

TSC—51/98

单片机实验开发系统
学生用实验指导书

闫有运 侯义青 编

焦作工学院电气工程系

二〇〇三年六月

目 录

第一章 TMSD 多种语言调试

1.1 TMSD 菜单方式级提示.....	2
1.2 窗口操作.....	4
1.3 对话框.....	8
1.4 各种子菜单目录项说明.....	8
1.5 TMSD 启动.....	13
1.6 程序预处理.....	13
1.7 程序调试.....	22
1.8 存贮器窗口.....	25
1.9 EPROM 空白检查、读、写操作.....	27
1.10 仿真功能.....	29
1.11 热键快速检索.....	34

第二章 TSC-51/98实验开发系统原理和组成

2.1 总体框图.....	36
2.2 元件位置图.....	37
2.3 微处理器.....	38
2.4 片外存贮器.....	41
2.5 系统仿真.....	41
2.6 I/O口地址.....	44
2.7 发光二极管电路.....	45
2.8 逻辑电平开关电路.....	45
2.9 数码显示电路.....	46
2.10 键盘电路.....	46
2.11 简单I/O口的扩展.....	47
2.12 可编程并行接口8255A及EPROM写入逻辑.....	50
2.13 串行接口逻辑.....	52
2.14 8279可编程键盘、显示控制器.....	53
2.15 A/D转换逻辑.....	57
2.16 D/A转换逻辑.....	57
2.17 串-并转换电路.....	58
2.18 音响和继电器驱动电路.....	58
2.19 步进电机驱动电路.....	59

2.20 打印机及通用扩展接口插座.....	60
2.21 TSC-51/98实验开发系统占用的资源.....	61
2.22 系统板上插座的设置.....	62

第三章 硬件及接口实验

实验一 P1口实验一.....	63
实验二 P1口实验二.....	65
实验三 简单I/O口扩展实验一(交通灯控制实验).....	66
实验四 简单I/O口扩展实验二(开关量输入).....	67
实验五 外部中断实验.....	68
实验六 定时器实验.....	69
实验七 8255A可编程并行接口实验一.....	71
实验八 8255A可编程并行接口实验二.....	72
实验九 8255A可编程并行接口实验三.....	73
实验十 数码显示实验.....	75
实验十一 8279键盘、显示接口实验一、-LED显示实验....	75
实验十二 8279键盘、显示接口实验二、-键盘实验.....	77
实验十三 8279键盘、显示接口实验三、-键盘、显示实验.	78
实验十四 串行口实验一、串并转换实验.....	80
实验十五 串行口实验二、与数据终端通讯.....	81
实验十六 串行口实验三、与P C机通讯实验.....	82
实验十七 D/A转换实验一、产生锯齿波、三角波、正弦波...	84
实验十八 D/A转换实验二、小直流电机转速控制.....	85
实验十九 A/D转换实验.....	87
实验二十 电子音响实验.....	88
实验二十一 计算机钢琴实验.....	89
实验二十二 继电器控制实验.....	90
实验二十三 步进电机控制实验.....	91
实验二十四 微型机印机实验.....	93

附录一: 8031特殊功能寄存器-表.....	97
附录二: 8255A命令和状态寄存器-表.....	101
附录三: 8279命令、数据和状态寄存器-表.....	102
附录四: TSC-51/98实验培训系统常用芯片管脚图.....	106

第一章 TMSD多种源语言调试程序

TMSD (Tsinghua Multiplex Source Language Debugger) 是清华大学计算机厂研制的多种源语言调试程序。它是我厂生产的TSC51/98实验开发系统的管理程序。目前,它支持PLM96、PLM51、ASM96、ASM51、C96五种源语言级符号调试,能自动识别实验台上是MCS51系统还是MCS98系统,也能自动识别采用的是汇编语言还是PLM语言,或C语言,从而使用户能够在统一界面下调试用户源程序。

TMSD的用户界面部分采用Turbo Vision编制,其界面风格与Borland的一些软件,如Turbo Pascal、Borland C++、Turbo Debugger等非常相似,因而用户能够很快地熟悉它和掌握应用它。该界面有以下基本特点:

