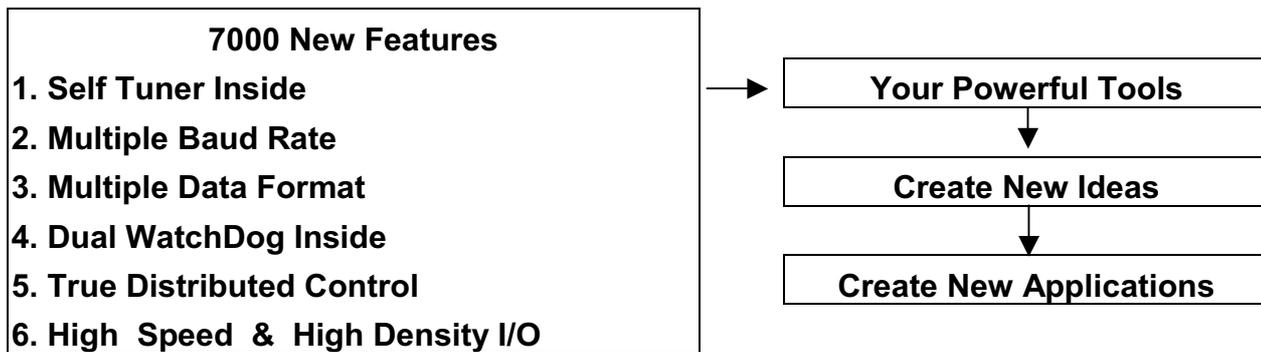


# 7188(D) Series

## 7188(D) 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 consequent to the use of this product. ICP DAS reserves the right to change this manual at any time without notice. The information furnished by ICP DAS is believed to be accurate and reliable. However, no responsibility is assumed by ICP DAS for its use, nor for any infringements of patents or other rights of third parties resulting from its use.

### Copyright

Copyright 2001 by ICP DAS. All rights are reserved.

### Trademark

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

## Table of Contents

<b>1.</b>	<b>INTRODUCTION.....</b>	<b>4</b>
1.1	QUICK START .....	6
1.1.1	<i>Software Installation</i> .....	6
1.1.2	<i>Connect Download Cable to Host-PC</i> .....	9
1.1.3	<i>Download Led.exe &amp; Autoexec.bat to 7188(D)</i> .....	10
1.1.4	<i>Download Hello.exe &amp; Autoexec.bat to 7188(D)</i> .....	15
1.1.5	<i>Demo Program of 7188(D)</i> .....	19
1.1.6	<i>More Demo Program of 7188(D)</i> .....	20
1.1.7	<i>More about MiniOS7</i> .....	23
1.1.8	<i>MiniOS7 Upgrade</i> .....	24
1.1.9	<i>7188X.EXE</i> .....	27
1.1.10	<i>Libraries</i> .....	29
1.1.11	<i>Compiler &amp; linker</i> .....	30
1.2	FEATURES.....	36
1.3	PIN ASSIGNMENT .....	37
1.4	SPECIFICATIONS.....	39
1.5	BLOCK DIAGRAM OF 7188.....	41
1.6	JUMPER SETTING .....	41
1.7	WIRING DIAGRAMS FOR APPLICATION .....	42
1.7.1	<i>Connect COM4 to PC</i> .....	42
1.7.2	<i>Connect COM3 to 3-wire RS-232 Device</i> .....	43
1.7.3	<i>Connect COM2 to 2-wire RS-485 Device</i> .....	44
1.7.4	<i>Connect COM2 to 7510 for Isolation</i> .....	45
1.7.5	<i>Connect COM1 to 9-wire Modem</i> .....	46
1.7.6	<i>Connect COM1 to two-wire RS-485 Device</i> .....	47
1.8	COMPARISON TABLE .....	48
<b>2.</b>	<b>OPERATION PRINCIPLE.....</b>	<b>49</b>
2.1	SYSTEM MAPPING .....	49
2.2	USE COM4 FOR DEBUG PROGRAM .....	51
2.3	USE COM4 TO DOWNLOAD PROGRAM.....	52
2.4	USE COM4 AS A COM PORT .....	52
2.5	USE COM3 FOR RS-232 APPLICATIONS.....	52
2.6	USE COM2 FOR RS-485 APPLICATIONS.....	53
2.7	USE COM1 FOR RS-232 APPLICATIONS.....	54

2.8	USE COM1 FOR RS-485 APPLICATIONS.....	54
2.9	COM PORT COMPARISON: 7188(D) & PC .....	55
2.10	HOW TO USE COM1/2/3/4 .....	56
2.11	USE RED LED & 7-SEG LED DISPLAY .....	59
2.12	HOW TO USE FLASH MEMORY .....	60
2.13	USE RTC & NVSRAM .....	62
2.14	USE EEPROM.....	63
2.15	USE WATCHDOG TIMER .....	64
<b>3.</b>	<b>APPLICATIONS .....</b>	<b>66</b>
3.1	EMBEDDED CONTROLLER.....	66
3.2	ADDRESSABLE RS-232 PROTOCOL CONVERTER .....	67
3.3	LOCAL REAL TIME CONTROLLER .....	68
3.4	REMOTE LOCAL CONTROLLER .....	69
3.5	I/O EXTENSION OF PLC APPLICATIONS .....	70
3.6	RADIO MODEM APPLICATIONS .....	71
3.7	USE ALL 4 COM PORT APPLICATION-1 .....	72
3.8	USE ALL 4 COM PORT APPLICATION-2 .....	73

# 1. Introduction

---

The 7188(D) family is a embedded controller designed for industry applications. It can be used to replace the PC or PLC for the harsh environment. The 7000 series is a family of network data acquisition and control modules. They provide A/D, D/A, DI/O, Timer/Counter and other functions. These modules can be remote controlled by a set of commands. The 7188(D) family can be ordered as following:

Order Item	Flash ROM size	5-digit LED	O. S.
<b>7188</b>	<b>512K Bytes</b>	<b>No</b>	<b>MiniOS7</b>
<b>7188D</b>	<b>512K Bytes</b>	<b>Yes</b>	<b>MiniOS7</b>

## What is DOS?

DOS is an acronym for Disk Operating System. Several brands of DOS' have been created by various companies. In all cases, DOS--whether PC-DOS, MS-DOS, or ROM-DOS--is a set of commands or code which tells the computer how to process information. DOS runs programs, manages files, controls information processing, directs input and output, and performs many other related functions.

## What is MiniOS7 ?

The MiniOS7 is a embedded O.S. designed for 7188(D)/7188X/7521/7188E/8000 series. It is developed by ICP DAS Co. Ltd. Several brands of DOS have been created by various companies. In all cases, DOS--whether PC-DOS, MS-DOS, or ROM-DOS--is a set of commands or code which tells the computer how to process information. DOS runs programs, manages files, controls information processing, directs input and output, and performs many other related functions. The MiniOS7 will provide equivalent functions of ROMDOS and provide more special functions. Refer to Sec. ??? for more information.

## More Information

Refer to “7000 Bus Converter User Manual” chapter 1 for more

information as following:

- 1.1 7000 Overview
- 1.2 7000 Related Documentation
- 1.3 7000 Common Features
- 1.4 7000 System Network Configuration
- 1.5 7000 Dimension

### **Package List**

In addition to this manual, the package includes the following items:

- One 7188(D) hardware module
- One companion CD for software drivers & user's manuals
- One "7188(D) user's manual" (this manual)
- One download cable, CA0910
- One release note

### **Accessory Order Information**

- CA0910: Download cable, 3-terminal to DB-9, for COM3, COM4 of 7188(D)
- CA0915: female DB-9 to male DB-9, for COM1 of 7188(D)

### **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.

## **1.1 Quick Start**

---

### **1.1.1 Software Installation**

Step 1: Execute the companion CD

Step 2: Click **Toolkis (Softwares) / Manuals**

Step 3: Click **7188(D) Embedded Controller**

Step 4: Click **Install 7188(D) Demo Program to C**  
or Click **Install 7188(D) Demo Program to D**

Then the installation program will create a DOS-BOX & make a directory in **C:\7188OS7** or **D:\7188OS7** & copy all demo program & user's manual to it.

Step 5: close this DOS-BOX

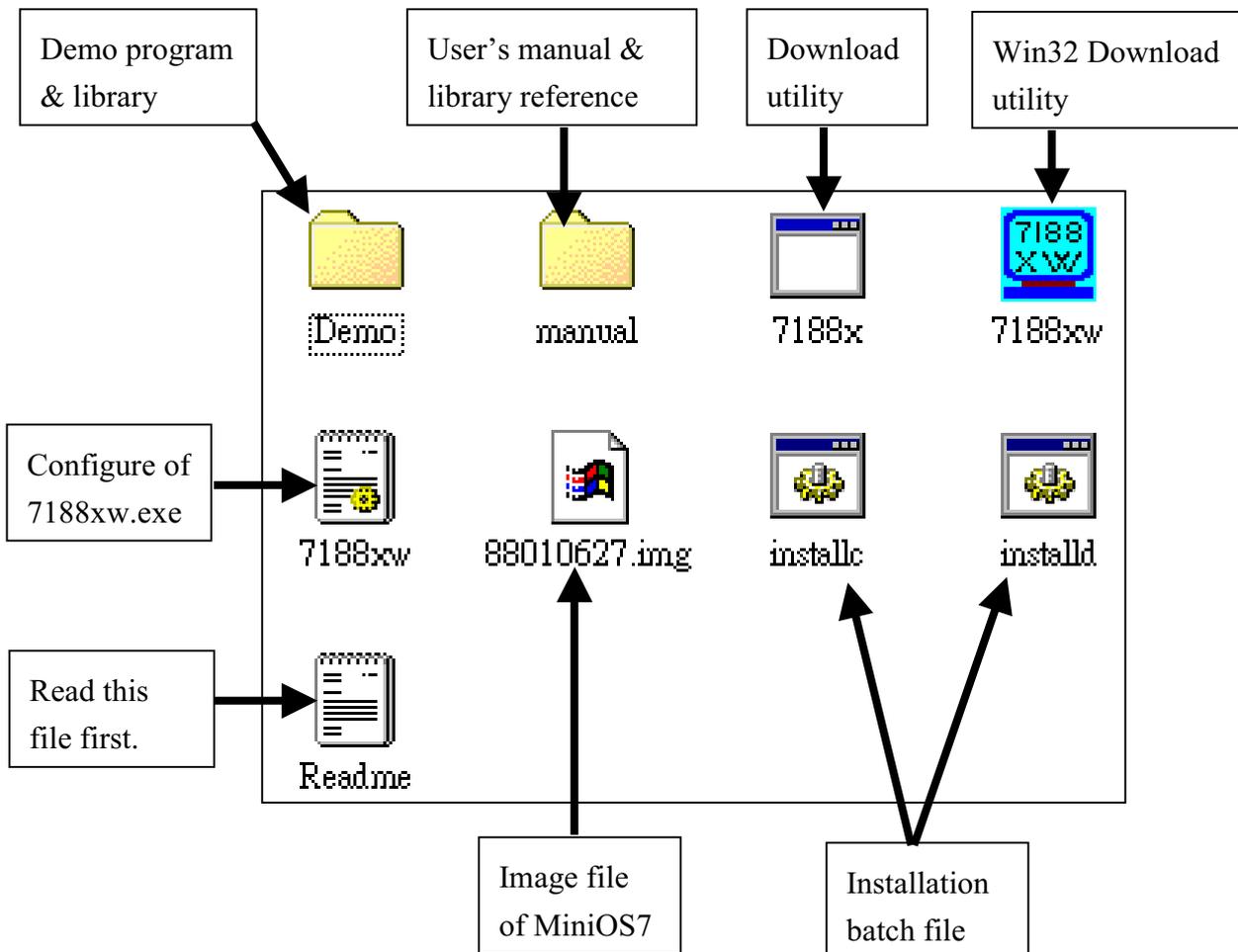
Step 6: close the installation program

Note 1: If you have to install in another location, you only have to copy all files under **CD\napdos\7188e\miniOS7\\*.\***

Note 2: The batch file of installation to c:\7188OS7 is given as follows:

```
@echo off
echo The installc.bat will copy all I-7188(D) files to c:\7188OS7\
md c:\7188OS7
xcopy . c:\7188OS7\*.* /s /v
echo Install complete.
@echo on
```

After all software are installed on C, the content of c:\7188OS7 is given as follows:



**Note:** the download utility, 7188X.EXE & 7188XW.EXE, is used as a bridge between 7188(D) & PC. So user have to copy one of these 2 files to C:\Windows for accessing it in any location.

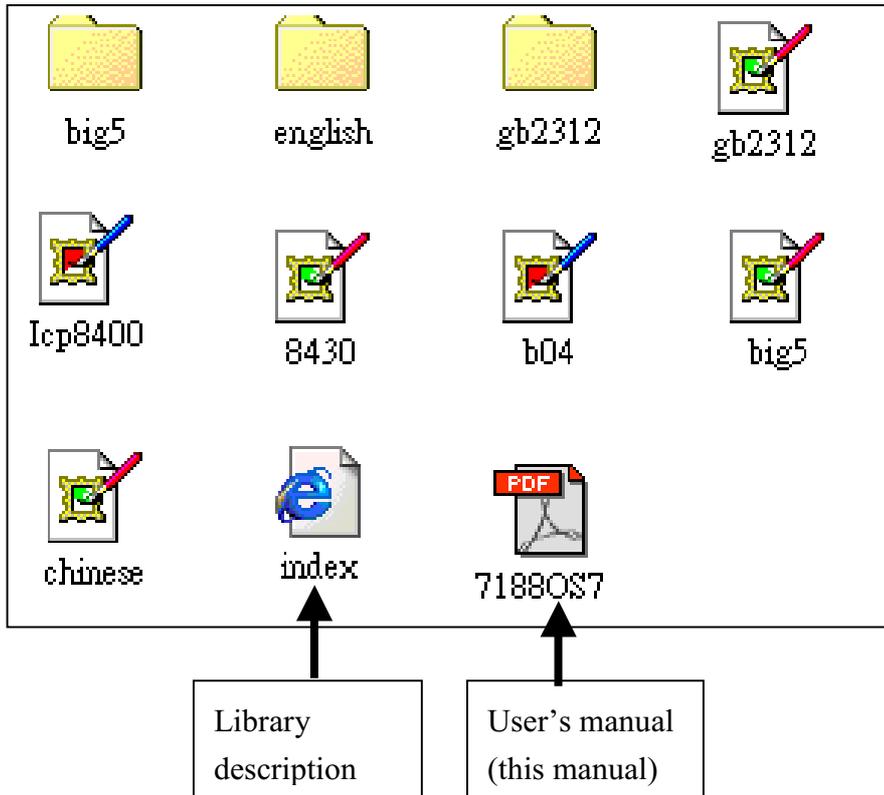
- Copy 7188X.EXE c:\windows
- Or copy 7188XW.EXE c:\windows

