How to run a program in RAM

You can run the program in RAM by editing the linker script and startup file.

1) Linker script editing

Edit the red and blue parts as shown below. In addition, the part in blue describes the name of the object file you want to place in RAM.

```
/* 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
```

Exclude the files you want to run on RAM from the text section (ROM area). In addition, "* (.text)" is deleted.

Exclude the files you want to run on RAM from the rodata section (ROM area).

How to run a program in RAM

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

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

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

Place the text of the file you want to execute on RAM in "__START_ram_text” to “__END_ram_text" in the “ram_text section” (new).

Place the rodata of the file you want to execute on RAM in "__START_ram_rodata” ~ “__END_ram_rodata" in the “ram_rodata” section (new).

Define the start address of the ram_text and ram_rodata sections placed in the initial ROM area with symbols.

How to run a program in RAM

2) Edit startup routine

At startup, “ObjectFileName.o” on the ROM is copied to RAM, so edit the startup routine “boot.c” and add the definition of each symbol and the function (memcpy) to copy.

```
…
#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);
}
```

How to run a program in RAM

```
......
/*******************************************************************************
****
 * _init_device function.
 *
 * @brief     Initialize device module.

********************************************************************************
**/
void _init_device(void)
{
                /// Initialize module.
                prepare_ram();
                return;
}
```

The above "_init_device()” is a function described in the startup routine (" crt0.c ") and is called immediately after startup. Also, the initial value is set by the "prepare_ram()“ in this function. The object of "ObjectFileName.o" placed in ROM as is copied to RAM, and the object in RAM is called at runtime.