GigaDevice Semiconductor Inc.

基于 GD32F30x 系列的 IEC60730 ClassB 库移植指南

应用笔记

AN136

1.0 版本

（2023 年 9 月）

目录

图索引

表索引

1. 简介

GD32 MCU 提供 IEC60730 ClassB 认证库支持，对于每个系列的 GD32 MCU，都会提供常用系列的工程模板，用户在对同一系列不同封装的芯片进行 IEC60730 自检认证时，可通过移植模板程序来匹配所需检测的芯片即可。本应用笔记将详细介绍在 GD32F30x 系列芯片移植过程中需要修改的注意事项，帮助客户移植 IEC60730 ClassB 认证库。

2. IEC60730 ClassB 认证库移植

GD32 MCU提供IAR、Keil、eclipse环境下的IEC60730 ClassB认证库支持，三种开发环境下模板工程所需配置的参数，以及修改的部分有所区别，下面将从三种开发环境展开说明。

本应用笔记基于GD32F305RC进行工程移植说明，对于其他M3/M4核芯片认证库移植同样适用。

2.1. IAR 环境认证库工程移植

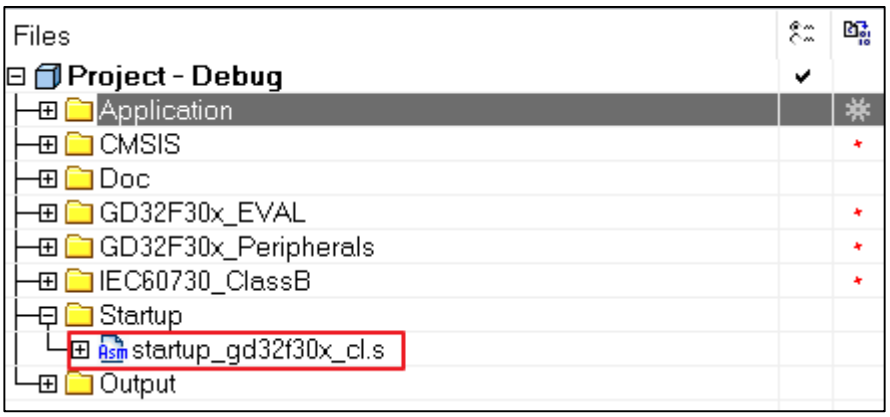
1、根据芯片数据手册修改gd32f30x_test.h文件中编译宏__IAR_SYSTEMS_ICC__下的RAM边界，如*图2-1. 修改gd32f30x_test.h中RAM边界*所示。

图 2-1. 修改 gd32f30x_test.h 中 RAM 边界

```
#ifdef __IAR_SYSTEMS_ICC__

/* used in RAM test during run time */
__no_init EXTERN uint32_t buffer_ram_run[RAMRUN_BLOCK_SIZE] @ "RAM_RUN_BUF";
/* used as RAM pointer during run time */
__no_init EXTERN uint32_t *ptr_ram_run                      @ "RAM_RUN_PTR";
/* used for program counter test */
__no_init EXTERN uint32_t (*test_pc_func[6])(void)          @ "IEC_TEST_RAM";
/* used in main program and increased in SysTick timer ISR */
__no_init EXTERN uint32_t systick_count                     @ "IEC_TEST_RAM";
/* flag which indicate a specified tick comes */
__no_init EXTERN FlagStatus test_interrupt_flag             @ "IEC_TEST_RAM";
/* pointer to FLASH for crc16 test in run time */
__no_init EXTERN uint8_t *ptr_crc16_run                     @ "IEC_TEST_RAM";
/* pointer to FLASH for crc32 tests in run time */
__no_init EXTERN uint32_t *ptr_crc32_run                    @ "IEC_TEST_RAM";
/* 32-bit CRC values in run time */
__no_init EXTERN uint32_t crc32_value                       @ "IEC_TEST_RAM";
/* 16-bit CRC values in run time */
__no_init EXTERN uint16_t crc16_value                       @ "IEC_TEST_RAM";
/* buffer used for stack overflow test */
__no_init EXTERN volatile uint32_t buffer_stack_overflow[6] @ "STACK_OV_TEST";

extern uint32_t __ICFEDIT_region_ROM_start__;
extern uint32_t __ICFEDIT_region_ROM_end__;
extern uint32_t __ICFEDIT_region_RAM_start__;
extern uint32_t __ICFEDIT_region_RAM_end__;
extern uint32_t __ICFEDIT_region_IECTEST_PARAM_start__;
extern uint32_t __ICFEDIT_region_IECTEST_PARAM_end__;

#define FLASH_START             (unsigned char *)&__ICFEDIT_region_ROM_start__
#define FLASH_SIZE              (unsigned int)&__ICFEDIT_region_ROM_end__ - 2 -(unsigned int)&__ICFEDIT_region_ROM_start__ + 1 /* FLASH_SIZE in byte */
#define FLASH_SIZE_WORDS        ((uint32_t)&__ICFEDIT_region_ROM_end__ -2  -(uint32_t)&__ICFEDIT_region_ROM_start__ + 1)/4 /* FLASH_SIZE in words */
#define FLASH_END               ((uint8_t *)(&__ICFEDIT_region_ROM_end__))

#define FLASH_BLOCK_SIZE        ((uint32_t)1024uL)
#define FLASH_BLOCKNUM          (uint32_t)((FLASH_SIZE+2) / FLASH_BLOCK_SIZE)
#define FLASH_BLOCKNUM_WORDS    ((uint32_t)(FLASH_SIZE_WORDS / FLASH_BLOCK_SIZE))

#define RAM_START               (uint32_t *)0x20000000
#define RAM_END                 (uint32_t *)0x20017F40  改为当前芯片的RAM大小减去0xC0

#define IEC_TEST_PARAM_START    ((uint32_t *)(&__ICFEDIT_region_IECTEST_PARAM_start__))
#define IEC_TEST_PARAM_END      ((uint32_t *)(&__ICFEDIT_region_IECTEST_PARAM_end__))

extern void __iar_program_start(void);
extern void Reset_Handler(void);
#define DefaultSystemStartUp() Reset_Handler()
```

2、检查工程目录Startup文件夹下的.s启动文件，是否符合当前芯片类型，如*图2-2. 启动文件检查*所示。

图 2-2. 启动文件检查



若符合，则无需修改；若不符合，则需要在固件库文件夹GD32F30x_Firmware_Library\CMSIS\GD\GD32F30x\Source\ARM下，重新选择符合当前芯片类型(cl/hd/xd）的.s启动文件，并如*图2-3. 修改.s启动文件*所示进行修改，使芯片在运行之前进行自检测。

图 2-3. 修改.s 启动文件

```
        MODULE  ?cstartup

        ;; Forward declaration of sections.
        SECTION CSTACK:DATA:NOROOT(3)

        SECTION .intvec:CODE:NOROOT(2)

        EXTERN  test_prerun

        EXTERN  __iar_program_start
        EXTERN  SystemInit
        PUBLIC  __vector_table

        DATA
__vector_table
        DCD     sfe(CSTACK)                 ; top of stack
        DCD     test_prerun                 ; Reset Handler --> test_prerun

        DCD     NMI_Handler                 ; Vector Number 2,NMI Handler
        DCD     HardFault_Handler           ; Vector Number 3,Hard Fault Handler
        DCD     MemManage_Handler           ; Vector Number 4,MPU Fault Handler
        DCD     BusFault_Handler            ; Vector Number 5,Bus Fault Handler
        DCD     UsageFault_Handler          ; Vector Number 6,Usage Fault Handler
        DCD     0                           ; Reserved
        DCD     0                           ; Reserved
        DCD     0                           ; Reserved
        DCD     0                           ; Reserved
        DCD     SVC_Handler                 ; Vector Number 11,SVCall Handler
        DCD     DebugMon_Handler            ; Vector Number 12,Debug Monitor Handler
        DCD     0                           ; Reserved
        DCD     PendSV_Handler              ; Vector Number 14,PendSV Handler
        DCD     SysTick_Handler             ; Vector Number 15,SysTick Handler
```

