

To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: <http://www.renesas.com>

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (<http://www.renesas.com>)

Send any inquiries to <http://www.renesas.com/inquiry>.

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: “Standard”, “High Quality”, and “Specific”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as “Specific” without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as “Specific” or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is “Standard” unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.
 - “Standard”: Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.
 - “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.
 - “Specific”: Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.



用户手册

RA78K0S

汇编包 Ver. 1.40 或更高版本

操作

目标设备

78K0S 系列

文档编号: U16656CA1V0UM00 (第 1 版)

发布日期: 2007 年 8 月 N CP(K)

© NEC Electronics Corporation 2007

日本印刷

[备忘录]

Windows 和 Windows NT 是微软公司在美国和/或其他国家的注册商标或商标。

Unix 是在美国和其他国家的注册商标，并通过 X/Open 有限公司独家许可。

PC/AT 是国际商用机器公司的商标。

HP9000 series 700 和 HP-UX 是惠普公司的商标。

SPARCstation 是 SPARC 国际公司的商标。

Solaris 和 SunOS 是 Sun 微系统公司的商标。

- 本档所刊登的内容有效期截至 2007 年 8 月。将来可能未经预先通知而更改。在实际进行生产设计时，请参阅各产品最新的数据表或数据手册等相关资料以获取本公司产品的最新规格。
- 并非所有的产品和/或型号都向每个国家供应。请向本公司销售代表查询产品供应及其他信息。
- 未经本公司事先书面许可，禁止复制或转载本文件中的内容。否则因本档所登载内容引发的错误，本公司概不负责。
- 本公司对于因使用本文件中列明的本公司产品而引起的，对第三者的专利、版权以及其它知识产权的侵权行为概不负责。本文件登载的内容不应视为本公司对本公司或他人所有的专利、版权以及其它知识产权作出任何明示或默示的许可及授权。
- 本文件中的电路、软件以及相关信息仅用以说明半导体产品的运作和应用实例。用户如在设备设计中应用本文件中的电路、软件以及相关信息，应自行负责。对于用户或其他人因使用了上述电路、软件以及相关信息而引起的任何损失，本公司概不负责。
- 虽然本公司致力于提高半导体产品的质量及可靠性，但用户应同意并知晓，我们仍然无法完全消除出现产品缺陷的可能。为了最大限度地减少因本公司半导体产品故障而引起的对人身、财产造成损害（包括死亡）的危险，用户务必在其设计中采用必要的安全措施，如冗余度、防火和防故障等安全设计。
- 本公司产品质量分为：

“标准等级”、“专业等级”以及“特殊等级”三种质量等级。

“特殊等级”仅适用于为特定用途而根据用户指定的质量保证程序所开发的日电电子产品。另外，各种日电电子产品的推荐用途取决于其质量等级，详见如下。用户在选用本公司的产品时，请事先确认产品的质量等级。

“标准等级”：计算机，办公自动化设备，通信设备，测试和测量设备，音频·视频设备，家电，加工机械以及产业用机器人。

“专业等级”：运输设备（汽车、火车、船舶等），交通用信号控制设备，防灾装置，防止犯罪装置，各种安全装置以及医疗设备（不包括专门为维持生命而设计的设备）。

“特殊等级”：航空器械，宇航设备，海底中继设备，原子能控制系统，为了维持生命的医疗设备、用于维持生命的装置或系统等。

除在本公司半导体产品的数据表或数据手册等资料中另有特别规定以外，本公司半导体产品的质量等级均为“标准等级”。如果用户希望在本公司设计意图以外使用本公司半导体产品，务必事先与本公司销售代表联系以确认本公司是否同意为该项应用提供支持。

(注)

(1) 本声明中的“本公司”是指日本电气电子株式会社（NEC Electronics Corporation）及其控股公司。

(2) 本声明中的“本公司产品”是指所有由日本电气电子株式会社所开发或制造，或为日本电气电子株式会社（定义如上）开发或制造的产品。

区域信息

本文档中的某些信息可能因国家不同而有所差异。用户在使用任何一种 NEC 产品之前，请与当地的 NEC 办事处联系，以获取权威的代理商和发行商信息。请验证以下内容：

- 设备的可用性
- 定货信息
- 产品发布进度表
- 相关技术资料的可用性
- 开发环境要求（例如：要求第三方工具和组件，主计算机，电源插头，AC 供电电源等）
- 网络要求

此外，对于商标、注册商标、出口限制条款和其他法律规定，不同的国家也有不同的要求。

详细信息请联系：

（中国区）

网址：

<http://www.cn.necel.com/>

<http://www.necel.com/>

[北京]

日电电子（中国）有限公司
中国北京市海淀区知春路 27 号
量子芯座 7, 8, 9, 15 层
电话: (+86)10-8235-1155
传真: (+86)10-8235-7679

[深圳]

日电电子（中国）有限公司深圳分公司
深圳市福田区益田路卓越时代广场大厦 39 楼
3901, 3902, 3909 室
电话: (+86)755-8282-9800
传真: (+86)755-8282-9899

[上海]

日电电子（中国）有限公司上海分公司
中国上海市浦东新区银城中路 200 号
中银大厦 2409-2412 和 2509-2510 室
电话: (+86)21-5888-5400
传真: (+86)21-5888-5230

[香港]

香港日电电子有限公司
香港九龙旺角太子道西 193 号新世纪广场
第 2 座 16 楼 1601-1613 室
电话: (+852)2886-9318
传真: (+852)2886-9022
2886-9044

上海恩益禧电子国际贸易有限公司
中国上海市浦东新区银城中路 200 号
中银大厦 2511-2512 室
电话: (+86)21-5888-5400
传真: (+86)21-5888-5230

引言

本手册旨在帮助使用 RA78K0S 开发软件的用户了解 RA78K0S 系列汇编包（下文简称“RA78K0S”）中各个程序的功能及正确使用该程序包的方法。

本手册不包括语言部分，例如汇编器指令的表达式及在 RA78K0S 中使用的源程序。因此，在阅读本手册前，请阅读 **RA78K0S 汇编包语言用户手册(U16657E)**（下文简称“语言”）

本手册的内容是针对 RA78K0S 的 Ver. 1.40 或更高版本的。

[读者对象]

RA78K0S 适用于了解要开发的微处理器（78K0S 系列）的功能和指令的用户。

[组织编排]

本手册由以下 12 章及附录组成。

第 1 章 概述

概括介绍 RA78K0S 在微处理器软件开发中的作用及其功能。

第 2 章 产品概况与安装

介绍 RA78K0S 提供的程序文件名与操作环境。

第 3 章 执行 RA78K0S

用示例程序介绍开发软件的步骤。

本章的目的是提供一个实际使用各个程序的机会。那些希望体验 RA78K0S 操作的读者应该阅读本章。

第 4 章 结构化汇编

第 5 章 汇编器

第 6 章 连接器

第 7 章 目标转换器

第 8 章 库管理程序

第 9 章 表转换程序

第 10 章 程序输出列表

介绍每个程序输出列表的格式。

第 11 章 高效使用 RA78K0S

介绍一些 RA78K0S 优化使用的措施。

第 12 章 错误信息

介绍每个程序输出的出错信息。

附录

介绍一系列程序选项、示例程序及使用 RA78K0S 的注意事项。

本手册未详细介绍指令集。

有关指令的内容，请参阅要开发的微处理器的用户手册。

[手册使用方法]

建议首次使用汇编的读者从**第1章 概述**开始阅读。那些对汇编器具有基本了解的读者可以跳过本章。在使用 RA78K0S 前，请阅读**第3章 执行 RA78K0S**。在熟悉各个程序的操作后，读者可以开始使用**附录**中的列表。

[注意事项]

在本手册中，假设使用 PC-9800 系列个人计算机或 IBM PC/AT™ 兼容机作为主机。当使用 HP9000 系列 700™ 或 SPARCstation™ 系列时，请注意它们的以下区别。

- 文件名格式不同。
 - EWS 版本，如 HP9000 系列 700 没有可执行文件的扩展名 .exe 。
 - EWS 版本，如 HP9000 系列 700 的批处理文件扩展名 .bat 变为了 .sh 。
 - 大写字母文件名对于 EWS 版本，如 HP9000 系列 700 则是小写字母。
- 本手册中说明的执行示例和环境设备不同。

[约定]

本手册中使用了以下符号及缩写。

:::	表示同一表达式被重复
[]:	方括号中的项目可以省略。
'':	'' (引号) 中的字符将按照它们显示的内容列出。
< >:	表示窗口名或对话框名。
" ":	"" (双引号) 中的字符是要求读者参考的章节、段落或图标标题。
___:	表示要输入到使用示例中的要点或字符。
□:	表示一个空格。
Δ:	表示一个或多个空白符或 TAB。
∇:	表示 0 个或多个空白符或 TAB (即空白可以省略)。
/:	表示字符间的断开。
~:	表示连续。
[↵]:	表示按回车键。
注:	正文中用注标记的脚注。
注意事项:	需要特别关注的信息
备注:	补充信息

[相关文档]

下面列出了与本手册相关的文档。

本手册中的相关文档可能包括前期版本。不过，前期版本不是这样标记的。

与开发工具（用户手册）相关的文档

文档名称		文档编号
CC78K0S C 编译器 Ver. 1.50 或更高	操作	U16654E
	语言	U16655E
RA78K0S 汇编包 Ver. 1.40 或更高	操作	本手册
	语言	U16657E
	结构化汇编语言	U11623E
SM78K0S 系统仿真器	操作	待准备
ID78K0S-NS 集成调试程序 Ver. 2.52 或更高	操作	U16584E
78K0S 系列 OS MX78K0S	原理	U12938E
PM+ Ver. 5.10		待准备

注意事项 上述相关文档的内容有可能更改，恕不另行通知。设计您的系统时，请确保使用各个文档的最新版本。

目录

第 1 章 概述	13
1.1 汇编器概述	13
1.1.1 什么是汇编器?	14
1.1.2 什么是浮动汇编器?	18
1.2 RA78K0S 功能概述	20
1.2.1 使用编辑器创建一个源模块文件.....	21
1.2.2 结构化汇编预处理程序.....	22
1.2.3 汇编器	23
1.2.4 连接器	24
1.2.5 目标转换器	25
1.2.6 库管理程序	26
1.2.7 表转换程序	27
1.2.8 调试器	28
1.3 程序开发前的提示	29
1.3.1 RA78K0S 的最高性能指标.....	29
1.4 RA78K0S 的功能	30
第 2 章 产品概况与安装	31
2.1 主机与提供介质	31
2.2 安装	32
2.2.1 Windows 版本的安装	32
2.2.2 UNIX 版本的安装.....	32
2.3 安装设备文件	33
2.3.1 Windows 版本的安装	33
2.3.2 UNIX 版本的安装.....	33
2.3.3 设备文件的注册表注册.....	33
2.4 目录配置	34
2.4.1 Windows 版本目录配置	34
2.4.2 UNIX 版本目录配置.....	35
2.5 卸载步骤	36
2.5.1 Windows 版本的卸载	36
2.5.2 UNIX 版本的卸载.....	36
2.6 环境设置	37
2.6.1 环境变量	37
2.6.2 源文件中的汉字代码.....	37
第 3 章 执行 RA78K0S	38
3.1 准备执行 RA78K0S	39
3.1.1 示例程序	39
3.1.2 示例程序的结构.....	42
3.2 RA78K0S 的执行步骤	42
3.3 ST78K0S 的执行步骤	47
3.4 从命令行中完成汇编、连接与目标程序转换 (DOS 提示, EWS)	53
3.5 使用参数文件	57
第 4 章 结构化汇编器	58

4.1	结构化汇编器的 I/O 文件.....	58
4.2	结构化汇编器的功能	59
4.3	结构化汇编器启动	60
4.3.1	结构化汇编器启动.....	60
4.3.2	执行启动和结束消息.....	62
4.4	结构化汇编器选项	64
4.4.1	结构化汇编器选项的类型.....	64
4.4.2	结构化汇编器选项说明.....	65
4.5	在 PM+中设置选项.....	78
4.5.1	设置选项方法	78
4.5.2	选项	79
第 5 章	汇编器	81
5.1	汇编器的 I/O 文件.....	81
5.2	汇编器的功能	83
5.3	汇编器启动	84
5.3.1	汇编器启动	84
5.3.2	执行启动和结束消息.....	86
5.4	汇编器选项	88
5.4.1	汇编器选项类型.....	88
5.4.2	汇编器选项的优先级.....	90
5.4.3	汇编器选项说明.....	91
5.5	在 PM+中设置选项.....	120
5.5.1	设置选项	120
5.5.2	选项	122
第 6 章	连接器	124
6.1	连接器的 I/O 文件.....	124
6.2	连接器的功能	125
6.3	存储空间和存储区	125
6.4	连接指令	126
6.4.1	指令文件	127
6.4.2	存储器指令	128
6.4.3	程序段位置指令.....	130
6.5	启动连接器	133
6.5.1	启动连接器	133
6.5.2	执行开始结束信息.....	135
6.6	连接器选项	137
6.6.1	连接器选项的类型.....	137
6.6.2	连接器选项的先后顺序.....	139
6.6.3	连接器选项说明.....	140
6.7	在 PM+中设置选项.....	166
6.7.1	选项设置方法	166
6.7.2	选项	169
第 7 章	目标转换器	171
7.1	目标转换器的 I/O 文件	172

7.2	目标转换器的功能	173
7.3	目标转换器的启动	177
7.3.1	目标转换器的启动.....	177
7.3.2	执行开始结束信息.....	179
7.4	目标转换器选项	181
7.4.1	目标转换器选项的类型.....	181
7.4.2	目标转换器选项说明.....	182
7.5	在 PM+中设置选项.....	191
7.5.1	选项设置方法	191
7.5.2	选项	194
第 8 章	库管理程序	195
8.1	库管理程序的 I/O 文件.....	195
8.2	库管理程序的功能	197
8.3	库管理程序的启动	199
8.3.1	库管理程序的启动.....	199
8.3.2	库管理程序执行开始和执行结束信息.....	201
8.4	库管理程序选项	202
8.4.1	库管理程序选项类型.....	202
8.4.2	库选项的说明	203
8.5	子命令	209
8.5.1	子命令类型	209
8.5.2	子命令说明	209
8.6	在 PM +中设置选项	218
8.6.1	选项设定方法	218
8.6.2	选项	220
第 9 章	表转换程序	221
9.1	表转换程序 I/O 文件.....	222
9.2	表转换程序功能	223
9.3	表转换程序的启动	226
9.3.1	表转换程序的启动.....	226
9.3.2	表转换程序执行开始和结束信息.....	228
9.4	表转换程序选项	229
9.4.1	表转换程序选项类型.....	229
9.4.2	表转换程序选项的说明.....	230
9.5	在 PM +中设置选项	236
9.5.1	选项设置方法	236
9.5.2	选项	238
第 10 章	程序输出列表	239
10.1	汇编器输出列表	240
10.1.1	汇编列表文件标题.....	240
10.1.2	汇编列表	241
10.1.3	符号列表	243
10.1.4	交叉参考列表.....	244
10.1.5	错误列表	245

10.2	连接器输出列表	246
10.2.1	连接列表文件标题	246
10.2.2	映像列表	247
10.2.3	公共符号列表	249
10.2.4	局部符号列表	250
10.2.5	错误列表	250
10.3	目标转换器输出列表	251
10.3.1	错误列表	251
10.4	库管理程序输出列表	252
10.4.1	库数据输出列表	252
10.5	列表转换器输出列表	253
10.5.1	绝对汇编列表	253
10.5.2	错误列表	253
第 11 章	高效使用 RA78K0S	254
11.1	提高操作效率 (EXIT 状态功能)	255
11.2	准备软件开发环境 (环境变量)	256
11.3	中断程序执行	256
11.4	使汇编列表易于读取	257
11.5	降低程序的启动时间	258
11.5.1	在源程序中指定控制指令	258
11.5.2	使用 PM+	258
11.5.3	创建参数文件和子命令文件	259
11.6	目标模块库生成	260
第 11 章	高效使用 RA78K0S	254
11.1	提高操作效率 (EXIT 状态功能)	255
11.2	准备软件开发环境 (环境变量)	256
11.3	中断程序执行	256
11.4	使汇编列表易于读取	257
11.5	降低程序的启动时间	258
11.5.1	在源程序中指定控制指令	258
11.5.2	使用 PM+	258
11.5.3	创建参数文件和子命令文件	259
11.6	目标模块库生成	260
第 12 章	错误消息	261
12.1	错误消息概述	261
12.2	结构化汇编器错误消息	262
12.3	汇编器错误消息	267
12.4	连接器错误信息。	275
12.5	目标转换程序错误消息	280
12.6	库管理程序错误消息	282
12.7	表转换程序错误消息	285
12.8	PM + 错误消息	287
附录 A	示例程序	289
A.1	K0smain.asm	289

A.2	K0sub.asm.....	290
A.3	test1.s.....	291
A.4	test2.s.....	292
A.5	testinc.s.....	293
A.6	st.bat.....	294
附录 B	使用注意事项列表.....	295
附录 C	选项列表.....	297
C.1	结构化汇编器选项列表.....	297
C.2	汇编器选项列表.....	299
C.3	连接器选项列表.....	301
C.4	目标转换器选项列表.....	303
C.5	库管理程序选项列表.....	304
C.6	表转换程序选项列表.....	305
附录 D	子命令列表.....	306
附录 E	索引.....	307

第 1 章 概述

本章介绍 RA78K0S 在微处理器软件开发中的作用及其功能。

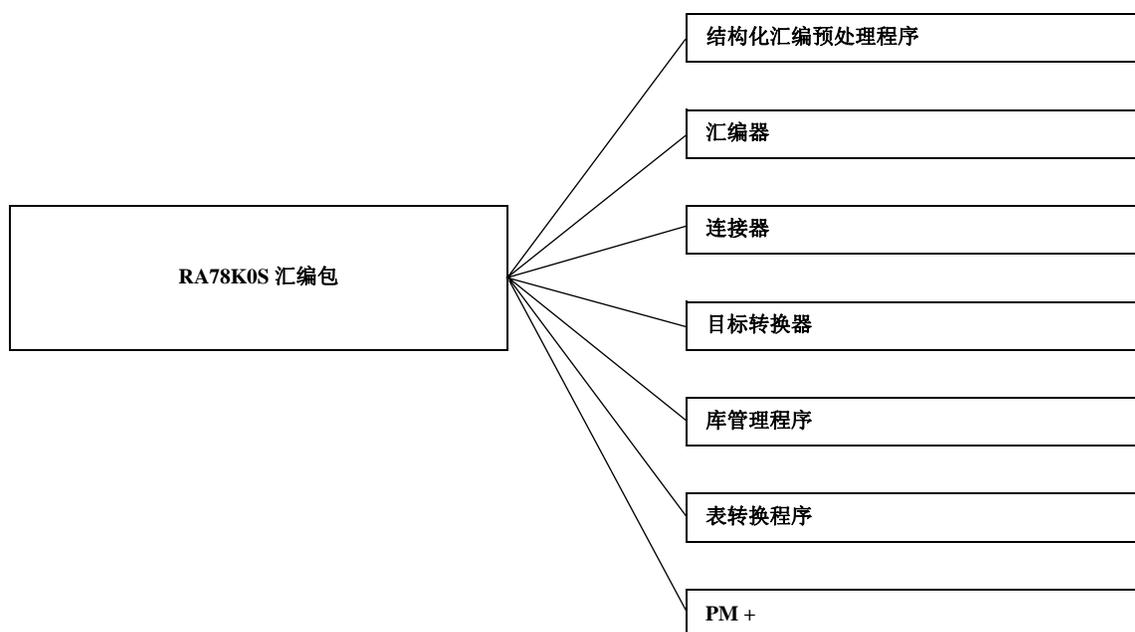
1.1 汇编器概述

RA78K0S 汇编包是对一系列程序的统称，这些程序用于将用汇编语言为 78K0S 系列微处理器编写的源程序翻译成机器语言代码。

RA78K0S 包含六个程序：结构化汇编预处理程序、汇编器、连接器、目标转换器、库文件管理程序及表转换程序。

此外，RA78K0S 还配备有 PM+，以便于用户方便地在 Windows™ 上执行一系列操作，包括编辑、编译/汇编、连接和调试等。

图 1-1 RA78K0S 汇编包



1.1.1 什么是汇编器？

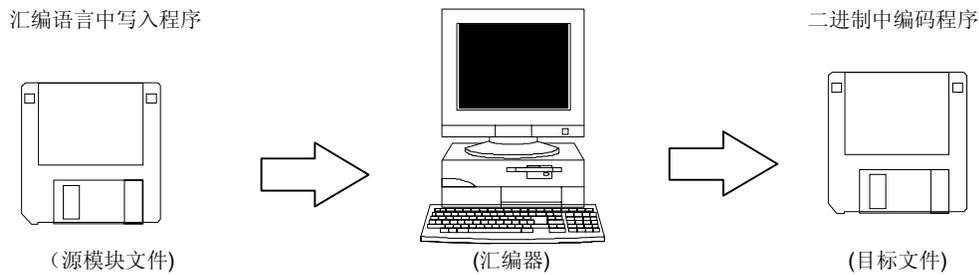
(1) 汇编语言与机器语言

汇编语言是处理器最基本的编程语言。

要使微控制器完成工作，程序和数据是必不可少的。程序和数据必须由人（程序员）来编写，并存储在微控制器的存储区。微处理器处理的程序和数据是二进制数的集合，称为机器语言。不过，对于程序员来说，机器语言代码难以记忆，容易出现错误。幸运的是，可以使用英语缩写（助记符）以人们易于理解的方式来表示原始机器语言代码的含义。使用这种符号编码的编程语言系统称为汇编语言。

因为微处理器必须以机器语言的形式处理程序，所以需要另一种程序来将用汇编语言编写的程序翻译成机器语言。这种程序称为汇编器。

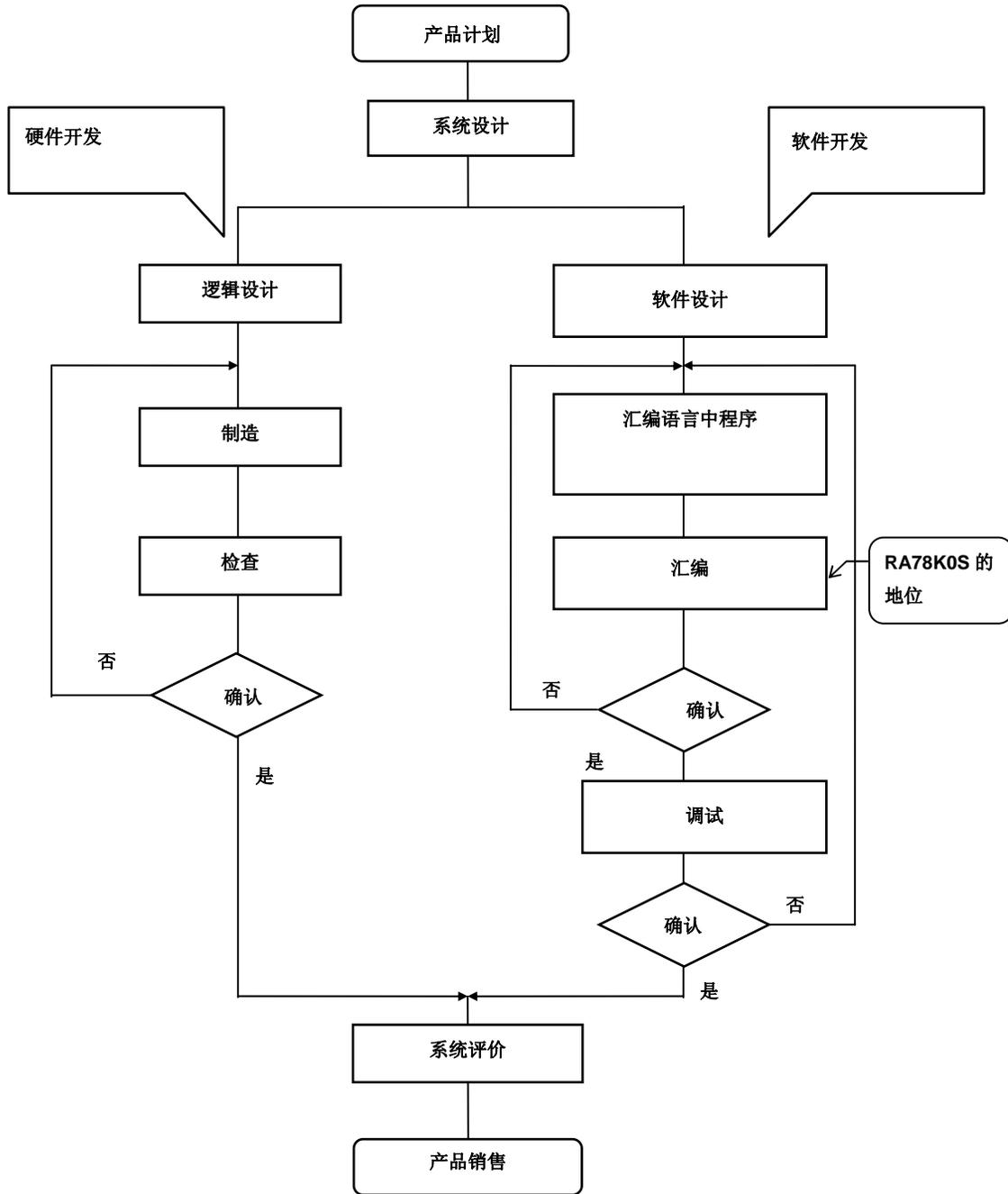
图 1-2 汇编流程



(2) 微处理器相关产品开发与 RA78K0S 的作用

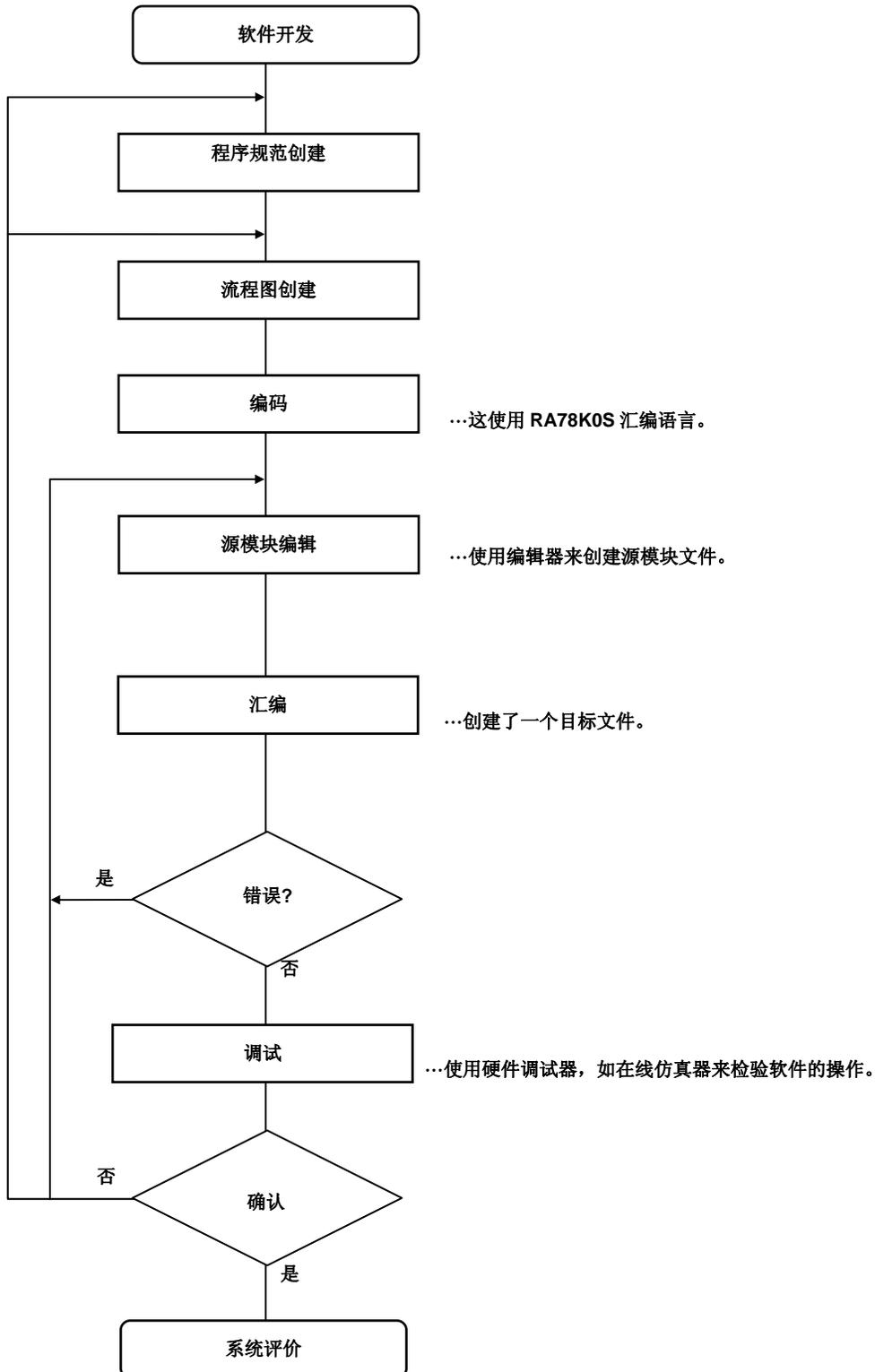
图 1-3 应用微处理器产品开发流程说明了汇编语言编程在（软件）产品开发流程中的地位。

图 1-3 应用微处理器产品开发流程



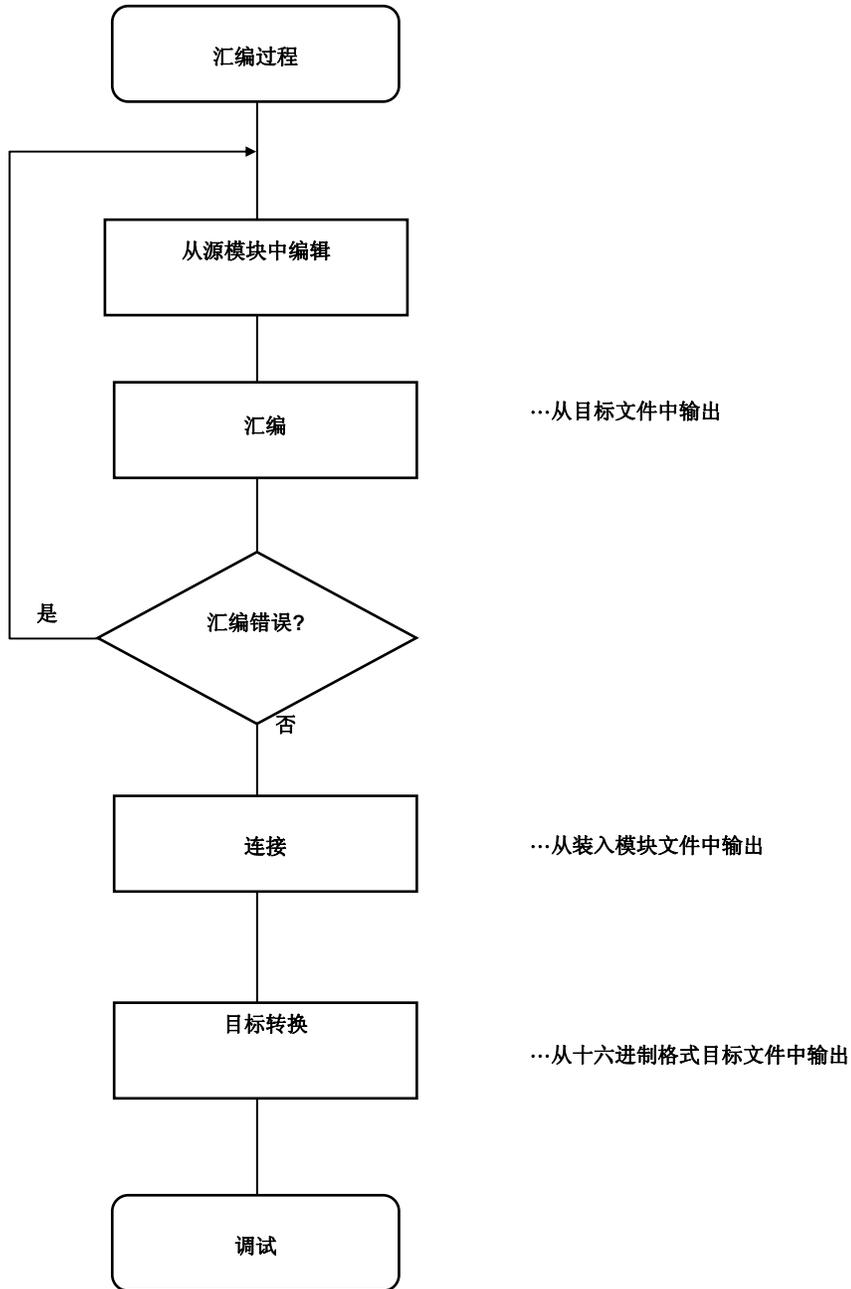
关于软件开发流程更详细的说明，请参见图 1-4 软件开发流程。

图 1-4 软件开发流程



因此，RA78K0S 适用于汇编过程。

图 1-5 RA78K0S 汇编流程



1.1.2 什么是浮动汇编器？

使用前，用汇编器从源语言翻译过来的机器语言存储在微处理器的存储器中。为此，用于存储每条机器语言指令的内存单元必须已经确定。所以，汇编器汇编的机器语言时必须添加一些信息，说明每条机器语言指令存储在内存中的什么位置。

根据定位机器语言指令地址的方法，汇编器可以大致分为绝对汇编器和浮动汇编器。

- **绝对汇编器**

绝对汇编器将从汇编语言汇编的机器语言指令定位到绝对地址。

- **浮动汇编器**

在浮动汇编器中，将从汇编语言汇编的机器语言指令定位到暂时地址。绝对地址由称为连接器的程序来确定。

在过去，当使用绝对汇编器创建一个程序时，通常，程序员必须同时完成编程。不过，如果同时创建一个大型程序的各个部分，程序将变得复杂化，使得程序的分析和维护变得困难。为避免这一问题，对于每个功能单元，这样的大型程序在开发时要分成称为模块的多个子程序。这种编程方法称为模块化编程。

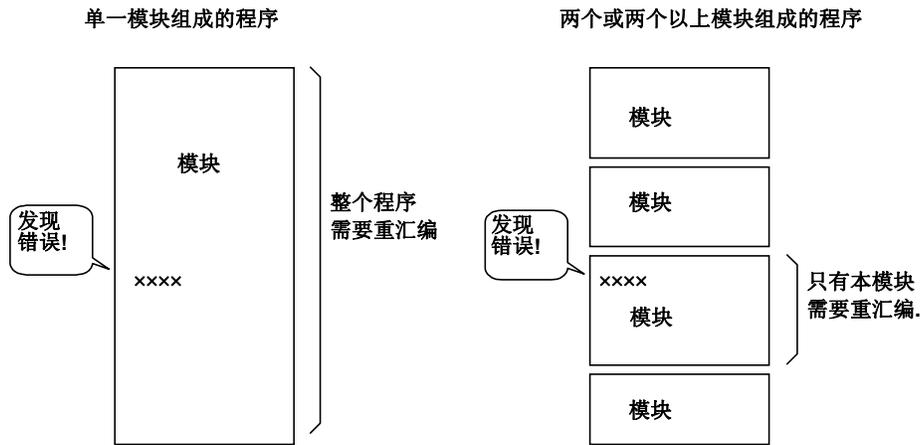
浮动汇编器是一种适于模块化编程的汇编器。使用浮动汇编器进行模块化编程具有以下优点：

- (1) **提高开发效率**

同时编写一个大型的程序是一件困难的事情。在此情况下，将程序分成各个功能模块可以使两个或多个程序员同时开发子程序以提高开发效率。

而且，如果在程序中发现了错误，不必要汇编整个程序来纠正程序的一部分，而只需重新汇编一个必须纠正的模块即可。这样缩短了调试时间。

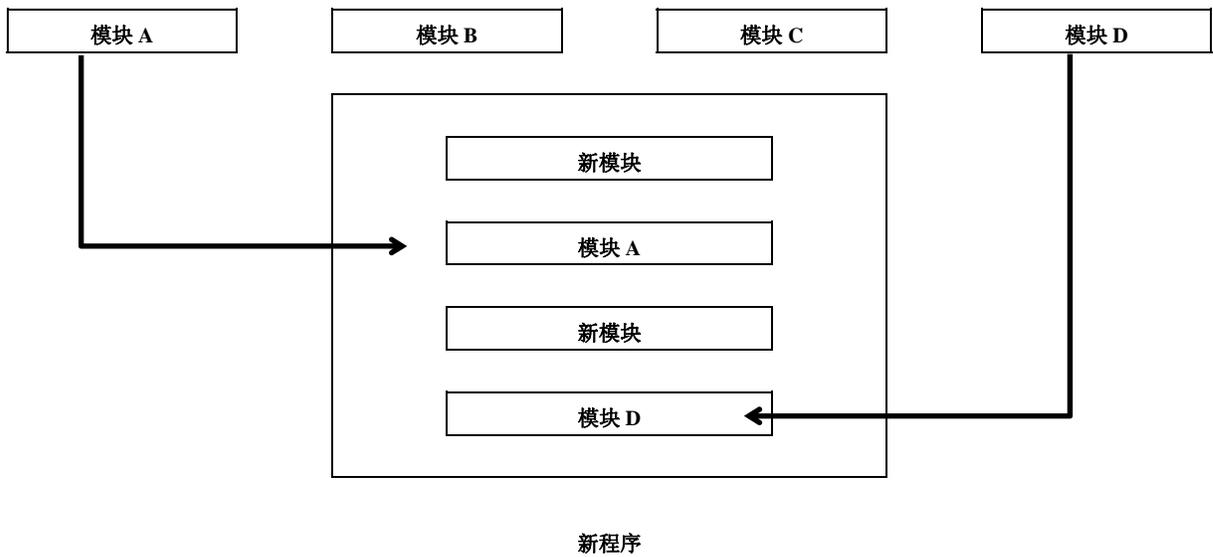
图 1-6 重汇编以便于调试



(2) 使用资源

以前创建的高度可靠、高度通用的模块可用于创建另一个程序。如果您积累了这样高度通用的模块资源，您可以在开发新程序时节省时间和劳动。

图 1-7 使用现有模块的程序开发

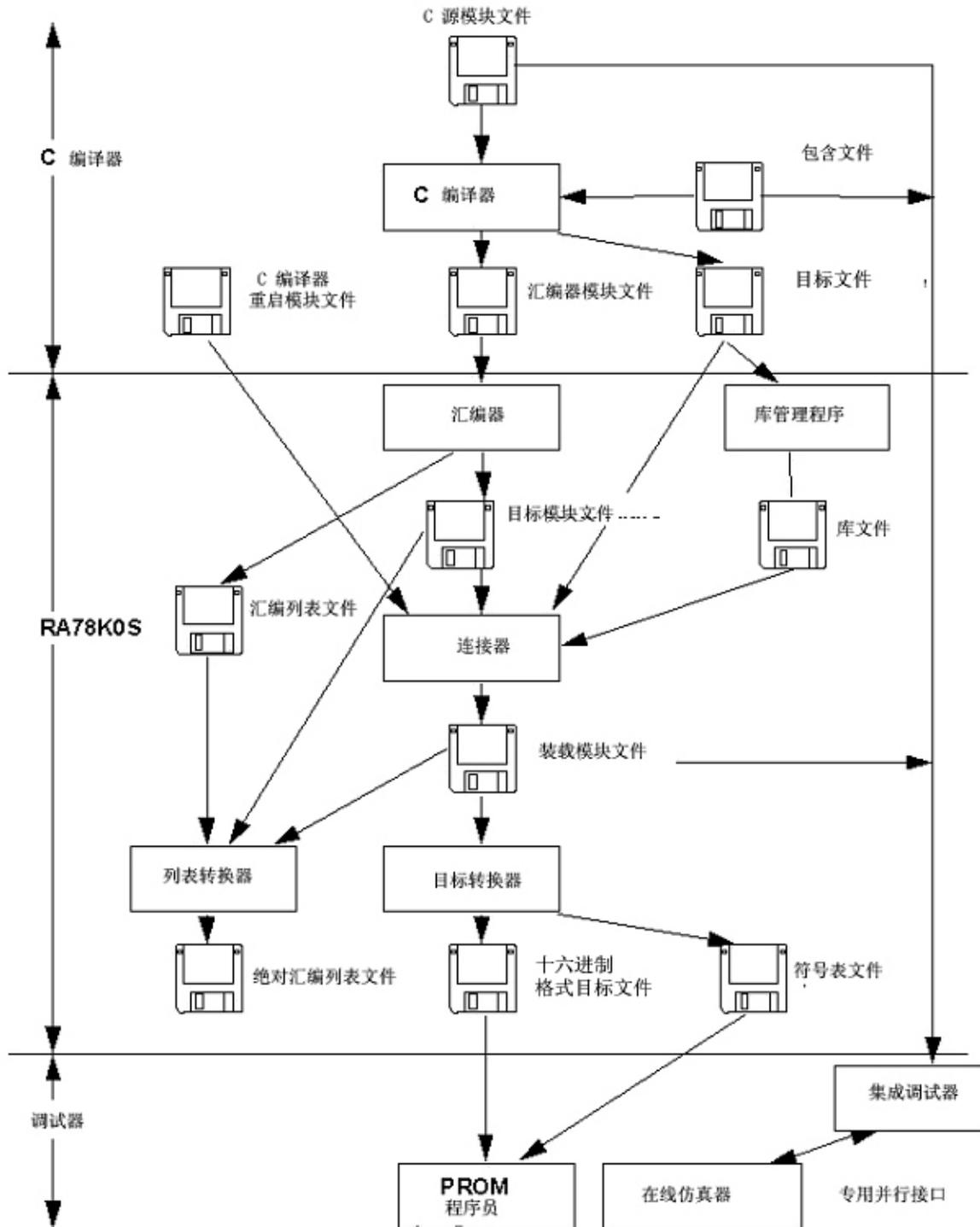


1.2 RA78K0S 功能概述

有关开发通用程序的步骤，请参见图 1-8 使用 RA78K0S 开发程序的步骤。程序开发基本上是从遵守汇编器到连接器再到目标转换器的流程。

汇编器、连接器、目标转换器及其他程序统称为“RA78K0S”。

图 1-8 使用 RA78K0S 开发程序的步骤



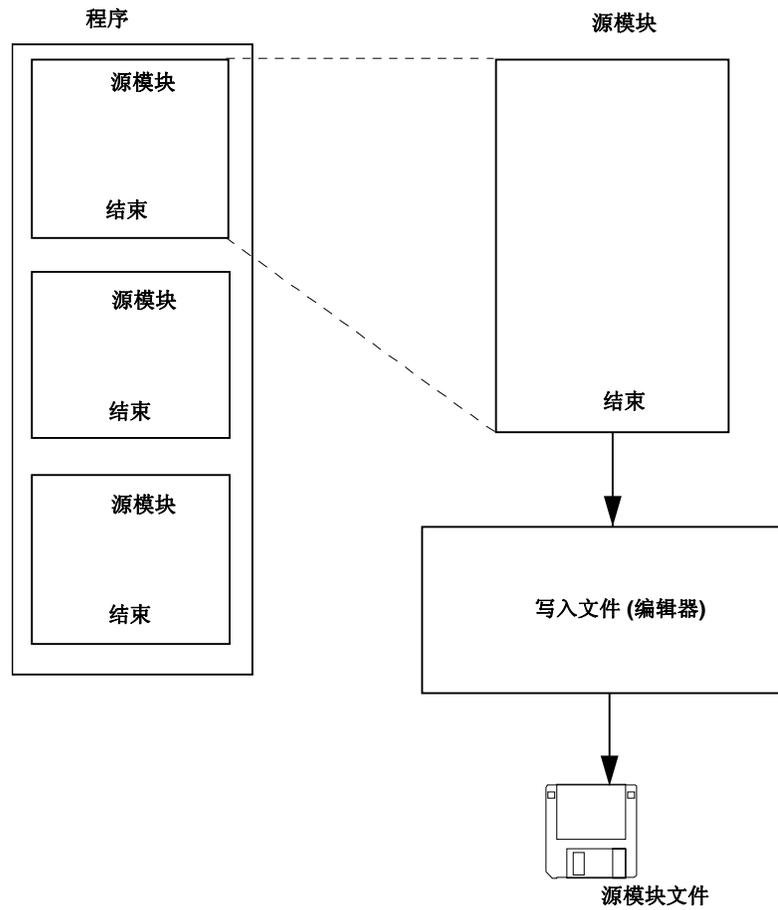
1.2.1 使用编辑器创建一个源模块文件

一个程序可以根据功能分为两个或多个模块。单个模块可以用作一个编码单元或汇编器输入单元。

一个汇编器输入单元的模块称为一个源模块。在各个源模块编码完成后，使用编辑器将源模块写入一个文件中。以这种方式创建的文件称为源模块文件。

源模块文件用作一个汇编器输入文件。

图 1-9 创建源模块文件

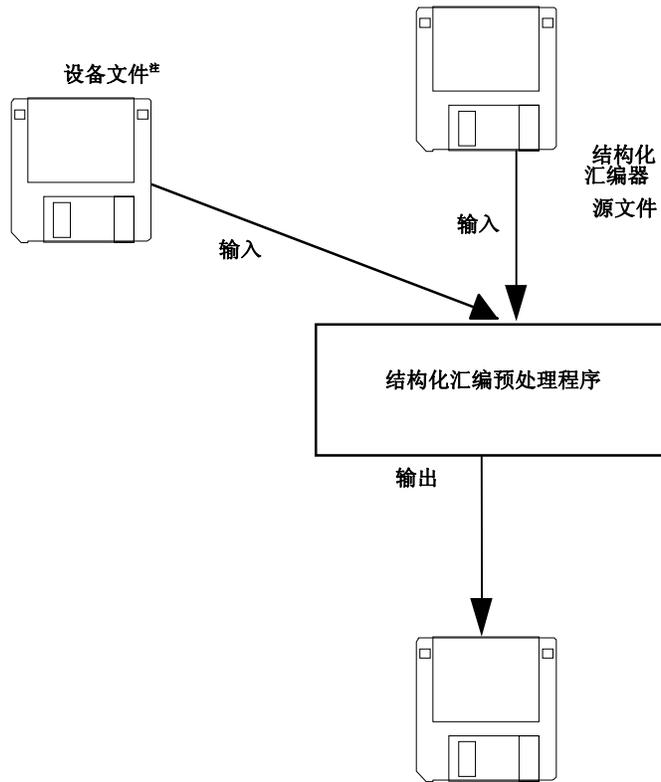


1.2.2 结构化汇编预处理程序

结构化汇编预处理程序是一个程序，其目的是使用汇编语言指令来进行结构化编程。结构化汇编预处理程序输入用结构化汇编语言编写的源程序以便向汇编器输出源程序。

有关结构化汇编预处理程序及结构化汇编语言的情况，请参阅文档 **RA78K0S 结构化汇编语言用户手册 (U11623E)**。

图 1-10 结构化汇编预处理程序功能

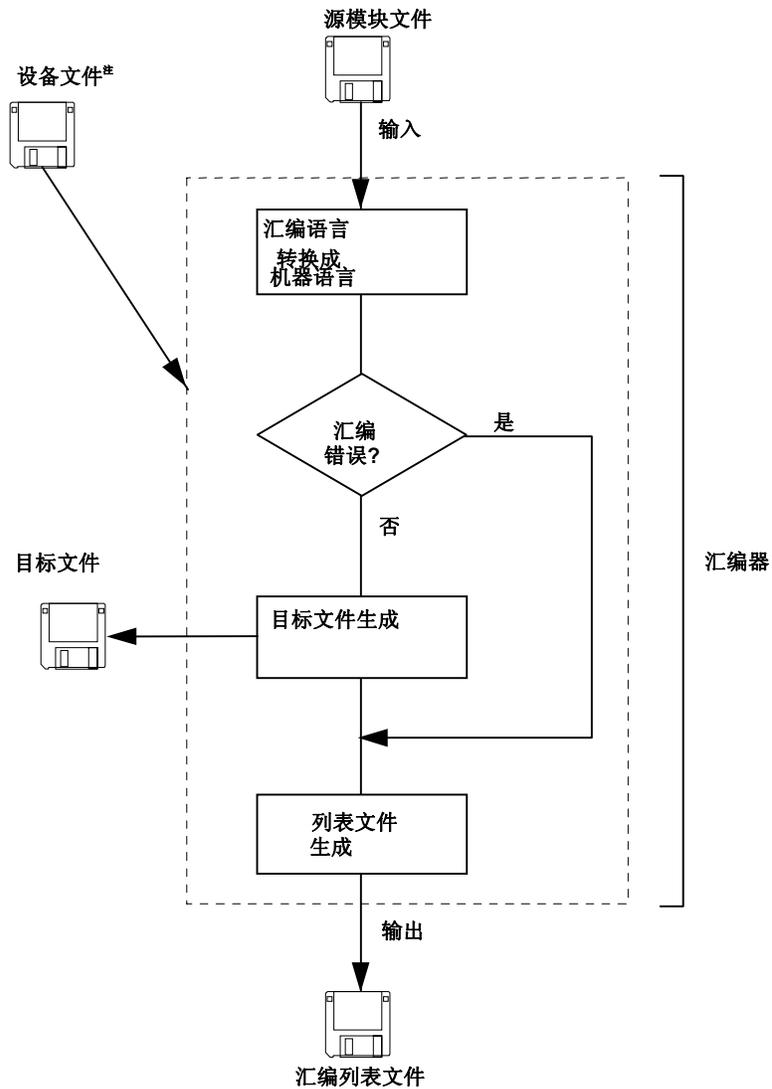


注 单独获得该设备文件。

1.2.3 汇编器

汇编器是这样一种程序，它输入源模块文件，并将汇编语言转换为二进制指令（机器语言）。如果汇编器在源模块中发现描述错误，它将输出一个汇编错误。如果未发现汇编错误，汇编器将输出一个目标文件，以指定位置数据，如机器语言数据及机器语言应存在内存中的什么位置。汇编数据作为一个汇编列表文件输出。

