

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 Ver. 2.00

汇编包

操作篇

目标设备

78K0S 微控制器

[备忘录]

Windows, 和 Windows NT 是 Microsoft Corporation 在美国和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 IBM（国际商用机器）公司的注册商标。

HP9000 系列 700 and HP-UX 是惠普公司的注册商标。

SPARCstation 是 SPARC 国际（有限）公司的注册商标。

Solaris 和 SunOS 是 Sun Microsystems 的注册商标

- 本文档所刊登的内容有效期截至 2009 年 1 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。否则因本文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其他人所有的专利、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。

前言

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(U17390E)**（下文简称“语言”）

本手册的内容是针对 RA78K0S 的 Ver. 1.50 版本。

[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]

本手册由以下章节及附录组成。

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。

[如何阅读本手册]

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。

[规则]

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、段落或图标的标题。
___:	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 Ver. 2.00 C 编译器	操作篇	U17416E
	语言篇	U17415E
RA78K0S Ver. 2.00 汇编包	操作篇	本手册
	语言篇	U17390E
	结构化汇编语言	U17389E
SM+ 系统仿真器	操作篇	U18601E
	用户开放接口	U18602E
SM78K 系列 Ver.2.52 系统仿真器	操作篇	U16768E
ID78K0S-NS Ver. 2.52 集成调试器	操作篇	U16584E
ID78K0S-QB Ver. 3.00 集成调试器	操作篇	U17287E
PM+ Ver. 6.30 项目管理器		U18416E

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。

[备忘录]

目录

第一章 概述 ...	17
1.1 汇编器概述 ...	17
1.1.1 汇编器 ...	18
1.1.2 开发流程和 RA78K0S ...	19
1.1.3 可重新定位汇编器 ...	22
1.1.4 可重新定位汇编器的优点 ...	22
1.2 RA78K0S 的功能概述 ...	24
1.2.1 使用编辑器创建一个源模块文件 ...	25
1.2.2 结构化汇编预处理程序 ...	26
1.2.3 汇编器 ...	27
1.2.4 链接器 ...	28
1.2.5 目标转换器 ...	29
1.2.6 库管理程序 ...	30
1.2.7 表转换程序 ...	31
1.2.8 调试器 ...	32
1.3 程序开发前的提示 ...	33
1.3.1 RA78K0S 的最高性能 ...	33
1.4 RA78K0S 的特性 ...	36
第二章 产品概述与安装 ...	37
2.1 主机和提供介质 ...	37
2.2 安装 ...	38
2.2.1 Windows 版本的安装 ...	38
2.2.2 UNIX 版本的安装 ...	38
2.3 设备文件的安装 ...	39
2.3.1 Windows 版本的安装 ...	39
2.3.2 UNIX 版本的安装 ...	39
2.3.3 设备文件的注册表注册 ...	39
2.4 目录配置 ...	40
2.4.1 Windows 版本目录配置 ...	40
2.4.2 UNIX 版本目录配置 ...	41
2.5 卸载 ...	42
2.5.1 Windows 版本的卸载 ...	42
2.5.2 UNIX 版本的卸载 ...	42
2.6 环境设置 ...	43
2.6.1 主机 (IBM PC/AT 兼容机) ...	43
2.6.2 环境变量 ...	43
2.6.3 源文件中的 Kanji 码 ...	44
第三章 RA78K0S 的执行步骤 ...	45
3.1 执行 RA78K0S 之前 ...	45
3.1.1 示例程序 ...	45
3.1.2 示例程序的结构 ...	48
3.2 RA78K0S 的执行步骤 ...	49
3.3 ST78K0S 的执行步骤 ...	55
3.4 从命令行中的执行步骤 ...	61
3.5 使用参数文件 ...	65
第四章 结构化汇编器预处理器 ...	66
4.1 结构化汇编器预处理器的 I/O 文件 ...	67
4.2 结构化汇编器预处理器的功能 ...	68
4.3 结构化汇编器预处理器启动 ...	69
4.3.1 结构化汇编器预处理器启动 ...	69
4.3.2 执行启动和结束消息 ...	71
4.4 结构化汇编器选项 ...	73

4.4.1 结构化汇编器选项的类型 ...	73
4.4.2 结构化汇编器选项的说明 ...	73
4.5 PM+ 中的选项设置 ...	88
4.5.1 设置选项 ...	88
4.5.2 选项 ...	90
4.5.3 编辑选项对话框 ...	92
第五章 汇编器 ...	93
5.1 汇编器的 I/O 文件 ...	94
5.2 汇编器的功能 ...	95
5.3 汇编器启动 ...	96
5.3.1 从命令行启动 ...	96
5.3.2 从参数文件中启动 ...	97
5.3.3 执行启动和结束消息 ...	98
5.4 汇编器选项 ...	100
5.4.1 汇编器选项的类型 ...	100
5.4.2 汇编器选项的优先级顺序 ...	102
5.4.3 汇编器选项的说明 ...	103
5.5 PM+ 中的选项设置 ...	139
5.5.1 选项设置方法 ...	139
5.5.2 选项设置 ...	142
5.5.3 编辑选项对话框 ...	145
第六章 链接器 ...	146
6.1 链接器的 I/O 文件 ...	146
6.2 链接器的功能 ...	147
6.3 存储空间和存储区域 ...	148
6.4 链接指令 ...	149
6.4.1 指令文件 ...	149
6.4.2 存储指令 ...	151
6.4.3 程序段定位指令 ...	153
6.5 链接器启动 ...	156
6.5.1 链接器启动 ...	156
6.5.2 执行开始和结束信息 ...	157
6.6 链接器选项 ...	159
6.6.1 链接器选项类型 ...	159
6.6.2 链接器选项优先顺序 ...	161
6.6.3 链接器选项说明 ...	161
6.7 在 PM+ 中选项设置 ...	190
6.7.1 选项设置方法 ...	190
6.7.2 选项设置 ...	194
6.7.3 编辑选项对话框 ...	197
第七章 目标转换器 ...	198
7.1 目标转换器的 I/O 文件 ...	199
7.2 目标转换器的功能 ...	200
7.2.1 HEX 格式目标模块文件 ...	200
7.2.2 符号表文件 ...	202
7.3 目标转换器启动 ...	204
7.3.1 目标转换器启动 ...	204
7.3.2 执行开始和结束信息 ...	205
7.4 目标转换器选项 ...	207
7.4.1 目标转换器选项类型 ...	207
7.4.2 目标转换器选项的说明 ...	207
7.5 在 PM+ 中选项设置 ...	219
7.5.1 选项设置方法 ...	219
7.5.2 选项设置 ...	222
第八章 库管理程序 ...	224
8.1 库管理程序的 I/O 文件 ...	225
8.2 库管理程序的功能 ...	226
8.3 库管理程序的启动 ...	228
8.3.1 库管理程序的启动 ...	228

8.3.2 执行启动和结束消息 ...	231
8.4 库管理程序选项 ...	232
8.4.1 库管理程序选项的类型 ...	232
8.4.2 库管理程序选项的说明 ...	232
8.5 子命令 ...	239
8.5.1 子命令的类型 ...	239
8.5.2 子命令说明 ...	239
8.6 PM+ 中的选项设置 ...	250
8.6.1 选项设置方法 ...	250
8.6.2 选项设置 ...	252
8.7 PM+ 中处理库文件的方法 ...	254
8.7.1 处理方法 ...	254
8.7.2 项目设置 ...	256
第九章 表转换程序 ...	257
9.1 表转换程序的 I/O 文件 ...	258
9.2 表转换程序的功能 ...	259
9.3 表转换程序启动 ...	262
9.3.1 表转换程序启动 ...	262
9.3.2 执行开始和结束信息 ...	264
9.4 表转换程序选项 ...	265
9.4.1 表转换程序选项的类型 ...	265
9.4.2 表转换程序选项的说明 ...	265
9.5 在 PM+ 中选项设置 ...	273
9.5.1 选项设置方法 ...	273
9.5.2 选项设置 ...	275
第十章 程序输出列表 ...	276
10.1 结构化汇编器预处理器输出列表 ...	277
10.1.1 错误列表 ...	277
10.2 汇编器输出列表 ...	278
10.2.1 汇编列表文件头 ...	278
10.2.2 汇编列表 ...	279
10.2.3 符号列表 ...	281
10.2.4 交叉引用列表 ...	282
10.2.5 错误列表 ...	283
10.3 链接器输出列表 ...	284
10.3.1 链接列表文件头 ...	284
10.3.2 映射列表 ...	286
10.3.3 公共符号列表 ...	288
10.3.4 局部符号列表 ...	289
10.3.5 错误列表 ...	290
10.4 目标转换器输出列表 ...	291
10.4.1 错误列表 ...	291
10.5 库管理程序输出列表 ...	292
10.5.1 库数据输出列表 ...	292
10.6 表转换程序输出列表 ...	293
10.6.1 绝对汇编列表 ...	293
10.6.2 错误列表 ...	293
第十一章 提高使用 RA78K0S 效率 ...	294
11.1 提高操作效率 (EXIT 状态功能) ...	294
11.2 准备开发环境 (环境变量) ...	295
11.3 中断程序执行 ...	296
11.4 使汇编列表易于读取 ...	297
11.5 减少程序启动时间 ...	298
11.5.1 在源程序中指定控制指令 ...	298
11.5.2 使用 PM+ ...	298
11.5.3 创建参数文件和子命令文件 ...	299
11.6 目标模块库生成 ...	300
第十二章 错误消息 ...	301
12.1 错误消息概述 ...	301

12.2 结构化汇编预处理器错误消息 ...	302
12.3 汇编器错误消息 ...	308
12.4 链接器错误消息 ...	319
12.5 目标转换器错误消息 ...	327
12.6 库管理程序错误消息 ...	331
12.7 表转换器错误消息 ...	335
12.8 PM+ 错误消息 ...	339
12.8.1 结构化汇编预处理器 (ST78K0S) ...	339
12.8.2 汇编器 (RA78K0S) ...	342
12.8.3 链接器 (LK78K0S) ...	345
12.8.4 目标转换器 (OC78K0S) ...	347
12.8.5 库管理程序 (LB78K0S) ...	348
12.8.6 表转换器 (LC78K0S) ...	349
附录 A 示例程序 ...	350
A.1 k0smain.asm ...	350
A.2 k0ssub.asm ...	351
A.3 test1.s ...	352
A.4 test2.s ...	353
A.5 testinc.s ...	354
A.6 st.bat ...	355
附录 B 使用注意事项 ...	356
附录 C 选项列表 ...	358
C.1 结构化汇编器选项 ...	358
C.2 汇编器选项 ...	360
C.3 链接器选项列表 ...	363
C.4 目标转换器选项列表 ...	366
C.5 库管理程序选项列表 ...	367
C.6 列表转换程序选项列表 ...	368
附录 D 子命令列表 ...	369
附录 E 检索 ...	370

图形的列表

图形编号 标题 页码

1-1	RA78K0S 汇编包 ...	17
1-2	汇编流程 ...	18
1-3	应用微控制器产品开发流程 ...	19
1-4	软件开发流程 ...	20
1-5	RA78K0S 汇编流程 ...	21
1-6	用于调试的重汇编 ...	23
1-7	用现有模块开发程序 ...	23
1-8	用 RA78K0S 开发程序的步骤 ...	24
1-9	创建源模块文件 ...	25
1-10	结构化汇编器预处理器的功能 ...	26
1-11	汇编器的功能 ...	27
1-12	链接器的功能 ...	28
1-13	目标转换器的功能 ...	29
1-14	库管理程序的功能 ...	30
1-15	表转换程序的功能 ...	31
1-16	调试器的功能 ...	32
2-1	目录配置 ...	40
2-2	目录配置 ...	41
3-1	举例程序的结构 ...	45
3-2	RA78K0S 执行步骤 1 ...	53
3-3	RA78K0S 执行步骤 2 ...	54
3-4	ST78K0S 执行步骤 ...	60
3-5	链接指令 ...	62
4-1	结构化汇编器预处理器的 I/O 文件 ...	66
4-2	< Structured Assembler Options > 对话框 (选择 << Output >> 标签时) ...	88
4-3	< Assembler Source Options > 对话框 (选择 [Assembler Options] 按钮时) ...	89
4-4	< Structured Assembler Options > 对话框 (选择 << Others >> 标签时) ...	89
4-5	< Edit Option > 对话框 ...	92
4-6	< Add Option > 对话框 ...	92
5-1	汇编器的 I/O 文件 ...	93
5-2	< Assembler Options > 对话框 (选择 << Output1 >> 标签时) ...	139
5-3	< Assembler Options > 对话框 (选择 << Output2 >> 标签时) ...	140
5-4	< Assembler Options > 对话框 (选择 << Others >> 标签时) ...	141
5-5	< Edit Option > 对话框 ...	145
5-6	< Add Option > 对话框 ...	145
6-1	存储区名 ...	151
6-2	举例指定段分配 ...	155
6-3	< Linker Options > 对话框 (选择 << Output1 >> 标签时) ...	190
6-4	< Linker Options > 对话框 (选择 << Output2 >> 标签时) ...	191
6-5	< Linker Options > 对话框 (选择 << Library >> 标签时) ...	192
6-6	< Linker Options > 对话框 (选择 << Others >> 标签时) ...	193
6-7	< Edit Option > 对话框 ...	197
6-8	< Add Option > 对话框 ...	197
7-1	目标转换器的 I/O 文件 ...	198
7-2	Intel 标准格式 ...	201
7-3	符号值格式 ...	203
7-4	< Object Converter Options > 对话框 (选择 << Output1 >> 标签时) ...	219
7-5	< Object Converter Options > 对话框 (选择 << Output2 >> 标签时) ...	220
7-6	< Object Converter Options > 对话框 (选择 << Others >> 标签时) ...	221
8-1	库管理程序的 I/O ...	224
8-2	创建库文件的步骤 ...	227
8-3	< Librarian Options > 对话框 (选择 << Output >> 标签时) ...	250
8-4	< Librarian Options > 对话框 (选择 << Others >> 标签时) ...	251
8-5	< Library File Name > 对话框 ...	254

- 8-6 < Subcommand > 对话框 ... 255
- 9-1 表转换程序的 I/O 文件 ... 257
- 9-2 < List Converter Options > 对话框 (选择 << Output >> 标签时) ... 273
- 9-3 < List Converter Options > 对话框 (选择 << Others >> 标签时) ... 274

图表的列表

图表编号 标题 页码

1-1	结构化汇编器的最高性能 ...	33
1-2	汇编器的最高性能 ...	34
1-3	链接器的最高性能 ...	35
2-1	汇编包的提供介质 ...	37
4-1	结构化汇编器预处理器的 I/O 文件 ...	67
4-2	结构化汇编器选项 ...	73
5-1	汇编器的 I/O 文件 ...	94
5-2	汇编器选项 ...	100
5-3	汇编器选项的优先级顺序 ...	102
5-4	可写作标题的字符 ...	122
6-1	链接器的 I/O 文件 ...	146
6-2	程序段分配组 (外部 ROM 等待) ...	148
6-3	指令类型 ...	149
6-4	按照存储区名指定与存储空间名组合确定程序段位置 ...	154
6-5	链接器选项 ...	159
6-6	链接器选项的优先级顺序 ...	161
7-1	目标转换器的 I/O 文件 ...	199
7-2	符号属性的值 ...	203
7-3	目标转换器选项 ...	207
8-1	库管理程序的 I/O 文件 ...	225
8-2	库管理程序选项 ...	232
8-3	子命令 ...	239
9-1	表转换程序的 I/O 文件 ...	258
9-2	表转换程序启动时指定文件的类型 ...	262
9-3	表转换程序选项 ...	265
10-1	结构化汇编器预处理器输出列表 ...	277
10-2	错误列表输出项的说明 (启动结构化汇编器预处理器时) ...	277
10-3	汇编器输出列表 ...	278
10-4	汇编列表文件头输出项的说明 ...	278
10-5	汇编列表输出项的说明 ...	280
10-6	符号列表输出项的说明 ...	281
10-7	交叉引用输出项的说明 ...	282
10-8	错误列表输出项的说明 (启动汇编器时) ...	283
10-9	链接器输出列表 ...	284
10-10	链接列表文件头输出项的说明 ...	285
10-11	映射列表输出项的说明 ...	287
10-12	公共符号列表输出项的说明 ...	288
10-13	局部符号列表输出项的说明 ...	289
10-14	错误列表输出项的说明 (启动链接器时) ...	290
10-15	目标转换器输出项的说明 ...	291
10-16	库管理程序输出列表 ...	292
10-17	库数据输出列表输出项的说明 ...	292
10-18	表转换程序输出列表 ...	293
12-1	结构化汇编器预处理器错误信息 ...	302
12-2	汇编器错误信息 ...	308
12-3	链接器错误信息 ...	319
12-4	目标转换器错误信息 ...	327
12-5	库管理程序错误信息 ...	331
12-6	表转换程序错误信息 ...	335
12-7	由结构化汇编器预处理器 (ST78K0S) DLL 显示的错误信息 ...	339
12-8	由汇编器 (RA78K0S) DLL 显示的错误信息 ...	342
12-9	由链接器 (LK78K0S) DLL 显示的错误信息 ...	345
12-10	由目标转换器 (OC78K0S) DLL 显示的错误信息 ...	347
12-11	由库管理程序 (LB78K0S) DLL 显示的错误信息 ...	348

12-12	由表转换程序 (LC78K0S) DLL 显示的错误信息 ...	349
C-1	结构化汇编器选项 ...	358
C-2	汇编器选项 ...	360
C-3	链接器选项列表 ...	363
C-4	目标转换器选项列表 ...	366
C-5	库管理程序选项列表 ...	367
C-6	表转换程序选项列表 ...	368
D-1	子命令列表 ...	369

第一章 概述

本章说明 RA78K0S 在微处理器软件开发中的作用以及 RA78K0S 的功能。

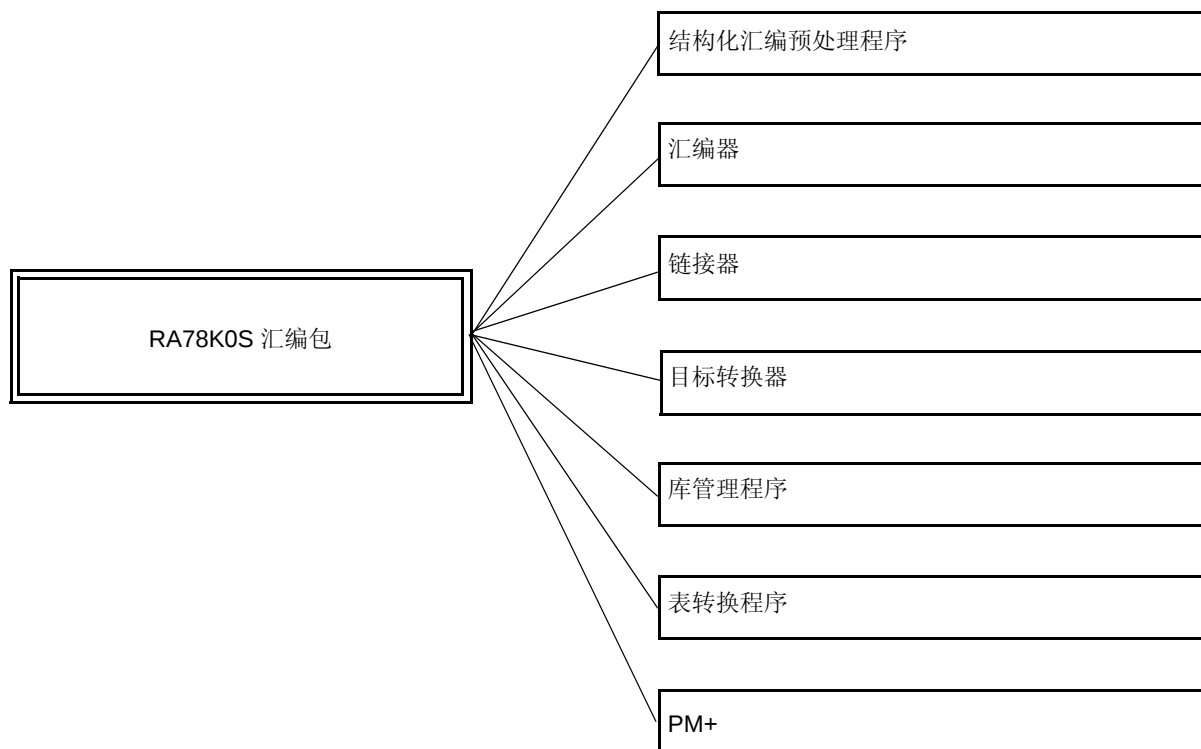
1.1 汇编器概述

RA78K0S 汇编包（以下指“RA78K0S”）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、链接器、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、链接和调试等。

图 1-1 RA78K0S 汇编包



1.1.1 汇编器

(1) 汇编语言和机器语言

汇编语言是微控制器最基本的编程语言。

在微控制器中，用为控制需要的程序和数据来完成工作。这些程序和数据必须由用户来写入到微控制器的存储区。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。但是，对于用户来说，机器语言代码难以记忆，经常产生错误。幸运的是，可以使用英文缩写或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基本编程语言系统称为汇编语言。

但是，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。

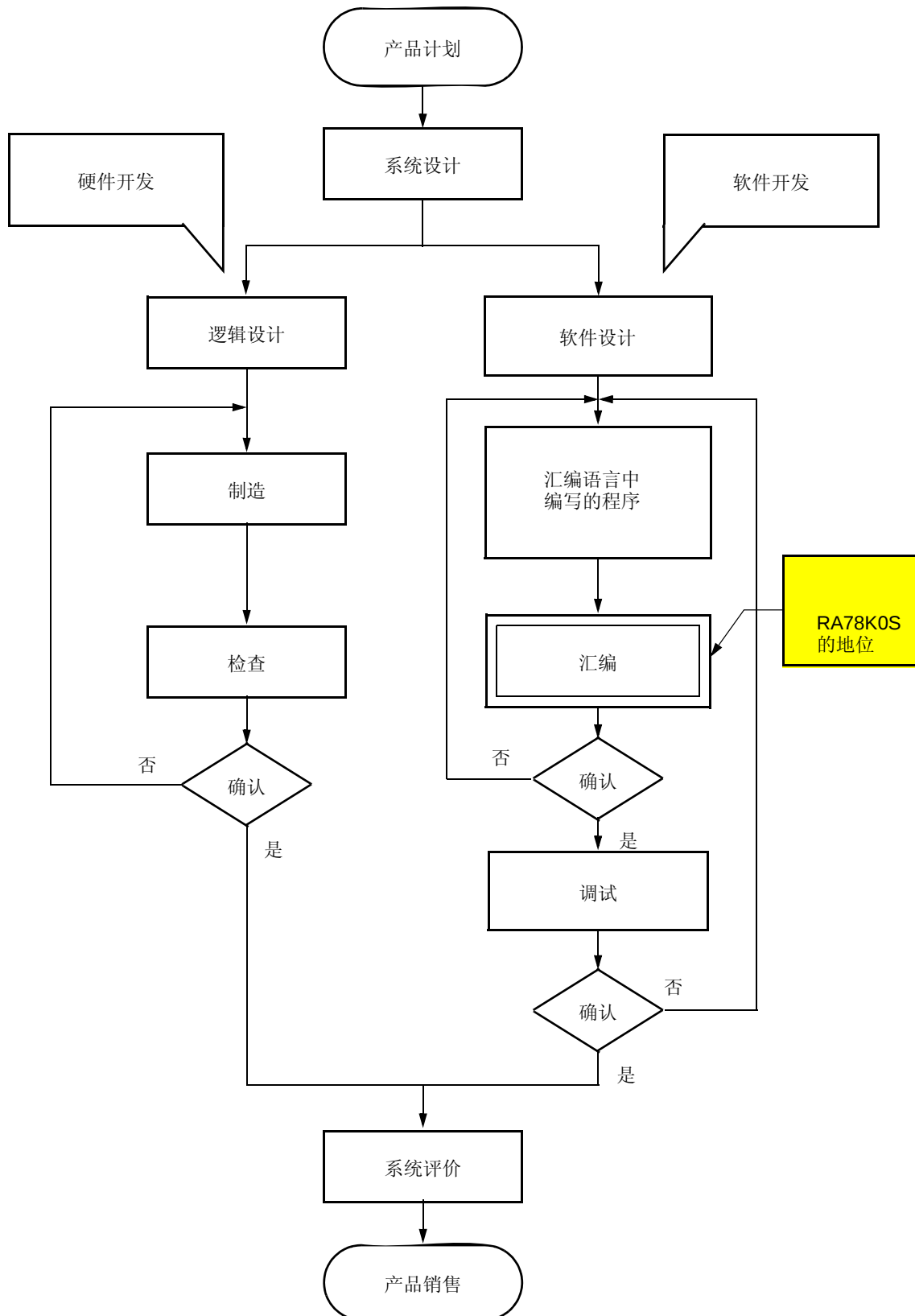
图 1-2 汇编流程



1.1.2 开发流程和 RA78K0S

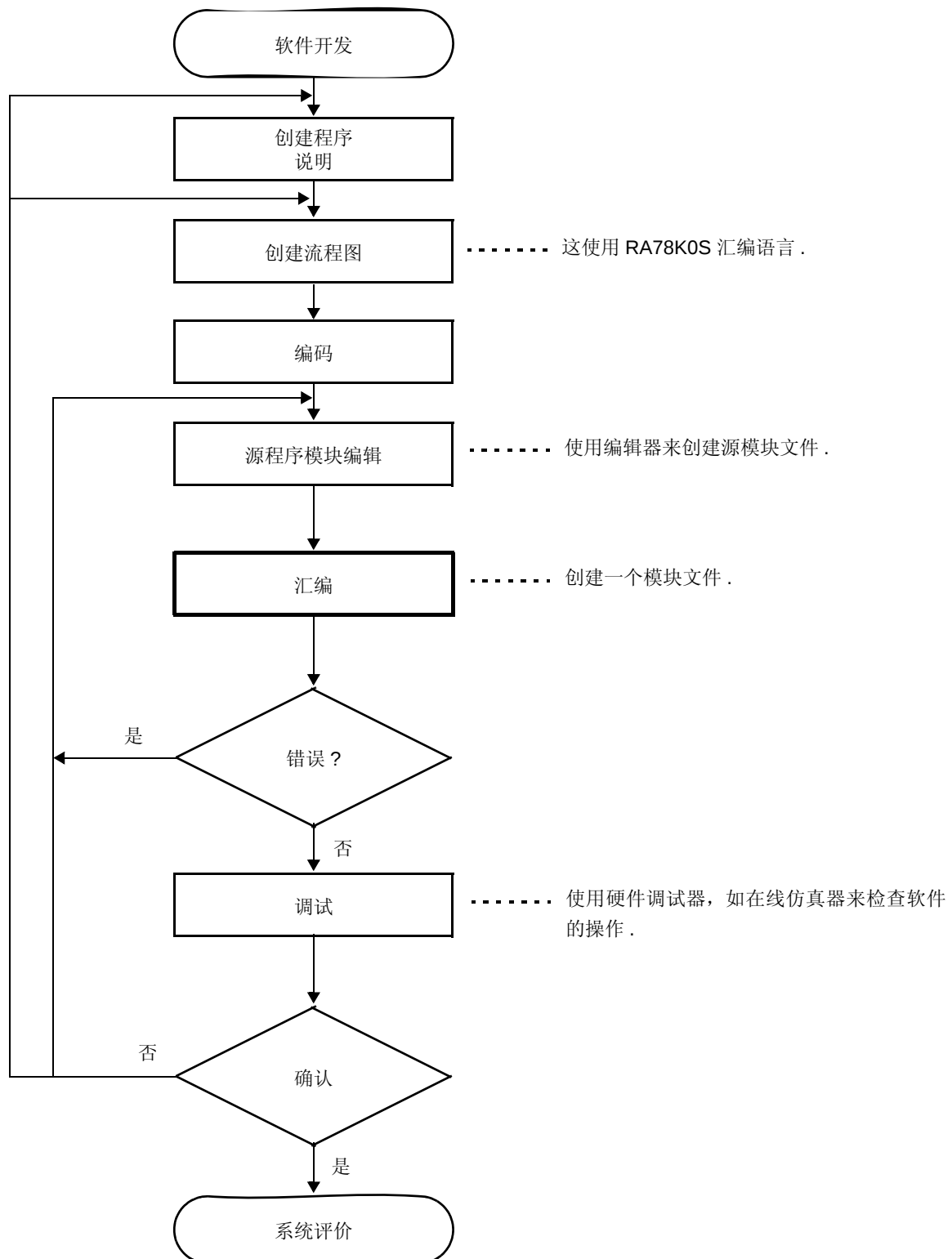
- (1) 微处理器相关产品开发与 RA78K0S 的作用
图 1-3 说明了“产品开发流程中汇编”的地位。

图 1-3 应用微处理器产品开发流程



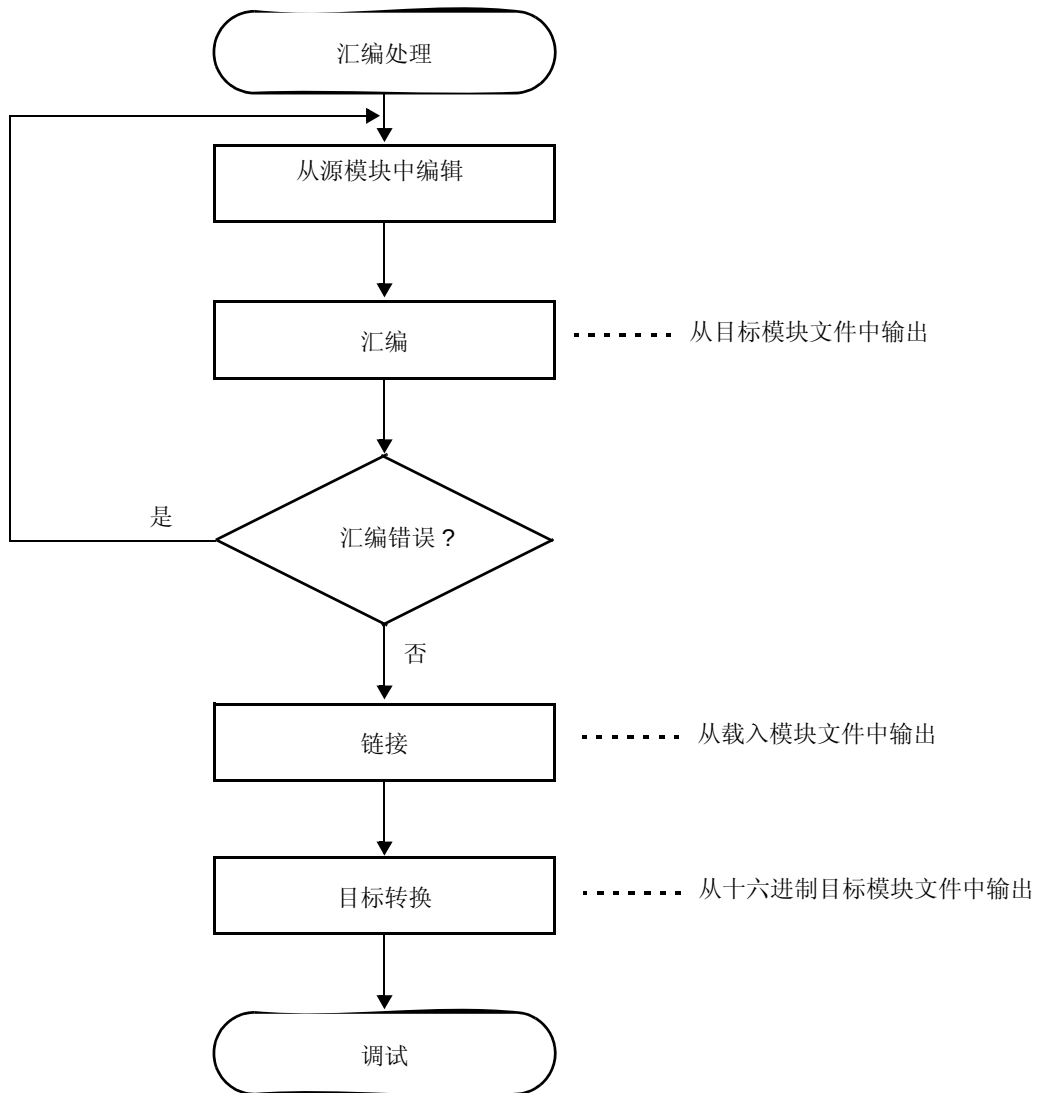
软件开发流程更为详细的说明，参见图 1-4.

图 1-4 软件开发流程



然后将 RA78K0S 应用于汇编过程。

图 1-5 RA78K0S 汇编处理



1.1.3 可重新定位汇编器

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，汇编器汇编机器语言时必须添加一些信息，以说明每条机器语言指令存储在内存中的位置。

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“可重新定位汇编器”。

- 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。

- 可重新定位汇编器

在可重新定位汇编器中，将从汇编语言汇编的机器语言指令指定到临时地址。绝对地址由称为链接器的程序来确定。

以前，当使用绝对汇编器创建一个程序时，规定程序员必须同时完成编程。但是，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。

1.1.4 可重新定位汇编器的优点

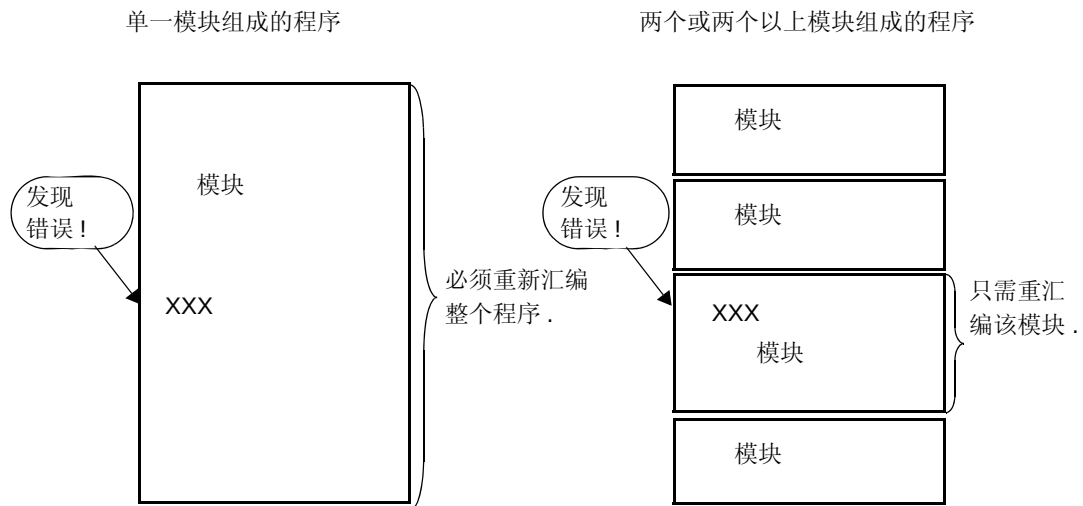
可重新定位汇编器是一种适于模块化编程的汇编器，它具有以下优点：

(1) 提高开发效率

同时编写一个大型程序是很困难的。在这种情况下，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。

此外，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。这会缩短调试时间。

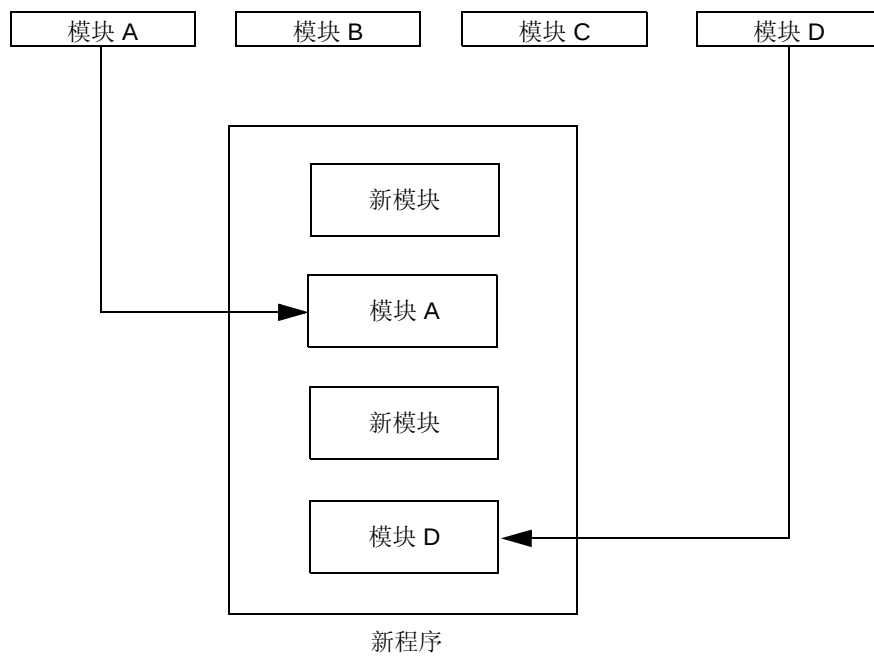
图 1-6 重汇编用于调试



(2) 资源的利用

先前已经创建的高度可靠、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精力。

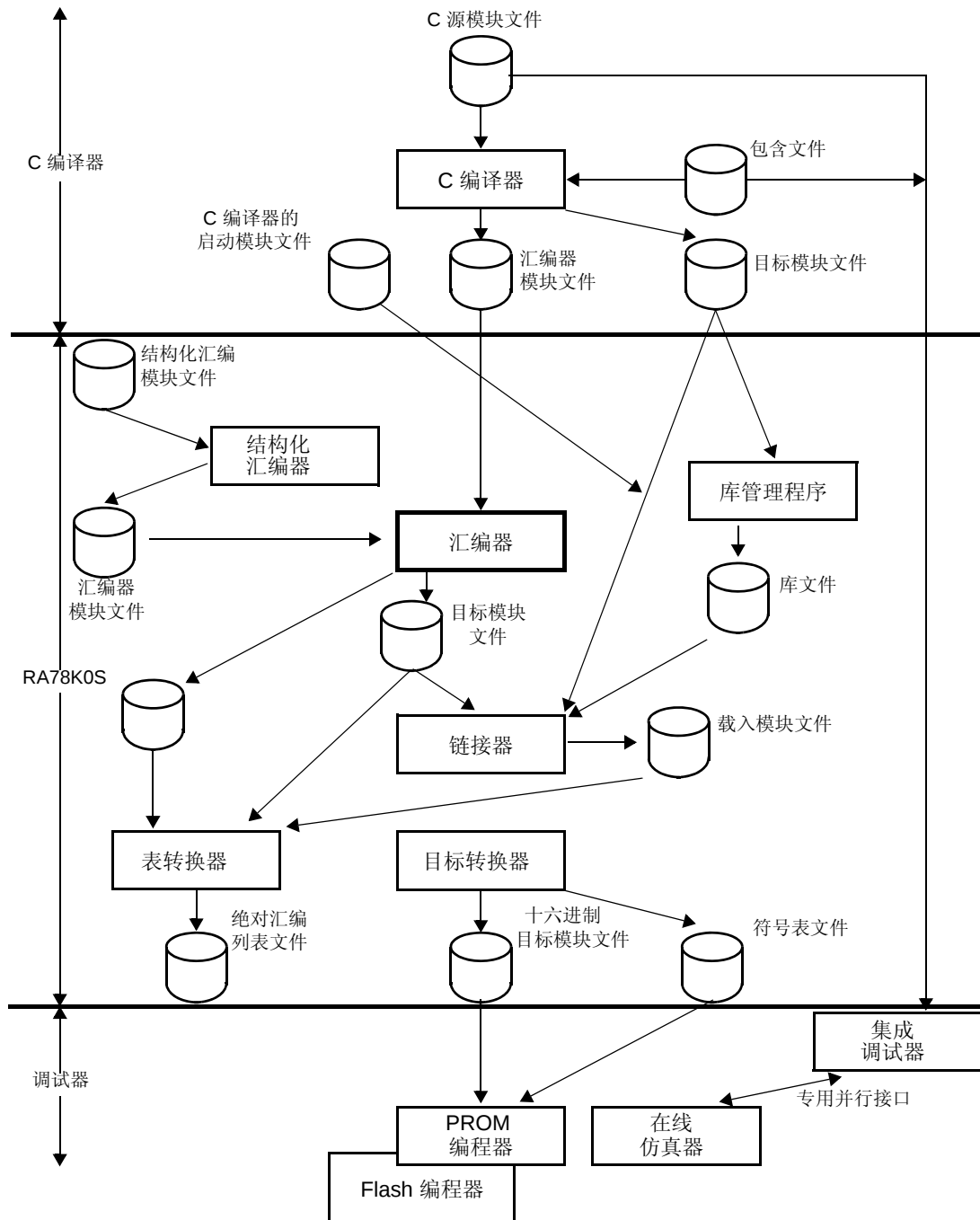
图 1-7 使用现有模块的程序开发



1.2 RA78K0S 的功能概述

通用开发程序的步骤，参见图 1-8。程序开发基本上是从遵守汇编器到链接器再到目标转换器的流程。汇编器、链接器、目标转换器及其他程序统称为“RA78K0S”。而汇编程序称为“汇编器”。

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

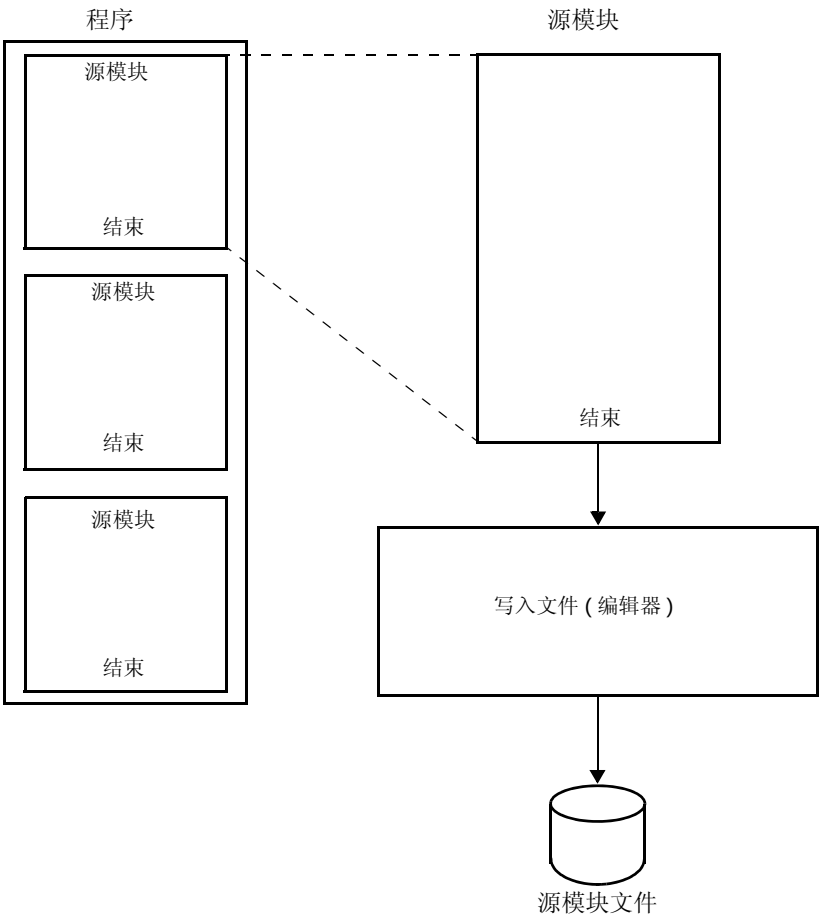
可根据功能将一个程序分为两个或多个模块。单个模块可用作一个编码单元或汇编器输入单元。

用作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

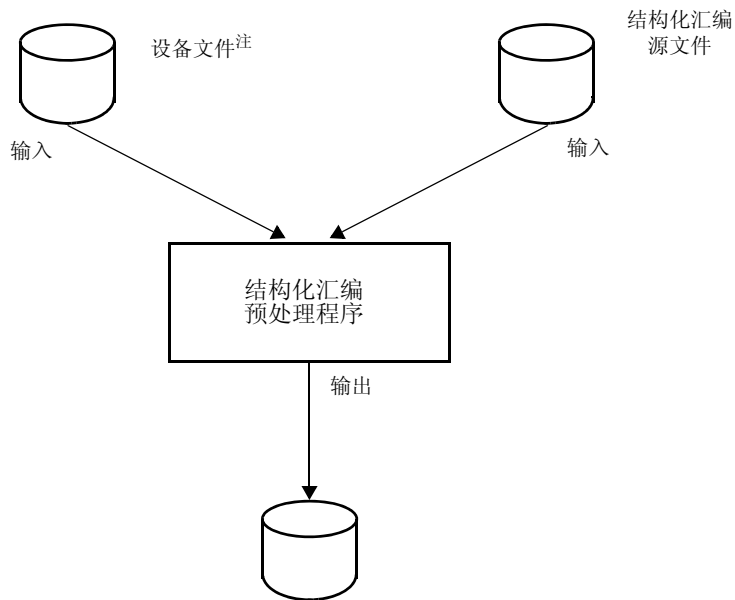


1.2.2 结构化汇编预处理程序

结构化汇编预处理程序是一个为使用汇编语言指令来进行结构化编程的程序。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更多信息，参见各个的 **RA78K0S 结构化汇编语言用户手册**。

图 1-10 结构化预处理程序的功能



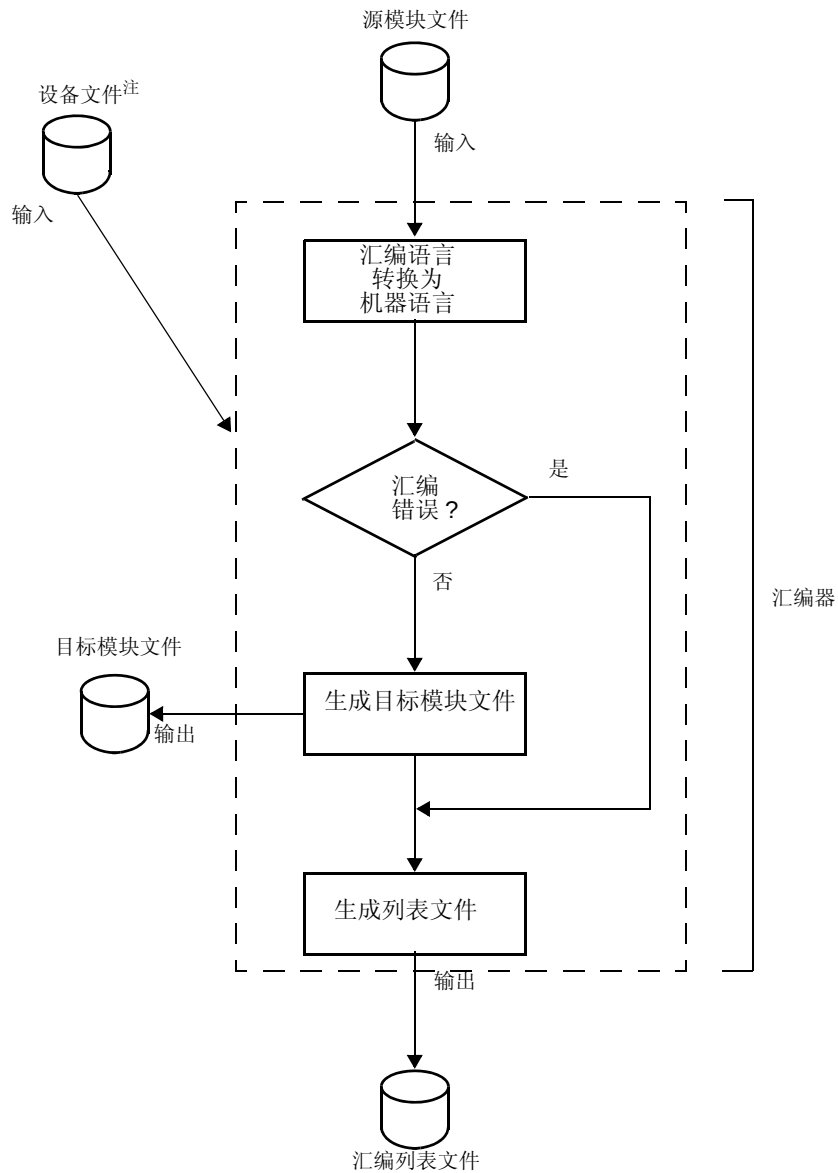
注 可通过从联机发送服务 (ODS) 下载来得到设备文件，联机发送服务可由以下网址进行访问。

<http://www.necel.com/micro/ods/eng/tool/DeviceFile/list.html>

1.2.3 汇编器

汇编器是输入源模块文件，并将汇编语言转换为一组二进制指令（机器语言）的程序。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 汇编器的功能



注 可通过从联机发送服务（ODS）下载来得到设备文件，联机发送服务可由以下网址进行访问。

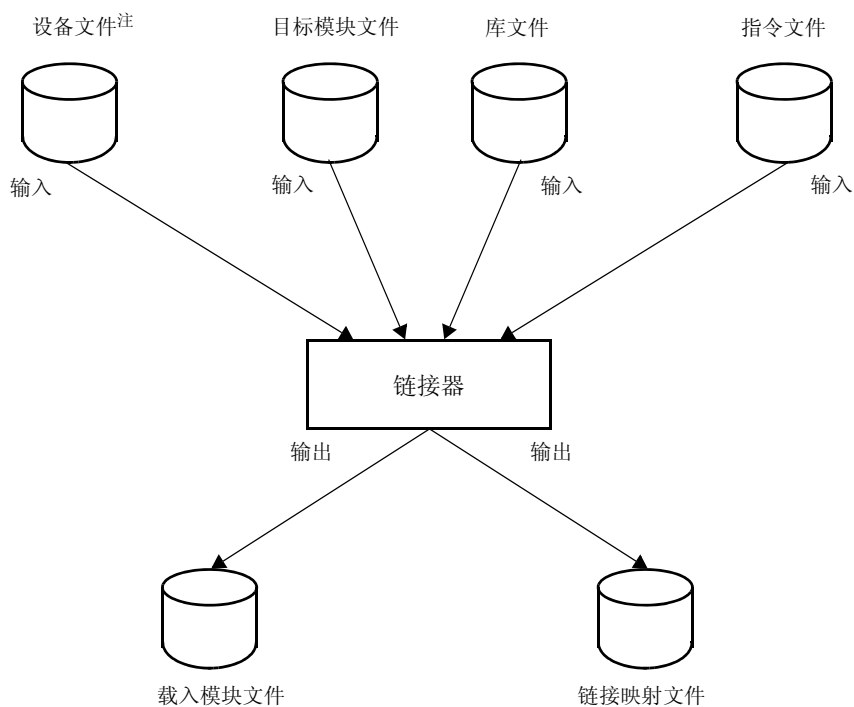
<http://www.necel.com/micro/ods/eng/tool/DeviceFile/list.html>

1.2.4 链接器

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

链接器决定输入模块中可重新定位段的单元地址。这就确定了可重新定位符号及外部引用符号的值，以便正确的值被嵌入到载入模块文件中。

图 1-12 链接器的功能



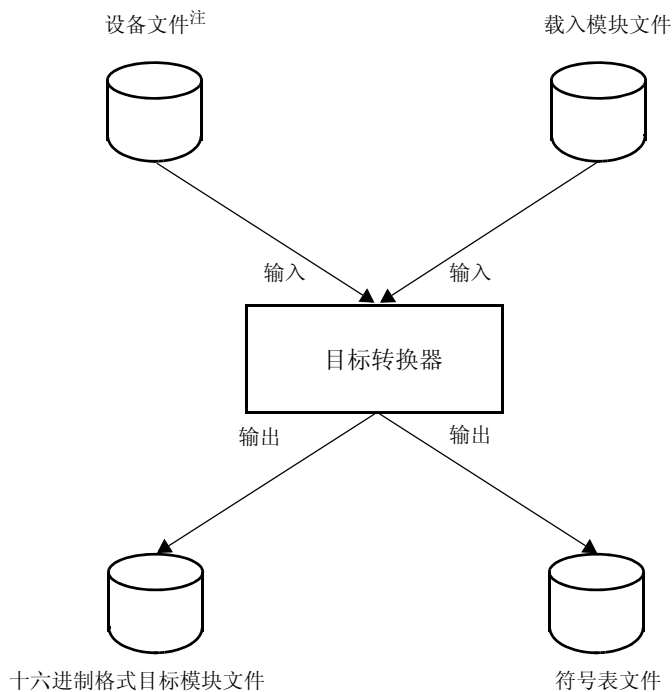
注 可通过从联机发送服务（ODS）下载来得到设备文件，联机发送服务可由以下网址进行访问。

<http://www.necel.com/micro/ods/eng/tool/DeviceFile/list.html>

1.2.5 目标转换器

目标转换器输入链接器输出的载入模块文件，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 目标转换器的功能



注 可通过从联机发送服务 (ODS) 下载来得到设备文件，联机发送服务可由以下网址进行访问。
<http://www.necel.com/micro/ods/eng/tool/DeviceFile/list.html>

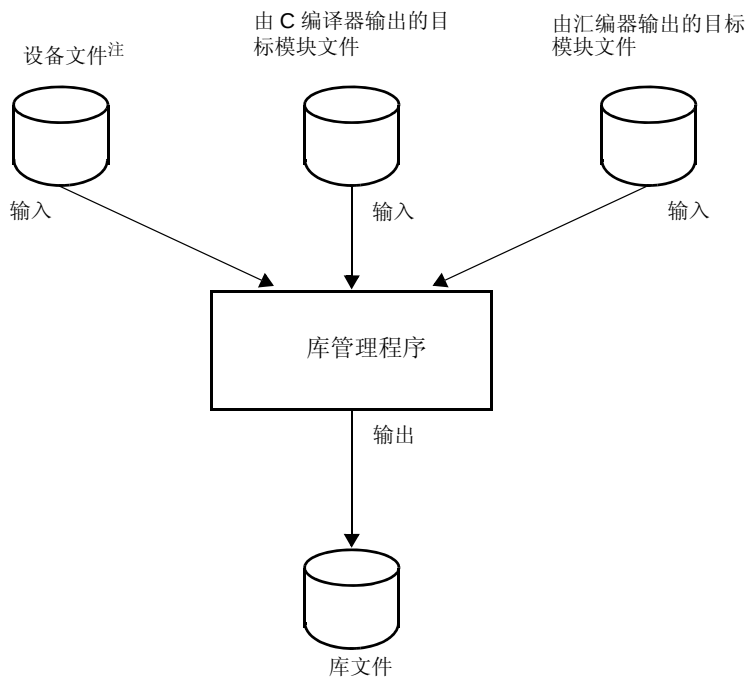
1.2.6 库管理程序

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。

链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链接模块文件。

库管理程序就是用于创建和更新库文件的程序。

图 1-14 库管理程序的功能



注 可通过从联机发送服务 (ODS) 下载来得到设备文件，联机发送服务可由以下网址进行访问。

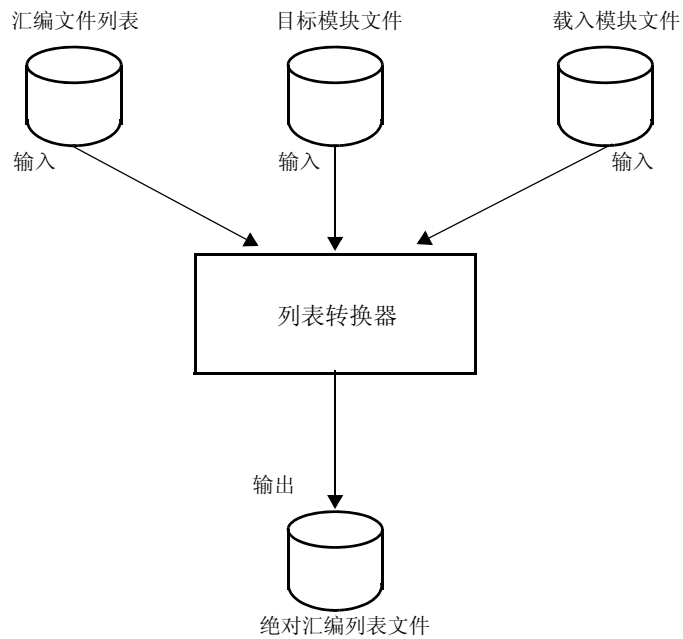
<http://www.necel.com/micro/ods/eng/tool/DeviceFile/list.html>