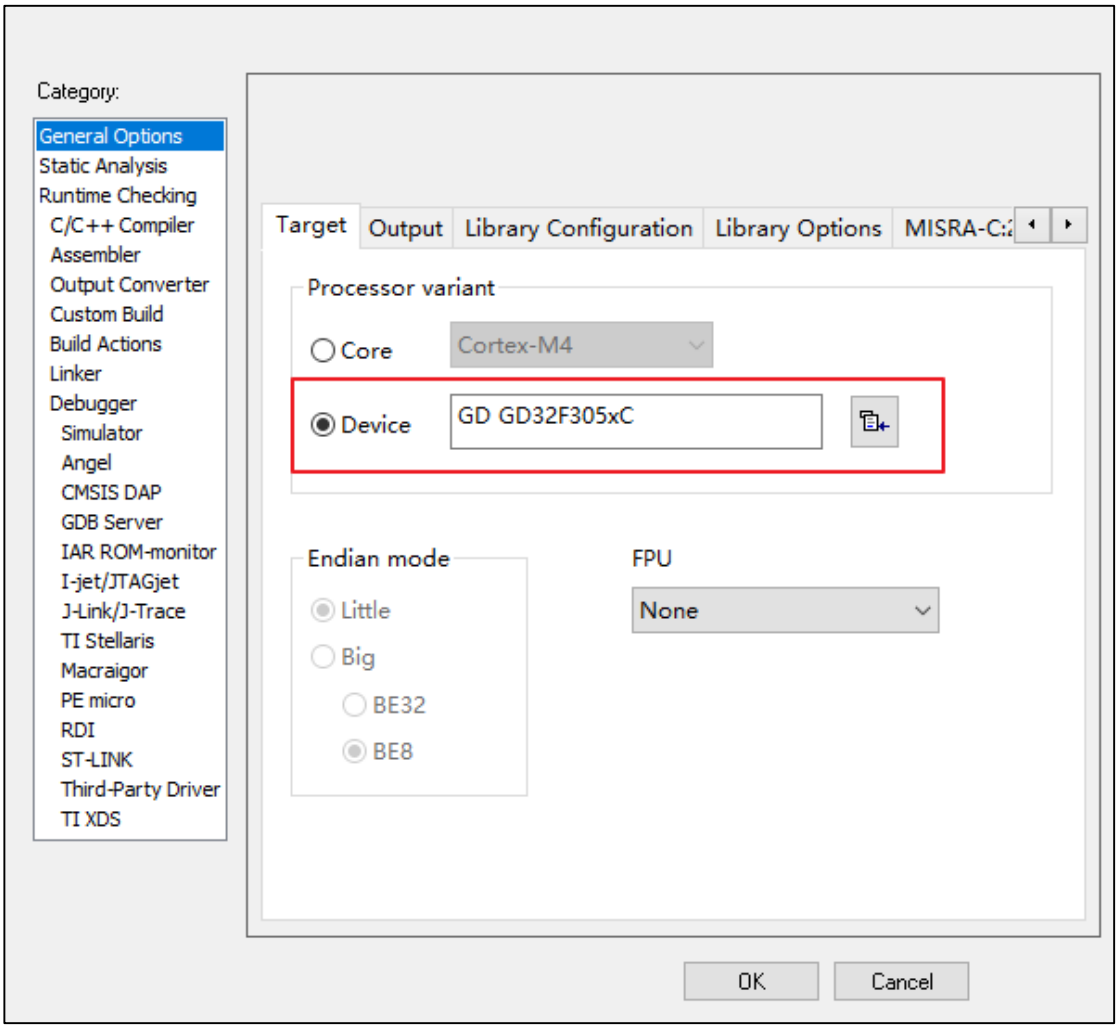
3、修改工程所在文件夹（在本例中为：..\GD32305_IEC_Test\Projects\IEC_Test\EWARM）下的IEC_TEST_BOOT_FLASH.icf文件，如*图2-4. 修改分散加载文件*所示为修改的部分，根据芯片数据手册修改Flash和RAM的大小匹配当前芯片。

图 2-4. 修改分散加载文件

```
/*###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__       = 0x0803FFFF;   修改为Flash大小
define symbol __ICFEDIT_region_RAM_start__     = 0x200000B0;
define symbol __ICFEDIT_region_RAM_end__       = 0x20017FFF;   修改为RAM大小
define symbol __ICFEDIT_region_IECTEST_PARAM_start__ = 0x20000040;
define symbol __ICFEDIT_region_IECTEST_PARAM_end__   = 0x200000B0;
/*-Sizes-*/
```

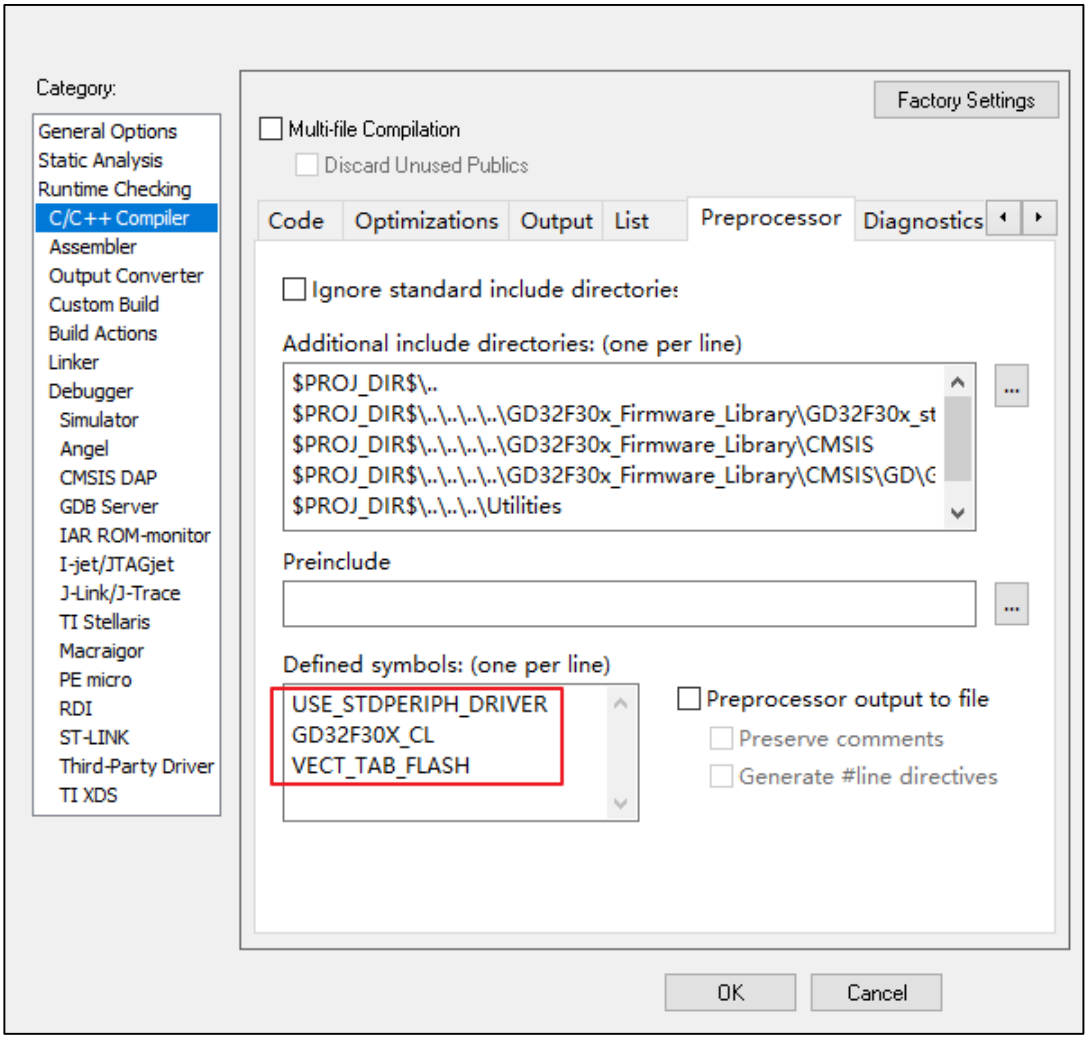
4、修改Options for node ‘Project’中Target的配置，选择当前芯片的型号，如*图2-5. Device配置*所示：

