

RL78開発環境移行ガイド

R8C/M16C, H8S/H8SXからRL78への移行

(統合開発環境編)

(High-performance Embedded Workshop→CS+)

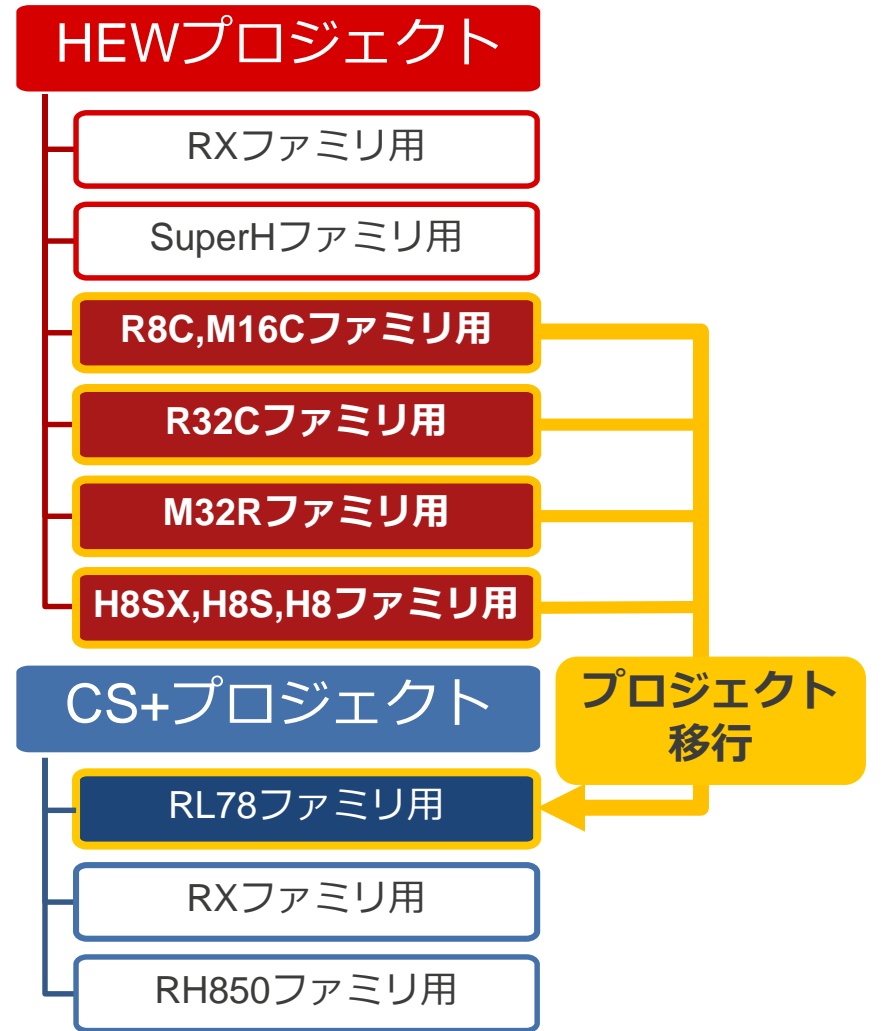
2017/4/7

R20UT2087JJ0103

ソフトウェア事業部ソフトウェア技術部
ルネサスシステムデザイン株式会社

はじめに

- 本資料は、統合開発環境High-performance Embedded Workshop（以降、HEW）で作成したプロジェクトを統合開発環境CS+のRL78ファミリ用コンパイラ(以降、CC-RL)のプロジェクトに移行する際の注意事項を記載しています。
- 本資料で対象とする、CS+／HEW／CC-RLのバージョンは以下の通りです。
 - CS+ for CC V5.00.00
 - HEW V4.09.01.007
 - CC-RL V1.04.00



アジェンダ

- セクション配置 ページ 04

- マイコン制御機能 ページ 08
 - 周辺機能制御 ページ 09

 - 割り込み制御 ページ 11

 - オンチップデバッグ設定 ページ 13

セクション配置

セクション配置 (1/3)

- プログラムやデータのセクションの配置の指定は、プロパティのリンクオプションで設定してください。

CC-RL のプロパティ

変数/関数配置情報

セクション

セクションを自動的に配置する	いいえ
セクションの開始アドレス	const, text, RLIB, SLIB, textf, constf, data, sdata/03000, dataR, bss/FAF00, sdataR, sbss/FFE20
外部定義シンボルをファイル出力するセクション	外部定義シンボルをファイル出力するセクション[0]
ROMからRAMへマップするセクション	ROMからRAMへマップするセクション[2]