1.2.7 列表转换程序

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链接器输出的载入模块文件，并输出一个绝对汇编列表文件。

可重新定位汇编列表文件具有这样的优势：表中的地址和重定位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 列表转换程序功能



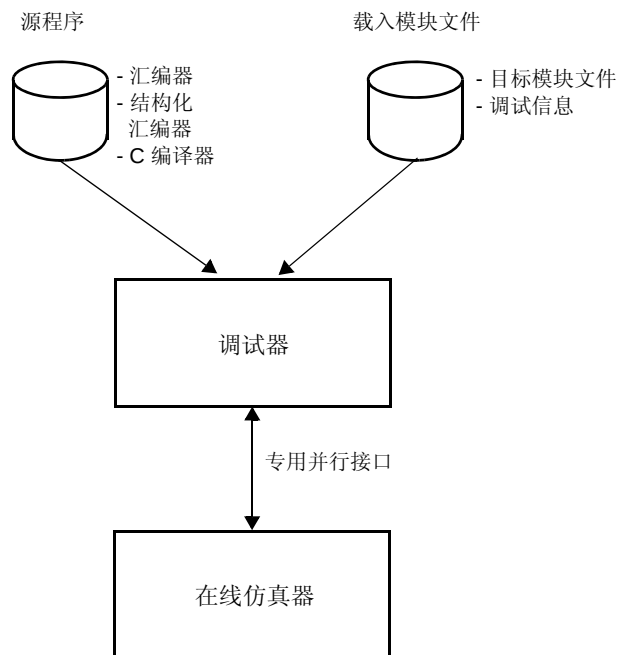
1.2.8 调试器

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。

调试器将链接器输出的载入模块文件下载到目标系统的在线仿真器（IE）。它还可以通过读取源程序文件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 调试器的功能



1.3 程序开发前的注意

开发程序之前请注意以下几点。

1.3.1 RA78K0S 的最高性能

(1) 结构化汇编器的最高性能

表 1-1 结构化汇编器的最高性能

项目	最大值
行长度（不包括 LF 或 CR）	2048 个字符
#define 指令中记录的符号数（除保留字之外）	512 个字符
#define 指令中记录的符号的字符长度	31 个符号
控制语句的嵌套层数	31 层
#ifdef 指令的嵌套层数	8 层
#defcallt 指令	32
#include 指令的嵌套	不支持
由 #define 指令重新定义的次数	31 次
一列中赋值的操作数个数	33 ^{注 1}
逻辑运算符操作数	17 ^{注 2}
由 -d 选项定义的符号个数	30
由 -i 选项指定的库文件路径的个数	64

注 1 最大值表示如下。

S1 = S2 = ... = S32 = S33

最多可插入 33 个符号和 32 个等号 (=)。

注 2 最大值表示如下。

表达式 1&& 表达式 2&& ... && 表达式 16&& 表达式 17

最多可插入 17 个表达式和 16 个 "&&" (或 "|")

(2) 汇编器的最高性能

表 1-2 汇编器的最高性能

项目	最高性能
符号数（本地 + 公用）	65535
可以输出到交叉引用表中的符号数	65534 ^{注 1}
一个宏可以引用的最大宏指令体大小	1 MB
所有宏指令体的总大小	10 MB
一个文件中的段数	256
宏以及一个文件中的包含规格	10000
宏以及一个包含文件中的包含规格	10000
重定位信息 ^{注 2}	65535
行数信息	65535
一个文件中的 BR 伪指令	32767
一个源代码行的字符数	2048 ^{注 3}
符号长度	256 个字符
分支名定义的数目 ^{注 4}	1000
分支名的字符长度 ^{注 4}	31 个字符
段名的字符长度	8 个字符
模块名的字符长度（ NAME 伪指令）	8 个字符
MACRO 伪指令中的虚参量个数	16
宏引用中实际参量的个数	16
IRP 伪指令中的实际参量个数	16
宏指令体中本地符号的个数	64
展开宏中本地符号的总数	65535
宏中的嵌套层数（宏引用， REPT 伪指令， IRP 伪指令）	8 层
由 TITLE 控制指令， -lh 选项指定的字符个数	60 ^{注 5}
由 SUBTITLE 控制指令指定的字符个数	72
一个文件中包含文件的嵌套层数	8 层
条件汇编嵌套层数	8 层
由 -i 选项指定的包含文件路径的个数	64
可由 -d 选项定义的符号个数	30

注 1 不包括模块名和段名。

使用存储器。如果没有存储器，则使用文件

注 2 如汇编器不能解析符号值时将传递给链接器的信息。

例如，如果 MOV 指令引用一个外部引用符号，则在 .rel 文件中会生成两段重定位信息。

注 3 包含 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，同时忽略之后的字符。

注 4 分支名由 SET/RESET 伪指令设置为真 / 假，并通过 \$IF 等使用。

注 5 如果在汇编列表文件一行中可以指定的字符数最多为 119，则该值为 "X - 60" 或更少。

(3) 链接器的最高性能

表 1-3 链接器的最高性能

项目	最高性能
符号数（本地 + 公用）	65535
相同段的行数信息	65535
段数	65535
输入模块	1024
存储区名称的字符长度	31 个字符
存储区的个数	100 注
可由 -b 选项指定的库文件个数	10
可由 -i 选项指定的包含文件路径的个数	64

注 包含默认定义的区域个数。

1.4 RA78K0S 的特性

RA78K0S 具有以下特性：

(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。通过使用该宏功能，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）。

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。但是，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，汇编器会根据分支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，指定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。当不再需要调试语句时，可以不对程序进行大的修改来汇编源程序。

第二章 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。提供介质因主机不同而异。

表 2-1 汇编包的提供介质

主机	操作系统	提供介质
IBM PC/AT 兼容机	日文 Windows (98/Me/2000/XP/NT4.0) 注	CD-ROM
	英文 Windows (98/Me/2000/XP/NT4.0) 注	
SPARCstation 系列	Solaris™ (Rel. 2.5.1 or later)	CD-ROM

注 要在 Windows 中使用汇编器，必须有 PM+。

如果不使用 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

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。

2.2.1 Windows 版本的安装

(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（CD-ROM 驱动器）中设置 RA78K0S 提供介质。安装程序会自动开始。按照显示器屏幕尚显示的信息执行安装。

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

(3) 确认文件

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。

关于各文件夹的详细情况，请参见 "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 /nec tools/bin 路径中。

(1) 登陆

进入主机中。

(2) 目录选择

转到安装目录。

```
%cd /nec tools/bin
```

(3) 设置提供介质

在 CD-ROM 驱动器中设置 CD-ROM。

(4) 复制文件

执行 cp 命令从 CD-ROM 中复制文件（确认在 CD-ROM 驱动器中设置了 CD-ROM 后复制文件）。

(5) 设置环境变量 PATH

将 /nec tools/bin 添加至环境变量 PATH。

2.3 安装设备文件

可通过从联机发送服务 (ODS) 下载来得到设备文件，联机发送服务可由以下网址进行访问。

<http://www.necel.com/micro/ods/eng/tool/DeviceFile/list.html>

2.3.1 Windows 版本的安装

使用“设备文件安装程序”来安装设备文件。“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 [Browse...] 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

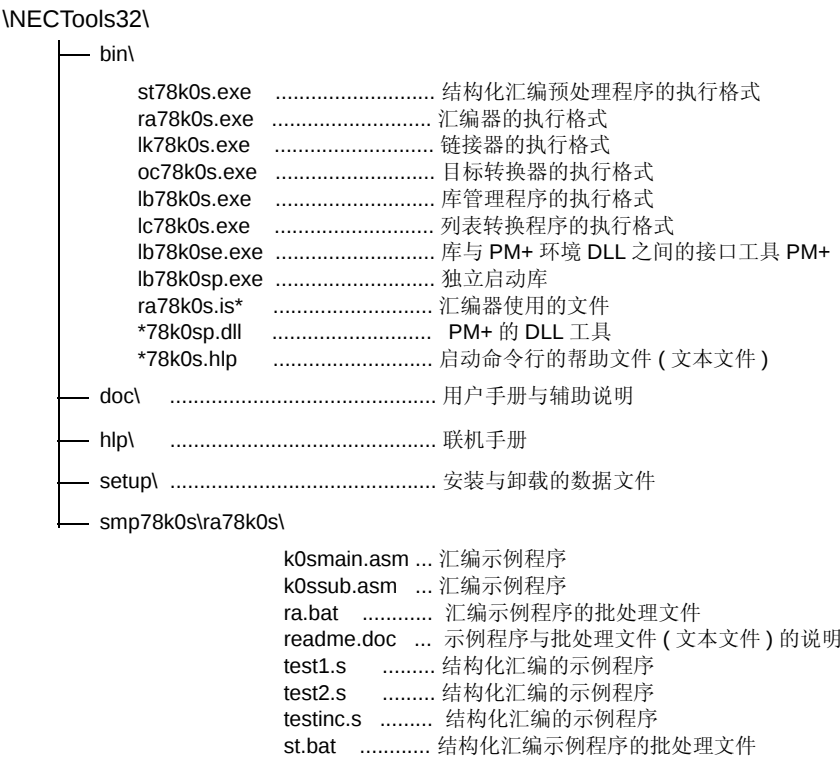
单击 [Move] 按钮，将文件注册到注册表（32 位环境）。

2.4 目录配置

2.4.1 Windows 版本目录配置

安装期间显示的标准目录是 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

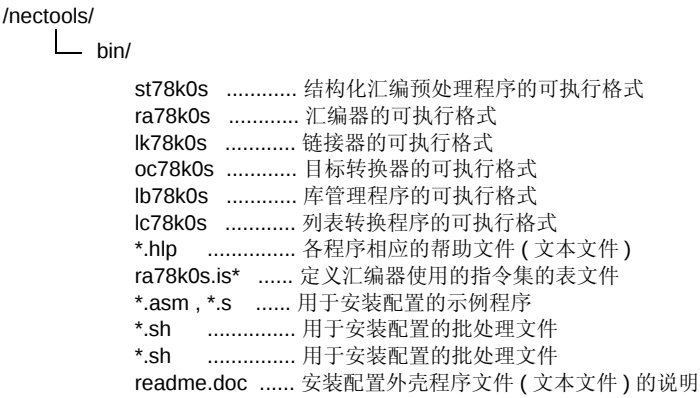


备注 根据安装程序的默认指示，本手册中的说明假设安装至具有默认程序文件夹名“NECTools32”的标准目录中。

2.4.2 UNIX 版本安装目录

安装后的文件配置如下，以下假设安装在 /nec tools/bin 中。

图 2-2 目录配置



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

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 / 删除程序] 窗口

双击 [控制面板] 窗口中的 [添加删除程序] 图标，打开 [添加 / 删除程序] 窗口。

注意事项 在 Windows XP 中，显示 [添加或删除程序] 而不是显示 [添加 / 删除程序]。

(4) 删除 RA78K0S

在选中 [添加 / 删除程序] 窗口中 [安装 / 卸载] 标签中显示的安装软件列表中的 "NEC RA78K0S 78K/0S Assembler Vx.xx" 后，单击 [添加 / 删除 ...] 按钮。

当打开 [系统设置更改] 窗口后，点击 [是] 按钮。

(5) 文件确认

使用 Windows 资源管理等，检查主机中安装的文件是否被卸载。关于各文件夹的详细情况，请参见 "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 主机 (IBM PC/AT 兼容机)

RA 78K0S 在具有 32- 位 CPU 的诸如 i386™ 或更高的机器中运行。

由于汇编包通过使用 DOS 扩展器在 32- 位 CPU 上运行，所以将其设计为在以下操作系统中运行：

- Windows 98/Me 的 DOS 提示符
- Windows 2000/XP/NT 4.0 的命令提示符

运行需要 7 M 字节或更多的保护存储空间。

各操作系统环境变量的限制如下：

2.6.2 环境变量

设置以下环境变量。如果已使用 Windows 安装程序安装了汇编包，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 IBM PC/AT 兼容机有效)

INC78K0S： 指定用于搜索包含文件的目录。

LIB78K0S： 如果使用库，指定用于搜索库的目录。

LANG78K： 指定在注释中描述的 kanji 代码 (2- 字节代码)。

[示例]

- 使用 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
```

- 使用 SPARCstation 系列

< 使用 csh 时的示例 >

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

< 使用 sh 时的示例 >

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

2.6.3 源文件中的 Kanji 码

- Kanji (2- 字节字符) 可用于在源文件中指定位置 (注释等).
- 使用环境变量 (LANG78K) , kanji码控制指令 (KANJI CODE) 或 kanji码定义选项 (-ze/-zs/-zn) 来指定 kanji类型.

第三章 RA78K0S 的执行步骤

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、链接和目标程序转换。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"。

该节当中介绍如何在命令行运行汇编包。

3.1 执行 RA78K0S 之前

3.1.1 示例程序

在系统盘中存储的文件有 "k0smain.asm" 和 "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。

k0smain.asm：主程序模块

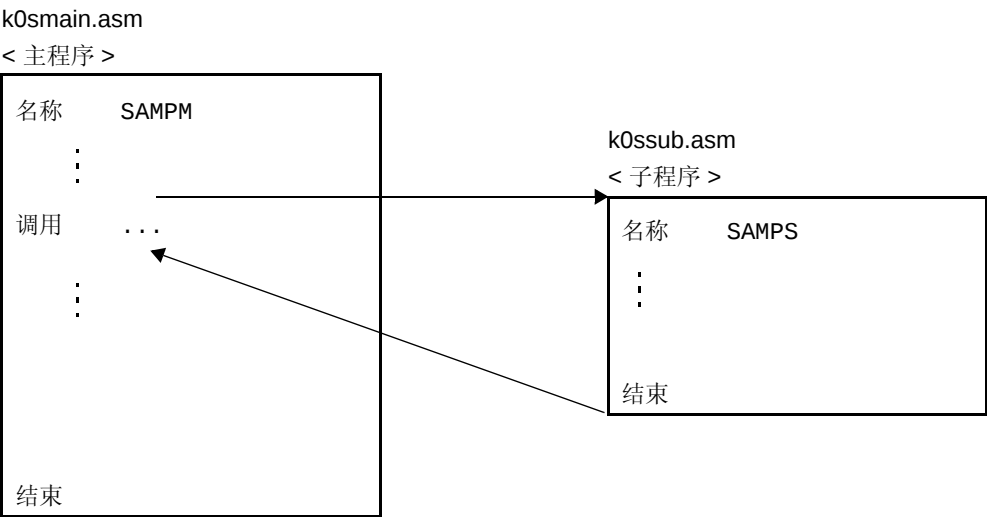
k0ssub.asm：子程序模块

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，它存储于 (k0smain.asm) 中。

子程序模块的名称是 SAMPS，它存储于 (k0ssub.asm) 中。

图 3-1 示例程序的结构



- k0smain.asm (主程序)

```

NAME      SAMPM
; *****
; *
; *      十六进制 -> ASCII 转换程序程序      *
; *
; *      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      ; 在 HL 寄存器中设置十六进制 2- 代码数据

CALL      !CONVAH          ; 转换 ASCII <- HEX
; 输出 BC- 寄存器 <- ASCII 代码
; 设置 DE <- 存储 ASCII 代码表
MOVW      DE , #STASC
MOV       A , B
MOV       [ DE ] , A
INCW      DE
MOV       A , C
MOV       [ DE ] , A

BR        $$

END

```

- k0ssub.asm (子程序)

```

NAME      SAMPS
;
; *****
; *
; *      十六进制 -> ASCII 转换程序
; *
; *
; *      子程序
; *
; *
; *      输入条件      : ( HL ) <- 十六进制 2 代码
; *
; *
; *      输出条件      : BC- 寄存器 <- ASCII 2 代码
; *
; *
; *****
;

PUBLIC    CONVAH

CONVAH :
CSEG
MOV       A , [HL]
ROR       A , 1
ROR       A , 1
ROR       A , 1
ROR       A , 1
AND       A , #0FH      ; 十六进制高代码载入
CALL      !SASC
MOV       B , A          ; 存储结果

XOR       A , A
XCH       A , [HL]
AND       A , #0FH      ; 十六进制低代码载入
CALL      !SASC
MOV       C , A          ; 存储结果

RET

;
; *****
; *
; *      子程序转换 ASCII 代码
; *
; *      输入 Acc ( 低 4 位 )      <- 十六进制代码
; *
; *      输出 Acc                  <- ASCII 代码
; *
; *****
;
; *****
;

SASC :
CMP       A , #0AH      ; 检查十六进制代码 > 9
BC        $SASC1
ADD       A , #07H      ; 偏差 ( +7 )
SASC1 :
ADD       A , #30H      ; 偏差 ( +30 )
RET

END

```

备注 该示例程序是一个参考程序，用于向读者介绍 RA78K0S 的功能以及操作。它不能用作应用程序。

3.1.2 示例程序的结构

以下介绍用作下述操作举例的示例程序。

k0smain.asm：主程序模块

k0ssub.asm：子程序模块

mylib.lib：库管理文件 (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 (ra.bat) 用于 RA78K0S 的操作。

汇编器，链接器，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。如果发生错误，则输出一条消息，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（从联机发送服务 (ODS) 下载获得设备文件）。

以下说明以 uPD789024 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 -lrasample.lmf -rk0smain.rel k0smain.prn
if errorlevel 1 set LEVEL = 1
lc78k0s -lrasample.lmf -rk0ssub.rel k0ssub.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 : ra.bat chiptype

: END

echo on
```

(1) 执行批处理文件。

指定目标设备规范名称，并执行 RA78K0S 操作验证批处理程序。

```
C>ra.bat 9024
```

将以下信息输出到显示器。

```
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 Vx.xx [ xx xxx xxxx ]
  Copyright ( C ) NEC Electronics Corporation xxxx , xxxx

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

清除屏幕。

```
No error.
```

(2) 检查驱动器 C 的内容。

输出以下的文件。

k0smain.rel :	目标模块文件
k0smain.prn :	汇编列表文件
k0ssub.rel :	目标模块文件
k0ssub.prm :	汇编列表文件
rasample.lmf :	载入模块文件
rasample.map :	链接列表文件
rasample.hex :	十六进制格式目标模块文件
rasample.sym :	符号表格文件
k0smain.p :	绝对汇编列表文件
k0ssu.p :	绝对汇编列表文件

(3) RA78K0S 执行步骤概要

以下是对 RA78K0S 执行过程的概略性介绍。

图 3-2 RA78K0S 执行步骤 1

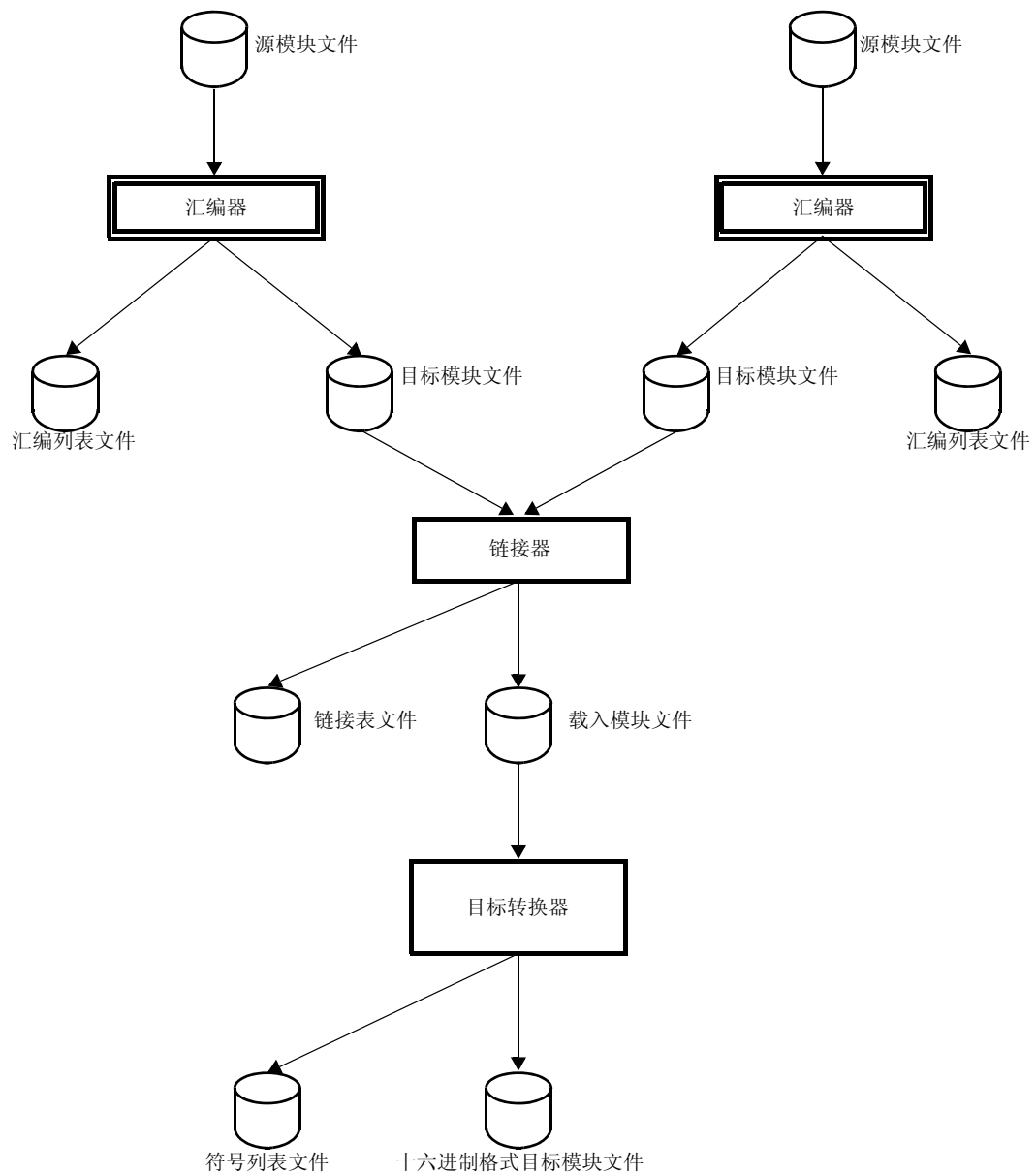
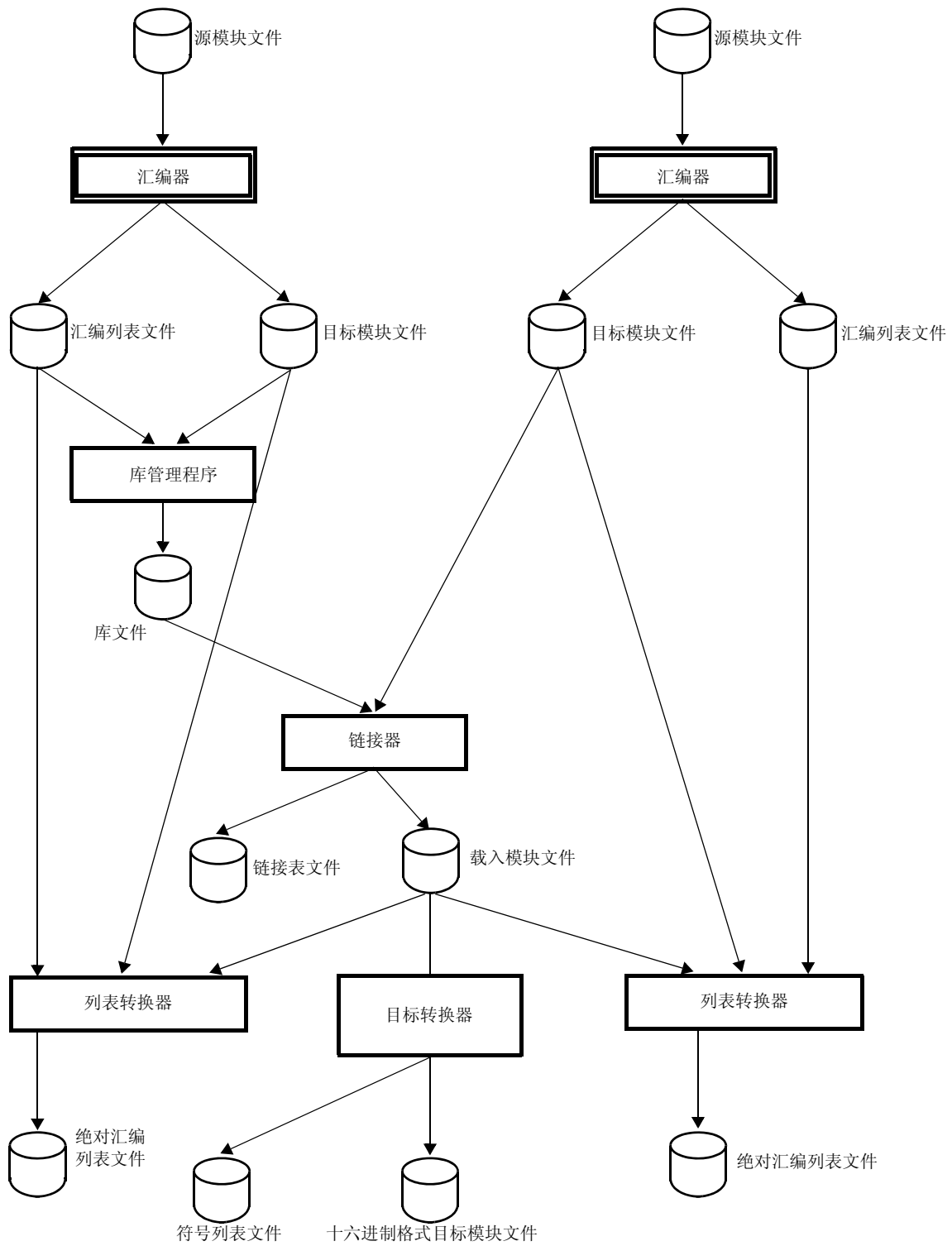


图 3-3 RA78K0S 执行步骤 2



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)。

汇编器，链接器，目标转换器和表转换程序使用 "test1.s" 和 "test2.s" 以上述顺序进行执行，它们是用汇编语言在 st.bat 中编写的源文件。输出任何错误之后都将终止批处理文件。

目标设备的型号规格可被输入到该批处理文件中（设备文件单独出售）。

以下的说明以 uPD789024 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 -ostsample.lmf -pstsmaple.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 -lstsample.lmf -rtest1.rel test1.prn
if errorlevel 1 set LEVEL = 1
lc78k0s -lstsample.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

```

(1) 执行批处理文件。

指定目标设备规范名称，并执行 ST78K0S 操作验证批处理程序。

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

将以下信息输出到显示器。

```
Structured assembler preprocessor for RA78K/0S 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/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.

Structured assembler preprocessor for RA78K/0S 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/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 Vx.xx [ xx xxx xxxx ]  
  Copyright ( C ) NEC Electronics Corporation xxxx , xxxx  
  
Pass1 : start..  
Pass2 : start..  
Conversion complete.
```

清除屏幕。

```
No error.
```

(2) 检查驱动器 C 的内容。

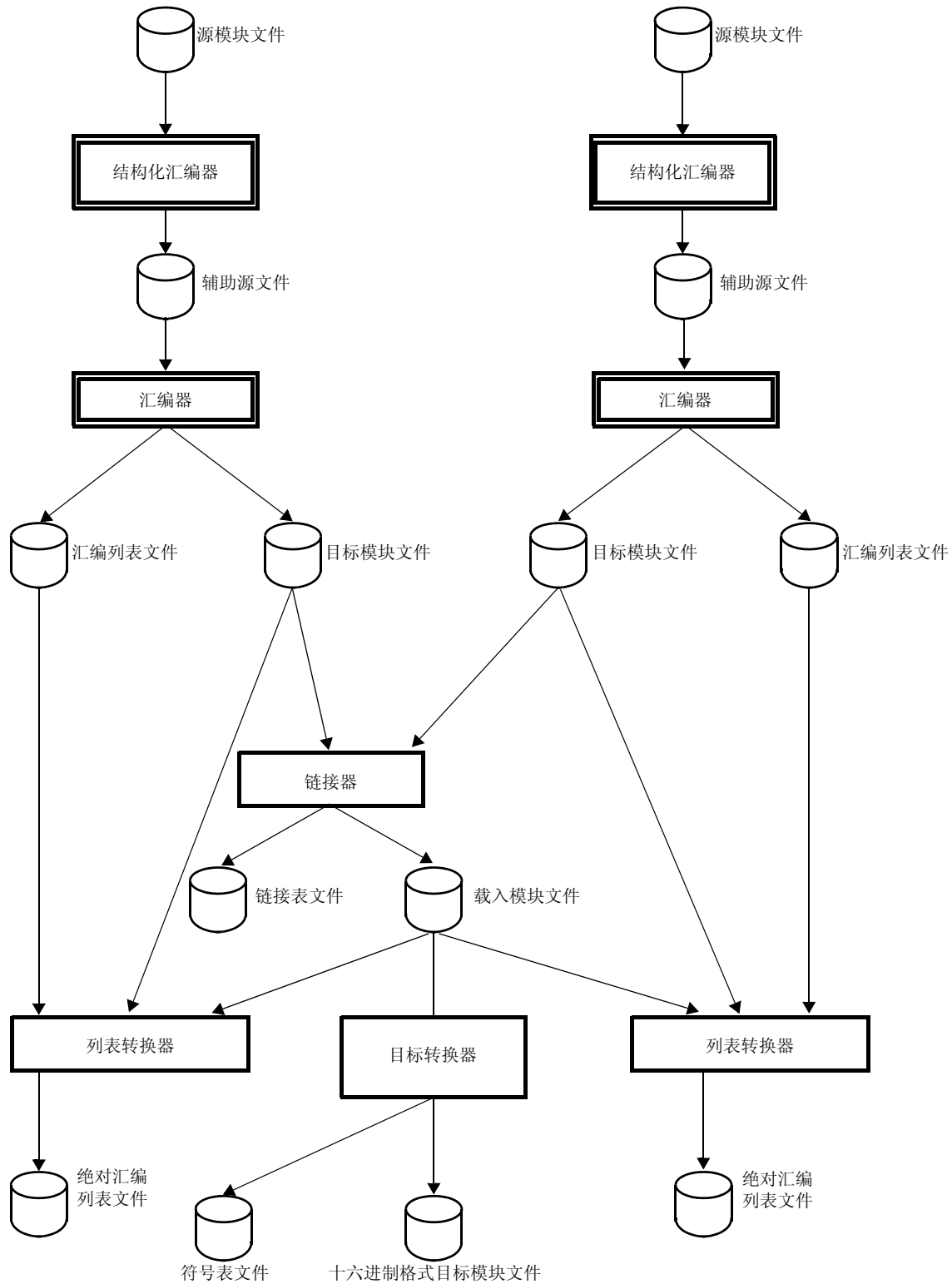
输出以下的文件。

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

(3) ST78K0S 执行步骤概要

以下是 ST78K0S 执行步骤的概要

图 3-4 ST78K0S 执行步骤



3.4 从命令行中的执行步骤

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 k0smain.asm.

在命令行中输入以下内容。

```
C>ra78k0s -c054 k0smain.asm
```

将以下信息输出到显示器。

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

PASS_PARSE Start
PASS_OUTOBJ Start

Target chip : uPD789176
Device file : Vx.xx

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

(2) 检查驱动器 C 的内容。

汇编器输出目标文件 (k0smain.rel) 及汇编表文件 (k0smain.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
```

将以下信息输出到显示器。

```
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) 检查驱动器 C 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) 创建一个指令文件。

指令文件是为链接器指示各段的存储单元的文件。

当您需要扩大默认的 ROM/RAM 存储区，或要定义一个新内存区时，请创建一个指令文件。

当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时，您也需要创建一个指令文件。

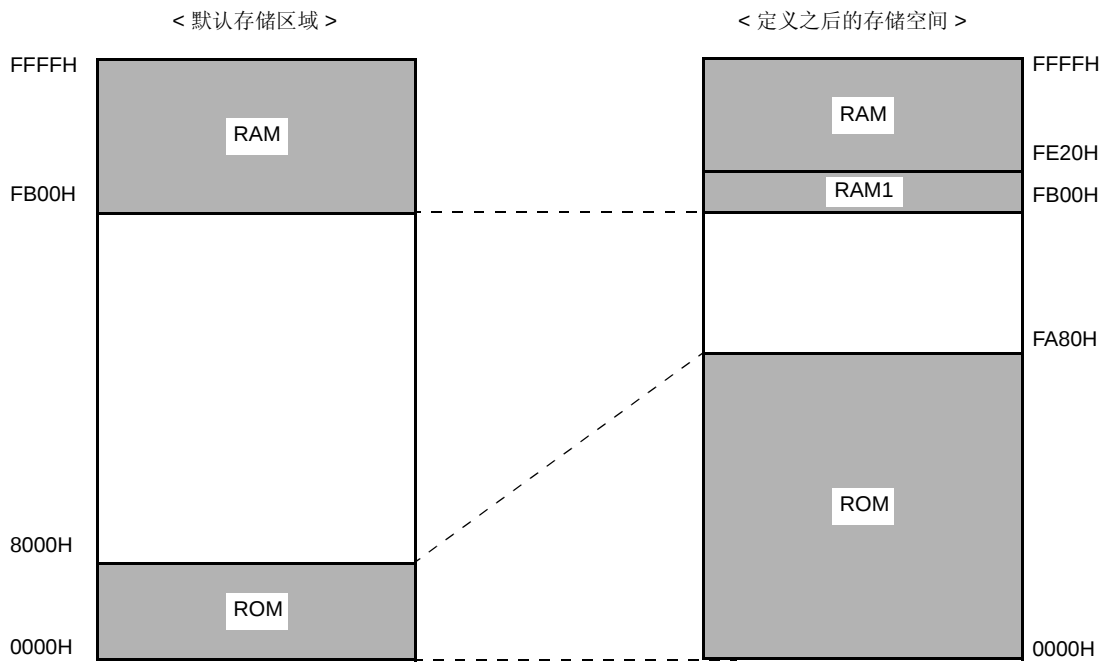
链接时，使用 -d 选项将指示文件输入给链接器。

举例 将 ROM 区从 (0H 至 7FFFH) 扩展至 (0H 至 FA7FH)，并将 RAM 区 (FE20H 至 FFFFH) 和 RAM1 (FB00H 至 FE1FH)。

将以下内容写入指令文件。

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

图 3-5 链接指令



- (6) 汇编的结果是，输出目标模块文件 "k0smain.rel" 和 "k0ssub.rel" 被链接起来。

输入 k0s.dr 作为指令文件。

在命令行中键入以下内容。

```
C>lk78k0s k0smain.rel k0ssub.rel -dk0s.dr注1 -ok0s.lmf -pk0s.map -s注2
```

注 1 如果未指定指令文件则不必要。

注 2 堆栈解析符号 (_@STBEG) 生成选项。

将以下信息输出到显示器。

```
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) 检查驱动器 C 的内容。

链接器输出载入模块文件 (k0s.lmf) 和链接列表文件 (k0s.map)。

如果在链接时指定了选项 -e, 则链接器会输出一个错误表文件。

- (8) 链接的结果是将输出载入模块文件 (k0s.lmf) 转换为十六进制格式文件。

在命令行中键入以下内容。

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

将以下信息输出到显示器。

```
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) 检查驱动器 C 的内容。

目标转换器输出十六进制格式目标模块文件 (k0s.hex) 和符号表文件 (k0s.sym)。

(10) 创建库文件，如下所示。

将汇编器输出的目标模块文件 (k0ssub.rel) 存储为一个库文件。

在命令行中键入以下内容。

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

将以下信息输出到显示器。

```
78K/0S Series Librarian Vx.xx [ xx xxx xx ]
  Copyright ( C ) NEC Electronics Corporation xxxx , xxxx
*create k0s.lib                      ; Contents of "k0s.job"
*add k0s.lib k0ssub.rel              ; Contents of "k0s.job"
*exit
```

(11) 检查驱动器 C 的内容。

库管理程序输出库文件 (k0s.lib)。

(12) 创建一个绝对汇编列表，如下所示。

要创建绝对汇编表 k0smain.asm，需向列表转换程序输入 "k0smain.rel", "k0smain.asm" 和 "k0s.lmf"。

在命令行中键入以下内容。

```
C>lc78k0s k0smain -lk0s.lmf
```

将以下信息输出到显示器。

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

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

(13) 检查驱动器 C 的内容。

表转换程序输出绝对汇编列表文件 (k0smain.p)。

3.5 使用参数文件

如果当汇编器或链接器启动时输入了两个或两个以上的选项,则不能在命令行中完整地指定启动所需信息,或将重复同一个指定信息。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。

汇编器或链接器通过参数文件启动,如下所示:

```
>[ path-name ] ra78k0s Δ -f parameter-file-name  
>[ path-name ] lk78k0s Δ -f parameter-file-name  
>[ path-name ] oc78k0s Δ -f parameter-file-name
```

这里给出其使用的示例。

```
C>ra78k0s -fpara.pra  
C>lk78k0s -fpara.plk  
C>oc78k0s -fpara.poc
```

该参数文件是使用编辑器创建的。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。

这里给出一个使用编辑器创建参数文件的示例。

< para1.pra 的内容 >

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

< para1.plk 的内容 >

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

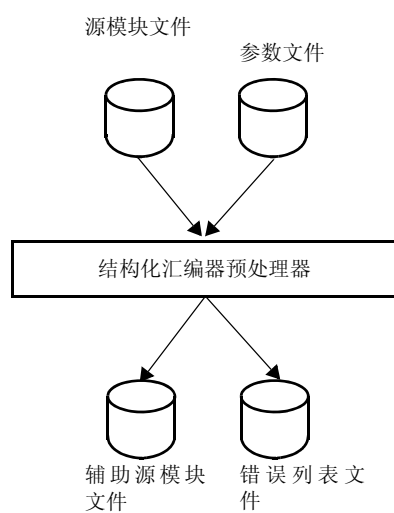
< para1.poc 的内容 >

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

第四章 结构化汇编器预处理器

结构化汇编器预处理器输入用 78K0S 系列微控制器的结构化汇编语言编写的源模块文件，将它们转换成汇编语言，并作为辅助源模块文件输出。

图 4-1 结构化汇编器预处理器的 I/O 文件



4.1 结构化汇编器预处理器的 I/O 文件

结构化汇编器预处理器的 I/O 文件如下表所示。

表 4-1 结构化汇编器预处理器的 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	- 这些是用结构化汇编语言编写的源模块文件。 - 这些文件是由用户创建的。	无
	参数文件	- 这些是包含用于从文件中指定结构化汇编器预处理器选项的文件。 - 这些文件是由用户创建的。	.pst
输出文件	辅助源模块文件	- 这些是用汇编语言编写的源模块文件。	.asm
	错误列表文件	- 这些是包含结构化汇编器预处理器错误数据的文件。	.est

4.2 结构化汇编器预处理器的功能

- (1) 结构化汇编器预处理器读出源模块文件，将它们转换成汇编器输入源文件，并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，结构化汇编器预处理器输出一个异常错误，如果在源模块中发现写错误，则输出一个致命的错误或警告错误。
如果发生异常错误或致命错误，则辅助源文件将不能正常输出。但是，当已指定 `-j` 选项时，即使发生致命错误，也能输出辅助源文件。
- (3) 对应启动时所指定的选项，执行结构化汇编预处理器的处理。关于结构化汇编器预处理器选项的详细说明，请参见“4.4 结构化汇编器选项”。
- (4) 如果结构化汇编预处理器的处理已经正错无误地完成，则结构化汇编器预处理器输出一个“已完成”信息，并将控制权返回给操作系统。

4.3 结构化汇编器预处理器启动

4.3.1 结构化汇编器预处理器启动

可以使用两种方法来启动结构化汇编器预处理器。

(1) 从命令行启动

通过输入以下命令来启动结构化汇编器预处理器。

X> [路径名]	st78k0s	[Δ 选项]	...	源模块文件名	[Δ 选项]...
(a)	(b)	(c)		(e)	(d)

(a) 当前的驱动器名

(b) 当前的目录名

(c) 汇编器的命令文件名

(d) 输入结构化汇编器预处理器操作的详细指令。当指定一个以上的结构化汇编器选项时，用空格隔开各选项。在结构化汇编器选项中不区分大小写字母。关于结构化汇编器选项的详细说明，请参见“4.4 结构化汇编器选项”。

将包含空格的路径放入一对引号（" "）内。

(e) 正进行结构化汇编的源模块文件名。

通过将其包含在一对引号（" "）内，来指定包含空格的路径的文件名。

示例 C>st78k0s -c9024 test1.s -e

(2) 从参数文件中启动

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预处理器选项涉及的不便。

要从参数文件启动结构化汇编器，需要在命令行指定参数文件指定选项（-f）。

从一个参数文件中启动结构化汇编器的程序，如下所示。

```
x> [ 路径名 ] st78k0s [ Δ 源模块文件名 ] Δ -f 参数文件名
|      |              |              |
(a)    (b)            (c)            (d)
```

(a) 当前的驱动器名

(b) 当前的目录名

(c) 参数文件指定选项

(d) 参数文件名

编写参数文件内容的规则如下所示。

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

如果在命令行中省略了源模块文件名，则在参数文件中只能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。

在参数文件中写入所有选项，并输出在命令行中指定的文件名。

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
```

用参数文件（test1.pst）来启动结构化汇编器预处理器。

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

4.3.2 执行启动和结束消息

(1) 执行启动消息

当启动结构化汇编器预处理器时，显示屏上会出现执行启动的消息。

```
Structured assembler preprocessor for RA78K0S Vx.xx [ xx xxx xx ]  
Copyright ( C ) NEC Electronics Corporation xxxx , xxxx
```

(2) 处理显示消息

结构化汇编器预处理器处理的每 100 行显示一次“.”。

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

(3) 执行结束消息

如果未检测到错误，则在显示屏上输出以下消息，并把控制权返回给主机操作系统。

```
Target chip : uPD78xxxx  
Device file : Vx.xx  
  
Conversion complete , 0 error ( s ) found.
```

如果检测到错误，则在显示屏上输出检测到的错误数量，并把控制权返回给主机操作系统。

```
TEST1.S ( 8 ) : RA78K0S error E1209 : Syntax error  
  
Target chip : uPD78xxxx  
Device file : Vx.xx  
  
Conversion complete , 1 error ( s ) found.
```

如果在结构化汇编过程中检测到一个致命错误，该错误使结构化汇编处理不能持续下去，那么结构化汇编器预处理器在显示屏上输出一条消息，取消结构化汇编，并把控制权返回给操作系统。

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 xxxx , xxxx

RA78K0S error F1006 : File not found 'SAMPLE.S'

Program aborted.
```

在上面的例子中，指定一个不存在的源模块文件会产生错误，同时停止汇编。

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 , xxxx

RA78K0S error F1012 : Missing parameter '-z'
Please enter 'ST78K0S --' , if you want help messages.

Program aborted.
```

在上面的例子中，指定一个不存在的选项会产生错误，同时停止汇编。

当显示一条错误消息且停止汇编时，在“第十二章 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

结构化汇编器预处理器选项给出了结构化汇编器预处理器操作的详细说明。

结构化汇编器预处理器选项分为以下 13 种。

表 4-2 结构化汇编器选项

类别	选项	说明
设备类型指定	-c	指定目标设备的类型。
字符号字符指定	-sc	指定字符号名最后的字符。
符号定义指定	-d	指定给 #IFDEF 伪指令等的符号。
制表符数指定	-wt	指定转换指令的输出位置。
包含文件读取路径指定	-i	从在包含文件中指定的路径中读取。
辅助源文件指定	-o	指定辅助源文件名。
错误列表文件指定	-e	指定错误列表文件名。
参数文件指定	-f	指定参数文件名。
调试数据输出指定	-gs	指定结构化汇编器源代码层次调试数据的输出。
	-ngs	
辅助源文件强制输出指定	-j	指定辅助源文件的强制输出。
KanJI 码（2 字节代码）指定	-zs	指定在注释语句中描述的 Kanji 日本汉字码类型。
	-ze	
	-zn	
设备文件搜索路径指定	-y	指定搜索设备文件将通过的路径。
帮助指定	--	在显示屏上输出帮助信息。

4.4.2 结构化汇编器选项的说明

下面详细介绍了结构化汇编器选项。

(1) 设备类型指定

设备类型指定 (-c)

[语法]

-c 设备类型

- 默认假定
不能省略

[功能]

- 指定结构化汇编器预处理器目标的设备。

[说明]

- 总是指定 -c 选项。结构化汇编器预处理器在指定的目标设备上执行处理，并生成汇编器源代码。
请注意，如果省略了 -c 选项，则会发生错误。
- 如果由 -c 选项和由处理器设备类型指定控制指令指定的设备类型不同，则会发出一个警告。在这种情况下，结构化汇编器预处理器将优先使用 -c 选项指定的设备类型。
- 由 -c 选项指定的设备类型作为一个处理器设备类型指定控制指令输出至辅助源文件中。但是，如果制定了一个与处理器设备类型指定控制指令具有相同名称的设备类型，这不会发生。

[使用示例]

- uD789024 指定为目标设备。

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

[注意]

- 不能省略 -c 选项。但是，如果在源文件的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内容，参考各设备中的“设备文件中的使用注意事项”。

(2) 字符符号指定

字符符号指定 (-sc)

[语法]

-sc 字符

- 默认假定
-P 或 p

[功能]

- 当字节 / 字必须区分符号名时，指定作为判断目标的符号的最后字符。

[说明]

- 根据将要处理的数据是一个字节还是一个字，结构化汇编器预处理器将生成不同的指令。
如果是一个代用，对于字节，生成 **MOV** 指令；对于字，生成 **MOVW** 指令。
如果这是一个字符符号保留字，则生成一个字指令。
- 如果指定了一个非保留字的符号，则根据其最后的字符判断该符号是字节符号还是字符符号，并生成一个指令。
- 如果没有指令 -sc 选项，那么以 "P" 或 "p" 结束的符号将判断为一个字符符号。
- 待判断的字符仅仅是字母等效字符。需注意的是，字母不区分大小写。
- 如果执行了一个以上的指定，最后指定的项有效。

[使用示例]

- 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

符号定义说明 (-d)

[语法]

`-d 符号名 [ = 数值 ] [ , 符号名 [ = 数值 ] ... ]`

[功能]

- 定义符号。

[说明]

- 赋予符号的数值必须是二进制、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。
- 使用该选项定义一个符号等同于使用 `#define` 伪指令定义一个符号。
- 使用逗号作为分隔符，最多可在命令行中定义 30 项。
- 最多可用 31 个字符来描述一个符号名。
- 该选项通常与 `#ifdef` 伪指令结合使用。
- 如果执行了一个以上的指定，则最后指定的项有效。
- 如果该选项与 `#define` 伪指令一起指定，则输出一条警告信息，且 `#define` 伪指令将被视为有效。
- 字母不区分大小写。

[使用示例]

- 符号 "TRUE" 定义为 1。

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

(4) 制表符数指定

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
```

- 默认假定

-wt2, 3, 4

[功能]

- 指定转换的汇编语言的制表符数。

[说明]

- -wt 选项允许将汇编器源指令的输出位置自由调整，从而提高程序的可读性。

- 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 指定指令命令输出前的制表符数。

将数值指定为下列范围内的十进制数。

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

- 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]

- 将 "3" 指定给数值 1, "4" 指定给数值 2, "5" 指定给数值 3.