图 2-5. Device 配置



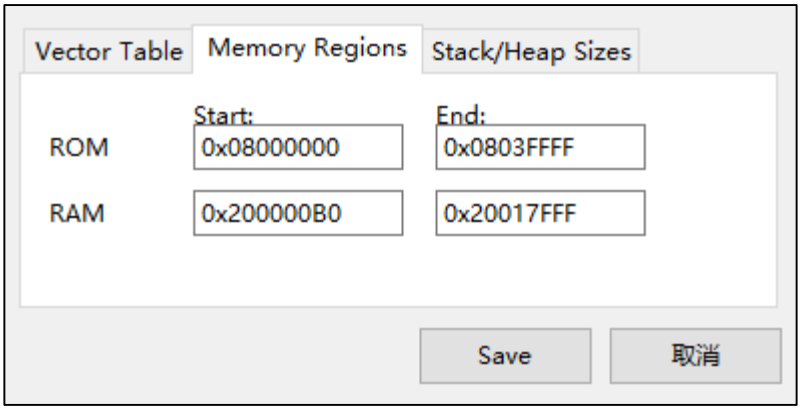
5、在Options for node ‘Project’->C/C++ Complier->Preprocessor中添加当前工程需要的预编译宏，主要修改符合当前芯片类型的预编译宏（在本例中为GD32F30X_CL），如*图2-6. 修改预编译宏*所示：

图 2-6. 修改预编译宏



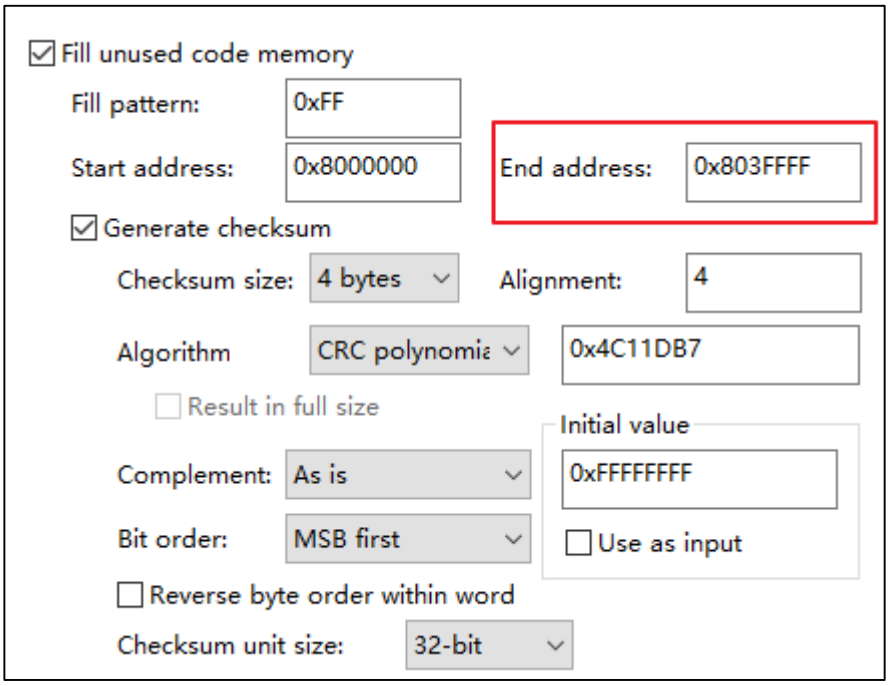
6、点击Options for node ‘Project’->Linker->config->Edit->Memory Regions,根据当前芯片的数据手册修改ROM(Flash)和RAM的边界值，如*图2-7. 修改工程设置中的边界值*所示

图 2-7. 修改工程设置中的边界值



7、在Options for node ‘Project’->Linker->Checksum中，修改End address为Flash大小，如*图2-8. 修改工程属性中的Checksum配置*所示：

图 2-8. 修改工程属性中的 Checksum 配置



2.2. Keil 环境认证库工程移植

1、根据当前芯片型号的数据手册修改gd32f30x_test.h文件中RAM和Flash边界，如*图2-9. 修改gd32f30x_test.h文件中RAM和Flash边界*所示。

图 2-9. 修改 gd32f30x_test.h 文件中 RAM 和 Flash 边界

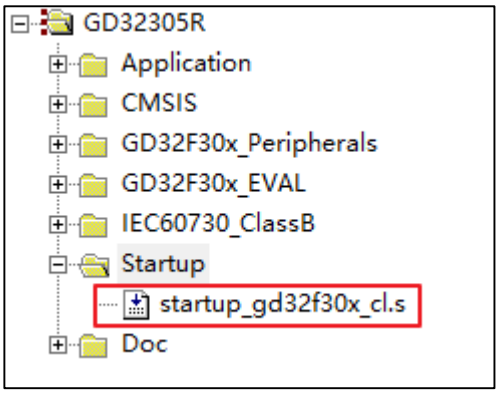
```
#ifdef __CC_ARM

/* used in RAM test during run time */
EXTERN uint32_t buffer_ram_run[RAMRUN_BLOCK_SIZE] __attribute__((section("RAM_RUN_BUF")));
/* used as RAM pointer during run time */
EXTERN uint32_t *ptr_ram_run                      __attribute__((section("RAM_RUN_PTR")));
/* used for program counter test */
EXTERN uint32_t (*test_pc_func[6])(void)          __attribute__((section("IEC_TEST_RAM"), zero_init));
/* used in main program and increased in SysTick timer ISR */
EXTERN uint32_t systick_count                     __attribute__((section("IEC_TEST_RAM"), zero_init));
/* flag which indicate a specified tick comes */
EXTERN FlagStatus test_interrupt_flag             __attribute__((section("IEC_TEST_RAM"), zero_init));
/* pointer to FLASH for crc16 test in run time */
EXTERN uint8_t *ptr_crc16_run                     __attribute__((section("IEC_TEST_RAM"), zero_init));
/* pointer to FLASH for crc32 tests in run time */
EXTERN uint32_t *ptr_crc32_run                    __attribute__((section("IEC_TEST_RAM"), zero_init));
/* 32-bit CRC values in run time */
EXTERN uint32_t crc32_value                       __attribute__((section("IEC_TEST_RAM"), zero_init));
/* 16-bit CRC values in run time */
EXTERN uint16_t crc16_value                       __attribute__((section("IEC_TEST_RAM"), zero_init));
/* buffer used for stack overflow test */
EXTERN volatile uint32_t buffer_stack_overflow[6] __attribute__((section("STACK_OV_TEST"), zero_init));

#define FLASH_START          ((uint8_t *)0x08000000)
#define FLASH_SIZE           ((uint32_t)0x00040000 - 4)
#define FLASH_SIZE_WORDS     (uint32_t)(((uint32_t)FLASH_END-(uint32_t)FLASH_START)/4)
#define FLASH_END            ((uint8_t *)0x0803FFFF)

#define FLASH_BLOCK_SIZE     ((uint32_t)1024uL)
#define FLASH_BLOCKNUM       (uint32_t)(((uint32_t)FLASH_END-(uint32_t)FLASH_START+1) / FLASH_BLOCK_SIZE)
#define FLASH_BLOCKNUM_WORDS (uint32_t)((FLASH_SIZE_WORDS) / FLASH_BLOCK_SIZE)

#define IEC_TEST_PARAM_START ((uint32_t *)0x20000040)
#define IEC_TEST_PARAM_END   ((uint32_t *)0x200000B0)

#define RAM_START            (uint32_t *)0x20000000
#define RAM_END              (uint32_t *)0x20017F40
```

