

32 位微控制器

HC32L110 / HC32F003 / HC32F005 系列的 FLASH

适用对象

| 系列 | 产品型号 |
|-----------------|--------------|
| HC32L110 | HC32L110C6UA |
| | HC32L110C6PA |
| | HC32L110C4UA |
| | HC32L110C4PA |
| | HC32L110B6PA |
| | HC32L110B4PA |
| HC32F003 | HC32F003C4UA |
| | HC32F003C4PA |
| HC32F005 | HC32F005C6UA |
| | HC32F005C6PA |
| | HC32F005D6UA |

目 录

| | | |
|----------|------------------------------|-----------|
| 1 | 摘要 | 3 |
| 2 | FLASH 简介 | 3 |
| 3 | FLASH | 4 |
| 3.1 | 简介 | 4 |
| 3.2 | 说明 | 4 |
| 3.2.1 | 寄存器介绍 | 4 |
| 3.2.2 | 工作流程介绍 | 5 |
| 4 | 样例代码 | 7 |
| 4.1 | 代码介绍 | 7 |
| 4.2 | 代码运行 | 8 |
| 5 | 总结 | 10 |
| 6 | 版本信息 & 联系方式 | 11 |

1 摘要

本篇应用笔记主要介绍如何使用 HC32L110 / HC32F003 / HC32F005 系列的 FLASH 进行擦除、编程和读取。

2 FLASH 简介

什么是 FLASH?

闪存器件的一种，闪存是一种非易失性（Non-Volatile）内存，在没有电流供应的条件下也能够长久地保持数据，其存储特性相当于硬盘，这项特性正是闪存得以成为各类便携型数字设备的存储介质的基础。

（引自‘百度百科’，‘互动百科’，‘维基百科’）

FLASH 特点?

Flash 闪存是非易失存储器,可以对称为块的存储器单元块进行擦写和再编程。任何 flash 器件的写入操作只能在空或已擦除的单元内进行,所以大多数情况下,在进行写入操作之前必须先执行擦除。

FLASH 的应用?

FLASH 被广泛用于移动存储、MP3 播放器、数码相机、掌上电脑等新兴数字设备中。

3 FLASH

3.1 简介

本设备包含一颗 32kByte 容量的 FLASH 存储器，共划分为 64 个 Sector，每个 Sector 容量为 512Byte。本模块支持对该存储器的擦除、编程以及读取操作。此外，本模块支持对 FLASH 存储器擦写的保护，以及控制寄存器的写保护。

3.2 说明

本小节介绍 HC32L110 / HC32F003 / HC32F005 系列的 FLASH 控制器模块，包括寄存器和工作流程。

本 FLASH 控制器支持对 eFLASH 的 Byte（8bits）、Half-word（16bits）、Word（32bits）三种位宽读写操作。注意，Byte 操作的地址必须按 Byte 对齐，Half-word 操作的目标地址必须按 Half-word 对齐（地址最低位为 1'b0），Word 操作的地址必须按 Word 对齐（地址最低两位为 2'b00）。如果读写操作的地址没有按照位宽规定对齐，该操作无效，并且系统会进入 Hard Fault 出错中断。