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

(5) 包含文件读取路径指定

包含文件读取路径指定 (-i)

[语法]

`-i 路径名 [ , 路径名 ] ...` （可以指定两个或多个路径名。）

- 默认假定
按下列顺序搜索包含文件。
 - (i) 源文件存在的路径
 - (ii) 由环境变量指定的路径 (INC78K0S)

[功能]

- 指定输入到结构化汇编器中的包含文件的路径名。

[说明]

- 通过用逗号"," 隔开, 一次可以指定两个或多个路径名。
- 在"," 的前后均不能插入空格。
- 按下面的顺序搜索包含文件。
 - (i) 如果在 -i 选项下指定了两个或多个路径名, 则以特定的顺序搜索包含文件。
 - (ii) 如果指定了两个或多个 -i 选项, 则按照后来指定的优先选项搜索包含文件。
 - (iii) 在搜索完由 -i 选项指定的路径后, 则按与默认假定相同的顺序搜索包含文件。
- 如果在 -i 选项后指定了除路径名外的名称, 或者未指定任何路径名, 则发生异常错误。
- 如果指定 65 个或者更多的 -i 选项, 就会发生异常错误。

[使用示例]

- 包含文件的的目录被指定为 `c:\include`。

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

(6) 辅助源文件指定

辅助源文件指定 (-o)

[语法]

`-o [ [ [ 驱动器 : ] 目录 ] 文件名 ]`

- 默认假定
- o 输入文件名.asm

[功能]

- 指定转换后辅助源文件的输出目的地和文件名。

[说明]

- 指定转换后辅助源文件的输出驱动器、目录和文件名。
- 如果省略了 -o 选项，用 "ASM" 代替输入文件的文件类型，则会在当前目录中创建输出文件。
- 可以将 "NUL" 或 "AUX" 指定为文件名。
- 当由于致命错误而停止处理时，不输出辅助源文件。
- 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]

- 将 "sample.asm" 指定为辅助源文件。

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

(7) 错误列表文件指定

错误列表文件指定 (-e)

[语法]

`-e [ [ 驱动器 : ] [ 目录 ] 文件名 ]`

- 默认假定
- e 输入文件名 .est

[功能]

- 指定错误列表文件的输出目的地和文件名。

[说明]

- 指定错误列表文件的输出驱动器、目录和文件名。
- 如果省略了 -e 选项, 用 ".est" 代替输入文件的文件类型, 那么会在当前目录中创建错误列表文件。
- 可以将 "NUL" 或 "AUX" 指定为文件名。
- 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]

- 将 "sample.est" 指定为错误列表文件。

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

(8) 参数文件指定

参数文件指定 (-f)

[语法]

`-f [ [ 驱动器 : ] 目录 ] 文件名`

- 默认假定
无输入文件。

[功能]

- 指定参数文件的文件名。

[说明]

- 指定参数文件的输入驱动器、目录和文件名。
- 文件名不能省略。如果省略了文件类型,则类型将假定为 ".pst".
- 当使用选项 **-d** 在命令行中定义大量符号时,该选项有效。
- 多次指定该选项会导致错误。
- 禁止参数文件嵌套,它们的指定会产生错误。
- 出现换行码 (LF) 或 EOF 前,在参数文件中紧跟 ";" 或 "#" 之后的字符都视为注释。

[使用示例]

- 将 "sample.pst" 指定为参数文件。

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

(9) 调试数据输出指定

调试数据输出指定 (-gs/-ngs)

[语法]

```
-gs  
-ngs
```

- 默认假定

-gs

[功能]

- 指定结构化汇编器源代码层次调试数据的输出。

[说明]

- -gs 选项指定向辅助源文件输出调试数据。
- -ngs 选项使 -gs 选项设置不可用。
- 如果在输入源文件中存在编译器调试数据，则在文件的顶部 -gs 选项用 ";" 代替 "\$"。
- 如果同时指定了 -gs 和 -ngs 选项，则最后指定的选项有效。

[注意]

- 当在结构化汇编器源代码层次调试时，确保指定调试数据输出指定 (-gs/-ngs)。在汇编辅助源文件时，取保指定调试数据输出指定选项 (-g/-ga)。结构化汇编器预处理器将所需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

辅助源文件强制输出指定 (-j)

[语法]

-j

- 默认假定

当由于致命错误而停止处理时，不输出辅助源文件。

[功能]

- 当由于致命错误而使处理停止时，强制输出辅助源文件。

[说明]

- 当由于致命错误而停止处理时，强制输出辅助源文件。
- 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]

- 指定辅助源文件的强制输出。

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

(11) Kanji 代码（2 字节代码）指定

Kanji 码指定 (-zs/-ze/-zn)

[语法]

```
-zs
-ze
-zn
```

- 默认假定

根据操作系统的不同，解释如下。

Windows : -zs

Solaris : -ze

[功能]

- 指定在注释中描述的 Kanji 码类型。

[说明]

- 如下指定 Kanji 码。
 - zs : 解释为移位 JIS 码。
 - ze : 解释为 EUC 码。
 - zn : 不解释为汉字。
- 这些选项对应以下汉字码指定控制指令。
 - zs : \$KANJI CODE SJIS
 - ze : \$KANJI CODE EUC
 - zn : \$KANJI CODE NOTE
- Kanji 码指定的优先级顺序如下。
 - (i) -zs/-ze/-zn 选项的指定
 - (ii) Kanji 码指定控制指令 (\$KANJI CODE) 的指定
 - (iii) 环境变量 LANG78K 的指定
 - (iv) 各操作系统默认设置的指定

[使用示例]

- 指定解释为移位 JIS 码的 Kanji 码。

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

(12) 设备文件搜索路径指定

设备文件搜索路径指定 (-y)

[语法]

`-y [ 驱动器 : ] 目录`

- 默认假定

默认假定：设备文件将按以下顺序进行搜索。

- (i) `<..\dev>` （用于启动 `st78k0s.exe` 的路径）
- (ii) 启动 `st78k0s.exe` 的路径
- (iii) 当前路径
- (iv) 由环境变量 `PATH` 指定的路径

[功能]

- 指定搜索设备文件将通过的路径。

[说明]

- 从指定路径读取设备文件。
- 如果指定了除路径名外的名称，则会发生错误。
- 即使在目录的结尾没有写入目录指定符号，也会假定它在那里。
- 设备文件将按以下顺序进行搜索。
 - (i) 由 `-y` 选项指定的路径
 - (ii) `<..\dev>` （用于启动 `st78k0s.exe` 的路径）
 - (iii) 启动 `st78k0s.exe` 的路径
 - (iv) 当前路径
 - (v) 由环境变量 `PATH` 指定的路径

[使用示例]

- 指定从目录 `c:\NECTools32\dev` 中读取设备文件。

```
C>st78k0s test.s -yc:\NECTools32\dev
```

(13) 帮助指定

帮助指定 (--)

[语法]

--

- 默认假定
无显示

[功能]

- -- 选项在显示屏上显示帮助信息。

[应用]

- 帮助信息是结构化汇编器选项的列表以及它们的定义。当执行结构化汇编器预处理器时参考这些帮助。

[说明]

- 当指定了 -- 选项时，所有其它结构化汇编器选项均无效。

注意事项 不能从 PM+ 中指定该选项。

要参考 PM+ 帮助，在 < 结构化汇编器预处理器选项 > 对话框中单击 [Help] 按钮。

【使用示例】

- 当指定了 -- 选项时, 在显示屏上输出帮助信息.

```

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.
               -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 PM+ 中的选项设置

本部分将介绍如何从 PM+ 中设置结构化汇编器。

4.5.1 设置选项

如果从 PM+ 的 [Tools] 菜单中选择了 [Structured Assembler Options], 或者按下工具栏中的 [ST] 按钮, 就会打开 < Structured Assembler Options > 对话框。

通过在该对话框中键入所需选项, 就能设置结构化汇编器选项。

图 4-2 < Structured Assembler Options > 对话框 (当选中 << Output >> 标签时)

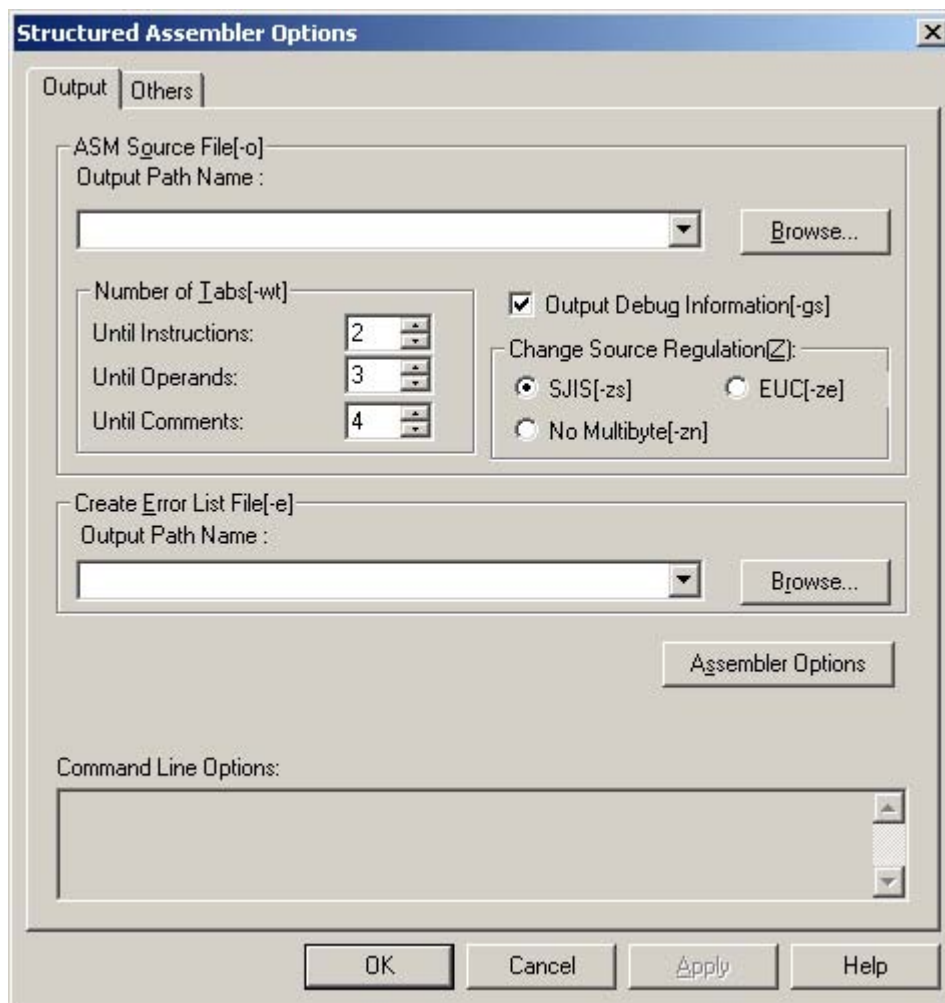


图 4-3 < Assembler Source Options > 对话框（当选中 [Assembler Options] 按钮时）

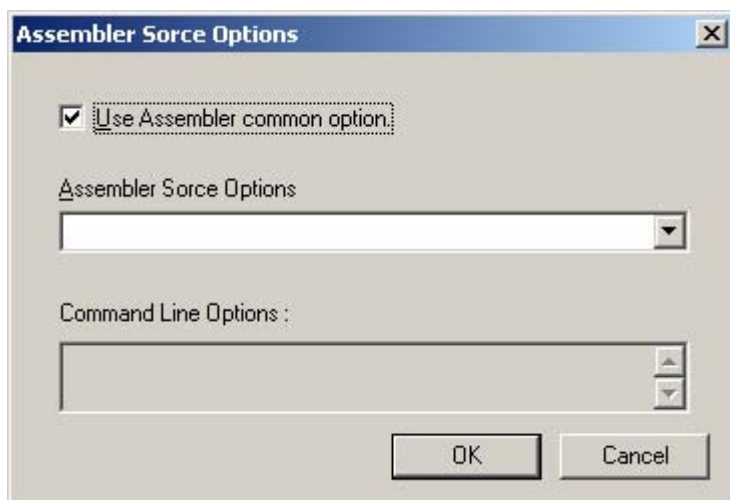
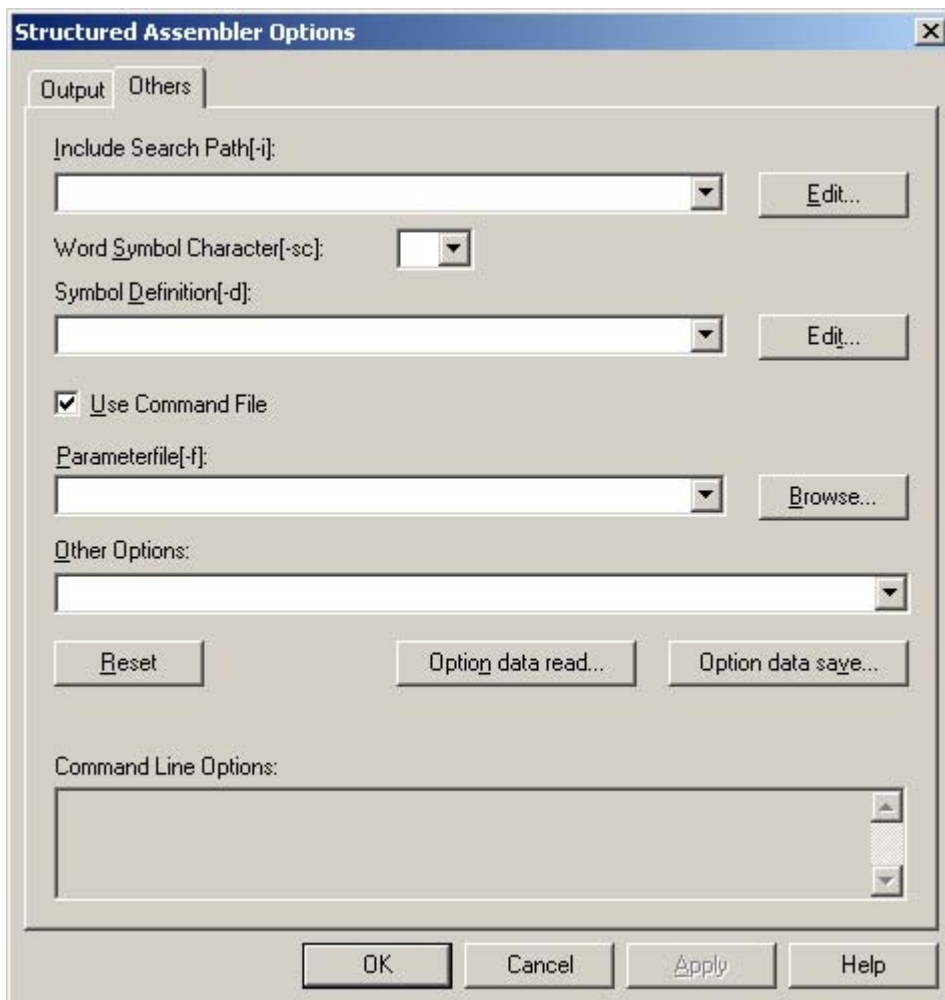


图 4-4 < Structured Assembler Options > 对话框（当选中 << Others >> 标签时）



4.5.2 选项

下面是 < Structured Assembler Options > 对话框中各选项的说明。

(1) << Output >> 标签

- ASM Source File [-o]
（由通用选项指定时）Output Path Name
使用 [Browse] 按钮或直接输入一个路径名来指定 ASM 源文件（辅助源文件）的路径。
（由专用选项指定时）Output File Name
使用 [Browse] 按钮或直接输入路径名和文件名来指定 ASM 源文件（辅助源文件）的路径和名称。
- Number of Tabs [-wt]
指定翻译的汇编语言的制表符数。
指定、操作数和注释前的制表符数可以单独设置。
- Output Debug Information [-gs]
选中该选项，在 ASM 源文件（辅助源文件）中输出调试信息。
- Change Source Regulation [Z]
选择用作源文件注释的 Kanji 码的类型（SJIS[-zs]、UEC[-ze]、或无 Kanji 码 [-zn]）。
- Create Error List File [-e]
（由通用选项指定时）Output Path Name
要输出一个错误列表文件，要使用 [Browse] 按钮或直接输入路径名来指定错误列表文件的路径。
（由专用选项指定时）Output File Name
要输出一个错误列表文件，使用 [Browse] 按钮或直接输入路径名和文件名来指定错误列表文件的路径和名称。
- Assembler Options
指定汇编器源模块文件的汇编器选项。
- Command Line Options
这是只读编辑框。显示当前设置的选项字符串。
- Use Assembler common option
选中该选项，在 < Assembler Options > 对话框中启用通用选项设置。
- Assembler Source Options
输入包含选项名的字符串，启用输出汇编器源文件选项。
- Command Line Options
这是只读编辑框。显示当前设置的选项字符串。

(2) << Others >> 标签

- Include Search Path [-i]
通过按下 [Edit] 按钮或者直接输入路径名，指定读取包含文件通过的路径。
- Word Symbol Character [-sc]

当有必要区分字节和字时，指定符号的最后字符。

- **S**ymbol **D**efinition [-d]

使用 [Edit] 按钮或直接输入一个值，来输入待定义为符号的值。

- **U**se Command File

选中该选项，以创建一个命令文件。

- **P**arameterfile [-f]

使用 [Browse] 按钮或直接输入参数文件名，读取由用户定义参数文件。

- **O**ther Options

要指定能在对话框中设置的选项之外的选项，就要在输入框中输入该选项。

注意事项 不能在 PM+ 上指定帮助指定 (--) 选项。

- **R**eset

清除输入内容。

- **O**ption **d**ata read

打开 < Read Option Data > 对话框，在指定了选项数据文件后，读取该文件。

- **O**ption data **s**ave

打开 < Save Option Data > 对话框，按指定的名称将选项数据保存到选项数据文件中。

- **C**ommand Line Options

这是只读编辑框。显示当前设置的选项字符串。

4.5.3 编辑选项对话框

在 < Edit Option > 对话框的列表格式中编辑各项。

< Edit Option > 对话框描述如下。

图 4-5 < Edit Option > 对话框

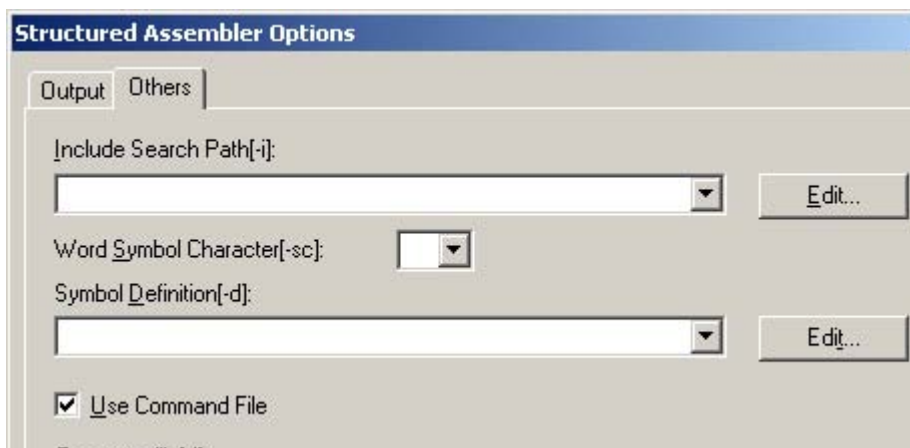
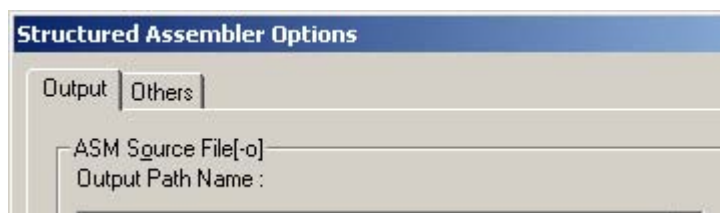


图 4-6 < Add Option > 对话框



- [Add] 按钮

添加一个列表项。

如果待添加的项是一个文件或目录，则打开相应的 < Browse for Folder > 对话框。

在其它情况中，打开 < Add Option > 对话框。指定该框中待添加项的细节。

- [Delete] 按钮

删除选中的列表项。

- [Up] 按钮

向上移动选中的列表项。

- [Down] 按钮

向下移动选中的列表项。

- [Add Sub Directory] 按钮

当指定该项时，在 << Others >> 标签下将子目录添加到选中的列表项作为包含搜索路径 [-i]。

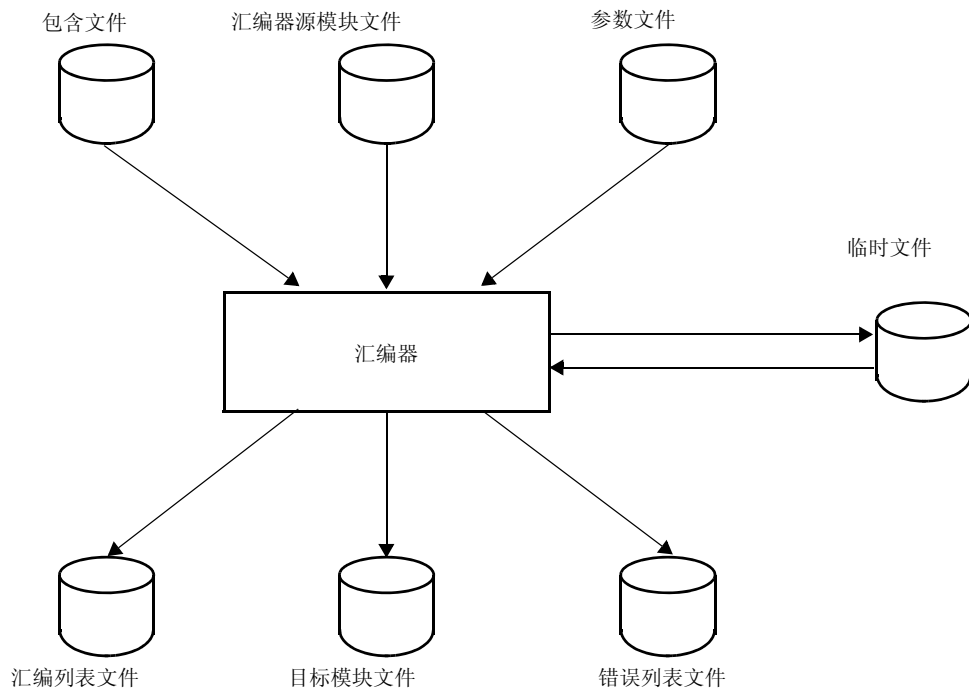
第五章 汇编器

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，并将它们转换成机器语言代码。

汇编器还能输出列表文件，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。

图 5-1 汇编器的 I/O 文件



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	- 这些是用 78K0S 系列微控制器的汇编语言编写的源模块文件。 - 这些文件是由用户创建的。	.asm
	包含文件	- 这些文件用于汇编源模块文件引用。 - 这些是用 78K0S 系列微控制器的汇编语言编写的文件。 - 这些文件是由用户创建的。	-
	参数文件	- 这些文件包含执行过的文件的参数。 - 这些文件是由用户创建的。	.pra
输出文件	目标模块文件	- 这些是二进制文件, 包含有关机器语言数据和机器语言重定位地址的重定位数据和符号数据。	.rel
	汇编列表文件	- 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	- 这些是包含汇编过程中生成的错误数据的文件。	.era
I/O 文件	临时文件	- 这些是汇编器为汇编目的自动创建的文件。当汇编结束时, 删除临时文件。	RAxxxxx.\$n (n = 1-4)

5.2 汇编器的功能

- (1) 汇编器读入源模块文件，并将它们从汇编语言文件转换成机器语言文件。
- (2) 如果发生错误，汇编器将输出一个异常错误。如果在源模块中发现写错误，汇编器将输出一条“致命错误”或“警告错误”消息。
如果输出“异常错误”或“致命错误”消息，则不能正常地输出目标模块文件。但是，当指定了选项 `-j`，即使已发生致命错误，也能输出目标模块文件。
- (3) 根据汇编器启动时指定的汇编器选项，汇编器执行汇编。关于汇编器选项的详细说明，请参见“5.4 汇编器选项”。
- (4) 如果汇编正确无误地完成，则汇编器输出一条“汇编已完成”消息，并把控制权返回给操作系统。

5.3 汇编器启动

可以使用两种方法来启动汇编器。

5.3.1 从命令行启动

X>	[路径名]	ra78k0s	[Δ 选项]	...	源模块文件名	[Δ 选项]	...	[Δ]
	(a)	(b)	(c)		(d)	(e)		(d)

- (a) 当前的驱动器名
- (b) 当前的目录名
- (c) 汇编器的命令文件名
- (d) 输入汇编器操作的详细指令。

当指定两个以上（包括两个）的汇编器选项时，用空格隔开各选项。在汇编器选项中不区分大小写字母。

关于汇编器选项的详细说明，请参见“5.4 汇编器选项”。

将包含空格的路径放入一对引号（" "）内。

- (e) 要进行汇编的源模块的文件名

通过将其包含在一对引号（" "）内，来指定包含空格的路径的文件名。

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

5.3.2 从参数文件中启动

当启动汇编器所需的数据不适合命令行，或者对两个或多个汇编操作重复使用同一汇编器选项时，使用参数文件。

要从参数文件启动汇编器，需要在命令行中指定参数文件选项（-f）。

如下所示，从参数文件启动汇编器。

```
x> ra78k0s [ Δ 源模块文件 ] Δ -f 参数文件名
                |           |
                (b)        (a)
```

(a) 包含启动汇编器所需的数据的文件

(b) 参数文件（指定的选项）

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。

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

如果在命令行中省略了源模块文件，则在参数文件中只能指定 1 个源模块文件名。

源模块文件名还可以书在选项后面。

在参数文件中写入所有的汇编器选项，并输出在命令行中指定的文件名。

有关参数文件的详细说明，请参见“3.5 使用参数文件”。

示例 使用编辑器创建参数文件（k0smain.pra）。

< k0smain.pra 的内容 >

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

使用参数文件（k0smain.pra）启动汇编器。

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

5.3.3 执行启动和结束消息

(1) 执行启动消息

当启动汇编器时，显示屏上会出现执行启动消息。

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

(2) 执行结束消息

如果未检测到由汇编器造成的汇编错误，汇编器在显示屏上输出以下信息，并把控制权返回给操作系统。

```
PASS_PARSE Start  
PASS_OUTOBJ Start  
  
Target chip : uPD78xxx  
Device file : Vx.xx  
  
Assembly complete , 0 error ( s ) and 0 warning ( s ) found.
```

如果检测到由汇编器造成的汇编错误，则汇编器在显示屏上输出错误数，并把控制权返回给操作系统。

```
PASS_PARSE Start  
K0SMAIN.ASM ( 12 ) : RA78K0S error E2201 : Syntax error  
PASS_OUTOBJ Start  
K0SMAIN.ASM ( 29 ) : RA78K0S error E2407 : Undefined symbol reference  
'CONVAH'  
K0SMAIN.ASM ( 29 ) : RA78K0S error E2303 : Illegal expression  
  
Target chip : uPD78xxx  
Device file : Vx.xx  
  
Assembly complete , 3 error ( s ) and 0 warning ( s ) found.
```

如果在汇编过程中汇编器检测到一个致命错误，该错误使汇编处理不能持续下去，那么汇编器在显示屏上输出一条消息，取消汇编，并把控制权返回给操作系统。

示例 1 指定一个不存在的源模块文件。

```
C>ra78k0s sample.asm

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

RA78K0S error F2006 : File not found 'SAMPLE.ASM'
Program aborted.
```

在上面的例子中，指定了一个不存在的源模块文件。 从而发生错误，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

RA78K0S error F2012 : Missing parameter '-z'
Please enter 'RA78K0S--' , if you want help messages.
Program aborted.
```

在上面的例子中，指定了一个不存在的汇编器选项。 从而发生错误，汇编器中止汇编。当显示一条错误消息且中止汇编时，在“第十二章 错误消息”中查找错误原因，并采取相应操作。

5.4 汇编器选项

5.4.1 汇编器选项的类型

汇编器选项是汇编器操作的详细说明。汇编器选项分为 17 种。

表 5-2 汇编器选项

类别	选项	说明
设备类型指定	-c	指定目标设备的设备类型。
目标模块文件输出指定	-o	指定目标模块文件的输出。
	-no	
强制目标模块文件输出指定	-j	指定目标模块文件的强制输出。
	-nj	
调试数据输出指定	-g	输出调试数据（局部符号数据）至一个目标模块文件。
	-ng	
	-ga	输出汇编器源调试数据至一个目标模块文件。
	-nga	
包含文件读取路径指定	-i	从在包含文件中指定的路径中读取。
汇编列表文件输出指定	-p	指定汇编列表文件的输出。
	-np	
汇编列表文件数据指定	-ka	将汇编列表输出到汇编列表文件。
	-nka	
	-ks	将符号列表输出到汇编列表文件。
	-nks	
	-kx	将交叉引用列表输出到汇编列表文件。
	-nkx	
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中 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	在汇编列表文件的末尾插入一个换行代码。
	-nlf	
错误列表文件输出指定	-e	输出一个错误列表文件。
	-ne	
参数文件指定	-f	从指定文件中输入输入文件名和汇编器选项。

表 5-2 汇编器选项

类别	选项	说明
临时文件创建路径的指定	-t	在指定的路径中创建一个临时文件。
Kanji 码指定	-zs	注释中描述的 Kanji 解释为移位 JIS 码。
	-ze	注释中描述的 Kanji 解释为 EUC 码。
	-zn	注释中描述的字符不解释为 Kanji。
设备文件搜索路径指定	-y	从指定路径中读出设备文件。
符号定义指定	-d	定义符号。
系列通用目标指定	-common	指定输出通用于 78K0S 系列的目标模块文件。
自编程指定	-self	当使用自编程时指定。
帮助指定	--	在显示屏上输出帮助信息。

5.4.2 汇编器选项的优先级顺序

下表说明了当同时指定两个汇编器选项时，哪个汇编器选项有效。

表 5-3 汇编器选项的优先级顺序

	-no	-np	-nka	-nks	-kx	-nkx	--
-j	NG						NG
-g	NG						NG
-p			Δ	Δ		Δ	NG
-ka		NG					NG
-ks		NG			NG		NG
-kx		NG					NG
-lw		NG					NG
-ll		NG					NG
-lh		NG					NG
-lt		NG					NG
-lf		NG					NG

[用 NG 标记的项]

当指定水平轴上的选项时，在垂直轴选项中显示的选项不可用。

示例 `C>ra 78k0s -c9024 k0smain.asm -no -lw80 -lf`

选项 `-lw` 和 `-lf` 不可用。

[用 Δ 标记的项]

当水平轴上所有三个选项均被指定时，垂直轴选项上显示的选项不可用。

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

同时指定 `-nka`、`-nks` 和 `-nkx` 选项，所以选项 `-p` 不可用。

当同时指定了一个选项及其对应的 "N"（例如，`-o` 和 `-no`）时，两选项中仅最后指定的选项可用。

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

选项 `-no` 在 `-o` 选项后指定，所以选项 `-o` 不可用，而 `-no` 可用。

在表 5-3 中没有描述的选项对其他选项没有特别的影响。但是，当指定了帮助选项（`--`）时，所有其他选项均变得不可用。

5.4.3 汇编器选项的说明

下面详细介绍了汇编器选项。

(1) 设备类型指定

设备类型指定 (-c)

[语法]

-c 设备类型

- 默认假定
不能省略。

[功能]

- 选项 -c 指定目标设备的设备类型。

[应用]

- 谨慎地使用 -c 选项。汇编器对目标设备执行汇编，并为此设备生成目标代码。

[说明]

- 对于可由 -c 选项指定的目标设备，请参见“使用设备文件时要注意的问题”。

[注意]

- 不能省略 -c 选项。但是，如果在源模块的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。

Δ \$ Δ PROCESSOR Δ (Δ 设备类型 Δ)
 Δ \$ Δ PC Δ (Δ 设备类型 Δ) ; 缩写形式

有关控制指令的信息，参考 RA 78K0S 汇编器封装语言用户手册。

[使用示例]

- 在命令行中指定选项 -c, 如下所示。

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

(2) 目标模块文件输出指定

目标模块文件输出指定 (-o/-no)

[语法]

-o [输出文件名]
-no

- 默认假定

-o (输出文件名).rel

[功能]

- 选项 -o 指定目标模块文件的输出。此外还指定文件输出的位置和文件名。
- 选项 -no 使 -o、-j、-g 和 -ga 选项不可用。

[应用]

- 使用选项 -o 指定目标模块文件输出的位置或更改其文件名。
- 当执行汇编时，指定 -no 选项仅输出汇编列表文件。这样会缩短汇编时间。

[说明]

- 磁盘类型文件名、设备类型文件名 NUL 和 AUX、以及路径名均可在 [output file name] 中指定。当指定了设备类型文件名 CON、PRN 和 CLOCK 时，会发生异常错误。
- 即使指定了 -o 选项，如果发生致命错误时，也无法输出目标模块文件。
- 当指定了 -o 选项，如果省略了驱动器名，目标模块文件仍会输出到当前驱动器。
- 当指定 -o 选项时，如果省略了输出文件名，那么输出文件名将变为 "input-file-name.rel".
- 如果同时指定了 -o 和 -no 选项，那么最后指定的选项优先级较高。

[使用示例]

- 指定目标模块文件（sample.rel）的输出。

C>ra 78k0s -c9024 k0smain.asm -osample.rel

(3) 强制目标模块文件输出指定

强制目标模块文件输出指定 (-j/-nj)

[语法]

<code>-j</code> <code>-nj</code>

- 默认假定

`-nj`

[功能]

- 指定 `-j` 选项, 即使发生致命错误仍能输出目标模块文件。
- 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, 无法输出目标模块文件。当你想要在出现已发生致命错误的通知时执行程序, 要指定选项 `-j` 输出目标模块文件。

[说明]

- 当指定 `-j` 选项时, 即使发生致命错误仍能输出目标模块文件。
- 如果同时指定了 `-j` 和 `-nj` 选项, 那么最后指定的选项优先级较高。
- 如果指定 `-no` 选项, 则选项 `-j` 不可用。

[使用示例]

- 即使发生致命错误, 指定目标模块文件的输出。

```
C>ra 78k0s -c9024 k0smain.asm -j
```

(4) 调试数据输出指定

调试数据输出指定 (-g/-ng, -ga/-nga)

(a) -g/-ng

[语法]

```
-g
-ng
```

- 默认假定

-g

[功能]

- 选项 **-g** 指定将调试数据（局部符号数据）添加到目标模块文件。
- 选项 **-ng** 使选项 **-g** 不可用。

[应用]

- 当执行包含局部符号数据的数据符号调试时，使用 **-g** 选项。
- 在如下 3 种情况下使用 **-ng** 选项。
 - (i) 仅进行全局符号的符号调试
 - (ii) 进行无符号的调试
 - (iii) 仅当需要目标时（使用 **PROM** 赋值，等等）

[说明]

- 如果同时指定了 **-g** 和 **-ng** 选项，那么最后指定的选项优先。
- 不管在什么位置被指定，选项 **-ga** 都优先于其它选项。
- 如果指定 **-no** 选项，则选项 **-g** 不可用。

[注意]

- 在源模块的开始处可以写入与选项 **-g** 和 **-ng** 具有相同功能的控制指令。

```
Δ $ Δ DEBUG
Δ $ Δ DG          ; 缩写形式
Δ $ Δ NODEBUG
Δ $ Δ NODG        ; 缩写形式
```

有关控制指令的信息，参考 RA 78K0S 汇编器封装语言用户手册。

【使用示例】

- 指定添加调试数据到目标模块文件。

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

(b) -ga/-nga

【语法】

```
-ga
-nga
```

- 默认假定

-ga

【功能】

- 选项 **-ga** 指定由结构化的汇编器将源调试数据添加到目标模块文件。
- 选项 **-nga** 使选项 **-ga** 不可用。

【应用】

- 当在汇编器或结构化汇编器的源文件层面上执行调试时使用 **-ga** 选项。要在源文件层面上执行调试，您 will 需要单独可用的集成调试器。
- 在如下 2 种情况下使用 **-nga** 选项。
 - (i) 无汇编源文件的调试
 - (ii) 仅当需要目标时（使用 **PROM** 赋值，等等）
 - (iii) 在 **C** 编译器 / 结构化汇编器源文件层面上的调试

【说明】

- 如果同时指定了 **-ga** 和 **-nga** 选项，那么最后指定的选项优先级较高。
- 不管在什么位置被指定，选项 **-ga** 都优先于其它选项。
- 如果指定了 **-no** 选项，则选项 **-ga** 不可用。

【注意】

- 在源模块的开始处可以写入与选项 **-ga** 和 **-nga** 具有相同功能的控制指令。

```
Δ $ Δ DEBUG
Δ $ Δ NODEBUGA
```

有关控制指令的信息，参考 **RA 78K0S** 汇编器封装语言用户手册。

【使用示例】

- 指定添加汇编器源调试数据到目标模块文件。

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

(5) 包含文件读取路径指定

包含文件读取路径指定 (-i)

[语法]

`-i 路径名 [ , 路径名 ] ...` (可以指定两个或多个路径名。)

- 默认假定
- 按下列顺序搜索包含文件。
 - (i) 源文件存在的路径
 - (ii) 由环境变量指定的路径 (INC78K0S)

[功能]

- 选项 `-i` 指定从指定的路径输入源模块中由 `"$include"` 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]

- 通过用 `","` 隔开,一次可指定两个或多个路径名。
- 在 `","` 前或后均不能键入空格。
- 按下面的次序搜索由 `"$include"` 指定的包含文件。
 - (i) 如果在 `-i` 选项后指定了两个或多个路径名,则按指定的顺序搜索包含文件。
 - (ii) 如果指定了两个或多个 `-i` 选项,则按照后来指定的优先选项搜索包含文件。
 - (iii) 在搜索完由 `-i` 选项指定的路径后,将按与默认假定相同的顺序搜索包含文件。
- 如果在指定了 `-i` 选项后,再指定除路径名以外的东西,或者省略路径名,就会发生异常错误。
- 如果用 `-i` 选项指定 65 个或更多的路径名,则会发生异常错误。

[使用示例]

- 从目录 `c:\sample` 中读取包含文件。

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

(6) 汇编列表文件输出指定

汇编列表文件输出指定 (-p/-np)

[语法]

-p [输出文件名] -np

- 默认假定

-p 输入文件名 .prn

[功能]

- 选项 -p 指定汇编列表文件的输出。还指定输出文件的目的地和文件名。
- 选项 -np 使选项 -p、-ka、-ks、-kx、-lw、-ll、-lh、-lt 和 -lf 不可用。

[应用]

- 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。
- 当执行汇编时, 指定 -np 选项仅输出目标模块文件。这样会缩短汇编时间。

[说明]

- 可以将文件名指定为磁盘类型的文件名或设备类型的文件名。但是, 仅可将 CON、PRN、NUL 和 AUX 指定为设备类型的文件名。如果指定了 CLOCK, 则会发生异常错误。
- 当指定 -p 选项时, 如果省略了输出文件名, 那么汇编列表文件名将会变为 "input-file-name. prn".
- 当指定了 -p 选项, 即使省略了驱动器名, 汇编列表文件仍会输出到当前驱动器。
- 如果同时指定了 -p 和 -np 选项, 那么最后指定的选项优先。

[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

汇编列表文件数据指定 (-ka/-nka, -ks/-nks, -kx/-nkx)

(a) -ka/-nka

【语法】

-ka -nka

- 默认假定

-ka

【功能】

- 选项 -ka 将汇编列表输出到汇编列表文件。
- 选项 -nka 使选项 -ka 不可用。

【应用】

- 指定 -ka 选项以输出汇编列表。

【说明】

- 如果同时指定了 -ka 和 -nka 选项，那么最后指定的选项优先级较高。
- 如果 -nka、-nks 和 -nkx 选项全被指定，则无法输出汇编列表文件。
- 如果指定 -np 选项，则选项 -ka 不可用。

【使用示例】

- 输出汇编列表文件。

C>ra 78k0s -c9024 k0smain.asm -ka

引用 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 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	RF00000		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<- HEX
33	33					; output BC-register <- ASCII code
34	34	000E	RF80100		MOVW	DE , #STASC ; set DE <- store
ASCII 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	

(c) -ks/-nks

[语法]

```
-ks
-nks
```

- 默认假定

-nks

[功能]

- 选项 -ks 可以在符号列表之前将符号列表输出到汇编列表文件。
- 选项 -nks 使选项 -ks 不可用。

[应用]

- 指定选项 -ks 以输出符号列表。

[说明]

- 如果同时指定了 -ks 和 -nks 选项，那么最后指定的选项优先级较高。
- 如果同时指定了 -ks 和 -kx 选项，则忽略 -ks 选项。
- 如果 -nka、-nks 和 -nkx 选项全被指定，则无法输出汇编列表文件。
- 如果指定 -np 选项，则选项 -ks 不可用。

[使用示例]

- 输出符号列表。

```
C>ra 78k0s -c9024 k0smain.asm -ks
```

引用 k0smain.prn.

(在符号列表之前输出汇编列表。)

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

(d) -kx/-nkx

[语法]

```
-kx
-nkx
```

- 默认假定

-nkx

[功能]

- 选项 -kx 可以在交叉引用列表之前将汇编列表输出至汇编列表文件。
- 选项 -nkx 使选项 -kx 不可用。

[应用]

- 当想知道在源模块中在何位置以何种程度引用源模块中定义的各个符号时,或者想知道诸如某个符号被引用在汇编列表中的哪一行之类的信息时,指定 -kx 选项以输出交叉引用列表。

[说明]

- 如果同时指定了 -kx 和 -nkx 选项,那么最后指定的选项优先。
- 如果同时指定了 -ks 和 -kx 选项,则忽略 -ks 选项。
- 如果 -nka、-nks 和 -nkx 选项全被指定,则无法输出汇编列表文件。
- 如果指定 -np 选项,则选项 -kx 不可用。

[注意]

- 在源模块的开始处可以写入与选项 -kx/-nkx 具有相同功能的控制指令。

```
Δ $ Δ XREF
Δ $ Δ XR      ; 缩写形式
Δ $ Δ NOXREF
Δ $ Δ NOXR     ; 缩写形式
```

有关控制指令的信息,参考 RA 78K0S 汇编器封装语言用户手册。

【使用示例】

- 输出交叉引用列表。

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

引用 k0smain.prn.

(在交叉引用列表之前输出汇编列表。)

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

汇编列表文件格式指定 (-lw, -ll, -lh, -lt, -lf/-nlf)

(a) -lw

【语法】

-lw [字符数]

- 默认假定

-lw132 (在显示输出时为 80 个字符)

【功能】

- 选项 -lw 更改可在列表文件中 1 行上打印的字符数。

【应用】

- 指定选项 -lw 以更改可在任何类型列表文件中 1 行上打印的字符数。

【说明】

- 可使用选项 -lw 指定的字符数的范围如下所示。

- (在显示输出时为 80 个字符)

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

如果指定的数值超出了这个范围,或者指定了数值以外的东西,则会发生异常错误。

- 如果省略了字符数,则会指定 132.
- 但是,当将汇编列表文件输出到显示屏时,将指定 80.
- 指定的字符数不包括结束符 (CR,LF)。
- 如果指定 -np 选项,则选项 -lw 不可用。

【注意】

- 在源模块的开始处可以写入与选项 -lw 具有相同功能的控制指令。

△ \$ △ WIDTH

有关控制指令的信息,参考 RA 78K0S 汇编器封装语言用户手册。

【使用示例】

- 指定 80 作为汇编列表文件中每一行的字符数。

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

这里引用汇编列表。

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					
						:

(e) -ll

【语法】

-ll [行数]

- 默认假定

-ll66（显示输出时无分页符）

【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】

- 可使用选项 -ll 指定的行数的范围如下所示。

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

如果指定的数值超出了这个范围，或者指定了数值以外的东西，则会发生异常错误。

- 如果省略了行数，则会指定 66。
- 如果指定的行数为 0，则不会出现任何分页符。
- 如果指定 -np 选项，则选项 -ll 不可用。

【注意】

- 在源模块的开始处可以写入与选项 -ll 具有相同功能的控制指令。

△ \$ △ LENGTH

有关控制指令的信息，参考 RA 78K0S 汇编器封装语言用户手册。

【使用示例】

- 指定 20 作为汇编列表文件中每一页的行数。

C>ra78k0s -c9024 k0smain.asm -ll20

这里引用 k0smain.prn.

```

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

Command: -c9024 k0smain.asm -ll20
Para-file:
In-file: K0SMAIN.ASM
Obj-file: K0SMAIN.REL
Prn-file: K0SMAIN.PRN

      Assemble list
-----

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

ALNO  STNO  ADRS  OBJECT  M I  SOURCE  STATEMENT

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

-----

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

ALNO  STNO  ADRS  OBJECT  M I  SOURCE  STATEMENT

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

```

(f) -lh

【语法】

-lh 字符串

- 默认假定
无

【功能】

- 选项 -lh 指定汇编列表文件首部标题栏中打印的字符串。

【应用】

- 指定 -lh 选项以显示简要说明汇编列表文件内容的标题。
- 通过在每页上打印标题，一眼就能明白汇编列表文件的内容。

【说明】

- 在标题中最多可指定 60 个字符。字符串中不能包含空格。
- 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。
一个双字节字符按 2 个字符计算。
如果每行的最大字符数少于或等于 119，则有效字符串的长度变化如下。
有效长度 = （每行的最大字符数） - 60
- 如果没有指定字符串的长度，则会发生异常错误。
- 如果指定 -np 选项，则选项 -lh 不可用。
- 如果省略 -lh 选项，则汇编列表文件的标题栏将会空白。

- 可写入标题栏的字符设置如下所示。

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。	可以写入。 即使附上“”，仍同命令行所释。
;	如果附上“”，则可写入。	不可写入。 (假定为注释。)
#	可以写入。	不可写入。 (假定为注释。)
" (双引号)	不可作为有效字符写入。	不可作为有效字符写入。
00H	不可写入。	可以写入 但是，它可解释为字符串的末尾。
03H, 06H, 08H, 0DH, 0EH, 10H, 15H, 17H, 18H, 1BH, 7FH	不可写入。	可以写入。 但是，它们会以 '!' 的形式出现在汇编列表文件中。 (单独的 0DH 不会输出到列表中。)
01H, 02H, 04H, 05H, 07H, 0BH, 0CH, 0FH, 11H, 12H, 13H, 14H, 16H, 19H, 1CH, 1DH, 1EH, 1FH	可以写入。 但是，它们将会以 '!' 的形式出现在汇编列表文件中。	可以写入。 但是，它们会以 '!' 的形式出现在汇编列表文件中。
1AH	可以写入。 但是，它会以 '!' 的形式出现在汇编列表文件中。	可以写入。 (文件的结尾)
字母字符	大小写字母按原样输入。	大小写字母按原样输入。
其他	可以写入。	可以写入。

备注 如果启动行上的星号 (*) 不是通配符扩展的目标，那么即使没有用“”括上，仍可将其写入。

【注意】

- 在启动行的开始处可以写入与选项 -lh 具有相同功能的控制指令。

Δ \$ Δ TITLE Δ (Δ ' 字符串 ' Δ)	
Δ \$ Δ TT Δ (Δ ' 字符串 ' Δ)	; 缩写形式

有关控制指令的信息，参考 RA 78K0S 汇编器封装语言用户手册。

[使用示例]

- 在汇编列表文件头打印标题。

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

这里引用 78k0smain.prn.

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

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

(g) `-lt`

[语法]

`-lt [ 字符数 ]`

- 默认假定

`-lt8`

[功能]

- 通过为任何类型列表指定字符数，选项 `-lt` 执行制表处理从而为源模块中的 HT（纵向制表）代码代替和输出大量空格。

[应用]

- 当使用选项 `-lw` 为任何类型列表的每行指定少量字符时，指定选项 `-lt` 以插入一个制表符代替一系列的空格，因此节省使用字符的数量。

[说明]

- 可使用选项 `-lt` 指定的字符数的范围如下所示。
 $0 \leq \text{可指定的字符数} \leq 8$
 如果指定的数值超出了这个范围，或者指定了数值以外的东西，则发生异常错误。
- 如果指定了 `-lt0`，那么将不会执行制表处理，且输出制表代码。
- 如果指定 `-np` 选项，则选项 `-lt` 不可用。

[注意]

- 在源模块的开始处可以写入与选项 `-lt` 具有相同功能的控制指令。

`Δ $ Δ TAB Δ 制表符数`

有关控制指令的信息，参考 RA 78K0S 汇编器封装语言用户手册。

【使用示例】

示例 1 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 个空格 .

C>ra78k0s -c9024 sample.asm -lt1

这里引用 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 代码键入的空白数为 1.

(h) -lf/-nlf

[语法]

```
-lf
-nlf
```

- 默认假定
-nlf

[功能]

- 选项 -lf 在汇编列表文件的末尾插入一个分页符（FF）。
- 选项 -nlf 使选项 -lf 不可用。

[应用]

- 如果想在要打印的汇编列表文件的内容后添加分页符，可指定 -lf 选项以添加分页符。

[说明]

- 如果指定 -np 选项，则选项 -lf 不可用。
- 如果同时指定了 -lf 和 -nlf 选项，那么最后指定的选项优先。

[注意]

- 在源模块的开始处可以写入与选项 -lf/-nlf 具有相同功能的控制指令。

```
△ $ △ FORMFEED
△ $ △ NOFORMFEED
```

有关控制指令的信息，参考 RA 78K0S 汇编器封装语言用户手册。

[使用示例]

- 在汇编列表文件的末尾添加分页符。

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

(9) 错误列表文件输出指定

错误列表文件输出指定 (-e/-ne)

[语法]

```
-e [ 输出文件名 ]  
-ne
```

- 默认假定

-ne

[功能]

- 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。
- 选项 -ne 使选项 -e 不可用。

[应用]

- 指定 -e 选项将错误消息保存到文件中。
- 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]

- 错误列表文件可保存为磁盘类型的文件或设备类型的文件。但是，如果指定了设备类型的文件名 CLOCK, 则会发生异常错误。
- 当指定 -e 选项和省略输出文件名时，错误列表文件名将变为 "input-file-name.era".
- 但指定选项 -e 和省略驱动器名时，错误列表文件将输出至当前目录。
- 如果同时指定了 -e 和 -ne 选项，那么最后指定的选项优先。

【使用示例】

- 创建错误列表文件（k0smain.era）。

C>ra78k0s -c9024 k0smain.asm -ek0smain.era

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

PASS_PARSE Start
K0SMAIN.ASM ( 31 ) : RA78K0S error E2202 : illegal operand
PASS_OUTOBJ Start
K0SMAIN.ASM ( 26 ) : RA78K0S error E2312 : Operand out of range ( byte )
K0SMAIN.ASM ( 31 ) : RA78K0S error E2202 : illegal operand

Target chip : uPD789024
Device file : Vx.xx

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

这里引用错误列表文件（k0smain.era）。

```
PASS_PARSE Start
K0SMAIN.ASM ( 31 ) : RA78K0S error E2202 : illegal operand
PASS_OUTOBJ Start
K0SMAIN.ASM ( 26 ) : RA78K0S error E2312: Operand out of range ( byte )
K0SMAIN.ASM ( 31 ) : RA78K0S error E2202: illegal operand
```

(10) 参数文件指定

参数文件指定 (-f)

[语法]

-f 文件名

- 默认假定
无输入文件。

[功能]

- 选项 -f 从指定的文件输入汇编器选项和输入文件名。

[应用]

- 当启动汇编器所需的数据不适合命令行时，指定选项 -f 选项。
- 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]

- 仅磁盘类型的文件名能指定为 " 文件名 "。如果指定了设备类型的文件名，则会发生异常错误。
- 如果省略文件名，则会发生异常错误。
- 不允许参数文件的嵌套。如果在参数文件内指定了选项 -f，则会发生异常错误。
- 可写入参数文件内的字符数没有限制。
- 使用空格、制表符或者换行符（LF）来分隔选项或文件名。
- 在为命令行上的参数文件指定的位置处扩展参数文件内的参数和输入文件名。
- 最后指定的扩展选项优先。
- 出现换行码（LF）或 EOF 前，在参数文件中紧跟 ";" 或 "#" 之后的字符都视为注释。
- 如果指定了两次或多次 -f 选项，则会发生异常错误。

【使用示例】

- 用参数文件执行汇编。

设置参数文件（k0smain.pra）的内容如下。

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

从命令行中键入以下内容。

```
C>ra 78k0s -fk0smain.pra
```

(11) 临时文件创建路径的指定

临时文件创建路径的指定 (-t)

[语法]

-t 路径名

- 默认假定

在由环境变量 **TMP** 所指定的路径中创建一个临时文件。

当没有指定路径时，则在当前路径中创建临时文件。

[功能]

- 选项 **-t** 指定创建临时文件的路径。

[应用]

- 使用 **-t** 选项指定创建临时文件的位置。

[说明]

- 仅能将一个路径指定为路径名。
- 不可省略路径名。
- 即使存在之前创建的临时文件，如果文件不受保护，则将其覆盖重写。
- 一旦所需的存储器空间可用，就会在存储器中扩展临时文件。如果没有足够的存储空间，则临时文件的内容将被写入磁盘。
- 稍后可通过所保存的磁盘文件访问这些临时文件。
- 汇编结束时删除临时文件。通过按 (**CTRL-C**) 键而使汇编终止时，也需删除临时文件。
- 根据以下顺序确定要创建的临时文件所在的路径。

(i) 由选项 **-t** 指定的路径

(ii) 由环境变量 **TMP** 指定的路径（省略 **-t** 选项时）

(iii) 当前路径（未设置 **TMP** 时）

当指定了 (i) 或 (ii) 时，如果无法在指定的路径中创建临时文件，则会发生终止错误。

[使用示例]

- 指定临时文件输出到目录 **c:\tmp**。

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

(12) Kanji 码指定

Kanji 码指定 (-zs/-ze/-zn)

[语法]

```
-zs
-ze
-zn
```

- 默认假定
- 根据操作系统的不同，解释如下。

-zs (Windows)

-ze (Solaris)

[功能]

- 注释中描述的 Kanji 解释为指定的汉字码。
- 根据选项，Kanji 码解释如下。

-zs : 移位 JIS 码

-ze : EUC code

-zn : 不能解释为汉字

[应用]

- 这些选项用于指定注释行汉字中的 Kanji 码的解释。

[说明]

- 如果同时指定 -zs、-ze 和 -zn 选项，则最后指定的选项优先。
 - 在源模块的开始处可以描述与 -zs、-ze 和 -zn 选项具有相同功能的控制指令。
- 句法如下所示。

```
Δ $ Δ KANJICODE Δ SJIS
Δ $ Δ KANJICODE Δ EUC
Δ $ Δ KANJICODE Δ NONE
```

有关控制指令的细节，参考 RA 78K0S 汇编器封装语言用户手册。

- 使用环境变量 LANF78K 也可以指定 Kanji 码。有关环境变量的细节，参考 "11.2 准备开发环境（环境变量）"。

[使用示例]

- Kanji 码解释为 EUC 码。

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

(13) 设备文件搜索路径指定

设备文件搜索路径指定 (-y)

[语法]

-y 路径名

- 默认假定将按以下顺序从确定的路径中读取设备文件。
 - (i) 1) <..\dev> (用于启动 ra78k0s.exe 的路径)
 - (ii) 2) 启动 RA78K0S 的路径
 - (iii) 3) 当前目录
 - (iv) 4) 环境变量 PATH

[功能]

- 从指定路径中读出设备文件。

[应用]

- 指定设备文件存在的路径。

[说明]

- 如果在指定 -y 选项后再指定除路径名以外的东西, 则会发生异常错误。
- 如果在选项 -y 后省略路径名, 就会发生异常错误。
- 按如下确定的顺序的路径读取设备文件。
 - (i) 由选项 -y 指定的路径
 - (ii) <..\dev> (用于启动 ra 78k0s.exe 的路径)
 - (iii) 启动 RA78K0S 的路径
 - (iv) 当前目录
 - (v) 环境变量 PATH

[使用示例]

- 将设备文件的路径指定为目录 c:\78k0s\dev

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

(14) 符号定义指定

[语法]

`-d 符号名 [= 数值] [, 符号名 [= 数值] ...]`

- 默认假定

None

[功能]

- 选项 **-d** 定义符号。

[应用]

- 定义符号时，指定选项 **-d**。

[说明]

- 赋予符号的数值必须是二进制、八进制、十进制或十六进制的。当省略数值指定时，将指定 **1**。
- 通过使用逗号作为分隔符，最多可指定 **30** 个符号。
- 最多可用 **31** 个字符来描述一个符号名。
- 当指定了相同的名字时，最后指定的符号名有效。
- 符号名区分大小写。
- 由 **-d** 定义的符号用来替代 **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** 为符号定义时：

`C>ra 78k0s k0smain.asm -c9024 -dSYM = 2`

(15) 系列通用目标指定

系列通用目标指定 (-common)

[语法]

`-common`

- 默认假定
- 输出可支持的指定设备的目标文件。

[功能]

- -common 选项指定通用于 78K0S 系列的目标模块文件的输出。

[应用]

- 该选项生成通用于 78K0S 系列的目标代码，与设备类型指定选项（-c）无关。
- 输出目标模块文件可以用在 78K0S 系列中指定的其他设备目标文件链接。

[说明]

- 指定该选项可以生成通用于 78K0S 系列的目标代码。

[注意]

- 即使指定了 -common 选项，也不能省略设备类型指定选项（-c）或者拥有相同功能的控制指令。
- 如果为所有即将链接起来的输入目标模块文件指定了系列通用目标指定选项（-common），则会发生错误。

[使用示例]

```
C>ra 78k0s k0ssub.c -c9024 -common
```

(16) 自编程指定

自编程指定 (-self)

[语法]

-self

- 默认假定

无

[功能]

- 即使地址 8100H 访问越界（如，不存在内部 ROM），当描述 "CALL!8100H" 时，-self 选项阻止错误发生。

[应用]

- 使用自编程时指定 -self 选项。

[说明]

- 当在自编程期间描述 -self 时，如果发生错误，则要指定该选项。

(17) 帮助指定

帮助指定 (--)

[语法]

--

- 默认假定

无显示

[功能]

- 选项 -- 显示帮助消息。

[应用]

- 帮助消息就是汇编选项的说明列表。当执行汇编器时，参阅帮助信息。

[说明]

- 当指定 -- 选项时，所有其他的选项均不可用。
- 要读取帮助消息的下一部分，请按返回键。
- 要退出帮助显示时，请按下除返回键之外的任何键，然后再按返回键。

注意事项 不能在 **PM+** 中指定该选项。

要参考 **PM+** 帮助，在 < 汇编器选项 > 对话框中单击 [帮助] 按钮。

[使用示例]

- 当指定了 -- 选项时, 在显示屏上输出帮助消息.