1. 采用全下拉式菜单进行功能选择

采用此种菜单方式,功能清晰明了。常用的菜单项均配有热键,可通过按一个或两个键完成对应菜单项的工作。

2. 采用层叠式窗口技术

用户使用的全部窗口,如源程序窗口、目标码窗口、寄存器窗口、存储器窗口、监视窗口等均可按任意大小和位置布置在屏幕上,各窗口可部分或全部重叠。用户可方便地移动、关闭窗口。

3. 全面支持鼠标

使用鼠标的用户能够以直观而方便的方式操作TMSD。某些功能,如选择窗口、移动窗口等,使用鼠标比使用键盘简便得多。

4. 友好的帮助功能

按<F1>键都可得到有关当前操作的提示。菜单的Help条目中有详尽的系统帮助文本。

下面对TMSD做详细介绍。为了叙述方便,特做如下约定:

■ 键盘上的一个键用一对尖括号括起它的键名表示,如<F1>代表功能键F1,<F10>代表功能键F10,<ESC>代表ESC键,<Enter>代表回车键。

■ 当需要同时按下键盘上的两个键时,把这两个键名用一号连起来,再用一对尖括号括起来表示,如<Ctrl-A>代表同时按下Ctrl键和字母A键,<Alt-0>表示同时按下Alt键和数字0键。

- ↑、↓、→、← 四个键称之为方向键或光标移动键。
- PgUp、PgDn称为翻页键
- 当表示子目录中的某一目录项时，前面是子目录名，后面是子目录项的名，中间用竖线分开。例如：Memory
| Dump表示 Memory子目录中的 Dump 目录项，Register
| Disp表示 Register子目录中的 Disp目录项。

1.1 TMSD菜单方式及提示行

1.1.1 主菜单项选择

TMSD启动以后，屏幕顶部就会出现一个主菜单条，它是全局的菜单系统。这个菜单系统不管当前哪一个窗口是活动的（在活动的窗口中包含光标）都可以访问。

TMSD主菜单条形式如下：

System Memory Register Run Debug Eprom Window Help
--

主菜单条上的每项都可以引出一个下拉式菜单，通过下拉式菜单，用户可以

- 执行一个命令。
- 打开弹出式菜单，进一步进行选项。
- 打开对话框，进行会话。

选择主菜单项有下列几种方式：

- 按F10键，这时主菜单条上出现一亮条，用←或→键将亮条移至欲选菜单项上，按回车键，对应子菜单即出现。
- 每个主菜单项的英文单词中都有一个字母是以醒目的高亮显示的（一般是单词的第一个字母，例如，System中的第一个S高亮显示）。同时按下ALT键和高亮字母即选中对应的菜单项并拉下子菜单。例如，按<ALT—S>键即弹出系统（System）子菜单。
- 首先按下<F10>键，然后再按相应主菜单项的高亮字母。
- 移动鼠标使鼠标光点移至待选主菜单项，然后按一下鼠标左键，即拉出该子菜单。
- 使用→键和←键，可以从一个下拉菜单移动到另一个下拉

菜单。

1.1.2 子菜单项选择

一个子菜单项打开后，即可选择其中的子菜单项完成相应的功能。
一个打开的子菜单的菜单条象下面这个样子。

System	Memory	Register	Run	Debug	Eprom	Window	Help
Open Program							
Open Module F3							
Emulator							
Debug Mode							
Dosshell							
Quit Alt-X							

图 1.1 子菜单选择项

选择子菜单项有下列几种方法：

- 子菜单打开后，其中有一横亮条停在某子菜单项上。按 ↑ 键和 ↓ 键移动此亮条至待选的子菜单项上，按回车键，即选中该项，系统即完成相应功能。
- 子菜单项打开后，直接按待选子菜单项中的加亮字母选择该子菜单项。
- 使用 <Home> 键和 <End> 键分别移到第一个或最后一个子菜单项上。
- 移动鼠标使鼠标光点移至待选子菜单项，按下鼠标左键即选中子菜单项。

1.1.3 退出一个菜单的方法

当在一个菜单中，需退出时，有下列方法可供选择：

- 按 <Esc> 键退出低层菜单并返回到上一级菜单。

- 在一个下拉式菜单中，按 <Esc> 键则退出菜单系统并返回到活动窗口中。
- 移动鼠标使鼠标光点进入窗口，并按左键，则退出菜单系统，并使该窗口成为活动窗口。

1.1.4 菜单热键

一些常用的菜单功能项对应着一些功能键，这些功能键称为热键。热键使用户可以通过简单地按下某功能键或组合键来执行一条菜单项的功能，而无需打开菜单并选择。这些热键出现在菜单中命令栏的右边。

