FRDMK64F SD Bootloader

目录

[1、介绍 1](#_Toc21419994)

[2、Bootloader的实现 1](#_Toc21419995)

[3、内存重分配 6](#_Toc21419996)

[4、运行例程 8](#_Toc21419997)

[5、参考 8](#_Toc21419998)

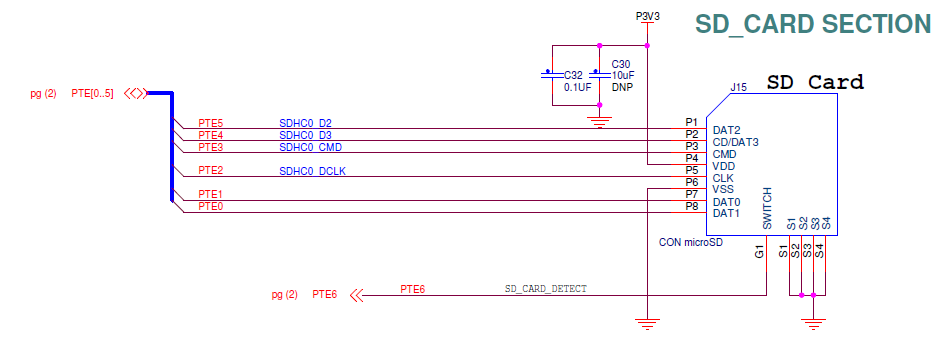
# 介绍

众所周知，我们使用仿真器来下载调试程序。FRDMK64F拥有openSDA接口，所以我们并不需要外置的仿真器。但是如果我们想设计一个不用仿真器但是能下载程序的板子。我们可以使用bootloader。bootloader是一个小型程序，通过使用一些接口比如UART,I2C,SPI，来更新单片机的程序。

这篇文档描述一个简单的bootloader用于FRDMK64F。这个开发板使用SD卡来更新应用。用户将二进制文件放入sd卡，当sd卡插入板子，板子就会自动更新程序。Bootloader的代码和应用代码都将提供以便于测试。

# 2、Bootloader的实现

SD卡与单片机连接原理图如下，板子使用SDHC模块来与SD卡通信。

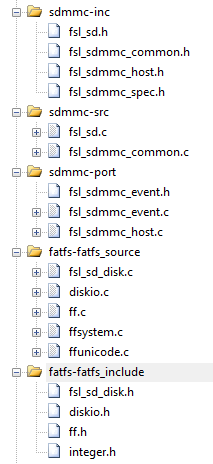


**图1.SD卡原理图**

我们使用2.6.0版本的FRDM-K64F固件，你可以从这个链接下载固件。

“mcuxpresso.nxp.com”.

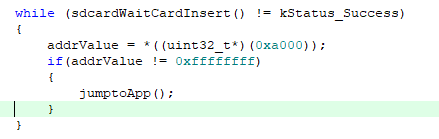
Bootloader使用了SDHC和fatfs文件系统，所以我们应该添加一些文件来支持，



**图2.支持文件**

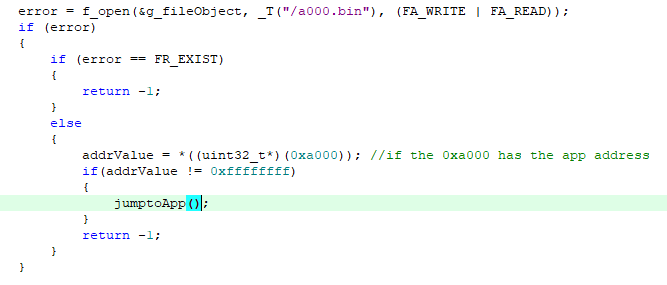
在主代码中，程序会一直等待卡片插入，同时会检测0xa000地址是否有程序，没有程序就会一直等插卡，有程序就会执行。卡插入以后，就会找卡里面名字叫做“a000.bin”的文件。找到了就开始更新程序，没找到就直接执行app代码，如果没有app代码，程序就会结束。

