

# RH850コンパイラ CC-RH セクションと メモリマッピング

R20UT3212JJ0101

2018.10.24

ソフトウェア開発統括部、ソフトウェア技術部  
ルネサスエレクトロニクス株式会社

# アジェンダ

---

- セクション
- メモリマッピング

ページ 03

ページ 11

# セクション

# デフォルトのセクション構成

## CC-RH がデフォルトで配置するセクション

配置対象	セクション名	ベースレジスタ	アクセス命令	容量制限
プログラム・コード	<b>.text</b>	—	—	制限なし
定数データ	<b>.const</b>	r0(0番地)	movhi+ld/st 2命令	制限なし
初期値ありデータ	<b>.data</b>	r0(0番地)	movhi+ld/st 2命令	制限なし
初期値なしデータ	<b>.bss</b>	r0(0番地)	movhi+ld/st 2命令	制限なし

- セクションによって**アクセス方法と容量**がそれぞれ異なります。
- **#pragma section** 指令により、異なるセクションへの変更が可能です。

# 定数データのセクション構成

配置対象	セクション名	ベースレジスタ	アクセス命令	容量制限
定数データ	<b>.const</b> (デフォルト)	r0(0番地)	movhi+ld/st 2命令	制限なし
	<b>.zconst</b>		ld/st 1命令	32Kバイト
	<b>.zconst23</b>		ld23/st23 1命令	4Mバイト

- movhi/ld/stは4バイト長の命令、ld23/st23は6バイト長の命令です。

# 初期値ありデータのセクション構成(data属性)

配置対象	セクション名	ベースレジスタ	アクセス命令	容量制限
初期値ありデータ	<b>.data (デフォルト)</b>	r0(0番地)	movhi+ld/st 2命令	制限なし
	.zdata		ld/st 1命令	32Kバイト
	.zdata23		ld23/st23 1命令	4Mバイト
	.edata	ep(エレメント・ポインタ)	ld/st 1命令	64Kバイト
	.edata23		ld23/st23 1命令	8Mバイト
	.tdata4		sld/sst 1命令	16バイト
	.tdata5		sld/sst 1命令	32バイト
	.tdata7		sld/sst 1命令	128バイト
	.tdata8		sld/sst 1命令	256バイト
	.sdata	gp(グローバル・ポインタ)	ld/st 1命令	64Kバイト
	.sdata23		ld23/st23 1命令	8Mバイト

■ sld/sstは2バイト長の命令です。

# 初期値なしデータのセクション構成(bss属性)

配置対象	セクション名	ベースレジスタ	アクセス命令	容量制限
初期値なしデータ	<b>.bss (デフォルト)</b>	r0(0番地)	movhi+ld/st 2命令	制限なし
	.zbss		ld/st 1命令	32Kバイト
	.zbss23		ld23/st23 1命令	4Mバイト
	.ebss	ep(エレメント・ポインタ)	ld/st 1命令	64Kバイト
	.ebss23		ld23/st23 1命令	8Mバイト
	.tbss4		sld/sst 1命令	16バイト
	.tbss5		sld/sst 1命令	32バイト
	.tbss7		sld/sst 1命令	128バイト
	.tbss8		sld/sst 1命令	256バイト
	.sbss	gp(グローバル・ポインタ)	ld/st 1命令	64Kバイト
	.sbss23		ld23/st23 1命令	8Mバイト

# セクションの変更方法

```
#pragma section 属性指定文字 ["セクション名"]  
変数の宣言／定義  
#pragma section default
```

- 「属性指定文字」 に指定可能な文字列はマニュアルに記しております。この文字列によりセクションが決定されます。
- 「セクション名」 は省略可能です。指定した場合は、コンパイラが生成するセクションの前に指定した文字列を付加します。

例1：変数"a","b"を.sdata/.sbss セクションに変更

```
#pragma section gp_disp16  
int a=3;          <- .sdataセクションに配置  
int b;           <- .sbss セクションに配置  
#pragma section default
```

