

# 基于 C8051F34x 单片机的 USB Bootloader 设计

刘云龙<sup>1</sup>, 张伟霞<sup>2</sup>

(1. 广州海格通信集团股份有限公司, 广东 广州 510663;

2. 广州广电运通金融电子股份有限公司, 广东 广州 510663)

**摘要:** 介绍了一种在 Silicon Labs 公司的 C8051F34x 系列单片机上实现 USB Bootloader 的设计方法, 使应用程序可以通过 USB 或 COM 通信实现远程在线升级。首先, 简述了 USB Bootloader; 然后, 详细阐述了 USB Bootloader 程序的设计和 APP 固件程序的设计, 以及设计中需要注意的问题; 最后, 用 VC++ 开发上位机软件来完成该 Bootloader 的远程在线下载功能。该 Bootloader 可以很方便地在其他类似单片机上移植, 通用性强。

**关键词:** C8051F34x; USB Bootloader; 远程升级

中图分类号: TP368.1

文献标识码: A

文章编号: 1674-7720(2011)23-0056-04

## The design of USB Bootloader based on C8051F34x MCU

Liu Yunlong<sup>1</sup>, Zhang Weixia<sup>2</sup>

(1. Guangzhou Haige Communications Group Co., Ltd, Guangzhou 510663, China;

2. GRG Banking Equipment Co., Ltd, Guangzhou 510663, China)

**Abstract:** This paper describes the design of USB bootloader based on Silicon Labs' C8051F34x serial MCU, that possesses the ability to receive new application firmware through USB or COM. Firstly, it introduces the USB Bootloader of MCU. Then, it expands the design of USB bootloader firmware and the design of application firmware, as well as the problems of design process that need attentions. Lastly, the bootloader software is designed using Visual C++ 6.0, achieving remote update the APP FW. This USB bootloader is very universal and can be easily ported to other architecture like MCUs.

**Key words:** C8051F34x; USB Bootloader; remote update

Bootloader(以下简称 BL)是一段引导程序, 在单片机上电或复位后在应用程序(以下简称 APP)之前先运行, 来判断当前是否需要进入升级状态。如果不需要升级, 就直接跳转到 APP 运行; 如果需要升级, 首先擦除旧的 APP, 然后通过某种通信接收 APP 固件程序, 同时写入 Flash 中。

BL 固件程序能以多种方式获取数据, 包括串口、并口、I<sup>2</sup>C、SPI、USB 等, 但是从实际使用和成熟度来看, 使用串口无疑是最方便的。如今, USB 总线凭借其方便、快速、灵活、稳定、应用范围广等优点被广泛地应用、发展和普及, 使用 USB 进行数据传输是一种趋势。本文设计的 BL 主要基于 USB 通信, 同时考虑到模块兼容, 保留了串口通信。

一般来说, 一个 BL 应该能够完成以下功能: (1) 通

《微型机与应用》2011 年第 30 卷第 23 期

过某种通信收发数据; (2) 擦除并升级 APP 应用固件程序; (3) 判断 APP 固件的完整性; (4) APP 与 BL 的中断跳转问题。而基于 USB 通信的 BL, 除了要完成一般 BL 的功能, 还需要考虑 BL 与 APP 共用除 USB 中断外的 USB 一般处理函数等问题。下面对 BL 固件程序设计、APP 固件程序设计以及上层软件设计进行详细介绍。

### 1 USB Bootloader 设计

#### 1.1 硬件平台

本文设计的 USB BL 是基于 Silicon Labs 公司 C8051F34x 系列单片机实现的; C8051F34x 器件是完全集成的混合信号片上系统型 MCU, 具有片内上电复位、VDD 监视器、电压调整器、看门狗定时器、时钟丢失检测器、时钟振荡器、USB、SMBus/I<sup>2</sup>C、UART、SPI、定时器、I/O、多达 4 352 B 片内 RAM 和 64 KB 的片内 Flash 存储

欢迎网上投稿 [www.pcachina.com](http://www.pcachina.com) 63

## 网络与通信 Network and Communication

器,Flash 存储器还具有在系统重新编程的能力,可用于非易失性数据存储,并允许现场更新 8051 固件。

C8051F34x 器件集成了一个完整的全速/低速 USB 功能控制器,刚开始设计时采用 C8051F34x 自带的 USB,但是静电测试不符合要求,最终选用了 C8051F34x 和 PDIUSBD12 组合,解决了静电问题。

