

GigaDevice Semiconductor Inc.

一种基于 **MDK** 实现的 **Lib** 库调用方案介绍

应用笔记

AN075

目录

目录.....	2
图索引.....	3
表索引.....	4
1. 前言.....	5
2. 应用的场景分析.....	6
2.1. 应用的场景.....	6
3. Demo 实现的方法.....	6
3.1. 概述.....	6
3.2. MPEG-2 软件算法工程创建.....	6
3.3. 分离 CRC 计算代码.....	11
3.4. 在新的工程中调用 CRC 算法.....	12
4. 版本历史.....	15

图索引

图 3-1. Template 目录修改	7
图 3-2. Mpeg2 目录内容.....	7
图 3-3. 代码 CRC 计算结果	10
图 3-4. 第三方软件 MPEG-2 CRC 计算结果	10
图 3-5. Linker 设置	11
图 3-6. map 文件.....	12
图 3-7. crc 库文件	13
图 3-8. 新工程目录结构	13
图 3-9. 基本的合作开发的框架	14

表索引

表 3-1. crc.h 代码单元格.....	7
表 3-2. crc32_mpeg2_init 代码单元格.....	8
表 3-3. crc32_mpeg2_calc 代码单元格	9
表 3-4. CRC 测试代码单元格	9
表 3-5. 分散加载文件代码单元格.....	12
表 4-1. 版本历史	15

1. 前言

本应用笔记是专门用于软件代码设计，旨在通过分散 image 文件来实现协作开发和减少代码下载量等用途。

本应用笔记将通过实现一个计算 CRC MPEG-2 的软件算法，来讲述分散 image 的方法，当然实现的方法不局限于本应用笔记所述的方法，客户也能够通过其它方法来达到相同的目的。

本应用笔记理论上适用于所有使用 MDK 开发的 GD32 MCU 产品。

2. 应用的场景分析

2.1. 应用的场景

在产品的应用开发过程中经常会遇到如下情景：

情景一，在比较大的应用代码开发中往往会出现绝大多数代码功能已经正常，有小部分代码需要不断调整，在调整该部分代码的时候会经历将整个应用代码重新编译和下载的过程，当整个应用比较大的时候（如 bin 文件大于 1MB）会使得整个代码下载的时间变得十分漫长，往往会超过 3 分钟的时长，此时如果只更新修改部分的代码，而不用修改其它代码将能极大的提高开发效率和改善开发体验。

情景二，一些算法开发商会将自己的开发的算法固化到芯片的固定位置，方案商会基于固化的算法开发自己的应用。

本应用笔记基于软件的 CRC 算法，提供了一种合作开发的方法。

3. Demo 实现的方法

3.1. 概述

CRC 即循环冗余校验码（Cyclic Redundancy Check）：是数据通信领域中最常用的一种查错校验码，其特征是信息字段和校验字段的长度可以任意选定。循环冗余检查（CRC）是一种数据传输检错功能，对数据进行多项式计算，并将得到的结果附在帧的后面，接收设备也执行类似的算法，以保证数据传输的正确性和完整性。

本小节将实现软件计算 CRC MPEG-2 的算法，并实现将该算法执行文件与主要调用代码的执行文件进行分离，来实现合作开发，由于本文主要在于讲述实现分离开发的方法，所以并不会详细介绍 CRC 算法的实现。

3.2. MPEG-2 软件算法工程创建

本节使用的软件工程将基于 GD32W51x 的固件库工程，开发板使用 GD32W515P-EVAL-V1.1。

首先从官网获取 GD32W51x 的固件库工程，本文使用的固件库工程为 V1.0.0 版本。获取成功后需先确认软硬件功能是否正常。

创建新的目录和源代码文件，如 [图 3-1. Template 目录修改](#) 在 Template 目录下创建 mpeg2 目录用于保存我们需要存放的 CRC 代码文件，[图 3-2. Mpeg2 目录内容](#) 显示了该文件夹下内容，该文件夹下目前只有 2 个文件，crc.c 和 crc.h。