**Now you can execute 7188X.EXE/7188XW.EXE in any location.**

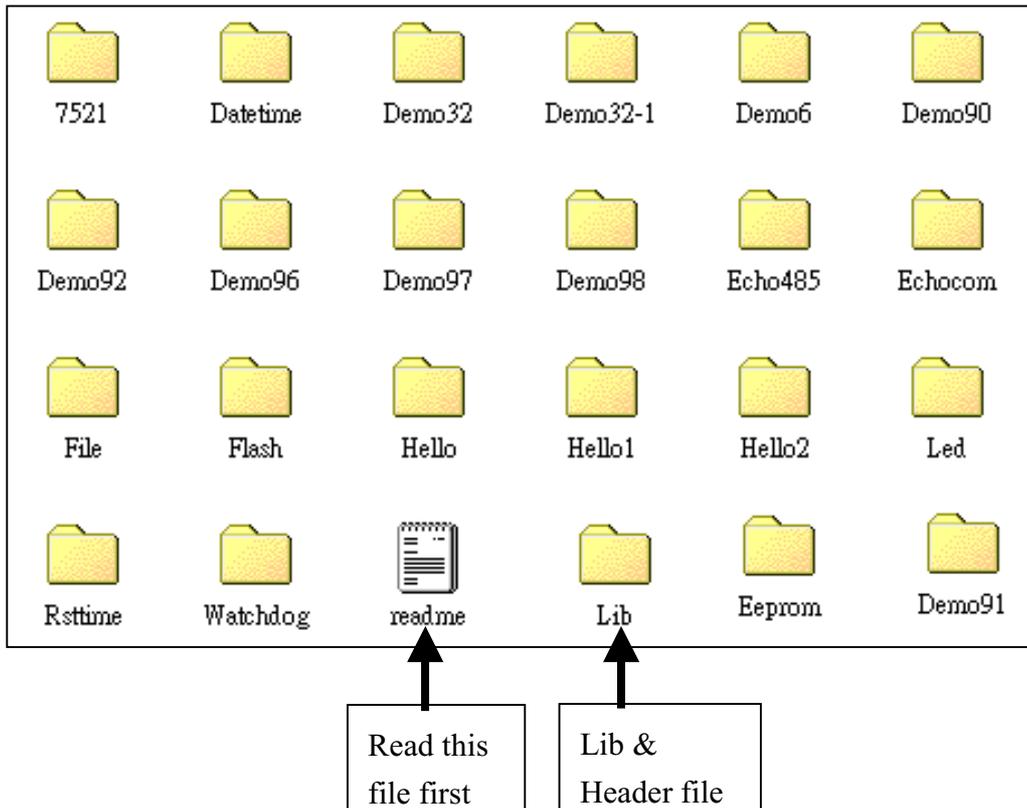
## (1) Introduction

---

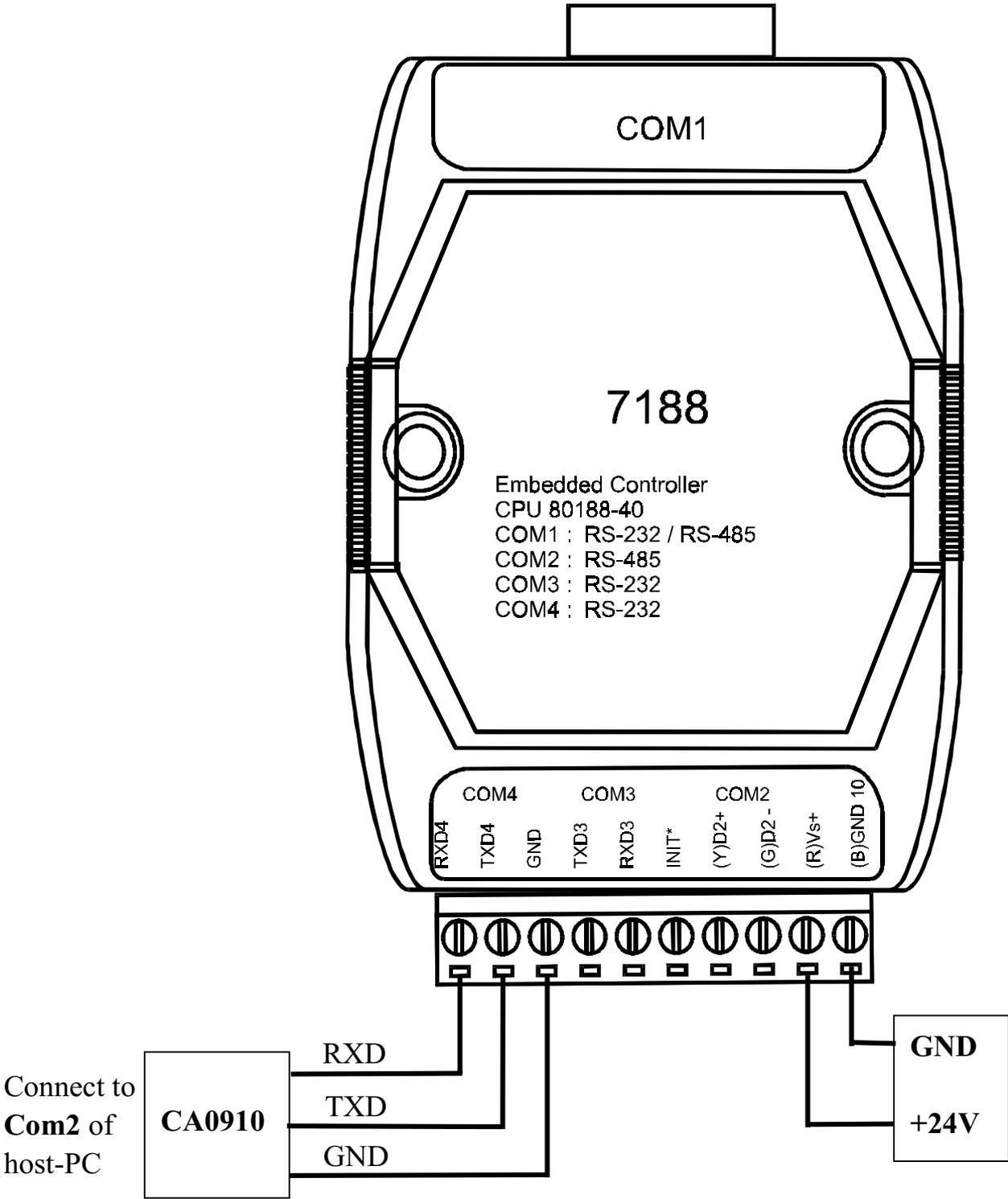
The content of directory **Manual** is given as follows:



The content of directory **Demo** is given as follows:



1.1.2 Connect Download Cable to Host-PC



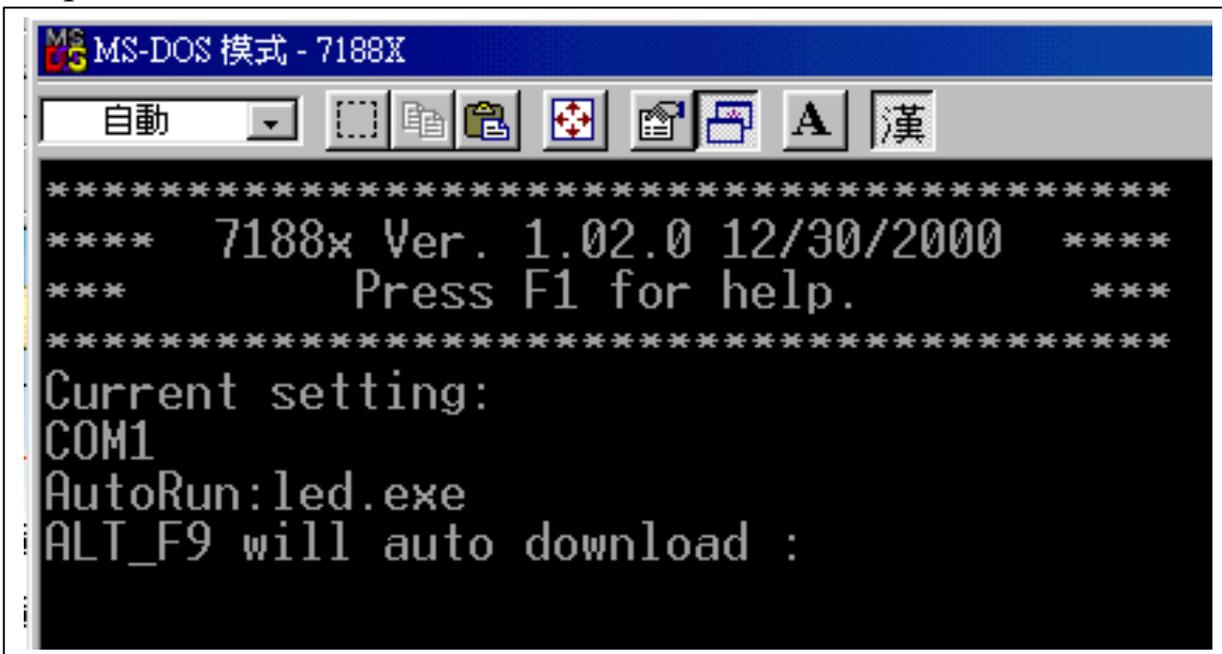
### 1.1.3 Download Led.exe & Autoexec.bat to 7188(D)

Note: now COM4 of 7188(D) is linked to **COM2** of host-PC by download cable, CA0910

Step 1: execute a DOS-BOX in windows

Step 2: cd c:\7188OS7\demo\led

Step 3: execute 7188X.EXE as follows:



Now the default download file=**led.exe** & the default status of download port is COM1, 115200 BPS & N81 as follows:



Step 4: Press **ALT & 2** at the same time to change to COM2 as follows:



Note: if the configuration of com2 is not equal to 115200, N81, you can press **ALT & C** to change configuration as follows:

- ALT + C → change configuration
- Space → change to next value
- Enter → select this value

Step 5: Press **Enter** twice as follows:

```
*****  
**** 7188x Ver. 1.02.0 12/30/2000 ****  
***          Press F1 for help.      ***  
*****  
Current setting:  
COM2  
AutoRun:led.exe  
ALT_F9 will auto download :  
  
I-7188>  
  
I-7188>_
```

Step 5: Key-in **ver** & press **Enter** as follows:

```
I-7188>ver  
  
ICP_DAS MiniOS7 for 7188(D) Ver. 1.02  
SRAM:256K, FLASH MEMORY:512K
```

Step 6: Key-in **dir** & press **Enter** as follows:

```
I-7188>dir  
  
Total File number is 0 Free space=458720 bytes  
I-7188>_
```

Now the led of 7188(D) is ON.

## ***(1)Introduction***

---

Step 7: press F8 to download & execute **led.exe** as follows:  
(refer to step 3)

```
I-7188>load
File will save to 8000:0000
StartAddr-->7000:FFFF
Press ALT_E to download file!
Load file:led.exe

Send file info. total 39 blocks
Block 39
Transfer time is: 2.307692 seconds

I-7188>led.exe
Test functions: LedOn,LedOff,StopWa
and KbhIt,Getch,Putch
Press 'q' to quit.
```

Now the **led.exe** is executed & the LED of 7188(D) will be ON/OFF every 0.5 second.

Step 8: Press **123456q** to stop **led.exe** as follows:

```
Press 'q' to quit.
123456
I-7188>_
```

Note: **123456** is used for testing, **q** is used to stop **led.exe**.

Step 9: key-in **dir** & **Enter** as follows:

```
I-7188>dir

 0)led.exe      11/08/2000 16:49:08
Total File number is 1  Free space=
I-7188>_
```

Step 10: key-in **load** & **Enter** first

Then press **ALT & E** at the same time

And key-in **autoexec.bat** to download autoexec.bat to 7188(D) as follows:

```
I-7188>load
File will save to 8266:000A
StartAddr-->8000:2669
Press ALT_E to download file!
Input filename:autoexec.bat
Send file info. total 1 blocks
Block 1
Transfer time is: 0.054945 seconds
I-7188>
```

Step 11: key-in **dir** & **Enter** as follows:

```
I-7188>dir
 0)led.exe      11/08/2000 16:49:08
 1)autoexec.bat 09/22/2001 13:24:09
Total File number is 2 Free space=
I-7188>
```

Step 12: key-in **type autoexec.bat** & **Enter** as follows:

```
I-7188>type autoexec.bat
led.exe
→
I-7188>
```

Step 13: power off 7188(D) & power-on 7188(D) again. **Now the MiniOS7 will execute autoexec.bat automatically.** So the LED of 7188(D) now will be ON/OFF every 0.5 second.

Press **ALT & X** at the same time to return to DOS-BOX.

## ***(1)Introduction***

---

The content of LED.C is given as follows:

---

```
#include"..\\lib\\i7188.h"

int quit=0;
int LedMode=0;
int data;
unsigned long time;
void main(void)
{
    Print("Test functions: LedOn,LedOff,
        StopwatchStart ,StopWatchReadValue\\n\\r");
    Print("and Kbhit,Getch,Putch\\n\\r");
    Print("Press 'q' to quit.\\n\\r");
    TimerOpen();
    StopwatchStart(0);
    LedOff();
    while(!quit){
        StopwatchReadValue(0,&time);
        if(time>=500){
            StopwatchStart(0);
            LedMode=!LedMode;
            if(LedMode) LedOn();
            else LedOff();
        }
        if(Kbhit()){
            data=Getch();
            if(data=='q' || data=='Q') quit=1;
            else {
                Putch(data);
                if(data=='\\r') Putch("\\n");
            }
        }
    }
    TimerClose();
}
```

## 1.1.4 Download Hello.exe & Autoexec.bat to 7188(D)

Step 1: power-off 7188(D). Connect INIT\*-pin & GND-pin together. Power-on 7188(D). Now you will find the LED of 7188(D) is ON. Compare to step 13 of Sec. 1.1.3, the only difference is INIT\*-pin. When the INIT\*-pin is open, the MiniOS7 will search & execute autoexec.bat. If the INIT\*-pin is connected to GND-pin, the MiniOS7 will don't care the autoexec.bat

Step 2: execute a DOS-BOX in windows

Step 3: cd c:\7188OS7\demo\hello

Step 4: execute hello.exe in PC as follows:

```
C:\7188os7\DEMO\HELLO>hello
*** Hello I-7188! ***
C:\7188os7\DEMO\HELLO>
```

You will find this program can be executed in the DOS-BOX of PC.

Step 5: execute 7188X.EXE:

Step 6: key-in **dir** & **Enter** as follows:

```
I-7188>dir
 0)led.exe      11/08/2000 16:49:08
 1)autoexec.bat 09/22/2001 13:24:09
Total File number is 2  Free space=
I-7188>_
```

## ***(1)Introduction***

---

Step 7: Press **F2** first

Then key-in **hello.exe** & Enter

And press **F8** to download & execute hello.exe as follows:

```
I-7188>
Input filename:hello.exe
When Press F8/F9 will auto download the file:hello.exe
load
File will save to 8269:0004
StartAddr-->8000:2693
Press ALT_E to download file!
Load file:hello.exe

Send file info. total 17 blocks
Block 17
Transfer time is: 0.989011 seconds

I-7188>hello.exe
*** Hello I-7188! ***

I-7188>_
```

Step 8: key-in **load** & **Enter** first

Then press **ALT & E** at the same time

And key-in **autoexec.bat** to download autoexec.bat to 7188(D) as follows:

```
I-7188>load
File will save to 8377:0008
StartAddr-->8000:3777
Press ALT_E to download file!
Input filename:autoexec.bat
Send file info. total 1 blocks
Block 1
Transfer time is: 0.109890 seconds

I-7188>_
```

Step 9: key-in **dir** & **Enter** as follows:

```
I-7188>dir
 0)led.exe      11/08/2000 16:49:08
 1)autoexec.bat 09/22/2001 14:02:27
 2)hello.exe    11/08/2000 16:44:05
 3)autoexec.bat 09/22/2001 14:11:20
Total File number is 4 Free space=44
I-7188>
```

Step 10: key-in **Type autoexec.bat** & **Enter** as follows:

```
I-7188>type autoexec.bat
hello
*
I-7188>
```

In step 9, you can find 2 autoexec.bat. In step 10, you can find that MiniOS7 will access the most new one. In general, MiniOS7 will always access to the most new one if there are multiple files with same filename. This feature make the program development stage more easy as follows:

Step a: edit \*.c

Step b: compiler \*.c to generate the filename.exe

Step c: download filename.exe to 7188(D)

Step d: execute & test filename.exe

Step e: if there are some bugs, go to step a.

In step c, you can download multiple filename.exe to 7188(D), MiniOS7 will always access the most new one & execute it.

Step 11: Power-off 7188(D) & let INIT\*-pin open. Power-on 7188(D) as follows:

```
I-7188>*** Hello I-7188! ***
I-7188>
```

You can find that MiniOS7 find the most new autoexec.bat & execute it.

## ***(1)Introduction***

---

Step 12: key-in **del /y** & **Enter**

And **dir** & **Enter** as follows:

```
I-7188>del /y
Total File number is 4, do you really want to delete(y/n)?
I-7188>dir
Total File number is 0 Free space=458720 bytes
I-7188>
```

Note: the MiniOS7 only support **delete all** command.

Step 13: Press F8 to download & execute hello.exe (step 7)

Step 14: Press F8 to download & execute hello.exe

Step 15: Press F8 to download & execute hello.exe

Step 16: key-in **dir** & **Enter** as follows:

```
-7188>dir
0)hello.exe    11/08/2000 16:44:05
1)hello.exe    11/08/2000 16:44:05
2)hello.exe    11/08/2000 16:44:05
otal File number is 3 Free space=44
-7188>
```

The content of hello.c is given as follows:

```
#include"..\lib\i7188.h"
```

```
void main(void)
```

```
{
  Puts("*** Hello I-7188! ***\n\r");
}
```

## **1.1.5 Demo Program of 7188(D)**

There are many demo program given in C:\7188OS7\demo\\*.\* as follows:

7521 : Demo program for 7521 as an Addressable RS-485 to RS-232 converter.

DateTime : Get the system Date and Time

eeprom: Multi Read and write EEPROM test

file: Open File

flash: All of the flash function

led : LED function test

reset: 7188 boot time test

watchdog: demo program for I-7188 to test if I-7188 is reset by watchdog timer.

-----  
demo6: read-EEPROM & write-EEPROM test  
-----

demo32: Connect to Touch7(Touch 200) +7060+7013+7017+7021(DemoBox2)

demo32-1: Demo program run on I-7188 with MiniOS7. For Training BOX II  
Using new driver to connect to Touch-200  
-----

demo90: Demonstration for use Timer

function used:TimerOpen,TimerClose,TimerReadValue,TimerResetValue.

demo91: Demonstration for use Timer functions.

use CountdownTimer channel 0 for LED ON/OFF

demo92: Demonstration for use Timer functions.

use Stopwatch channel 0 for LED ON/OFF

functions: TimerOpen,TimerClose,StopWatchStart,StopWatchReadValue.

demo96: Demonstration for use Timer functions.

use User timer function to do the same thing as demo94.c

functions: TimerOpen,TimerClose,InstallUserTimer;

demo97: Demonstration for use Timer functions.

use DelayMs for LED ON/OFF

demo98: use I-7188 timer function to send/receive data to/from 7000's modules.

and set timeout in time unit ms. COM2: 9600,N,8,1 RS-485  
-----

## **1.1.6 More Demo Program of 7188(D)**

There are many demo program given in CD\napdos\7188\romdos\C\\*. \* designed for 7188(D)/DOS. All these demo programs are designed for ROMDOS. But most of them can be converted to MiniOS7 very easy.

torom: download ROM-DISK.IMG to Flash Memory  
ld3: very high speed download program to RAM-DISK  
loadram: low speed download program to RAM-DISK  
loadram2: high speed download program to RAM-DISK  
echo485: monitor/send command from/to COM2(RS-485)  
echoCOM: monitor/send command from/to COM1/COM2/COM3  
emu7520: use I-7188 emulation I-7520 or as protocol converter  
flash: read/write/erase FLASH memory,check I-7188's flash size.  
WatchDog: Check if 7188 was reset by watchdog timer  
COM3COM4: Show how to write COM3/COM4 ISR  
MMICON: Connect to MMICON(PC mode)(use MMICON as terminal)  
MMICON2: Connect to MMICON(PC mode)(use MMICON as terminal)[method 2]  
7000tot: 7000 TimeOut Test. Test 7000 timeout count range  
for function ReceiveResponseFrom7000.  
printf: demo for use function printCom,printCom1/2/3/4  
MODBUS\MASTER: demo for ModBus protocol(for master mode)  
MODBUS\SLAVE: demo for ModBus protocol(for slave mode)  
USE485:(08/30/99) demo program for use half-duplex RS-485  
RTSDTR:(09/02/99) demo for Set RTS & DTR pin of COM1  
LOAD:(12/31/99) download file to RAM-DISK (7188.exe Ver.1.11.0,ALT\_E)  
-----  
QC\QC1: QC program 1 for 7188  
QC\QC2: QC program 2 for 7188  
QC\QC3: QC program 3 for 7188  
QC\QC4: QC program 4 for 7188  
-----  
demo0: read/write RTC by ROM-DOS  
demo1: basic I/O test  
demo2: 2\*RS-232 + 1\*RS-485 test  
demo3: 2\*RS-485 test  
demo4: 3\*MMI\_CON + 1\*7011 + 1\* 7012

demo5: read-NVRAM & write-NVRAM test  
demo6: read-EEPROM & write-EEPROM test  
demo7: watchdog enable/disable  
demo8: modem application & how to write COM1/COM2 ISR  
demo9: write data to FlashRom

