変数セクションの追加方法

RL78ファミリ用Cコンパイラ CC-RL

ルネサス システムデザイン株式会社
ツールビジネス本部 ツール技術部

2015/06/30　Rev. 1.00

R20UT3477JJ0100

はじめに

- **本資料は、RL78ファミリ用Cコンパイラ CC-RLを使用して、デフォルトで生成されるセクション名を変更し、新たなセクションを追加する方法を説明しています。**

- **本資料は、次のツール、バージョンで説明をしています。**
  - RL78ファミリ用Cコンパイラ CC-RL V1.01.00
  - **統合開発環境** e2 studio V4.0.0.26
  - **統合開発環境** CS+ V3.01.00

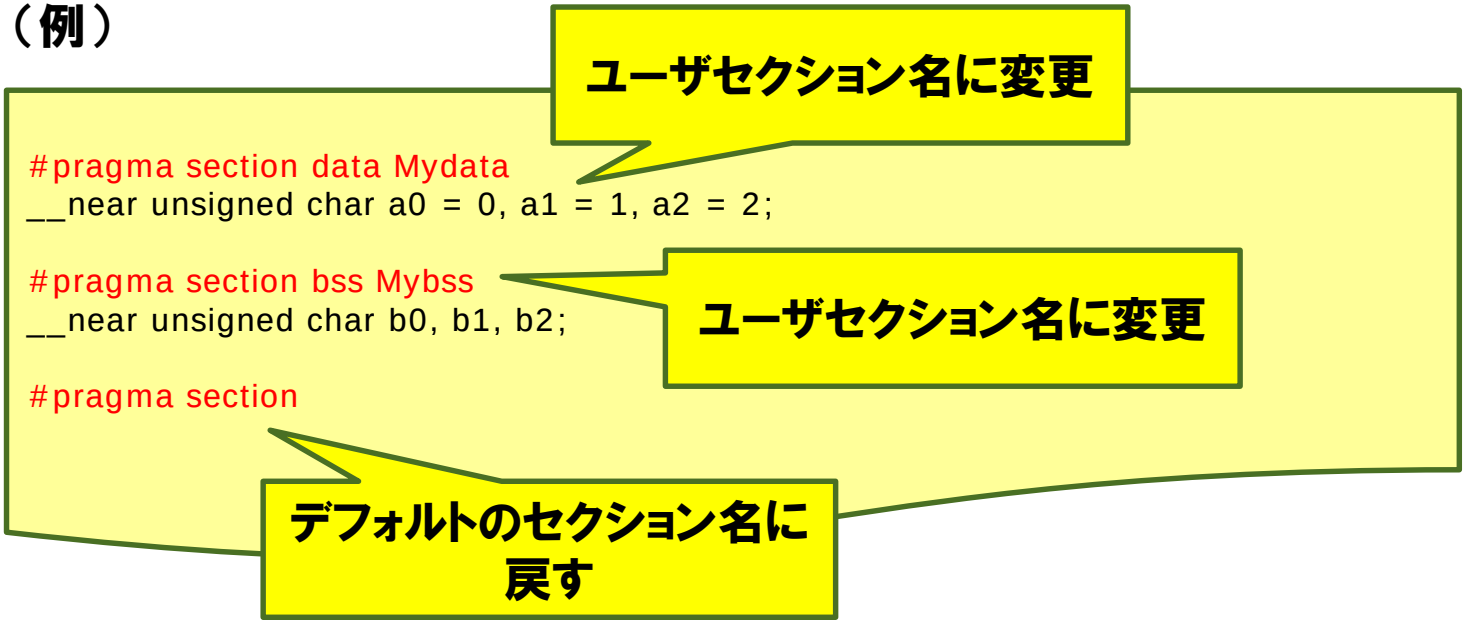
- 変数のセクションを変更するには
- Cソース上でのセクション指定の追加
- リンカの設定の追加
- 初期化処理の追加
  - 初期化ルーチン(C言語)の作成
  - スタートアップルーチン(cstrat.asm)の変更

