

数据处理模块（7188） 使用手册



中国·河北石家庄亿邦万达科技有限公司

ShiJiaZhuang InBand Technology Co.,Ltd.

二〇一〇年一月

目 录

I 硬件使用手册	4
1 概述	4
2 系统详述	5
2.1 结构图.....	5
2.2 特性详述.....	6
2.2.1 CPU.....	6
2.2.2 SRAM	6
2.2.3 Flash Memory	6
2.2.4 NVRAM.....	6
2.2.5 EEPROM	6
2.2.6 实时时钟.....	6
2.2.7 COM1	6
2.2.8 COM2	7
2.2.9 COM3	7
2.2.10 COM4	7
2.2.11 5 位数字显示器 (7 段数码管) & LED 显示	7
2.2.12 数字量输入和初始化端子.....	7
2.2.13 一个 LED.....	7
3 端子分布	8
3.1 结构图.....	8
3.2 10-pin 端子分布表.....	9
3.3 COM1 端子分布 (DB9 M)	9
3.4 跳线.....	10
4 系统	10
4.1 内存映射.....	10
4.2 中断映射.....	10
4.3 GPIO 分布状态.....	11
5 操作	12
5.1 调试系统&COM4	12
5.2 使用 COM4 下载程序.....	12
5.3 COM3: RS232 端口	13
5.4 COM2: RS485 端口	13
5.5 COM1: RS232 端口	13
5.6 COM1: RS485 端口	14
5.7 7 段数码管 & LED 显示控制	14
5.8 EEPROM.....	15
5.9 实时时钟 (RTC) & NVRAM.....	15
5.10 看门狗定时器 (WDT)	16
II 数据处理软件使用说明	16
1 快速入门	16
2 系统映射	17
3 Mini BIOS	17

3.1	BIOS.....	17
3.2	MiniBIOS 特点.....	18
4	ROM-DOS	19
5	超级终端	19
5.1	Comm. exe.....	19
5.2	超级终端.....	21
6	文件传送	22
6.1	Transfer. exe.....	22
6.2	Xm. exe.....	23
7	ROMDISK. EXE	24
8	TOROM. EXE	24
9	VDISK	25
10	附录	26
	导入次序.....	26
11	数据处理函数库	27
11.1	使用 TC 和 BC.....	28
11.2	函数调用.....	30
11.3	C 的局限性.....	36
11.4	数据处理库函数使用技巧.....	37

-

I 硬件使用手册

1 概述

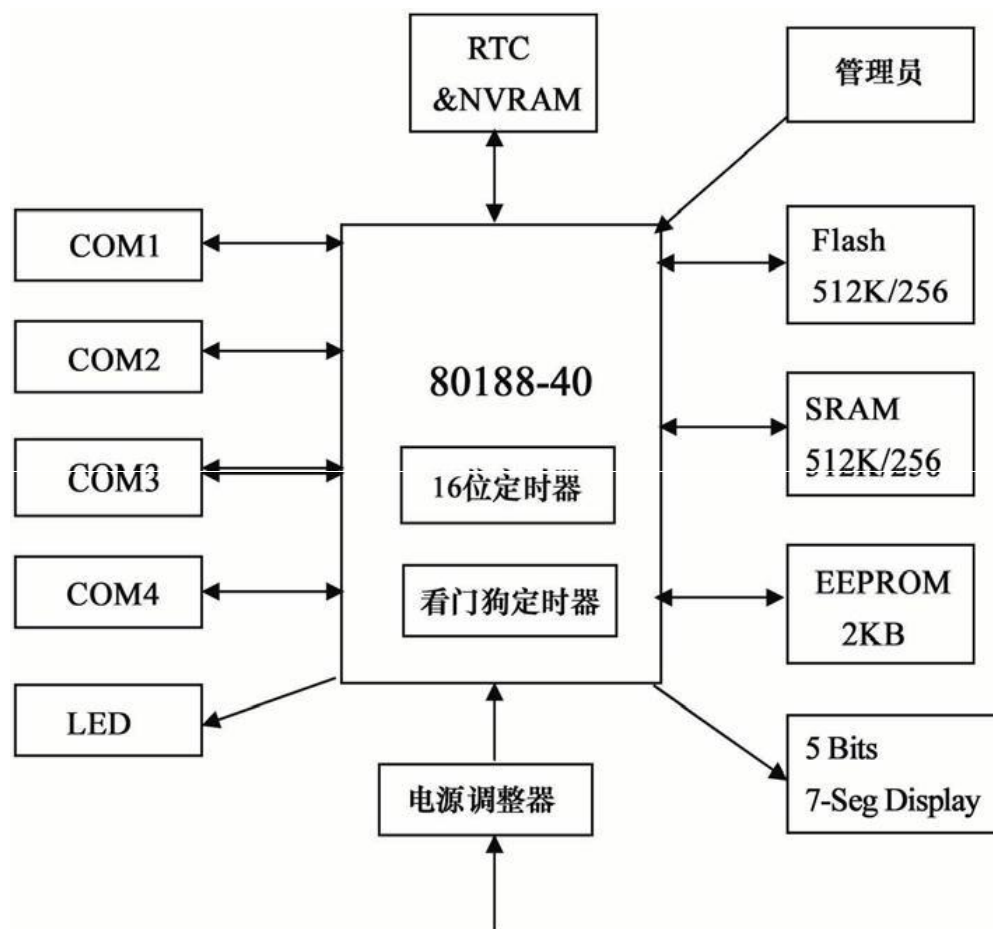
数据处理模块是一个多功能嵌入式通信控制器，其性能高、功耗低、可编程，适合不同行业的应用。数据处理模块能把不同的设备连接在一起。 $-40^{\circ}\text{C} \sim +85^{\circ}\text{C}$ 的宽工作温度范围能满足工业恶劣环境下的苛刻要求，可以取代其他控制器如工业 PC 或 PLC。

数据处理配有 AMD 80188 微处理器、SRAM、Flash ROM、EEPROM、实时时钟（RTC）、4 个串行通讯口(2 个 RS232 串口（COM3 和 COM4），1 个 RS485 串口（COM2），1 个 RS232/RS485 串口（COM1）)。

数据处理模块中含有操作系统（ROM-DOS），ROM-DOS 功能等同于 MS-DOS，一个标准的 DOS 运行的程序同样可在 ROM-DOS 中运行，用户可以通过 COM4 上传或下载程序。

