STM32 Bootloader设计

# FLAHS分配

## 1.1 STM32 Flash存储分配

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 单片机型号 | FLASH总大小 | BootLoader地址 | 参数分配表地址（可选） | APP地址` |
| STM32F103RCT6 | 0x08000000~0x08040000 | 0x08000000~0x0800FBFF | 0x0800FC00~0x0800FFFF | 0x08010000~0x0803FFFF |
| 256KB | 63KB | 1KB | 192KB |
| STM32F407ZET6 | 0x8000000~0x08080000 | 0x08000000~0x0800FBFF | 0x0800FC00~0x0800FFFF | 0x08010000~0x0807FFFF |
| 512KB | 63KB | 1KB | 448KB |
|  |  |  |  |  |
|  |  |  |  |  |

## 1.2 SPI Flash存储分配

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Flash型号 | Flash总大小 | 参数分配表地址 | 固件地址 | 字体&其他地址 |
| W25Q64 | 0x00000000~0x007FFFFF | 0x00000000~0x00000FFF | 0x00001000~0x000FFFFF | 0x00100000~0x001FFFFF |
| 8MB | 4KB | 1020KB | 1M |
| Flash型号 | 应用APP使用部分 |  |  |  |
| W25Q64 | 0x00100000~0x007FFFFF |  |  |  |
| 8MB |  |  |  |

# BootLoader参数分配表

## 2.1参数分配表格

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 项目 | 参数 | 长度 | 数据类型 | 备注 |
| Bootloader相关信息 | 是否已写入标识 | 15 | ASCII | 固定为：Copyright\_By\_Cc |
| BootLoader固件版本 | 7 | ASCII | 1.0.0.0 |
| 固件build时间 | 40 | ASIIC | Sep 8 2018 17:51:26 |
| 是否进行APP效验 | 1 | INT | 0:不效验，1：效验 |
| APP程序状态 | 1 | INT | 0:程序正常，1：文件效验不正确 |
| 是否支持SPIFLASH备份固件 | 1 | INT | 0:不支持 1：支持 |
| SPI FLASH备份固件起始地址 | 4 | INT | 例如:0X08000000 |
| SPI FLASH备份固件区域大小 | 4 | INT | 例如: 0X00100000 |
| 设备SN | 16 | ASIIC | 例如“MULTI01234567890” |
| 保留1 | 111 |  |  |
| 升级文件相关信息 | APP固件版本 | 7 | ASCII | 1.0.0.0 |
| 固件编译时间 | 8 | INT | unix时间戳 |
| 更新密码 | 8 | ASCII | 固定为：19920514 |
| 固件CRC32效验 | 4 | INT |  |
| app文件大小 | 4 | INT |  |
| app固件写入地址 | 4 | INT |  |
| Spi数据CRC32效验 | 4 |  |  |
| Spi文件大小 | 4 | INT |  |
| SPI写入地址 | 4 | INT |  |
| Eeprom数据CRC32效验 | 4 |  |  |
| Eeprom文件大小 | 4 | INT |  |
| Eeprom写入地址 | 4 | INT |  |
| 是否使能版本效验 | 1 | INT | 1：启用，启用后版本相同不更新  0：不启用 |
| 保留2 | 40 |  |  |
| 指定SD卡文件升级相关信息 | 升级标志位 | 1 | INT | 0：不升级 1：升级 |
| 升级文件名长度 | 1 | INT |  |
| 升级文件名 | 100 | ASCII |  |
| 升级文件路径长度 | 1 | INT |  |
| 升级文件路径（含文件名） | 255 | ASCII |  |
| 保留3 | 42 |  |  |
| SPI FLASH固件存储相关信息 | SPIFLASH升级标志位 | 1 | INT |  |
| 固件所在SPIFLASH地址 | 4 | INT | 该固件格式和SD卡升级固件格式一致 |
| 固件大小 | 4 | INT |  |
| 保留4 | 91 |  |  |
|  | 保留5 | 224 |  |  |
| 合计 |  |  |  |  |

# 三．串口升级

## 3.1串口升级通讯协议

通讯协议说明

采用大端模式：即高字节在前

### 3.1.1通讯请求0x03

1.请求通讯命令

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数据头2byte | 协议版本1byte | 通讯编号1byte | 功能码1byte | 数据长度2byte | 数据内容 | CRC16  2byte |
| 0x55,0x7a | 01 | 自增 | 0X03 | 0X08 | 19920514 |  |

2.请求通讯命令响应

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数据头2byte | 协议版本1byte | 通讯编号1byte | 功能码1byte | 数据长度2byte | 数据内容 | CRC16  2byte |
| 0x55,0x7a | 01 | 自增 | 0X03 | 0X08 | 19920514 |  |