C>ra78k0s --

```

78K/0S Series Assembler Vx.xx [ xx xxx 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 ].
-common         : Create the common object module file for 78K0S Series.
-self          : Use Self-programming.
-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 PM+ 中的选项设置

本节介绍了从 PM+ 中设置汇编器选项的方法。

5.5.1 选项设置方法

如果从 PM+ 的 [Tools] 菜单中选择了 [Assembler Options], 或者按下工具栏中的 [RA] 按钮, 就会打开 < Assembler Options > 对话框。

通过在该对话框中输入所需的选项, 就能设置汇编器选项。

图 5-2 < Assembler Options > 对话框 (当选中 << Output1 >> 标签时)

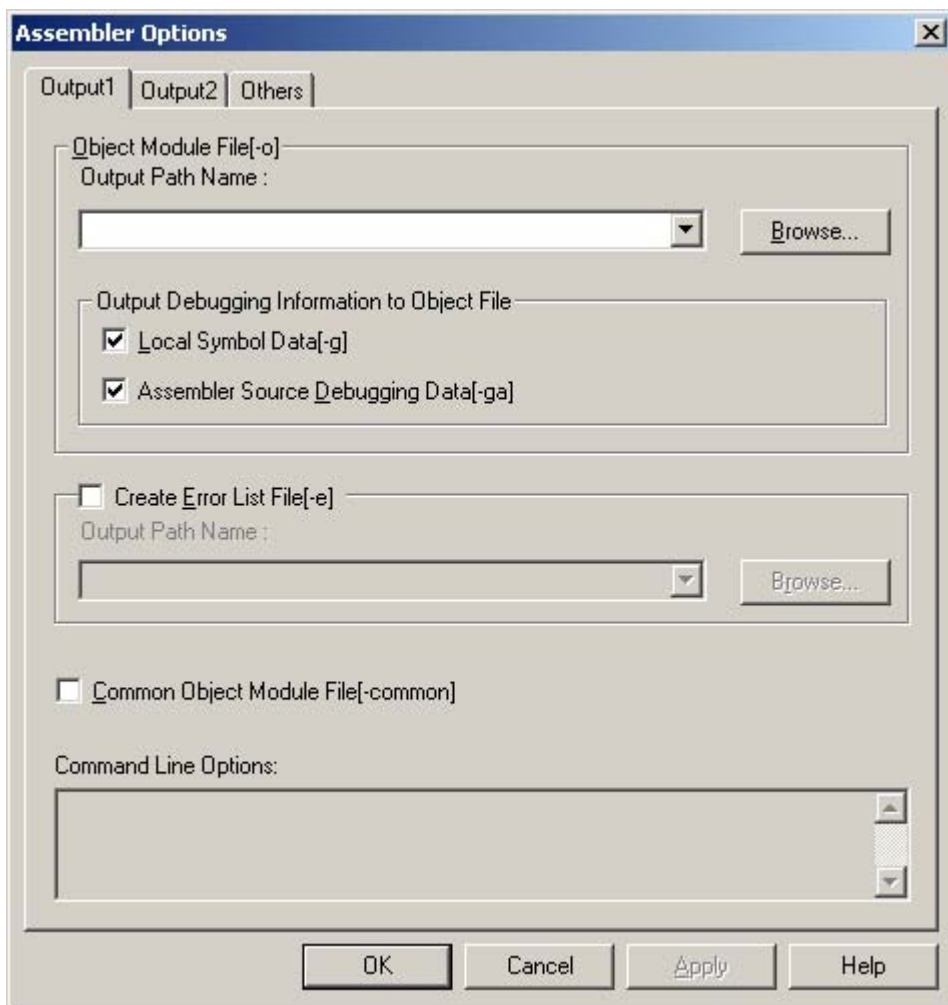


图 5-3 < Assembler Options > 对话框（当选中 << Output2 >> 标签时）

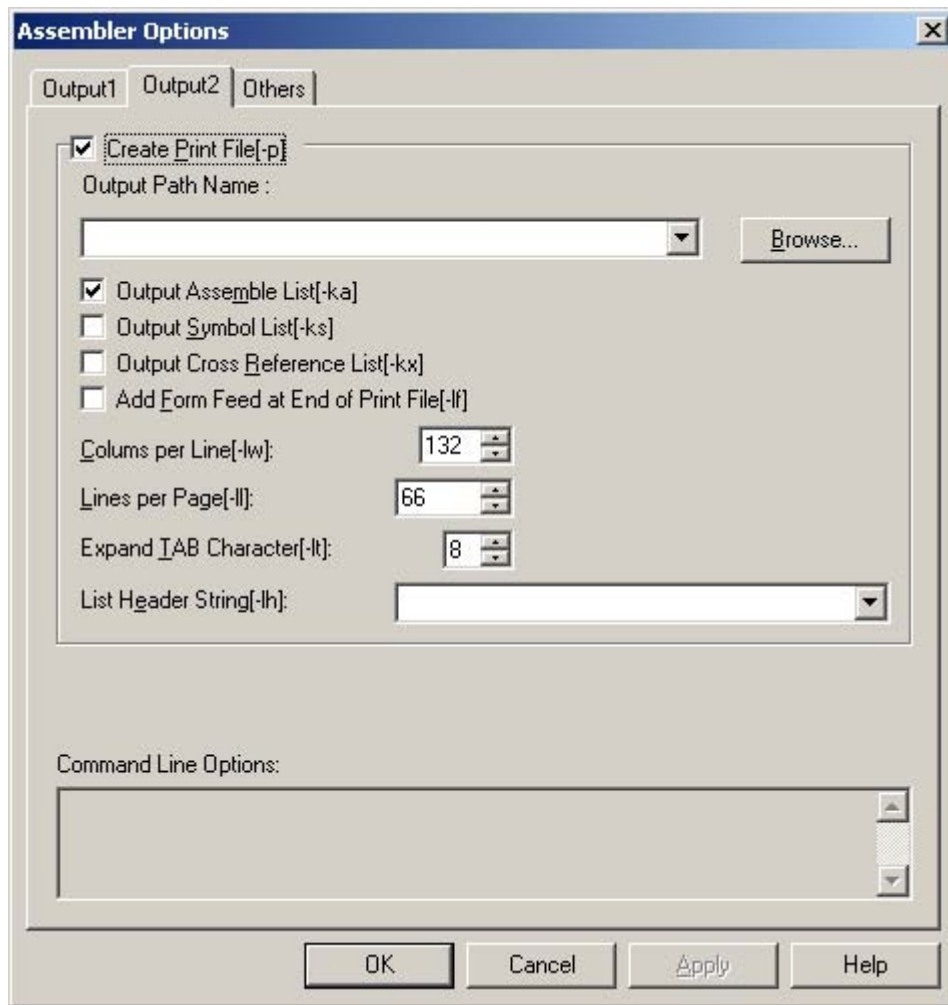
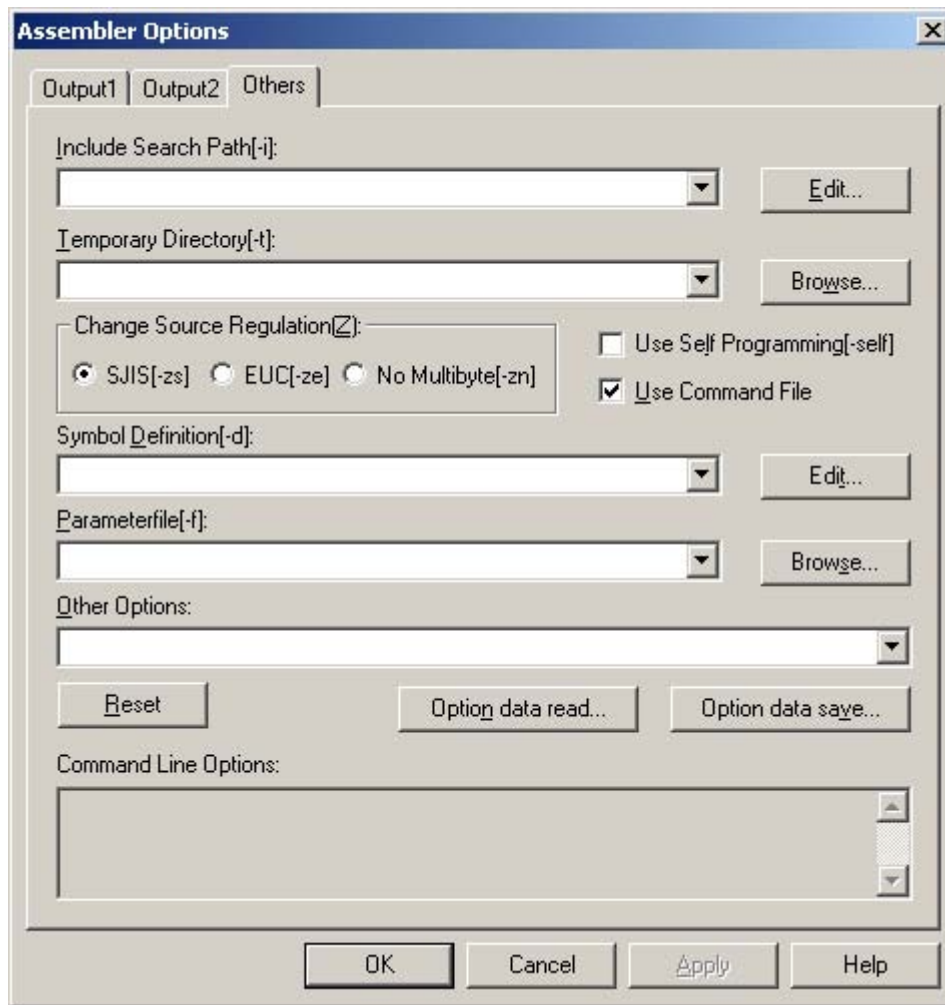


图 5-4 < Assembler Options > 对话框（当选中 << Others >> 标签时）



5.5.2 选项设置

在 < Assembler Options > 对话框中的各个选项描述如下。

(1) << Output1 >> 标签

- Object Module File [-o]
（由通用选项指定时）Output Path Name
使用 [Browse] 按钮或直接输入路径名来指定目标模块文件的路径。
（由专用选项指定时）Output File Name
使用 [Browse] 按钮或直接输入路径名和文件名来指定目标模块文件的路径和文件名。
- Local Symbol Data [-g]
选中该选项，将把调试信息（局部符号信息）添加到目标模块文件中。
- Asembler Source Debugging Data [-ga]
选中该选项，将把源调试信息添加到目标模块文件中。
- Create Error List File [-e]
选中该选项，输出一个错误列表文件。
（由通用选项指定时）Output Path Name
使用 [Browse] 按钮或直接输入路径名来指定错误列表文件的路径。
（由专用选项指定时）Output File Name
使用 [Browse] 按钮或直接输入路径名和文件名来指定错误列表文件的路径和文件名。
- Command Object Module File [-common]
选中该选项，输出通用于 78K0S 系列的目标模块文件。
- Command Line Options
这是只读编辑框。显示当前设置的选项字符串。

(2) << Output2 >> 标签**- Create Print File [-p]**

选中该选项，输出一个汇编列表文件。

（由通用选项指定时） **Output File Name**

使用 **[Browse]** 按钮或直接输入路径名来指定汇编列表文件的路径。

（由专用选项指定时） **Output File Name**

使用 **[Browse]** 按钮或直接输入路径名和文件名来指定汇编列表文件的路径和文件名。

- Output Assemble List [-ka]

选中该选项，在汇编列表文件中输出汇编列表。

- Output Symbol List [-ks]

选中该选项，在汇编列表后，在汇编列表文件中输出符号列表文件。

- Output Cross Reference List [-kx]

选中该选项，在汇编列表后，在汇编列表文件中输出交叉引用列表。

- Add Form Feed at End of Print File [-lf]

选中该选项，在汇编列表文件的内容后后缀换页符（FF）。

- Columns per Line [-lw]

指定汇编列表文件中每行的字符数，范围是 72 至 2046。

- Lines per page [-ll]

指定汇编列表文件中每页的行数，范围是 20 至 32767。

- Expand TAB character [-lt]

指定制表符的长度，范围是 0 至 8。

- List Header String [-lh]

指定汇编列表文件首部标题栏中要打印的字符串。最多输入的字符数为 60。

- Command Line Options

这是只读编辑框。显示当前设置的选项字符串。

(3) << Others >> 标签**- Include Search Path [-i]**

使用 [Edit] 按钮或者直接输入路径名，指定读取包含文件通过的路径。

- Temporary Directory [-t]

使用 [Browse] 按钮或直接输入路径名来指定要创建的临时文件的路径。

- Change Source Regulation [Z]

该选项选择要用作源文件注释的 Kanji 码的类型（SJIS[-zs]、EUC[-ze]、或 No Multibyte [-zn]）。

- Use Self Programming[-self]

选中该选项，使用自编程功能。

- Use Command File

选中该选项，以创建一个命令文件。

- Symbol Definition [-d]

使用 [Edit] 按钮或直接输入一个值，来输入待定义为符号的值。

- Parameterfile [-f]

使用 [Browse] 按钮或直接输入文件名，来指定要作为由用户定义的参数文件输入的文件。

- Other Options

要指定能在对话框中设置的选项之外的选项，就要在输入框中输入该选项。

注意事项 不能在 PM+ 上指定帮助指定（--）选项。

- Reset

清除输入内容。

- Option data read

打开 < Read Option Data > 对话框，在指定了选项数据文件后，读取该文件。

- Option data save

打开 < Save Option Data > 对话框，用名称将选项数据保存到选项数据文件中。

- Command Line Options

这是只读编辑框。显示当前设置的选项字符串。

5.5.3 编辑选项对话框

在 < Edit Option > 对话框的列表格式中编辑各项。

< Edit Option > 对话框描述如下。

图 5-5 < Edit Option > 对话框

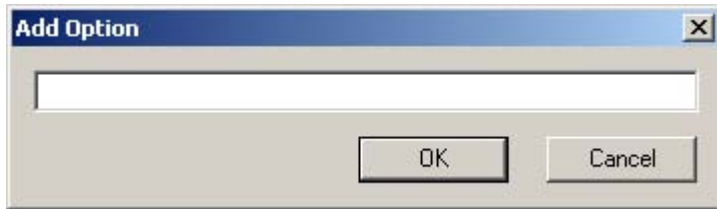
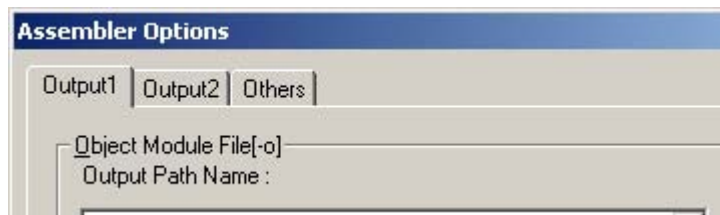


图 5-6 < Add Option > 对话框



- [Add] 按钮

添加一个列表项。

如果待添加的项是一个文件或目录，则打开相应的 < Browse for Folder > 对话框。

在其它情况中，打开 < Add Option > 对话框。指定该框中待添加项的细节。

- [Delete] 按钮

删除选中的列表项。

- [Up] 按钮

向上移动选中的列表项。

- [Down] 按钮

向下移动选中的列表项。

- [Add Sub Directory] 按钮

当指定该项时，在 << Others >> 标签下将子目录添加到选中的列表项作为包含搜索路径 [-i](I)。

第六章 链接器

链接器输入许多由 78K0S 汇编器输出的目标模块文件，确定位置地址并将这些文件以单个载入模块文件的形式输出。

链接器还能输出列表文件，如链接列表文件、错误列表文件。

如果发生链接错误，则会将错误信息输出到错误列表文件，以说明错误原因。当发生错误时，将不会输出载入模块文件。

6.1 链接器的 I/O 文件

链接器的 I/O 文件如下所示：

表 6-1 链接器的 I/O 文件

类型	文件名	说明	默认文件类型
输入文件	目标模块文件	- 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	- 这些文件包含两个 (或多个) 目标模块文件的文件。 - 这些文件由库管理程序输出。	.lib
	指令文件	- 这些文件包含链接中用到的链接命令。 - 这些文件由用户创建。	.dr
	参数文件	- 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	- 这些二进制映像文件包含了在链接中创建的所有数据。这些文件输入到目标转换器中。	.lmf
	链接列表文件	- 这些是显示链接结果的列表文件。	.map
	错误列表文件	- 这些文件包含在链接过程中生成的出错数据。	.elk
I/O 文件	临时文件	- 这些文件由链接器自动生成，在链接中使用。汇编完成时删除。	LKxxxxx.\$n (n = 1-3)

6.2 链接器的功能

(1) 链接输入程序段

链接器决定和控制着每个程序段的定位地址。

链接器识别相同的程序段，将其链接成单个程序段，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

链接器确定输入模块各程序段的存储地址。如果在源文件中指定了程序段的地址属性，那么根据它们的属性存放程序段。链接器还能在其链接指令文件中指定地址属性。

(4) 校正目标代码

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

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。

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。

注意事项 因为只有一个存储空间可用，所以不能指定除 /REGULAR 外的其它空间。

表 6-2 程序段分配组（外部 ROM 等等）

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 则直到 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的各程序段的位置。

备注 2 具体例子参见 "3.4 (5) 创建指令文件"。

6.4 链接指令

链接指令(以下称为“指令”)是一组指令,用来在链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。

举例 CALLT 区
 内部 ROM
 外部 ROM
 SADDR
 除 SADDR 外的内部 RAM

- (3) 由链接器指定程序段的位置

指定各程序段的下列项。

- 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)。在链接器启动时,指定 **-d** 选项来读入创建的文件。

链接器从该文件读取指令并翻译来进行链接。

如下所示,可以使用两类指令。

表 6-3 指令类型

指令类型	说明
存储指令	- 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
程序段定位指令	- 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。

- 存储指令
MEMORY 存储区名 : (起始地址值 , 长度) / 存储空间名]
- 程序段定位指令
MERGE 程序段名 : [AT Δ (Δ 起始地址 Δ)]
[= 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。

举例 MEMORY

memory

Memory: 禁止使用

当源文件中有两个 (或更多) 程序段同名时,指定 "COMPLETE" 以避免合并程序段并产生错误。若要合并程序段,在指令中指定 "SEQUENT (默认)"。

SEQUENT: 按程序段出现的次序进行合并,没有间隔。BSEG 按出现的顺序以位为单位进行合并。

COMPLETE: 当两个 (或更多) 程序段同名时,出现错误。

举例 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区分大小写字符。

(3) 数值

若要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-F,则先加 "0"。

举例 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。

举例 加下划线的部分为注释。

:DIRECTIVE FILE FOR 789024

MEMORY MEM1: (01000H, 1000H) #SECOND MEMORY AREA

6.4.2 存储指令

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]

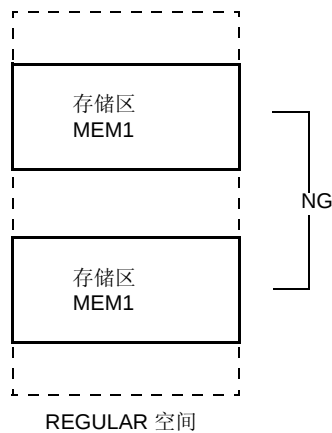
MEMORY Δ 存储区名 Δ : Δ (Δ 起始地址值 Δ , Δ 长度 Δ) [/ Δ 存储空间名]

(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：

- 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。
- 大小写字符认为是不同的字符。
- 大小写字符可混在一起使用。
- 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。
- 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，即使这些存储区位于不同的存储空间也不可以。

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：

- 使用 1 或大于 1 的数值。
- 如果长度指定更改为链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“使用注意事项”。

(4) 存储空间名

存储空间名显示于 64 KB 空间 REGULAR 中。

存储空间名的指定要求如下：

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

【功能】

- 对存储区名指定的存储区定义指定的存储空间。
- 用 1 条存储器指令可定义 1 个存储区。
- 可以多次指定存储器指令。但是，按指定次序进行多次定义将导致错误。
- 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，则只会指定各设备链接器带有的默认存储区。
- 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"。

【应用举例】

- 在存储空间 (EX1) 将地址 0H 至 1FFH 定义为 ROMA。

MEMORY ROMA : (0H , 200H)

6.4.3 程序段定位指令

程序段位置指令用来定位存储器指定区或特定地址中的指定程序段。

[语法]

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

(1) 程序段名

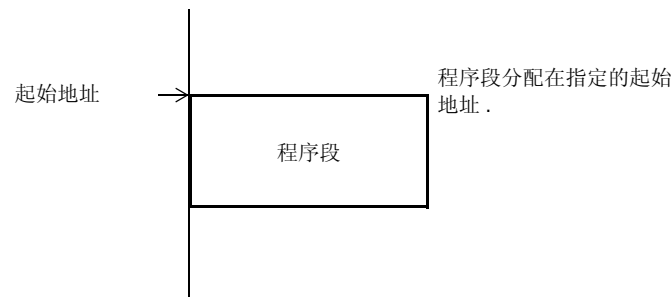
程序段名是输入链接器的目标文件中包含的程序段的名称。

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

(2) 起始地址

起始地址将程序段分配到 " 起始地址 " 指定的区。

- 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。
- 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范围，则会导致错误。

注意事项 2 对于由程序段指令的 **AT** 指定或由 **ORG** 指令指定的存储地址的程序段来说，不能用链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。

- 只有 **REGULAR** 可指定为存储区名。
- 存储空间名必须用大写字符指定。
- 当省略存储空间名时，则假定指定为 **REGULAR**。

确定程序段放置的目标地点的情况如下。

表 6-4 根据指定的存储区名以及存储空间名分配程序段

存储区名	存储空间名	程序段分配的目的地地址
未指定	未指定	REGULAR 空间的默认存储区
存储区名	未指定	REGULAR 空间的指定存储区

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。

[注意事项]

- 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。
- 若已定义的程序段名不存在程序段，则会出现错误。
- 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。

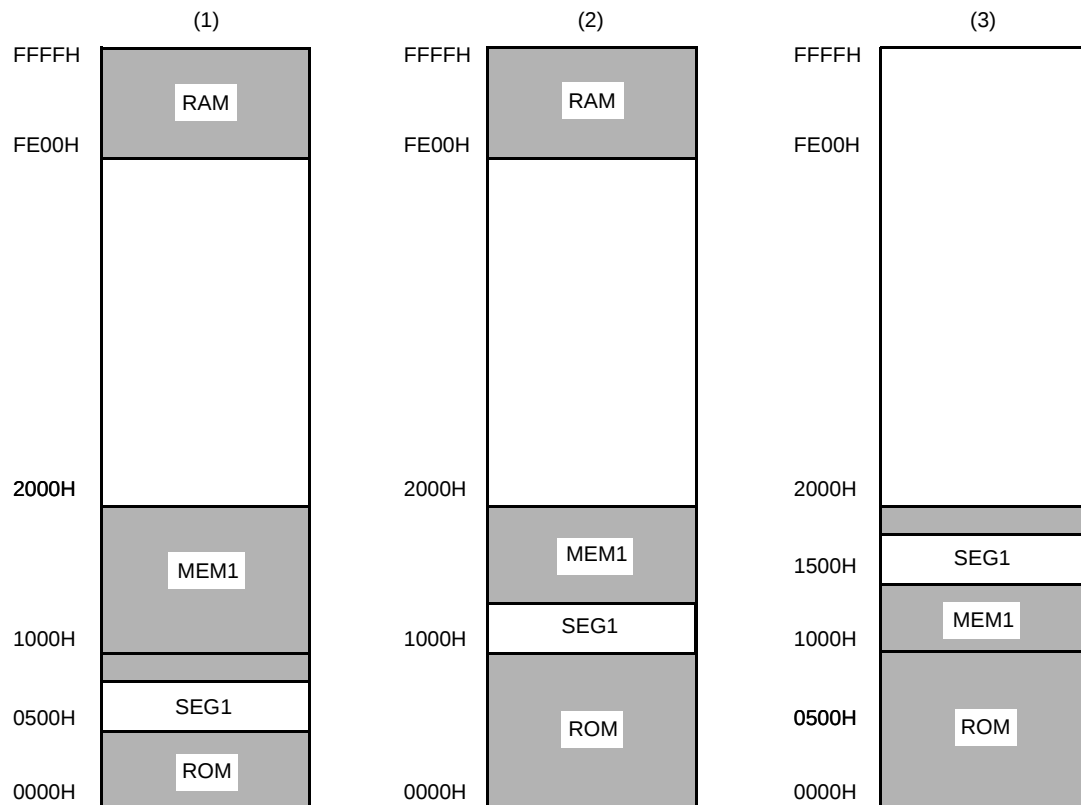
[应用举例]

- 为程序段 **SEG1** 分配地址，其段类型和重置属性为 **"CSEG UNIT"**。本例中声明的存储区如下。

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

- 当输入程序段 **SEG1** 分配到存储区 **ROM** 的 **500H** 处 (参见图 6-2 (1)).
MERGE SEG1 : AT (500H)
- 当输入程序段 **SEG1** 分配到存储区 **MEM1** 处 (参见图 6-2 (2)).
MERGE SEG1 : =MEM1
- 当输入程序段 **SEG1** 分配到存储区 **MEM1** 的 **1500H** 处 (参见图 6-2 (3)).
MERGE SEG1 : AT (1500H)=MEM1

图 6-2 程序段分配举例



6.5 启动链接器

6.5.1 启动链接器

可以用以下两种方法启动链接器。

(1) 从命令行中启动

X>	[路径名]	lk78k0s	[Δ 选项]	...	目标模块文件名	[Δ 选项]	...	[Δ]
	(a)	(b)	(c)			(e)		(d)

(a) 当前驱动器名

(b) 当前目录名

(c) 链接器命令文件名

(d) 包含链接器动作的细节指令。

如果指定不止一个链接器选项，则应用空格将选项隔开。在链接器选项中不区分大小写字母。关于链接器选项的详细说明，参见 "6.6 链接器选项"。

将包含空格的路径放入引号 (" ") 内。

(e) 包含链接器动作的细节指令。

输入模块中最多可输入 1024 项。

指定包含空格的路径的文件放入引号 (" ") 内。

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

(2) 从参数文件中启动

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

从参数文件启动链接器，在命令行指定参数文件选项 (-f)。

按如下所示，从参数文件中启动链接器

X>LK78K0S	[Δ 目标模块文件]	Δ -f 参数文件名
	(a)	(b)

(a) 指定参数文件选项。

(b) 包含启动链接器所必需的数据的文件。

备注 编辑器用于创建参数文件。

书写参数文件内容的规则如下所示：

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

- 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。
- 还可以在设置完选项后书写目标模块文件名。
- 在参数文件中写入所有的链接器选项，并输出在命令行中指定的文件名。

举例 用编辑器创建参数文件 (k0s.plk)。

< 参数文件 k0s.plk > 的内容

```
; 参数文件
k0smain.rel k0ssub.rel -ok0s.lmf -pk0s.map -e
-tc:\tmp
```

用参数文件 k0s.plk 启动链接器。

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

6.5.2 执行开始和结束信息

(1) 执行开始信息

当启动链接器时，屏幕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

如果没有检测到由链接造成的链接错误，链接器在屏幕上输出以下信息，控制权返回给主机操作系统。

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

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

如果检测到由链接造成的链接错误，链接器在屏幕上输出错误号，控制权返回给主机操作系统。

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

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

如果在链接期间链接器检测到致命错误，使其无法继续链接处理，则链接器会将一条信息输出到显示屏上，取消链接，将控制权还给操作系统。

举例 1 指定不存在目标模块文件。

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

```
78K/0S Series Linker Vx.xx [ xx xxx xx ]  
Copyright ( C ) NEC Electronics Corporation xxxx , xxxx  
  
RA78K0S error F3006 : File not found 'SAMP1.REL'  
RA78K0S error F3006 : File not found 'SAMP2.REL'  
Program Aborted.
```

在上面的例子中，指定了不存在的目标模块文件，导致错误，链接器异常中止链接。

举例 2 指定不存在的链接选项。

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

```
78K/0S Series Linker Vx.xx [ xx xxx xx ]  
Copyright ( C ) NEC Electronics Corporation xxxx , xxxx  
  
RA78K0S error F3018 : Option is not recognized '-z'  
Please enter 'LK78K0S --' , if you want help messages.  
Program Aborted.
```

在上面的例子中，指定了不存在的链接选项，导致错误，链接器异常中止链接。

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

6.6 链接器选项

6.6.1 链接器选项的类型

链接器选项是链接器操作的详细指令。链接器选项分为 19 种类型。

表 6-5 链接器选项

分类	选项	说明
载入模块文件输出指定	-o	指定载入模块文件的输出。
	-no	
强制载入模块文件输出指定	-j	强制载入模块文件输出。
	-nj	
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	
栈判定符生成指定	-s	自动生成栈判定公共符号。
	-ns	
指令文件指定	-d	输入指定文件作为指令文件。
链接列表文件输出指定	-p	指定输出链接列表文件。
	-np	
链接列表文件数据说明	-km	将映象表输出到链接列表文件中。
	-nkm	
	-kd	将链接指令文件输出到链接列表文件。
	-nkd	
	-kp	将公共符号列表输出到链接列表文件中。
	-nkp	
	-kl	将局部符号列表输出到链接列表文件中。
	-nkl	
链接列表格式指定	-ll	更改链接列表文件中 1 页能打印的行数。
	-lf	在每个列表文件的末尾插入换行代码。
	-nlf	
错误列表文件输出指定	-e	输出一个错误列表文件。
	-ne	
库文件指定	-b	输入指定文件作为库文件。
库文件读取路径指定	-i	从指定路径中读出库文件。
参数文件指定	-f	从指定文件中输入文件名和选项。
临时文件创建路径的指定	-t	在指定的路径中创建临时文件。
设备文件搜索路径指定	-y	从指定路径中读出设备文件。

表 6-5 链接器选项

分类	选项	说明
报警信息输出指定	-W	指定是否将警告信息输出到控制台。
帮助指定	--	在屏幕上输出帮助信息。

6.6.2 链接器选项的先后顺序

下表说明了当同时指定两个链接器选项时哪个链接器选项优先。

表 6-6 链接器选项的先后顺序

	-no	-ng	-np	-nkm	-nkp	-nkl	--
-j	NG						NG
-g	NG						NG
-p				Δ	Δ	Δ	NG
-km			NG				NG
-kd			NG	NG			NG
-kp		NG	NG				NG
-kl		NG	NG				NG
-ll			NG				NG
-lf			NG				NG

[用 NG 标出的项]

当指定水平轴的选项时，垂直轴所示选项不可用。

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

选项 -km 不可用。

[用 Δ 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。

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

同时指定 -nkm, -nkp 和 -nkl 项，所以 -p 项不可用。

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。

举例 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 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 链接器选项的说明

本节包含对各链接器选项的具体说明。

(1) 载入模块文件输出说明

载入模块文件输出说明 (-o/-no)

[语法]

```
-o [ 输出文件名 ]  
-no
```

- 默认预设

-o 输入文件名 .lmf

[功能]

- 选项 -o 指定载入模块文件的输出。此外还指定文件的输出地址和文件名。
- 选项 -no 使得选项 -o, -j 和 -g 不可用。

[应用]

- 使用 -o 选项指定载入模块文件的输出地址或者更改文件名。
- 当仅为了输出链接列表文件而进行链接时, 指定选项 -no。这样会缩短链接时间。

[说明]

- 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。
- 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。
- 如果在指定了 -o 选项时省略 "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。
- 如果在 "输出文件名" 中只指定了路径名, 则 "输入文件名 .lmf" 将输出到指定的路径。
- 如果同时指定了 -o 和 -no 选项, 那么最后指定的选项优先级较高。

[使用举例]

- 输出载入模块文件 k0s.lmf。

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

(2) 强制载入模块文件输出指定

强制载入模块文件输出指定 (-j/-nj)

[语法]

```
-j  
-nj
```

- 默认预设

-nj

[功能]

- 选项 -j 指定即便出现致命错误也要输出载入模块。
- 选项 -nj 使得 -j 不可用。

[应用]

- 通常情况下，当发生致命错误时，不能输出载入模块文件。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]

- 当选项 -j 指定时，即便出现致命错误也要输出载入模块。
- 如果同时指定了 -j 和 -nj 选项，那么最后指定的选项优先级较高。
- 如果指定 -no 选项，则 -j 选项不可用。

[使用举例]

- 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

调试数据输出指定 (-g/-ng)

[语法]

<pre>-g -ng</pre>

- 默认预设

-g

[功能]

- 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 .
- 选项 -ng 使得选项 -g, -kp 和 -kl 不可用 .

[应用]

- 当执行源程序器的符号调试时 , 确保使用 -g 选项 .

[说明]

- 当指定 -ng 选项时 , 公共符号列表和局部符号列表不能被输出 .
- 如果同时指定了 -g 和 -ng 选项 , 那么最后指定的选项优先级较高 .
- 如果指定 -no 选项 , 则 -g 选项不可用 .

[使用举例]

- 将指定调试数据添加到载入模块文件中 .

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

(4) 栈判定符生成的指定

栈判定符生成的指定 (-s/-ns)

[语法]

<pre>-s [存储区名] -ns</pre>

- 默认预设

-ns

[功能]

- 选项 -S 生成栈判定公共符号 "_@STBEG" 和 "_@STEND".
- 选项 -ns 使得 -s 不可用.

[应用]

- 使用 -s 选项来保留栈区.

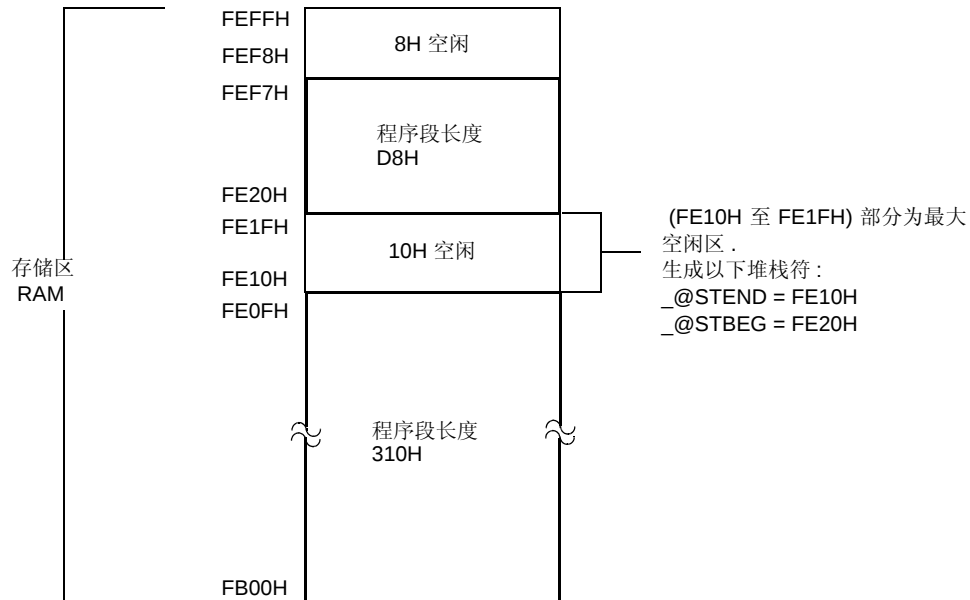
[说明]

- "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.
- "存储区名" 区分大小写字符.
- 链接器在 -s 选项指定的存储区中搜索未分配程序段的最大地址. 然后链接器生成公共符号 "_@STEND" 和 "_@STBEG", 前者的取值是最大地址区的起始地址, 后者取值是末尾地址 +1.
这些符号作为公开声明的 **NUMBER** 属性符号来处理, 在链接器符号表的末尾进行注册. 当将这些符号输出到链接列表文件时, 模块名称栏留空.
- 如果最大开放区为 10 字节或者更小, 则输出警告信息.
- 如果不存在最大空闲区, 则输出警告信息, 并且 "_@STEND" 和 "_@STBEG" 的取值都为末尾地址 +1.
- 如果省略 "存储区名", 则需指定 "RAM".
- 如果同时指定了 -s 和 -ns 选项, 那么最后指定的选项优先级较高.

[使用举例]

- 在存储区 RAM 中保留堆栈区 (但是 , 链接器将假定在 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

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]

- 当希望定义一个新的存储区域时，重新定义默认存储区，或者将程序段分配给特殊地址或存储区，此时就需要创建指令文件。使用 -d 选项将该指令文件输入到链接器。

[说明]

- 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，就会导致异常错误。
- 若省略文件名，则会导致异常错误。
- 不允许指令文件的嵌套。
- 在指令文件中可描述的字符数没有限制。
- 如果不止一次指定选项 -d, 或如果指定了不止一个文件名，则会产生异常错误。
- 关于指令文件的详细说明参见 "6.4 链接指令"。

[使用举例]

- 重定义默认存储区 ROM/RAM。

< 指令文件 k0s.dr 的内容 >

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

用 k0s.dr 执行链接。

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

(6) 链接列表文件输出指定

链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]

- 选项 -p 指定链接列表文件的输出。还指定输出文件的目标及文件名。
- 选项 -np 使得选项 -p, -km, -kd, -kp, -kl, -ll 和 -lf 不可用。

[应用]

- 指定选项 -p 用来更改链接列表文件的输出目标或输出文件名。
- 当仅为了输出载入模块文件而进行链接时，指定选项 -np。这样会缩短链接时间。

[说明]

- 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, 则会产生异常错误。
- 当指定 -p 选项时如果省略 "输出文件名", 则在当前目录的链接列表文件名变为 "输入文件名 .map"。
- 如果只指定 "输出文件名", 则 "输入文件名 .map" 将输出到指定的路径。
- 如果同时指定了 -p 和 -np 选项, 那么最后指定的选项优先级较高。

[使用举例]

- 创建链接列表文件 (k0s.map)。

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

(7) 链接列表文件数据指定

链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]

- 选项 -km 将映像列表输出到链接列表文件中。
- 选项 -nkm 使得 -km 不可用。

[应用]

- 指定 -km 选项将映像列表输出到链接列表文件中。

[说明]

- 如果同时指定 -nkm, -nkp 和 -nkl 选项, 则不能输出链接列表文件。
- 如果指定了 -nkm 选项, 则无法将链接指令文件输出到链接列表文件中。
- 如果同时指定了 -km 和 -nkm 选项, 那么最后指定的选项优先级较高。
- 如果指定 -np 选项, 则 -km 选项不可用。

[使用举例]

- 将映像列表输入到链接列表文件 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 xxx xxxx Page : 1

Command:k0smain.rel k0ssub.rel -pk0s.map -km
Para-file :
Out-file : K0SMAIN.LMF
Map-file : K0S.MAP
Direc-file :
Directive :

*** 链接信息 ***

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

*** 存储映射 ***

SPACE = REGULAR

MEMORY = ROM
BASE ADDRESS = 0000H  SIZE = 4000H
      OUTPUT  INPUT  INPUT  BASE    SIZE
      SEGMENT SEGMENT MODULE ADDRESS
      CODE    CODE    0000H  0002H  CSEG AT
      CODE    SAMPM    0000H  0002H
* gap *
      ?CSEG      0002H  007EH
      ?CSEG      0080H  0035H  CSEG
      ?CSEG      0080H  0015H
      ?CSEG      0095H  0020H
* gap *
      00B5H  3F4BH

MEMORY = RAM
BASE ADDRESS = FD00H  SIZE = 0300H
      OUTPUT  INPUT  INPUT  BASE    SIZE
      SEGMENT SEGMENT MODULE ADDRESS
* gap *
      DATA    DATA    SAMPM  FE20H  0003H  DSEG AT
      DATA    DATA    SAMPM  FE20H  0003H
* gap *
      FE23H  00DDH
* gap ( Not Free Area ) *
      FF00H  0100H
Target chip:uPD789026
Device file:Vx.xx

```

映射列表

(b) `-kd/-nkd`

[语法]

<code>-kd</code> <code>-nkd</code>

- 默认预设

`-kd`

[功能]

- 选项 `-kd` 将链接指令文件输出到链接列表文件中。
- 选项 `-nkd` 使得 `-kd` 不可用。

[应用]

- 指定选项 `-kd` 将链接指令文件输出到链接列表文件中。

[说明]

- 如果同时指定 `-nkm`, `-nkp` 和 `-nkl` 选项, 则不能输出链接列表文件。
- 如果指定了 `-nkm` 选项, 则无法将链接指令文件输出到链接列表文件中。
- 如果同时指定了 `-kd` 和 `-nkd` 选项, 那么最后指定的选项优先级较高。
- 如果指定 `-np` 选项, 则 `-kd` 选项不可用。

[使用举例]

- 将链接指令文件输出到链接列表文件 (k0s.map).

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

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 , 10000h )                <-- 指令文件内容
              memory RAM : ( 0fe20h , 1e0h )

*** 链接信息 ***

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

*** 存储映射 ***

SPACE = REGULAR

MEMORY = ROM
BASE ADDRESS = 0000H    SIZE = 1000H
      OUTPUT  INPUT    INPUT  BASE    SIZE
      SEGMENT SEGMENT  MODULE  ADDRESS
      CODE
      :
      0000H    0002H    CSEG AT
```

(c) `-kp/-nkp`

[语法]

<code>-kp</code> <code>-nkp</code>

- 默认预设

`-nkp`

[功能]

- 选项 `-kp` 将公共符号列表输出到链接列表文件中。
- 选项 `-nkp` 使得 `-kp` 不可用。

[应用]

- 指定选项 `-kp` 将公共符号列表输出到链接列表文件中。

[说明]

- 如果同时指定 `-nkm`, `-nkp` 和 `-nkl` 选项, 则不能输出链接列表文件。
- 如果指定了 `-ng` 选项, 则不能输出公共符号列表。
- 如果同时指定了 `-kp` 和 `-nkp` 选项, 那么最后指定的选项优先级较高。
- 如果指定 `-np` 选项, 则 `-kp` 选项不可用。

[使用举例]

- 将公共符号列表输出到链接列表文件。

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

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 : K0SMAIN.LMF
Map-file : K0S.MAP
Direc-file :
Directive :

*** 链接信息 ***

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

*** 存储映射 ***

SPACE = REGULAR
:

-----

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

*** 公共符号列表 ***
MODULE      ATTR      VALUE      NAME
SAMPM       ADDR      0000H      MAIN
SAMPM       ADDR      0080H      START
SAMPs       ADDR      009AH      CONVAH
              NUM      FE20H      _@STBEG
              NUM      FD00H      _@STEND

Target chip : uPD78xxx
  
```

公共符号列表

(d) -kl/-nkl

[语法]

```
-kl  
-nkl
```

- 默认预设

-nkl

[功能]

- 选项 -kl 将局部符号列表输出到链接列表文件中。
- 选项 -nkl 使得 -kl 不可用。

[应用]

- 指定选项 -kl 将局部符号列表输出到链接列表文件中。

[说明]

- 如果同时指定 -nkm, -nkp 和 -nkl 选项, 则不能输出链接列表文件。
- 如果指定了 -ng 选项, 则不能输出局部符号列表。
- 如果同时指定了 -kl 和 -nkl 选项, 那么最后指定的选项优先级较高。
- 如果指定 -np 选项, 则 -kl 选项不可用。

[使用举例]

- 将局部符号列表输出到链接列表文件 (k0s.map).

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

引用 k0s.map.

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

Command:k0smain.rel k4sub.rel -g -pk0s.map -kl -s
Para-file :
Out-file : K0SMAIN.LMF
Map-file : K0S.MAP
Direc-file :
Directive :

*** 链接信息 ***

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

*** 存储映射 ***

SPACE = REGULAR
:
-----

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

*** 局部符号列表 ***
```

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	00A4H	SASC
SAMPS	ADDR	00B2H	SASC1

局部符号列表

```

Target chip : uPD78xxx

```

(8) 链接列表格式指定

链接列表格式指定 (-ll, -llf/-nlf)

(a) -ll

[语法]

-ll [行数]

- 默认预设

-ll66 (以显示输出不分页为例)

[功能]

- 选项 -ll 更改链接列表文件中 1 页能打印的行数。

[应用]

- 指定选项 -ll 更改链接列表文件中 1 页能打印的行数。

[说明]

- 可用选项 -ll 指定的行数范围如下所示。

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

若指定为范围之外的数值或非数值，则会产生异常异常错误。

- 若省略行数，则指定为 66。
- 如果指定了 0 行，则不会出现分页符。
- 如果指定 -np 选项，则 -ll 选项不可用。

[使用举例]

- 在链接列表文件中指定每页 20 行。

```
C>lk78k0s k0smain.rel k0ssub.rel -pk0s.map -l120 -s
```

引用 k0s.map.

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

Command:k0smain.rel k0ssub.rel -pk0s.map -l120 -s
Para-file :
Out-file : K0SMAIN.LMF
Map-file : K0S.MAP
Direc-file :
Directive :

*** 链接信息 ***

      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

*** 存储映射 ***

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

MEMORY = LRAM
BASE ADDRESS = FE00H    SIZE = 0020H
      OUTPUT  INPUT    INPUT  BASE    SIZE
      :
```

(b) `-lf/-nlf`

[语法]

```
-lf  
-nlf
```

- 默认预设

`-nlf`

[功能]

- 选项 `-lf` 在链接列表文件的末尾插入分页符 (FF).
- 选项 `-nlf` 使得选项 `-lf` 不可用 .

[应用]

- 如果希望在打印链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]

- 如果指定 `-np` 选项, 则 `-lf` 选项不可用 .
- 如果同时指定了 `-lf` 和 `-nlf` 选项, 那么最后指定的选项优先级较高 .

[使用举例]

- 在链接列表文件的末尾添加分页符 .

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

(9) 错误列表文件输出指定

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]

- 指定选项 **-e** 用于指定错误列表文件的输出目标和文件名。
- 选项 **-ne** 使得 **-e** 不可用。

[应用]

- 指定选项 **-e** 用于改变错误列表文件的输出目标和输出文件名。

[说明]

- 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，如果指定设备型文件名 **CLOCK**，就会导致异常错误。
- 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 "。
- 当指定选项 **-e** 且省略驱动器名时，错误列表文件名将输出到当前驱动器。
- 如果同时指定了 **-e** 和 **-ne** 选项，那么最后指定的选项优先级较高。

[使用举例]

- 创建错误列表文件 (k0s.elk)。

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

在指令文件的内容中已经发生错误。引用 k0s.elk。

```
K0S.DR ( 3 ) : RA78K0S error E3102: Directive syntax error
```

(10) 库文件指定

库文件指定 (-b)

[语法]

-b 文件名

- 默认预设

无

[功能]

- 选项 -b 指定将文件作为库文件输入。

[应用]

- 链接器从库文件中检索输出模块引用的模块，仅将模块与输入模块联接。
- 库文件由于在单个文件中注册两个 (或多个) 模块。
- 通过创建可在许多程序中通用的库文件，文件管理和操作将变得更简单、更高效。指定 -d 选项将库文件输入到链接器。

[说明]

- 只有磁盘型文件名可指定为文件名。
- 文件名不可以被忽略。
- 如果指定了一个包含路径名的文件名，那么会从该路径输入库文件。如果在指定的路径中不存在库文件，就会发生错误。
- 如果指定了不包含路径名的文件名，则将从由 -i 选项指定的路径或从默认搜索路径输入库文件。
- 如果指定了两个以上的 -b 选项，那么将按指定的顺序输入库文件。最多可指定 10 个 -b 选项。

注意事项 在 PM+ 的 < 链接器选项 > 对话框中若要指定两个或更多的库，则需用逗号 (,) 将其隔开。

- 关于创建库文件的方法的信息说明，参见 " 第八章 库 "。

[使用举例]

- 输入库文件 (k0s.lib).
- (在库文件中注册 k0ssub.rel).

C>lk78k0s k0smain.rel -bk0s.lib

(11) 库文件读取路径指定

库文件读取路径指定 (-i)

[语法]

```
-i 路径名 [ , 路径名 ] ... ( 可指定两个或更多的路径名 )
```

- 默认预设

由环境变量 "LIB78K0S" 指定的路径

如果未指定路径, 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]

- 仅当 -b 选项指定了不包含路径名的库文件名时, -i 选项才有效。
- 可指定两个或更多的 -i。当指定两个或更多的路径时用 "," 将其隔开。在 "," 的之前或之后都不能键入空格。
- 每个链接最多可指定 10 个路径名。当指定两个或更多的路径名时, 链接器按指定顺序搜索库文件。
- 即便指定路径不存在库文件也不会导致错误。
- 若省略路径名, 则会导致异常错误。
- 如果 -b 指定了不包含路径名的库文件, 链接器将按下列顺序搜索路径。

(i) 用 -i 选项指定路径

(ii) 由环境变量 "LIB 78K0S" 指定路径。

(iii) 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。

[使用举例]

- 从路径 c:\libB 搜索库文件。

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

(12) 参数文件指定

参数文件指定 (-f)

[语法]

-f 文件名

- 默认预设

没有输入文件。

[功能]

- 选项 **-f** 指定链接器选项的输入和来自指定文件的输入文件名。

[应用]

- 当启动链接器所需的数据不适合命令行使用时，指定选项 **-f**。如果希望每次进行汇编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]

- 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，就会导致异常错误。
- 若省略文件名，则会导致异常错误。
- 不允许参数文件的嵌套。如果在参数文件内指定选项 **-f**，会产生异常异常错误。
- 在参数文件中可写入的字符数没有限制。
- 使用空格、制表符或者换行符 (**LF**) 来分隔选项或文件名。
- 命令行中，在为参数文件指定的位置上，参数文件中的选项和输入文件名会进行扩展。
- 最后指定的扩展选项优先级较高。
- 参数文件中紧跟 ";" 或 "#" 之后的字符都解释为注释，直到出现换行符 (**LF**) 或 EOF。
- 如果两次或更多次指定选项 **-f**，会产生异常异常错误。

[使用举例]

- 用参数文件执行链接。

< 参数文件 (k0s.plk) 的内容 >

```
; 参数文件  
k0smain.rel k0ssub.rel -ok0s.lmf -pk0s.map -e -s  
-tc:\tmp -g
```

在下命令行中键入以下内容。

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

(13) 临时文件创建路径的指定

临时文件创建路径的指定 (-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。

不指定路径时，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]

- 使用 -t 选项指定临时文件的创建地址。

[说明]

- 只能将一个路径指定为路径名。
- 路径名不可以被忽略。
- 即使存在先前创建的临时文件，如果文件不受保护，那么它将被覆盖。
- 一旦所需的存储器空间可用，临时文件就会在存储器中扩展。如果没有足够的存储空间，临时文件的内容将被写入磁盘。稍后可以通过所保存的磁盘文件访问这些临时文件。
- 当汇编结束时，删除临时文件。当通过按下 (CTRL-C) 键而使汇编终止时，也需删除临时文件。
- 根据以下顺序确定创建临时文件的路径。

(i) 由选项 -t 指定的路径

(ii) 由环境变量 TMP 指定的路径 (省略选项 -t 时)

(iii) 当前路径 (未设置 TMP 时)

当指定了 (i) 或 (ii) 时，如果无法在指定的路径中创建临时文件，则会发生终止错误。

[使用举例]

- 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

设备文件搜索路径指定 (-y)

[语法]

-y 路径名

- 默认预设

将按以下顺序确定的路径中读取设备文件 D.

- (i) <..\dev> (lk78k0s.exe 的启动路径)
- (ii) LK78K0S 的启动路径
- (iii) 当前命令
- (iv) 环境变量 PATH

[功能]

- 从指定路径中读出设备文件.

[应用]

- 指定存在设备文件的路径.

[说明]

- 如果在选项 -y 之后指定了非路径名, 则产生异常错误.
- 若在选项 -y 后省略路径名, 则会导致异常错误.
- 设备文件的读取路径按照如下次序确定.

- (i) 用 -y 选项指定路径
- (ii) <..\dev> (lk78k0s.exe 的启动路径)
- (iii) LK78K0S 的启动路径
- (iv) 当前目录
- (v) 环境变量 PATH

[使用举例]

- 指定设备文件路径为目录 c:\78k0s\dev

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

(15) 报警信息输出指定

报警信息输出指定 (-w)

[语法]

`-w [ 级别 ]`

- 默认预设

`-w1`

[功能]

- 选项 `-w` 指定是否将警告信息输出到控制台。

[应用]

- 指定输出报警信息的级别。

[说明]

- 如果在选项 `-w` 之后指定了级别名, 则产生异常错误。
- 只能指定 0 级、1 级和 2 级。
- 有下列输出级别:

0: 不输出警告信息。

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 "12.4 链接器错误信息"。

[使用举例]

- 在选项 `-w` 中指定 2 级。

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

(16) 帮助指定

帮助指定 (--)

[语法]

--

- 默认预设

不显示

[功能]

- 选项 -- 在显示屏显示帮助信息。

[应用]

- 帮助信息就是链接器选项的说明列表。在执行链接器时参考这些内容。

[说明]

- 当指定选项 -- 时，所有其它链接器选项均不可用。

注意事项 该选项不能在 PM+ 上指定。

若要参考 PM+ 的帮助，单击 < 链接器选项 > 对话框的 [Help] 按钮。

[使用举例]

- 指定选项 -- 时, 帮助信息输出到显示屏.

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 [ n ]         : Change warning level ( n = 0 to 2 ).
--              : Show this message.