ペリファイ
メッセージ
その他

セクション

共通オプション / コンパイラ・オプション / アセンブル・オプション / **リンク・オプション** / ヘキサ出力オプション

配置は、コンパイラが生成するセクション名を指定

セクション設定

アドレス	セクション
0x02000	.const
	.text
	.data
	.sdata
	.textf
	.constf
0xFE00	.dataR
	.bss
	.stack_bss
0xFFE20	.sdataR
	.sbss

追加(A)...
変更(M)...
複数割り付け(O)...
削除(R)
↑(U) ↓(D)
インポート(I)...
エクスポート(E)...
OK キャンセル ヘルプ(H)

任意のアドレスに任意のセクションを指定可能

セクション配置 (2/3)

- CC-RLでは、デフォルト・セクション名でセクションを生成します。

デフォルト・セクション名	再配置属性	内容
.callt0	CALLT0	callt関数呼び出しのテーブル用セクション
.text	TEXT	コード部用セクション (near領域配置)
.textf	TEXTF	コード部用セクション (far領域配置)
.textf_unit64kp	TEXTF_UNIT64KP	コード部用セクション (セクションを先頭が偶数番地になるように、64KB-1境界にまたがらないように配置)
.const	CONST	ROMデータ (near領域配置) (ミラー領域内)
.constf	CONSTF	ROMデータ (far領域配置)
.data	DATA	初期化データ用セクション (初期値あり、near領域配置)
.dataf	DATAF	初期化データ用セクション (初期値あり、far領域配置)
.sdata	SDATA	初期化データ用セクション (初期値あり、saddr配置変数)
.sbss_bit	SBSS_BIT	ビットデータ領域用セクション (初期値なし、saddr配置変数)
.bss_bit	BSS_BIT	ビットデータ領域用セクション (初期値なし、near領域配置)

セクション配置 (3/3)

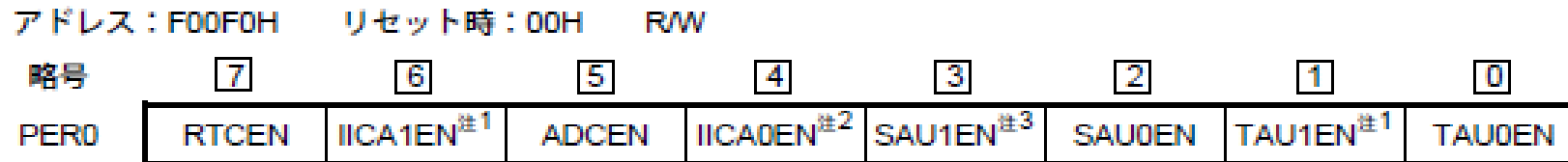
デフォルト・セクション名	再配置属性	内容
.bss	BSS	データ領域用セクション (初期値なし, near領域配置)
.bssf	BSSF	データ領域用セクション (初期値なし, far領域配置)
.sbss	SBSS	データ領域用セクション (初期値なし, saddr配置変数)
.option_byte	OPT_BYTE	ユーザ・オプション・バイト, およびオンチップ・デバッグ指定専用セクション
.security_id	SECUR_ID	セキュリティID指定専用セクション
.vect ^注	AT	割り込みベクタテーブル
.dataR	DATA	初期化データRAM用セクション (初期値あり, near領域配置) スタートアップファイルで定義
.sdataR	DATA	初期化データRAM用セクション (初期値あり, saddr領域配置) スタートアップファイルで定義
.RLIB ^注	TEXTF	ランタイム・ライブラリ コード用セクション
.SLIB ^注	TEXTF	標準ライブラリ コード用セクション

注 #pragma sectionでセクション名を変更することができません。

マイコン制御機能

周辺機能制御

CS+でプロジェクトを作成時にiodefine.hファイルが生成されます。このファイルをインクルードすることで、周辺 I Oに対応したシンボルでのアクセスが可能となります。記述するレジスタ名称は、マニュアルに記載されているレジスタ略称名およびビット名称です。



※ 四角印のビットについては、ビット名での記述が可能

```
#include "iodefine.h"    // 周辺I/Oレジスタ定義ファイルの指定
```

