

お客様各位

---

## カタログ等資料中の旧社名の扱いについて

---

2010年4月1日を以ってNECエレクトロニクス株式会社及び株式会社ルネサステクノロジが合併し、両社の全ての事業が当社に承継されております。従いまして、本資料中には旧社名での表記が残っておりますが、当社の資料として有効ですので、ご理解の程宜しくお願ひ申し上げます。

ルネサスエレクトロニクス ホームページ (<http://www.renesas.com>)

2010年4月1日  
ルネサスエレクトロニクス株式会社

【発行】ルネサスエレクトロニクス株式会社 (<http://www.renesas.com>)

【問い合わせ先】<http://japan.renesas.com/inquiry>

## ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
3. 当社製品を改造、改変、複製等しないでください。
4. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器の設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途の目的で使用しないでください。また、当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器に使用することができません。
6. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。  
標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット  
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）  
特定水準： 航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害などを生じさせないようお客様の責任において冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、詳細につきましては製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関し、当社は、一切その責任を負いません。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

注 1. 本資料において使用されている「当社」とは、ルネサスエレクトロニクス株式会社およびルネサスエレクトロニクス株式会社とその総株主の議決権の過半数を直接または間接に保有する会社をいいます。

注 2. 本資料において使用されている「当社製品」とは、注 1 において定義された当社の開発、製造製品をいいます。

ユーザーズ・マニユア

RENESAS

LK17K

リンカ

---

資料番号 U12518JJ2V0UM00 (第2版)  
(旧資料番号 EEU-5005)  
発行年月 April 1997 N

© NEC Corporation 1995

**NEC**

ユーザース・マニュアル

**LK17K**

リンカ

---

資料番号 U12518JJ2V0UM00 (第2版)  
(旧資料番号 EEU-5005)  
発行年月 April 1997 N

© NEC Corporation 1995

{x 屯}

## 目 次 要 約

### 第Ⅰ編 言語編

概 要 … 3

機 能 … 7

### 第Ⅱ編 操作編

概 要 … 31

システム構成 … 33

操作方法 … 43

エラー・メッセージ … 81

*emlC-17K, SIMPLEHOST* は、日本電気株式会社の登録商標です。

MS-DOSおよびWindowsは、米国Microsoft Corporationの米国およびその他の国における登録商標または商標です。

PC DOS, PC/ATは、米国IBM社の商標です。

- 本資料の内容は、後日変更する場合があります。
- 文書による当社の承諾なしに本資料の転載複製を禁じます。
- 本資料に記載された製品の使用もしくは本資料に記載の情報の使用に際して、当社は当社もしくは第三者の知的所有権その他の権利に対する保証または実施権の許諾を行うものではありません。上記使用に起因する第三者所有の権利にかかわる問題が発生した場合、当社はその責を負うものではありませんのでご了承ください。



## 本版で改訂された主な箇所

箇所	内容
はじめに	関連資料 追加
p.21	図 2-13 セクションの配置例 変更
p.22	表 2-2 再配置の優先順位 変更
p.33	2.1 システム環境 変更
p.35	2.2 ファイル構成 追加
p.41	2.5 シンボル数 変更
p.102	エラー・メッセージ(A406) 追加

本文欄外の★印は、本版で改訂された主な箇所を示しています。

巻末にアンケート・コーナーを設けております。このドキュメントに対するご意見をお気軽にお寄せください。

# はじめに

- 製品概要** LK17Kは、RA17KとemiC-17K\*が作成したファイルをリンクする、リロケータブル・アセンブラ・パッケージのリンクです。
- 対象者** このマニュアルは、4ビット・シングルチップ・マイクロコントローラ 17Kシリーズを採用し、RA17KまたはemiC-17Kを使ってその応用システムを設計、開発するエンジニアを対象としています。
- 構成** このマニュアルは、大きく分けて次の内容で構成しています。

言語編

操作編

エラー・メッセージ

- 読み方** このマニュアルを読むにあたっては、RA17KまたはemiC-17Kについての知識が必要です。

- 凡例** このマニュアルでは、次のような記号、表記方法を使用しています。

… : 同一形式を繰り返します。

[ ] : 大かっこの中は省略できます。

< > : < > で囲まれた文字そのもの、特にタイトルを表します。

××× : 任意の文字列を示します。

△ : スペースを示します。

数の表記 : 2進数…××××B

10進数…×××または××D

16進数…×××H

## ファイル名の規則

[ドライブ名:] [¥ディレクトリ名¥…] ファイル名 [. 拡張子]

ファイル名には、ドライブ名、ディレクトリ名を含みます。パス名は、ドライブ名、ディレクトリ名だけの場合に使用します。

★ 関連資料 関連資料は暫定版の場合がありますが、この資料では「暫定」の表示をしておりません。あらかじめご了承ください。

・ソフトウェア・ツールに関する資料

RA17K ユーザーズ・マニュアル	: U10305J
MAKE/CNV17K ユーザーズ・マニュアル	: U10596J
LK17K ユーザーズ・マニュアル	: このマニュアル
DOC17K ユーザーズ・マニュアル	: EEU-5006
emlC-17K バージョン1.00 ユーザーズ・マニュアル	: EEU-876
プロジェクト・マネージャ バージョン1.00 ユーザーズ・マニュアル	: EEU-989

・SIMPLEHOST®に関する資料

SIMPLEHOST バージョン2.0 emlC-17K/RA17K対応版 ユーザーズ・マニュアル	
入門編	: EEU-5009
レファレンス編	: EEU-5007

ソフトウェア・パッケージには、README.DOCファイルが含まれている場合があります。  
README.DOCファイルには、マニュアルには掲載されていない最新情報が含まれていますので、  
実際にご使用になる前に必ず目を通してください。

# 目 次

## 第I編 言語編 … 1

### 第1章 概 要 … 3

- 1.1 機能概要 … 3
- 1.2 17Kシリーズのアーキテクチャ … 3
  - 1.2.1 プログラム・メモリ … 3
  - 1.2.2 セグメントの構成 … 4
  - 1.2.3 EPA領域 … 5

### 第2章 機 能 … 7

- 2.1 ディレクティブ … 7
  - 2.1.1 概 要 … 7
  - 2.1.2 ディレクティブ・ファイル … 7
  - 2.1.3 数 値 … 8
  - 2.1.4 コメント文 … 8
  - 2.1.5 セクション配置ディレクティブ (マージ・ディレクティブ) … 8
- 2.2 入力セクションの結合 (マージ) … 16
  - 2.2.1 セクションの結合型 … 16
  - 2.2.2 結合型の決定 … 16
  - 2.2.3 結合型ごとのマージ方法 … 16
- 2.3 セクションの再配置属性 … 18
  - 2.3.1 再配置属性の決定 … 18
  - 2.3.2 セクション配置アドレスの決定 … 19
  - 2.3.3 セクションの配置 … 20
- 2.4 シンボル値の決定と出力 … 23
  - 2.4.1 外部定義シンボル/外部参照シンボルの参照関係のチェック … 23
- 2.5 プログラム・メモリ・オーバフロー時のエラー・メッセージ … 24
- 2.6 インクリメンタル・ロード用差分ファイル … 26
  - 2.6.1 差分ファイルの作成 … 26

## 第II編 操 作 編 … 29

### 第1章 概 要 … 31

### 第2章 システム構成 … 33

- 2.1 システム環境 … 33
  - 2.1.1 ハードウェア環境 … 33
  - 2.1.2 ソフトウェア環境 … 34
- 2.2 ファイル構成 … 35
- 2.3 入出力ファイル … 36

- 2.3.1 入出力ファイル一覧 … 36
- 2.3.2 出力先 … 36
- 2.3.3 入力ファイルの指定 … 37
- 2.3.4 出力ファイルの指定 … 38
- 2.3.5 出力ファイル名の解釈 … 38
- 2.3.6 拡張子の省略時解釈 … 39
- 2.4 テンポラリ・ファイル … 40
- 2.5 シンボル数 … 41
- 2.6 環境変数 … 41
- 2.7 処理の中断 … 41

### 第3章 操作方法 … 43

- 3.1 起 動 … 43
  - 3.1.1 コマンド・ライン上にオプションを指定する場合 … 43
  - 3.1.2 パラメータ・ファイルを使用する場合 … 43
  - 3.1.3 実行開始メッセージ … 44
  - 3.1.4 ヘルプ・メッセージ … 45
  - 3.1.5 プログラムの終了 … 46
  - 3.1.6 エラー・レベル … 47
- 3.2 入 力 … 48
  - 3.2.1 オブジェクト・モジュール・ファイル … 48
  - 3.2.2 リンク・オプション … 48
  - 3.2.3 リンク・オプションの種類 … 49
- 3.3 出 力 … 62
  - 3.3.1 出力メッセージ … 62
  - 3.3.2 出力ファイル … 64

### 第4章 エラー・メッセージ … 81

# 図の目次

図番号

タイトル、ページ

---

## 第I編 言語編

- 1-1 メモリ領域（最大構成の場合） … 4
- 1-2 17Kのセグメント化アドレス空間例（16Kステップ=32Kバイトの場合） … 5
- 1-3 17Kシリーズのセグメント内アドレス空間 … 6
  
- 2-1 再配置属性（AT）とその配置方法 … 9
- 2-2 再配置属性（BOOT）とその配置方法 … 10
- 2-3 再配置属性（CROM）とその配置方法 … 10
- 2-4 再配置属性（VECTn）とその配置方法 … 10
- 2-5 再配置属性（DSYS）とその配置方法 … 11
- 2-6 再配置属性（SSYS）とその配置方法 … 11
- 2-7 再配置属性（DSBR）とその配置方法 … 12
- 2-8 再配置属性（SBR）とその配置方法 … 12
- 2-9 再配置属性（DVECTn）とその配置方法 … 13
- 2-10 統合型がSEQUENTのときのマージ方法 … 16
- 2-11 不正な配置の例 … 19
- 2-12 EPA領域へのセクションの配置 … 20
- 2-13 セクションの配置例 … 21
- 2-14 17Kシリーズのセグメント内アドレス空間 … 25

## 第II編 操作編

- 1-1 処理概要 … 31
  
- 3-1 .ICEファイルのコード出力形式 … 67
- 3-2 PROMファイルのコード出力形式 … 69
- 3-3 差分ファイルのコード出力形式 … 71
- 3-4 リンク・マップ・ファイルの全体図 … 73

# 表の目次

表番号

タイトル, ページ

---

## 第I編 言語編

- 1-1 17Kシリーズのプログラム・メモリ (最大構成の場合) … 3
- 2-1 再配置属性によるセクションへの分岐命令生成の有無 … 18
- 2-2 再配置の優先順位 … 22

## 第II編 操作編

- 2-1 入出力ファイル一覧 … 36
- 2-2 出力デバイス … 37
- 2-3 出力ファイル名の解釈 … 39
- 2-4 出力ファイル名の指定の例 … 39
- 2-5 拡張子の省略時解釈 … 39
- 2-6 テンポラリ・ファイル一覧 … 40
  
- 3-1 エラー・レベル … 47
- 3-2 リンク・オプション一覧 … 49
- 3-3 リンク・マップ・ファイル … 72
- 3-4 ヘッダ部の内容 … 74
- 3-5 ID部の内容 … 75
- 3-6 メモリ・マップの内容 … 76
- 3-7 セクション・タイプ表示方法 … 77
- 3-8 セクション名表示方法 … 77
- 3-9 配置領域表示方法 … 77
- 3-10 ローカル・シンボル・リストの内容 … 77
- 3-11 シンボル属性表示方法 … 78
- 3-12 パブリック・シンボル・リストの内容 … 78
- 3-13 シンボル属性表示方法 … 79
- 3-14 アロケートできなかったセクション・リストの内容 … 79

# 第 I 編

## 言語編



[× 毛]

# 第1章 概要

## 1.1 機能概要

LK17Kは、リロケータブル・アセンブラ (RA17K) またはコンパイラ (*emlC-17K*) が出力したオブジェクト・モジュール・ファイル (.REL) を読み込み、次のファイルを出力します。

- ・リンク・オブジェクト・モジュール・ファイル (.LNK)
- ・ロード・モジュール・ファイル (.ICE/.PRO)
- ・差分ファイル (.DIF)
- ・リンク・マップ・ファイル (.LMP)

LK17Kが出力したリンク・オブジェクト・モジュール・ファイル (.LNK) を再びLK17Kの入力ファイルとして使うことはできません。

## 1.2 17Kシリーズのアーキテクチャ

### 1.2.1 プログラム・メモリ

プログラム・メモリ領域は、8Kステップ単位のセグメントという単位で管理されており、セグメントには名前 (SEG0-SEG7) が定義されています。17Kシリーズでは最大8つのセグメントを実装できます。デバイスに実装しているセグメントの数は、各デバイスにより異なります。LK17Kは、実装しているセグメント数の情報をデバイス・ファイルから得ます。

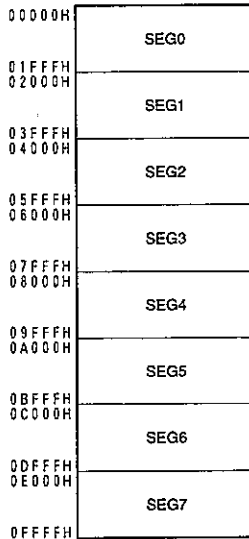
17Kでサポートするプログラム・メモリ領域は、次のように定義されています。

表1-1 17Kシリーズのプログラム・メモリ (最大構成の場合)

メモリ領域	スタート・アドレス	セグメント・サイズ
SEG0 (セグメント0)	00000H	2000H (8 Kステップ)
SEG1 (セグメント1)	02000H	
SEG2 (セグメント2)	04000H	
SEG3 (セグメント3)	06000H	
SEG4 (セグメント4)	08000H	
SEG5 (セグメント5)	0A000H	
SEG6 (セグメント6)	0C000H	
SEG7 (セグメント7)	0E000H	

LK17Kは、セクションを最小単位としてリンク処理 (再配置) を行います。セクションとは、RA17KのCSEG疑似命令で定義された領域です。セクションは、デバイスに実装されているメモリ領域に配置されます。デバイスに実装されていないアドレスにはセクションを配置できません。

図1-1 メモリ領域 (最大構成の場合)



PROGRAM空間 (64Kステップ = 128 Kバイト)

### 1.2.2 セグメントの構成

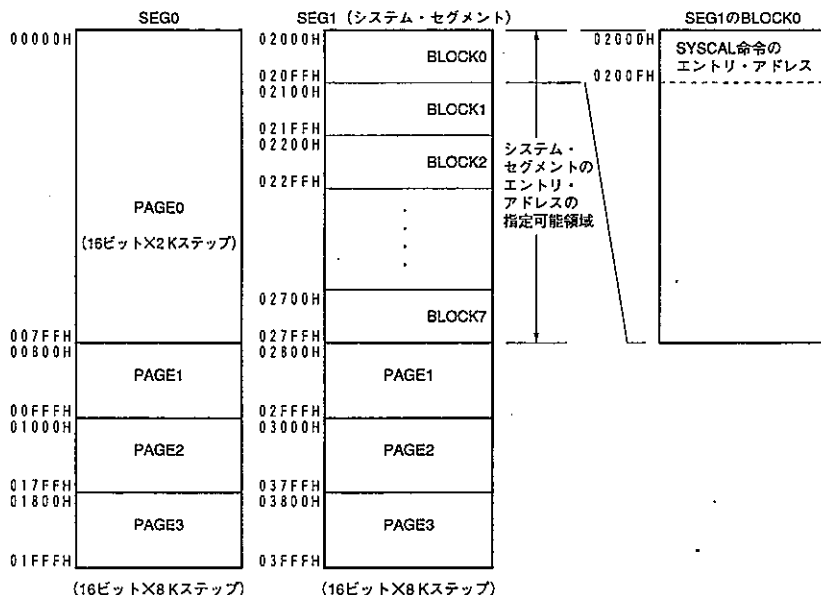
各セグメントは、2Kステップ単位のページ (Page) で構成されます。また複数のセグメントを持つデバイスでは、最終セグメントをSYSCAL命令で分岐できるセグメントとして扱える場合があります。この最終セグメントをシステム・セグメントと呼びます。システム・セグメントのページ0は、さらに256ステップ単位のブロック (BLOCK0-BLOCK7) に分割されます。

17Kシリーズでは、分岐命令として直接分岐命令と間接分岐命令の2種類を用意しています。このうち直接分岐命令 (BR addr命令, CALL addr命令) ではセグメント境界を越える分岐はできません。これは、プログラム・カウンタのビット長が最大13ビット (8Kステップ) のためです。セグメントは、プログラム・カウンタとは別に用意されたセグメント・レジスタ (SGR) の値で指定されます。

セグメント境界を越えて分岐する場合は、システム・レジスタの1つであるアドレス・レジスタ (AR) を使用した間接分岐命令 (BR @AR命令, CALL @AR命令) か、SYSCAL命令で行います。SYSCAL命令は、1命令でセグメントを越えてシステム・セグメントに分岐できます。システム・セグメントの各ブロックの先頭16ワードがSYSCAL命令のエントリ・アドレスとなります。

なお、CALL addr命令のオペランドには11ビット分のアドレスしか指定できないため、CALL addr命令の分岐先はPage0内に限られます。

図1-2 17Kのセグメント化アドレス空間例 (16Kステップ=32Kバイトの場合)



### 1.2.3 EPA領域

プログラムのディバグ時には、一時的にプログラム・サイズが製品のROM容量を越える場合があります。しかし、その場合にROMを越えた部分のディバグができないと不便です。17Kシリーズのインサーキット・エミュレータIE-17Kでは、ディバグ時に本チップのプログラム・カウンタを使用しているため、製品のROM容量を越えた場合にプログラム・カウンタのビット長が不足してしまい、超過部分のプログラムを制御できなくなります。

IE-17Kでは、この問題を解決するためにEPA (Extra Program Address) ビットを導入し、少なくとも本チップのROM容量の2倍のプログラムまで制御できるようにしています。

つまり、本チップのプログラム・カウンタから送られるアドレスにEPAを1ビット付加し、等価的にプログラム・カウンタを1ビット増やして制御可能なアドレス空間を本チップの2倍にする機能です。したがってROM容量を越えた場合も、IE-17K上でディバグできます。

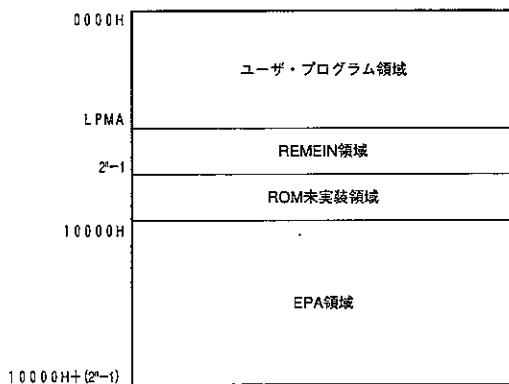
EPAビットは、製品のROMアドレスの最上位ビットよりさらに上位に1ビットのアドレスとして付加されるため、エミュレートできるROM容量は製品の2倍になります。EPAビットが1であるようなアドレス空間をEPA領域と呼び、パッチ領域 (Patch Area) として使用します。

なお、製品の最終ROMアドレス (LPMA) が $2^n-1$ に満たない場合、LPMAを越えて $2^n-1$ までのアドレス空間は、EPAビットなしで扱えるエミュレーション領域となります。この領域をREMAIN領域と呼び、EPA

領域と同様にバッチ領域として使用します。

プログラムがROMの実効アドレスを越えた場合、LK17Kはエラーを発生します。通常のエラーと異なるのは、不正であるにもかかわらず正常なオブジェクトを生成することです。ROMコード発注時は、このエラーが出力されなくなるまでプログラム・サイズを縮めてください。

図1-3 17Kシリーズのセグメント内アドレス空間



## 第2章 機能

この章では、LK17Kの機能について説明します。

### 2.1 ディレクティブ

#### 2.1.1 概要

LK17Kは、セクションを最小単位としてリンク処理（再配置）します。セクションはRA17KのCSEG疑似命令で定義された領域です。

ディレクティブは、LK17Kに対して入力ファイルや使用可能なメモリ領域、セクションの配置など、リンク時の各種指示を行うための命令群です。ディレクティブは、以下のような手順で使用します。

- ① ユーザは、ディレクティブを次に述べるディレクティブ・ファイルに記述します。
- ② LK17K起動時にリンカ・オプション（-DIR）でディレクティブ・ファイル名を指定します。

LK17Kはディレクティブ・ファイルを読み込み、その中に記述されたディレクティブを解釈しながらリンク処理を行います。

ディレクティブには、セクションの配置アドレスを指定するセクション配置ディレクティブがあります。

#### 2.1.2 ディレクティブ・ファイル

ディレクティブ・ファイルとは、LK17Kにセクションの配置を指定するためにディレクティブを記述したテキスト・ファイルです。ディレクティブの記述フォーマットについては、2.1.5 セクション配置ディレクティブ（マージ・ディレクティブ）を参照してください。

ディレクティブ・ファイル中では、次の20個の単語は予約語になっています。

```
MERGE  
AT DSBR DSYS SBR SSYS BOOT VECTn DVECTn CROM  
SEQUENT COMPLETE  
SEG0 SEG1 SEG2 SEG3 SEG4 SEG5 SEG6 SEG7
```

ディレクティブ・ファイル中で予約語をほかの意味（セクション名など）に使用することはできません。予約語は大文字でも小文字でもかまいません。ただし、大文字と小文字を混在させないでください。LK17Kに、ディレクティブ・ファイルは1つだけ指定できます。2つ以上のディレクティブ・ファイルが指定されるとエラーになります。

1つのディレクティブ・ファイル中に、複数のディレクティブを記述できます。

## 【ディレクティブ・ファイルの記述例】

TEST.DR

```
MERGE SEC1 : BOOT
MERGE SEC2 : SBR = SEG0
MERGE SEC3 : AT (0100H)
```

## 2.1.3 数 値

ディレクティブに数値定数を記述する場合（たとえば、再配置属性定義を絶対アドレス指定する場合は、10進数か16進数で記述してください。

## 2.1.4 コメント文

ディレクティブ・ファイル中に“;”か“#”が記述されたときは、そこから改行文字（LF）かファイルの終わり（EOF）までをコメントとして扱います。

## 2.1.5 セクション配置ディレクティブ（マージ・ディレクティブ）

マージ・ディレクティブは、指定したセクションをメモリ上の特定番地に配置するディレクティブです。

## 【書 式】

```
MERGE△〈セクション名〉 [△] : [△] [〈再配置属性定義〉] [△〈結合型定義〉] [ [△] =
[△] 〈メモリ領域指定〉 ]
```

## (1) 〈セクション名〉

〈セクション名〉は、オブジェクト・ファイル中に含まれるセクション名です。〈セクション名〉として、オブジェクト・ファイル中に含まれないものは指定できません。

なお、〈セクション名〉は大文字/小文字を区別するので、ソース・ファイル中に記述したとおりに指定してください。

## (2) 〈再配置属性定義〉

〈再配置属性定義〉として指定できるものを次に示します。

## 【再配置属性定義】

```
AT [△] ( [△] 〈スタート・アドレス〉 [△] ) ; 絶対アドレス指定
BOOT ; スタートアップ・ルーチン指定
VECTn ; インダイレクト割り込み処理ルーチン指定
(割り込みベクタ・アドレスにBR命令を生成
する)
DVECTn ; ダイレクト割り込み処理ルーチン指定 (割り
```