改为当前芯片的Flash大小

改为当前芯片的RAM大小减去0xC0

2、检查工程目录下Starup文件夹下的.s启动文件，是否符合当前芯片类型，如*图2-10. .s启动文件检查*所示：

图 2-10. .s 启动文件检查



若启动文件适用当前芯片型号，则无需修改；否则需要在固件库文件夹GD32F30x_Firmware_Library\CMSIS\GD\GD32F30x\Source\ARM下，重新选择适用当前芯片型号的.s启动文件，并做如下代码中红色标注所示修改：

```
Stack_Size      EQU     0x00000400

                AREA    STACK, NOINIT, READWRITE, ALIGN=3
Stack_Mem       SPACE   Stack_Size
__initial_sp
```

```
; <h> Heap Configuration
;     <o>    Heap Size (in Bytes) <0x0-0xFFFFFFFF:8>
; </h>

Heap_Size          EQU       0x00000400

                   AREA      HEAP, NOINIT, READWRITE, ALIGN=3
__heap_base
Heap_Mem           SPACE     Heap_Size
__heap_limit

          IMPORT   test_prerun

                   PRESERVE8
                   THUMB

;                  /* reset Vector Mapped to at Address 0 */
                   AREA      RESET, DATA, READONLY
                   EXPORT    __Vectors
                   EXPORT    __Vectors_End
                   EXPORT    __Vectors_Size

__Vectors          DCD       __initial_sp                    ; Top of Stack
                   DCD       test_prerun                      ; Reset Handler --> test_prerun
                   DCD       NMI_Handler                       ; NMI Handler
                   DCD       HardFault_Handler                ; Hard Fault Handler
                   DCD       MemManage_Handler                    ; MPU Fault Handler
                   DCD       BusFault_Handler                 ; Bus Fault Handler
                   DCD       UsageFault_Handler                ; Usage Fault Handler
                   DCD       0                                ; Reserved
                   DCD       0                                ; Reserved
                   DCD       0                                ; Reserved
                   DCD       0                                ; Reserved
                   DCD       SVC_Handler                       ; SVCall Handler
                   DCD       DebugMon_Handler                     ; Debug Monitor Handler
                   DCD       0                                ; Reserved
                   DCD       PendSV_Handler                    ; PendSV Handler
                   DCD       SysTick_Handler                  ; SysTick Handler

;                  /* external interrupts handler */
                   DCD        WWDGT_IRQHandler                      ; 16:Window Watchdog
Timer
```

```
                DCD     LVD_IRQHandler                  ; 17:LVD through EXTI Line
detect
                DCD     TAMPER_IRQHandler               ; 18:Tamper through EXTI
Line detect
                DCD     RTC_IRQHandler                  ; 19:RTC through EXTI Line
                DCD     FMC_IRQHandler                  ; 20:FMC
                DCD     RCU_CTC_IRQHandler              ; 21:RCU and CTC
                DCD     EXTI0_IRQHandler                ; 22:EXTI Line 0
                DCD     EXTI1_IRQHandler                ; 23:EXTI Line 1
                DCD     EXTI2_IRQHandler                ; 24:EXTI Line 2
                DCD     EXTI3_IRQHandler                ; 25:EXTI Line 3
                DCD     EXTI4_IRQHandler                ; 26:EXTI Line 4
                DCD     DMA0_Channel0_IRQHandler        ; 27:DMA0 Channel0
                DCD     DMA0_Channel1_IRQHandler        ; 28:DMA0 Channel1
                DCD     DMA0_Channel2_IRQHandler        ; 29:DMA0 Channel2
                DCD     DMA0_Channel3_IRQHandler        ; 30:DMA0 Channel3
                DCD     DMA0_Channel4_IRQHandler        ; 31:DMA0 Channel4
                DCD     DMA0_Channel5_IRQHandler        ; 32:DMA0 Channel5
                DCD     DMA0_Channel6_IRQHandler        ; 33:DMA0 Channel6
                DCD     ADC0_1_IRQHandler               ; 34:ADC0 and ADC1
                DCD     CAN0_TX_IRQHandler              ; 35:CAN0 TX
                DCD     CAN0_RX0_IRQHandler             ; 36:CAN0 RX0
                DCD     CAN0_RX1_IRQHandler             ; 37:CAN0 RX1
                DCD     CAN0_EWMC_IRQHandler            ; 38:CAN0 EWMC
                DCD     EXTI5_9_IRQHandler              ; 39:EXTI5 to EXTI9
                DCD     TIMER0_BRK_TIMER8_IRQHandler    ; 40:TIMER0 Break and
TIMER8
                DCD     TIMER0_UP_TIMER9_IRQHandler     ; 41:TIMER0 Update and
TIMER9
                DCD     TIMER0_TRG_CMT_TIMER10_IRQHandler ; 42:TIMER0 Trigger and
Commutation and TIMER10
                DCD     TIMER0_Channel_IRQHandler       ; 43:TIMER0 Channel
Capture Compare
                DCD     TIMER1_IRQHandler               ; 44:TIMER1
                DCD     TIMER2_IRQHandler               ; 45:TIMER2
                DCD     TIMER3_IRQHandler               ; 46:TIMER3
                DCD     I2C0_EV_IRQHandler              ; 47:I2C0 Event
                DCD     I2C0_ER_IRQHandler              ; 48:I2C0 Error
                DCD     I2C1_EV_IRQHandler              ; 49:I2C1 Event
                DCD     I2C1_ER_IRQHandler              ; 50:I2C1 Error
                DCD     SPI0_IRQHandler                 ; 51:SPI0
                DCD     SPI1_IRQHandler                 ; 52:SPI1
                DCD     USART0_IRQHandler               ; 53:USART0
```

```
                DCD     USART1_IRQHandler                ; 54:USART1
                DCD     USART2_IRQHandler                ; 55:USART2
                DCD     EXTI10_15_IRQHandler             ; 56:EXTI10 to EXTI15
                DCD     RTC_Alarm_IRQHandler             ; 57:RTC Alarm
                DCD     USBFS_WKUP_IRQHandler            ; 58:USBFS Wakeup
                DCD     TIMER7_BRK_TIMER11_IRQHandler    ; 59:TIMER7 Break and
TIMER11
                DCD     TIMER7_UP_TIMER12_IRQHandler     ; 60:TIMER7 Update and
TIMER12
                DCD     TIMER7_TRG_CMT_TIMER13_IRQHandler ; 61:TIMER7 Trigger and
Commutation and TIMER13
                DCD     TIMER7_Channel_IRQHandler        ; 62:TIMER7 Channel
Capture Compare
                DCD     0                                ; Reserved
                DCD     EXMC_IRQHandler                  ; 64:EXMC
                DCD     0                                ; Reserved
                DCD     TIMER4_IRQHandler                ; 66:TIMER4
                DCD     SPI2_IRQHandler                  ; 67:SPI2
                DCD     UART3_IRQHandler                 ; 68:UART3
                DCD     UART4_IRQHandler                 ; 69:UART4
                DCD     TIMER5_IRQHandler                ; 70:TIMER5
                DCD     TIMER6_IRQHandler                ; 71:TIMER6
                DCD     DMA1_Channel0_IRQHandler         ; 72:DMA1 Channel0
                DCD     DMA1_Channel1_IRQHandler         ; 73:DMA1 Channel1
                DCD     DMA1_Channel2_IRQHandler         ; 74:DMA1 Channel2
                DCD     DMA1_Channel3_IRQHandler         ; 75:DMA1 Channel3
                DCD     DMA1_Channel4_IRQHandler         ; 76:DMA1 Channel4
                DCD     ENET_IRQHandler                  ; 77:Ethernet
                DCD     ENET_WKUP_IRQHandler             ; 78:Ethernet Wakeup
through EXTI Line
                DCD     CAN1_TX_IRQHandler               ; 79:CAN1 TX
                DCD     CAN1_RX0_IRQHandler              ; 80:CAN1 RX0
                DCD     CAN1_RX1_IRQHandler              ; 81:CAN1 RX1
                DCD     CAN1_EWMC_IRQHandler             ; 82:CAN1 EWMC
                DCD     USBFS_IRQHandler                 ; 83:USBFS

__Vectors_End

                AREA CHECKSUM, DATA, READONLY, ALIGN=2
                EXPORT __Check_Sum
                ALIGN
                                                ;MARCHC         MARCHX
__Check_Sum     DCD 0xEEF15A05                  ;0xEEF15A05     0x56A3240D  This
```

```
value is different according to actual condition
;__Check_Sum            DCD  0x14FF                    ;0x14FF          0xBA87          This
value is different according to actual condition


__Vectors_Size  EQU     __Vectors_End - __Vectors

                AREA    |.text|, CODE, READONLY

;/* reset Handler */
Reset_Handler   PROC
                EXPORT  Reset_Handler                  [WEAK]
                IMPORT  SystemInit
                IMPORT  __main
                LDR     R0, =SystemInit
                BLX     R0
                LDR     R0, =__main
                BX      R0
                ENDP
```