图 3-1. Template 目录修改

名称	修改日期	类型	大小
IAR_Project	2022/7/4 10:29	文件夹	
Keil_Project	2022/7/4 10:29	文件夹	
mpeg2	2022/7/4 10:29	文件夹	
gd32w51x_it.c	2022/6/30 14:45	C 文件	4 KB
gd32w51x_it.h	2022/6/30 14:45	C/C++ Header	3 KB
gd32w51x_libopt.h	2022/6/30 14:45	C/C++ Header	3 KB
main.c	2022/6/30 16:12	C 文件	4 KB
main.h	2022/6/30 14:45	C/C++ Header	2 KB
readme.txt	2022/6/30 14:45	文本文档	2 KB
systick.c	2022/6/30 14:45	C 文件	3 KB
systick.h	2022/6/30 14:45	C/C++ Header	2 KB

图 3-2. Mpeg2 目录内容

名称	修改日期	类型	大小
crc.c	2022/6/30 16:13	C 文件	2 KB
crc.h	2022/6/30 16:13	C/C++ Header	1 KB

[表 3-1. crc.h 代码单元格](#)为 crc.h 中的代码，在 crc.h 中有两个函数的声明，这两个函数是在 crc.c 中实现的，供外部调用。

表 3-1. crc.h 代码单元格

```
#ifndef _CRC_H
#define _CRC_H

void crc32_mpeg2_init(unsigned int poly, unsigned int *crc_table);
unsigned int crc32_mpeg2_calc(unsigned int crc, unsigned int *crc_table, void* input,
int len);

#endif
```

函数 crc32_mpeg2_init 用来初始化 CRC 计算表，[表 3-2. crc32_mpeg2_init 代码单元格](#)为其实现。

表 3-2. crc32_mpeg2_init 代码单元格

```
/*!
 * \brief      generate crc MPEG-2 table
 * \param[in]  poly: crc polynomial, for MPEG-2 it is 0x4C11DB7
 * \param[in]  crc_table: a pointer to a polynomial table, at least size is 256
 * \param[out] crc_table: a pointer to generated polynomial table
 * \retval     none
 */
void crc32_mpeg2_init(unsigned int poly, unsigned int *crc_table)
{
    int i;
    int j;
    unsigned long c;

    for (i = 0; i < 256; i++) {
        c = i;
        c = i << 24;

        for (j = 0; j < 8; j++){
            if (c & 0x80000000){
                c = (c << 1) ^ poly;
            } else {
                c = ((c << 1));
            }
        }
        crc_table[i] = c;
    }
}
```

函数 `crc32_mpeg2_calc` 用来进行 CRC 计算，[表 3-3. `crc32_mpeg2_calc` 代码单元格](#)为其实现。

表 3-3. crc32_mpeg2_calc 代码单元格

```
/*!
 * \brief      calculate crc MPEG-2
 * \param[in]  crc: initial value, for MPEG-2 it is 0xFFFFFFFF
 * \param[in]  crc_table: a pointer to a polynomial table, at least size is 256
 * \param[in]  input: a pointer to need calculate data
 * \param[in]  len: length of need calculate data
 * \param[out] none
 * \retval    calculate result
 */
unsigned int crc32_mpeg2_calc(unsigned int crc, unsigned int *crc_table, void* input, int len)
{
    int i;
    unsigned char index;
    unsigned char* pch;
    pch = (unsigned char*)input;
    for (i = 0; i < len; i++) {
        index = (unsigned char)((crc >> 24) ^ *pch);
        crc = (crc << 8) ^ crc_table[index];
        pch++;
    }
    return crc;
}
```

接下来在 main.c 中添加软件 [表 3-4. CRC 测试代码单元格](#) 计算的测试代码 crc_test, 并在 main 函数中调用该函数, 该测试函数将对输入数据"123456789"计算其 MPEG-2 的 CRC 校验值。

表 3-4. CRC 测试代码单元格

```
unsigned int crc_table[256] = {0};

void crc_test(void)
{
    unsigned int crc;
    crc32_mpeg2_init(0x4C11DB7, crc_table);

    crc = 0xFFFFFFFF;
    crc = crc32_mpeg2_calc(crc, crc_table, "123456789", 9);
    printf("\r\n crc result is: 0x%08x\r\n", crc);
}
```

