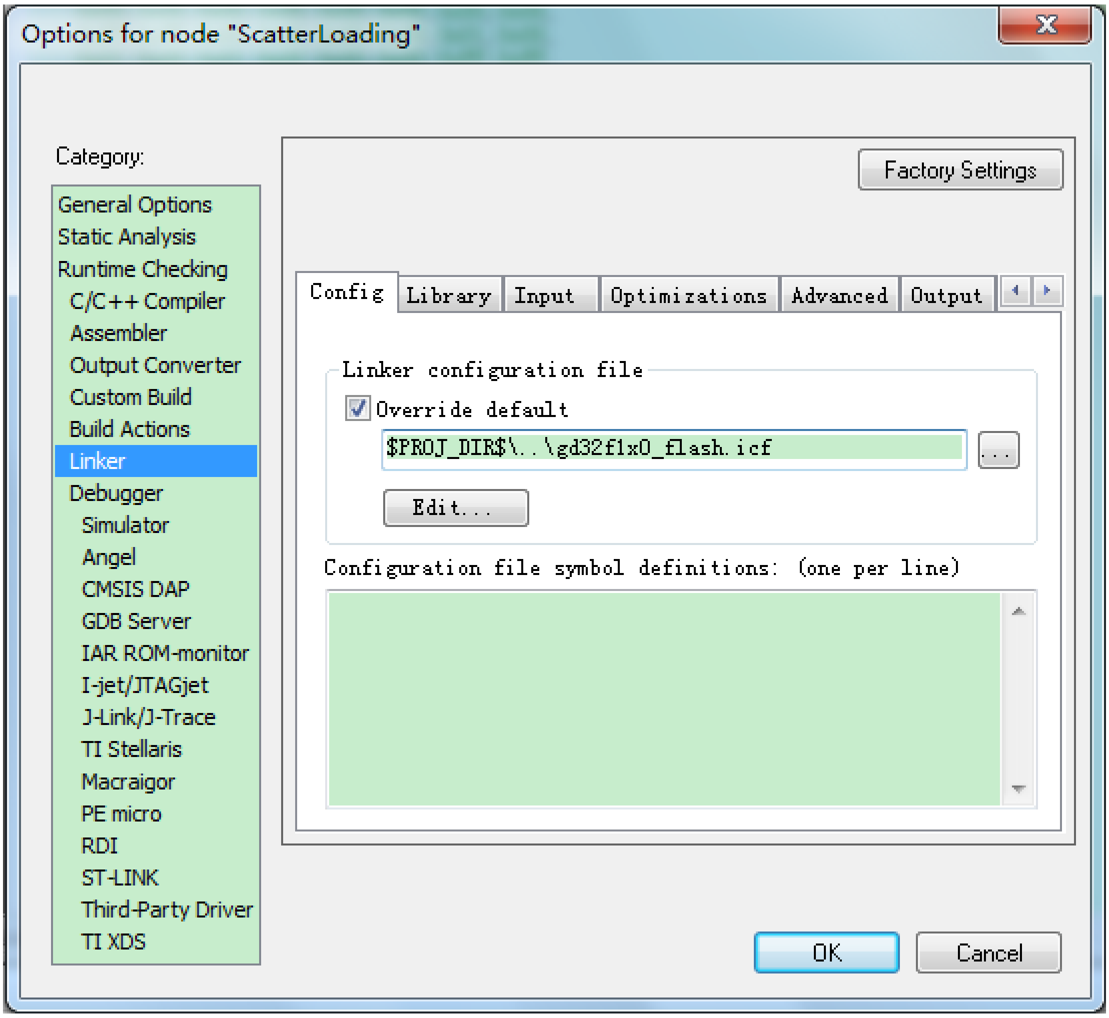
1. **Specify the.C file to the flash specified location**

   This project will load the hw_config.c file at the address 0x08002000

1、 IAR configration：



Options for node "ScatterLoading"

Linker configuration file

☑ Override default

$PROJ_DIR$\..\gd32f1x0_flash.icf

Configuration file symbol definitions: (one per line)

In the linker config, select the Override default, and choose our own .icf file.

In this way, the IAR compiler will link to the gd32f1x0_flash.icf file

Open the gd32f1x0_flash.icf file in the IAR project, modify as follows:

```
/*###ICF### Section handled by ICF editor, don't touch! ****/
/*-Editor annotation file-*/
/* IcfEditorFile="$TOOLKIT_DIR$\config\ide\IcfEditor\cortex_v1_0.xml" */
/*-Specials-*/
define symbol __ICFEDIT_intvec_start__ = 0x08000000;
/*-Memory Regions-*/
define symbol __ICFEDIT_region_ROM_start__   = 0x08000000;
define symbol __ICFEDIT_region_ROM_end__     = 0x0800FFFF;
define symbol __ICFEDIT_region_ROM1_start__ = 0x08002000;
define symbol __ICFEDIT_region_ROM1_end__    = 0x0800FFFF;
define symbol __ICFEDIT_region_RAM_start__   = 0x20000000;
define symbol __ICFEDIT_region_RAM_end__        = 0x20017FFF;
/*-Sizes-*/
define symbol __ICFEDIT_size_cstack__ = 0x400;
define symbol __ICFEDIT_size_heap__     = 0x200;
/**** End of ICF editor section. ###ICF###*/


define memory mem with size = 4G;
define region ROM_region   = mem:[from __ICFEDIT_region_ROM_start__   to __ICFEDIT_region_ROM_end__];
define region ROM1_region   = mem:[from __ICFEDIT_region_ROM1_start__   to __ICFEDIT_region_ROM1_end__];
define region RAM_region   = mem:[from __ICFEDIT_region_RAM_start__   to __ICFEDIT_region_RAM_end__];

define block CSTACK    with alignment = 8, size = __ICFEDIT_size_cstack__   { };
define block HEAP      with alignment = 8, size = __ICFEDIT_size_heap__     { };

initialize by copy { readwrite };
do not initialize  { section .noinit };

place at address mem:__ICFEDIT_intvec_start__ { readonly section .intvec };
/* load the hw_config.o file at the address 0x08002000*/
place at address mem:0x08020000 { section .text   object hw_config.o };
/*load section .funflash at the address 0x08002000*/
place at address mem:0x0800F000 { readonly section .funflash};

place in ROM_region   { readonly };
place in RAM_region   { readwrite,
                        block CSTACK, block HEAP };
```

