GigaDevice Semiconductor Inc.

Arm® Cortex®-M3/4/23/33 32-bit MCU

应用笔记

AN038

目录

图索引

表索引

1. 介绍

在IEC60730自检测试中，要求对mcu片上flash进行自检，为了实现CRC值的自动计算和添加，需要在IDE中添加CRC校验步骤，为此，本应用笔记讲述如何在Keil环境中添加CRC校验批处理方法，过程描述如下。

2. CRC 校验批处理添加

Keil 环境中没有直接计算 CRC 值的功能，因此需要借助开源工具计算 CRC 值，并在编译过程中通过批处理将其自动添加到.hex 文件的末尾，以便于 flash 自检程序的正常运行。

2.1. 添加 CRC 值所需文件

1. 首先下载 Srecord 工具，在工程目录下新建 bin 文件夹，将 srec_cat.exe、srec_cmp.exe 和 srec_info.exe 拷贝到该文件夹下。

2. 在工程目录下添加 gen_crc.bat、IEC_TEST_BOOT_FLASH.sct 和 CRC_LOAD.ini 文件。gen_crc.bat 文件用于调用 Srecord 工具计算 CRC 值，并将该值存储到.hex 文件的末尾以便于对整个 flash 进行自检测试，并在 build 窗口打印通过 Srecord 工具计算得到的 CRC 值。该文件的命令代码如下所示：

```
SET MAP_FILE=Listings\Project.map

::-----------------get CRC address information line
SET TMP_FILE=crc_temp.txt
FINDSTR /R /C:"^    *CHECKSUM" %MAP_FILE%>%TMP_FILE%
SET /p crc_search=<%TMP_FILE%
DEL %TMP_FILE%
::-----------------CRC address
for /f "tokens=1 delims=(" %%a in ("%crc_search%") do set crc_search=%%a
SET crc_search=%crc_search:CHECKSUM=%
for /f "tokens=1 delims= " %%a in ("%crc_search%") do set CRC_ADDR=%%a


SET /a CRC_ADDR_END=%CRC_ADDR%+4
::-----------------choose CRC32 or CRC16
FINDSTR /R /C:"^      *crc_block_data_calculate" %MAP_FILE% > nul && call :OK||call :NO


goto :eof
:OK
::-------------------CRC32
bin\srec_cat.exe Objects\Project.hex -intel -crop 0x08000000 %CRC_ADDR% -fill 0xff 0x08000000 %CRC_ADDR% -stm32-l-e %CRC_ADDR% -o Objects\Project_checked.hex -intel
bin\srec_cat.exe Objects\Project.hex -intel -crop 0x08000000 %CRC_ADDR% Objects\Project_checked.hex -intel -crop %CRC_ADDR% %CRC_ADDR_END% -o
```

```
Objects\Project.hex -intel
bin\srec_cat.exe Objects\Project.hex -intel -crop %CRC_ADDR% %CRC_ADDR_END% -o -hex-dump
goto :eof

:NO
::--------------------CRC16
bin\srec_cat.exe Objects\Project.hex -intel -crop 0x08000000 %CRC_ADDR% -fill 0xff 0x08000000 %CRC_ADDR% -crc16-l-e %CRC_ADDR% -POLYnomial ccitt -XMODEM -o Objects\Project_checked.hex -intel
bin\srec_cat.exe Objects\Project.hex -intel -crop 0x08000000 %CRC_ADDR% Objects\Project_checked.hex -intel -crop %CRC_ADDR% %CRC_ADDR_END% -o Objects\Project.hex -intel
bin\srec_cat.exe Objects\Project.hex -intel -crop %CRC_ADDR% %CRC_ADDR_END% -o -hex-dump
goto :eof

exit
```

IEC_TEST_BOOT_FLASH.sct 是分散加载文件，其定义各个程序段和变量的加载地址，通过如下所示的代码，将 CRC 值（CHECKSUM）固定在 FLASH 空间的末尾位置。

```
; *************************************************************
; *** Scatter-Loading Description File generated by uVision ***
; *************************************************************

LR_IROM1 0x08000000 0x0001FFF8 {
    ER_IROM1 0x08000000 0x0001FFF8 {
        *.o (RESET, +First)
        *(InRoot$$Sections)
        .ANY (+RO)
    }

    ; RAM test during run time
    RAM_BUF 0x20000004
    {
        gd32e10x_test_prerun.o (RAM_RUN_BUF)
    }

    ; RAM pointer during run time
    RAM_PTR 0x20000030
    {
        gd32e10x_test_prerun.o (RAM_RUN_PTR)
    }
```

```
    ; variables of IEC test
    IEC_TEST_VAR 0x20000040 UNINIT 0x0000070
    {
        gd32e10x_test_prerun.o (IEC_TEST_RAM)
    }

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

    ; stack overflow test
    STACK_IRAM2 0x200050B0 UNINIT 0x00002F40
    {
        gd32e10x_test_prerun.o (STACK_OV_TEST)
        startup_gd32e10x.o (STACK, +Last)
    }
}

LR_IROM2 0x0801FFFC 0x0000004 {
    ER_IROM2 0x0801FFFC 0x0000004
    {
        *.o (CHECKSUM, +Last)
    }
}
```

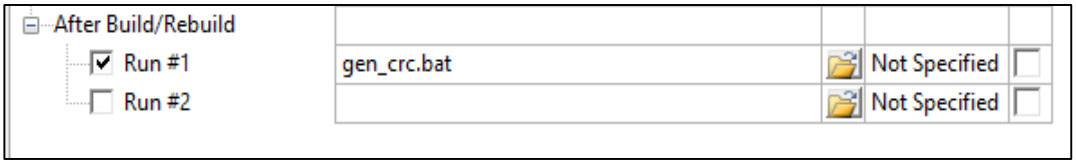
CRC_LOAD.ini 文件用于将添加 CRC 值之后的.hex 文件下载到被测 MCU 中，以实现 CRC 值自动添加到自检程序中，无需手动添加，其代码如下：

```
LOAD "Objects\\Project.hex"
```

