7188 Series

7188 (D)/DOS Software User's Manual

7000 New Features

1. **Self Tuner Inside**
2. **Multiple Baud Rate**
3. **Multiple Data Format**
4. **Dual WatchDog Inside**
5. **True Distributed Control**
6. **High Speed & High Density I/O**

Your Powerful Tools

Create New Ideas

Create New Applications

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

Trademark

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

Table of Contents

1. INTRODUCTION 4

- 1.1 INSTALLATION 5
- 1.2 7188 LIBRARY 6
- 1.3 USING MSC OR MSVC++ 7
- 1.4 USING TC OR TC++ OR BC++ 8
- 1.5 USING QUICKBASIC 9
- 1.6 QUICK START OF 7188(D)/DOS 10
- 1.7 USING 7188.EXE 13
- 1.8 DATE OF ROM-DOS 16
- 1.9 TIME OF ROM-DOS 17
- 1.10 VDISK.SYS OF ROM-DOS 19
- 1.11 USING LD3.EXE 20
- 1.12 USING TOROM.EXE 22
- 1.13 USING INIT* 23
- 1.14 USING ROMDISK.EXE 24
- 1.15 USING ECHO485.EXE 25
- 1.16 USING ECHOCOM.EXE 26

2. OPERATION PRINCIPLE 27

- 2.1 SYSTEM LAYOUT 28
- 2.2 MINIBIOS 29
- 2.3 HOW TO DEVELOP APPLICATION PROGRAM 31
- 2.4 HOW TO DOWNLOAD FILE TO RAM-DISK 32
- 2.5 HOW TO DOWNLOAD FILE TO ROM-DISK 33
- 2.6 HOW TO USE 192(448)K ROM-DISK 34

3. C LANGUAGE LIBRARY 35

- 3.1 CONST DEFINED IN 7188.H 35
- 3.2 FUNCTION BY TYPE 36
  - *3.2.1 Functions for COM ports 36*
  - *3.2.2 Functions for LED 40*
  - *3.2.3 Functions for 5-Digit LED 40*
  - *3.2.4 Functions for data storage 41*
  - *3.2.5 Functions for WatchDog Timer 42*
  - *3.2.6 Functions for timer 43*
  - *3.2.7 Functions for stdio 44*
  - *3.2.8 MISC functions 44*

3.3 LIBRARY REFERENCE 45
3.4 C FUNCTIONS CAN'T BE USED ON 7188 46

4. DEMO PROGRAM 47

4.1 HOW TO USE NVRAM 50
4.2 HOW TO USE EEPROM 51
4.3 HOW TO USE FLASH MEMORY 52
4.4 HOW TO USE 5-DIGIT LED DISPLAY 54
4.5 HOW TO USE WATCHDOG TIMER 55
4.6 HOW TO USE COM1/2/3/4 56

1. Introduction

The 7188 module is an 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 module can be ordered as follows.

Order Item	Flash ROM size	5-digit LED	O. S.
7188/DOS	256K Bytes	No	ROM-DOS
7188D/DOS	256K Bytes	Yes	ROM-DOS
7188/DOS/512	512K Bytes	No	ROM-DOS
7188D/DOS/512	512K Bytes	Yes	ROM-DOS

What is DOS?

DOS is an acronym for Disk Operating System. Various companies have created several brands of DOS'. 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 ROM-DOS?

ROM-DOS is an operating system, which can be embedded in ROM (Read Only Memory) and can run entirely from within ROM. ROM-DOS is functionally equivalent to other brands of DOS, and can run programs that are executable under a standard DOS (which executes out of RAM). With ROM-DOS, the executable program may reside in RAM or may be placed in ROM along with ROM-DOS. **The 7188D/DOS(512) & 7188/DOS(512) are equipped with Datalight's ROM-DOS.**

1.1 Installation

The contents of companion floppy are given as follows.

- \napdos\History.txt ....**Please read this file first**
- \napdos\7188.exe ....utility of 7188/DOS run on PC
- \napdos\7188DOS.EXE ....Utilities & Demo Program
- \napdos\demo1.exe ....demo program
- \napdos\disk14.txt.............NAPDOS14 disk file
- \napdos\Install.bat.............Please run from DISK A:
- \napdos\rom-disk.img ....ROM-DISK image file
- \readme.txt........................Description file of NAPDOS

To install NAPDOS14 to your PC (for example the sub-directory c:\7188dos), using the following steps:

Step 1. c:
Step 2. cd \
Step 3. md 7188dos
Step 4. cd 7188dos
Step 5. xcopy a:\napdos\*.* c: /s /v
Step 6. 7188dos ....*** Ver 1.09 or latter need not add -d ***
install.bat will do these steps for you.

You will have the following new files or sub-directories in the directory 7188dos.

C	sub-directory for C demo programs
QB	sub-directory for QuickBASIC demo programs
DISK	sub-directory contain the files in ROM-DISK (A:) of 7188(D)/DOS module
7188UTIL	sub-directory contain some utilities for 7188 (D)/DOS module
7188D.EXE	old version of 7188.exe
ROMDISK.EXE	utility to make ROM-DISK.IMG
RDUSER.DOC	ROM-DOS user manual(a MS WORD file)

NOTE: the copyright of ROMDISK.EXE and RDUSER.DOC belongs to Datalight.

1.2 7188 library

The 7188(D)/DOS supports 7188S.lib, 7188L.lib and including file 7188.h. The user can use **TC** or **BC++** or **MSC** or **MSVC++** or **QuickBasic** to development the program for 7188(D)/DOS.

