



复旦微电子

***FM3316/FM3313/FM3312***

***低功耗系列 MCU***

***应用笔记***

**BOOTLOAD**

---

**AN0001**

**V1.1**



本资料是为了让用户根据用途选择合适的上海复旦微电子集团股份有限公司（以下简称复旦微电子）的产品而提供的参考资料，不转让属于复旦微电子或者第三者所有的知识产权以及其他权利的许可。

在使用本资料所记载的信息最终做出有关信息和产品是否适用的判断前，请您务必将所有信息作为一个整体系统来进行评价。

采购方对于选择与使用本文描述的复旦微电子的产品和服务全权负责，复旦微电子不承担采购方选择与使用本文描述的产品和服务的责任。除非以书面形式明确地认可，复旦微电子的产品不推荐、不授权、不担保用于包括军事、航空、航天、救生及生命维持系统在内的，由于失效或故障可能导致人身伤亡、严重的财产或环境损失的产品或系统中。

未经复旦微电子的许可，不得翻印或者复制全部或部分本资料的内容。

今后日常的产品更新会在适当的时候发布，恕不另行通知。在购买本资料所记载的产品时，请预先向复旦微电子在当地的销售办事处确认最新信息，并请您通过各种方式关注复旦微电子公布的信息，包括复旦微电子的网站(<http://www.fmsh.com/>)。

如果您需要了解有关本资料所记载的信息或产品的详情，请与上海复旦微电子集团股份有限公司在当地的销售办事处联系。

## 商 标

上海复旦微电子集团股份有限公司的公司名称、徽标以及“复旦”徽标均为上海复旦微电子集团股份有限公司及其分公司在中国的商标或注册商标。

上海复旦微电子集团股份有限公司在中国发布，版权所有。



# 目录

1.....	说明	6
2.....	原理	7
2.1 背景 .....		7
2.1.1 功能描述 .....		7
2.2 原理 .....		7
3.....	实现方法	8
3.1 普通本地上电通信升级程序 .....		8
通过上电前一秒，模块端不断查询通信口，PC 端不断发送报文，建立通信，实现升级程序...		8
3.2 上电引脚选择通信升级程序 .....		9
3.3 远程通信备份升级程序 .....		11
3.4 远程通信不备份升级程序 .....		13
3.5 芯片结构介绍 .....		16
3.5.1 FLASH 寄存器结构.....		16
3.5.2 FLASH 擦写库函数.....		16
3.6 参考例程使用说明 .....		19
3.6.1 如何将普通用户程序修改为可以 bootload 的用户程序 .....		19
3.6.2 如何合并 bootloader 程序与用户程序 .....		20
3.6.3 bootloader 的 4K 如何修改为 6K, 8K .....		20
4.....	建议的实现步骤	23
5.....	注意事项	24
5.1 软件设计 .....		24
版本信息.....		25
上海复旦微电子集团股份有限公司销售及服务中心 .....		26



## 表目录

表 3-1: bootloader 文件说明 .....	8
表 3-2: 上电引脚升级程序文件说明 .....	9
表 3-3: 远程通信备份升级程序文件说明 .....	11
表 3-4: 远程通信备份升级程序文件说明 .....	14
表 3-5: FLASH 寄存器结构 .....	16
表 3-6: FLASH 擦写库函数 .....	16
表 3-7: Flash_Erase_SECTOR 函数 .....	16
表 3-8: Flash_Write_BYTE 函数 .....	16
表 3-9: Flash_Write_SHORT 函数 .....	17
表 3-10: Flash_Write_STRING 函数 .....	17
表 3-11: Flash_Write_SECTOR 函数 .....	17
表 3-12: Flash_Read_BYTE 函数 .....	18
表 3-13: Flash_Read_SHORT 函数 .....	18
表 3-14: Flash_Read_STRING 函数 .....	18
表 3-15: Flash_Int_Cfg 函数 .....	18

## 图目录

图 1-1: 程序存储空间结构框图.....	7
图 2-1: bootloader 程序位置.....	8
图 3-2: bootloader 程序升级流程框图.....	9
图 3-3: 上电引脚升级程序.....	9
图 3-4: 上电引脚升级程序升级流程框图.....	10
图 3-5: 远程通信备份升级程序.....	11
图 3-6: 远程通信备份升级程序流程图.....	12
图 3-7: 前导源程序流程图.....	13
图 3-8: 远程通信备份升级程序.....	14
图 3-9: 远程通信备份升级程序流程图.....	15
图 3-10: 前导源程序流程图.....	15
图 3-11: 修改程序空间起始地址.....	19
图 3-12: 修改中断向量偏移量.....	19
图 3-13: 修改程序复位地址.....	20
图 3-14: 去除工程文件.....	20
图 3-15: 合并 bootloader 程序与用户程序.....	20
图 3-16: 修改 mainproc.c 文件.....	21
图 3-17: 修改 INT_REMAP.asm.....	21
图 3-18: 修改程序空间起始地址.....	22
图 3-19: 修改中断向量偏移量.....	22
图 3-20: 修改程序复位地址.....	22



# 1 说明

本文档为 FM3316/FM3313/FM3312 系列低功耗 MCU 的应用笔记，用于说明 bootload 的原理和方法。FM3316/FM3313/FM3312 系列是复旦微电子公司开发的低功耗 MCU 芯片，请联系复旦微电子公司提供更多相关文档支持设计开发。

## 2 原理

### 2.1 背景

很多客户生产的模块,因功能的改变或 bug 修正等等原因,需要重新编程。但使用外部编程器又比较麻烦,因为有可能涉及到打开模块带来的不确定后果,或者已经安装不方便碰触。这时候,可以使用 bootloader 功能,通过串口等通信接口远程实现对产品重新编程、在线升级。本文档将简单讲一下原理与实现方法。

#### 2.1.1 功能描述

bootloader程序最必要的功能是可以对用户代码区进行擦除与编写。bootloader是在单片机上电启动时执行的一小段程序,只是一段程序(可以通过编程器预先下载到FLASH),其大小可以是任意的。默认的大小是4KByte(支持硬件代码保护),也可以改成6K,8K等等,但超出4K部分不支持硬件代码保护。