例如：在 System 子菜单中可以看到，其中有两条命令具有热键：

```
Open  Module      F3
Quit    ALT-X
```

这就是说用户能够在除对话框外的任意时刻通过按 <F3>，打开一个待调试程序模块，其作用与在 System 子菜单项中选择 Open Module 一样。同样，用户也可以通过按下 <ALT-X> 退出 TMSD 系统，返回 DOS。

本章的 4.11 节有全部热键清单。

1.1.5 状态行

状态行是位于屏幕最底行的横条，它的作用是：

- (1) 提示用户目前情况下可用的一部分热键。
 - (2) 指示用户即将进行的操作，或当前所处的状态。
- 例如，有一如下状态行

F1 Help	Open source module
---------	--------------------

其中竖线左边提示用户可用热键 <F1>。竖线右边提示用户下一步操作是打开源程序模块。当然，提示行中只是把目前情况下常用的热键显示出来，它不可能也没有必要显示出目前情况下全部可用的热键。

使用鼠标的用户还可以用状态行执行命令。例如，移动鼠标使光点到达 F1 Help 的位置时，敲击左键，就相当于在键盘上按下 <F1> 键，从而进入 Help 功能。

1.2 窗口操作

TMSD 采用层叠式窗口方式。这就是说，屏幕上可以同时有多个窗口，它们之间可以部分或全部重叠。

在TMSD中，有下列几种窗口：

- 源代码窗口 显示待调试的源程序
Source window

- 目标代码窗口 显示源程序对应的机器代码，并对机器
Object window 码进行反汇编，因此也称之为反汇编（
Disassembly）窗口。

- 存储器窗口 供显示或修改存储器数据
Memory window

- 寄存器窗口 供显示或修改寄存器的内容
Register window

- 监视窗口 供显示或修改用户变量或指定的存储器
Watch window 地址的值。

一个窗口通常有如下几部分构成：

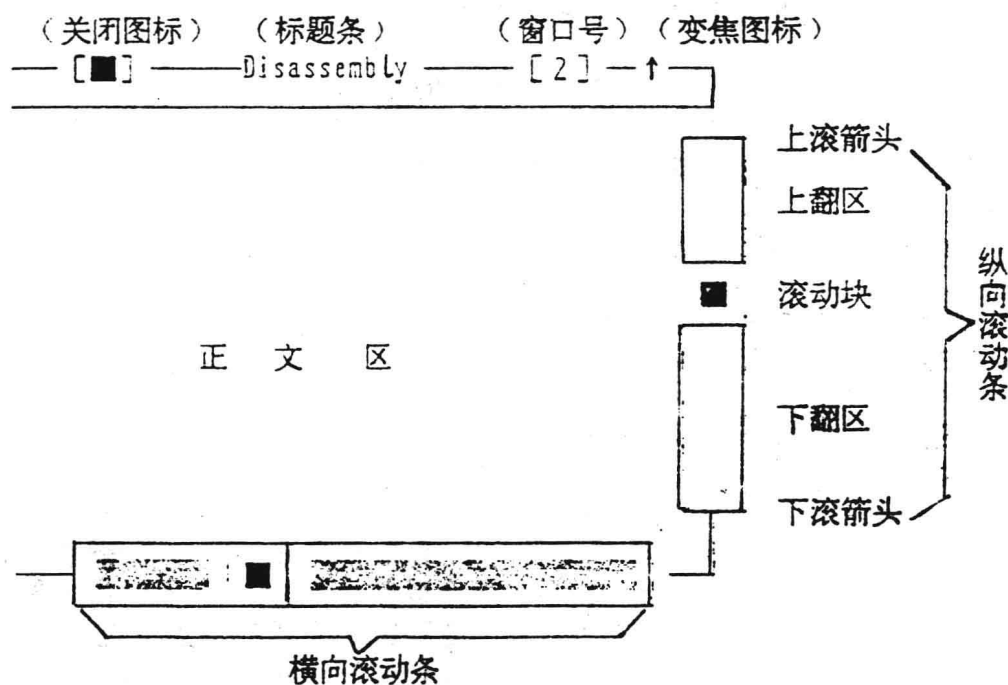


图 1.2 窗口示意图

1.2.1 窗口的生成（打开）

进入TMSD后，屏幕上通常只有一个目标代码窗口，其它窗口都是后来生成的。下面简单说明一下各窗口的生成方式，详细说明请看4.3节。

- 源代码窗口

它通过选择 System | Open Program 生成。

■ 存储器窗口

它通过选择 Memory | Dump 生成。

■ 监视器窗口

它通过选择 Debug | Add Watch 生成，也可以通过按 <Ctrl-F7> 生成。

■ 寄存器窗口

它通过选择 Register | Display 生成，也可以通过按 <F2> 生成。

1.2.2 选择当前窗口

在屏幕上的多个窗口中，每一时刻只有一个窗口是当前窗口。当前窗口又叫活动窗口。只有对当前窗口才能进行移动、放大/缩小、关闭等操作。当前窗口具有几个特征，可以很容易地与其它窗口区分：