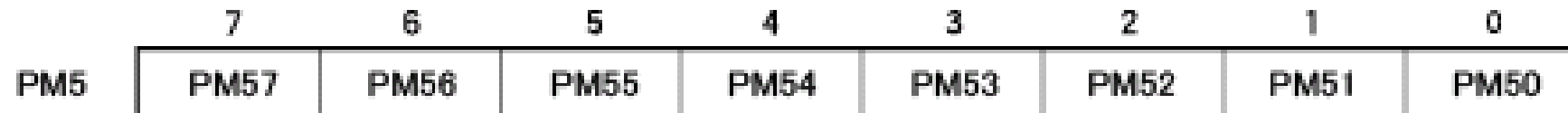
```
PER0 = 0x01;    // レジスタ全体のアクセス
```

```
TAU0EN = 0;    // 1ビットのみアクセス
```

周辺機能制御

アクセス例

- 四角印がないビットに対してのアクセス例



```
#include "iodefine.h" // 周辺I/Oレジスタ定義ファイルの指定
```

```
PM5_bit.no0 = 0; // PM5レジスタのビット0のみ0クリア
```

※iodefine.hで定義されている_bitの名前の付いたバイトとワードの予約語を利用してアクセスしてください。

割り込み制御

- 割り込み禁止・許可を行う組み込み関数
void __DI(void); ... 割り込み禁止
void __EI(void); ... 割り込み許可

- 割り込み関数のベクタ登録と割り込み前後処理のコード出力
#pragma interrupt [()割り込みハンドラ名[(割り込み仕様 [,...])][]]

<割り込み仕様>

項目	形式
ベクタテーブル指定	vect=アドレス
レジスタ・バンク指定	bank={ RB0 RB1 RB2 RB3 }
多重割り込み許可指定	enable={ true false }

割り込み仕様に応じて
・ベクタに関数の先頭番地を登録
・レジスタの退避・回復とRETI命令
などのコードを出力

割り込み制御

プログラム記述例

```
#include "iodefine.h" // 周辺I/Oレジスタ定義ファイルの指定
// ベクタ登録&割り込み関数指定
#pragma interrupt intp0_int(vect=INTP0)
```

割り込み仕様

```
void main(void)
{
    :
    __EI(); // 割り込み許可
    while(1);
}

void intp0_int(void) // INTP0割り込み
{
    :
}
```

割り込み関数名

オンチップデバッグ設定

- オンチップデバッグの動作制御を行う「オンチップ・デバッグ・オプション・バイト」レジスタの値をビルド・ツール→「リンク・オプション」タブの「デバイス」カテゴリで設定します。
- また「ユーザ・オプション・バイト」レジスタの値も設定し、高速内蔵発信回路の周波数の設定と、デバッグ用にウォッチ・ドッグ・タイマを停止しておきます。

プロジェクト・ツリー

sample_pri (プロジェクト)*

- R5F100LE (マイクロコントローラ)
- CC-RL (ビルド・ツール)**
- RL78 シミュレータ (デバッグ・ツール)

ファイル

- ビルド・ツール生成ファイル
 - cstart.asm
 - hdwinit.asm
 - stkinit.asm
 - main.c
 - iodefine.h

プロパティ

CC-RL のプロパティ

- 出力ファイル
- ライブラリ
- デバイス
 - オンチップ・デバッグの許可/禁止をリンク・オプションで設定する (はい(-OCDBG))
 - オンチップ・デバッグ・オプション・バイト制御値 (HEX 84)
 - デバッグ・モニタ領域を設定する (いいえ)
 - ユーザ・オプション・バイトを設定する (はい(-USER_OPT_BYTE))
 - ユーザ・オプション・バイト値 (HEX EEF E8)
 - セルフラム領域への配置を制御する (いいえ)
- 出力コード

オンチップ・デバッグの許可/禁止をリンク・オプションで設定する
オンチップ・デバッグ用制御値の設定を行う場合に指定します。
オンチップ・デバッグ・オプション・バイト制御値を、必ず設定してください。
rlinkコマンドの-OCDBGオプションに相当します。

「はい」を選択し、制御値は **84h** (オンチップデバッグの動作を許可) を設定

「はい」を選択し、値は **EEFFE8h** (ウォッチドッグタイマ停止、高速内蔵発振：32MHz) を設定

共通オプション / コンパイル・オプション / アセンブル・オプション / **リンク・オプション** / ヘキサ出力オプション / I/O

ルネサス システムデザイン株式会社