

リンクスクリプトとスタートアップファイルを編集することにより、プログラムをRAM上で実行させることが可能です。

1)リンクスクリプトの編集

以下のように赤字と青字部分を編集してください。また、青字部分はRAM上に配置したいオブジェクトファイル名を記述します。

```
/* Default linker script, for normal executables */
...
SECTIONS
{
...
.text :
{
    PROVIDE (__START_text = .) ;
    *(.text.*)
    *(EXCLUDE_FILE (*ObjectFileName.o) .text)
    *(.text)
    PROVIDE(_init_device=DEFINED(_crt0_init_device) ? _crt0_init_device : _crt0_init_dummy) ;
...
    PROVIDE (__END_text = .) ;
} > irom
...
.rodata :
{
    PROVIDE (__START_rodata = .) ;
    *(EXCLUDE_FILE (*crt0.o *ObjectFileName.o) .rodata)
    *(.rodata.*)
    PROVIDE (__END_rodata = .) ;
} > irom
```

RAM上で実行させたいファイルをtextセクション
(ROMエリア) から除外する
なお、“*(.text)”は、削除する。

RAM上で実行させたいファイルをrodataセクション
(ROMエリア) から除外する。

プログラムをRAM上で実行させる方法

....

.ram_text :

```
{  
  __START_ram_text = . ;  
  *ObjectFileName.o(.text);  
  __END_ram_text = . ;  
} > iram AT > irom
```

RAM上で実行させたいファイルのtextをram_textセクション（新設）の__START_ram_text～__END_ram_textに配置する。

.ram_rodata :

```
{  
  __START_ram_rodata = . ;  
  *ObjectFileName.o(.rodata);  
  __END_ram_rodata = . ;  
} > iram AT > irom
```

RAM上で実行させたいファイルのrodataをram_rodataセクション（新設）の__START_ram_rodata～__END_ram_rodataに配置する。

```
__START_ram_text_lma = LOADADDR(.ram_text);  
__START_ram_rodata_lma = LOADADDR(.ram_rodata);
```

初期のROMエリアに配置されるram_text, ram_rodataセクションの先頭アドレスをシンボルで定義する。

2)スタートアップルーチンの編集

スタートアップ時、ROM上にある“ObjectFileName.o”をRAMにコピーしますので、スタートアップルーチンの“boot.c”を編集し、各シンボルの定義とコピーする関数（memcpy）を追加します。

```
...  
#include <string.h>  
...  
  
// RAM text  
extern char __START_ram_text[];  
extern char __START_ram_text_lma[];  
extern char __END_ram_text[];  
// RAM rodata  
extern char __START_ram_rodata[];  
extern char __START_ram_rodata_lma[];  
extern char __END_ram_rodata[];  
  
static void prepare_ram(void) {  
    memcpy(__START_ram_text, __START_ram_text_lma, __END_ram_text - __START_ram_text);  
    memcpy(__START_ram_rodata, __START_ram_rodata_lma,  
           __END_ram_rodata - __START_ram_rodata);  
}
```

プログラムをRAM上で実行させる方法

```
*****
/*****
* _init_device function.
*
* @brief   Initialize device module.
*****
**/
void _init_device(void)
{
    /// Initialize module.
    prepare_ram();
    return;
}
```

上記の”_init_device()”関数は、スタートアップルーチン(“crt0.c”)内に記述されている関数で、起動直後に呼び出されます。また、本関数内の”prepare_ram()”関数により、初期値としてROM内に置かれた”ObjectFileName.o”のオブジェクトは、RAM上にコピーされ、実行時はRAM上のオブジェクトが呼び出されます。