

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、誤りが無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。



ユーザース・マニュアル

RX78K4

リアルタイム・オペレーティング・システム

インストレーション編

---

対象デバイス

78K4シリーズ

資料番号 U10604JJ4V0UM00 (第4版)

発行年月 August 2005 N CP(K)

© NEC Electronics Corporation 1994, 2005

(メモ)

# 目次要約

第1章	概 要	...	9
第2章	RX78K4のインストール	...	11
第3章	コンフィギュレータ	...	18
第4章	アプリケーション・プログラムの構築	...	55
付録A	初期化情報テーブル(ラージモデル)	...	61
付録B	オブジェクト管理テーブル(ラージモデル)	...	63
付録C	初期化情報テーブル(スモールモデル)	...	66
付録D	オブジェクト管理テーブル(スモールモデル)	...	69
付録E	システムコール・エントリ・テーブル (ラージモデル/スモールモデル)	...	72
付録F	CALLTテーブル	...	73
付録G	Cコンパイラの使用についての注意事項	...	74
付録H	エラー・メッセージ一覧表	...	75

Windows, Windows XPおよびWindows NT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/ATは、米国IBM Corp.の商標です。

その他、記載の会社名 / 製品名は、各社の商標あるいは登録商標です。

- 本資料に記載されている内容は2005年8月現在のものです。今後、予告なく変更することがあります。量産設計の際には最新の個別データ・シート等をご参照ください。
- 文書による当社の事前の承諾なしに本資料の転載複製を禁じます。当社は、本資料の誤りに関し、一切その責を負いません。
- 当社は、本資料に記載された当社製品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、一切その責を負いません。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。
- 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、一切その責を負いません。
- 当社は、当社製品の品質、信頼性の向上に努めておりますが、当社製品の不具合が完全に発生しないことを保証するものではありません。当社製品の不具合により生じた生命、身体および財産に対する損害の危険を最小限度にするために、冗長設計、延焼対策設計、誤動作防止設計等安全設計を行ってください。
- 当社は、当社製品の品質水準を「標準水準」、「特別水準」およびお客様に品質保証プログラムを指定していただく「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。

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

特別水準：輸送機器（自動車、電車、船舶等）、交通信号機器、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

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、生命維持のための装置またはシステム等

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。意図されていない用途で当社製品の使用をお客様が希望する場合には、事前に当社販売窓口までお問い合わせください。

(注)

(1) 本事項において使用されている「当社」とは、NECエレクトロニクス株式会社およびNECエレクトロニクス株式会社が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う。

(2) 本事項において使用されている「当社製品」とは、(1)において定義された当社の開発、製造製品をいう。

# はじめに

このたびは、NECエレクトロニクス 78K4シリーズの組み込み用ソフトウェアである、「78K4シリーズ用リアルタイムOS RX78K4」をお買いいただきまして、誠にありがとうございます。

本マニュアルは、78K4シリーズ用リアルタイムOS RX78K4のインストール方法を、正しく理解していただくことを目的として書かれています。

**対象者** 本マニュアルは、デバイスのユーザーズ・マニュアル一読程度の知識があり、ソフトウェア・プログラミングの経験がある方を対象として書かれています。

ただし、本パッケージはリアルタイムOS単体であるため、実際に本リアルタイムOSをご使用になる場合には、“CC78K4 Cコンパイラ”と“RA78K4 アセンブラ・パッケージ”が必要になります。

**構成** 本マニュアルの構成を以下に示します。

## 第1章 概要

提供ファイルの説明とインストール方法を説明します。

## 第2章 RX78K4のインストール

本リアルタイムOSのファイル構成、提供ファイルや本リアルタイムOSのホスト・マシンへのインストール方法を説明します。

## 第3章 コンフィギュレータ

コンフィギュレータの操作方法を説明します。

## 第4章 アプリケーション・プログラムの構築

本リアルタイムOSを組み込んだアプリケーションの構築方法を説明します。

**凡 例** 本マニュアル中で共通に使用される記号などの意味を示します。

- ... : 同一の形式を繰り返す
- [ ] : [ ] 内は省略可能
- 「 」 : 「 」 で囲まれた文字そのもの
- “ ” : “ ” で囲まれた文字そのもの
- ‘ ’ : ‘ ’ で囲まれた文字そのもの
- ( ) : ( ) で囲まれた文字そのもの
- 太文字 : 文字そのもの
- : 重要箇所，使用例での下線は入力文字
- : 1文字以上の空白
- : : プログラム記述の省略形
- / : 区切り記号
- \ : バックスラッシュ

#### 関連資料

##### 開発ツール（ソフトウェア）の資料（ユーザーズ・マニュアル）

資料名		資料番号
RX78K4 リアルタイム・オペレーティング・システム	基礎編	U10603J
	インストレーション編	このマニュアル
	タスク・デバッグ編	U15757J

**注意** 上記関連資料は予告なしに内容を変更することがあります。設計などには、必ず最新の資料をご使用ください。



# 目 次

<b>第1章 概 要</b> ...	9
1.1 アプリケーションの開発環境	9
1.2 提供媒体	9
1.3 ロード・モジュール作成手順	10
<b>第2章 RX78K4のインストール</b> ...	11
2.1 インストール手順	11
2.2 インストール後のフォルダ構成	12
2.2.1 オブジェクト・リリース版	12
2.2.2 ソース・リリース版	15
2.2.3 ドキュメント	17
<b>第3章 コンフィギュレータ</b> ...	18
3.1 起 動	21
3.2 処理の流れ	22
3.3 データの入力方法	23
3.4 コンフィギュレーション情報入力	24
3.5 初期化情報テーブル	26
3.5.1 システム情報入力	27
3.5.2 メモリプール情報入力	28
3.5.3 タスク情報入力	30
3.5.4 周期ハンドラ情報入力	34
3.5.5 イベントフラグ情報入力	36
3.5.6 セマフォ情報入力	38
3.5.7 メールボックス情報入力	40
3.6 システムコール・エントリ・テーブル	42
3.6.1 タスク関連システムコール	42
3.6.2 イベントフラグ関連システムコール	45
3.6.3 セマフォ関連システムコール	47
3.6.4 メッセージ関連システムコール	49
3.6.5 メモリ関連システムコール	51
3.6.6 その他のシステムコール	53
<b>第4章 アプリケーション・プログラムの構築</b> ...	55
4.1 必要ファイル	55
4.2 アプリケーション・ロード・モジュール作成手順	59
<b>付録A 初期化情報テーブル(ラージモデル)</b> ...	61
<b>付録B オブジェクト管理テーブル(ラージモデル)</b> ...	63

付録C	初期化情報テーブル (スモールモデル) ...	66
付録D	オブジェクト管理テーブル (スモールモデル) ...	69
付録E	システムコール・エントリ・テーブル (ラージモデル/スモールモデル) ...	72
付録F	CALLTテーブル ...	73
付録G	Cコンパイラの使用についての注意事項 ...	74
付録H	エラー・メッセージ一覧表 ...	75

# 第1章 概 要

## 1.1 アプリケーションの開発環境

RX78K4を組み込んだアプリケーションを開発する上で必要となる環境を以下に示します。

ホスト・マシン	IBM PC/AT™互換機	Microsoft Windows®98/Me/NT®4.0/2000/XP®注
クロス開発ツール	リロケータブル・アセンブラ・パッケージ RA78K4 (NECエレクトロニクス製)	V1.60以上
	コンパイラ・パッケージ CC78K4 (NECエレクトロニクス製)	V2.40以上

注 各オペレーティング・システムの最新のパッチが当たっていること

## 1.2 提供媒体

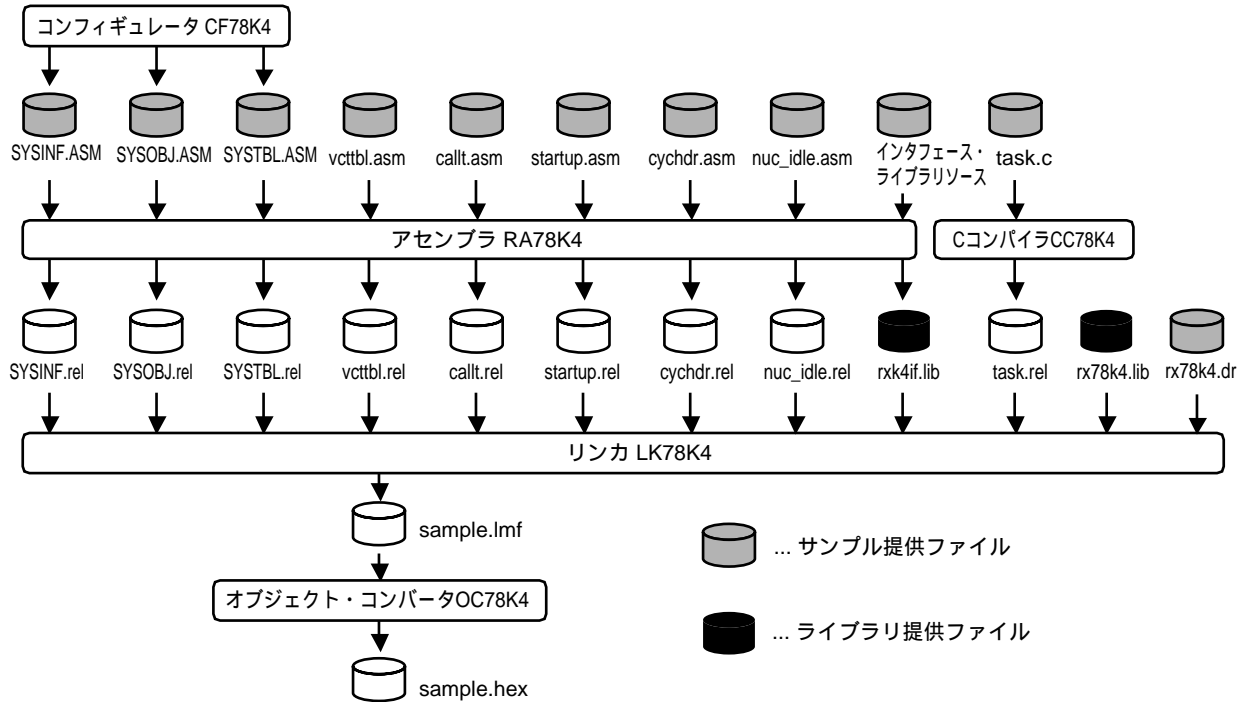
RX78K4は以下の媒体により提供されます。

開発環境	提供形式	提供媒体
Windowsベース	オブジェクト・リリース版 ソース・リリース版	CD-ROM

### 1.3 ロード・モジュール作成手順

アプリケーション・プログラムのロード・モジュール作成の流れを図1-1に示します。各ファイルの詳細については2.2 インストール後のフォルダ構成を参照してください。

図1-1 ロード・モジュール作成手順例



## 第2章 RX78K4のインストール

この章ではRX78K4のインストール手順とインストール後のフォルダ構成について説明します。

オブジェクト・リリース版に収録されているRD78K4については、RX78K4 **リアルタイム・オペレーティング・システム ユーザーズ・マニュアル タスク・デバッグ編** (U15757J) を参照してください。

### 2.1 インストール手順

ホスト・マシンへのインストールは以下に示す手順にて行います。以下の手順はオブジェクト・リリース版、ソース・リリース版で共通です。

#### 1. CD-ROMのセット

RX78K4パッケージが格納されているCD-ROMを、インストールするホスト・マシンのCD-ROMドライブにセットします。ここでは、CD-ROMドライブを“D:”ドライブとします。

#### 2. セットアップ・プログラムの起動

媒体をセットしたドライブが挿入の自動通知に対応していれば、自動的にインストーラが起動します。自動起動しない場合、エクスプローラもしくはコマンド・プロンプトからセットアップ・プログラムを起動します。