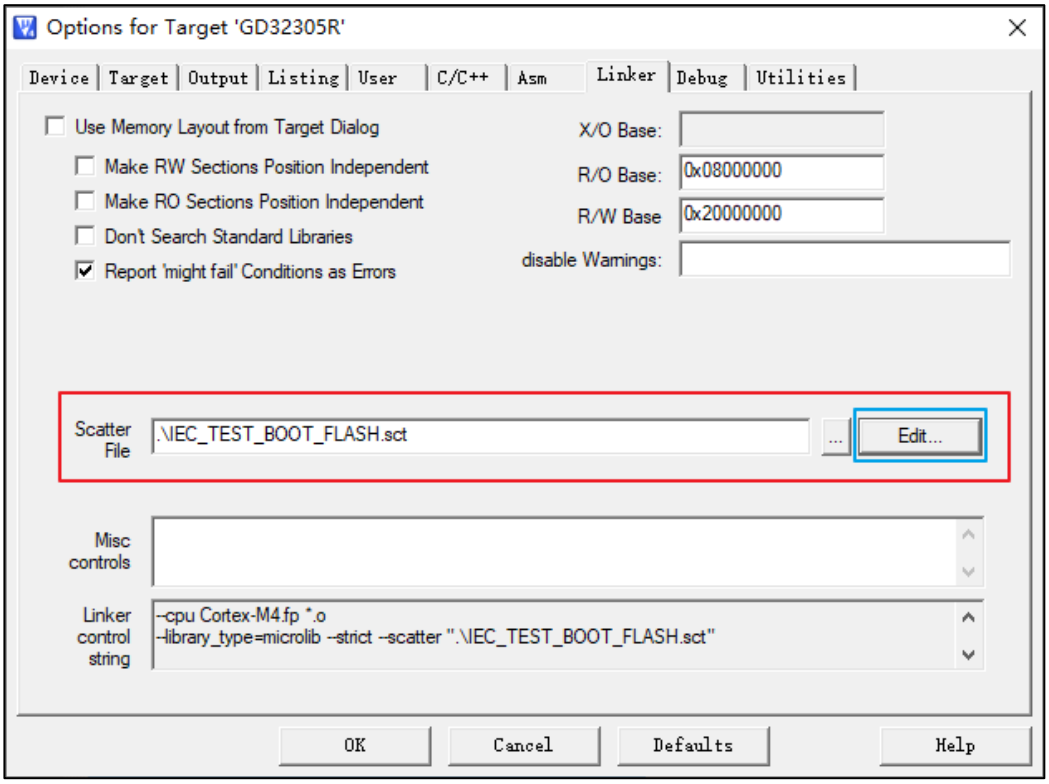
3、修改分散加载文件IEC_TEST_BOOT_FLASH.sct中的代码，如下表红色标注为需要修改的部分（根据当前芯片的数据手册修改Flash边界；修改与.s启动文件相对应.o文件的名称）：

```
LR_IROM1 0x08000000 0x0003FFFF{
    ER_IROM1 0x08000000 0x0003FFFF {
        *.o (RESET, +First)
        *(InRoot$$Sections)
        .ANY (+RO)
    }

    ; RAM test during run time
    RAM_BUF 0x20000004
    {
        gd32f30x_test_prerun.o (RAM_RUN_BUF)
    }

    ; RAM pointer during run time
    RAM_PTR 0x20000030
    {
        gd32f30x_test_prerun.o (RAM_RUN_PTR)
    }

    ; variables of IEC test
    IEC_TEST_VAR 0x20000040 UNINIT 0x0000070
```

```
    {
        gd32f30x_test_prerun.o (IEC_TEST_RAM)
    }

    ; RW data
    RW_IRAM1 0x200000B0 UNINIT 0x00005000
    {
        .ANY (+RW +ZI)
    }

    ; stack overflow test
    STACK_IRAM2 0x200050B0 UNINIT 0x00006F40
    {
        gd32f30x_test_prerun.o (STACK_OV_TEST)
        startup_gd32f30x_cl.o (STACK, +Last)
    }
}

LR_IROM2 0x0803FFFC 0x0000004 {
    ER_IROM2 0x0803FFFC 0x0000004
    {
        *.o (CHECKSUM, +Last)
    }
}
```

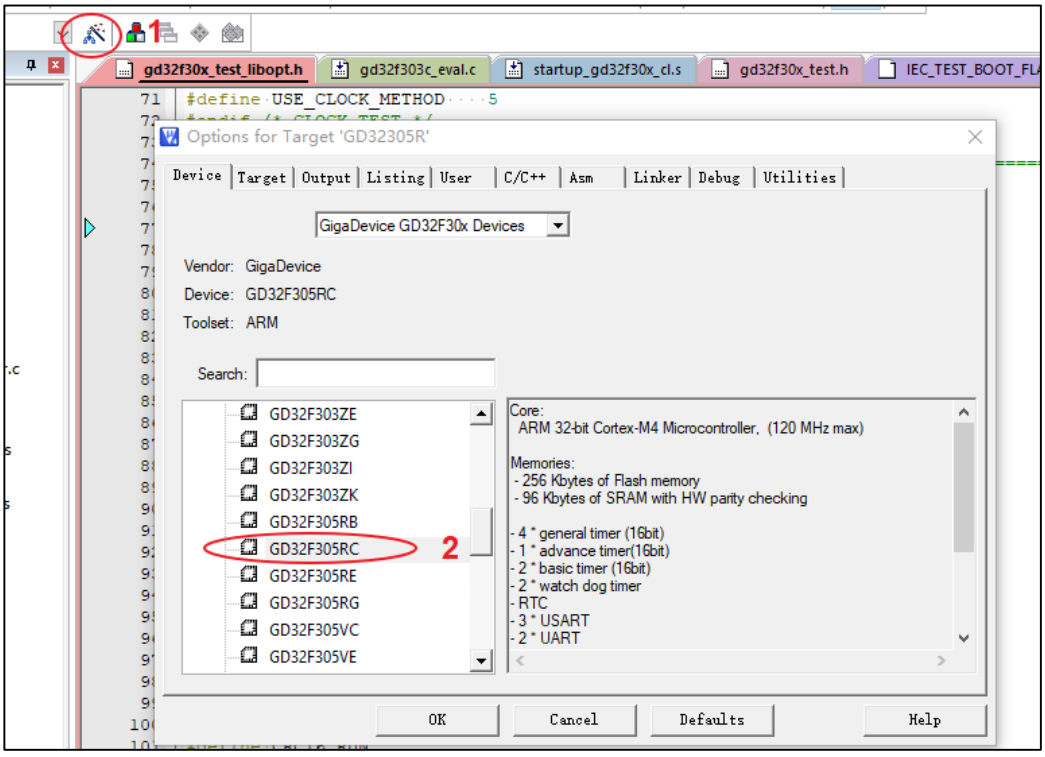
分散加载文件可在“Options for Target ‘GD32F305R’-->Linker-->Scatter File”中点击Edit按钮即可修改，如*图2-11. 编辑分散加载文件*所示：