### 2.2 原理

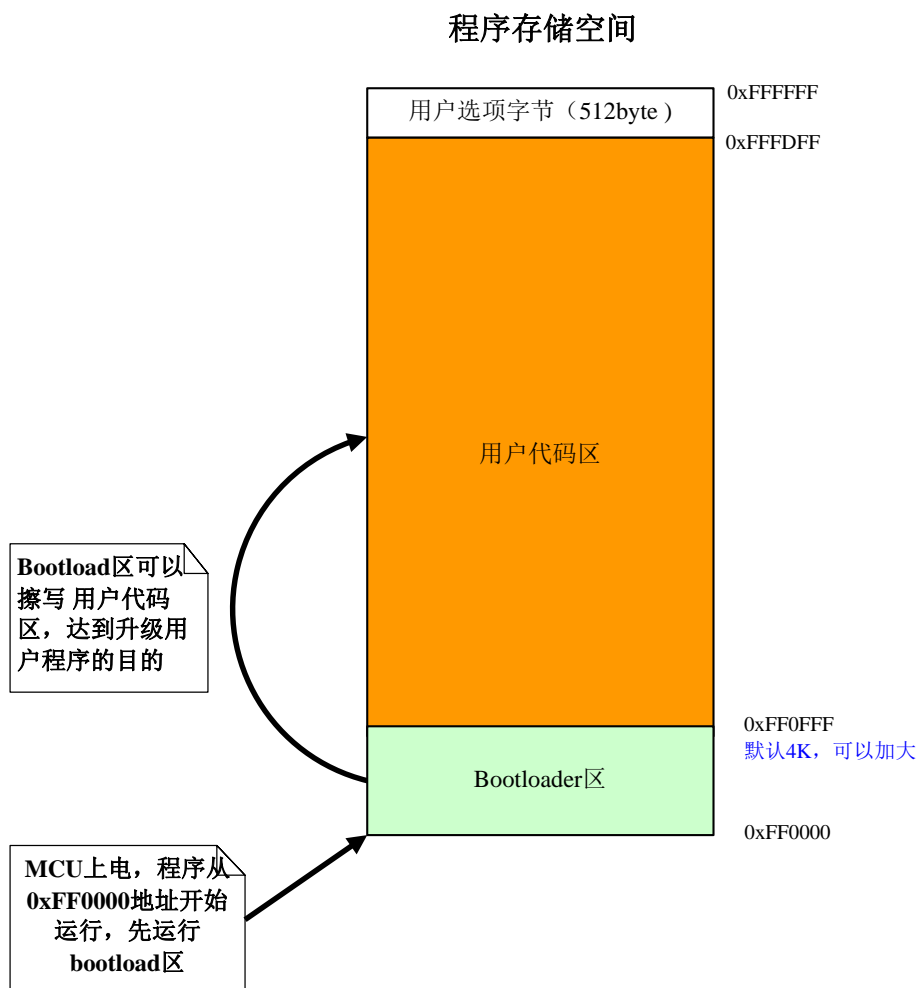


图 1-1 : 程序存储空间结构框图

## 3 实现方法

FM3316 的例程中提供了 4 种 bootload 的方法。

### 3.1 普通本地上电通信升级程序

通过上电前一秒，模块端不断查询通信口，PC 端不断发送报文，建立通信，实现升级程序

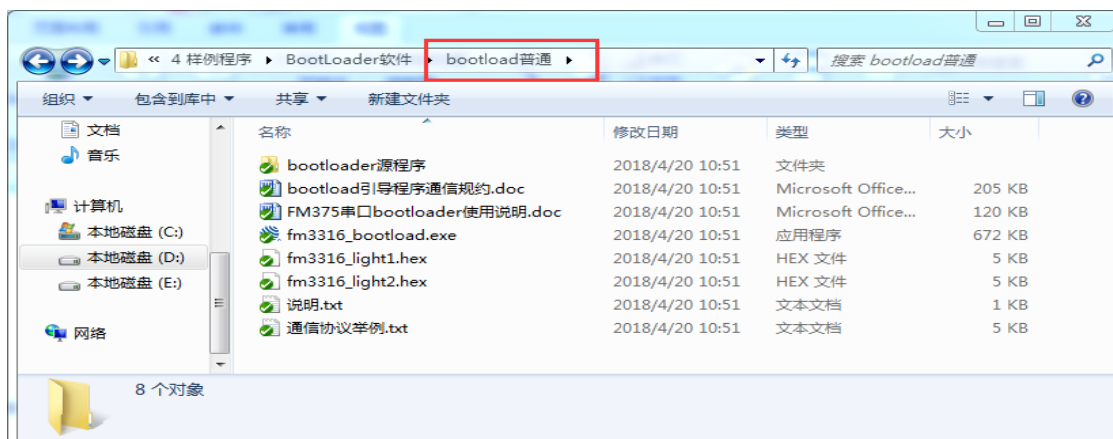


图 2-1 : bootloader 程序位置

表 1-1 : bootloader 文件说明

<bootloader 源程序>	文件夹 是 bootloader 程序的工程，编译出的 hex 就是 flash 前 4Kbootloader 代码。hex 代码名字是 “fm3316-gw-16-keil37.hex”
<bootload 引导程序通信规约.doc>	bootload 引导程序与 PC 端约定的升级通信规约
<FM375 串口 bootloader 使用说明>	bootload PC 端软件的使用说明
<fm3316_bootload.exe>	bootload PC 端软件
<fm3316_light1.hex>	bootload 用户升级程序
<fm3316_light2.hex>	bootload 用户升级程序

- 首先将 bootloader 程序 “fm3316-gw-16-keil37.hex” 写入 3316 芯片，可通过编程器写入
- bootloader 程序在上电的前一秒，会在 uart2 口引脚 PG0 PG1，接收 通信报文，如果接收到 PC 发来的配置帧报文，则程序进入到 bootload 状态，并将整个用户区擦除。随后接收升级程序报文，将升级程序逐一写入用户区。最后接收验证报文。如果验证通过，跳转到用户程序，完成升级。



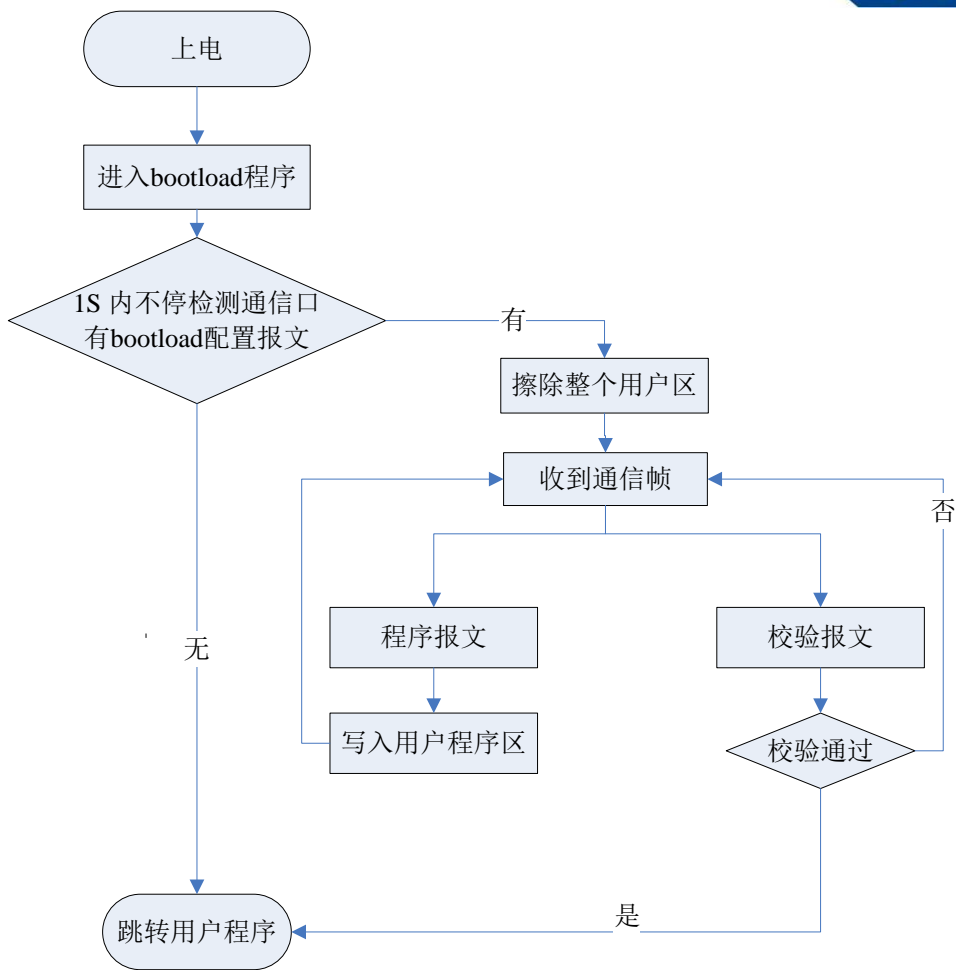


图 3-2 : bootloader 程序升级流程框图

3.2 上电引脚选择通信升级程序

通过上电前 10ms，不断查询 PG2 引脚（可修改为其他引脚）电平，如果引脚为高，实现升级程序

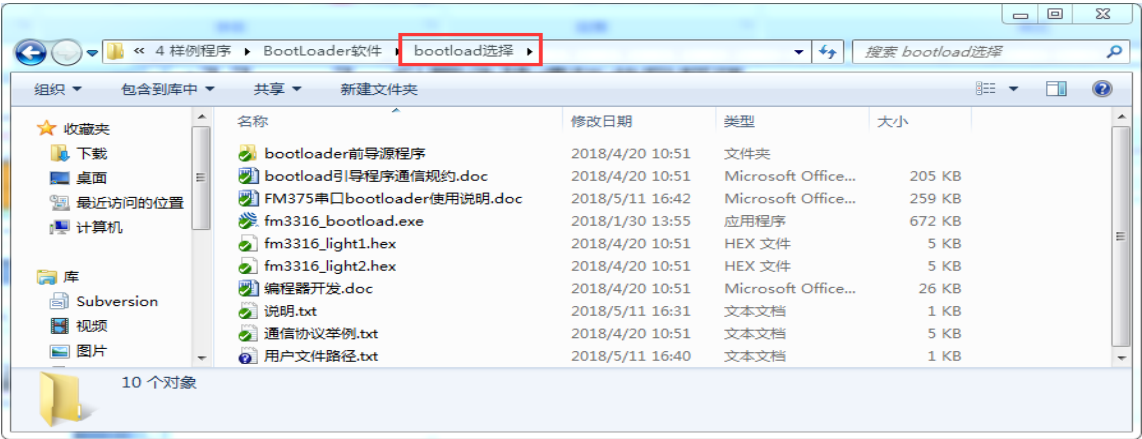


图 4-3 : 上电引脚升级程序

表 2-2 : 上电引脚升级程序文件说明

<bootloader 前导源程序>	文件夹是 bootloader 程序的工程，编译出的 hex
--------------------	--------------------------------

