

32 位微控制器

HC32M120 系列的嵌入式 FLASH

适用对象

系列	产品型号
HC32M120	HC32M120J6TB
	HC32M120F6TB

目 录

1	摘要	3
2	FLASH 简介	3
3	HC32M120 系列的 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	11
4.1	代码介绍	11
4.2	代码运行	13
5	版本信息 & 联系方式	15

1 摘要

本篇应用笔记主要介绍如何使用 HC32M120 系列芯片的嵌入式 FLASH 读写数据。

2 FLASH 简介

什么是 FLASH?

FLASH 接口通过系统总线对 FLASH 进行访问，该接口可对 FLASH 执行编程、擦除和全擦除操作；通过指令预取加速代码执行。

FLASH 的重要特征?

FLASH 读、编程、扇区擦除和全擦除操作，总线 4Bytes 预取值，支持安全保护。

3 HC32M120 系列的 FLASH

3.1 简介

FLASH 接口通过系统总线对 FLASH 进行访问，该接口可对 FLASH 执行编程、扇区擦除和全擦除操作；通过指令预取加速代码执行。

3.2 说明

FLASH 读、编程、扇区擦除和全擦除操作。

总线 4Bytes 预取值。

支持安全保护。

容量为 32Kbytes，分为 64 个扇区，每个扇区为 512Bytes。

编程单位为 1Byte、Half Word、Word，擦除单位为 512Bytes。

32 位宽数据读取。

3.2.1 寄存器介绍

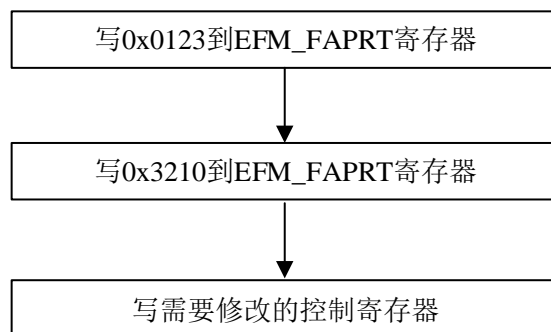
- 1) EFM_FAPRT: 访问 EFM 寄存器保护寄存器。
- 2) EFM_FSTP: FLASH 停止寄存器。
- 3) EFM_FRMC: 读模式寄存器。可配置插入等待周期，缓存功能、预取指功能等。
- 4) EFM_FWMC: 擦写模式寄存器。配置编程擦除模式。
- 5) EFM_FSR: 状态寄存器。查看 FLASH 状态，结束标志、错误标志等。
- 6) EFM_FSCLR: 状态清除寄存器。
- 7) EFM_FITE: 中断许可寄存器。配置操作结束或错误中断许可。
- 8) EFM_FPMTSW: FLASH 窗口保护起始地址寄存器。
- 9) EFM_FPMTEW: FLASH 窗口保护结束地址寄存器。
- 10) EFM_UQID0~2: Unique ID 寄存器。

