

## 前言

USB MSD IAP 是一个固件升级工具，它不依赖于其它上位机工具，可直接接入到 PC 或手机上进行设备的固件升级。本应用笔记主要重点介绍 USB MSD IAP 的原理及过程实现。

注：本应用笔记对应的代码是基于雅特力提供的V2.x.x 板级支持包（BSP）而开发，对于其他版本BSP，需要注意使用上的区别。

支持型号列表：

| 支持型号 | 具备 USB 的型号 |
|------|------------|
|------|------------|

## 目录

|          |                                       |           |
|----------|---------------------------------------|-----------|
| <b>1</b> | <b>IAP 在线升级原理概述 .....</b>             | <b>5</b>  |
| <b>2</b> | <b>AT32 USB MSD IAP 简介 .....</b>      | <b>7</b>  |
| 2.1      | AT32 USB MSD IAP 设计功能特点 .....         | 7         |
| 2.2      | 程序设计 .....                            | 7         |
| 2.2.1    | 地址空间 .....                            | 7         |
| 2.2.2    | 升级状态 .....                            | 7         |
| 2.2.3    | 升级 BIN 档文件名格式 .....                   | 7         |
| 2.2.4    | 升级 HEX 档文件格式(后缀 .HEX) .....           | 8         |
| 2.2.5    | 升级加密档文件格式（后缀 .SEC） .....              | 8         |
| 2.2.6    | 使用 Option Byte 标志判断当前是否升级 .....       | 8         |
| 2.2.7    | 跳转到 APP code 执行 .....                 | 8         |
| <b>3</b> | <b>使用 AT32 USB MSD IAP 进行升级 .....</b> | <b>9</b>  |
| 3.1      | 硬件资源 .....                            | 9         |
| 3.2      | 软件资源 .....                            | 9         |
| 3.3      | 使用流程 .....                            | 9         |
| <b>4</b> | <b>版本历史 .....</b>                     | <b>12</b> |

## 表目录

|                   |    |
|-------------------|----|
| 表 1. 地址空间分布 ..... | 7  |
| 表 2. 文档版本历史 ..... | 12 |