PDIUSBD12 是一款性价比很高的 USB 器件,它符合 USB1.1 版规范,可与任何外部微控制器实现高速并行接口(2 Mb/s),具有良好的 EMI 特性,高于 8 kV 的在片静电防护电路等,详细资料请查询参考文献[3]。

### 1.2 BL 和 APP 固件的地址分配

USB BL 预计将占用 8 KB 的地址空间,从 0x0000h 到 0x1FFFh,包括 USB BL 固件程序本身和用来判断程序是运行 APP 还是 BL 的一段签名程序。BL 与 APP 地址空间分配如图 1 所示,APP 固件程序存放的地址空间从 0x2000h 开始。

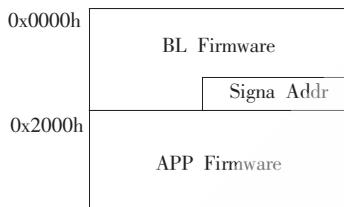


图 1 BL 与 APP 地址空间分配

### 1.3 BL 和 APP 的自动跳转

在程序中,设置一个设备模式标志位,用来判断程序是应该运行在 BL 中还是在 APP 中,此标志位存储在 RAM 的一个固定地址 2F7h 处。本文设计的 BL,DEVICE\_MODE 为设备模式标志位,其值等于 1 时为 BL\_MODE(BL 模式),其值等于 0 为 APP\_MODE(APP 模式)。

有两种情况设备模式为 BL 模式,可以下载更新 APP 固件程序:一是在指定的签名地址处没有指定的两字节的签名;二是 Flash 写错误,在 C8051F34x 单片机中,Flash 读地址超出了用户代码空间,即 MOVC 操作的地址大于 0xFBFF,发生 Flash 错误复位后,FERROR 位(RSTSRC.6)被置位。

上电后是否运行 APP 固件程序通过第一种情况判断。当需要更新 APP 固件程序时,造成 Flash 写错误,通过第二种情况进入 BL 模式,开始更新程序,如图 2 所示。

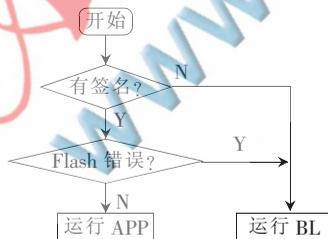


图 2 BL 和 APP 的自动跳转

### 1.4 中断重定位

一般情况下,MCU 中断向量分布在复位(0x0000)以后,位于低地址空间。由于 BL 程序占据了此段空间,因此,除了 USB0 中断(中断序列列表第 8 号中断)和串口中断(中断序列列表第 5 号中断),其他所有的中断(C8051F34x 共有 16 个通用中断)都需要做中断二次映射。也就是说,需要在原中断向量入口地址处手动添加

二次跳转函数,使新的中断向量指向用户的中断程序,这样才能保证正常运行 APP 固件程序的中断程序。具体的跳转地址由 APP 固件程序起始地址决定,这一部分在 START51.A51 中通过编写函数来完成。假设固件的起始地址设置为 0x2000H,则中断跳转的实现过程如下。

首先定义几个常量:

```
HW_INTVEC_TABLE EQU 0003h
HW_INTVEC_SEPARATION EQU 8
INTVEC_TABLE EQU START_APPLICATION+3
INTVEC_SEPARATION EQU 8
START_APPLICATION EQU 2000h
```

中断向量重映射:

```
CSEG AT HW_INTVEC_TABLE +
(HW_INTVEC_SEPARATION*INT_NUM)LJMP
INTVEC_TABLE+(INTVEC_SEPARATION
*INT_NUM)
```

//以定时器 2 中断为例(第 5 号中断)

```
CSEG AT0003h+8h*5h=002Bh
LJMP 2000h+0003h+8h*5h=202Bh
```

### 1.5 中断处理函数

8 号 USB 中断由于同时要被 BL 固件和 APP 固件调用,因此不能对其进行二次跳转,而通过共享的 USB 库文件中的 USB\_ISR 主中断处理函数进行处理,通过 DEVICE\_MODE 判断当前设备处于 BL 模式或 APP 模式来自动地二次跳转到 BL 固件或 APP 固件的 USB 中断处理函数处,如图 3 所示。