■ 当前窗口的标题条，边缘都是高亮度显示的。

■ 只有当前窗口才有关闭图标和变焦图标。

■ 当前窗口周围是双边框，而不是单边框。

■ 如果几个窗口重叠在一起，当前窗口总是最上边一个。

在多个窗口之中选择一个窗口成为当前窗口的操作称为选择窗口。选择窗口的方法有以下几种：

■ 逐次按下 <F6>，能按一定顺序使屏幕上的窗口依次成为当前窗口。

■ 每个窗口的标题上都有窗口号。目标代码窗口、存储器窗口、寄存器窗口、监视器窗口、源代码窗口的窗口号分别依次是 1、2、3、4、5。同时按下 ALT 键和窗口号对应的数字键，就可选择相应窗口。例如，按 <ALT-2>，就使存储器窗口成为当前窗口。再如，按 <ALT-5>，则使源代码窗口成为当前窗口。

■ 使用鼠标选择窗口最为快捷。只需将鼠标光点移动至该窗口范围之内，敲击左键，该窗口就被选中。当然，如果待选窗口被其它窗口全部遮住的话，就不能直接用鼠标选择它为当前窗口。

1.2.3 移动、放大、缩小窗口

窗口可以被移动、放大或缩小。使用键盘对窗口进行移动、放大、缩小的步骤如下：

- (1) 首先使该窗口成为当前窗口。
- (2) 选择菜单 Window | Move / Resize 或按 <Ctrl-F5> 键，此时窗口边缘颜色发生变化，且变为单线框。
- (3) 按 ↑、↓、←、→ 等方向键可以移动窗口位置。按 <Shift+↑>、<Shift+↓>、<Shift+←>、<Shift+→> 键（即同时按下 Shift 和方向键）能够改变窗口大小。
- (4) 待窗口位置和大小满意后，按回车键或 ESC 键结束修改。

使用鼠标移动窗口时，首先将鼠标光点移动至窗口的标题栏上，（如寄存器窗口的 Register），按住鼠标左键不放手，且同时移动鼠标，则窗口随之移动，移动至满意位置后松开左键即可。改变窗口大小时，将鼠标光点停在窗口的右下角，按下左键不放开，再移动鼠标时，可同时拖动窗口的两条边，拖动至大小符合要求时，再松开左键即可。

1.2.4 窗口变焦

有时候，需要临时将窗口放至最大，以便观察其内容，然后再恢复其原来大小。这样的操作可利用窗口变焦完成。

使用键盘对窗口变焦操作步骤如下：

- (1) 选择该窗口为当前窗口。
- (2) 选择菜单 Window | Zoom 或按 <F5>，该窗口即放至最大。
- (3) 再按一次 <F5>，窗口又恢复原来大小。

使用鼠标对窗口变焦时，首先将鼠标光点移至窗口的变焦图标（[↑]）上，敲击左键，该窗口即放至最大，同时变焦图标变为（[↓]）。以后再点一下变焦图标，窗口即恢复原来大小。

1.2.5 滚动条使用

许多窗口在一屏之内显示不下所有的正文内容，这时用户可通过按 ↑、↓、←、→ 等方向键、PgUp、PgDn、Home、End 等键来滚动窗口内容。对于使用鼠标的用户，还有一种方便的方法用于滚动窗口的内容，就是使用滚动条。