The red part is what I add, the other part should be the same as your original file.

2. The function is loaded to the flash specified location

In this project, the function delay in main.c is loaded in the starting position of 0x0800F000.

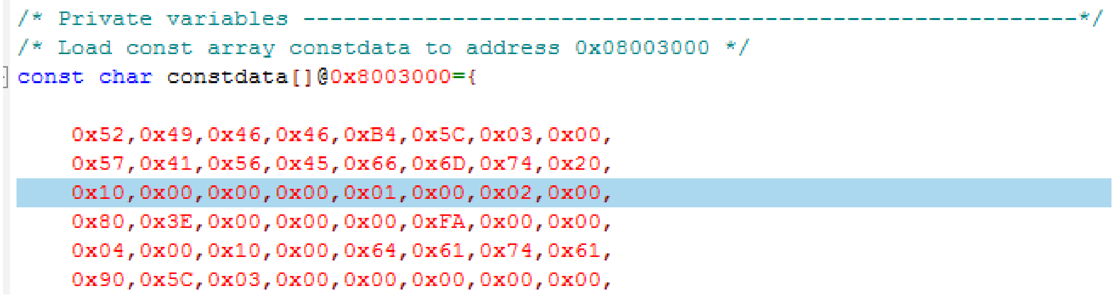
(1) Add “place at address mem: 0x0800F000 {readonly section .funflash};” in gd32f1x0_flash.icf file

(2) In the definition of the function add “@. funflash "

```
void delay(void) @".funflash"
{
   for(i=0;i<0xffff;i++);
}
```

3. Load an const array into the specified location

In this project, the constdata (const can not be omitted) is loaded to 0x08003000

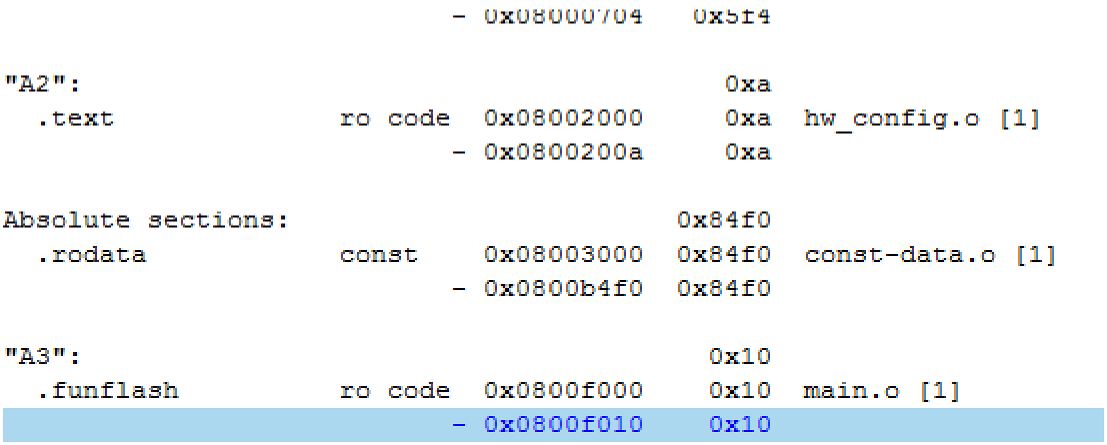


```
/* Private variables ---------------------------------------------------------*/
/* Load const array constdata to address 0x08003000 */
const char constdata[]@0x8003000={

    0x52,0x49,0x46,0x46,0xB4,0x5C,0x03,0x00,
    0x57,0x41,0x56,0x45,0x66,0x6D,0x74,0x20,
    0x10,0x00,0x00,0x00,0x01,0x00,0x02,0x00,
    0x80,0x3E,0x00,0x00,0x00,0xFA,0x00,0x00,
    0x04,0x00,0x10,0x00,0x64,0x61,0x74,0x61,
    0x90,0x5C,0x03,0x00,0x00,0x00,0x00,0x00,
```

4. The results as follows

In the project directory \Debug\List, find the ScatterLoading.map file, open, as follows:



```
                                        - 0x08000704    0x5f4

"A2":                                                     0xa
  .text                   ro code   0x08002000      0xa   hw_config.o [1]
                                  - 0x0800200a      0xa

Absolute sections:                                0x84f0
  .rodata                 const     0x08003000    0x84f0  const-data.o [1]
                                  - 0x0800b4f0    0x84f0

"A3":                                               0x10
  .funflash               ro code   0x0800f000      0x10  main.o [1]
                                  - 0x0800f010      0x10
```