	込みベクタ・アドレスにBR命令を生成しない)
CROM	;非プログラム領域 (DTSのCROMなど) のデータ指定
DSYS	;ダイレクト・システム・サブルーチン指定 (SYSCAL命令のエントリ・アドレスにBR命令を生成しない)
SSYS	;システム・サブルーチン指定 (SYSCAL命令のエントリ・アドレスにBR命令を生成する)
DSBR	;ダイレクト・サブルーチン指定 (CALL命令のエントリ・アドレスにBR命令を生成しない)
SBR	;サブルーチン指定 (CALL命令のエントリ・アドレスにBR命令を生成する)

注意 上記予約語は、大文字か小文字のどちらかで記述してください。大文字と小文字が混在していると、エラーになります。

〈スタート・アドレス〉には、数値定数を記述します。シンボルは使用できません。

VECTn/DVECTnのnには、割り込み要因ごとのベクタ・アドレスを指定します ( $1 \leq n \leq$  最大ベクタ・アドレス)。なお、最大ベクタ・アドレスは各デバイスにより異なります。またアドレスは16進数で指定しますが、16進数を示す“H”は付加しないでください。

各再配置属性の処理を次に示します。

図2-1 再配置属性 (AT) とその配置方法

再配置属性：AT (〈スタート・アドレス〉)		
スタート・アドレス		[配置方法] 指定されたスタート・アドレスにセクションを配置する。
	セクション	



図2-2 再配置属性 (BOOT) とその配置方法

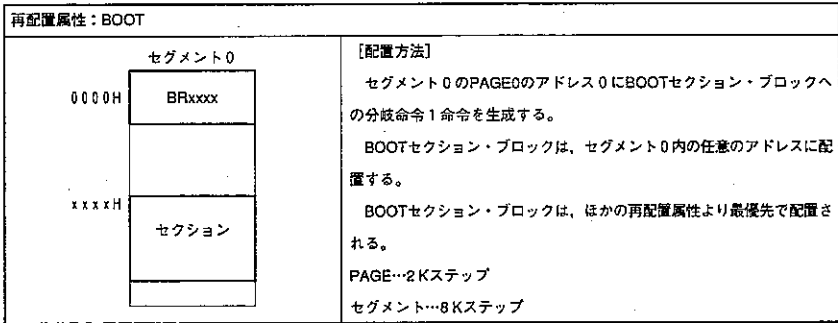


図2-3 再配置属性 (CROM) とその配置方法

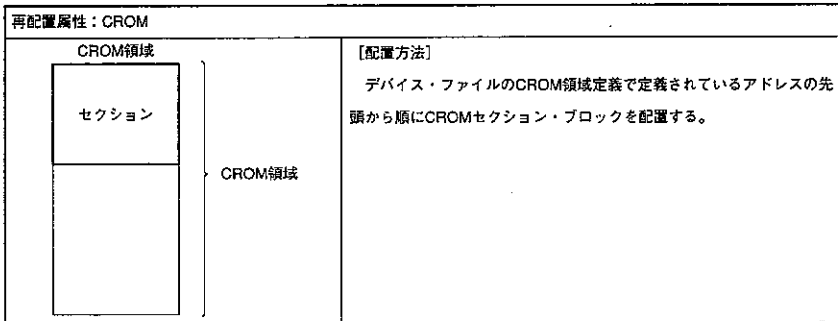


図2-4 再配置属性 (VECTn) とその配置方法

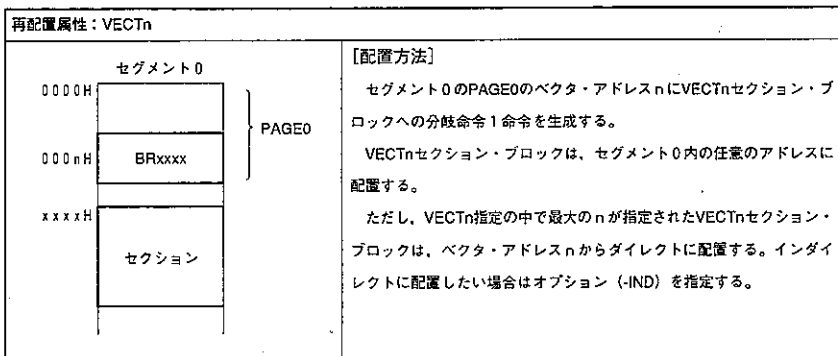


図 2-5 再配置属性 (DSYS) とその配置方法

再配置属性：DSYS											
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 15%; text-align: center;">xn00H</td> <td style="width: 15%; text-align: center;">システム・セグメント</td> <td rowspan="4" style="width: 10%; text-align: center; vertical-align: middle;">} 先頭 16ステップ</td> <td rowspan="4" style="width: 10%; text-align: center; vertical-align: middle;">} BLOCKn</td> </tr> <tr> <td style="text-align: center;">xn0FH</td> <td style="text-align: center;">セクション</td> </tr> <tr> <td style="text-align: center;">xnFFH</td> <td style="text-align: center;">-----</td> </tr> <tr> <td style="text-align: center;">xnFFH</td> <td style="text-align: center;">-----</td> </tr> </table>	xn00H	システム・セグメント	} 先頭 16ステップ	} BLOCKn	xn0FH	セクション	xnFFH	-----	xnFFH	-----	<p><b>【配置方法】</b></p> <p>セクション・ブロックの先頭がシステム・セグメントのPAGE0の各ブロックの先頭16ステップに入るように順に配置する。</p> <p>ただし、セクションの先頭アドレスが先頭16ワードの最上位アドレスに入るように配置する。</p> <p>システム・セグメントの各エントリは、PAGE0の各ブロックの先頭16ステップであるため、配置アドレスが不連続になる可能性がある。</p> <p>この再配置属性は、SYSよりも優先して配置される。</p> <p>BLOCK...256ステップ</p>
xn00H	システム・セグメント	} 先頭 16ステップ			} BLOCKn						
xn0FH	セクション										
xnFFH	-----										
xnFFH	-----										

図 2-6 再配置属性 (SSYS) とその配置方法

再配置属性：SSYS											
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 15%; text-align: center;">xn00H</td> <td style="width: 15%; text-align: center;">システム・セグメント</td> <td rowspan="4" style="width: 10%; text-align: center; vertical-align: middle;">} 先頭 16ステップ</td> <td rowspan="4" style="width: 10%; text-align: center; vertical-align: middle;">} BLOCKn</td> </tr> <tr> <td style="text-align: center;">xn0FH</td> <td style="text-align: center;">BRxxxx</td> </tr> <tr> <td style="text-align: center;">xnFFH</td> <td style="text-align: center;">-----</td> </tr> <tr> <td style="text-align: center;">xxxxH</td> <td style="text-align: center;">セクション</td> </tr> </table>	xn00H	システム・セグメント	} 先頭 16ステップ	} BLOCKn	xn0FH	BRxxxx	xnFFH	-----	xxxxH	セクション	<p><b>【配置方法】</b></p> <p>セクション・ブロックへの分岐命令をシステム・セグメントのPAGE0の各ブロックの先頭16ステップにあるシステム・コール・エントリに順に生成する。</p> <p>SSYSセクション・ブロックは、各ブロックの先頭16ステップ以外の領域に配置する。</p> <p>ただし、1つのブロック内に配置できない場合はPAGE0以外の任意のアドレスに配置する。</p>
xn00H	システム・セグメント	} 先頭 16ステップ			} BLOCKn						
xn0FH	BRxxxx										
xnFFH	-----										
xxxxH	セクション										

図 2-7 再配置属性 (DSBR) とその配置方法

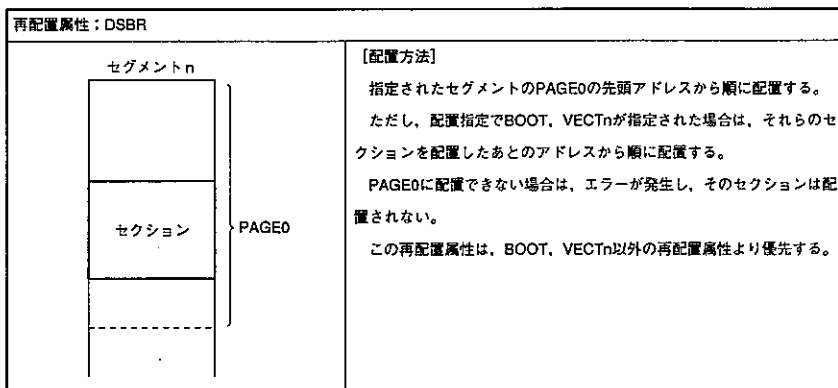


図 2-8 再配置属性 (SBR) とその配置方法

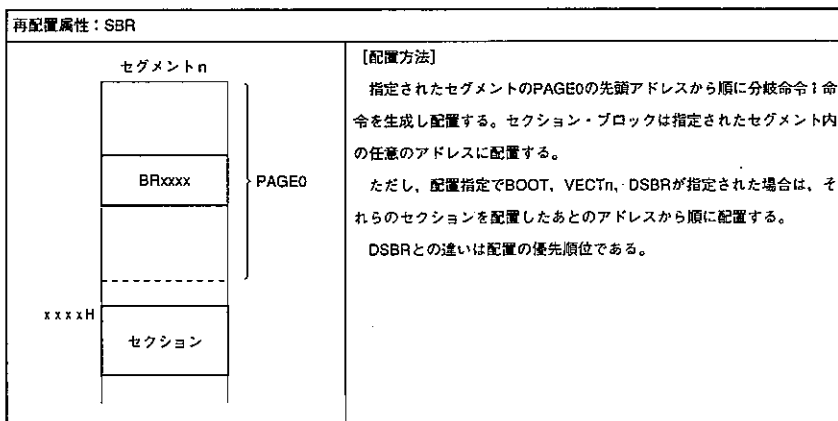
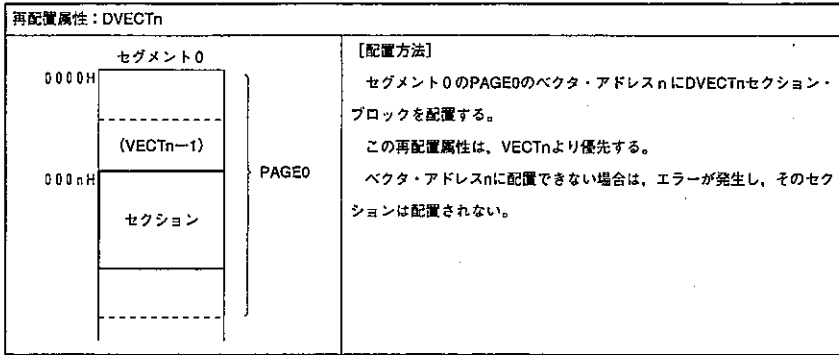


図2-9 再配置属性 (DVECTn) とその配置方法



## 【再配置属性定義における注意】

- ① 〈再配置属性定義〉の指定が省略されたときは、メモリ領域中の任意のアドレスに配置しません。
- ② 指定された〈スタート・アドレス〉で配置を行うと、そのセクションが配置されるメモリ領域の範囲を越えてしまうときは、EPA領域に配置します。EPA領域も越える場合はエラーとなります。
- ③ 1つのマージ・ディレクティブで、〈再配置属性定義〉が複数記述された場合はエラーとなります。このとき、エラー・メッセージを出力したあとはかのディレクティブのエラーをチェックし、ディレクティブの解釈が終了した時点でLK17Kの実行を中止します。
- ④ 〈再配置属性定義〉に従い配置を行うと、そのセクションの一部がEPA領域に配置される場合は、EPA領域に配置される部分に新たにセクションを発生します。発生するセクション名は、次のとおりです。

@ 〈セクション名〉      〈セクション名〉: EPA領域に配置される対象となったセクション名 (〈再配置属性定義〉をした元のセクション名)。

- ⑤ 〈再配置属性定義〉で分岐命令1命令を生成する場合は、その分岐命令に対し新たにセクションを発生します。発生するセクション名は、次のとおりです。

? 〈セクション名〉      〈セクション名〉: 分岐命令1命令を生成する対象となったセクション名 (〈再配置属性定義〉をした元のセクション名)

## (3) &lt;結合型定義&gt;

〈結合型定義〉は指定されたセクションに対し、同名の入力セクションが複数あるときにセクションの結合方法を指定するものです。

〈結合型定義〉として指定できる型とその意味を次に示します。結合型についての詳細は、2.2 入力セクションの結合（マージ）を参照してください。

## 【結合型定義】

- SEQUENT LK17Kに入力された順序に従って逐次結合を行う（デフォルト）。  
COMPLETE 同名のセクションが存在したときはエラーとする。

予約語SEQUENTとCOMPLETEは、大文字か小文字のどちらか一方で記述してください。大文字と小文字が混在して記述された場合、エラーとなります。

## 【結合型定義における注意】

- ① 〈結合型定義〉の指定が省略されたときは、SEQUENTの指定があったものとします。
- ② 1つのマージ・ディレクティブで、〈結合型定義〉が複数指定された場合はエラーとなります。このとき、エラー・メッセージを出力したあとほかのディレクティブのエラーをチェックし、ディレクティブの解釈が終了した時点でLK17Kの実行を中止します。

## (4) &lt;メモリ領域指定&gt;

セクションを配置するプログラム・メモリ領域を指定します。

〈メモリ領域指定〉として以下の8つが記述できます。

SEG0（：セグメント0）、SEG1（：セグメント1）、SEG2（：セグメント2）、SEG3（：セグメント3）、SEG4（：セグメント4）、SEG5（：セグメント5）、SEG6（：セグメント6）、SEG7（：セグメント7）

## 【メモリ領域指定における注意】

- ① 〈メモリ領域指定〉が省略されたときは、空いているアドレスから順次配置します。
- ② 次の場合はエラーとなります。このとき、エラー・メッセージを出力したあとほかのディレクティブのエラーをチェックし、ディレクティブの解釈が終了した時点でLK17Kの実行を中止します。
  - ・ 〈メモリ領域指定〉が複数指定されたとき。
  - ・ 〈メモリ領域指定〉として、SEG0、SEG1、SEG2、SEG3、SEG4、SEG5、SEG6、SEG7以外のものが記述されたとき。

## 【マージ・ディレクティブ全体にわたる注意】

- ① マージ・ディレクティブの指定を受けなかった入力セクションは、リンク時に任意のアドレスに配置します。
- ② 次の場合はエラーとなります。このとき、エラー・メッセージを出力したあとほかのディレクティブのエラーをチェックし、ディレクティブの解釈が終了した時点でLK17Kの実行を中止します。
  - ・〈セクション名〉として指定したセクションが存在しないとき。
  - ・同一のセクションに対して、マージ・ディレクティブが複数回指定されたとき。

## 【ディレクティブ・ファイルの記述例】

TEST.DR

MERGE SEC1 : BOOT
MERGE SEC2 : SBR = SEG0
MERGE SEC3 : AT (0100H)

## 2.2 入力セクションの結合 (マージ)

この節では、セクションの結合について説明します。

### 2.2.1 セクションの結合型

セクションは、それぞれ結合型を持ちます。結合型とは、セクションを結合するときどのような形で結合するかを規定する型です。結合型には、次の2つがあります。

- ・ SEQUENT (デフォルト)
- ・ COMPLETE

### 2.2.2 結合型の決定

セクションの結合型は、次のように決定します。

- (1) リンク時にディレクティブ中で結合型が明示的に指定されたときは、ディレクティブで指定された結合型を持ちます。
- (2) リンク時にディレクティブによる結合型の指定がなかったときは、結合型としてSEQUENTが指定されたものとして処理を行います。

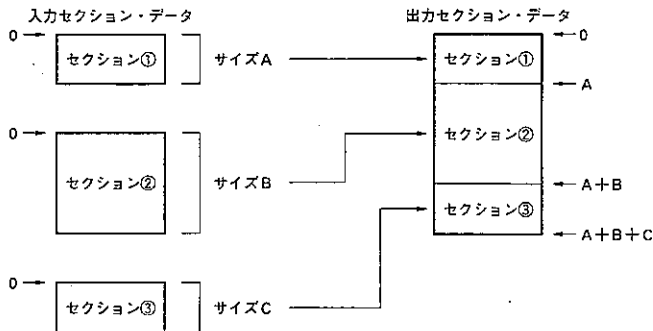
### 2.2.3 結合型ごとのマージ方法

各結合型ごとの、セクションのマージ方法を次に示します。

#### (1) 結合型がSEQUENTのとき

リンカ内での出現順にセクションを順次ずきを空けずにマージします。

図 2-10 結合型がSEQUENTのときのマージ方法



(2) 結合型がCOMPLETEのとき

同名の入力セクションが複数存在する場合は、エラーとなります。



## 2.3 セクションの再配置属性

LK17Kはセクションのマージ処理が終わったあと、セクションの配置アドレスを決定します。

この節では、セクションの配置アドレス決定の詳細について説明します。

### 2.3.1 再配置属性の決定

セクションの再配置属性は、次のように決定します。

- (1) リンク時にディレクティブによって再配置属性が指定されたときは、指定されたとおりの再配置属性になります。
- (2) リンク時に再配置属性が指定されなかったときは、アセンブラ・ソース・プログラム中で指定された再配置属性になります。
- (3) アセンブラ・ソース・プログラム中に再配置属性の指定がなく、かつリンク時に再配置属性が指定されなかったときは、そのセクションをメモリ上の任意のアドレスに配置します。

再配置属性によっては分岐命令 (BR addr) をLK17Kが自動的に生成するものがあります。次に再配置属性ごとの分岐命令生成の有無とセクション・ブロックの配置先を示します。

表2-1 再配置属性によるセクションへの分岐命令生成の有無

再配置属性	分岐命令 (BR addr) の生成	セクション・ブロックの配置先
AT	なし	指定されたアドレス
BOOT	あり (SEG0の0番地に生成)	SEG0
VECTn (ダイレクト) 注1	なし	SEG0のn番地
VECTn (インダイレクト)	あり (SEG0のn番地に生成)	SEG0
DVECTn注1	なし	SEG0のn番地
CROM	なし	SEG0-SEG7 (デバイス固有)
DSYS	なし	システム・セグメントのPAGE0の各BLOCKの先頭16ステップ内
SSYS	あり (システム・セグメントのPAGE0の各BLOCKの先頭16ステップ内に生成)	SEG0-SEG7 (システム・セグメント) (デバイス固有)
DSBR	なし	SEG0-SEG7のPAGE0
SBR	あり (SEG0-SEG7のPAGE0に生成)	分岐命令と同じセグメント内
TABLE注2	なし	SEG0-SEG7
指定なし	なし	SEG0-SEG7

注1、-INDを指定してリンクした場合は、強制的にインダイレクトな配置となります。

2. TABLE属性は、アセンブラのCSEG疑似命令でTABLE定義されたテーブル・ブロックに自動的に割り当てられます。マージ・ディレクティブでは指定できません。

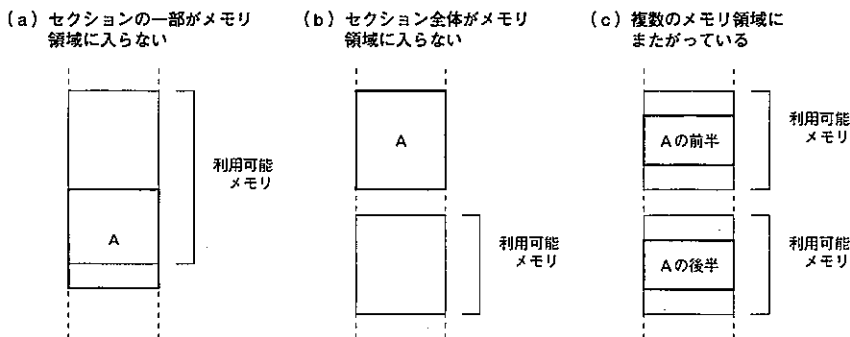
### 2.3.2 セクション配置アドレスの決定

セクションの配置アドレスは、次のように決定します。

- (1) すべてのセクションは、その全サイズが1つのメモリ領域内に入るように配置されます。つまり、次のような配置は行いません。このような配置が指定されたときは、エラー・メッセージを出力したあとリンクの実行を中止します。

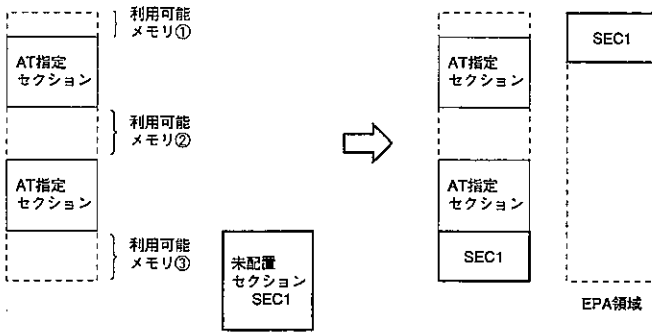
- ・セクションの一部が全部がメモリ領域に入らないような割り付け
- ・アドレス的に不連続な（2つ以上の利用可能メモリ領域をまたぐような）割り付け

図 2-11 不正な配置の例



- (2) マージ・ディレクティブによってセクションにメモリ領域が指定されているときは、そのセクションに指定されたメモリ領域に配置します。
- (3) すべてのセクションは再配置属性を持ち、LK17Kはその再配置属性に従って配置アドレスを決定します。再配置属性ごとの配置アドレス決定ルールについては、2.3.3 セクションの配置を参照してください。
- (4) メモリ空間の最後の利用可能メモリにセクションを配置する場合、その一部が利用可能メモリに入らなければ、EPA領域にその一部を配置します。