	就是 flash 前 4Kbootloader 代码。hex 代码名字是 “fm3316-gw-16-keil37.hex”
<bootload 引导程序通信规约.doc>	bootload 引导程序与 PC 端约定的升级通信规约
<FM375 串口 bootloader 使用说明>	bootload PC 端软件的使用说明
<fm3316_bootload.exe>	bootload PC 端软件
<fm3316_light1.hex>	bootload 用户升级程序
<fm3316_light2.hex>	bootload 用户升级程序

- 首先将 bootloader 程序 “fm3316-gw-16-keil37.hex” 写入 3316 芯片，可通过编程器写入。
- 将 PG2 引脚 置高电平。
- bootloader 程序在上电的前 10ms，会在 PG2 口引脚检测是否为高，如果位高则程序进入 bootload 状态，如果接收到 PC 发来的配置帧报文，则程序进入到 bootload 状态，并将整个用户区擦除。随后接收升级程序报文，将升级程序逐一写入用户区。最后接收验证报文。如果验证通过，跳转到用户程序，完成升级。

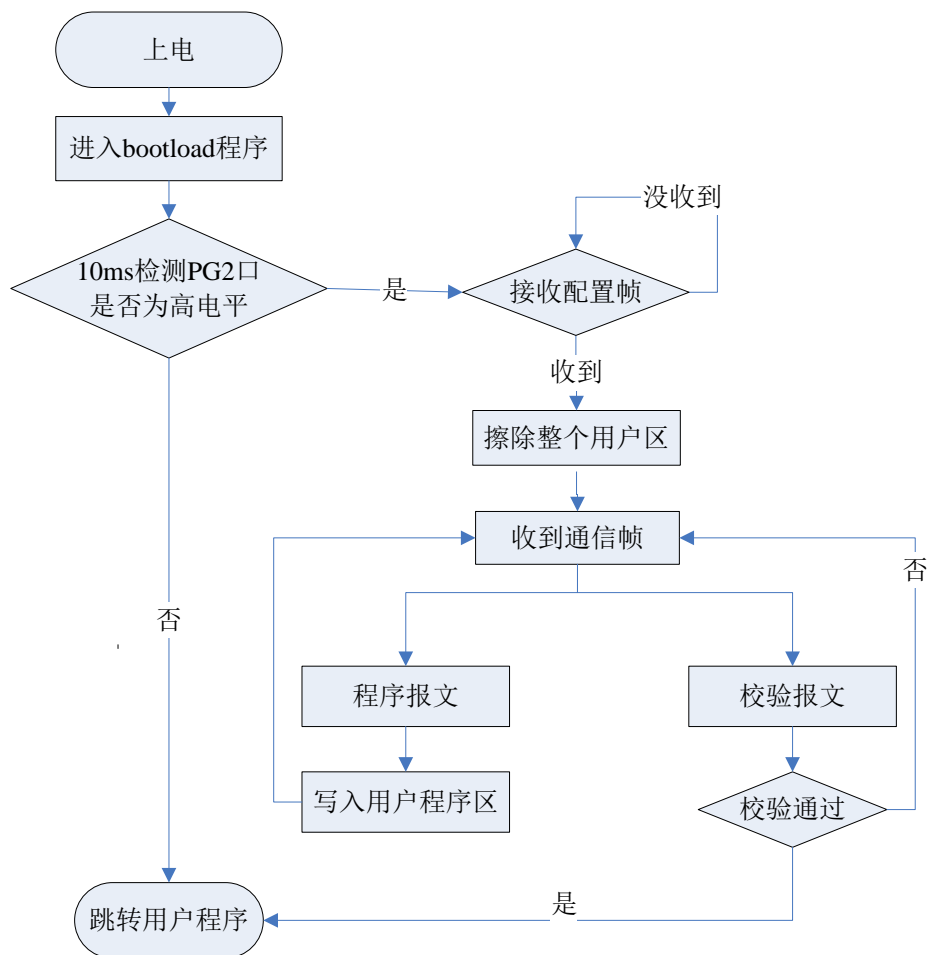


图 5-4：上电引脚升级程序升级流程框图

### 3.3 远程通信备份升级程序

通过写入备份程序，复位后，运行 bootloader，实现升级程序。

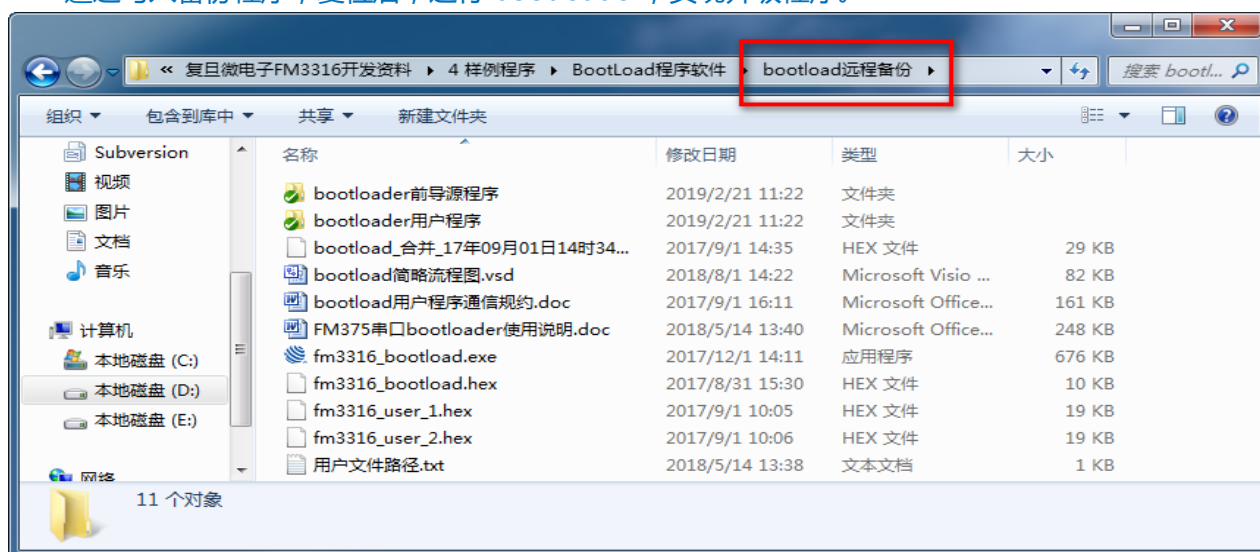


图 6-5：远程通信备份升级程序

表 3-3：远程通信备份升级程序文件说明

<bootloader 前导源程序>	文件夹 是 bootloader 程序的工程，编译出的 hex 就是 flash 前 4Kbootloader 代码。hex 代码名字是 “fm3316-gw-16-keil37.hex”
<bootloader 用户程序>	文件夹 是 bootloader 用户程序，编译出的 hex 就是 flash 用户代码 4K 后的代码。hex 代码名字是 “fm3316_user.hex”
<bootload 用户程序通信规约.doc>	Bootload 用户程序与 PC 端约定的升级通信规约
<FM375 串口 bootloader 使用说明>	bootload PC 端软件的使用说明
<fm3316_bootload.exe>	bootload PC 端软件
<bootload_合并_17年09月01日14时34分58秒.hex>	bootload 引导程序与用户程序的合并程序
<fm3316_bootload.hex>	bootload 引导程序
<fm3316_light1.hex>	bootload 用户升级程序
<fm3316_light2.hex>	bootload 用户升级程序

- 首先将 bootloader 程序 “bootload\_合并\_17年09月01日14时34分58秒.hex” 写入 3316 芯片，可通过编程器写入。
- bootloader 用户程序，会接收 PC 下发的升级报文。包括配置帧报文，升级程序报文，最后接收验证报文。如果验证通过，则将配置信息写入 60 ~ 61 K，然后复位运行前 4 K，bootloader 程序。