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 ) / EX1
          MERGE CSEG1 : = ROM
          MERGE DSEG1 : AT ( 80H )

```

6.7 PM+ 指定选项设置

本节介绍 PM+ 链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 [LK], 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链接器选项。

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

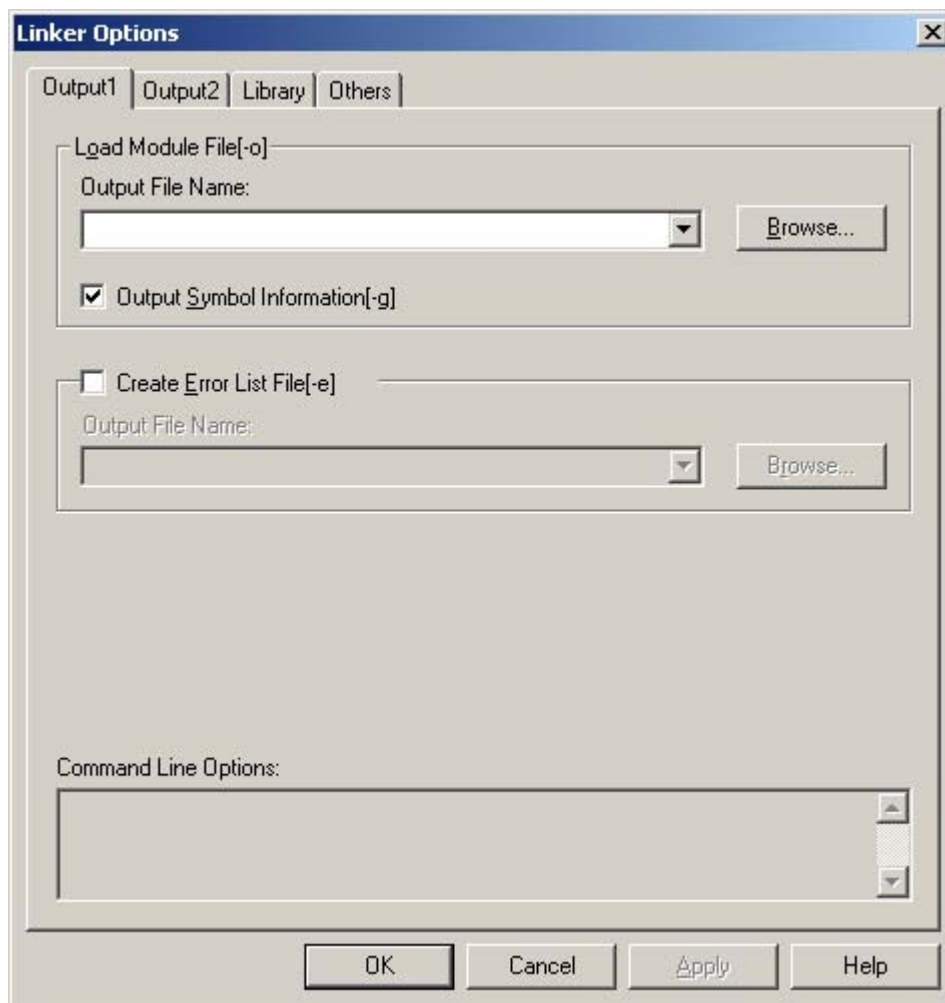


图 6-4 < Linker Options > 对话框（选择 << Output2 >> 标签时）

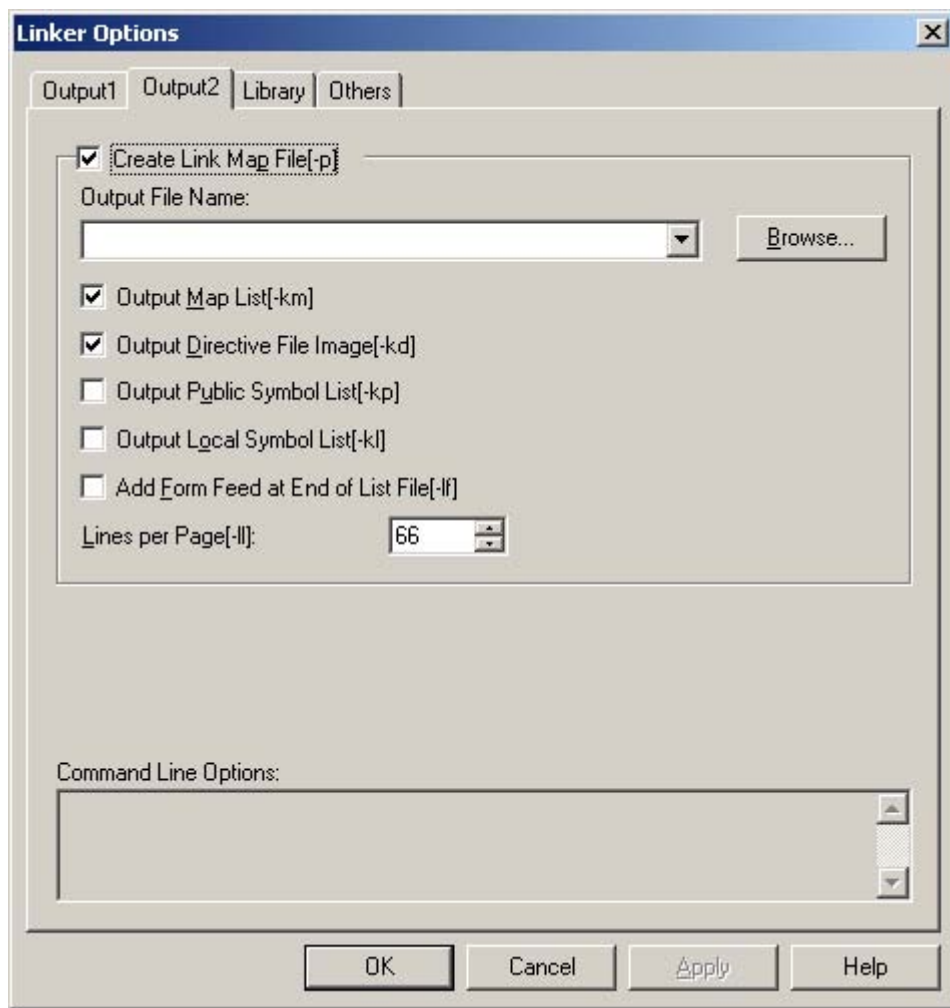


图 6-5 < Linker Options > 对话框（选择 << Library >> 标签时）

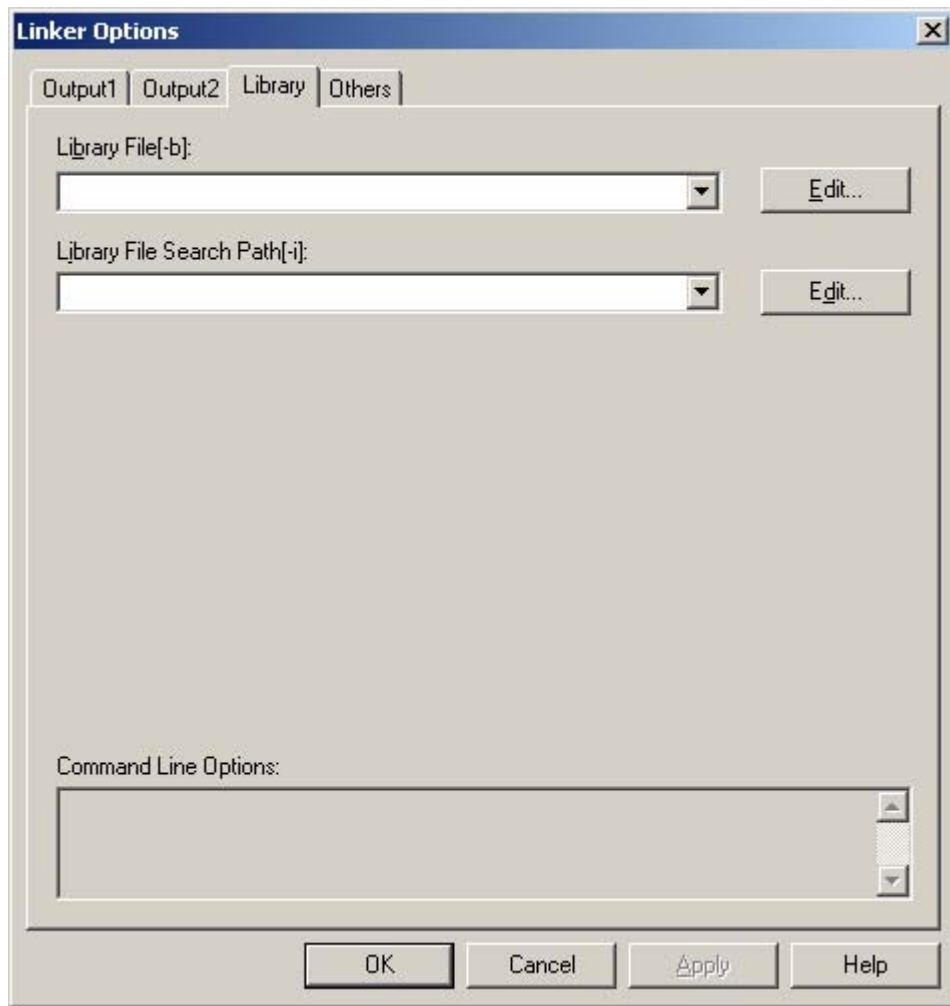
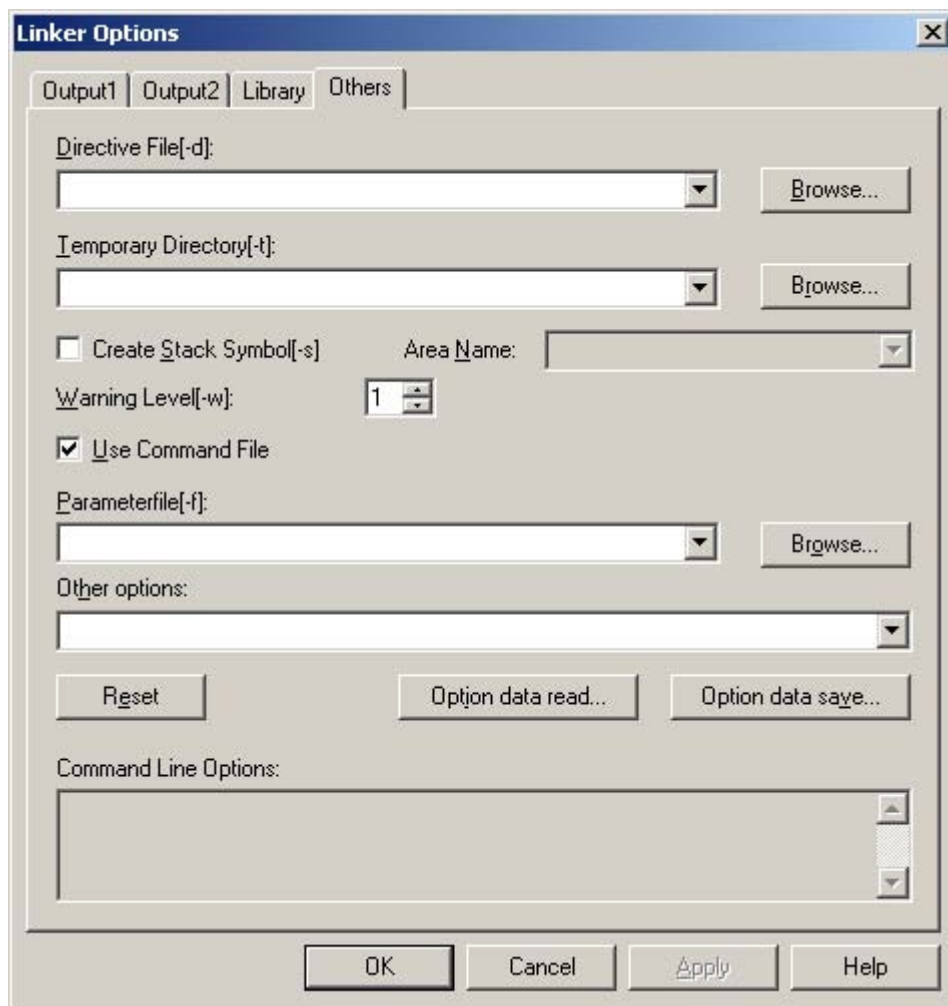


图 6-6 < Linker Options > 对话框（选择 << Others >> 标签时）



6.7.2 选项设置

下面介绍 < Linker Options > 对话框中的各选项。

(1) << Output1 >> 标签

- 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]

选取该选项以使得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]

选取该选项以输出错误列表文件。

输出文件名。

通过 [Browse] 按钮或直接输入，指定错误列表文件的路径名和文件名。

- Command Line Options

是只读编辑框，显示当前设置选项字符串。

(2) << Output2 >> 标签**- Create Link Map File [-p]**

选取该选项以输出链接列表文件。

输出文件名。

通过 **[Browse]** 按钮或直接输入，指定链接列表文件的路径名和文件名。

- Output Map List [-km]

选取该选项使得在链接列表文件中输出映射文件。

- Output Directive File Image [-kd]

选取该选项使得在链接列表文件中输出链接指令文件。

- Output Public Symbol List [-kp]

选取该选项使得在链接列表文件中输出公共符号列表。

- Output Local Symbol List [-kl]

选取该选项使得在链接列表文件中输出局部符号列表。

- Add Form Feed at End of List File [-lf]

选取该项使得在链接列表文件中添加分页符 (FF)。

- Lines per Page [-ll]

在 20 到 32767 中指定链接列表文件的单页的行数。

- Command Line Options

是只读编辑框。显示当前设置选项字符串。

(3) << Library >> 标签**- Library File [-b]**

通过 **[Edit]** 按钮或直接输入，将输入文件指定为库文件。

- Library File Search Path [-i]

通过 **[Edit]** 按钮或直接输入，指定从读取库文件的路径。

- Command Line Options

是只读编辑框。显示当前设置选项字符串。

(4) << Others >> 标签

- **D**irective File [-d]
通过 [B]rowse 按钮或直接输入，将输入文件指定为指令文件。
- **T**emporary Directory [-t]
通过 [B]rowse 按钮或直接输入，指定临时文件的创建路径。
- **C**reate **S**tack Symbol [-s]
当选中该选项时，保证存储区中的最大空区为栈区。
- **A**rea **N**ame
指定由用户定义的存储区名称或默认定义的存储区名称作为存储区名。
- **W**arning Level [-w]
指定报警信息输出级别。
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。
- **U**se Command File
选取该项创建命令文件。
- **P**arameterfile [-f]
通过 [B]rowse 按钮或直接输入，将输入文件指定为用户定义参数文件。
- **O**ther options
若要指定对话框中选项外的其它选项，可在输入框中输入该选项。

注意事项 帮助指定 (--) 选项不能在 PM+ 上指定。
- **R**eset
复位输入内容。
- **O**ption data read
打开 < Read Option Data > 对话框，在指定选项数据文件后该文件被读取。
- **O**ption data save
在打开 < Save Option Data > 对话框后，将选项数据文件保存到选项数据文件中并命名。
- **C**ommand Line Options
是只读编辑框。显示当前设置选项字符串。

6.7.3 编辑选项对话框

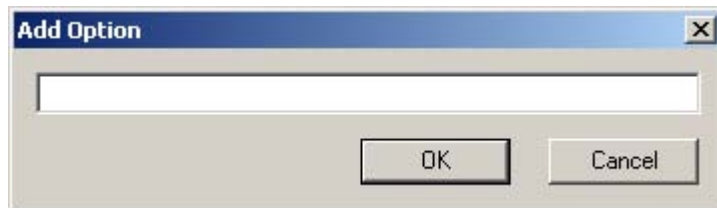
在 < Edit Option > 对话框中以列表形式编辑各项。

< Edit Option > 对话框描述如下。

图 6-7 < Edit Option > 对话框



图 6-8 < Add Option > 对话框



- [Add] 按钮

添加一个列表项。

如果添加的项为文件或目录，相应的打开 < Browse for Folder > 对话框。

其它情况下打开 < Add Option > 对话框。在该对话框中指定要添加项的详细内容。

- [Delete] 按钮

删除选定的列表项。

- [Up] 按钮

将选定的列表项上移。

- [Down] 按钮

将选定的列表项下移。

- [Add Sub Directory] 按钮

当项指定为 << Library >> 标签上的 Library File Search Path [-i] 时，则子目录可添加为选定列表的项。

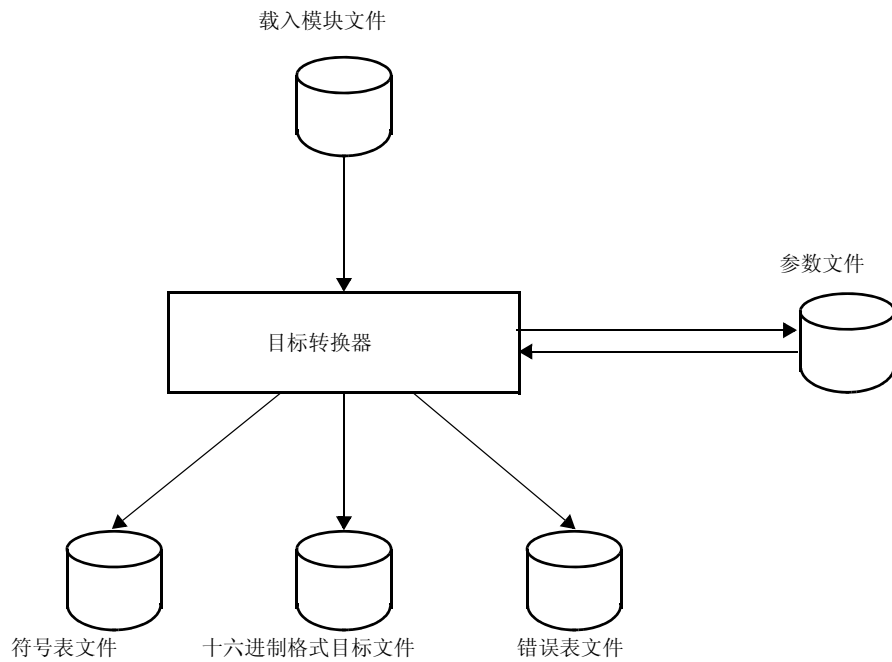
第七章 目标转换器

目标转换器输入由 RA 78K0S 链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模块文件 .

目标转换器还将用于符号调试的符号数据输出为符号列表文件 .

当发生目标转换器错误时 , 在屏幕上出现错误信息 , 以澄清错误原因 .

图 7-1 目标转换器的 I/O 文件



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	- 这些文件为目标码的二进制映射文件，输出为链接的结果。 - 这些文件由链接器输出。	.lmf
	参数文件	- 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	十六进制格式目标模块文件	- 通过将载入模块文件转换为十六进制对象格式而创建的文件。 - 这些文件用在掩膜ROM开发和PROM编程中。	.hex
	符号表文件	- 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	- 这些文件包含对象转换中产生的出错数据。	.eoc

7.2 目标转换器功能

7.2.1 十六进制格式目标模块文件

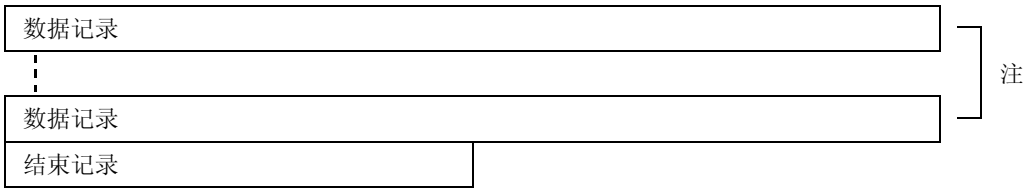
目标转换器输出的十六进制格式目标文件可以输入到十六进制的加载器中，如 PROM 编程器或调试器。

以下为举例程序的十六进制格式目标模块文件。

```
: 0200000080007E
: 10008000F5201AFC20FE229500F821FE0A27EB88B5
: 100090000A25EB30FE2F00000000630F22AC000A9F
: 1000A000E70A430F630F22AC000AE520130A380267
: 0500B0008307833020EE
: 00000001FF
```

[Intel 标准 HEX 格式目标模块文件格式]

图 7-2 Intel 标准格式



注 这里重复了数据记录。

(1) 数据记录

<u>.</u>	<u>02</u>	<u>0000</u>	<u>00</u>	<u>8000</u>	<u>7E</u>
↑	↑	↑	↑	↑	↑
(i)	(ii)	(iii)	(iv)	(v)	(vi)

- (i) 记录标记
指示记录的开头。
- (ii) 代码数 (2 位)
记录中存储代码的字节数 .. 最多可存储 16 个字节。
- (iii) 载入地址 (偏移量)
记录中显示的代码的起始地址 (偏移量) 以 4 位十六进制数字显示。
- (iv) 记录类型 (2 位)
固定在 00.
- (v) 代码 (最大 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。
- (vi) 校验和 (2 位)
从代码编号到代码递减计数的数据按顺序减去 0 得到的值被输入。

(2) 结束记录

<u>.</u>	<u>00</u>	<u>0000</u>	<u>01</u>	<u>FF</u>
↑	↑	↑	↑	↑
(i)	(ii)	(iii)	(iv)	(v)

- (i) 记录标记
- (ii) 代码数 , 固定在 00
- (iii) 固定在 0000
- (iv) 记录类型 , 固定在 01
- (v) 校验和 , 固定在 FF

7.2.2 符号表文件

由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]

符号表的开始	#	04	CR	LF			
公共符号的开始	;	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
		:	4 个空格	:	:	:	
	;	FF	4 个空格	模块名 1	CR	LF	各模块的局部符号
局部符号的开始	<	符号属性	符号值	局部符号名	CR	LF	
		符号属性	符号值	局部符号名	CR	LF	
		:	:	:	:	:	
	;	FF	4 个空格	模块名 2	CR	LF	
以目标模块为单位进行重复		:	:	:	:	:	
符号表结束标志	=	CR	LF				

注 1 符号属性是表 7-2 中显示的值。

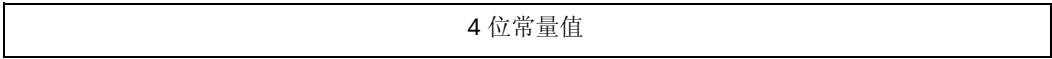
注 2 关于符号值，参见图 7-3.

表 7-2 符号属性的值

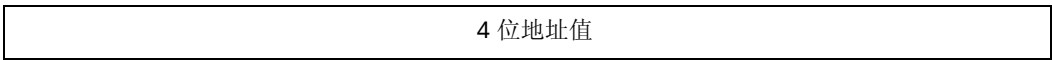
值	符号属性
00	由 EQU 伪指令定义的常量
01	代码段中的标号
02	数据段中的标号
03	ABBIT 属性的位符号
04	AWBIT 属性的位符号
05	saddr.bit/bsaddr.bit3 的位符号
06	wsaddr.bit4 的位符号
07	sfr.bit/bsfr.bit3 的位符号
08	wsfr.bit4 的位符号
09	RBBIT 属性的位符号
10	RWBIT 属性的位符号
11	byte register.bit 的位符号
FF	模块名

图 7-3 符号值格式

- 当符号属性为 NUMBER 时



- 当符号属性为 LABEL 时



- 当符号属性为位符号时



高 13 位： 从 0FE00H 开始的相对地址
低 3 位： 位位置 (0 至 7)

7.3 目标转换器的启动

7.3.1 目标转换器的启动

可以用以下两种方法启动目标转换器。

(1) 从命令行中启动

X>[路径名] oc78k0s [Δ 选项] ... 载入模块文件名 [Δ 选项] ... [Δ]
| | | | |
(a) (b) (c) (d) (e) (d)

- (a) 当前驱动器的名称
- (b) 当前目录名
- (c) 目标转换器命令文件名
- (d) 包含目标转换器动作的细节指令。
将包含空格的路径放入引号 (" ") 内。

(e) 载入模块的文件名将被转换。
如果指定多个目标转换器选项，用空格将选项隔开。在目标转换器选项中不区分大小写字母。关于目标转换器选项的详细说明，参见“7.4 目标转换器选项”。指定包含空格的路径的文件放入引号 (" ") 内。

例 C>oc78k0s k0s.lmf -osample.hex

(2) 从参数文件中启动

当启动目标转换器所必需的数据与命令行不符时，或者每次执行目标转换都重复指定同一目标转换选项时，使用参数文件。

从参数文件启动目标转换器，在命令行指定参数文件选项 (-f)。

按如下所示，从参数文件中启动目标转换器。

X>oc78k0s [Δ 载入模块文件] Δ -f 参数文件名
| |
(a) (b)

- (a) 指定参数文件选项
- (b) 包含启动目标转换器所必需的数据的文件

备注 编辑器用于创建参数文件。

书写参数文件内容的规则如下所示：

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

备注 1 如果从命令行中省略载入模块文件名，则在参数文件中只能指定 1 个载入模块文件名。

备注 2 还可以在设置完选项后书写载入模块件名。

备注 3 在参数文件中写入所有的目标转换器选项，并输出在命令行中指定的文件名。

例 用编辑器创建参数文件 (k0s.poc)。

< k0s.poc 的内容 >

```
; 参数文件
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 执行开始和结束信息

(1) 执行开始信息

当启动目标转换器时，屏幕上出现执行启动的信息。

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

(2) 执行结束信息

如果没有检测到由目标转换造成的目标转换错误，目标转换器在屏幕上输出以下信息，控制权返回给主机操作系统。

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

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

如果检测到由目标转换造成的目标转换错误，目标转换器在屏幕上输出错误码，控制权返回给主机操作系统。

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

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

如果在目标转换期间目标转换器检测到致命错误，使其无法继续链接处理，则目标转换器会将一条信息输出到显示屏上，取消目标转换，将控制权还给操作系统。

例 1 指定不存在载入模块文件名。

```
C>oc78k0s sample.lmf

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

RA78K0S error F4006 : 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。发生错误，目标转换器中止目标转换。

例 2 指定不存在目标转换器选项。

```
C>oc78k0s k0s.lmf -a

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

RA78K0S error F4018 : 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。发生错误，目标转换器中止目标转换。
当显示报错信息且目标转换异常中止时，在“第十二章 错误信息”中查找原因，采取相应措施。

7.4 目标转换器选项

7.4.1 目标转换器选项的类型

目标转换器选项是目标转换器操作的详细指令。目标转换器选项分为 10 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

类型	选项	说明
HEX 格式目标模块文件输出指定	-o	指定十六进制格式目标模块文件的输出。
	-no	
符号表文件输出指定	-s	指定输出符号表文件。
	-ns	
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	
对象补充指定	-u	将指定的补充值作为目标码输出到无 HEX 格式对象输出的地址区。
	-nu	
错误列表文件输出指定	-e	输出一个错误列表文件。
	-ne	
参数文件指定	-f	从指定文件中输入文件名和选项。
设备文件搜索路径指定	-y	从指定路径中读出设备文件。
帮助指定	--	在屏幕上输出帮助信息。

7.4.2 目标转换器选项的说明

本节包含对各目标转换器选项的具体说明。

(1) HEX 格式目标模块文件输出指定

HEX 格式目标模块文件输出指定 (-o/-no)

[语法]

```
-o [ 输出文件名 ]  
-no
```

- 默认预设

-o 输入文件名 .HEX

[功能]

- 选项 -o 指定十六进制格式目标模块文件的输出。选项 -o 还指定输出目标和输出文件名。
- 选项 -no 指定不输出十六进制格式目标文件。

[应用]

- 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。
- 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]

- 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，就会导致异常错误。
- 如果在指定了 -o 选项时省略 "输出文件名"，则 HEX 格式目标模块文件 "输入文件名 .hex" 将输出到当前目录。
- 如果在 "输出文件名" 中只指定了路径名，则 "输入文件名 .hex" 将输出到指定的路径。
- 如果同时指定了 -o 和 -no 选项，那么最后指定的选项优先级较高。

[使用举例]

- 输出 HEX 格式目标模块文件 (sample.hex)。

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

(2) 符号表文件输出指定

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]

- 选项 -s 指定输出符号表文件。选项 -s 还指定输出目标和输出文件名。
- 选项 -ns 指定没有符号表文件输出。

[应用]

- 指定选项 -s 用于改变符号表文件的输出目标和输出文件名。
- 在仅为了输出 HEX 格式目标模块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]

- 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，就会导致异常错误。
- 如果在指定了 -s 选项时省略 " 输出文件名 "，则符号表文件 " 输入文件名 .sym " 将输出到当前目录。
- 如果在 " 输出文件名 " 中只指定了路径名，则 " 输入文件名 .sym " 将输出到指定的路径。
- 如果同时指定了 -s 和 -ns 选项，那么最后指定的选项优先级较高。

[使用举例]

- 输出符号表文件 (sample.sym)。

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

(3) 以目标地址次序进行排序的指定

以目标地址次序进行排序的指定 (-r/-nr)

[语法]

```
-r  
-nr
```

- 默认预设

-r

[功能]

- 选项 -r 输出以地址次序排序的 HEX 格式对象。
- 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]

- 当不需要以地址次序对 HEX 格式对象进行排序时，指定选项 -nr。

[说明]

- 如果同时指定了 -r 和 -nr 选项，那么最后指定的选项优先级较高。
- 如果指定 -no 选项，则 -r/-nr 选项不可用。

[使用举例]

- 按地址次序进行 HEX 格式目标排序。

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

(4) 目标补充指定

目标补充指定 (-u/-nu)

[语法]

-u 补充值 [, [起始] , 长度] -nu

- 默认预设

-u0FFH (用 0FFH 填充)

[功能]

- 选项 -u 将指定的补充值作为目标码输出到无 HEX 格式对象输出的地址区。
- 选项 -nu 使得 -u 不可用。

[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，其作用是不可预料的。通过指定选项 -u, 可将代码预先写入 HEX 格式对象尚未输出的地址区。

[说明]

- 可指定为补充值的取值范围如下。

$$0H \leq \text{补充值} \leq 0FFH$$

补充值可用二进制、八进制、十进制或十六进制指定。若指定为范围之外的数值或非数值,则会产生异常异常错误。

- " 起始 " 指定进行补充的起始地址区。

可指定为起始地址的取值范围如下。

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

起始值可用二进制、八进制、十进制或十六进制指定。若指定为范围之外的数值或非数值,则会产生异常异常错误。如果省略起始值,则假定指定 0。

- " 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。

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

长度可用二进制、八进制、十进制或十六进制指定。若指定为范围之外的数值或非数值,则会产生异常异常错误。如果指定起始值,则不能省略长度。

- 如果都省略起始值和长度,则目标转换器执行以下处理。

(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。

- 如果同时指定了 -u 和 -nu 选项, 那么最后指定的选项优先级较高。
- 如果指定了 -no 选项, 则 -u/-nu 选项无效。
- 不能用选项 -u 指定两个 (或更多) 地址区。
- 下面是选项 -u 中起始值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 到长度 - 1 地址

(c) -u 补充值, 起始, 长度

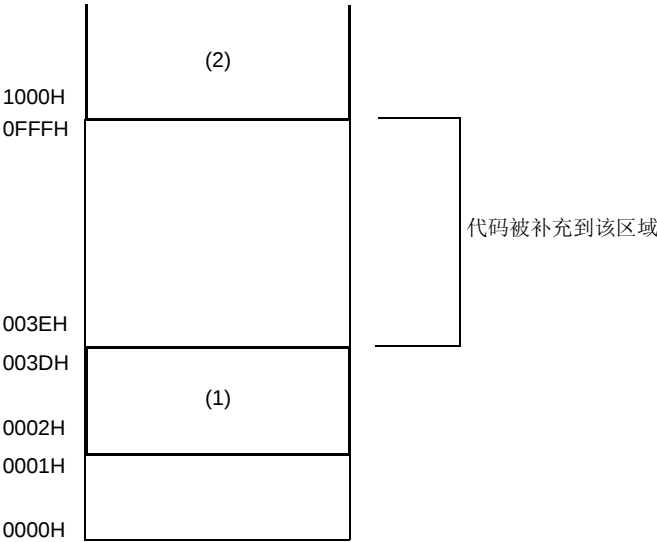
从起始地址到起始地址 + 长度 - 1 地址

[使用举例]

- 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      ; (1)
: 100012001A6720FE2822006521FED350D25014FE1A      ; (1)
: 10002200B900059F2835002431B900059F28350005      ; (1)
: 0C003200242156AF0A8302A807A830560C
: 01000003B5D0d0026A3...                             ; (2)
: 1010100024A5F622B667...                             ; (2)
:
: 00000001FF
```



00H 被补充到 003EH 至 0FFFH 地址区域 .

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

(5) 错误列表文件输出指定

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]

- 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。
- 选项 -ne 使得 -e 不可用。

[应用]

- 指定选项 -e 用于改变错误列表文件的输出目标和输出文件名。

[说明]

- 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，如果指定设备型文件名 CLOCK, 就会导致异常错误。
- 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 "。
- 当指定选项 -e 且省略驱动器名时，错误列表文件名将输出到当前驱动器。
- - 如果同时指定了 -e 和 -ne 选项，那么最后指定的选项优先级较高。

[使用举例]

- 创建错误列表文件 (k0s.eoc).

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

(6) 参数文件指定

参数文件指定 (-f)

[语法]

-f 文件名

- 默认预设

没有输入文件。

[功能]

- 选项 **-f** 指定链接器选项的输入和来自指定文件的输入文件名。

[应用]

- 当启动目标转换器所需的数据不适合命令行使用时，指定选项 **-f**。
- 每次执行目标转换时指定选项 **-f** 以重复指定相同的选项，并将这些选项存储到参数文件中。

[说明]

- 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，就会导致异常错误。
- 若省略文件名，则会导致异常错误。
- 不允许参数文件的嵌套。如果在参数文件内指定选项 **-f**，会产生异常异常错误。
- 在参数文件中可写入的字符数没有限制。
- 使用空格、制表符或者换行符 (**LF**) 来分隔选项或输入的文件名。
- 命令行中，在为参数文件指定的位置上，参数文件中的选项和输入文件名会进行扩展。
- 最后指定的扩展选项优先级较高。
- ";" 或 "#" 之后的字符以及换行码 (**LF**) 或 "EOF" 之前的字符均当做注释。
- 如果两次或更多次指定选项 **-f**，会产生异常异常错误。

[使用举例]

- 用参数文件执行目标转换 .

< 按以下设置参数文件 78k0s.poc 的内容 >.

```
; 参数文件  
k0s.lmf -osample.hex  
-ssample.sym -r
```

在命令行键入以下内容 .

```
C>oc78k0s k0s.lmf -f78k0s.poc
```

(7) 设备文件搜索路径指定

设备文件搜索路径指定 (-y)

[语法]

-y 路径名

- 默认预设

设备文件将按以下顺序的路径读取。

- (i) <..\dev> (oc78k0s.exe 的启动路径)
- (ii) OC78K0S 的启动路径
- (iii) 当前目录
- (iv) 环境变量 PATH

[功能]

- 从指定路径中读出设备文件。

[应用]

- 指定存在设备文件的路径。

[说明]

- 如果在选项 -y 之后指定了非路径名，则产生异常错误。
- 若在选项 -y 后省略路径名，则会导致异常错误。
- 设备文件的读取路径按照如下次序确定。
 - (i) 由选项 -y 指定的路径
 - (ii) <..\dev> (oc78k0s.exe 的启动路径)
 - (iii) OC78K0S 的启动路径
 - (iv) 当前目录
 - (v) 环境变量 PATH

[使用举例]

- 将设备文件路径指定为目录 c:\78k0s\dev

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

(8) 帮助指定

帮助指定 (--)

[语法]

```
--
```

- 默认预设

不显示

[功能]

- 选项 -- 在屏幕上输出帮助信息。

[应用]

- 帮助信息就是目标转换器选项的说明列表。在执行目标转换器时参考这些内容。

[说明]

- 当指定选项 -- 时,所有其它选项均不可用。

注意事项 不能从 PM+ 中指定该选项。

若要参考 PM+ 帮助,单击 < Object Converter Options > 对话框中的 [Help] 按钮。

[使用举例]

- 当指定选项 -- 时,帮助信息在显示屏上输出。

```
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 the specified name] / Not.
-r/-nr         : Sort HEX object by address / Not.
-uvalue [ , [ start ] , size]/-nu :Fill up HEX object with specified value /
Not.
-ydirectory    : Set device file search path.
--            : Show this message.
DEFAULT ASSIGNMENT : -o -s -r -nr
```

7.5 PM+ 中的选项设置

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 [OC], 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（选择 << Output1 >> 标签时）

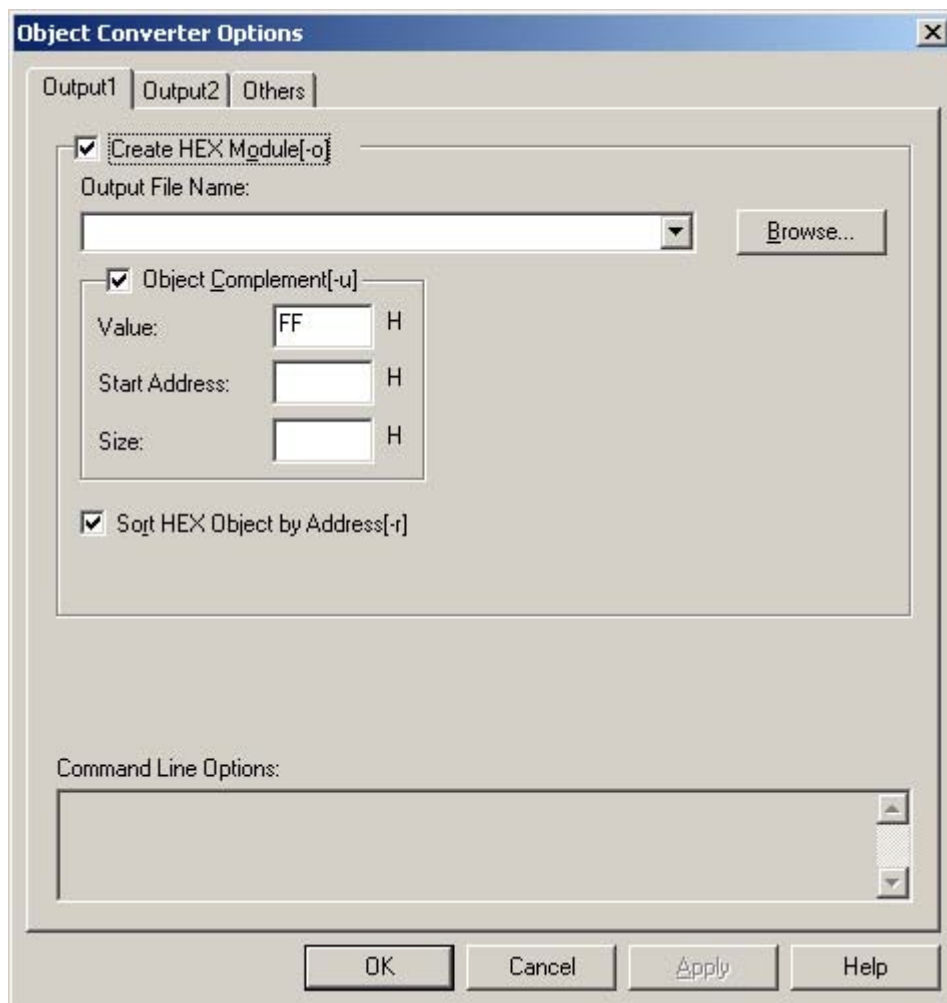


图 7-5 < Object Converter Options > 对话框（选择 << Output2 >> 标签时）

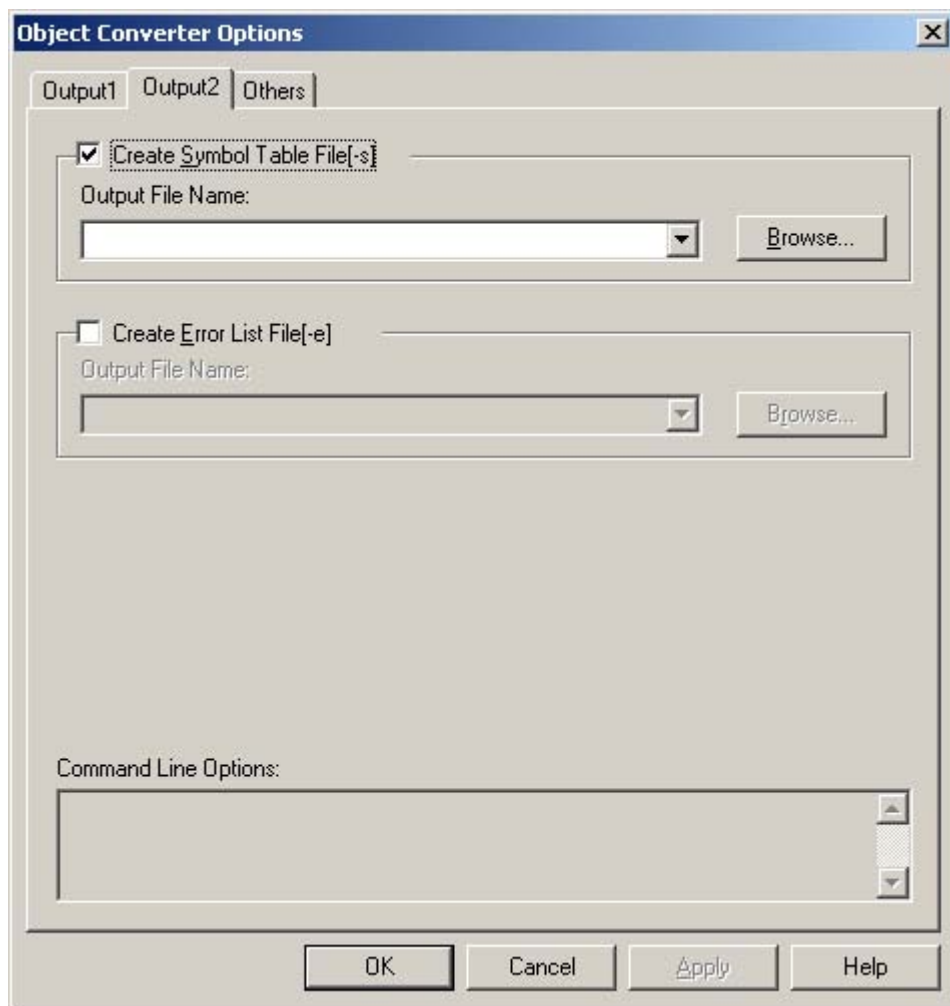
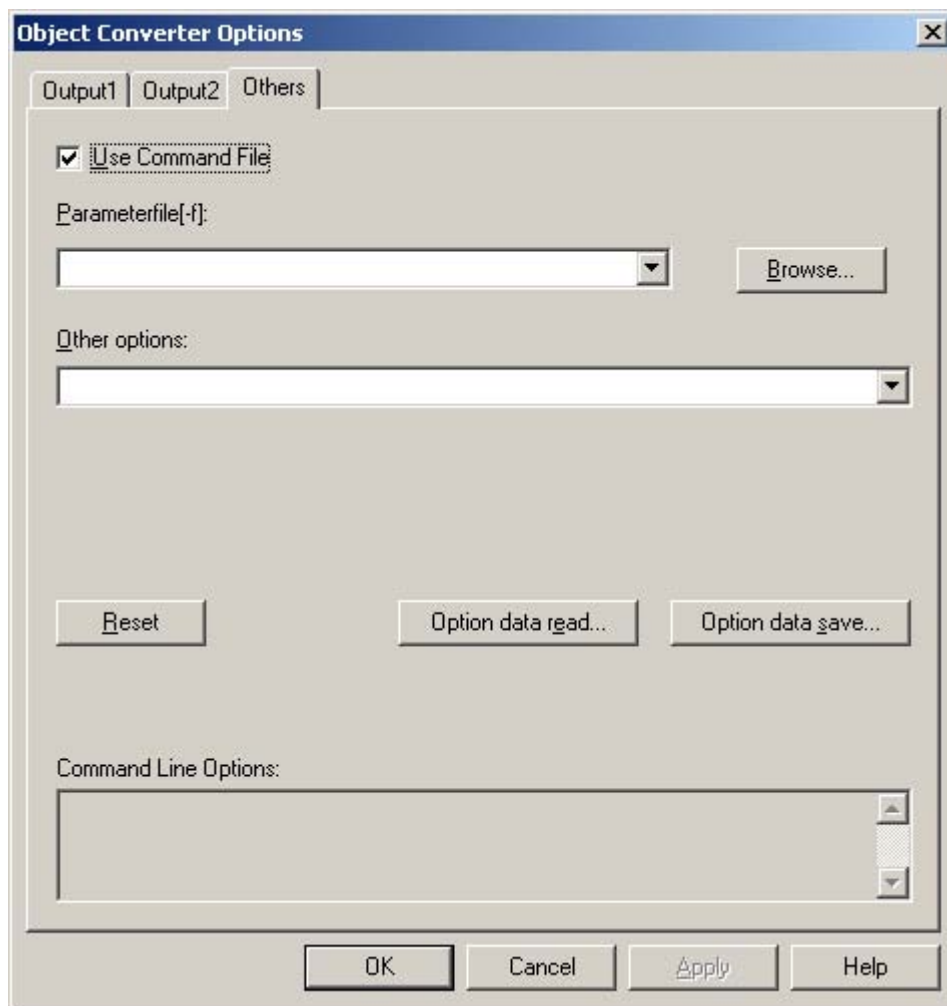


图 7-6 < Object Converter Options > 对话框（选择 << Others >> 标签时）



7.5.2 选项设置

< Object Converter Options > 对话框中的各选项描述如下。

(1) << Output1 >> 标签

- Create HEX Module [-o]

选取该项输出 HEX 格式目标模块文件，
输出文件名。

通过 [Browse] 按钮或直接输入，指定目标模块文件的输出路径。

- Object Complement [-u]

指定对象预先将代码写入无 HEX 格式对象输出的地址，防止非法代码写入并防止发生程序挂起。

- Sort HEX Object by Address [-r]

如果 HEX 格式对象必须以地址次序排序则需选择该项。

- Command Line Options

是只读编辑框。显示当前设置选项字符串。

(2) << Output2 >> 标签

- Create Symbol Table File [-s]

选取该选项以输出符号表文件。
输出文件名。

通过 [Browse] 按钮或直接输入，指定符号表文件的路径名和文件名。

- Create Error List File [-e]

选取该选项以输出错误列表文件。
输出文件名。

通过 [Browse] 按钮或直接输入，指定错误列表文件的路径名和文件名。

- Command Line Options

是只读编辑框。显示当前设置选项字符串。

(3) << Others >> 标签**- Use Command File**

选取该项创建命令文件。

- Parameterfile [-f]

通过 [Browse] 按钮或直接输入，将输入文件指定为用户定义参数文件。

- Other options

若要指定对话框中选项外的其它选项，可在输入框中输入该选项。

注意事项 帮助指定 (--) 选项不能在 PM+ 上指定。

- Reset

复位输入内容。

- Option data read

打开 < Read Option Data > 对话框，在指定选项数据文件后读取该文件。

- Option data save

在打开 < Save Option Data > 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

是只读编辑框。显示当前设置选项字符串。

第八章 库管理程序

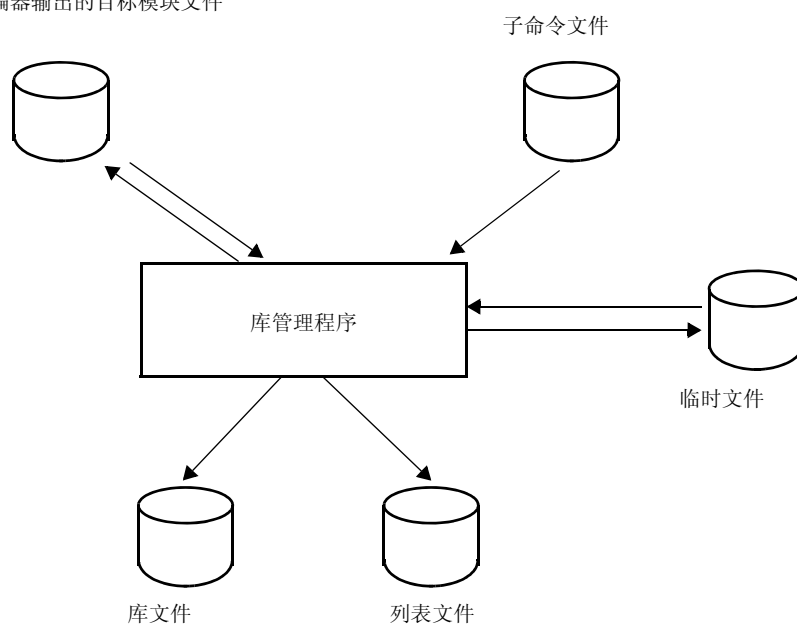
库管理程序编辑 RA 78K0S 目标模块文件和以一个模块为单位的库文件。

库管理器也输出列表文件。

如果库管理程序发生错误，则会有错误信息输出到显示屏上，以显示错误原因。

图 8-1 库管理程序的 I/O 文件

由编译器或汇编器输出的目标模块文件



8.1 库管理程序的 I/O 文件

库管理程序的 I/O 文件如下所示。

表 8-1 库管理程序的 I/O 文件

类型	文件名	说明	默认文件类型
输入文件	子命令文件	- 这些文件包含了执行程序命令和参数。 - 这些文件是由用户创建的。	无
输出文件	列表文件	- 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	- 这些是由汇编器或编译器输出的目标模块文件。	.rel
	库文件	- 这些文件输入由库管理程序输出的文件并更新其内容。	.lib
	临时文件	- 这些文件是在形成库时由库管理程序自动生成的。当库管理程序的执行完成时，删除这些文件。	Lbxxxxxx.\$y (y = 1-6)

8.2 库管理程序的功能

(1) 形成模块库

编译器和链接器为其输出的每个模块都创建一个文件。

这意味着如果创建了大量模块，则文件数也会增大。因此，RA 78K0S 包括一个将大量目标模块收集到一个单一文件的功能。该功能就是所谓的模块库形成，而作为库的文件就称为库文件。

可以将库文件输入到链接器。通过创建一个由模块组成的可通用于许多程序的库文件，在执行模块化编程时，用户就能方便有效地进行文件的管理和操作。

(2) 编辑库文件

该库管理程序结合了以下库文件编辑功能。

- (a) 向库文件中添加模块
- (b) 从库文件中删除模块
- (c) 替换库文件中模块
- (d) 从库文件中拾取模块

有关这些功能的详细说明，参考 "8.5 子命令"。

(3) 库文件数据的输出

库管理程序结合了编辑和输出如下存储在库文件中的数据项的功能。

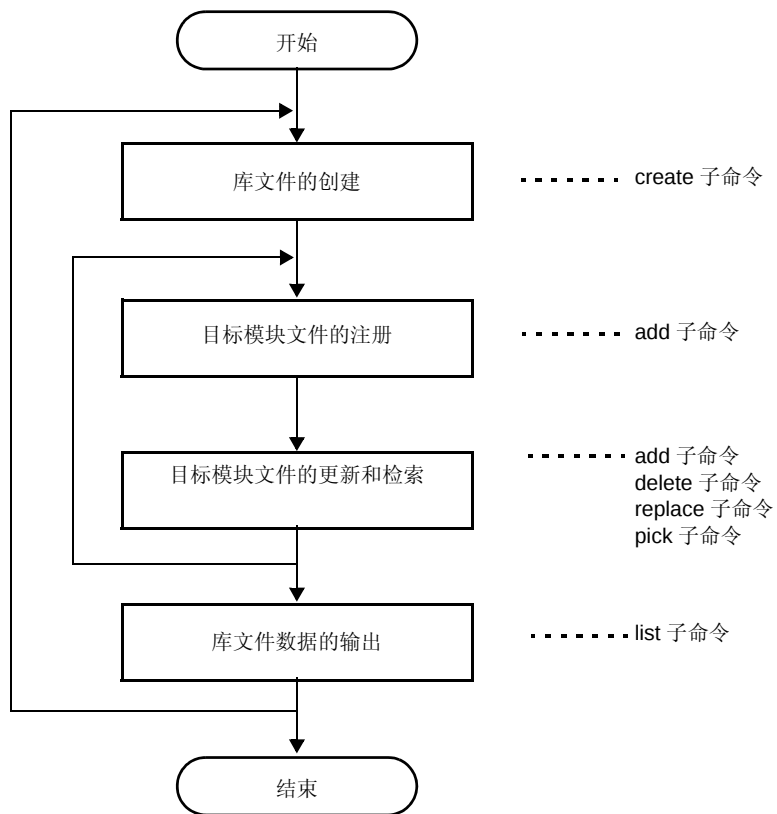
- (a) 模块名
- (b) 创建的程序
- (c) 注册日期
- (d) 更新日期
- (e) PUBLIC 符号数据

注意事项 库管理程序使用子命令执行以上阐述的第 (2) 和第 (3) 项功能。在执行处理时，库管理程序按顺序确定各个子命令。有关子命令的操作说明，参考 "8.5 子命令"。

(4) 创建库文件的步骤

创建库文件的大体步骤如下所示。

图 8-2 创建库文件的步骤



8.3 库管理程序的启动

8.3.1 库管理程序的启动

可以用以下两种方法启动库管理程序。

(1) 从命令行启动

```
x>[ 路径名 ] lb78k0s [ Δ 选项 ] ...
|      |           |           |
(a)    (b)         (c)         (d)
```

- (a) 当前驱动器名
- (b) 当前目录名
- (c) 库管理程序命令文件名
- (d) 这包括库管理程序操作的详细说明

如果指定一个以上的库管理程序选项，则用空格隔开各选项。库管理程序选项不区分大小写字母。关于库管理程序选项的详细说明，请参见“8.4 库管理程序选项”。

将包含空格的路径放入一对引号（" "）内。

示例 C>lb78k0s -ll20 -lw80

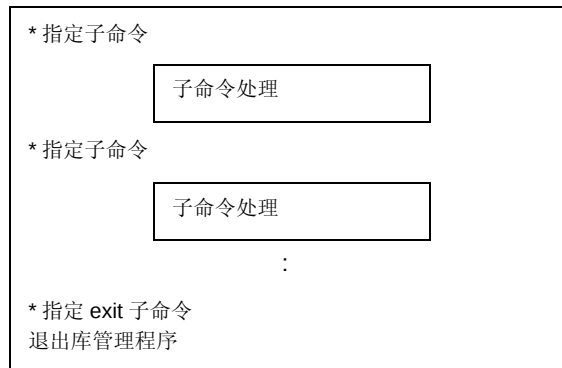
当启动库管理程序时，显示屏上会出现如下启动信息。

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

在星号（*）后，指定一个库管理程序子命令。

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

当子命令输入完成时，就开始各子命令的处理。当完成一个子命令的处理时，在屏幕上又出现 "*", 库管理程序等待下一个待输入的子命令。库管理程序一直重复这种操作直到有 **exit** 子命令输入为止。



一行中最多可指定 128 个字符。

如果所有所需的操作数据在一行中写不下时，使用 "&" 在下一行继续指定。最多可以连续指定 15 行。

(2) 从子命令文件启动

所谓的子命令文件就是存储了库管理程序子命令的文件。

当启动库管理程序时,如果没有指定子命令文件,则在 "*" 出现后,必须指定多次子命令。通过创建子命令文件,这些多次指定的子命令文件就能同时一起处理。

每次指定库形成过程,在重复地指定同一子命令时,还可以使用子命令文件。

当使用子命令文件时,在文件名前需用 "<" 加以描述。

如下所示,从子命令文件启动库管理程序。

```
x>lb78k0s Δ < 子命令文件名 [ Δ 选项 ] ...
          |         |
          (i)       (ii)
```

(i) 当指定一个子命令文件时,务必将添加此符号

(ii) 存储子命令的文件

(a) 使用编辑器创建子命令文件。

(b) 书写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) 重复一个子命令时,在每行的结尾描述 "&" 符号以指示继续执行。

(d) 从分号 (";") 到行结尾之间所描述的内容将作为注释,且不会由库管理程序命令对其进行解释。

(e) 如果子命令文件的最后一个子命令不是 **exit** 子命令,则库管理程序将自动将其解释为 **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。在处理完子命令文件中的所有子命令后,退出库管理程序。

示例 使用编辑器创建子命令文件 (k0s.slb)。

< k0s.slb 的内容 >

```
; 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 执行启动和结束消息

(1) 执行启动消息

当启动库管理程序时，屏幕上出现执行启动的消息。

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

(2) 执行结束消息

库管理程序不输出执行结束消息。当用户在所有处理完成后键入 **exit** 子命令时，库管理程序将控制权返回给操作系统

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

如果库管理程序检测到致命错误，该错误使库管理程序处理不能持续下去，则库管理程序在显示屏上输出一条消息并将控制权返回给操作系统。

示例 指定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
RA78K0S error F5018 : Option is not recognized '-z'
Usage : LB78K0S [ options ]
```

在上面的例子中，指定了不存在的库管理程序选项，导致出错，库管理程序中止执行。

当显示一条错误消息且中止库的形成时，在“第十二章 错误消息”中查找原因，并采取相应操作。

8.4 库管理程序选项

8.4.1 库管理程序选项的类型

库管理程序选项用于指定列表文件的格式和临时文件的文件创建路径。库管理程序选项分为 4 种类型。

表 8-2 库管理程序选项

分类	选项	说明
列表文件格式指定	-lw	更改可在列表文件中 1 行上打印的字符数。
	-ll	更改可在列表文件中 1 页上打印的行数。
	-lf	在列表文件的末尾插入分页符。
	-nlf	
临时文件创建路径的指定	-t	在指定的路径中创建临时文件。
设备文件搜索路径指定	-y	从指定路径中读出设备文件。
帮助指定	--	在显示屏上显示帮助信息。

8.4.2 库管理程序选项的说明

下面是库选项的详细说明。

(1) 列表文件格式指定

列表文件格式指定 (-lw, -ll, -lf/-nlf)

(a) -lw

[语法]

-lw [字符数]

- 默认假定

-lw132 (在显示输出中为 80 个字符)

[功能]

- 选项 -lw 更改可在列表文件中 1 行上打印的字符数。

[应用]

- 指定选项 -lw 以更改可在列表文件中 1 行上打印的字符数。

[说明]

- 可使用选项 -lw 指定的字符数的范围如下所示。

(显示输出情况下为 80 个字符)

$72 \leq 1$ 行中可打印的字符时 ≤ 260

如果指定的数值超出了此范围,或者指定了除数值之外的东西,则会发生异常错误。

- 如果省略了字符数,则会指定 132. 如果将列表文件输出到显示屏上,则指定 80.
- 指定的字符数不包括结束符 (CR,LF)。
- 如果没有指定 list 子命令,则忽略 -lw 选项。
- 如果两次或多次指定 -lw 选项,则最后指定的项优先。

[使用示例]

- 指定 80 作为列表文件中每一行的字符数。

```
C>lb 78k0s -lw80
```

(b) -ll

[语法]

-ll [行数]

- 默认假定

-ll66 (显示输出时无分页符)

[功能]

- 选项 -ll 指定可在列表文件中 1 页上打印的行数。

[应用]

- 指定选项 -ll 以更改可在列表文件中 1 页上打印的行数。

[说明]

- 可使用选项 -ll 指定的行数的范围如下所示。

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

- 如果指定的数值超出了这个范围,或者指定了数值以外的东西,则发生异常错误。

- 如果省略了行数,则会指定 66。

- 如果指定的行数为 0,则不会出现分页符。

- 如果没有指定 list 子命令,则忽略 -ll 选项。

- 如果两次或多次指定 -ll 选项,则最后指定的项优先。

[使用示例]

- 指定 20 作为列表文件中每一页的行数。

```
C>lb78k0s -ll20
```

(c) -lf/-nlf

[语法]

```
-lf  
-nlf
```

- 默认假定
-nlf

[功能]

- 选项 -lf 会在列表文件的末尾插入分页符（FF）。
- 选项 -nlf 使选项 -lf 不可用。

[应用]

- 如果希望在打印的列表文件内容后面添加分页符，则指定选项 -lf 以添加一个换页码。

[说明]

- 如果没有指定 list 子命令，则忽略 -lf 选项。
- 如果同时指定了 -lf 和 -nlf 选项，那么最后指定的选项优先。

[使用示例]

- 向列表文件添加一个分页符。