図 2-12 EPA領域へのセクションの配置



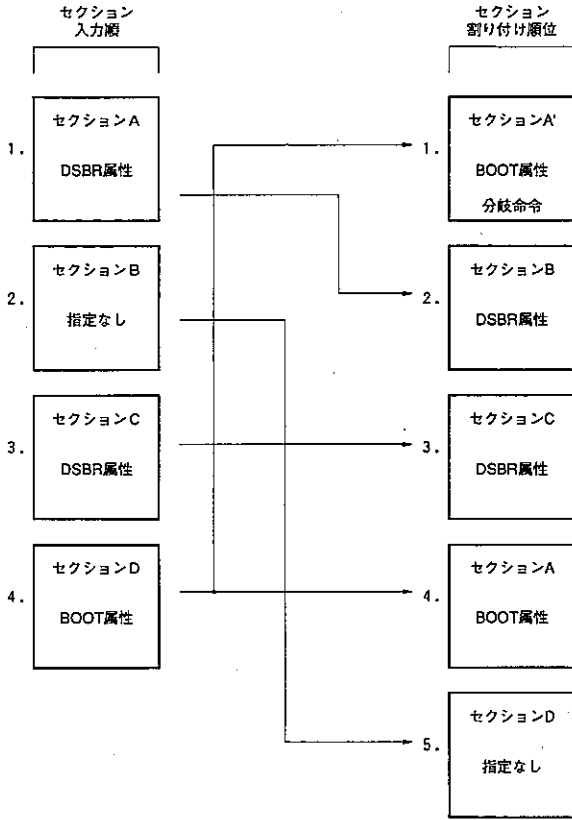
### 2.3.3 セクションの配置

セクションの配置作業は、次の手順で行います。

- (1) 配置の優先順位が高い再配置属性を持つセクションの配置アドレスは、優先順位の低い再配置属性を持つセクションよりも先に決定します。再配置の優先順位については表 2-2 再配置の優先順位を参照してください。
- (2) 再配置属性が同じときは、セクションを入力した順にアドレスを決定します。
- (3) セクションを割り付けられるメモリ領域が複数ある場合は、アドレスが最も低い領域に割り付けます。

★

図 2-13 セクションの配置例



★

表2-2 再配置の優先順位

優先順位	再配置される内容
1	AT属性のセクション BOOT属性に対してLK17Kが生成する分岐命令
2	VECTn属性でダイレクトに実体が配置されるセクション DVECTn属性のセクション
3	VECTn属性に対してLK17Kが生成する分岐命令
4	SSYS属性に対してLK17Kが生成する分岐命令
5	CROM属性のセクション DSYS属性のセクション
6	SSYS属性のセクション (実体)
7	TABLE属性のセクション (LK17Kでは定義不可。アセンブラで定義) SBR属性に対してLK17Kが生成する分岐命令
8	DSBR属性のセクション
9	BOOT属性のセクション (実体) VECTn属性のセクション (実体)
10	SBR属性のセクション (実体)
11	再配置属性の指定はないが、メモリ領域は指定されているセクション 再配置属性の指定がなく、メモリ領域の指定もないセクション

備考 同一優先順位同士の優先順位は、LK17Kへの入力順となります。

## 2.4 シンボル値の決定と出力

LK17Kは、次のシンボル処理を行います。

- (1) 外部定義シンボル/外部参照シンボルの参照関係のチェック
- (2) シンボル値の決定と出力

### 2.4.1 外部定義シンボル/外部参照シンボルの参照関係のチェック

LK17Kは、外部定義シンボル（アセンブラ・ソース・プログラム中でPUBLIC疑似命令で外部定義であることを宣言したシンボル。以下、PUBLICシンボルという）と、外部参照シンボル（アセンブラ・ソース・プログラム中でEXTRN疑似命令で外部参照であることを宣言したシンボル。以下EXTRNシンボルという）間の定義、参照関係をチェックします。

定義/参照関係は次のようにチェックされます。

#### (1) 二重定義のチェック

LK17Kは、入力されたすべてのオブジェクト・モジュール・ファイルの中で、同じ名前のPUBLICシンボルが2回以上定義されていないことをチェックします。もし2回以上定義されていたときはエラー・メッセージを出力し、リンク処理を中止します。

#### (2) 未定義シンボルのチェック

LK17Kは、すべてのEXTRNシンボルに対して、入力されたオブジェクト・モジュール・ファイルのどこかに同名のシンボルがPUBLICで定義されていることをチェックします。EXTRNシンボルだけが存在し、PUBLICシンボルがどこにも定義されていなかった場合は、エラー・メッセージを出力します。さらに-JUNKが指示されている場合は、参照している命令に対してアセンブラが出力したオブジェクト・コードをリロケーション未解決のまま利用してオブジェクト・ファイルを生成します。

## 2.5 プログラム・メモリ・オーバーフロー時のエラー・メッセージ

17Kでは、ROM容量を越えた部分をデバッグできます（1.2.3 EPA領域参照）。これに対応してLK17Kでは、プログラムがROMの容量を越えても、ROM容量の約2倍のアドレスに記述されたプログラムまでは、オブジェクト・コードを生成します。ただしROM容量を越えた場合、実際のデバイスで動作させることはできないため、次のエラー・メッセージを出力します。ROMコード発注時には、このエラーが出力されなくなるまでプログラム・サイズを縮めてください。

(1) プログラムが②の領域にある場合

①の領域を越えた時点で、そのセクションにエラー（F306）が発生します。ただし、オブジェクト・コードは正常に生成します。

(2) プログラムが④の領域にある場合

セクション全体が④の領域に入った場合は、エラー（F317）が発生します。セクションの一部が④の領域に入った場合は、エラー（F318）が発生します。ただし、オブジェクト・コードは正常に生成します。

(3) プログラムが③の領域にある、または④の領域を越えた場合

エラー（F206）が発生します。オブジェクト・コードは生成しません。

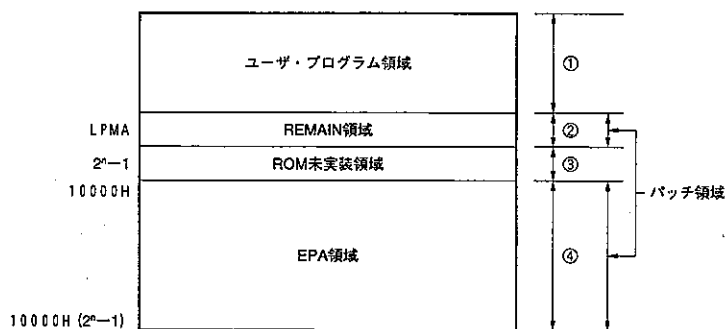
(4) 飛び先番地が②の領域にある場合（REMAIN AREA）

エラー（F309）が発生します。

(5) 飛び先番地が④の領域にある場合（EPA AREA）

エラー（F310）が発生します。

図2-14 17Kシリーズのセグメント内アドレス空間



備考 LPMA (Last Program Memory Area) : 製品に実装されているプログラム・メモリの最終アドレス



## 2.6 インクリメンタル・ロード用差分ファイル

### 2.6.1 差分ファイルの作成

RA17K/emlC-17K対応のSIMPLEHOSTでは、修正のあったモジュールだけ再アSEMBル（再コンパイル）します。LK17Kは、リンカ・オプション（-INC）が指定された場合に、オブジェクト・コード／ディバグ情報のうち、差分が発生した部分だけを差分ファイルに出力します。この差分ファイルをIE-17Kにロードすることで、バッチ処理時間を短縮します。

#### (1) 差分ファイル出力時の注意

- ① -INCと同時に、-NOOBや-NOICを指定することはできません。
- ② -INC指定時、前回のリンク時にあったセクションをソース修正で削除しないでください。削除するとエラーとなります。

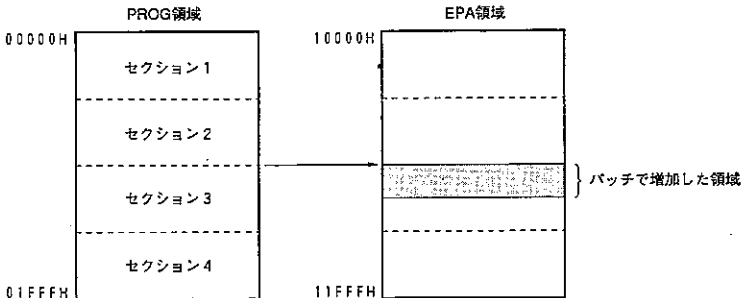
#### (2) 差分ファイルの出力例

バッチによりオブジェクト・コードとサイズが変化した場合の差分ファイルの出力内容の概要を次に示します。

##### 例1. セクション2をバッチして、セクション2のサイズが増えた場合

差分ファイルには、オブジェクト・コード／EPAビット・マップで差分が現れた箇所以降のすべてのオブジェクト・コード／EPAビット・マップを出力します。

セクション2のバッチによりセクション2のサイズが増加した場合、セクション2のサイズを越えた部分のコードは、増加したアドレスにEPAビットを立てた領域、つまりセクション3のEPA領域に配置します。

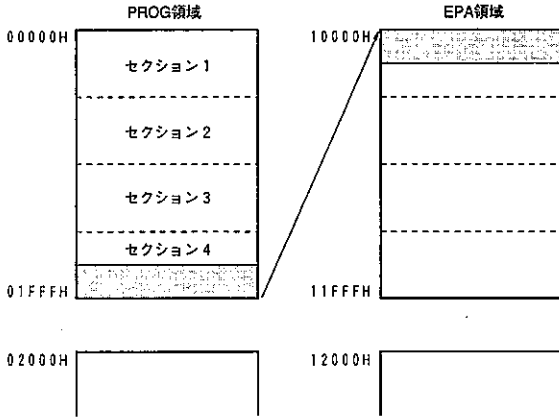


##### 2. セクション4（セグメント内の最高位アドレス側に配置）をバッチして、セクション4のサイズが増えた場合

差分ファイルには、オブジェクト・コード／EPAビット・マップで差分が現れた箇所以降の

すべてのオブジェクト・コード/EPAビット・マップを出力します。

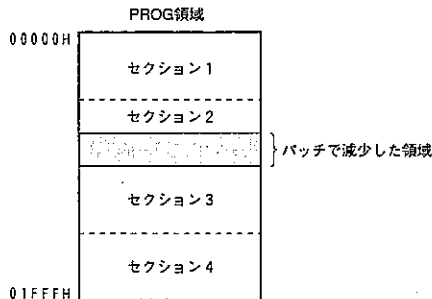
セクション4のバッチによりセクション4のサイズが増加した場合は、増加した部分のコードは、プログラム領域の連続した空き領域から配置します。プログラム領域の最終アドレスまでに収まらない場合は、当該セグメントのEPA領域(10000H-)の先頭から配置します。プログラム領域の最終アドレス(セグメント単位)から当該セグメントのEPA領域の先頭への分岐は、プログラム領域の最終アドレスの命令に対し、EPAビットを付加することで制御します。このため、最終アドレスの命令に対するEPAビット・マップを“1”にします。



備考 □ はバッチで増加した領域です。

例3. セクション2をバッチして、セクション2のサイズが減少した場合

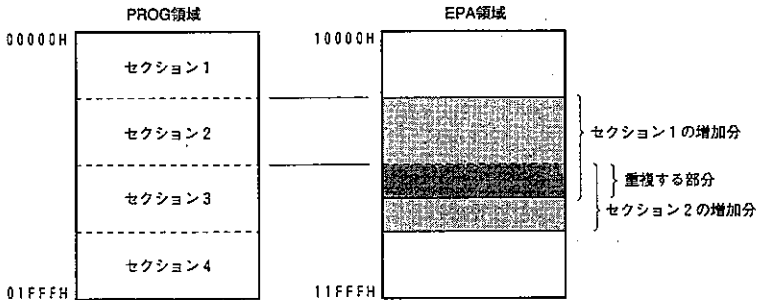
差分ファイルには、オブジェクト・コード/EPAビット・マップで差分が現れた箇所以降のすべてのオブジェクト・コード/EPAビット・マップを出力します。



例4. 連続するセクションをバッチし、前セクションのバッチ部分が、あとのセクション・サイズを越える場合（複数セクションをバッチ）

連続するセクション、セクション1、セクション2をバッチして、どちらもコードが増加する場合は、セクション1のバッチ部分がセクション2のセクション・サイズを越えないようにしてください。越えてしまうと2つのバッチ部分で重複する部分が生じ、正常にバッチできません。この場合、リンカでは差分ファイル作成時にエラーが発生し、差分ファイルを作成しません。

複数セクションをバッチしても、バッチ部分が重複しない場合は、差分ファイルを正常に出力します。



## 第Ⅱ編

### 操作編

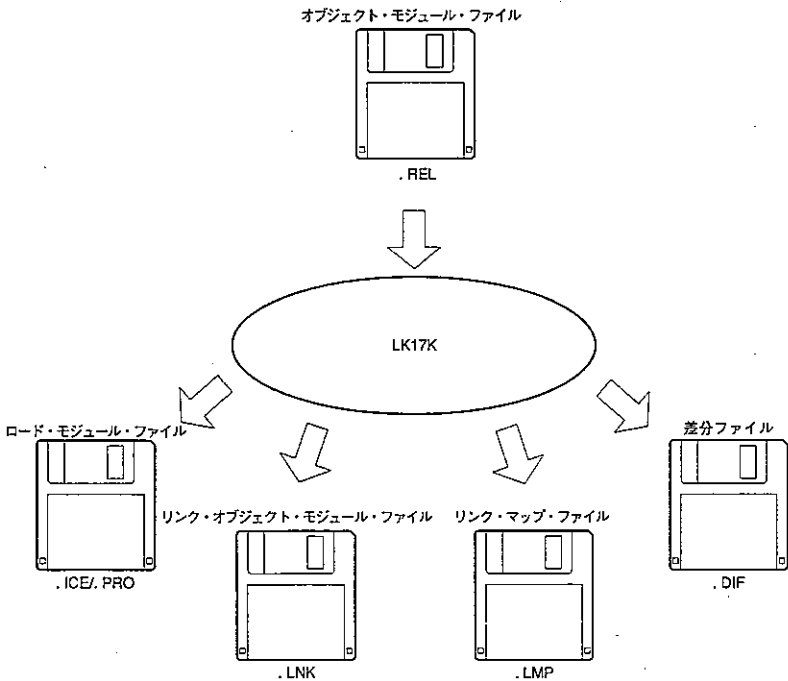
(× 毛)

# 第1章 概要

LK17Kは、アセンブラ (RA17K) / コンパイラ (em/C-17K) が作成したオブジェクト・モジュール・ファイル (.REL) を入力して、次のファイルを出力します。

- ・ロード・モジュール・ファイル (.ICE/.PRO)
- ・リンク・オブジェクト・モジュール・ファイル (.LNK)
- ・リンク・マップ・ファイル (.LMP)
- ・差分ファイル (.DIF)

図 1-1 処理概要



[x ㄷ]

## 第2章 システム構成

### ★ 2.1 システム環境

LK17Kは、次のような環境で動作します。

#### 2.1.1 ハードウェア環境

(1) ホスト・マシン

- ① PC-9800シリーズ
- ② PC/AT™

(2) OS

MS-DOS™ : Ver.3.30以上  
PC DOS™ : Ver.5.02以上  
Windows™3.1, Windows95

(3) CPU

80386以上

(4) 最低動作可能メモリ

コンベンショナル・メモリ : 300 KB以上  
プロテクト・メモリ : 1.5 MB以上

(5) フロッピー・ディスク・ドライブ



## 2.1.2 ソフトウェア環境

## (1) コマンド・ライン環境

メモリ・ドライバとして最低限、himem.sysあるいはそれに準ずるものが必要です。

次の環境での動作保証をしてください。

- ・ himem.sysのみの環境
- ・ himem.sys+emm386.exeの環境

## (a) PC-9800シリーズ

OS	環境	himem.sysのみ	himem.sys+emm386.exe
MS-DOS 3.30D		○注1	×注1
MS-DOS 5.00A		○	×
MS-DOS 6.2		○	○注2, 3
Windows95 (dos prompt only)		○	○注2

注1. MS-DOS 3.30Dにはhimem.sysおよびemm386.exeは付属されていませんので、Windows3.1付属のドライバ使用を前提とします。

2. emm386.exeの最後に/DPMIのスイッチ追加が必要です。
3. DOS付属のDPMIサーバを組み込んだ場合は動作保証できません。

MS-DOS 5.00AおよびWindows3.1付属のemm386.exeには/DPMIの機能がありませんので、UMB/EMSを確保したい場合はサード・パーティ品と組み合わせるか、またはコマンド・ラインでは使用せず、Windows3.1のDOS窓を使用してください。

## (b) PC/AT, AT互換機

OS	環境	himem.sysのみ	himem.sys+emm386.exe
PC DOS 5.0		○	○
PC DOS 6.3		○	○
MS-DOS 6.2		○	○
MS-DOS 7.0		○	○
Windows95 (dos prompt only)		○	○

## (2) DOS窓環境

## (a) PC-9800シリーズ, PC/AT, AT互換機共通

OS	環境	himem.sysのみ	himem.sys+emm386.exe
Windows3.1		○注	○注
Windows95		○	○

注 system.iniの [386Enh] フィールドに添付のwindpmi.386の組み込みが必要です。RA17Kアセンブラ・パッケージ添付のインストーラで組み込むことができます。

## ★ 2.2 ファイル構成

LK17Kの起動には次のファイル構成が必要です。

## (1) リンカ本体

- ・LK17K.EXE : LK17Kの本体 (32ビット・アプリケーション)

## (2) DOS-Extender用のファイルを添付

Borland Cに添付されている次の3本のファイルを添付します。

- ・32RTM.EXE
- ・DPMI32VM.OVL
- ・WINDPMI.386

上記ファイルは再配布可能となっています。

なお、上記ファイルは、RA17Kアセンブラ・パッケージ内のほかの32ビット・アプリケーションでも共通に使用します。

## 2.3 入出力ファイル

### 2.3.1 入出力ファイル一覧

LK17Kが取り扱うファイルを次に示します。

表 2-1 入出力ファイル一覧

ファイル、略称	入出力	出力先	説明
オブジェクト・モジュール・ファイル (.REL)	入力	—	アセンブル結果のオブジェクト・コードのバイナリ・イメージ・ファイル。
デバイス・ファイル (.DEV)	入力	—	各製品固有の情報を格納したファイル。
ディレクティブ・ファイル (.DR)	入力	—	LK17Kに対するリンク指示を記述したファイル。
パラメータ・ファイル (.PLK)	入力	—	LK17Kのパラメータを記述したファイル。
リンク・オブジェクト・モジュール・ファイル(.LNK)	出力	ファイルまたは出力デバイス	リンク結果のオブジェクト・コードのバイナリ・イメージ・ファイル。
ロード・モジュール・ファイル (.ICE_PRO)	出力	ファイルまたは出力デバイス	.ICE：インテルHEX形式のロード・モジュール・ファイル。 デバッグ情報を含む。 .PRO：インテルHEX形式のロード・モジュール・ファイル。 デバッグ情報を含まない。
テンポラリ・ファイル (LKxxxxxx)	入出力	ファイルまたは出力デバイス	リンクのための中間ファイル。
ログ・ファイル (LK17K.LOG)	出力	ファイルまたは出力デバイス	リンク実行時にディスプレイに出力されるメッセージを格納したファイル。ファイル名はLK17K.LOGに固定。
差分ファイル (.DIF)	出力	ファイルまたは出力デバイス	インクリメンタル・リンク・オプション (-INC) 指定時に出力するファイル。新/旧.ICEファイルの差分を格納したファイル。
リンク・マップ・ファイル (.LMP)	出力	ファイルまたは出力デバイス	リンクの結果（配置情報、シンボル情報）を格納したファイル。

### 2.3.2 出力先

出力先には、次の2つがあります。

- ・ファイル
- ・出力デバイス

## (1) ファイル

## 【形式】

[ 〈バス名〉 ] 〈ファイル名〉

つまり

[ 〈ドライブ名〉 : ] [ ¥ ] [ [ 〈ディレクトリ名〉 ¥ ] ... ] 〈プライマリ・ネーム〉 [ , [ 〈ファイル・タイプ〉 ] ]

## 【説明】

カレント・ドライブ、カレント・ディレクトリに指定ファイルがない場合は、〈ファイル名〉に〈バス名〉を付けて指定します。

〈バス名〉が省略された場合は、カレント・ドライブのカレント・ディレクトリとみなされます。

## (2) 出力デバイス

## 【形式】

〈デバイス名〉

## 【説明】

- ① 出力デバイスとして、次の出力先を指定できます。

表 2-2 出力デバイス

デバイス名	出力先
CON	コンソール
PRN	プリンタ バイナリ形式ファイルの出力先としては指定できません。
NUL	ヌル・デバイス
AUX	シリアル・インタフェース (RS-232-C)

## 2.3.3 入力ファイルの指定

## (1) 入力ファイル名の指定数

- ・LK17Kでは、複数の入力ファイルからロード・モジュール・ファイルを作成する必要があるため、複数の入力ファイル名を指定できるようになっています。
- ・入力ファイルは、最高128ファイルまで指定できます。129ファイル以上指定された場合は、エラー・メッセージを出力したあと、LK17Kの実行を中止します。

### (2) 誤った入力ファイル名の指定

- ・指定された入力ファイルが存在しない場合は、アボート・エラーになります。
- ・入力ファイル名として、デバイス型ファイル名やパス名だけが指定された場合はアボート・エラーになります。

### (3) 入力ファイル名の重複指定

- ・1つの入力ファイルが重複して指定された場合、アボート・エラーになります。
- ・入力ファイル名とパラメータ・ファイル名が一致している場合は、アボート・エラーになります。コマンド・ライン上の記述が異なっている場合（たとえば一方がパス名付きで指定され、もう一方がパス名なしで指定されている場合）でも、パス名を補った結果同一ファイルになる（実体の一致）場合はアボート・エラーになります。

## 2.3.4 出力ファイルの指定

### (1) 出力ファイル名の指定方法

- ・出力ファイル名はオプション中に記述します。
- ・出力ファイル名はパス名だけでも指定できます。  
パス名だけが指定されたときには、その指定されたパスにデフォルト名で作成します（デフォルト名については2.3.5 出力ファイル名の解釈を参照）。  
なお、パス名としてディレクトリ名だけを指定する場合、最後の“\*”は省略できます。

### (2) 誤った出力ファイル名の指定

- ・指定された出力ファイルがリード・オンリー・ファイルとしてすでに存在する場合は、アボート・エラーになります。リード・オンリー・ファイルでない場合は、そのファイルに上書きされます。
- ・アスキー形式ファイルの出力先としてデバイス型ファイル名のCLOCK(clock)が指定された場合は、アボート・エラーになります。また、バイナリ形式ファイルの出力先としてCON(con)、PRN(prn)、NUL(nul)またはCLOCK(clock)が指定された場合も、アボート・エラーになります。
- ・指定された出力ファイル名中に存在しないパス名が含まれている場合は、アボート・エラーになります。

### (3) 出力ファイル名の重複指定

- ・入力ファイル名、出力ファイル名、またはパラメータ・ファイル名と、出力ファイル名が重複して指定された場合は、アボート・エラーになります。  
このときコマンド・ライン上の記述が異なっている場合（たとえば、一方がパス名付きで指定され、もう一方がパス名なしで指定されている場合）でも、パス名を補うと同一ファイルになる（実体の一致）場合はアボート・エラーになります。