**オブジェクト・リリース版** “D:¥install.exe”

**ソース・リリース版** “D:¥install.exe”

以降の操作については、起動したインストーラの指示に従います。

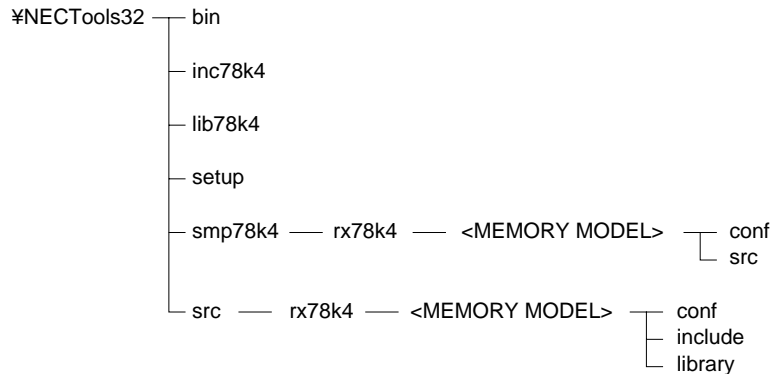
## 2.2 インストール後のフォルダ構成

フォルダ構成と各フォルダの内容について説明します。

### 2.2.1 オブジェクト・リリース版

以下に、RX78K4オブジェクト・リリース版のフォルダ構成を示します。

図2 - 1 RX78K4 (オブジェクト・リリース版) のフォルダ構成



<MEMORY MODEL>にはsmallもしくはlargeのいずれかが入ります。

NECTools32%bin

以下のユーティリティ・ツールのファイルが格納されています。

ファイル名	説明
cf78k4.exe	RX78K4用コンフィギュレータ (ラージ・モデル用)
cf78k4s.exe	RX78K4用コンフィギュレータ (スモール・モデル用)

NECTools32%inc78k4

C言語用インクルード・ファイルが格納されています。

ファイル名	説明
rxk4_def.h	C言語用インクルード・ファイル

NECTools32%lib78k4

RX78K4のライブラリ・ファイルが格納されています。

ファイル名	説明
rx78k4.lib	ニュークリアス・ライブラリ (ラージ・モデル用)
rx78k4s.lib	" (スモール・モデル用)
rxk4if.lib	C言語用インタフェース・ライブラリ (ラージ・モデル用)
rxk4ifs.lib	" (スモール・モデル用)

NECTools32%setup

RX78K4のアンインストールに必要な情報が格納されています。

NECTools32¥smp78k4¥rx78k4¥<MEMORY MODEL>¥conf

RX78K4を使用したサンプル・アプリケーションのビルド環境（プロジェクト・ワークスペース・ファイル）が格納されています。

ファイル名	説明
sample.prw	サンプル・アプリケーションのプロジェクト・ワークスペース・ファイル
sample.prj	サンプル・アプリケーションのプロジェクト・ファイル

NECTools32¥smp78k4¥rx78k4¥<MEMORY MODEL>¥src

RX78K4を使用したサンプル・アプリケーションのソース・ファイルが格納されています。

ファイル名		説明
<MEMORY MODEL> = large (ラージ・モデル)	<MEMORY MODEL> = small (スモール・モデル)	
task.c	同左	タスク部ソース・ファイル (C言語記述)
cychdr.asm	cychdrs.asm	周期起動ハンドラ・ソース・ファイル (アセンブラ記述)
callt.asm	同左	共通分岐処理モジュール, 割り込みハンドラから発行できるシステム・コールのCALLTテーブル
nuc_idle.asm	同左	アイドル状態処理 (READY状態のタスクがひとつも存在しない状態)
vcttbl.asm	同左	割り込みベクタ・テーブル (μPD784218A用)
SYSTBL.ASM	SYSTBLS.ASM	システム・コール・エントリ・テーブル
SYSOBJ.ASM	SYSOBJS.ASM	初期化情報テーブル
SYSINF.ASM	SYSINFS.ASM	
startup.asm	startups.asm	スタート・アップ・ルーチン (μPD784218A用)
rx78k4.dr	同左	リンク・ディレクティブ・ファイル

NECTools32¥src¥rx78k4¥<MEMORY MODEL>¥conf

RX78K4ライブラリのビルド環境が格納されています。オブジェクト・リリース版ではインタフェース・ライブラリのビルド環境のみ提供されます。

ファイル名		説明
<MEMORY MODEL> = large (ラージ・モデル)	<MEMORY MODEL> = small (スモール・モデル)	
rxk4_ifm.bat	rxk4_ifms.bat	インタフェース・ライブラリのビルド用バッチ・ファイル
rxk4_ifa.bat	rxk4_ifas.bat	rxk4_ifm.bat/rxk4_ifms.bat内部からコールされるバッチ・ファイル

NECTools32¥src¥rx78k4¥<MEMORY MODEL>¥include

RX78K4ライブラリのビルドに必要なインクルード・ファイルが格納されます。オブジェクト・リリース版ではインタフェース・ライブラリのビルドに必要なファイルのみ提供されます。

ファイル名		説明
<MEMORY MODEL> = large (ラージ・モデル)	<MEMORY MODEL> = small (スモール・モデル)	
equ.tbl	equs.tbl	EQU定義ファイル (アセンブラ用)

NECTools32¥src¥rx78k4¥<MEMORY MODEL>¥library

C言語インタフェース・ライブラリのソース・ファイルが格納されています。

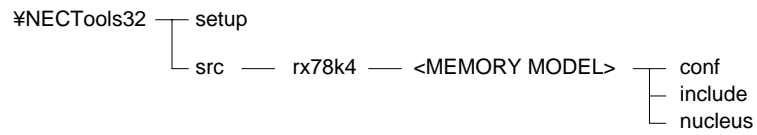
ファイル名		説 明
<MEMORY MODEL> = large (ラージ・モデル)	<MEMORY MODEL> = small (スモール・モデル)	
actcyc.asm	actcycs.asm	C言語インタフェース・ライブラリのソース・ファイル。
canwup.asm	canwups.asm	
chgpri.asm	chgpris.asm	
clrflg.asm	clrflgs.asm	
cpolflg.asm	cpolflgs.asm	
cwaiflg.asm	cwaiflgs.asm	
exttsk.asm	exttsks.asm	
getver.asm	getvers.asm	
iactcyc.asm	iactcycs.asm	
ichgpri.asm	ichgpris.asm	
irotrdq.asm	irotrdqs.asm	
isetflg.asm	isetflgs.asm	
isigsem.asm	isigsems.asm	
isndmsg.asm	isndmsgs.asm	
iwuptsk.asm	iwuptsks.asm	
pgetblk.asm	pgetblks.asm	
polflg.asm	polflgs.asm	
prcvmsg.asm	prcvmsgs.asm	
preqsem.asm	preqsems.asm	
rcvmsg.asm	rcvmsgs.asm	
relblk.asm	relblks.asm	
retint.asm	retints.asm	
retwup.asm	retwups.asm	
rotrdq.asm	rotrdqs.asm	
setflg.asm	setflgs.asm	
sigsem.asm	sigsems.asm	
slptsk.asm	slptsks.asm	
sndmsg.asm	sndmsgs.asm	
statsk.asm	statsks.asm	
tertsk.asm	tertsks.asm	
tsksts.asm	tskstss.asm	
waiflg.asm	waiflgs.asm	
waisem.asm	waisems.asm	
waitsk.asm	waitisks.asm	
wuptsk.asm	wuptsks.asm	



## 2.2.2 ソース・リリース版

以下に、RX78K4ソース・リリース版のフォルダ構成を示します。

図2 - 2 RX78K4 (ソース・リリース版) のフォルダ構成



NECTools32¥setup

RX78K4のアンインストールに必要な情報が格納されています。

NECTools32¥src¥rx78k4¥<MEMORY MODEL>¥conf

RX78K4ライブラリのビルド環境が格納されています。

ファイル名		説 明
<MEMORY MODEL> = large ( ラージ・モデル )	<MEMORY MODEL> = small ( スモール・モデル )	
rxk4_smk.bat	rxk4_smks.bat	インタフェース・ライブラリのビルド用バッチ・ファイル
rxk4_lia.bat	rxk4_lias.bat	rxk4_smk.bat/rxk4_smks.bat内部からコールされるバッチ・ファイル

NECTools32¥src¥rx78k4¥<MEMORY MODEL>¥include

ニュークリアス・ライブラリをソースからビルドする際に必要なインクルード・ファイルが格納されています。

ファイル名		説 明
<MEMORY MODEL> = large ( ラージ・モデル )	<MEMORY MODEL> = small ( スモール・モデル )	
que.mac	ques.mac	OS内部ルーチン用マクロ・ファイル (アセンブラ用)

NECTools32¥src¥rx78k4¥<MEMORY MODEL>¥nucleus

ニュークリアス・ライブラリのソース・ファイル。

ファイル名		説 明
<MEMORY MODEL> = large ( ラージ ・ モデル )	<MEMORY MODEL> = small ( スモール ・ モデル )	
actcyc.asm	actcycs.asm	ニュークリアス・ライブラリのソース・ファイル。
brproc.asm	brprocs.asm	
canwup.asm	canwups.asm	
chgpri.asm	chgpris.asm	
clrflg.asm	clrflgs.asm	
cntsav.asm	cntsavs.asm	
cpolflg.asm	cpolflgs.asm	
cwaiflg.asm	cwaiflgs.asm	
exttsk.asm	exttsks.asm	
getver.asm	getvers.asm	
iactcyc.asm	iactcycs.asm	
ichgpri.asm	ichgpris.asm	
intdsp.asm	intdsps.asm	
irotrdq.asm	irotrdqs.asm	
isetflg.asm	isetflgs.asm	
isigsem.asm	isigsems.asm	
isndmsg.asm	isndmsgs.asm	
iwuptsk.asm	iwuptsks.asm	
pgetblk.asm	pgetblks.asm	
polflg.asm	polflgs.asm	
prcvmsg.asm	prcvmsgs.asm	
preqsem.asm	preqsems.asm	
quecan.asm	quecans.asm	
quein.asm	quein1s.asm	
quein1.asm	queins.asm	
queout.asm	queouts.asm	
queout1.asm	rcvmsgs.asm	
rcvmsg.asm	relblks.asm	
relblk.asm	retints.asm	
retint.asm	retwups.asm	
retwup.asm	rotrdqs.asm	
rotrdq.asm	setflgs.asm	
setflg.asm	sigsems.asm	
sigsem.asm	slptsks.asm	
slptsk.asm	sndmsgs.asm	
sndmsg.asm	statsks.asm	
statsk.asm	sysrts.asm	
sysrt.asm	tertsks.asm	
tertsk.asm	timdsps.asm	
timdsp.asm	tskdsp.asm	
tskdsp.asm	tskstss.asm	
tskst.asm	waiflgs.asm	
waiflg.asm	waisems.asm	
waisem.asm	waitsks.asm	
waitsk.asm	wuptsks.asm	
wuptsk.asm		

### 2.2.3 ドキュメント

以下に、RX78K4ドキュメントのフォルダ構成を示します。

図2 - 3 RX78K4 (ドキュメント) のフォルダ構成

¥NECTools32 — doc

NECTools32¥doc

RX78K4のドキュメントが格納されています。

## 第3章 コンフィギュレータ

本コンフィギュレータは、RX78K4を利用し、ユーザ・システムを構築する際に、リセット・ルーチンに渡すべきシステムの初期化情報と、使用するシステムコールのエントリ・テーブルを、開発マシン上で会話形式に作成するユーティリティ・プログラムです。

コンフィギュレータを起動すると、会話形式で初期化情報の入力を促します。入力された情報はテーブル形式の情報に組み立てられ、アセンブラ・ソース・プログラムの形で出力されます。