7188S.lib is for **SMALL MODEL**
7188L.lib is for **LARGE MODEL** & for Microsoft's **QuickBASIC**

1.3 Using MSC or MSVC++

- The including file is **7188.H**
- There are 2 different model library files:
  7188S.lib for SMALL MODEL
  7188L.lib for LARGE MODEL
- Support MSC 5.x to 6.x, MSVC++ 1.0 to 1.52
- When using MSVC++, be sure to generate a DOS-EXE file.
- When using floating point, be sure to use the EMULATION mode.

1.4 Using TC or TC++ or BC++

- The including file is **7188.h**
- There are 2 different model library files:
  7188S.lib for SMALL MODEL
  7188L.lib for LARGE MODEL
- Support TC 2.x , TC++ up to 3.0 and BC++ up to 5.02
- When using TC 2.0, the user should create a project file (.prj), the file is a ASCII text file, and user must put all file name that the project needed to the project file. For example:
  userprog.c
  7188S.lib
- When using TC++ or BC++ on the integrated development environment (IDE), the user should create a new project and add the needed files (userprog.c and 7188S(L).lib) to it.
- Be sure to choose the correct compiler model & CPU type (or Instruction set) on the integrated development environment (IDE). 80186 is the correct choice for 7188 module. When using floating point, the user should choose Emulation and turn off the debug information.

1.5 Using QuickBASIC

The user can use *Microsoft QuickBASIC* to write program for 7188(D)/DOS. The including file for QuickBASIC is **7188.bi. The user should also include printf.bas to control the com-ports.** Therefore the following code should be added into the BASIC program:

```
REM $INCLUDE: '7188.bi'
REM $INCLUDE: 'printf.bas'
```

The compiler option must add /o. The library must use 7188L.lib. The following steps are used to compile demo1.bas to demo1.exe.

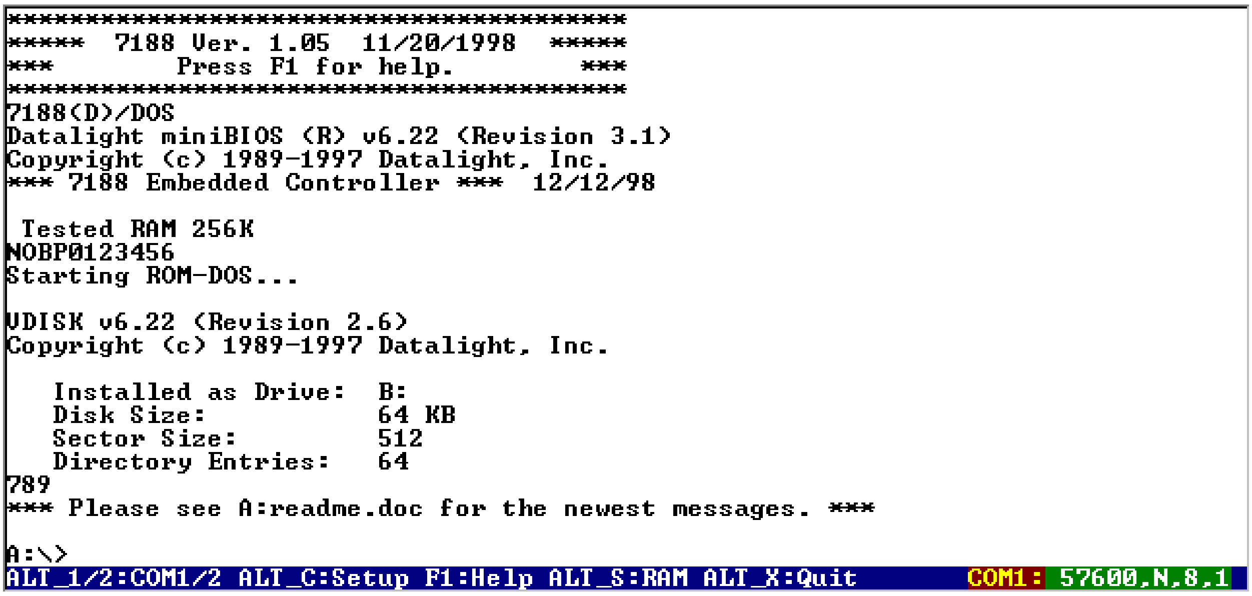
1. BC /o demo1
2. link demo1,,,7188L.lib;

After compiling, demo1.exe can be downloaded to the 7188 (D)/DOS module and to be executed.

1.6 Quick Start of 7188(D)/DOS

1. Refer to “7188(D)/DOS Hardware User’s Manual” Sec. 1.6.1 for wire connection between PC and 7188(D)/DOS module.
2. Connect the download cable between COM4 of 7188(D)/DOS and COM1 (2) of PC.
3. Power-on PC and execute 7188.EXE.
4. 7188.EXE is the utility of 7188(D)/DOS. It can be run on **DOS** or on the **DOS box of WIN9x/ WinNT.** ***It can only be used for standard COM1 or COM2 of PC.***
5. Press ALT_1 to use COM1 or press ALT_2 to use COM2.
6. Press ALT_C to set baud rate & data format of RS-232. Please set to 57600,N,8,1. The user can use LEFT/RIGHT key to change fields and press the Enter key to confirm. Other keys are used to change value.
7. Power-on 7188(D)/DOS again. Now the 7188 module will send messages to COM4, and 7188.exe will receive message from COM port. The message can be displayed on the screen of PC.