変数のセクションを変更するには

- Cソース上でのセクション指定の追加
  - #pragma section で変数のセクション名を変更
- リンカの設定の追加
  - 初期値あり変数のセクションをROMからRAMへマップするセクションとして指定
- 初期化処理の追加
  - スタートアップルーチンには、デフォルトのセクションの処理しか入っていないので以下のいずれかの処理の追加が必要
  - 初期化ルーチン(C言語)の作成
    - 初期化テーブル、関数を作成し、その関数を呼び出す
  - スタートアップルーチン(cstrat.asm)の変更
    - デフォルトのセクションの処理しか入っていないので追加が必要
      - 初期値なし変数領域の0初期化処理の追加
      - 初期値あり変数領域への初期値のコピー処理の追加

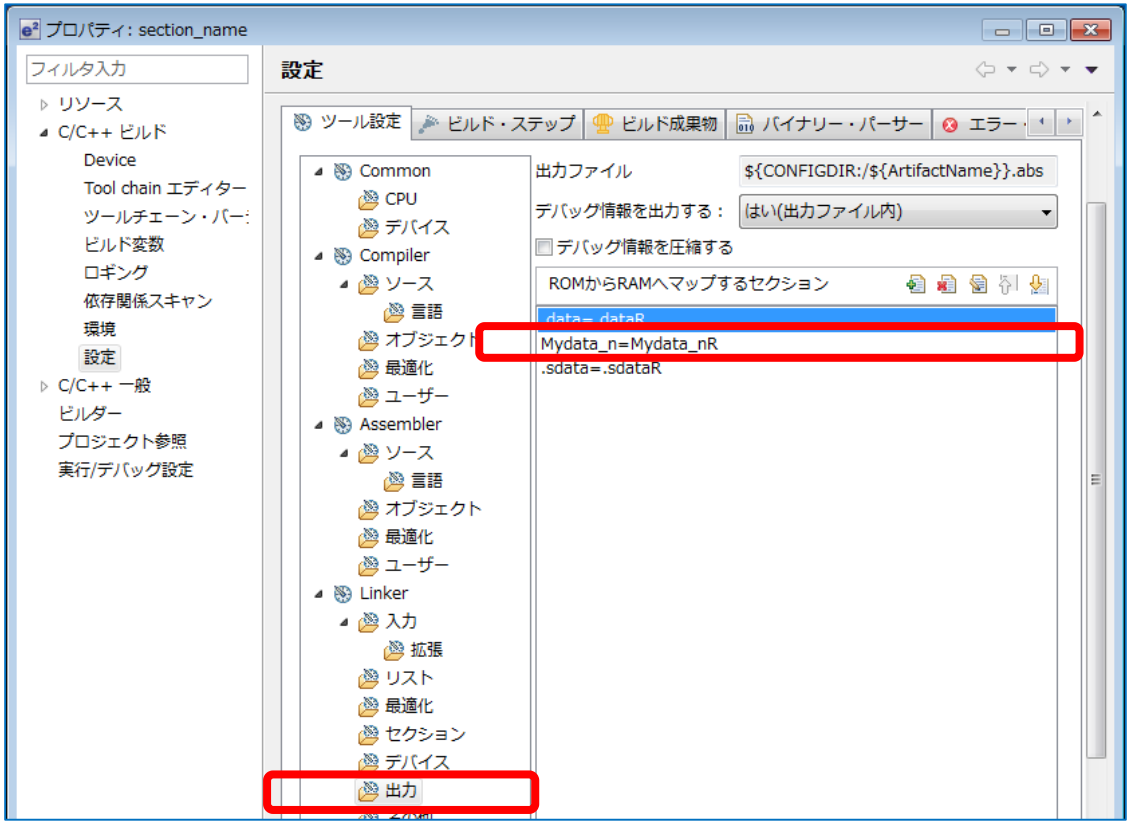
Cソース上でのセクション指定の追加

- #pragma section を使用
  - デフォルトで出力されるセクションの名前を変更する
  - 記述形式
    - #pragma section [*セクション種別*] [*変更セクション名*]
    - セクション種別
      - text、const、data、bss
  - （例）



