**基于STM32H5的反无产品**

**固件升级方案**

|  |  |  |  |
| --- | --- | --- | --- |
| 拟制 | 陈安东 | 日期 | 2023年08月06日 |
| 修正 |  | 日期 |  |
| 审批 |  | 日期 |  |

**变更记录**

|  |  |  |  |
| --- | --- | --- | --- |
| **版本** | **说明** | **拟制人** | **日期** |
| V1.0 | 初稿 | 陈安东 | 2023-08-06 |
|  |  |  |  |
|  |  |  |  |

# 1 概述

本文基于对STM32H5芯片平台设备启动及固件升级流程的研究，介绍了基于STM32H5芯片平台的反无产品固件升级方案，用于设计评审及开发参考。

本文适用于系统工程师、软件架构师、软件开发工程师、软件测试工程师等领域人员。

# 2 背景知识

本节对STM32H5芯片平台的启动流程及升级原理进行简要介绍。未尽详细信息可以参考附录A 参考资料中列出的文档中的相关章节。

## 2.1 STM32启动流程

XIP设备，eXecute In Place，即芯片内执行，指应用程序可以直接在 flash 闪存内运行，不必再把代码读到系统 RAM中。在我们的印象里，应用程序必须要从硬盘中加载到内存当中才可以被运行，但实际上应用程序是可以直接在flash 闪存运行的，也就是说，cpu 可以直接从 flash 中取出指令，当前系统就是直接运行于片内flash上。

# 3 方案设计

## 3.1 总体设计

基于背景知识，本节对基于STM32系列芯片平台的反无产品固件升级方案设计进行说明。反制枪OTA升级功能实现需要上位机和软件协同开发，上位机实现固件文件下发和界面显示功能，嵌入式实现具体升级功能，升级流程需要在十分钟能完成。

### 3.1.1 约束条件

基于STM32H5系列芯片平台特点，有以下约束条件：

1. 所有固件的大小之和不超过1MB(flash共2M)。

### 3.1.2 Flash分区

Flash分区如表 1所示。

表 1 Flash分区表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 分区名称 | 分区描述 | 起始地址 | 结束地址 | 大小 |
| 系统参数 | 系统配置参数存储区域 | 0x081C\_E000 | 0x081F\_FFFF | 100KB |
| 保留区域 | 暂不使用 | 0x080A\_F000 | 0x081C\_DFFF | 1248KB |
| 业务镜像B  IMAGE\_B | 实现业务功能B镜像 | 0x0807\_D000 | 0x080A\_EFFF | 200KB |
| 保留区域 | 暂不使用 | 0x0806\_4000 | 0x0807\_CFFF | 100KB |
| 业务镜像A  IMAGE\_A | 实现业务功能的A镜像 | 0x0803\_2000 | 0x0806\_3FFF | 200KB |
| 启动镜像bootloader | 通过启动标志跳转到指定业务镜像 | 0x0800\_0000 | 0x0803\_1FFF | 200KB |

Bootloader存放在Flash的起始地址，BootROM默认加载的位置，占用200K地址空间。Bootloader在每次启动后都会读取boot\_flag标志，通过标志跳转到指定镜像，标志缺失时默认跳转到镜像A。

业务镜像A存放在Flash起始地址为0x0803\_2000处，当前分配大小为200K，生成的APP\_A.bin文件大小不能超过该值。业务镜像B存放在Flash起始地址为0x0807\_D000处，当前分配大小为200K，生成的APP\_B.bin文件大小不能超过该值。

系统参数存放在flash最后，当前分配的空间为100K，主要用于存储系统中需要持久化保存的参数。

### 3.1.3 镜像内容

Bootloader只做跳转，镜像A和镜像B功能相同，只是连接文件指定地址不同，升级流程主要实现升级指令和数据接收与解析，执行固件刷新动作，业务镜像接收升级指令升级对区镜像，升级成功后更新启动标志为对区标志，之后由对区启动。

### 3.1.4 数据传输

数据传输与具体的产品形态和需求相关，一般会使用串口或网络传输。具体形式和协议在对应的产品固件升级协议说明文档中进行描述，此处不再赘述。

一般地，通过通讯接口将待升级的固件文件下载到RAM中，校验后写入业务镜像所在的区域，指定跳转地址跳转后即可运行更新后的固件。

## 3.2 工作流程

### 3.2.1 业务镜像

业务镜像中与固件升级相关的流程如图 7所示。

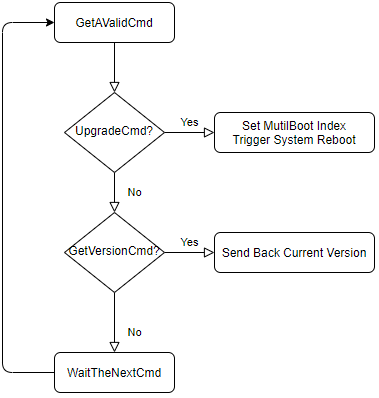


图 7 业务镜像中升级任务的工作流程

业务镜像接收与固件升级相关的指令，主要需要支持启动升级功能和查询当前固件版本号的功能。当接收到启动升级的指令之后设置MuitiBoot寄存器，指示到升级镜像所在的位置，然后触发系统复位，BootROM就会加载并执行升级镜像，从而执行与业务镜像更新相关的工作。

### 3.2.2 升级镜像

升级镜像的核心功能就是升级业务镜像，其工作流程如图 8所示。

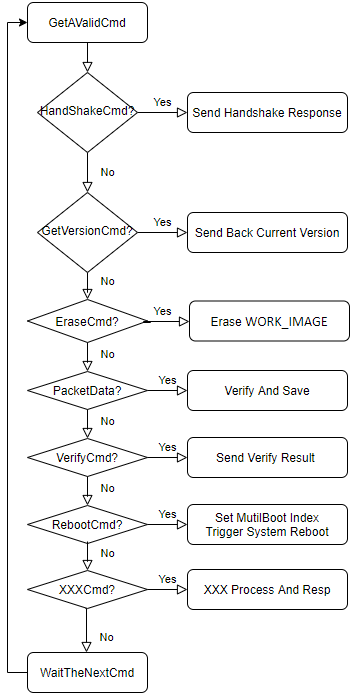


图 8 升级镜像的工作流程

升级镜像接收与固件升级相关的指令，主要需要支持以下功能：

1. 消息握手/版本号查询，用于上位机检查升级镜像是否正常运行。
2. Flash擦除，用于擦除待升级固件所需的Flash空间。
3. 固件传输，用于接收固件文件。
4. 固件校验，用于校验最终的固件文件是否成功烧写完成。
5. 固件运行，用于启动升级后的固件文件。

## 3.3 备选方案

# 附录A 参考资料

1、<https://shatang.github.io/2020/08/12/IAP%E5%8D%87%E7%BA%A7-Bootloader%E5%88%B6%E4%BD%9C/>

 2、<https://www.21ic.com/article/900001.html>

 3、<https://blog.51cto.com/u_11947739/6540532>