以图 4.2 所示的窗口为例，窗口中有横向滚动条和纵向滚动条各一个，分别用于上下或左右滚动窗口内容。纵向滚动条由上滚箭头、

下滚箭头、上翻区、下翻区和滚动块共五部分组成。在上滚箭头上敲击鼠标左键，可使窗口内容上滚一行，在下滚箭头上敲击鼠标左键，可使窗口内容下滚一行，在上翻区敲击鼠标左键，可使窗口内容上翻一页（相当于使用PgUp键），在下翻区敲击鼠标左键，可使窗口内容下翻一页（相当于使用PgDn键）。在滚动块上按住鼠标左键不放开，再上、下移动鼠标，左键放开后文本改变，滚动块在滚动行中的位置相当于光标行在整个文本中的位置。关于横向滚动条的使用，可仿照使用纵向滚动条处理，这里就不详述了。

1.2.6 关闭窗口

当一段时间内不需要某个窗口时可以将它关闭，使之从屏幕上消失。使用键盘关闭一个窗口的步骤如下：

(1) 选择待关闭窗口为当前窗口。

(2) 选择菜单Window | Close或按<ALT-F3>，该窗口即被关闭。使用鼠标时，将鼠标光点移至当前窗口的关闭图标（），按鼠标左键，该窗口即被关闭。

1.3 对话框

对话框是一个特殊的窗口。它一般在需要临时通知用户一些消息，或者从用户得到一些输入信息时产生。和一般窗口比较，对话框有一些特殊之处：

- 外观上对话框没有窗口号，也不能改变大小。但它可象其它窗口一样移动、关闭。
- 每当对话框出现时，它自动成为当前窗口。同时，它封锁了与之无关的其它功能。例如，当一个对话框出现后，你不能选择其它窗口，不能选择菜单等。总之，在对话框完成其操作之前，不能进行其它操作。
- 对话框内一般有一些标有文字的小方块，称为按钮。用户可以用鼠标点这些按钮，完成相应的工作，也可用TAB键在各个按钮之间切换到要选择的按钮上再按下回车。一般对话框总有OK和Cancel两个按钮，对应“接受”和“取消”操作。一般按<ESC>键总相当于按下Cancel按钮。关闭对话框也相当于按Cancel。
- 对话框的任务完成后，自动从屏幕上消失。

1.4 各子菜单目录项说明

1.4.1 System（系统）子目录

- **Open Program** 打开一个源程序。该源程序必须被预处理过。（有关程序预处理，请参看第4.6节）只输入文件名，不输入扩展名。执行此命令后源文件窗口打开，同时将目标文件装入 RAM，目标代码窗口显示与源代码窗口相应的目标代码。
- **Open Module** 打开源代码中的一个模块。一般在调试源语句时，如果采用跟踪调试，则遇上模块时会自动打开调用的模块。此命令主要用于非跟踪的情况下，用户在自己需要时打开想看的模块。
- **Emulator** 设置仿真参数。主要是确定是否仿真。如果采用仿真，是否出借存储器。实验台上的存储器以 16K 字节为一块出借给用户，并确定每一块出借给用户的存储器是供用户作为 RAM 使用，还是作为 ROM 使用（对于 MCS98，只能做 RAM 使用）。
- **Debug Mode** 确定是源代码调试方式，还是目标代码调试方式，二者可以切换。系统 TMSD 启动后，如没有打开源程序，自动进入目标代码调试方式。打开源程序后，自动进入源代码调试方式。
- **Status** 显示当前系统状态。
- **DosShell** 临时退出 TMSD，返回 Command 控制下，进行用户希望的操作，如编辑，列目录等操作。操作结束后，使用 EXIT DOS 命令返回到 TMSD 系统。
- **Quit** 退出 TMSD 系统，返回 DOS 系统。

1.4.2 Memory（存储器）子目录

此子目录管理与实验台上的 RAM 有关的操作。

- **Dump** 显示从指定地址开始的 RAM 内容。
- **Fill** 向实验台上的 RAM 块填充新的内容。填充方式，可以是以数字方式，各数字用空格分开，也可以是字符串方式，字符串需用单引号或双引号括起来，例如“abcd”。