- bootloader 程序会判断是否有没，需要升级的文件，如果有就用新程序将旧的用户程序覆盖掉。

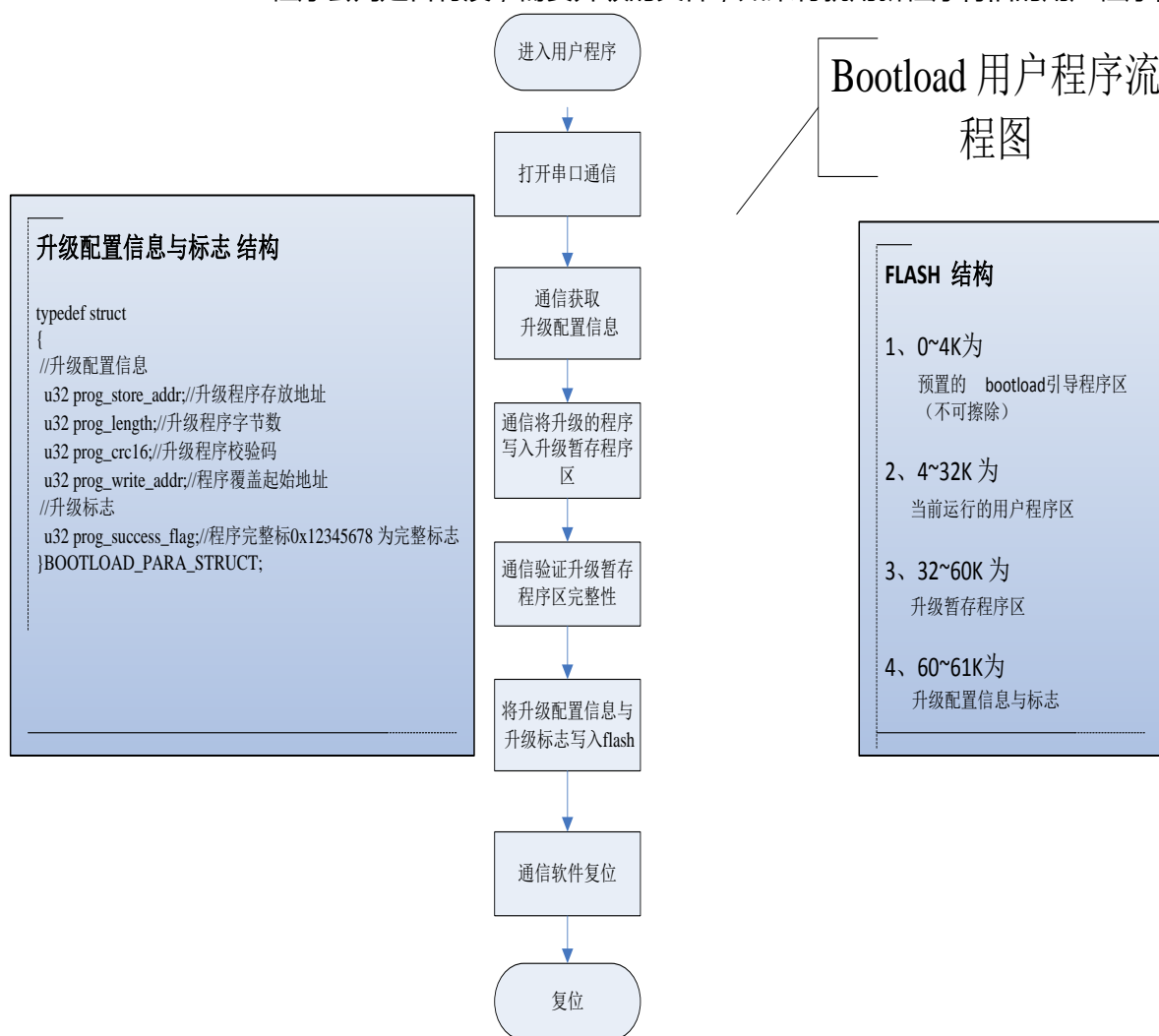


图 7-6：远程通信备份升级程序流程图

## Bootload 前导源程序流程图

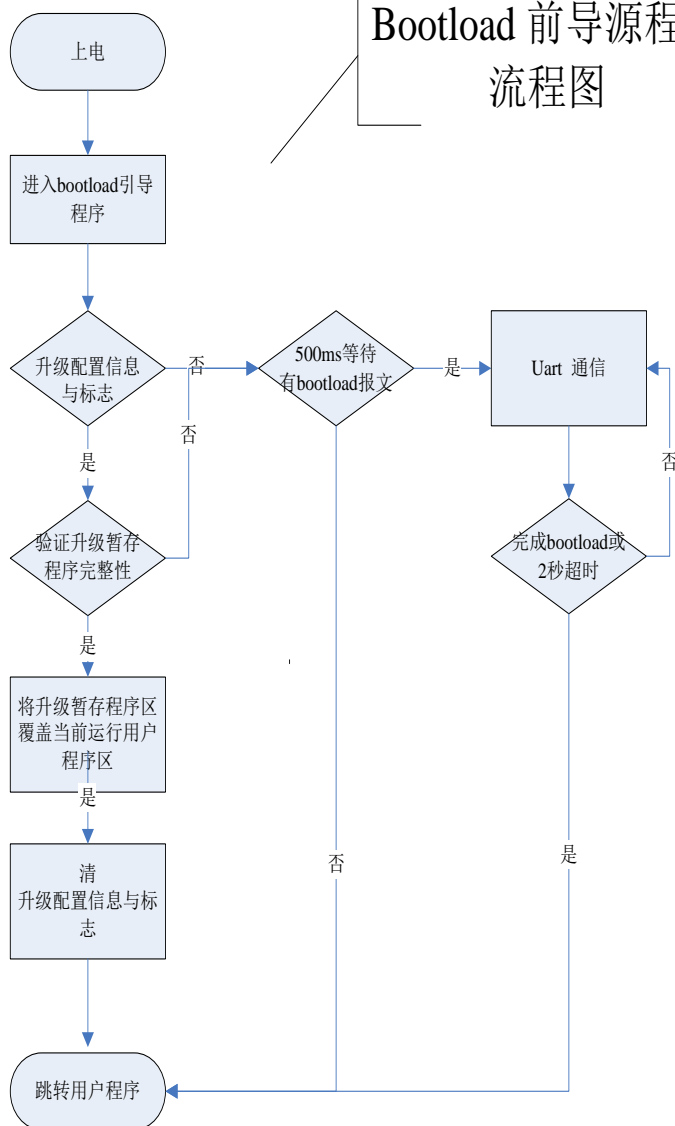


图 8-7：前导源程序流程图

### 3.4 远程通信不备份升级程序

通过写入 bootload 模式标志，复位后，运行 bootloader，实现升级程序。

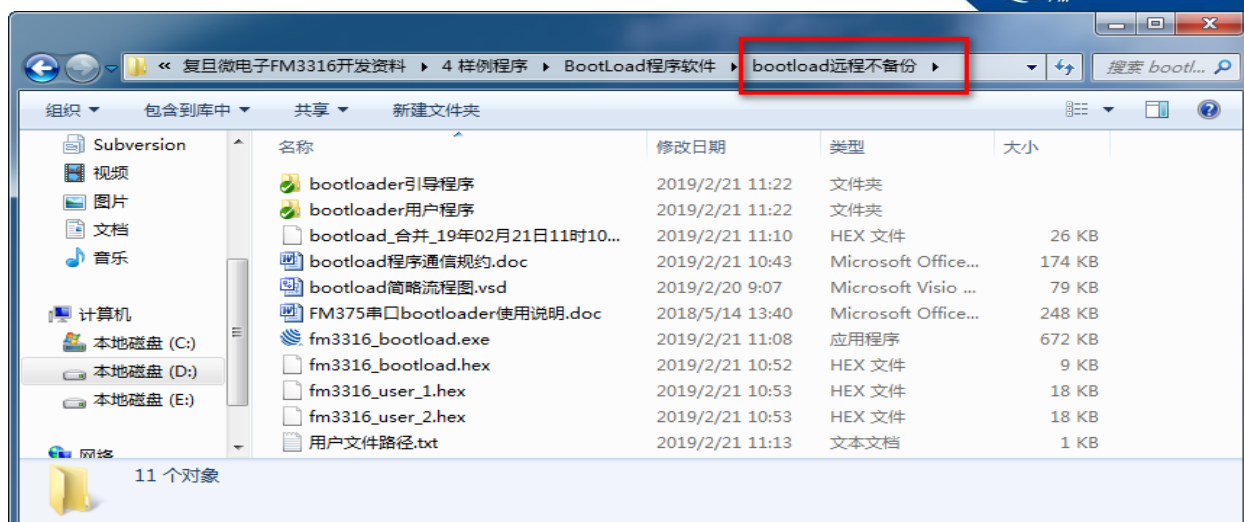


图 9-8：远程通信备份升级程序

表 4-4：远程通信备份升级程序文件说明

<bootloader 前导源程序>	文件夹 是 bootloader 程序的工程，编译出的 hex 就是 flash 前 4Kbootloader 代码。hex 代码名字是“fm3316-gw-16-keil37.hex”
<bootloader 用户程序>	文件夹 是 bootloader 用户程序，编译出的 hex 就是 flash 用户代码 4K 后的代码。hex 代码名字是“fm3316_user.hex”
<bootload 用户程序通信规约.doc>	Bootload 用户程序与 PC 端约定的升级通信规约
<FM375 串口 bootloader 使用说明>	bootload PC 端软件的使用说明
<fm3316_bootload.exe>	bootload PC 端软件
<bootload_合并_19年02月21日11时10分28秒.hex>	bootload 引导程序与用户程序的合并程序
<fm3316_bootload.hex>	bootload 引导程序
<fm3316_light1.hex>	bootload 用户升级程序
<fm3316_light2.hex>	bootload 用户升级程序