-----  
demo10: connect to 7000, 4\*7052 + 2\*7043 + 1\*7053  
demo11: connect to 7000, 7053 + 7053  
demo12: connect to 7000, 7053 + 7051  
demo13: connect to 7000, 7017 + display  
demo14: connect to 7\*7053(16\*7=112 DI)  
demo18: connect to 7000, 4\*7067 + 4\*7042 + 4\*7052  
test18: test program for demo18.(run on the master controller)

-----  
demo20: PLC I/O expansion, SIEMENS's S7  
demo21: connect to FACON PLC

-----  
demo30: Connect to Touch Screen, TOUCH7(Touch 200)  
demo31: Connect to Touch Screen, HITECT, PWS-1700  
demo32: Connect to Touch7(Touch 200) +7060+7013+7017+7021(DemoBox2)  
demo33: Connect to 7060+7013+7017+7021+7080D(DemoBox3)  
demo34: Connect to Touch7(Touch 200)+7060+7013+7017+7021+Step motor(DemoBox4)  
demo35: Connect to Touch7(Touch 200)+7060D+7017+7021+Step motor  
          +SST-2400 +PC1 (demo box for Taipei computex show 99')  
demo36: Connect to MMICON+7065D+7016D+7033D+SST-2400+PC2  
          (demo board for Taipei computex show 99')  
demo37: Connect to 7013D+7021+7012D+7044D

-----  
demo40: Connect to Card Reader

-----  
demo50: IKIT-07 demo program  
demo51: IKIT-08 demo program  
demo52: IKIT-09 demo program(Not finished)

-----  
demo60: Set 7188 address(for connect more than one 7188s with 485 bus)  
demo61: Connect PC to more than one 7188 (run on 7188)  
demo62: Connect PC to more than one 7188 (run on PC)

## ***(1)Introduction***

---

demo70: PC0+7520--+-- 7188-1 + PC1  
+-- 7188-2 + PC1  
.....  
+-- 7188-n + PCn  
(RS-485) (RS-232)

-----  
demo80: User timer Interrupt Service Routine(ISR)(INT 0x1c)

demo81: D/O and Watchdog Timer(Not finished)

demo82: D/A and Watchdog Timer(Not finished)

demo83: use 7188 COM1 for 2\*D/O and 4\*D/I(Not finished)

-----  
demo90: Use I-7188 Lib's Timer function (1ms)

demo91: Use I-7188 Lib's Count Down Timer

demo92: Use I-7188 Lib's StopWatch Timer

demo93: Use I-7188 Lib's StopWatch to calculate other functions  
spent how much time.

demo94: Use I-7188 Lib's StopWatch to do something in fixed time interval.

demo95: Use I-7188 Lib's Timer to install user timer function.

demo96: Use I-7188 Lib's user timer function to do something in fixed  
time interval.

demo97: use function DelayMs

demo98: Use user Timer function for receive 7000's response and  
set timeout in time.

## 1.1.7 More about MiniOS7

### Typical Command Set Table of MiniOS7

command	Description
LED5 pos value	Show a HEX value in the specified position of 5-digit LED
USE NVRAM	Into the service routine for read/write NVARM
USE EEPROM	Into the service routine of read/write EEPROM
USE EEPROM	Into the service routine of read/write Flash-ROM.
USE COM2 /option	Into the service routine of send/receive to/from COM2 (RS-485).
DATE [mm/dd/yyyy]	Set the date of RTC.
TIME [hh:mm:ss]	Set the time of RTC
MCB	Test current memory block.
UPLOAD	The first step to update the MiniOs7.
BIOS1	The last step to update the MiniOs7.
LOAD	DOWNLOAD the user program into the Flash-Memory
DIR [/crc]	Show the information of all files download in the Flash-Memory.
RUN fileno	Execute the file with file-number=fileno.
name	Execute the file with file-name=name.
DELETE (or DEL)	DELETE all files stored in the Flash-Memory. It will delete all files.
RESET	Reset the CPU.
DIAG [option]	Hardware Diagnostic.
BAUD baudrate	Set the new value of communication-baudrate to baudrate.
TYPE filename [/b]	List the content of file.
REP [/#] command	repeat execute the same command # times
RESERVE [n]	reserve n Flash Memory sector for USER's program
LOADR	download to SRAM
RUNR [param1 [param2...]]	Run the program save on SRAM.(download by command LOADR)
I/INP/IW/INPW port	Read data from the hardware PORT
O/OUTP/OW/OUTP W port value	Output to hardware PORT
..... more .....	..... more .....

Note: Refer to **CD\napdos\8000\common\minios7\doc\index.html** for more information about MiniOS7. The MiniOS7 is designed for 7188/7188X/7188E/8000 family, so you will find some information which are not related to 7188(D).

## 1.1.8 MiniOS7 Upgrade

We will add more & more new features to MiniOS7. So user can upgrade his MiniOS7 for new features. There are two methods, using 7188X.EXE or using EEPROM writer, to upgrade MiniOS7 as follows:

### 1.1.8.1 Upgrade MiniOS7 by Using 7188X.EXE

**Step 1:** Get the new MiniOS7 image file

The format of image file name is given as → TTYMMDD.img

TT → TYPE, "xc"=7188XC/7521, "80"=8000

YY → year of this image released

MM → month of this image released

DD → day of this image release

**Step2 :** On PC, please change to the directory where the image can be found. Then run 7188X.EXE connect to 8000 that run MiniOS7.

**Step3:** Key-in command: **UPLOAD**, then according to the message show on screen, please press keys: ALT + E.

```
*****
**** 7188x Ver. 1.01.2 04/14/2000 ****
***          Press F1 for help.          ***
*****
Current setting:
COM1
AutoRun:main.exe
ALT_F9 will auto download :

7188x>upload
Press ALT_E to download file!
```

**Step 4:** Then key-in the image filename.(TTYMMDD.img)and press ENTER. after UPLOAD is finished.(the IMAGE will save on SRAM)

```
7188x>upload
Press ALT_E to download file!

Input filename:xc000522.img
Send file info. total 256 blocks
Block 256
Transfer time is: 34.780220 seconds

Back to Terminal mode
CRC16=7EDE
1000:FFFE=7188

7188x>
```

**Step 5:** Key-in command:BIOS1.(MiniOS7 will check the image file, and display the image file's version,date. If it is OK, then write the image file to FLASH.) The update time is ablut 10-20 seconds. After update it will auto reset, If not, please power off then power on it. The user can use command ver to check if the new OS in on it.

```
I-7188(D)>upload
Press ALT_E to download file!
Input filename:80010928.img
Send file info. total 256 blocks
Block 256
Transfer time is: 13.681319 seconds
CRC16=D687
3000:FFFE=7188

I-7188(D)>bios1
MiniOs7 for 7188(D) Ver 1.02.015, date=09/28/2001
Checking CRC-16...OK.
Write to Flash memory, please wait about 10 secs.
After update finished, the system will auto reboot.
If it take too long, please reset by power On-Off.
I-7188(D)>ver

ICP_DAS MiniOS7 for 7188(D) Ver. 1.02 build 015,Sep 28 2001
SRAM:256K, FLASH MEMORY:512K
```

### 1.1.8.2 Upgrade MiniOS7 by using EEPROM writer

**Step 1:** Get the new MiniOS7 image file

The format of image file name is given as → TTYMMDD.img

TT → TYPE, "xc"=7188XC/7521, "80"=8000

YY → year of this image released

MM → month of this image released

DD → day of this image release

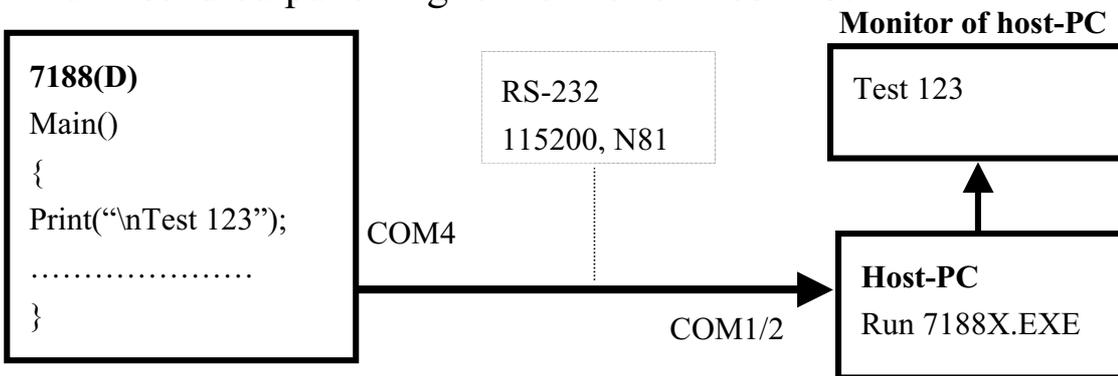
**Step2:** Please unplug the flash of 8000. For 256K byte is 29F002T, for 512K is 29F040.

**Step3:** Please write the IMAGE file to FLASH MEMORY's last 64K.( offset of 256K is 0x30000, offset for 512K is 0x70000)

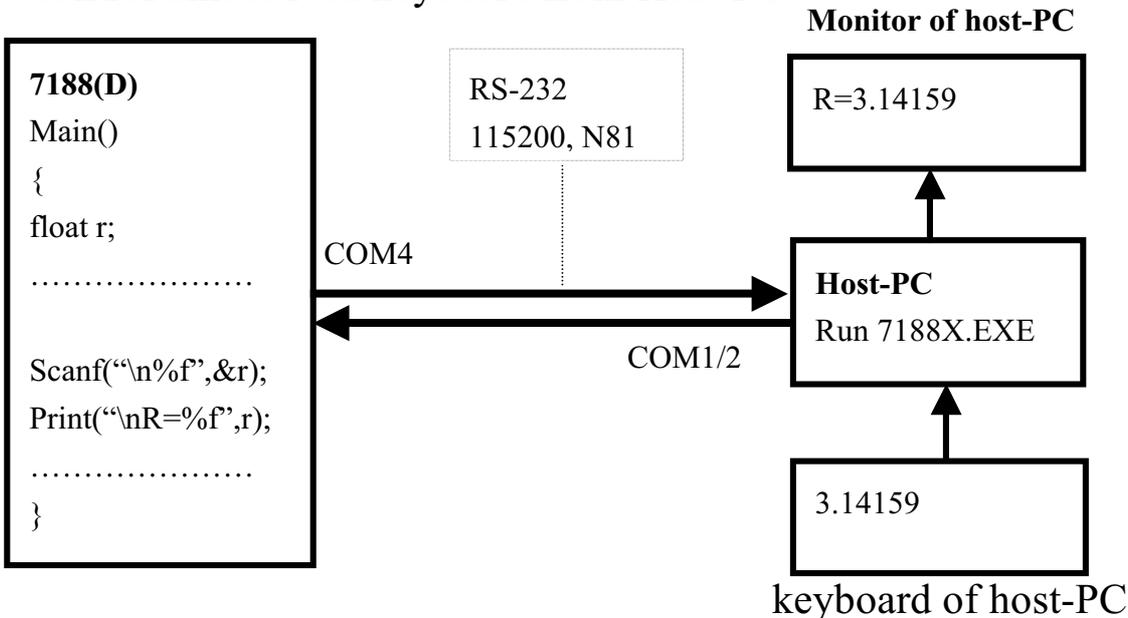
### 1.1.9 7188X.EXE

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

- **download user's program from host-PC into main control unit**  
refer to Sec. 2.6 for more information.
- **show debug string to monitor of host-PC**  
Three standard output libraries, Putch, Print & Puts, will allow main control unit send output string to monitor of Host-PC.



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



## ***(1)Introduction***

---

### Typical Function Tables of 7188X.EXE:

command	Description
ALT_1	select PC's COM1
ALT_2	select PC's COM2
ALT_C	change baudrate & data format Press ENTER to confirm the setting. Press LEFT/RIGHT-ARROW KEY to select different field. Press ANY KEY to switching between different value. Press ENTER in the last field will stop this operation.
ALT_D	Set the date of RTC as same as PC's date.
ALT_T	Set the time of RTC as same as PC's time
ALT_E	Switch to download file mode, then user must input the download filename.
ALT_L	Switch to line-mode. In line-mode, all characters-pressed will not send to COM until the ENTER is pressed. It is designed for 7000 series. Press ALT_L again will return to the original mode.
ALT_X	Quit the 7188X.EXE.
ALT_F2	Set multiple filenames for download.
ALT_F9	download all files specified by ALT_F2.
CTRL_F2	View the COM1 & COM2 at the same time. It is dsigned for easy testing. Press ALT_X will return to the original mode. Press TAB can switch the cursor between these two windows.
F2	Set the filename for download (no download operation).
F5	execute the file specified by use F2 function.
F6	Set the input areguments of the execution file.
F8	F8=F9+F5.
F9	Download the file specified by F2.
F10	Auto download and excute, the same function as F8,.but the file is download to SRAM and will overwrite last download file.
F12	For 7521/7522/7523 to test RS-232.
..... more .....	..... more .....

### 1.1.10 Libraries

There are two libraries for 7188(D) as following:

(1)I7188S.LIB for SMALL MODEL program.(TC/BC++/MSC/MSVC++)

(2)I7188L.LIB for LARGE MODEL program.(TC/BC++/MSC/MSVC++)

All the functions declared are described in I7188.h. The application program has to add the following: to the start of code.

#include "I7188.h".

There are hundreds of functions are supported in I7188s.lib/I7188l.lib as following:

Function description	Example
COM port	InstallCom0, InstallCom1, InstallCom2, InstallCom3, InstallCom4 IsCom0, IsCom1, IsCom2, IsCom3, IsCom4 ToCom0, ToCom1, ToCom2, ToCom3, ToCom4 ReadCom0, ReadCom1, ReadCom2, ReadCom3, ReadCom4 ..... more .....
EEPROM	WriteEEP, ReadEEP, EnableEEP, ProtectEEP ..... more .....
NVRAM & RTC	ReadNVRAM, WriteNVRAM, GetTime, SetTime, GetDate, SetDate ..... more .....
LED & 5DigitLed	LedOn, LedOff, Init5DgitLed, Show5DigitLed, Show5DigitLedWithDot ..... more .....
Flash Memory	FlashReadId, FlashErase, FlashRead, FlashWrite ..... more .....
Timer & Watchdog Timer	TimerOpen, TimerClose, TimerResetValue, TimerReadValue StopWatchReset, StopWatchRead, StopWatchStop InstallUserTimer EnableWDT, DisableWDT, RefreshWDT ..... more .....
File	GeFileNo, GetFileName, GetFilePositionByNo, GetFilePositionByName ..... more .....
Connect to 7000	SendCmdTo7000, ReceiveResponseFrom7000
Programmable I/O	SetDio4Dir, SetDio4High, SetDio4Low, GetDio4
Others	Kbhit, Getch, Putch, LineInput, Scanf ..... more .....

Refer to CD\napdos\7188\minios7>manual\index.html for more information

### **1.1.11 Compiler & linker**

User must use C Language to write program. You can use TC, TC++, BC++, MSC or MSVC++ ( before 1.52). Please take care of the following items:

Generate a standard DOS executable program.

select CPU=80188/80186.

select EMULATION if the floating point computation is required.(can not select 8087)

Remove DEBUG INFORMATION to reduce program size.(MiniOS7 does not support it)

MiniOS7 provide some special libraries to replace standard I/O-libraries as following:

standard I/O-library	→ MiniOS7 provided IO-library
getch is replaced by	→ Getch
kbhit is replaced by	→ Kbhit
putchar is replaced by	→ Putch
ungetch is replaced by	→ Ungetch
puts is replaced by	→ Puts
printf is replaced by	→ Print
scanf is replaced by	→ Scanf

NOTE: We use '\n' in printf & puts for line changed display. It must be changed to '\n\r' or '\r\n' in the MiniOS7 environment.

The source code of HELLO.C is given as follows:

```
#include"..\lib\I7188.h"
void main(void)
{
Print("***Hello 7188***\n\r");
}
```

NOTE: The I7188.H is a included file.

