

H8/300Hシリーズ
C言語：アセンブラ
インタフェース仕様

概要：

マイコンのファーム開発においても、開発効率の向上を目指し、C言語による開発が行われる事が一般的になってきた。

そのような中で、H8/300Hシリーズは、32bitレジスタ構成の16bitCPUで、C言語をインプリメントするのに適したCPUである。

しかし、全てのコードをCで記述出来る訳ではない。

ベクトルテーブルの定義や、CPUのフラグ操作、スタックポインタの設定、割り込み処理など、アセンブラで記述しなければ実装できない部分もある。

また、言語により処理内容による適正もあるので、ビット操作、I/O操作などはアセンブラの方が適している場合もある。よって、用途によってCから、アセンブラのサブルーチン呼び出す必要が生じる。このような場合に明確にしておかなければならない事がC言語とアセンブラとのインタフェースである。

この資料は、Cとアセンブラのインタフェースの説明を記述するものとする。

アセンブラ側で宣言した変数を、Cで参照する場合の例：

Cプログラム

```
extern int a, b;

main()
{
    a += b;
}
```

アセンブラプログラム

```
.EXPORT _a, _b
.SECTION D. DATA, ALIGN=2

_a .DATA.W 1
_b .DATA.W 1

.END
```

C側で宣言した変数を、アセンブラで参照する場合の例：

Cプログラム

```
char a, b;
```

アセンブラプログラム

```
.IMPORT _a, _b
.SECTION P. CODE, ALIGN=2

mov.b @_a, r0l
mov.b r0l, @_b

.END
```

Cの変数は、Cのソースファイル上では現れないが、先頭に「_」が付いている。よって、Cのソース上で test という変数名があった場合に、リンクに渡される時は、_test という変数名になっている。

よって、アセンブラ側では、Cの外部変数をアクセスする場合、あるいはC側からアクセスする変数を用意する場合に、先頭に _ を付ける必要がある。

変数に限らず関数名も同様である。

関数呼び出し時の規約：

レジスタに関して

内容を破壊してもいいレジスタ：

ER0、ER1（ 保障しないレジスタ ）

内容を保護しなければならないレジスタ：

ER2～ER6（ 保障するレジスタ ）

ER0、ER1は、引数、関数値の 型により、
引数のレジスタ渡し（ 1st：ER0、2nd：ER1 ）
関数値の返却（ ER0 ）が 使用される。

引数の型がレジスタ渡しに適さない場合、あるいは、引数の個数が多い
場合（ 1st、2nd が ER0、ER1に乗った場合、3番目以降の引数 ）は、ス
タックに積み上げられる。

関数値も、レジスタ渡しに適さない場合は、呼び出し側の変数エリアに
格納する。

レジスタ渡し可能な型：

char、	unsigned char、
short、	unsigned short、
int、	unsigned int、
long、	unsigned long、
float、	ポインタ

レジスタ渡し可能な関数値

R0L：	char、 unsigned char
R0：	short、 unsigned short int、 unsigned int
ER0：	ポインタ（ アドバンスモード ）
ER0：	long、 unsigned long、 float
メモリ：	double、 long double 構造体、 共用体

ROM、RAMの割付け：

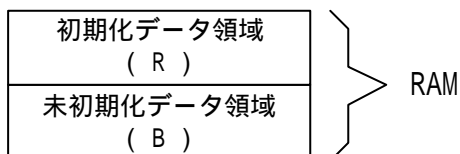
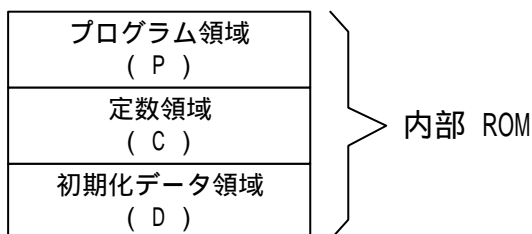
プログラムを ROM化する場合、静的な領域を、以下のように ROMと RAMに分けて割り付けます。

プログラム領域：	(セクション P)	ROM
定数領域	(セクション C)	ROM
未初期化データ領域	(セクション B)	RAM
初期化データ領域	(セクション D)	ROM 、 RAM

初期化データの割り付け：

初期化データ領域は、初期値を持ったデータ領域ですが、値の変更が可能なのでリンク時は、ROM上に置き、プログラム実行開始時に RAM上にコピーします。したがって初期化データ領域は、ROM上と、RAM上に2重に領域を確保する必要があります。但し、初期値を指定した静的変数を変更しないようにプログラムを作成すれば、初期化データの領域は ROM上に置くだけでよく二重に割り付ける必要はありません。

割り込みベクタ



動的領域

内部 RAM

P,C,D,B C コンパイラが生成するデフォルトのセクション名

R リンケージエディタのROM化支援機能で指定したセクション名