图 1-11 汇编器功能



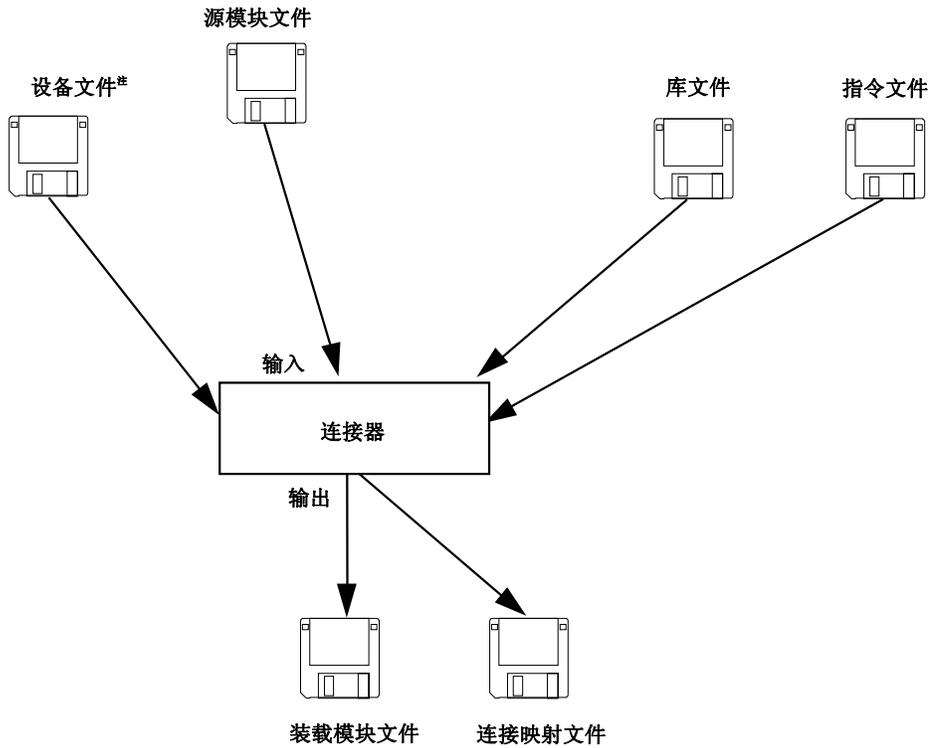
注 单独获得该设备文件。

1.2.4 连接器

连接器输入由编译器和汇编器输出的多个目标文件，并连接它们以便输出一个装入模块文件（即使只输入了一个目标文件，也必须执行连接）。

连接器决定输入模块中浮动段的单元地址。这样就确定了浮动符号及外部引用符号的值，以便正确的值被嵌入到装入模块文件中。

图 1-12 连接器功能

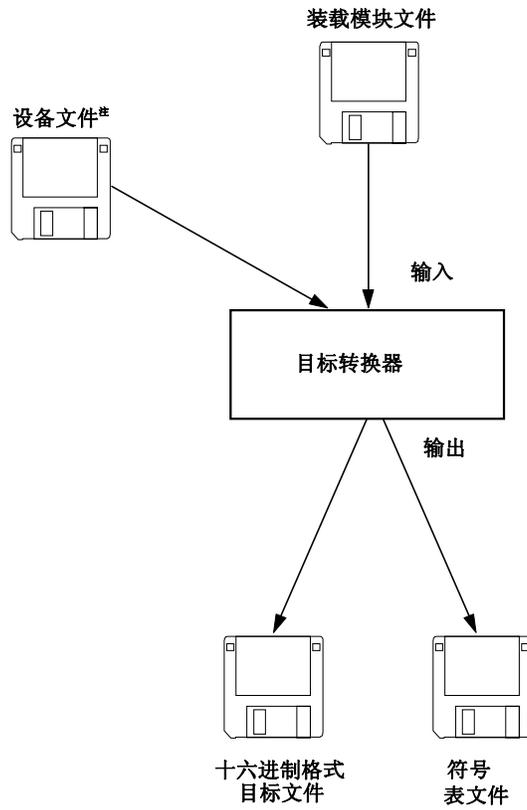


注 单独获得该设备文件。

1.2.5 目标转换器

目标转换器输入连接器输出的装入模块文件，并转换文件格式。生成的文件将作为十六进制格式的目标文件输出。目标转换器还以符号表文件输出符号调试所需的符号数据。

图 1-13 目标转换器功能



注 单独获得该设备文件。

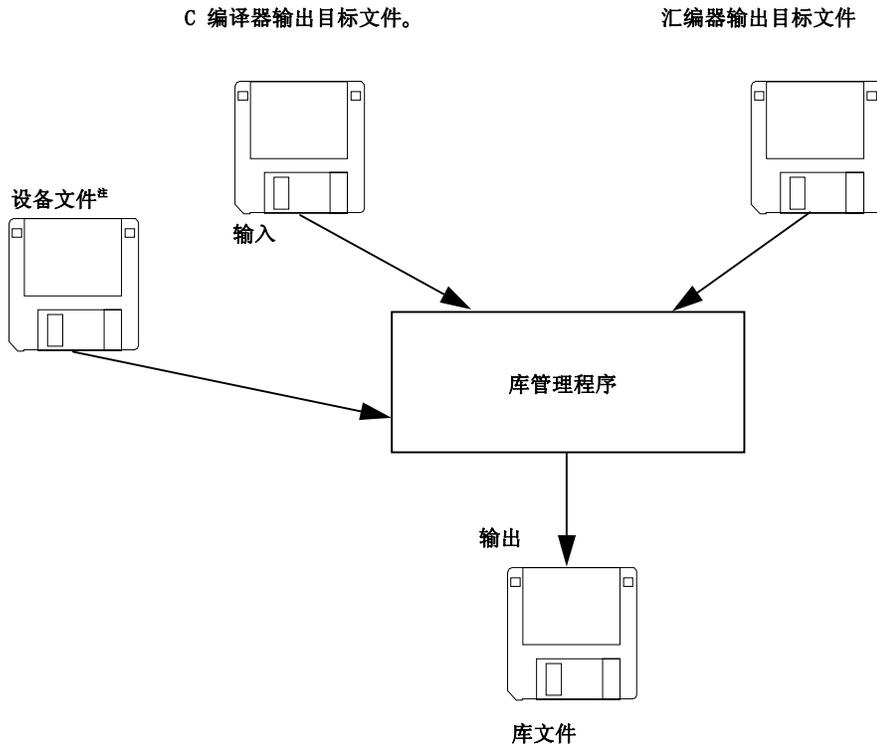
1.2.6 库管理程序

为了方便和易用，可以在库中存储一个具有清晰接口的通用模块。通过创建一个库，可以在单个文件中存储多个目标模块，使它们便于处理。

连接器内含这样的功能，即从库文件中只检索需要的模块。当多个模块被存储到一个库文件中时，可以无需指定各个模块文件名就能连接模块文件。

库管理程序就是用于创建和更新库文件的程序。

图 1-14 库管理程序功能



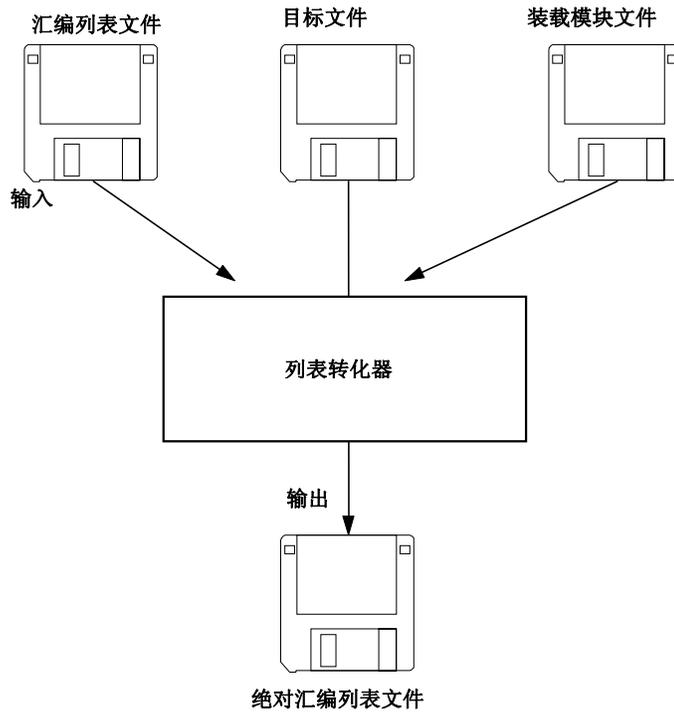
注 单独获得该设备文件。

1.2.7 表转换程序

表转换程序输入汇编器输出的目标文件和汇编列表文件及连接器输出的装入模块文件，并输出一个绝对汇编列表文件。

浮动汇编列表文件具有这样的优势：表中的地址和浮动值可以与它们的实际值不同。绝对汇编列表文件确定这些值，从而使调试和程序维护更为容易。

图 1-15 表转换程序功能



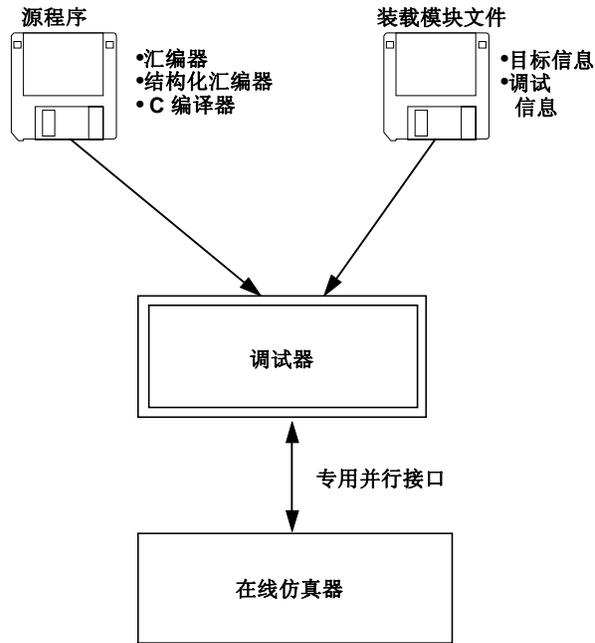
1.2.8 调试器

78K0S 系列的调试器是一个软件工具，它可以将来自源程序、寄存器和内存的数据显示在各自的窗口中，并执行调试。

调试器将连接器输出的装入模块文件下载到目标系统的在线仿真器（IE）。它还可以读出源程序文件，从而在源程序级执行调试。

调试器与 IE 是不同的程序包，与 RA78K0S 分开出售。

图 1-16 调试程序功能



1.3 程序开发前的提示

在开发程序前，请注意以下几点。

1.3.1 RA78K0S 的最高性能指标

(1) 汇编器的最高性能

表 1-1 汇编器的最高性能

项目	最高性能	
	PC 版本	EWS 版本
符号数（本地 + 公用）	65,535 ^{注1}	65,535 ^{注1}
可以输出到交叉引用表中的符号数	65,534 ^{注2}	65,534 ^{注2}
一个宏可以引用的最大宏指令体大小	1 MB	1 MB
所有宏指令体的总大小	10 MB	10 MB
一个文件中的段数	100	100
宏及一个文件中的包含规格	10,000	10,000
宏及一个包含文件中的包含规格	10,000	10,000
浮动信息 ^{注3}	65,535	65,535
行数信息	65,535	65,535
一个文件中的 BR/CALL 指令	32,767	32,767
一个源代码行中的字符数	2,048 ^{注4}	2,048 ^{注4}
符号长度	256 字符	256 字符
分支名定义数	100	100
分支名的字符长度	31	31
一个文件中的包含 文件嵌套层数	8 层	8 层

- 注
1. 如果使用了 2001 个以上的公共符号，则速度会下降，因为访问了临时文件。
 2. 不包括模块名和节名的数量。
 3. 当符号值不能被汇编器解析时，传递给连接器的信息。
例如，如果一个外部引用的符号要被 MOV 指令引用，将在 .rel 文件中产生两条浮动信息。
 4. 包括 CR 和 LF 代码。 如果在一行中写了 2048 个以上的字符，则将输出一个错误消息，同时处理结束。

(2) 连接器的最高性能

表 1-2 连接器的最高性能

项目	最高性能	
	PC 版本	EWS 版本
符号数（本地 + 公用）	65,535	65,535
相同段的行数信息	65,535	65,535
段数	65,535	65,535
输入模块	1,024	1,024

1.4 RA78K0S 的功能

RA78K0S 具有以下功能:

(1) 宏功能

当同一组指令必须在源程序中多次重复地描述时, 可以给这组指令一个宏名来定义一个宏。通过使用宏功能, 可以提高编码效率和程序的可读性。

(2) 分支指令的优化功能

RA78K0S 有一个自动选择分支指令的指令 (即 BR 指令、CALL 指令)。

要创建一个具有高内存效率的程序, 必须根据分支指令的分支目的地范围描述一个字节分支指令。不过, 要让程序员注意每次分支的分支目的地范围来描述一个分支指令是令人棘手的。通过描述 BR 指令或 CALL 指令, 汇编器会根据分子目的地范围产生适当的分支指令。这称为分支指令的优化功能。

(3) 条件汇编功能

使用该功能, 可以根据预定的条件, 指令一部分源程序进行汇编或不汇编。如果在源程序中描述了一个调试语句, 则可以通过设置一个进行条件汇编的分支, 来选择调试语句是否应翻译成机器语言。当调试语句不再需要时, 可以不对程序进行大的修改来汇编源程序。

第 2 章 产品概况与安装

本章介绍将存储在提供的 RA78K0S 介质中的文件安装在用户开发环境（主机）中的步骤，以及将这些文件从用户开发环境中卸载的步骤。

2.1 主机与提供介质

汇编包支持如表 2-1 所示的开发环境。提供介质因主机而异。

表 2-1 提供介质与汇编包的记录格式

主机	对应的操作系统（版本）	提供介质	记录格式
PC-9800 系列	日语 Windows (98/Me/2000/XP/NT™4.0) 注	CD-ROM	支持 Windows 标准安装程序
IBM PC/AT 兼容机	日语 Windows (98/Me/2000/XP/NT 4.0) 注 英语 Windows (98/Me/2000/XP/NT 4.0) 注		
HP9000 系列 700	HP-UX™ (Rel. 10.10 或更高)	CD-ROM	cp 命令
SPARCstation 系列	SunOS™ (Rel. 4.1.4 或更高) Solaris™ (Rel. 2.5.1 或更高)		

注 要在 Windows 中使用汇编器，必须有 PM+。

如果未使用 PM+，可以从 DOS 提示符（Windows 98/Me）或命令提示符（Windows 2000/XP/NT 4.0）下使用汇编包中包含的各个工具。

2.2 安装

2.2.1 Windows 版本的安装

下面介绍将 RA78K0S 提供介质中提供的文件安装到主机中的步骤。

<1> 启动 Windows

启动主机和外围设备电源，并启动 Windows。

<2> 设置提供介质

在主机的适当驱动器（光盘驱动器）中设置 RA78K0S 提供介质。安装程序会自动开始。按照显示器屏幕上显示的消息执行安装。

注意事项 如果安装程序未自动开始，则执行 RA78K0S\DISK1 文件夹中的 SETUP.EXE。

<3> 文件确认

使用 Windows 资源管理器等，检查 RA78K0S 提供介质中包含的文件是否安装到主机中。

关于每个文件夹的详细情况，请参阅 2.4 目录配置。

2.2.2 UNIX 版本的安装

使用以下步骤安装 UNIX 版本。这里假定安装到 /necools/bin 路径中。

<1> 进入系统

进入主机中。

<2> 目录选择。

转到安装目录。

```
%cd /necools/bin
```

<3> 设置提供介质

在光驱中设置光盘。

<4> 复制文件。

执行 cp 命令从光盘中复制文件（确认在光驱中设置了光盘后复制文件）。

<5> 设置环境变量 PATH

将 /necools/bin 添加至环境变量 PATH。

2.3 安装设备文件

2.3.1 Windows 版本的安装

使用设备文件安装程序安装设备文件。设备文件安装程序与 RA78K0S 同时安装。

2.3.2 UNIX 版本的安装

可以使用 `-y` 选项指定设备文件的目录，或者指定目录（例如：`-y/nectools/dev`），然后将设备文件复制到具有汇编执行格式的目录中（例如：`/nectools/bin`）。

2.3.3 设备文件的注册表注册

如果设备文件已经安装，在 RA78K0S 安装过程中，会显示提示您执行设备文件的注册表注册的消息。

如果当前使用的是 32 位环境，则将用于 RA78K0S 的设备文件（Ver. 1.30 或更高，16 位环境）注册到注册表中（32 位环境）。

还可以在 RA78K0S 安装完毕后，使用设备文件安装程序来完成注册表注册。

注册表注册步骤如下：

<1> 启动设备文件安装程序

<2> 源文件选择。

单击 [**B**rowse...] 按钮，然后选择 16 位环境中使用的“NECDEV.INI”。

从源文件列表框中显示的设备文件中选择注册到注册表中的文件。

<3> 移动

单击 [**M**ove] 按钮，将文件注册到注册表（32 位环境）。

2.4 目录配置

2.4.1 Windows 版本目录配置

安装时显示的标准目录是 Windows 所在驱动器上的“\NECTools32”。安装目录的配置如下图所示。安装时可以更改驱动与安装目录。在同一目录中安装 PM+ 及 RA78K0S。

图 2-1 目录配置

— NECTools32\	
bin\	
ra78k0s.exe	汇编器执行格式
st78k0s.exe	结构化汇编预处理程序执行格式
lk78k0s.exe	连接器
oc78k0s.exe	目标转换器
lc78k0s.exe	表转换程序
lb78k0s.exe	库管理程序
lb78k0se.exe	库与 PM+ 环境的动态链接库之间的接口工具
lb78k0sp.exe	独立启动库
ra78k0s.is	汇编器使用的文件
*78k0sp.dll	PM+ 的 DLL 工具
*78k0s.hlp	开始命令行的帮助文件
doc\	用户手册与辅助说明
hlp\	联机手册
setup\	安装与卸载的数据文件
smp78k0s\ra78k0s\	
k0smain.asm	汇编示例程序
k0ssub.asm	汇编示例程序
ra.bat	汇编示例程序的批处理文件
readme.doc	示例程序与批处理程序（文本文件）的说明
test1.s	结构化汇编的示例程序
test2.s	结构化汇编的示例程序
testinc.s	结构化汇编的示例程序
st.bat	结构化汇编示例程序的批处理文件

备注 根据安装程序的默认指示，本手册中的说明假设安装至具有默认程序文件夹名“NECTools32”的标准目录中。

2.4.2 UNIX 版本目录配置

安装后的文件配置如下。下面假设安装在 /necools/bin 中。

—necools/	
bin/	
ra78k0s	汇编器的可执行格式
st78k0s	结构化汇编预处理程序的可执行格式
lk78k0s	连接器的可执行格式
oc78k0s	目标转换器的可执行格式
lc78k0s	表转换程序的可执行格式
lb78k0s	库管理程序的可执行格式
*.hlp	各个程序相应的帮助文件（文本文件）
ra78k0s.is*	定义汇编器使用的指令集的表文件
.asm,.s	用于安装配置的示例程序
*.sh	用于安装配置的批处理文件
*.sh	用于安装配置的批处理文件
readme.doc	安装配置外壳程序文件（文本文件）的说明

建议将 C 编译器、集成调试器、系统仿真器及设备文件安装在与安装汇编包相同的目录中。

2.5 卸载步骤

2.5.1 Windows 版本的卸载

下面介绍卸载安装到主机中的文件的步骤。

- (1) Windows 启动
启动主机和外围设备电源，并启动 Windows。
- (2) 打开“控制面板”窗口
按[开始]按钮，然后选择 [设置]-[控制面板]，打开<控制面板>窗口。
- (3) 打开 <添加或删除程序>窗口
双击<控制面板>窗口中的[添加或删除程序]图标，打开 <添加或删除程序>窗口。
- (4) 删除 RA78K0S
在选中<添加或删除程序>窗口中<<添加或删除>> 标签中显示的 安装软件列表中的“NEC RA78K0S 78K/0 Assembler Vx.xx”后，单击[添加/删除...]按钮。
当<系统设置更改>窗口打开后，单击[是]按钮。
- (5) 文件确认
使用 Windows 资源管理器等，检查主机中安装的文件是否被卸载。关于每个文件夹的详细情况，请参阅 **2.4 目录配置**。

2.5.2 UNIX 版本的卸载

使用 rm 命令卸载 **2.2.2 UNIX 版本的安装**中复制的文件。

2.6 环境设置

2.6.1 环境变量

设置以下环境变量。如果是使用 Windows 安装程序安装的汇编包，则必要的环境变量将自动被设置。

PATH	指定可执行格式的汇编器的存储目录。
TMP	指定用于创建临时文件的目录（仅对 PC-9800 系列和 IBM PC/AT 兼容机有效）
INC78K0S	指定用于搜索包含文件的目录。
LIB78K0S	指定用于搜索库的目录，如果使用库的话。
LANG78K	指定在注释中描述的汉字代码（2 字节代码）。

[示例]

对于 PC-9800 系列或 IBM PC/AT 兼容机

```
PATH = %PATH%; C:\NECTools32\bin
set TMP = C:\tmp
set INC78K0S = C:\NECTools32\inc78k0s
set LIB78K0S = C:\NECTools32\lib78k0s
set LANG78K = SJIS
```

对于 HP9000 系列 700 或 SPARCstation 系列

- 使用 **cs**h 的示例

```
set path = ($path /ra78k0s)
setenv INC78K0S /ra78k0s
setenv LIB78K0S /ra78k0s
setenv LANG78K EUC
```

- 使用 **sh** 的示例

```
PATH = $PATH:/ra78k0s
INC78K0S = /ra78k0s
LIB78K0S = /ra78k0s
export PATH INC78K0S LIB78K0S
setenv LANG78K EUC
```

2.6.2 源文件中的汉字代码

- 可以在源文件中的特定地方（注释等）使用汉字（2 字节字符）。
- 使用环境变量（LANG78K）指定汉字编码类型，汉字编码控制指令（KANJI CODE）或汉字编码指定选项（-ZE/-ZS/-ZN）。

第 3 章 执行 RA78K0S

本章介绍使用汇编包 RA78K0S 的步骤，内容包括从汇编到目标程序转换。

根据本章介绍的执行方法对示例程序“k0smain.asm”及“k0ssub.asm”进行了汇编、连接和目标程序转换。在执行从结构化汇编程序到目标程序时，使用了示例程序“test1.s”及“test2.s”。

使用 PC-9800 系列或 IBM PC/AT 兼容机，介绍了如何在 PM + 与命令行中运行汇编包。

3.1 准备执行 RA78K0S

3.1.1 示例程序

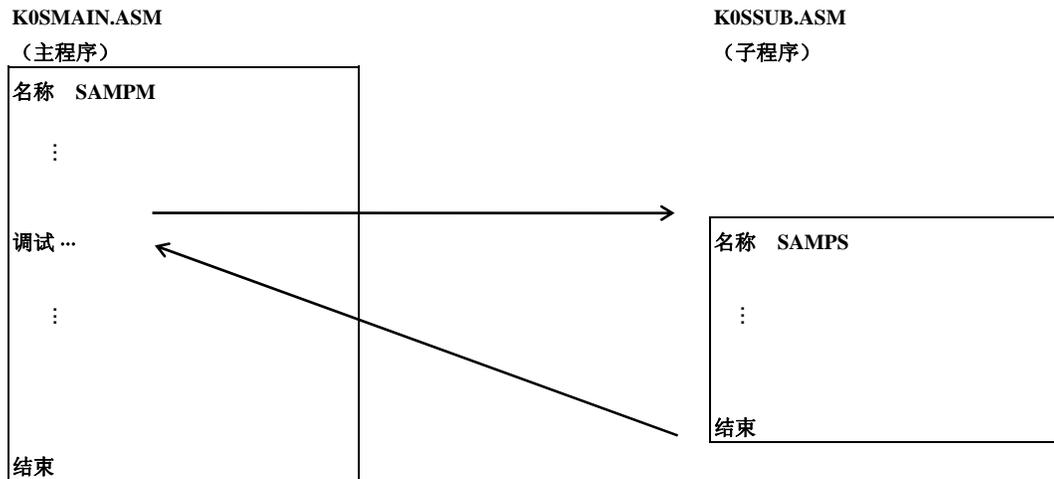
在系统盘中存储的文件有[K0SMMAIN.ASM]及[K0SSUB.ASM]。这些文件是示例程序，用于验证汇编包的运行。在稍后的汇编器操作中，这些文件将作为源程序文件输入到汇编器中。

下面是对示例程序内容的简单介绍。这些程序由转换成 ASCII 码的十六进制数据组成。程序由两个模块组成：主程序与子程序。

主程序模块的名称是 SAMPM，它存储在源程序模块文件中（K0SMMAIN.ASM）。

子程序模块的名称是 SAMPS，它存储在源程序模块文件（K0SSUB.ASM）中。

图 3-1 源程序的结构



■K0SMAN.ASM (主程序)

```

NAME  SAMPM
;*****
;
;   HEX -> ASCII Conversion Program
;
;   main-routine
;*****

PUBLIC MAIN, START
EXTRN  CONVAH
EXTRN  @_STBEG

DATA  DSEG  saddr
HDTSA: DS    1
STASC: DS    2

CODE  CSEG  AT 0H
MAIN: DW    START

CSEG
START:
;chip initialize
MOVW  AX, @_STBEG
MOVW  SP, AX

MOV   HDTSA, #1AH
MOVW  HL, #HDTSA           ;set hex 2-code data in HL register

CALL  !CONVAH              ;convert ASCII <- HEX
                                ;output BC-register <- ASCII code

MOVW  DE, #STASC          ;set DE <- store ASCII code table
MOV   A, B
MOV   [DE], A
INCW  DE
MOV   A, C
MOV   [DE], A

BR    $$

END

```

■K0SSUB.ASM (子程序)

```

NAME  SAMPS
;*****
;
;  HEX -> ASCII Conversion Program
;          sub-routine
;
;  input condition : (HL) <- hex 2 code
;  output condition : BC-register <-ASCII 2 code
;
;*****

PUBLIC  CONVAH

CSEG
CONVAH:
MOV    A, [HL]
ROR    A, 1
AND    A, #0FH          ;hex upper code load
CALL   !SASC
MOV    B, A            ;store result

XOR    A, A
XCH    A, [HL]
AND    A, #0FH          ;hex lower code load
CALL   !SASC
MOV    C, A            ;store result

RET

;*****
;  subroutine  convert ASCII code
;
;  input  Acc (lower 4bits) <- hex code
;  output Acc          <- ASCII code
;*****

SASC:  CMP    A, #0AH          ;check hex code > 9
BC     $SASC1
ADD    A, #07H          ;bias (+7H)
SASC1: ADD    A, #30H          ;bias (+30H)
RET

END

```

备注 示例程序是一个参考程序，用于向读者介绍 RA78K0S 的功能与操作。它不能用作一个应用程序。

3.1.2 示例程序的结构

下面介绍用作下述操作例子的示例程序。

K0smain.asm	主程序程序
K0ssub.asm	子程序模块
mylib.lib	库文件（此处未用到）
sample.dr	指示文件

3.2 RA78K0S 的执行步骤

系统盘中的批处理文件(ta.bat)用于 RA78K0S 的操作。

汇编器、连接器、目标转换器和表转换程序使用“k0smain.asm”和“k0ssub.asm”以上述顺序进行执行，它们是用汇编语言在 ra.bat 中编写的源文件。发生错误时，将输出一条消息，批处理文件终止。

用作目标设备的设备型号规格将被输入到该批处理文件中（分别获得设备文件）。

下面的解释将 μ PD789024 作为目标设备。

■ ra.bat（用于验证 RA78K0S 操作的批处理程序）

```

echo off
cls
set      LEVEL=0

if "%1" == "" goto ERR_BAT

ra78k0s -C%1 k0smain.asm
if errorlevel 1 set LEVEL=1
ra78k0s -C%1 k0ssub.asm
if errorlevel 1 set LEVEL=1
if %LEVEL% == 1 echo Assemble error !!
if %LEVEL% == 1 goto END

cls
lk78k0s k0smain.rel k0ssub.rel -s -orasample.lmf -prasample.map
if errorlevel 1 echo Link error !!
if errorlevel 1 goto END

cls
oc78k0s rasample
if errorlevel 1 echo Object conversion error !!
if errorlevel 1 goto END

cls
set LEVEL=0
lc78k0s -ltest.lmf -rk0smain.rel k0smain.prn
if errorlevel 1 set LEVEL=1
lc78k0s -lrasample.lmf -k0ssub.rel k0ssub.pn
if errorlevel 1 set LEVEL=1
if %LEVEL% == 1 echo List conversion error !!

```

```
if %LEVEL% == 1 goto END
```

```
cls  
echo No error.  
goto END
```

```
:ERR_BAT
```

```
echo Usage : st.bat chiptype
```

```
:END
```

```
echo on
```

(1) 执行批处理文件。

指定目标设备型号，执行 RA78K0S 操作验证批处理程序。

```
A>ra.bat 9024
```

显示屏上显示以下消息。

```
78K/0S Series Assembler Vx.xx [xx xxx xxxx]  
Copyright (C) NEC Electronics Corporation xxxx,xxxx
```

```
PASS_PARSE Start  
PASS_OUTOBJ Start
```

```
Target chip : uPD789024  
Device file : Vx.xx
```

```
Assembly complete, 0 error(s) and 0 warning(s) found.
```

```
78K/0S Series Assembler Vx.xx [xx xxx xxxx]  
Copyright (C) NEC Electronics Corporation xxxx,xxxx
```

```
PASS/PARSE Start  
PASS/OUTOBJ Start
```

```
Target chip : uPD789024  
Device file : Vx.xx
```

```
Assembly complete, 0 error(s) and 0 warning(s) found.
```

清除屏幕。

```
78K/0S Series Linker Vx.xx [xx xxx xxxx]  
Copyright (C) NEC Electronics Corporation xxxx,xxxx  
Target chip : uPD789024  
Device file : Vx.xx
```

Link complete, 0 error(s) and 0 warning(s) found.

清除屏幕。

78K/0S Series Object Converter Vx.xx [xx xxx xxxx]
Copyright (C) NEC Electronics Corporation xxxx,xxxx

Target chip : uPD789024
Device file : Vx.xx

Object Conversion Complete, 0 error(s) and 0 warning(s) found.

清除屏幕。

List Conversion Program for RA78K/0S Vx.xx [xx xxx xxxx]
Copyright (C) NEC Electronics Corporation xxxx,xxxx

Pass1: start...
Pass2: start...
Conversion complete.

List Conversion Program for RA78K/0S x.xx [xx xxx xxxx]
Copyright (C) NEC Electronics Corporation xxxx,xxxx

Pass1: start...
Pass2: start...
Conversion complete.

清除屏幕。

No error.

(2) 检查 C 驱动器的内容。

输出了以下文件。

k0smain.rel:	目标文件
k0smain.prm:	汇编表文件
k0ssub.rel:	目标文件
k0ssub.prn:	汇编表文件
rasample.lmf:	装入模块文件
rasample.map:	连接表文件
rasample.hex:	十六进制格式目标文件
rasample.sym:	符号表文件
k0smain.p:	绝对汇编表文件
k0ssub.p:	绝对汇编表文件

(3) RA78K0S 执行过程概要

下面是对 RA78K0S 执行过程的概略性介绍。

图 3-2 RA78K0S 执行过程 1

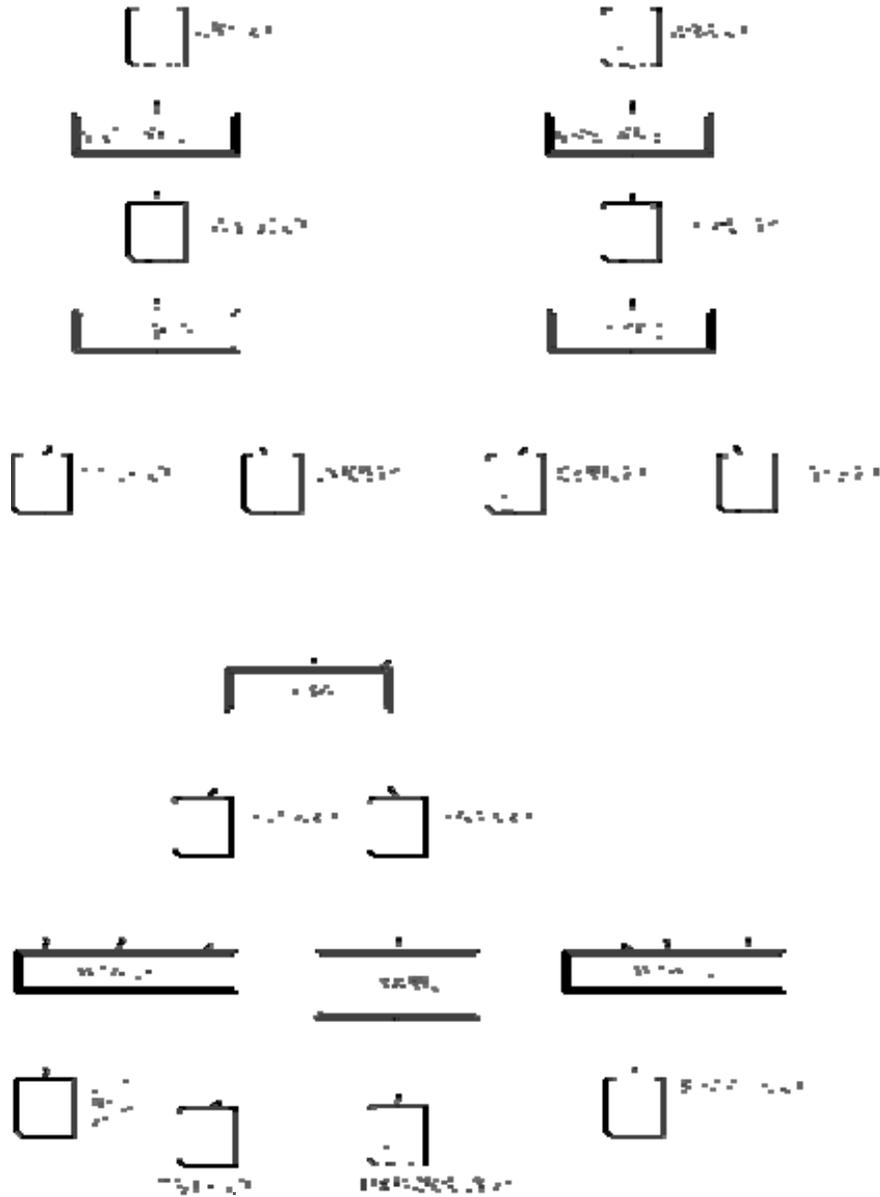
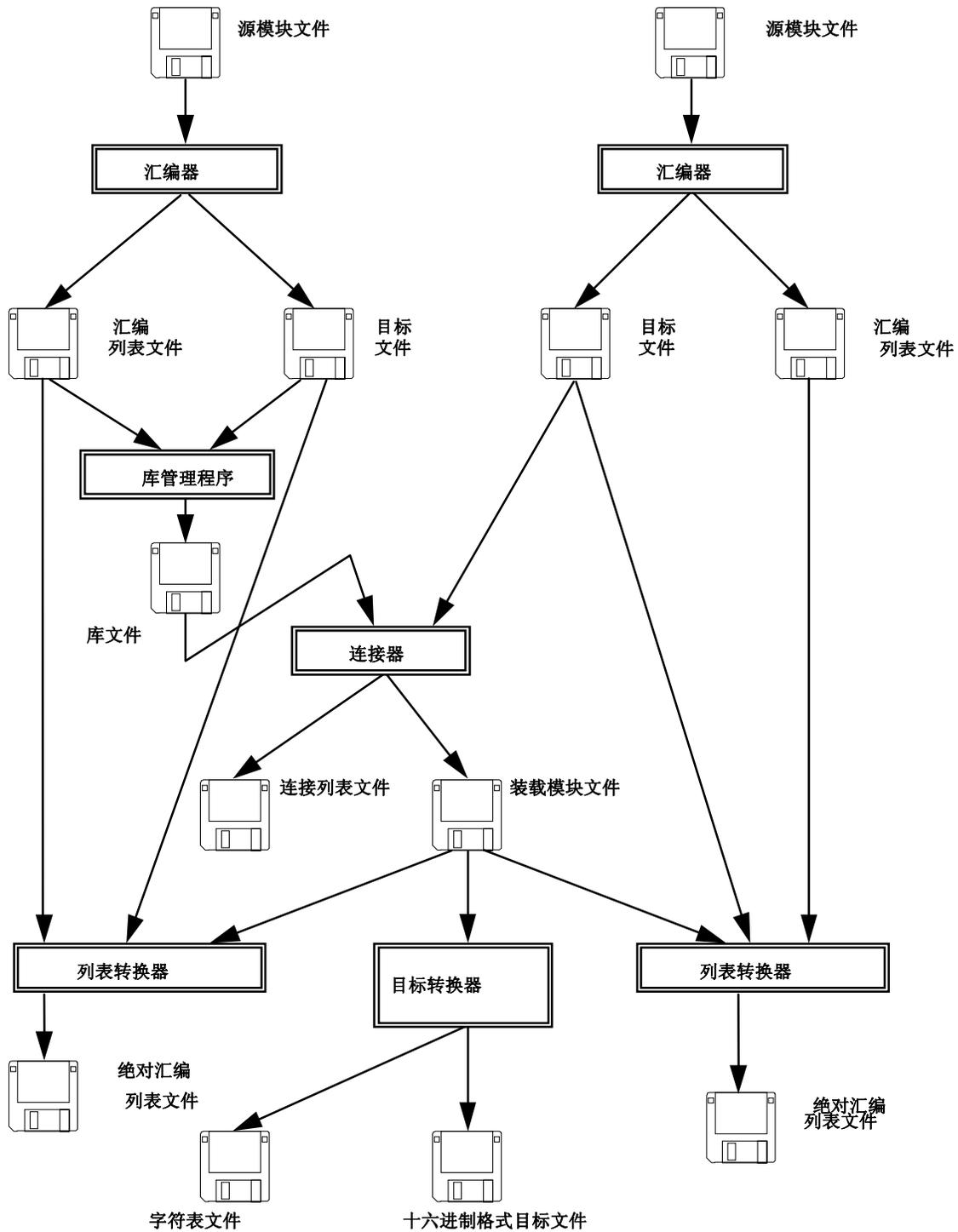


图 3-3 RA78K0S 执行过程 2



3.3 ST78K0S 的执行步骤

要验证 ST78K0S 的操作，请使用系统盘中的批处理文件（st.bat）。

结构化汇编器、连接器、目标转换器和表转换程序使用示例程序“test1.s”和“test2.s”顺序进行执行，它们是用结构化汇编语言中在 ra.bat 中编写的源文件。当输出任何错误消息时，批处理文件将终止。

目标设备的型号规格可以被输入到该批处理文件中（分别获得设备文件）。

下面的解释将 μ PD789024 作为目标设备。

■st.bat（ST78K0S 操作验证批处理程序）

```

echo off
cls
set  LEVEL=0

if "%1" == "" goto ERR_BAT

st78k0s -C%1 test1.s
ra78k0s test1.asm
if errorlevel 1 set LEVEL=1
st78k0s -C%1 test2.s
ra78k0s test2.asm
if errorlevel 1 set LEVEL=1
if %LEVEL% == 1 echo Assemble error !!
if %LEVEL% == 1 goto END

cls
lk78k0s test1.rel test2.rel -s -otest.lmf -ptest.map
if errorlevel 1 echo Link error !!
if errorlevel 1 goto END

cls
oc78k0s stsample
if errorlevel 1 echo Object conversion error !!
if errorlevel 1 goto END

cls
set LEVEL=0
lc78k0s -ltsample.lmf -rttest1.rel test1.prn
if errorlevel 1 set LEVEL=1
lc78k0s -ltest.lmf -rttest2.rel test2.prn
if errorlevel 1 set LEVEL=1
if %LEVEL% == 1 echo List conversion error !!
if %LEVEL% == 1 goto END

cls
echo No error.
goto END

```

```
:ERR_BAT
```

```
    echo Usage : st.bat chiptype
```

```
:END
```

```
    echo on
```

(1) 执行批处理文件。

指定目标设备型号，执行 ST78K0S 操作验证批处理程序。

```
C>st.bat 9024
```

将输出以下消息。

```
Structured assembler preprocessor for RA78K/OS Vx.xx [xx xxx xxxx]
  Copyright (C) NEC Electronics Corporation xxxx,xxxx
```

```
start
```

```
Target chip:uPD789024
Device file:Vx.xx
```

```
Conversion complete,          0 error (s) found.
```

```
78K/OS Series Assembler Vx.xx [xx xxx xxxx]
  Copyright (C) NEC Electronics Corporation,xxxx,xxxx
```

```
PASS_PARSE Start
PASS_OUTOBJ Start
```

```
Target chip:uPD789024
Device file:Vx.xx
```

```
Assembly complete,          0 error (s) and 0 warning (s) found.
```

```
Structured assembler preprocessor for RA78K/OS Vx.xx [xx xxx xxxx]
  Copyright (C) NEC Electronics Corporation xxxx,xxxx
```

```
start
```

```
Target chip:uPD789024
Device file:Vx.xx
```

```
Conversion complete,          0 error (s) found.
```

```
78K/OS Series Assembler Vx.xx [xx xxx xxxx]
  Copyright (C) NEC Electronics Corporation xxxx,xxxx
```

```
PASS_PARSE Start
PASS_OUTOBJ Start
```

```
Target chip:uPD789024
Device file:Vx.xx
```

```
Assembly complete,          0 error (s) and 0 warning (s) found.
```

清除屏幕。

78K/0S Series Linker Vx.xx [xx xxx xxxx]

Copyright (C) NEC Electronics Corporation xxxx,xxxx

Target chip:uPD789024

Device file:Vx.xx

Link complete, 0 error (s) and 0 warning (s) found.

清除屏幕。

78K/0S Series Object Converter Vx.xx [xx xxx xxxx]

Copyright (C) NEC Electronics Corporation xxxx,xxxx

Target chip:uPD789024

Device file:Vx.xx

Object Conversion Complete, 0 error (s) and 0 warning (s) found.

清除屏幕。

List Conversion Program for RA78K/0S Vx.xx [xx xxx xxxx]

Copyright (C) NEC Electronics Corporation xxxx,xxxx

Pass1:start...

Pass2:start...

Conversion complete.

List Conversion Program for RA78K/0S Vx.xx [xx xxx xxxx]

Copyright (C) NEC Electronics Corporation xxxx,xxxx

Pass1:start...

Pass2:start...

Conversion complete.

清除屏幕。

No errors.

(2) 检查 C 驱动器的内容。

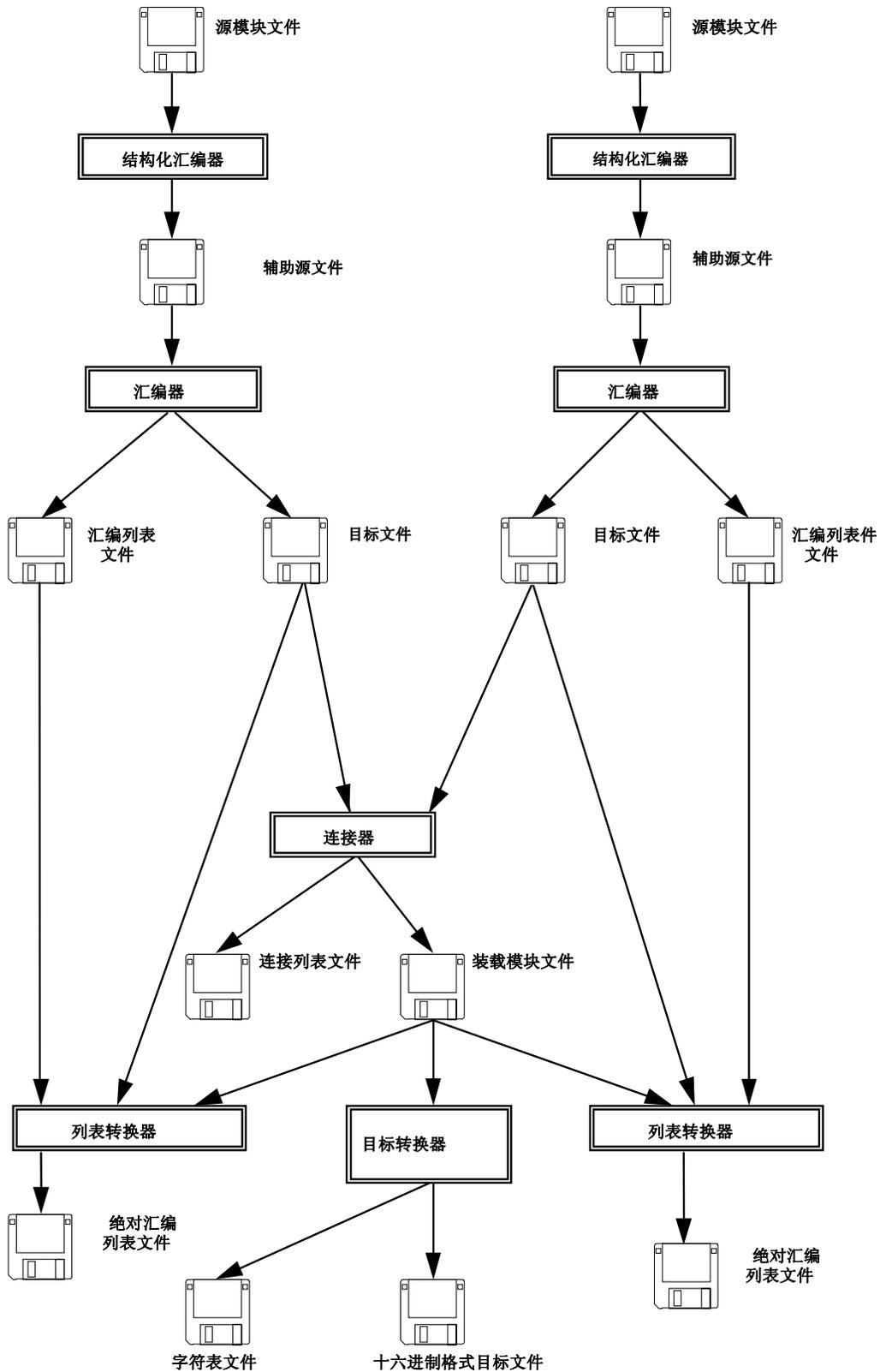
将输出以下文件。

test1.asm:	辅助源文件
test1.rel:	目标文件
test1.prn:	汇编表文件
test2.asm:	辅助源文件
test2.rel:	目标文件
test2.prn:	汇编列表文件
stsample.lmf:	载入模块文件
stsample.map:	连接列表文件
stsample.hex:	十六进制格式目标模块文件
stsample.sym:	符号表文件
test1.p:	绝对汇编表文件
test2.p:	绝对汇编表文件

(3) ST78K0S 执行过程概要

下面是 ST78K0S 执行过程的概要。

图 3-4 ST78K0S 执行过程 1



3.4 从命令行中完成汇编、连接与目标程序转换（DOS 提示，EWS）

本节介绍如何从命令行中执行汇编及目标程序转换。

(1) 汇编示例程序 K0SMMAIN.ASM。

在命令行中输入以下内容。

```
C>>ra78k0s -c9176 k0smain.asm
```

显示屏上显示以下消息。

```
78K/0S Series Assembler Vx.xx [xx xxx xx]  
Copyright (C) NEC Electronics Corporation xxxx
```

```
Pass1 Start
```

```
Pass2 Start
```

```
Target chip : uPD789176
```

```
Device file : Vx.xx
```

```
Assembly complete, 0 error(s) and 0 warning(s) found.
```

(2) 检查 C 驱动器的内容。

汇编器输出目标文件（K0SMMAIN.REL）及汇编表文件（K0SMMAIN.PRN）。

如果汇编时指定了 -E 选项，则汇编器将输出一个错误表文件（包含汇编错误的行的列表及其错误信息内容）。

(3) 汇编示例程序 K0SSUB.ASM。在命令行中输入以下内容。

```
C>>ra78k0s -c9176 k0ssub.asm
```

显示屏上显示以下消息。

```
78K/0S Series Assembler Vx.xx [xx xxx xx]  
Copyright (C) NEC Electronics Corporation xxxx
```

```
Pass1 Start
```

```
Pass2 Start
```

```
Target chip : uPD789176
```

```
Device file : Vx.xx
```

```
Assembly complete, 0 error(s) and 0 warning(s) found.
```

(4) 检查 C 驱动器的内容。

汇编器输出目标文件（K0SSUB.REL）及汇编表文件（K0SSUB.PRN）。

汇编时，如果指定了选项 -E，则汇编器将输出一个错误表文件。

(5) 创建一个指示文件。

指示文件是为连接器指示各段的存储单元的文件。

如果您需要扩大默认的 ROM/RAM 存储区，或要定义一个新内存区，请创建一个指示文件。

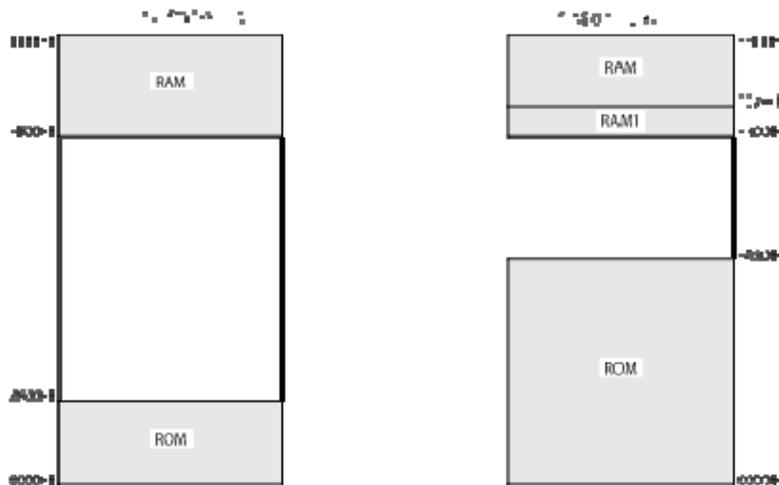
如果您希望为源模块文件中未定义为绝对段的段指定一个内存中的特定地址，你也需要创建一个指示文件。

连接时，使用 -D 选项将指示文件输入给连接器。

示例 要将 ROM 存储区从 (0H~7FFFH) 扩展至 (0H~FA7FH)，并扩展 RAM 存储区 (FE20H 至 FFFFH) 及 RAM1 (FB00H 至 FE1FH)，请将以下代码写入指令文件。

```
MEMORY ROM: (0H, 0FA80H)
MEMORY RAM1: (0FB00H, 320H)
MEMORY RAM: (0FE20H, 1E0H)
```

图 3-5 连接指示



- (6) 汇编的结果是，目标文件[K0SMMAIN.REL]与 [K0SSUB.REL]被连接起来。
输入 K0S.DR 作为指示文件。
在命令行中输入以下内容。

```
C>lk78k0s k0main.rel k0ssub.rel -dk0s.dr註1 -ok0s.lmf -pk0s.map -S註2
```

- 注 1. 如果未指定一个指示文件，则不必要。
2. 栈解析符号（_@STBEG）创建选项

显示屏上显示以下消息。

```
78K/0S Series Linker Vx.xx [xx xxx xx]
Copyright (C) NEC Electronics Corporation xxxx,xxxx
```

```
Target chip : uPD789176
Device file : Vx.xx
```

```
Link complete, 0 error(s) and 0 warning(s) found.
```

- (7) 检查 C 驱动器的内容。
连接器输出装入模块文件（K0S.lmf）及连接表文件（K0S.MAP）。
连接时如果指定了选项 -E，则连接器将输出一个错误表文件。
- (8) 连接的结果是，输出的装入模块文件（K0S.lmf）被转换为一个十六进制格式的文件。
在命令行中输入以下内容。

```
C>oc78k0s k0s.lmf -r -u0FFH
```

显示屏上显示以下消息。

```
78K/0S Series Object Converter Vx.xx [xx xxx xx]
Copyright (C) NEC Electronics Corporation xxxx,xxxx
```

```
Target chip : uPD789176
Device file : Vx.xx
```

```
Object Conversion Complete, 0 error(s) and 0 warning(s) found.
```

- (9) 检查 C 驱动器的内容。
目标转换器输出十六进制的目标模块文件（K0S.HEX）及符号表文件（K0S.SYM）。

(10) 创建如下的库文件。

将汇编器输出的目标文件（K0SSUB.REL）存储为一个库文件。
在命令行中输入以下内容。

```
C><lb78k0s < k0s.job
```

显示屏上显示以下消息。

```
78K/0S Series Librarian Vx.xx [xx xxx xx]  
Copyright (C) NEC Electronics Corporation xxxx,xxxx
```

```
*create k0s.lib  
*add k0s.lib k0sub.rel  
*exit
```

备注 下面划线的部分是“k0s.job”的内容。

(11) 检查 C 驱动器的内容。

库管理程序输出库文件（K0S.LIB）。

(12) 创建一个绝对汇编表文件，如下所示。

要创建绝对汇编表 K0SMMAIN.ASM，请向表转换程序输入[K0SMMAIN.REL]、[K0SMMAIN.ASM] 及[K0S.lmf]。
在命令行中输入以下内容。

```
C>lc78k0s k0smmain -lk0s.lmf
```

显示屏上显示以下消息。

```
List Conversion Program for RA78K0S Vx.xx [xx xxx xx]  
Copyright (C) NEC Electronics Corporation xxxx,xxxx
```

```
Pass1: start...  
Pass2: start...  
Conversion complete.
```

(13) 检查 C 驱动器的内容。

表转换程序输出绝对汇编表文件（K0SMMAIN.P）。

3.5 使用参数文件

如果当编译器或连接器启动时输入了两个或多个选项，则不能在命令行中完整地指定启动所需信息，或者会重复同一个指定。在这种情况下，要使用参数文件。

要使用参数文件，请在命令行中指定参数文件指定选项。

注意事项 不能用 **PM + 选项设置** 来指定参数文件。

汇编程序或连接程序是使用以下参数文件启动的。

```
>[path-name] RA78K0S Δ -F parameter-file-name
>[path-name] LK78K0S Δ -F parameter-file-name
>[path-name] OC78K0S Δ -F parameter-file-name
```

下面是一个使用例子。

```
示例      C>ra78k0S -Fpara.pra
           C>lk78k0S -Fpara.plk
           C>oc78k0S -Fpara.poc
```

参数文件是用编辑器创建的。所有应在命令行中指定的选项及输出文件名均可在参数文件中写入。

下面是一个使用编辑器创建参数文件的例子。

(para1.pra 的内容)

```
-c9024 k0smain.asm -e
```

(para.plk 的内容)

```
k0smain.rel k0ssub.rel -bmylib.lib -osample.lmf -s
```

(para.poc 的内容)

```
sample.lmf -u0FFH -osample.hex -r
```

第 4 章 结构化汇编器

结构化汇编器输入用 78K0S 系列微控制器的结构化汇编语言编写的源模块文件，将它们转换为汇编语言，然后将它们作为辅助源模块文件输出。

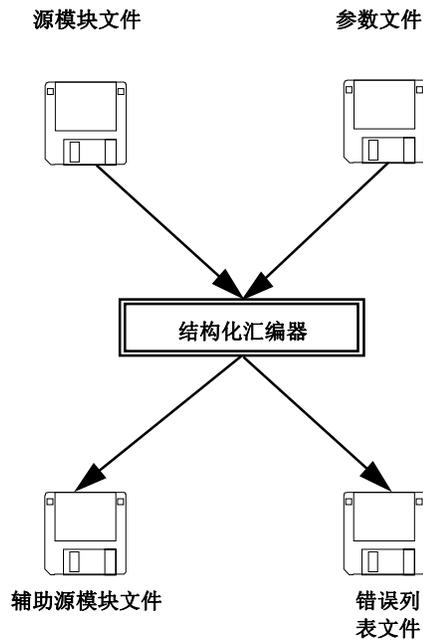
4.1 结构化汇编器的 I/O 文件

结构化汇编器的 I/O 文件如下表所示。

表 4-1 结构化汇编器的 I/O 文件

类型	文件名	说明	默认文件类型
输入文件	源模块文件	<ul style="list-style-type: none">• 这些是用结构化汇编语言编写的源模块文件。• 这些文件是用户创建的。	无
	参数文件	<ul style="list-style-type: none">• 这些是包含用于从文件中指定结构化汇编器选项的文件• 这些文件是用户创建的。	.PST
输出文件	辅助源模块文件	<ul style="list-style-type: none">• 这些是用汇编语言编写的源模块文件。	.ASM
	错误表文件	<ul style="list-style-type: none">• 这些是包含结构化汇编器错误数据的文件。	.EST

图 4-1 结构化汇编器的 I/O 文件



4.2 结构化汇编器的功能

- (1) 结构化汇编器读出源模块文件，将它们转换为汇编器输入源文件，然后将它们作为汇编器源文件输出。
- (2) 如果发生了与文件或系统有关的错误，结构化汇编器将输出一个异常中止错误；如果在源文件中发现了一个写错误，则它将输出一个致命错误或警告错误。
如果发生异常中止错误或致命错误，则辅助源文件将不能正常输出。不过，如果指定了-J 选项，则即使发生致命错误也可以输出辅助源文件。
- (3) 结构化汇编处理将根据启动时指定的选项来执行。关于结构化汇编器选项的详细说明，请参阅 **4.4 结构化汇编器选项**。
- (4) 如果结构化汇编处理已经正确无误地完成，则结构化汇编器将输出一个“已完成”消息，并将控制权返回给操作系统。

4.3 结构化汇编器启动

4.3.1 结构化汇编器启动

可以使用以下两个方法来启动结构化汇编器。

(1) 从命令行启动

通过输入以下命令来启动结构化汇编器。

```
X> [path-name] st78k0s [ $\Delta$ option] ...  $\Delta$ source-module-file-name [ $\Delta$ option]...  
|           |           |           |           |           |  
(1)      (2)      (3)      (4)      (5)      (4)
```

- (1) 当前的驱动器名
- (2) 当前的目录名
- (3) 结构化汇编器命令文件名
- (4) 输入操作结构化汇编器所需的详细指令。当指定一个以上的选项时，用空格隔开选项。关于结构化汇编器选项的详细说明，请参阅 [4.4 结构化汇编器选项](#)。
- (5) 要进行结构化汇编的源模块的文件名。

示例 `C>st78k0s -c9024 test1.s -e`

(2) 从一个参数文件启动

使用一个参数文件以避免在启动两个或多个结构化汇编操作时重复同一个结构化汇编器选项涉及的不便。

要从参数文件中启动结构化汇编器，请在命令行中指定参数文件指定选项（-F）。

从一个参数文件启动结构化汇编器，如下所示。

```
X>[path-name] st78k0s [ source-module-file-name] -F parameter-file-name
|           |           |           |
(1)        (2)          (3)          (4)
```

- (1) 当前的驱动器名
- (2) 当前的目录名
- (3) 参数文件指定选项
- (4) 参数文件名

编写参数文件内容的规则如下所示。

```
[[[Δ] option [Δoption]... [Δ] Δ]]...
```

如果在命令行中省略了源模块文件名，则只能在参数文件中指定一个源模块文件名。

源模块文件名还可以写在选项后面。

将在命令行中指定的所有选项和输出文件名写入参数文件。

示例 使用编辑器创建参数文件（test1.pst）。

- test1.pst 的内容

```
;Parameterfile
test1.s -osample.asm
-esample.est -c9024
```

- 使用参数文件（test1.pst）启动结构化汇编器。

```
C>st78k0s -ftest1.pst
```

4.3.2 执行启动和结束消息

(1) 执行启动消息

当结构化汇编器启动时，在显示屏上会出现以下执行启动消息。

```
Structured assembler preprocessor for RA78K/0S Vx.xx [xx xxx xx]
Copyright (C) NEC Electronics Corporation xxxx
```

(2) 处理显示消息

每隔结构化汇编器处理的 100 行显示一次“.”。

```
start.....
```

(3) 执行结束消息

如果未检测到任何错误，则以下消息将被输出到显示屏，同时控制权被移交给操作系统。

```
Target chip:uPD78xxxx
Device file:Vx.xx
```

```
Conversion complete,          0 error (s) found.
```

如果检测到错误，则显示屏上将显示检测到的错误数量，同时控制权被移交给操作系统。

```
TEST1.S (8) :F209 Syntax error
```

```
Target chip:uPD78xxxx
Device file:Vx.xx
```

```
Conversion complete,          1 error (s) found.
```

如果在结构化汇编时检测到一个使结构化汇编处理不能持续下去的致命错误，则结构化汇编器将向显示屏输出一条消息，取消结构化汇编处理，并将控制权返回操作系统。

示例 1 当指定一个不存在的源模块文件时

```
C>st78k0s sample.s
```

```
Structured assembler preprocessor for RA78K/0S Vx.xx [xx xxx xx]
Copyright (C) NEC Electronics Corporation 1996
```

```
A006 File not found 'SAMPLE.S'
```

```
Program aborted.
```

在上例中，指定一个不存在的源模块文件会产生一个错误，同时汇编停止。

示例 2 当指定一个不存在的选项时

```
C>st78k0s test1.s -z
```

```
Structured assembler preprocessor for RA78K/0S Vx.xx [xx xxx xx]
```

```
Copyright (C) NEC Electronics Corporation xxxx
```

```
A012 Missing parameter '-z'
```

```
Please enter 'ST78K0S --', if you want help messages.
```

```
Program aborted.
```