## 图目录

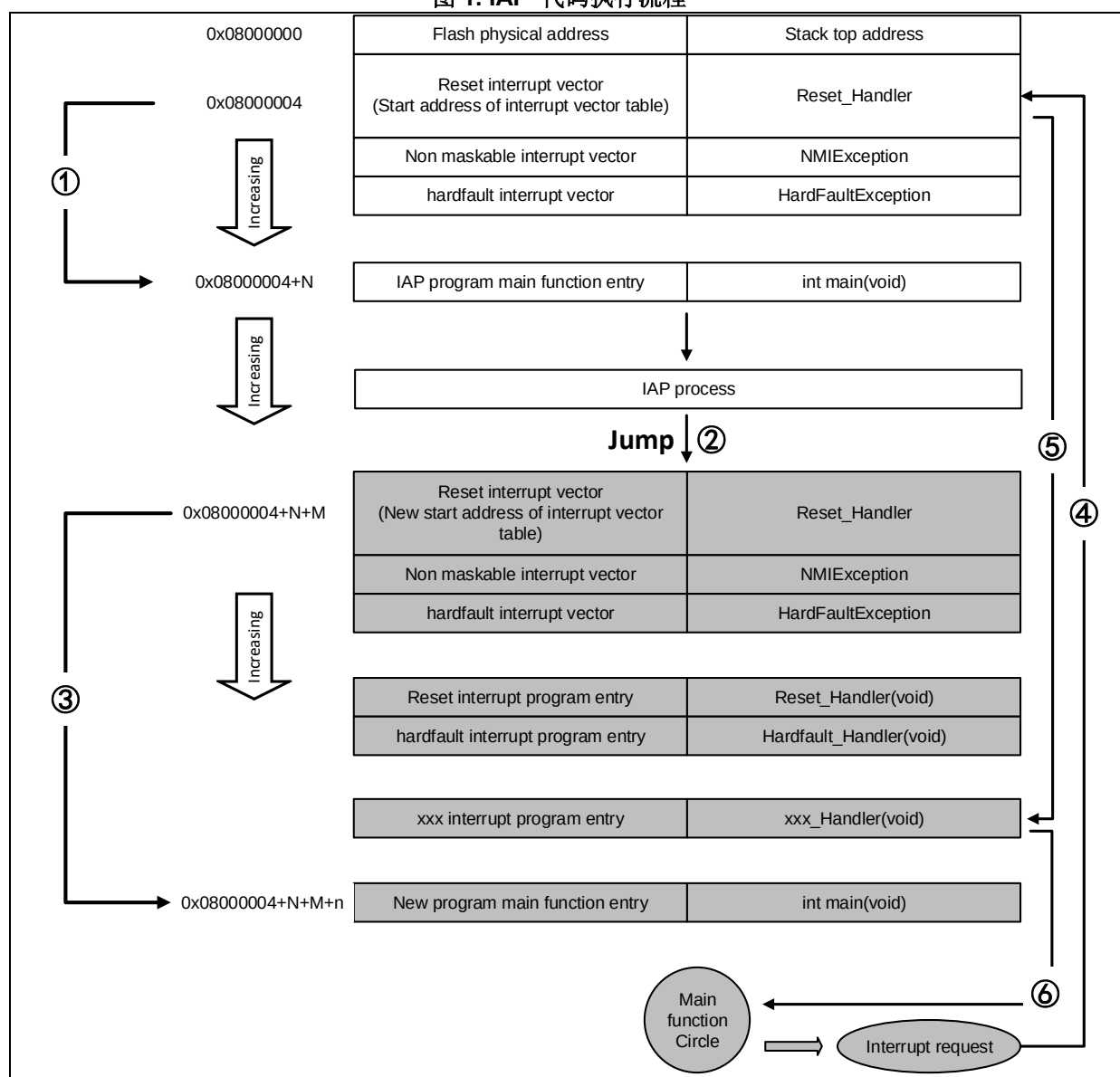
|                            |    |
|----------------------------|----|
| 图 1. IAP 代码执行流程 .....      | 5  |
| 图 2. AT-START-F403A .....  | 9  |
| 图 3. 进入 IAP Mode 按键.....   | 10 |
| 图 4. USB 设备与 PC 连接 .....   | 10 |
| 图 5. PC 识别到 AT32 IAP ..... | 10 |
| 图 6. 拷贝 Bin 文件 .....       | 10 |
| 图 7. 设备升级完成提示.....         | 11 |

# 1 IAP 在线升级原理概述

IAP (In Application Programming) 即在应用编程，IAP 是用户自己的程序在运行过程中对 User Flash 的部分区域进行烧写，目的是为了在产品发布后可以方便地通过预留的通信口对产品中的固件程序进行更新升级。通常实现 IAP 功能时，即用户程序运行中作自身的更新操作，需要在设计固件程序时编写两个项目代码，第一个项目程序不执行正常的功能操作，而只是通过某种通信方式(如 USB、USART)接收程序或数据，执行对第二部分代码的更新；第二个项目代码才是真正的功能代码。这两部分项目代码都同时烧录在 User Flash 中，当芯片上电后，首先是第一个项目代码开始运行，它作如下操作：

- 1) 检查是否需要第二部分代码进行更新
- 2) 如果不需要更新则转到4)
- 3) 执行更新操作
- 4) 跳转到第二部分代码执行

图 1. IAP 代码执行流程



在图1所示流程中，AT32 复位后，还是从 0X08000004 地址取出复位中断向量的地址，并跳转到复位中断服务程序，在运行完复位中断服务程序之后跳转到 IAP 的 main 函数，如图标号①所示，在执行完 IAP 以后（即将新的 APP 代码写入 AT32 的 FLASH，灰底部分。新程序的复位中断向量起始地址为 0X08000004+N+M），跳转至新写入程序的复位向量表，取出新程序的复位中断向量的地址。

