**STM32L151-IAP**

适用于HCF710/HCF1100

2019-6-11

目录

[1. 技术背景 3](#_Toc11160134)

[2. BootLoader程序技术点 3](#_Toc11160135)

[3. 关键字 4](#_Toc11160136)

[4. 启动过程 4](#_Toc11160137)

[5. 系统流程图 5](#_Toc11160138)

[6. 存储空间准备 7](#_Toc11160139)

[6.1 Flash划分 7](#_Toc11160146)

[6.2 EEPROM划分 8](#_Toc11160154)

[7 BootLoader程序设置 9](#_Toc11160155)

[8 APP程序设置 9](#_Toc11160156)

[9 传输协议 10](#_Toc11160157)

[指令0-0：获取BootLoader信息 10](#_Toc11160158)

[指令0-1：升级命令 11](#_Toc11160159)

[指令1：握手 11](#_Toc11160160)

[指令2：同步指令 12](#_Toc11160161)

[指令3：数据指令 12](#_Toc11160162)

[10 补充说明 14](#_Toc11160163)

# 技术背景

ISP：在系统编程（In System Programing）

ICP：在线编程（In-Circuit Programming）:

通过JTAG/SWD协议或者系统加载程序(Bootloader)下载用户应用程序到微控制器中。

IAP：在应用中编程(IAP,In Application Programming):

通过任何一种通信接口(如IO端口,USB,CAN,UART,I2C,SPI等)下载程序或者应用数据到存储器中。

IAP提供了一种新的更新APP程序的手段。

说明：

BootLoader代码通过原有的ICP方式进行烧写。

APP应用程序通过IAP方式进行烧写，文件可使用\*.axf或\*.hex或.bin格式，本文中采用的是二进制文件(.bin)。

# BootLoader程序技术点

BootLoader应具备以下驱动

* Systick
* 1-2个GPIO
* USART通信
* FLASH读写
* EEPROM读写

BootLoader应具备以下功能：

RS485地址初始化及读取

BootLoader版本信息

ModBus部分协议支持（基本应答功能）

SN号操作

APP程序更新

CRC校验

异常处理，保证bootloader的安全性

# 关键字

STM32L151CB-A、采用小端模式、页擦除方式、按照字（4B）写入、可跨页、读-写起始地址必须是4的整数倍

Flash支持字节读、半字读（short）、全字读

时钟：7.3728M X 4

读一个Page（256B、写一个Page（256B）

# 启动过程

无IAP APP程序运行过程：



加入IAP后的运行过程：



# 系统流程图

STM32L151-BootLoader程序运行图：



# 存储空间准备



STM32L151CB Memory map(0x00000000-0x20000000)