### 3.1.2设置升级0x04

1.设置升级命令

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数据头2byte | 协议版本1byte | 通讯编号1byte | 功能码1byte | 数据长度2byte | 数据内容 | CRC16  2byte |
| 0x55,0x7a | 01 | 自增 | 0X04 | 0X0d | 见下表 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 文件大小4byte | 文件类型1byte | 文件效验CRC32 4byte | 固件烧录地址4byte |
|  | 见下表 |  |  |

|  |  |  |
| --- | --- | --- |
| 文件类型 | 解释 | 备注 |
| 1 | 写到MCU的APP程序 |  |
| 2 | 写到外部SPI FLASH数据 |  |
| 3 | 写到外部EEPROM的数据 |  |

2.设置升级命令响应

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数据头2byte | 协议版本1byte | 通讯编号1byte | 功能码1byte | 数据长度2byte | 数据内容 | CRC16  2byte |
| 0x55,0x7a | 01 | 自增 | 0X04 | 0X01 | 0x00 |  |

0:ok

1:不支持串口升级

### 3.1.3发送升级数据0x05

1.发送升级数据

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数据头2byte | 协议版本1byte | 通讯编号1byte | 功能码1byte | 数据长度2byte | 数据内容 | CRC16  2byte |
| 0x55,0x7a | 01 | 自增 | 0X05 | 2+1024 | 见下表 |  |

|  |  |
| --- | --- |
| 数据包计数2byte | 升级数据 |
| CNT | DATA |

2.升级数据响应

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数据头2byte | 协议版本1byte | 通讯编号1byte | 功能码1byte | 数据长度2byte | 数据内容 | CRC16  2byte |
| 0x55,0x7a | 01 | 自增 | 0X05 | 0X02 | 见下表 |  |

|  |  |
| --- | --- |
| 状态1byte | 数据包计数号2byte |
| 0：OK  1：请求重发  2:写入flash失败  3：固件成功接收 | CNT |

### 3.1.4发送功能命令0x06

1.发送升级数据

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数据头2byte | 协议版本1byte | 通讯编号1byte | 功能码1byte | 数据长度2byte | 数据内容 | CRC16  2byte |
| 0x55,0x7a | 01 | 自增 | 0X06 | 1 | 1：重启设备  2：进入U盘模式  3：跳转到APP |  |

2.升级数据响应

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数据头2byte | 协议版本1byte | 通讯编号1byte | 功能码1byte | 数据长度2byte | 数据内容 | CRC16  2byte |
| 0x55,0x7a | 01 | 自增 | 0X06 | 0X01 | 0：ok  其他：未正确执行 |  |

## 3.2串口数据加密

### 3.2.1加密算法

## 3.3串口升级流程

# 四．SD卡升级

## 4.1文件格式

目前仅支持bin文件升级

Bin文件格式如下，文件前100个字节和参数表中间区段一一对应，后面为编译bin文件加密后数据。

|  |  |  |  |
| --- | --- | --- | --- |
| 文件结构 | 字节 | 格式 | 备注 |
| APP固件版本 | 7 | ASCII | 1.0.0.0 |
| 固件编译时间 | 8 | INT | unix时间戳 |
| 更新密码 | 8 | ASCII | 固定为：19920514 |
| 固件CRC32效验 | 4 | INT |  |
| 文件大小 | 4 | INT |  |
| 固件写入地址 | 4 | INT |  |
| 保留2 | 65 |  |  |
| 文件数据 |  |  |  |

## 4.2文件升级

## 4.3文件加密算法

# 五．USB升级

# 六．SPI Flash升级

# 七．附录

## 7.1计算CRC16源码

CRC16计算方式：

#define POLY 0x1021

uint16\_t crc16(unsigned char \*addr, int num, uint16\_t crc)

{

int i;

for (; num > 0; num--) /\* Step through bytes in memory \*/

{

crc = crc ^ (\*addr++ << 8); /\* Fetch byte from memory, XOR into CRC top byte\*/

for (i = 0; i < 8; i++) /\* Prepare to rotate 8 bits \*/

{

if (crc & 0x8000) /\* b15 is set... \*/

crc = (crc << 1) ^ POLY; /\* rotate and XOR with polynomic \*/

else /\* b15 is clear... \*/

crc <<= 1; /\* just rotate \*/

} /\* Loop for 8 bits \*/

crc &= 0xFFFF; /\* Ensure CRC remains 16-bit value \*/

} /\* Loop until num=0 \*/

return(crc); /\* Return updated CRC \*/

}

## 7.2计算CRC32计算方式

unsigned int CRC32(unsigned char \*puchMsg, unsigned int usDataLen)

{

int i;

unsigned int wCRCin = 0xFFFFFFFF;

unsigned int wCPoly = 0x04C11DB7;

unsigned int wChar = 0;

while (usDataLen--)

{

wChar = \*(puchMsg++);

InvertUint8((unsigned char \*)&wChar,(unsigned char \*)&wChar);

wCRCin ^= (wChar << 24);

for(i = 0;i < 8;i++)

{

if(wCRCin & 0x80000000)

wCRCin = (wCRCin << 1) ^ wCPoly;

else

wCRCin = wCRCin << 1;

}

}

InvertUint32(&wCRCin,&wCRCin);

return (wCRCin ^ 0xFFFFFFFF) ;

}