以下代码展示了程序怎样等待插卡过程。



**图3.等待插卡代码**

以下代码展示如何打开二进制文件，如果没有文件就直接进入app代码。



**图4.打开二进制文件**

如果bootloader正常打开文件，更新就会开始。程序会从0xa000开始擦除200k的空间。擦除大小可以根据实际情况更改。

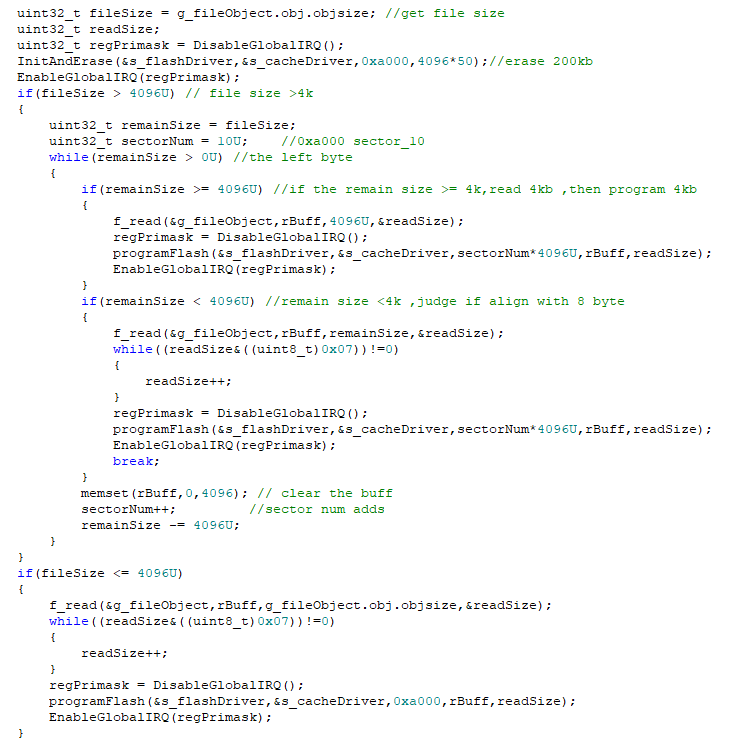
下面我将详细描述更新思路。我们从文件读取的数据是写到叫做“rBuff”的缓存中。缓存大小是4K。在写入数据之前，已经被清0。

注意擦除和编写闪存操作都应该禁用中断，并在完成操作后启用中断。

文件大小决定了用哪种方式更新数据。

1、如果文件小于4k，我们就直接读取整个文件写入缓存，同时判断文件大小是否是8字节对齐的。如果是不是，我们增加“readSize”来从缓存里读到更多数据，读的更多数据在缓存里就是0。将缓存写入到flash中

2、如果文件大于4k，我们使用变量“remainSize”来记录剩余数据量，我们每次读4k到缓存，然后写入flash，直到剩余数据量小于4k，然后使用步骤1的思路。当我们完成一次操作就清空缓存，并且增加扇区号来准备下一次传输。



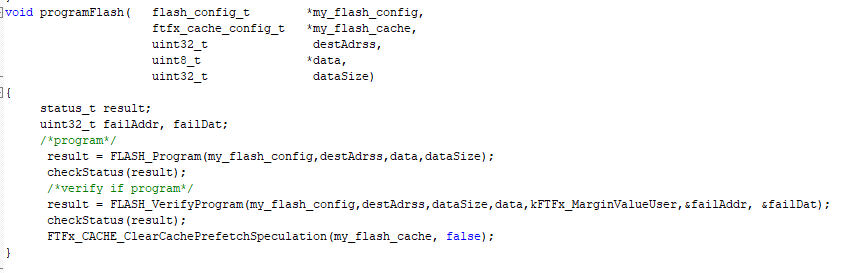
**图5.写操作代码**

擦除空间方法代码如下。他会擦除从给定地址开始给定的大小。“SectorNum”用作扇区偏移，擦除一个就自增。



**图6.擦除代码**

以下展示怎样写入数据

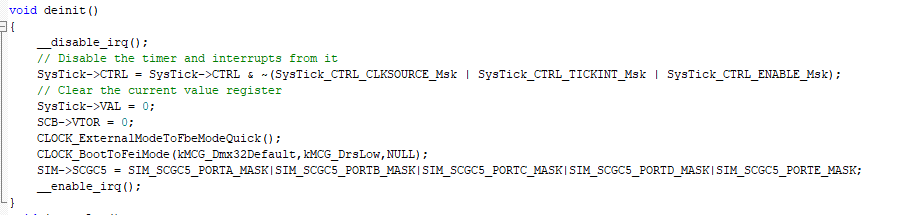


**图7.编写操作代码**

在我们进入app代码之前，我们需要反初始化设备。

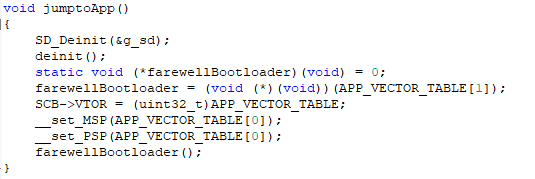
1. 关闭systick, 清除它的值.
2. 设置VTOR为默认值
3. 我们bootloader运行在PEE模式，所以应该让他返回FEI模式。
4. 禁用所有引脚