2 系统详述

2.1 80188 微处理器结构图



2.2 特性详述

2.2.1 CPU

Am188 ES, 40Hz; 1MB 存储容量, 1MB I/O 空间, 32 位 GPIO, 6 位中断, 2 个定时器。

2.2.2 SRAM

静态存储 A617308V-12x2, 256KB, 存储控制程序, 在 ROM-DOS 中可以成为 ROM-DISK 的驱动盘。

2.2.3 Flash Memory

Flash ROM Am29F010-70EC, 512KB, 在 ROM-DOS 中可以成为 ROM-DISK 的驱动盘。

2.2.4 NVRAM

DS1302, 32 位字节, 上电期间, 保存数据长达数年之久。

2.2.5 EEPROM

EEPROM, 2048 位字节, 存储参数或变量数据。

2.2.6 实时时钟

DS1302, 自 2003 年之后顺序显示。

2.2.7 COM1

RS-232/RS-485, 最大波特率可达 115200bps。

2.2.8 COM2

RS-485, 最大波特率可达 115, 200bps。

2.2.9 COM3

RS-232, 最大波特率可达 115200bps。

2.2.10 COM4

RS-232, 最大波特率可达 115200bps。

2.2.11 5 位数字显示器 (7 段数码管) & LED 显示

7 段数码管能显示 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f 的浮点数

2.2.12 数字量输入和初始化端子

用户可用 INIT*信号端子连接初始化程序。

注意: 执行文件传输功能后, 在屏幕上显示 DOS 系统提示符, 数据处理的 INIT*端子和 GND 必须短接。

2.2.13 一个 LED

用户可以控制 LED 的开或关。

2.2.14 电源消耗

电源输入: 5V—30V。

数据处理模块静态电源消耗是 2.0W, 动态电源消耗是 3.0W。

2.2.15 尺寸和外部环境

模块尺寸：117mmX71mmX26mm

工作温度：-20℃~+60℃

外界温度：-40℃~+80℃

湿度：0%~90%

3 端子分布

3.1 结构图



3.2 10-pin 端子分布表

PIN	名称	描述
1	RXD4	RXD pin of COM4 (RS-232)
2	TXD4	TXD pin of COM4 (RS-232)
3	GND	GND pin of COM4 & COM3
4	TXD3	TXD pin of COM3 (RS-232)
5	TXD3	RXD pin of COM3 (RS-232)
6	INIT*	初始端
7	D2+	DATA+ pin of COM2 (RS-485)
8	D2-	DATA- pin of COM2 (RS-485)
9	+VS	电源端 (V+)
10	GND	电源端 (GND)

注意:

COM2= (D2+, D2-)

2.COM3= (TXD3, RXD3, GND)

3.COM4= (TXD4, RXD4, GND)

COM3&COM4 共用 GND-pin

3.3 COM1 端子分布 (DB9 M)

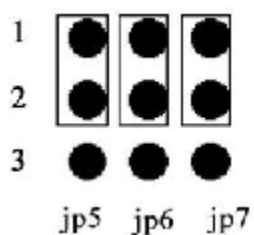
PIN	名称	描述
1	DCD	监测传输数据
2	RXD	结束数据 (RS232: jp5,jp6,jp7)
	D1+	DATA+ of RS-485 (RS232: jp5,jp6,jp7)
3	TXD	传输数据 (RS232: jp5,jp6,jp7)
	D1-	DATA- of RS-485 (RS232: jp5,jp6,jp7)
4	DTR	终端数据
5	GND	底线
6	RTS	数据设置完毕
7	RTS	发送请求
8	CTS	清除传输
9	R1	报警器

注意：

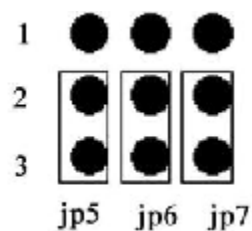
1. 通过跳线选择，COM1 可设为 RS-232 或 RS-485 端口
2. COM1 的连线和 DB9 与 PC 的连接方式一样

3.4 跳线

COM1: RS-232 端口



COM1: RS-485 端口



4 系统

4.1 内存映射

设备	地址
Flash ROM	8000:0000 到 F000:FFFF
SRAM	0000:0000 到 3000:FFFF
COM1	0x200
COM2	0x100
COM3	FF80 到 FF88
COM4	FF10 到 FF18

4.2 中断映射

中断号	中断映射
0	
1	Trace
2	NMI
3	断点
4	删除溢出除外
5	组件除外
6	未使用操作码除外
7	ESC 操作码除外
8	定时器 0
9	保留
0A	DMA-0
0B	DMA-1
0C	COM1
0D	COM2
0E	保留
11	COM4
12	定时器 1
13	定时器 2
14	COM3

4.3 GPIO 分布状态

0		X
1		X
2	PCS6	E2-WP
3	PCS5	E2-SDA
4		X
5	DEN	COM1-485DIR
6	SRDY	COM2-485DIR
7		X
8		X
9		X
0A		X
0B		X
0C	DRQ0	RTC-RST
0D	DRQ1	RTC-SDA
0E		X

0F	MCS1	INIT
10		X
11	PCS1	COM1CS
12	PCS2	COM2CS
13	PCS3	LED
14	RTS0	7seg-SDA
15	CTS0	7seg-LOAD
16		X
17		X
18		X
19		X
1A	UZI	SCLK
1B		X
1C		X
1D		X
1E		X
1F		X

5 操作

5.1 调试系统&COM4

数据处理模块的 COM4，用来传递系统和 PC 之间的数据。

数据处理上电，WIN98/WIN2000 环境下使用超级终端，窗口出现一组 DOS 命令

5.2 使用 COM4 下载程序

数据处理模块串口 COM4 与 PC 连接，将 INIT*端子和 GND 短接，系统上电，下载的程序就被存放在模块的 Flash ROM 中；如果用户想通过 COM4 调试程序，通过操作强制端，上电即可。

注意：执行文件传输完成后，在屏幕上显示 DOS 系统提示符，数据处理模块的 INIT*端子和 GND 必须短接，否则，系统将自动报警。

5.3 COM3: RS232 端口

COM3 将控制 数据处理的库函数,“InstallCOM3 {}”是用来初始化格式化后的 COM3 的波特率,此时中断将被 1KB 的缓冲器占用。“ToCOM3 (3, data)”是给 COM 端口传输数据。若用“IsCOM (3)”接受任何数据,那么,“ReadCOM (3)”就会首先得到数据。

5.4 COM2: RS485 端口

COM2 是 2 路的 RS485 端口,在 RS485 网络中 D+和 D-应与 Data+和 Data-相连。“installCOM2 {}”是用来初始化格式化后的 COM2 的波特率,此时中断将被 1KB 的缓冲器占用。