址，并跳转执行新程序的复位中断服务程序，随后跳转至新程序的 `main` 函数，如图标号②和③所示，同样 `main` 函数为一个死循环，并且注意到此时 AT32 的 FLASH，在不同位置上，共有两个中断向量表。

在 `main` 函数执行过程中，如果 CPU 得到一个中断请求，PC 指针仍强制跳转到地址 `0X08000004` 中断向量表处，而不是新程序的中断向量表，如图标号④所示；程序再根据我们设置的中断向量表偏移量，跳转到对应中断源新的中断服务程序中，如图标号⑤所示；在执行完中断服务程序后，程序返回 `main` 函数继续运行，如图标号⑥所示。

通过以上两个过程的分析，我们知道 IAP 程序必须满足两个要求：

- 1) 新程序必须在 IAP 程序之后的某个偏移量为 `x` 的地址开始
- 2) 必须将新程序的中断向量表相应的移动，移动的偏移量为 `x`

## 2 AT32 USB MSD IAP 简介

USB MSD IAP 是一个固件升级工具，它不依赖于其它上位机工具，可直接接入到 PC 或手机上进行设备的固件升级。

实现原理：将 FLASH 虚拟成一个存储设备让 PC 访问。

极其简单的操作步骤：

- 1) 将 USB 接口接入 PC
- 2) PC 识别到盘符 “AT32 IAP “
- 3) 将需要升级的固件拷贝到 AT32 IAP 盘符里面
- 4) 升级完成

### 2.1 AT32 USB MSD IAP 设计功能特点

- IAP 目前保留使用 20K byte 空间，APP 的起始地址需在 20K 以后
- 使用 USB 大容量存储设备进行虚拟设备
- 升级后自动 reset USB 设备并返回升级状态
- 支持下载后自动读回进行 CRC 校验，保证固件正确性
- 支持设置下载地址（需按照 page 2K 对齐,并要大于 IAP 保留地址）
- 支持各种系统 windows, Android 等
- 支持升级完成之后跳转到 APP 运行
- 支持 BIN 文件升级
- 支持 HEX 文件升级（后续版本添加）
- 支持加密文件升级（后续版本添加）

## 2.2 程序设计

### 2.2.1 地址空间

表 1. 地址空间分布

| ITEM                  | Address                |
|-----------------------|------------------------|
| USB MSD IAP           | 0x08000000 ~0x08005000 |
| APP available address | 0x08005000~.....       |

### 2.2.2 升级状态

当连接 Host 之后，在盘符里面会对应有当前状态的 TXT 文档，根据文档名称不同来确定当前的状态。

准备升级状态（Ready.TXT）

升级成功（Success.TXT）

升级失败（Failed.TXT）

未知文件或错误（Unknown.TXT）

升级文件大于 FLASH 大小（Large.TXT）

注意：设备必须在 Ready.TXT 状态下才能进行升级，否则不会升级。

### 2.2.3 升级 BIN 档文件名格式

1. 需指定下载地址（格式 1）

文件名格式：(1Byte)A+(6Byte)offset+.BIN

如：要下载一个 BIN 文件到 0x08005000 为起始的地址空间去

文件名为：A005000.BIN

注意 6Byte 的 offset 地址需要保证在 APP 可使用的范围之内，否则将使用 IAP 内部默认的 APP 起始地址进行升级

当不满足格式 1 时，IAP 将使用内部默认 APP 的起始地址进行升级：

如：ABCDEFGH.BIN，A11111.BIN，jkakkkddkfj.BIN

## 2.2.4 升级 HEX 档文件格式(后缀 .HEX)

后续版本添加

## 2.2.5 升级加密档文件格式 （后缀 .SEC）

后续版本添加

## 2.2.6 使用 Option Byte 标志判断当前是否升级

IAP 中使用 Option Byte 中 HID[0] 记录升级是否成功，当进入 IAP 时，设置 HID[0]=1,当升级完成时设置 HID[0]=0;当设备启动时，会自动判断 HID[0] 是否已升级固件，如果是，就跳转到 APP 地址执行，否则就继续运行 IAP。