图 2-11. 编辑分散加载文件



4、修改“Options for Target ‘GD32F305R’中DEVICE”的配置，选择当前芯片的型号，如*图2-12. Device配置*所示：

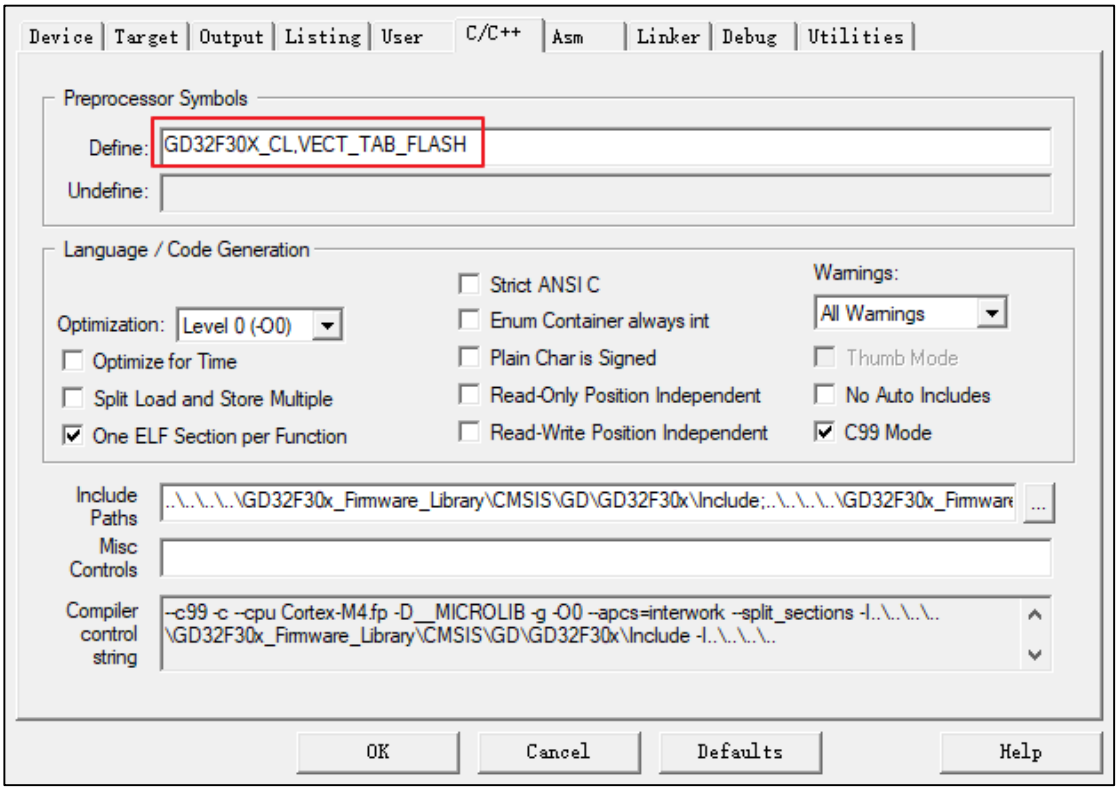
图 2-12. Device 配置



5、在Options for Target ‘GD32F305R’->C/C++->Preprocessor Symbols中添加当前工程需要

的预编译宏与当前芯片一致，如*图2-13. 修改预编译宏*所示：

图 2-13. 修改预编译宏

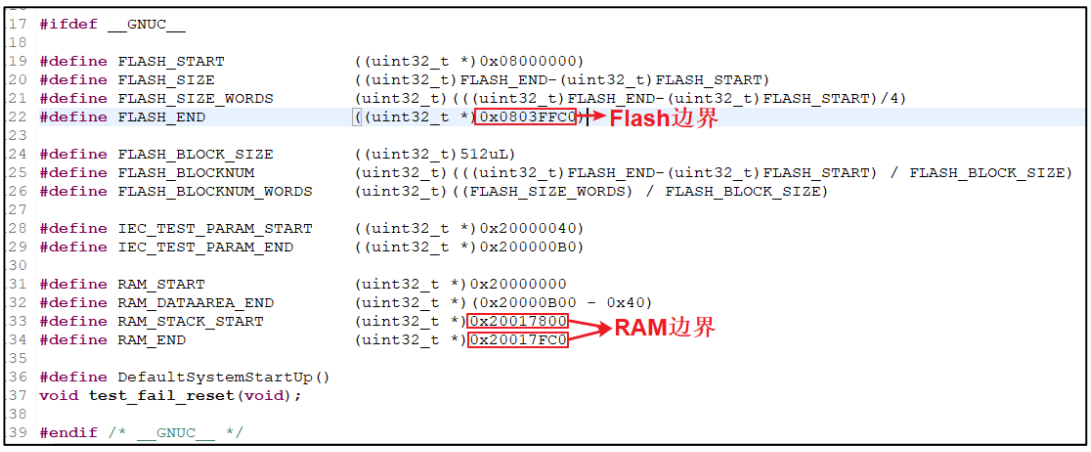


2.3. eclipse 环境认证库工程移植

认证库工程在各个环境下的移植步骤类似，但是各开发环境下的编译链有所区别，导致 RAM 和 Flash 边界设置有所不同，eclipse 环境下的移植步骤如下：

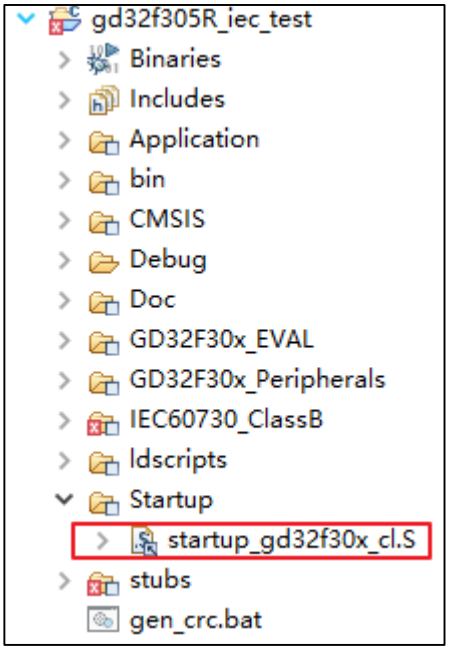
1、根据芯片数据手册修改gd32f30x_test.h文件中编译宏__GNU__下的RAM边界，保证芯片的Flash和RAM全部空间得到检测，如*图2-14. 修改gd32f30x_test.h中RAM边界*所示。

图 2-14. 修改 gd32f30x_test.h 中 RAM 边界



2、检查工程中的.S 启动文件是否适用于当前芯片，如*图2-15. 启动文件检查*所示：

图 2-15. 启动文件检查



若符合，则无需修改；若不符合，则需，重新选择符合当前芯片类型(cl/hd/xd）的.s启动文件，并如*图2-16. 修改启动文件*所示进行修改，使芯片在运行之前进行自检测。

图 2-16. 修改启动文件

```
48
49  bl  test_prerun /* SystemInit */
50
51/* Call SystemInit function */
52  bl  SystemInit
53///* Call static constructors */
54//  bl __libc_init_array
55/*Call the main function */
56  bl main
57  bx lr
58.size Reset_Handler, .-Reset_Handler
59
60
61
62    .section .text.Default_Handler,"ax",%progbits
63Default_Handler:
64Infinite_Loop:
65  b Infinite_Loop
66  .size Default_Handler, .-Default_Handler
67
68
69   .section  .isr_vector,"a",%progbits
70   .global __gVectors
71
72
73
74__gVectors:
75                    .word _estack                    /* Top of Stack */
```

3、修改如*图2-17. 分散加载文件位置*目录下的分散加载文件，如*图2-18. 修改eclipse工程分散加载文件*所示为修改的部分，根据芯片数据手册修改Flash和RAM的大小匹配当前芯片。