## 2.3.5 出力ファイル名の解釈

出力ファイル名は、出力ファイル名変更オプションで指定します。

表 2-3 出力ファイル名の解釈

オプションの有無 処理	出力ファイル名を変更した場合			出力ファイル名を変更しない場合
	ファイル名を指定	パス名だけを指定	デバイス名を指定	
LK17Kの処理	指定ファイル名(指定ファイル名に拡張子が付いていないときは、ディフォルトの拡張子を補った名前)	指定パスに、最初に指定された入力ファイル名の拡張子をディフォルトの拡張子に変更した名前	指定デバイス名	カレント・パスに、最初に指定された入力ファイル名の拡張子をディフォルトの拡張子に変更した名前

備考 ディフォルトの拡張子については、次の 2.3.6 拡張子の省略時解釈を参照してください。

表 2-4 出力ファイル名の指定の例

例	コマンド・ライン上の記述	入力ファイル名	出力ファイル名
1	X>LK17K -OBJ = OUT.LNK IN1.REL IN2.REL	IN1.REL, IN2.REL	OUT.LNK, IN1.ICE
2	X>LK17K -OBJ = VLM IN1.REL IN2.REL	IN1.REL, IN2.REL	パス "*VLM" がある場合 VLM*IN1.LNK, IN1.ICE
			パス "*VLM" がいない場合 VLM.LNK, IN1.ICE
3	X>LK17K -OBJ = VLM* IN1.REL IN2.REL	IN1.REL, IN2.REL	パス "*VLM*" がある場合 VLM*IN1.LNK, IN1.ICE
			パス "*VLM*" がいない場合 アボート・エラー
4	X>LK17K -OBJ = NULL IN1.REL IN2.REL	IN1.REL, IN2.REL	NULL, IN1.ICE
5	X>LK17K IN1.REL IN2.REL	IN1.REL, IN2.REL	IN1.LNK, IN1.ICE

### 2.3.6 拡張子の省略時解釈

拡張子を省略して入力ファイルを指定した場合、次に示す拡張子を省略時解釈とします。

表 2-5 拡張子の省略時解釈

ファイルの種類	省略時解釈
オブジェクト・モジュール・ファイル	.REL
ディレクティブ・ファイル	.DR
パラメータ・ファイル	.PLK
ロード・モジュール・ファイル (.ICE)	.ICE
ロード・モジュール・ファイル (.PRO)	.PRO
リンク・オブジェクト・モジュール・ファイル	.LNK
リンク・マップ・ファイル	.LMP

## 2.4 テンポラリ・ファイル

LK17Kは、処理中にいくつかのテンポラリ・ファイルを作成します。

テンポラリ・ファイルは、LK17Kの実行が終了（正常/異常）したとき、およびキー入力（CTRL-C）でLK17Kの実行が中断されたとき、自動的に削除されます。

同名のファイルがすでにある場合でも、書き込み保護がされていないならば上書きされます。

テンポラリ・ファイルは次の優先順位に従って決定されたドライブ、ディレクトリに作成されます。

- ① 作業パス指定オプション (-WORK) で指定されたドライブ、ディレクトリ
- ② 環境変数TMPに指定されたドライブ、ディレクトリ
- ③ カレント・ドライブ、カレント・ディレクトリ

テンポラリ・ファイルの一覧を次に示します。

表 2-6 テンポラリ・ファイル一覧

テンポラリ・ファイル名	説 明
LKxxxxxx	LK17Kがシンボル処理で使用するテンポラリ・ファイル
LKyyyyyy	
LKzzzzzz	
LKmmmmmm.ICE	LK17Kが出力ファイル作成処理で使用するテンポラリ・ファイル
LKmmmmmm.PRO	
LKmmmmmm.DIF	
LKmmmmmm.LMP	
LKnnnnnn	LK17Kが入力ファイルに対して使用するテンポラリ・ファイル

備考 mmmmmm, nnnnnn, xxxxxx, yyyyyy, zzzzzzは、任意の数字列です。

## ★ 2.5 シンボル数

シンボル名、セクション名が8文字（日本語の場合4文字）以下で、ホスト・マシンのメモリの空き容量が480 Kバイトの場合のシンボル数を次に示します。

シンボル総数 パブリック・シンボル：65535個

ローカル・シンボル：アセンブラから出力されるシンボル総数を保証します

## 2.6 環境変数

LK17Kで使用できる環境変数を次に示します。

・TMP：テンポラリ・ファイルを作成するパスを指定するための環境変数

環境変数TMPには、RAMディスクなどの高速なファイル装置を指定しておくことをお勧めします。

## 2.7 処理の中断

キー入力（CTRL-C）でOSに制御を戻します。このときには、開いていたすべてのテンポラリ・ファイルを削除します。



[x 毛]

## 第3章 操作方法

### 3.1 起 動

#### 3.1.1 コマンド・ライン上にオプションを指定する場合

【形 式】

X>[パス名]LK17K△[オプション[△オプション…]]△(オブジェクト・モジュール・ファイル名)[△オプション[△オプション…]]

| | | | | | |  
① ② ③ ④ ⑤ ④

- ① カレント・ドライブ名
- ② ディレクトリ名
- ③ LK17Kの実行ファイル名
- ④ LK17Kに対して動作の詳細を指示するオプション。

複数のリンク・オプションを指定する場合は、それぞれのリンク・オプション間を空白で区切る。

リンク・オプションはオブジェクト・モジュール・ファイル指定の前後、途中どこにでも指定できる。

- ⑤ LK17Kに入力するオブジェクト・モジュール・ファイル名を指定する。

備考 LK17Kだけ指定した場合、ヘルプ・メッセージが表示されます。

#### 3.1.2 パラメータ・ファイルを使用する場合

パラメータ・ファイルは、入出力ファイル名やオプションなど、LK17Kの起動に必要な情報を記述したファイルです。

パラメータ・ファイルは、起動に必要な情報がコマンド・ラインに指定しきれない場合や、リンクするたびに同じオプションを繰り返し指定するような場合に使用します。

パラメータ・ファイルを使用する場合には、コマンド・ラインにパラメータ・ファイル指定オプション(-PAR)を指定してください。

【形 式】

X>LK17K [△ (オブジェクト・モジュール・ファイル) ] △-PAR= (パラメータ・ファイル名)

| |  
① ②

- ① パラメータ・ファイル指定オプション
- ② パラメータ・ファイルのファイル名

パラメータ・ファイル内での記述規則を次に示します。

- (1) コマンド・ライン行でオブジェクト・モジュール・ファイル名を省略する場合は、必ずパラメータ・ファイル内でオブジェクト・モジュール・ファイル名を指定してください。
- (2) コマンド・ライン上に記述するオブジェクト・モジュール・ファイル名は、-PARのあとにも記述できます。
- (3) パラメータ・ファイルには、コマンド・ラインで指定すべきすべてのリンク・オプション、出力ファイル名を指定してください。
- (4) パラメータ・ファイルには、コメントも記述できます。  
“;”や“#”以降の文字は、CRかEOFまですべてコメントと解釈します。

【記述例】

```
TEST.PLK

; Link options
-OBJ -ICE -LMAP -MP
;input files
A.REL B.REL C.REL
```

### 3.1.3 実行開始メッセージ

LK17Kが起動されると、次の実行開始メッセージを出力デバイスに出力し、リンクを開始します。

```
X>[バスの名]LK17K[△オプション[△オプション…]]△[オブジェクト・ファイル・リスト][△オプション[△オプション…]] ←コマンド・ライン
```

```
17K Series Linker Vx.yz   [DD MMM YY]                               ←実行開始メッセージ
Copyright (C) NEC Corporation XXXXX                               ← 同上
```

```
--- Link start hh:mm:ss yy/dd/mm ---                               ← 同上
```

```
x.yz                      : バージョン番号
DD MMM YY                 : 作成日付
hh:mm:ss yy/dd/mm        : リンク開始時刻と日付
```

## 3.1.4 ヘルプ・メッセージ

パラメータを指定せずにLK17Kを起動した場合は、次のヘルプ・メッセージを出力デバイスに出力します。

```
X> [バス名] LK17K [△]                                ←コマンド・ライン

17K Series Linker Vx.yz  [DD MMM YY]
  Copyright (C) NEC Corporation XXXXX

usage :  LK17K [option [...] ] input-file [...] [option [...] ]

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

-OBJ [ECT]  [= file] /-NOOB [JECT]                   : Create object module file
                                                    [with specified name] / Not.
-ICE [= file] /-NOIC [E]                             : Create ice file [with specified name] / Not.
-PROM [= file] /-NOP [ROM]                          : Create pro file [with specified name] / Not.
-HOS [T] /-NOH [OST]                               : Use SIMPLEHOST / Not.
-WOR [K] = path-name                               : Set temporary directory.
-WAR [NING] = n                                     : Set warning level.
-INC [REMENTAL] /-NOIN [CREMENTAL]                 : Create dif file / Not.
-DIR [ECTIVE] = file                               : Read directive file from specified file.
-PAR [AMETER] = file                               : Read parameter file from specified file.
-PROG = "prog-name"                               : Set program name.
-IND [IRECT] = section-name                       : Indirect locate.
-JUN [K] /-NOJ [UNK]                              : Create object module file
                                                    if fatal error occurred / Not.
-LMA [P]  [= file] /-NOL [MAP]                    : Create link map file [with specified name] / Not.
-ML/-NOML                                         : Output local symbol list to link map file / Not.
-MP/-NOMP                                         : Output public symbol list to link map file / Not.

DEFAULT ASSIGNMENT : -OBJ -ICE -NOP -NOHOST -WAR0 -NOIN -NOJUNK -LMA
                   -NOML -NOMP

directive file usage :
MERGE section-name : [location-type-definition] [merge-type-definition]
                   [= segment-name]

example : MERGE SEC1 : BOOT
          MERGE SEC2 : SBR SEQUENT = SEGO

X>                                ←OSのプロンプト
```

ヘルプ・  
メッセージ

### 3.1.5 プログラムの終了

LK17Kは処理が正常に終了したときは正常終了メッセージを、アボート・エラーが起きたときには異常終了メッセージを出力します。

次に正常終了メッセージと異常終了メッセージを示します。

#### (1) 正常終了メッセージ

```
X> [バス名] LK17KΔ [-オプション [Δ-オプション…]] Δ (オブジェクト・モジュール・ファイル) [Δ-
オプション [Δ-オプション…]]

17K Series Linker Vx.yz [DD MMM YY]
  Copyright (C) NEC Corporation XXXXX

--- Link start hh:mm:ss yy/dd/mm ---
--- Link end hh:mm:ss yy/dd/mm ---                                ←正常終了メッセージ

Device file name : MMMMMMMM.DEV (Version : NN)

Total error (s) : XXXXX Total warning (s) : YYYYY                ←正常終了メッセージ

X>                                                                ←OSのプロンプト

MMMMMMMM : デバイス・ファイル名
NN        : デバイス・ファイル・バージョン番号
XXXXX     : エラー数 (10進)
YYYYY     : ワーニング数 (10進)
```

#### (2) 異常終了メッセージ (指定ファイルが見つからなかった場合)

```
X> [バス名] LK17KΔ [-オプション [Δ-オプション…]] Δ (オブジェクト・モジュール・ファイル) [Δ-
オプション [Δ-オプション…]]

17K Series Linker Vx.yz [DD MMM YY]
  Copyright (C) NEC Corporation XXXXX

--- Link start hh:mm:ss yy/dd/mm ---
error A003 : ファイル 'xxxxx' が見つかりません
--- Link end hh:mm:ss yy/dd/mm ---

Program aborted                                                  ←異常終了メッセージ

X>                                                                ←OSのプロンプト
```

### 3.1.6 エラー・レベル

LK17Kは終了状態として、MS-DOSのerrorlevelに次の値を返します。

表3-1 エラー・レベル

終了状態	errorlevel
正常終了	0
ワーニング	0
フェータル・エラー	1
アボート・エラー	2

## 3.2 入 力

LK17Kの入力となるものを次に示します。

- ・オブジェクト・モジュール・ファイル (.REL)
- ・リンク・オプション
- ・ディレクティブ・ファイル
- ・パラメータ・ファイル

リンク・オプションとオブジェクト・モジュール・ファイル名は、コマンド・ラインかパラメータ・ファイル中に記述できます。

ディレクティブ・ファイル名は、ディレクティブ・ファイル名指定オプション (-DIR) で指定します。

パラメータ・ファイル名は、パラメータ・ファイル名指定オプション (-PAR) で指定します。

### 3.2.1 オブジェクト・モジュール・ファイル

LK17Kには、アセンブラ (RA17K) かコンパイラ (*emlC-17K*) が出力したオブジェクト・モジュール・ファイル (.REL) だけが入力できます。

### 3.2.2 リンク・オプション

リンク・オプションは、リンク処理の実行に先立って必要な情報を指示するためのものです。オプションの記述にエラーがあったときはエラー・メッセージを出力し、LK17Kの実行を中止します。

オプションの記述には大文字/小文字の区別はありません。詳細は3.2.3 リンク・オプションの種類を参照してください。

#### (1) 指定方法

オプションの指定方法には次の2つがあります。

- ・コマンド・ライン上で記述する。
- ・パラメータ・ファイル中に記述する。

#### (2) 記述形式

コマンド・ライン上でオプションを記述する場合については、3.1.1 コマンド・ライン上にオプションを指定する場合を参照してください。

パラメータ・ファイル中にオプションを記述する場合については、3.1.2 パラメータ・ファイルを使用する場合を参照してください。

### 3.2.3 リンク・オプションの種類

リンク・オプションの一覧を次に示します。意味が相反するオプションを同時に指定した場合、最後に指定したオプションが有効となります。

表3-2 リンク・オプション一覧

オプション	内 容	デフォルト	-HOST指定時の解釈	参照ページ
-OBJ[ECT] [=〈ファイル名〉] -NOOB[JECT]	リンク・オブジェクト・モジュール (.LNK) の出力制御	-OBJ	-OBJに強制設定 (指定無視)	50
-ICE [=〈ファイル名〉] -NOIC[E]	ロード・モジュール (.ICE) の出力制御	-ICE	-ICEに強制設定 (指定無視)	50
-PROM [=〈ファイル名〉] -NOP[ROM]	ロード・モジュール (.PRO) の出力制御	-NOP	指定のとおり	51
-HOS[T] -NOH[OST]	SIMPLEHOST情報制御	-NOH	-HOST	52
-WOR[K] =〈パス名〉	作業バス(ドライブ、ディレクトリ)の指定	指定なし	指定のとおり	53
-WAR[NING] =n(0 ≤ n ≤ 15)	ワーニング出力禁止指定	-WAR = 0	指定のとおり	54
-INC[REMENTAL] -NOIN[CREMENTAL]	差分ファイル出力制御	-NOIN	指定のとおり	55
-DIR[ECTIVE] =〈ファイル名〉	ディレクティブ・ファイル指定	指定なし	指定のとおり	55
-PAR[AMETER] =〈ファイル名〉	パラメータ・ファイル指定	指定なし	指定のとおり	56
-PROG = "プログラム・ネーム"	プログラム・ネーム出力制御	指定なし	指定のとおり	57
-IND[RECT] =〈セクション名〉	インダイレクト配置指定	指定なし	指定のとおり	57
-JUN[K] -NOJ[UNK]	リンク・オブジェクト・モジュール(.LNK)の強制出力制御 ただし、-NOOBの場合の-JUNK指定は無効	-NOJ	指定のとおり	58
-LMA[P] [=〈ファイル名〉] -NOL[MAP]	リンク・マップ・ファイル(.LMP)の出力制御	-LMA	指定のとおり	59
-ML -NOML	リンク・マップ・ファイル中ヘローカ ル・シンボル・リストの出力制御 ただし、-NOLの場合の-ML指定は無効	-NOML	指定のとおり	60
-MP -NOMP	リンク・マップ・ファイル中ヘバブリッ ク・シンボル・リストの出力制御 ただし、-NOLの場合の-MP指定は無効	-NOMP	指定のとおり	60

備考 相反するオプションを指定した場合、あとに記述したオプションが有効になります。これは、パラメータ・ファイル中に記述する場合も同様です。オプションとパラメータ・ファイルを併用した場合、次のようになります。

例 オプションに-OBJを指定し、パラメータ・ファイルEX.PLKの中で-NOOBを指定した場合

- ① LK17K -OBJ -PAR = EXと指定した場合：-NOOBが有効
- ② LK17K -PAR = EX -OBJと指定した場合：-OBJが有効



## (1) リンク・オブジェクト・モジュール・ファイル (.LNK) 出力制御

## 【書 式】

-OBJ [ECT] [= <ファイル名> ]  
 -NOOB [JECT]  
 (デフォルト:-OBJ [ECT] )

## 【概 能】

リンク・オブジェクト・モジュール・ファイル (.LNK) の出力を制御します。

## 【説 明】

## (1) -OBJ [ECT] [= &lt;ファイル名&gt; ]

.LNKファイルの出力を指定します。

<ファイル名> で.LNKファイルのファイル名を指定します。

<ファイル名> には、パス名 (ドライブ/ディレクトリ名) も指定できます。

<ファイル名> の指定がない場合、つまり-OBJだけが指定されている場合は (3) のデフォルトが指定されたものとして処理します。パス名だけが指定された場合、指定されたパスに (3) のデフォルトのファイル名で出力します。

## (2) -NOOB [JECT]

.LNKファイルを出力しません。

## (3) オプションが指定されていない場合 (デフォルト)

## ① 入力オブジェクト・モジュール・ファイル (.REL) が複数の場合

出力先     カレント・パス

ファイル名 <第1入力オブジェクト・モジュール・ファイル名> + 拡張子 (.LNK)

## ② 入力オブジェクト・モジュール・ファイル (.REL) が1つの場合

出力先     カレント・パス

ファイル名 <オブジェクト・モジュール・ファイル名> + 拡張子 (.LNK)

## 【ほかのオプションとの関係】

-HOST指定時、-OBJ = <ファイル名>、-NOOBは無効となり、-OBJが指定されたものとして処理します。

## (2) ロード・モジュール・ファイル (.ICE) 出力制御

## 【書 式】

-ICE [= <ファイル名> ]  
 -NOIC [E]  
 (デフォルト:-ICE)

## 【機能】

ロード・モジュール・ファイル (.ICE) の出力を制御します。

## 【説明】

## (1) -ICE [= &lt;ファイル名&gt; ]

.ICEファイルの出力を指定します。

<ファイル名> で .ICEファイルのファイル名を指定します。

<ファイル名> には、パス名 (ドライブ/ディレクトリ名) も指定できます。

<ファイル名> の指定がない場合、つまり -ICE だけが指定されている場合は (3) のデフォルトが指定されたものとして処理します。パス名だけが指定された場合、指定されたパスに (3) のデフォルトのファイル名で出力します。

## (2) -NOIC [E]

.ICEファイルを出力しません。

## (3) オプションが指定されていない場合 (デフォルト)

## ① オブジェクト・モジュール・ファイルが複数の場合

出力先      カレント・パス

ファイル名   <第1入力オブジェクト・モジュール・ファイル名> + 拡張子 (.ICE)

## ② オブジェクト・モジュール・ファイルが1つの場合

出力先      カレント・パス

ファイル名   <オブジェクト・モジュール・ファイル名> + 拡張子 (.ICE)

## 【ほかのオプションとの関係】

(1) -HOST指定時、-ICE = <ファイル名>、-NOICは無効となり、-ICEが指定されたものとして処理します。

(2) -ICE、-NOIC指定により影響を受けるオプションはありません。

## (3) ロード・モジュール・ファイル (.PRO) 出力制御

## 【書式】

-PROM [= <ファイル名> ]

-NOP [ROM]

(デフォルト: -NOP [ROM] )

## 【機能】

ロード・モジュール・ファイル (.PRO) の出力を制御します。

## 【説明】

## (1) -PROM [= &lt;ファイル名&gt; ]

.PROファイルの出力を指定します。

<ファイル名> で、.PROファイルのファイル名を指定します。

<ファイル名> には、パス名（ドライブ/ディレクトリ名）も指定できます。

<ファイル名> の指定がない場合、つまり-PROMだけが指定されている場合は、次の規則に従って.PROファイルを出力します。

## ① オブジェクト・モジュール・ファイルが複数の場合

出力先      カレント・パス

ファイル名    <第1入力オブジェクト・モジュール・ファイル名> + 拡張子 (.PRO)

## ② オブジェクト・モジュール・ファイルが1つの場合

出力先      カレント・パス

ファイル名    <オブジェクト・モジュール・ファイル名> + 拡張子 (.PRO)

パス名だけが指定された場合、指定されたパスに上述の規則に従ったファイル名で出力します。

## (2) -NOP [ROM]

.PROファイルを出力しません。

## (3) オプションが指定されていない場合（デフォルト）

-NOPが指定されたものとして処理します。

## 【ほかのオプションとの関係】

-PROM、-NOPにより影響を受けるオプションはありません。

## (4) SIMPLEHOST 情報制御

## 【書式】

-HOS [T]

-NOH [OST]

（デフォルト：-NOH [OST]）

## 【機能】

17 KシリーズのSIMPLEHOSTを使用するときに必要な情報の出力を制御します。

## 【説明】

## (1) -HOS [T]

SIMPLEHOSTを使用する場合は、このオプションを指定します。

## (2) -NOH [OST]

SIMPLEHOSTを使用しない場合は、このオプションを指定します。

- (3) オプションが指定されていない場合 (デフォルト)

-NOHが指定されたものとして処理します。

【ほかのオプションとの関係】

- (1) -NOHOST指定により影響を受けるオプションはありません。

- (2) -HOST指定により、次のオプションは影響を受けます。

-ICE = 〈ファイル名〉、-NOIC→-ICE (ファイル名指定なし) が設定されているものとして処理します。

-OBJ = 〈ファイル名〉、-NOOB→-OBJ (ファイル名指定なし) が設定されているものとして処理します。

【注意事項】

- (1) リンク時に-HOSTを指定した場合、入力されたオブジェクト・モジュール・ファイル (.REL) 中に1つでも-HOSTを指定せずにアセンブルされたものがあるとエラーになります。
- (2) -HOSTを指定してアセンブルしたオブジェクト・モジュール・ファイル (.REL) を入力ファイルとして指定した場合、リンク時に-HOST指定がなくてもエラーになりません。ただし、この場合にはSIMPLEHOSTは使用できません。

- (5) 作業パス制御

【書 式】

-WOR [K] = 〈パス名〉

【機 能】

リンク実行中に使用するテンポラリ・ファイルを確認するパス名 (ドライブ/ディレクトリ名) を指定します。

環境変数TMPが指定されている場合でも、-WORKの指定が優先されます。

【説 明】

- (1) ドライブ名は1つだけ指定できます。
- (2) ドライブ名だけ指定したとき、テンポラリ・ファイルは指定ドライブのカレント・ディレクトリに作成されます。
- (3) ドライブ名に続きディレクトリ名を指定した場合、指定したディレクトリにテンポラリ・ファイルが作成されます。
- (4) ディレクトリ名だけ指定した場合、テンポラリ・ファイルはカレント・ドライブの指定されたディレクトリに作成されます。