### 1.1.11.1 Using TC

The working steps to use TC2.0 are given as following:

**Method 1: command line (refer to hello1\gotc.bat for more information)**

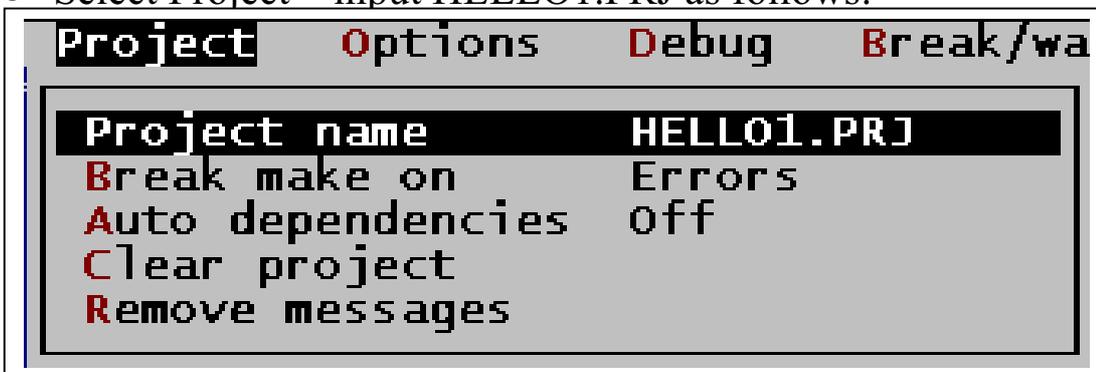
```
tcc -Ic:\tc\include -Lc:\tc\lib hello1.c ..\lib\i7188s.lib
```

**Method 2 : Use TC Integrated Environment**

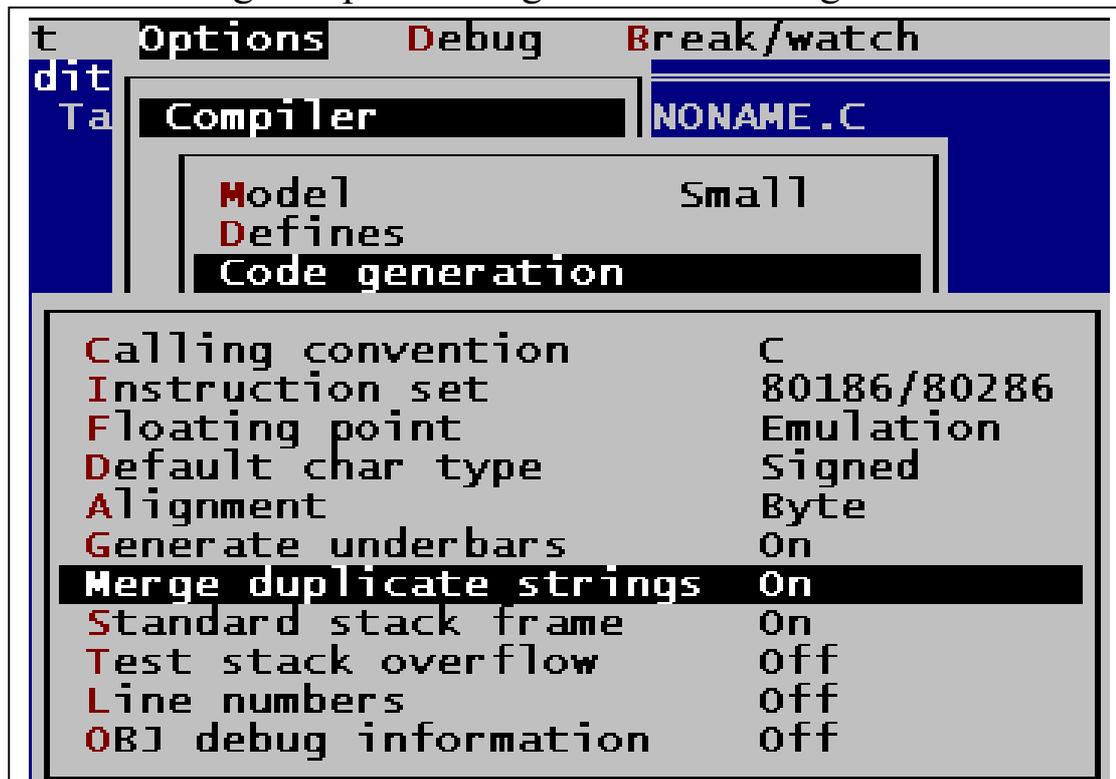
- edit the project file HELLO1.PRJ as following:

```
hello1.c  
\lib\i7188s.lib
```

- Select Project , input HELLO1.PRJ as follows:



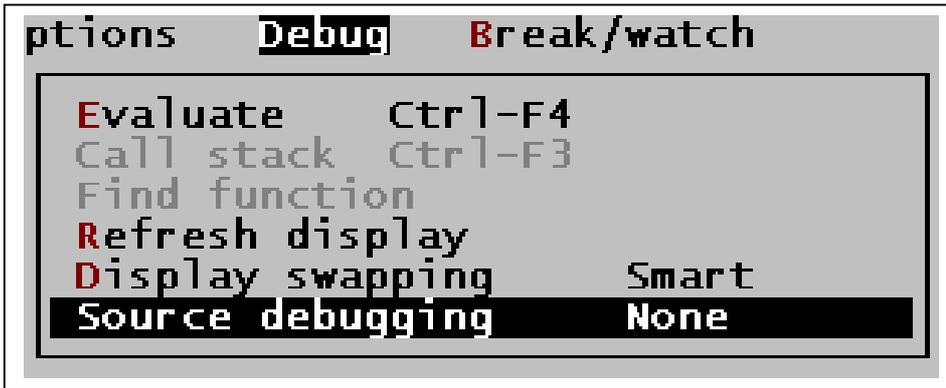
- The setting of Options are given as following:



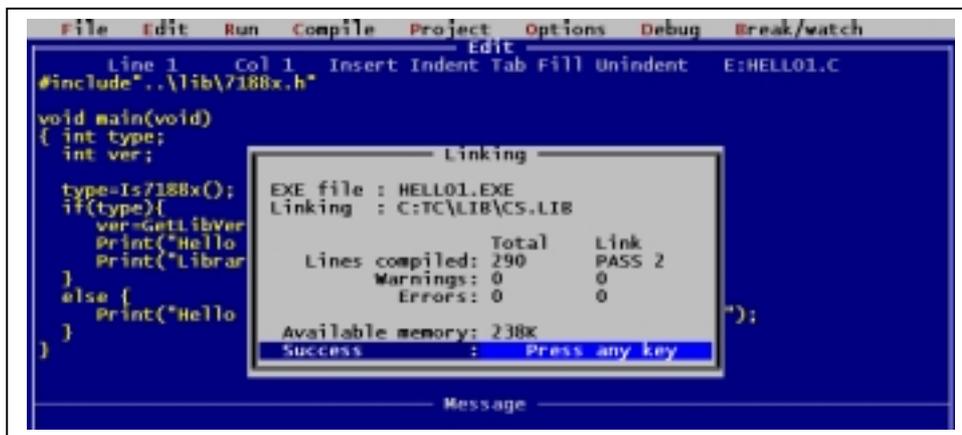
## (1) Introduction

---

- disable source debugging in the DEBUG field



- Press F9 to generate 'HELLO1.EXE' as follows:

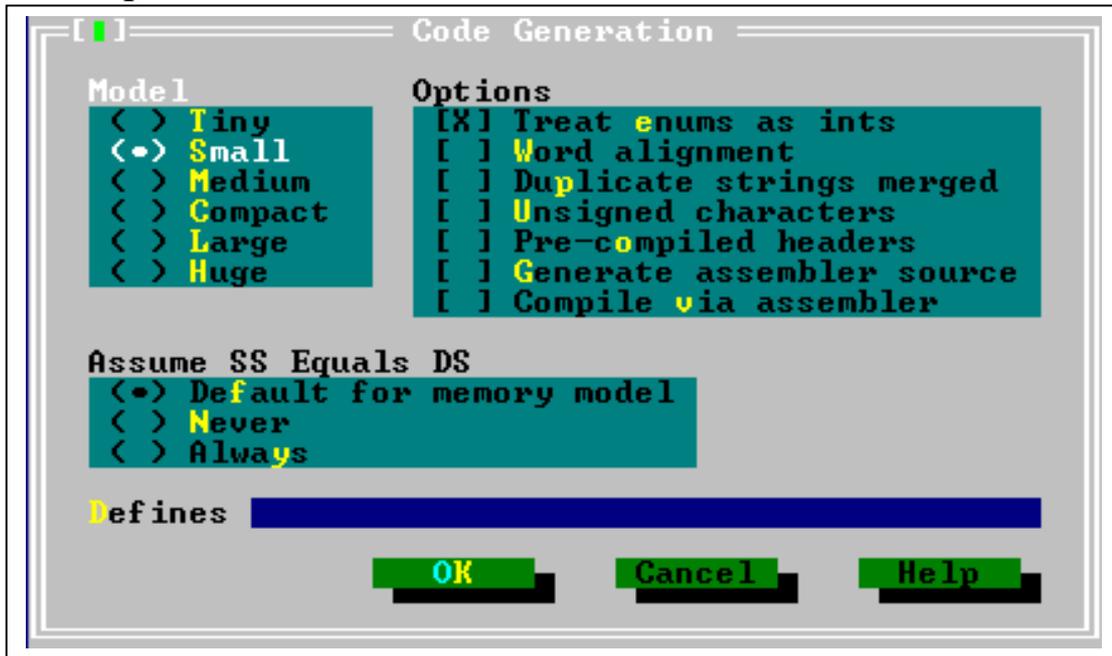




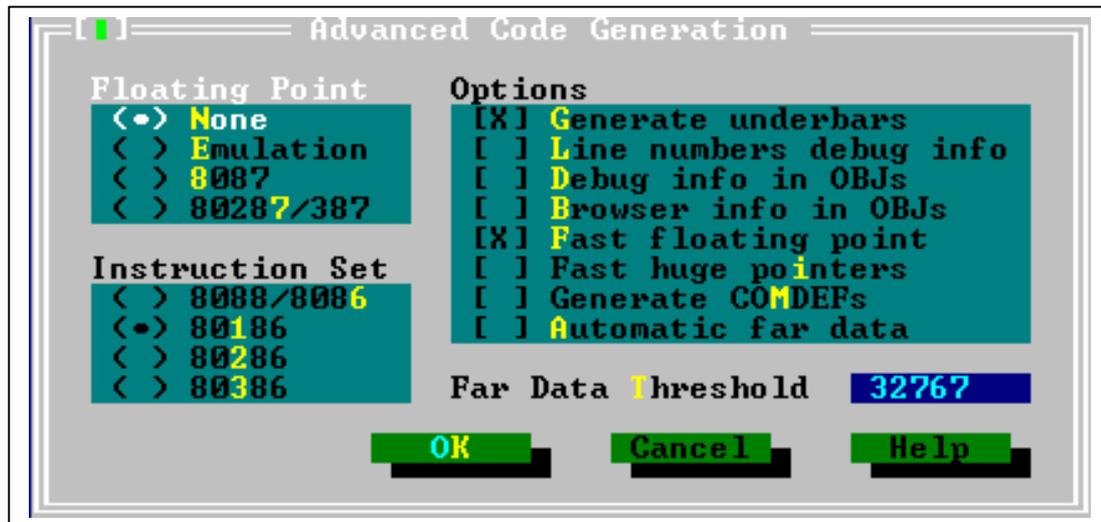
## (1) Introduction

---

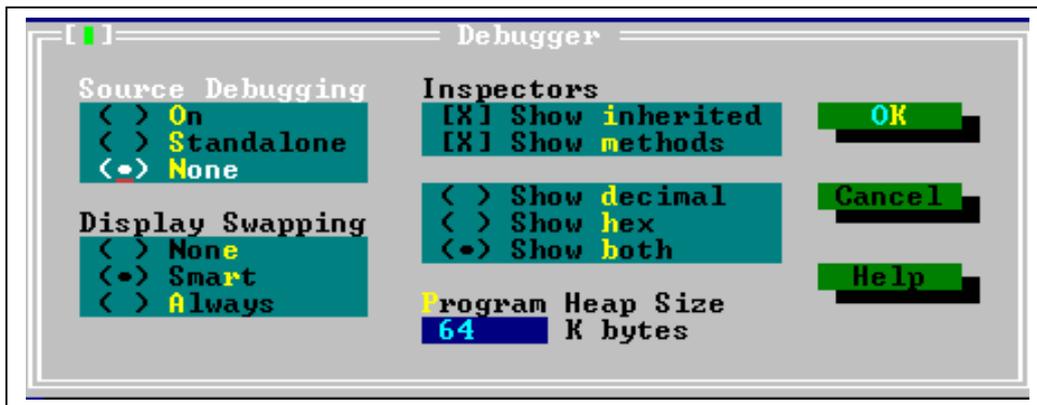
- Set some options as following:
  1. Compiler -> Code Generation



2. Compiler -> Advance Code Generation



3. Debugger (Turn source debugging off)



- Press F9 now will start the compiler&link steps as follows:

```
==== Linking =====
EXE file : HELLO1.EXE
Linking  : C:BC\LIB\GS.LIB

      Total      Link
Lines compiled: 289    PASS 2
Warnings: 0          0
Errors: 0            0

Available memory: 1979K
Success          : Press any key
```

### 1.1.11.3 Using MSC or MSVC++

The working steps to use MSC or MSVC++ are given as following:

Execute the following two commands sequentially will generate HELLO1.EXE : (refer to hello1\gomsc.bat for more information)

```
cl /c /Gs /Odi /Fm /G2 HELLO1.c
link /MA /NOE /NOI HELLO1,,,\lib\i7188s;
```

## **1.2 Features**

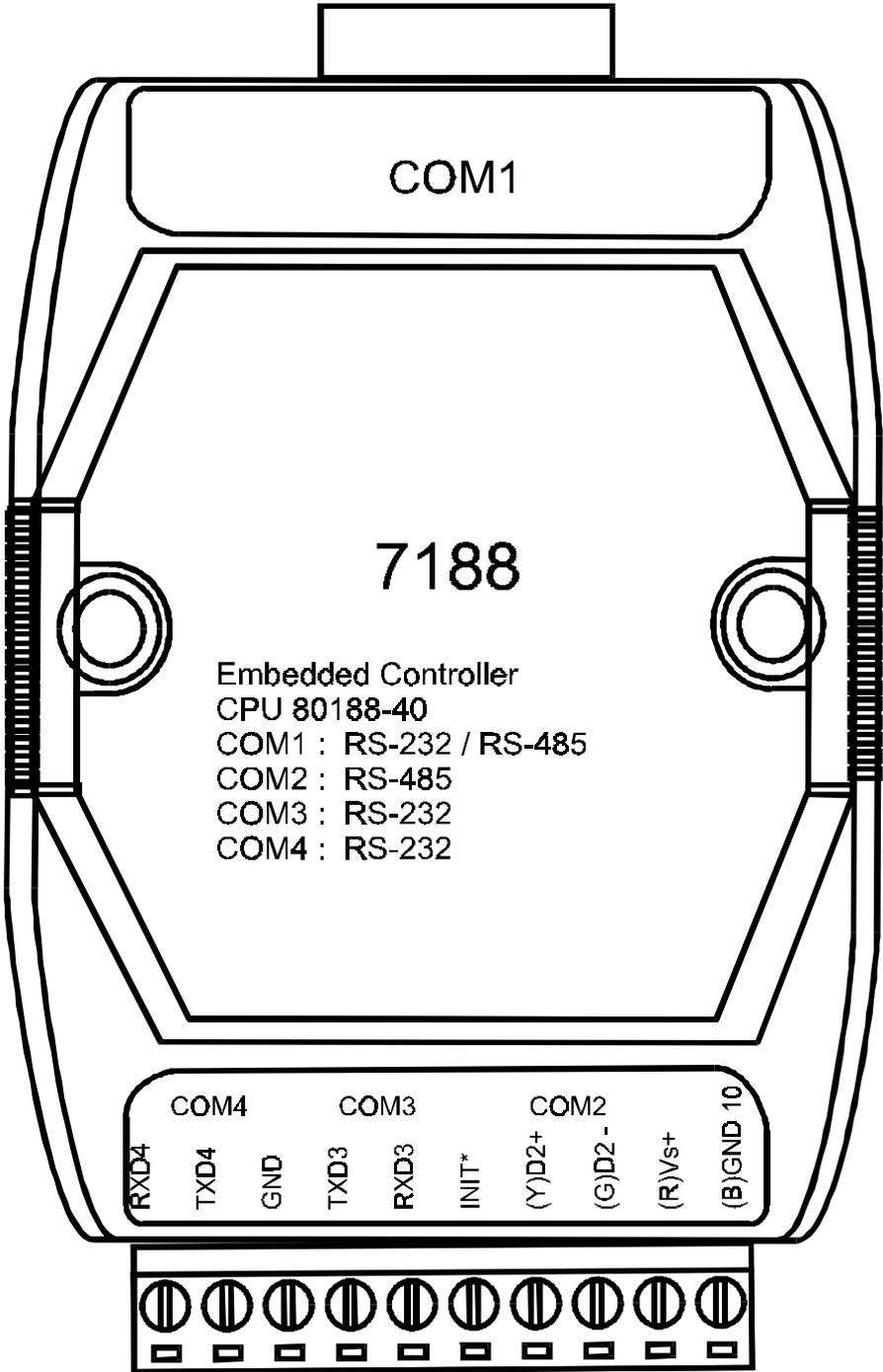
---

- AMD's embedded CPU, Am188™ES, 40M Hz
- 256K SRAM & 512K Flash ROM
- Built-in RTC & NVSRAM & EEPROM
- Built-in 4 channels of COM port
- COM driver support interrupt & 1K QUEUE input buffer
- Built-in watchdog timer for harsh environment
- Built-in power protection circuit
- Built-in RS-485 network protection circuit
- Program download from PC
- MiniOS7 inside
- **Year 2000-Compliance**
- **RTC leap year compensation from 1980 to 2079**
- BIOS support RTC time & date
- MiniOS7 support RTC time & date
- Built-in 5-digit LED display (7188D)

**WARNING 1: The COM2 is not isolated to CPU. If there is large noise in the RS-485 network, the 7188(D) may be damaged. It is recommended to add one 7510 between the COM2 & external RS-485 network for harsh environment. Refer to Sec. 1.6.4 for more information.**

**WARNING 2: The COM1 can be used as RS-232 or RS-485 port. If the COM1 is used as RS-485, it is recommended to add one 7510 between the COM1 & external RS-485 network for harsh environment. Refer to Sec. 1.6.4 for more information.**

# 1.3 Pin Assignment



**WARNING:** The COM2 is not isolated to CPU. If there is large noise in the RS-485 network, the 7188 may be damaged. It is recommended to add one 7510 between the COM2 & external RS-485 network for harsh environment. Refer to Sec. 1.6.4 for more information.

