

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

RA78K0 Ver. 3.80

汇编包

操作

目标设备
78K0 系列

文档编号: U17199CA1V0UM00 (第 1 版)
发布日期: 2007 年 9 月 CP(K)

© NEC Electronics Corporation 2007

日本印刷

[备忘录]

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

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

i386 是英特尔公司的商标

PC/AT 是国际商用机器公司的商标。

HP9000 series 700 和 **HP-UX** 是惠普公司的商标。

SPARCstation 是 **SPARC** 国际公司的商标。

Solaris 和 **SunOS** 是 **Sun** 微系统公司的商标。

- 本档所刊登的内容有效期截至 **2007 年 9 月**。将来可能未经预先通知而更改。在实际进行生产设计时，请参阅各产品最新的数据表或数据手册等相关资料以获取本公司产品的最新规格。并非所有的产品和/或型号都向每个国家供应。请向本公司销售代表查询产品供应及其他信息。
- 未经本公司事先书面许可，禁止复制或转载本文件中的内容。否则因本档所登载内容引发的错误，本公司概不负责。
- 本公司对于因使用本文件中列明的本公司产品而引起的，对第三者的专利、版权以及其它知识产权的侵权行为概不负责。本文件登载的内容不应视为本公司对本公司或其他人所有的专利、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），交通信号控制设备，防灾装置，防止犯罪装置，各种安全装置以及医疗设备（不包括专门为维持生命而设计的设备）。

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

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

（注）

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

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

区域信息

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

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

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

详细信息请联系：

（中国区）

网址：

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

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

[北京]

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

[深圳]

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

[上海]

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

[香港]

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

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

引言

本手册旨在帮助使用 RA78K0 开发软件的用户了解 RA78K0 系列汇编包（下文简称“RA78K0”）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 RA78K0 中使用的源程序。因此，在阅读本手册前，请阅读 RA78K0 汇编包语言用户手册(U17198E)（下文简称“语言”）

本手册的内容是针对 RA78K0 的 Ver. 3.80 或更高版本的。

[读者对象]

RA78K0 适用于了解要开发的微处理器（78K0 系列）的功能和指令的用户。

[组织结构]

本手册由以下 12 章及附录组成。

第 1 章 概述

概括介绍 RA78K0 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 RA78K0 提供的程序文件名与操作环境。

第 3 章 RA78K0 的执行过程

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 RA78K0 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 链接器

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 RA78K0

介绍一些 RA78K0 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 RA78K0 的注意事项。

本手册未详细介绍指令集。

有关指令的内容，请参阅要开发的微处理器的用户手册。

[手册使用方法]

建议首次使用汇编的读者从**第 1 章 概述**开始阅读。那些对汇编器具有基本了解的读者可以跳过本章。
在使用 RA78K0 前，请阅读**第 3 章 RA78K0 的执行过程**。
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 700TM 或 SPARCstation™ 系列时，请注意它们的以下区别。

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

[约定]

本手册中使用了以下符号及缩写。

∴	表示同一表达式被重复
[]:	方括号中的项目可以省略。
'':	'' (引号) 中的字符将按照它们显示的内容列出。
< >:	表示窗口名或对话框名。
" ":	" " (双引号) 中的字符是要求读者参考的章节、段落或图标的标题。
___:	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		文档编号
RA78K0 3.80 版本汇编器包	操作	本手册
	语言	U17198E
	结构化汇编语言	U17197E
CC78K0 3.70 版本 C 编译器	操作	U17201E
	语言	U17200E
SM+系统模拟器	操作	U17200E
	用户界面	U17246E
SM78K0 系列 2.52 版本系统模拟器	操作	U16768E
PM+5.20 版本		U16934E
ID78K0-NS 2.52 版本集成调试器	操作	U16488E
ID78K0-QB 2.81 版本 集成调试器	操作	U16996E
78K0 系列	指令	U12326E

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。

[备注]

目录

第 1 章 概述	17
1.1 汇编器概述.....	17
1.1.1 什么是汇编器?	18
1.1.3 什么是重定位汇编器?	22
1.1.4 重定位汇编器的优势.....	22
1.2 RA78K0 功能概述.....	24
1.2.1 使用编辑器创建一个源模块文件.....	25
1.2.1 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
1.2.6 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 RA78K0 的最高性能指标.....	33
1.4 RA78K0 的功能.....	36
第 2 章 产品概况与安装	37
2.1 主机与提供介质.....	37
2.2 安装.....	38
2.2.1 Windows 版本的安装.....	38
2.3 安装设备文件.....	40
2.3.1 Windows 版本的安装.....	40
2.3.2 UNIX 版本的安装.....	40
2.3.3 设备文件的注册表注册.....	40
2.4 目录配置.....	41
2.4.1 Windows 版本目录配置.....	41
2.4.2 UNIX 版本目录配置.....	42
2.5 卸载步骤.....	43
2.5.1 Windows 版本的卸载.....	43
2.5.2 UNIX 版本的卸载.....	43
2.6 环境设置.....	44
2.6.2 环境变量.....	44
2.6.3 源文件中的汉字代码.....	45
第 3 章 RA78K0 的执行过程	46
3.1 准备执行RA78K0.....	46
3.1.1 示例程序.....	46
3.1.2 示例程序的结构.....	49
3.2 RA78K0 的执行过程.....	50
3.3 ST78K0 的执行步骤.....	56
3.4 从命令行中完成汇编、链接与目标程序转换 (DOS提示, UNIX)	62
3.5 使用参数文件.....	66
第 4 章 结构化汇编器预处理器	67
4.1 结构化汇编器的I/O文件.....	67
4.2 结构化汇编器的功能.....	68
4.3 结构化汇编器启动.....	69
4.3.1 结构化汇编器启动.....	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 结构化汇编器选项说明	73
4.5 在PM+中设置选项	88
4.5.1 设置选项方法	88
4.5.2 选项	90
4.5.3 编辑选项对话框	92
第5章 汇编器	93
5.1 汇编器的I/O文件	93
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 从一个参数文件启动	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 汇编器选项类型	100
5.4.2 汇编器选项的优先级	102
5.4.3 汇编器选项说明	103
5.5 在PM+中设置选项	140
5.5.1 设置选项	140
5.5.2 选项	142
第6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	148
6.4 链接指令	149
6.4.1 指令文件	149
6.4.2 存储器指令	151
6.4.3 程序段位置指令	153
6.5 启动链接器	156
6.5.1 启动链接器	156
6.5.2 执行开始结束信息	157
6.6 链接器选项	159
6.6.1 链接器选项的类型	159
6.6.2 链接器选项的先后顺序	161
6.6.3 链接器选项说明	161
6.7 在PM+中设置选项	194
6.7.1 选项设置方法	194
6.7.2 选项	197
6.7.3 编辑选项对话框	200
第7章 目标转换器	201
7.1 目标转换器的I/O文件	201
7.2 目标转换器的功能	203
7.3 目标转换器的启动	218
7.3.1 目标转换器的启动	218
7.3.2 执行开始结束信息	219
7.4 目标转换器选项	221
7.4.1 目标转换器选项的类型	221
7.4.2 目标转换器选项说明	222
7.5 在PM+中设置选项	237
7.5.1 选项设置方法	237

7.5.2 选项	239
第 8 章 库管理程序.....	241
8.1 库管理程序的I/O 文件	241
8.3 库管理程序的启动	245
8.3.1 库管理程序的启动.....	245
8.3.2 库管理程序执行开始和执行结束信息.....	248
8.4 库管理程序选项.....	249
8.4.1 库管理程序选项类型.....	249
8.4.2 库选项的说明	249
8.5 子命令	257
8.5.1 子命令类型	257
8.5.2 子命令说明	257
8.6 在PM +中设置选项.....	268
8.6.1 选项设定方法	268
8.6.2 选项设定	270
8.7.2 选项设置.....	273
第 9 章 列表转换器.....	275
9.1 列表转换器I/O文件	275
9.2 列表转换器功能.....	277
9.3 列表转换器的启动	280
9.3.1 列表转换器的启动.....	280
9.3.2 列表转换器执行开始和结束信息.....	282
9.4 列表转换器选项	283
9.4.1 列表转换器选项类型.....	283
9.4.2 列表转换器选项的说明	283
9.5 在PM +中设置选项.....	291
9.5.1 选项设置方法	291
9.5.2 选项.....	293
第 10 章 程序输出列表.....	294
10.1 结构化汇编器预处理器输出的列表	294
10.2 汇编器输出列表	296
10.2.1 汇编列表文件标题	297
10.2.2 汇编列表.....	298
10.2.3 符号列表.....	300
10.2.4 交叉参考列表	301
10.2.5 错误列表.....	303
10.3 链接器输出列表	304
10.3.1 连接列表文件标题	304
10.3.2 映像列表.....	306
10.3.3 公共符号列表	308
10.3.4 局部符号列表	309
10.3.5 错误列表.....	310
10.4 目标转换器输出列表	311
10.4.1 错误列表.....	311
10.5 库管理程序输出列表	312
10.5.1 库数据输出列表	312
10.6 列表转换器输出列表	313
10.6.1 绝对汇编列表	313
10.6.2 错误列表.....	313
第 11 章 RA78K0 的高效使用.....	314

11.1 提高操作效率 (EXIT 状态功能)	314
11.2 准备软件开发环境 (环境变量)	316
11.3 中断程序执行	317
11.4 使汇编列表易于读取	318
11.5 减少程序的启动时间	319
11.5.1 在源程序中指定控制指令	319
11.5.2 使用 PM+	319
11.5.3 创建参数文件和子命令文件	320
11.6 目标模块库生成	321
第 12 章 错误消息	322
12.1 错误消息概述	322
12.2 结构化汇编器错误消息	323
12.3 汇编器错误消息	329
12.4 连接器错误信息	340
12.5 目标转换程序错误消息	349
12.6 库管理程序错误消息	353
12.7 表转换程序错误消息	357
12.8 PM+错误消息	361
12.8.1 结构化汇编器预处理器 (ST78K0)	361
12.8.2 汇编程序 (RA78K0)	364
12.8.3 连接器 (LK78K0)	367
12.8.4 目标转换器 (OC78K0)	370
12.8.5 库管理程序 (LB78K0)	371
12.8.6 列表转换器 (LCNV78K0)	372
附录A 示例程序	373
A.1 K0main.asm	374
A.2 K0ssub.asm	375
A.3 test1.s	376
A.4 test2.s	377
A.5 testinc.s	378
A.6 st.bat	379
附录B 使用注意事项列表	380
附录C 选项列表	385
C.1 结构化汇编器选项	385
C.2 汇编器选项	387
C.3 连接器选项列表	390
C.4 目标转换器选项	393
C.5 库管理程序选项	395
C.6 表转换程序选项	396
附录D 子命令列表	397
附录E 索引	398

插图列表

插图编号 插图标题	页码
图 1-1 RA78K0 汇编包.....	17
图 1-2 RA78K0 汇编包.....	18
图 1-3 应用微处理器产品开发流程.....	19
图 1-4 软件开发流程.....	20
图 1-5 RA78K0 汇编流程.....	21
图 1-6 重汇编以便于调试.....	23
图 1-7 使用现有模块的程序开发.....	23
图 1-8 使用RA78K0 开发程序的步骤.....	24
图 1-9 创建源模块文件.....	25
图 1-10 结构化汇编预处理程序功能.....	26
图 1-11 汇编器功能.....	27
图 1-12 链接器功能.....	28
图 1-13 目标转换器功能.....	29
图 1-14 库管理程序功能.....	30
图 1-15 表转换程序功能.....	31
图 1-16 调试程序功能.....	32
图 2-1 目录配置.....	41
图 2-2 目录配置.....	42
图 3-1 源程序的结构.....	46
图 3-2 RA78K0 执行过程 1.....	54
图 3-3 RA78K0 执行过程 2.....	55
图 3-4 ST78K0 执行过程.....	61
图 3-5 链接指示.....	63
图 4-1 结构化汇编器的I/O文件.....	67
图 4-2 <Structured Assembler Options>对话框（当选中 <<Output>>标签时）.....	88
图 4-3 <Assembler Source Options>对话框（当选中 [Assembler Options]标签时）.....	88
图 4-4 <Structured Assembler Options>对话框（当选中 <<Others>>标签时）.....	89
图 4-5 <Edit Option>对话框.....	92
图 4-6 <Add Option>对话框.....	92
图 5-1 汇编器的I/O文件.....	94
图 5-2. <Assembler Options> 对话框（选择 <<Output1>> 标签时）.....	140
图 5-3. <Assembler Options> 对话框（选择 <<Output2>> 标签时）.....	141
图 5-4. <Assembler Options> 对话框（选择 <<Others>> 标签时）.....	141
图 5-5 <Edit Option>对话框.....	145
图 5-6 <Add Option>对话框.....	145
图 6-1 存储区名.....	151
图 6-2 程序段分配的具体例子.....	155
图 6-3 < Linker Options >对话框（当选择<< Output 1>>标签时）.....	194
图 6-4 < Linker Options >对话框（当选择<< Output 2>>标签时）.....	195
图 6-5 < Linker Options >对话框（当选择<<Library>>标签时）.....	195
图 6-6 < Linker Options >对话框（当选择<<Others>>标签时）.....	196
图 7-1 目标转换器的I/O文件.....	202
图 7-2 Intel标准格式.....	205
图 7-3 英特尔扩展格式.....	206
图 7-4 摩托罗拉S型格式.....	212
图 7-5 符号值的格式.....	217
图 7-6 <Object Converter Options> 对话框 (当选择 <<Output1>> 标签时).....	237

图 7-7 <Object Converter Options> 对话框（当选择 <<Output2>> 标签时）	238
图 7-8 <Object Converter Options> 对话框（当选择 <Others> 标签时）	238
图 8-1 库管理程序的I/O文件.....	242
图 8-2 库文件创建过程步骤.....	244
图 8-3 < Librarian Options >对话框（当选择<< Output >>标签时）	268
图 8-4 < Librarian Options >对话框（当选择<< Others >>标签时）	269
图 8-5. <<Library File Name> 对话框	272
图 8-6. < Subcommand > 对话框	273
图 9-1. 列表转换器I/O文件	276
图 9-2. <List Converter Options> 对话框 (当选择了 <<Output>> 标签时).....	291
图 9-3 <List Converter Options> 对话框 (当选择了<<Others>> 标签时).....	292
图形B-1 地址一览	382

表格列表

表格编号 表格标题	页码
表 1-1 结构化汇编器的最高性能.....	33
表 1-2 汇编器的最高性能.....	34
表 1-3 链接器最佳性能.....	35
表 2-1 提供介质与汇编包的记录格式.....	37
表 4-1 结构化汇编器的I/O文件.....	67
表 4-2 结构化汇编器选项.....	73
表 5-1 汇编器的I/O文件.....	93
表 5-2 汇编器选项.....	100
表 5-2 汇编器选项.....	101
表 5-3 汇编器选项的优先级.....	102
表 5-4 可作为标题写入的字符.....	122
表 5-4 可作为标题写入的字符.....	123
表 6-1 链接器的I/O文件.....	146
表 6-2 程序段分配组（外部ROM等）.....	148
表 6-3 指令类型.....	149
表 6-4 存储区名指定及存储空间名共同确定的程序段位置.....	154
表 6-5 链接器选项.....	159
表 6-5 链接器选项.....	160
表 6-6 链接器选项的先后顺序.....	161
表 7-1 目标转换器的I/O文件.....	201
表 7-2: 扩展空间的输出文件类型.....	203
表 7-4 扩展技术标题字段.....	208
表 7-5 校验和评估的字符值.....	208
表 7-6 扩展技术的数据块格式.....	209
表 7-7 扩展技术的终止块格式.....	209
表 7-8 扩展技术的符号块格式.....	210
表 7-9 扩展技术的符号块段定义字段.....	211
表 7-10 扩展技术的符号块段定义字段.....	211
表 7-11 摩托罗拉十六进制文件记录类型.....	212
表 7-12 每个记录的一般格式.....	212
表 7-13 字段和含义.....	213
表 7-14 符号属性的值.....	217
表 7-15 目标转换器选项.....	221
表 7-16 在指定了-ZF选项时的文件类型.....	223
表 7-17 扩展空间十六进制格式对象模型文件的类型.....	224
表 8-1. 库管理程序的I/O文件.....	241
表 8-2. 库管理程序选项.....	249
表 8-3 子命令.....	257
表 9-1 列表转换器I/O文件.....	275
表 9-2 当列表转换器启动时指定的文件类型.....	280
表 9-3 列表转换器选项.....	283
表 10-1 结构化汇编器预处理器输出的列表.....	294
表 10-2 错误列表输出项目的说明（当结构化汇编器预处理器启动时）.....	295
表 10-3 汇编器输出列表.....	296
表 10-4 汇编列表文件标题各输出项的说明.....	297
表 10-5 汇编列表各输出项的说明.....	298
表 10-6 符号列表各输出项的说明.....	300

表 10-7 交叉参考列表各输出项的说明	301
表 10-8 错误列表输出项目的说明（当汇编器启动时）	303
表 10-9 链接器输出的列表	304
表 10-10 连接列表文件标题各输出项的说明.....	304
表 10-11 连接映像列表各输出项的说明	306
表 10-12 连接公共符号列表各输出项的说明.....	308
表 10-13 局部公共符号列表各输出项的说明.....	309
表 10-14 错误列表各输出项的说明（当链接器启动时）	310
表 10-15 目标转换器各输出项的说明	311
表 10-16 库管理程序输出的列表.....	312
表 10-17 库数据输出列表各输出项的说明	312
表 10-18 列表转换器输出的列表.....	313
表 12-1 结构化汇编器错误消息.....	323
表 12-2 汇编器错误消息	329
表 12-3. 连接器错误信息	340
表 12-4 目标转换程序错误消息.....	349
表 12-5 库管理程序错误消息.....	353
表 12-6 表转换程序错误消息.....	357
表 12-7 DLL中显示的结构化汇编器(ST78K0)的错误消息.....	361
表 12-8 汇编程序（RA78K0）DDL 错误信息	364
表 12-9 连接器（LK78K0）DLL显示的错误信息.....	367
表 12-10 目标转换器（OC78K0）DLL显示的错误信息	370
表 12-11 库管理程序（LB78K0）DLL显示的错误信息	371
表 12-12 库管理程序（LVNV78K0）DLL显示的错误信息.....	372
表B-1 输出格式为“Intel扩展HEX”的示例（flash存储器实地址）	383
表B-2 输出格式为“Intel扩展HEX”的示例（BANK数字+CPU地址）	384
表C-1 结构化汇编器选项.....	385
表C-2 结构化汇编器选项.....	387
表C-3 连接器选项列表.....	390
表C-4 目标转换器选项.....	393
表C-5 库管理选项列表.....	395
表C-6 表转换程序选项.....	396
表D-1 子命令列表	397

第 1 章 概述

本章介绍 RA78K0 在微处理器软件开发中的作用及其功能。

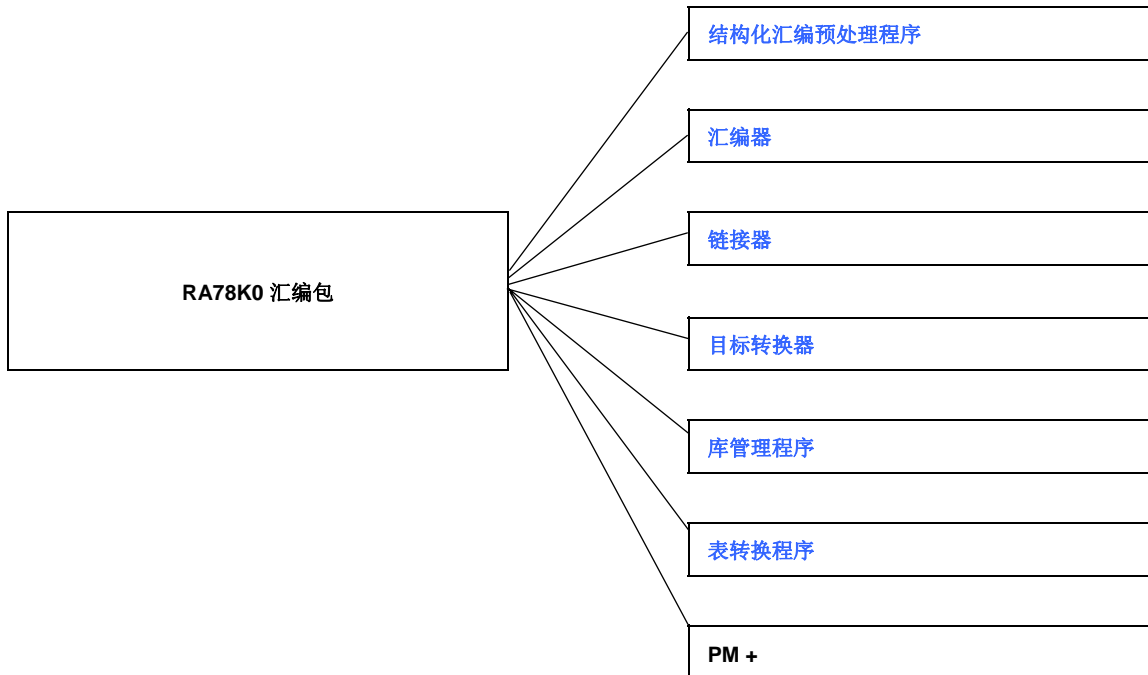
1.1 汇编器概述

RA78K0 汇编包是对一系列程序的统称，这些程序用于将用汇编语言为 78K0 系列微处理器编写的源程序翻译成机器语言代码。

RA78K0 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，RA78K0 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 RA78K0 汇编包



1.1.1 什么是汇编器?

(1) 汇编语言与机器语言

汇编语言是处理器最基本的编程语言。

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

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

图 1-2 RA78K0 汇编包

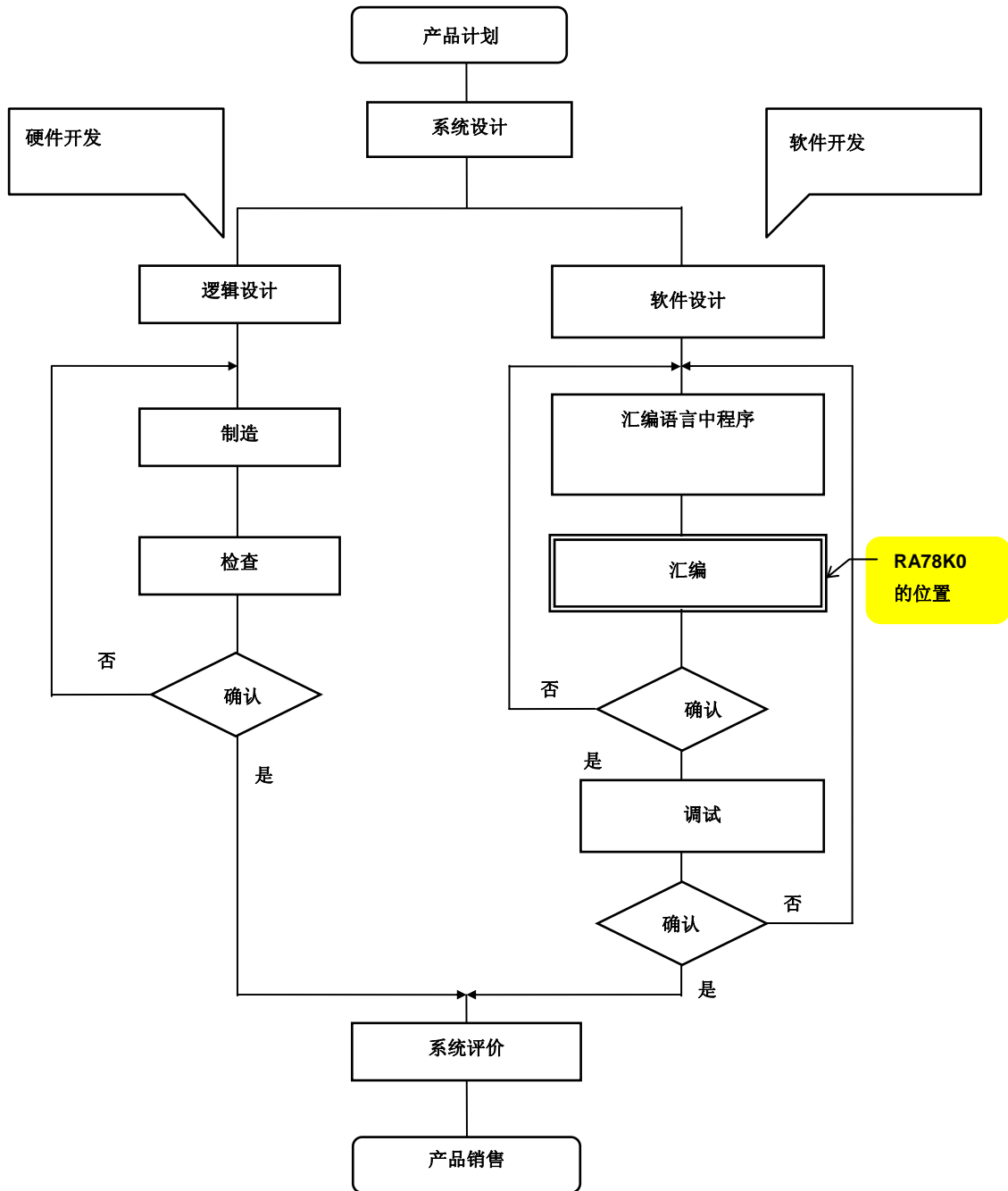


1.1.2 RA78K0 的开发过程

(1) 微处理器相关产品开发与 RA78K0 的作用

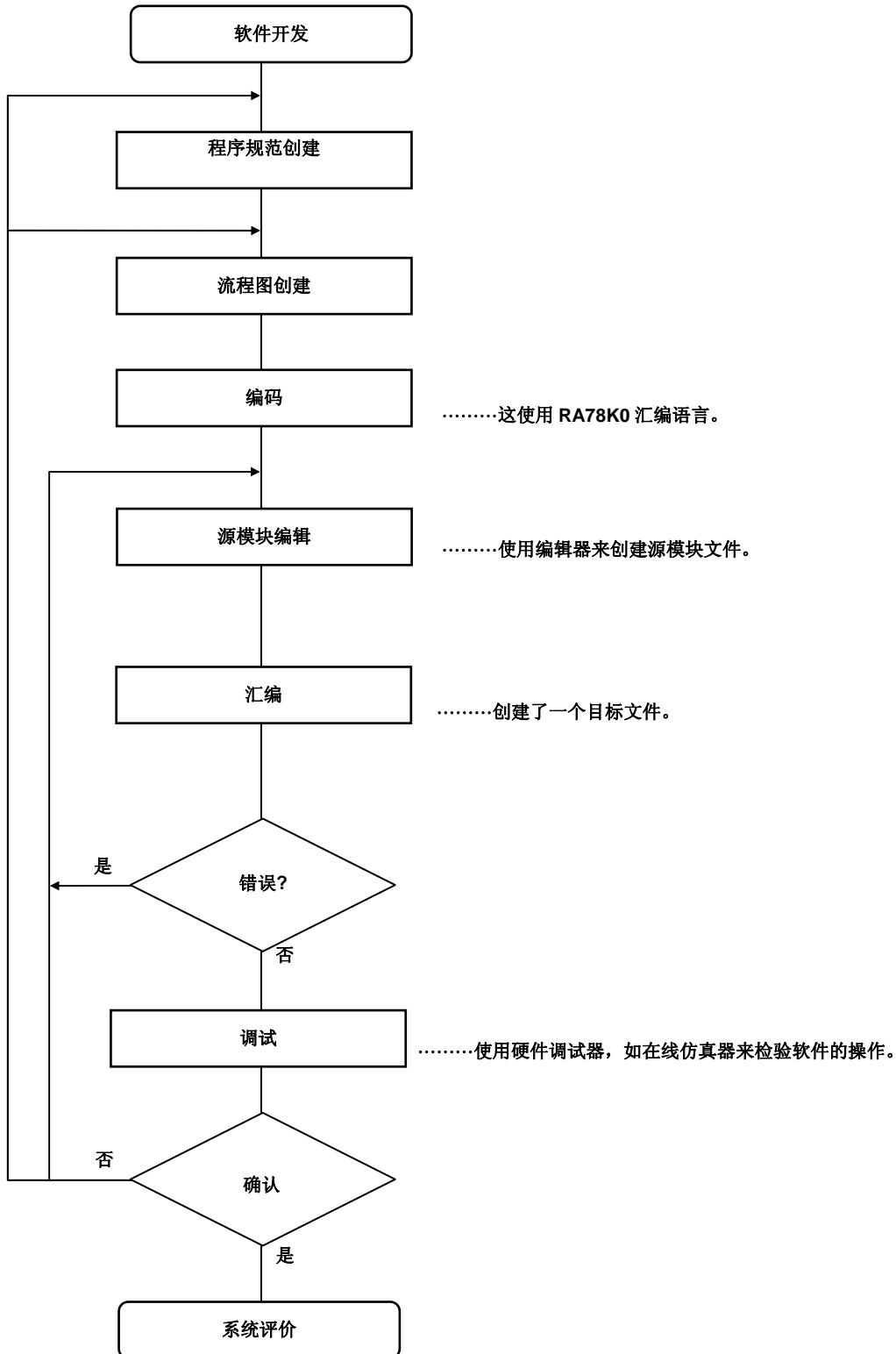
图 1-3 应用微处理器产品开发流程说明了汇编语言编程在（软件）产品开发流程中的地位。

图 1-3 应用微处理器产品开发流程



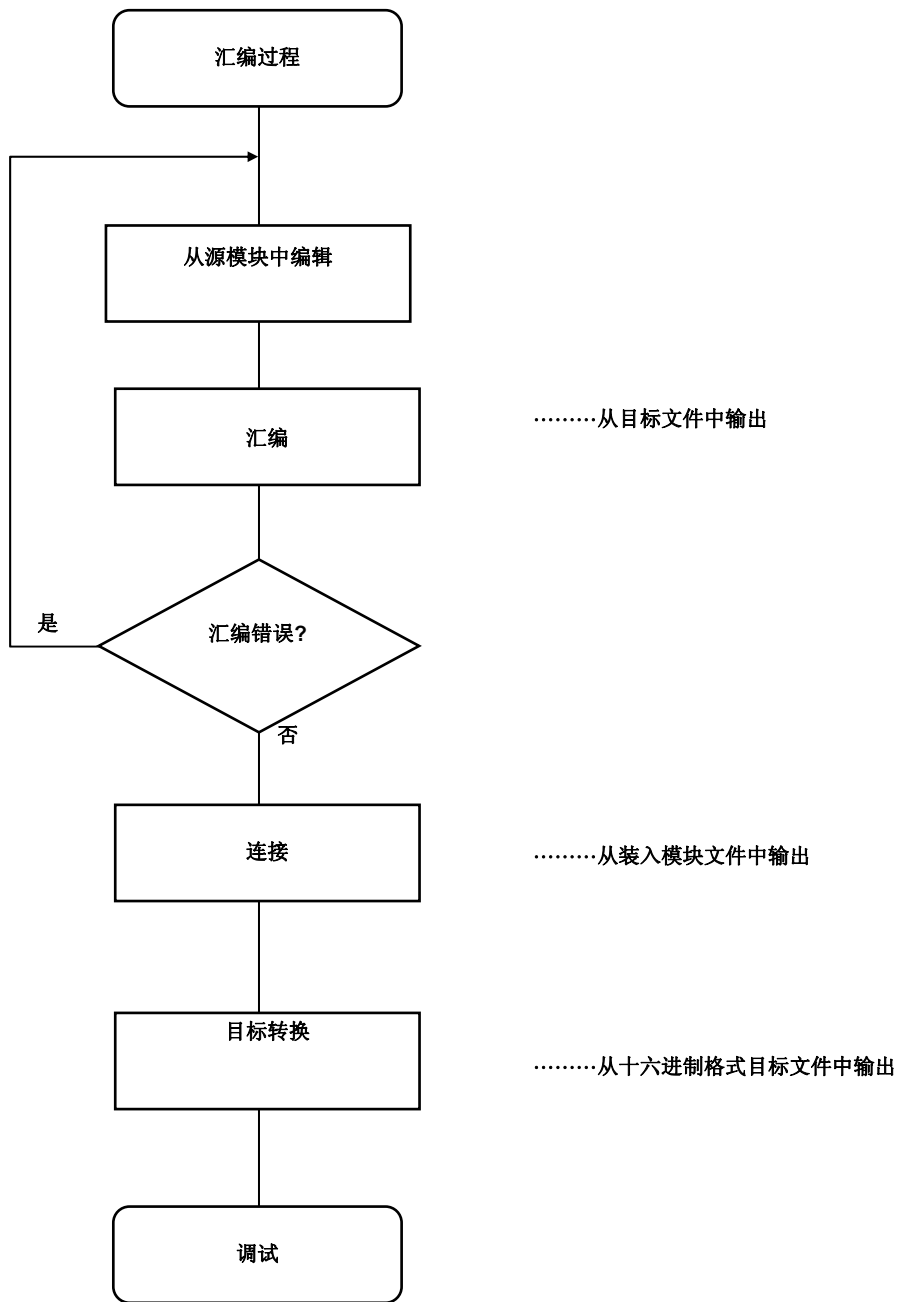
关于软件开发流程更详细的说明，请参见图 1-4 软件开发流程。

图 1-4 软件开发流程



因此，RA78K0 适用于汇编过程。

图 1-5 RA78K0 汇编流程



1.1.3 什么是重定位汇编器？

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，说明每条机器语言指令存储在内存中的什么位置。

根据定位机器语言指令地址的方法，汇编器可以大致分为绝对汇编器和重定位汇编器。

- **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。

- **重定位汇编器**

在重定位汇编器中，将从汇编语言汇编的机器语言指令定位到暂时地址。绝对地址由称为链接器的程序来确定。

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

1.1.4 重定位汇编器的优势

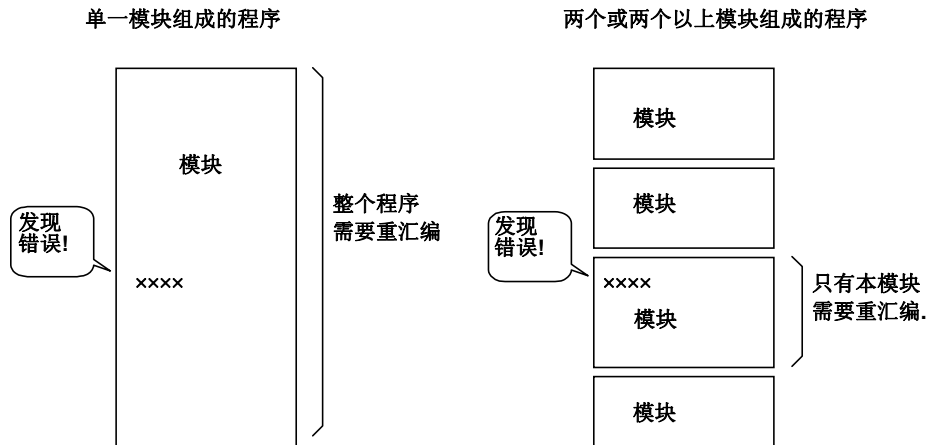
重定位汇编器是一种适于模块化编程的汇编器。使用重定位汇编器进行模块化编程具有以下优点：

- (1) **提高开发效率**

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

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

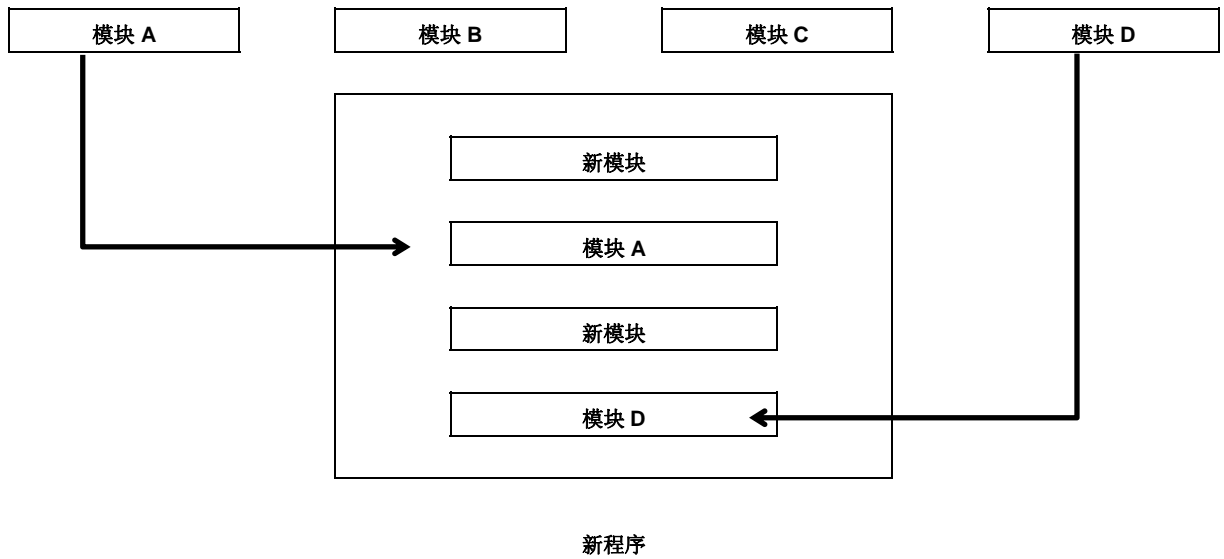
图 1-6 重汇编以便于调试



(2) 使用资源

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

图 1-7 使用现有模块的程序开发

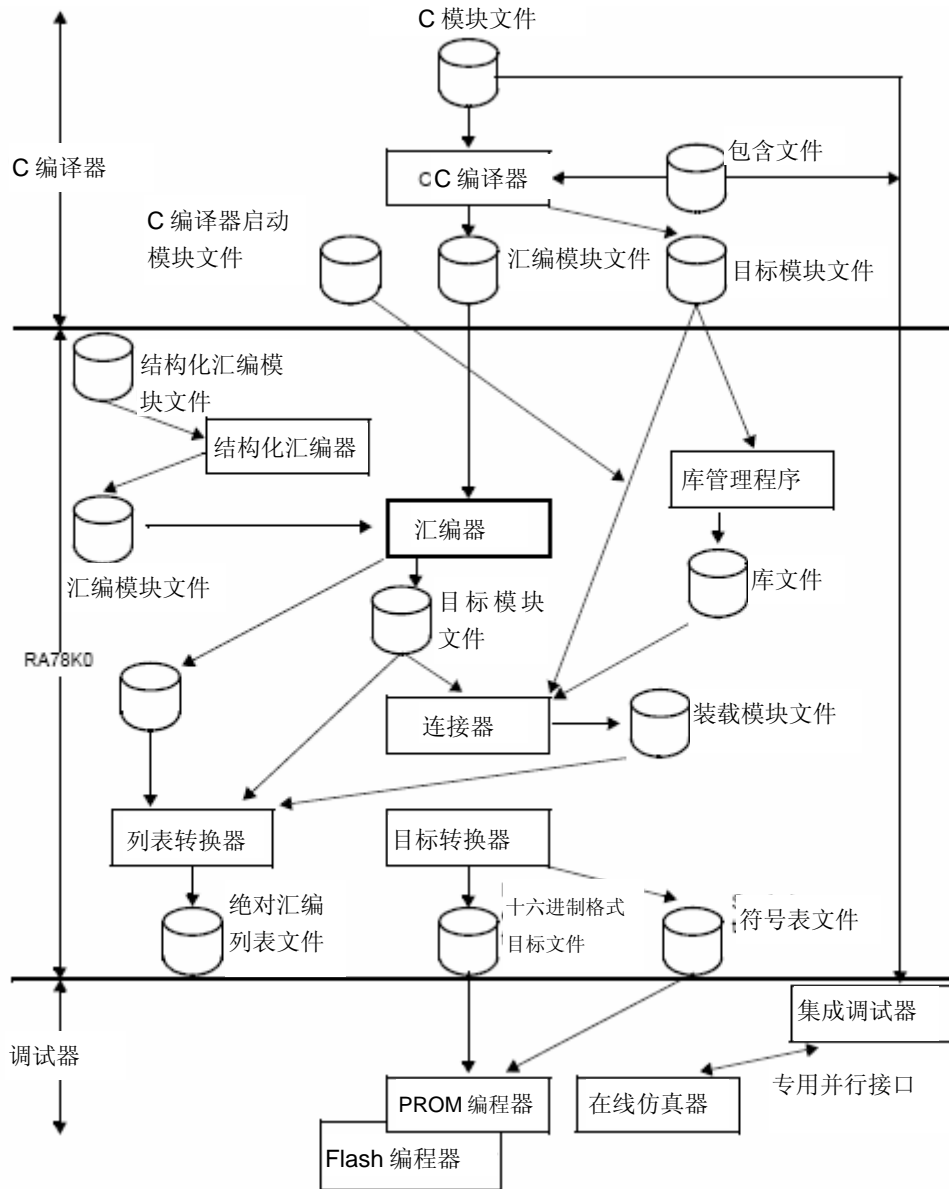


1.2 RA78K0 功能概述

有关开发通用程序的步骤，请参见图 1-8 使用 RA78K0 开发程序的步骤。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、目标转换器及其它程序统称为“RA78K0”。汇编器程序简称“汇编器”。

图 1-8 使用 RA78K0 开发程序的步骤



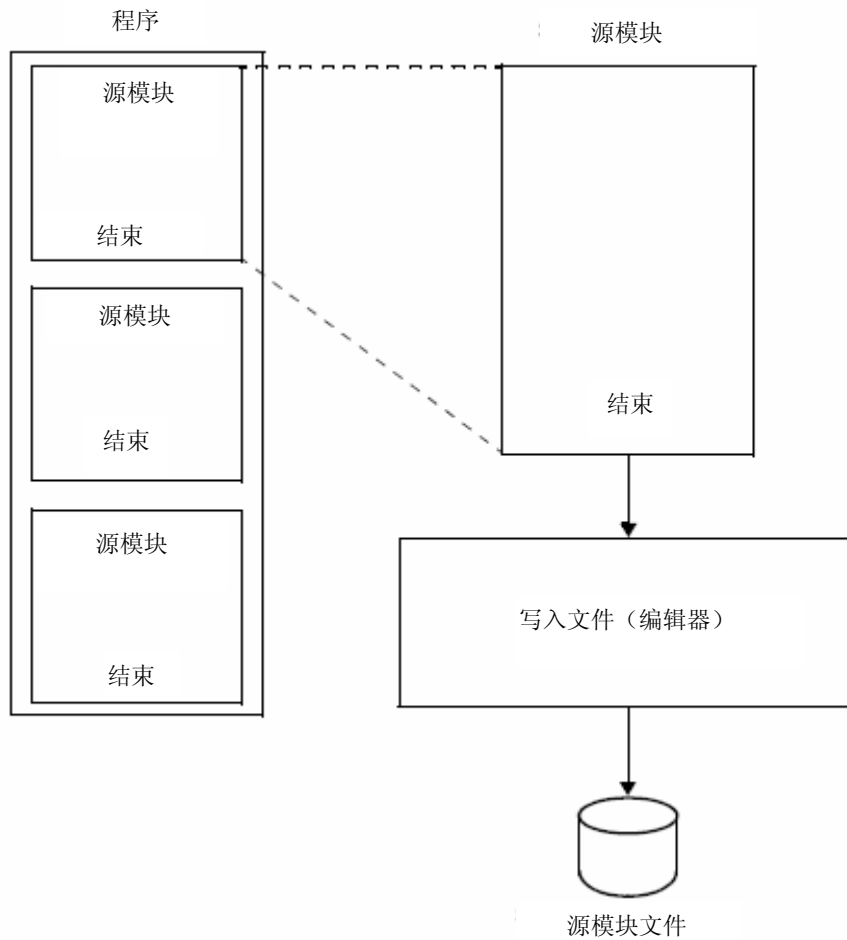
1.2.1 使用编辑器创建一个源模块文件

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

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

源模块文件用作一个汇编器输入文件。

图 1-9 创建源模块文件

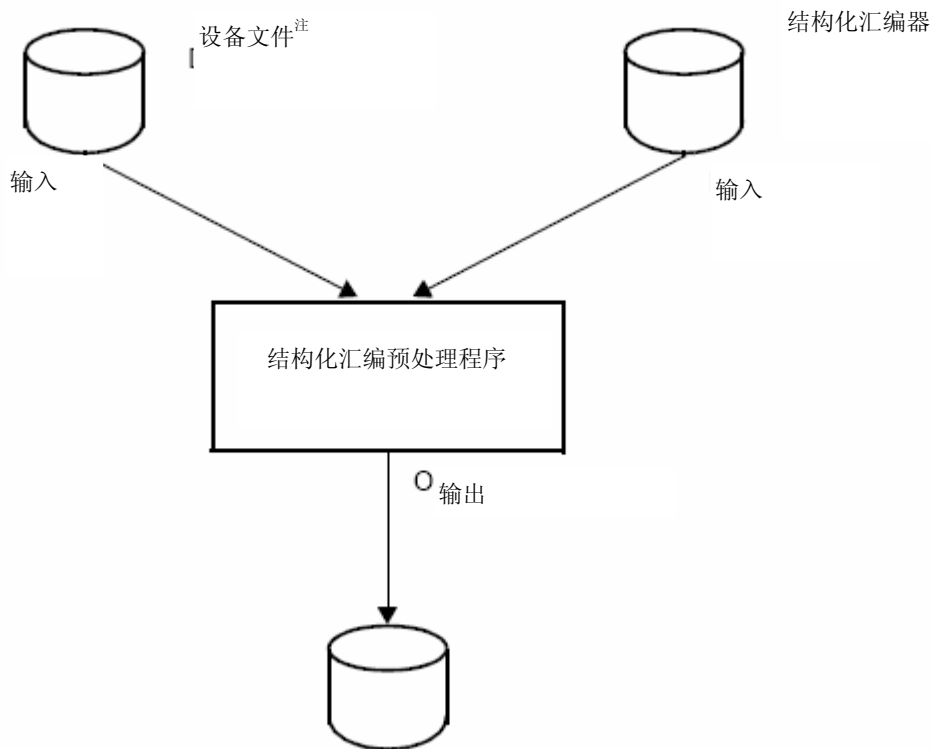


1.2.2 结构化汇编预处理程序

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

有关结构化汇编预处理程序及结构化汇编语言的情况，请参阅文档 RA78K0 结构化汇编语言用户手册。

图 1-10 结构化汇编预处理程序功能



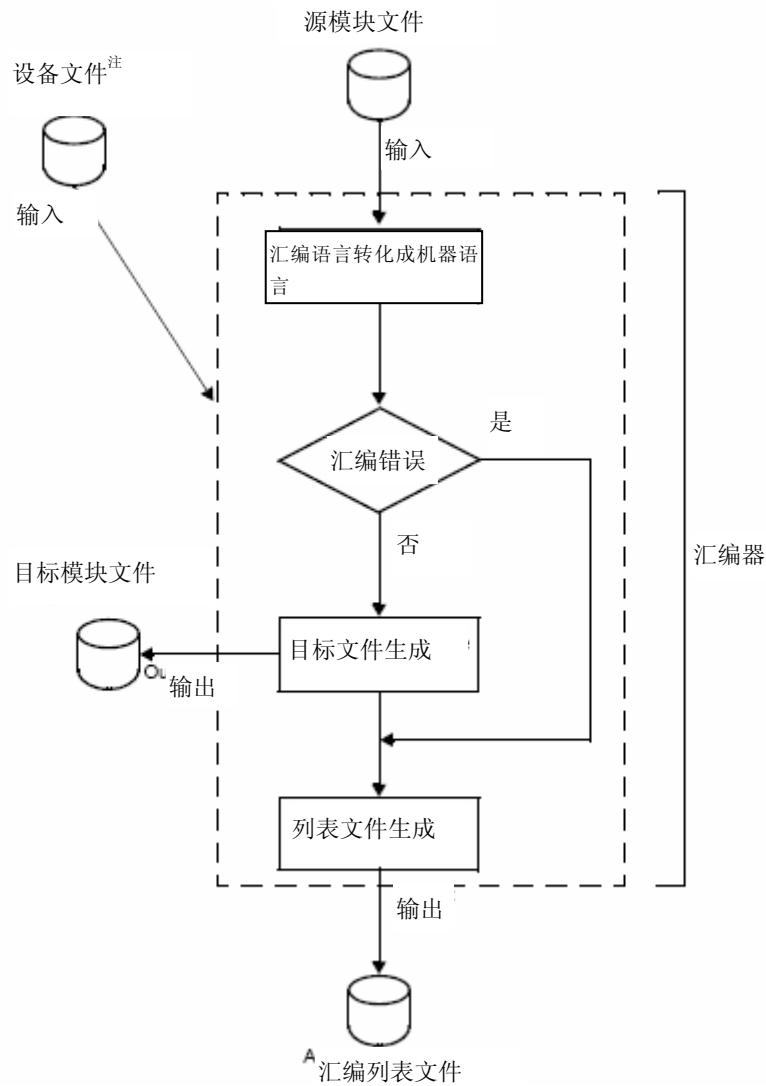
注：您可以通过以下网址从在线传输服务（ODS）上下载设备文件

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

1.2.3 汇编器

汇编器是这样—个程序，它输入源模块文件，并将汇编语言转换为—组二进制指令（机器语言）。如果汇编器在源模块中发现描述错误，它将输出—个汇编错误。如果未发现汇编错误，汇编器将输出—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—个汇编列表文件输出。

图 1-11 汇编器功能



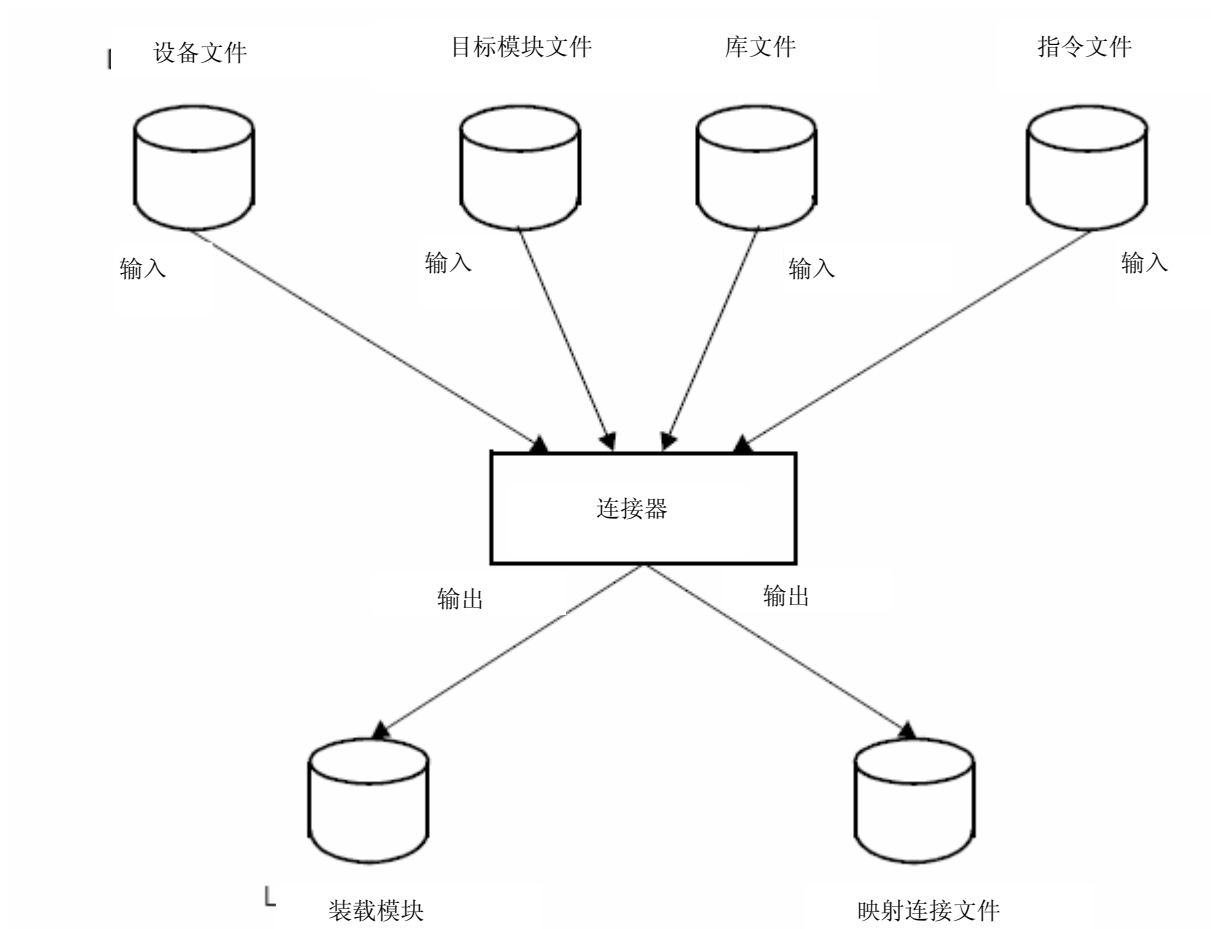
注 通过从在线传输(ODS)服务系统下载该设备，网址为：
<http://www.necel.com/micro/ods/eng/tool/DeviceFile/list.html>

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装入模块文件（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重定位段的单元地址。这样就确定了重定位符号及外部引用符号的值，以便正确的值被嵌入到装入模块文件中。

图 1-12 链接器功能



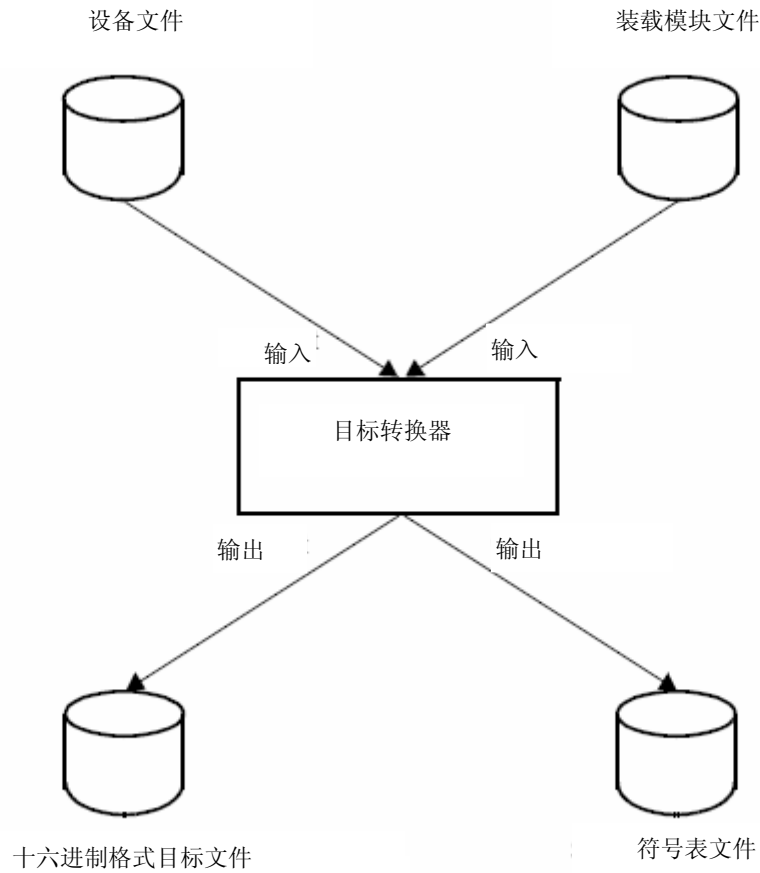
注 通过从在线传输(ODS)服务系统下载该设备，网址为：
<http://www.necel.com/micro/ods/eng/tool/DeviceFile/list.html>

1.2.5 目标转换器

目标转换器输入链接器输出的装入模块文件，并转换文件格式。生成的文件将作为十六进制格式的目标文件输出。

目标转换器还以符号表文件输出符号调试所需的符号数据。

图 1-13 目标转换器功能



注 通过从在线传输(ODS)服务系统下载该设备，网址为：
<http://www.necel.com/micro/ods/eng/tool/DeviceFile/list.html>

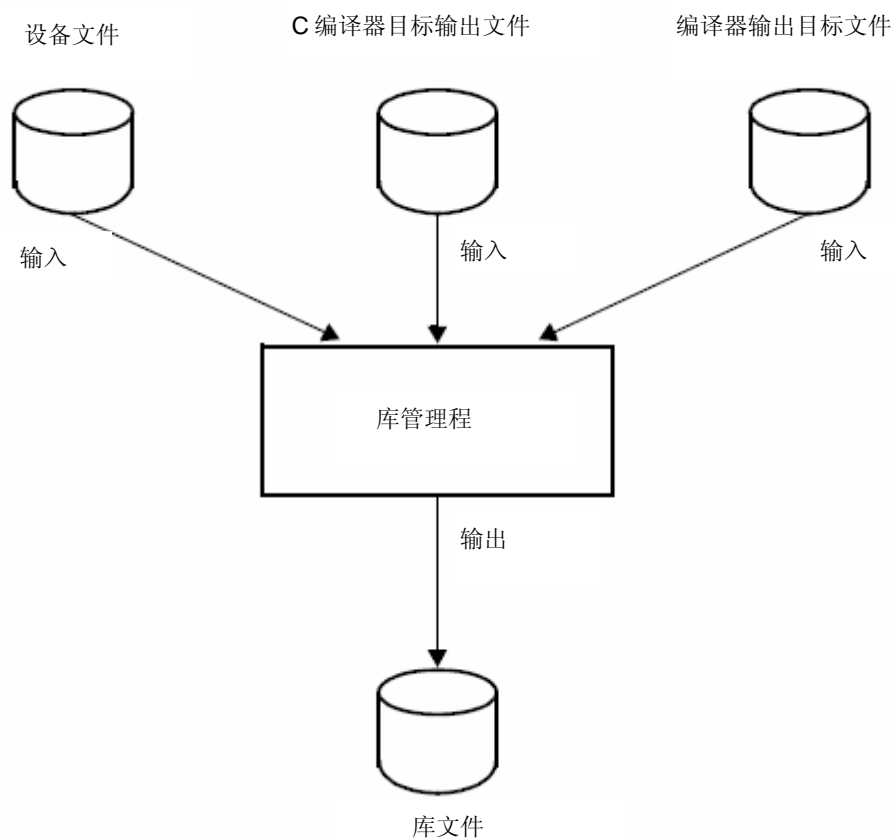
1.2.6 库管理程序

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，即在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 库管理程序功能



