

Relocating Code and Data Using the MCUXpresso IDE

This document provides some sample codes and steps for relocating Code and Data into RAM/FLASH. Please look at them with projects in attachment. It is based on "MCUXpresso_IDE_User_Guide.pdf", so about the understanding of some macro, you can check on that document.

This document is based on FRDM-K64, MCUXpressoIDE_10.0.2_411. You can use it on other chips too.

1. Introduce the "Memory configuration editor"

2.1. Relocating Data into RAM

2.2. Relocating Code into RAM

2.3. Relocating Code into Flash

3. Reference

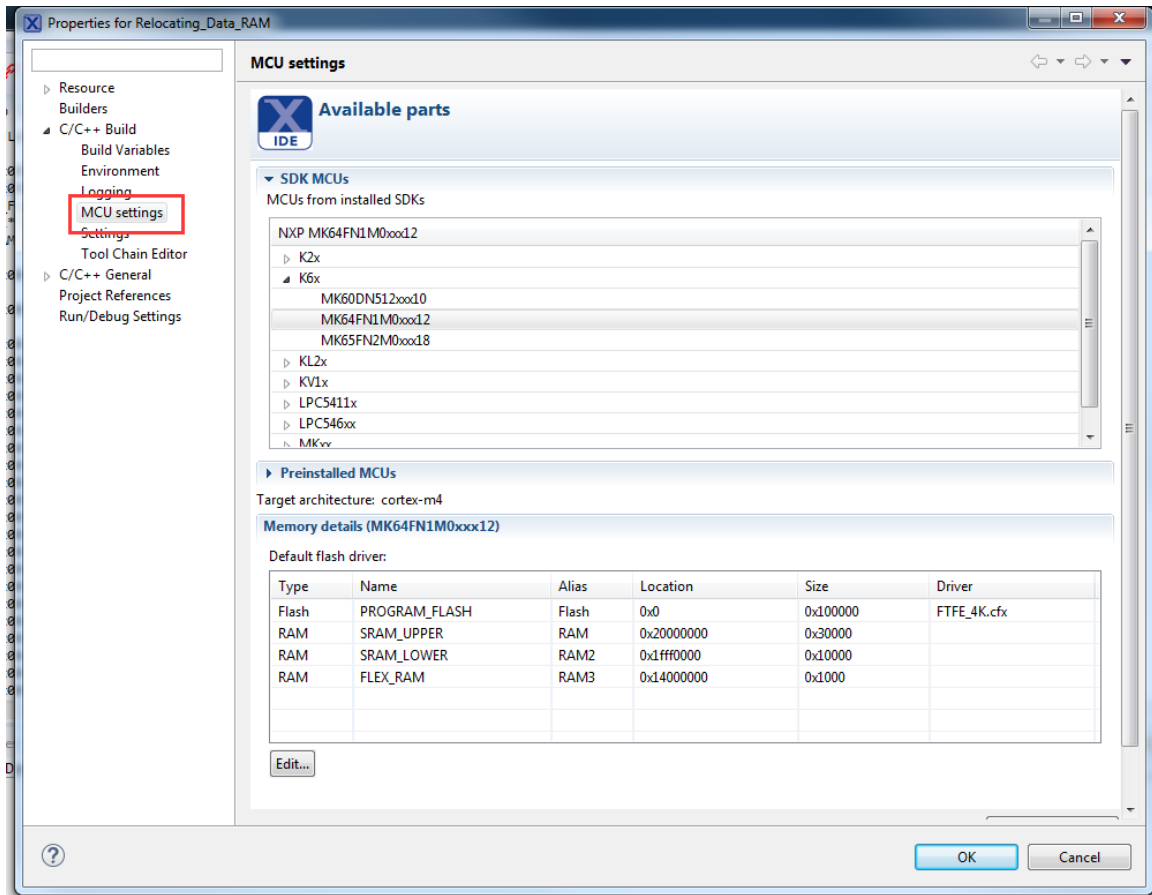
1. Introduce the "Memory configuration editor"

The "Memory configuration editor" lists the memory configuration of selected MCU, we can also modify the memory region refer to our requirement on it.

How to find it ?