(5) 作業パスは、次の優先順位に従って決定します。

- ① 作業パス指定オプション (-WORK) で指定されたドライブ、ディレクトリ
- ② 環境変数TMPに指定されたドライブ、ディレクトリ
- ③ カレント・ドライブ、カレント・ディレクトリ

(6) 作業パスには、RAMディスクなどの高速なファイル装置を指定しておくことをお勧めします。

【ほかのオプションとの関係】

-WORK指定により影響を受けるオプションはありません。

【注意事項】

- (1) 指定したドライブ名、またはディレクトリ名がない場合、エラーが発生します。
- (2) リンク終了後、テンポラリ・ファイルはすべて消去されます。

(6) ワーニング出力禁止制御

【書 式】

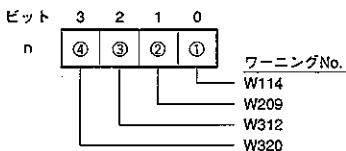
-WAR [NING] = n  
(デフォルト：-WAR = 0)

【機 能】

リンク実行中に出力される各種ワーニングの出力の可否を指定します。

【説 明】

- (1) 数値 n を 2 進数表現した場合に “1” になるビットに対応するワーニングが出力されなくなります。



- (2) 数値 n として指定できるのは、2進、10進、16進の数値です。

【指定例】

-WAR = 2 : W209のワーニングをチェックしません

## 【ほかのオプションとの関係】

-WAR指定により影響を受けるオプションはありません。

## 【注意事項】

数値  $n$  に不正な値 ( $n < 0$  または  $15 < n$ ) を記述するとエラーが発生します。

## (7) 差分ファイル出力制御

## 【書 式】

-INC [REMENTAL]

-NOIN [CREMENTAL]

(デフォルト: -NOIN [CREMENTAL])

## 【機 能】

インクリメンタル・ロード処理のための差分ファイル (DIF) の出力を制御します。

インクリメンタル・ロードについては第I編 2.6 インクリメンタル・ロード用差分ファイルを参照してください。

## 【説 明】

## (1) -INC [REMENTAL]

新/旧.ICEファイルの差分を取った差分ファイルを出力します。

## (2) -NOIN [CREMENTAL]

差分ファイルを出力しません。

## (3) オプションが指定されていない場合 (デフォルト)

-NOINが指定されたものとして処理します。

## 【ほかのオプションとの関係】

-INC、-NOIN指定により影響を受けるオプションはありません。

## 【注意事項】

(1) -INCが指定された場合で.LNKファイルがない場合、エラーになりリンク処理を中止します。

(2) -INCと同時に-NOOBか-NOICを指定するとエラーになります。

## (8) ディレクティブ・ファイル指定

## 【書 式】

-DIR [EACTIVE] = [ <バス名> ] <ファイル名>

**【機能】**

指定されたファイルをディレクティブ・ファイルとして入力します。

ディレクティブ・ファイルの詳細については、第I編 2.1.2 ディレクティブ・ファイルを参照してください。

**【説明】**

〈ファイル名〉には、ディレクティブ・ファイルとして使用するファイル名を指定します。

また、パス名（ドライブ/ディレクトリ名）も指定できます。

拡張子が省略された場合には、.DRが指定されたものとして処理します。

**【ほかのオプションとの関係】**

-DIR指定により影響を受けるオプションはありません。

**【注意事項】**

(1) 存在しないファイル名が指定されたときは、アボート・エラーになります。

(2) 〈ファイル名〉が指定されていない場合は、エラーになります。

**(9) パラメータ・ファイル指定****【書式】**

-PAR [AMETER] = [ 〈パス名〉 ] 〈ファイル名〉

**【機能】**

オプションや入力ファイル名をパラメータ・ファイルから入力するよう指示します。

このオプションには、次のような用途があります。

- ・コマンド・ラインでは、LK17Kの起動に必要な情報を指定しきれないときに、このオプションを指定します。
- ・リンクするたび繰り返し同じようにオプションを指定する場合には、それらをパラメータ・ファイルに記述しておき、このオプションで指定します。

**【説明】**

(1) 〈ファイル名〉には、パラメータ・ファイルとして使用するファイル名を指定します。

また、パス名（ドライブ/ディレクトリ名）も指定できます。

拡張子が省略された場合には、.PLKが指定されたものとして処理します。

存在しないファイル名が指定された場合は、アボート・エラーになります。

(2) ファイル名が指定されていない場合は、アボート・エラーになります。

- (3) パラメータ・ファイルのネストはできません。パラメータ・ファイル中に-PARが指定されている場合は、アボート・エラーになります。
- (4) “;” か “#” 以降の文字は、CRまたはEOFまですべてコメントと解釈します。
- (5) -PARが複数指定された場合は、アボート・エラーになります。

**【ほかのオプションとの関係】**

-PAR指定により影響を受けるオプションはありません。

**(10) プログラム・ネーム出力制御****【書 式】**

-PROG= "プログラム・ネーム"

**【機 能】**

指定された文字列（プログラム・ネーム）をロード・モジュール・ファイル中に出力します。

ドキュメント・プロセッサ（DOC17K）が出力する絶対番地リスト、ローカル・クロスレファレンス・リスト、パブリック・クロスレファレンス・リスト、マップ・リスト、レポート・リストの各プログラム・ネーム欄へ出力する文字列を指定します。

**【説 明】**

- (1) 〈プログラム・ネーム〉には最大127文字まで記述できます。ただしロード・モジュール・ファイルには、先頭の32バイト分だけが出力されます。  
128文字以上を記述した場合はエラーとなり、リンクを中止します。
- (2) このオプションを省略したときは、プログラム・ネーム欄には何も出力されません。

**【ほかのオプションとの関係】**

-PROG指定により影響を受けるオプションはありません。

**(11) インダイレクト配置指定****【書 式】**

-IND [IRECT] = 〈セクション名〉

**【機 能】**

再配置属性VECTn (n = max) が指定されたセクション・ブロックをインダイレクトで配置します。



## 【説明】

(1) 再配置属性VECTn (n=max) が指定されたセクション・ブロックは、デフォルトではセグメント0のn番地 (PAGE0) にダイレクトに配置されます。-INDを指定すると、セグメント0のPAGE0のn番地にBR addr命令を生成し、再配置属性VECTn(n=max) が指定されたセクション・ブロックはセグメント0内の任意のアドレスに配置します (インダイレクト配置)。

サブルーチンの分岐先は、必ずセグメント0のPAGE0内に配置されるため、-INDを指定してセグメント0のPAGE0の空き領域を増すことで、より多くのサブルーチンをセグメント0のPAGE0内に配置できるようになります。

(2) -INDを指定するとBR addr命令を生成するため、オブジェクト・コードが長くなります。

## 【ほかのオプションとの関係】

-IND指定により影響を受けるオプションはありません。

## (12) リンク・オブジェクト・ファイル強制出力制御

## 【書式】

-JUN [K]

-NOJ [UNK]

(デフォルト:-NOJ [UNK])

## 【機能】

フェータル・エラー発生時でも、強制的にリンク・オブジェクト・ファイル (.LNK) を出力します。ただし、同時に-NOOBが指定されている場合は出力しません。

## 【説明】

(1) -JUN [K]

フェータル・エラー発生時でも、リンク・オブジェクト・ファイルを出力します。ただし-NOOBが指定されているときは、-JUN指定は無効となります。

(2) -NOJ [UNK]

フェータル・エラー発生時には、リンク・オブジェクト・ファイルを出力しません。

(3) オプションが指定されていない場合 (デフォルト)

-NOJが指定されたものとして処理します。

## 【ほかのオプションとの関係】

-NOOBが指定されている場合は、-JUNは無効になります。

## (13) リンク・マップ・ファイル (.LMP) 出力制御

## 【書式】

-LMA [P] [= <ファイル名> ]  
-NOL [MAP]  
(デフォルト: -LMA [P] )

## 【機能】

リンク・マップ・ファイル (.LMP) の出力を制御します。

## 【説明】

## (1) -LMA [P] [= &lt;ファイル名&gt; ]

.LMPファイルの出力を指定します。

<ファイル名> により、.LMPファイルのディレクトリやファイル名を指定します。

<ファイル名> の指定がない場合、つまり-LMAだけが指定されている場合は、(3) のデフォルトが指定されたものとして処理します。パス名だけが指定された場合、指定されたパスに(3) のデフォルトのファイル名で出力します。

## (2) -NOL [MAP]

リンク・マップ・ファイルを出力しません。

## (3) オプションが指定されていない場合 (デフォルト)

## ① オブジェクト・モジュール・ファイルが複数の場合

出力先      カレント・パス

ファイル名   <第1入力オブジェクト・モジュール・ファイル名> +拡張子 (.LMP)

## ② オブジェクト・モジュール・ファイルが1つの場合

出力先      カレント・パス

ファイル名   <オブジェクト・モジュール・ファイル名> +拡張子 (.LMP)

## 【ほかのオプションとの関係】

-LMA、-NOL指定により影響を受けるオプションはありません。

## 【注意事項】

-LMAに対する <ファイル名> には、パス名 (ドライブ/ディレクトリ名) も指定できます。

(14) ローカル・シンボル・リスト出力制御

**【書 式】**

-ML

-NOML

(デフォルト:-NOML)

**【機 能】**

ローカル・シンボル・リストの出力を制御します。

なお、ローカル・シンボル・リストはリンク・マップ・ファイル (.LMP) 中に出力されます。

**【説 明】**

(1) -ML

.LMPファイル中にローカル・シンボル・リストを出力します。

(2) -NOML

.LMPファイル中にローカル・シンボル・リストを出力しません。

(3) オプションが指定されていない場合 (デフォルト)

-NOMLが指定されたものとして処理します。

**【ほかのオプションとの関係】**

-NOLが指定されている場合は、-MLは無効になります。

(15) パブリック・シンボル・リスト出力制御

**【書 式】**

-MP

-NOMP

(デフォルト:-NOMP)

**【機 能】**

パブリック・シンボル・リストの出力を制御します。

なお、パブリック・シンボル・リストは、リンク・マップ・ファイル (.LMP) 中に出力されます。

**【説明】**

(1) -MP

.LMPファイル中にパブリック・シンボル・リストを出力します。

(2) -NOMP

.LMPファイル中にパブリック・シンボル・リストを出力しません。

(3) オプションが指定されていない場合 (デフォルト)

-NOMPが指定されたものとして処理します。

**【ほかのオプションとの関係】**

-NOLが指定されている場合は、-MPは無効になります。

## 3.3 出力

LK17Kが出力するものには、次のものがあります。

### ・メッセージ

- 通知メッセージ
  - 実行開始メッセージ
  - ヘルプ・メッセージ
  - 終了メッセージ
- エラー・メッセージ

### ・出力ファイル

- ロード・モジュール・ファイル (.ICE)
- ロード・モジュール・ファイル (.PRO)
- リンク・オブジェクト・モジュール・ファイル (.LNK)
- ログ・ファイル (LK17K.LOG)
- 差分ファイル (.DIF)
- リンク・マップ・ファイル (.LMP)

### 3.3.1 出力メッセージ

LK17Kが出力するメッセージには、次の2つがあります。

- ・通知メッセージ
- ・エラー・メッセージ

#### {1} 通知メッセージ

通知メッセージには、次の3つがあります。

- ・実行開始メッセージ
- ・ヘルプ・メッセージ
- ・終了メッセージ

#### (a) 実行開始メッセージ

リンク処理が開始されたことを知らせるために出力されるメッセージです。

メッセージの形式は、3.1.3 実行開始メッセージを参照してください。

## (b) ヘルプ・メッセージ

パラメータを指定せずにLK17Kを起動した場合に、出力されるメッセージです。

内容はLK17Kの使い方とオプションの簡単な説明です。メッセージの形式は3.1.4 ヘルプ・メッセージを参照してください。

## (c) 終了メッセージ

LK17Kの処理が終了したことを知らせるために出力されるメッセージです。

終了メッセージには、正常終了メッセージと異常終了メッセージがあります。

## (i) 正常終了メッセージ

リンクが正常に行われた場合に出力されます。

メッセージの形式は、3.1.5 プログラムの終了を参照してください。

## (ii) 異常終了メッセージ

LK17Kの実行が不可能となる次のような場合に出力されます。

- ・ファイル名やオプション指定に致命的エラーがあったとき
- ・致命的なファイル/IOエラーが発生したとき
- ・LK17Kの処理中にエラーが発生したとき

メッセージの形式は、3.1.5 プログラムの終了を参照してください。

## (2) エラー・メッセージ

エラー・メッセージは、次の形式で出力します。各エラー・メッセージは、ログ・ファイル(LK17K.LOG)にも出力されます。

## 【形 式】

## ① ディレクティブ・ファイル中のエラーの場合

入力ファイル名(行番号) △:△エラー番号△:△エラー・メッセージ
-----------------------------------

## ② 上記以外の場合

error△エラー番号△:△エラー・メッセージ
-------------------------

## 【説明】

## ●入力ファイル名

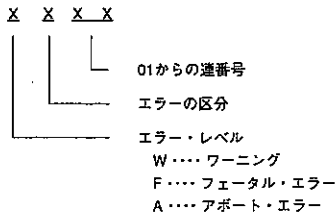
- ・英小文字で入力されたファイル名は、すべて大文字に変換して出力します（日本語文字の場合は変換しません）。
- ・入力されたパス名は、指定されたとおりに出力します（省略時解釈などの内容は出力しません。）
- ・インクルード・ファイル中にエラーがある場合、インクルード・ファイル名とその行番号を出力します。

## ●行番号

- ・行番号は左詰めで出力します。

## ●エラー番号

- ・エラー番号は次の4文字で表します。



## ●エラー・メッセージ

詳細は第4章 エラー・メッセージを参照してください。

## 3.3.2 出力ファイル

LK17Kが出力するファイルには次のものがあります。

- ロード・モジュール・ファイル (.ICE) : インサーキット・エミュレータのロード用
- ロード・モジュール・ファイル (.PRO) : コード受注とOTP製品の書き込み用  
 実行可能なロード・モジュールとして、ICEファイルと、PROファイルを出力します。  
 .ICEファイルと、PROファイルの出力形式は、拡張インテルHEX形式です。

LK17Kでは、ロード・モジュール・ファイルを出力するまでに使用したツール (emIC-17K, RA17K, LK17K) とデバイス・ファイルのバージョンをチェックします。その結果、どれか1つでも正式リリース版でない場合は、ワーニング (W115) を出力します。

## ○リンク・オブジェクト・モジュール・ファイル (.LNK)

このファイルは、リンク結果をバイナリ形式で出力したものです。ドキュメント・プロセッサ (DOC17K) とSIMPLEHOSTの入力ファイルとして使用します。

## ○ログ・ファイル (LK17K.LOG)

実行中に画面に表示するメッセージを格納したファイルです。リンク開始、終了時間も出力します。

## ○差分ファイル (.DIF)

インクリメンタル・ロード実行時に使用する新/旧.ICEファイルの差分ファイルです。

## ○リンク・マップ・ファイル (.LMP)

リンク実行後の各セクションの配置情報、ローカル・シンボル・リスト、パブリック・シンボル・リストを格納したファイルです。

## (1) ロード・モジュール・ファイル (.ICE) の出力形式

.ICEファイルは2つの部分から構成されます。

## ・オブジェクト領域

ソース・プログラムのオブジェクトで、正規のプログラム領域を超えた部分 (パッチ領域) もデバッグ用に出力します。

出力は拡張インテルHEX形式で、オブジェクトが最大256 Kバイト (3FFFFH) になる場合までサポートします。

17Kシリーズでは、ROM128 Kバイトの製品まで対応できます (残りの128 Kバイトはパッチ領域として使用)。

## ・デバッグ情報領域

SIMPLEHOSTまたはIE-17Kのデバッグ用のデータを出力します。

.ICEファイルの出力形式を図3-1に示します。なお、図中のEARとEND RECORDの内容は次のとおりです。

・EAR: 拡張アドレス・レコード (Extended Address Record)

020000021000EC (EAR 1000の場合)

①      ②      ③      ④

- |             |                             |
|-------------|-----------------------------|
| ①: データ数     | (単位: バイト)                   |
| ②: レコード・タイプ | (02で拡張アドレス・レコード)            |
| ③: データ      | (EARのデータで、オフセットの値。図中のEARの値) |
| ④: チェック・サム  |                             |



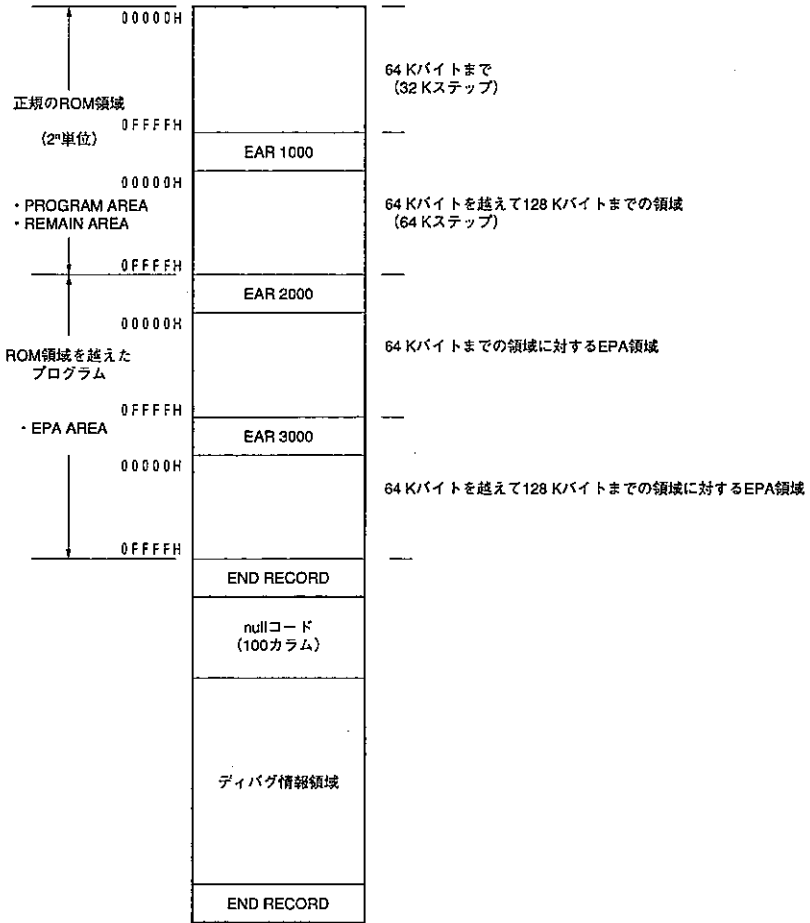
・ END RECORD：オブジェクト領域とディバグ情報領域のデータの終了を示す

0000001FF

①            ② ③

- ①：データ数                    (単位：バイト)
- ②：レコード・タイプ        (01で最終レコード)
- ③：チェック・サム

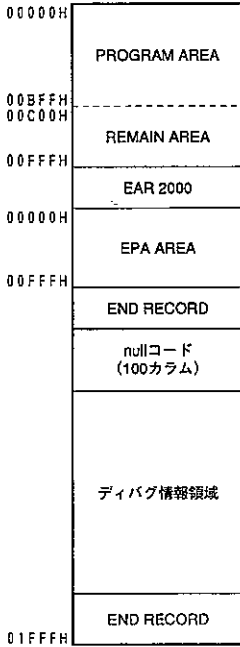
図3-1 JICEファイルのコード出力形式



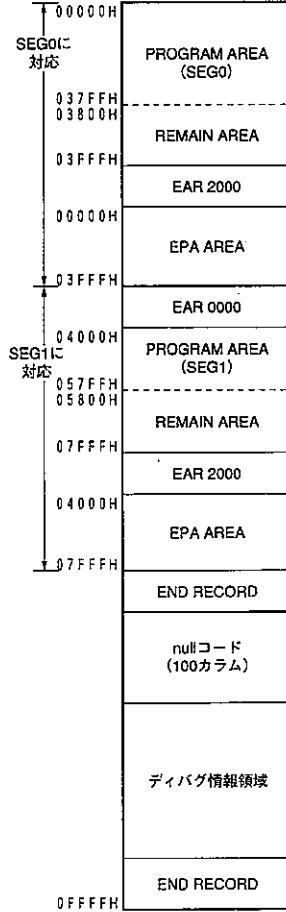
注意 JICEファイル中のアドレス範囲は、17Kシリーズの各製品ごとに異なります。詳細は各製品に対応するデバイス・ファイルのユーザーズ・マニュアルを参照してください。

【.ICEファイルの出力例】

例1. PROGRAM領域が  
1.5 Kステップのデバイスの場合



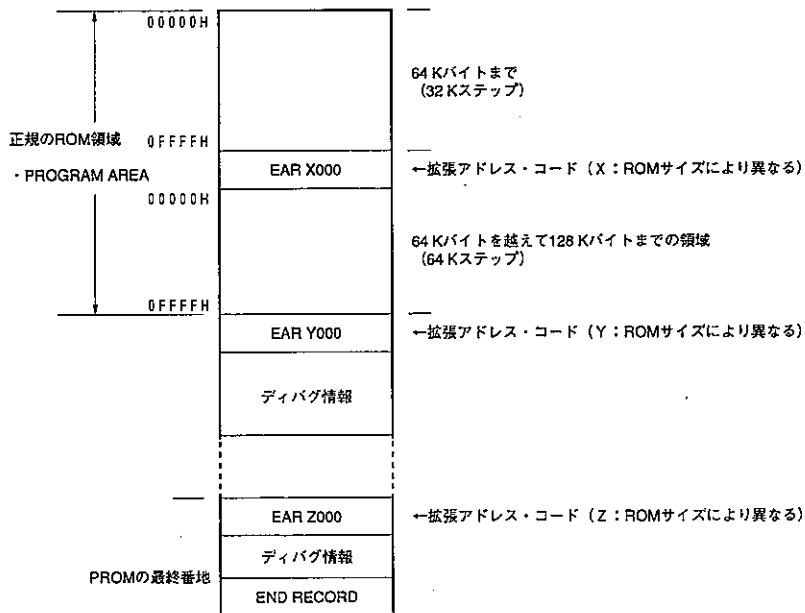
例2. PROGRAM領域が  
SEG0 7 Kステップ  
SEG1 3 Kステップのデバイスの場合



(2) ロード・モジュール・ファイル (.PROファイル) の出力形式

.PROファイルは、プログラム領域のオブジェクト・コードとデバッグ情報が出力された拡張インテルHEX形式のファイルです。デバッグ時は.PROファイルをUVPROMに書き込み、SEボードに実装して使用します。またOTP製品に書き込んで、システム・デバッグに使用します。