在上例中，指定一个不存在的选项会产生一个错误，同时汇编停止。

当显示一条错误消息且汇编停止时，请在**第 12 章 错误消息**中查找错误原因，并采取相应操作。

4.4 结构化汇编器选项

4.4.1 结构化汇编器选项的类型

结构化汇编器选项给出了结构化汇编器操作的详细说明。

结构化汇编器选项分为以下 13 种。

表 4-2 结构化汇编器选项

编号	种类	选项	说明
1	设备类型指定	-C	指定目标设备的类型。
2	字符符号指定	-SC	指定字符符号名的结束字符。
3	符号定义	-D	指定给 #IFDEF 指令等的符号
4	制表符数指定	-WT	指定转换的指令的输出位置。
5	包含文件路径指定	-I	指定驱动器 and 包含文件的目录
6	辅助源文件指定	-O	指定辅助源文件名。
7	错误列表文件指定	-E	指定错误列表文件名。
8	参数文件指定	-F	指定参数文件名。
9	调试数据输出指定	-GS	指定结构化汇编器源代码级调试数据的输出。
		-NGS	
10	辅助源文件强制输出指定	-J	指定辅助源文件的强制输出。
11	汉字码（2 字节代码）指定	-ZS	指定在注释语句中描述的汉字码类型。
		-ZE	
		-ZN	
12	设备文件搜索路径指定	-Y	指定用于搜索设备文件的路径。
13	帮助指定	--	在显示屏上输出帮助信息。

4.4.2 结构化汇编器选项说明

本节详细介绍各个结构化汇编器选项。

(1) 设备类型指定 (-C)

语法: -C 设备类型
默认预设: 不能省略

[功能]

指定作为结构化汇编器目标的设备。

[说明]

- 1) 务必指定选项 -C。结构化汇编器对指定的目标设备执行预处理，并生成汇编器源代码。请注意，如果省略了 -C，会发生错误。
- 2) 如果选项 -C 指定的设备类型与处理器设备类型指定控制指令指定的型号不同，将会发出一个警告。在这种情况下，结构化汇编器将优先使用选项 -C 指定的设备类型。
- 3) 选项 -C 指定的设备类型将作为一个处理器设备类型指定控制指令输出到辅助源文件中。不过，如果指定了一个与处理器设备类型指定控制指令相同名称的设备类型，这不会发生。

[使用示例]

μ PD789024 被指定为目标设备。

```
C>st78k0s test.s -c9024
```

[注]

不能省略选项 -C。不过，如果在源文件顶部写入了处理器设备类型指定控制指令（\$PROCESSOR），则在命令行中的指定可以省略。

有关设备类型的详细情况，请参阅各个设备的设备文件中的**使用说明**。

(2) 字符符号指定 (-SC)

语法: -SC 字符
默认预设: P 或 p

[功能]

当需要对字节/字区分符号名时，指定作为被判断目标的符号的最后字符。

[说明]

- 1) 根据待处理的数据是一个字节还是一个字，结构化汇编器将产生不同的指令。
如果这是一个代入，对于字节，产生 MOV 指令；对于字，产生 MOVW。
如果它是个字符符号保留字，则会产生一个字指令。
- 2) 如果指定了一个非保留字的符号，则根据其结束字符判断它是字节符号还是字符符号，并产生一个指令。
- 3) 如果未指定-SC 选项，则以 ‘P’ 或 ‘p’ 结束的符号将被判断为一个字符符号。
- 4) 待判断的字符仅限于字母等效字符。请注意，字母不分大小写。
- 5) 如果执行了一个以上的指定，则最后指定的项有效。

[使用示例]

一个以 @ 结束的符号被指定为一个字符符号。

```
C>st78k0s test.s -sc@
```

```
<test.s>
```

```
A = #3
```

```
AX = #3
```

```
SYM = #3
```

```
SYM@ = #3
```

```
<test.asm>
```

```
MOV A, #3 ;A = #3
```

```
MOVW AX, #3 ;AX = #3
```

```
MOV SYM, #3 ;SYM = #3
```

```
MOVW SYM@, #3 ;SYM@ = #3
```

(3) 符号定义指定(-D)

语法: `-Dsymbol-name[=numerical-value][,symbol-name[=numerical-value]...]`

[功能]

定义符号。

[说明]

- 1) 赋予符号的数值可以是二进制、八进制、十进制或十六进制的。
如果省略了数值指定, 则值将变为 1。
- 2) 使用该选项定义一个符号等于使用 `#define` 指令来定义一个符号。
- 3) 使用逗号作为分隔符, 最多可以在命令行中定义 30 项。
- 4) 该选项通常与 `#ifdef` 指令结合使用。
- 5) 如果执行了一个以上的指定, 则最后指定的项有效。
- 6) 如果该选项与 `#define` 指令一起指定, 将会输出一条警告消息, 且 `#define` 指令将被视为无效。
- 7) 字母不分大小写。

[使用示例]

符号“TRU”并定义为 1。

```
C>st78k0s test.s -dTRUE=1
```

(4) 制表符数指定 (-WT)

语法: -WT 数值 1
 : -WT [数值 1], 数值 2
 : -WT [数值 1], [数值 2], 数值 3
默认预设: -WT2, 3, 4

[功能]

指定制表符数，直到转换的汇编语言输出。

[说明]

- 1) 选项 -WT 允许汇编源指令的输出位置自由调整，以改进程序的可读性。
- 2) 数值 1 指定指令输出前的制表符数。
数值 2 指定指令操作数输出前的制表符数。
数值 3 指定指令注释输出前的制表符数。
- 3) 将数值指定为一个以下范围内的十进制数。
数值 1: 0 至 97
数值 2: 1 至 98
数值 3: 2 至 99
数值 1 < 数值 2 < 数值 3
- 4) 如果执行了一个以上的指定，则最后指定的项有效。

[使用示例]

“3”被指定给数值 1，“4”被指定给数值 2，“5”被指定给数值 3。

```
C>st78k0s test.s -wt3,4,5
```

(5) 包含文件路径指定 (-I)

语法: -I [drive:] 目录
默认预设: 当前目录

[功能]

指定待输出到结构化汇编器的包含文件路径的名称。

[说明]

- 1) 指定一个包含文件所在的驱动器和目录。
- 2) 如果省略选项-I, 则包含文件将被假定位于当前驱动器和当前目录中。
- 3) 如果执行了一个以上的指定, 则最后指定的项有效。

[使用示例]

包含文件的目录被指定为 b:\include。

```
C>st78k0s test.s -ib:\include
```

(6) 辅助源文件指定 (-O)

语法: -O [[[驱动器:] 目录] 文件名]
默认预设: -O input-file-name.ASM

[功能]

指定转换后辅助源文件的输出目的地及文件名。

[说明]

- 1) 指定转换后辅助源文件的输出驱动器、目录和文件名。
- 2) 如果省略了 -O 选项，则将在当前的目录中创建输出文件，方法是用 .ASM 代替输入文件的文件类型。
- 3) “NUL” 或 “AUX” 均可指定为文件名。
- 4) 当由于致命错误而停止处理时，辅助源文件将不被输出。
- 5) 如果执行了一个以上的指定，则最后指定的项有效。

[使用示例]

“sample.asm” 被指定为辅助源文件。

```
C>st78k0s test.s -osample.asm
```

(7) 错误列表文件指定 (-E)

语法: -E [[驱动器:][目录] 文件名]
默认预设: -E input-file-name.EST

[功能]

指定错误列表文件的输出目的地及文件名。

[说明]

- 1) 指定错误列表文件的输出驱动器、目录和文件名。
- 2) 如果省略了 -E 选项，则将在当前的目录中创建输出文件，方法是用.EST 代替输入文件的文件类型。
- 3) “NUL”或“AUX”均可指定为文件名。
- 4) 如果执行了一个以上的指定，则最后指定的项有效。

[使用示例]

“sample.est”被指定为错误列表文件。

```
C>st78k0s test.s -esample.est
```

(8) 参数文件指定 (-F)

语法: -F [[驱动器:] 目录] 文件名

[功能]

指定参数文件的文件名。

[说明]

- 1) 指定参数文件的输入驱动器、目录和文件名。
- 2) 文件名不能省略。如果省略了文件类型，则类型将被假定为“.PST”。
- 3) 该选项在使用选项 -D 在命令行中定义大量符号时有效。
- 4) 多次指定该选项会产生一个错误。
- 5) 禁止参数文件嵌套，否则会产生错误。
- 6) 参数文件中“;”或“#”以后的字符将被视为注释，直到左侧或文件结束。

[使用示例]

“sample.pst”被指定为参数文件。

```
C>st78k0s -fsample.pst
```

(9) 调试数据输出指定 (-GS/-NGS)

语法: -GS
 : -NGS
默认预设: -GS

[功能]

指定结构化汇编器源代码级调试数据的输出。

[说明]

- 1) 选项 -GS 指定向辅助源文件输出调试数据。
- 2) 选项 -NGS 使得选项 -GS 不可用。
- 3) 如果在输入源文件中有编译程序调试数据，则在文件顶部 -GS 选项将用 “;” 代替 “\$”。
- 4) 如果同时指定了 -GS 选项和 -NGS 选项，则最后指定的选项有效。
- 5) 如果省略，则假设指定了 -GS 选项。

[注]

在结构化汇编器源代码级调试时，确保指定调试数据输出说明（-GS/-NGS）。在汇编辅助源文件时，确保指定调试数据输出说明选项（-G/-GA）。结构化汇编器将所需的选项作为控制指令输出到辅助源文件中。

[使用示例]

指定向辅助源文件输出调试数据。

```
C>st78k0s test.s -gs
```

(10) 辅助源文件强制输出指定 (-J)

语法: -J

[功能]

当由于致命错误而停止处理时，强制输出辅助源文件。

[说明]

- 1) 当由于致命错误而停止处理时，辅助源文件将被输出。
- 2) 致命错误行按原样将输入源文件的映像输出到辅助源文件中。

[使用示例]

指定强制输出辅助源文件。

```
C>st78k0s test.s -j
```

(11) 汉字码指定 (-ZS/-ZE/-ZN)

语法: -ZS
 -ZE
 -ZN

默认预设: 根据操作系统的不同解释如下:
 Windows、HP-UX: -ZS
 SunOS、Solaris: -ZE

[功能]

指定在注释中描述的汉字码类型。

[说明]

- 1) 汉字码指定介绍如下
 - ZS: 解释为移位 JIS 码。
 - ZE: 解释为 EUC 码。
 - ZN: 不解释为汉字。
- 2) 这些选项对应于以下的汉字码指定控制指令。
 - ZS: \$KANJI CODE SJIS
 - ZE: \$KANJI CODE EUC
 - ZN: \$KANJI CODE NOTE
- 3) 汉字码指定的优先级顺序如下。
 - (1) -ZS/-ZE/-ZN 选项的指定
 - (2) 指定汉字码指定控制指令(\$KANJI CODE)
 - (3) 指定环境变量 LANG78K
 - (4) 指定每个操作系统的默认设置

[使用示例]

规定汉字被解释为移位 JIS 码。

```
A>st78k0s test.s -zs
```

(12) 设备文件搜索路径指定 (-Y)

- 语法: -Y [驱动器:] 目录
- 默认预设: 设备文件将按以下顺序进行搜索。
- 1) <.\dev> (用于启动 st78k0s.exe 的路径)
 - 2) 启动 st78k0s.exe 的路径
 - 3) 当前路径
 - 4) 环境变量 PATH 指定的路径

[功能]

指定用于搜索设备文件的路径。

[说明]

- 1) 从指定的路径中读出设备文件。
- 2) 如果指定了其他路径名, 则会发生错误。
- 3) 即使路径指定符号未写在目录的结尾, 也会假设它在那里。
- 4) 设备文件将按以下顺序进行搜索。
 1. -Y 选项指定的路径
 2. <.\dev> (用于启动 st78k0s.exe 的路径)
 3. 启动 st78k0s.exe 的路径
 4. 当前路径
 5. 环境变量 PATH 指定的路径

[使用示例]

指定从目录 c:\nectools\dev 中读出设备文件。

```
C>st78k0s test.s -ya:\nectools\dev
```

(13) 帮助指定 (--)

语法: --
 默认预设: 无显示

[功能]

选项 -- 显示一个帮助信息。

[应用]

帮助信息是对结构化汇编器选项的一系列说明。当执行结构化汇编器时请参阅这些帮助。

[说明]

当指定了 -- 选项时，所有其他的结构化汇编器选项均不可用。

注意事项 该选项不能在 **PM +** 上指定。

要参考 **PM+** 帮助，请在 <结构化汇编器选项设置> 对话框中单击帮助按钮。

[使用示例]

当指定了 -- 选项时，显示屏上将输出一个帮助信息。

```
C>st78k0s --
```

```
Structured assembler preprocessor for RA78K/0S Vx.xx [xx xxx xx]
Copyright (C) NEC Electronics Corporation xxxx
```

```
Usage:st78k0s [option[...]] input-file [option[...]]
```

The option is as follows ([] means omissible, ... means repetition).

```
-cx       :Select target chip. (x = 9002, 9014, etc.) *Must be specified.
-o[file]:Create the assembler source file[with the specified name].
-e[file]:Create the error list file[with the specified name].
-ffile   :Input options or source file name from specified file.
-idirectory   :Set include search path.
-sc[character]:Specify the last character of word symbol.
-wtn1/-wt[n1], n2/-wt[n1], [n2], n3
      :Specify the number of tabs up to output position of each field.
      n1:Output position mnemonic field.
      n2:Output position operand field. *Must be
      n3:Output position comment field. 0 <= n1 < n2 < n3 < 100.
-dname[=data][,name[=data][...]]
      :Define name[with data].
-gs/-ngs:Output the structured assembler source debug information to
      assembler source file / Not.
-j       :Create the assembler source file if fatal error occurred.
-zs/-ze/-zn   :Change source regulation.
      -zs:SJIS code usable in comment.
      -ze:EUC code usable in comment.
```

Press RETURN to continue...

```
-zn:no multibyte code in comment.
-ydirectory   :Set device file search path.
--           :Show this message.
```

```
DEFAULT ASSIGNMENT:-o -e -scp -wt2,3,4 -gs
```

4.5 在 PM+ 中设置选项

本节将介绍如何从 PM + 中设置结构化汇编器选项。

4.5.1 设置选项方法

从 PM+ 的 [Tools] 菜单中选择 [Structured Assembler Options...], 弹出 <Structured Assembler Options> 对话框。在此对话框中输入所需选项, 即可设置结构化汇编器选项。

图 4-2 <Structured Assembler Options>对话框 (当选中 <<Output>>选项卡时)

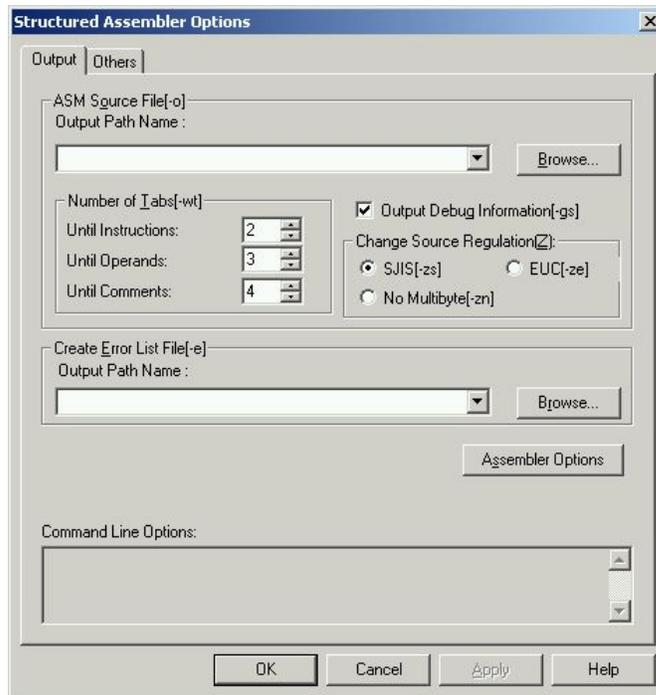


图 4-3 <Assembler Source Options>对话框

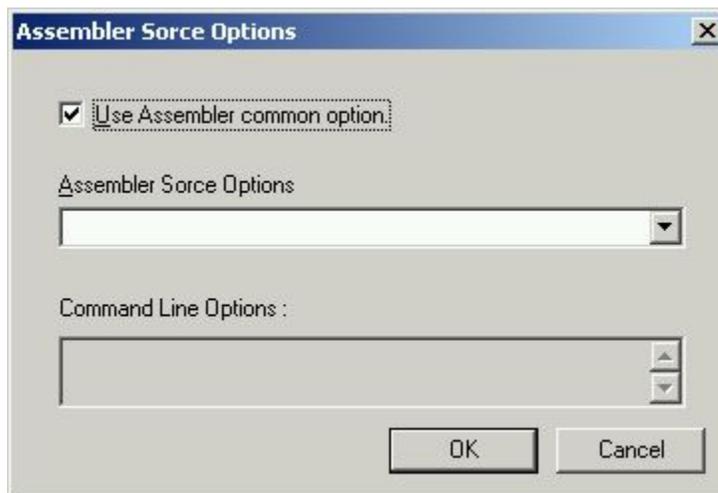
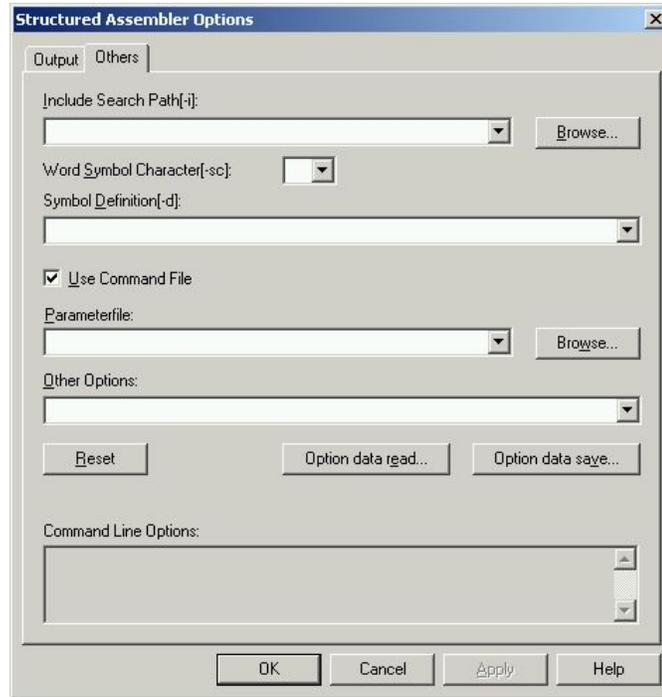


图 4-4 <Structured Assembler Options>对话框（当选中 <<Others>>选项卡时）



4.5.2 选项

下面介绍<Structured Assembler Options Setup>对话框中的各个选项。

- **ASM Source File [-o]**
Output Path Name
使用 [Browse...] 按钮或通过直接输入指定 ASM 源文件输出路径。
- **Number of Tabs [-wt]**
指定制表符数，直到转换的汇编语言输出。
指令、操作数和注释前的制表符数可以单独指定。
- **Output Debug Information [-gs]**
将调试信息输出至辅助源文件。
- **Change Source Regulation [Z]**
选择在源代码注释中使用的多字节代码类型（SJIS [-zs]、EUC [-ze]、无 Multibyte [-zn]）。
- **Create Error List File [-e]**
Output Path Name
要输出一个错误表文件，请在文本框中输入文件名。
使用 [Browse...] 按钮指定路径。
- **Assembler Options**
指定汇编器源模块文件的汇编器选项。
 - **Use Assembler common option**
启用在<Structured Assembler Options>对话框中设置的汇编器的各个通用选项。
 - **Assembler Source Options**
输入包括选项名在内的字符串，以启用输出汇编器源文件的选项。
- **Include Search Path [-i]**
使用 [Browse...] 按钮或通过直接输入指定包含源文件路径。
- **Word Symbol Character [sc-]**

指定被定义为一个字的符号的最后字符。此后，生成一个指令，以使用最后一个字符决定它是一个字符还是一个字节符号。

- **S**ymbol Definition [-d]
输入要定义为一个符号的数值。
- **U**se Command File
选择该复选框来创建一个命令文件。
- **P**arameterfile
使用 [**B**rowse...] 按钮或通过直接输入读取用户定义的参数文件。
- **O**ther Options
要指定一个非用复选框或单选按钮选择的选项，请在输入框中输入选项。
- **R**eset
清除输入内容。
- **O**ption data read...
打开 <Option Data Read>对话框，在指定选项数据文件后，读取该文件。
- **O**ption data save...
打开 <Option Data Save>对话框，用指定的名称将选项数据保存至选项数据文件中。
- **C**ommand Line Options
该编辑框是只读的。显示当前设置的选项字符串。

第 5 章 汇编器

汇编器输入用 78K0S 系列微控制器的汇编语言编写的源模块文件，并将它们转换为机器语言代码。

汇编器还输出列表文件，例如汇编列表文件和错误列表文件。

如果发生汇编错误，将向汇编列表文件和错误列表文件输出一条错误消息，以澄清错误原因。

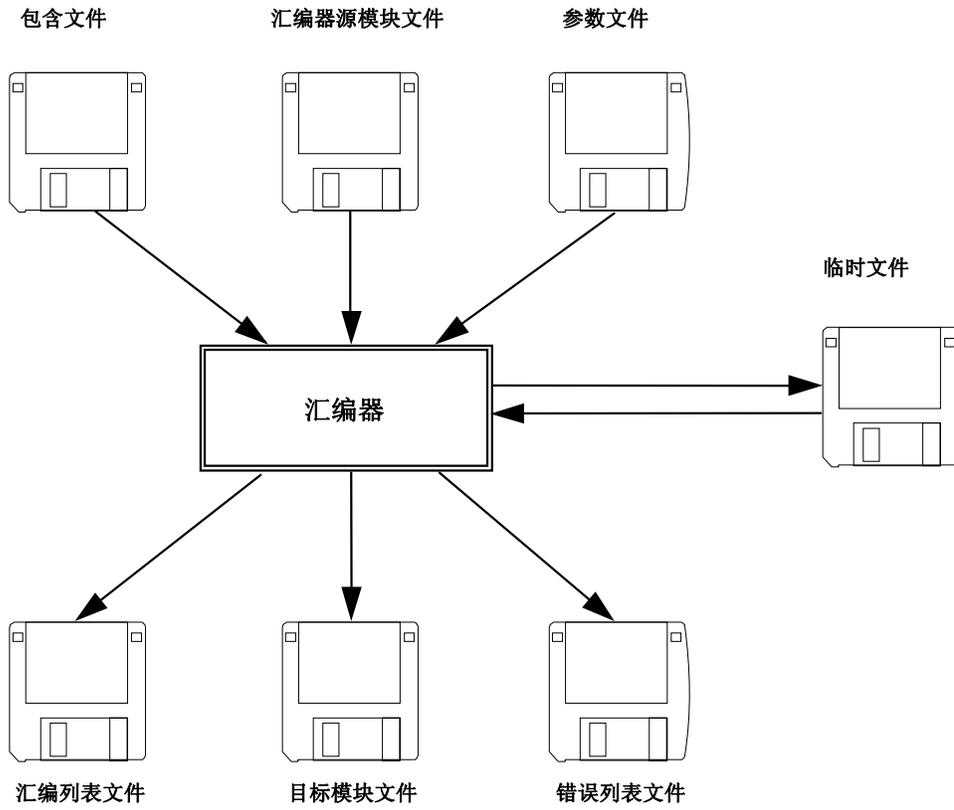
5.1 汇编器的 I/O 文件

汇编器的 I/O 文件如下表所示。

表 5-1 汇编器的 I/O 文件

类型	文件名	说明	默认文件类型
输入文件	汇编器源模块文件	<ul style="list-style-type: none">• 这些是用 78K0S 系列微处理器的汇编语言编写的源模块文件。• 这些文件是用户创建的。	.ASM
	包含文件	<ul style="list-style-type: none">• 这些文件用于汇编源模块文件引用。• 这些是用 78K0S 系列微处理器的汇编语言编写的文件。• 这些文件是用户创建的。	无
	参数文件	<ul style="list-style-type: none">• 这些文件包含执行过的文件的参数。• 这些文件是用户创建的。	.PRA
输出文件	目标文件	<ul style="list-style-type: none">• 这些是二进制文件，包括有关机器语言数据和机器语言单元地址的浮动数据和符号数据。	.REL
	汇编列表文件	<ul style="list-style-type: none">• 这些是包含诸如汇编列表和交叉引用列表等汇编数据的文件。	.PRN
	错误列表文件	<ul style="list-style-type: none">• 这些是包含汇编时产生的错误数据的文件。	.ERA
I/O 文件	临时文件	<ul style="list-style-type: none">• 这些是为了汇编由汇编器自动创建的文件。当汇编结束时，临时文件被删除。	RAxxxx.\$n (n = 1 至 4)

图 5-1 汇编器的 I/O 文件



5.2 汇编器的功能

- (1) 汇编器读出源模块文件，将它们从汇编语言文件转换为汇编语言文件。
- (2) 当发生错误时，汇编器将输出一个异常中止错误。如果汇编器在源模块中发现编写错误，它将输出一条“致命错误”或“警告错误”消息
如果输出“异常中止错误”或“致命错误”消息，则目标文件将不能正常输出。不过，当指定了 `-J` 选项时，即使发生了致命错误，目标文件也能被输出。
- (3) 汇编器将根据汇编器启动时指定的汇编器选项来执行汇编。关于汇编器选项的详细说明，请参阅 **5.4 汇编器选项**。
- (4) 如果汇编正确无误地完成，则汇编器将输出一条“汇编已完成”消息，并将控制权返回给操作系统。

5.3 汇编器启动

5.3.1 汇编器启动

可以使用以下两个方法来启动汇编器。

(1) 从命令行启动

```
X > [path-name] RA78K0S [ $\Delta$ option] ...  $\Delta$ source-module-file-name [ $\Delta$ option] ... [ $\Delta$ ]  
|         |         |         |         |         |  
(1)      (2)      (3)      (4)      (5)      (4)
```

- (1) 当前的驱动器名
- (2) 当前的目录名
- (3) 汇编器的命令文件名
- (4) 输入操作汇编器所需的详细指令。

当指定两个或多个汇编器选项时，请用一个空格将各个选项隔开。关于汇编器选项的详细说明，请参阅 **5.4 汇编器选项**。

- (5) 要进行汇编的源模块的文件名。

示例 C>ra78k0s -c9024 k0smain.asm -e -np

(2) 从一个参数文件启动

当启动汇编器所需的数据不适合命令行，或当每次执行汇编都要指定相同的汇编器选项时，使用参数文件。

要从参数文件中启动汇编器，请在命令行中指定参数文件指定选项（-F）。

从一个参数文件启动汇编器的方法如下所示。

```
X > RA78K0S [ $\Delta$ source-module-file]  $\Delta$ -F parameter-file-name
```

```

          |           |
          (2)        (1)

```

(1) 一个包含启动汇编器所需数据的文件。

(2) 参数文件（指定选项）

使用编辑器创建参数文件。

编写参数文件内容的规则如下所示。

```
[[[ $\Delta$ ] option [ $\Delta$ option] ... [ $\Delta$ ] $\Delta$ ] ...
```

如果在命令行中省略了源模块文件名，则只能在参数文件中指定一个源模块文件名。

源模块文件名还可以写在选项后面。

将在命令行中指定的所有汇编器选项和输出文件名写入参数文件。

关于参数文件的详细说明，请参阅 **5.4.3 汇编器选项说明**。

示例 使用编辑程序创建参数文件（K0SMMAIN.PRA）。

- K0SMMAIN.PRA 的内容

```
;parameter file
k0smmain.asm -osample.rel
-psample.prn
```

- 使用参数文件（K0SMMAIN.PRA）启动汇编器。

```
C>ra78k0s -fk0smmain.pra
```

5.3.2 执行启动和结束消息

(1) 执行启动消息

当汇编器启动时，在显示屏上会出现以下执行启动消息。

```
78K/0S Series Assembler Vx.xx [xx xxx xx]
Copyright (C) NEC Electronics Corporation xxxx,xxxx
```

(2) 执行结束消息

如果未检测到汇编产生的错误，则汇编器将在显示屏上输出以下消息，同时控制权被移交给操作系统。

```
Pass1 Start
Pass2 Start

Target chip:uPD78xxx
Device file:Vx.xx

Assembly complete, 0 error(s) and 0 warning(s) found.
```

如果检测到汇编产生的错误，则汇编器将在显示屏上输出错误编号，同时控制权被移交给操作系统。

```
Pass1 Start
K0SMAIN.ASM(12) :F201 Syntax error
Pass2 Start
K0SMAIN.ASM(12) :F201 Syntax error
K0SMAIN.ASM(29) :F407 Undefined symbol reference 'CONVAH'
K0SMAIN.ASM(29) :F303 Illegal expression

Target chip:uPD78xxx
Device file:Vx.xx

Assembly complete, 3 error(s) and 0 warning(s) found.
```

如果在汇编时检测到一个使汇编处理不能持续下去的致命错误，则汇编器将向显示屏输出一条消息，取消汇编处理，并将控制权返回操作系统。

示例 1 指定一个不存在的源模块文件时。

```
C>ra78k0s sample.asm

78K/0S Series Assembler Vx.xx [xx xxx xx]
  Copyright (C) NEC Electronics Corporation xxxx,xxxx

A006 File not found 'SAMPLE.ASM'
Program aborted.
```

上例指定了一个不存在的源模块文件。发生了一个错误，同时汇编器中止汇编。

示例 2 指定一个不存在的汇编器选项。

```
C>ra78k0s k0smain.asm -z

78K/0S Series Assembler Vx.xx [xx xxx xx]
  Copyright (C) NEC Electronics Corporation xxxx,xxxx

A012 Missing parameter '-z'
Please enter 'RA78K0S--', if you want help messages.
Program aborted.
```

上例指定了一个不存在的汇编器选项。发生了一个错误，同时汇编器中止汇编。当显示一条错误消息且汇编异常中止时，请在**第 12 章 错误消息**中查找原因，并采取相应操作。

5.4 汇编器选项

5.4.1 汇编器选项类型

汇编器选项是汇编器操作的详细说明。汇编器选项分为以下 15 种。

表 5-2 汇编器选项 (1/2)

编号	种类	选项	说明
1	设备类型指定	-C	指定目标设备的设备类型。
2	目标文件输出指定	-O	指定输出目标文件。
		-NO	
3	强制目标文件输出指定	-J	强制输出目标文件。
		-NJ	
4	调试数据输出指定	-G	输出调试数据（局部符号数据）至一个目标文件。
		-NG	
		-GA	输出汇编器源调试数据至目标文件。
		-NGA	
5	包含文件读取路径指定	-I	从包含文件中指定的路径读取。
6	汇编列表文件输出指定	-P	指定输出汇编列表文件。
		-NP	
7	汇编列表文件数据指定	-KA	将汇编表输出至汇编列表文件。
		-NKA	
		-KS	将符号表输出至汇编列表文件。
		-NKS	
		-KX	将交叉引用表输出至汇编列表文件。
		-NKX	
8	汇编列表文件格式指定	-LW	更改可在汇编列表文件中 1 行上打印的字符数。
		-LL	更改可在汇编列表文件 1 页中打印的行数。
		-LH	输出在汇编列表文件的首部指定的字符串。
		-LT	更改一个制表符中的空格数。
		-LF	在汇编列表文件结尾处插入一个换行代码。
		-NLF	

表 5-2 汇编器选项 (2/2)

编号	种类	选项	说明
9	错误列表文件输出指定	-E	输出一个错误列表文件。
		-NE	
10	参数文件指定	-F	从指定的文件输入输入文件名和汇编器选项。
11	临时文件创建路径指定	-T	在指定的路径中创建一个临时文件。
12	汉字码指定	-ZS	注释中所描述的汉字解释为移位 JIS 码。
		-ZE	注释中所描述的汉字解释为 EUC 码。
		-ZN	注释中所描述的字符不解释为汉字。
13	设备文件搜索路径指定	-Y	从指定的路径读取设备文件。
14	符号定义指定	-D	定义符号。
15	帮助指定	--	在显示屏上显示帮助信息。

5.4.2 汇编器选项的优先级

表 5-3 说明了当同时指定了两个汇编器选项时，哪个选项有效。

表 5-3 汇编器选项的优先级

	-NO	-NP	-NKA	-NKS	-KX	-NKX	--
-J	×						×
-G	×						×
-P			△	△		△	×
-KA		×					×
-KS		×			×		×
-KX		×					×
-LW		×					×
-LL		×					×
-LH		×					×
-LT		×					×
-LF		×					×

↑
垂直轴

←
水平轴

[用×标记的项]

当在水平轴上的选项被指定时，则在垂直轴选择中显示的选项不可用。

示例 `C>ra78k0s -c9024 k0smain.asm -no -lw80 -lf`

-LW 和 -LF 选项不可用。

[用△标记的项]

当在水平轴上的所有三个选项被指定时，则在垂直轴选择中显示的选项不可用。

示例 `C>ra78k0s -c9024 k0smain.asm -p -nka -nks -nkx`

同时指定了 -NKA、-NKS 和 -NKX 选项，所以 -P 选项不可用。

如果同时指定了一个选项及其对应的‘N’（例如，-O 与-NO） 则最后指定的选项可用。

示例 `C>ra78k0s -c9024 k0smain.asm -o -no`

-NO 选项在 -O 选项后指定，所以 -O 选项不可用，而 -NO 选项可用。

在表 5-3 汇编器选项的优先级中没有介绍的选项对其他选项没有特别的影响。不过，当指定选项‘--’时，所有其他的选项均不可用。

5.4.3 汇编器选项说明

本节详细介绍各个汇编器选项。

(1) 设备类型指定 (-C)

语法: -C 设备类型
默认预设: 不能省略

[功能]

选项-C 指定目标设备的设备类型。

[应用]

务必指定选项 -C。汇编器对目标设备执行汇编，并生成该设备的目标代码。

[说明]

对于使用选项 -C 指定的设备，请参阅“使用设备时要注意的问题”。

[注]

不能省略选项 -C。不过，如果在源模块开始描述了具有相同功能的控制指令，则可以省略命令行指定。

<p>▽\$▽PROCESSOR▽(▽device-type▽) ▽\$▽PC▽(▽device-type▽) ;简化形式</p>

有关控制指令的信息，请参阅语言部分第 4 章 控制指令。

[使用示例]

在命令行中指定选项 -C 的方法如下所示。

```
C>ra78k0s -c9024 k0smain.asm
```

(2) 目标文件输出指定(-O/-NO)

语法: -O [输出文件名]
 : -NO
默认预设: -O input-file-name.REL

[功能]

- 1) 选项 -O 指定输出目标文件。它还指定文件输出的位置和文件名。
- 2) 选项 -NO 指定不输出任何目标文件。

[应用]

使用选项 -O 指定目标文件输出的位置或更改其文件名。
在执行汇编时，指定选项 -NO 仅输出汇编列表文件。这将会缩短汇编的时间。

[说明]

- 1) 均可在[输出文件名]中指定磁盘类型文件名，设备类型文件名 NUL 和 AUX，以及路径名。当指定了设备类型文件名 CON、PRN 和 CLOCK 时，则会发挥终止错误。
- 2) 即使指定了选项-O，如果发生致命错误，则无法输出目标文件。
- 3) 如果在指定选项-O 时省略了驱动器名，目标文件名将会输出到当前驱动器。
- 4) 如果在指定选项-O 时省略了输出文件名，输出文件名将会变为'input file name.REL'。
- 5) 如果同时指定了选项-O 和选项-NO，则后指定的选项优先。

[使用示例]

指定输出目标文件 (SAMPLE.REL)。

```
C>ra78k0s -c9024 k0smain.asm -osample.rel
```

(3) 强制目标文件输出指定 (-J/-NJ)

语法: -J
 :
默认预设: -NJ

[功能]

- 1) 选项 -J 指定，即使发生致命错误也可输入目标文件
- 2) 选项 -NJ 使选项 -J 不可用。

[应用]

正常情况下，当发生致命错误时，无法输出目标文件。当您想要在出现已发生致命错误的通知时执行程序，指定选项 -J 输出目标文件。

[说明]

- 1) 当指定选项 -J 时，即使发生致命错误也会输入目标文件
- 2) 如果同时指定了选项 -J 和选项 -NJ，则后指定的选项优先。

[使用示例]

即使发生致命错误，指定输出目标文件。

```
C>ra78k0s -c9024 k0smain.asm -j
```

(4) 调试数据输出指定 (-G/-NG, -GA/-NGA)

(a) -G/-NG

语法: -G
 : -NG
 默认预设: -G

[功能]

- 1) 选项 -G 指定将调试数据（局部符号数据）添加到目标文件。
- 2) 选项 -NG 使选项 -G 不可用。

[应用]

- 1) 在执行包含局部符号数据的数据符号调试时使用选项 -G。
- 2) 在如下 3 种情况下使用选项 -NG。
 1. 仅进行全局符号的符号调试
 2. 进行没有符号的调试
 3. 仅当需要目标（使用 PROM 等赋值）时

[说明]

如果同时指定了选项 -G 和选项 -NG，则后指定的选项优先。

[注意]

还可在源模块开始处写入与选项 -G 和选项 -NG 具有相同功能的控制指令。

∇\$∇DEBUG	
∇\$∇DG	; 缩写形式
∇\$∇NODEDEBUG	
∇\$∇NODG	; 缩写形式

有关控制指令的信息，请参阅语言部分的第 4 章控制指令。

[使用示例]

指定添加调试数据到目标文件。

```
C>ra78k0s -c9024 k0smain.asm -g
```

(b) -GA/-NGA

语法: -GA
 :
默认预设: -GA

[功能]

- 1) 选项 -GA 指定由结构化的汇编器将源调试数据添加到目标文件。
- 2) 选项 -NGA 使选项 -GA 不可用。

[应用]

- 1) 当执行汇编器或结构化的汇编器源级别的调试时使用选项 -GA。要执行源级别的调试，您将需要单独可用的集成调试程序。
- 2) 在如下 2 种情况下使用选项 -NGA。
 1. 在没有汇编器源的情况下调试
 2. 仅当需要目标（使用 PROM 等赋值）时

[说明]

- 1) 如果同时指定了选项 -GA 和选项 -NGA，则后指定的选项优先。
- 2) 不管它所指定的位置如何，选项 -GA 都优先其他选项。

[注意]

还可在源模块开始处写入与选项 -GA 和选项 -NGA 具有相同功能的控制指令。

<pre>▽\$▽DEBUGA ▽\$▽NODEBUGA</pre>

有关控制指令的信息，请参阅语言部分的第 4 章控制指令。

[使用示例]

指定添加汇编器源调试数据到目标文件。

```
C>ra78k0s -c9024 k0smain.asm -ga
```

(5) 包含文件读取路径指定 (-I)

语法: -I 路径名 [, 路径名] ... (可指定两个或多个路径名)

默认预设: 源文件中所包含的路径

 : 环境变量 (INC78K0S) 所指定的路径

[功能]

选项 -I 指定从所指定的路径输入由源模块中 'Sinclude' 所指定的包含文件。

[应用]

使用选项 -I 检索来自某个路径的包含文件。

[说明]

- 1) 通过使用 ';' 分隔, 一次可指定两个或多个路径名。
- 2) 在 ';' 前或后不可输入空格。
- 3) 在指定选项 -I 或多个选项 -I 之后指定了两个或多个路径名, 则将会以指定的顺序检索使用 'Sinclude' 所指定的文件。此后, 将会以默认的顺序检索文件。
- 4) 如果在指定选项 -I 后未指定任何路径名, 或如果省略了路径名, 则会发生终止错误。
- 5) 如果使用选项 -I 指定 9 个或更多的路径名, 则会发生终止错误。

[使用示例]

从目录 B:\SAMPLE 读取包含文件。

```
C>ra78k0s -c9024 k0smain.asm -ib:\sample
```

(6) 汇编列表文件输出指定 (-P/-NP)

语法: -P [输出文件名]
 :
 -NP
默认预设: -P input-file-name.PRN

[功能]

- 1) 选项 -P 指定输出汇编列表文件。它还可指定输出文件的目的地和文件名。
- 2) 选项 -NP 可使选项 -P 不可用。

[应用]

- 1) 指定选项 -P 更改汇编列表文件的输出目的地或输出文件名。
- 2) 在执行汇编时，指定选项 -NP 仅输出目标文件。这将会缩短汇编的时间。

[说明]

- 1) 可将文件名指定为磁盘类型的文件名或设备类型的文件名。然而，仅可将 CON、PRN、NUL 和 AUX 指定为设备类型文件名。如果指定了 CLOCK，则会发生终止错误。
- 2) 如果在指定选项 -P 时省略了输出文件名，汇编列表文件名将会变为 'input file name.PRN'。
- 3) 如果在指定选项 -P 时省略了驱动器名，汇编列表文件名将会输出到当前驱动器。
- 4) 如果同时指定了选项 -P 和选项 -NP，则后指定的选项优先。

[使用示例]

创建汇编列表文件 (SAMPLE.PRN)。

```
C>ra78k0s -c9024 k0smain.asm -psample.prn
```

(7) 汇编列表文件数据指定 (-KA/-NKA, -KS/-NKS, -KX/-NKX)

(a) -KA/-NKA

语法: -KA
 :
默认预设: -KA

[功能]

- 1) 选项 -KA 指定输出汇编表到汇编列表文件。
- 2) 选项 -NKA 使选项 -KA 不可用。

[应用]

指定选项 -KA 以输出汇编表。

[说明]

- 1) 如果同时指定了选项 -KA 和选项 -NKA，则后指定的选项优先。
- 2) 如果指定了选项 -NKA、选项 -NKS 和选项 -NKX，则无法输出汇编列表文件。

[使用示例]

输出汇编列表。

```
C>ra78k0s -c9024 k0smain.asm -ka
```

引用 KOSMAIN.PRN。

Assemble list

ALNO STNO ADRS OBJECT M I SOURCE STATEMENT

```

1      1
2      2          NAME  SAMPM
3      3          ;*****
4      4          ;
5      5          ;  HEX -> ASCII Conversion Program
6      6          ;
7      7          ;   main-routine
8      8          ;
9      9          ;*****
10     10
11     11          PUBLIC MAIN, START
12     12          EXTRN CONVAH
13     13          EXTRN @_STBEG
14     14
15     15 ----          DATA  DSEG  saddr
16     16 0000          HDTSA: DS 1
17     17 0001          STASC: DS 2
18     18
19     19 ----          CODE  CSEG AT 0H
20     20 0000 R0000    MAIN: DW START
21     21
22     22 ----          CSEG
23     23 0000          START:
24     24
25     25          ;chip initialize
26     26 0000 RF0000    MOVW  AX, @_STBEG
27     27 0003 E61C      MOVW  SP, AX
28     28
29     29 0005 RF5001A    MOV   HDTSA, #1AH
30     30 0008 RFC0000    MOVW  HL, #HDTSA          ;set hex 2-code data
in HL register
31     31
32     32 000B R220000    CALL  !CONVAH          ;convert ASCII<- HE
X
33     33          ;output BC-register
<- ASCII code
34     34 000E RF80100    MOVW  DE, #STASC          ;set DE <- store ASC
II code table
35     35 0011 0A27      MOV   A, B
36     36 0013 EB        MOV   [DE], A
37     37 0014 88        INCW  DE
38     38 0015 0A25      MOV   A, C
39     39 0017 EB        MOV   [DE], A
40     40
41     41 0018 30FE      BR   $$
42     42
43     43          END

```

(b) -KS/-NKS

语法: -KS
 : -NKS
 默认预设: -NKS

[功能]

- 1) 选项 -KS 指定将在符号表之前的汇编列表输出到汇编列表文件。
- 2) 选项 -NKS 可使选项 -KS 不可用。

[应用]

指定选项 -KS 以输出符号表。

[说明]

- 1) 如果同时指定了选项 -KS 和选项 -NKS，则后指定的选项优先。
- 2) 如果同时指定了选项 -KS 和选项 -KX，则会忽略选项 -KS。
- 3) 如果指定了选项 -NKA、选项 -NKS 和选项 -NKX，则无法输出汇编列表文件。

[使用示例]

输出符号表。

```
C>ra78k0s -c9024 k0smain.asm -ks
```

引用 K0SMAIN.PRN。

(汇编列表在符号列表之前输出。)

Symbol	Table	List	Symbol	Table	List		
VALUE	ATTR	RTYP	NAME	VALUE	ATTR	RTYP	NAME
	CSEG		?CSEG		CSEG		CODE
---H		EXT	CONVAH		DSEG		DATA
FE20H	ADDR		HD TSA	0H	ADDR	PUB	MAIN
	MOD		SAMP M	0H	ADDR	PUB	START
FE21H	ADDR		STASC				

(c) **-KX/-NKX**

语法: -KX
 : -NKX
 默认预设: -NKX

[功能]

- 1) 选项 **-KX** 指定将在交叉引用表之前的汇编表输出到汇编列表文件。
- 2) 选项 **-NKX** 使选项 **-KX** 不可用。

[应用]

当想要知道在源模块中在何位置以何种程度引用源模块中所定义的符号时，或如某个符号被引用到汇编列表中哪一行的信息时，指定选项 **-KX** 以输出交叉引用表。

[说明]

- 1) 如果同时指定了选项 **-KX** 和选项 **-NKX**，则后指定的选项优先。
- 2) 如果同时指定了选项 **-KS** 和选项 **-KX**，则会忽略选项 **-KS**。
- 3) 如果指定了选项 **-NKA**、选项 **-NKS** 和选项 **-NKX**，则无法输出汇编列表文件。

[注意]

还可在源模块开始处写入与选项 **-KX/-NKX** 具有相同功能的控制指令。

∇\$∇XREF	
∇\$∇XR	; 缩写形式
∇\$∇NOXREF	
∇\$∇NOXR	; 缩写形式

有关控制指令的信息，请参阅语言部分的**第 4 章控制指令**。

[使用示例]

输出交叉引用表。

```
C>ra78k0s -c9024 k0smain.asm -kx
```

引用 KOSMAIN.PRN。

汇编列表在交叉引用列表之前输出。

Cross-Reference List

NAME	VALUE	R	ATTR	RTYP	SEGNAME	XREFS
?CSEG			CSEG		?CSEG	21#
CODE			CSEG		CODE	18#
CONVAH	----H	E		EXT		12@ 29
DATA			DSEG		DATA	14#
HDTSA	FE20H	ADDR			DATA	15# 26 27
MAIN	0H	ADDR		PUB	CODE	11@ 19
SAMPM			MOD			2#
START	0H	R	ADDR	PUB	?CSEG	11@ 19 22#
STASC	FE21H	ADDR			DATA	16# 31

(8) 汇编列表文件格式指定 (-LW, -LL, -LH, -LT, -LF/-NLF)**(a) -LW**

语法: -LW [字符数]
 默认预设: -LW132 (显示输出时 80 个字符)

[功能]

选项 -LW 更改可在汇编列表文件中第 1 行上打印的字符数。

[应用]

指定选项 -LW 以更改可在汇编列表文件中第 1 行上打印的字符数。

[说明]

- 1) 可使用选项 -LW 指定的字符数量范围显示如下。
 (显示输出时最多 80 个字符)

$$72 \leq 1 \text{ 行打印的字符数} \leq 2046$$

如果指定的数值超出该范围，或指定的不是数值，则会发生终止错误。

- 2) 如果省略字符数，则会指定 132。
 然而，在汇编列表文件输出显示时，将会指定 80。
- 3) 所指定的字符数不包括终结符 (CR, LF)。
- 4) 如果指定了选项 -NP，则选项 -LW 不可用。

[注意]

可在源模块开始处介绍与 -LW 选项具有相等功能的控制指令。
 概要如下：

$\nabla \$ \nabla \text{WIDTH}$

有关控制指令的信息，请参阅语言部分的第 4 章控制指令。

[使用示例]

指定 80 为汇编列表文件中每行的字符数。

```
C>ra78k0s -c9024 k0smain.asm -lw80
```

此引用汇编列表。

Assemble list

```
ALNO  STNO  ADRS  OBJECT  M I SOURCE STATEMENT
1      1
2      2          NAME  SAMPM
3      3          ;*****
4      4          ;*                                     *
5      5          ;*  HEX -> ASCII Conversion Program *
6      6          ;*                                     *
7      7          ;*      main-routine *
8      8          ;*                                     *
9      9          ;*****
10     10
11     11          PUBLIC MAIN,START
12     12          EXTRN  CONVAH
13     13          EXTRN  _@STBEG
14     14 ----          DATA  DSEG  AT 0FE20H
15     15 FE20          HDTSA:  DS    1
16     16 FE21          STASC:  DS    2
17     17
18     18 ----          CODE   CSEG  AT 0H
19     19 0000 R0000    MAIN:   DW    START
20     20
21     21 ----          CSEG
22     22 0000          START:
23     23
24     24
25     25
      :
      :
```

(b) -LL

语法: -LL [行数]
默认预设: -LL66 (显示输出时无页面分页符)

[功能]

选项 -LL 更改可在汇编列表文件中 1 页上打印的行数。

[应用]

指定选项 -LL 以更改可在汇编列表文件中 1 页上打印的行数。

[说明]

- 1) 可使用选项 -LL 指定的行数范围显示如下。

$$20 \leq 1 \text{ 页打印的行数} \leq 32767$$

如果指定的数值超出该范围，或指定的不是数值，则会发生终止错误。

- 2) 如果省略行数，则会指定 66。
- 3) 如果指定的行数为 0，则不会出现任何分页符。
- 4) 如果指定了选项 -NP，则选项 -LL 不可用。

[注意]

可在源模块开始处介绍与 -LL 选项具有相等功能的控制指令。

概要如下：

∇ \$VLENGTH

有关控制指令的信息，请参阅语言部分的第 4 章控制指令。

[使用示例]

指定 20 为汇编列表文件中每页的行数。

```
C>ra78k0s -c9024 k0smain.asm -ll20
```

此引用 KOSMAIN.PRN。

78K/OS Series Assembler Vx.xx

Date:xx xxx xxxx Page: 1

Command:-c9024 k0smain.asm -ll20

Para-file:

In-file:KOSMAIN.ASM

Obj-file:KOSMAIN.REL

Prn-file:KOSMAIN.PRN

Assemble list

78K/OS Series Assembler Vx.xx

Date:xx xxx xxxx Page: 2

ALNO STNO ADRS OBJECT M I SOURCE STATEMENT

```

1      1
2      2          NAME  SAMPM
3      3          ;*****
4      4          ;*
5      5          ;*  HEX -> ASCII Conversion Program
6      6          ;*
7      7          ;*  main-routine

```

78K/OS Series Assembler Vx.xx

Date:xx xxx xxxx Page:

3

ALNO STNO ADRS OBJECT M I SOURCE STATEMENT

```

8      8          ;*
9      9          ;*****
10     10          PUBLIC MAIN, START
11     11
      :
      :

```

(c) -LH

语法: -LH 字符串
默认预设: 空

[功能]

选项 -LH 指定汇编列表文件头标题栏中打印的字符串。

[应用]

- 1) 指定选项 -LH 以显示简要说明汇编列表文件内容的标题。
- 2) 通过在每页上打印标题，一眼就可明白汇编列表文件的内容。

[说明]

- 1) 最多可在标题中指定 60 个字符。字符串不可包含空格。
- 2) 如果写入的字符多于 61 个，则会识别头 60 个字符且不会输出错误消息。
1 个日文汉字或平假名记作 2 个字符。
如果每行的最大字符数少于或等于 119，则有效的字符串长度变化如下。

$$\text{有效长度} = (\text{每行的最大字符数}) - 60$$

- 3) 如果没有指定字符串的长度，则会发生终止错误。
- 4) 如果指定了选项 -NP，则选项 -LH 不可用。
- 5) 如果省略了 -LH 选项，则汇编列表文件的标题栏将会空白。
- 6) 可写入标题栏的字符设置如下。

表 5-4. 可作为标题写入的字符

字符	命令行	参数文件
*?><	如果附上 "", 则可写入。	可写入。 即使附上 "", 同命令行所释。
;	如果附上 "", 则可写入。	不可写入。 (假设为注释。)
#	可写入。	不可写入。 (假设为注释。)
" (双引号)	不可作为一个有效的字符写入。	不可作为一个有效的字符写入。
00H	不可写入。	可写入。 然而, 它可解释为字符串的结尾。
03H, 06H, 08H, 0DH, 0EH, 10H, 15H, 17H, 18H, 1BH, 7FH	不可写入。	可写入。 然而, 这些字符将会在汇编列表文件中 作为 '!' 出现。(单个 0DH 将不会输入 到表中。)
01H, 02H, 04H, 05H, 07H, 0BH, 0CH, 0FH, 11H, 12H, 13H, 14H, 16H, 19H, 1CH, 1DH, 1EH, 1FH	可写入。 然而, 这些字符将会在汇编列表文件中 作为 '!' 出现。	可写入。 然而, 这些字符将会在汇编列表文件中 作为 '!' 出现。
1AH	可写入。 然而, 该字符将会在汇编列表文件中作 为 '!' 出现。	不可写入。 (文件的结尾)
字母字符	大写或小写字符按原样输入。	大写或小写字符按原样输入。
其他	可写入。	可写入。

备注 如果命令行上的星号(*)不是通配符扩展的目标, 即使没有用""引上, 则可将其写入。

[注意]

还可在源模块开始处写入与选项 -LL 具有相同功能的控制指令。

<p> ∇\$$\nabla$TITLE$\nabla$($\nabla$'character-string'$\nabla$) ∇\$$\nablaTT\nabla$($\nabla$'character-string'$\nabla$) ; abbreviated form </p>

有关控制指令的信息, 请参阅语言部分的第4章控制指令。

[使用示例]

打印汇编列表文件头的标题。

```
C>ra78k0s -c9024 k0smain.asm -lhRA78K0S_MAINROUTINE
```

此引用 K0SMMAIN.PRN。

```
78K/0S Series Assembler Vx.xx RA78K0S_MAINROUTINE Date:xx xxx xxxx Page: 1
```

标题

Command: -c9024 k0smain.asm -lhRA78K0S_MAINROUTINE

Para-file:

In-file:K0SMMAIN.ASM

Obj-file:K0SMMAIN.REL

Prn-file:K0SMMAIN.PRN

Assemble list

ALNO STNO ADRS OBJECT M I SOURCE STATEMENT

```
1      1
2      2          NAME  SAMPM
3      3          ;*****
4      4          ;*
5      5          ;*  HEX -> ASCII Conversion Program
6      6          ;*
7      7          ;*  main-routine
      .
      .
```

(d) -LT

语法: -LT [字符数]
默认预设: -LT8

[功能]

通过为任何类型的列表指定字符数，选项 -LT 为列表执行制表处理以为源模块中 HT（纵向制表）代码替代和输出大量的空格。

[应用]

在使用 -LW 选项为任何列表指定每行最少数量的字符时，指定选项 -LT 以插入 tab 制表符取代一系列的空格，因此节省了所使用的字符数量。

[说明]

- 1) 可使用选项 -LT 指定的字符数量范围显示如下。

$$0 \leq \text{可指定的字符数量} \leq 8$$

如果指定的数值超出该范围，或指定的不是数值，则会发生终止错误。

- 2) 如果指定了 -LT0，将不会执行制表处理，且会输出制表代码。
- 3) 如果指定了选项 -NP，则选项 -LT 不可用。

[注意]

可在源模块开始处介绍与 -LT 选项具有相等功能的控制指令。
概要如下：

∇ \$ ∇ TAB ∇ tab-count

有关控制指令的信息，请参阅语言部分的第 4 章控制指令。

[示例 1]

在省略选项 -LT 时引用 Sample.prn。

Assemble list

ALNO	STNO	ADRS	OBJECT	M I	SOURCE	STATEMENT
1	1				NAME	SAMPLE
2	2					
3	3	----			CODE	CSEG
4	4	0000	0A27		MOV	A, B
5	5	0002	0A12		SET1	A, 1
6	6				END	