- 首先将 bootloader 程序 “bootload\_合并\_19年02月21日11时10分28秒.hex” 写入 3316 芯片，可通过编程器写入。
- bootload 用户程序，会接收 PC 下发的设置 bootload 模式标志报文，将标志写入 flash 后。接收到复位芯片报文，复位芯片。进入前 4K 的 bootloader 程序。
- bootloader 程序会判断是否有 bootload 模式标志，如果有，就一直运行前 4K 的 bootload 报文。接收到 PC 发来的配置帧报文，将整个用户区擦除。随后接收升级程序报文，将升级程序逐一写入用户区。最后接收验证报文。如果验证通过，删除 bootload 模式标志，跳转到用户程序，完成升级。

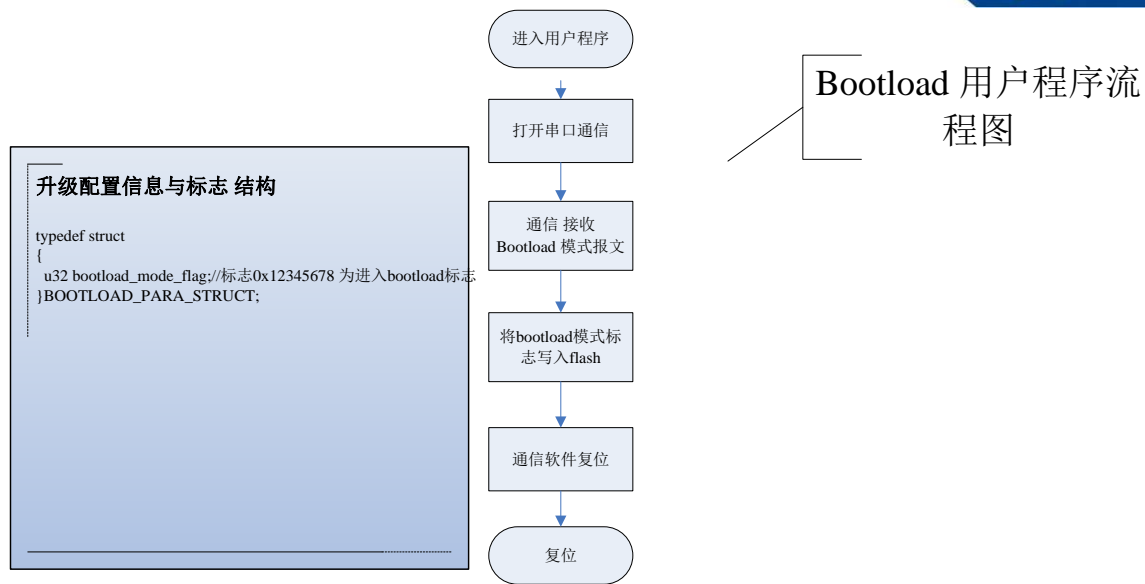


图 10-9：远程通信备份升级程序流程图

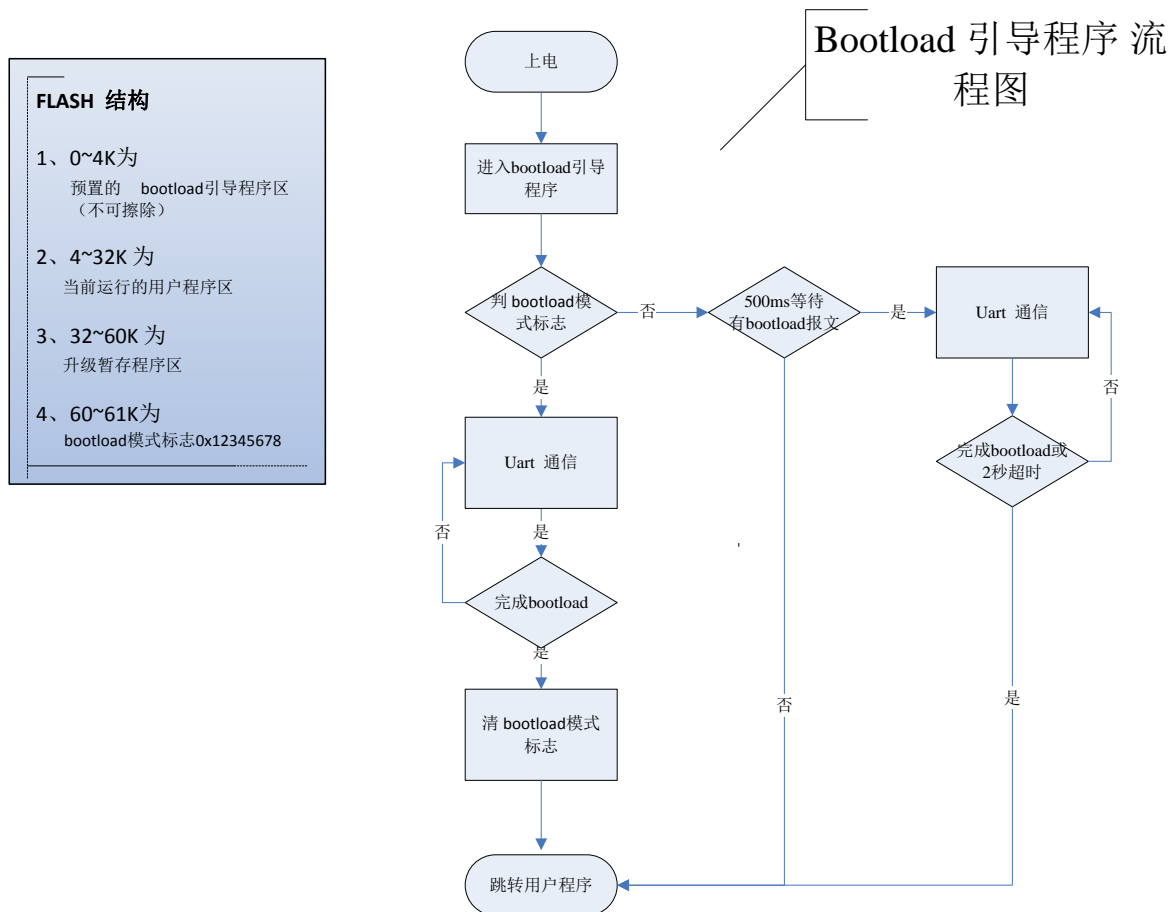


图 11-10：前导源程序流程图



## 3.5 芯片结构介绍

### 3.5.1 FLASH 寄存器结构

表 5-5 : FLASH 寄存器结构

寄存器	描述
ERCSR	Flash 擦除控制寄存器
PRCSR	Flash 编程控制寄存器
FLSKEY	FlashKey 寄存器
FLSCIE	Flash 中断使能寄存器
EPFLAG	Flash 擦写标志寄存器
RPARERR	RAM 校验错误标志寄存器
RPARIE	RAM 校验错误中断使能寄存器

### 3.5.2 FLASH 擦写库函数

表 6-6 : FLASH 擦写库函数

函数名	描述
Flash_Erase_SECTOR	FLASH 扇区擦除
Flash_Write_BYTE	FLASH 单字节写入
Flash_Write_SHORT	FLASH 双字节写入
Flash_Write_STRING	FLASH 多字节写入
Flash_Write_SECTOR	FLASH 扇区写入
Flash_Read_BYTE	FLASH 单字节读取
Flash_Read_SHORT	FLASH 双字节读取
Flash_Read_STRING	FLASH 多字节读取
Flash_Int_Cfg	FLASH 擦写中断配置

#### ● Flash\_Erase\_SECTOR

表 7-7 : Flash\_Erase\_SECTOR 函数

函数名	Flash_Erase_SECTOR
函数原型	unsigned char Flash_Erase_SECTOR(unsigned long Addr)
功能描述	FLASH 扇区擦除
输入参数 1	Addr : 被擦扇区任意字节地址
输出参数 1	无
返回值	0 : 成功 ; 其他 : 发生错误