ユーザの入力すべき情報には、初期化情報とシステムコール・エントリ情報があり、“表3 - 1”、“表3 - 2”にその内容を示します。

表3 - 1 初期化情報

パラメータ名称	説明
システム情報入力	
Kernel_location_address	カーネル・ロケーション・アドレス
Maximum_priority	指定可能最大プライオリティ
メモリプール情報入力	メモリプールID以下は生成数分設定
Memory_pool_count	メモリプールの生成数
Memory_pool_ID	メモリプールID
Memory_block_count	メモリ・ブロックの生成数
Memory_block_size	1メモリ・ブロックのサイズ
タスク情報入力	タスクID以下は生成数分設定
Task_count	タスクの生成数
Task_ID	タスクID
Task_priority	タスク・プライオリティ
Reg_bank_ID	レジスタ・バンクID
Task_start_address	タスク・スタート・アドレス
Initial_stack_pointer	イニシャル・スタック・ポインタ
Stack_size	スタック・サイズ
Initial_task_ID	イニシャル・タスクのタスクID
周期ハンドラ情報入力	周期ハンドラID以下は生成数分設定
Cyc_count	周期ハンドラの生成数
Cyc_ID	周期ハンドラID
Cyc_activation	周期ハンドラの活性状態
Cyc_pointer	周期ハンドラ・アドレス
Cyc_time	待ち時間
イベントフラグ情報入力	イベントフラグID以下は生成数分設定
Eventflag_count	イベントフラグの生成数
Eventflag_ID	イベントフラグID
Eventflag	イベントフラグの初期値
セマフォ情報入力	セマフォID以下は生成数分設定
Semaphore_count	セマフォの生成数
Semaphore_ID	セマフォID
Resource	リソース（資源保持数）
メールボックス情報入力	メールボックスIDは生成数分設定
Mailbox_count	メールボックスの生成数
Mailbox_ID	メールボックスID

表3 - 2 システムコール・エントリ情報

システムコール名称	機能
<b>タスク関連</b>	
sta_tsk (Start task)	タスクの起動
ext_tsk (Exit task)	自タスクの正常終了
ter_tsk (Terminate task)	他タスクの強制終了
chg_pri (Change task priority)	タスク優先度の変更
rot_rdq (Rotate ready queue)	指定優先度のレディキュー - の回転
tsk_sts (Get task status)	タスクの状態を見る
slp_tsk (Sleep task)	タスクを待ち状態へ移行
wai_tsk (Wait for wakeup task)	タスクを一定時間待ち状態へ移行
wup_tsk (Wakeup task)	タスクの起床
can_wup (Cancel wakeup task)	タスクの起床要求を無効にする
<b>イベントフラグ関連</b>	
set_flg (Set eventflag)	イベントフラグのセット
clr_flg (Clear eventflag)	イベントフラグのクリア
wai_flg (Wait eventflag)	イベントフラグを待つ (クリア無し)
cwai_flg (Wait & clear eventflag)	イベントフラグを待つ (クリア有り)
pol_flg (Poll eventflag)	イベントフラグを見る (クリア無し)
cpol_flg (Poll & clear eventflag)	イベントフラグを見る (クリア有り)
<b>セマフォ関連</b>	
sig_sem (Signal semaphore)	セマフォに対する信号操作 (V命令)
wai_sem (Wait on semaphore)	セマフォに対する待ち操作 (P命令)
preq_sem (Poll & request semaphore)	セマフォ資源を得る
<b>メッセージ関連</b>	
snd_msg (Send message to mailbox)	メッセ - ジの送信
rcv_msg (Receive message from mailbox)	メールボックスからの受信を待つ
prcv_msg (Poll & receive message from mailbox)	メッセ - ジを受信する
<b>メモリ関連</b>	
pget_blk (Poll & get fixed-length memory block)	固定長メモリブロックの獲得
rel_blk (Release fixed-length memory block)	固定長メモリブロックの返却
<b>その他</b>	
get_ver (Get version number)	RX78K4のバ - ジョン番号獲得
act_cyc (Activate cyclic handler)	周期ハンドラの活性制御

## 3.1 起 動

本コンフィギュレータは、下記のコマンドを入力することにより起動します。

コンフィギュレータの起動（ラージモデル用）

CF78K4

コンフィギュレータの起動（スモールモデル用）

CF78K4S

コンフィギュレータが起動されると、下記に示すコンフィギュレータ開始メッセージが表示され、コンフィギュレーションの処理が始まります。以後、表示されたメッセージの指示に従ってデータを入力してください。

コンフィギュレータ開始メッセージ（ラージモデル）

```
Realtime Configurator Ver 1.20 [ xx xxx 96 ]  
Copyright ( C ) NEC Corporation 1994   xxxxxxxxxxxx
```

```
*****  
*                                                                 *  
*           --- RX78K/IV Configuration Start ---                *  
*           < Large & Midium Model >                             *  
*                                                                 *  
*****
```

```
Please press return key
```

コンフィギュレータ開始メッセージ (スモールモデル)

```

Realtime Configurator Ver 1.20 [ xx xxx 96 ]
  Copyright ( C ) NEC Corporation 1994   xxxxxxxxxxxx

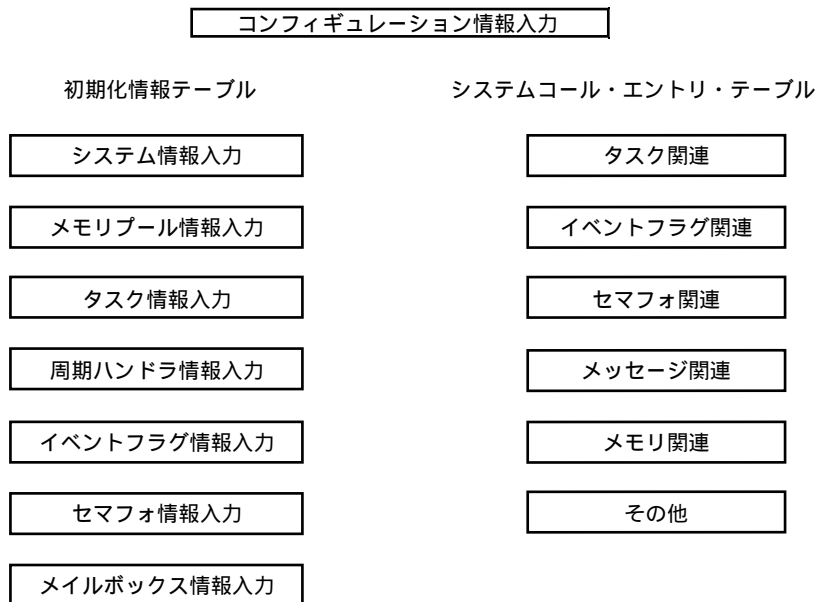
*****
*                                                                              *
*          --- RX78K/IV Configuration Start ---                             *
*          < Small Model >                                                 *
*                                                                              *
*****

Please press return key
    
```

### 3.2 処理の流れ

本コンフィギュレータの処理は、下図に示す順番で行なわれます。

図3 - 1 コンフィギュレータの処理の流れ





### 3.3 データの入力方法

初期化情報のデータ入力

初期化情報のデータ入力は、データを直接書き込む場合と番号を選択する場合の2通りがあります。

#### (1) データを直接書き込む場合

メッセージ中に入力フォーム“XXXX”が表示されている場合は、データの入力を示しますので、10進数を入力してください。

入力フォームが“XXXXh”の時は、アドレス値またはシンボル名の入力を示しますので、16進数または文字列を入力してください。16進数で入力する場合、最後に“h”を入力する必要はありません。

また、入力フォームが“Symbol”の時は、シンボル名の入力を示しますので、文字列を入力してください。

#### (2) 番号を選択する場合

メッセージの先頭の( )内に番号が表示されていますので、該当する番号を選択してください。

尚、適正なデータが入力されない場合は、エラー・メッセージが表示され入力値は無効になりますので再入力してください。

システムコールのエントリ

システムコールのエントリは、表示されたシステムコールを使用する時は“Y” (yes) を、使用しない時は“N” (no) を入力してください。

尚、適正なデータが入力されない場合は、エラー・メッセージが表示され入力値は無効になりますので、再入力してください。

### 3.4 コンフィギュレーション情報入力

コンフィギュレーション情報入力では、初期化情報テーブルと、システムコール・エントリ・テーブルのどちらを作成するかを入力します。

#### (1) コンフィギュレーション情報入力の開始

コンフィギュレーション情報入力では、作成する情報テーブルを、( )内の番号で選択してください。

“E” (exit) を選択すると、コンフィギュレータの処理が終了します。

各情報テーブルの作成が終了すると、再度コンフィギュレーション情報入力のメッセージが表示されま  
す。

#### コンフィギュレーション情報入力テーブル

```

*****
*
*          --- Configuration Information ---
*
*
*****

No.          Item          Current_Value
=====
( 1)         Create Table   SYS_INF_TABLE, SYS_OBJ_TABLE
( 2)         Create Table   SYS_ENT_TABLE

Please Input Item Number or Command
( Change all / Exit ) : [ 番号 or コマンド ]
    
```

SYS\_INF\_TABLE,SYS\_OBJ\_TABLE は、初期化情報テーブルを表わします。

SYS\_ENT\_TABLEは、システムコール・エントリ・テーブルを表わします。

## (2) ファイル名の入力

作成するテーブルを指定した後、下記に示すファイル名入力メッセージが表示されますので、各ファイル名を入力してください。拡張子はコンフィギュレータが付加しますので指定しないでください。付加される拡張子は、“.ASM”です。

既存のファイル名を指定した場合は、各情報テーブルの最終画面が出力されますので、手順に従って情報の更新を行なってください。

ファイル名入力メッセージ

(コンフィギュレーション情報入力で“1”を選択した場合)

```
Please Input Filename ( SYS_INF_TABLE ) : [ ファイル名 ]  
Please Input Filename ( SYS_OBJ_TABLE ) : [ ファイル名 ]
```

入力ファイル名の例として、“sysinf”、“sysobj”を指定します。

ファイル名入力メッセージ

(コンフィギュレーション情報入力で“2”を選択した場合)

```
Please Input Filename ( SYS_ENT_TABLE ) : [ ファイル名 ]
```

入力ファイル名の例として、“systbl”を指定します。

既存のファイルがあり、そのファイルが正しい形式ではなかった場合には、以下のメッセージを出力し、コンフィギュレータは処理を終了します。

ファイル名およびファイル形式を確認して、再度、コンフィギュレータを起動してください。

ファイル読み込み失敗時のメッセージ

```
***** ILLEGAL FILE ERROR *****
```

```
***** BREAK IN *****
```

## 3.5 初期化情報テーブル

初期化情報テーブルとは、リセット・ルーチンに渡すべき初期化情報を保持したテーブルと、指定されたオブジェクトの定義が行なわれているテーブルです。以下に、各情報入力について説明します。

### 3.5.1 システム情報入力

システム情報入力では、RX78K4のカーネル部をメモリ上のどこに配置したかについてと、指定可能な最大プライオリティについて入力します。

#### (1) システム情報入力の開始

システム情報入力は、下記に示すメッセージで開始されます。以後、表示されたメッセージに従ってデータを入力してください。

システム情報入力開始メッセージ

```

*****
*
*           --- System Information ---
*
*
*****

```

#### (2) カーネル部のロケーション・アドレスのデータ設定

カーネル部のロケーション・アドレスの指定を行ないます。RX78K4では、カーネル部の先頭が初期化のためのエントリ・アドレスとなっているので、OS起動時に実行されるリセット・ルーチンは、ここで指定されたアドレスに分岐します。

カーネルのロケーション・アドレスの指定は、リンク・ディレクティブ・ファイル (rx78k4.dr)で行ってください。また、カーネルのロケーション・アドレスが再配置可能(リロケータブル)の場合は、“?sysrt”を指定してください。