[示例 2]

使用 HT 代码指定 1 个表格。

C>ra78k0s -c9024 sample.asm -lt1

此引用 sample.prn。

Assemble list

ALNO	STNO	ADRS	OBJECT	M I	SOURCE	STATEMENT
1	1				NAME	SAMPLE
2	2					
3	3	----			CODE	CSEG
4	4	0000	0A27		MOV	A, B
5	5	0002	0A12		SET1	A, 1
6	6				END	

备注 由 HT 代码输入的表格数为 1。

(e) -LF/-NLF

语法: -LF
 :
默认预设: -NLF

[功能]

- 1) 选项 -LF 在汇编列表文件结尾处插入一个换页 (FF) 代码。
- 2) -NLF 选项会使 -LF 选项不可用。

[应用]

如果在打印汇编列表文件的内容之后您想要添加分页符，指定选项 -LF 以添加换页代码。

[说明]

- 1) 如果指定了选项 -NP，则选项 -LF 不可用。
- 2) 如果同时指定了选项 -LF 和选项 -NLF，则后指定的选项优先。

[注意]

可在源模块开始处介绍与 -LF/-NLF 选项具有相等功能的控制指令。
概要如下：

<pre>▽\$▽FORMFEED ▽\$▽NOFOMFEED</pre>

有关控制指令的信息，请参阅语言部分的第 4 章控制指令。

[使用示例]

在汇编列表文件结尾处添加一个换页代码。

```
C>ra78k0s -c9024 k0smain.asm -pprn -lf
```

(9) 错误列表文件输出指定 (-E/-NE)

语法: -E [输出文件名]
 : -NE
 默认预设: -NE

[功能]

- 1) 选项 -E 输出错误列表文件，并指定错误列表文件的输出目的地和输出文件名。
- 2) -NE 选项会使 -E 选项不可用。

[应用]

- 1) 指定选项 -E 将错误消息保存到文件。
- 2) 指定选项 -E 更改错误列表文件的输出目的地和输出文件名。

[说明]

- 1) 错误列表文件可另存为磁盘类型的文件或设备类型文件。然而，如果指定了设备类型文件名 CLOCK，则会发生终止错误。
- 2) 在指定选项 -E 和省略输出文件名时，错误列表文件名将会为 'input file name.ERA'。
- 3) 在指定选项 -E 和省略驱动器名时，错误列表文件名将会输出到当前驱动器。
- 4) 如果同时指定了选项 -E 和选项 -NE，则后指定的选项优先。

[使用示例]

创建一个错误列表文件 (sample.era)。

```
C>ra78k0s -c9024 k0smain.asm -esample.era
```

```
78K/0S Series Assembler Vx.xx [xx xxx xx]  

  Copyright (C) NEC Electronics Corporation xxxx
```

```
PASS_PARSE Start  

K0SMMAIN.ASM(26) :F202 Illegal operand  

PASS_OUTOBJ Start  

K0SMMAIN.ASM(32) :F407 Undefined symbol reference 'F'  

K0SMMAIN.ASM(41) :F407 Undefined symbol reference 'F'
```

```
Target chip:uPD78xxxx  

Device file:Vx.xx
```

```
Assembly complete, 3 error(s) and 0 warning(s) found.
```

引用该错误列表文件 (sample.era)。

```
PASS_PARSE Start  

K0SMMAIN.ASM(26) :F202 Illegal operand  

PASS_OUTOBJ Start  

K0SMMAIN.ASM(32) :F407 Undefined symbol reference 'F'  

K0SMMAIN.ASM(41) :F407 Undefined symbol reference 'F'  

Pass1 Start
```

(10) 参数文件指定 (-F)

语法: -F 文件名
默认预设: 仅可在命令行输入该选项和输入的文件名。

[功能]

选项 -F 从指定的文件输入汇编器选项和输入的文件名。

[应用]

- 1) 当启动汇编器所需的数据不适合命令行时指定选项 -F。
- 2) 指定选项 -F 以在每次执行汇编时重复指定同一选项并将这些选项保存到参数文件中。

[说明]

- 1) 仅可将磁盘类型的文件名指定为“文件名”。如果指定了设备类型文件名，则会发生终止错误。
- 2) 如果省略了文件名，则会发生终止错误。
- 3) 不允许嵌套参数文件。如果在参数文件内指定了选项 -F，则会发生终止错误。
- 4) 可写入参数文件内的字符数没有限制。
- 5) 使用空格、制表符或回车 [↵] 分隔选项活文件名。
- 6) 将会在为命令行上的参数文件指定的位置扩展参数文件内的参数和输入的文件名。
- 7) 后指定的扩展选项优先。
- 8) ';' 或 '#' 后以及 [↵] 或 'EOF' 前所输入的所有字符将会解释为注释。
- 9) 如果两次或多次指定了选项 -F，则会发生终止错误。

[使用示例]

使用参数文件执行汇编。

设置参数文件 (K0SMMAIN.PRA) 的内容如下。

```
;parameter file  
k0smain.asm -osample.rel -g -c9024  
-psample.prn
```

在命令行中输入以下内容。

```
C>>ra78k0s -fk0smain.pra
```

(11) 临时文件创建路径指定 (-T)

语法: -T 路径名
默认预设: 在由环境变量 **TMP** 所指定的路径中创建一个临时文件。
 未指定路径时, 则临时文件将会创建在当前路径内。

[功能]

选项 **-T** 指定创建临时文件所在的路径。

[应用]

使用选项 **-T** 指定创建临时文件的位置。

[说明]

- 1) 仅可将路径指定为路径名。
- 2) 不可省略路径名。
- 3) 即使之前所创建的临时文件已存在, 如果该文件未受到保护, 则会将其覆盖重写。
- 4) 只要所需的内存大小可用, 则将会在内存中扩展临时文件。如果没有足够可用的内存, 则临时文件的内容将会被写入磁盘。
稍后可通过所保存的磁盘文件访问这些临时文件。
- 5) 汇编结束时将会删除临时文件。在按 (**CTRL-C**) 终止汇编时, 也可删除这些临时文件。
- 6) 可根据如下顺序确定要创建的临时文件所在的路径。
 - a. 由选项 **-T** 所指定的路径
 - b. 由环境变量 **TMP** (省略选项 **-T** 时) 所指定的路径
 - c. 当前路径 (未设置 **TMP** 时)在指定了 a. 或 b. 时, 如果无法在指定的路径创建临时文件, 则会发生终止错误。

[使用示例]

指定临时文件输入到目录 **A:\TMP**。

```
C>ra78k0s -c9024 k0smain.asm -ta:\tmp
```

(12) 汉字码指定 (-ZS/-ZE/-ZN)

语法: -ZS
 -ZE
 -ZN

默认预设: 根据不同的操作系统, 解释如下。
 -ZS (Windows/HP-UX)
 -ZE (SunOS/Solaris)

[功能]

- 1) 注释中所描述的汉字解释为指定的汉字码。
- 2) 根据选项, 汉字码解释如下。
 - ZS: 转换 JIS 代码
 - ZE: EUC 代码
 - ZN: 不能解释为汉字。

[应用]

这些选项用于指定注释行汉字的汉字码的解释。

[说明]

- 1) 如果同时指定了选项 -ZS, 选项 -ZE 和选项 -ZN, 则最后指定的选项优先。
- 2) 可在源模块开始处介绍与 -ZS, -ZE 和 ZN 选项具有相等功能的控制指令。Syntax 如下所示。

▲\$▲KANJICODE△SJIS ▲\$▲KANJICODE△EUC ▲\$▲KANJICODE△NONE

有关控制指令的详细信息, 请参阅语言手册的**第 4 章控制指令**。

- 3) 还可通过使用环境变量 LANG78K 指定汉字码。有关环境变量的详细信息, 请参阅 **11.2 准备开发环境 (环境变量)**。

[使用示例]

汉字码解释为 EUC 代码

```
A>ra78k0s k0smain.asm -c9024 -ze
```

(13) 设备文件搜索路径指定 (-Y)

语法: -Y 路径名
默认预设: 将会按以下顺序确定的路径读取设备文件。
 1) <..\dev> (适用于 ra78k0s.exe 启动路径)
 2) 启动 RA78K0S 的路径
 3) 当前目录
 4) 环境变量 PATH

[功能]

从指定的路径读取设备文件。

[应用]

指定设备文件所在的路径。

[说明]

- 1) 如果在指定选项 -Y 后未指定任何路径名, 则会发生终止错误。
- 2) 如果在指定选项 -Y 后省略了路径名, 则会发生终止错误。
- 3) 按如下所确定的顺序从其读取设备文件的路径。
 - a. 由选项 -Y 所指定的路径
 - b. <..\dev> (适用于 ra78k0s.exe 启动路径)
 - c. 启动 RA78K0S 的路径
 - d. 当前目录
 - e. 环境变量 PATH

[使用示例]

将设备文件的路径指定为目录 a:\78k0s\dev。

```
C>ra78k0s k0smain.asm -c9024 -ya:\78k0s\dev
```

(14) 符号定义指定 (-D)

语法: -D 符号名 [=数值][, 符号名 [=数值]...]
 默认预设: 空

[功能]

定义符号。

[应用]

在想要定义符号时，指定选项 -D。

[说明]

- 1) 分配给符号的数值必须是二进制、八进制、十进制或十六进制的数字。如果省略了数值的规格，则预设已指定了 1。
- 2) 最多可指定 30 个字符，使用逗号分隔。
- 3) 最多可指定 30 个字符作为符号名。
- 4) 如果不止一次指定相同名称，最后指定的名称有效。
- 5) 符号名中的字母区分大小写。
- 6) 使用选项 -D 所定义的符号用于取代 EQU/\$SET/\$RESET。如果在源文件中还定义了为选项 -D 指定的符号名，则会发生错误。

[使用示例]

指定 2 为符号定义。

```
C>ra78k0s k0smain.asm -c9024 -dSYM=2
```

(15) 帮助指定 (--)

语法: --
 默认预设: 无显示

[功能]

选项 - 显示帮助消息。

[应用]

帮助消息是汇编选项的说明列表。执行汇编器时，请参阅这帮助消息。

[说明]

- 1) 在指定选项 - 时，所有其他的汇编器选项不可用。
- 2) 要读取帮助消息的下一部分，请按“返回”键。
 要退出帮助显示，请按除“返回”键之外的任何键，然后按“返回”键。

注意事项 该选项无法在 PM + 上指定。

要参阅 PM+帮助，单击 <Assembler Options> 对话框中的“帮助”按钮。

【使用示例】

指定选项 - 时，帮助消息会输出在显示屏上。

78K/0S Series Assembler Vx.xx [xx xx xxxx]

Copyright (C) NEC Electronics Corporation xxxx,xxxx

usage : ra78k0s [option[...]] input-file [option[...]]

The option is as follows ([] means omissible).

- cx :Select target chip. (x = 9012,p9014 etc.) *Must be specified.
- o[file]/-no :Create the object module file [with the specified name] / Not.
- e[file]/-ne :Create the error list file [with the specified name] / Not.
- p[file]/-np :Create the print file [with the specified name] / Not.
- ka/-nka :Output the assemble list to print file / Not.
- ks/-nks :Output the symbol table list to print file / Not.
- kx/-nkx :Output the cross reference list to print file / Not.
- lw[width] :Specify print file columns per line.
- ll[length] :Specify print file lines per page.
- lf/-nlf :Add Form Feed at end of print file / Not.
- lt[n] :Expand TAB character for print file(n=1 to 8)/ Not expand(n=0).
- lhstring :Print list header with the specified string.
- g/-ng :Output debug information to object file / Not.
- j/-nj :Create object file if fatal error occurred / Not.
- idirectory[,directory..] :Set include search path.
- tdirectory :Set temporary directory.
- ydirectory :Set device file search path.
- ffile :Input option or source module file name from specified file.
- ga/-nga :Output assembler source debug information to object file / Not.
- dname[=data][,name[=data][...]] :Define name [with data].
- zs/-ze/-zn :Change source regulation.
 - zs:SJIS code usable in comment.
 - ze:EUC code usable in comment.
 - zn:no multibyte code in comment.
- :Show this message.

DEFAULT ASSIGNMENT:

-o -ne -p -ka -nks -nkx -lw132 -ll66 -nlf -lt8 -g -nj -ga

5.5 在 PM+ 中设置选项

本节介绍从 PM+ 设置汇编器选项的方法。

5.5.1 设置选项

从 PM+ 菜单的 [Tools] 选择 [Assembler Options...] 或单击  显示 <Assembler Options> 对话框。通过在该对话框中输入所需的选项可设置汇编器的选项。

图 5-2. <Assembler Options> 对话框（选择 <<Output1>> 选项卡时）

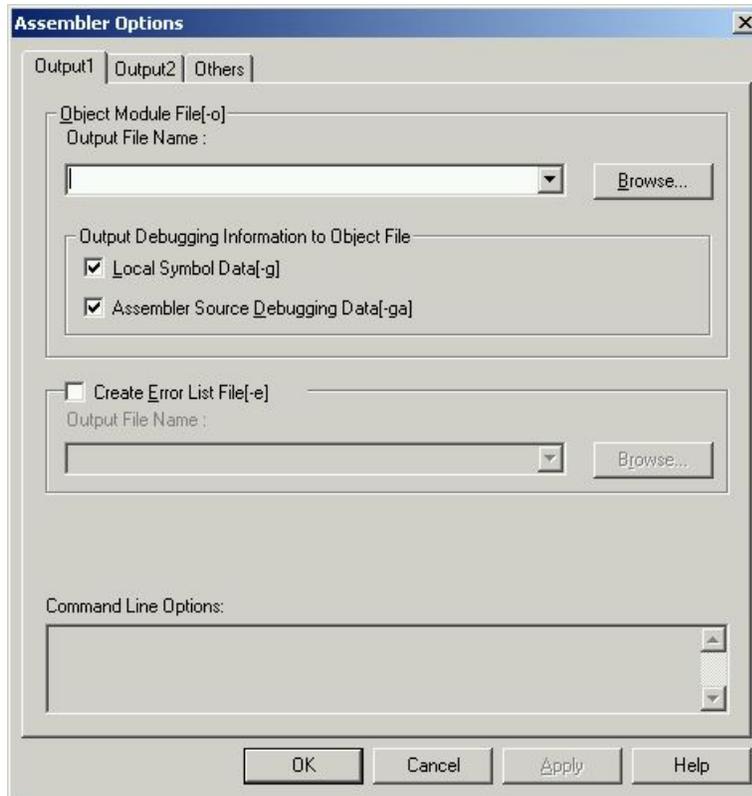


图 5-3. <Assembler Options> 对话框 (选择 <<Output2>> 选项卡时)

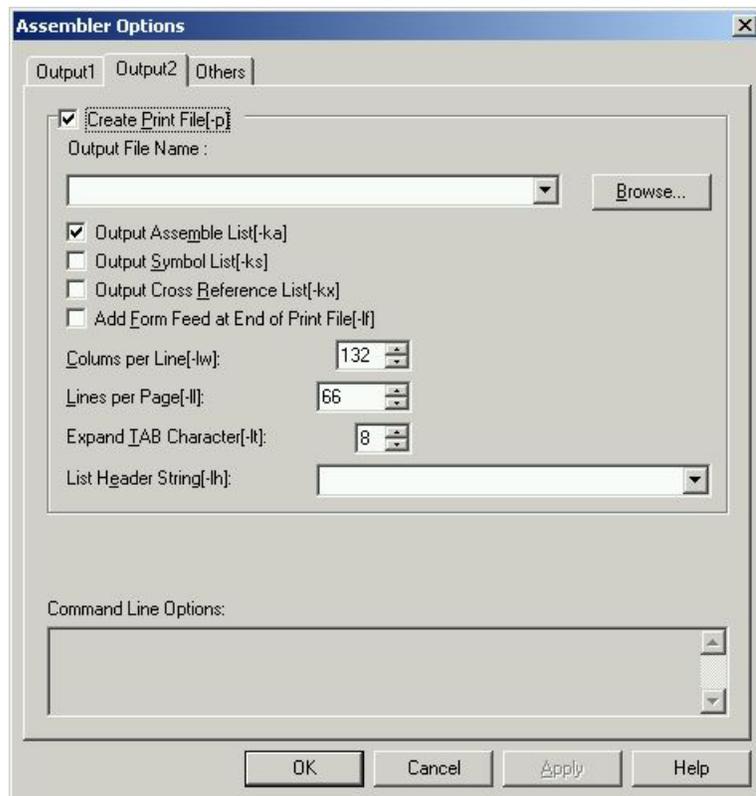
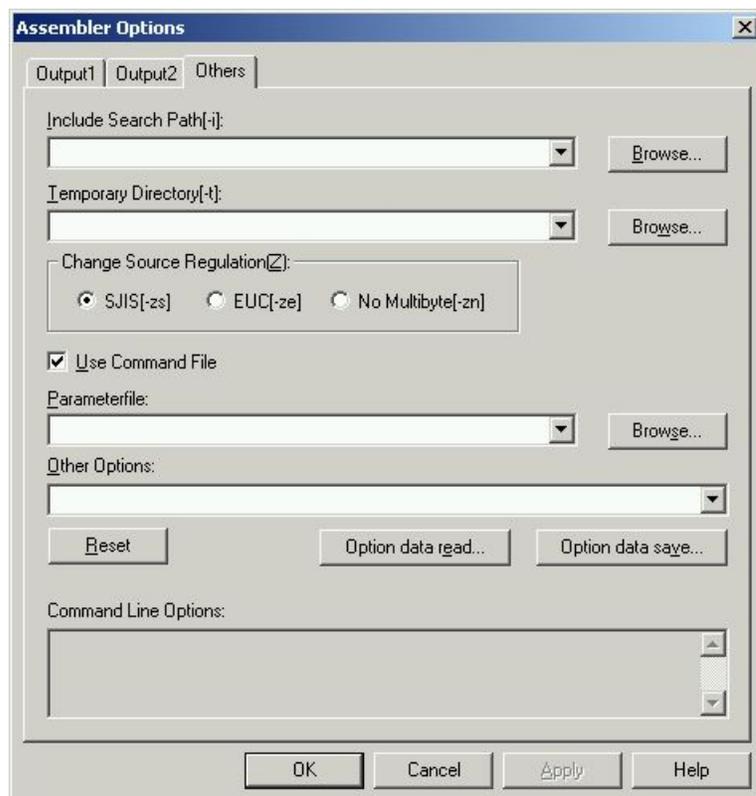


图 5-4. <Assembler Options> 对话框 (选择 <<Others>> 选项卡时)



5.5.2 选项

<Assembler Options> 对话框中的各个选项介绍如下。

- **Object Module File [-o]**
Output File Name:
通过使用 [Browse...] 按钮或直接输入路径，指定目标文件的输出路径。
- **Local Symbol Data [-g]**
执行包括局部符号的符号调试。
- **Assembler Source Debugging Data [-ga]**
添加汇编器源调试数据。
- **Create Error List File [-e]**
Output File Name:
要输出“错误列表”文件，在输入框中输入文件名。
要指定路径，使用 [Browse...] 按钮。
- **Create Print File [-p]**
Output File Name:
通过使用 [Browse...] 按钮或直接输入文件名，指定目标模块汇编列表文件的输出路径。
- **Output Assemble List [-ka]**
在汇编列表文件中输出汇编表。
- **Output Symbol List [-ks]**
在汇编表后输出符号。
- **Output Cross Reference List [-kx]**
在汇编表之后输出交叉引用表。
- **Add Form Feed at End of Print File [-lf]**
在打印汇编列表文件的内容后添加换页代码。
- **Columns per Line [-lw]**
指定汇编列表文件中每行的字符数（从 72 到 2,046 字符数中选择）。
- **Lines per page [-ll]**
指定汇编列表文件中每页的行数（从 20 到 32,767 字符数中选择）。
- **Expand TAB character [-lt]**
指定制表符字符的长度（从 0 到 8 个字符选择）
- **List Header String [-lh]**
指定汇编列表文件头的标题框中要打印的字符串（最多 60 个字符）。
- **Include Search Path [-i]**
通过使用 [Browse...] 按钮或直接输入路径，指定读取包含文件的输出路径。
- **Temporary Directory [-t]**
通过使用 [Browse...] 按钮或直接输入路径，指定要创建的临时文件的位置。
- **Change Source Regulation [Z]**
选择要在源注释中使用的多字节代码类型 (SJIS[-zs], EUC[-ze], No Multibyte[-zn])。
- **Use Command File**
选择该复选框创建指令文件。
- **Parameterfile**
通过使用 [Browse...] 按钮或直接输入参数文件，指定所选的用户自定义的参数文件。

- **Other Options**

如果想要指定除可使用复选框或单选按钮选择选项之外的选项，请在该输入框内输入该选项。

- **Reset**

重置输入的内容。

- **Option data read...**

打开 <Option Data Read> 对话框，并在已指定选项数据文件之后，读取该文件。

- **Option data save ...**

打开 <Option Data Read> 对话框，并输入文件名将选项数据保存到选项数据文件。

- **Command Line Options**

该编辑框只读。显示当前设置选项字符串。

第 6 章 连接器

连接器输入一些由 78K0S 系列汇编器输出的目标文件，确定位置地址并将这些文件输出为单个载入模块文件。

连接器还输出列表文件，如连接列表文件、错误列表文件。

如果出现连接错误，则会输出错误信息到错误列表文件中对出错原因进行说明。在出现错误时，不会输出载入模块文件。

6.1 连接器的 I/O 文件

连接器的 I/O 文件如下所示。

表 6-1 连接器的 I/O 文件

类型	文件名	说明	默认文件类型
输入文件	目标文件	<ul style="list-style-type: none">• 这些二进制文件包含机器语言数据的重定位和符号数据及机器语言数据的存储地址。• 这些文件由汇编器输出。	.REL
	库文件	<ul style="list-style-type: none">• 这些文件中包含两个（或更多）目标文件• 这些文件由库管理程序输出。	.LIB
	指令文件	<ul style="list-style-type: none">• 这些文件包含连接中用到的连接命令。• 这些文件由用户创建。	.DR
	参数文件	<ul style="list-style-type: none">• 这些文件包含程序执行的参数。• 这些文件由用户创建。	.PLK
输出文件	载入模块文件	<ul style="list-style-type: none">• 这些二进制映像文件包含了在连接中创建的所有数据 这些文件输入到对象转换器中。	.LMF
	连接列表文件	<ul style="list-style-type: none">• 为显示连接结果的编目文件。	.MAP
	错误列表文件	<ul style="list-style-type: none">• 这些文件包含在连接过程中生成的出错数据。	.ELK
I/O 文件	临时文件	<ul style="list-style-type: none">• 这些文件由连接器自动生成，在连接中使用。汇编完成时删除。	LKxxxxx.\$n (n = 1 至 3)

6.2 连接器的功能

连接器的功能如下。

(1) 连接输入程序段

连接器确定并控制各程序段的存储地址。

连接器识别相同的程序段，将其连接成单个程序段，哪怕其位于独立的目标模块文件中也是一样。

(2) 确定输入模块

当指定要输入的库文件时，输入目标模块文件引用的模块从库中取出，并作为输入模块进行处理。

(3) 确定输入程序段的存储地址

连接器确定输入模块各程序段的存储地址。如果源模块文件中指定了程序段的位置属性，则按照这些属性存放程序段。连接器还可以在其连接指令文件中指定位置属性。

(4) 校正目标码

当存储地址隐藏在目标码中时，连接器能根据上述(3)确定的存储地址校正目标码。

6.3 存储空间和存储区

存储空间是用来定义存储区的空间。存储区是在存储器中定义用于存放程序段的区域。

注意事项 无法指定除/REGULAR 空间之外的空间，因为只有一个存储空间可用。

存储空间： 各 64 KB
 存储区： 每个存储空间分为几个存储区。
 存储区对安装的存储器的存储地址进行声明。

表 6-2 程序段分配组（外部 ROM 等）

存储区名	默认地址	默认分配的程序段
ROM	内部 ROM：若未安装 ROM 则到 RAM 的开头为止	CSEG
RAM	内部 RAM	DSEG, BSEG

备注

1. 用指令文件来更改存储区的默认地址，或者用来指定写入程序各程序段的位置。
2. 具体例子请参见“图 3-5 连接指令文件的内容（sample.dr）”。

6.4 连接指令

连接指令（以下称为“指令”）是一组指令，用来在连接过程中发出不同的命令，如文件输入、可用存储区及分配程序段。

指令文件的作用是：

- (1) 在已安装存储器中声明地址。
- (2) 将存储器分成两个（或更多）区。

示例

- CALLT 区
- 内部 ROM
- 外部 ROM
- SADDR 区
- 除 SADDR 区外的内部 RAM

- (3) 由连接器指定程序段的位置。

指定各程序段的下列事项。

- 绝对地址
- 仅指定存储地址

用编辑器来创建指令文件（指定指令的文件）。在连接器启动时，指定-D 选项来读入创建的文件。

连接器从该文件读取指令并翻译来进行连接。

如下所示，可以使用两类指令。

表 6-3 指令类型

编号	指令类型	说明
1	存储器指令	<ul style="list-style-type: none"> • 在已安装存储器中声明地址 • 将存储器分为两个（或更多）区并指定存储区
2	程序段位置指令	<ul style="list-style-type: none"> • 指定程序段的位置

6.4.1 指令文件

指令文件中的指定指令的格式如下。

可在单个指令文件中指定一些指令。

1) 存储器指令

MEMORY 存储区名: (起始地址值, 长度) [/存储空间名]^注

注 存储空间名实际上不需要进行描述, 因为默认为/REGULAR。

2) 程序段位置指令

MERGE 程序段名: [AT(起始地址)]
[=存储区名说明] [/存储空间名]

(1) 保留字

以下为指令文件中的保留字。

MEMORY、MERGE、AT、SEQUENT、COMPLETE

在指令文件中保留字不能用于表达其它含义（程序段名、存储区名等）。

保留字可写成大写或小写字符, 但不能大小混写。

示例 MEMORY
 memory
 Memory: 不能用

当源文件中有两个（或更多）程序段同名时, 在指令中指定“COMPLETE”将保证产生出错提示且不会将程序段连成一体, 而指定“SEQUENT（默认）”将保证程序段的连接。

SEQUENT: 以程序段出现的次序进行合并, 不加空格。BSEG 以其出现的次序以 1 位为单位进行合并。

COMPLETE: 当两个（或更多）程序段同名时, 出现错误。

示例 MERGE DSEG1:COMPLETE=RAM

(2) 符号

在指定程序段名、存储区名和存储空间名时是区分大小写字符的。

(3) 数值

要想在指令中为各个项指定数值常量, 应以十进制或十六进制形式书写。

其方法与源程序相同; 在十六进制末尾加上“H”。如果开头出现 A 至 F, 则先放上“0”。

示例 23H, 0FC80H

(4) 注释

在指令文件中，当写下 ‘;’ 或 ‘#’ 时，从这里开始到回车（换行符）为止的所有字符都作为注释进行处理。如果指令文件在回车之前终结，则文件末尾之前的所有内容都作为注释处理。

示例 下划线部分为注释。

```
:DIRECTIVE FILE FOR 789024
MEMORY MEM1: (01000H, 1000H) #SECOND MEMORY AREA
```

6.4.2 存储器指令

存储器指令是定义存储区（已安装存储器的名称和地址）的指令。

已定义存储区的名称（存储区名）用来引用程序段位置指令。

最多可定义 100 个存储区，其中包括默认存储区。

[语法]

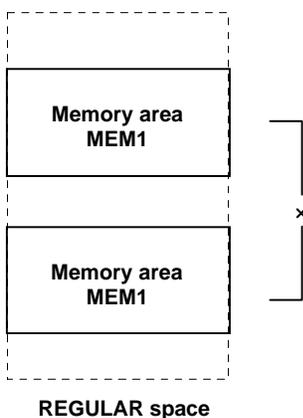
存储器Δ 存储区名∇:∇ (∇起始地址值∇, ∇长度∇) [/∇存储空间名]

(1) 存储区名

指定已定义存储区的名称。指定存储区名的有关内容如下。

- 1) 可用于存储区名的字符为 A 至 Z、a 至 z、0 至 9、_、?、@。但是，存储区名不能以 0 至 9 开头。
- 2) 大小写字符认为是不同的字符。
- 3) 大小写字符可混在一起使用。
- 4) 存储区名的最大长度为 31 个字符。若使用了 32 个（或更多）字符，则会导致错误。
- 5) 每个存储区名在整个存储空间中必须只出现在 1 个位置处。同一个存储区名不能用于不同的存储区，哪怕这些存储区位于不同的存储空间也不行。

图 6-1 存储区名



(2) 起始地址

指定要定义的存储区的起始地址。

使用从 0H 至 FFFFH 的数值。

(3) 长度

指定要定义的存储区的长度。

指定要求如下。

- 1) 使用 1 或大于 1 的数值。
- 2) 如果长度指定更改为连接器定义的默认存储区长度，则会应用对可定义范围的限制。

关于各设备定义的默认存储区长度及各设备的可重新定义范围，请见各设备文件的“使用注意（Considerations on Use）”。

(4) 存储空间名

存储空间名显示于 64 KB 空间 REGULAR 中。

对指定的要求如下。

- 1) 存储空间名必须用大写字符指定。
- 2) 当省略存储空间名时，假定指定为 REGULAR。
- 3) 如果在写了 '/' 后省略存储空间名，则会出现错误。

[功能]

- 1) 对存储区名指定的存储区定义指定的存储空间。
- 2) 用 1 条存储器指令可定义 1 个存储区。
- 3) 可不止一次指定存储器指令。但是，以指定次序进行的多次定义将导致错误。
- 4) 只要存储器指令中未重新定义相同的存储区，默认的存储区就是有效的。如果省略对存储器指令的指定，则只会指定各设备连接器带有的默认存储区。
- 5) 如果希望不使用默认存储区而使用其它存储区，则应将默认存储区名的长度指定为“0”。

[使用示例]

将存储空间中的 0H 至 1FFH 地址定义为 ROMA。

```
MEMORY ROMA: (0H, 200H)
```

6.4.3 程序段位置指令

程序段位置指令用来定位存储器指定区或特定地址的指定程序段。

[语法]

```
MERGEΔ程序段▽:▽ [AT▽ (▽起始地址▽)]
                [▽=▽存储区名] [▽/▽存储空间名]
```

(1) 程序段名

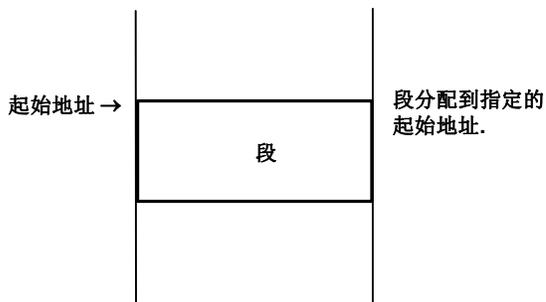
程序段名是输入连接器的目标文件中包含的程序段的名称。

- 1) 只有输入程序段可以用程序段名进行指定。
- 2) 程序段名指定方法必须与在汇编源程序中相同。

(2) 起始地址

起始地址将程序段分配到“起始地址”指定的区。

- 1) 保留字 AT 必须用全部大写或全部小写字符进行指定。不能混用大小写字符进行指定。
- 2) 起始地址指定数值常量。



- 注意事项**
1. 在将程序段放置到指定起始地址时，如果它超出了所在存储区的存储区范围，则会导致错误。
 2. 对于由程序段定义指令的 AT 指定或由 ORG 指令指定的存储地址的程序段来说，不能用连接指令来指定其起始地址。

(3) 存储空间名

存储空间名指定程序段放置的存储区。

- 1) 只有 REGULAR 可指定为存储区名。
- 2) 存储空间名必须用大写字母进行指定。
- 3) 当省略存储空间名时，假定指定为 REGULAR。

确定程序段放置的目标地点的情况如下。

表 6-4 存储区名指定及存储空间名共同确定的程序段位置

存储区名	存储空间名	程序段目标位置
×	×	REGULAR 空间中的默认存储区
存储区名	×	REGULAR 空间中的指定存储区

本表的重点是定义程序段放置的存储区。当实际存储地址确定后，如果指定了[AT (起始地址)]，则程序段分配到该地址起始的位置。

[注意]

- 1) 如果输入程序段没有程序段位置指令指定，则在汇编过程中会根据程序段定义指令指定的重定位属性对其存储地址进行指定。
- 2) 若已定义的程序段名不存在程序段，则会出现错误。
- 3) 若同一个程序段指定了多于一个程序段位置指令，则会出现错误。

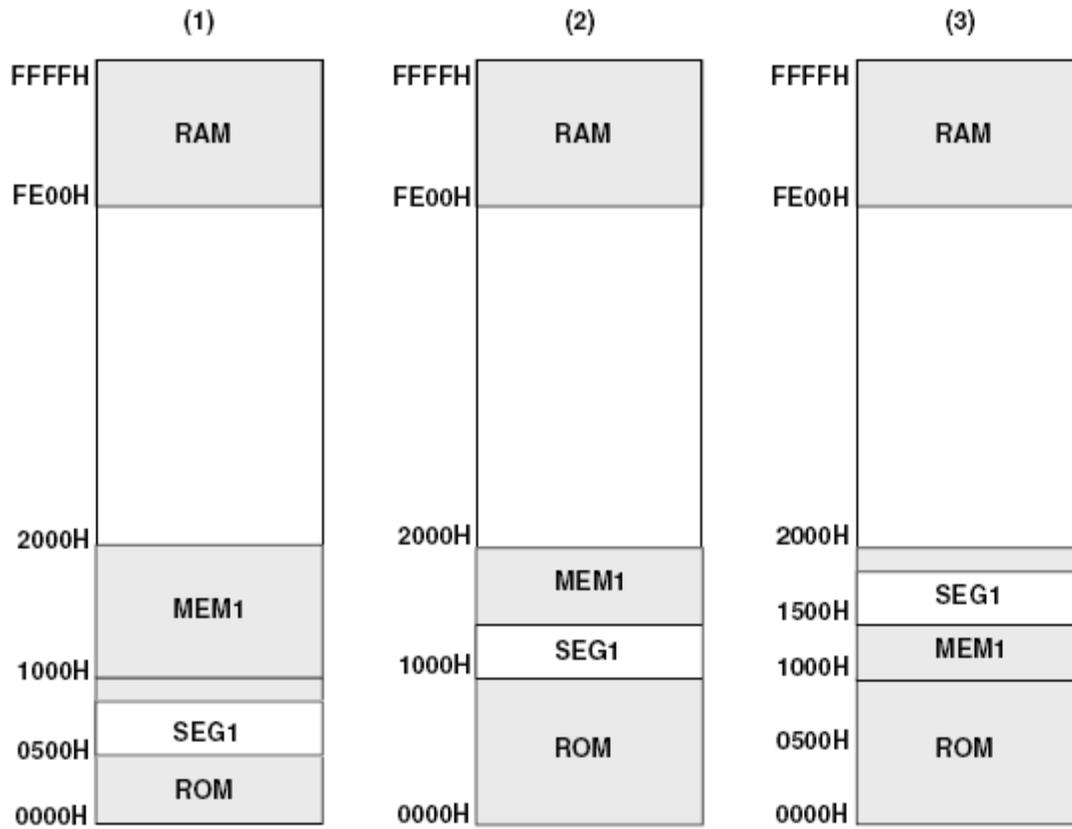
[使用示例]

给程序段 SEG1 分配地址，其段类型及重定位属性为‘CSEG UNIT’。在本例中，声明的存储区如下。

```
MEMORY ROM: (0000H, 1000H)
MEMORY MEM1: (1000H, 2000H)
MEMORY RAM: (0FE00H, 200H)
```

- (1) 若输入程序段 SEG1 分配到 ROM 区的 500H 处。
MERGE SEG1: AT (500H)
- (2) 若输入程序段 SEG1 分配到存储区 MEM1 中。
MERGE SEG1:= MEM1
- (3) 若输入程序段 SEG1 分配到存储区 MEM1 的 1500H 处。
MERGE SEG1: AT (1500H) = MEM1

图 6-2 程序段分配的具体例子



6.5 启动连接器

6.5.1 启动连接器

下面的两种方法可用来启动连接器。

(1) 从命令行启动

```
X>[路径名] lk78k0s [Δ选项] ... Δ目标模块文件名 [Δ选项]...[Δ]
```

(1)	(2)	(3)	(4)	(5)	(4)

- (1) 当前驱动器名
- (2) 当前目录名
- (3) 连接器命令文件名
- (4) 包含连接器动作的细节指令。

如果指定不止一个连接器选项，则应用空格将选项隔开。

- (5) 包含连接器动作的细节指令。
输入模块中最多可输入 256 项。

示例 C>lk78k0s k0smain.rel k0ssub.rel -ok0s.lmf -g

(2) 从参数文件启动

当启动连接器所需的数据不适合命令行时，或当每次进行连接都反复使用同样的连接器选项时，可使用参数文件。

要想从参数文件启动连接器，应在命令行中指定参数文件说明选项（-F）。

从参数文件启动连接器，如下。

```
X>LK78K0S [Δ目标模块文件] Δ-f  参数文件名
                |          |
                (1)       (2)
```

- (1) 参数文件说明选项
- (2) 包含启动连接器所需数据的文件

备注 用编辑器来创建参数文件。

书写参数文件内容的规则如下。

[[[Δ] 选项 [Δ选项] ... [Δ]Δ]] ...

- 1) 如果命令行中省略目标模块文件名，则应在参数文件中指定目标模块文件名。
- 2) 目标模块文件名也可以写在选项的后面。
- 3) 在参数文件中写入应在命令行中指定的所有连接器选项及输出文件名。

示例 用编辑器创建参数文件（K0S.PLK）。

参数文件 K0S.PLK 的内容：

```
;parameter file
k0smain.rel k0ssub.rel -ok0s.lmf -pk0s.map -e
-ta:\tmp
```

用参数文件 K0S.PLK 启动连接器。

```
C>lk78k0s -fk0s.plk
```

6.5.2 执行开始结束信息

(1) 执行开始信息

在连接器启动时，显示屏上出现执行开始信息。

```
78K/0S Series Linker Vx.xx [xx xxx xx]
Copyright (C) NEC Electronics Corporation xxxx, xxxx
```

(2) 执行结束信息

如果没有检测到由连接导致的连接错误，连接器就会将下列信息输出到显示屏上，并将控制权还给操作系统。

```
Target chip:uPD78xxx
Device file:Vx.xx
```

```
Link complete, 0 error(s) and 0 warning(s) found.
```

如果检测到由该连接导致的连接错误，连接器就会将错误号输出到显示屏上，并将控制权还给操作系统。

```
Target chip:uPD78xxx
Device file:Vx.xx
```

```
Link complete, 1 error(s) and 0 warning(s) found.
```

如果在连接期间连接器检测到致命错误，使其无法继续连接处理，则连接器会将一条信息输出到显示屏上，退出连接，将控制权还给操作系统。

示例 1 指定了不存在的目标文件。

```
C>lk78k0s samp1.rel samp2.rel

78K/0S Series Linker Vx.xx [xx xxx xx]
Copyright (C) NEC Electronics Corporation xxxx, xxxx

A006 File not found 'SAMP1.REL'
A006 File not found 'SAMP2.REL'
Program Aborted.
```

在上面的例子中指定了不存在的目标文件。导致错误，连接器异常中止连接。

示例 2 指定了不存在的连接器选项。

```
C>lk78k0s k0smain.rel k0ssub.rel -z
```

```
78K/0S Series Linker Vx.xx [xx xxx xx]
```

```
Copyright (C) NEC Electronics Corporation xxxx, xxxx
```

```
A018 Option is not recognized '-z'
```

```
Please enter 'LK78K0S --', if you want help messages.
```

```
Program Aborted.
```

在上面的例子中，指定了不存在的连接器选项。导致错误，连接器异常中止连接。

当显示报错信息且连接异常中止时，在“**第十二章 错误信息**”中查找原因，采取相应措施。

6.6 连接器选项

6.6.1 连接器选项的类型

连接器选项是关于连接器操作的细节指令。连接器选项分成 16 种类型。

表 6-5 连接器选项(1/2)

编号	分类	选项	说明
1	载入模块文件输出指定	-O	指定载入模块文件的输出。
		-NO	
2	强制的载入模块文件输出指定	-J	强制载入模块文件输出。
		-NJ	
3	调试数据输出指定	-G	将调试数据输出到载入模块文件中。
		-NG	
4	栈判定 (stack decision) 符生成指定	-S	自动生成栈判定的公共符号
		-NS	
5	指令文件指定	-D	输入指定文件作为指令文件。
6	连接列表文件输出指定	-P	指定输出连接列表文件。
		-NP	
7	连接列表文件数据指定	-KM	将映象表输出到连接列表文件中。
		-NKM	
		-KD	将连接指令文件输出到连接列表文件中。
		-NKD	
		-KP	将公共符号列表输出到连接列表文件中。
		-NKP	
		-KL	将局部符号列表输出到连接列表文件中。
-NKL			
8	连接列表文件格式指定	-LL	更改连接列表文件中 1 页能打印的行数。
		-LF	在编目文件的末尾插入换行代码。
		-NLF	

表 6-5 连接器选项 (2/2)

编号	分类	选项	说明
9	错误列表文件输出指定	-E	输出错误列表文件。
		-NE	
10	库文件指定	-B	输入指定文件作为库文件。
11	库文件读取路径指定	-I	从指定路径读取库文件。
12	参数文件指定	-F	从指定文件输入文件名和选项。
13	临时文件创建路径的指定	-T	在指定路径创建临时文件。
14	设备文件搜索路径指定	-Y	从指定路径读取设备文件。
15	报警信息输出指定	-W	指定是否将报警信息输出到控制台。
16	帮助指定	--	在显示屏上显示帮助信息。

备注 关于连接器选项的详情请参见“附件 C.3 连接器选项列表”。

6.6.2 连接器选项的先后顺序

表 6-6 说明了当同时指定两个连接器选项时哪个连接器选项优先。

表 6-6 连接器选项的先后顺序

	-NO	-NG	-NP	-NKM	-NKP	-NKL	--
-J	×						×
-G	×						×
-P				△	△	△	×
-KM			×				×
-KD			×	×			×
-KP		×	×				×
-KL		×	×				×
-LL			×				×
-LF			×				×

↑
垂直轴

←
水平轴

[用×标出的项]

当指定水平轴的选项时，垂直轴所示选项不可用。

示例 C>lk78k0s k0smain.rel k0ssub.rel -np -km -s

选项-KM 不可用。

[用△标出的项]

当水平轴的三个选项都被指定时，垂直轴所示选项不可用。

示例 C>lk78k0s k0smain.rel k0ssub.rel -p -nkm -nkp -nkl -s

选项 -NKM、-NKP、-NKL 被同时指定，所以选项 -P 不可用。

当同时指定选项及其‘N’对立选项（例如，-O 和-NO）时，二选项中只有最后指定的选项是可用的。

示例 C>lk78k0s k0smain.rel k0ssub.rel -o -no -s

选项-NO 在-O 之后指定，所以-O 不可用，-NO 可用。

未在“表 6-6 连接器选项的先后顺序”中指定的选项对其它选项没有特别的影响。但是，当指定帮助选项‘--’时，其它选项均变为不可用。

6.6.3 连接器选项说明

本节包含对各连接器选项的具体说明。

(1) 载入模块文件输出指定 (-O/-NO)

语法: -O [输出文件名]
 : -NO
默认预设: -O 输入文件名.lmf

[功能]

- 1) 选项-O 指定载入模块文件的输出。还指定输出的位置及文件名。
- 2) 选项-NO 指定不输出载入模块文件。

[应用]

- 1) 用选项-O 来指定载入模块文件的输出位置或更改其文件名。
- 2) 当仅为了输出连接列表文件而进行连接时，指定选项-NO。这将缩短连接时间。

[说明]

- 1) 可将磁盘型文件名及设备型文件名、NUL 和 AUX 指定为输出文件名。
- 2) 即使指定了选项-O，若出现致命错误，还是会无法输出载入模块文件。
- 3) 若在指定选项-O 时省略‘输出文件名’，则载入模块文件‘输入文件名.lmf’将输出到当前目录。
- 4) 只要‘输出文件名’中指定了路径名，‘输入文件名.lmf’就会输出到指定路径。
- 5) 如果同时指定选项-O 和-NO，则最后指定的选项优先。

[使用示例]

输出载入模块文件 k0s.lmf。

```
C>lk78k0s k0smain.rel k0ssub.rel -ok0s.lmf
```

(2) 强制的载入模块文件输出指定 (-J/-NJ)

语法: -J
 : -NJ
默认预设: -NJ

[功能]

- 1) 选项 -J 指定即便出现致命错误也要输出载入模块。
- 2) 选项 -NJ 使选项 -J 不可用。

[应用]

通常，在出现致命错误时无法输出载入模块文件。当希望收到出现致命错误的通知并执行程序时，应指定选项 -J 来输出载入模块文件。

[说明]

- 1) 当指定选项 -J 时，即便出现致命错误也会输出载入模块。
- 2) 若同时指定选项 -J 和 -NJ，则最后指定的选项优先。

[使用示例]

指定即便出现致命错误也要输出载入模块文件。

```
C>lk78k0s k0smain.rel k0ssub.rel -j
```

(3) 调试数据输出指定 (-G/-NG)

语法: -G
 : -NG
默认预设: -G

[功能]

- 1) 选项 -G 指定将调试数据（局部符号数据）添加到载入模块文件中。
- 2) 选项 -NG 使选项 -G 不可用。

[应用]

当进行源程序级的符号调试时，一定要使用选项 -G。

[说明]

- 1) 若指定了选项 -NO，选项 -G 不可用。
- 2) 若省略选项 -G，则无法添加调试数据。
- 3) 若同时指定选项 -G 和 -NG，则最后指定的选项优先。
- 4) 当指定选项 -NG 时，不管是否指定 -KP 或 -KL 都无法输出公共符号列表和局部符号列表。

[使用示例]

指定将调试数据添加到载入模块文件中。

```
C>lk78k0s k0smain.rel k0ssub.rel -g
```

(4) 栈判定符生成指定 (-S/-NS)

语法: -S [存储区名]
 : -NS
默认预设: -NS

[功能]

- 1) 选项 -S 生成栈判定公共符号 ‘_@STBEG’ 和 ‘_@STEND’。
- 2) 选项 -NS 使选项 -S 不可用。

[应用]

指定选项 -S 来保留栈区。

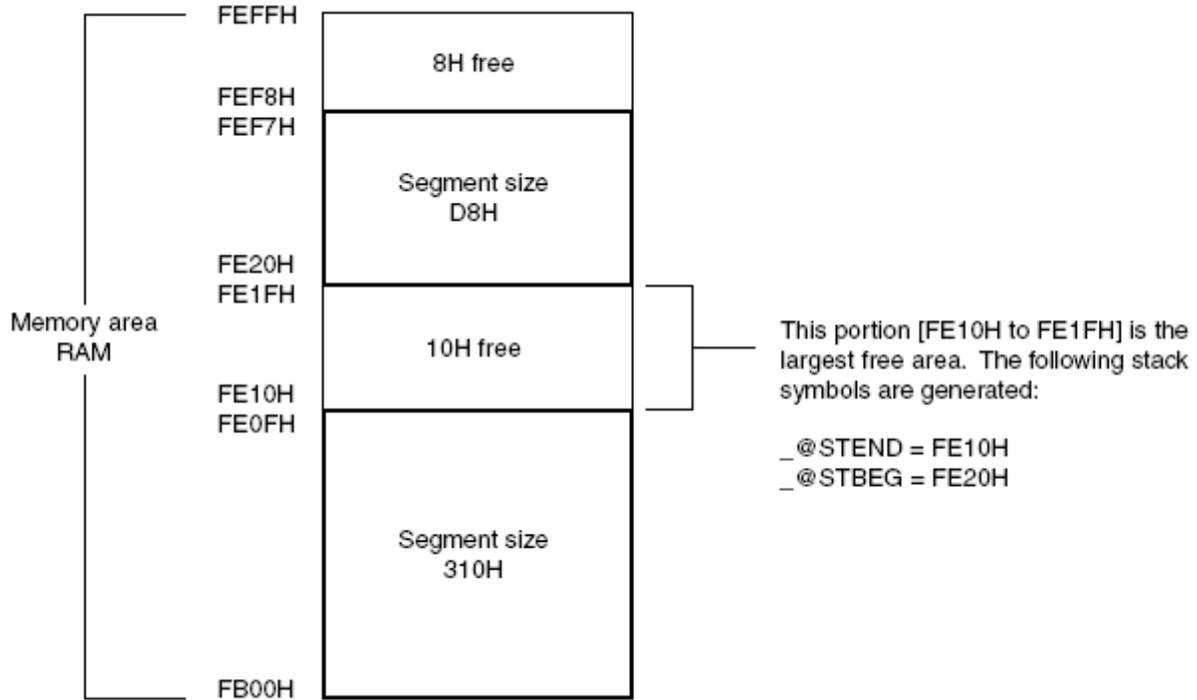
[说明]

- 1) ‘存储区名’ 是由用户定义的存储区名或指定的默认定义存储区名。
- 2) ‘存储区名’ 分大小写字符。
- 3) 连接器在选项 -S 指定的存储区中搜索未放置程序段的最大地址。然后连接器生成公共符号 ‘_@STEND’ 和公共符号 ‘_@STBEG’，前者的取值为最大地址区的首地址，后者取值为末尾地址+1。
这些符号作为公共声明的 NUMBER 属性符号处理，在连接器符号表末尾进行注册。当这些符号输出到连接列表文件时，模块名这列留空。
- 4) 如果最大开放区为 10 字节或更小，则输出报警信息。
- 5) 如果不存在闲置区，则输出报警信息，‘_@STEND’ 和 ‘_@STBEG’ 取值为最后地址+1。
- 6) 若省略‘存储区名’，则指定为‘RAM’。
- 7) 若同时指定选项 -S 和-NS，则最后指定的选项优先。

[使用示例]

在存储区 RAM 中保留栈区（但是，连接器将假定在 RAM 中输入了长度为 310H 的程序段，在 saddr 区中输入了长度为 D8H 的程序段）。

C>lk78k0s k0smain.rel k0ssub.rel -s



(5) 指令文件指定 (-D)

语法: -D 文件名
默认预设: 无

[功能]

选项 -D 指定将指定文件输入为指令文件。

[应用]

在希望定义新存储区、重新定义默认存储区或将程序段分配到特定地址或存储区时，就需要创建指令文件。指定选项 -D，将此指令文件输入连接器。

[说明]

- 1) 只有磁盘型文件名可指定为‘文件名’。如果指定设备型文件名，就会导致异常中止错误。
- 2) 若省略文件名，则会导致异常中止错误。
- 3) 不允许指令文件的嵌套。
- 4) 指令文件中可指定的字符数无限制。
- 5) 如果不止一次指定选项 -D，或如果指定了不止一个文件名，则会产生异常中止错误。
- 6) 关于指令文件的详细说明请参见“6.4 连接指令”。

[使用示例]

重新定义默认存储区 ROM/RAM。

指令文件 K0S.DR 的内容:

```
memory ROM: (0000h, 1000h)
memory RAM: (0FE20h, 1E0h)
```

用 K0S.DR 进行连接。

```
C>lk78k0s k0smain.rel k0ssub.rel -dk0s.dr -s
```

(6) 连接列表文件输出指定 (-P/-NP)

语法: -P [输出文件名]
 :
 -NP
默认预设: -P 输入文件名.MAP

[功能]

- 1) 选项 -P 指定连接列表文件的输出。还指定输出文件的目标及文件名。
- 2) 选项 -NP 使选项 -P 不可用。

[应用]

- 1) 指定选项 -P 用来更改连接列表文件的输出目标或输出文件名。
- 2) 当只为了输出载入模块文件而进行连接时，应指定选项 -NP。这将缩短连接时间。

[说明]

- 1) 文件名可指定为磁盘型文件名或设备型文件名。但是，只有 CON、PRN、NUL 和 AUX 可指定为设备型文件名。若指定 CLOCK，则会产生异常中止错误。
- 2) 如果在指定选项 -P 时省略‘输出文件名’，则当前目录的连接列表文件名变为‘输入文件名.MAP’。
- 3) 只要‘输出文件名’中指定了路径名，‘输入文件名.MAP’就会输出到指定路径。
- 4) 若同时指定选项 -P 和 -NP，则最后指定的选项优先。

[使用示例]

创建连接列表文件 (K0S.MAP)。

```
C>lk78k0s k0smain.rel k0ssub.rel -pk0s.map -s
```

(7) 连接列表文件数据指定 (-KM/-NKM, -KD/-NKD, -KP/-NKP, -KL/-NKL)

(a) -KM/-NKM

语法: -KM
 : -NKM
默认预设: -KM

[功能]

- 1) 选项 -KM 输出映像列表到连接列表文件中。
- 2) 选项 -NKM 使选项 -KM 不可用。

[应用]

指定选项 -KM 用于输出映像列表到连接列表文件中。

[说明]

- 1) 若同时指定选项 -KM 和 -NKM，则最后指定的选项优先。
- 2) 若指定了选项 -NKM，则即便指定了选项 -KD 也无法输出连接指令文件到连接列表文件中。
- 3) 如果选项 -NKM、-NKP 和 -NKL 全部指定，则即便指定了选项 -P 也无法输出连接列表文件。

[使用示例]

将映像列表输出到连接列表文件 K0S.MAP 中。

```
C>lk78k0s k0smain.rel k0ssub.rel -pk0s.map -km -s
```

引用 K0S.MAP。

78K/0S Series Linker Vx.xx Date:xx Dec xxxx Page: 1

Command: k0smain.rel k0sub.rel -km -pk0s.map -s
 Para-file:
 Out-file: K0SMMAIN.LMF
 Map-file: K0S.MAP
 Direc-file:
 Directive:

*** Link information ***

3 output segment(s)
 37H byte(s) real data
 25 symbol(s) defined

*** Memory map ***

SPACE=REGULAR

MEMORY=ROM

BASE ADDRESS=0000H SIZE=4000H

OUTPUT SEGMENT CODE	INPUT SEGMENT CODE	INPUT MODULE	BASE ADDRESS	SIZE	
			0000H	0002H	CSEG AT
* gap *		SAMPM	0000H	0002H	
?CSEG		0002H	007EH		
			0080H	0035H	CSEG
	?CSEG	SAMPM	0080H	0015H	
	?CSEG	SAMPS	0095H	0020H	
* gap *			00B5H	3F4BH	

映象表

MEMORY=RAM

BASE ADDRESS=FD00H SIZE=0300H

OUTPUT SEGMENT	INPUT SEGMENT	INPUT MODULE	BASE ADDRESS	SIZE	
* gap *			FD00H	0120H	
DATA			FE20H	0003H	DSEG AT
	DATA	SAMPM	FE20H	0003H	
* gap *			FE23H	00DDH	
* gap (Not Free Area) *			FF00H	0100H	

Target chip:uPD789026
 Device file:Vx.xx

(b) -KD/-NKD

语法: -KD
 -NKD
默认预设: -KD

[功能]

- 1) 选项 -KD 将连接指令文件输出到连接列表文件中。
- 2) 选项 -NKD 使选项 -KD 不可用。

[应用]

指定选项 -KD 用于将连接指令文件输出到连接列表文件中。

[说明]

- 1) 若同时指定选项 -KD 和 -NKD，则最后指定的选项优先。
- 2) 若指定了选项 -NKM，则即便指定了选项 -KD 也无法将连接指令文件输出到连接列表文件中。
- 3) 如果选项 -NKM、-NKP 和 -NKL 全部指定，则即便指定了选项 -P 也无法输出连接列表文件。

[使用示例]

将连接指令文件输出到连接列表文件（K0S.MAP）中。

```
C>lk78k0s k0smain.rel k0ssub.rel -dk0s.dr -pk0s.map -kd -s
```

