# GigaDevice Semiconductor Inc.

# Arm® Cortex®-M3/4/23/33 32-bit MCU

## 应用笔记

## AN039

# 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eclipse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

在 eclipse 环境中，没有提供自动计算 CRC 的功能，为了实现 IEC60730 标准要求的 flash 自检，通常预先计算出 CRC 值通过手动方式添加到分散加载文件中，这样的方式操作复杂，为此，通过添加批处理步骤来完成 CRC 值的自动计算和添加。

## 2.1. 添加 CRC 值批处理所需文件

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、gd32e10x_flash.ld 文件。在此以 gd32e10x 系列芯片为例进行说明，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=gd32e10x_iec_test.map
::-----------------get CRC address information line
SET TMP_FILE=crc_temp.txt
FINDSTR /R /C:"^ *.check_sum" %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:.check_sum=%
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 Project.hex -intel -crop 0x08000000 %CRC_ADDR% -fill 0xff 0x08000000 %CRC_ADDR% -stm32-l-e %CRC_ADDR% -o Project_checked.hex -intel
..\bin\srec_cat.exe Project.hex -intel -crop 0x08000000 %CRC_ADDR%
```

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

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

exit
```

gd32e10x_flash.ld 文件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。

```

ENTRY(Reset_Handler)

/* end of Stack */
_estack = 0x20008000;    /*ram size*/

/* memory map */
MEMORY
{
  FLASH (rx)          : ORIGIN = 0x08000000, LENGTH = 128K       /*flash size*/
  iec_test (wxa!ri) : ORIGIN = 0x20000000, LENGTH = 0xB0
  RAM (xrw)           : ORIGIN = 0x200000B0, LENGTH = 0x7F50    /*32K*/
  flash_end (rxai!w) : ORIGIN = 0x0801FFC0, LENGTH = 0x40 /*flash size - 0x40*/
}

SECTIONS
{
  __stack_size = DEFINED(__stack_size) ? __stack_size : 2K;

```

```
.isr_vector :
{
  . = ALIGN(4);
  KEEP(*(.isr_vector))
  . = ALIGN(4);
} >FLASH

.text :
{
  . = ALIGN(4);
  *(.text)
  *(.text*)
  *(.glue_7)
  *(.glue_7t)
  *(.eh_frame)

  KEEP (*(.init))
  KEEP (*(.fini))

  . = ALIGN(4);
  /* the symbol '_etext' will be defined at the end of code section */
  _etext = .;
} >FLASH

.rodata :
{
  . = ALIGN(4);
  *(.rodata)
  *(.rodata*)
  . = ALIGN(4);
} >FLASH

 .ARM.extab :
{
   *(.ARM.extab* .gnu.linkonce.armextab.*)
} >FLASH

  .ARM : {
  __exidx_start = .;
    *(.ARM.exidx*)
    __exidx_end = .;
  } >FLASH
```

```
.ARM.attributes : { *(.ARM.attributes) } > FLASH

.preinit_array     :
{
  PROVIDE_HIDDEN (__preinit_array_start = .);
  KEEP (*(.preinit_array*))
  PROVIDE_HIDDEN (__preinit_array_end = .);
} >FLASH

.init_array :
{
  PROVIDE_HIDDEN (__init_array_start = .);
  KEEP (*(SORT(.init_array.*)))
  KEEP (*(.init_array*))
  PROVIDE_HIDDEN (__init_array_end = .);
} >FLASH

.fini_array :
{
  PROVIDE_HIDDEN (__fini_array_start = .);
  KEEP (*(.fini_array*))
  KEEP (*(SORT(.fini_array.*)))
  PROVIDE_HIDDEN (__fini_array_end = .);
} >FLASH

._iec_classb(NOLOAD) :
{
  . = ALIGN(4);
  *(.ram_run_buf)
  *(.ram_run_buf*)
  *(.ram_run_ptr)
  *(.ram_run_ptr*)

  . = ALIGN(4);
  PROVIDE(_iec_start = . );

  *(.iec_test_ram)
  *(.iec_test_ram*)

  . = ALIGN(4);
  PROVIDE(_iec_end = . );
} >iec_test AT>iec_test

/* provide some necessary symbols for startup file to initialize data */
_sidata = LOADADDR(.data);
.data :
{
   . = ALIGN(4);
   /* the symbol '_sdata' will be defined at the data section end start */
   _sdata = .;
   *(.data)
   *(.data*)
   . = ALIGN(4);
   /* the symbol '_edata' will be defined at the data section end */
   _edata = .;
} >RAM AT> FLASH

.check_sum          :
{
   . = ALIGN(64);
   PROVIDE( __Check_Sum = . );

   LONG(0x904eae51);
} >flash_end AT>flash_end

. = ALIGN(4);
.bss :
{
   /* the symbol '_sbss' will be defined at the bss section start */
   _sbss = .;
   __bss_start__ = _sbss;
   *(.bss)
   *(.bss*)
   *(COMMON)
   . = ALIGN(4);
   /* the symbol '_ebss' will be defined at the bss section end */
   _ebss = .;
   __bss_end__ = _ebss;
} >RAM

PROVIDE ( end = _ebss );
PROVIDE ( _end = _ebss );
```