図3-2 PROMファイルのコード出力形式



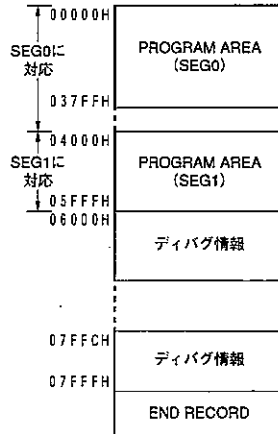
注意 .PROファイル中のアドレス範囲は、17Kシリーズの各製品ごとに異なります。詳細は各製品に対応するデバイス・ファイルのユーザーズ・マニュアルを参照してください。

【.PROファイルの出力例】

例1. プログラム領域がSEG0以内  
6Kステップ (12 Kバイト) の場合



例2. プログラム領域が  
SEG0→7Kステップ (14 Kバイト)  
SEG1→4Kステップ (8 Kバイト) の場合

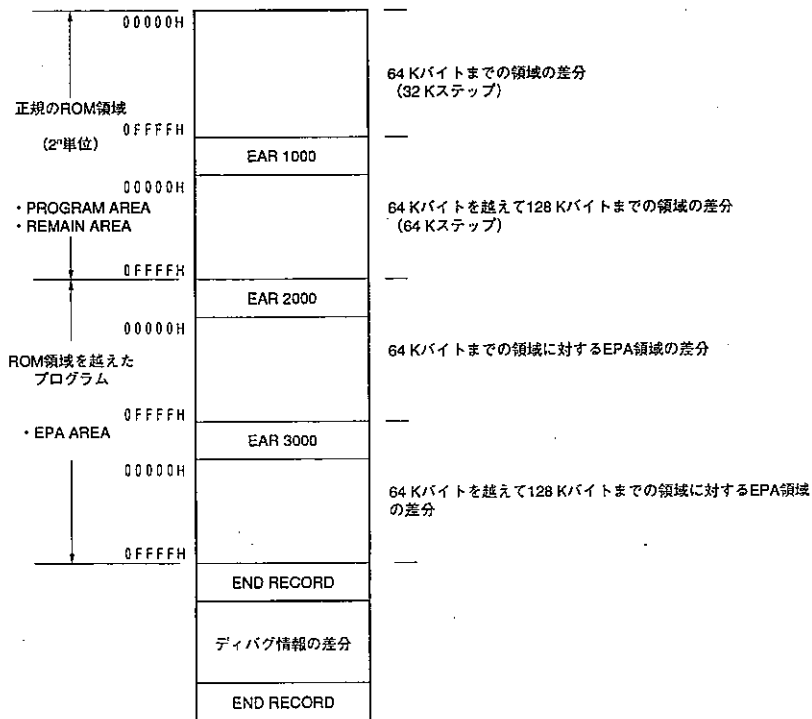


(3) 差分ファイル(DIF) の出力形式

差分ファイルには、.ICEファイルの中のオブジェクトの差分と、デバッグ情報の差分が出力されま  
す。

各差分の出力領域の最後には、必ずEND RECORDが出力されます。つまりEND RECORDは、各領  
域の差分がない場合にも出力されます。また、拡張アドレス・レコード (EAR) も必要に応じて出力さ  
れます。

図 3-3 差分ファイルのコード出力形式



## (4) リンク・マップ・ファイル

LK17Kはリンクした結果をユーザに知らせるため、リンク・マップ・ファイルを出力します。  
リンク・マップ・ファイルは次の5つの部分に分かれ、この順に出力します。

- (a) ID欄
- (b) マップ・リスト
- (c) ローカル・シンボル・リスト
- (d) パブリック・シンボル・リスト
- (e) アロケートできなかったセクション・リスト

表3-3 リンク・マップ・ファイル

リンク・マップ・ファイル	出力指定オプション	デフォルト	出力順序
ID欄	-LMAP	出力する	1
マップ・リスト	-LMAP	出力する	2
ローカル・シンボル・リスト	-ML	出力しない	3
パブリック・シンボル・リスト	-MP	出力しない	4
アロケートできなかったセクション・リスト	-LMAP	出力する	5

次にリンク・マップ・ファイルのレイアウトの詳細について説明します。

## (a) 全体図

リンク・マップ・ファイルの全体図を次に示します。

図3-4 リンク・マップ・ファイルの全体図

LK17K V1.00 <<D17xxx V1>> HH:MM:SS YY/MM/DD PAGE nnn

Command: TEST1 TEST2 -ML -MP  
 Para-file:  
 Out-file: TEST1.LNK  
 Map-file: TEST1.LMP  
 Direc-file:  
 Directive:  
 Dev-file: J:¥D17xxx.DEV

\*\*\* Memory map \*\*\*

ROM AREA : xxxxH - xxxxH

OUTPUT SECTION	INPUT SECTION	INPUT MODULE	START ADDR	END ADDR	ATTR	ROM
SEC1			0H	10H	SEC	PRO
	SEC1	TEST1	0H	10H	SEC	PRO
* gap *			11H	1fffH		

LK17K V1.00 <<D17xxx V1>> HH:MM:SS YY/MM/DD PAGE nnn

\*\*\* Local symbol list \*\*\*

MODULE	ATTR	VALUE	SYMBOL NAME
TEST1	MEM	1H	MEM1
TEST2	MEM	2H	MEM2

LK17K V1.00 <<D17xxx V1>> HH:MM:SS YY/MM/DD PAGE nnn

\*\*\* Public symbol list \*\*\*

MODULE	ATTR	VALUE	SYMBOL NAME
TEST1	DAT	5H	DAT1

LK17K V1.00 <<D17xxx Vn>> HH:MM:SS YY/MM/DD PAGE nnn

\*\*\* Not allocated section \*\*\*

TYPE	SIZE	SECTION
AT	10H	SEC2



(b) ページ

1 ページ当たりの行数は66行です。

1 行当たりの文字数は最大72文字で、それ以上の表示が必要な場合は折り返し処理をします。

(c) ヘッダ部

^L

LK17K V1.00 <<D17xxx Vn>>

HH:MM:SS YY/MM/DD PAGE nnn

表3-4 ヘッダ部の内容

No.	表示項目	桁数	表示方法	表示内容
1	D17xxx	最大 8	桁可変	使用したデバイス・ファイルのファイル名を表示する。
2	Vn	1	桁固定	使用したデバイス・ファイルのバージョンを表示する。
3	HH:MM:SS YY/MM/DD	17	右記	リンク・マップ・ファイルを作成した時分秒と年月日を表示する。 HH:時 (0-12, 右詰め0サプレス) MM:分 (0-59, 右詰め0サプレス) SS:秒 (0-59, 右詰め0サプレス) DD:日 (1-31, 右詰め0サプレス) MM:月 (1-12, 右詰め0サプレス) YY:年 (94-, 西暦下2桁表示)
4	PAGE nnn	3	右詰め0サプレス	リンク・マップ・ファイル内のページ数を表示する。 初期値=1, ページ替えごとに+1した値を表示する (10進)。

① 各ページの仕様を次に示します。

(a) 上から3行あける (空行とする)。

(b) 4行目に上記のタイトルを表示する。No.3に表示する時分秒、年月日はOSから取得する。

(c) 5行目は空行とする。

(d) 6行目から63行目までにマップ・リストなどの本体を出力する。

(e) 64行目に改ページ・コードを出力する。

② リンク・マップ、ローカル・シンボル・リスト、パブリック・シンボル・リスト、アロケートできなかったセクション・リストの終了時、最後の出力行の改行文字の直後に改ページ・コード1文字とEOF文字 (ascii code 1AH) 1文字を出力します。

③ デバイス・ファイル名は8桁可変長とし、左詰めで表示します。

## (d) ID部

Command: TEST1 TEST2 -ml -mp  
 Para-file:  
 Out-file: TEST1.LNK  
 Map-file: TEST2.LMP  
 Direc-file:  
 Directive:  
 Dev-file: J:\D17xxx.DEV

表 3-5 ID部の内容

No.	表示項目	桁数	表示方法	表示内容
1	Command:	61	左詰め	コマンド・ラインに指定されたファイル名やオプションを、それぞれ1つの空白で区切って表示する。
2	Para-file:			パラメータ・ファイルの内容をそのまま表示する。
3	Out-file:			LK17Kが生成する出力ファイル名を表示する。
4	Map-file:			LK17Kが生成するリンク・マップ・ファイル名を表示する。
5	Direc-file:			LK17Kが入力するディレクティブ・ファイル名を表示する。
6	Directive:			ディレクティブ・ファイルの内容をそのまま表示する。
7	Dev-file:			LK17Kが使用したデバイス・ファイル名をフルバス付きで表示する。

- ① 各表示欄で表示する内容がない場合は、見出しだけを表示する。このとき“:”のあとに空白は出力しません。
- ② 各表示欄で表示内容が1行に表示しきれない場合は、72コラムまで表示した残りの部分を、次行の12コラムから出力します。出力が2行以上にわたる場合、2行目以降の1コラム目から11コラム目までは空白を出力します。
- ③ コマンド・ラインに記述された入力ファイルとオプションを、No.1に表示します。コマンド・ラインで指定したオプションのあとに余分な空白が記述されているときは、その空白を無視します。
- ④ パラメータ・ファイルの内容をNo.2に表示します。パラメータ・ファイルに含まれるタブは1つの空白文字に展開し、大文字/小文字はそのまま出力します。不正文字 (00H-08H,0BH,0CH,0EH-19H,1BH-1FH,7FH) は“!”に置き換えて表示します。
- ⑤ 出力するロード・モジュール・ファイル名をNo.3に表示します。このときロード・モジュール・ファイルをオープンするときに使用したバスを付加して表示します。
- ⑥ ディレクティブ・ファイルの内容をNo.6に表示します。ディレクティブ・ファイルに含まれるタブは12コラム目から数えて8文字単位になるように空白文字に展開します。
- ⑦ デバイス・ファイル名をNo.7に表示します。LK17Kがどこのデバイス・ファイルを使用したかが分かるようにフルバスを付加して出力します。

(e) メモリ・マップ

\*\*\* Memory map \*\*\*

ROM AREA : xxxxH - yyyyH

OUTPUT SECTION	INPUT SECTION	INPUT MODULE	START ADDR	END ADDR	ATTR	ROM
SEC1			0H	10H		PRO
	SEC1	TEST1	0H	10H	SEC	PRO
* gap *			11H	1FH		

表3-6 メモリ・マップの内容

No.	表示項目	桁数	表示方法	表示内容
1	ROM AREA :	4	右詰め0サプレス	デバイスのROMのスタート・アドレスとエンド・アドレスを表示する。
2	OUTPUT SECTION	18	左詰め	LK17Kが出力するセクション名を表示する。
3	INPUT SECTION	18	左詰め	LK17Kに入力されたセクション名を表示する。
4	INPUT MODULE	8	左詰め	LK17Kに入力されたモジュール名 (ファイル名) を表示する。
5	START ADDR	6	右詰め0サプレス	セクションが配置された先頭アドレスを表示する (16進)。
6	END ADDR	6	右詰め0サプレス	セクションが配置された最終アドレスを表示する (16進)。
7	ATTR	5	左詰め	セクションのタイプを表示する。
8	ROM	5	左詰め	セクションが配置された領域がプログラム領域かEPA/REMAIN領域かを表示する。

- ① ID欄のデバイス・ファイル名の表示が終わったあと、空行2行を出力し、次の行の1カラム目から見出し“\*\*\* Memory map \*\*\*”を表示します。
- ② 1行空行を出力したあと、メモリ・マップの見出しを表示します。
- ③ 見出しの表示後、1行空白を表示したあと、ROMエリアを表示します。
- ④ No. 2には、LK17Kが出力するセクション名を表示します。No. 3にはNo. 2のセクションに対する入力セクション名を表示します。表示の内容については表3-8を参照してください。
- ⑤ No. 4には、No. 3に出力したセクションのモジュール名を表示します。
- ⑥ No. 5、6はNo. 2、3のセクションが配置された先頭アドレスと終了アドレスを表示します。
- ⑦ No. 7はNo. 2、3のセクションのタイプを表示します。表示の内容については表3-7を参照してください。
- ⑧ No. 8はNo. 2、3のセクションが配置されたメモリ領域がPROGRAM/REMAIN/EPAのどれかを表示します。ただしREMAIN領域の場合、そのセクションの一部がREMAIN領域に入っていればREMAINの表示を出力します。表示の内容については表3-9を参照してください。

表3-7 セクション・タイプ表示方法

表示内容	説明
AT	AT属性のセクション
BOOTT	BOOT属性に対する分枝テーブル
CROM	CROM属性のセクション
DVECT	DVECTn属性のセクション
VECTT	VECTn属性に対する分枝テーブル
DSYS	DSYS属性のセクション
SYST	SYS属性に対する分枝テーブル
SYS	SYS属性のセクション
DSBR	DSBR属性のセクション
SBRT	SBR属性に対する分枝テーブル
SBR	SBR属性のセクション
BOOT	BOOT属性のセクション
VECT	VECTn属性のセクション
TABLE	TABLE属性のセクション
SEC	指定なしのセクション

表3-8 セクション名表示方法

表示内容	説明
セクション名	ユーザの指定したセクション名を表示する。セクション名が19文字以上のときは18文字目に継続を示す記号“*”を出力し、18文字目以降は表示しない。
* gap *	セクションが配置されていないメモリ領域に対して表示する。表示される場所は、No.1 だけである。

表3-9 配置領域表示方法

表示内容	説明
PRO	PROGRAM領域
REM	REMAIN領域
EPA	EPA領域

(f) ローカル・シンボル・リスト

^L

LK17K V1.00 <<D17xxx Vn>>

HH:MM:SS YY/MM/DD PAGE nnn

\*\*\* Local symbol list \*\*\*

MODULE	ATTR	VALUE	SYMBOL NAME
TEST1	MEM	1H	MEM1
TEST2	MEM	2H	MEM2

表3-10 ローカル・シンボル・リストの内容

No.	表示項目	桁数	表示方法	表示内容
1	MODULE	8	左詰め	ローカル・シンボルが定義された入力オブジェクト・モジュール名を表示する。
2	ATTR	5	左詰め	ローカル・シンボルの属性を表示する。
3	VALUE	9	右詰め0サプレス	ローカル・シンボルの値を表示する(16進)。注
4	SYMBOL NAME	50	左詰め	ローカル・シンボル名を表示する。

- ① 前項で述べたマップ・リストを出力したあと、改ページします。
- ② 4行目に(c)ヘッダ部で述べたタイトルを出力したあと、1行空行を出力します。
- ③ “\*\*\* Local symbol list \*\*\*”を表示し、空行を1行出力したあと、ローカル・シンボル・リストの表示項目の見出しを出力します。

- ④ 1行空行を出力し、10行目からローカル・シンボル・リストを出力します。シンボル属性の表示方法は表3-11を参照してください。
- ⑤ シンボル名が50文字を越える場合、50文字目に継続を示す記号“\*”を出力し、50文字目以降は表示しません。
- ⑥ ローカル・シンボル・リストの出力が1ページで納まらない場合、2ページ目以降については6行目に見出しを出力し、1行空行を出力したあと、8行目から続きを出力します。

注 メモリ型シンボルの場合“.”を使用した表示はせず、連続した数字で表示します。

例 1.23H → 123H

表3-11 シンボル属性表示方法

表示内容	説明
DAT	データ型シンボル
MEM	メモリ型シンボル
LAB	レーベル型シンボル
FLG	フラグ型シンボル

(g) パブリック・シンボル・リスト

^L

LK17K V1.00 <<D17xxx Vn>>

HH:MM:SS YY/MM/DD PAGE nnn

\*\*\* Public symbol list \*\*\*

MODULE	ATTR	VALUE	SYMBOL NAME
TEST1	DAT	1H	DAT1

表3-12 パブリック・シンボル・リストの内容

No.	表示項目	桁数	表示方法	表示内容
1	MODULE	8	左詰め	パブリック・シンボル定義された入力オブジェクト・モジュール名を表示する。
2	ATTR	5	左詰め	パブリック・シンボルの属性を表示する。
3	VALUE	9	右詰め0サプレス	パブリック・シンボルの値を表示する(16進)。注
4	SYMBOL NAME	50	左詰め	パブリック・シンボル名を表示する。

- ① 前項で述べたマップ・リストを出力したあと、改ページします。
- ② 4行目に(c)ヘッダ部で述べたタイトルを出力したあと、1行空行を出力します。
- ③ “\*\*\* Public symbol list \*\*\*”を表示し、空行を1行出力したあと、パブリック・シンボル・リストの表示項目の見出しを出力します。

- ④ 1行空行を出力し、10行目からパブリック・シンボル・リストを出力します。シンボル属性の表示方法は表3-13を参照してください。
- ⑤ シンボル名が50文字を越える場合、50文字目に継続を示す記号“\*”を出力し、50文字目以降は表示しません。
- ⑥ パブリック・シンボル・リストの出力が1ページで納まらない場合、2ページ目以降については6行目に見出しを出力し、1行空行を出力したあと、8行目から続きを出力します。

注 メモリ型シンボルの場合“.”を使用した表示はせず、連続した数字で表示します。

例 1.23H → 123H

注意 パブリック・シンボル・リストに出力される情報は、定義側の情報です。参照側の情報は出力しません。

表3-13 シンボル属性表示方法

表示内容	説明
DAT	データ型シンボル
MEM	メモリ型シンボル
LAB	レーベル型シンボル
FLG	フラグ型シンボル
FUNC	emlC-17Kが関数の先頭を示すために出力したレーベル

(h) アロケートできなかったセクション・リスト

```

^L
LK17K V1.00 <<D17xxx Vn>>                               HH:MM:SS YY/MM/DD PAGE nnn

*** Not allocated section ***

TYPE      SIZE      SECTION
AT        10H      SEC2
    
```

表3-14 アロケートできなかったセクション・リストの内容

No.	表示項目	桁数	表示方法	表示内容
1	TYPE	5	左詰め	アロケートできなかったセクションのタイプを表示する。 表示の内容は(f)ローカル・シンボル・リストと同じ。
2	SIZE	6	右詰め0サプレス	アロケートできなかったセクションのサイズを表示する(16進)。
3	SECTION	59	左詰め	アロケートできなかったセクション名を表示する。

- ① 前項で述べた各リストを出力したあと、改ページを行う。
- ② 4行目に(c)ヘッダ部で述べたタイトル行を出力したあと、1行空白行を出力し、6行目に見出し“Not

allocated section” を出力します。この次に1行空行を出力し、8行目に表示項目の見出しを出力します。

- ③ 1行空行を出力して、10行目からアロケートできなかったセクション・リスト (No.1-3) を出力します。
- ④ セクション名が59文字を越える場合、59文字目に継続を示す記号 “\*” を出力し、59文字目以降は表示しません。
- ⑤ アロケートできなかったセクション・リストの出力が1ページで納まらない場合、2ページ目以降については、6行目に表示項目の見出しを出力し、1行空行を出力したあと、8行目からNo.1-3の続きを出力します。
- ⑥ このリストは、アロケートできなかったセクションがあった場合だけ出力します。

## 第4章 エラー・メッセージ

エラー・メッセージの番号は、次のように分類しています。

000台：ファイル指定

    オプション指定

100台：オブジェクト・モジュール・ファイルの入力

    ディレクティブ・ファイルの入力/解釈

    リンク実行の準備（テンポラリ・ファイルのオープン）

200台：セクションの結合、配置

300台：リロケーションの解決

400台：シンボルの解決

500台：ホスト・マシンの環境

900台：ファイル I/O



エラー・メッセージ (A001)	和文: “入力ファイル指定誤り” ----- 英文: “Missing input file”					
原因	入力ファイルが1つも指定されていません。					
プログラムの処理	プログラムの実行を中止します。					
ファイルの出力	-JUNK指定 あり	リンク・オブジェクト ---	ロード・モジュール ---	-JUNK指定 なし	リンク・オブジェクト ---	ロード・モジュール ---
ユーザの処置	入力ファイルを指定してください。					

エラー・メッセージ (A002)	和文: “入力ファイル総数が制限を越えている” ----- 英文: “Too many input files”					
原因	入力ファイル総数が制限を越えて指定されています。					
プログラムの処理	プログラムの実行を中止します。					
ファイルの出力	-JUNK指定 あり	リンク・オブジェクト ---	ロード・モジュール ---	-JUNK指定 なし	リンク・オブジェクト ---	ロード・モジュール ---
ユーザの処置	分割しているソース・プログラムを、リンカの最大入力ファイル総数以内になるようにまとめてください。					

エラー・メッセージ (A005)	和文: “ファイルに不当なものが指定されている'ファイル名'” ----- 英文: “Illegal file specification 'filename'”					
原因	ファイル名に不当なものが指定されています。入力ファイルとしてデバイス名やバス名だけの指定はできません。アスキー・ファイルの出力先としてのCLOCKを指定することはできません。バイナリ・ファイルの出力先としてCON, PRN, CLOCKを指定することはできません。					
プログラムの処理	プログラムの実行を中止します。					
ファイルの出力	-JUNK指定 あり	リンク・オブジェクト ---	ロード・モジュール ---	-JUNK指定 なし	リンク・オブジェクト ---	ロード・モジュール ---
ユーザの処置	指定したファイル名を確認してください。					

エラー・メッセージ (A006)	和文: "ファイルが見つかりません'ファイル名'" ----- 英文: "File not found 'filename'"	
原因	指定された入力ファイルが存在しません。	
プログラムの処理	プログラムの実行を中止します。	
ファイルの出力	-JUNK指定, リンク・オブジェクト, ロード・モジュール あり, ---, ---	-JUNK指定, リンク・オブジェクト, ロード・モジュール なし, ---, ---
ユーザの処置	指定した入力ファイルのパス名, ファイル名を確認してください。	

エラー・メッセージ (A007)	和文: "入力ファイルが重複指定されている'ファイル名'" ----- 英文: "Input file specification overlapped 'filename'"	
原因	入力ファイル名とパラメータ・ファイル名が重複して指定されています。	
プログラムの処理	プログラムの実行を中止します。	
ファイルの出力	-JUNK指定, リンク・オブジェクト, ロード・モジュール あり, ---, ---	-JUNK指定, リンク・オブジェクト, ロード・モジュール なし, ---, ---
ユーザの処置	指定したファイル名を確認してください。	

エラー・メッセージ (A008)	和文: "入出力ファイルが重複指定されている'ファイル名'" ----- 英文: "File specification conflicted 'filename'"	
原因	出力ファイル名同士, 入力ファイル名と出力ファイル名, またはパラメータ・ファイル名と出力ファイル名が重複して指定されています。	
プログラムの処理	プログラムの実行を中止します。	
ファイルの出力	-JUNK指定, リンク・オブジェクト, ロード・モジュール あり, ---, ---	-JUNK指定, リンク・オブジェクト, ロード・モジュール なし, ---, ---
ユーザの処置	指定する入力ファイル名, 出力ファイル名, パラメータ・ファイル名が重複しないように直してください。	

