

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い申し上げます。

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・マニュアル

RX850 Ver.3.20

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

対象ツール

RX850 Ver.3.20

〔メ モ〕

目次要約

第1章	概 説	...	13
第2章	インストール	...	16
第3章	システム構築	...	23
第4章	メモリとその容量の見積もり	...	34
第5章	システム・コンフィギュレーション・ファイル	...	38
第6章	コンフィギュレータの操作方法	...	63
付録A	ウインドウ・リファレンス	...	76
索引		...	83

MS-DOS, Windows, Windows NT , およびWindows XP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。

Green Hills Software, MULTIは米国Green Hills Software, Inc.の商標です。

UNIXはX/Openカンパニーリミテッドがライセンスしている米国ならびに他の国における登録商標です。

PC/ATは米国IBM Corp.の商標です。

TRONは , The Real-time Operating system Nucleusの略称です。

ITRONは , Industrial TRONの略称です。

μ ITRONは , Micro Industrial TRONの略称です。

TRON, ITRON , および μ ITRONは , コンピュータの仕様に対する名称であり , 特定の商品ないし商品群を指すものではありません。

μ ITRON3.0仕様は , (社) トロン協会が策定したオープンリアルタイムカーネル仕様です。

μ ITRON3.0仕様の仕様書は , 「 μ ITRON3.0標準ハンドブック改訂新版」(パーソナルメディア株式会社刊) として発行されています。

μ ITRON仕様の著作権は (社) トロン協会に属しています。

- 本資料に記載されている内容は2005年4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

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。

〔メモ〕

は じ め に

対 象 者 このマニュアルは、V850シリーズの各製品の応用システムを設計、開発するユーザを対象としています。

目 的 このマニュアルは次の構成に示すRX850の機能をユーザに理解していただくことを目的としています。

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。

- ・ 解 説
- ・ インストレーション
- ・ システム構築
- ・ メモリとその容量の見積もり
- ・ システム・コンフィギュレーション・ファイル
- ・ コンフィギュレータの操作方法

読 み 方 このマニュアルの読者には、電気、論理回路、マイクロコンピュータ、C言語、アセンブラの一般知識を必要とします。

V850シリーズのハードウェア機能を知りたいとき

→ 各製品の**ユーザーズ・マニュアル ハードウェア編**を参照してください。

V850シリーズの命令機能を知りたいとき

→ V850ES **ユーザーズ・マニュアル アーキテクチャ編** (U15943J) または

V850E1 **ユーザーズ・マニュアル アーキテクチャ編** (U14559J) を参照してください。

凡 例 注 : 本文中につけた注の説明

注意 : 気をつけて読んでいただきたい内容

備考 : 本文の補足説明

数の表記 : 2進数...XXXXまたはXXXXB

10進数...XXXX

16進数...0xXXXX

2のべき数を示す接頭語 (アドレス空間、メモリ容量) :

K (キロ) $2^{10} = 1024$

M (メガ) $2^{20} = 1024^2$

関連資料 このマニュアルを使用する場合は、次の資料もあわせてご覧ください。

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。

開発ツールに関する資料（ユーザズ・マニュアル）

資 料 名		資料番号	
		和文	英文
CA850 Ver.3.00 C コンパイラ・パッケージ	操作編	U17293J	U17293E
	C 言語編	U17291J	U17291E
	アセンブリ言語編	U17292J	U17292E
	リンク・ディレクティブ編	U17294J	U17294E
ID850 Ver.3.00 統合デバッガ	操作編	U17358J	U17358E
ID850NW Ver.3.00, 3.10 統合デバッガ	操作編	U17369J	U17369E
ID850NWC Ver.2.51 統合デバッガ	操作編	U16525J	U16525E
ID850QB Ver.3.10 統合デバッガ	操作編	U17435J	U17435E
SM+ システム・シミュレータ	操作編	U17246J	U17246E
	ユーザ・オープン・インタフェース編	U17247J	U17247E
SM850 Ver.2.50 システム・シミュレータ	操作編	U16218J	U16218E
SM850 Ver.2.00以上 システム・シミュレータ	外部部品ユーザ・オープン・インタフェース仕様編	U14873J	U14873E
RX850 Ver.3.20 リアルタイムOS	基礎編	U13430J	U13430E
	インストレーション編	このマニュアル	U17419E
	テクニカル編	U13431J	U13431E
	タスク・デバッグ編	U17420J	U17420E
AZ850 Ver.3.30 システム・パフォーマンス・アナライザ		U17423J	U17423E
PG-FP4 フラッシュ・メモリ・プログラマ		U15260J	U15260E
TW850 Ver.2.00 性能解析チューニング・ツール		U17241J	U17241E
PM+ Ver.6.00 プロジェクト・マネージャ		U17178J	U17178E

目次

第 1 章 概説	13
1.1 概要	13
1.2 特徴	13
1.3 実行環境	14
1.4 開発環境	14
1.4.1 ハードウェア環境	14
1.4.2 ソフトウェア環境	15
第 2 章 インストール	16
2.1 インストール	16
2.2 アンインストール	16
2.3 フォルダ構成	17
2.3.1 オブジェクト・リリース版 /CA850 対応版	17
2.3.2 オブジェクト・リリース版 /GHS 製コンパイラ対応版	19
2.3.3 ソース・リリース版 /CA850 対応版	21
2.3.4 ソース・リリース版 /GHS 製コンパイラ対応版	22
第 3 章 システム構築	23
3.1 概要	23
3.2 システム・コンフィギュレーション・ファイルの作成	26
3.3 情報ファイルの生成	26
3.4 システム初期化部の作成	27
3.4.1 ブート処理	28
3.4.2 ニュークリアス初期化部	29
3.4.3 初期化ハンドラ	29
3.4.4 割り込みエントリ	29
3.5 アイドル・ハンドラの作成	30
3.6 処理プログラムの作成	31
3.7 初期化データの退避領域の作成	31
3.8 リンク・ディレクティブ・ファイルの作成	32
3.9 ロード・モジュールの作成	33
3.10 システムへの組み込み	33
第 4 章 メモリとその容量の見積もり	34
4.1 .pool0 と .pool1	34
4.2 管理領域のメモリ容量	35
4.3 スタックのメモリ容量	36
4.3.1 タスク・スタック領域	36
4.3.2 システム・スタック領域	36
第 5 章 システム・コンフィギュレーション・ファイル	38
5.1 概要	38
5.2 システム・コンフィギュレーション・ファイルの記述方法	38
5.3 コンフィギュレーション情報	39
5.3.1 リアルタイム OS 情報	39
5.3.2 SIT (System Information Table) 情報	39

5.4	リアルタイム OS 情報の記述形式	41
5.4.1	RX シリーズ情報	41
5.5	SIT 情報の記述形式	42
5.5.1	システム情報	42
5.5.2	システム最大値情報	44
5.5.3	タスク実行権グループ情報	45
5.5.4	タスク情報	46
5.5.5	セマフォ情報	47
5.5.6	イベントフラグ情報	48
5.5.7	1 ビット・イベントフラグ情報	49
5.5.8	メールボックス情報	50
5.5.9	間接起動割り込みハンドラ情報	51
5.5.10	固定長メモリ・プール情報	52
5.5.11	可変長メモリ・プール情報	53
5.5.12	周期起動ハンドラ情報	54
5.6	記述上の注意点	55
5.7	記述例	56
第 6 章 コンフィギュレータの操作方法		63
6.1	概要	63
6.2	起動方法	64
6.2.1	コマンド・ラインからの起動	64
6.2.2	PM+ からの起動	66
6.2.3	コマンド・ファイル	67
6.3	コマンド入力例	68
6.3.1	CA850 対応版の場合	68
6.3.2	GHS 製コンパイラ対応版の場合	69
6.4	メッセージ	70
6.4.1	致命的エラー	71
6.4.2	致命的でないエラー	72
6.4.3	警告	75
付録 A ウインドウ・リファレンス		76
A.1	概要	76
A.2	ウインドウ解説	77
索引		83

図目次

図 2-1	フォルダ構成 (オブジェクト・リリース版 /CA850 対応版)	17
図 2-2	フォルダ構成 (オブジェクト・リリース版 /GHS 製コンパイラ対応版)	19
図 2-3	フォルダ構成 (ソース・リリース版 /CA850 対応版)	21
図 2-4	フォルダ構成 (ソース・リリース版 /GHS 製コンパイラ対応版)	22
図 3-1	システム構築手順 (CA850 対応版)	24
図 3-2	システム構築手順 (GHS 製コンパイラ対応版)	25
図 3-3	システム初期化部の流れ	27
図 5-1	システム・コンフィギュレーション・ファイルの記述イメージ	55
図 5-2	システム・コンフィギュレーション・ファイルの記述例 (CA850 対応版)	59
図 5-3	システム・コンフィギュレーション・ファイルの記述例 (GHS 製コンパイラ対応版)	61
図 6-1	コマンド・ファイルの記述例 (CA850 対応版)	67
図 6-2	メッセージの出力形式	70

表目次

表 3-1	サンプル・プログラムの格納フォルダ	26
表 3-2	システム初期化部の構成	27
表 3-3	パワー・セーブ機能のサンプル	30
表 3-4	アセンブル時の起動オプション	30
表 3-5	処理プログラムの構成	31
表 3-6	RX850 の必須セクション	32
表 3-7	リンク時の参照ライブラリ	33
表 4-1	.pool0 と .pool1 に割りつけられるもの	34
表 4-2	オブジェクト管理領域のサイズ	35
表 6-1	CF850 の動作環境	63
表 A-1	ダイアログの一覧	76

第1章 概説

1.1 概要

マイクロプロセッサは半導体技術の進歩に伴って急速に普及し、今日ではあらゆる分野で利用されるようになってきました。しかし、マイクロプロセッサを取り巻く処理プログラム量は増大し、各種ハードウェアにあわせた固有のプログラムをその都度作成することが困難となりました。

そこで、高性能、多機能化へと進むマイクロプロセッサの能力を完全に引き出すために、オペレーティング・システム (Operating System : OS) の必要性が高まってきました。

しかし、制御機器に組み込まれて使用される制御用の OS は、各々のシステムによってハードウェア構成が異なり、しかも、用途に応じた効率の良い動作が要求されるため、標準的な OS が流通しにくい環境にあります。

NEC エレクトロニクスでは、このような市場の状況に答え、高機能なマイクロプロセッサ、V850 ファミリーが持つ機能を十分に引き出すため、また、将来にわたっての体系的なソフトウェアの構築を支援するために、RX850 を開発、発売しました。

RX850 は、効率の良いリアルタイム、マルチタスク処理環境を提供するとともに対象プロセッサの制御機器分野における応用範囲を拡大することを目的として開発された、組み込み型制御用リアルタイム・マルチタスク OS です。

また、ターゲット・システムに組み込んで使用することを前提として開発されているため、ROM 化を意識し、高速で、かつ、コンパクトな OS となっています。

1.2 特徴

次に、RX850 の特徴を示します。

1) μITRON3.0 仕様に準拠

RX850 は組み込み型制御用 OS のアーキテクチャとして代表的な μITRON3.0 仕様に準拠しており、レベル S までの機能を実装しています。

なお、μITRON3.0 仕様とは組み込み型制御用リアルタイム・システムのオペレーティング・システム仕様です。

2) 高い汎用性

RX850 は μITRON3.0 仕様で規定されているシステム・コールを提供し、アプリケーション・システムの汎用性を高めています。

なお、RX850 では、アプリケーション・システムが使用する機能 (システム・コール) のみを選択することができ、ため、コンパクトでありながらユーザのニーズに最適リアルタイム・マルチタスク OS を構築することができます。

3) リアルタイム、マルチタスク処理の実現

RX850 は完全なリアルタイム、マルチタスク処理を実現するために、豊富な機能を提供しています。

- タスク管理機能
- タスク付属同期機能
- 同期通信機能
- 割り込み管理機能
- メモリ・プール管理機能
- 時間管理機能
- システム管理機能
- スケジューリング機能

4) スケジューリングのロック機能

RX850 はディスパッチ処理 (タスクのスケジューリング処理) をユーザの処理プログラムから禁止 / 再開する機能を提供しています。

5) ROM 化の実現

RX850 はターゲット・システムに組み込んで使用することを想定したリアルタイム・マルチタスク OS であるため、ROM 化を意識し、コンパクトな設計が行われています。

6) オリジナル命令の活用

RX850 は V850 ファミリーの高速な命令実行速度とオリジナル命令の活用により、高速処理を実現しています。

7) ユーティリティ・ツールの提供

RX850 はシステムを構築するうえで有益なユーティリティ・ツールを提供しています。

- コンフィギュレータ CF850
- タスク・デバッガ RD850
- システム・パフォーマンス・アナライザ AZ850

注意 RX850 用タスク・デバッガを “ RD850 ” と呼んでいます。

1.3 実行環境

RX850 は組み込み型制御用の OS として開発されているため、次に示すハードウェアを備えたターゲット・システム上で動作します。

1) 動作対象 CPU

V851, V852, V853, V854,
V850/SA1, V850/SBx, V850/SV1,
V850E/MS1, V850E/MA1, NB85E コア, V850E/IA1

2) 周辺コントローラ

RX850 では、様々な実行環境に対応するためにニュークリアス内のハードウェア依存部を切り出し、サンプル・ソース・ファイルで提供しています。このため、サンプル・ソース・ファイルを各ターゲット・システム用に書き換えることで、特定の周辺コントローラを要求しません。

1.4 開発環境

次に、システムを開発するうえで必要となるハードウェア環境とソフトウェア環境を示します。

1.4.1 ハードウェア環境

1) ホスト・マシン

- IBM PC/AT 互換機

2) インサーキット・エミュレータ

- IE-703002-MC (V851, V852, V853, V854, V850/SA1, V850/SBx, V850/SV1)
- IE-703102-MC (V850E/MS1)
- IE-V850E-MC-A (V850E/MA1, NB85E コア)
- IE-V850E-MC (V850E/IA1)

3) インサーキット・エミュレータ用 I/O ボード

- IE-703003-MC-EM1 (V853)
- IE-703008-MC-EM1 (V854)
- IE-703017-MC-EM1 (V850/SA1)
- IE-703037-MC-EM1 (V850/SBx)
- IE-703040-MC-EM1 (V850/SV1)
- IE-703102-MC-EM1 (V850E/MS 1.5V)
- IE-703102-MC-EM1-A (V850E/MS1 3.3V)
- IE-703107-MC-EM1 (V850E/MA1)
- IE-703116-MC-EM1 (V850E/IA1)

- IE-V850E-MC-EM1-A (NB85E コア 5V)
- IE-V850E-MC-EM1-B (NB85E コア 3.3V)

注意 これらの I/O ボードは、インサーキット・エミュレータと組み合わせて使用します。

4) PC インタフェース・ボード

- IE-70000-PC-IF-C (IBM PC/AT 互換機 ISA バス用)
- IE-70000-CD-IF-A (PCMCIA ソケット用)
- IE-70000-PCI-IF (PCI バス用)

1.4.2 ソフトウェア環境

1) OS (カッコ内はホスト・マシン)

- Windows 98 , Me , NT4.0 , 2000 , XP (IBM PC/AT 互換機)

2) クロス・ツール

- CA850 (NEC エレクトロニクス製)
- CCV850/CCV850E (米国 Green Hills Software, Inc. 製)

3) デバッガ

- ID850 (NEC エレクトロニクス製)
- SM850 (NEC エレクトロニクス製)
- MULTI (米国 Green Hills Software, Inc. 製)
- PARTNER (京都マイクロコンピュータ製)

4) タスク・デバッガ

- RD850 (NEC エレクトロニクス製)

5) システム・パフォーマンス・アナライザ

- AZ850 (NEC エレクトロニクス製)

第2章 インストール

この章では、RX850 のインストール / アンインストールについて説明します。

2.1 インストール

ここでは、RX850 をインストールする方法について説明します。なお、RX850 をインストールし直す場合は、あらかじめアンインストールしてから行ってください。

RX850 の提供媒体は、オブジェクト・リリース版、ソース・リリース版ともに、それぞれ CD-ROM (1 枚) です。また、RX850 のパッケージには、RD850、AZ850、および、オンライン・ヘルプも含まれています。これらも同時にインストールすることができます。

次に、インストールの手順を示します。

- 1) Windows を起動します。
- 2) CD ドライブに CD-ROM を挿入します。そのあと、自動的にセットアップ・プログラムが起動されます。もし自動的にセットアップ・プログラムが起動されない場合は、エクスプローラを起動し、CD ドライブにある INSTALL.EXE をダブルクリックして起動してください。以降、モニタ画面に表示されるメッセージにしたがってインストール作業を実行します。
- 3) Windows の標準アプリケーション Explorer 等を用いて、RX850 の提供媒体に格納されていたファイル群がホスト・マシンにインストールされたことを確認します。なお、各フォルダについては、「[2.3 フォルダ構成](#)」を参照してください。

2.2 アンインストール

ここでは、RX850 をアンインストールする方法について説明します。

- 1) Windows を起動します。
- 2) コントロール・パネルの“アプリケーションの追加と削除”を起動し、アンインストールしたい項目 (“ NEC EL RX850 NEC EL (オブジェクト版)”等)を選択してください。

2.3 フォルダ構成

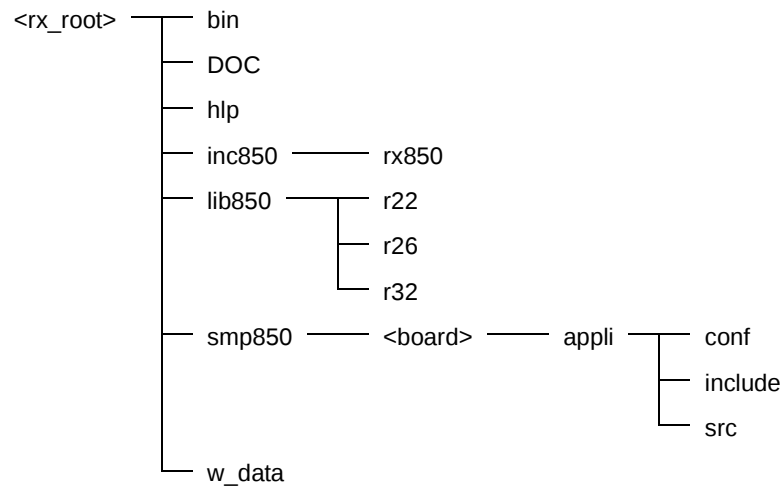
RX850 をインストールした結果、提供媒体から読み込まれた各種ファイルのフォルダ構成について説明します。RX850 の提供形態には、オブジェクト・リリース版とソース・リリース版があり、それぞれ CA850 対応版と GHS 製コンパイラ対応版があります。