2.2. 配置批处理

1. 在工程目录中添加了上述三个文件之后，打开工程，在 Options for Target->user->After Build/Rebuild 中选择 gen_crc.bat 文件，如*图 2-1. 选择 gen_crc.bat 文件*所示。

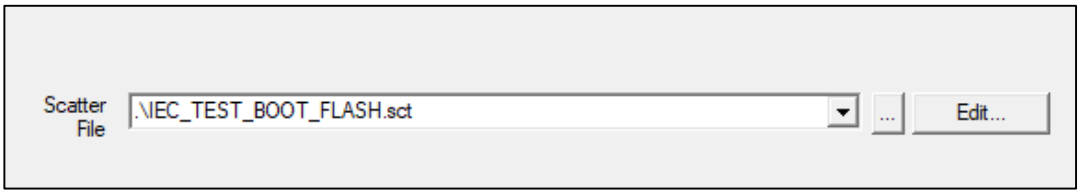
图 2-1. 选择 gen_crc.bat 文件



2. 在 Options for Targe->Linker 的 Scatter File 选项中选择 IEC_TEST_BOOT_FLASH.sct

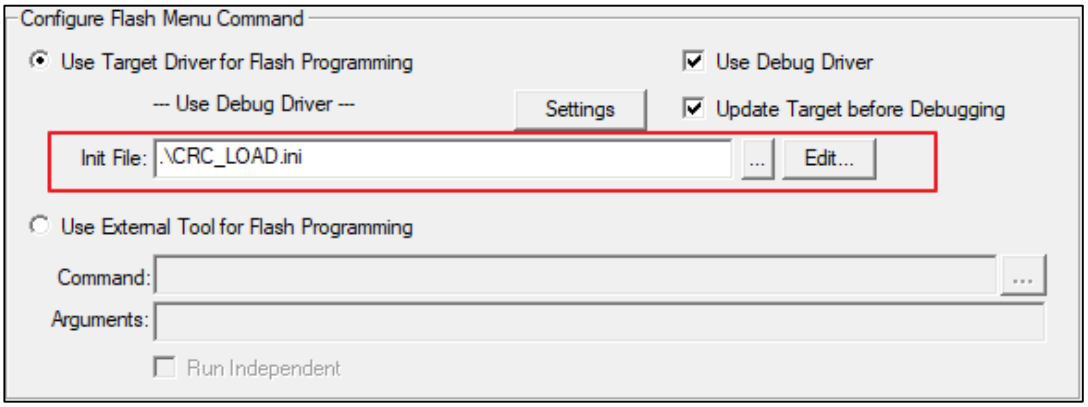
文件，如*图 2-2. 选择 IEC_TEST_BOOT_FLASH.sct 文件* 所示。

图 2-2. 选择 IEC_TEST_BOOT_FLASH.sct 文件



3. 在 Options for Targe->Utilities 的 Init File 选项中选择 CRC_LOAD.ini 文件,如*图 2-3. 选择 CRC_LOAD.ini 文件*所示。

图 2-3. 选择 CRC_LOAD.ini 文件



3. 结果

配置完成后，点击编译，可以在 Build Output 窗口观察 build 信息，如*图 3-1. Build 信息*所示，build 信息显示 CRC 值已存储在 0x0801FFF0 之后的位置；点击调试，并在 memory 观察窗口查询 0x0801FFF0 地址，如*图 3-2. 0x0801FFF0 地址*所示，可以看出 memory 中的值与 Build Output 窗口显示的 CRC 值一致，CRC 值计算批处理添加成功。

图 3-1. Build 信息

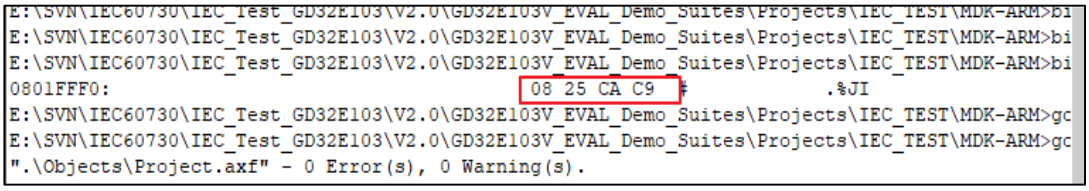
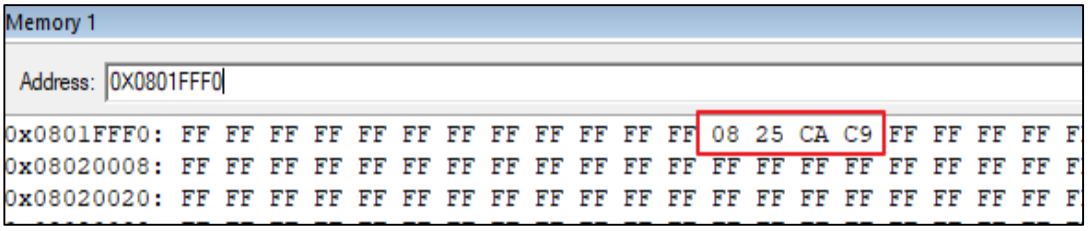


图 3-2. 0x0801FFF0 地址



4. 版本历史

表 4-1. 版本历史

版本号.	描述	日期
1.0	首次发布	2021 年 10 月 19 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.