3.2.2 工作流程介绍

3.2.2.1 寄存器解除保护和写保护

本模块的寄存器受 EFM_FAPRT 寄存器保护，当处于保护状态，屏蔽普通的写操作。

解除保护的步骤如下图：



在解除保护的条件下，对 EFM_FAPRT 寄存器写任意值，EFM 寄存器再次进入保护状态。

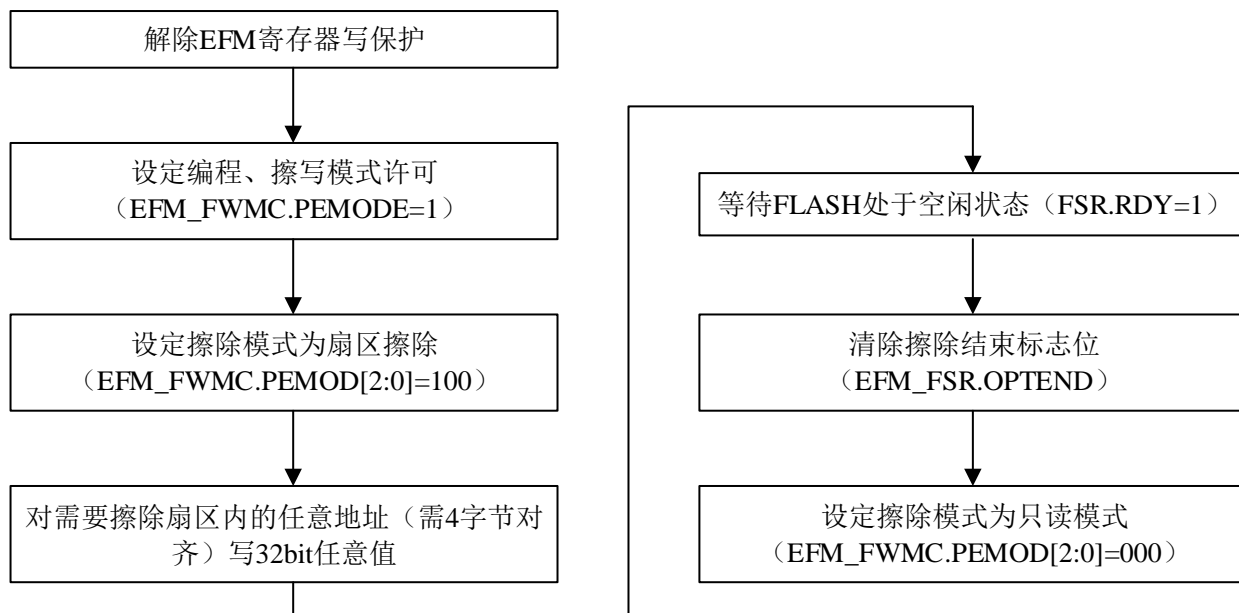
注意：

- 在实际应用中，发现寄存器值未写入成功，应首先检查 EFM 寄存器访问保护是否有效，保护有效时，EFM_FAPRT 寄存器值读数为 0x00000000。

3.2.2.2 扇区擦除

EFM 提供扇区擦除和全擦除两种擦除方式。对 FLASH 进行扇区擦除操作后，该扇区内地址（512Bytes 空间）数据刷新为全 F。

扇区擦除的设定步骤如下：



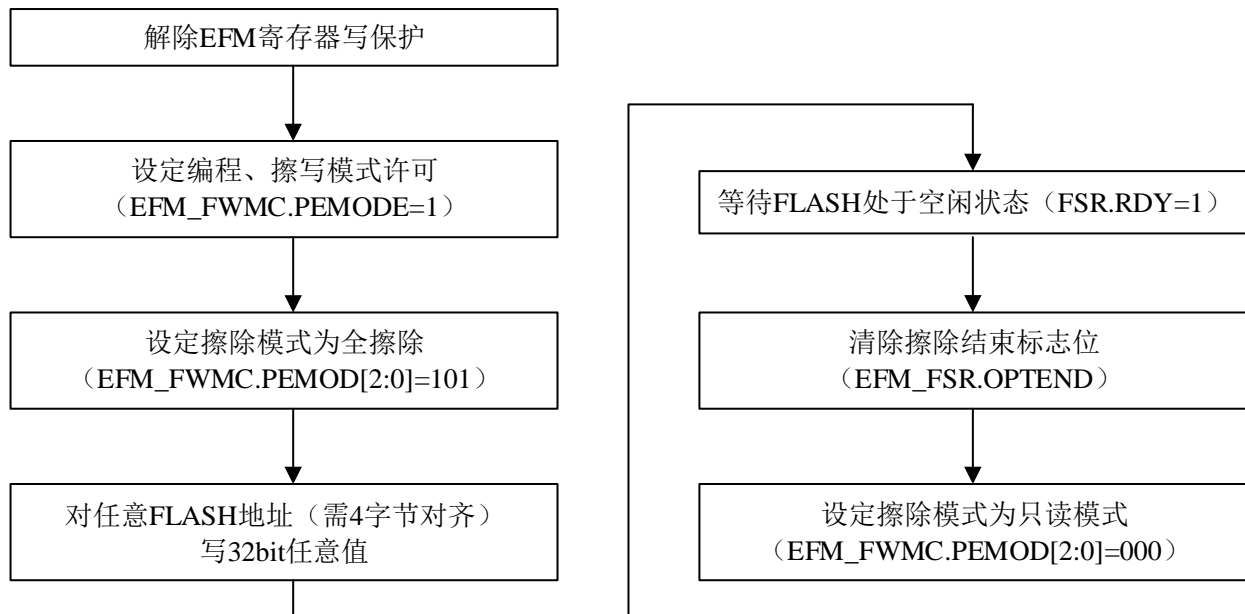
注意：

- EFM_FWMC.PEMODE 是对 EFM_FWMC.PEMOD[2:0]设定的许可；
- 擦除完成后需把 EFM_FWMC.PEMOD[2:0]设定为只读模式，预防对 FLASH 误操作，导致整个扇区擦除。