カーネル部のロケーション・アドレス

```
Kernel_location_address (XXXXh) : [ アドレス値 ]
```

アドレス値は 0FFFFh までの実アドレス、またはシンボルが入力可。

入力例として、“80” (h) を指定します。

(3) 指定可能最大プライオリティの設定

指定可能な最大プライオリティの設定を行います。RX78K4では、この値によりレディ・キューを作成しますので、ここで指定された値を越えたプライオリティをタスクに指定することはできません。

指定可能最大プライオリティ

Maximum\_priority ( XX ) : [ 最大プライオリティ値 ]

最大プライオリティ値は2～15まで入力可。

入力例として、“4”を指定します。

(4) システム情報入力の終了

データの入力を終了すると、確認のために下記に示すテーブルが表示されます。このテーブルで、入力したデータを確認し、問題がなければ“E”（exit）を指定して、次の情報入力処理を行いません。データの変更を行なう時は、“C”（change all）または（ ）内の該当する番号を選択して、更新処理を行なってください。

システム情報入力テーブル

```

*****
*                                     *
*                               --- System Information ---                       *
*                                     *
*                                     *
*****

```

No.	Item	Current_Value
=====		
( 1 )	Kernel_location_address	<input style="width: 50px;" type="text" value="0080h"/>
( 2 )	Maximum_priority	<input style="width: 50px;" type="text" value="04"/>

Please Input Item Number or Command  
( Change all / Exit ) : [ 番号 or コマンド ]

の部分は、ユーザが設定したデータです。

“h”はコンフィギュレータが付加します。

### 3.5.2 メモリプール情報入力

メモリプール情報入力では、ユーザが使用するメモリプール数、各メモリプールのメモリ・ブロック数と1メモリ・ブロックのサイズの設定を行ないます。

#### (1) メモリプール情報入力の開始

メモリプール情報入力は、下記に示すメッセージで開始されます。以後、表示されたメッセージに従ってデータを入力してください。

メモリプール情報入力開始メッセージ

```

*****
*
*          --- Memorypool Information ---
*
*
*****
    
```

#### (2) メモリプール情報入力のデータ設定

メモリプール情報入力のデータ設定では、生成するメモリプール数、各メモリプールのメモリブロック数、1メモリ・ブロックのサイズを入力します。

“Memory\_pool\_ID”は省略する事が可能ですが、省略した場合はコンフィギュレータが自動的に“mpl1 ”, “mpl2 ”, ……の順番に設定していきます。

メモリプール情報の入力データ

```

Memory_pool_count ( XXX )      : [ 生成する数 ]

Memory_pool_ No.X
Memory_pool_ID ( Symbol )      : [ メモリプ - ルID ]
Memory_block_count ( XXX )     : [ ブロック数 ]
Memory_block_size ( XXX )      : [ ブロックのサイズ ]
    
```

No.Xは、生成する数に対応する番号が表示されます。

カウント及びサイズ値は255まで入力可。

**注意** メモリブロックをメッセージ領域として使用する場合は、メモリブロックのサイズを2の倍数に設定してください。

(3) メモリプール情報入力終了

データの入力を終了すると確認の為、下記に示すテーブルが表示されます。このテーブルで、入力したデータを確認し、問題がなければ“E”(exit)を指定して次の情報入力処理を行ないます。データの変更を行なう時は、表中の左端にある( )内の該当する番号を選択して、更新処理を行なってください。

(change all)を指定した場合は、入力したメモリプール情報のデータ全てに対し更新処理が行なわれま  
ず。

メモリプール情報入力テーブル

```

*****
*
*          --- Memorypool Information ---
*
*****

No.          Item          Current_Value
=====
( 0)         Memory_pool_count          [ 3 ]
( 1)         Memory_pool_No.1
                Memory_pool_ID          [ mpl_id1 ]
                Memory_block_count       [ 5 ]
                Memory_block_size       [ 32 ]
( 2)         Memory_pool_No.2
                Memory_pool_ID          [ mpl1 ]
                Memory_block_count       [ 5 ]
                Memory_block_size       [ 32 ]
( 3)         Memory_pool_No.3
                Memory_pool_ID          [ mpl_id2 ]
                Memory_block_count       [ 5 ]
                Memory_block_size       [ 32 ]

Please Input Item Number or Command
( Change all / Exit ) : [ 番号 or コマンド ]
    
```

の部分は、ユーザが設定したデータです。

### 3.5.3 タスク情報入力

タスク情報入力では、各タスクの初期化情報を設定します。

#### (1) タスク情報入力の開始

タスク情報入力は、下記に示すメッセージで開始されます。以後、表示されたメッセージに従ってデータを入力してください。

タスク情報入力開始メッセージ

```
*****  
*                                           *  
*           --- Task Information ---       *  
*                                           *  
*****
```



**(2) タスク情報入力データの設定**

タスク情報入力データの設定では、生成するタスク数と、個々のタスクの初期化情報を入力します。

“Task\_ID”は省略する事が可能ですが、省略した場合はコンフィギュレータが自動的に“tsk1”，“tsk2”，・・・の順番に設定していきます。

タスク情報の入力データ

Task_count (XXX)	: [生成する数]
Task_No.X	
Task_ID (Symbol)	: [タスクID]
Task_priority (XX)	: [タスクの初期優先度]
Reg_bank_ID (X)	: [レジスタ・バンクID]
Task_start_address (XXXXh)	: [タスク・スタート・アドレス]
Initial_stack_pointer (XXXXh)	: [スタック・ポインタの初期値]
Stack_size (XXXX)	: [スタック・サイズ]

No.Xは、生成する数に対応する番号が表示されます。

カウント値は255まで入力可。

タスクの優先度は2 ~ Maximum\_priorityまで入力可。

レジスタ・バンクIDは1 ~ 7まで入力可。

タスク・スタート・アドレスはラージモデルの場合0FFFFFFhまでの実アドレス、またはシンボルが入力可。

タスク・スタート・アドレスはスモールモデルの場合0FFFFFFhまでの実アドレス、またはシンボルが入力可。

イニシャル・スタック・ポインタはラージモデルの場合0FFFFFFhまでの実アドレス、またはシンボルが入力可。

イニシャル・スタック・ポインタはスモールモデルの場合0FFFFFFhまでの実アドレス、またはシンボルが入力可。

スタック・サイズは9999まで入力可。

**(3) タスク情報入力の終了**

データの入力を終了すると確認の為、下記に示すテーブルが表示されます。

このテーブルで、入力したデータを確認し、問題がなければ“E” (exit)を指定して次の情報入力処理を行ないます。データの変更を行なう時は、表中の左端にある( )内の該当する番号を選択して、更新処理を行なってください。

(change all)を指定した場合は、入力したタスク情報のデータ全てに対し更新処理が行なわれます。

タスク情報入力テーブル

```

*****
*
*          --- Task Information ---
*
*
*****

No.          Item          Current_Value
=====
( 0)         Task_count    [ 3 ]
( 1)         Task_No.1
              Task_ID      [ tsk_id1 ]
              Task_priority [ 2 ]
              Reg_bank_ID   [ 7 ]
              Task_start_address [ 004000h ]
              Initial_stack_pointer [ 00fe00h ]
              Stack_size     [ 64 ]

( 2)         Task_No.2
              Task_ID      [ tsk1 ]
              Task_priority [ 3 ]
              Reg_bank_ID   [ 6 ]
              Task_start_address [ 004100h ]
              Initial_stack_pointer [ intsp1 ]
              Stack_size     [ 64 ]

( 3)         Task_No.3
              Task_ID      [ tsk_tcb3 ]
              Task_priority [ 4 ]
              Reg_bank_ID   [ 5 ]
              Task_start_address [ 004200h ]
              Initial_stack_pointer [ 00fe80h ]
              Stack_size     [ 64 ]

Please Input Item Number or Command
( Change all / Exit ) : [ 番号 or コマンド ]
    
```

の部分は、ユーザが設定したデータです。

“ h ” は、コンフィギュレータが付加します。

**(4) イニシャル・タスクのタスクIDの選択**

イニシャル・タスクのタスクIDを選択します。イニシャル・タスクは、RX78K4の初期化用リセット・ルーチン起動後に、タスク・ディスパッチャにより一番最初に選出されるタスクです。

対象となるタスクは1つだけで、その他のタスクはイニシャル・タスクをきっかけにして起動されます。

選択方法は、入力したタスク情報の中からイニシャル・タスクとなるタスクの番号 (Task\_No.X) を入力してください。

イニシャル・タスクのタスクID

Initial\_task\_ID (Task\_No.X) : **[ 番号 ]**

イニシャル・タスクとして “tsk\_id1” (Task\_No.1) を指定します。この場合の入力値は、“1” です。

### 3.5.4 周期ハンドラ情報入力

周期ハンドラ情報入力では、個々の周期ハンドラの情報を設定します。

#### (1) 周期ハンドラ情報入力の開始

周期ハンドラ情報入力は、下記に示すメッセージで開始されます。以後、表示されたメッセージに従ってデータを入力してください。

周期ハンドラ情報入力開始メッセージ

```

*****
*
*          --- Cyclic Handler Information ---
*
*
*****
    
```

#### (2) 周期ハンドラ情報入力のデータ設定

周期ハンドラ情報入力のデータ設定では、生成する周期ハンドラ数と、個々の周期ハンドラの情報を入力します。

“Cyc\_ID”は省略可能ですが、省略した場合はコンフィギュレータが自動的に、“cyc1”、“cyc2”、…の順番に設定していきます。

周期ハンドラ情報の入力データ

```

Cyc_count (XXX)          : [ 生成する数 ]

Cyc_No.X
  Cyc_ID (Symbol)        : [ 周期ハンドラID ]
  Cyc_activation (X)     : [ 周期ハンドラの活性状態 ]
  Cyc_pointer (XXXXXh)   : [ 周期ハンドラ・アドレス ]
  Cyc_time (XXXX)       : [ 待ち時間 ]
    
```

No.Xは、生成する数に対応する番号が表示されます。

カウント値は255まで入力可。

活性状態は、0または1が入力可。

周期ハンドラ・アドレスは、ラージモデルの場合0FFFFFFhまで入力可、またはシンボルが入力可。

周期ハンドラ・アドレスは、スモールモデルの場合0FFFFFFhまで入力可、またはシンボルが入力可。

待ち時間は9999まで入力可。

(2) 周期ハンドラ情報入力データの設定

データの入力を終了すると、確認のために下記に示すテーブルが表示されます。このテーブルで入力したデータを確認し、問題がなければ“E”(exit)を指定して次の情報入力処理を行います。データの変更を行う時は、表中の左端にある( )内の該当する番号を選択して更新処理を行ってください。“C”を指定した場合は、入力した周期ハンドラ情報の全データに対し更新処理が行われます。

周期ハンドラ情報入力テーブル

```

*****
*
*          --- Cyclic Handler Information ---
*
*
*****
No.          Item          Current_Value
=====
( 0)         Cyc_count     [ 3 ]
( 1)         Cyc_No.1
              Cyc_ID       [ cyc_id1 ]
              Cyc_activation [ 1 ]
              Cyc_pointer   [ cyc_ptr_1 ]
              Cyc_time      [ 100 ]

( 2)         Cyc_No.2
              Cyc_ID       [ cyc1 ]
              Cyc_activation [ 0 ]
              Cyc_pointer   [ 005000h ]
              Cyc_time      [ 9999 ]

( 3)         Cyc_No.3
              Cyc_ID       [ cyc_id2 ]
              Cyc_activation [ 1 ]
              Cyc_pointer   [ cyc_ptr_2 ]
              Cyc_time      [ 50 ]

Please Input Item Number or Command
( Change all / Exit ) : [ 番号 or コマンド ]
    
```

の部分は、ユーザが設定したデータです。

### 3.5.5 イベントフラグ情報入力

イベントフラグ情報入力では、個々のイベントフラグの情報を設定します。

#### (1) イベントフラグ情報入力の開始

イベントフラグ情報入力は、下記に示すメッセージで開始されます。以後、表示されたメッセージに従ってデータを入力してください。

イベントフラグ情報入力開始メッセージ

```

*****
*                                                                *
*                                                                *
*                        --- Eventflag Information ---          *
*                                                                *
*                                                                *
*****
    
```