运行反初始化代码时候应当禁用全局中断。并且完成操作后使能中断。



**图8.反初始化代码**

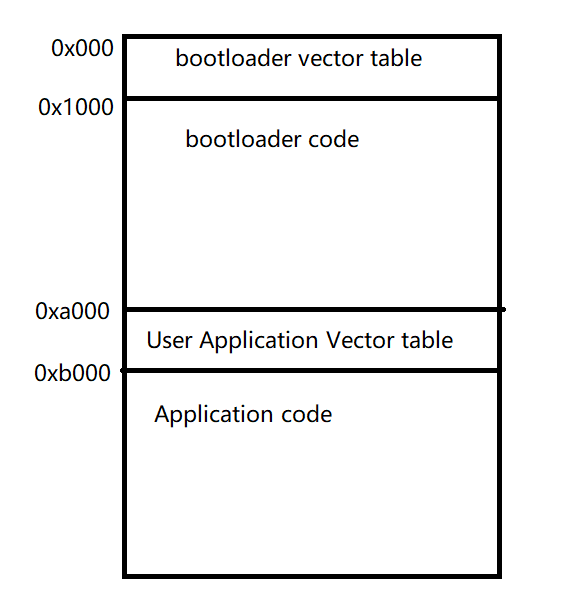
然后我们跳转到app代码。



**图9.进入APP代码**

# 3、内存重分配

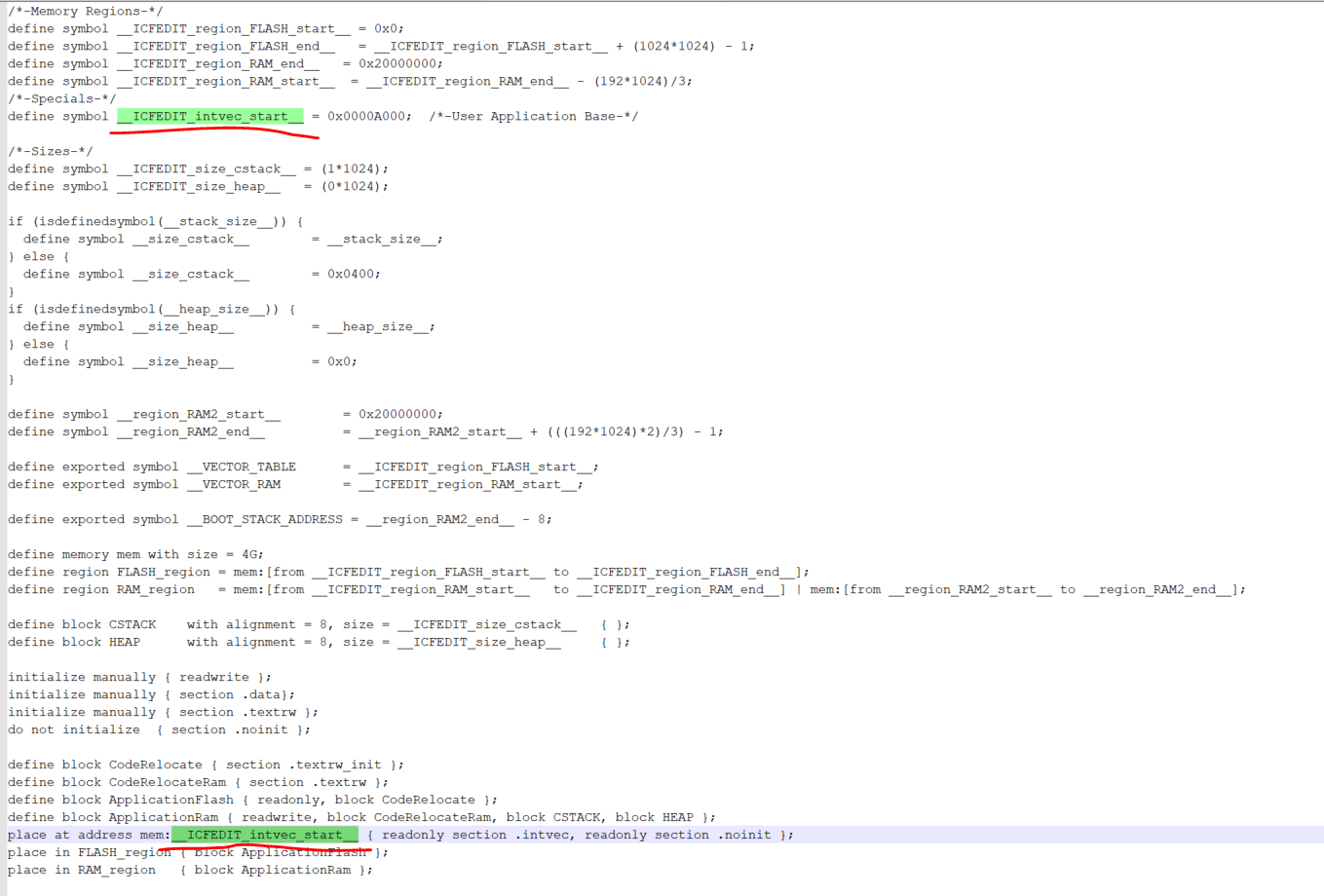
FRDMK64F拥有1M flash内存。从0x00000000 到 0x00100000。如图所示,我们使用0xa000作为应用的开始地址。



