|  |  |
| --- | --- |
| FT8836 ROMBOOT设计文档 | |
| Project name | FT8836 |
| Document ref |  |
| Version | V0.1 |
| Release date | 3 Jan 2017 |
| Owner | tangsujun |
| Classification | INTERNAL USE ONLY |
| Distribution List |  |
| Approval |  |

This document contains information proprietary to FocalTech Systems, Ltd., and may not be reproduced, disclosed or used in whole or part without the express written permission of FocalTech Systems, Ltd.

Copyright © 2017, FocalTech Systems, Ltd

All rights reserved

Shenzhen, Gungdong, P.R. China

ZIP :518057

T +86 755 26588222

F +86 755 26712499

E <support@focaltech-systems.com>

<www.focaltech-systems.com>

Revision History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date** | **Version** | **List of changes** | **Author** | **Approved by** |
| 3 Jan ,2017 | 0.1 | Initial draft | Tangsujun |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Table of Contents

[1 概述 1](#_Toc471308481)

[2 接口方式 2](#_Toc471308482)

[3 升级流程 2](#_Toc471308483)

[3.1 BOOTLOADER基本流程 2](#_Toc471308484)

[3.2 HOST基本流程 6](#_Toc471308485)

[3.3 APP合法性检测流程 8](#_Toc471308486)

[4 Pram 结构 9](#_Toc471308487)

[4.1 Pram Map 9](#_Toc471308488)

[APP 验证信息(App info) 9](#_Toc471308489)

[5 Spi-flash 结构 10](#_Toc471308490)

[5.1 Spi-Flash Map 10](#_Toc471308491)

[5.2 FW 配置信息(Configuration) 10](#_Toc471308492)

[6 通讯协议 11](#_Toc471308493)

[6.1 通讯的基本步骤 11](#_Toc471308494)

[6.1.1 读ID 11](#_Toc471308495)

[6.1.2 解锁停止搬运 11](#_Toc471308496)

[6.1.3 发送数据报 11](#_Toc471308497)

[6.1.4 校验 11](#_Toc471308498)

[6.1.5 读操作 12](#_Toc471308499)

[6.1.6 启动app 12](#_Toc471308500)

[6.1.7 复位 12](#_Toc471308501)

[6.1.8 查询状态 13](#_Toc471308502)

[6.1.9 读Vendor ID 13](#_Toc471308503)

[7 基本时序图 14](#_Toc471308504)

[7.1.1 上电时序 14](#_Toc471308505)

[7.1.2 Reset 时序 14](#_Toc471308506)

[7.1.3 I2C 时序图 15](#_Toc471308507)

[8 Upgrade使用方式 15](#_Toc471308508)

[9 启动时间 15](#_Toc471308509)

# 概述

该文文件描述了FT8836的内部romboot程序工作流程。当FT8836启动后，首先运行内部romboot程序。romboot程序启动后，会从host接收fw到pram,或者从spi-flash导入fw到pram,然后remap运行app程序。

romboot将实现如下的功能:

* Host导入程序到pram
  + 支持I2C协议
* 把app从spi-flash导入到pram
* 可配置system时钟 24/ 48MHz, 默认24MHz
* 可配置spi时钟分频 (8 , 16 ,32), 默认DIV4
* 可配置XSI接口时钟分频 (2 , 4 ,8 ,16)
* Remap，启动app程序
* 软复位或wdt复位时，校验pram中app的有效性。
* 复位后，发中断告诉host，host读状态位，做相应处理。
* 主机通过读报点帧数据为全0xEF，识别是romboot中断, app丢失
* I2C接口slave地址可选