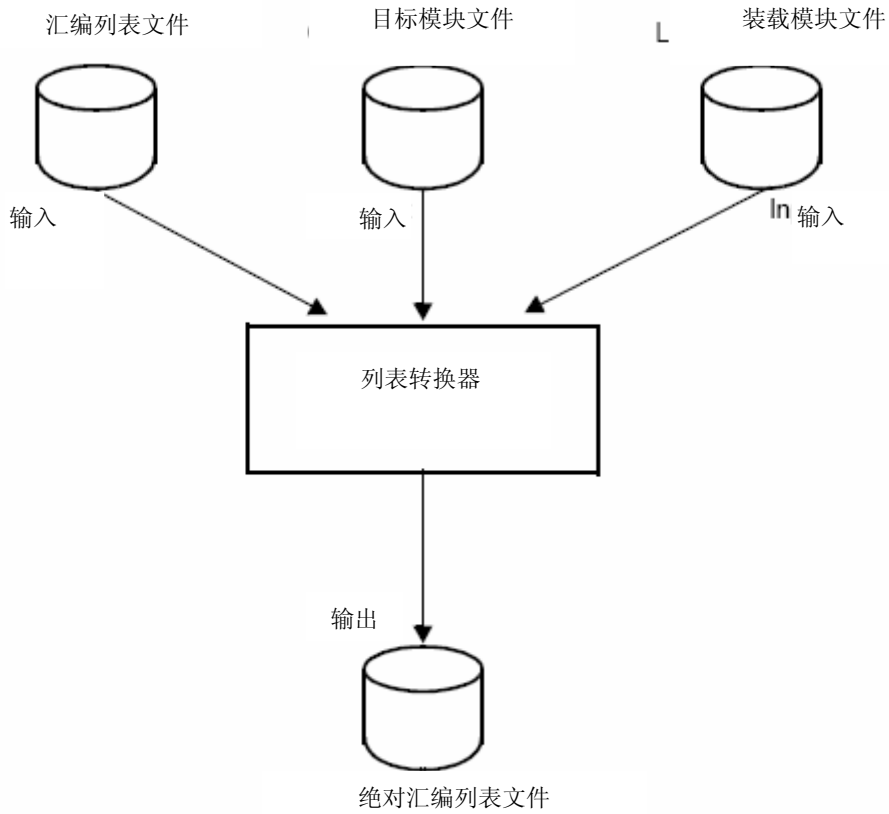
注 通过从在线传输(ODS)服务系统下载该设备，网址为：
<http://www.necel.com/micro/ods/eng/tool/DeviceFile/list.html>

1.2.7 表转换程序

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

重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 表转换程序功能



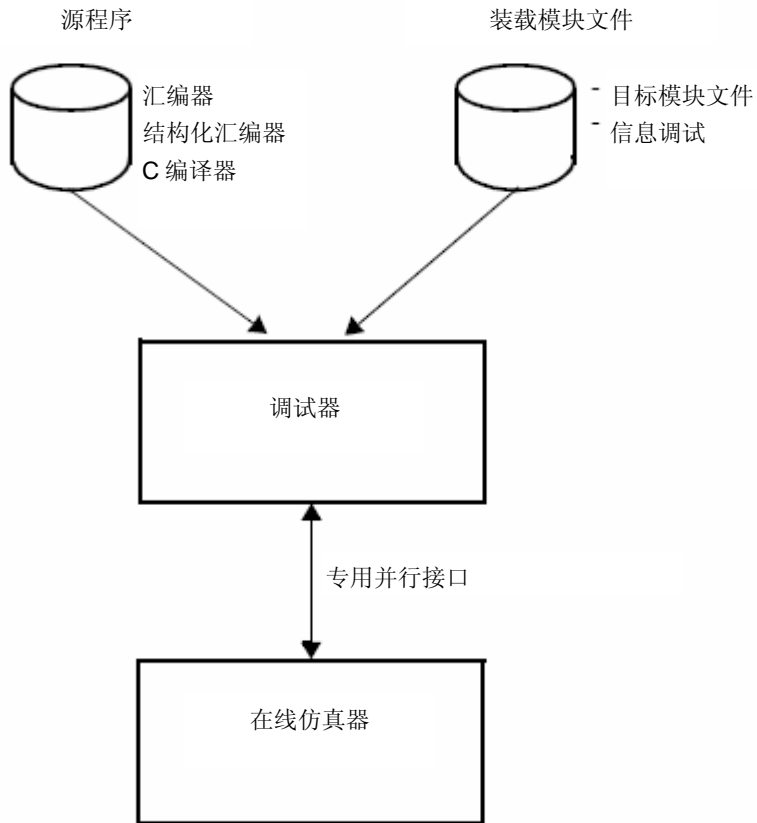
1.2.8 调试器

78K0 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装入模块文件下载到目标系统的在线仿真器（IE）。它还可以读出源程序文件，从而在源程序级执行调试。

调试器与 IE 是不同的程序包，与 RA78K0 分开出售。

图 1-16 调试程序功能



1.3 程序开发前的提示

在开发程序前，请注意以下几点。

1.3.1 RA78K0 的最高性能指标

(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 字符
在#define 指示中注册的符号数(不包括保留字)	512 符号
在#define 指示中注册的符号长度	31 字符
控制语句的嵌套级数	31 级
#ifdef 指示中的嵌套级数	8 级
#defcallt 指示	32
#include 指示的嵌套	不支持
被#define 指示重定义的次数	31 次
在一个序列中被赋值的操作数的个数	33 ^{注意 1}
逻辑操作符的操作数	17 ^{注意 2}
被-D 选项定义的符号数	30
被-I 选项指定的 INCLUDE 文件路径数	64

注意 1.最大值表示如下：

$S1=S2= \dots S32=S33$

最多可插入 33 个符号，其中包括 32 个等号 (=)。

注意 2.最大值表示为：

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“||”号。

(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(局部+公共)	65535
可以输出至交叉引用列表的符号数	65534 注意 1
单一宏可引用的最大宏体大小	1MB
所有宏体的总大小	10MB
单一文件中的段数	256
单一文件中的宏及 include 指定	10000
单一 include 文件中的宏及 include 指定	10000
重定位信息注意 2	65535
行数信息	65535
单一文件中的类 BR 指示	32767
一行代码的字符数	2048 注意 3
符长	256 字节
交换名定义数注意 4	1000
交换名字符长注意 4	31 字符
段名字符长	8 字符
模块名字符长(类 NAME 指示)	8 字符
类 MACRO 指示中虚参数	16
引用实参数	16
类 IRP 指示中实参数	16
宏体中本地符号数	64
扩展宏中的本地符号总数	65535
宏嵌套级数(宏引用, 类 REPT 指示, 类 IRP 指示)	8 级
TITLE 控制指令, 选项-LH 指定字符数	60 注意 5
SUBTITLE 控制指令指定字符数	72
单一文件中 include 文件嵌套级数	8 级
条件汇编嵌套级数	8 级
选项-I 指定 include 文件数	64
选项-D 指定符号数	30

- 注意 1** 不包括模块名和段名
使用内存。如果没有内存，则使用文件
- 注意 2** 如果汇编器不能解析符号值，则信息将传递给链接器。
例如，如果一个外部引用符需用 MOV 指令引用，将于 .rel 文件中生成两段重定位信息。
- 注意 3** 包括 CR 和 LF。如果一行字符数超过 2048 个，则将输出警告信息然后从第 2049 个字符起将被忽略。
- 注意 4** 交换名是通过类 SET/RESET 指示设为 true/false, 然后被 \$IF 等调用。
- 注意 5** 一行汇编列表文件能指定的最大字符数是 119，该数字可能是 "X-60" 或者更小。

3) 链接器最佳性能

表 1-3 链接器最佳性能

项目	最大性能
符号数(局部+公共)	65535
相同段的行数信息	65535
段数	65535
输入模块	1024
存储区名字符长	31 字符
存储区数	100 注意
由选项-B 指定的库文件数	10
由选项-I 指定的 include 文件路径数	64

注 包括默认定义

1.4 RA78K0 的功能

RA78K0 具有以下功能：

(1) 宏功能

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

(2) 分支指令的优化功能

RA78K0 有一个自动选择分支指令的指令（即 BR 指令、CALL 指令）。

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

(3) 条件汇编功能

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

第 2 章 产品概况与安装

本章介绍将存储在提供的 RA78K0 介质中的文件安装在用户开发环境（主机）中的步骤，以及将这些文件从用户开发环境中卸载的步骤。

2.1 主机与提供介质

汇编包支持如表 2-1 所示的开发环境。提供介质因主机而异。

表2-1 提供介质与汇编包的记录格式

主机	对应的操作系统（版本）	提供介质
IBM PC/AT 兼容机	日语 Windows (98/Me/2000/XP/NT 4.0) ^注	CD-ROM
	英语 Windows (98/Me/2000/XP/NT 4.0) ^注	
HP9000 系列 700 TM	HP-UX TM (Rel. 10.10 或更高)	CD-ROM
SPARCstation 系列	SunOS TM (Rel. 4.1.4 或更高) Solaris TM (Rel. 2.5.1 或更高)	

注 要在 Windows 中使用汇编器，必须有 PM+。

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

2.2 安装

下面介绍将 RA78K0 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（光盘驱动器）中设置 RA78K0 提供介质。安装程序会自动开始。按照显示器屏幕上显示的消息执行安装。

注意事项 如果安装程序未自动开始，则执行 RA78K0\DISK1 文件夹中的 SETUP.EXE 。

<3> 文件确认

使用 Windows 资源管理器等，检查 RA78K0 提供介质中包含的文件是否安装到主机中。关于每个文件夹的详细情况，请参阅 [2.4 目录配置](#)。

2.2.2 UNIX 版本的安装

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

<1> 进入系统

进入主机中。

<2> 目录选择。

转到安装目录。

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

<3> 设置提供介质

在光驱中设置光盘。

<4> 复制文件。

执行 `cp` 命令从光盘中复制文件（确认在光驱中设置了光盘后复制文件）。

<5> 设置环境变量 PATH

将 /necools/bin 添加至环境变量 PATH。

2.3 安装设备文件

通过从在线分发服务系统下载该设备，网址为：
<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安装设备文件。设备文件安装程序与 RA78K0 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 RA78K0 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 RA78K0 的设备文件（Ver. 3.30 或更高，16 位环境）注册到注册表中（32 位环境）。

还可以在 RA78K0 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

安装时显示的标准目录是 Windows 所在驱动器上的“NECTools32”。安装目录的配置如下图所示。安装时可以更改驱动与安装目录。在同一目录中安装 PM+ 及 RA78K0。

图 2-1 目录配置

—NECTools32\		
bin\		
st78K0.exe		结构化汇编预处理程序执行格式
ra78K0.exe		汇编器执行格式
lk78K0.exe		链接器
oc78K0.exe		目标转换器
lb78K0.exe		库管理程序
lcnv78K0.exe		表转换程序
lb78K0e.exe		库与 PM+ 环境的动态链接库之间的接口工具
lb78K0p.exe		独立启动库
ra78K0.is*		汇编器使用的文件
*78K0p.dll		PM+ 的 DLL 工具
*78K0.hlp		开始命令行的帮助文件
doc\		用户手册与辅助说明
hlp\		联机手册
setup\		安装与卸载的数据文件
smp78K0\ra78K0\		
K0main.asm		汇编示例程序
K0sub.asm		汇编示例程序
ra.bat		汇编示例程序的批处理文件
readme.doc		示例程序与批处理程序（文本文件）的说明
test1.s		结构化汇编的示例程序
test2.s		结构化汇编的示例程序
testinc.s		结构化汇编的示例程序
st.bat		结构化汇编示例程序的批处理文件

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

2.4.2 UNIX 版本目录配置

安装后的文件配置如下。下面假设安装在 /necools/bin 中。

图 2-2 目录配置

/necools/	
└─ bin/	
st78K0	结构化汇编预处理程序的可执行格式
ra78K0	汇编器的可执行格式
lk78K0	链接器的可执行格式
oc78K0	目标转换器的可执行格式
lb78K0	库管理程序的可执行格式
lcnv78K0	表转换程序的可执行格式
*.hlp	各个程序相应的帮助文件（文本文件）
ra78K0.is*	定义汇编器使用的指令集的列表文件
.asm,.s	用于安装配置的示例程序
*.sh	用于安装配置的批处理文件
*.sh	用于安装配置的批处理文件
readme.doc	安装配置外壳程序文件（文本文件）的说明

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

2.5 卸载步骤

2.5.1 Windows 版本的卸载

下面介绍卸载安装到主机中的文件的步骤。

(1) Windows 启动

启动主机和外围设备电源，并启动 Windows。

(2) 打开“控制面板”窗口

按[开始]按钮，然后选择 [设置]-[控制面板]，打开<控制面板>窗口。

(3) 打开 <添加或删除程序>窗口

双击<控制面板>窗口中的[添加或删除程序]图标，打开 <添加或删除程序>窗口。

注意 在 Windows XP 系统中，<添加或删除程序>显示为<添加/删除程序>

(4) 删除 RA78K0

在选中<添加或删除程序>窗口中<<添加或删除>> 标签中显示的安装软件列表中的“NEC RA78K0 78K/0 Assembler Vx.xx”后，单击[添加/删除...]按钮。

当<系统设置更改>窗口打开后，单击[是]按钮。

(5) 文件确认

使用 Windows 资源管理器等，检查主机中安装的文件是否被卸载。关于每个文件夹的详细情况，请参阅 [2.4 目录配置](#)。

2.5.2 UNIX 版本的卸载

使用 rm 命令卸载 [2.2.2 UNIX 版本的安装](#)中复制的文件。

2.6 环境设置

2.6.1 主机(IBM PC/AT 兼容)

RA78K0 运行于 32 位 CPU 的主机，如 i386™或以上。

因为汇编器包运行于 DOS 扩展的 32 位 CPU，按设计运行于以下系统：

Windows 98/Me 下 DOS 提示符

Windows Xp/NT 4.0 下命令提示符

7M 或者更多的保护内存是运行必须的。

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

- PATH 指定可执行格式的汇编器的存储目录。
 TMP 指定用于创建临时文件的目录（仅对 IBM PC/AT 兼容机有效）
 INC78K0 指定用于搜索包含文件的目录。
 LIB78K0 指定用于搜索库的目录，如果使用库的话。
 LANG78K 指定在注释中描述的汉字代码（2 字节代码）。

[示例]

- 对于 IBM PC/AT 兼容机

```
PATH = %PATH%; C:\NECTools32\bin
set TMP = C:\tmp
set INC78K0 = C:\NECTools32\inc78K0
set LIB78K0 = C:\NECTools32\lib78K0
set LANG78K = SJIS
```

- 对于 HP9000 系列 700 或 SPARCstation 系列

<使用 csh 的示例>

```
set path = ($path /ra78K0)
setenv INC78K0 /ra78K0
setenv LIB78K0 /ra78K0
setenv LANG78K EUC
```

<使用 sh 的示例>

```
PATH = $PATH:/ra78K0
INC78K0 = /ra78K0
LIB78K0 = /ra78K0
LANG78K = EUC
export PATH INC78K0 LIB78K0 LANG78K
```

2.6.3 源文件中的汉字代码

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

第 3 章 RA78K0 的执行过程

本章介绍使用汇编包 RA78K0 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“K0main.asm”及“K0sub.asm”进行了汇编、链接和目标程序转换。

本章节介绍了如何通过命令行运行汇编器包。

3.1 准备执行 RA78K0

3.1.1 示例程序

在系统盘中存储的文件有[k0main.asm]及[k0sub.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。

k0main.asm: 主程序模块

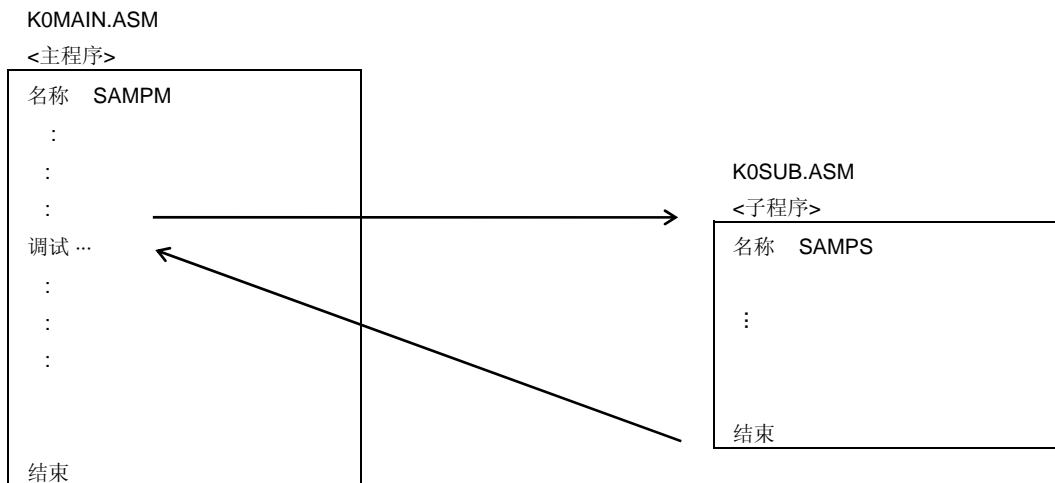
k0sub.asm: 子程序模块

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，它存储在源程序模块文件（k0main.asm）中。

子程序模块的名称是 SAMPS，它存储在源程序模块文件（k0sub.asm）中。

图 3-1 源程序的结构



■KOMAIN.ASM (主程序)

```

NAME    SAMPM
;*****
;
;
;    HEX -> ASCII Conversion Program
;
;    main-routine
;*****
;
PUBLIC  MAIN, START
EXTRN  CONVAH
EXTRN  @_STBEG

DATA   DSEG    saddr
HDTSA: DS      1
STASC: DS      2

CODE   CSEG    AT 0H
MAIN:  DW      START

CSEG

START:
;chip initialize
MOVW   SP,#_STBEG

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

CALL   !CONVAH             ;convert ASCII <- HEX
; output BC-register <- ASCII code
; set DE <- store ASCII code table

MOVW   DE, #STASC
MOV    A, B
MOV    [DE], A
INCW   DE
MOV    A, C
MOV    [DE], A

BR     $$

END

```

■K0SUB.ASM (子程序)

```

NAME    SAMPS
;
;
; *****
;
;   HEX -> ASCII Conversion Program
;           sub-routine
;
;   input condition : (HL) <- hex 2 code
;   output condition : BC-register <-ASCII 2 code
;
; *****
;
PUBLIC  CONVAH
;
CSEG
CONVAH: XOR    A, A
        ROL4   [HL]           ; hex upper code load
        CALL   !SASC
        MOV    B, A           ; store result

        MOV    A, A
        ROL4   [HL]           ; hex lower code load
        CALL   !SASC
        MOV    C, A           ;store result

        RET
;
; *****
;
;   subroutine  convert ASCII code          *
;   input  Acc (lower 4bits) <- hex code    *
;   output Acc      <- ASCII code          *
; *****
;
SASC:   CMP    A, #0AH         ; check hex code > 9
        BC    $SASC1
        ADD   A, #07H         ; bias (+7)
SASC1:  ADD   A, #30H         ;bias (+30)
        RET
;
        END

```

备注 1 示例程序是一个参考程序，用于向读者介绍 RA78K0 的功能与操作。它不能用作一个应用程序。

备注2 示例程序不操作寄存器组选择标记（RBS0, RBS1）的默认设置。这些项目的设置方法如下：
Register bank 0(FEF8H to FEFFH)

3.1.2 示例程序的结构

下面介绍用作下述操作例子的示例程序。

k0main.asm	主程序模块
k0sub.asm	子程序模块
mylib.lib	库文件（此处未用到）
sample.dr	指示文件

3.2 RA78K0 的执行过程

系统盘中的批处理文件(ra.bat)用于 RA78K0 的操作。

汇编器、链接器、目标转换器和表转换程序使用“k0main.asm”和“k0sub.asm”以上述顺序进行执行，它们是用汇编语言在 ra.bat 中编写的源文件。发生错误时，将输出一条消息，批处理文件终止。

用作目标设备的设备型号规格将被输入到该批处理文件中（可通过在线分发服务系统(ODS)下载获得设备文件）。

下面的解释将 uPD780058 作为目标设备。

■ ra.bat (用于验证 RA78K0 操作的批处理程序)

```
echo off
cls
set      LEVEL=0

if "%1" == "" goto ERR_BAT

ra78k0 -C%1 k0main.asm
if errorlevel 1 set LEVEL=1
ra78k0 -C%1 k0sub.asm
if errorlevel 1 set LEVEL=1
if %LEVEL% == 1 echo Assemble error !!
if %LEVEL% == 1 goto END

cls
lk78k0 k0main.rel k0sub.rel -s -orasample.lmf -prasample.map
if errorlevel 1 echo Link error !!
if errorlevel 1 goto END

cls
oc78k0 rasample
if errorlevel 1 echo Object conversion error !!
if errorlevel 1 goto END

cls
set LEVEL=0
lcnv78k0 -lrasample.lmf -rk0main.rel k0main.prn
if errorlevel 1 set LEVEL=1
lcnv78k0 -lrasample.lmf -rk0sub.rel k0sub.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。

指定目标设备型号，执行 RA78K0 操作验证批处理程序。

C>ra.bat 0058

显示屏上显示以下消息。

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

PASS1 Start
PASS2 Start

Target chip : uPD780058
Device file : Vx.xx

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

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

PASS1 Start
PASS2 Start

Target chip : uPD780058
Device file : Vx.xx
Assembly complete,  0 error(s) and  0 warning(s) found.
```

清除屏幕。

```
78K/0 Series Linker Vx.xx [xx xxx xxxx]
Copyright (C) NEC Electronics Corporation xxxx,xxxx
Target chip: uPD780058
Device file: Vx.xx

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

清除屏幕。

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

Target chip : uPD780058

Device file : Vx.xx

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

清除屏幕。

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

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

```
List Conversion Program for RA78K/0 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 C 驱动器的内容。

输出了以下文件。

k0main.rel:	目标文件
k0main.prn:	汇编列表文件
k0sub.rel:	目标文件
k0sub.prm:	汇编列表文件
rasample.lmf:	装入模块文件
rasample.map:	链接列表文件
rasample.hex:	十六进制格式目标模块文件
rasample.sym:	符号列表文件
k0main.p:	绝对汇编列表文件
k0sub.p:	绝对汇编列表文件

(3) RA78K0 执行过程概要

下面是对 RA78K0 执行过程的简要介绍。

图 3-2 RA78K0 执行过程 1

目标模块文件

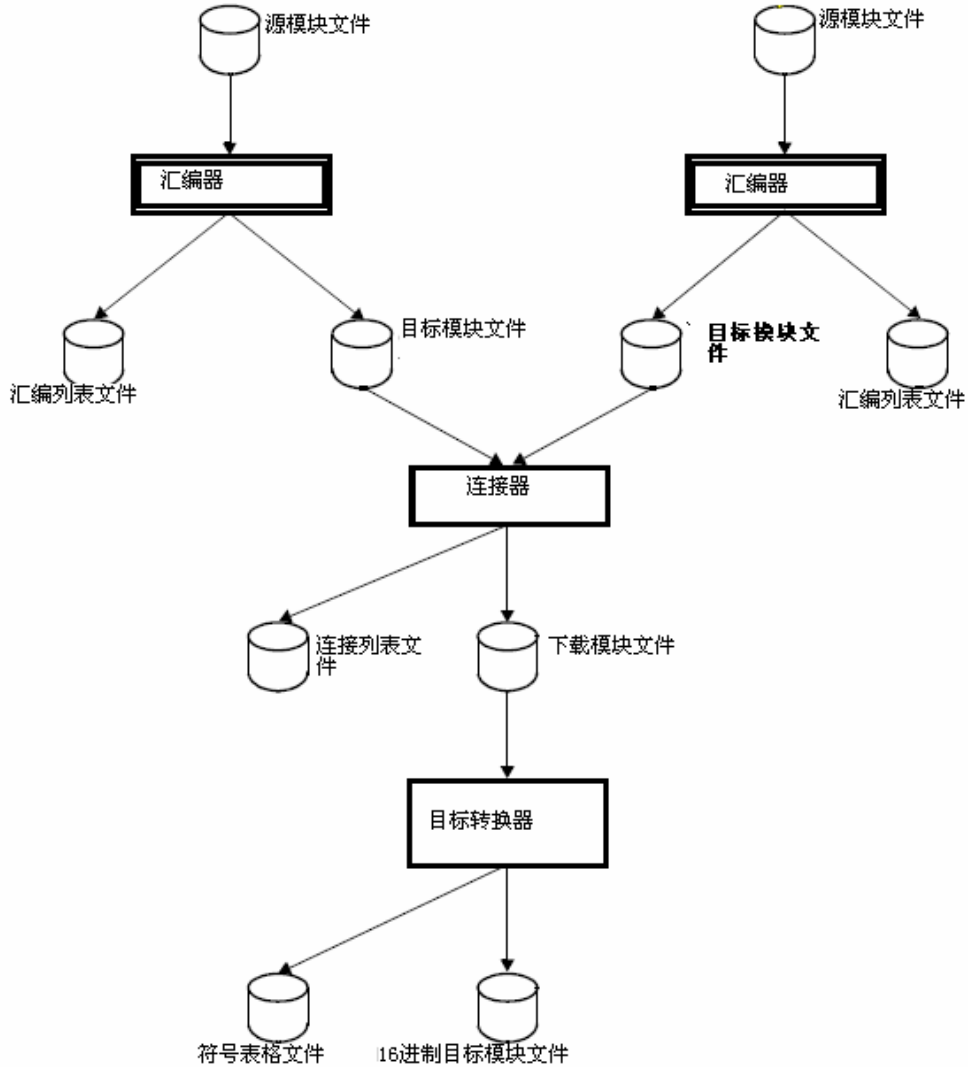
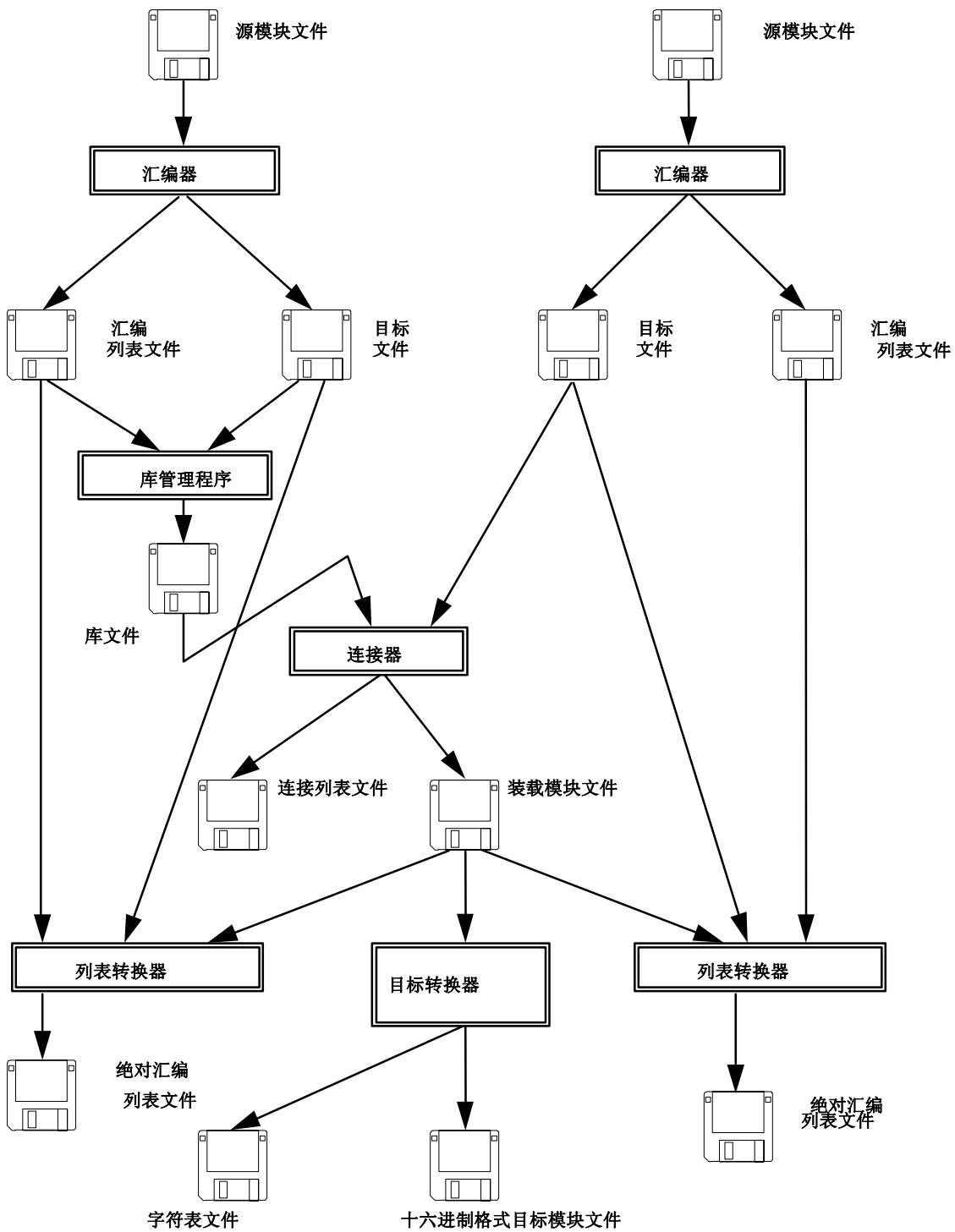


图 3-3 RA78K0 执行过程 2



3.3 ST78K0 的执行步骤

要验证 ST78K0 的操作，请使用系统盘中的批处理文件（st.bat）。

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

目标设备的型号规格可以被输入到该批处理文件中（分别获得设备文件）。

下面的解释将 uPD780058 作为目标设备。

■st.bat (ST78K0 操作验证批处理程序)

```
echo off
cls
setLEVEL=0

if "%1" == "" goto ERR_BAT

st78K0 -C%1 test1.s
ra78K0 test1.asm
if errorlevel 1 set LEVEL=1
st78K0 -C%1 test2.s
ra78K0 test2.asm
if errorlevel 1 set LEVEL=1
if %LEVEL% == 1 echo Assemble error !!
if %LEVEL% == 1 goto END

cls
lk78K0 test1.rel test2.rel -s -ostsample.lmf -pstsampl.map
if errorlevel 1 echo Link error !!
if errorlevel 1 goto END

cls
oc78K0 stsample
if errorlevel 1 echo Object conversion error !!
if errorlevel 1 goto END

cls
setLEVEL=0
lcnv78k0 -1stsample.lmf -rtest1.rel test1.prn
if errorlevel 1 set LEVEL=1
lc78k0 -lstsampl.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。

指定目标设备型号，执行 ST78K0 操作验证批处理程序。

```
C>st.bat 0058
```

将输出以下消息。

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

start

Target chip:uPD780058
Device file:Vx.xx

Conversion complete,          0 error (s) found.

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

PASS1 Start
PASS2 Start

Target chip:uPD780058
Device file:Vx.xx

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

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

start

Target chip:uPD780058
Device file:Vx.xx

Conversion complete,          0 error (s) found.

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

PASS1 Start
PASS2 Start

Target chip:uPD780058
Device file:Vx.xx

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

清除屏幕

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

  Target chip:uPD780058
  Device file:Vx.xx

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

清除屏幕。

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

  Target chip:uPD780058

  Device file:Vx.xx

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

清除屏幕。

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

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

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

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

清除屏幕。

```
No errors.
```

(2) 检查 C 驱动器的内容。

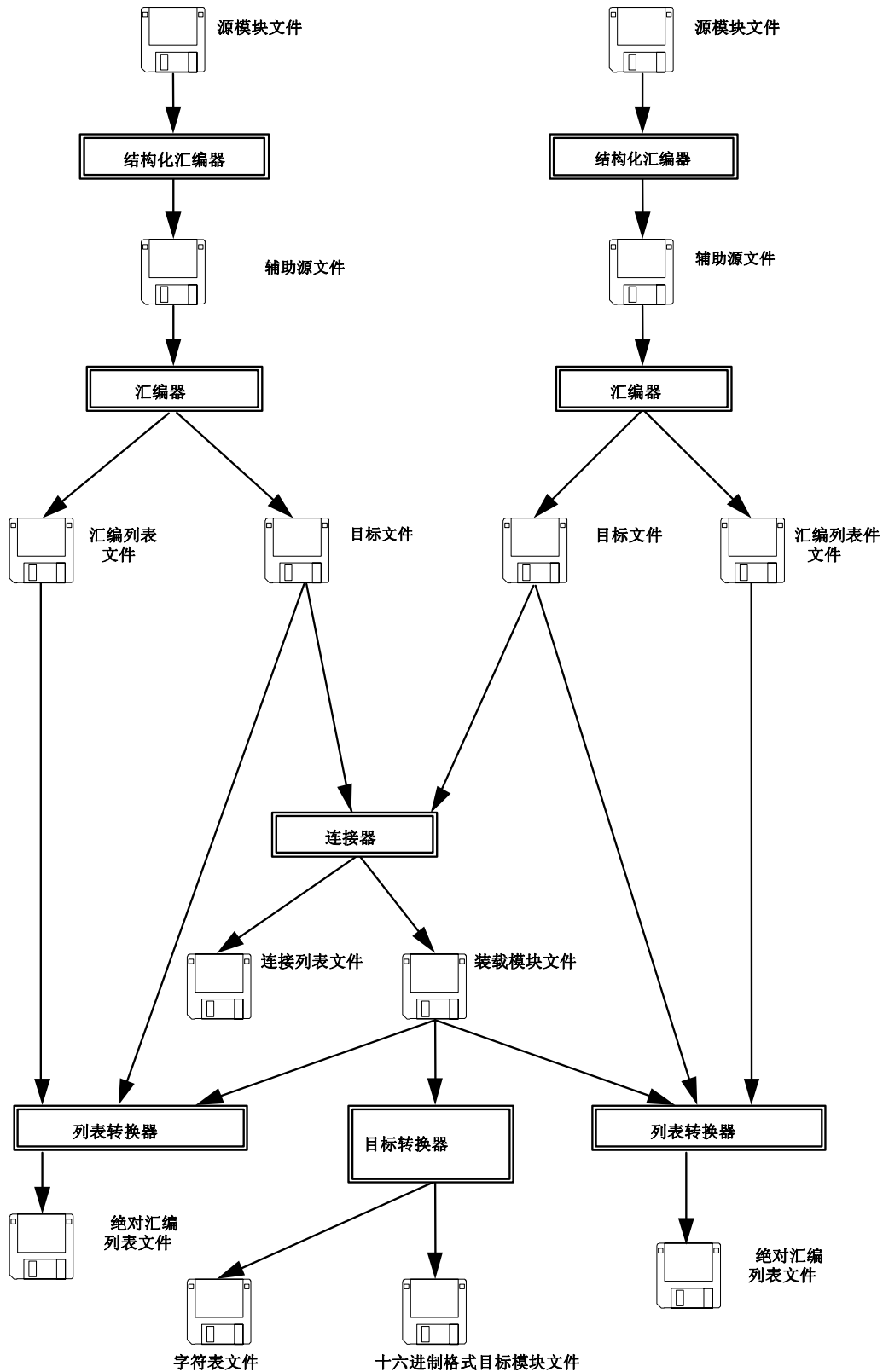
将输出以下文件。

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

(3) ST78K0 执行过程概要

下面是 ST78K0 执行过程的概要。

图 3-4 ST78K0 执行过程



3.4 从命令行中完成汇编、链接与目标程序转换（DOS 提示，UNIX）

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 KOMAIN.ASM。

在命令行中输入以下内容。

```
C>ra78k0 -c054 k0main.asm
```

显示屏上显示以下消息。

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

  Pass1 Start
  Pass2 Start

  Target chip : uPD78054
  Device file : Vx.xx

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

- (2) 检查 C 驱动器的内容。

汇编器输出目标文件（k0main.rel）及汇编列表文件（k0main.prn）。

如果汇编时指定了 -E 选项，则汇编器将输出一个错误列表文件（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 k0sub.asm。在命令行中输入以下内容。

```
C>ra78k0 -c054 k0sub.asm
```

显示屏上显示以下消息。

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

  Pass1 Start
  Pass2 Start

  Target chip : uPD78054
  Device file : Vx.xx

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

(4) 检查 C 驱动器的内容。

汇编器输出目标文件 (k0sub.rel) 及汇编列表文件 (k0sub.prn)。

汇编时, 如果指定了选项 -E, 则汇编器将输出一个错误列表文件。

(5) 创建一个指示文件。

指示文件是为链接器指示各段的存储单元的文件。

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

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

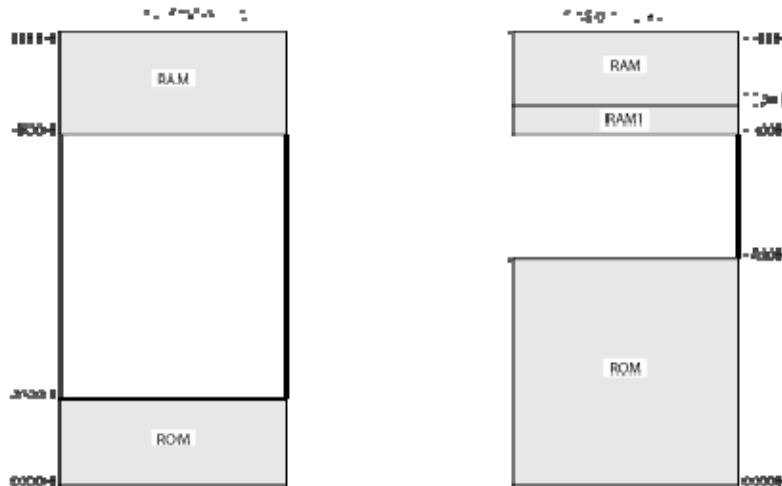
链接时, 使用 -D 选项将指示文件输入给链接器。

示例

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

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

图 3-5 链接指示



(6) 汇编的结果是, 输出的目标文件[k0main.rel] 与 [k0sub.rel] 被链接起来。

输入 k0.dr 作为指示文件。

在命令行中输入以下内容。

```
C>lk78k0 k0main.rel k0sub.rel -dk0.dr❶ -ok0.lmf -pk0.map -S❷
```

注: 1.如果未指定一个指示文件, 则不必要。

注 2 栈解析符号（_@STBEG）创建选项

显示屏上显示以下消息。

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

Target chip : uPD78054
Device file : Vx.xx

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

- (7) 检查 C 驱动器的内容。
链接器输出装入模块文件（k0.lmf）及链接列表文件（k0.map）。
链接时如果指定了选项 -E，则链接器将输出一个错误列表文件。
- (8) 链接的结果是，输出的装入模块文件（k0.lmf）被转换为一个十六进制格式的文件。
在命令行中输入以下内容。

```
C>oc78k0 k0.lmf -r -u0FFH
```

显示屏上显示以下消息。

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

Target chip : uPD78054
Device file : Vx.xx

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

- (9) 检查 C 驱动器的内容。
目标转换器输出十六进制的目标模块文件（k0.hex）及符号列表文件（k0.sym）。
- (10) 创建如下的库文件。
将汇编器输出的目标文件（k0sub.rel）存储为一个库文件。
在命令行中输入以下内容。

```
C>lb78k0 < k0.job
```

显示屏上显示以下消息。

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

*create K0.lib                ; Contents of "K0.job"
*add k0.lib k0sub.rel         ; Contents of "K0.job"
*exit
```

(11) 检查 C 驱动器的内容。

库管理程序输出库文件（k0.lib）。

(12) 创建一个绝对汇编列表文件，如下所示。

要创建绝对汇编表 k0main.asm，请向表转换程序输入[k0main.rel]、[k0main.asm] 及[k0.lmf]。

在命令行中输入以下内容。

```
C>lcnv78k0 k0main -lk0.lmf
```

显示屏上显示以下消息。

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

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

(13) 检查 C 驱动器的内容。

表转换程序输出绝对汇编列表文件（k0main.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多个选项，则不能在命令行中完整地指定启动所需信息，或者会重复同一个指定。在这种情况下，要使用参数文件。

要使用参数文件，请在命令行中指定参数文件指定选项。

汇编程序或链接程序是使用以下参数文件启动的。

```
>[path-name] ra78k0 Δ -F parameter-file-name  
>[path-name] lk78k0 Δ -F parameter-file-name  
>[path-name] oc78k0 Δ -F parameter-file-name
```

下面是一个使用例子。

```
C>ra78k0 -Fpara.pra  
C>lk78k0 -Fpara.plk  
C>oc78k0 -Fpara.poc
```

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

下面是一个使用编辑器创建参数文件的例子。

(para1.pra 的内容)

```
-c054 k0main.asm -e
```

(para1.plk 的内容)

```
k0main.rel k0sub.rel -bmylib.lib -osample.lmf -S
```

(para1.poc 的内容)

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

第 4 章 结构化汇编器预处理器

结构化汇编器输入用 78K0 系列微控制器的结构化汇编语言编写的源模块文件，将它们转换为汇编语言，然后将它们作为辅助源模块文件输出。

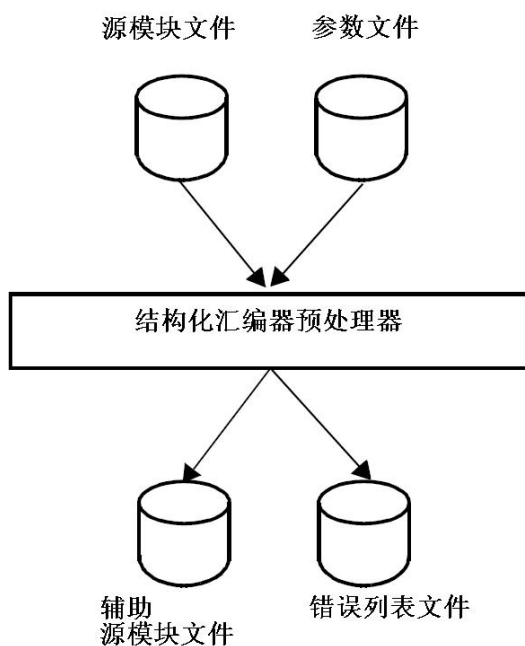
4.1 结构化汇编器的 I/O 文件

结构化汇编器的 I/O 文件如下表所示。

表4-1 结构化汇编器的I/O文件

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

图 4-1 结构化汇编器的 I/O 文件



4.2 结构化汇编器的功能

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

4.3 结构化汇编器启动

4.3.1 结构化汇编器启动

可以使用以下两个方法来启动结构化汇编器。

(1) 从命令行启动

通过输入以下命令来启动结构化汇编器。

X>	[path-name]	st78k0	[\Deltaoption]	...	\Delta	source-module-file-name	[\Deltaoption]	...
(a)	(b)	(c)	(d)		(e)		(d)	

(a) 当前的驱动器名

(b) 当前的目录名

(c) 结构化汇编器命令文件名

(d) 输入操作结构化汇编器所需的详细指令。当指定一个以上的选项时，用空格隔开选项。关于结构化汇编器选项的详细说明，请参阅第 4.4 节结构化汇编器选项。

通过使用双引号来包括含有空格的路径。

(e) 要进行结构化汇编的源模块的文件名。

通过使用双引号来包括含有空格的文件名。

示例 C>st78k0 -c054 test1.s -e

(2) 从一个参数档启动

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

要从参数文件中启动结构化汇编器，请在命令行中指定参数文件指定选项（-F）。

从一个参数文件启动结构化汇编器，如下所示。

X>	[path-name]	st78K0	[\Delta	source-module-file-name]	\Delta-F	parameter-file-name
(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] option [Δoption]... [Δ] Δ]]...
```

如果在命令中省略了源模块文件名，则只能在参数档中指定一个源模块文件名。

源模块文件名还可以写在选项后面。

将在命令中指定的所有选项和输出文件名写入参数档。

示例 使用编辑器创建参数档（test1.pst）。

- test1.pst 的内容

```
;Parameterfile  
test1.s -osample.asm  
-esample.est -c054
```

- 使用参数档（test1.pst）启动结构化汇编器。

```
C>st78k0 -ftest1.pst
```

4.3.2 执行启动和结束消息

(1) 执行启动消息

当结构化汇编器启动时，在显示屏上会出现以下执行启动消息。

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

(2) 处理显示消息

每隔结构化汇编器处理的 100 行显示一次“.”。

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

(3) 执行结束消息

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

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

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

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

```
TEST1.S (8): RA78KO error E1209: Syntax error

Target chip:uPD78xxxx
Device file:Vx.xx

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

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

示例 1 当指定一个不存在的源模块文件时

```
C>st78K0 sample.s

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

RA78K0 error F1006: File not found 'SAMPLE.S'

Program aborted.
```

在上例中，指定一个不存在的源模块文件会产生一个错误，同时汇编停止。

示例 2 当指定一个不存在的选项时

```
C>st78K0 test1.s -z

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

RA78K0 error F1012: Missing parameter '-z'
Please enter 'ST78K0 --', if you want help messages.

Program aborted.
```

在上例中，指定一个不存在的选项会产生一个错误，同时汇编停止。

当显示一条错误消息且汇编停止时，请在第 12 章 [错误消息](#) 中查找错误原因，并采取相应操作。

4.4 结构化汇编器选项

4.4.1 结构化汇编器选项的类型

结构化汇编器选项给出了结构化汇编器操作的详细说明。

结构化汇编器选项分为以下 13 种。

表4-2 结构化汇编器选项

种类	选项	说明
设备类型指定	-C	指定目标设备的类型。
字符符号指定	-SC	指定字符符号名的结束字符。
符号定义	-D	指定给 #IFDEF 指令等的符号
制表符数指定	-WT	指定转换的指令的输出位置。
包含文件路径指定	-I	指定驱动器和包含文件的目录
辅助源文件指定	-O	指定辅助源文件名。
错误列表档指定	-E	指定错误列表文件名。
参数档指定	-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级调试数据的输出。
	-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。
汉字码 (2 字节代码) 指定	-ZS	指定在注释语句中描述的汉字码类型。
	-ZE	
	-ZN	
设备文件搜索路径指定	-Y	指定用于搜索设备文件的路径。
说明指定	--	在显示屏上输出帮助信息。

4.4.2 结构化汇编器选项说明

本节详细介绍各个结构化汇编器选项。

(1) 设备类型指定**设备类型指定 (-C)****[语法]**

-C 设备类型

- 默认预设：
不能省略

[功能]

指定作为结构化汇编器目标的设备。

[说明]

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

[使用示例]

- uPD78054 被指定为目标设备。

```
C>st78k0 test.s -c054
```

[注]

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

有关设备类型的详细情况，请参阅各个设备的设备文档中的使用说明。

(2) 字符符号指定**字符符号指定 (-SC)****[语法]**

-SC 字符

- 默认预设
- SCP

[功能]

当字节/字必须区分符号名时，指定作为被判断目标的符号的最后字符。

[说明]

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

[使用示例]

一个以 @ 结束的符号被指定为一个字符符号。

```
C>st78k0 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

符号定义指定(-D)

[语法]

```
-D symbol-name [=numerical-value] [, symbol-name[=numerical-value]...]
```

[功能]

- 定义符号

[说明]

- 赋予符号的数值可以是二进制、八进制、十进制或十六进制的。如果省略了数值指定，则值将变为 1。
- 使用该选项定义一个符号等于使用 `#define` 指令来定义一个符号。
- 使用逗号作为分隔符号号，最多可以在命令行中定义 30 项。
- 符号名最多可包含 31 个字符
- 该选项通常与 `#ifdef` 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` 指令一起指定，将会输出一条警告消息，且 `#define` 指令将被视为无效。
- 字母不分大小写。

[使用示例]

符号“TRUE”并定义为 1。

```
C>st78k0 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
```

- 默认预设:

```
-WT2, 3, 4
```

[功能]

指定制表符数，直到转换的汇编语言输出。

[说明]

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

[使用示例]

“3”被指定给数值 1，“4”被指定给数值 2，“5”被指定给数值 3。

```
C>st78k0 test.s -wt3,4,5
```

(5) 包含文件路径指定**包含文件路径指定 (-I)****[语法]**

-I 路径名 [, 路径名] ... (可指定2个或者以上的路径名.)

- 默认预设:

include 文件按照以下顺序搜索

- (i) 1.源文件所在路径
- (ii) 2.环境变量所指定的路径(INC78K0)

[功能]

指定待输出到结构化汇编器的包含文件路径的名称。

[说明]

- 可以一次指定两个或者以上的路径名，由分隔符","隔开。
- 在","前后不可插入空格
- **include** 文件按照以下顺序搜索
 - (i) 如果 -I选项指定了两个或者以上的路径名，**include**文件按照指定顺序搜索
 - (ii) 如果 -I选项被多次定义，则按照最后一次定义的-I选项进行搜索
 - (iii) 在进行按照-I选项指定路径的搜索完毕之后，再搜索默认路径
- 如果-I选项后指定的是文件名而非路径名，或者未指定路径名，将发生中止错误。
- 当定义65或者以上次-I选项时也将出现中止错误。

[使用示例]

- 包含文件的目录被指定为 c:\include。

```
C>st78k0 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
```

- 默认预设

```
-O input-file-name.asm
```

[功能]

指定转换后辅助源文件的输出目的地及文件名。

[说明]

- 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，则将在当前的目录中创建输出文件，方法是用 .ASM 代替输入档的档类型。
- “NUL”或“AUX”均可指定为文件名。
- 当由于致命错误而停止处理时，辅助源文件将不被输出。
- 如果执行了一个以上的指定，则最后指定的项有效。

[使用示例]

“sample.asm”被指定为辅助源文件。

```
C>st78k0 test.s -osample.asm
```

(7) 错误列表档指定**错误列表档指定 (-E)****[语法]**

-E [[驱动器:][目录] 文件名]

- 默认预设:

-E input-file-name.est

[功能]

指定错误列表档的输出目的地及文件名。

[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，则将在当前的目录中创建输出文件，方法是用 .est 代替输入档的档类型。
- “NUL”或“AUX”均可指定为文件名。
- 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]

“sample.est”被指定为错误列表档。

```
C>st78k0 test.s -esample.est
```

(8) 参数档指定

参数档指定 (-F)

[语法]

-F [[驱动器:] 目录] 文件名

- 默认预设
没有输入档

[功能]

- 指定参数档的文件名。

[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。如果省略了档类型，则类型将被假定为“.pst”。
- 该选项在使用选项 -D 在命令行中定义大量符号时有效。
- 多次指定该选项会产生一个错误。
- 禁止参数档嵌套，否则会产生错误。
- 参数文件中“;”或“#”以后的字符将被视为注释，直到左侧或文件结束。

[使用示例]

“sample.pst”被指定为参数档。

```
C>st78k0 -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

-默认预设

-GS

[功能]

指定结构化汇编器源代码级调试数据的输出。

[说明]

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

[注]

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

[使用示例]

指定向辅助源文件输出调试数据。

```
C>st78k0 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

- 默认预设
当由于致命错误而停止处理时，辅助源文件不被输出。

[功能]

当由于致命错误而停止处理时，强制输出辅助源文件。

[说明]

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

[使用示例]

指定强制输出辅助源文件。

```
C>st78k0 test.s -j
```

(11) 汉字码 (2 字节代码) 指定**汉字码指定 (-ZS/-ZE/-ZN)****[语法]**

-ZS
-ZE
-ZN

- 默认预设:

根据操作系统的不同解释如下:

Windows、HP-UX: -ZS

SunOS、Solaris: -ZE

[功能]

指定在注释中描述的汉字码类型。

[说明]

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

[使用示例]

规定汉字被解释为移位 JIS 码。

```
C>st78k0 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

- 默认预设:

默认预设: 设备文件将按以下顺序进行搜索

- (i) <..\dev> (用于启动 st78K0.exe 的路径)
- (ii) 启动 st78K0.exe 的路径
- (iii) 当前路径
- (iv) 环境变量 PATH 指定的路径

[功能]

指定用于搜索设备文件的路径。

[说明]

- 从指定的路径中读出设备文件。
- 如果指定了其它路径名, 则会发生错误。
- 即使路径指定符号未写在目录的结尾, 也会假设它在那里。
- 设备文件将按以下顺序进行搜索。
 - (i) -Y 选项指定的路径
 - (ii) <..\dev> (用于启动 st78k0.exe 的路径)
 - (iii) 启动 st78k0.exe 的路径
 - (iv) 当前路径
 - (v) 环境变量 PATH 指定的路径

[使用示例]

指定从目录 c:\NECTools32\dev 中读出设备档。

```
C>st78k0 test.s -yc:\NECTools32\dev
```

(13) 说明指定

说明指定 (-)

[语法]

--

- 默认预设
 无显示

[功能]

选项 -- 显示一个帮助信息。

[应用]

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

[说明]

当指定了 -- 选项时，所有其它的结构化汇编器选项均不可用。

注意事项 该选项不能在 PM + 上指定。

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

[使用示例]