- オブジェクト・リリース版 /CA850 対応版
- オブジェクト・リリース版 /GHS 製コンパイラ対応版
- ソース・リリース版 /CA850 対応版
- ソース・リリース版 /GHS 製コンパイラ対応版

2.3.1 オブジェクト・リリース版 /CA850 対応版

図 2-1 に、RX850 の提供媒体に格納されているファイル群 (オブジェクト・リリース版 /CA850 対応版) をホスト・マシン上にインストールした際に生成されるフォルダ構成を示します。

図 2-1 フォルダ構成 (オブジェクト・リリース版 /CA850 対応版)



以下に、各フォルダの詳細を示します。

- 1) <rx_root>
インストール時に指定した「RX850 のインストール・フォルダ」です。
- 2) <rx_root>\bin
RX850 のユーティリティ・ツールが格納されているフォルダです。

cf850.exe	: コンフィギュレータ CF850
rx703100p.dll	: CF850 用 DLL ファイル
- 3) <rx_root>\DOC
RX850 のドキュメント・ファイルが格納されているフォルダです。
- 4) <rx_root>\hlp
RX850 のヘルプ・ファイルが格納されているフォルダです。
- 5) <rx_root>\inc850
RX850 の標準ヘッダ・ファイルが格納されているフォルダです。

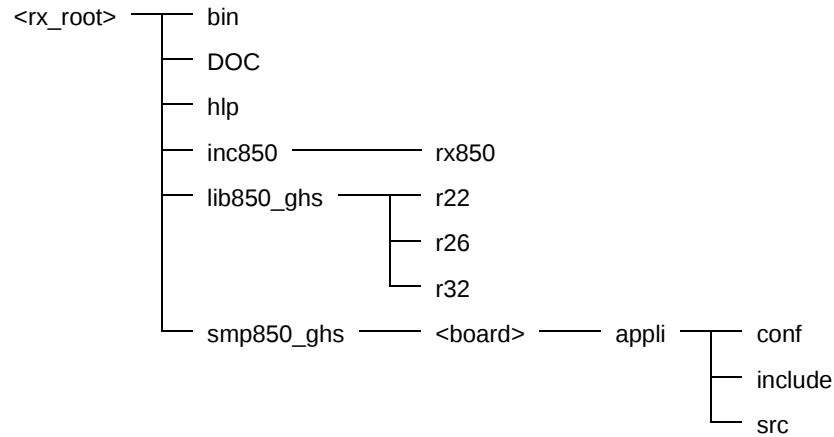
stdrx850.h	: 標準ヘッダ・ファイル (C 言語用)
stdrx850.inc	: 標準ヘッダファイル (アセンブリ言語用)

- 6) <rx_root>\inc850\rx850
RX850 のヘッダ・ファイルが格納されているフォルダです。
- 7) <rx_root>\lib850\vr22
RX850 のライブラリ・ファイル (22 レジスタ・モード) が格納されているフォルダです。
librx.a : ニュークリアス・ライブラリ
- 8) <rx_root>\lib850\vr26
RX850 のライブラリ・ファイル (26 レジスタ・モード) が格納されているフォルダです。
librx.a : ニュークリアス・ライブラリ
- 9) <rx_root>\lib850\vr32
RX850 のライブラリ・ファイル (32 レジスタ・モード) が格納されているフォルダです。
librx.a : ニュークリアス・ライブラリ
- 10) <rx_root>\smp850\<board>
RX850 のサンプル・プログラムが格納されているフォルダです。
- 11) <rx_root>\smp850\<board>\appli\conf
RX850 のロード・モジュールを生成するためのコマンド・ファイルが格納されているフォルダです。
なお、本フォルダのコマンド・ファイルを用いることにより、ロード・モジュール sample.out が本フォルダに生成されます。
sample.prj : ロード・モジュール用プロジェクト・ファイル
sample.prw : ロード・モジュール用ワーク・スペース・ファイル
- 12) <rx_root>\smp850\<board>\appli\include
サンプル・プログラムのヘッダ・ファイルが格納されているフォルダです。
- 13) <rx_root>\smp850\<board>\appli\src
サンプル・プログラムのソース・ファイル, および, リンク・ディレクティブ・ファイルが格納されているフォルダです。
- 14) <rx_root>\lw_data
PM+ 用の RX850 を使用した場合のサンプル・リンク・ディレクティブ・ファイルが格納されているフォルダです。

2.3.2 オブジェクト・リリース版 /GHS 製コンパイラ対応版

図 2-2 に、RX850 の提供媒体に格納されているファイル群 (オブジェクト・リリース版 /GHS 製コンパイラ対応版) をホスト・マシン上にインストールした際に生成されるフォルダ構成を示します。

図 2-2 フォルダ構成 (オブジェクト・リリース版 /GHS 製コンパイラ対応版)



以下に、各フォルダの詳細を示します。

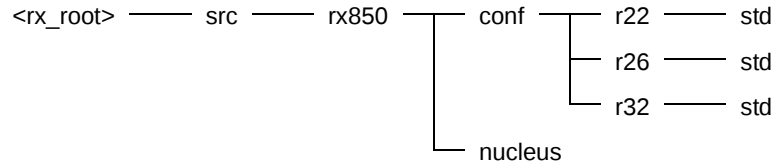
- 1) <rx_root>
インストール時に指定した「RX850 のインストール・フォルダ」です。
- 2) <rx_root>\bin
RX850 のユーティリティ・ツールが格納されているフォルダです。
cf850_ghs.exe : コンフィギュレータ CF850
- 3) <rx_root>\DOC
RX850 のドキュメント・ファイルが格納されているフォルダです。
- 4) <rx_root>\hlp
RX850 のヘルプ・ファイルが格納されているフォルダです。
- 5) <rx_root>\inc850
RX850 の標準ヘッダ・ファイルが格納されているフォルダです。
stdrx850.h : 標準ヘッダ・ファイル
- 6) <rx_root>\inc850\rx850
RX850 のヘッダ・ファイルが格納されているフォルダです。
- 7) <rx_root>\lib850_ghs\r22
RX850 のライブラリ・ファイル (22 レジスタ・モード) が格納されているフォルダです。
librx.a : ニュークリアス・ライブラリ
- 8) <rx_root>\lib850_ghs\r26
RX850 のライブラリ・ファイル (26 レジスタ・モード) が格納されているフォルダです。
librx.a : ニュークリアス・ライブラリ
- 9) <rx_root>\lib850_ghs\r32
RX850 のライブラリ・ファイル (32 レジスタ・モード) が格納されているフォルダです。
librx.a : ニュークリアス・ライブラリ

- 10) <rx_root>\smp850_ghs\<board>
RX850 のサンプル・プログラムが格納されているフォルダです。
- 11) <rx_root>\smp850_ghs\<board>\appli\conf
RX850 のロード・モジュールを生成するためのコマンド・ファイルが格納されているフォルダです。
なお、本フォルダのコマンド・ファイルを用いることにより、ロード・モジュール sample.out が本フォルダに生成されます。
sample.bld : ロード・モジュール用ビルド・ファイル
- 12) <rx_root>\smp850_ghs\<board>\appli\include
サンプル・プログラムのヘッダ・ファイルが格納されているフォルダです。
- 13) <rx_root>\smp850_ghs\<board>\appli\src
サンプル・プログラムのソース・ファイル、および、リンク・ディレクティブ・ファイルが格納されているフォルダです。

2.3.3 ソース・リリース版 /CA850 対応版

図 2-3 に、RX850 の提供媒体に格納されているファイル群 (ソース・リリース版 /CA850 対応版) をホスト・マシン上にインストールした際に生成されるフォルダ構成を示します。

図 2-3 フォルダ構成 (ソース・リリース版 /CA850 対応版)



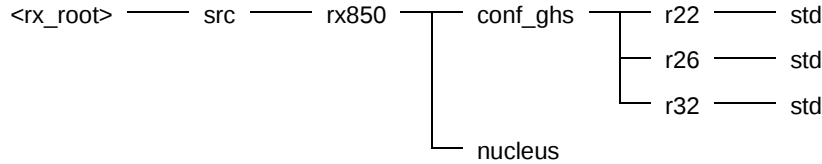
以下に、各フォルダの詳細を示します。

- 1) <rx_root>
インストール時に指定した「RX850 のインストール・フォルダ」です。
- 2) <rx_root>\src\rx850\conf\r22\std
ニュークリアス・ライブラリ (22 レジスタ・モード) を生成するためのコマンド・ファイルが格納されているフォルダです。
なお、本フォルダのコマンド・ファイルを用いることにより、ニュークリアス・ライブラリが <rx_root>\lib850\r22 に生成されます。
makefile : ニュークリアス・ライブラリ librx.a 用メイク・ファイル
- 3) <rx_root>\src\rx850\conf\r26\std
ニュークリアス・ライブラリ (26 レジスタ・モード) を生成するためのコマンド・ファイルが格納されているフォルダです。
なお、本フォルダのコマンド・ファイルを用いることにより、ニュークリアス・ライブラリが <rx_root>\lib850\r26 に生成されます。
makefile : ニュークリアス・ライブラリ librx.a 用メイク・ファイル
- 4) <rx_root>\src\rx850\conf\r32\std
ニュークリアス・ライブラリ (32 レジスタ・モード) を生成するためのコマンド・ファイルが格納されているフォルダです。
なお、本フォルダのコマンド・ファイルを用いることにより、ニュークリアス・ライブラリが <rx_root>\lib850\r32 に生成されます。
makefile : ニュークリアス・ライブラリ librx.a 用メイク・ファイル
- 5) <rx_root>\src\rx850\nucleus
ニュークリアス・ライブラリ librx.a のソース・ファイルが格納されているフォルダです。

2.3.4 ソース・リリース版 /GHS 製コンパイラ対応版

図 2-4 に、RX850 の提供媒体に格納されているファイル群（ソース・リリース版 /GHS 製コンパイラ対応版）をホスト・マシン上にインストールした際に生成されるフォルダ構成を示します。

図 2-4 フォルダ構成（ソース・リリース版 /GHS 製コンパイラ対応版）



以下に、各フォルダの詳細を示します。

- 1) <rx_root>
インストール時に指定した「RX850 のインストール・フォルダ」です。
- 2) <rx_root>\src\rx850\conf_ghs\r22\std
ニュークリアス・ライブラリ (22 レジスタ・モード) を生成するためのコマンド・ファイルが格納されているフォルダです。
なお、本フォルダのコマンド・ファイルを用いることにより、ニュークリアス・ライブラリが <rx_root>\lib850_ghs\r22 に生成されます。
nucleus.bld : ニュークリアス・ライブラリ librx.a 用ビルド・ファイル
- 3) <rx_root>\src\rx850\conf_ghs\r26\std
ニュークリアス・ライブラリ (26 レジスタ・モード) を生成するためのコマンド・ファイルが格納されているフォルダです。
なお、本フォルダのコマンド・ファイルを用いることにより、ニュークリアス・ライブラリが <rx_root>\lib850_ghs\r26 に生成されます。
nucleus.bld : ニュークリアス・ライブラリ librx.a 用ビルド・ファイル
- 4) <rx_root>\src\rx850\conf_ghs\r32\std
ニュークリアス・ライブラリ (32 レジスタ・モード) を生成するためのコマンド・ファイルが格納されているフォルダです。
なお、本フォルダのコマンド・ファイルを用いることにより、ニュークリアス・ライブラリが <rx_root>\lib850_ghs\r32 に生成されます。
nucleus.bld : ニュークリアス・ライブラリ librx.a 用ビルド・ファイル
- 5) <rx_root>\src\rx850\nucleus
ニュークリアス・ライブラリ librx.a のソース・ファイルが格納されているフォルダです。

第 3 章 システム構築

この章では、システムの構築手順について説明しています。

3.1 概要

システム構築とは、RX850 の提供媒体からユーザの開発環境（ホスト・マシン）上に転送したファイル群を用いて、ロード・モジュールを作成したのち、ターゲット・システムへ組み込むことです。
次に、システムを構築する際の手順を示します。

1) システム・コンフィギュレーション・ファイルの作成

2) 情報ファイルの生成

- システム情報テーブル・ファイル
- システム情報ヘッダ・ファイル

備考 情報ファイルは、CF850 を利用して作成します。

3) システム初期化部の作成

- ブート処理
- 初期化ハンドラ
- 割り込みエントリ

4) アイドル・ハンドラの作成

5) 処理プログラムの作成

- タスク
- 割り込みハンドラ
- 周期起動ハンドラ

備考 処理プログラムは、C 言語、または、アセンブリ言語を用いて作成します。

6) 初期化データの退避領域の作成 (CA850 使用時のみ)

7) リンク・ディレクティブ・ファイルの作成

8) ロード・モジュールの作成

9) システムへの組み込み

図 3-1 に、CA850 対応版を使用した場合のシステム構築手順を、図 3-2 に、GHS 製コンパイラ対応版を使用した場合のシステム構築手順を示します。

図 3-1 システム構築手順 (CA850 対応版)

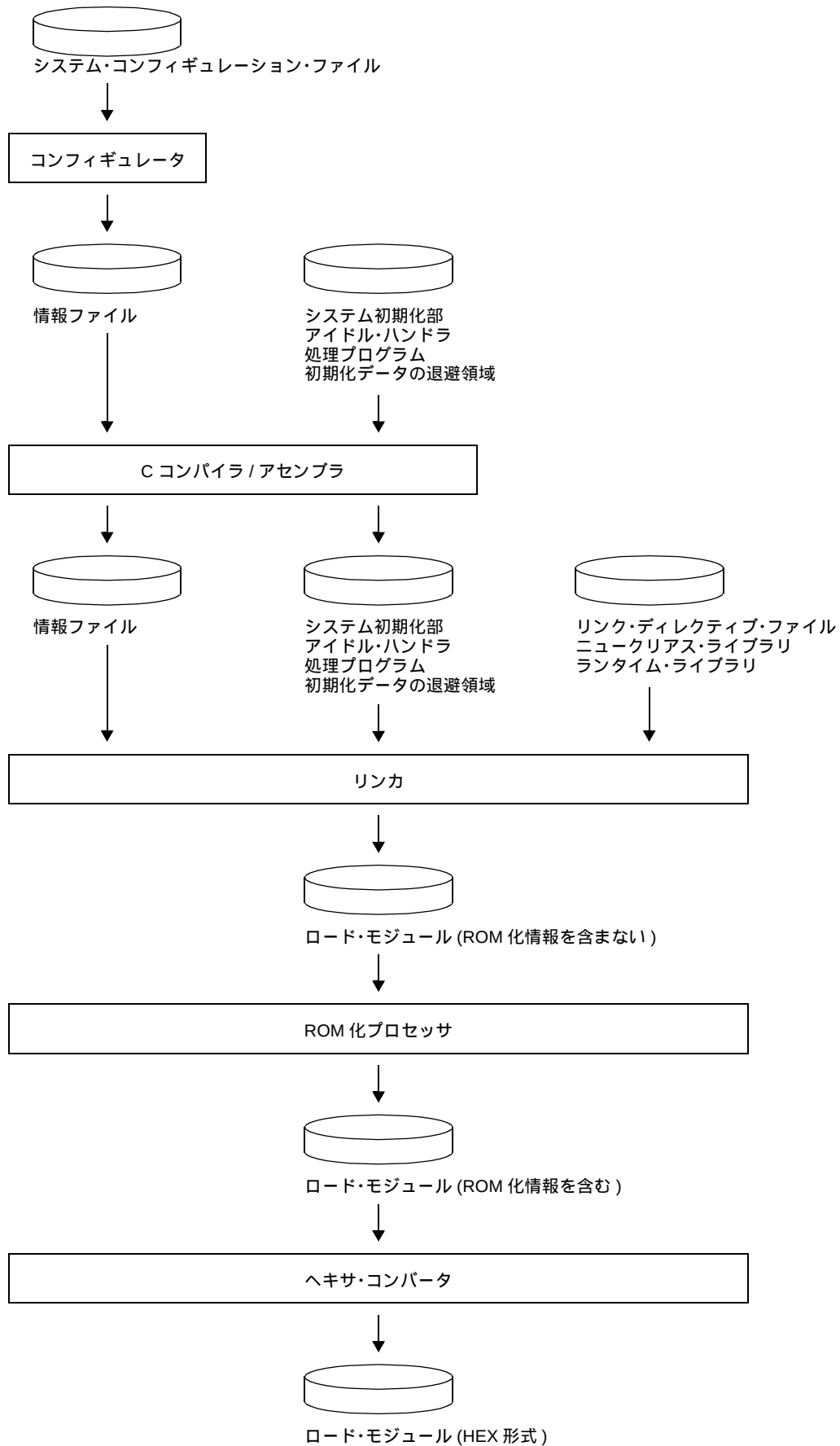
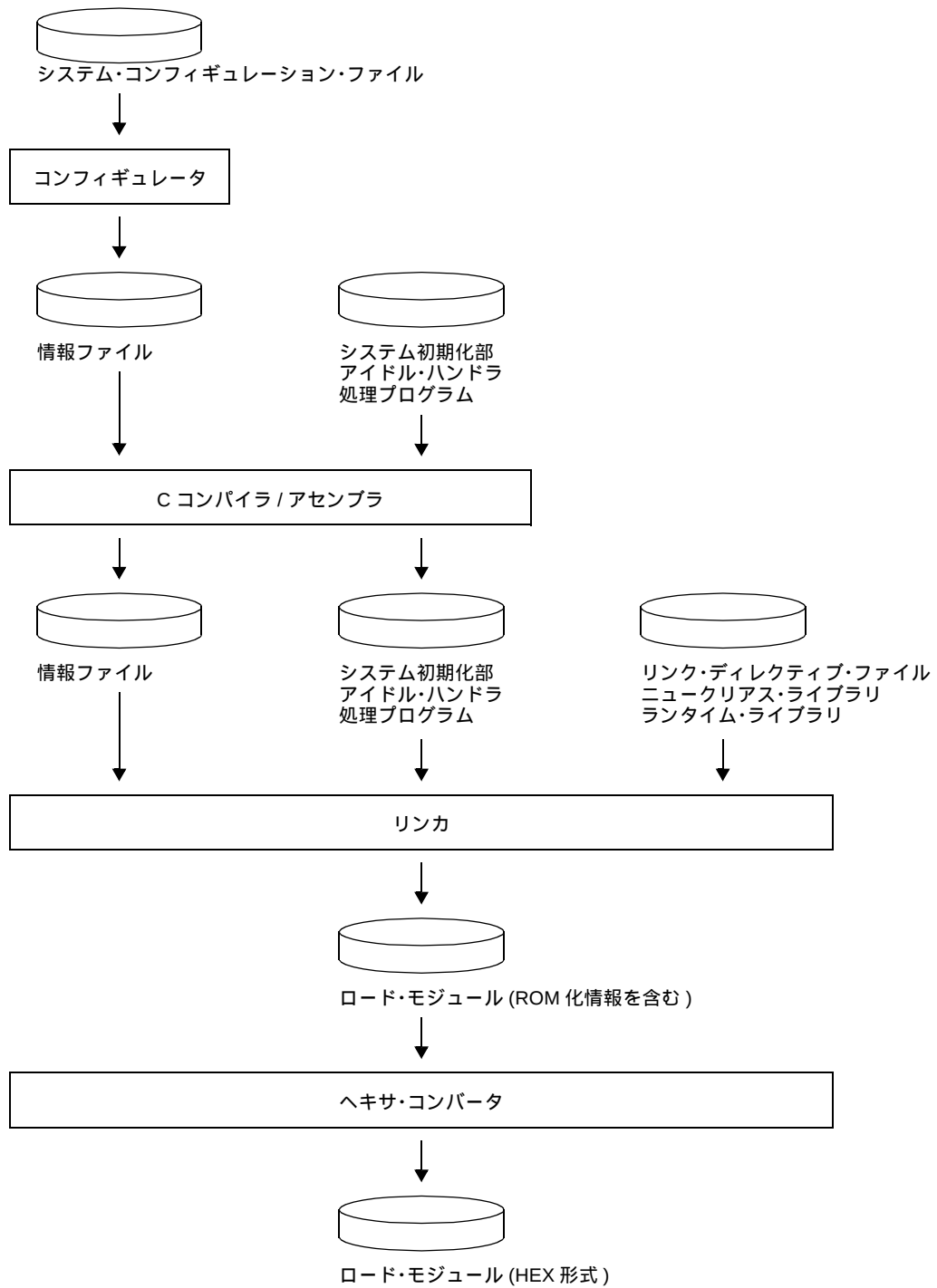