3.2.2.3 全擦除

EFM 提供扇区擦除和全擦除两种擦除方式。对 FLASH 进行全擦除操作后整个 FLASH 区域所有地址数据刷新为全 1。

全擦除的设定步骤如下：

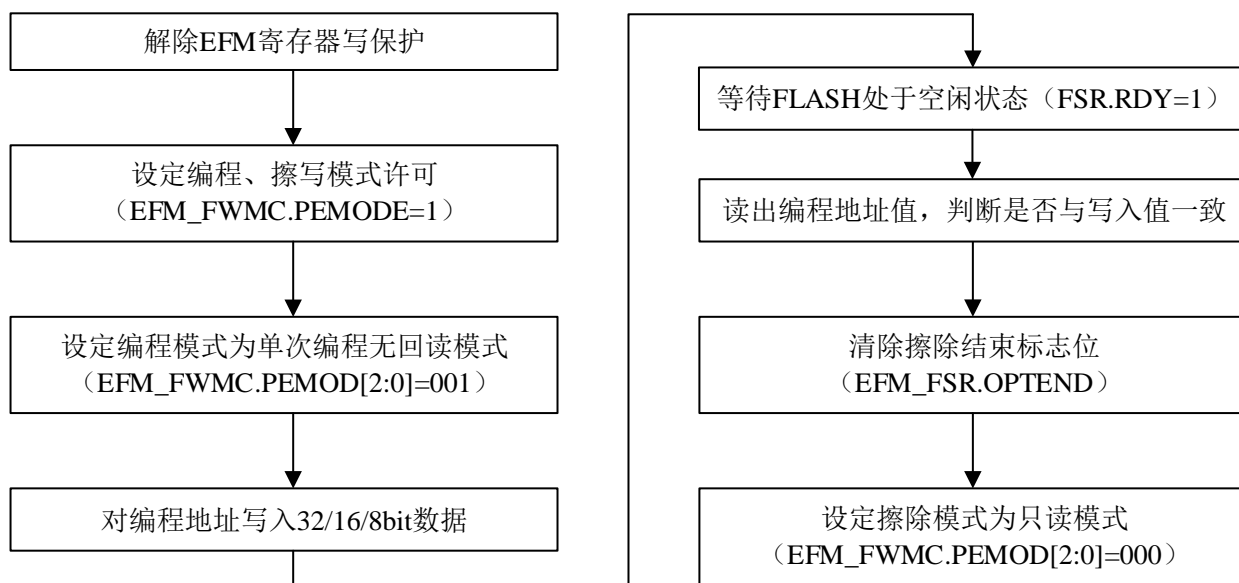


注意：

- 擦除完成后需把 EFM_FWMC.PEMOD[2:0] 设定为只读模式，预防对 FLASH 误操作，导致整个扇区擦除。

3.2.2.4 单次编程无回读

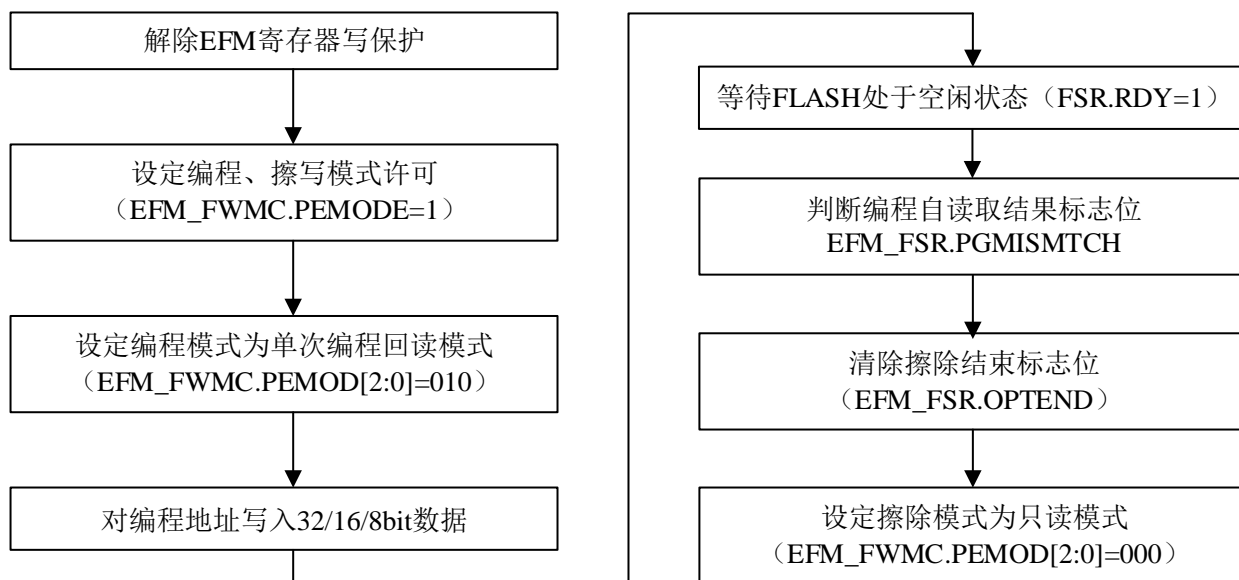
单编程无回读模式设定步骤如下：



3.2.2.5 单次编程回读

单编程回读模式是指编程结束后自动读取编程地址并和写入数据对比，输出判断一致标志位 EFM_FSR.PGMISMTCH。

单编程回读模式设定步骤如下：



注意：

- EFM_FSR.PGMISMTCH 为 0，表示编程成功，为 1 表示该 FLASH 地址已遭破坏，永废弃。

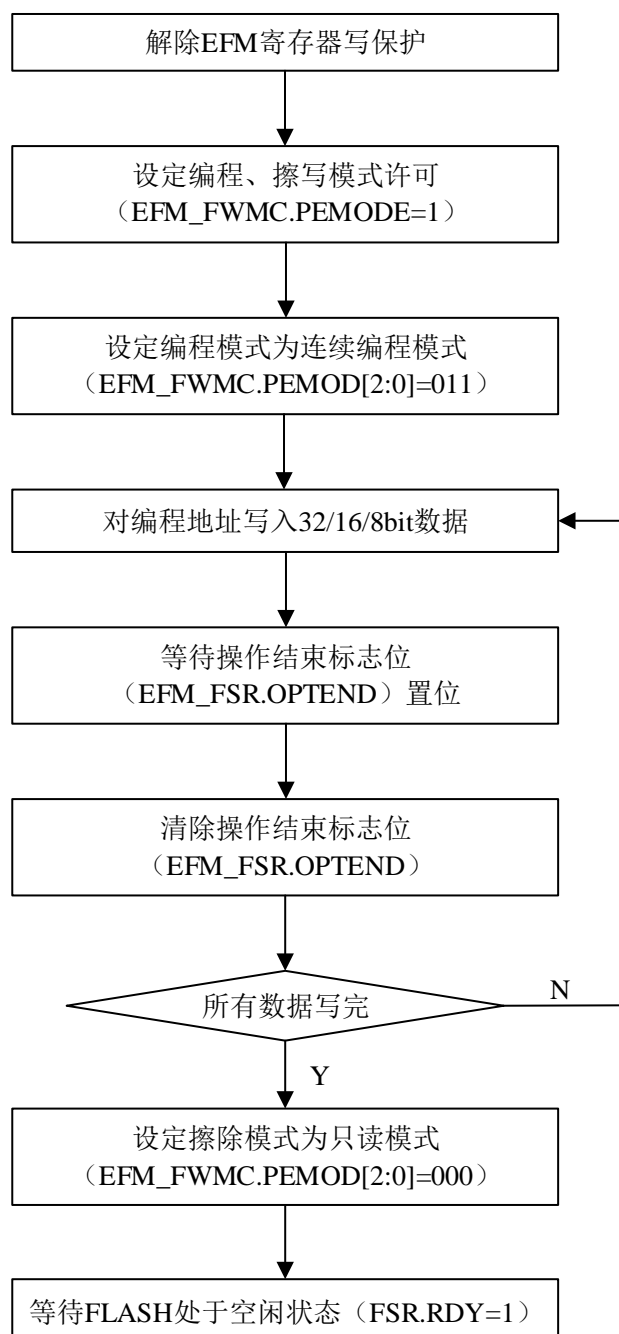
3.2.2.6 连续编程

当连续对 FLASH 地址进行编程时，推荐使用连续编程模式。连续编程模式比单编程模式可以节约时间 50% 以上。

连续编程设定步骤如下：

注意：

- 在 FLASH 连续编程期间，如果对 FLASH 进行读操作，将会读到不定值。