```
C>lb 78k0s -lf
```

(2) 临时文件创建路径的指定

临时文件创建路径的指定 (-t)

[语法]

`-t 路径名`

- 默认假定
在由环境变量 **TMP** 所指定的路径中创建。
如果没有指定路径, 则在当前路径中创建临时文件。

[功能]

- 选项 **-t** 在指定的路径中创建临时文件。

[应用]

- 使用选项 **-t** 来指定临时文件创建的位置。

[说明]

- 仅能将一个路径指定为路径名。
- 不可省略路径名。
- 即使存在先前创建的临时文件, 如果文件不受保护, 那么它将被覆盖重写。
- 一旦所需的存储器空间可用, 临时文件就会在存储器中扩展。如果没有足够的存储空间可用, 正创建的临时文件的内容将被写入磁盘。
稍后可以通过所保存的磁盘文件访问这些临时文件。
- 当库形成完成时, 删除临时文件。当库形成因按下 (**CTRL-C**) 键而中止时, 也会删除临时文件。
- 根据以下顺序确定待创建的临时文件的路径。
 - (i) 由选项 **-t** 指定的路径
 - (ii) 由环境变量 **TMP** 指定的路径 (省略 **-t** 选项时)
 - (iii) 当前路径 (未设置 **TMP** 时)

当指定了 (i) 或 (ii) 时, 如果无法在指定的路径中创建临时文件, 则会发生异常错误。

[使用示例]

- 指定临时文件输出到目录 **c:\tmp**。

```
C>lb 78k0s -tc:\tmp
```

(3) 设备文件搜索路径指定

设备文件搜索路径指定 (-y)

[语法]

-y 路径名

- 默认假定

将按以下顺序从确定的路径中读取设备文件。

- (i) <..\dev> （用于 lb78k0s.exe 的启动路径）
- (ii) 启动 LB78K0S 的路径
- (iii) 当前目录
- (iv) 环境变量 PATH

[功能]

- 从指定路径中读出设备文件。

[应用]

- 指定设备文件存在的路径。

[说明]

- 如果在选项 -y 后指定的不是路径名内容，则会发生异常错误。
- 如果在选项 -y 后省略路径名，就会发生异常错误。
- 按如下确定的顺序的路径读取设备文件。
 - (i) 由选项 -y 指定的路径
 - (ii) <..\dev> （用于 lb 78k0s.exe 的启动路径）
 - (iii) 启动 LB78K0S 的路径
 - (iv) 当前目录
 - (v) 环境变量 PATH

[使用示例]

- 将设备文件的路径指定为目录 c:\78k0s\dev

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

(4) 帮助指定

帮助指定 (--)

[语法]

```
--
```

- 默认假定
无显示

[功能]

- 选项 -- 在显示屏上显示帮助信息。

[应用]

- 帮助信息就是子命令的说明列表。指定库管理程序时参阅这些说明。

[说明]

- 当指定 -- 选项时，所有其他的选项均不可用。

注意事项 不能从 PM+ 中指定该选项。

要参考 PM+ 帮助，在 < 库文件名 > 对话框中单击 [帮助] 按钮。

[使用示例]

- 当指定了 -- 选项时，在显示屏上输出帮助信息。

```
C>lb78k0s --

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


(1) create

create

[语法]

```
create Δ 库文件名 [ Δ 处理 ]
```

- 默认假定
c

[功能]

- create 子命令用来创建一个新的库文件。

[说明]

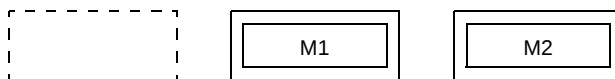
- 创建的库文件的大小变为 0。
- 当指定某一处理时，创建库文件的同时也完成了模块的注册。
- 库文件名：如果同名的文件已经存在，则会覆盖该文件。
- 处理：如果目标模块文件带有与库文件内相同的公共符号，则该目标模块文件不能注册。
与库文件中模块同名的模块不能注册。
- 如果发生错误，则会中断处理，且不能创建库文件了。

[使用示例]

- 在创建库文件的同时注册模块 M1 和 M2。

```
*create k0s.lib m1.rel m2.rel
```

< 文件创建前 >



< 文件创建后 >

k0s.lib



(2) add

add

[语法]

```
add Δ 库文件名 Δ 处理
```

- 默认假定
a

[功能]

- add 子命令就是向库文件中添加一个模块。

[说明]

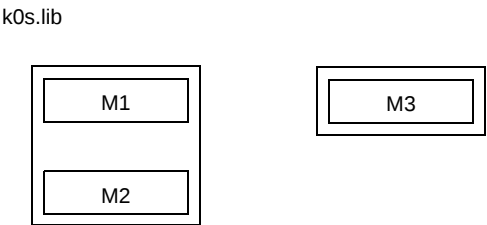
- 即使库中当前没有存储任何模块，仍能向库文件中添加模块。
- 如果在库文件中已经存在与待添加模块同名的模块，则会发生错误。
- 如果待添加的模块带有与库文件中相同的公共符号，则会发生错误。

[使用示例]

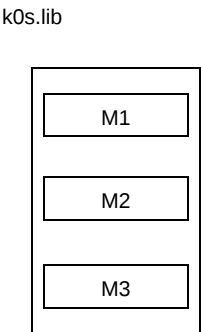
- 向库文件（k0s.lib）添加一个模块（M3）。

*add k0s.lib m3.rel

< 添加前 >



< 添加后 >



(3) delete

delete

[语法]

```
delete Δ 库文件名 Δ ( Δ 模块名 [ Δ , ... ] Δ )
```

- 默认假定
d

[功能]

- delete 子命令就是从库文件中删除一个模块。

[说明]

- 如果指定的模块在库文件中并不存在，则会发生错误。
- 如果发生错误，则会中断处理，且不会改变库文件的条件。

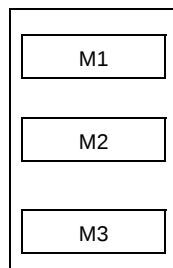
[使用示例]

- 从库文件（k0s.lib）中删除模块（M1, M3）。

```
*delete k0s.lib ( m1.rel, m3.rel )
```

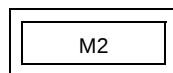
< 删除前 >

k0s.lib



< 删除后 >

k0s.lib



(4) replace

replace

[语法]

<code>replace Δ 库文件名 Δ 处理</code>

- 默认假定
r

[功能]

- replace 子命令就是用其它目标模块文件替换库文件中的模块。

[说明]

- 如果库文件中没有与替换模块名称相同的模块，则会导致错误发生。
- 如果替换模块中包含的公共符号与库文件中的公共符号同名，则会发生错误。
- 替换目标模块的文件名必须与注册中使用的文件名相同。
- 如果发生错误，则会中断处理，且不会改变库文件的条件。

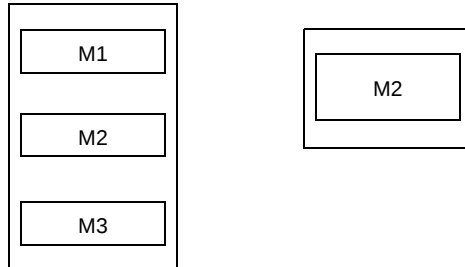
【使用示例】

- 替换库文件（k0s.lib）中的一个模块（M2）。

*replace k0s.lib m2.rel

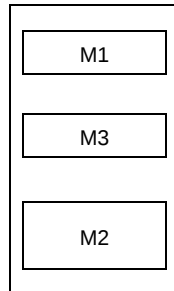
< 替换前 >

k0s.lib



< 替换后 >

k0s.lib



由于在库文件中模块（M2）被删除后便注册了新模块（M2），所以 M2 在库文件中次序的最后。

(5) pick

pick

[语法]

```
pick Δ 库文件名 Δ ( Δ 模块名 [ Δ , ... ] Δ )
```

- 默认假定
p

[功能]

- pick 子命令就是从存在的库文件中拾取指定的模块。

[说明]

- 拾取的模块变为目标模块文件，且该文件的文件名为其在库文件中注册的文件名。
- 如果指定的模块在库文件中并不存在，则会发生错误。
- 如果发生错误，则会中断处理。但是，如果当指定两个或多个模块时发生错误，则在导致错误出现的模块之前拾取的模块变为可用，并存储至磁盘上。

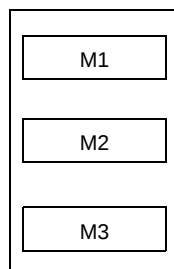
[使用示例]

- 从库文件（k0s.lib）中拾取一个模块（M2）。

```
*pick k0s.lib ( m2.rel )
```

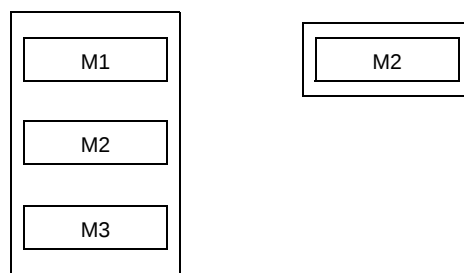
< 拾取前 >

k0s.lib



< 拾取后 >

k0s.lib



(6) list

list

[语法]

```
list [ Δ 选项 ] Δ 库文件名 [ Δ ( Δ 模块名 [ Δ , ... ] Δ ) ]
      选项   : -public/-nopublic
              : -o Δ 文件名
```

- 默认假定

|

[功能]

- list 子命令就是输出库文件中的模块数据。

[说明]

- 可以指定多个选项。选项中不区分大小写字母。
- -o :
可以将设备类型文件名指定为输出文件名。
如果省略了输出文件名,则会发生错误。
如果省略了文件类型,则库管理程序认为键入了 " 输入文件名 .LST".
- -public/-nopublic :
该选项可以由仅指定下划线的字母来进行选定。
-public 指定公共符号数据的输出。
-nopublic 使 -public 不可用。
如果同时指定 -public 和 -nopublic, 则最后指定的选项优先。

【使用示例】

- 将库文件（k0s.lib）中的模块数据输出至列表（k0s.lst）中。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）。

```
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

help

[语法]

```
help
```

- 默认假定
h

[功能]

- Help 子命令就是在显示屏上显示帮助信息。

[说明]

- 帮助信息是子命令的列表及对每个命令的说明。在指定库管理程序时，指定 **help** 命令或 -- 选项以参考该帮助信息。

[使用示例]

- 指定 **help** 命令以输出帮助信息。

```
*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

exit

[语法]

exit

- 默认假定
e

[功能]

- exit 子命令就是退出库管理程序。

[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 PM+ 中的选项设置

本节介绍了从 PM+ 中指定库文件的方法。

8.6.1 选项设置方法

从 PM+ 的 [Tools] 菜单中选择 [Librarian Options] 或者在工具栏中单击 [LB] 按钮以显示 < Librarian Options > 对话框。指定完路径和文件名后，单击 [Next] 以显示 < Subcommand > 对话框。

通过在该对话框中输入所需的选项，就能设置不同的库管理程序选项。

图 8-3 < Librarian Options > 对话框（当选中 << Output >> 标签时）

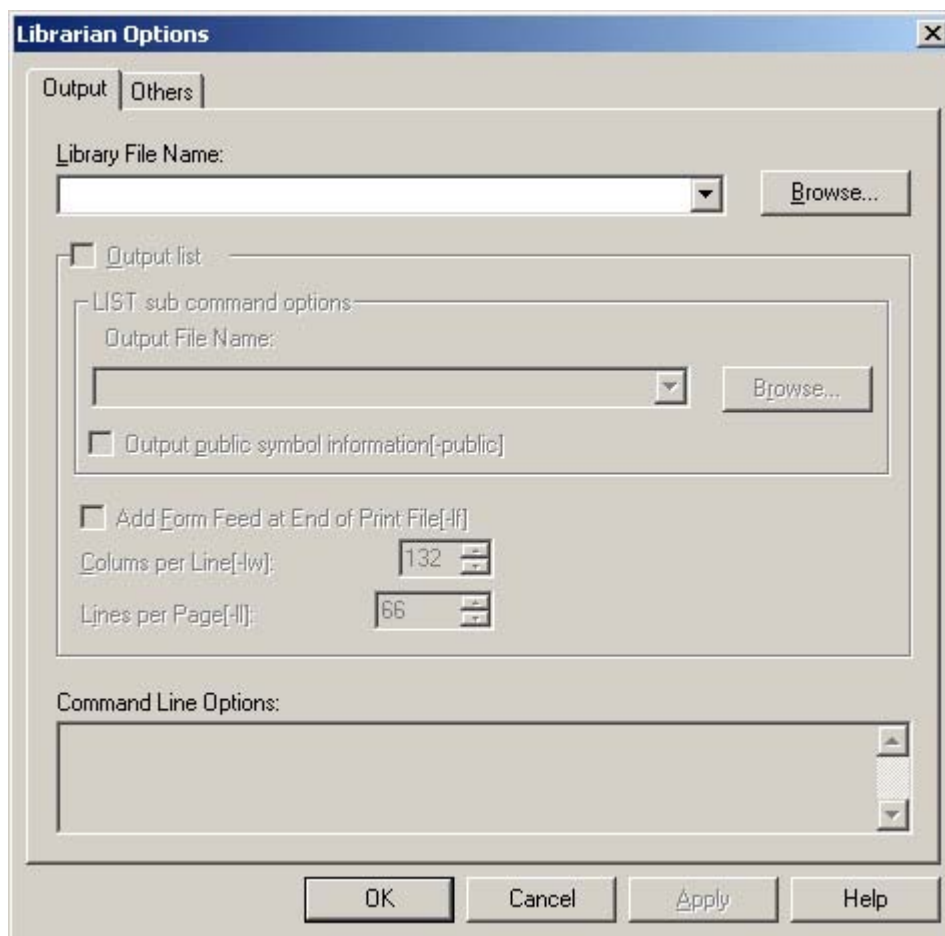
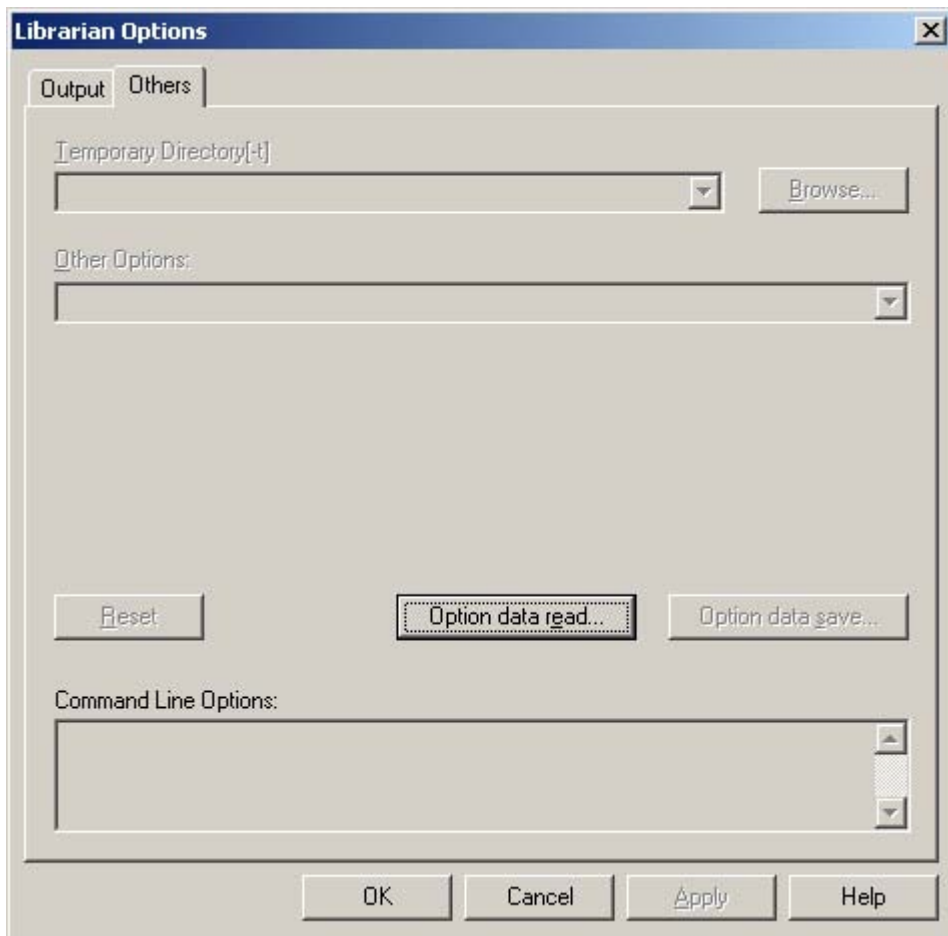


图 8-4 < Librarian Options > 对话框（当选中 << Others >> 标签时）



8.6.2 选项设置

< Librarian > 对话框中的各个选项描述如下。

(1) << Output >> 标签

- Library File Name

通过使用 [Browse] 按钮或直接输入文件名来指定库文件的名称。

- LIST sub command options

选中该框以输出列表文件。

- Output File Name

通过使用 [Browse] 按钮或直接输入文件名来指定列表文件的名称和路径。

- Output public symbol information[-public]

选中该框，从而将公共符号信息添加至待输出的列表文件。

- Add Form Feed at End of Print File[-lf]

选中该框，从而在库文件的末尾添加分页符 (FF)。

- Columns per Line[-lw]

指定库文件中一行的字符数。可以指定的字符数范围为 72 ~ 260。

- Lines per Page[-ll]

指定库文件中一页的行数。可以指定的行数范围为 20 ~ 32767。

- Command Line Options

这是只读编辑框。显示当前设置的选项字符串。

(2) << Others >> 标签**- Temporary directory [-t]**

使用 [Browse] 按钮或直接输入一个路径名来指定临时文件要创建的路径。

- Other Options

要指定能在对话框中设置的选项之外的选项，就要在输入框中输入该选项。

注意事项 不能在 **PM+** 上指定帮助指定 (--) 选项。

- Reset

清除输入内容。

- Option data read

打开 < Read Option Data > 对话框，在指定了选项文件后，读取该文件。

- Option data save

打开 < Save Option Data > 对话框，在指定的名称下保存选项文件。

- Command Line Options

这是只读编辑框。显示当前设置的选项字符串。

8.7 PM+ 中处理库文件的方法

本部分描述了从 PM+ 中处理库文件的方法。 < Edit Option > 对话框如下所示。

8.7.1 处理方法

通过从 PM+ 的 [Tools] 菜单中选择 [Register Ex-tool], 对独立模式中开始 LB 的执行格式进行注册。

选中注册的图标, 然后打开 < Library File Name > 对话框。

指定完路径和文件名后, 单击 [Next] 以显示 < Subcommand > 对话框。

图 8-5 < Library File Name > 对话框

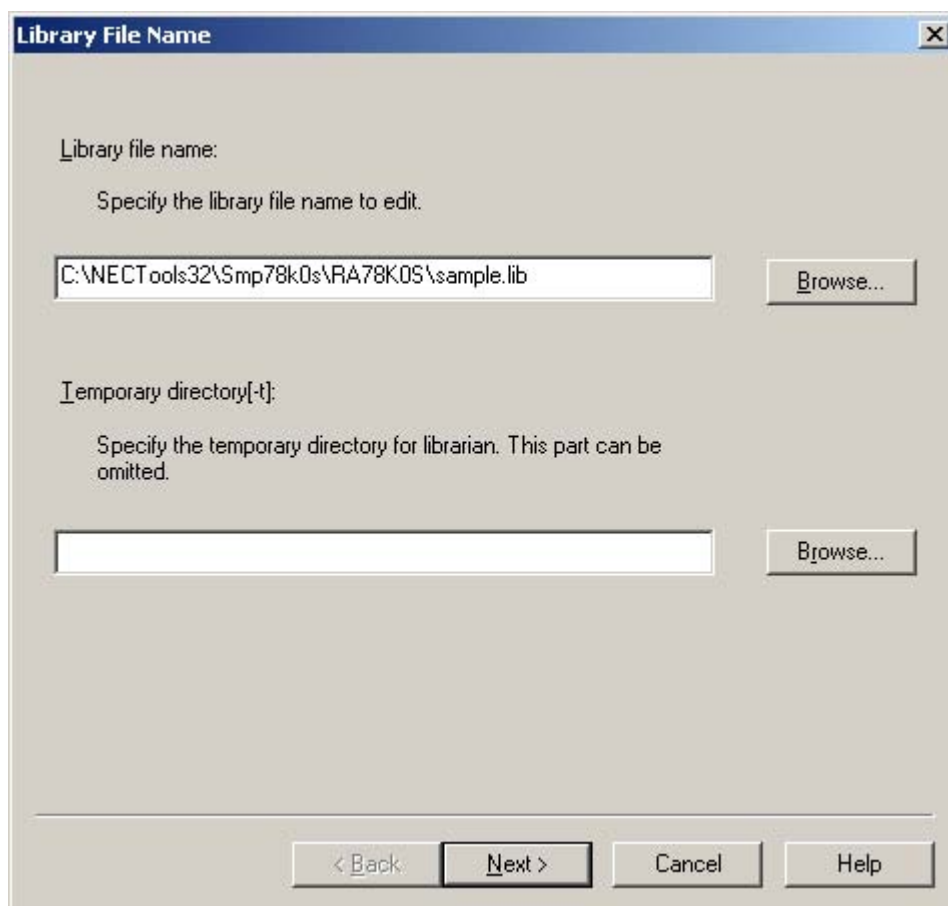
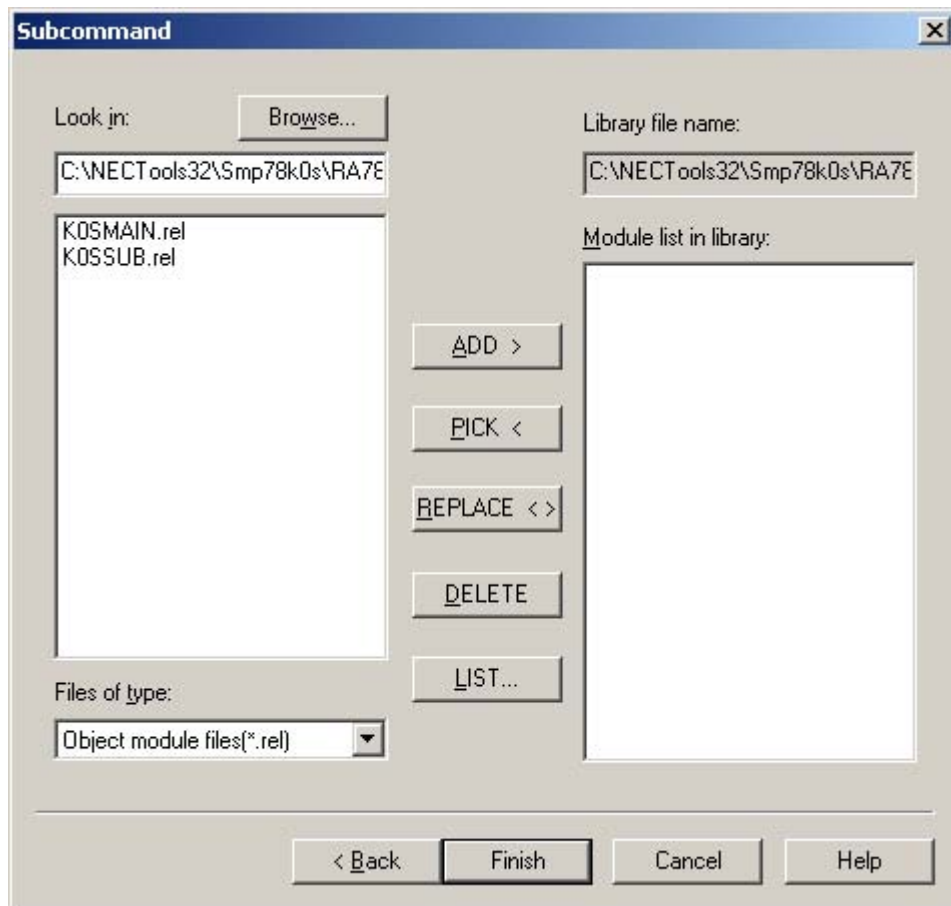


图 8-6 < Subcommand > 对话框



8.7.2 项目设置

< Subcommand > 对话框中的各个项目描述如下。

< Library File Name > 对话框

- **L**ibrary file name

通过使用 [**B**rowse] 按钮或直接输入文件名来指定库文件的名称。

- **T**emporary directory [-t]

使用 [**B**rowse] 按钮或直接输入一个路径名来指定临时文件要创建的路径。

< Subcommand > 对话框

- **L**ook in

通过使用 [**B**rowse] 按钮或者直接输入一个路径，指定要用作库的目标模块文件存在的路径。

如果指定了路径，则显示一个文件列表。

- **F**iles of type

指定在文件列表中显示的文件类型。

- **L**ibrary file name

这是只读编辑框。它显示当前指定的库文件名。

- **M**odule list in library

显示当前在库中指定的目标模块文件的列表。

- **A**DD >

向现存的库文件中添加一个模块。

- **P**ICK <

从现存的库文件中拾取一个指定的模块。

- **R**EPLACE < >

用其它目标模块文件的模块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

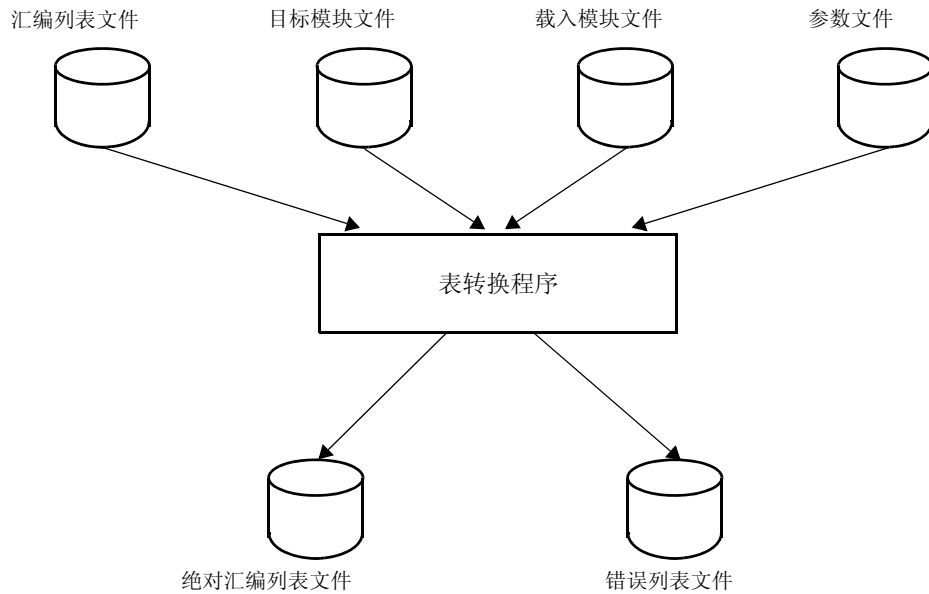
输出库文件中模块数据。

第九章 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链接器输出的载入文件。

表转换程序将实际地址嵌入可重定位地址中，将符号嵌进输入文件中，并输出绝对汇编列表。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。

图 9-1 表转换程序的 I/O 文件



9.1 表转换程序的 I/O 文件

表转换程序的 I/O 文件如下所示。

表 9-1 表转换程序的 I/O 文件

类型	文件名	说明	默认文件类型
输入文件	目标模块文件	- 这些二进制文件包括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	- 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	- 这些二进制映像文件包含作为链接结果的目标代码。	.lmf
	参数文件	- 这些文件包含用于执行程序的参数。 - 这些文件由用户创建。	.plv
输出文件	绝对汇编列表文件	- 该列表文件将实际地址嵌入可重新定位地址中，将符号嵌进输入文件中。	.p
	错误列表文件	- 这些文件包含了一些在列表转换期间生成的错误数据。	.elv

9.2 表转换程序的功能

表转换程序解决了绝对汇编器和可重置汇编器的缺点。

下面比较了可重新定位汇编程序与绝对汇编程序的优缺点。

[优点]

- (1) 可重新定位汇编程序可由多个成员共同开发。
- (2) 可重新定位汇编程序可以分成模块以易于开发和存储。
- (3) 可重置汇编器支持库管理。
- (4) 可重新定位汇编器适合于开发大规模程序。

[缺点]

- (1) 可重新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) 可重新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，还必需参考链接映像。
- (3) 汇编列表中的可重新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。表转换程序解决了可重置汇编程序的缺点。

- (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) 可重新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。

嵌入在绝对汇编列表文件中的定位和目标代码可通过例 1 和例 2 的表转换程序获得。

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 寄存器中设置十六进制 2 位代码数据
28	28				
29	29	0006	R220000	CALL	!CONVAH ; ASCII <- HEX 转换
30	30				; 将 ASCII 码输出到 BC 寄存器
31	31	0009	F821FE	MOVW	DE , #STASC ; 设置存储 ASCII 码表至 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 寄存器中设置十六进制 2 位代码数据
28	28				
29	29	0086	R229500	CALL	!CONVAH ; ASCII<- HEX 转换
30	30				; 将 ASCII 输出至 BC 寄存器
31	31	0089	F821FE	MOVW	DE , #STASC ; 设置存储 ASCII 码表至 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

例 2 目标代码的嵌入

< 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 寄存器中设置十六进制 2 位代码数据
28 28
29 29 0006 R220000 CALL !CONVAH ; ASCII <- HEX 转换
30 30 ; 将 ASCII 输出至 BC 寄存器
31 31 0009 F821FE MOVW DE , #STASC ; 设置存储 ASCII 码表至 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 寄存器中设置十六进制 2 位代码数据
28 28
29 29 0086 R229500 CALL !CONVAH ; ASCII <- HE 转换
30 30 ; 将 ASCII 输出至 BC 寄存器
31 31 0089 F821FE MOVW DE , #STASC ; 设置存储 ASCII 码表至 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

可以用以下两种方法启动表转换程序。

(1) 由命令行启动

X>lc78k0s	[Δ 选项]	...	输入文件名	[Δ 选项]	... [Δ]
(a)	(b)		(d)	(c)	

(a) 当前驱动器的名称

(b) 表转换程序的命令文件名

(c) 输入表转换程序操作的详细指令

当指定两个和或两个以上的表转换程序选项时，用空格隔开各选项。在表转换程序选项中不区分大小写字母。关于表转换程序选项的详细说明，参见 "9.4 表转换程序选项"。

将包含空格的路径放入引号 (" ") 内。

(d) 汇编列表的基本名

指定包含空格的路径的文件放入引号 (" ") 内。

使用扩展名 .prn。

如果命令行中仅指定汇编列表的基本名，则目标模块文件和载入模块文件的基本名必须与汇编列表文件的基本名完全相同。

文件类型必须为如下所示：

表 9-2 启动表转换程序时指定的文件类型

文件名	类型
目标模块类型	.rel
载入模块类型	.lmf

例 C>lc78k0s k0smain -lk0s.lmf

当指定与基本名不同的文件时请使用选项进行设置。

(2) 由参数文件启动

当表转换程序所必需的数据与命令行不符时，或者每次执行表转换都重复指定同一表转换选项时，使用参数文件。

从参数文件启动表转换程序，在命令行指定参数文件选项 (-f)。

按如下所示，从参数文件中启动表转换程序。

C>lc78k0s	[Δ 输入文件名]	Δ	-f	参数文件名
			(a)	(b)

(a) 指定参数文件选项

(b) 包含表转换程序所必需的数据的文件

备注 使用编辑器创建参数文件。

书写参数文件内容的规则如下所示：

[[[Δ] 选项 [Δ 选项] ... [Δ] Δ]] ...

- 如果从命令行中省略输入文件名，则在参数文件中只能指定 1 个输入文件名。
- 还可以在设置完选项后书写输入文件名。
- 在参数文件中写入所有的表转换程序选项，并输出在命令行中指定的文件名。

例 使用编辑器创建参数文件 (k0s.plv)。

< k0s.plv 的内容 >

<pre>; 参数文件 k0smain -lk0s.lmf -ek0s.elv</pre>

用参数文件 (k0s.plv) 启动表转换程序。

C>lc78k0s -fk0s.plv

9.3.2 执行开始和结束的信息

(1) 执行开始信息

当启动表转换程序时，屏幕上出现执行启动的信息。

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

Pass1 : start ...
Pass2 : start ...
```

(2) 执行结束信息

如果没有检测到由表转换造成的表转换错误，表转换程序在屏幕上输出以下信息，控制权返回给主机操作系统。

```
Conversion complete.
```

如果在表转换期间表转换程序检测到致命错误，使其无法继续转换处理，则表转换程序会将一条信息输出到显示屏上，取消表转换，将控制权还给操作系统。

例 指定不存在表转换程序选项。

```
C>lc78k0s k0smain.prn -a

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

RA78K0S error F6018 : Option is not recognized '-a'
Program aborted.
```

当表转换程序输出错误且表转换异常中止时，在“第十二章 错误信息”中查找原因，采取相应措施。

9.4 表转换程序选项

9.4.1 表转换程序选项类型

表转换程序选项是表转换程序操作的详细指令。表转换程序选项分为 6 种类型。

表 9-3 表转换程序选项

类型	选项	说明
目标模块文件输入指定	-r	输入目标模块文件。
载入模块文件输入指定	-l	输入载入模块文件。
绝对汇编列表文件输出指定	-o	指定输出绝对汇编列表文件。
错误列表文件输出指定	-e	输出一个错误列表文件。
参数文件指定	-f	从指定文件中输入文件名和选项。
帮助指定	--	在屏幕上输出帮助信息。

9.4.2 表转换程序选项的说明

以下为各表转换程序选项的详细描述。

(1) 目标模块文件输入指定

目标模块文件输入指定 (-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 指定目标模块文件的输入。

[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，指定 -r 选项。

[说明]

- 如果有致命错误发生，不能输出绝对汇编列表文件。
- 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。

[使用举例]

- 汇编列表文件名为 k0smain.prn, 目标模块文件名为 sample.rel, 载入模块文件名为 k0s.lmf.

```
C>lc78k0s k0smain.prn -rsample.rel -lk0s.lmf
```

(2) 载入模块文件输入指定

载入模块文件输入指定 (-I)

[语法]

-I [输入文件名]

- 默认预设

-I 汇编列表文件名 .lmf

[功能]

- 选项 -I 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，指定 -I 选项。

[说明]

- 如果有致命错误发生，不能输出绝对汇编列表文件。
- 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。

[使用举例]

- 汇编列表文件名为 k0smain.prn 而载入模块文件名为 sample.rel。

```
C>lc78k0s k0smain.prn -lsample.lmf
```

(3) 绝对汇编列表文件输出指定

绝对汇编列表文件输出指定 (-o)

[语法]

<code>-o [输出文件名]</code>

- 默认预设

`-o` 汇编列表文件名 .p

[功能]

- 选项 `-o` 指定绝对汇编列表文件的输出。选项 `-o` 还指定输出目标和输出文件名。

[应用]

- 用选项 `-o` 用于改变绝对汇编列表文件的输出目标和输出文件名。

[说明]

- 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`, 则会产生异常错误。
- 如果同一个设备既指定为文件名又指定为错误文件名, 则会发生异常错误。
- 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变为 "汇编列表文件名 .p".
- 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 ".p" 并输出该文件。
- 当指定 `-o` 选项时省略驱动器的名称, 则绝对汇编列表文件会输出到当前驱动器。

[使用举例]

- 创建绝对汇编列表文件 (sample.p).

```
C>lc78k0s k0smain.prn -osample.p -lk0s.lmf
```

(4) 错误列表文件输出指定

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]

- 选项 **-e** 指定错误列表文件的输出。该选项还指定输出目标和输出文件名。
- 选项 **-ne** 使得 **-e** 不可用。

[应用]

- 指定选项 **-e** 将错误信息保存到文件中。

[说明]

- 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，如果指定设备型文件名 **CLOCK**，就会导致异常错误。
- 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，则会发生异常错误。
- 如果指定选项 **-e** 且省略输出文件名，错误列表文件名将是 "汇编列表文件名.elv"。
- 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。
- 当指定 **-e** 选项时省略驱动器的名称，则错误列表文件会输出到当前驱动器。
- 如果同时指定了 **-e** 和 **-ne** 选项，那么最后指定的选项优先级较高。

[使用举例]

- 创建错误列表文件 (sample.elv)。

```
C>lc78k0s k0smain.prn -esample.elv
```

引用错误列表文件 (sample.elv)。

```
RA78K0S warning W6701: Load module file is older than object module file
'K0SMAIN.LMF', K0SMAIN.REL'
Pass1: start
RA78K0S error F6105: Segment name is not found is load module file 'DATA'
```

(5) 参数文件指定

参数文件指定 (-f)

[语法]

-f 文件名

- 默认预设

没有输入文件

[功能]

- 选项 -f 指定选项的输入和来自指定文件的输入文件名。

[应用]

- 当启动表转换程序所需的数据不适合命令行使用时，指定选项 -f。
- 如果希望每次执行表转换程序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]

- 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，就会导致异常错误。
- 若省略文件名，则会导致异常错误。
- 如果只指定文件名的基本名，则表转换程序将指定该文件类型为 ".plv" 并打开该文件。
- 不允许参数文件的嵌套。如果在参数文件内指定选项 -f, 会产生异常错误。
- 在参数文件中可写入的字符数没有限制。
- 使用空格、制表符或者换行符 (LF) 来分隔选项或输入文件名。
- 命令行中，在为参数文件指定的位置上，参数文件中的选项和输入文件名会进行扩展。
- 最后指定的扩展选项优先级较高。
- 如果两次或更多次指定选项 -f, 会产生异常异常错误。
- ";" 或 "#" 之后的字符以及换行码 (LF) 或 "EOF" 之前的字符均当做注释。

[使用举例]

- 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
```

在命令行键入以下内容.

```
C>lc78k0s -fk0s.plv
```

(6) 帮助指定

帮助指定 (--)

[语法]

```
--
```

- 默认预设

不显示

[功能]

- 选项 -- 在屏幕上输出帮助信息。

[应用]

- 帮助信息就是表转换程序选项的说明列表。在执行表转换程序时参考这些内容。

[说明]

- 当指定选项 -- 时,所有其它选项均不可用。

注意事项 不能从 PM+ 中指定该选项。

若要参考 PM+ 帮助,单击 < List Converter Options > 对话框中的 [Help] 按钮。

[使用举例]

- 指定选项 -- 时,帮助信息输出到显示屏。

```
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 ).
-Ffile      : Input option or input-file name from specified file.
-E [ file ] : Create error list file.
--          : Show this message.
```

9.5 PM+ 中选项设置

本节介绍 PM+ 表转换程序选项设置的方法。

9.5.1 选项设置方法

从 PM+ 的 [Tools] 菜单选择 [List converter Options] 或点击 [LC], 显示 < List Converter Options > 对话框。
可通过在对话框中输入所需的选项来设置表转换程序选项。

图 9-2 < List Converter Options > 对话框（选择 << Output >> 标签时）

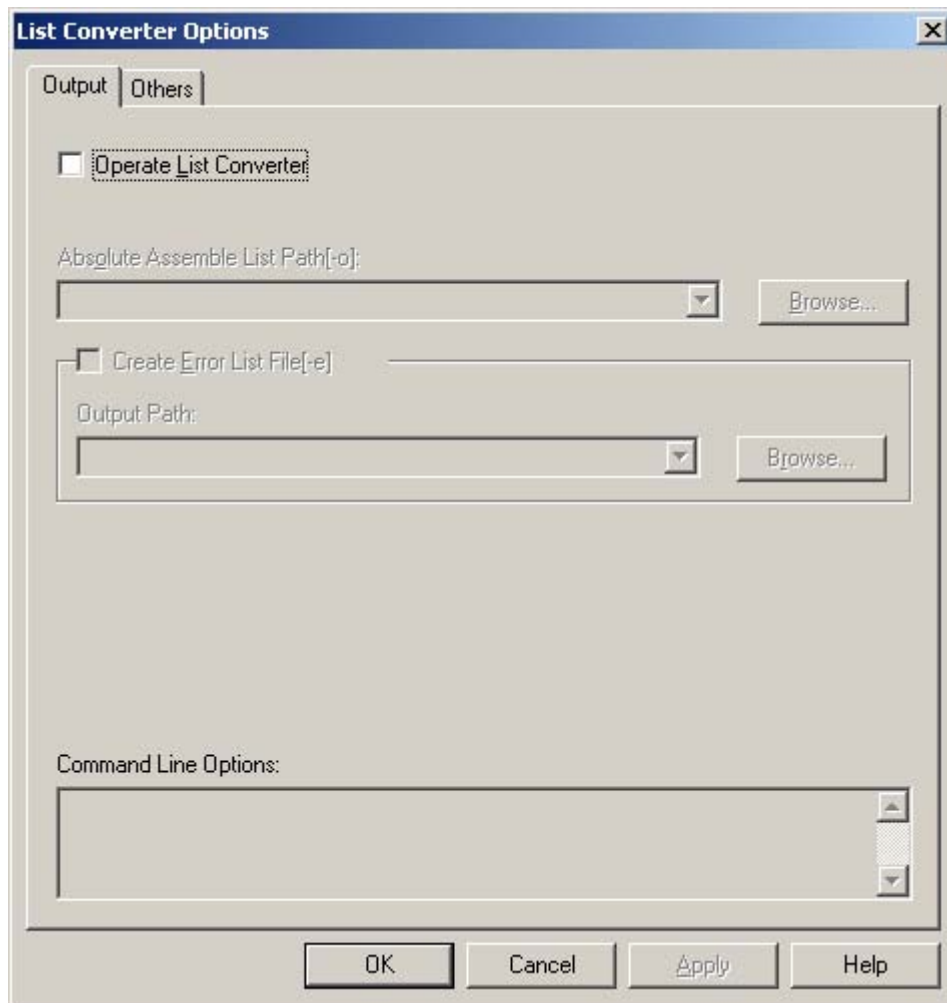
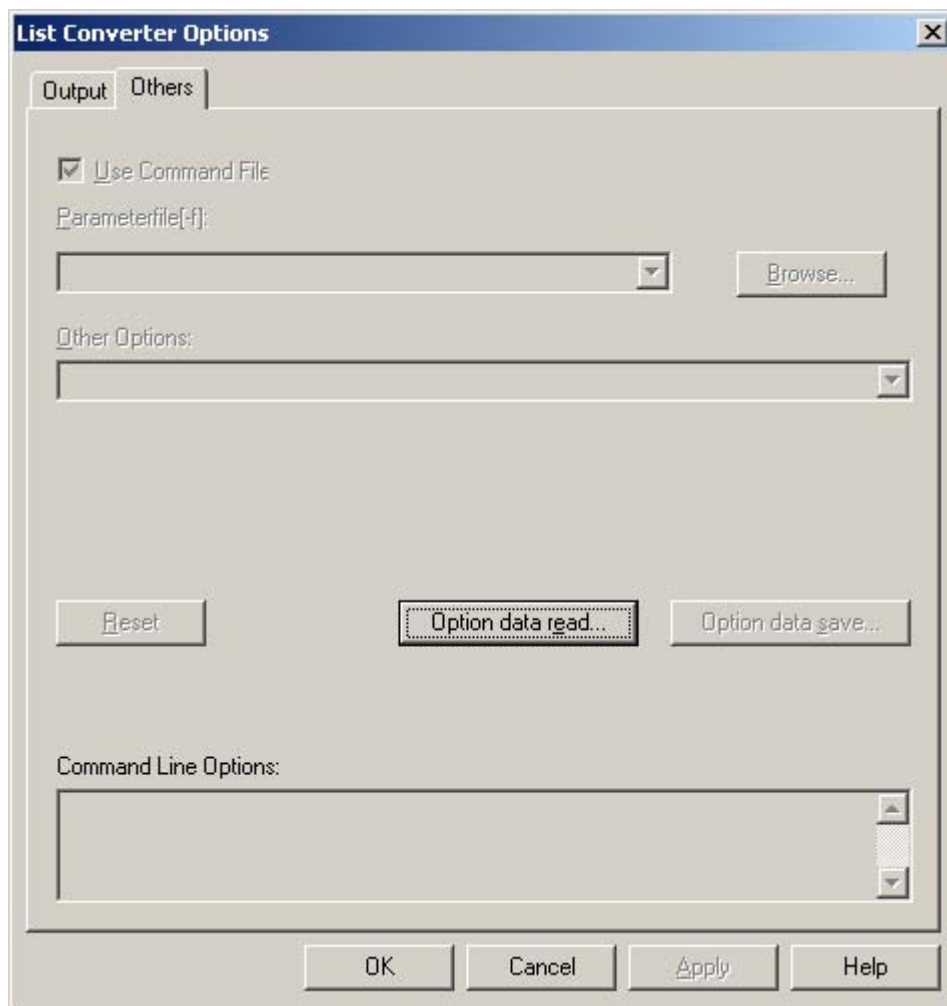


图 9-3 < List Converter Options > 对话框（选择 << Others >> 标签时）



9.5.2 选项设置

< List Converter Options > 对话框中的各选项描述如下。

(1) << Output >> 标签

- Operate List Converter
选择该项以启动表转换程序。
- Abs^lolute 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]
选取该选项以输出错误列表文件。
- Output Path
通过使用 [Browse] 按钮指定错误列表文件路径或直接输入。
- Command Line Options
是只读编辑框。显示当前设置选项字符串。

(2) << Others >> 标签

- Use Command File
选取该项创建命令文件。
- Parameterfile [-f]
通过 [Browse] 按钮或直接输入，将输入文件指定为用户定义参数文件。
- Other Options
若要指定对话框中选项外的其它选项，可在输入框中输入该选项。

注意事项 帮助指定 (--) 选项不能在 PM+ 上指定。
- Reset
复位输入内容。
- Option data read
打开 < Read Option Data > 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 < Save Option Data > 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
是只读编辑框。显示当前设置选项字符串。

第十章 程序输出列表

以下是关于各程序输出列表的格式说明及一些其它信息。

- 结构化汇编器预处理器输出列表

- 错误列表

- 汇编器输出列表

- 汇编列表文件头

- 汇编列表

- 符号列表

- 交叉引用列表

- 错误列表

- 链接器输出列表

- 链接列表文件标题

- 映射列表

- 公共符号列表

- 局部符号列表

- 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

- 表转换程序输出列表

- 绝对汇编列表

- 错误列表

10.1 结构化汇编器预处理器输出列表

结构化汇编器预处理器输出以下列表。

表 10-1 结构化汇编器预处理器输出列表

输出列表文件名	输出列表名
错误列表文件	错误列表

10.1.1 错误列表

当结构化汇编器预处理器启动时输出存储错误信息的错误列表。

[输出格式]

Start

(1) TTT.S((2) 4) : RA78K0S (3) error (4) E1221: (5) Missing ENDIF

[输出项的说明]

表 10-2 错误列表输出项的说明（当结构化汇编器预处理器启动时 p）

项	详细说明
(1)	发生错误的源模块文件的名称
(2)	发生错误的行
(3)	错误类型
(4)	错误编号
(5)	错误信息

注意事项 发生错误的文件名和行可能不被显示。

10.2 汇编器输出列表

汇编器输出以下列表。

表 10-3 汇编器输出列表

输出列表文件名	输出列表名
汇编列表文件	汇编列表
	符号列表
	交叉引用列表
错误列表文件	错误列表

10.2.1 汇编列表文件头

文件头总是在汇编列表文件的起始处输出。

[输出格式]

```
78K/OS Series Assembler (1)Vx.xx (2)SAMPLE_TITLE Date: (3)xx xxx xxxx Page: (4)1
(5)SAMPLE_SUBTITLE
Command: (6) k0smain.asm -c9024
Para-file: (7) -ks -kx
In-fine: (8) KOSMAIN.ASM
Obj-file: (9) KOSMAIN.REL
Prn-file: (10) KOSMAIN.PRN
```

[输出项的说明]

表 10-4 汇编列表文件头输出项的说明

项	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 指定的字符串
(3)	汇编列表创建的日期
(4)	页码号
(5)	副标题字符串 由控制指令 SUBTITLE 指定的字符串
(6)	命令行映射
(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

10.2.2 汇编列表

汇编列表输出带错误信息（如果发生错误）的汇编结果

[输出格式]

```

      Assemble list

(1)ALNO (2)STNO (6)ADRS (8)OBJECT (3)M (4)I (5)SOURCE STATEMENT
      1      1
      2      2                      NAME      SAMPM
      :
      28     28
      29     29     0006     R220000          CALL  !CONVAH
                                   ; ASCII <- HEX 转换
      30     30                                   ; 将 ASCII 码输出到 BC 寄存器
      31     31     0009     00000000        MOV    DE , #STASC
                                   ; 设置存储 ASCII 代码表到 DE
(7) ** ERROR E2202 , STNO 31 ( 0 ) Illegal operand
              000D      00
      32     32     000E     0A27          MOV    A , B
      33     33     0010     EB          MOV    [ DE ] , A
      :
Segment informations :

(9)ADRS (10)LEN  (11) NAME

      FE20    0003H  DATA
      0000    0002H  CODE
      0000    0017H  ?CSEG

Target chip : (12) uPD78xxx
Device file : (13) Vx.xx
Assembly complete , (14)1 error ( s ) and (15)0 warning ( s ) found. ( (16)31 )
```

[输出项的说明]

表 10-5 汇编列表输出项的说明

项	详细说明
(1)	源模块映射的行号
(2)	行号 (包括 INCLUDE 文件和宏的扩展)
(3)	宏显示 M : 为宏定义行 #n : 为宏扩展行 .n 为嵌套级数 . Blank : 不是宏定义或宏扩展行
(4)	INCLUDE 显示 In : 在 INCLUDE 文件中 . n 为嵌套级数 . Blank : I 不使用 INCLUDE 文件 .
(5)	源程序语句
(6)	定位计数器的值 (4 位)
(7)	错误发生行
(8)	重定位数据 R : 目标代码或符号值被链接器改变 . Blank : 目标代码或符号值不被链接器改变
(9)	段地址 (4 位或 5 位)
(10)	段长度 (4 位或 5 位)
(11)	段名
(12)	RA78K0S 目标设备
(13)	设备文件版本数目
(14)	致命错误数目
(15)	报警数目
(16)	最后出错行

10.2.3 符号列表

符号列表输出源模块中定义的符号（包括局部符号）。

[输出格式]

Symbol Table List							
(1)VALUE	(2)ATTR	(3)RTYP	(4)NAME	(1)VALUE	(2)ATTR	(3)RTYP	(4)NAME
	CSEG		?CSEG		CSEG		CODE
----H		EXT	CONVAH		DSEG		DATA
FE20H	ADDR		HDTSA	0H	ADDR	PUB	MAIN
	MOD		SAMPM	0H	ADDR	PUB	START
FE21H	ADDR		STASC				

[输出项的说明]

表 10-6 符号列表输出项的说明

项	详细说明
(1)	符号值（4 位）
(2)	符号属性 CSEG：代码段名 DSEG：数据段名 BSEG：位段名 MAC：宏名 MOD：模块名 SET：由 SET 指令定义的符号 NUM：NUMBER 属性符号 ADDR：ADDRESS 属性符号 BIT：BIT 属性符号 (addr.bit) 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, PSW.bit) Blank：由 EXTRN 或 EXTBIT 声明的外部引用符号 *****：未定义的符号
(3)	符号引用格式 EXT：由 EXTRN 声明的外部引用符号（SADDR 属性） EXTB：由 EXTBIT 声明的外部引用符号 (saddr.bit) PUB：由 PUBLIC 声明的外部引用符号 Blank：局部符号，段名，宏名，模块名 *****：未定义的符号
(4)	定义的符号名

10.2.4 交叉引用列表

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]

Cross-Reference List									
(1)NAME	(2)VALUE	(3)R	(4)ATTR	(5)RTYP	(6)SEGNAME	(7)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		
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		

[输出项的说明]

表 10-7 交叉引用输出项的说明

项	详细说明	
(1)	定义的符号名	
(2)	符号值 (4 位或 5 位)	
(3)	可重置属性 R : 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 DSEG : 数据段名 BSEG : 位段名 MAC : 宏名 MOD : 模块名 SET : 由 SET 指令定义的符号 NUM : NUMBER 属性符号	ADDR : ADDRESS 属性符号 BIT : BIT 属性符号 (addr.bit) 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, PSW.bit) Blank : 由 EXTRN 或 EXTBIT 声明的外部引用符号 ***** : 未定义的符号
(5)	符号引用格式 EXT : 由 EXTRN 声明的外部引用符号 (SADDR 属性) EXTB : 由 EXTBIT 声明的外部引用符号 (saddr.bit) PUB : 由 PUBLIC 声明的外部引用符号 Blank : 局部符号, 段名, 宏名, 模块名 ***** : v	
(6)	定义的符号名	
(7)	定义 / 引用的行号 定义行 : xxxxx# 引用行 : xxxxx D (D = 1 个空格) EXTRN 声明, EXTBIT 声明, PUBLIC 声明 : xxxxx@	

10.2.5 错误列表

当汇编器启动时输出存储错误信息的错误列表。

[输出格式]

```
PASS_PARSE Start
(1)ERROR. ASM ( (2)26 ) : RA78K0S (3)error (4)E2202 : (5)Illegal operand
(1)ERROR. ASM ( (2)32 ) : RA78K0S (3)error (4)E2202 : (5)Illegal operand PASS_OUTOBJ Start
(1)ERROR. ASM ( (2)26 ) : RA78K0S (3)error (4)E2202 : (5)Illegal operand
(1)ERROR. ASM ( (2)29 ) : RA78K0S (3)error (4)E2407 : (5)Undefined symbol reference 'DTSA'
(1)ERROR. ASM ( (2)29 ) : RA78K0S (3)error (4)E2303 : (5)Illegal expression
(1)ERROR. ASM ( (2)32 ) : RA78K0S (3)error (4)E2202 : (5)Illegal operand
(1)ERROR. ASM ( (2)37 ) : RA78K0S (3)error (4)E2407 : (5)Undefined symbol reference 'F'
(1)ERROR. ASM ( (2)37 ) : RA78K0S (3)error (4)E2303 : (5)Illegal expression
```

[输出项的说明]

表 10-8 错误输出项的说明（当汇编器启动时）

项	详细说明
(1)	发生错误的源模块名
(2)	发生错误的行
(3)	错误类型
(4)	错误编号
(5)	错误信息

注意事项 发生错误的文件名和行可能不被显示。

10.3 链接器输出列表

链接器输出以下列表。

表 10-9 链接器输出列表

输出列表文件名	输出列表名
链接列表文件	映射列表
	公共符号列表
	局部符号列表

10.3.1 链接列表文件头

文件头总是在链接列表文件的起始处输出。

[输出格式]

```
78K/0S 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.MAP
Map-File :          (7) K0SMAIN.MAP
Direc-File :        (8)
Directive :         (9)

*** Link information ***
(10) 3 output segment ( s )
(11) 37H byte ( s ) real data
(12) 23 symbol ( s ) defined
```

[输出项的说明]

表 10-10 链接列表文件头输出项的说明

项	详细说明
(1)	链接器版本编号
(2)	链接列表文件创建时间
(3)	页码号 (4 位或 5 位)
(4)	命令行映射 (4 位或 5 位)
(5)	参数文件的内容
(6)	输出载入模块文件名
(7)	链接列表文件名
(8)	指令文件名
(9)	指令文件内容 (4 位或 5 位)
(10)	输出至载入模块文件的段的数目 (4 位或 5 位)
(11)	输出至载入模块文件的数据的长度
(12)	输出至载入模块文件的符号的数目