Now the screen will as follow:



```
****************************************
*****  7188 Ver. 1.05  11/20/1998  *****
***        Press F1 for help.        ***
****************************************
7188(D)/DOS
Datalight miniBIOS (R) v6.22 (Revision 3.1)
Copyright (c) 1989-1997 Datalight, Inc.
*** 7188 Embedded Controller ***  12/12/98

 Tested RAM 256K
NOBP0123456
Starting ROM-DOS...

VDISK v6.22 (Revision 2.6)
Copyright (c) 1989-1997 Datalight, Inc.

   Installed as Drive:   B:
   Disk Size:            64 KB
   Sector Size:          512
   Directory Entries:    64
789
*** Please see A:readme.doc for the newest messages. ***

A:\>
ALT_1/2:COM1/2 ALT_C:Setup F1:Help ALT_S:RAM ALT_X:Quit          COM1: 57600,N,8,1
```

8. At the same time, the 7188.exe will re-direct the input from PC's keyboard to COM port and the 7188(D)/DOS will receive these input from COM4. So, the user can use the keyboard of PC to send command to 7188 module and see the response on the screen of PC.

9. The user can type command, such as DIR A: or DIR B: , to see the files on disk A: (ROM-DISK) or DISK B: (RAM-DISK)

10. TIME and DATE command are used to show/set the TIME or DATE of 7188 module.

11. The user can use LD3 command to download DEMO1.EXE to the RAM-DISK of 7188 module & run it. (Please refers to "1.11 Using LD3.EXE ") The LD3.EXE is a utility of 7188 module and must be run on 7188 module. It is used to download program to RAM-DISK of 7188 module. User can find ld3.exe from the ROM-disk A: of 7188 module.

12. If COM2 of 7188 module is connected to another 7000's module, user can type command echo485 in order to send command to 7000's module. (Please refers to "1.15 Using ECHO485.EXE "). The echo485.exe is a utility of 7188 module and must be run on 7188 module. The user can find echo485.exe from the ROM-disk A: of 7188 module.

13. If the user want to change the contains of DISK A (ROM-DISK), please use command TOROM. (Please refers to “1.12 Using TOROM.EXE ”). The torom.exe is a utility of 7188 module and must be run on 7188 module.

1.7 Using 7188.EXE

The 7188.EXE is the main utility of 7188(D)/DOS. The major functions of 7188.EXE are given as follows,

1. Download any files from PC to the RAM-DISK of 7188 module (must co-operation with LD3.EXE)
2. Download ROM-DISK.IMG from PC to the Flash memory of 7188 module and update ROM-DISK (DISK A). It should be worked with miniBIOS or TOROM.EXE.
3. Show the output messages of 7188(D)/DOS on the screen of PC
4. Send the keyboard input of PC to 7188(D)/DOS
5. Communicate with RS-232 device

Modifying setup of 7188.EXE:

1. Press ***ALT+1*** to choose COM1 of PC
2. Press ***ALT+2*** to choose COM2 of PC
3. Press ***ALT+C*** to set baud rate & data format of COM port (RS-232), then
   Press **SPACE bar** to select parity and **Enter** key to next field,
   Press **SPACE bar** to select data and **Enter** key to next field,
   Press **SPACE bar** to select stop and **Enter** key to exit.

Notice: The correct way to complete the setup procedure is to press **ENTER** key on the **stop bit field**.

Other hotkey of 7188.exe:

1. **F1**............ show help screen.(press ESC to quit help screen)

2. **ALT_A**.....dir a:
3. **ALT_B**.....dir b:
4. **ALT_D**.....set time/date of PC to 7188(D)/DOS
5. **F2**............set execution file name to be download & run
6. **F6**............set the parameter(now, only can set 1 parameter)
7. **F9**............auto download program to 7188(D)/DOS
8. **F8**............auto download program to 7188 module & run the program
9. **F5**............run the program (set by use F2)

Example :

OBJECTIVE: To learn how to download echo485.exe to 7188 (D)/DOS and run echo485 using baud rate 115200

- The user should go to directory c:\7188dos\c\echo485 first.
- press F2
- Type “echo485.exe”+CR
- Press F6 and type “115200”+CR
- The setting is finished.
- Then press F8.
  7188.exe will automatically download echo485.exe to 7188 module and Send the command ”echo485.exe 115200” to 7188 module.
- If press F9, only automatically download echo485.exe to 7188 module.
- If press F5, 7188.exe will send command “echo485.exe 115200” to 7188 module.

10. **CTRL_F2...** Split the screen to 2 sections.

The upper screen is for COM1 and the lower screen is for COM2. Press ALT_X to back to normal mode and press TAB to toggle between COM1/2 screen.

11. **ALT_L** ....Change to line command mode. This mode is designed to send commands to 7000's modules. PC needs a 7520 converter to connect to 7000's modules.
    Press **ALT_L** again go back to normal mode.
12. **ALT_S**........Use with LD3.EXE to download program (Must run ld3.exe first)
13. **ALT_R**........Use with TOROM.EXE to download rom-disk.img (Must run torom.exe first)
14. **ALT_M** ....Use with miniBIOS to download rom-disk.img

1.8 Date of ROM-DOS

(Year is valid from 1980 to 2079)

Purpose

Display the current date (day, month, year) as known to ROM-DOS and allows the user to change it.