3.2.2.7 总线保持功能

通过设定寄存器 EFM_FWMC.BUSHLCTL 位，可设定 FLASH 编程、擦除期间，总线处于保持还是释放状态。

FLASH 编程、擦除指令在 FLASH 上执行时，该控制位必须设定为 0。擦除指令在 FLASH 以外空间（例如 RAM）执行时，可根据需求自由设定。

当 EFM_FWMC.BUSHLCTL 为 1（即 FLASH 编程、擦除期间，总线释放状态）时，在编程（连续编程除外）、擦除结束前（EFM_FSR.RDY=1）对 FLASH 的读写访问将会被忽视，标志位 EFM_FSR.BUSCOLERR 位置位。

4 样例代码

4.1 代码介绍

用户可根据上述的工作流程编写自己的代码来学习验证该模块，也可以直接通过华大半导体的网站下载到设备驱动库（Device Driver Library, DDL）的样例代码并使用其中的 FLASH 的样例进行验证。

以下部分简要介绍本 AN 基于 DDL 的 FLASH 模块样例 efm_base 代码所涉及的各项配置。

1) 解除 FLASH 寄存器保护：

```
/* Unlock EFM. */  
EFM_Unlock();
```

2) 使能 FLASH：

```
/* Enable flash. */  
EFM_Cmd(Enable);
```

3) 等待 FLASH ready：

```
/* Wait flash ready. */  
while(Set != EFM_GetFlagStatus(EFM_FLAG_RDY));
```

4) 扇区擦除：

```
/* Erase sector 10. */  
EFM_SectorErase(EFM_SECTOR10_ADRR)
```

5) 按 Byte 编程 FLASH：

```
u32Addr = EFM_SECTOR10_ADRR;  
  
if(Ok != EFM_ProgramByte(u32Addr, u8Data))  
{  
    LED0_ON();  
    LED1_OFF();  
}
```

6) 按 Half Word 编程 FLASH:

```
if(Ok != EFM_ProgramHalfWord (u32Addr, u16Data))
{
    LED0_ON();
    LED1_OFF();
}
```

7) 按 Word 编程 FLASH:

```
if(Ok != EFM_ProgramWord (u32Addr, u32Data))
{
    LED0_ON();
    LED1_OFF();
}
```

8) 锁 FLASH 寄存器:

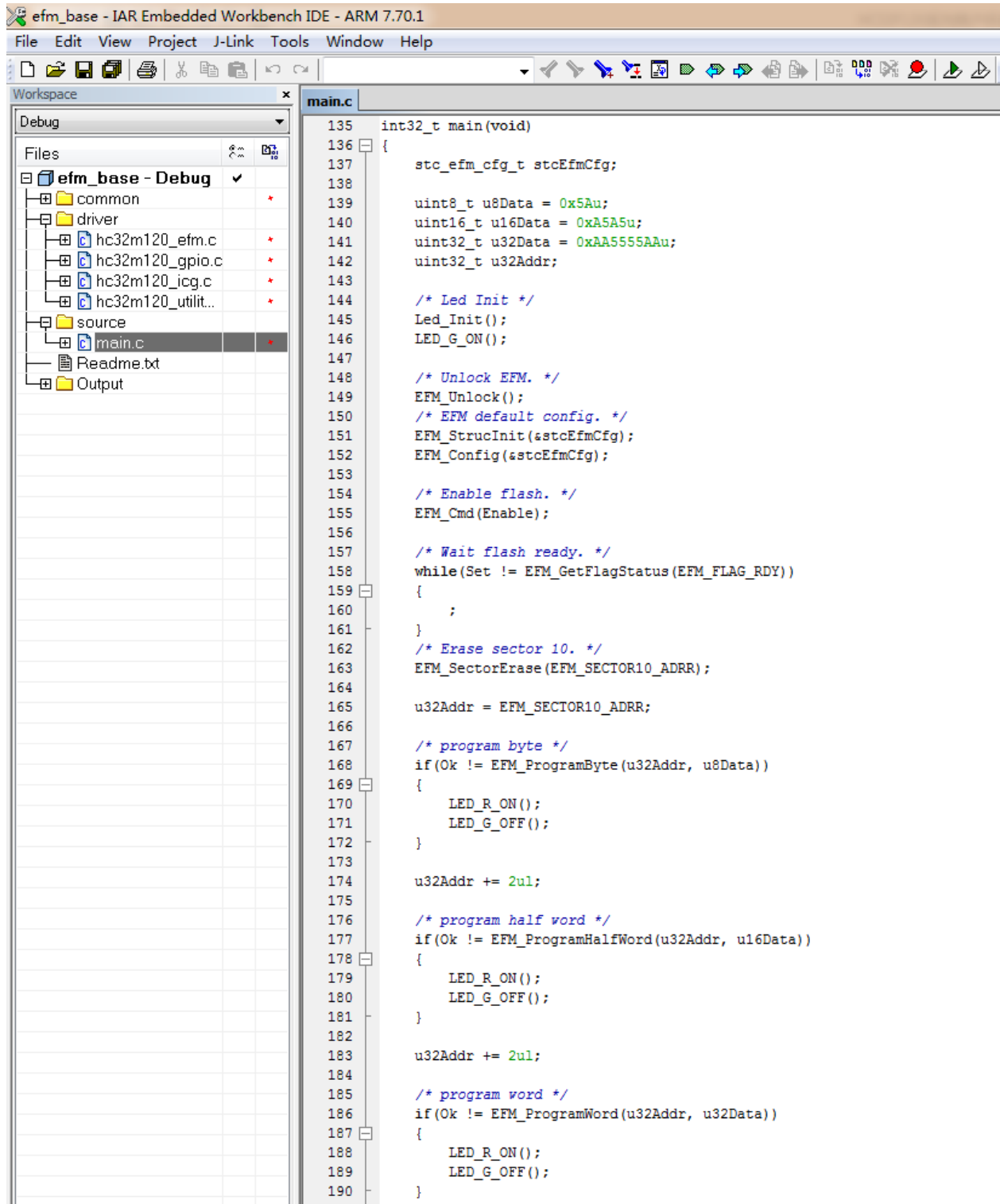
```
/* Lock EFM. */
EFM_Lock();
```



4.2 代码运行

用户可以通过华大半导体的网站下载到 HC32M120 的 DDL 的样例代码（efm_base），并配合评估用板（STK_HC32M120_LQFP48_050_V11）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 EWARM v7.7 工具（请从 IAR 官方网站下载相应的安装包，并参考用户手册进行安装）。
- 从华大半导体网站下载 HC32M120 DDL 代码。
- 下载并运行 efm\efm_base\中的工程文件：
 - 1) 打开 efm_base \工程，并打开 ‘main.c’ 如下视图：



- 2) 点击  重新编译整个项目；
- 3) 点击  将代码下载到评估板上，全速运行。
- 4) 可观察到绿灯亮。

也可以查看 memory，看对应地址是否写入预期值。

5 版本信息 & 联系方式

日期	版本	修改记录
2019/12/10	Rev1.0	初版发布



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@hdsc.com.cn

网址: www.hdsc.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