- Move 把存储区中一个块的内容传送到另一个块去。
- Compare 比较两块存储区的内容，显示出两块中内容不一致的第一个地址及其内容。
- Load 将一个.BIN文件装入RAM。指定文件名时需要键入其文件名与键入扩展名。如指定的长度大于文件长度，则实际装入长度为文件长度。如指定的长度小于文件长度，则装入长度为指定的长度。如不指定长度，直接键入回车，则按文件大小装入。指定装入地址时，指定值可以小于4000（十六进制）。但小于4000的部分，只能装入与中断向量区有关的部分，小于4000H的其余部分不予装入（与监控程序冲突）。
- Save 将一块存储区内容保存在文件中。
- Save Disasm Uode 以符号反汇编形式（.ASM）或列表形式（.LST）将存储区内的一段内容以文件形式保存在磁盘上。
- Disassemble from 对存储区的内容进行反汇编，显示在目标码窗口内。
- Assemble 从指定地址进行简单的汇编。

1.4.3 Register（寄存器）子目录

- Display 显示有关寄存器的内容。
- Add Register 增加显示的寄存器。
- Delete Register 删掉指定的寄存器。
- Modify PC 修改程序计数器。
- Modify PSW 修改程序状态字。
- Write Register 修改寄存器。
- Set User Register 此功能用于汇编时使用。给用户建立一组寄存器，AX、BX、CX、DX。其中每个寄存器包含高八位和低八位。例如AX包含AH、AL。用户在汇编时可以使用这些符号。只用于MSC98。

1. 4. 4 RUN (运行) 子目录

- Go 从当前PC启动程序运行。
- Step 单步执行。在源代码调试方式下，每次执行一条源语句。在目标代码调试方式下，每次执行一条指令。但在遇上子程序或过程时，不跟踪进子程序或过程中。
- Trace 跟踪执行，能够跟踪进过程或子程序。
- Run to Cursor 运行到光标。从当前PC运行到光标所在的行。
- Set PC to Cursor 将程序计数器PC置到光标所在行。
- Reload 将源程序重新装入。此时TMSD使用上一次指定的文件名。
- Restart 将PC置到程序入口处。
- Demonstrate 以流览方式执行程序。用户能够从屏幕上看到程序的执行过程。

1. 4. 5 Debug (调试) 子目录

此子目录主要处理与断点和变量监视有关的命令。

- Add Watch 在 Watch (监视) 窗口增加被监视的变量。
- Delete Watch 删掉被指定的变量，使其不出现在 Watch 窗口中。
- Delete All Watch 删掉所有变量，使其不出现在 Watch 窗口中。
- Toggle Breakpoint 断点乒乓开关。光标所在行如果是断点，则删除此断点。光标所在行如果不是断点，则将该行置为断点。
- Clear All Breakpoint 删除全部断点。
- List Breakpoint 列出全部断点。
- Evaluate 表达式计算。此功能可用于进行计算。

例如计算 28×3275 ，则显示出168（舍掉小数位）。除此以外，还可以进行变量运算。

1.4.6 EPROM子目录

本子目录处理与读、写、校验 EPROM 有关的操作。

- Check 检查 EPROM 是否为空。
- Read Eprom 读 EPROM 的内容。
- Program 对 EPROM 进行编程，即写 EPROM。
- Compare 将一个 EPROM 的内容读出来，与指定文件的内容比较。

1.4.7 Window

Window 子目录管理各窗口的放大、缩小、移动、变焦、关闭等操作。

- Size/Move 对当前窗口进行移动或修改大小。
- Zoom 对当前窗口进行变焦操作。
- Close 关闭当前窗口。
- Next 使下一个窗口成为当前窗口。
- Previous 使上一个窗口成为当前窗口。所谓上一个窗口和下一个窗口指的是 TMSD 中规定的顺序。五个窗口的轮转顺序是 5、4、3、2、1、5、……。例如对 4 号窗口（监视窗口）来讲，3 号窗口（寄存器窗口）是下一个窗口，5 号窗口（源代码窗口）是上一个窗口。但屏幕上不满 5 个窗口时，则只是在轮转顺序中去掉没打开的窗口而已。例如，屏幕上只有 1、3、5 号窗口时，轮转顺序为 5、3、1、5、……。
- Close All 关闭所有窗口。