エラー・メッセージ (A009)	和文: "ファイルを作成できないファイル名" ----- 英文: "Unable to make file 'filename'"				
原因	指定された出力ファイルがリード・オンリー・ファイルとしてすでにあるため、作成できません。				
プログラムの処理	プログラムの実行を中止します。				
ファイルの出力	-JUNK指定、 あり	リンク・オブジェクト ---	ロード・モジュール ---	-JUNK指定 なし	リンク・オブジェクト --- ロード・モジュール ---
ユーザの処置	指定した出力ファイル名を、別のファイル名に変更してください。				

エラー・メッセージ (A010)	和文: "ドライブまたはディレクトリが見つかりません'ファイル名" ----- 英文: "Directory not found 'filename'"				
原因	出力ファイル名中に存在しないドライブまたはディレクトリが含まれています。				
プログラムの処理	プログラムの実行を中止します。				
ファイルの出力	-JUNK指定、 あり	リンク・オブジェクト ---	ロード・モジュール ---	-JUNK指定 なし	リンク・オブジェクト --- ロード・モジュール ---
ユーザの処置	指定したドライブ、ディレクトリ名を確認してください。				

エラー・メッセージ (A011)	和文: "不正なパスが指定されているパス名" ----- 英文: "Illegal path 'passname'"				
原因	オプション・パラメータにパス名を指定するオプションで、パス名以外が指定されました。				
プログラムの処理	プログラムの実行を中止します。				
ファイルの出力	-JUNK指定、 あり	リンク・オブジェクト ---	ロード・モジュール ---	-JUNK指定 なし	リンク・オブジェクト --- ロード・モジュール ---
ユーザの処置	オプション・パラメータのパス名を確認してください。パス名には存在するパス名を指定してください。				

エラー・メッセージ (A012)	和文: "オプション・パラメータがないオプション" ----- 英文: "Missing parameter 'option'"					
原因	必要なオプション・パラメータが指定されていません。					
プログラムの処理	プログラムの実行を中止します。					
ファイルの出力	-JUNK指定	リンク・オブジェクト	ロード・モジュール	-JUNK指定	リンク・オブジェクト	ロード・モジュール
	あり	---	---	なし	---	---
ユーザの処置	必要なオプション・パラメータを指定してください。					

エラー・メッセージ (A013)	和文: "オプション・パラメータは不要オプション" ----- 英文: "Parameter not needed 'option'"					
原因	不要なオプション・パラメータが指定されています。					
プログラムの処理	プログラムの実行を中止します。					
ファイルの出力	-JUNK指定	リンク・オブジェクト	ロード・モジュール	-JUNK指定	リンク・オブジェクト	ロード・モジュール
	あり	---	---	なし	---	---
ユーザの処置	不要なオプション・パラメータを削除してください。					

エラー・メッセージ (A014)	和文: "パラメータに範囲外の数値が指定されたオプション" ----- 英文: "Out of range 'option'"					
原因	オプション・パラメータに範囲外の数値が指定されています。					
プログラムの処理	プログラムの実行を中止します。					
ファイルの出力	-JUNK指定	リンク・オブジェクト	ロード・モジュール	-JUNK指定	リンク・オブジェクト	ロード・モジュール
	あり	---	---	なし	---	---
ユーザの処置	オプション・パラメータを確認して、範囲内の数値を指定してください。					

エラー・メッセージ (A015)	和文: “パラメータの文字数が制限値を越えているオプション” ----- 英文: “Parameter is too long 'option'”					
原因	オプション・パラメータの文字数が制限を越えて指定されています。					
プログラムの処理	プログラムの実行を中止します。					
ファイルの出力	-JUNK指定 あり	リンク・オブジェクト ---	ロード・モジュール ---	-JUNK指定 なし	リンク・オブジェクト ---	ロード・モジュール ---
ユーザの処置	オプション・パラメータの文字数を制限値以内にしてください。					

エラー・メッセージ (A018)	和文: “不正なオプションが指定されたオプション” ----- 英文: “Option is not recognized 'option'”					
原因	誤ったオプションが指定されています。					
プログラムの処理	プログラムの実行を中止します。					
ファイルの出力	-JUNK指定 あり	リンク・オブジェクト ---	ロード・モジュール ---	-JUNK指定 なし	リンク・オブジェクト ---	ロード・モジュール ---
ユーザの処置	正しいオプションを指定してください。					

エラー・メッセージ (A019)	和文: “パラメータ・ファイル中に-PARオプションが指定されている” ----- 英文: “Parameter file nested”					
原因	パラメータ・ファイル中に-PARオプションが指定されています。 パラメータ・ファイルのネストはできません。					
プログラムの処理	プログラムの実行を中止します。					
ファイルの出力	-JUNK指定 あり	リンク・オブジェクト ---	ロード・モジュール ---	-JUNK指定 なし	リンク・オブジェクト ---	ロード・モジュール ---
ユーザの処置	パラメータ・ファイル中の-PARオプションを削除してください。					

エラー・メッセージ (A020)	和文: "パラメータ・ファイルを読み込めないファイル名" ----- 英文: "Parameter file read error 'filename'"					
原因	パラメータ・ファイルを読み込みません。					
プログラムの処理	プログラムの実行を中止します。					
ファイルの出力	-JUNK指定 あり	リンク・オブジェクト ---	ロード・モジュール ---	-JUNK指定 なし	リンク・オブジェクト ---	ロード・モジュール ---
ユーザの処置	パラメータ・ファイル名の指定が正しいことを確認してください。 パラメータ・ファイルが存在するディスクの状態を確認してください。					

エラー・メッセージ (A023)	和文: "-NOOBJまたは-NOICオプションと-INCオプションが同時に指定されている" ----- 英文: "-NOOBJ or -NOIC and -INC specified at the same time"					
原因	-NOOBJまたは-NOICオプションと-INCオプションが同時に指定されています。					
プログラムの処理	プログラムの実行を中止します。					
ファイルの出力	-JUNK指定 あり	リンク・オブジェクト ---	ロード・モジュール ---	-JUNK指定 なし	リンク・オブジェクト ---	ロード・モジュール ---
ユーザの処置	-NOOBJまたは-NOICオプションと-INCオプションは同時に指定できません。 -INCオプション指定時は、-NOOBJ、-NOICオプションを指定しないでください。					

エラー・メッセージ (F102)	和文: "ディレクティブの構文誤り" ----- 英文: "Directive syntax error"					
原因	ディレクティブの記述に誤りがあります。					
プログラムの処理	そのディレクティブを無視して処理を続行します。					
ファイルの出力	-JUNK指定 あり	リンク・オブジェクト LNK	ロード・モジュール ICE/PRO	-JUNK指定 なし	リンク・オブジェクト ---	ロード・モジュール ---
ユーザの処置	ディレクティブの構文を確認してください。					

エラー・メッセージ (A104)	和文: "ファイル名対象品種が異なっている" ----- 英文: "'filename' Different processor type"
原因	最初に入力したオブジェクト・モジュール・ファイルとアセンブルまたはコンパイル対象の異なるオブジェクト・モジュール・ファイルを入力しました。
プログラムの処理	プログラムの実行を中止します。
ファイルの出力	-JUNK指定、リンク・オブジェクト、ロード・モジュール あり、---、--- -JUNK指定、リンク・オブジェクト、ロード・モジュール なし、---、---
ユーザの処置	異なるアセンブルやコンパイル対象品種を指定したオブジェクト・モジュール・ファイルはリンクできませんので、リンクするすべてのオブジェクト・モジュール・ファイルの対象品種が同じになるように、アセンブルまたはコンパイルし直してください。

エラー・メッセージ (A106)	和文: "テンポラリ・ファイルが作成できないファイル名" ----- 英文: "Can't create temporary file 'filename'"
原因	テンポラリ・ファイルを作成できません。
プログラムの処理	プログラムの実行を中止します。
ファイルの出力	-JUNK指定、リンク・オブジェクト、ロード・モジュール あり、---、--- -JUNK指定、リンク・オブジェクト、ロード・モジュール なし、---、---
ユーザの処置	-WORKオプションが、環境変数TMPで指定したディレクトリ名を確認してください。 テンポラリ・ファイルを作成しようとしたディスクの状態(空き容量、メディアの状態など)を確認してください。

エラー・メッセージ (A107)	和文: "ファイルファイル名が-HOSTオプション指定でアセンブルされていない" ----- 英文: "File 'filename' isn't assembled with -HOST option"
原因	-HOSTオプション(リンク)指定時、入力ファイルの中に-HOSTオプション(アセンブラ)指定をしてアセンブルされていないものがあります。
プログラムの処理	プログラムの実行を中止します。
ファイルの出力	-JUNK指定、リンク・オブジェクト、ロード・モジュール あり、---、--- -JUNK指定、リンク・オブジェクト、ロード・モジュール なし、---、---
ユーザの処置	エラーになった入力ファイルを-HOSTオプション(アセンブラ)指定をしてアセンブルし直してください。

エラー・メッセージ (A109)	和文: "リンカ内部エラー" ----- 英文: "Linker internal error"					
原因	内部エラーです。					
プログラムの処理	プログラムの実行を中止します。					
ファイルの出力	-JUNK指定 あり	リンク・オブジェクト ---	ロード・モジュール ---	-JUNK指定 なし	リンク・オブジェクト ---	ロード・モジュール ---
ユーザの処置	特約店または日本電気までご連絡ください。					

エラー・メッセージ (F110)	和文: "不正な数値" ----- 英文: "illegal number"					
原因	ディレクティブ中の数値の記述が正しくありません。					
プログラムの処理	そのディレクティブを無視して処理を続行します。					
ファイルの出力	-JUNK指定 あり	リンク・オブジェクト LNK	ロード・モジュール ICE/PRO	-JUNK指定 なし	リンク・オブジェクト ---	ロード・モジュール ---
ユーザの処置	ディレクティブの数値記述を確認してください。					

エラー・メッセージ (A112)	和文: "セクション'セクション名'製品にないベクタ・アドレスが指定された" ----- 英文: "Section 'section' bad vector address"					
原因	再配置属性VECTn/DVECTnで製品にないベクタ・アドレスを指定しています。					
プログラムの処理	プログラムの実行を中止します。					
ファイルの出力	-JUNK指定 あり	リンク・オブジェクト ---	ロード・モジュール ---	-JUNK指定 なし	リンク・オブジェクト ---	ロード・モジュール ---
ユーザの処置	再配置属性には製品で実装しているベクタ・アドレスを指定してください。					



エラー・メッセージ (A113)	和文: “ディレクティブで指定したアドレスがセグメント指定と一致していない (section 'セクション名’) ” 英文: “Not same address (section 'section’) ”	
原因	ディレクティブで指定したアドレスとセグメント指定の対応が正しくありません。	
プログラムの処理	プログラムの実行を中止します。	
ファイルの出力	-JUNK指定、リンク・オブジェクト、ロード・モジュール あり、---、---	-JUNK指定、リンク・オブジェクト、ロード・モジュール なし、---
ユーザの処置	ディレクティブで指定したアドレスとセグメント指定の対応を確認してください。	

エラー・メッセージ (W114)	和文: “ファイル'ファイル名'のデバイス・ファイルが最初のファイルと異なる” 英文: “'filename' Different device file from first input file”	
原因	入力ファイルをアセンブルするときに使用したデバイス・ファイルが、異なるディレクトリで、同名のファイルです。	
プログラムの処理	ワーニングが発生したあと、処理を続行します。 最初の入力ファイルで使用したデバイス・ファイルを使い、処理します。	
ファイルの出力	-JUNK指定、リンク・オブジェクト、ロード・モジュール あり、LNK、ICE/PRO	-JUNK指定、リンク・オブジェクト、ロード・モジュール なし、LNK、ICE/PRO
ユーザの処置	複数ファイルを入力する場合、アセンブル時には同一のデバイス・ファイルを使用してください。	

エラー・メッセージ (W115)	和文: “生成したオブジェクト・コードでROM受注はできません” 英文: “Can't order ROM code with generated object code”	
原因	ロード・モジュール・ファイルを生成するために使用したツール (アセンブラ、コンパイラ、リンカ・デバイス・ファイル) の中に、バージョンが正式リリース版でないものがあります。	
プログラムの処理	ワーニングが発生したあと、処理を続行します。 ACROSSチェック用の項目にバージョン情報を格納します。	
ファイルの出力	-JUNK指定、リンク・オブジェクト、ロード・モジュール あり、LNK、ICE/PRO	-JUNK指定、リンク・オブジェクト、ロード・モジュール なし、LNK、ICE/PRO
ユーザの処置	ROM受注する場合には、ツールはすべて正式リリース版を使用してください。	

エラー・メッセージ (W116)	和文: “ファイル‘filename’が見つからないのでフルリンクを行う” ----- 英文: “File ‘filename’ file not found”					
原因	-INCREMENTAL指定時、旧.LNKファイルまたは旧.ICEファイルが見つかりません。					
プログラムの処理	ワーニングを発生したあと、処理を続行します。 差分ファイルは出力せず、フルリンクを行います。					
ファイルの出力	-JUNK指定 あり	リンク・オブジェクト LNK	ロード・モジュール ICE/PRO	-JUNK指定 なし	リンク・オブジェクト LNK	ロード・モジュール ICE/PRO
ユーザの処置	旧.LNKファイルまたは旧.ICEファイルがない場合、-INCREMENTALを指定しないでください。					

エラー・メッセージ (F201)	和文: “セクション‘section’は複数のマージ・ディレクティブで指定されている” ----- 英文: “Multiple section definition ‘section’ in merge directive”					
原因	マージ・ディレクティブで指定されたセクションはすでに登録されています（同じセクションを複数のマージ・ディレクティブで割り付け指定しようとしています）。					
プログラムの処理	そのマージ・ディレクティブを無視して処理を続行します。					
ファイルの出力	-JUNK指定 あり	リンク・オブジェクト LNK	ロード・モジュール ICE/PRO	-JUNK指定 なし	---	---
ユーザの処置	1つのセクションに対するリンク指示は、1つのマージ・ディレクティブで行ってください。					

エラー・メッセージ (A203)	和文: “セクション‘section’名のセクション情報が不正” ----- 英文: “Section ‘section’ unknown section type”					
原因	入力したオブジェクト・モジュール・ファイルのセクション情報に誤りがあります。					
プログラムの処理	プログラムの実行を中止します。					
ファイルの出力	-JUNK指定 あり	---	---	-JUNK指定 なし	---	---
ユーザの処置	そのオブジェクト・モジュール・ファイルが正しいものであることを確認してください。もう一度アセンブルまたはコンパイルし直してください。					

エラー・メッセージ (A204)	和文: "同名のセクションが存在する" ----- 英文: "Exist same name sections"					
原因	マージ・ディレクティブの結合型指定でCOMPLETEが指定されている場合、入力ファイル中に同名のセクションが複数存在しています。					
プログラムの処理	プログラムの実行を中止します。					
ファイルの出力	-JUNK指定 あり	リンクオブジェクト ---	ロード・モジュール ---	-JUNK指定 なし	リンクオブジェクト ---	ロード・モジュール ---
ユーザの処置	同名のセクションが複数存在するセクションに対してマージ・ディレクティブの結合型指定でCOMPLETEを指定しないでください。					

エラー・メッセージ (F206)	和文: "セクション'セクション名'をメモリ領域に配置できない" ----- 英文: "Section 'section' can't allocate to memory - ignored"					
原因	セクションをメモリ領域に割り付けられません (セクションを割り付けるのに十分なメモリ領域がありません)。					
プログラムの処理	そのセクションを無視して処理を続行します。 配置できなかったセクション中のシンボル定義は有効とします。					
ファイルの出力	-JUNK指定 あり	リンクオブジェクト LNK	ロード・モジュール ICE/PRO	-JUNK指定 なし	リンクオブジェクト ---	ロード・モジュール ---
ユーザの処置	リンクしようとするすべてのセクションの大きさが、メモリ領域に適合していることをチェックしてください。このセクションがほかのセクションの再配置属性の配置条件 (アブソリュート・セクションなど) により、セクションの大きさが小さくても配置できないことがあります。					

エラー・メッセージ (F207)	和文: "セクション'セクション名'のセクション型情報が不正" ----- 英文: "Section 'section' has illegal section type"					
原因	このセクションの型情報が不正です。					
プログラムの処理	そのセクションを無視して処理を続行します。					
ファイルの出力	-JUNK指定 あり	リンクオブジェクト LNK	ロード・モジュール ICE/PRO	-JUNK指定 なし	リンクオブジェクト ---	ロード・モジュール ---
ユーザの処置	このセクションが含まれるソース・プログラムをもう一度アSEMBルまたはコンパイルし直してください。マージ・ディレクティブの記述を確認してください。					

エラー・メッセージ (F208)	和文: "セクション'セクション名'が存在しない" ----- 英文: "Section 'section' is not exist- ignored"					
原因	ディレクティブで指定されたセクションが存在しません。					
プログラムの処理	そのセクションを無視して処理を続行します。					
ファイルの出力	-JUNK指定 あり	リンク・オブジェクト LNK	ロード・モジュール ICE/PRO	-JUNK指定 なし	リンク・オブジェクト ---	ロード・モジュール ---
ユーザの処置	存在しないセクションに対し、ディレクティブで指定しないでください。					

エラー・メッセージ (W209)	和文: "セクション'セクション名'がアセンブル時と配置指定が異なる" ----- 英文: "The type specified for section 'section' is different from that specified at assemble time"					
原因	ディレクティブで指定された配置指定がアセンブル時の指定と異なります。					
プログラムの処理	ワーニングを発生したあと、処理を続行します。 ディレクティブで指定された配置指定が有効となり、処理を行います。					
ファイルの出力	-JUNK指定 あり	リンク・オブジェクト LNK	ロード・モジュール ICE/PRO	-JUNK指定 なし	リンク・オブジェクト LNK	ロード・モジュール ICE/PRO
ユーザの処置	アセンブル時とディレクティブで異なった再配置属性を指定しないでください。					

エラー・メッセージ (F210)	和文: "セクション'セクション名'をテーブル領域に配置できない" ----- 英文: "Section 'section' can't allocate to memory (table area) - ignored"					
原因	セクションを指定されたテーブル領域に配置できません。					
プログラムの処理	そのセクションを無視して処理を続行します。					
ファイルの出力	-JUNK指定 あり	リンク・オブジェクト LNK	ロード・モジュール ICE/PRO	-JUNK指定 なし	リンク・オブジェクト ---	ロード・モジュール ---
ユーザの処置	リンクしようとするすべてのセクションの大きさが、メモリ領域に適合していることをチェックしてください。このセクションがほかのセクションの再配置属性の配置条件（アブソリュート・セクションなど）により、セクションの大きさが小さくても配置できないことがあります。					

エラー・メッセージ (A211)	和文: “インクリメンタル・リンク前に存在していたセクション‘セクション名’がない” 英文: “Section ‘section’ is not exist at incremental-link time”					
原因	インクリメンタル・リンク時、それ以前に存在していたセクションがありません。					
プログラムの処理	プログラムの実行を中止します。					
ファイルの出力	-JUNK指定 あり	リンク・オブジェクト ---	ロード・モジュール ---	-JUNK指定 なし	リンク・オブジェクト ---	ロード・モジュール ---
ユーザの処置	インクリメンタル・リンク時は、セクションを削除しないでください。					

エラー・メッセージ (F212)	和文: “ORGアドレス・エラー (file‘ファイル名’, org addr xxxxH)” 英文: “Origin address error (file ‘filename’, org addr xxxxH)”					
原因	アブソリュート・モードでアセンブルしたモジュール中で指定されているORG疑似命令のアドレスがリンカで割り付けるアドレスよりも低いアドレスです。					
プログラムの処理	そのORG疑似命令を無視して処理を続行します。					
ファイルの出力	-JUNK指定 あり	リンク・オブジェクト LNK	ロード・モジュール ICE/PRO	-JUNK指定 なし	リンク・オブジェクト ---	ロード・モジュール ---
ユーザの処置	アブソリュート・モードのORG疑似命令で指定するアドレスは、昇順にしてください。					

エラー・メッセージ (W213)	和文: “アセンブル・モードの異なるオブジェクト・モジュール・ファイルをリンクした” 英文: “Different assemble mode”					
原因	アセンブル・モード (アブソリュート/リロケータブル) の異なるオブジェクト・モジュール・ファイルをリンカに入力しました。					
プログラムの処理	ワーニングが発生したあと、処理を続行します。アブソリュート・モードのオブジェクト・モジュール・ファイルは、リロケータブル・モードとして処理します。					
ファイルの出力	-JUNK指定 あり	リンク・オブジェクト LNK	ロード・モジュール ICE/PRO	-JUNK指定 なし	リンク・オブジェクト LNK	ロード・モジュール ICE/PRO
ユーザの処置	アセンブル・モードの異なるオブジェクト・モジュール・ファイルを一緒にリンクしないでください。					

エラー・メッセージ (A214)	和文: "タイプの異なるセクションが同名で定義されている (section 'セクション名')" ----- 英文: "Different section type (section 'section')"
原因	配置型の異なるセクションが同名で定義されています。
プログラムの処理	プログラムの実行を中止します。
ファイルの出力	-JUNK指定、リンク・オブジェクト、ロード・モジュール あり、---、---、---、なし、---、---
ユーザの処置	配置型の異なるセクションは、同名で定義しないでください。

エラー・メッセージ (F215)	和文: "CROMのセグメント指定に誤りがある (section 'セクション名')" ----- 英文: "Illegal CROM segment (section 'section')"
原因	再配置属性指定 (CROM) とセグメント指定の両方をした場合、CROM領域とセグメント番号の対応が取れていません。
プログラムの処理	セグメント指定を無視して処理を続行します。
ファイルの出力	-JUNK指定、リンク・オブジェクト、ロード・モジュール あり、LNK、ICE/PRO、なし、---、---
ユーザの処置	再配置属性 (CROM) とセグメント番号を指定する場合は、CROM領域に対応したセグメント番号を指定してください。

エラー・メッセージ (F301)	和文: "リロケータブル・オブジェクト・コードが領域外のアドレスに対して出力されている (file 'ファイル名', section 'セクション名', address xxxxH)" ----- 英文: "Relocatable object code address out of range (file 'filename', section 'section', address xxxxH)"
原因	入力したオブジェクト・モジュール・ファイル中に含まれるリロケータブル・オブジェクト・コードの修正情報が、オブジェクト・コードの存在しないアドレスに対して出力されています。
プログラムの処理	そのリロケーションを無視して処理を続行します。
ファイルの出力	-JUNK指定、リンク・オブジェクト、ロード・モジュール あり、LNK、ICE/PRO、なし、---、---
ユーザの処置	シンボルの参照が正しいことを確認してください。 もう一度アセンブルまたはコンパイルし直してください。 アドレス xxxxHとして表示されるアドレスは、セクション配置後の絶対アドレスです。

