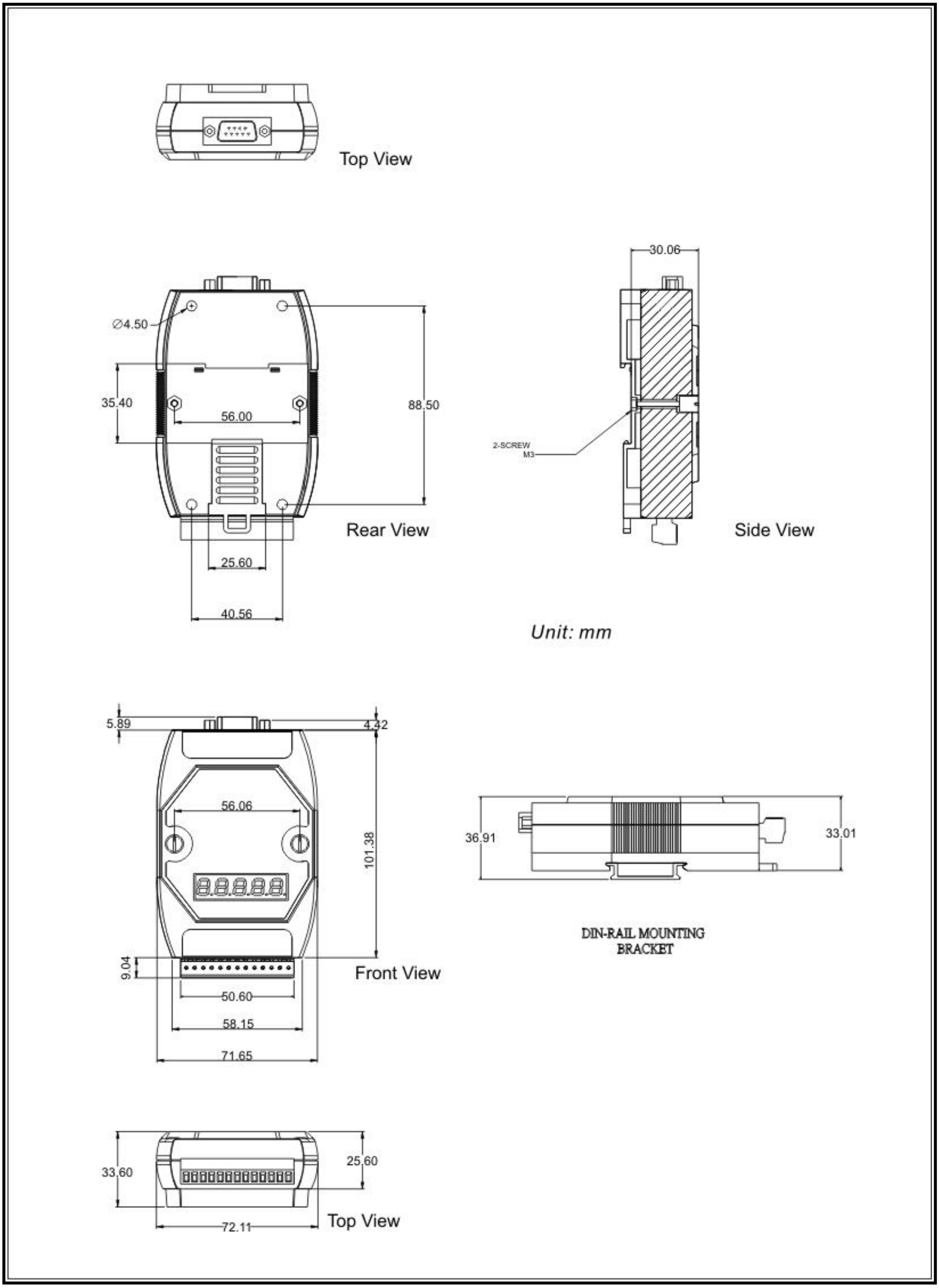
1.2. Features

General

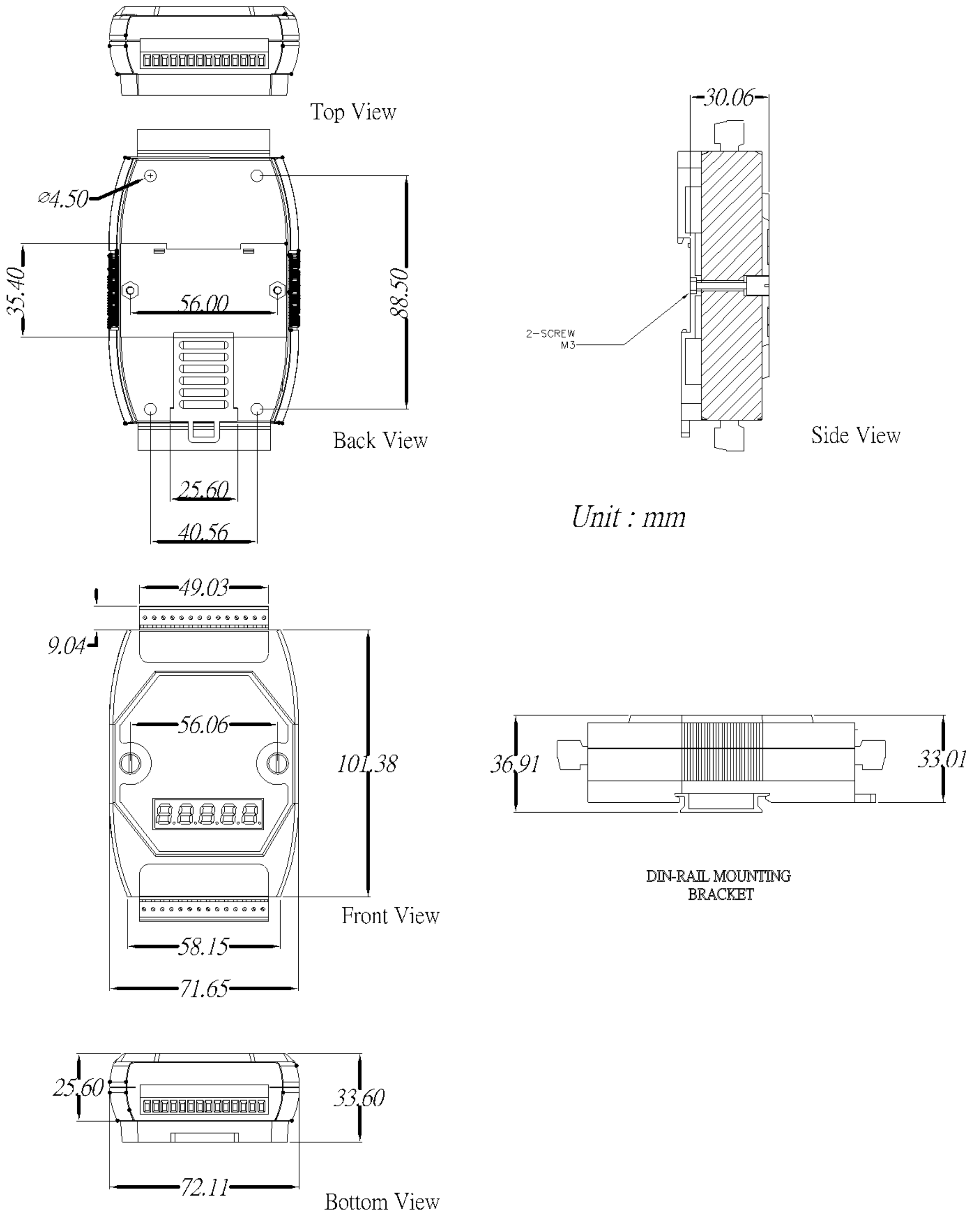
- Built-in “Addressable RS-232 Converter” firmware
- Supports Dual-Watchdog commands
- Supports Power-up value & safe value for DO
- The firmware code is open source and well documented
- Source code could be modified depending on according specific user requirements.
- The firmware can monitor the onboard DI and control the onboard DO in real-time depending on user requirements
- The firmware can monitor the RS-232 device and control the onboard DO in real-time according to user’s requirements
- Watchdog timer provides fault tolerance and recovery
- Low power consumption
- R.O.C. Invention Patent No. 086674
- R.O.C. Invention Patent No. 103060
- R.O.C. Patent No. 132457

1.3. Dimensions

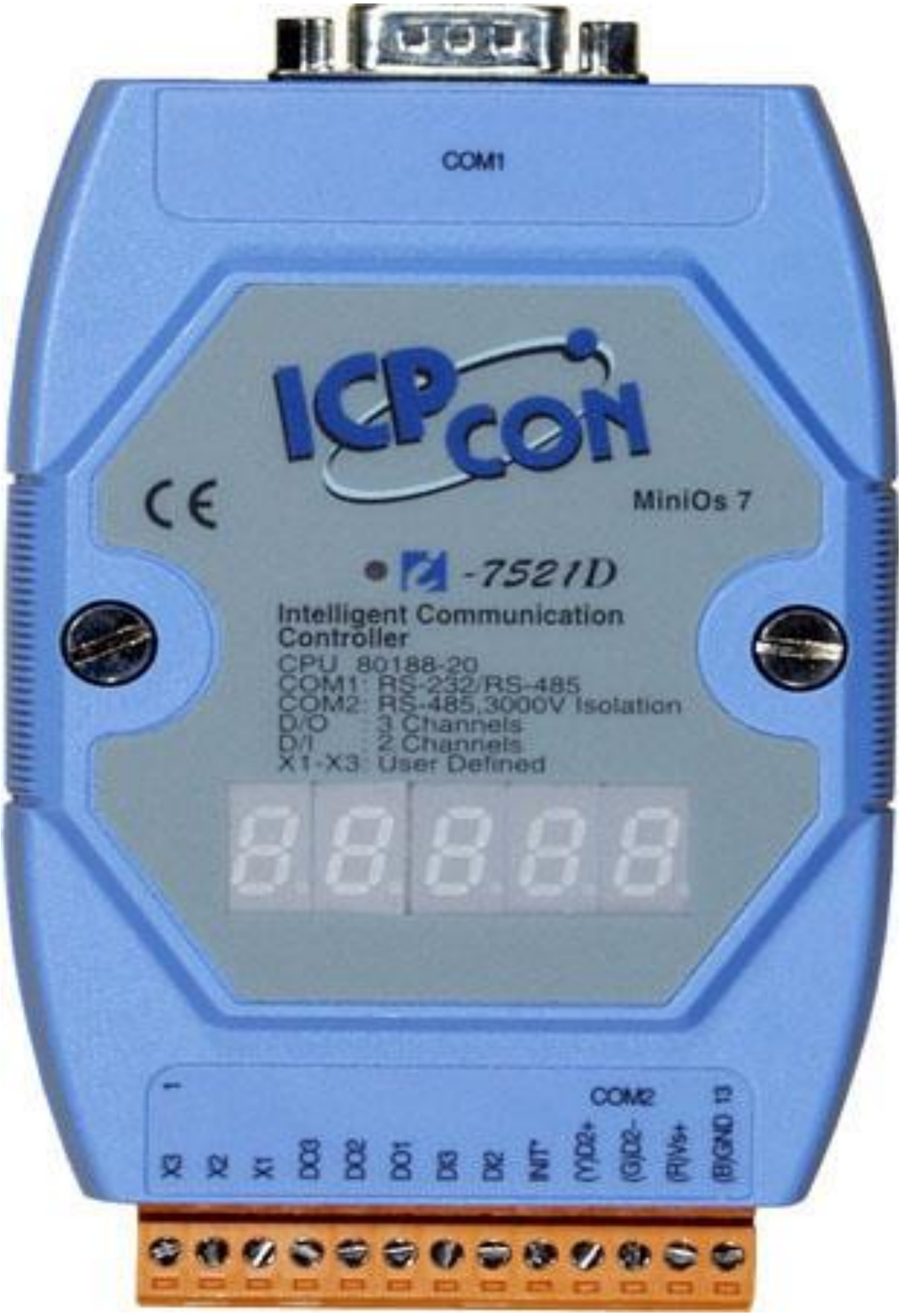
For I-7521(D)/I-7522(D)/I-7523(D)



For I-7522A(D)/I-7524(D)/I-7527(D)



1.4. Pin Assignment



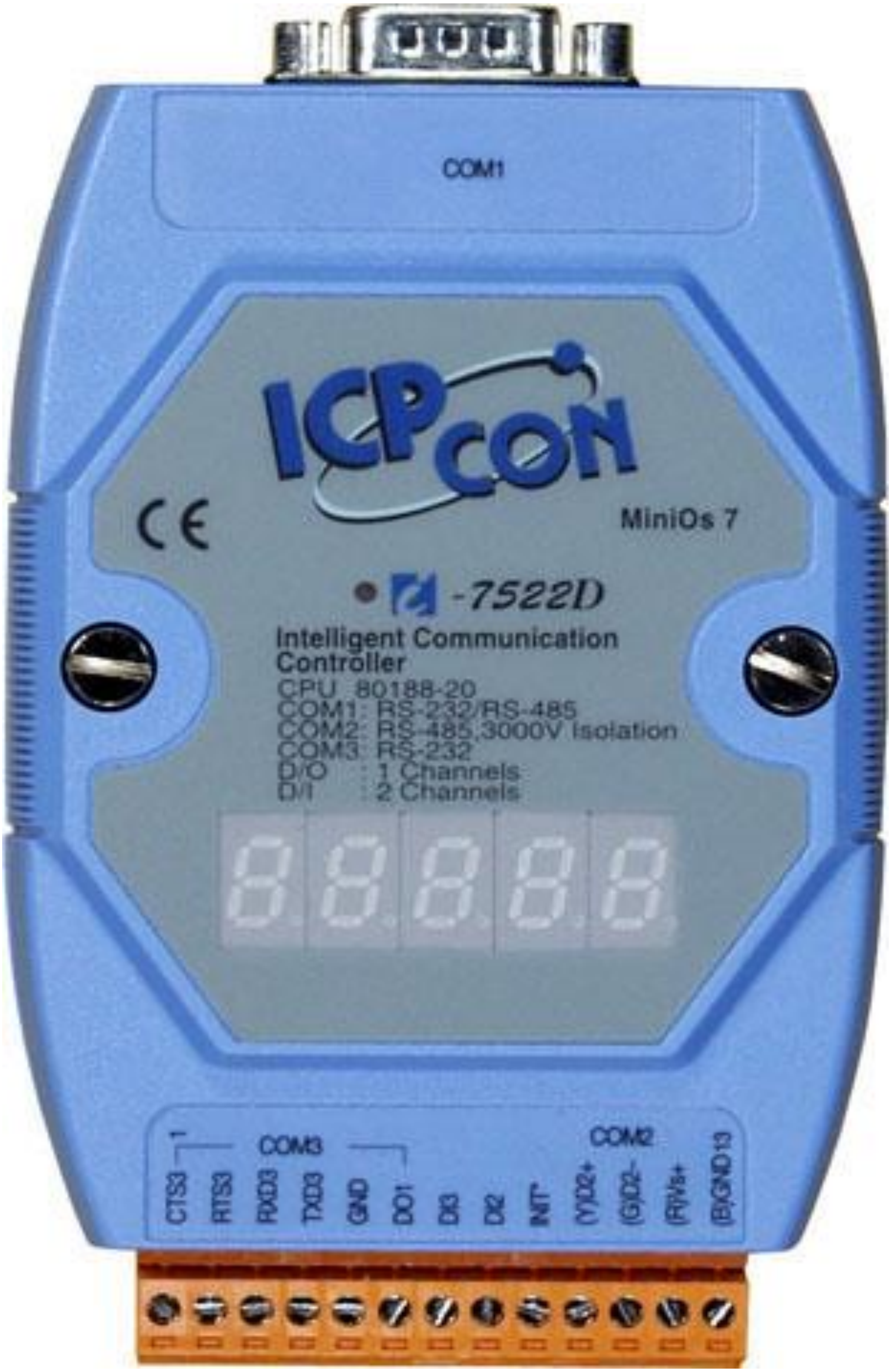
Pin assignment of 13-pin screw terminal block(I-7521/I-7521D):

Pin	Name	Description
1	X3	Connects to I/O expansion board
2	X2	Connects to I/O expansion board
3	X1	Connects to I/O expansion board
4	DO3	Digital output, 150mA, 30V
5	DO2	Digital output, 150mA, 30V
6	DO1	Digital output, 150mA, 30V
7	DI3	Digital input, 3.5V ~ 30V
8	DI2	Digital input, 3.5V ~ 30V
9	INIT*	Initial pin, 3.5V ~ 30V
10	D2+	DATA+ pin of COM2 (RS-485)
11	D2-	DATA- pin of COM2 (RS-485)
12	+VS	V+ of power supply (+10 to +30VDC unregulated)
13	GND	GND of power supply

Pin assignment of COM1 connector (DB-9 Male):

Pin	Name	Description
1	Data+	DATA+ of RS-485 port
2	TXD	Transmits Data (RS-232)
3	RXD	Receives Data (RS-232)
4	N/C	No Connection
5	GND	Signal ground of RS-232
6	N/C	No Connection
7	CTS	Clear To Send (RS-232)
8	RTS	Request To Send (RS-232)
9	Data-	DATA- of RS-485 port

Note: The COM1 can be used as s RS-232 port or s RS-485 port. It is not recommended to use both RS-232 & RS-485 at the same time.


COM1
ICP CON
CE
MiniOs 7
-7522D
Intelligent Communication Controller
CPU 80188-20
COM1: RS-232/RS-485
COM2: RS-485,3000V Isolation
COM3: RS-232
D/O : 1 Channels
D/I : 2 Channels
1
COM3
COM2
CTS3
RTS3
RXD3
TXD3
GND
DO1
DI3
DI2
INIT*
(Y)D2+
(G)D2-
(R)Vs+
(B)GND13

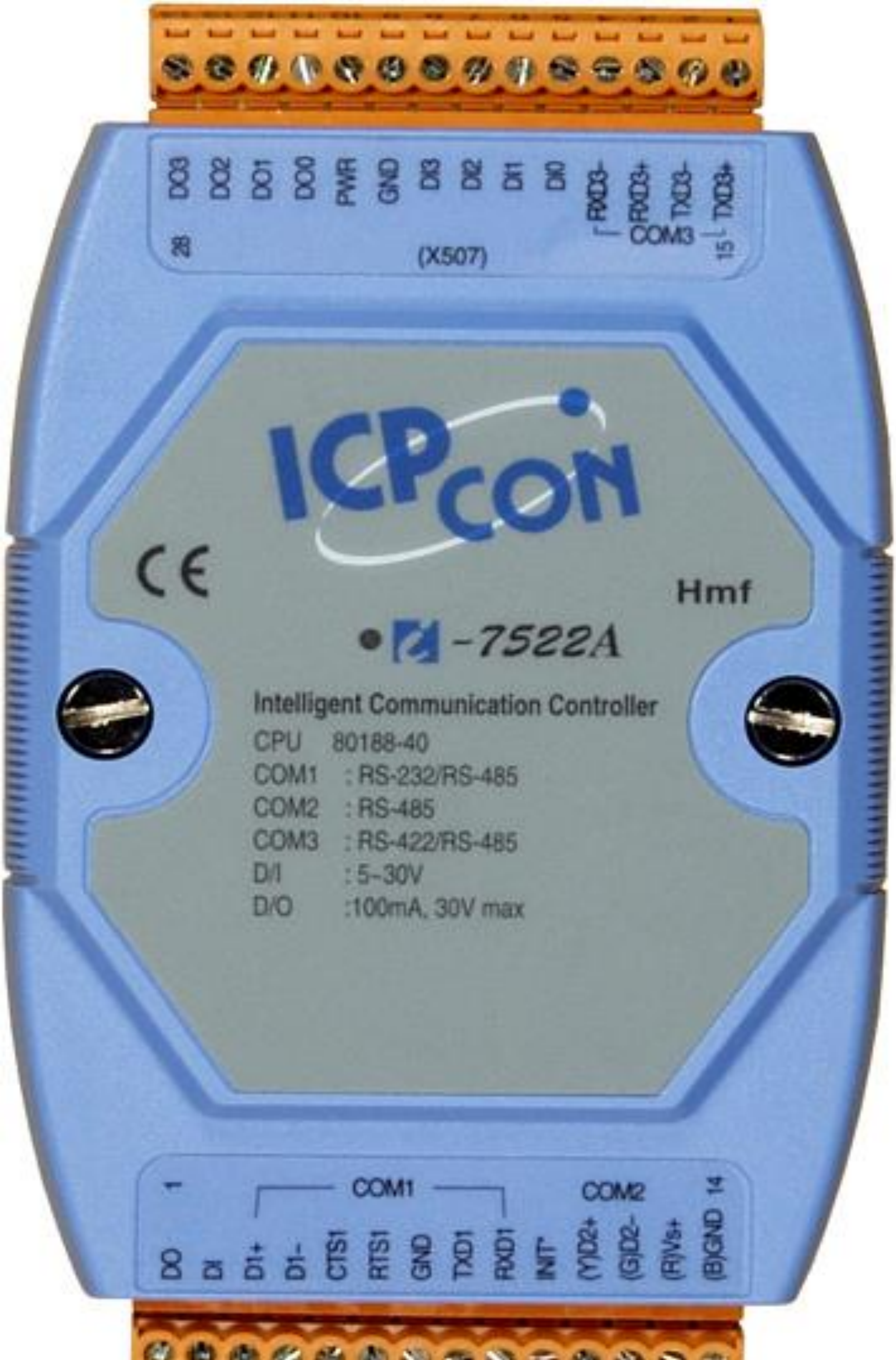
Pin assignment of 13-pin screw terminal block(I-7522/I-7522D):

Pin	Name	Description
1	CTS3	Clear To Send of COM3 (RS-232)
2	RTS3	Request To Send of COM3 (RS-232)
3	RXD3	Receives Data of COM3 (RS-232)
4	TXD3	Transmits Data of COM3 (RS-232)
5	GND	Signal ground of COM3 & COM4
6	DO1	Digital output, 150mA, 30V
7	DI3	Digital input, 3.5V ~ 30V
8	DI2	Digital input, 3.5V ~ 30V
9	INIT*	Initial pin, 3.5V ~ 30V
10	D2+	DATA+ pin of COM2 (RS-485)
11	D2-	DATA- pin of COM2 (RS-485)
12	+VS	V+ of power supply (+10 to +30VDC unregulated)
13	GND	GND of power supply

Pin assignment of COM1 connector (DB-9 Male):

Pin	Name	Description
1	Data+	DATA+ of RS-485 port
2	TXD	Transmits Data (RS-232)
3	RXD	Receives Data (RS-232)
4	N/C	No Connection
5	GND	Signal ground of RS-232
6	N/C	No Connection
7	CTS	Clear To Send (RS-232)
8	RTS	Request To Send (RS-232)
9	Data-	DATA- of RS-485 port

Note: The COM1 can be used as a RS-232 port or a RS-485 port. It is not recommended to use both RS-232 & RS-485 at the same time.


DO3
DO2
DO1
DO0
PWR
GND
DI3
DI2
DI1
DI0
RXD3-
RXD3+
TXD3-
TXD3+
28
COM3
15
(X507)
ICPCON
CE
Hmf
I-7522A
Intelligent Communication Controller
CPU 80188-40
COM1 : RS-232/RS-485
COM2 : RS-485
COM3 : RS-422/RS-485
D/I : 5~30V
D/O :100mA, 30V max
1
COM1
COM2
14
DO
DI
D1+
D1-
CTS1
RTS1
GND
TXD1
RXD1
INIT*
(Y)D2+
(G)D2-
(R)Vs+
(B)GND

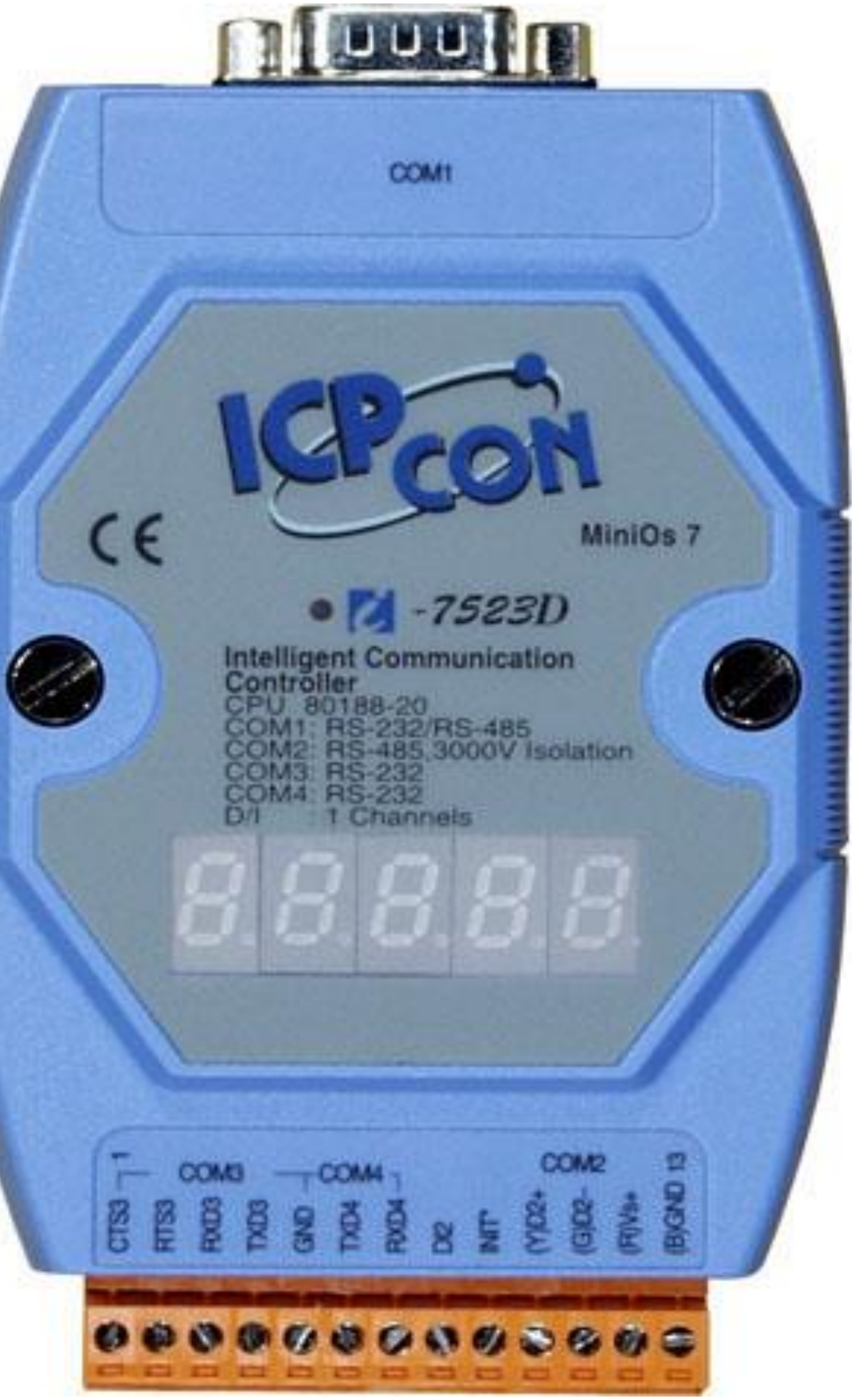
Pin assignment of bottom 14-pin screw terminal block (I-7522A /I7522AD):

Pin	Name	Description
1	DO	100 mA, 30V max. **DO1**
2	DI	3.5V ~ 30V,**DI1**
3	D1+	DATA+ pin of COM1 (RS-485)
4	D1-	DATA - pin of COM1 (RS-485)
5	CTS1	Clear To Send of COM1 (RS-232)
6	RTS1	Request To Send of COM1 (RS-232)
7	GND	Signal ground of RS-232
8	TXD1	Transmits Data of COM1 (RS-232)
9	RXD1	Receives Data of COM1 (RS-232)
10	INIT*	Initial pin, 3.5V ~ 30V
11	D2+	DATA+ pin of COM2 (RS-485)
12	D2-	DATA - pin of COM2 (RS-485)
13	+VS	V+ of power supply (+10 to +30VDC unregulated)
14	GND	GND of power supply

Note: The COM1 can be used as a RS-232 port or a RS-485 port. It is not recommended to use both RS-232 and RS-485 at the same time.

Pin assignment of top 14-pin screw terminal block:

15	TXD3+	TXD+ pin of COM3 (RS-422/RS-485)
16	TXD3-	TXD - pin of COM3 (RS-422/RS-485)
17	RXD3+	RXD+ pin of COM3 (RS-422)
18	RXD3-	RXD - pin of COM3 (RS-422)
19	DI0	Digital Input, 3.5V ~ 30V, **DI2**
20	DI1	Digital Input, 3.5V ~ 30V, **DI3**
21	DI2	Digital Input, 3.5V ~ 30V, **DI4**
22	DI3	Digital Input, 3.5V ~ 30V, **DI5**
23	GND	GND of Digital Output
24	PWR	Power of Digital Output
25	DO0	Digital Output, 100 mA, 30V max., **DO2**
26	DO1	Digital Output, 100 mA, 30V max., **DO3**
27	DO2	Digital Output, 100 mA, 30V max., **DO4**
28	DO3	Digital Output, 100 mA, 30V max., **DO5**


COM1
ICPCON
CE
MiniOs 7
-7523D
Intelligent Communication Controller
CPU 80188-20
COM1: RS-232/RS-485
COM2: RS-485,3000V Isolation
COM3: RS-232
COM4: RS-232
D/I : 1 Channels
1
COM3
COM4
COM2
13
CTS3
RTS3
RXD3
TXD3
GND
TXD4
RXD4
DI2
INIT*
(Y)D2+
(G)D2-
(R)Vs+
(B)GND

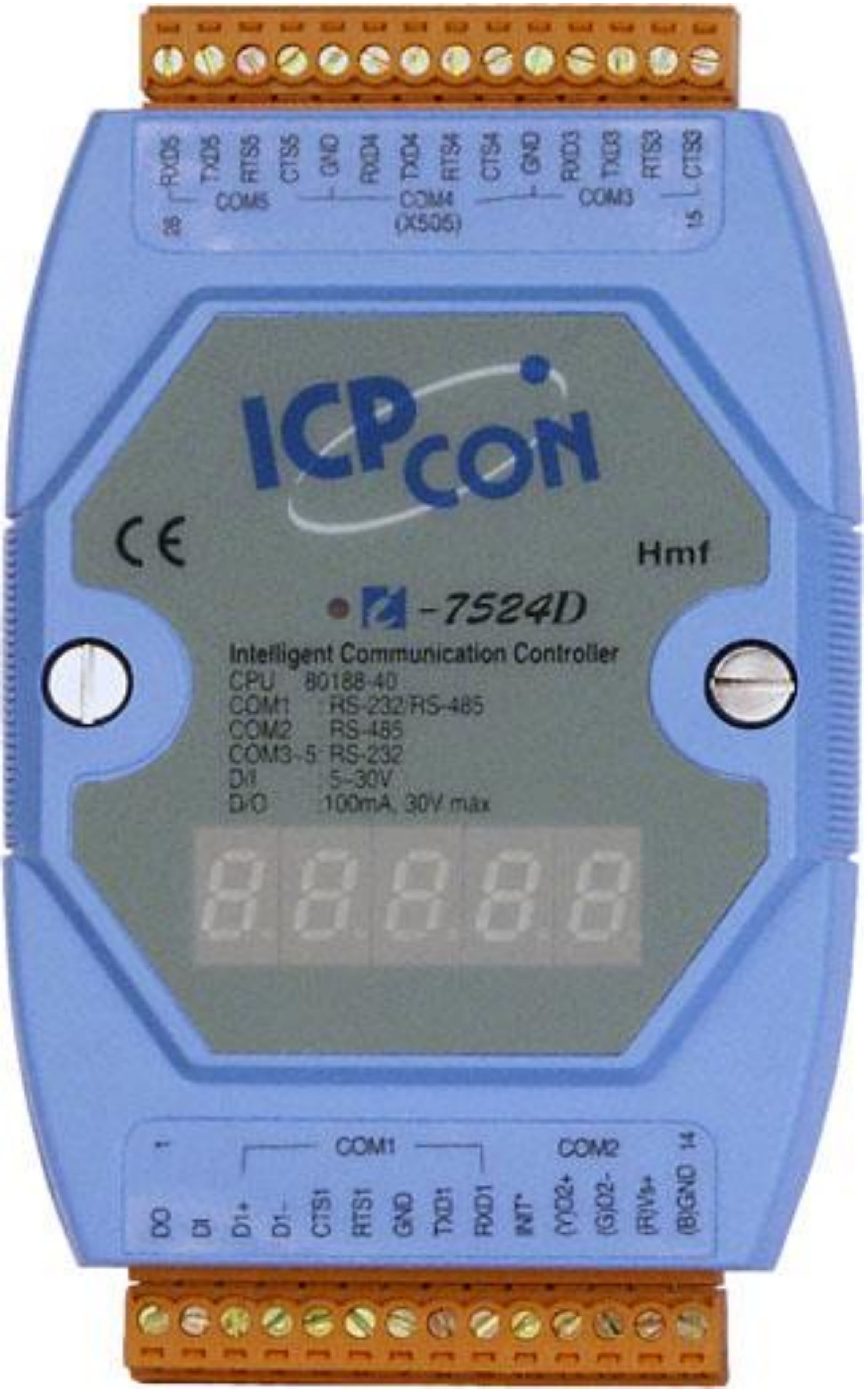
Pin assignment of 13-pin screw terminal block(I-7523/I-7523D):

Pin	Name	Description
1	CTS3	Clear To Send of COM3 (RS-232)
2	RTS3	Request To Send of COM3 (RS-232)
3	RXD3	Receives Data of COM3 (RS-232)
4	TXD3	Transmits Data of COM3 (RS-232)
5	GND	Signal ground of COM3 & COM4
6	TXD4	Transmits Data of COM4 (RS-232)
7	RXD4	Receives Data of COM4 (RS-232)
8	DI2	Digital input, 3.5V ~ 30V
9	INIT*	Initial pin, 3.5V ~ 30V
10	D2+	DATA+ pin of COM2 (RS-485)
11	D2-	DATA- pin of COM2 (RS-485)
12	+VS	V+ of power supply (+10 to +30VDC unregulated)
13	GND	GND of power supply

Pin assignment of COM1 connector (DB-9 Male):

Pin	Name	Description
1	Data+	DATA+ of RS-485 port
2	TXD	Transmits Data (RS-232)
3	RXD	Receives Data (RS-232)
4	N/C	No Connection
5	GND	Signal ground of RS-232
6	N/C	No Connection
7	CTS	Clear To Send (RS-232)
8	RTS	Request To Send (RS-232)
9	Data-	DATA- of RS-485 port

Note: The COM1 can be used as a RS-232 port or a RS-485 port. It is not recommended to use both RS-232 and RS-485 at the same time.


RXD5
TXD5
RTS5
CTS5
GND
RXD4
TXD4
RTS4
CTS4
GND
RXD3
TXD3
RTS3
CTS3
28
COM5
COM4
(X505)
COM3
15
ICPCON
CE
Hmf
I-7524D
Intelligent Communication Controller
CPU 80188-40
COM1 : RS-232/RS-485
COM2 : RS-485
COM3~5: RS-232
D/I : 5~30V
D/O : 100mA, 30V max
1
COM1
COM2
14
DO
DI
D1+
D1-
CTS1
RTS1
GND
TXD1
RXD1
INIT*
(Y)D2+
(G)D2-
(R)Vs+
(B)GND

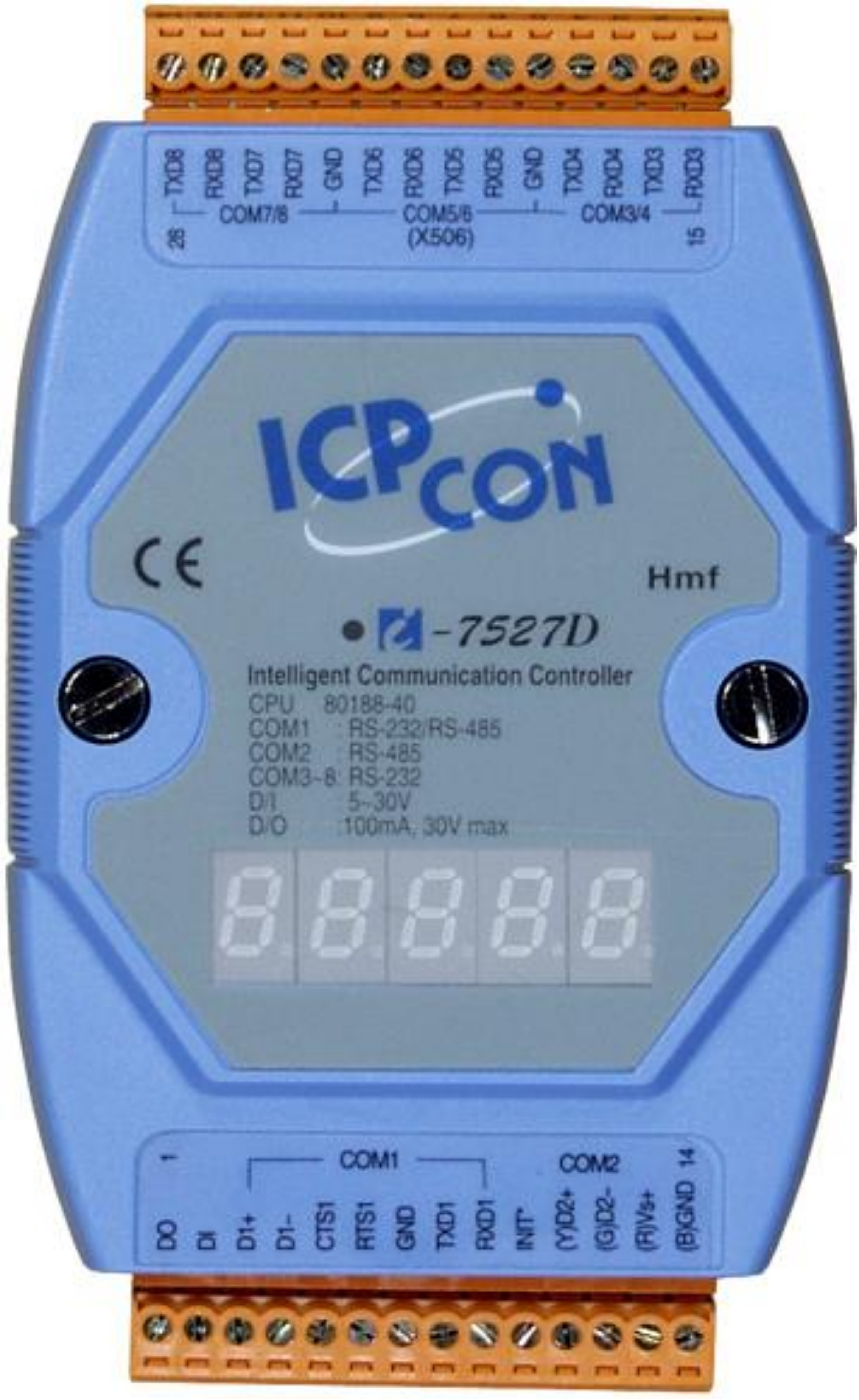
Pin assignment of bottom 14-pin screw terminal block (I-7524/ I-7524D):

Pin	Name	Description
1	DO	100 mA, 30V max.
2	DI	3.5V ~ 30V
3	D1+	DATA+ pin of COM1 (RS-485)
4	D1-	DATA - pin of COM1 (RS-485)
5	CTS1	Clear To Send of COM1 (RS-232)
6	RTS1	Request To Send of COM1 (RS-232)
7	GND	Signal ground of RS-232
8	TXD1	Transmits Data of COM1 (RS-232)
9	RXD1	Receives Data of COM1 (RS-232)
10	INIT*	Initial pin, 3.5V ~ 30V
11	D2+	DATA+ pin of COM2 (RS-485)
12	D2-	DATA - pin of COM2 (RS-485)
13	+VS	V+ of power supply (+10 to +30VDC unregulated)
14	GND	GND of power supply

Note: The COM1 can be used as a RS-232 port or a RS-485 port. It is not recommended to use both RS-232 & RS-485 at the same time.

Pin assignment of top 14-pin screw terminal block:

15	CTS3	CTS pin of COM3 (RS-232)
16	RTS3	RTS pin of COM3 (RS-232)
17	TXD3	TXD pin of COM3 (RS-232)
18	RXD3	RXD pin of COM3 (RS-232)
19	GND	GND pin of COM3/COM4 (RS-232)
20	CTS4	CTS pin of COM4 (RS-232)
21	RTS4	RTS pin of COM4 (RS-232)
22	TXD4	TXD pin of COM4 (RS-232)
23	RXD4	RXD pin of COM4 (RS-232)
24	GND	GND pin of COM4/5 (RS-232)
25	CTS5	CTS pin of COM5 (RS-232)
26	RTS5	RTS pin of COM5 (RS-232)
27	TXD5	TXD pin of COM5 (RS-232)
28	RXD5	RXD pin of COM5 (RS-232)


TXD8
RXD8
TXD7
RXD7
GND
TXD6
RXD6
TXD5
RXD5
GND
TXD4
RXD4
TXD3
RXD3
26
COM7/8
COM5/6
(X506)
COM3/4
15
ICPCON
CE
Hmf
I-7527D
Intelligent Communication Controller
CPU 80188-40
COM1 : RS-232/RS-485
COM2 : RS-485
COM3~8: RS-232
D/I : 5~30V
D/O :100mA, 30V max
1
COM1
COM2
14
DO
DI
D1+
D1-
CTS1
RTS1
GND
TXD1
RXD1
INIT*
(Y)D2+
(G)D2-
(R)Vs+
(B)GND

Pin assignment of bottom 14-pin screw terminal block:

Pin	Name	Description
1	DO	100 mA, 30V max.
2	DI	3.5V ~ 30V
3	D1+	DATA+ pin of COM1 (RS-485)
4	D1-	DATA - pin of COM1 (RS-485)
5	CTS1	Clear To Send of COM1 (RS-232)
6	RTS1	Request To Send of COM1 (RS-232)
7	GND	Signal ground of RS-232
8	TXD1	Transmits Data of COM1 (RS-232)
9	RXD1	Receives Data of COM1 (RS-232)
10	INIT*	Initial pin, 3.5V ~ 30V
11	D2+	DATA+ pin of COM2 (RS-485)
12	D2-	DATA - pin of COM2 (RS-485)
13	+VS	V+ of power supply (+10 to +30VDC unregulated)
14	GND	GND of power supply

Note: The COM1 can be used as a RS-232 port or a RS-485 port. It is not recommended to use both RS-232 & RS-485 at the same time.

Pin assignment of top 14-pin screw terminal block:

15	RXD3	RXD pin of COM3 (RS-232)
16	TXD3	TXD pin of COM3 (RS-232)
17	RXD4	RXD pin of COM4 (RS-232)
18	TXD4	TXD pin of COM4 (RS-232)
19	GND	GND pin of COM3/4/5/6 (RS-232)
20	RXD5	RXD pin of COM5 (RS-232)
21	TXD5	TXD pin of COM5 (RS-232)
22	RXD6	RXD pin of COM6 (RS-232)
23	TXD6	TXD pin of COM6 (RS-232)
24	GND	GND pin of COM5/6/7/8 (RS-232)
25	RXD7	RXD pin of COM7 (RS-232)
26	TXD7	TXD pin of COM7 (RS-232)
27	RXD8	RXD pin of COM8 (RS-232)
28	TXD8	TXD pin of COM8 (RS-232)

1.5. Specifications

Operating environment	
Operating Temperature	-25°C to +75°C
Storage Temperature	-40°C to +80°C
Humidity	5 to 90%
Built-in Watch Dog Timer	1.6. seconds for MiniOS7 1.0 0.8 seconds for MiniOS7 2.0
Built-in power protection & network protection circuit	
COM port	
Program download port	**COM1**
RS-485 port with self-tuner inside, 3000V isolation	For I-7521(D)/I-7522(D)/I-7523(D)
RS-485 port with self-tuner inside	For I-7522A(D)/I-7524(D) /I-7527(D)
Communication speed	
All ports	115200 bps max
Digital Input speed	
Digital Input	Input type: non-isolated On voltage level: +1V max (Connect to GND) Off voltage level: +3.5V~30V (Open)
Digital Output	Output type: Open-collector Output current: 100mA Max load voltage: +30V/DC
Dimensions	
I-7521(D)/I-7522(D)/I-7523(D)	119 x 72 x 33 mm
I-7522A(D)/I-7524(D) /I-7527(D)	123 x 72 x 33 mm
Power	
Protection	Power reverse polarity protection
Power requirement	10 to 30 V/DC(non-regulated)
Power consumption	2W (without display) 3W (with display)

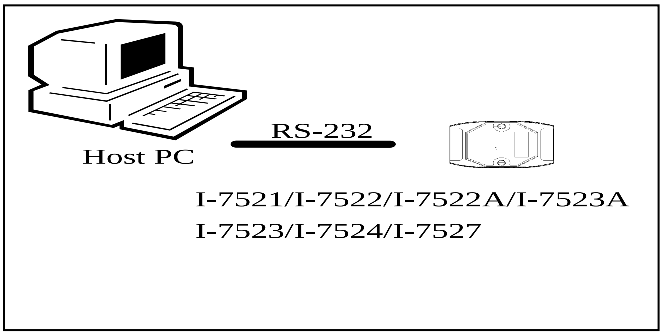
For ODM User

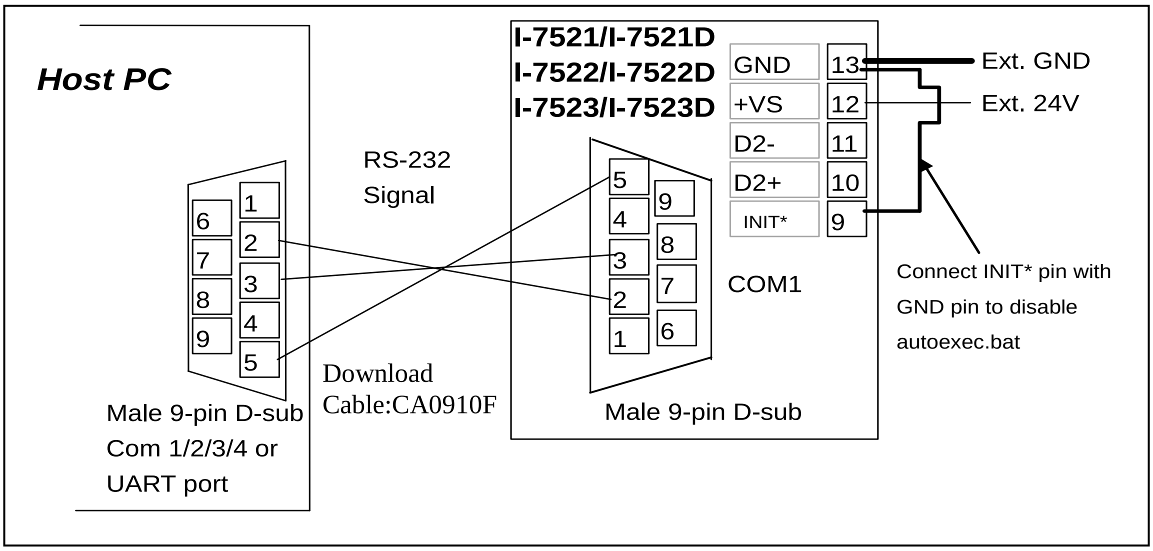
- SRAM can be 512K max.