通过串口工具显示的计算结果如 [图 3-3. 代码 CRC 计算结果](#) 所示。

图 3-3. 代码 CRC 计算结果

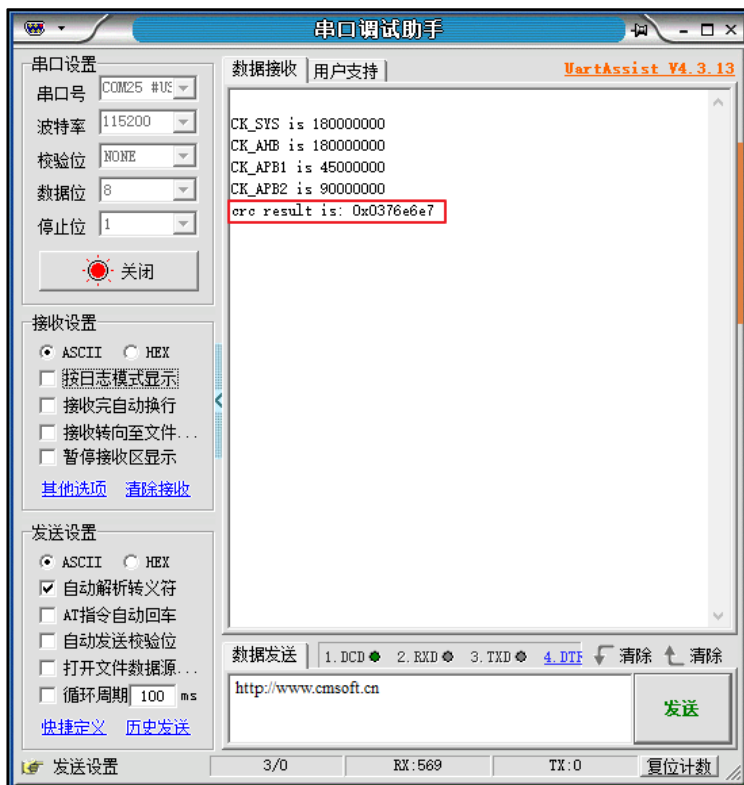


图 3-4. 第三方软件 MPEG-2 CRC 计算结果为第三方软件计算的结果，与我们设计的程序计算结果是一致的，由此能够知道设计的软件 MPEG-2 CRC 计算算法是正确的。