# 接口方式

romboot采用标准I2C接口方式，FT8836作为slave 端。

* slave address 有二种方式来源：选择优先级
  + Flash中地址有效则从Flash配置
  + Flash无效时用默认地址0x70

# 升级流程

## BOOTLOADER基本流程







图表1Bootloader程序流程

## HOST基本流程

下图为HOST 操作(TOOLS)的基本流程, 具体通讯协议请参照[通讯协议](#_通讯协议)。



图表2 Host下载pram操作流程



## APP合法性检测流程

下图为romboot进行APP合法性检测的基本流程, 具体pram的结构请参照[PRAM结构图](#_App合法性检测)。



图表 5 APP 合法性检测流程

# Pram 结构

## Pram Map

下图是Flash的一个整体结构图：

由于romboot阶段，remap=0，因此地址要加上偏移量0Xfce800（romboot内部处理，与host无关）.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **0** | **1** |  | **…** | **…** | **1022** | **1023** |
| **0k=0x0000** |  | | | | | | |
| **0x100** | **App info（32B）** | | **FW配置参数备份区(0x80)** | | |  |  |
| **1k=0x0400** | **App <=70K** | | | | | | |
| **…** |
| **…** |
| **10k = 0x2800** |
| **11k = 0x2C00** |
| **…** |
| **…** |
| **…** |
| **66k=0x10800** |
| **67k=0x10C00** |
| **68k=0x11000** |  | | | | | | |
| **69k=0x11400** |  | | | |  | | |
| 2. App代码验证信息存放在flash(0x1100~0x1120) | | | | | | | |
| 3. App代码验证信息存放在pram(0x0100~0x011F) | | | | | | | |
| 4. FW配置信息备份存放在pram(0x0120~0x019F) | | | | | | | |

## APP 验证信息(App info)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0x00 | 0x01 | 0x02 | 0x03 | 0x04 | 0x05 | 0x06 | 0x07 |
| Code length | | Negation of code length | | App Part1 ECC | | Negation of App part1 ECC | |
| 0x08 | 0x09 | 0x0A | 0x0B | 0x0C | 0x0D | 0x0E | 0x0F |
| App Part2 ECC | | Negation of App part2 ECC | | Vendor | ~Vendor | FW Version | ~FW Version |
| 0x10 | 0x11 | 0x12 | 0x13 | 0x14 | 0x15 | 0x16 | 0x17 |
| Upgrade flag1 | Upgrade flag2 | CodeLenH | | ~CodeLenH | |  |  |
| 0x18 | 0x19 | 0x1A | 0x1B | 0x1C | 0x1D | 0x1E | 0x1F |
|  |  |  |  |  |  | APP\_ID\_H | APP\_ID\_L |

Ecc表示APP的所有数据的CRC校验,具体数据如何使用请参照[基本流程](#_基本流程)。

# Spi-flash 结构

## Spi-Flash Map

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **0** | **1** | **…** | | | **1022** | **1023** |
| **0k=0x0000** | **Lcd Init Code** | | | | | | |
| **1k=0x0400** |
| **2k=0x0800** |
| **3k=0x0C00** |  |  |  |  | **FW配置参数区(0x80Byte)** | | |
| **4k=0x1000** |  |  |  |  |  |  |  |
| **0x1100** | **App info（0x20Byte）** | | **FW配置参数备份区(0x80Byte)** | | |  |  |
| **5k=0x1400** |  | | | | | | |
| **…** |
| **…** |
| **20k=0x5000** |
| **21k=0x5400** |
| **…** |
| **…** |
| **70k=0x11800** |
| **71k=0x11C00** |
| **72k=0x12000** |
| **73k=0x12400** |
| 1. lcd init code信息存放在flash(0x0000~0x0F7F) | | | | | | | |
| 2. FW配置信息存放在flash(0x0F80~0x0x0FFF) | | | | | | | |
| 3. App验证信息存放在Flash(0x1100~0x111F) | | | | | | | |
| 4. FW配置信息备份存放在Flash(0x1120~0x119F) | | | | | | | |

## FW 配置信息(Configuration)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0x00 | 0x01 | 0x02 | 0x03 | 0x04 | 0x05 | 0x06 | 0x07 |
| I2c address | ~ I2c addr | iovoltage | Negation of Iovoltage | Panel Vendor ID | Negation of Panel Vendor ID | Spi clk Select | ~Spi clk Select |
| 0x08 | 0x09 | 0x0A | 0x0B | 0x0C | 0x0D | 0x0E | 0x0F |
| System Clk Select | ~System Clk select | LCD XBUS Clk | ~LCD XBUS Clk | SPI\_IO\_MODE | ~SPI\_IO\_MODE |  |  |
| 0x10 | … | | | | | | 0x1f |
| 0x20 | … | | | | | | 0x3f |
| Project Code(ASCII) | | | | | | | |
| 0x40 | … |  |  |  |  |  | 0x4f |
| Customer code(ASCII) | | | | | | | |

# 通讯协议

定义：

Slv\_addr -------- I2C的slave address

R --------- 表示值1，I2C 读

W --------- 表示值0， I2C 写

## I2C通讯的基本步骤

### 读ID

FW进行Upgrade时先读ID，否则不能进行其他操作。

I2C命令格式如下：

第一步：发送命令字

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Start | Slv\_addr + W | 0x90 | 0x00 | 0x00 | 0x00 | Stop |