- 当指定了 -- 选项时，显示屏上将输出一个帮助信息。

```
C>st78k0 --

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

Usage: st78k0 [option [...]] input-file [option[...]]

The option is as follows ([ ] means omissible, ...means repetition).
-cx          : Select target chip. (x = 000, 012, 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.

Press RETURN to continue...
-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 在 PM+ 中设置选项

本节将介绍如何从 PM + 中设置结构化汇编器选项。

4.5.1 设置选项方法

从 PM+ 的 [Tools] 菜单中选择 [Structured Assembler Options...], 或者点击工具栏上的 [ST] 按钮, 弹出 <Structured Assembler Options> 对话框。

在此对话框中输入所需选项, 即可设置结构化汇编器选项。

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

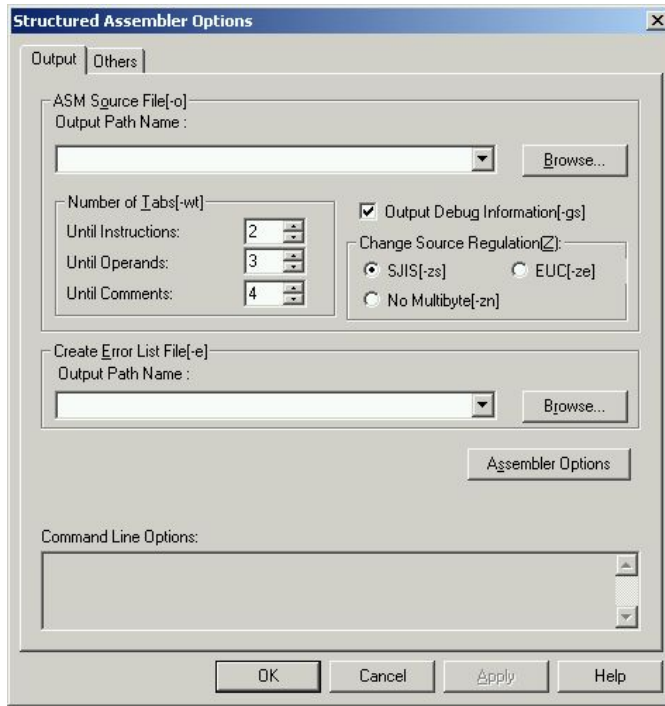


图 4-3 <Assembler Source Options> 对话框 (当选中 [Assembler Options] 标签时)

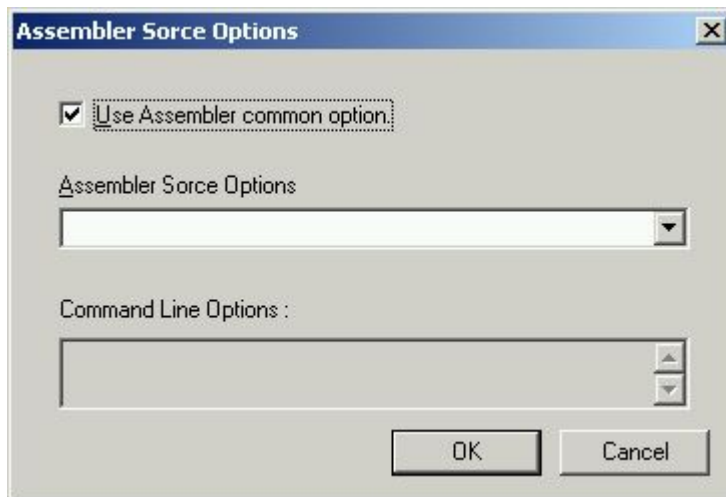
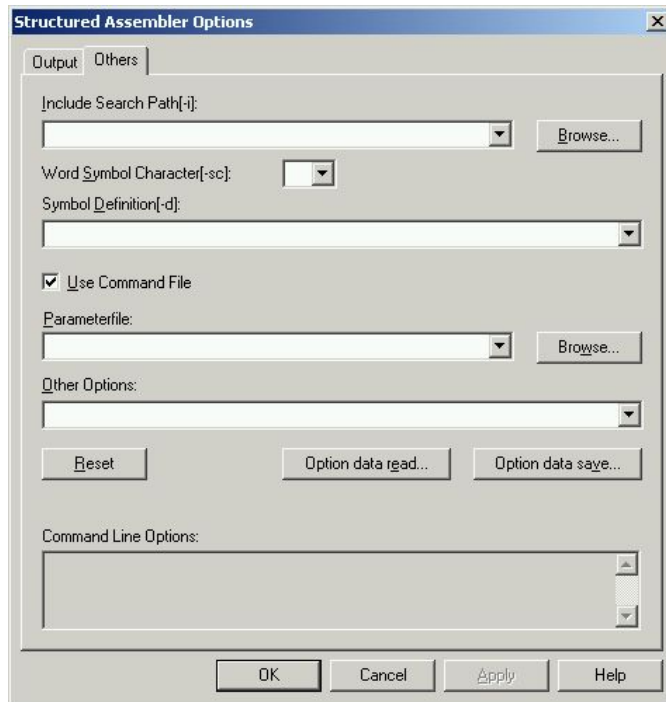


图 4-4 <Structured Assembler Options>对话框（当选中 <<Others>>标签时）



4.5.2 选项

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

<Output >标签

- **ASM Source File [-o]**
Output Path Name (选择 common 选项时)
使用 [Browse...] 按钮或通过直接输入指定 ASM 源文件输出路径。
- **Output File Name(选择individual选项时)**
通过[Browse...] 按钮或者直接输入路径及文件名来指定ASM源文件。
- **Number of Tabs [-wt]**
指定制表符数，直到转换的汇编语言输出。
指令、操作数和注释前的制表符数可以单独指定。
- **Output Debug Information [-gs]**
通过该选项将输出 ASM 源文件（辅助文件）中的调试信息。
- **Change Source Regulation [Z]**
选择在源代码注释中使用的多字节代码类型（SJIS [-zs]、EUC [-ze]、无汉字码[-zn]）。
- **Create Error List File [-e]**
Output Path Name(选定 common 选项时)
要输出一个错误表档，请在文本框中输入文件名。
使用 [Browse...] 按钮指定路径。
Output Path Name(选定 individual 选项时)
要输出错误列表文件，可以通过[Browse...] 按钮或者直接输入路径文件名来指定错误列表文件名。
- **Assembler Options**
指定汇编器源模块文件的汇编器选项。
- **Command Line Options**
编辑框为只读。将显示当前设置的选项字符串
- **Use Assembler common option**
启用在<Structured Assembler Options> 对话框中设置的汇编器的各个通用选项。
- **Assembler Source Options**
输入包括选项名在内的字符串，以启用输出汇编器源文件的选项。
- **Command Line Options**
编辑框为只读。将显示当前设置的选项字符串

<Others>标签

- **I**nclude Search Path [-i]
使用 [**E**dit] 按钮或通过直接输入指定包含源文件的路径。
- **W**ord **S**ymbol Character [sc-]
指定被定义为一个字的符号的最后字符。此后，生成一个指令，以使用最后一个字符决定它是一个字符还是一个字节符号。
- **S**ymbol Definition [-d]
输入要定义为一个符号的数值。
- **U**se Command File
选择该复选框来创建一个命令档。
- **P**arameterfile [-f]
使用 [**B**rowse...] 按钮或通过直接输入读取用户定义参数文件。
- **O**ther Options
要指定一个非用复选框或单选按钮选择的选项，请在输入框中输入选项。
注意：帮助选项(-)在 **PM+**中不可使用。
- **R**eset
清除输入内容。
- **O**ption data **r**ead
打开 < Read Option Data >对话框，在指定选项数据文件后，读取该档。
- **O**ption data **s**ave
打开 < Save Option Data >对话框，用指定的名称将选项数据保存至选项数据文件中。
- **C**ommand Line Options
该编辑框是只读的。显示当前设置的选项字符串。

4.5.3 编辑选项对话框

项目在<Edit Option>对话框中以列表格式编辑。

<Edit Option>对话框如下所述。

图 4-5 <Edit Option>对话框

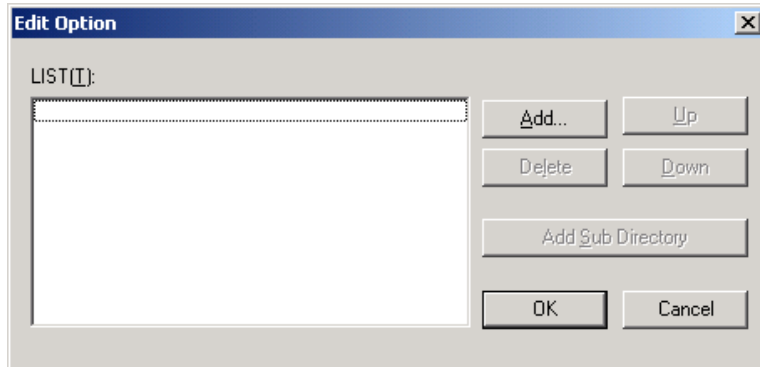
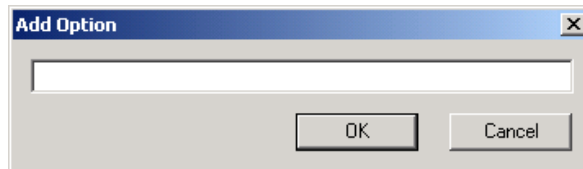


图 4-6 <Add Option>对话框



—[Add]按钮

添加一个列表项。

如果要添加的项目是一个文件或目录，则相应的 < Browse for Folder > 对话框打开。

在其他情况下，则会打开< Add Option >对话框。在此对话框中指定要添加的项目的详细信息。

—[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

当在<< Others >> 标签指定项目为 Include Search Path[-i](I)时，可以添加一个子目录到选定的列表项。

第 5 章 汇编器

汇编器输入用 78k0 系列微控制器的汇编语言编写的源模块文件，并将它们转换为机器语言代码。

汇编器同时输出列表文件，例如汇编列表文件和错误列表文件。

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

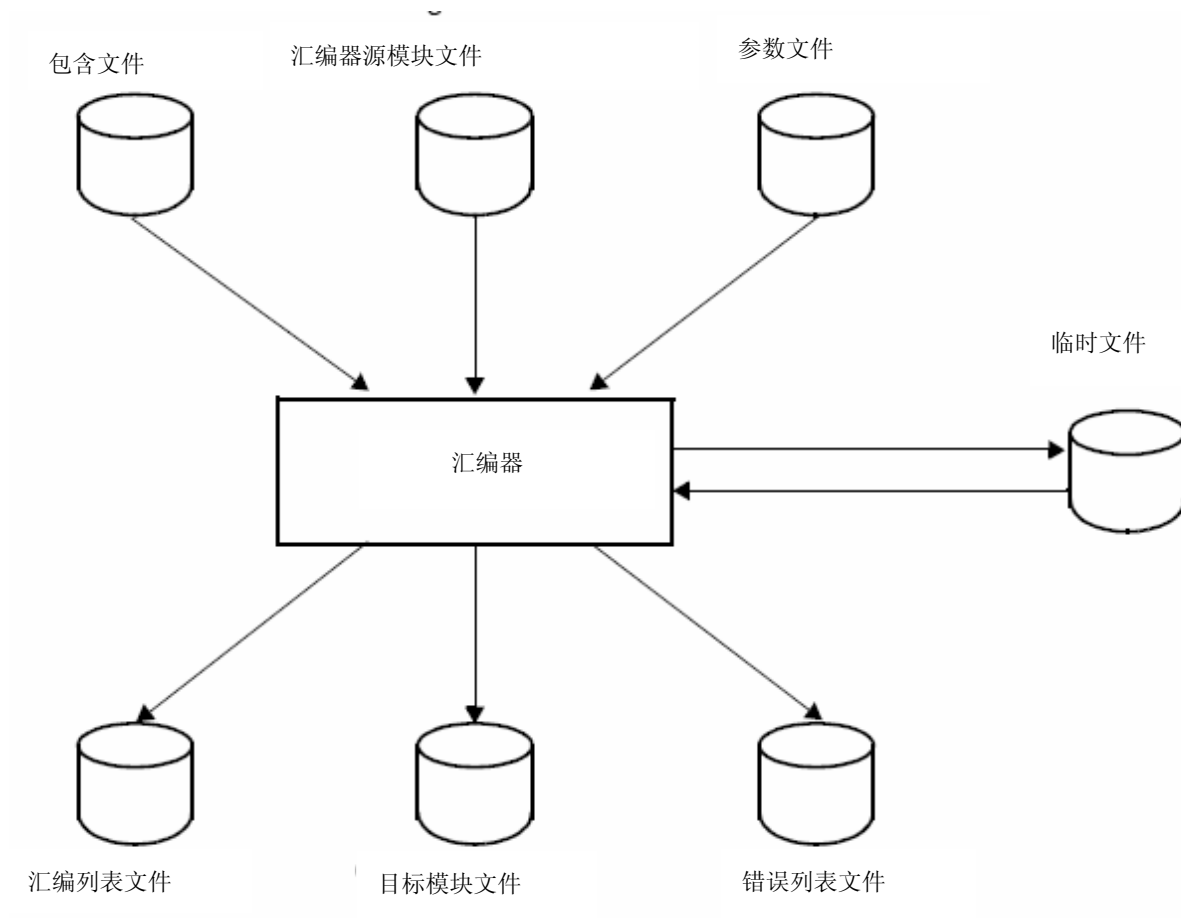
5.1 汇编器的 I/O 文件

汇编器的 I/O 文件如下表所示。

表5-1 汇编器的I/O文件

类型	文件名	说明	默认文件类型
输入文件	汇编器源模块文件	<ul style="list-style-type: none">• 这些是用 78k0 系列微处理器的汇编语言编写的源模块文件。• 这些文件是用户创建的。	.ASM
	包含文件	<ul style="list-style-type: none">• 这些文件用于汇编源模块文件引用。• 这些是用 78k0 系列微处理器的汇编语言编写的文件。• 这些文件是用户创建的。	无
	参数文件	<ul style="list-style-type: none">• 这些文件包含执行过的文件的参数。• 这些文件是用户创建的。	.PRA
输出文件	目标文件	<ul style="list-style-type: none">• 这些是二进制文件，包括有关机器语言数据和机器语言单元地址的浮动数据和符号数据。	.REL
	汇编列表文件	<ul style="list-style-type: none">• 这些是包含诸如汇编列表和交叉引用列表等汇编数据的文件。	.PRN
	错误列表文件	<ul style="list-style-type: none">• 这些是包含汇编时产生的错误数据的文件。	.ERA
I/O 文件	临时文件	<ul style="list-style-type: none">• 这些是为了汇编由汇编器自动创建的文件。当汇编结束时，临时文件被删除。	RAxxxx.\$n (n = 1 至 4)

图 5-1 汇编器的 I/O 文件



5.2 汇编器的功能

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

5.3 汇编器启动

可以使用以下两个方法来启动汇编器。

5.3.1 从命令行启动

X >	[path-name]	RA78K0	[\Deltaoption]	...	\source-module-file-name	[\Deltaoption]	...	[\Delta]
(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) 输入操作汇编器所需的详细指令。

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

通过使用双引号来包括含有空格的路径

(e) 要进行汇编的源模块的文件名。

通过使用双引号来指定含有空格的文件名

示例 C>ra78k0 -c054 k0main.asm -e -np

5.3.2 从一个参数文件启动

当启动汇编器所需的数据不适合命令行，或当每次执行汇编都要指定相同的汇编器选项时，使用参数文件。要从参数文件中启动汇编器，请在命令行中指定参数文件指定选项（-F）。从一个参数文件启动汇编器的方法如下所示。

```
X > RA78k0 [ $\Delta$ source-module-file]  $\Delta$ -F parameter-file-name
           |           |
           (a)        (b)
```

- (a) 一个包含启动汇编器所需数据的文件。
- (b) 参数文件（指定选项）
使用编辑器创建参数文件。

编写参数文件内容的规则如下所示。

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

如果在命令行中省略了源模块文件名，则只能在参数文件中指定一个源模块文件名。源模块文件名还可以写在选项后面。

将在命令行中指定的所有汇编器选项和输出文件名写入参数文件。

关于参数文件的详细说明，请参阅第 5.4.3 节 汇编器选项说明。

示例 使用编辑程序创建参数文件（k0main.pra）。

- k0main.pra 的内容

```
; parameter file
k0main.asm -osample.rel
-psample.prn
```

- 使用参数文件（k0main.pra）启动汇编器。

```
C>ra78k0 -fk0main.pra
```

5.3.3 执行启动和结束消息

(1) 执行启动消息

当汇编器启动时，在显示屏上会出现以下执行启动消息。

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

(2) 执行结束消息

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

```
Pass1 Start
Pass2 Start

Target chip:uPD78xxx
Device file:Vx.xx

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

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

```
Pass1 Start
K0MAIN.ASM(12) : RA78K0 E2201 : Syntax error
Pass2 Start
K0MAIN.ASM(12) : RA78K0 E2201 : Syntax error
K0MAIN.ASM(29) : RA78K0 E2407 : symbol reference 'CONVAH'
K0MAIN.ASM(29) : RA78K0 E2303 : Illegal expression

Target chip:uPD78xxx
Device file:Vx.xx

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

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

示例 1 指定一个不存在的源模块文件时。

```
C>ra78k0 sample.asm

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

RA78K0 error F2006: File not found 'SAMPLE.ASM'
Program aborted.
```

上例指定了一个不存在的源模块文件。发生了一个错误，同时汇编器中止汇编。

示例 2 指定一个不存在的汇编器选项。

```
C>ra78k0 k0main.asm -z

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

RA78K0 error F2012: Missing parameter '-z'
Please enter 'RA78k0--', if you want help messages.
Program aborted.
```

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

5.4 汇编器选项

5.4.1 汇编器选项类型

汇编器选项是汇编器操作的详细说明。汇编器选项分为以下 17 种。

表5-2 汇编器选项

种类	选项	说明
设备类型指定	-C	指定目标设备的设备类型。
目标文件输出指定	-O	指定输出目标文件。
	-NO	
强制目标文件输出指定	-J	强制输出目标文件。
	-NJ	
调试数据输出指定	-G	输出调试数据（局部符号数据）至一个目标文件。
	-NG	
	-GA	输出汇编器源调试数据至目标文件。
	-NGA	
包含文件读取路径指定	-I	从包含文件中指定的路径读取。
汇编列表文件输出指定	-P	指定输出汇编列表文件。
	-NP	
汇编列表文件数据指定	-KA	将汇编表输出至汇编列表文件。
	-NKA	
	-KS	将符号表输出至汇编列表文件。
	-NKS	
	-KX	将交叉引用表输出至汇编列表文件。
汇编列表文件格式指定	-LW	更改可在汇编列表文件中 1 行上打印的字符数。
	-LL	更改可在汇编列表文件 1 页中打印的行数。
	-LH	输出在汇编列表文件的首部指定的字符串。
	-LT	更改一个制表符中的空格数。
	-LF	在汇编列表文件结尾处插入一个换行代码。
	-NLF	
错误列表文件输出指定	-E	输出一个错误列表文件。
	-NE	

表5-2 汇编器选项

种类	选项	说明
参数文件指定	-F	从指定的文件输入输入文件名和汇编器选项。
临时文件创建路径指定	-T	在指定的路径中创建一个临时文件。
汉字码指定	-ZS	注释中所描述的汉字解释为移位 JIS 码。
	-ZE	注释中所描述的汉字解释为 EUC 码。
	-ZN	注释中所描述的字符不解释为汉字。
设备文件搜索路径指定	-Y	从指定的路径读取设备文件。
符号定义指定	-D	定义符号。
系列常用对象指定	-COMMON	设置 78K0 系列常用目标模块文件的输出
自编程指定	-SELF	当使用自编程时指定
帮助指定	--	在显示屏上显示帮助信息。

5.4.2 汇编器选项的优先级

表 5-3 说明了当同时指定了两个汇编器选项时，哪个选项有效。

表5-3 汇编器选项的优先级

	-NO	-NP	-NKA	-NKS	-KX	-NKX	--
-J	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]

当在水平轴上的选项被指定时，则在垂直轴选择中显示的选项不可用。

示例 C>ra78k0 -c054 k0main.asm -no -lw80 -lf

-LW 和 -LF 选项不可用。

[用Δ标记的项]

当在水平轴上的所有三个选项被指定时，则在垂直轴选择中显示的选项不可用。

示例 C>ra78k0 -c054 k0main.asm -p -nka -nks -nkx

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

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

示例 C>ra78k0 -054k0main.asm -o -no

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

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

5.4.3 汇编器选项说明

本节详细介绍各个汇编器选项。

(1) 设备类型指定**设备类型指定 (-C)****[语法]**

-C 设备类型

- 默认预设:

不能省略

[功能]

选项-C 指定目标设备的设备类型。

[应用]

务必指定选项 -C。汇编器对目标设备执行汇编，并生成该设备的目标代码。

[说明]

对于使用选项 -C 指定的设备，请参阅“使用设备时要注意的问题”。

[注]

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

$\Delta\$\Delta$ PROCESSOR $\Delta(\Delta$ device-type $\Delta)$
 $\Delta\$\Delta$ PC $\Delta(\Delta$ device-type $\Delta)$;简化形式

有关控制指令的信息，请参阅语言部分第 4 章 控制指令。

[使用示例]

在命令行中指定选项 -C 的方法如下所示。

```
C>ra78k0 -c054 k0main.asm
```

(2) 目标文件输出指定

目标文件输出指定(-O/-NO)

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

[功能]

- 选项 **-O** 指定输出目标文件。它还指定文件输出的位置和文件名。
- 选项 **-NO** 使 **-O**, **-J**, **-G**和**-GA**选项不可用。

[应用]

使用选项 **-O** 指定目标文件输出的位置或更改其文件名。

在执行汇编时, 指定选项 **-NO** 仅输出汇编列表文件。这将会缩短汇编的时间。

[说明]

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

[使用示例]

指定输出目标模块文件 (sample.rel)。

```
C>ra78k0 -c054 k0main.asm -osample.rel
```

(3) 强制目标文件输出指定**强制目标文件输出指定 (-J/-NJ)****[语法]**

-J -NJ

默认预设:

-NJ

[功能]

- 选项 -J 指定，即使发生致命错误也可输入目标文件
- 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，无法输出目标文件。当您想要在出现已发生致命错误的通知时执行程序，指定选项 -J 输出目标模块文件。

[说明]

- 当指定选项 -J 时，即使发生致命错误也会输入目标文件
- 如果同时指定了选项 -J 和选项 -NJ，则后指定的选项优先。
- 若已定义 -NO 选项，-J 选项将不可用。

[使用示例]

即使发生致命错误，指定输出目标文件。

```
C>ra78k0 -c054 k0main.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

默认预设:

-G

[功能]

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

[应用]

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

[说明]

- 如果同时指定了选项 **-G** 和选项 **-NG**，则后指定的选项优先。
- 无论在何处指定，**-GA** 选项都比其它选项有更高的优先级。
- 若已经指定 **-NO** 选项，**-G** 选项将无效。

[注意]

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

$\Delta\$\Delta$ DEBUG $\Delta\$\Delta$ DG ; 缩写形式 $\Delta\$\Delta$ NODEBUG $\Delta\$\Delta$ NODG ; 缩写形式
--

有关控制指令的信息，请参阅语言部分的第 4 章控制指令。

[使用示例]

指定添加调试数据到目标模块文件。

```
C>ra78k0 -c054 k0main.asm -g
```

(b) -GA/-NGA**[语法]**

-GA -NGA

默认预设:

-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]

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

[说明]

- 如果同时指定了选项 **-GA** 和选项 **-NGA**，则后指定的选项优先。
- 不管它所指定的位置如何，选项 **-GA** 都优先其他选项。
- 若已经指定 **-NO** 选项，则 **-GA** 将无效。

[注意]

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

$\Delta\$\Delta$ DEBUGA $\Delta\$\Delta$ NODEBUGA
--

有关控制指令的信息，请参阅语言部分的第 4 章控制指令。

[使用示例]

指定添加汇编器源调试数据到目标文件。

```
C>ra78k0 -c054 k0main.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] ... (可指定两个或多个路径名)

默认预设:

包含文件按照如下顺序进行搜索:

- (i) 源文件中所包含的路径
- (ii) 环境变量 (INC78k0) 所指定的路径

[功能]

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

[应用]

- 使用选项 -I 检索来自某个路径的包含文件。

[说明]

- 通过使用 ';' 分隔, 一次可指定两个或多个路径名。
- 在 ';' 前或后不可输入空格。
- \$include 指定的文件搜索顺序如下:
 - (iv) 如果 -I 选项指定了两个或者以上的路径名, include 文件按照指定顺序搜索
 - (v) 如果 -I 选项被多次定义, 则按照最后一次定义的 -I 选项进行搜索
- (iii) 在进行按照 -I 选项指定路径的搜索完毕之后, 再搜索默认路径
- 如果在指定选项 -I 后未指定任何路径名, 或如果省略了路径名, 则会发生终止错误。
- 如果使用选项 -I 指定 9 个或更多的路径名, 则会发生终止错误。

[使用示例]

- 从目录 c:\SAMPLE 读取包含文件。

```
C>ra78k0 -c054 k0main.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
```

默认预设:

```
-P input-file-name.PRN
```

[功能]

- 选项 **-P** 指定输出汇编列表文件。它还可指定输出文件的目的地和文件名。
- 选项 **-NP** 可使选项 **-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。
- 在执行汇编时, 指定选项 **-NP** 仅输出目标文件。这将会缩短汇编的时间。

[说明]

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

[使用示例]

创建汇编列表文件 (SAMPLE.PRN)。

```
C>ra78k0 -c054 k0main.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

默认预设:

-KA

[功能]

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

[应用]

指定选项 -KA 以输出汇编表。

[说明]

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

[使用示例]

输出汇编列表。

C>ra78k0 -c054 k0main.asm -ka

引用 K0MAIN.PRN。

汇编列表

ALNO	STNO	ADRS	OBJECT	MISOURCE	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				
14	14	----	DATA	DSEG AT OFE20H	
15	15	FE20	HDTSA: DS	1	
16	16	FE21	STASC: DS	1 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				
26	26	0000 11201A	MOV	HDTSA, #1AH	
27	27	0003 1620FE	MOVW	HL, #HDTSA ; set hex 2-code data in	
HL					
					:
					:

(b) -KS/-NKS**[语法]**

```
-KS
-NKS
```

默认预设:

-NKS

[功能]

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

[应用]指定选项 **-KS** 以输出符号表。**[说明]**

- 如果同时指定了选项 **-KS** 和选项 **-NKS**，则后指定的选项优先。
- 如果同时指定了选项 **-KS** 和选项 **-KX**，则会忽略选项 **-KS**。
- 如果指定了选项 **-NKA**、选项 **-NKS** 和选项 **-NKX**，则无法输出汇编列表文件。
- 如果已经指定选项 **-NP**，则选项 **-KS** 将无效。

[使用示例]

输出符号表。

```
C>ra78k0 -c054 k0main.asm -ks
```

引用 K0MAIN.PRN。

(汇编列表在符号列表之前输出。)

符号表格列表							
VALUE	ATTR	RTYP	NAME	VALUE	ATTR	RTYP	NAME
--H	CSEG		?CSEG		CSEG		CODE
		EXT	CONVAH		DSEG		DATA
FE20H	ADDR		HDTSA	0H	ADDR	PUB	MAIN
	MOD	SAMPM		0H	ADDR	PUB	START
FE21H	ADDR		STASC				

(c) -KX/-NKX**[语法]**

-KX -NKX

默认预设:

-NKX

[功能]

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

[应用]

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

[说明]

- 如果同时指定了选项 **-KX** 和选项 **-NKX**，则后指定的选项优先。
- 如果同时指定了选项 **-KS** 和选项 **-KX**，则会忽略选项 **-KS**。
- 如果指定了选项 **-NKA**、选项 **-NKS** 和选项 **-NKX**，则无法输出汇编列表文件。
- 如果已经指定选项 **-NP**，选项 **-KX** 将无效。

[注意]

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

$\Delta\$\Delta XREF$ $\Delta\$\Delta XR$; 缩写形式 $\Delta\$\Delta NOXREF$ $\Delta\$\Delta NOXR$; 缩写形式
--

有关控制指令的信息，请参阅语言部分的第 4 章控制指令。

【使用示例】

输出交叉引用表。

```
C>ra78k0 -c054 k0main.asm -kx
```

引用 KOMAIN.PRN。

汇编列表在交叉引用列表之前输出。

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

(8) 汇编列表文件格式指定

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

(a) -LW

[语法]

-LW [字符数]

默认预设:

-LW132 (显示输出时 80 个字符)

[功能]

选项 -LW 更改可在汇编列表文件中第 1 行上打印的字符数。

[应用]

指定选项 -LW 以更改可在汇编列表文件中第 1 行上打印的字符数。

[说明]

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

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

如果指定的数值超出该范围，或指定的不是数值，则会发生终止错误。

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

[注意]

可在源模块开始处介绍与 -LW 选项具有相等功能的控制指令。

△\$△WIDTH

有关控制指令的信息，请参阅 RA78K0 语言篇的“第 4 章控制指令”。

[使用示例]

指定 80 为汇编列表文件中每行的字符数。

C>ra78k0 -c054 k0main.asm -lw80

此引用汇编列表。

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

(b) -LL**[语法]**

-LL [行数]

默认预设:

-LL66 (显示输出时无页面分页符)

[功能]

选项 -LL 更改可在汇编列表文件中 1 页上打印的行数。

[应用]

指定选项 -LL 以更改可在汇编列表文件中 1 页上打印的行数。

[说明]

- 可使用选项 -LL 指定的行数范围显示如下。

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

如果指定的数值超出该范围，或指定的不是数值，则会发生终止错误。

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

[注意]

可在源模块开始处介绍与 -LL 选项具有相等功能的控制指令。

概要如下:

$\Delta\$ \Delta \text{LENGTH}$

有关控制指令的信息，请参阅语言部分的第 4 章控制指令。

[使用示例]

指定 20 为汇编列表文件中每页的行数。

C>ra78k0 -c054 k0main.asm -l20

此引用 KOMAIN.PRN。

```

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

Command:-c054 k0main.asm -l20
Para-file:
In-file:KOMAIN.ASM
Obj-file:KOMAIN.REL
Prn-file:KOMAIN.PRN

  Assemble list

-----

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

ALNO STNO ADRS  OBJECT   M I SOURCE STATEMENT

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

-----

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

ALNO STNO ADRS  OBJECT   M I SOURCE STATEMENT

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

(c) -LH

[语法]

-LH 字符串

默认预设:

空

[功能]

选项 -LH 指定汇编列表文件头标题栏中打印的字符串。

[应用]

- 指定选项 -LH 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，一眼就可明白汇编列表文件的内容。

[说明]

- 最多可在标题中指定 60 个字符。字符串不可包含空格。
 - 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。
1 个日文汉字或平假名记作 2 个字符。
- 如果每行的最大字符数少于或等于 119，则有效的字符串长度变化如下。

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

- 如果没有指定字符串的长度，则会发生终止错误。
- 如果指定了选项 -NP，则选项 -LH 不可用。
- 如果省略了 -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。

表5-4 可作为标题写入的字符

字符	命令行	参数文件
*?><	如果附上 " "，则可写入。	可写入。 即使附上 " "，同命令行所释。
;	如果附上 " "，则可写入。	不可写入。 (假设为注释。)
#	可写入。	不可写入。 (假设为注释。)
" (双引号)	不可作为一个有效的字符写入。	不可作为一个有效的字符写入。

表5-4 可作为标题写入的字符

字符	命令行	参数文件
00H	不可写入。	可写入。 然而，它可解释为字符串的结尾。
03H, 06H, 08H, 0DH, 0EH, 10H, 15H, 17H, 18H, 1BH, 7FH	不可写入。	可写入。 然而，这些字符将会在汇编列表文件中作为 '!' 出现。（单个 0DH 将不会输入到表中。）
01H, 02H, 04H, 05H, 07H, 0BH, 0CH, 0FH, 11H, 12H, 13H, 14H, 16H, 19H, 1CH, 1DH, 1EH, 1FH	可写入。 然而，这些字符将会在汇编列表文件中作为 '!' 出现。	可写入。 然而，这些字符将会在汇编列表文件中作为 '!' 出现。
1AH	可写入。 然而，该字符将会在汇编列表文件中作为 '!' 出现。	不可写入。 (文件的结尾)
字母字符	大写或小写字符按原样输入。	大写或小写字符按原样输入。
其他	可写入。	可写入。

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

[注意]

还可在源模块开始处写入与选项 -LL 具有相同功能的控制指令。

$\Delta\$ \Delta \text{TITLE} \Delta (\Delta \text{'character-string'} \Delta)$ $\Delta\$ \Delta \text{TT} \Delta (\Delta \text{'character-string'} \Delta)$; 缩写形式
--

有关控制指令的信息，请参阅语言部分的“第4章控制指令”。

[使用示例]

打印汇编列表文件头的标题。

C>ra78k0 -c054 k0main.asm -lhRA78k0_MAINROUTINE

此引用 78k0main.PRN。

```

78K/0 Series Assembler Vx.xx RA78k0_MAINROUTINE          Date:xx xxx xxxx Page: 1
      |
      标题

Command: -c054 k0main.asm -lhRA78k0_MAINROUTINE
Para-file:
In-file:KOMAIN.ASM
Obj-file:KOMAIN.REL
Prn-file:KOMAIN.PRN

Assemble list

ALNO STNO ADRS OBJECT M I SOURCE STATEMENT

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

```

(d) -LT

[语法]

-LT [字符数]

默认预设:

-LT8

[功能]

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

[应用]

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

[说明]

- 可使用选项 `-LT` 指定的字符数量范围显示如下。

$$0 \leq \text{可指定的字符数量} \leq 8$$

如果指定的数值超出该范围，或指定的不是数值，则会发生终止错误。

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

[注意]

可在源模块开始处介绍与 `-LT` 选项具有相等功能的控制指令。

概要如下：

<code>△\$△TAB△ tab-count</code>

有关控制指令的信息，请参阅语言部分的“**第 4 章控制指令**”。

[示例 1]

在省略选项 -LT 时引用 Sample.prn。

Assemble list						
ALNO	STNO	ADRS	OBJECT	M I	SOURCE	STATEMENT
1	1				NAME	SAMPLE
2	2					
3	3	----			CODE	CSEG
4	4	0000	63		MOV	A, B
5	5	0001	619A		SET1	A, 1
6	6				END	

[示例 2]

使用 HT 代码指定 1 个表格。

C>ra78k0 -c054 sample.asm -lt1

此引用 sample.prn。

Assemble list						
ALNO	STNO	ADRS	OBJECT	M I	SOURCE	STATEMENT
1	1				NAME	SAMPLE
2	2					
3	3	----			CODE	CSEG
4	4	0000	63		MOV	A, B
5	5	0001	619A		SET1	A, 1
6	6				END	

备注 由 HT 代码输入的表格数为 1。

(e) -LF/-NLF

[语法]

-LF -NLF

默认预设:

-NLF

[功能]

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

[应用]

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

[说明]

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

[注意]

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

$\Delta\$\Delta\text{FORMFEED}$ $\Delta\$\Delta\text{NOFOMFEED}$

有关控制指令的信息, 请参阅语言部分的第 4 章控制指令。

[使用示例]

在汇编列表文件结尾处添加一个换页代码。

```
C>ra78k0 -c054 k0main.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

默认预设:

-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。
- -NE 选项会使 -E 选项不可用。

[应用]

- 指定选项 -E 将错误消息保存到文件。
- 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]

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

[使用示例]

创建一个错误列表文件 (k0main.era)。

```
C>ra78k0 -c054 k0main.asm -ek0main.era
```

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

```
PASS 1 Start  
K0MAIN.ASM(31) : RA78K0 error E2202: Illegal operand  
PASS 2 Start  
K0MAIN.ASM(26) : RA78K0 error E2312: Operand out of range(byte)  
K0MAIN.ASM(31) : RA78K0 error E2202: Illegal operand
```

```
Target chip:uPD78054  
Device file:Vx.xx
```

```
Assembly complete, 2 error(s) and 0 warning(s) found.
```

此应用错误列表 k0main.era

```
PASS 1 Start  
K0MAIN.ASM(31) : RA78K0 error E2202: Illegal operand  
PASS 2 Start  
K0MAIN.ASM(26) : RA78K0 error E2312: Operand out of range(byte)  
K0MAIN.ASM(31) : RA78K0 error E2202: Illegal operand
```

(10) 参数文件指定**参数文件指定 (-F)****[语法]**

-F 文件名

默认预设:

没有输入文件。

[功能]

选项 -F 从指定的文件输入汇编器选项和输入的文件名。

[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。
- 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]

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

【使用示例】

使用参数文件执行汇编。

设置参数文件 (K0MAIN.PRA) 的内容如下。

```
; parameter file  
k0main.asm -osample.rel -g -c054  
-psample.prn
```

在命令行中输入以下内容。

```
C>ra78k0 -fk0main.pra
```

(11) 临时文件创建路径指定**临时文件创建路径指定 (-T)****[语法]**

-T	路径名
----	-----

默认预设:

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

[功能]

选项 **-T** 指定创建临时文件所在的路径。

[应用]

使用选项 **-T** 指定创建临时文件的位置。

[说明]

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

[使用示例]

指定临时文件输入到目录 **C:\TMP**。

```
C>ra78k0 -c054 k0main.asm -tc:\tmp
```

(12) 汉字码指定**汉字码指定 (-ZS/-ZE/-ZN)****[语法]**

```
-ZS
-ZE
-ZN
```

默认预设:

根据不同的操作系统，解释如下。

-ZS (Windows/HP-UX)

-ZE (SunOS/Solaris)

[功能]

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

[应用]

这些选项用于指定注释行汉字的汉字码的解释。

[说明]

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

```
△$△KANJICODE SJIS
△$△KANJICODE EUC
△$△KANJICODE NONE
```

有关控制指令的详细信息，请参阅语言手册的第 4 章控制指令。

- 3) 还可通过使用环境变量 LANF78K 指定汉字码。有关环境变量的详细信息，请参阅 11.2 准备开发环境（环境变量）。

[使用示例]

汉字码解释为 EUC 代码

```
A>ra78k0 k0main.asm -c054 -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

默认预设:

将会按以下顺序确定的路径读取设备文件。

- (i) 1) <..\dev> (适用于 ra78k0.exe 启动路径)
- (ii) 2) 启动 RA78k0 的路径
- (iii) 3) 当前目录
- (iv) 4) 环境变量 PATH

[功能]

从指定的路径读取设备文件。

[应用]

指定设备文件所在的路径。

[说明]

- 如果在指定选项 -Y 后未指定任何路径名, 则会发生终止错误。
- 如果在指定选项 -Y 后省略了路径名, 则会发生终止错误。
- 按如下所确定的顺序从其读取设备文件的路径。
 - (i). 由选项 -Y 所指定的路径
 - (ii). <..\dev> (适用于 ra78k0.exe 启动路径)
 - (iii). 启动 RA78k0 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:\78k0\dev。

```
C>ra78k0 k0main.asm -c054 -yc:\78k0\dev
```

(14) 符号定义指定**符号定义指定 (-D)****[语法]**

`-D 符号名 [=数值][, 符号名 [=数值]...]`

默认预设:

空

[功能]

定义符号。

[应用]在想要定义符号时，指定选项 `-D`。**[说明]**

- 分配给符号的数值必须是二进制、八进制、十进制或十六进制的数字。如果省略了数值的规格，则预设已指定了 1。
- 最多可指定 30 个字符，使用逗号分隔。
- 最多可指定 31 个字符作为符号名。
- 如果不止一次指定相同名称，最后指定的名称有效。
- 符号名中的字母区分大小写。

使用选项 `-D` 所定义的符号用于取代 `EQU/$SET/$RESET`。如果在源文件中还定义了为选项 `-D` 指定的符号名，则会发生错误。

[使用示例]

指定 2 为符号定义。

```
C>ra78k0 k0main.asm -c054 -dSYM=2
```

(15) 系列通用对象指定**系列通用对象指定 (-COMMON)****[语法]**

-COMMON

默认预设:

输出支持指定设备的目标文件

[功能]

选项 `-COMMON` 指定 78K0 系列通用目标模块文件的输出。

[应用]

该选项生成通用于 78K0 系列的目标码，独立于设备类型选定 `-C` 选项。
输出目标模块文件可以与指定 78K0 系列中的其他设备的目标文件连接。

[说明]

设置该选项以生成可以通用于 78K0 系列的目标码。

[注意事项]

- 即使已经指定选项 `-COMMON`, 设备指定选项 `-C` 或者拥有同样功能的控制指令仍然不可省略。
如果系列通用对象指定设置于所有要连接的输入对象模块文件，将会出错。

[使用示例]

```
C>ra78k0 k0sub.c -c054 -common
```

(16) 自编程指定

自编程指定(-SELF)

【语法】

-SELF

-默认

无

[功能]

选项-SELF 可防止访问越界错误，当调用"CALL!8100H"访问域外的地址 8100H（如，无 ROM 时）。

[应用]

-当使用自编程时使用该选项。

[说明]

当使用自编程出现"CALL!8100H"错误时使用该选项。

(17) 帮助指定

帮助指定 (--)

[语法]

--

默认预设:

无显示

[功能]

选项 - 显示帮助消息。

[应用]

帮助消息是汇编选项的说明列表。 执行汇编器时，请参阅这帮助消息。

[说明]

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

注意事项 该选项无法在 **PM +** 上指定。

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

【使用示例】

指定选项 - 时，帮助消息会输出在显示屏上。

```
c>ra78k0 -
```

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

usage : ra78k0 [option[...]] input-file [option[...]]
The option is as follows ([ ] means omissible).
-cx          :Select target chip. ( x = 012, 014 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 78k0 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 在 PM+ 中设置选项

本节介绍从 PM+ 设置汇编器选项的方法。

5.5.1 设置选项


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

图 5-2. <Assembler Options> 对话框（选择 <<Output1>> 标签时）

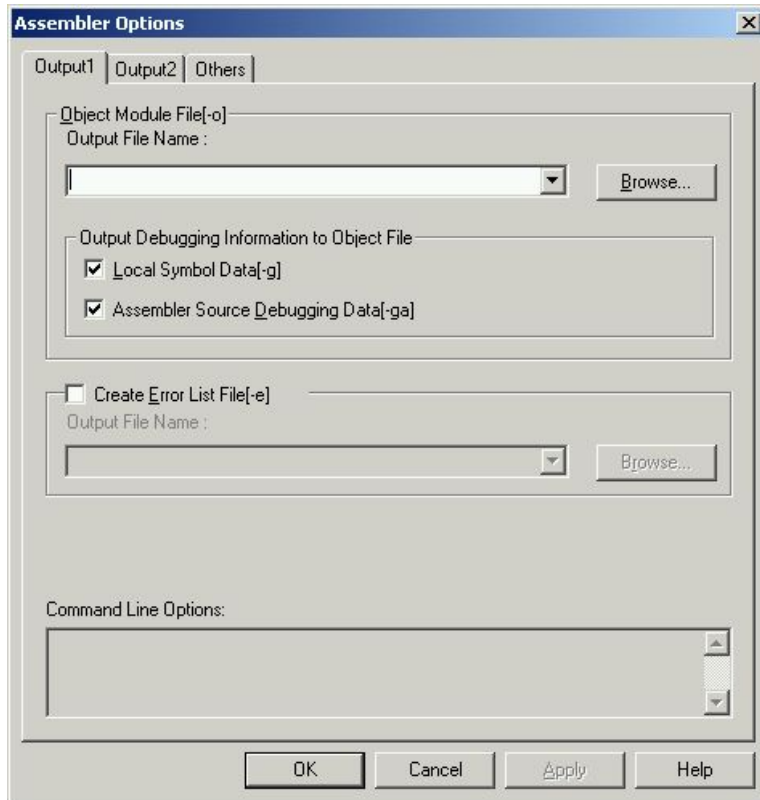


图 5-3. <Assembler Options> 对话框（选择 <<Output2>> 标签时）

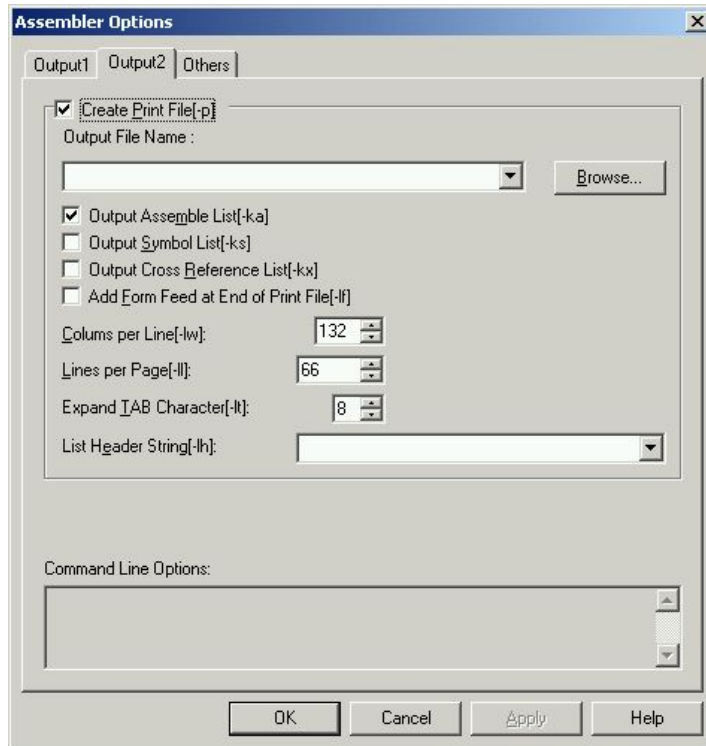
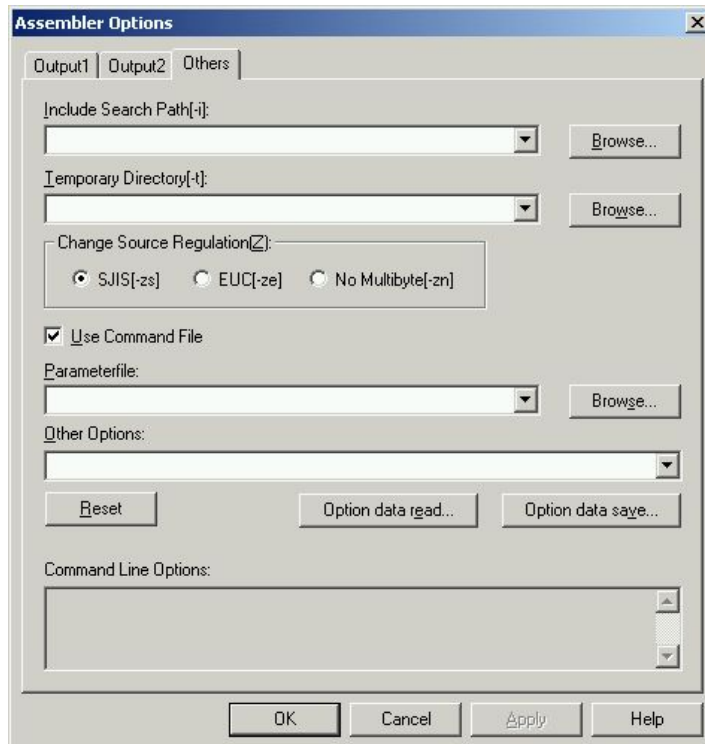


图 5-4. <Assembler Options> 对话框（选择 <<Others>> 标签时）



5.5.2 选项

<Assembler Options> 对话框中的各个选项介绍如下。

<< Output 1 >> 标签

- Object Module File [-o]
Output File Name (当选定 common 选项时)
通过使用 [Browse...] 按钮或直接输入路径, 指定目标文件的输出路径。
Output File Name (当选定 individual 选项时)
通过使用 [Browse...] 按钮或直接输入路径, 指定目标文件的输出路径及文件名。
- Local Symbol Data [-g]
执行包括局部符号的符号调试。
- Assembler Source Debugging Data [-ga]
添加汇编器源调试数据。
- Create Error List File [-e]
Output File Name: (当选定 common 选项时)
通过使用 [Browse...] 按钮或直接输入文件名, 指定错误列表文件的输出路径。
Output File Name: (当选定 individual 选项时)
通过使用 [Browse...] 按钮或直接输入文件名, 指定错误列表文件的输出路径及文件名。
- Command Object Module File [-common]
通过该选项将输出78K0系列通用目标模块文件
- Command Line Options
编辑框为只读。显示当前设定的选项字符串。

<< Output 2 >> 标签

- Create Print File [-p]
Output File Name: (当选定 common 选项时)
通过使用 [Browse...] 按钮或直接输入文件名, 指定目标模块汇编列表文件的输出路径。
Output File Name: (当选定 individual 选项时)
通过使用 [Browse...] 按钮或直接输入文件名, 指定目标模块汇编列表文件的输出路径及文件名。
- Output Assemble List [-ka]
在汇编列表文件中输出汇编表。
- Output Symbol List [-ks]
在汇编表后输出符号。

- **Output Cross Reference List [-kx]**
在汇编表之后输出交叉引用表。
- **Add Form Feed at End of Print File [-lf]**
在打印汇编列表文件的内容后添加换页代码。
- **Columns per Line [-lw]**
指定汇编列表文件中每行的字符数（从 72 到 2,046 字符数中选择）。
- **Lines per page [-ll]**
指定汇编列表文件中每页的行数（从 20 到 32,767 字符数中选择）。
- **Expand TAB character [-lt]**
指定制表符字符的长度（从 0 到 8 个字符选择）
- **List Header String [-lh]**
指定汇编列表文件头的标题框中要打印的字符串（最多 60 个字符）。
- **Command Line Options**
编辑框为只读。显示当前设定的选项字符串。

<< Others >> 标签

- **Include Search Path [-i]**
通过使用 [Browse...] 按钮或直接输入路径，指定读取包含文件的输出路径。
- **Temporary Directory [-t]**
通过使用 [Browse...] 按钮或直接输入路径，指定要创建的临时文件的位置。
- **Change Source Regulation [Z]**
选择要在源注释中使用的多字节代码类型 (SJIS[-zs], EUC[-ze], No Multibyte[-zn])。
- **Use Self Programming [-self]**
当使用自编程时使用该选项。
- **Use Command File**
选择该复选框创建指令文件。
- **Symbol Definition [-d]**
通过使用 [Edit] 按钮或者直接输入一个数字值来定义为符号。
- **Parameter file**
通过使用 [Browse...] 按钮或直接输入参数文件，指定所选的用户自定义的参数文件。

- **Other Options**

要指定一个非用复选框或单选按钮选择的选项，请在输入框中输入选项。

注意：帮助选项(-)在 PM+中不可使用。

- **Reset**

清除输入内容。

- **Option data read**

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

- **Option data save**

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

- **Command Line Options**

该编辑框是只读的。显示当前设置的选项字符串。

5.5.3 编辑选项对话框

项目在<Edit Option>对话框中以列表格式编辑。

<Edit Option>对话框如下所述。

图 5-5 <Edit Option>对话框

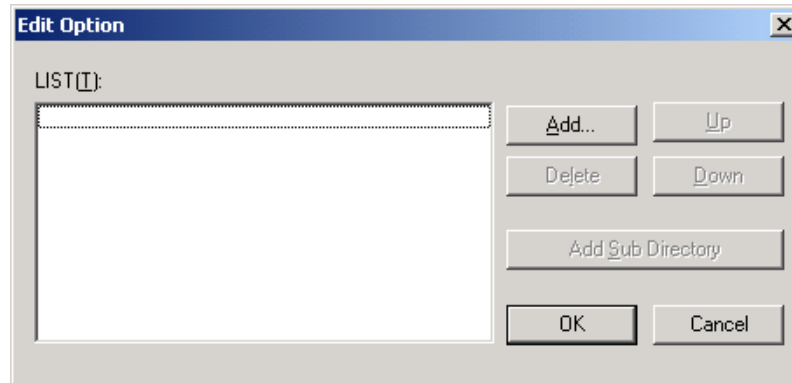
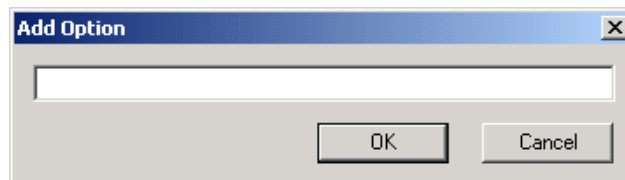


图 5-6 <Add Option>对话框



—[Add]按钮

添加一个列表项。

如果要添加的项目是一个文件或目录，则相应的 < Browse for Folder > 对话框打开。

在其他情况下，则会打开< Add Option >对话框。在此对话框中指定要添加的项目的详细信息。

—[Delete]按钮

删除选定项。

—[Up]按钮

将选定项上移。

—[Down]按钮

将选定项下移。

—[Add Sub Directory]按钮

当在<< Others >> 标签指定项目为 Include Search Path[-i](l)时，可以添加一个子目录到选定的列表

第 6 章 链接器

链接器输入一些由 78K0 系列汇编器输出的目标文件，确定位置地址并将这些文件输出为单个载入模块文件。链接器还输出列表文件，如连接列表文件、错误列表文件。

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

6.1 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	目标文件	<ul style="list-style-type: none">这些二进制文件包含机器语言数据的重定位和符号数据及机器语言数据的存储地址。这些文件由汇编器输出。	.REL
	库文件	<ul style="list-style-type: none">这些文件中包含两个（或更多）目标文件这些文件由库管理程序输出。	.LIB
	指令文件	<ul style="list-style-type: none">这些文件包含链接中用到的链接命令。这些文件由用户创建。	.DR
	参数文件	<ul style="list-style-type: none">这些文件包含程序执行的参数。这些文件由用户创建。	.PLK
输出文件	载入模块文件	<ul style="list-style-type: none">这些二进制映像文件包含了在链接中创建的所有数据。这些文件输入到对象转换器中。	.LMF
	链接列表文件	<ul style="list-style-type: none">为显示链接结果的列表文件。	.MAP
	错误列表文件	<ul style="list-style-type: none">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	<ul style="list-style-type: none">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 至 3)

6.2 链接器的功能

链接器的功能如下。

(1) 链接输入程序段

链接器确定并控制各程序段的存储地址。

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。如果源模块文件中指定了程序段的位置属性，则按照这些属性存放程序段。链接器还可以在其链接指令文件中指定位置属性。

(4) 校正目标码

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

6.3 存储空间和存储区

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

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

表6-2 程序段分配组（外部ROM等）

存储区名	默认地址	默认分配的段
ROM	内部 ROM: 若未内置 ROM 则到 RAM 的开头为止	CSEG
RAM	内部 RAM	DSEG, BSEG

- 备注**
1. 用指令文件来更改存储区的默认地址，或者用来指定写入程序的各项程序段的位置。
 2. 具体例子请参见“[3.4\(5\) 创建一个指令文件](#)”。

MEMORY、MERGE、AT、SEQUENT、COMPLETE

在指令文件中保留字不能用于表达其它含义（程序段名、存储区名等）。保留字可写成大写或小写字符，但不能大小混写。

示例 MEMORY
 memory
 Memory: 不能用

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

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

COMPLETE: 当两个（或更多）程序段同名时，出现错误。

示例 MERGE DSEG1:COMPLETE=RAM

(2) 符号

在指定程序段名、存储区名和存储空间名时是区分大小写字符的。

(3) 数值

要想在指令中为各个项指定数值常量，应以十进制或十六进制形式书写。其方法与源程序相同：在十六进制末尾加上“H”。如果开头出现 A 至 F，则先放上“0”。

示例 23H, 0FC80H

(4) 注释

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

示例 下划线部分为注释。

```
;DIRECTIVE FILE FOR 78054
MEMORY MEM1: (01000H, 1000H) #SECOND MEMORY AREA
```

6.4.2 存储器指令

存储器指令是定义存储区（已内置存储器的名称和地址）的指令。

已定义存储区的名称（存储区名）用来引用程序段位置指令。

最多可定义 100 个存储区，其中包括默认存储区。

[语法]

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

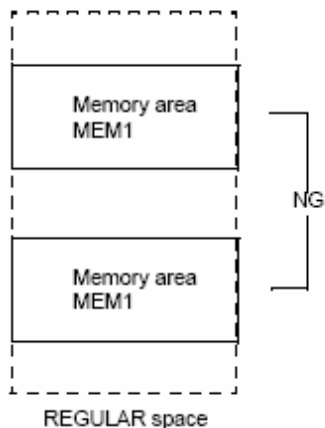
(1) 存储区名

指定已定义存储区的名称。指定存储区名的有关内容如下。

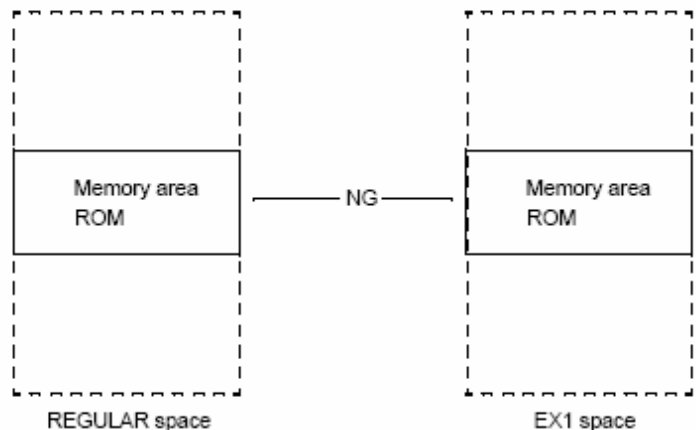
- 可用于存储区名的字符为 A 至 Z、a 至 z、0 至 9、_、?、@。但是，存储区名不能以 0 至 9 开头。
- 大小写字符认为是不同的字符。
- 大小写字符可混在一起使用。
- 存储区名的最大长度为 31 个字符。若使用了 32 个（或更多）字符，则会导致错误。
- 每个存储区名在整个存储空间中必须只出现在 1 个位置处。同一个存储区名不能用于不同的存储区，哪怕这些存储区位于不同的存储空间也不行。

图 6-1 存储区名

<相同存储区域的示例>



<不同存储区域的示例>



(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（Notes on Use）”。

(4) 存储区名

以下为 16 个 64KB 的存储区的名字。

REGULAR, EX1, EX2, EX3, EX4, EX5, EX6, EX7, EX8, EX9, EX10, EX11, EX12, EX13, EX14, EX15

使用内存地址来分配内存地址到某特定存储区。

对指定的要求如下。

- 存储空间名必须用大写字符指定。
- 当省略存储空间名时，假定指定为 REGULAR。
- 如果在写了“/”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。
- 可不止一次指定存储器指令。但是，以指定次序进行的多次定义将导致错误。
- 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”。

[使用示例]

将存储空间（EX1）中的 0H 至 1FFH 地址定义为 ROMA。

```
MEMORY ROMA: (0H, 200H)/EX1
```

6.4.3 程序段位置指令

程序段位置指令用来定位存储器指定区或特定地址的指定程序段。

[语法]

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

(1) 程序段名

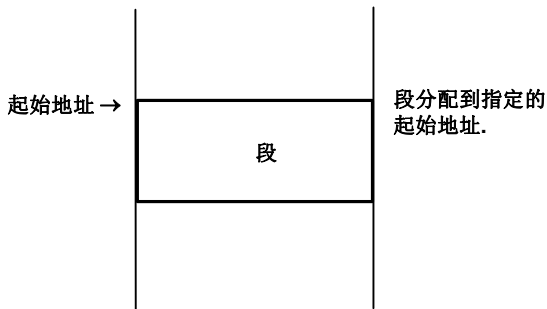
程序段名是输入链接器的目标文件中包含的程序段的名称。

- 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汇编源程序中相同。

(2) 起始地址

起始地址将程序段分配到“起始地址”指定的区。

- 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区范围，则会导致错误。
 2. 对于由程序段定义指令的 **AT** 指定或由 **ORG** 指令指定的存储地址的程序段来说，不能用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。

- 以下 16 个名字都可指定为存储区名
REGULAR, EX1, EX2, EX3, EX4, EX5, EX6, EX7, EX8, EX9, EX10, EX11, EX12, EX13, EX14, EX15
- 存储空间名必须用大写字母进行指定。
- 当省略存储空间名时，假定指定为 **REGULAR**。

确定程序段放置的目标地点的情况如下。

表6-4 存储区名指定及存储空间名共同确定的程序段位置

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

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

例如，如果“EX1”被指定为某具有重定位属性“CSEG.FIXED”段的存储空间名，该段将被定位于 800H 到 FFFH 之间。

【注意】

- 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。

【使用示例】

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

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

(1) 若输入程序段 SEG1 分配到 ROM 区的 500H 处(参考图 6-2 (1))。

```
MERGE SEG1: AT (500H)
```

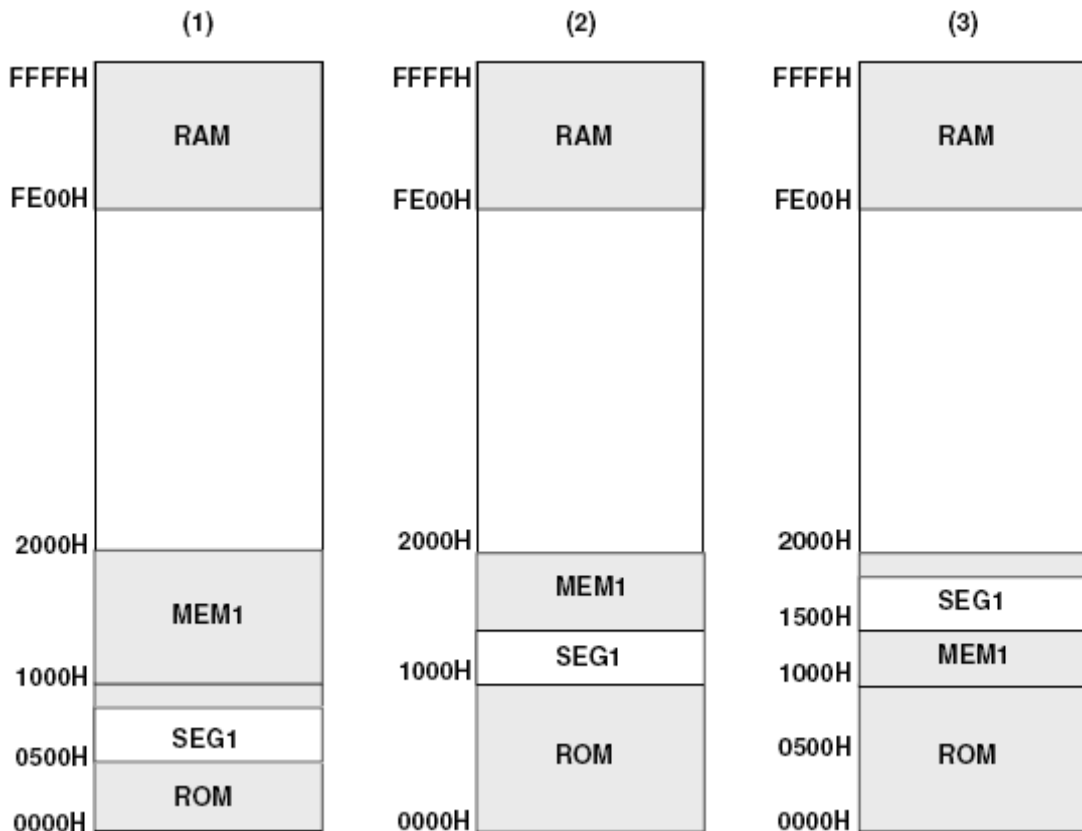
(2) 若输入程序段 SEG1 分配到存储区 MEM1 中(参考图 6-2 (2))。

```
MERGE SEG1:= MEM1
```

(3) 若输入程序段 SEG1 分配到存储区 MEM1 的 1500H 处(参考图 6-2 (3))。

```
MERGE SEG1: AT (1500H) = MEM1
```

图 6-2 程序段分配的具体例子



6.5 启动链接器

6.5.1 启动链接器

下面的两种方法可用来启动链接器。

(1) 从命令行启动

X>	[路径名]	lk78K0	[Δ选项]	...	Δ目标模块文件名	[Δ选项]...[Δ]
	(a)	(b)	(c)		e5)	(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，则利用空格将选项隔开。
- (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。
通过使用双引号来指定含有空格的文件名。

示例 C>lk78K0 k0main.rel k0sub.rel -ok0.lmf -g

(2) 从参数文件启动

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。

要想从参数文件启动链接器，应在命令行中指定参数文件说明选项（-F）。

从参数文件启动链接器，如下。

X>	LK78K0	[Δ目标模块文件]	Δ-f	参数文件名
			(a)	(b)

- (a) 参数文件说明选项
- (b) 包含启动链接器所需数据的文件

备注 用编辑器来创建参数文件。

书写参数文件内容的规则如下。

[[[Δ] 选项 [Δ选项] ... [Δ]Δ] ...

- 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。
- 目标模块文件名也可以写在选项的后面。
- 在参数文件中写入应在命令行中指定的所有链接器选项及输出文件名。

示例 用编辑器创建参数文件（K0.PLK）。

参数文件 K0.PLK 的内容：

```
;parameter file
k0main.rel k0sub.rel -ok0.lmf -pk0.map -e
-tc:\tmp
```

用参数文件 K0.PLK 启动链接器。

```
C>lk78K0 -fk0.plk
```

用参数文件 k0.plk 来启动链接器

```
c>lk78k0 -fk0.plk
```

6.5.2 执行开始结束信息

(1) 执行开始信息

在链接器启动时，显示屏上出现执行开始信息。

```
78K/0 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导致的链接错误，链接器就会将下列信息输出到显示屏上，并将控制权还给操作系统。

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

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

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

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

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

如果在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，退出链接，将控制权还给操作系统。

示例 1 指定了不存在的目标文件。

```
C>lk78K0 samp1.rel samp2.rel
78K/0 Series Linker Vx.xx [xx xxx xx]
Copyright (C) NEC Electronics Corporation xxxx, xxxx

RA78K0 error F3006: File not found 'SAMP1.REL'
RA78K0 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文件。导致错误，链接器异常中止链接。

示例 2 指定了不存在的链接器选项。

```
C>lk78K0 k0main.rel k0sub.rel -z
78K/0 Series Linker Vx.xx [xx xxx xx]
Copyright (C) NEC Electronics Corporation xxxx, xxxx