## 2.2.7 跳转到 APP code 执行

当成功下载固件到 FLASH 之后，如果需要跳转到用户 code 执行，有两种方法：

- 1) Reset
- 2) 向识别到的 AT IAP 虚拟磁盘上拷贝一个 JUMP.TXT 的文件，注意这个文件的大小不能为 0

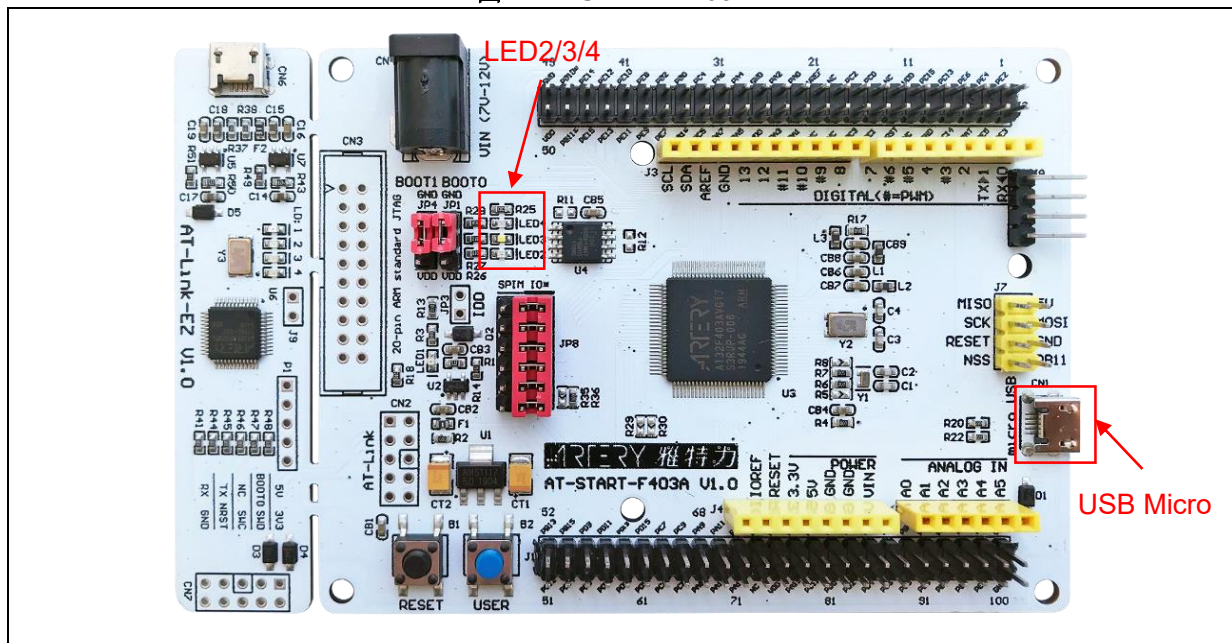


## 3 使用 AT32 USB MSD IAP 进行升级

### 3.1 硬件资源

- 1) 指示灯 LED2/LED3/LED4
- 2) USB(PA11/PA12)
- 3) AT-START-F403A V1.0 实验板

图 2. AT-START-F403A



注: 该 IAP demo 是基于 AT32F403A 的硬件条件, 若使用者需要在 AT32 其他型号上使用, 请修改相应配置即可。

### 3.2 软件资源

- 1) SourceCode
  - AN0012\_SourceCode\_V2.0.0\utilities\AN0012\_demo, IAP 源程序
  - AN0012\_SourceCode\_V2.0.0\libraries, AT32 外设库
  - AN0012\_SourceCode\_V2.0.0\middlewares, 其他资源
- 2) Doc
  - 《AN0012\_USB\_MSD\_IAP\_V2.x.x.docx》