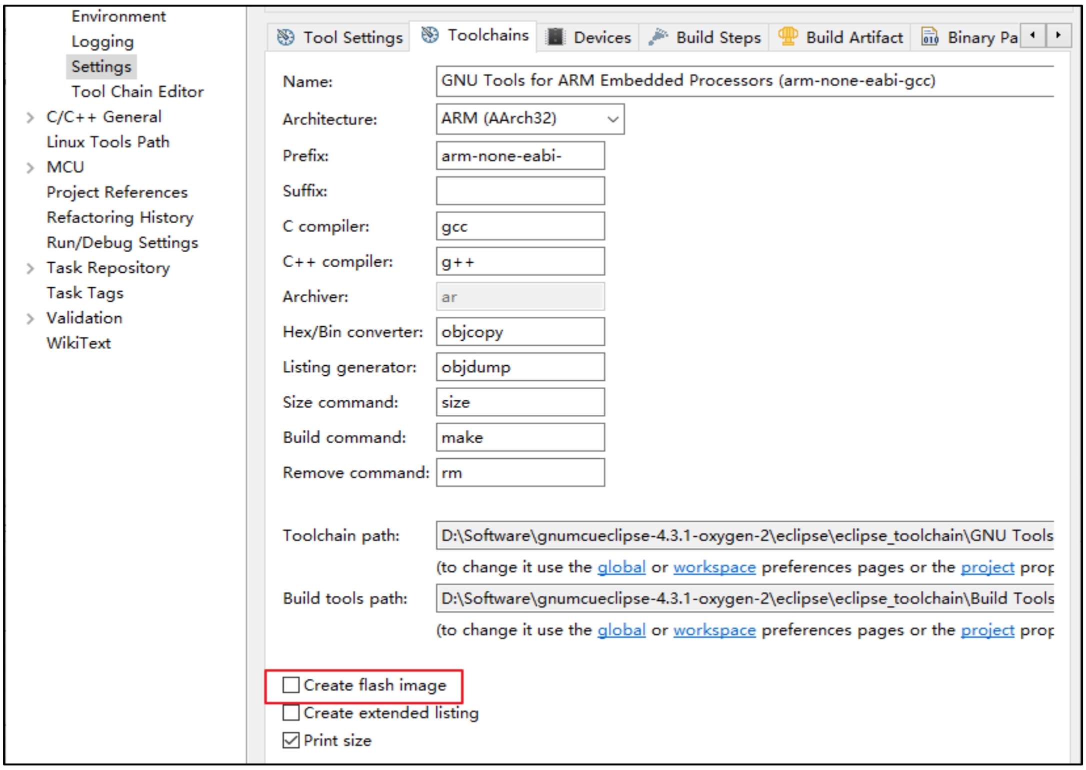
```
  .stack ORIGIN(RAM) + LENGTH(RAM) - __stack_size - 0x18:
  {
    *(.stack_ov_test)
    *(.stack_ov_test*)
    . = 0x18;
    PROVIDE( _heap_end = . );
    . = __stack_size + 0x18;
    PROVIDE( _sp = . );
  }
}

 /* input sections */
GROUP(libgcc.a libc.a libm.a libnosys.a)
```