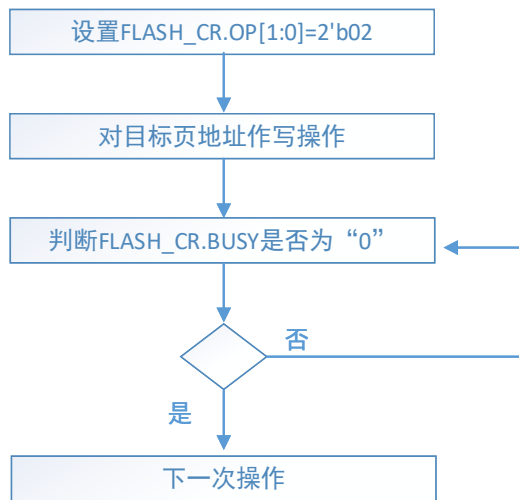
3.2.1 寄存器介绍

对于 FLASH 的操作主要通过以下寄存器进行：

| 缩写 | 寄存器名称 |
|---------------|-----------------------------|
| FLASH_TNVS | Tnvs时间参数 |
| FLASH_TPGS | Tpgs时间参数 |
| FLASH_TPROG | Tprog时间参数 |
| FLASH_TSERASE | Tserase时间参数 |
| FLASH_TMERASE | Tmerase时间参数 |
| FLASH_TPRCV | Tprcv时间参数 |
| FLASH_TSRCV | Tsrcv时间参数 |
| FLASH_TMRCV | Tmrcv时间参数 |
| FLASH_CR | 控制寄存器 |
| FLASH_IFR | 中断标志寄存器 |
| FLASH_ICLR | 中断标志清除寄存器 |
| FLASH_BYPASS | (0x5a5a-0xa5a5) Bypass序列寄存器 |
| FLASH_SLOCK | Sector擦写保护寄存器 |

3.2.2 工作流程介绍

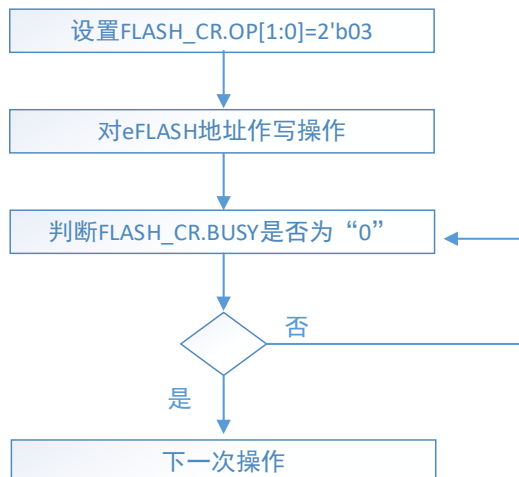
Sector 擦除



注意:

1. 控制器忽略目标地址的低9位，只要目标地址落在该页地址范围之内即可。
2. 该写操作用于触发页擦除操作，所写数据也会被控制器忽略。
3. 如果当前擦除指令是在eFLASH内执行，则CPU取值会停止，硬件自动等待eFLASH的BUSY状态结束
4. 如果当前擦除指令是在RAM内执行，则CPU取值不会停止，在对eFLASH进行任何操作之前，软件必须判断eFLASH的BUSY状态是否结束

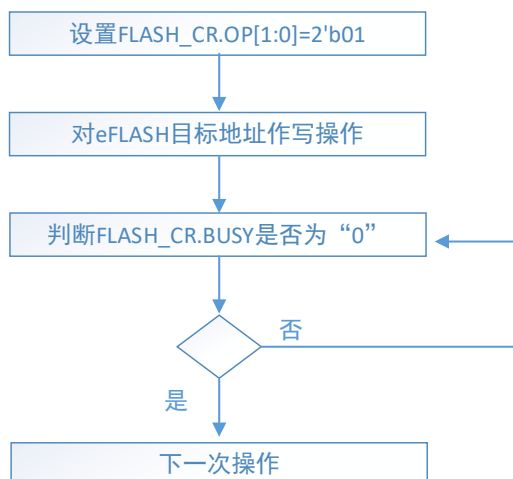
Chip 擦除



注意:

1. 控制器忽略目标地址的低15位，只要目标地址落在eFLASH地址范围之内即可。
2. 该写操作用于触发页擦除操作，所写数据也会被控制器忽略。
3. 如果当前擦除指令是在eFLASH内执行，则CPU取值会停止，硬件自动等待eFLASH的BUSY状态结束
4. 如果当前擦除指令是在RAM内执行，则CPU取值不会停止，在对eFLASH进行任何操作之前，软件必须判断eFLASH的BUSY状态是否结束