为了传输数据,“Set485DirToTransmit(2)”必须设置正确的数据,“ToCom(2,data)”用来传送数据,完毕后,COM2 通过命令设置接收方式,“WaitTransmitOver (2)”和“Set485DirToReceive (2)”若用“IsCOM (2)”接受任何数据,那么,“ReadCOM (2)”就会首先得到数据。

5.5 COM1: RS232 端口

COM1(9 pin)与数据终端设备连接,选择跳线 pin2&3。

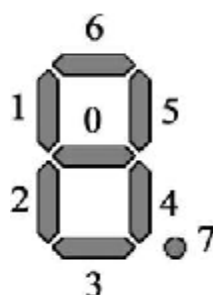
COM1 将控制 数据处理的库函数,“InstallCOM1 {}”是用来初始化格式化后的 COM1 的波特率,此时中断将被 1KB 的缓冲器占用。“ToCOM1 (1,data)”是给 COM 端口传输数据。若用“IsCOM (1)”

接受任何数据，那么，“ReadCOM (1)” 就会首先得到数据。

5.6 COM1: RS485 端口

选择跳线详见 3.4 节，“InstallCOM1 {}” 是用来初始化格式化后的 COM1 的波特率，此时中断将被 1KB 的缓冲器占用。为了传输数据，“Set485DirToTransmit (1)” 必须设置正确的数据，“ToCom (1,data)” 用来传送数据，完毕后，COM1 通过命令设置接收方式，“WaitTransmit Over (1)” 和 “Set485DirToReceive (1)”；若用 “IsCOM (1)” 接受任何数据，那么，“ReadCOM (1)” 就会首先得到数据。

5.7 7 段数码管 & LED 显示控制



在 数据处理模块中有 5 位数字显示器（7 段数码管）和 1 个 LED，单路的 LED 是通过 “LEDON ()” 和 “LEDOFF ()” 来控制的，7 段数码管也可通过软件来控制。“Init 5digit

led ()” 首先初始化，“Show5DigitLed (pos,data)” 显示 7 段数码管上的数字，“pos” 是 7 段数码管的表现形式，“data” 数据必须是 0~18 之间的整数，分别是 ‘0’，‘1’，‘2’，‘3’，‘4’，‘5’，‘6’，

‘7’，‘8’，‘9’，‘A’，‘B’，‘C’，‘D’，‘E’，‘F’，‘-’ ‘.’。

“ Show5DigitLedWithDot (num,data) ” 显示浮点数字，
“Show5DigitLedSeg (num,data)” 控制。

7 段数码管上每个数字。“data” 必须是 0~7 之间的整数，在图表的正面将逐次显示。

5.8 EEPROM

EEPROM，用来存储参数和变量。比如：ID 配置，COM 端口设置，数据库。“EnableEEP ()”，“DisableEEP ()”，“ReadEEP ()”，“ProtectEEP ()”，“WriteEEP ()” 是用来读和写 EEPROM 的。

5.9 实时时钟 (RTC) & NVRAM

在实时时钟 (RTC) 和 NVRAM 中均有一个时钟芯片 DS1302，安有 1 块锂电池，保存数据长达十年左右。特征如下：

ROMDOS 支持 RTC 的时间和日期

MiniBIOS 支持 RTC 的时间和日期

NVRAM: 31 位字节读写周期没有限制

“ReadNVRAM ()”，“WriteNVRAM ()” 是用来读和写 NVRAM 的。

5.10 看门狗定时器 (WDT)

看门狗定时器可以通过命令“Enable WDT()”, “Disable WDT()”激活和退出。

WDT 激活后, 定时器开始以 1.6 秒累加, 直到重启系统。

“Refresh WDT()”是用来重设 WDT 的。

II 数据处理 软件使用说明

1 快速入门

1. 请参考数据处理模块硬件用户手册的接线说明
2. 数据处理模块的 COM4 与 PC 的 COM1(2)连接在一起
3. 进入 PC 的超级终端
4. 数据处理上电
5. PC 的超级终端将显示导入的 数据处理模块信息和 DOS 提示符 (A:\>)
6. 执行 DOS 命令, 如 A:\ DIR
7. 传送或下载文件
8. 详见用户手册中的函数说明, 连接 ROM DISK, VDISK

2 系统映射

数据处理模块是一个多功能的控制器，可以在 ROM-DOS 下进行操作。打开电源，MiniBIOS 系统初始化，ROM-DOS 进行控制。

系统存储器交换

存储器地址	存储器状态	H/W 成分
0x00000~0x3FFFF	SRAM (256K)	SRAM 空
0x40000~0x7FFFF	存储 (256K)	
0x80000~0xEFFFF	Flash 存储 (448K) ROM DISK 初始化, 0x80000	Flash 空
0xF0000~0xFBFFF	ROM-DOS (48K)	
0xFC000~0xFDFFF	存储 (8K)	
0xFE000~0xFFFFF	MiniBIOS (8K)	

3 MiniBIOS

MiniBIOS, 像它的名字一样, 是微型的 BIOS, 在 ROM-DOS 中, 它不是替代 BIOS, 而是在不请求 BIOS 的情况下为内部环境服务。

3.1 BIOS

BIOS 全名为(Basic Input Output System)即基本输入/输出系统, 是电脑中最基础的而又最重要的程序。我们把这一段程序存放在一个不需要电源的记忆体(芯片)中, 这就是平时所说的 BIOS。它为计算机提供最低级的、最直接的硬件控制, 计算机的原始操作都是依照

固化在 BIOS 里的内容来完成的。准确地说, BIOS 是硬件与软件程序之间的一个接口(虽然它本身也只是一个程序), 负责解决硬件的即时需求, 并按软件对硬件的操作要求具体执行。计算机用户在使用计算机的过程中, 都会接触到 BIOS, 它在计算机系统中起着非常重要的作用。

MiniBIOS 作为 BIOS 的一个子集, 在 ROM-DOS 中的操作和 BIOS 一样, 包括 BIOS 支持的远程控制台、定时器、BIOS 扩展名、硬件设备等。MiniBIOS 不支持软驱、硬盘、打印机、键盘和鼠标。

3.2 Mini BIOS 特点

ROM-DOS 是个非常强大实用的 DOS, 可以执行在硬盘环境下的所有的 BIOS 命令。

无硬盘系统下 MiniBIOS 命令集

名称	中断号	功能
Coprocessor ESC instruction	07H	保留
Timer 0 tick	08H	保留 (硬件)
Keyboard Input	09H	高级电源管理
Serial receive char	0BH	保留 (硬件)
Video TTY output	10H	0EH
Get equipment list	11H	
Get memory size in K	12H	
Disk I/O	13H	保留
Extended Memory Support	15H	保留
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	