## ***(1)Introduction***

---

Pin assignment of 10-pin screw terminal block:

Pin	Name	Description
1	RXD4	RXD pin of COM4 (RS-232)
2	TXD4	TXD pin of COM4 (RS-232)
3	GND	GND pin of COM3 & COM4
4	TXD3	TXD pin of COM3 (RS-232)
5	RXD3	RXD pin of COM3 (RS-232)
6	INIT*	Initial pin for ROM-DISK download
7	D2+	DATA+ pin of COM2 (RS-485)
8	D2-	DATA- pin of COM2 (RS-485)
9	+VS	V+ of power supply (+10 to +30VDC unregulated)
10	GND	GND of power supply

Note 1: COM2=(D2+, D2-)

Note 2: COM3=(TXD3, RXD3, GND)

Note 3: COM4=(TXD4, RXD4, GND)

Note 4: COM3&COM4 share the same GND-pin(pin 3)

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

Pin	Name	Description
1	DCD	Data Carrier Detect
2	RXD	Receive Data (JP1 select RS-232)
	D1+	DATA+ of RS-485 (JP1 select RS-485)
3	TXD	Transmit Data (JP1 select RS-232)
	D1-	DATA- of RS-485 (JP1 select RS-485)
4	DTR	Data Terminal Ready
5	GND	Signal ground
6	DSR	Data Set Ready
7	RTS	Request To Send
8	CTS	Clear To Send
9	RI	Ring Indicator

Note 5: The COM1 can be used as RS-232 or RS-485 port selected by JP1.  
Refer to Sec. 1.5 for JP1 setting.

Note 6: The connector of COM1 is the same as the DB-9 RS-232 connector of PC. (JP1 select RS-232)

Note 7: The COM1 can direct drive 7000 modules if JP1 is set in the RS-485.  
Refer to Sec. 1.6.6 for wire connection.

## 1.4 Specifications

---

### System

- CPU: Am188™ES, 40M Hz
- SRAM: 256K bytes
- FLASH ROM: 512K bytes
- OS: MiniOS7 of ICP DAS
- COM port: COM1, COM2, COM3, COM4
- Built-in Watch Dog Timer (1.6 seconds)
- Built-in power protection & network protection circuit
- Program download from PC
- EEPROM support (2 K bytes)
- RTC & NVSRAM(31 bytes) support

### Real Time Clock

- Year-2000 compliance
- seconds, minutes, hours, date of the month
- month, year, valid up **form 1980 to 2079**
- NVSRAM: 31 bytes, data valid up to 10 years
- Lithium battery for RTC & NVSRAM

### EEPROM

- 2048 bytes ( 8 blocks, each block has 256 bytes)
- Data retention > 100 years
- 1,000,000 erase/write cycles

### Flash Memory

- 256/512K bytes
- 64K bytes are used by miniOS7
- At least 64K bytes are used by MiniOS7
- Others can be use to store data. (128/384K bytes max.)
- Erase unit is one sector(64K bytes)
- 100,000 erase/write cycles

## ***(1)Introduction***

---

### **COM1**

- RS-232 or RS-485 jumper select(JP1), default is RS-232
- RS-232: TXD,RXD,RTS,CTS,DTR,DSR,DCD,RI,GND
- RS-485: D1+, D1+
- Communication speed: 115200 max.
- 16 bytes FIFO

### **COM2**

- RS-485: D2+, D2-
- Communication speed: 115200 max.
- 16 bytes FIFO
- Direct connect to 7000 RS-485 network

### **COM3**

- RS-232: TXD3, RXD3, GND
- Communication speed: 115200 max.

### **COM4**

- RS-232: TXD4, RXD4, GND
- Communication speed: 115200 max.
- The standard I/O for 7188(D) (115200,N,8,1)
- Can be used as general purpose RS-232

### **Display**

- 7-segment LED: 5-digit (for 7188D)

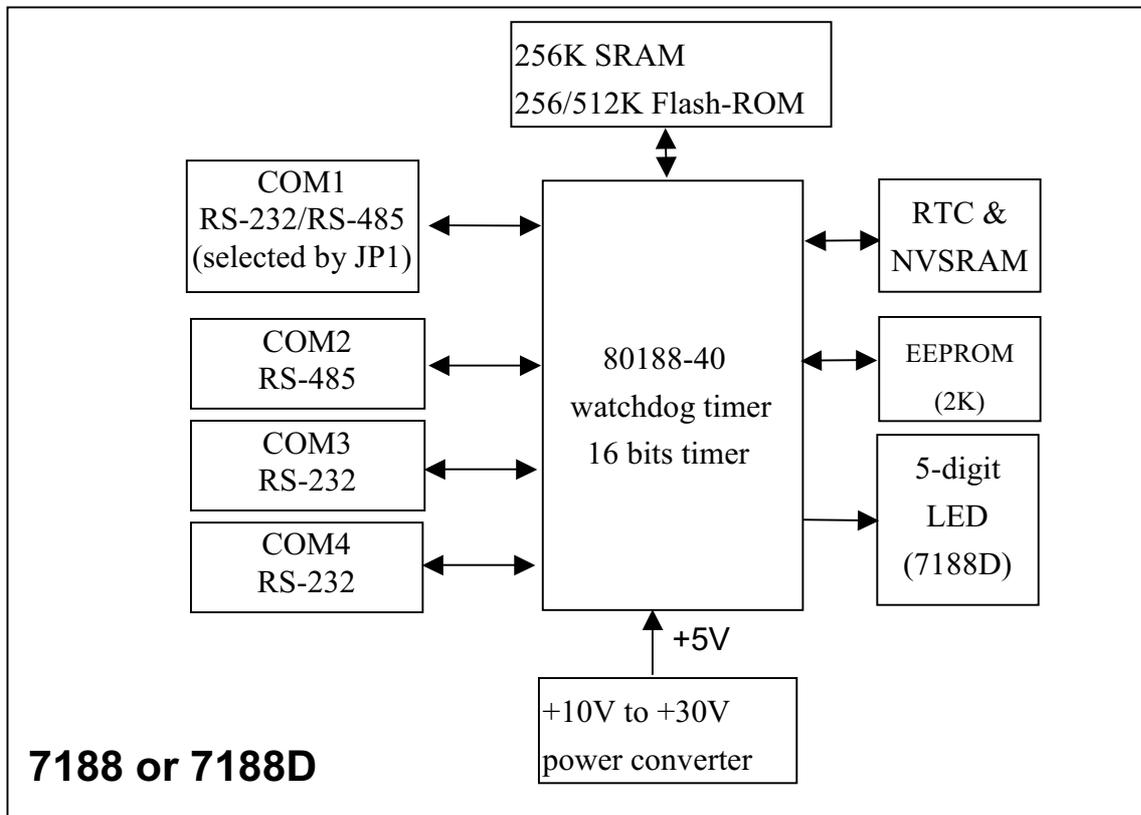
### **Power**

- Power requirements: 10 to 30VDC(non-regulated)
- Power consumption: 2.0W for 7188  
3.0W for 7188D

### **General environment**

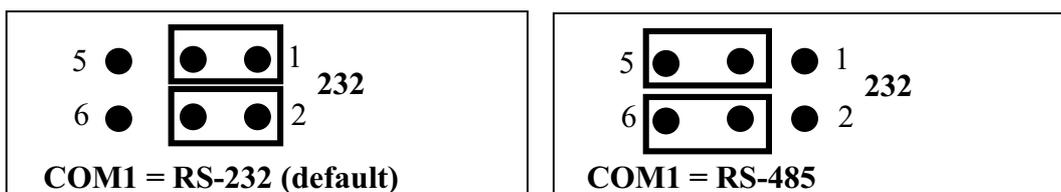
- Operating temperature: -20°C to +75°C
- Storage temperature:-40°C to +85°C
- Humidity: 0 to 90 %

## 1.5 Block Diagram of 7188



## 1.6 Jumper Setting

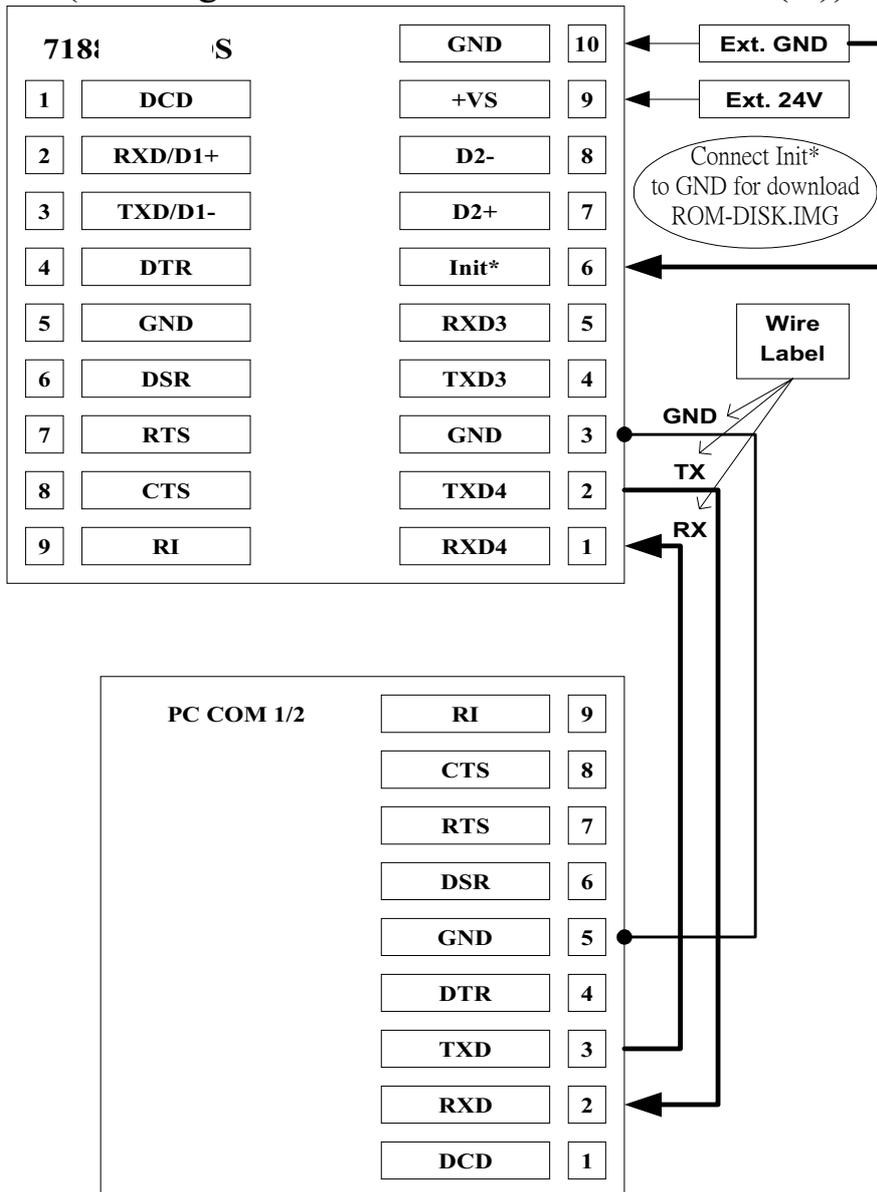
The JP1 setting is given as follows.



# 1.7 Wiring Diagrams for Application

## 1.7.1 Connect COM4 to PC

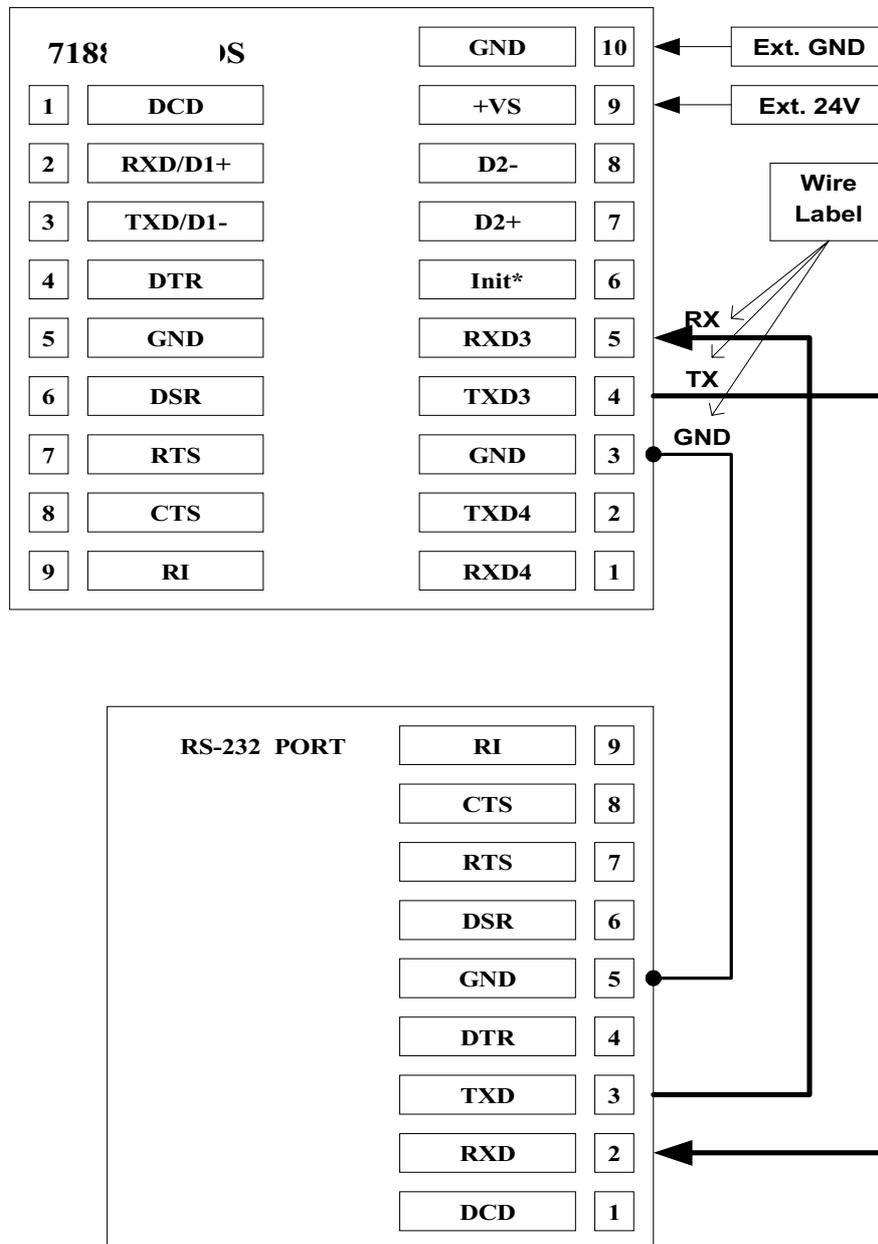
(for Program Download from PC to 7188(D))



**Note:**

- There are 3 wires in the download cable
- Connect the wire-1, label RX, to pin-1 of 7188(D)
- Connect the wire-2, label TX, to pin-2 of 7188(D)
- Connect the wire-3, label GND, to pin-3 of 7188(D)
- Connect the DB-9 of download cable to PC COM-1/2 port

## 1.7.2 Connect COM3 to 3-wire RS-232 Device

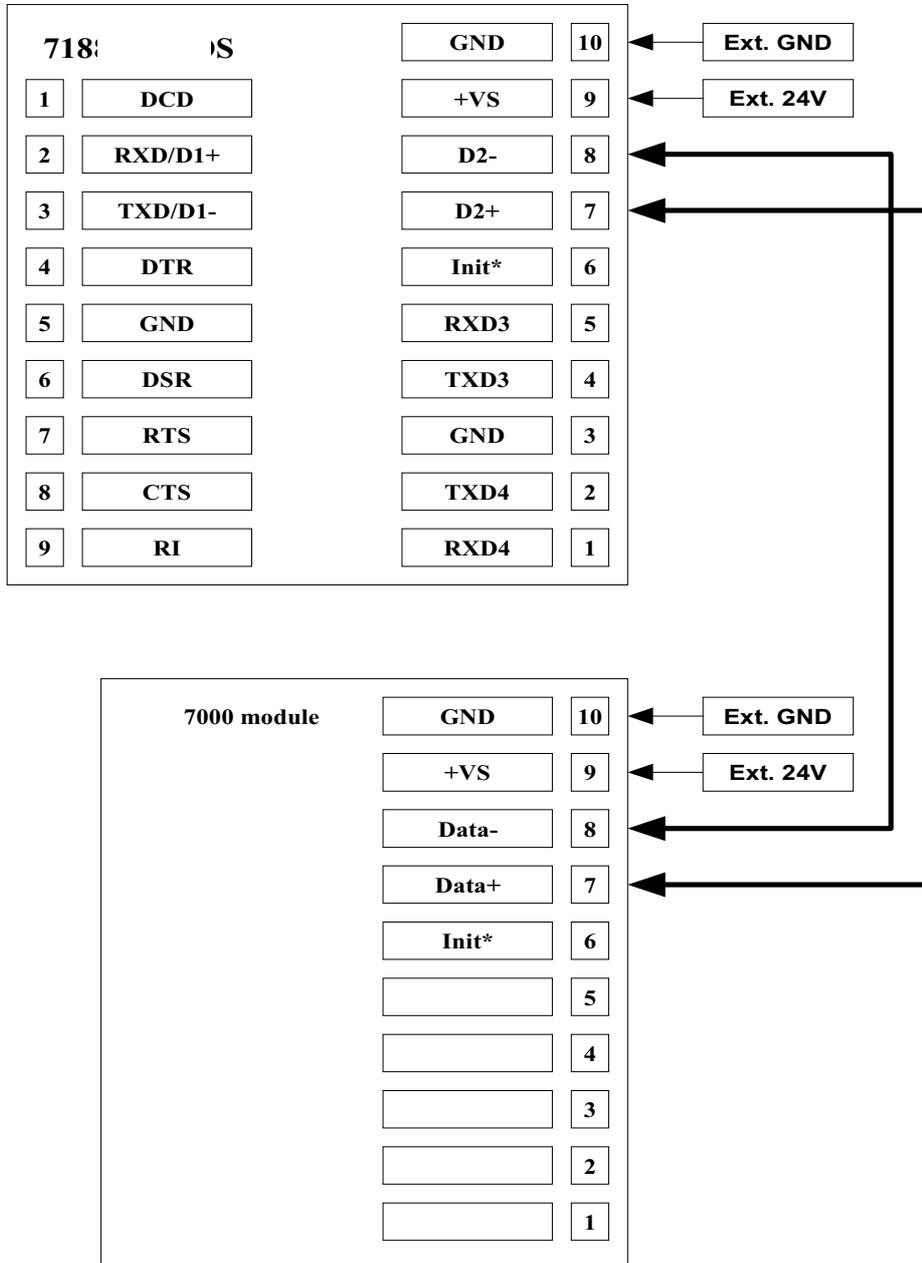


**Note:**

- There are 3 wires in the download cable
- Connect the wire-1, label RX, to pin-5 of 7188(D)
- Connect the wire-2, label TX, to pin-4 of 7188(D)
- Connect the wire-3, label GND, to pin-3 of 7188(D)
- Connect the DB-9 of download cable to RS-232 COM port

