

変数を特定のアドレスに配置したい

通常、変数の配置は、コンパイラが決めてしまいますが、グローバル変数であれば、セクション単位で配置の順序、アドレスを指定することは可能です。以下のようにリンクスクリプトを編集し、ソースコードを本リンクスクリプトに適合させてください。

例えば、`unsigned char test[16]`という配列を0x100番地に割り付けたい場合は、リンクスクリプトとソースコードを以下のようにします。

<リンクスクリプト例>

SECTIONS

{

 .var_ram_section1 (0x100) : ←RAM配置アドレスを定義

 {

 *(.var_ram_section1); ←任意のセクション名を定義

 . = ALIGN(0x10); ←16バイトのエリア確保

 } > iram

 : : : :

<ソースコード例>

`unsigned char __attribute__((section (".var_ram_section1"))) test[16];`

←スクリプトファイルで定義したセクション名

変数を特定のアドレスに配置したい

但し、この場合は、RAMエリアに対してもコードが吐き出され、Build時のpsaファイルを生成する過程で、エラーがでますが、問題はありません。

変数を特定のアドレスに配置したい

また、ソースファイル内に定義された変数すべてを特定のアドレスに配置するのであれば、リンクスクリプトを定義するだけで、特定のアドレスに配置することが可能です。

例えば、“variables.c”に定義された変数すべてを0x200番地から配置したい場合は以下のよう記述できます。

<リンクスクリプト例>

MEMORY

```
{          vram                : ORIGIN = 0x000200,          LENGTH = 16
}
```

←RAM配置アドレスを定義

SECTIONS

```
{ .bss (NOLOAD) :
{   PROVIDE (__START_bss = .) ;
    *(EXCLUDE_FILE (*variables.o) .bss) ←bss領域からvariables.cの変数を除外
    PROVIDE (__END_bss = .) ;
} > iram

.var_ram_section2 (NOLOAD) : ←任意のセクション名を定義
{   PROVIDE (__START_var_ram_section2 = .) ;
    KEEP(*variables.o(.bss))
    PROVIDE (__END_var_ram_section2 = .) ;
} > vram

:          :          :          :
```