RA78K0 error F3018: Option is not recognized '-z'
Please enter 'LK78K0 --', if you want help messages.
Program Aborted.
```

在上面的例子中，指定了不存在的链接器选项。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

链接器选项是关于链接器操作的细节指令。链接器选项分成 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	
强制的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 (stack decision) 符生成指定	-S	自动生成栈判定的公共符号
	-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	
链接列表文件数据指定	-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中。
	-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	
链接列表文件格式指定	-LL	更改链接列表文件中 1 页能打印的行数。
	-LF	在编目文件的末尾插入换行代码。
	-NLF	

表6-5 链接器选项

分类	选项	说明
错误列表文件输出指定	-E	输出错误列表文件。
	-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	从指定路径读取库文件。
参数文件指定	-F	从指定文件输入文件名和选项。
临时文件创建路径的指定	-T	在指定路径创建临时文件。
设备文件搜索路径指定	-Y	从指定路径读取设备文件。
报警信息输出指定	-W	指定是否将报警信息输出到控制台。
闪存 ROM 模型的启动区 ROM 程序的链接指定	-ZB	指定闪存 ROM 区域的起始地址。
片上调试程序大小指定	-GO	指定片上调试程序的大小。
安全 ID 指定	-GI	指定一个安全 ID。
帮助指定	--	在显示屏上显示帮助信息。

6.6.2 链接器选项的先后顺序

表 6-6 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x标出的项]

当指定水平轴的选项时，垂直轴所示选项不可用。

示例 C>lk78K0 k0main.rel k0sub.rel -np -km

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。

示例 C>lk78K0 k0main.rel k0sub.rel -p -nkm -nkp -nkl -s

选项 -NKM、-NKP、-NKL 被同时指定，所以选项 -P 不可用。

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

示例 C>lk78K0 k0main.rel k0sub.rel -o -no

选项-NO 在-O 之后指定，所以-O 不可用，-NO 可用。

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

6.6.3 链接器选项说明

本节包含对各链接器选项的具体说明。

(1) 载入模块文件输出指定

载入模块文件输出指定 (-O/-NO)

[语法]

-O [输出文件名] -NO

默认预设:

-O 输入文件名.lmf

[功能]

- 选项-O 指定载入模块文件的输出。还指定输出的位置及文件名。
- 选项-NO使选项-O, -J, 和 -G无效。

[应用]

- 用选项-O 来指定载入模块文件的输出位置或更改其文件名。
- 当仅为了输出链接列表文件而进行链接时, 指定选项-NO。这将缩短链接时间。

[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。
- 若在指定选项-O 时省略‘输出文件名’, 则载入模块文件‘输入文件名.lmf’将输出到当前目录。
- 只要‘输出文件名’中指定了路径名, ‘输入文件名.lmf’就会输出到指定路径。
- 如果同时指定选项-O 和-NO, 则最后指定的选项优先。

[使用示例]

输出载入模块文件 k0.lmf。

```
C>lk78K0 k0main.rel k0sub.rel -ok0.lmf
```

(2) 强制的载入模块文件输出指定**强制的载入模块文件输出指定 (-J/-NJ)****[语法]**

-J -NJ

默认预设:

-NJ

[功能]

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

[应用]

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

[说明]

- 当指定选项 -J 时，即便出现致命错误也会输出载入模块。
- 若同时指定选项 -J 和 -NJ，则最后指定的选项优先。
- 若指定选项 -NO，则选项 -J 不可用。

[使用示例]

指定即便出现致命错误也要输出载入模块文件。

```
C>lk78K0 k0main.rel k0sub.rel -j
```

(3) 调试数据输出指定**调试数据输出指定 (-G/-NG)**

[语法]

-G -NG

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**使选项**-G**, **-KP**, 和 **-KL**不可用。

[应用]

当进行源程序级的符号调试时，一定要使用选项 **-G**。

[说明]

- 当指定选项 **-NG** 时，无法输出公共符号列表和局部符号列表。
- 若同时指定选项 **-G** 和**-NG**，则最后指定的选项优先。
- 若指定了选项 **-NO**，选项 **-G** 不可用。

[使用示例]

指定将调试数据添加到载入模块文件中。

```
C>lk78K0 k0main.rel k0sub.rel -g
```

(4) 栈判定符生成指定**栈判定符生成指定 (-S/-NS)****[语法]**

-S [存储区名]
-NS

默认预设:

-NS

[功能]

- 选项 -S 生成栈判定公共符号 '@STBEG' 和 '@STEND'。
- 选项 -NS 使选项 -S 不可用。

[应用]

指定选项 -S 来保留栈区。

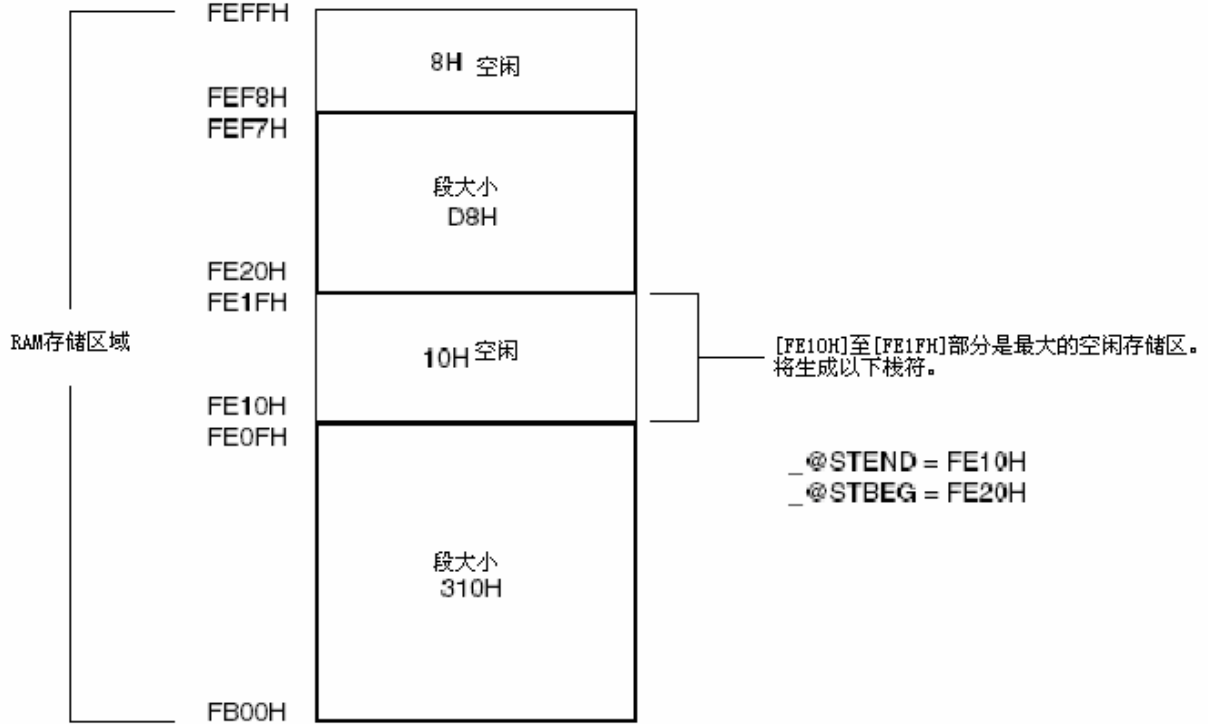
[说明]

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

[使用示例]

在存储区 RAM 中保留栈区（但是，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>lk78K0 k0main.rel k0sub.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]

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

[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，就会导致异常中止错误。
- 若省略文件名，则会导致异常中止错误。
- 不允许指令文件的嵌套。
- 指令文件中可指定的字符数无限制。
- 如果不止一次指定选项 -D，或如果指定了不止一个文件名，则会产生异常中止错误。
- 关于指令文件的详细说明请参见“6.4 链接指令”。

[使用示例]

重新定义默认存储区 ROM/RAM。

指令文件 K0.DR 的内容:

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

用 K0.DR 进行链接。

```
C>lk78K0 k0main.rel k0sub.rel -dk0.dr
```

(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。
- 当只为了输出载入模块文件而进行链接时，应指定选项 -NP。这将缩短链接时间。

[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，则会产生异常中止错误。
- 如果在指定选项 -P 时省略‘输出文件名’，则当前目录的链接列表文件名变为‘输入文件名.MAP’。
- 只要‘输出文件名’中指定了路径名，‘输入文件名.MAP’就会输出到指定路径。
- 若同时指定选项 -P 和-NP，则最后指定的选项优先。

[使用示例]

创建链接列表文件（K0.MAP）。

```
C>lk78K0 k0main.rel k0sub.rel -pk0.map
```

(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 输出映像列表到链接列表文件中。
- 选项 -NKM 使选项 -KM 不可用。

[应用]

指定选项 -KM 用于输出映像列表到链接列表文件中。

[说明]

- 如果选项 -NKM、-NKP 和 -NKL 全部指定，则无法输出链接列表文件。
- 若指定了选项 -NKM，则无法输出链接指令文件到链接列表文件中。
- 若同时指定选项 -KM 和 -NKM，则最后指定的选项优先。
- 若指定选项 -NP，则选项 -KM 不可用。

[使用示例]

将映像列表输出到链接列表文件 K0.MAP 中。

C>lk78K0 k0main.rel k0sub.rel -pk0.map -km

引用 K0.MAP。

78K/0 Series Linker Vx.xx

Date:xx xxx xxxx Page: 1

Command: k0main.rel k0sub.rel-pk0.map -km
 Para-file:
 Out-file: KOMAIN.LMF
 Map-file: K0.MAP
 Direc-file:
 Directive:

*** Link information ***

3 output segment(s)
 2FH byte(s) real data
 23 symbol(s) defined

*** Memory map ***

SPACE=REGULAR

MEMORY=ROM

BASE ADDRESS=0000H SIZE=8000H

OUTPUT SEGMENT	INPUT SEGMENT	INPUT MODULE	BASE ADDRESS	SIZE	
CODE			0000H	002H	CSEG AT
CODE	SAMPM		0000H	0002H	
* gap *		0002H	007EH		
?CSEG			0080H	0020H	CSEG
	?CSEG	SAMPM	0080H	0013H	
	?CSEG	SAMPS	0093H	001AH	
* gap *			00ADH	7F53H	

映象表

MEMORY=LRAM

BASE ADDRESS=FAC0H SIZE=0020H

OUTPUT SEGMENT	INPUT SEGMENT	INPUT MODULE	BASE ADDRESS	SIZE	
* gap *			FB00H	03H	
MEMORY = RAM					
BASE ADDRESS = FB00H	SIZE = 0500H				
DATA			FE20H	0003H	DSEG AT
	DATA	SAMPM	FE20H	0003H	
* gap *			FE23H	00DDH	
* gap (Not Free Area) *			FF00H	0100H	

(b) -KD/-NKD**[语法]**

-KD -NKD

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 使选项 -KD 不可用。

[应用]

指定选项 -KD 用于将链接指令文件输出到链接列表文件中。

[说明]

- 如果选项 -NKM、-NKP 和 -NKL 全部指定，则无法输出链接列表文件。
- 若指定了选项 -NKM，则无法将链接指令文件输出到链接列表文件中。
- 若同时指定选项 -KD 和 -NKD，则最后指定的选项优先。
- 若指定选项 -NP，则选项 -KD 不可用。

[使用示例]

将链接指令文件输出到链接列表文件（K0.MAP）中。

```
C>lk78K0 k0main.rel k0sub.rel -dk0.dr -pk0.map -kd
```

这里引用 K0.MAP。

```

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

Command:k0main.rel k0sub.rel -dk0.dr -pk0.map -kd
Para-file:
Out-file:K0MAIN.LMF
Map-file:K0.MAP
Direc-file:K0.DR                ← 指令文件名
Directive:      memory ROM: (0h, 1000h)      ← 指令文件的内容
                memory RAM: (0fe20h, 1000h)

*** Link information ***

    3 output segment(s)
    48H byte(s) real data
    23 symbol(s) defined

*** Memory map ***

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]

-KP -NKP

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** 使选项 **-KP** 不可用。

[应用]指定选项 **-KP** 用于将公共符号列表输出到链接列表文件中。**[说明]**

- 如果选项 **-NKM**、**-NKP** 和 **-NKL** 全部指定，则无法输出链接列表文件。
- 若指定了选项 **-NG**，则即便指定了选项 **-KP** 也无法输出公共符号列表。
- 若同时指定选项 **-KP** 和 **-NKP**，则最后指定的选项优先。
- 若指定选项 **-NP**，选项 **-KP** 将无效。

[使用示例]

将公共符号列表输出到链接列表文件（K0.MAP）中。

```
C>lk78K0 k0main.rel k0sub.rel -g -pk0.map -kp
```

这里引用 K0.MAP。

```

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

Command:k0main.rel k0sub.rel -g -pk0.map -kp
Para-file:
Out-file:K0MAIN.LMF
Map-file:K0.MAP
Direc-file:
Directive:

*** Link information ***

    3 output segment(s)
    2FH byte(s) real data
    23 symbol(s) defined

*** Memory map ***

SPACE = REGULAR

MEMORY = ROM
BASE ADDRESS = 0000  SIZE = 8000H

-----

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

*** Public symbol list ***

MODULE  ATTR  VALUE  NAME
SAMPM   ADDR  0000H  MAIN
SAMPM   ADDR  0080H  START
SAMPS   ADDR  0093H  CONVAH
      ⋮
Target chip:uPD78xxx
Device file:Vx.xx

```

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** 使选项 **-KL** 不可用。

[应用]指定选项 **-KL** 用于将局部符号列表输出到链接列表文件中。**[说明]**

- 如果选项 **-NKM**、**-NKP** 和 **-NKL** 全部指定，则无法输出链接列表文件。
- 若指定了选项 **-NG**，则无法输出局部符号列表。
- 若同时指定选项 **-KL** 和 **-NKL**，则最后指定的选项优先。
- 若指定选项 **-NP**，则选项 **-KL** 将无效。

[使用示例]

将局部符号列表输出到链接列表文件（K0.MAP）中。

C>lk78K0 k0main.rel k0sub.rel -g -pk0.map -kl

这里引用 K0.MAP。

```

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

Command:k0main.rel k4sub.rel -g -pk0.map -kl
Para-file:
Out-file:K0MAIN.LMF
Map-file:K0.MAP
Direc-file:
Directive:

*** Link information ***

    3 output segment(s)
    2FH byte(s) real data
    23 symbol(s) defined

*** Memory map ***

SPACE = REGULAR

:
-----

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

*** Local symbol list ***

MODULE   ATTR   VALUE   NAME
SAMPM    MOD           SAMPM
SAMPM    DSEG          DATA
SAMPM    ADDR    FE20H   HDTSA
SAMPM    ADDR    FE21H   STASC
SAMPM    CSEG          CODE
SAMPM    CSEG          ?CSEG
SAMPS    MOD           SAMPS
SAMPS    CSEG          ?CSEG
SAMPS    ADDR    00A4H   SASC
SAMPS    ADDR    00AAH   SASC1

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

局部符号列表

(8) 链接列表文件格式指定**链接列表文件格式指定 (-LL, -LF/-NLF)****(a) -LL****[语法]**

-LL [行数]

默认预设:

-LL66 (在显示输出的情况下不分页)

[功能]

选项 -LL 更改链接列表文件中 1 页能打印的行数。

[应用]

指定选项 -LL 用于更改链接列表文件中 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。

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

若指定为范围之外的数值或非数值, 则会产生中止错误。

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

[使用示例]

将链接列表文件每页行数指定为 20。

```
C>lk78K0 k0main.rel k0sub.rel -pk0.map -ll20
```

这里引用 K0.MAP。

78K/0 Series Linker Vx.xx

Date:xx xxx xxxx Page: 1

Command:k0main.rel k0sub.rel -pk0.map -li20

Para-file:

Out-file:K0MAIN.LMF

Map-file:K0.MAP

Direc-file:

Directive:

*** Link information ***

3 output segment(s)
2FH byte(s) real data

78K/0 Series Linker Vx.xx

Date:xx xxx xxxx Page: 2

23 symbol(s) defined

*** Memory map ***

*** Memory map ***

SPACE = REGULAR

MEMORY = ROM

BASE ADDRESS = 0000H

SIZE = 8000H

OUTPUT SEGMENT	INPUT SEGMENT	INPUT MODUL	BASE ADDRESS	SIZE
-------------------	------------------	----------------	-----------------	------

78K/0 Series Linker Vx.xx

Date:xx xxx xxxx Page: 3

CODE	0000H	0000002H	CSEG	AT
* gap *	CODE	SAMPM	0000H	0000002H
			0002H	0000007EH
?CSEG			0080H	00000046H CSEG
	?CSEG	SAMPM	0080H	0000002AH
	?CSEG	SAMPS	0093H	0000001CH
* gap *			00ADH	0000FF3AH

MEMORY = LRAM

BASE ADDRESS = FAC0H

SIZE = 0020H

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

:

(b) -LF/-NLF**[语法]**

-LF -NLF

默认预设:

-NLF

[功能]

- 选项 **-LF** 在链接列表文件末尾插入换页（FF）码。
- 选项 **-NLF** 使选项 **-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]

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

[使用示例]

在链接列表文件末尾添加换页代码。

```
C>lk78K0 k0main.rel k0sub.rel -pk0.map -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 [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** 使选项 **-E** 不可用。

[应用]

指定选项 **-E** 用于更改错误列表文件的输出目标和输出文件名。

[说明]

- 错误列表文件的文件名可指定为磁盘型文件名或设备型文件名。但是，若指定设备型文件名，**CLOCK** 则会产生中止错误。
- 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，错误列表文件将输出到当前驱动器。
- 若同时指定选项 **-E** 和 **-NE**，则最后指定的选项优先。

[使用示例]

创建错误列表文件（K0.ELK）。

```
C>lk78K0 k0main.rel k0sub.rel -dk0.dr -ek0.elk
```

在指令文件内容中产生了错误。引用 K0.ELK。

```
K0.DR ( 3 ) : RA78K0 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** 指定输入为库文件的文件。**[应用]**

- 链接器在库文件中检索输入模块引用的模块，仅将该模块与输入模块链接。
- 库文件用于在单个文件中注册两个（或更多）模块。
- 通过创建可在许多程序中通用的库文件，文件管理和操作变得更简单、更有效。指定选项 **-B** 来输入库文件到链接器中。

[说明]

- 只有磁盘型文件名可指定为文件名。
 - 文件名不可省略。
 - 如果指定了包含路径名的文件名，库文件就会从该路径输入。如果指定路径不存在库文件，就会产生错误。
 - 如果指定了不包含路径名的文件名，则库文件会从选项 **-I** 指定的路径或从默认搜索路径输入。
 - 若指定选项 **-B** 两次（或更多次），库文件会以指定的顺序进行输入。最多可指定 **10** 个 **-B** 选项。
- 注意 当在 **PM+** 中通过链接器选项对话框指定两个或更多库时，用逗号（，）隔开。
- 关于创建库文件方法的详细说明请参见“第八章 库管理程序”。

[使用示例]输入库文件（**K0.LIB**）。（库文件注册了 **K0SUB.REL**）。

```
C>lk78K0 k0main.rel -bk0.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) 路径名)

默认预设:

由环境变量 'LIB78K0' 指定的路径
若未指定路径则为当前路径

[功能]

选项 `-I` 指定从指定路径输入库文件。

[应用]

用选项 `-I` 从某个路径检索库文件。

[说明]

- 只有在选项 `-B` 指定了库文件名而未包含路径名时, 选项 `-I` 才是可用的。
- 可以指定两个 (或更多) `-I`。可指定两个 (或更多) 路径, 用 ';' 隔开。在 ';' 前后不能插入空白。
- 每个链接最多可指定 10 个路径名。当指定两个 (或更多) 路径名时, 链接器以指定的顺序搜索库文件。
- 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, 则会产生中止错误。
- 如果由选项 `-B` 指定了库文件而未包含路径名, 则链接器将以下列次序搜索路径。
 - a. 选项 `-I` 指定的路径
 - b. 由环境变量 'LIB78K0' 指定的路径
 - c. 当前路径

如果在这些路径下均未找到指定名称的库文件, 则会出现错误。

[使用示例]

从目录 `c:\LIB` 搜索库文件。

```
C>lk78K0 k0main.rel k0sub.rel -bk0.lib -ic:\lib
```

(12) 参数文件指定**参数文件指定 (-F)****[语法]**

-F 文件名

默认预设:

该选项及输入文件名只能用命令行输入。

[功能]

选项-F 指定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。如果希望每次进行链接时重复指定相同选项，可在参数文件中指定这些选项并指定选项 -F。

[说明]

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

[使用示例]

用参数文件进行链接。

如下设置参数文件（K0.PLK）的内容。

```
;parameter file  
k0main.rel k0sub.rel -ok0.lmf -pk0.map -e  
-tc:\tmp -g
```

在命令行中键入下列内容。

```
C>lk78K0 -fk0.plk
```

(13) 临时文件创建路径指定**临时文件创建路径指定 (-T)****[语法]**

-T 路径名

默认预设:

在环境变量 **TMP** 指定的路径创建临时文件。
未指定路径时, 在当前路径创建临时文件。

[功能]选项 **-T** 指定创建临时文件的路径。**[应用]**用选项 **-T** 来指定创建临时文件的位置。**[说明]**

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

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

指定临时文件输出到目录'TMP'。

```
C>lk78K0 k0main.rel k0sub.rel -ttmp
```

(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:

设备文件将从下列次序确定的路径读取。

- 1) <..\dev> (lk78K0.exe 的启动路径)
- 2) LK78K0 启动路径
- 3) 当前目录
- 4) 环境变量 PATH

[功能]

从指定路径读取设备文件。

[应用]

指定存在设备文件的路径。

[说明]

- 如果在选项 -Y 之后指定了非路径名，则产生异常中止错误。
- 如果在选项 -Y 后省略路径名，产生异常中止错误。
- 设备文件的读取路径按照如下次序确定。
 - (i). 选项 -Y 指定的路径
 - (ii). <..\dev> (lk78K0.exe 的启动路径)
 - (iii). LK78K0 的启动路径
 - (iv). 当前目录
 - (v). 环境变量 PATH

[使用示例]

将设备文件路径指定为目录 c:\78K0\dev。

```
C>lk78K0 k0main.rel k0sub.rel -yc:\78K0\dev
```

(15) 报警信息输出指定**报警信息输出指定 (-W)****[语法]**

-W [级]

默认预设:

-W1

[功能]

选项 -W 指定是否输出报警信息到控制台。

[应用]

指定输出报警信息的级别。

[说明]

- 如果在选项 -W 之后指定了非路径名，则产生异常中止错误。
- 只能指定 0 级、1 级、2 级。
- 有下列输出级别：
 - 0 ... 不输出报警信息。
 - 1 ... 输出正常报警信息。
 - 2 ... 输出详细报警信息。

有关输出报警条件的详细说明请参见“表 12-4 链接错误信息”。

[使用示例]

在选项 -W 中指定 2 级。

```
C>lk78K0 k0main.rel k0sub.rel -w2
```

(16) 闪存ROM模型的启动区ROM程序的链接指定

闪存ROM模型的启动区ROM程序的链接指定 (-ZB)

[语法]

-ZB

默认预设:

无链接指定

[功能]

指定闪存只读存储区的首地址。

[说明]

-指定闪存只读存储模块的启动区只读存储区程序链接

-若无地址被指定，将发生错误。

注意 不要为无闪存区自编程功能的设备指定该选项。

[使用示例]

C>lk78k0 k0main.asm -zb2000h

(17) 片上调试程序空间指定**片上调试程序空间指定(-GO)****[语法]**

-GO{空间}

默认预设:

片上调试未使用

[功能]

指定是否启用片上调试

[应用]可通过指定选项-**GO** 改变片上调试程序空间**[说明]**

- 如果指定非数值，将发生错误。
- 若无程序空间被指定，将默认设为 256 字节。
关于程序空间的细节，请参见 ID78K0 提供的文档。
- 如果选项-**GO**被指定，则段不能分配至地址02H ~ 03H和84H,以及从地址8FH起的指定程序空间+1的存储区。
- 如果为无片上调试功能的设备指定该选项，将发生错误。

[使用示例]

分配地址8FH~18FH(256位+1位)作为片上调试的程序分配存储区
C>lk78k0 k0main.rel -go256

(18) 安全ID指定**安全ID指定 (-GI)****[语法]**

-GI 安全ID

默认预设:

无安全ID

[功能]

指定安全 ID

[应用]

指定-GI 选项以设置安全 ID

[说明]

- 指定一个以“H”结尾的十六进制值。如果指定其他类型值，将出现中止错误。

- 指定 10 字节以内的安全 ID。如果指定的安全 ID 少于 10 字节，高位将以 0 填充。

- 安全 ID 设置于地址 85H~8EH。如果设置了安全 ID，则不能在 85H~8EH 再分配段。

- 如果在无安全 ID 功能的设备上指定该选项，将出现错误。

- 安全 ID 也可以通过定义以下汇编源中的重定位属性段来指定。但是，确保指定“SECUR_ID”作为段的重定位属性。

[任意段名]	CSEG	SECUR_ID	
		DB	11H
		DB	22H
		DB	33H
		DB	44H
		DB	55H
		DB	66H
		DB	77H
		DB	88H
		DB	99H
		DB	0AAH

如果汇编源中的指定与该选项的指定重复，该选项优先。

[注意] 如果未给有安全 ID 功能的设备指定该选项，任何代码都将可分配到该区域。

[使用示例]

-指定与上面汇编源所指定同样的" 112233445566778899AA"
C>lk78k0 k0main.rel -gi112233445566778899aah

(19) 帮助指定

帮助指定 (--)

[语法]

--

默认预设:

不显示

[功能]

选项 -- 在显示屏上显示帮助信息。

[应用]

帮助信息是链接器选项的说明表。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。
要想参考 **PM +** 帮助，可点击 <链接器选项设置> 对话框的“帮助”按钮。

【使用示例】

在指定选项 -- 时，显示屏上输出帮助信息。

C>lk78K0 --

```

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

usage: lk78K0 [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 ).
-zbaddress      :Create Boot file ( address:flash start address ).
-go [ n ]       :Change On-chip debug program size ( n = 256 to 1024 ).
-giid          :Set Security ID.
--             :Show this message.

Press RETURN to continue ...
DEFAULT ASSIGNMENT :-o -p -ne -km -kd -nkp -nkl -ll66 -nlf -ns -g -nj -w1

directive file usage:
MEMORY memory-area-name: (origin-value, size) [/memory-space-name]
MERGE segment-name: [location-type-definition] [merge-type-definition]
    [= memory-area-name] [/memory-space-name]

example:MEMORY ROM:(0H,4000H)
MEMORY RAMA:(0H,100H)/EX1
MERGE CSEG1:= ROM
MERGE DSEG1:AT(80H)

```

6.7 在 PM+ 中设置选项

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 ，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（当选择 << Output 1 >> 标签时）

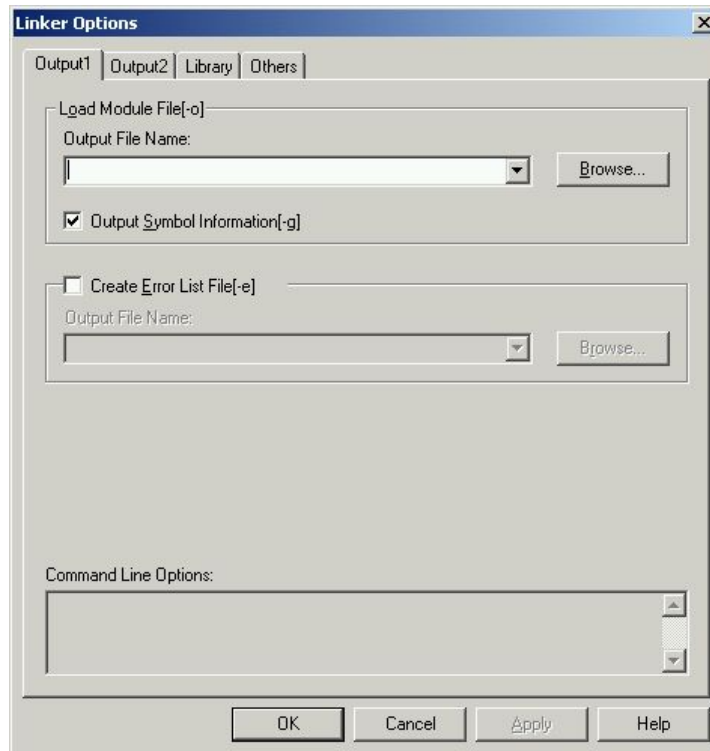


图 6-4 < Linker Options >对话框（当选择<< Output 2>>标签时）

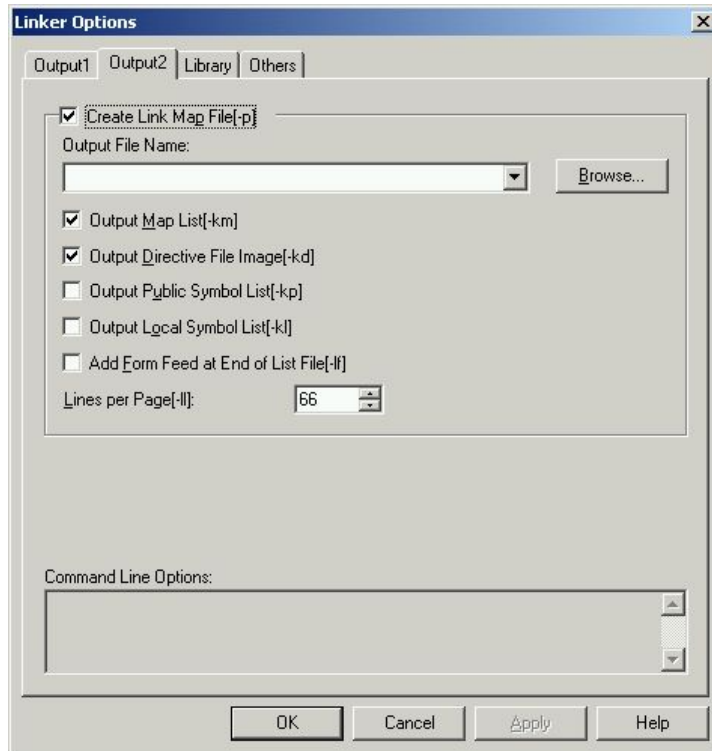


图 6-5 < Linker Options >对话框（当选择<<Library>>标签时）

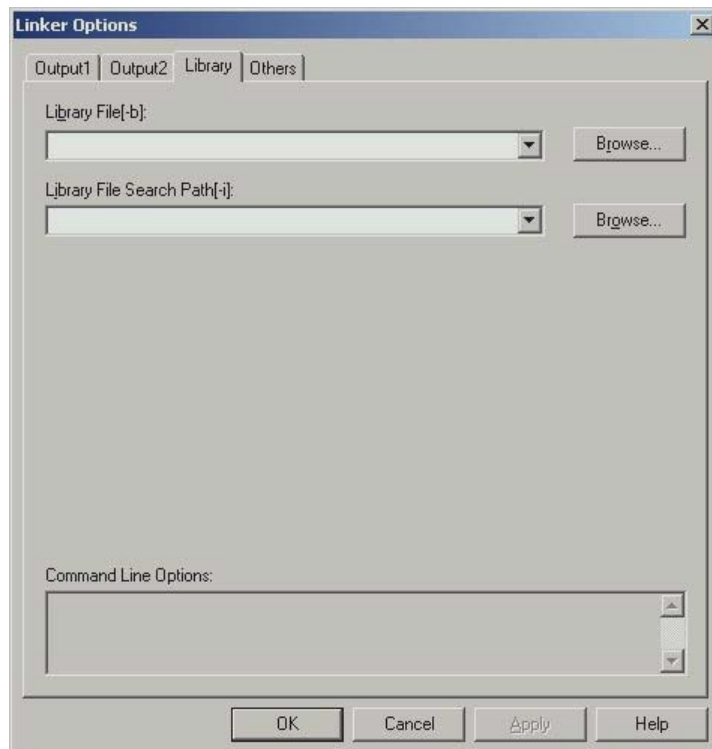
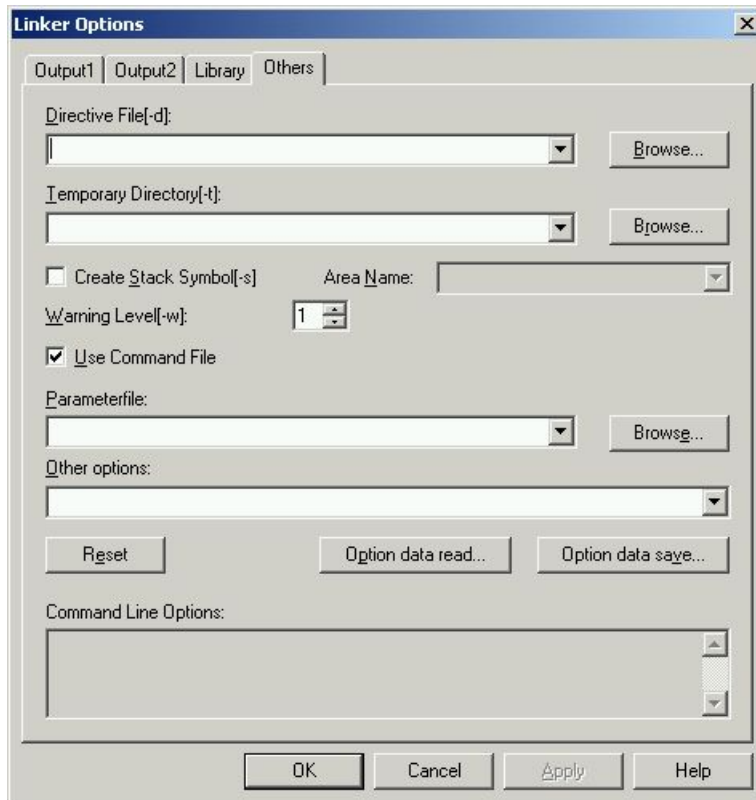


图 6-6 < Linker Options >对话框（当选择<<Others>>标签时）



6.7.2 选项

下面介绍 < Linker Options > 对话框中的各选项。

<< 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]**
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]**
通过该选项输出错误列表文件
输出文件名
要想输出错误列表文件, 应用[Browse...]按钮或直接在输入框中输入路径及文件名。
- **On-Chip Debug [-go]**
通过该选项启用片上调试功能
SIZE
指定片上调试程序空间大小
- **Security ID [-gi]**
通过该选项指定安全 ID
ID
指定安全 ID
- **Command Line Options**
编辑框为只读。显示当前设定的选项字符串。

<< Output2 >> 标签

- **Create Link Map File [-p]**
选择该复选框来输出链接列表文件。
Output File Name
通过[Browse...]按钮或直接输入, 指定链接列表文件的输出路径。
- **Output Map List [-km]**
在链接列表文件中输出映象文件。
- **Output Directive File Image [-kd]**
在链接列表文件中输出链接指令文件。

- **Output Public Symbol List [-kp]**
在链接列表文件中输出公共符号列表。
- **Output Local Symbol List [-kl]**
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]**
在打印链接列表文件内容后附加分页代码。
- **Lines per Page [-ll]**
指定链接列表文件每页的行数（可选 20 至 32767 字符）。
- **Command Line Options**
编辑框为只读。显示当前设定的选项字符串。

<< Library >> 标签

- **Library File [-b]**
通过[Edit...]按钮或直接输入，指定将指定文件输入为库文件。
- **Library File Search Path [-i]**
通过[Edit...]按钮或直接输入，指定从指定路径输入库文件。
- **Command Line Options**
编辑框为只读。显示当前设定的选项字符串。

<< Others >> 标签

- **Directive File [-d]**
通过[Browse...]按钮或直接输入，指定输入指定文件为指令文件。
- **Temporary Directory [-t]**
通过[Browse...]按钮或直接输入，指定临时文件的创建路径。
- **Create Stack Symbol [-s]**
当选中该选项时，保证存储区中的最大空区为栈区。
- **Area Name**
指定用户定义的存储区名或默认存储区名。
- **Warning Level [-w]**
指定报警信息的输出级别。
0: 不输出报警信息。
1: 输出正常报警信息。
2: 输出详细报警信息。

- **Flash Start Address for the Product with Flash ROM [-zb]**
指定 flash 内存产品的启动存储区起始地址
注意 不要为无 flash 内存自编程功能的设备指定该选项

- **Use Command File**
选择此复选框来创建命令文件。

- **Parameterfile**
通过[Browse...]按钮或直接输入，读取用户定义的参数文件。

- **Other options**
如果在复选框或单选按钮可选择的选项之外还希望指定其它选项，可在输入框中输入该选项。

- **Reset**
使输入内容复位。

- **Option data read...**
打开 <选项数据读取>对话框，在指定选项数据文件后该文件被读取。

- **Option data save...**
在打开<选项数据保存>对话框后，将选项数据文件保存到选项数据文件中并命名。

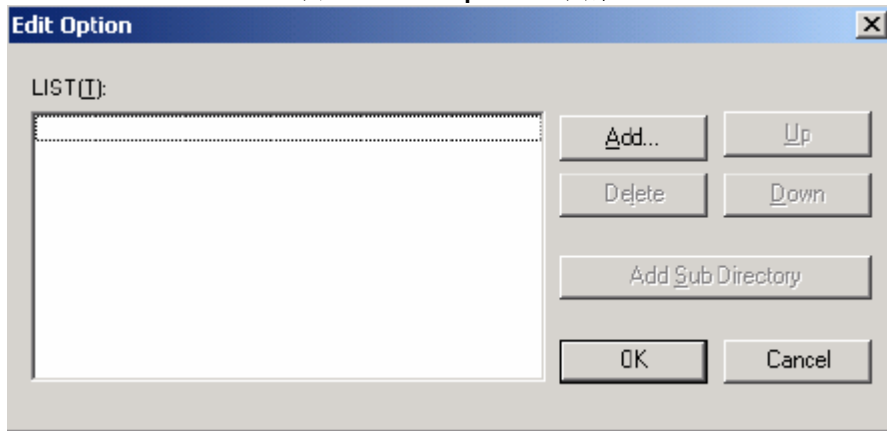
- **Command Line Options**
此编辑框只读。显示当前设置选项字符串。

6.7.3 编辑选项对话框

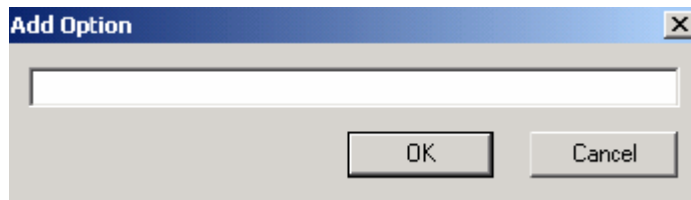
<Edit Option>对话框中的各项以列表格式编辑。

<Edit Option>对话框描述如下：

图 6-7 < Edit Option > 对话框



图标 6-8 < Add Option > 对话框



-[Add]按钮

添加列表项

如果需添加项目是文件或目录，将会打开相应的<Browse for Folder>对话框

在所有情况中，都会打开<Add Option>对话框。指定该项的具体内容添加到列表框

-[Delete]按钮

删除选定项

-[Up]按钮

将选定项上移

-[Down]按钮

将选定项下移

-[Add Sub Directory]按钮

当该选定项于<<Library>>标签被指定为库文件搜索路径[-i]时，可将子目录附加于该选项。

第7章 目标转换器

目标转换器输入由 RA78K0 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。

当产生目标转换器错误时，显示屏上出现错误信息对出错原因进行说明。

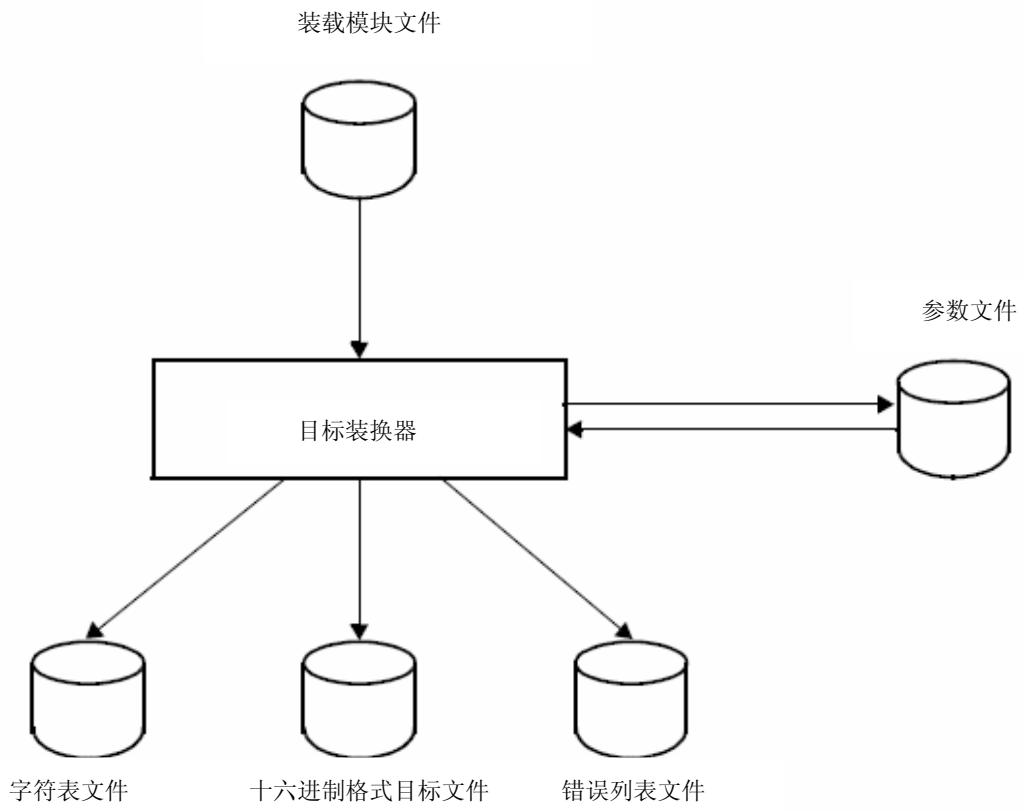
7.1 目标转换器的 I/O 文件

目标转换器的 I/O 文件如下所示。

表7-1 目标转换器的I/O文件

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

图 7-1 目标转换器的 I/O 文件



7.2 目标转换器的功能

7.2.1 目标转换器如何处理扩展空间

当代码被输出至扩展内存的空间段时，目标转换器为每个空间生成独立的十六进制格式目标模块文件。

要为每个空间输出独立的十六进制文件，指定内存空间并合并链接指令文件中的目录。要了解链接指令文件，请参见“6.4.1 指令文件”。

当输出代码到位于扩展空间中的段时，对象转换器为每个空间生成一个单独的十六进制格式对象模型文件。所有具有NUMBER属性的符号被输出至为普通空间生成的符号表格文件。

表 7-2 显示了十六进制目标模块文件和为扩展空间生成的符号表格文件的文件类型。

表7-2: 扩展空间的输出文件类型

文件	常规空间	扩展空间							
	普通	EX1	EX2	EX3	EX4	...	EX13	EX14	EX15
十六进制	.hex	.H1	.H2	.H3	.H4H13	.H14	.H15
符号	.sym	.S1	.S2	.S3	.S4S13	.S14	.S15

7.2.2 flash 只读存储器自重写模式支持

当启用 flash 只读存储区自重写模式时，目标转换器能在启动区和 flash 存储区为位于 flash 只读存储区的代码创建独立的十六进制目标模块文件。要输出十六进制文件，指定目标转换器选项-ZF。文件类型如下：

表 7-3 指定选项-ZF 时的文件类型

文件	文件类型
输出文件至启动只读存储区程序端	.hxb
输出文件至非启动只读存储区	.hxf

7.2.3 十六进制格式目标文件

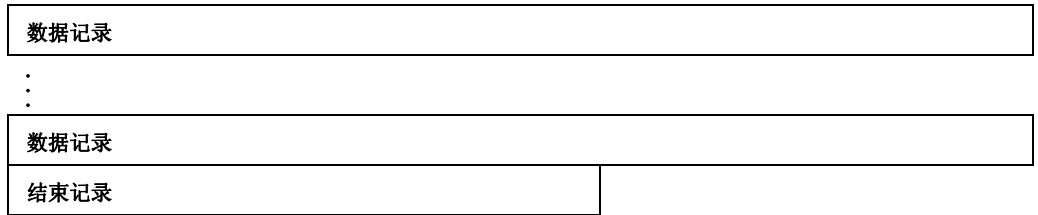
目标转换器输出的十六进制格式目标文件可以输入 HEX 载入程序中，如 PROM 编程器或调试器。

下面是示例程序的十六进制格式目标文件。

```
:0200000080007E  
:1000800011201A1620FE9A93001421FE63958462B3  
:1000900095FAFE617131809AA40073617131809A82  
:0D00A000A40072AF4D8D020D070D30AFA8  
:00000001FF
```

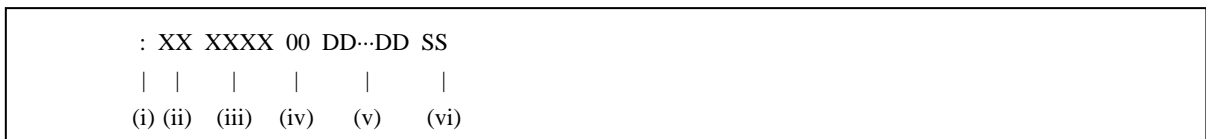
[Intel 标准 HEX 格式目标文件格式]

图 7-2 Intel 标准格式



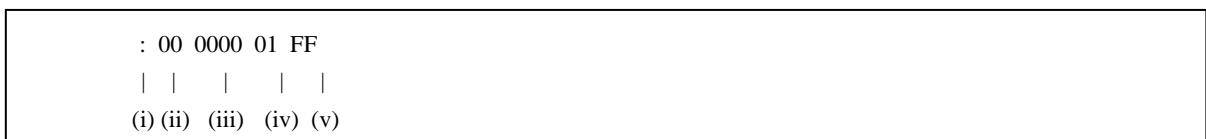
注 这里重复了数据记录。

(1) 数据记录



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

(2) 结束记录



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

[英特尔扩展十六进制目标模块文件格式]

图 7-3 英特尔扩展格式



注意：这里扩展地址记录和数据记录是重复的

(1) 扩展地址记录

:	02	0000	02	XXXX	SS
↑	↑	↑	↑	↑	↑
(i)	(ii)	(iii)	(iv)	(v)	(vi)

- (i) 记录标记
表示记录开始
- (ii) 代码数，固定为 02
- (iii) 0 固定为 0000。
- (iv) 记录类型，固定为 02。
- (v) 段的段落值
段的段落值以四位十六进制数字显示。
- (vi) 校验和（2 位）
将输入值按顺序去减 0 得到的值。

(2) 数据记录

:	XX	XXXX	00	DD ... DD	SS
↑	↑	↑	↑	↑	↑
(i)	(ii)	(iii)	(iv)	(v)	(vi)

- (i) 记录标记
表示记录开始

- (ii) 代码数 (2 位)
代码中保存在记录中的字节数。最多可以存储 16 字节。
- (iii) 位置地址 (偏移)
记录中显示代码的起始位置 (偏移) 以四位十六进制数字显示。
- (iv) 记录类型 (2 位)
固定为 00H。
- (v) 代码 (最多 32 位)
目标代码每次显示一个字节, 其中 4 个较高位和 4 个较低位分开。在代码中最多可以表示 16 字节。
- (vi) 校验和 (2 位)
将输入值按顺序去减 0 得到的值。

(3) 起始地址码

:	04	0000	03	0000	0000	F9
↑	↑	↑	↑	↑	↑	↑
(i)	(ii)	(iii)	(iv)	(v)	(vi)	(vii)

- (i) 记录标记
- (ii) 固定为 04
- (iii) 固定为 0000
- (iv) 固定为 03
- (v) 固定为 0000
- (vi) 固定为 0000
- (vii) 固定为 F9

(4) 结束记录

:	00	0000	01	FF
↑	↑	↑	↑	↑
(i)	(ii)	(iii)	(iv)	(v)

- (i) 记录标记
- (ii) 固定为 00
- (iii) 固定为 0000
- (iv) 固定为 01
- (v) 固定为 FF

[扩展技术十六进制格式目标模块文件格式]

十六进制文件由三类块组成。

- (i) 数据块
- (ii) 符号块（这个块不使用。符号数据使用符号列表文件。）
- (iii) 终止块

每个块由一个包含 6 个字符的标题字段开始，以串的行结尾结束。每个块的最大长度是 255，不包括开始符号 % 和行结尾。

普通标题字段的格式如表 7-4 所示。

表7-4 扩展技术标题字段

项目	ASCII 字符的数量	解释
%	1	百分号表示此块位为扩展技术格式。
块长度	2	这是一个 2 位的十六进制数字，表示块中的字符数。字符数不包括起始符 % 和行结尾。
块类型	1	6=数据块 3=符号块 8=终止块
校验和	2	这是一个 2 位的十六进制数字，表示当块中字符的总值（不包括开始符 % 和行结尾）除以 256 的余数。字符的总值如表 7-5 所示。

表7-5 校验和评估的字符值

字符	值（十进制）
0-9	0-9
A-Z	10-35
\$	36
%	37
.（小数点）	38
_（下划线）	39
a-z	40-65

(1) 数据块

数据块的格式如表 7-6 所示。

表7-6 扩展技术的数据块格式

字段	ASCII 字符的数量	解释
标题	6	标准标题字段 块类型=6
加载地址	2-17	目标代码加载的地址。 字符数是变量。
目标代码	2n	字节数 n，显示为一个 2 位的十六进制数。

注意：在扩展技术中，一个具体的字段中的字符数是 2-17 范围内的变量（实际数据中的 1-16 个字符）。这个变量字段的第一个字符表示字段的长度。数字 0 表示包括 16 个字符的字符行。因此，字符串的长度为 1-16 个字符，变量长度字段的长度包括字符串长度指示符 2-17。

%	15	6	1C	3	100	0202020202
↑	↑	↑	↑	↑	↑	↑
(i)	(ii)	(iii)	(iv)	(v)	(vi)	(vii)

- (i) 标题字符
- (ii) 块长度：15H=21
- (iii) 块类型：6
- (iv) 校验和：1CH
- (v) 加载地址中的位数
- (vi) 加载地址：100H
- (vii) 目标代码：6 字节

(2) 终止块

终止块的格式如表 7-7 所示。

表7-7 扩展技术的终止块格式

字段	ASCII 字符的数量	解释
标题	6	标准标题字段 块类型=8
加载地址	2-17	程序执行的开始地址。 字符数是变量。

%	08	8	1A	2	80
↑	↑	↑	↑	↑	↑
(i)	(ii)	(iii)	(iv)	(v)	(vi)

- (i) 标题字符
 - (ii) 块长度: 8H
 - (iii) 块类型: 8
 - (iv) 校验和: 1AH
 - (v) 加载地址中的位数
 - (vi) 加载地址: 80H

(3) 符号块

扩展技术符号块用于象征性的调试。其中应有以下符号:

- 符号: 1 到 16 个大小写字母、数字、小数点和下划线。起始字符不能为数字。
- 值: 最多为 64 位 (16 位十六进制数字)。
- 类型: 地址或标量 (表示地址外数值的标量)。地址分为代码地址 (指令地址) 和数据地址 (数据项的地址)。
- 全局/局部指定: 指定一个符号为全局 (允许外部引用) 的还是局部的。
- 段成员: 一个区域可以被认为是内存名给定的范围。一个程序中的每个地址至少属于一个区域。标量不属于任何区域。

符号块的格式如表 7-8 所示。

表7-8 扩展技术的符号块格式

字段	ASCII 字符的数量	解释
标题	6	标准标题字段 块类型=3
段名称	2-17	段名称 2-17。包含块中定义的符号的段名称。 字符数是变量。
段定义	5-35	每个符号块必须有一个此类字段。此字段可以放在字符定义字段数的前面或后面。此格式在表 7-9 中显示。
符号定义	每个符号 5-35	这是一个大于 0 的符号定义字段, 如表 7-10 所示。

程序中包含的符号以符号块传递。每个块包括一个段名称和一系列属于该段的符号。如果有必要, 段中可以包括标量。

同一个段中的一个符号可以放在一个或多个块中。

符号块中的段定义字段和符号定义字段的格式如表 7-9 和表 7-10 所示。

表7-9 扩展技术的符号块段定义字段

字段	ASCII 字符的数量	说明
0	1	0 表示该字段是段定义字段。
基址	2-17	这是一个段起始地址。 字符数是变量。
长度	2-17	表示字段长度。 字符数是变量，计算如下： 1-(较高地址-基址)

表7-10 扩展技术的符号块段定义字段

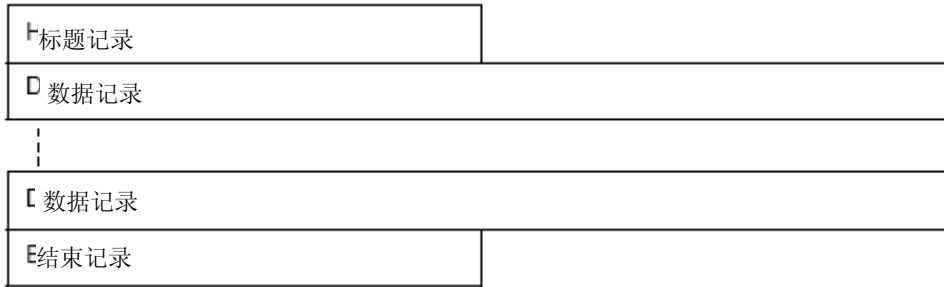
字段	ASCII 字符的数量	说明
类型	1	1 位十六进制数值，表示全局/局部符号指定和显示值的类型。 1=全局地址 2=全局标量 3=全局代码地址 4=全局数据地址 5=局部地址 6=局部标量 7=局部代码地址 8=局部数据地址
符号	2-17	表示符号长度。变量
数值	2-17	与符号对应的值。字符数是变量。

[摩托罗拉 S 型格式]

摩托罗拉 S 型格式文件由 5 个记录转为 3 种类型。整个文件的组成如图 7-4 所示。记录的类型如表 7-11 所示。

表7-11 摩托罗拉十六进制文件记录类型

项目	记录类型
标题记录（可选）	S0
数据记录	S2（标准 24 位） S3（32 位）
结束记录	S8（标准 24 位） S7（32 位）

图 7-4 摩托罗拉 S 型格式

摩托罗拉十六进制文件被分为 24 位地址和 32 位地址。标准的地址由记录 S0、S2 和 S8 组成。32 位地址由记录 S0、S3 和 S7 组成。标题记录 S0 是可选的，不输出。CR 符号放在每个 S 记录结束的位置。

每个记录中的每个字段的一般格式及其含义如表 7-12 和 7-13 所示。

表7-12 每个记录的一般格式

记录类型	一般格式
S0	S0XXYY ... YYZZZ
S2	S2XXWWWWWDD ... DDZZ
S3	S3XXWWWWWDD ... DDZZ
S7	S7XXWWWWWZZ
S8	S8XXWWWWWZZ

表7-13 字段和含义

字段	含义
Sn	记录类型
XX	数据记录的长度 地址、十六进制数据和校验和中的字节数
YY ... YY	文件名 由十六进制数字表示的输入文件名的 ASCII 码。
WWWWWW [WW]	第 24 (第 32) 位地址
DD ... DD	十六进制数据 一个字节的数表示为 2 位的十六进制数字
ZZ	校验和 记录长度、数据和十六进制数据每个字节和的补码 1 的低 1 个字节。

S2	08	00FF11	D4520A20	A0
↑	↑	↑	↑	↑
(i)	(ii)	(iii)	(iv)	(v)

- (i) 记录类型: S2
- (ii) 记录长度: 8 字节
- (iii) 加载地址 (24 位地址)
- (iv) 十六进制数据
- (v) 校验和

(1) S0 记录

S0	XX	YYYYYYYY	ZZ
↑	↑	↑	↑
(i)	(ii)	(iii)	(iv)

- (i) 记录类型
- (ii) 记录长度
记录长度为(iii)中的字节数加上(iv)中的字节数。
- (iii) 文件名
- (iv) 校验和

(2) S2 记录

S2	XX	WWWWWWW DD ... DD	ZZ
↑	↑	↑	↑
(i)	(ii)	(iii)	(iv)

- (i) 记录类型
- (ii) 记录长度
记录长度为(iii)中的字节数加上(iv)中的字节数再加上(v)中的字节数。
- (iii) 加载地址
加载地址是(iv)中数据在 0H 到 FFFFFFFH 范围内的 24 位加载地址。
- (iv) 数据
这里指的是加载的数据本身。
- (v) 校验和

(3) S3 记录

S3	XX	WWWWWWW DD ... DD	ZZ
↑	↑	↑	↑
(i)	(ii)	(iii)	(iv)

- (i) 记录类型
- (ii) 记录长度
记录长度为(iii)中的字节数加上(iv)中的字节数再加上(v)中的字节数。
- (iii) 加载地址
加载地址是(iv)中数据在 0H 到 FFFFFFFH 范围内的 24 位加载地址。
- (iv) 数据
这里指的是加载的数据本身。
- (v) 校验和

(4) S7 记录

S7	XX	WWWWWWW ZZ
↑	↑	↑
(i)	(ii)	(iii)

- (i) 记录类型
- (ii) 记录长度
记录长度为(iii)中的字节数加上(iv)中的字节数。
- (iii) 输入地址
这里是在 0H 到 FFFFFFFH 范围内的 32 位输入地址。
- (iv) 校验和

(5) S8 记录

S7	XX	XXXXXXXXXX	ZZ
↑	↑	↑	↑
(i)	(ii)	(iii)	(iv)

- (i) 记录类型
- (ii) 记录长度
记录长度为(iii)中的字节数加上(iv)中的字节数。
- (iii) 输入地址
这里是在 0H 到 FFFFFFFFH 范围内的 24 位输入地址。
- (iv) 校验和

7.2.4 符号列表文件

目标转换器输出的符号列表文件输入到调试器中。
下面是示例程序的符号列表文件。

```
#04
;FF PUBLIC
010097CONVAH
010000MAIN
010080START
00FE20_@STBEG
00FB00_@STEND
;FF SAMPM
<02FE20HDTSA
02FE21STASC
;FF SAMPS
<0100A8SASC
0100AESASC1
=
```

[符号列表文件格式]

符号表开头	#	0	C	LF		
公共符号开头	;		4 个空格	PUBLIC	C	LF
注 1→		符号属性	符号值	公共符号名	C	LF
		•	•	•	•	•
		•	•	•	•	•
		•	•	•	•	•
局部符号开头	;	F	C	模块名 1	C	LF
	<	符号属性	符号值	局部符号名	C	LF
		符号属性	符号值	局部符号名	C	LF
		•	•	•	•	•
		•	•	•	•	•
		•	•	•	•	•
以对象模块为单位进行重复	;	F	4 个空格	模块名 2	C	LF
		•	•	•	•	•
		•	•	•	•	•
		•	•	•	•	•
符号表结束标记	=	C	LF			

- 注
1. 符号属性为“表 7-14 符号属性的值”中所示的值。
 2. 有关符号值，请参见“图 7-5”。

表7-14 符号属性的值

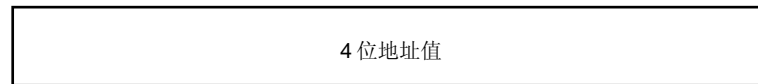
值	符号属性
00	由 EQU 指令定义的常量
01	代码段内的标号
02	数据段内的标号
03	ABBIT 属性位符号
04	AWBIT 属性位符号
05	saddr.bit/bsaddr.bit3位符号
06	wsaddr.bit4位符号
07	sfr.bit/bsfr.bit3位符号
08	wsfr.bit4位符号
09	RBBIT属性位符号
10	RWBIT属性位符号
11	字节注册位符号
FF	模块名

图 7-5 符号值的格式

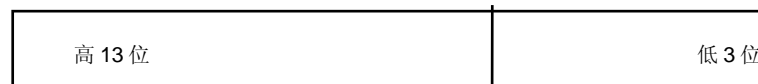
- 当符号属性为 NUMBER 时



- 当符号属性为 LABEL



- 当符号属性为位符号时



高13位：关于0FE00H的相对地址

低3位：比特位置(0-7)