## Flash划分

STM32L151内部Flash空间为128KB，对Flash空间划分可有一下两种方式，本次将采用方式2。

方式1 方式2



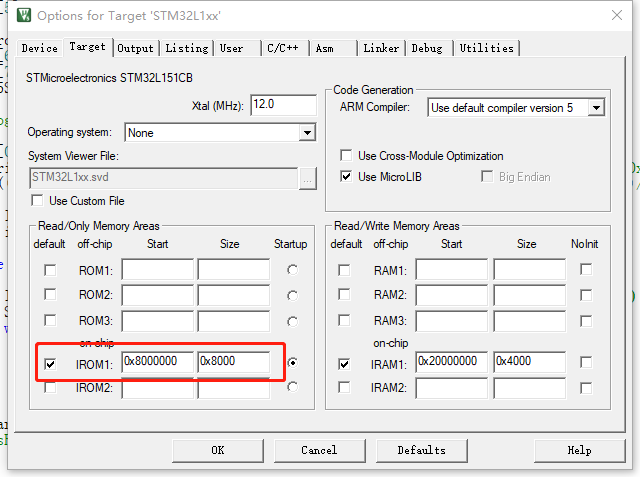
## EEPROM划分

STM32L151CB-A内部包含4KB EEPROM，对EEPROM空间进行划分，如下图所示：



# BootLoader程序设置

此次设计中将BootLoader占用空间设置为32Kb，当然也可以设置成其他容量，需要在工程配置中进行如下设置：



# APP程序设置

**对于APP程序，由于程序起始位置更改，因此需要设置中断向量偏移。**

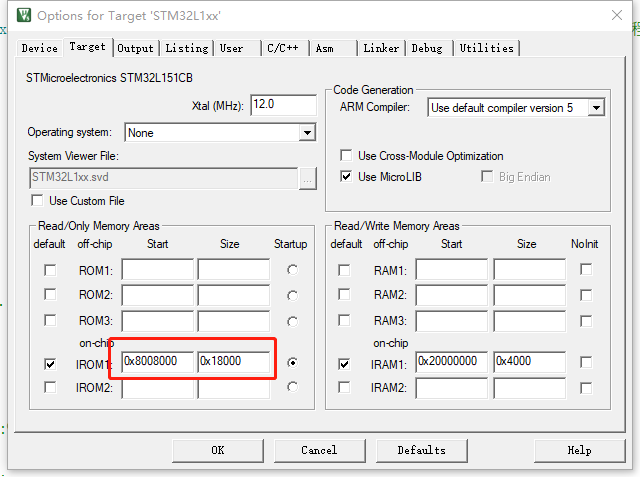
**在main函数起始位置定义：**

**SCB->VTOR = FLASH\_BASE | 0x8000; //设置向量表偏移值，Bootloader起始地址为0x0800 0000，共32KB空间,APP程序起始地址为0x0800 8000**

**即可，其中0x8000即为偏移量**

**对于SRAM的偏移，IAP和APP是两个独立的程序，不会同时运行。可以独立占用全部RAM，可以不设置。**

**同时在编译器的ROM设置区域，需要设置偏移量**



# 传输协议

数据传输采用私有化协议（也可是使用Xmodem、Ymodem、Zmodem协议），RS485总线，由于其半双工的性质，因而需要做一些规定, 每包数据包含256B字节数据(Flash写操作按照字进行，256字节数据组成64个字)：

## 指令0-0：获取BootLoader信息

说明：设备上电后，首先进入BootLoader程序，前5秒，BootLoader等待主机命令。

0-0-0： 主机发送获取信息指令；

0-0-1： 从机返回设备类型、BootLoader软件版本；

格式：

MT：地址（1B）+获取信息（1B）+附加字节（4B）+CRC检验（2B）

SR：地址（1B）+回复（1B）+设备类型（2B）+软件版本（2B）+CRC校验（2B）

|  |  |  |  |
| --- | --- | --- | --- |
| MT | MT | SR | SR |
| 地址（1B） | 01-247 | 地址（1B） | 01-247 |
| 更新标志位（1B） | 22 | 回复信息（1B） | A2 |
| 附加字节（4B） | 44 66 88 aa | 设备类型（2B） |  |
| CRC校验（2B） |  | 软件版本（2B） |  |
|  |  | CRC校验（2B） |  |

## 指令0-1：升级命令

说明：设备上电后，首先进入BootLoader程序，前5秒，BootLoader等待主机命令，判断是否升级。

0-0-0： 主机发送执行Update指令；

0-0-1： 从机擦除对应的标志位，发送ACK，进入Update Firmware；

格式：

MT：地址（1B）+更新标志位（1B）+附加字节（4B）+CRC检验（2B）

SR：地址（1B）+BL准备好（1B）+附加字节（4B）+CRC校验（2B）

|  |  |  |  |
| --- | --- | --- | --- |
| MT | MT | SR | SR |
| 地址（1B） | 01-247 | 地址（1B） | 01-247 |
| 更新标志位（1B） | 23 | BL准备好（1B） | A3 |
| 附加字节（4B） | 44 66 88 aa | 附加字节（4B） | 44 66 88 aa |
| CRC校验（2B） |  | CRC校验（2B） |  |

