TM4C1290NCZAD Bootloader

目录

[1 引言 3](#_Toc133308953)

[1.1 背景 3](#_Toc133308954)

[1.2 术语和缩写 3](#_Toc133308955)

[1.3 参考文献 3](#_Toc133308956)

[2 概述 4](#_Toc133308957)

[2.1 系统体系结构 4](#_Toc133308958)

[2.2 软件需求概述 4](#_Toc133308959)

[3 约定及假定 5](#_Toc133308960)

[3.1 硬件约束 5](#_Toc133308961)

[3.1.1 硬件开发平台约束 5](#_Toc133308962)

[3.1.2 开发环境约束 5](#_Toc133308963)

[3.1.3 其他硬件约束 5](#_Toc133308964)

[3.2 软件约束 5](#_Toc133308965)

[3.2.1 软件开发平台约束 5](#_Toc133308966)

[3.2.2 操作系统系统约束 6](#_Toc133308967)

[3.2.3 其他软件约束 6](#_Toc133308968)

[3.3 设计与实现的约束 6](#_Toc133308969)

[3.4 设计与实现的假定 6](#_Toc133308970)

[4 软件需求 6](#_Toc133308971)

[4.1 功能需求 6](#_Toc133308972)

[4.1.1 通信数据管理 6](#_Toc133308973)

[4.1.2 Flash数据管理 7](#_Toc133308974)

[4.1.3 配置管理 7](#_Toc133308975)

[4.2 接口需求 8](#_Toc133308976)

[4.2.1 RS485接口需求 8](#_Toc133308977)

[4.2.2 SPI接口需求 8](#_Toc133308978)

[4.3 属性需求 9](#_Toc133308979)

[4.3.1 性能需求 9](#_Toc133308980)

[4.3.2 可靠性需求 9](#_Toc133308981)

[4.3.3 安全性需求 9](#_Toc133308982)

[5 通信协议设计 9](#_Toc133308983)

[5.1 RS484协议 9](#_Toc133308984)

[5.2 SPI协议 11](#_Toc133308985)

[6 内存设计 11](#_Toc133308986)

# 引言

## 背景

本文档描述了APM2锂电模块Bootloader BMS软件需求规格部分，此软件隶属于APM2锂电项目。软件主要完成的功能为：MSP430FR5962芯片的APP更新功能。

## 术语和缩写

BMS: Battery Management System(电池管理系统)

TM4C：TM4C1290NCZAD芯片

MSP：MSP430FR5962芯片

## 参考文献

《TM4C1290NCZAD Data Sheet》

《MSP430FR5962 Data Sheet》

# 概述

## 系统体系结构



本Boot只针对PC到锂电模块实现一对一升级。

## 软件需求概述

• 通过SPI接收新的固件数据

• 对接收到的数据进行校验

• 将校验通过的数据写入MSP的FRAM中

• 支持FRAM擦除和编程

## 软件启动流程



# 约定及假定

## 硬件约束

### 硬件开发平台约束

CPU：MSP430FR5962

RAM：256KB

FLASH：1MB

|  |  |  |
| --- | --- | --- |
| **序号** | **端口** | **数量** |
| 1 | SPI | 1 |

### 开发环境约束

编程软件运行环境：Windows10

编程语言： C

### 其他硬件约束

无

## 软件约束

### 软件开发平台约束

软件编辑环境：

Code Composer Studio 10.3.0；

编译器：TI V20.2.4

编程语言：

C/C++

### 操作系统系统约束

Windows10

### 其他软件约束

无

## 设计与实现的约束

无

## 设计与实现的假定

无

# 软件需求

## 功能需求

### 通信数据管理

|  |  |
| --- | --- |
| 功能描述 | 提供统一数据写入接口，并具备数据过滤功能 |
| 输入 | SPI报文 |
| 输出 | 解析处理后的缓存数据 |
| 性能要求 | a. 数据接收：接收各个数据采集模块通过总线发过来的数据，进行解析  b. 数据发送：将解析到的各个模块的数据存储到数据缓冲模块。  c. 数据过滤：对于各个模块发送过来的数据进行合法性判断，判断数据是否正常，保证各个模块能够收到合法数据。 |
| 响应时间 | / |

### Flash数据管理

|  |  |
| --- | --- |
| 功能描述 | 解析SPI数据，封装到结构体，提供给各个模块的高速访问数据接口。 |
| 输入 | SPI数据 |
| 输出 | Flash内存、SPI结构体 |
| 性能要求 | b. 数据输出:对接收到的数据转到SPI通信结构体 |
| 响应时间 | / |