図 3-2 システム構築手順 (GHS 製コンパイラ対応版)



次に、パッケージに添付されているサンプル・プログラムをもとに、システム構築の流れを説明します。
 なお、サンプル・プログラムの格納フォルダは、RX850 のインストール・フォルダを <rx_root> とした場合、次のようになります。

表 3-1 サンプル・プログラムの格納フォルダ

コンパイラ種別	格納フォルダ
CA850	<rx_root>\smp850\rx850\src
GHS 製コンパイラ	<rx_root>\smp850_ghs\rx850\src

それぞれご使用のコンパイラ種別にあわせて参照してください。

3.2 システム・コンフィギュレーション・ファイルの作成

RX850 で使用する各種データを保持した情報ファイル (システム・コンフィギュレーション・ファイル) を作成します。
 この情報ファイルは、CF850 を使用し、次のものを作成するために必要となります。

- システム情報テーブル・ファイル
- システム情報ヘッダ・ファイル

作成方法については、「[3.3 情報ファイルの生成](#)」を参照してください。

システム情報テーブル・ファイルは、タスク、セマフォ、メモリ・プール等の RX850 の資源情報です。システム情報ヘッダ・ファイルは、システム情報テーブル・ファイルで作成したタスクやセマフォ等、資源 ID として指定されたシンボル名と実際の ID 番号を、`#define` 命令で対応させる記述があります。

サンプルのシステム・コンフィギュレーション・ファイルは

- sit.cf

です。システム・コンフィギュレーション・ファイルの内容と書式については、「[第 5 章 システム・コンフィギュレーション・ファイル](#)」を参照してください。

3.3 情報ファイルの生成

「[3.2 システム・コンフィギュレーション・ファイルの作成](#)」で生成したシステム・コンフィギュレーション・ファイルから、CF850 を使用してシステム情報テーブル・ファイルとシステム情報ヘッダ・ファイルを作成します。

それぞれ、次のファイル名を推奨しています。

【システム情報テーブル・ファイル】

- CA850 対応版 : sit.s
- GHS 製コンパイラ対応版 : sit.850

【システム情報ヘッダ・ファイル】

- sys.h

RX850 を使用したアプリケーションを構築する際は、sit.s (、sit.850) をアセンブルして生成したオブジェクトをリンクし、C 言語ソース・ファイルでは sys.h をインクルードする必要があります。

これらのファイルを作るための CF850 の使用方法については、「[第 6 章 コンフィギュレータの操作方法](#)」を参照してください。

3.4 システム初期化部の作成

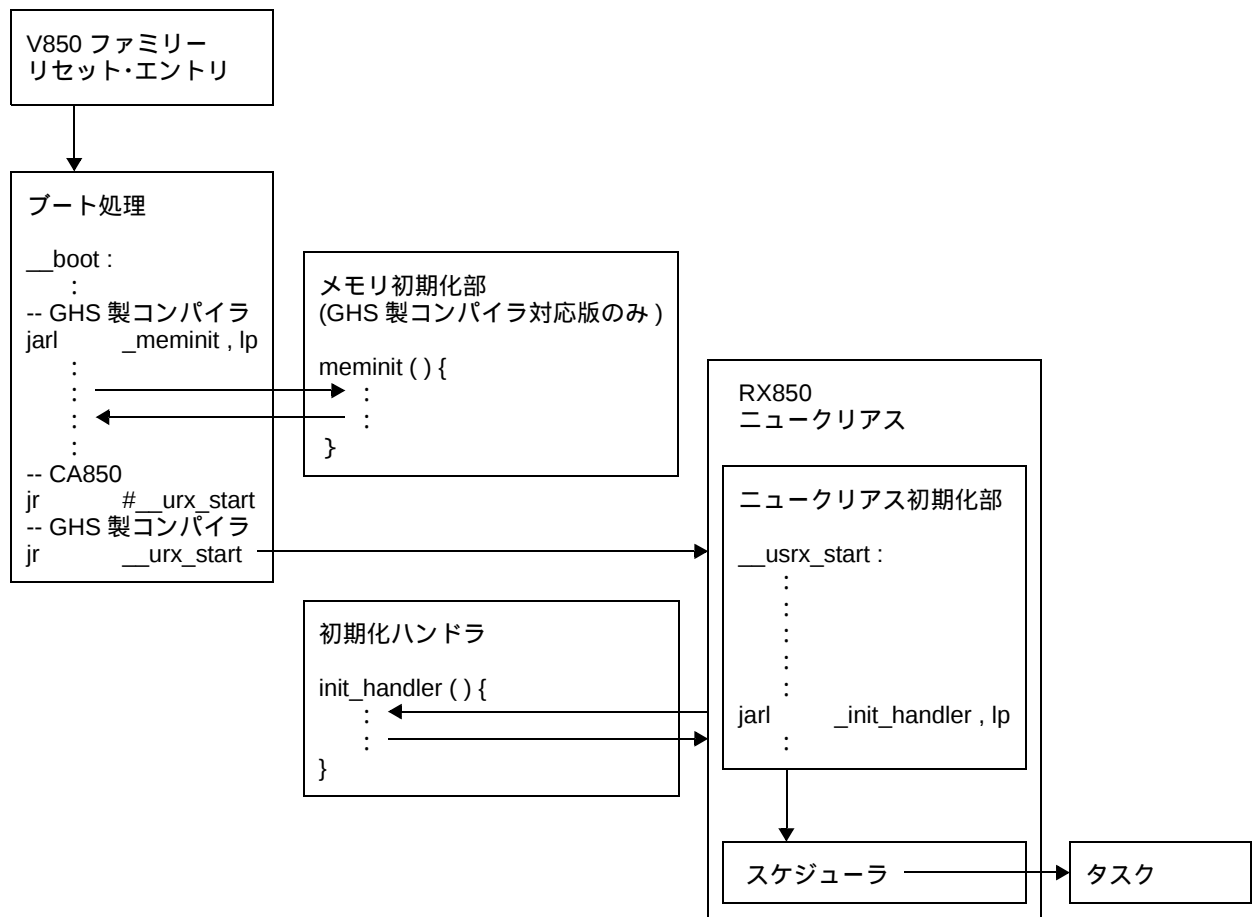
システム初期化部とは、ユーザのターゲット・システムに依存する部分を切り出し、移植性の向上をはかるとともに、カスタマイズを容易にするために用意された関数です。
システム初期化部の構成は、次のとおりです。

表 3-2 システム初期化部の構成

サンプル・ファイル名	種別	関数名	機能
boot.s (CA850 対応版) boot.850 (GHS 製コンパイラ対応版)	ブート処理	start	システムのブート処理
inithdr.c	初期化ハンドラ	init_handler	ハードウェア等の初期化处理
entry.s (CA850 対応版) entry.850 (GHS コンパイラ対応版)	割り込みエントリ	無し	割り込み処理への分岐処理

また、システム初期化部の大まかな流れは次のようになります。

図 3-3 システム初期化部の流れ



3.4.1 ブート処理

ブート処理は、V850 ファミリーのリセット・エントリ（ハンドラ・アドレス：0x0 番地）に割りつけられ、システム初期化処理の中で最初に実行されます。

サンプルでは、boot.c（boot.850）ファイル内の __boot というラベル以降が、ブート処理の本体になります。リセット・エントリからこのラベルにジャンプさせている部分は、CA850 対応版では同じ entry.s ファイルに、GHS 製コンパイラ対応版では entry.850 ファイルにある次の命令に当たります。

【 CA850 対応版 】

.section	"RESET"
.extern	__boot
mov	#__boot, lp
jmp	[lp]

【 GHS 製コンパイラ対応版 】

.org	0x00000000
.extern	__boot
mov	__boot, lp
jmp	[lp]

それぞれ最下行が、ハンドラ・アドレス“0x0 番地”に割りつけられます。したがって、リセットされると、この命令が実行され、__boot ヘジャンプしてブート処理が実行されます。

ブート処理内では、次のことをする必要があります。

- 1) tp（テキスト・ポインタ）、gp（グローバル・ポインタ）、ep（エレメント・ポインタ）の設定
- 2) ブート処理内で使う sp（スタック・ポインタ）の設定
- 3) jr 命令で __urx_start シンボルヘジャンプすることにより、RX850 ニュークリアス初期化部へ制御移行

このほか、GHS 製コンパイラ対応版のサンプルでは、メモリ初期化部である meminit 関数にジャンプする処理（jarl __meminit, lp）が 2) と 3) の間にあります。この meminit 関数内で bss 領域の初期化処理、および、初期値データのコピー処理を行っています。CA850 対応版のサンプルでは、関数ではなく、boot.s 内で RAM 上の bss 領域の初期化（ゼロ・クリア）が行われています。CA850 対応版の初期値データのコピー処理は、初期値データの退避領域の作成（rompcrt.s）と __rcopy 関数の使用によって行われます。詳細については、「CA850 ユーザーズ・マニュアル 操作編」を参照してください。

CA850 対応版のサンプルでは行っていないが、GHS 製コンパイラ対応版のサンプルでは、2) のスタック・ポインタの設定を行っています。ここで設定するスタック・ポインタは、タスクや割り込みハンドラのスタックとは無関係です。RX850 起動後は、タスクや割り込みハンドラ等が使用するスタックは、システム情報テーブル・ファイルにより RX850 自身が管理しており、タスク・スイッチや割り込みにより自動的にスタック・ポインタを切り換えます。つまり、ブート処理内で指定するスタック・ポインタは RX850 起動前に使用するもので、たとえば、関数にジャンプし、その関数内でスタックに保存すべきデータがある場合等に使用されます。サンプルの meminit 関数でスタックを使用する必要があるときには、このスタック・ポインタが使用されます。

そしてブート処理の最後では、3) の処理が必要になります。次の処理を行ってください。

【 CA850 対応版 】

.extern	__urx_start
jr	__urx_start

【 GHS 製コンパイラ対応版 】

.extern	__urx_start
jr	__urx_start

RX850 内部にあるシンボル `__urx_start` 以降が RX850 のニュークリアス初期化処理に当たります。ブート処理が完了した際は `jr` 命令によってニュークリアス初期化処理に移行します。初期化処理では、システム・コンフィギュレーション・ファイルから作ったシステム情報テーブル・ファイルをもとに、資源生成や初期化を行います。

ブート処理の記述は、サンプルのブート処理をもとに、ユーザにあわせた環境へ変更して行くことを推奨します。

3.4.2 ニュークリアス初期化部

ニュークリアス初期化部は、ブート処理終了後に実行される RX850 内部のルーチンです。ここでは、システム・コンフィギュレーション・ファイルより作成されたシステム情報テーブル・ファイルより、RX850 のシステム管理ブロックの作成、タスクやセマフォ、メモリ・プール等の情報を作成、初期化を行っています。

ニュークリアス初期化部において初期化が終了すると、初期化ハンドラが呼び出されます。初期化ハンドラは、RX850 では関数名が `init_handler` (アセンブリ言語記述時は、ラベル `_init_handler`) と決められていますので、この名前の関数を作成する必要があります。この関数については、「[3.4.3 初期化ハンドラ](#)」を参照してください。

初期化ハンドラから制御が戻ると、スケジューラが起動され、RX850 が起動します。

3.4.3 初期化ハンドラ

初期化ハンドラは、ニュークリアス初期化部から呼び出される関数 (ハンドラ) です。ここでは、RX850 起動直前にやっておきたい処理やハードウェア初期化等を記述します。初期化ハンドラ内では、割り込みハンドラ、周期起動ハンドラで発行可能なシステム・コールを発行することが可能です。初期化ハンドラは、引数を持たない FP 型の関数で、関数名は `init_handler` (アセンブリ言語記述時は、ラベル `_init_handler`) と決められていますので、この名前で関数を作成する必要があります。特に処理が必要ない場合でも、何もしない関数として生成してください。また、ハンドラの最後では `return` 命令によって、ニュークリアス初期化処理へ戻してください。

サンプルでは、周辺 I/O の初期化や割り込み制御レジスタの設定、タスクの起動等を行っています。なお、CA850 対応版において、初期化ハンドラ内で初期値データのコピー処理を行うことも可能です。初期値データのコピー処理については、「CA850 ユーザーズ・マニュアル 操作編」を参照してください。

3.4.4 割り込みエントリ

割り込みエントリは、割り込みが発生した際に実行される命令を記述したもので、V850 ファミリーが持つ割り込みハンドラ・アドレスに割りつけられます。ユーザが使用するすべての割り込みに対し、割り込みエントリを定義する必要があります。これらはアセンブリ言語で記述する必要があり、サンプルでは CA850 対応版は `entry.s`、GHS 製コンパイラ対応版では `entry.850` に記述されています。

RX850 の割り込み処理には、直接起動割り込みハンドラと間接起動割り込みハンドラの 2 種類があり、直接起動割り込みハンドラの場合にのみ、エントリを記述します。

エントリは、通常の割り込みエントリと同様に分岐命令を記述します。サンプルでは割り込み “INTP01 (ハンドラ・アドレス: 0x130 番地)” を直接起動割り込みハンドラの例としています。CA850 対応版では `.section` 疑似命令を、GHS 製コンパイラ対応版では `.org` 命令を使用します。それぞれの命令については、「CA850 ユーザーズ・マニュアル アセンブリ言語編」、または、GHS 製コンパイラの言語関連のマニュアルを参照してください。

直接起動割り込みハンドラのエントリは、次のようになります。

【CA850 対応版】

<code>.section</code>	<code>"INTP01"</code>
<code>jr</code>	<code>entry_P01</code>

【GHS 製コンパイラ対応版】

<code>.org</code>	<code>0x00000130</code>
<code>jr</code>	<code>entry_P01</code>

なお、飛び先のラベル `entry_P01` が別ファイルに定義されている場合は、`.extern` 命令を使ってください。

ラベル `entry_P01` は、CA850 対応版では `intp01.c`、GHS 製コンパイラ対応版では `intp01.850` で定義されており、直接起動割り込みハンドラの前処理、後処理の記述 (マクロ記述) をしたあと、ハンドラ本体 `inthdrP01` へジャンプしています。直接起動割り込みハンドラの前処理の記述方法については、「RX850 ユーザーズ・マニュアル 基礎編」を参照してください。

3.5 アイドル・ハンドラの作成

アイドル・ハンドラは、アプリケーションでスケジューリング対象のタスクが 1 つも存在しなくなったときに起動される関数（ハンドラ）です。これを使うことによって、システムのアイドル状態を決めることができ、CPU のパワー・セーブ機能を活用することができます。たとえば、このアイドル・ハンドラ中に CPU を HALT にする処理を記述しておくと、システムのアイドル状態を HALT にすることができます。アイドル・ハンドラは、引数を持たない FP 型の関数で、関数名は `idle_handler`（アセンブリー言語記述時は、ラベル `_idle_handler`）と決められていますので、この名前で関数を作成する必要があります。特にアイドル処理が必要ない場合でも、何もしない関数として生成してください。ただし、このハンドラからの復帰は割り込みによりますので、ハンドラ内は割り込み許可の状態にする処理（EI 命令）を記述する必要があります。

サンプルでアイドル・ハンドラが記述されているファイルは

- `idlehdr.c`

です。

V850 ファミリーでは、パワー・セーブ機能として HALT モード、IDLE モード、STOP モードを持っています。これらのモードにするためには必要な処理があるため、それぞれのモードに対応したサンプルを提供しています。各モードとサンプル・ファイル、および、関数名の対応は、次のようになっています。

表 3-3 パワー・セーブ機能のサンプル

モード	サンプル・ファイル	関数名
HALT モード	halt.s (CA850 対応版) halt.850 (GHS 製コンパイラ対応版)	<code>__halt</code>
IDLE モード	idle.s (CA850 対応版) idle.850 (GHS 製コンパイラ対応版)	<code>__idle</code>
STOP モード	stop.s (CA850 対応版) stop.850 (GHS 製コンパイラ対応版)	<code>__stop</code>

なお、CA850 対応版において、これらのファイルをそのままコンパイル、アセンブルして利用する際、レジスタ・モードを指定して行う必要があります（ファイルの先頭にある `#if ~ #elif ~ #endif` で切り分けられている `.option` 疑似命令が、レジスタ・モードを決定する際に必要な命令です）。デフォルトの場合は、32 レジスタ・モードとしてコンパイル、アセンブルします。22、26 レジスタ・モードで行う場合は、アセンブル時に次の起動オプションをつけてください。

表 3-4 アセンブル時の起動オプション

レジスタ・モード	起動オプション (CA850 対応版)
22 レジスタ・モード	<code>-D__850_22__</code>
26 レジスタ・モード	<code>-D__850_26__</code>

GHS 製コンパイラ対応版に関しては、レジスタ・モードによって切り分けるファイルはありません。

なお、V850 ファミリーのパワー・セーブ機能の内容については、各デバイスの「ユーザーズ・マニュアル ハードウェア編」を参照してください。

3.6 処理プログラムの作成

処理プログラム (アプリケーション本体) を作成します。

RX850 で必要となるアプリケーションの処理単位は、大きく分けて次のようになります。

- タスク
- 直接起動割り込みハンドラ
- 間接起動割り込みハンドラ
- 周期起動ハンドラ

次に、サンプルの内容を示します。

表 3-5 処理プログラムの構成

サンプル・ファイル名	種別	関数名	機能
task.c	タスク	task1 task2	タスク本体
handler.c	間接起動割り込みハンドラ	inthdrP00	ハンドラ処理
	直接起動割り込みハンドラ	inthdrP01	ハンドラ処理
	周期起動ハンドラ	cychdr0 cychdr1	ハンドラ処理