图 3-4. 第三方软件 MPEG-2 CRC 计算结果

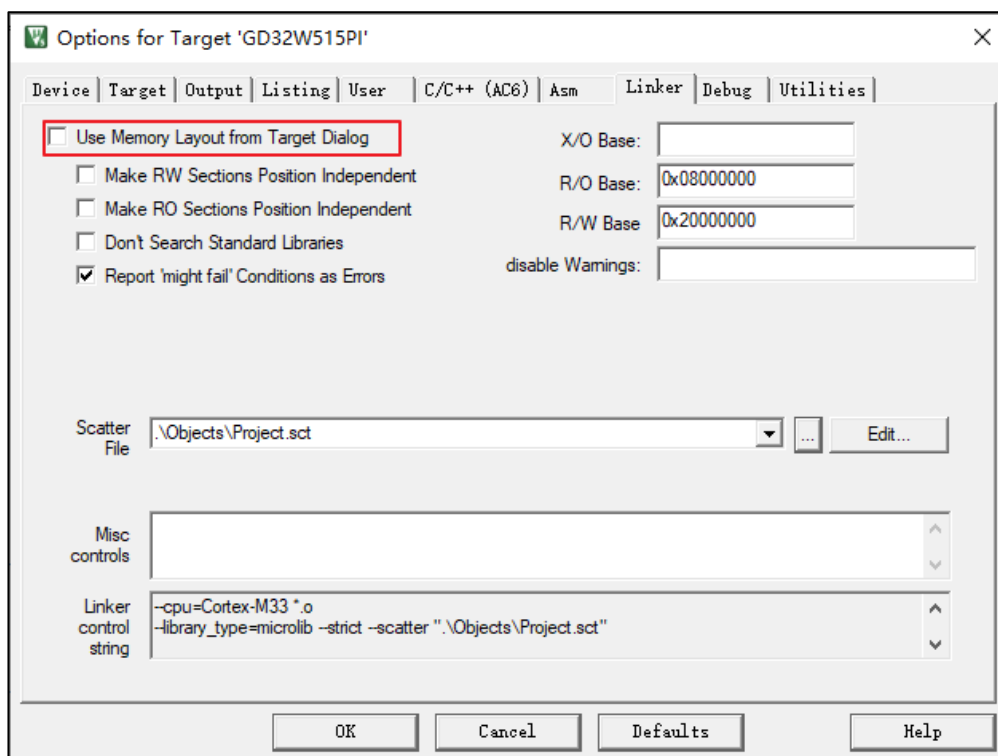


3.3. 分离 CRC 计算代码

本节内容将在上一小节内容的基础上，将 MPEG-2 CRC 计算的代码分散加载到指定的位置，其它工程就可以直接调用指定位置的代码来实现 CRC 计算。

打开工程的选项页面，[图 3-5. Linker 设置](#)去掉 Linker 下默认的选项，使用分散加载文件来分配工程的内存。

图 3-5. Linker 设置



在默认的分散加载文件中按[表 3-5. 分散加载文件代码单元格](#)做如下修改，将第一个加载区域 LR_IROM1 的大小改为 0x0000FFFF，将其第一个执行区 ER_IROM1 大小也改为 0x0000FFFF，同时新建一个 LR_CODE_2 的加载区，其起始地址紧接着 LR_IROM1，大小也是 0x0000FFFF，在其中定义一个执行区 CRC_TEST 起始地址与 LR_CODE_2 一致，并将 crc.o 所有的代码都放置到该区域。

表 3-5. 分散加载文件代码单元格

```

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

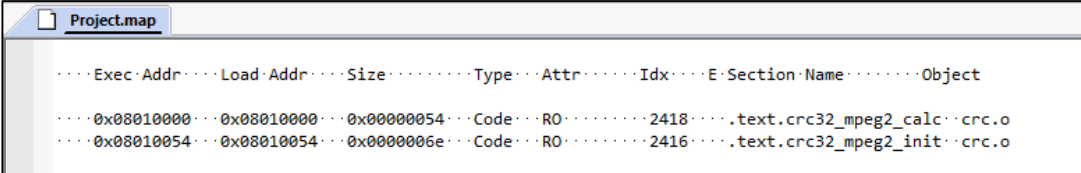
LR_IROM1 0x08000000 0x0000FFFF { ; load region size_region
  ER_IROM1 0x08000000 0x0000FFFF { ; load address = execution address
    *.o (RESET, +First)
    *(InRoot$$Sections)
    .ANY (+RO)
    .ANY (+XO)
  }
  RW_IRAM1 0x20000000 0x00070000 { ; RW data
    .ANY (+RW +ZI)
  }
}

LR_CODE_2 0x08010000 0xFFFF {
  CRC_TEST 0x08010000 {
    crc.o
  }
}
}

```

此时再编译工程，通过查看工程编译后的 map 文件，能够看到在 [图 3-6. map 文件](#) 中 crc.o 里面定义的两个函数 `crc32_mpeg2_calc` 和 `crc32_mpeg2_init` 分别位于 `0x08010000` 和 `0x08010054` 位置，与我们在分散加载文件中定义的一致。将程序下载到开发板中能够通过串口助手看到打印的结果与未修改之前的一致。此时芯片 `0x08010000` 开始的位置就存放了能够进行 mpeg-2 计算的可执行代码，如果不擦除，代码会一直存放在此处，其它程序都能够调用该函数。

图 3-6. map 文件



Exec Addr	Load Addr	Size	Type	Attr	Idx	E	Section Name	Object
0x08010000	0x08010000	0x00000054	Code	RO	2418		.text.crc32_mpeg2_calc	crc.o
0x08010054	0x08010054	0x0000006e	Code	RO	2416		.text.crc32_mpeg2_init	crc.o

3.4. 在新的工程中调用 CRC 算法

现在我们能够创建其它工程并通过调用 `0x08010000` 开始处的代码来实现 CRC 计算，不过为了调用的方便我们将创建一个库文件来参与新工程的编译，而不是直接通过地址来调用，本节