#### (2) イベントフラグ情報入力のデータ設定

イベントフラグ情報入力のデータ設定では、生成するイベントフラグ数と、個々のイベントフラグ情報を入力します。

“Eventflag\_ID”は省略する事が可能ですが、省略した場合はコンフィギュレータが自動的に“evt1”，“evt2”，・・・の順番に設定していきます。

イベントフラグ情報の入力データ

```

Eventflag_count ( XXX )      : [ 生成する数 ]

Eventflag_ No.X

Eventflag_ID ( Symbol )     : [ イベントフラグID ]
Eventflag ( XXX )          : [ イベントフラグ初期値 ]
    
```

No.Xは、生成する数に対応する番号が表示されます。

カウント値は255まで入力可。

イベントフラグ初期値は、0または1が入力可。

(3) イベントフラグ情報入力終了

データの入力を終了すると確認の為、下記に示すテーブルが表示されます。このテーブルで、入力したデータを確認し、問題がなければ“E”(exit)を指定して次の情報入力処理を行ないます。データの変更を行なう時は、表中の左端にある( )内の該当する番号を選択して、更新処理を行なってください。

(change all)を指定した場合は、入力したイベントフラグ情報のデータ全てに対し更新処理が行なわれます。

イベントフラグ情報入力テーブル

```

*****
*
*
*          --- Eventflag Information ---
*
*
*****
No.          Item          Current_Value
=====
( 0)         Eventflag_count      [ 3 ]
( 1)         Eventflag_No.1
              Eventflag_ID        [ flg_id1 ]
              Eventflag            [ 0 ]
( 2)         Eventflag_No.2
              Eventflag_ID        [ evt1 ]
              Eventflag            [ 1 ]
( 3)         Eventflag_No.3
              Eventflag_ID        [ flg_id2 ]
              Eventflag            [ 1 ]

Please Input Item Number or Command
( Change all / Exit ) : [ 番号 or コマンド ]
    
```

の部分は、ユーザが設定したデータです。

### 3.5.6 セマフォ情報入力

セマフォ情報入力では、各セマフォの情報を設定します。

#### (1) セマフォ情報入力の開始

セマフォ情報入力は、下記に示すメッセージで開始されます。以後、表示されたメッセージに従ってデータを入力してください。

セマフォ情報入力開始メッセージ

```

*****
*                                     *
*                               --- Semaphore Information ---                *
*                                     *
*****

```

#### (2) セマフォ情報入力のデータ設定

セマフォ情報入力のデータ設定では、生成するセマフォ数と、個々の保持資源数情報を入力します。

“ Semaphore\_ID ”は省略する事が可能ですが、省略した場合はコンフィギュレータが自動的に“ sem1 ”，“ sem2 ”，・・・の順番に設定していきます。

セマフォ情報の入力データ

```

Semaphore_count ( XXX )      : [ 生成する数 ]

Semaphore_ No.X
  Semaphore_ID ( Symbol )    : [ セマフォID ]
  Resource ( XXX )          : [ セマフォの保持資源数 ]

```

No.Xは、生成する数に対応する番号が表示されます。

カウント値及び保持資源数は255まで入力可。



(3) セマフォ情報入力終了

データの入力を終了すると確認の為、下記に示すテーブルが表示されます。このテーブルで、入力したデータを確認し、問題がなければ“E”（exit）を指定して次の情報入力処理を行ないます。データの変更を行なう時は、表中の左端にある（ ）内の該当する番号を選択して、更新処理を行なってください。

（change all）を指定した場合は、入力したセマフォ情報のデータ全てに対し更新処理が行なわれます。

セマフォ情報入力テーブル

```

*****
*
*          --- Semaphore Information ---
*
*
*****

No.          Item          Current_Value
=====
( 0)         Semaphore_count  [ 3 ]
( 1)         Semaphore_No.1
              Semaphore_ID    [ sem_id1 ]
              Resource        [ 5 ]

( 2)         Semaphore_No.2
              Semaphore_ID    [ sem_id2 ]
              Resource        [ 3 ]

( 3)         Semaphore_No.3
              Semaphore_ID    [ sem1 ]
              Resource        [ 6 ]

Please Input Item Number or Command
( Change all / Exit ) : [ 番号 or コマンド ]
    
```

の部分は、ユーザが設定したデータです。



(3) メールボックス情報入力終了

データの入力を終了すると確認の為、下記に示すテーブルが表示されます。このテーブルで、入力したデータを確認し、問題がなければ“E” ( exit ) を指定して次の情報入力処理を行ないます。データの変更を行なう時は、表中の左端にある ( ) 内の該当する番号を選択して、更新処理を行なってください。

( change all ) を指定した場合は、入力したメールボックス情報のデータ全てに対し更新処理が行なわれます。

メールボックス情報入力テーブル

```

*****
*
*          --- Mailbox Information ---
*
*****

No.          Item          Current_Value
=====
( 0)         Mailbox_count  [ 3 ]
( 1)         Mailbox_No.1
                Mailbox_ID  [ mbx_id1 ]
( 2)         Mailbox_No.2
                Mailbox_ID  [ mbx1 ]
( 3)         Mailbox_No.3
                Mailbox_ID  [ mbx_id2 ]

Please Input Item Number or Command
( Change all / Exit ) : [ 番号 or コマンド ]
    
```

の部分は、ユーザが設定したデータです。

## 3.6 システムコール・エントリ・テーブル

システムコール・エントリ・テーブルとは、RX78K4で提供されているシステムコールの中から、ユーザが使用したいシステムコールの選択を行い、選択したシステムコールのエントリ・アドレスが格納されるテーブルです。選択されないシステムコールのエントリ・アドレスについては、そのシステムコールが格納されるべき場所に対して“0000h”が設定されます。以下に、各システムコールの選択について説明します。

### 3.6.1 タスク関連システムコール

タスク関連システムコールで使用するシステムコールの選択を行ないます。

#### (1) タスク関連システムコールの選択開始

タスク関連システムコールの選択は、下記に示すメッセージで開始されます。

以後、表示されたメッセージに従ってシステムコールを選択してください。

タスク関連システムコールの選択開始メッセージ

```

*****
*
*          --- Task Management System Call ---
*
*
*****

```

タスク関連システムコールの使用選択

タスク関連システムコールを使用するか、否かについて設定します。使用する時は“Y”（yes）、使用しない時は“N”（no）を入力してください。

尚“N”を入力すると、タスク関連システムコールの選択は行なわず、次のイベントフラグ関連の処理に移ります。

タスク関連システムコールの使用選択

```

Do you use Task system call? ( Yes / No ) : [ " Y " or " N " ]

```

**(3) タスク関連システムコールの選択**

タスク関連システムコールの選択は、RX78K4でサポートされているシステムコールが対象となり、表示されたシステムコールを、使用する時は“ Y ” (yes)、使用しない時は“ N ” (no)を入力してください。

システムコールに関する情報は表3 - 2 システムコール・エントリ情報を参照してください。

## タスク関連システムコールの選択

sta_tsk ( Yes / No )	: [ “ Y ” or “ N ” ]
ext_tsk ( Yes / No )	: [ “ Y ” or “ N ” ]
ter_tsk ( Yes / No )	: [ “ Y ” or “ N ” ]
chg_pri ( Yes / No )	: [ “ Y ” or “ N ” ]
rot_rdq ( Yes / No )	: [ “ Y ” or “ N ” ]
tsk_sts ( Yes / No )	: [ “ Y ” or “ N ” ]
slp_tsk ( Yes / No )	: [ “ Y ” or “ N ” ]
wai_tsk ( Yes / No )	: [ “ Y ” or “ N ” ]
wup_tsk ( Yes / No )	: [ “ Y ” or “ N ” ]
can_wup ( Yes / No )	: [ “ Y ” or “ N ” ]

(4) タスク関連システムコールの選択終了

システムコールの選択が終了すると確認のため、下記に示すエントリ・テーブルが表示されます。

このテーブルで、選択したシステムコールを確認し、変更がなければ“E”(exit)を指定して次のシステムコール選択処理を行ないます。システムコールの設定(“Yes”, “No”)を変更する時は、表中の左端にある( )内の該当する番号を選択して、更新処理を行なってください。

(change all)を指定した場合は、設定したタスク関連システムコールの全てに対し更新処理が行なわれます。

タスク関連システムコールのエントリ・テーブル

```

*****
*
*          --- Mailbox Information ---
*
*****

No.          Item          Current_Value
=====
( 1)         sta_tsk        Yes
( 2)         ext_tsk        Yes
( 3)         ter_tsk        Yes
( 4)         chg_pri        No
( 5)         rot_rdq        No
( 6)         tsk_sts        Yes
( 7)         slp_tsk        Yes
( 8)         wai_tsk        Yes
( 9)         wup_tsk        Yes
(10)         can_wup        Yes

Please Input Item Number or Command
( Change all / Exit ) : [ 番号 or コマンド ]
    
```

の部分は、ユーザが設定したデータです。

### 3.6.2 イベントフラグ関連システムコール

イベントフラグ関連システムコールで使用するシステムコールの選択を行ないます。

#### (1) イベントフラグ関連システムコールの選択開始

イベントフラグ関連システムコールの選択は、下記に示すメッセージで開始されます。以後、表示されたメッセージに従ってシステムコールを選択してください。

イベントフラグ関連システムコールの開始メッセージ

```

*****
*                                     *
*           --- Eventflag Management System Call ---           *
*                                     *
*****
    
```

#### (2) イベントフラグ関連システムコールの使用選択

イベントフラグ関連システムコールを使用するか、否かについて設定します。使用する時は“ Y ” ( yes ) , 使用しない時は “ N ” ( no ) を入力してください。

尚 “ N ” を入力すると、イベントフラグ関連システムコールの選択は行なわず、次のセマフォ関連の処理に移ります。

イベントフラグ関連システムコールの使用選択

```

Do you use Eventflag system call? ( Yes / No ) : [ “ Y ” or “ N ” ]
    
```

#### (3) イベントフラグ関連システムコールの選択

イベントフラグ関連システムコールの選択は、RX78K4でサポートされているシステムコールが対象となり、表示されたシステムコールを、使用する時は“ Y ” ( yes ) , 使用しない時は “ N ” ( no ) を入力してください。

システムコールに関する情報は表3 - 2 システムコール・エントリ情報を参照してください。

イベントフラグ関連システムコールの選択

```

set_flg ( Yes / No )      : [ “ Y ” or “ N ” ]
clr_flg ( Yes / No )     : [ “ Y ” or “ N ” ]
wai_flg ( Yes / No )     : [ “ Y ” or “ N ” ]
cwai_flg ( Yes / No )    : [ “ Y ” or “ N ” ]
pol_flg ( Yes / No )     : [ “ Y ” or “ N ” ]
cpol_flg ( Yes / No )    : [ “ Y ” or “ N ” ]
    
```

(4) イベントフラグ関連システムコールの選択終了

システムコールの選択が終了すると確認のため、下記に示すエントリ・テーブルが表示されます。このテーブルで、選択したシステムコールを確認し、問題がなければ“E”(exit)を指定して次のシステムコール選択処理を行いません。システムコールの設定(“Yes”, “No”)を変更する時は、表中の左端にある( )内の該当する番号を選択して、更新処理を行なってください。

(change all)を指定した場合は、設定したイベントフラグ関連システムコールの全てに対し更新処理が行なわれます。

イベントフラグ関連システムコールのエントリ・テーブル

```

*****
*
*          --- Eventflag Management System Call ---
*
*****

No.          Item          Current_Value
=====
( 1)         set_flg        Yes
( 2)         clr_flg        No
( 3)         wai_flg        Yes
( 4)         cwai_flg       Yes
( 5)         pol_flg        No
( 6)         cpol_flg       Yes

Please Input Item Number or Command
( Change all / Exit ) : [ 番号 or コマンド ]
    
```