7.3 目标转换器的启动

7.3.1 目标转换器的启动

可用下列两种方法来启动目标转换器。

(1) 从命令行启动

X>	[路径名]	oc78k0	[Δ选项]	...Δ	载入模块文件名	[Δ选项]...[Δ]
(1)	(2)	(3)	(4)	(5)	(4)	

- (1) 当前驱动器名
- (2) 当前目录名
- (3) 目标转换器命令文件名
- (4) 包含目标转换器操作的具体指令。
用一对双引号(“”)引用带有空格的路径。
- (5) 要转换的载入模块的文件名
用一对双引号(“”)引用路径中带有空格的文件名。

示例 C>oc78k0 k0.lmf -osample.hex

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

(2) 从参数文件启动

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

要想从参数文件启动目标转换器，应在命令行中指定参数文件说明选项（-F）。

从参数文件启动目标转换器，如下。

X>oc78k0	[Δ载入模块文件]	Δ-f	参数文件名
	(a)	(b)	

- (a) 参数文件说明选项
- (b) 包含启动目标转换器所需数据的文件

注 用编辑器来创建参数文件。

书写参数文件内容的规则如下。

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

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

示例 用编辑器创建参数文件（K0.POC）。

K0.POC 的内容

```
;parameter file
K0.lmf -osample.hex
-ssample.sym -r
```

用参数文件 K0.POC 启动目标转换器。

```
C>oc78k0 -fK0.poc
```

7.3.2 执行开始结束信息

(1) 执行开始信息

在目标转换器启动时，显示屏上出现执行开始信息。

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

(2) 执行结束信息

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

```
Target chip:uPD780xxx
Device file:Vx.xx

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

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

```
Target chip:uPD780xxx
Device file:Vx.xx

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

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

示例 1 指定了不存在的载入模块文件名。

```
C>oc78k0 sample.lmf

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

RA78K0 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。结果产生错误，目标转换器使对象转换中止。

示例 2 指定了不存在的目标转换器选项。

```
C>oc78k0 K0.lmf -a

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

RA78K0 error F4018: Option is not recognized '-a'
Please enter 'OC78k0 --', if you want help messages.
Program aborted.
```

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

7.4 目标转换器选项

7.4.1 目标转换器选项的类型

目标转换器选项是关于目标转换器操作的细节指令。目标转换器选项分成 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7-15 目标转换器选项

分类	选项	说明
十六进制格式目标文件输出说明	-O	指定十六进制格式目标文件的输出。
	-NO	
符号列表文件输出说明	-S	指定符号列表文件的输出。
	-NS	
以对象地址次序进行排序说明	-R	以地址次序对 HEX 格式对象进行排序。
	-NR	
对象补充说明	-U	将指定补充值作为目标码输出到无 HEX 格式对象输出的地址区。
	-NU	
错误列表文件输出说明	-E	输出错误列表文件。
	-NE	
参数文件说明	-F	从指定文件输入输入文件名和选项。
十六进制格式说明	-KI	英特尔标准的十六进制格式
	-KIE	英特尔扩展的十六进制格式
	-KT	扩展技术格式
	-KM	摩托罗拉 S 型格式 (24 位标准地址)
	-KME	摩托罗拉 S 型格式 (32 位地址)
设备文件搜索路径说明	-Y	从指定路径读取设备文件。
闪存 ROM 模型的文件单独输出说明	-ZF	<ul style="list-style-type: none"> — 在指定了一个闪存 ROM 模型的启动区域 ROM 程序的链接时添加一个输出启动区域和其他区域到单独的十六进制格式文件的选项。 — 如果指定了-ZF 选项，在启动区域 ROM 程序侧的输出文件类型是“HXB”，在其他程序侧的输出文件类型是“HXF”。
帮助说明	--	在显示屏上显示帮助信息。

7.4.2 目标转换器选项说明

本节包含对各目标转换器选项的具体说明。

(1) 十六进制格式目标文件输出指定**十六进制格式目标文件输出指定 (-O/-NO)****[语法]**

-O [输出文件名] -NO

默认预设:

- O 输入文件名.HEX
(扩展空间的文件类型是.H1 到.H15。)

[功能]

- 选项 -O 指定十六进制格式目标文件的输出。选项 -O 还指定输出目标和输出文件名。
- 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。
- 在仅为了输出符号列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，就会导致中止错误。
- 如果在指定选项 -O 时省略'输出文件名'，则将输出十六进制格式目标文件'输入文件名.HEX'到当前目录。
- 只要'输出文件名'中指定了路径名，'输入文件名.HEX'就会输出到指定路径。
- 如果同时指定选项-O 和-NO，则最后指定的选项优先。
- 如果指定了-ZF 选项，文件类型如下：

表7-16在指定了-ZF选项时的文件类型

文件	文件类型
启动区 ROM 程序侧的输出文件	.hxb
启动区 ROM 以外的程序侧的输出文件	.hxf

当输出代码到位于扩展空间中的段时，对象转换器为每个空间生成一个单独的十六进制格式对象模型文件。

为扩展空间生成的十六进制格式对象模型文件类型如下：

表7-17 扩展空间十六进制格式对象模型文件的类型

文件	常规空间	扩展空间							
	普通	EX1	EX2	EX3	EX4	...	EX13	EX14	EX15
十六进制	.hex	.H1	.H2	.H3	.H4H13	.H14	.H15

[使用示例]

输出十六进制格式目标文件（SAMPLE.HEX）。

```
C>oc78k0 K0.lmf -osample.hex
```

(2) 符号列表文件输出指定**符号列表文件输出指定 (-S/-NS)****[语法]**

```
-S [输出文件名]
-NS
```

默认预设:

```
-S 输入文件名.SYM
(扩展空间的文件类型是.S1到.S15。)
```

[功能]

- 选项 **-S** 指定符号列表文件的输出。选项 **-S** 还指定输出目标和输出文件名。
- 选项 **-NS** 指定不输出符号列表文件。

[应用]

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

[说明]

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

当为扩展空间中的一个段定义了一个有**ADDRESS**或者**BIT**属性的符号时，目标转换器转换器为每个空间生成一个单独的符号表。

所有具有**NUMBER**属性的符号被输出至为普通空间生成的符号表格文件。

为扩展空间生成的符号表格文件的文件类型如下：

表 7-18 扩展空间符号表格文件的类型

文件	常规空间	扩展空间							
	普通	EX1	EX2	EX3	EX4	...	EX13	EX14	EX15
十六进制	.hex	.S1	.S2	.S3	.S4S13	.S14	.S15

[使用示例]

输出符号列表文件（SAMPLE.SYM）。

```
C>oc78k0 K0.lmf -ssample.sym
```

(3) 以对象地址次序进行排序指定

以对象地址次序进行排序指定 (-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】

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

【使用示例】

以地址次序对 HEX 格式对象进行排序。

C>oc78k0 K0.lmf -r

(4) 对象补充指定**对象补充指定 (-U/-NU)****[语法]**

-U 补充值 [, [起始], 长度] -NU

默认预设:

U0FFH (填写 0FFH)

[功能]

- 选项 -U 将指定补充值作为目标码输出到 HEX 格式对象尚未输出的地址区。
- 选项 -NU 使选项 -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，其行为可能是不可预料的。通过指定选项 -U，可将代码写入 HEX 格式对象尚未输出的地址区。

[说明]

- 可指定为补充值的取值范围如下。

$$0H \leq \text{补充值} \leq 0FFH$$

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

- “起始”指定进行补充的起始地址区。
可指定为起始区的取值范围如下。

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

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

- “长度”指定进行补充的地址区的长度。可指定为长度的取值范围如下。

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

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

- 如果起始地址和长度都未指定，目标转换器将假定已指定内部 ROM 范围，并完成处理过程。

- 如果同时指定选项 **-U** 和 **--NU**，则最后指定的一项优先。
- 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** 中起始点和长度的指定格式及其解释。
 - (a) **-U** 补充值
如果目标设备中包含内部 ROM，则为内部 ROM 区域
 - (b) **-U** 补充值, 长度
从地址 0 至长度为-1 的地址。
 - (c) **-U** 补充值, 起始, 长度
从起始地址至起始地址+长度为-1 的地址。

[使用示例]

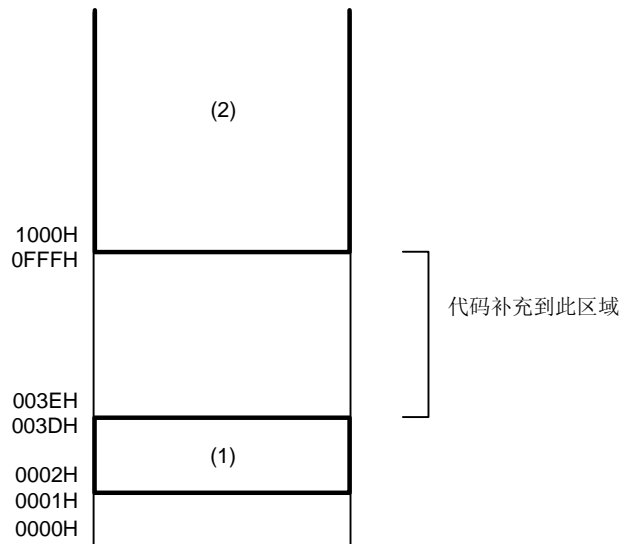
补充 HEX 格式对象代码尚未输出的地址区。

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

```

:020000000200FC
:100002002B41000BFC80FE2B40000944F7083A20EC      ;(1)
:100012001A6720FE2822006521FED350D25014FE1A      ;(1)
:10002200B900059F2835002431B900059F28350005      ;(1)
:0C003200242156AF0A8302A807A830560C
:01000003B5D0d0026A3...                            ;(2)
:1010100024A5F622B667...                            ;(2)

:000000001FF
    
```



00H 补充到地址区 003EH 至 0FFFH 中。

C>oc78k0 K0.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** 使选项 **-E** 不可用。

[应用]

指定选项 **-E** 来更改错误列表文件的输出目标和输出文件名。

[说明]

- 错误列表文件的文件名可指定为磁盘型文件名或设备型文件名。但是，若指定设备型文件名 **CLOCK** 则会产生中止错误。
- 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，错误列表文件将输出到当前驱动器。
- 如果同时指定选项 **-E** 和 **-NE**，则最后指定的选项优先。

[使用示例]

创建错误列表文件 (K0.EOC)。

```
C>oc78k0 K0.lmf -eK0.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:

选项和输入文件名只能从命令行指定。

[功能]

选项 **-F** 指定从指定文件输入的选项和输入文件名。

[应用]

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

[说明]

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

[使用示例]

用参数文件进行对象转换。

设置参数文件 K0.POC 的内容，如下。

```
;parameter file  
K0.lmf -osample.hex  
-ssample.sym -r
```

在命令行中键入下列内容。

```
C>oc78k0 K0.lmf -f78K0.poc
```

(7) 十六进制格式指定

十六进制格式指定 (-KI/-KIE/-KT/-KM/-KME)

[语法]

-KI
-KIE
-KT
-KM
-KME

默认预设:

- KIE

[功能]

- 指定要输出的十六进制对象模块文件的格式。

[应用]

- 使用这些选项指定要从[英特尔标准十六进制格式]、[英特尔扩展十六进制格式]、[扩展技术格式]、[摩托罗拉 S 型格式（标准地址）]和[摩托罗拉 S 型格式（32 位地址）]输出的十六进制对象模块文件的格式。

[说明]

- 每个选项指定的地址范围如下：
 - KI: 英特尔标准十六进制格式
0H 到 FFFFH（最高为 64KB）
 - KIE: 英特尔扩展十六进制格式
0H 到 FFFFFH（最高为 1MB）
 - KT: 扩展技术格式
0H 到 FFFFFFFFH（最高为 4GB）
 - KM: 摩托罗拉 S 型格式（标准地址）
0H 到 FFFFFFFFH（最高为 16MB）
 - KME: 摩托罗拉 S 型格式（32 位地址）
0H 到 FFFFFFFFH（最高为 4GB）

[使用示例]

- 指定摩托罗拉 S 型格式（标准地址）对象。

C>oc78k0 k0.lmf -km

(8) 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:

设备文件将从下列次序确定的路径读取。

- 1) <..\dev> (oc78k0.exe 起始路径)
- 2) OC78k0 的启动路径
- 3) 当前目录
- 4) 环境变量 PATH

[功能]

从指定路径读取设备文件。

[应用]

指定存在设备文件的路径。

[说明]

- 如果在选项 -Y 之后指定了非路径名, 则产生中止错误。
- 如果在选项 -Y 后省略路径名, 产生中止错误。
- 设备文件的读取路径按照如下次序确定。
 - (i). 选项 -Y 指定的路径
 - (ii). <..\dev> (oc78k0.exe 起始路径)
 - (iii). OC78k0 的启动路径
 - (iv). 当前目录
 - (v). 环境变量 PATH

[使用示例]

将设备文件路径指定为目录 c:\78k0\dev。

```
C>oc78k0 K0.lmf -yc:\78k0\dev
```

(9) 闪存 ROM 模型的文件单独输出指定**闪存 ROM 模型的文件单独输出指定 (-ZF)****[语法]****-ZF****默认预设:**

不单独输出。

[功能]

- 指定闪存区域的起始地址。

[说明]

- 当指定链接闪存 ROM 模型的启动区域 ROM 程序时，添加一个输出启动区域和其他区域到单独的十六进制格式文件的选项。
- 如果指定了 -ZF 选项，在启动区域 ROM 程序侧的输出文件类型是“HXB”，在其他程序侧的输出文件类型是“HXF”。

注意：对于没有闪存自编程功能的设备，请不要指定此选项。

[使用示例]

```
C>oc78k0 k0.lmf -zf
```

(10) 帮助指定**帮助指定 (--)****[语法]**

--

默认预设:

无显示

[功能]

选项 -- 在显示屏上显示帮助信息。

[应用]

帮助信息是目标转换器选项说明的列表。在执行目标转换器时参考这些内容。

[说明]

在指定选项 --时，所有其它目标转换器选项均不可用。

注意：不能从 PM+指定此选项。

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

[使用示例]

在指定选项 --时，显示屏上输出帮助信息。

```

C>oc78k0 --

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


usage : oc78k0 [option [...]] input-file [option [...]]
The option is as follows ([ ] means omissible).
-f file           :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.
-r/-nr           :Sort HEX object by address / Not.
-u value [,start], size]/nu :Fill up HEX object with specified value/Not.
-k kind          :Select hex format.1; intel format IE; intel extend format T; tex format M
                 ; s format ME; s-32bit format
-y directory     :Set device file search path.
-zf              :Create boot hex module file (HXB), and flash hex module file(HXF)
--              :Show this message.
DEFAULT ASSIGNMENT: -o -s -r -u0ffh

```

7.5 在 PM+ 中设置选项

本节介绍 PM + 目标转换器选项设置的方法。

7.5.1 选项设置方法

从 PM + 的 [Tools] 菜单选择 [Object Converter Options...] 或点击 ，显示 <Object Converter Options> 对话框。

可通过在对话框中输入所需的选项来设置目标转换器选项。

图 7-6 <Object Converter Options> 对话框 (当选择 <<Output1>> 标签时)

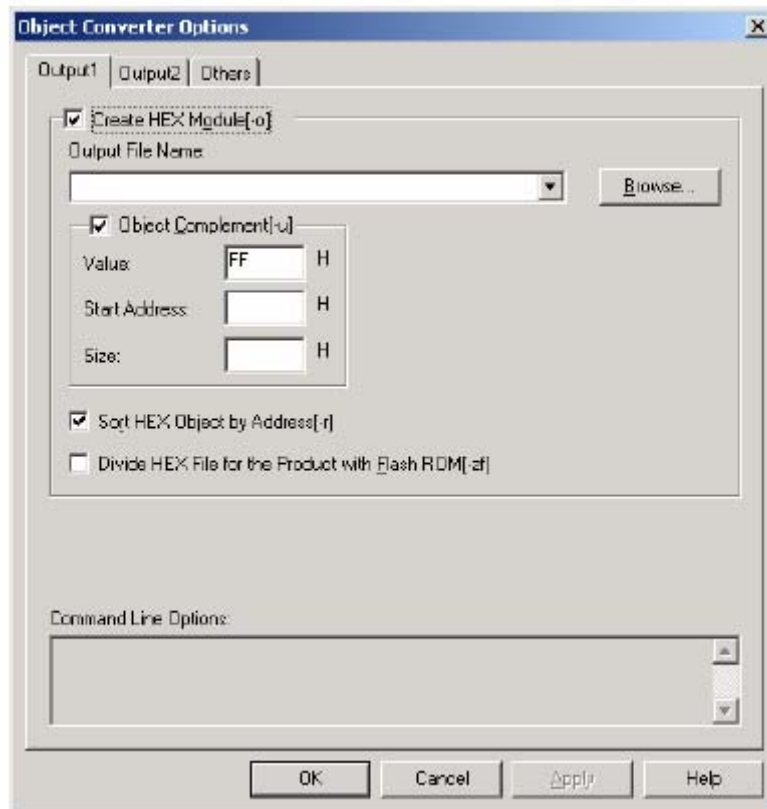
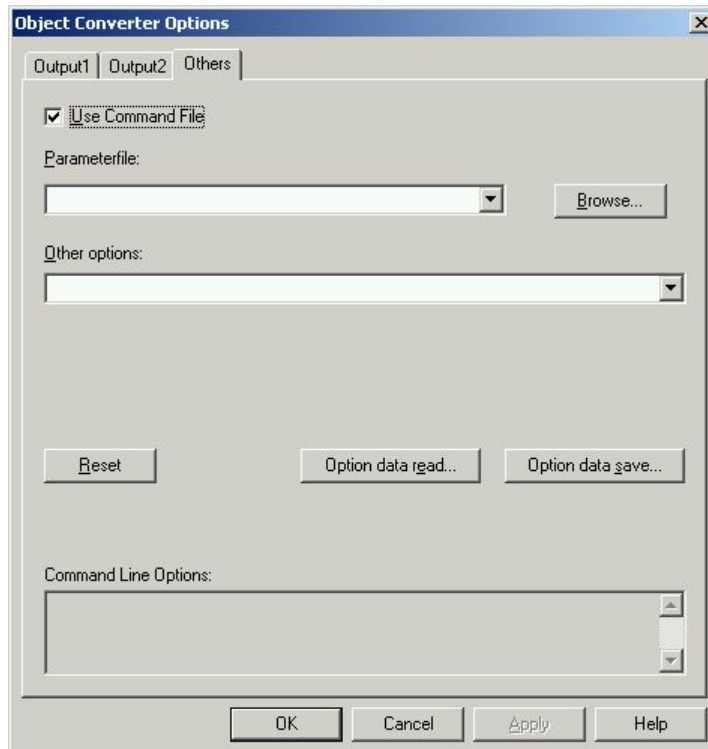


图 7-7 <Object Converter Options> 对话框（当选择 <<Output2>> 标签时）



图 7-8 <Object Converter Options> 对话框（当选择 <Others> 标签时）



7.5.2 选项

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

<<Output1>> 标签

- **Create HEX module [-o]**
创建此选项以输出一个十六进制格式的对象模块文件。
输出文件名。
使用[Browse]按钮指定对象模块文件的路径和文件名或者直接输入路径或者文件名。
- **Object Complement [-u]**
预先写入代码，防止非法代码写入无 HEX 格式对象输出的地址并防止发生程序循环。
- **Sort HEX Object by Address [-r]**
如果需要用地地址排序 HEX 格式对象，应指定此选项。
- **Divide HEX Files for the Product with Flash ROM [-zf]**
选定此选项输出包含闪存的产品的启动区域和其他区域到单独的十六进制格式文件。

注意：对于没有闪存自编程功能的设备，请不要指定此选项。
- **Command Line Options**
此编辑框是只读的。显示当前设置的选项字符串。

<<Output2>> 标签

- **Create Symbol Table File [-s]**
选择此选项输出一个符号表格文件。
输出文件名：
使用[Browse]按钮指定符号表格文件的路径和文件名或者直接输入路径或者文件名。
- **Create Error List File [-e]**
选择此选项输出一个错误列表文件。
输出文件名：
使用[Browse]按钮指定错误列表文件的路径和文件名或者直接输入路径或者文件名。
- **Command Line Options**
此编辑框是只读的。显示当前设置的选项字符串。

<<Others>> 标签

- Use Command File

选择此复选框来创建命令文件。

- Parameter file

通过[Browse...]按钮或直接输入，读取用户定义的参数文件。

- Other options

如果在复选框或单选按钮可选择的选项之外还希望指定其它选项，可在输入框中输入该选项。

注意：不能在 PM+ 上指定帮助指定(--)选项。

- Reset

使输入内容复位。

- Option data read...

打开 <选项数据读取>对话框，在指定选项数据文件后读取该文件。

- Option data save...

打开<选项数据保存>对话框，将选项数据保存到选项数据文件中并命名。

- Command Line Options

此编辑框只读。显示当前设置选项字符串。

第 8 章 库管理程序

库管理程序用于编辑 RA78K0 目标模块文件和以 1 个模块为单位的库文件。库管理程序同时也输出列表文件。如果有库管理程序错误发生，则就会有错误信息输出显示错误产生的原因。

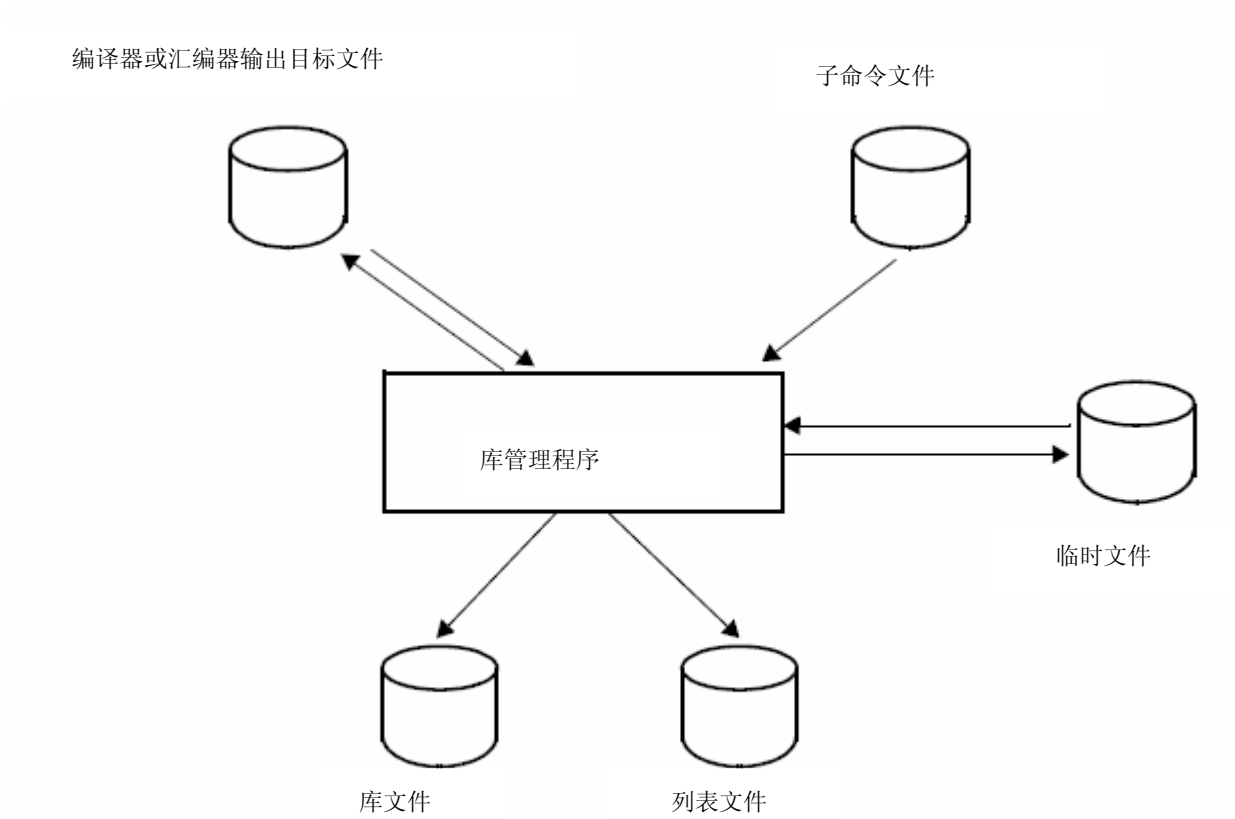
8.1 库管理程序的 I/O 文件

库管理程序的 I/O 文件显示如下。

表 8-1. 库管理程序的 I/O 文件

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

图 8-1 库管理程序的 I/O 文件



8.2 库管理程序的功能

(1) 生成模块库

汇编器及链接器为其输出的每一个模块都创建一个文件。

这就意味着如果所创建模块的数量巨大，那么与模块所对应的文件数也会大量增加。因此，RA78K0 中有一个文件用来将这些目标模块收集到一个单一的文件中，这种功能就是所谓的模块目标成生。而作为库组织的文件称为库文件。

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

(2) 编辑库文件

该库管理文件包含有如下库文件编辑功能。

- (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) 从库文件中取出模块

(需要了解这些功能的详细说明，请参阅 8.5 子命令。)

(3) 库文件数据的输出

该库管理文件包含用于编辑和输出如下的存储于库文件中的数据项的函数。

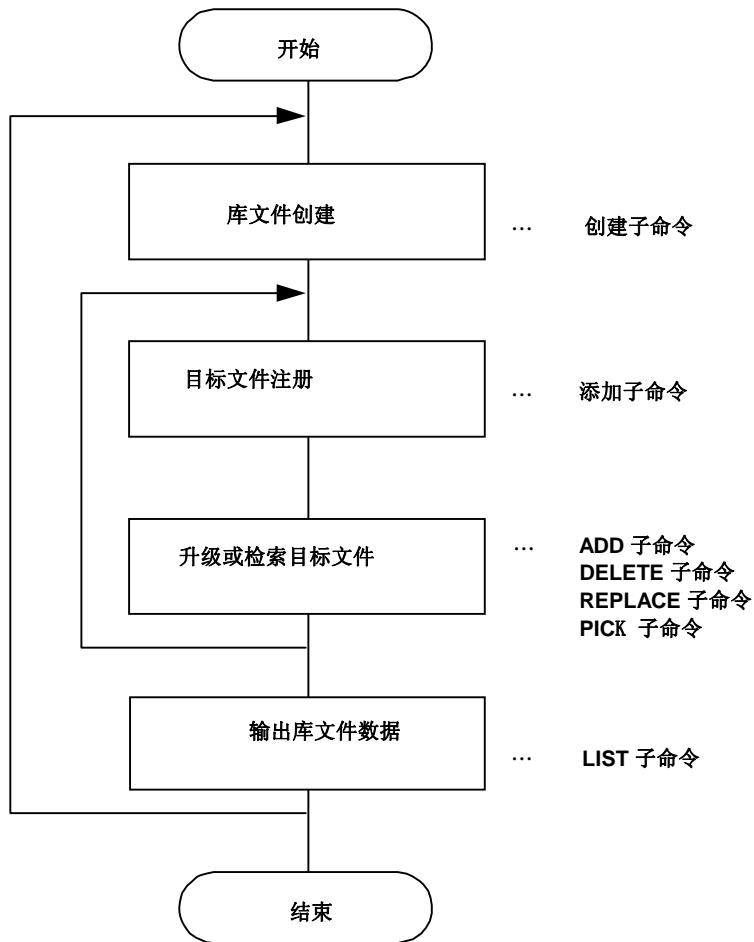
- a) 模块名
- b) 所创建的文件
- c) 注册日期
- d) 更新日期
- e) PUBLIC 型符号数据

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

(4) 库文件的创建过程步骤

创建库文件的一般过程如下。

图 8-2 库文件创建过程步骤



8.3 库管理程序的启动

8.3.1 库管理程序的启动

有如下两种方法可以用于库管理程序的启动。

(1) 从命令行启动

```
X>[路径名] lb78k0 [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) 此项包含库管理程序功能的详细说明。
- 用双引号 (“ ”) 引用包含空格的路径。

示例 C>lb78k0 -ll20 -lw80

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

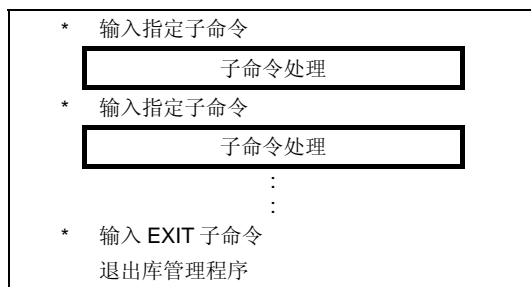
当库管理程序启动时，会显示如下启动信息。

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

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

```
*create k0.lib
*add k0.lib k0main.rel k0sub.rel
*exit
```

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



一行最多可以输入 128 个字符。

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

(2) 由子命令文件启动

所谓的子命令文件就是在其中存储库管理程序子命令的文件。

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

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

子命令文件使用方法是, 使用时, 在文件名前用'<'符号加以描述。

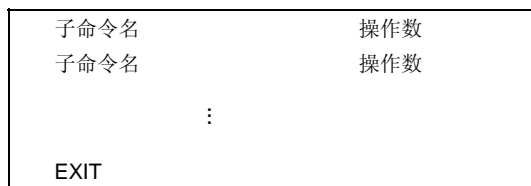
由子命令文件启动库管理程序方法如下。



(i) 在输入指定的子命令文件时, 确保一定要在子命令文件名前加入此符号。

(ii) 子命令存储在该文件中。

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



(c) 如果重复执行一个子命令, 请在每一行的末尾加'&'符号以示继续执行。

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

- (e) 如果子命令文件中的最后一个子命令不是 EXIT 子命令，库管理程序会自动将其解释为 EXIT 子命令。
- (f) 库管理程序从子命令文件中读取并执行子命令，并在处理完子命令文件中所有的子命令后退出。

示例 用编辑器创建子命令文件(K0.SLB)。

K0.SLB 文件内容

```
; library creation command
create k0.lib
add k0.lib k0main.rel &
k0sub.rel
;
exit
```

使用子命令文件 K0.SLB 启动库管理程序。

```
C>lb78k0 <k0.slb
```

8.3.2 库管理程序执行开始和执行结束信息

(1) 执行开始信息

当库管理程序启动时，在显示器上就会有一条执行启动信息出现。

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

(2) 执行结束信息

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

```
*create k0.lib
*add k0.lib k0main.rel k0sub.rel
*exit
```

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

示例 指定了并不存在的库管理程序选项

```
C>lb78k0 -a

78K/0 Series Librarian Vx.xx [xx xxx xx]
Copyright (C) NEC Electronics Corporation xxxx, xxxx
RA78K0 error F5018: Option is not recognized '-z'
Usage: LB78k0 [options]
```

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

8.4 库管理程序选项

8.4.1 库管理程序选项类型

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

表 8-2. 库管理程序选项

分类	选项	说明
列表文件格式指定	-LW	更改列表文件里一行可打印的字符数
	-LL	更改列表文件里一页可打印的行数
	-LF	在列表文件末端插入一换页(FF)码
	-NLF	
临时文件创建路径指定	-T	在指定路径创建一临时文件.
设备文件搜索路径指定	-Y	由指定路径读取设备文件
帮助信息指定	--	在显示器上显示一帮助信息

8.4.2 库选项的说明

如下所述是关于库选项的详细说明。

(1) 列表文件格式指定**列表文件格式指定 (-LW, -LL, -LF/-NLF)****(a) -LW****[语法]**

-LW [字符数目]

默认预设

-LW132 (显示器输出情况下为 80 个字符)

[功能]

选项 -LW 指定在列表文件内一行可输入的字符个数。

[应用]

指定-LW 选项用于更改列表文件内一行可输入的字符个数。

[说明]

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

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

如果所指定数字值超出此范围，或指定了一个非数字值，则会产生异常中止错误。

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

[使用示例]

指定列表文件中每行输入字符为 80 个。

```
C>lb78k0 -lw80
```

(b) -LL**[语法]**

-LL [行数目]

默认预设

-LL66 (在显示器输出情况下, 没有页分隔符)

[功能]

选项 -LL 指定在列表文件内一页可输入的行数。

[应用]

指定 -LL 选项用于更改列表文件内一页可输入的行数。

[说明]

- 可用 -LL 选项指定的行数目范围显示如下。

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

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

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

[使用示例]

指定列表文件中每页行数为 20 个。

```
C>lb78k0 -ll20
```

(c) -LF/-NLF**[语法]**

-LF -NLF

默认预设

-NLF

[功能]

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

[应用]

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

[说明]

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

[使用示例]

向列表文件添加一个换页码。

```
C>lb78k0 -lf
```

(2) 为临时文件创建指定路径**为临时文件创建指定路径 (-T)****[语法]**

-T 路径名

默认预设

创建至由环境变量 **TMP** 所指定的路径下。

如果没有指定路径，则临时文件创建至当前路径下。

[功能]

选项 **-T** 指定将临时文件创建至指定路径下。

[应用]

使用选项 **-T** 指定临时文件创建的位置。

[说明]

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

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

[使用示例]

指定将临时文件输出至目录 **c:\TMP** 下。

```
C>lb78k0 -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

默认预设

设备文件将会按照如下顺序确定的路径读出。

- (i) <...\dev> (用于 lb78k0.exe 的启动路径)
- (ii) LB78k0 的启动路径
- (iii) 当前目录
- (iv) 环境变量 PATH

[功能]

由指定路径读取设备文件。

[应用]

指定设备文件所在的路径。

[说明]

- 如果在-Y 选项后指定的不是路径名内容，便会产生异常中止错误。
- 如果在-Y 选项后省略了路径名，也会产生异常中止错误。
- 按照下列顺序来确定读取设备文件的路径。
 - (i). 由选项-Y 指定路径
 - (ii). <...\dev> (用于 lb78k0.exe 的启动路径)
 - (iii). LB78k0 的启动路径
 - (iv). 当前目录
 - (v). 环境变量 PATH

[使用示例]

指定设备文件路径为目录 c:\78k0\dev.

```
C>lb78k0 -yc:\78k0\dev
```

(4) 帮助信息指定

帮助信息指定 (--)

[语法]

--

默认预设

不显示

[功能]

选项 -- 在显示器上显示帮助信息。

[应用]

显示的帮助信息为一子命令说明列表。在执行库管理程序时请参阅这些说明。

[说明]

当指定了-- 选项时，所有其它库管理程序选项均不可用。

注意：PM+不能指定此选项

要参考 PM+的帮助，点击< Library File Name >对话框中的[帮助]按钮。

[使用示例]

当指定了--选项时，帮助信息输出至显示器。

```
C>lb78k0 --
78K/0 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
+-----+
```

8.5 子命令

8.5.1 子命令类型

子命令为库管理程序的操作提供详细的指导说明。子命令分为八种类型。

表8-3 子命令

子命令名	缩写形式	说明
CREATE	C	创建一新的库文件
ADD	A	向库文件添加一个模块
DELETE	D	从库文件中删除一个模块
REPLACE	R	用其它模块替换库文件中的模块
PICK	P	从库文件中取回一个模块
LIST	L	输出库文件中模块数据
HELP	H	在显示器上显示一帮助信息
EXIT	E	退出库管理程序

8.5.2 子命令说明

如下是有关各子命令的详细功能和操作说明。

命令文件的一般格式为

*子命令 [Δ选项] Δ库文件名 [Δ选项] 处理 [Δ选项]		
	(a)	(b)

- (1) 先前刚刚指定的库文件名可以被'!'符号替换掉。
- (2) 处理 = Δ目标模块文件名 Δ库文件名 [Δ (Δ模块名 [Δ,...])]

(1) CREATE**CREATE****[语法]**

CREATA库文件名 [Δ处理]

缩略格式:

C

[功能]

CREATE 子命令功能是创建一个新的库文件。

[说明]

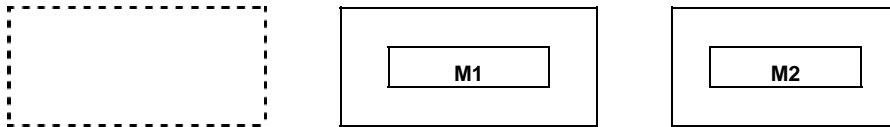
- 创建的库文件大小变为 0。
- 当指定某一处理时，在库文件创建的同时也就完成了模块注册。
- 库文件名: 如果有同名文件存在，就覆盖该同名文件。
- 处理: 如果目标模块文件带有与库文件内相同的公共符号，则该目标模块文件不能注册。
与库文件中模块同名的模块不能注册。
- 如果有错误产生，便中断处理过程，同时也就不能创建库文件了。

[使用示例]

当创建库文件时，同时注册模块 M1 和模块 M2。

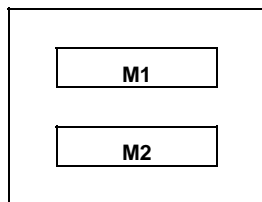
```
*CREAT k0.lib m1.rel m2.rel
```

<文件创建前>



<文件创建后>

K0.LIB



(2) ADD**ADD****[语法]**

ADD△库文件名 △处理

缩略格式:

A

[功能]

ADD 子命令的功能是向库文件中添加一个模块。

[说明]

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

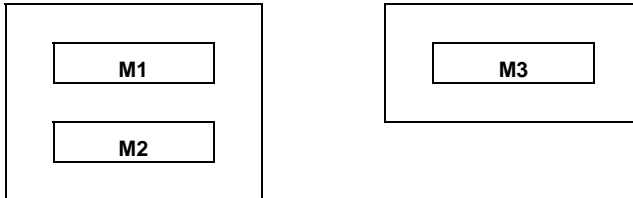
[使用示例]

向库文件(K0.LIB)中添加一个模块(M3)。

```
*add k0.lib m3.rel
```

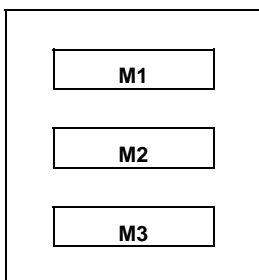
<添加前>

K0.LIB



<添加后>

K0.LIB



(3) DELETE**DELETE****[语法]**

DELETE Δ库文件名 Δ (Δ模块名 [Δ,...] Δ)

缩略格式:

D

[功能]

DELETE 子命令的功能是从库文件中删除一个模块。

[说明]

- 如果指定删除的模块在库文件中并不存在，则会产生错误。
- 如果有错误产生，就会中断处理过程而且不会改变库文件状况。

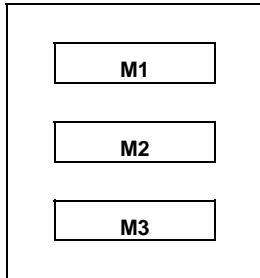
[使用示例]

从库文件(K0.LIB)中删除模块(M1, M3)。

```
*delete k0.lib (m1.rel m3.rel)
```

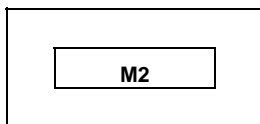
<删除之前>

K0.LIB



<删除之后>

K0.LIB



(4) REPLACE

REPLACE

[语法]

REPLACEΔ库文件名Δ处理

缩略格式:

R

[功能]

REPLACE 子命令的功能是用其它目标模块文件替换库文件中的模块。

[说明]

- 如果库文件中没有与替换模块名相同的模块存在, 则会产生错误。
- 如果替换模块中含有与库文件内相同的公共符号, 则会产生错误。
- 替换对象模块的文件名必需与注册变中所使用的文件名相同。
- 如果有错误产生, 就会中断处理过程而且不会改变库文件状况。

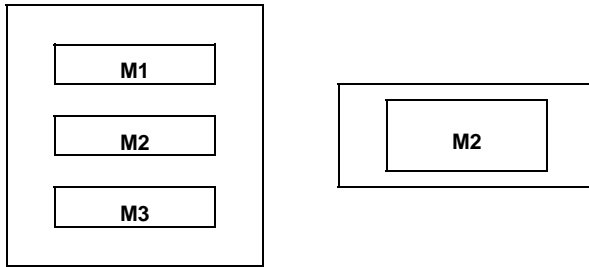
[使用示例]

替换库文件(K0.LIB)中的一个模块(M2)。

*replace k0.lib m2.rel

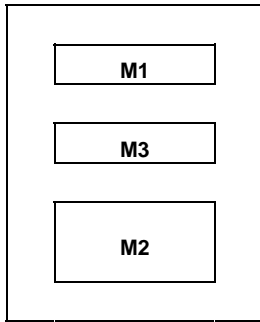
<替换之前>

K0.LIB



<替换之后>

K0.LIB



由于在库文件中模块(M2)删除之后便注册了一新的模块(M2)，所以 M2 在库文件中次序为最后一个模块。

(5) PICK**PICK****[语法]**

PICK Δ库文件名 Δ (Δ模块名 [Δ, …] Δ)

缩略格式:

P

[功能]

PICK 子命令的功能是从现存的库文件中拾取指定的模块。

[说明]

- 被拾取的模块就成为目标模块文件且该目标模块文件名为其在库文件中注册的文件名。
- 如果指定模块名在库文件中并不存在，则会产生错误。
- 错误产生时，处理过程会中断。但是，如果指定两个或多个模块时产生错误，则在导致错误出现的模块之前所拾取的那些模块是可用的而且被保存至磁盘上。

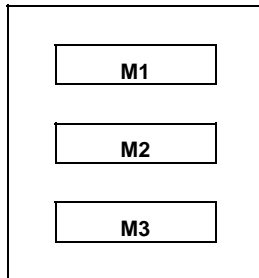
[使用示例]

从库文件(K0.LIB)中拾取一个模块(M2)。

```
*pick k0.lib (m2.rel)
```

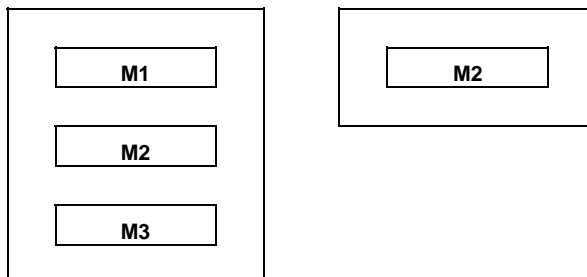
<拾取之前>

K0.LIB



<拾取之后>

K0.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

缩略格式:

L

[功能]

LIST 子命令的功能是输出库文件中的模块数据。

[说明]

- 可以指定多个选项。
- -O:
可以将设备类型文件名指定作为输出文件名。
如果省略输出文件名, 则会产生错误。
如果省略文件类型, 那么库管理程序便假定'输入文件名.LST' 作为输入。
- -PUBLIC/-NOPUBLIC:
该选项可以通过仅指定下划线字来进行选定。
-PUBLIC 选项指定公共符号数据的输出。
-NOPUBLIC 选项使得 -PUBLIC 选项不可用。
如果同时指定 -PUBLIC 选项 和 -NOPUBLIC 选项, 则最后指定的选项优先。

【使用示例】

将库文件(K0.LIB)中模块数据输出至列表文件 (K0.LST)中去。指定选项 -P 以便输出公共符号数据。

```
*list -p -ok0.lst k0.lib
```

引用列表文件 (K0.LST)。

```
78K/0 Series librarian Vx.xx DATE : xx xxx xx PAGE 1
LIB-FILE NAME : K0.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

缩略格式:

H

[功能]

HELP 命令的功能是在显示器上显示帮助信息。

[说明]

显示的帮助信息是一个子命令及其说明列表。在执行库管理程序时，通过指定 HELP 命令或--选项以参考该帮助信息。

[使用示例]

指定 HELP 命令以输出 HELP 信息。

*help

```

+-----+
| Subcommands : create, add, delete, replace, pick, list, help, exit
|
| Usage : subcommand[ option]masterLBF[ option]transaction[ option]
|
|           transaction ::=      OMFname
|                           LBFname [(modulename [...])]
|
| <create   >:      create masterLBF [ transaction]
| <add      >:      add masterLBF transaction
| <delete   >:      delete masterLBF (modulename [...])
| <replace  >:      replace masterLBF transaction
| <pick     >:      pick masterLBF (modulename [...])
| <list     >:      list [ option] masterLBF [(modulename [...])
|                   option :  -p = output public symbol
|                             -np = no output public symbol
|                             -o filename = specify output file name
|
| <help     >:      help
| <exit     >:      exit
+-----+

```

(8) EXIT

EXIT

[语法]

EXIT

缩略格式:

E

[功能]

EXIT 子命令的功能是退出库管理程序。

[说明]

使用该子命令来退出库管理程序。

[使用示例]

退出库管理程序。

*exit

8.6 在 PM + 中设置选项

本节描述由 PM + 指定库文件的方法。

8.6.1 选项设定方法

在 PM+[Tools]菜单中的[Tools]选项中选择 [Librarian Options] 或点击  以显示 <Library Options> 对话框。在指定了路径和文件名后，按[Next]以显示 <Subcommand> 对话框。

可以在该对话框中通过输入需要的选项来设置不同库管理程序选项。

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

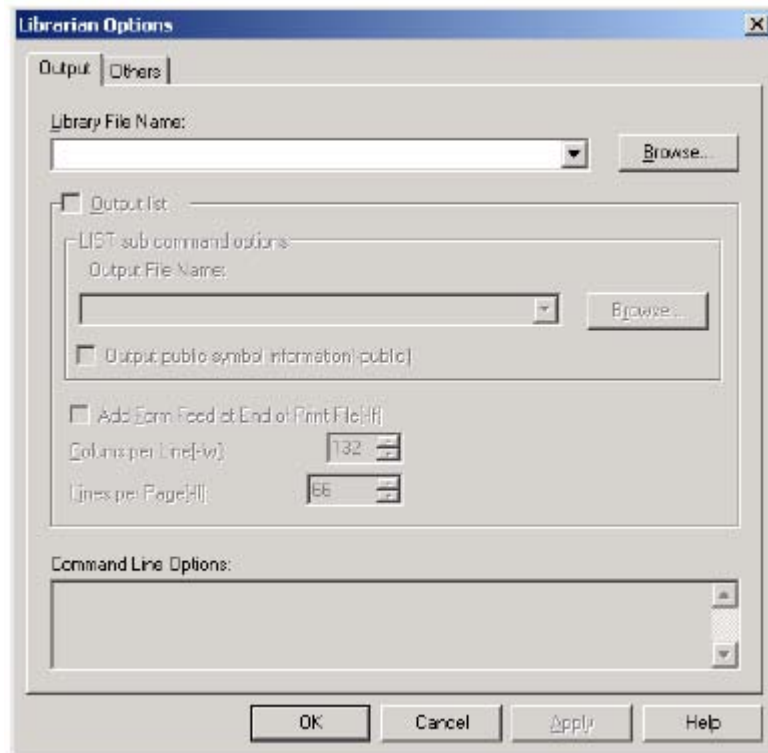
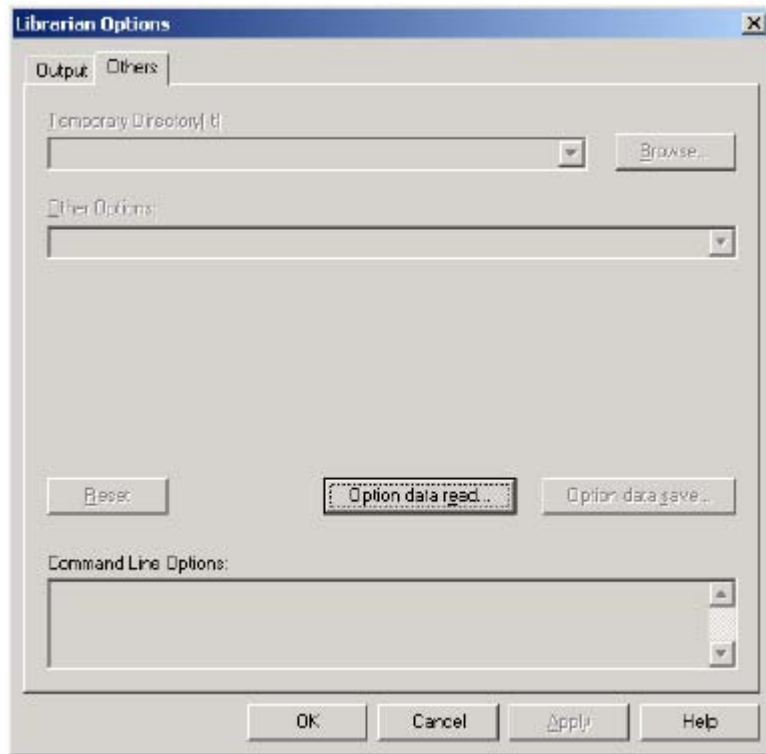


图 8-4 < Librarian Options >对话框（当选择<< Others >>标签时）



8.6.2 选项设定

< Librarian Options >对话框中的各个选项如下所述:

<< Output >>标签

—Library File Name

使用[**B**rowse]按钮指定库文件的文件名或直接输入文件名。

—LIST sub command options

选中此框输出一个列表文件。

—Output File Name

使用[**B**rowse]按钮指定列表文件的文件名和地址或直接输入文件名。

—Output public symbol information[-public]

选中此框添加公共符号信息到要输出的列表文件。

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

选中此框添加一个页源代码（FF）到库文件的结尾。

—Columns per Line[-lw]

指定库文件的一行中的字符数。可指定的字符在 72—260 个之间。

—Lines per Page[-ll]

指定库文件的一页中的行数。可指定的字符在 20—32767 个之间。

— 命令行选项

此编辑框是只读的。显示当前设置的选项字符串。

<< Others >>标签

—Temporary directory [-t]

使用[**B**rowse]按钮指定要创建的临时文件的路径或直接输入路径名。

—Other Options

要设置无法在对话框中指定的选项，则在输入框中输入。

注意：不能在 PM+ 上指定帮助指定(--)选项。

—Reset

重置输出内容。

—Option data read

打开< Read Option Data >对话框，在指定选项文件后读取此文件。

—Option data save

打开< Save Option Data >对话框，用指定的名称保存对话框。

— 命令行选项

此编辑框是只读的。显示当前设置的选项字符串。

8.7 从 PM+ 操作库文件的方法

本节描述从 PM+ 操作库文件的方法。< Edit Option > 对话框如下所述。

8.7.1 操作方法

从 PM+ 的 [Tools] 菜单选择 [Register Ex-tool]，寄存以单独模式启动 LB 的执行格式。

选择寄存的图标，然后点击 < Library File Name > 对话框。

在指定了路径和文件名后，点击 [Next] 以显示 < Subcommand > 对话框。

图 8-5. <<Library File Name> 对话框

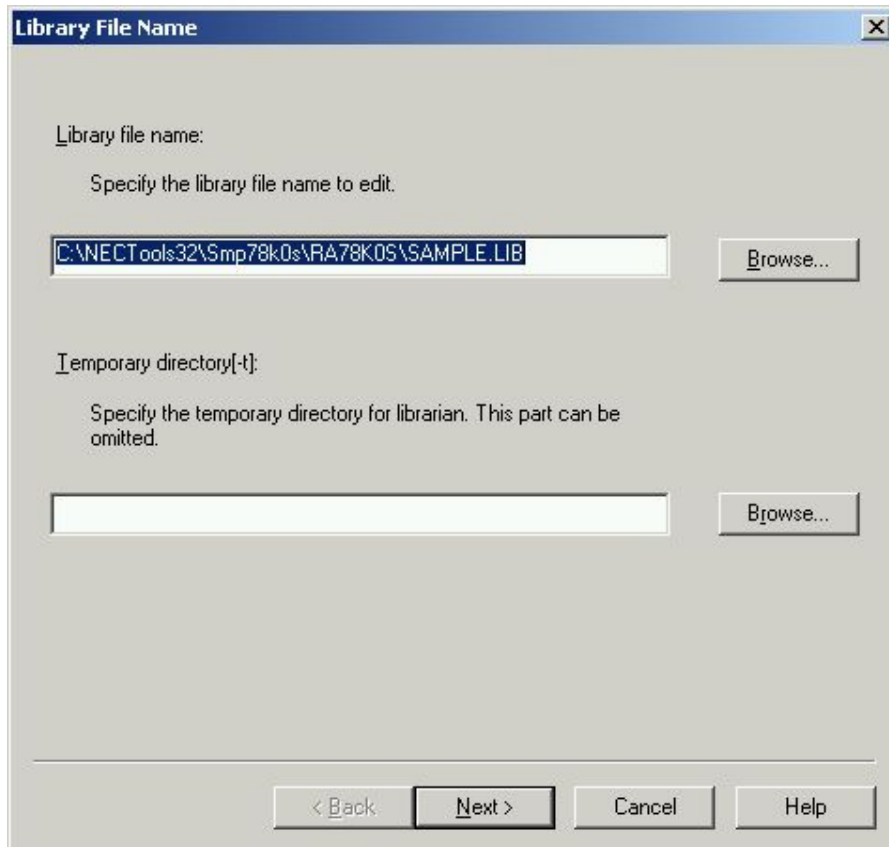
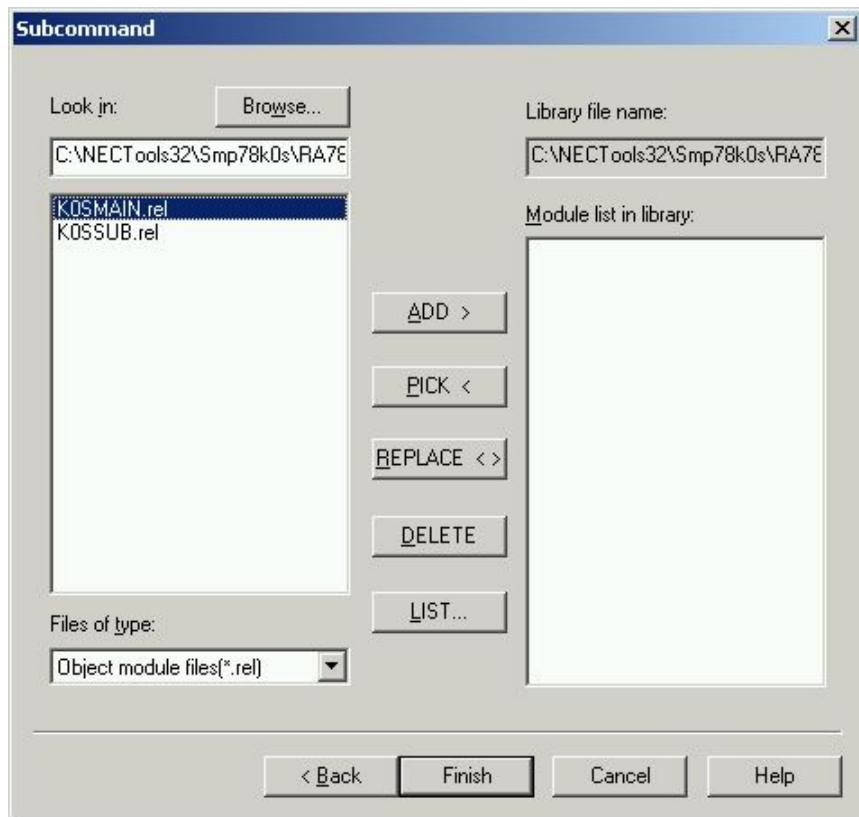


图 8-6. < Subcommand > 对话框



8.7.2 选项设置

<Subcommand>对话框里各种选项描述如下。

<Library File Name> 对话框

- **L**ibrarian file name
通过使用[Browse...]按钮指定需要编辑的库文件路径或直接输入。
- **T**emporary directory [-t]
通过使用[Browse...]按钮指定待创建的临时文件路径或直接输入。

< Subcommand > 对话框

- **L**ook in
通过使用[Browse...]按钮选定要创建至库的目标模块文件或直接输入。
如果指定了路径，则会显示一个文件列表。
- **F**iles of type
指定文件列表中显示的文件类型。
- **L**ibrary file name
此对话框是只读的，显示当前指定库的名称。

- **M**odule list in library
显示当前指定库中的对象模块文件列表。
- **A**DD >
向现存库文件中添加一个模块。
- **P**ICK <
从现存的库文件中拾取指定的模块。
- **R**EPLACE <>
用其它目标模块文件替换现存库文件模块。
- **D**ELETE
从现存库文件中删除一个模块。
- **L**IST
输出库文件中模块数据。

第9章 列表转换器

列表转换器输入汇编列表文件以及由汇编器输出的目标模块文件以及由连接器输出的装载文件。

列表转换器将实际地址嵌入可重定位地址中去，将符号嵌进输入文件中，并输出绝对汇编列表文件。这就可以免除在引用链接映像时需要查看汇编列表这种十分麻烦的工作任务。

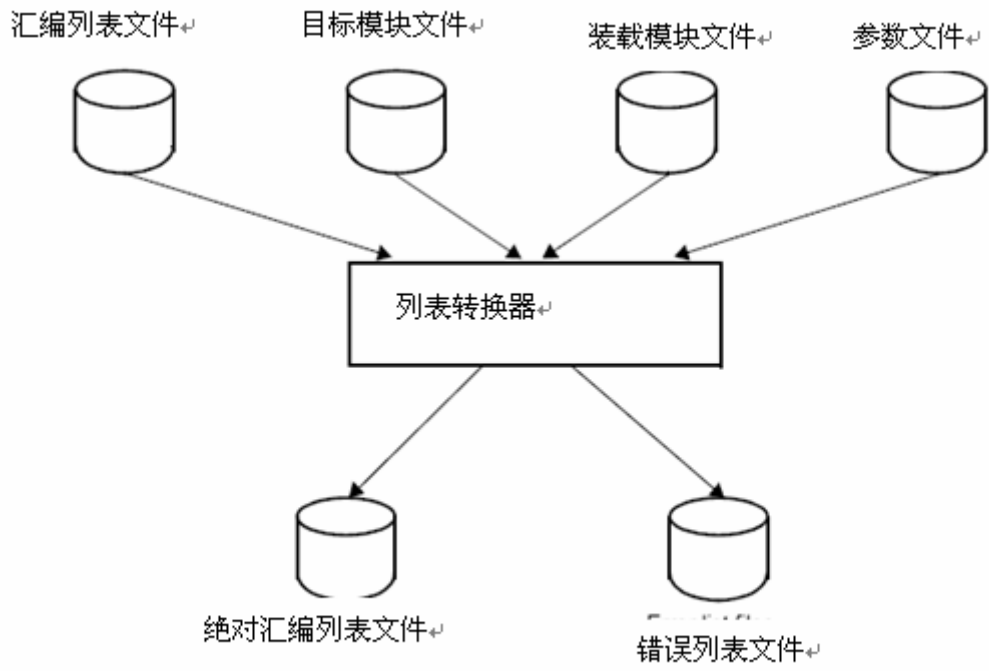
9.1 列表转换器 I/O 文件

列表转换器的 I/O 文件如下所示。

表 9-1 列表转换器 I/O 文件

类型	文件名	说明	缺省文件类型
输入文件	目标模块文件	<ul style="list-style-type: none">• 这些二进制文件重定位数据和一些关于机器语言数据的符号数据以及机器语言定位地址。	.REL
	汇编列表文件	<ul style="list-style-type: none">• 这些文件包含诸如汇编列表和交叉参考列表之类的汇编数据	.PRN
	载入模块文件	<ul style="list-style-type: none">• 这些二进制映像文件包含作为链接结果的对象代码	.LMF
	参数文件	<ul style="list-style-type: none">• 这些文件包含用于执行程序的参数• 这些文件是由用户创建的	.PLV
输出文件	绝对汇编列表文件	<ul style="list-style-type: none">• 该列表文件将实际地址嵌入可重定位地址中去，将符号嵌进输入文件中	.P
	错误列表文件	<ul style="list-style-type: none">• 这些文件包含了一些在列表转换期间生成的错误数据	.ELV

图 9-1. 列表转换器 I/O 文件



9.2 列表转换器功能

下面比较了可重定位汇编程序与绝对汇编程序的优缺点。

【优点】

- 1) 可重定位汇编程序可由多成员组共同开发。
- 2) 可重定位汇编程序可以分成模块以易于开发和存储。
- 3) 可重定位汇编程序支持库管理。
- 4) 可重定位汇编程序适合于开发大规模程序。

【缺点】

- 1) 可重定位汇编程序的汇编列表中的地址与实际物理地址并不一致。
- 2) 可重定位汇编程序的汇编列表中的外部符号值为 0。如果要查找外部符号实际值，还必须参考连接映像。
- 3) 汇编列表中的可重定位数值与实际数值是不同的。

上述这些关于可重定位汇编程序的缺点显著地降低了程序调试和存储的效率，这是由于程序调试和存储需要大量的文件。而列表转换器就为克服可重定位汇编程序的这些缺点提供了一种解决的方法。

- 1) 由列表转换器输出的绝对汇编列表中的地址与实际程序操作所用的地址完全一致。
- 2) 外部符号的实际值嵌入在该列表中。
- 3) 可重定位数值作为实际值嵌入在该列表中。
- 4) 对于符号表或交叉参考表中的符号值，其实际值也嵌入在该列表中。

示例 1. 重定位嵌入

- 汇编列表

据	21 2	----		CSEG	
	22 22	0000		START:	
	23 23				
	24 24				;芯片初始化
	25 25				
	26 26	0000	12201A	MOV	HDTSA, #1AH
	27 27	0003	1620FE	MOVW	HL, #HDTSA ;在 HL 中设定 hex 格式的 2 代码数
	28 28				
	29 29	0006	R9A0000	CALL	!CONVAH ;ASCII<- HEX 转换
	30 30				;将 ASCII 码输出至 BC 寄存器中去
	31 31	0009	1421FE	MOVW	DE, #STASC ;设置存储 ASCII 码的表地址至 DE
	32 32	000C	63	MOV	A, B
	33 33	000D	95	MOV	[DE], A
	34 34	000E	84	INCW	DE
	35 35	000F	62	MOV	A, C
	36 36	0010	95	MOV	[DE], A
	37 37				
	38 38	0011	FAFE	BR	\$\$
	39 39				
	40 40				END

- 绝对汇编列表

据	21 21	----		CSEG	
	22 22	0080		START:	
	23 23				
	24 24				;芯片初始化
	25 25				
	26 26	0080	11201A	MOV	HDTSA, #1AH
	27 27	0083	1620FE	MOVW	HL, #HDTSA ;在 HL 中设定十六进制 2 位代码数
	28 28				
	29 29	0086	R9A9300	CALL	!CONVAH ;ASCII<- HEX 转换
	30 30				;将 ASCII 码输出至 BC 寄存器中去
	31 31	0089	1421FE	MOVW	DE, #STASC ;设置存储 ASCII 码的表地址至 DE
	32 32	008C	63	MOV	A, B
	33 33	008D	95	MOV	[DE], A
	34 34	008E	84	INCW	DE
	35 35	008F	62	MOV	A, C
	36 36	0090	95	MOV	[DE], A
	37 37				
	38 38	0091	FAFE	BR	\$\$
	39 39				
	40 40				END

示例 2. 对象代码的嵌入

- 汇编列表

	21	21	----				CSEG		
	22	22	0000		START:				
	23	23							
	24	24						;芯片初始化	
	25	25							
	26	26	0000	11201A	MOV	HDTSA, #1AH			
据	27	27	0003	1620FE	MOVW	HL, #HDTSA		;在 HL 中设定十六进制 2 位代码数	
	28	28							
	29	29	0006	R9A0000	CALL	!CONVAH		;ASCII<- HEX 转换	
	30	30						;将 ASCII 码输出至 BC 寄存器中去	
	31	31	0009	1421FE	MOVW	DE, #STASC		;设置存储 ASCII 码的表地址至 DE	
	32	32	000C	63	MOV	A, B			
	33	33	000D	95	MOV	[DE], A			
	34	34	000E	84	INCW	DE			
	35	35	000F	62	MOV	A, C			
	36	36	0012	95	MOV	[DE], A			
	37	37							
	38	38	0011	FAFE	BR	\$\$			
	39	39							
	40	40					END		

- 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	11201A	MOV	HDTSA, #1AH			
据	27	27	0083	1620FE	MOVW	HL, #HDTSA		;在 HL 中设定十六进制 2 位代码数	
	28	28							
	29	29	0086	R9A9300	CALL	!CONVAH		;ASCII<- HEX 转换	
	30	30						;将 ASCII 码输出至 BC 寄存器中去	
	31	31	0089	F1421FE	MOVW	DE, #STASC		;设置存储 ASCII 码的表地址至 DE	
	32	32	008C	63	MOV	A, B			
	33	33	008D	95	MOV	[DE], A			
	34	34	008E	84	INCW	DE			
	35	35	000F	62	MOV	A, C			
	36	36	0090	95	MOV	[DE], A			
	37	37							
	38	38	0091	FAFE	BR	\$\$			
	39	39							
	40	40					END		

9.3 列表转换器的启动

9.3.1 列表转换器的启动

有如下两种方法可以用于启动列表转换器。

(1) 由命令行启动

X>lcnv78k0	[Δ选项]...	Δ输入文件名	[Δ选项]...	[Δ]
(a)	(b)	(c)	(d)	(c)

- (a) 当前驱动器名
- (b) 列表转换器的命令行名
- (c) 输入用于列表文件操作的详细说明
用双引号（"）引用包含空格的路径。
- (d) 汇编列表的基本名
用双引号（"）引用包含空格的路径文件名。

示例 C>lcnv78k0 k0main -lk0.lmf

- 注意事项**
- 在上述第(3)项中，当指定了两个或两个以上的列表转换器选项时，要用空格将各个列表转换器选项分割开来。要了解列表转换器选项详细的说明，请参阅 [9.4 列表转换器选项](#)。
 - 对上述第(4)项，请使用扩展名 .PRN。
 - 对上述第(4)项，只要汇编列表的基本名是在命令行中指定的，那么目标文件和载入模块文件的基本名就必须要与汇编列表文件的基本名相同。
文件类型也必须为如下所示类型。

表 9-2 当列表转换器启动时指定的文件类型

文件名	类型
目标模块类型	.REL
载入模块文件	.LMF

当指定与基本名不同的文件时请使用选项进行设置。

(2) 由参数文件进行启动

当启动列表转换器所需要的数据不满足命令行数据格式要求时，或者每当执行列表转换器都需要重复执行相同的列表转换器选项时，请使用参数文件启动列表转换器。

请在命令行中指定参数文件选项(-F)，以从参数文件启动列表转换器。

由参数文件启动列表转换器方法如下。