Syntax

DATE [*mm-dd-yy*]

Examples

If the command is entered without the specification of *mm-dd-yy*, the current date as known to ROM-DOS will be displayed and you will be prompted to enter a new date as follows,

```
Current date is Sat 8-18-1998
Enter new date (mm-dd-yy):
```

If the user doesn’t want to change the date, just press ENTER key. Otherwise, type the current date and press ENTER key. Alternatively, the user may skip the display and prompting by entering the current date on the command line. To enter June 10, 1989, the user can type the DATE command as follows (the century number 19 is assumed):

```
DATE 6-10-89
```

Valid entries for months, days, and years are:
mm = 1-12 *dd* = 1-31 *yy* = 80-99 (for years with 19 as the century)
ROM-DOS will figure out the day of the week and do not include it in your entry. **The earliest year that ROM-DOS can accept is 1980**. **ROM-DOS** assumes that the century is 19 unless something different is entered, such as 20, next to the year. For example, February 18, 2014 would be:

```
DATE 02-18-2014
```

ROM-DOS will adopt your entry as the current date.

1.9 Time of ROM-DOS

Purpose

Displays the current time as shown on the system's internal clock. Allows resetting of the clock.

Syntax

TIME [*hh*:*mm*:*ss*] [pm|am]

Remarks

If you just want to check the time maintained by ROM-DOS, type the TIME command by itself. ROM-DOS will display something like:

```
Current time is 3:00:02.48p
Enter new time:_
```

If you want to change the time, you can include the desired time on the prompt line after the word TIME. Or you may type the command with no option (as you do to check the time) and enter the new time before pressing <Enter>.

ROM-DOS displays the time according to the 24-hour clock with the a or p indicator to show AM or PM. The AM / PM indicator can be typed as “a” or “p” or as “am” or “pm”. The time may be entered in a 24 hour format or a 12 hour format with the AM or PM designator. The allowed options for hours and minutes are:

***hh* = 0-24 *mm* = 0-59 *indicator* = a, p, am, or pm**

ROM-DOS displays time to hundredths of seconds. When entering time, however, you needn't enter seconds or hundredths; ROM-DOS will assume a value of zero if they are not specified.

You may skip the display and prompting by simply typing the current time after the word TIME on the command line:

```
TIME 23:24
```

ROM-DOS will accept your entry as the current time.

Examples

To enter the time 11:15 pm, you can type:

TIME 23:15

-or-

TIME 11:15 p

To enter the time as 9:26 am, you can type:

TIME 9:26

-or-

TIME 9:26 am

1.10 VDISK.SYS of ROM-DOS

Purpose

VDISK is a device driver that allows you to use memory as a disk.

Syntax

device = VDISK [*size* [*secs*[*dirs*]]] [/E]

Remarks

VDISK allows you to partition some of your computers memory as a disk. This disk is called a RAM disk or Virtual Disk. A RAM disk is much faster than either a floppy or hard disk.

Any data on the VDISK is lost when the system power is off. The size argument specifies the size of the VDISK in kilobytes. The default is 64K bytes.

The secs argument specifies the sector size in bytes. The default is 512 bytes per sector. This value must be 128, 256, 512 or 1024. All other values are not valid and the default of 512 bytes will be used.

The dirs argument specifies the number of root directory entries. The default is 64 directory entries. There may be any number of root directory entries between 2 and 1024. If an odd number is given, it will be rounded up to the nearest multiple of 16, in order to fill the entire sector.

The VDISK increases the resident size of DOS.

Examples

device = VDISK.SYS

The above example builds a 64K byte RAM disk in DOS memory.

device = C:\DOS\VDISK.SYS 220 /E

This example builds a 220K bytes RAM disk in extended memory. The VDISK device driver is loaded from the C: drive and the \DOS directory. VDISK assumes the default 512 bytes sector size, and 64 directory entries.

device = VDISK.SYS 45 128 18

The above example builds a 45K byte RAM disk in DOS memory. There will be 128 bytes sectors and 18 root directory entries.

1.11 Using LD3.EXE

LD3.exe is a utility of 7188(D)/DOS that allows you to download program to the ram-disk of 7188 module. The download steps are as follows.

1. After connecting 7188(D)/DOS with PC, the user can type the command "***LD3***" and press ***ENTER*** *key.* The screen shows,
   ```
   LD3.EXE Ver1.01
   7188 filename=_
   ```
   Please type the file name of the file will be download. The file will be saved on disk B:

2. For example, type demo1.exe , then the screen shows,
   ```
   7188 RAM-DISK filename=B:demo1.exe
   Press ALT-S for Download from PC (baud=115200)
   ```
   then press ***ALT_S*** , and the screen shows,
   ```
   PC-filename=_
   ```
   type the file name of the file want to be download again.

   (The file must can be found on the current directory of PC or must type the path of the file to be found.

3. Then the file (demo1.exe) will be download to disk B: of 7188 module, and the screen shows,
   ```
   Size=12544
   Download 12632 Bytes OK. .........................The file size
   Transfer time is: 1.483516 seconds.....total time of download
   Test count=35403 .........................................test message
   Back to Terminal mode
   A:\>
   ```

4. Then press dir b: will get the message:
   ```
   A:\>dir b:
   Volume in drive B is VDISK v6.22
   Directory of B:\

   DEMO1    EXE        12,632 06-30-1999  12:11a
           1 file(s)         12,632 bytes
                             49,152 bytes free
   ```

```
A:\>
```

5. Now can type “***demo1***” to run it on 7188(D)/DOS

1.12 Using TOROM.EXE

TOROM.EXE is a utility of 7188(D)/DOS that allows you to download the image file “rom-disk.img” to 7188(D)/DOS. The download steps is as follows:

1. Connect 7188(D)/DOS with PC. Type the command “LD3”and press ENTER. The screen shows,

```
ToRom Ver1.04

(1) Flash ID=B001
(2) Start to Download....
(3) Wait PC to Download ROM-DISK.Press ALT_R in
7188.exe
```

   Press ***ALT_R***, and then the image file “*rom-disk.img*” can be download to the flash memory of 7188(D)/DOS.