例2：変数"c","d"をuser.zdata23/user.zbss23 セクションに変更

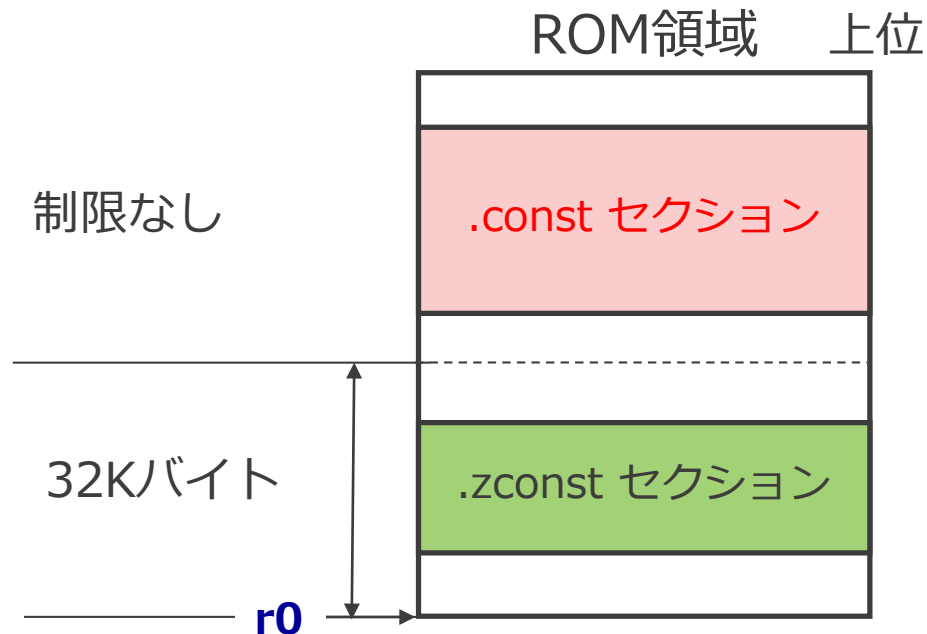
```
#pragma section r0_disp23 "user"  
int c=3;          <- user.zdata23セクションに配置  
int d;           <- user.zbss23 セクションに配置  
#pragma section default
```



# 定数データのセクション変更

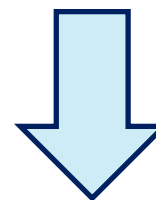
例：.constセクションの定数データ"c"を.zconst セクションに変更

```
#pragma section zconst  
const int c = 1;    <- .zconst セクションに配置  
#pragma section default
```



.const セクション r0相対/2命令

```
movhi HIGHW1(#_c), r0, r2  
ld.w LOWW(#_c)[r2], r5
```



メリット：サイズ・アクセス効率向上  
デメリット：配置制限

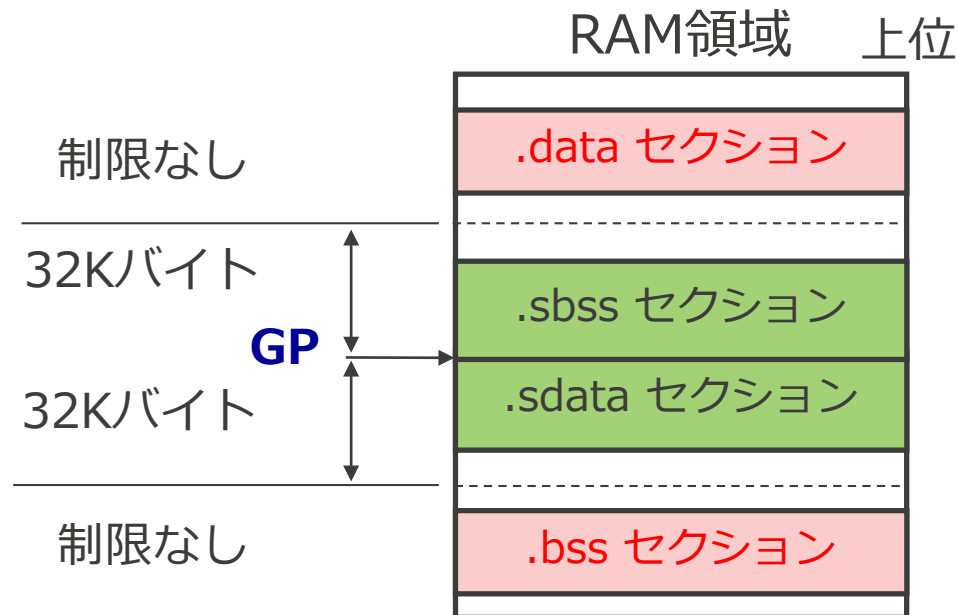
.zconst セクション r0相対/1命令

```
ld.w !_c[r0], r5
```