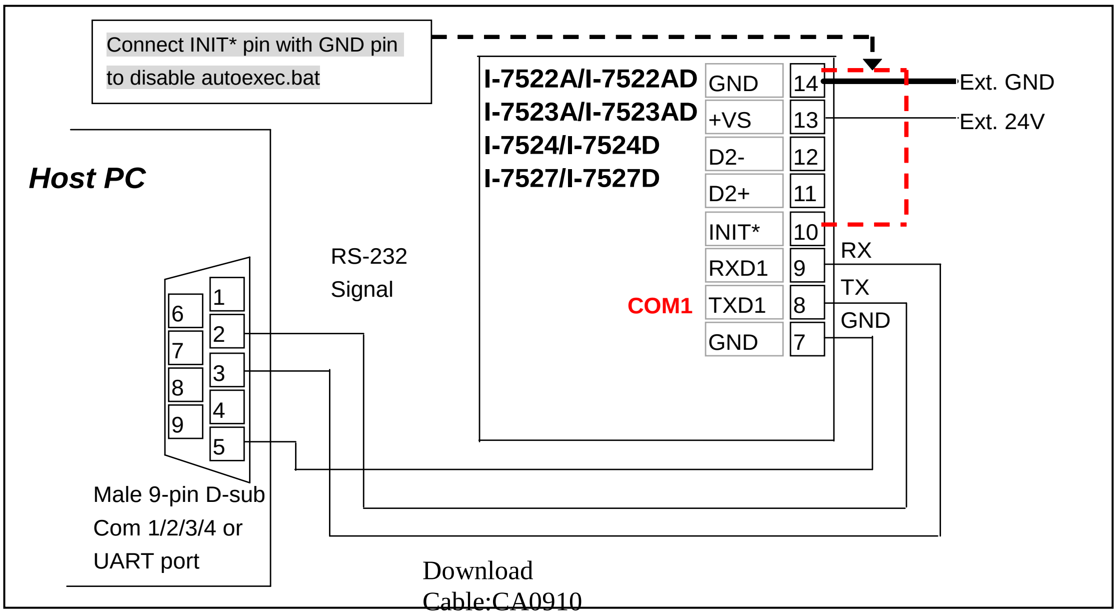
- COM2 can be 3000V isolated
- EEPROM can be 32K bytes

1.6. Wiring Diagrams for Application

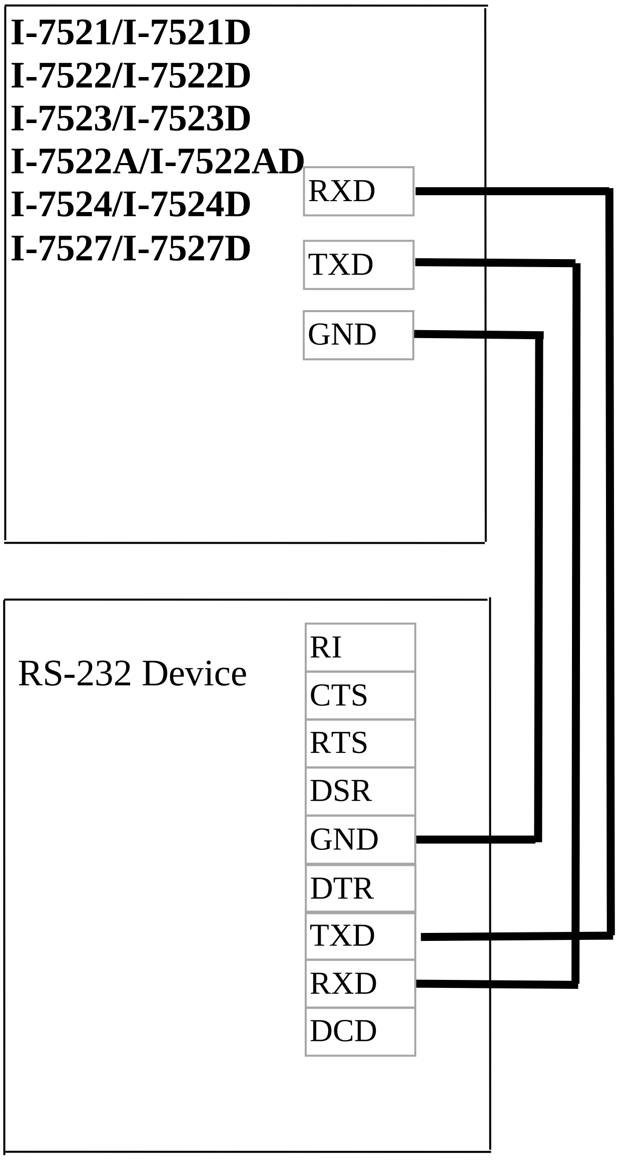
1.6.1. Program download







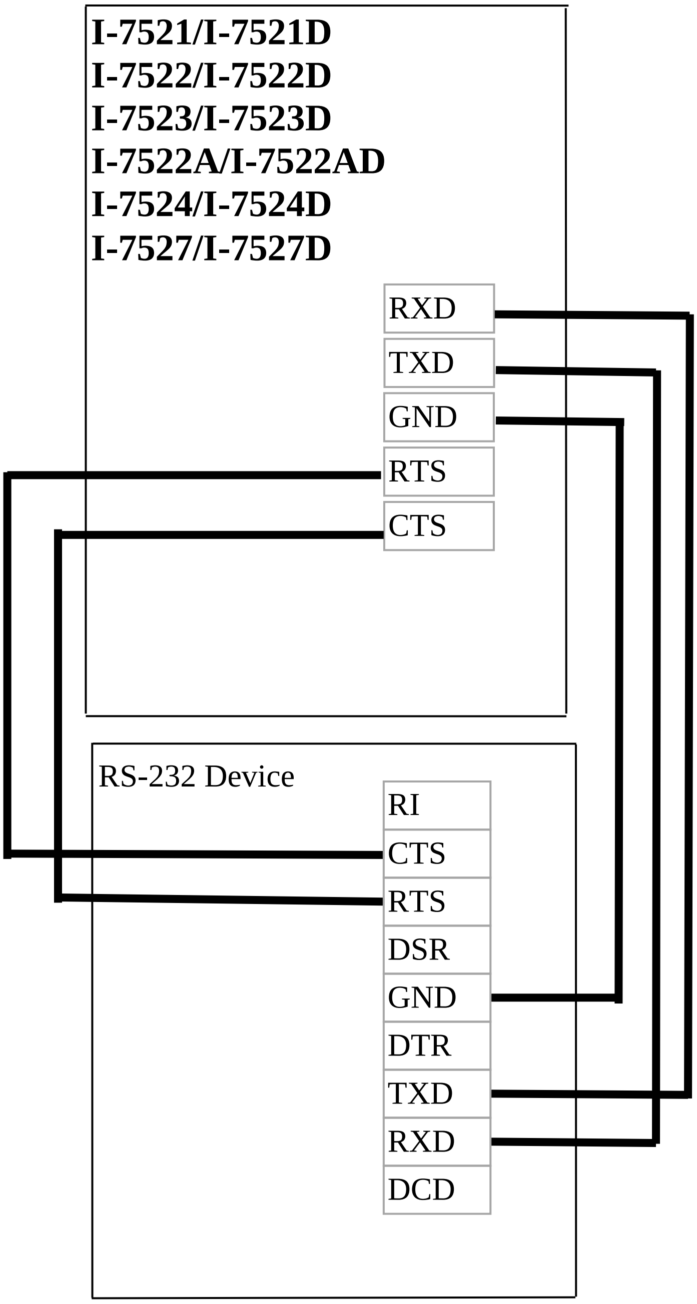
1.6.2. Using a 3-wire RS-232 port



There are 3 wires as following:

- Connect RXD to TXD of RS-232 device
- Connect TXD to RXD of RS-232 device
- Connect GND to GND of RS-232 device

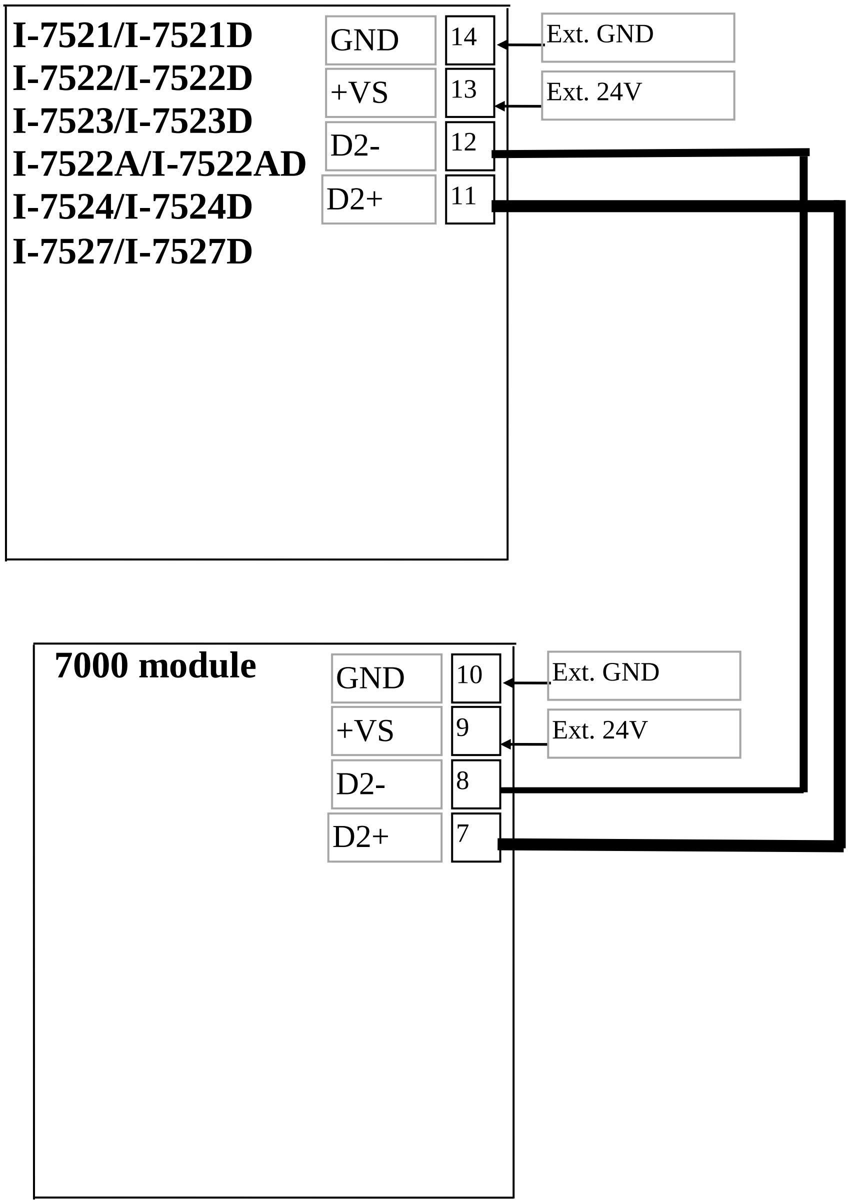
1.6.3. Using a 5-wire RS-232 port



There are 5 wires as follows:

- Connect RXD to TXD of RS-232 device
- Connect TXD to RXD of RS-232 device
- Connect RTS to CTS of RS-232 device
- Connect CTS to RTS of RS-232 device
- Connect GND to GND of RS-232 device

1.6.4. Using a RS-485 port

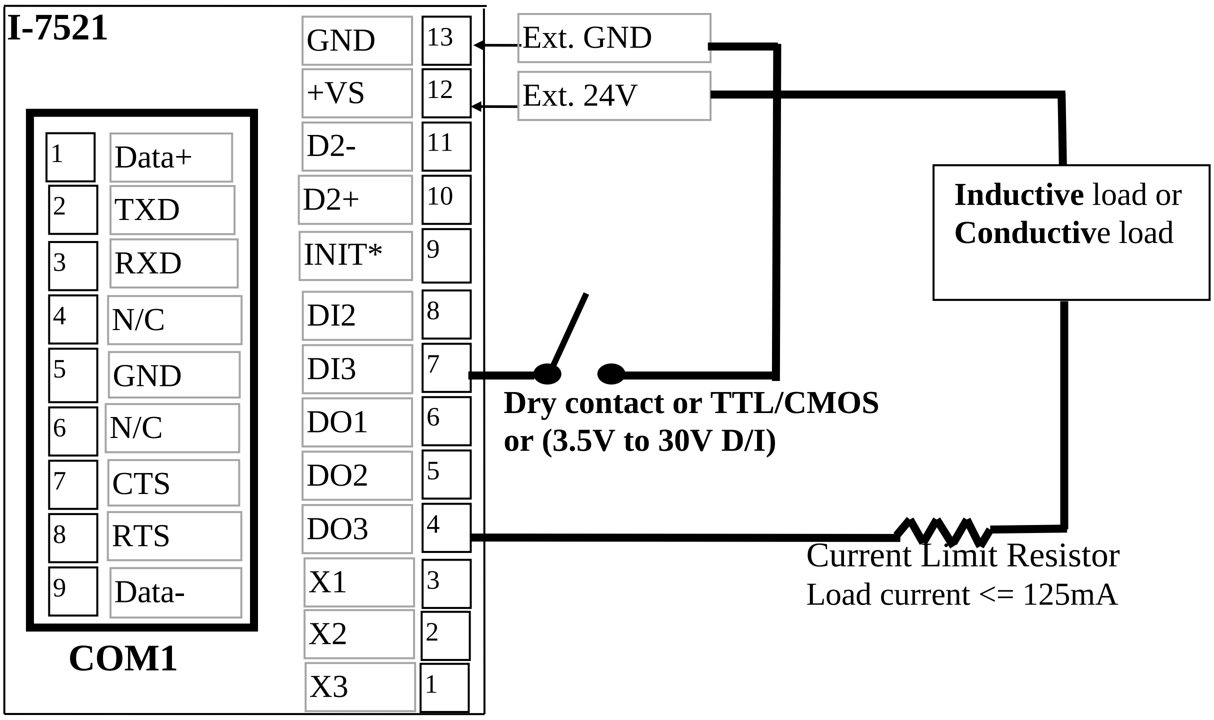


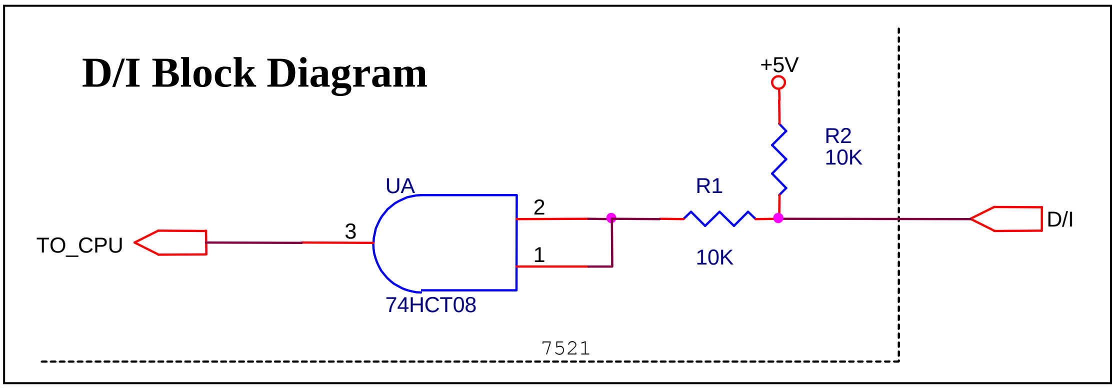
Note:

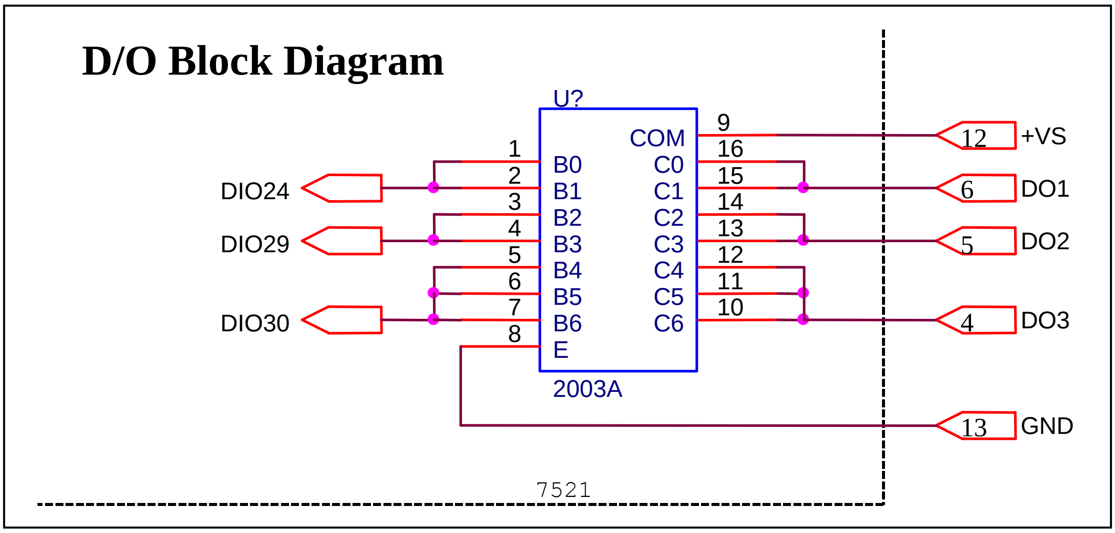
- The RS-485 interface can directly drive 256 sets of 7000 modules without a repeater

1.6.5. Using DI/DO of I-7521(D)

(valid for I-7521(D)/I-7522(D)/I-7523(D))

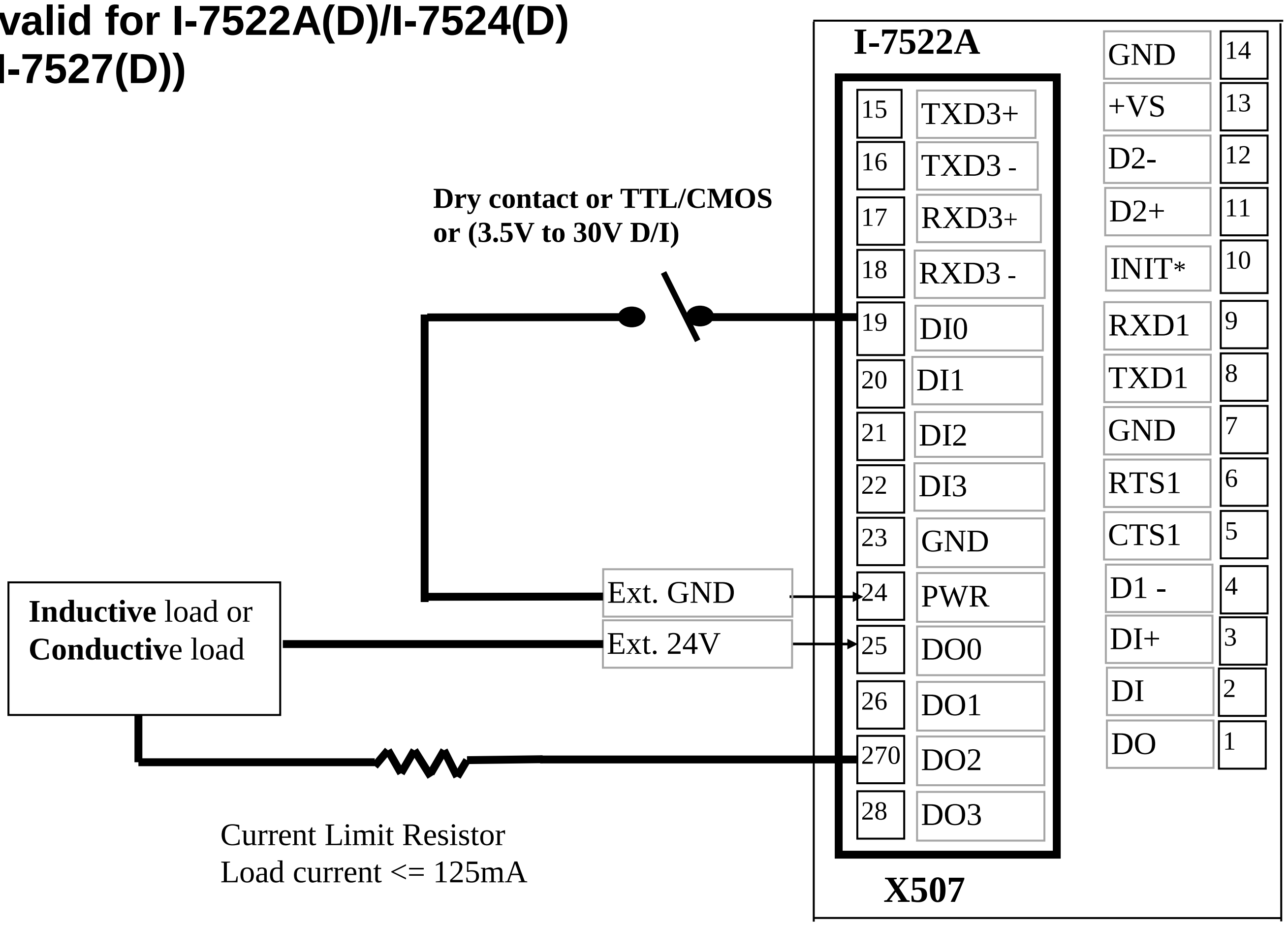






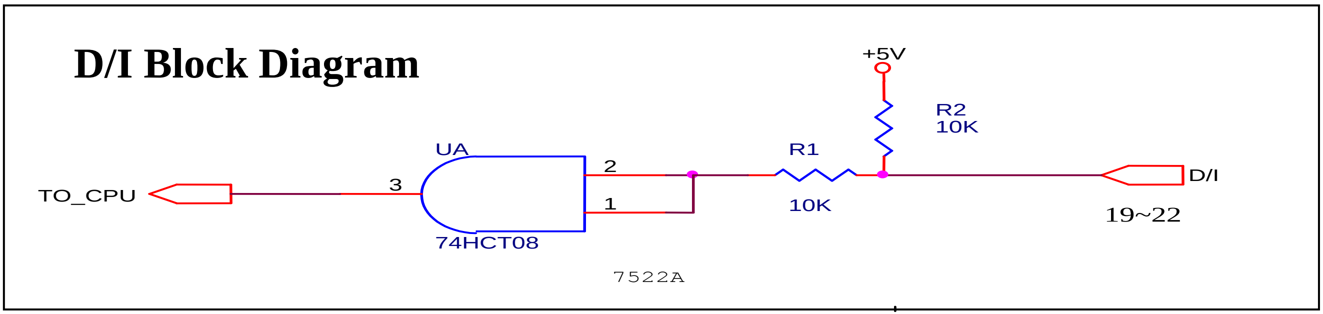
1.6.6. Using DI/DO of I-7522A(D)

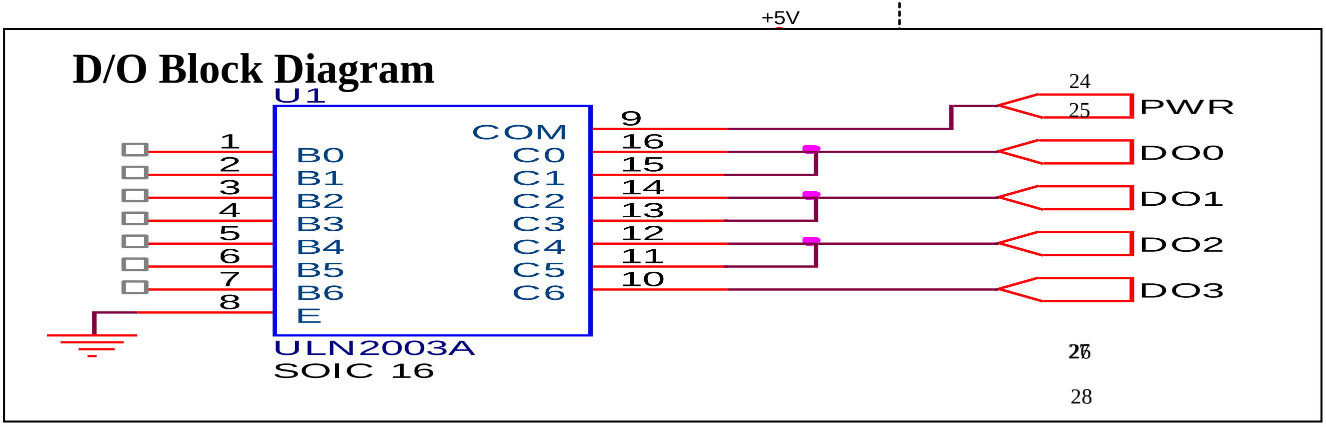
(valid for I-7522A(D)/I-7524(D) /I-7527(D))



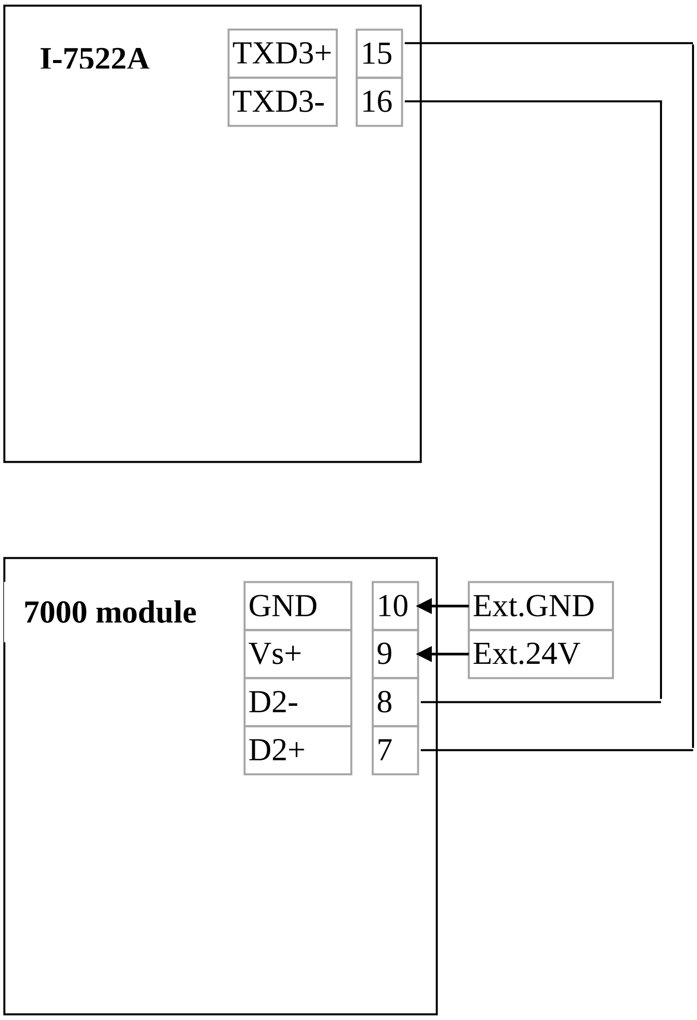
GND	14	15	TXD3+
+VS	13	16	TXD3 -
D2-	12	17	RXD3+
D2+	11	18	RXD3 -
INIT*	10	19	DI0
RXD1	9	20	DI1
TXD1	8	21	DI2
GND	7	22	DI3
RTS1	6	23	GND
CTS1	5	24	PWR
D1 -	4	25	DO0
DI+	3	26	DO1
DI	2	270	DO2
DO	1	28	DO3

Current Limit Resistor
Load current <= 125mA


D/I Block Diagram


D/O Block Diagram

1.6.7. Using a RS-485 of I-7522A(D)



Note:

- The RS-485 interface can directly drive 256 sets of 7000 modules without a repeater

1.7. Comparison Table

Comparison Table of I-7521(D),I-7522(D),I-7523(D)

	I-7521(D)	I-7522(D)	I-7523(D)
CPU clock	80188 or compatible, 20M Hz	80188 or compatible, 20M Hz	80188 or compatible, 20M Hz
SRAM	128K	128K	128K
Flash ROM	512K	512K	512K
COM1	RS-232 or RS-485 with self-tuner inside	RS-232 or RS-485 with self-tuner inside	RS-232 or RS-485 with self-tuner inside
COM2	RS-485 with self-tuner inside, 3000V isolation	RS-485 with self-tuner inside, 3000V isolation	RS-485 wtih self-tuner inside, 3000V isolation
COM3	No Com3	RS-232 (txd,rxd,rts,cts)	RS-232 (txd,rxd,rts,cts)
COM4	No Com4	No Com4	RS-232 (txd,rxd)
Transmission Speed	Up to 115.2 kbps	Up to 115.2 kbps	Up to 115.2 kbps
RTC	No	No	No
EEPROM	2K bytes	2K bytes	2K bytes
D/I(3.5V~30V)	2 channels	2 channels	1 channels
D/O(100mA)	3 channels	1 channel	0
Operation system	MiniOS7	MiniOS7	MiniOS7
Program download	COM1	COM1	COM1

Comparison Table of I-7522A(D),I-7524(D),I-7527(D)

	I-7522A(D)	I-7524(D)	I-7527(D)
CPU clock	80188, or compatible 40M Hz	80188, or compatible 40M Hz	80188, or compatible 40M Hz
SRAM	256K	256K	256K
Flash ROM	512K	512K	512K
COM1	RS-232 or RS-485 with self-tuner inside	RS-232 or RS-485 with self-tuner inside	RS-232 or RS-485 with self-tuner inside
COM2	RS-485 with self-tuner inside	RS-485 with self-tuner inside	RS-485 wtih self-tuner inside
COM3	RS-422/RS-485 (txd3-,rxd3+ , txd3+,rxd3-)	RS-232 (txd,rxd,rts,cts)	RS-232(txd,rxd)
COM4	No Com4	RS-232 (txd,rxd,rts,cts)	RS-232(txd,rxd)
COM5	No Com5	RS-232 (txd,rxd,rts,cts)	RS-232(txd,rxd)
COM6	No Com6	No Com6	RS-232(txd,rxd)
COM7	No Com7	No Com7	RS-232(txd,rxd)
COM8	No Com8	No Com8	RS-232(txd,rxd)
Transmission Speed	Up to 115.2 kbps	Up to 115.2 kbps	Up to 115.2 kbps
RTC	Yes	Yes	Yes
EEPROM	2K bytes	2K bytes	2K bytes
D/I(3.5V~30V)	5 channels	1 channel	1 channel
D/O(100mA)	5 channels	1 channel	1 channel
Operation system	MiniOS7	MiniOS7	MiniOS7
Program download	COM1	COM1	COM1

Chapter 2 Operating Principles

2.1 INIT* pin operating principles

The INIT* pin has two functions:

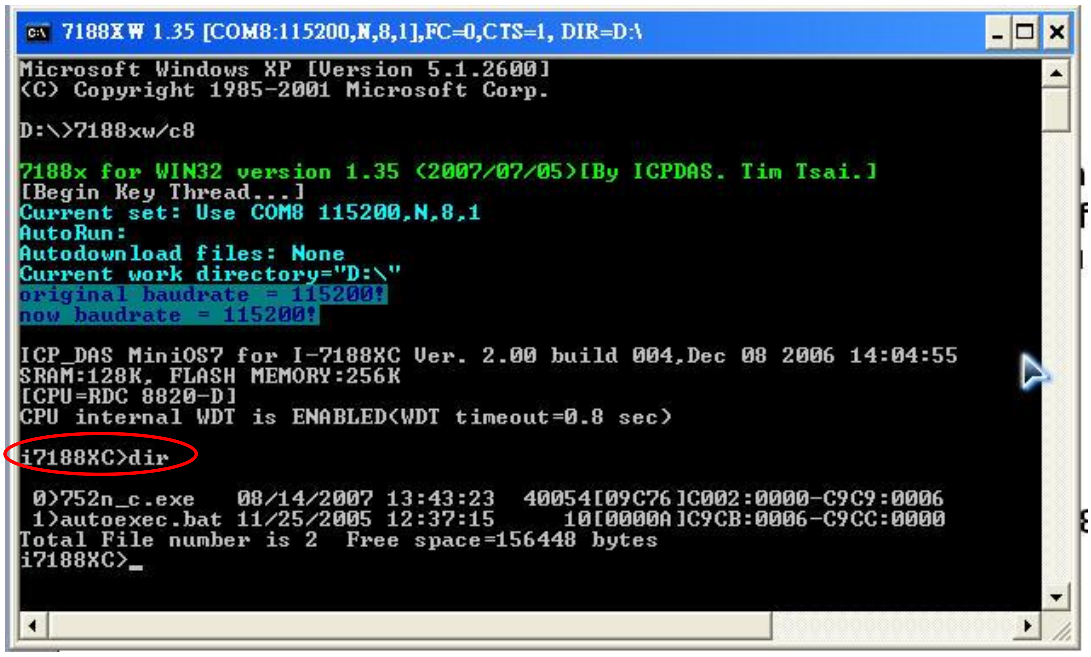
1) If provides help for downloading program. (Refer to Sec. 3.4. for details)

2) If retrieves configuration data stored in EEPROM.

If the configuration data for the I-752N series module is forgotten, resulting in a communication failure, the procedure for retrieving the configuration data stored in the EEPROM is as follows (Applicable to firmware version number 3.0 and above):

Step 1: Refer to Sec. 1.6.1 and steps 2 to 9 in Sec. 3.1 to set the configuration parameters to 115200, N, 8, 1. After pressing “Enter”, either “i7188XC>” or “i7188xB>” will be displayed on the screen.

Step 2: Type the “dir” command, then press “Enter.” The following screen will be displayed:



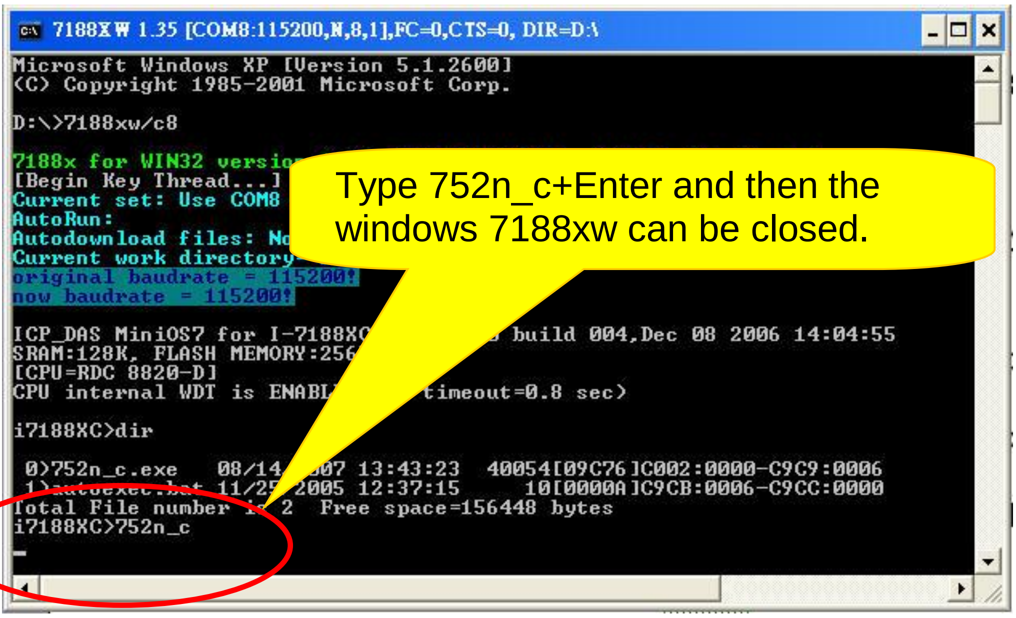
```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

D:\>7188xw/c8

7188x for WIN32 version 1.35 (2007/07/05)[By ICPDAS. Tim Tsai.]
[Begin Key Thread...]
Current set: Use COM8 115200,N,8,1
AutoRun:
Autodownload files: None
Current work directory="D:\"
original baudrate = 115200!
now baudrate = 115200!

ICP_DAS MiniOS7 for I-7188XC Ver. 2.00 build 004,Dec 08 2006 14:04:55
SRAM:128K, FLASH MEMORY:256K
[CPU=RDC 8820-D]
CPU internal WDT is ENABLED(WDT timeout=0.8 sec)

i7188XC>dir

 0)752n_c.exe    08/14/2007 13:43:23   40054[09C76]C002:0000-C9C9:0006
 1)autoexec.bat 11/25/2005 12:37:15      10[0000A]C9CB:0006-C9CC:0000
Total File number is 2  Free space=156448 bytes
i7188XC>_
```

NOTE: the text shown in the ellipse region is “i7188XC>” and will be displayed for I-7521/I-7522 /I-7523 modules, but “i-7188XB>” will be displayed for I-7522A/I-7524/I-7527 modules.

Step 3: Type “752n_c” (for I-7521/I-7522/I-7523) or “752n_b” (for I-7522A/I-7524/ I-7527) and press the Enter key to execute the 752n_c.exe or 752n_b.exe file (Note: the INIT* pin should be connected to the GND pin). The I-752N module will **revert to the factory default settings** without changing the EEPROM data. The factory default settings are as follows:

COM Port	**= 2**
Module Address	**= 00**
Baud Rate	**= 9600**
Checksum	**= DISABLED**
CrLfmode	**= 0 (0x0D)**

Data format = 1 start bit + 8 data bits + 1 stop bit (N, 8, 1)



```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

D:\>7188xw/c8

7188x for WIN32 versio
[Begin Key Thread...]
Current set: Use COM8
AutoRun:
Autodownload files: No
Current work directory
original baudrate = 115200!
now baudrate = 115200!

ICP_DAS MiniOS7 for I-7188XC          build 004,Dec 08 2006 14:04:55
SRAM:128K, FLASH MEMORY:256
[CPU=RDC 8820-D]
CPU internal WDT is ENABL          timeout=0.8 sec)

i7188XC>dir

 0)752n_c.exe    08/14  007 13:43:23   40054[09C76]C002:0000-C9C9:0006
 1)autoexec.bat 11/25 2005 12:37:15      10[0000A]C9CB:0006-C9CC:0000
Total File number is 2  Free space=156448 bytes
i7188XC>752n_c
```

Type 752n_c+Enter and then the windows 7188xw can be closed.

Step 4: Refer to Sec. 3.1 and send commands to **COM2** to read the configuration for COM 2 or other ports. Some examples are as follows:

(a) Send the command string **$00M[0x0D]** to retrieve the module name and record it

(b) Send the command string **$00A[0x0D]** to retrieve the module address and record it

(c) Send the command string **$00B0[0x0D]** to retrieve the Baud Rate for COM2 and record it
(d) Send the command string **$00T0[0x0D]** to retrieve the CrLfmode for COM2 and record it

Step5: Power off the module and disconnect the INIT* and GND pins

Step6: Power on the module and communicate with the I-752N module using the same status settings as those you previously recorded.

If the INIT* and GND pins are disconnected, the module will execute 752n_c.exe or 752n_b.exe and the autoexec.bat files, and then the module will auto configure itself based on the configuration data stored in the EEPROM.

2.2 Demo Code, Firmware and 7188xw.exe

To locate the demo code, firmware and the 7188xw.exe file:

- Insert the installation CD and wait for the auto run function to activate (or run auto32.exe)
- Click “Toolkits (Software)/Manuals”
- Click “7521/2/3A/2A/3/4/7 Series”
- Click “Demo Program”. Several ODM programs and demo codes for I-752N modules will be displayed. The code can be modified for individual applications. All code is well-documented, so the code can be easily changed.
  Location:
  Companion CD: napdos/752n/source_code/
  or http://ftp.icpdas.com.tw/pub/cd/8000cd/napdos/752n/source_code/
- Click “View 752n Firmware” and the firmware can be obtained:
  Location:
  Companion CD: Napdos\752N\Firmware_V3
  Or http://ftp.icpdas.com.tw/pub/cd/8000cd/napdos/752n/firmware_v3

  NOTE: There are two files, autoexec.bat and 752n_c.exe/ 752n_b.exe, stored in the flash ROM of each I-752N module. The

752n_c.exe/752n_b.exe file will be executed once the power is supplied to the module and the INIT* pin is floating.

The readme.txt file recording upgrade or modification information related to the firmware in the firmware_v3 folder.

- Click “BACK”
- Click “MiniOS7”
- Double click “utility,” and 7188XW.exe can be obtained.The icon for 7188xw.exe is as below:

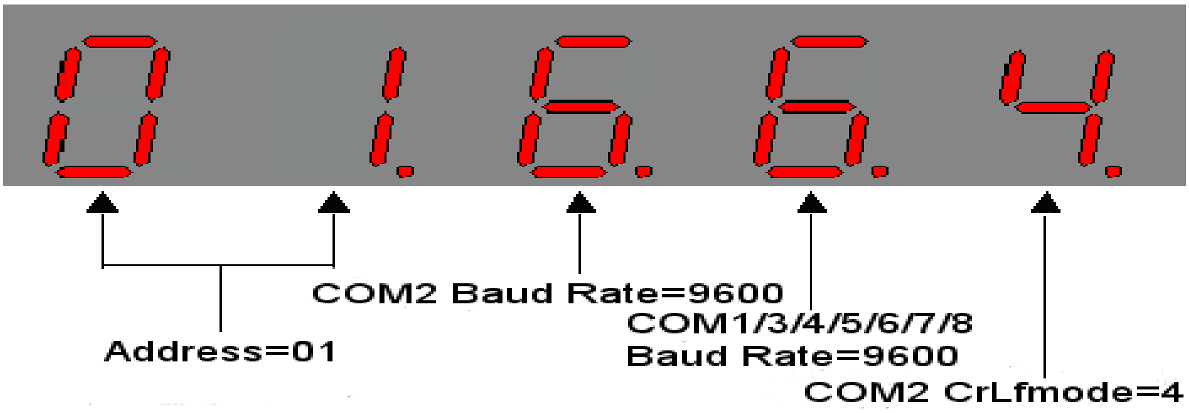
Location:
Companion CD: Napdos\MiniOS7\utility
or http://ftp.icpdas.com.tw/pub/cd/8000cd/napdos/minios7/utility/

2.3 The RS-485 Port and Self-Turner ASIC

The 2-wire RS-485 port is half-duplex. Send/receive directional control in a 2-wire RS-485 network is very important. Therefore, each I-752N module is equipped with a Self-Tuner ASIC controller for all RS-485 ports. The Self-Tuner ASIC controller will automatically detect and control the send/receive direction of the RS-485 network. First and foremost, the application program does not have to make allowance for the send/receive direction control of the RS-485 network.

2.4 7-Segment LED Display

The red LED of the I-752N series can be turned ON or OFF using software. The 5 digits of the 7-segment LED are also programmable. **The 5-digit LED is very useful in real world applications,** and can be used to replace the monitor and touch-screen for many applications. The 5 digits of the 7-segment LED will show the address, Baud Rate and CrLfmode for each COM port in turns when the I-752n firmware is executed.



2.5 752N protocol description

[Address]

There is a initial address for I-752N series module.
For example: (assume initial address is 01, CrLfmode is 4, data format=9600, N, 8, 1 for COM2 of a I-7522A module)

```
    command       response
     $01A           !01   ----> the initial address is 01
     $02A           !01   ----> the initial address is 01
     $03A          ?01   ----> an invalid command

     $01A05         !01 ----> changes the initial address to 05, so COM1 is
                                    now 05 and COM3 is now 06
     $06A           !05 ----> the initial address is 05
```

[CrLfmode]

There are five modes for I-752N when receiving commands (COM2) and receiving responses (COM1, 3~8).
We call this mode "CrLfmode". The default CrLfmode is 4 for all COM ports.
Note: More information about the CrLfmode command can be found in Sec. 5.3.12.

a: CrLfmode=0/1/2/3

In these modes, the I-752N module judges when the command/response string ends based on the end character of the command/response string.

For example: (initial address is 01, I-7522A)
Assume CrLfmode=0 for COM2,
COM2 determines the end of a command string when it encounters the 0x0D (CR) character. When manually entering individual command characters using a PC terminal application, the 0x0D (CR) character must be sent as the final

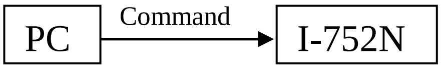
character of the command string. If this character is not sent, COM2 will continue to receive command characters and consider them as part of the command string, resulting in an error.

For more information about RS-232 COM ports, refer to the Notes in Sec. 5.3.33.

b: CrLfmode=4

The factory default value for CrLfmode is 4 for all COM ports. In this mode, the I-752N module determines the end of a command string once no characters are received for a specified timeout interval.

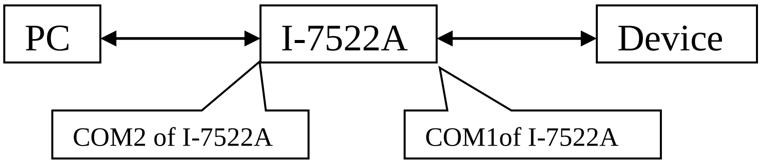
For example:



Assume the timeout interval is 10 ms.
Command string: ":01T"
If you type the first character ':', then type the second character,'0', after 5 ms, type the third character ,'1', after 1 second and type the fourth character ,'T', after 3 ms
Two command string will be received by COM2, which will be ":0" and "1T", not ":01T". So the command received is not the same as the original command.
The timeout interval is so small that it is impossible to manually enter individual command characters. In this situation, the command string must be sent at one time using software. The interval between transmission of the characters can be very small if the command string is sent using software.

c: **Set the corresponding CrLfmode for all COM ports depending on the command and response**.

For example:



Command: ":01ABCD<CR>" sent by PC
Device response: "EFGH<CR>," when the command is "ABCD<CR>"

Situation 1:

I-7522A, COM2 CrLfmode=0, COM1 CrLfmode=4

The data received by COM2 is "ABCD" where CrLfmode=0 (the <CR> character is removed) and "ABCD" is sent to COM1 where CrLfmode=4 (nothing adding). So the device receives the string "ABCD", not "ABCD<CR>", and doesn't give any response.

Command transformation:
":01ABCD<CR>",PC-->"ABCD",com2-->"ABCD",com1-->"ABCD", device

Situation 2:

I-7522A, COM2 CrLfmode=0, COM1 CrLfmode=0

The data received by COM2 is "ABCD" where CrLfmode=0 (the <CR> character is removed) and "ABCD<CR>" is sent to COM1 where CrLfmode=0 (the <CR> character is removed). So the device receives the string "ABCD<CR>", and gives the response "EFGH<CR>".

The data received by COM1 is "EFGH" where CrLfmode=0 (the <CR> character is removed and "EFGH<CR>" is sent to COM2 where CrLfmode=0 (the <CR> character is added). So the PC receives the string "EFGH<CR>".

Command transformation:
":01ABCD<CR>",PC -->"ABCD",com2 -->"ABCD<CR>",com1 -->
"ABCD<CR>", device
Response transformation:
"EFGH<CR>",device -->"EFGH",com1 -->"EFGH<CR>",com2 -->
"EFGH<CR>",PC

Situation 3:

I-7522A, COM2 CrLfmode=4, COM1 CrLfmode=4

The data received for COM2 is "ABCD<CR>" where CrLfmode=4 (nothing added) and "ABCD<CR>" is sent to COM1 where CrLfmode=4 (nothing added). So the device receives the string "ABCD<CR>", and gives the response "EFGH<CR>".