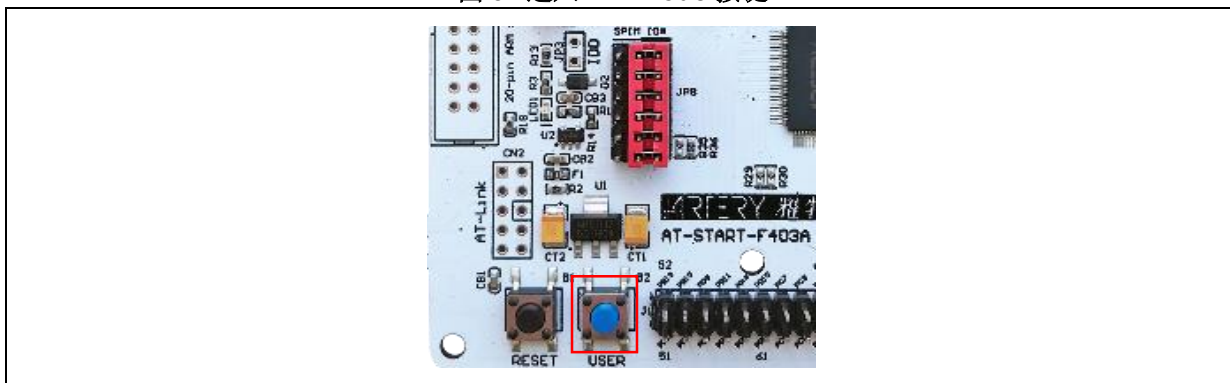
Note: 所有 project 都是基于 keil 5 而建立, 若用户需要在其他编译环境上使用, 请参考 AT32F403A\_407\_Firmware\_Library\project\at\_start\_f403a\templates 中各种编译环境 (例如 IAR6/7/8, keil 4/5) 进行简单修改即可。

### 3.3 使用流程

#### 1) 进入 IAP Mode 方式

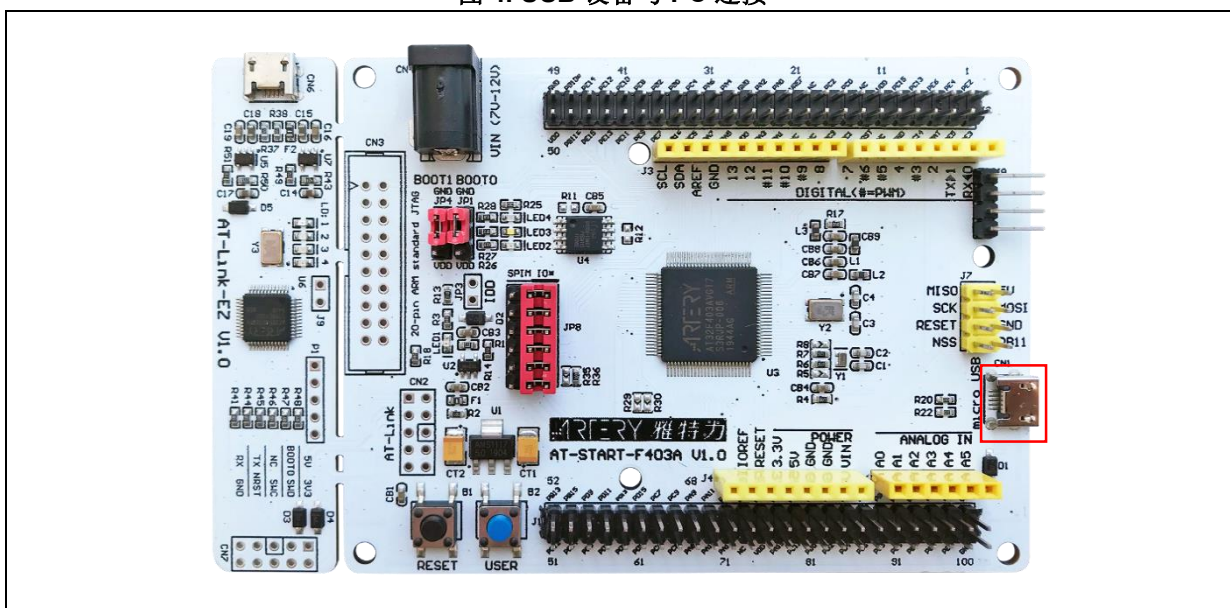
如果已经升级过固件, 一直按住 User 按键, 再按 Reset 键, 进入 IAP 模式(LED4 闪烁), 未升级时直接 reset 就会进入 IAP 模式