## 2.2. 配置批处理

1. 在工程目录中添加了上述两个文件之后，打开工程，右击工程名，并在 Properties->C/C++ Build->Setting->Toolchains 中将 Create flash image 的勾选取消，这是为了实现利用批处理来生成可执行文件，如*图 2-1. 取消 Create flash image 勾选*所示。

图 2-1. 取消 Create flash image 勾选



Name: GNU Tools for ARM Embedded Processors (arm-none-eabi-gcc)
Architecture: ARM (AArch32)
Prefix: arm-none-eabi-
Suffix:
C compiler: gcc
C++ compiler: g++
Archiver: ar
Hex/Bin converter: objcopy
Listing generator: objdump
Size command: size
Build command: make
Remove command: rm

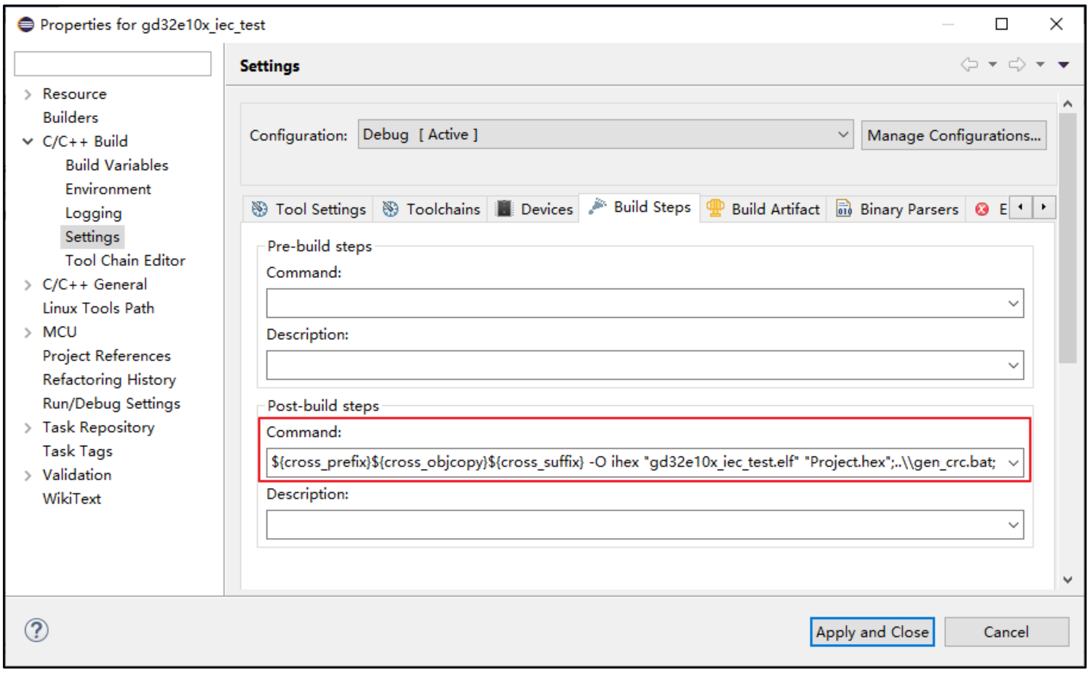
Toolchain path: D:\Software\gnumcueclipse-4.3.1-oxygen-2\eclipse\eclipse_toolchain\GNU Tools
(to change it use the global or workspace preferences pages or the project prop
Build tools path: D:\Software\gnumcueclipse-4.3.1-oxygen-2\eclipse\eclipse_toolchain\Build Tools
(to change it use the global or workspace preferences pages or the project prop

□ Create flash image
□ Create extended listing
☑ Print size

2. 右击工程名称->Properties->C/C++build->settings->build steps，在 Post-build steps 中添加以下命令：${cross_prefix}${cross_objcopy}${cross_suffix} -O ihex