## 指令1：握手

指令说明：主机和Bootloader中的Update Firmware程序建立联系。

1-1： 主机发送更新代码指令（本步骤间隔1S执行，等待从机重启完毕）；

1-2：从机回复ACK信息；

格式：

MT：地址（1B）+更新APP（1B）+附加字节（4B）+CRC检验（2B）

SR：地址（1B）+ 连接成功（1B）+附加字节（4B）+CRC校验（2B）

|  |  |  |  |
| --- | --- | --- | --- |
| MT | MT | SR | SR |
| 地址（1B） | 01-247 | 地址（1B） | 01-247 |
| 更新APP（1B） | 0x25 | 连接成功（1B） | 0xA5 |
| 附加字节（4B） | 44 66 88 aa | 附加字节（4B） | 44 66 88 aa |
| CRC校验（2B） |  | CRC校验（2B） |  |

## 指令2：同步指令

指令说明：主机将APP Image文件信息（包含软件版本、程序总字节数、帧数）同步至BootLoader，BootLoader根据对应信息对Flash相应空间进行擦除，同时根据帧数进行数据接收。

2-1：主机发送代码基本信息；

2-2：从机根据信息擦除对应空间，完成后返回ACK；

格式：

MT：地址（1B）+镜像标识（1B）+软件版本号（2B）+程序总字节数（4B）+总数据条数（2B）+CRC检验（2B）

SR：地址（1B）+镜像标识（1B）+附加字节（4B）+CRC校验（2B）

|  |  |  |  |
| --- | --- | --- | --- |
| MT | MT | SR | SR |
| 地址（1B） | 01-247 | 地址（1B） | 01-247 |
| 镜像标识（1B） | 0x26 | 镜像标识（1B） | 0xA6 |
| 软件版本号（2B） |  | 附加字节（4B） | 44 66 88 aa |
| 程序总字节数（4B） |  | CRC校验（2B） |  |
| 总数据条数（2B） |  |  |  |
| CRC校验（2B） |  |  |  |

## 指令3：数据指令

指令说明：主机将包含有APP Image的数据发送至BootLoader。

3-1：主机发送第一条数据包；

3-2：从机接收数据包，执行校验，执行对应的更新操作，返回ACK；

格式：

MT：地址（1B）+数据包标识（1B）+当前数据编号（2B）+数据（256B）+CRC检验（2B）

SR：地址（1B）+数据包标识（1B）+附加字节（4B）+CRC校验（2B）

|  |  |  |  |
| --- | --- | --- | --- |
| MT | MT | SR | SR |
| 地址（1B） | 01-247 | 地址（1B） | 01-247 |
| 数据标识（1B） | 27 | 数据标识（1B） | A7 |
| 当前数据包编号（2B） |  | 当前数据包编号（2B） |  |
| 数据包（256B） |  | 附加字节（2B） | 88 aa |
| CRC校验（2B） |  | CRC校验（2B） |  |

3-3：主机发送最后一包数据；

3-4：从机写对应的标志位，返回ACK，并执行IAP跳转；

格式：

MT：地址（1B）+完成标识（2B）+当前数据编号（1B）+数据（256B）+CRC检验（2B）

SR：地址（1B）+完成标识（1B）+ +程序总字节数（4B）+ +CRC校验（2B）

|  |  |  |  |
| --- | --- | --- | --- |
| MT | MT | SR | SR |
| 地址（1B） | 01-247 | 地址（1B） | 01-247 |
| 数据标识（1B） | 27 | 完成标识（1B） | B7 |
| 当前数据包编号（2B） |  | 程序总字节数（4B） |  |
| 数据包（256B） |  | CRC校验（2B） |  |
| CRC校验（2B） |  |  |  |

3-5：主机根据4-2ACK信息，展示相应的更新对应界面；

# 补充说明

关于SN号说明：

程序烧写时SN号嵌入在BootLoader程序中，与现有方法一致。

BootLoader获取程序段中的SN信息，并将其存储在EEPROM中的共享空间中，每次上电，BootLoader获取程序段中的SN信息和EEPROM中的SN信息，并进行比对，不一致时对EEPROM中的SN信息进行覆盖。

APP程序镜像中不包含SN信息，APP程序运行后，去EEPROM共享空间中获取SN信息，并保存在临时变量中。

关于IAP方式进行APP镜像更新

为减少对正常程序的改动（添加升级指令），在正常运行的程序中，如果需要升级固件，则需要对设备重启，重启后前5S中，发送升级指令，以进入固件升级程序。