4 ROM-DOS

数据处理模块中配有 ROM-DOS，ROM-DOS 是一个系统操作盘，能加载和完全运行 ROM，也能对硬盘或软盘进行操作。例如，在系统桌面上，ROM-DOS 和其他的 DOS 一样，在标准的 DOS 下运行程序。运行的程序可以存放在磁盘上或是在 ROM 中。

5 超级终端

超级终端可以控制 数据处理模块，在 DOS 下 数据处理通过超级终端可执行 Comm.exe。

5.1 Comm. exe

在 DOS 盘符下执行 comm.exe，如下：

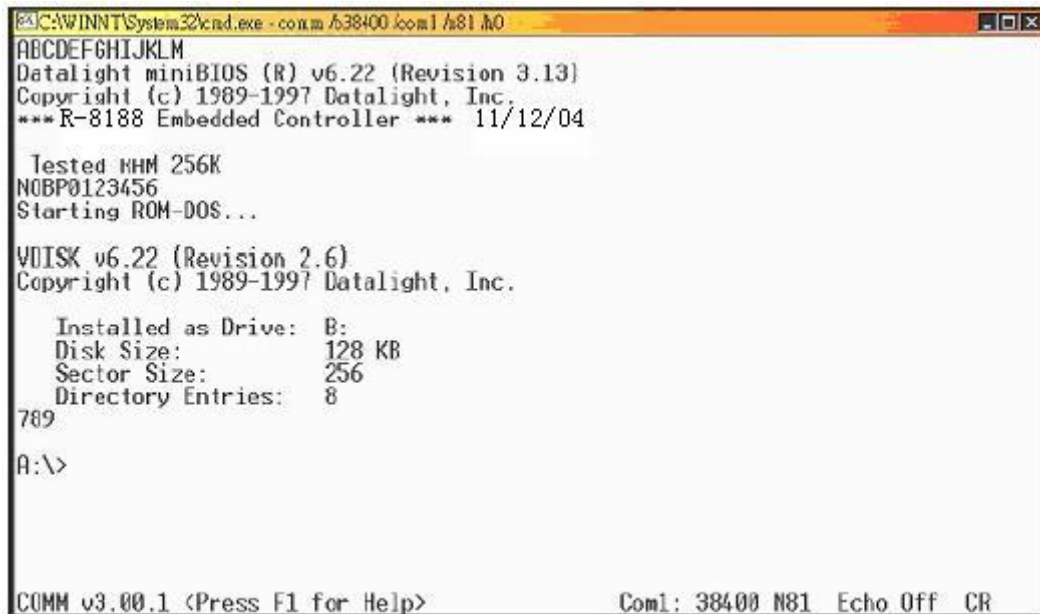
/B# 初始化波特率（300，1200，9600……115200）

/Com# COM 端口（1..4）

/Irq# IRQ 数 (3-15)
/N81, /E71 初始化变量端
/H# 0=none , 1=XON , 2=RTS

<e,g> 数据处理通过 COM1 与 PC 连接, 波特率=19200, 8 个数据位, 没有奇偶, 1 个停止位: comm.exe /com1 /b19200 /n81 /h0

数据处理模块上电, 在 DOS 盘符下超级终端显示 A:\>,例如:



```
C:\WINNT\System32\cmd.exe - com1 /b38400 /com1 /n81 /h0
ABCDEFGHJKLM
Datalight miniBIOS (R) v6.22 (Revision 3.13)
Copyright (c) 1989-1997 Datalight, Inc.
*** R-8188 Embedded Controller *** 11/12/04

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

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

Installed as Drive: B:
Disk Size: 128 KB
Sector Size: 256
Directory Entries: 8
789
A:\>

COMM v3.00.1 <Press F1 for Help> Com1: 38400 N81 Echo Off CR
```

其他热键如下:

F1 : 帮助
Alt-P: N81/E71
Alt-B: 波特率
Alt-K: 暂停
Alt-C: 清屏

Alt-T: CR/LF 平移

Alt-D: 用调制解调器拨号

Alt-X: 退出

Alt-H: 终止调制解调器拨号

PgUp: 上翻

PgDn: 下翻

简单方法:

在 Windows 下，用户可以在桌面上建立一个快捷方式，按住鼠标的右键，桌面→新建→快捷方式→命令行 COMM.EXE→下一步→选择快捷方式的名字→选择一个图标→OK。

按住鼠标的右键，属性→程序→命令行：COMM.EXE/com1/b57→杂项→后台（选择始终挂起）→OK。

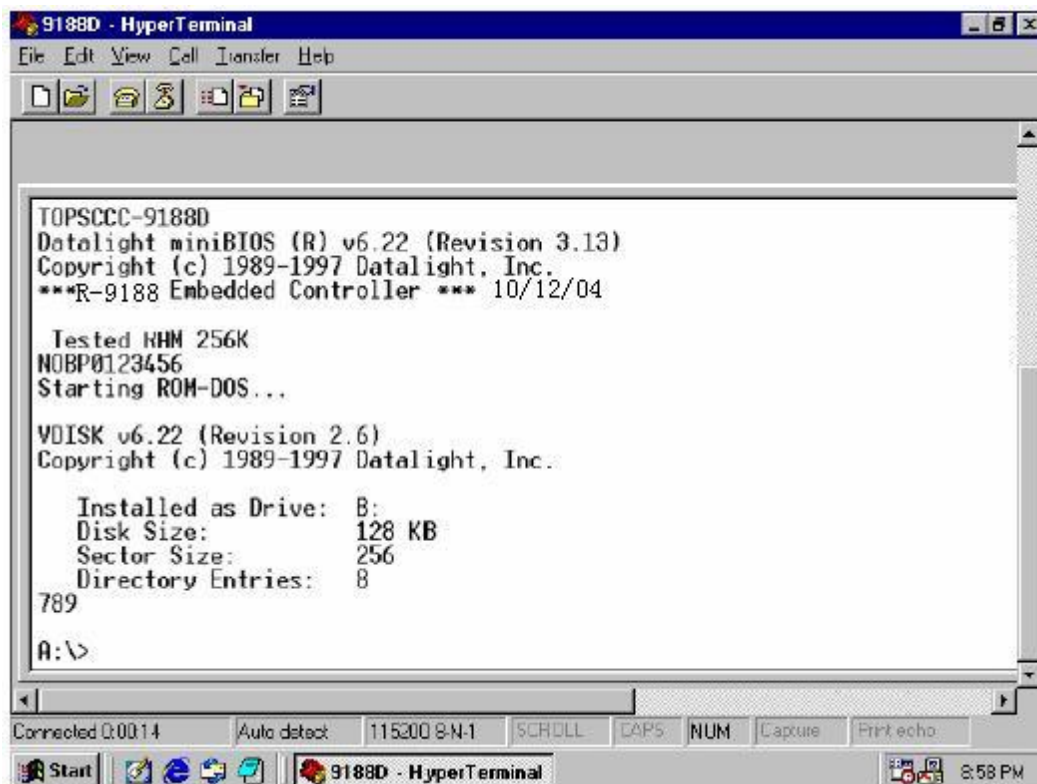
下次运行 COMM.EXE 就可直接点击快捷方式。

5.2 超级终端

“开始→程序→附件→通讯→超级终端”