の部分は、ユーザが設定したデータです。



### 3.6.3 セマフォ関連システムコール

セマフォ関連システムコールで使用するシステムコールの選択を行ないます。

セマフォ関連システムコールの選択開始

セマフォ関連システムコールの選択は、下記に示すメッセージで開始されます。  
以後、表示されたメッセージに従ってシステムコールを選択してください。

セマフォ関連システムコールの開始メッセージ

```

*****
*
*          --- Semaphore Management System Call ---
*
*
*****

```

#### (2) セマフォ関連システムコールの使用選択

セマフォ関連システムコールを使用するか、否かについて設定します。使用する時は“ Y ” ( yes ) , 使用しない時は“ N ” ( no ) を入力してください。

尚“ N ”を入力すると、セマフォ関連システムコールの選択は行なわず、次のメッセージ関連の処理に移ります。

セマフォ関連システムコールの使用選択

```

Do you use Semaphore system call? ( Yes / No ) : [ “ Y ” or “ N ” ]

```

#### (3) セマフォ関連システムコールの選択

セマフォ関連システムコールの選択は、RX78K4でサポートされているシステムコールが対象となり、表示されたシステムコールを、使用する時は“ Y ” ( yes ) , 使用しない時は“ N ” ( no ) を入力してください。

システムコールに関する情報は表3 - 2 システムコール・エントリ情報を参照してください。

セマフォ関連システムコールの選択

```

sig_sem ( Yes / No )   : [ “ Y ” or “ N ” ]
wai_sem ( Yes / No )   : [ “ Y ” or “ N ” ]
peq_sem ( Yes / No )   : [ “ Y ” or “ N ” ]

```

(4) セマフォ関連システムコールの選択終了

システムコールの選択が終了すると確認のため、下記に示すエントリ・テーブルが表示されます。

このテーブルで、選択したシステムコールを確認し、問題がなければ“E”(exit)を指定して次のシステムコール選択処理を行ないます。システムコールの設定(“Yes”, “No”)を変更する時は、表中の左端にある( )内の該当する番号を選択して、更新処理を行なってください。

(change all)を指定した場合は、設定したセマフォ関連システムコールの全てに対し更新処理が行なわれます。

セマフォ関連システムコールのエントリ・テーブル

```

*****
*
*          --- Semaphore Management System Call ---
*
*****

No.          Item          Current_Value
=====
( 1)         sig_sem        Yes
( 2)         wai_sem        Yes
( 3)         preq_sem       No

Please Input Item Number or Command
( Change all / Exit ) : [ 番号 or コマンド ]
    
```

の部分は、ユーザが設定したデータです。

### 3.6.4 メッセージ関連システムコール

メッセージ関連システムコールで使用するシステムコールの選択を行いません。

#### (1) メッセージ関連システムコールの選択開始

メッセージ関連システムコールの選択は、下記に示すメッセージで開始されます。以後、表示されたメッセージに従ってシステムコールを選択してください。

メッセージ関連システムコールの開始メッセージ

```

*****
*
*          --- Message Management System Call ---
*
*
*****

```

#### (2) メッセージ関連システムコールの使用選択

メッセージ関連システムコールを使用するか、否かについて設定します。使用する時は“Y”（yes）、使用しない時は“N”（no）を入力してください。

尚“N”を入力すると、メッセージ関連システムコールの選択は行わず、次のメモリ関連の処理に移ります。

メッセージ関連システムコールの使用選択

```

Do you use Message system call? ( Yes / No ) : [ " Y " or " N " ]

```

#### (3) メッセージ関連システムコールの選択

メッセージ関連システムコールの選択は、RX78K4でサポートされているシステムコールが対象となり、表示されたシステムコールを、使用する時は“Y”（yes）、使用しない時は“N”（no）を入力してください。

システムコールに関する情報は表3-2 システムコール・エントリ情報を参照してください。

メッセージ関連システムコールのシステムコール選択

```

snd_msg ( Yes / No ) : [ " Y " or " N " ]
rcv_msg ( Yes / No ) : [ " Y " or " N " ]
prcv_msg ( Yes / No ) : [ " Y " or " N " ]

```

(4) メッセージ関連システムコールの選択終了

システムコールの選択が終了すると確認のため、下記に示すエントリ・テーブルが表示されます。

このテーブルで、選択したシステムコールを確認し、問題がなければ“E”(exit)を指定して次のシステムコール選択処理を行ないます。システムコールの設定(“Yes”, “No”)を変更する時は、表中の左端にある( )内の該当する番号を選択して、更新処理を行なってください。

(change all)を指定した場合は、設定したメッセージ関連システムコールの全てに対し更新処理が行なわれます。

メッセージ関連システムコールのエントリ・テーブル

```

*****
*
*          --- Message Management System Call ---
*
*****

No.          Item          Current_Value
=====
( 1)         snd_msg       Yes
( 2)         rcv_msg       Yes
( 3)         prcv_msg      Yes

Please Input Item Number or Command
( Change all / Exit ) : [ 番号 or コマンド ]
    
```

の部分は、ユーザが設定したデータです。

### 3.6.5 メモリ関連システムコール

メモリ関連システムコールで使用するシステムコールの選択を行ないます。

#### (1) メモリ関連システムコールの選択開始

メモリ関連システムコールの選択は、下記に示すメッセージで開始されます。  
以後、表示されたメッセージに従ってシステムコールを選択してください。

メモリ関連システムコールの開始メッセージ

```

*****
*
*          --- Memory Management System Call ---
*
*
*****

```

#### (2) メモリ関連システムコールの使用選択

メモリ関連システムコールを使用するか、否かについて設定します。使用する時は“Y”（yes）、使用しない時は“N”（no）を入力してください。

尚“N”を入力すると、メッセージ関連システムコールの選択は行なわず、次のメモリ関連の処理に移ります。

メモリ関連システムコールの使用選択

```
Do you use Memory system call? ( Yes / No ) : [ " Y " or " N " ]
```

#### (3) メモリ関連システムコールの選択

メモリ関連システムコールの選択は、RX78K4でサポートされているシステムコールが対象となり、表示されたシステムコールを、使用する時は“Y”（yes）、使用しない時は“N”（no）を入力してください。  
システムコールに関する情報は表3-2 システムコール・エントリ情報を参照してください。

メモリ関連システムコールのシステムコール選択

```
pget_blk ( Yes / No )   : [ " Y " or " N " ]
rel_blk ( Yes / No )    : [ " Y " or " N " ]
```

(4) メモリ関連システムコールの選択終了

システムコールの選択が終了すると確認のため、下記に示すエントリ・テーブルが表示されます。

このテーブルで、選択したシステムコールを確認し、問題がなければ“E”(exit)を指定して次のシステムコール選択処理を行ないます。システムコールの設定(“Yes”, “No”)を変更する時は、表中の左端にある( )内の該当する番号を選択して、更新処理を行なってください。

(change all)を指定した場合は、設定したメモリ関連システムコールの全てに対し更新処理が行なわれます。

メモリ関連システムコールのエントリ・テーブル

```

*****
*
*          --- Memory Management System Call ---
*
*****

No.          Item          Current_Value
=====
( 1)         pget_blk      
( 2)         rel_blk       

Please Input Item Number or Command
( Change all / Exit ) : 
    
```

の部分は、ユーザが設定したデータです。

### 3.6.6 その他のシステムコール

その他のシステムコールの選択を行ないます。

#### (1) その他のシステムコールの選択開始

その他のシステムコールの選択は、下記に示すメッセージで開始されます。

以後、表示されたメッセージに従ってシステムコールを選択してください。

その他のシステムコールの開始メッセージ

```

*****
*
*          --- Another Management System Call ---
*
*
*****

```

#### (2) その他のシステムコールの選択

その他のシステムコールの選択は、RX78K4でサポートされているシステムコールが対象となり、使用する時は“Y”（yes）、使用しない時は“N”（no）を入力してください。

システムコールに関する情報は表3-2 システムコール・エントリ情報を参照してください。

その他のシステムコールの選択

```

get_ver ( Yes / No )      : [ " Y " or " N " ]
act_cyc ( Yes / No )     : [ " Y " or " N " ]

```

(3) その他のシステムコールの選択終了

システムコールの選択が終了すると確認のため、下記に示すエントリ・テーブルが表示されます。

このテーブルで、選択したシステムコールを確認し、問題がなければ“E” (exit) を指定してコンフィギュレーション情報入力の処理に戻ります。システムコールの設定 (“Yes”, “No”) を変更する時は、“C” (change all) または、表中の左端にある ( ) 内の該当する番号を選択して、更新処理を行なってください。

その他のシステムコールのエントリ・テーブル

```

*****
*
*          --- Another Management System Call ---
*
*****

No.          Item          Current_Value
=====
( 1)         get_ver       
( 2)         act_cyc       

Please Input Item Number or Command
( Change all / Exit ) : [ 番号 or コマンド ]
    
```

の部分は、ユーザが設定したデータです。



## 第4章 アプリケーション・プログラムの構築

本章では、アプリケーション・プログラムのロード・モジュールの作成方法について説明します。

### 4.1 必要ファイル

ユーザ・アプリケーションのロード・モジュールを作成するには下記のファイルが必要になります。

#### [ ユーザ・タスク関連ファイル ]

- ・初期化情報テーブル
- ・オブジェクト管理テーブル
- ・ユーザ・タスク
- ・システム初期化用リセット・ルーチン分岐プログラム ( startup.asm/startups.asm )

#### [ ニュークリアス関連ファイル ]

- ・システム・コールエントリ・テーブル
- ・ベクタ・テーブル ( vcttbl.asm )
- ・CALLTテーブル ( callt.asm )
- ・ニュークリアス本体 ( rx78k4.lib/rx78k4s.lib )
- ・C言語インターフェース・ライブラリ ( rxk4if.lib/rxk4ifs.lib )

#### (1) 初期化情報テーブル

本ファイルは第2章 RX78K4のインストールで記述したように、コンフィギュレータに情報を入力することによって作成されます。リアルタイムOS初期化時に参照するテーブルです。

#### (2) オブジェクト管理テーブル

本ファイルは初期化情報テーブルを作成すると、自動的に生成されます。このファイルでは、タスク管理テーブルとフラグやセマフォなどのオブジェクトの管理領域が確保されます。

#### (3) ユーザ・タスク

タスクとして登録する処理モジュールのファイルです。このファイル数は複数でもかまいません。

#### (4) システム初期化用リセット・ルーチン分岐プログラム

本ファイルは、ハードウェアの初期化を行ない、初期化情報テーブルの先頭アドレスを参照してニュークリアスに分岐することを目的としています。サンプル・プログラムには、ハードウェアの初期化は記述されていないので、ユーザ・システムに合わせた処理を追加する必要があります。

また、SPの初期値はイニシャル・タスクのスタック領域以外の値を設定してください。

**(5) システムコール・エントリ・テーブル**

本ファイルは、コンフィギュレータでシステムコールを使用するか使用しないかを指示することによって作成されます。添付しているファイルは、全てのシステムコールを使用する(Yes)と指示したものです。

**(6) ベクタ・テーブル**

本ファイルは、ベクタ・テーブルの設定用のテーブルです。タイマ処理ルーチン(タイマ・ディスパッチャ)用ベクタ・アドレスの設定などを行っています。タイマ処理で使用するタイマを変更する場合は、使用するタイマのベクタ・アドレスに“?tmdsp”を指定してください。

**(7) CALLTテーブル**

本ファイルは、callt テーブルに割り込みハンドラからシステムコールの処理へ分岐するためのエントリを設定します。本ファイルを変更することにより、割り込みハンドラから呼ぶシステムコール(ixxx\_xxx)を、callt テーブル内の任意の番地に設定することができます。変更した際は、変更したシステムコールのC言語インタフェース・アセンブラ・ソース・モジュール・ファイル内のcallt参照アドレスも変更してください。