```
X>lcnv78k0 [Δ输入文件名] Δ-f 参数文件名
           |         |
           (a)      (b)
```

- (a) 参数文件名设置选项
- (b) 该文件包含启动列表转换器时所需要的数据。

备注 使用编辑器来创建参数文件。

描述参数文件内容的规则如下。

```
[[[Δ] 选项 [Δ选项] ... [Δ]Δ]]...
```

- 1) 如果在命令行中省略了输入文件名，那么在参数文件中只能指定 1 个输入文件名。
- 2) 输入文件名也可以在选项后写入。
- 3) 在参数文件中写入所有列表转换器选项及那些应该在命令行中指定的输出文件名。

示例 使用编辑器创建参数文件 (K0.PLV)。

K0.PLV 中的内容

```
;参数文件名
k0main -lk0.lmf
-ek0.elv
```

使用参数文件名 (K0.PLV) 启动列表转换器。

```
C>lcnv78k0 -fk0.plv
```

9.3.2 列表转换器执行开始和结束信息

(1) 列表转换器执行开始信息

当列表转换器启动时，在显示器上会显示一条执行启动信息。

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

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

(2) 列表转换器执行结束信息

如果没有检测到列表转换错误发生，列表转换器会向显示器输出如下信息并将控制权返回至操作系统。

```
Conversion complete
```

如果列表转换器检测到使其无法继续执行列表转换处理的致命错误出现时，列表转换就会向显示器输出一条信息并取消列表转，将控制权返回至操作系统。

示例 指定了并不存在的列表转换器选项

```
C>lcnv78k0 k0main -a

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

RA78K0 error F6018: Option is not recognized '-a'
Program aborted
```

当列表转换器输出错误信息并中止列表转换执行时，请在第 12 章 [错误信息](#) 中查找原因并采取相应措施。

9.4 列表转换器选项

9.4.1 列表转换器选项类型

这些列表转换器选项为列表转换器操作的详细说明，列表转换器选项分为以下六种类型。

表 9-3 列表转换器选项

分类	选项	说明
目标文件输入指定	-R	输入目标文件
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	输出绝对汇编列表文件
错误列表文件输出指定	-E	输出错误列表文件
参数文件名指定	-F	输入文件名和指定文件选项
帮助信息显示指定	--	在显示器上显示帮助信息

9.4.2 列表转换器选项的说明

本节包含每一个列表转换器选项的详细说明。

(1) 目标文件输入指定

目标文件输入指定 (-R)

[语法]

-R [输入文件名]

默认设置: -R 汇编列表文件名.REL

[功能]

选项 -R 的功能是指定输入目标模块文件

[应用]

当目标模块文件的基本名与汇编列表文件里的基本名不同时，或者其文件类型不是".REL"时，请指定-R选项。

[说明]

- 1) 如果有致命错误发生，不能输出绝对汇编列表文件。
- 2) 如果只指定输入文件名的基本名，则列表转换器将指定该文件类型为'.REL' 并输入该文件。

[使用示例]

汇编列表文件名为 K0MAIN.PRN，目标模块文件名为 SAMPLE.REL 以及装载文件名为 K0.LMF。

```
C>|cnv78k0 k0main.prn -rsample.p -lk0.lmf
```

(2) 载入模块文件输入指定**载入模块文件输入指定 (-L)****[语法]**

-L [输入文件名]

默认设置: -L 汇编列表文件名.LMF

[功能]

选项 -L 的功能是指定载入模块文件的输入。

[应用]

当载入模块文件的基本名与汇编列表文件里的基本名不同时，或者其文件类型不是".LMF"时，请指定-L 选项。

[说明]

- 1) 如果有致命错误发生，不能输出绝对汇编列表文件。
- 2) 如果只指定输入文件名的基本名，则列表转换器将指定该文件类型为'.LMF' 并输入该文件。

[使用示例]

汇编列表文件名为 K0MAIN.PRN，装载文件名为 SAMPLE.LMF。

```
C>lcnv78k0 k0main -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]

-O [输出文件名]

默认设置: -O 汇编列表文件名.P

[功能]

选项 -O 功能是指定输出绝对汇编列表文件。同时选项 -O 也可以指定输出的目标文件和输出文件名。

[应用]

使用选项 -O 来更改输出目标文件和绝对汇编列表文件的输出文件名。

[说明]

- 1) 文件名可以指定为磁盘型文件名或设备型文件名。但是只有 CON, PRN, NUL 及 AUX 可以指定作为设备型文件名。如果指定了 CLOCK 型, 则会发生异常中止错误。
- 2) 如果同一个设备既指定为文件名又指定为错误文件名, 则会发生异常中止错误。
- 3) 如果在指定-O 选项时省略了输出文件名, 则绝对汇编列表文件名便为'汇编列表文件名'.P'。
- 4) 如果只指定输入文件名的基本名, 则列表转换器将指定该文件类型为'.P'并输入该文件。
- 5) 如果在指定-O 选项时省略了驱动器名, 则绝对汇编列表文件将被输出到当前驱动器。

[使用示例]

创建绝对汇编列表文件 (SAMPLE.P)。

```
C>|cnv78k0 k0main.prn -osample.p -lk0.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
```

默认设置: -NE

[功能]

- 1) 指定选项-E 用以指定输出错误列表文件。同时该选项也可以指定输出的目标文件和输出文件名。
- 2) 选项 -NE 的作用是使得 -E 选项不可用。

[应用]

指定选项-E 用以将错误信息保存至文件中。

[说明]

- 1) 错误列表文件名可以被指定为磁盘型文件名或设备型文件名。但是如果指定为 CLOCK 设备型文件名，则会发生异常中止错误。
- 2) 如果同一个设备既指定为文件名又指定为绝对汇编列表文件名，则会发生异常中止错误。
- 3) 如果在指定-E 选项时省略了输出文件名，则错误列表文件名便为'汇编列表文件名.ELV'。
- 4) 如果只指定输出文件名的基本名，则列表转换器将指定该文件类型为'.ELV'并输出该文件。
- 5) 如果在指定-E 选项时省略了驱动器名，则错误列表文件将被输出到当前驱动器。
- 6) 如果同时指定了-E 选项和-NE 选项，则最后指定的选项优先。

[使用示例]

示例 创建一个错误列表文件(SAMPLE.ELV)。

```
C>lcnv78k0 k0main -esample.elv
```

参考错误列表文件(SAMPLE.ELV)。

```
RA78K0 WARNING W6701: Load module file is older than object module file'KOMAIN.LMF, KOMAIN.REL'
Pass1: start
RA78K0 error F6105: Segment name is not found in load module file 'DATA'
```

(5) 参数文件名指定**参数文件名指定 (-F)****[语法]**

-F 文件名

默认设置: 只能在命令行中输入选项和输入文件名。

[功能]

选项 -F 的功能是指定输入选项和由指定文件中输入文件名。

[应用]

- 1) 当启动列表转换器所需要的数据不满足命令行数据格式要求时, 请指定选项 -F。
- 2) 每当执行列表转换器需要重复执行相同的列表转换器选项时, 请在参数列表文件中对这些选项进行描述并指定选项-F。

[说明]

- 1) 只有磁盘型文件名可以指定为 '文件名', 如果指定了设备型文件名, 则会发生异常中止错误。
- 2) 如果省略了文件名, 则会发生异常中止错误。
- 3) 如果只指定了文件名的基本名, 则列表转换器将指定该文件类型为'.PLV'并打开该文件。
- 4) 不允许参数文件嵌套。如果在参数文件中指定选项-F, 则会发生异常中止错误。
- 5) 可写入参数文件的字符数目没有限制约束。
- 6) 用空格符, 制表符或回车符[`\n`]将选项或输入文件名隔离开来。
- 7) 写入至参数文件中的选项和输入文件名扩展存放在由命令行所指定的参数文件处。
- 8) 最后指定的扩展选项优先。
- 9) 如果选项-F 被两次或多次指定, 则会发生异常中止错误。
- 10) 所有在";" 或"#"后和换行码 (LF) 或"EOF"前输入的字符将被解释为评注。

[使用示例]

使用参数文件启动列表转换器。

参数文件(K0.PLV)的内容如下。

```
; parameter file  
k0main -lk0.lmf  
-ek0.elv
```

在命令行中输入如下命令。

```
C>lcnv78k0 -fk0.plv
```

(6) 帮助信息显示指定**帮助信息显示指定 (--)**

语法:

```
--
```

默认设置: 无显示

[功能]

选项 -- 在显示器上显示帮助信息

[应用]

显示的帮助信息为列表转换器选项的说明列表。在执行列表转换器时请参阅这些说明。

[说明]

当指定了--选项时，所有其它列表转换器选项均不可用。

注意：不能在 PM+ 上指定此选项。

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

[使用示例]

当指定了 -- 选项时，帮助信息便输出至显示器。

```
C>lcnv78k0 --
```

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

```
usage : LCNV78K0 [option [...]] input-file [option [...]]
```

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

```
-R [file] : Specify object module file.
-L [file] : Specify load module file.
-O [file] : Specify output list file (absolute assemble list file).
-F [file] : Input option or input-file name from specified file.
-E [file] : Create error list file.
-- : Show this message.
```

9.5 在 PM + 中设置选项

本节描述由 PM + 设定列表转换器选项的方法。

9.5.1 选项设置方法


在 PM +[Tools]菜单中选择[**L**ist converter options]选项，或点击  以显示 <List Converter Options> 对话框。可以在该对话框中通过输入需要的选项来设置列表转换器选项。

图 9-2. <List Converter Options> 对话框 (当选择了 <<Output>> 标签 时)

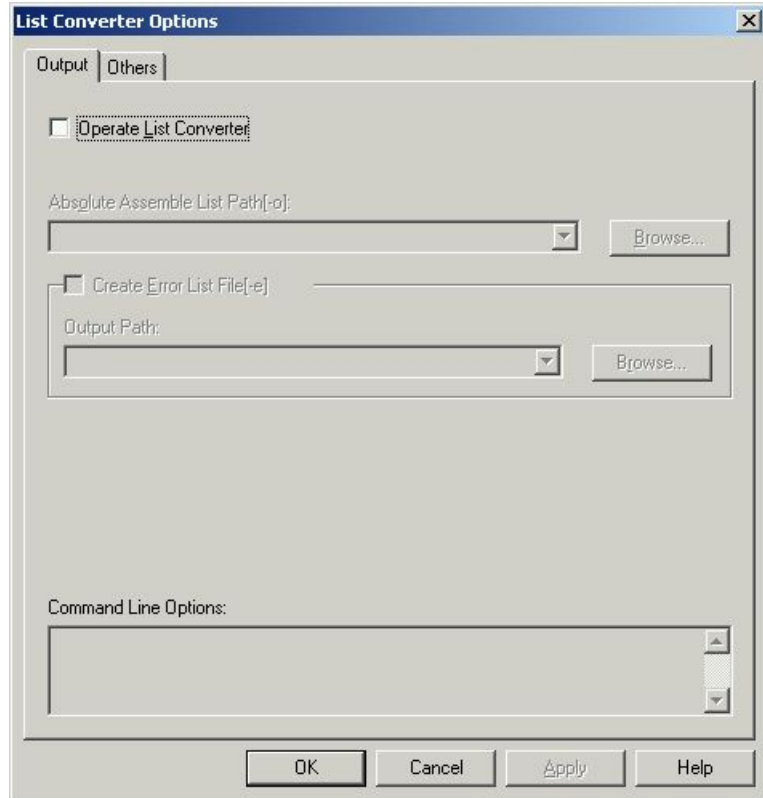
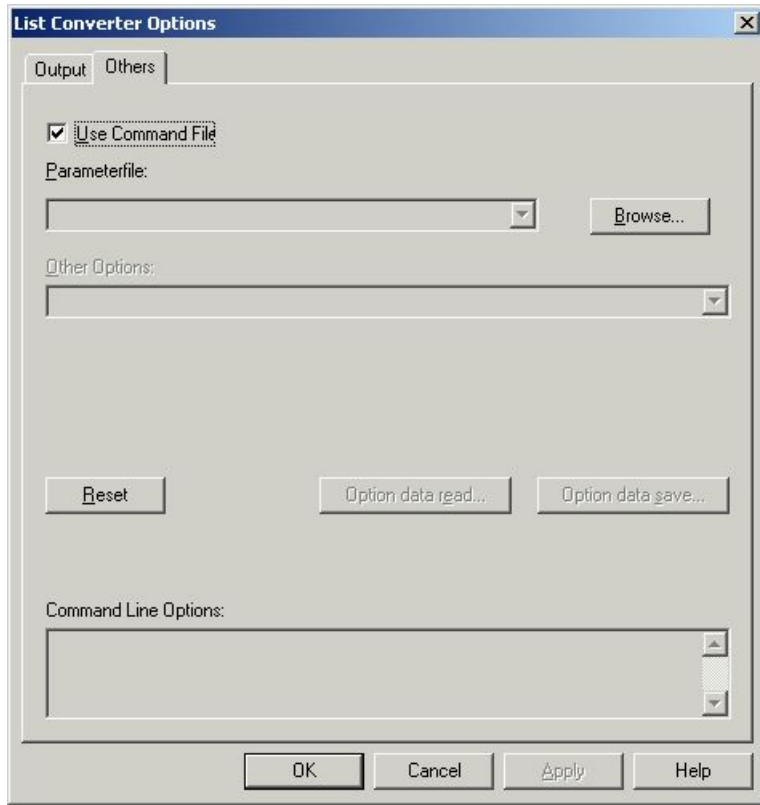


图 9-3 <List Converter Options> 对话框 (当选择了<<Others>> 标签时)



9.5.2 选项

< List Converter Options >对话框里各种选项描述如下。

<<Output>> 标签

- Operate List Converter
选中该复选框以启动列表转换器。
- Absolute Assemble List Path [-o]
通过使用[Browse...]按钮指定绝对汇编列表路径或直接输入。
- Create Error List File [-e]
选中该复选框以输出错误列表文件
- Output Path
使用[Browse]按钮错误列表文件的路径或直接输入路径名。
- 命令行选项
此编辑框是只读的。显示当前设置的选项字符串。

<<Others>> 标签

- Use Command File
选中该复选框以创建命令文件。
- Parameterfile
通过使用[Browse...]按钮指定读入所选中的用户定义参数文件或直接输入。
- Other Options
如果希望指定用复选框或单选按钮选择之外的选项，请在输入框中输入要指定的选项。
注意：不能在 PM+ 上指定帮助指定(--选项)。
- Reset
复位输入内容。
- Option data read...
打开 < Read Option Data > 对话框并在选项数据文件制定后，读入该文件。
- Option data save...
打开< Save Option Data > 对话框，将选项数据保存至选项数据文件中并为之命名。
- Command Line Options
该编辑框是只读的，显示当前设置选项字符串。

第 10 章 程序输出列表

如下是关于各功能程序输出列表的格式说明及一些其它信息。

- 结构化汇编器预处理器输出的列表
 - 错误列表
- 汇编器输出表
 - 汇编列表文件标题
 - 汇编列表
 - 符号列表
 - 交叉参考列表
 - 错误列表
- 链接器输出列表
 - 连接列表文件标题
 - 映像列表
 - 公共符号列表
 - 本地符号列表
 - 错误列表
- 目标转换器输出列表
 - 错误列表
- 库管理程序输出列表
 - 库数据输出列表
- 列表转换器输出列表
 - 绝对汇编列表
 - 错误列表

10.1 结构化汇编器预处理器输出的列表

结构化汇编器预处理器输出如下列表

表10-1 结构化汇编器预处理器输出的列表

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

错误列表存储结构化汇编器预处理器启动时输出的错误消息。

[输出格式]

Start

(1) TTT.S((2) 4) : RA78K0 (3) error (4) E1221: (5) Missing ENDIF

[输出项目的说明]

表10-2 错误列表输出项目的说明（当结构化汇编器预处理器启动时）

项目	详细说明
(1)	出现错误的源模块的名称。
(2)	出现错误的行。
(3)	错误类型。
(4)	错误编号。
(5)	错误消息。

注意：出现错误的文件名和行可能不显示。

10.2 汇编器输出列表

汇编器输出如下列表。

表10-3 汇编器输出列表

输出列表文件名称	输出列表名称
汇编列表文件	汇编列表
	符号列表
	交叉参考列表
错误列表文件	错误列表

10.2.1 汇编列表文件标题

在汇编列表文件的起始处总会输出该标题。

【输出格式】

78K/0 Series Assembler (1)Vx.xx (2)	Date:(3)xx xxx xxxx Page:(4) 1
(5)	
Command: (6)k0main.asm -c054	
Para-file:(7)	
In-file: (8)K0MAIN.ASM	
Obj-file: (9)K0MAIN.REL	
Prn-file: (10)K0MAIN.PRN	

【各输出项的说明】

表10-4 汇编列表文件标题各输出项的说明

条目	详细说明
(1)	汇编器版本号
(2)	标题字符串 由选项-LH 或控制指令 TITLE 所指定的字符串
(3)	汇编列表创建的日期
(4)	页码号
(5)	次标题字符串 由控制指令 SUBTITLE 所指定的字符串
(6)	命令行映像
(7)	参数文件的内容
(8)	输入源模块文件名
(9)	输出目标模块文件名
(10)	汇编列表文件名

10.2.2 汇编列表

汇编列表输出附带错误信息(如果有错误产生的话)的汇编结果。

[输出格式]

```

Assemble list

ALNO  STNO  ADRS  OBJECT (3) M  (4) I  SOURCE STATEMENT
(1)1  (2)1
(2)2  (2)2                (5)  NAME  SAMPM

:
28    28
29    29    (6)0006  (8)R220000(5)      CALL !CONVAH      ;ASCII<--HEX 转换
30    30                                (5)                ;将 ASCII 码输出至 BC 中
31    31    (6)0009  00000000      MOV DE, #STASC    ;ASCII 码的表地址存至 DE
(7)** ERROR F2202, STNO      31 ( 0) 非法操作数
      (6)000D  00
32    32    (6)000E  (8)0A27      (5)  MOV  A, B
33    33    (6)0010  (8)EB        (5)  MOV  [DE], A

Segment informations:

ADRS  LEN  NAME
(9)FE20 (10)0003H (11)DATA
(9)0000 (10)0002H (11)CODE
(9)0000 (10)0017H (11)?CSEG

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

[各输出项的说明]

表10-5 汇编列表各输出项的说明

条目	详细说明
(1)	源模块映像行号
(2)	行号 (包括 INCLUDE 文件的扩展及宏)
(3)	宏显示 M: 为宏定义行 #n: 为宏扩展行, n 为嵌套级数 Blank: 不是宏定义行或宏扩展行

表 10-5 汇编列表各输出项的说明

条目	详细说明
(4)	INCLUDE 显示 In: 在 INCLUDE 文件内, n 为嵌套级数 Blank: 没有使用 INCLUDE 文件。
(5)	源程序语句
(6)	位置计数器值 (4 或 5 位)
(7)	有致命错误/警告产生行
(8)	重定位数据 R: 由链接器更改的目标代码或符号值 Blank: 目标代码或符号值没有被链接器更改
(9)	段地址 (4 或 5 位)
(10)	段大小 (4 或 5 位)
(11)	段名
(12)	RA78K0 目标设备
(13)	设备文件版本号
(14)	致命错误数目
(15)	警告错误数目
(16)	最后出错行

10.2.3 符号列表

符号列表输出在源模块中定义的符号(包括局部符号)。

[输出格式]

符号表列表

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

[各输出项的说明]

表10-6 符号列表各输出项的说明

条目	详细说明
(1)	符号值 (4 或 5 位)
(2)	符号属性 CSEG: 代码段名称 ADDR: ADDRESS 属性符号 DSEG: 数据段名称 BIT: BIT 属性符号(addr.bit) BSEG: 位段名称 SABIT: BIT 属性符号(saddr.bit) MAC: 宏名称 SFBIT: BIT 属性符号(sfr.bit) MOD: 模块名称 RBIT: BIT 属性符号(A.bit, X.bit) SET: 由 SET 指令定义的符号 Blank: 由 EXTRN 或 EXTBIT 声明的外部 NUM: NUMBER 属性符号 参考符号 *****: 未定义的符号
(3)	符号参考格式 EXT: 由 EXTRN(SADDR 属性) 声明的外部参考符号 EXTB: 由 EXTBIT (saddr.bit)声明的外部参考符号 PUB: 由 PUBLIC 声明的外部定义符号 Blank: 局部符号, 段名称, 宏名称, 模块名称 *****: 未定义的符号
(4)	定义的符号名

10.2.4 交叉参考列表

交叉参考表输出的数据指示符号是在源模块中哪里(在哪一行)定义的。

【输出格式】

交叉参考列表						
NAME	VALUE	R	ATTR	RTYP	SEGNAME	XREFS
(1)?CSEG			(4)CSEG		(6)?CSEG	(7)21#
(1)CODE			(4)CSEG		(6)CODE	(7)18#
(1)CONVAH	(2)---H	(3)E		(5)EXT		(7)12 29
(1)DATA			(4)DSEG		(6)DATA	(7)14#
(1)HDTSA	(2)FE20H		(4)ADDR		(6)DATA	(7)15# 26
(1)MAIN	(2)0H		(4)ADDR	(5)PUB	(6)CODE	(7)11@ 19#
(1)SAMPM			(4)MOD			(7)2#
(1)START	(2)0H	(3)R	(4)ADDR	(5)PUB	(6)?CSEG	(7)11@ 19 22#
(1)STASC	(2)FE21H		(4)ADDR		(6)DATA	(7)16# 31

【各输出项的说明】

表10-7 交叉参考列表各输出项的说明

条目	详细说明
(1)	定义的符号名
(2)	符号值(4或5位)
(3)	重定位属性 R: 可重定位符号 E: 外部符号 Blank: 绝对符号 *: 未定义的符号
(4)	符号属性 CSEG: 代码段名称 ADDR: ADDRESS 属性符号 DSEG: 数据段名称 BIT: BIT 属性符号(addr.bit) BSEG: 位段名称 SABIT: BIT 属性符号(saddr.bit) MAC: 宏名称 SFBIT: BIT 属性符号(sfr.bit) MOD: 模块名称 RBIT: BIT 属性符号(A.bit, X.bit) SET: 由 SET 指令定义的符号 Blank: 由 EXTRN 或 EXTBIT 声明的外部 NUM: NUMBER 属性符号 参考符号 *****: 未定义的符号
(5)	符号参考格式 EXT: 由 EXTRN(SADDR 属性) 声明的外部参考符号 EXTB: EXTBIT (saddr.bit)声明的外部参考符号 PUB: 由 PUBLIC 声明的外部定义符号 Blank: 局部符号, 段名称, 宏名称, 模块名称 *****: 未定义的符号
(6)	定义的符号名

表 10-7 交叉参考列表各输出项的说明

条目	详细说明
(7)	定义/参考行号 定义行: XXXXX# 参考行: XXXXX Δ ($\Delta=1$ 空) EXTRN 声明, EXTBIT 声明, PUBLIC 声明: XXXXX@

10.2.5 错误列表

错误列表存储汇编器启动时输出的错误信息。

【输出格式】

Pass1 Start	
(1)ERROR.ASM ((2)26) : RA78K0(3) error(4) E2202:	(5)非法操作数
(1)ERROR.ASM ((2)32) : RA78K0(3) error(4) E2202:	(5)非法操作数
Pass2 Start	
(1)ERROR.ASM ((2)26) : RA78K0(3) error(4) E2202:	(5)非法操作数
(1)ERROR.ASM ((2)29) : RA78K0(3) error(4) E2407:	(5)未定义的符号参考 ' DTSA'
(1)ERROR.ASM ((2)29) : RA78K0(3) error(4) E2303:	(5)非法表达式
(1)ERROR.ASM ((2)32) : RA78K0(3) error(4) E2202:	(5)非法操作数
(1)ERROR.ASM ((2)37) : RA78K0(3) error(4) E2407:	(5)未定义的符号参考 ' F'
(1)ERROR.ASM ((2)37) : RA78K0(3) error(4) E2303:	(5)非法表达式

【各输出项的说明】

表10-8 错误列表输出项目的说明（当汇编器启动时）

条目	详细说明
(1)	有错误产生的源模块名称
(2)	错误产生行
(3)	错误类型
(4)	错误号
(5)	错误信息

注意：出现错误的文件名和行可能不显示。

10.3 链接器输出列表

链接器输出如下一些列表。

表10-9 链接器输出的列表

输出列表文件名称	输出列表名称
连接列表文件	映像列表
	公共符号列表
	局部符号列表

10.3.1 连接列表文件标题

总会在连接列表文件的起始处输出标题。

[输出格式]

```

78K/0 Series Linker (1)Vx.xx                               Date:(2)xx xxx xxxx Page:(3)1

Command:  (4)k0main.rel k0sub.rel -ok0.map -dk0.dr
Para-file: (5)
Out-file:  (6)K0.MAP
Map-file:  (7)K0MAIN.MAP
Direc-file: (8)
Directive: (9)

*** Link information ***

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

```

[各输出项的说明]

表10-10 连接列表文件标题各输出项的说明

条目	详细说明
(1)	链接器版本号
(2)	连接列表文件创建的日期
(3)	页码号 (4 或 5 位)
(4)	命令行映像 (4 或 5 位)
(5)	参数文件的内容
(6)	输出载入模块文件名
(7)	连接列表文件名
(8)	指令文件名

表 10-10 连接列表文件标题各输出项的说明

条目	详细说明
(9)	指令文件内容 (4 或 5 位)
(10)	输出至载入模块文件的段的数目 (4 或 5 位)
(11)	输出至载入模块文件的数据的大小
(12)	输出至载入模块文件的符号数

10.3.2 映像列表

映像列表输出段存储单元数据。

[输出格式]

```

*** 内存映像 ***

(1) SPACE = REGULAR

MEMORY =(2)ROM
BASE ADDRESS =(3)0000H   SIZE =(4)2000H
      OUTPUT      INPUT      INPUT      BASE      SIZE
      SEGMENT     SEGMENT    MODULE     ADDRESS
(6)CODE                                     (9)0000H   (10)0002H
                                                (11)CSEG AT

(5)* gap *                               (7)CODE    (8)SAMPM   (9)0000H   (10)0002H
(6)?CSEG                                   (9)0002H   (10)007EH
(9)0080H                                   (10)0035H
                                                (11)CSEG

(5)* gap *                               (7)?CSEG   (8)SAMPM   (9)0080H   (10)0015H
(7)?CSEG                                   (8)SAMPS   (9)0095H   (10)0020H
(9)00B5H                                   (10)00B5H   (10)1F4BH

MEMORY = RAM
BASE ADDRESS = (3)FE00H   SIZE =(4)0200H
      OUTPUT      INPUT      INPUT      BASE      SIZE
      SEGMENT     SEGMENT    MODULE     ADDRESS
(5)* gap *                               (9)FE00H   (10)0020H
(6)DATA                                   (9)FE20H   (10)0003H
                                                (11)DSEG AT

(5)* gap *                               (7)DATA    (8)SAMPM   (9)FE20H   (10)0003H
(9)FE23H                                   (10)00DDH
(5)* gap (Not Free Area) *              (9)FE00H   (10)0100H

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

[各输出项的说明]

表10-11 连接映像列表各输出项的说明

条目	详细说明
(1)	存储空间名称
(2)	存储区域名称
(3)	存储区起始地址 (4 或 5 位)
(4)	存储区大小 (4 或 5 位)

表 10-11 连接映像列表各输出项的说明

条目	详细说明
(5)	输出组 显示'gap'表示该存储区没有存储任何内容
(6)	输出至载入模块文件的段名
(7)	由目标模块文件读取的段名
(8)	输入模块名称
(9)	段起始地址（4 或 5 位）
(10)	输出/输入段的大小（4 或 5 位）
(11)	段类型及重定位属性
(12)	用于该汇编器的目标设备
(13)	设备文件版本号

10.3.3 公共符号列表

公共符号列表输出在输入模块内定义的公共符号数据。

[输出格式]

*** Public symbol list ***			
MODULE	ATTR	VALUE	NAME
(1)SAMPM	(2)ADDR	(3)0000H	(4)MAIN
(1)SAMPM	(2)ADDR	(3)0080H	(4)START
(1)SAMPS	(2)ADDR	(3)0095H	(4)CONVAH

[各输出项的说明]

表10-12 连接公共符号列表各输出项的说明

条目	详细说明
(1)	在其中定义公共符号的模块名称
(2)	符号属性 CSEG: 代码段名称 ADDR: ADDRESS 属性符号 DSEG: 数据段名称 BIT BIT 属性符号(addr.bit) BSEG: 位段名称 SABIT: BIT 属性符号(saddr.bit) MAC: 宏名称 SFBIT: BIT 属性符号(sfr.bit) MOD: 模块名称 RBIT: BIT 属性符号(A.bit, X.bit) SET: 由 SET 指令定义的符号 Blank: 由 EXTRN 或 EXTBIT 声明的外部 NUM: NUMBER 属性符号 参考符号 *****: 未定义的符号
(3)	符号值 (4 或 5 位)
(4)	公共符号名称

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 ***			
MODULE	ATTR	VALUE	NAME
(1)SAMPM	(2)MOD		(4)SAMPM
(1)SAMPM	(2)DSEG		(4)DATA
(1)SAMPM	(2)ADDR	(3)FE20H	(4)HDTSA
(1)SAMPM	(2)ADDR	(3)FE21H	(4)STASC
(1)SAMPM	(2)CSEG		(4)CODE
(1)SAMPM	(2)CSEG		(4)?CSEG
(1)SAMPMS	(2)MOD		(4)SAMPMS
(1)SAMPMS	(2)CSEG		(4)?CSEG
(1)SAMPMS	(2)ADDR	(3)00ACH	(4)SASC
(1)SAMPMS	(2)ADDR	(3)00B2H	(4)SASC1

【各输出项的说明】

表10-13 局部公共符号列表各输出项的说明

条目	详细说明
(1)	在其中定义局部符号的模块名
(2)	符号属性 CSEG: 代码段名称 ADDR: ADDRESS 属性符号 DSEG: 数据段名称 BIT: BIT 属性符号(addr.bit) BSEG: 位段名称 SABIT: BIT 属性符号(saddr.bit) MAC: 宏名称 SFBIT: BIT 属性符号(sfr.bit) MOD: 模块名称 RBIT: BIT 属性符号(A.bit, X.bit) SET: 由 SET 指令定义的符号 Blank: 由 EXTRN 或 EXTBIT 声明的外部 NUM: NUMBER 属性符号 参考符号 *****: 未定义的符号
(3)	符号值 (4 或 5 位)
(4)	局部符号名

10.3.5 错误列表

错误列表存储链接器启动时输出的错误信息。

[输出格式]

```
RA78K0 (1) error      (2) E3405:      (3) Undefined symbol 'CONVAH' in file 'K0MAIN.REL'
```

[各输出项的说明]

表10-14 错误列表各输出项的说明（当链接器启动时）

条目	详细说明
(1)	错误类型
(2)	错误号
(3)	错误信息

10.4 目标转换器输出列表

目标转换器输出列表如下。

表10-15 目标转换器各输出项的说明

输出列表文件名称	输出列表名称
错误列表文件	错误列表

10.4.1 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

库管理程序输出列表如下。

表10-16 库管理程序输出的列表

输出列表文件名称	输出列表名称
列表文件	库数据输出列表

10.5.1 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/0 Series librarian(1) Vx.xx	DATE:(2) xx xxx xx	PAGE(3)1
LIB-FILE NAME:(4)K0.LIB	((5)xx xxx xx)	
(6)0001 (7)K0MAIN.REL	((8)xx xxx xx)	
(9)MAIN	(9)START	
NUMBER OF PUBLIC SYMBOLS:	(10)2	
(6)0002 (7)K0SUB.REL	((8)xx xxx xx)	
(9)CONVAH		
NUMBER OF PUBLIC SYMBOLS:	(10)1	

[各输出项的说明]

表10-17 库数据输出列表各输出项的说明

条目	详细说明
(1)	库管理程序版本号
(2)	列表创建日期
(3)	页码数
(4)	库文件名称
(5)	库文件创建日期
(6)	模块序列号 (从 0001 开始)
(7)	模块名称
(8)	模块创建日期
(9)	公共符号名称
(10)	模块中定义的公共符号数

10.6 列表转换器输出列表

列表转换器输出如下一些列表。

表10-18 列表转换器输出的列表

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

10.6.1 绝对汇编列表

绝对汇编列表将绝对地址值嵌入汇编列表并输出该列表。

[输出格式]

与汇编器输出的汇编列表格式相同。

10.6.2 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 11 章 RA78K0 的高效使用

本章描述如何高效使用 RA78K0。

11.1 提高操作效率 (EXIT 状态功能)

当 RA78K0 中任何功能程序完成处理时，相应程序就会将其在处理过程中产生的最高错误等级存储为"EXIT status"，并将控制权返回至操作系统。

EXIT 状态如下所示：

- 正常操作: 0
- 警告产生: 0
- 致命错误产生: 1
- 异常中止产生: 2

可以用这些退出状态创建一个批处理文件，提高操作效率。

[使用示例]

批处理文件 (RA.BAT) 的内容

```
RA78K0 -c014 k0main. -g -e
echo off
IF ERRORLEVEL 1 GOTO ERR
echo\
echo on
RA78K0 -c014 k0sub.asm -g -e
echo off
IF ERRORLEVEL 1 GOTO ERR
echo\
echo on
lk78K0 k0main.rel k0sub.rel -ok0.lmf -g
echo off
IF ERRORLEVEL 1 GOTO ERR
echo\
echo on
oc78K0 k0.lmf
echo off
IF ERRORLEVEL 1 GOTO ERR
GOTO EXIT
:ERR
echo Error occurred
:EXIT
```

- 用批处理文件(RA.BAT)执行处理过程。

```
C>ra.bat
```

11.2 准备软件开发环境 (环境变量)

RA78K0 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
INC78K0:	搜索包含文件路径 (适用于结构化汇编器和汇编器)
LIB78K0:	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) 类型设定

[使用示例]

<AUTOEXEC.BAT 的内容>

```

;AUTOEXEC.BAT
Verify on
break on
PATH C:\BIN;C:\BAT;C:\RA78K0;      ← (1)
SET INC78K0 = C:\RA78K0\INCLUDE    ← (2)
SET LIB78K0 = C:\RA78K0\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:\RA78K0 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:\RA78K0\INCLUDE 中取回包含文件。
- (3) 链接器从目录 C:\RA78K0\LIB 中取回库文件。
- (4) 各功能程序将临时文件创建在目录路径 C:\TMP 下。
- (5) 注释语句中的 Kanji 被解释成 shift JIS 码。

注意事项 如果程序安装需要使用 Windows 安装器，程序会自动设置所需要的环境变量。

11.3 中断程序执行

每个功能程序都可以通过(CTRL + C)中断执行。

如果在批处理文件执行期间指定了'break on'，不管何时输入，控制权都会返回至操作系统。当指定'break off'时，仅在屏幕显示期间，控制权会返回至操作系统。在这种情况下，所有临时文件及输出文件都会被删除。

11.4 使汇编列表易于读取

使用-LH 选项或 TITLE 控制指令，在汇编列表头显示一个标题。这样通过显示一个简要指示汇编列表内容的标题，可以很容易一眼就看出汇编列表内容。

当使用了 SUBTITLE 控制指令时，同样可以显示次标题。需要了解控制指令的详细信息，请参阅 **RA78K0 结构化汇编语言篇用户手册**。

[使用示例]

打印汇编列表文件头中的标题。

```
C>RA78K0 -c054 k0main.asm -lhRA78K0_MAINROUTINE
```

这要引用 K0MAIN.PRN.

```

78K/0 Series Assembler Ex.xx RA78K0_MAINROUTINE 日期:xx xxx xxxx 页码: 1
                                     |
                                     | 标题
Command: -c054 k0main.asm -lhRA78K0_MAINROUTINE
Para- file:
In- file: K0MAIN.ASM
Obj- file:K0MAIN.REL
Prn- file:K0MAIN.PRN

Assemble list

ALNO STNO ADRS OBJECT   M I SOURCE STATEMENT

1 1
2 2          NAME SAMPM
3 3          ;*****
4 4          ;*
5 5          ;*  HEX -> ASCII 转换程序 *
6 6          ;*
7 7          ;*   主程序 *
:

```

11.5 减少程序的启动时间

11.5.1 在源程序中指定控制指令

与通常在汇编器启动程序中指定的选项具有相同功能控制指令，可以预先在源程序中指定。这样就可以不需要在每次启动汇编器时都要指定选项。

[使用示例]

```

$ PROCESSOR (054)      ;控制指令
$ XREF                ;控制指令

NAME  SAMPM
. *****
. *
. *
. *   HEX -> ASCII 转换程序  *
. *
. *   主程序                *
. *
. *****

```

11.5.2 使用 PM+

RA78K0 中的每一个程序选项都会自动存储在 PM+ 内的一个项目文件(.PRJ)中，这是因为这些存储的选项要用于第二个或紧接着的编译(MAKE)操作。所以使用 PM+，就可以避免在每次启动程序时都必需指定选项。

11.5.3 创建参数文件和子命令文件

当执行 RA78K0 的任何一个功能程序(结构化汇编器, 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, 或者相同选项在每次程序执行时都要指定时, 就可创建参数文件。

同时, 也可以将子命令注册到在库管理程序中的子命令文件中。这就使得目标模块库形成更容易。

[使用示例 1]

创建参数文件并执行汇编。

参数文件 K0MAIN.PRA 的内容。

```
;参数文件  
k0main.asm -osample.rel -g  
-psample.prn
```

在命令行中输入如下命令。

```
C>RA78K0 -fk0main.pra
```

[使用示例 2]

创建参数文件并执行汇编。

参数文件 K0.SLB 的内容。

```
;  
;库创建命令。  
;  
create k0.lib  
;  
add k0.lib k0main.rel &  
k0sub.rel  
;  
exit
```

在命令行中输入如下命令。

```
C>lb78k0 <k0.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。因此，RA78K0 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生成模块库。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，用户可以通过创建由多个程序通用模块组成的库文件，方便而有效地对程序进行管理和操作。

第 12 章 错误消息

本章解释由汇编软件包 RA78K0（结构化汇编预处理器，汇编器，连接器，目标转换器及库管理程序）输出的错误消息产生的原因，以及用户需要采取的相应措施。

12.1 错误消息概述

由 RA78K0 输出的错误消息可以分为以下 4 个级别层次。

(1) 异常中止错误(Fxxxx)

该种类型错误发生时，会使得程序无法继续执行。程序会立即退出(中断)。

如果在命令行上发现异常中止错误，如果发现另外一个命令行错误时则结束程序处理。

(2) 致命错误(Exxxx)

已经有执行错误产生。如果又发现一个错误，那么程序便退出(中断)而且不生成输出对象。

当有致命错误发生时，为了表明程序退出时没有生成输出对象，这时如果有同名对象存在，该对象也被删除。

(3) 内部错误(Cxxxx)

当出现内部错误时程序处理立刻终止

(4) 警告错误(Wxxx)

输出对象可能产生和用户预期不同。

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。

RA78K0 的错误消息分类如下。

- 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
 - E2xx--- 语句规范错误
 - E3xx --- 表达式错误
 - E4xx--- 符号错误
 - E5xx--- 段错误
 - E6xx--- 控制指令或宏错误
 - Wnxxx --- 任何类型的警告错误
- n=1~6
- 1——结构化汇编预处理器
 - 2——汇编器
 - 3——连接器
 - 4——目标转换器
 - 5——库管理程序
 - 6——列表转换器

12.2 结构化汇编器错误消息

表12-1 结构化汇编器错误消息

错误编号	错误消息	
F1001	消息	输入文件不存在 (Missing input file)
	错误产生的原因	没有指定输入文件
	用户采取的措施	指定一个输入文件
F1002	消息	输入文件过多 (Too many input files)
	错误产生的原因	指定了两个或多个输入文件
	用户采取的措施	仅指定一个输入文件
F1004	消息	非法文件名 '文件名' (Illegal file name 'file name')
	错误产生的原因	要么是文件名中有非法字符，要么是文件名的字符个数超出了限制范围。
	用户采取的措施	输入合法字符文件名并且保证字符数不要超出限制范围。
F1005	消息	非法文件指定 '文件名' (Illegal file specification 'file name')
	错误产生的原因	指定了一个非法文件
	用户采取的措施	指定一合法文件
F1006	消息	没有找到 '文件名' (File not found 'file name')
	错误产生的原因	指定的文件不存在。
	用户采取的措施	指定已存在的文件。
F1008	消息	文件指定与 '文件名' 冲突 (File specification conflicted 'file name')
	错误产生的原因	I/O 文件名重复指定
	用户采取的措施	指定不同的 I/O 文件名
F1009	消息	无法生成文件 '文件名' (Unable to make file 'file name')
	错误产生的原因	指定的文件为写保护文件
	用户采取的措施	将指定文件的写保护取消掉
F1010	消息	文件目录没有找到 '文件名' (Directory not found 'file name')
	错误产生的原因	驱动器不存在和/或输出文件名中所包含的目录不存在
	用户采取的措施	指定已存在驱动器和/或目录
F1011	消息	非法 '选项' 路径 (Illegal path 'option')
	错误产生的原因	指定的路径与选项中指定的参数路径不同
	用户采取的措施	指定一正确的路径名
F1012	消息	缺少参数 '选项' (Missing parameter 'option')
	错误产生的原因	必需的参数没有指定
	用户采取的措施	指定需要的参数

表 12-1 结构化汇编器错误消息

错误编号	错误消息	
F1013	消息	无需参数选项 (Parameter not needed 'option')
	错误产生的原因	指定了不需要的参数
	用户采取的措施	删除不需要的参数
F1014	消息	超出 '选项'范围 (Out of range 'option')
	错误产生的原因	指定的数值超出范围
	用户采取的措施	指定正确的数值
F1015	消息	参数太长'选项' (Parameter is too long 'option')
	错误产生的原因	参数中的字符数超出限制范围
	用户采取的措施	指定字符数在限制范围的参数
F1016	消息	非法的参数 '选项' (Illegal parameter 'option')
	错误产生的原因	参数语法不正确
	用户采取的措施	指定正确的参数
F1017	消息	参数过多'选项' (Too many parameters 'option')
	错误产生的原因	参数数目超出了限制范围
	用户采取的措施	使指定参数数目在限制范围内
F1018	消息	'选项'选项不可识别 (Option is not recognized 'option')
	错误产生的原因	选项名不正确
	用户采取的措施	指定正确的选项名
F1019	消息	参数文件嵌套 (Parameter file nested)
	错误产生的原因	在参数文件内指定了选项-F
	用户采取的措施	不要在参数文件内指定了选项-F
F1020	消息	参数文件读错误'文件名' (Parameter file read error 'file name')
	错误产生的原因	无法读取参数文件
	用户采取的措施	指定正确的参数文件
F1021	消息	存储器定位失败 (Memory allocation failed)
	错误产生的原因	没有足够的存储空间
	用户采取的措施	确保必需的存储空间
F1100	消息	无法设置控制选项-C (Can't set Control-C)
	错误产生的原因	执行结构化汇编预处理器的控制选项-C 无法设置
	用户采取的措施	再次执行结构化汇编预处理器
F1101	消息	打开/读/写/关闭'文件'错误 (Open/read/write/close error on 'file name')
	错误产生的原因	由于文件 I/O 错误, 不能正常地 打开/读/写/关闭 文件
	用户采取的措施	指定正确的文件

表 12-1 结构化汇编器错误消息

错误编号	错误消息	
F1102	消息	找不到 '文件名' (Can't find 'file name')
	错误产生的原因	要么是包含文件名不存在, 要么是包含文件名与输入文件或输出文件名一起指定。
	用户采取的措施	指定正确的路径, 目录和文件名
F1103	消息	非法的包含文件 '文件名' (Illegal include file 'file name')
	错误产生的原因	指定了非法的包含文件的文件名
	用户采取的措施	指定正确的文件名
F1104	消息	非法(-sc) 字符 (Illegal (-sc) character)
	错误产生的原因	在-SC 选项中指定了不能用作符号的字符
	用户采取的措施	指定正确的字符
F1105	消息	无法定义保留字符 (Can't define the reserved symbol)
	错误产生的原因	在选项-D 中指定了保留字
	用户采取的措施	不要在选项-D 中指定保留字
F1106	消息	相同的 PROCESSOR 控制 (Duplicate PROCESSOR control)
	错误产生的原因	在源文件中多次指定了 PROCESSOR 控制指令。 产品类型与选项-C 中指定的类型不同。
	用户采取的措施	只指定 PROCESSOR 控制指令一次。 更正产品类型名称。
F1107	消息	没有指定处理器 (No processor specified)
	错误产生的原因	没有指定设备类型。
	用户采取的措施	指定设备类型。
F1108	消息	指定了非法的处理器类型 (Illegal processor type specified)
	错误产生的原因	源文件里 PROCESSOR 控制指令中的设备类型指定不正确。
	用户采取的措施	指定正确的设备类型。
F1109	消息	指定了非法的处理器类型 -C (Illegal processor type specified -C)
	错误产生的原因	-C 选项中的设备类型指定不正确的。
	用户采取的措施	指定正确的设备类型。
F1110	消息	在模块标题之外无法使用这种控制 (Can't use this control outside module header)
	错误产生的原因	应该写入源模块头的指令却写入了通常源文件行中去了。
	用户采取的措施	在源模块头写入该指令。
F1111	消息	模块头语法错误 (Syntax error in module heade)
	错误产生的原因	写入源模块头的指令语法不正确。
	用户采取的措施	使用正确的语法写入该指令。

表 12-1 结构化汇编器错误消息

错误编号	错误信息	
C1112	消息	结构化汇编器预处理内部错误 (Structured assembler preprocessor internal error)
	错误产生的原因	结构化汇编器预处理内部产生错误。
	用户采取的措施	联系日电电子公司或 NEC 代理商。
E1201	消息	非法的#ELSE/#ENDIF (Illegal #ELSE/#ENDIF)
	错误产生的原因	#ELSE 及#ENDIF 语句写入了错误位置处。
	用户采取的措施	将#ELSE 及#ENDIF 语句写入了正确位置处。
E1202	消息	非法的#ENDCALLT (Illegal #ENDCALLT)
	错误产生的原因	#ENDCALLT 语句写入了错误位置处。
	用户采取的措施	将#ENDCALLT 语句写入正确位置处。
E1203	消息	#ENDIF 不存在 (Missing #ENDIF)
	错误产生的原因	#ENDIF 语句丢失。
	用户采取的措施	将#ENDIF 语句写入到正确位置处。
E1204	消息	#ENDCALLT 不存在 (Missing #ENDCALLT)
	错误产生的原因	#ENDCALLT 语句丢失。
	用户采取的措施	将#ENDCALLT 语句写入正确位置处。
E1205	消息	#DEFCALLT 定义过多 (Too many #DEFCALLT definition)
	错误产生的原因	所注册的 callt 指令转换模式数超出了限定范围。
	用户采取的措施	减少所注册的#defcallt 指令数目。
E1206	消息	CALL 指令过多 (Too many CALL instruction)
	错误产生的原因	由#DEFCALLT 至 #ENDCALLT 所定义的指令过多。
	用户采取的措施	指定只由#DEFCALLT 至 #ENDCALLT 定义一条指令。
E1207	消息	重复定义 (Duplicate definition)
	错误产生的原因	同时定义了相同的转化模式
	用户采取的措施	更正#DEFCALLT 注册项。
E1208	消息	符号表溢出 (Symbol table overflow)
	错误产生的原因	符号数目超出了限定范围。
	用户采取的措施	减少符号数目。
E1209	消息	语法错误 (Syntax error)
	错误产生的原因	所写入的语句语法不正确。
	用户采取的措施	使用正确的语法。
E1210	消息	嵌套级数错误 (Nest level error)
	错误产生的原因	嵌套时出现了错误 (溢出, 嵌套级数等等)
	用户采取的措施	使用正确的控制语句。

表 12-1 结构化汇编器错误消息

错误编号	错误信息	
E1211	消息	一行内字符数过多 (Too many characters in a line)
	错误产生的原因	行内字符数长度超出了限定数目范围。
	用户采取的措施	在一行内指定 2046 个字符或更少数目。
E1212	消息	包含文件数过多 (Too many include files)
	错误产生的原因	包含文件内包含伪指令。
	用户采取的措施	不要在包含文件内包含伪指令。
E1214	消息	非法 BREAK 语句 (Illegal BREAK)
	错误产生的原因	BREAK 语句写入到了错误的位置处
	用户采取的措施	将 BREAK 语句写入正确位置处。
E1215	消息	非法 CONTINUE 语句 (Illegal CONTINUE)
	错误产生的原因	CONTINUE 语句写入到了错误的位置处
	用户采取的措施	将 CONTINUE 语句写入正确位置处。
E1216	消息	非法的 CASE/DEFAULT/ENDS 语句 (Illegal CASE/DEFAULT/ENDS)
	错误产生的原因	CASE/DEFAULT/ENDS 语句写入到了错误的位置处
	用户采取的措施	将 CASE/DEFAULT/ENDS 语句写入正确位置处。
E1217	消息	非法的 ELSEIF/ELSE/ENDIF 语句 (Illegal ELSEIF/ELSE/ENDIF)
	错误产生的原因	ELSEIF/ELSE/ENDIF 语句写入到了错误的位置处
	用户采取的措施	将 ELSEIF/ELSE/ENDIF 语句写入正确位置处。
E1218	消息	非法的 NEXT 语句 (Illegal NEXT)
	错误产生的原因	NEXT 语句写入到了错误的位置处
	用户采取的措施	将 NEXT 语句写入正确位置处。
E1219	消息	非法的 ENDW 语句 (Illegal ENDW)
	错误产生的原因	ENDW 语句写入到了错误的位置处
	用户采取的措施	将 ENDW 语句写入正确位置处。
E1220	消息	非法的 UNTIL/UNTIL_BIT 语句 (Illegal UNTIL/UNTIL_BIT)
	错误产生的原因	UNTIL 和 UNTIL_BIT 语句写入到了错误的位置处
	用户采取的措施	将 UNTIL 和 UNTIL_BIT 语句写入正确位置处。
E1221	消息	ENDIF 不存在 (Missing ENDIF)
	错误产生的原因	ENDIF 语句丢失。
	用户采取的措施	将 ENDIF 语句写入正确位置处。
E1222	消息	ENDS 不存在 (Missing ENDS)
	错误产生的原因	ENDS 语句丢失。
	用户采取的措施	将 ENDS 语句写入正确位置处。

表 12-1 结构化汇编器错误消息

错误编号	错误信息	
E1223	消息	ENDW 不存在 (Missing ENDW)
	错误产生的原因	ENDW 语句丢失。
	用户采取的措施	将 ENDW 语句写入正确位置处。
E1224	消息	NEXT 不存在 (Missing NEXT)
	错误产生的原因	NEXT 语句丢失。
	用户采取的措施	将 NEXT 语句写入正确位置处。
E1225	消息	UNTIL/UNTIL_BIT 不存在 (Missing UNTIL/UNTIL_BIT)
	错误产生的原因	UNTIL 和 UNTIL_BIT 语句丢失。
	用户采取的措施	将 UNTIL 和 UNTIL_BIT 语句写入正确位置处。
E1226	消息	行中有非法字符 (Illegal character in a line)
	错误产生的原因	源程序行中写入了错误的字符
	用户采取的措施	删除写入源程序行中的错误字符
E1227	消息	行中有非法操作数 (Illegal operand in a line)
	错误产生的原因	替换数据大小及比较条件格式不正确
	用户采取的措施	指定正确的数据大小
E1228	消息	操作数中非法 SFR 访问 (Illegal SFR access in operand)
	错误产生的原因	写入了无法访问替换格式的 sfr 符号。
	用户采取的措施	检查 sfr 符号的访问状态并写入正确的 sfr 符号。
E1229	消息	该符号是保留“符号名” (This symbol is reserved 'symbol name')
	错误产生的原因	所使用的符号是保留字。
	用户采取的措施	更改符号名。
E1230	消息	外部资源行溢出 (Out source line overflow)
	错误产生的原因	外部资源行的字符数已经超过上限
	用户采取的措施	删除对外部资源行的不必要的描述
W1301	消息	符号重新定义 (Symbol redefinition)
	错误产生的原因	该符号由 #define 语句定义一次以上。
	程序处理情况	最近定义的符号有效。
	用户采取的措施	使最先定义的符号有效, 更正语法。
W1302	消息	重复的 PROCESSOR 选项和控制 (Duplicate PROCESSOR option and control)
	错误产生的原因	由选项-C 指定的设备类型与在 \$PROCESSOR 控制命令中指定的设备类型不一样
	程序处理情况	由选项-C 指定的设备类型有效, 忽略在 \$PROCESSOR 控制命令中指定的设备类型。
	用户采取的措施	检查在选项-C 指定的设备类型是否正确。