The data received for COM1 is "EFGH<CR>" where CrLfmode=4 (nothing added) and "EFGH<CR>" is sent for COM2 where CrLfmode=4 (nothing added). So the PC receives the string "EFGH<CR>".

Command transformation:
":01ABCD<CR>",PC -->"ABCD<CR>",com2 -->"ABCD<CR>",com1 -->
"ABCD<CR>", device
Response transformation:
"EFGH<CR>",device --> "EFGH<CR>",com1 -->"EFGH<CR>",com2
--> "EFGH<CR>",PC

[Timeout]

The $AAJN[timeout] command is only supported by firmware version 3.0 and above.

If the length of a command or response string is too long, or the data transmission time between the I-752N and peripheral RS-485 devices/RS-232 device is too long, the timeout value can be adjusted to ensure the amount of full data is received.
If the timeout value for the RS-232 COM Port is too small, the response part will be received by the 1K byte Queue buffer for RS-232 Ports.

The $AAU command can be used to read the buffer.

Note: More information about the timeout command can be found in Sec. 5.3.33.

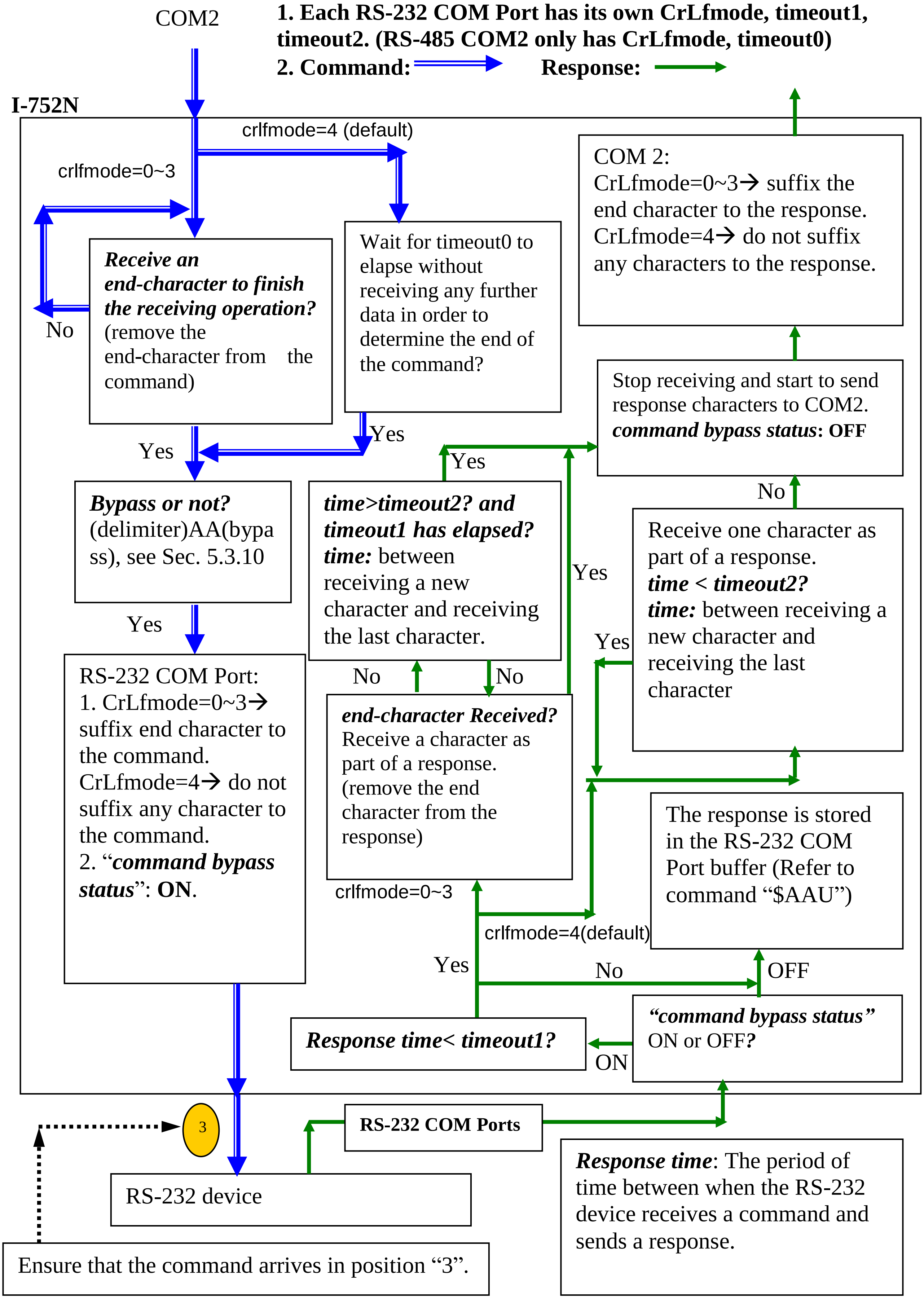
[Summary]

COM Ports receive data based CrLfmode rules.
Each port has its own data format, timeout values and CrLfmode. Setting correct parameters will ensure that the COM Ports work well and efficiently.

[Command and Response Flow Chart]

1. Each RS-232 COM Port has its own CrLfmode, timeout1, timeout2. (RS-485 COM2 only has CrLfmode, timeout0)

2. Command: ⇒ **Response:** →


COM2
I-752N
crlfmode=4 (default)
crlfmode=0~3
Receive an end-character to finish the receiving operation? (remove the end-character from the command)
No
Yes
Wait for timeout0 to elapse without receiving any further data in order to determine the end of the command?
Yes
COM 2: CrLfmode=0~3→ suffix the end character to the response. CrLfmode=4→ do not suffix any characters to the response.
Stop receiving and start to send response characters to COM2. command bypass status: OFF
Bypass or not? (delimiter)AA(bypass), see Sec. 5.3.10
Yes
time>timeout2? and timeout1 has elapsed? time: between receiving a new character and receiving the last character.
Yes
Yes
No
No
No
Receive one character as part of a response. time < timeout2? time: between receiving a new character and receiving the last character
Yes
RS-232 COM Port: 1. CrLfmode=0~3→ suffix end character to the command. CrLfmode=4→ do not suffix any character to the command. 2. "command bypass status": ON.
end-character Received? Receive a character as part of a response. (remove the end character from the response)
crlfmode=0~3
crlfmode=4(default)
The response is stored in the RS-232 COM Port buffer (Refer to command "$AAU")
Yes
No
OFF
Response time< timeout1?
"command bypass status" ON or OFF?
ON
3
RS-232 COM Ports
RS-232 device
Ensure that the command arrives in position "3".
Response time: The period of time between when the RS-232 device receives a command and sends a response.

2.6 Writing Program on 752N

Module	Manual Location
I-7521/I-7522/I-7523	http://www.icpdas.com/products/PAC/i-7188_7186/i-7188xc.htm ---> Click “Manual” ---> Download 7188xc_manual_english.pdf
I-7522A/I-7524 /I-7527	http://www.icpdas.com/products/PAC/i-7188_7186/i-7188xb.htm ---> Click “Manual” ---> Download 7188xb_manual_english.pdf

7188xc_manual_english.pdf and 7188xb_manual_english.pdf will teach how to programming. User can write a program and execute it on I-752N hardware. Here I-752N is as an embedded controller.

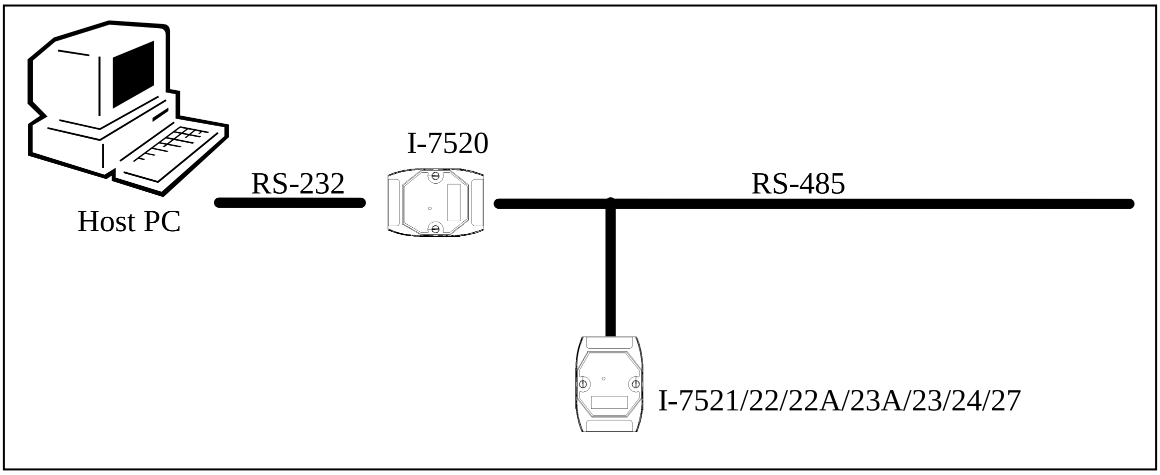
Chapter 3 Quick Start

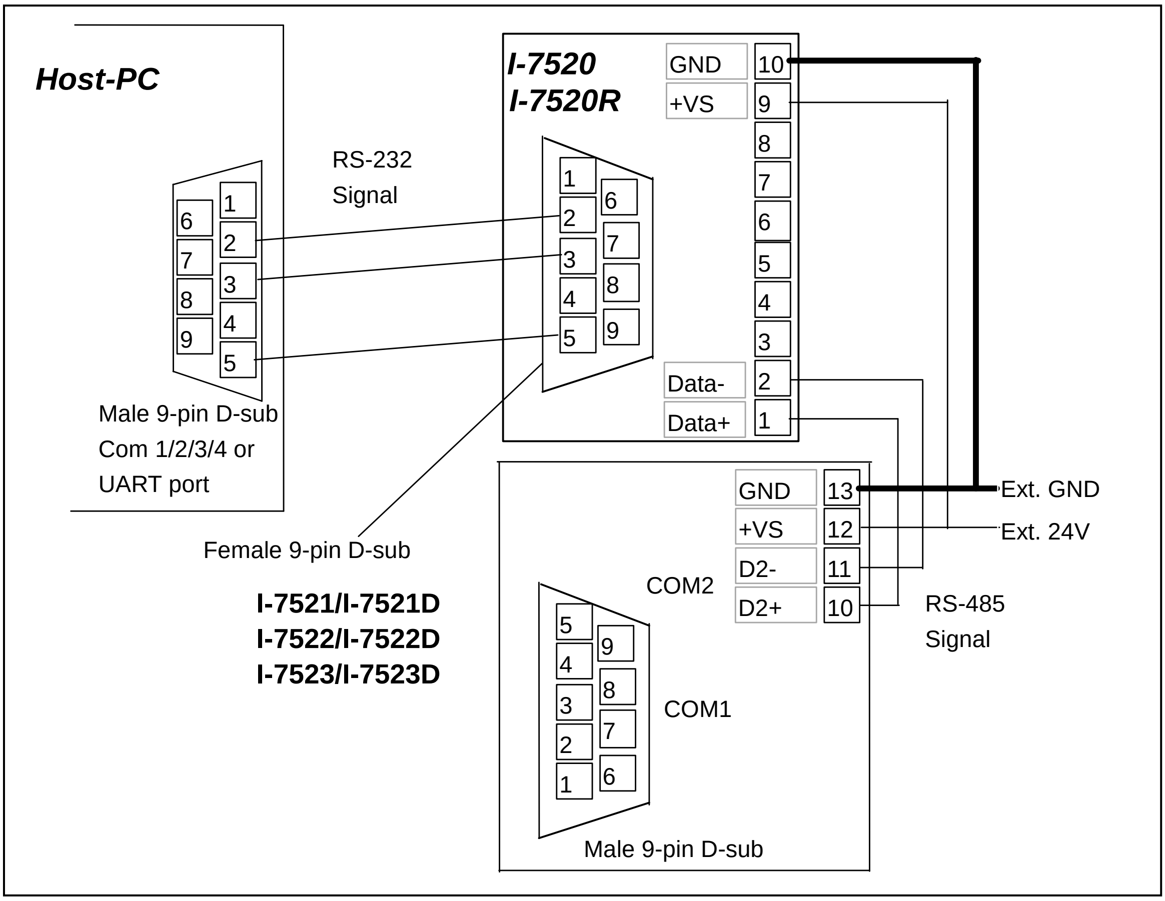
Factory default setting values about configurations for each COM port

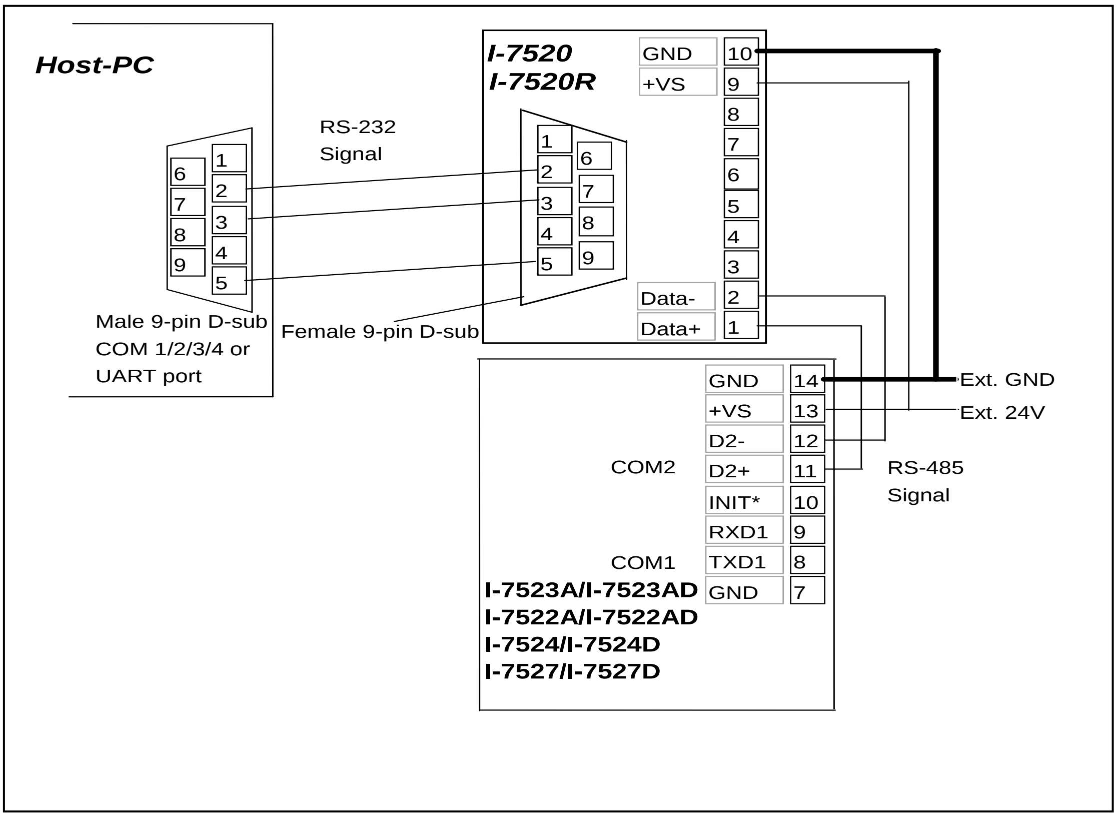
	COM1	COM2	COM3	COM4	COM5	COM6	COM7	COM8
Baud Rate	9600 bps	9600 bps	9600 bps	9600 bps	9600 bps	9600 bps	9600 bps	9600 bps
Data bit	8	8	8	8	8	8	8	8
Parity bit	None	None	None	None	None	None	None	None
Stop bit	1	1	1	1	1	1	1	1
Checksum	No	No	No	No	No	No	No	No
CrLfmode	4	4	4	4	4	4	4	4
delimiter (Refer to Sec 5.3.10)	**:**	**:**	**:**	**:**	**:**	**:**	**:**	**:**
WatchDog	disable	disable	disable	disable	disable	disable	disable	disable
Prefixed address (Refer to Sec.5.3.38)	disable		disable	disable	disable	disable	disable	disable
Bypass to COM2 (Refer to Sec.5.3.39)	disable		disable	disable	disable	disable	disable	disable

3.1 Connecting the I-752N module

Step 1: Connect the I-752N module to the RS-485 network as follows:

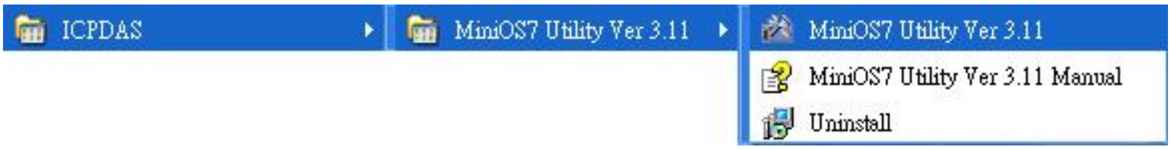




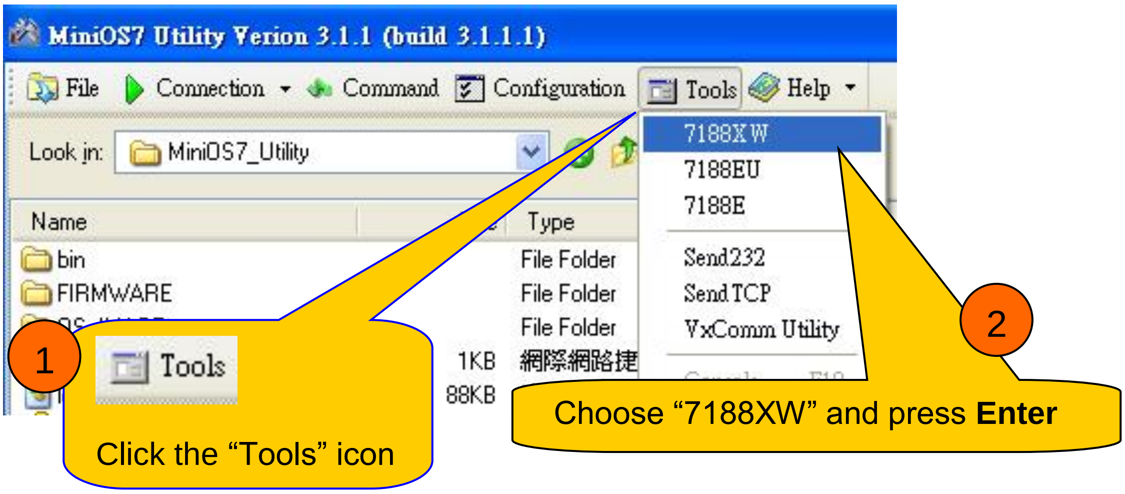


Step 2: Install the MiniOS7 Utility. Refer to Appendix A MiniOS7 Utility for more details.

Step 3: From the Windows START menu, go to the Programs/ICPDAS/MiniOS7 Utility Ver 3.11 folder and locate the MiniOS7 Utility Ver 3.11.

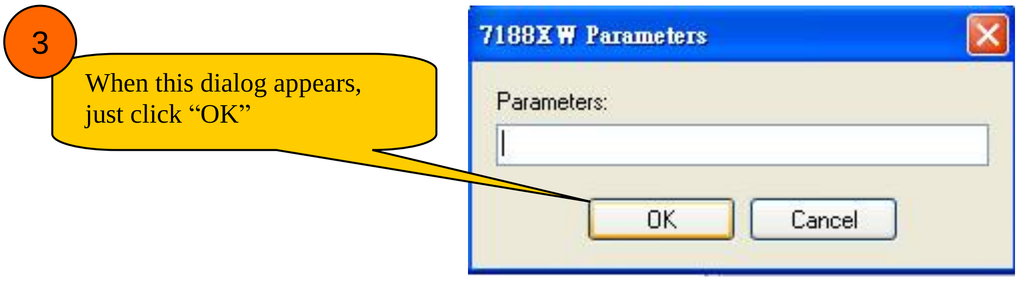


Step 4: Execute 7188xw.exe on the Host PC.



Click the “Tools” icon

Choose “7188XW” and press **Enter**

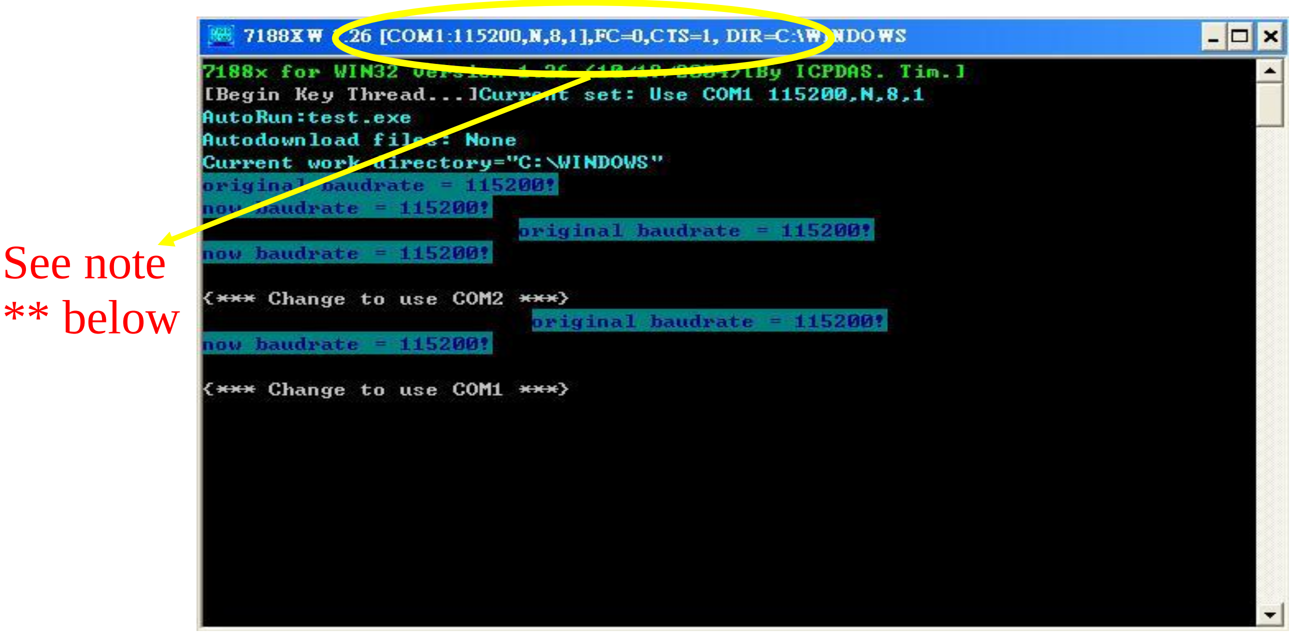


Step 5: Select the active COM port for the Host PC.

If the I-752N is connected to COM1 on the PC, then press **ALT + 1**

If the I-752N is connected to COM2 on the PC, then press **ALT + 2**

The following screen will be shown:



```
7188X W 1.26 [COM1:115200,N,8,1],FC=0,CTS=1, DIR=C:\WINDOWS
7188x for WIN32 Version 1.26 (10/18/2004)[By ICPDAS. Tim.]
[Begin Key Thread...]Current set: Use COM1 115200,N,8,1
AutoRun:test.exe
Autodownload files: None
Current work directory="C:\WINDOWS"
original baudrate = 115200!
now baudrate = 115200!
                                original baudrate = 115200!
now baudrate = 115200!

{*** Change to use COM2 ***}
                                 original baudrate = 115200!
now baudrate = 115200!

{*** Change to use COM1 ***}
```

**: Illustrates a real time configuration for the 7188xw. If the configuration is not 9600, N, 8, 1, refer to Steps 6 to 9 below to make the correct settings. Otherwise, go directly to Step 10.

Step 6: Set the Baud Rate of the 7188xw to 9600.

Press **ALT + C**

Type **b9600**

Press **ENTER** to confirm

Step 7: Set the Parity bit of the 7188xw to N.

Press **ALT + C**

Type **n**

Press **ENTER** to confirm

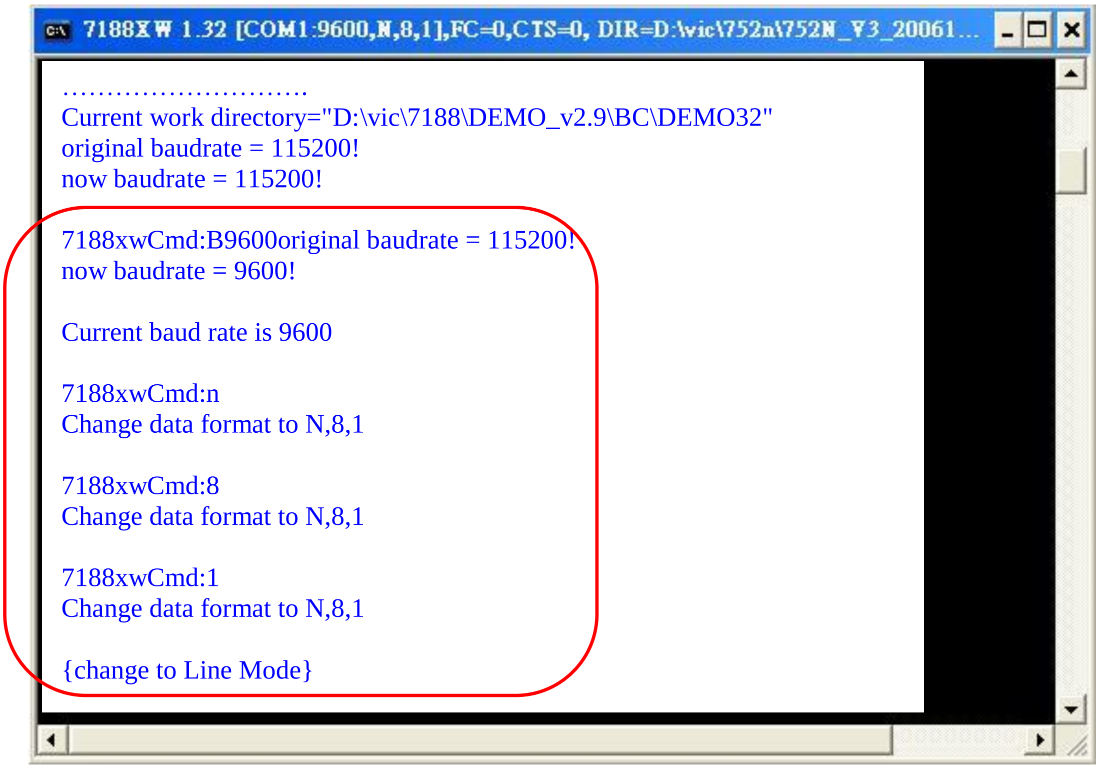
Step 8: Set the Data bit of the 7188xw to 8.

Press **ALT + C**

Type **8**
Press **ENTER** to confirm

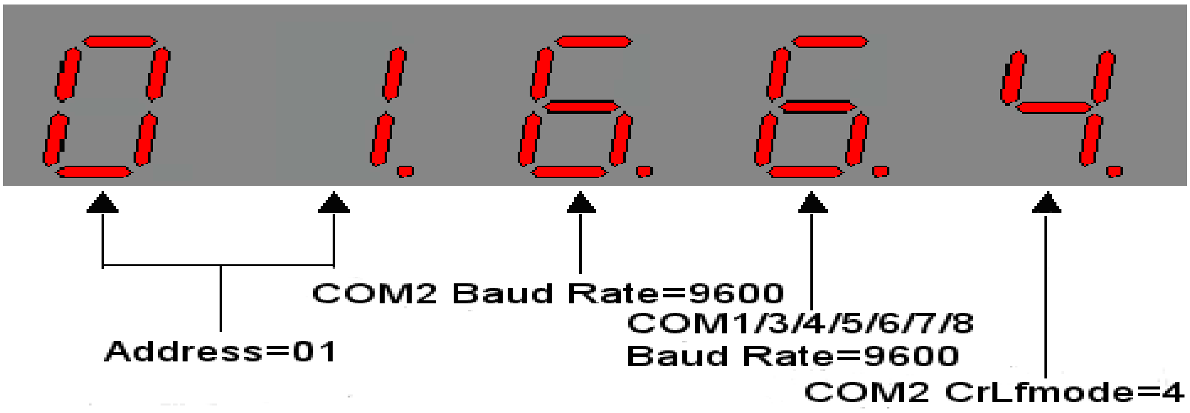
Step 9: Set the Stop bit of the 7188xw to 1.
Press **ALT + C**. Type **1**
Press **ENTER** to confirm

Step 10: Change the 7188xw to Command Line Mode.
Press **ALT+ L**
The following screen will be shown:


7188XW 1.32 [COM1:9600,N,8,1],FC=0,CTS=0, DIR=D:\vic\752n\752N_V3_20061...

```
………………………….
Current work directory="D:\vic\7188\DEMO_v2.9\BC\DEMO32"
original baudrate = 115200!
now baudrate = 115200!

7188xwCmd:B9600original baudrate = 115200!
now baudrate = 9600!

Current baud rate is 9600

7188xwCmd:n
Change data format to N,8,1

7188xwCmd:8
Change data format to N,8,1

7188xwCmd:1
Change data format to N,8,1

{change to Line Mode}
```

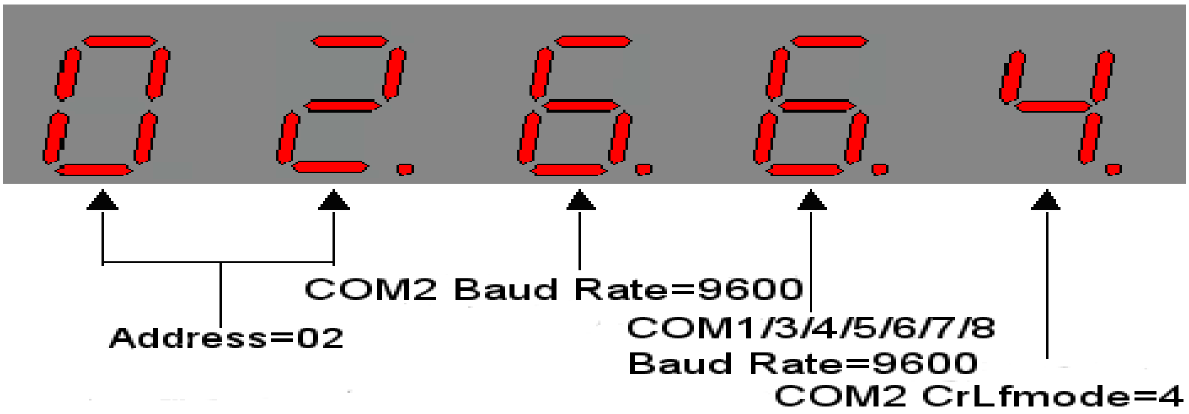
Step 11: Switch on the power to the I-752N module (disconnect the INIT* and GND pins) and check that the display on the 5-digit, 7-seg LED is as below: (For firmware version 3.02 and above, see Note*)



Step 12: Retrieve the Module Name of the I-752N module.
Type the command → **$01M**
Press **ENTER** to send the command to the I-752N module
Check that the module returns → **!017521** (for example: I-7521)

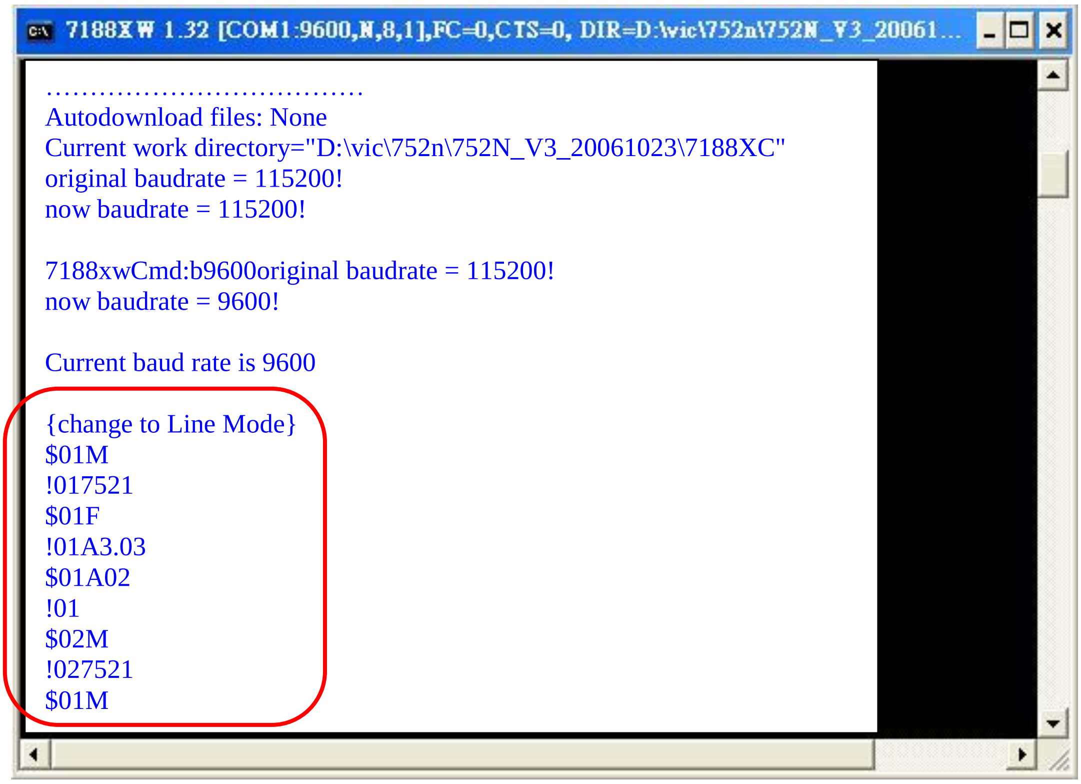
Step 13: Retrieve the Version number of the I-752N module.
Type the command → **$01F**
Press **ENTER** to send the command to the I-752N module
Check that the module returns→ **!01(A1.0/A2.0/A3.0/A3.02/..)**

Step 14: Change the Module Address of the I-752N module.
Type the command → **$01A02**
Press **ENTER** to send the command to the I-752N module
Check that the display on the 5-digit, 7-seg LED is as follows:
(For firmware version 3.02 and above, see Note*)



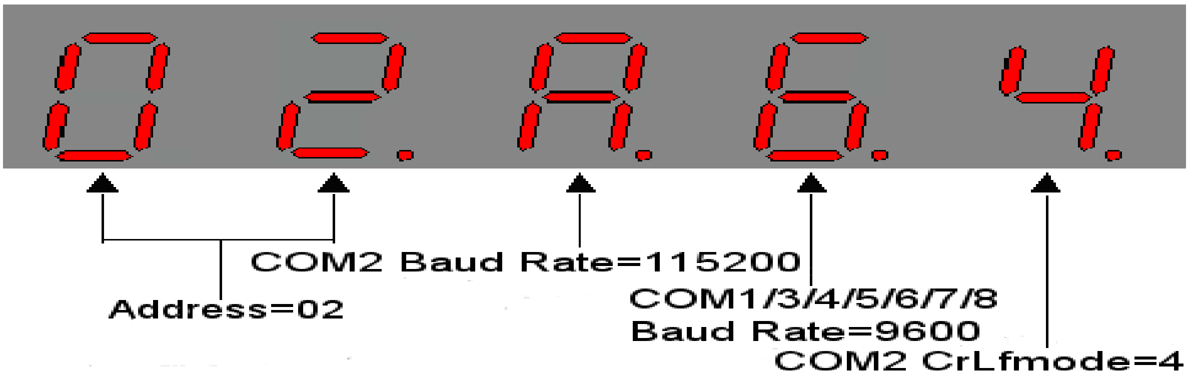
Type the command → **$02M**
Press **ENTER** to send the command to the I-752N module
Check that the module returns → **!027521** (for example: I-7521)

Type the command → **$01M**
Press **ENTER** to send the command to the I-752N module
Check that the module returns → **No response** (for example: I-7521)



```
7188XW 1.32 [COM1:9600,N,8,1],FC=0,CTS=0, DIR=D:\vic\752n\752N_V3_20061...
....................................
Autodownload files: None
Current work directory="D:\vic\752n\752N_V3_20061023\7188XC"
original baudrate = 115200!
now baudrate = 115200!

7188xwCmd:b9600original baudrate = 115200!
now baudrate = 9600!

Current baud rate is 9600

{change to Line Mode}
$01M
!017521
$01F
!01A3.03
$01A02
!01
$02M
!027521
$01M
```

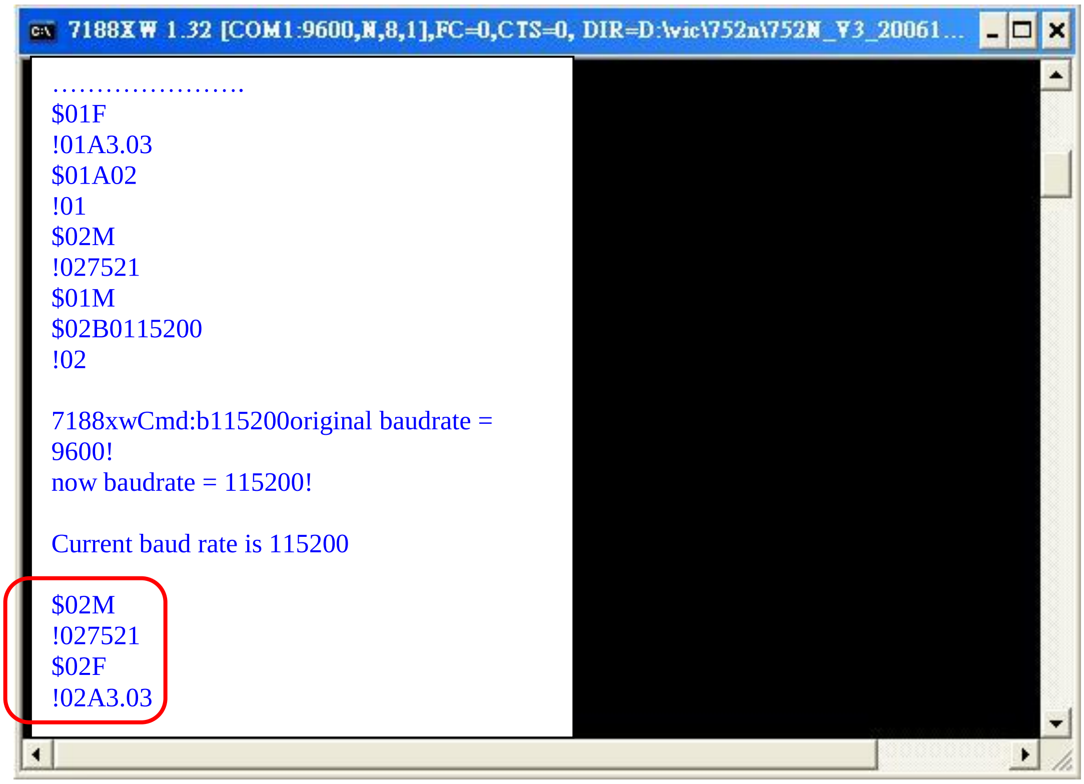
Step 15: Change the Baud Rate for COM2 of the I-752N module
Type the command → **$02B0115200**
Press **ENTER** to send the command to the I-752N module
Check that the display on the 5-digit, 7-seg LED is as follows:
(For firmware version 3.02 and above, see Note*)



Press **ALT + C.**
Type **b115200** to change the Baud Rate of the PC side.
Press **ENTER** to confirm the Baud Rate=115200.

Type the command → **$02M**
Press **ENTER** to send the command to the I-752N module.
Check that the module returns → **!027521** (for example I-7521).

Type the command → **$02F**
Press **ENTER** to send the command to the I-752N module
Check that the module returns → **!02(A1.0/A2.0/A3.0/A3.02/..)** (For example: I-7521).


7188XW 1.32 [COM1:9600,N,8,1],FC=0,CTS=0, DIR=D:\wic\752n\752N_V3_20061...

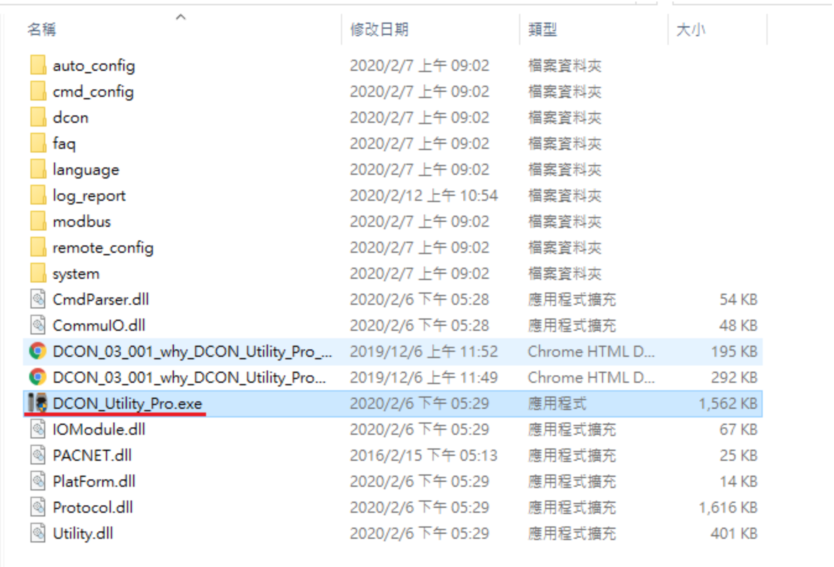
```
……………………
$01F
!01A3.03
$01A02
!01
$02M
!027521
$01M
$02B0115200
!02

7188xwCmd:b115200original baudrate =
9600!
now baudrate = 115200!

Current baud rate is 115200

$02M
!027521
$02F
!02A3.03
```

Note*: Refer to Sec. 5.3.2 for more details regarding short codes for Baud Rates.

3.2 Configure the Module using DCON Utility Pro

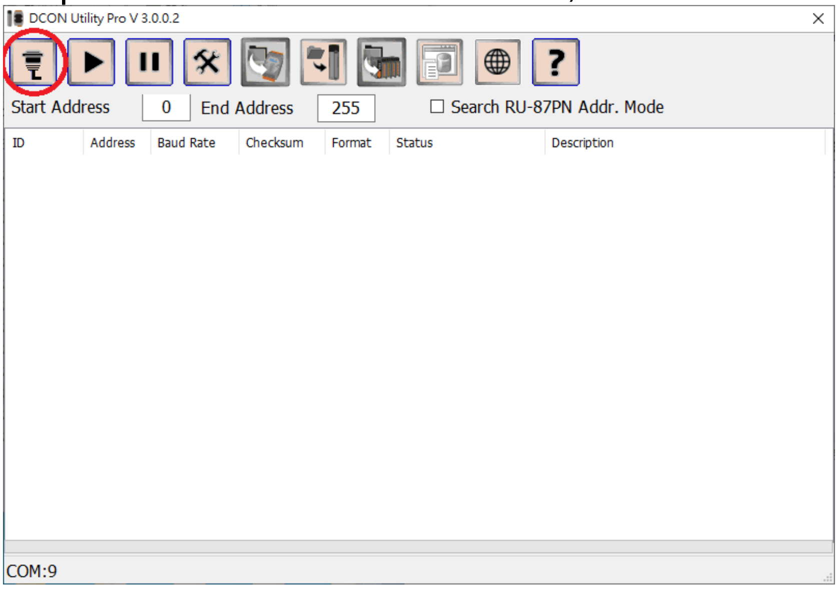
Step 1: Download the DCON Utility Pro utility software from:
http://www.icpdas.com/root/product/solutions/software/utilities/dcon_utiltiy_pro.html

Step 2: After downloading the software, unzip the file to a safe location and directly launch the DCON_Utility_Pro.exe file. There is no need for any installation.



名稱	修改日期	類型	大小
auto_config	2020/2/7 上午 09:02	檔案資料夾	
cmd_config	2020/2/7 上午 09:02	檔案資料夾	
dcon	2020/2/7 上午 09:02	檔案資料夾	
faq	2020/2/7 上午 09:02	檔案資料夾	
language	2020/2/7 上午 09:02	檔案資料夾	
log_report	2020/2/12 上午 10:54	檔案資料夾	
modbus	2020/2/7 上午 09:02	檔案資料夾	
remote_config	2020/2/7 上午 09:02	檔案資料夾	
system	2020/2/7 上午 09:02	檔案資料夾	
CmdParser.dll	2020/2/6 下午 05:28	應用程式擴充	54 KB
CommuIO.dll	2020/2/6 下午 05:28	應用程式擴充	48 KB
DCON_03_001_why_DCON_Utility_Pro_...	2019/12/6 上午 11:52	Chrome HTML D...	195 KB
DCON_03_001_why_DCON_Utility_Pro...	2019/12/6 上午 11:49	Chrome HTML D...	292 KB
DCON_Utility_Pro.exe	2020/2/6 下午 05:29	應用程式	1,562 KB
IOModule.dll	2020/2/6 下午 05:29	應用程式擴充	67 KB
PACNET.dll	2016/2/15 下午 05:13	應用程式擴充	25 KB
PlatForm.dll	2020/2/6 下午 05:29	應用程式擴充	14 KB
Protocol.dll	2020/2/6 下午 05:29	應用程式擴充	1,616 KB
Utility.dll	2020/2/6 下午 05:29	應用程式擴充	401 KB

Step 3: Click the COM Port button, as shown below:

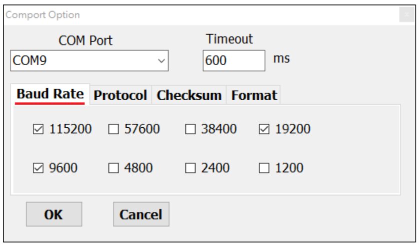


Start Address 0 End Address 255 ☐ Search RU-87PN Addr. Mode

ID	Address	Baud Rate	Checksum	Format	Status	Description

COM:9

Set the COM Port, Timeout, Baud Rate, and Protocol parameters using the values indicated in the following images.