## CALLTテーブル・サンプル・ファイル

```

name    clttbl
;
extrn   brproc
extrn   ichg_pri, irot_rdq, iwup_tsk, iset_flg, isig_sem, isnd_msg, iact_cyc
;
org     0040h
clt_tbl:
dw      brproc           ; 0040h
dw      ichg_pri         ; 0042h
dw      irot_rdq         ; 0044h
dw      iwup_tsk         ; 0046h
dw      iset_flg         ; 0048h
dw      isig_sem         ; 004ah
dw      isnd_msg         ; 004ch
dw      iact_cyc         ; 004eh
;
end

```

上記のサンプル・ファイルをシステムに合わせて変更します。以下に、変更例を記載します。例として、“irot\_rdq”、“isig\_sem”を使用せず、その分の空きテーブルを詰めることにします。

## ・変更例（囲み部分を変更）

```

name    clttbl
;
extrn   brproc
extrn   ichg_pri, irot_rdq, iwup_tsk, iset_flg, isig_sem, isnd_msg, iact_cyc
;
                                削除                削除
org     0040h
clt_tbl:
dw      brproc           ; 0040h
dw      ichg_pri         ; 0042h
dw      irot_rdq         ; 0044h      削除
dw      iwup_tsk         ; 0046h
dw      iset_flg         ; 0048h
dw      isig_sem         ; 004ah      削除
dw      isnd_msg         ; 004ch
dw      iact_cyc         ; 004eh
;
end

```

“irot\_rdq”、“isig\_sem”を削除したことにより、“iwup\_tsk”、“iset\_flg”、“isnd\_msg”のC言語インタフェース・アセンブラ・ソース・モジュール・ファイル内の“CALLT”の参照アドレスを変更します。“iwup\_tsk”のアドレスは「44H」，“iset\_flg”のアドレスは「46H」，“isnd\_msg”のアドレスは、「48H」とします。以下に、“iwup\_tsk”のインタフェース・ライブラリの変更例を示します。

liwuptsk.asm

```
省略  
:  
callt      [ 46h ]  
:  
省略
```

変更後

```
省略  
:  


|       |         |    |
|-------|---------|----|
| callt | [ 44h ] | 変更 |
|-------|---------|----|

  
:  
省略
```

**(8) ニュークリアス本体**

ニュークリアス (リアルタイムOS) 本体です。ライブラリ・ファイルで提供しています。

**(9) C言語インターフェース・ライブラリ**

Cソース記述で発行したシステム・コールをリアルタイムOSが認識できる型に変換するファイルです。  
C言語でタスクを記述した場合、一緒にリンクしてください。

## 4.2 アプリケーション・ロード・モジュール作成手順

次に、アプリケーションのロード・モジュール・ファイルを作成する手順について示します。

ユーザ・タスクをアセンブラまたはC言語で記述する。

コンフィギュレータで初期化情報ファイルとオブジェクト管理テーブル、システムコール・エントリ・テーブルの3つのファイルを作成する。

システム初期化用リセット・ルーチン分岐プログラムをターゲット・システムに合わせて修正する。

ニュークリアス関連ファイルをターゲット・システムに合わせて作成または修正する。

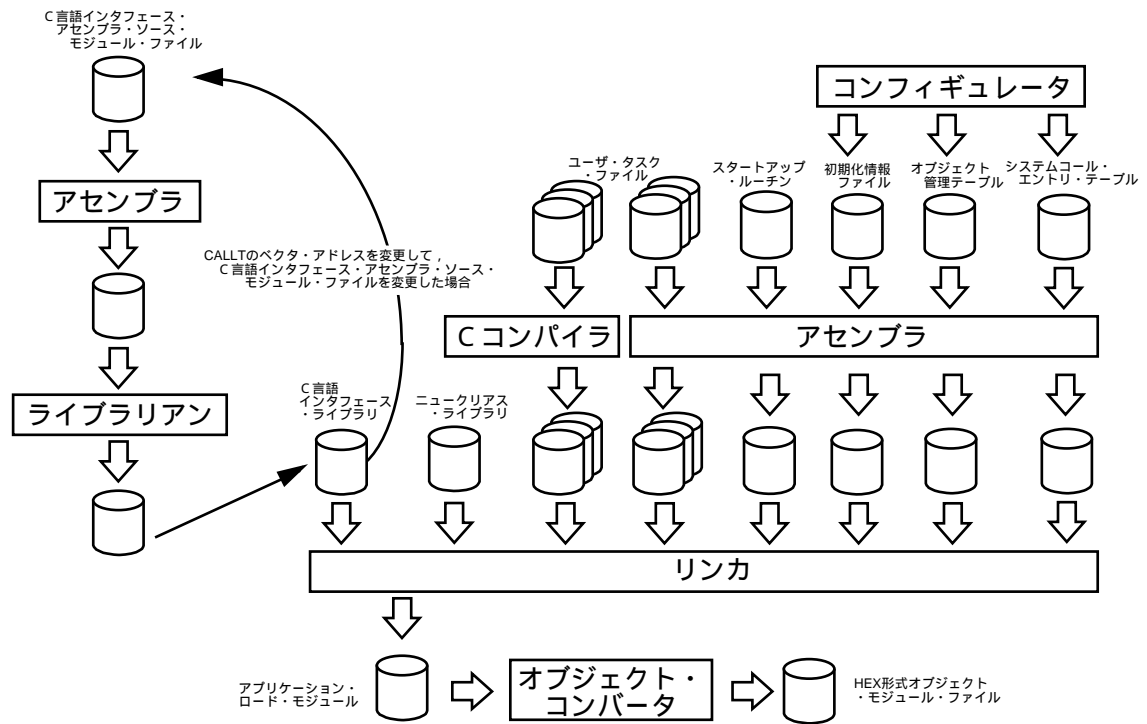
全ファイルをアセンブルまたはコンパイルしてオブジェクト・モジュール・ファイルを生成する。

生成したオブジェクト・モジュール・ファイルとニュークリアスのライブラリをリンクする。このとき、添付の“rx78k4.dr”を用いる。システムコールをC言語で記述している場合は、C言語インタフェース・ライブラリもリンクする。

ユーザが“callt.asm”を変更した場合は、変更したシステム・コールのC言語インタフェース・アセンブラ・ソース・モジュール・ファイルのアセンブルしてください。また必要に応じて、ライブラリアンでライブラリ化してください。

図4 - 1にアプリケーション・ロード・モジュールの作成手順を示します。

図4-1 アプリケーション・ロード・モジュールの作成手順



## 付録A 初期化情報テーブル (ラージモデル)

```

name      sysinf
;
public   sys_inf, inf_end
;

extrn    tsk_id1, mpl_id1, mpl1, mpl_id2, tsk1, tsk_tcb3, intsp1, cyc_id1, cyc1
extrn    cyc_id2, cyc_ptr_1, cyc_ptr_2, flg_id1, evt1, flg_id2, sem_id1, sem_id2
extrn    sem1, mbx_id1, mbx1, mbx_id2
;
;*****
;*
;*      --- System Initialize Information Table ---
;*
;*
;*****
;

cseg
sys_inf :
nuc_loc : dw      0080h          ;/* ニュークリアスのロケーション・アドレス */
;
init_id :  dg      tsk_id1       ;/* イニシャル・タスクのID */
;
mp_cnt  :  db      03h           ;/* メモリ・プールの数 */
mp_id0  :  dg      mpl_id1       ;/* メモリ・プールのID */
b_cnt0  :  db      05h           ;/* メモリ・ブロック数 */
b_siz0  :  db      20h           ;/* メモリ・ブロック・サイズ */
;
mp_id1  :  dg      mpl1          ;
b_cnt1  :  db      05h           ;
b_siz1  :  db      20h           ;
;
mp_id2  :  dg      mpl_id2       ;
b_cnt2  :  db      05h           ;
b_siz2  :  db      20h           ;
;
t_cnt   :  db      03h           ;/* タスクの生成数 */
c_cnt   :  db      03h           ;/* 周期ハンドラの生成数 */
;
t_id0   :  dg      tsk_id1       ;/* タスクのID */
t_pr_rb0 : db      27h           ;/* プライオリティ (2~15) , レジスタ・バンクID (1~7) */
t_adr0  :  dg      004000h       ;/* スタート・アドレス */
t_sp0   :  dg      00FE00h       ;/* スタック・アドレス */
;
t_id1   :  dg      tsk1          ;
t_pr_rb1 : db      36h           ;
t_adr1  :  dg      004100h       ;
t_sp1   :  dg      intsp1        ;
;
t_id2   :  dg      tsk_tcb3      ;
t_pr_rb2 : db      45h           ;
t_adr2  :  dg      004200h       ;

```

```

t_sp2 :    dg    00FE80h
;
c_id0 :    dg    cyc_id1        ; /* 周期ハンドラID */
c_act0 :    db    01h            ; /* 周期ハンドラの活性状態 */
c_ptr0 :    dg    cyc_ptr_1      ; /* 周期ハンドラ・アドレス */
c_tim0 :    dw    0064h          ; /* 待ち時間 */
c_dmy0 :    db    00h
;
c_id1 :    dg    cyc1
c_act1 :    db    00h
c_ptr1 :    dg    005000h
c_tim1 :    dw    270Fh
c_dmy1 :    db    00h
;
c_id2 :    dg    cyc_id2
c_act2 :    db    01h
c_ptr2 :    dg    cyc_ptr_2
c_tim2 :    dw    0032h
c_dmy2 :    db    00h
;
e_cnt :    db    03h            ; /* イベント・フラグ数 */
e_id0 :    dg    flg_id1         ; /* イベント・フラグID */
e_flg0 :    db    00h           ; /* イベント・フラグの初期値 */
;
e_id1 :    dg    evt1
e_flg1 :    db    01h
;
e_id2 :    dg    flg_id2
e_flg2 :    db    01h
;
s_cnt :    db    03h            ; /* セマフォ数 */
s_id0 :    dg    sem_id1         ; /* セマフォID */
s_res0 :    db    05h           ; /* セマフォの初期リソース値 */
;
s_id1 :    dg    sem_id2
s_res1 :    db    03h
;
s_id2 :    dg    sem1
s_res2 :    db    06h
;
m_cnt :    db    03h            ; /* メール・ボックス数 */
m_id0 :    dg    mbx_id1         ; /* メール・ボックスID */
;
m_id1 :    dg    mbx1
;
m_id2 :    dg    mbx_id2
;
max_pri :   db    05h            ; /* (指定可能最大プライオリティ) + 1 */
inf_end :
;
;
end

```



## 付録B オブジェクト管理テーブル (ラージモデル)

```

        name      sysobj
;
        public    ?objhead, rdyqtp, az_arg, wtcbh, wtcbt, runtcb, timnst, wqlnkp, retdata
        public    pribit, cnftbtp, tranos
        public    tsk_di1, tsk1, tsk_tcb3, cyc_id1, cyc1, cyc_id2, flg_id1, evt1, flg_id2
        public    sem_id1, sem_id2, sem1, mbx_id1, mbx1, mbx_id2, mpl_id1, mpl1, mpl_id2
        public    intsp1
;
; *****
; *                                                    *
; *      --- System Initialize Information Data Table --- *
; *                                                    *
; *****
;
sys_dat      dseg      unitp
?objhead :
;
; *      --- OS Work Area Start --- *
;
az_arg :
                ds      2      ; OS管理領域

wtcbh :
                ds      2

wtcbt :
                ds      2

runtcb :
                ds      3

timnst :
                ds      1

wqlnkp :
                ds      3

retdata :
                ds      1

pribit :
                ds      2

cnftbtp :
                ds      3

tranos :
                ds      1
;
tsk_id1 :
                ds      10      ; TCB領域
;
tsk1 :
                ds      10
;
tsk_tcb3 :
                ds      10
;
cyc_id1 :
                ds      10      ; 周期ハンドラ領域