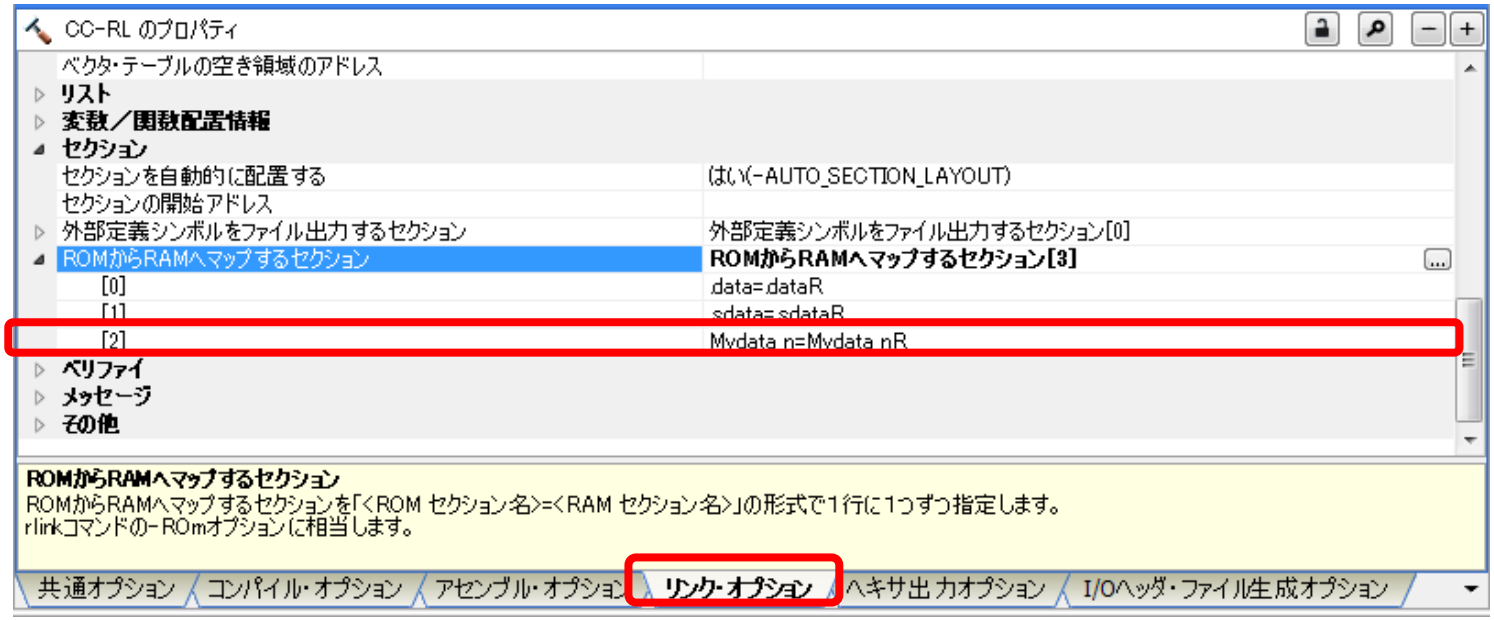
リンカの設定の追加(1/2)

- **初期値あり変数のセクションのROMからRAMへマップするセクションとして指定**
  - **リンカの-romオプションで、セクションを指定する。**
  - **（例）** e² studio



リンカの設定の追加(2/2)

- （例）CS+



初期化ルーチン(C言語)の作成(1/4)

- **初期化テーブル(初期値なし変数用)の定義**
  - **初期化関数で使用するセクションのアドレス、サイズを定義する。**
  - **補足**
    - **「初期化ルーチン(C言語)の作成」では、例を記載していますが、複数のセクションを対象にできるような構造にしています。**
  - **(例)** initsct.c

    青字の部分を#pragma section **で指定して出力されたセクション名に、した、次の処理を追加してください。**

```
#define BSEC_MAX 1                  /*0 初期化するBSS セクションの数*/

const struct bsec_t {
  char __near *ram_sectop;          /* セクション先頭アドレス*/
  char __near *ram_secend;          /* セクション末尾アドレス+1*/
} bsec_table[BSEC_MAX] = {
  {(char __near *)__sectop("Mybss_n"),
   (char __near *)__secend("Mybss_n")}};
```