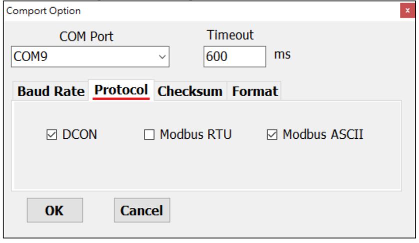
Comport Option

COM Port: COM9 Timeout: 600 ms

Baud Rate | Protocol | Checksum | Format

☑ 115200 ☐ 57600 ☐ 38400 ☑ 19200

☑ 9600 ☐ 4800 ☐ 2400 ☐ 1200

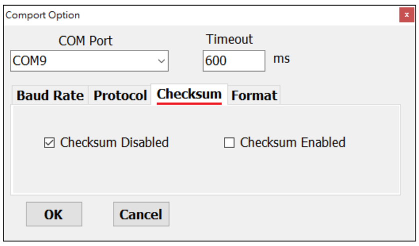


Comport Option

COM Port: COM9 Timeout: 600 ms

Baud Rate | Protocol | Checksum | Format

☑ DCON ☐ Modbus RTU ☑ Modbus ASCII

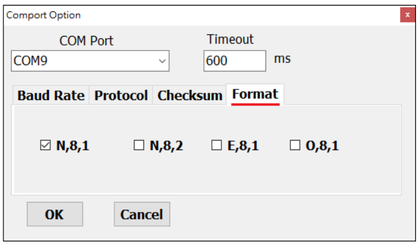


Comport Option

COM Port: COM9 Timeout: 600 ms

Baud Rate | Protocol | Checksum | Format

☑ Checksum Disabled ☐ Checksum Enabled



Comport Option

COM Port: COM9 Timeout: 600 ms

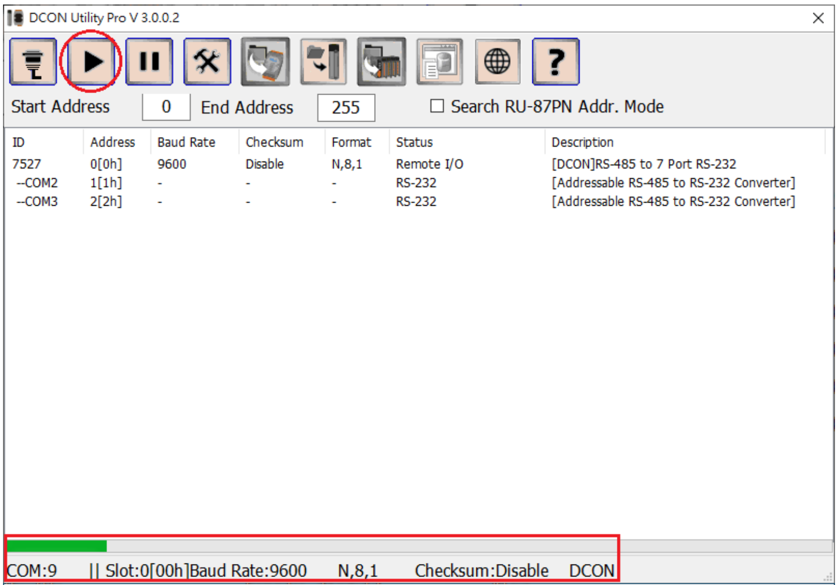
Baud Rate | Protocol | Checksum | Format

☑ N,8,1 ☐ N,8,2 ☐ E,8,1 ☐ O,8,1

* If you don’t remember the configuration parameter, you can try to use a wise to connect the pin *INIT and GND to let the module go to initial mode which will use default communication parameters:

Module Address: 0,
Baud rate: 9600 bps,
Checksum: disable,
Data Format: N81,
Protocol: DCON.

Step 4: You can now use the Search function in DCON Utility Pro by clicking the search button, which is shown in the image below. Once the function receives a response from the module, the module information will be displayed in the software.

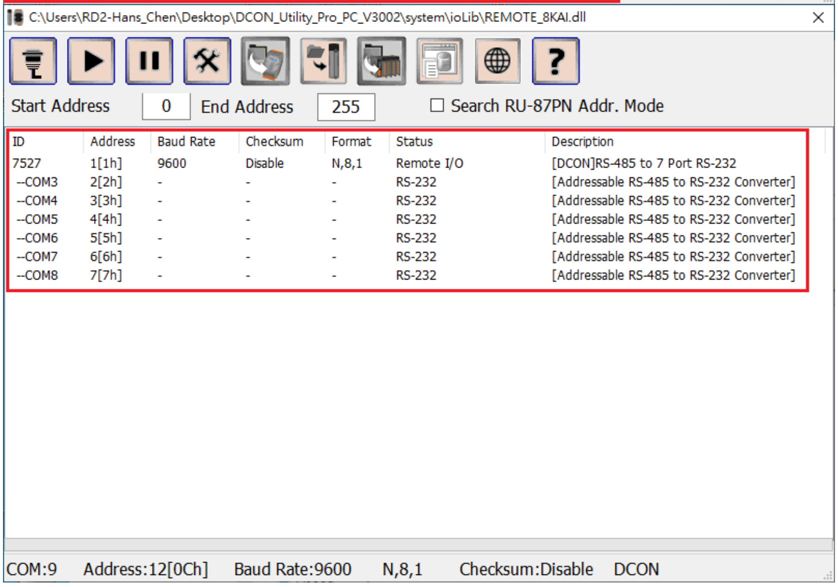


DCON Utility Pro V 3.0.0.2

Start Address 0 End Address 255 □ Search RU-87PN Addr. Mode

ID	Address	Baud Rate	Checksum	Format	Status	Description
7527	0[0h]	9600	Disable	N,8,1	Remote I/O	[DCON]RS-485 to 7 Port RS-232
--COM2	1[1h]	-	-	-	RS-232	[Addressable RS-485 to RS-232 Converter]
--COM3	2[2h]	-	-	-	RS-232	[Addressable RS-485 to RS-232 Converter]

COM:9 || Slot:0[00h]Baud Rate:9600 N,8,1 Checksum:Disable DCON



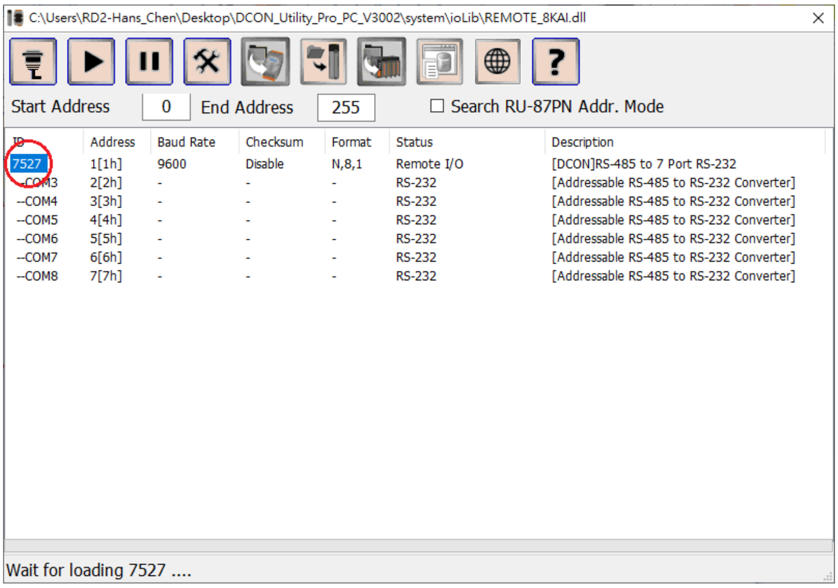
C:\Users\RD2-Hans_Chen\Desktop\DCON_Utility_Pro_PC_V3002\system\ioLib\REMOTE_8KAI.dll

Start Address 0 End Address 255 □ Search RU-87PN Addr. Mode

ID	Address	Baud Rate	Checksum	Format	Status	Description
7527	1[1h]	9600	Disable	N,8,1	Remote I/O	[DCON]RS-485 to 7 Port RS-232
--COM3	2[2h]	-	-	-	RS-232	[Addressable RS-485 to RS-232 Converter]
--COM4	3[3h]	-	-	-	RS-232	[Addressable RS-485 to RS-232 Converter]
--COM5	4[4h]	-	-	-	RS-232	[Addressable RS-485 to RS-232 Converter]
--COM6	5[5h]	-	-	-	RS-232	[Addressable RS-485 to RS-232 Converter]
--COM7	6[6h]	-	-	-	RS-232	[Addressable RS-485 to RS-232 Converter]
--COM8	7[7h]	-	-	-	RS-232	[Addressable RS-485 to RS-232 Converter]

COM:9 Address:12[0Ch] Baud Rate:9600 N,8,1 Checksum:Disable DCON

Step 5: Click the ID for the required module to display the configuration UI.

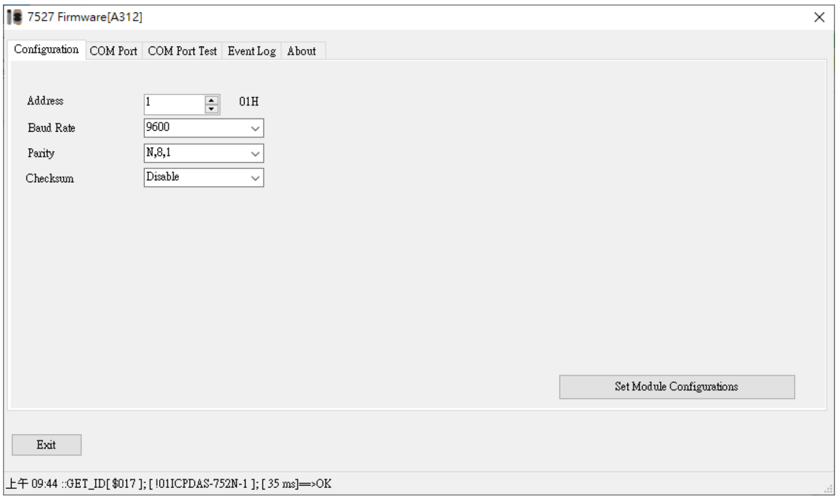


C:\Users\RD2-Hans_Chen\Desktop\DCON_Utility_Pro_PC_V3002\system\ioLib\REMOTE_8KAI.dll

Start Address 0 End Address 255 □ Search RU-87PN Addr. Mode

ID	Address	Baud Rate	Checksum	Format	Status	Description
7527	1[1h]	9600	Disable	N,8,1	Remote I/O	[DCON]RS-485 to 7 Port RS-232
--COM3	2[2h]	-	-	-	RS-232	[Addressable RS-485 to RS-232 Converter]
--COM4	3[3h]	-	-	-	RS-232	[Addressable RS-485 to RS-232 Converter]
--COM5	4[4h]	-	-	-	RS-232	[Addressable RS-485 to RS-232 Converter]
--COM6	5[5h]	-	-	-	RS-232	[Addressable RS-485 to RS-232 Converter]
--COM7	6[6h]	-	-	-	RS-232	[Addressable RS-485 to RS-232 Converter]
--COM8	7[7h]	-	-	-	RS-232	[Addressable RS-485 to RS-232 Converter]

Wait for loading 7527 ....



7527 Firmware[A312]

Address 1 01H

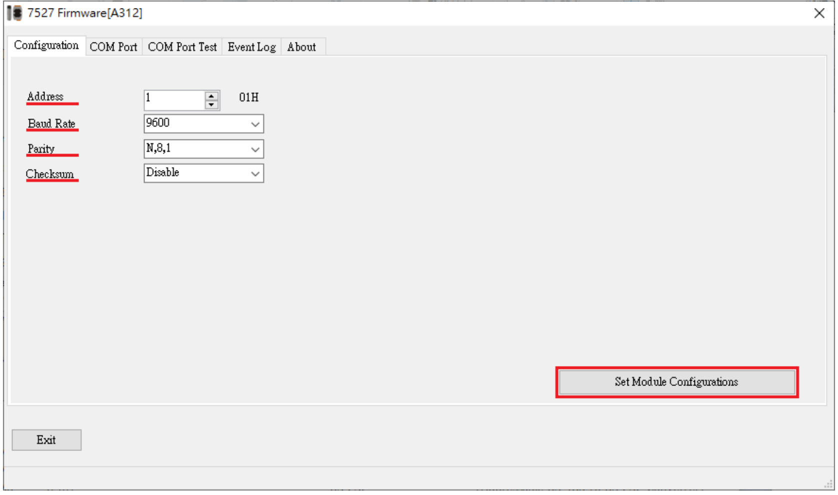
Baud Rate 9600

Parity N,8,1

Checksum Disable

上午 09:44 ::GET_ID[$017]; [!01ICPDAS-752N-1]; [35 ms]==>OK

Step 6: The Configuration page is used to set the parameters for an RS-485 interface (COM2). You can change the information for the module address, Baud Rate, parity and checksum here.



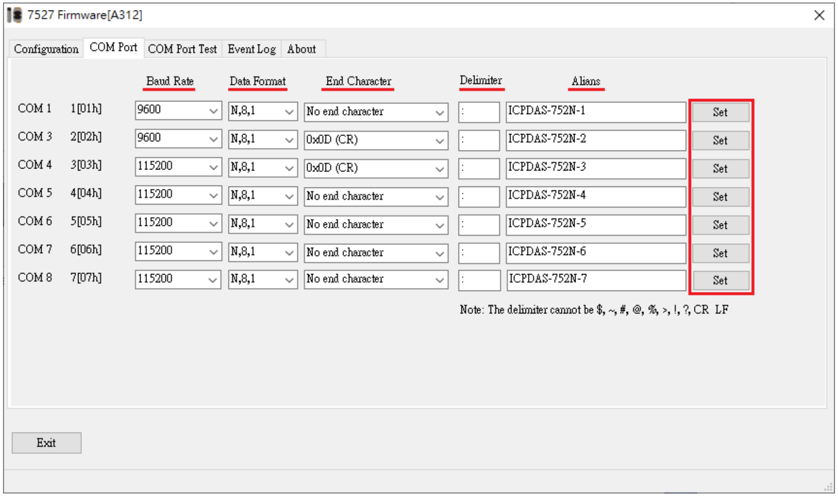
Address 1 01H

Baud Rate 9600

Parity N,8,1

Checksum Disable

Step 7: The Configuration page is also used to set the parameters for an RS-232 interface. You can change the information for the Baud Rate, data format, end character, delimiter and RS-232 alias name here.

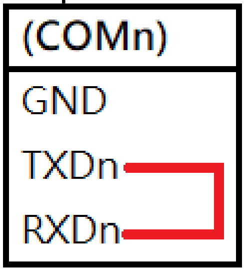


		Baud Rate	Data Format	End Character	Delimiter	Alians	
COM 1	1[01h]	9600	N,8,1	No end character	:	ICPDAS-752N-1	Set
COM 3	2[02h]	9600	N,8,1	0x0D (CR)	:	ICPDAS-752N-2	Set
COM 4	3[03h]	115200	N,8,1	0x0D (CR)	:	ICPDAS-752N-3	Set
COM 5	4[04h]	115200	N,8,1	No end character	:	ICPDAS-752N-4	Set
COM 6	5[05h]	115200	N,8,1	No end character	:	ICPDAS-752N-5	Set
COM 7	6[06h]	115200	N,8,1	No end character	:	ICPDAS-752N-6	Set
COM 8	7[07h]	115200	N,8,1	No end character	:	ICPDAS-752N-7	Set

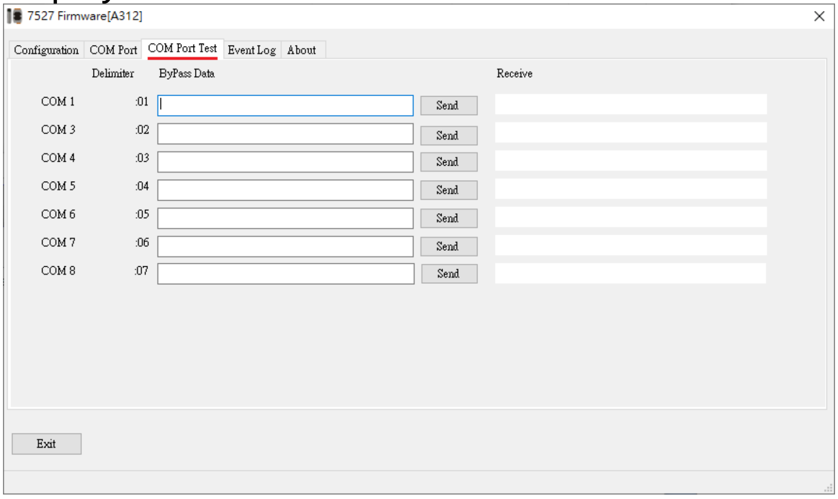
Note: The delimiter cannot be $, ~, #, @, %, >, !, ?, CR LF

COM Port function Test:

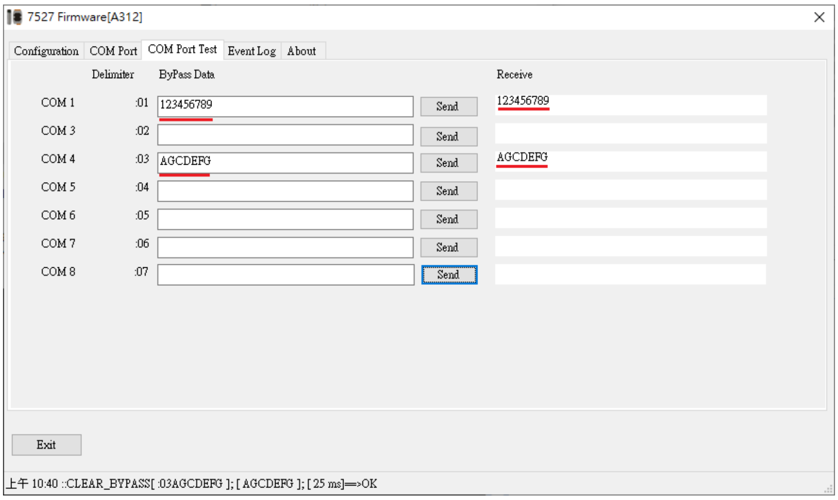
Step 1: Use a wire to connect the TXD and RXD pins on an RS-232 interface.



Click the COM Port Test tab and input a string in the input text field, then click the Send button. For example, you can input “1234567” in the input text field for COM1 and then click the Send button. You will then see the string “1234567” displayed in the Receive text field.



	Delimiter	ByPass Data		Receive
COM 1	:01		Send	
COM 3	:02		Send	
COM 4	:03		Send	
COM 5	:04		Send	
COM 6	:05		Send	
COM 7	:06		Send	
COM 8	:07		Send	

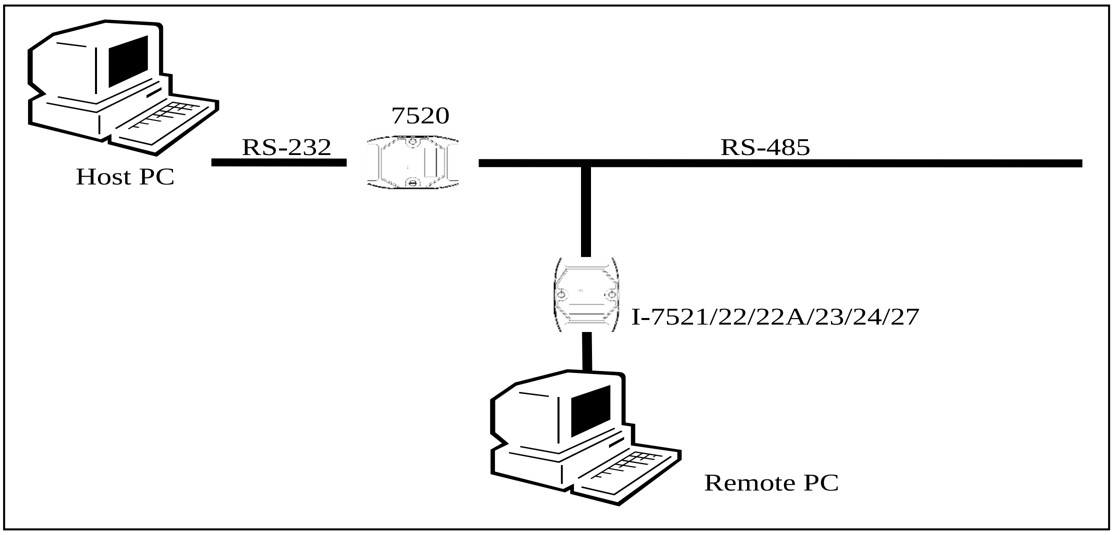


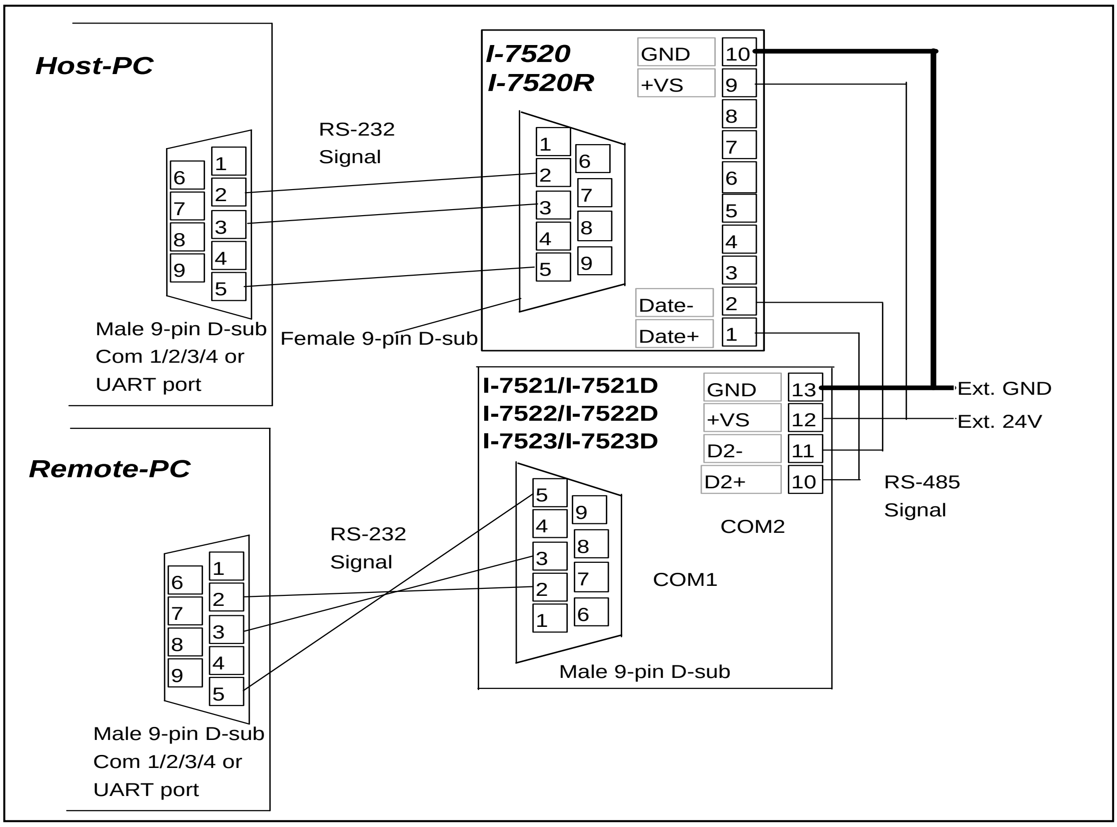
	Delimiter	ByPass Data		Receive
COM 1	:01	123456789	Send	123456789
COM 3	:02		Send	
COM 4	:03	AGCDEFG	Send	AGCDEFG
COM 5	:04		Send	
COM 6	:05		Send	
COM 7	:06		Send	
COM 8	:07		Send	

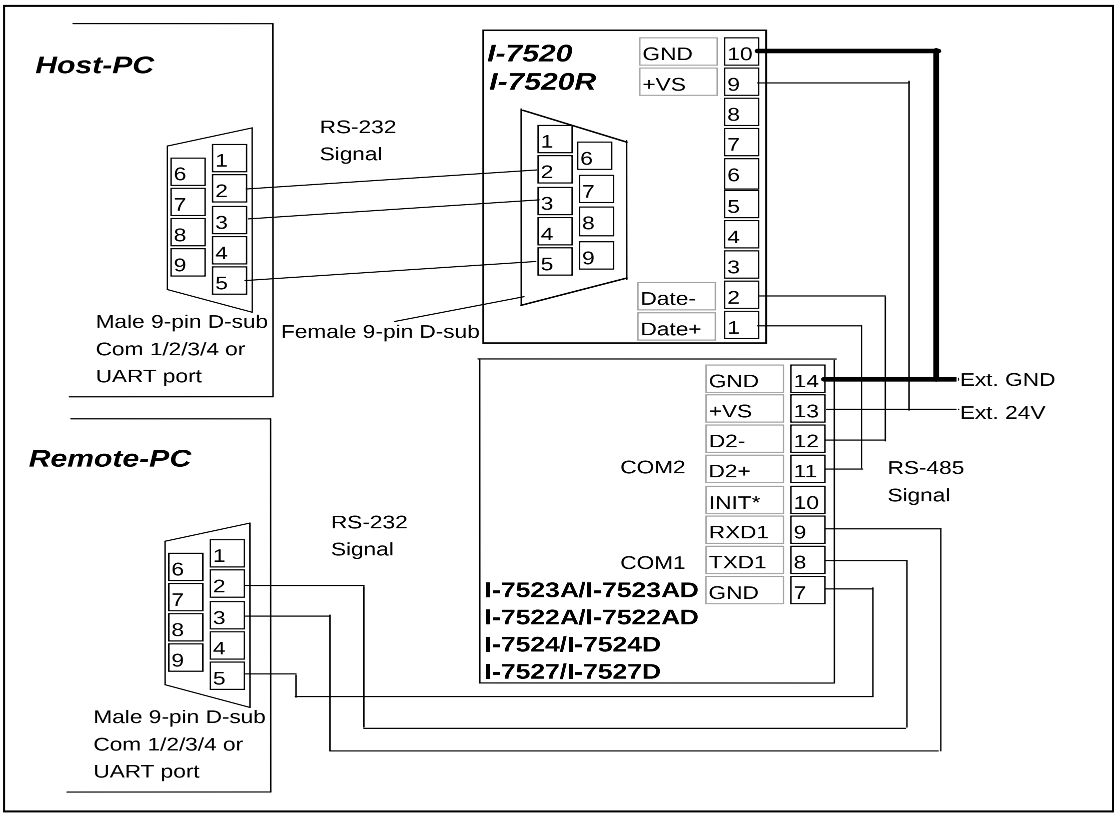
```
上午 10:40 ::CLEAR_BYPASS[ :03AGCDEFG ]; [ AGCDEFG ]; [ 25 ms]==>OK
```

3.3 Connecting to a Single Remote RS-232 Device

Step 1: Connect the I-752N module to the RS-485 network and the remote PC as follows:



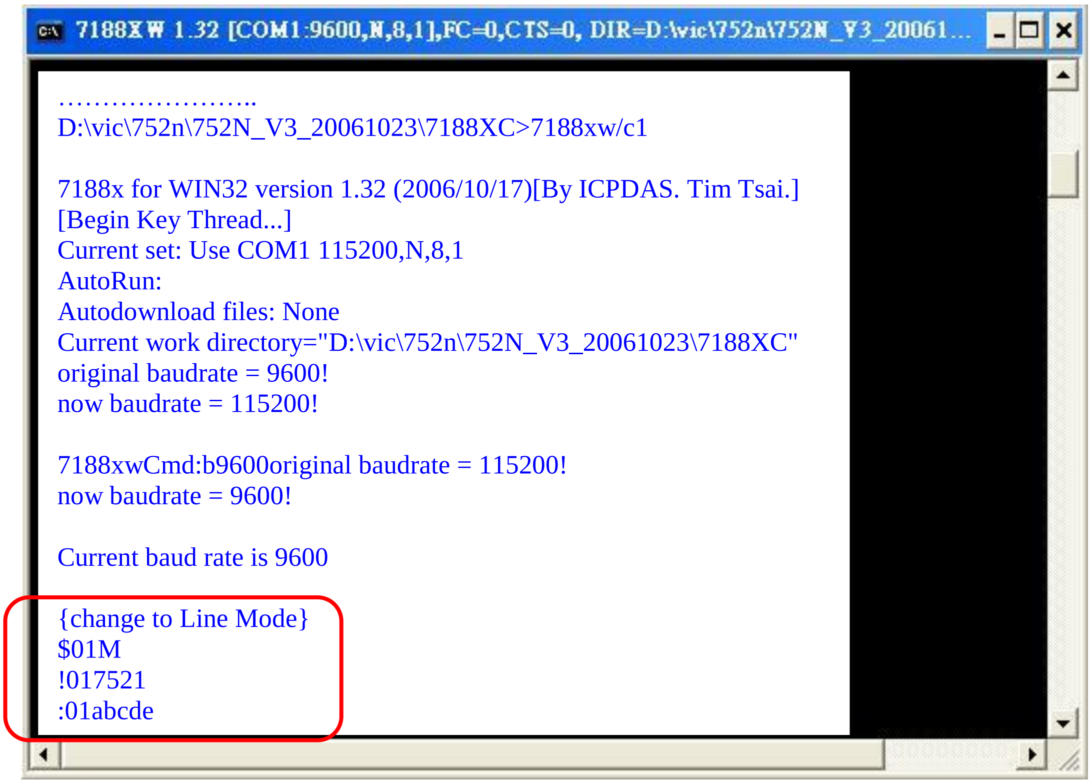




Step 2: Refer to Step 2 to 4 in Sec3.1 to execute 7188xw.exe on the Host PC
Refer to Steps 5 to 9 in Sec.3.1 for details to know how to change the COM port and status settings to **9600, N, 8, 1**

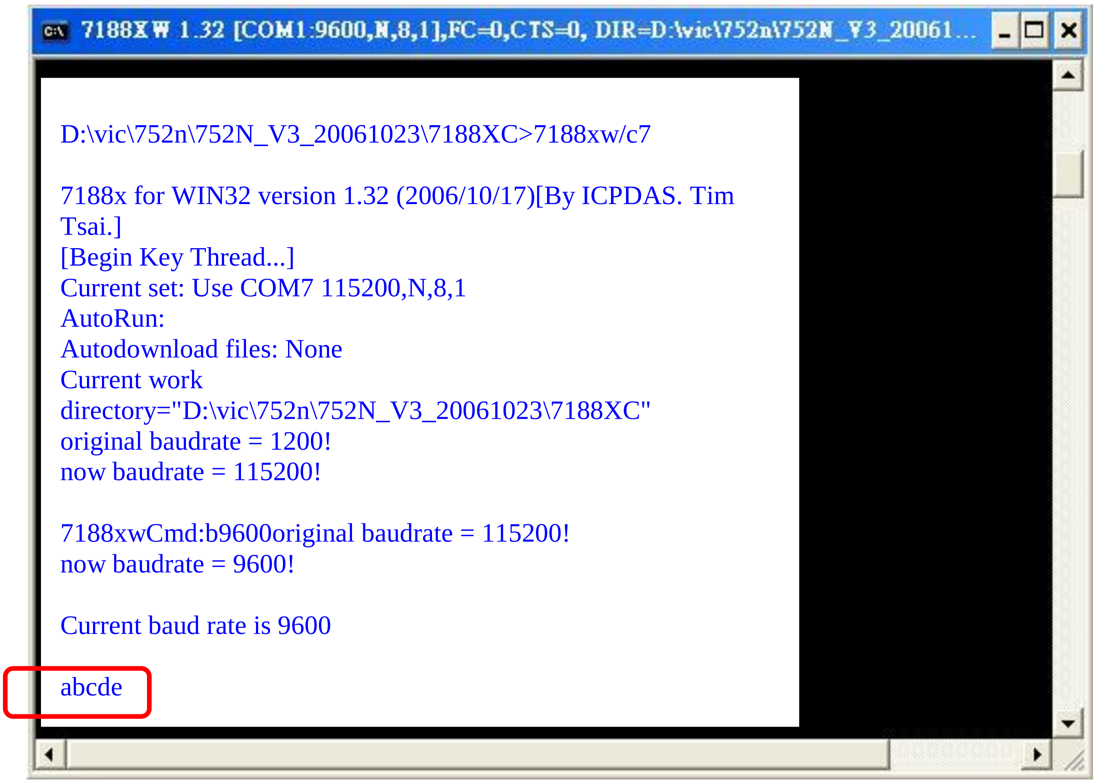
Step 3: Execute 7188xw.exe on the Remote PC
Change the COM port and status settings to **9600, N, 8, 1**

Step 4: The Host PC sends "**abcde**" string to the Remote PC
Type **:01abcde**
Press **ENTER** to send the command string to the I-752N module
Check that the response string on the Remote PC is **abcde**
The following screen should be shown on the Host PC:


7188XW 1.32 [COM1:9600,N,8,1],FC=0,CTS=0, DIR=D:\vic\752n\752N_V3_20061...

```
……………………..
D:\vic\752n\752N_V3_20061023\7188XC>7188xw/c1

7188x for WIN32 version 1.32 (2006/10/17)[By ICPDAS. Tim Tsai.]
[Begin Key Thread...]
Current set: Use COM1 115200,N,8,1
AutoRun:
Autodownload files: None
Current work directory="D:\vic\752n\752N_V3_20061023\7188XC"
original baudrate = 9600!
now baudrate = 115200!

7188xwCmd:b9600original baudrate = 115200!
now baudrate = 9600!

Current baud rate is 9600

{change to Line Mode}
$01M
!017521
:01abcde
```

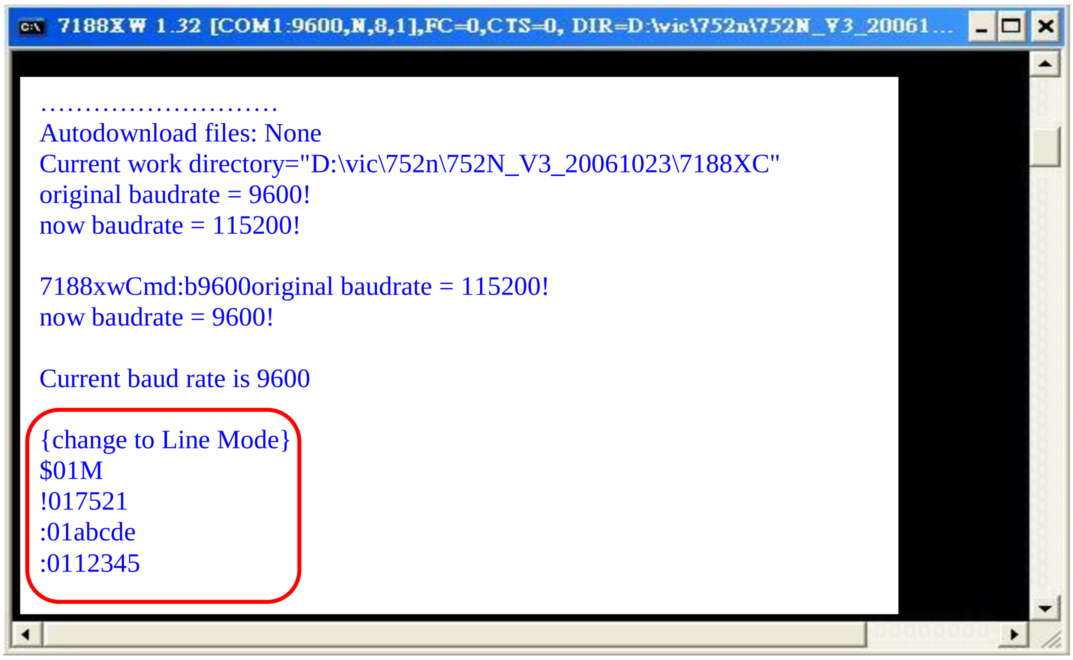
The following screen should be shown on the Remote PC:


7188XW 1.32 [COM1:9600,N,8,1],FC=0,CTS=0, DIR=D:\vic\752n\752N_V3_20061...

```
D:\vic\752n\752N_V3_20061023\7188XC>7188xw/c7

7188x for WIN32 version 1.32 (2006/10/17)[By ICPDAS. Tim
Tsai.]
[Begin Key Thread...]
Current set: Use COM7 115200,N,8,1
AutoRun:
Autodownload files: None
Current work
directory="D:\vic\752n\752N_V3_20061023\7188XC"
original baudrate = 1200!
now baudrate = 115200!

7188xwCmd:b9600original baudrate = 115200!
now baudrate = 9600!

Current baud rate is 9600

abcde
```

Step 5: Send "**12345**" string from the Host PC to the Remote PC

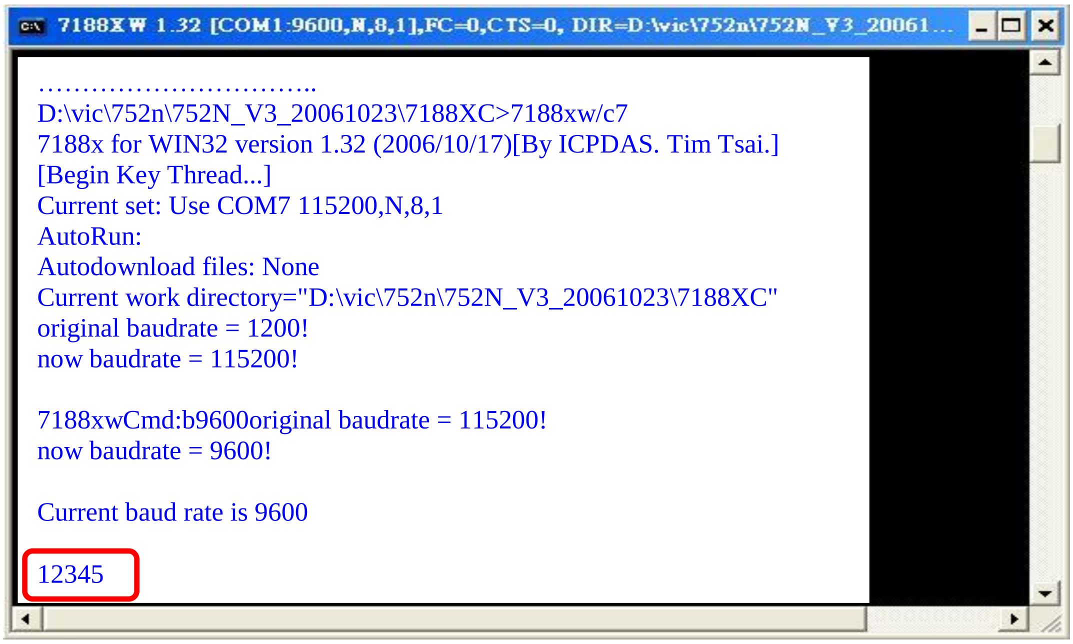
Type **:0112345**
Press **ENTER** to send the command string to the I-752N module
Check that the response string on the Remote PC is **12345**
The following screen should be shown on the Host PC:



7188XW 1.32 [COM1:9600,N,8,1],FC=0,CTS=0, DIR=D:\vic\752n\752N_V3_20061...

```
……………………….
Autodownload files: None
Current work directory="D:\vic\752n\752N_V3_20061023\7188XC"
original baudrate = 9600!
now baudrate = 115200!

7188xwCmd:b9600original baudrate = 115200!
now baudrate = 9600!

Current baud rate is 9600

{change to Line Mode}
$01M
!017521
:01abcde
:0112345
```

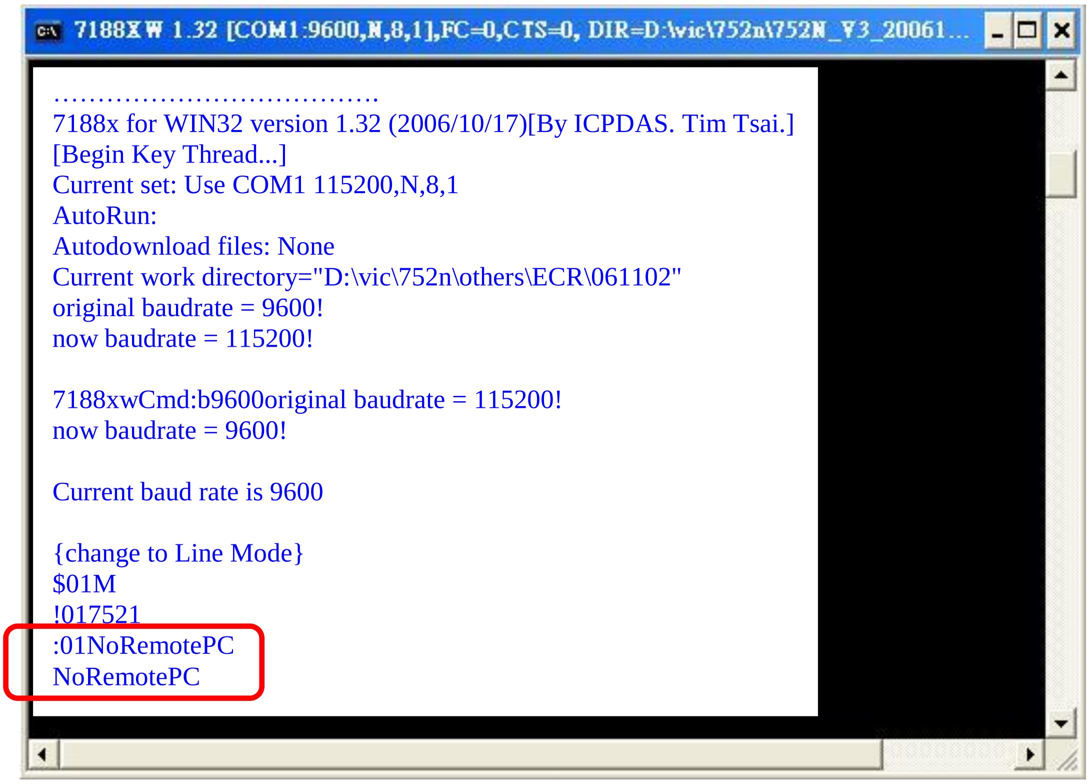
The following screen should be shown on the Remote PC:



7188XW 1.32 [COM1:9600,N,8,1],FC=0,CTS=0, DIR=D:\vic\752n\752N_V3_20061...

```
…………………………..
D:\vic\752n\752N_V3_20061023\7188XC>7188xw/c7
7188x for WIN32 version 1.32 (2006/10/17)[By ICPDAS. Tim Tsai.]
[Begin Key Thread...]
Current set: Use COM7 115200,N,8,1
AutoRun:
Autodownload files: None
Current work directory="D:\vic\752n\752N_V3_20061023\7188XC"
original baudrate = 1200!
now baudrate = 115200!

7188xwCmd:b9600original baudrate = 115200!
now baudrate = 9600!

Current baud rate is 9600

12345
```

Note: If no Remote PC is available, the test can be performed by connecting TxD and RxD to the same COM port.

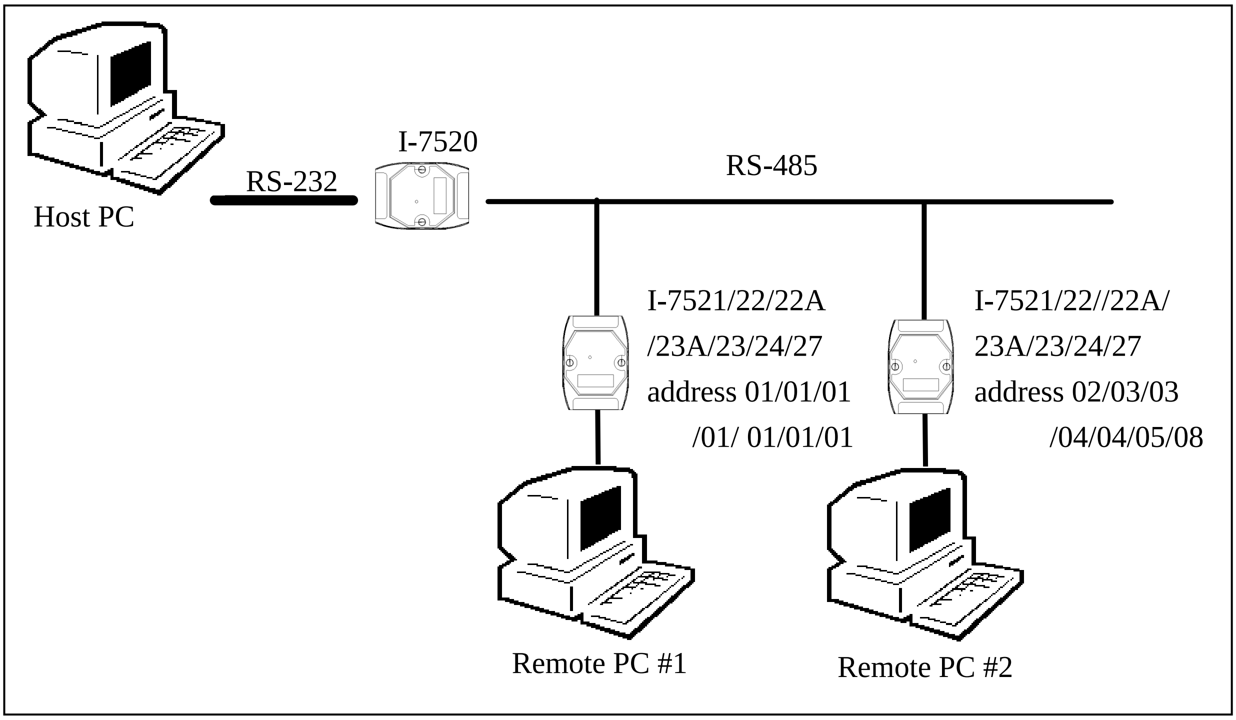

7188XW 1.32 [COM1:9600,N,8,1],FC=0,CTS=0, DIR=D:\vic\752n\752N_V3_20061...

```
………………………………….
7188x for WIN32 version 1.32 (2006/10/17)[By ICPDAS. Tim Tsai.]
[Begin Key Thread...]
Current set: Use COM1 115200,N,8,1
AutoRun:
Autodownload files: None
Current work directory="D:\vic\752n\others\ECR\061102"
original baudrate = 9600!
now baudrate = 115200!

7188xwCmd:b9600original baudrate = 115200!
now baudrate = 9600!

Current baud rate is 9600

{change to Line Mode}
$01M
!017521
:01NoRemotePC
NoRemotePC
```

3.4 Connecting to Multiple Remote RS-232 Device

Step 1: Refer to Sec.3.1 for wiring details and the method used to change the address and default parameters of the I-752N module

Step 2: Connect the second I-752N module to the RS-485 network and the two Remote PCs as follows:



There should now be two I-752N modules connected to the RS-485 network. The module address of the first I-752N module is **address 01**, and the second is **address 02/03/03/04/04/05/08 according to different modules**. The communication status parameters of the two I-752N modules will be same, i.e. **9600**, **N, 8, 1**.

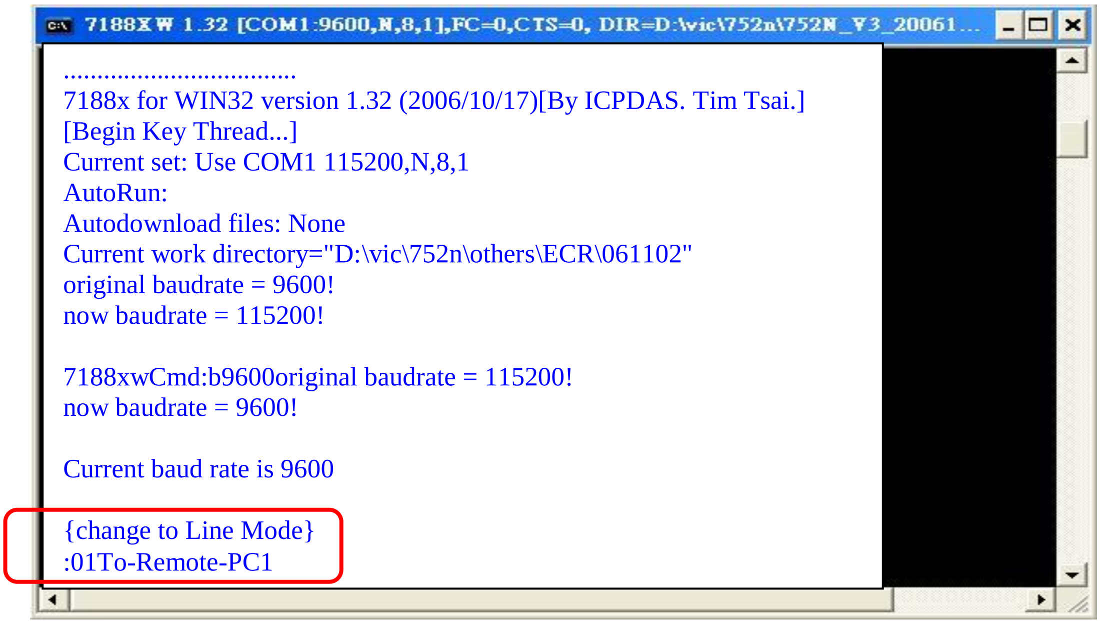
Step 3: Refer to Step 2 to 4 in Sec.3.1 to execute 7188xw.exe on the two Remote PCs. Refer to Steps 5 to 9 in Sec.3.1 for details to see how to change the COM port and status settings to **9600, N, 8, 1**.