图 2-17. 分散加载文件位置

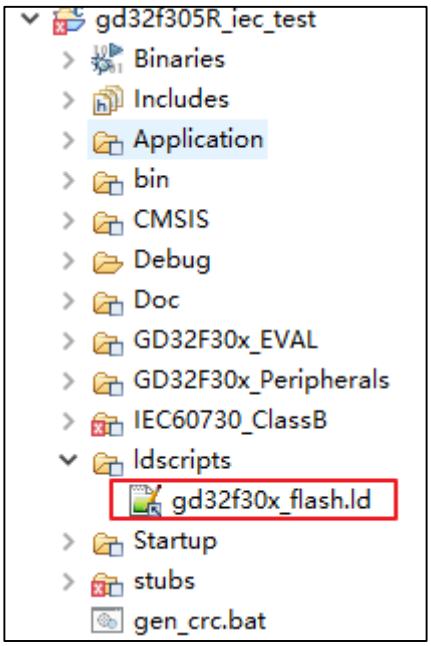
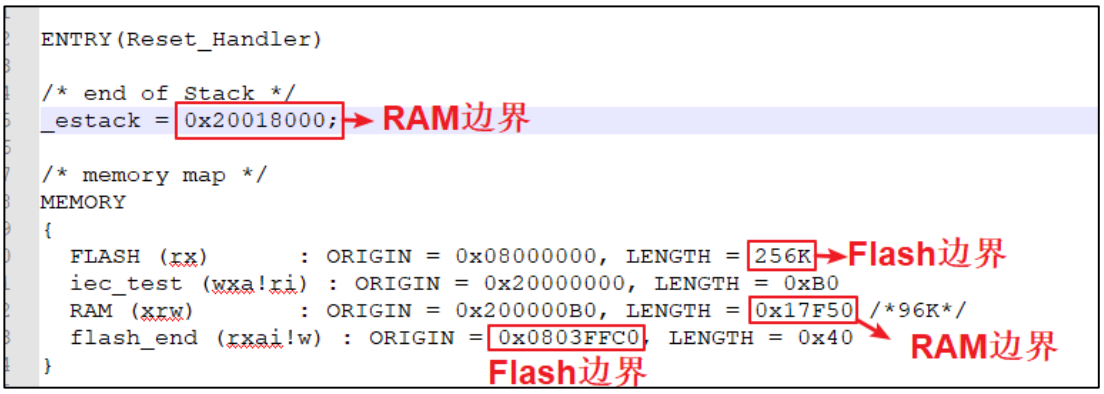
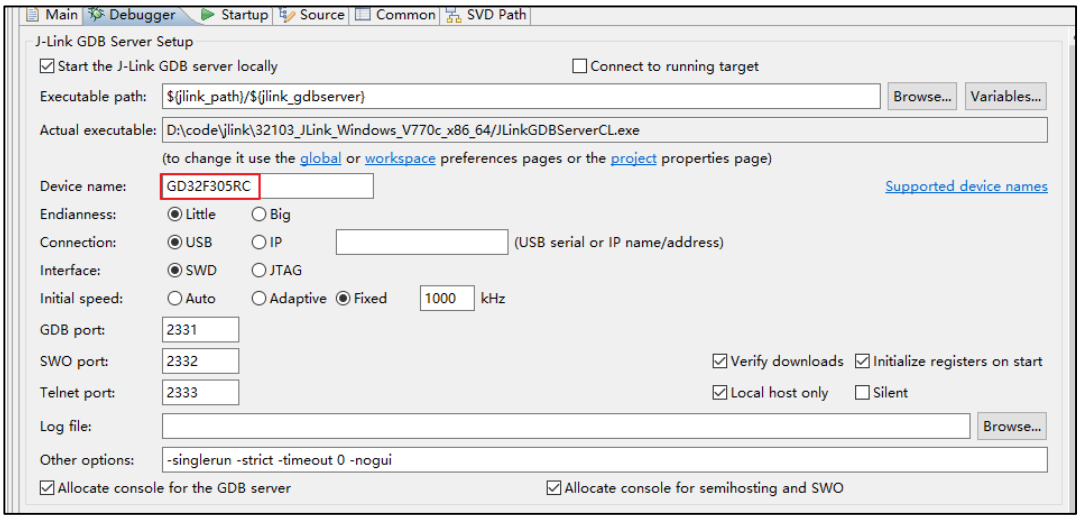


图 2-18. 修改 eclipse 工程分散加载文件



4、修改“Debug Configurations->Debugger”中Device name的配置，选择当前芯片的型号，如*图2-19. Device name配置*所示：

图 2-19. Device name 配置



5、在工程属性“C/C++ Build->Settings->Tool Settings->Cross ARM GNU Assembler->Preprocessor”和“C/C++ Build->Settings->Tool Settings->Cross ARM GNU C Complier->Preprocessor”中添加当前工程需要的预编译宏，主要修改符合当前芯片类型的预编译宏（在本例中为GD32F30X_CL），如*图2-20. 修改汇编编译器预编译宏*和*图2-21. 修改C语言编译器预编译宏*所示：

图 2-20. 修改汇编编译器预编译宏

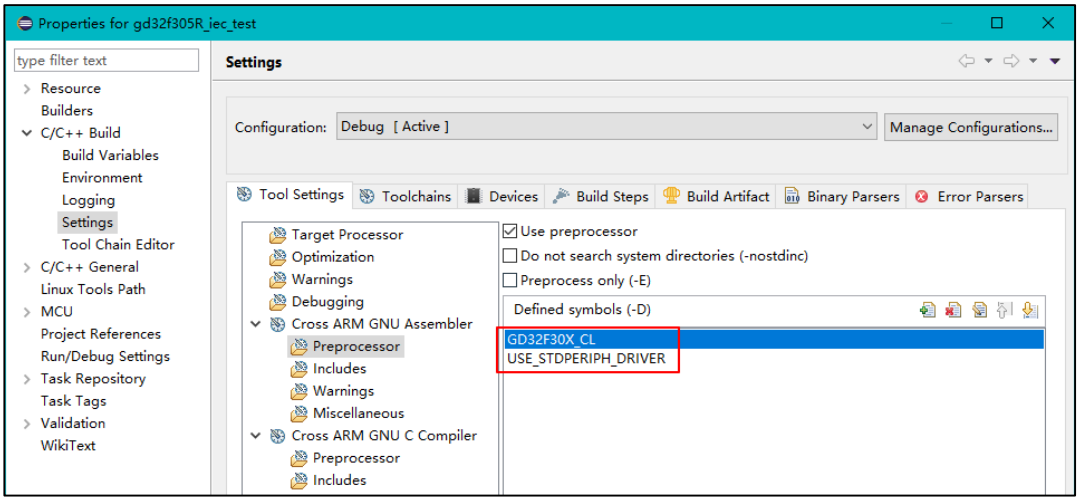
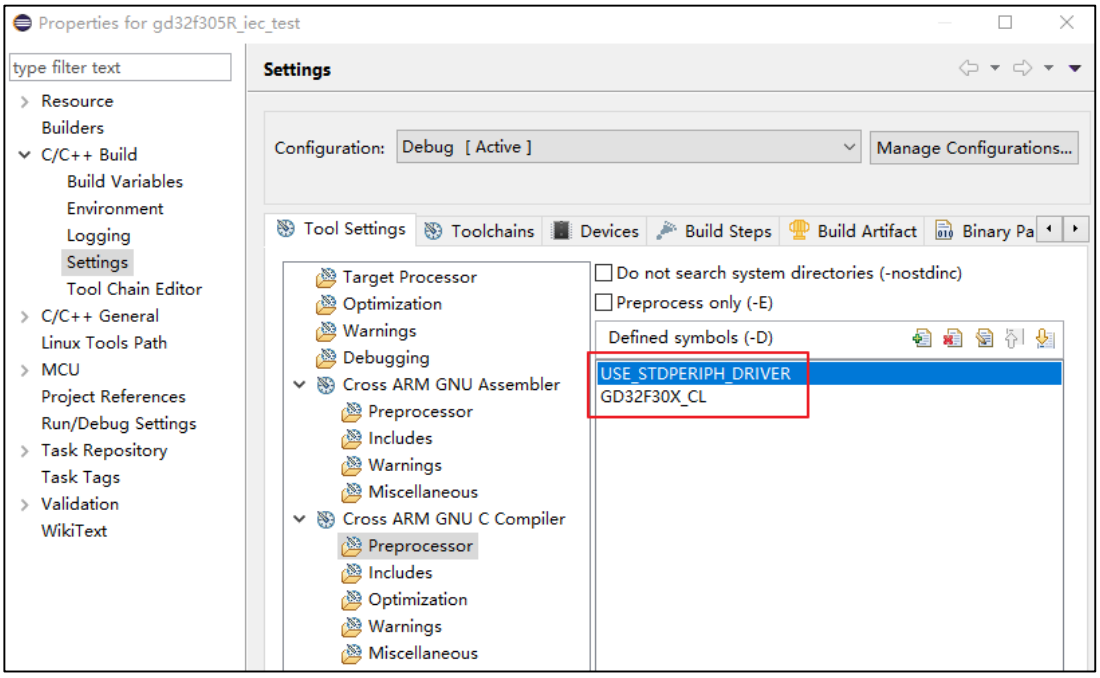
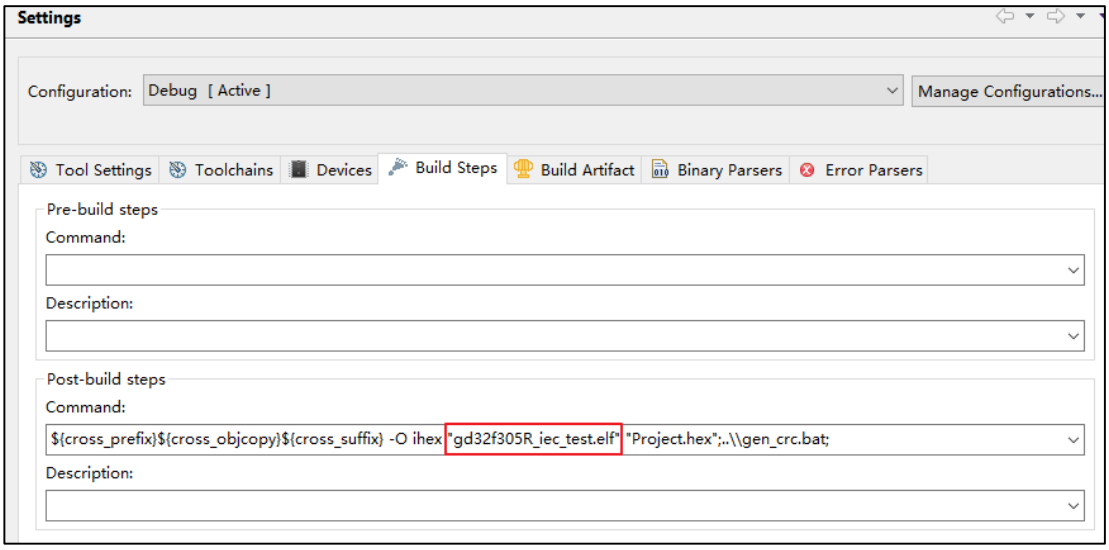