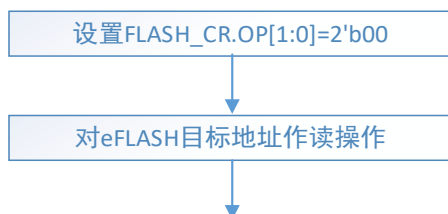
写操作



注意:

1. 如果当前擦除指令是在eFLASH内执行，则CPU取值会停止，硬件自动等待eFLASH的BUSY状态结束
2. 如果当前擦除指令是在RAM内执行，则CPU取值不会停止，在对eFLASH进行任何操作之前，软件必须判断eFLASH的BUSY状态是否结束

读操作

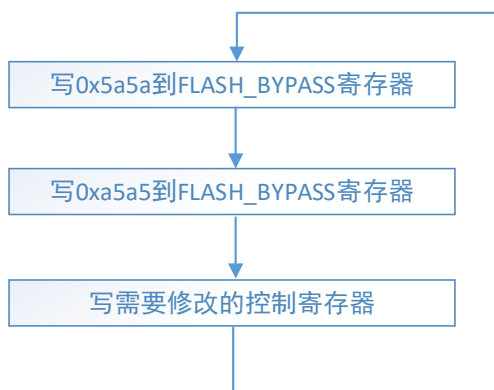


注意:

1. 第一步设置FLASH_CR.OP[1:0]的步骤实际可以省略，无论FLASH_CR.OP[1:0]为何值，读操作都可以进行

寄存器写保护

本模块的控制器屏蔽普通的写操作，必须用写序列方式才能修改。具体操作步骤如下图所示：



注意:

- 一 写 0x5a5a 和写 0xa5a5 这两步操作之间不可插入任何写操作，否则该 Bypass 序列会失效，需要重新写入 0x5a5a-0xa5a5 序列。

4 样例代码

4.1 代码介绍

用户可以根据上述的工作流程编写自己的代码来学习验证该模块，也可以直接通过小华半导体的网站下载到 FLASH 的样例代码直接使用 FLASH 驱动库提供的 API 函数进行编码及校验的应用。

以下部分简要介绍该代码的各个部分的功能：

1) FLASH 数据声明及初始化：

```
//FLASH TEST DATA INIT
uint32_t    u32Addr = 0x7000;
uint8_t     u8Data  = 0x5a;
uint16_t    u16Data = 0x5a5a;
uint32_t    u32Data = 0x5a5a5a5a;
```

2) FLASH 初始化及 Sector 擦除编码：

```
Flash_Init(FlashInt, 0);

Flash_SectorErase(u32Addr);
```

3) FLASH 编程及校验：

```
if (Ok == Flash_WriteByte(u32Addr, u8Data))
{
    if (((volatile uint8_t*)u32Addr) == u8Data)
    {
        enResult = Ok;
    }
    else
    {
        return enResult;
    }
}
else
{
    return enResult;
}
```