C 言語で記述された処理プログラムで、システム・コールを発行している場合は、RX850 が提供している stdrx850.h というヘッダ・ファイルをインクルードしてください。システム・コールを使用するうえで必要な定義が含まれています。

また、サンプルにあるヘッダ・ファイル sample.h (<rx_root>\smp850\rx850\include にあります) では、上記の stdrx850.h や V850 の周辺 I/O 領域の定義を行っている port.h をさらにインクルードしています。ヘッダ・ファイルは必要に応じて関数が使用する定数等を定義し、プログラム中でインクルードしてください。

3.7 初期化データの退避領域の作成

CA850 を使用するとき、この初期化データの退避領域を作成する必要があります。

これは、初期化データを ROM に格納しておき、プログラム実行前にその初期値を RAM にコピーする必要があるためです。格納元である ROM 領域を確保するのが、この初期化データの退避領域の作成に当たります。

この作成方法については、「CA850 ユーザーズ・マニュアル 操作編」の「ROM 化プロセッサ」の項を参照してください。

GHS コンパイラ対応版では、この処理はサンプルの meminit 関数内で行っています。

3.8 リンク・ディレクティブ・ファイルの作成

リンカがリンクするときに参照するセクション情報、アドレス情報が記述された、リンク・ディレクティブ・ファイル (セクション・マップ・ファイル) を作成します。サンプルでは

- sample.dir (CA850 対応版)
- sample.ld (GHS 製コンパイラ対応版)

RX850 において必須のセクションが存在し、それは次のようになっています。

表 3-6 RX850 の必須セクション

セクション名	領域の種類
.sit	システム情報領域
.text	RX850 システム・コール配置領域
.pool0	システム・メモリ・プール 0
.pool1	システム・メモリ・プール 1

.sit セクション、および、.text セクションはテキスト属性のセクション、.pool0 セクション、および、.pool1 セクションは RAM 領域に配置するセクションです。これらの情報をリンク・ディレクティブ・ファイルに定義する必要があります。

.sit セクションと .pool0 セクションを生成することは必須となっています。また、RX850 はこの 2 つのセクションに対し、0 番地相対で 1 命令アクセスするように作られているため、0 番地から ± 32K バイト内 (0xffff8000 ~ 0x7fff 内) に配置する必要があります。この範囲内に配置しなければ、RX850 の管理ブロック等に正しくアクセスできませんのでご注意ください。V850 ファミリーのアーキテクチャの仕様上、.pool0 は内蔵 RAM に配置することを推奨しています。.pool0 セクション、および、.pool1 セクションについては、「[4.1 .pool0 と .pool1](#)」を参照してください。

サンプルのファイルに記述してあるように、これらの配置情報を定義します。

次に、サンプルの該当部分を示します。

【 CA850 対応版 】

TEXT :	!LOAD	?RX {		
	.sit	= \$PROGBITS	?AX	.sit ;
	.text	= \$PROGBITS	?AX	.text ;
	};			
EDATA :	!LOAD	?RW V0x00100000 {		
	.pool1	= \$NOBITS	?AW	.pool1 ;
	};			
IDATA :	!LOAD	?RW {		
	.pool0	= \$NOBITS	?AW	.pool0 ;
	.data	= \$PROGBITS	?AW	.data ;
	.sdata	= \$PROGBITS	?AWG	.sdata ;
	.sbss	= \$NOBITS	?AWG	.sbss ;
	.bss	= \$NOBITS	?AW	.bss ;
	};			
	:			
	:			

【 GHS 製コンパイラ対応版 】

-sec {			
	.sit	0x0000160	:
	.text		:
	.pool1	0x0100000	:
	.pool0	0x0ffe000	:
	:		:
	:		:

そのほか、.data/.bss セクション等の RAM 領域に関するセクション、const セクション等の ROM 領域に関するセクションについて、必要なものも定義していきます。リンク・ディレクティブ・ファイルの記述は、サンプルをもとに、ユーザにあわせた環境へ変更して行くことを推奨します。

なお、リンク・ディレクティブ・ファイルの記述方法については、「CA850 ユーザーズ・マニュアル 操作編」、または、GHS 製コンパイラの言語関連のマニュアルを参照してください。

3.9 ロード・モジュールの作成

次に、ロード・モジュール(実行可能モジュール)を作成します。

これまで作成した C 言語ソース・ファイル、アセンブリ言語ソース・ファイルをコンパイル、アセンブルし、できたオブジェクト(.o ファイル)群を「[3.8 リンク・ディレクティブ・ファイルの作成](#)」で作成したリンク・ディレクティブ・ファイルに基づいてリンクします。

RX850 を使用したアプリケーションをリンクする際、次のライブラリの参照が必要になります。

表 3-7 リンク時の参照ライブラリ

ライブラリ	内容
librx.a	ニュークリアス・ライブラリ

このライブラリは 22, 26, 32 レジスタ・モード毎に用意されているので、それぞれ使用しているレジスタ・モードにあったライブラリを使用してください。

リンクに成功すると実行可能モジュール(.out ファイル)が完成します。この段階でデバッガに読み込んでアプリケーションを実行させることができます。

リンクにより作成されたロード・モジュールは、初期化データに関しては RAM に正しく配置された状態になっています。そのため、CA850 対応版版の場合で、かつ、初期化データがアプリケーションに存在する場合、初期化データ退避領域を確保し、コピー・ルーチンを組み込んだモジュールを作成する必要があります。この場合は、リンクが生成したロード・モジュールに対し、ROM 化プロセッサを通したロード・モジュールを作成する必要があります。

ROM 化プロセッサの使用方法、および、コピー・ルーチンについては、「CA850 ユーザーズ・マニュアル 操作編」の「ROM 化プロセッサ」の項を参照してください。

3.10 システムへの組み込み

次に、完成したロード・モジュールをシステムに組み込みます。

組み込むためには、「[3.9 ロード・モジュールの作成](#)」で作成したロード・モジュールをヘキサ・ファイルにコンバートする必要があります。

CA850 対応版、および、GHS 製コンパイラ対応版のそれぞれに用意されているヘキサ・コンバータを使って、必要なフォーマットのヘキサ・ファイルにします。そして、ROM ライタ等を使用してシステムに組み込みます。

第 4 章 メモリとその容量の見積もり

この章では、RX850 におけるメモリ (RAM) の管理と、使用する容量について説明します。

4.1 .pool0 と .pool1

RX850 で使用する RAM 領域には、次の 2 つのセクションがあります。

- .pool0 セクション (システム・メモリ・プール 0)
- .pool1 セクション (システム・メモリ・プール 1)

これら 2 つのセクションをリンク・ディレクティブ・ファイルで RAM 上に定義します。

.pool0 セクションに割りつけられるものと、.pool1 セクションに割りつけられるものは、次のとおりです

表 4-1 .pool0 と .pool1 に割りつけられるもの

セクション	割りつけられるもの
.pool0	システム・ベース・テーブル レディー・キュー 各種管理ブロック タスク・スタック 割り込みハンドラ用スタック (システム・スタック) 可変長メモリ・プール 固定長メモリ・プール
.pool1	タスク・スタック 割り込みハンドラ用スタック (システム・スタック) 可変長メモリ・プール 固定長メモリ・プール

.pool0 セクションには、RX850 のシステム情報が入るため、作成が必須のセクションとなります。.pool1 セクションは、.pool0 セクションだけでまかなえる場合は、作成しなくても良いセクションです。

タスク・スタック、割り込みハンドラ用スタック、メモリ・プールを .pool0、および、.pool1 どちらの領域から確保するかは、システム・コンフィギュレーション・ファイルにて指定します。指定方法については、「[第 5 章 システム・コンフィギュレーション・ファイル](#)」を参照してください。

なお、.pool0 セクションは配置場所に関して制限があります。RX850 が .pool0 セクションへアクセスする際、0 番地相対で 1 命令アクセスするため、0 番地から ± 32K バイト内 (0xffff8000 ~ 0x7fff 内) に配置する必要があります。この範囲内に配置しなければ、RX850 の管理ブロック等に正しくアクセスできません。.pool0 セクションは RAM に配置する必要があるため、この範囲内にある内蔵 RAM 領域に配置することを推奨しています。

.pool1 セクションに関しては、この制限はありません。

4.2 管理領域のメモリ容量

RX850 における、システム・ベース・テーブル、レディー・キュー、各種管理ブロックが使用するサイズについて説明します。これらは .pool0 セクションより確保されます。表 4-2 に各オブジェクト 1 つ当たりが使用する管理領域のサイズとその算出方法を示します。

表 4-2 オブジェクト管理領域のサイズ

オブジェクト	使用する管理領域のサイズ (1 オブジェクト当たり)	サイズの算出方法
システム・ベース・テーブル	20 バイト	システムの状態(タスクの数等)によらず一定のサイズ(20 バイト)を持ちます。
レディー・キュー	2 ~ 32 バイト	システム・コンフィギュレーション・ファイルで指定した優先度のレベル数 + 1 (単位: バイト)
タスク実行権	18 バイト	次に示す場合、さらに使用領域が追加されます。 タスク実行権グループを利用している場合 【追加サイズの算出方法】 “タスク実行権待ちの可能性あり”と指定した実行権グループ数 * 1 (単位: バイト) 可変長メモリ・プール, または, イベントフラグのいずれか, または, 両方を利用している場合 【追加サイズの算出方法】 全タスク実行権の数 * 8 (単位: バイト)
タスク	9 バイト	-
イベントフラグ	5 バイト	-
1 ビット・イベントフラグ	2 バイト	-
セマフォ	2 バイト	-
メールボックス	8 バイト	TA_MPRI 属性のメールボックスを使用する場合は, TA_MPRI 属性のメールボックス 1 つにつき, さらに 1 バイトの領域を使用します。
可変長メモリ・プール	16 バイト	管理領域のほかにメモリ・プール領域が必要です。
固定長メモリ・プール	4 バイト	管理領域のほかにメモリ・プール領域が必要です。
周期起動ハンドラ	10 バイト	-

4.3 スタックのメモリ容量

4.3.1 タスク・スタック領域

各タスクのスタック領域は、タスクを通常の C 関数と考えた場合に使用する領域（レジスタ変数の退避領域等）のサイズに、次に示す 1) ~ 3) のサイズを加算して確保します。

1) コンテキスト領域

システム・コールを発行する等してディスパッチが発生した場合、次に示す各レジスタ・モードに応じたサイズのスタックを消費します。

レジスタ 32 本モード : 136 バイト
 レジスタ 26 本モード : 112 バイト
 レジスタ 22 本モード : 96 バイト

2) 可変長メモリ・プール作業領域

RX850 はシステム・コール処理中には、スタックを使用しないことを前提としていますが、可変長メモリ・プールを使用している場合に限りタスク・スタックを最大 12 バイト使用します。

次に、該当する（可変長メモリ・プールを使用している）システム・コールを示します。

`get_blk`, `pget_blk`, `tget_blk`, `rel_blk`, `ter_tsk`, `rel_wai`

3) 割り込み発生時のテンポラリ・レジスタ退避領域

割り込み発生時にテンポラリ・レジスタを退避する領域として、次に示す各レジスタ・モードに応じたサイズのスタックを消費します。

なお、`reti` (RTOS_IntExit) で復帰する直接起動の割り込みハンドラを記述した場合には、そのハンドラが消費するスタックのサイズも加算する必要があります。

レジスタ 32 本モード : 68 バイト
 レジスタ 26 本モード : 56 バイト
 レジスタ 22 本モード : 48 バイト

4.3.2 システム・スタック領域

RX850 では、各種ハンドラ（割り込みハンドラ、周期起動ハンドラ、初期化ハンドラ、アイドル・ハンドラ等）の起動時には、スタック領域をタスク・スタックからハンドラ・スタックに切り替えて使用します。

次に、必要となるスタック・サイズの算出方法について示します。

1) 初期化ハンドラ

初期化ハンドラを通常の C 関数として考え、その関数が消費するサイズのスタックが必要です。

ただし、初期化ハンドラ実行中は割り込みは禁止されており、実行中に他ハンドラや他タスクに処理が切り替わることはないので、割り込みハンドラ等で消費するスタックのサイズが十分に大きい場合は、考慮する必要はありません。

2) アイドル・ハンドラ

アイドル・ハンドラを通常の C 関数として考え、その関数が消費するサイズのスタックが必要です。アイドル・ハンドラ実行中は割り込みが発生することを前提としていますので、割り込みハンドラや周期起動ハンドラが消費するスタックとは別にサイズを確保しなければなりません。

3) 割り込みハンドラ

割り込みハンドラは起動した時点で 8 バイトのハンドラ・スタックを消費しています。割り込みハンドラがネストする場合は、ネストする毎にさらに 8 バイトのスタックを消費します。つまり、各割り込みハンドラが C 関数として消費するスタックのサイズ（ネストする場合は、その合計）に加えて“割り込みハンドラの最大ネスト数 * 8” バイトのスタック領域を必要とします。

4) タイマ・ハンドラ

タイマ・ハンドラは間接起動の割り込みハンドラとして実装されており、タイマ・ハンドラが起動すると 20 バイトのスタックを消費します。

ただし、タイマ・ハンドラ実行中にタイマ・ハンドラがネストしても新たな 20 バイトのスタック領域は必要ありません。

- 5) 周期起動ハンドラ
周期起動ハンドラはタイマ・ハンドラから呼び出される C 関数として実装されます。したがって、この C 関数が消費するサイズのスタックを消費します。
- 6) 可変長メモリ・プール
タスク・スタックと同様に、可変長メモリ・プールに関するシステム・コールを発行した場合、最大 12 バイトのスタックを消費します。ただし、ハンドラから発行する場合は、ネストしたハンドラからこれらのシステム・コールが再び発行される場合があり、最大で “ 割り込みハンドラの最大ネスト数 * 12 バイト ” を加算する必要があります。

第 5 章 システム・コンフィギュレーション・ファイル

この章では、システム・コンフィギュレーション・ファイルと、その記述方法について説明します。

5.1 概要

RX850 を使ったシステムを構築する場合、必要とする各種データ（システム情報や資源情報等）を保持した情報が必要となります。この情報を格納したファイルは、システム情報テーブル・ファイルと呼ばれます。

システム情報テーブル・ファイルは、アセンブリ言語で記述されており、規定された形式のデータ羅列となっています。したがって、システム情報テーブル・ファイルは、各種エディタで記述することは可能ですが、変更や追加が非常に難しいものであるため、記述にはかなりの労力を必要とします。

そこで、RX850 では、コンフィギュレータ CF850 というアプリケーションを提供しています。

これは、システム・コンフィギュレーション・ファイルという、RX850 のシステムや資源に関する情報を記述した、独自のフォーマットのファイルを、システム情報テーブル・ファイルに変換するアプリケーションです。つまり、ユーザはシステム・コンフィギュレーション・ファイルを作成し、CF850 によって、システム情報テーブル・ファイルを得ることができます。

CF850 は、システム・コンフィギュレーション・ファイルからシステム情報テーブル・ファイルとシステム情報ヘッダ・ファイルの 2 種類のファイルを出力します。システム情報ヘッダ・ファイルは、作成したタスクやセマフォ等、資源 ID として指定されたシンボル名と実際の ID 番号を、#define 命令で対応させる記述があるファイルです。

CF850 の起動方法については、「第 6 章 コンフィギュレータの操作方法」を参照してください。

次に、システム・コンフィギュレーション・ファイルの記述方法について説明します。

5.2 システム・コンフィギュレーション・ファイルの記述方法

CF850 に入力するためのシステム・コンフィギュレーション・ファイルの記述方法を示します。

1) 文字コード

システム・コンフィギュレーション・ファイルは、ASC コードで記述します。

なお、英小文字と英大文字は区別されます。

また、単語（数値、シンボル名、キー・ワード等）の区切りはスペース、または、タブで行い、文の区切りは改行で行います。

注意 日本語は、EUC コード、または、シフト JIS コードのものをコメント内でのみ記述することができます。

2) 数値

数値は、指定のない限り 32 ビット幅 (0x0 ~ 0xffffffff) の範囲で記述することができます。

3) シンボル名

シンボル名は、31 文字までの英数字で記述することができます。

ただし、シンボル名の先頭 1 文字は、“_”，または、“英字” でなければなりません。

4) コメント

システム・コンフィギュレーション・ファイルでは、“--” 以降行末までがコメントとして扱われます。

5) 行の併合

システム・コンフィギュレーション・ファイルでは、行末に“\” を記述した場合、次の行との併合を意味します。

ただし、“\” の直前 1 文字は、スペース、または、タブでなければなりません。

6) キー・ワード

CF850 では、次に示す文字列をキー・ワードとして予約しています。

したがって、これらの文字列をほかの用途に使用することは禁止されています。

auto	az	clkhdr	cyc	di	ei
flg	flg1	inthdr	intstk	maxpri	mbx
mpf	mpl	multi	no_use	no_wait	pool0
pool1	regmod	r22	r26	r32	RX850
rxsers	sem	ser_def	sit_def	TA_MFIFO	TA_MPRI

TCY_ULNK	TCY_OFF	TCY_ON	trace	tsk	tskgrp
TTS_DMT	TTS_RDY				

5.3 コンフィギュレーション情報

システム・コンフィギュレーション・ファイルに記述するコンフィギュレーション情報は、次の2種類に大別されます。

- [リアルタイム OS 情報](#)
使用するリアルタイム OS に関するデータ
- [SIT \(System Information Table\) 情報](#)
RX850 が動作するうえで必要となるデータ

5.3.1 リアルタイム OS 情報

システム・コンフィギュレーション・ファイルに記述するリアルタイム OS 情報は、次に示す1項目です。

- 1) [RX シリーズ情報](#)
RX シリーズ情報として、次のデータを定義します。
 - リアルタイム OS 名
 - バージョン番号

5.3.2 SIT (System Information Table) 情報

システム・コンフィギュレーション・ファイルに記述する SIT 情報は、次に示す2項目です。

- 1) [システム情報](#)
システム情報として、次のデータを定義します。
 - レジスタ・モード
 - システム・スタック情報
 - クロック割り込み要因
 - トレース情報
- 2) [システム最大値情報](#)
システム最大値情報として、次のデータを定義します。
 - タスクの優先度範囲
- 3) [タスク実行権グループ情報](#)
個々のタスク実行権グループに対して、次のデータを定義します。
 - タスク実行権グループ名
 - タスク実行権グループ用スタック情報
 - タスク実行権グループ待ち情報
- 4) [タスク情報](#)
個々のタスクに対して、次のデータを定義します。
 - タスク名
 - タスクの起動アドレス
 - タスク用スタック情報
 - タスクの初期優先度
 - タスクの初期状態
 - タスクの起動コード
 - 割り込み状態