Step 4: Send “**To-Remote-PC1**” string from the Host PC to the first Remote PC (#1)

Type **:01To-Remote-PC1**

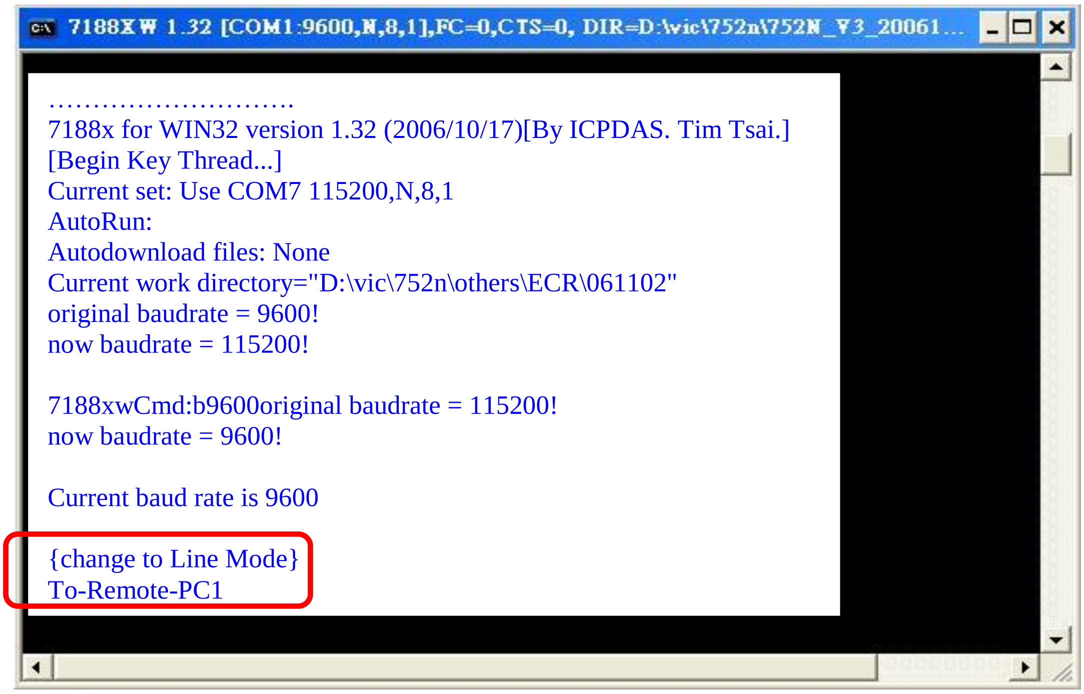
Press **ENTER** to send the command string to the I-752N module

The following screen will be shown on the Host PC:


7188XW 1.32 [COM1:9600,N,8,1],FC=0,CTS=0, DIR=D:\vic\752n\752N_V3_20061...

```
..................................
7188x for WIN32 version 1.32 (2006/10/17)[By ICPDAS. Tim Tsai.]
[Begin Key Thread...]
Current set: Use COM1 115200,N,8,1
AutoRun:
Autodownload files: None
Current work directory="D:\vic\752n\others\ECR\061102"
original baudrate = 9600!
now baudrate = 115200!

7188xwCmd:b9600original baudrate = 115200!
now baudrate = 9600!

Current baud rate is 9600

{change to Line Mode}
:01To-Remote-PC1
```

The following screen will be shown on the first Remote PC (#1):
(Press ALT+L to change to Line Mode)


7188XW 1.32 [COM1:9600,N,8,1],FC=0,CTS=0, DIR=D:\vic\752n\752N_V3_20061...

```
…………………………
7188x for WIN32 version 1.32 (2006/10/17)[By ICPDAS. Tim Tsai.]
[Begin Key Thread...]
Current set: Use COM7 115200,N,8,1
AutoRun:
Autodownload files: None
Current work directory="D:\vic\752n\others\ECR\061102"
original baudrate = 9600!
now baudrate = 115200!

7188xwCmd:b9600original baudrate = 115200!
now baudrate = 9600!

Current baud rate is 9600

{change to Line Mode}
To-Remote-PC1
```

Step 5: Send “**To-Remote-PC2**” string from the Host PC the second Remote PC(#2)

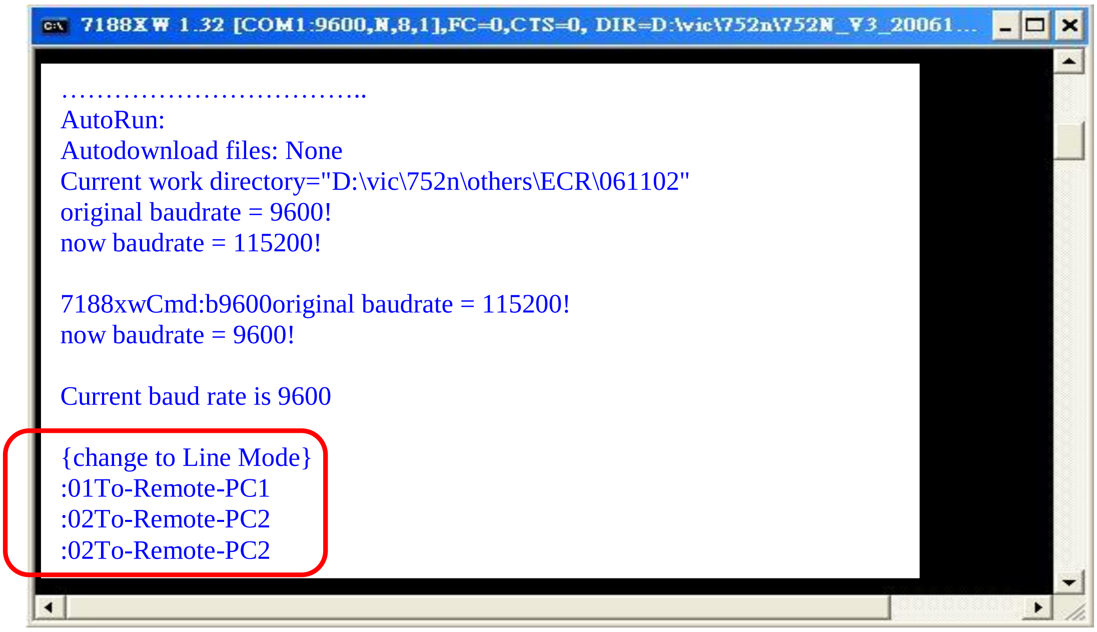
Type **:02To-Remote-PC2**

Press **ENTER** to send the command string to the I-752N module

Type **:02To-Remote-PC2**

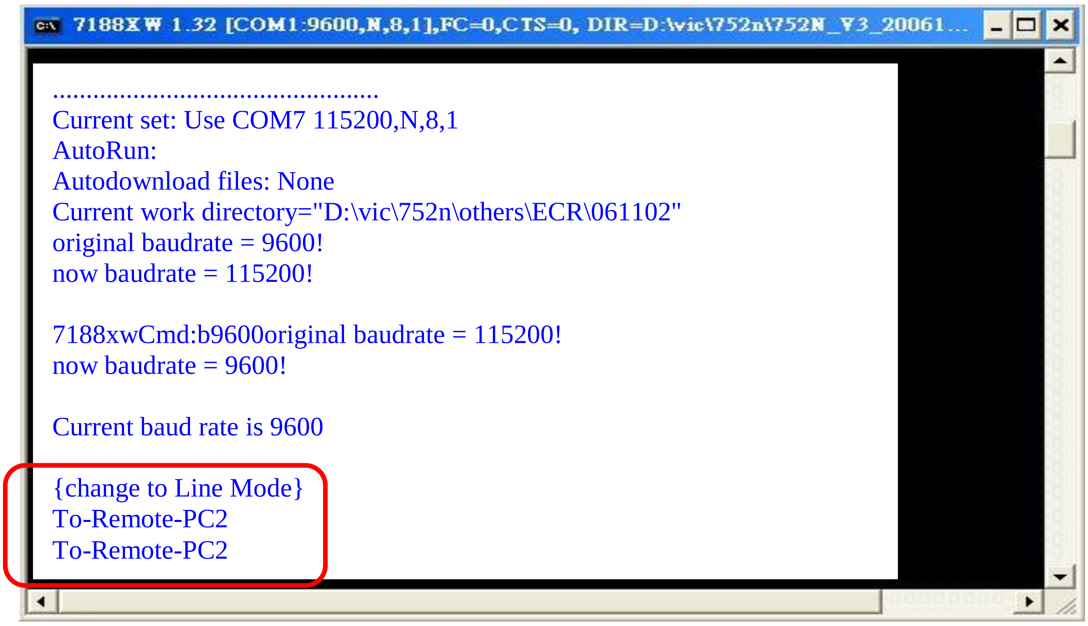
Press **ENTER** to send the command string to the I-752N module

The following screen will be shown on the Host PC:


7188XW 1.32 [COM1:9600,N,8,1],FC=0,CTS=0, DIR=D:\vic\752n\752N_V3_20061...

```
………………………………..
AutoRun:
Autodownload files: None
Current work directory="D:\vic\752n\others\ECR\061102"
original baudrate = 9600!
now baudrate = 115200!

7188xwCmd:b9600original baudrate = 115200!
now baudrate = 9600!

Current baud rate is 9600

{change to Line Mode}
:01To-Remote-PC1
:02To-Remote-PC2
:02To-Remote-PC2
```

The following screen will be shown on the second Remote PC (#2):


7188XW 1.32 [COM1:9600,N,8,1],FC=0,CTS=0, DIR=D:\vic\752n\752N_V3_20061...

```
...............................................
Current set: Use COM7 115200,N,8,1
AutoRun:
Autodownload files: None
Current work directory="D:\vic\752n\others\ECR\061102"
original baudrate = 9600!
now baudrate = 115200!

7188xwCmd:b9600original baudrate = 115200!
now baudrate = 9600!

Current baud rate is 9600

{change to Line Mode}
To-Remote-PC2
To-Remote-PC2
```

3.5 Downloading new Firmware to the I-752N Module

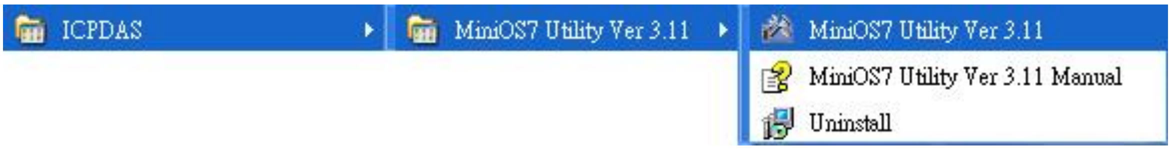
3.5.1. Using MiniOS7 Utility to download new firmware

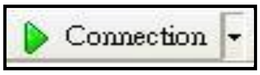
Step 1: Power off I-752N

Step 2: Connect the I-752N module to the Host PC. Refer to Sec. 1.6.1 for details. (Notice: The download port is **COM1,** INIT* and GND pins must be connected.)

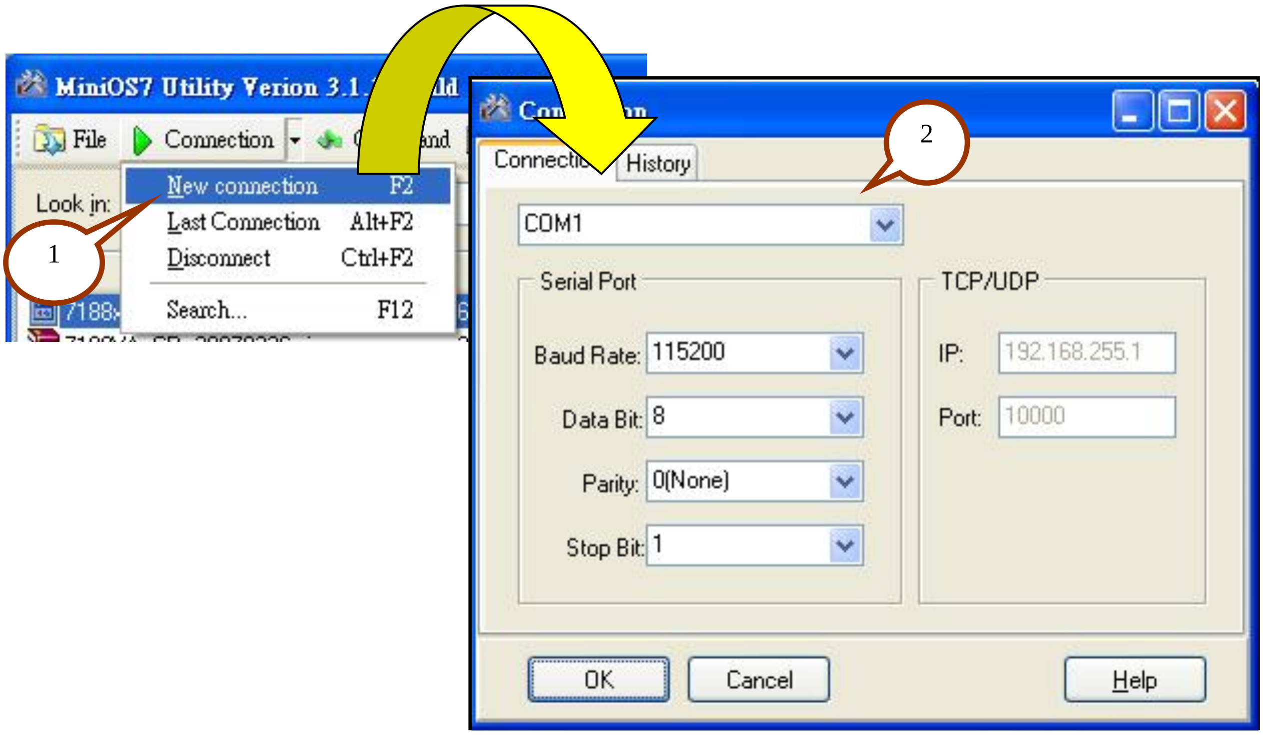
Step 3: Power on I-752N. Install the MiniOS7 Utility.
Locate and execute the minios7_utility_v311.exe file. Installation details can be found in Appendix A MiniOS7 Utility.

Step 4: From the Windows START menu, go to Programs/ICPDAS/MiniOS7 Utility Ver 3.11/and locate the MiniOS7 Utility Ver 3.11.





Step 5: From the menu, select “New connection”. Select the correct COM port and set the other parameters. Click the OK button and the utility will automatically search for the module.



COM1

Serial Port

Baud Rate: 115200

Data Bit: 8

Parity: 0(None)

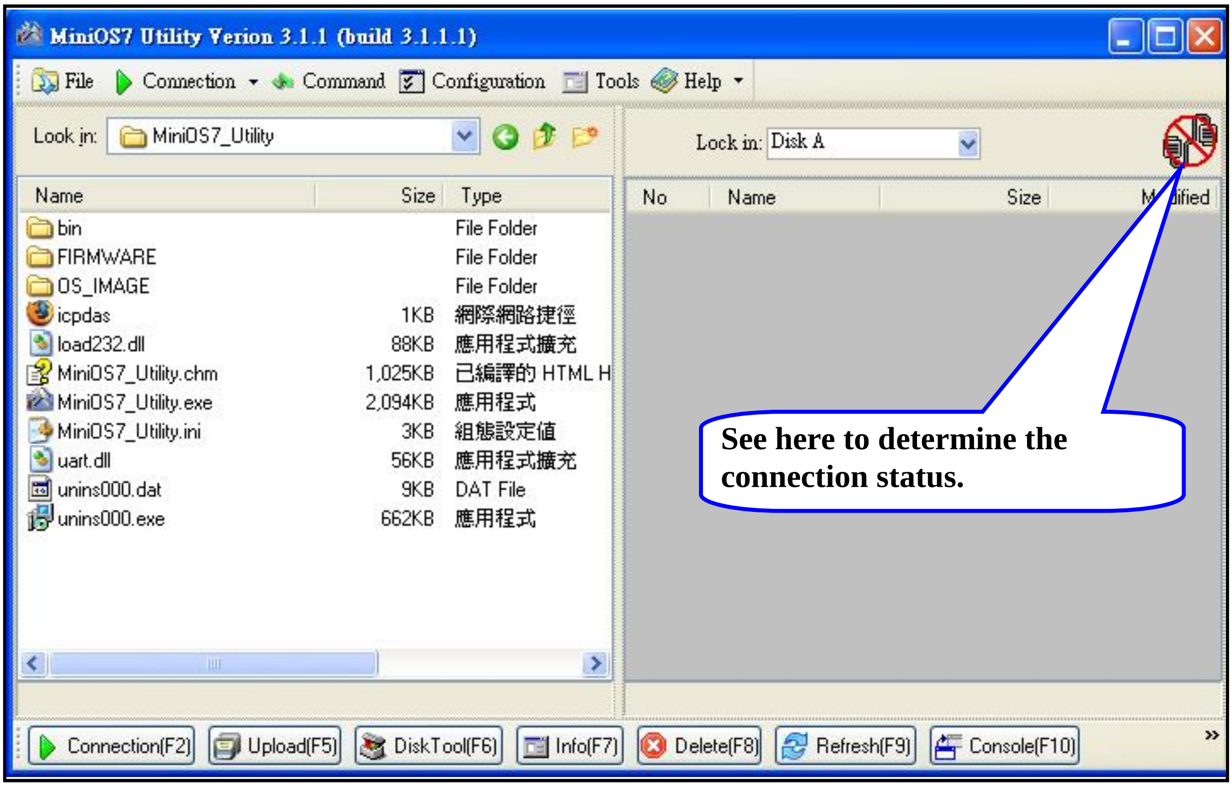
Stop Bit: 1

TCP/UDP

IP: 192.168.255.1

Port: 10000

Step 6: Ensure that the MiniOS7 Utility is connected to the I-752N module. indicates it is connected. indicates that it is disconnected.



Look in: MiniOS7_Utility

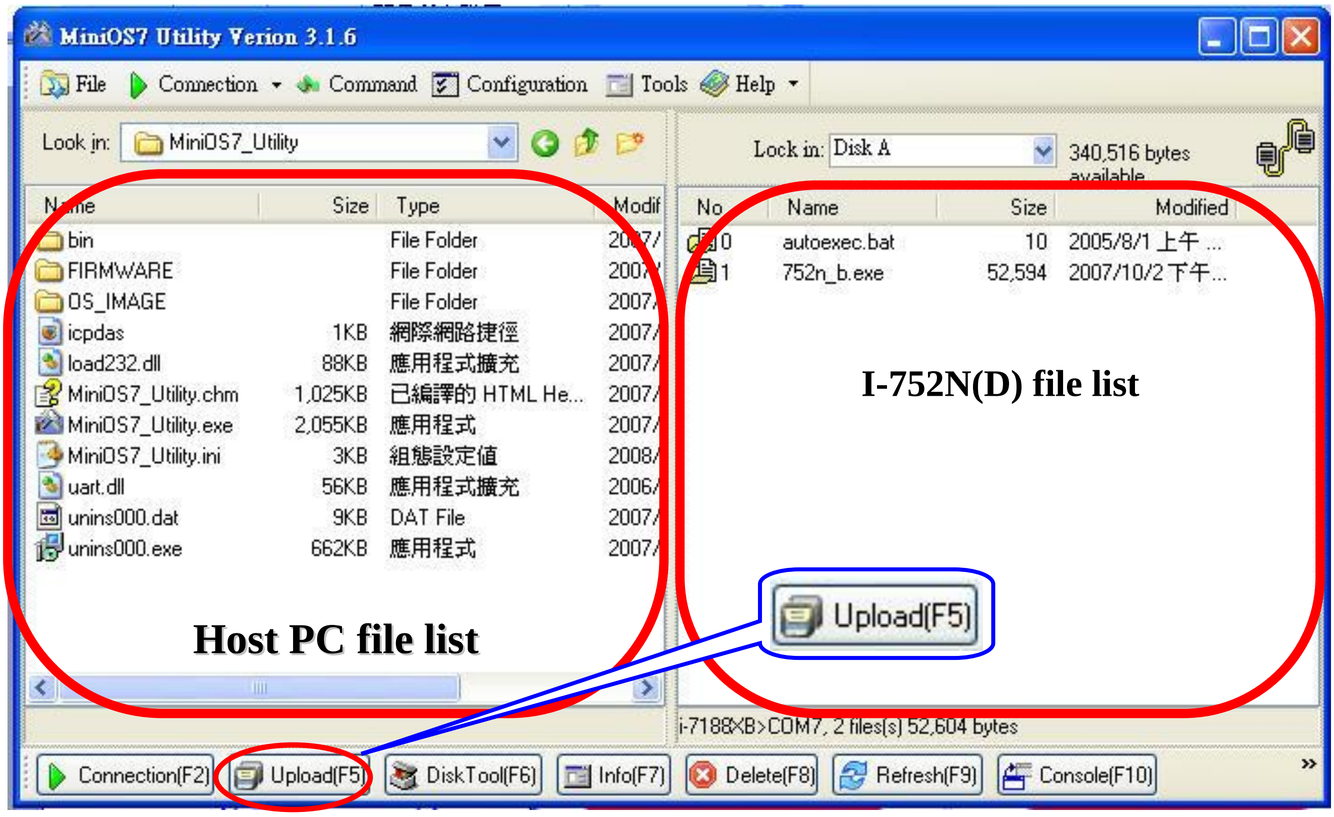
Name	Size	Type
bin		File Folder
FIRMWARE		File Folder
OS_IMAGE		File Folder
icpdas	1KB	網際網路捷徑
load232.dll	88KB	應用程式擴充
MiniOS7_Utility.chm	1,025KB	已編譯的 HTML H
MiniOS7_Utility.exe	2,094KB	應用程式
MiniOS7_Utility.ini	3KB	組態設定值
uart.dll	56KB	應用程式擴充
unins000.dat	9KB	DAT File
unins000.exe	662KB	應用程式

Look in: Disk A

No	Name	Size	Modified

See here to determine the connection status.

Step 7: Delete all files in the I-752N module and then select the file to be loaded from the left hand panel. Drag the file to the right hand panel or click Upload(F5) to load file into module.



Name	Size	Type	Modif
bin		File Folder	2007/
FIRMWARE		File Folder	2007/
OS_IMAGE		File Folder	2007/
icpdas	1KB	網際網路捷徑	2007/
load232.dll	88KB	應用程式擴充	2007/
MiniOS7_Utility.chm	1,025KB	已編譯的 HTML He...	2007/
MiniOS7_Utility.exe	2,055KB	應用程式	2007/
MiniOS7_Utility.ini	3KB	組態設定值	2008/
uart.dll	56KB	應用程式擴充	2006/
unins000.dat	9KB	DAT File	2007/
unins000.exe	662KB	應用程式	2007/

No	Name	Size	Modified
0	autoexec.bat	10	2005/8/1 上午 ...
1	752n_b.exe	52,594	2007/10/2 下午...

Step 8: Power off I-752N and disconnect INIT* and GND pins.

Note: I-7521/I-7522/I-7523 modules need 752N_C.exe and autoexec.bat. I-7522A/I-7524/I-7527 modules need 752N_B.exe and autoexec.bat.

3.5.2. Using 7188xw.exe to download a file

Refer to Appendix B, 7188XW.EXE, for more information.

Note:

The latest firmware version can be obtained from：
http://ftp.icpdas.com.tw/pub/cd/8000cd/napdos/752n/firmware_v3/

Chapter 4 DI/DO and WatchDog

4.1 DO Operation Principles

- The DO of the I-752N module will revert to its initial start values when powering on for the first time.
- If the “$AAZNV” command is received, the DO output will change to desired state. All DO will then remain in the same states until the next “$AAZNV” command is received.
- If the I-752N module is reset by the hardware watchdog, all DO will immediately revert to their power-on values. However, these power-on values may be different from the original states that existed prior to the reset, so the DO states stored on the Host PC may be different from real DO states that were latched in the I-752N module. An “$AAZNV” command must be sent from the Host PC in order to reset these DO to their expected states.
- The “$AA5” command can be used to detect a hardware watchdog reset. Refer to Sec.4.5 for more information. If the I-752N module is reset by the hardware watchdog, an “$AAZNV” command must be sent from the Host PC to reset the DO to their expected states.
- If the host watchdog fails, all DO values will revert to their safe values immediately and the module status will be set to 04. If an “$AAZNV” is sent by the Host PC, the modules will ignore the command and return “!” as a warning. The“~AA1” command can be used to clear the module status to 00, then the I-752N module will again be able to accept an “$AAZNV” command.

4.2 DI Operation Principles

The I-752N series DI commands are as follows:

(1) **#**** synchronized sampling. All modules will sample the DI at the same time

(2) **$AA4** reads the synchronized sampling data.

(3) **$AAYN** reads the current state of the DI

The host computer can only send one command string at a time. If there are two modules, the host computer must send and receive the commands from module1 before sending/receiving any commands from module2, which means that there is a time delay between the two commands. The “synchronize sampling” command is designed for all input modules. When receiving the “#**[0x0D]” synchronized sampling command, all input modules in an RS-485 network will perform input functions at the same time and store these values into the module’s memory. The host computer can then send an “$AA4” read synchronized data command to read the data separately

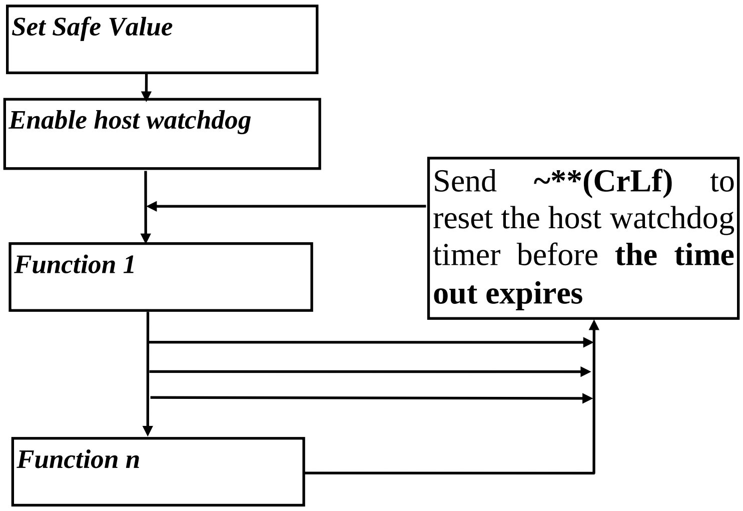
4.3 Dual WatchDog Operation Principles

All I-752N modules are equipped with a hardware module watchdog and a software host watchdog. The I-752N modules are designed for industrial application, therefore they are able to operate in harsh environments. As there are usually a great amount of noise or energy transients in such an environment, the modules may automatically shut down if the noise becomes too large. The built-in hardware module watchdog will reset the module if it shuts down or malfunctions because the noise signal has become too large. Sometimes, even the Host PC may shut down or malfunction because of hardware or software reasons. The software host watchdog can monitor the status of Host PC. If the Host PC fails or is off-line for any reason, the output state of the I-752N modules will revert to its predefined safe states for safety protection purposes.

If the RS-485 network is open, any commands from the Host PC will not be able to be sent to the remote modules. This is very dangerous in real world applications. The I-752N series output module will force the output to revert to its predefined safe state for safety considerations if the host watchdog is enabled. This dual watchdog feature will greatly increase the system reliability.

4.4 Host WatchDog Application Notes

The software host watchdog is designed to monitor the host computer. If the host computer fails, the output of the I-752N modules will automatically revert to its safe state to avoid unnecessary damage. The flow chart for the host computer is as follows:

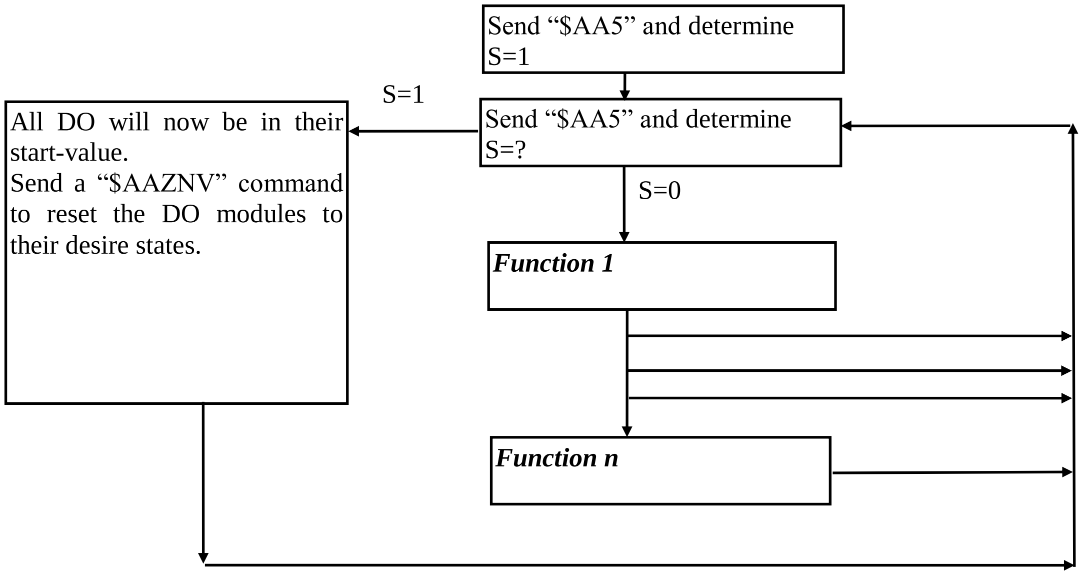


Set Safe Value
Enable host watchdog
Send **~**(CrLf)** to reset the host watchdog timer before **the time out expires**
Function 1
Function n

4.5 Module WatchDog Application Notes

The “$AA5” command is designed to detect any module hardware watchdog failure. If the module was malfunctioned, the module hardware watchdog circuit will reset the module. After reset, the output state of the module will revert to its initial start values. The start value may be different from the output values prior to the module reset. Therefore the user must send the output commands to the module again in order to maintain the same output state both before and after the module watchdog reset.

The flow chart for module hardware watchdog failure detection is as follows:


Send “$AA5” and determine S=1
S=1
All DO will now be in their start-value.
Send a “$AAZNV” command to reset the DO modules to their desire states.
Send “$AA5” and determine S=?
S=0
Function 1
Function n

Chapter 5 Command Sets

5.1 Command Set Table

Command	Response	Description	Reference
$AAA[addr]	!AA	Read/Set the Module Address	Sec. 5.3.1
$AABN[baud rate]	!AA[baud rate]	Read/Set the Baud Rate for COM-1/2/3/4/5/6/7/8	Sec. 5.3.2
$AADN[data-bit]	!AA[data-bit]	Read/Set the Data Bit for COM-1/2/3/4/5/6/7/8	Sec. 5.3.3
$AAPN[parity-bit]	!AA[parity-bit]	Read/Set the Parity Bit for COM-1/2/3/4/5/6/7/8	Sec. 5.3.4
$AAON[stop-bit]	!AA[stop-bit]	Read/Set the Stop Bit for COM-1/2/3/4/5/6/7/8	Sec. 5.3.5
$AA6[ID]	!AA	Set the ID-string for COM-1/3/4/5/6/7/8	Sec. 5.3.6
$AA7	!AA[ID]	Read the ID-string for COM-1/3/4/5/6/7/8	Sec. 5.3.7
$AAC[delimiter]	!AA[delimiter]	Read/Set the delimiter for COM-1/3/4/5/6/7/8	Sec. 5.3.8
$AAD	!AA[delimiter]	Read the delimiter for COM-1/3/4/5/6/7/8	Sec. 5.3.9
(delimiter)AA(bypass)	Depend on device	Bypass the data string to COM-1/3/4/5/6/7/8	Sec. 5.3.10
$AAKV	!AA[checksum]	Read/Set the checksum status of COM2 (RS485)	Sec. 5.3.11
$AATN[CrLfmode]	!AA[CrLfmode]	Read/Set the end char which is used to judge the end of command/response for COM1/2/3/4/5/6/7/8	Sec. 5.3.12
$AAW	!AA(status)	Read the CTS_status of COM-1/3/4/5	Sec. 5.3.13
$AAXV	!AA	Set the RTS_state of COM-1/3/4/5	Sec. 5.3.14
$AAYN	!AA(status)	Read the onboard DI-1/2/3/4/5	Sec. 5.3.15
$AAZNV	!AA(status)	Read/Set the onboard DO-1/2/3/4/5	Sec. 5.3.16
#**	No Response	Synchronized Sampling	Sec. 5.3.17
$AA4	!AASV	Read the synchronized data	Sec. 5.3.18
$AA5	!AAS	Read the Reset status	Sec. 5.3.19
$AAF	!AA[number]	Read the firmware version number	Sec. 5.3.20
$AAM	!AA[name]	Read the module name	Sec. 5.3.21
$AA2	!AABDPK	Read the configuration of COM2 (RS485)	Sec. 5.3.22
~**	No Response	Host is OK	Sec. 5.3.23
~AA0	!AASS	Read the module status	Sec. 5.3.24
~AA1	!AA	Reset the module status	Sec. 5.3.25
~AA2	!AASTT	Read the host watchdog status and value	Sec. 5.3.26
~AA3ETT	!AASTT	Enable the host watchdog timer	Sec. 5.3.27
~AA4P/~AA4S	!AAV	Read the power-on/safe value	Sec. 5.3.28
~AA5P/~AA5S	!AAV	Set the power-on/safe value	Sec. 5.3.29

$AAU	[data]	Read data from the RS-232 COM port buffer.	Sec. 5.3.30
$AAL[data]	!AA	Write to the expansion board DO-0/1/2/3	Sec. 5.3.31
$AAR	!AA[data]	Read the expansion board DI-0/1/2/3	Sec. 5.3.32
$AAJN[timeout]	!AA[timeout]	Read/Set the delay time to determine whether the end of a command/response was been transmitted /receiver	Sec. 5.3.33
$AAGN[trigger level]	!AA[triggerlevel]	Read/Set the Trigger Level	Sec. 5.3.34
@AA[data]	>AA[data]	Read/Set the onboard DI/DO-1/2/3/4/5	Sec. 5.3.35
#AABBHH	>	Set the multiple onboard DO-1/2/3/4/5	Sec. 5.3.36
#AABCDD	>	Set the single onboard DO-1/2/3/4/5	Sec. 5.3.37
$AAEV	!AA(status)	Read/Set prefixed address status on the response. (Supported by the firmware version 3.05 and upward)	Sec. 5.3.38
$AAHV	!AA(status)	Read/Set the status for bypassing the data string to COM2.(Supported by the firmware version 3.08 and upward)	Sec. 5.3.39
$AAIV	!AA	Restore factory default setting. (Supported by the firmware version 3.08 and upward)	Sec. 5.3.40

5.2 Address Mapping Table

	I-7521	I-7522	I-7523	I-7522A	I-7524	I-7527
COM1 (RS-232)	AA	AA	AA	AA	AA	AA
COM2 (RS-485)	AA	AA	AA	AA	AA	AA
COM3 (RS-232) /(RS-422)	N/A	AA+1	AA+1	AA+1 (RS-422)	AA+1	AA+1
COM4 (RS-232)	N/A	N/A	AA+2	N/A	AA+2	AA+2
COM5 (RS-232)	N/A	N/A	N/A	N/A	AA+3	AA+3
COM6 (RS-232)	N/A	N/A	N/A	N/A	N/A	AA+4
COM7 (RS-232)	N/A	N/A	N/A	N/A	N/A	AA+5
COM8 (RS-232)	N/A	N/A	N/A	N/A	N/A	AA+6

5.3 Commands

5.3.1 $AAA[addr]

For: I-7521/I-7522/I-7522A/I-7523/I-7524/I-7527

- **Description**:
  This function reads/sets the module address
  $AAA[chk](CrLf) → reads the module address stored in the EEPROM
  $AAA[addr][chk](CrLf) → sets the module address

- **Syntax**:
  $AAA[addr][chk](CrLf)

$	Delimiter character
AA	2-character module address in HEX format. The valid range is from 00 to FF
[chk]	2-character checksum. If the checksum is disabled → no [chk]
(CrLf)	End Character

- **Response**: valid command → !AA[chk](CrLf)
  invalid command → ?AA[chk](CrLf)
  no response → syntax error, communication error, or address error

!	Delimiter character indicating a valid command
?	Delimiter character indicating an invalid command
AA	2-character module address in HEX format
[chk]	2-character checksum. If the checksum is disabled → no [chk]
(CrLf)	End Character

- **Example**:

```
command:  $01A02(CrLf)
response: !01(CrLf)
command:  $02AA0(CrLf)
response: !02(CrLf)
command:  $00A(CrLf)
response: !02(CrLf)
```

address 01 is changed to 02

address 02 is changed to 0xA0

address stored in EEPROM=02
(see NOTE 2)

- **Notes:**
  (1) The AA address value will be displayed on LED1 and LED2. Refer to Sec.3.1 for more information.
  (2) Connect the DI/INIT* pin to GND pin and use the $00A command to read the address stored in the EEPROM.
  (Refer to Sec. 2.1 INIT* Pin Operating Principles.)

5.3.2 $AABN[baud rate]

For: I-7521/I-7522/I-7522A/I-7523/I-7524/I-7527

- **Description**:
  This function reads/sets the Baud Rate for COM 1/2/3/4/5/6/7/8
  $AABN[chk](CrLf) → Reads the Baud Rate for COM 1/2/3/4/5/6/7/8 stored in the EEPROM
  $AABN[baud rate][chk](CrLf) → Sets the Baud rate for COM 1/2/3/45/6/7/8

- **Syntax**:
  $AABN[baud rate][chk](CrLf)

$	Delimiter character
AA	2-character module address in HEX format. The valid range is from 00 to FF
N=0	Read/Set the Baud Rate for the COM 2
N=1	Read/Set the Baud Rate for the COM 1/3/4/5/6/7/8
[baud rate]	Valid values are 300/600/1200/2400/4800/9600/19200/38400/ 57600/115200
[chk]	2-character checksum. If the checksum is disabled → no [chk]
(CrLf)	End Character

- **Response**: valid command → !AA[baud rate][chk](CrLf)
  invalid command → ?AA[chk](CrLf)
  no response → syntax error, communication error, or address error

!	Delimiter character indicating a valid command
?	Delimiter character indicating an invalid command
AA	2-character module address in HEX format
[chk]	2-character checksum. If the checksum is disabled → no [chk]
(CrLf)	End Character

- **Example**: (Assume the AA address value of the I-7523 module is 01)

command: $01B0115200(CrLf)
response: !01(CrLf)

Changes the RS-485 (COM2) Baud Rate to 115200 bps

command: $01B19600(CrLf)
response: !01(CrLf)

Changes the RS-232 (COM1) Baud Rate to 9600 bps

command: $02B138400(CrLf)
response: !02(CrLf)

Changes the RS-232 (COM3) Baud Rate to 38400 bps

command: $03B1(CrLf)
response: !0357600(CrLf)

Reads the RS-232 (COM4) Baud Rate.

- **Notes:**
  (1) Address mapping refers to Sec.5.2
  (2) Short code for the baud rates:
  300=1, 600=2, 1200=3, 2400=4, 4800=5, 9600=6, 19200=7, 38400=8, 57600=9,115200=A. the short code for the Baud Rate will be shown on LED3 of the 7-segment.
  Refer to Sec. 3.1 for more information.

5.3.3 $AADN[data-bit]

For: I-7521/I-7522/I-7522A/I-7523/I-7524/I-7527

- **Description**:
  This function reads/sets the data bit for COM 1/2/3/4/5/6/7/8
  $AADN[chk](CrLf) → Reads the data bit for COM 1/2/3/4/5/6/7/8 stored in the EEPROM
  $AADN[data-bit][chk](CrLf) → Sets the data bit for COM 1/2/3/4/5/6/7/8

- **Syntax**:
  $AADN[data-bit][chk](CrLf)
  $ Delimiter character
  AA 2-character module address in HEX format. The valid range is from 00 to FF
  N=0 Reads/Sets the data bit for the COM 2
  N=1 Reads/Sets the data bit for the COM 1/3/4/5/6/7/8
  [data-bit] will be either 7 or 8
  [chk] 2-character checksum. If the checksum is disabled → no [chk]
  (CrLf) End Character

- **Response**: valid command → !AA[data-bit][chk](CrLf)
  invalid command → ?AA[chk](CrLf)
  no response → syntax error, communication error, or address error
  ! Delimiter character indicating a valid command
  ? Delimiter character indicating an invalid command
  AA 2-character module address in HEX format
  [chk] 2-character checksum. If the checksum is disabled → no [chk]
  (CrLf) End Character

- **Example**: (Assume the AA address value of the I-7523 module is 01)
  command: $01D08(CrLf)
  response: !01(CrLf)

  Changes the data bit to 8 for the RS-485 (COM2)

  command: $01D17(CrLf)
  response: !01(CrLf)

  Changes the data bit to 7 for the RS-232 (COM1)

  command: $02D17(CrLf)
  response: !02(CrLf)

  Changes the data bit to 7 for the RS-232 (COM3)

  command: $03D17(CrLf)
  response: !03(CrLf)

  Changes the data bit to 7 for the RS-232 (COM4)

- **Notes:**
  (1) Address mapping refers to Sec.5.2

(2) Valid data bit values:

	I-7521	I-7522	I-7522A	I-7523	I-7524	I-7527
COM1	7/8	7/8	7/8	7/8	7/8	7/8
COM2	7/8	7/8	7/8	7/8	7/8	7/8
COM3	N/A	7/8	7/8	7/8	7/8	7/8
COM4	N/A	N/A	N/A	7/8	7/8	7/8
COM5	N/A	N/A	N/A	N/A	7/8	7/8
COM6	N/A	N/A	N/A	N/A	N/A	7/8
COM7	N/A	N/A	N/A	N/A	N/A	7/8
COM8	N/A	N/A	N/A	N/A	N/A	7/8

5.3.4 $AAPN[parity-bit]

For: I-7521/I-7522/I-7522A/I-7523/I-7524/I-7527

- **Description**:
  This function reads/sets the parity bit for COM 1/2/3/4/5/6/7/8
  $AAPN[chk](CrLf) → Reads the parity bit for COM 1/2/3/4/5/6/7/8 stored in the EEPROM
  $AAPN[parity-bit][chk](CrLf) → Sets the parity bit for COM 1/2/3/4/5/6/7/8

- **Syntax**:
  $AAPN[parity-bit][chk](CrLf)
  $ Delimiter character
  AA 2-character module address in HEX format. The valid range is from 00 to FF
  N=0 Reads/Sets the parity bit for the COM 2
  N=1 Reads/Sets the parity bit for the COM 1/3/4/5/6/7/8
  [parity-bit] 0=NONE, 1=EVEN, 2=ODD
  [chk] 2-character checksum. If the checksum is disabled → no [chk]
  (CrLf) End Character

- **Response**: valid command → !AA[parity-bit][chk](CrLf)
  invalid command → ?AA[chk](CrLf)
  no response → syntax error, communication error, or address error
  ! Delimiter character indicating a valid command
  ? Delimiter character indicating an invalid command
  AA 2-character module address in HEX format
  [chk] 2-character checksum. If the checksum is disabled → no [chk]
  (CrLf) End Character

- **Example**: (Assume the AA address value of the I-7523 module is 01)
  command: $01P00(CrLf)
  response : !01(CrLf)
  Changes parity-bit to NONE for RS-485 (COM2)
  command: $01P10(CrLf)
  response : !01(CrLf)
  Changes parity-bit to NONE for RS-232 (COM1)
  command: $02P11(CrLf)
  response : !02(CrLf)
  Changes parity-bit to EVEN for RS-232 (COM3)
  command: $03P12(CrLf)
  response : !03(CrLf)
  Changes parity-bit to ODD for RS-232 (COM4)

- **Notes:**
  (1) Address mapping refers to Sec.5.2
  (2) Valid parity bit values:

	I-7521	I-7522	I-7522A	I-7523	I-7524	I-7527
COM1 (RS-232)	N/E/O	N/E/O	N/E/O	N/E/O	N/E/O	N/E/O
COM2 (RS-485)	N/E/O	N/E/O	N/E/O	N/E/O	N/E/O	N/E/O
COM3 (RS-232)*	N/A	N/E/O	N/E/O	N/E/O	N/E/O	N/E/O
COM4 (RS-232)	N/A	N/A	N/A	N/E/O	N/E/O	N/E/O
COM5 (RS-232)	N/A	N/A	N/A	N/A	N/E/O	N/E/O
COM6 (RS-232)	N/A	N/A	N/A	N/A	N/A	N/E/O
COM7 (RS-232)	N/A	N/A	N/A	N/A	N/A	N/E/O
COM8 (RS-232)	N/A	N/A	N/A	N/A	N/A	N/E/O

N: None E: Even O: Old

***COM3 of the I-7522A module is RS-422/485**

5.3.5 $AAON[stop-bit]

For: I-7521/I-7522/I-7522A/I-7523/I-7524/I-7527

- **Description**:

  This function reads/sets the stop bit for COM 3/4/5/6/7/8

  $AAON[chk](CrLf) → Reads the stop bit of COM 3/4/5/6/7/8 stored in the EEPROM

  $AAON[stop-bit][chk](CrLf) → Sets the stop bit for COM 3/4/5/6/7/8

- **Syntax**:

  $AAON[stop-bit][chk](CrLf)

$	Delimiter character
AA	2-character module address in HEX format. The valid range is from 00 to FF
N=0	Reads the stop bit for the COM 2
N=1	Reads/Sets the stop bit for the COM 1/3/4/5/6/7/8
[stop-bit]	1/2 for COM1/3/4/5/6/7/8
[chk]	2-character checksum. If the checksum is disabled → no [chk]
(CrLf)	End Character

- **Response**: valid command → !AA[stop-bit][chk](CrLf)

  invalid command → ?AA[chk](CrLf)

  no response → syntax error, communication error, or address error

!	Delimiter character indicating a valid command
?	Delimiter character indicating ad invalid command
AA	2-character module address in HEX format
[chk]	2-character checksum. If the checksum is disabled → no [chk]
(CrLf)	End Character

- **Example**: (Assume the AA address value of the I-7523 module is 01)

command:	$02O12(CrLf)	Changes the stop bit to 2 for the RS-232 (COM3)
response:	!02(CrLf)	
command:	$03O12(CrLf)	Changes the stop bit to 2 of the RS-232 (COM4)
response:	!03(CrLf)	

- **Notes:**

  (1) Address mapping refers to Sec.5.2

(2) Valid stop bit values:

	I-7521	I-7522	I-7522A	I-7523	I-7524	I-7527
COM1 (RS-232)	1 or 2	1 or 2	1 or 2	1 or 2	1 or 2	1 or 2
COM2 (RS-232)	1	1	1	1	1	1
COM3 (RS-232)	N/A	1 or 2	1 or 2	1 or 2	1 or 2	1 or 2
COM4 (RS-232)	N/A	N/A	N/A	1 or 2	1 or 2	1 or 2
COM5 (RS-232)	N/A	N/A	N/A	N/A	1 or 2	1 or 2
COM6 (RS-232)	N/A	N/A	N/A	N/A	N/A	1 or 2
COM7 (RS-232)	N/A	N/A	N/A	N/A	N/A	1 or 2
COM8 (RS-232)	N/A	N/A	N/A	N/A	N/A	1 or 2

(3) COM3 of the I-7522A module is RS-422/485

(4) The stop bit for COM1 & COM2 is always 1 when data bit is 8.