选择正确的 COM 端口，波特率（57600），数据位（8），奇偶（无），停止位（1），数据流控制，选择文件名，属性（如果需要）→ OK。

显示以下图面：



6 文件传送

数据处理与 PC 连接，在 com.exe 中使用 transfer.exe 命令；
数据处理与 PC 连接，在超级终端中使用 xm.exe 命令。

Com.exe ← → transfer.exe

Hyper Terminal ← → xm.exe

6.1 Transfer.exe

1. 从目录中下载一个文件：

a. A:\>transfer.exe /s /bc <filename>

- b. Press PageDown
 - c. Enter X ----Xmodem
 - d. Enter filename (包括路径)
 - e. “RRRRRRR……”
2. 从目录中上传一个文件
- a. A:\> transfer.exe /r /bc <filename>
 - b. Press PageUp
 - c. Enter X
 - d. Enter filename (包括路径) …上传的文件名
 - e. “TTTTTTT…”

6.2 Xm.exe

1. 从目录中下载一个文件：
 - a. A:\xm.exe /s <filename>。
 - b. 在超级终端下选择：传送→接收文件→使用文件传输协议→放置文件(下载)→OK。
 - c. 文件下载完毕。
2. 从目录中上传一个文件
 - a. A:\>xm.exe /r <filename>。
 - b. 在超级终端下选择：传送→输出文件→使用文件传输协议→文件名→OK。

(此步骤在 60 秒内完成)

- c. 文件上传完毕。

7 ROMDISK.EXE

在 ROM-DOS 中 romdisk.exe 是有效的，它可以建立一个 ROM-DOS 的图案，步骤如下：

1. 在 ROM 中建立一个子目录。
2. 拷贝 ROMDISK 需要的所有文件。
3. 运行“ROMDISK.EXE”。
4. 通过“TOROM.EXE”下载文件到 数据处理模块中。
5. 重新启动 数据处理 模块，你会发现 RemoDAQ- 8188XD 模块的 A 盘中包含 ROM 所有的文件目录。

8 TOROM.EXE

TOROM.EXE 是用来传送 ROMDISK 的图象档案到 数据处理模块的 FLASH ROM 中。步骤如下：

1. 连接 数据处理模块和 PC，使用超级终端作为终端。
2. 运行 TOROM.EXE。

3. 在超级终端下选择：传送→输出文件→使用文件传输协议→文件名→OK。

4. 传输完毕后，重新启动 数据处理模块。

注意：在 数据处理模块中执行 TOROM.EXE，第 3 步必须在 60 秒内完成，否则发生时间溢出，在溢出的情况下，INIT*端子将下载程序到 FLASH ROM 中，详见 5.2 节。

5. 你就可以看到更新后的 ROM DISK 。

9 VDISK

VDISK 是一个设备驱动程序，就像从存储系统中创建一个虚拟磁盘或 RAM DISK。断电后虚拟磁盘的目录将丢失。Vdisk.sys 是一个可以在 config.sys 里配置的系统文件，格式如下：

```
Device=vdisk.sys[size[secs[dirs]]]/E
```

Size 是 RAM DISK 空间的大小，默认为 64KB；Secs 是扇区的大小，默认是 512 字节，有效大小是 128, 256, 512；Dirs 是 RAM -DISK 根目录下的目录和文件的数量，默认为 64。

例如：在 config.sys:

```
device=vidsk.sys
```

```
device=vdisk.sys 32 128 16
```

设置 64KB RAM DISK B:; 512 字节扇区（当 RAM DISK A: 存在）。

设置 32KB RAN DISK C:; 128 字节扇区，16 个文件在根目录。

10 附录

导入次序

重新上电后，MiniBIOS 将会初始化 H/W 并且 ROM-DOS 使能操作系统。导入次序如下表：、

类型	导入诊断	说明
MiniBios	A	UMCS 和 PIO 设置成功
MiniBios	B	LCS 设置成功
MiniBios	C	禁能 DMA0
MiniBios	D	禁能 DMA1
MiniBios	E	禁能硬件输入
MiniBios	F	禁能定时器和看门
MiniBios	G	无
MiniBios	H	设置中断提示
MiniBios	I	初始化 BIOS 堆栈，当前堆栈 RAM 未被测试
MiniBios	J	上电 Ram 测试
MiniBios	K	为 C 设置本地环境
MiniBios	L	建立中断向量表
MiniBios	M	初始化 BIOS 中断
MiniBios	CopyRightMessege	显示正确信息
MiniBios	N	RAM 大小检测
MiniBios	O	扫描系统环境，并且配置 BIOS，跳入
ROMDOS	B	获得控制 BIOS 的范围
MiniBios	P	INT19，跳到 ROM-DOS 入口点
ROMDOS	0	中断使能，ROM-DOS 可控，且第一指令执行
ROMDOS	1	完成代码启动，启动代码拷贝数据到 RAM 中。DOS 代码拷贝到 ROM-DOS 根目录或者使能的 RAM 中。获取数据压缩空间，启动代码重新部署 ROM-DOS 代码到存储器的顶端，在存储器的底端减压数据的大小。堆栈被设立并且未被初始化的数据为零。在此处导入失败表示 RAM 空间不足或者 ROM-DOS 中存在一个不完整的映像。
ROMDOS	2	分配最小的 DOS 构造，设立存储池且默认的构造压入 RAM 顶端，DOS 中断设立