图**10.内存**

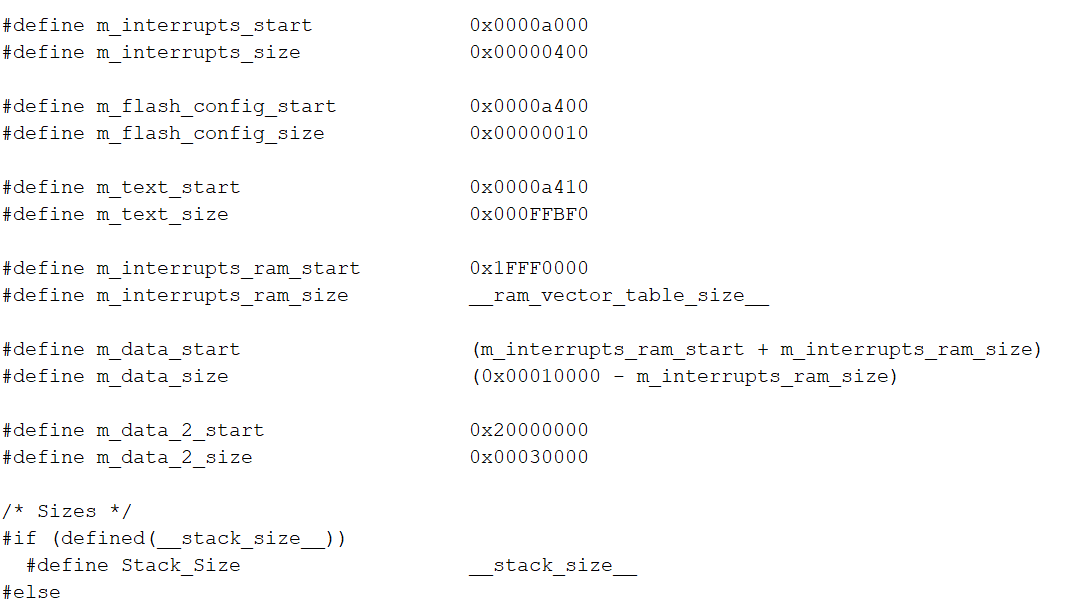
现在将介绍怎样为我们的应用修改不同IDE链接文件。

在IAR中



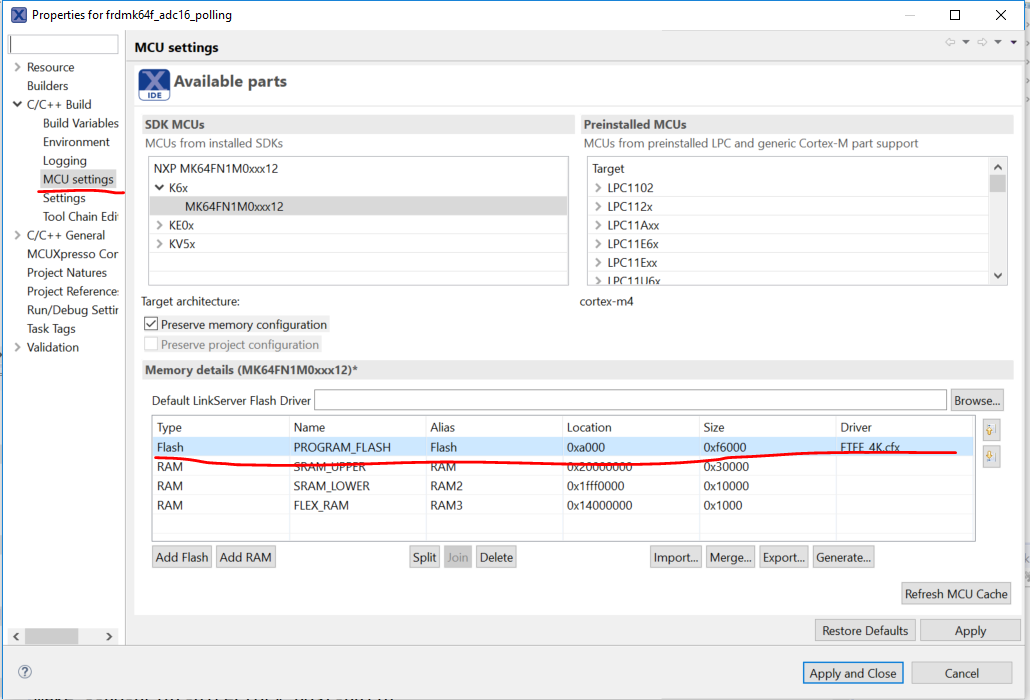
图**11.IAR’s ICF**

在MDK中



**图12.MDK’s SCF**

在MCUXpresso



**图13.MCUXpresso’s flash configuration**

# 4、运行例程

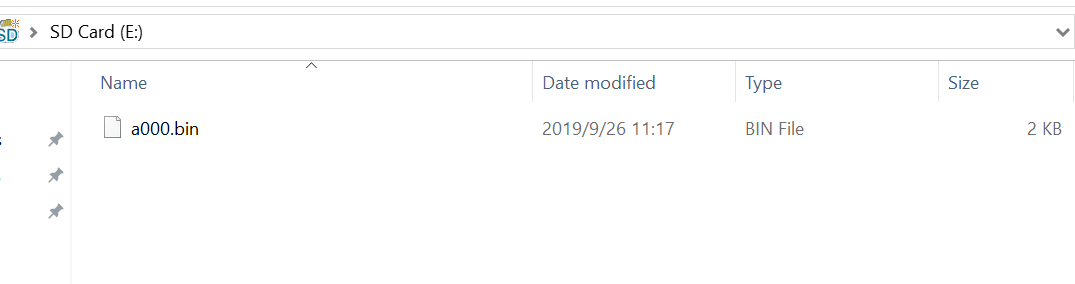