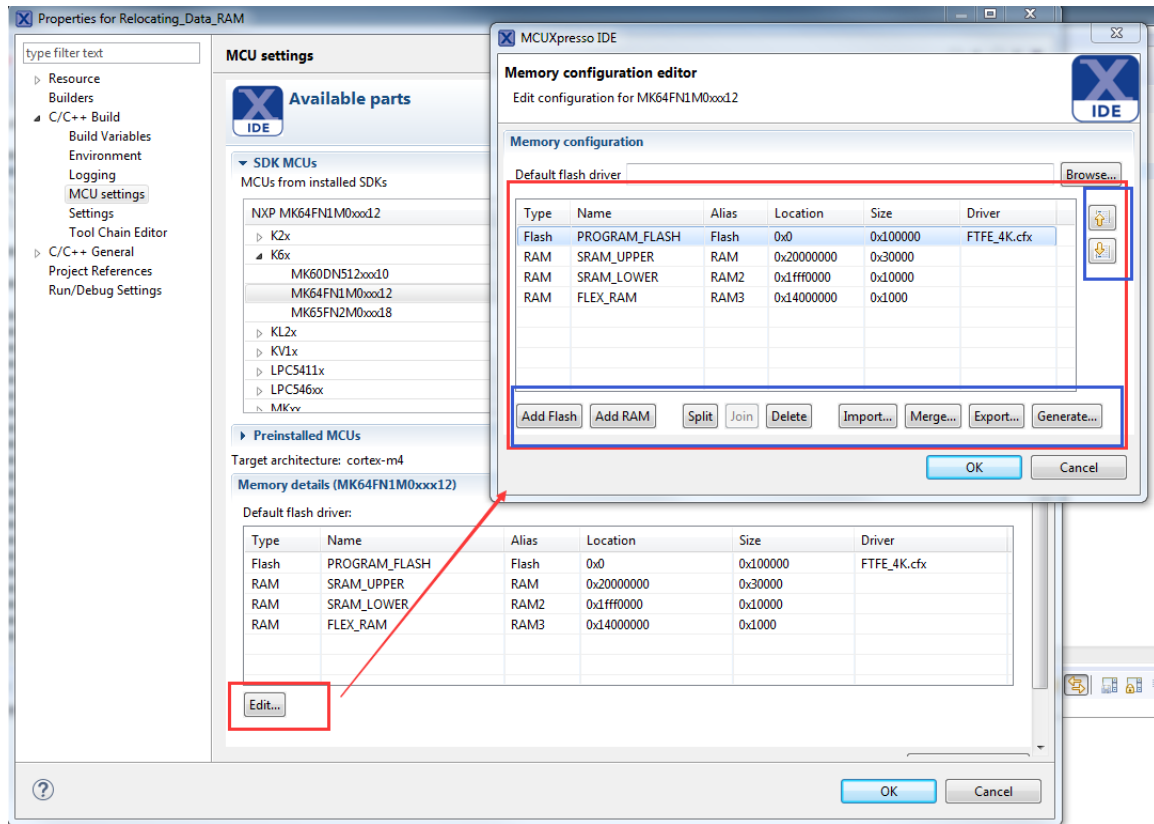
Right click on your project-> Properties -> C/C++ Build -> MCU settings

Below is the default configuration for MK64FN1M0xxx12 chip.



How to change it ?

Click the Edit button, it shows another view, we can configure the memory sections:



About the memory editor controls, there is a detail introduction in "MCUXpresso_IDE_User_Guide.pdf" -> Table 12.1. Memory editor controls

2.1. Relocating Data into RAM

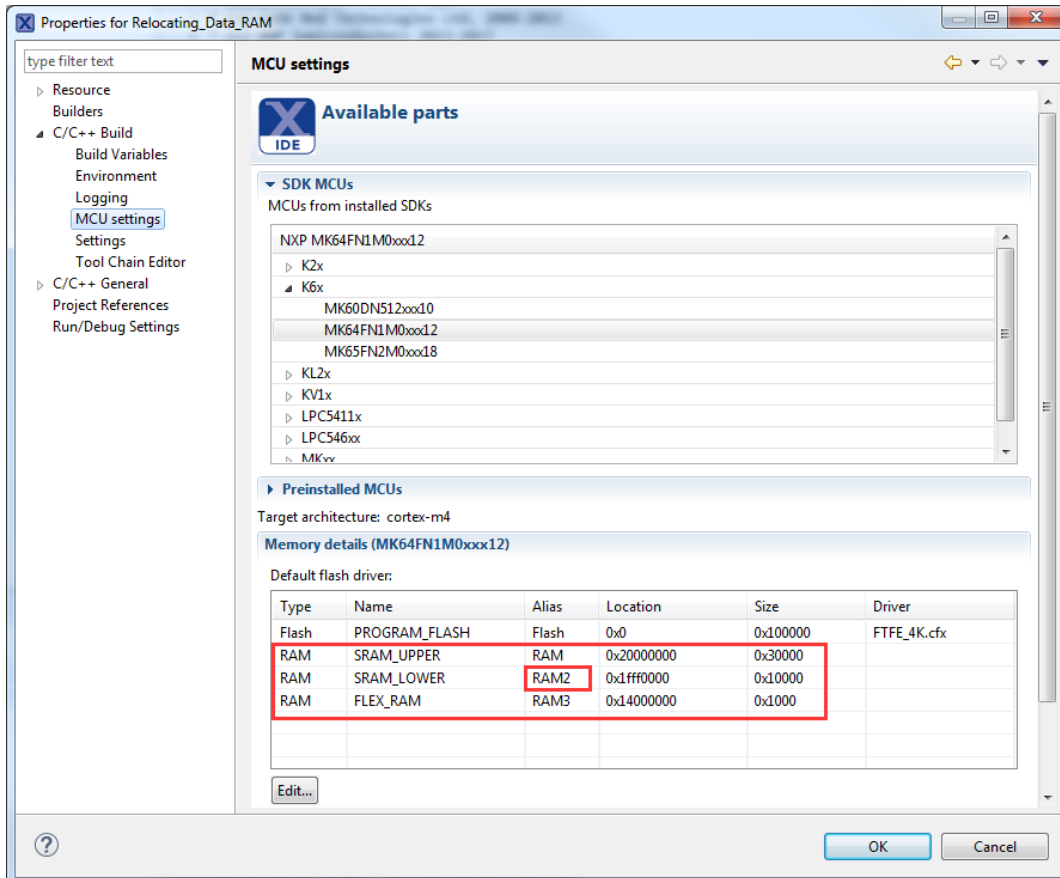
By default the managed linker script mechanism will place all of the application data into the first bank of RAM. Now we relocate the array "data_buffer[]" into the second RAM(RAM2), only need two steps:

- Includes the header file which includes macros we are using.
- Relocate Data into RAM2.

```
#include <cr_section_macros.h>
```

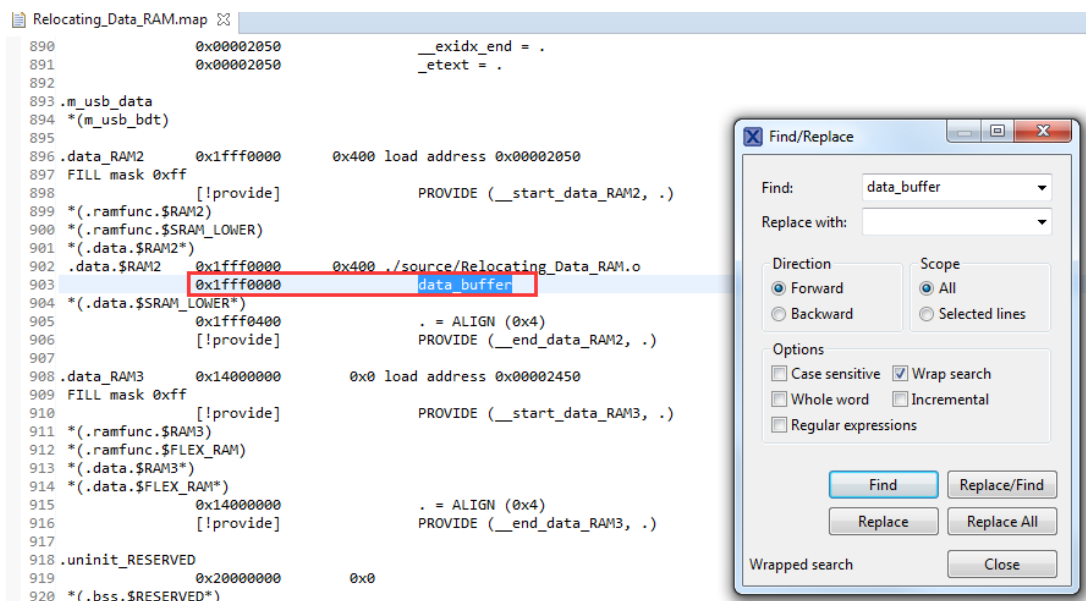
```
__DATA(RAM2) char data_buffer[1024];
```

We can check the RAM2 in "MCU setting":



The related demo is “Relocating_Data_RAM”.

Go to menu Project > Build Project and then search for the *.map file inside {Project_path}/Debug. Here you can see that variable is placed in RAM2- 0x1fff0000, the length is 0x400.



2.2. Relocating Code into RAM

For example, place the function “test_function()” to RAM2, only need two steps:

- Includes the header file which includes macros we are using.
- Relocate function into RAM2.

```
#include <cr_section_macros.h>
```

```
_RAMFUNC(RAM2) void test_function();
```

The related project is “Relocating_Code_RAM”.