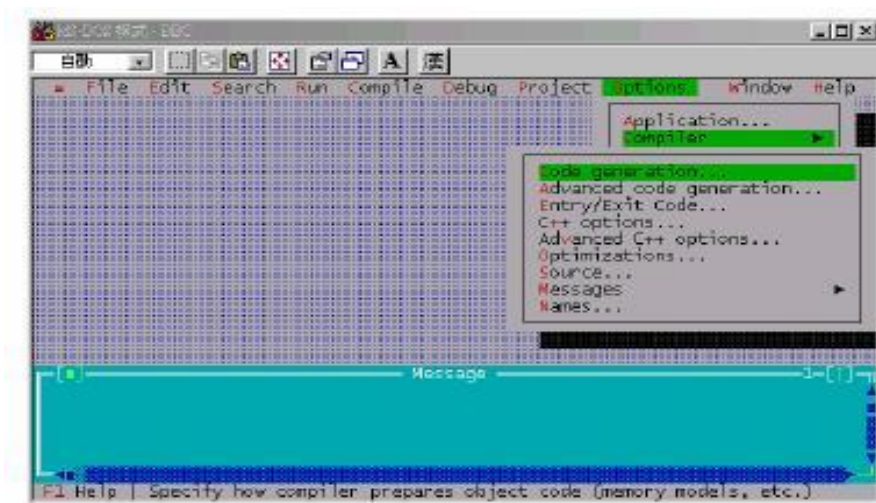
ROMDOS	3	初始化中断，如果在此导入失败将会导致 ROM-DOS 另外一个使用中断的进程被挂起。关于此类例子见看门狗定时器功能 INT21h。
ROMDOS	4	初始化 Built-in 设备，在初始化过程中获得 BIOS 中断，导入失败将导致不完整的 BIOS 中断或者不能查找系统文件。
ROMDOS	5	存在 Root PSP
ROMDOS	6	决定默认的驱动
ROMDOS	7	CONFIG.SYS 申明完成第一次处理，除了 INSTALL=命令所有的声明都被完成。PRN, AUX, 和 CON 等标准句柄被打开。
ROMDOS	8	完成内部结构，在 CONFIG.SYS 列出 TSR 的程序，声明 INSTALL=, 错误的 TSR 程序将会导致错误。
ROMDOS	9	ROM-DOS 处于高位 (DOS=HIGH), 建立 DOS 存储区如果空间足够，将拷贝到 HMA。
ROMDOS	Dos prompt or application start	标准句柄再次打开，最后的程序通过 Int 21h 调用，这些程序代表 COMMAMD.EXE 或者一个应用程序。如果在诊断 9 后 DOS 命令并未执行，那么经常是由于未找到程序（未存在于 DISK 上），文件破坏，另外一个操作系统的命令，或者是错误的请求。也可能是由于 RAM 空间不足导致。

11 数据处理函数库

数据处理函数库提供了用户实现操作端口，LED，数据存储，看门狗，定时器等功能的函数，程序员通过数据处理模块函数库，快速入门并且开发各种数据处理的应用程序。

11.1 使用 TC 和 BC

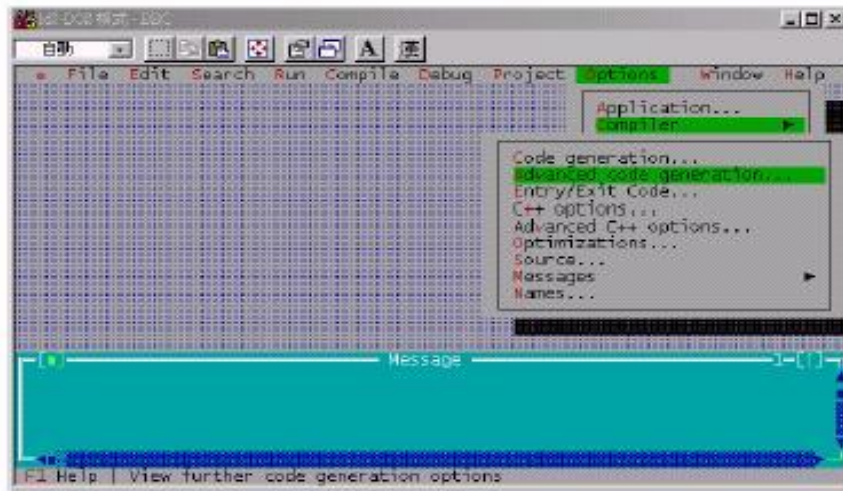
在综合的开发环境（IDE）使用 TC 或者 BC 时，用户需要建立一个包含（*.C 和 8188.LIB）的新工程。在 IDE 上必须选者的合适的编译模式和 CPU 类型，开发的程序应该是小模式并且 80186 是数据处理最好的选择。当用户使用 Floating 时应该选择“emulation”并且关闭调试信息。以下图表显示了如何配置 BC。