### 1.7.3 Connect COM2 to 2-wire RS-485 Device

---

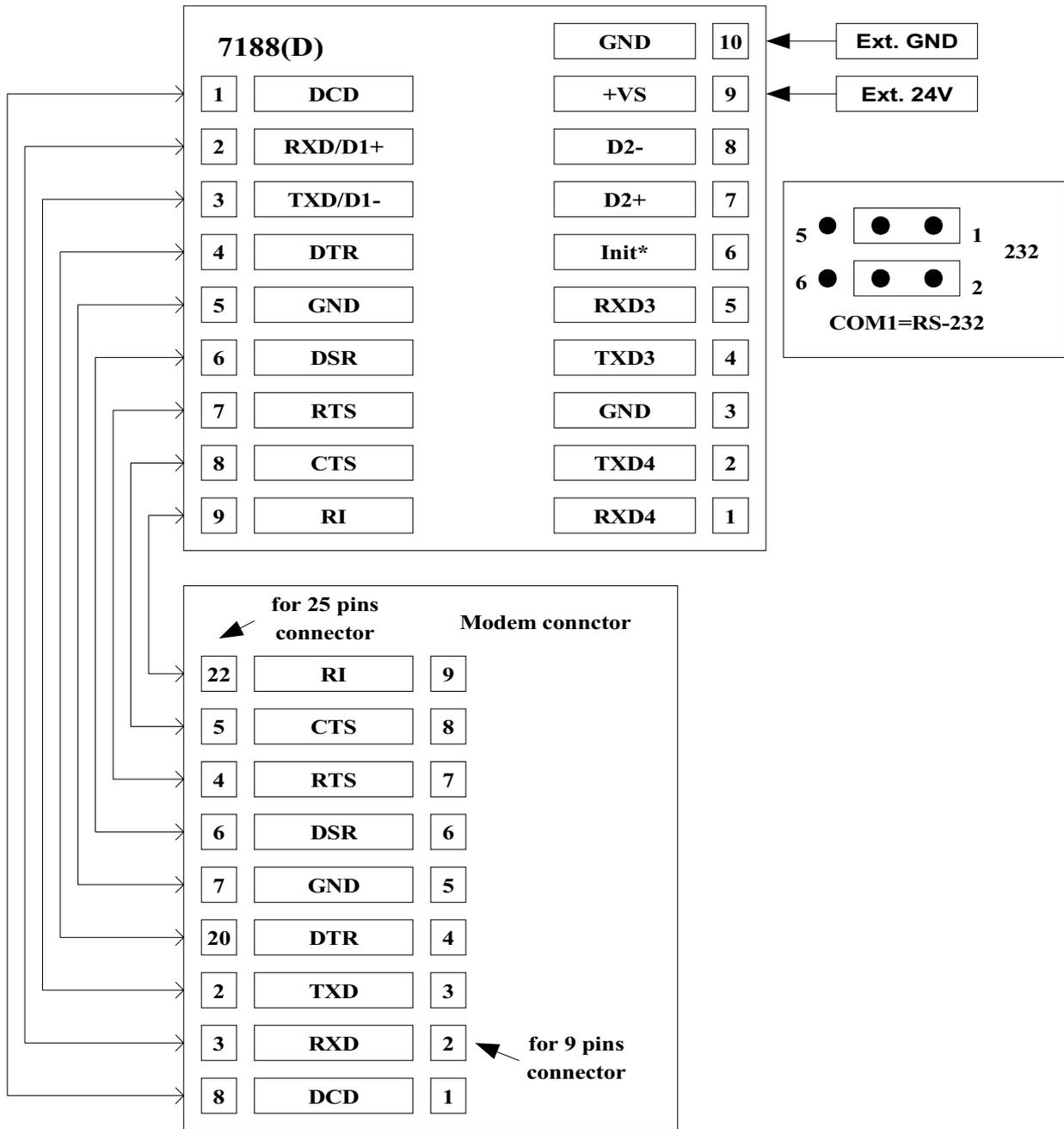


**Note:**

- The 7188(D) can directly drive 256 sets of 7000 modules without repeater
- The 7188(D) can connect to the RS-485 repeater 7510 for long distance applications.
- The 7188(D) can drive any 2-wire RS-485 network



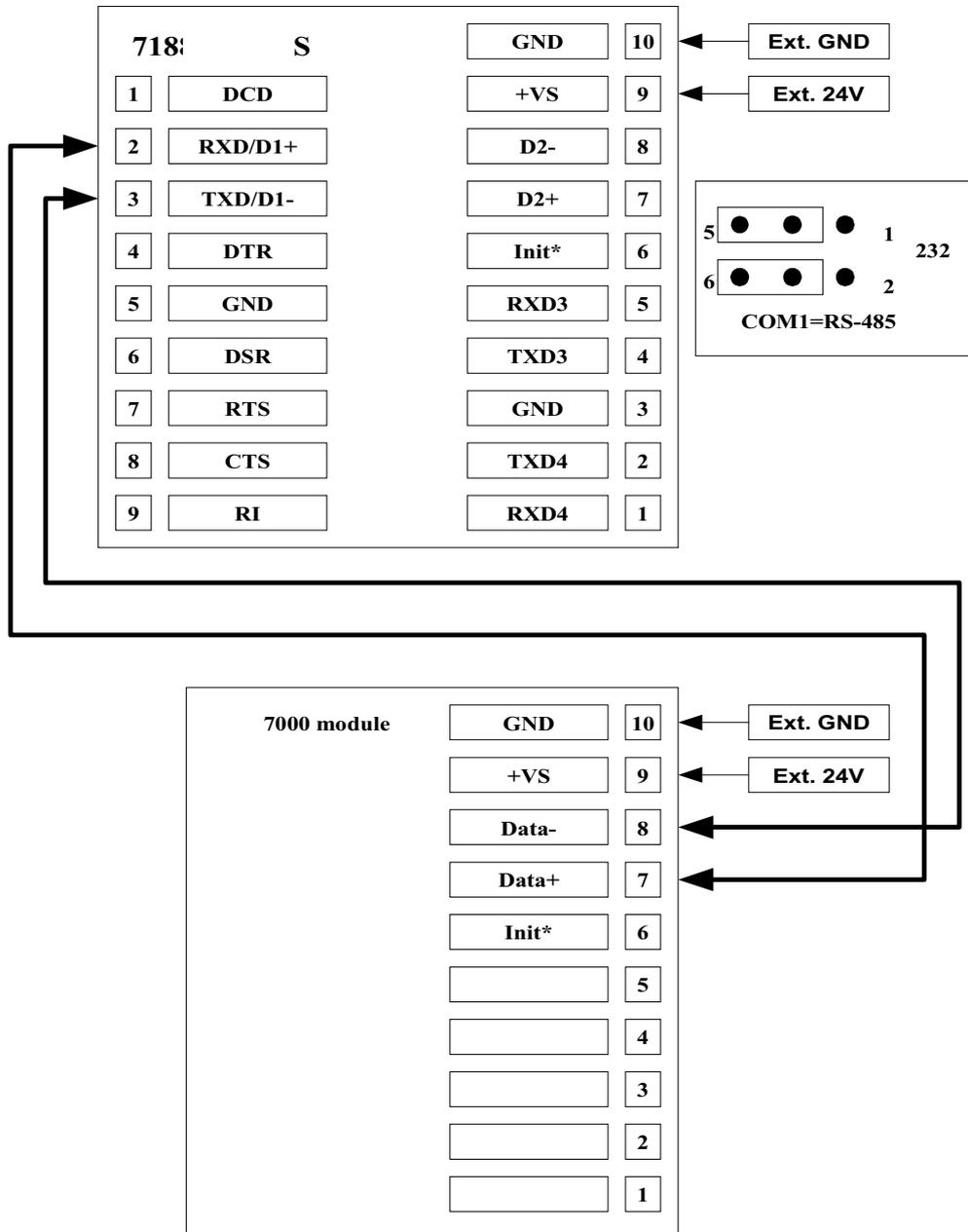
### 1.7.5 Connect COM1 to 9-wire Modem



**Note:**

- The COM1 pin assignment of 7188(D) is the same as PC.

### 1.7.6 Connect COM1 to two-wire RS-485 Device



**Note:**

- The 7188(D) can directly drive 256 set of 7000-modules without repeater
- **The 7188(D) can connect to the RS-485 repeater 7510 for safety & long distance applications. (Refer to Sec. 1.1 for more information)**
- The 7188(D) can drive any 2-wire RS-485 network

## **1.8 Comparison Table**

---

	<b>7188(D)</b>	<b>7188EX(D)</b>
<b>Module name</b>	<b>Embedded Controller</b>	<b>Internet Communication Controller</b>
<b>CPU</b>	<b>AMD 80188, 40M</b>	<b>AMD 80188, 40M</b>
<b>RAM</b>	<b>256K</b>	<b>512K</b>
<b>Flash ROM</b>	<b>512K</b>	<b>512K</b>
<b>Watch Dog CKT</b>	<b>Yes</b>	<b>Yes</b>
<b>RTC</b>	<b>Yes</b>	<b>Yes</b>
<b>EEPROM</b>	<b>2K bytes</b>	<b>2K bytes</b>
<b>Hardware Serial number</b>	<b>No</b>	<b>Yes</b>
<b>I/O expansion bus</b>	<b>No</b>	<b>Yes</b>
<b>COM1</b>	<b>RS-232 or RS-485</b>	<b>RS-232, 3-wire</b>
<b>COM2</b>	<b>RS-485, non-isolated</b>	<b>RS-485, non-isolated, self-tuner ASIC inside</b>
<b>COM3</b>	<b>RS-232</b>	<b>No</b>
<b>COM4</b>	<b>RS-232</b>	<b>No</b>
<b>7-Seg LED</b>	<b>5-digit for 7188D</b>	<b>5-digit for 7188EXD</b>
<b>Ethernet 10M</b>	<b>No</b>	<b>Yes</b>
<b>OS</b>	<b>MiniOS7</b>	<b>MiniOS7</b>
<b>Program Download</b>	<b>Yes</b>	<b>Yes</b>

- **7188(D): Embedded Controller**
- **7188XA/XB/XC(D): Expandable Controller**
- **7521/22/23/25/27(D): Addressable Communication Controller**
- **7188E1/2/3/4/5/8(D): Internet Communication Controller**
- **7188E2X/EX/EA(D): Embedded Internet/Ethernet Controller**
- **8000 Series: Compact Distributed Embedded Controller**
- **7000 Series: Network Data Acquisition & Control Modules**

## 2. Operation Principle

---

### 2.1 System Mapping

---

Device	Address mapping
Flash ROM	From C000(0r 8000):0000 to F000:FFFF
SRAM	From 0000:0000 to 3000:FFFF
COM1 BASE	0x200
COM2 BASE	0x100
COM3	FF80 to FF88
COM4	FF10 to FF18

Interrupt No.	Interrupt mapping
0	Divided by zero
1	Trace
2	NMI
3	Break point
4	Detected overflow exception
5	Array bounds exception
6	Unused opcode exception
7	ESC opcode exception
8	Timer 0
9	Reserved
0A	DMA-0
0B	DMA-1
0C	COM 1
0D	COM 2
0E-10	Reserved
11	COM 4
12	Timer 1
13	Timer 2
14	COM 3

**Four independent COM port:**

Port	Description
COM1	<ul style="list-style-type: none"><li>• For general purpose 9-wire RS-232 application</li><li>• For general purpose 3-wire RS-232 application</li><li>• For general purpose 2-wire RS-485 application</li><li>• For remote control application (by Modem)</li><li>• 16C550 compatible</li></ul>
COM2	<ul style="list-style-type: none"><li>• Direct control 7000 series modules</li><li>• For general purpose 2-wire RS-485 application</li><li>• 16C550 compatible</li></ul>
COM3	<ul style="list-style-type: none"><li>• For general purpose 3-wire RS-232 application</li><li>• UART of AMD-80188 compatible</li></ul>
COM4	<ul style="list-style-type: none"><li>• For program download</li><li>• For program debug</li><li>• For general purpose 3-wire RS-232 application</li><li>• UART of AMD-80188 compatible</li></ul>

**WARNING 1: The COM2 is not isolated to CPU. If there is large noise in the RS-485 network, the 7188 may be damaged. It is recommended to add one 7510 between the COM2 & external RS-485 network for harsh environment. Refer to Sec. 1.6.4 for more information.**

**WARNING 2: The COM1 can be used as RS-485 port. If the COM1 is used as RS-485, it is recommended to add one 7510 between the COM1 & external RS-485 network for harsh environment. Refer to Sec. 1.6.4 for more information.**

**NOTE:**

- The 7510 can be used to isolate the 7188(D) from the noisy RS-485 network.
- The 7510 can be used to extend the RS-485 network distance to more than 1.2Km.
- The 7510 can be used to extend the 7000 modules to more than 256 modules.

## 2.2 Use COM4 for Debug Program

---

The COM4 of 7188(D) has three major functions.

- The first function is to download program from PC.
- The second function is to link PC for program debug.
- The last function is to be used as a general-purpose COM port.

When the 7188(D) was power on, it will initialize the COM4 to the following configuration:

- **Start-bit=1, data-bit=8, stop-bit=1, no parity**
- **Baud rate = 115200**

Then the 7188(D) will check the state of INIT\*-pin. If the INIT\*-pin is short to GND, 7188(D) will send the version number of MiniOS7 to COM4 & enter console mode for user's download/debug program. If the INIT\*-pin is open, 7188(D) will search AUTOEXEC.BAT, if AUTOEXEC.BAT is present, 7188(D) will execute AUTOEXEC.BAT. If no AUTOEXEC.BAT, 7188(D) will enter condole mode for user's download/debug program.

After the power-on stage, the 7188(D) will use the COM4 as its standard input/output. So the standard output of 7188(D) will be shown in PC's monitor. If the user press any key in PC's keyboard, this key will be sent to 7188(D) as standard input. Therefore the keyboard & monitor of PC can be used as standard input & output of 7188(D) as following:

- Use 7188X.EXE or 7188XW.EXE as a bridge between 7188(D) & PC
- Run 7188X.EXE or 7188XW.EXE in PC to setup this bridge
- **Keyboard of PC → standard input of 7188(D)**
- **Monitor of PC → standard output of 7188(D)**

In this way, the 7188(D) can read some data from keyboard & show some information in monitor. So the program debug will become more easy & effective.

**Note: 7188X.EXE & 7188XW.EXE is given in the companion CD.**

## **2.3 Use COM4 to Download Program**

---

Refer to Sec. 1.1.3 & Sec. 1.1.4

## **2.4 Use COM4 as a COM Port**

---

The user can use COM4 as a general purpose RS-232 port as following:

- Download user's program & autoexec.bat to 7188(D) first.
- Power off 7188(D) & remove the download cable from PC.
- Disconnect the INIT\*-pin from GND-pin of 7188(D) if they are connected
- Power on 7188(D) (no standard input, no standard output, no debug information)
- Install the download cable between new RS-232 device & COM4 of 7188(D)
- Initialize the COM4 to new configuration.
- The COM4 of 7188(D) is a general purpose RS-232 port now.

## **2.5 Use COM3 for RS-232 Applications**

---

The COM3 is a 3-wire RS-232 port. It includes 3 pins as following:

- GND: signal ground, shared with COM4
- TXD3: transmit data to external RS-232 device
- RXD3: receive data from external RS-232 device
- Refer to **Sec. 1.7.2** for wire connection

## 2.6 Use COM2 for RS-485 Applications

---

The COM2 is a 2-wire RS-485 port. It includes 2 pins as following:

- D2+: connect to DATA+ of RS-485 network
- D2-: connect to DATA- of RS-485 network

The COM2 is designed to directly drive 7000 series modules. It is a half-duplex 2-wire RS-485 network. The direction control in 2-wire RS-485 network is very important. It is recommended to control the direction of COM2 as following:

- Initiate the direction of COM2 to RECEIVE state
  - Keep the RECEIVE state until 7188(D) has to transmit command
  - If 7188(D) has to transmit command
    1. set the direction of COM2 to TRANSMIT state
    2. transmit the command string out
    3. **wait until all command chars are transmitted over to the RS-485 network (not to the FIFO)**
    4. **set the direction of COM2 to RECEIVE state**
    5. receive the response string
    6. keep in the RECEIVE state until next transmit
- Refer to DEMO3 for more information

The working steps for 7000 related applications are given as follows:

1. 7188(D) send command string to 7000 modules
2. Destination 7000 module execute this command
3. Destination 7000 module **delay 1 byte for settling time**
4. Destination 7000 module echo back the result string to 7188(D)

**The delay time in step 3 is only 1 byte, the 7188(D) has to change the direction from TRANSMIT to RECEIVE before the delay time is up.**

NOTE: the COM2 can not be used in the full-duplex 4-wire RS-485 application.

## **2.7 Use COM1 for RS-232 Applications**

---

The features of COM1 are given as following:

- Pin assignment is compatible to RS-232 DB-9 connector of PC.
- Programming is 16C550 compatible
- I/O base address is different from of COM1 PC
- Interrupt handling sequence is different from PC
- Can be used as 9-wire RS-232 signal
- Can be used as 3-wire RS-232 signal
- Can be used as 2-wire RS-485 signal
- RS-232/RS-485 is selected by JP1
- Refer to Sec. 1.6.5 for wire connection & jumper setting

The I/O base address & the interrupt handling sequence is different from PC. Therefore if user download the PC's RS-232 application program into 7188(D), it will not work.