(5) COM (1/3/4/5/6/7/8) of the I-7521/I-7522/7523/7523A/7524/7527 module can be linked to the HP34401A

(6) The stop bit can be set to 2 as the data bit is 7 for COM 1, otherwise the stop bit can only be set to 1 as the data bit is 8 for COM 1.

5.3.6 $AA6[ID]

For: I-7521/I-7522/I-7522A/I-7523/I-7524/I-7527

- **Description**:
  This function sets the ID string for COM 1/3/4/5/6/7/8.
  Max-number of characters =50.

- **Syntax**:
  $AA6[ID][chk](CrLf)

$	Delimiter character
AA	2-character module address in the format. The valid range is from 00 to FF
[ID]	ID string, Max-number of character =50
[chk]	2-character checksum. If the checksum is disabled → no [chk]
(CrLf)	End Character

- **Response**: valid command → !AA[chk](CrLf)
  invalid command → ?AA[chk](CrLf)
  no response → syntax error, communication error, or address error

!	Delimiter character indicating a valid command
?	Delimiter character indicating an invalid command
AA	2-character module address in HEX format
[chk]	2-character checksum. If the checksum is disabled → no [chk]
(CrLf)	End Character

- **Example**: (Assume the AA address value of the I-7523 module is 01)

command: $016Temperature1(CrLf)
response: !01(CrLf)

The ID for the RS-232 (COM1) is Temperature1

command: $026HP34401A-1(CrLf)
response: !02(CrLf)

The ID for the RS-232 (COM3) is HP34401A-1

command: $036HP34401A-2(CrLf)
response: !03(CrLf)

The ID for the RS-232 (COM4) is HP34401A-2

- **Note:**
  Address mapping refers to Sec.5.2

5.3.7 $AA7

For: I-7521/I-7522/I-7522A/I-7523/I-7524/I-7527

- **Description**:
  This function reads the ID string for COM 1/3/4/5/6/7/8

- **Syntax**:
  $AA7[chk](CrLf)
  $ Delimiter character
  AA 2-character module address in HEX format. The valid range is from 00 to FF
  [chk] 2-character checksum. If the checksum is disabled → no [chk]
  (CrLf) End Character

- **Response**: valid command → !AA(ID)[chk](CrLf)
  invalid command → ?AA[chk](CrLf)
  no response → syntax error, communication error ,or address error
  ! Delimiter character indicating a valid command
  ? Delimiter character indicating an invalid command
  AA 2-character module address in HEX format
  (ID) ID string. Maximum number of characters=50
  [chk] 2-character checksum. If the checksum is disabled → no [chk]
  (CrLf) End Character

- **Example**: (Assume the AA address value of the I-7523 module is 01)
  command: $017(CrLf)
  response: !01Temperature1(CrLf)
  The ID for the RS-232 (COM1) is Temperature1
  command: $027(CrLf)
  response: !02HP34401A-1(CrLf)
  The ID for the RS-232 (COM3) is HP34401A-1
  command: $037(CrLf)
  response: !03HP34401A-2(CrLf)
  The ID for the RS-232 (COM4) is HP34401A-2

- **Note:**
  Address mapping refers to Sec.5.2

5.3.8 $AAC[delimiter]

For: I-7521/I-7522/I-7522A/I-7523/I-7524/I-7527

- **Description**:

  This reads/sets the delimiter for COM 1/3/4/5/6/7/8

  $AAC[chk](CrLf) → Reads the delimiter for COM 1/3/4/5/6/7/8stored in the EEPROM

  $AAC[delimiter][chk](CrLf) → Sets the delimiter for COM1/3/4/5/6/7/8

- **Syntax**:

  $AAC[delimiter][chk](CrLf)

$	Delimiter character
AA	2-character module address in HEX format. The valid range is from 00 to FF
[delimiter]	Default delimiter is **:**
[chk]	2-character checksum. If the checksum is disabled → no [chk]
(CrLf)	End Character

- **Response**: valid command → !AA[delimiter][chk](CrLf)

  invalid command → ?AA[chk](CrLf)

  no response → syntax error, communication error ,or address error

!	Delimiter character indicating a valid command
?	Delimiter character indicating an invalid command
AA	2-character module address in HEX format
[chk]	2-character checksum. If the checksum is disabled → no [chk]
(CrLf)	End Character

- **Example**: (Assume the AA address value of the I-7523 module is 01)

command:	$01C(CrLf)	Reads the delimiter for the RS-232 (COM1) → **:**
response:	!01:(CrLf)	
command:	$02C(CrLf)	Reads the delimiter for the RS-232 (COM3) → **:**
response:	!02:(CrLf)	
command:	$03C*(CrLf)	Changes the delimiter for the RS-232 (COM4) → *****
response:	!03(CrLf)	

- **Notes:**

  (1) Address mapping refers to Sec.5.2

  (3) The delimiter of COM1/3/4/5/6/7/8 can be different.

  (4) The default delimiter is → :

  (5) The delimiter cannot be $, ~, #, @, %, CR & LF

5.3.9 $AAD

For: I-7521/I-7522/I-7522A/I-7523/I-7524/I-7527

- **Description**:
  This function reads the delimiter for COM 1/3/4/5/6/7/8

- **Syntax**:
  $AAD[chk](CrLf)
  $ Delimiter character
  AA 2-character module address in HEX format. The valid range is from 00 to FF
  [chk] 2-character checksum. If the checksum is disabled → no [chk]
  (CrLf) End Character

- **Response**: valid command → !AA(delimiter)[chk](CrLf)
  invalid command → ?AA[chk](CrLf)
  no response → syntax error, communication error, or address error
  ! Delimiter character indicating a valid command
  ? Delimiter character indicating an invalid command
  AA 2-character module address in HEX format
  (delimiter) the default delimiter is → **:**
  [chk] 2-character checksum. If the checksum is disabled → no [chk]
  (CrLf) End Character

- **Example**: (Assume the AA address value of the I-7523 module is 01)
  command: $01D(CrLf)
  response: !01:(CrLf)

  Reads the delimiter for the RS-232 (COM1) → **:**

  command: $02D(CrLf)
  response: !02:(CrLf)

  Reads the delimiter for the RS-232 (COM3) → **:**

  command: $03D(CrLf)
  response: !03*(CrLf)

  Reads the delimiter for the RS-232 (COM4) → *****

- **Note:**
  Address mapping refers to Sec.5.2

5.3.10 [delimiter]AA[bypass]

For: I-7521/I-7522/I-7522A/I-7523/I-7524/I-7527

- **Description**:
  This function bypasses the data string to COM 1/3/4/5/6/7/8

- **Syntax**:
  (delimiter)AA(pass)[chk](CrLf)
  (delimiter) Refer to Section 5.3.8
  AA 2-character module address in HEX format. The valid range is from 00 to FF
  (bypass) The data string sent to COM 1/3/4/5/6/7/8
  [chk] 2-character checksum. If the checksum is disabled → no [chk]
  (CrLf) End Character

- **Response**: The response received will depend on the device used

- **Example**: (Assume the AA address value of the I-7523 module is 01. The delimiters for **COM1/3/4** are **: , ; ,** and *****, **respectively**)

  command: :01abcde(CrLf)
  response: Depends on the device

  Send **abcde** to COM1

  command: ;02123456789(CrLf)
  response: Depends on the device

  Send **123456789** to COM3

  command: *03test(CrLf)
  response: Depends on the device

  Send **test** to COM4

- **Note:**
  Address mapping refers to Sec.5.2

5.3.11 $AAKV

For: I-7521/I-7522/I-7522A/I-7523/I-7524/I-7527

- **Description:**
  This function reads/sets the checksum status
  $AAK[chk](CrLf) → Reads the checksum status stored in the EEPROM
  $AAKV[chk](CrLf) → Sets the checksum status

- **Syntax**:
  $AAK[V][chk](CrLf)

$	Delimiter character
AA	2-character module address in HEX format. The valid range is from 00 to FF
V=0	checksum disabled
V=1	checksum enabled
[chk]	2-character checksum. If the checksum is disabled → no [chk]
(CrLf)	End Character

- **Response**: valid command → !AA[V][chk](CrLf)
  invalid command → ?AA[chk](CrLf)
  no response: → syntax error, communication error, or address error

!	Delimiter character indicating a valid command
?	Delimiter character indicating an invalid command
AA	2-character module address in HEX format
V=0	checksum disabled
V=1	checksum enabled
[chk]	2-character checksum. If the checksum is disabled → no [chk]
(CrLf)	End Character

- **Example**: (Assume the first AA address value of the I-7523 module is 01, the other is 04)

```
command:  $01K000(CrLf)
response: !0182(CrLf)
command:  $04K1(CrLf)
response: !04(CrLf)
```

Checksum=1. Disables the checksum
chk: 00,82

The checksum is enabled

- **Notes:**
  (1) Address mapping refers to Sec.5.2
  (2) The checksum enable/disable function is valid for COM2 only.

5.3.12 $AATN[CrLfmode]

For: I-7521/I-7522/I-7522A/I-7523/I-7524/I-7527

- **Description**:
  This function reads/sets what the characters are as judging the end of command or response string.
  $AATN[chk](CrLf) → Reads the setting value of CrLfmode stored in the EEPROM.
  $AATN(CrLfmode)[chk](CrLf) → Sets the setting value of CrLfmode for the command/response string.

- **Syntax**:
  $AATN[CrLfmode][chk](CrLf)

$	Delimiter character
AA	2-character module address in HEX format. The valid range is from 00 to FF
N=0	Reads/Sets the CrLfmode value of the COM 2
N=1	Reads/Sets the CrLfmode value of the COM 1/3/4/5/6/7/8
(CrLfmode):	0 → (CrLf)=0x0D (CR) 1 → (CrLf)=0x0D+0x0A (CR+LF) 2 → (CrLf)=0x0A (LF) 3 → (CrLf)=0x0A+0x0D (LF+CR) 4 → No end character
[chk]	2-character checksum. If the checksum is disabled → no [chk]
(CrLf)	End Character

- **Response**: valid command → !AA[CrLfmode][chk](CrLf)
  invalid command → ?AA[chk](CrLf)
  no response → syntax error, communication error, or address error

!	Delimiter character indicating a valid command
?	Delimiter character indicating an invalid command
AA	2-character module address in HEX format
[chk]	2-character checksum. If the checksum is disabled → no [chk]
(CrLf)	End Character

- **Example**: (Assume the AA address value of the I-7523 module is 01)

command:	$01T0(CrLf)	The end char for COM2 is no end character
response:	!014(CrLf)	
command:	$01T1(CrLf)	The end char for COM1 is 0x0D+0x0A
response:	!011(CrLf)	
command:	$02T1(CrLf)	The end char for COM3 is 0x0A
response:	!022(CrLf)	

command: $03T1(CrLf)
response: !033(CrLf)

The end char for COM4 is 0x0A+0x0D

- **Notes:**
  (1) Address mapping refers to Sec.5.2.
  (2) The default CrLfmode = 4 → ie.the default (CrLf)=NONE for all port.

5.3.13 $AAW

For: I-7521/I-7522/I-7522A/I-7523/I-7524/I-7527

- **Description**:
  This function reads the CTS status for COM 1/3/4/5

- **Syntax**:
  $AAW[chk](CrLf)

$	Delimiter character
AA	2-character module address in HEX format. The valid range is from 00 to FF
[chk]	2-character checksum. If the checksum is disabled → no [chk]
(CrLf)	End Character

- **Response**: valid command → !AAS[chk](CrLf)
  invalid command → ?AA[chk](CrLf)
  no response → syntax error, communication error, or address error

S=0	CTS is inactive
S=1	CTS is active HIGH
!	Delimiter character indicating a valid command
?	Delimiter character indicating an invalid command
AA	2-character module address in HEX format
[chk]	2-character checksum. If the checksum is disabled → no [chk]
(CrLf)	End Character

- **Example**: (Assume the AA address value of the I-7523 module is 01)

```
command:  $01W(CrLf)
response: !010(CrLf)
command:  $02W(CrLf)
response: !021(CrLf)
```

The CTS status for COM1 is inactive now.

The CTS status for COM3 is active-HIGH now.

Address mapping for CTS status:

	Com1 (RS-232)	Com3 (RS-232)	Com4 (RS-232)	Com5 (RS-232)
I-7521	AA	N/A	N/A	N/A
I-7522	AA	AA+1	N/A	N/A
I-7522A*	AA	N/A	N/A	N/A
I-7523	AA	AA+1	N/A	N/A
I-7524	AA	AA+1	AA+2	AA+3
I-7527	AA	N/A	N/A	N/A

- **Note:**
  (1) The CTS status is valid for COM1, COM3, COM4 and COM5
  (2) *COM3 of the I-7522A module is RS-422/485

5.3.14 $AAXV

For: I-7521/I-7522/I-7522A/I-7523/I-7524/I-7527

- **Description**:
  This function sets the RTS state for COM 1/3/4/5

- **Syntax**:
  $AAXV[chk](CrLf)

$	Delimiter character
AA	2-character module address in HEX format. The valid range is from 00 to FF
V=0	Sets the RTS state to inactive
V=1	Sets the RTS state to active_HIGH
[chk]	2-character checksum. If the checksum is disabled → no [chk]
(CrLf)	End Character

- **Response**: valid command → !AA[chk](CrLf)
  invalid command → ?AA[chk](CrLf)
  no response → syntax error, communication error ,or address error

!	Delimiter character indicating a valid command
?	Delimiter character indicating an invalid command
AA	2-character module address in HEX format
[chk]	2-character checksum. If the checksum is disabled → no [chk]
(CrLf)	End Character

- **Example**: (Assume the AA address value of the I-7523 module is 01)
  command: $01X0(CrLf)
  response: !01(CrLf)

  Sets the RTS status of COM1 to inactive

  command: $02X1(CrLf)
  response: !02(CrLf)

  Sets the RTS status of COM3 to active-HIGH

Address mapping for RTS status

	Com1 (RS-232)	Com3 (RS-232)	Com4 (RS-232)	Com5 (RS-232)
I-7521	AA	N/A	N/A	N/A
I-7522	AA	AA+1	N/A	N/A
I-7522A*	AA	N/A	N/A	N/A
I-7523	AA	AA+1	N/A	N/A
I-7524	AA	AA+1	AA+2	AA+3
I-7527	AA	N/A	N/A	N/A

- **Note:**
  (1)The RTS status is valid for COM1, COM3, COM4 and COM5
  (2) *COM3 of the I-7522A module is RS-422/485

5.3.15 $AAYN

For: I-7521/I-7522/I-7522A/I-7523/I-7524/I-7527

- **Description**:
  This function reads the onboard DI 1/2/3/4/5

- **Syntax**:
  $AAYN[chk](CrLf)
  $ Delimiter character
  AA 2-character module address in HEX format. The valid range is from 00 to FF
  N 1 → Reads DI1, 2 → Reads DI2, 3→ Reads DI3
  4 → Reads DI4, 5 → Reads DI5 (I-7522A refer to Sec.1.4)
  [chk] 2-character checksum. If the checksum is disabled → no [chk]
  (CrLf) End Character

- **Response**: valid command → !AAS[chk](CrLf)
  invalid command → ?AA[chk](CrLf)
  no response → syntax error, communication error, or address error
  ! Delimiter character indicating a valid command
  ? Delimiter character indicating an invalid command
  AA 2-character module address in HEX format
  S=0 DI=Low,
  S=1 DI=High (DI floating will return High)
  [chk] 2-character checksum. If the checksum is disabled → no [chk]
  (CrLf) End Character

- **Example**: (Assume the AA address value of the I-7521 module is 01)
  command: $01Y1(CrLf)
  response: !010(CrLf)
  DI1=Low
  command: $01Y2(CrLf)
  response: !011(CrLf)
  DI2=High
  command: $01Y3(CrLf)
  response: !010(CrLf)
  DI3=Low

DI mapping table:

	DI1	DI2	DI3	DI4	DI5
I-7521	No	Yes	Yes	No	No
I-7522	No	Yes	Yes	No	No
I-7522A	Yes	Yes	Yes	Yes	Yes
I-7523	No	Yes	No	No	No
I-7524	Yes	No	No	No	No
I-7527	Yes	No	No	No	No

5.3.16 $AAZNV

For: I-7521/I-7522/I-7522A/I-7524/I-7527

- **Description**:
  This function reads/sets the onboard DO 1/2/3/4/5
  $AAZN[chk](CrLf) → Reads onboard DO 1/2/3/4/5
  $AAZNV[chk](CrLf) → Sets onboard DO 1/2/3/4/5

- **Syntax**:
  $AAZNV[chk](CrLf)
  $ Delimiter character
  AA 2-character module address in HEX format. The valid range is from 00 to FF
  N 1→ Writes DO1, 2→ Writes DO2, 3→ Writes DO3
  4→ Writes DO4, 5→ Writes DO5
  (I-7522A pin assignment refer to Sec.1.4)
  V=0 Sets the DO to off,
  V=1 Sets the DO to on
  [chk] 2-character checksum. If the checksum is disabled → no [chk]
  (CrLf) End Character

- **Response**: valid command → !AAS[chk](CrLf)
  invalid command → ?AA[chk](CrLf)
  no response → syntax error, communication error, or address error
  ! Delimiter character indicating a valid command
  ? Delimiter character indicating an invalid command
  AA 2-character module address in HEX format
  S=0 DO = off
  S=1 DO = on
  [chk] 2-character checksum. If the checksum is disabled → no [chk]
  (CrLf) End Character

- **Example**: (Assume the AA address value of the I-7521 module is 01)
  command: $01Z10(CrLf)
  response: !01(CrLf)

  Set DO1=OFF

  command: $01Z21(CrLf)
  response: !01(CrLf)

  Set DO2=ON

  command: $01Z3(CrLf)
  response: !010(CrLf)

  Read DO3=OFF

Do mapping table:

	DO1	DO2	DO3	DO4	DO5
I-7521	Yes	Yes	Yes	No	No
I-7522	Yes	No	No	No	No

I-7522A	Yes	Yes	Yes	Yes	Yes
I-7523	No	No	No	No	No
I-7524	Yes	No	No	No	No
I-7527	Yes	No	No	No	No

- **Note:**
  If the host fails, the $AAZNY command will be ignored. And the response string will be !
  (In normal situation, the response string will be !AA(S))

5.3.17 #**

For: I-7521/I-7522/I-7522A/I-7523/I-7524/I-7527

- **Description**:
  This function regards each input module on the network, both digital and analog, to sample its entire input data immediately and store the data in the internal register of the module. Later the host computer can read the data individually by using the **read synchronized data $AA4 command.**

- **Syntax**:
  #**[chk](CrLf)
  # Delimiter character
  * Command character
  [chk] 2-character checksum. If the checksum is disabled → no [chk]
  (CrLf) End Character

- **Response**: no response

- **Example**: (Assume the AA address value of the I-7521 module is 01)

command:	#**(CrLf)	Orders all modules to perform synchronized sampling
response:	no response	
command:	$014(CrLf)	DI1=DI2=DI3=1
response:	!0117(CrLf)	
command:	$014(CrLf)	DI1=DI2=1, DI3=0
response:	!0113(CrLf)	
command:	$014(CrLf)	DI1=1, DI2=DI3=0
response:	!0111(CrLf)	

What is "synchronized sampling?"
The host computer can only send one command string at a time. If there are two modules, the host computer must send and receive the commands from module 1 then the commands from module 2 in response, **so there is a time delay between the two commands**. The "synchronized sampling" command is designed for all input modules. When the synchronized sampling command **#**[CrLf] is received**, **all input modules on theRS-485 network will perform the input function at the same time and store the values into the intend memory of the module.** The host computer can then send a read synchronized data command $AA4 to read the data separately.

5.3.18 $AA4

For: I-7521/I-7522/I-7522A/I-7523/I-7524/I-7527

- **Description**:
  This function reads the synchronized data.

- **Syntax** :
  $AA4[chk](CrLf)
  $ Delimiter character
  AA 2-character module address in HEX format. The valid range is from 00 to FF
  [chk] 2-character checksum. If the checksum is disabled → no [chk]
  (CrLf) End Character

- **Response** : valid command → !AASV[chk](CrLf)
  invalid command → ?AA[chk](CrLf)
  no response → syntax error, communication error, or address error
  ! Delimiter character indicating a valid command
  ? Delimiter character indicating an invalid command
  AA 2-character module address in HEX format
  S=1 first reading,
  S=0 not the first reading
  V D0=DI1, D1=DI2, D2=DI3, D3=DI4, D4=DI5
  [chk] 2-character checksum. If the checksum is disabled → no [chk]
  (CrLf) End Character

- **Example** : (Assume the AA address value of the I-7521 module is 01)
  command: #**(CrLf)
  response: no response

  Orders all modules to perform synchronized sampling

  command: $014(CrLf)
  response: !0117(CrLf)

  DI1=DI2=DI3=1

  command: $014(CrLf)
  response: !0113(CrLf)

  DI1=DI2=1, DI3=0

What is “synchronized sampling? ”
The host computer can only send one command string at a time. If there are two modules, the host computer must send and receive the commands from module 1 then the commands from module 2 in response, **so there is a time delay between the two commands**. The “synchronized sampling” command is designed for all input modules. When the synchronized sampling command **#**[CrLf] is received**, **all input modules on theRS-485 network will perform the input function at the same time and store the values into the intend memory of the module.** The host computer can then send a read synchronized data command $AA4 to read the data separately.

5.3.19 $AA5

For: I-7521/I-7522/I-7522A/I-7523/I-7524/I-7527

- **Description**:

  This function reads the reset status of the module. This is the only command that detects the module hardware watchdog failure. If the module is malfunction, the module hardware watchdog circuit will reset the module. This reset will cause the output state of the module to return to its start-up values. The start-up values may be different from the output values in use prior to the module reset. Therefore, an output command needs to be resent to the module in order to maintain the same output state once the module hardware watchdog circuit reset the module.

- **Syntax**:

  $AA5[chk](CrLf)
  $ Delimiter character
  AA 2-character module address in HEX format. The valid range is from 00 to FF
  [chk] 2-character checksum. If the checksum is disabled → no [chk]
  (CrLf) End Character

- **Response** : valid command → !AAS[chk](CrLf)
  invalid command → ?AA[chk](CrLf)
  no response → syntax error, communication error, or address error
  ! Delimiter character indicating a valid command
  ? Delimiter character indicating an invalid command
  AA 2-character module address in HEX format
  S = 0 the module has not been reset since the last reset status was read
  S = 1 the module has been reset since the last reset status was read
  [chk] 2-character checksum. If the checksum is disabled → no [chk]
  (CrLf) End Character

- **Example**: (Assume the AA address value of the I-7521 module is 01)

  command: $015(CrLf)
  response: !011(CrLf)
  command: $015(CrLf)
  response: !010(CrLf)
  command: $015(CrLf)
  response: !011(CrLf)

It is first time power-on reset

The reset status is normal

This module has been **reset** by the module Hardware watchdog. Therefore all output will be returned to its initial **start values**

- **Note:**

  When the module is first powered-on, the module should be read **once** and will find that S=1. Then the user should read the module **continually** and find that S=0. **If S is changed to 1, this module has been reset by module hardware watchdog circuit at least once. And all output is going to its start-value now.** Therefore the user needs to send an output command again to control all output values to desired states.

5.3.20 $AAF

For: I-7521/I-7522/I-7522A/I-7523/I-7524/I-7527

- **Description**:
  This function reads the firmware version number.

- **Syntax**:
  $AAF[chk](CrLf)

$	Delimiter character
AA	2-character module address in HEX format. The valid range is from 00 to FF
[chk]	2-character checksum. If the checksum is disabled → no [chk]
(CrLf)	End Character

- **Response**: valid command → !AA(number)[chk](CrLf)
  invalid command → ?AA[chk](CrLf)
  no response → syntax error, communication error, or address error

!	Delimiter character indicating a valid command
?	Delimiter character indicating an invalid command
AA	2-character module address in HEX format
number	4 or 5-character value denoting the version number
[chk]	2-character checksum. If the checksum is disabled → no [chk]
(CrLf)	End Character

- **Example**: (Assume one I-7521 module is addressed 01, another is 02)

command:	$01F(CrLf)	The version number of module 01 version is 2.0
response:	!01A2.0(CrLf)	
command:	$02F(CrLf)	The version number of module 02 version is 3.0
response:	!02A3.0(CrLf)	

5.3.21 $AAM

For: I-7521/I-7522/I-7522A/I-7523/I-7524/I-7527

- **Description**:
  This function reads the module name.

- **Syntax**:
  $AAM[chk](CrLf)

$	Delimiter character
AA	2-character HEX module address, from 00 to FF
[chk]	2-character checksum. If the checksum is disabled → no [chk]
(CrLf)	End Character

- **Response** : valid command → !AA(name)[chk](CrLf)
  invalid command → ?AA[chk](CrLf)
  no response →syntax error, communication error, or address error

!	Delimiter character indicating a valid command
?	Delimiter character indicating an invalid command
AA	2-character HEX module address
name	4 or 5-character value/string denoting the module name
[chk]	2-character checksum. If the checksum is disabled → no [chk]
(CrLf)	End Character

- **Example**:

command:	$01M(CrLf)	The name of module 01 is 7521
response:	!017521(CrLf)	
command:	$02M(CrLf)	The name of module 02 is 7523
response:	!027523(CrLf)	

5.3.22 $AA2

For: I-7521/I-7522/I-7522A/I-7523/I-7524/I-7527

- **Description**:
  This function reads the configuration code of COM2 (RS-485) stored in the EEPROM

- **Syntax**:
  $AA2[chk](CrLf)

$	Delimiter character
AA	2-character module address in HEX format. The valid range is from 00 to FF
[chk]	2-character checksum. If the checksum is disabled → no [chk]
(CrLf)	End Character

- **Response** : valid command → !AA40BDPK[chk] (CrLf)
  invalid command → ?AA[chk] (CrLf)
  no response → syntax error, communication error, or address error

!	Delimiter character indicating a valid command
?	Delimiter character indicating an invalid command
AA	2-character module address in HEX format
40	type code of module
B	short code for the Baud Rate. Refer to Sec.5.3.2 for more details
D	data bit. Refer to Sec.5.3.3 for more details
P	parity bit. Refer to Sec.5.3.4 for more details
K	checksum status. Refer to Sec.5.3.11 for more details
[chk]	2-character checksum. If the checksum is disabled → no [chk]
(CrLf)	End Character

- **Example**: (Assume that the INIT* Pin and GND Pin is connected)

```
command:  $002(CrLf)
response: !00406800(CrLf)
```

The COM2 (RS-485) has a Baud Rate of 9600 BPS, data bit 8, parity bit none, and the checksum is disabled

```
command:  $002(CrLf)
response: !0040A801(CrLf)
```

The COM2 (RS-485) has a Baud Rate of 115200 BPS, data bit 8, parity bit none, and the checksum is enabled

5.3.23 ~**

For: I-7521/I-7522/I-7522A/I-7523/I-7524/I-7527

- **Description**:
  The Host computer sends this command to tell all modules that the “Host is OK”. The host watchdog timer is reset and restarted.

- **Syntax**:
  ~**[chk](CrLf)

~	Delimiter character
[chk]	2-character checksum. If the checksum is disabled → no [chk]
(CrLf)	End Character

- **Response**: no response

- **Example**:

command:	~**(CrLf)
response:	No Response

5.3.24 ~AA0

For: I-7521/I-7522/I-7522A/I-7523/I-7524/I-7527

- **Description**:
  This function reads the module status, which will be latched until the ~AA1 command is sent**.**
  If the module status=0x04, all output commands will be ignored.

- **Syntax**:
  ~AA0[chk](CrLf)
  ~ Delimiter character
  AA 2-character module address in HEX format. The valid range is from 00 to FF
  [chk] 2-character checksum. If the checksum is disabled → no [chk]
  (CrLf) End Character

- **Response** : valid command → !AASS[chk](CrLf)
  invalid command → ?AA[chk](CrLf)
  no response → syntax error, communication error, or address error
  ! Delimiter character indicating a valid command
  ? Delimiter character indicating an invalid command
  AA 2-character module address in HEX format
  SS 2-character status value in HEX format as follows:
  Bit_0 = reserved
  Bit_1 = reserved
  Bit_2 = 0 → OK,
  1 → Host watchdog failure
  Bit_7 = 1 → Host watchdog timer is enabled.
  [chk] 2-character checksum. If the checksum is disabled → no [chk]
  (CrLf) End Character

- **Example**:
  command: ~010(CrLf)
  response: !0100(CrLf)

  The status of module 01 is OK

  command: ~020(CrLf)
  response: !0204(CrLf)

  The status of module 02 is “host watchdog failure” → HOST is off-line now

  command: ~020(CrLf)
  response: !0280(CrLf)

  The status of module 02 is ---host watchdog is running

5.3.25 ~AA1

For: I-7521/I-7522/I-7522A/I-7523/I-7524/I-7527

- **Description**:
  This function resets the module status, which will be latched until the ~AA1 command is sent. **If the module status=0x04, all output commands will be ignored.** Therefore the module status should be read first to ensure that the current status is 0. If the module status is not 0, it can only be cleared by sending the ~AA1 command.

- **Syntax**:
  ~AA1[chk](CrLf)
  ~ Delimiter character
  AA 2-character module address in HEX format. The valid range is from 00 to FF
  [chk] 2-character checksum. If the checksum is disabled → no [chk]
  (CrLf) End Character

- **Response** : valid command → !AA[chk](CrLf)
  invalid command → ?AA[chk](CrLf)
  no response → syntax error, communication error, or address error
  ! Delimiter character indicating a valid command
  ? Delimiter character indicating an invalid command
  AA 2-character module address in HEX format
  [chk] 2-character checksum. If the checksum is disabled → no [chk]
  (CrLf) End Character

- **Example**: (Assume the AA address value of the I-7521 module is 01)

command:	~010(CrLf)	The module status=0x04 → host is off-line
response:	!0104(CrLf)	
command:	**$01Z11(CrLf)**	The output command is ignored
response:	**!(CrLf)**	
command:	~011(CrLf)	Clears the module status
response:	!01(CrLf)	
command:	~010(CrLf)	Reads the module status=0x00
response:	!0100(CrLf)	
command:	**$01Z11** (CrLf)	The output command is OK
response:	!01(CrLf)	

I-752N Series Module Status Comparison:

(1) Module hardware watchdog reset

- All DO values return to their initial start values
- **No change in module status**
- Accepts host DO command to change the DO state

(2) Host software watchdog failure

- All DO values return to their saved values
- **Module status=04 → host watchdog failure**
- **Ignores all host DO commands until the module status is cleared to 0 using the ~AA1 command**

5.3.26 ~AA2

For: I-7521/I-7522/I-7522A/I-7523/I-7524/I-7527

- **Description**:
  This function reads the status of the host watchdog and the host watchdog timer value. The host watchdog timer is designed for the software host watchdog. When the software host watchdog is enabled, the host must send a ~** "HOST is OK command"(See sec.5.3.23) to all modules before the timer expires. When the ~** command is received, the host watchdog timer is reset and restarted. Use the ~AA3ETT command to enable/disable/set the host watchdog timer.

- **Syntax**:
  ~AA2[chk](CrLf)
  ~ Delimiter character
  AA 2-character module address in HEX format. The valid range is from 00 to FF
  [chk] 2-character checksum. If the checksum is disabled → no [chk]
  (CrLf) End Character

- **Response** : valid command → !AASTT[chk](CrLf)
  invalid command → ?AA[chk](CrLf)
  no response → syntax error, communication error, or address error
  ! Delimiter character indicating a valid command
  ? Delimiter character indicating an invalid command
  AA 2-character module address in HEX format
  S=0 The host watchdog timer is disabled
  S=1 The host watchdog timer is enabled
  TT 2-character timer value in HEX format. The valid range is from 00 to FF. Units=0.1 sec
  [chk] 2-character checksum. If the checksum is disabled → no [chk]
  (CrLf) End Character

- **Example**:
  command: ~012(CrLf)
  response: !01000(CrLf)
  command: ~022(CrLf)
  response: !0210A(CrLf)

The host watchdog timer of module 01 is disabled

The host watchdog timer of module 02 is enabled and is set to 0.1*10 =1 second.

I-752N Series Module Status Comparison:

(1) Module hardware watchdog reset

- All DO values return to their initial start values
- **No change in module status**
- Accepts a host DO command to change the DO state

(2) Host software watchdog failure

- All DO values return to their initial save values
- **Module status=04 → host watchdog failure**
- **Ignores all host DO commands until the module status is cleared to 0 using the ~AA1 command**

5.3.27 ~AA3ETT

For: I-7521/I-7522/I-7522A/I-7523/I-7524/I-7527

- **Description**:
  This function enables/disables the host watchdog timer and sets the value. The host watchdog timer is designed for the software host watchdog. When the software host watchdog is enabled, the host must send a ~** "HOST is OK" command (see Sec 5.3.23), to all modules before the timer expires. When the ~** command is received, the host watchdog timer is reset and restarted. Using the ~AA2 command to read the host watchdog status and value.

- **Syntax**:
  ~AA3ETT[chk](CrLf)

~	Delimiter character
AA	2-character module address in HEX format. The valid range is from 00 to FF
E=0	Disables the host watchdog timer
E=1	Enables the host watchdog timer
TT	2-character timer value in HEX format. The valid range is from 00 to FF. Units=0.1sec
[chk]	2-character checksum. If the checksum is disabled → no [chk]
(CrLf)	End Character

- **Response** : valid command → !AASTT[chk](CrLf)
  invalid command → ?AA[chk](CrLf)
  no response → syntax error, communication error, or address error

!	Delimiter character indicating a valid command
?	Delimiter character indicating an invalid command
AA	2-character module address in HEX format
S=0	The host watchdog timer is disabled
S=1	The host watchdog timer is enabled
TT	2-character timer value in HEX format. The valid range is from 00 to FF.Units=0.1 sec
[chk]	2-character checksum. If the checksum is disabled → no [chk]
(CrLf)	End Character

- **Example**:

```
command:  ~013000(CrLf)
response :  !01000(CrLf)
```

Disables the host watchdog timer for module 01

```
command:  ~02310A(CrLf)
response :  !0210A(CrLf)
```

The host watchdog timer of module 02 is enabled and is set to 0.1*10 =1 second.

5.3.28 ~AA4P & ~AA4S

For: I-7521/I-7522/I-7522A/I-7524/I-7527

- **Description**:
  This function reads power-on/safe value.
  (1) When the module is **first powered-on**, all output channels will revert to their initial **power-on values**.
  (2) If the module is **malfunction**, the hardware module watchdog will reset the module and all output channels will **also revert to their power-on values**. **These power-on values may be different from the previous values that existed before the module was reset.** Therefore, new output commands must be set to return all output values to their desired states.
  (3) When the host watchdog is enabled and the **host is malfunction**, all output values will change to their **safe values** and the module status will be set to 0x04. If the module status is 0x04, all output commands will be ignored until the module status is cleared using the ~AA1 command (see sec.5.3.23). Therefore an ~AA1 command must be sent before any new output commands are sent to return all output values to their desired states.

- **Syntax**:
  ~AA4P[chk](CrLf) → reads the power-on value
  ~AA4S[chk](CrLf) → reads the safe value
  ~ Delimiter character
  AA 2-character module address in HEX format. The valid range is from 00 to FF
  [chk] 2-character checksum. If the checksum is disabled → no [chk]
  (CrLf) End Character

- **Response** : valid command → !AAV[chk](CrLf)
  invalid command → ?AA[chk](CrLf)
  no response → syntax error, communication error, or address error
  ! Delimiter character indicating a valid command
  ? Delimiter character indicating an invalid command
  AA 2-character module address in HEX format
  V D0=DO0, D1=DO1, D2=DO3, D3=DO4, D4=DO5
  [chk] 2-character checksum. If the checksum is disabled → no [chk]
  (CrLf) End Character

- **Example**: (Assume the AA address value of the I-7521 module is 01)
  command: ~014P(CrLf)
  response: !017(CrLf)

  The power-on value is DO-1/2/3 all ON

  command: ~014S(CrLf)
  response: !010(CrLf)

  The safe value is DO-1/2/3 all OFF

5.3.29 ~AA5P & ~AA5S

For: I-7521/I-7522/I-7522A/I-7524/I-7527

- **Description**:
  This function sets the current state of the digital output as a power-on/safe value.
  (1) When the module is **first powered-on**, all output channels will revert to their **power-on values**.
  (2) If the module is **malfunction**, the hardware module watchdog will reset the module and all output channels will **also revert to their power-on values**. **These power-on values may be different from the previous values that existed before the module was reset.** Therefore, new output commands must be set to return all output values to their desired states.
  (3) When the host watchdog is enabled and the **host is malfunction**, all output values will change to their **safe values** and module status will change to 0x04. If the module status is 0x04, all output command will be ignored until the module status is cleared using the ~AA1 command (see sec3.23). Therefore an ~AA1 command must be sent before any new output command are sent to return all output values to their desired states.

- **Syntax**:
  ~AA5P[chk](CrLf) → sets the power-on value
  ~AA5S[chk](CrLf) → sets the safe value
  ~ Delimiter character
  AA 2-character module address in HEX format. The valid range is from 00 to FF
  [chk] 2-character checksum. If the checksum is disabled → no [chk]
  (CrLf) End Character

- **Response** : valid command → !AAV[chk](CrLf)
  invalid command → ?AA[chk](CrLf)
  no response → syntax error, communication error, or address error
  ! Delimiter character indicating a valid command
  ? Delimiter character indicating an invalid command
  AA 2-character module address in HEX format
  V D0=DO0, D1=DO1, D2=DO3, D3=DO4, D4=DO5
  [chk] 2-character checksum. If the checksum is disabled → no [chk]
  (CrLf) End Character

- **Example**: (Assume the AA address value of the I-7521 module is 01)
  command: ~015P(CrLf)
  response: !017(CrLf)

  Set the power-on value to DO-1/2/3 all ON

  command: ~025S(CrLf)
  response: !020(CrLf)

  Sets the safe value to DO-1/2/3 all OFF

5.3.30 $AAU

For: I-7521/I-7522/I-7522A/I-7523/I-7524/I-7527

- **Description**:
  This function reads data from the RS-232 COM port buffer.
  There is 1Kb queue buffer for its local RS-232 device. All input data from RS-232 COM port can be stored in the queue buffer until the Host PC has time to read it. These features allow the Host PC to be linked to thousands of RS-232 devices without any loss of data. Any RS-232 device should obey the rules of the request-reply protocol. In other words, RS-232 devices are passive. If they do not receive any commands, they will not send any messages out. However, since active devices are frequently developed, ICPDAS controllers are designed with a buffer to receive these messages in situations such as this.
  Buffer operation rules:
  Rule 1: The buffer is enabled after being powered-on.
  Rule 2: The (delimiter) AA command (Refer to Sec.5.3.10) disables the buffer operation for that port
  Rule 3: After disabling the buffer, the controller will wait for X seconds (=timeout1, refer to Sec.5.3.33) for a response from the RS-232 device. The response will then be transfered to COM2. If no message is received, the buffer will be re-enabled.

- **Syntax**:
  $AAU[chk](CrLf)

$	Delimiter character
AA	2-character module address in HEX format. The valid range is from 00 to FF
[chk]	2-character checksum. If the checksum is disabled → no [chk]
(CrLf)	End Character

- **Response :** valid command → (data)[chk](CrLf)
  invalid command → ?AA[chk](CrLf)
  no response → The buffer is empty, syntax error or communication error, or address error

!	Delimiter character indicating a valid command
?	Delimiter character indicating an invalid command
AA	2-character module address in HEX format
[chk]	2-character checksum. If the checksum is disabled → no [chk]
(CrLf)	End Character

- **Example:**

Command:	$01U(CrLf)	Retrieves "data1" from the buffer of the port that is addressed 01
Response:	data1(CrLf)	
Command:	$01U(CrLf)	Retrieves another data: "data2" from the buffer of the port that is addressed 01
Response:	data2	
Command:	$02U(CrLf)	No data in the buffer of the port that is addressed 02
Response:		

- **Note:**

(1) If the messages from the RS-232 device includes an End Character (refer to Sec. 5.3.12), set the "fit CrLf mode" for the RS-232 port before sending the $AAU command. If the CrLf mode of the RS-232 port is 4, all messages in the buffer can be read at one time, regardless of the End Character of the messages. If the "fit CrLf mode" is set to either 0,1, 2 or 3, the messages will be read individually.

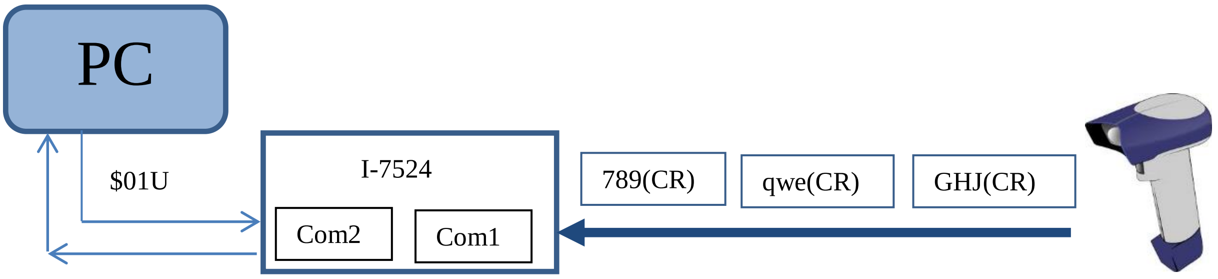
(2) If the End Character of the message from the RS-232 device is 0x0D, only CrLf mode = 0 or CrLf mode = 4 can be set. Changing to other CrLf modes will corrupt the integrity of the unread data in the buffer.

(3) Repeat this command several times to ensure that the buffer is empty (CrLf mode=0/1/2/3).

(4) There are two mode for "$AAU" to read com port ring buffer.

First mode: Read all data at one time.

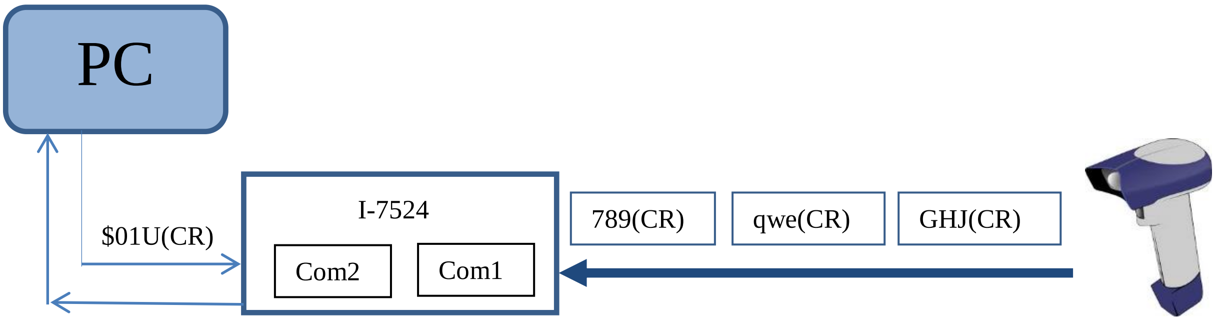
a. Assume barcode sends data three times. First is "789(CR)", second is "qwe(CR)", three is "GHJ(CR)". "(CR)" is 0x0D.
b. The CrLf mode of Com1 is 4. Refer to Sec.5.3.12 about command "$AATN[CrLfmode]".
c. The CrLf mode of Com2 is 4.
d. PC sends "$01E1" to enable COM1 prefixed address function.
e. PC sends command "$01U" will read all data at one time.
The data is "!01789(CR)qwe(CR)GHJ(CR)".



!01789(CR)qwe(CR)GHJ(CR)

Second mode: Read the data individually

a. Assume barcode sends data three times. First is "789(CR)", second is "qwe(CR)", three is "GHJ(CR)". "(CR)" is 0x0D.
b. The CrLf mode of Com1 is 0. Refer to Sec.5.3.12 about command "$AATN[CrLfmode]".
c. The CrLf mode of Com2 is 4.
d. PC sends "$01E1" to enable COM1 prefixed address function.
e. PC sends command "$01U(CR)" will read one data one time.
First time, send "$01U" ➔ Get the data "!01789(CR)"
Second time, send "$01U" ➔ Get the data "!01qwe(CR)"
Third time, send "$01U" ➔ Get the data "!01GHJ(CR)"



5.3.31 $AAL[data]

For: I-7522A

- **Description**:
  This function will write data to the expansion board DO 0/1/2/3
  Note: Supported by the firmware version 3.01 and upward.

- **Syntax**:
  $AALbbbb[chk](CrLf)
  $AALcb[chk](CrLf)
  $AALh[chk](CrLf)
  $ Delimiter character
  AA 2-character module address in HEX format. The valid range is from 00 to FF

	0	1	2	3
b	Set to On	Set to Off		
c	DO0(pin25)	DO1(pin26)	DO2(pin27)	DO3(pin28)
Pin assignment refers to Sec.1.4				

  h 4-bit DO value in HEX format. DO0 is at LSB. The valid values are 0~9, a~f and A~F.
  [chk] 2-character checksum. If the checksum is disabled → no [chk]
  (CrLf) End Character

- **Response:** valid command → !AA[chk](CrLf)
  invalid command → ?AA[chk](CrLf)
  no response → syntax error, communication error, or address error
  ! Delimiter character indicating a valid command
  ? Delimiter character indicating an invalid command
  AA 2-character module address in HEX format
  [chk] 2-character checksum. If the checksum is disabled → no [chk]
  (CrLf) End Character

- **Example:** (Assume the AA address value of the I-7522A module is 01)
  command: $01L1000(CrLf)
  response: !01 (CrLf)

  Sets DO3=ON, DO2, DO1, DO0=0

  command: $01L21(CrLf)
  response: !01 (CrLf)

  Sets DO2=ON, other DOn values are unchanged.