2. After download procedure, the 7188(D)/DOS will be reset. And ROM-DISK A: will contain the new files.

1.13 Using INIT*

When there is no torom.exe to use(Note 1,2), user can use another method to download new rom-disk.img to 7188(D)/DOS. The steps are as follows:

1. Turn off the power of 7188(D)/DOS
2. Connect INIT* pin (pin 6 of CON1) to Ground.
3. Power-on 7188(D)/DOS, then screen shows,

```
7188(D)/
(1) Start to Download
(2) Wait PC to Download Rom-disk(Press ALT_M)
```

(Now can disconnect INIT* pin from GND) Please press ***ALT_M*** , then 7188.exe will download rom-disk.img to 7188(D)/DOS.

4. After download, please turn off the power of 7188 module and power-on it again.

Note1: Why lose torom.exe?
The user doesn’t include it with rom-disk.img before download. **Note2**: With torom.exe, but the program can’t stop run on 7188 module and back to ROM-DOS.)

1.14 Using ROMDISK.EXE

ROMDISK.EXE is a utility of ROM-DOS and is used for making a ROM-DISK image file. The steps for create a new ROM-DISK are as follows.

1. Create a sub-directory called ***DISK.***
2. Copy all the files need to the directory.
3. Run "***romdisk disk***". (romdisk.exe will put all files in the directory **DISK** into a file "**rom-disk.img**".)
4. Now, the user can download the image file to 7188(D)/DOS.
5. After download and re-boot, the disk A: will contain all the files the same as in the sub-directory ***DISK***.

(User can just run "***romdisk***" without parameter, and then romdisk.exe will show the help message.)

1.15 Using ECHO485.EXE

Echo485.exe is a utility to send command from COM2 of 7188 (D)/DOS to 7000's modules. Just type the command "echo485 [baudrate]" and the default baud rate is 9600. Before selecting other baud rate, be sure to add the baud rate on command line. For example, using echo485 with baud rate 115200, the user should type "echo485 115200". The screen will show the following message:

```
A:\>echo485 115200

Echo485 Ver. 1.02
Use COM2(RS485) connect to 7000's
echo485 [baudrate] -->  default:9600
's' to toggle checksum ON (1)/OFF(0)
CTRL_S to search 7000's modules *** NEW ***
'q' to quit to ROM-DOS
Current mode:baudrate=115200,checksum=0
```

- Press **'s'** to toggle checksum mode. (The default checksum mode is disabled)
- Press **'CTRL_S'** to search 7000's modules using current baud rate and checksum mode.
- Press **'q'** will end the program and return to ROM-DOS.

Echo485.exe is very useful for user to configure or test 7000 modules.

1.16 Using ECHOCOM.EXE

EchoCom.exe is similar to **echo485.exe.** It can be used to control COM1/2/3, but **echo485.exe** can only be used to control COM2. EchoCom.exe can be found on demo program directory **C:\echocom.**

The command format is "**echocom [/pn] [/bm]",**
n is port number can be 1 or 2 or 3, default is 3.
m is baud rate, default is 9600.

After running echocom.exe, user can

- press **'s'** to toggle checksum mode.
- press **CTRL_A** change to use COM1.
- press **CTRL_B** change to use COM2.
- press **CTRL_C** change to use COM3.
- type "***bnnnn**" to change baud rate(nnnn is baudrate).

For example:

Type "***b57600**" to change baud rate to 57600

2. Operation Principle

After power-on the 7188(D)/DOS, the mini-bios will start to work as follows.

- initial the 80188 hardware
- send out diagnostic message to COM4
- setup minibios interrupt table & handle
- check INIT* pin status, if connected to GND then goto download rom-disk.img procedure.
- enable the watchdog timer to 1.6 secs
- reset the watchdog timer
- transfer control to the ROM-DOS

The ROM-DOS will start to work as follows.

- initial the ROM-DOS
- send out diagnostic message to COM4
- reset the watchdog timer
- search all ROM-DISK in the Flash Memory area(7188 modules support at most 2 ROM-DISK)
- execute command in config.sys
- execute command in autoexec.bat
- accept standard input from COM4
- send standard output to COM4

2.1 System Layout

F000:E000 to FFFF	**MiniBIOS (8K)**
F000:C000 to DFFF	**Reserved (8K)**
F000:0000 to BFFF	**ROM-DOS (48K)**
C000:0000 to E000:FFFF (for 7188(D)/DOS)	**Flash Memory(192K) ,ROM-DISK is begin from segment 0xC000**
8000:0000 to E000:FFFF (for 7188(D)/DOS/512)	**Flash Memory(448K) ,ROM-DISK is begin from segment 0x8000**
0000:0000 to 3000:FFFF	**SRAM (256K)**

The Flash Memory that is not used by MiniBIOS or ROM-DOS or ROM-DISK can be used to store data.
For example:

- If the size of rom-disk.img is **60K (<64K)**, ROM-DISK occupies the segment 0xc000 (or 0x8000 for 512K type), and the segment 0xF000 is occupied by miniBIOS & ROM-DOS, so user can use segment from 0xD000 (or 0x9000 for 512K type) to 0xE000 to store data. Therefore there are 128K Flash memory (or 384K for 512K type) to store data.