图 2-21. 修改 C 语言编译器预编译宏



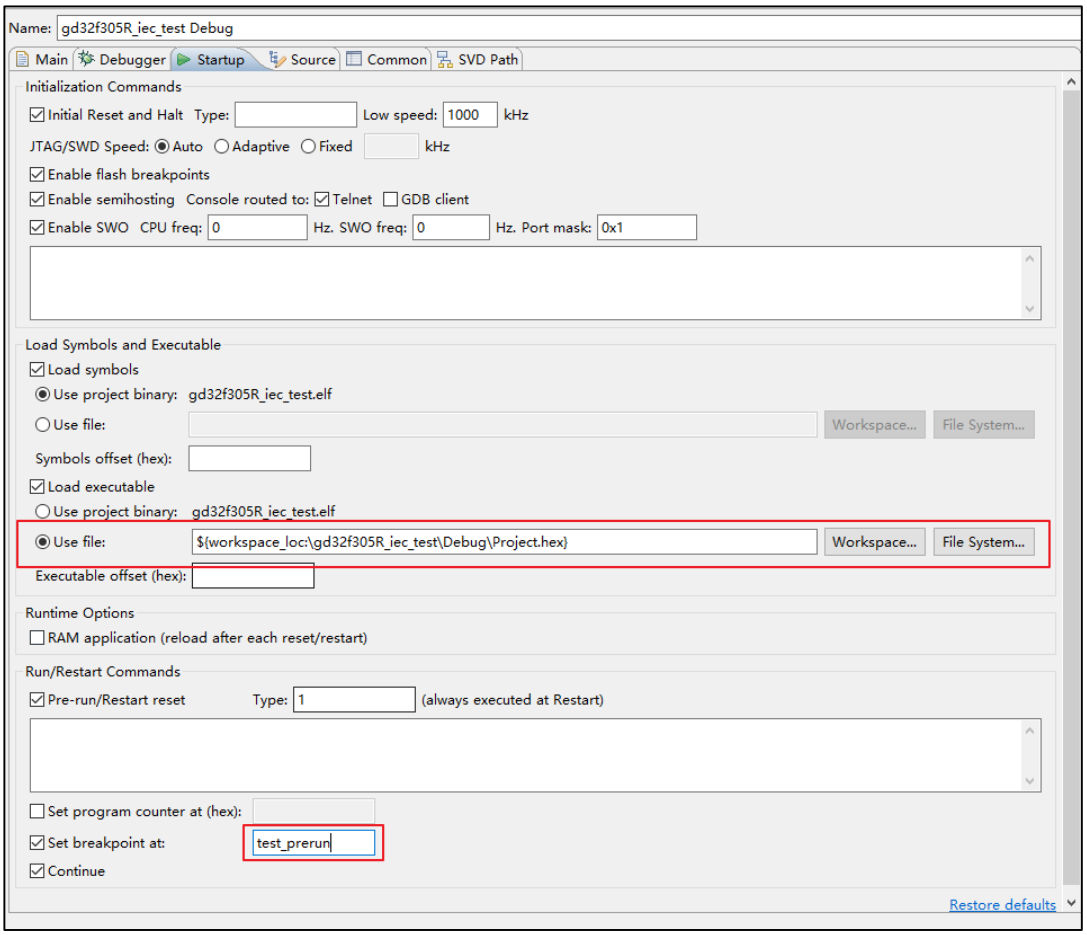
6、修改“C++ Build->Settings->Build Steps->Post-build steps->Command”中的命令，如*图2-22. 修改Post-build指令*所示更换为当前工程名称的elf文件。

图 2-22. 修改 Post-build 指令



7、修改“Debug Configurations->Startup”中的可执行文件的配置，选择当前工作空间中编译生成的Project.hex文件，如*图2-23. 修改可执行文件配置*所示：

图 2-23. 修改可执行文件配置



3. 三种环境下的测试结果

按照第二章的步骤修改代码和配置工程，在各个环境下编译下载工程并运行，测试结果如*图 3-1. 测试结果*所示。

图 3-1. 测试结果

```
>>>>>>>>>>>>>>>  IEC60730 Test Board Init Success  <<<<<<<<<<<<<<<

CPU Test(PreRun) Success!

... Power reset or software reset, next step --> FWDGT reset test ...

>>>>>>>>>>>>>>>  IEC60730 Test Board Init Success  <<<<<<<<<<<<<<<

CPU Test(PreRun) Success!

FWDGT reset
... FWDGT reset test OK, next step --> WWDGT reset test ...

>>>>>>>>>>>>>>>  IEC60730 Test Board Init Success  <<<<<<<<<<<<<<<

CPU Test(PreRun) Success!

FWDGT reset
WWDGT reset
... WWDGT reset test OK, WDGT test completed ...

Full
 RAM Test Success!

FLASH CRC16 Test(PreRun) Success!

lock Frequency Test Success!

Program counter test(PreRun) Success!

***************************** Main program starts *****************************

FLASH CRC(Run-Time) Test running! Next Address --> 0x08000200

FLASH CRC(Run-Time) Test running! Next Address --> 0x08000400

FLASH CRC(Run-Time) Test running! Next Address --> 0x08000600
```

4. 版本历史

表 4-1. 版本历史

版本号.	说明	日期
1.0	首次发布	2023 年 09 月 4 日

Important Notice

This document is the property of GigaDevice Semiconductor Inc. and its subsidiaries (the "Company"). This document, including any product of the Company described in this document (the “Product”), is owned by the Company under the intellectual property laws and treaties of the People’s Republic of China and other jurisdictions worldwide. The Company reserves all rights under such laws and treaties and does not grant any license under its patents, copyrights, trademarks, or other intellectual property rights. The names and brands of third party referred thereto (if any) are the property of their respective owner and referred to for identification purposes only.

The Company makes no warranty of any kind, express or implied, with regard to this document or any Product, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The Company does not assume any liability arising out of the application or use of any Product described in this document. Any information provided in this document is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Except for customized products which has been expressly identified in the applicable agreement, the Products are designed, developed, and/or manufactured for ordinary business, industrial, personal, and/or household applications only. The Products are not designed, intended, or authorized for use as components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, atomic energy control instruments, combustion control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or Product could cause personal injury, death, property or environmental damage ("Unintended Uses"). Customers shall take any and all actions to ensure using and selling the Products in accordance with the applicable laws and regulations. The Company is not liable, in whole or in part, and customers shall and hereby do release the Company as well as it’s suppliers and/or distributors from any claim, damage, or other liability arising from or related to all Unintended Uses of the Products. Customers shall indemnify and hold the Company as well as it’s suppliers and/or distributors harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of the Products.

Information in this document is provided solely in connection with the Products. The Company reserves the right to make changes, corrections, modifications or improvements to this document and Products and services described herein at any time, without notice.