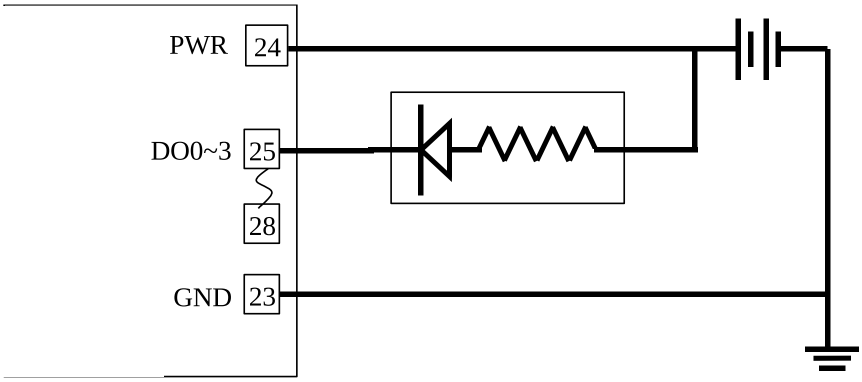
  command: $01L30(CrLf)
  response: !01 (CrLf)

  Sets DO3=OFF, other DOn values are unchanged.

  command: $01LE(CrLf)
  response: !01 (CrLf)

  Sets DO0=OFF, DO1, DO2, DO3=ON

- **Note:**
  If the host fails, the $AAL command will be ignored, and the response string will be **!**
  (In normal situations, the response string will be !AA)



5.3.32 $AAR

For: I-7522A

- **Description:**
  This function reads the expansion board DI0(pin25), DI1(pin26), DI2(pin27), DI3(pin28).
  Note: Supported by the firmware version 3.01 and upward.

- **Syntax**:
  $AAR[chk](CrLf)

$	Delimiter character
AA	2-character module address in HEX format. The valid range is from 00 to FF
[chk]	2-character checksum. If the checksum is disabled → no [chk]
(CrLf)	End Character

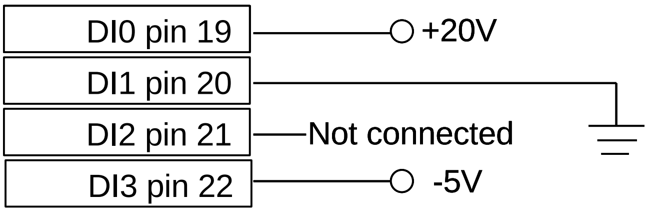
- **Response:** valid command → !AAS[chk](CrLf)
  invalid command → ?AA[chk](CrLf)
  no response → syntax error, communication error, or address error

!	Delimiter character indicating a valid command
?	Delimiter character indicating an invalid command
AA	2-character module address in HEX format
S	4-bit DI value in HEX format. DI0 is at LSB. The valid values are 0~9,a~f and A~F.

DI=low→0
DI=high→1
DI floating will return a High

[chk]	2-character checksum. If the checksum is disabled → no [chk]
(CrLf)	End Character

- **Example:** (Assume the AA address value of the I-7522A module is 01)



command: $01R(CrLf)
response: !015(CrLf)

DI3, DI1=low(0), DI2, DI0=high(1)

5.3.33 $AAJN[timeout]

For: I-7521/I-7522/I-7522A/I-7523/I-7524/I-7527

- **Description**:
  The function reads/sets the delay time before determining whether the end of a command/response has been sent/received
  $AAJN [chk] (CrLf)→Reads the timeout value
  $AAJN[timeout] [chk] (CrLf)→Sets the timeout value
  Note: Supported by the firmware version 3.0 and upward.

- **Syntax:**
  $AAJN[timeout] [chk] (CrLf)

$	Delimiter character
AA	2-character module address in HEX format. The valid range is from 00 to FF
N=0	COM 2 timeout0
N=1	COM 1/3/4/5/6/7/8 timeout1
N=2	COM 1/3/4/5/6/7/8 timeout2
[timeout]	Delay time value (ms). Valid range is 0 to 4294967259.
[chk]	2-character checksum. If the checksum is disabled → no [chk]
(CrLf)	End Character

- **Response :** valid command → !AA[timeout value][chk](CrLf)
  invalid command → ?AA[chk](CrLf)
  no response→ buffer is empty, syntax error, communication error, address error

!	Delimiter character indicating a valid command
?	Delimiter character indicating an invalid command
AA	2-character module address in HEX format
[chk]	2-character checksum. If the checksum is disabled → no [chk]
(CrLf)	End Character

- **Example:** (Assume the AA address value of the I-7522 module is 01)

command: $01J01000(CrLf)
response: !01(CrLf)

Sets the timeout value for the RS-485(COM 2) to 1000ms

command: $01J11500(CrLf)
response: !01(CrLf)

Sets the timeout value of the RS-232(COM 1) to 1500ms

command: $01J1(CrLf)
response: !011500(CrLf)

Reads the timeout1 value of COM1
The timeout1 value of COM1 is 1500ms.

The default timeout1 value for all RS-232 COM ports is 1000ms.

- **Notes:**
  RS-485: timeout0

[A] Only valid when CrLf mode=4 is used.

[B] When CrLf mode is set to 4, that is no command END character, the I-752N modules must wait for a specific time period (timeout0) without receiving any further data in order to determine the end of the command.

RS-232: timeout1, timeout2

[A] When data is sent from the command port (RS-485) is required to be passed to the RS-232 port, the I-752N modules will send the data to the RS-232 port and wait for a response from the RS-232 port.

(a) If no response has been received before the "timeout1" period has elapsed, then the module will stop waiting.
(If the RS-232 port is not in wait mode and the I-752N modules receive data from that port, the data will be sent to the buffer of that port. The data can then be read using the "$AAU" command.)

(b) If any data is received before the timeout1 period has elapsed,

(1) If CrLf mode=0~3, any response data received will be send to the command port (RS-485) after the I-752N modules receive the END character. However, if the “timeout1” period elapsed and the I-752N modules still haven’t received the END character, and the "timeout2" period has elapsed without receiving any further data, then the I-752N modules will send the received data to the COMMAND port.

(2) If CrLf mode=4, the I-752N modules will wait for "timeout2" to elapse without receiving any more data, then will send the received data to the COMMAND port.

[B] "timeout1", begins when the I-752N modules start to send the bypass data to RS-232 port, so the command transmission time from I-752n to RS-232 device must also be considered.

[C] For "timeout2", begins at when RS-232 COM port receives one character and restarts to count “timeout2” until next character is received.
The user must consider the buffer trigger level setting of the UART (COM1 with only a byte buffer, COM3 or later using 16c550 with a 16 byte FIFO, the default trigger level is set to 8. The trigger level can be set using the "$AAG1N" command, N can be either "1," "4," "8" or "14."

[D]The timeout1must be greater than timeout2.

5.3.34 $AAGN[triggerlevel]

For: I-7521/I-7522/I-7522A/I-7523/I-7524/I-7527

- **Description:**
  This function reads/sets the Trigger Level
  $AAGN[chk](CrLf) → Reads the Trigger Level
  $AAGN[Trigger Level][chk](CrLf) → Sets the Trigger Level
  Note: Supported by the firmware version 3.0 and upward.

- **Syntax**:
  $AAGN[trigger level][chk](CrLf)
  $ Delimiter character
  AA 2-character module address in HEX format. The valid range is from 00 to FF
  [Trigger Level]: 1/4/8/14 (COM3/4/5/6/7/8 only)
  N=0 Reads/Sets the Trigger Level of the COM 2
  N=1 Reads/Sets the Trigger Level of the COM 1/3/4/5/6/7/8
  [chk] 2-character checksum. If the checksum is disabled → no [chk]
  (CrLf) End Character

- **Response:** valid command → !AA[trigger level][chk](CrLf)
  invalid command → ?AA[chk](CrLf)
  no response → syntax error, communication error, or address error
  ! Delimiter character indicating a valid command
  ? Delimiter character indicating an invalid command
  AA 2-character module address in HEX format
  [chk] 2-character checksum. If the checksum is disabled → no [chk]
  (CrLf) End Character

- **Example:** (Assume the AA address value of the I-7523 module is 01)
  command: $01G0(CrLf)
  response: !011(CrLf)
  Reads the trigger level of COM2 is 1
  command: $02G14(CrLf)
  response: !02(CrLf)
  Sets the trigger level of COM3 to 4
  command: $01G18(CrLf)
  response : ?01(CrLf)
  only COM3/4/5/6/7/8 can be set

- **Note:**
  The Trigger Level for COM1 and COM2 can’t be changed.

5.3.35 @AA[data]

For: I-7521/I-7522/I-7522A/I-7523/I-7524/I-7527

- **Description:**
  This function reads/sets the onboard DI 1/2/3/4/5 and DO 1/2/3/4/5 values
  @AA → Reads the onboard DI 1/2/3/4/5 and DO 1/2/3/4/5
  @AAh → Sets the onboard DO 1/2/3 for the 7521/7522/7523 modules
  @AAhh → Sets the onboard DO 1/2/3/4/5 for the 7523A/7522A/7524/7527 modules
  Note: Supported by the firmware version 3.0 and upward.

- **Syntax**:
  @AA[chk](CrLf)
  @AAh[chk](CrLf)
  @AAhh[chk](CrLf)
  $ Delimiter character
  AA 2-character module address in HEX format. The valid range is from 00 to FF
  h 4-bit DO value in HEX format. DO1 is set at LSB. The valid values is 0~9, a~f , A~F
  [chk] 2-character checksum. If the checksum is disabled → no [chk]
  (CrLf) End Character

- **Response:**
  @AA valid command → >AAbbcc[chk](CrLf)
  invalid command → ?[chk](CrLf)
  no response → syntax error, communication error, or address error
  > Delimiter character indicating a valid command
  AA 2-character module address in HEX format
  bb 2-character hex value of DO. DO1 is at LSB. Valid value is from 00 to FF
  cc 2-character hex value of DI. DI1 is at LSB. Valid value is from 00 to FF
  DI=low→0
  DI=high→1
  DI floating will get High
  DO=1 on
  DO=0 off
  [chk] 2-character checksum. If the checksum is disabled → no [chk]
  (CrLf) End Character
  @AAh/hh valid command → >[chk](CrLf)
  invalid command → ?[chk](CrLf)
  no response → syntax error, communication error, or address error
  > Delimiter character indicating a valid command
  ? Delimiter character indicating an invalid command
  [chk] 2-character checksum. If the checksum is disabled → no [chk]
  (CrLf) End Character

- **Example:** (Assume the AA address value of the I-7522A module is 01)

  command: @0108(CrLf)
  response: >(CrLf)

  Sets DO1=OFF, DO2=OFF, DO3=OFF, DO4=ON.

  command: @02
  response: >02081F(CrLf)

  Reads DO1=OFF, DO2=OFF, DO3=OFF, DO4=ON,
  DO5=OFF, DI1=high(1), DI2=high(1), DI3=high(1), DI4=high(1), DI5=high(1),

  (Assume the AA address value of the I-7521 module is 01)

  command: @017
  response: >(CrLf)

  Sets DO1=ON, DO2=ON, DO3=ON

- **Note:**
  If the host fails, the @AAh/hh command will be ignored and the response string will be"!." (In normal situations, the response string will be ">")

5.3.36 #AABBHH

For: I-7521/I-7522/I-7522A/I-7524/I-7527

- **Description:**
  The function sets the multiple onboard DO-1/2/3/4/5 values.
  (I-7522A pin assignment refers to Sec.1.4)
  Note: Supported by the firmware version 3.0 and upward.

- **Syntax**:
  #AABBHH[chk](CrLf)

#	Delimiter character
AA	2-character module address in HEX format. The valid range is from 00 to FF
BB	00/0A
HH	2-character DO value in HEX format. DO1 is set at LSB. The valid value range is from 00 to FF
[chk]	2-character checksum. If the checksum is disabled → no [chk]
(CrLf)	End Character

- **Response:** valid command → >[chk](CrLf)
  invalid command → ?[chk](CrLf)
  no response → syntax error, communication error, or address error

>	Delimiter character indicating a valid command
?	Delimiter character indicating an invalid command
[chk]	2-character checksum. If the checksum is disabled → no [chk]
(CrLf)	End Character

- **Example:** (Assume the AA address value of the I-7521 module is 01)

```
command:    #010003(CrLf)
response:    >(CrLf)
```

Sets DO1=ON, DO2=ON, DO3=OFF

```
command:    #010A02(CrLf)
response:    >(CrLf)
```

Sets DO1=OFF, DO2=ON, DO3=OFF

- **Note:**
  If the host fails, the #AABBHH command will be ignored and the response string will be "!." (In normal situations, the response string will be " >")

5.3.37 #AABCDD

For: I-7521/I-7522/I-7522A/I-7524/I-7527

- **Description:**
  This function sets the single onboard DO-1/2/3/4/5 values.
  (I-7522A pin assignment refers to Sec.1.4)
  Note: Supported by the firmware version 3.0 and upward.

- **Syntax**:
  #AABCDD[chk](CrLf)

#	Delimiter character
AA	2-character module address in HEX format. The valid range is from 00 to FF
B	1/A
C	0 → DO1, 1 → DO2, 2 → DO3, 3 → DO4, 4 → DO5
DD	00 → OFF, 01 → ON
[chk]	2-character checksum. If the checksum is disabled → no [chk]
(CrLf)	End Character

- **Response:** valid command → >[chk](CrLf)
  invalid command → ?[chk](CrLf)
  no response → syntax error, communication error, or address error

>	Delimiter character indicating a valid command
?	Delimiter character indicating an invalid command
AA	2-character module address in HEX format
[chk]	2-character checksum. If the checksum is disabled → no [chk]
(CrLf)	End Character

- **Example:** (Assume the AA address value of the I-7521 module is 01)

```
command:    #011201(CrLf)
response:    >(CrLf)
```

Sets DO3=ON

```
command:    #01A000(CrLf)
response:    >(CrLf)
```

Sets DO1=OFF

- **Note:**
  If the host fails, the #AABCDD command will be ignored and the response string will be "!." (In normal situations, the response string will be ">")

5.3.38 $AAEV

For: I-7521/I-7522/I-7522A/I-7523/I-7524/I-7527

- **Description:**
  This function reads/sets the status of the prefixed address byte **on the response,** which lets the host know RS-232 device the response comes from.
  Note: Supported by the firmware version 3.05 and upward.

- **Syntax**:
  $AAEV[chk](CrLf)

$	Delimiter character
AA	2-character module address in HEX format. The valid range is from 00 to FF
V=0	Prefixed address byte disabled
V=1	Prefixed address byte enabled
[chk]	2-character checksum. If the checksum is disabled → no [chk]
(CrLf)	End Character

- **Response:** valid command →!AA[V] [chk](CrLf)
  invalid command → ?[chk](CrLf)
  No response → syntax error, communication error, or address error

!	Delimiter character indicating a valid command
?	Delimiter character indicating an invalid command
AA	2-character module address in HEX format
V=0	Prefixed address byte disabled
V=1	Prefixed address byte enabled
[chk]	2-character checksum. If the checksum is disabled → no [chk]
(CrLf)	End Character

- **Example:** (Assume the AA address value of the I-7521 module is 01)

command: $01E(CrLf)
response: !010(CrLf)

Reads the status of the prefixed address byte for COM1. The prefixed address byte is disabled.

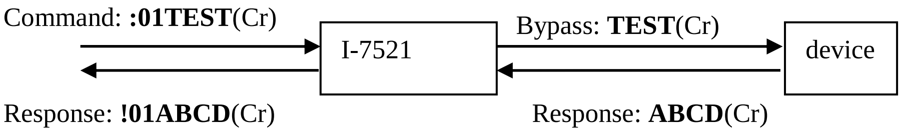
command: $01E1(CrLf)
response: !01(CrLf)

Sets the status of the prefixed address byte to enabled.

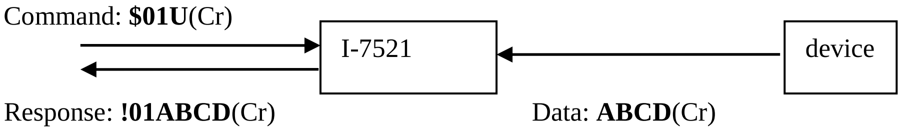
- **Note:**

If the prefixed address byte is enabled, the response for [delimiter]AA[bypass data] and $AAU will be prefixed with !AA.

Example 1: **[delimiter]AA[bypass data]**



Example 2: **$AAU**



5.3.39 $AAHV

For: I-7521/I-7522/I-7522A/I-7523/I-7524/I-7527

- **Description:**
  This function reads/sets the mode of bypassing the data string to COM2**.**
  When this function is enabled, the data from the COM1/3/4/5/6/7/8 will be sent to COM2 directly.
  Note: Supported by the firmware version 3.08 and upward.

- **Syntax**:
  $AAHV[chk](CrLf)

$	Delimiter character
AA	2-character module address in HEX format. The valid range is from 00 to FF
V=0	Disable the mode for bypassing the data string to COM2.
V=1	Enable the mode for bypassing the data string to COM2.
[chk]	2-character checksum. If the checksum is disabled → no [chk]
(CrLf)	End Character

- **Response:** valid command →!AA[V] [chk](CrLf)
  invalid command → ?[chk](CrLf)
  No response → syntax error, communication error, or address error

!	Delimiter character indicating a valid command
?	Delimiter character indicating an invalid command
AA	2-character module address in HEX format
V=0	The mode for bypassing the data string to COM2 is disabled.
V=1	The mode for bypassing the data string to COM2 is enabled.
[chk]	2-character checksum. If the checksum is disabled → no [chk]
(CrLf)	End Character

- **Example:** (Assume the AA address value of the I-7522 module is 01)

```
command:   $02H1(CrLf)
response:   !02(CrLf)
```

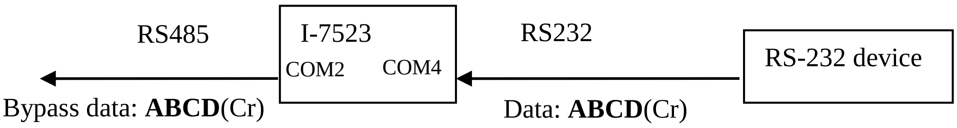
Set the mode of COM3 to enable.

```
command:   $02H(CrLf)
response:   !021(CrLf)
```

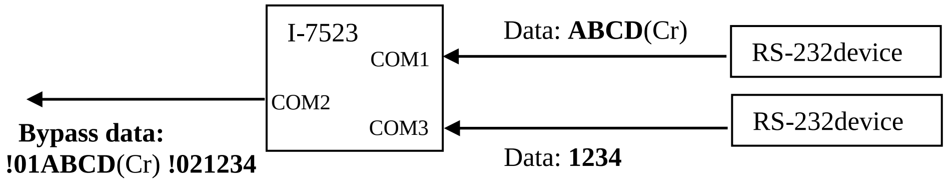
Read the mode of COM3.
The mode is enabled.

- **Note:**

1. Example : Assume the mode of COM4 of I-7523 is enabled. I-7523 will bypass the data "ABCD(Cr)" to COM2 directly.



2. When the mode is enabled, please note the communication of COM2 because the COM2 is connected to RS-485 network. The RS-485 network is half-duplex.
3. The command can cooperate with the command “$AAEV”.
   Assume the AA address value of the I-7523 module is 01.
   Set COM1: $01E1 ➔ Enable the mode of the prefixed address byte on the response.
   $01H1➔ Enable the mode of bypassing the data string to COM2.
   Set COM3: $02E1 ➔ Enable the mode of the prefixed address byte on the response.
   $02H1➔ Enable the mode of bypassing the data string to COM2.
   Assume the data “ABCD(Cr)” is quicker than data “1234”.



5.3.40 $AAIV

For: I-7521/I-7522/I-7522A/I-7523/I-7524/I-7527

- **Description:**
  This function will restore the system to factory default setting right now.
  Note: Supported by the firmware version 3.08 and upward.

- **Syntax**:
  $AAIV[chk](CrLf)

$	Delimiter character
AA	2-character module address in HEX format. The valid range is from 00 to FF
V=1	Enable the mode for restoring factory default setting.
[chk]	2-character checksum. If the checksum is disabled → no [chk]
(CrLf)	End Character

- **Response:** valid command →!AA[chk](CrLf)
  invalid command → ?[chk](CrLf)
  No response → syntax error, communication error, or address error

!	Delimiter character indicating a valid command
?	Delimiter character indicating an invalid command
AA	2-character module address in HEX format
[chk]	2-character checksum. If the checksum is disabled → no [chk]
(CrLf)	End Character

- **Example:** (Assume the AA address value of the I-7522 module is 01)

```
command:   $02I1(CrLf)
response:   !02(CrLf)
```

Restore to factory default setting for all com ports.

```
command:   $01I1(CrLf)
response:   !01(CrLf)
```

Restore to factory default setting for all com ports.

- **Note:**
  Please refer to the beginning of Sec.3 “Quick Start” for factory default setting.

Chapter 6 Applications

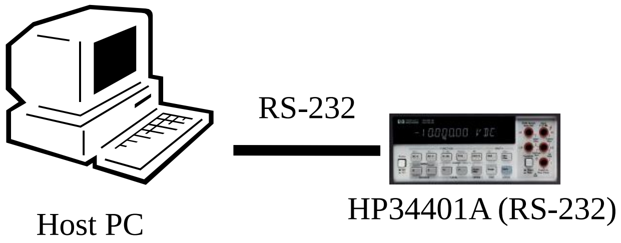
6.1. Cnnecting to the HP34401A

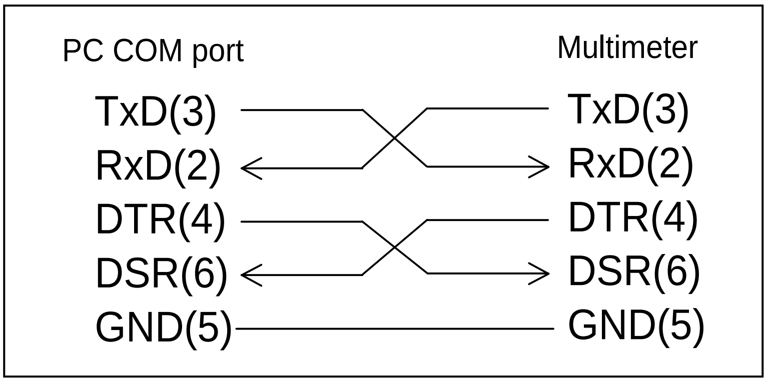
6.1.1 Connect the I-752N module to the HP34401A

The stop-bit of the HP34401A must be set as two stop-bits. The COM1/COM3 port of the I-7522 module and COM1/COM3/COM4 port of the I-7523 module can support two stop-bits, so they can be linked to the HP34401A. The I-7522 module can be linked to a single HP34401A. The I-7523 module can be linked to two HP34401As. Please refer to Section 6.1.2 ~ 6.1.6 for more information.

6.1.2 Connecting a PC to the HP34401A

A PC can be linked to the HP34401A using either COM1 or COM2 as below:



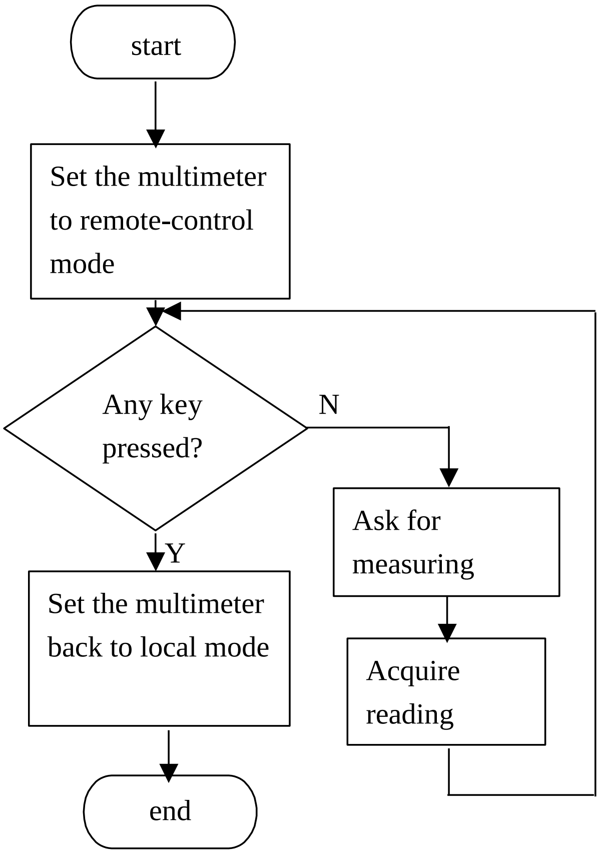


The default settings of the HP34401A are as follows:

- Baud Rate=9600
- Data-bit=7
- Parity-bit=EVEN
- Stop-bit=2

- TxD: send commands to the RS-232 HOST
- RxD: receive commands from the RS-232 HOST
- DTR: Set the HP34401A as active HIGH to enable the RS-232 HOST to send commands
- DSR: Set the RS-232 HOST as active HIGH to enable the HP34401A to send the results back to the RS-232 HOST

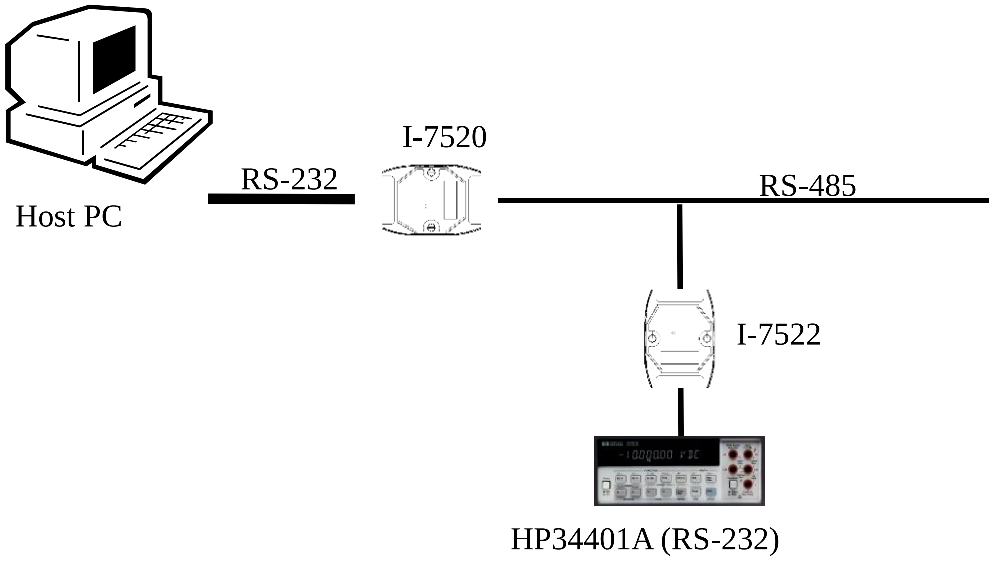
The demo program, hp34401a.c, is designed to allow a Host PC to be connected to an HP34401A. Refer to the companion CD for the source code of the hp34401a.c file. The flow chart illustrating the functionality of the hp34401a.c file is shown below:

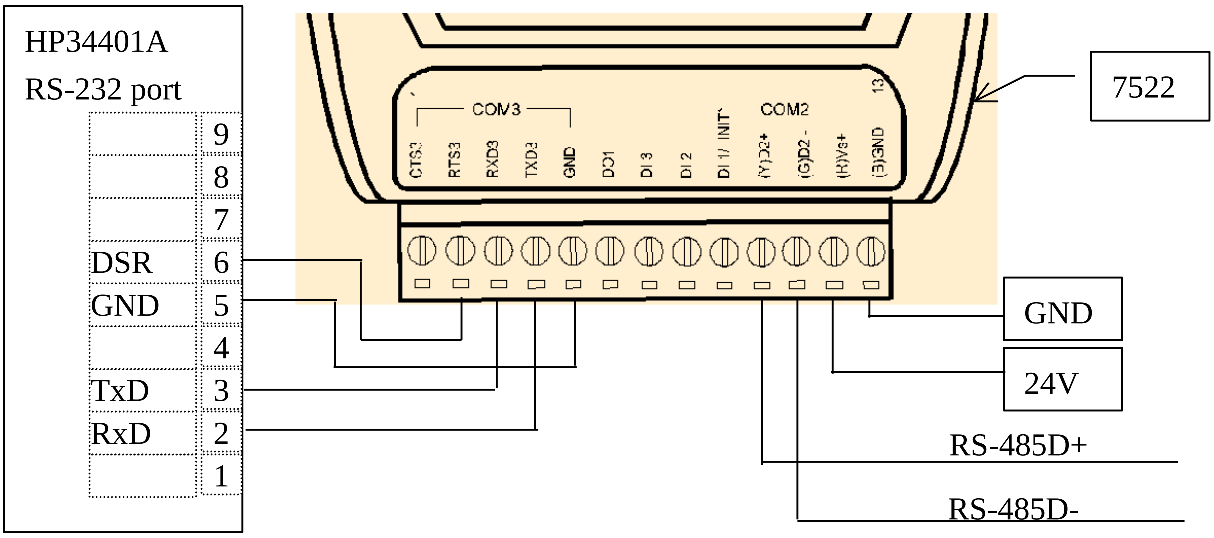

start
Set the multimeter to remote-control mode
Any key pressed?
N
Ask for measuring
Acquire reading
Y
Set the multimeter back to local mode
end

Note: the PC COM port should be 16550 compatible.

6.1.3 Connecting a single I-7522 to a single HP34401A

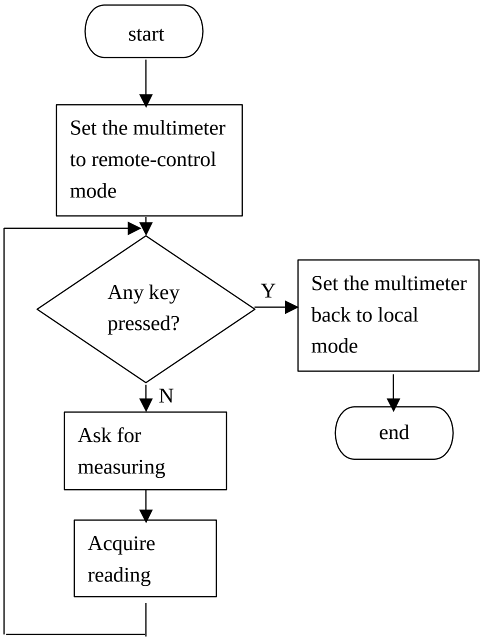
The diagram below shows how to connect a single PC to a remote HP34401A on an RS-485 network. The I-7520 module is used to convert the RS-232 signal from PC to an RS-485 signal. The I-7522 module is used as an “Addressable RS-232 converter” for the HP34401A since there is no address setting in the HP34401A.





The demo program, hp22_1.c, is designed to allow a Host PC to be linked to a remote HP34401A. Refer to the companion CD for the source code of the hp22_1.c file. The key features of hp22_1.c are as follows:

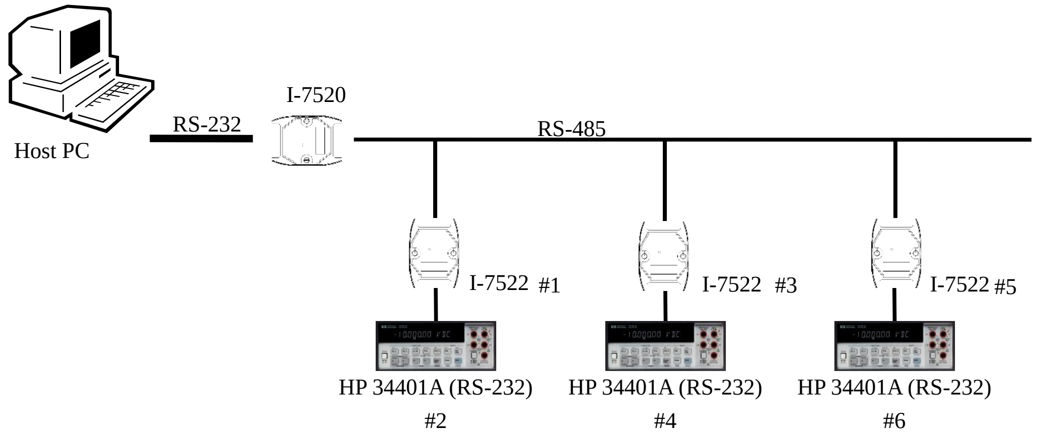
- The RTS3 pin of COM3 port must first be set as active HIGH to enable the HP34401A.
- A flow chart illustrating the functionality of the hp22_1.c file is shown below:


start
Set the multimeter to remote-control mode
Any key pressed?
Y
N
Set the multimeter back to local mode
end
Ask for measuring
Acquire reading

Note: the PC COM port should be 16550 compatible.

6.1.4 Connecting multiple I-7522 modules and multiple HP34401A

The diagram below shows how to connect a single PC to multiple remote HP34401As on an RS-485 network. The I-7520 module is used to convert the RS-232 signal from the PC to an RS485 signal. The I-7522 module is used as an "Addressable RS232 converter" for the HP34401A, since there is no address setting on the HP34401A.
Each I-7522 module has a unique address on the RS485 network. Each HP34401A shares the same address-range as its corresponding I-7522 module, which allows each HP34401A to have a unique address in this configuration.



The demo program, hp22_m.c, is designed to allow a Host PC to be linked to a remote HP34401A. Refer to the companion CD for the source code of the hp22_m.c file. The key features of hp22_m.c are as follows:

- **The configuration of the COM port on the I-7522 module is as follows:**

	COM2 (485)	COM3 (232)
Baud Rate	9600 (default)	9600 (default)
Parity	None (default)	Even
Data	8 (default)	7
Stop-bit	1 (default)	2

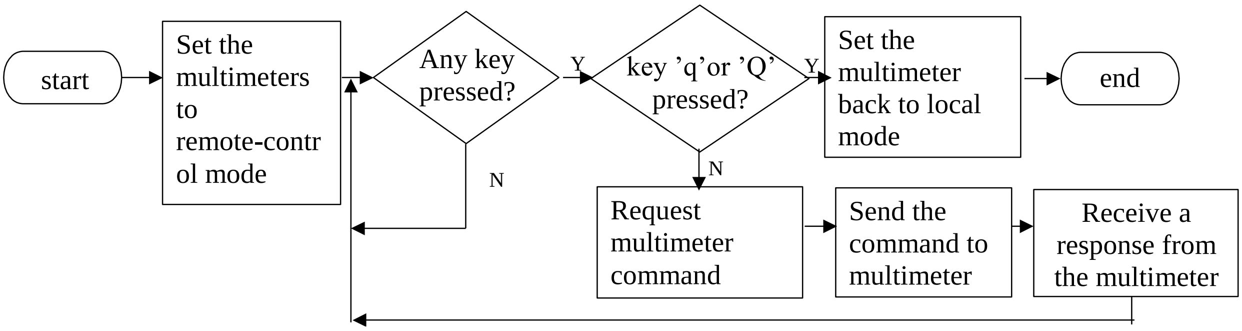
- **The RTS3 pin of the COM3 port on each module must first be set as active HIGH to enable the HP34401A.**

- **The address mapping for this configuration is as follows:**

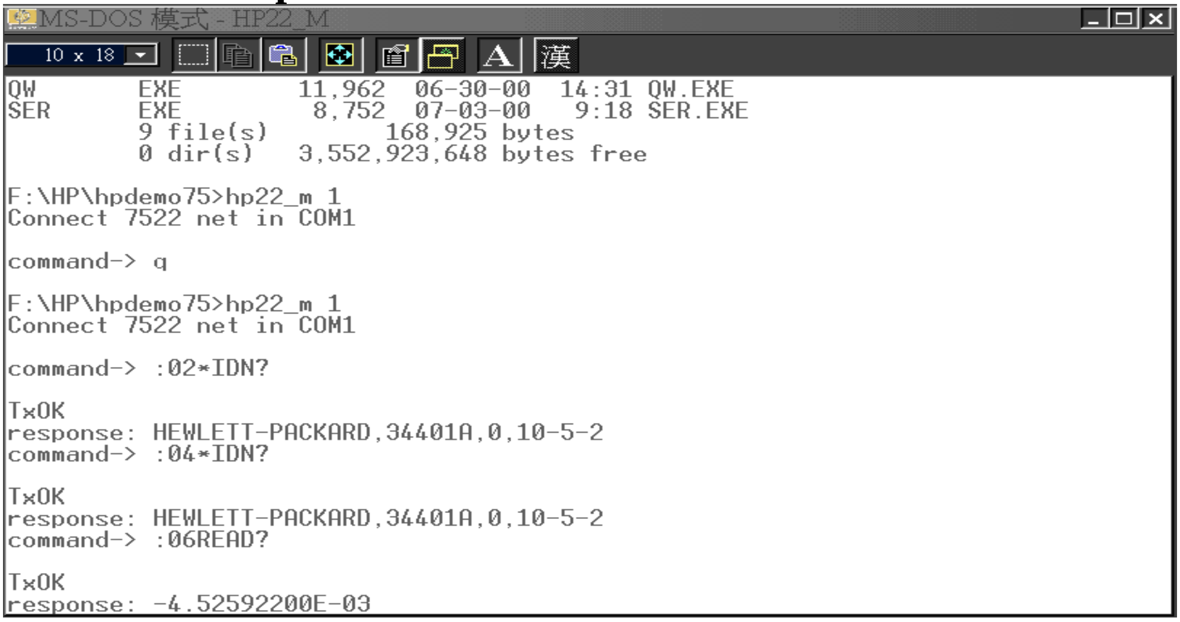
I-7522 NO.	I-7522 Address
#1	01h
#3	03h
#5	05h

HP NO.	HP34401A Address
#2	02h
#4	04h
#6	06h

- **A flow chart illustrating the functionality of the hp22_m.c file is as follows:**


start
Set the multimeters to remote-control mode
Any key pressed?
Y
N
key 'q'or 'Q' pressed?
Y
N
Set the multimeter back to local mode
end
Request multimeter command
Send the command to multimeter
Receive a response from the multimeter

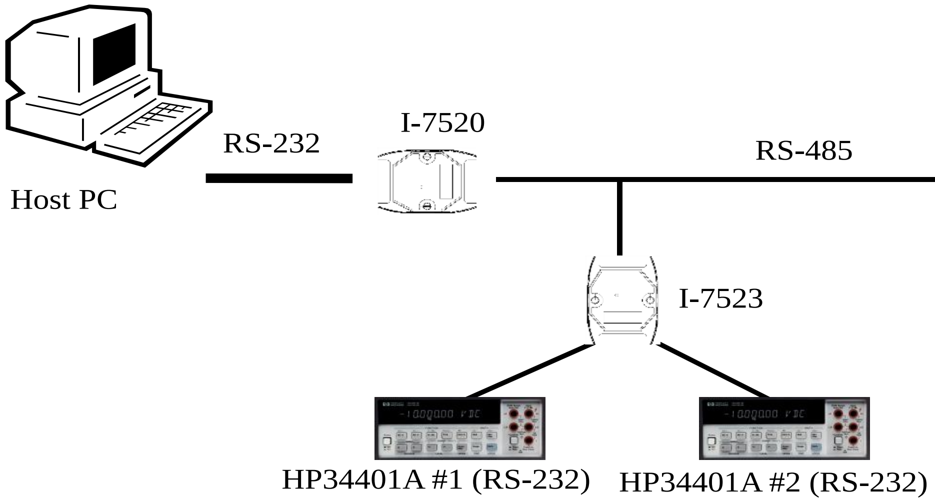
- **Execution examples:**

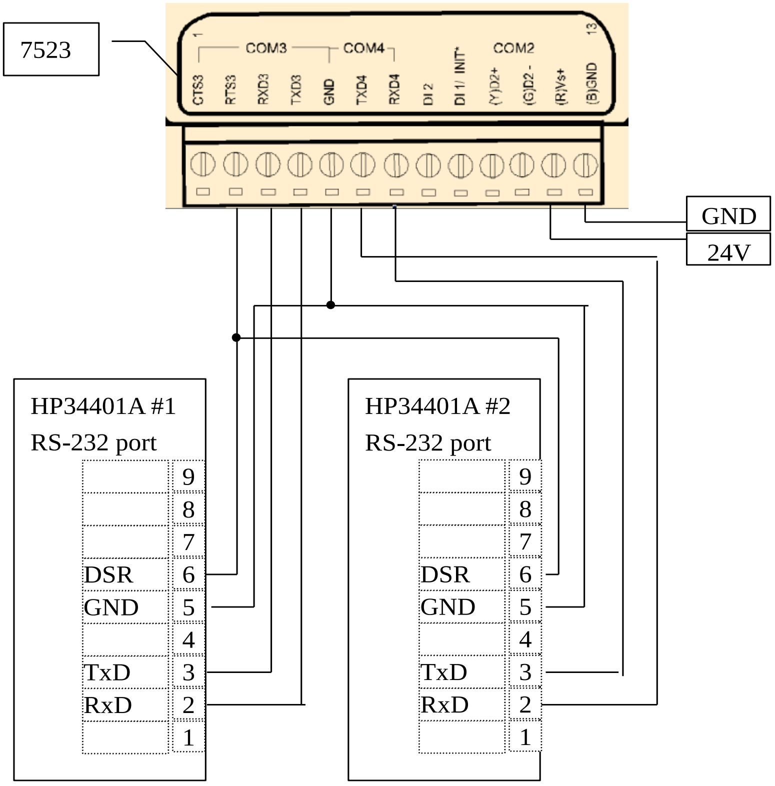


```
QW       EXE        11,962  06-30-00  14:31 QW.EXE
SER      EXE         8,752  07-03-00   9:18 SER.EXE
         9 file(s)        168,925 bytes
         0 dir(s)   3,552,923,648 bytes free

F:\HP\hpdemo75>hp22_m 1
Connect 7522 net in COM1

command-> q

F:\HP\hpdemo75>hp22_m 1
Connect 7522 net in COM1

command-> :02*IDN?

TxOK
response: HEWLETT-PACKARD,34401A,0,10-5-2
command-> :04*IDN?

TxOK
response: HEWLETT-PACKARD,34401A,0,10-5-2
command-> :06READ?

TxOK
response: -4.52592200E-03
```

6.1.5 Connecting a single I-7523 modules and two HP34401A

The following diagram below shows how to connect a single PC connected to two remote HP34401As on an RS-485 network. The I-7520 module is used to convert the RS-232 signal from to PC to an RS-485 signal. The I-7523 module is used as an “Addressable RS-232 converter” for the HP34401A, since there is no address setting on theHP34401A. Two HP34401A can be connected to a single I-7523 module.





The demo program, hp23_1.c, is designed to allow a Host PC to be linked to a remote HP34401A. Refer to the companion CD for the source code of the hp23_1.c file. The key features of hp23_1.c are as follows:

- **The RTS3 pin of COM3 port on each module must first be set as active HIGH to enable the HP34401A.**
- **The configuration of the COM ports on the I-7523 module is as follows:**

	COM2 (485)	COM3 (232)	COM4 (232)
Baud Rate	9600 (default)	9600 (default)	9600 (default)
Parity	None (default)	Even	Even
Data	8 (default)	7	7
Stop-bit	1 (default)	2	2

- **The address mapping for this configuration is given as follows:**

I-7523 Address	Corresponding COM3 address	Corresponding COM4 address
01h	02h	03h

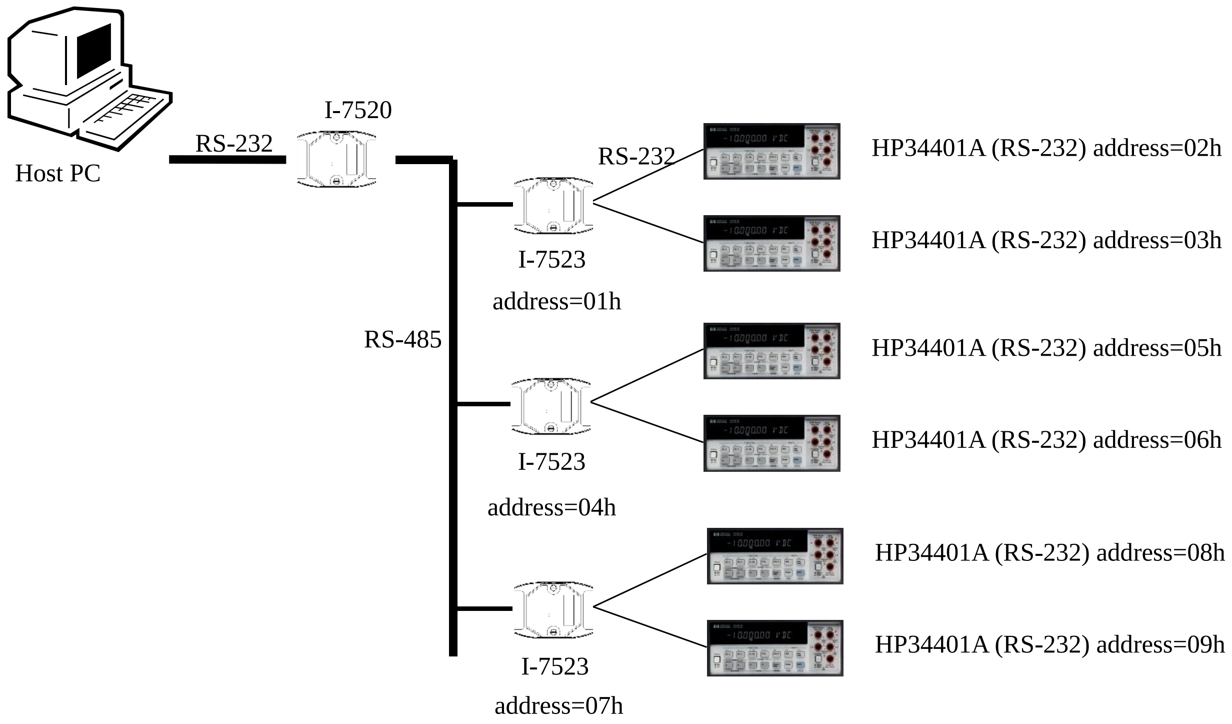
- **The InitHP() and CloseHP() function of hp23_1.c are as follows:**

```
 void InitHP(int ComNo)
 {
  com[ComNo].ComNo = ComNo;
  com[ComNo].BaudRate = 9600L;
  com[ComNo].DataFormat = Data8bit + NonParity + Stop1bit;
  com[ComNo].CheckSum = CHKSUMdisable;
  OpenCOM(ComNo);

  HPSendCommand(ComNo, ":02SYST:REM");            // #1 HP
  HPSendCommand(ComNo, ":02*CLS");               // #1 HP
  HPSendCommand(ComNo, ":03SYST:REM");            // #2 HP
  HPSendCommand(ComNo, ":03*CLS");               // #2 HP
  WaitClock(18);
}

void CloseHP(int ComNo)
{
  HPSendCommand(ComNo, ":02*CLS");               // #1 HP
  HPSendCommand(ComNo, ":02SYST:LOC");            // #1 HP
  HPSendCommand(ComNo, ":03*CLS");               // #2 HP
  HPSendCommand(ComNo, ":03SYST:LOC");            // #2 HP
  CloseCOM(ComNo);
}
```

6.1.6 Connecting multiple I-7523 modules and multiple HP34401A

The diagram below shows how to connect a single PC to multiple remote HP34401As on an RS-485 network. The I-7520 module is used to convert the RS-232 signal from to PC to an RS-485 signal. The I-7523 module is used as an "Addressable RS-232 converter" for the HP34401A, since there is no address setting on the HP34401A. Two HP34401As can be connected to a single I-7523 module.

Each I-7523 module has a unique address on the RS-485 network. Each HP34401A shares the same address range as its corresponding I-7523 module, which allows each HP34401A to have a unique address in this configuration.