这里引用 K0S.MAP。

```
78K/0S Series Linker Vx.xx        Date:xx xxx xxxx Page: 1
```

```
Command:k0smain.rel k0ssub.rel -dk0s.dr -pk0s.map -kd -s
```

```
Para-file:
```

```
Out-file:K0SMMAIN.LMF
```

```
Map-file:K0S.MAP
```

```
Direc-file:K0S.DR
```

← 指令文件名

```
Directive: memory ROM: (0h, 1000h)
```

← 指令文件的内容

```
          memory RAM: (0fe20h, 1e0h)
```

```
*** Link information ***
```

```
3 output segment(s)
```

```
37H byte(s) real data
```

```
23 symbol(s) defined
```

```
*** Memory map ***
```

```
SPACE = REGULAR
```

```
MEMORY = ROM
```

```
BASE ADDRESS = 0000H   SIZE = 1000H
```

OUTPUT SEGMENT CODE	INPUT SEGMENT	INPUT MODULE	BASE ADDRESS 0000H	SIZE 0002H CSEG AT
:	:	:	:	:

(c) -KP/-NKP

语法: -KP
 : -NKP
默认预设: -NKP

[功能]

- 1) 选项 -KP 将公共符号列表输出到连接列表文件中。
- 2) 选项 -NKP 使选项 -KP 不可用。

[应用]

指定选项 -KP 用于将公共符号列表输出到连接列表文件中。

[说明]

- 1) 若同时指定选项 -KP 和 -NKP，则最后指定的选项优先。
- 2) 如果选项 -NKM、-NKP 和 -NKL 全部指定，则即便指定了选项 -P 也无法输出连接列表文件。
- 3) 若指定了选项 -NG，则即便指定了选项 -KP 也无法输出公共符号列表。

[使用示例]

将公共符号列表输出到连接列表文件（K0S.MAP）中。

```
C>lk78k0s k0smain.rel k0ssub.rel -g -pk0s.map -kp -s
```

这里引用 K0S.MAP。

78K/0S Series Linker Vx.xx Date:xx xxx xxxx Page: 1

Command:k0smain.rel k0ssub.rel -g -pk0s.map -kp -s

Para-file:

Out-file:K0SMMAIN.LMF

Map-file:K0S.MAP

Direc-file:

Directive:

*** Link information ***

3 output segment(s)
37H byte(s) real data
23 symbol(s) defined

*** Memory map ***

SPACE = REGULAR

⋮

78K/0S Series Linker Vx.xx Date:xx xxx xxxx Page: 2

*** Public symbol list ***

MODULE	ATTR	VALUE	NAME
SAMPM	ADDR	0000H	MAIN
SAMPM	ADDR	0080H	START
SAMPS	ADDR	009AH	CONVAH
	NUM	FE20H	__@STBEG
	NUM	FD00H	__@STEND

公共符号列表

Target chip:uPD78xxx

Device file:Vx.xx

(d) -KL/-NKL

语法: -KL
 : -NKL
默认预设: -NKL

[功能]

- 1) 选项 -KL 将局部符号列表输出到连接列表文件中。
- 2) 选项 -NKL 使选项 -KL 不可用。

[应用]

指定选项 -KL 用于将局部符号列表输出到连接列表文件中。

[说明]

- 1) 若同时指定选项 -KL 和 -NKL，则最后指定的选项优先。
- 2) 如果选项 -NKM、-NKP 和 -NKL 全部指定，则即便指定了选项 -P 也无法输出连接列表文件。
- 3) 若指定了选项 -NG，则即便指定选项 -KL 也无法输出局部符号列表。

[使用示例]

将局部符号列表输出到连接列表文件（K0S.MAP）中。

```
C>lk78k0s k0smain.rel k0ssub.rel -g -pk0s.map -kl -s
```

这里引用 K0S.MAP。

78K/0S Series Linker Vx.xx Date:xx xxx xxxx Page: 1

Command:k0smain.rel k0ssub.rel -g -pk0s.map -kl -s

Para-file:

Out-file:K0SMAIN.LMF

Map-file:K0S.MAP

Direc-file:

Directive:

*** Link information ***

3 output segment(s)
37H byte(s) real data
23 symbol(s) defined

*** Memory map ***

SPACE = REGULAR

⋮

78K/0S Series Linker Vx.xx Date:xx xxx xxxx Page: 2

*** Local symbol list ***

MODULE	ATTR	VALUE	NAME
SAMPM	MOD		SAMPM
SAMPM	DSEG		DATA
SAMPM	ADDR	FE20H	HDTSA
SAMPM	ADDR	FE21H	STASC
SAMPM	CSEG		CODE
SAMPM	CSEG		?CSEG
SAMPS	MOD		SAMPS
SAMPS	CSEG		?CSEG
SAMPS	ADDR	00ACH	SASC
SAMPS	ADDR	00B2H	SASC1

局部符号列表

Target chip:uPD78xxx

Device file:Vx.xx

(8) 连接列表文件格式指定 (-LL, -LF/-NLF)**(a) -LL**

语法: -LL [行数]
默认预设: -LL66 (在显示输出的情况下不分页)

[功能]

选项 -LL 更改连接列表文件中 1 页能打印的行数。

[应用]

指定选项 -LL 用于更改连接列表文件中 1 页能打印的行数。

[说明]

- 1) 可用选项 -LL 指定的行数范围如下所示。

$$20 \leq 1 \text{ 页中打印的行数} \leq 32767$$

若指定为范围之外的数值或非数值, 则会产生异常异常中止错误。

- 2) 若省略行数, 则指定为 66。
- 3) 若行数指定为 0, 则不进行分页。
- 4) 若指定了选项 -NP, 则选项 -LL 不可用。

[使用示例]

将连接列表文件每页行数指定为 20。

```
C>lk78k0s k0smain.rel k0ssub.rel -pk0s.map -ll20 -s
```

这里引用 K0S.MAP。

78K/0S Series Linker Vx.xx Date:xx xxx xxxx Page: 1

Command:k0smain.rel k0ssub.rel -pk0s.map -ll20 -s

Para-file:

Out-file:K0SMAIN.LMF

Map-file:K0S.MAP

Direc-file:

Directive:

*** Link information ***

3 output segment(s)

37H byte(s) real data

78K/0S Series Linker Vx.xx Date:xx xxx xxxx Page: 2

23 symbol(s) defined

*** Memory map ***

SPACE = REGULAR

MEMORY = ROM

BASE ADDRESS = 0000H SIZE = 2000H

OUTPUT	INPUT	INPUT	BASE	SIZE
SEGMENT	SEGMENT	MODULE	ADDRESS	

78K/0S Series Linker Vx.xx Date:xx xxx xxxx Page: 3

	CODE		0000H	0002H	CSEG AT
		CODE	SAMPM	0000H	0002H
* gap *				0002H	007EH
	?CSEG			0080H	0035H CSEG
		?CSEG	SAMPM	0080H	0015H
		?CSEG	SAMPS	0095H	0020H
* gap *				00B5H	1F4BH

MEMORY = RAM

BASE ADDRESS = FE00H SIZE = 0200H

OUTPUT	INPUT	INPUT	BASE	SIZE
--------	-------	-------	------	------

(b) -LF/-NLF

语法: -LF
 : -NLF
默认预设: -NLF

[功能]

- 1) 选项 -LF 在连接列表文件末尾插入换页 (FF) 码。
- 2) 选项 -NLF 使选项 -LF 不可用。

[应用]

如果希望在打印连接列表文件内容后添加分页符, 应指定选项 -LF, 添加换页码。

[说明]

- 1) 若指定了选项 -NP, 则选项 -LF 不可用。
- 2) 若同时指定选项 -LF 和 -NLF, 则最后指定的选项优先。

[使用示例]

在连接列表文件末尾添加换页代码。

```
C>lk78k0s k0smain.rel k0ssub.rel -pk0s.map -lf -s
```

(9) 错误列表文件输出指定 (-E/-NE)

语法: -E [文件名]
 : -NE
默认预设: -NE

[功能]

- 1) 指定选项 -E 用于指定错误列表文件的输出目标和文件名。
- 2) 选项 -NE 使选项 -E 不可用。

[应用]

指定选项 -E 用于更改错误列表文件的输出目标和输出文件名。

[说明]

- 1) 错误列表文件的文件名可指定为磁盘型文件名或设备型文件名。但是，若指定设备型文件名 **CLOCK** 则会产生异常中止错误。
- 2) 当指定选项 -E 且省略输出文件名时，错误列表文件名将是‘输入文件名.ELK’。
- 3) 当指定选项 -E 且省略驱动器名时，错误列表文件将输出到当前驱动器。
- 4) 若同时指定选项 -E 和 -NE，则最后指定的选项优先。

[使用示例]

创建错误列表文件（K0S.ELK）。

```
C>lk78k0s k0smain.rel k0ssub.rel -dk0s.dr -ek0s.elk -s
```

在指令文件内容中产生了错误。引用 K0S.ELK。

```
K0S.DR(3) : F102 Directive syntax error
```

(10) 库文件指定 (-B)

语法: -B 文件名
默认预设: 无

[功能]

选项 -B 指定输入为库文件的文件。

[应用]

连接器在库文件中检索输入模块引用的模块，仅将该模块与输入模块连接。

库文件用于在单个文件中注册两个（或更多）模块。

通过创建可在许多程序中同样使用的库文件，文件管理和操作变得更简单、更有效。指定选项 -B 来输入库文件到连接器中。

[说明]

- 1) 只有磁盘型文件名可指定为文件名。
- 2) 文件名不可省略。
- 3) 如果指定了包含路径名的文件名，库文件就会从该路径输入。如果指定路径不存在库文件，就会产生错误。
- 4) 如果指定了不包含路径名的文件名，则库文件会从选项 -I 指定的路径或从默认搜索路径输入。
- 5) 若指定选项 -B 两次（或更多次），库文件会以指定的顺序进行输入。最多可指定 10 个 -B 选项。
- 6) 关于创建库文件方法的详细说明请参见“第八章 库管理程序”。

[使用示例]

输入库文件（KOS.LIB）。

（库文件注册了 KOSSUB.REL）。

```
C>lk78k0s k0smain.rel -bk0s.lib
```

(11) 库文件读取路径指定 (-I)

语法: -I 路径名 [, 路径名] ... (可指定两个 (或更多) 路径名)
默认预设: 由环境变量 'LIB78K0S' 指定的路径
 若未指定路径则为当前路径

[功能]

选项 -I 指定从指定路径输入库文件。

[应用]

用选项 -I 从某个路径检索库文件。

[说明]

- 1) 只有在选项 -B 指定了库文件名而未包含路径名时, 选项 -I 才是可用的。
- 2) 可以指定两个 (或更多) -I。可指定两个 (或更多) 路径, 用 ',' 隔开。在 ',' 前后不能插入空白。
- 3) 每个连接最多可指定 10 个路径名。当指定两个 (或更多) 路径名时, 连接器以指定的顺序搜索库文件。
- 4) 即便指定路径不存在库文件也不会导致错误。
- 5) 若省略路径名, 则会产生异常异常中止错误。
- 6) 如果由选项 -B 指定了库文件而未包含路径名, 则连接器将以下列次序搜索路径。
 - a. 选项 -I 指定的路径
 - b. 由环境变量 'LIB78K0S' 指定的路径
 - c. 当前路径

如果在这些路径下均未找到指定名称的库文件, 则会出现错误。

[使用示例]

从目录 A:\LIB 搜索库文件。

```
C>lk78k0s k0smain.rel k0ssub.rel -bk0s.lib -ia:\lib -s
```

(12) 参数文件指定 (-F)

语法: -F 文件名
默认预设: 该选项及输入文件名只能用命令行输入。

[功能]

选项-F指定连接器选项的输入和来自指定文件的输入文件名。

[应用]

当启动连接器所需的数据不适合命令行使用时，指定选项 -F。如果希望每次进行连接时重复指定相同选项，可在参数文件中指定这些选项并指定选项 -F。

[说明]

- 1) 只有磁盘型文件名可指定为‘文件名’。若指定设备型文件名，则会产生异常异常中止错误。
- 2) 如果省略文件名，会产生异常异常中止错误。
- 3) 不允许参数文件的嵌套。如果在参数文件内指定选项 -F，会产生异常异常中止错误。
- 4) 参数文件中可写入的字符数无限制。
- 5) 用空格、制表符或[-]隔开选项或文件名。
- 6) 在命令行指定给参数文件的位置处，参数文件中写入的选项和输入文件名将进行扩展。
- 7) 最后指定的扩展选项优先。
- 8) 所有在‘;’后，[-]或‘EOF（文件结束）’前键入的字符均当作注释。
- 9) 如果指定选项 -F 两次（或更多次），会产生异常异常中止错误。

[使用示例]

用参数文件进行连接。

如下设置参数文件（K0S.PLK）的内容。

```
;parameter file  
k0smain.rel k0ssub.rel -ok0s.lmf -pk0s.map -e -s  
-ta:\tmp -g
```

在命令行中键入下列内容。

```
C>>lk78k0s -fk0s.plk -s
```

(13) 临时文件创建路径指定 (-T)

语法: -T 路径名
默认预设: 在环境变量 **TMP** 指定的路径创建临时文件。
 未指定路径时, 在当前路径创建临时文件。

[功能]

选项 -T 指定创建临时文件的路径。

[应用]

用选项 -T 来指定创建临时文件的位置。

[说明]

- 1) 只能将一个路径指定为路径名。
- 2) 路径名不能省略。
- 3) 即使存在先前创建的临时文件, 只要文件未受保护就会被改写。
- 4) 只要所需的存储器长度可用, 临时文件就会在存储器中进行扩展。如果没有足够的存储器可用, 则临时文件的内容将被写入磁盘中。这些临时文件之后可通过保存的磁盘文件进行访问。
- 5) 连接结束时临时文件删除。当按 (**CTRL-C**) 异常中止连接时也会将其删除。
- 6) 创建临时文件的路径按照下列次序确定。
 - a. 由选项 -T 指定的路径
 - b. 由环境变量 **TMP** 指定的路径 (省略选项 -T 时)
 - c. 当前路径 (未设置 **TMP** 时)

当指定 a. 或 b. 时, 如果在指定路径无法创建临时文件就会产生异常中止错误。

[使用示例]

指定临时文件输出到目录 'TMP' 。

```
C>lk78k0s k0smain.rel k0ssub.rel -t\tmp -s
```

(14) 设备文件搜索路径指定 (-Y)

语法: -Y 路径名
默认预设: 设备文件将从下列次序确定的路径读取。
 1) <..\dev> (lk78k0s.exe 的启动路径)
 2) LK78K0S 启动路径
 3) 当前目录
 4) 环境变量 PATH

[功能]

从指定路径读取设备文件。

[应用]

指定存在设备文件的路径。

[说明]

- 1) 如果在选项 -Y 之后指定了非路径名, 则产生异常中止错误。
- 2) 如果在选项 -Y 后省略路径名, 产生异常中止错误。
- 3) 设备文件的读取路径按照如下次序确定。
 - a. 选项 -Y 指定的路径
 - b. <..\dev> (lk78k0s.exe 的启动路径)
 - c. LK78K0S 的启动路径
 - d. 当前目录
 - e. 环境变量 PATH

[使用示例]

将设备文件路径指定为目录 a:\78k0s\dev。

```
C>>lk78k0s k0smain.rel k0ssub.rel -ya:\78k0s\dev -s
```

(15) 报警信息输出指定 (-W)

语法: -W [级]
默认预设: 输出普通报错信息。

[功能]

选项 -W 指定是否输出报警信息到控制台。

[应用]

指定输出报警信息的级别。

[说明]

- 1) 如果在选项 -W 之后指定了非路径名，则产生异常中止错误。
- 2) 只能指定 0 级、1 级、2 级。
- 3) 有下列输出级别：
 - 0 ... 不输出报警信息。
 - 1 ... 输出正常报警信息。
 - 2 ... 输出详细报警信息。有关输出报警条件的详细说明请参见“表 12-3 连接错误信息”。

[使用示例]

在选项 -W 中指定 2 级。

```
C>lk78k0s k0smain.rel k0ssub.rel -w2 -s
```

(16) 帮助指定 (--)

语法: --
默认预设: 不显示

[功能]

选项 -- 在显示屏上显示帮助信息。

[应用]

帮助信息是连接器选项的说明表。在执行连接器时参考这些内容。

[说明]

当指定选项 --时，所有其它连接器选项均不可用。

注意事项 该选项不能在 PM +上指定。

要想参考 PM +s 帮助，可点击<连接器选项设置>对话框的“帮助”按钮。

[使用示例]

在指定选项 -- 时，显示屏上输出帮助信息。

```
C>lk78k0s --
```

```
78K/0S Series Linker Vx.xx [xx xxx xx]
```

```
Copyright (C) NEC Electronics Corporation xxxx, xxxx
```

```
usage:lk78k0s [option [...]] input-file [option [...]]
```

```
The option is as follows ([ ] means omissible).
```

```
-ffile          :Input option or input-file name from specified file.
-dfile         :Read directive file from specified file.
-bfile         :Read library file from specified file.
-idirectory [,directory..] :Set library file search path.
-o [file] /-no  :Create load module file [with specified name] / Not.
-p [file] /-np  :Create link map file [with specified name] / Not.
-e [file] /-ne  :Create error list file [with specified name] / Not.
-tdirectory    :Set temporary directory.
-km/nkm        :Output map list to link map file / Not.
-kd/-nkd       :Output directive file image to link map file / Not.
-kp/-nkp       :Output public symbol list to link map file / Not.
-kl/-nkl       :Output local symbol list to link map file / Not.
-ll [page length] :Specify link map file lines per page.
-lf/-nlf       :Add Form Feed at end of the link map file / Not.
-s [memory area] /-ns :Create stack symbol [in specified memory area] / Not.
-g/-ng         :Output symbol information to load module file / Not.
-ydirectory    :Set device file search path.
-j/-nj        :Create load module file if fatal error occurred / Not.
-w            :Change warning level (n = 0 to 2).
--           :Show this message.
```

```
Press RETURN to continue ...
```

```
DEFAULT ASSIGNMENT :-o -p -ne -km -kd -nkp -nkl -ll66 -nlf -ns -g -nj -w1
```

```
directive file usage:
```

```
MEMORY memory-area-name: (origin-value, size) [/memory-space-name]
```

```
MERGE segment-name: [location-type-definition] [merge-type-definition]
```

```
    [= memory-area-name] [/memory-space-name]
```

```
example:MEMORY ROM:(0H,4000H)
```

```
MEMORY RAMA:(0H,100H)
```

```
MERGE CSEG1:= ROM
```

```
MERGE DSEG1:AT(80H)
```

6.7 在 PM+中设置选项

本节介绍 PM+连接器选项设置的方法。

6.7.1 选项设置方法

从PM+的[Tools]菜单选择[Linker Options...]或点击，显示< Linker Options>对话框。
可通过在对话框中输入所需的选项来设置连接器选项。

图 6-3 < Linker Options >对话框（当选择<<Output1>>标签时）

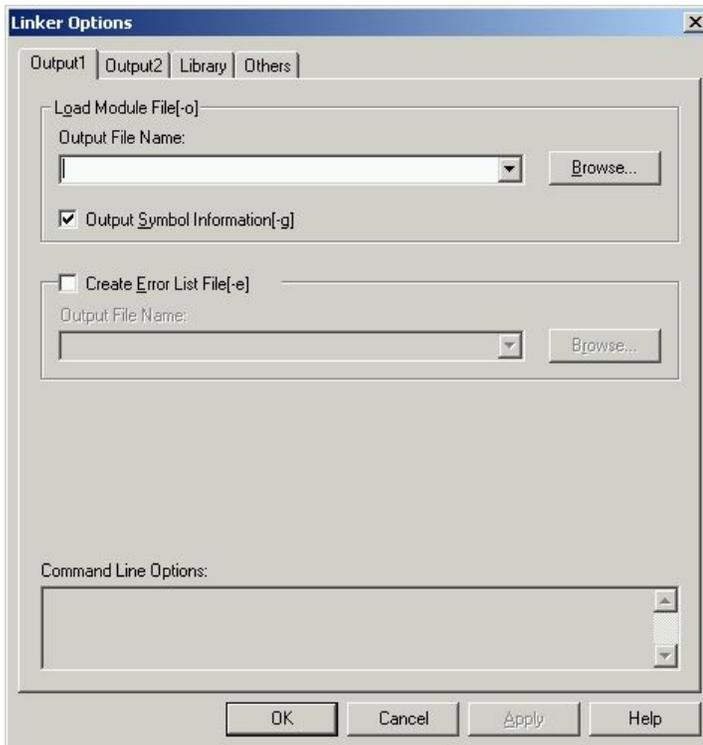


图 6-4 < Linker Options >对话框（当选择<< Output 2>>标签时）

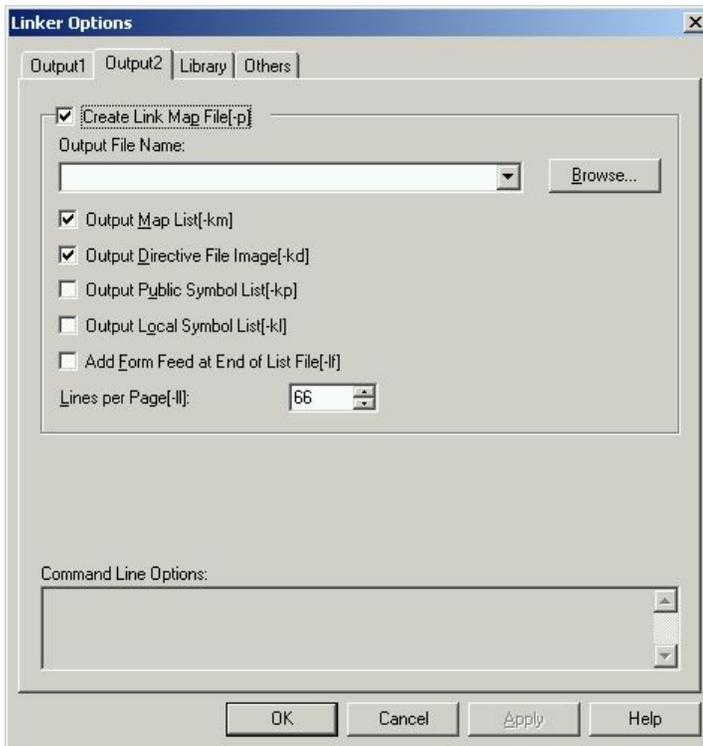


图 6-5 < Linker Options >对话框（当选择<<Library>>标签时）

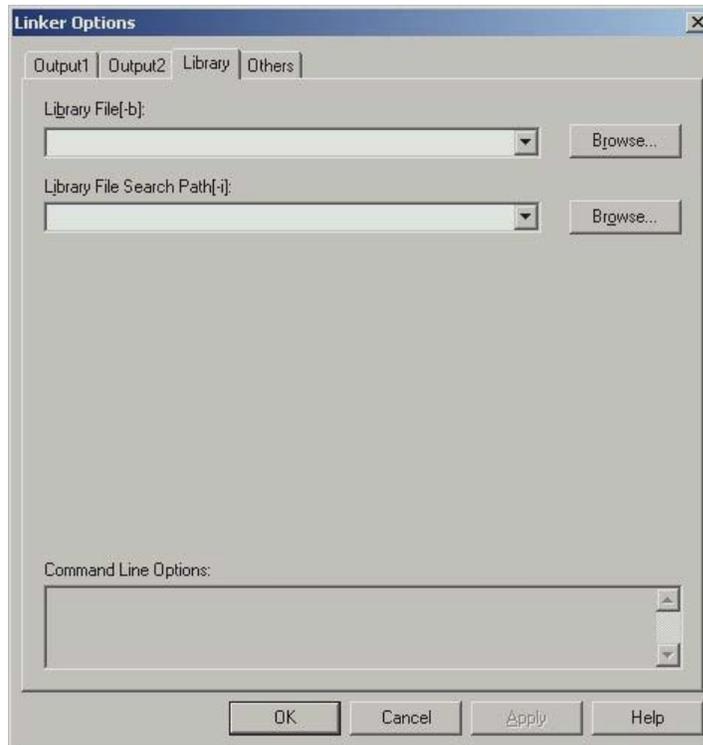
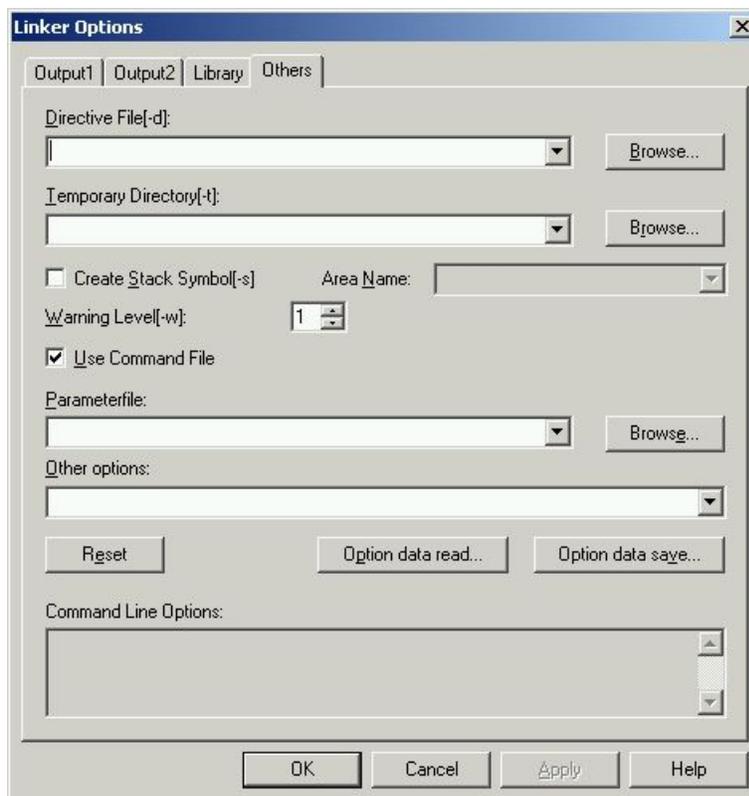


图 6-6 < Linker Options >对话框（当选择<<Others>>标签时）



6.7.2 选项

下面介绍< Linker Options >对话框中的各选项。

- **Load Module File [-o]**
输出文件名:
通过[Browse...]按钮或直接输入, 指定载入模块文件的输出路径。
- **Output Symbol Information [-g]**
在载入模块文件中附加调试数据(局部符号信息)。
- **Create Error List File [-e]**
输出文件名:
要想输出错误列表文件, 应在输入框中输入文件名。
指定路径用[Browse...]按钮。
- **Create Link Map File [-p]**
选择该复选框来输出连接列表文件。
通过[Browse...]按钮或直接输入, 指定连接列表文件的输出路径。
- **Output Map List [-km]**
在连接列表文件中输出映象文件。
- **Output Directive File Image [-kd]**
在连接列表文件中输出连接指令文件。
- **Output Public Symbol List [-kp]**
在连接列表文件中输出公共符号列表。
- **Output Local Symbol List [-kl]**
在连接列表文件中输出局部符号列表。
- **Add Form Feed at End of List File [-lf]**
在打印连接列表文件内容后附加分页代码。
- **Lines per Page [-ll]**
指定连接列表文件每页的行数(可选 20 至 32767 字符)。
- **Library File [-b]**
通过[Browse...]按钮或直接输入, 指定将指定文件输入为库文件。
- **Library File Search Path [-i]**
通过[Browse...]按钮或直接输入, 指定从指定路径输入库文件。
- **Directive File [-d]**
通过[Browse...]按钮或直接输入, 指定输入指定文件为指令文件。
- **Temporary Directory [-t]**
通过[Browse...]按钮或直接输入, 指定临时文件的创建路径。
- **Create Stack Symbol [-s]**
当选中该选项时, 保证存储区中的最大空区为栈区。
- **Area Name**
指定用户定义的存储区名或默认存储区名。
- **Warning Level [-w]**
指定报警信息的输出级别。
0: 不输出报警信息。
1: 输出正常报警信息。
2: 输出详细报警信息。
- **Use Command File**
选择此复选框来创建命令文件。

- **Parameterfile**
通过[Browse...]按钮或直接输入，读取用户定义的参数文件。
- **Other options**
如果在复选框或单选按钮可选择的选项之外还希望指定其它选项，可在输入框中输入该选项。
- **Reset**
使输入内容复位。
- **Option data read...**
打开 <选项数据读取>对话框，在指定选项数据文件后该文件被读取。
- **Option data save...**
在打开<选项数据保存>对话框后，将选项数据文件保存到选项数据文件中并命名。
- **Command Line Options**
此编辑框只读。显示当前设置选项字符串。

第 7 章 目标转换器

目标转换器输入由 RA78K0S 连接器输出的载入模块文件（必须在此时确定所有引用地址数据），然后将该数据转换成十六进制格式并输出为目标文件。

目标转换器还将用于符号调试的符号数据输出为符号表文件。

当产生目标转换器错误时，显示屏上出现错误信息对出错原因进行说明。

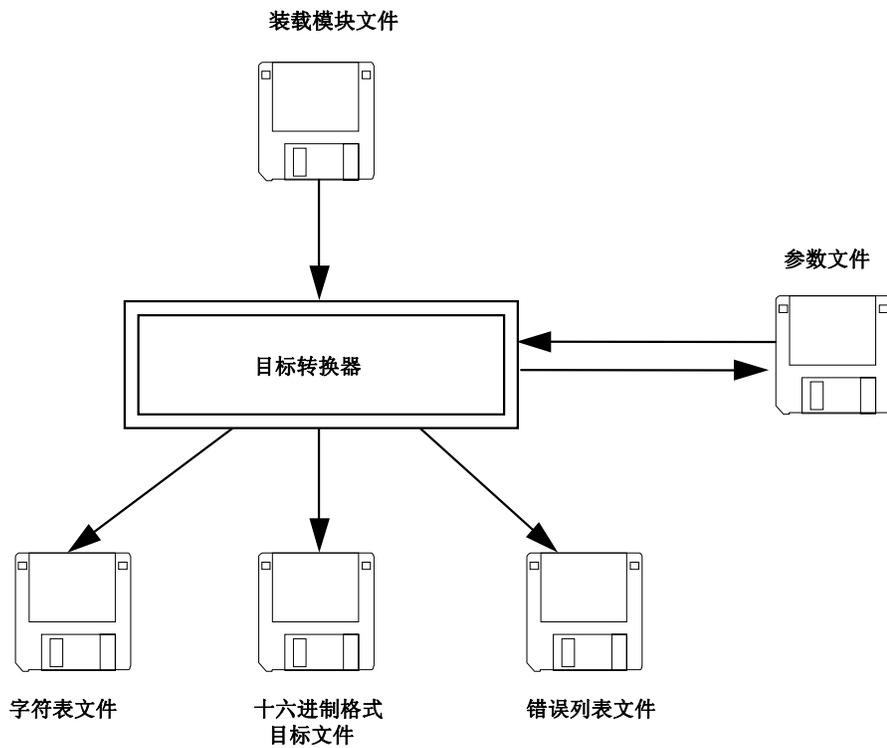
7.1 目标转换器的 I/O 文件

目标转换器的 I/O 文件如下所示。

表 7-1 目标转换器的 I/O 文件

类型	文件名	说明	默认文件类型
输入文件	载入模块文件	<ul style="list-style-type: none"> • 这些文件为目标码的映像文件，连接后输出。 • 这些文件由连接器输出。 	.LMF
	参数文件	<ul style="list-style-type: none"> • 这些文件包含执行程序的参数 • 这些文件由用户创建。 	.POC
输出文件	十六进制格式目标文件	<ul style="list-style-type: none"> • 这些文件通过将载入模块文件转换为十六进制对象格式进行创建。 • 这些文件用在掩膜 ROM 开发和 PROM 编程中。 	.HEX
	符号表文件	<ul style="list-style-type: none"> • 这些文件包含输入文件各模块中含有的符号数据。 	.SYM
	错误列表文件	<ul style="list-style-type: none"> • 这些文件包含对象转换中产生的出错数据。 	.EOC

图 7-1 目标转换器的 I/O 文件



7.2 目标转换器的功能

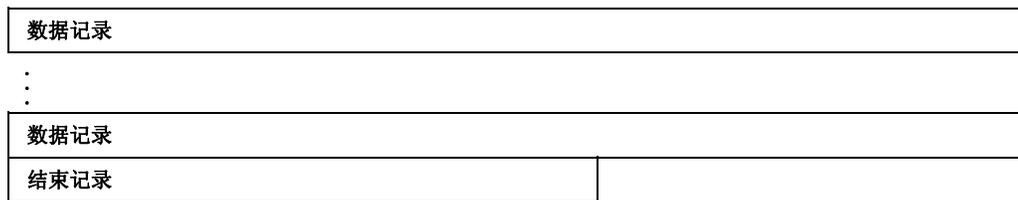
(1) 十六进制格式目标文件

目标转换器输出的十六进制格式目标文件可以输入 HEX 载入程序中，如 PROM 编程器或调试器。下面是示例程序的十六进制格式目标文件。

```
:0200000080007E
:10008000F5201AFC20FE229500F821FE0A27EB88B5
:10009000A25EB30FE2F00000000630F22AC000A9F
:1000A000E70A430F630F22AC000AE520130A380267
:0500B0008307833020EE
:00000001FF
```

[Intel 标准 HEX 格式目标文件格式]

图 7-2 Intel 标准格式



注

注 这里重复了数据记录。

(a) 数据记录

```
: XX XXXX 00 DD...DD SS
| | | | | |
(1) (2) (3) (4) (5) (6)
```

- (1) 记录标记
指示记录的开头。
- (2) 码数 (2 位)
记录中存储代码的字节数。最多可存储 16 字节。
- (3) 存储地址 (偏移量)
记录中显示代码的起始地址 (偏移量) 显示为 4 位十六进制。
- (4) 记录类型 (2 位)
固定为 00。
- (5) 代码 (最多 32 位)
目标码每次显示一个字节，高 4 位和低 4 位分开。代码中最多可以表示 16 字节。
- (6) 校验和 (2 位)
将输入值按顺序去减 0 得到的值。

(b) 结束记录

```
: 00 0000 01 FF
| | | | |
(1) (2) (3) (4) (5)
```

- (1) 记录标记
- (2) 码数, 固定为 00
- (3) 固定为 0000
- (4) 记录类型, 固定为 01
- (5) 校验和, 固定为 FF

(2) 符号表文件

目标转换器输出的符号表文件输入到调试器中。
下面是示例程序的符号表文件。

```
#04
;FF PUBLIC
010095CONVAH
010000MAIN
010080START
;FF SAMPM
<02FE20HDTSA
02FE21STASC
;FF SAMPS
<0100ACSASC
0100B2SASC1
=
```

[符号表文件格式]

符号表开头	#	04	CR	LF			
公共符号开头	;	FF	4个空格	PUBLIC	CR	LF	
注 1→		符号属性	符号值	公共符号名	CR	LF	
		•	•	•	•	•	公共符号名
		•	•	•	•	•	
		•	•	•	•	•	
局部符号开头	;	FF	4个空格	模块名 1	CR	LF	各模块的局部符号
	<	符号属性	符号值 注 2	局部符号名	CR	LF	
		符号属性	符号值	局部符号名	CR	LF	
		•	•	•	•	•	
		•	•	•	•	•	
		•	•	•	•	•	
以对象模块为单位进行重复	;	FF	4个空格	模块名 2	CR	LF	
		•	•	•	•	•	
		•	•	•	•	•	
		•	•	•	•	•	
符号表结束标记	=	CR	LF				

- 注
1. 符号属性为“表 7-2 符号属性的值”中所示的值。
 2. 有关符号值，请参见“图 7-3 符号值的格式”。

表 7-2 符号属性的值

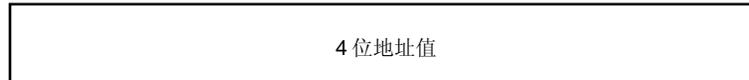
值	符号属性
00	由 EQU 指令定义的常量
01	代码段内的标号
02	数据段内的标号
03	位符号
FF	模块名

图 7-3 符号值的格式

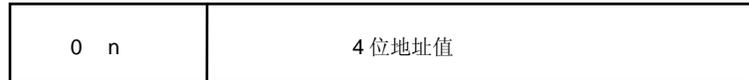
- 当符号属性为 NUMBER 时



- 当符号属性为 ADDRESS



- 当符号属性为位符号时



↑

n: 位的位置 (0 至 7)

7.3 目标转换器的启动

7.3.1 目标转换器的启动

可用下列两种方法来启动目标转换器。

(1) 从命令行启动

```
X>[路径名] oc78k0s [Δ选项] ...Δ载入模块文件名 [Δ选项]...[Δ]
```

(1)	(2)	(3)	(4)	(5)	(4)

- (1) 当前驱动器名
- (2) 当前目录名
- (3) 目标转换器命令文件名
- (4) 包含目标转换器操作的具体指令。
- (5) 要转换的载入模块的文件名

示例 C>>oc78k0s k0s.lmf -osample.hex

注意事项 如果指定了不止一个目标转换器，则应用空格隔开单个目标转换器选项。关于目标转换器选项的详细说明请参见“7.4 目标转换器选项”。

(2) 从参数文件启动

如果启动目标转换器所需数据不适合命令行，或如果进行对象转换时反复指定相同的目标转换器选项，就应使用参数文件。

要想从参数文件启动目标转换器，应在命令行中指定参数文件说明选项（-F）。

从参数文件启动目标转换器，如下。

```
X>oc78k0s [Δ载入模块文件] Δ-f  参数文件名
           |                    |
           (1)                  (2)
```

(1) 参数文件说明选项

(2) 包含启动目标转换器所需数据的文件

注 用编辑器来创建参数文件。

书写参数文件内容的规则如下。

```
[[[Δ] 选项 [Δ选项] ... [Δ] Δ]] ...
```

- 注**
1. 如果命令行中省略载入模块文件名，则在参数文件中只能指定一个载入模块文件名。
 2. 载入模块文件名也可以在选项后面指定。
 3. 在参数文件中写入所有应该在命令行中指定的目标转换器选项和输出文件名。

示例 用编辑器创建参数文件（K0S.POC）。

K0S.POC 的内容

```
;parameter file
k0s.lmf -osample.hex
-ssample.sym -r
```

用参数文件 K0S.POC 启动目标转换器。

```
C>oc78k0s -fk0s.poc
```

7.3.2 执行开始结束信息

(1) 执行开始信息

在目标转换器启动时，显示屏上出现执行开始信息。

```
78K/0S Series Object Converter Vx.xx [xx xxx xx]
Copyright (C) NEC Electronics Corporation xxxx, xxxx
```

(2) 执行结束信息

如果没有检测到对象转换导致的对象转换错误，则目标转换器输出下列信息到显示屏上，并将控制还给操作系统。

```
Target chip:uPD789xxx
Device file:Vx.xx
```

```
Object Conversion Complete, 0 error(s) and 0 warning(s) found.
```

如果检测到由该对象转换导致的对象转换错误，目标转换器就会将错误号输出到显示屏上，并将控制还给操作系统。

```
Target chip:uPD789xxx
Device file:Vx.xx
```

```
Object Conversion Complete, 3 error(s) and 0 warning(s) found.
```

如果在对象转换期间目标转换器检测到致命错误，使其无法继续对象转换处理，则目标转换器会将一条信息输出到显示屏上，退出对象转换，将控制还给操作系统。

示例 1 指定了不存在的载入模块文件名。

```
C>>oc78k0s sample.lmf

78K/0S Series Object Converter Vx.xx [xx xxx xx]
Copyright (C) NEC Electronics Corporation xxxx, xxxx

A006 File not found 'SAMPLE.LMF'
Program aborted.
```

在上面的例子中，指定了不存在的载入模块文件。结果产生错误，目标转换器使对象转换中止。

示例 2 指定了不存在的目标转换器选项。

```
C>oc78k0s k0s.lmf -a

78K/0S Series Object Converter Vx.xx [xx xxx xx]
  Copyright (C) NEC Electronics Corporation xxxx, xxxx

A018 Option is not recognized '-a'
Please enter 'OC78K0S --', if you want help messages.
Program aborted.
```

在上面的例子中，指定了不存在的目标转换器选项。结果产生错误，目标转换器使对象转换中止。如果显示错误信息且对象转换中止，请在“**第十二章 错误信息**”中查找原因，采取相应措施。

7.4 目标转换器选项

7.4.1 目标转换器选项的类型

目标转换器选项是关于目标转换器操作的细节指令。目标转换器选项分成 8 种类型。

目标转换器选项的分类及各类型的说明如下所示。

表 7-3 目标转换器选项

编号	分类	选项	说明
1	十六进制格式目标文件输出指定	-O	指定十六进制格式目标文件的输出。
		-NO	
2	符号表文件输出指定	-S	指定符号表文件的输出。
		-NS	
3	以对象地址次序进行排序指定	-R	以地址次序对 HEX 格式对象进行排序。
		-NR	
4	对象补充指定	-U	将指定补充值作为目标码输出到无 HEX 格式对象输出的地址区。
5	错误列表文件输出指定	-E	输出错误列表文件。
		-NE	
6	参数文件指定	-F	从指定文件输入输入文件名和选项。
7	设备文件搜索路径指定	-Y	从指定路径读取设备文件。
8	帮助指定	--	在显示屏上显示帮助信息。

注 本表为目标转换器选项的简介。在实际使用目标转换器选项时，请参见“C.4 目标转换器选项列表”。

7.4.2 目标转换器选项说明

本节包含对各目标转换器选项的具体说明。

(1) 十六进制格式目标文件输出指定 (-O/-NO)

语法: -O [输出文件名]
 : -NO
默认预设: -O 输入文件名.HEX

[功能]

- 1) 选项 -O 指定十六进制格式目标文件的输出。选项 -O 还指定输出目标和输出文件名。
- 2) 选项 -NO 指定不输出十六进制格式目标文件。

[应用]

- 1) 指定选项 -O 来更改十六进制格式目标文件的输出目标和输出文件名。
- 2) 在仅为了输出符号表文件而进行对象转换时，应指定选项 -NO。这将缩短对象转换时间。

[说明]

- 1) 给‘输出文件名’指定一个磁盘型文件名。
如果指定设备型文件名，就会导致中止错误。
- 2) 如果在指定选项 -O 时省略‘输出文件名’，则将输出十六进制格式目标文件‘输入文件名.HEX’到当前目录。
- 3) 只要‘输出文件名’中指定了路径名，‘输入文件名.HEX’就会输出到指定路径。
- 4) 如果同时指定选项-O和-NO，则最后指定的选项优先。

[使用示例]

输出十六进制格式目标文件（SAMPLE.HEX）。

```
C>>oc78k0s k0s.lmf -osample.hex
```

(2) 符号表文件输出指定 (-S/-NS)

语法: -S [输出文件名]
 :
 -NS
默认预设: -S 输入文件名.SYM

[功能]

- 1) 选项 -S 指定符号表文件的输出。选项 -S 还指定输出目标和输出文件名。
- 2) 选项 -NS 指定不输出符号表文件。

[应用]

- 1) 指定选项 -S 来更改符号表文件的输出目标和输出文件名。
- 2) 在仅为了输出十六进制格式目标文件而进行对象转换时，应指定选项 -NS。这将缩短对象转换时间。

[说明]

- 1) 给‘输出文件名’指定一个磁盘型文件名。
如果指定设备型文件名，就会导致中止错误。
- 2) 如果在指定了选项 -S 时省略‘输出文件名’，则会输出符号表文件‘输入文件名.SYM’到当前目录。
- 3) 只要‘输出文件名’中指定了路径名，‘输入文件名.SYM’就会输出到指定路径。
- 4) 若同时指定选项 -S 和 -NS，则最后指定的选项优先。

[使用示例]

输出符号表文件 (SAMPLE.SYM)。

```
C>oc78k0s k0s.lmf -ssample.sym
```

(3) 以对象地址次序进行排序指定 (-R/-NR)

语法: -R
 : -NR
默认预设: -NR

[功能]

- 1) 选项 -R 输出以地址次序排序的 HEX 格式对象。
- 2) 选项 -NR 输出以载入模块文件中的存储次序排列的 HEX 格式对象。

[应用]

当需要以地址次序对 HEX 格式对象进行排序时，指定选项 -R。

[说明]

- 1) 若同时指定选项 -R 和 -NR，则最后指定的选项优先。
- 2) 如果指定了选项 -NO，则选项 -R/-NR 不可用。

[使用示例]

以地址次序对 HEX 格式对象进行排序。

```
C>oc78k0s k0s.lmf -r
```

(4) 对象补充指定 (-U)

语法: -U 补充值 [, [起始], 长度]
 默认预设: 无

[功能]

选项 -U 将指定补充值作为目标码输出到 HEX 格式对象尚未输出的地址区。

[应用]

HEX 格式对象尚未输出的地址区可能写入不必要代码。如果由于某种原因这些地址被程序访问，其行为可能是不可预料的。通过指定选项 -U，可将代码写入 HEX 格式对象尚未输出的地址区。

[说明]

- 1) 可指定为补充值的取值范围如下。

$$0H \leq \text{补充值} \leq 0FFH$$

补充值可以二进制、八进制、十进制或十六进制数进行指定。如果指定了范围以外的值或非数值，则会产生中止错误。

- 2) “起始”指定进行补充的起始地址区。

可指定为起始区的取值范围如下。

$$0H \leq \text{起始} \leq \text{除 SFR 区外程序空间中的最大地址}$$

起始区可以二进制、八进制、十进制或十六进制数进行指定。如果指定了范围以外的值或非数值，则会产生中止错误。如果省略起始点，则假定指定为 0。

- 3) “长度”指定进行补充的地址区的长度。可指定为长度的取值范围如下。

$$1H \leq \text{长度} \leq \text{除 SFR 区的程序空间中的最大地址}$$

长度可以二进制、八进制、十进制或十六进制数进行指定。如果指定了范围以外的值或非数值，则会产生中止错误。如果指定了起始点，则不能省略长度。

- 4) 如果省略起始点和长度，目标转换器进行下列处理。

- (a) 如果汇编的目标设备中包含内部 ROM，则目标转换器认为起始点和长度取值为内部 ROM 中指定的值。
- (b) 如果汇编的目标设备中不包含内部 ROM，则目标转换器认为产生错误，中止执行。

- 5) 如果指定选项 -U 两次（或更多次），则最后指定的一项优先。

- 6) 不能用选项 -U 指定两个（或更多）地址区。

- 7) 下面是选项 -U 中起始点和长度的指定格式及其解释。

- (a) -U 补充值

如果目标设备中包含内部 ROM，则为内部 ROM 区域

如果目标设备中不包含内部 ROM，中止错误

- (b) -U 补充值, 长度

从地址 0 至长度地址

- (c) -U 补充值, 起始, 长度

从起始地址至长度地址

[使用示例]

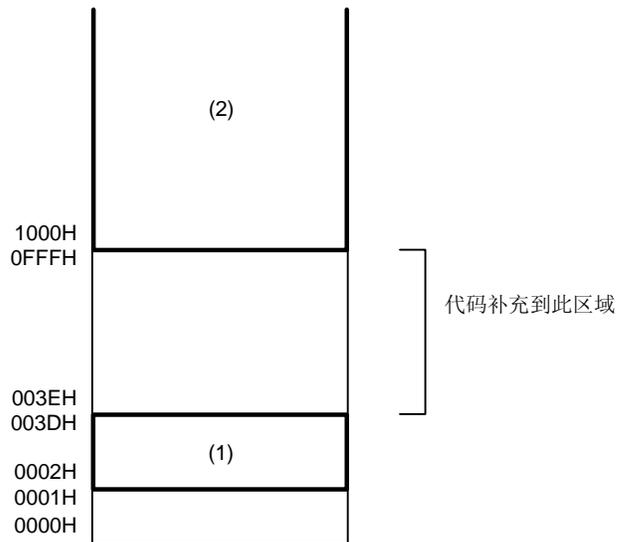
补充 HEX 格式对象代码尚未输出的地址区。

在下面的例子中，假定存在十六进制格式目标文件。在这种情况下，代码无法写入地址区 003EH 至 0FFFH。

```

:020000000200FC
:100002002B41000BFC80FE2B40000944F7083A20EC
:100012001A6720FE2822006521FED350D25014FE1A
:10002200B900059F2835002431B900059F28350005
:0C003200242156AF0A8302A807A830560C
:01000003B5D0d0026A3...
:1010100024A5F622B667...

:
:00000001FF
    
```



00H 补充到地址区 003EH 至 0FFFH 中。

C>>oc78k0s k0s.lmf -u00h.003eh.0fc2h

(5) 错误列表文件输出指定 (-E/-NE)

语法: -E [输出文件名]
 :
D 默认预设: -NE

[功能]

- 1) 选项 -E 指定错误列表文件的输出。选项 -E 还指定输出目标和输出文件名。
- 2) 选项 -NE 使选项 -E 不可用。

[应用]

指定选项 -E 来更改错误列表文件的输出目标和输出文件名。

[说明]

- 1) 错误列表文件的文件名可指定为磁盘型文件名或设备型文件名。但是，若指定设备型文件名 **CLOCK** 则会产生中止错误。
- 2) 当指定选项 -E 且省略输出文件名时，错误列表文件名将是‘输入文件名.EOC’。
- 3) 当指定选项 -E 且省略驱动器名时，错误列表文件将输出到当前驱动器。
- 4) 如果同时指定选项 -E 和-NE，则最后指定的选项优先。

[使用示例]

创建错误列表文件（K0S.EOC）。

```
C>oc78k0s k0s.lmf -ek0s.eoc
```

(6) 参数文件指定 (-F)

语法: -F 文件名
默认预设: 选项和输入文件名只能从命令行指定。

[功能]

选项 -F 指定从指定文件输入的选项和输入文件名。

[应用]

- 1) 当启动目标转换器所需数据不适合命令行时，应指定选项 -F。
- 2) 指定选项 -F 用来在每次进行对象转换时重复指定相同的选项，并将这些选项保存到参数文件中。

[说明]

- 1) 只有磁盘型文件名可指定为‘文件名’。若指定设备型文件名，则会产生中止错误。
- 2) 如果省略文件名，会产生中止错误。
- 3) 不允许参数文件的嵌套。如果在参数文件内指定选项 -F，会产生中止错误。
- 4) 参数文件中可写入的字符数无限制。
- 5) 用空格、制表符或[-]隔开选项或输入文件名。
- 6) 在命令行指定给参数文件的位置处，参数文件中写入的选项和输入文件名将进行扩展。
- 7) 最后指定的扩展选项优先。
- 8) 所有在‘;’或‘#’后、[-]或‘EOF（文件结束）’前键入的字符均当作注释。
- 9) 如果指定选项 -F 两次（或更多次），会产生中止错误。

[使用示例]

用参数文件进行对象转换。

设置参数文件 K0S.POC 的内容，如下。

```
;parameter file  
k0s.lmf -osample.hex  
-ssample.sym -r
```

在命令行中键入下列内容。

```
C>>oc78k0s k0s.lmf -fk0s.poc
```

(7) 设备文件搜索路径指定 (-Y)

- 语法: -Y 路径名
- 默认预设: 设备文件将从下列次序确定的路径读取。
- 1) <..\dev> (oc78k0s.exe 起始路径)
 - 2) OC78K0S 的启动路径
 - 3) 当前目录
 - 4) 环境变量 PATH

[功能]

从指定路径读取设备文件。

[应用]

指定存在设备文件的路径。

[说明]

- 1) 如果在选项 -Y 之后指定了非路径名, 则产生中止错误。
- 2) 如果在选项 -Y 后省略路径名, 产生中止错误。
- 3) 设备文件的读取路径按照如下次序确定。
 - a. 选项 -Y 指定的路径
 - b. <..\dev> (oc78k0s.exe 起始路径)
 - c. OC78K0S 的启动路径
 - d. 当前目录
 - e. 环境变量 PATH

[使用示例]

将设备文件路径指定为目录 a:\78k0s\dev。

```
C>>oc78k0s k0s.lmf -ya:\78k0s\dev
```

(8) 帮助指定 (--)

语法: --
 默认预设: 无显示

[功能]

选项 -- 在显示屏上显示帮助信息。

[应用]

帮助信息是目标转换器选项说明的列表。在执行目标转换器时参考这些内容。

[说明]

在指定选项 --时，所有其它目标转换器选项均不可用。

[使用示例]

在指定选项 --时，显示屏上输出帮助信息。

```
C>>oc78k0s --
```

```
78K/0S Series Object Converter Vx.xx [xx xxx xx]  

  Copyright (C) NEC Electronics Corporation xxxx, xxxx
```

```
usage : oc78k0s [option [...]] input-file [option [...]]  

  The option is as follows ([ ] means omissible).  

  -ffile        :Input option or input-file name from specified file.  

  -o [file] /-no :Create HEX module file [with specified name] / Not.  

  -s [file] /-ns :Create symbol table file [with specified name] / Not.  

  -e [file] /-ne :Create the error list file [with specified name] / Not.  

  -r/-nr       :Sort HEX object by address / Not.  

  -uvalue [, [start], size] :Fill up HEX object with specified value.  

  -ydirectory  :Set device file search path.  

  --            :Show this message.  

  DEFAULT ASSIGNMENT: -o -s -nr
```

7.5 在 PM+ 中设置选项

本节介绍 PM + 目标转换器选项设置的方法。

7.5.1 选项设置方法

从 PM + 的 [Tools] 菜单选择 [Object Converter Options...] 或点击 ，显示 <Object Converter Options> 对话框。可通过在对话框中输入所需的选项来设置目标转换器选项。

图 7-4 <Object Converter Options> 对话框 (当选择 <<Output1>> 标签时)

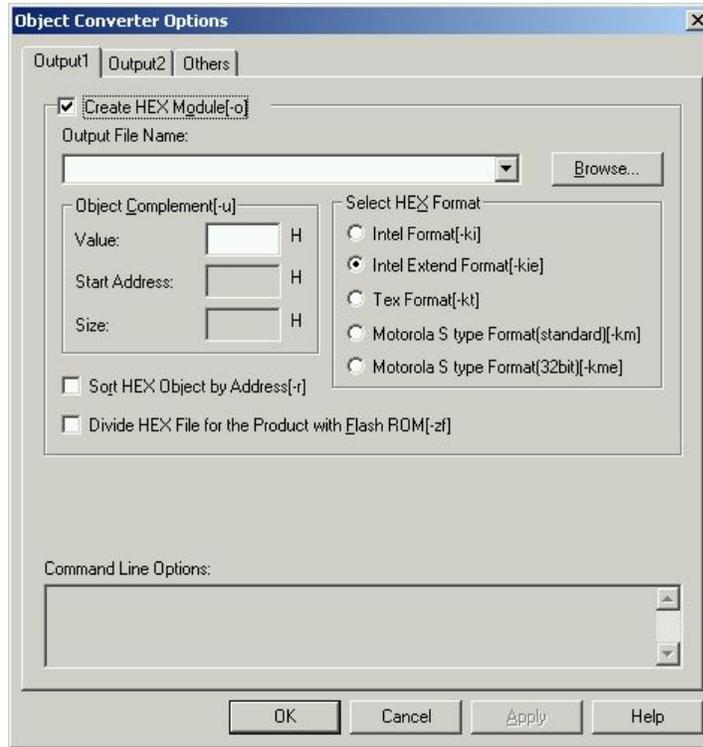


图 7-5 <Object Converter Options> 对话框 (当选择 <<Output2>> 标签时)