第二步：读ID号

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Start | Slv\_addr + R | 0x88 | 0x36 | Stop |
|  |  |  |  |  |

由到读ID号后，后清除数据接收的ecc值

W： 90 00 00 00

R： 02 (Boot返回 0x88 0x36)

### 解锁停止搬运

发送55

|  |  |  |  |
| --- | --- | --- | --- |
| Start | Slv\_addr + W | 0x55 | Stop |

收到55 ,后清除数据接收的ecc值

W： 55

### 发送数据报

0xAE: pram操作

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Start | Slv\_addr + W | 0xAE | Add\_H | Add\_M | Add\_L | Len\_H | Len\_L | Data1 | ... | Data | Stop |

W: AE AH AM AL LH LL D1 … Dn (xx 表示总长度，应该是n+6)

在这一步，Host可以根据档的大小，可以拆分成N个这样的数据报，数据报长度不能超过240 ， Table1写操作各字段描述

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Default** |  |  |
| Add\_H | 程序开始地址的高字节 | 1 Byte |  |  |
| Add\_M | 程序开始地址的中间字节 | 1Byte |  |  |
| Add\_L | 程序开始地址的低字节 | 1Byte |  |  |
| Len\_H | 发送的Data的长度的高字节 | 1Byte |  |  |
| Len\_L | 发送的Data的长度的低字节 | 1Byte |  |  |

### 校验

第一步：

|  |  |  |  |
| --- | --- | --- | --- |
| Start | Slv\_addr + W | 0xcc | Stop |

第二步：

|  |  |  |  |
| --- | --- | --- | --- |
| Start | Slv\_addr + R | Ecc | Stop |

Ecc是所有Data(程序代码)的异或校验。

W: CC R 01 （Boot返回校验码）

Host在发送Data的时候，自己也会计算整个程序代码的异或校验，当读回Ecc的时候，两个校验将会做个对比，如果对比是匹配的，说明发送成功，否则说明是失败的。失败后，可以从6.1.2章节重新开始。

### 读操作

Host 可以读pram的数据出来，每个数据把不能超过240字节，读取总长度不能超过70K

0x85：pram操作

第一步：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Start | Slv\_addr + W | 0x85 | ADD\_H | ADD\_M | ADD\_L | Stop |

第二步：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Start | Slv\_addr + R | Data1 | Data2 | …… | DataN | Stop |

W: 85 Ah Am Al: 先设置基址地址

R: N (Boot返回pram数据，地址自加)

Table2 读操作各字段描述

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Default** |  |  |
| Add\_H | 程序开始地址的高字节 | 1 Byte |  |  |
| Add\_M | 程序开始地址的中间字节 | 1Byte |  |  |
| Add\_L | 程序开始地址的低地址 |  |  |  |
| Data | 程序代码 |  |  |  |
|  |  |  |  |  |

### 启动app

Rowboot接收到此命令时，remap，启动app程序

|  |  |  |  |
| --- | --- | --- | --- |
| Start | Slv\_addr + W | 0x08 | Stop |

W: 08

### 复位

发送此命令可以让FT8836Reset，重新运行程序

第一步：

|  |  |  |  |
| --- | --- | --- | --- |
| Start | Slv\_addr + W | 0x07 | Stop |

W: 07

### 查询状态

第一步：

|  |  |  |  |
| --- | --- | --- | --- |
| Start | Slv\_addr + W | 0xD0 | Stop |

第二步：

|  |  |  |  |
| --- | --- | --- | --- |
| Start | Slv\_addr + R | state | Stop |

W: D0 R 01 (Boot返回状态标志)

### 读Vendor ID

第一步：

|  |  |  |  |
| --- | --- | --- | --- |
| Start | Slv\_addr + W | 0xA8 | Stop |

第二步：

|  |  |  |  |
| --- | --- | --- | --- |
| Start | Slv\_addr + R | Upgrade Version | Stop |
|  |  |  |  |

W: A8 R 01 (Boot返回状态 Vendor ID号)

## SPI通讯的基本步骤

### 读ID

FW进行Upgrade时先读ID，否则不能进行其他操作。

SPI命令格式如下：

输入：发送命令字

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Start | 0x90 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 | Stop |

输出：ID号

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| X | X | X | X | X | 0x88 | 0x36 | X |
|  |  | |  |  | |  | |