# 初期値あり・なしデータのセクション変更

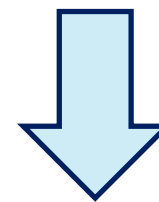
例：.data/.bssセクションの変数"a","b"を.sdata/.sbss セクションに変更

```
#pragma section gp_disp16
int a=3;          <- .sdataセクションに配置
int b;           <- .sbss セクションに配置
#pragma section default
```



.data セクション r0相対/2命令

```
movhi HIGHW1(#_a), r0, r2
ld.w LOWW(#_a)[r2], r5
```



メリット：サイズ・アクセス効率向上  
デメリット：配置制限

.sdata セクション gp相対/1命令

```
ld.w $_a[gp], r5
```

# メモリマッピング

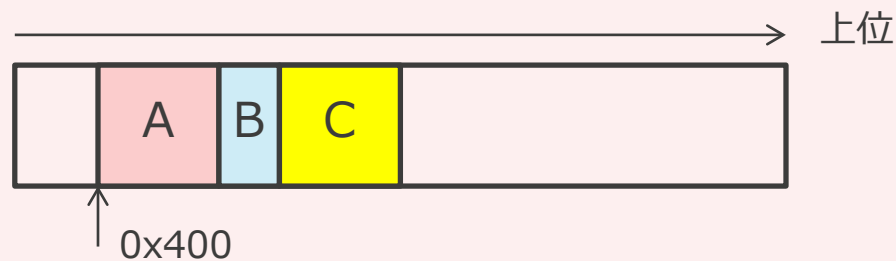
# リンカオプション“-START”による指定

セクションの配置アドレスはリンカオプション“-start”で指定します。

```
-start=section1,section2,.../address
```

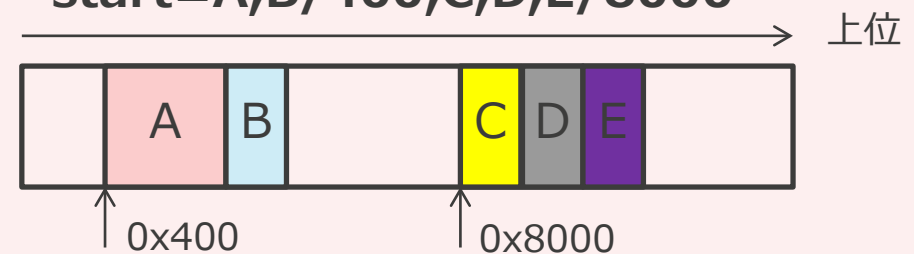
例1：0x400番地から上位方向にA, B, C  
セクションを配置させる場合

**-start=A,B,C/400**



例2：0x400番地から上位方向にA, B セク  
ションを、0x8000番地から上位方向に  
C, D, E セクションを配置させる場合

**-start=A,B/400,C,D,E/8000**



# 統合開発環境CS+からの指定方法

CS+上からGUI操作により-startオプションを指定することも可能です。

[リンク・オプション]タブ => [セクション]カテゴリ => [セクションの開始アドレス]から [セクション設定]ダイアログを起ち上げて指定してください。

The screenshot shows the 'プロパティ' (Properties) window for 'CC-RH のプロパティ'. The 'リンク・オプション' (Link Options) tab is selected. Under the 'セクション' (Section) category, 'セクションの開始アドレス' (Section Start Address) is highlighted with a red box. A red arrow points from this entry to the 'セクション設定' (Section Settings) dialog box. The dialog box contains a table with the following data:

アドレス	セクション
0x00000000	RESET
	EIINTTBL
	const
	INIT_DSEC...
	INIT_BSEC...
	text
	data
0xFEDE0000	data.R
	bss
	stack.bss

---

[Renesas.com](https://www.renesas.com)