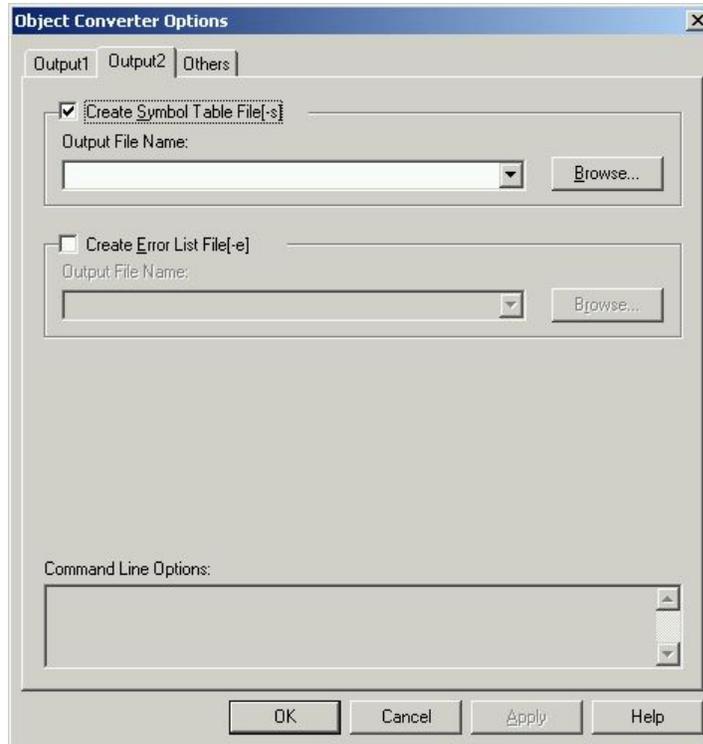
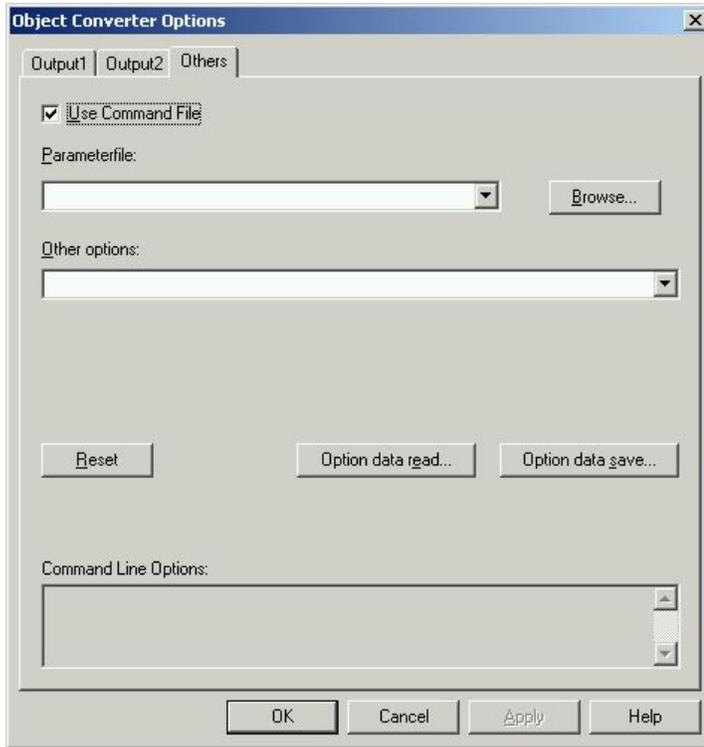


图 7-6 <Object Converter Options> 对话框（当选择 <Others> 标签时）



7.5.2 选项

下面介绍<Object Converter Options>对话框中的各选项。

- **Create HEX module [-o]**
通过[Browse...]按钮或直接输入，指定十六进制格式目标文件的输出路径。
- **Object Complement [-u]**
预先写入代码，防止非法代码写入无 HEX 格式对象输出的地址并防止发生程序循环。
- **Sort HEX Object by Address [-r]**
如果需要用地址排序 HEX 格式对象，应指定此选项。
- **Create Symbol Table File [-s]**
输出文件名:
通过[Browse...]按钮或直接输入，指定符号表文件的输出路径。
- **Create Error List File [-e]**
输出文件名:
要想输出错误列表文件，应在输入框中输入文件名。
指定路径用[Browse...]按钮。
- **Use Command File**
选择此复选框来创建命令文件。
- **Parameter file**
通过[Browse...]按钮或直接输入，读取用户定义的参数文件。
- **Other options**
如果在复选框或单选按钮可选择的选项之外还希望指定其它选项，可在输入框中输入该选项。
- **Reset**
使输入内容复位。
- **Option data read...**
打开 <选项数据读取>对话框，在指定选项数据文件后读取该文件。
- **Option data save...**
打开 <选项数据保存>对话框，将选项数据保存到选项数据文件中并命名。
- **Command Line Options**
此编辑框只读。显示当前设置选项字符串。

第 8 章 库管理程序

库管理程序用于编辑 RA78K0S 目标模块文件和以 1 个模块为单位的库文件。

库管理程序同时也输出列表文件。

如果有库管理程序错误发生，则就会有错误信息输出显示错误产生的原因。

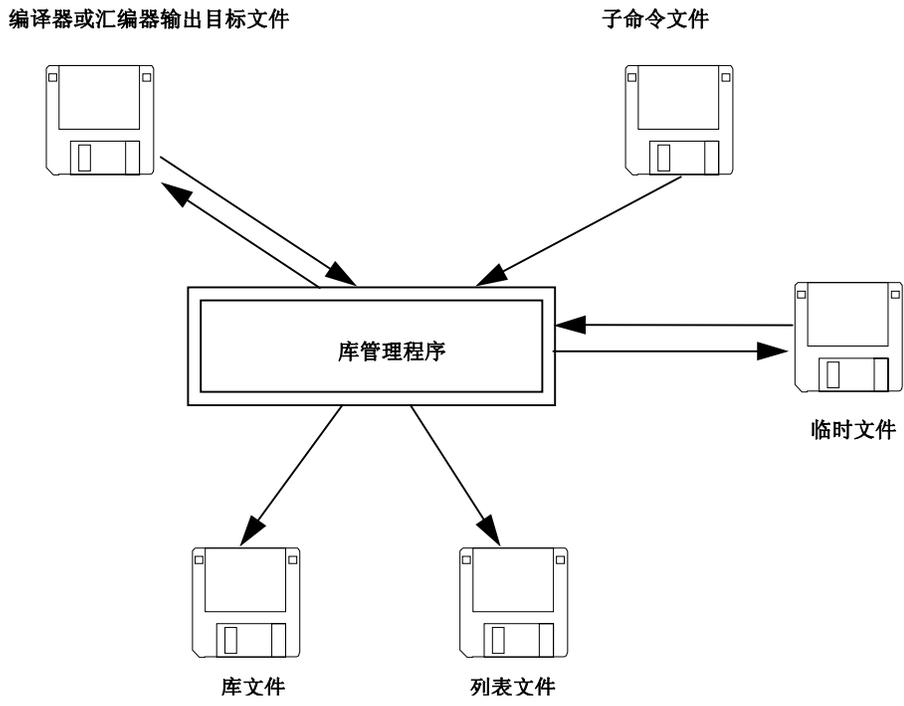
8.1 库管理程序的 I/O 文件

库管理程序的 I/O 文件显示如下。

表 8-1. 库管理程序的 I/O 文件

类型	文件名	说明	缺省的文件类型
输入文件	子命令文件	<ul style="list-style-type: none">• 这些文件中包含了执行程序命令及参数• 这些文件是由用户创建的。	None
输出文件	列表文件	<ul style="list-style-type: none">• 这些文件是库文件数据输出结果。	.LST
I/O 文件	目标模块文件	<ul style="list-style-type: none">• 这些文件是由编译器或编译器输出的目标模块文件。	.REL
	库文件	<ul style="list-style-type: none">• 这些文件输入由库管理程序输出的库文件并更新其内容。	.LIB
	临时文件	<ul style="list-style-type: none">• 这些文件是在生成库时由库管理程序自动产生的。当该库管理程序执行完成之后，这些临时文件也就删除了。	Lbxxxxxx.\$y (y = 1 至 6)

图 8-1. 库管理程序的 I/O 文件



8.2 库管理程序的功能

(1) 生成模块库

汇编器及连接器为其输出的每一个模块都创建一个文件。

这就意味着如果所创建模块的数量巨大，那么与模块所对应的文件数也会大量增加。因此，RA78K0S 中有一个文件用来将这些目标模块收集到一个单一的文件中，这种功能就是所谓的模块目标成生。而作为库组织的文件称为库文件。

库文件可以输入至连接器。当进行模块化编成时，用户可以通过创建由多程序通用模块组成的库文件，方便而有效地对程序进行管理和操作。

(2) 编辑库文件

该库管理文件包含有如下库文件编辑功能。

- 1) 向库文件中添加模块
 - 2) 从库文件中删除模块
 - 3) 替换库文件中模块
 - 4) 从库文件中取出模块
- (需要了解这些功能的详细说明，请参阅 8.5 子命令。)

(3) 库文件数据的输出

该库管理文件包含用于编辑和输出如下的存储于库文件中的数据项的函数。

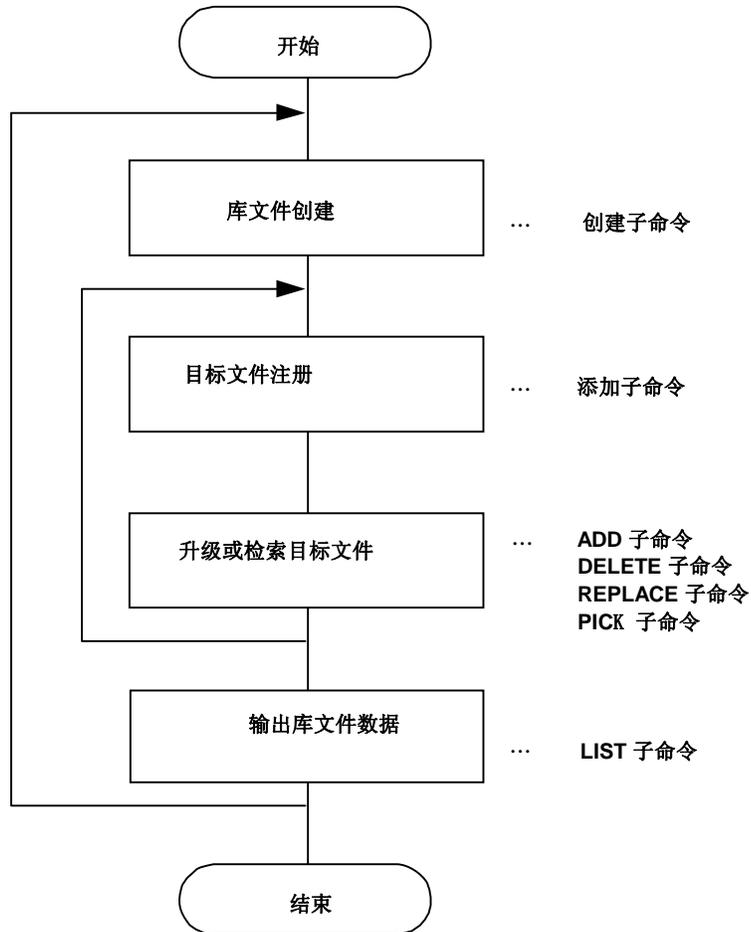
- 1) 模块名
- 2) 所创建的文件
- 3) 注册日期
- 4) 更新日期
- 5) PUBLIC 型符号数据

注意事项 库管理程序使用子命令来执行上述所列的第 2) 和第 3)项功能。在执行处理过程时，由库程序管理器决定每个子命令的执行顺序。如需要了解这些子命令的操作说明，请参阅 8.5 子命令。

(4) 库文件的创建过程步骤

创建库文件的一般过程如下。

图 8-2. 库文件创建过程步骤



8.3 库管理程序的启动

8.3.1 库管理程序的启动

有如下两种方法可以用于库管理程序的启动。

(1) 从命令行启动

X>[路径名] lb78k0s [Δ选项]...

| | | |
(1) (2) (3) (4)

- (1) 当前驱动器名
- (2) 当前目录名
- (3) 库管理程序命令文件名
- (4) 此项包含库管理程序功能的详细说明。

示例 C>lb78k0s -ll20 -lw80

注意事项 如果指定一个以上库管理程序选项，则要通过空格将各个库管理程序隔离开来。需要了解库管理程序的详细说明，请参阅 8.4 库管理程序选项。

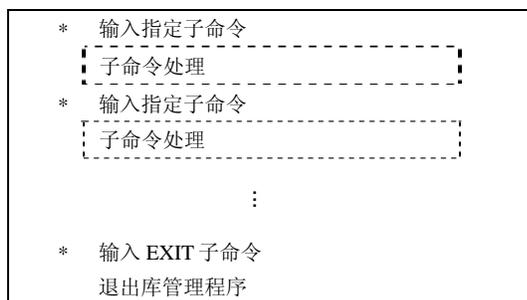
当库管理程序启动时，会显示如下启动信息。

```
78K/0S Series Librarian Vx.xx [xx xxx xx]
Copyright (C) NEC Electronics Corporation xxxx, xxxx
*
```

After an asterisk (*), specify a librarian subcommand.

```
*create k0s.lib
*add k0s.lib k0smain.rel k0ssub.rel
*exit
```

当子命名输入完成时，就开始了每个子命令的处理过程。当完成一个子命令处理时，在屏幕就又显示 '*', 库管理程序等待下一个子命令的输入。库管理程序一直重复这种操作直到有 EXIT 子命令输入为止。



一行最多可以输入 128 个字符。

如果所有需要的操作数一行输入不了时，请使用'&'连接符在下一行接着输入，最多可以连续输入 15 行。

(2) 由子命令文件启动

所谓的子命令文件就是在其中存储库管理程序子命令的文件。

如果在库管理程序启动时没有指定子命令文件，那么必需以星号'*'开头来输入多个子命令。通过创建子命令文件，这些多子命令文件就可以同时一起处理。

每当执行库形成过程，需要重复输入同一个子命令时，也可以使用子命令文件来进行处理。

子命令文件使用方法是，使用时，在文件名前用'<'符号加以描述。

由子命令文件启动库管理程序方法如下。

X>lb78k0sΔ <子命令文件名 [Δ选项]...

(1)	(2)

(1) 在输入指定的子命令文件时，确保一定要在子命令文件名前加入此符号。

(2) 子命令存储在其中的文件

- (a) 使用编辑器创建子命令文件。
- (b) 写入子命令文件内容的规则如下所示。

子命令名	操作数
子命令名	操作数
⋮	
EXIT	

(c) 如果重复执行一个子命令，请在每一行的末尾加'&'符号以示继续执行。

(d) 行末分号(;)后的所有内容均为注释内容，库管理程序命令不对其进行解释。

(e) 如果子命令文件中的最后一个子命令不是 EXIT 子命令，库管理程序会自动将其解释为 EXIT 子命令。

(f) 库管理程序从子命令文件中读取并执行子命令，并在处理完子命令文件中所有的子命令后退出。

示例 用编辑器创建子命令文件(K0S.SLB)。

K0S.SLB 文件内容

```

; library creation command
create k0s.lib
add k0s.lib k0smain.rel &
k0ssub.rel
;
exit

```

使用子命令文件 K0S.SLB 启动库管理程序。

C>lb78k0s <k0s.slb

8.3.2 库管理程序执行开始和执行结束信息

(1) 执行开始信息

当库管理程序启动时，在显示器上就会有一条执行启动信息出现。

```
78K/0S Series Librarian Vx.xx [xx xxx xx]
      Copyright (C) NEC Electronics Corporation xxxx, xxxx
*
```

(2) 执行结束信息

库管理程序不输出执行结束信息。当用户在所有子命令处理完成后输入 EXIT 子命令时，库管理程序将控制权返回至操作系统。

```
*create k0s.lib
*add k0s.lib k0smain.rel k0ssub.rel
*exit
```

如果库管理程序检测到使其无法继续执行处理的致命错误出现时，库管理程序就会向显示器输出一条信息并将控制权返回至操作系统。

示例 指定了并不存在的库管理程序选项

```
C>lb78k0s -a

78K/0S Series Librarian Vx.xx [xx xxx xx]
      Copyright (C) NEC Electronics Corporation xxxx, xxxx
A018 Option is not recognized '-z'
Usage: LB78K0S [options]
```

在上述示例中，输入了一个并不存在的库管理程序选项，导致出错，于是库管理程序中止执行。当有错误显示且库形成过程中止时，请在**第十二章 错误信息**中查找原因并采取相应措施。

8.4 库管理程序选项

8.4.1 库管理程序选项类型

库管理程序选项用于指定列表文件的格式以及临时文件的创建路径。库管理程序选项分为以下四种类型。

表 8-2. 库管理程序选项

序号	分类	选项	说明
1	列表文件格式指定	-LW	更改列表文件里一行可打印的字符数
		-LL	更改列表文件里一页可打印的行数
		-LF	在列表文件末端插入一换页(FF)码
		-NLF	
2	临时文件创建路径指定	-T	在指定路径创建临时文件.
3	设备文件搜索路径指定	-Y	由指定路径读取设备文件
4	帮助信息指定	--	在显示器上显示一帮助信息

备注 本表所示只是库管理程序的简单介绍，实际使用时，请参阅 **C.5 库管理程序选项列表**。

8.4.2 库选项的说明

如下所述是关于库选项的详细说明。

(1) 列表文件格式指定 (-LW, -LL, -LF/-NLF)

(a) -LW

语法: -LW [字符数目]
默认预设 : -LW132 (显示器输出情况下为 80 个字符)

[功能]

选项 -LW 指定在列表文件内一行可输入的字符个数。

[应用]

指定-LW 选项用于更改列表文件内一行可输入的字符个数。

[说明]

- 1) 可用-LW 选项指定的字符数目范围显示如下。
(显示器输出情况下为至多为 80 个字符)

$$72 \leq \text{一行可输入的字符数目} \leq 260$$

如果所指定数字值超出此范围，或指定了一个非数值，则会产生异常中止错误。

- 2) 如果省略了字符数，则会自动指定为 132 个字符，如果列表文件输出至显示器，则会指定为 80 个字符数。
- 3) 所指定的字符数目不包括(CR, LF)等行输入结束字符。
- 4) 如果没有指定 LIST 子命令，可省略-LW 选项。
- 5) 如果两次或多次指定-LW 选项，则最后指定项优先。

[使用示例]

指定列表文件中每行输入字符为 80 个。

```
C>>lb78k0s -lw80
```

(b) -LL

语法: -LL [行数目]
默认预设 : -LL66 (在显示器输出情况下, 没有页分隔符)

[功能]

选项 -LL 指定在列表文件内一页可输入的行数。

[应用]

指定 -LL 选项用于更改列表文件内一页可输入的行数。

[说明]

- 1) 可用 -LL 选项指定的行数目范围显示如下。

$$20 \leq \text{一页可输入的行数} \leq 32767$$

如果所指定数字值超出此范围, 或指定了一个非数字值, 则会产生异常中止错误。

- 2) 如果省略了行数, 则会自动指定为 66 个行数。
- 3) 如果所指定的行数为 0, 则不会设置页分隔符。
- 4) 如果没有指定 LIST 子命令, 可省略 -LL 选项。
- 5) 如果两次或多次指定 -LL 选项, 则最后指定项优先。

[使用示例]

指定列表文件中每页行数为 20 个。

```
C>lb78k0s -ll20
```

(c) -LF/-NLF

语法: -LF
 : -NLF
默认预设 : -NLF

[功能]

- 1) 选项 -LF 指定在列表文件末端插入换页码(FF)。
- 2) 选项-NLF 的作用是使得 -LF 选项不可用。

[应用]

如果希望在列表文件内容输入之后添加一个页分隔符，请指定-LF 选项以添加一个换页码。

[说明]

- 1) 如果没有指定 LIST 子命令，可省略-LF 选项。
- 2) 如果同时指定了-LF 选项和-NLF 选项，则最后指定的选项优先。

[使用示例]

向列表文件添加一个换页码。

```
C>lb78k0s -lf
```

(2) 为临时文件创建指定路径 (-T)

语法: -T 路径名
默认预设 : 创建至由环境变量 TMP 所指定的路径下。
 如果没有指定路径, 则临时文件创建至当前路径下。

[功能]

选项 -T 指定将临时文件创建至指定路径下。

[应用]

使用选项 -T 指定来临时文件创建的位置。

[说明]

- 1) 只有路径才可指定为路径名。
- 2) 路径名不可省略。
- 3) 即使有先前创建的临时文件存在, 如果该文件没有保护, 那么就会被覆盖掉。
- 4) 只要有存放临时文件所需要的内存空间, 临时文件就会在内存中创建而扩展。如果没有足够的内存空间可用, 所创建的临时文件会写入磁盘中去。
这种临时文件稍后可以通过保存磁盘文件而进行访问。
- 5) 当库生成完成后, 临时文件也就删除了。当由于按下(CTRL-C)而引起库生成中止时也会删除临时文件。
- 6) 按照下列顺序来确定待创建的临时文件路径。
 - a. 由选项-T 指定路径
 - b. 由环境变量 TMP 指定路径 (没有省略选项 -T 时)
 - c. 当前路径 (当没有设定 TMP 时)

当指定 a. 或 b. 时, 如果临时文件不能创建至指定路径下, 便会产生异常中止错误。

[使用示例]

指定将临时文件输出至目录 b:\TMP 下。

```
C>lb78k0s -tb:\tmp
```

(3) 设备文件搜索路径指定 (-Y)

语法: -Y 路径名
默认预设 : 设备文件将会由按照如下顺序确定的路径读出。
 1) <..\dev> (用于 lb78k0s.exe 的启动路径)
 2) LB78K0S 的启动路径
 3) 当前目录
 4) 环境变量 PATH

[功能]

由指定路径读取设备文件。

[应用]

指定设备文件所在的路径。

[说明]

- 1) 如果在-Y 选项后指定的不是路径名内容, 便会产生异常中止错误。
- 2) 如果在-Y 选项后省略了路径名, 也会产生异常中止错误。
- 3) 按照下列顺序来确定读取设备文件的路径。
 - a. 由选项-Y 指定路径
 - b. <..\dev> (用于 lb78k0s.exe 的启动路径)
 - c. LB78K0S 的启动路径
 - d. 当前目录
 - e. 环境变量 PATH

[使用示例]

指定设备文件路径为目录 a:\78k0s\dev.

```
C>lb78k0s -ya:\78k0s\dev
```


(1) CREATE

语法: CREATE△库文件名 [△处理]

缩略格式: C

[功能]

CREATE 子命令功能是创建一个新的库文件。

[说明]

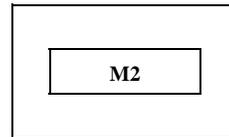
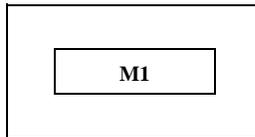
- 1) 创建的库文件大小变为 0。
- 2) 当指定某一处理时，在库文件创建的同时也就完成了模块注册。
- 3) 库文件名: 如果有同名文件存在，就覆盖该同名文件。
- 4) 处理: 如果目标模块文件带有与库文件内相同的公共符号，则该目标模块文件不能注册。
与库文件中模块同名的模块不能注册。
- 5) 如果有错误产生，便中断处理过程，同时也就不能创建库文件了。

[使用示例]

当创建库文件时，同时注册模块 M1 和模块 M2。

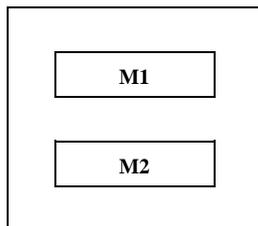
*CREAT k0s.lib m1.rel m2.rel

<文件创建前>



<文件创建后>

K0S.LIB



(2) ADD

语法: ADDΔ库文件名 Δ处理

缩略格式: A

[功能]

ADD 子命令的功能是向库文件中添加一个模块。

[说明]

- 1) 即便库中当前没有存储任何模块，也可以向库中添加模块。
- 2) 如果库文件中已有与欲添加模块同名的模块存在，则会产生错误。
- 3) 如果待添加的模块带有与库文件内相同的公共符号，则会产生错误。

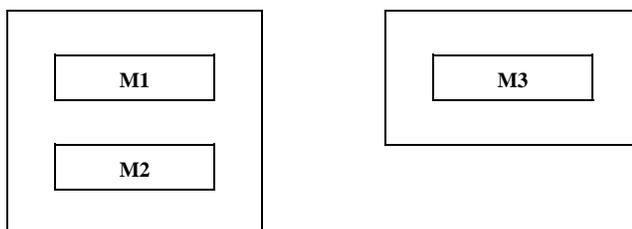
[使用示例]

向库文件(K0S.LIB)中添加一个模块(M3)。

```
*add k0s.lib m3.rel
```

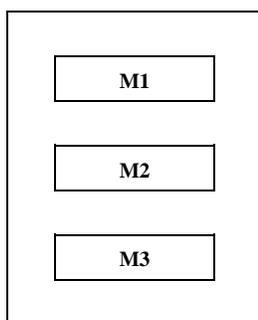
<添加前>

K0S.LIB



<添加后>

K0S.LIB



(3) DELETE

语法: DELETEΔ库文件名 ▽(▽模块名 [▽,...]▽)

缩略格式: D

[功能]

DELETE 子命令的功能是从库文件中删除一个模块。

[说明]

- 1) 如果指定删除的模块在库文件中并不存在, 则会产生错误。
- 2) 如果有错误产生, 就会中断处理过程而且不会改变库文件状况。

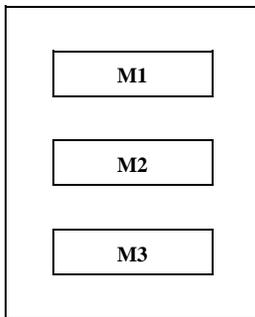
[使用示例]

从库文件(K0S.LIB)中删除模块(M1, M3)。

*delete k0s.lib (m1.rel m3.rel)

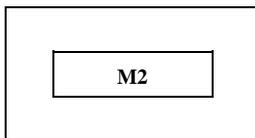
<删除之前>

K0S.LIB



<删除之后>

K0S.LIB



(4) REPLACE

语法: REPLACEΔ库文件名Δ处理

缩略格式: R

[功能]

REPLACE 子命令的功能是用其它目标模块文件替换库文件中的模块。

[说明]

- 1) 如果库文件中没有与替换模块名相同的模块存在, 则会产生错误。
- 2) 如果替换模块中含有与库文件内相同的公共符号, 则会产生错误。
- 3) 替换对象模块的文件名必需与注册变中所使用的文件名相同。
- 4) 如果有错误产生, 就会中断处理过程而且不会改变库文件状况。

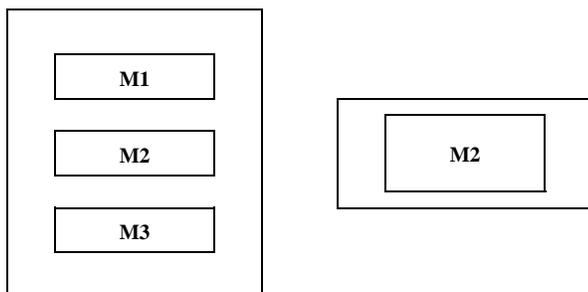
[使用示例]

替换库文件(K0S.LIB)中的一个模块(M2)。

```
*replace k0s.lib m2.rel
```

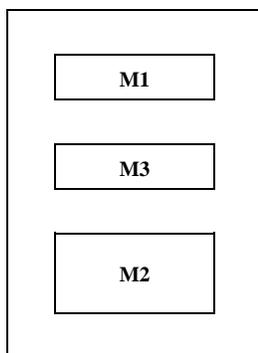
<替换之前>

K0S.LIB



<替换之后>

K0S.LIB



由于在库文件中模块(M2)删除之后便注册了一新的模块(M2), 所以 M2 在库文件中次序为最后一个模块。

(5) PICK

语法: PICKΔ库文件名 ▽(▽模块名 [▽,...]▽)

缩略格式: P

[功能]

PICK 子命令的功能是从现存的库文件中拾取指定的模块。

[说明]

- 1) 被拾取的模块就成为目标模块文件且该目标模块文件名为其在库文件中注册的文件名。
- 2) 如果指定模块名在库文件中并不存在，则会产生错误。
- 3) 错误产生时，处理过程会中断。但是，如果指定两个或多个模块时产生错误，则在导致错误出现的模块之前所拾取的那些模块是可用的而且被保存至磁盘上。

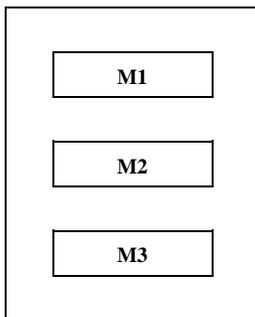
[使用示例]

从库文件(K0S.LIB)中拾取一个模块(M2)。

```
*pick k0s.lib (m2.rel)
```

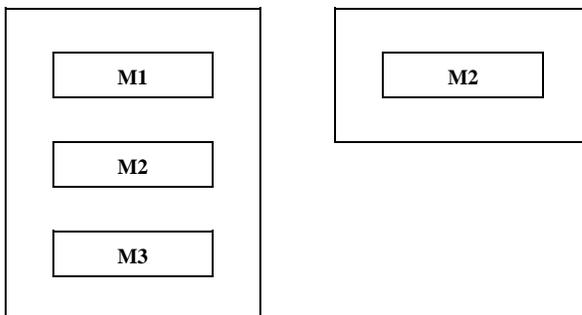
<拾取之前>

K0S.LIB



<拾取之后>

K0S.LIB



(6) LIST

语法: LIST [Δ选项] Δ库文件名 [▽(▽模块名 [▽,...]▽)]

选项: -PUBLIC/-NOPUBLIC
: -O ▽文件名

缩略格式: L

[功能]

LIST 子命令的功能是输出库文件中的模块数据。

[说明]

- 1) 可以指定多个选项。
- 2) -O:
可以将设备类型文件名指定作为输出文件名。
如果省略输出文件名, 则会产生错误。
如果省略文件类型, 那么库管理程序便假定'输入文件名.LST' 作为输入。
- 3) -PUBLIC/-NOPUBLIC:
该选项可以通过仅指定下划线字来进行选定。
-PUBLIC 选项指定公共符号数据的输出。
-NOPUBLIC 选项使得 -PUBLIC 选项不可用。
如果同时指定-PUBLIC 选项 和-NOPUBLIC 选项, 则最后指定的选项优先。

[使用示例]

将库文件(K0S.LIB)中模块数据输出至列表文件 (K0S.LST)中去。指定选项 -P 以便输出公共符号数据。

```
*list -p -ok0s.lst k0s.lib
```

引用列表文件 (K0S.LST) I。

```
78K/0S Series librarian Vx.xx DATE : xx xxx xx PAGE 1
```

```
LIB-FILE NAME : K0S.LIB (xx xxx xx)
```

```
0001 M1.REL (xx xxx xx)
```

```
sym1 sym2 sym3
```

```
NUMBER OF PUBLIC SYMBOLS : 3
```

```
0002 M3.REL (xx xxx xx)
```

```
NUMBER OF PUBLIC SYMBOLS : 0
```

```
0003 M2.REL (xx xxx xx)
```

```
bit1 bit2
```

```
NUMBER OF PUBLIC SYMBOLS : 2
```

(7) HELP

语法: HELP
 缩略格式: H

[功能]

HELP 命令的功能是在显示器上显示帮助信息。

[说明]

显示的帮助信息是一个子命令及其说明列表。在执行库管理程序时，通过指定 HELP 命令或--选项以参考该帮助信息。

[使用示例]

指定 HELP 命令以输出 HELP 信息。

*help

```

+-----+
| Subcommands : create, add, delete, replace, pick, list, help, exit      |
|                                                                           |
| Usage : subcommand[ option]masterLBF[ option]transaction[ option]     |
|                                                                           |
| transaction := OMFname                                                  |
| LBFname [(modulename [...] )]                                         |
|                                                                           |
| <create >:   create masterLBF [ transaction]                          |
| <add >:      add masterLBF transaction                                  |
| <delete >:   delete masterLBF (modulename [...])                       |
| <replace >:  replace masterLBF transaction                             |
| <pick >:    pick masterLBF (modulename [...])                          |
| <list >:    list [ option] masterLBF [(modulename [...])               |
|           option :  -p = output public symbol                          |
|                   -np = no output public symbol                        |
|                   -o filename = specify output file name              |
| <help >:    help                                                         |
| <exit >:    exit                                                         |
+-----+

```

(8) EXIT

语法: EXIT
缩略格式: E

[功能]

EXIT 子命令的功能是退出库管理程序。

[说明]

使用该子命令来退出库管理程序。

[使用示例]

退出库管理程序。

*exit

8.6 在 PM + 中设置选项

本节描述由 PM + 指定库文件的方法。

8.6.1 选项设定方法

在 PM+[Tools] 菜单中的 [Start external tool] 选项中选择 [l lb78k0sp] 或点击  以显示 <Library File Name> 对话框。在指定了路径和文件名后，按 [Next] 以显示 <Subcommand> 对话框。

可以在该对话框中通过输入需要的选项来设置不同库管理程序选项。

图 8-3. <<Library File Name> 对话框

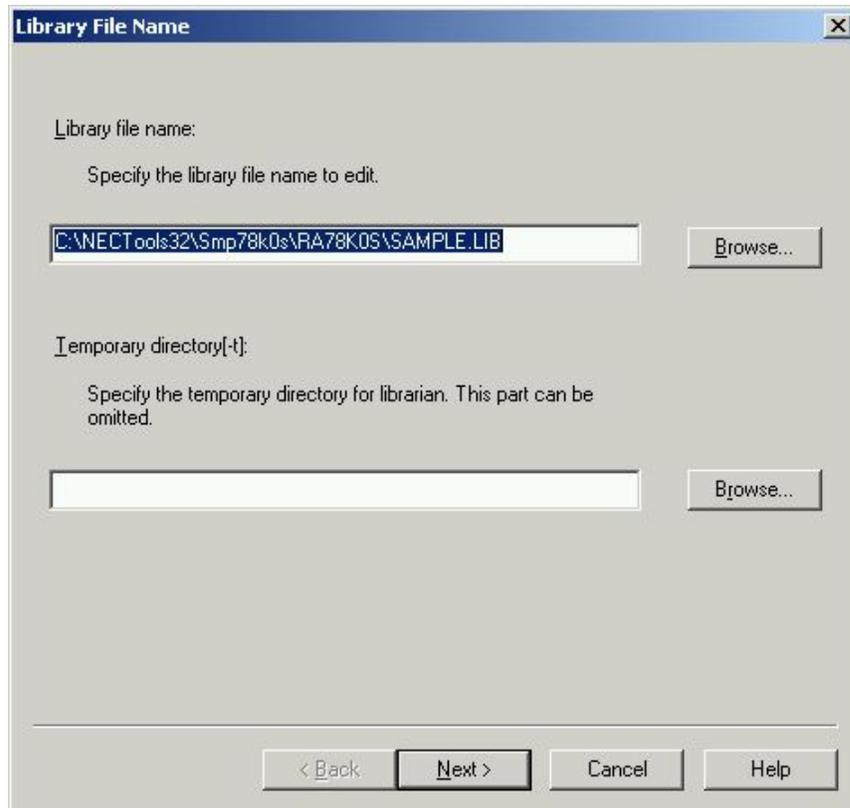
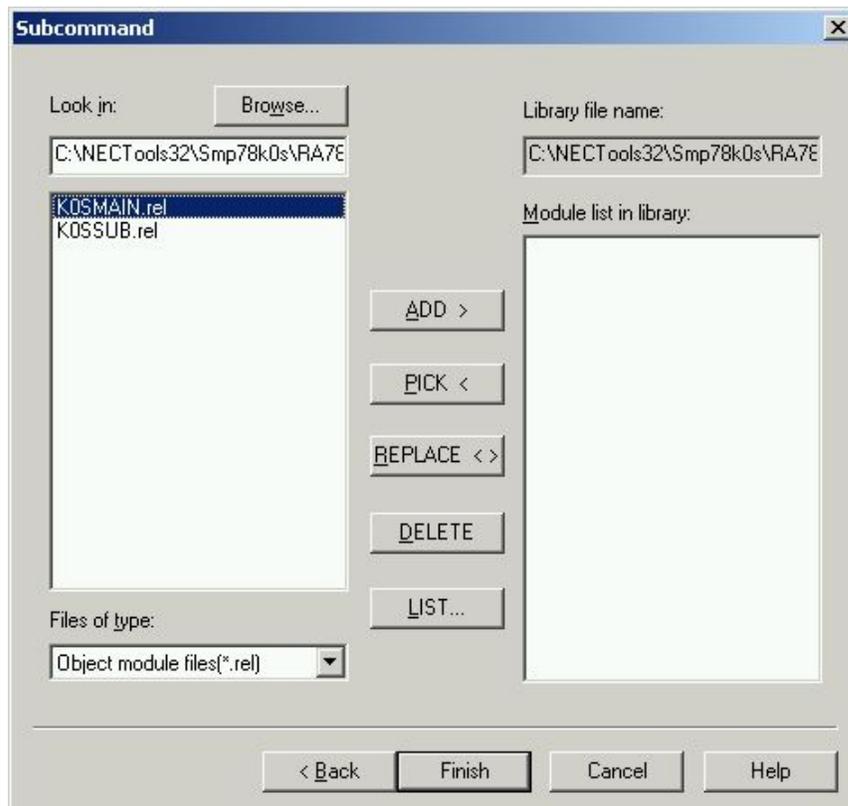


图 8-4. < Subcommand > 对话框



8.6.2 选项

<Librarian>对话框里各种选项描述如下。

- **L**ibrarian file name
通过使用[Browse...]按钮指定需要编辑的库文件路径或直接输入。
- **T**emporary directory [-t]
通过使用[Browse...]按钮指定待创建的临时文件路径或直接输入。
- **L**ook **i**n
通过使用[Browse...]按钮选定要创建至库的目标模块文件或直接输入。
- **A**DD
向现存库文件中添加一个模块。
- **P**ICK
从现存的库文件中拾取指定的模块。
- **R**EPLACE
用其它目标模块文件替换现存库文件模块。
- **D**ELETE
从现存库文件中删除一个模块。
- **L**IST
输出库文件中模块数据。

第 9 章 表转换程序

表转换程序输入汇编列表文件以及由汇编器输出的目标模块文件以及由连接器输出的装载文件。

表转换程序将实际地址嵌入可重定位地址中去，将符号嵌进输入文件中，并输出绝对汇编列表文件。这就可以免除在引用链接映像时需要查看汇编列表这种十分麻烦的工作任务。

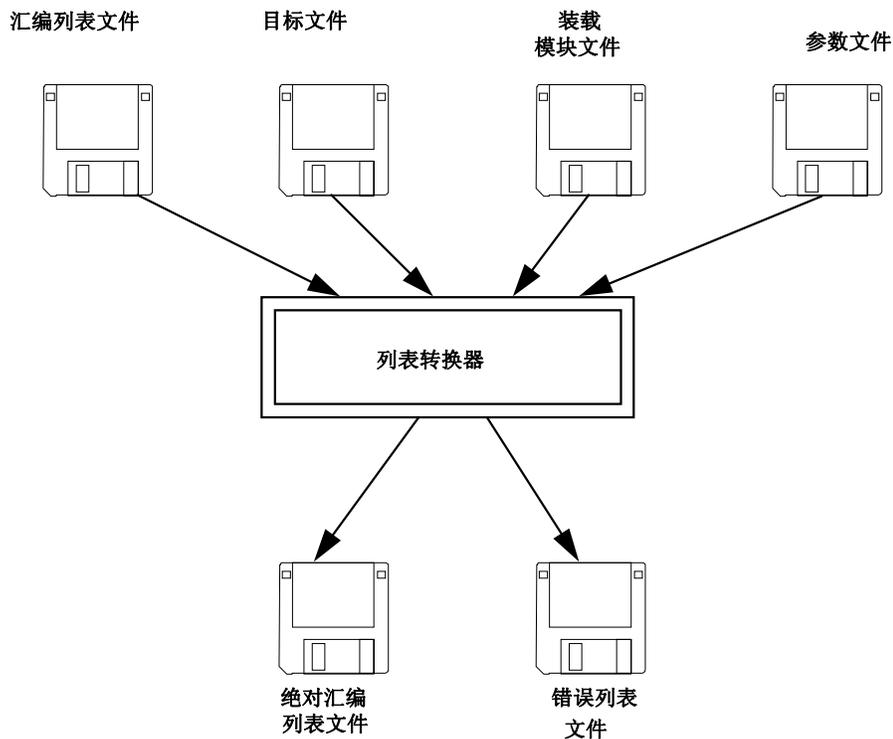
9.1 表转换程序 I/O 文件

表转换程序的 I/O 文件如下所示。

表 9-1. 表转换程序 I/O 文件

类型	文件名	说明	缺省文件类型
输入文件	目标模块文件	<ul style="list-style-type: none"> 这些二进制文件重定位数据和一些关于机器语言数据的符号数据以及机器语言定位地址。 	.REL
	汇编列表文件	<ul style="list-style-type: none"> 这些文件包含诸如汇编列表和交叉参考列表之类的汇编数据 	.PRN
	载入模块文件	<ul style="list-style-type: none"> 这些二进制映像文件包含作为链接结果的对象代码 	.LMF
	参数文件	<ul style="list-style-type: none"> 这些文件包含用于执行程序的参数 这些文件是由用户创建的 	.PLV
输出文件	绝对汇编列表文件	<ul style="list-style-type: none"> 该列表文件将实际地址嵌入可重定位地址中去，将符号嵌进输入文件中 	.P
	错误列表文件	<ul style="list-style-type: none"> 这些文件包含了一些在列表转换期间生成的错误数据 	.ELV

图 9-1. 表转换程序 I/O 文件



9.2 表转换程序功能

下面比较了可重定位汇编程序与绝对汇编程序的优缺点。

[优点]

- 1) 可重定位汇编程序可由多成员组共同开发。
- 2) 可重定位汇编程序可以分成模块以易于开发和存储。
- 4) 可重定位汇编程序适合于开发大规模程序。

[缺点]

- 1) 可重定位汇编程序的汇编列表中的地址与实际物理地址并不一致。
- 2) 可重定位汇编程序的汇编列表中的外部符号值为 0。如果要查找外部符号实际值，还必需参考连接映像。
- 3) 汇编列表中的可重定位数值与实际数值是不同的。

上述这些关于可重定位汇编程序的缺点显著地降低了程序调试和存储的效率，这是由于程序调试和存储需要大量的文档。而表转换程序就为克服可重定位汇编程序的这些缺点提供了一种解决的方法。

- 1) 由表转换程序输出的绝对汇编列表中的地址与实际程序操作所用的地址完全一致。
- 2) 外部符号的实际值嵌入在该列表中。
- 3) 可重定位数值作为实际值嵌入在该列表中。
- 4) 对于符号表或交叉参考表中的符号值，其实际值也嵌入在该列表中。

示例 1. 重定位嵌入

• 汇编列表

21	21	----			CSEG	
22	22	0000		START:		
23	23					
24	24					;芯片初始化
25	25					
26	26	0000	F5201A	MOV	HDTSA, #1AH	
27	27	0003	FC20FE	MOVW	HL, #HDTSA	;在 HL 寄存器中设定十六进制 2 位代码数据
28	28					
29	29	0006	R220000	CALL	!CONVAH	;ASCII<- HEX 转换
30	30					;将 ASCII 码输出至 BC 寄存器中去
31	31	0009	F821FE	MOVW	DE, #STASC	;设置存储 ASCII 码的表地址至 DE
32	32	000C	0A27	MOV	A, B	
33	33	000E	EB	MOV	[DE], A	
34	34	000F	88	INCW	DE	
35	35	0010	0A25	MOV	A, C	
36	36	0012	EB	MOV	[DE], A	

• 绝对汇编列表

21	21	----			CSEG	
22	22	0080		START:		
23	23					
24	24					;芯片初始化
25	25					
26	26	0080	F5201A	MOV	HDTSA, #1AH	
27	27	0083	FC20FE	MOVW	HL, #HDTSA	;在 HL 寄存器中设定十六进制 2 位代码数据
28	28					
29	29	0086	R229500	CALL	!CONVAH	;ASCII<- HEX 转换
30	30					;将 ASCII 码输出至 BC 寄存器中去
31	31	0089	F821FE	MOVW	DE, #STASC ;	设置存储 ASCII 码的表地址至 DE
32	32	008C	0A27	MOV	A, B	
33	33	008E	EB	MOV	[DE], A	
34	34	008F	88	INCW	DE	
35	35	0090	0A25	MOV	A, C	
36	36	0092	EB	MOV	[DE], A	

示例 2. 对象代码的嵌入

• 汇编列表

```

21 21 ---- CSEG
22 22 0000 START:
23 23
24 24                                     ;芯片初始化
25 25
26 26 0000 F5201A MOV HDTSA, #1AH
27 27 0003 FC20FE MOVW HL, #HDTSA ;在 HL 寄存器中设定十六进制 2 位代码数据
28 28
29 29 0006 R220000 CALL !CONVAH ;ASCII<- HEX 转换
30 30 ;将 ASCII 码输出至 BC 寄存器中去
31 31 0009 F821FE MOVW DE, #STASC ;设置存储 ASCII 码的表地址至 DE
32 32 000C 0A27 MOV A, B
33 33 000E EB MOV [DE], A
34 34 000F 88 INCW DE
35 35 0010 0A25 MOV A, C
36 36 0012 EB MOV [DE], A

```

• 绝对汇编列表

```

21 21 ---- CSEG
22 22 0080 START:
23 23
24 24                                     ;芯片初始化
25 25
26 26 0080 F5201A MOV HDTSA, #1AH
27 27 0083 FC20FE MOVW HL, #HDTSA ;在 HL 寄存器中设定十六进制 2 位代码数据
28 28
29 29 0086 R229500 CALL !CONVAH ;ASCII<- HEX 转换
30 30 ;将 ASCII 码输出至 BC 寄存器中去
31 31 0089 F821FE MOVW DE, #STASC ;设置存储 ASCII 码的表地址至 DE
32 32 008C 0A27 MOV A, B
33 33 008E EB MOV [DE], A
34 34 008F 88 INCW DE
35 35 0090 0A25 MOV A, C
36 36 0092 EB MOV [DE], A

```

9.3 表转换程序的启动

9.3.1 表转换程序的启动

有如下两种方法可以用于启动表转换程序。

(1) 由命令行启动

```
X>lc78k0s [Δ选项]...Δ输入文件名 [Δ选项]...[Δ]
```

```
|      |      |      |      |
(1)    (2)    (3)    (4)    (3)
```

- (1) 当前驱动器名
- (2) 表转换程序的命令行名
- (3) 输入用于列表文件操作的详细说明
- (4) 汇编列表的基本名

示例 C>lc78k0s k0smain -lk0s.lmf

- 注意事项**
1. 在上述第(3)项中，当指定了两个或两个以上的表转换程序选项时，要用空格将各个表转换程序选项分割开来。要了解表转换程序选项详细的说明，请参阅 9.4 表转换程序选项。
 2. 对上述第(4)项，请使用扩展名 .PRN。
 3. 对上述第(4)项，只要汇编列表的基本名是在命令行中指定的，那么目标文件和载入模块文件的基本名就必需要与汇编列表文件的基本名相同。
文件类型也必需为如下所示类型。

表 9-2. 当表转换程序启动时指定的文件类型

文件名	类型
目标模块类型	.REL
载入模块文件	.LMF

当指定与基本名不同的文件时请使用选项进行设置。

(2) 由参数文件进行启动

当启动表转换程序所需要的数据不满足命令行数据格式要求时，或者每当执行表转换程序都需要重复执行相同的表转换程序选项时，请使用参数文件启动表转换程序。

请在命令行中指定参数文件选项(-F)，以从参数文件启动表转换程序。

由参数文件启动表转换程序方法如下。

X>lc78k0s [Δ输入文件名] Δ-f 参数文件名

(1)	(2)

(1) 参数文件名设置选项

(2) 该文件包含启动表转换程序时所需要的数据。

备注 使用编辑程序来创建参数文件。

描述参数文件内容的规则如下。

[[[Δ] 选项 [Δ选项] ... [Δ]Δ]]...