10.3.2 映射列表

映射列表输出段存储数据。

[输出格式]

```

*** Memory map ***

(1)SPACE = REGULAR

    MEMORY = (2)ROM
    BASE ADDRESS = (3)0000H    SIZE = (4)2000H
      (6)OUTPUT (7)INPUT (8)INPUT (9)BASE (10)SIZE
        SEGMENT  SEGMENT  MODULE  ADDRESS
        CODE
                                0000H    0002H
                                (11)CSEG AT
                                CODE      SAMPM    0000H    0002H
(5)*gap *                                0002H    007EH
                                ?CSEG      0080H    0035H
                                (11)CSEG
                                ?CSEG      SAMPM    0080H    0015H
                                ?CSEG      SAMPS    0095H    0020H
(5)*gap *                                00B5H    1F4BH

    MEMORY = RAM
    BASE ADDRESS = (3)FE00H    SIZE = (4)0200H
      (6)OUTPUT (7)INPUT (8)INPUT (9)BASE (10)SIZE
        SEGMENT  SEGMENT  MODULE  ADDRESS
(5)* gap *
        DATA
                                FE00H    0020H
                                FE20H    0003H
                                (11)DSEG AT
                                DATA      SAMPM    FE20H    0003H
(5)*gap *                                FE23H    00DDH
(5)*gap ( Not Free Area ) *            FE00H    0100H

Target chip : (12)uPD78xxx
Device File : (13)Vx.xx

```

[输出项的说明]

表 10-11 映射列表输出项的说明

项	详细说明
(1)	内存空间名
(2)	内存区域名
(3)	内存区域开始地址 (4 位)
(4)	内存区域长度 (4 位)
(5)	输出组 显示 "gap" 表示该区域没有任何存储内容 .
(6)	输出至载入模块文件的段名
(7)	从目标模块文件读取的段名
(8)	输入模块名
(9)	段开始地址 (4 位)
(10)	输出 / 输入段的长度 (4 位)
(11)	段类型和重置属性
(12)	用于本次汇编的目标设备
(13)	设备文件版本编号

10.3.4 局部符号列表

局部符号列表输出在输入模块内定义的局部符号数据。

[输出格式]

*** Local symbol list ***			
(1)MODULE	(2)ATTR	(3)VALUE	(4)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

[输出项的说明]

表 10-13 局部符号列表输出项的说明

项	详细说明	
(1)	定义局部符号的模块名	
(2)	符号属性 CSEG： 代码段名 DSEG： 数据段名 BSEG： 位段名 MAC： 宏名 MOD： 模块名 SET： 由 SET 指令定义的符号 NUM： NUMBER 属性符号	ADDR： ADDRESS 属性符号 BIT： BIT 属性符号 (addr.bit) 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, PSW.bit) Blank： 由 EXTRN 或 EXTBIT 声明的外部引用符号 *****： 未定义的符号
(3)	符号值 (4 位)	
(4)	局部符号名	

10.3.5 错误列表

当链接器启动时输出存储错误信息的错误列表。

[输出格式]

RA78K0S (1) error (2) E3405 : (3) Undefined symbol 'CONVAH' in file 'K0SMAIN.REL'

[输出项的说明]

表 10-14 错误列表输出项的说明（链接器启动时）

项	详细说明
(1)	错误类型
(2)	错误编号
(3)	错误信息

10.4 目标转换器输出列表

目标转换器输出以下列表。

表 10-15 目标转换器输出项的说明

输出列表文件名	输出列表名
错误列表文件	错误列表

10.4.1 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]

与链接器输出的错误列表格式相同。

10.5 库管理程序输出列表

库管理程序输出以下列表。

表 10-16 库管理程序输出列表

输出列表文件名	输出列表名
列表文件	库数据输出列表

10.5.1 库数据输出列表

库数据输出列表输出库文件中的模块数据。

[输出格式]

78K/0S Series librarian (1)Vx.xx	DATE : (2)xx xxx xx	PAGE (3)1
LIB-FILE NAME : (4)K0S.LIB	((5)xx xxx xx)	
(6)0001 (7)K0SMAIN.REL	((8)xx xxx xx)	
(9)MAIN	(9)START	
NUMBER OF PUBLIC SYMBOLS : (10)2		
(6)0002 (7)K0SSUB.REL	((8)xx xxx xx)	
(9)CONVAH		
NUMBER OF PUBLIC SYMBOLS : (10)1		

[输出项的说明]

表 10-17 库数据输出列表输出项的说明

项	详细说明
(1)	库管理程序版本编号
(2)	创建列表的日期
(3)	页码数
(4)	库文件名
(5)	库文件创建的日期
(6)	模块序列号 (从 0001 开始)
(7)	模块名
(8)	模块创建日期
(9)	公共符号名
(10)	模块中定义的公共符号的数目

10.6 列表转换器输出列表

列表转换器输出以下列表。

表 10-18 列表转换器输出列表

输出列表文件名	输出列表名
绝对汇编列表文件	绝对汇编列表
错误列表文件	错误列表

10.6.1 绝对汇编列表

绝对汇编列表将绝对值嵌入在汇编列表中并输出该列表。

[输出格式]

与汇编器输出的汇编列表格式相同。

10.6.2 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。

第十一章 提高使用 RA78K0S 效率

本章描述如何提高使用 RA78K0S 效率。

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
- WARNING 发生：0
- FATAL ERROR 发生：1
- ABORT：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。

[使用举例]

< 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 (适用于结构化汇编器预处理器和汇编器)
LIB78K0S :	搜索库文件路径 (仅适用于链接器)
TMP :	用于创建临时文件的路径
LANG78K :	Kanji (2 字节字符) 类型指定

[使用举例]

< 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) 链接器从目录 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。
- (5) 注释语句中的 Kanji 被解释成 shift JIS 码。

注意事项 如果使用 Windows 安装器安装汇编器软件包, 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从键盘键入 **CTRL-C** 进行中断。

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，所有打开的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，同样可以显示副标题。需要了解控制指令的详细信息，参见 RA 78K0S 汇编器软件包语言用户手册。

[使用举例]

打印汇编列表文件头中的标题。

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

引用 k0smain.prn.

```
78K/0S Series 78K0SAssembler Ex.xx RA78K0S_MAINROUTINE Date : xx xxx xxxx
Page : 1

                                |
                                Title

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
    3        3
    4        4
    5        5
    6        6
    7        7
    :

                                NAME SAMPM
                                ; *****
                                ; *
                                ; *   HEX -> ASCII 转换程序   *
                                ; *
                                ; *   主程序                       *
                                ; *
```

11.5 减少程序启动时间

11.5.1 在源程序中指定控制指令

通常在汇编器启动程序中指定的选项具有相同功能的控制指令，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。

[使用举例]

```
$ PROCESSOR ( 9024 )      ; 控制指令
$ XREF                   ; 控制指令

      NAME      SAMP.M
; *****
; *
; *          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。这些存储的选项要用于紧接着的编译 (MAKE) 操作。因此不必在每次启动程序时都指定选项。

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 (结构化汇编器预处理器, 汇编器, 链接器, 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, 或者在每次程序执行时都指定相同选项时, 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。这就使得目标模块库更容易形成。

[使用举例 1]

创建参数文件并执行汇编。

< 参数文件 k0smain.pra 的内容 >

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

从命令行中键入以下内容。

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

[使用举例 2]

创建参数文件并执行汇编。

< 参数文件 k0s.slb 的内容 >

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

从命令行中键入以下内容。

```
C>lb78k0s <k0s.slb
```

11.6 目标模块库生成

汇编器及链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。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。

库文件可以输入至链接器中。因此，当进行模块化编程时，可以创建包含通用模块的库文件，方便而有效地对程序进行管理和操作。

第十二章 错误消息

本章解释由 RA78K0S 程序 (结构化汇编器，汇编器，链接器，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 输出的错误信息可分为 4 个级别。

(1) 异常中止错误 (Fxxxx)

发生了使得程序不能继续执行的错误。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 (Exxxx)

发生了执行错误。如果发现另外一个错误，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) 内部错误 (Cxxxx)

由于发生了内部错误会立即中止处理 (异常中止)。

(4) 警告 (Wxxxx)

产生非用户预期的输出对象。

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。

汇编包的错误信息分类如下。

- Fn0xx --- 命令行分析错误 - Fn9xx --- 文件或系统错误 - Fn1xx --- 其它异常中止错误 - Cnxxx --- 内部错误 - En2xx --- 语句规范错误 - En3xx --- 表达式错误 - En4xx --- 符号错误 - En5xx --- 段错误 - En6xx --- 控制指令或宏错误 - Wnxxx --- 任何类型的警告错误	n = 1 至 6 - 1 --- 结构化汇编预处理程序 - 2 --- 汇编器 - 3 --- 链接器 - 4 --- 目标转换器 - 5 --- 库管理程序 - 6 --- 表转换程序
--	---

12.2 结构化汇编器预处理程序错误消息

表 12-1 结构化汇编器预处理程序错误消息

错误号	错误消息	
F1001	消息	Missing input file
	产生原因	未指定输入文件。
	用户应对措施	指定一个输入文件。
F1002	消息	Too many input files
	产生原因	指定了两个或两个以上的输入文件。
	用户应对措施	仅指定一个输入文件。
F1004	消息	Illegal file name 'file name'
	产生原因	在文件名中有非法字符或字符个数超过了限制范围。
	用户应对措施	输入具有合法字符且不超出限制范围的文件名。
F1005	消息	Illegal file specification 'file name'
	产生原因	指定了非法文件。
	用户应对措施	指定一个合法文件。
F1006	消息	File not found 'file name'
	产生原因	指定的文件不存在。
	用户应对措施	指定一个存在的文件。
F1008	消息	File specification conflicted 'file name'
	产生原因	重复指定了 I/O 文件名。
	用户应对措施	指定不同的 I/O 文件名。
F1009	消息	Unable to make file 'file name'
	产生原因	指定的文件为写保护。
	用户应对措施	取消指定文件的写保护。
F1010	消息	Directory not found 'file name'
	产生原因	在输出文件名中包括不存在的驱动器和 / 或目录。
	用户应对措施	指定一个存在的驱动器和 / 或目录。
F1011	消息	Illegal path 'option'
	产生原因	指定的路径与选项中指定的参数路径不同。
	用户应对措施	指定正确的路径名。
F1012	消息	Missing parameter 'option'
	产生原因	未指定必要的参数。
	用户应对措施	指定参数。

表 12-1 结构化汇编器预处理程序错误消息

错误号	错误消息	
F1013	消息	Parameter not needed 'option'
	产生原因	指定了不必要的参数。
	用户应对措施	删除不必要的参数。
F1014	消息	Out of range 'option'
	产生原因	指定的数值超出了范围。
	用户应对措施	指定正确的数值。
F1015	消息	Parameter is too long 'option'
	产生原因	参数中的字符数超出了限制范围。
	用户应对措施	指定一个字符数在限制内范围的参数。
F1016	消息	Illegal parameter 'option'
	产生原因	参数的语法不正确。
	用户应对措施	指定正确的参数。
F1017	消息	Too many parameters 'option'
	产生原因	参数的总数超出了限制范围。
	用户应对措施	指定限制范围内的参数。
F1018	消息	Option is not recognized 'option'
	产生原因	选项名称不正确。
	用户应对措施	指定正确的选项名。
F1019	消息	Parameter file nested
	产生原因	已在参数文件内指定了 -f 选项。
	用户应对措施	不要在参数文件内部指定 -f 选项。
F1020	消息	Parameter file read error 'file name'
	产生原因	不能读取参数文件。
	用户应对措施	指定正确的参数文件。
F1021	消息	Memory allocation failed
	产生原因	没有足够的存储空间。
	用户应对措施	确保必须的存储空间。
F1100	消息	Can't set Control-C
	产生原因	不能设置让结构化汇编预处理程序停止执行的控制 -C。
	用户应对措施	再次执行结构化汇编预处理程序。
F1101	消息	Open/read/write/close error on 'file name'
	产生原因	由于文件 I/O 错误，不能正常打开、写入或关闭文件。
	用户应对措施	指定正确的文件名。

表 12-1 结构化汇编器预处理程序错误消息

错误号	错误消息	
F1102	消息	Can't find 'file name'
F1103	消息	Illegal include file 'file name'
	产生原因	为包含文件指令了非法名称。
	用户应对措施	指定正确的文件。
F1104	消息	Illegal (-sc) character
	产生原因	在 -sc 选项中指定了不能用作符号的字符。
	用户应对措施	指定正确的字符。
F1105	消息	Can't define the reserved symbol
	产生原因	不要在 -d 选项中指定保留字。
	用户应对措施	不要在 -d 选项中指定保留字。
F1106	消息	Duplicate PROCESSOR control
	产生原因	在源文件在中多次指定 PROCESSOR 控制指令。 产品类型与 -c 选项中指定的类型不同。
	用户应对措施	仅指定 PROCESSOR 控制指令一次，更正产品类型名称。
F1107	消息	No processor specified
	产生原因	未指定设备类型。
	用户应对措施	指定设备类型。
F1108	消息	Illegal processor type specified
	产生原因	源文件中 PROCESSOR 控制指令中的设备类型指定不正确。
	用户应对措施	指定正确的设备类型。
F1109	消息	Illegal processor type specified -C
	产生原因	-c 选项中指定的设备类型不正确。
	用户应对措施	指定正确的设备类型。
F1110	消息	Can't use this control outside module header
	产生原因	应写入源文件头部的控制指令却写入到了正常源文件行中。
	用户应对措施	在源模块头写入该指令。
F1111	消息	Syntax error in module header
	产生原因	写入源模块头的指令语法不正确。
	用户应对措施	使用正确的语法写入该指令。
C1112	消息	Structured assembler preprocessor internal error
	产生原因	在结构化汇编预处理程序内部出现错误。
	用户应对措施	联系日电电子公司或日电电子发行人。

表 12-1 结构化汇编器预处理程序错误消息

错误号	错误消息	
E1201	消息	Illegal #ELSE/#ENDIF
	产生原因	#ELSE 和 #ENDIF 语句写入了错误位置处。
	用户应对措施	在正确的位置写入 #ELSE 和 #ENDIF 语句。
E1202	消息	Illegal #ENDCALLT
	产生原因	#ENDCALLT 语句写入了错误位置处。
	用户应对措施	在正确的位置处写入 #ENDCALLT 语句。
E1203	消息	Missing #ENDIF
	产生原因	#ENDIF 语句不存在。
	用户应对措施	在正确的位置处写入 #ENDIF 语句。
E1204	消息	Missing #ENDCALLT
	产生原因	#ENDCALLT 语句不存在。
	用户应对措施	在正确的位置处写入 #ENDCALLT 语句。
E1205	消息	Too many #DEFCALLT definition
	产生原因	所注册的 callt 指令转换模式超过了限定范围。
	用户应对措施	减少所注册的 #defcallt 指令数目。
E1206	消息	Too many CALL instruction
	产生原因	由 #DEFCALLT 至 #ENDCALLT 定义的指令过多。
	用户应对措施	仅由 #DEFCALLT 至 #ENDCALLT 定义一条指令。
E1207	消息	Duplicate definition
	产生原因	再一次定义了相同的转换模式。
	用户应对措施	更正 #DEFCALLT 注册项。
E1208	消息	Symbol table overflow
	产生原因	符号数超出了限制范围。
	用户应对措施	减少符号个数。
E1209	消息	Syntax error
	产生原因	写入语句的语法不正确。
	用户应对措施	使用正确的语法。
E1210	消息	Nest level error
	产生原因	嵌套时存在错误 (溢出, 嵌套级数等等)
	用户应对措施	使用正确的控制语句。
E1211	消息	Too many characters in a line
	产生原因	一行内的字符长度超过了范围。
	用户应对措施	在一行内指定 2048 或少于 2048 个的字符。

表 12-1 结构化汇编器预处理程序错误消息

错误号	错误消息	
E1212	消息	Too many include files
	产生原因	在包含文件内存在包含伪指令。
	用户应对措施	不要在包含文件内指定包含伪指令。
E1214	消息	Illegal BREAK
	产生原因	BREAK 语句写入到了错误位置处。
	用户应对措施	在正确的位置处写入 BREAK 语句。
E1215	消息	Illegal CONTINUE
	产生原因	CONTINUE 语句写入到了错误位置处。
	用户应对措施	在正确的位置处写入 CONTINUE 语句。
E1216	消息	Illegal CASE/DEFAULT/ENDS
	产生原因	CASE/DEFAULT/ENDS 语句写入到了错误位置处。
	用户应对措施	在正确的位置写入 CASE/DEFAULT/ENDS 语句。
E1217	消息	Illegal ELSEIF/ELSE/ENDIF
	产生原因	ELSEIF/ELSE/ENDIF 语句写入了错误位置处。
	用户应对措施	在正确的位置写入 ELSEIF/ELSE/ENDIF 语句。
E1218	消息	Illegal NEXT
	产生原因	NEXT 语句写入到了错误位置处。
	用户应对措施	在正确的位置处写入 NEXT 语句。
E1219	消息	Illegal ENDW
	产生原因	ENDW 语句写入到了错误位置处。
	用户应对措施	在正确的位置处写入 ENDW 语句。
E1220	消息	Illegal UNTIL/UNTIL_BIT
	产生原因	UNTIL 和 UNTIL_BIT 语句写入了错误位置处。
	用户应对措施	在正确的位置写入 UNTIL 和 UNTIL_BIT 语句。
E1221	消息	Missing ENDIF
	产生原因	ENDIF 语句不存在。
	用户应对措施	在正确的位置写入 ENDIF 语句。
E1222	消息	Missing ENDS
	产生原因	ENDS 语句不存在。
	用户应对措施	在正确的位置写入 ENDS 语句。
E1223	消息	Missing ENDW
	产生原因	ENDW 语句不存在。
	用户应对措施	在正确的位置写入 ENDW 语句。

表 12-1 结构化汇编器预处理程序错误消息

错误号	错误消息	
E1224	消息	Missing NEXT
	产生原因	NEXT 语句不存在。
	用户应对措施	在正确的位置写入 NEXT 语句。
E1225	消息	Missing UNTIL/UNTIL_BIT
	产生原因	UNTIL 和 UNTIL_BIT 语句不存在。
	用户应对措施	在正确的位置写入 UNTIL 和 UNTIL_BIT 语句。
E1226	消息	Illegal character in a line
	产生原因	源程序行中写入了错误字符。
	用户应对措施	删除写入到源程序行中的错误字符。
E1227	消息	Illegal operand in a line
	产生原因	替换数据大小及比较条件格式不正确。
	用户应对措施	指定正确的数据大小。
E1228	消息	Illegal SFR access in operand
	产生原因	写入了无法访问替换格式的 sfr 符号。
	用户应对措施	检查 sfr 符号的访问状态并写入正确的 sfr 符号。
E1229	消息	This symbol is reserved 'symbol name'
	产生原因	所用的符号是保留字。
	用户应对措施	更改符号名。
E1230	消息	Out source line overflow
	产生原因	源文件行中的字符数超过了限制范围。
	用户应对措施	在源文件行中删除不必要的描述。
W1301	消息	Symbol redefinition
	产生原因	该符号由 #define 语句定义了多次。
	程序的措施	最近定义的符号有效。
	用户应对措施	要使首先定义的符号有效，需更正语法。
W1302	消息	Duplicate PROCESSOR option and control
	产生原因	由选项 -c 指定的设备类型与 \$PROCESSOR 控制命令中指定的设备类型不一样。
	程序的措施	由选项 -c 指定的设备类型有效，而忽略 \$PROCESSOR 控制命令中指定的设备类型。
	用户应对措施	检查在选项 -c 中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

错误号	错误消息	
F2001	消息	Missing input file
	产生原因	未指定输入文件。
	用户应对措施	指定一个输入文件。
F2002	消息	Too many input files
	产生原因	指定了两个或两个以上的输入文件。
	用户应对措施	仅指定一个输入文件。
F2004	消息	Illegal file name 'file name'
	产生原因	在文件名中有非法字符或字符个数超过了限制范围。
	用户应对措施	输入具有合法字符且不超出限制范围的文件名。
F2005	消息	Illegal file specification 'file name'
	产生原因	指定了非法文件。
	用户应对措施	指定一个合法文件。
F2006	消息	File not found 'file name'
	产生原因	指定的文件不存在。
	用户应对措施	指定一个存在的文件。
F2008	消息	File specification conflicted 'file name'
	产生原因	重复指定了 I/O 文件名。
	用户应对措施	指定不同的 I/O 文件名。
F2009	消息	Unable to make file 'file name'
	产生原因	指定的文件为写保护文件。
	用户应对措施	取消指定文件的写保护。
F2010	消息	Directory not found 'file name'
	产生原因	在输出文件名中包括不存在的驱动器和 / 或目录。
	用户应对措施	指定一个存在的驱动器和 / 或目录。
F2011	消息	Illegal path 'option'
	产生原因	指定的路径与选项中指定的参数路径不同。
	用户应对措施	指定正确的路径名。
F2012	消息	Missing parameter 'option'
	产生原因	未指定必要的参数。
	用户应对措施	指定参数。

表 12-2 汇编器错误消息

错误号	错误消息	
F2013	消息	Parameter not needed 'option'
	产生原因	指定了不必要的参数。
	用户应对措施	删除不必要的参数。
F2014	消息	Out of range 'option'
	产生原因	指定的数值超出了范围。
	用户应对措施	指定正确的数值。
F2015	消息	Parameter is too long 'option'
	产生原因	参数中的字符数超出了限制范围。
	用户应对措施	指定一个字符数在限制内范围的参数。
F2016	消息	Illegal parameter 'option'
	产生原因	参数的语法不正确。
	用户应对措施	指定正确的参数。
F2017	消息	Too many parameters 'option'
	产生原因	参数的总数超出了限制范围。
	用户应对措施	在数目限制范围内指定参数。
F2018	消息	Option is not recognized 'option'
	产生原因	选项名称不正确。
	用户应对措施	指定正确的选项名。
F2019	消息	Parameter file nested
	产生原因	已在参数文件内指定了 -f 选项。
	用户应对措施	不要在参数文件内部指定 -f 选项。
F2020	消息	Parameter file read error 'file name'
	产生原因	不能读取参数文件。
	用户应对措施	指定正确的参数文件。
F2021	消息	Memory allocation failed
	产生原因	没有足够的存储空间。
	用户应对措施	确保必须的存储空间。
F2101	消息	Source file size 0 'file-name'
	产生原因	输入了文件大小为 0 的源模块。
F2102	消息	Illegal processor type specified
	产生原因	在目标设备指定时产生了错误。
F2103	消息	Syntax error in module header
	产生原因	可以写入源模块头的控制指令格式产生错误。

表 12-2 汇编器错误消息

错误号	错误消息	
F2104	消息	Can't use this control outside module header
	产生原因	在源模块头指定的控制指令写进了普通源程序中去了。
F2105	消息	Duplicate PROCESSOR control
	产生原因	PROCESSOR 控制指令不止一次写进了源模块头。
F2106	消息	Illegal source file name for module name
	产生原因	模块名称无法创建，因为源文件名的基本名中包含有非法符号结构字符。
F2107	消息	Default segment ?CSEG is already used
	产生原因	试图用缺省段定义一个未定义段。
F2108	消息	Symbol table overflow 'symbol name'
	产生原因	符号数超出了可定义的范围。
F2109	消息	Too many DS
	产生原因	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。
F2110	消息	String table overflow
	产生原因	超出了字符串表的限制范围。
	用户应对措施	将符号数减少为 9 或更少。
F2111	消息	Object code more than 128bytes
	产生原因	在源程序语句中每行的目标代码超过了 128 个字节。
F2112	消息	No processor specified
	产生原因	未在命令行或源模块文件中指定目标设备。
F2114	消息	Local symbol name of asm statement must begin with '?L' in C source.
	产生原因	C 源程序中，用 #asm 语句描述的本地符号不是以 '?L' 开头。
F2115	消息	Too long source line
	产生原因	超出了行长度的限制范围 (2048 个字符)。
E2201	消息	Syntax error
	产生原因	使用了错误的语句格式。
E2202	消息	Illegal operand
	产生原因	指定的操作数非法。
E2203	消息	Illegal register
	产生原因	指定了不允许被指定的寄存器。
E2204	消息	Illegal character
	产生原因	在源程序模块指定了非法字符。

表 12-2 汇编器错误消息

错误号	错误消息	
E2205	消息	Unexpected LF in string
	产生原因	在字符串关闭之前,字符串中出现回车代码。
E2206	消息	Unexpected EOF in string
	产生原因	在字符串关闭之前,字符串中出现文件结束代码。
E2207	消息	Unexpected null code in string
	产生原因	在字符串中写入了 null 代码 (00H)。
E2209	消息	Too many line number
	产生原因	文件中描述的行数超过了限制范围。
E2301	消息	Too complex expression
	产生原因	表达式太复杂。
E2302	消息	Absolute expression expected
	产生原因	指定了可重定位表达式。
E2303	消息	Illegal expression
	产生原因	所用的表达式格式不正确。
E2304	消息	Illegal symbol in expression 'file name'
	产生原因	表达式中有不可使用的符号。
E2305	消息	Too long string as constant
	产生原因	超出了字符串常量长度限制范围 (4 个字符)。
E2306	消息	Illegal number
	产生原因	指定了错误的数值。
E2307	消息	Division by zero
	产生原因	数值被零除。
E2308	消息	Too large integer
	产生原因	常量的值超过了 16 位。
E2309	消息	Illegal bit value
	产生原因	指定了错误的位型数值。
E2310	消息	Bit value out of range
	产生原因	位型数值超出了 0 至 7 的范围。
E2311	消息	Operand out of range (n)
	产生原因	指定的数值超出了范围 n(0 至 7)。
E2312	消息	Operand out of range (byte)
	产生原因	操作数的值超出了范围 (00H 至 FFH), 或操作数中字节的值超出了范围 (-128 至 +127)。

表 12-2 汇编器错误消息

错误号	错误消息	
E2313	消息	Operand out of range (addr5)
	产生原因	操作数超出了 addr5 的可指定范围 (40H 至 7EH).
E2314	消息	Operand out of range (addr11)
	产生原因	操作数超出了 addr11 的可指定范围 (800H 至 FFFH).
E2315	消息	Operand out of range (saddr)
	产生原因	操作数超出了 saddr 的可指定范围 (0FE20H 至 0FF1FH).
E2316	消息	Operand out of range (addr16)
	产生原因	操作数超出了 addr16 的可指定范围 (根据目标设备不同而异).
E2317	消息	Even expression expected
	产生原因	为字访问指定了奇数地址.
E2318	消息	Operand out of range (sfr)
	产生原因	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.
E2326	消息	Illegal SFR access in operand
	产生原因	描述了不能访问操作数的 SFR 符号.
E2401	消息	Illegal symbol for PUBLIC 'symbol name'
	产生原因	该符号不能被声明为 PUBLIC.
E2402	消息	Illegal symbol for EXTRN/EXBIT 'symbol name'
	产生原因	该符号不能被声明为 EXTRN/EXBIT.
E2403	消息	Can't define PUBLIC symbol 'symbol name'
	产生原因	该符号已具有一个 PUBLIC 声明, 不能对具有 PUBLIC 声明的符号进行 PUBLIC 定义.
	用户应对措施	用位项而不是用 saddr.bit 定义的符号不能够进行 PUBLIC 声明. 取消 PUBLIC 声明或更改 EQU 定义.
E2404	消息	Public symbol is undefined 'symbol name'
	产生原因	没有定义具有 PUBLIC 声明的符号.
E2405	消息	Illegal bit symbol
	产生原因	前向引用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.
	用户应对措施	为位符号指定后向引用或进行 EXTBIT 声明.
E2406	消息	Can't refer to forward bit symbol 'symbol name'
	产生原因	指定前向引用位符号或引用表达式中的位符号.
E2407	消息	Undefined symbol reference 'symbol name'
	产生原因	使用了未定义的符号.

表 12-2 汇编器错误消息

错误号	错误消息	
E2408	消息	Multiple symbol definition 'symbol name'
	产生原因	符号名被定义了一次以上。
E2409	消息	Too many symbols in operand
	产生原因	写入操作数的符号数超过了一行可以描述的符号个数。
E2410	消息	Phase error
	产生原因	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)。
E2411	消息	This symbol is reserved 'symbol name'
	产生原因	定义的符号是保留字。
E2502	消息	Illegal segment name
	产生原因	符号使用非法段名写入。
E2503	消息	Different segment type 'segment name'
	产生原因	用同一个名称定义两个或两个以上的段, 但是类型不同。
E2504	消息	Too many segment
	产生原因	定义的段数超过了限定范围 (256)。
E2505	消息	Current segment is not exist
	产生原因	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。
E2506	消息	Can't describe DB, DW, DS, ORG, label in BSEG
	产生原因	DB, DW, DS, ORG 指令是在位段中定义的。
E2507	消息	Can't describe opcodes outside CSEG
	产生原因	在代码段之外定义了机器语言指令或 BR 指令。
E2508	消息	Can't describe DBIT outside BSEG
	产生原因	在位段之外定义了 DBIT 指令。
E2509	消息	Illegal address specified
	产生原因	定位至绝对段的地址超出了该段的范围。
E2510	消息	Location counter overflow
	产生原因	地址计数器超出了相应段的范围。
E2511	消息	Segment name expected
	产生原因	没有指定段定义指令的段名, 因为重定位属性是 AT。
E2512	消息	Segment size is odd numbers 'segment name'
	产生原因	用奇数描述了重定位属性 callt0 段的大小。

表 12-2 汇编器错误消息

错误号	错误消息	
E2601	消息	Nesting over of include
	产生原因	包含文件嵌套超出了限定范围 (2 级)。
E2602	消息	Must be specified switches
	产生原因	未指定分支名。
E2603	消息	Too many switches described
	产生原因	Switch 名超出了限定范围 (每个模块限定为 5)。
E2604	消息	Nesting over of IF-classes
	产生原因	IF/_IF 嵌套超出了限定范围 (8 级)。
E2605	消息	Needless ELSE statement exists
	产生原因	在不需要 ELSE 语句处存在 ELSE 语句。
E2606	消息	Needless ENDIF statement exists
	产生原因	在不需要 ENDIF 语句处存在 ENDIF 语句。
E2607	消息	Missing ELSE or ENDIF
	产生原因	IF/_IF 语句所需的 ELSE 或 ENDIF 语句不存在。
E2608	消息	Missing ENDIF
	产生原因	IF/_IF 语句所需的 ENDIF 语句不存在。
E2609	消息	Illegal ELSEIF statement
	产生原因	ELSEIF 或 _ELSEIF 语句在 ELSE 语句之后写入。
E2610	消息	Multiple symbol definition (MACRO) 'symbol name'
	产生原因	已定义了用于定义宏名的符号。
E2611	消息	Illegal syntax of parameter
	产生原因	宏的形式参数不正确。
E2612	消息	Too many parameter
	产生原因	宏定义的形式参数个数超过了限定范围 (16)。
E2613	消息	Same name parameter described 'symbol name'
	产生原因	指定的宏定义符号与形式参量的符号相同。
E2614	消息	Can't nest macro definition
	产生原因	宏定义不能在另外的宏定义中进行嵌套。
E2615	消息	Illegal syntax of local symbol
	产生原因	LOCAL 指令中指定的操作数不正确。
E2616	消息	Too many local symbols
	产生原因	可以在一个宏体内描述的局部符号数超出了限定范围 (64)。

表 12-2 汇编器错误消息

错误号	错误消息	
E2617	消息	Missing ENDM
	产生原因	宏定义指令所需的 ENDM 语句不存在。
E2618	消息	Illegal syntax of ENDM
	产生原因	ENDM 语句不正确。
E2619	消息	Illegal defined macro
	产生原因	引用宏未被正确定义。
E2620	消息	Illegal syntax of actual parameter
	产生原因	宏实际参数的定义不正确。
E2621	消息	Nesting over of macro reference
	产生原因	宏引用嵌套超出了限定范围 (8 级)。
E2622	消息	Illegal syntax of EXITM
	产生原因	EXITM 语句不正确。
E2623	消息	Illegal operand of REPT
	产生原因	在 REPT 指令操作数内指定了一个不被允许的表达式。
E2624	消息	More than ??RAFFFF
	产生原因	宏扩展期间, 所替代的局部符号对于 65535 个。
E2625	消息	Unexpected ENDM
	产生原因	发现了非预期的 ENDM。
E2626	消息	Can't describe LOCAL outside macro definition
	产生原因	在正常源程序语句而不是在宏体中指定 LOCAL 指令。
E2627	消息	More than two segments in this include / macro
	产生原因	在包含文件、宏体、rept-endm 块或 irp-endm 块中发现了两个会两个以上的段。
W2701	消息	Too long source line
	产生原因	在源程序语句的一行内描述的字符数超过了 2048 个。
	程序处理	第 2049 个以及以后的字符会被忽略。
W2702	消息	Duplicate PROCESSOR option and control
	产生原因	同时定义了目标设备 (-c) 的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) 的命令行指定选项有效, 而忽略源程序头中的 PROCESSOR 指令。
W2703	消息	Multiple defined module name
	产生原因	NAME 指令被定义 2 次或多次。
	程序处理	NAME 指令不可用而已定义的模块名可用。

表 12-2 汇编器错误消息

错误号	错误消息	
W2704	消息	Already declared EXTRN symbol 'symbol name'
	产生原因	该符号已声明为 EXTRN.
	用户应对措施	在 1 个模块内指定一次 EXTRN 声明 .
W2705	消息	Already declared EXTBIT symbol 'symbol name'
	产生原因	该符号已声明为 EXTBIT.
	用户应对措施	在 1 个模块内指定一次 EXTBIT 声明 .
W2706	消息	Missing END statement
	产生原因	在源文件结束时未写入 END 语句 .
	程序处理	假定源文件在结束时描述了 END 语句 .
W2707	消息	Illegal statement after END directive
	产生原因	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 语句之后的所有语句 .
W2708	消息	Already declared LOCAL symbol 'symbol name'
	产生原因	该符号已声明为 LOCAL.
	用户应对措施	在每个宏内只声明一个符号 LOCAL.
W2709	消息	Few count of actual parameter
	产生原因	设定的实际参数少于形式参数 .
	程序处理	实际参数不足处的形式参数按照空字符串进行处理 .
W2710	消息	Over count of actual parameter
	产生原因	设定的实际参数多于形式参数 .
	程序处理	忽略多余的参数 .
W2711	消息	Too many errors to report
	产生原因	在单行中要报告的错误过多 (即 6 个或更多)
	程序处理	第六个及其以后的错误信息不作输出但是继续进行处理 .
W2712	消息	Insufficient cross-reference work area
	产生原因	用于要处理输出交叉引用列表的存储区不足 .
	程序处理	交叉引用列表不作输出但处理继续进行 .
W2717	消息	Normal, callt and callf functions must be described together respectively.
	产生原因	调试信息可能会由于不同时描述 normal、callt 和 callf 函数非法 .
	程序处理	同时描述 normal 和 callt 函数 .

表 12-2 汇编器错误消息

错误号	错误消息	
E2801	消息	Illegal debug information
	产生原因	源文件中的调试信息非法。
	用户应对措施	再次执行编译或结构化汇编。
F2901	消息	Can't open source file 'file name'
	产生原因	源文件无法打开。
F2902	消息	Can't open parameter file 'file name'
	产生原因	参数文件无法打开。
F2903	消息	Can't open include file 'file name'
	产生原因	包含文件无法打开。
F2904	消息	Illegal include file 'file name'
	产生原因	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。
F2905	消息	Can't open overlay file 'file name'
	产生原因	重叠文件无法打开。
	用户应对措施	确保重叠文件与汇编执行格式文件在相同的目录之下。
F2906	消息	Illegal overlay file 'file name'
	产生原因	重叠文件的内容非法。
F2907	消息	Can't open object file 'file name'
	产生原因	目标文件无法打开。
	用户应对措施	使用在其目录路径下带有打开区域的磁盘。
F2908	消息	Can't open print file 'file name'
	产生原因	不能打开汇编列表文件。
	用户应对措施	使用在其目录路径下带有打开区域的磁盘。
F2909	消息	Can't open error list file 'file name'
	产生原因	不能打开错误列表文件。
	用户应对措施	使用在其目录路径下带有打开区域的磁盘。
F2910	消息	Can't open temporary file 'file name'
	产生原因	临时文件无法打开。
	用户应对措施	使用在其目录路径下带有打开区域的磁盘。

表 12-2 汇编器错误消息

错误号	错误消息	
F2913	消息	Can't read source file 'file name'
	产生原因	在源文件中发生文件输入 / 输出错误。
F2914	消息	Can't read parameter file 'file name'
	产生原因	在参数文件中发生文件输入 / 输出错误。
F2915	消息	Can't read include file 'file name'
	产生原因	在包含文件中发生文件输入 / 输出错误。
F2916	消息	Can't read overlay file 'file name'
	产生原因	在覆盖文件中发生文件输入 / 输出错误。
F2917	消息	Can't write object file 'file name'
	产生原因	在目标文件中发生文件输入 / 输出错误。
	用户应对措施	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。
F2918	消息	Can't write print file 'file name'
	产生原因	在汇编列表文件中发生文件输入 / 输出错误。
	用户应对措施	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。
F2919	消息	Can't write error list file 'file name'
	产生原因	在错误列表文件中发生文件输入 / 输出错误。
	用户应对措施	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。
F2920	消息	Can't read / write temporary file 'file name'
	产生原因	在临时文件中发生文件输入 / 输出错误。
	用户应对措施	将临时文件输出至另外一个目录下或者在指定磁盘创建一个打开区。
C2921	消息	Assembler internal error
	产生原因	发生了汇编器内部错误。
	用户应对措施	如果不能解决错误, 请联系日电电子公司或日电电子公司发行人。
F2922	消息	Insufficient memory in hostmachine
	产生原因	系统没有足够的空间来执行汇编。
F2923	消息	Insufficient memory for macro in hostmachine
	产生原因	在宏处理期间用于宏处理的空间变得不足。
	用户应对措施	减少定义的宏数。

12.4 链接器错误消息

表 12-3 链接器错误消息

错误号	错误消息	
F3001	消息	Missing input file
	产生原因	未指定输入文件。
	用户应对措施	指定一个输入文件。
F3004	消息	Illegal file name 'file name'
	产生原因	在文件名中有非法字符或字符个数超过了限制范围。
	用户应对措施	输入具有合法字符且不超出限制范围的文件名。
F3005	消息	Illegal file specification 'file name'
	产生原因	指定了非法文件。
	用户应对措施	指定一个合法文件。
F3006	消息	File not found 'file name'
	产生原因	指定的文件不存在。
	用户应对措施	指定一个存在的文件。
F3007	消息	Input file specification overlapped 'file name'
	产生原因	输入文件名已在别处进行了定义。
	用户应对措施	输入一个不同的文件名。
F3008	消息	File specification conflicted 'file name'
	产生原因	重复指定了 I/O 文件名。
	用户应对措施	指定不同的 I/O 文件名。
F3009	消息	Unable to make file 'file name'
	产生原因	指定的文件为写保护文件。
	用户应对措施	取消指定文件的写保护。
F3010	消息	Directory not found 'file name'
	产生原因	在输出文件名中包括不存在的驱动器和 / 或目录。
	用户应对措施	指定一个存在的驱动器和 / 或目录。
F3011	消息	Illegal path 'option'
	产生原因	指定的路径与选项中指定的参数路径不同。
	用户应对措施	指定正确的路径名。
F3012	消息	Missing parameter 'option'
	产生原因	未指定必要的参数。
	用户应对措施	指定参数。

表 12-3 链接器错误消息

错误号	错误消息	
F3013	消息	Parameter not needed 'option'
	产生原因	指定了不必要的参数。
	用户应对措施	删除不必要的参数。
F3014	消息	Out of range 'option'
	产生原因	指定的数值超出了范围。
	用户应对措施	指定正确的数值。
F3015	消息	Parameter is too long 'option'
	产生原因	参数中的字符数超出了限制范围。
	用户应对措施	指定一个字符数在限制内范围的参数。
F3016	消息	Illegal parameter 'option'
	产生原因	参数的语法不正确。
	用户应对措施	指定正确的参数。
F3017	消息	Too many parameters 'option'
	产生原因	参数的总数超出了限制范围。
	用户应对措施	在数目限制范围内指定参数。
F3018	消息	Option is not recognized 'option'
	产生原因	选项名称不正确。
	用户应对措施	指定正确的选项名。
F3019	消息	Parameter file nested
	产生原因	已在参数文件内指定了 -f 选项。
	用户应对措施	不要在参数文件内部指定 -f 选项。
F3020	消息	Parameter file read error 'file name'
	产生原因	不能读取参数文件。
	用户应对措施	指定正确的参数文件。
F3021	消息	Memory allocation failed
	产生原因	没有足够的存储空间。
	用户应对措施	确保必须的存储空间。
F3101	消息	'File name' invalid input file (or made by different hostmachine)
	产生原因	输入了目标模块文件之外的文件，或试图与非兼容主机上创建的目标模块文件进行链接。
E3102	消息	Directive syntax error
	产生原因	指令的定义不正确。

表 12-3 链接器错误消息

错误号	错误消息	
F3103	消息	'File name' Illegal processor type
	产生原因	汇编器或编译器的目标设备不是该链接器的目标设备。
	用户应对措施	检查以确保目标模块文件正确。检查确保汇编器或编译器的目标设备可由该链接器处理。还要检查覆盖文件是否为正确版本。(汇编器覆盖文件的引用部分,该部分在目标设备获取字符数据。)
F3104	消息	'File name' Different processor type from first input file 'first input file name'
	产生原因	输入目标设备与首次输入目标模块文件不同的目标模块文件。
W3105	消息	Library file 'file name' has no public symbol
	产生原因	库文件没有全局符号。因此,不能链接库文件中包含的目标模块文件。
F3106	消息	Can't create temporary file 'file name'
	产生原因	不能创建临时文件。
E3107	消息	Name 'name' in directive already defined
	产生原因	试图将保留字或先前定义的名称定义为指令的存储区。该名称(保留字,存储空间名,存储区域名)已被定义。
E3108	消息	Overlapped memory area 'Memory area 1' and 'Memory area 2'
	产生原因	在存储指令中定义的存储区域地址被覆盖。
E3109	消息	Memory area 'Memory area name' too long name (up to 31 characters)
	产生原因	在指令中指定的存储区域名过长。 在指令中指定的存储区域名称为 32 或 32 个以上的字符。
E3110	消息	Memory area 'Memory area name' already defined
	产生原因	在存储指令中指定的存储区域已被注册。
E3111	消息	Memory area 'Memory area name' redefinition out of range
	产生原因	在存储指令中指定的存储区域范围超出了可定义范围。
E3112	消息	Segment 'segment name' wrong allocation type
	产生原因	为合并指令的段指定了错误的定位类型。
C3113	消息	Linker internal error
	产生原因	内部错误。
	用户应对措施	联系日电电子公司或日电电子发行人。
E3114	消息	Illegal number
	产生原因	指令中数值的定义不正确。
E3115	消息	Too large value (up to 65535/0FFFFH)
	产生原因	在指令中描述了大于 65535 (0FFFFH) 的值。

表 12-3 链接器错误消息

错误号	错误消息	
E3116	消息	Memory area 'Memory area name' definition out of range
	产生原因	存储指令中指定的起始地址的和与存储区域的大小超过了 65535 (0FFFFH).
E3117	消息	Can't find target chip in all modules
	产生原因	无法识别目标设备, 因为给所有的输入目标模块文件都指定了系列共用目标指定选项 (-common).
	用户应对措施	删除不必要的系列共用目标指定选项 (-common).
E3201	消息	Multiple segment definition 'segment name' in merge directive
	产生原因	合并指令中指定的段已被注册 (相同的段试图使用重复合并指令来指定位置).
E3202	消息	Segment type mismatch 'segment 1' in file 'segment 2' - ignored
	产生原因	发现了与该段名称相同但是具有不同段类型属性的段.
F3203	消息	Segment 'segment name' unknown segment type
	产生原因	在输入目标模块文件的数据段存在一个错误 (输出段的链接定义不正确).
E3204	消息	Memory area/space 'name' not defined
	产生原因	合并指令中的存储区域 / 空间未定义.
E3205	消息	Name 'name' in directive has bad attribute
	产生原因	在指令中描述了在段名, 存储区域名或存储空间名中不能描述的项 (例如, 在需要存储区域名处描述了一个存储空间名).
E3206	消息	Segment 'segment name' can't allocate to memory - ignored
	产生原因	不能将段分配给存储区 (不存在足够的存储空间来分配段).
E3207	消息	Segment 'segment name' has illegal segment type
	产生原因	该段类型数据不合法.
E3208	消息	Segment 'segment name' may not change attribute
	产生原因	试图改变使用汇编期间指定的重定位属性 'AT xxxxH' 创建的段, 或使用 ORG 指令创建的段的链接类型.
E3209	消息	Segment 'segment name' may not change arrangement
	产生原因	试图改变使用汇编期间指定的重定位属性 'AT xxxxxxH' 创建的段, 或使用 ORG 指令创建的段的分配地址.
	用户应对措施	在链接期间, 不要为汇编器中将要指定链接类型的段指定分配地址.
E3210	消息	Segment 'segment name' is not exist - ignored
	产生原因	指令中指定的段不存在.

表 12-3 链接器错误消息

错误号	错误消息	
E3211	消息	Bank type mismatch 'symbol name' in file 'file name' - ignored
	产生原因	在指定的符号组编号中存在不匹配。
	用户应对措施	确认符号的组标号正确。
E3301	消息	Relocatable object code address out of range (file 'file name', segment 'segment name', address xxxxH, type 'addressing type')
	产生原因	将输入目标模块文件中包含的可重定位目标代码的修正数据输出到了不存在目标代码的地址 (重定位入口地址超出了源数据的范围)。
	用户应对措施	检查引用符号是正确的。
E3302	消息	Illegal symbol index in line number (file 'file name', segment 'segment name')
	产生原因	输入目标模块文件中包含的用于调试的行号数据不正确, 且没有正确引用符号数据。行号索引和符号索引不对应。
E3303	消息	Can't find symbol index in relocatable object code (file 'file name', segment 'segment name', address xxxxH, type 'addressing type')
	产生原因	输入目标模块文件中包含的可重定位代码的修正数据不正确, 且没有正确引用符号数据。重定位入口和符号索引不对应。
	用户应对措施	检查符号和变量的引用方法是否正确。
E3304	消息	Operand out of range (segment 'segment name', address xxxxH, type 'addressing type')
	产生原因	可重定位目标代码结果中所用的操作数值超出了指令对应的操作数值的范围。
	用户应对措施	源程序中描述操作数的值, 源程序在由各寻址类型所决定的范围之内。
E3305	消息	Even value expected (segment 'segment name', address xxxxH, type 'addressing type')
	产生原因	用于决定 callt 或 saddrp 寻址可重定位目标代码的操作数值是奇数 (callt 和 saddrp 寻址操作数必须为偶数)。
E3306	消息	A multiple of 4 value expected (segment 'segment name', address xxxxH, type 'addressing type')
	产生原因	用于处理 saddr 寻址的可重定位目标代码的操作数值不是 4 的倍数。
F3401	消息	'File name' Bad symbol table
	产生原因	输入模块文件的符号数据不合法。输入文件的符号条目没有以 '.file' 开头。
F3402	消息	File 'file name' has no string table for symbol
	产生原因	输入模块文件的符号数据不合法。
	用户应对措施	再次执行汇编或编译。 通过使得汇编器可识别的字符为 8 或编译器可识别的字符为 7 可避免此种情况。

表 12-3 链接器错误消息

错误号	错误消息	
E3403	消息	Symbol 'symbol name' unmatched type in file 'file name1'. First defined in file 'file name2'
	产生原因	文件 1 和文件 2 中具有相同名称的外部定义 / 引用符号的类型不同。
E3404	消息	Multiple Symbol definition 'symbol name' in file 'file name1'. First defined in file 'file name2'
	产生原因	目标模块文件 1 中定义的全局符号已在目标模块文件 2 中声明了 PUBLIC。
E3405	消息	Undefined symbol 'symbol name' in file 'file name'
	产生原因	文件中声明 EXTRN 的符号没有在其它文件中声明 PUBLIC。
W3406	消息	Stack area less than 10 bytes
	产生原因	保护堆栈区域的大小为 10 个字节或更少 (在存储区域中用 -s 选项指定的保护堆栈区域大小为 10 个字节或更少)。
W3407	消息	Can't allocate stack area
	产生原因	在保护堆栈区的存储区域没有可用的空闲区域 (在使用 -s 选项指定的存储区内不能保护堆栈区)。
E3408	消息	Can't find -A symbol
	产生原因	全局符号中不存在程序入口地址定义链接器选项 -a 之后写入的符号。
E3409	消息	-A symbol 'symbol name' is unmatched type
	产生原因	由程序入口地址链接器选项 -a 的定义所查找的符号类型不正确。
	用户应对措施	使用允许的符号类型, 该符号类型将由程序入口地址定义查找。
E3410	消息	Multiple module name definition 'module name' in file 'file 1' First defined in file 'file 2'
	产生原因	目标模块文件 1 的模块名称与目标模块文件 2 的模块名称相同。
W3411	消息	Different REL type in 'file name'
	产生原因	目标模块文件版本的类型不同。
	用户应对措施	使用最新版本重新汇编或重新编译。
E3415	消息	-QD/QF/etc. and Not -QD/QF/etc. REL are mixed
	产生原因	输入目标模块文件对编译器优化选项具有不同的定义, 而优化选项必须与整个程序相同。使用与程序其它部分相同的值进行编译。
W3416	消息	Multiple CAP/NOCAP are in file 'file name (option)' Defined first one in file 'file name (option)'
	产生原因	所有输入目标模块文件中 CAP/NOCAP 汇编或编译选项不相同。

表 12-3 链接器错误消息

错误号	错误消息	
W3417	消息	The version of tool name in file 'file name' are more than one. Used the first one in file 'file name'
	产生原因	所有输入目标模块文件链接阶段之前所用的各工具 (CC78K0S,ST78K0S,RA78K0S) 与设备文件版本之间存在差异。
W3418	消息	File 'file name' is old. Can't find TOOL information
	产生原因	该消息当输入目标模块文件中找不到 TOOL 信息时输出 .. 通常, 此消息总是当使用旧 (DF- 不兼容) 目标模块文件进行链接时输出。
W3420	消息	File 'file name' has already had error(s)/warning(s) by 'tool name'
	产生原因	输出链接阶段之前所用各工具 (CC 78K0S,ST78K0S,RA78K0S) 的错误信息或警告信息。
E3426	消息	Multiple input BOOT file 'file name'
	产生原因	当 flash ROM 模型的引导区域 ROM 程序的载入模块与 flash 区域程序的目标模块链接时, 输入两个或两个以上的引导区域 ROM 程序载入模块。
E3431	消息	There are different function name in same ID (function name) (file 'file name')
	产生原因	由编译器声明为 EXT_FUNC 的两个或多个函数具有相同的 ID 值。
F3901	消息	Can't open overlay file 'file name'
	产生原因	覆盖文件无法打开。
	用户应对措施	确保覆盖文件位于正确的目录 (包含执行程序的目录)。
F3902	消息	File 'file name' not found
	产生原因	无法打开指定的库文件。
F3903	消息	Can't read input file 'file name'
	产生原因	不能读取指定为输入文件的目标模块文件。
F3904	消息	Can't open output file 'file name'
	产生原因	输出文件无法打开。
	用户应对措施	磁盘的检查条件 (打开容量, 介质条件等) 用于创建输出文件。
F3905	消息	Can't create temporary file 'file name'
	产生原因	不能创建用于符号输入项的临时文件。
	用户应对措施	磁盘的检查条件 (打开容量, 介质条件等) 用于试图创建临时文件。
F3906	消息	Can't write map file 'file name'
	产生原因	数据无法写入链接表文件。
	用户应对措施	磁盘的检查条件 (打开容量, 介质条件等) 用于试图创建链接表文件。

表 12-3 链接器错误消息

错误号	错误消息	
F3907	消息	Can't write output file 'file name'
	产生原因	数据无法写入载入模块文件。
	用户应对措施	磁盘的检查条件 (打开容量, 介质条件等) 用于试着创建输出文件。
F3908	消息	Can't access temporary file 'file name'
	产生原因	无法写入临时文件。
	用户应对措施	磁盘的检查条件 (打开容量, 介质条件等) 用于试图创建临时文件。
F3909	消息	Can't read DEVICE_FILE file 'device file name'
	产生原因	由链接阶段之前所用各工具 () 指定的设备对应的设备文件不能读取。

注意事项 E3301 至 E3306 的信息中所示的 ' 地址 xxxxH' 是段定位之后的绝对地址。

12.5 目标转换器错误消息

表 12-4 目标转换器错误消息

错误号	错误消息	
F4001	消息	Missing input file
	产生原因	未指定输入文件。
	用户应对措施	指定一个输入文件。
F4002	消息	Too many input files
	产生原因	指定了两个或两个以上的输入文件。
	用户应对措施	仅指定一个输入文件。
F4004	消息	Illegal file name 'file name'
	产生原因	在文件名中有非法字符或字符个数超过了限制范围。
	用户应对措施	输入具有合法字符且不超出限制范围的文件名。
F4005	消息	Illegal file specification 'file name'
	产生原因	指定了非法文件。
	用户应对措施	指定一个合法文件。
F4006	消息	File not found 'file name'
	产生原因	指定的输入文件不存在。
	用户应对措施	如果链接了 C 编译器的启动程序，则文件输出 " 启动程序名称。这种情况下，指定带有链接器选项的输出文件名，比如 "-o*.lmf"。
F4008	消息	File specification conflicted 'file name'
	产生原因	重复指定了 I/O 文件名。
	用户应对措施	指定不同的 I/O 文件名。
F4009	消息	Unable to make file 'file name'
	产生原因	指定的文件为写保护文件。
	用户应对措施	取消指定文件的写保护。
F4010	消息	Directory not found 'file name'
	产生原因	在输出文件名中包括不存在的驱动器和 / 或目录。
	用户应对措施	指定一个存在的驱动器和 / 或目录。
F4011	消息	Illegal path 'option'
	产生原因	指定的路径与选项中指定的参数路径不同。
	用户应对措施	指定正确的路径名。

表 12-4 目标转换器错误消息

错误号	错误消息	
F4012	消息	Missing parameter 'option'
	产生原因	未指定必要的参数。
	用户应对措施	指定参数。
F4013	消息	Parameter not needed 'option'
	产生原因	指定了不必要的参数。
	用户应对措施	删除不必要的参数。
F4014	消息	Out of range 'option'
	产生原因	指定的数值超出了范围。
	用户应对措施	指定正确的数值。
F4015	消息	Parameter is too long 'option'
	产生原因	参数中的字符数超出了限制范围。
	用户应对措施	指定一个字符数在限制内范围的参数。
F4016	消息	Illegal parameter 'option'
	产生原因	参数的语法不正确。
	用户应对措施	指定正确的参数。
F4017	消息	Too many parameters 'option'
	产生原因	参数的总数超出了限制范围。
	用户应对措施	在数目限制范围内指定参数。
F4018	消息	Option is not recognized 'option'
	产生原因	选项名称不正确。
	用户应对措施	指定正确的选项名。
F4019	消息	Parameter file nested
	产生原因	已在参数文件内指定了 -f 选项。
	用户应对措施	不要在参数文件内部指定 -f 选项。
F4020	消息	Parameter file read error 'file name'
	产生原因	不能读取参数文件。
	用户应对措施	指定正确的参数文件。
F4021	消息	Memory allocation failed
	产生原因	没有足够的存储空间。
	用户应对措施	确保必须的存储空间。

表 12-4 目标转换器错误消息

错误号	错误消息	
F4100	消息	'File name' Illegal processor type
	产生原因	汇编器或编译器的目标设备与该程序的目标设备不同。
	用户应对措施	检查载入模块文件是否正确并检查汇编器或编译器的目标设备。还要检查设备文件的版本是否正确。
F4101	消息	'File name' invalid input file (or made by different hostmachine)
	产生原因	试图输入载入模块文件之外的文件，或试图转换在非兼容主机上创建的载入模块文件。
F4103	消息	Symbol 'symbol name' Illegal attribute
	产生原因	在输入文件的符号属性中存在错误。
F4104	消息	'File name' Illegal input file - not linked
	产生原因	试图输入一个目标模块文件。
F4105	消息	Insufficient memory in hostmachine
	产生原因	存储空间不足以运行程序。
F4106	消息	Illegal symbol table
	产生原因	在输入载入模块文件的符号表中存在错误。
	用户应对措施	如果源程序是用 C 语言编写的，则应确保 C 源程序中的汇编代码满足以下的条件： < 条件 > 如果使用局部符号，请使用以字符串 ?L (例如，?L@01 或 ?L@sym) 开头的符号。但是要保证该符号在 8 个字符以内。不要在外部分定义该符号 (PUBLIC 声明)。
F4107	消息	Can't specify -U option for ROMless device
	产生原因	已经为没有内部 ROM 的设备指定了对象补充选项 (-U)。
E4200	消息	Undefined symbol 'symbol name'
	产生原因	发现了地址不确定的符号。
	用户应对措施	定义符号的值。 该符号引用为外部引用符号。所以如果没有进行外部定义，则在指定该符号值的模块外部对其进行外部定义。
E4201	消息	Out of address range
	产生原因	载入模块文件中目标的地址超出了范围。
W4300	消息	xxxxxxH - yyyyyyH overlapped
	产生原因	输出重叠地址的对象，地址重叠范围：xxxxxxH 至 yyyyyyH。
W4301	消息	Can't initialize RAM area 'address' - 'address'
	产生原因	初始化数值输出到 RAM 区。
	用户应对措施	如果汇编源程序的 DSEG 段中描述 DB/DW，则将 DB/DW 更改为 DS 或在 CSEG 中描述 DB/DW 指令。

表 12-4 目标转换器错误消息

错误号	错误消息	
F4900	消息	Can't open file 'file name'
	产生原因	文件不能打开。
F4901	消息	Can't close file 'file name'
	产生原因	文件不能关闭。
F4902	消息	Can't read file 'file name'
	产生原因	不能正确读取文件。
F4903	消息	Can't access file 'file name'
	产生原因	不能正确读写文件。
F4904	消息	Can't write file 'file name'
	产生原因	数据不能正确写入到输出文件。
F4905	消息	Can't open overlay file 'file name'
	产生原因	覆盖文件无法打开。
	用户应对措施	检查重叠文件是否与汇编执行格式文件在相同的目录之下。
C4999	消息	Object Converter internal error
	产生原因	这是内部错误。
	用户应对措施	联系日电电子公司或日电电子发行人。

12.6 库管理程序错误消息

表 12-5 库管理程序错误消息

错误号	错误消息	
F5001	消息	Missing input file
	产生原因	仅指定了选项，没有指定输入文件。
F5002	消息	Too many input files
	产生原因	输入文件的总数超过了限定范围。
F5003	消息	Unrecognized string '??'
	产生原因	在对话格式命令行中指定了选项以外的其它内容。
F5004	消息	Illegal file name 'file name'
	产生原因	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。
F5005	消息	Illegal file specification 'file name'
	产生原因	在文件名中指定了非法条目项。
F5006	消息	File not found 'file name'
	产生原因	指定的输入文件不存在。
F5007	消息	Input file specification overlapped 'file name'
	产生原因	输入文件名定义被覆盖。
F5008	消息	File specification conflicted 'file name'
	产生原因	输入或输出文件名指定重叠。
F5009	消息	Unable to make file 'file name'
	产生原因	不能创建指定的输出文件。
F5010	消息	Directory not found 'file name'
	产生原因	输出文件名中含有并不存在的驱动器或目录。
F5011	消息	Illegal path 'file name'
	产生原因	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。
F5012	消息	Missing parameter 'option'
	产生原因	未指定所需的参数。
F5013	消息	Parameter not needed 'option'
	产生原因	指定了不必要的参数。
F5014	消息	Out of range 'option'
	产生原因	指定的值超出了范围。
F5015	消息	Parameter is too long 'option'
	产生原因	参数中指定的字符数超出了限定范围。

表 12-5 库管理程序错误消息

错误号	错误消息	
F5016	消息	Illegal parameter 'option'
	产生原因	在参数语法中存在错误。
F5017	消息	Too many parameters 'option'
	产生原因	参数的总数超出了限定范围。
F5018	消息	Option is not recognized 'option'
	产生原因	指定了不正确的选项。
F5019	消息	Parameter file nested
	产生原因	在参数文件中指定了 -f 选项。
F5020	消息	Parameter file read error 'file name'
	产生原因	在读取参数文件时发生了错误。
F5021	消息	Memory allocation failed
	产生原因	存储器定位失败。
F5022	消息	Memory allocation failed
	产生原因	存储器定位失败。
F5023	消息	Illegal character ',' before file name
	产生原因	输入文件之前存在必须的 ','。
F5024	消息	Illegal character
	产生原因	发现了非法字符或字符串。
F5025	消息	Qualifier is not unique.
	产生原因	修饰符的缩写类型不唯一。
F5026	消息	Umbiguous input redirect.
	产生原因	'<' 之后没有指定文件名, 或不止一次指定 '< 文件名'。
C5100	消息	Internal error
	产生原因	发生了内部错误。
E5101	消息	Invalid sub command
	产生原因	子命令名不正确。
E5102	消息	Invalid syntax
	产生原因	子命令中的参数定义不正确。
E5103	消息	Illegal input file - different target chip (file: file name)
	产生原因	输入目标模块文件中的目标设备指定不正确。
E5104	消息	Illegal library file - different target chip (file: file name)
	产生原因	库文件中的目标设备指定不正确。

表 12-5 库管理程序错误消息

错误号	错误消息	
E5105	消息	Module not found (module: file name)
	产生原因	库文件中指定的模块不存在。
E5106	消息	Module already exists (module: file name)
	产生原因	在更新库文件或其它输入文件中已经存在相同名称的模块。
E5107	消息	Master library file is not specify
	产生原因	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 '!'。
E5108	消息	Multiple transaction file (file: file name)
	产生原因	输入目标模块文件重叠。
E5109	消息	Public symbol already exists (symbol: symbol name)
	产生原因	在更新库文件或其它输入文件中已经存在外部定义符号名。
E5110	消息	File specification conflicted (file: file name)
	产生原因	指定的输入文件名与输出文件名相同。
E5111	消息	Illegal file format (file: file name)
	产生原因	更新库文件格式或其它输入文件格式不正确。
E5112	消息	Library file not found (file: file name)
	产生原因	未找到指定的库文件。
E5113	消息	Object module file not found (file: file name)
	产生原因	未找到指定的目标模块文件。
E5114	消息	No free space for temporary file
	产生原因	磁盘中不存在足够的空间来创建临时文件。
E5115	消息	Not enough memory
	产生原因	存储空间不足以运行程序。
E5116	消息	Sub command Buffer full
	产生原因	子命令连续行长度超出了限定范围 (128 x 15 个字符)。 子命令一行长度超出了限定范围 (128 个字符)。
E5117	消息	Can not use device file
	产生原因	在输入文件中指定设备类型文件。 在输入文件或输出文件的命令列表中指定 CLOCK。 在输出目标模块文件或输出库文件中指定 PRN,CON 或 CLOCK。
E5118	消息	Illegal path (file: file name)
	产生原因	指定文件中的路径名不正确。
W5201	消息	Module not found (module: file name)
	产生原因	在库文件中没有指定置换的模块。

表 12-5 库管理程序错误消息

错误号	错误消息	
F5901	消息	File open error (file: file name)
	产生原因	文件不能打开。
F5902	消息	File read error (file: file name)
	产生原因	不能正确读取文件。
F5903	消息	File write error (file: file name)
	产生原因	数据不能正确写入到文件。
F5904	消息	File seek error (file: file name)
	产生原因	发生了文件查找错误。
F5905	消息	File close error (file: file name)
	产生原因	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

错误号	错误消息	
F6001	消息	Missing input file
	产生原因	未指定输入文件。
	用户应对措施	指定一个输入文件。
F6002	消息	Too many input files
	产生原因	指定了两个或两个以上的输入文件。
	用户应对措施	仅指定一个输入文件。
F6004	消息	Illegal file name 'file name'
	产生原因	在文件名中有非法字符或字符个数超过了限制范围。
	用户应对措施	输入具有合法字符且不超出限制范围的文件名。
F6005	消息	Illegal file specification 'file name'
	产生原因	指定了非法文件。
	用户应对措施	指定一个合法文件。
F6006	消息	File not found 'file name'
	产生原因	指定的文件不存在。
	用户应对措施	指定一个存在的文件。
F6008	消息	File specification conflicted 'file name'
	产生原因	重复指定了 I/O 文件名。
	用户应对措施	指定不同的 I/O 文件名。
F6009	消息	Unable to make file 'file name'
	产生原因	指定的文件为写保护文件。
	用户应对措施	取消指定文件的写保护。
F6010	消息	Directory not found 'file name'
	产生原因	在输出文件名中包括不存在的驱动器和 / 或目录。
	用户应对措施	指定一个存在的驱动器和 / 或目录。
F6011	消息	Illegal path 'option'
	产生原因	指定的路径与选项中指定的参数路径不同。
	用户应对措施	指定正确的路径名。
F6012	消息	Missing parameter 'option'
	产生原因	未指定必要的参数。
	用户应对措施	指定参数。

表 12-6 表转换程序错误消息

错误号	错误消息	
F6013	消息	Parameter not needed 'option'
	产生原因	指定了不必要的参数。
	用户应对措施	删除不必要的参数。
F6014	消息	Out of range 'option'
	产生原因	指定的数值超出了范围。
	用户应对措施	指定正确的数值。
F6015	消息	Parameter is too long 'option'
	产生原因	参数中的字符数超出了限制范围。
	用户应对措施	指定一个字符数在限制内范围的参数。
F6016	消息	Illegal parameter 'option'
	产生原因	参数的语法不正确。
	用户应对措施	指定正确的参数。
F6017	消息	Too many parameters 'option'
	产生原因	参数的总数超出了限制范围。
	用户应对措施	在数目限制范围内指定参数。
F6018	消息	Option is not recognized 'option'
	产生原因	选项名称不正确。
	用户应对措施	指定正确的选项名。
F6019	消息	Parameter file nested
	产生原因	已在参数文件内指定了 -f 选项。
	用户应对措施	不要在参数文件内部指定 -f 选项。
F6020	消息	Parameter file read error 'file name'
	产生原因	不能读取参数文件。
	用户应对措施	指定正确的参数文件。
F6021	消息	Memory allocation failed
	产生原因	没有足够的存储空间。
	用户应对措施	确保必须的存储空间。
F6101	消息	File is not 78K/OS 'file name'
	产生原因	输入文件名不是 78K/OS 文件名。
F6102	消息	Load module file is not executable 'file name'
	产生原因	试图输入载入模块文件之外的文件，或试图转换在非兼容主机上创建的载入模块文件。

表 12-6 表转换程序错误消息

错误号	错误消息	
F6103	消息	Load module file has relocation data 'file name'
	产生原因	没有确定载入模块文件的地址。
F6104	消息	Object module file is executable 'file name'
	产生原因	目标模块文件为可执行格式。
F6105	消息	Segment name is not found in module file 'segment name'
	产生原因	未在载入模块文件中发现目标模块文件的段名。
F6106	消息	Segment name is not found in object module file 'segment name'
	产生原因	未在目标模块文件中发现汇编列表文件的段名。
F6107	消息	Not enough memory
	产生原因	存储空间不足以运行程序。
F6108	消息	Load module file has no symbol data 'load module name'
	产生原因	因为在链接器中指定了 -ng 选项，所以不能输出载入模块文件中的符号数据。
F6109	消息	Overlay file can not open 'path name'
	产生原因	汇编器覆盖文件无法打开。
F6110	消息	Illegal assembler list file 'file name'
	产生原因	输入汇编表是汇编表文件类型之外的类型。
W6701	消息	Load module file is older than object module file 'load module file name, object module file name'
	产生原因	指定了比目标模块文件时间要早的载入模块文件。
W6702	消息	Load module file is older than assemble module file 'load module file name, assemble list file name'
	产生原因	指定了比汇编表文件时间要早的载入模块文件。
W6703	消息	Assemble list has error statement 'file name'
	产生原因	在汇编表中存在错误。
W6704	消息	Segment name is not found in assemble list file 'segment name'
	产生原因	未在汇编表中发现目标模块文件的段名。
W6705	消息	Segment data length is different 'segment name'
	产生原因	汇编表文件中的段数据长度与目标模块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进行处理。
F6901	消息	File open error has occurred 'file name'
	产生原因	文件不能打开。
F6902	消息	File read error has occurred 'file name'
	产生原因	不能正确读取文件。

表 12-6 表转换程序错误消息

错误号	错误消息	
F6903	消息	File write error has occurred 'file name'
	产生原因	数据不能正确写入到文件。
F6904	消息	File seek error has occurred 'file name'
	产生原因	发生了文件查找错误。
C6999	消息	Internal error
	产生原因	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，请参见 PM+ 的在线帮助。

12.8.1 结构化汇编预处理器 (ST78K0S)

表 12-7 结构化汇编预处理器 (ST78K0S) DLL 显示的错误消息

错误类型	错误消息	
x	消息	Cannot find ST78K0S.EXE shown in environmental variable PATH .
	产生原因	ST78K0S.EXE 执行格式没有在指定的目录下。
	用户应对措施	重新安装 RA78K0S 汇编包。
	按钮	[OK] ... 关闭消息框
!	消息	Cannot find folder. Will you create?
	产生原因	指定的文件夹不存在。
	用户应对措施	创建一个文件夹或选择另外的文件夹。
	按钮	[OK] ... 创建一个文件夹并关闭消息框。 [Cancel] ... 关闭消息框。
!	消息	Not make folder.
	产生原因	不能创建指定的文件夹。
	用户应对措施	指定另一个文件夹。
	按钮	[OK] ... 关闭消息框。
!	消息	Invalid order size with TAB number.
	产生原因	将大小关系非法的值指定为了制表符数。
	用户应对措施	在可用范围内指定一个值。
	按钮	[OK] ... 关闭消息框。
!	消息	Invalid value. 指令之前的制表符数目范围为 0 至 97.
	产生原因	范围外的一个值被描述为指令前的制表符个数。
	用户应对措施	在可用范围内指定一个值。
	按钮	[OK] ... 关闭消息框。
!	消息	Invalid value. 操作数之前的制表符数目范围为 0 至 98.
	产生原因	范围之外的数值被描述为操作数前制表符的个数。
	用户应对措施	Specify a value in the usable range.
	按钮	[OK] ... 关闭消息框。

表 12-7 结构化汇编预处理器 (ST78K0S) DLL 显示的错误消息

错误类型	错误消息	
!	消息	Invalid value. 注释之前制表符的数目范围为 0 至 99.
	产生原因	范围外的值被指定为注释前制表符的个数
	用户应对措施	在可用范围内指定一个值
	按钮	[OK] ... 关闭消息框
!	消息	Invalid word symbol character.
	产生原因	指定了不能用作符号的字符。
	用户应对措施	指定一个可用字符。
	按钮	[OK] ... 关闭消息框。
!	消息	Too many Include Search Path. Up to 64 can be specified for Include Search Path.
	产生原因	指定的包含文件路径太多。
	用户应对措施	在可用范围内指定包含文件路径的数目。
	按钮	[OK] ... 关闭消息框。
!	消息	Too many characters for Include Search Path.
	产生原因	将包含文件路径长度指定为了超出输入范围的字符个数。
	用户应对措施	在输入范围内指定字符个数。
	按钮	[OK] ... 关闭消息框。
!	消息	(Include file path) Multiple Include Search Path definition.
	产生原因	包含文件路径被重复描述。
	用户应对措施	删除重复的路径和复执。
	按钮	[OK] ... 关闭消息框。
!	消息	Too many symbols for Symbol Definition. Up to 30 symbols can be specified.
	产生原因	给符号定义描述了超出输入范围的符号数。
	用户应对措施	使符号数保持在可用范围之内并复执。
	按钮	[OK] ... 关闭消息框。
!	消息	(symbol) Too many characters for Symbol Definition. Up to 31 characters can be described for symbol name.
	产生原因	已定义符号的长度大于输入范围中的字符数。
	用户应对措施	使符号的字符数保持在范围之内并复执。
	按钮	[OK] ... 关闭消息框。

表 12-7 结构化汇编预处理器 (ST78K0S) DLL 显示的错误消息

错误类型	错误消息	
!	消息	(symbol) Multiple symbol definition.
	产生原因	给符号定义重复描述了一个符号。
	用户应对措施	删除重复的符号并复执。
	按钮	[OK] ... 关闭消息框。
!	选项	Can't set options to (source file name).
	产生原因	如果在一个普通选项由一个专用选项产生，则专用选项的定义变为非法。
	用户应对措施	检查专用选项的定义并复执。
	按钮	[OK] ... 个别消息框。

12.8.2 汇编器 (RA78K0S)

表 12-8 汇编器 (RA78K0S) DLL 显示的错误消息

错误类型	错误消息	
x	消息	Cannot find RA78K0S.EXE shown in environmental variable PATH.
	产生原因	RA78K0S.EXE 执行格式不在指定的目录下。
	用户应对措施	重新安装 RA78K0S 汇编包。
	按钮	[OK] ... 关闭消息框。
!	消息	Cannot find folder. Will you create?
	产生原因	指定的文件夹不存在。
	用户应对措施	创建一个文件夹或消息另外的文件夹。
	按钮	[OK] ... 创建一个文件夹并关闭消息框。 [Cancel] ... 关闭消息框。
!	消息	Not make folder.
	产生原因	不能创建指定的文件夹。
	用户应对措施	指定另一个文件夹。
	按钮	[OK] ... 关闭消息框。
!	消息	Invalid value. The range of columns per line is from 72 to 2046.
	产生原因	超出限定范围的值被指定为一行中的字符个数。
	用户应对措施	指定范围内的值。
	按钮	[OK] ... 关闭消息框。
!	消息	Invalid value. The range of line per page is from 0, or 20 to 32767.
	产生原因	超出限定范围的值被指定为一页中的行数。
	用户应对措施	指定限定范围内的值。
	按钮	[OK] ... 关闭消息框。
!	消息	Invalid value. The range of TAB character is from 0 to 8.
	产生原因	超出输入限定范围的值被描述为操作数前的制表符个数。
	用户应对措施	指定限定范围内的值。
	按钮	[OK] ... 关闭消息框。

表 12-8 汇编器 (RA78K0S) DLL 显示的错误消息

错误类型	错误消息	
!	消息	Too many Include Search Path. Up to 64 can be specified for Include Search Path.
	产生原因	包含文件的路径数超过了指定的输入限定范围。
	用户应对措施	指定限定范围内的值。
	按钮	[OK] ... 关闭消息框。
!	消息	Too many characters for Include Search Path.
	产生原因	包含文件路径长度被指定为超出输入范围的字符个数。
	用户应对措施	在输入范围内指定字符个数。
	按钮	[OK] ... 关闭消息框。
!	消息	(Include file path) Multiple Include Search Path definition.
	产生原因	包含文件路径被重复描述。
	用户应对措施	删除重复路径并复执。
	按钮	[OK] ... 关闭消息框。
!	消息	Too many symbols for Symbol Definition. Up to 30 symbols can be specified.
	产生原因	为符号定义描述了超出输入范围的符号数。
	用户应对措施	将符号数保持在可用范围内并复执。
	按钮	[OK] ... 关闭消息框。
!	消息	(symbol) Too many characters for Symbol Definition. Up to 31 characters can be described for symbol name.
	产生原因	已定义符号的长度大于输入范围内的字符个数。
	用户应对措施	将符号的字符个数保持在范围之内并复执。
	按钮	[OK] ... 关闭消息框。
!	消息	(symbol) Multiple symbol definition.
	产生原因	为符号定义重复描述了一个符号。
	用户应对措施	删除重复的符号并复执。
	按钮	[OK] ... 关闭消息框。
!	消息	Can't set options to (source file name).
	产生原因	如果一个普通选项由专用消息产生，则专用消息变为非法。
	用户应对措施	检查专用选项的定义并复执。
	按钮	[OK] ... 关闭消息框。

12.8.3 链接器 (LK78K0S)

表 12-9 链接器 (LK78K0S) DLL 显示的错误消息

错误类型	错误消息	
x	消息	Cannot find LK78K0S.EXE shown in environmental variable PATH.
	产生原因	LK78K0S.EXE 执行格式不在指定的目录下。
	用户应对措施	重新安装 RA78K0S 汇编包。
	按钮	[OK] ... 关闭消息框。
!	消息	Invalid value. The range of warning level is from 0 to 2.
	产生原因	超出限定输入范围的值被指定为警告级。
	用户应对措施	指定限定范围内的值。
	按钮	[OK] ... 关闭消息框。
!	消息	Cannot find path or file. Make sure path or filename.
	产生原因	找不到指定的路径或文件。
	用户应对措施	指定正确的路径或文件名。
	按钮	[OK] ... 关闭消息框。
!	消息	Cannot find folder. Will you create?
	产生原因	指定的文件夹不存在。
	用户应对措施	创建一个文件夹或选择另外的文件夹。
	按钮	[OK] ... 创建一个文件夹并关闭消息框。 [Cancel] ... 关闭消息框。
!	消息	Not make folder.
	产生原因	不能创建指定的文件夹。
	用户应对措施	指定另外的文件夹。
	按钮	[OK] ... 关闭消息框。
!	消息	Too many files for Library File. Up to 10 can be specified for Library File.
	产生原因	库文件的个数超出了指定的输入限定范围。
	用户应对措施	指定限定范围之内的库文件个数。
	按钮	[OK] ... 关闭消息框。

表 12-9 链接器 (LK78K0S) DLL 显示的错误消息

错误类型	错误消息	
!	消息	(library file) Too long file name for Library File.
	产生原因	指定为库文件名的文件名的字符数超出了限定范围。
	用户应对措施	使用限定范围的字符数进行复执。
	按钮	[OK] ... 关闭消息框。
!	消息	(library file) Multiple Library File definition.
	产生原因	库文件被重复描述。
	用户应对措施	删除重复的库文件并复执。
	按钮	[OK] ... 关闭消息框。
!	消息	Too many path for Library File Search Path. Up to 64 can be specified.
	产生原因	描述的读取库文件路径数超出了限定的输入范围。
	用户应对措施	在限定范围内指定路径数并复执。
	按钮	[OK] ... 关闭消息框。
!	消息	(library file read path) Too many characters for Library File Search Path.
	产生原因	库文件读取路径使用超出了限定输入范围的字符数进行描述。
	用户应对措施	使用限定范围内的字符数指定库文件读取路径。
	按钮	[OK] ... 关闭消息框
!	消息	(library file read path) Multiple Library File Search Path definition.
	产生原因	库文件读取路径被重复描述。
	用户应对措施	删除重复的路径并复执。
	按钮	[OK] ... 关闭消息框。

12.8.4 目标转换器 (OC78K0S)

表 12-10 目标转换器 (OC78K0S) DLL 显示的错误消息

错误类型	错误消息	
x	消息	Cannot find OC78K0S.EXE shown in environmental variable PATH.
	产生原因	OC78K0S.EXE 执行格式不在指定的目录下。
	用户应对措施	重新安装 RA78K0S 汇编包。
	按钮	[OK] ... 关闭消息框。
!	消息	Invalid description format with object complement.
	产生原因	对象填充值被指定为非法格式。
	用户应对措施	指定可描述的格式。
	按钮	[OK] ... 关闭消息框
!	消息	Invalid value. The range of complement value is from 0h to 0ffh.
	产生原因	超出输入限定范围的值被指定为填充值。
	用户应对措施	指定限定范围内的值。
	按钮	[OK] ... 关闭消息框。
!	消息	Out of range. The range of start address is from 0h to (the maximum address of program area except sfr area).
	产生原因	超出输入限定范围的值被指定为起始地址。
	用户应对措施	指定限定范围内的值。
	按钮	[OK] ... 关闭消息框。
!	消息	Out of range. The range of size is from 1h to ((the maximum address of program area except sfr area) - (start address) + 1h).
	产生原因	超出限定输入范围的值被指定为大小。
	用户应对措施	指定限定范围内的值。
	按钮	[OK] ... 关闭消息框。
!	消息	Cannot find folder. Will you create?
	产生原因	指定的文件夹不存在。
	用户应对措施	创建一个文件夹或选择另一个文件夹。
	按钮	[OK] ... 创建一个文件夹并关闭消息框。 [Cancel] ... 关闭消息框。

表 12-10 目标转换器 (OC78K0S) DLL 显示的错误消息

错误类型	错误消息	
!	消息	Not make folder.
	产生原因	不能创建指定的文件夹。
	用户应对措施	指定另外的文件夹。
	按钮	[OK] ... 关闭消息框。

12.8.5 库管理程序 (LB78K0S)

表 12-11 库管理程序 (LB78K0S) DLL 显示的错误消息

错误类型	错误消息	
x	消息	Cannot find LB78K0S.EXE shown in environmental variable PATH.
	产生原因	LB78K0S.EXE 执行格式不在指定的目录下。
	用户应对措施	重新安装 RA78K0S 汇编包。
	按钮	[OK] ... 关闭消息框。
!	消息	Cannot find folder. Will you create?
	产生原因	指定的文件夹不存在。
	用户应对措施	创建一个文件夹或选择另一个文件夹。
	按钮	[OK] ... 创建一个文件夹并关闭消息框。 [Cancel] ... 关闭消息框。
!	消息	Not make folder.
	产生原因	不能创建指定的文件夹。
	用户应对措施	指定另外的文件夹。
	按钮	[OK] ... 关闭消息框。

12.8.6 表转换程序 (LC78K0S)

表 12-12 表转换程序 (LC78K0S) DLL 显示的错误消息

错误类型	错误消息	
x	消息	Cannot find LC78K0S.EXE shown in environmental variable PATH.
	产生原因	LC78K0S.EXE 执行格式不在指定的目录下。
	用户应对措施	重新安装 RA78K0S 汇编包。
	按钮	[OK] ... 关闭消息框。
!	消息	Cannot find folder. Will you create?
	产生原因	指定的文件夹不存在。
	用户应对措施	创建一个文件夹或选择另一个文件夹。
	按钮	[OK] ... 创建一个文件夹并关闭消息框。 [Cancel] ... 关闭消息框。
!	消息	Not make folder.
	产生原因	不能创建指定的文件夹。
	用户应对措施	指定另外的文件夹。
	按钮	[OK] ... 关闭消息框。

附录 A 示例程序

本章介绍了适用于 RA78K0S 的示例列表。

A.1 k0smain.asm