- If the size of rom-disk.img is **80K (>64K & < 128K)**, ROM-DISK occupy segment 0xc000 & 0xd000 (or 0x8000 & 0x9000 for 512K type). The user can use segment 0xe000(or 0xa000 to 0xe000 for 512K type) to store data. Therefore there are 64K Flash memory (or 320K for 512K type) to store data.

2.2 miniBIOS

The Datalight's miniBIOS, as the name implies, is the minimum BIOS needed to run Datalight's ROM–DOS. It is not intended to replace full BIOS but to serve those embedded situations that do not require full BIOS support.

What is BIOS?

The BIOS (Basic input/output System) is the first program to gain control when power is applied to any computer system. The job of the BIOS is to bring the system up, initialize relevant hardware and RAM, as well as provide a software layer between DOS and various hardware devices. ROM–DOS requires BIOS to run.

The Datalight miniBIOS is a BIOS subset that provides support for only the BIOS features absolutely essential to the operation of ROM–DOS. The miniBIOS includes BIOS support for a remote console, timer, BIOS extensions, and hardware equipment identification. The miniBIOS does not provide support for floppy or hard disks, printers, the standard PC keyboard, or monitors.

What Features Does the miniBIOS Support

The following table shows the calls made by the ROM–DOS kernel to the BIOS. ROM–DOS makes no hardware assumptions, but instead, performs all interactions with the hardware through BIOS calls.

Minimum BIOS Calls used in a Diskless System

Name	Interrupt	Sub-function
Coprocessor Esc instruction	07H	reserved
Timer 0 tick	08H	Reserved (hardware)
Keyboard Input	09H	(stubbed for APM)
Serial receive char	0BH	Reserved (hardware)
Video TTY Output	10H	0EH
Get equipment list	11H	
Get memory size in K	12H	
Disk I/O	13H	reserved
Extended Memory Support	15H	reserved
Keyboard Support	16H	00H, 01H, 02H
Keyboard, push Scan code into Buffer	16H	05H
Boot failure message	18H	
System DOS-boot	19H	
Time of Day	1AH	00H, 01H
Timer Tick	1CH	

2.3 How to develop Application Program

It is recommended to develop application program as follows.

1. Compiling & link in PC, download the .EXE file into RAM-DISK of 7188 module for testing.
2. Use standard input & standard output of 7188 module to debug program
3. If there are any bugs, go to step 1
4. Make a new ROM-DISK.IMG and download it into 7188 module's ROM-DISK

2.4 How to download file to RAM-DISK

Refer to “***1.11 Using LD3.EXE***”, to download execution file into the RAM-DISK of 7188 (D)/DOS

2.5 How to Download file to ROM-DISK

The user can't download (or copy) a file into the ROM-DISK (disk A) directly. He should use romdisk.exe (a utility of Datalight) to make an image file (default is rom-disk.img), and then use torom.exe or INIT* download the image file to 7188(D)/DOS.

Please refers to "***1.12 Using TOROM.EXE***" and "***1.13 Using INIT****" & "***1.14 Using ROMDISK.EXE***".

2.6 How to use 192(448)K ROM-DISK

The steps to make ROM-DISK.IMG are given as follows.

1. Copy all files into directory **DISK**
2. Execute ***ROMDISK.EXE DISK*** to generate ROM-DISK.IMG

There are 14 files on the ROM-DISK of 7188(D)/DOS, but most of these files are not used on user's final application. If the user want to use ROM-DISK space as big as possible, remove all other files except CONFIG.SYS and contain one line "shell=userprog.exe" in the config.sys file. Argument can be added on the end of this line if needed. User files can be added to it too, such as userprog.exe, and other data files. Eventually creating a new ROM-DISK.IMG and download it to 7188 (D)/DOS.

If **torom.exe** file can't be found on 7188 (D)/DOS, the user should use INIT* method to download new ROM-DISK.IMG file to 7188(D)/DOS.

3. C Language Library

3.1 CONST defined in 7188.H

```
#define   NoError              0
#define   InitPinIsOpen        0
#define   InitPinIsNotOpen     1
#define   QueueIsEmpty         0
#define   QueueIsNotEmpty      1
#define   PortError           -1
#define   DataError           -2
#define   ParityError         -3
#define   StopError           -4
#define   TimeOut             -5
#define   QueueEmpty          -6
#define   QueueOverflow       -7
#define   PosError            -8
#define   AddrError           -9
#define   BlockError         -10
#define   WriteError         -11
#define   SegmentError       -12
#define   BaudRateError      -13
#define   CheckSumError      -14
#define   ChannelError       -15
#define   TimeIsUp             1
```

3.2 Function By Type

3.2.1 Functions for COM ports

Install COM port driver

Function name	description
InstallCom	For any one of 4 COM ports
InstallCom1	For COM1
InstallCom2	For COM2
InstallCom3	For COM3
InstallCom4	For COM4

UN-install COM port driver

Function name	description
RestoreCom	For any one of 4 COM ports
RestoreCom1	For COM1
RestoreCom2	For COM2
RestoreCom3	For COM3
RestoreCom4	For COM4

Check if the data is in the input buffer of COM port

Function name	description
IsCom	For any one of 4 COM ports
IsCom1	For COM1
IsCom2	For COM2
IsCom3	For COM3
IsCom4	For COM4

Read one byte data from input buffer of COM port

Function name	description
ReadCom	For any one of 4 COM ports
ReadCom1	For COM1
ReadCom2	For COM2
ReadCom3	For COM3
ReadCom4	For COM4

Send one byte data to COM port

Function name	description
ToCom	For any one of 4 COM ports
ToCom1	For COM1
ToCom2	For COM2
ToCom3	For COM3
ToCom4	For COM4