图表 1



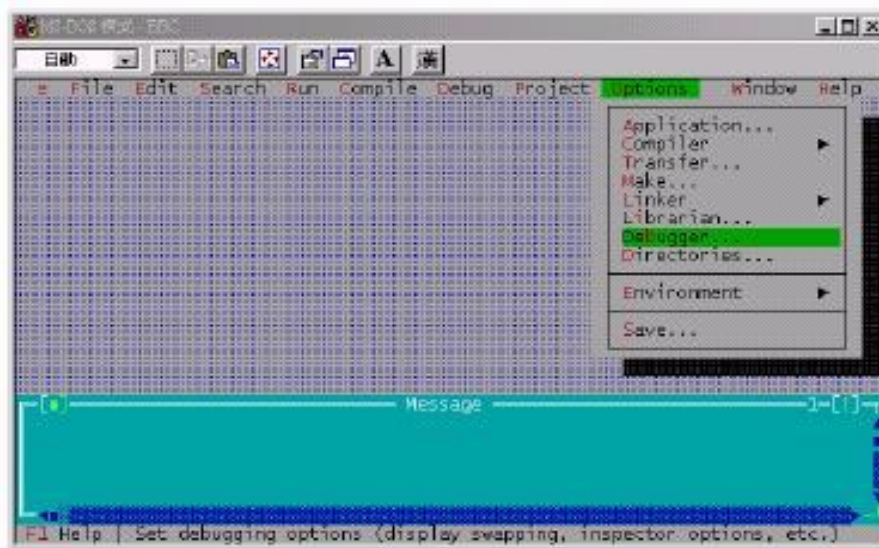
图表 2



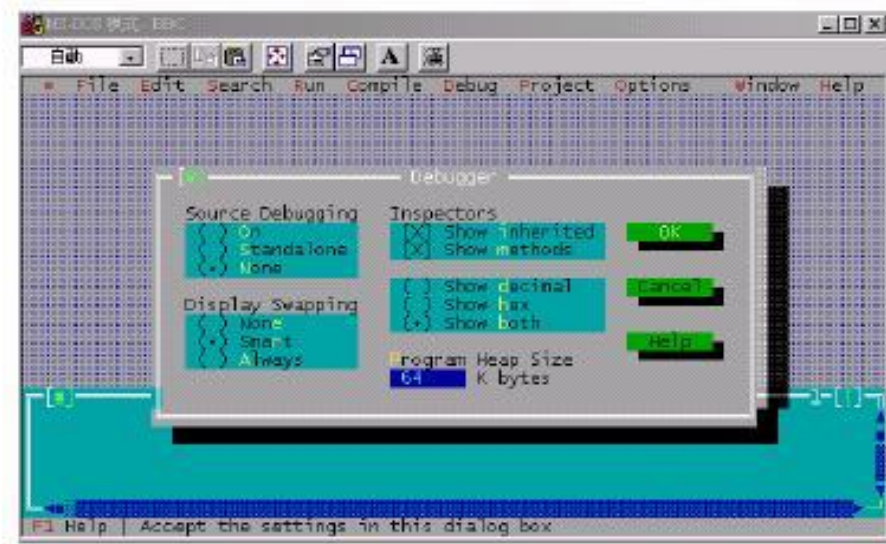
图表 3



图表 4



图表 5



图表 6

11.2 函数调用

装载端口驱动

函数名	描述
InstallCom	装载 4 个 COM 端口其中一个
InstallCom1	装载 COM1
InstallCom2	装载 COM2
InstallCom3	装载 COM3
InstallCom4	装载 COM4

卸载端口驱动

函数名	描述
RestoreCom	卸载 4 个 COM 端口其中之一
RestoreCom1	卸载 COM1
RestoreCom2	卸载 COM2
RestoreCom3	卸载 COM3
RestoreCom4	卸载 COM4

核对输入器里有数据

函数名	描述
IsCom	核对 4 个 COM 端口其中之一
IsCom1	核对 COM1

IsCom	核对 COM2
IsCom	核对 COM3
IsCom	核对 COM4

从 COM 端口的输入存储器中读取一个字节的的数据

函数名	描 述
ReadCom	从 4 个 COM 端口其中之一读取数据
ReadCom1	读取 COM1
ReadCom2	读取 COM2
ReadCom3	读取 COM3
ReadCom4	读取 COM4

向 COM 端口发送一个字节的的数据

函数名	描 述
ToCom	向 4 个 COM 端口其中之一发送数据
ToCom1	COM1
ToCom2	COM2
ToCom 3	COM3
ToCom4	COM4

向 COM 端口发送 N 个字节的数

函数名	描 述
ToComBufn	向 4 个 COM 端口其中之一发送数据
ToCom1Bufn	COM1
ToCom2Bufn	COM2
ToCom 3Bufn	COM3
ToCom4Bufn	COM4

向 COM 端口发送一串数据

函数名	描 述
ToComStr	向 4 个端口其中之一发送一串数据

ToCom1Str	COM1
ToCom2Str	COM2
ToCom3Str	COM3
ToCom4Str	COM4

COM 端口输入存储器

函数名	描述
ClearCom	清除 4 个端口其中之一
ClearCom1	COM1
ClearCom2	COM2
ClearCom3	COM3
ClearCom4	COM4

核对传输数据是否完成

函数名	描述
WaitTransmitOver	核对 4 个端口其中之一
WaitTransmitOver1	COM1
WaitTransmitOver2	COM2
WaitTransmitOver3	COM3
WaitTransmitOver4	COM4

设置 RS-485 方式

函数名	描述
Set485DirToTransmit	设置传输方式
Set485DirToReceive	设置接收方式

核对输出存储器是否为空

函数名	描述
IsCom3OutBufEmpty	COM3
IsCom4OutBufEmpty	COM4

读取输入存储器数据

函数名	描述
DataSizeInCom	4 个 COM 端口其中之一
DataSizeInCom1	COM1
DataSizeInCom2	COM2
DataSizeInCom3	COM3

DataSizeInCom4	COM4
----------------	------

LED 开关 ON/OFF

函数名	描述
LedOff	LED 关
LedOn	LED 开

与 RemoDAQ8000' S 模块之间的通信

函数名	描述
SendCmdT8000	向 RemoDAQ8000' S 发送命令
ReceiveResponseFrom8000	接收 RemoDAQ8000' S 命令

5 位数字显示器

函数名	描述
Init5DigitLed	初始化 5 位数字显示器
Show5DigitLed	显示 5 位数字显示器中的任一数字
Show5DigitLedWithDot	显示一点数字
Show5DigitLedSeg	显示 5 位数字显示器中的任一段
Set5DigitLedTestMode	设置测试方式
Set5DigitLedIntensity	设置亮度
Disable5DigitLed	禁能 5 位数字显示器
Enablr5DigitLed	使能 5 位数字显示器

NVRAM

函数名	描述
ReadNVRAM	从 NVRAM 中读一个字节的数据
WriteNVRAM	写一字节的数据到 NVRAM

EEPROM

函数名	描述
WriteEEP	向 EEPROM 中写一字节的数据

ReadEEP	读 EEPROM 中一字节的数据
EnableEEP	EEPROM 禁止写操作
ProtectEEP	EEPROM 写保护

Flash Memory

函数名	描述
FlashReadId	读 Flash Memory 类型
FlashWrite	向 Flash Memory 中写一字节的数据
FlashErase	擦除一段数据
FlashRead	读数据

看门狗定时器

函数名	描述
EnableWDT	使能看门狗定时器
RefreshWDT	更新看门狗定时器
DisableWDT	禁能看门狗定时器
IsResetBy WatchDogTimer	重新设置 数据处理看门狗定时器

定时器

函数名	描述
TimerOpen	安装定时器驱动程序
TimerClose	卸载定时器
TimerResetValue	定时器值复位
TimerReadValue	读定时器值
StopWatchReset	定时器秒表复位
StopWatchStart	定时器秒表开始计时
StopWatchStop	定时器秒表停止计时
StopWatchPause	定时器秒表暂停计时
StopWatchContinue	定时器秒表继续计时
StopWatchReadValue	读定时器秒表值
CountDownTimerStart	定时器秒表开始倒数计时
CountDownTimerReadValue	读定时器秒表倒数计时值
InstallUserTimer	安装定时器使用函数