エラー・メッセージ (F302)	和文: “リロケータブル・オブジェクト・コードのシンボル・インデックスを参照できない (file 'ファイル名', section 'セクション名', address xxxxxH)” 英文: “Can't find symbol index in relocatable object code (file 'filename', section 'section', address xxxxxH)”
原因	入力したオブジェクト・モジュール・ファイル中に含まれるリロケータブル・オブジェクト・コードの修正情報に誤りがあり、シンボル情報を正しく参照していません。
プログラムの処理	そのリロケーション・エントリを無視して処理を続行します。
ファイルの出力	-JUNK指定、リンク・オブジェクト、ロード・モジュール あり LNK ICE/PRO      なし --- ---
ユーザの処置	シンボル、変数などの参照方法が正しいことを確認してください。 もう一度アセンブルまたはコンパイルし直してください。 アドレス xxxxxHとして表示されるアドレスは、セクション配置後の絶対アドレスです。

エラー・メッセージ (F304)	和文: “オペランドの値が不正 (section 'セクション名', address xxxxxH)” 英文: “Operand out of range (section 'section', address xxxxxH)”
原因	命令のオペランドに不正な値が記述されています。
プログラムの処理	エラーが発生したあと、処理を続行します。
ファイルの出力	-JUNK指定、リンク・オブジェクト、ロード・モジュール あり LNK ICE/PRO      なし --- ---
ユーザの処置	命令のオペランドには、製品で許される範囲の値を記述してください。

エラー・メッセージ (F305)	和文: “ROM領域オーバフロー、EPAビット・オン (file 'ファイル名', section 'セクション名', address xxxxxH)” 英文: “ROM address overflow, EPA bit on (file 'filename', section 'section', address xxxxxH)”
原因	プログラムの一部がEPA領域に配置またはEPA領域を参照しています。
プログラムの処理	エラーが発生したあと、そのオブジェクト・コードに対しEPAビットを“1”にセットして、処理を続行します (オブジェクト・コードは正常に出力します)。
ファイルの出力	-JUNK指定、リンク・オブジェクト、ロード・モジュール あり LNK ICE/PRO      なし LNK ICE/PRO
ユーザの処置	プログラムのオブジェクト・コードのサイズを縮めてください。

エラー・メッセージ (F306)	和文: "プログラム・メモリを越えた領域に文が記述されている [REMAIN領域内] (section 'セクション名', address xxxxH, size yyyyH) " ----- 英文: "Out of address range [REMAIN AREA] (section 'section', address xxxxH size yyyyH) "					
原因	プログラムがPROGRAM空間を越え、REMAIN領域に入っています。					
プログラムの処理	エラーを発生したあと、処理を続行します (オブジェクト・コードは正常に出力します)。					
ファイルの出力	-JUNK指定 あり	リンク・オブジェクト LNK	ロード・モジュール ICE/PRO	-JUNK指定 なし	リンク・オブジェクト LNK	ロード・モジュール ICE/PRO
ユーザの処置	プログラムのオブジェクト・コードのサイズを縮めてください。					

エラー・メッセージ (F309)	和文: "プログラム・メモリを越えた領域を参照している [REMAIN領域参照] (file 'ファイル名', section 'セクション名', address xxxxH) " ----- 英文: "Referring the address over area [REMAIN AREA] (file 'filename' section 'section', address xxxxH) "					
原因	プログラム中でREMAIN領域内のアドレスを参照しています。					
プログラムの処理	エラーを発生したあと、処理を続行します (オブジェクト・コードは正常に出力します)。					
ファイルの出力	-JUNK指定 あり	リンク・オブジェクト LNK	ロード・モジュール ICE/PRO	-JUNK指定 なし	リンク・オブジェクト LNK	ロード・モジュール ICE/PRO
ユーザの処置	プログラムのオブジェクト・コードのサイズを縮めてください。					

エラー・メッセージ (F310)	和文: "プログラム・メモリを越えた領域を参照している [EPA領域参照] , EPAビット・オン (file 'ファイル名', section 'セクション名', address xxxxH) " ----- 英文: "Referring the address over area [EPA AREA], EPA bit on (file 'filename', section 'section', address xxxxH) "					
原因	プログラム中でEPA領域内のアドレスを参照しています。					
プログラムの処理	エラーを発生したあと、そのオブジェクト・コードにEPAビットを "1" にセットして、処理を続行します (オブジェクト・コードは正常に出力します)。					
ファイルの出力	-JUNK指定 あり	リンク・オブジェクト LNK	ロード・モジュール ICE/PRO	-JUNK指定 なし	リンク・オブジェクト LNK	ロード・モジュール ICE/PRO
ユーザの処置	プログラムのオブジェクト・コードのサイズを縮めてください。					



エラー・メッセージ (W312)	和文: "未参照パブリック・シンボル" ----- 英文: "Unreference public symbol"					
原因	パブリック宣言されたシンボルがプログラム中で参照されていません。					
プログラムの処理	ワーニングを発生したあと、処理を続行します。					
ファイルの出力	-JUNK指定 あり	リンク・オブジェクト LNK	ロード・モジュール ICE/PRO	-JUNK指定 なし	リンク・オブジェクト LNK	ロード・モジュール ICE/PRO
ユーザの処置	外部参照しないシンボルはパブリック宣言しないでください。					

エラー・メッセージ (F313)	和文: "OPTION疑似命令が二重定義されている (file'ファイル名') " ----- 英文: "Duplicated OPTION directive (file 'filename') "					
原因	複数の入力ファイル中でマスク・オプション定義ブロックが2回以上定義されています。					
プログラムの処理	最初の定義を有効として処理を続行します。					
ファイルの出力	-JUNK指定 あり	リンク・オブジェクト LNK	ロード・モジュール ICE/PRO	-JUNK指定 なし	リンク・オブジェクト ---	ロード・モジュール ---
ユーザの処置	必要のないマスク・オプション定義ブロックは削除してください。					

エラー・メッセージ (F314)	和文: "マスク・オプション定義ブロックがない" ----- 英文: "Not found Mask-option block"					
原因	マスク・オプションのある品種で入力ファイル中にマスク・オプション定義ブロックが記述されていません。					
プログラムの処理	エラーを発生したあと、処理を続行します。					
ファイルの出力	-JUNK指定 あり	リンク・オブジェクト LNK	ロード・モジュール ICE/PRO	-JUNK指定 なし	リンク・オブジェクト ---	ロード・モジュール ---
ユーザの処置	マスク・オプションのある品種ではマスク・オプション定義ブロックを記述してください。					

エラー・メッセージ (W315)	和文：“EPA領域に文が記述されているため間接アドレス命令が正常に動作しない場合がある” 英文：“Indirect addressing instructions may not work properly due to the program exceeded to EPA area”					
原因	プログラムがEPA領域に達したため、間接分岐命令などの飛び先がPROGRAM空間なのかEPA領域なのか判断できません。					
プログラムの処理	ワーニングを発生したあと、処理を続行します。					
ファイルの出力	-JUNK指定 あり	リンク・オブジェクト LNK	ロード・モジュール ICE/PRO	-JUNK指定 なし	リンク・オブジェクト LNK	ロード・モジュール ICE/PRO
ユーザの処置	プログラムのオブジェクト・コードのサイズを縮めてください。					

エラー・メッセージ (F317)	和文：“セクションをすべてEPA領域に配置した (file 'ファイル名', section 'セクション名', address xxxxH, size yyyyH) ” 英文：“All of the section was allocated to EPA area (file 'filename', section 'section', address xxxxH, size yyyyH)”					
原因	セクション全体をPROGRAM領域に配置できません。					
プログラムの処理	エラーを発生したあと、処理を続行します (オブジェクト・コードは正常に出力します)。					
ファイルの出力	-JUNK指定 あり	リンク・オブジェクト LNK	ロード・モジュール ICE/PRO	-JUNK指定 なし	リンク・オブジェクト LNK	ロード・モジュール ICE/PRO
ユーザの処置	プログラムのオブジェクト・コードのサイズを縮めてください。					

エラー・メッセージ (F318)	和文：“セクションの一部をEPA領域に配置した (file 'ファイル名', section 'セクション名', address xxxxH, size yyyyH) ” 英文：“Part of the section was allocated to EPA area (file 'filename', section 'section', address xxxxH, size yyyyH)”					
原因	セクションの一部をPROGRAM領域に配置できません。					
プログラムの処理	エラーを発生したあと、処理を続行します (オブジェクト・コードは正常に出力します)。					
ファイルの出力	-JUNK指定 あり	リンク・オブジェクト LNK	ロード・モジュール ICE/PRO	-JUNK指定 なし	リンク・オブジェクト LNK	ロード・モジュール ICE/PRO
ユーザの処置	プログラムのオブジェクト・コードのサイズを縮めてください。					

エラー・メッセージ (F319)	和文: "アドレス境界誤り (file 'ファイル名', section 'セクション名', address xxxxH) " ----- 英文: "Boundary error (file 'filename', section 'section', address xxxxH) "					
原因	アドレス境界が誤っています。DCP疑似命令のオブジェクト・コードが下位4ビットが0FHのアドレスに配置されています。					
プログラムの処理	エラーが発生したあと、処理を続行します。					
ファイルの出力	-JUNK指定 あり	リンク・オブジェクト LNK	ロード・モジュール ICE/PRO	-JUNK指定 なし	リンク・オブジェクト ---	ロード・モジュール ---
ユーザの処置	DCP疑似命令が生成するオブジェクト・コードを、下位4ビットが0FH以外のアドレスに配置してください。					

エラー・メッセージ (W320)	和文: "指定値が規定値をオーバーしたためその分は切り捨てた (section 'セクション名', address xxxxH) " ----- 英文: "Omitted a surplus due to an input value is over a regular value (section 'section', address xxxxH) "					
原因	DW/DBのオペランドに指定した値が規定値を越えています。					
プログラムの処理	ワーニングが発生したあと、処理を続行します。オーバーした値は切り捨てます。					
ファイルの出力	-JUNK指定 あり	リンク・オブジェクト LNK	ロード・モジュール ICE/PRO	-JUNK指定 なし	リンク・オブジェクト LNK	ロード・モジュール ICE/PRO
ユーザの処置	DW/DBのオペランドには規定値を越えない値を指定してください。					

エラー・メッセージ (A401)	和文: "ファイル'ファイル名'のシンボル・テーブルが不正" ----- 英文: "File 'filename' Bad symbol table"					
原因	入力したオブジェクト・モジュール・ファイルのシンボル情報が不正です。					
プログラムの処理	プログラムの実行を中止します。					
ファイルの出力	-JUNK指定 あり	リンク・オブジェクト ---	ロード・モジュール ---	-JUNK指定 なし	リンク・オブジェクト ---	ロード・モジュール ---
ユーザの処置	もう一度アセンブルまたはコンパイルし直してください。					

エラー・メッセージ (A402)	和文: "ファイル'ファイル名'のシンボルのストリング・テーブルが不正" ----- 英文: "File 'filename' has no string table for symbol!"					
原因	入力したオブジェクト・モジュール・ファイルのシンボル情報不正です (文字情報がありません)。					
プログラムの処理	プログラムの実行を中止します。					
ファイルの出力	-JUNK指定 あり	リンク・オブジェクト ---	ロード・モジュール ---	-JUNK指定 なし	リンク・オブジェクト ---	ロード・モジュール ---
ユーザの処置	もう一度アセンブルまたはコンパイルし直してください。					

エラー・メッセージ (F403)	和文: "ファイル'ファイル名1'で指定したシンボル'シンボル名'がファイル'ファイル名2'で最初 に定義したものと型が異なっている" ----- 英文: "Symbol 'symbol' unmatched type in file 'filename1', First defined in file 'filename2'"					
原因	同名外部定義/参照シンボルの型が'ファイル名1'と'ファイル名2'で異なっています。					
プログラムの処理	あとから入力したファイル'ファイル名1'に出現したシンボルの型を無視して処理を続行します。					
ファイルの出力	-JUNK指定 あり	リンク・オブジェクト LNK	ロード・モジュール ICE/PRO	-JUNK指定 なし	リンク・オブジェクト ---	ロード・モジュール ---
ユーザの処置	シンボルを外部定義/参照するときは、そのシンボルを外部定義/参照しているすべてのソース・プログラム中で、そのシンボルの型を同じにしてください。					

エラー・メッセージ (F404)	和文: "ファイル'ファイル名1'で定義されているシンボル'シンボル名'はファイル'ファイル名2' ですすでに定義されている" ----- 英文: "Multiple Symbol definition 'symbol' in file 'filename1', First defined in file 'filename2'"					
原因	オブジェクト・モジュール・ファイル'ファイル名1'中で定義されているPUBLICシンボルは、オブジェクト・モジュール・ファイル'ファイル名2'ですすでにPUBLIC宣言されています (PUBLICシンボルの二重定義)。					
プログラムの処理	あとから入力したファイル'ファイル名1'に出現したシンボルの定義を無視して処理を続行します。					
ファイルの出力	-JUNK指定 あり	リンク・オブジェクト LNK	ロード・モジュール ICE/PRO	-JUNK指定 なし	リンク・オブジェクト ---	ロード・モジュール ---
ユーザの処置	同一名のシンボルを複数のソース・プログラムで外部定義しないように、どちらかのシンボルの外部定義を取り消すか、シンボルの名前を変えてください (シンボルの型が同一であっても異なっても同一のシンボル名を2回以上外部定義することはできません)。					

エラー・メッセージ (F405)	和文: "未定義シンボル 'シンボル名' — ファイル 'ファイル名'" ----- 英文: "Undefined symbol 'symbol' in file 'filename'"					
原因	ファイルでEXTRN宣言されているシンボルがほかのファイルでPUBLIC宣言されていません。					
プログラムの処理	未定義シンボルの値を0として処理を続行します。					
ファイルの出力	-JUNK指定 あり	リンク・オブジェクト LNK	ロード・モジュール ICE/PRO	-JUNK指定 なし	リンク・オブジェクト ---	ロード・モジュール ---
ユーザの処置	このシンボルをいずれかのソース・プログラムで外部定義してください。または、外部定義されているほかのシンボルを参照するように、ソース・プログラムを修正してください。					

★

エラー・メッセージ (A406)	和文: "パブリック・シンボルが多すぎる" ----- 英文: "Too many public symbol"					
原因	パブリック・シンボルの総数が上限である65535個を越えています。					
プログラムの処理	プログラムの実行を中止します。					
ファイルの出力	-JUNK指定 あり	リンク・オブジェクト ---	ロード・モジュール ---	-JUNK指定 なし	リンク・オブジェクト ---	ロード・モジュール ---
ユーザの処置	ソース・プログラムで使用しているパブリック・シンボルの総数を65535個以下にしてください。					

エラー・メッセージ (A501)	和文: "システムのメモリが足りない" ----- 英文: "Insufficient memory in hostmachine"					
原因	システムにプログラムが動作するための十分なメモリがありません。					
プログラムの処理	プログラムの実行を中止します。					
ファイルの出力	-JUNK指定 あり	リンク・オブジェクト ---	ロード・モジュール ---	-JUNK指定 なし	リンク・オブジェクト ---	ロード・モジュール ---
ユーザの処置	ホスト・マシンにメモリを増設できるときは、メモリを増設してください。 その他、アプリケーション・プログラムが利用可能なメモリを増やしてください。 増設できないとき、このホスト・マシンではリンクできません。					

エラー・メッセージ (A901)	和文: "デバイス・ファイルを開けない'ファイル名'" ----- 英文: "Can't open device file 'filename'"					
原因	デバイス・ファイルを開けません。					
プログラムの処理	プログラムの実行を中止します。					
ファイルの出力	-JUNK指定	リンク・オブジェクト	ロード・モジュール	-JUNK指定	リンク・オブジェクト	ロード・モジュール
	あり	---	---	なし	---	---
ユーザの処置	デバイス・ファイルが正しいディレクトリにあることを確認してください。					

エラー・メッセージ (A904)	和文: "出力ファイル'ファイル名'を開けない" ----- 英文: "Can't open output file 'filename'"					
原因	出力ファイルを開けません (ファイルI/Oエラー)。					
プログラムの処理	プログラムの実行を中止します。					
ファイルの出力	-JUNK指定	リンク・オブジェクト	ロード・モジュール	-JUNK指定	リンク・オブジェクト	ロード・モジュール
	あり	---	---	なし	---	---
ユーザの処置	出力ファイル名の指定が正しいことを確認してください。 出力ファイルを作成しようとしたディスクの状態 (空き容量、メディアの状態など) を確認してください。					

エラー・メッセージ (A905)	和文: "テンポラリ・ファイル'ファイル名'を作成できない" ----- 英文: "Can't create temporary file 'filename'"					
原因	シンボル・エントリ用のテンポラリ・ファイルを作成できません (ファイルI/Oエラー)。					
プログラムの処理	プログラムの実行を中止します。					
ファイルの出力	-JUNK指定	リンク・オブジェクト	ロード・モジュール	-JUNK指定	リンク・オブジェクト	ロード・モジュール
	あり	---	---	なし	---	---
ユーザの処置	-WORKオプションが環境変数TMPで指定したディレクトリ名を確認してください。 テンポラリ・ファイルを作成しようとしたディスクの状態 (空き容量、メディアの状態など) を確認してください。					

エラー・メッセージ (A907)	和文: "出力ファイル'ファイル名'に書き込みができない" ----- 英文: "Can't write output file 'filename'"
原因	出力ファイルに書き込みができません (ファイル I/O エラー)。
プログラムの処理	プログラムの実行を中止します。
ファイルの出力	-JUNK指定   リンク・オブジェクト   ロード・モジュール   -JUNK指定   リンク・オブジェクト   ロード・モジュール あり   ---   ---   なし   ---   ---
ユーザの処置	出力ファイル名の指定が正しいことを確認してください。 出力ファイルを作成しようとしたディスクの状態 (空き容量、メディアの状態など) を確認してください。

エラー・メッセージ (A908)	和文: "テンポラリ・ファイル'ファイル名'に書き込みができない" ----- 英文: "Can't access temporary file 'filename'"
原因	テンポラリ・ファイルに書き込みができません (ファイル I/O エラー)。
プログラムの処理	プログラムの実行を中止します。
ファイルの出力	-JUNK指定   リンク・オブジェクト   ロード・モジュール   -JUNK指定   リンク・オブジェクト   ロード・モジュール あり   ---   ---   なし   ---   ---
ユーザの処置	-WORK オプションか環境変数 TMP で指定したディレクトリ名を確認してください。 テンポラリ・ファイルを作成しようとしたディスクの状態 (空き容量、メディアの状態など) を確認してください。

[× ㊦]



— お問い合わせは、最寄りのNECへ —

**【営業関係お問い合わせ先】**

半導体第一販売事業部 半導体第二販売事業部 半導体第三販売事業部	〒108-01 東京都港区芝五丁目7番1号 (NEC本社ビル)	東京 (03)3454-1111 (大代表)				
中部支社 半導体第一販売部 半導体第二販売部	〒460 名古屋市中区錦一丁目17番1号 (NEC中部ビル)	名古屋 (052)222-2170 名古屋 (052)222-2190				
関西支社 半導体第一販売部 半導体第二販売部 半導体第三販売部	〒540 大阪市中央区城見一丁目4番24号 (NEC関西ビル)	大阪 (06) 945-3178 大阪 (06) 945-3200 大阪 (06) 945-3208				
北海道支社 東北支社 福手支社 山形支店 郡山支店 いわき支店 長岡支店 土浦支店 水戸支店 神奈川支社 群馬支店	札幌 (011)231-0161 仙台 (022)267-8740 盛岡 (019)651-4344 山形 (0236)23-5511 郡山 (0249)23-5511 いわき (0246)21-5511 長岡 (0258)36-2155 土浦 (0298)23-6161 水戸 (029)225-1717 横浜 (045)324-5524 高崎 (0273)26-1255	太田支店 宇都宮支店 小山支店 長野支店 甲府支店 埼玉支店 立川支店 千葉支店 群馬支店 群馬支店 北陸支店 福井支店	太田 (0276)46-4011 宇都宮 (028)621-2281 小山 (0285)24-6011 松本 (0263)35-1662 甲府 (055)24-4141 大宮 (048)641-1411 立川 (0425)26-5981 千葉 (043)238-8116 群馬 (054)255-2211 群馬 (0762)23-1621 北陸 (0776)22-1866	西山支店 三葉支店 京都支店 神戸支店 中国支店 鳥取支店 岡山支店 四国支店 新居浜支店 新居浜支店 松山支店 九州支店	高津 徳戸 神戸 鳥取 島取 高松 高松 高松 高松 高松 高松 高松 高松	山 (0764)31-8461 徳 (0592)25-7341 戸 (075)344-7824 神 (078)333-3854 島 (082)242-5504 取 (0857)27-5311 岡 (086)225-4455 松 (0878)36-1200 高 (0897)32-5001 高 (089)945-4149 松 (092)271-7700

**【本資料に関する技術お問い合わせ先】**

半導体ソリューション技術本部 マイクロコンピュータ技術部	〒210 川崎市幸区塚越三丁目484番地	川崎 (044)548-7950	半導体 インフォメーションセンター FAX(044)548-7900 (FAXにてお問い合わせ下さい)
半導体販売技術本部 東日本販売技術部	〒108-01 東京都港区芝五丁目7番1号 (NEC本社ビル)	東京 (03)3798-9619	
半導体販売技術本部 中部販売技術部	〒460 名古屋市中区錦一丁目17番1号 (NEC中部ビル)	名古屋 (052)222-2125	
半導体販売技術本部 西日本販売技術部	〒540 大阪市中央区城見一丁目4番24号 (NEC関西ビル)	大阪 (06) 945-3383	

アンケート記入のお願い

お手数ですが、このドキュメントに対するご意見をお寄せください。今後のドキュメント作成の参考にさせていただきます。

[ドキュメント名] LK17K ユーザーズ・マニュアル

(U12518JJ2V0UM00 (第2版))

[お名前など] (さしつかえのない範囲で)

御社名(学校名、その他) ( )  
ご住所 ( )  
お電話番号 ( )  
お仕事の内容 ( )  
お名前 ( )

1. ご評価 (各欄に○をご記入ください)

項 目	大変良い	良 い	普 通	悪 い	大変悪い
全体の構成					
説明内容					
用語解説					
調べやすさ					
デザイン、字の大きさなど					
その他 ( )					
( )					

2. わかりやすい所 (第 章, 第 章, 第 章, 第 章, その他) ( )

理由 [ ]

3. わかりにくい所 (第 章, 第 章, 第 章, 第 章, その他) ( )

理由 [ ]

4. ご意見, ご要望

5. このドキュメントをお届けしたのは

NEC販売員, 特約店販売員, NEC半導体ソリューション技術本部員,  
その他 ( )

ご協力ありがとうございました。

下記あてにFAXで送信いただくか, 最寄りの販売員にコピーをお渡しください。

NEC半導体インフォメーションセンター

FAX: (044) 548-7900