"gd32f30x_iec_test.elf" "Project.hex";..\\gen_crc.bat;点击 apply，此命令是为了保证将.elf 文件转换成.hex 文件的操作在.bat 文件命令操作之前，设置方式如*图 2-2. steps 设置* 所示。

图 2-2. Post-build steps 设置



Properties for gd32e10x_iec_test

Settings

Configuration: Debug [ Active ]

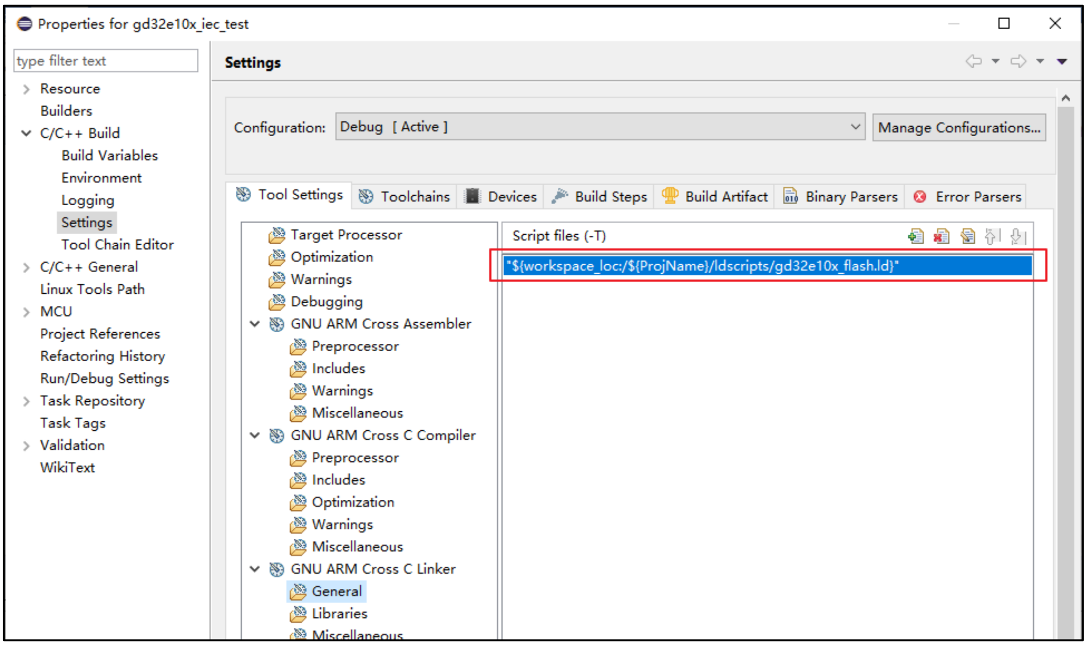
Post-build steps

Command:

```
${cross_prefix}${cross_objcopy}${cross_suffix} -O ihex "gd32e10x_iec_test.elf" "Project.hex";..\\gen_crc.bat;
```

3. 右击工程名称->Properties->C/C++build->settings->GNU ARM Cross C Linker->General, 在此添加链接文件，如*图 2-3. 添加链接文件*所示，修改之后点击 Apply and Close。

图 2-3. 添加链接文件



Properties for gd32e10x_iec_test

Settings

Configuration: Debug [ Active ]