- 5) セマフォ情報
個々のセマフォに対して、次のデータを定義します。
 - セマフォ名
 - セマフォの初期資源数
- 6) イベントフラグ情報
個々のイベントフラグに対して、次のデータを定義します。
 - イベントフラグ名
- 7) 1ビット・イベントフラグ情報
個々の1ビット・イベントフラグに対して、次のデータを定義します。
 - 1ビット・イベントフラグ名
- 8) メールボックス情報
個々のメールボックスに対して、次のデータを定義します。
 - メールボックス名
 - メッセージのキューイング方式
- 9) 間接起動割り込みハンドラ情報
個々の間接起動割り込みハンドラに対して、次のデータを定義します。
 - 割り込み要因
 - 間接起動割り込みハンドラの起動アドレス
- 10) 固定長メモリ・プール情報
個々の固定長メモリ・プールに対して、次のデータを定義します。
 - 固定長メモリ・プール名
 - メモリ・ブロック情報
 - メモリ・ブロックの総数
- 11) 可変長メモリ・プール情報
個々の可変長メモリ・プールに対して、次のデータを定義します。
 - 可変長メモリ・プール名
 - 可変長メモリ・プール情報
- 12) 周期起動ハンドラ情報
個々の周期起動ハンドラに対して、次のデータを定義します。
 - 周期起動ハンドラ名
 - 周期起動ハンドラの起動アドレス
 - 周期起動ハンドラの初期活性状態
 - 周期起動ハンドラの起動時間間隔

5.4 リアルタイム OS 情報の記述形式

ここでは、システム・コンフィギュレーション・ファイルに記述するリアルタイム OS 情報の記述形式を示します。
なお、表記中のゴシック書体は予約語であることを、イタリック書体はユーザが該当する数値、シンボル名、キー・ワードを記述する部分であることを表しています。

5.4.1 RX シリーズ情報

RX シリーズ情報では、使用するリアルタイム OS のリアルタイム OS 名、バージョン番号を定義します。
なお、システム・コンフィギュレーション・ファイルに、RX シリーズ情報を記述することは必須です。
以下に、RX シリーズ情報の記述形式を示します。

<i>rxsers</i>	<i>rtos_nam</i>	<i>rtos_ver</i>
---------------	-----------------	-----------------

次に、RX シリーズ情報で記述する各項目について示します。

- *rtos_nam*
リアルタイム OS の名前 (リアルタイム OS 名) を指定します。
なお、*rtos_nam* として指定可能なキー・ワードは、RX850 に限られます。
- *rtos_ver*
リアルタイム OS のバージョン番号を指定します。
なお、*rtos_ver* として指定可能なキー・ワードは、V32x (x は 0 ~ 9 の任意の数字) です。

5.5 SIT 情報の記述形式

ここでは、システム・コンフィギュレーション・ファイルに記述する SIT 情報の記述形式を示します。

なお、表記中のゴシック書体は予約語であることを、イタリック書体はユーザが該当する数値、シンボル名、キー・ワードを記述する部分であることを表しています。

5.5.1 システム情報

システム情報では、レジスタ・モード、システム・スタック情報、クロック割り込み要因、トレース情報を定義します。なお、システム・コンフィギュレーション・ファイルに、システム・スタック情報を記述することは必須です。

以下に、システム情報の記述形式を示します。

<code>regmod</code>	<i>reg_mode</i>
<code>intstk</code>	<i>intstk_siz</i> : <i>mem_nam</i>
<code>clkhdr</code>	<i>int_nam</i>
<code>trace</code>	<i>tracer</i>

次に、システム情報で記述する各項目について示します。

- *reg_mode*

システム構築時にリンクするニュークリアス・ライブラリ `librx.a` のレジスタ・モードを指定します。

なお、*reg_mode* として指定可能なキー・ワードは `r22`、`r26`、`r32` のいずれかに限られます。

<code>r22</code>	: 22 レジスタ・モード
<code>r26</code>	: 26 レジスタ・モード
<code>r32</code>	: 32 レジスタ・モード

注意 1 `regmod` の定義を省略した場合、“`regmod r32`” が記述されたもとして扱われます。

注意 2 CF850 の起動オプションとして `-regxx` が指定された場合、本項目の定義は無視され、CF850 の起動オプションが有効情報となります。

- *intstk_siz* : *mem_nam*

システム・スタックのサイズ (単位: バイト)、および、システム・スタックを割り付けるシステム・メモリの種類を指定します。

なお、*intstk_siz* として指定可能な値は `0x0` ~ `0xffffffc` の 4 バイト境界値に、*mem_nam* として指定可能なキー・ワードは `pool0`、または、`pool1` のいずれかに限られます。

<code>pool0</code>	: システム・メモリ・プール 0 (<code>pool0</code> セクション) にシステム・スタックを割り付けます。
<code>pool1</code>	: システム・メモリ・プール 1 (<code>pool1</code> セクション) にシステム・スタックを割り付けます。

- *int_nam*

システム・クロックとして使用するクロック割り込みの割り込み要因を指定します。

なお、*int_nam* として指定可能な値 / キー・ワードは使用する C コンパイラ・パッケージにより異なります。

【CA850 対応版】

デバイス・ファイルで規定されている割り込み要因名、“(例外コード - `0x80`) / `0x10`” で計算される値、また、`no_use` に限られます。

【GHS 製コンパイラ対応版】

“(例外コード - `0x80`) / `0x10`” で計算される値、または、`no_use` に限られます。

注意 1 `clkhdr` の定義を省略した場合、“`clkhdr no_use`” が記述されたものとして扱われます。

注意 2 `no_use` を指定した場合、RX850 が提供するタイマ・オペレーション機能 (周期起床、遅延起床、タイムアウト等) を使用しないことを意味します。

- *tracer*

トレース情報を使用するツールの種別を指定します。

なお、*tracer* として指定可能なキー・ワード、および、*trace* の定義を省略した場合の動作は、使用するクロス・ツールにより異なります。

【 CA850 対応版 】

指定可能なキー・ワードは *az* に限られます。

az : AZ850 が提供するトレース機能を使用します。

注意 *trace* の定義を省略した場合、“ *trace az* ” が記述されたものとして扱われます。

【 GHS 製コンパイラ対応版 】

指定可能なキー・ワードは *az*、または、*multi* のいずれかに限られます。

az : AZ850 が提供するトレース機能を使用します。

multi : MULTI が提供するタスク・デバッグ機能を使用します。

注意 *trace* の定義を省略した場合、“ *trace multi* ” が記述されたものとして扱われます。

5.5.2 システム最大値情報

システム最大値情報では、システム内で使用するタスクの優先度範囲を定義します。
以下に、システム最大値情報の記述形式を示します。

<code>maxpri</code> <code>pri_lvl</code>
--

次に、システム最大値情報で記述する各項目について示します。

- `pri_lvl`

タスクの優先度範囲を指定します。

なお、`pri_lvl` として指定可能な値 / キー・ワードは 0x1 ~ 0x1f, auto に限られます。

注意 1 `maxpri` の定義を省略した場合、“`maxpri auto`” が記述されたものとして扱われます。

注意 2 `auto` を指定した場合、CF850 が [タスク情報](#) で指定された初期優先度を参照し、該当する優先度範囲を出力します。

5.5.3 タスク実行権グループ情報

タスク実行権グループ情報では、タスク実行権グループ名、タスク実行権グループ用スタック情報、タスク実行権グループ待ち情報を個々のタスク実行権グループに対して定義します。

なお、タスク実行権グループ情報として定義可能な数は、0 ~ 127 個の範囲に限られます。

以下に、タスク実行権グループ情報の記述形式を示します。

<code>tskgrp</code>	<code>tskgrp_id</code>	<code>stk_siz : mem_nam</code>	<code>wait</code>
---------------------	------------------------	--------------------------------	-------------------

次に、タスク実行権グループ情報で記述する各項目について示します。

- `tskgrp_id`

タスク実行権グループ名を指定します。

なお、`tskgrp_id` として指定可能な値はシンボル名に限られます。

- `stk_siz : mem_nam`

タスク実行権グループ用スタックのサイズ (単位: バイト)、および、タスク実行権グループ用スタックを割り付けるシステム・メモリの種類を指定します。

なお、`stk_siz` として指定可能な値は 0x0 ~ 0xfffffc の 4 バイト境界値に、`mem_nam` として指定可能なキー・ワードは `pool0`、または、`pool1` のいずれかに限られます。

- `pool0` : システム・メモリ・プール 0 (`pool0` セクション) にタスク実行権グループ用スタックを割り付けます。
- `pool1` : システム・メモリ・プール 1 (`pool1` セクション) にタスク実行権グループ用スタックを割り付けます。

- `wait`

タスク実行権待ちの可能性の有り / 無しを指定します。

なお、`wait` として指定可能なキー・ワードは `no_wait` に限られます。

- `no_wait` : タスク起動時、“タスク実行権グループ待ち状態へと遷移する可能性は、無し” となります。

注意 `wait` の指定を省略した場合、タスク起動時、“タスク実行権グループへと遷移する可能性は、有り” となります。

5.5.4 タスク情報

タスク情報では、タスク名、タスクの起動アドレス、タスク用スタック情報、タスクの初期優先度、タスクの初期状態、起動コード、割り込み状態を個々のタスクに対して定義します。

なお、システム・コンフィギュレーション・ファイルにタスク情報を 1 個以上記述することは必須です。

また、タスク情報として定義可能な数は、1 ~ 127 個の範囲に限られます。

以下に、タスク情報の記述形式を示します。

tsk	<i>tsk_id</i>	<i>sta_adr</i>	<i>tskgrp_id</i> <i>stk_siz</i> : <i>mem_nam</i>	<i>pri</i>	
	<i>sts</i>	<i>sta_code</i>	<i>intr</i>		

次に、タスク情報で記述する各項目について示します。

- *tsk_id*

タスク名を指定します。

なお、*tsk_id* として指定可能な値は、シンボル名に限られます。

注意 CF850 により *tsk_id* と ID 番号の関係が、次に示す形式でシステム情報ヘッダ・ファイルに出力されます。

```
#define      tsk_id          タスクの ID 番号
```

- *sta_adr*

タスクの起動アドレスを指定します。

なお、*sta_adr* として指定可能な値は 0x0 ~ 0x3ffffe の 2 バイト境界値、シンボル名に限られます。

注意 *sta_adr* にシンボル名を指定する場合は、アセンブリ言語で記述する際のラベル名を指定します。

- *tskgrp_id*, *stk_siz* : *mem_nam*

タスク用スタックのサイズ (単位: バイト)、および、タスク用スタックを割り付けるシステム・メモリの種類を指定します。

なお、*tskgrp_id* として指定可能な値は[タスク実行権グループ情報](#)で指定されたタスク実行権グループ名に限られます。

また、*stk_siz* として指定可能な値は 0x0 ~ 0xfffffc の 4 バイト境界値に、*mem_nam* として指定可能なキー・ワードは pool0、または、pool1 のいずれかに限られます。

pool0 : システム・メモリ・プール 0 (.pool0 セクション) にタスク用スタックを割り付けます。

pool1 : システム・メモリ・プール 1 (.pool1 セクション) にタスク用スタックを割り付けます。

- *pri*

タスクの初期優先度を指定します。

なお、*pri* として指定可能な値は 0x1 ~ 0x1f に限られます。

- *sts*

タスクの初期状態を指定します。

なお、*sts* として指定可能なキー・ワードは TTS_DMT、または、TTS_RDY のいずれかに限られます。

TTS_DMT : システム起動時、dormant 状態へと遷移します。

TTS_RDY : システム起動時、rdady 状態へと遷移します。

- *sta_code*

タスクの起動コードを指定します。

なお、*sta_code* として指定可能な値は 0x0 ~ 0xfffffff、シンボル名に限られます。

注意 *sta_code* は、*sts* に TTS_RDY を指定した場合にのみ有効となり、TTS_DMT を指定した場合には無効となります。

- *intr*

タスクの割り込み状態を指定します。

なお、*intr* として指定可能なキー・ワードは、ei、または、di のいずれかに限られます。

ei : タスク起動時、“すべてのマスカブル割り込みを許可” となります。

di : タスク起動時、“すべてのマスカブル割り込みを禁止” となります。

注意 *intr* の指定を省略した場合、タスク起動時、“すべてのマスカブル割り込みを許可” となります。

5.5.5 セマフォ情報

セマフォ情報では、セマフォ名、セマフォの初期資源数を個々のセマフォに対して定義します。
なお、セマフォ情報として定義可能な数は、0 ~ 127 個の範囲に限られます。
以下に、セマフォ情報の記述形式を示します。

<code>sem</code>	<code>sem_id</code>	<code>init_cnt</code>
------------------	---------------------	-----------------------

次に、セマフォ情報で記述する各項目について示します。

- `sem_id`

セマフォ名を指定します。

なお、`sem_id` として指定可能な値は、シンボル名に限られます。

注意 CF850 により `sem_id` と ID 番号の関係が、次に示す形式でシステム情報ヘッダ・ファイルに出力されます。

```
#define            sem_id            セマフォの ID 番号
```

- `init_cnt`

セマフォの初期資源数を指定します。

なお、`init_cnt` として指定可能な値は 0x0 ~ 0x7f に限られます。

5.5.6 イベントフラグ情報

イベントフラグ情報では、イベントフラグ名を個々のイベントフラグに対して定義します。
なお、イベントフラグ情報として定義可能な数は、0 ~ 127 個の範囲に限られます。
以下に、イベントフラグ情報の記述形式を示します。

<code>flg</code>	<code>flg_id</code>
------------------	---------------------

次に、イベントフラグ情報で記述する各項目について示します。

- `flg_id`

イベントフラグ名を指定します。

なお、`flg_id` として指定可能な値は、シンボル名に限られます。

注意 CF850 により `flg_id` と ID 番号の関係が、次に示す形式でシステム情報ヘッダ・ファイルに出力されます。

```
#define            flg_id            イベントフラグの ID 番号
```

5.5.7 1 ビット・イベントフラグ情報

1 ビット・イベントフラグ情報では、1 ビット・イベントフラグ名を個々の 1 ビット・イベントフラグに対して定義します。
なお、1 ビット・イベントフラグ情報として定義可能な数は、0 ~ 127 個の範囲に限られます。
以下に、1 ビット・イベントフラグ情報の記述形式を示します。

<code>flg1</code>	<code>flg1_id</code>
-------------------	----------------------

次に、1 ビット・イベントフラグ情報で記述する各項目について示します。

- `flg1_id`

1 ビット・イベントフラグ名を指定します。

なお、`flg1_id` として指定可能な値は、シンボル名に限られます。

注意 CF850 により `flg1_id` と ID 番号の関係が、次に示す形式でシステム情報ヘッダ・ファイルに出力されます。

```
#define            flg1_id            1 ビット・イベントフラグの ID 番号
```

5.5.8 メールボックス情報

メールボックス情報では、メールボックス名、メッセージのキューイング方式を個々のメールボックスに対して定義します。

なお、メールボックス情報として定義可能な数は、0 ~ 127 個の範囲に限られます。

以下に、メールボックス情報の記述形式を示します。

<i>mbx</i>	<i>mbx_id</i>	<i>mwai_opt</i>
------------	---------------	-----------------

次に、メールボックス情報で記述する各項目について示します。

- *mbx_id*

メールボックス名を指定します。

なお、*mbx_id* として指定可能な値はシンボル名に限られます。

注意 CF850 により *mbx_id* と ID 番号の関係が、次に示す形式でシステム情報ヘッダ・ファイルに出力されます。

```
#define            mbx_id            メールボックスの ID 番号
```

- *mwai_opt*

メッセージのキューイング方式を指定します。

なお、*mwai_opt* として指定可能なキー・ワードは TA_MFIFO、または、TA_MPRI のいずれかに限られます。

TA_MFIFO : メッセージのキューイング方式は、メッセージの送信を行った順になります。

TA_MPRI : メッセージのキューイング方式は、メッセージの優先度順になります。

5.5.9 間接起動割り込みハンドラ情報

間接起動割り込みハンドラ情報では、割り込み要因、間接起動割り込みハンドラの起動アドレスを個々の間接起動割り込みハンドラに対して定義します。

なお、間接起動割り込みハンドラ情報として定義可能な数は、各割り込み要因に対して 1 個に限られます。

以下に、間接起動割り込みハンドラ情報の記述形式を示します。

<i>inthdr</i>	<i>int_nam</i>	<i>hdr_adr</i>
---------------	----------------	----------------

次に、間接起動割り込みハンドラ情報で記述する各項目について示します。

- *int_nam*

割り込み要因を指定します。

なお、*int_nam* として指定可能な値 / キー・ワードは、使用する C コンパイラ・パッケージにより異なります。

【 CA850 対応版 】

デバイス・ファイルで規定されている割り込み要因名、または、“(例外コード - 0x80) / 0x10” で計算される値に限られます。

【 GHS 製コンパイラ対応版 】

“(例外コード - 0x80) / 0x10” で計算される値に限られます。

- *hdr_adr*

間接起動割り込みハンドラの起動アドレスを指定します。

なお、*hdr_adr* として指定可能な値は 0x0 ~ 0x3ffffe の 2 バイト境界値、シンボル名に限られます。

注意 *hdr_adr* にシンボル名を指定する場合は、アセンブリ言語で記述する際のラベル名を指定します。

5.5.10 固定長メモリ・プール情報

固定長メモリ・プール情報では、メモリ・プール名、メモリ・ブロック情報、メモリ・ブロックの総数を個々の固定長メモリ・プールに対して定義します。

なお、固定長メモリ・プール情報として定義可能な数は、0 ~ 127 個の範囲に限られます。

以下に、固定長メモリ・プール情報の記述形式を示します。

<i>mpf</i>	<i>mpf_id</i>	<i>blk_siz : mem_nam</i>	<i>blk_cnt</i>
------------	---------------	--------------------------	----------------

次に、固定長メモリ・プール情報で記述する各項目について示します。

- *mpf_id*

固定長メモリ・プール名を指定します。

なお、*mpf_id* として指定可能な値はシンボル名に限られます。

注意 CF850 により *mpf_id* と ID 番号の関係が、次に示す形式でシステム情報ヘッダ・ファイルに出力されます。

```
#define      mpf_id      固定長メモリ・プールの ID 番号
```

- *blk_siz : mem_nam*

1 ブロック当たりのメモリ・ブロックのサイズ (基本ブロック・サイズ, 単位: バイト), および, 固定長メモリ・プールを割り付けるシステム・メモリの種類を指定します。