- 1) 如果在命令行中省略了输入文件名，那么在参数文件中只能指定 1 个输入文件名。
- 2) 输入文件名也可以在选项后写入。
- 3) 在参数文件中写入所有表转换程序选项及那些应该在命令行中指定的输出文件名。

示例 使用编辑程序创建参数文件 (K0S.PLV)。

K0S.PLV 中的内容

```

;参数文件名
k0smain -lk0s.lmf
-ek0s.elv

```

使用 参数文件名 (K0S.PLV) 启动表转换程序。

```
C>lc78k0s -fk0s.plv
```

9.3.2 表转换程序执行开始和结束信息

(1) 表转换程序执行开始信息

当表转换程序启动时，在显示器上会显示一条执行启动信息。

```
List Conversion Program for RA78K/0S Vx.xx [xx xxx xx]  
Copyright (C) NEC Electronics Corporation xxxx
```

```
Pass1: start...
```

```
Pass2: start...
```

(2) 表转换程序执行结束信息

如果没有检测到列表转换错误发生，表转换程序会向显示器输出如下信息并将控制权返回至操作系统。

```
Conversion complete
```

如果表转换程序检测到使其无法继续执行列表转换处理的致命错误出现时，列表转换就会向显示器输出一条信息并取消列表转，将控制权返回至操作系统。

示例 指定了并不存在的表转换程序选项

```
C>>lc78k0s k0smain -a
```

```
List Conversion Program for RA78K/0S Vx.xx [xx xxx xx]  
Copyright (C) NEC Electronics Corporation xxxx
```

```
A018 Option is not recognized '-a'
```

```
Program aborted
```

当表转换程序输出错误信息并中止列表转换执行时，请在**第十二章 错误信息**中查找原因并采取相应措施。

9.4 表转换程序选项

9.4.1 表转换程序选项类型

这些表转换程序选项为表转换程序操作的详细说明，表转换程序选项分为以下六种类型。

表 9-3. 表转换程序选项

序号 r	分类	选项	说明
1	目标文件输入指定	-R	输入目标文件
2	载入模块文件输入指定	-L	输入载入模块文件
3	绝对汇编列表文件输出指定	-O	输出绝对汇编列表文件
4	错误列表文件输出指定	-E	输出错误列表文件
5	参数文件名指定	-F	输入文件名和指定文件选项
6	帮助信息显示指定	--	在显示器上显示帮助信息

备注 需要了解表转换程序的详细内容，请参阅 C.6 表转换程序选项。

9.4.2 表转换程序选项的说明

本节包含每一个表转换程序选项的详细说明。

(1) 目标文件输入指定 (-R)

语法: -R [输入文件名]
默认预设: -R 汇编列表文件名.REL

[功能]

选项 -R 的功能是指定输入目标模块文件

[应用]

当目标模块文件的基本名与汇编列表文件里的基本名不同时，或者其文件类型不是".REL"时，请指定-R 选项。

[说明]

- 1) 如果有致命错误发生，不能输出绝对汇编列表文件。
- 2) 如果只指定输入文件名的基本名，则表转换程序将指定该文件类型为'.REL' 并输入该文件。

[使用示例]

汇编列表文件名为 K0SMAIN.PRN，目标模块文件名为 K0SMAIN.PRN 以及装载文件名为 K0S.LMF。

```
C>lc78k0s k0smain -rsample.rel -lk0s.lmf
```

(2) 载入模块文件输入指定 (-L)

语法: -L [输入文件名]
默认预设: -L 汇编列表文件名.LMF

[功能]

选项 -L 的功能是指定载入模块文件的输入。

[应用]

当载入模块文件的基本名与汇编列表文件里的基本名不同时，或者其文件类型不是".LMF"时，请指定-L 选项。

[说明]

- 1) 如果有致命错误发生，不能输出绝对汇编列表文件。
- 2) 如果只指定输入文件名的基本名，则表转换程序将指定该文件类型为'.LMF' 并输入该文件。

[使用示例]

汇编列表文件名为 K0SMAN.PRN，装载文件名为 SAMPLE.LMF。

```
C>lc78k0s k0smain -lsample.lmf
```

(3) 绝对汇编列表文件输出指定 (-O)

语法: -O [输出文件名]
默认预设: -O 汇编列表文件名.P

[功能]

选项 -O 功能是指定输出绝对汇编列表文件。同时选项 -O 也可以指定输出的目标文件和输出文件名。

[应用]

使用选项 -O 来更改输出目标文件和绝对汇编列表文件的输出文件名。

[说明]

- 1) 文件名可以指定为磁盘型文件名或设备型文件名。但是只有 CON, PRN, NUL 及 AUX 可以指定作为设备型文件名。如果指定了 CLOCK 型, 则会发生异常中止错误。
- 2) 如果同一个设备既指定为文件名又指定为错误文件名, 则会发生异常中止错误。
- 3) 如果在指定 -O 选项时省略了输出文件名, 则绝对汇编列表文件名便为'汇编列表文件名'.P'。
- 4) 如果只指定输入文件名的基本名, 则表转换程序将指定该文件类型为'.P'并输入该文件。
- 5) 如果在指定 -O 选项时省略了驱动器名, 则绝对汇编列表文件将被输出到当前驱动器。

[使用示例]

创建绝对汇编列表文件 (SAMPLE.P)。

```
C>>lc78k0s k0smain -osample.p -lk0s.lmf
```

(4) 错误列表文件输出指定 (-E/-NE)

语法: -E [输出文件名]
 : -NE
默认预设: -NE

[功能]

- 1) 指定选项-E用以指定输出错误列表文件。同时该选项也可以指定输出的目标文件和输出文件名。
- 2) 选项-NE的作用是使得-E选项不可用。

[应用]

指定选项-E用以将错误信息保存至文件中去。

[说明]

- 1) 错误列表文件名可以被指定为磁盘型文件名或设备型文件名。但是如果指定为CLOCK设备型文件名型,则会发生异常中止错误。
- 2) 如果同一个设备既指定为文件名又指定为绝对汇编列表文件名,则会发生异常中止错误。
- 3) 如果在指定-E选项时省略了输出文件名,则错误列表文件名便为'汇编列表文件名.ELV'。
- 4) 如果只指定输出文件名的基本名,则表转换程序将指定该文件类型为'.ELV'并输出该文件。
- 5) 如果在指定-E选项时省略了驱动器名,则错误列表文件将被输出到当前驱动器。
- 6) 如果同时指定了-E选项和-NE选项,则最后指定的选项优先。

[使用示例]

示例 创建一个错误列表文件(SAMPLE.ELV)。

```
C>>lc78k0s k0smain -esample.elv
```

引用错误列表文件(SAMPLE.ELV)。

```
*** WARNING W101 Load module file is older than object module file'K0SMMAIN.LMF,  
K0SMMAIN.REL'  
Pass1: start  
*** ERROR A105 Segment name is not found in load module file 'DATA'
```

(5) 参数文件名指定 (-F)

语法: -F 文件名
默认预设: 只能在命令行中输入选项和输入文件名。

[功能]

选项 -F 的功能是指定输入选项和由指定文件中输入文件名。

[应用]

- 1) 当启动表转换程序所需要的数据不满足命令行数据格式要求时, 请指定选项 -F。
- 2) 每当执行表转换程序需要重复执行相同的表转换程序选项时, 请在参数列表文件中对这些选项进行描述并指定选项-F。

[说明]

- 1) 只有磁盘型文件名可以指定为 '文件名', 如果指定了设备型文件名, 则会发生异常中止错误。
- 2) 如果省略了文件名, 则会发生异常中止错误。
- 3) 如果只指定了文件名的基本名, 则表转换程序将指定该文件类型为'PLV'并打开该文件。
- 4) 不允许参数文件嵌套。如果在参数文件中指定选项-F, 则会发生异常中止错误。
- 5) 可写入参数文件的字符数目没有限制约束。
- 6) 用空格符, 制表符或回车符[↵]将选项或输入文件名隔离开来。
- 7) 写入至参数文件中的选项和输入文件名扩展存放在由命令行所指定的参数文件处。
- 8) 最后指定的扩展选项优先。
- 9) 如果选项-F 被两次或多次指定, 则会发生异常中止错误。

[使用示例]

使用参数文件启动表转换程序。

参数文件(K0S.PLV)的内容如下。

```
; 参数文件  
k0smain -lk0s.lmf  
-ek0s.elv
```

在命令行中输入如下命令。

```
C>>lc78k0s -fk0s.plv
```

(6) 帮助信息显示指定 (--)

语法: --
 默认预设: 无显示

[功能]

选项 -- 在显示器上显示帮助信息

[应用]

显示的帮助信息为表转换程序选项的说明列表。在执行表转换程序时请参阅这些说明。

[说明]

当指定了--选项时，所有其它表转换程序选项均不可用。

[使用示例]

当指定了 -- 选项时，帮助信息便输出至显示器。

```
C>lc78k0s --
```

```
List Conversion Program for RA78K/0S Vx.xx [xx xxx xx]
Copyright (C) NEC Electronics Corporation xxxx, xxxx
```

```
usage : LC78K0S [option [...]] input-file [option [...]]
The option is as follows ([ ] means omissible).
-r [file] :Specify object module file.
-l [file] :Specify load module file.
-o [file] :Specify output list file (absolute assemble list file).
-f [file] :Input option or input-file name from specified file.
-e [file] :Create error list file.
-- :     Show this message.
```

9.5 在 PM + 中设置选项

本节描述由 PM + 设定表转换程序选项的方法。

9.5.1 选项设置方法

在 PM +[Tools]菜单中选择[List converter options...]选项，或点击  以显示 <List Converter Options> 对话框。可以在该对话框中通过输入需要的选项来设置表转换程序选项。

图 9-2. <List Converter Options> 对话框 (当选择了 <<Output>> Tab 时)

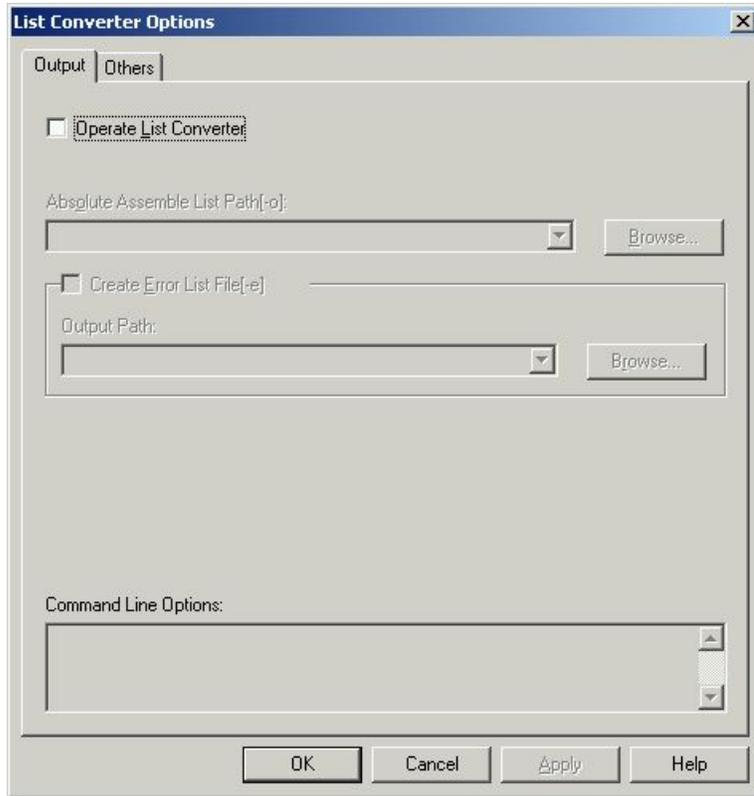
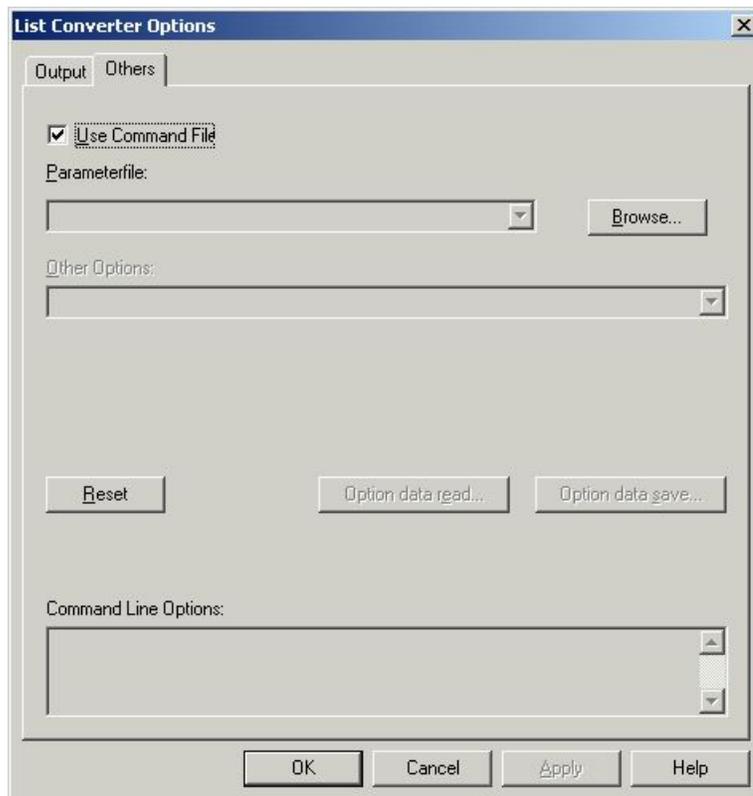


图 9-3. <List Converter Options> 对话框 (当选择了 <<Others>> Tab 时)



9.5.2 选项

< List Converter Options >对话框里各种选项描述如下。

- **Operate List Converter**
选中该复选框以启动表转换程序。
- **Absolute Assemble List Path [-o]**
通过使用[Browse...]按钮指定绝对汇编列表路径或直接输入。
- **Create Error List File [-e]**
输出路径
输出错误列表文件，在输入对话框中输入文件名。
使用[Browse...]按钮指定路径。
- **Use Command File**
选中该复选框以创建命令文件。
- **Parameterfile**
通过使用[Browse...]按钮指定读入所选中的用户定义参数文件或直接输入。
- **Other Options**
如果希望指定用复选框或单选按钮选择之外的选项，请在输入框中输入要指定的选项。
- **Reset**
复位输入内容。
- **Option data read...**
打开 <Option Data Read> 对话框并在选项数据文件制定后，读入该文件。
- **Option data save...**
打开<Option Data Save> 对话框，将选项数据保存至选项数据文件中去并为之命名。
- **Command Line Options**
该编辑框是只读的，显示当前设置选项字符串。

第 10 章 程序输出列表

以下是关于各功能程序输出列表的格式说明及一些其它信息。

- 汇编器输出表
 - 汇编列表文件标题
 - 汇编列表
 - 符号列表
 - 交叉参考列表
 - 错误列表

- 连接器输出列表
 - 连接列表文件标题
 - 映像列表
 - 公共符号列表
 - 错误列表

- 目标转换器输出列表
 - 错误列表

- 库管理程序输出列表
 - 库数据输出列表

- 列表转换器输出列表
 - 绝对汇编列表
 - 错误列表

10.1 汇编器输出列表

汇编器输出如下列表。

输出列表文件名称	输出列表名称
汇编列表文件	汇编列表
	符号列表
	交叉参考列表
错误列表文件	错误列表

10.1.1 汇编列表文件标题

在汇编列表文件的起始处总会输出该标题。

[输出格式]

78K/0S Series Assembler (1)Vx.xx (2) Date:(3)xx xxx xxxx Page:(4) 1

(5)

Command: (6)k0smain.asm -c9024

Para-file:(7)

In-file: (8)K0SMMAIN.ASM

Obj-file: (9)K0SMMAIN.REL

Prn-file: (10)K0SMMAIN.PRN

[各输出项的说明]

条目	详细说明
(1)	汇编器版本号
(2)	标题字符串 由选项-LH 或控制指令 TITLE 所指定的字符串
(3)	汇编列表创建的日期
(4)	页码号
(5)	次标题字符串 由控制指令 SUBTITLE 所指定的字符串
(6)	命令行映像
(7)	参数文件的内容
(8)	输入源模块文件名
(9)	输出目标模块文件名
(10)	汇编列表文件名

10.1.2 汇编列表

汇编列表输出附带错误信息(如果有错误产生的话)的汇编结果。

[输出格式]

Assemble list

```

ALNO STNO ADRS  OBJECT M I SOURCE STATEMENT
      (3)(4)
(1)1 (2)1
(2)2 (2)2      (5) NAME  SAMPM
      :
28 28
29 29 (6)0006 (8)R220000 (5)      CALL !CONVAH ;ASCII<- HEX 转换
30 30      (5)      ;将 ASCII 码输出至 BC 寄存器中去
31 31 (6)0009 00000000      MOV DE, #STASC ;设置存储 ASCII 码的表地址至 DE
(7)** ERROR F202, STNO 31 ( 0)非法操作数
      (6)000D 00
32 32 (6)000E (8)0A27      (5)      MOV A, B
33 33 (6)0010 (8)EB      (5)      MOV [DE], A
      :
Segment informations:

ADRS  LEN  NAME

(9)FE20 (10)0003H (11)DATA
(9)0000 (10)0002H (11)CODE
(9)0000 (10)0017H (11)?CSEG

Target chip:(12)uPD78xxxx
Device file:(13)Vx.xx
Assembly complete,      (14) 1 error(s) and (15) 0 warning(s) found.      ( (16) 31)

```

[各输出项的说明]

条目	详细说明
(1)	源模块映像行号
(2)	行号 (包括 INCLUDE 文件的扩展及宏)
(3)	宏显示 M: 为宏定义行 #n: 为宏扩展行, n 为嵌套级数 Blank: 不是宏定义行或宏扩展行
(4)	INCLUDE 显示 In: 在 INCLUDE 文件内, n 为嵌套级数 Blank: 没有使用 INCLUDE 文件。
(5)	源程序语句
(6)	单元数
(7)	有致命错误/警告产生行
(8)	重定位数据 R: 由连接器更改的目标代码或符号值 Blank: 目标代码或符号值没有被连接器更改
(9)	段地址
(10)	段大小
(11)	段名
(12)	RA78K0S 目标设备
(13)	设备文件版本号
(14)	致命错误数目
(15)	警告错误数目
(16)	最后出错行

10.1.4 交叉参考列表

交叉参考表输出的数据指示符号是在源模块中哪里(在哪一行)定义的。

[输出格式]

交叉参考列表

NAME	VALUE	R	ATTR	RTYP	SEGNAME	XREFS
(1)?CSEG			(4)CSEG		(6)?CSEG	(7)21#
(1)CODE			(4)CSEG		(6)CODE	(7)18#
(1)CONVAH	(2)---H	(3)E		(5)EXT		(7)12 29
(1)DATA			(4)DSEG		(6)DATA	(7)14#
(1)HD TSA	(2)FE20H		(4)ADDR		(6)DATA	(7)15# 26 27
(1)MAIN	(2)0H		(4)ADDR	(5)PUB	(6)CODE	(7)11@ 19#
(1)SAMP M			(4)MOD			(7)2#
(1)START	(2)0H	(3)R	(4)ADDR	(5)PUB	(6)?CSEG	(7)11@ 19 22#
(1)STASC	(2)FE21H		(4)ADDR		(6)DATA	(7)16# 31

[各输出项的说明]

条目	详细说明
(1)	定义的符号名
(2)	符号值
(3)	重定位属性 R: 可重定位符号 E: 外部符号 Blank: 绝对符号 *: 未定义的符号
(4)	符号属性 CSEG: 代码段名称 DSEG: 数据段名称 BSEG: 位段名称 MAC: 宏名称 MOD: 模块名称 SET: 由 SET 指令定义的符号 NUM: NUMBER 属性符号 ADDR: ADDRESS 属性符号 SABIT: BIT 属性符号(saddr.bit) SFBIT: BIT 属性符号(sfr.bit) RBIT: BIT 属性符号(A.bit, X.bit) Blank: 由 EXTRN 或 EXTBIT 声明的外部参考符号 *****: 未定义的符号
(5)	符号参考格式 EXT: 由 EXTRN(SADDR 属性) 声明的外部参考符号 EXTB: EXTBIT (saddr.bit)声明的外部参考符号 PUB: 由 PUBLIC 声明的外部定义符号 Blank: 局部符号, 段名称, 宏名称, 模块名称 *****: 未定义的符号
(6)	定义的符号名
(7)	定义/参考行号 定义行: XXXXX# 参考行: XXXXX∇ (∇= 1 空) EXTRN 声明, EXTBIT 声明, PUBLIC 声明: XXXXX@

10.1.5 错误列表

错误列表存储汇编器启动时输出的错误信息。

[输出格式]

Pass1 Start

(1)ERROR.ASM ((2)26) :(3)F202(4)非法操作数

(1)ERROR.ASM ((2)32) :(3)F202(4)非法操作数

Pass2 Start

(1)ERROR.ASM ((2)26) :(3)F202(4)非法操作数

(1)ERROR.ASM ((2)29) :(3)F407(4)未定义的符号参考 ' DTSA'

(1)ERROR.ASM ((2)29) :(3)F303(4)非法表达式

(1)ERROR.ASM ((2)32) :(3)F202(4)非法操作数

(1)ERROR.ASM ((2)37) :(3)F407(4)未定义的符号参考 ' F'

(1)ERROR.ASM ((2)37) :(3)F303(4)非法表达式

[各输出项的说明]

条目	详细说明
(1)	有错误产生的源模块名称
(2)	错误产生行
(3)	错误号
(4)	错误信息

10.2 连接器输出列表

连接器输出如下一些列表。

输出列表文件名称	输出列表名称
连接列表文件	映像列表
	公共符号列表
	局部符号列表

10.2.1 连接列表文件标题

总会在连接列表文件的起始处输出标题。

[输出格式]

78K/OS Series Linker (1)Vx.xx Date:(2)xx xxx xxxx Page:(3)1

Command: (4)k0smain.rel k0ssub.rel -ok0s.map -dk0s.dr

Para-file: (5)

Out-file: (6)K0S.LMF

Map-file: (7)K0SMMAIN.MAP

Direc-file: (8)

Directive: (9)

*** Link information ***

(10) 3 output segment(s)

(11) 37H byte(s) real data

(12) 23 symbol(s) defined

[各输出项的说明]

条目	详细说明
(1)	连接器版本号
(2)	连接列表文件创建的日期
(3)	页码号
(4)	命令行映像
(5)	参数文件的内容
(6)	输出载入模块文件名
(7)	连接列表文件名
(8)	指令文件名
(9)	指令文件内容
(10)	输出至载入模块文件的段的数目
(11)	输出至载入模块文件的数据的大小
(12)	输出至载入模块文件的符号数

10.2.2 映像列表

映像列表输出段存储单元数据。

[输出格式]

*** 内存映像 ***

(1) SPACE = REGULAR

MEMORY =(2)ROM

BASE ADDRESS =(3)0000H SIZE =(4)2000H

OUTPUT SEGMENT	INPUT SEGMENT	INPUT SEGMENT	BASE	SIZE MODULE	ADDRESS
(6)CODE			(9)0000H	(10)0002H	(11)CSEG AT
(5)* gap *	(7)CODE	(8)SAMPM	(9)0000H	(10)0002H	
(6)?CSEG			(9)0002H	(10)007EH	
	(7)?CSEG	(8)SAMPM	(9)0080H	(10)0035H	(11)CSEG
	(7)?CSEG	(8)SAMPS	(9)0080H	(10)0015H	
(5)* gap *			(9)0095H	(10)0020H	
			(9)00B5H	(10)1F4BH	

MEMORY = RAM

BASE ADDRESS = (3)FE00H SIZE =(4)0200H

OUTPUT SEGMENT	INPUT SEGMENT	INPUT SEGMENT	BASE	SIZE MODULE	ADDRESS
(5)* gap *			(9)FE00H	(10)0020H	
(6)DATA			(9)FE20H	(10)0003H	(11)DSEG AT
	(7)DATA	(8)SAMPM	(9)FE20H	(10)0003H	
(5)* gap *			(9)FE23H	(10)00DDH	
(5)* gap (Not Free Area) *			(9)FE00H	(10)0100H	

Target chip:(12)uPD78xxx

Device file:(13)Vx.xx

[各输出项的说明]

条目	详细说明
(1)	存储空间名称
(2)	存储区域名称
(3)	存储区起始地址
(4)	存储区大小
(5)	输出组 显示'gap'表示该存储区没有存储任何内容
(6)	输出至载入模块文件的段名
(7)	由目标模块文件读取的段名
(8)	输入模块名称
(9)	段起始地址
(10)	输出/输入段的大小
(11)	段类型及重定位属性
(12)	用于该汇编器的目标设备
(13)	设备文件版本号

10.3 目标转换器输出列表

目标转换器输出列表如下。

输出列表文件名称	输出列表名称
错误列表文件	错误列表

10.3.1 错误列表

错误列表存储目标转换器启动时输出的错误信息。

[输出格式]

与连接器输出的错误列表格式相同。

10.4 库管理程序输出列表

库管理程序输出列表如下。

输出列表文件名称	输出列表名称
列表文件	库数据输出列表

10.4.1 库数据输出列表

库数据输出列表输出库文件中模块数据。

[输出格式]

```

78K/OS Series librarian Vx.xx      DATE:(1) xx xxx xx  PAGE(2)1
LIB-FILE NAME:(3)K0S.LIB          ((4)xx xxx xx)
(5)0001 (6)K0SMAN.REL              ((7)xx xxx xx)
(8)MAIN                            (8)START
NUMBER OF PUBLIC SYMBOLS: (9)2
(5)0002 (6)K0SSUB.REL              ((7)xx xxx xx)
(8)CONVAH
NUMBER OF PUBLIC SYMBOLS: (9)1

```

[各输出项的说明]

条目	详细说明
(1)	列表创建日期
(2)	页码数
(3)	库文件名称
(4)	库文件创建日期
(5)	模块序列号 (从 0001 开始)
(6)	模块名称
(7)	模块创建日期
(8)	公共符号名称
(9)	模块中定义的公共符号数

10.5 列表转换器输出列表

列表转换器输出如下一些列表。

输出列表文件名称	输出列表名称
绝对汇编列表文件	绝对汇编列表
错误列表文件	错误列表

10.5.1 绝对汇编列表

绝对汇编列表将绝对地址值嵌入汇编列表并输出该列表。

[输出格式]

与汇编器输出的汇编列表格式相同。

10.5.2 错误列表

错误列表存储列表转换器启动时输出的错误信息。

[输出格式]

与汇编器输出的错误列表格式相同。

第 11 章 高效使用 RA78K0S

本章描述如何高效使用 RA78K0S。

11.1 提高操作效率 (EXIT 状态功能)

当 RA78K0S 中任何功能程序完成处理时，相应程序就会将其在处理过程中产生的最高错误等级存储为 "EXIT status"，并将控制权返回至操作系统。

EXIT 状态如下所示：

- 正常操作: 0
- 警告产生: 0
- 致命错误产生: 1
- 异常中止产生: 2

可以用这些退出状态创建一个批处理文件，提高操作效率。

[使用示例]

批处理文件 (RA.BAT) 的内容

```

ra78K0s -c9024 k0smain. -g -e
echo off
IF ERRORLEVEL 1 GOTO ERR
echo\
echo on
ra78K0s -c9024 k0ssub.asm -g -e
echo off
IF ERRORLEVEL 1 GOTO ERR
echo\
echo on
lk78K0s k0smain.rel k0ssub.rel -ok0s.lmf -g
echo off
IF ERRORLEVEL 1 GOTO ERR
echo\
echo on
oc78K0s k0s.lmf
echo off
IF ERRORLEVEL 1 GOTO ERR
GOTO EXIT
:ERR
echo Error occurred
:EXIT

```

- 用批处理文件(RA.BAT)执行处理过程。

C>>ra.bat

11.2 准备软件开发环境 (环境变量)

RA78K0S 支持如下一些用于准备软件开发环境的环境变量。

PATH: 搜索执行格式路径
 INC78K0S: 搜索包含文件路径 (仅适用于汇编器)
 LIB78K0S: 搜索库文件路径 (仅适用于连接器)
 TMP: 用于创建临时文件的路径
 LANG78K; Kanji (2 字节字符) 类型设定

在开发程序时, 创建一个用于收集所有相关文件的子目录, 不失一个很好的主意。因为这样可以使得程序开发更加容易且更加方便。

[使用示例]

AUTOEXEC.BAT 的内容

```
;AUTOEXEC.BAT
Verify on
break on
PATH C:\BIN;C:\BAT;C:\RA78K0S;          ← (1)
SET INC78K0S = C:\RA78K0S\INCLUDE       ← (2)
SET LIB78K0S = C:\RA78K0S\LIB           ← (3)
SET TMP = C:\TMP                         ← (4)
SET LANG78K = SJIS                       ← (5)
```

- (1) 由于指定了该路径, 就可以按照 C:\BIN, C:\BAT, C:\RA78K0S 这样的顺序从目录中取回执行格式文件。
- (2) 汇编器从目录 C:\RA78K0S\INCLUDE 中取回包含文件。
- (3) 连接器从目录 C:\RA78K0S\LIB 中取回库文件。
- (4) 各功能程序将临时文件创建在目录路径 C:\TMP 下。
- (5) 注释语句中的 Kanji 被解释成 shift JIS 码。

注意事项 如果程序安装需要使用 Windows 安装器, 程序会自动设置所需要的环境变量。

11.3 中断程序执行

每个功能程序都可以通过(CTRL + C)中断执行。

如果在批处理文件执行期间指定了'break on', 不管何时输入, 控制权都会返回至操作系统。当指定'break off'时, 仅在屏幕显示期间, 控制权会返回至操作系统。在这种情况下, 所有临时文件及输出文件都会被删除。

11.4 使汇编列表易于读取

使用-LH 选项或 TITLE 控制指令，在汇编列表头显示一个标题。这样通过显示一个简要指示汇编列表内容的标题，可以很容易一眼就看出汇编列表内容。

当使用了 SUBTITLE 控制指令时，同样可以显示次标题。需要了解控制指令的详细信息，请参阅语言中的第四章 控制指令。

[使用示例]

打印汇编列表文件头中的标题。

```
C>ra78k0s -c9024 k0smain.asm -lhRA78K0S_MAINROUTINE
```

这要引用 K0SMAIN.PRN.

```
日期:xx xxx xxxx 页码: 1
78K/0S Series Assembler Ex.xx RA78K0S_MAINROUTINE
└── 标题
```

```
Command: -c9024 k0smain.asm -lhRA78K0S_MAINROUTINE
Para- file:
In- file: K0SMAIN.ASM
Obj- file:K0SMAIN.REL
Prn- file:K0SMAIN.PRN
```

Assemble list

```
ALNO STNO ADRS OBJECT M I SOURCE STATEMENT
```

```
1 1
2 2          NAME SAMPM
3 3          ;*****
4 4          ;*
5 5          ;*  HEX -> ASCII 转换程序
6 6          ;*
7 7          ;*  主程序
          *
```

⋮

11.5 降低程序的启动时间

11.5.1 在源程序中指定控制指令

与通常在汇编器启动程序中指定的选项具有相同功能控制指令，可以预先在源程序中指定。这样就可以不需要在每次启动汇编器时都要指定选项。

[使用示例]

```

$ PROCESSOR (9024)          ] 控制指令
$ XREF

NAME SAMPM
; *****
; *
; *  HEX -> ASCII 转换程序   *
; *
; *   主程序                 *
; *
; *****
:

```

11.5.2 使用 PM+

RA78K0S 中的每一个程序选项都会自动存储在 PM+内的一个项目文件(.PRJ)中，这是因为这些存储的选项要用于第二个或紧接着的编译(MAKE)操作。所以使用 PM+，就可以避免在每次启动程序时都必需指定选项。

11.5.3 创建参数文件和子命令文件

当执行 RA78K0S 的任何一个功能程序(汇编器, 连接器, 目标转换器和表转换程序)时, 如果所有需要的数据不满足命令行数据格式要求时, 或者相同选项在每次程序执行时都要指定时, 就可创建参数文件。

同时, 也可以将子命令注册到在库管理程序中的子命令文件中。这就使得目标模块库形成更容易。

[使用示例 1]

创建参数文件并执行汇编。

参数文件 K0SMAIN.PRA 的内容。

```
;参数文件  
k0smain.asm -osample.rel -g  
-psample.prn
```

在命令行中输入如下命令。

```
C>ra78k0s -fk0smain.pra
```

[使用示例 2]

创建参数文件并执行汇编。

参数文件 K0S.SLB 的内容。

```
;  
;库创建命令。  
;  
create k0s.lib  
;  
add k0s.lib k0smain.rel &  
k0ssub.rel  
;  
exit
```

在命令行中输入如下命令。

```
C>lb78k0s <k0s.slb
```

11.6 目标模块库生成

汇编器及连接器为每一个输出模块都创建了一个文件。当目标模块很多时，与模块所对应的文件数也会大量增加。因此，RA78K0S 中有一个功能就是将这些大量的目标模块收集到一个单一的文件中，这种功能就是所谓的模块库生成。而形成该库的文件称为库文件。

库文件可以输入至连接器中，因此，当进行模块化编程时，用户可以通过创建由多个程序通用模块组成的库文件，方便而有效地对程序进行管理和操作。

第 12 章 错误消息

本章解释由汇编软件包(结构化汇编器, 汇编器, 连接器, 目标转换器及库管理程序)输出的错误消息产生的原因, 以及用户要采取的相应措施。

12.1 错误消息概述

由 RA78K0S 输出的错误消息可以分为以下 3 个级别层次。

(1) 异常中止错误 (Axxx)

该种类型错误发生时, 会使得程序无法继续执行。程序会立即退出(中断)。

如果在命令行上发现异常中止错误, 这时当发现另外一个命令行错误时便结束程序处理。

(2) 致命错误 (Fxxx)

有执行错误产生。如果又发现一个错误, 那么程序便退出(中断)而且不生成输出对象。

当有致命错误发生时, 为了表明程序退出时没有生成输出对象, 这时如果有同名对象存在, 也要将该对象删除。

(3) 警告错误 (Wxxx)

产生非用户所预期的输出对象。

备注 在以对话格式执行的程序中, 除非有异常中止发生, 否则程序执行会正常结束。

RA78K0S 错误消息分类如下。

从下一页开始对每一个 RA78K0S 错误消息进行解释说明。

- A0xx --- 命令行分析错误
- A9xx --- 文件或系统错误
- A1xx --- 其它异常中止错误
- F2xx --- 语句规范错误
- F3xx --- 表达式错误
- F4xx --- 符号错误
- F5xx --- 段错误
- F6xx --- 控制指令或宏错误
- W7xx --- 任何类型的警告错误

12.2 结构化汇编器错误消息

表 12-1. 结构化汇编器错误消息 (1/5)

A001	消息	输入文件不存在
	错误产生的原因	没有指定输入文件
	用户采取的措施	指定一输入文件
A002	消息	输入文件过多
	错误产生的原因	指定了两个或多个输入文件
	用户采取的措施	仅指定一个输入文件
A004	消息	非法文件名 '文件名'
	错误产生的原因	要么是文件名中有非法字符, 要么是文件名的字符个数超出了限制范围。
	用户采取的措施	输入合法字符文件名并且保证字符数不要超出限制范围。
A005	消息	非法文件指定 '文件名'
	错误产生的原因	指定了一个非法文件
	用户采取的措施	指定一合法文件
A006	消息	没有找到 '文件名'
	错误产生的原因	指定的文件不存在。
	用户采取的措施	指定已存在的文件。
A008	消息	文件指定与 '文件名' 冲突
	错误产生的原因	I/O 文件名重复指定
	用户采取的措施	指定不同的 I/O 文件名
A009	消息	无法生成文件 '文件名'
	错误产生的原因	指定的文件为写保护文件
	用户采取的措施	将指定文件的写保护取消掉
A010	消息	文件目录没有找到 '文件名'
	错误产生的原因	驱动器不存在和/或输出文件名中所包含的目录不存在
	用户采取的措施	指定已存在驱动器和/或目录
A011	消息	非法 '选项' 路径
	错误产生的原因	指定的路径与选项中指定的参数路径不同
	用户采取的措施	指定一正确的路径名
A012	消息	缺少参数 '选项'
	错误产生的原因	必需的参数没有指定
	用户采取的措施	指定需要的参数
A014	消息	超出 '选项' 范围
	错误产生的原因	指定的数值超出范围
	用户采取的措施	指定正确的数值
A015	消息	参数太长 '选项'
	错误产生的原因	参数中的字符数超出限制范围
	用户采取的措施	指定字符数在限制范围的参数

表 12-1. 结构化汇编器错误消息 (2/5)

A016	消息	非法的参数 '选项'
	错误产生的原因	参数语法不正确
	用户采取的措施	指定正确的参数
A017	消息	参数过多 '选项'
	错误产生的原因	参数数目超出了限制范围
	用户采取的措施	使指定参数数目在限制范围内
A018	消息	'选项'选项不可识别
	错误产生的原因	选项名不正确
	用户采取的措施	指定正确的选项名
A019	消息	参数文件嵌套
	错误产生的原因	在参数文件内指定了选项-F
	用户采取的措施	不要在参数文件内指定了选项-F
A020	消息	参数文件读错误 '文件名'
	错误产生的原因	无法读取参数文件
	用户采取的措施	指定正确的参数文件
A021	消息	存储器定位失败
	错误产生的原因	没有足够的存储空间
	用户采取的措施	确保必需的存储空间
A101	消息	打开/读/写/关闭 '文件' 错误
	错误产生的原因	由于文件 I/O 错误, 不能正常地 打开/读/写/关闭 文件
	用户采取的措施	指定正确的文件
A102	消息	找不到 '文件名'
	错误产生的原因	要么是包含文件名不存在, 要么是包含文件名与输入文件或输出文件名一起指定。
	用户采取的措施	指定正确的路径, 目录和文件名
A103	消息	非法的包含文件 '文件名'
	错误产生的原因	指定了非法的包含文件的文件名
	用户采取的措施	指定正确的文件名
A104	消息	非法(-sc) 字符
	错误产生的原因	在-SC 选项中指定了不能用作符号的字符
	用户采取的措施	指定正确的字符
A105	消息	无法定义保留字符
	错误产生的原因	在选项-D 中指定了保留字
	用户采取的措施	不要在选项-D 中指定保留字
A106	消息	相同的 PROCESSOR 控制
	错误产生的原因	在源文件中多次指定了 PROCESSOR 控制指令。 产品类型与选项-C 中指定的类型不同。
	用户采取的措施	只指定 PROCESSOR 控制指令一次。 更正产品类型名称。

表 12-1. 结构化汇编器错误消息 (3/5)

A107	消息	没有指定处理器
	错误产生的原因	没有指定设备类型。
	用户采取的措施	指定设备类型。
A108	消息	指定了非法的处理器类型。
	错误产生的原因	源文件里 PROCESSOR 控制指令中的设备类型指定不正确。
	用户采取的措施	指定正确的设备类型。
A109	消息	指定了非法的处理器类型。 -C
	错误产生的原因	-C 选项中的设备类型指定不正确的。
	用户采取的措施	指定正确的设备类型。
A110	消息	在模块标题之外无法使用这种控制
	错误产生的原因	应该写入源模块头的指令却写入了通常源文件行中去了。
	用户采取的措施	在源模块头写入该指令。
A111	消息	模块头语法错误。
	错误产生的原因	写入源模块头的指令语法不正确。
	用户采取的措施	使用正确的语法写入该指令。
A112	消息	结构化汇编器预处理内部错误。
	错误产生的原因	结构化汇编器预处理内部产生错误。
	用户采取的措施	联系日电电子有限公司。

F201	消息	非法的#ELSE/#ENDIF
	错误产生的原因	#ELSE 及#ENDIF 语句写入了错误位置处。
	用户采取的措施	将#ELSE 及#ENDIF 语句写入了正确位置处。
F202	消息	非法的#ENDCALLT
	错误产生的原因	#ENDCALLT 语句写入了错误位置处。
	用户采取的措施	将#ENDCALLT 语句写入正确位置处。
F203	消息	#ENDIF 不存在
	错误产生的原因	#ENDIF 语句丢失。
	用户采取的措施	将#ENDIF 语句写入到正确位置处。
F204	消息	#ENDCALLT 不存在
	错误产生的原因	#ENDCALLT 语句丢失。
	用户采取的措施	将#ENDCALLT 语句写入正确位置处。
F205	消息	#DEFCALLT 定义过多
	错误产生的原因	所注册的 callt 指令转换模式数超出了限定范围。
	用户采取的措施	减少所注册的#defcallt 指令数目。
F206	消息	CALL 指令过多。
	错误产生的原因	由#DEFCALLT 至 #ENDCALLT 所定义的指令过多。
	用户采取的措施	指定只由#DEFCALLT 至 #ENDCALLT 定义一条指令。

表 12-1. 结构化汇编器错误消息 (4/5)

F207	消息	重复定义
	错误产生的原因	同时定义了相同的转化模式
	用户采取的措施	更正#DEFCALLT 注册项。
F208	消息	符号表溢出
	错误产生的原因	符号数目超出了限定范围。
	用户采取的措施	减少符号数目。
F209	消息	语法错误
	错误产生的原因	所写入的语句语法不正确。
	用户采取的措施	使用正确的语法。
F210	消息	嵌套级数错误
	错误产生的原因	嵌套时出现了错误 (溢出, 嵌套级数等等)
	用户采取的措施	使用正确的控制语句。
F211	消息	一行内字符数过多。
	错误产生的原因	行内字符数长度超出了限定数目范围。
	用户采取的措施	在一行内指定 218 个字符或更少数目。
F212	消息	包含文件数过多。
	错误产生的原因	包含文件内包含伪指令。
	用户采取的措施	不要在包含文件内包含伪指令。
F214	消息	非法 BREAK 语句
	错误产生的原因	BREAK 语句写入到了错误的位置处
	用户采取的措施	将 BREAK 语句写入正确位置处。
F215	消息	非法 CONTINUE 语句
	错误产生的原因	CONTINUE 语句写入到了错误的位置处
	用户采取的措施	将 CONTINUE 语句写入正确位置处。
F216	消息	非法的 CASE/DEFAULT/ENDS 语句
	错误产生的原因	CASE/DEFAULT/ENDS 语句写入到了错误的位置处
	用户采取的措施	将 CASE/DEFAULT/ENDS 语句写入正确位置处。
F217	消息	非法的 ELSEIF/ELSE/ENDIF 语句
	错误产生的原因	ELSEIF/ELSE/ENDIF 语句写入到了错误的位置处
	用户采取的措施	将 ELSEIF/ELSE/ENDIF 语句写入正确位置处。
F218	消息	非法的 NEXT 语句
	错误产生的原因	NEXT 语句写入到了错误的位置处
	用户采取的措施	将 NEXT 语句写入正确位置处。
F219	消息	非法的 ENDW 语句
	错误产生的原因	ENDW 语句写入到了错误的位置处
	用户采取的措施	将 ENDW 语句写入正确位置处。
F220	消息	非法的 UNTIL/UNTIL_BIT 语句
	错误产生的原因	UNTIL 和 UNTIL_BIT 语句写入到了错误的位置处

用户采取的措施	将 UNTIL 和 UNTIL_BIT 语句写入正确位置处。
---------	--------------------------------

表 12-1. 结构化汇编器错误消息 (5/5)

F221	消息	ENDIF 不存在
	错误产生的原因	ENDIF 语句丢失。
	用户采取的措施	将 ENDIF 语句写入正确位置处。
F222	消息	ENDS 不存在
	错误产生的原因	ENDS 语句丢失。
	用户采取的措施	将 ENDS 语句写入正确位置处。
F223	消息	ENDW 不存在
	错误产生的原因	ENDW 语句丢失。
	用户采取的措施	将 ENDW 语句写入正确位置处。
F224	消息	NEXT 不存在
	错误产生的原因	NEXT 语句丢失。
	用户采取的措施	将 NEXT 语句写入正确位置处。
F225	消息	UNTIL/UNTIL_BIT 不存在
	错误产生的原因	UNTIL 和 UNTIL_BIT 语句丢失。
	用户采取的措施	将 UNTIL 和 UNTIL_BIT 语句写入正确位置处。
F226	消息	行中有非法字符
	错误产生的原因	源程序行中写入了错误的字符
	用户采取的措施	删除写入源程序行中的错误字符
F227	消息	行中有非法操作数
	错误产生的原因	替换数据大小及比较条件格式不正确
	用户采取的措施	指定正确的数据大小
F228	消息	操作数中非法 SFR 访问
	错误产生的原因	写入了无法访问替换格式的 sfr 符号。
	用户采取的措施	检查 sfr 符号的访问状态并写入正确的 sfr 符号。
F229	消息	该符号是保留“符号名”
	错误产生的原因	所使用的符号是保留字。
	用户采取的措施	更改符号名。
W301	消息	符号重新定义。
	错误产生的原因	该符号由 #define 语句定义一次以上。
	程序处理情况	最近定义的符号有效。
	用户采取的措施	使最先定义的符号有效，更正语法。
W302	消息	重复的 PROCESSOR 选项和控制
	错误产生的原因	由选项-C 指定的设备类型与在 \$PROCESSOR 控制命令中指定的设备类型不一样
	程序处理情况	由选项-C 指定的设备类型有效，忽略在 \$PROCESSOR 控制命令中指定的设备类型。
	用户采取的措施	检查在选项-C 指定的设备类型是否正确。

12.3 汇编器错误消息

表 12-2. 汇编器错误消息 (1/8)

A101	消息	源程序大小为 0‘文件名’。
	错误产生的原因	输入了文件大小为 0 模块。
A102	消息	指定了非法的处理器类型。
	错误产生的原因	在目标设备指定时产生错误。
A103	消息	模块头语法错误。
	错误产生的原因	可以写入源模块头的控制指令格式产生错误。
A104	消息	在模块标题之外无法使用这种控制
	错误产生的原因	在源模块头指定的控制指令写进了普通源程序中去了。
A105	消息	相同的 PROCESSOR 控制
	错误产生的原因	PROCESSOR 控制指令不止一次写进了源模块头。
A106	消息	用于模块名的非法源文件名。
	错误产生的原因	模块名无法创建，因为源文件名的基本名中包含有非法符号结构字符。
A107	消息	缺省段?CSEG 已被使用。
	错误产生的原因	试图用缺省段定义一未定义段。
A108	消息	符号表溢出‘符号名’
	错误产生的原因	已经超出可定义的符号数。
A109	消息	DS 数太多。
	错误产生的原因	段中目标代码之间空隙太多，原因是 DS 指令使用过多，因此数据无法输出至目标文件中去。
A110	消息	字符串表溢出。
	错误产生的原因	超出了字符串表限定范围。
	用户采取的措施	将符号数减少至 9 个或 9 个以下字符。
A111	消息	目标代码超过了个 128 字节。
	错误产生的原因	在源程序语句中每行目标代码超过了个 128 字节。
A112	消息	没有指定处理器
	错误产生的原因	在命令行或源模块文件中没有指定目标设备。
A114	消息	在 C 语言源程序中，asm 语句的局部符号名必需要以‘?L’开头。
	错误产生的原因	在 C 语言源程序中，用#asm 语句描述的局部符号名不是以‘?L’开头。
A115	消息	源程序语句行太长。
	错误产生的原因	超出了行长度的限定范围(2048 个字符)。

F201	消息	语法错误
	错误产生的原因	使用了错误的语句格式。
F202	消息	操作数非法。
	错误产生的原因	指定的操作数非法。
F203	消息	寄存器非法。
	错误产生的原因	指定了不允许指定的寄存器。
F204	消息	非法字符。
	错误产生的原因	在源程序模块中指定了一非法字符。

表 12.2. 汇编器错误消息 (2/8)

F205	消息	字符串中出现意外字符 LF。
	错误产生的原因	在字符串关闭之前，字符串中出现回车代码。
F206	消息	字符串中出现意外 EOF。
	错误产生的原因	在字符串关闭之前，字符串中出现文件结束代码。
F207	消息	字符串中出现意外 null 代码。
	错误产生的原因	有一 null 代码(00H)写入字符串中。

F301	消息	表达式太复杂。
	错误产生的原因	表达式太复杂。
F302	消息	希望是绝对表达式。
	错误产生的原因	指定了可重定位表达式。
F303	消息	非法的表达式。
	错误产生的原因	所使用的表达式格式不正确。
F304	消息	在'文件名'表达式中有非法的符号。
	错误产生的原因	表达式中有不可以使用的符号。
F305	消息	字符常量太长。
	错误产生的原因	超出了字符常量的长度限定范围(4 个字符)。
F306	消息	非法数字。
	错误产生的原因	指定了错误的数值。
F307	消息	除数为零。
	错误产生的原因	有数值被零除。
F308	消息	数值太大。
	错误产生的原因	数值常量超过了 16 位。
F309	消息	非法的位型数值。
	错误产生的原因	指定了错误的位型数值。
F310	消息	位型数值超出了范围。
	错误产生的原因	位型数值超出了 0 至 7 的范围。
F311	消息	操作数超出了范围 (n)
	错误产生的原因	指定的数值超出了范围 n(0 至 7)。
F312	消息	数值超出了范围 (字节)
	错误产生的原因	操作数数值超出了范围(00H 至 FFH)，或者字节操作数的数值超出了(-128 至 +127)范围。
F313	消息	数值超出了范围 (addr5)
	错误产生的原因	操作数超出了 addr5 的可指定范围(40H 至 7EH)。
F314	消息	数值超出了范围 (addr11)
	错误产生的原因	操作数超出了的可指定范围(800H 至 FFFH)。
F315	消息	数值超出了范围 (saddr)
	错误产生的原因	操作数超出了 saddr 的可指定范围(0FE20H 至 0FF1FH)。
F316	消息	数值超出了范围 (addr16)
	错误产生的原因	操作数超出 addr16 的可指定范围(根据目标设备而不同)。

表 12-2. 汇编器错误消息 (3/8)

F317	消息	期望为偶数表达式。
	错误产生的原因	为字访问指定了奇数地址。
F318	消息	数值超出了范围 (sfr)
	错误产生的原因	指定的 SFR/SFRP 指令操作数超出了限定范围，或者为 SFR 指令指定了一个奇数值操作数。
F401	消息	非法的 PUBLIC 符号 '符号名'。
	错误产生的原因	此符号不能声明为 PUBLIC 型
F402	消息	非法的 EXTRN/EXTBIT 符号 '符号名'。
	错误产生的原因	此符号不能声明为 EXTRN/EXTBIT 型
F403	消息	不能定义 PUBLIC 符号 '符号名'
	错误产生的原因	该符号已经有了一个 PUBLIC 声明，不能对具有 PUBLIC 声明的符号进行 PUBLIC 定义。
	用户采取的措施	用位项而不是用 saddr.bit 定义的符号不能够进行 PUBLIC 声明。取消声明或更改 EQU 定义。
F404	消息	没有定义的 Public 符号 '符号名'
	错误产生的原因	具有 PUBLIC 声明的符号不可定义为 PUBLIC 符号
F405	消息	非法的位符号
	错误产生的原因	前向参考符号使用了非法符号，或者机器语言指令的操作数的位符号使用了非法符号。
	用户采取的措施	为位符号指定后向参考或进行 EXTBIT 声明。
F406	消息	不可参考前向位符号 '符号名'
	错误产生的原因	指定前向引用位符号或引用表达式中的位符号
F407	消息	未定义的符号参考 '符号名'
	错误产生的原因	使用了未定义的符号
F408	消息	多重符号定义 '符号名'
	错误产生的原因	符号名定义一次以上。
F409	消息	操作数中的符号太多
	错误产生的原因	写入操作数的符号数超过了一行可以描述的符号个数。
F410	消息	状态错误
	错误产生的原因	在汇编期间符号值发生了变化 (例如,在操作数中定义了一个 EQU 符号标号, 该符号标号由于 BR 指令的最优处理而被更改。)
F411	消息	该符号是保留值 '符号名'
	错误产生的原因	指定的符号名为保留字。
F502	消息	非法段名
	错误产生的原因	非法段名符号
F503	消息	不同段类型 '段名'
	错误产生的原因	用同一个名称定义了两个或两个以上的段, 但它们的类型是不同的。
F504	消息	段数目太多
	错误产生的原因	定义的段数目超出了限定范围 (100)。

表 12-2. 汇编器错误消息 (4/8)

F505	消息	当前段不存在
	错误产生的原因	在段生成之前就写入了 ENDS 指令, 或在本段结束之后, 下一个段生成之前就立即写入了 ENDS 指令。
F506	消息	DB, DW, DS, ORG 标号不可在 BSEG 中描述
	错误产生的原因	在位段中定义了 DB, DW, DS, ORG 指令。
F507	消息	不能在 CSEG 之外描述操作码
	错误产生的原因	在代码段之外定义了机器语言指令或者 BR 指令。
F508	消息	不能在 BSEG 之外描述 DBIT
	错误产生的原因	在位段之外定义了 DBIT 指令。
F509	消息	指定了非法地址
	错误产生的原因	定位至绝对地址段的地址超出了该段范围。
F510	消息	地址计数器溢出
	错误产生的原因	地址计数器超出了一个段的范围
F511	消息	期望是段名
	错误产生的原因	没有指定段定义指令的段名, 因为重定位属性是 AT。
F512	消息	段大小为奇数 '段名'
	错误产生的原因	用奇数描述了重定位属性 callt0 段的大小。

F601	消息	包含文件嵌套溢出
	错误产生的原因	包含文件嵌套超出了限定范围 (2 级)。
F602	消息	必需指定选择语句
	错误产生的原因	没有指定 Switch 名称
F603	消息	选择语句过多
	错误产生的原因	Switch 名称超出了限定范围(每个模块限定 5)
F604	消息	IF 语句嵌套过多
	错误产生的原因	IF/_IF 语句嵌套超出了限定范围(8 级)。
F605	消息	不需要 ELSE 语句存在
	错误产生的原因	在不需要 ELSE 语句的地方有一条 ELSE 语句。
F606	消息	不需要 ENDIF 语句存在
	错误产生的原因	在不需要 ENDIF 语句的地方有一条 ENDIF 语句。
F607	消息	ELSE 或 ENDIF 不存在
	错误产生的原因	IF/_IF 语句所要求的 ELSE 或 ENDIF 语句缺失。
F608	消息	ENDIF 不存在
	错误产生的原因	IF/_IF 语句所要求的 ENDIF 语句缺失。
F609	消息	非法的 ELSEIF 语句
	错误产生的原因	ELSE 语句之后写入一条 ELSEIF 或 _ELSEIF 语句。
F610	消息	多符号定义(MACRO) '符号名'
	错误产生的原因	用于定义宏名的符号已被定义。
F611	消息	非法参数语法
	错误产生的原因	宏的形式参数不正确。

表 12-2. 汇编器错误消息 (5/8)

F612	消息	参数太多
	错误产生的原因	宏定义的形式参数数目超出了限定范围 (16).
F613	消息	参数重名'符号名'
	错误产生的原因	指定的宏定义符号与形式参数符号相同。
F614	消息	不能嵌套宏定义
	错误产生的原因	宏定义不能与另外一个宏定义嵌套定义。
F615	消息	局部符号的非法语法
	错误产生的原因	LOCAL 指令中指定的操作数不正确。
F616	消息	局部符号过多
	错误产生的原因	可以在 1 个宏结构体描述的局部符号数超出了限定范围(64)。
F617	消息	ENDM 不存在
	错误产生的原因	宏定义指令所要求的 ENDM 语句不存在。
F618	消息	非法 ENDM 语法
	错误产生的原因	ENDM 语句不正确
F619	消息	所定义的宏非法
	错误产生的原因	引用宏定义不正确
F620	消息	非法实际参数语法
	错误产生的原因	宏的实际参数指定不正确。
F621	消息	宏引用嵌套过多
	错误产生的原因	宏引用嵌套超出了限定范围(8 级)。
F622	消息	非法 EXITM 语法
	错误产生的原因	EXITM 语句不正确
F623	消息	REPT 操作数非法
	错误产生的原因	在 REPT 指令操作数内指定了一个不允许的表达式。
F624	消息	多于??RAFFFF
	错误产生的原因	在宏扩展期间, 所替代的局部符号数目多于 65,535。
F625	消息	意外的 ENDM
	错误产生的原因	发现了意外的 ENDM。
F626	消息	无法描述 LOCAL 宏定义
	错误产生的原因	在一般源程序语句而不是宏结构体中指定了 LOCAL 指令。
F627	消息	在该包含文件/宏中有两个以上的段
	错误产生的原因	在包含文件, 宏结构体, rept-endm 块或 irp-endm 块中发现了 2 个更多的段。
W702	消息	重复的 PROCESSOR 选项和控制
	错误产生的原因	同时指定了目标设备(-C)命令行指定选项和源程序头中的 PROCESSOR 指令。
	程序处理	目标设备(-C)命令行指定选项有效, 而忽略源程序头中的 PROCESSOR 指令。
W703	消息	多定义模块名
	错误产生的原因	NAME 指令被定义 2 次或更多次
	程序处理	NAME 指令不可用且已经定义的模块名可用。

表 12-2. 汇编器错误消息 (6/8)

W704	消息	已经声明的 EXTRN 符号 '符号名'
	错误产生的原因	该符号为已经声明的 EXTRN.
	用户采取的措施	在 1 个模块中指定 EXTBIT 声明一次
W705	消息	已经声明的 EXTBIT 符号 '符号名'
	错误产生的原因	该符号为已经声明的 EXTBIT.
	用户采取的措施	在 1 个模块中指定 EXTBIT 声明一次。
W706	消息	END 语句不存在
	错误产生的原因	在源文件未没有写入 END 语句。
	程序处理	假定源文件未已写入 END 语句。
W707	消息	END 指令后有非法语句
	错误产生的原因	END 语句后有注释, 空格, 制表符, 或 CR 码以外的语句项目。
	程序处理	忽略 END 语句后的所有语句项。
W708	消息	已经声明了的 LOCAL 符号 '符号名'
	错误产生的原因	该符号为已经声明的 LOCAL.
	用户采取的措施	在每个宏中只声明一个符号 LOCAL 。
W709	消息	实际参数数目不够
	错误产生的原因	设定的实际参数比形式参数数目少。
	程序处理	实际参数不足处的形式参数处理为空字符串。
W710	消息	实际参数数目过多
	错误产生的原因	设定的实际参数比形式参数数目多。
	程序处理	忽略多余的实际参数。
W711	消息	要报告的错误过多
	错误产生的原因	单行中要报告的错误过多(也即 6 或更多错误)
	程序处理	第六个及其后的错误消息不输出但处理将继续进行。
W712	消息	交叉参考工作区不足
	错误产生的原因	用于要处理输出交叉参考列表的存储区不足。
	程序处理	不输出交叉参考列表但处理将继续进行。
A901	消息	无法打开源程序文件'文件名'
	错误产生的原因	源程序文件不能打开。
A902	消息	无法打开参数文件'文件名'
	错误产生的原因	参数文件不能打开。
A903	消息	无法打开包含文件'文件名'
	错误产生的原因	包含文件不能打开。
A904	消息	非法的包含文件'文件名'
	错误产生的原因	仅有驱动器名, 路径名或设备类型文件名才可以指定为包含文件名。

表 12-2. 汇编器错误消息 (7/8)

A905	消息	不能打开重叠文件 '文件名'
	错误产生的原因	重叠文件不能打开。
	用户采取的措施	确保重叠文件与汇编执行格式文件在相同的目录之下。
A906	消息	非法的重叠文件 '文件名'
	错误产生的原因	重叠文件内容非法
A907	消息	不能打开目标文件 '文件名'
	错误产生的原因	目标文件不能打开。
	用户采取的措施	使用在其目录路径下带有打开区域的驱动器
A908	消息	不能打开打印文件 '文件名'
	错误产生的原因	汇编列表文件不能打开。
	用户采取的措施	使用在其目录路径下带有打开区域的驱动器
A909	消息	不能打开错误列表文件 '文件名'
	错误产生的原因	错误列表文件不能打开。
	用户采取的措施	使用在其目录路径下带有打开区域的驱动器
A910	消息	不能打开临时文件 '文件名'
	错误产生的原因	临时文件不能打开。
	用户采取的措施	使用在其目录路径下带有打开区域的驱动器
A911	消息	系统错误
	错误产生的原因	产生了系统错误
	用户采取的措施	确认执行环境并重新执行汇编操作
A912	消息	无法设置 CTRL+C
	错误产生的原因	CTRL+C 无法设置，因为汇编执行已经异常中止。
	用户采取的措施	确认执行环境并重新执行汇编操作
A913	消息	无法读取源程序文件 '文件名'
	错误产生的原因	在源程序文件中有文件输入/输出错误产生
A914	消息	无法读取参数文件 '文件名'
	错误产生的原因	在参数文件中有文件输入/输出错误产生
A915	消息	无法读取包含文件 '文件名'
	错误产生的原因	在包含文件中有文件输入/输出错误产生
A916	消息	无法读取覆盖文件 '文件名'
	错误产生的原因	在覆盖文件中有文件输入/输出错误产生
A917	消息	无法写入目标文件 '文件名'
	错误产生的原因	在目标文件中有文件输入/输出错误产生
	用户采取的措施	将目标文件输出至另外一个目录下或者只指定磁盘创建一个打开区
A918	消息	无法写入打印文件 '文件名'
	错误产生的原因	在汇编列表文件中有文件输入/输出错误产生
	用户采取的措施	将汇编列表文件输出至另外一个目录下或者只指定磁盘创建一个打开区
A919	消息	无法写入错误列表文件 '文件名'
	错误产生的原因	在错误列表文件中有文件输入/输出错误产生
	用户采取的措施	将错误列表文件输出至另外一个目录下或者只指定磁盘创建一个打开区

表 12-2. 汇编器错误消息 (8/8)

A920	消息	无法读出/写入临时‘文件’文件名’
	错误产生的原因	在临时文件中有文件输入/输出错误产生
	用户采取的措施	将临时文件输出至另外一个目录下或者只指定磁盘创建一个打开区
A921	消息	汇编器内部错误
	错误产生的原因	产生一个汇编器内部错误
	用户采取的措施	重新执行汇编。
A922	消息	主机上没有足够的存储空间。
	错误产生的原因	系统没有足够的存储空间来执行汇编器。
A923	消息	主机上没有足够的存储空间来执行宏。
	错误产生的原因	在宏处理期间，用于宏处理的存储空间不足。
	用户采取的措施	降低所定义宏的数量。

12.4 连接器错误信息。

表 12-3. 连接器错误信息。(1/5)

A101	消息	'文件名' 是无效输入文件(或者有不同主机生成)
	错误产生的原因	输入了目标文件以外的文件, 或者试图与在兼容主机上创建的目标文件进行连接。
F102	消息	指令语法错误
	错误产生的原因	指令设定不正确
A103	消息	'文件名' 为非法处理器类型
	错误产生的原因	汇编或编译的目标设备不是本连接器的目标设备。
	用户采取的措施	检查确保目标文件无误。检查确保汇编或编译的目标设备可以由本连接器进行处理。同时检查确保覆盖文件版本正确(连接器引用汇编器部分覆盖文件以获取目标设备上的特征数据)。
A104	消息	'文件名' 处理器类型不同于第一次输入文件的'第一次输文件名'。
	错误产生的原因	输入了其目标设备与第一次输入的目标文件不同的目标文件。
W105	消息	库文件 '文件名' 没有公共符号。
	错误产生的原因	库文件没有公共符号, 因此, 无法连接包含于库文件中的目标模块。
A106	消息	无法创建临时文件 '文件名'。
	错误产生的原因	无法创建临时文件。
F107	消息	指令中的名称'名称'已经定义。
	错误产生的原因	试图将保留字或者先前定义的名称定义为指令的存储区名。 该名称(保留字, 存储空间名, 存储区名)已经定义过了。
F108	消息	重叠存储区 '存储区 1' 及 '存储区 2'
	错误产生的原因	在存储区指令中定义的存储区地址重叠了。
F109	消息	存储区 '存储区名' 太长 (至多 31 个字符)
	错误产生的原因	在指令中指定的存储区名过长。 在指令中指定的存储区名为 32 个字符长度或更长。
F110	消息	存储区的 '存储区名' 已经定义。
	错误产生的原因	在存储器指令中指定的存储区已经注册。
F111	消息	存储区的 '存储区名' 再定义出界。
	错误产生的原因	在存储器指令中指定的存储区范围超出了可再定义范围。
F112	消息	'段名'段存储单元类型错误
	错误产生的原因	在合并指令中为该段指定了错误的存储单元类型。
F113	消息	连接器内部错误
	错误产生的原因	连接器内部错误
	用户采取的措施	请联系授权代表或日电电子公司
F114	消息	非法数字。
	错误产生的原因	指令中指定的数字值不正确。
F115	消息	数值太大 (最大 65,535/0FFFFH)
	错误产生的原因	指令中数值大于 65,535 (0FFFFH)。