GNU ARM Cross C Linker → General

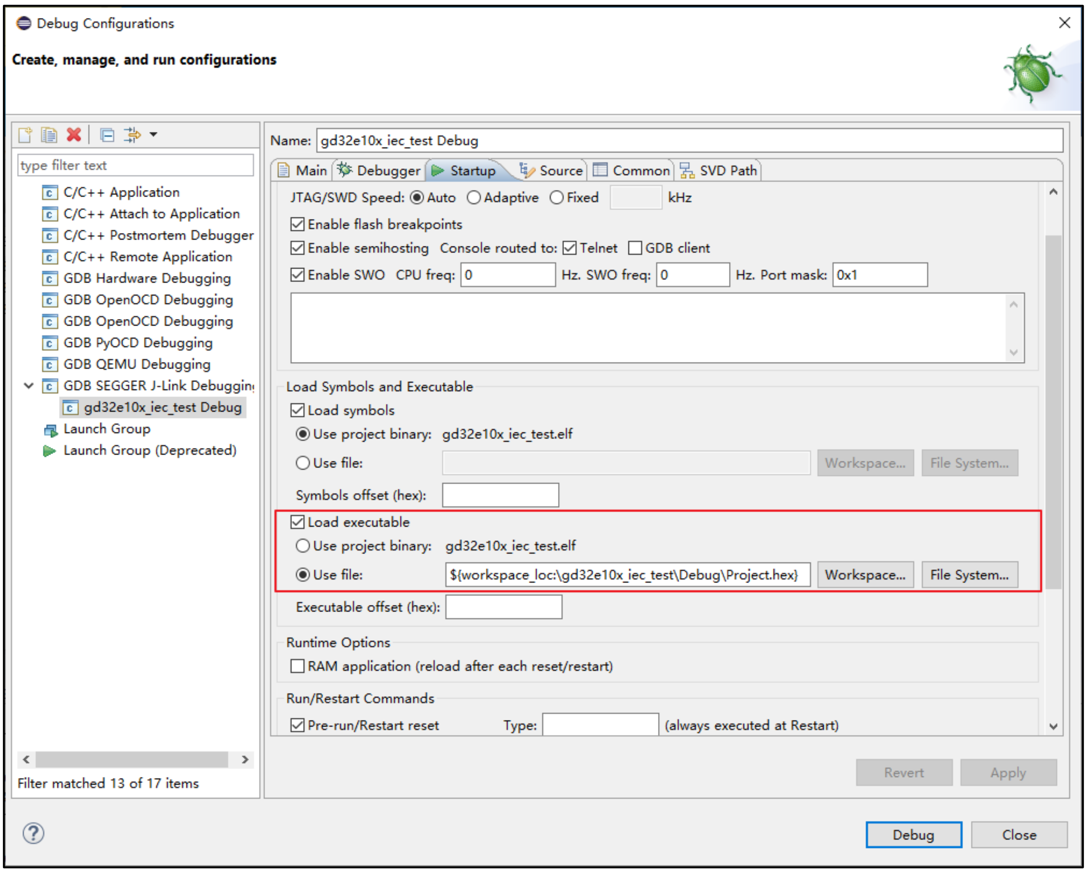
Script files (-T)

```
"${workspace_loc:/${ProjName}/ldscripts/gd32e10x_flash.ld}"
```

4. 在进行调试时，需要将可执行文件选择为批处理之后生成的 Project.hex 文件，在 Run->Debug Configuration->GDB SEGGER J-Link Debugging->gd32e10x_iec_test

Debug->Startup 界面的 Load excutable 中选择 Use file,批处理生成的 Project.hex 文件存储在工作空间中的 Debug 文件下，如*图2-4. 选择可执行文件*所示，修改之后点击 Apply.

**图 2-4. 选择可执行文件**



Debug Configurations

**Create, manage, and run configurations**

Name: gd32e10x_iec_test Debug

JTAG/SWD Speed: ● Auto ○ Adaptive ○ Fixed kHz

☑ Enable flash breakpoints

☑ Enable semihosting Console routed to: ☑ Telnet □ GDB client

☑ Enable SWO CPU freq: 0 Hz. SWO freq: 0 Hz. Port mask: 0x1

Load Symbols and Executable

☑ Load symbols

● Use project binary: gd32e10x_iec_test.elf

○ Use file: Workspace... File System...

Symbols offset (hex):

☑ Load executable

○ Use project binary: gd32e10x_iec_test.elf

● Use file: ${workspace_loc:\gd32e10x_iec_test\Debug\Project.hex} Workspace... File System...