なお、*blk_siz* として指定可能な値は 0x4 ~ 0xffffffffc の 4 バイト境界値に、*mem_nam* として指定可能なキー・ワードは、pool0、または、pool1 のいずれかに限られます。

```
pool0      : システム・メモリ・プール 0 (pool0 セクション) に固定長メモリ・プールを割り付けます。
pool1      : システム・メモリ・プール 1 (pool1 セクション) に固定長メモリ・プールを割り付けます。
```

- *blk_cnt*

メモリ・ブロックの総数を指定します。

なお、*blk_cnt* として指定可能な値は 0x1 ~ 0xffffffff に限られます。

5.5.11 可変長メモリ・プール情報

可変長メモリ・プール情報では、可変長メモリ・プール名、メモリ・プール情報を個々の可変長メモリ・プールに対して定義します。

なお、可変長メモリ・プール情報として定義可能な数は、0 ~ 127 個の範囲に限られます。

以下に、可変長メモリ・プール情報の記述形式を示します。

<code>mpl</code>	<code>mpl_id</code>	<code>mpl_siz : mem_nam</code>
------------------	---------------------	--------------------------------

次に、可変長メモリ・プール情報で記述する各項目について示します。

- `mpl_id`

可変長メモリ・プール名を指定します。

なお、`mpl_id` として指定可能な値は、シンボル名に限られます。

注意 CF850 により、`mpl_id` と ID 番号の関係が次に示す形式でシステム情報ヘッダ・ファイルに出力されます。

```
#define      mpl_id      可変長メモリ・プールの ID 番号
```

- `mpl_siz : mem_nam`

可変長メモリ・プールのサイズ (単位: バイト)、および、可変長メモリ・プールを割り付けるシステム・メモリの種類を指定します。

なお、`mpl_siz` として指定可能な値は 0x4 ~ 0xffffffff の 4 バイト境界値に、`mem_nam` として指定可能なキー・ワードは `pool0`、または、`pool1` のいずれかに限られます。

```
pool0      : システム・メモリ・プール 0 (.pool0 セクション) に可変長メモリ・プールを割り付けます。
pool1      : システム・メモリ・プール 1 (.pool1 セクション) に可変長メモリ・プールを割り付けます。
```

5.5.12 周期起動ハンドラ情報

周期起動ハンドラ情報では、周期起動ハンドラ名、周期起動の起動アドレス、周期起動ハンドラの初期活性状態、周期起動ハンドラの起動時間間隔を個々の周期起動ハンドラに対して定義します。

なお、周期起動ハンドラ情報として定義可能な数は、0 ~ 127 個の範囲に限られます。

以下に、周期起動ハンドラ情報の記述形式を示します。

<i>cyc</i>	<i>cyc_no</i>	<i>hdr_adr</i>	<i>act</i>	<i>intvl</i>
------------	---------------	----------------	------------	--------------

次に、周期起動ハンドラ情報で記述する各項目について示します。

- *cyc_no*

周期起動ハンドラ名を指定します。

なお、*cyc_no* として指定可能な値は、シンボル名に限られます。

注意 CF850 により、*cyc_no* と指定番号の関係が次に示す形式で、システム情報ヘッダ・ファイルに出力されます。

```
#define          cyc_no          周期起動ハンドラの指定番号
```

- *hdr_adr*

周期起動ハンドラの起動アドレスを指定します。

なお、*hdr_adr* として指定可能な値は 0x0 ~ 0x3ffffe の 2 バイト境界値、シンボル名に限られます。

注意 *hdr_adr* にシンボル名を指定する場合は、アセンブリ言語で記述する際のラベル名を指定します。

- *act*

周期起動ハンドラの初期活性状態を指定します。

なお、*act* として指定可能なキー・ワードは TCY_ON、TCY_OFF、または、TCY_ULNK のいずれかに限られます。

TCY_ON : システム起動時、活性状態は ON 状態になります。

TCY_OFF : システム起動時、活性状態は OFF 状態になります。

TCY_ULNK : システム起動時、タイマ・キューから外れた状態になります。

- *intvl*

周期起動ハンドラの起動時間間隔 (単位: 基本クロック周期) を指定します。

なお、*intvl* として指定可能な値は 0x1 ~ 0xfffffff に限られます。

5.6 記述上の注意点

システム・コンフィギュレーション・ファイルにコンフィギュレーション情報（リアルタイム OS 情報，SIT 情報）を記述する場合，次の順番で行います。

- 1) リアルタイム OS 情報の記述が開始されることの宣言
- 2) リアルタイム OS 情報の記述
- 3) SIT 情報の記述が開始されることの宣言
- 4) SIT 情報の記述

図 5-1 に，システム・コンフィギュレーション・ファイルの記述イメージを示します。

図 5-1 システム・コンフィギュレーション・ファイルの記述イメージ

```
-----
-- リアルタイム OS 情報の記述が開始されることの宣言
-----

ser_def

-----
-- リアルタイム OS 情報の記述
-----
.....
.....
.....

-----
-- SIT (System Information Table) 情報の記述が開始されることの宣言
-----

sit_def

-----
-- SIT (System Information Table) 情報の記述
-----
.....
.....
.....
```

5.7 記述例

図 5-2, 図 5-3 に, V851 使用時のシステム・コンフィギュレーション・ファイルの記述例を示します。

図 5-2 に CA850 対応版を使用した場合の記述例を, 図 5-3 に GHS 製コンパイラ対応版を使用した場合の記述例を示します。

なお, 図 5-2, 図 5-3 の記述例では, 次に示すデータが記述されています。

- RX シリーズ情報

リアルタイム OS 名 : RX850
バージョン番号 : V320

- システム情報

レジスタ・モード : 32 レジスタ・モード
システム・スタック情報 : 0x100 バイトをシステム・メモリ・プール 1 から確保
クロック割り込み要因 : 例外コード
CA850 対応版の場合 : ...INTCM4
GHS 製コンパイラ対応版の場合 : ...0x5
トレース情報 : ツールの種別
CA850 対応版の場合 : ...az
GHS 製コンパイラ対応版の場合 : ...multi

- システム最大値情報

タスクの優先度範囲 : 0x1f

- タスク実行権グループ情報

タスク実行権グループ名 : txc0
スタック情報 : 0x100 バイトをシステム・メモリ・プール 0 から確保
待ち情報 : タスク実行権グループ待ち状態へと遷移する可能性は, 有り
タスク実行権グループ名 : txc1
スタック情報 : 0x200 バイトをシステム・メモリ・プール 1 から確保
待ち情報 : タスク実行権グループ待ち状態へと遷移する可能性は, 無し

- タスク情報

タスク名 : task1
起動アドレス : ラベル名 _task1
スタック情報 : 0x100 バイトをシステム・メモリ・プール 1 から確保
起動時優先度 : 0x1
初期状態 : ready 状態
起動コード : 0xa
割り込み状態 : すべての割り込みを禁止

タスク名 : task2
起動アドレス : ラベル名 _task2
スタック情報 : タスク実行権グループ txc0 を使用
起動時優先度 : 0x2
初期状態 : ready 状態
起動コード : 0x14
割り込み状態 : すべての割り込みを許可

タスク名 : task3
起動アドレス : ラベル名 _task3
スタック情報 : タスク実行権グループ txb1 を使用
起動時優先度 : 0x3
初期状態 : dormant 状態
起動コード : 0x1e
割り込み状態 : すべての割り込みを許可

- セマフォ情報

セマフォ名 : sem1
初期資源数 : 0x0

セマフォ名	: sem2
初期資源数	: 0x0
セマフォ名	: sem3
初期資源数	: 0x1
セマフォ名	: sem4
初期資源数	: 0x7f
- イベントフラグ情報	
イベントフラグ名	: evf32_1
- 1 ビット・イベントフラグ情報	
1 ビット・イベントフラグ名	: evf1_1
- メールボックス情報	
メールボックス名	: mbx1
キューイング方式	: FIFO 順
メールボックス名	: mbx2
キューイング方式	: FIFO 順
メールボックス名	: mbx3
キューイング方式	: FIFO 順
メールボックス名	: mbx4
キューイング方式	: FIFO 順
メールボックス名	: pmbx1
キューイング方式	: 優先度順
メールボックス名	: pmbx2
キューイング方式	: 優先度順
メールボックス名	: pmbx3
キューイング方式	: 優先度順
メールボックス名	: pmbx4
キューイング方式	: 優先度順
- 間接起動割り込みハンドラ情報	
割り込み要因	: 例外コード CA850 対応版の場合 ...INTP00 GHS 製コンパイラ対応版の場合 ...0xa
起動アドレス	: ラベル名 _inthdr
- 固定長メモリ・プール情報	
メモリ・プール名	: mpf1
メモリ・ブロック情報	: 基本ブロック・サイズ 0x8 バイトのメモリ・ブロックをシステム・メモリ・プール 1 から確保
メモリ・ブロックの総数	: 0xa
- 可変長メモリ・プール情報	
メモリ・プール名	: mpl1
メモリ・プール・サイズ情報	: 0x20 バイトをシステム・メモリ・プール 1 から確保
- 周期起動ハンドラ情報	
周期起動ハンドラ名	: cychdr1
起動アドレス	: ラベル名 _cyhdr1
初期活性状態	: OFF 状態

起動時間間隔	: 0x1 (単位: クロック割り込み周期)
周期起動ハンドラ名	: cyhdr2
起動アドレス	: ラベル名 _cyhdr2
初期活性状態	: ON 状態
起動時間間隔	: 0x2 (単位: クロック割り込み周期)
周期起動ハンドラ名	: cyhdr3
起動アドレス	: ラベル名 _cyhdr3
初期活性状態	: タイマ・キューから外す
起動時間間隔	: 0x3 (単位: クロック割り込み周期)

図 5-2 システム・コンフィギュレーション・ファイルの記述例 (CA850 対応版)

```

-----
-- リアルタイム OS 情報の記述が開始されることの宣言
-----

ser_def

-----

-- リアルタイム OS 情報の記述
-----

-- RX シリーズ情報
rxsers          RX850          V320

-----

-- SIT (System Information Table) 情報の記述が開始されることの宣言
-----

sit_def

-----

-- SIT (System Information Table) 情報の記述
-----

-- システム情報
regmod          r32
intstk          0x100 : pool1
clkhdr          INTCM4
trace          az

-- システム最大値情報
maxpri          0x1f

-- タスク実行権グループ情報
tskgrp          txc0           0x100 : pool0
tskgrp          txc1           0x200 : pool1   no_wait

-- タスク情報
tsk             task1          _task1         0x100 : pool1   0x1          TTS_RDY      \
               0xa            di
tsk             task2          _task2         txc0           0x2          TTS_RDY      \
               0x14           ei
tsk             task3          _task3         txc1           0x3          TTS_DMT      \
               0x1            ei

-- セマフォ情報
sem             sem1           0x0
sem             sem2           0x0
sem             sem3           0x1
sem             sem4           0x7f

-- イベントフラグ情報
flg             exv32_1

```

-- 1 ビット・イベントフラグ情報

flg1	exv1_1
------	--------

-- メールボックス情報

mbx	mbx1	TA_MFIFO
mbx	mbx2	TA_MFIFO
mbx	mbx3	TA_MFIFO
mbx	mbx4	TA_MFIFO
mbx	pmbx1	TA_MPRI
mbx	pmbx2	TA_MPRI
mbx	pmbx3	TA_MPRI
mbx	pmbx4	TA_MPRI

-- 間接起動割り込みハンドラ情報

inthdr	INTP00	_inthdr
--------	--------	---------

-- 固定長メモリ・プール情報

mpf	mpf1	0x8 : pool1	0xa
-----	------	-------------	-----

-- 可変長メモリ・プール情報

mpl	mpl1	0x20 : pool1
-----	------	--------------

-- 周期起動ハンドラ情報

cyc	cychdr1	_cychdr1	TCY_OFF	0x1
cyc	cychdr2	_cychdr2	TCY_ON	0x2
cyc	cychdr3	_cychdr3	TCY_ULNK	0x3

図 5-3 システム・コンフィギュレーション・ファイルの記述例 (GHS 製コンパイラ対応版)

```

-----
-- リアルタイム OS 情報の記述が開始されることの宣言
-----
ser_def

-----
-- リアルタイム OS 情報の記述
-----

-- RX シリーズ情報
rxsers          RX850          V320

-----
-- SIT (System Information Table) 情報の記述が開始されることの宣言
-----
sit_def

-----
-- SIT (System Information Table) 情報の記述
-----

-- システム情報
regmod          r32
intstk          0x100 : pool1
clkhdr          0x5
trace           multi

-- システム最大値情報
maxpri          0x1f

-- タスク実行権グループ情報
tskgrp          txc0           0x100 : pool0
tskgrp          txc1           0x200 : pool1    no_wait

-- タスク情報
tsk             task1          _task1          0x100 : pool1    0x1          TTS_RDY      \
               0xa            di
tsk             task2          _task2          txc0           0x2          TTS_RDY      \
               0x14           ei
tsk             task3          _task3          txc1           0x3          TTS_DMT      \
               0x1            ei

-- セマフォ情報
sem             sem1           0x0
sem             sem2           0x0
sem             sem3           0x1
sem             sem4           0x7f

-- イベントフラグ情報
flg             exv32_1

```

-- 1 ビット・イベントフラグ情報

flg1	exv1_1
------	--------

-- メールボックス情報

mbx	mbx1	TA_MFIFO
mbx	mbx2	TA_MFIFO
mbx	mbx3	TA_MFIFO
mbx	mbx4	TA_MFIFO
mbx	pmbx1	TA_MPRI
mbx	pmbx2	TA_MPRI
mbx	pmbx3	TA_MPRI
mbx	pmbx4	TA_MPRI

-- 間接起動割り込みハンドラ情報

inthdr	0xa	_inthdr
--------	-----	---------

-- 固定長メモリ・プール情報

mpf	mpf1	0x8 : pool1	0xa
-----	------	-------------	-----

-- 可変長メモリ・プール情報

mpl	mpl1	0x20 : pool1
-----	------	--------------

-- 周期起動ハンドラ情報

cyc	cychdr1	_cychdr1	TCY_OFF	0x1
cyc	cychdr2	_cychdr2	TCY_ON	0x2
cyc	cychdr3	_cychdr3	TCY_ULNK	0x3

第 6 章 コンフィギュレータの操作方法

この章では、コンフィギュレータ CF850 を操作して、システム・コンフィギュレーション・ファイルから情報ファイル (システム情報テーブル・ファイル、システム情報ヘッダ・ファイル) を生成する方法について解説しています。

6.1 概要

RX850 が提供している機能を利用したシステム (ロード・モジュール) を構築する場合、RX850 に提供するデータを保持した情報が必要となります。

基本的に情報ファイルは、規定された形式のデータ羅列であるため、各種エディタを用いて記述することは可能です。しかし、情報ファイルは、記述性 / 可読性の面で劣ったものとなっているため、記述に際してはかなりの時間と労力を必要としています。

そこで、RX850 では、記述性 / 可読性の面で優れたシステム・コンフィギュレーション・ファイルから情報ファイルへと変換するユーティリティ・ツール (コンフィギュレータ CF850) を提供しています。

CF850 は、システム・コンフィギュレーション・ファイルを入力ファイルとして読み込んだあと、情報ファイルを出します。

以下に、CF850 が出力する情報ファイルについて示します。

- システム情報テーブル・ファイル
RX850 が動作するうえで必要なるデータ (タスク、セマフォ、イベントフラグ等の RX850 の資源情報) を保持した情報ファイルです。
- システム情報ヘッダ・ファイル
システム・コンフィギュレーション・ファイルに記述されたオブジェクト名 (タスク名、セマフォ名、イベントフラグ名など) と ID 番号の対応付けを保持した情報ファイルです。

表 6-1 に、CF850 の動作環境を示します。

表 6-1 CF850 の動作環境

ホスト・マシン	オペレーティング・システム
Windows ベース - IBM PC/AT 互換機	Windows 98, Me, NT4.0, 2000, XP

6.2 起動方法

6.2.1 コマンド・ラインからの起動

以下に、CF850 をコマンド・ラインから起動する際の起動方法を示します。
 ただし、入力例中の “C>” はコマンド・プロンプトを、“ ” はスペース・キーの入力を表しています。
 また、“[]” で囲まれた起動オプションは、省略可能な起動オプションであることを表しています。

【 CA850 対応版の場合 】

```
C> cf850 [ @cmd_file ] [ -cpu name ] [ -devpath=path ] [ -regxx ] [ -i sit_file ] [ -ni ] [ -d
h_file ] [ -nd ] [ -V ] [ -help ] cf_file
```

【 GHS 製コンパイラ対応版の場合 】

```
C> cf850 [ @cmd_file ] [ -regxx ] [ -i sit_file ] [ -ni ] [ -d h_file ] [ -nd ] [ -V ] [ -help ] cf_file
```

以下に、各起動オプションの詳細を示します。

- @cmd_file

コマンド・ファイル名を指定します。

省略時 コマンド・ライン上で指定された起動オプションが有効となります。

注意 コマンド・ファイルについての詳細は、「[6.2.3 コマンド・ファイル](#)」を参照してください。

- -cpu name

ターゲット・デバイスの品種指定名を指定します。

省略時 CF850 はデバイス・ファイルの読み込みを行いません。このため、システム・コンフィギュレーション・ファイル内で “デバイス・ファイルで規定されている割り込み要因名” を用いた定義が行えなくなります。

注意 本起動オプションは、CA850 対応版でのみ指定可能となります。

- -devpath=path

-cpu name で指定されたターゲット・デバイスに対応したデバイス・ファイルを path フォルダから検索します。

省略時 カレント・フォルダ、..\..\dev の順序で検索処理を行います。

注意 本起動オプションは、CA850 対応版でのみ指定可能となります。

- -regxx

出力ファイルのファイル形式 (レジスタ・モード) を指定します。

なお、xx として指定可能なキー・ワードは 22, 26, 32 のいずれかに限られます。

22	: 22 レジスタ・モード
26	: 26 レジスタ・モード
32	: 32 レジスタ・モード

省略時 [システム情報](#)において指定されたレジスタ・モードとなります。

なお、本起動オプション、および、[システム情報](#)の両方においてレジスタ・モードの指定が行われなかった場合、CF850 は -reg32 が指定されていたものとして処理を行います。

- -i sit_file

出力するシステム情報テーブル・ファイル名を指定します。

省略時 以下に示した起動オプションが指定されていたものとして処理を行います。

CA850 対応版	: -i sit.s
GHS 製コンパイラ対応版	: -i sit.850

注意 1 出力ファイル名 sit_file として指定可能な文字数は、パス名を含めて 255 文字以内に限られます。

注意 2 本起動オプションと `-ni` を同時に指定した場合、前者に入力した起動オプションは無効となり、後者に入力した起動オプションが有効となります。

- `-ni`

システム情報テーブル・ファイルを出力しません。

省略時 `cf_file` で指定されたシステム・コンフィギュレーション・ファイル名の末尾に“`i`”を付加し、拡張子を“`.tbl`”に置き換えたシステム情報テーブル・ファイルを出力します。