表 12-3. 连接器错误信息。(2/5)

F116	消息	存储区 '存储区名' 定义出界。
	错误产生的原因	起始地址与指令中指定的存储区大小之和超过了 65,535 (0FFFFH).
F201	消息	合并指令中的多重段定义'段名'
	错误产生的原因	合并指令中指定的段已经注册 (试图用同一个段指定使用多个合并指令的存储单元).
F202	消息	段类型'段 1'错误匹配至文件中的'段 2' - 忽略
	错误产生的原因	发现有与该段段名相同的而又具有不同段类型重定位属性的段。
A203	消息	段 '段名' 为未知的段类型
	错误产生的原因	输入目标文件的段数据有错(输出段连接指定不正确).
F204	消息	存储区/空间'名'未定义。
	错误产生的原因	在合并指令中指定的存储区/空间'名'未定义。
F205	消息	指令中的名称 '名称' 属性错误
	错误产生的原因	在指令中描述了一个不能在段名、存储区名或存储空间名进行描述的名称条目(例如, 在要求存储区名出给出存储空间名)。
F206	消息	段 '段名' 不能定位至存储区- 忽略
	错误产生的原因	段不能定位至存储区(没有段定位所需要的足够的存储区)。
F207	消息	段 '段名'的段类型非法
	错误产生的原因	该段类型数据非法
F208	消息	段'段名' 属性可能不可更改
	错误产生的原因	对由汇编期间指定的重定位属性'AT xxxxH'所创建的段, 或由 ORG 指令所创建的段, 试图在指令中更改其连接类型。
F209	消息	段'段名' 排列可能不可更改
	错误产生的原因	对由汇编期间指定的重定位属性'AT xxxxH'所创建的段, 或由 ORG 指令所创建的段, 试图在指令中更改其单元地址。
	用户采取的措施	连接属性在连接期间指定的段, 其单元地址不可在汇编器中指定。
F210	消息	段'段名' 并不存在 - 忽略
	错误产生的原因	指令中指定的段并不存在。
F211	消息	库类型错误匹配'符号名'与文件中的'文件名' - 忽略
	错误产生的原因	在指定的符号库号中有错误匹配存在
	用户采取的措施	确保符号的库号正确。

表 12-3. 连接器错误信息。(3/5)

F301	消息	可重定位目标代码地址超出限定范围(文件'文件名',段'段名',地址 xxxxH,类型'寻址类型')
	错误产生的原因	输入目标文件内可重定位目标代码的更正数据输出至并没有目标代码存在的地址处(重定位入口地址超出了原始数据的地址范围)。
	用户采取的措施	检查符号参考是否正确
F302	消息	行号数据有非法符号索引(文件'文件名',段'段名')
	错误产生的原因	输入目标文件内用于调试的行号数据不正确,而且不能正确地引用符号数据。行号索引与符号索引不对应。
F303	消息	在可重定位目标代码中无法找到符号索引(文件'文件名',段'段名',地址 xxxxH,类型'寻址类型')
	错误产生的原因	输入目标文件内可重定位代码的更正数据不正确,而且不能正确地引用符号数据。可重定位条目与符号索引不对应。
	用户采取的措施	检查符号及变量引用方法是否正确。
F304	消息	数值超出了范围(段'段名',地址 xxxxH,类型'寻址类型')
	错误产生的原因	用于决定可重定位目标代码的操作数值超出了相应指令操作数的范围。
	用户采取的措施	在源程序中指定操作数值,使其在每一种寻址类型所要求的范围之内。
F305	消息	期望是偶数值 (段'段名',地址 xxxxH,类型'寻址类型')
	错误产生的原因	用于决定 callt 或 saddrp 寻址的可重定位目标代码的操作数值为一奇数值(callt 或 saddrp 寻址的操作数必需为偶数值)。
F306	消息	期望是一个 4 的倍数数值(段'段名',地址 xxxxH,类型'寻址类型')
	错误产生的原因	用于决定 saddrp 寻址的可重定位目标代码的操作数值不是 4 的倍数。

注意事项 段定位后,消息 F301 至 F306 内'地址 xxxxH'中所显示的地址为绝对地址。

A401	消息	'文件名'错误符号表
	错误产生的原因	输入目标文件的符号数据非法。输入文件的符号条目项不是以'.file'开头的。
A402	消息	文件'文件名'没有符号字符串表。
	错误产生的原因	输入目标文件的符号数据非法。
	用户采取的措施	重新执行一次汇编或编译。 通过为汇编器指定 8 个识别字符,编译器指定 7 个识别字符,可以避免这种错误。
F403	消息	符号'符号名'与文件内'文件名 1'的类型不匹配。 文件内第一个定义的是'文件名 2'
	错误产生的原因	具有相同名称的外部定义/参考符号类型在文件 1 和文件 2 中是不同的。
F404	消息	文件'文件名 1'具有多重符号定义。 文件第一个定义的是'文件名 2'
	错误产生的原因	在目标文件 1 中定义的公共符号已经在目标文件 2 中进行了 PUBLIC 声明。
F405	消息	文件'文件名'中有未定义的符号
	错误产生的原因	在文件中 EXTRN 声明符号在另一个文件中未作 PUBLIC 声明。

表 12-3. 连接器错误消息 (4/5)

W406	消息	堆栈区小于 10 个字节。
	错误产生的原因	保护堆栈区小于为 10 个字节或者更少。(由-S 选项指定的存储区中保护堆栈的大小为 10 字节或更少).
W407	消息	无法分配堆栈区。
	错误产生的原因	在存储区中没有空闲区用于堆栈区保护(在以选项-S 所指定的存储区中, 保护堆栈区不可执行)。
F408	消息	找不到符号-A
	错误产生的原因	连接器选项的程序入口地址中设定选项-A 后所输入的符号在公共符号中并不存在。
F409	消息	-A 符号 '符号名' 与类型不匹配。
	错误产生的原因	由选项-A 指定搜索符号的类型不正确。选项-A 是在连接器选项的程序入口地址中进行设定。
	用户采取的措施	对由程序入口地址设定搜索的符号, 使用允许类型。
W410	消息	文件'文件名 1'中有多模块名定义存在。 文件内第一个定义的是'文件名 2'
	错误产生的原因	目标文件 1 与目标文件 2 的模块名相同。
W411	消息	文件'文件名 1'中有不同的 REL 类型。
	错误产生的原因	目标文件类型版本不同。
	用户采取的措施	使用最新版本的汇编器或连接器。
F415	消息	-QD/QF/等及 Not -QD/QF/等. REL 混淆了
	错误产生的原因	输入目标文件对编译器最优选项有不同的指定, 而要求其对整个程序来讲必须是相同的。所以要使用与剩余程序相同的值。
W416	消息	文件'文件名(选项)'中有多 CAP/NOCAP 存在。
	错误产生的原因	所有输入目标文件 CAP/NOCAP 汇编或编译选项并不是相同的。
W417	消息	文件'文件名'中工具名版本多于一个 使用文件'文件名'的第一个名称
	错误产生的原因	所有输入目标文件一直到连接阶段使用的每一个工具(CC78K0S, ST78K0S, RA78K0S)版本与设备文件版本不同。
W418	消息	文件'文件名'版本太旧, 无法找到 TOOL 信息。
	错误产生的原因	当在输入目标文件中无法找到 TOOL 信息时, 会输出此错误信息。 通常, 当用老版本(DF-兼容型)目标文件执行连接时, 总会输出此错误信息。
F420	消息	文件'文件名'已经有'工具名'错误/警告存在。
	错误产生的原因	直到连接阶段, 才输出每个工具(CC78K0S, ST78K0S, RA78K0S)的错误消息或警告消息。
F425	消息	相同名称的“函数名”(文件'文件名')具有不同函数 ID 值。
	错误产生的原因	与由编译器声明的 EXT-FUNC 函数具有相同函数名, 但是却具有不同的 ID 值。
F431	消息	相同函数名(文件'文件名')ID 值下对应不同函数名。
	错误产生的原因	有两个或更多的编译器声明的 EXT-FUNC 函数具有相同的 ID 值。

表 12-3. 连接器错误信息。(5/5)

A901	消息	无法打开覆盖文件'文件名'
	错误产生的原因	重叠文件不能打开。
	用户采取的措施	确保覆盖文件位于正确的路径下（包含有执行程序的路径）。
A902	消息	找不到文件'文件名'
	错误产生的原因	指定库文件文件不能打开。
A903	消息	无法读取指定库文件'文件名'
	错误产生的原因	无法读取指定为输入文件的目标文件。
A904	消息	不能打开输入文件'文件名'
	错误产生的原因	输出文件不能打开。
	用户采取的措施	检查用于创建输出文件的磁盘状况(打开能力, 介质状况等)。
A905	消息	无法创建临时文件'文件名'。
	错误产生的原因	无法创建用于符号项的临时文件。
	用户采取的措施	检查用于创建临时文件的磁盘状况(打开能力, 介质状况等)。
A906	消息	无法写映像文件'文件名'
	错误产生的原因	数据无法写入连接列表文件'文件名'中去。
	用户采取的措施	检查用于创建连接列表文件的磁盘状况(打开能力, 介质状况等)。
A907	消息	无法写输出文件'文件名'
	错误产生的原因	数据无法写入载入模块文件'文件名'中去。
	用户采取的措施	检查用于创建输出文件的磁盘状况(打开能力, 介质状况等)。
A908	消息	无法访问临时文件'文件名'
	错误产生的原因	临时文件无法写入。
	用户采取的措施	检查用于创建临时文件的磁盘状况(打开能力, 介质状况等)。
A909	消息	无法读取设备文件'设备文件名'
	错误产生的原因	无法读取由选项指定或由控制命令指定的设备相应的设备文件。

12.5 目标转换程序错误消息

表 12-4. 目标转换程序错误消息 (1/2)

A006	消息	没有找到文件名'
	错误产生的原因	所指定的输入文件并不存在。
	用户采取的措施	当连接至编译器启动程序时，文件输出为["启动程序名称".lmf]。在这种情况下，使用连接器选项指定输出文件名为 [-o*.lmf]。
A100	消息	'文件名' 非法处理器类型
	错误产生的原因	汇编器或编译器的目标对象与该程序的目标对象不一样。
	用户采取的措施	检查载入模块文件是否正确，检查汇编器或编译器的目标对象，同时检查设备文件版本是否正确。
A101	消息	'文件名' 无效的输入文件 (或者是由不同的主机生成的)
	错误产生的原因	企图输入非载入模块文件，或者试图将载入模块文件转换为在兼容主机上创建的文件。
A103	消息	符号'符号名' 非法属性
	错误产生的原因	输入文件的符号属性存在错误。
A104	消息	'文件名' 非法输入文件 - 不予连接
	错误产生的原因	试图输入一目标文件。
A105	消息	主机上没有足够的存储空间。
	错误产生的原因	程序运行所需要的存储空间不足。
A106	消息	非法符号表
	错误产生的原因	输入载入模块文件的符号表存在错误。
	用户采取的措施	<p>当用 C 语言编写源程序时，请检查是否注意到了以下一些注意事项。</p> <p><注意事项></p> <ul style="list-style-type: none"> 当使用局部符号时，请用以字符串?L (?L@01, ?L@sym 等)开头的符号。但是要保证符号字符数为 8 个或更少。同时注意不要在外部定义该符号(PUBLIC 声明)。
A107	消息	不能为无 ROM 设备指定-U 选项
	错误产生的原因	已经为没有内部 ROM 的设备指定了对象补充选项(-U)。
F200	消息	未定义的符号'符号名'
	错误产生的原因	发现了没有指定地址的符号。
	用户采取的措施	<p>定义符号值。</p> <p>该符号作为外部参考符号引用。所以如果没有进行外部定义，就在指定该符号值的模块外部对其进行外部定义。</p>
F201	消息	超出地址范围
	错误产生的原因	载入模块文件地址超出了限定的范围。T
W300	消息	xxxxxxH - yyyyyyH 地址重叠
	错误产生的原因	输出了重叠地址的对象，地址重叠范围为： xxxxxxH 至 yyyyyyH。
W301	消息	不能初始化 RAM 区 '地址' - '地址'
	错误产生的原因	初始化数值输出到 RAM 区。
	用户采取的措施	如果汇编源程序的 CSEG 段写入 DB/DW，要么将对象更改至 DS，要么将 DB/DW 写入 DSEG 段。

表 12-4. 目标转换器错误消息 (2/2)

A900	消息	无法打开文件'文件名'
	错误产生的原因	文件不能打开。
A901	消息	无法关闭文件'文件名'
	错误产生的原因	文件不能关闭。
A902	消息	无法读取文件'文件名'
	错误产生的原因	文件不能正确读取。
A903	消息	无法存取文件'文件名'
	错误产生的原因	文件不能正确读取或写入。
A904	消息	无法写入文件'文件名'
	错误产生的原因	数据不能正确地写到输出文件中。

12.6 库管理程序错误消息

表 12-5. 库管理程序错误消息 (1/3)

A001	消息	输入文件不存在
	错误产生的原因	仅仅指定了选项，而没有指定任何输入文件。
A002	消息	输入文件过多
	错误产生的原因	输入文件总数超出了限定的范围。
A003	消息	字符串 '???' 不可识别
	错误产生的原因	在对话格式命令行中输入了选项以外的其它内容。
A004	消息	非法文件名 '文件名'
	错误产生的原因	文件名包含有操作系统不允许的字符，或者文件名的字符数超出了限定范围。
A005	消息	非法文件指定 '文件名'
	错误产生的原因	文件名中指定了非法条目项。
A006	消息	文件没有找到'文件名'
	错误产生的原因	指定的输入文件并不存在。
A007	消息	输入文件名指定重叠'文件名'
	错误产生的原因	输入文件名指定重叠
A008	消息	文件指定与'文件名'冲突。
	错误产生的原因	输入或输出文件名指定重叠。
A009	消息	无法创建文件'文件名'。
	错误产生的原因	无法创建制定的输出文件。
A010	消息	无法找到'文件名'中的目录
	错误产生的原因	输出文件名中含有并不存在的驱动器或目录。
A011	消息	非法路径 '文件名'
	错误产生的原因	在指定参数路径名的选项中输入了路径名以外的其它内容。
A012	消息	缺少参数 '选项'
	错误产生的原因	所要求的参数未指定。
A013	消息	'选项'中不需要的参数
	错误产生的原因	指定了并不需要的参数。
A014	消息	超出 '选项'范围
	错误产生的原因	指定值超出了限定范围。
A015	消息	参数字符太长 '选项'
	错误产生的原因	参数中指定的字符数超出了限定范围。
A016	消息	非法的参数 '选项'
	错误产生的原因	参数中存在语法错误。
A017	消息	'选项'中参数太多。
	错误产生的原因	参数总数超出了限定范围。
A018	消息	选项无法识别'选项'
	错误产生的原因	制定了错误选项。
A019	消息	参数文件嵌套
	错误产生的原因	参数文件中指定-F选项。
A020	消息	参数文件读错误'文件名'
	错误产生的原因	在读取参数文件时产生了错误。

表 12-5. 库管理程序错误消息 (2/3)

A021	消息	存储器定位失败
	错误产生的原因	出现存储器定位错误
A022	消息	存储器定位失败
	错误产生的原因	存储器定位失败。
A023	消息	文件名前存在非法字符','
	错误产生的原因	输入文件名之前有必需的字符','。
A024	消息	非法字符。
	错误产生的原因	发现了非法的字符或字符串
A025	消息	限定符不唯一
	错误产生的原因	修饰符的缩略类型不唯一。
A026	消息	输入重定向不明确
	错误产生的原因	'<'后没有指定任何文件名或 不止一次指定'<Δ 文件名'

A100	消息	内部错误
	错误产生的原因	产生了内部错误。
F101	消息	无效的子命令
	错误产生的原因	子命令名不正确。
F102	消息	无效语法
	错误产生的原因	子命令中参数指定不正确。
F103	消息	非法输入文件 - 不同的目标芯片 (文件: 文件名)。
	错误产生的原因	输入目标文件中目标设备指定不正确。
F104	消息	非法库文件 - 不同的目标芯片 (文件: 文件名)。
	错误产生的原因	库文件中目标设备指定不正确。
F105	消息	没有找到模块(模块: 文件名)
	错误产生的原因	库文件中指定的模块不存在。
F106	消息	模块已经不存在 (模块: 文件名)
	错误产生的原因	在更新库文件或另外一个输入文件中已经存在同名模块。
F107	消息	没有指定主库文件
	错误产生的原因	在先前的操作中没有指定更新的库文件, 但是库文件名被替换为'.'。
F108	消息	重复处理文件 (文件: 文件名)
	错误产生的原因	输入目标文件名重叠。
F109	消息	公共符号已经存在 (符号: 符号名)
	错误产生的原因	在更新库文件或其它输入文件中已经存在同名外部定义符号名。
F110	消息	文件指定冲突(文件: 文件名)
	错误产生的原因	指定的输入文件与输出文件名相同。
F111	消息	非法文件格式(文件: 文件名)
	错误产生的原因	更新库文件格式或其它输入文件格式不正确。
F112	消息	没有找到库文件(文件: 文件名)
	错误产生的原因	没有找到指定的库文件。

表 12-5. 库管理程序错误消息 (3/3)

F113	消息	没有找到目标文件(文件: 文件名)
	错误产生的原因	没有找到指定的目标文件。
F114	消息	没有空闲空间存储临时文件
	错误产生的原因	磁盘上没有足够空间用来创建临时文件。
F115	消息	磁盘空间不足
	错误产生的原因	没有足够空间用以运行操作系统。
F116	消息	子命令缓冲器满
	错误产生的原因	子命令连续行长度超出限定范围(128 × 15 个字符)。 子命令一行长度超出限定范围(128 个字符)。
F117	消息	设备文件不能使用。
	错误产生的原因	在输入文件中指定设备文件。 在输入或输出文件的命令列表里指定 CLOCK 设备。 在输出目标文件或输出库文件里指定 PRN, CON, 或 CLOCK 设备。
F118	消息	非法路径(文件: 文件名)
	错误产生的原因	指定文件的路径名不正确。
A901	消息	文件打开错误(文件: 文件名)
	错误产生的原因	文件不能打开。
A902	消息	文件读出错误(文件: 文件名)
	错误产生的原因	文件不能正确读出
A903	消息	文件写入错误(文件: 文件名)
	错误产生的原因	数据不能正写入到文件中去
A904	消息	文件查找错误(文件: 文件名)
	错误产生的原因	产生文件查找错误
A905	消息	文件关闭错误(文件: 文件名)
	错误产生的原因	文件不能关闭

12.7 表转换程序错误消息

表 12-6. 表转换程序错误消息 (1/2)

A101	消息	不是 78K/OS 文件 ‘文件名’
	错误产生的原因	输入的文件名不是 78K/OS 文件名。
W101	消息	载入模块文件比目标文件存在时间要早 ‘载入模块文件名, 目标文件名’
	错误产生的原因	载入模块文件指定比目标文件指定时间要早
A102	消息	载入模块文件为不可执行文件 ‘文件名’
	错误产生的原因	企图输入非载入模块文件, 或者试图将载入模块文件转换为在兼容主机上创建的载入模块文件。
W102	消息	载入模块文件比汇编模块文件存在时间要早 ‘载入模块文件名, 汇编列表文件名’
	错误产生的原因	载入模块文件指定比汇编列表文件指定时间要早。
A103	消息	载入模块文件有重定位数据‘文件名’
	错误产生的原因	没有指定载入模块文件地址。
W103	消息	汇编列表文件有错误语句‘文件名’
	错误产生的原因	汇编列表中有错误存在。
A104	消息	目标文件为可执行文件 ‘文件名’
	错误产生的原因	目标文件为可执行格式
W104	消息	在汇编列表文件‘段名’中没有找到段名。
	错误产生的原因	在汇编列表中没有找到目标文件段名。
A105	消息	在载入列表文件‘段名’中没有找到段名。
	错误产生的原因	在载入模块文件中没有找到目标文件段名。
W105	消息	段数据长度不同‘段名’
	错误产生的原因	汇编列表文件中的段数据长度与目标文件中的段数据长度不同。
	程序处理	忽略多余的段数据并继续执行处理。
A106	消息	在目标文件‘文件名’中没有找到段数据。
	错误产生的原因	在目标文件中没有找到汇编列表文件段名。
A107	消息	内存空间不足
	错误产生的原因	没有足够内存空间用以程序运行。
A108	消息	载入模块文件没有符号数据 ‘载入模块名’
	错误产生的原因	在连接器中指定了-NG 选项, 因此不能输出载入模块文件中的符号数据。
A109	消息	覆盖文件不能打开‘路径名’
	错误产生的原因	汇编覆盖文件不能打开。
A110	消息	非法汇编列表文件 ‘文件名’
	错误产生的原因	输入汇编列表文件类型不是汇编列表文件。

表 12-6. 表转换程序错误消息(2/2)

A901	消息	打开'文件名'文件时产生错误
	错误产生的原因	文件不能打开。
A902	消息	读出'文件名'文件时产生错误
	错误产生的原因	文件不能正确读出。
A903	消息	写入'文件名'文件时产生错误
	错误产生的原因	数据不能正写入到文件中。
A904	消息	查找'文件名'文件时产生错误
	错误产生的原因	产生文件查找错误
A999	消息	内部错误
	错误产生的原因	程序内部错误

12.8 PM + 错误消息

本节说明 PM + 帮助文件中没有描述的错误消息。需要详细了解其它 PM+ 错误消息，请参阅 PM+ 的在线帮助。

<1> DLL 中显示的结构化汇编器(ST78K0S)的错误消息

!	消息	找不到环境变量 PATH 中显示的 ST78K0S.EXE
	错误产生的原因	在环境变量 PATH 显示的目录下没有注册 ST78K0S.EXE 执行格式。
	用户采取的措施	将 ST78K0S.EXE 及其它一些与 ST78K0S 相关的文件放入环境变量 PATH 显示的目录下。
	按钮	点击 OK 按钮，关闭该消息对话框
×	消息	存储空间不足
	错误产生的原因	没有足够的存储空间
	用户采取的措施	将其它应用程序关闭，再试一次
	按钮	点击 OK 按钮，关闭该消息对话框
×	消息	无法锁定存储器
	错误产生的原因	没有足够的存储器空间，或者 Windows 系统可能发生故障。
	用户采取的措施	要么将其它应用程序关闭，再试一次；要么重新启动 Windows。
	按钮	点击 OK 按钮，关闭该消息对话框

<2> DLL 中显示的汇编器(RA78K0S)的错误消息

!	消息	找不到环境变量 PATH 中显示的 RA78K0S.EXE
	错误产生的原因	在环境变量 PATH 显示的目录下没有注册 RA78K0S.EXE 执行格式。
	用户采取的措施	将 RA78K0S.EXE 及其它一些与 RA78K0S 相关的文件放入环境变量 PATH 显示的目录下。
	按钮	点击 OK 按钮，关闭该消息对话框
×	消息	存储空间不足
	错误产生的原因	没有足够的存储空间
	用户采取的措施	将其它应用程序关闭，再试一次
	按钮	点击 OK 按钮，关闭该消息对话框
×	消息	无法锁定存储器
	错误产生的原因	没有足够的存储器空间，或者 Windows 系统可能发生故障。
	用户采取的措施	要么将其它应用程序关闭，再试一次；要么重新启动 Windows。
	按钮	点击 OK 按钮，关闭该消息对话框

<3> DLL 中显示的连接器(LK78K0S)的错误消息

!	消息	找不到环境变量 PATH 中显示的 LK78K0S.EXE
	错误产生的原因	在环境变量 PATH 显示的目录下没有注册 LK78K0S.EXE 执行格式。
	用户采取的措施	将 LK78K0S.EXE 及其它一些与 LK78K0S 相关的文件放入环境变量 PATH 显示的目录下。
	按钮	点击 OK 按钮，关闭该消息对话框
×	消息	存储空间不足
	错误产生的原因	没有足够的存储空间
	用户采取的措施	将其它应用程序关闭，再试一次
	按钮	点击 OK 按钮，关闭该消息对话框
×	消息	无法锁定存储器
	错误产生的原因	没有足够的存储器空间，或者 Windows 系统可能发生故障。
	用户采取的措施	要么将其它应用程序关闭，再试一次；要么重新启动 Windows。
	按钮	点击 OK 按钮，关闭该消息对话框

<4> DLL 中显示的目标转换器(OC78K0S)的错误消息

!	消息	找不到环境变量 PATH 中显示的 OC78K0S.EXE
	错误产生的原因	在环境变量 PATH 显示的目录下没有注册 OC78K0S.EXE 执行格式。
	用户采取的措施	将 OC78K0S.EXE 及其它一些与 OC78K0S 相关的文件放入环境变量 PATH 显示的目录下。
	按钮	点击 OK 按钮，关闭该消息对话框
×	消息	存储空间不足
	错误产生的原因	没有足够的存储空间
	用户采取的措施	将其它应用程序关闭，再试一次
	按钮	点击 OK 按钮，关闭该消息对话框
×	消息	无法锁定存储器
	错误产生的原因	没有足够的存储器空间，或者 Windows 系统可能发生故障。
	用户采取的措施	要么将其它应用程序关闭，再试一次；要么重新启动 Windows。
	按钮	点击 OK 按钮，关闭该消息对话框

附录 A 示例程序

如下所描述的是 RA78K0S 软件包所附的示例程序列表。

A.1 K0smain.asm

```
NAME SAMPMM
;*****
;
;   HEX -> ASCII 转换程序
;
;   主程序
;
;*****

PUBLIC MAIN, START
EXTRN CONVAH
EXTRN @_STBEG

DATA DSEG saddr
HDTSA: DS 1
STASC: DS 2

CODE CSEG AT 0H
MAIN: DW START

    CSEG
START:

;芯片初始化
MOVW AX, @_STBEG
MOVW SP, AX

MOV HDTSA, #1AH
MOVW HL, #HDTSA ;将 2 位十六进制代码数据置 HL 寄存器

CALL !CONVAH ;进行 ASCII <- HEX 转换
;将转换后的 ASCII 输出至 BC 寄存器中去
;将 ASCII 码放入存储表 DE 中

MOVW DE, #STASC
MOV A, B
MOV [DE], A
INCW DE
MOV A, C
MOV [DE], A

BR $$

END
```

A.2 K0ssub.asm

```

NAME SAMPS
;*****
;
;   HEX -> ASCII 转换程序
;   子程序
;
; 输入条件: (HL) <- 2 位十六进制代码
; 输出条件: BC 寄存器 <- 2 位 ASCII 代码
;
;*****

PUBLIC CONVAH

CSEG
CONVAH:
MOV  A, [HL]
ROR  A, 1
ROR  A, 1
ROR  A, 1
ROR  A, 1
AND  A, #0FH    ;装入待转换的十六进制高位数码
CALL !SASC
MOV  B, A       ;存储转换结果

XOR  A, A
XCH  A, [HL]
AND  A, #0FH    ;装入待转换的十六进制低位数码
CALL !SASC
MOV  C, A       ;存储转换结果

RET

;*****
;   子程序 ASCII 码转化
;
; 输入 Acc (低 4 位) <- 十六进制码
; 输出 Acc    <- ASCII 码
;*****

SASC:
CMP  A, #0AH    ;检查十六进制码是否大于 9
BC   $SASC1
ADD  A, #07H    ;修正值(+7H)
SASC1:
ADD  A, #30H    ;修正值(+30H)
RET

END

```

A.3 test1.s

```
EXTRN SEARCH, STABLE
EXTRN @_STBEG
PUBLIC MAIN, START
;*****
;      字符串数据搜索
;*****
SDATA:
    DB 04,12H,34H,56H,78H
;
;
CODE CSEG AT 0H
START: DW MAIN

MAIN:
    MOVW AX, @_STBEG
    MOVW SP, AX
    DE = #STABLE
    HL = #SDATA
    CALL !SEARCH
    if_bit (!CY)
SLI:
    repeat
    until (forever)
else
SERR:
    repeat
    until (forever)
endif
END
```

A.4 test2.s

```
#include "testinc.s"
```

```
PUBLIC SEARCH
```

```
CSEG
```

```

;*****
;* 数据搜索 *
;* 输入 HL 搜索数据地址置 HL 寄存器中 *
;* DE 表起始地址置 DE 寄存器中 *
;* 输出 CY = 1 未找到 *
;* CY = 0 找到 (DE<-表地址) *
;*****

```

```
SEARCH:
```

```

while ([DE]! = #0) (A)
    BC = #0
    A = [DE]
    C = A
    PUSH HL
    PUSH DE
    while ([DE] == [HL]) (A)
        DE++
        HL++
        if (C == #0) (A)

            POP DE
            POP HL
            CLR1 CY
            RET
        endif
        C--
    endwhile
    POP DE
    POP HL
    A = [DE]
    A += E
    E = A
    if (CY)
        D++
    endif
endwhile
SET1 CY
RET
END

```

A.5 testinc.s

PUBLIC STABLE

```
*****  
; 数据表  
*****
```

CSEG

STABLE:

```
DB 03, 12H, 34H, 78H  
DB 04, 55H, 66H, 77H, 88H  
DB 05, 12H, 34H, 56H, 78H, 10H  
DB 03, 12H, 34H, 56H  
DB 04, 12H, 34H, 0AH, 78H  
DB 04, 12H, 34H, 56H, 70H  
DB 04, 12H, 34H, 56H, 78H  
DB 01, 0ABH  
DB 02, 34H, 78H  
DB 00
```

A.6 st.bat

```
echo off
cls
set          LEVEL=0

if "%1" == "" goto ERR_BAT

st78k0s -C%1 test1.s
ra78k0s test1.asm
if errorlevel 1 set LEVEL=1
st78k0s -C%1 test2.s
ra78k0s test2.asm
if errorlevel 1 set LEVEL=1
if %LEVEL% == 1 echo Assemble error !!
if %LEVEL% == 1 goto END

cls
lk78k0s test1.rel test2.rel -s -otest.lmf -ptest.map
if errorlevel 1 echo Link error !!
if errorlevel 1 goto END

cls
oc78k0s test
if errorlevel 1 echo Object conversion error !!
if errorlevel 1 goto END

cls
set LEVEL=0
lc78k0s -ltest.lmf -rtest1.rel test1.prn
if errorlevel 1 set LEVEL=1
lc78k0s -ltest.lmf -rtest2.rel test2.prn
if errorlevel 1 set LEVEL=1
if %LEVEL% == 1 echo List conversion error !!
if %LEVEL% == 1 goto END

cls
echo No error.
goto END

:ERR_BAT

echo Usage : st.bat chiptype

:END

echo on
```

附录 B 使用注意事项列表

要执行 RA78K0S 必需要有设备文件。而 RA78K0S 软件包并不包含有设备文件，必需单独购买。

(1) 设备文件

要执行 RA78K0S 必需要有设备文件。而 RA78K0S 软件包并不包含有设备文件，必需单独购买。

(2) 存储器指令

每个设备缺省存储区名不可擦除，而没有用作缺省存储区的大小必需为零。

需要了解缺省存储区名，请参阅各设备的设备文件的**使用注意事项**。

务请注意，由于有些段定位至缺省区,所以在改变存储区名称时必需多加留心。

(3) 调试操作

当调试数据由 C 编译器及结构化汇编器预处理器输出时，然后就会被编译或进行结构化汇编，所以当在汇编输出汇编源程序时，请指定-NGA 选项以避免输出调试数据。如果输出了调试数据，那么就不能在 C 编译器或结构化汇编器源程序级进行调试。

(4) 存储器初始化指令

即使存储器初始化指令(DW 或 DB)写入数据段(DSEG)，也会输出代码。

当定位 ROM 代码时，如果代码不是存放在内部 ROM 地址单元内，则会产生错误。

(5) SFR 名及 EQU 定义

虽然 SFR 名可以指定为伪指令 EQU 的操作数，但是当 saddrr 存储区外的 SFR 名指定为 PUBLIC 时，也会产生汇编错误。

(6) CC78K0S

当使用由 CC78K0S 输出的汇编编译源程序来执行 C 语言级源程序时，有几点必需注意。

需要了解详细信息，请参阅**C 编译器软件包附带的文档 (使用注意事项)**。

(7) ID78K0S-NS 及 SM78K0S

当用 ID78K0S-NS 和 SM78K0S 进行调试时，要使用 D78K0S-NS 和 SM78K0S 限制范围内的符号数和行数。

需要了解详细信息，请参阅**调试器/仿真器软件包附带的文档 (使用注意事项)**。

(8) 当使用网络时

如果创建临时文件的目录放置到网络共享的文件系统中去，可能会产生文件冲突，从而导致异常操作。可以通过设置适当的选项及环境变量来避免这种情况发生。

(9) 目标转换器操作设定

当使用目标转换器选项-U 指定起始地址时，地址填充会从放置起始地址或代码的最小地址处开始。SFR 区(FF00H 至 FFFFH)不执行地址填充。

语法: -U 补充值 [, [起始],大小]
[]内的值可以省略。

附录 C 选项列表

在本附录中，总结了程序选项并列入表格中。

在开发程序时，请参考这些程序选项。

该选项列表也可以作索引之用。

C.1 结构化汇编器选项列表

序号	分类	格式	功能	与其它选项之间的关系	忽略指定时有关说明	参考页码
1	设备类型指定	-C 型设备	指定目标设备的类型	独立选项程序	不可省略	p.65
2	字符符号指定	-SC 字符	指定字符的最后一个字符	独立选项程序	-SCP	p.66
3	符号定义指定	-D 符号 [数值]	指定用于#IFDEF 等指令的符号	独立选项程序	无	p.67
4	制表符指定	-WT 数值,数值,字值	指定转换指令的输出位置	独立选项程序	-WT2, 3, 4	p.68
5	包含文件路径指定	-I 路径名 (多参数 OK)	由指定路径读取包含文件	独立选项程序	由环境变量 'INC78K0S' 指定路径	p.69
6	辅助源文件指定	-O [文件名]	指定辅助源文件名	独立选项程序	-O [输入文件名.ASM]	p.70
7	错误列表文件指定	-E [文件名]	输出错误列表文件	独立选项程序	-E [输入文件名.EST]	p.71
8	参数文件指定	-F 文件名	由指定文件输入输入文件名和选项	独立选项程序	只能由命令行输入选项和文件名	p.72
9	调试数据输出指定	-GS	指定结构化汇编程序源程序级调试数据的输出	如果同时指定-GS 选项和-NGS 选项，则最后指定的选项优先。	-GS	p.73
		-NGS	使选项 -GS 不可用			

序号	分类	格式	功能	与其它选项之间的关系	忽略指定时有关说明	参考页码
10	辅助源文件强制输出指定	-J	强制输出辅助源文件	独立选项程序	禁止强制输出	p.74
11	汉字码指定	-ZS	在注释内描述的汉字码翻译为 shift JIS 码	如果同时指定 -ZS, -ZE, 及 -ZN 选项, 则最后指定选项优先	在 Windows 中/ HP-UX: -ZS 在 SunOS, Solaris 中: -ZE	p. 75
		-ZE	在注释内描述的汉字码翻译为 EUC 码			
		-ZN	在注释内描述的字符不解释为汉字码			
12	设备文件搜索路径指定	-Y 路径名	由指定路径读取设备文件	独立选项程序	用于启动 ST78K0S.路径 <..\dev>	p.76
13	帮助指定	--	在显示器(控制台)上显示帮助信息.	所有其它选项均不可用	不显示	p.77

C.2 汇编器选项列表

序号	分类	格式	功能	与其它选项之间的关系	忽略指定时有关说明	参考页码
1	设备类型指定	-C 型设备	指定目标设备的类型	独立选项程序	不可省略	p.91
2	目标文件输出指定	-O [文件名]	指定目标文件的输出	如果同时指定 -O 选项及 -NO 选项, 则最后指定选项优先	-O [输入文件名.REL]	p.92
		-NO	指定不输出目标文件			
3	强制目标文件输出指定	-J	目标文件强制输出	如果同时指定 -J 选项及 -NJ 选项, 则最后指定选项优先	-NJ	p.93
		-NJ	使 -J 选项不可用			
4	调试数据输出指定	-G	指定将局部符号数据添加至目标文件中	如果同时指定 -G 选项及 -NG 选项, 则最后指定选项优先	-G	p.94
		-NG	使 -G 选项不可用			
		-GA	指定将源程序调试数据添加至目标文件中	如果同时指定 -GA 选项及 -NGA 选项, 则最后指定选项优先	-GA	p.95
		-NGA	使 -GA 选项不可用			
5	包含文件读取路径指定	-I 路径名 [, 路径名] ... (可以指定两个或多个路径名)	指定从指定路径输入包含文件.	独立选项程序	由环境变量 (INC78K0S) 指定路径	p.96
6	汇编列表文件输出指定	-P [文件名]	输出汇编列表文件	如果同时指定 -P 选项及 -NP 选项, 则最后指定选项优先	-P [输入文件名.REL]	p.97
		-NP	不输出汇编列表文件			
7	汇编列表文件数据指定	-KA	将汇编列表输出至汇编列表文件中.	如果同时指定 -KS 选项及 -KX 选项, 则忽略 -KS 选项。	-KA	p.98
		-NKA	不输出汇编列表文件			
		-KS	将后随符号列表的汇编列表输出至汇编列表文件中	如果 -KA 选项及 -NKA 选项同时指定或 -KS 选项及 -NKS 选项同时指定或 -KX 选项及 -NKS 选项同时指定 -NKX, 则最后指定选项优先。	-NKS	p.100
		-NKS	使 -KS 选项不可用.			
		-KX	将后随交叉参考列表的汇编列表输出至汇编列表文件中	如果一起指定了选项 -NKA, -NKS 及 -NKX, 则 -P 选项不可用.	-NKX	p.101
		-NKX	使 -KX 选项不可用.			

序号	分类	格式	功能	与其它选项之间的关系	忽略指定时有关说明	参考页码
8	汇编列表文件格式指定	-LW [字符数]	改变汇编列表文件中一行可打印的字符数目	如果指定 -NP 选项, 则-LW 选项不可用	-LW132	p.103
		-LL [行数]	改变汇编列表文件中一页可打印的行数目.	如果指定-NP 选项, 则-LL 选项不可用	-LL66	p.105
		-LH 字符串	指定汇编列表文件标题栏可打印的字符串.	如果指定-NP 选项, 则-LH 选项不可用	无	p.106
		-LT [字符数]	指定标签页中的字符数	如果指定-NP 选项, 则-LT 选项不可用	-LT8	p.110
		-LF	在汇编列表文件末端插入一换页 (FF)码	如果同时指定-LF 选项 及 -NLF 选项, 则最后指定选项优先	-NLF	p.112
		-NLF	使 -LF 选项不可用	如果指定-NP 选项, 则-LF 选项不可用.		
9	错误列表文件输出指定	-E [文件名]	输出错误列表文件.	如果同时指定-E 选项 及 -NE 选项, 则最后指定选项优先.	-NE	p.113
		-NE	使 -E 选项不可用.			
10	参数文件指定	-F 文件名	由指定文件输入汇编选项和输入文件名.	独立选项程序	只能由命令行指定输入选项和文件名	p.114
11	临时文件创建路径指定	-T 路径名	在指定路径创建临时文件	独立选项程序	由环境变量 TMP 指定路径	p.115
12	汉字码指定	-ZS	在注释内描述的汉字码翻译为 shift JIS 码	如果同时指定 -ZS, -ZE, 及 -ZN 选项, 则最后指定选项优先	在 Windows 中/ HP-UX: -ZS 在 SunOS, Solaris 中: -ZE	p.116
		-ZE	在注释内描述的汉字码翻译为 EUC 码			
		-ZN	在注释内描述的字符不解释为汉字码			
13	设备文件搜索路径指定	-Y 路径名	由指定路径读取设备文件	独立选项程序	用于启动 ST78K0S 路径 <..\dev>	p.117
14	符号定义指定	-D 符号名 [=数值] [, 符号名 [=数值] ...]	定义一符号	独立选项程序	无	p.118
15	帮助指定	--	在显示器(控制台)上显示帮助信息.	所有其它选项均不可用	不显示	p.118

C.3 连接器选项列表

序号	分类	格式	功能	与其它选项之间的关系	忽略指定时有关说明	参考页码
1	载入模块文件输出指定	-O [文件名]	输出载入模块文件.	如果同时指定 -O 和 -NO 选项, 则最后指定的选项优先	-O [输入文件名.LMF]	p.140
		-NO	不输出载入模块文件.			
2	强制载入模块文件输出指定	-J	强制输出载入模块文件.	如果同时指定 -J 和 -NJ 选项, 则最后指定的选项优先	-NJ	p.141
		-NJ	使选项 -J 不可用			
3	调试数据输出指定	-G	向载入模块文件输出调试数据	如果同时指定 -G 和 -NG 选项, 则最后指定的选项优先 当指定了选项-NG 时, 不论是否指定了-KP 或 -KL 选项, 公共符号列表及局部符号列表均不能输出.	-G	p.142
		-NG	使选项-G 不可用			
4	栈判定符生产指定	-S [区名称]	自动生成栈判定公共符号	如果同时指定 -S 和 -NS 选项, 则最后指定的选项优先	-NS	p.143
		-NS	使选项-S 不可用			
5	指令文件指定	-D 文件名	将指定文件输入为指令文件	独立选项程序	无	p.145
6	连接列表文件输出指定	-P [文件名]	指定输出连接列表文件	如果同时指定 -P 和 -NP 选项, 则最后指定的选项优先	-P [文件名.MAP]	p.146
		-NP	使选项-P 不可用			
7	连接列表文件数据指定	-KM	将映像列表输出至连接列表文件中去	如果同时指定 -KM 和 -NKM 选项, 则最后指定的选项优先	-KM	p.147
		-NKM	使选项 -KM 不可用.			
		-KD	将链接指令文件输出至连接列表文件中去	如果选项 -NKM, -NKP 及 -NKL 同时指定, 那么选项 -P 不可用。 如果指定了 -NKM 选项, 则选项-KD 不可用	-KD	p.149
		-NKD	使选项 -KD 不可用.			
		-KP	将公共符号列表输出至连接列表文件中去	如果同时指定 -KD 和 -NKD, -KP 和 -NKP, 或 -KL 和 -NKL 选项, 则最后指定的选项优先	-NKP	p.150
		-NKP	使选项 -KP 不可用.			
		-KL	将局部符号列表输出至连接列表文件中去	如果指定了 -NG 选项, 这时, 即使指定了 -KP 或 -KL 选项, 公共符号列表及局部符号列表均不能输出	-NKL	p.152
-NKL	使选项 -KL 不可用.					

序号	分类	格式	功能	与其它选项之间的关系	忽略指定时有关说明	参考页码
8	连接列表文件格式指定	-LL [行数]	指定连接列表文件中一页可打印的行数	如果指定 -NP 选项, 择选项 -LL 不可用	-LL66	p.154
		-LF	在连接列表文件末插入换页(FF)码.	如果同时指定 -LF 和 -NLF 选项, 则最后指定的选项优先	-NLF	p.156
		-NLF	使选项 -LF 不可用.	如果指定 -NP 选项, 则最后指定选项优先。.		
9	错误列表文件输出指定	-E [文件名]	输出错误列表文件	如果同时指定 -E 和 -NE 选项, 则最后指定的选项优先	-NE	p.157
		-NE	使选项 -E 不可用.			
10	库文件指定	-B 文件名	将特定文件作为库文件输入	独立选项程序	无	p.158
11	库文件读取路径指定	-I 路径名 [, 路径名] ... (可以指定两个或多个路径名)	由指定路径读取库文件	如果由选项-B 指定了一没有路径名的库文件, 则选项-I 不可用	由环境变量 'LIB78K0S'指定路径	p.159
12	参数文件指定	-F 文件名	由指定文件输入连接器选项和输入文件名.	独立选项程序	只有在命令行中输入该选项和输入文件名	p.160
13	临时文件创建路径指定	-T 路径名	在指定路径创建临时文件	独立选项程序	由环境变量 TMP 指定路径	p.161
14	设备文件搜索路径指定	-Y 路径名	由指定路径读取设备文件	独立选项程序	LK78K0S 启动路径<.\dev>	p.162
15	警告信息输出指定	-W [level]	指定是否向控制台输出警告信息	独立选项程序	一般错误信息输出	p.163
16	帮助指定	--	在显示器(控制台)上显示帮助信息。	所有其它选项均不可用	不显示	p.164

C.4 目标转换器选项列表

序号	分类	格式	功能	与其它选项之间的关系	忽略指定时有关说明	参考页码
1	十六进制格式目标文件输出指定	-O [文件名]	输出十六进制格式目标文件	如果同时指定 -O 和 -NO 选项, 则最后指定的选项优先	-O [输入文件名].HEX (文件类型 H1 至 H15 用于扩展空间)	p.182
		-NO	不输出十六进制格式目标文件			
2	符号表文件输出指定	-S [文件名]	输出符号表文件	如果同时指定 -S 和 -NS 选项, 则最后指定的选项优先	-S [输入文件名].SYM (文件类型 H1 至 H15 用于扩展空间)	p.183
		-NS	不输出符号表文件			
3	以目标地址次序进行排序指定	-R	以地址次序对十六进制格式目标进行排序	如果同时指定 -R 和 -NR 选项, 则最后指定的选项优先 如果指定 -NO 选项, 择选项 -R 不可用.	-NR	p.184
		-NR	使选项 -R 不可用.			
4	目标补充指定	-U 补充值 [, [起始], 大小]	将指定的补充值作为地址区目标代码输出至没有十六进制格式目标输出的地址区中去。	如果指定 -NO 选项, 择选项 -U 不可用.	无	p.185
5	错误列表文件输出指定	-E [文件名]	输出错误列表文件.	如果同时指定 -E 和 -NE 选项, 则最后指定的选项优先	-NE	p.187
		-NE	使选项 -E 不可用.			
6	参数文件指定	-F 文件名	由指定文件输入选项和输入文件名	独立选项程序	只有在命令行中指定选项和输入文件名	p.188
7	设备文件搜索路径指定	-Y 路径名	由指定路径读取设备文件	独立选项程序	OC78K0S 启动路径<..\dev>	p.189
8	帮助指定	--	在显示器(控制台)上显示帮助信息。	所有其它选项均不可用	不显示	p.190

C.5 库管理程序选项列表

序号	分类	格式	功能	与其它选项之间的关系	忽略指定时有关说明	参考页码
1	列表文件格式指定	-LW [字符数]	改变列表文件中一行可打印的字符数目	如果没有指定 LIST 子命令, 该选项不可用 如果同时指定 -LF 和 -NLF 选项, 则最后指定的选项优先	-LW132	p.203
		-LL [行数]	改变列表文件中一页可打印的行数目.		-LL66	p.204
		-LF	在列表文件末插入换页(FF)码。		-NLF	p.205
		-NFL	使选项 -LF 不可用.			
2	临时文件创建路径指定	-T 路径名	在指定路径创建临时文件	独立选项程序	由环境变量 TMP 指定路径	p.206
3	设备文件搜索路径指定	-Y 路径名	由指定路径读取设备文件	独立选项程序	<.\dev>(LB78K OS 启动路径)	p.207
4	帮助指定	--	在显示器(控制台)上显示帮助信息。	所有其它选项均不可用	不显示	p.208

C.6 表转换程序选项列表

序号	分类	格式	功能	与其它选项之间的关系	忽略指定时有关说明	参考页码
1	目标文件输入指定	-R [文件名]	输入目标文件	独立选项程序	-R [汇编列表文件名.REL]	p.230
2	载入模块文件输入指定	-L [文件名]	输入载入模块文件	独立选项程序	-L [汇编列表文件名.LMF]	p.231
3	绝对汇编列表文件输出指定	-O [文件名]	输出绝对汇编列表文件	独立选项程序	-O [汇编列表文件名.P]	p.232
4	错误列表文件输出指定	-E [文件名]	输出错误列表文件.	如果同时指定 -E 和 -NE 选项, 则最后指定的选项优先	-NE	p.233
		-NE	使选项 -E 不可用.			
5	参数文件指定	-F 文件名	由指定文件输入选项和输入文件名.	独立选项程序	只有在命令行中指定选项和输入文件名	p.234
6	帮助指定	--	在显示器(控制台)上显示帮助信息。	所有其它选项均不可用	不显示	p.235

附录 D 子命令列表

本附录以列表的形式对子命令进行了总结。
 在开发软件程序时，参考本列表有助于程序开发。
 该子命令列表也可以作索引之用。

序号	分类	格式	功能	缩略格式	参考页码
1	CREATE	CREATEΔ库文件名 [Δ处理]	创建一新的库文件	C	p.210
2	ADD	ADDΔ库文件名 Δ处理	向库文件添加一个模块。	A	p.211
3	DELETE	DELETEΔ库文件名▽ (▽模块名 [▽, ...]▽)	从库文件中删除一个模块。	D	p.212
4	REPLACE	REPLACEΔ库文件名 Δ处理	将库文件中模块用另一个模块替代。	R	p.213
5	PICK	PICKΔ库文件名▽ (▽模块名 [▽, ...]▽)	由现存的库文件中拾取指定模块。	P	p.214
6	LIST	LIST[Δ选项]Δ库文件名 [▽(▽模块名 [▽, ...]▽)]	向库文件中模块输出数据。	L	p.215
7	HELP	HELP	在显示器(控制台)上显示帮助信息。	H	p.216
8	EXIT	EXIT	退出库管理程序	E	p.217

附录 E 索引

[A]

异常中止错误	261
绝对汇编列表	56, 223
ADD	211
.ASM	58, 81
汇编列表	223, 241, 257
汇编器	14, 23, 81
AT	127

[B]

-B (LK78K0S)	158
--------------------	-----

[C]

-C (RA78K0S)	91
-C (ST78K0S)	65
COMPLETE	127
CREATE	210
交叉参考列表	244

[D]

-D (LK78K0S)	145
-D (RA78K0S)	118
-D (ST78K0S)	67
DELETE	212
.DR (LK78K0S)	124

[E]

-E (LC78K0S)	233
-E (LK78K0S)	157
-E (OC78K0S)	187
-E (RA78K0S)	113
-E (ST78K0S)	71
.ELK	124
.ELV	222
环境变量	37
.EOC	172
.ERA	81
错误列表	245, 250
.EST	58
执行过程	38, 42, 47
EXIT	217

[F]

-F (LC78K0S)	234
--------------------	-----

-F (LK78K0S)	160
-F (OC78K0S)	188
-F (RA78K0S)	114
-F (ST78K0S)	72
致命错误	261

[G]

-G (LK78K0S)	142
-G (RA78K0S)	94
-GA (RA78K0S)	95
-GS (ST78K0S)	73

[H]

帮助	216
.HEX	172

[I]

-I (LK78K0S)	159
-I (RA78K0S)	96
-I (ST78K0S)	69
INC78K0S	256
安装	32
Intel 标准 HEX 格式	173

[J]

-J (LK78K0S)	141
-J (RA78K0S)	93
-J (ST78K0S)	74

[K]

-KA (RA78K0S)	98
Kanji 码	37
-KD (LK78K0S)	149
-KL (LK78K0S)	152
-KM (LK78K0S)	147
-KP (LK78K0S)	150
-KS (RA78K0S)	100
-KX (RA78K0S)	101

[L]

-L (LC78K0S)	231
LANG78K	256
-LF (LB78K0S)	205
-LF (LK78K0S)	156
-LF (RA78K0S)	112

- LH (RA78K0S)..... 107
- .LIB 124, 195
- LIB78K0S 256
- 库 26, 195
- 连接指令 54, 126
- 连接列表文件..... 55, 246
- 连接器 24, 124
- LIST 215
- List 转换程序 27, 221
- LL (LB78K0S) 204
- LL (LK78K0S) 154
- LL (RA78K0S) 105
- .LMF 124, 172, 222
- 载入模块文件..... 55, 124, 172, 222
- 载入符号列表..... 250
- .LST 195
- LT (RA78K0S) 110
- LW (LB78K0S) 203
- LW (RA78K0S)..... 103
- [M]**
- .MAP..... 124
- 映像列表 247
- 性能最大化 29
- MEMORY 127
- 存储程序区 125
- 存储程序指令..... 126, 128
- 存储程序空间..... 125
- MERGE 127
- [N]**
- NE (LC78K0S) 233
- NE (LK78K0S)..... 157
- NE (OC78K0S)..... 187
- NE (RA78K0S)..... 113
- NG (LK78K0S) 142
- NG (RA78K0S) 94
- NGA (RA78K0S) 95
- NGS (ST78K0S)..... 73
- NJ (LK78K0S)..... 141
- NJ (RA78K0S) 93
- NKA (RA78K0S) 98
- NKD (LK78K0S)..... 149
- NKL (LK78K0S) 152
- NKM (LK78K0S)..... 147
- NKP (LK78K0S) 150
- NKS (RA78K0S) 100
- NKX (RA78K0S) 101
- NLF (LB78K0S)..... 205
- NLF (LK78K0S)..... 156
- NLF (RA78K0S) 112
- NO (LK78K0S) 140
- NO (OC78K0S) 182
- NO (RA78K0S) 92
- NP (LK78K0S) 146
- NP (RA78K0S)..... 97
- NR (OC78K0S) 184
- NS (LK78K0S) 143
- NS (OC78K0S)..... 183
- [O]**
- O (LC78K0S) 232
- O (LK78K0S) 140
- O (OC78K0S) 182
- O (RA78K0S) 92
- O (ST78K0S)..... 70
- 目标补充指定 185
- 目标转换器 25, 171
- [P]**
- .P 222
- P (LK78K0S)..... 146
- P (RA78K0S)..... 97
- 参数文件 57, 58, 61, 81, 85, 124, 134, 172,
..... 178, 222, 227
- PATH..... 256
- PICK 214
- .PLK..... 124
- .PLV..... 222
- PM + 78, 120, 166, 191, 218, 236, 258, 287
- .POC..... 172
- .PRA..... 81
- .PRN..... 81, 222
- .PST 58
- 公用符号列表 549
- [R]**
- R (LC78K0S)..... 230
- R (OC78K0S) 184
- RAM 125
- REGULAR..... 125, 127
- .REL..... 81, 124, 195, 222
- REPLACE..... 213
- ROM 125
- [S]**
- S (LK78K0S)..... 143

-S (OC78K0S)	183
示例程序	289
-SC (ST78K0S)	66
段定位指令	126, 130
SEQUENT	127
结构化汇编程序	22, 58
子命令	209
.SYM	172

[T]

-T (LB78K0S)	206
-T (LK78K0S)	161
-T (RA78K0S)	115
TMP	256

[U]

-U (OC78K0S)	185
--------------------	-----

[W]

-W (LK78K0S)	163
警告错误	261
-WT (ST78K0S)	68

[Y]

-Y (LB78K0S)	207
-Y (LK78K0S)	162
-Y (OC78K0S)	189
-Y (RA78K0S)	117
-Y (ST78K0S)	76

[Z]

-ZE (RA78K0S)	116
-ZE (ST78K0S)	75
-ZN (RA78K0S)	116
-ZN (ST78K0S)	75
-ZS (RA78K0S)	116
-ZS (ST78K0S)	75

[Symbol]

-- (LB78K0S)	208
-- (LC78K0S)	235
-- (LK78K0S)	164
-- (OC78K0S)	190
-- (RA78K0S)	118
-- (ST78K0S)	77