的内容将着重来描述实现的方法。

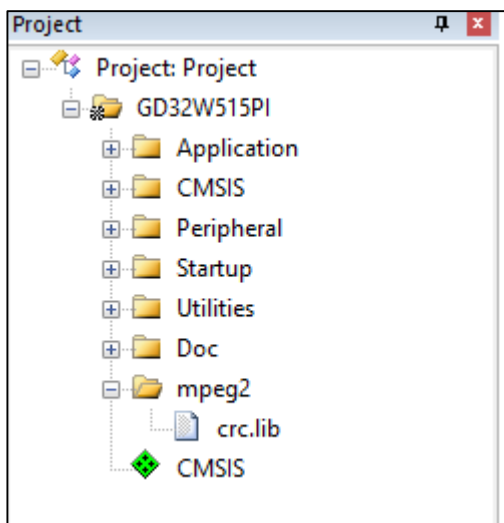
首先重新打开一份固件库工程，同样的创建一个 mpeg2 文件夹，在工程中将上一小结的 crc.h 文件拷贝到该目录下，创建一个 crc.lib 的库文件，在其中添加 [图 3-7. crc 库文件](#) 所示内容，也就是将上一节中函数的地址和函数名一一对应添加进入。

图 3-7. crc 库文件

```
0x08010000 T crc32_mpeg2_calc  
0x08010054 T crc32_mpeg2_init
```

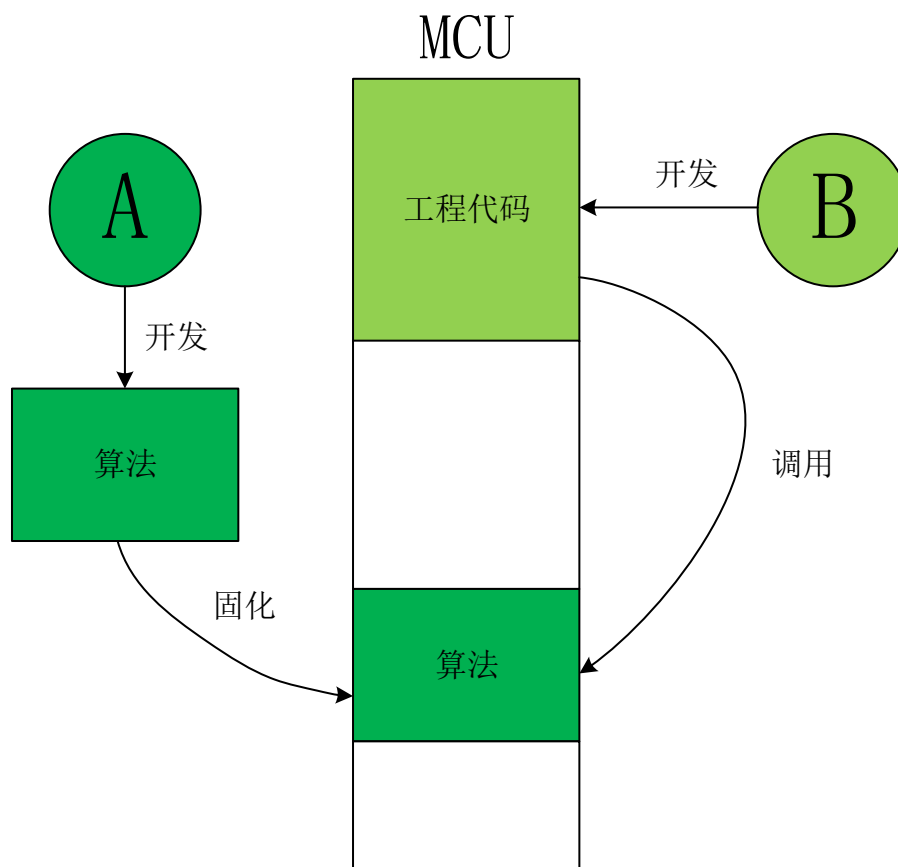
将库添加到工程中，[图 3-8. 新工程目录结构](#) 为新工程的目录结构，在 main.c 中添加 [表 3-4. CRC 测试代码单元格](#) 所示的 crc_test 测试函数，编译下载到开发板中，能够发现串口输出计算的 crc 值依然是正确的。

图 3-8. 新工程目录结构



如果我们将软件 crc 计算的代码开发视作 A 开发的功能代码或者算法代码，A 在完成开发后能够将其下载并固化到芯片中，供 B 继续使用和开发，至此可以构成一个 [图 3-9. 基本的合作开发的框架](#) 所示的最基本的合作开发的框架。

图 3-9. 基本的合作开发的框架



注意：将代码固化到芯片中的方法有很多，如可通过镜像工具将算法生成 hex 文件由 B 固化到芯片中，并不局限于上面描述的方法。

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	2022 年 6 月 10 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 its 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 its 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.