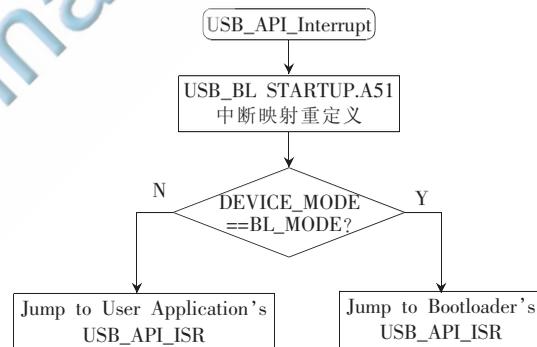


图 3 BL 中断函数分流处理

需要注意的是,4 号串口中断同样要被两者所调用,因此对 4 号中断的处理与 8 号中断相同。先由 4 号中断入口地址跳转到原地址处,然后在此地址处根据设备模式进行中断分流,决定是到 BL 还是到 APP 的中断处理函数处。

### 1.6 USB BL 命令函数

BL 固件程序中的命令函数如表 1 所示。

- (1)Erase Page:擦除 APP 固件程序和签名;
- (2)Write Page:将 APP 固件程序的 HEX 文件写入 Flash;
- (3)Write Signature:APP 固件程序写 Flash 成功后,将签名写入指定的地址处,表示 APP 固件已经存在于

《微型机与应用》2011 年第 30 卷第 23 期

Flash 中;

(4)Get Version:取 BL 程序的版本号。

表 1 BL 固件程序中的命令函数

命令函数	描述
Erase Page	擦除 Flash 页
Write Page	写 Flash 页
Write Signature	写签名
Get Version	取版本, BL 版本号

## 2 APP 固件程序设计

使用 USB BL, 需要对 APP 固件程序进行一些添加和修改。

(1) 由于 BL 占用了 0x0000~0x1FFF 的空间, APP 固件程序是以 0x2000h 作为起始地址的, 这样就需要修改 APP 程序的偏移量。

①修改 STARTUP.A51 文件, 把“CSEG AT 0”变为“CSEG AT 2000h”;

②点击 Project->Options for Target ‘Target1’, 点击 C51 项目栏, 把 Interrupt vectors address 栏选中, 内容改为 0x2000, 点击 BL51 Locate 项目栏, 将 code 项改为 0x2000。

(2)APP 固件程序应该具备从 APP 转到 BL 的功能, 需要增添以下代码:

①main()主函数中增加接收更新 APP 固件程序的命令字以及对此命令的处理代码, 使用 BOOTLOAD\_REQ() 命令来触发一次 Flash 读复位, 使器件进入 BL 模式;

②在 STARTUP.A51 文件增加以下代码:

```
//造成 Flash 写错误地址定义
PUBLIC BOOTLOAD_REQ
BOOTLOAD_REQ EQU 0FFFFh
```

③在头文件中添加函数声明:

```
void BOOTLOAD_REQ(void)
```

(3)去掉与 BL 重复的 USB 通信函数部分, 特别是要去掉 Control\_USB() 函数(该函数主要完成设备请求处理函数), 因为此函数在 BL 中已经实现, 并且用绝对地址固定, 应用程序只需跳转到固定的绝对地址处即可, 修改如下:

①在 STARTUP.A51 文件增加以下代码:

```
//control_usb 地址定义
PUBLIC Control_USB
//USB 通信产生的外部中断 1 在 APP 固件的入口地址
Control_USB EQU 1300h
```

②注意需要在 BL 中先定义 Control\_USB 的入口地址, 方法如下: 在 BL 工程下, Project->Options for Target ‘Target1’, 点击 BL51 Locate 项目栏, 将 code 项修改为

```
?PR?VCONTROL_USB?BOOTLOADER_F340(0x1300);
```

(4)修改 USB 中断处理函数: 由于 USB 控制器采用 PDIUSB12, 其中断引脚 INT\_N 接 C8051F34x 的 P0.7 引

脚, 且该引脚被配置为外部中断 1, 电平触发方式, 低电平有效。因此应在外部中断 1 中断处理函数中获取 USB 中断源并进入相应的子程序进行处理。

(5)USB 设备的枚举过程在 BL 中完成, 因此 PID、VID 是 BL 程序所决定的, 需要在 BL 中改变此处的值以适应自己的模块。

(6)保护被 BL 使用的位, DEVICE\_MODE 的位地址, 在 STARTUP.A51 文件中, 在宏定义和代码段开始之前增加以下的代码:

```
//Last bit in bit-addressable space(2F.7h)
MEM_DEVICE_MODE BIT 07Fh
PUBLIC DEVICE_MODE
BSEGATMEM_DEVICE_MODE
DEVICE_MODE: DBIT 1
```

