AN0001
应用笔记

AT32 IAP using the USART

前言

对于AT32 MCU，除了出厂固化的bootloader，用户也可以在应用程序中编写自己的bootloader程序，用于最终产品的固件更新。此功能称为应用程序内编程（IAP）。

本应用笔记的目的是提供在AT32微控制器上创建IAP应用程序的一般准则。

AT32微控制器可以运行用户特定的固件来对微控制器中嵌入的闪存执行IAP。此功能可以使用产品可用和支持的任何通信接口。使用自定义协议协议的USART是本应用笔记中的示例。

IAP_Programmer.exe上位机软件和嵌入式IAP USART示例的源代码位于BSP固件库的utilities文件夹内。

支持型号列表：

支持型号	AT32 全系列

目录

表目录

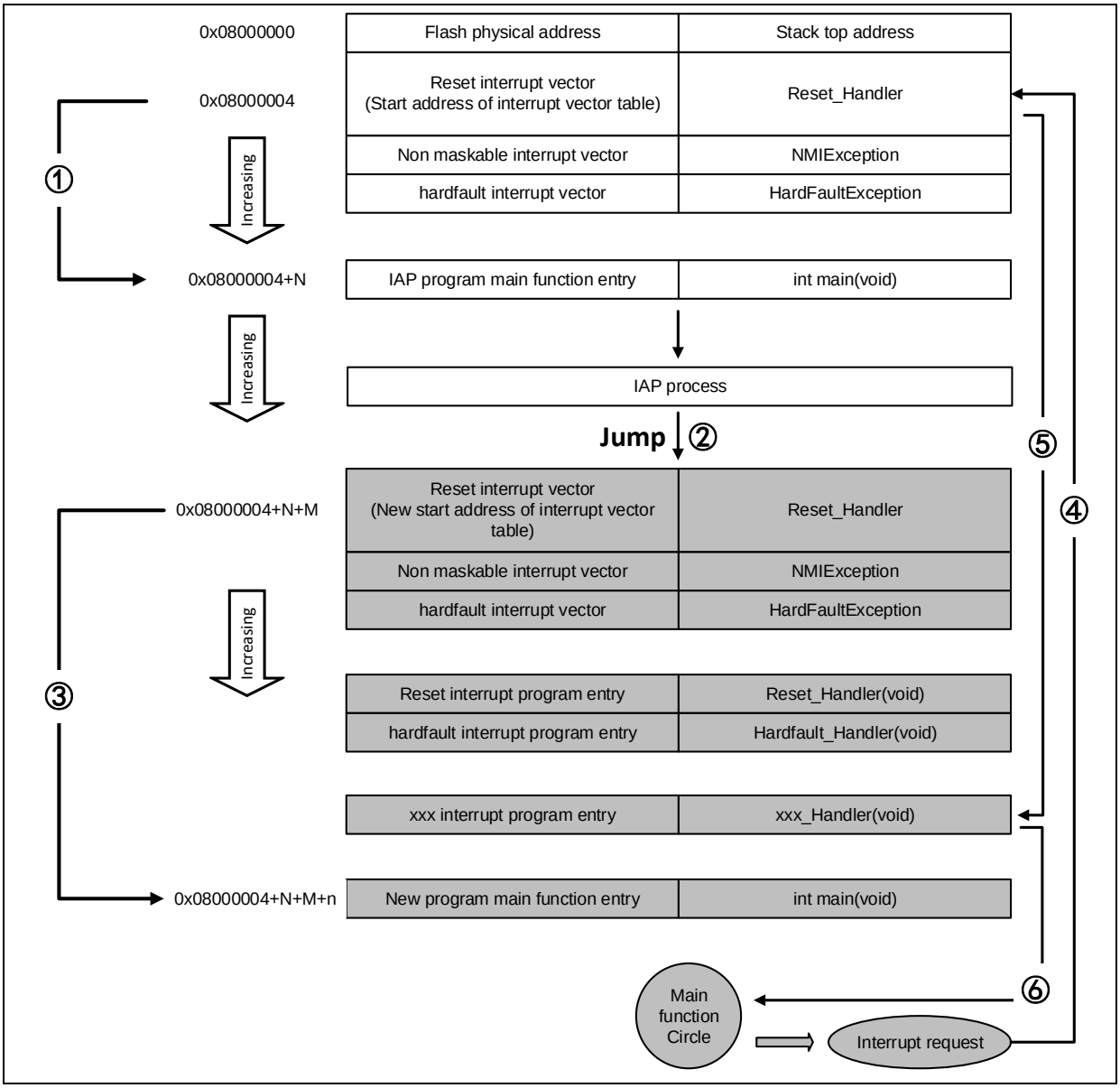
图目录

1 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) 检查是否需要对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