1.4.8 Help（帮助）子目录

Help 子目录中只有两个子菜单项。选择 Help | Contents 子菜单项时，显示出 Help 主屏幕，通过 ↑、↓、→、← 光标键将光标移至待选项内后或移动鼠标光点至待选项内后，待选项则以反视频显示，按 <Enter> 键或鼠标左键则显示出相应内容。用户可通过 ↑、↓、→、←、<PgUp>、<PgDn>、<Home>、<End> 等键翻阅内容，翻阅完毕后按

<Esc>退出。选择Help|About子菜单项时，则屏幕上出现TMSD字样的标题屏幕，按<Enter>键或<Esc>键则退出。

除了Help子目录外，按<F1>键，则屏幕上会出现有关的帮助信息，同样可以使用上段中所提到的各种键翻阅这些信息。结束后按<Esc>键退出帮助功能。

以上简单叙述了TMSD中各子目录项功能。用户可以从其中浏览各子目录项的作用及使用方法。有些比较复杂的子目录项的使用，还会在本章今后各节中陆续介绍。

1.5 TMSD启动

TMSD启动是在DOS状态下进行的。当屏幕上出现了DOS提示符后，键入命令：

```
TMSD <Enter>
```

即进入TMSD系统。命令中的<Enter>是指回车键，当屏幕上出现TMSD字样时，用户应打开TSC-51/98实验台电源，在实验台电源已打开的情况下应按一下实验台上的Reset键。然后在主机上按回车键。这时屏幕上提示用户选择通讯口，用户应根据实际连接情况选择一个通讯口，选中后，再按一次<Enter>键，即进入TMSD系统。

TMSD启动时可以有一个开关，供选择通讯口。例如用户键入命令
TMSD/1 <Enter>

则表示使用通讯口COM1，启动后TMSD不再显示通讯口选择屏幕。开关值可以为1、2、3或4，分别代表COM1、COM2、COM3、COM4。

另外，TMSD命令后可以带被调试的文件名。

例如：键入命令

```
TMSD/1 PLC24 <Enter>
```

则表示使用COM1进行主机和实验台之间的通讯，被调试程序为PLC24.PLM或PLC24.ASM。为了便于识别，特作如下约定，汇编源文件使用扩展名.ASM，PL/M源文件使用扩展名.PLM，在当前目录下不能有同名的汇编源文件和PL/M语言源文件。例如当前目录下不能既有PLC24.PLM，又有PLC24.ASM。

1.6 程序预处理

程序必须经过预处理才能被TMSD调试，程序预处理是在进入TMSD之前进行的。

1.6.1 文件扩展名约定

在程序预处理过程中，会产生多种文件，以下给出文件扩展名的约定：

■ .PLM PL/M语言源文件

■ .ASM 汇编语言源文件

- .LST 列表文件
- .OBJ 目标文件
- .M96 MCS96目标映象文件
- .M51 MCS51目标映象文件
- .HEX 十六进制文件
- .BIN 二进制文件
- 无扩展名 Intel格式可执行目标文件

1.6.2 程序预处理步骤

- (1) 使用文本编辑程序编辑源文件。
 - (2) 使用文件过滤程序FILTER.EXE对源文件进行预处理。不经过预处理的文件，编译时会产生错误。此点请用户务必注意。
 - (3) 使用高级语言编译程序（例如PLM51、PLM96、ASM51、ASM96）对源文件进行编译，编译时应使用DB（Debug）和SB（Symbol）控制项，以产生与调试有关的符号表等信息。编译结果产生.OBJ文件和.LST文件。
 - (4) 使用连接和定位程序（RL96或RL51）连接目标文件和库文件（如有的话），产生Intel格式可执行文件（无扩展名）和目标映象文件（.M96或.M51）。
 - (5) 使用目标码转换程序OH.EXE，将无扩展名的Intel无扩展名的可执行文件转换成十六进制（.HEX）文件。
 - (6) 使用转换程序THCON.EXE将十六进制（.HEX）文件转换成TMSD可执行的二进制（.BIN）文件。
- 只有经过上述预处理的程序才能被TMSD所调试。

对MCS51来讲，如果CODE段或XDATA段是可以浮动的，则在步骤(4)和(5)之间要加入以下二个步骤：

- (4.1) 使用CONVERT程序，对(4)产生的.M51文件进行处理，产生文件T1234567，T1234567包含了各模块未来的浮动地址。
- (4.2) 重新使用RL51连接、定位程序连接目标文件和库文件（