12.3 汇编器错误消息

表12-2 汇编器错误消息

错误编号	错误消息	
F2001	消息	输入文件不存在 (Missing input file)
	错误产生的原因	没有指定输入文件
	用户采取的措施	指定输入文件
F2002	消息	输入文件过多 (Too many input files)
	错误产生的原因	指定了两个或多个输入文件
	用户采取的措施	仅指定一个输入文件
F2004	消息	非法文件名 '文件名' (Illegal file name 'file name')
	错误产生的原因	要么是文件名中有非法字符, 要么是文件名的字符个数超出了限制范围。
	用户采取的措施	输入合法字符文件名并且保证字符数不要超出限制范围。
F2005	消息	非法文件指定 '文件名' (Illegal file specification 'file name')
	错误产生的原因	指定了一个非法文件
	用户采取的措施	指定一合法文件
F2006	消息	在 '文件名' 找不到文件 (File not found 'file name')
	错误产生的原因	指定的文件不存在
	用户采取的措施	指定一存在的文件
F2008	消息	文件指定与 '文件名' 冲突 (File specification conflicted 'file name')
	错误产生的原因	I/O 文件名重复指定
	用户采取的措施	指定不同的 I/O 文件名
F2009	消息	无法生成文件 '文件名' (Unable to make file 'file name')
	错误产生的原因	指定的文件为写保护文件
	用户采取的措施	将指定文件的写保护取消掉
F2010	消息	文件目录没有找到 '文件名' (Directory not found 'file name')
	错误产生的原因	驱动器不存在和/或输出文件名中所包含的目录不存在
	用户采取的措施	指定已存在驱动器和/或目录
F2011	消息	非法 '选项' 路径 (Illegal path 'option')
	错误产生的原因	指定的路径与选项中指定的参数路径不同
	用户采取的措施	指定一正确的路径名
F2012	消息	缺少参数 '选项' (Missing parameter 'option')
	错误产生的原因	必需的参数没有指定
	用户采取的措施	指定需要的参数

表 12-2 汇编器错误消息

错误编号	错误消息	
F2013	消息	无需参数选项 (Parameter not needed 'option')
	错误产生的原因	指定了一个不必要的参数
	用户采取的措施	删除不必要的参数
F2014	消息	超出 '选项' 范围 (Out of range 'option')
	错误产生的原因	指定的数值不在范围内
	用户采取的措施	指定一正确的数值
F2015	消息	参数选择过长 (Parameter is too long 'option')
	错误产生的原因	参数的字符数超过限度
	用户采取的措施	指定一字符数在限度内的参数
F2016	消息	非法的参数 '选项' (Illegal parameter 'option')
	错误产生的原因	参数语法不正确
	用户采取的措施	指定一正确的参数
F2017	消息	参数选择过多 (Too many parameters 'option')
	错误产生的原因	参数总数超过限度
	用户采取的措施	指定在限度内的参数
F2018	消息	'选项' 选项不可识别 (Option is not recognized 'option')
	错误产生的原因	选择名不正确
	用户采取的措施	指定正确的选项名
F2019	消息	数据文件嵌套 (Parameter file nested)
	错误产生的原因	在参数文件内指定了选项-F
	用户采取的措施	不要在参数文件内指定选项-F
F2020	消息	参数文件读错误 '文件名' (Parameter file read error 'file name')
	错误产生的原因	无法读取参数文件
	用户采取的措施	指定正确的参数文件
F2021	消息	存储器定位失败 (Memory allocation failed)
	错误产生的原因	没有足够的存储空间
	用户采取的措施	确保必需的存储空间
F2101	消息	源程序大小为 0 '文件名' (Source file size 0 'file-name')
	错误产生的原因	输入了文件大小为 0 模块。
F2102	消息	指定了非法的处理器类型 (Illegal processor type specified)
	错误产生的原因	在目标设备指定时产生错误。
F2103	消息	模块头语法错误 (Syntax error in module header)
	错误产生的原因	可以写入源模块头的控制指令格式产生错误。

表 12-2 汇编器错误消息

错误编号	错误消息	
F2104	消息	在模块标题之外无法使用这种控制 (Can't use this control outside module header)
	错误产生的原因	在源模块头指定的控制指令写进了普通源程序中去了。
F2105	消息	相同的 PROCESSOR 控制 (Duplicate PROCESSOR control)
	错误产生的原因	PROCESSOR 控制指令不止一次写进了源模块头。
F2106	消息	用于模块名的非法源文件名 (Illegal source file name for module name)
	错误产生的原因	模块名无法创建, 因为源文件名的基本名中包含有非法符号结构字符。
F2107	消息	缺省段?CSEG 已被使用 (Default segment ?CSEG is already used)
	错误产生的原因	试图用缺省段定义一未定义段。
F2108	消息	符号表溢出 '符号名' (Symbol table overflow 'symbol name')
	错误产生的原因	已经超出可定义的符号数。
F2109	消息	DS 数太多 (Too many DS)
	错误产生的原因	段中目标代码之间空隙太多, 原因是 DS 指令使用过多, 因此数据无法输出至目标文件中去。
F2110	消息	字符串表溢出 (String table overflow)
	错误产生的原因	超出了字符串表限定范围。
	用户采取的措施	将符号数减少至 9 个或 9 个以下字符。
F2111	消息	目标代码超过了个 128 字节 (Object code more than 128 bytes)
	错误产生的原因	在源程序语句中每行目标代码超过了个 128 字节。
F2112	消息	没有指定处理器 (No processor specified)
	错误产生的原因	在命令行或源模块文件中没有指定目标设备。
F2114	消息	在 C 语言源程序中, asm 语句的局部符号名必需要以 '?L' 开头 (Local symbol name of asm statement must begin with '?L' in C source.)
	错误产生的原因	在 C 语言源程序中, 用 #asm 语句描述的局部符号名不是以 '?L' 开头。
E2201	消息	语法错误 (Syntax error)
	错误产生的原因	使用了错误的语句格式。
E2202	消息	操作数非法 (Illegal operand)
	错误产生的原因	指定的操作数非法。
E2203	消息	寄存器非法 (Illegal register)
	错误产生的原因	指定了不允许指定的寄存器。
E2204	消息	非法字符 (Illegal character)
	错误产生的原因	在源程序模块中指定了一非法字符。
E2205	消息	字符串中出现意外字符 LF (Unexpected LF in string)
	错误产生的原因	在字符串关闭之前, 字符串中出现回车代码。

表 12-2 汇编器错误消息

错误编号	错误消息	
E2206	消息	字符串中出现意外 EOF (Unexpected EOF in string)
	错误产生的原因	在字符串关闭之前, 字符串中出现文件结束代码。
E2207	消息	字符串中出现意外 null 代码 (Unexpected null code in string)
	错误产生的原因	有一个 null 代码(00H)写入字符串中。
E2209	消息	行号过多 (Too many line number)
	错误产生的原因	单个文件中的行号超过了限制。
E2301	消息	表达式太复杂 (Too complex expression)
	错误产生的原因	表达式太复杂。
E2302	消息	希望是绝对表达式 (Absolute expression expected)
	错误产生的原因	指定了可重定位表达式。
E2303	消息	非法的表达式 (Illegal expression)
	错误产生的原因	所使用的表达式格式不正确。
E2304	消息	在'文件名'表达式中有非法的符号 (Illegal symbol in expression 'file name')
	错误产生的原因	表达式中有不可以使用的符号。
E2305	消息	字符常量太长 (Too long string as constant)
	错误产生的原因	超出了字符常量的长度限定范围(4 个字符)。
E2306	消息	非法数字 (Illegal number)
	错误产生的原因	指定了错误的数值。
E2307	消息	除数为零 (Division by zero)
	错误产生的原因	有数值被零除。
E2308	消息	数值太大 (Too large integer)
	错误产生的原因	数值常量超过了 16 位。
E2309	消息	非法的位型数值 (Illegal bit value)
	错误产生的原因	指定了错误的位型数值。
E2310	消息	位型数值超出了范围 (Bit value out of range)
	错误产生的原因	位型数值超出了 0 至 7 的范围。
E2311	消息	操作数超出了范围 (n) (Operand out of range (n))
	错误产生的原因	指定的数值超出了范围 n(0 至 7)。
E2312	消息	操作数超出了范围 (字节) (Operand out of range (byte))
	错误产生的原因	操作数数值超出了范围(00H 至 FFH), 或者字节操作数的数值超出了(-128 至 +127)范围。
E2313	消息	操作数超出了范围 (addr5) (Operand out of range (addr5))
	错误产生的原因	操作数超出了 addr5 的可指定范围(40H 至 7EH)。

表 12-2 汇编器错误消息

错误编号	错误消息	
E2314	消息	数值超出了范围 (addr11) (Operand out of range (addr11))
	错误产生的原因	操作数超出了的可指定范围(800H 至 FFFH)。
E2315	消息	数值超出了范围 (saddr) (Operand out of range (saddr))
	错误产生的原因	操作数超出了 saddr 的可指定范围(0FE20H 至 0FF1FH)。
E2316	消息	数值超出了范围 (addr16) (Operand out of range (addr16))
	错误产生的原因	操作数超出 addr16 的可指定范围(根据目标设备而不同)。
E2317	消息	期望为偶数表达式 (Even expression expected)
	错误产生的原因	为字访问指定了奇数地址。
E2318	消息	数值超出了范围 (sfr) (Operand out of range (sfr))
	错误产生的原因	指定的 SFR/SFRP 指令操作数超出了限定范围, 或者为 SFR 指令指定了一个奇数值操作数。
E2326	消息	SFR 非法访问操作符 (Illegal SFR access in operand)
	错误产生的原因	一个 SFR 符号越权访问了操作符
E2327	消息	非法访问操作符 (Illegal bank access in operand)
	错误产生的原因	一个符号非法访问了操作符
E2401	消息	非法的 PUBLIC 符号 '符号名' (Illegal symbol for PUBLIC 'symbol name')
	错误产生的原因	此符号不能声明为 PUBLIC 型
E2402	消息	非法的 EXTRN/EXTBIT 符号 '符号名' (Illegal symbol for EXTRN/EXTBIT 'symbol name')
	错误产生的原因	此符号不能声明为 EXTRN/EXTBIT 型
E2403	消息	不能定义 PUBLIC 符号 '符号名' (Can't define PUBLIC symbol 'symbol name')
	错误产生的原因	该符号已经有了一个 PUBLIC 声明, 不能对具有 PUBLIC 声明的符号进行 PUBLIC 定义。
	用户采取的措施	用位项而不是用 saddr.bit 定义的符号不能够进行 PUBLIC 声明。取消声明或更改 EQU 定义。
E2404	消息	没有定义的 Public 符号 '符号名' (Public symbol is undefined 'symbol name')
	错误产生的原因	具有 PUBLIC 声明的符号不可定义为 PUBLIC 符号
E2405	消息	非法的位符号 (Illegal bit symbol)
	错误产生的原因	前向参考符号使用了非法符号, 或者机器语言指令的操作数的位符号使用了非法符号。
	用户采取的措施	为位符号指定后向参考或进行 EXTBIT 声明。
E2406	消息	不可参考前向位符号 '符号名' (Can't refer to forward bit symbol 'symbol name')
	错误产生的原因	指定前向引用位符号或引用表达式中的位符号
E2407	消息	未定义的符号参考 '符号名' (Undefined symbol reference 'symbol name')
	错误产生的原因	使用了未定义的符号

表 12-2 汇编器错误消息

错误编号	错误消息	
E2408	消息	多重符号定义 '符号名' (Multiple symbol definition 'symbol name')
	错误产生的原因	符号名定义一次以上。
E2409	消息	操作数中的符号太多 (Too many symbols in operand)
	错误产生的原因	写入操作数的符号数超过了一行可以描述的符号个数。
E2410	消息	状态错误 (Phase error)
	错误产生的原因	在汇编期间符号值发生了变化 (例如,在操作数中定义了一个 EQU 符号标号, 该符号标号由于 BR 指令的最优处理而被更改。)
E2411	消息	该符号是保留值 '符号名' (This symbol is reserved 'symbol name')
	错误产生的原因	指定的符号名为保留字。
E2502	消息	非法段名 (Illegal segment name)
	错误产生的原因	非法段名符号
E2503	消息	不同段类型 '段名' (Different segment type 'segment name')
	错误产生的原因	用同一个名称定义了两个或两个以上的段, 但它们的类型是不同的。
E2504	消息	段数目太多 (Too many segment)
	错误产生的原因	定义的段数目超出了限定范围 (256)。
E2505	消息	当前段不存在 (Current segment is not exist)
	错误产生的原因	在段生成之前就写入了 ENDS 指令, 或在本段结束之后, 下一个段生成之前就立即写入了 ENDS 指令。
E2506	消息	DB, DW, DS, ORG 标号不可在 BSEG 中描述 (Can't describe DB, DW, DS, ORG, label in BSEG)
	错误产生的原因	在位段中定义了 DB, DW, DS, ORG 指令。
E2507	消息	不能在 CSEG 之外描述操作码 (Can't describe opcodes outside CSEG)
	错误产生的原因	在代码段之外定义了机器语言指令或者 BR 指令。
E2508	消息	不能在 BSEG 之外描述 DBIT (Can't describe DBIT outside BSEG)
	错误产生的原因	在位段之外定义了 DBIT 指令。
E2509	消息	指定了非法地址 (Illegal address specified)
	错误产生的原因	定位至绝对地址段的地址超出了该段范围。
E2510	消息	地址计数器溢出 (Location counter overflow)
	错误产生的原因	地址计数器超出了一个段的范围
E2511	消息	期望是段名 (Segment name expected)
	错误产生的原因	没有指定段定义指令的段名, 因为重定位属性是 AT。
E2512	消息	段大小为奇数 '段名' (Segment size is odd numbers 'segment name')
	错误产生的原因	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

错误编号	错误消息	
E2515	消息	该设备不支持安全 ID (Security ID is not supported for this device)
	错误产生的原因	指定设备不能使用安全 ID
E2517	消息	非法 bank 编号 (Illegal bank number)
	错误产生的原因	指定了非法 bank
E2601	消息	包含文件嵌套溢出 (Nesting over of include)
	错误产生的原因	包含文件嵌套超出了限定范围 (2 级)。
E2602	消息	必需指定选择语句 (Must be specified switches)
	错误产生的原因	没有指定 Switch 名称
E2603	消息	选择语句过多 (Too many switches described)
	错误产生的原因	Switch 名称超出了限定范围(每个模块限定 5)
E2604	消息	IF 语句嵌套过多 (Nesting over of IF-classes)
	错误产生的原因	IF/_IF 语句嵌套超出了限定范围(8 级)。
E2605	消息	不需要 ELSE 语句存在 (Nesting over of IF-classes)
	错误产生的原因	在不需要 ELSE 语句的地方有一条 ELSE 语句。
E2606	消息	不需要 ENDIF 语句存在 (Needless ENDIF statement exists)
	错误产生的原因	在不需要 ENDIF 语句的地方有一条 ENDIF 语句。
E2608	消息	ENDIF 不存在 (Missing ENDIF)
	错误产生的原因	IF/_IF 语句所要求的 ENDIF 语句缺失。
E2609	消息	非法的 ELSEIF 语句 (Illegal ELSEIF statement)
	错误产生的原因	ELSE 语句之后写入一条 ELSEIF 或 _ELSEIF 语句。
E2610	消息	多符号定义(MACRO) '符号名' (Multiple symbol definition (MACRO) 'symbol name')
	错误产生的原因	用于定义宏名的符号已被定义。
E2611	消息	非法参数语法 (Illegal syntax of parameter)
	错误产生的原因	宏的形式参数不正确。
E2612	消息	参数太多 (Too many parameter)
	错误产生的原因	宏名定义的形式参数数目超出了限定范围 (16)。
E2613	消息	参数重名'符号名' (Same name parameter described 'symbol name')
	错误产生的原因	指定的宏定义符号与形式参数符号相同。
E2614	消息	不能嵌套宏定义 (Can't nest macro definition)
	错误产生的原因	宏定义不能与另外一个宏定义嵌套定义。
E2615	消息	局部符号的非法语法 (Illegal syntax of local symbol)
	错误产生的原因	LOCAL 指令中指定的操作数不正确。
E2616	消息	局部符号过多 (Too many local symbols)
	错误产生的原因	可以在 1 个宏结构体描述的局部符号数超出了限定范围(64)。

表 12-2 汇编器错误消息

错误编号	错误消息	
E2617	消息	ENDM 不存在 (Missing ENDM)
	错误产生的原因	宏定义指令所要求的 ENDM 语句不存在。
E2618	消息	非法 ENDM 语法 (Illegal syntax of ENDM)
	错误产生的原因	ENDM 语句不正确
E2619	消息	宏定义非法 (Illegal defined macro)
	错误产生的原因	引用宏定义不正确
E2620	消息	非法实际参数语法 (Illegal syntax of actual parameter)
	错误产生的原因	宏的实际参数指定不正确。
E2621	消息	宏引用嵌套过多 (Nesting over of macro reference)
	错误产生的原因	宏引用嵌套超出了限定范围(8 级)。
E2622	消息	非法 EXITM 语法 (Illegal syntax of EXITM)
	错误产生的原因	EXITM 语句不正确
E2623	消息	REPT 操作数非法 (Illegal operand of REPT)
	错误产生的原因	在 REPT 指令操作数内指定了一个不允许的表达式。
E2624	消息	多于??RAFFFF (More than ??RAFFFF)
	错误产生的原因	在宏扩展期间, 所替代的局部符号数目多于 65,535。
E2625	消息	意外的 ENDM (Unexpected ENDM)
	错误产生的原因	发现了意外的 ENDM。
E2626	消息	无法描述 LOCAL 宏定义 (Can't describe LOCAL outside macro definition)
	错误产生的原因	在一般源程序语句而不是宏结构体中指定了 LOCAL 指令。
E2627	消息	在该包含文件/宏中有两个以上的段 (More than two segments in this include / macro)
	错误产生的原因	在包含文件, 宏结构体, rept-endm 块或 irp-endm 块中发现了 2 个更多的段。
W2701	消息	资源行过长 (Too long source line)
	错误产生的原因	源程序中某单行语句超过 2048 个字符
	程序处理	第 2049 及其后续字符被忽略
W2702	消息	重复的 PROCESSOR 选项和控制 (Duplicate PROCESSOR option and control)
	错误产生的原因	同时指定了目标设备(-C)命令行指定选项和源程序头中的 PROCESSOR 指令。
	程序处理	目标设备(-C)命令行指定选项有效, 而忽略源程序头中的 PROCESSOR 指令。

表 12-2 汇编器错误消息

错误编号	错误消息	
W2703	消息	多定义模块名 (Multiple defined module name)
	错误产生的原因	NAME 指令被定义 2 次或更多次
	程序处理	NAME 指令不可用且已经定义的模块名可用。
W2704	消息	已经声明的 EXTRN 符号 '符号名' (Already declared EXTRN symbol 'symbol name')
	错误产生的原因	该符号为已经声明的 EXTRN。
	用户采取的措施	在 1 个模块中指定 EXTBIT 声明一次
W2705	消息	已经声明的 EXTBIT 符号 '符号名' (Already declared EXTBIT symbol 'symbol name')
	错误产生的原因	该符号为已经声明的 EXTBIT。
	用户采取的措施	在 1 个模块中指定 EXTBIT 声明一次。
W2706	消息	END 语句不存在 (Missing END statement)
	错误产生的原因	在源文件未没有写入 END 语句。
	程序处理	假定源文件未已写入 END 语句。
W2707	消息	END 指令后有非法语句 (Illegal statement after END directive)
	错误产生的原因	END 语句后有注释, 空格, 制表符, 或 CR 码以外的语句项目。
	程序处理	忽略 END 语句后的所有语句项。
W2708	消息	已经声明了的 LOCAL 符号 '符号名' (Already declared LOCAL symbol 'symbol name')
	错误产生的原因	该符号为已经声明的 LOCAL。
	用户采取的措施	在每个宏中只声明一个符号 LOCAL。
W2709	消息	实际参数数目不够 (Few count of actual parameter)
	错误产生的原因	设定的实际参数比形式参数数目少。
	程序处理	实际参数不足处的形式参数处理为空字符串。
W2710	消息	实际参数数目过多 (Over count of actual parameter)
	错误产生的原因	设定的实际参数比形式参数数目多。
	程序处理	忽略多余的实际参数。
W2711	消息	要报告的错误过多 (Too many errors to report)
	错误产生的原因	单行中要报告的错误过多(也即 6 或更多错误)
	程序处理	第六个及其后的错误消息不输出但处理将继续进行。

表 12-2 汇编器错误消息

错误编号	错误消息	
W2712	消息	交叉参考工作区不足 (Insufficient cross-reference work area)
	错误产生的原因	用于要处理输出交叉参考列表的存储区不足。
	程序处理	不输出交叉参考列表但处理将继续进行。
W2717	消息	Normal, callt 和 callf 功能必须一起各自指定 (Normal, callt and callf functions must be described together respectively.)
	错误产生的原因	调试信息可能非法, 因为 Normal, callt 和 callf 功能没有一起指定
	程序处理	一起指定 normal 和 callt 功能
E2801	消息	非法调试信息 (Illegal debug information)
	错误产生的原因	资源行的调试信息非法
	程序处理	再次执行编译器或者结构化汇编器
F2901	消息	无法打开源程序文件'文件名' (Can't open source file 'file name')
	错误产生的原因	源程序文件不能打开。
F2902	消息	无法打开参数文件'文件名' (Can't open parameter file 'file name')
	错误产生的原因	参数文件不能打开。
F2903	消息	无法打开包含文件'文件名' (Can't open include file 'file name')
	错误产生的原因	包含文件不能打开。
F2904	消息	非法的包含文件'文件名' (Illegal include file 'file name')
	错误产生的原因	仅有驱动器名, 路径名或设备类型文件名才可以指定为包含文件名。
F2905	消息	不能打开重叠文件'文件名' (Can't open overlay file 'file name')
	错误产生的原因	重叠文件不能打开。
	用户采取的措施	确保重叠文件与汇编执行格式文件在相同的目录之下。
F2906	消息	非法的重叠文件'文件名' (Illegal overlay file 'file name')
	错误产生的原因	重叠文件内容非法
F2907	消息	不能打开目标文件'文件名' (Can't open object file 'file name')
	错误产生的原因	目标文件不能打开。
	用户采取的措施	使用在其目录路径下带有打开区域的驱动器
F2908	消息	不能打开打印文件'文件名' (Can't open print file 'file name')
	错误产生的原因	汇编列表文件不能打开。
	用户采取的措施	使用在其目录路径下带有打开区域的驱动器
F2909	消息	不能打开错误列表文件'文件名' (Can't open error list file 'file name')
	错误产生的原因	错误列表文件不能打开。
	用户采取的措施	使用在其目录路径下带有打开区域的驱动器

表 12-2 汇编器错误消息

错误编号	错误消息	
F2910	消息	不能打开临时文件 '文件名' (Can't open temporary file 'file name')
	错误产生的原因	临时文件不能打开。
	用户采取的措施	使用在其目录路径下带有打开区域的驱动器
F2913	消息	无法读取源程序文件'文件名' (Can't read source file 'file name')
	错误产生的原因	在源程序文件中有文件输入/输出错误产生
F2914	消息	无法读取参数文件'文件名' (Can't read parameter file 'file name')
	错误产生的原因	在参数文件中有文件输入/输出错误产生
F2915	消息	无法读取包含文件'文件名' (Can't read include file 'file name')
	错误产生的原因	在包含文件中有文件输入/输出错误产生
F2916	消息	无法读取覆盖文件'文件名' (Can't read overlay file 'file name')
	错误产生的原因	在覆盖文件中有文件输入/输出错误产生
F2917	消息	无法写入目标文件'文件名' (Can't write object file 'file name')
	错误产生的原因	在目标文件中有文件输入/输出错误产生
	用户采取的措施	将目标文件输出至另外一个目录下或者只指定磁盘创建一个打开区
F2918	消息	无法写入打印文件'文件名' (Can't write print file 'file name')
	错误产生的原因	在汇编列表文件中有文件输入/输出错误产生
	用户采取的措施	将汇编列表文件输出至另外一个目录下或者只指定磁盘创建一个打开区
F2919	消息	无法写入错误列表文件'文件名' (Can't write error list file 'file name')
	错误产生的原因	在错误列表文件中有文件输入/输出错误产生
	用户采取的措施	将错误列表文件输出至另外一个目录下或者只指定磁盘创建一个打开区
F2920	消息	无法读出/写入临时文件'文件名' (Can't read / write temporary file 'file name')
	错误产生的原因	在临时文件中有文件输入/输出错误产生
	用户采取的措施	将临时文件输出至另外一个目录下或者只指定磁盘创建一个打开区
C2921	消息	汇编器内部错误 (Assembler internal error)
	错误产生的原因	产生一个汇编器内部错误
	用户采取的措施	如果无法解决错误, 联系 NEC 公司或其分销商
F2922	消息	主机上没有足够的存储空间 (Insufficient memory in hostmachine)
	错误产生的原因	系统没有足够的存储空间来执行汇编器。
F2923	消息	主机上没有足够的存储空间来执行宏 (Insufficient memory for macro in hostmachine)
	错误产生的原因	在宏处理期间, 用于宏处理的存储空间不足。
	用户采取的措施	降低所定义宏的数量。

12.4 连接器错误信息

表 12-3. 连接器错误信息

错误编号	错误消息	
F3001	消息	输入文件不存在 (Missing input file)
	错误产生的原因	一个输入文件未指定。
	用户采取的措施	指定输入文件
F3004	消息	非法文件名“文件名” (Illegal file name 'file name')
	错误产生的原因	要么是文件名中有非法字符，要么是文件名的字符个数超出了限制范围。
	用户采取的措施	输入合法字符文件名并且保证字符数不要超出限制范围。
F3005	消息	非法文件指定“文件名” (Illegal file specification 'file name')
	错误产生的原因	指定了一个非法文件
	用户采取的措施	指定一个合法文件
F3006	消息	没有找到“文件名” (File not found 'file name')
	错误产生的原因	所指定的输入文件并不存在。
	用户采取的措施	制定一个存在的文件
F3007	消息	输入文件名指定重叠“文件名” (Input file specification overlapped 'file name')
	错误产生的原因	输入文件名指定重叠
	用户采取的措施	输入唯一的文件名
F3008	消息	文件指定与“文件名”冲突 (File specification conflicted 'file name')
	错误产生的原因	I/O 文件名重复指定
	用户采取的措施	指定不同的 I/O 文件名
F3009	消息	无法生成文件“文件名” (Unable to make file 'file name')
	错误产生的原因	指定的文件为写保护文件
	用户采取的措施	将指定文件的写保护取消掉
F3010	消息	文件目录没有找到“文件名” (Directory not found 'file name')
	错误产生的原因	驱动器不存在和/或输出文件名中所包含的目录不存在
	用户采取的措施	指定已存在驱动器和/或目录
F3011	消息	非法‘选项’路径 (Illegal path 'option')
	错误产生的原因	指定的路径与选项中指定的参数路径不同
	用户采取的措施	指定一个正确的路径名
F3012	消息	缺少参数‘选项’ (Missing parameter 'option')
	错误产生的原因	必需的参数没有指定
	用户采取的措施	指定需要的参数

表 12-3 连接器错误信息

错误编号	错误消息	
F3013	消息	无需参数选项 (Parameter not needed 'option')
	错误产生的原因	指定了不必要的参数
	用户采取的措施	删除不必要的参数
F3014	消息	超出 '选项' 范围 (Out of range 'option')
	错误产生的原因	指定的数值超出范围
	用户采取的措施	指定正确的数值
F3015	消息	参数太长 '选项' (Parameter is too long 'option')
	错误产生的原因	参数中的字符数超出限制范围
	用户采取的措施	指定字符数在限制范围的参数
F3016	消息	非法的参数 '选项' (Illegal parameter 'option')
	错误产生的原因	参数语法不正确
	用户采取的措施	指定正确的参数
F3017	消息	参数过多 '选项' (Too many parameters 'option')
	错误产生的原因	参数数目超出了限制范围
	用户采取的措施	使指定参数数目在限制范围内
F3018	消息	'选项' 选项不可识别 (Option is not recognized 'option')
	错误产生的原因	选项名不正确
	用户采取的措施	指定正确的选项名
F3019	消息	参数文件嵌套 (Parameter file nested)
	错误产生的原因	在参数文件内指定了选项-F
	用户采取的措施	不要在参数文件内指定了选项-F
F3020	消息	参数文件读错误 '文件名' (Parameter file read error 'file name')
	错误产生的原因	无法读取参数文件
	用户采取的措施	指定正确的参数文件
F3021	消息	存储器定位失败 (Memory allocation failed)
	错误产生的原因	没有足够的存储空间
	用户采取的措施	确保必需的存储空间
F 3030	消息	此设备不支持芯片内调试 (On-chip debug is not supported for this device)
	错误产生的原因	指定设备不能使用芯片内调试功能
	用户采取的措施	不要指定芯片内调试
F 3031	消息	该设备不支持安全 ID (Security ID is not supported for this device)
	错误产生的原因	指定设备不能使用安全 ID
	用户采取的措施	不要指定安全 ID

表 12-3. 连接器错误信息

错误编号	错误消息	
F3101	消息	'文件名' 是无效输入文件(或者有不同主机生成) ('File name' invalid input file (or made by different hostmachine))
	错误产生的原因	输入了目标文件以外的文件, 或者试图在与兼容主机上创建的目标文件进行连接。
F3102	消息	指令语法错误 (Directive syntax error)
	错误产生的原因	指令设定不正确
F3103	消息	'文件名' 为非法处理器类型 ('File name' Illegal processor type)
	错误产生的原因	汇编或编译的目标设备不是本连接器的目标设备。
	用户采取的措施	检查确保目标文件无误。检查确保汇编或编译的目标设备可以由本连接器进行处理。同时检查
F3104	消息	'文件名' 处理器类型不同于第一次输入文件的'第一次输文件名' ('File name' Different processor type from first input file 'first input file name')
	错误产生的原因	输入了其目标设备与第一次输入的目标文件不同的目标文件。
W3105	消息	库文件 '文件名' 没有公共符号 (Library file 'file name' has no public symbol)
	错误产生的原因	库文件没有公共符号, 因此, 无法连接包含于库文件中的目标模块。
F3106	消息	无法创建临时文件 '文件名' (Can't create temporary file 'file name')
	错误产生的原因	无法创建临时文件。
E3107	消息	指令中的名称'名称'已经定义 (Name 'name' in directive already defined)
	错误产生的原因	试图将保留字或者先前定义的名称定义为指令的存储区名。
E3108	消息	重叠存储区 '存储区 1' 及 '存储区 2' (Overlapped memory area 'Memory area 1' and 'Memory area 2')
	错误产生的原因	在存储区指令中定义的存储区地址重叠了。
E3109	消息	存储区 '存储区名' 太长 (至多 31 个字符) (Memory area 'Memory area name' too long name (up to 31 characters))
	错误产生的原因	在指令中指定的存储区名过长。
E3110	消息	存储区的 '存储区名' 已经定义 (Memory area 'Memory area name' already defined)
	错误产生的原因	在存储器指令中指定的存储区已经注册。
E3111	消息	存储区的 '存储区名' 重定义出界 (Memory area 'Memory area name' redefinition out of range)
	错误产生的原因	在存储器指令中指定的存储区范围超出了可重定义范围。
E3112	消息	'段名'段存储单元类型错误 (Segment 'segment name' wrong allocation type)
	错误产生的原因	在合并指令中为该段指定了错误的存储单元类型。

表 12-3. 连接器错误信息

错误编号	错误消息	
C3113	消息	连接器内部错误 (Linker internal error)
	错误产生的原因	连接器内部错误
	用户采取的措施	请联系授权代表或日电电子公司
E3114	消息	非法数字 (Illegal number)
	错误产生的原因	指令中指定的数字值不正确。
E3115	消息	数值太大 (最大 1,048,575/0FFFFFFH) (Too large value (up to 1048575/0FFFFFFH))
	错误产生的原因	指令中数值大于 1,048,575 (0FFFFFFH)。
E3116	消息	存储区 '存储区名' 定义出界 (Memory area 'Memory area name' definition out of range)
	错误产生的原因	起始地址与指令中指定的存储区大小之和超过了 65,535 (0FFFFFFH)。
E3117	消息	在所有模块中都无法找到目标芯片 (Can't find target chip in all modules)
	错误产生的原因	因为所有输入对象模块文件都指定了系统通用对象指定选项(-COMMON)，所以目标设备无法确
	用户采取的措施	删除不需要的系列通用对象指定选项 (-COMMON)
E3201	消息	合并指令中的多重段定义'段名' (Multiple segment definition 'segment name' in merge
	错误产生的原因	合并指令中指定的段已经注册 (试图用同一个段指定使用多个合并指令的存储单元)。
E3202	消息	段类型'段 1'错误匹配至文件中的'段 2' -忽略 (Segment type mismatch 'segment 1' in file
	错误产生的原因	发现有与该段段名相同的而又具有不同段类型重定位属性的段。
E3203	消息	段 '段名' 为未知的段类型 (Segment 'segment name' unknown segment type)
	错误产生的原因	输入目标文件的段数据有错(输出段连接指定不正确)。
E3204	消息	存储区/空间'名'未定义 (Memory area/space 'name' not defined)
	错误产生的原因	在合并指令中指定的存储区/空间'名'未定义。
E3205	消息	指令中的名称'名称' 属性错误 (Name 'name' in directive has bad attribute)
	错误产生的原因	在指令中描述了一个不能在段名、存储区名或存储空间名进行描述的名称条目(例如，在要求存
E3206	消息	段 '段名' 不能定位至存储区- 忽略 (Segment 'segment name' can't allocate to memory - ignored)
	错误产生的原因	段不能定位至存储区(没有段定位所需要的足够的存储区)。
E3207	消息	段 '段名'的段类型非法 (Segment 'segment name' has illegal segment type)
	错误产生的原因	该段类型数据非法

表 12-3. 连接器错误信息

错误编号	错误信息	
E3208	消息	段'段名' 属性不可更改 (Segment 'segment name' may not change attribute)
	错误产生的原因	对由汇编期间指定的重定位属性'AT xxxxH'所创建的段, 或由 ORG 指令所创建的段, 试图在指令中更改其连接类型。
E3209	消息	段'段名' 排列可能不可更改 (Segment 'segment name' may not change arrangement)
	错误产生的原因	对由汇编期间指定的重定位属性'AT xxxxH'所创建的段, 或由 ORG 指令所创建的段, 试图在指令中更改其单元地址。
	用户采取的措施	连接属性在连接期间指定的段, 其单元地址不可在汇编器中指定。
E3210	消息	段'段名' 并不存在 - 忽略 (Segment 'segment name' is not exist - ignored)
	错误产生的原因	指令中指定的段并不存在。
E3211	消息	bank 类型错误匹配'符号名'与文件中的'文件名' - 忽略 (Bank type mismatch 'symbol name' in file 'file name' - ignored)
	错误产生的原因	在指定的符号库号中有错误匹配存在
	用户采取的措施	确保符号的库号正确。
E3301	消息	可重定位目标代码地址超出限定范围 (文件 '文件名', 段 '段名', 地址 xxxxH, 类型 '寻址类型') (Relocatable object code address out of range (file 'file name', segment 'segment name', address xxxxH, type 'addressing type'))
	错误产生的原因	输入目标文件内可重定位目标代码的更正数据输出至并没有目标代码存在的地址处(重定位入口地址超出了原始数据的地址范围)。
	用户采取的措施	检查符号参考是否正确
E3302	消息	行号数据有非法符号索引(文件 '文件名', 段 '段名') (Illegal symbol index in line number (file 'file name', segment 'segment name'))
	错误产生的原因	输入目标文件内用于调试的行号数据不正确, 而且不能正确地引用符号数据。行号索引与符号索引不对应。
E3303	消息	在可重定位目标代码中无法找到符号索引 (文件 '文件名', 段 '段名', 地址 xxxxH, 类型 '寻址类型') (Can't find symbol index in relocatable object code (file 'file name', segment 'segment name', address xxxxH, type 'addressing type'))
	错误产生的原因	输入目标文件内可重定位代码的更正数据不正确, 而且不能正确地引用符号数据。可重定位条目与符号索引不对应。
	用户采取的措施	检查符号及变量引用方法是否正确。
E3304	消息	数值超出了范围 (段 '段名', 地址 xxxxH, 类型 '寻址类型') (Operand out of range (segment 'segment name', address xxxxH, type 'addressing type'))
	错误产生的原因	用于决定可重定位目标代码的操作数值超出了相应指令操作数的范围。
	用户采取的措施	在源程序中指定操作数值, 使其在每一种寻址类型所要求的范围之内。
E3305	消息	期望是偶数值(段'段名', 地址 xxxxH, 类型 '寻址类型') (Even value expected (segment 'segment name', address xxxxH, type 'addressing type'))
	错误产生的原因	用于决定 callt 或 saddrp 寻址的可重定位目标代码的操作数值为一奇数值(callt 或 saddrp 寻址的操作数必需为偶数值)。

表 12-3. 连接器错误信息

错误编号	错误信息	
E3306	消息	期望是一个 4 的倍数数值(段'段名', 地址 xxxxH, 类型 '寻址类型') (A multiple of 4 value expected (segment 'segment name', address xxxxH, type 'addressing type'))
	错误产生的原因	用于决定 saddrp 寻址的可重定位目标代码的操作数值不是 4 的倍数。
F3401	消息	'文件名' 错误符号表 ('File name' Bad symbol table)
	错误产生的原因	输入目标文件的符号数据非法。输入文件的符号条目项不是以'.file'开头的。
F3402	消息	文件 '文件名' 没有符号字符串表 (File 'file name' has no string table for symbol)
	错误产生的原因	输入目标文件的符号数据非法。
	用户采取的措施	重新执行一次汇编或编译。 通过为汇编器指定 8 个识别字符, 编译器指定 7 个识别字符, 可以避免这种错误。
E3403	消息	符号 '符号名' 与文件内 '文件名 1' 的类型不匹配。文件内第一个定义的是 '文件名 2' (Symbol 'symbol name' unmatched type in file 'file name1'. First defined in file 'file name2')
	错误产生的原因	具有相同名称的外部定义/参考符号类型在文件 1 和文件 2 中是不同的。
E3404	消息	文件 '文件名 1' 具有多重符号定义。文件第一个定义的是 '文件名 2' (Multiple Symbol definition 'symbol name' in file 'file name1'. First defined in file 'file name2')
	错误产生的原因	在目标文件 1 中定义的公共符号已经在目标文件 2 中进行了 PUBLIC 声明。
E3405	消息	文件 '文件名' 中有未定义的符号 (Undefined symbol 'symbol name' in file 'file name')
	错误产生的原因	在文件中 EXTRN 声明符号在另一个文件中未作 PUBLIC 声明。
W3406	消息	堆栈区小于 10 个字节 (Stack area less than 10 bytes)
	错误产生的原因	保护堆栈区小于为 10 个字节或者更少。(由 -S 选项指定的存储区中保护堆栈的大小为 10 字节或更少)。
W3407	消息	无法分配堆栈区 (Can't allocate stack area)
	错误产生的原因	在存储区中没有空闲区用于堆栈区保护 (在以选项 -S 所指定的存储区中, 保护堆栈区不可执行)。
W3410	消息	在文件 1 中有多个模块名称定义"模块名称", 首次定义出现在文件 2 中 (Multiple module name definition 'module name' in file 'file 1' First defined in file 'file 2')
	错误产生的原因	目标模块文件 1 中的模块名称和目标模块文件 2 中的模块名称相同。
W3411	消息	文件 '文件名 1' 中有不同的 REL 类型 (Different REL type in 'file name')
	错误产生的原因	目标文件类型版本不同。
	用户采取的措施	使用最新版本的汇编器或连接器。
E3415	消息	-QD/QF/等和 Not -QD/QF/等.REL 混用 (-QD/QF/etc. and Not -QD/QF/etc. REL are mixed)
	错误产生的原因	输入目标文件对编译器最优选项有不同的指定, 而要求其对整个程序来讲必须是相同的。所以要使用与剩余程序相同的值。

表 12-3 连接器错误信息

错误编号	错误信息	
W3416	消息	文件'文件名'(选项)'中有多 CAP/NOCAP 存在 (Multiple CAP/NOCAP are in file 'file name (option)' Defined first one in file 'file name (option)')
	错误产生的原因	所有输入目标文件 CAP/NOCAP 汇编或编译选项并不是相同的。
W3417	消息	文件'文件名'中工具名版本多于一个, 使用文件'文件名'的第一个名称 (The version of tool name in file 'file name' are more than one. Used the first one in file 'file name')
	错误产生的原因	所有输入目标文件一直到连接阶段使用的每一个工具(CC78K0, ST78K0, RA78K0)版本与设备文件版本不同。
W3418	消息	文件'文件名'版本太旧, 无法找到 TOOL 信息 (File 'file name' is old. Can't find TOOL information)
	错误产生的原因	当在输入目标文件中无法找到 TOOL 信息时, 会输出此错误信息。 通常, 当用老版本(DF-兼容型)目标文件执行连接时, 总会输出此错误信息。
W3420	消息	文件“文件名”中工具名已经出现错误/警告 (File 'file name' has already had error(s)/warning(s) by 'tool name')
	错误产生的原因	每个工具使用了(CC78K0, ST78K0, RA78K0)错误信息或警告信息, 直到输出连接阶段
E3424	消息	-ZF REL 与非 ZF REL 在文件“文件名”中被混淆 (-ZF REL and no -ZF REL are mixed in file 'file name')
	错误产生的原因	当将 flashROM 模型引导区域 ROM 程序的输入模块与 flash 区域程序的目的模块连接, 一些目的模块在编辑时不能指定 -ZF 选项
E3425	消息	相同名称的“函数名” (文件'文件名')具有不同函数 ID 值 (There are different function ID in same name 'function name' (file 'file name'))
	错误产生的原因	与由编译器声明的 EXT-FUNC 函数具有相同函数名, 但是却具有不同的 ID 值。
E3426	消息	多次输入 BOOT 文件“文件名” (Multiple input BOOT file 'file name')
	错误产生的原因	当 flashROM 模型引导区域 ROM 程序的输入模块与 flash 区域程序的目的模块连接时, 输入了两个或两个以上的引导区域 ROM 程序
E3427	消息	-BOOT REL 与非-ZF REL 在文件“文件名”中被混淆 (BOOT REL and -ZF REL are mixed in file 'file name')
	错误产生的原因	在编辑与指定的 -ZF 选项连接的过程中, 输入了 -ZF 选项指定的目的模块
E3428	消息	FLASH 的起始地址超出 ROM 的最大地址 (FLASH start address larger than ROM max address)
	错误产生的原因	flashROM 区域的起始地址超出比目标设备 ROM 的终端地址
E3429	消息	在 FLASH 文件“文件名”中发现 BOOT 段的段名 (BOOT segment 'segment name' are found in FLASH file 'file name')
	错误产生的原因	当 flashROM 模型引导区域 ROM 程序的输入模块与 flash 区域 ROM 程序的目的模块连接时, 在目的模块中有字单元地址小于 flashROM 区域起始地址的段
E3430	消息	在文件“文件名”出现不同的 FLASH 地址 (Different FLASH address in file 'file name')
	错误产生的原因	输入文件的 flashROM 区域的所有起始地址不相同

表 12-3 连接器错误信息

错误编号	错误信息	
E3431	消息	相同函数名 (文件'文件名')ID 值下对应应有不同函数名 (There are different function name in same ID (function name) (file 'file name'))
	错误产生的原因	有两个或更多的编译器声明的 EXT-FUNC 函数具有相同的 ID 值。
E3432	消息	EXT_FUNC 函数'函数名'(文件'文件名')的非法分配 (Illegal allocation of an EXT_FUNC function 'function name' (file 'file name'))
	错误产生的原因	当连接在选项-ZB 指定情况下运行时, 编译器声明为 EXT_FUNC 的函数实体存在
F3901	消息	无法打开覆盖文件'文件名' (Can't open overlay file 'file name')
	错误产生的原因	重叠文件不能打开。
	用户采取的措施	确保覆盖文件位于正确的路径下 (包含有执行程序的路径)。
F3902	消息	找不到文件'文件名' (File 'file name' not found)
	错误产生的原因	指定库文件文件不能打开。
F3903	消息	无法读取指定库文件'文件名' (Can't read input file 'file name')
	错误产生的原因	无法读取指定为输入文件的目标文件。
F3904	消息	不能打开输入文件'文件名' (Can't open output file 'file name')
	错误产生的原因	输出文件不能打开。
	用户采取的措施	检查用于创建输出文件的磁盘状况(打开能力, 介质状况等)。
F3905	消息	无法创建临时文件'文件名' (Can't create temporary file 'file name')
	错误产生的原因	无法创建用于符号项的临时文件。
	用户采取的措施	检查用于创建临时文件的磁盘状况(打开能力, 介质状况等)。
F3906	消息	无法写映像文件'文件名' (Can't write map file 'file name')
	错误产生的原因	数据无法写入连接列表文件'文件名'中去。
	用户采取的措施	检查用于创建连接列表文件的磁盘状况(打开能力, 介质状况等)。
F3907	消息	无法写输出文件'文件名' (Can't write output file 'file name')
	错误产生的原因	数据无法写入选入模块文件'文件名'中去。
	用户采取的措施	检查用于创建输出文件的磁盘状况(打开能力, 介质状况等)。
F3908	消息	无法访问临时文件'文件名' (Can't access temporary file 'file name')
	错误产生的原因	临时文件无法写入。
	用户采取的措施	检查用于创建临时文件的磁盘状况(打开能力, 介质状况等)。
F3909	消息	无法读取设备文件'设备文件名' (Can't read DEVICE_FILE file 'device file name')
	错误产生的原因	无法读取由选项指定或由控制命令指定的设备相应的设备文件。

注意 E3301~E3306 消息中显示的地址“address.xxxxH”是段分配后的绝对地址

12.5 目标转换程序错误消息

表12-4 目标转换程序错误消息

错误编号	错误消息	
F4001	消息	输入文件不存在 (Missing input file)
	错误产生的原因	没有指定输入文件
	用户采取的措施	指定输入文件
F4002	消息	输入文件过多 (Too many input files)
	错误产生的原因	指定了两个或多个输入文件
	用户采取的措施	仅指定一个输入文件
F4004	消息	非法文件名 '文件名' (Illegal file name 'file name')
	错误产生的原因	要么是文件名中有非法字符, 要么是文件名的字符个数超出了限制范围。
	用户采取的措施	输入合法字符文件名并且保证字符数不要超出限制范围。
F4005	消息	非法文件指定 '文件名' (Illegal file specification 'file name')
	错误产生的原因	指定了一个非法文件
	用户采取的措施	指定一合法文件
F4006	消息	没有找到文件名 (File not found 'file name')
	错误产生的原因	所指定的输入文件并不存在。
	用户采取的措施	当连接至编译器启动程序时, 文件输出为["启动程序名称".lmf]。在这种情况下, 使用连接器选项指定输出文件名为 [-o*.lmf]。
F4008	消息	文件指定与 '文件名' 冲突 (File specification conflicted 'file name')
	错误产生的原因	I/O 文件名重复指定
	用户采取的措施	指定不同的 I/O 文件名
F4009	消息	无法生成文件 '文件名' (Unable to make file 'file name')
	错误产生的原因	指定的文件为写保护文件
	用户采取的措施	将指定文件的写保护取消掉
F4010	消息	文件目录没有找到 '文件名' (Directory not found 'file name')
	错误产生的原因	驱动器不存在和/或输出文件名中所包含的目录不存在
	用户采取的措施	指定已存在驱动器和/或目录
F4011	消息	非法 '选项' 路径 (Illegal path 'option')
	错误产生的原因	指定的路径与选项中指定的参数路径不同
	用户采取的措施	指定一正确的路径名

表 12-4 目标转换程序错误消息

错误编号	错误信息	
F4012	消息	缺少参数 '选项' (Missing parameter 'option')
	错误产生的原因	必需的参数没有指定
	用户采取的措施	指定需要的参数
F4013	消息	无需参数选项 (Parameter not needed 'option')
	错误产生的原因	指定了不需要的参数
	用户采取的措施	删除不需要的参数
F4014	消息	超出 '选项'范围 (Out of range 'option')
	错误产生的原因	指定的数值超出范围
	用户采取的措施	指定正确的数值
F4015	消息	参数太长'选项' (Parameter is too long 'option')
	错误产生的原因	参数中的字符数超出限制范围
	用户采取的措施	指定字符数在限制范围的参数
F4016	消息	非法的参数 '选项' (Illegal parameter 'option')
	错误产生的原因	参数语法不正确
	用户采取的措施	指定正确的参数
F4017	消息	参数过多'选项' (Too many parameters 'option')
	错误产生的原因	参数数目超出了限制范围
	用户采取的措施	使指定参数数目在限制范围内
F4018	消息	'选项'选项不可识别 (Option is not recognized 'option')
	错误产生的原因	选项名不正确
	用户采取的措施	指定正确的选项名
F4019	消息	参数文件嵌套 (Parameter file nested)
	错误产生的原因	在参数文件内指定了选项-F
		不要在参数文件内指定了选项-F
F4020	消息	参数文件读错误'文件名' (Parameter file read error 'file name')
	错误产生的原因	无法读取参数文件
	用户采取的措施	指定正确的参数文件
F4021	消息	存储器定位失败 (Memory allocation failed)
	错误产生的原因	没有足够的存储空间
	用户采取的措施	确保必需的存储空间

表 12-4 目标转换程序错误消息

错误编号	错误信息	
F4100	消息	'文件名' 非法处理器类型 ('File name' Illegal processor type)
	错误产生的原因	汇编器或编译器的目标对象与该程序的目标对象不一样。
	用户采取的措施	检查载入模块文件是否正确，检查汇编器或编译器的目标对象，同时检查设备文件版本是否正确。
F4101	消息	'文件名' 无效的输入文件 (或者是由不同的主机生成的) ('File name' invalid input file (or made by different hostmachine))
	错误产生的原因	企图输入非载入模块文件，或者试图将载入模块文件转换为在兼容主机上创建的文件。
F4103	消息	符号'符号名' 非法属性 (Symbol 'symbol name' Illegal attribute)
	错误产生的原因	输入文件的符号属性存在错误。
F4104	消息	'文件名' 非法输入文件 - 不予连接 ('File name' Illegal input file - not linked)
	错误产生的原因	试图输入一目标文件。
F4105	消息	主机上没有足够的存储空间 (Insufficient memory in hostmachine)
	错误产生的原因	程序运行所需要的存储空间不足。
F4106	消息	非法符号表 (Illegal symbol table)
	错误产生的原因	输入载入模块文件的符号表存在错误。
	用户采取的措施	<p>当用 C 语言编写源程序时，请检查是否注意到了以下一些注意事项。</p> <p><注意事项></p> <p>当使用局部符号时，请用以字符串?L (?L@01, ?L@sym 等)开头的符号。但是要保证符号字符数为 8 个或更少。同时注意不要在外部定义该符号(PUBLIC 声明)。</p>
F4107	消息	不能为无 ROM 设备指定-U 选项 (Can't specify -U option for ROMless device)
	错误产生的原因	已经为没有内部 ROM 的设备指定了对象补充选项(-U)。
E4200	消息	未定义的符号'符号名' (Undefined symbol 'symbol name')
	错误产生的原因	发现了没有指定地址的符号。
	用户采取的措施	<p>定义符号值。</p> <p>该符号作为外部参考符号引用。所以如果没有进行外部定义，就在指定该符号值的模块外部对其进行外部定义。</p>
E4201	消息	超出地址范围 (Out of address range)
	错误产生的原因	载入模块文件地址超出了限定的范围。
W4300	消息	xxxxxxH - yyyyyyH 地址重叠 (xxxxxxH - yyyyyyH overlapped)
	错误产生的原因	输出了重叠地址的对象，地址重叠范围为：xxxxxxH 至 yyyyyyH。
W4301	消息	不能初始化 RAM 区 '地址' - '地址' (Can't initialize RAM area 'address' - 'address')
	错误产生的原因	初始化数值输出到 RAM 区。
	用户采取的措施	如果汇编源程序的 DSEG 段写入 DB/DW，要么将对象更改至 DS，要么将 DB/DW 写入 CSEG 段。

表 12-4 目标转换器错误消息

错误编号	错误编号	
F4900	消息	无法打开文件'文件名' (Can't open file 'file name')
	错误产生的原因	文件不能打开。
F4901	消息	无法关闭文件'文件名' (Can't close file 'file name')
	错误产生的原因	文件不能关闭。
F4902	消息	无法读取文件'文件名' (Can't read file 'file name')
	错误产生的原因	文件不能正确读取。
F4903	消息	无法存取文件'文件名' (Can't access file 'file name')
	错误产生的原因	文件不能正确读取或写入。
F4904	消息	无法写入文件'文件名' (Can't write file 'file name')
	错误产生的原因	数据不能正确地写到输出文件中去。
F4905	消息	不能打开覆盖文件'文件名' (Can't open overlay file 'file name')
	错误产生的原因	覆盖文件无法打开
	用户采取的措施	核查覆盖文件是否与执行格式在同一个目录
C4999	消息	对象转换器内部错误 (Object Converter internal error)
	错误产生的原因	这是一个内部错误
	用户采取的措施	联系 NEC 公司或其分销商

12.6 库管理程序错误消息

表12-5 库管理程序错误消息

错误编号	错误消息	
F5001	消息	输入文件不存在 (Missing input file)
	错误产生的原因	仅仅指定了选项, 而没有指定任何输入文件。
F5002	消息	输入文件过多 (Too many input files)
	错误产生的原因	输入文件总数超出了限定的范围。
F5003	消息	字符串'???'不可识别 (Unrecognized string '???')
	错误产生的原因	在对话格式命令行中输入了选项以外的其它内容。
F5004	消息	非法文件名 '文件名' (Illegal file name 'file name')
	错误产生的原因	文件名包含有操作系统不允许的字符, 或者文件名的字符数超出了限定范围。
F5005	消息	非法文件指定 '文件名' (Illegal file specification 'file name')
	错误产生的原因	文件名中指定了非法条目项。
F5006	消息	文件没有找到'文件名' (File not found 'file name')
	错误产生的原因	指定的输入文件并不存在。
F5007	消息	输入文件名指定重叠'文件名' (Input file specification overlapped 'file name')
	错误产生的原因	输入文件名指定重叠
F5008	消息	文件指定与'文件名'冲突 (File specification conflicted 'file name')
	错误产生的原因	输入或输出文件名指定重叠。
F5009	消息	无法创建文件'文件名' (Unable to make file 'file name')
	错误产生的原因	无法创建制定的输出文件。
F5010	消息	无法找到'文件名'中的目录 (Directory not found 'file name')
	错误产生的原因	输出文件名中含有并不存在的驱动器或目录。
F5011	消息	非法路径 '文件名' (Illegal path 'file name')
	错误产生的原因	在指定参数路径名的选项中输入了路径名以外的其它内容。
F5012	消息	缺少参数 '选项' (Missing parameter 'option')
	错误产生的原因	所要求的参数未指定。
F5013	消息	'选项'中不需要的参数 (Parameter not needed 'option')
	错误产生的原因	指定了并不需要的参数。
F5014	消息	超出 '选项'范围 (Out of range 'option')
	错误产生的原因	指定值超出了限定范围。
F5015	消息	参数字符太长 '选项' (Parameter is too long 'option')
	错误产生的原因	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

错误编号	错误消息	
F5016	消息	非法的参数'选项' (Illegal parameter 'option')
	错误产生的原因	参数中存在语法错误。
F5017	消息	'选项'中参数太多 (Too many parameters 'option')
	错误产生的原因	参数总数超出了限定范围。
F5018	消息	选项无法识别'选项' (Option is not recognized 'option')
	错误产生的原因	制定了错误选项。
F5021	消息	存储器定位失败 (Memory allocation failed)
	错误产生的原因	出现存储器定位错误
F5024	消息	非法字符 (Illegal character)
	错误产生的原因	发现了非法的字符或字符串
F5025	消息	限定符不唯一 (Qualifier is not unique.)
	错误产生的原因	修饰符的缩略类型不唯一。
F5026	消息	输入重定向不明确 (Umbiguous input redirect.)
	错误产生的原因	'<'后没有指定任何文件名或 不止一次指定'< Δ 文件名'
C5100	消息	内部错误 (Internal error)
	错误产生的原因	产生了内部错误。
E5101	消息	无效的子命令 (Invalid sub command)
	错误产生的原因	子命令名不正确。
E5102	消息	无效语法 (Invalid syntax)
	错误产生的原因	子命令中参数指定不正确。
E5103	消息	非法输入文件 - 不同的目标芯片 (文件: 文件名) (Illegal input file - different target chip (file: file name))
	错误产生的原因	输入目标文件中目标设备指定不正确。
E5104	消息	非法库文件 - 不同的目标芯片 (文件: 文件名) (Illegal library file - different target chip (file: file name))
	错误产生的原因	库文件中目标设备指定不正确。
E5105	消息	没有找到模块(模块: 文件名) (Module not found (module: file name))
	错误产生的原因	库文件中指定的模块不存在。
E5106	消息	模块已经存在 (模块: 文件名) (Module already exists (module: file name))
	错误产生的原因	在更新库文件或另外一个输入文件中已经存在同名模块。
E5107	消息	没有指定主库文件 (Master library file is not specify)
	错误产生的原因	在先前的操作中没有指定更新的库文件, 但是库文件名义被替换为'.'.。

表 12-5 库管理程序错误消息

错误编号	错误消息	
E5108	消息	重复处理文件 (文件: 文件名) (Multiple transaction file (file: file name))
	错误产生的原因	输入目标文件名重叠。
E5109	消息	公共符号已经存在 (符号: 符号名) (Public symbol already exists (symbol: symbol name))
	错误产生的原因	在更新库文件或其它输入文件中已经存在同名外部定义符号名。
E5110	消息	文件指定冲突(文件: 文件名) (File specification conflicted (file: file name))
	错误产生的原因	指定的输入文件与输出文件名相同。
E5111	消息	非法文件格式(文件: 文件名) (Illegal file format (file: file name))
	错误产生的原因	更新库文件格式或其它输入文件格式不正确。
E5112	消息	没有找到库文件(文件: 文件名) (Library file not found (file: file name))
	错误产生的原因	没有找到指定的库文件。
E5113	消息	没有找到目标文件(文件: 文件名) (Object module file not found (file: file name))
	错误产生的原因	没有找到指定的目标文件。
E5114	消息	没有空闲空间存储临时文件 (No free space for temporary file)
	错误产生的原因	磁盘上没有足够空间用来创建临时文件。
E5115	消息	磁盘空间不足 (Not enough memory)
	错误产生的原因	没有足够空间用以运行操作系统。
E5116	消息	子命令缓冲器满 (Sub command Buffer full)
	错误产生的原因	子命令连续行长度超出限定范围(128 × 15 个字符)。 子命令一行长度超出限定范围(128 个字符)。
E5117	消息	设备文件不能使用 (Can not use device file)
	错误产生的原因	在输入文件中指定设备文件。 在输入或输出文件的命令列表里指定 CLOCK 设备。 在输出目标文件或输出库文件里指定 PRN, CON, 或 CLOCK 设备。
E5118	消息	非法路径(文件: 文件名) (Illegal path (file: file name))
	错误产生的原因	指定文件的路径名不正确。
W5201	消息	模块未找到(模块: 文件名) (Module not found (module: file name))
	错误产生的原因	指定 REPLACE 的模块不在库文件中
E5901	消息	文件打开错误(文件: 文件名) (File open error (file: file name))
	错误产生的原因	文件不能打开。
E5902	消息	文件读出错误(文件: 文件名) (File read error (file: file name))
	错误产生的原因	文件不能正确读出
E5903	消息	文件写入错误(文件: 文件名) (File write error (file: file name))
	错误产生的原因	数据不能正写入到文件中去