Executable offset (hex):

Runtime Options

□ RAM application (reload after each reset/restart)

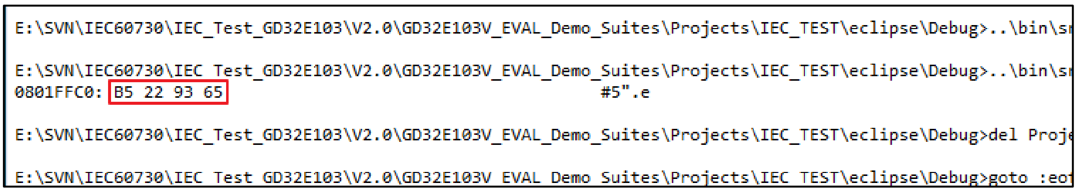
Run/Restart Commands

☑ Pre-run/Restart reset Type: (always executed at Restart)

# 3. 结果

配置完成后，点击编译，可以在 Build Output 窗口观察 build 信息，build 信息显示 CRC 值已存储在 0x0801FFF0 之后的位置,如*图 3-1. Build 信息*；配置 Debug 参数后，点击调试，并在 memory 观察窗口查询 0x0801FFF0 地址，如*图 3-2. memory 信息*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**



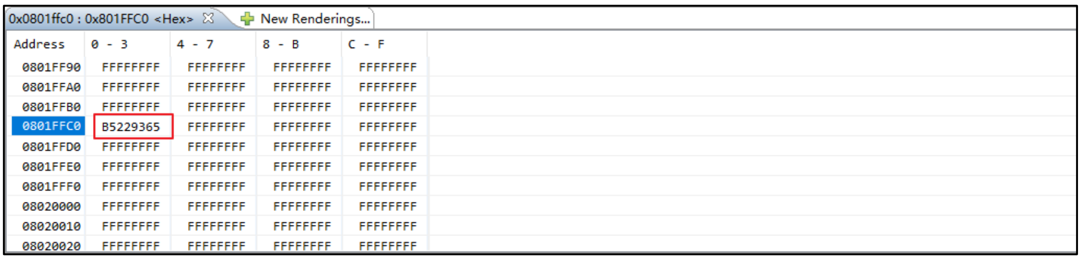
```
E:\SVN\IEC60730\IEC_Test_GD32E103\V2.0\GD32E103V_EVAL_Demo_Suites\Projects\IEC_TEST\eclipse\Debug>..\bin\s

E:\SVN\IEC60730\IEC_Test_GD32E103\V2.0\GD32E103V_EVAL_Demo_Suites\Projects\IEC_TEST\eclipse\Debug>..\bin\s
0801FFC0: B5 22 93 65                                        #5".e

E:\SVN\IEC60730\IEC_Test_GD32E103\V2.0\GD32E103V_EVAL_Demo_Suites\Projects\IEC_TEST\eclipse\Debug>del Proj

E:\SVN\IEC60730\IEC_Test_GD32E103\V2.0\GD32E103V_EVAL_Demo_Suites\Projects\IEC_TEST\eclipse\Debug>goto :eof
```

**图 3-2. memory 信息**



| Address | 0 - 3 | 4 - 7 | 8 - B | C - F |
|---|---|---|---|---|
| 0801FF90 | FFFFFFFF | FFFFFFFF | FFFFFFFF | FFFFFFFF |
| 0801FFA0 | FFFFFFFF | FFFFFFFF | FFFFFFFF | FFFFFFFF |
| 0801FFB0 | FFFFFFFF | FFFFFFFF | FFFFFFFF | FFFFFFFF |
| 0801FFC0 | B5229365 | FFFFFFFF | FFFFFFFF | FFFFFFFF |
| 0801FFD0 | FFFFFFFF | FFFFFFFF | FFFFFFFF | FFFFFFFF |
| 0801FFE0 | FFFFFFFF | FFFFFFFF | FFFFFFFF | FFFFFFFF |
| 0801FFF0 | FFFFFFFF | FFFFFFFF | FFFFFFFF | FFFFFFFF |
| 08020000 | FFFFFFFF | FFFFFFFF | FFFFFFFF | FFFFFFFF |
| 08020010 | FFFFFFFF | FFFFFFFF | FFFFFFFF | FFFFFFFF |
| 08020020 | FFFFFFFF | FFFFFFFF | FFFFFFFF | FFFFFFFF |

# 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.

© 2021 GigaDevice – All rights reserved