Send n bytes data to COM port

Function name	description
ToComBufn	For any one of 4 COM ports
ToCom1Bufn	For COM1
ToCom2Bufn	For COM2
ToCom3Bufn	For COM3
ToCom4Bufn	For COM4

Send string to COM port

Function name	description
ToComStr	For any one of 4 COM ports
ToCom1Str	For COM1
ToCom2Str	For COM2
ToCom3Str	For COM3
ToCom4Str	For COM4

Clear all data in COM port input buffer

Function name	description
ClearCom	For any one of 4 COM ports
ClearCom1	For COM1
ClearCom2	For COM2
ClearCom3	For COM3
ClearCom4	For COM4

Wait transmitted data and send out to COM port

Function name	description
WaitTransmitOver	For any one of 4 COM ports
WaitTransmitOver1	For COM1
WaitTransmitOver2	For COM2
WaitTransmitOver3	For COM3
WaitTransmitOver4	For COM4

Set the direction of RS-485 bus

Function name	description
Set485DirToTransmit	Set to transmit mode, for COM1/2
Set485DirToReceive	Set to receive mode, for COM1/2

Check if the output buffer of com-port is empty

Function name	description
IsCom3OutBufEmpty	For COM3
IsCom4OutBufEmpty	For COM4

Get the data size in COM port input buffer

Function name	description
DataSizeInCom	For any one of 4 COM ports
DataSizeInCom1	For COM1
DataSizeInCom2	For COM2
DataSizeInCom3	For COM3
DataSizeInCom4	For COM4

Communicate with 7000's module

Function name	description
SendCmdTo7000	Send command to 7000's module
ReceiveResponseFrom7000	Receive response from 7000's module

Formatted output (just like use printf)

Function name	description
printCom	For any of 4 COM ports
PrintCom1	For COM1
PrintCom2	For COM2
PrintCom3	For COM3
PrintCom4	For COM4

3.2.2 Functions for LED

For red LED ON/OFF

Function name	description
LedOff	Set LED Off
LedOn	Set LED On

3.2.3 Functions for 5-Digit LED

Function name	description
Init5DigitLed	Initialize 5-Digit Led
Show5DigitLed	Show digit number(0-F)
Show5DigitLedWithDot	Show digit number and the dot '.'
Show5DigitLedSeg	Show any segment
Set5DigitLedTestMode	Set to test mode, all segment will be turn ON
Set5DigitLedIntensity	Set the intensity of 5-Digit LED
Disable5DigitLed	Disable 5-Digit LED, all segment will be turn OFF
Enable5DigitLed	Back to normal mode

3.2.4 Functions for data storage

For NVRAM

Function name	description
ReadNVRAM	Read one byte data from NVRAM
WriteNVRAM	Write one byte data to NVRAM

For EEPROM

Function name	description
WriteEEP	Write one byte data to EEPROM
ReadEEP	Read one byte data from EEPROM
EnableEEP	Enable EEPROM, so it can be write
ProtectEEP	Set EEPROM to write protect

For flash memory

Function name	description
FlashReadId	Read the Flash Memory type
FlashWrite	Write one byte data to Flash Memory
FlashErase	Erase one sector of Flash Memory
FlashRead	Read one byte data from Flash Memory

3.2.5 Functions for WatchDog Timer

Function name	description
EnableWDT	Enable WatchDog timer
RefreshWDT	Refresh WatchDog timer
DisableWDT	Disable WatchDog timer
IsResetByWatchDogTimer	Check if 7188 module is reset by WatchDog timer
IsResetByPowerOff	Check if 7188 module is reset by Power Off

3.2.6 Functions for timer

Function name	**description**
TimerOpen	Install timer driver
TimerClose	Uninstall timer driver
TimerResetValue	Reset timer ticks to 0
TimerReadValue	Read timer ticks
StopWatchReset	Reset stopwatch timer value to 0
StopWatchStart	Start stopwatch
StopWatchStop	Stop stopwatch
StopWatchPause	Pause stopwatch
StopWatchContinue	Continue stopwatch
StopWatchReadValue	Read the timer value of stopwatch
CountDownTimerStart	Start countdown timer
CountDownTimerReadValue	Read the timer value of countdown timer
InstallUserTimer	Install user's timer function
InstallUserTimer1C	Install user's timer function on INT 0x1C
DelayTimeMs	Delay time, unit is 1ms (S/W)
DelayMs	Delay time, unit is 1ms(H/W)
Delay	Delay time, unit is 1ms(H/W)
Delay_1	Delay time, unit is 0.1ms(H/W)

NOTE:

S/W means use software loop to implement the function.
H/W means check hardware timer to implement the function.

3.2.7 Functions for stdio

Function name	description
getch4	For instead of getch
kbhit4	For instead of kbhit
ungetch4	For instead of ungetch
putch4	For instead of putchar

3.2.8 MISC functions

Function name	description
InitLib	Initialize 7188 library
GetLibVersion	Get version number of 7188 library
Is7188	Check if the program is running on 7188 module
ReadInitPin	Read the status of INIT* pin
hex_to_ascii[16]	Global variable, for convert 0-15 to ‘0’-‘F’
ascii_to_hex	For convert ‘0’-‘F’ to 0-15
_MK_FP	Make a far pointer

3.3 Library Reference

Please refer to **7188LIB.htm** or **7188libe.pdf** or **7188libc.pdf**. **(7188libe.pdf is English version, 7188libc.pdf is Chinese version)**