注意 本起動オプションと `-i sit_file` を同時に指定した場合、前者に入力した起動オプションは無効となり、後者に入力した起動オプションが有効となります。

- `-d h_file`

出力するシステム情報ヘッダ・ファイル名を指定します。

省略時 `cf_file` で指定されたシステム・コンフィギュレーション・ファイル名の拡張子を“`.h`”に置き換えたシステム情報ヘッダ・ファイルを出力します。

注意 1 出力ファイル名 `h_file` として指定可能な文字数は、パス名を含めて 255 文字以内に限られます。

注意 2 本起動オプションと `-nd` を同時に指定した場合、前者に入力した起動オプションは無効となり、後者に入力した起動オプションが有効となります。

- `-nd`

システム情報ヘッダ・ファイルを出力しません。

省略時 `cf_file` で指定されたシステム・コンフィギュレーション・ファイル名の拡張子を“`.h`”に置き換えたシステム情報ヘッダ・ファイルを出力します。

注意 本起動オプションと `-d h_file` を同時に指定した場合、前者に入力した起動オプションは無効となり、後者に入力した起動オプションが有効となります。

- `-V`

CF850 のバージョン情報を標準出力に出力します。

省略時 CF850 のバージョン情報を出力しません。

注意 本起動オプションを指定した場合、ほかの起動オプションはすべて無効となります。

- `-help`

CF850 の起動オプションの使い方を標準出力に出力します。

省略時 CF850 の起動オプションの使い方を出力しません。

注意 本起動オプションを指定した場合、ほかの起動オプションはすべて無効となります。

- `cf_file`

CF850 への入力ファイルとなる、システム・コンフィギュレーション・ファイル名を指定します。

省略時 本起動オプションを省略することはできません。

注意 入力ファイル名 `cf_file` として指定可能な文字数は、パス名を含めて 255 文字以内に限られます。

6.2.2 PM+ からの起動

以下に、CF850の起動オプションをCA850が提供する統合開発環境プラットフォームPM+から設定する方法を示します。なお、本項では、既存のプロジェクト・ファイルに対する設定方法を例にとり、説明を行っています。

1) PM+ の起動

ショート・カット (デフォルト: [スタート] → [プログラム (P)] → [NEC Electronics Tools] → [PM+] → [Vx.xx] → [PM+ Vx.xx]) をクリック、または、実行形式ファイル (デフォルト: C:\Program Files\NEC Electronics Tools\PM+\Vx.xx\bin\PMplus.exe) をダブル・クリックし、PM+ を起動します。

2) [ワークスペースを開く] ダイアログのオープン

[ファイル(E)] → [ワークスペースを開く(W)...] を選択し、[ワークスペースを開く] ダイアログをオープンします。

注意 [ワークスペースを開く] ダイアログについての詳細は、「PM+ ユーザーズ・マニュアル」を参照してください。

3) ワークスペース・ファイルの指定

[ファイルの場所(I)] エリア、[ファイル名(N)] エリア、および、[ファイルの種類(I)] エリアをそれぞれに選択 / 指定したあと、<開く(Q)> ボタンをクリックし、CF850 の起動オプションを設定するワークスペース・ファイル(、または、プロジェクト・ファイル) を指定します。

4) [OSの選択] ダイアログのオープン

[ツール(I)] → [OSの選択(Q)...] を選択し、[OSの選択] ダイアログをオープンします。

注意 [OSの選択] ダイアログについての詳細は、「付録A ウィンドウ・リファレンス」を参照してください。

5) [RX850 詳細設定] ダイアログのオープン

[OSの選択(Q)] エリアのリスト・ボックスから “RX850 V3.xx” を選択したあと、<OK> ボタンをクリックし、[RX850 詳細設定] ダイアログをオープンします。

注意 [RX850 詳細設定] ダイアログについての詳細は、「付録A ウィンドウ・リファレンス」を参照してください。

6) システム・コンフィギュレーション・ファイルの指定

[システム・コンフィギュレーション・ファイル(C)] エリアにおいて、入力ファイル名(システム・コンフィギュレーション・ファイル名) を指定します。

7) システム情報テーブル・ファイルの指定

[システム情報テーブル・ファイルを出力する[-i/-n](I)] チェック・ボックスをチェック状態としたあと、[ファイル名(E)] エリアにおいて、出力ファイル名(システム情報テーブル・ファイル名) を指定します。

8) システム情報ヘッダ・ファイルの指定

[システム情報ヘッダ・ファイルを出力する[-d/-nd](H)] チェック・ボックスをチェック状態としたあと、[ファイル名(A)] エリアにおいて、出力ファイル名(システム情報ヘッダ・ファイル名) を指定します。

9) 起動オプションの確認

6) ~ 8) の指定により設定されるCF850の起動オプション指定形式が[コマンドライン・オプション] エリアに表示されます。これにより、6) ~ 8) の指定結果が意図したものと一致しているか否かを明示的に確認することができます。

10) プロジェクト・ファイルに対する反映

<OK> ボタンをクリックし、上記操作結果をプロジェクト・ファイルに対して反映します。

6.2.3 コマンド・ファイル

CF850 では、コマンド・ライン上で指定可能な起動オプションの文字数制限を解消する目的からコマンド・ファイル対応を行っています。

以下に、コマンド・ファイルの記述形式を示します。

1) コメント行

行頭に # が記述された行については、コメント行として扱われます。

2) 起動オプション

起動オプションと起動オプションは、スペース・コード、または、改行コードで区切って記述します。

なお、-cpu, -i, -d といった -xxx 部とパラメータ部から構成される起動オプションについては、-xxx 部とパラメータ部をスペース・コード、または、改行コードで区切って記述します。

注意 -devpath のパラメータ部 (パス名 *path*) にスペース・コードを含むフォルダ名が存在する場合は、パラメータ部をダブル・クォーテーション " でくる必要があります。

-devpath="Program Files\DEV"

3) 文字数制限

コマンド・ファイル内の 1 行に対する文字数制限は 4096 文字となっています。

図 6-1 に、CA850 対応版におけるコマンド・ファイルの記述例を示します。

なお、図 6-1 の記述例では、次に示す起動オプションが記述されています。

ターゲット・デバイス	: μ PD703000
デバイス・ファイルの検索フォルダ	: C:\NECTools32\DEV
レジスタ・モード	: r26
システム情報テーブル・ファイル	: sitfile.tbl
システム情報ヘッダ・ファイル	: hfile.h
システム・コンフィギュレーション・ファイル	: cffile.cf

図 6-1 コマンド・ファイルの記述例 (CA850 対応版)

```
# Command File
-cpu
3000
-devpath=C:\NECTools32\DEV
-reg26
-i
sitfile.tbl
-d
hfile.h
cffile.cf
```

6.3 コマンド入力例

6.3.1 CA850 対応版の場合

次に、CA850 に対応した cf850 のコマンド入力例を示します。

なお、この例では、ターゲット・デバイスとして μ PD703000 を指定しています。

- cf850 -cpu 3000 -devpath=C:\NECTools32\DEV -reg26 -i sitfile.tbl -d hfile.h cffile.cf
システム・コンフィギュレーション・ファイル cffile.cf をカレント・フォルダから、品種指定名 3000 に対応したデバイス・ファイルを C:\NECTools32\DEV フォルダから入力ファイルとして読み込んだあと、システム情報テーブル・ファイル sitfile.tbl、および、システム情報ヘッダ・ファイル hfile.h を 26 レジスタ・モードのファイル形式で出力します。
- cf850 -cpu 3000 -devpath=C:\NECTools32\DEV -i sitfile.tbl cffile.cf
システム・コンフィギュレーション・ファイル cffile.cf をカレント・フォルダから、品種指定名 3000 に対応したデバイス・ファイルを C:\NECTools32\DEV フォルダから入力ファイルとして読み込んだあと、システム情報テーブル・ファイル sitfile.tbl、および、システム情報ヘッダ・ファイル cffile.h を 32 レジスタ・モードのファイル形式で出力します。
- cf850 -cpu 3000 -devpath=C:\NECTools32\DEV -d hfile.h cffile.cf
システム・コンフィギュレーション・ファイル cffile.cf をカレント・フォルダから、品種指定名 3000 に対応したデバイス・ファイルを C:\NECTools32\DEV フォルダから入力ファイルとして読み込んだあと、システム情報テーブル・ファイル sit.s、および、システム情報ヘッダ・ファイル hfile.h を 32 レジスタ・モードのファイル形式で出力します。
- cf850 -cpu 3000 -devpath=C:\NECTools32\DEV cffile.cf
システム・コンフィギュレーション・ファイル cffile.cf をカレント・フォルダから、品種指定名 3000 に対応したデバイス・ファイルを C:\NECTools32\DEV フォルダから入力ファイルとして読み込んだあと、システム情報テーブル・ファイル sit.s、および、システム情報ヘッダ・ファイル cffile.h を 32 レジスタ・モードのファイル形式で出力します。
- cf850 -cpu 3000 -nd cffile.cf
システム・コンフィギュレーション・ファイル cffile.cf、および、品種指定名 3000 に対応したデバイス・ファイルをカレント・フォルダ、または、..\..\dev フォルダから入力ファイルとして読み込んだあと、システム情報テーブル・ファイル sit.s を 32 レジスタ・モードのファイル形式で出力します。
- cf850 -cpu 3000 -ni cffile.cf
システム・コンフィギュレーション・ファイル cffile.cf、および、品種指定名 3000 に対応したデバイス・ファイルをカレント・フォルダか、または、..\..\dev フォルダから入力ファイルとして読み込んだあと、システム情報ヘッダ・ファイル cffile.h を 32 レジスタ・モードのファイル形式で出力します。
- cf850 -V
CF850 のバージョン情報を標準出力に出力します。
- cf850 -help
CF850 の起動オプションの使い方を標準出力に出力します。

6.3.2 GHS 製コンパイラ対応版の場合

次に、GHS 製コンパイラに対応した CF850 のコマンド入力例を示します。

- cf850 -reg26 -i sitfile.tbl -d hfile.h cffile.cf
システム・コンフィギュレーション・ファイル cffile.cf をカレント・フォルダから入力ファイルとして読み込んだあと、システム情報テーブル・ファイル sitfile.tbl、および、システム情報ヘッダ・ファイル hfile.h を 26 レジスタ・モードのファイル形式で出力します。
- cf850 -i sitfile.tbl cffile.cf
システム・コンフィギュレーション・ファイル cffile.cf をカレント・フォルダから入力ファイルとして読み込んだあと、システム情報テーブル・ファイル sitfile.tbl、および、システム情報ヘッダ・ファイル cffile.h を 32 レジスタ・モードのファイル形式で出力します。
- cf850 -d hfile.h cffile.cf
システム・コンフィギュレーション・ファイル cffile.cf をカレント・フォルダから入力ファイルとして読み込んだあと、システム情報テーブル・ファイル sit.850、および、システム情報ヘッダ・ファイル hfile.h を 32 レジスタ・モードのファイル形式で出力します。
- cf850 cffile.cf
システム・コンフィギュレーション・ファイル cffile.cf をカレント・フォルダから入力ファイルとして読み込んだあと、システム情報テーブル・ファイル sit.850、および、システム情報ヘッダ・ファイル cffile.h を 32 レジスタ・モードのファイル形式で出力します。
- cf850 -nd cffile.cf
システム・コンフィギュレーション・ファイル cffile.cf をカレント・フォルダから入力ファイルとして読み込んだあと、システム情報テーブル・ファイル sit.850 を 32 レジスタ・モードのファイル形式で出力します。
- cf850 -ni cffile.cf
システム・コンフィギュレーション・ファイル cffile.cf をカレント・フォルダから入力ファイルとして読み込んだあと、システム情報ヘッダ・ファイル cffile.h を 32 レジスタ・モードのファイル形式で出力します。
- cf850 -V
CF850 のバージョン情報を標準出力に出力します。
- cf850 -help
CF850 の起動オプションの使い方を標準出力に出力します。

6.4.1 致命的エラー

次に、致命的エラーが発生した際に出力されるメッセージを示します。

なお、メッセージ内のイタリック書体 (*file_name* 等) は、致命的エラーが発生した際に決定されるものです。

F2001: Usage : cf850 [@command_file] [-cpu <name>] [devpath=<path>] [-reg22] [-reg26] [-reg32] [-i <SIT file>] [-d <include file>] [-ni] [-nd] [-V] [-help] <file>

CA850 に対応した CF850 の起動オプションの指定が不正です。

F2001: Usage : cf850 [@command_file] [-reg22] [-reg26] [-reg32] [-i <SIT file>] [-d <include file>] [-ni] [-nd] [-V] [-help] <file>

GHS 製コンパイラに対応した CF850 の起動オプションの指定が不正です。

F2002: Can't allocate memory.

メモリが足りません。

F2003: Can't open file *file_name*.

ファイル *file_name* がオープンできません。

F2004: Out of memory.

メモリが足りません。

F2005: Can't Open device file.

デバイス・ファイルがオープンできません。

F2006: Can't ready device file.

デバイス・ファイルが読み込めません。

F2007: Unknown device file format

未対応のデバイス・ファイルが指定されています。

F2008: Output file "*file_name*" names are the same.

出力ファイル(システム情報テーブル・ファイル、システム情報ヘッダ・ファイル)に同じファイル名が指定されています

6.4.2 致命的でないエラー

次に、致命的でないエラーが発生した際に出力されるメッセージを示します。
なお、メッセージ内のイタリック書体 (*name* 等) は、致命的でないエラーが発生した際に決定されるものです。

E2001: `ser_def` not defined.

[リアルタイム OS 情報](#)の開始宣言 `ser_def` が先頭行で行われていません。

E2002: `Illegal ser_def`.

[リアルタイム OS 情報](#)の開始宣言 `ser_def` が誤った場所で行われています。

E2003: `ser_def` already defined.

[リアルタイム OS 情報](#)の開始宣言 `ser_def` が二重定義されています。

E2004: `sit_def` not defined.

[SIT \(System Information Table\) 情報](#)の開始宣言 `sit_def` が定義されていません。

E2005: `Illegal sit_def`.

[SIT \(System Information Table\) 情報](#)の開始宣言 `sit_def` が誤った場所で行われています。

E2006: `sit_def` already defined.

[SIT \(System Information Table\) 情報](#)の開始宣言 `sit_def` が二重定義されています。

E2007: `Out of sit_def division`.

[SIT \(System Information Table\) 情報](#)に含まれるデータが [SIT \(System Information Table\) 情報](#)の開始宣言 `sit_def` 前に定義されています。

E2012: `rxsers` not defined.

[RX シリーズ情報](#) `rxsers` が定義されていません。

E2013: `Illegal rxsers`.

[RX シリーズ情報](#) `rxsers` の定義が誤った場所で行われています。

E2014: `rxsers` already definiend.

[RX シリーズ情報](#) `rxsers` が二重定義されています。

E2101: `Integer overflow`.

数値が 32 ビット・データ範囲を越えています。

E2102: `Syntax error`.

システム・コンフィギュレーション・ファイルの記述形式が誤っています。

E2103: `Word too long`.

シンボル名が最大文字数を越えています。

E2104: `Address out of range`.

アドレスに範囲外の値を指定しています。

E2105: `Address must be aligned by 2`.

アドレスには、2 バイト境界値を指定します。

E2106: `Stack size out of range`.

スタック・サイズに範囲外の値を指定しています。

E2107: `Symbol name already defined`.

シンボル名 *name* が二重定義されています。

E2201: `System stack size not defined`.

システム・スタック情報 `intstk` が定義されていません。

E2202: `System stack size already defined`.

システム・スタック情報 `intstk` が二重定義されています。

E2203: Interrupt level *name* already defined.

同一割り込み要因 *name* に対して 2 つ以上の間接起動割り込みハンドラが登録されています。

E2204: Interrupt level of system clock already defined.

クロック割り込み要因 *clkhdr* が二重定義されています。

E2206: Priority level already defined.

[システム最大値情報](#) *maxpri* が二重定義されています。

E2207: Priority level out of range.

[システム最大値情報](#) *maxpri* に範囲外の値を指定しています。

E2208: Task group *name* already defined.

タスク実行権グループ名 *name* が二重定義されています。

E2209: Task not defined.

[タスク情報](#) *tsk* が定義されていません。

E2210: Too many tasks.

[タスク情報](#) *tsk* が 128 個以上定義されています。

E2211: Task *name* already defined.

タスク名 *name* が二重定義されています。

E2212: Task group *name* in task not defined.

タスク実行権グループ名 *name* が定義されていません。

E2213: Task priority out of range.

初期優先度 *pri* が 0x1 ~ 0x1f の範囲外です。

E2214: Too many eventflags.

[イベントフラグ情報](#) *flg* が 128 個以上定義されています。

E2215: Eventflag *name* already defined.

イベントフラグ名 *name* が二重定義されています。

E2216: Too many 1bit eventflags.

[1 ビット・イベントフラグ情報](#) *flg1* が 128 個以上定義されています。

E2217: 1bit eventflag *name* already defined.

1 ビット・イベントフラグ名 *name* が二重定義されています。

E2218: Too many Semaphores.

[セマフォ情報](#) *sem* が 128 個以上定義されています。

E2219: Semaphore *name* already defined.

セマフォ名 *name* が二重定義されています。

E2220: Semaphore resource count out of range.

初期資源数 *init_cnt* が 0x0 ~ 0x7f の範囲外です。

E2221: Too many mailboxes.

[メールボックス情報](#) *mbx* が 128 個以上定義されています。

E2222: Mailbox *name* already defined.

メールボックス名 *name* が二重定義されています。

E2223: Too many fixed-size memorypools.

[固定長メモリ・プール情報](#) *mpf* が 128 個以上定義されています。

E2224: Fixed-size memorypool *name* already defined.

固定長メモリ・プール名 *name* が二重定義されています。

E2225: Memory block size or block count is 0.

基本ブロック・サイズ *blk_siz*, または, メモリ・ブロックの総数 *blk_cnt* に 0x0 が定義されています。

E2226: Memory block size must be aligned by 4.

基本ブロック・サイズ *blk_siz* には, 4 バイト境界値を指定します。

E2227: Memory total size exceeds 4Gbytes.

システム・メモリの合計サイズが 4G バイトを越えています。

E2228: Too many cyclic handlers.

[周期起動ハンドラ情報](#) *cyc* が 128 個以上定義されています。

E2229: Cyclic handler *name* already defined.

周期起動ハンドラ名 *name* が二重定義されています。

E2230: Interval time out of range.