Go to menu Project > Build Project and then search for the *.map file inside {Project_path}/Debug. Here you can see that function “test_function()” is placed in RAM2- 0x1fff000.

```
Relocating_Code_RAM.map ⌘
893
894 .m_usb_data
895 *(m_usb_bdt)
896
897 .data_RAM2      0x1fff0000      0x20 load address 0x0000202c
898 FILL mask 0xff
899               [!provide]      PROVIDE (__start_data_RAM2, .)
900 *(.ramfunc.$RAM2)
901 .ramfunc.$RAM2
902               0x1fff0000      0x14 ./source/Relocating_Code_RAM.o
903               0x1fff0000      test function
904 *fill*         0x1fff0014      0x4 ff
905 .ramfunc.$RAM2.__stub
906               0x1fff0018      0x8 linker stubs
907 *(.ramfunc.$SRAM_LOWER)
908 *(.data.$RAM2*)
909 *(.data.$SRAM_LOWER*)
910               0x1fff0020      . = ALIGN (0x4)
911               [!provide]      PROVIDE (__end_data_RAM2, .)
912
```

2.3. Relocating Code into Flash

For example, place the function “test_functon ()” to Flash2, only need two steps:

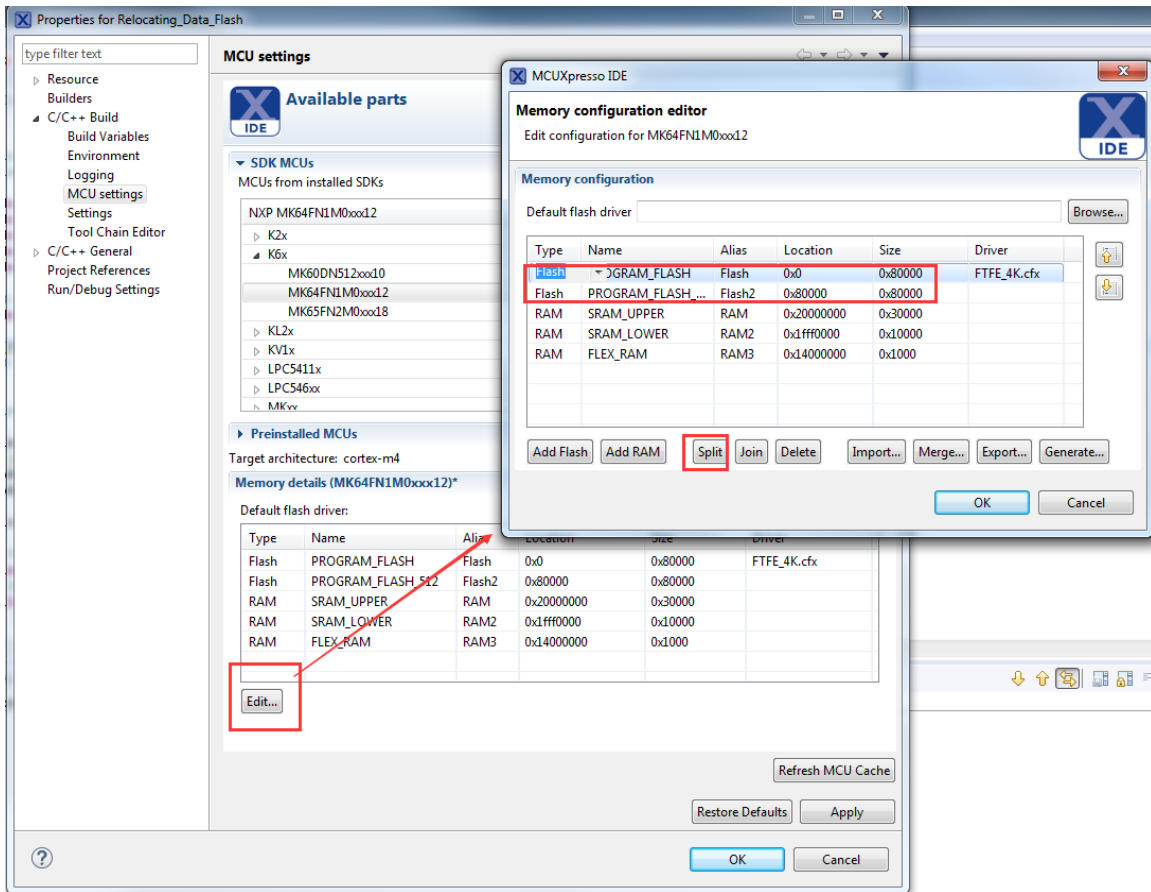
- Includes the header file which includes macros we are using.
- Relocate function into Flash2.

The related project is “Relocating_Code_Flash”.

Go to menu Project > Build Project and then search for the *.map file inside {Project_path}/Debug. Here you can see that variable is placed in Flash2 - 0x80000.

```
Relocating_Code_Flash.map
356 *(.text_PROGRAM_FLASH_512*)
357 *(.text.$Flash2*)
358 .text.$Flash2 0x00080000 0x14 ./source/Relocating_Code_Flash.o
359 0x00080000 test_function
360 *(.text.$PROGRAM_FLASH_512*)
```

The Flash2 is configured at “Properties->C/C++ Build->MCU settings”



3. Reference

“MCUXpresso_IDE_User_Guide.pdf”

“Relocating Code and Data Using the MCUXpresso IDE.pdf”