**印联盒V2.0**

**Boot功能说明**

|  |  |  |  |
| --- | --- | --- | --- |
| **版本历史** | | | |
| 版本号 | 描述 | 作者 | 日期 |
| 0.1 | 初稿 | 蒋攀 | 2020/06/17 |

目录

[1 目的及适用范围 4](#_Toc43301139)

[1.1 目的 4](#_Toc43301140)

[1.2 适用范围 4](#_Toc43301141)

[2 文件结构 4](#_Toc43301142)

[2.1 STM32CubeMX驱动层 4](#_Toc43301143)

[2.2 应用层 4](#_Toc43301144)

[3 功能说明 4](#_Toc43301145)

[3.1 串口数据帧说明 5](#_Toc43301146)

[3.2 程序分区说明 5](#_Toc43301147)

[3.3 配置说明 5](#_Toc43301148)

[3.3 功能流程 5](#_Toc43301149)

[3.4 IAP流程 6](#_Toc43301150)

[4 备注 7](#_Toc43301151)

[5 小结 7](#_Toc43301152)

# 目的及适用范围

## 1.1 目的

为明确印联盒V2.0Boot的一些功能，特编写此文档。

## 1.2 适用范围

本文档仅适用于印联盒V2.0Boot的功能说明。

# 文件结构

文件结构分为STM32CubeMX驱动层与应用层代码：

## 2.1 STM32CubeMX驱动层

使用STMEWCubeMX配置好外设之后，自动生成代码，不再赘述；

## 2.2 应用层

分为CRC、FLASH、IAP、LED、MAIN、SYSTEM\_CONFIG、USART等模块；

⚫CRC：用于CRC校验：

⚫FLASH：提供FLASH读写功能，由IAP调用；

⚫IAP：用于将DataCollectSoftware写入到FLASH中；

⚫LED：提供LED初始化函数，由SYSTEM\_CONFIG调用；

⚫SYSTEM\_CONFIG：定义配置、串口数据结构体，初始化系统设置；

⚫USART：提供USART的初始化函数由SYSTEM\_CONFIG调用；用于串口数据的

接收与解析、发送数据的打包与发送；

# 功能说明

Boot是一个使用上位机，且通过RS232串口升级DataCollectSoftware的程序。

Boot定义了与DataCollectSoftware相同的串口数据帧，但数据缓存区长度要大得多（如果不这样做，那么升级过程就会很慢），并使用DataCollectSoftware的配置串口来进行升级。

## 3.1 串口数据帧说明

BOOT下串口数据帧（以下简称数据帧）包含帧头、设备源ID、设备目标ID、命令字、数据长度、数据缓冲区（包含需要传输的数据CRC校验码以及数据）、帧尾。

## 3.2 程序分区说明

BootLoad设定为前64KB，Application设定为832KB，配置区域设定为128KB，一共1M的空间。

## 3.3 配置说明

Boot使用STM32CubeMX作为工具、选择STM32F407ZGT6作为主芯片，配置主芯片的调试接口、时钟树、USART1、UART4、LED1、LED3等外设，USART1用于接收/发送串口数据帧，UART4用于打印调试信息，LED1（PC6）、LED2（PC8）用于指示Boot状态。

## 3.3 功能流程

① 从0x080E0000地址读取配置信息，如果配置信息中u16upgradeflag标志位未被置位则跳转并执行DataCollectSoftware（跳转之前需要关闭中断），被置位则进行DataCollectSoftware的IAP升级；

② 开启中断；

③ STM32F407的外设配置：

1. 调试接口；
2. 时钟树

⚫使用HSE时钟，晶振为8MHz；

⚫系统时钟为168MHz；

1. USART1

⚫波特率：115200bps；

⚫数据位：8bits；

⚫校验位：None；

⚫停止位：1bit；

⚫硬件流控：None；

⚫开启TX的DMA通道；

⚫开启RX的DMA通道；

⚫开启USART1中断

1. UART4

⚫波特率：115200bps；

⚫数据位：8bits；

⚫校验位：None；

⚫停止位：1bit；

⚫硬件流控：None；

⚫开启TX的DMA通道；

⚫开启RX的DMA通道；

1. LED1（PC6）

⚫output模式

1. LED2（PC8）

⚫ output模式

④ 通过UART4打印Boot的版本信息；

⑤ 进入DataCollectSoftware的IAP流程；

## 3.4 IAP流程

① 上位机与Boot的握手机制

1. Boot进入IAP流程之后等待上位机发送握手信息；
2. 上位机发送一帧带有数据帧结构的“BOOT？”信息；
3. Boot接收到（二）中的信息后，回应一帧带有数据帧结构的“BOOT：OK”信息；
4. 上位机接收到（三）中的信息后，代表握手成功；
5. 进行②步的操作；

② 接收文件总帧数

1. 上位机发送一帧带有文件总帧数的数据帧；
2. Boot接收到（一）中的信息后，进行CRC校；校验成功，将文件总帧数保存下来，将当前帧数置零，并回应一帧带有数据帧结构的“OK\n”信息；校验失败回应一帧带有数据帧结构的“Error\n”信息；
3. 上位机收到（二）中的一帧带有数据帧结构的“OK\n”信息，代表文件总帧数发送成功，进行③步操作；上位机收到（二）中的一帧带有数据帧结构的“Error\n”信息，则重新发送一帧带有文件总帧数的数据帧；

③ 发送bin文件信息

1. 上位机发送一帧带有bin文件内容的数据帧；
2. Boot接收到（一）中的信息后，进行CRC校验；校验成功，比较Boot的当前帧是否与数据帧的当前帧（第2、3字节）是否相等，如相等将bin文件信息内容写入到FLASH中，并回应一帧带有数据帧结构的“OK\n”信息；校验失败回应一帧带有数据帧结构的“Error\n”信息；
3. 上位机收到（二）中的一帧带有数据帧结构的“OK\n”信息，当前帧的文件内容发送成功，当前帧数自增一，如果当前帧数等于总帧数代表文件已经全部发送完成了，进行④步操作，读取bin文件的下一帧内容，进行（一）步操作；上位机收到（二）中的一帧带有数据帧结构的“Error\n”信息，则重新发送一帧当前帧的带有bin文件内容的数据帧；

④ bin文件接收完成

（一）boot下判断当前帧数与文件总帧数是否相等；相等则说明文件已接收完毕，可以进行DataCollectSoftware的跳转了，将u16upgradeflag清位，写入FLASH中，关闭中断，软件复位；

# 备注

BootLoader协议需要注意以下几点：

1. 文件总帧数，包含在其数据帧的第2、3字节；
2. 当前帧数，包含在其数据帧的第2、3字节；
3. CRC校验值包含在数据帧的第0、1字节；
4. 每帧的数据帧帧头固定为0xFEEF、升级命令字固定为0x0004、帧尾固定为0xEAAE，源ID与目标ID号，暂时未使用。

# 小结

1. 目前BootLoader未做整个文件的CRC校验；
2. 文件总的字节数没有发送给MCU；
3. BootLoader未做超时处理功能，就是长时间没有升级成功，一直卡在BootLoader程序，需要处理此类问题；