## **2.8 Use COM1 for RS-485 Applications**

---

The COM1 can be used as a 2-wire RS-485 port. It includes 2 pins as follows:

- RXD/D2+: connect to DATA+ of RS-485 network
- TXD/D2-: connect to DATA- of RS-485 network

The COM1 is very similar to COM2 in RS-485 applications. Refer to **Sec. 2.6** for more information.

## **2.9 COM port Comparison: 7188(D) & PC**

---

The COM ports of 7188(D) are given as following:

COM port	Hardware
COM1	16C550
COM2	16C550
COM3	80188's on-chip UART-0
COM4	80188's on-chip UART-1

The programming of 16C550 is very different to 80188's UART. The interrupt handling of 80188 is also very different to PC's 8259. **Therefore if user download the PC's RS-232 application program into 7188(D), it will not work.**

The software driver of 7188(D) is a interrupt driven library which provide 1K QUEUE buffer for every COM port. The software is well-designed & easy to use.

The software driver provide the same interface for all these 4 COM ports. The user can use these COM port in the same way without any difficulty.

## **2.10 How To Use COM1/2/3/4**

---

The 7188(D) has 4 communication ports.

- COM1 can be used as an RS-232 or RS-485 port.
- COM2 is an RS-485 port.
- COM3 is an RS-232 port.
- COM4 is an RS-232 port.

Before using com-port, the user must call “**InstallCom**” (or **InstallCom1/2/3/4**) to install the driver for the comport.

Before exiting the program, the user must call “**RestoreCom**” (or **RestoreCom1/2/3/4**) to un-install the driver.

After calling “**InstallCom**”, the user should use “**IsCom**” to check if any data coming from com-port and use “**ReadCom**” to read data from input buffer or use “**ToCom**” to send data to com-port. For example, to echo the data back to the com-port, the code is shown below.

```
int port=4; // for use COM4
int quit=0,data;

InstallCom(port,57600L,8,0,1);
while(!quit){
    if(IsCom(port)) {
        data=ReadCom(port);
        ToCom(data);
        If(data=='q') quit=1; // if receive 'q', exit the program
    }
}
RestoreCom(port)
```

Use the variable port to change from using COM4 to using COM3. Just change `port=4;` to `port=3.` If the program is fixed to use COM4, the code can be as follows.

```
int quit=0,data;

InstallCom4(115200L,8,0);
while(!quit){
    if(IsCom4()) { // check if has data in
        data=ReadCom4(); // read data from input buffer
        ToCom4(data);    // send data out
        If(data=='q') quit=1; // if data='q', then quit
    }
}
RestoreCom4();
```

The library of 7188(D) also support functions like **printf** of standard C library for user to do a formatted output.

**printCom** is for all comports, **printCom1/2/3/4** are for individual port. Before use printCom must call InstallCom. The code can be as follows.

```
int port=4;
int i;

InstallCom(port,57600,8,0,1);
for(i=0; i<10;i++){
    printCom(port,"Test %d\n\r",i);
}
RestoreCom(port);
```

## *(2) Operation Principle*

---

When using RS-485 (COM1 or COM2), the user must control the direction of RS-485 bus. At the moment the 7188 module do not build in “self-tuner”, therefore the user need to control the direction of data flow.

We will add a “self-tuner” ASIC to the new version 7188 module in the near future. The “ self-tuner” ASIC will control the direction of RS-485 bus itself. The user doesn't have to take care of the direction of RS-485 bus.

The default direction of RS-485 bus is in receiving mode.

When the 7188 module want to transmit data, the bus must be set to transmit mode (call **Set485DirToTransmit**).

After sending data, the bus must be set back to receive mode (call **Set485DirToReceive**).

Before calling **Set485DirToReceive**, the user must be sure all the data has been sent out by hardware (call **WaitTransmitOver**). The code for sending data to COM2 (RS-485) is as below.

```
int port=2;
int i;
char data[5]="$01M\r";

InstallCom(port,9600,8,0,1);
Set485DirToTransmit(port);
for(i=0;i<5;i++) ToCom(port,data[i]);
WaitTransmitOver(port);
Set485DirToReceive(port);
RestoreCom(port);
```

The **SendCmdTo7000** will auto control the direction of RS-485 bus.

## 2.11 Use Red LED & 7-SEG LED Display

---

The red LED of 7188(D) can be turned-ON or turned-OFF by software program. The 5 digits of seven-segment LED are also programmable. **The 5-digit LED is very useful in real world applications.** It can be used to replace the monitor & touch-screen in many applications.

7188	Red LED
7188D	Red LED & 5-digit LED

The 7188D has a 5-Digit 7-segment with dot. The LED is named as 1,2,3,4,5 from left to right. Before using it, the user must call **Init5DigitLed first**, and then call Show5DigitLed to show digit on it. For example:

```
Init5DigitLed();  
Show5DigitLed(1,7);  
Show5DigitLed(2,1);  
Show5DigitLed(3,8);  
Show5DigitLed(4,8);  
Show5DigitLed(5,13); // 'd' is 13
```

will show “7188d” on it.

If want to show the ‘.’(DOT), such as show “12.345” just call:

```
Show5DigitLed(1,1);  
Show5DigitLedWithDot(2,2); // will show “2.”  
Show5DigitLed(3,3);  
Show5DigitLed(4,4);  
Show5DigitLed(5,5);
```

## **2.12 How To Use Flash Memory**

---

The 7188(D) module has 256K or 512K flash memory size. It contains the MiniOS7, and the ROM-DISK. MiniOS7 occupy segment 0xF000 (the last segment). The ROM-DISK occupies the first segment, such as 0xC000 for 256K memory or 0x9000 for 512K memory. So 7188 module will occupy at least two segments. The user can use non-used segments to store data. With 256K memory, there are maximum 128K memory left to store data. With 512K memory, there are maximum 384K memory left to store data.

Flash memory can be written from 1 to 0 only and can't be written from 0 to 1., The only way to change the data bit from 0 to 1 is to call " EraseFlash " to erase a block Flash Memory (64K bytes). The user should decide whether to write or to erase it.

If the original data bit is 0, and want to write it to 1(for example: current value is 0x55,want write to 0x57, so want write bit 1 from 0 to 1).

If the user want to write an integer to Flash memory on segment 0xD000, offset 0x1234, the code can be as follows.

```
int data=0xAA55,data2;  
char *dataptr;  
int *dataptr2;  
  
dataptr=(char *)&data;  
FlashWrite(0xd000,0x1234,*dataptr++);  
FlashWrite(0xd000,0x1235,*dataptr);  
// read data from Flash memory method 1  
dataptr=(char *)&data2;  
*dataptr=FlashRead(0xd000,0x1234);  
*(dataptr+1)=FlashRead(0xd000,0x1235);
```

```
// read data from Flash memory method 2  
dataptr2=(int far *)_MK_FP(0xd000,0x1234);  
data2=*dataptr2;  
// or just writ:  
data2=*(int far *)_MK_FP(0xd000,0x1234);
```

Reading data from Flash Memory is somewhat like reading data from SRAM. The user should make a far pointer point to the memory location first, and then use the pointer to access the memory. Before writing data to Flash Memory, the user must call “**FlashWrite**” first, and then check if the data can be written to it or not. After calling “**EraseFlash**”, any data can be written to that segment at first time.

## **2.13 Use RTC & NVSRAM**

---

The RTC & NVSRAM are located in the same chip. There is a Li-battery to backup the RTC & NVSRAM for 10 years. The features of RTC are given as following:

- **Year 2000 Compliance**
- BIOS support RTC time & date
- MiniOS7 support RTC time & date
- seconds, minutes, hours, date of the month
- month, day of week, year, Leap year valid up to 2079
- NVSRAM: 31 bytes

The NVSRAM can be read/write any number of times. The features of NVSRAM are given as following:

- Data Valid : 10 years
- Read/write cycles: without limit
- Total 31 bytes

The user can use *ReadNVRAM* to read one byte data from NVRAM and use *WriteNVRAM* to write one byte data to NVRAM. If the user want to write data to NVRAM address 0, the program is shown below.

```
int data=0x55,data2;  
WriteNVRAM(0,data);  
data2=ReadNVRAM(0); // now data2=data=0x55
```

If the user want to write an integer (two bytes) to NVRAM, the program is shown below.

```
int data=0xAA55,data2;  
char *dataptr=(char *)&data;  
WriteNVRAM(0,*dataptr); // write the low byte  
WriteNVRAM(1,*dataptr+1 ); // write the high byte  
dataptr=(char *)&data2;  
*dataptr=ReadNVRAM(0); // read the low byte  
*dataptr+1=ReadNVRAM(1); // read the high byte,  
// now data2=0xAA55
```

## 2.14 Use EEPROM

---

The EEPROM is designed to store the data which are not changed very frequently. These data are given as following:

- Module ID, configuration setting
- COM port configuration setting
- Small data base

**The erase/write cycle of EEPROM is limited**, the user should not change the EEPROM frequently for testing. The EEPROM can be **erase/write in single byte**, so it is very useful in real world applications.

The 7188(D) module has 2K bytes EEPROM. There are 8 blocks EEPROM and each block has 256 bytes. Therefore totally it has 2048 bytes EEPROM. Normally EEPROM is in protection mode. In this mode, the user can't write any data into EEPROM. Must call "EnableEEP" to UN-protect it before writing data. For example, if the user want to write data to EEPROM block 1, address 10, he should call EnableEEP first. The program is shown below.

```
int data=0x55,data2;
```

```
EnableEEP();
```

```
WriteEEP(1,10,data);
```

```
ProtectEEP();
```

```
Data2=ReadEEP(1,10); // now data2=data=0x55
```

If the user want to write an integer to EEPROM, he must call " WriteEEP " twice. Just like writing data to NVRAM.

## **2.15 Use Watchdog Timer**

---

The watchdog timer of 7188(D) is fixed at 1.6 sec. **When the 7188(D) is power on, this watchdog timer will be always enable.** If the 7188(D) do not reset this watchdog timer for 1.6 second, the watchdog will RESET the 7188(D).

The MiniOS7 of 7188(D) will automatically reset the watchdog after first power on. The user program can call the software driver to tell MiniOS7 stop to reset the watchdog timer, then the user program must reset the watchdog timer. If the user program does not reset the watchdog timer for 1.6 second, the watchdog timer will RESET the 7188(D).

The user program can tell the MiniOS7 to reset the watchdog timer again, then the user program can stop & return to the prompt of MiniOS7.

The WatchDog Timer is default fixed on 1.6 seconds. The user can use “**EnableWDT**” to enable it or use “**DisableWDT**” to disable it. After watchdog is enabled, the program should call “**RefreshWDT**” before the timer count up to 1.6 seconds, otherwise it will reset 7188(D) module. The code can be as follows.

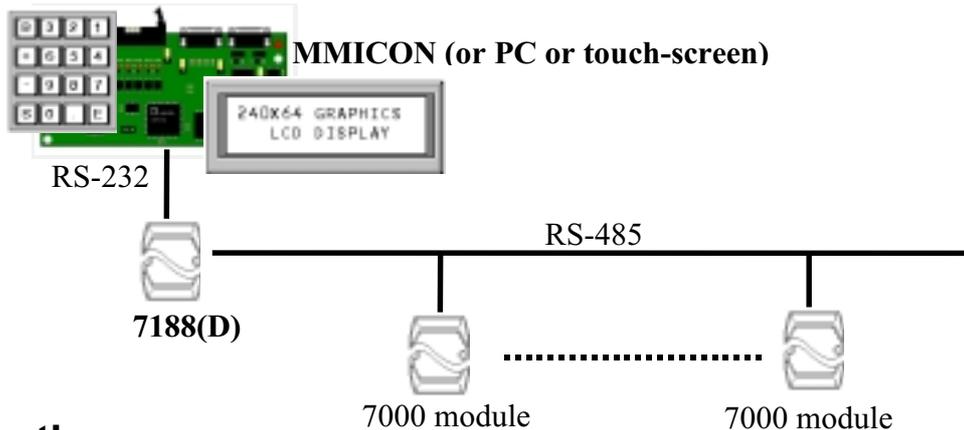
```
EnableWDT();  
while(!quit){  
    RefreshWDT();  
    User_function();  
}  
DisableWDT();
```

The function **IsResetByWatchDogTimer** is used to check if the 7188 module is reset by WatchDog Timer. This function must be inserted on the beginning of program. The code can be as follows.

```
main()
{
  if(IsResetByWatchDogTimer()){
    // here do something to check the system
  }
  quit=0;
  EnableWDT();
  while(!quit){
    RefreshWDT();
    User_function();
  }
  DisableWDT();
}
```

### 3. Applications

#### 3.1 Embedded Controller



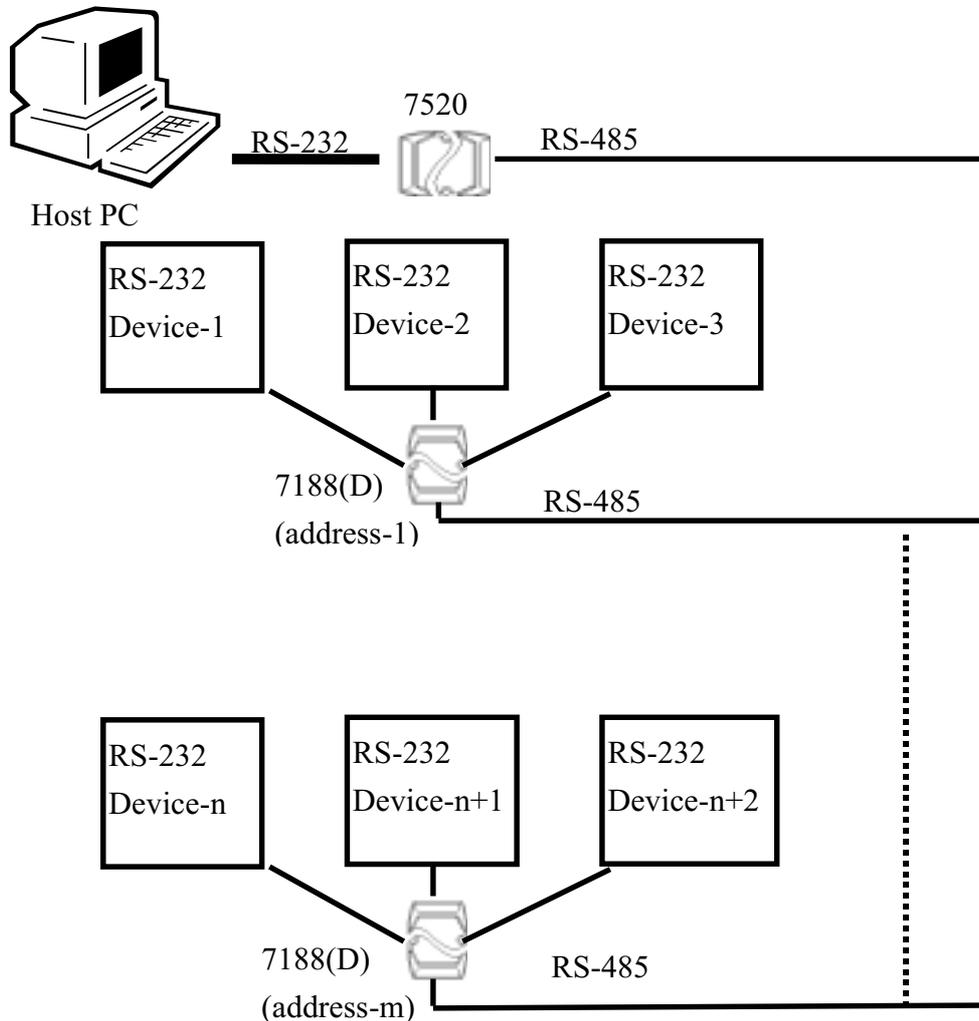
**Applications:**

- **PC-Based controller replacement**
- **PLC replacement**
- **Special controller replacement**

The 7188(D) can be use as embedded controller for general application. Therefore it can be used to replace PC, PLC or special controller.