```

```

;
cyc1 :
                ds      10
;
cyc_id2 :
                ds      10
;
flg_id1 :
                ds      4          ; イベントフラグ領域
;
evt1 :
                ds      4
;
flg_id2 :
                ds      4
;
sem_id1 :
                ds      4          ; セマフォ領域
;
sem_id2 :
                ds      4
;
sem1 :
                ds      4
;
mbx_id1 :
                ds      4          ; メールボックス領域
;
mbx1 :
                ds      4
;
mbx_id2 :
                ds      4
;
mpl_id1 :
                ds      164         ; メモリプール領域
;
mpl1 :
                ds      164
;
mpl_id2 :
                ds      164
;
; *
; *      --- Ready queue Area Start ---      *
;
rdyqtp :
                ds      14h         ; レディ・キュー領域
;
; *
; *      --- Stack Area Start ---      *
;
                org      00FDC0h
;
                ds      0040h         ; スタック領域
;
                ds      0040h
intsp1 :

```

```
;
      org      00FE40h
;
      ds       0040h
;
;
      end
```

## 付録C 初期化情報テーブル (スモールモデル)

```

name      sysinf
;
public   sys_inf, inf_end
;

extrn    tsk_id1, mpl_id1, mpl1, mpl_id2, tsk1, tsk_tcb3, intsp1
extrn    cyc_id1, cyc1, cyc_id2, cyc_ptr_1, cyc_ptr_2, flg_id1, evt1, flg_id2
extrn    sem_id1, sem_id2, sem1, mbx_id1, mbx1, mbx_id2
;
; *****
; *
; *      --- System Initialize Information Table ---
; *
; *
; *****
;

cseg
sys_inf :
nuc_loc :  dw      0080h          ; /* ニュークリアスのロケーション・アドレス */
;
init_id :  dw      tsk_id1       ; /* イニシャル・タスクのID */
;
mp_cnt :   db      03h          ; /* メモリ・プールの数 */
mp_id0 :   dw      mpl_id1      ; /* メモリ・プールのID */
b_cnt0 :   db      05h          ; /* メモリ・ブロック数 */
b_siz0 :   db      20h          ; /* メモリ・ブロック・サイズ */
;
mp_id1 :   dw      mpl1         ;
b_cnt1 :   db      05h
b_siz1 :   db      20h
;
mp_id2 :   dw      mpl_id2      ;
b_cnt2 :   db      05h
b_siz2 :   db      20h
;
t_cnt :    db      03h          ; /* タスクの生成数 */
c_cnt :    db      03h          ; /* 周期ハンドラの生成数 */
;
t_id0 :    dw      tsk_id1      ; /* タスクのID */
t_pri0 :   db      02h          ; /* タスクのプライオリティ (2~15) */
t_rbk0 :   db      07h          ; /* レジスタ・バンクID (1~7) */
t_adr0 :   dw      4000h        ; /* スタート・アドレス */
t_sp0 :    dw      0FE00h       ; /* スタック・アドレス */
;
t_id1 :    dw      tsk1         ;
t_pri1 :   db      03h
t_rbk1 :   db      06h
t_adr1 :   dw      4100h
t_sp1 :    dw      intsp1
;

```

```

t_id2 :      dw      tsk_tcb3
t_pri2 :      db      04h
t_rbk2 :      db      05h
t_adr2 :      dw      4200h
t_sp2 :      dw      0FE80h
;
c_id0 :      dw      cyc_id1          ;/* 周期ハンドラID */
c_act0 :      db      01h          ;/* 周期ハンドラの活性状態 */
c_ptr0 :      dw      cyc_ptr_1      ;/* 周期ハンドラ・アドレス */
c_tim0 :      dw      0064h         ;/* 待ち時間 */
c_dmy0 :      db      00h          ;/* リザーブエリア */
;
c_id1 :      dw      cyc1
c_act1 :      db      00h
c_ptr1 :      dw      5000h
c_tim1 :      dw      270Fh
c_dmy1 :      db      00h
;
c_id2 :      dw      cyc_id2
c_act2 :      db      01h
c_ptr2 :      dw      cyc_ptr_2
c_tim2 :      dw      0032h
c_dmy2 :      db      00h
;
e_cnt :      db      03h          ;/* イベント・フラグ数 */
e_id0 :      dw      flg_id1        ;/* イベント・フラグID */
e_flg0 :      db      00h          ;/* イベント・フラグの初期値 */
;
e_id1 :      dw      evt1
e_flg1 :      db      01h
;
e_id2 :      dw      flg_id2
e_flg2 :      db      01h
;
s_cnt :      db      03h          ;/* セマフォ数 */
s_id0 :      dw      sem_id1        ;/* セマフォID */
s_res0 :      db      05h          ;/* セマフォの初期リソース値 */
;
s_id1 :      dw      sem_id2
s_res1 :      db      03h
;
s_id2 :      dw      sem1
s_res2 :      db      06h
;
m_cnt :      db      03h          ;/* メール・ボックス数 */
m_id0 :      dw      mbx_id1        ;/* メール・ボックスID */
;
m_id1 :      dw      mbx1
;
m_id2 :      dw      mbx_id2
;

max_pri :    db      05h          ;/* (指定可能最大プライオリティ) + 1 */
inf_end :

```

;  
;

end

## 付録D オブジェクト管理テーブル (スモールモデル)

```

        name      sysobj
;
        public   ?objhead, rdyqtp, az_arg, wtcbh, wtcbt, runtc, timnst, retdata, tranos
        public   wqlnkp, pribit, cnftbpl
        public   tsk_id1, tsk1, tsk_tcb3, cyc_id1, cyc1, cyc_id2, flg_id1, evt1, flg_id2
        public   sem_id1, sem_id2, sem1, mbx_id1, mbx1, mbx_id2, mpl_id1, mpl1, mpl_id2
        public   fe00, intsp1, fe80
;
; *****
; *
; *      --- System Initialize Information Data Table ---
; *
; *
; *****
;
sys_dat      org      0f700h
?objhead :
;
; *
; *      --- OS Work Area Start ---
; *
;
az_arg :
                ds      2          ; OS管理領域

wtcbh :
                ds      1

wtcbt :
                ds      1

runtc :
                ds      1

timnst :
                ds      1

retdata :
                ds      1

tranos :
                ds      1

wqlnkp :
                ds      1

pribit :
                ds      2

cnftbpl :
                ds      2
;
tsk_id1 :
                ds      8          ; TCB領域
;
tsk1 :
                ds      8
;
tsk_tcb3 :
                ds      8
;
cyc_id1 :

```

```

;
; ds      8          ; 周期ハンドラ領域
;
cyc1 :
; ds      8
;
cyc_id2 :
; ds      8
;
flg_id1 :
; ds      2          ; イベントフラグ領域
;
;
; ds      2
;
;
flg_id2 :
; ds      2
;
;
sem_id1 :
; ds      2          ; セマフォ領域
;
;
sem_id2 :
; ds      2
;
;
sem1 :
; ds      2
;
;
mbx_id1 :
; ds      2          ; メールボックス領域
;
;
mbx1 :
; ds      2
;
;
mbx_id2 :
; ds      2
;
;
mpl_id1 :
; ds     162          ; メモリプール領域
;
;
mpl1 :
; ds     162
;
;
mpl_id2 :
; ds     162
;
;
; *        --- Ready queue Area Start ---          *
;
;
rdyqtp :
; ds      0Ah          ; レディ・キュー領域
;
;
; *        --- Stack Area Start ---          *
;
; org     0FDC0h
;
; ds      0040h          ; スタック領域
;
;
; ds      0040h

```



```
intsp1 :  
;  
           org      0FE40h  
;  
           ds       0040h  
;  
;  
           end
```

## 付録E システムコール・エントリ・テーブル (ラージモデル/スモールモデル)

```

name      systbl
;
public   ent_tbl
;
extrn    sta_tsk, ext_tsk, ter_tsk, tsk_sts, slp_tsk, wai_tsk, wup_tsk, can_wup
extrn    set_flg, wai_flg, cwai_flg, cpol_flg, sig_sem, wai_sem, snd_msg
extrn    rcv_msg, prcv_msg, pget_blk, rel_blk, get_ver, act_cyc
;
;*****
;*                                           *
;*           --- System Call Entry Table --- *
;*                                           *
;*****
;
;
cseg
ent_tbl :
    dw      sta_tsk
    dw      ext_tsk
    dw      ter_tsk
    dw      0000h
    dw      0000h
    dw      tsk_sts
    dw      slp_tsk
    dw      wai_tsk
    dw      wup_tsk
    dw      can_wup
    dw      set_flg
    dw      0000h
    dw      wai_flg
    dw      cwai_flg
    dw      0000h
    dw      cpol_flg
    dw      sig_sem
    dw      wai_sem
    dw      0000h
    dw      snd_msg
    dw      rcv_msg
    dw      prcv_msg
    dw      pget_blk
    dw      rel_blk
    dw      0000h
    dw      get_ver
    dw      act_cyc
;
;
end

```

## 付録F CALLTテーブル

```
name      clttbl
;

extrn     brproc
extrn     ichg_pri, irot_rdq, iwup_tsk, iset_flg, isig_sem, isnd_msg
extrn     iact_cyc

org       0040h
;
; *****
; *                                           *
; *           callt table org ( 0040h )       *
; *                                           *
; *****
;
;
clt_tbl :
    dw     brproc           ; 0040h
    dw     ichg_pri        ; 0042h
    dw     irot_rdq        ; 0044h
    dw     iwup_tsk        ; 0046h
    dw     iset_flg        ; 0048h
    dw     isig_sem        ; 004ah
    dw     isnd_msg        ; 004ch
    dw     iact_cyc        ; 004eh

;
;

end
```

## 付録G Cコンパイラの使用についての注意事項

### 1. Cコンパイラのオプションについて

Cコンパイラを使用する際は、次のオプションは指定しないでください。

メモリ配置指定 : -RD[n], -RS[n]

最適化指定 : -Q, -QR[n], -QX[n], -QV, 省略不可

機能拡張指定 : -ZO, -ZR

また、ロケーション機能指定-CSでは、スモールモデルはcs0、ミディアム/ラージモデルは、cs0fhを前提とします。

### 2. リンク時の注意事項について

リンクする際には、システム初期化用リセット・ルーチン分岐プログラム (startup.rel) を最初にリンクしてください (パラメータ・ファイルには、必ずシステム初期化用リセット・ルーチン分岐プログラム (startup.rel) から記述してください)。

## 付録H エラー・メッセージ一覧表

エラー・メッセージ	内 容
FILE NAME ERROR	ファイル名入力時に拡張子が指定された。
FILE NOT OPEN	特定ファイルがオープンできない。
MEMORY ERROR	メモリを確保できない。
PARAMETER ERROR	入力値の誤りがある。
ILLEGAL FILE ERROR	既存ファイル名の入力時、ファイルフォーマットに誤りがある。
NONE PARAMETER	起動時に、番号選択をしないで“c”が入力された。
BREAK IN	stop key が押された。
DUPLICATE ERROR	重複できない値が入力された。

## 【発 行】

NECエレクトロニクス株式会社

〒211-8668 神奈川県川崎市中原区下沼部1753

電話（代表）：044(435)5111

—— お問い合わせ先 ——

---

## 【ホームページ】

NECエレクトロニクスの情報がインターネットでご覧になれます。

URL(アドレス) <http://www.necel.co.jp/>

---

## 【営業関係，技術関係お問い合わせ先】

半導体ホットライン

(電話：午前 9:00～12:00，午後 1:00～5:00)

電 話 : 044-435-9494

E-mail : [info@necel.com](mailto:info@necel.com)

---

## 【資料請求先】

NECエレクトロニクスのホームページよりダウンロードいただくか，NECエレクトロニクスの販売特約店へお申し付けください。

---