图 3. 进入 IAP Mode 按键



## 2) 连接 USB 设备到 PC

图 4. USB 设备与 PC 连接



## 3) PC 识别到 AT32 IAP 可看到 Ready.TXT

图 5. PC 识别到 AT32 IAP

| 名称    | 修改日期           | 类型   | 大小   |
|-------|----------------|------|------|
| Ready | 2008/4/18 8:20 | 文本文档 | 0 KB |

## 4) 拷贝 BIN 档到磁盘目录下

图 6. 拷贝 Bin 文件

| 名称          | 修改日期            | 类型     | 大小   |
|-------------|-----------------|--------|------|
| Ready       | 2008/4/18 8:20  | 文本文档   | 0 KB |
| A005000.bin | 2022/1/19 17:35 | BIN 文件 | 5 KB |

## 5) 设备升级完成之后将 Reset USB, 此时重新打开磁盘, 会有 Success.TXT

图 7. 设备升级完成提示

| 名称      | 修改日期           | 类型   | 大小   |
|---------|----------------|------|------|
| Success | 2008/4/18 8:20 | 文本文档 | 0 KB |

#### 6) 升级结束

此时 Reset 或向识别到的 AT IAP 虚拟磁盘上拷贝一个大小不为 0 的 JUMP.TXT 文件，即可跳转到用户代码。

如果升级一次之后，不管是成功还是失败，如果需要再次升级，需要 reset 整个设备。设备进入 Ready.TXT 状态。

## 4 版本历史

表 2. 文档版本历史

| 日期        | 版本    | 变更   |
|-----------|-------|------|
| 2022.1.25 | 2.0.0 | 最初版本 |

**重要通知 - 请仔细阅读**

买方自行负责对本文所述雅特力产品和服务的选择和使用，雅特力概不承担与选择或使用本文所述雅特力产品和服务相关的任何责任。

无论之前是否有过任何形式的表示，本文档不以任何方式对任何知识产权进行任何明示或默示的授权或许可。如果本文档任何部分涉及任何第三方产品或服务，不应被视为雅特力授权使用此类第三方产品或服务，或许可其中的任何知识产权，或者被视为涉及以任何方式使用任何此类第三方产品或服务或其中任何知识产权的保证。

除非在雅特力的销售条款中另有说明，否则，雅特力对雅特力产品的使用和/或销售不做任何明示或默示的保证，包括但不限于有关适销性、适合特定用途（及其依据任何司法管辖区的法律的对应情况），或侵犯任何专利、版权或其他知识产权的默示保证。

雅特力产品并非设计或专门用于下列用途的产品：（A）对安全性有特别要求的应用，例如：生命支持、主动植入设备或对产品功能安全有要求的系统；（B）航空应用；（C）航天应用或航天环境；（D）武器，且/或（E）其他可能导致人身伤害、死亡及财产损害的应用。如果采购商擅自将其用于前述应用，即使采购商向雅特力发出了书面通知，风险及法律责任仍将由采购商单独承担，且采购商应独立负责在前述应用中满足所有法律和法规要求。

经销的雅特力产品如有不同于本文档中提出的声明和/或技术特点的规定，将立即导致雅特力针对本文所述雅特力产品或服务授予的任何保证失效，并且不应以任何形式造成或扩大雅特力的任何责任。

© 2022 雅特力科技 保留所有权利