Programming Tool	TC or BC++ or MSC for a DOS real mode execution program.
Debug Tool	<ul style="list-style-type: none"><li>● Program via standard input or standard output (keyboard &amp; monitor of PC)</li></ul>
Man Machine Interface	<ul style="list-style-type: none"><li>● MMICON</li><li>● Keyboard &amp; monitor of PC</li><li>● Touch Screen (RS-232 or RS-485 interface)</li></ul>
program	<ul style="list-style-type: none"><li>● Stored in Flash ROM</li></ul>
Input/Output	<ul style="list-style-type: none"><li>● 7000 series modules</li><li>● can directly control 256 modules = thousands of I/O points</li></ul>

## 3.2 Addressable RS-232 Protocol Converter



### Applications:

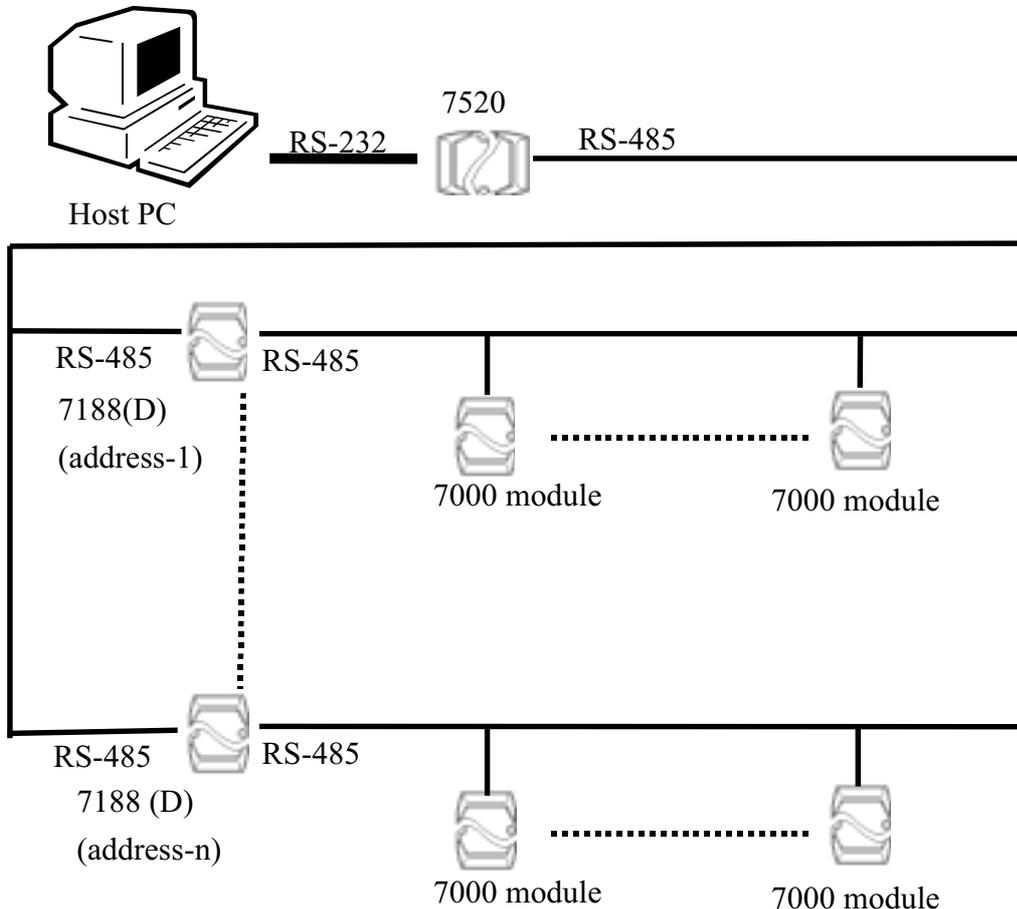
- **Multiple RS-232 devices networking**

The key points are given as following:

- Connect RS-232 devices to COM1/3/4 of 7188(D)
  - Connect COM2 of 7188 to RS-485 network
  - Store the module address into EEPROM(inside of 7188(D))
  - 7188(D) communicate to its local RS-232 devices & store data
- Host PC communicate to every 7188(D) for reading data from these RS-232 devices

### 3.3 Local Real Time Controller

---



The 7000 series modules are slave devices, the control program are implemented in host PC. The working steps are given as following:

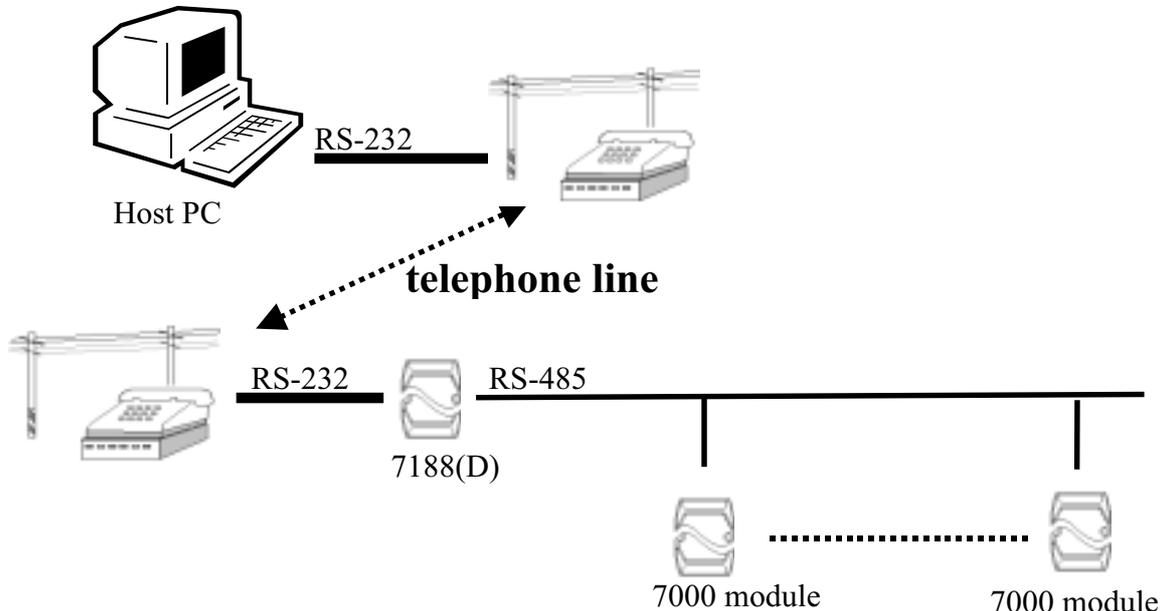
1. PC send out command to 7000 modules for input data.
2. PC analyze these input data & generate some output data
3. PC send out command to 7000 modules for output data

If there are hundred of 7000 modules, the PC will take a long time to analyze & control these modules.

The control program can be implemented in local 7188(D). The PC only has to send control arguments to 7188(D), the 7188(D) will control the local 7000 modules based on these control arguments. In this way, the PC can handle thousands of 7000 modules with 7188(D).

Some control functions are **timing critical**. The local 7188(D) can real time handle these functions without PC.

## 3.4 Remote Local Controller



The control program is implemented in local 7188(D). The 7188(D) will direct control the 7000 modules based on these control arguments.

The PC can access the remote 7188(D) for the following items:

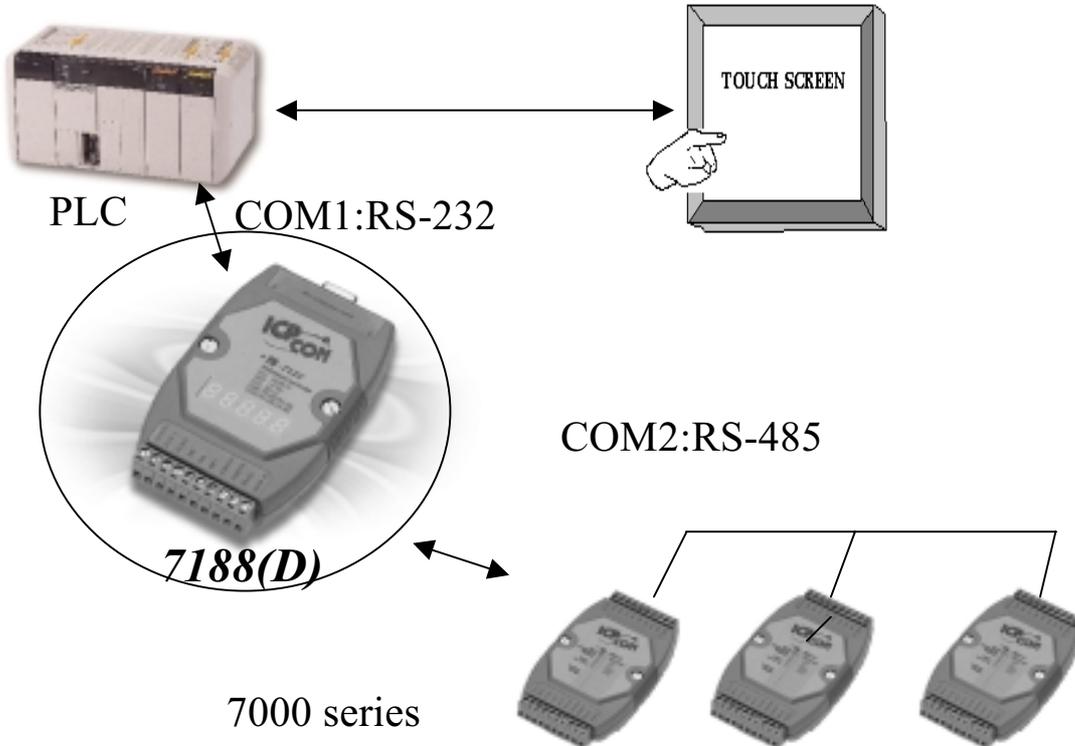
- query & record the remote system status
- download the control arguments to remote 7188(D)

The remote 7188(D) can communicate to PC for the following items:

- emergency events call back
- remote system status send back

## **3.5 I/O Extension of PLC applications**

---



There is a Man Machine Interface in most PLC. This interface is originally designed for MMI application. The 7188(D) can use this interface to build a bridge between PLC & 7000 modules.

The 7188(D) can direct read/write the internal memory of PLC. Therefore the **PLC can access the 7000 input modules** as following:

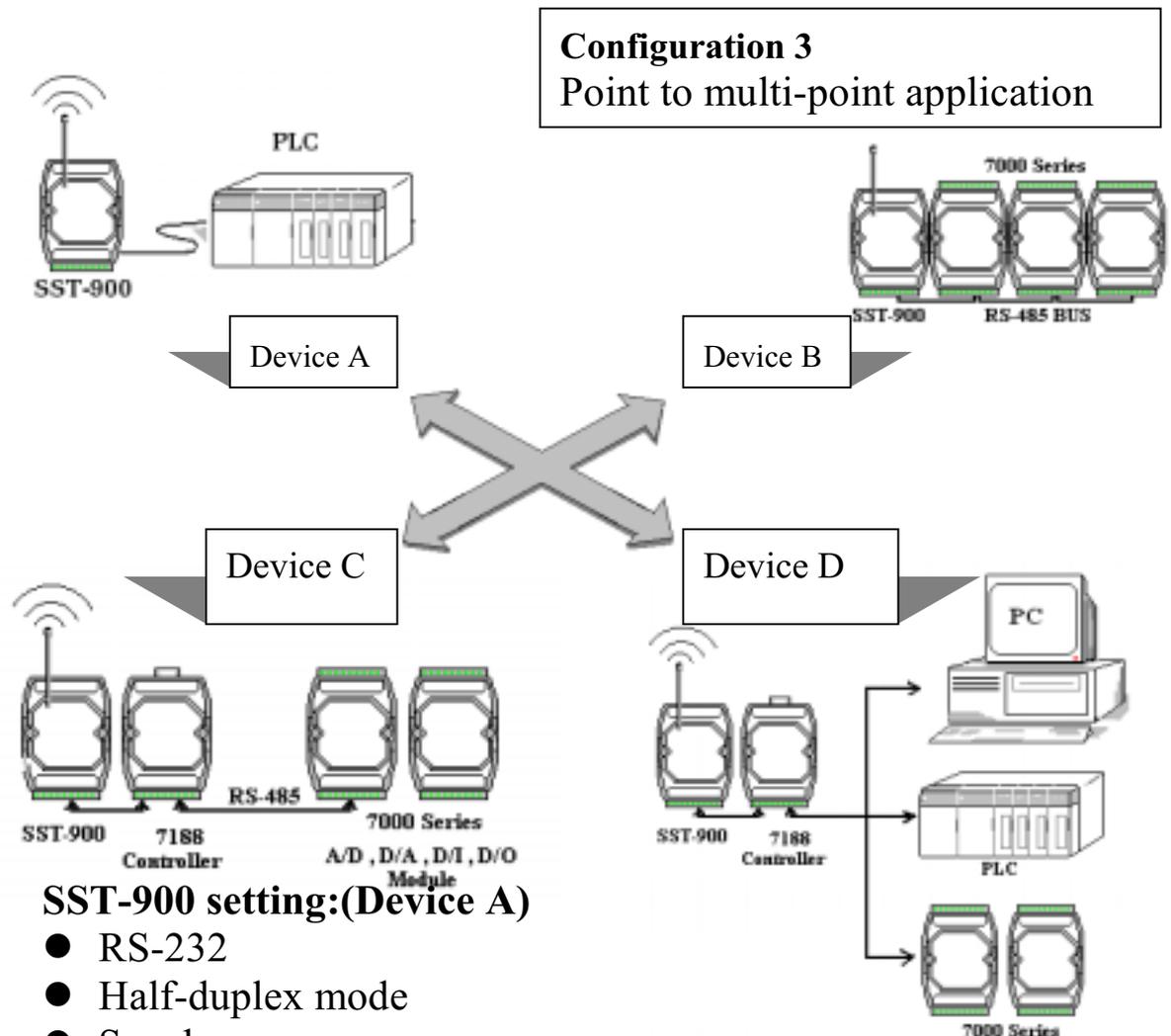
1. 7188(D) send command to 7000 input modules
2. 7188(D) write these data to PLC's internal memory
3. PLC access these data from internal memory

The **PLC can control the 7000 output modules** as following:

1. PLC write data to its internal memory
2. 7188(D) read these output data from PLC's memory
3. 7188(D) send command to 7000 output modules

In this way, the input data of 7000 modules can be shown in touch screen. Also, the user can control the output of 7000 modules in touch screen.

## 3.6 Radio Modem Applications



The 7188(D) is an embedded controller. Because the 7188(D) is programmable, **it can be used as a bridge between the SST-900 & external devices.** These devices can be PLC, controller or 7000 modules.

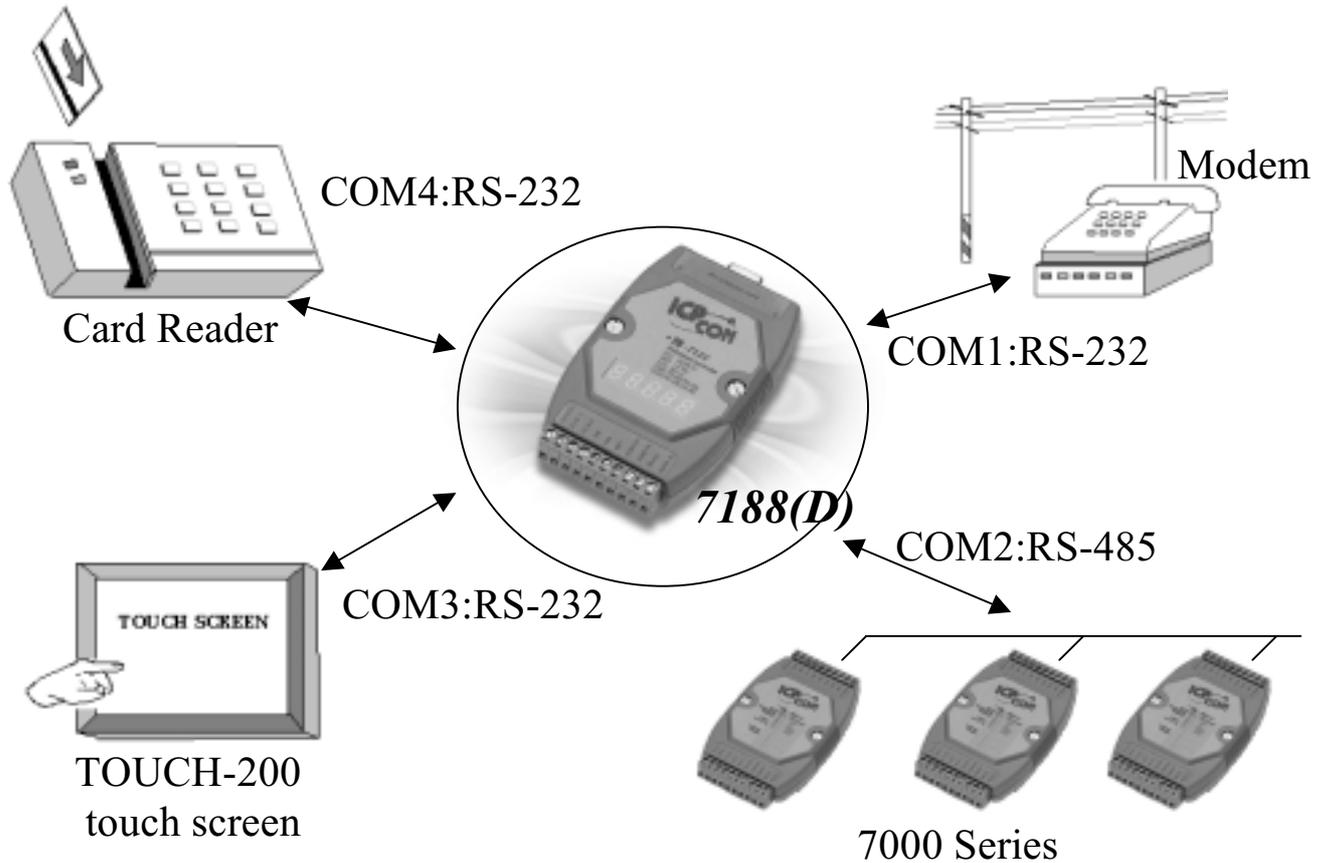
### **3.7 Use all 4 COM port Application-1**

---



COM1: PLC can access the I/O states of 7000 modules  
COM2: direct control 7000 input/output modules  
COM3: MMICON is used as the local MMI  
COM4: PC is used to monitor & record the system data.

## 3.8 Use all 4 COM port Application-2



COM1: link to remote host PC

COM2: direct control 7000 input/output modules

COM3: TOUCH-200 is used as the local MMI

COM4: Card reader is used as standard input device.

## Appendix: Specifications of TOUCH-200

Input power	24VDC +/- 5%
CE	complies with EN50081-2 and EN50082-2 standards
EMI	complies with FCC Class A
Isolation resistance	Exceed 50MW at 500VDC
Vibration endurance	10 to 25 Hz(X,Y,Z direction 2G 30minutes)
Protection structure	IP65 front panel(O ring seal)
Operating Temperature	0~45C
Operation humidity	10-90% RH
Dimensions HxWxD	204x150x75mm
Weight	Approx. 1.0kg

### Hardware Specification

Display	<b>5.7" high contrast, blue mode LCD</b>
Resolution	320(W) x 240(H) dots
Display area	120(W) x 90(H) mm
Dot size	0.33(W) x 0.33 (H) mm
Back light	CCFT (MTBF 10,000hr, w/ auto shutdown, replaceable)
Touch panel	resistive type, audible feedback on touch
Touch resolution(grid)	40(W) x 30(H)
Surface hardness	4H
Serial port	RS-232 x1 RS-485 x1 (2/4 wires auto setup)
Memory	512K bytes flash ROM
System diagnostic	power failure detection

### Functional Specification

Screen edit	by EasyBuilder screen editor (running under MS Windows 95/98)
No. of screen	256
No. of object	5370
Text string	1792 (max. capacity 52k bytes)
Bitmap graphics	288K bytes

### NOTE:

- The TOUCH-200 is a special OEM version of EasyView.
- EasyBuilder is a screen editor provided by EasyView.