Host PC
RS-232
I-7520
RS-485
RS-232
I-7523
address=01h
I-7523
address=04h
I-7523
address=07h
HP34401A (RS-232) address=02h
HP34401A (RS-232) address=03h
HP34401A (RS-232) address=05h
HP34401A (RS-232) address=06h
HP34401A (RS-232) address=08h
HP34401A (RS-232) address=09h

The demo program, hp23_M.c, is designed to allow a Host PC to be linked to remote HP34401A. Refer to the companion CD for the source code of hp23_M.c file. The key features of hp23_M.c are as follows:

- **The RTS3 pin of the COM3 port on each module must first be set as active HIGH to enable the HP34401A.**

- **The configuration of the COM ports on each I-7523 module is as follows:**

	COM2 (485)	COM3 (232)	COM4 (232)
Baud Rate	9600 (default)	9600 (default)	9600 (default)
Parity	None (default)	Even	Even
Data	8 (default)	7	7
Stop-bit	1 (default)	2	2

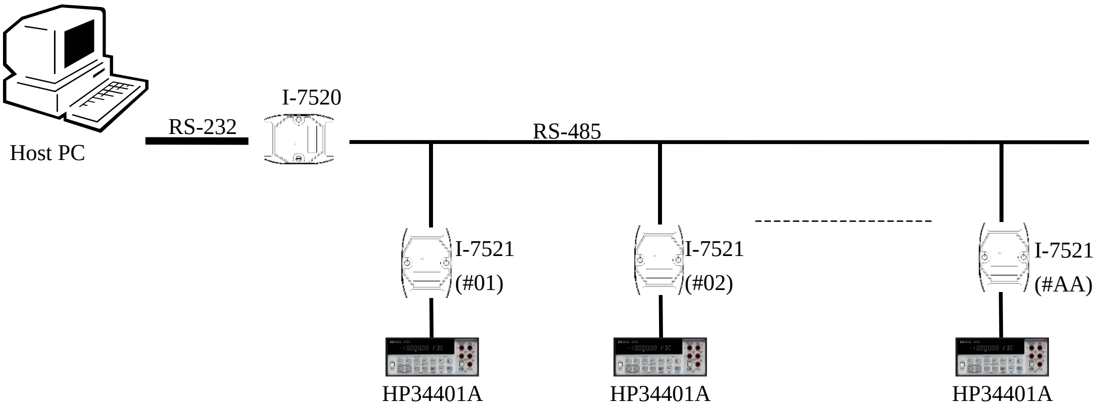
- **The InitHP() and CloseHP() functions of hp23_M.c are given as follows:**

```
void InitHP(int ComNo)
{
  com[ComNo].ComNo = ComNo;
  com[ComNo].BaudRate = 9600L;
  com[ComNo].DataFormat = Data8bit + NonParity + Stop1bit;
  com[ComNo].CheckSum = CHKSUMdisable;
  OpenCOM(ComNo);

  HPSendCommand(ComNo, ":02SYST:REM");             // #1 HP
  HPSendCommand(ComNo, ":02*CLS");                // #1 HP
  HPSendCommand(ComNo, ":03SYST:REM");             // #2 HP
  HPSendCommand(ComNo, ":03*CLS ");               // #2 HP
  HPSendCommand(ComNo, ":05SYST:REM");             // #3 HP
  HPSendCommand(ComNo, ":05*CLS ");               // #3 HP
  HPSendCommand(ComNo, ":06SYST:REM");             // #4 HP
  HPSendCommand(ComNo, ":06*CLS ");               // #4 HP
  HPSendCommand(ComNo, ":08SYST:REM");             // #5 HP
  HPSendCommand(ComNo, ":08*CLS ");               // #5 HP
  HPSendCommand(ComNo, ":09SYST:REM");             // #6 HP
  HPSendCommand(ComNo, ":09*CLS ");               // #6 HP
  WaitClock(18);
}
void CloseHP(int ComNo)
{
  HPSendCommand(ComNo, ":02*CLS");                // #1 HP
  HPSendCommand(ComNo, ":02SYST:LOC");             // #1 HP
  HPSendCommand(ComNo, ":03*CLS");                // #2 HP
  HPSendCommand(ComNo, ":03SYST:LOC");             // #2 HP
  HPSendCommand(ComNo, ":05*CLS");                // #3 HP
  HPSendCommand(ComNo, ":05SYST:LOC");             // #3 HP
  HPSendCommand(ComNo, ":06*CLS");                // #4 HP
  HPSendCommand(ComNo, ":06SYST:LOC");             // #4 HP
  HPSendCommand(ComNo, ":08*CLS");                // #5 HP
  HPSendCommand(ComNo, ":08SYST:LOC");             // #5 HP
  HPSendCommand(ComNo, ":09*CLS");                // #6 HP
  HPSendCommand(ComNo, ":09SYST:LOC");             // #6 HP
  CloseCOM(ComNo);
}
```

6.2. Typical Applications

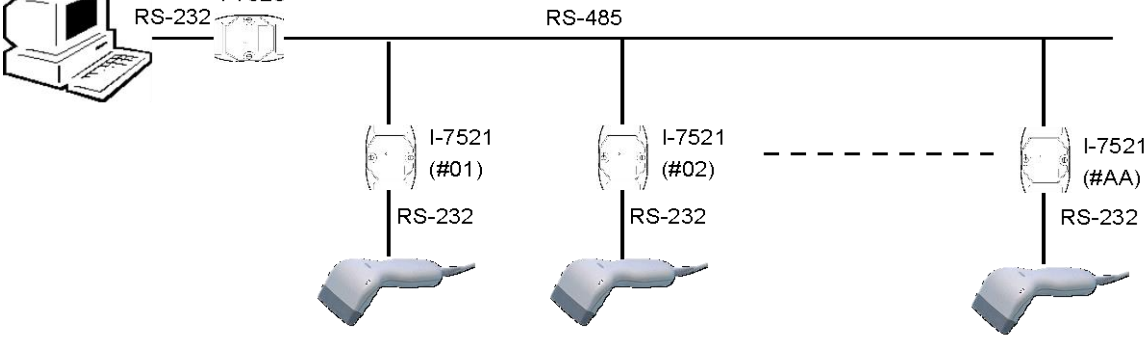
Application 1: Addressable RS-232 Controller (Command Type)

- **Each I-7521 module has a unique address**
- **The Host PC first sends a command to all I-7521 module**
- **The destination I-7521 module will pass the command to its local RS-232 device**
- **The destination I-7521 module will then send the response from the RS-232 device back to the Host PC**
- **Refer to the 7521.c file for the source code of the firmware**



Application 2: Addressable RS-232 Controller (Receive Data only Type)-Barcode Reader1

- **The barcode-reader can scan a barcode at anytime, and the I-7521 module will store these barcodes in an internal buffer (1K bytes)**
- **The Host PC first sends $AAU command to all I-7521 modules. The destination I-7521 module will check its internal buffer. If there are any barcodes in the buffer, the I-7521 module will then send a single barcode back to the Host PC.**
- **The Host PC can send more $AAU commands to read each barcode stored in the internal buffer of the I-7521 module**
- **Refer to the 7521.c file for the source code of the firmware**



Application 3: Bypass data to COM2-Barcode Reader2

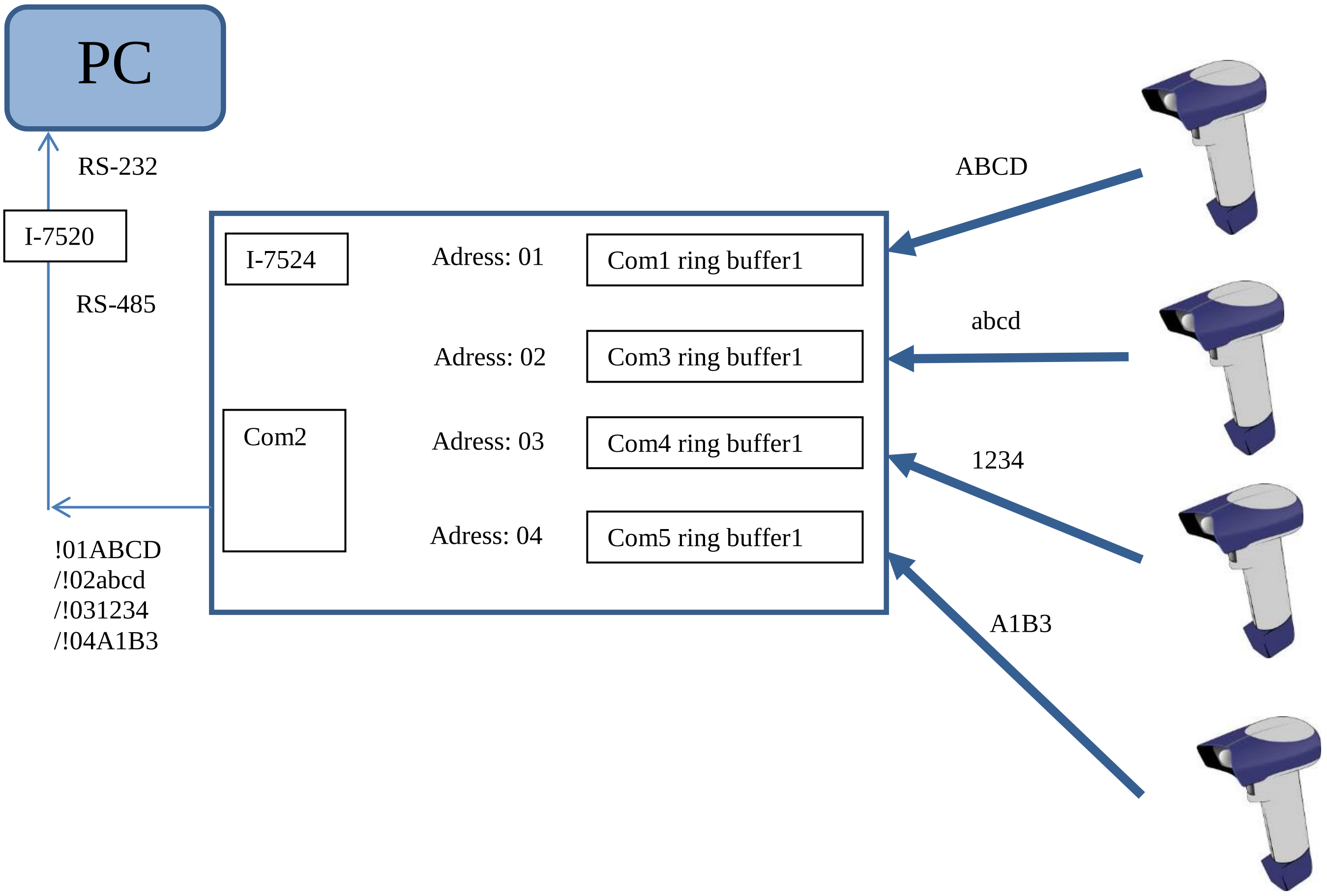
- **I-7524 module address is 01,02,03,04**
- **PC sends “$01E1”, “$02E1”, “$03E1”, “$04E1” to enable “prefixed address byte” function.**
- **PC sends “$01H1”, “$02H1”, “$03H1”, “$04H1” to enable “bypass to COM2” function.**
- **The barcode sends data to I-7524. I-7524 will directly bypass the data to COM2 right now.**

  For example:

  (1)The barcode sends “ABCD” to COM1.

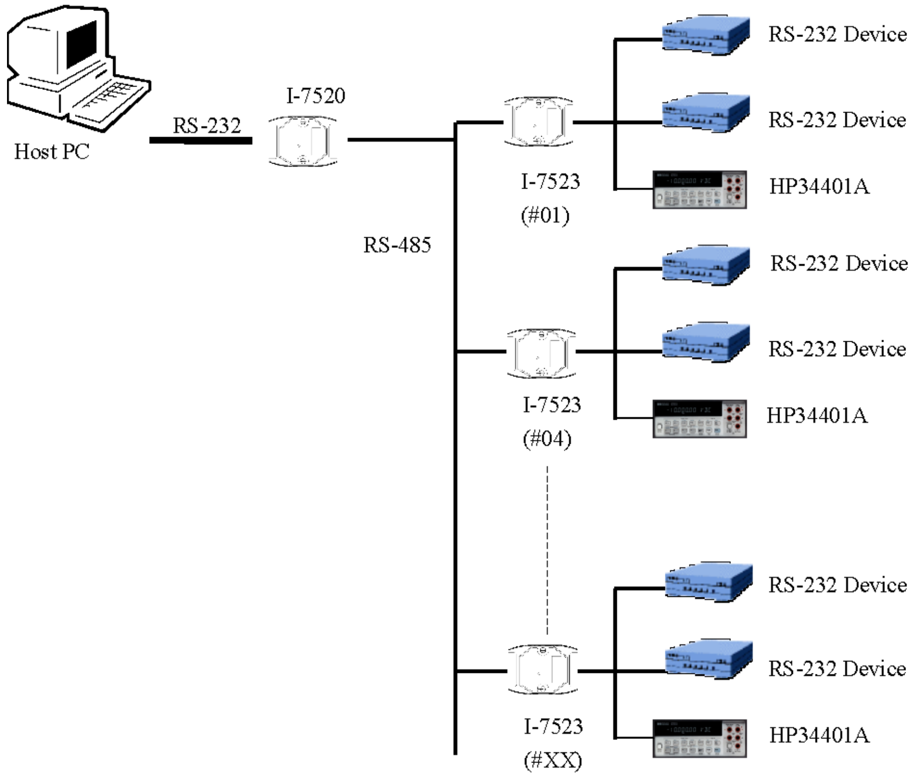
  (2)I-7524 sends “ABCD” to COM2 and adds prefixed address “!01” directly.

  (3)PC receives “!01ABCD” data.



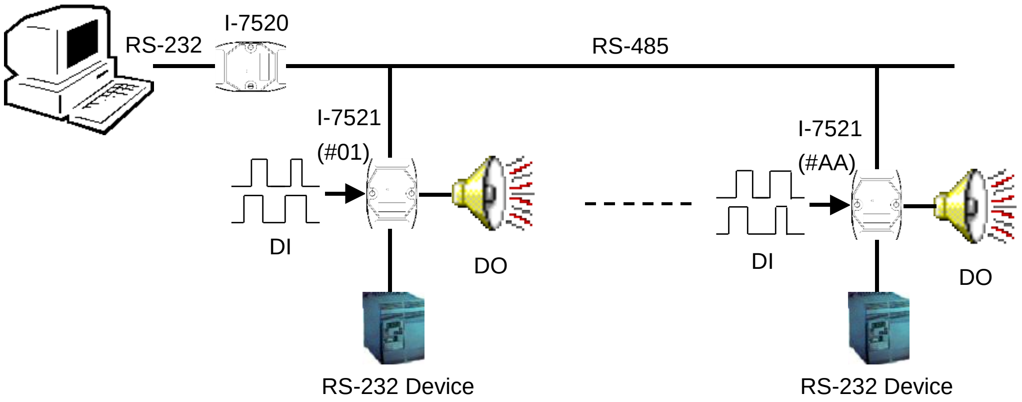
Application 4: Addressable RS-232 Controller (Three channels)

- **Each I-7523 module has a unique address**
- **Each I-7523 module can support three RS-232 devices, AA, AA+1 and AA+2**
- **The Host PC first sends a command to all I-7523 module**
- **The destination I-7523 module will pass the command to its local RS-232 device 1, RS-232 device 2 or RS-232 device 3.**
- **The I-7523 module then sends the response from the RS-232 device back to the Host PC**
- **COM4 port of the I-7523 module can support either 1 or 2 stop bits, so it can support a two stop-bit device, such as the HP34401A.**
- **The RS-232 device can be used for command (Application 1) or null command (Application 2) type controller applications**
- **Refer to the 7523.c file for the source code of the firmware**



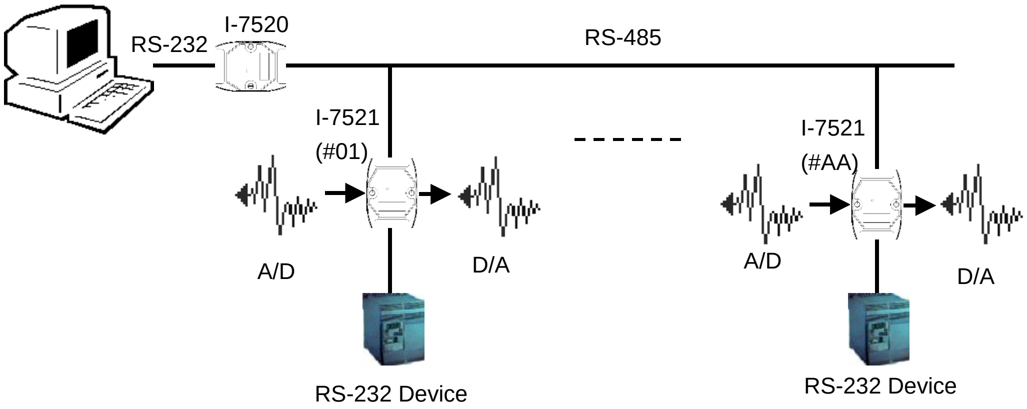
Application 5: Real time DI Monitoring and DO Alarm (Master type)

- **Refer to Applications 1 and 2 for more information**
- **The I-7521 module will scan and analyze the onboard DI. If the DI shows a match with the alarm states, the onboard DO will trigger the alarm device to allow for alarm or safety control**
- **All DI and DO control operations are performed in the I-7521 module. The Host PC only reads the DI and DO values as part of its system monitoring function**
- **Refer to the 7521ODM1.c file for the source code of the firmware**



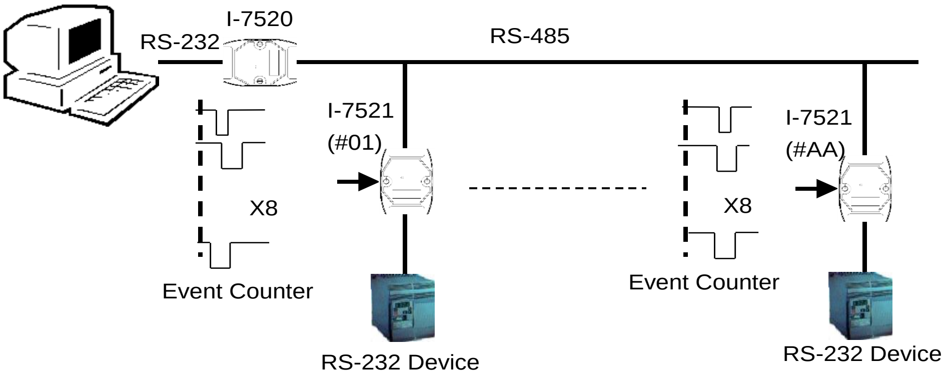
Application 6: Real time A/D Monitoring and D/A Control (Master type)

- **Refer to Applications 1 and 2 for more information**
- **An X301 daughter board supports a single A/D and a single D/A and can be connected the I-7521 module. The I-7521+X301 combination can read and analyze the A/D in real time. The D/A output is controlled based on the A/D value**
- **All A/D and D/A control operations are performed in the I-7521 module. The Host PC only reads the A/D and D/A values as part of its system monitoring functions**
- **Refer to the 7521ODM2.c file for the source code of the firmware**



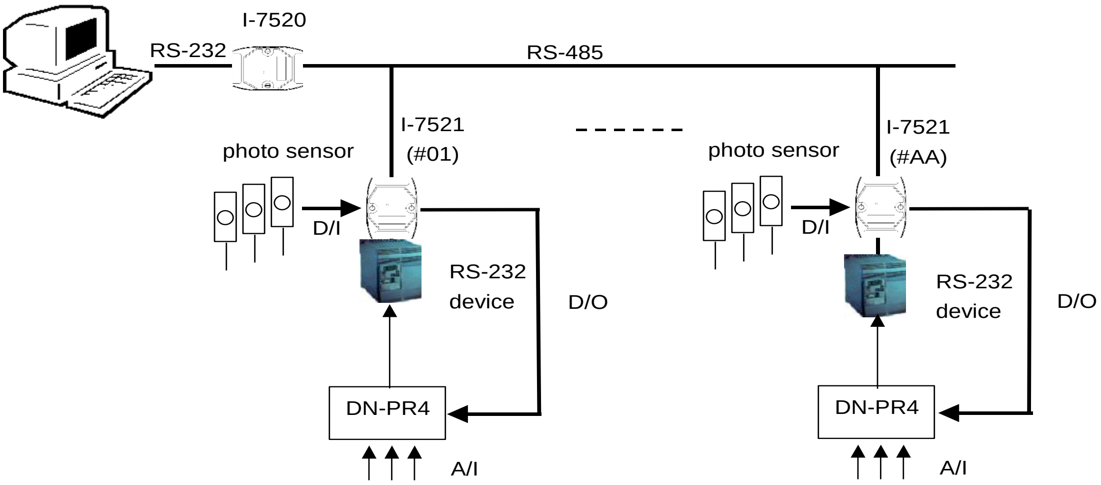
Application 7: 8-channel long term event counters (Master type)

- **Refer to Applications 1 and 2 for more information**
- **The X100 daughter supports 8 DI channels. The I-7521+X100 combination can read and analyze the 8 event counters in real time. The timing diagram of the event-counter will be latched until a clear command is sent by the Host PC**
- **All analysis operations are performed in the I-7521 module. The Host PC only reads the timing values of the event counter as past of its system monitoring functions**
- **Refer to the 7521ODM3.c file for the source code of the firmware**



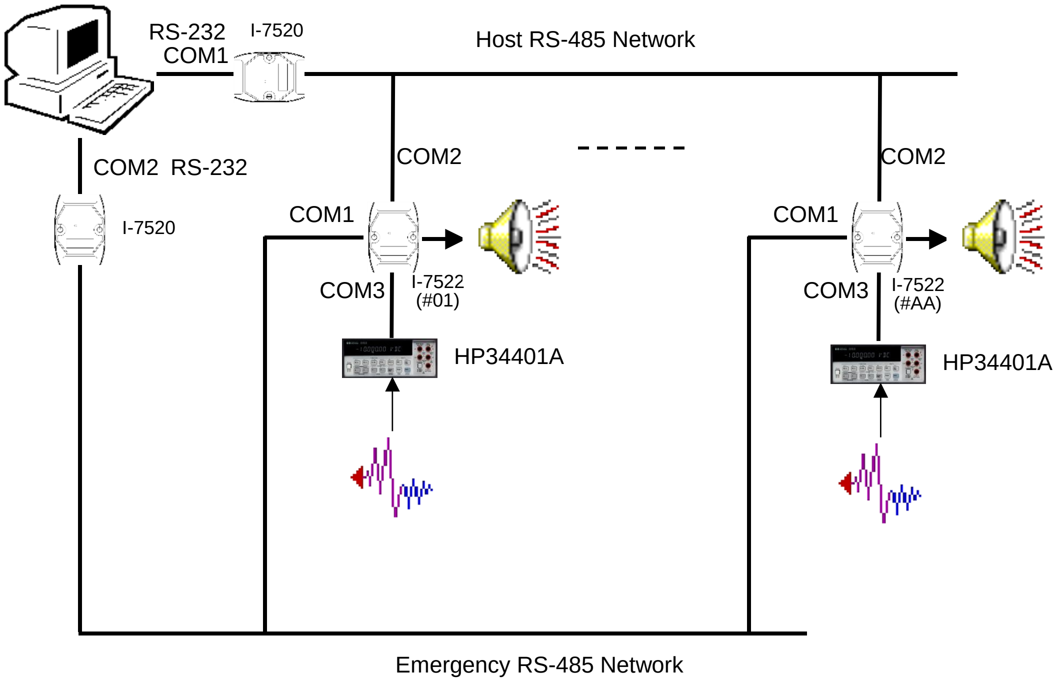
Application 8: Multiplex Control (Master type)

- **Refer to Applications 1 and 2 for more information**
- **The onboard DO of the I-7521 module can directly drive a relay. The onboard DI can be linked to photo sensors to allow for event triggering. The I-7521+DN-PR4 can be triggered by a photo-sensor and can control the multiplex to select the expected analog signal**
- **All control operations are performed in the I-7521 module. The Host PC is able to read the 3-channel A/D signals without requiring the multiplex control**
- **Refer to the 7521ODM4.c file for the source code of the firmware**



Application 9: Real time Analog Signal Monitoring and Alarm Control (Master type)

- **Refer to Applications 1 and 2 for more information**
- **COM1 of the Host PC is used as the Host RS-485 network. The Host PC will send commands and receive responses through the RS-485 network**
- **COM2 of the Host PC is used as an emergency RS-485 network. All I-7522 modules will automatically monitor the analog signal connected to an HP34401A. If an emergency event occurs, the I-7522 module will send an emergency command to this RS485 network. If multiple I-7522 modules send emergency commands to the Host PC at the same time, the I-7522 modules will continue to send emergency commands to the Host PC until confirmation is received from the Host PC**
- **All analysis operations are performed in the I-7522 module. The Host PC only reads the analog values as part of its system monitoring function**
- **Refer to the 7522ODM5.c file for the source code of the firmware**



Appendix A MiniOS7 Utility

The **MiniOS7 Utility** program provides three main functions:

- Allows the MiniOS7 image to be upgraded
- Allows program files to be downloaded to the Flash Memory
- Allows the COM port settings to be configured

Location of the MiniOS7 utility:
The MiniOS7 utility is located in:
CD: \NAPDOS\MINIOS7\UTILITY \MiniOS7_utility\
minios7_utility_v311.exe
or http://ftp.icpdas.com.tw/pub/cd/8000cd/napdos/minios7/utility
/minios7_utility/minios7_utility_v311.exe

Appendix B 7188XW.EXE

The 7188xw.exe file is the main utility for I-752N(D) modules; 7188xw.exe can be used as follows:

- **Download the user program files from the Host PC to the memory unit of the I-752N(D) module**.
- **Download the MiniOS7 image file from the Host PC to the Flash Memory of the I-752N(D) module and upgrade MiniOS7.**
- **Enter data into the I-752N(D) module using the keyboard of the Host PC**
- **Show the data from the I-752N(D) modules.**

Location of the 7188xw.exe file

CD:\Napdos\MiniOS7\utility\
or http://ftp.icpdas.com.tw/pub/cd/8000cd/napdos/minios7/utility/

For more information about 7188xw, refer to the

CD:\Napdos\MiniOS7\document\Lib_Manual_For_7188XABC\index.htm
or http://ftp.icpdas.com.tw/pub/cd/8000cd/napdos/minios7/document/lib_manual_for_7188e/eng/index.htm

Downloading files to the I-752N(D) modules

Step 1: Connect the I-752N module to the Host PC (Refer to Sec. 1.6.1).

Step 2: Connect the INT* pin to the GND pin to disable autoexec.bat before turnning on the power.

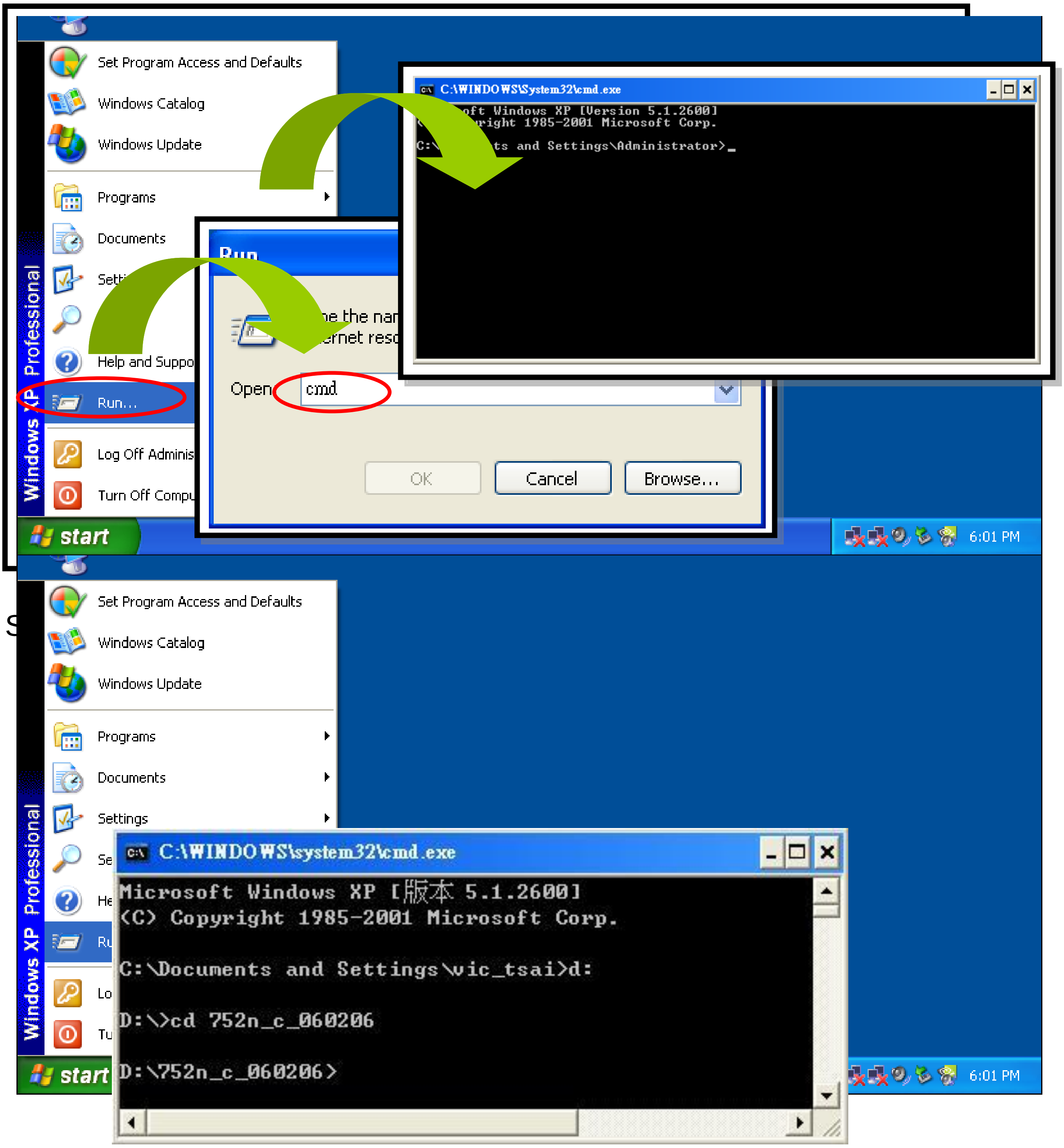
Step 3: Build a folder that includes the firmware.
There are two ways to use 7188xw.exe:

a) Put the firmware (Refer to Sec. 2.2) and the 7188xw.exe file in the same folder on the Host PC. Execute 7188xw.exe and jump to Step 8.
b) Go to Step 4

(Refer to the companion CD: Napdos\752N\Firmware_V3 or the release notes to find details related to the firmware.)

Step 4: Copy the 7188xw.exe file to the PATH directory, for example C:\DOS or C:\WINDOWS, and then it will be able to be executed to allow files to be downloaded from any location

Step 5: Execute MS-DOS in Windows as follows:



```
oft Windows XP [Version 5.1.2600]
right 1985-2001 Microsoft Corp.

C:\        ts and Settings\Administrator>_
```

```
Microsoft Windows XP [版本 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\vic_tsai>d:

D:\>cd 752n_c_060206

D:\752n_c_060206>
```

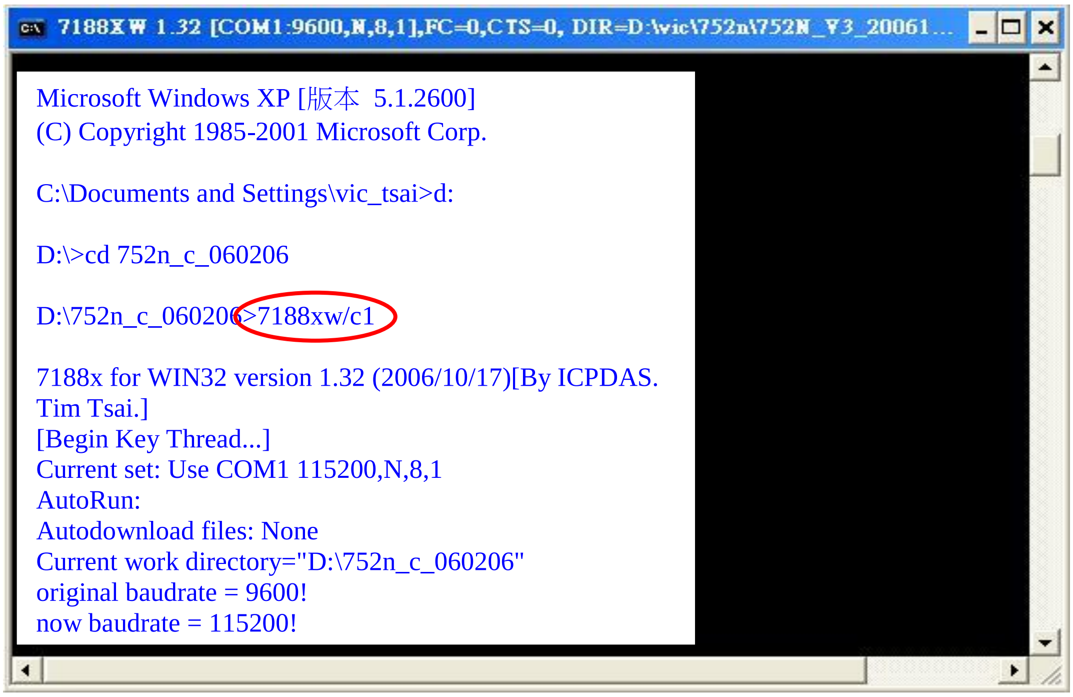
Step 7: Select the active COM Port of the Host PC and execute 7188xw.exe

If the I-752N module is connected to COM1 on the PC, then type **7188xw/c1**

If the I-752N module is connected to COM2 on the PC, then type **7188xw/c2**

Press **ENTER**

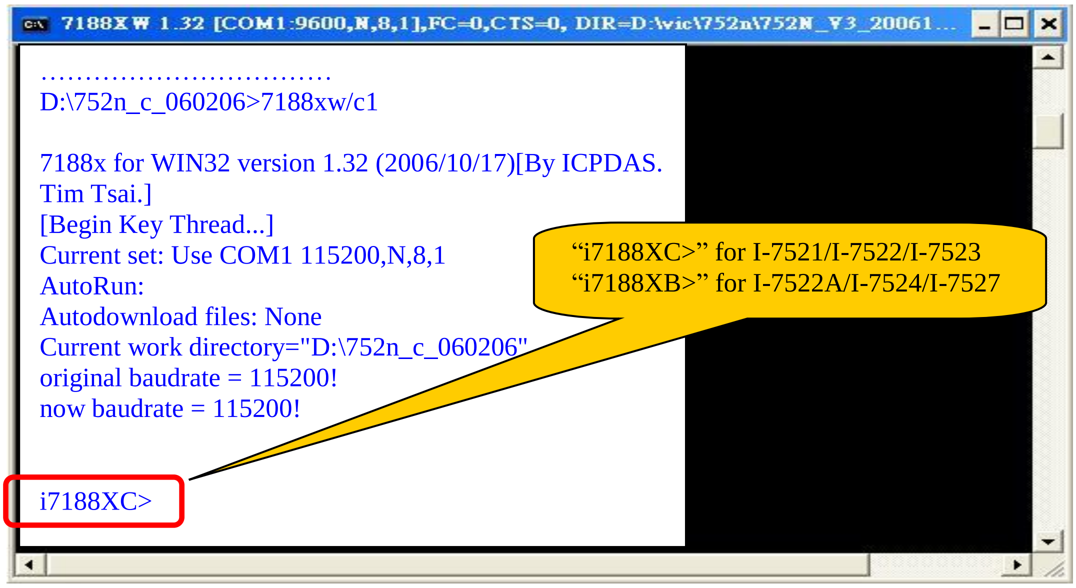
The following screen will be shown on the Host PC for I-7521/22/23:


7188XW 1.32 [COM1:9600,N,8,1],FC=0,CTS=0, DIR=D:\vic\752n\752N_V3_20061...

```
Microsoft Windows XP [版本 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\vic_tsai>d:

D:\>cd 752n_c_060206

D:\752n_c_060206>7188xw/c1

7188x for WIN32 version 1.32 (2006/10/17)[By ICPDAS.
Tim Tsai.]
[Begin Key Thread...]
Current set: Use COM1 115200,N,8,1
AutoRun:
Autodownload files: None
Current work directory="D:\752n_c_060206"
original baudrate = 9600!
now baudrate = 115200!
```

Step 8: Refer to Steps 6 to 9 of Sec 3.1 to change the configuration for 7188xw.exe to 115200, N, 8,1

Press **ENTER**


7188XW 1.32 [COM1:9600,N,8,1],FC=0,CTS=0, DIR=D:\vic\752n\752N_V3_20061...

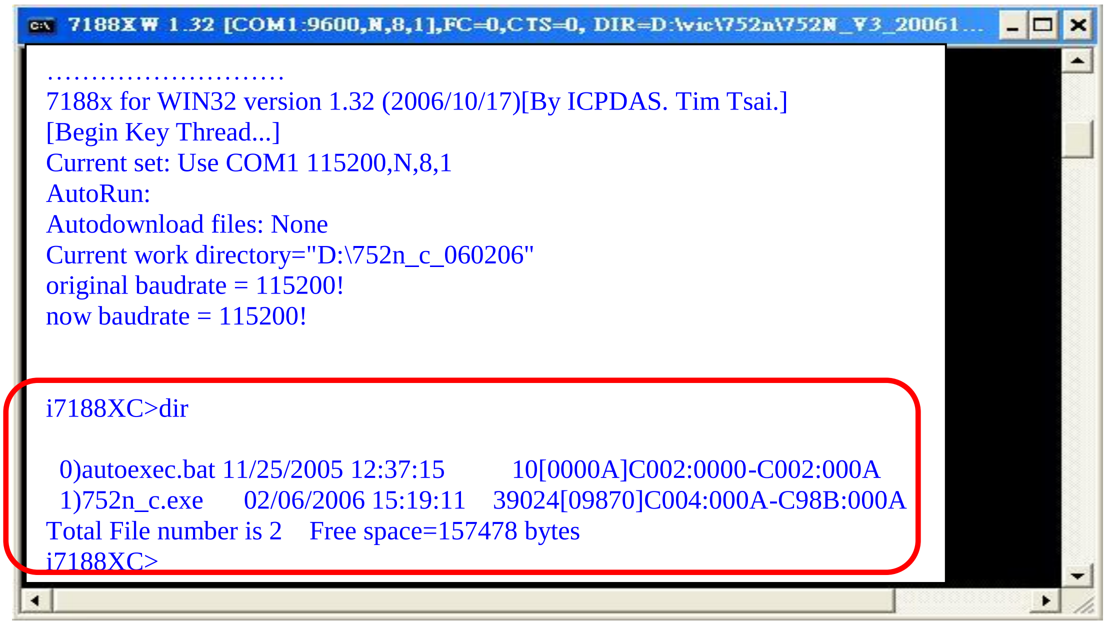
```
...............................
D:\752n_c_060206>7188xw/c1

7188x for WIN32 version 1.32 (2006/10/17)[By ICPDAS.
Tim Tsai.]
[Begin Key Thread...]
Current set: Use COM1 115200,N,8,1
AutoRun:
Autodownload files: None
Current work directory="D:\752n_c_060206"
original baudrate = 115200!
now baudrate = 115200!


i7188XC>
```

"i7188XC>" for I-7521/I-7522/I-7523
"i7188XB>" for I-7522A/I-7524/I-7527

Step 9: Type **dir**

Press **ENTER**

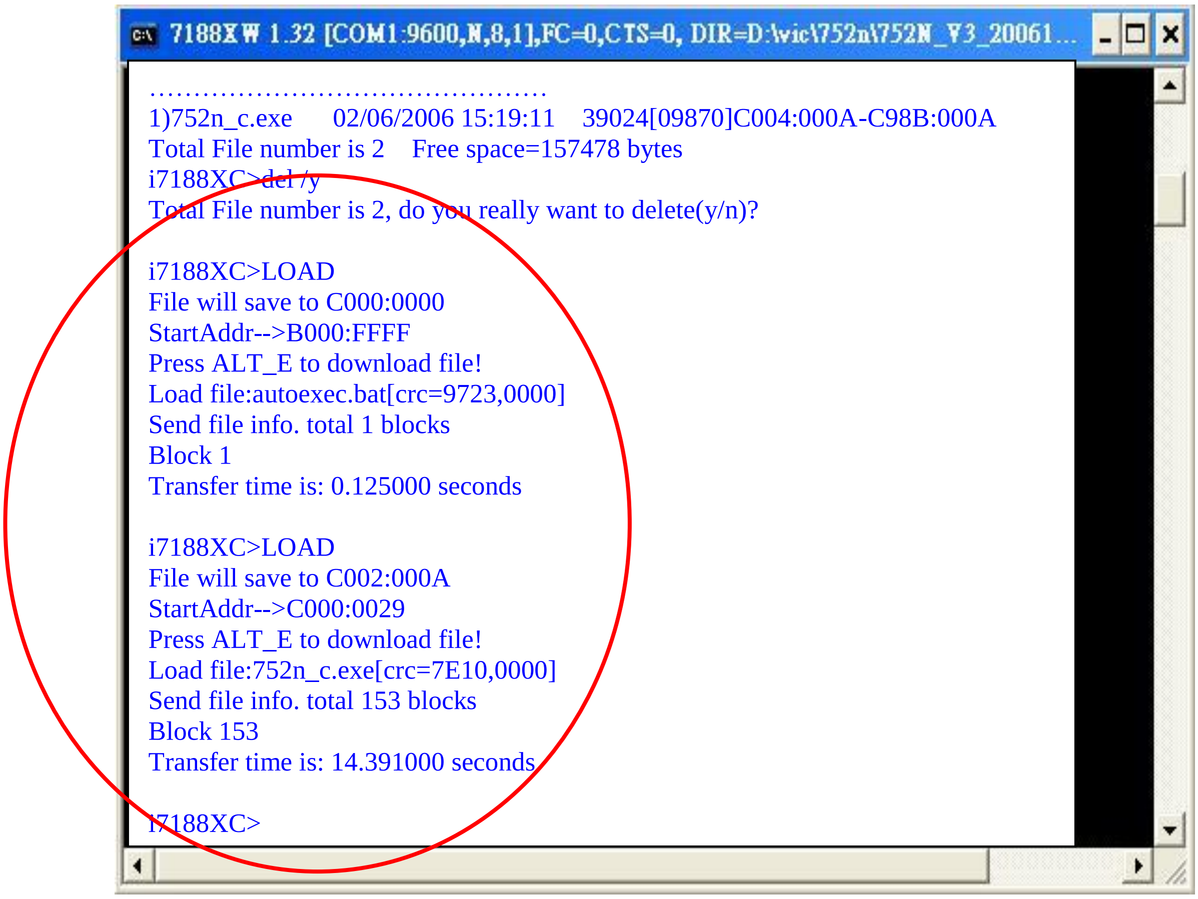
The following screen will be shown on the Host PC for I-7521/22/23:


7188XW 1.32 [COM1:9600,N,8,1],FC=0,CTS=0, DIR=D:\vic\752n\752N_V3_20061...

```
........................
7188x for WIN32 version 1.32 (2006/10/17)[By ICPDAS. Tim Tsai.]
[Begin Key Thread...]
Current set: Use COM1 115200,N,8,1
AutoRun:
Autodownload files: None
Current work directory="D:\752n_c_060206"
original baudrate = 115200!
now baudrate = 115200!

i7188XC>dir

  0)autoexec.bat 11/25/2005 12:37:15         10[0000A]C002:0000-C002:000A
  1)752n_c.exe      02/06/2006 15:19:11    39024[09870]C004:000A-C98B:000A
Total File number is 2     Free space=157478 bytes
i7188XC>
```

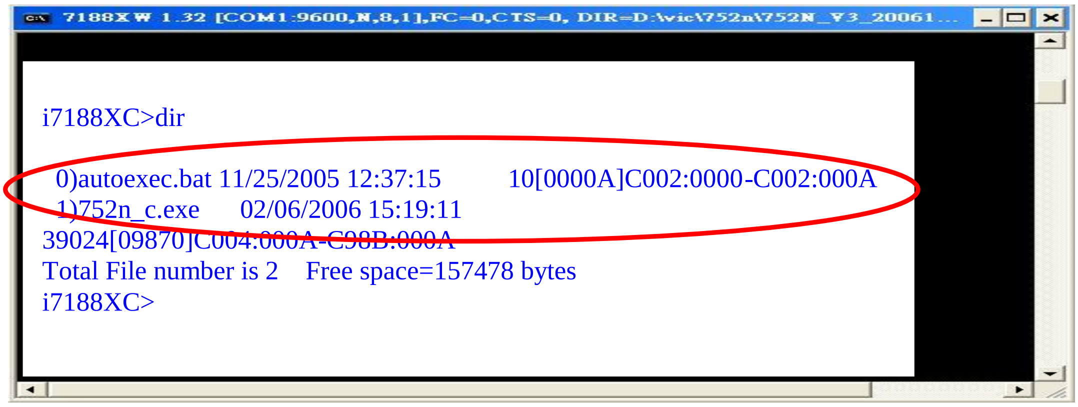
Step 10: Press **F4** to automatically download the files (be sure that the file 7188XW.F4 and the firmware are put together).

The 7188xw will then download 752n_c.exe and autoexec.bat from the Host PC to the module. After the download operation has been completed, the following screen will be shown:


7188XW 1.32 [COM1:9600,N,8,1],FC=0,CTS=0, DIR=D:\vic\752n\752N_V3_20061...

```
..........................................
1)752n_c.exe      02/06/2006 15:19:11    39024[09870]C004:000A-C98B:000A
Total File number is 2     Free space=157478 bytes
i7188XC>del /y
Total File number is 2, do you really want to delete(y/n)?

i7188XC>LOAD
File will save to C000:0000
StartAddr-->B000:FFFF
Press ALT_E to download file!
Load file:autoexec.bat[crc=9723,0000]
Send file info. total 1 blocks
Block 1
Transfer time is: 0.125000 seconds

i7188XC>LOAD
File will save to C002:000A
StartAddr-->C000:0029
Press ALT_E to download file!
Load file:752n_c.exe[crc=7E10,0000]
Send file info. total 153 blocks
Block 153
Transfer time is: 14.391000 seconds

i7188XC>
```

Step 11: Type **dir** and press **Enter** to see the downloaded files.
(752n_c.exe is for I-7521/22/23, 752n_b.exe is for I-7524/22A/27)


7188XW 1.32 [COM1:9600,N,8,1],FC=0,CTS=0, DIR=D:\vic\752n\752N_V3_20061...

```
i7188XC>dir

  0)autoexec.bat 11/25/2005 12:37:15        10[0000A]C002:0000-C002:000A
  1)752n_c.exe      02/06/2006 15:19:11
39024[09870]C004:000A-C98B:000A
Total File number is 2    Free space=157478 bytes
i7188XC>
```

Step 12: Disconnect the INIT* pin from the GND and power-off then power-on the I-7521/22/23/22A/24/27 module. The MiniOS7 will automatically execute the new firmware.

Appendix C Firmware Version information

Download a description document about firmware version information: http://ftp.icpdas.com/pub/cd/8000cd/napdos/752n/ → firmware_v3 folder → readme.txt.

The readme.txt illustrates some difference and improvement of firmware version.