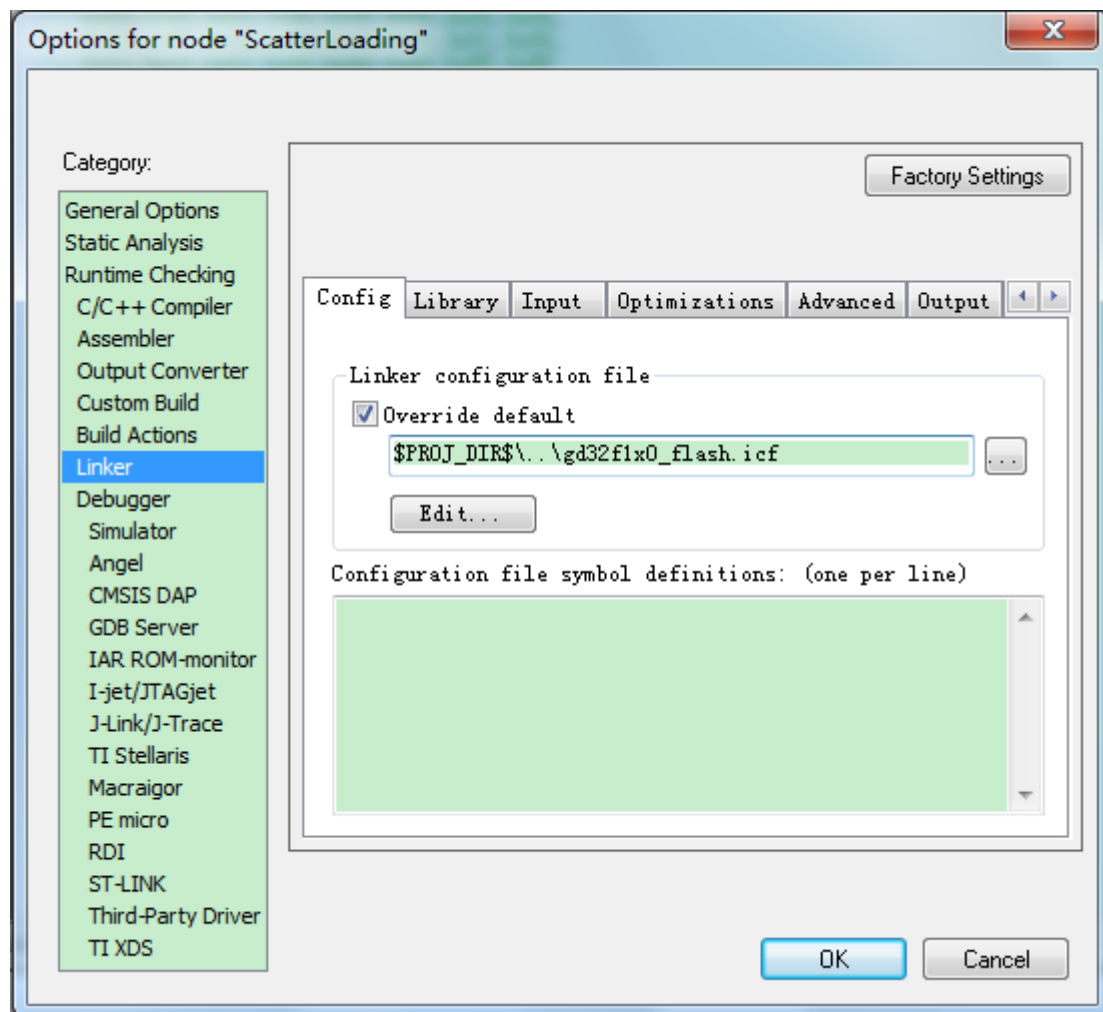


一、指定.c 文件到 flash 指定位置

本工程指定的是 hw_config.c 文件加载在地址 0x08002000 之后

1、IAR 配置：



Linker Config 标签页下 选中 Override default ，并且选择自己工程中的 icf 文件，如下：

这样，IAR 编译完链接的时候会链接 gd32f1x0_flash.icf 文件，我们要对这个文件稍作修改。

在 IAR 工程中打开 gd32f1x0_flash.icf 文件，修改如下：

```
/*###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;
```

定义一个 8K 以后的区域，为 ROM1

```

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__ { };

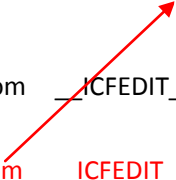
initialize by copy { readwrite };
do not initialize { section .noinit };

place at address mem:__ICFEDIT_intvec_start__ { readonly section .intvec };
place at address mem:0x08020000 { section .text object hw_config.o };// 可以实现将
hw_config.o 放在 0x8002000 起始位置
place at address mem:0x0800F000 { readonly section .funflash}; //可以实现将指定函数放在
0x0800F000 起始位置

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

```

定义 ROM1 的起始地址



红色部分是我添加的，其他部分应该跟你原来的文件中一样。

二、 将函数加载到 flash 指定位置

本工程中是将 main.c 中的函数 delay 加载在 0x0800F000 起始位置

1、在 gd32f1x0_flash.icf 文件中加入

```
place at address mem:0x0800F000 { readonly section .funflash};
```

2、在函数的定义出添加 @".funflash"

```

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

```

三、 将数组加载到指定的位置

本工程中是将 constdata(const 不可省略)加载到 0x08003000 起始位置

```
/*将只读数组constdata移到0x08003000以后的区域*/  
const char constdata[] @0x08003000 = {  
    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,  
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00  
};
```

效果如下：

在工程目录\Debug\List 下找到 ScatterLoading.map 文件，打开，如下：

```
                - 0x08000704      0x514  
  
"A2":  
    .text                ro code  0x08002000      0xa    hw_config.o [1]  
                        - 0x0800200a      0xa  
  
Absolute sections:  
    .rodata              const    0x08003000      0x84f0  const-data.o [1]  
                        - 0x0800b4f0      0x84f0  
  
"A3":  
    .funflash            ro code  0x0800f000      0x10    main.o [1]  
                        - 0x0800f010      0x10
```