1) 首先下载bootloader

2) 准备一个用户程序。我们使用“led blinky” 做例子

3) 修改应用的链接文件。

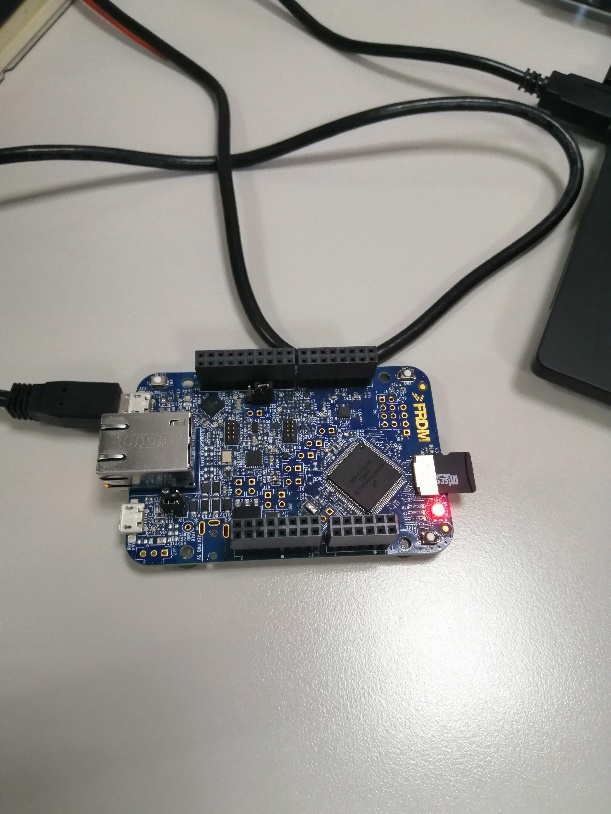
4) 通过IDE产生二进制文件, 命名为“a000.bin”。

5) 把他放入sd卡。



**图14.SD卡**

6) 插入SD卡。上电，等一会儿，应用就会自动执行。



# 5、参考

1) Kinetis MCU的bootloader解决方案

2) KEA128\_can\_bootloader