## 接口需求

### SPI接口需求

1）接口描述：

传输次序：所有数据项均先传送字节高位，后传送字节低位。

采用的vertiv自有协议：协议为《APM2 锂电SPI协议V0.0》

2) 性能要求：

能够按照通信协议实现正确的数据交互。

## 属性需求

### 性能需求

（1）软件CPU平均占用率小于50%；

（2）软件flash占用小于16K。

### 可靠性需求

通过看门狗监视APP是否能正常运行，如果更新异常能强制复位系统使得回到Bootloader中。

### 安全性需求

无

# 通信协议设计

## RS484协议

Bootloader 与上位机之间使用 UART 串行通信协议。通信协议使用 8 位无奇偶校验位、1 位起始位和 1 位停止位，波特率为 19200。

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 升级请求，写0x6000寄存器（应用程序，未进入bootloader） | PC | RackID(bit7~bit5)+ShlefID(bit4~bit0) | 0x06 | 0x6000 | 0x7E | CRC(L-H) |  |  |
| Bat | RackID(bit7~bit5)+ShlefID(bit4~bit0) | 0x06 | 0x6000 | 0x7E | CRC(L-H) |  |  |
| 2 | 读0x6001寄存器状态（确认是否能够升级） | PC | RackID(bit7~bit5)+ShlefID(bit4~bit0) | 0x03 | 0x6001 | 0x01 | CRC(L-H) |  |  |
| Bat | RackID(bit7~bit5)+ShlefID(bit4~bit0) | 0x03 | 0x02 | 0xAA / 0xBB | CRC(L-H) |  |  |
| 3 | 发送芯片选择信息，写0x6002寄存器 | PC | RackID(bit7~bit5)+ShlefID(bit4~bit0) | 0x06 | 0x6002 | 0xF0 / 0xF1 / 0xF2 | CRC(L-H) |  |  |
| Bat | RackID(bit7~bit5)+ShlefID(bit4~bit0) | 0x06 | 0x6002 | 0xF0 / 0xF1 / 0xF2 | CRC(L-H) |  |  |
| 4 | 握手，读0x6001寄存器状态（确认已进入bootloader） | PC | RackID(bit7~bit5)+ShlefID(bit4~bit0) | 0x03 | 0x6001 | 0x01 | CRC(L-H) |  |  |
| Bat | RackID(bit7~bit5)+ShlefID(bit4~bit0) | 0x03 | 0x02 | 0xCC / 0x33 | CRC(L-H) |  |  |
| 5 | 发送下载文件的大小及每包发送数据的大小，写0x6003等寄存器 | PC | RackID(bit7~bit5)+ShlefID(bit4~bit0) | 0x10 | 0x6003 | 0x03 | 0x06 | 下载文件大小size && 每包byte数n | CRC(L-H) |
| Bat | RackID(bit7~bit5)+ShlefID(bit4~bit0) | 0x10 | 0x6003 | 0x03 | CRC(L-H) |  |  |
| 6 | 读0x6003等寄存器状态（确认升级信息） | PC | RackID(bit7~bit5)+ShlefID(bit4~bit0) | 0x03 | 0x6003 | 0x03 | CRC(L-H) |  |  |
| Bat | RackID(bit7~bit5)+ShlefID(bit4~bit0) | 0x03 | 0x06 | 下载文件大小size && 每包byte数n | CRC(L-H) |  |  |
| 7 | 发送升级数据包，写0x6006等寄存器 | PC | RackID(bit7~bit5)+ShlefID(bit4~bit0) | 0x10 | 0x6006 | n/2 | n | 每包数据 | CRC(L-H) |
| Bat | RackID(bit7~bit5)+ShlefID(bit4~bit0) | 0x10 | 0x6006 | n/2 | CRC(L-H) |  |  |
| 8 | 读0x6001寄存器状态(确认是否接收到数据) | PC | RackID(bit7~bit5)+ShlefID(bit4~bit0) | 0x03 | 0x6001 | 0x01 | CRC(L-H) |  |  |
| Bat | RackID(bit7~bit5)+ShlefID(bit4~bit0) | 0x03 | 0x02 | 0x44 / 0x45 | CRC(L-H) |  |  |
|  | 重复10-11步骤，直到升级数据发送完毕。备注：不足包数的，按剩余的byte数发。 | **…** |  |  |  |  |  |  |  |
|  | 等待2s，升级时间 |  |  |  |  |  |  |  |  |
| 9 | 读0x6001寄存器状态(确认是否升级完成) | PC | RackID(bit7~bit5)+ShlefID(bit4~bit0) | 0x03 | 0x6001 | 0x01 | CRC(L-H) |  |  |
| Bat | RackID(bit7~bit5)+ShlefID(bit4~bit0) | 0x03 | 0x02 | 0xDD / 0x55 | CRC(L-H) |  |  |

## SPI协议

TM4C与MSP430之间使用SPI通信协议，采用内部协议；

# 内存设计

TM4C具有1MB闪存，对于TM4C的闪存的系统管理方案如下：

0~16K（0x00000000 ~ 0x00003FFF）：Bootloader Code 区域;

16~32K (0x00004000 ~ 0x00007FFF)：APP 参数存储区域：

32~1024K(0x00008000 ~ 0x000100000)：APP Code区域；

对于APP&Boot共用参数如下：

// 应用程序存储区

#define addr\_flash\_APP\_run\_flag 0x4004 // APP运行标志

#define addr\_flash\_Rack\_ID 0x4008      // RackID

#define addr\_flash\_BMS\_SN 0x400C       // SN

#define addr\_flash\_Project\_Code 0x4028 // 应用程序版本号

#define addr\_flash\_Shelf\_ID 0x4030     // ShelfID

上述参数是需要Boot在APP参数区域读取到的数据，为共享flash地址；

对于APP区的特定地址：

// Boot程序区定义

#define APP\_START\_ADDRESS 0x8000    // APP Start ADD 应用程序起始地址

#define VTABLE\_START\_ADDRESS 0x8000 // APP Vtable ADD 中断起始地址

#define APP\_SAVE\_ADDRESS 0x80000    // APP Download ADD 应用程序下载区起始地址