#### ● Flash\_Write\_BYTE 函数

表 8-8 : Flash\_Write\_BYTE 函数



函数名	Flash_Write_BYTE
函数原型	unsigned char Flash_Write_BYTE(unsigned long Addr, unsigned char WrByte)
功能描述	FLASH 单字节写入
输入参数 1	Addr：编程地址
输入参数 2	WrByte：编程数据
输出参数 1	无
返回值	0：成功；其他：发生错误

## ● Flash\_Write\_SHORT 函数

表 9-9：Flash\_Write\_SHORT 函数

函数名	Flash_Write_SHORT
函数原型	unsigned char Flash_Write_SHORT(unsigned long Addr, unsigned short WrShrot)
功能描述	FLASH 双字节写入
输入参数 1	Addr：编程地址
输入参数 2	WrShrot：编程数据
输出参数 1	无
返回值	0：成功；其他：发生错误

## ● Flash\_Write\_STRING 函数

表 10-10：Flash\_Write\_STRING 函数

函数名	Flash_Write_STRING
函数原型	unsigned char Flash_Write_STRING(unsigned long Addr, unsigned char *Buf, unsigned int Len)
功能描述	FLASH 多字节写入
输入参数 1	Addr：编程地址
输入参数 2	*Buf：编程数据数组地址
输入参数 3	Len：编程字节数
输出参数 1	无
返回值	0：成功；其他：发生错误

## ● Flash\_Write\_SECTOR 函数

表 11-11：Flash\_Write\_SECTOR 函数

函数名	Flash_Write_SECTOR
函数原型	unsigned char Flash_Write_SECTOR(unsigned long Addr)
功能描述	FLASH 扇区写入
输入参数 1	Addr：编程地址
输出参数 1	无
返回值	0：成功；其他：发生错误



## ● Flash\_Read\_BYTE 函数

**表 12-12 : Flash\_Read\_BYTE 函数**

函数名	Flash_Read_BYTE
函数原型	unsigned char Flash_Read_BYTE(unsigned long Addr)
功能描述	FLASH 单字节读取
输入参数 1	Addr : 读取地址
输出参数 1	无
返回值	对应地址的 FLASH 内容

## ● Flash\_Read\_SHORT 函数

**表 13-13 : Flash\_Read\_SHORT 函数**

函数名	Flash_Read_SHORT
函数原型	unsigned short Flash_Read_SHORT(unsigned long Addr)
功能描述	FLASH 双字节读取
输入参数 1	Addr : 读取地址
输出参数 1	无
返回值	对应地址的 FLASH 内容

## ● Flash\_Read\_STRING 函数

**表 14-14 : Flash\_Read\_STRING 函数**

函数名	Flash_Read_STRING
函数原型	unsigned char Flash_Read_STRING(unsigned long Addr, unsigned char *Buf, unsigned int Len)
功能描述	FLASH 多字节读取
输入参数 1	Addr : 读取地址
输入参数 2	*Buf : 读取缓冲区地址
输入参数 3	Len : 读取长度
输出参数 1	对应地址的 FLASH 内容
返回值	无

## ● Flash\_Int\_Cfg 函数

**表 15-15 : Flash\_Int\_Cfg 函数**

函数名	Flash_Int_Cfg
函数原型	void Flash_Int_Cfg(unsigned char Mode)
功能描述	FLASH 擦写中断配置
输入参数 1	Mode : 0=关闭中断 ; 1=开启擦写完成中断 ; 2=开启擦写完成中断+错误中断
输出参数 1	无
返回值	无