```
NAME      SAMPM
;
; *****
;      HEX -> ASCII 转换程序
;      主例程
; *****
;

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      ; 在 HL 寄存器中设置 2 位十六进制代码数据

          CALL      !CONVAH          ; 进行 ASCII <- HEX 转换
                                      ; 将 ASCII 代码输出到 BC 寄存器中
          MOVW      DE , #STASC      ; 将 ASCII 代码表存储到 DE 中
          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
;      主例程
;      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          ; 加载十六进制高位代码
        CALL    !SASC
        MOV     B , A            ; 存储结果

        XOR     A , A
        XCH     A , [HL]
        AND     A , #0FH          ; 加载十六进制低位代码
        CALL    !SASC
        MOV     C , A            ; 存储结果
        RET

; *****
;      子例程      转换 ASCII 代码
;
;      输入      Acc ( 低 4 位 )    <- 十六进制代码
;      输出      Ac                  <- ASCII 代码
; *****

SASC :
        CMP     A , #0AH          ; 检查十六进制代码是否大于 9
        BC      $SASC1
        ADD     A , #07H          ; 偏差 ( +7H )
SASC1 :
        ADD     A , #30H          ; 偏差 ( +30H )
        RET

        END

```

A.3 test1.s

```

EXTRN  SEARCH , STABLE
EXTRN  @_STBEG
PUBLIC MAIN , START
; *****
;
;      搜索数据
; *****
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               *
; *           DE      表头地址                     *
; *   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--
      endw
      POP     DE
      POP     HL
      A = [ DE ]
      A += E
      E = A
      if ( CY )
          D++
      endif
      endw
      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

本章中，当使用 RA 78K0S 时，需注意以下几点：

(1) 设备文件注意事项

要执行 RA78K0S 需要有设备文件。设备文件不包括在 RA78K0S 软件包中，必须单独购买。

从 NEC' 电子 Microcomputer 主页上下载相关的设备文件。

(<http://www.necel.com/micro/ods/eng/tool/DeviceFile/list.html> -> Development tools download (ODS))

(2) 目标转换器注意事项

通过指定 -r（目标的地址源）和 -u（指定填充值）选项，使用目标转换器。

这些选项均由默认选项指定。

当未存储目标的地址时，如果命令 ROM 代码（作为“交叉处理”或“磁带跳出”工作），则会发生错误。因此，务必指定 -r（不要取消指定）。

(3) 存储器初始化准伪指令注意事项

如果在数据段（DSEG）中描述存储器初始化准伪指令 DW 和 DB，也会输出目标代码，而目标转换器则输出警告信息 W4301。这是因为代码存放在地址中而不是 ROM 区（代码区）。

如果在该状态下命令一个 ROM 代码（作为“交叉处理”或“磁带跳出”工作），则会发生错误。

(4) 目标转换器操作指定注意事项

如果由目标转换器选项 -u 指定了起始地址，则将从起始地址或代码位置处的地址（较低地址）开始填充，不填充 SFR 区域（FF00H 至 FFFFH）。

描述格式：-u 填充值 [, [起始], 大小]

可能省略 []。

(5) 存储器伪指令注意事项

不能擦除各个设备的默认存储器区名。

将不使用的默认存储区名的大小设置为 0。

但是，将一些段分配至默认区域。当改变区名时，要多加留心。

有关默认存储器名，请参考要使用的设备的用户手册。

(6) 调试选项注意事项

如果执行由 C 编译器 / 结构化汇编器预处理器和编译器 / 结构化汇编器输出的调试信息，当对输出汇编源程序进行汇编（通过使用 -nga 选项指定）时，则不输出调试信息。如果输出调试信息，则在 C 编译器 / 结构化汇编器源程序层次上可能不执行调试。

(7) 有关 CC78K0S 的注意事项

当通过对 CC 78K0S 输出的汇编器源程序进行汇编, 来执行 C 源程序层次调试时, 有几点必须注意.

有关细节, 请参考 C 编译器软件包附带的文档 (使用注意事项).

(8) 有关 ID78K0S-NS 和 SM78K0S 的注意事项

当用 ID78K0S-NS 或 SM78K0S 执行调试时, 要保持符号数和源程序的行数在 ID78K0S-NS 和 SM78K0S 的限制范围内.

有关细节, 请参考调试器 / 仿真器附带的文档 (使用注意事项).

(9) 段名注意事项

当描述一个段名时, 不要描述与源文件名的初始名相同的名称. 否则, 将发生异常错误 F2106, 来作为汇编结果.

(10) 在结构化汇编器源程序中描述宏定义注意事项

使用汇编语言替代结构化汇编语言, 来描述结构化汇编器预处理器源程序中的宏定义. 否则, 可能发生重复标签定义错误.

(11) SFR 名 EQU 定义的注意事项

SFR 名可能描述为 EQU 准伪指令的操作数. 如果 saddr 区域外的 SFR 名指定为 PUBLIC, 则会发生汇编错误.

(12) 使用网络时的注意事项

如果将要创建临时文件的目录放置到网络共享的文件系统中, 则可能会发生文件冲突, 从而可能执行异常操作.

通过使用选项和环境变量可以避免这种冲突发生.

附录 C 选项列表

本附录中对程序选项按表格形式进行了总结。

开发程序时请参考本附录。

该选项列表也可用于索引。

C.1 结构化汇编器选项

表 C-1 结构化汇编器选项

分类	格式	功能	与其它选项的关系	省略时的说明
设备类型指定	-c 设备类型	指定目标设备的类型。	独立选项	不能省略。
子符号字符指定	-sc 字符	指定子符号名的最后字符。	独立选项	-scp
符号定义指定	-d 符号 [= 数字值] [, 符号名 [= 数字值] ...] ... (可指定两个或更多的路径名)	指定用于 #IFDEF 等指令的符号。	独立选项	无
制表符数目指定	-wt 数字值, 数字值, 数字值	指定转换指令的输出位置。	独立选项	-wt2, 3, 4
包含文件读取路径	-i 路径名 [, 路径名] ... (可指定两个或更多的路径名)	从指定路径读取输入文件。	独立选项	源文件存在的路径。 由环境变量 'INC78K0S' 指定的路径。
辅助源文件指定	-o [文件名]	指定辅助源文件。	独立选项	-o 输入文件名 .asm
错误列表文件指定	-e [文件名]	输出错误列表文件。	独立选项	-e 输入文件名 .est
参数文件指定	-f 文件名	从指定文件输入输入文件名和选项。	独立选项	选项和输入文件名只能从命令行输入。
调试数据输出指定	-gs	指定结构化汇编器源程序级调试数据输出。	如果同时指定选项 -gs 和选项 -ng , 则后指定的有效。	-gs
	-ngs	使选项 -gs 失效。		

表 C-1 结构化汇编器选项

分类	格式	功能	与其它选项的关系	省略时的说明
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	没有强制输出
Kanji 码 (2- 字节代码) 指定	-zs	注释中描述的 Kanji 可翻译为 shift JIS 码。 .	如果同时指定 -zs, -ze 和 - zn 选项, 则最后指定的优先。 .	在 Windows 中 : -zs 在 Solaris 中: - ze
	-ze	注释中描述的 Kanji 可翻译为 EUC 码。 .		
	-zn	注释中描述的字符不 能翻译为 kanji。 .		
设备文件搜索 路径指定	-y 路径名	从指定路径读取设备 文件。 .	独立选项	用于启动 ST78K0S 的路 径 <..\dev>。 .
帮助指定	--	在显示屏上输出显示 信息。 .	其它选项都无效	不显示

C.2 汇编器选项

表 C-2 汇编器选项

分类	格式	功能	与其它选项的关系	省略时的说明
设备类型指定	-c 设备类型	指定目标设备的设备类型。	独立选项	不能省略
目标模块文件输出指定	-o [文件名]	指定目标模块文件输出。	如果同时指定选项 -o 和 -no，则后指定的选项优先。	-o 输入文件名 .rel
	-no	指定没有目标模块文件输出。		
强制目标模块文件输出指定	-j	指定即使发生致命错误时也输出目标模块文件。	如果同时指定选项 -j 和 -nj，则后指定的选项优先。	-nj
	-nj	使选项 -j 无效。		
调试数据输出指定	-g	指定将局部符号数据添加到目标模块文件。	如果同时指定选项 -g 和 -ng，则后指定的选项优先。	-g
	-ng	使选项 -g 无效。		
	-ga	指定通过结构化汇编器将源程序调试数据添加到目标模块文件。	通过同时指定选项 -ga 和 -nga，则后指定的选项优先。	-ga
	-nga	使选项 -ga 无效。		
包含文件读取路径指定	-i 路径名 [, 路径名] ... (可指定两个或更多的路径名)	指定从指定路径输入包含文件。	独立选项	源文件存在的路径。 由环境变量指定的路径 (INC78K0S)
汇编列表文件输出指定	-p [文件名]	指定汇编列表文件输出。还可指定输出文件的目的地址和文件名。	如果同时指定选项 -p 和 -np，则后指定的选项优先。	-p 输入文件名 .rel
	-np	使选项 -p 无效。		
汇编列表文件数据指定	-ka	在汇编列表文件中输出汇编列表。	如果同时指定选项 -ks 和 -KX，则 -ks 被忽略。 如果同时指定选项 -ka 和 -nka，或者选项 -ks 和 -nks，或者选项 -kx 和 -nkx，则最后指定的选项优先。 如果都指定 -nka, -nks 和 -nkx，则选项 -p 无效。	-ka
	-nka	使选项 -ka 无效。		-nks
	-ks	在汇编列表文件中的符号列表后输出汇编列表。		
	-nks	使选项 -ks 无效。		-nkx
	-kx	在汇编列表文件中的交叉引用列表后输出汇编列表。		
	-nkx	使选项 -kx 无效。		

表 C-2 汇编器选项

分类	格式	功能	与其它选项的关系	省略时的说明
汇编列表文件格式指定	-lw [字符的数目]	改变在列表文件中 1 行可打印的字符数。	如果指定选项 -np 则选项 -lw 无效。	-lw132
	-ll [行数]	改变在汇编列表文件中 1 页可打印的行数。	如果指定选项 -np 则选项 -ll 无效。	-ll66
	-lh 字符串	指定在汇编列表文件标题栏打印的字符串。	如果指定选项 -np 则选项 -lh 无效。	无
	-lt [字符的数目]	指定标签中的字符数。	如果指定选项 -np i 则选项 -lt 无效。	-lt8
	-lf	在汇编列表文件的最末插入换页码 (FF)。	如果同时指定 -lf 和 -nlf 选项, 则后指定的选项优先。 如果指定选项 -np, 则选项 -lf 无效。	-nlf
	-nlf	使选项 -lf 无效。		
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, 则后指定的选项优先。	-ne
	-ne	使选项 -e 无效。		
参数文件指定	-f 文件名	从指定文件输入汇编器选项和输入文件名。	独立选项	选项和输入文件名只能从命令行输入。
创建临时文件路径指定	-t 路径名	在指定的路径创建临时文件。	独立选项	由环境变量 'TMP' 指定的路径
Kanji 码指定	-zs	注释中描述的 Kanji 可翻译为 shift JIS 码。	通过同时指定 -zs, -ze, 和 -zn 选项, 则最后指定的选项优先。	在 Windows 中 : -zs 在 Solaris 中 : -ze
	-ze	注释中描述的 Kanji 可翻译为 EUC 码。		
	-zn	注释中描述的字符不能翻译为 kanji。		
设备文件搜索路径指定	-y 路径名	从指定路径读取设备文件。	独立选项	<..\dev> (RA78K0S 启动的路径)
符号定义指定	-d 符号名 [= 值] [, 符号名 [= 值] ...] (可指定两个或更多的路径名)	定义符号。	独立选项	无
系列通用目标指定	-common	指定 78K0S 系列通用目标模块输出	独立选项	无
自编程指定	-self	使用自编程时指定	独立选项	无

表 C-2 汇编器选项

分类	格式	功能	与其它选项的关系	省略时的说明
帮助指定	--	在显示屏上显示帮助信息。	如果指定选项 --，则其它项都无效	不显示

C.3 链接器选项列表

表 C-3 链接器选项列表

分类	格式	功能	与其它选项之间的关系	忽略指定时的说明
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 are 选项，则最后指定的选项优先。	-o 输入文件名 .lmf
	-no	不输出载入模块文件。		
强制载入模块文件输出指定	-j	强制输出载入模块文件。	如果同时指定了 -j 和 -nj 选项，则最后指定的选项优先。	-nj
	-nj	使得 -j 选项不可用。		
调试数据输出指定	-g	向载入模块文件输出调试数据。	如果同时指定了 -g 和 -ng 选项，则最后指定的选项优先。当指定 -ng 选项时，不论是否指定了 -kp 或 -kl，全局符号列表及局部符号列表均不能输出。	-g
	-ng	使得 -g 选项不可用。		
堆栈判定符产生指定	-s [区名称]	自动生成堆栈判定全局符号。	如果同时指定了 -s 和 -ns 选项，则最后指定的选项优先。	-ns
	-ns	使得 -s 选项不可用。		
指令文件指定	-d 文件名	将指定的文件输入为指令文件。	独立选项程序	-
链接列表文件输出指定	-p [文件名]	指定输出链接列表文件。	如果同时指定了 -p 和 -np 选项，则最后指定的选项优先。	-p 文件名 .map
	-np	使得 -p 选项不可用。		

表 C-3 链接器选项列表

分类	格式	功能	与其它选项之间的关系	忽略指定时的说明
链接列表文件数据指定	-km	将映像列表输出至链接列表文件。	如果同时指定了 -km 和 -nkm 选项，则最后指定的选项优先。	-km
	-nkm	使得 -km 选项不可用。	如果 -nkm, -nkp 和 -nkl are 选项都被同时指定，则即使指定了 -p 选项，也不能输出链接列表文件。	
	-kd	将链接指令文件输出至链接列表文件。	如果指定了 -nkm 选项，则 -kd 选项变为不可用。 如果同时指定了 -kd 和 -nkd, -kp 和 -nkp, 或 -kl 和 -nkl a 选项，t 最后指定的选项优先。 如果指定了 -ng 选项，则即使指定了 -kp 或 -kl 选项也不能输出全局符号列表和局部符号列表。	-kd
	-nkd	使得 -kd 选项不可用。		-nkp
	-kp	将全局符号列表输出至链接列表文件。		
	-nkp	使得 -kp 选项不可用。		-nkl
	-kl	将局部符号列表输出至链接列表文件。		
	-nkl	使得 -kl 选项不可用。		
链接列表格式指定	-ll [行数]	指定链接列表文件中一页可以打印的行数。	如果指定了 -np 选项，则 -ll 选项不可用。	-ll66
	-lf	在链接列表文件末插入换页 (FF) 码。	如果同时指定了 -lf 和 -nlf 选项，则最后指定的选项优先。 如果指定了 -np 选项，则最后指定的选项优先。	-nlf
	-nlf	使得 -lf 选项不可用。		
错误列表文件输出指定	-e [文件名]	输出错误列表文件。	如果同时指定了 -e 和 -ne 选项，则最后指定的选项优先。	-ne
	-ne	缺省值：-ne 使得 -e 选项不可用。		
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	-
库文件读取路径指定	-i 路径名 [, 路径名] ... (可以指定两个或两个以上的路径名)	由指定路径读取库文件。	如果由选项 -b 指定了没有路径名的库文件，则选项 -i 不可用。	路径由环境变量 'LIB78K0S' 指定。

表 C-3 链接器选项列表

分类	格式	功能	与其它选项之间的关系	忽略指定时的说明
参数文件指定	-f 文件名	由指定文件输入链接器选项和输入文件名。	独立选项程序	选项和输入文件名仅能从命令行输入。
临时文件创建路径的指定	-t 路径名	在指定路径创建临时文件。	独立选项程序	路径由环境变量 'TMP' 指定。
设备文件查找路径指定	-y 路径名	由指定路径读取设备文件。	独立选项程序	从 <.\dev> 读取设备文件 (对于 lk78k0s.exe 启动路径)
警告消息输出指定	-w [level]	指定是否向控制台输出警告消息。	独立选项程序	输出一般错误消息
帮助指定	--	在显示器上显示帮助信息。	所有其它选项均不可用。	不显示

C.4 目标转换器选项列表

表 C-4 目标转换器选项列表

分类	格式	功能	与其它选项的关系	省略该选项时有 关说明
十六进制格式目标模块文件输出指定	-o [文件名]	输出十六进制格式目标模块文件。	如果同时指定了 -o 和 -no 选项，那么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)
	-no	不输出十六进制格式目标模块文件。		
符号表文件输出指定	-s [文件名]	输出符号表文件。	如果同时指定了 -s 和 -ns 选项，那么最后指定的选项优先。	-s [输入文件名].sym (文件类型 S1 至 S15 用于扩展空间)
	-ns	不输出符号表文件。		
由目标地址次序进行排序指定	-r	按地址次序对十六进制格式目标进行排序。	如果同时指定了 -s 和 -ns 选项，那么最后指定的选项优先。 如果指定 -no 选项，则选项 -r 不可用。	-r
	-nr	使选项 -r 不可用。		
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域目标代码输出至没有十六进制格式目标输出的地址区中。	如果同时指定 -u 和 -nu ，则最后指定的选项优先。 如果指定了选项 -no ，则 -u 变为不可用。	-u0FFH
	-nu			
错误列表文件输出指定	-e [文件名]	输出一个错误列表文件。	如果同时指定了 -e 和 -ne 选项，那么最后指定的选项优先。	-ne
	-ne	使选项 -e 不可用。		
参数文件指定	-f 文件名	从指定文件中输入选项和输入文件名。	独立选项	仅能从命令行输入选项和输入文件名。
设备文件搜索路径指定	-y 路径名	从指定路径中读出设备文件。	独立选项	用于启动 OC78K0S 的路径 <..\dev>。
帮助指定	--	在显示屏上显示帮助信息。	所有其它选项均不可用。	无显示

C.5 库管理程序选项列表

表 C-5 库管理程序列表

分类	格式	功能	与其它选项的关系	省略该选项时有 关说明
列表文件格式指定	-lw [字符数]	改变列表文件中可在 一行上打印的字符数 .	如果没有指定 list 子命令， 则该选项不可用。	-lw132
	-ll [行数]	改变列表文件中可在 一页上打印的行数。		-ll66
	-lf	在列表文件末尾插入 分页码 (FF).	如果同时指定选项 -lf 和 -nlf , 则最后指定的选项优先。	-nlf
	-nlf	是选项 -lf 不可用。		
临时文件创建 路径指定	-t 路径名	在指定的路径中创建 临时文件。	独立选项	由环境变量 'TMP' 在指定的 路径创建。
设备文件搜索 路径指定	-y 路径名	从指定的路径中读出 设备文件。	独立选项	<..\dev> (用于 lb78k0s 的启动 路径)
帮助制定	--	在显示屏上显示帮助 信息。	所有其它选项均不可用。	无显示

C.6 列表转换程序选项列表

表 C-6 列表转换程序选项列表

分类	格式	功能	与其它选项的关系	省略该选项时有 关说明
目标模块文件输入指定	-r [文件名]	指定目标模块文件的输入。	独立选项	-r 汇编列表文件名 .rel
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	-l 汇编列表文件名 .lnk
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	-o 汇编列表文件名 .p
错误列表文件输出指定	-e [文件名]	输出错误列表文件。	如果同时指定选项 -e 和 -ne, 则最后指定的选项优先。	-ne
	-ne	使选项 -e 不可用。		
参数文件指定	-f 文件名	从指定的文件输入选项和输入文件名。	独立选项	仅能从命令行输入选项和输入文件名。
帮助指定	--	在显示屏 (控制台) 上显示帮助信息。	所有其它选项均不可用。	无显示无显示

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。

开发软件程序时参考本列表有助于程序开发。

该子命令列表也可用作索引。

表 D-1 子命令列表

分类	格式	功能	缩写格式
create	create Δ 库文件名 [Δ 处理]	创建新的库文件。	c
add	add Δ 库文件名 Δ 处理	向库文件添加模块。	a
delete	delete Δ 库文件名 Δ (Δ 模块名 [Δ , ...] Δ)	从库文件删除模块。	d
replace	replace Δ 库文件名 Δ 处理	在库文件中用一个模块替代另一模块。	r
pick	pick Δ 库文件名 Δ (Δ 模块名 [Δ , ...] Δ)	从现有库文件中检索指定模块。	p
list	list [Δ 选项] 库文件名 [Δ (Δ 模块名 [Δ , ...] Δ)]	输出库文件中的模块数据。	l
help	help	在显示屏上显示帮助信息。	h
exit	exit	退出库管理程序。	e

附录 E 索引

Symbols

-- (LB78K0S) ... 238
-- (LC78K0S) ... 272
-- (LK78K0S) ... 188
-- (OC78K0S) ... 218
-- (RA78K0S) ... 137
-- (ST78K0S) ... 86

A

Abort error ... 301
Absolute assemble list ... 259, 293
add ... 241
.asm ... 67, 94
Assemble list ... 259, 279, 297
Assembler ... 17, 27
AT ... 150

B

-b (LK78K0S) ... 181

C

-c (RA78K0S) ... 104
-c (ST78K0S) ... 74
-common (RA78K0S) ... 135
COMPLETE ... 150
create ... 240
Cross-reference list ... 282

D

-d (LK78K0S) ... 167
-d (RA78K0S) ... 134
-d (ST78K0S) ... 76
delete ... 242
.dr ... 146

E

-e (LC78K0S) ... 269
-e (LK78K0S) ... 180
-e (OC78K0S) ... 214
-e (RA78K0S) ... 127
-e (ST78K0S) ... 80
.elk ... 146
.elv ... 258
Environmental variable ... 43, 295
.eoc ... 199
.era ... 94
Error list ... 277, 283, 290, 291, 293
.est ... 67
Execution Procedure ... 49, 55
exit ... 249

F

-f (LC78K0S) ... 270
-f (LK78K0S) ... 183
-f (OC78K0S) ... 215
-f (RA78K0S) ... 129
-f (ST78K0S) ... 81
Fatal error ... 301

G

-g (LK78K0S) ... 164
-g (RA78K0S) ... 107
-ga (RA78K0S) ... 109
-gs (ST78K0S) ... 82

H

help ... 248
.hex ... 199

I

-i (LK78K0S) ... 182
-i (RA78K0S) ... 110
-i (ST78K0S) ... 78
INC78K0S ... 295
Installation ... 38
Intel standard HEX-format ... 201
Internal error ... 301

J

-j (LK78K0S) ... 163
-j (RA78K0S) ... 106

K

-ka (RA78K0S) ... 112
Kanji code ... 44
-kd (LK78K0S) ... 171
-kl (LK78K0S) ... 175
-km (LK78K0S) ... 169
-kp (LK78K0S) ... 173
-ks (RA78K0S) ... 114
-kx (RA78K0S) ... 115

L

-l (LC78K0S) ... 267
LANG78K ... 295
-lf (LB78K0S) ... 235
-lf (LK78K0S) ... 179
-lf (RA78K0S) ... 126
-lh (RA78K0S) ... 121
.lib ... 146, 225
LIB78K0S ... 295
Librarian ... 30, 224

Link Directive ... 149
 Link list file ... 284
 Linker ... 28, 146
 list ... 246
 List converter ... 31, 257
 -ll (LB78K0S) ... 234
 -ll (LK78K0S) ... 177
 -ll (RA78K0S) ... 119
 .lmf ... 146, 199, 258
 Load module file ... 146, 199, 258
 Local symbol list ... 289
 .lst ... 225
 -lt (RA78K0S) ... 124
 -lw (LB78K0S) ... 233
 -lw (RA78K0S) ... 117

M

.map ... 146
 Map list ... 286
 Maximum performance ... 33
 MEMORY ... 150
 Memory Area ... 148
 Memory directive ... 151
 Memory Space ... 148
 MERGE ... 150

N

-ne (LC78K0S) ... 269
 -ne (LK78K0S) ... 180
 -ne (OC78K0S) ... 214
 -ne (RA78K0S) ... 127
 -ng (LK78K0S) ... 164
 -ng (RA78K0S) ... 107
 -nga (RA78K0S) ... 109
 -ngs (ST78K0S) ... 82
 -nj (LK78K0S) ... 163
 -nj (RA78K0S) ... 106
 -nka (RA78K0S) ... 112
 -nkd (LK78K0S) ... 171
 -nkl (LK78K0S) ... 175
 -nkm (LK78K0S) ... 169
 -nkp (LK78K0S) ... 173
 -nks (RA78K0S) ... 114
 -nkx (RA78K0S) ... 115
 -nlf (LB78K0S) ... 235
 -nlf (LK78K0S) ... 179
 -nlf (RA78K0S) ... 126
 -no (LK78K0S) ... 162
 -no (OC78K0S) ... 208
 -no (RA78K0S) ... 105
 -np (LK78K0S) ... 168
 -np (RA78K0S) ... 111
 -nr (OC78K0S) ... 210
 -ns (LK78K0S) ... 165
 -ns (OC78K0S) ... 209
 -nu (OC78K0S) ... 211

O

-o (LC78K0S) ... 268
 -o (LK78K0S) ... 162

-o (OC78K0S) ... 208
 -o (RA78K0S) ... 105
 -o (ST78K0S) ... 79
 Object converter ... 29, 198

P

.p ... 258
 -p (LK78K0S) ... 168
 -p (RA78K0S) ... 111
 Parameter file ... 67, 70, 94, 97, 146, 156, 199, 204, 258, 263
 PATH ... 295
 pick ... 245
 .plk ... 146
 .plv ... 258
 PM+ ... 88, 139, 190, 219, 250, 273, 298, 339
 .poc ... 199
 .pra ... 94
 .prn ... 94, 258
 .pst ... 67
 Public symbol list ... 288

R

-r (LC78K0S) ... 266
 -r (OC78K0S) ... 210
 RAM ... 148
 REGULAR ... 152, 154
 .rel ... 94, 146, 225, 258
 replace ... 243
 ROM ... 148

S

-s (LK78K0S) ... 165
 -s (OC78K0S) ... 209
 Sample program ... 350
 -sc (ST78K0S) ... 75
 Segment location directive ... 153
 -self (RA78K0S) ... 136
 SEQUENT ... 150
 Structured assembler preprocessor ... 26, 66
 .sym ... 199

T

-t (LB78K0S) ... 236
 -t (LK78K0S) ... 185
 -t (RA78K0S) ... 131
 TMP ... 295

U

-u (OC78K0S) ... 211

W

-w (LK78K0S) ... 187
 Warning ... 301
 -wt (ST78K0S) ... 77

Y

-y (LB78K0S) ... 237
-y (LK78K0S) ... 186
-y (OC78K0S) ... 217
-y (RA78K0S) ... 133
-y (ST78K0S) ... 85

Z

-ze (RA78K0S) ... 132
-ze (ST78K0S) ... 84
-zn (RA78K0S) ... 132
-zn (ST78K0S) ... 84
-zs (RA78K0S) ... 132
-zs (ST78K0S) ... 84

详细信息请联系：

中国区

MCU 技术支持热线：

电话：+86-400-700-0606 (普通话)

服务时间：9:00-12:00，13:00-17:00（不含法定节假日）

网址：

<http://www.cn.necel.com/>（中文）

<http://www.necel.com/>（英文）

[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

[成都]

日电电子（中国）有限公司成都分公司

成都市二环路南三段 15 号天华大厦 7 楼 703 室

电话：(+86)28-8512-5224

传真：(+86)28-8512-5334

[长春]

日电电子（中国）有限公司长春分公司

吉林省长春市朝阳区

西安大路 727 号中银大厦 A 座 1609 室

电话：(+86)431-8859-7533 / 8859-8533

传真：(+86)431-8680-2944

[大连]

日电电子（中国）有限公司长春分公司

大连市中山路 88 号天安国际大厦 2701 室

电话：(+86)411-8230-8815 / 8230-8825

传真：(+86)411-8230-8835