表 12-5 库管理程序错误消息

错误编号	错误消息	
E5904	消息	文件查找错误(文件: 文件名) (File seek error (file: file name))
	错误产生的原因	产生文件查找错误
E5905	消息	文件关闭错误(文件: 文件名) (File close error (file: file name))
	错误产生的原因	文件不能关闭

12.7 表转换程序错误消息

表12-6 表转换程序错误消息

错误编号	错误消息	
F6001	消息	输入文件不存在 (Missing input file)
	错误产生的原因	没有指定输入文件
	用户采取的措施	指定一输入文件
F6002	错误消息	输入文件过多 (Too many input files)
	错误产生的原因	指定了两个或多个输入文件
	用户采取的措施	仅指定一个输入文件
F6004	错误消息	非法文件名 '文件名' (Illegal file name 'file name')
	错误产生的原因	要么是文件名中有非法字符, 要么是文件名的字符个数超出了限制范围。
	用户采取的措施	输入合法字符文件名并且保证字符数不要超出限制范围。
F6005	错误消息	非法文件指定 '文件名' (Illegal file specification 'file name')
	错误产生的原因	指定了一个非法文件
	用户采取的措施	指定一合法文件
F6006	错误消息	没有找到 '文件名' (File not found 'file name')
	错误产生的原因	指定的文件不存在
	用户采取的措施	指定已存在的文件
F6008	消息	文件指定与 '文件名' 冲突 (File specification conflicted 'file name')
	错误产生的原因	I/O 文件名重复指定
	用户采取的措施	指定不同的 I/O 文件名
F6009	错误消息	无法生成文件 '文件名' (Unable to make file 'file name')
	错误产生的原因	指定的文件为写保护文件
	用户采取的措施	将指定文件的写保护取消掉
F6010	错误消息	文件目录没有找到 '文件名' (Directory not found 'file name')
	错误产生的原因	驱动器不存在和/或输出文件名中所包含的目录不存在
	用户采取的措施	指定已存在驱动器和/或目录
F6011	错误消息	非法 '选项' 路径 (Illegal path 'option')
	错误产生的原因	指定的路径与选项中指定的参数路径不同
	用户采取的措施	指定一正确的路径名
F6012	错误消息	缺少参数 '选项' (Missing parameter 'option')
	错误产生的原因	必需的参数没有指定
	用户采取的措施	指定需要的参数

表 12-6 表转换程序错误消息

错误编号	错误消息	
F6013	消息	无需参数选项 (Parameter not needed 'option')
	错误产生的原因	指定了不需要的参数
	用户采取的措施	删除不需要的参数
F6014	消息	超出 '选项' 范围 (Out of range 'option')
	错误产生的原因	指定的数值超出范围
	用户采取的措施	指定正确的数值
F6015	消息	参数太长 '选项' (Parameter is too long 'option')
	错误产生的原因	参数中的字符数超出限制范围
	用户采取的措施	指定字符数在限制范围的参数
F6016	消息	非法的参数 '选项' (Illegal parameter 'option')
	错误产生的原因	参数语法不正确
	用户采取的措施	指定正确的参数
F6017	消息	参数过多 '选项' (Too many parameters 'option')
	错误产生的原因	参数数目超出了限制范围
	用户采取的措施	使指定参数数目在限制范围内
F6018	消息	'选项' 选项不可识别 (Option is not recognized 'option')
	错误产生的原因	选项名不正确
	用户采取的措施	指定正确的选项名
F6019	消息	参数文件嵌套 (Parameter file nested)
	错误产生的原因	在参数文件内指定了选项-F
	用户采取的措施	不要在参数文件内指定了选项-F
F6020	消息	参数文件读错误 '文件名' (Parameter file read error 'file name')
	错误产生的原因	无法读取参数文件
	用户采取的措施	指定正确的参数文件
F6021	消息	存储器定位失败 (Memory allocation failed)
	错误产生的原因	没有足够的存储空间
	用户采取的措施	确保必需的存储空间
F6101	消息	不是 78K/0 文件 '文件名' (File is not 78K/0 'file name')
	错误产生的原因	输入的文件名不是 78K/0 文件名。
F6102	消息	载入模块文件为不可执行文件 '文件名' (Load module file is not executable 'file name')
	错误产生的原因	企图输入非载入模块文件, 或者试图将载入模块文件转换为在兼容主机上创建的载入模块文件。
F6103	消息	载入模块文件有重定位数据 '文件名' (Load module file has relocation data 'file name')
	错误产生的原因	没有指定载入模块文件地址。

表 12-6 表转换程序错误消息

错误编号	错误信息	
F6104	消息	目标文件为可执行文件 '文件名' (Object module file is executable 'file name')
	错误产生的原因	目标文件为可执行格式
F6105	消息	在载入列表文件'段名'中没有找到段名 (Segment name is not found in module file 'segment name')
	错误产生的原因	在载入模块文件中没有找到目标文件段名。
F6106	消息	在目标文件'文件名'中没有找到段数据 (Segment name is not found in object module file 'segment name')
	错误产生的原因	在目标文件中没有找到汇编列表文件段名。
F6107	消息	内存空间不足 (Not enough memory)
	错误产生的原因	没有足够内存空间用以程序运行。
F6108	消息	载入模块文件没有符号数据, '载入模块名' (Load module file has no symbol data 'load module name')
	错误产生的原因	在连接器中指定了-NG 选项, 因此不能输出载入模块文件中的符号数据。
F6109	消息	覆盖文件不能打开'路径名' (Overlay file can not open 'path name')
	错误产生的原因	汇编覆盖文件不能打开。
F6110	消息	非法汇编列表文件 '文件名' (Illegal assembler list file 'file name')
	错误产生的原因	输入汇编列表文件类型不是汇编列表文件。
W6701	消息	载入模块文件比目标文件更老, '载入模块文件名, 目标文件名' (Load module file is older than object module file 'load module file name, object module file name')
	错误产生的原因	载入模块文件指定比目标文件指定时间要早
W6702	消息	载入模块文件比汇编模块文件更老, '载入模块文件名, 汇编列表文件名' (Load module file is older than assemble module file 'load module file name, assemble list file name')
	错误产生的原因	载入模块文件指定比汇编列表文件指定时间要早。
W6703	消息	汇编列表文件有错误语句'文件名' (Assemble list has error statement 'file name')
	错误产生的原因	汇编列表中有错误存在。
W6704	消息	在汇编列表文件'段名'中没有找到段名 (Segment name is not found in assemble list file 'segment name')
	错误产生的原因	在汇编列表中没有找到目标文件段名。
W6705	消息	段数据长度不同'段名' (Segment data length is different 'segment name')
	错误产生的原因	汇编列表文件中的段数据长度与目标文件中的段数据长度不同。
	程序处理	忽略多余的段数据并继续执行处理。
F6901	消息	打开'文件名'文件时产生错误 (File open error has occurred 'file name')
	错误产生的原因	文件不能打开。
F6902	消息	读出'文件名'文件时产生错误 (File read error has occurred 'file name')
	错误产生的原因	文件不能正确读出。

表 12-6 表转换程序错误消息

错误编号	错误信息	
F6903	消息	写入'文件名'文件时产生错误 (File write error has occurred 'file name')
	错误产生的原因	数据不能正写入到文件中。
F6904	消息	查找'文件名'文件时产生错误 (File seek error has occurred 'file name')
	错误产生的原因	产生文件查找错误
C6999	消息	内部错误 (Internal error)
	错误产生的原因	程序内部错误

12.8 PM +错误消息

本节说明 PM +帮助文件中没有描述的错误消息。需要详细了解其它 PM+错误消息，请参阅 PM+的在线帮助。

12.8.1 结构化汇编器预处理器 (ST78K0)

表12-7 DLL中显示的结构化汇编器(ST78K0)的错误消息

错误类型	错误信息	
x	消息	找不到环境变量 PATH 中显示的 ST78K0.EXE (Cannot find ST78K0.EXE shown in environmental variable PATH)
	错误产生的原因	在指定的目录中没有 ST78K0.EXE 执行格式
	用户采取的措施	重新安装 RA78K0 汇编软件包
	按钮	点击 OK 按钮，关闭该消息对话框
!	消息	找不到文件夹；可否创建？ (Cannot find folder. Will you create?)
	错误产生的原因	指定的文件夹不存在
	用户采取的措施	创建一文件夹或选择另一个文件夹
	按钮	点击 Cancel (取消) 按钮，关闭该消息对话框
!	消息	无法创建文件夹 (Not make folder.)
	错误产生的原因	指定的文件夹无法创建
	用户采取的措施	指定另一个文件夹
	按钮	点击 OK 按钮，关闭该消息对话框
!	消息	与 TAB 号码的无效顺序排列 (Invalid order size with TAB number)
	错误产生的原因	一非法排列关系的数值被指定为制表数。
	用户采取的措施	在可使用的范围内指定一数值
	按钮	点击 OK 按钮，关闭该消息对话框
!	消息	无效数值； 指令为 0 至 97 的范围内的制表数值 (Invalid value.The range of number of tabs until the instruction is from 0 to 97.)
	错误产生的原因	将一在范围外的数值作为符合在指令的制表数进行了描述
	用户采取的措施	在可使用的范围内指定一数值
	按钮	点击 OK 按钮，关闭该消息对话框
!	消息	无效数值； 操作数从 0 至 98 的范围的制表数 (Invalid value. The range of number of tabs until an operand is from 0 to 98)
	错误产生的原因	将一在范围外的数值作为符合操作数的制表数进行了描述
	用户采取的措施	在可使用的范围内指定一数值
	按钮	点击 OK 按钮，关闭该消息对话框

表 12-7 DLL 中显示的结构化汇编器(ST78K0)的错误消息

错误类型	错误信息	
!	消息	无效数值；评论为从0至99 范围的制表数 (Invalid value. The range of number of tabs until a comment is from 0 to 99)
	错误产生的原因	将一在范围外的数值作为符合评论的制表数进行了指定
	用户采取的措施	在可使用的范围内指定一数值
	按钮	点击 OK 按钮，关闭该消息对话框
!	消息	无效单词符号字符 (Invalid word symbol character)
	错误产生的原因	指定了不能被当作符号的字符
	用户采取的措施	指定一可使用的字符
	按钮	点击 OK 按钮，关闭该消息对话框
!	消息	过多的包含搜索路径；最多可指定 64 个包含搜索路径 (Too many Include Search Path. Up to 64 can be specified for Include Search Path)
	错误产生的原因	指定了过多的包含文件路径
	用户采取的措施	在可使用范围内指定包含文件路径的数目
	按钮	点击 OK 按钮，关闭该消息对话框
!	消息	包含文件路径的字符过多 (Too many characters for Include Search Path)
	错误产生的原因	包含文件路径的长度是按照超过输入范围的几个字符指定的
	用户采取的措施	指定的字符数要在输入范围内
	按钮	点击 OK 按钮，关闭该消息对话框
!	消息	(包含文件路径) 多个包含搜索路径定义 ((Include file path) Multiple Include Search Path definition)
	错误产生的原因	文件包含搜索路径进行了 2 次描述
	用户采取的措施	删除重复的路径然后再试一次
	按钮	点击 OK 按钮，关闭该消息对话框
!	消息	符号定义中符号过多，最多可以指定 30 个符号 (Too many symbols for Symbol Definition. Up to 30 symbols can be specified)
	错误产生的原因	符号定义描述的符号数超过了输入范围
	用户采取的措施	使符号数在使用范围内，然后重试
	按钮	点击 OK 按钮，关闭该消息对话框
!	消息	(符号) 符号定义中字符过多，最多用 31 个字符描述符号名 ((symbol) Too many characters for Symbol Definition. Up to 31 characters can be described for symbol name)
	错误产生的原因	被定义的符号长度大于输入范围内的字符数
	用户采取的措施	使符号的字符数在使用范围内，然后重试
	按钮	点击 OK 按钮，关闭该消息对话框

表 12-7 DLL 中显示的结构化汇编器(ST78K0)的错误消息

错误类型	错误信息	
!	消息	(符号)多重符号定义 ((symbol) Multiple symbol definition.)
	错误产生的原因	符号定义中的符号被描述了 2 次
	用户采取的措施	删除重复的符号然后再试一次
	按钮	点击 OK 按钮, 关闭该消息对话框
!	消息	不能为 (源文件名) 设定选项 (Can't set options to (source file name))
	错误产生的原因	如果单独选项中有共同选项, 单独选项的指定就将无效
	用户采取的措施	核对单独选项的指定, 然后再试一次
	按钮	点击 OK 按钮, 关闭该消息对话框

12.8.2 汇编程序 (RA78K0)

表12-8汇编程序 (RA78K0) DDL 错误信息

错误类型	错误信息	
X	消息	找不到环境变量 PATH 中显示的 RA78K0.EXE (Cannot find RA78K0.EXE shown in environmental variable PATH.)
	错误产生的原因	在指定的目录下没有 RA78K0.EXE 执行格式
	用户采取的措施	重新安装 RA78K0 汇编软件包
	按钮	点击 OK 按钮, 关闭该消息对话框
!	消息	找不到文件夹; 可否创建? (Cannot find folder. Will you create?)
	错误产生的原因	指定的文件夹不存在
	用户采取的措施	创建一文件夹或选择另一个文件夹
	按钮	点击 OK 按钮, 创建一个文件夹并关闭该消息对话框 点击 Cancel (取消) 按钮, 关闭该消息对话框
!	消息	无法创建文件夹 (Not make folder)
	错误产生的原因	指定的文件夹无法创建
	用户采取的措施	指定另一个文件夹
	按钮	点击 OK 按钮, 关闭该消息对话框
!	消息	无效数值; 每行的栏目范围从 72 至 2046 (Invalid value. The range of columns per line is from 72 to 2046)
	错误产生的原因	将超过范围的数值指定为一行的字符数
	用户采取的措施	指定一在规定范围内的数值
	按钮	点击 OK 按钮, 关闭该消息对话框
!	消息	无效数值; 每页行数的范围是从 0 或 20 至 32767 (Invalid value. The range of line per page is from 0, or 20 to 32767)
	错误产生的原因	将超过范围的数值指定为一页上的行数
	用户采取的措施	指定一在规定范围内的数值
	按钮	点击 OK 按钮, 关闭该消息对话框
!	消息	无效数值; TAB 字符范围是从 0 至 8 (Invalid value. The range of TAB character is from 0 to 8.)
	错误产生的原因	将输入范围外的数值作为符合操作数的制表数进行了描述
	用户采取的措施	指定一在规定范围内的数值
	按钮	点击 OK 按钮, 关闭该消息对话框

表 12-8 汇编程序 (RA78K0) DDL 错误信息

错误类型	错误信息	
!	消息	过多的包含搜索路径，包含搜索路径可以有 64 个 (Too many Include Search Path. Up to 64 can be specified for Include Search Path.)
	错误产生的原因	指定了在输入极限范围外数目的包含搜索路径
	用户采取的措施	指定一在极限范围内的数值
	按钮	点击 OK 按钮，关闭该消息对话框
!	消息	含搜索路径的字符过多 (Too many characters for Include Search Path)
	错误产生的原因	包含搜索路径的长度被指定为超过了输入范围的一些字符
	用户采取的措施	指定一些在输入范围内的字符
	按钮	点击 OK 按钮，关闭该消息对话框
!	消息	(包含文件路径) 多个包含搜索路径定义 ((Include file path) Multiple Include Search Path definition)
	错误产生的原因	将包含文件路径进行了 2 次描述
	用户采取的措施	删除重复的路径然后再试一次
	按钮	点击 OK 按钮，关闭该消息对话框
!	消息	符号定义的符号过多，最多可指定 30 个符号 (Too many symbols for Symbol Definition. Up to 30 symbols can be specified.)
	错误产生的原因	一些超过了输入范围的符号被指定为符号定义
	用户采取的措施	使符号数在使用范围内然后重试
	按钮	点击 OK 按钮，关闭该消息对话框
!	消息	(符号) 符号定义的字符过多，最多可用 31 个符号来描述符号名 ((symbol) Too many characters for Symbol Definition. Up to 31 characters can be described for symbol name.)
	错误产生的原因	被定义的符号长度大于输入范围内的字符数
	用户采取的措施	使符号的字符数在使用范围内，然后重试
	按钮	点击 OK 按钮，关闭该消息对话框
!	消息	(符号) 多次符号定义 ((symbol) Multiple symbol definition.)
	错误产生的原因	将符号定义的符号进行了多次描述
	用户采取的措施	删除重复的符号然后再试
	按钮	点击 OK 按钮，关闭该消息对话框

表 12-8 汇编程序 (RA78K0) DDL 错误信息

错误类型	错误信息	
!	消息	不能为 (源文件名) 设定选项 (Can't set options to (source file name))
	错误产生的原因	如果单独选项中有共同选项, 单独选项的指定就将无效
	用户采取的措施	核对单独选项的指定, 然后再试一次
	按钮	点击 OK 按钮, 关闭该消息对话框

12.8.3 连接器 (LK78K0)

表12-9 连接器 (LK78K0) DLL显示的错误信息

错误类型	错误信息	
X	消息	找不到环境变量 PATH 中显示的 LK78K0.EXE (Cannot find LK78K0.EXE shown in environmental variable PATH.)
	错误产生的原因	在指定的目录下没有 LK78K0.EXE 执行格式
	用户采取的措施	重新安装 RA78K0 汇编软件包
	按钮	点击 OK 按钮, 关闭该消息对话框
!	消息	无效数值; 警告水平的范围是从 0 至 8 (Invalid value. The range of warning level is from 0 to 2.)
	错误产生的原因	指定了输入范围外的数值作为警告水平
	用户采取的措施	指定一在限制范围内的数值
	按钮	点击 OK 按钮, 关闭该消息对话框
!	消息	无效数值; 闪存起始地址的范围是从 0h 至 0ffffh (Invalid value. The range of flash start address is from 0h to 0ffffh.)
	错误产生的原因	指定了输入范围外的数值作为闪存起始地址
	用户采取的措施	指定一在限制范围内的数值
	按钮	点击 OK 按钮, 关闭该消息对话框
!	消息	无法找到路径或文件夹, 确认路径或文件名 (Cannot find path or file. Make sure path or filename)
	错误产生的原因	无法找到指定的路径或文件夹
	用户采取的措施	指定一个正确的路径或文件夹
	按钮	点击 OK 按钮, 关闭该消息对话框
!	消息	找不到文件夹; 可否创建? (Cannot find folder. Will you create?)
	错误产生的原因	指定的文件夹不存在
	用户采取的措施	创建一文件夹或选择另一个文件夹
	按钮	点击 OK 按钮, 创建一个文件并关闭该消息对话框 点击 Cancel (取消) 按钮, 关闭该消息对话框
!	消息	没有创建文件夹 (Not make folder)
	错误产生的原因	无法创建指定的文件夹
	用户采取的措施	指定另一个文件夹
	按钮	点击 OK 按钮, 关闭该消息对话框
!	消息	无效数值; 片上调试空间的范围是从 256 至 1024 (Invalid value. The range of Size for On-Chip Debug is from 256 to 1024)
	错误产生的原因	指定了输入范围外的数值作为片上调试的空间
	用户采取的措施	指定一在限制范围内的数值
	按钮	点击 OK 按钮, 关闭该消息对话框

表 12-9 连接器 (LK78K0) DLL 显示的错误信息

错误类型	错误信息	
!	消息	安全 ID 的数字太多；最多只能指定 20 位数字作为安全 ID (Too many figures for Security ID. Up to 20 can be specified for Security ID)
	错误产生的原因	安全 ID 的指定格式不正确
	用户采取的措施	以正确的格式指定安全 ID 然后重试
	按钮	点击 OK 按钮，关闭该消息对话框
!	消息	无效安全 ID；安全 ID 应以十六进制数值指定 (Invalid Security ID. Security ID is specified in hexadecimal numbers.)
	错误产生的原因	安全 ID 的指定格式不正确
	用户采取的措施	以正确的格式指定安全 ID 然后重试
	按钮	点击 OK 按钮，关闭该消息对话框
!	消息	无效安全 ID；缺少参数 (Invalid Security ID. Missing parameter)
	错误产生的原因	没有输入安全 ID
	用户采取的措施	输入一个安全 ID
	按钮	点击 OK 按钮，关闭该消息对话框
!	消息	库文件文件太多 (Too many files for Library File. Up to 10 can be specified for Library File)
	错误产生的原因	指定了超过输入限制范围的库文件数
	用户采取的措施	输入一指定范围内的库文件数
	按钮	点击 OK 按钮，关闭该消息对话框
!	消息	(库文件) 库文件文件名太长 ((library file) Too long file name for Library File)
	错误产生的原因	指定了超过输入限制范围的字符数作为库文件名。
	用户采取的措施	输入限制范围内的字符数并重试
	按钮	点击 OK 按钮，关闭该消息对话框
!	消息	(库文件) 多个库文件定义 ((library file) Multiple Library File definition.)
	错误产生的原因	库文件描述重复
	用户采取的措施	删除重复的库文件然后重试
	按钮	点击 OK 按钮，关闭该消息对话框
!	消息	库文件搜索路径的路径过多 (Too many path for Library File Search Path. Up to 64 can be specified.)
	错误产生的原因	描述了超出限制输入范围的读取库文件的路径数
	用户采取的措施	指定限制范围内的路径数然后重试
	按钮	点击 OK 按钮，关闭该消息对话框

表 12-9 连接器 (LK78K0) DLL 显示的错误信息

错误类型	错误信息	
!	消息	(库文件读取路径), 库文件文件搜索路径字符太多 (library file read path) Too many characters for Library File Search Path)
	错误产生的原因	指定了超过输入限制范围的字符数作为库文件的读取路径。
	用户采取的措施	输入限制范围内的读取文件路径的字符数并重试
	按钮	点击 OK 按钮, 关闭该消息对话框
!	消息	(库文件读取路径), 多个库文件搜索路径定义 (library file read path) Multiple Library File Search Path definition.)
	错误产生的原因	库文件读取路径重复
	用户采取的措施	删除重复的路径然后重试
	按钮	点击 OK 按钮, 关闭该消息对话框

12.8.4 目标转换器 (OC78K0)

表12-10 目标转换器 (OC78K0) DLL显示的错误信息

错误类型	错误信息	
X	消息	找不到环境变量 PATH 中显示的 OC78K0.EXE (Cannot find OC78K0.EXE shown in environmental variable PATH.)
	错误产生的原因	在指定的目录下没有 OC78K0.EXE 执行格式
	用户采取的措施	重新安装 RA78K0 汇编软件包
	按钮	点击 OK 按钮, 关闭该消息对话框
!	消息	对象补充描述格式无效 (Invalid description format with object complement.)
	错误产生的原因	对象填充值指定的格式非法。
	用户采取的措施	指定一个可以描述的格式
	按钮	点击 OK 按钮, 关闭该消息对话框
!	消息	无效数值; 补充值的范围是从 0h 至 0ffh (Invalid value. The range of complement value is from 0h to 0ffh.)
	错误产生的原因	指定了输入范围外的数值作为补充值
	用户采取的措施	指定一在限制范围内的数值
	按钮	点击 OK 按钮, 关闭该消息对话框
!	消息	超出范围; 起始地址的范围是从 0h 到 (编程区域的最大地址, 除 SFR 区域外) (Out of range. The range of start address is from 0h to (the maximum address of program area except sfr area).)
	错误产生的原因	指定了输入范围外的数值作为起始地址
	用户采取的措施	指定一在限制范围内的数值
	按钮	点击 OK 按钮, 关闭该消息对话框
!	消息	超出范围; 空间的范围是从 1h 到 ((编程区域的最大地址, 除 SFR 区域外)-(起始地址)+1h) (Out of range. The range of size is from 1h to ((the maximum address of program area except sfr area) - (start address) + 1h).)
	错误产生的原因	指定了输入范围外的数值作为起始地址
	用户采取的措施	指定一在限制范围内的数值
	按钮	点击 OK 按钮, 关闭该消息对话框
!	消息	找不到文件夹, 可否创建? (Cannot find folder.Will you create?)
	错误产生的原因	指定的文件夹不存在
	用户采取的措施	创建一文件夹或选择另一个文件夹
	按钮	点击 OK 按钮, 创建一个文件并关闭该消息对话框 点击 Cancel 按钮, 关闭该消息对话框
!	消息	没有创建文件夹 (Not make folder.)
	错误产生的原因	无法创建指定的文件夹
	用户采取的措施	指定另一个文件夹
	按钮	点击 OK 按钮, 关闭该消息对话框

12.8.5 库管理程序 (LB78K0)

表12-11 库管理程序 (LB78K0) DLL显示的错误信息

错误类型	错误信息	
X	消息	找不到环境变量 PATH 中显示的 LB78K0.EXE (Cannot find LB78K0.EXE shown in environmental variable PATH)
	错误产生的原因	在指定的目录下没有 LB78K0.EXE 执行格式
	用户采取的措施	重新安装 RA78K0 汇编软件包
	按钮	点击 OK 按钮, 关闭该消息对话框
!	消息	找不到文件夹; 可否创建? (Cannot find folder.Will you create?)
	错误产生的原因	指定的文件夹不存在
	用户采取的措施	创建一文件夹或选择另一个文件夹
	按钮	点击 OK 按钮, 创建一个文件并关闭该消息对话框 点击 Cancel 按钮, 关闭该消息对话框
!	消息	没有创建文件夹 (Not make folder)
	错误产生的原因	无法创建指定的文件夹
	用户采取的措施	指定另一个文件夹
	按钮	点击 OK 按钮, 关闭该消息对话框

12.8.6 列表转换器 (LCNV78K0)

表12-12 库管理程序 (LVNV78K0) DLL显示的错误信息

错误类型	错误信息	
X	消息	找不到环境变量 PATH 中显示的 LCNV78K0.EXE (Cannot find LCNV78K0.EXE shown in environmental variable PATH)
	错误产生的原因	在指定的目录下没有 LCNV78K0.EXE 执行格式
	用户采取的措施	重新安装 RA78K0 汇编软件包
	按钮	点击 OK 按钮, 关闭该消息对话框
!	消息	找不到文件夹; 可否创建? (Cannot find folder. Will you create?)
	错误产生的原因	指定的文件夹不存在
	用户采取的措施	创建一文件夹或选择另一个文件夹
	按钮	点击 OK 按钮, 创建一个文件并关闭该消息对话框 点击 Cancel 按钮, 关闭该消息对话框
!	消息	没有创建文件夹 (Not make folder)
	错误产生的原因	无法创建指定的文件夹
	用户采取的措施	指定另一个文件夹
	按钮	点击 OK 按钮, 关闭该消息对话框

附录 A 示例程序

本章描述的是 RA78K0 软件包附带的示例列表。

A.1 K0main.asm

```

NAME          SAMPM
;*****
;
;
;   HEX → ASCII 转换程序
;
;   主程序
;
;*****
PUBLIC        MAIN, START
EXTRN        CONVAH
EXTRN        @_STBEG

DATA         DSEG   saddr
HDTSA:      DS    1
STASC:      DS    2

CODE        CSEG   AT 0H
MAIN:      DW     START

START:     CSEG

;芯片初始化
MOVW      SP, #_@STBEG

MOV       HDTSA, #1AH
MOVW     HL, #HDTSA           ;将 2 位十六进制代码数据置于 HL 寄存器中

CALL     !CONVAH             ;将 16 进制数据转换为 ASCII 码
;将转换后的 ASCII 输出至 BC 寄存器中
;设置 DE, 指向保存 ASCII 码的表

MOVW     DE, #STASC
MOV      A, B
MOV      [DE], A
INCW     DE
MOV      A, C
MOV      [DE], A

BR       $$

END

```

A.2 K0ssub.asm

```

NAME          SAMPS
;*****
;
;
;      HEX → ASCII 转换程序
;      子程序
;
; 输入条件:(HL)      ← 2 位十六进制代码
; 输出条件:BC 寄存器 ← 2 位 ASCII 代码
;
;*****
PUBLIC        CONVAH

          CSEG
CONVAH:

          XOR     A, A
          ROL4   [HL]      ; 装入待转换的十六进制高位数码
          CALL   !SASC
          MOV    B, A      ; 存储转换结果

          XOR     A, A
          ROL4   A, [HL]   ; 装入待转换的十六进制低位数码

          CALL   !SASC
          MOV    C, A      ; 存储转换结果

          RET

;*****
;      子程序 ASCII 码转化
;
; 输入  Acc (低 4 位)   ← 十六进制码
; 输出  Acc             ← ASCII 码
;*****

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

          END

```

A.3 test1.s

```
EXTRN      SEARCH, STABLE
EXTRN      @_STBEG
PUBLIC     MAIN, START
;*****
; 字符串数据搜索
;*****
SDATA:
    DB      04,12H,34H,56H,78H
;
;
CODE       CSEG      AT 0H
START:     DW        MAIN

MAIN:
    MOVW    SP, @_STBEG
    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  表起始地址                *
          ; 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--
          endwhile
          POP    DE
          POP    HL
          A = [DE]
          E+ = A
          A=B
          ADDC    D, A
          endwhile
          SET1   CY
          RET
          END

```

A.5 testinc.s

```
PUBLIC      STABLE
;
; *****
; 数据表
; *****

        CSEG
STABLE:
        DB 03, 12H, 34H, 78H
        DB 04, 55H, 66H, 77H, 88H
        DB 05, 12H, 34H, 56H, 78H, 10H
        DB 03, 12H, 34H, 56H
        DB 04, 12H, 34H, 0AH, 78H
        DB 04, 12H, 34H, 56H, 70H
        DB 04, 12H, 34H, 56H, 78H
        DB 01, 0ABH
        DB 02, 34H, 78H
        DB 00
```

A.6 st.bat

```
echo off
cls
set     LEVEL=0

if "%1" == "" goto ERR_BAT

st78k0 -C%1 test1.s
RA78K0 test1.asm
if errorlevel 1 set LEVEL=1
st78k0s -C%1 test2.s
RA78K0 test2.asm
if errorlevel 1 set LEVEL=1
if %LEVEL% == 1 echo Assemble error !!
if %LEVEL% == 1 goto END

cls
lk78k0 test1.rel test2.rel -s -otest.lmf -ptest.map
if errorlevel 1 echo Link error !!
if errorlevel 1 goto END

cls
oc78k0 test
if errorlevel 1 echo Object conversion error !!
if errorlevel 1 goto END

cls
set LEVEL=0
lcnv78k0 -ltest.lmf -rtest1.rel test1.prn
if errorlevel 1 set LEVEL=1
lcnv78k0s -ltest.lmf -rtest2.rel test2.prn
if errorlevel 1 set LEVEL=1
if %LEVEL% == 1 echo List conversion error !!
if %LEVEL% == 1 goto END

cls
echo No error.
goto END

:ERR_BAT

    echo Usage : st.bat chiptype

:END
echo on
```

附录 B 使用注意事项列表

本章中描述在使用 RA78K0 时需要注意的内容:

(1) 设备文件的注意事项

要执行 RA78K0 必须要有设备文件的支持。而 RA78K0 软件包并不包含设备文件, 必需从另外的途径获取。从 NEC 电子微机的主页上下载相关的设备文件。

<http://www.necel.com/micro/ods/eng/tool/DeviceFile/list.html> → 开发工具下载 (ODS)

(2) 目标转换器的注意事项

指定选项-R (对象的地址源) 和-U (填充值说明) 来使用对象转换器。

这些选项是默认指定的。

如果需要 ROM 代码 (即工作中的“交叉处理”或“带输出 (tape out)”) 并且当目标的地址没有排序, 就会出现错误。因此, 务必要指定-R (不要取消指令)。

(3) 存储器初始化伪指令的注意事项

如果在数据段(DSEG)中出现存储器初始化伪指令 DW 或 DB, 就会输出目标代码, 但是目标转换器会输出警告消息 W4301。这是因为代码的地址不在 ROM 区域 (代码区域) 内。

如果在此状态下需要 ROM 代码 (即工作中的“交叉处理”或“带输出 (tape out)”), 就会出现错误。

(4) 目标转换器操作设定的注意事项

当使用目标转换器选项-U 指定起始地址时, 会从起始地址或代码的最低地址处开始填充。SFR 区(FF00H 至 FFFFH)不执行地址填充。

语法: -U 填充值 [, [起始], 大小]

[] 内的值可以省略。

(5) 存储器指令的注意事项

每个设备的默认存储区域名称不可擦除。

没有使用的默认存储区域名称的大小必须为零。

但是由于有些段定位至缺省区域, 所以在改变存储区域名称时务必多加留心。

需要了解缺省存储区域名称, 请参阅各个设备的**使用注意事项**。

(6) 调试操作的注意事项

当调试信息由 C 编译器/结构化汇编器预处理器输出时, 对输出汇编源程序进行汇编时, 请指定-NGA 选项以禁止输出调试信息。如果输出了调试信息, 那么就不能在 C 编译器/结构化汇编器中对源程序进行调试。

(7) CC78K0 相关的注意事项

当使用 CC78K0 输出的汇编代码来进行 C 源程序调试时，有几点必需注意。
需要了解详细信息，请参阅 **C 编译器软件包附带的文档 (使用注意事项)**。

(8) ID78K0/ID78K0-NS/ ID78K0-QB 及 SM78K0 的注意事项

当用 ID78K0/ID78K0-NS/ ID78K0-QB 或 SM78K0 进行调试时，要使用 ID78K0/ID78K0-NS/ ID78K0-QB 或 SM78K0 限制范围内的符号数量和行数。
需要了解详细信息，请参阅 **调试器/仿真器软件包附带的文档 (使用注意事项)**。

(9) 段名的注意事项

当描述段名称时，不要使用与源文件主程序名称相同的名字。否则，汇编时将会发生中止错误 F2106。

(10) 结构化汇编器源码中的宏定义描述

使用汇编语言来描述结构化汇编预处理器中的宏定义，而不要使用结构化汇编语言描述。否则，将出现重复标记定义错误。

(11) SFR 名及 EQU 定义的注意事项

虽然 SFR 名可以指定为伪指令 EQU 的操作数，但是当 saddrr 存储区外的 SFR 名指定为 PUBLIC 时，也会产生汇编错误。

(12) 当使用网络时的注意事项

如果用于创建临时文件的目录的位置在网络共享的文件系统中，可能会产生文件冲突，从而导致异常操作。
可以通过设置适当的选项及环境变量来避免这种情况发生。

汇编表					
ALNO	STNO	地址	目标 M 1	源	语句
1	1				
2	2	----		main_c	CSEG AT 100H
3	3				
4	4	00100	13F301		MOV BANK,#BANKNUM lab_bk1
5	5	00103	R9A0080		CALL !lab_bk1
6	6				
7	7				
8	8	----			CSEG BANK; 18000→0C000H
9	9	18000		label_bk1:	
10	10	18000	00		NOP
11	11	18001	00		NOP
12	12	18002	00		NOP
13	13	18003	AF		RET
14	14				
15	15				
16	16			END	

在以上程序中，lab_bk1 被指派到地址 18000H

表B-1 输出格式为“Intel扩展HEX”的示例（flash存储器实地址）

```

:100F000FFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFF10
:020000020000FC
:0601000013F3019A0080D8
:020000020000FC
;10010600FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF9
;10011600FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFE9
;10012600FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFD9
:
:
:
:10BFF000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF51
:020000020000FC
:04C00000000000AF8D
:020000020000FC
:10C00400FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF3C
:10C01400FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF2C

```

在 flash 存储器实地址的情况下，在地址 18000H 看到的代码在 0C000H 地址进行了描述，但是第 10 部分和后面的字节代码不发生变化。代码保持不变，而地址值会上升。（同时，每行最后字节的校验和也会变化。）用户无需为此变化有所顾虑。

表B-2 输出格式为“Intel扩展HEX”的示例（BANK数字+CPU地址）

```
:1000F000FFFFFFFFFFFFFFFFFFFFFFFFFFFF10
:020000020000FC
:0601000013F3019A0080DB
:020000020000FC
:10010600FFFFFFFFFFFFFFFFFFFFFFFFFFFF9
:10011600FFFFFFFFFFFFFFFFFFFFFFFFFFFFE9
:10012600FFFFFFFFFFFFFFFFFFFFFFFFFFFFD9
:
:
:
:10BFF000FFFFFFFFFFFFFFFFFFFFFFFFFFFF51
:020000021000EC
:04800000000000AFCD
:020000021000EC
:10800400FFFFFFFFFFFFFFFFFFFFFFFFFFFF7C
:10801400FFFFFFFFFFFFFFFFFFFFFFFFFFFF6C
```

地址 18000H 上的代码在 18000H 地址进行描述。

附录 C 选项列表

在本章节中，总结了程序选项并用表格形式列出。

在开发程序时，请参考这些程序选项。

该选项列表也可以作索引来用。

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	忽略指定时的解释
设备类型说明	-C 型设备	指定目标设备的类型	独立，和其它选项无关	不可省略
字符符号说明	-SC 字符	指定字符名称的最后一个字符	独立，和其它选项无关	-SCP
符号定义说明	-D 符号 [=数值] [, 符号名[= 数值]] ...] ... (可以指定两个或者 以上的路径名)	指定用于#IFDEF 指令的符号， 等	独立，和其它选项无关	无
制表符说明	-WT 数值，数值， 数值	指定转换指令的输出位置	独立，和其它选项无关	-WT2, 3, 4
包含文件路径说明	-I 路径名[, 路径名] ... (可以指定两个或者 以上的路径名)	由指定路径读取包含文件	独立，和其它选项无关	源文件所在路径。 由环境变量'INC78K0' 指定的路径
辅助源文件说明	-O [文件名]	指定辅助源文件名	独立，和其它选项无关	-O [输入文件名.ASM]
错误列表文件说明	-E [文件名]	输出错误列表文件	独立，和其它选项无关	-E [输入文件名.EST]

分类	格式	功能	与其它选项的关系	忽略指定时的解释
参数文件说明	-F 文件名	由指定文件传入输入文件名和选项	独立, 和其它选项无关	只能从命令行输入选项和文件名
调试信息输出说明	-GS	指定结构化汇编程序源代码级调试信息的输出	如果同时指定 -GS 选项和 -NGS 选项, 则最后指定的选项优先。	-GS
	-NGS	使选项 -GS 失效		
辅助源文件强制输出说明	-J	强制输出辅助源文件	独立, 和其它选项无关	禁止强制输出
汉字码 (双字节码) 说明	-ZS	在注释内描述的汉字码被认为是 shift JIS 码	如果同时指定 -ZS, -ZE, 及 -ZN 选项, 则最后指定选项优先。	在 Windows 中 /HP-UX: -ZS 在 SunOS, Solaris 中: -ZE
	-ZE	在注释内描述的汉字码被认为是 EUC 码		
	-ZN	在注释内描述的字符不被认为是汉字码		
设备文件搜索路径说明	-Y 路径名	由指定路径读取设备文件	独立, 和其它选项无关	<..ldev>用于指定启动 ST78K0 的路径
帮助说明	--	在显示器(控制台)上显示帮助信息.	所有其它选项均失效	不显示

C.2 汇编器选项

表C-2 结构化汇编器选项

分类	格式	功能	与其它选项的关系	忽略指定时的解释
设备类型说明	-C 型设备	指定目标设备的类型	独立, 和其它选项无关	不可省略
目标模块文件输出说明	-O [文件名]	指定目标模块文件的输出	如果同时指定 -O 选项及 -NO 选项, 则最后指定选项优先	-O [输入文件名.REL]
	-NO	指定不输出目标模块文件		
强制目标文件输出说明	-J	即使发生了严重错误, 目标模块文件仍然会输出	如果同时指定 -J 选项及 -NJ 选项, 则最后指定选项优先。	-NJ
	-NJ	使 -J 选项失效		
调试数据输出说明	-G	指定将要添加至目标文件中去的局部符号数据	如果同时指定 -G 选项及 -NG 选项, 则最后指定选项优先。	-G
	-NG	使 -G 选项失效		
	-GA	指定将源程序调试信息添加至目标文件中去	如果同时指定 -GA 选项及 -NGA 选项, 则最后指定选项优先。	-GA
	-NGA	使 -GA 选项失效		
包含文件读取路径说明	-I 路径名 [, 路径名] ... (可以指定两个或多个路径名)	规定从指定路径输入包含文件	独立, 和其它选项无关	源文件所在路径。 由环境变量 (INC78K0) 指定的路径
汇编列表文件输出说明	-P [文件名]	指定输出汇编列表文件。也指定输出文件的目录及文件名。	如果同时指定 -P 选项及 -NP 选项, 则最后指定选项优先。	-P [输入文件名.REL]
	-NP	使得 -P 选项失效		

分类	格式	功能	与其它选项的关系	忽略指定时的解释
汇编列表文件数据说明	-KA	将汇编列表输出至汇编列表文件中去.	如果同时指定-KS 选项及 -KX 选项, 则忽略-KS 选项。 如果同时指定-KA 选项及 -NKA 选项, 或-KS 选项及-NKS 选项, 或-KX 选项及-NKX 选项同时指定, 则最后指定选项优先。 如果同时指定了选项 -NKA, -NKS 及-NKX, 则 -P 选项失效.	-KA
	-NKA	使得 -KA 选项失效		-NKS
	-KS	将后随符号列表的汇编列表输出至汇编列表文件中去		-NKX
	-NKS	使-KS 选项失效.		
	-KX	将后随交叉参考列表的汇编列表输出至汇编列表文件中去		
	-NKX	使-KX 选项失效.		
汇编列表文件格式说明	-LW [字符数]	改变汇编列表文件中一行可打印的字符数目	如果指定 -NP 选项, 则-LW 选项失效	-LW132
	-LL [行数]	改变汇编列表文件中一页可打印的行数目.	如果指定-NP 选项, 则-LL 选项失效	-LL66
	-LH 字符串	指定汇编列表文件标题栏可打印的字符串.	如果指定-NP 选项, 则-LH 选项失效	无
	-LT [字符数]	指定标签页中的字符数	如果指定-NP 选项, 则-LT 选项失效	-LT8
	-LF	在汇编列表文件末端插入一换页 (FF)码	如果同时指定-LF 选项及 -NLF 选项, 则最后指定选项优先。	-NLF
	-NLF	使 -LF 选项失效	如果指定-NP 选项, 则-LF 选项失效。	
错误列表文件输出说明	-E [文件名]	输出错误列表文件.	如果同时指定-E 选项及 -NE 选项, 则最后指定选项优先。	-NE
	-NE	使 -E 选项失效		

分类	格式	功能	与其它选项的关系	忽略指定时的解释
参数文件说明	-F 文件名	由指定文件输入汇编选项和输入文件名。	独立, 和其它选项无关	只能由命令行指定输入选项和文件名
临时文件创建路径说明	-T 路径名	在指定路径创建临时文件	独立, 和其它选项无关	由环境变量 TMP 指定路径
汉字码说明	-ZS	在注释内描述的汉字码被认为是 shift JIS 码	如果同时指定 -ZS, -ZE, 及 -ZN 选项, 则最后指定选项优先。	在 Windows/HP-UX: -ZS 在 SunOS, Solaris 中: -ZE
	-ZE	在注释内描述的汉字码被认为是 EUC 码		
	-ZN	在注释内描述的字符不被认为是汉字码		
设备文件搜索路径说明	-Y 路径名	由指定路径读取设备文件	独立, 和其它选项无关	用于启动 RA78K0. 路径 <..\dev>
符号定义说明	-D 符号名 [=数值] [, 符号名 [=数值] ...] (可以指定两个或者以上的路径)	定义一个符号	独立, 和其它选项无关	无
系列通用目标说明	-COMMON	指定输出的目标模块文件可以适用于 78K0 系列。	独立, 和其它选项无关	无
自编程说明	-SELF	使用自编程时指定。	独立, 和其它选项无关	无
帮助说明	--	在显示器(控制台)上显示帮助信息。	所有其它选项均失效	不显示

C.3 连接器选项列表

表C-3 连接器选项列表

分类	格式	功能	与其它选项的关系	忽略指定时有 关说明
装载模块文件输出说明	-O [文件名]	输出装载模块文件	如果同时指定 -O 和 -NO 选项，则最后指定的选项优先	-O [输入文件名.LMF]
	-NO	不输出装载模块文件		
强制装载模块文件输出说明	-J	强制输出装载模块文件	如果同时指定 -J 和 -NJ 选项，则最后指定的选项优先	-NJ
	-NJ	使选项 -J 失效		
调试数据输出说明	-G	向装载模块文件输出调试信息	如果同时指定 -G 和 -NG 选项，则最后指定的选项优先。 当指定了选项-NG 时，不论是否指定了-KP 或 -KL 选项，公共符号列表及局部符号列表均不能输出。	-G
	-NG	使选项-G 失效		
栈判定符号的产生说明	-S [区域名称]	自动生成栈判定公用符号	如果同时指定 -S 和 -NS 选项，则最后指定的选项优先	-NS
	-NS	使选项-S 失效		
指令文件说明	-D 文件名	将指定文件输入为指令文件	独立，和其它选项无关	无
连接列表文件输出说明	-P [文件名]	指定输出连接列表文件	如果同时指定 -P 和 -NP 选项，则最后指定的选项优先	-P [文件名.MAP]
	-NP	使选项-P 失效		

分类	格式	功能	与其它选项的关系	忽略指定时有 关说明
连接列表文件数据说明	-KM	将映像列表输出至连接列表文件中	如果同时指定 -KM 和 -NKM 选项，则最后指定的选项优先。	-KM
	-NKM	使选项 -KM 失效。	如果选项 -NKM, -NKP 及 -NKL 同时指定，那么即使指定了选项 -P 连接列表文件也无法输出。	
	-KD	将链接指令文件输出至连接列表文件中	如果指定了 -NKM 选项，则选项 -KD 失效。	-KD
	-NKD	使选项 -KD 失效。	如果同时指定 -KD 和 -NKD, 或 -KP 和 -NKP, 或 -KL 和 -NKL 选项，则最后指定的选项优先。	-NKP
	-KP	将公共符号列表输出至连接列表文件中	如果指定了 -NG 选项，这时，即使指定了 -KP 或 -KL 选项，公共符号列表及局部符号列表均不能输出。	
	-NKP	使选项 -KP 失效。		
	-KL	将局部符号列表输出至连接列表文件中		-NKL
连接列表文件格式说明	-LL [行数]	指定连接列表文件中一页可打印的行数	如果指定 -NP 选项，则选项 -LL 失效	-LL66
	-LF	在连接列表文件末插入换页(FF)码。	如果同时指定 -LF 和 -NLF 选项，则最后指定的选项优先。	-NLF
	-NLF	使选项 -LF 失效。	如果指定 -NP 选项，则最后指定选项优先。	
错误列表文件输出说明	-E [文件名]	输出错误列表文件	如果同时指定 -E 和 -NE 选项，则最后指定的选项优先。	-NE
	-NE	使选项 -E 失效		
库文件说明	-B 文件名	将特定文件作为库文件输入	独立，和其它选项无关	无

分类	格式	功能	与其它选项的关系	忽略指定时有 关说明
库文件读取路径 说明	-I 路径名 [, 路径 名] ... (可以指定两个或多 个路径名)	由指定路径读取库文件	如果由选项-B 指定了某个 没有路径名的库文件, 则 选项-I 失效	由环境变量 'LIB78K0' 指定 路径
参数文件说明	-F 文件名	由指定文件输入连接器选项和输 入文件名.	独立, 和其它选项无关	只有在命令 行中输入该选 项和输入文件名
临时文件创建路 径说明	-T 路径名	在指定路径创建临时文件	独立, 和其它选项无关	由环境变 量 TMP 指定路径
设备文件搜索路 径说明	-Y 路径名	由指定路径读取设备文件	独立, 和其它选项无关	LK78K0 启动路 径<..\dev>
警告信息输出说 明	-W [level]	指定是否向控制台输出警告信息	独立, 和其它选项无关	一般错误信 息 输出
flashROM 模型的 导入区域 ROM 的连接说明	-ZB	指定 flashROM 区域的第一个地 址	独立, 和其它选项无关	无
指定芯片内调试 程序大小	-GO [大小]	指定芯片内调试程序大小	独立, 和其它选项无关	无
指定安全 ID	-GI 安全 ID	指定安全 ID	独立, 和其它选项无关	无
帮助说明	--	在显示器(控制台)上显示帮助信 息。	所有其它选项均失效	不显示

C.4 目标转换器选项

表C-4 目标转换器选项

分类	格式	功能	与其它选项的关系	忽略指定时有 关说明
十六进制格式目 标文件输出说明	-O [文件名]	输出 hex 格式目标文件	如果同时指定 -O 和 -NO 选项，则最后指定的选项 优先。	-O [输入文件 名].HEX(文件类 型 H1 至 H15 用于扩展空间)
	-NO	不输出 hex 格式目标文件		
符号表文件输出 说明	-S [文件名]	输出符号表文件	如果同时指定 -S 和 -NS 选 项，则最后指定的选项优 先。	-S [输入文件 名].SYM(文件 类型 S1 至 S15 用于扩展空间)
	-NS	不输出符号表文件		
以目标地址次序 进行排序说明	-R	以地址次序对十六进制格式目标 进行排序	如果同时指定 -R 和 -NR 选项，则最后指定的选项 优先。 如果指定 -NO 选项，则选 项 -R 失效。	-R
	-NR	使选项 -R 失效。		
目标补充说明	-U 补充值 [, [起始], 大小]	将指定的补充值作为目标代码输 出到没有十六进制格式目标的 地址区中。	如果指定 -NO 选项，则选 项 -U 失效。	-UOFFH
错误列表文件输 出说明	-E [文件名]	输出错误列表文件。	如果同时指定 -E 和 -NE 选 项，则最后指定的选项优 先。	-NE
	-NE	使选项 -E 失效。		
参数文件说明	-F 文件名	由指定文件输入选项和文件名	独立，和其它选项无关	只有在命令行 中指定选项和 输入文件名
设备文件搜索路 径说明	-Y 路径名	由指定路径读取设备文件	独立，和其它选项无关	<..dev> 存放 OC78K0 启动 路径

分类	格式	功能	与其它选项的关系	忽略指定时有 关说明
flashROM 模型文件单独输出的说明	-ZF	在连接 flash 存储器模型的引导区域 ROM 程序时，添加一个能单独输出引导区域和其他区域的选项，以便生成不同的 HEX 格式文件。	独立，和其它选项无关	无
帮助说明	--	在显示器(控制台)上显示帮助信息。	所有其它选项均失效	不显示

C.5 库管理程序选项

表C-5 库管理选项列表

分类	格式	功能	与其它选项之间的关系	忽略指定时有 关说明
列表文件格式说明	-LW [字符数]	改变列表文件中一行可打印的字符数目	如果没有指定 LIST 子命令, 该选项失效。	-LW132
	-LL [行数]	改变列表文件中一页可打印的行数目.		-LL66
	-LF	在列表文件未插入换页(FF)码。	如果同时指定 -LF 和 -NLF 选项, 则最后指定的选项优先。	-NLF
	-NFL	使选项 -LF 失效.		
临时文件创建路径说明	-T 路径名	在指定路径创建临时文件	独立, 和其它选项无关	由环境变量 TMP 指定路径
设备文件搜索路径说明	-Y 路径名	由指定路径读取设备文件	独立, 和其它选项无关	<..dev>(LB78K 0 启动路径)
帮助说明	--	在显示器(控制台)上显示帮助信息。	所有其它选项均失效	不显示

C.6 表转换程序选项

表C-6 表转换程序选项

分类	格式	功能	与其它选项之间的关系	忽略指定时有 关说明
目标文件输入说明	-R [文件名]	输入目标文件	独立, 和其它选项无关	-R [汇编列表文件名.REL]
装载模块文件输入说明	-L [文件名]	输入装载模块文件	独立, 和其它选项无关	-L [汇编列表文件名.LMF]
绝对汇编列表文件输出说明	-O [文件名]	输出绝对汇编列表文件	独立, 和其它选项无关	-O [汇编列表文件名.P]
错误列表文件输出说明	-E [文件名]	输出错误列表文件.	如果同时指定 -E 和 -NE 选项, 则最后指定的选项优先	-NE
	-NE	使选项 -E 失效.		
参数文件说明	-F 文件名	由指定文件输入选项和输入文件名.	独立, 和其它选项无关	只有在命令行中指定选项和输入文件名
帮助说明	--	在显示器(控制台)上显示帮助信息。	所有其它选项均失效	不显示

附录 D 子命令列表

本附录以列表形式对子命令进行了总结。
在开发软件程序时，参考本列表有助于程序开发。
该子命令列表也可以作索引之用。

表D-1 子命令列表

分类	格式	功能	缩略格式
CREATE	CREATE Δ 库文件名 [Δ 处理]	创建一个新的库文件	C
ADD	ADD Δ 库文件名 Δ 处理	向库文件添加一个模块。	A
DELETE	DELETE Δ 库文件名 Δ (Δ 模块名 [Δ , ...] Δ)	从库文件中删除一个模块。	D
REPLACE	REPLACE Δ 库文件名 Δ 处理	将库文件中模块用另一个模块替代。	R
PICK	PICK Δ 库文件名 Δ (Δ 模块名 [Δ , ...] Δ)	由现存的库文件中拾取指定模块。	P
LIST	LIST[Δ 选项] Δ 库文件名 [Δ (Δ 模块名 [Δ , ...] Δ)]	向库文件中模块输出数据。	L
HELP	HELP	在显示器(控制台)上显示帮助信息。	H
EXIT	EXIT	退出库管理程序	E

附录 E 索引

[Symbol]

-- (LB78K0).....	255
-- (LCNV78K0)	290
-- (LK78K0).....	192
-- (OC78K0).....	236
-- (RA78K0)	138
-- (ST78K0)	86

[A]

异常中止错误.....	322
绝对汇编列表.....	277, 313
ADD	259
.ASM	67, 93
汇编列表.....	277, 298, 318
汇编器	17, 27
AT	150

[B]

-B (LK78K0)	181
-------------------	-----

[C]

-C (RA78K0).....	104
-C (ST78K0).....	74
-COMMON (RA78K0).....	136
COMPLETE	150
CREATE.....	258
交叉参考列表.....	301

[D]

-D (LK78K0)	167
-D (RA78K0).....	135
-D (ST78K0).....	76
DELETE	260
.DR (LK78K0).....	146

[E]

-E (LCNV78K0)	287
-E (LK78K0)	180
-E (OC78K0)	230
-E (RA78K0).....	128
-E (ST78K0)	80
.ELK	146
.ELV	275
环境变量.....	44,316
.EOC	201
.ERA.....	93
错误列表.....	295,303,310,311,313
.EST	67
执行过程.....	50,56
EXIT	267

[F]

-F (LCNV78K0).....	288
-F (LK78K0).....	183
-F (OC78K0).....	231
-F (RA78K0)	130
-F (ST78K0).....	81
致命错误.....	322

[G]

-G (LK78K0)	164
-G (RA78K0).....	107
-GA (RA78K0)	109
-GI (LK78K0).....	190
-GO (LK78K0).....	189
-GS (ST78K0).....	82

[H]

帮助	266
.HEX.....	201

[I]

-I (LK78K0)	182
-I (RA78K0)	111
-I (ST78K0).....	78
INC78K0.....	316
安装.....	38
Intel 标准 HEX 格式.....	205
内部错误.....	322

[J]

-J (LK78K0)	163
-J (RA78K0).....	106
-J (ST78K0).....	83

[K]

-KA (RA78K0).....	113
Kanji 码.....	45
-KD (LK78K0)	171
-KI (OC78K0).....	233
-KIE (OC78K0)	233
-KL (LK78K0).....	175
-KM (LK78K0).....	169
-KM (OC78K0).....	233
-KME (LK78K0)	233
-KP (LK78K0)	173
-KS (RA78K0).....	115
-KT (OC78K0)	233
-KX (RA78K0).....	116

[L]

-L (LCNV78K0).....	285
LANG78K	316

- LF (LB78K0) 252
- LF (LK78K0) 179
- LF (RA78K0)..... 127
- LH (RA78K0) 122
- .LIB..... 146,241
- LIB78K0 316
- 库 30, 241
- 连接指令..... 149
- 连接列表文件 304
- 连接器 28
- LIST 264
- List 转换程序 31, 275
- LL (LB78K0)..... 251
- LL (LK78K0)..... 177
- LL (RA78K0) 120
- .LMF..... 146, 201, 275
- 装载模块文件 146,201,275
- 载入符号列表 309
- .LST 241
- LT (RA78K0)..... 125
- LW (LB78K0) 250
- LW (RA78K0)..... 118
- [M]**
- .MAP 146
- 映像列表..... 306
- 性能最大化 33
- MEMORY 150
- 存储程序区 148
- 存储程序指令 151
- 存储程序空间 148
- MERGE 150
- [N]**
- NE (LCNV78K0) 287
- NE (LK78K0)..... 180
- NE (OC78K0)..... 230
- NE (RA78K0) 128
- NG (LK78K0) 164
- NG (RA78K0)..... 107
- NGA (RA78K0) 109
- NGS (ST78K0)..... 82
- NJ (LK78K0) 163
- NJ (RA78K0)..... 106
- NKA (RA78K0)..... 113
- NKD (LK78K0) 171
- NKL (LK78K0)..... 175
- NKM (LK78K0)..... 169
- NKP (LK78K0) 173
- NKS (RA78K0)..... 115
- NKX (RA78K0)..... 116
- NLF (LB78K0)..... 252
- NLF (LK78K0)..... 179
- NLF (RA78K0) 127
- NO (LK78K0) 162
- NO (OC78K0) 223
- NO (RA78K0)..... 105
- NP (LK78K0) 168
- NP (RA78K0) 112
- NR (OC78K0)..... 227
- NS (LK78K0) 165
- NS (OC78K0)..... 225
- NU (OC78K0)..... 228
- [O]**
- O (LCNV78K0)..... 286
- O (LK78K0) 162
- O (OC78K0) 223
- O (RA78K0) 105
- O (ST78K0)..... 79
- 目标转换器 29, 201
- [P]**
- .P 275
- P (LK78K0) 168
- P (RA78K0)..... 112
- 参数文件... 67, 69, 93, 97, 146, 156, 201, 218,275,280
- PATH..... 316
- PICK 263
- .PLK 146
- .PLV 275
- PM +..... 88, 140, 194, 237, 268, 291, 319, 361
- .POC 201
- .PRA 93
- .PRN..... 93, 275
- .PST 67
- 公用符号列表..... 308
- [R]**
- R (LCNV78K0) 284
- R (OC78K0) 227
- RAM 148
- REGULAR 152, 153
- .REL 93,146,241,275
- REPLACE 261
- ROM..... 148
- [S]**
- S (LK78K0) 165
- S (OC78K0) 225
- 示例程序..... 373
- SC (ST78K0)..... 75
- 段定位指令 153
- SELF (RA78K0) 137
- SEQUENT 150
- 结构化汇编程序 26
- .SYM 201
- [T]**
- T (LB78K0)..... 253
- T (LK78K0)..... 185

-T (RA78K0)	132	-Y (OC78K0).....	234
TMP	316	-Y (RA78K0)	134
[U]		-Y (ST78K0)	85
-U (OC78K0)	228	[Z]	
[W]		-ZE (RA78K0).....	188
-W (LK78K0)	187	-ZE (ST78K0)	133
警告错误.....	322	-ZN (RA78K0).....	84
-WT (ST78K0).....	77	-ZN (ST78K0)	235
[Y]		-ZS (RA78K0).....	133
-Y (LB78K0)	254	-ZS (ST78K0)	84
-Y (LK78K0)	186		