## 3.6 参考例程使用说明

### 3.6.1 如何将普通用户程序修改为可以 bootload 的用户程序

- 修改程序空间起始地址

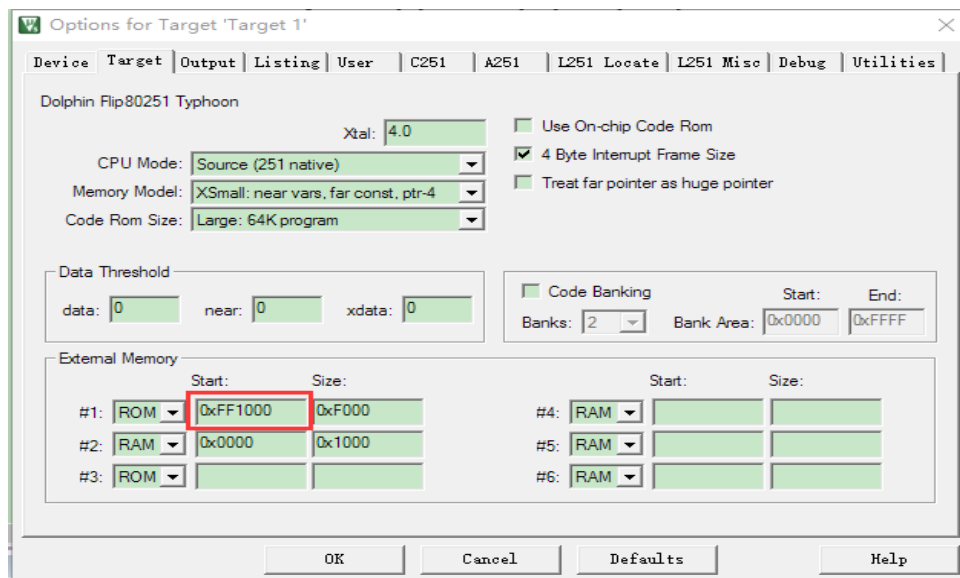


图 12-11：修改程序空间起始地址

- 修改中断向量偏移量

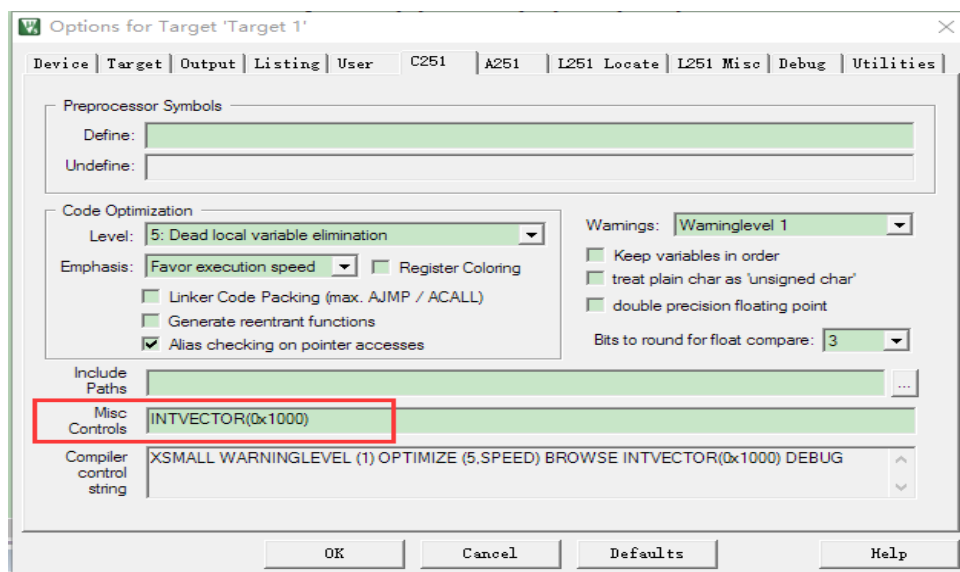


图 13-12：修改中断向量偏移量

- 修改程序复位地址为 0x1000（在启动文件 START251.A51 里修改）

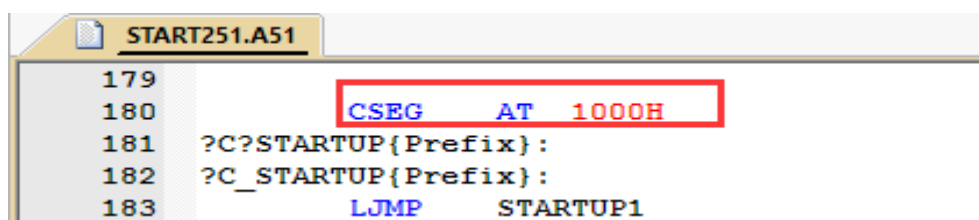


图 14-13：修改程序复位地址

- 去除工程中的 startaddr.inc、monitor.asm、monitor.inc 这三个文件。工程不能仿真。

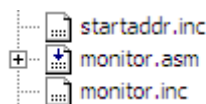


图 15-14：去除工程文件

- 重新编译工程，得到新的 hex 即可支持 bootloader 功能

### 3.6.2 如何合并 bootloader 程序与用户程序

将前 bootloader 程序与用户程序合并在一起，方便生产时用编程器烧写入芯片。参看使用说明

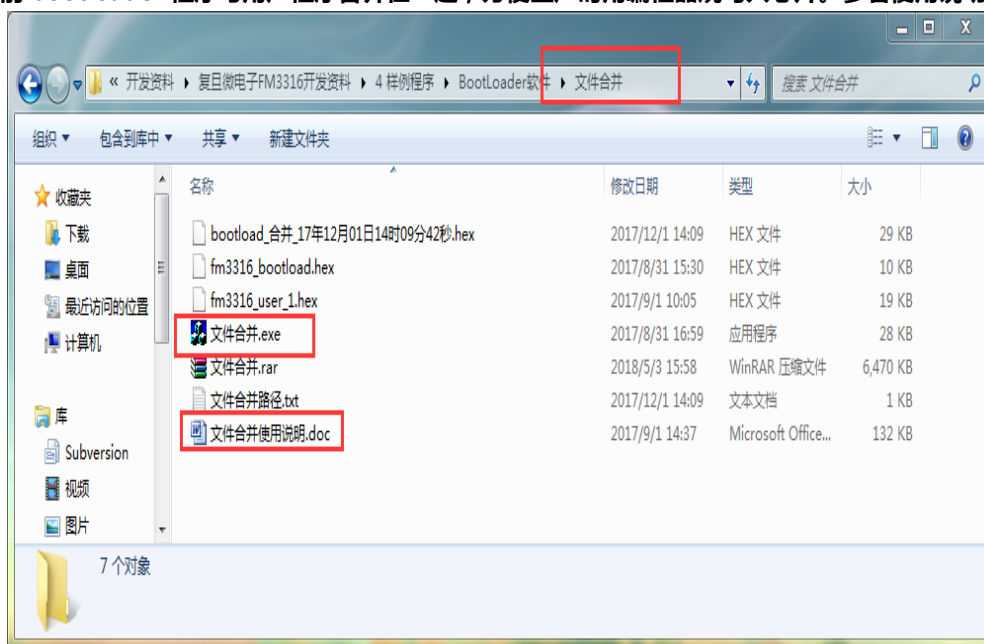


图 16-15：合并 bootloader 程序与用户程序

### 3.6.3 bootloader 的 4K 如何修改为 6K，8K

bootloader 4K 代码不够用怎么办，是否可以改成 6K 或 8K，下面介绍如何将 4K 改为 8K。需要修改哪些地方

- bootloader 代码需要修改的地方

将 mainproc.c 文件中的 0x1000 改为 0x2000

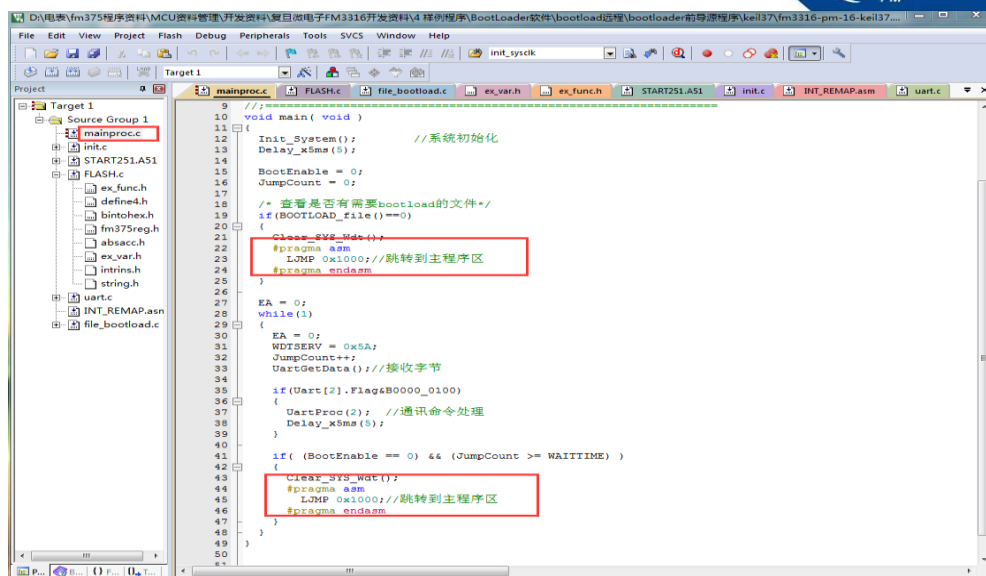


图 17-16 : 修改 mainproc.c 文件

将 INT\_REMAP.asm 文件中的中断跳转地址 1003h, 100Bh 改为 2003h, 100Bh 改为 200Bh .....等等, 全部改掉

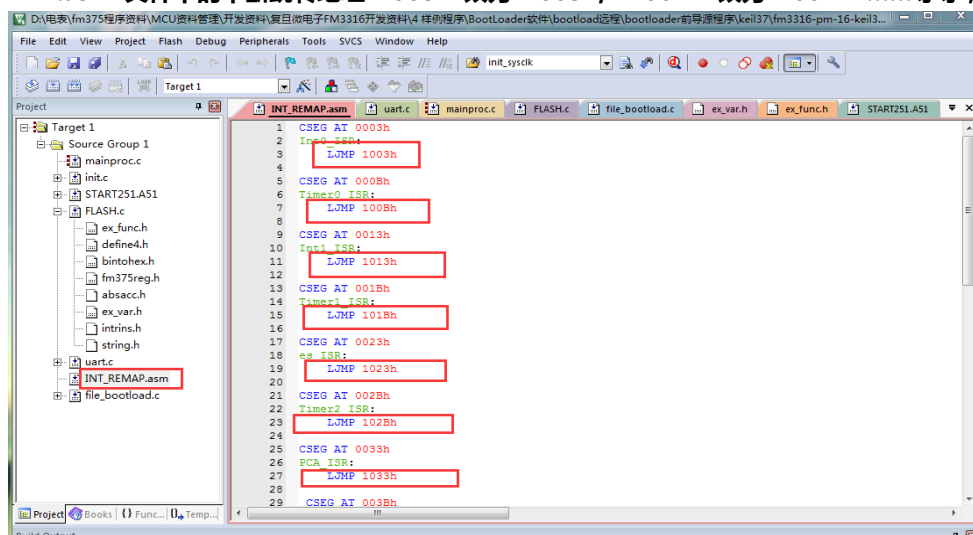


图 18-17 : 修改 INT\_REMAP.asm

## ● 用户代码需要修改的地方

修改程序空间起始地址 0xFF1000 改为 0xFF2000, Size 改为 0xE000

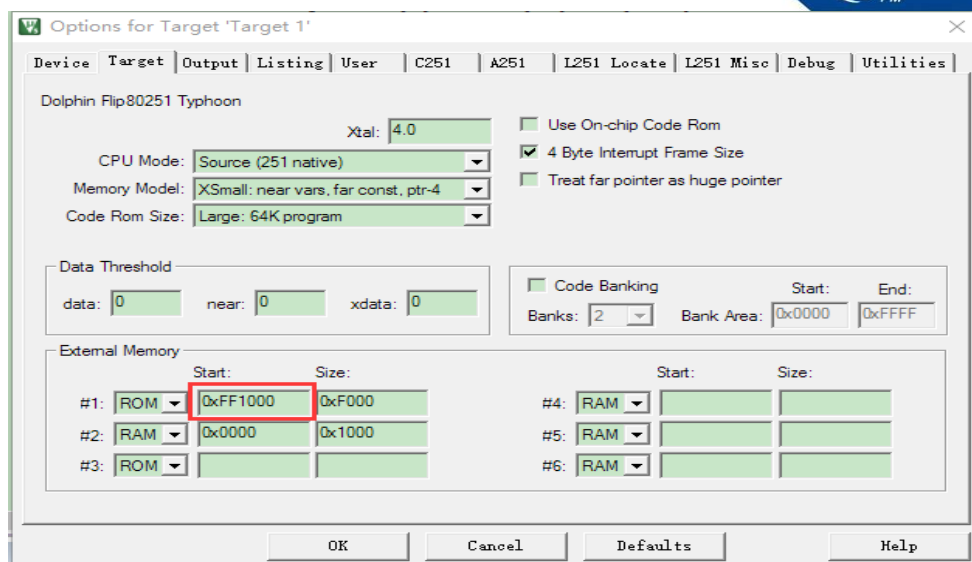


图 19-18：修改程序空间起始地址

修改中断向量偏移量 0x1000 改为 0x2000

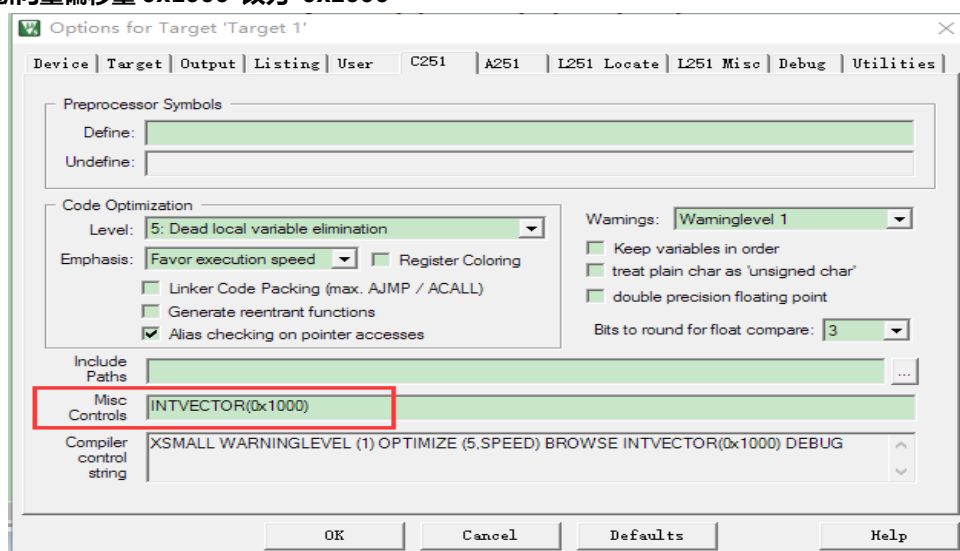


图 20-19：修改中断向量偏移量

修改程序复位地址为 0x1000 改为 0x2000 (在启动文件 START251.A51 里修改)

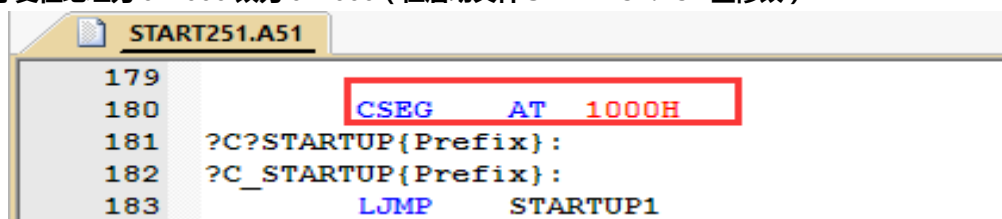


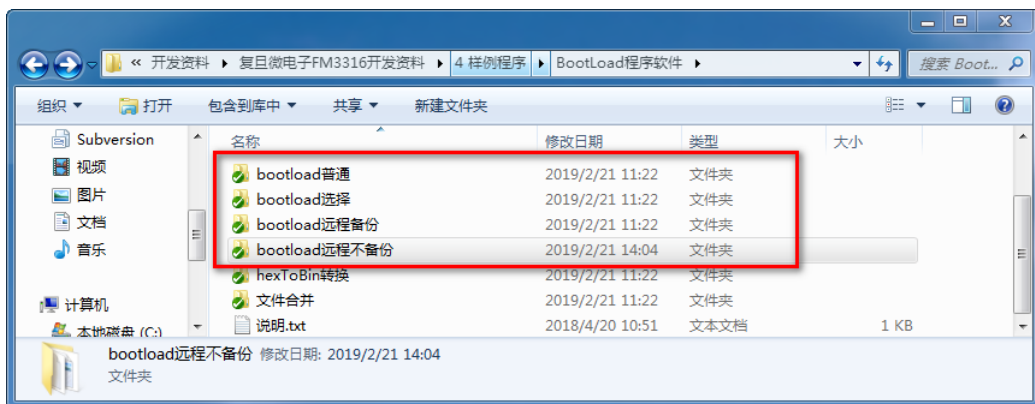
图 21-20：修改程序复位地址

