

持ち込み可能なプログラムの例 (マイコンに書き込まれたプログラムを動作させるためのブートプログラム)

```
/******  
/* 対象マイコン R8C/35A */  
/* ファイル内容 スタートアッププログラム(C言語版) */  
/* バージョン Ver.1.20 */  
/* Date 2010.04.19 */  
/* Copyright ジャパンマイコンカーラー実行委員会 */  
/* ルネサスマイコンカーラー事務局 */  
/* 日立インターメディックス株式会社 */  
/******  
/*=====*/  
/* インクルード */  
/*=====*/  
/*=====*/  
/* スタックサイズの設定 */  
/*=====*/  
#pragma STACKSIZE 0x100  
#pragma ISTACKSIZE 0x100  
/*=====*/  
/* CPU レジスタの宣言 */  
/*=====*/  
#pragma CREG _flg_ flg  
#pragma CREG _isp_ isp  
#pragma CREG _sp_ sp  
#pragma CREG _sb_ sb  
#pragma CREG _fb_ fb  
#pragma CREG _intbh_ intbh  
#pragma CREG _intbl_ intbl  
  
unsigned int _flg_;  
unsigned int _sb_;  
unsigned int _fb_;  
unsigned int * _sp_;  
unsigned int * _isp_;  
unsigned int * _intbh_;  
unsigned int * _intbl_;  
  
/*=====*/  
/* SB の値をコンパイラに設定 */  
/*=====*/  
_asm(" _glb_ _SB_ %n"  
      "_SB_ .equ 0400H ");  
  
/*=====*/  
/* オプション機能選択レジスタの設定 */  
/*=====*/  
_asm(" .ofsreg 0BFH"); /* OFS = 0xbf (ハワーオリセット使用) */  
  
/*=====*/  
/* ID コードの設定 */  
/*=====*/  
_asm(" .id ""%#FFFFFFFFFFFFFF%");  
  
/*=====*/  
/* RAM を初期化する関数の定義 */  
/*=====*/  
#define scopy(X,Y,Z) _asm(" .initsct "X","Y","Z"%n"%  
      " .initsct "X",I,rom"Y",noalign%N"%  
      " mov.w #(topof "X"D)&0ffff,A0%N"%  
      " mov.b #(topof "X"D)>>16,R1H%N"%  
      " mov.w #(topof "X")&0ffff,A1%N"%  
      " mov.w #sizeof "X",R3%N"%
```

" smovf.b")

```
#define sclear(X,Y,Z) _asm(" .initsct "X","Y","Z"%n"%  
      " mov.b #00H,R0L%N"%  
      " mov.w #(topof "X"),A1%N"%  
      " mov.w #sizeof "X",R3%N"%  
      " sstr.b")
```

```
/*=====*/  
/* セクションの先頭アドレスの型定義 */  
/*=====*/  
extern unsigned int _stack_top;  
extern unsigned int _istack_top;  
extern unsigned int _vector_top;
```

```
/*=====*/  
/* ヒープ領域の設定 */  
/*=====*/  
#pragma sectaddress heap_NE,DATA  
#define __HEAPSIZE__ 0x100
```

```
unsigned char heap_area[ __HEAPSIZE__ ]; /* ヒープ領域確保 */  
extern unsigned char _far * _mnext; /* 次に使用できるメモリの先頭アドレス */  
extern unsigned long _msize; /* 残りのバイト数 */
```

```
/*=====*/  
/* 固定割り込みベクタアドレスの設定 */  
/*=====*/  
#pragma sectaddress fvector,ROMDATA 0xffd8
```

```
_asm(" .addr 0FFFFFFFH"); /* 予約 */  
_asm(" .byte 0FFH"); /* OFS2 */  
#pragma interrupt/v _dummy_int /* 未定義命令割り込みベクタ */  
#pragma interrupt/v _dummy_int /* オーバフロー割り込みベクタ */  
#pragma interrupt/v _dummy_int /* BRK 命令割り込みベクタ */  
#pragma interrupt/v _dummy_int /* アドレス一致割り込みベクタ */  
#pragma interrupt/v _dummy_int /* シックステップ割り込みベクタ */  
#pragma interrupt/v _dummy_int /* ウォッチドッグタイムなどの割り込みベクタ */  
#pragma interrupt/v _dummy_int /* アドレスレイト割り込みベクタ */  
#pragma interrupt/v _dummy_int /* 予約 */  
#pragma interrupt/v start /* リセットベクタ */
```

```
/*=====*/  
/* 固定割り込みプログラム */  
/*=====*/  
#pragma sectaddress interrupt,CODE  
#pragma interrupt _dummy_int()  
void _dummy_int(void)  
{  
    /* ダミー関数 */  
}
```

```
/*=====*/  
/* 可変割り込みベクタの設定 */  
/*=====*/  
#pragma sectaddress vector,ROMDATA
```

/* ここではセクション名の設定のみ行う */

```
/*=====*/  
/* RAM の初期化 */  
/*=====*/
```

```

#pragma sectaddress program, CODE
void initsct(void)
{
    sclear("bss_SE", "data", "align");
    sclear("bss_SO", "data", "noalign");
    sclear("bss_NE", "data", "align");
    sclear("bss_NO", "data", "noalign");

    scopy("data_SE", "data", "align");
    scopy("data_SO", "data", "noalign");
    scopy("data_NE", "data", "align");
    scopy("data_NO", "data", "noalign");
}

/*=====*/
/* スタートアッププログラム          */
/*=====*/
#pragma entry start
void start( void )
{
    _isp_ = &_istack_top;      /* ISP に割り込みスタックのアドレス設定 */
    _flg_ = 0x0080;          /* FLG の U="1" */
    _sp_ = &_stack_top;     /* USP にユーザスタックのアドレス設定 */
    _sb_ = 0x0400U;         /* SB 相対アドレッシングの設定 */
    _intbh_ = (unsigned int*)0x00; /* INTBH = vector(上位)に設定 */
    _intbl_ = &_vector_top;  /* INTBL = vector(下位)に設定 */
    initsct();              /* RAM の初期化 */
    _mnext = &heap_area[0]; /* ヒープ領域変数の設定 */
    _msize = (unsigned long)_HEAPSIZE_;
    _fb_ = 0U;              /* FB = 0 */

    main();                 /* main 関数実行 */

    while(1);
}
/*****
/* end of file */
*****/

```