周期起動ハンドラの起動時間間隔 *intvl* が 0x1 ~ 0xffffffff の範囲外です。

E2231: Too many variable-size memorypools.

[可変長メモリ・プール情報](#) *mpl* が 128 個以上定義されています。

E2232: Variable-size memorypool *name* already defined.

可変長メモリ・プール名 *name* が二重定義されています。

E2233: Too small variable-size memorypool size.

[可変長メモリ・プール情報](#) *mpl* が 128 個以上定義されています。

E2234: Variable-size memorypool size must be aligned by 4.

可変長メモリ・プールのサイズには, 4 バイト境界値を指定します。

E2235: Illegal trace.

トレース情報 *trace* の定義が誤った場所で行われています。

E2236: Trace already defined.

トレース情報 *trace* が二重定義されています。

E2237: regmod already defined.

レジスタ・モード *regmod* が二重定義されています。

E2238: The character cannot specify for register_mode.

レジスタ・モード *reg_mode* に不正なキー・ワードが用いられています。

6.4.3 警告

次に、警告が発生した際に出力されるメッセージを示します。

なお、メッセージ内のイタリック書体 (*name* 等) は、警告が発生した際に決定されるものです。

W2201: Use *pri1* priority level, but priority *pri2* task is defined, define *pri2* priority level are used.

タスクの優先度範囲として *pri1* を定義していますが、タスクの初期優先度として *pri2* が指定されています。

なお、CF850 は、タスクの優先度範囲として *pri2* が指定されたものとして処理を続行します。

W2202: Task group *name* is not used in task, ignored.

タスク実行権グループとして *name* が定義されていますが、[タスク情報](#)内での指定が行われていません。

なお、CF850 は、タスク実行権グループ *name* が定義されていなかったものとして処理を続行します。

W2203: Stack size must be aligned by 4, round up stack size.

スタック・サイズとして定義された値が 4 バイト境界値ではありません。

なお、CF850 では、スタック・サイズを 4 バイト境界に切り上げて処理を続行します。

W2204: Register mode specified in CF file is different -reg option. (-reg option assumed)

[システム情報](#)として定義されたレジスタ・モードと起動オプションとして指定されたレジスタ・モードが異なっています。

なお、CF850 では、[システム情報](#)で定義されたレジスタ・モードを無視し、起動オプションを有効情報として扱い処理を続行します。

W2205: Nested command file "*file_name*".

コマンド・ファイル *file_name* で不正な起動オプション *@cmd_file* が定義されています。

CF850V4 は、定義されている *@cmd_file* を無視して処理を続行します。

付録 A ウィンドウ・リファレンス

本付録では、CF850 の起動オプションを CA850 が提供する統合開発環境プラットフォーム PM+ から設定する際に必要となる各種ダイアログについて説明しています。

A.1 概要

表 A-1 に、各種ダイアログの一覧を示します。

表 A-1 ダイアログの一覧

ダイアログ名	機能概要
[OS の選択] ダイアログ	以下に示した情報を入力するためのダイアログです。 - リアルタイム OS 名
[RX850 詳細設定] ダイアログ	以下に示した情報を CF850 に対する起動オプションとして設定するためのダイアログです。 - システム・コンフィギュレーション・ファイル名 - システム情報テーブル・ファイル名 - システム情報ヘッダ・ファイル名
[システム・コンフィギュレーション・ファイルの指定] ダイアログ	既存のシステム・コンフィギュレーション・ファイルを読み込むためのダイアログです。
[RX850 ERROR] ダイアログ	エラー情報を表示するためのダイアログです。

A.2 ウィンドウ解説

次頁から、各種ダイアログについて、以下の記述フォーマットにしたがって解説します。

1) ↓

名称

2) → **概要**

3) → **表示イメージ**

4) → **機能**

- 1) 名称
ダイアログの名称を示しています。
- 2) 概要
ダイアログの機能概要、および、オープン方法を示しています。
- 3) 表示イメージ
ダイアログの表示イメージを示しています。
- 4) 機能
ダイアログの機能詳細を構成要素別に示しています。

[OS の選択] ダイアログ

概要

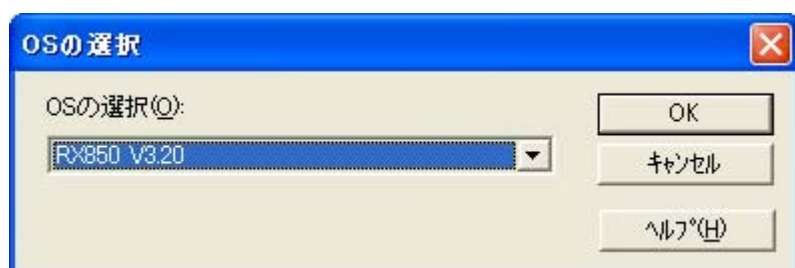
以下に示した情報を入力するためのダイアログです。

- リアルタイム OS 名

なお、本ダイアログは、次の方法でオープンすることができます。

- PM+ のメイン・ウィンドウにおいて、[ツール (T)] → [OS の選択 (Q)...] を選択

表示イメージ



機能

1) [OS の選択 (Q)] エリア

- リスト・ボックス
リアルタイム OS の名前 (リアルタイム OS 名) を選択します。
なお、本リスト・ボックスにおいて指定可能なメニューは、RX850 V3.xx に限られます。

2) 機能ボタン

- < OK > ボタン
[\[RX850 詳細設定 \] ダイアログ](#)をオープンします。
- < キャンセル > ボタン
本ダイアログを閉じます。
- < ヘルプ (H) > ボタン
本ダイアログのオンライン・ヘルプをオープンします。

[RX850 詳細設定] ダイアログ

概要

以下に示した情報を CF850 に対する起動オプションとして設定するためのダイアログです。

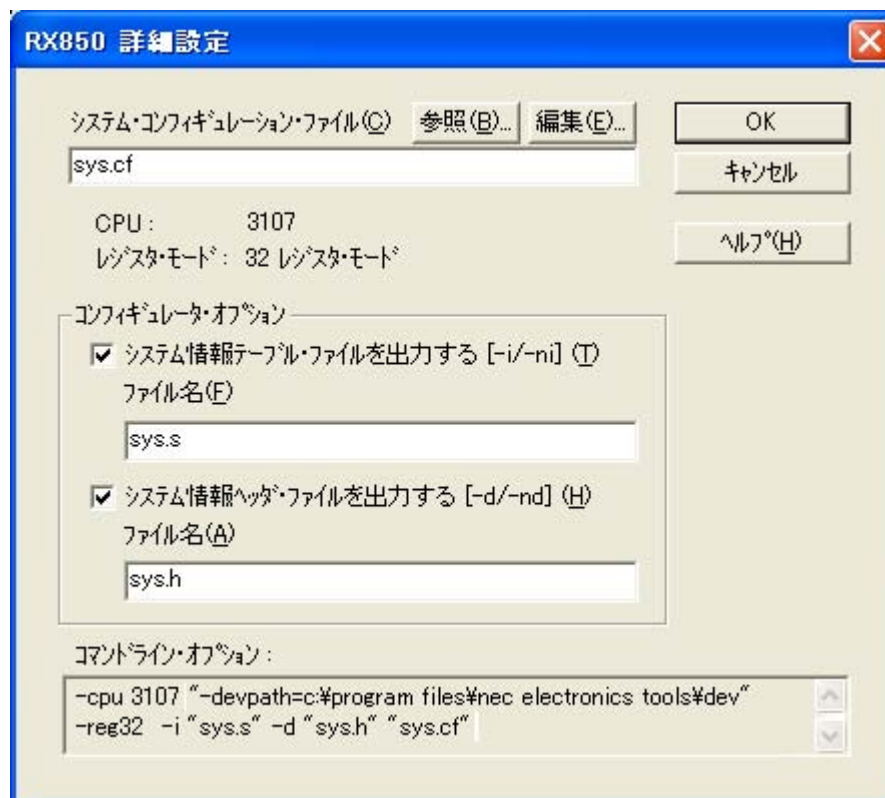
- システム・コンフィギュレーション・ファイル名
- システム情報テーブル・ファイル名
- システム情報ヘッダ・ファイル名

なお、本ダイアログは、次の方法でオープンすることができます。

- [OS の選択] ダイアログにおいて、“RX850 V3.xx” を選択したあと、< OK > ボタンをクリック

備考 品種指定名、デバイス・ファイルの検索対象フォルダ名、および、レジスタ・モードについては、CA850 が提供する統合開発環境プラットフォーム PM+ の [プロジェクトの設定] ダイアログで指定された情報が反映されるため、本ダイアログにおいて設定する必要がありません。
 なお、[プロジェクトの設定] ダイアログについての詳細は、「PM+ ユーザーズ・マニュアル」を参照してください。

表示イメージ



機能

1) [システム・コンフィギュレーション・ファイル(C)] エリア

- テキスト・ボックス
CF850 への入力ファイル名 (システム・コンフィギュレーション・ファイル名) を指定します。

注意 入力ファイル名として指定可能な文字数は、パス名を含めて 255 文字以内に限られます。

- < 参照 (R)... > ボタン
[システム・コンフィギュレーション・ファイルの指定] ダイアログをオープンします。
- < 編集 (E)... > ボタン
[編集] ウインドウをオープンします。

注意 1 [編集] ウインドウについての詳細は、「PM+ ユーザーズ・マニュアル」を参照してください。

注意 2 [編集] ウインドウに表示されているシステム・コンフィギュレーション・ファイルを PM+ 付属のエディタ idea-L を用いて操作する際には、本ダイアログ、および、[OS の選択] ダイアログを閉じる必要があります。

2) [コンフィギュレータ・オプション] エリア

- [システム情報テーブル・ファイルを出力する [-i/-ni] (I)] チェック・ボックス
CF850 を起動した際、システム情報テーブル・ファイルを出力するか否かを指定します。
 チェック状態 : [ファイル名 (E)] エリアで指定されたファイル名でシステム情報テーブル・ファイルを出力します。
 非チェック状態 : システム情報テーブル・ファイルを出力しません。
- [ファイル名 (E)] エリア
CF850 を起動した際の出力ファイル名 (システム情報テーブル・ファイル名) を指定します。
 注意 出力ファイル名として指定可能な文字数は、パス名を含めて 255 文字以内に限られます。
- [システム情報ヘッダ・ファイルを出力する [-d/-nd] (H)] チェック・ボックス
CF850 を起動した際、システム情報ヘッダ・ファイルを出力するか否かを指定します。
 チェック状態 : [ファイル名 (A)] エリアで指定されたファイル名でシステム情報ヘッダ・ファイルを出力します。
 非チェック状態 : システム情報ヘッダ・ファイルを出力しません。
- [ファイル名 (A)] エリア
CF850 を起動した際の出力ファイル名 (システム情報ヘッダ・ファイル名) を指定します。
 注意 出力ファイル名として指定可能な文字数は、パス名を含めて 255 文字以内に限られます。

3) [コマンドライン・オプション] エリア

[システム・コンフィギュレーション・ファイル (C)] エリア、および、[コンフィギュレータ・オプション] エリアで指定された情報 (統合開発環境プラットフォーム PM+ の [プロジェクトの設定] ダイアログで指定された情報を含む) を、CF850 に対するコマンド入力形式として表示します。

4) 機能ボタン

- < OK > ボタン
[システム・コンフィギュレーション・ファイル (C)] エリア、および、[コンフィギュレータ・オプション] エリアにおいて指定した情報 (統合開発環境プラットフォーム PM+ の [プロジェクトの設定] ダイアログで指定された情報を含む) を “CF850 に対する起動オプション” として PM+ に反映したあと、本ダイアログを閉じます。
- < キャンセル > ボタン
本ダイアログを閉じます。
- < ヘルプ > ボタン
本ダイアログのオンライン・ヘルプをオープンします。

[システム・コンフィギュレーション・ファイルの指定] ダイアログ

概要

既存のシステム・コンフィギュレーション・ファイルを読み込むためのダイアログです。
なお、本ダイアログは、次の方法でオープンすることができます。

- [RX850 詳細設定] ダイアログにおいて、<参照 (R)...> ボタンをクリック

表示イメージ



機能

- 1) [ファイルの場所 (F)] エリア
 - コンボ・ボックス
システム・コンフィギュレーション・ファイルの格納されているフォルダ名を指定します。
- 2) ファイル名表示エリア
[ファイルの場所 (F)] エリア、および、[ファイルの種類 (T)] エリアで指定された情報をもとに、該当ファイルを表示します。
- 3) [ファイル名 (N)] エリア
 - テキスト・ボックス
読み込むシステム・コンフィギュレーション・ファイルのファイル名を指定します。
- 4) [ファイルの種類 (T)] エリア
 - コンボ・ボックス
ファイル名表示エリアに表示させるファイルの種類 (ファイル・タイプ) を指定します。
- 5) 機能ボタン
 - <開く (O)> ボタン
[ファイルの場所 (F)] エリア、および、[ファイル名 (N)] エリアで指定されたシステム・コンフィギュレーション・ファイルを読み込みます。
 - <キャンセル> ボタン
本ダイアログを閉じます。

[RX850 ERROR] ダイアログ

概要

エラー情報を表示するためのダイアログです。

なお、本ダイアログは、[\[RX850 詳細設定 \] ダイアログ](#)などにおいて不正な情報が設定された際、自動的にオープンします。

表示イメージ



機能

1) メッセージ表示エリア

検出したエラーに対応したメッセージ (エラー情報: メッセージ・レベル / 識別番号, エラーの発生要因) を表示します。

以下に、本エリアに表示されるメッセージの一覧を示します。

メッセージ・レベル / 識別番号	エラーの発生要因
E1006:	ファイル名が長すぎます。
E1022:	システム・コンフィギュレーション・ファイル名, またはパス名が不正です。
E1023:	システム情報テーブル・ファイル名, またはパス名が不正です。
E1024:	システム情報ヘッダ・ファイル名, またはパス名が不正です。
E2011:	システム・コンフィギュレーション・ファイル名が指定されていません。
E2012:	システム情報テーブル・ファイル名が指定されていません。
E2014:	システム情報ヘッダ・ファイル名が指定されていません。
E2040:	オンライン・ヘルプ・ファイルが見つかりません。

2) 機能ボタン

- < OK > ボタン

本ダイアログを閉じます。

索引

Numerics

1 ビット・イベントフラグ情報 49

C

cf850.exe 17
cf850_ghs.exe 19
CF850 用 DLL ファイル 17
 rx703100p.dll 17

M

makefile 21

O

OS 15
[OS の選択] ダイアログ 78

P

PC インタフェース・ボード 15

R

rx703100p.dll 17
RX850 13
 インストール 16
 開発環境 14
 システム構築 23
 実行環境 14
 特徴 13
[RX850 ERROR] ダイアログ 82
[RX850 詳細設定] ダイアログ 79
RX シリーズ情報 41

S

sample.bld 20
sample.prj 18
sample.prw 18
SIT 情報 39
 1 ビット・イベントフラグ情報 49
 イベントフラグ情報 48
 可変長メモリ・プール情報 53
 間接起動割り込みハンドラ情報 51
 固定長メモリ・プール情報 52
 システム最大値情報 44
 システム情報 42
 周期起動ハンドラ情報 54
 セマフォ情報 47

タスク実行権グループ情報 45
タスク情報 46
メールボックス情報 50
stdrx850.h 17
stdrx850.inc 17

あ

アイドル・ハンドラ	30
アンインストール	16

い

イベントフラグ情報	48
インサーキット・エミュレータ	14
インサーキット・エミュレータ用 I/O ボード	14
インストール	16
インストレーション	16
アンインストール	16
インストール	16
フォルダ構成	17

う

ウインドウ・リファレンス	76
[OS の選択] ダイアログ	78
[RX850 ERROR] ダイアログ	82
[RX850 詳細設定] ダイアログ	79
[システム・コンフィギュレーション・ファイルの指定] ダイアログ	81

か

開発環境	14
ソフトウェア環境	15
ハードウェア環境	14
可変長メモリ・プール情報	53
間接起動割り込みハンドラ	31
間接起動割り込みハンドラ情報	51

き

起動オプション	64
コマンド・ファイル	67

く

クロス・ツール	15
---------------	----

け

警告	75
----------	----

こ

固定長メモリ・プール情報	52
コマンド・ファイル	67
コンフィギュレーション情報	39
SIT 情報	39
リアルタイム OS 情報	39

コンフィギュレータ	14
cf850.exe	17
cf850_ghs.exe	19
コマンド入力例	68
操作方法	63
メッセージ	70

し

システム構築	23
システム・コンフィギュレーション・ファイル	38
記述上の注意点	55
記述方法	38
記述例	56
システム最大値情報	44
システム情報	42
システム情報テーブル・ファイル	26
システム情報ヘッダ・ファイル	26
システム初期化部	27
初期化ハンドラ	29
ニュークリアス初期化部	29
ブート処理	28
割り込みエントリ	29
システム・パフォーマンス・アナライザ	15
実行環境	14
周辺コントローラ	14
動作対象 CPU	14
周期起動ハンドラ	31
周期起動ハンドラ情報	54
周辺コントローラ	14
情報ファイル	26
システム情報テーブル・ファイル	26
システム情報ヘッダ・ファイル	26
初期化データの退避領域	31
初期化ハンドラ	29
処理プログラム	31
間接起動割り込みハンドラ	31
周期起動ハンドラ	31
タスク	31
直接起動割り込みハンドラ	31

せ

セマフォ情報	47
--------------	----

そ

ソフトウェア環境	15
OS	15
クロス・ツール	15
システム・パフォーマンス・アナライザ	15
タスク・デバッグ	15
デバッグ	15

た

タスク	31
-----------	----

タスク実行権グループ情報	45
タスク情報	46
タスク・デバッグ	15

ち

致命的エラー	71
致命的でないエラー	72
直接起動割り込みハンドラ	31

て

デバッグ	15
------------	----

と

動作対象 CPU	14
特徴	13

に

ニュークリアス初期化部	29
-------------------	----

は

ハードウェア環境	14
PC インタフェース・ボード	15
インサーキット・エミュレータ	14
インサーキット・エミュレータ用 I/O ボード	14
ホスト・マシン	14

ひ

標準ヘッダ・ファイル	17
stdrx850.h	17
stdrx850.inc	17
ビルド・ファイル	20
sample.bld	20

ふ

ブート処理	28
フォルダ構成	17
プロジェクト・ファイル	18
sample.prj	18

ほ

ホスト・マシン	14
---------------	----

め

メールボックス情報	50
メッセージ	70
警告	75
致命的エラー	71
致命的でないエラー	72

り

リアルタイム OS 情報	39
RX シリーズ情報	41
リンク・ディレクティブ・ファイル	32

ろ

ロード・モジュール	33
-----------------	----

わ

ワーク・スペース・ファイル	18
sample.prw	18
割り込みエントリ	29

【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

【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。