If there are new functions added, the reference of these functions will be found in C\LIB\Newlib.txt or C\LIB\7188lib.htm

3.4 C functions can’t be used on 7188

Because the hardware structure of 7188(D)/DOS is not fully compatible to PC, so all the functions for some special hardware that cannot be found on 7188(D)/DOS will not work on 7188(D)/DOS. Such as:

1. Graphics mode functions.
2. Text mode functions, but direct write data to screen memory. Such as **cprintf**.
3. Protect mode function, such as use **EMS** or **XMS** or use **DOS-extender**.
4. Use print port, such as **biosprint** or **_bios_printer**.
5. Use PC standard COM port, such as **bioscom** or **_bios_serialcom**.
6. Use 8087. (If want use floating point function, please set compiler option to use emulation)
7. Win32 functions. (7188(D)/DOS can’t run a windows program, even the Win32 console mode program)

4. Demo Program

There are many demo programs in the companion floppy disk for reference. All the program source & documentation are given in the same directory. The common features are shown as follows.

The entire demo program written in **C language** is on the sub-directory **C** and the library is put on **C\LIB**. Each demo program is copied on a sub-directory, which has files as follows.

- readme.txt(or *.txt) .. documentation file
- *.c............................ source file
- *.h .... include file(7188.h)
- gotc.bat .... compiler & link batch file use TC 2.0
- gomsc.bat .... compiler & link batch file use MSC 6.0
- *.exe........................ execution file

These demo programs are shown as follows.

Sub-dir name	Demonstrative topic
COM3COM4	writing **COM3/4 ISR(**interrupt service routine**)**
ECHO485	source of **ECHO485.EXE**
ECHOCOM	using **COM1/2/3**
EMU7520	using 7188 to **emulation 7520**
FLASH	reading/writing/erasing **Flash Memory**
LD3	source of **LD3.exe**
LOADRAM	download EXE file into 7188 module's RAM-DISK
LOADRAM2	download EXE file into 7188 module's RAM-DISK
MMICON	connecting to **MMICON**
MMICON2	connecting to **MMICON** (use method 2)

Sub-dir name	**Demonstrative topic**
QC\QC1	QC program 1
QC\QC2	QC program 2
QC\QC3	QC program 3
QC\QC4	Test all components on 7188(D)/DOS
TOROM	Source of **TOROM.EXE**
WATCHDOG	Test **WatchDog Timer**
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/write **NVSRAM**
DEMO6	read/write **EEPROM**
DEMO7	enable/disable/refresh **Watchdog Timer**
DEMO8	COM1 connect to **MODEM**. Write modem application & how to **write COM1/COM2 ISR**
DEMO9	write data to **Flash Memory**
DEMO10	connect to 7000, 4*7052+1*7053+2*7043
DEMO11	connect to 7000, 7053 + 7053
DEMO12	connect to 7000, 7053 + 7051
DEMO14	connect to 7000, 7*7053(16*7=112 DI)
DEMO18 TEST18	TEST18 run on master controller DEMO18 run on slave controller On slave node: COM2 ...... master controller COM1 ...... 7520+4*7052(4*8=32 DI) COM3 ...... 7520+4*7067(4*7=28 Relay DO) COM4 ...... 7520+4*7042(4*13=52 DO)
DEMO20	for connect to **SIEMEMS's S7**
DEMO21	for connect to **Facon PLC**
DEMO30	connect to touch screen, **TOUCH-200**
DEMO31	connect to touch screen, **HITECH, PWS-1700**
DEMO32	source code for **DEMOBOX2**
DEMO33	source code for **DEMOBOX3**

Sub-dir name	Demonstrative topic
DEMO34	source code for **DEMOBOX4**
DEMO35	source code for **DEMOBOX for Taipei computex show 99'**
DEMO36	source code for **DEMOBOARD for Taipei computex show 99'**
DEMO37	source code for **DEMOBOX5**
DEMO40	connect to card reader
DEMO50	IKIT-07 demo program (7188 + 7060)
DEMO51	IKIT-08 demo program (7188 + 7060 + touch 200)
DEMO60 DEMO61 DEMO62	For PC use RS-485 bus connect to many 7188 module . Use DEMO60 to set address of 7188 module. DEMO61 run on 7188 module. DEMO62 run on PC.
DEMO70	Use 7188 module convert PC to addressed device
DEMO80	Write **user's int 0x1C interrupt routine**
DEMO90	use Timer function, use **Timer ticks**
DEMO91	use Timer function, use **CountDownTimer**
DEMO92	use Timer function, use **StopWatch**
DEMO93	use Timer function, use **StopWatch**
DEMO94	use Timer function, use **StopWatch**
DEMO95	use Timer function, **user's timer function**
DEMO96	use Timer function, **user's timer function**
DEMO97	use Timer function, use function **DelayMs**
DEMO98	use Timer function, **user's timer function**.

More and more demo programs will be added. Call for more information.

4.1 How To Use NVRAM

The 7188 module has 31 bytes NVRAM. 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
```

4.2 How To Use EEPROM

The 7188 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.

4.3 How To Use Flash Memory

The 7188 module has 256K or 512K flash memory size. It contains the miniBIOS, ROM-DOS, and the ROM-DISK (disk A: on 7188 module). ROM-DOS and miniBIOS 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.

4.4 How To Use 5-Digit LED Display

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);
```

4.5 How To Use WatchDog Timer

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 module. The code can be as follows.

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

4.6 How To Use COM1/2/3/4

The 7188(D)/DOS 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(57600L,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)/DOS 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);
```

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 control the direction of RS-485 bus.