由到读ID号后，后清除数据接收的ecc值

W： 90 00 00 00 R： 02 (Boot返回 0x88 0x36)



### 解锁停止搬运

发送55

|  |  |  |
| --- | --- | --- |
| Start | 0x55 | Stop |

收到55 ,后清除数据接收的ecc值

W： 55



### 发送数据包

0xAE: pram操作

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Start | 0xAE | Add\_H | Add\_M | Add\_L | Len\_H | Len\_L | Data1 | ... | Data | Stop |

W: AE AH AM AL LH LL D1 … Dn (xx 表示总长度，应该是n+6)



在这一步，Host可以根据档的大小，可以拆分成N个这样的数据报，数据报长度不能超过240 ， Table1写操作各字段描述

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Default** |  |  |
| Add\_H | 程序开始地址的高字节 | 1 Byte |  |  |
| Add\_M | 程序开始地址的中间字节 | 1Byte |  |  |
| Add\_L | 程序开始地址的低字节 | 1Byte |  |  |
| Len\_H | 发送的Data的长度的高字节 | 1Byte |  |  |
| Len\_L | 发送的Data的长度的低字节 | 1Byte |  |  |

### 校验

第一步：

|  |  |  |  |
| --- | --- | --- | --- |
| Start | 0XCC | 0x00 | Stop |

第二步：

|  |  |  |  |
| --- | --- | --- | --- |
| X | X | Ecc | X |

Ecc是所有Data(程序代码)的异或校验。

W: CC R 01 （Boot返回校验码）



Host在发送Data的时候，自己也会计算整个程序代码的异或校验，当读回Ecc的时候，两个校验将会做个对比，如果对比是匹配的，说明发送成功，否则说明是失败的。失败后，可以从6.2.2章节重新开始。

### 读操作

Host 可以读pram的数据出来，每个数据包不能超过240字节，读取总长度不能超过70K

0x85：pram操作

DI：

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Start | 0x85 | ADD\_H | ADD\_M | ADD\_L | 0 | 0 | …… | 0 | Stop |

DO：

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | Data0 | Data1 | …… | DataN | Stop |

W: 85 Ah Am Al: 先设置基址地址

R: N (Boot返回pram数据，地址自加)



Table2 读操作各字段描述

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Default** |  |  |
| Add\_H | 程序开始地址的高字节 | 1 Byte |  |  |
| Add\_M | 程序开始地址的中间字节 | 1Byte |  |  |
| Add\_L | 程序开始地址的低地址 |  |  |  |
| Data | 程序代码 |  |  |  |
|  |  |  |  |  |

### 启动app

Rowboot接收到此命令时，remap，启动app程序

|  |  |  |
| --- | --- | --- |
| Start | 0x08 | Stop |

W: 08



### 复位

发送此命令可以让FT8836Reset，重新运行程序

第一步：

|  |  |  |
| --- | --- | --- |
| Start | 0x07 | Stop |

W: 07



### 查询状态

DI：

|  |  |  |  |
| --- | --- | --- | --- |
| Start | 0xD0 | 0x00 | Stop |

DO：

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | state |  |

W: D0 R 01 (Boot返回状态标志)



### 读Vendor ID

DI：

|  |  |  |  |
| --- | --- | --- | --- |
| Start | 0xA8 | 0x00 | Stop |

DO：

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Upgrade Version |  |
|  |  |  |  |

W: A8 R 01 (Boot返回状态 Vendor ID号)



# 基本时序图

可以通过两种方式启动Upgrade：上电或者Reset

### 上电时序

可以在上电时通过接收命令判断下一步进行的操作，这是需要一个严格的时序的，上电后接收命令的时序如下：



5ms <Tp< 25ms

### Reset 时序



5ms <Tp< 25ms

### I2C 时序图



# Upgrade使用方式

Host 升级pram的方法：

1. 发upgrade 命令，(同app约定好) delay 6~10ms
2. 写0x55
3. 同图表2 Host下载pram操作流程…

当过程出现错误时：

* pin rst/ porrst delay 6~10ms
* 发upgrade 命令，(同app约定好) delay 6~10ms
* 写 0x55
* …

# 启动时间

|  |  |
| --- | --- |
|  |  |
| 等待UPGRADE命令时间 | 6--20ms |
| 第一个命令必须为55解锁 | 只发55命令 |
| 不升级时正常启动时间 | Flash Device:   * 35ms – 270ms (3M) * 35ms – 135ms (6M) |