通过以上代码即可完成一次 FLASH 的擦除、编程及读取。

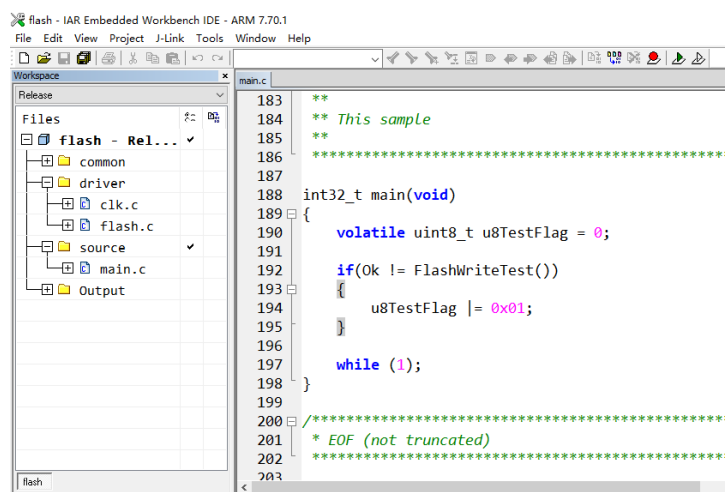
4.2 代码运行


用户可以通过小华半导体的网站下载到 FLASH 样例代码，并配合评估用板运行相关代码学习使用 FLASH 模块。

以下部分主要介绍如何在评估板上运行 FLASH 样例代码并观察结果：

- 确认安装正确的 IAR（或 Keil,此处使用 IAR 做样例说明，Keil 中操作方法类似）工具（请从小华半导体完整下载相应的安装包，并参考用户手册进行安装）。
- 从小华半导体网站下载 FLASH 样例代码。
- 下载并运行样例代码：

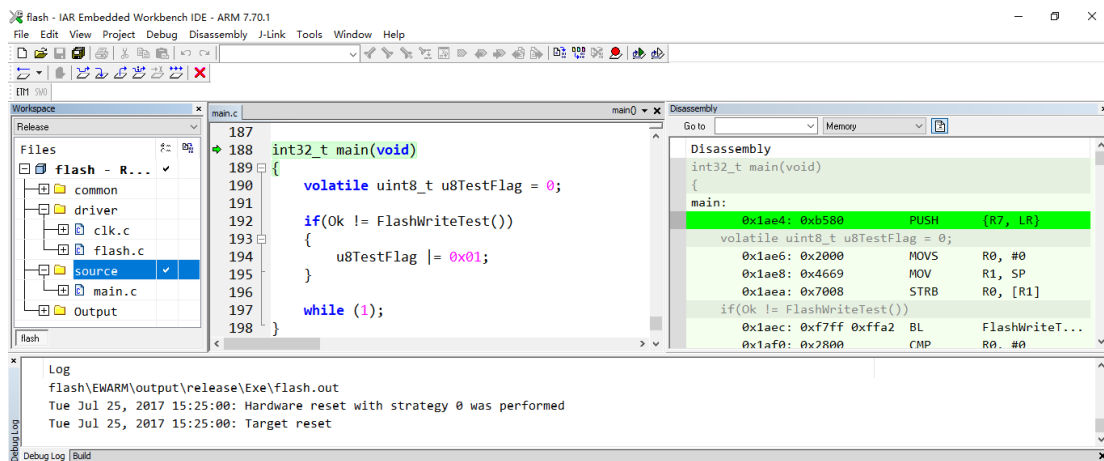
1) 打开 FLASH 工程，并打开‘main.c’如下视图：



2) 点击  重新编译链接整个项目。

3) 点击  将代码下载到评估板上。

4) 可以看见类似如下的视图:

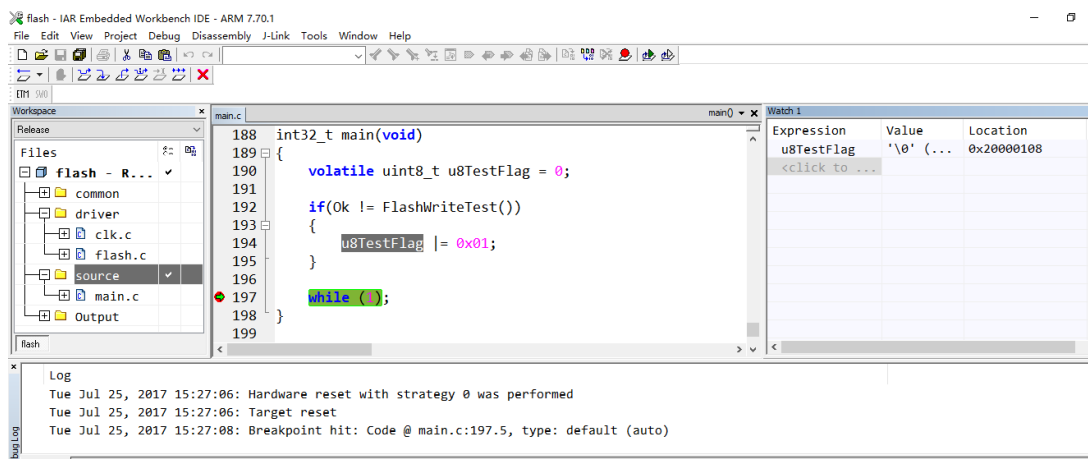


5) 在‘main(void)’的最后一行设置断点:

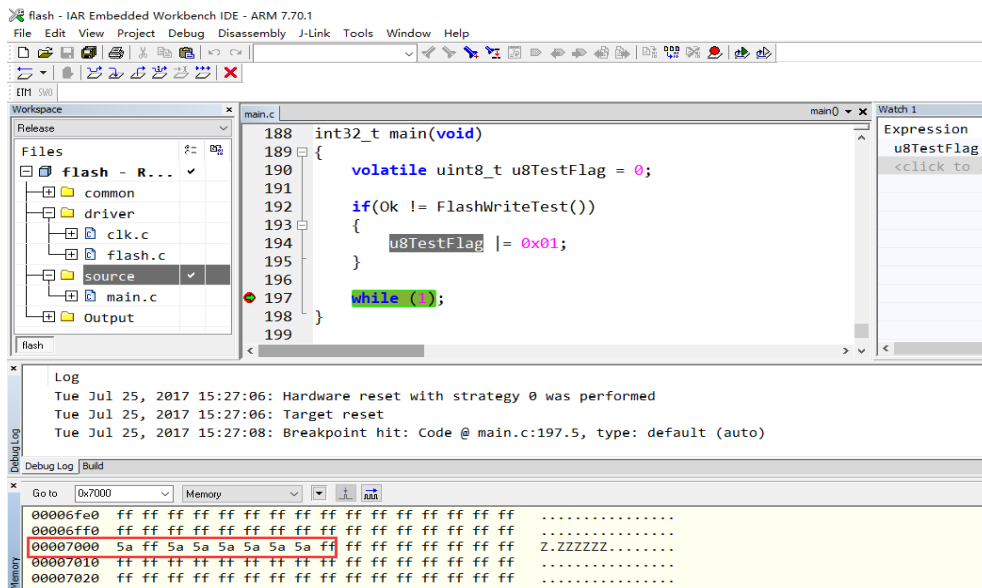
6) 点击“View -> Watch -> Watch1”打开一个‘watch1’窗口,并添加‘u8TestFlag’变量来观测其数值。

7) 点击  运行。

8) 代码运行并会停止在‘main(void)’的断点处,如果‘u8TestFlag = 0’,表示编码及校验功能正确执行,如下图。



9) 也可以点击“View -> Memory”打开一个‘Memory’窗口,观测程序编程区域的数据, 如下图。



10) 运行完毕后可以关闭项目文件。

11) 用户亦可通过修改代码中 FLASH 的测试数据(数据及长度)及操作地址(区域)来进一步学习 FLASH 模块的功能。

5 总结

以上章节简要介绍了 HC32L110 / HC32F003 / HC32F005 系列的 FLASH，并详细说明了 FLASH 模块的寄存器及操作流程，演示了如何使用相关的样例代码进行擦除、编程及读取操作，在实际开发中用户可以根据自己的需要配置和使用 FLASH 的擦除、编程及读取功能。

6 版本信息 & 联系方式

| 日期 | 版本 | 修改记录 |
|-----------|--------|-------------|
| 2018/6/4 | Rev1.0 | 初版发布。 |
| 2022/7/15 | Rev1.1 | 公司 Logo 更新。 |
| | | |



如果您在购买与使用过程中有任何意见或建议，请随时与我们联系。

Email: mcu@xhsc.com.cn

网址: <http://www.xhsc.com.cn>

通信地址: 上海市浦东新区中科路 1867 号 A 座 10 层

邮编: 201203