## ● PC 软件修改

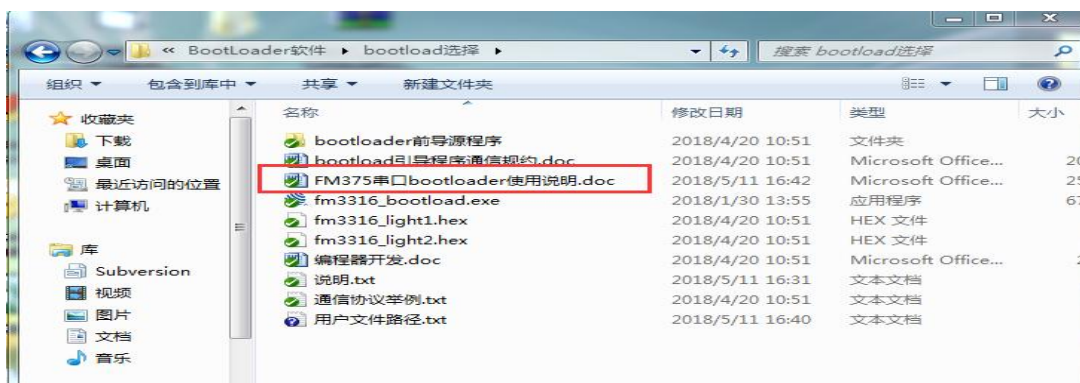
请联系我们软件人员或销售人员，获取相应的 bootloader 大小的 PC 软件。

## 4 建议的实现步骤

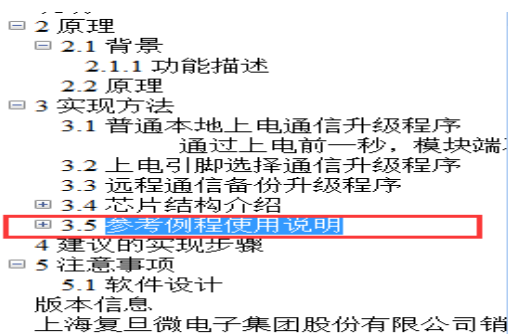
- 参看说明，选择适合的 bootload 方法



- 按照使用说明，在 demo 板上验证下功能



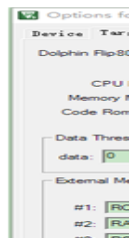
- 按着本手册 3.5 章节，修改各个程序，将例程匹配你的硬件



### 3.5 参考例程使用

#### 3.5.1 如何将普通用户程

##### 3.5.1.1 修改程序空



- 根据自己的需求，修改通信规约，修改 PC 软件。





## 5 注意事项

### 5.1 软件设计

- 波特率与 uart 口是可变的，用户可以自己在程序中修改
- flash 扇区写功能需要用到 ram 最后 512 字节。所有例程中这 512 字节都是留出的。
- 远程升级程序，示例的最大升级程序大小是 28K。如用户代码超出 28K，请联系我们的软件人员。





## 版本信息

版本号	发布日期	更改说明
1.0	2018.05	首次发布
1.1	2019.02	增加远程不备份方式



# 上海复旦微电子集团股份有限公司销售及服务中心

## 上海复旦微电子集团股份有限公司

地址：上海市国泰路 127 号 4 号楼

邮编：200433

电话：(86-021) 6565 5050

传真：(86-021) 6565 9115

## 上海复旦微电子（香港）股份有限公司

地址：香港九龙尖沙咀东嘉连威老道 98 号东海商业中心 5 楼 506 室

电话：(852) 2116 3288 2116 3338

传真：(852) 2116 0882

## 北京办事处

地址：北京市东城区东直门北小街青龙胡同 1 号歌华大厦 B 座 423 室

邮编：100007

电话：( 86-10 ) 8418 6608

传真：( 86-10 ) 8418 6211

## 深圳办事处

地址：深圳市华强北路 4002 号圣廷苑酒店世纪楼 1301 室

邮编：518028

电话：(86-0755) 8335 0911 8335 1011 8335 2011 8335 0611

传真：(86-0755) 8335 9011

## 台湾办事处

地址：台北市 114 内湖区内湖路一段 252 号 12 楼 1225 室

电话：(886-2) 7721 1889

传真：(886-2) 7722 3888

## 新加坡办事处

地址：237, Alexandra Road, #07-01, The Alexcior, Singapore 159929

电话：(65) 6472 3688

传真：(65) 6472 3669

## 北美办事处

地址：2490 W. Ray Road Suite#2 Chandler, AZ 85224 USA

电话：(480) 857-6500 ext 18

公司网址：<http://www.fmsh.com/>