初期化ルーチン(C言語)の作成(2/4)

- **初期化テーブル(初期値あり変数用)の定義**
  - **初期化関数で使用するセクションのアドレス、サイズを定義する。**
  - **(例)** initsct.c
    青字の部分を#pragma section **で指定して出力されたセクション名に、紫字の部分を**-rom**オプションで指定したセクション名にした、次の処理を追加してください。**

```
#define DSEC_MAX 1                    /*コピーするDATA セクションの数*/

const struct dsec_t {
  char __far *rom_sectop; /* コピー元セクション先頭アドレス*/
  char __far *rom_secend; /* コピー元セクション末尾アドレス+1*/
  char __near *ram_sectop; /* コピー先セクション先頭アドレス*/
} dsec_table[DSEC_MAX] = {
  {__sectop("Mydata_n"),
   __secend("Mydata_n"),
   (char __near *)__sectop("Mydata_nR")}};
```

初期化ルーチン(C言語)の作成(3/4)

- **初期化関数の作成**
  - **初期化テーブルを使用して初期値なし変数の0クリア、初期値あり変数の初期値のコピーする関数を作成する。**
  - main**関数等から、この関数を呼び出す。**

初期化ルーチン(C言語)の作成(4/4)

- （例）initsct.c

```
#define BSEC_MAX 1    /*0 初期化するBSS セクションの数*/
#define DSEC_MAX 1    /* コピーするDATA セクションの数*/

void INITSCT_RL(void)
{
 unsigned int i;
 char __far *rom_p;
 char __near *ram_p;
 for (i = 0; i < BSEC_MAX; i++) {
  ram_p = bsec_table[i].ram_sectop;
  for ( ; ram_p != bsec_table[i].ram_secend; ram_p++) {
   *ram_p = 0;
  }
 }
 for (i = 0; i < DSEC_MAX; i++) {
  rom_p = dsec_table[i].rom_sectop;
  ram_p = dsec_table[i].ram_sectop;
  for ( ; rom_p != dsec_table[i].rom_secend; rom_p++, ram_p++) {
   *ram_p = *rom_p;
  }
 }
}
```

0で初期化

初期値をコピー

スタートアップルーチン(cstrat.asm)の変更(1/2)

- **初期値なし変数領域の0初期化処理の追加**
  - **セクション名を使用して、そのセクションの領域を0クリアする処理を追加する**
  - **(例)**
    青字の部分を#pragma section で指定して出力されたセクション名にした、次の処理を追加してください。

```
                ; clear external variables which doesn't have initial value (Mybss_n)
                MOVW    HL,#LOWW(STARTOF(Mybss_n))
                MOVW    AX,#LOWW(STARTOF(Mybss_n) + SIZEOF(Mybss_n))
                BR      $.L2_Mybss_n
.L1_Mybss_n:
                MOV     [HL+0],#0
                INCW    HL
.L2_Mybss_n:
                CMPW    AX,HL
                BNZ     $.L1_Mybss_n
```

セクションの開始アドレス

セクションのサイズ

0で初期化

スタートアップルーチン(cstrat.asm)の変更(2/2)

- 初期値あり変数領域への初期値のコピー処理の追加
  - セクション名を使用して、初期値をコピーする処理を追加する
  - (例)

    青字の部分を#pragma section で指定して出力されたセクション名に、
    紫字の部分を-romオプションで指定したセクション名にした、次の処理を
    追加してください。

```
                ; copy external variables having initial value (Mydata_n)
                MOV         ES,#HIGHW(STARTOF(Mydata_n))
                MOVW        BC,#LOWW(SIZEOF(Mydata_n))
                BR          $.L2_Mydata_n
.L1_Mydata_n:
                DECW        BC
                MOV         A,ES:LOWW(STARTOF(Mydata_n))[BC]
                MOV         LOWW(STARTOF(Mydata_nR))[BC],A
.L2_Mydata_n:
                CLRW        AX
                CMPW        AX,BC
                BNZ         $.L1_Mydata_n
```

セクションの開始アドレス(ROM側)

セクションのサイズ

初期値をコピー

セクションの開始アドレス(RAM側)

ルネサス システムデザイン株式会社