

How to place a variable at a specific address

Normally, the placement of variables is decided by the compiler, but if it is a global variable, it is possible to specify the placement order and address for each section. Edit the linker script as shown below and adapt the source code to this linker script.

For example, if you want to assign an array called unsigned char test [16] to address 0x100, make the linker script and source code as follows.

<Example of linker script>

```
SECTIONS
```

```
{
```

```
    .var_ram_section1 (0x100) : ← Define RAM allocation address
```

```
    {
```

```
        *(.var_ram_section1); ←Define any section name
```

```
        . = ALIGN(0x10); ←Reserve a 16-byte area.
```

```
    } > iram
```

```
        :           :           :           :
```

<Example of source code>

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

← Section name defined in the script file

How to place a variable at a specific address

However, in this case, the code is also generated for the RAM area, and an error occurs in the process of generating the psa file at Build, but this is not a problem.

How to place a variable at a specific address

Also, if you want to place all the variables defined in the source file at a specific address, you can place them at a specific address simply by defining a linker script.

For example, if you want to place all the variables defined in “variables.c” from address 0x200, you can write as follows.

<Example of linker script>

```
MEMORY
```

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

←Define RAM allocation address

```
SECTIONS
```

```
{          .bss (NOLOAD) :
  {          PROVIDE (__START_bss = .) ;
            *(EXCLUDE_FILE (*variables.o) .bss) ← Exclude variables in variables.c
            PROVIDE (__END_bss = .) ;
  } > iram
```

```
.var_ram_section2 (NOLOAD) : ←Define any section name
{          PROVIDE (__START_var_ram_section2 = .) ;
            KEEP(*variables.o(.bss))
            PROVIDE (__END_var_ram_section2 = .) ;
} > vram
```

```
          :          :          :          :
```