## 3 远程在线下载

### 3.1 下载步骤

(1)从 APP 切换到 BL。此时, 程序正常运行在 APP 模式, 发送更新程序命令, 致使 Flash 写错误进入 BL 模式。

(2)擦除 Flash。在 BL 模式下, 发送擦除 Flash 命令, 擦除签名和 APP 固件程序, 返回成功 ERASE\_OK。

(3)写 Flash。擦除 Flash 成功后, 可以将新的 APP 固件程序的 HEX 文件写进 Flash。校验失败, 返回 WRITE\_FAILED, 成功返回 WRITE\_OK。

(4)写签名。写 Flash 成功后, 将 2 B 的签名写到指定的地址处, 表示 APP 固件已经存在于 Flash 中。

(5)从 BL 切换到 APP。写签名成功后, 使程序跳转到 APP 固件程序处执行。

### 3.2 上层软件设计

本文使用 VC++6.0 开发了 BL 上层软件, 如图 4 所示。

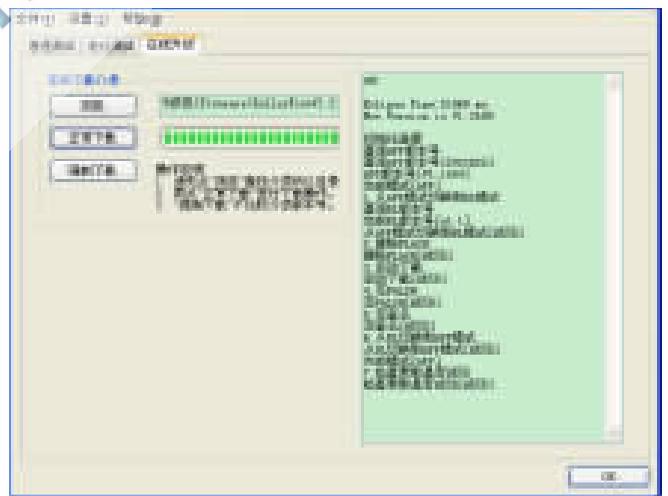


图 4 远程在线下载工具

在线下载时, 有两种方式: (1)正常下载, 这是常用的一种方式, 这种下载方式在下载前和下载后会进行 APP 固件程序版本比较, 如果是不同版本的程序, 可以进行升级, 如果是同一版本的程序, 直接返回成功; (2)强制

下载,这种下载方式不进行 APP 固件程序版本比较,点击即可进行升级,一般在 APP 固件程序调试时多次下载使用。

在线下载使用方法:首先点击“浏览”按钮,查找到用于升级的新版本的 HEX 文件;再点击“正常下载”或“强制下载”进行程序升级;然后在右边查看返回结果,看升级是否成功。

### 3.3 设计注意点

在 APP 转 BL 以及 BL 转 APP 时,需要考虑 USB 枚举时间,枚举成功后才能正常地发送和接收。远程下载过程中,需要考虑一些异常情况,如 PC 主机死机、模块 CPU 死机、死循环或复位等,针对这些情况,本设计均作了冗错处理。

一个好的 BootLoader 程序应该具有良好的可维护性并可以正确处理异常情况,不会因为意外情况引起系统的损坏和崩溃。本文结合实际应用,设计了一个实用的 USB Bootloader。经大量测试和实际应用,可满足开发和维护人员的要求。

### 参考文献

- [1] Silicon Labs. USB Bootloader with shared USB [DB/OL]. Xpress Library, 2008.2.
- [2] 潘琢金,译.C8051F340/1/2/3/4/5/6/7 全速 USB FLASH 微控制器数据手册[Z].新华龙电子有限公司,2006.01.
- [3] 周立功.PDIUSBD12 USB 固件编程与驱动开发 [M].北京:北京航空航天大学出版社,2002.
- [4] 王朔,李刚.USB 接口器件 PDIUSBD12 的接口应用设计[J].单片机与嵌入式系统应用,2002(1).
- [5] 缪德芳,李绍胜.单片机 Bootloader 设计与实现[J].中国科技论文在线.
- [6] 虹信公司.在 PIC18 单片机中使用 BootLoader[J].单片机与嵌入式系统应用,2005(12).

(收稿日期:2011-07-12)

### 作者简介:

刘云龙,男,1983 年生,硕士,主要研究方向:通信技术。

张伟霞,女,1982 年生,硕士,主要研究方向:机电自动控制。