InstallUserTimer1C	在 INIT 1C 中安装定时器使用函数
DelayTimerMs	软件延时以 1 毫秒为单位
DelayMs	硬件延时以 1 毫秒为单位
Delay_1	硬件延时以 1 毫秒为单位

标准 IO

函数名	描述
Getch4	代替 getch
Kbhit4	代替 kbhit
Ungetch4	代替 ungetch
Putch4	代替 putchar

MISC 函数

函数名	描述
InitLib	初始化 数据处理库函数
GetLib Version	库函数的版本
ReadInitPin	Init 端子状态
Hex_to_ascii	十六进制转换为 ASCII
Ascii_to_hex	ASCII 转换为十六进制
MK_FP	远程指示器

DTR/RTS

函数名	描述
SetDtrActive	设置 DTR 为传送文件状态
SetDtrInactive	设置 DTR 为暂停状态
SetRtsActive	设置 RTS 为传送文件状态
SetRtsInactive	设置 RTS 为暂停状态

格式化输出（类似于打印）

函数名	描 述
PrintCom	4 个 COM 端口其中之一
PrintCom1	COM1
PrintCom2	COM2
PrintCom3	COM3
PrintCom4	COM4

CTS/DSR 状态

函数名	描 述
GetCtsStatus	CTS 状态
GetDsrStatus	DSR 状态

11.3 C 的局限性

由于数据处理模块硬件的构造，对于 C 库函数来说存在一定的局限性，下列功能在数据处理中不能实现：

1. 图形模式函数。
2. 文本函数，即出现的数字立即在屏幕上显示，例如每秒的字符数。
3. 加密函数，例如内存扩展规范。
4. 并行口，如：biosprint 或_bios_printer 。
5. 串行口，如：bioscom 或_bios_serialcom 。
6. 8087(协处理器)有关系的函数。
7. Win32 函数。

11.4 数据处理库函数使用技巧

11.4.1 NVRAM

在数据处理模块中 NVRAM 占有 31 个字节,“ReadNVRAM ()”是从 NVRAM 中读取 1 个字节的数据 “Write -NVRAM ()”是写 1 个字节的数据到 NVRAM,举例如下:

```
Int data=0x55,data2;  
WriteNVRAM (0,data);  
Data2=ReadNVRAM (0); //data2=data=0x55
```

下面的例子是向 NVRAM 中写 2 个字节的数据

```
Int data=0xAA55;  
Char *dataptr=(char*) & data;  
WriteNVRAM (0,*dataptr);  
WriteNVRAM (1,*dataptr+1);  
  
Dataptr=(char*) & data2;  
*dataptr=ReadNVRAM (0);  
*(dataptr+1)=ReadNVRAM (1); //data2=0xAA55
```

11.4.2 EEPROM

EEPROM 在数据处理模块中的容量是 2048KB，它可分为 8 部分，每部分是 256 字节，在 EEPROM 写程序之前先打开 EEPROM 的保护锁，举例如下：

```
Int data=0x55,data2;  
  
EnableEEP ();  
  
WriteEEP (1,10,data);  
  
ProtectEEP ();  
  
Data2=ReadEEP (1,10); //data2=data=0x55
```

11.4.3 FLASH MEMORY

FLASH MEMORY 在数据处理模块中的容量是 512KB，MiniBIOS、ROM-DOS、ROM-DISK (A:盘)都存放在FLASH MEMORY 中。MiniBIOS 和ROM-DOS 的段地址为 0Xf000。ROM-DISK 占用的是 0x8000，其他的空间均可被安全利用。FLASH MEMORY 的特点是每个字节的数据都能从 1 写到 0，但不能从 0 写到 1。如果最初的数据是 0xFF，在同样位置写的任何数据都是有效的；反之，如果最初数据是 0x00，那么则是无效的。唯一的办法就是擦掉从 0 到 1 的数据，即命令“EraseFlash ()”

```
Int data=0xAA55,data2;

char *dataptr;

int *dataptr2;

dataptr=(char*) & data;

FlashWrite (0xd000,0x1234,*dataptr++);

// read data from flash memory

dataptr=(char*) & data2;

*dataptr=FlashRead (0xd000,0x1234);

*(dataptr+1)=FlashRead(0xd000,0x1235);

// another way of read data

dataptr2=(int far*)_MK-FP (0xd000,0x1234);

data2=*dataptr2;
```

11.4.4 5 位数字显示器

在数据处理中有 5 位数字显示器（7 段数码管），从左边开始第一个直到最右边的第十五个。

```
Init5DigitLed (1,9);

Show5Digitled (2,1);

Show5Digitled (3,8);

Show5Digitled (4,8);

Show5Digitled (5,13); // 'd' is 13
```

11.4.5 看门狗定时器

看门狗定时器的默认值是 1.6 秒，执行“EnableWDT”后，定时器开始计数，超过 1.6 秒时，数据处理重置看门狗定时器，即命令“RefreshWDT”。如果在定时器累加之和小于 1.6 秒之前使用“RefreshWDT”，数据处理将不会重置。

“DisableWDT”禁用看门狗定时器。

```
EnableWDT ();  
  
While (!quit) {  
    RefresWDT ();  
    // do something  
}  
  
DisableWDT ();
```

11.4.6 COM 端口

数据处理模块有 4 个通讯端口：

COM1: RS232 或 RS485 端口

COM2: RD485 端口

COM3、COM4: RS232 端口

“InstallCOM”或“Install COM1/2/3/4”：安装 COM 端口

“RestoreCOM”或“Restore COM1/2/3/4”：退出程序前卸载 COM 驱动。

“IsCOM”：核对从 COM 端口传输的数据

“ReadCOM” 从缓冲器中读得数据

“ToCOM”：向 COM 端口发送数据

```
Int port=4;

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;

    // do something

}

RestreCOM (port)
```

“PrintCom” 适合所有的 COM 端口，在使用“PrintCom”命令之前，必须先使用“InstallCom”

```
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)
```

在使用 RS-485 (COM1/2) 时, RS-485 端口的数据必须受计算机硬件的控制, RS-485 默认是接收方式, 当数据处理传输数据时, 必须设置传输方式 (命令 “Set485DirToTransmit”) 传输完毕后, “Set485DirToReceive” 必须更改设置为接收方式。“WaitTransmitOver” 系统设置更改为接收方式之前所有的数据均被传出

```
int i,potr=2  
char data [5]="$01M\r";  
InstallCOM (port,9600,8,0,1);  
Set485DirTransmit (port);  
for (i=0;i<5;i++) ToCOM (port,data [i]);  
while (!WaitTransmitOver (port));  
Set485DirToReceive (port);  
RestoreCOM (port)
```