在上图所示流程中，MCU 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，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

1.1 AT32 USART IAP 快速使用方法

1.1.1 硬件资源

文档中是用 AT-START-AT32F403A 实验板的硬件条件为例，IAP demo 源代码还包括 AT32 其他型号，用户只需编译对应型号工程烧录于 AT-START 实验板运行即可。

1) 指示灯LED2/LED3/LED4
2) USART1(PA9/PA10)
3) AT-START实验板

1.1.2 软件资源

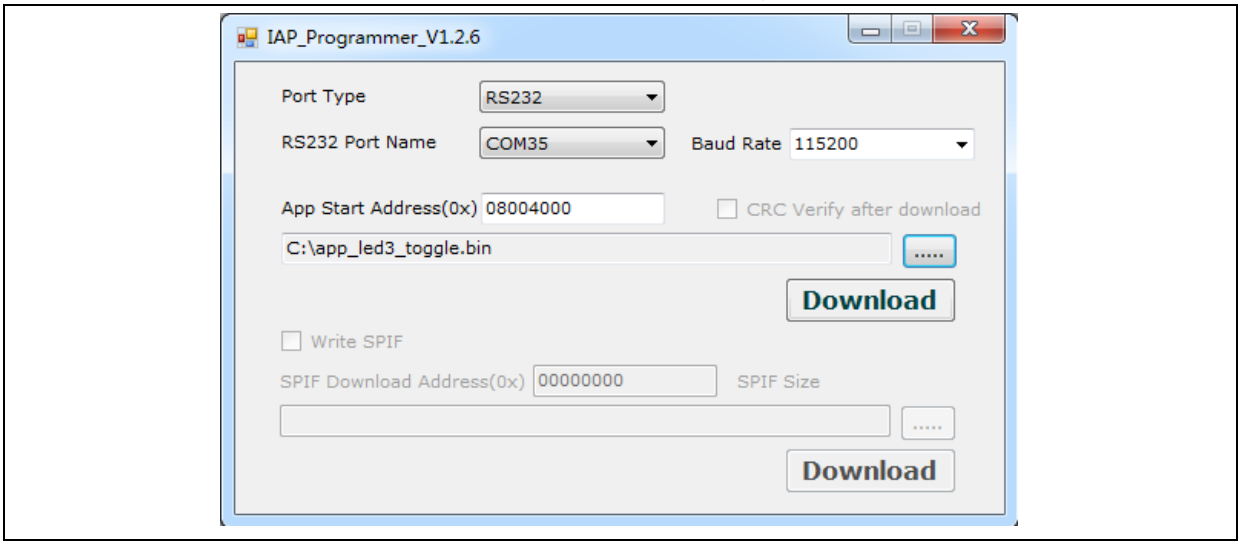
1) tool_release
   - IAP_Programmer.exe，PC机tool，用于演示IAP升级流程
2) source_code
   - bootloader，bootloader源程序，运行LED2闪烁
   - app_led3_toggle，app1源程序，运行LED3闪烁
   - app_led4_toggle，app2源程序，运行LED4闪烁

注：源码位于AT32F403A_407_Firmware_Library_V2.x.x\utilities\at32f403a_407_usart_iap_demo\source_code，示例工程基于keil v5和IAR8.2建立，若用户需要在其他编译环境上使用，请参考AT32F403A_407_Firmware_Library_V2.x.x\project\at_start_f403a\templates中各种编译环境（例如IAR6/7/8,keil 4/5, eclipse_gcc）进行对应修改即可。

1.2 IAP demo 使用

1) 打开bootloader工程源程序，选择对应MCU型号的target编译后下载到实验板
2) 打开IAP_Programmer.exe
3) 选择正确的串口、APP下载地址和bin文档，点击Download下载，如下图
4) 观察LED2/3/4闪烁，LED2闪烁-bootloader工作，LED3闪烁-app1工作，LED4闪烁-app2工作

图 2. IAP demo 上位机



2 AT32 USART IAP 程序设置

2.1 地址分布

表 1. 地址分布

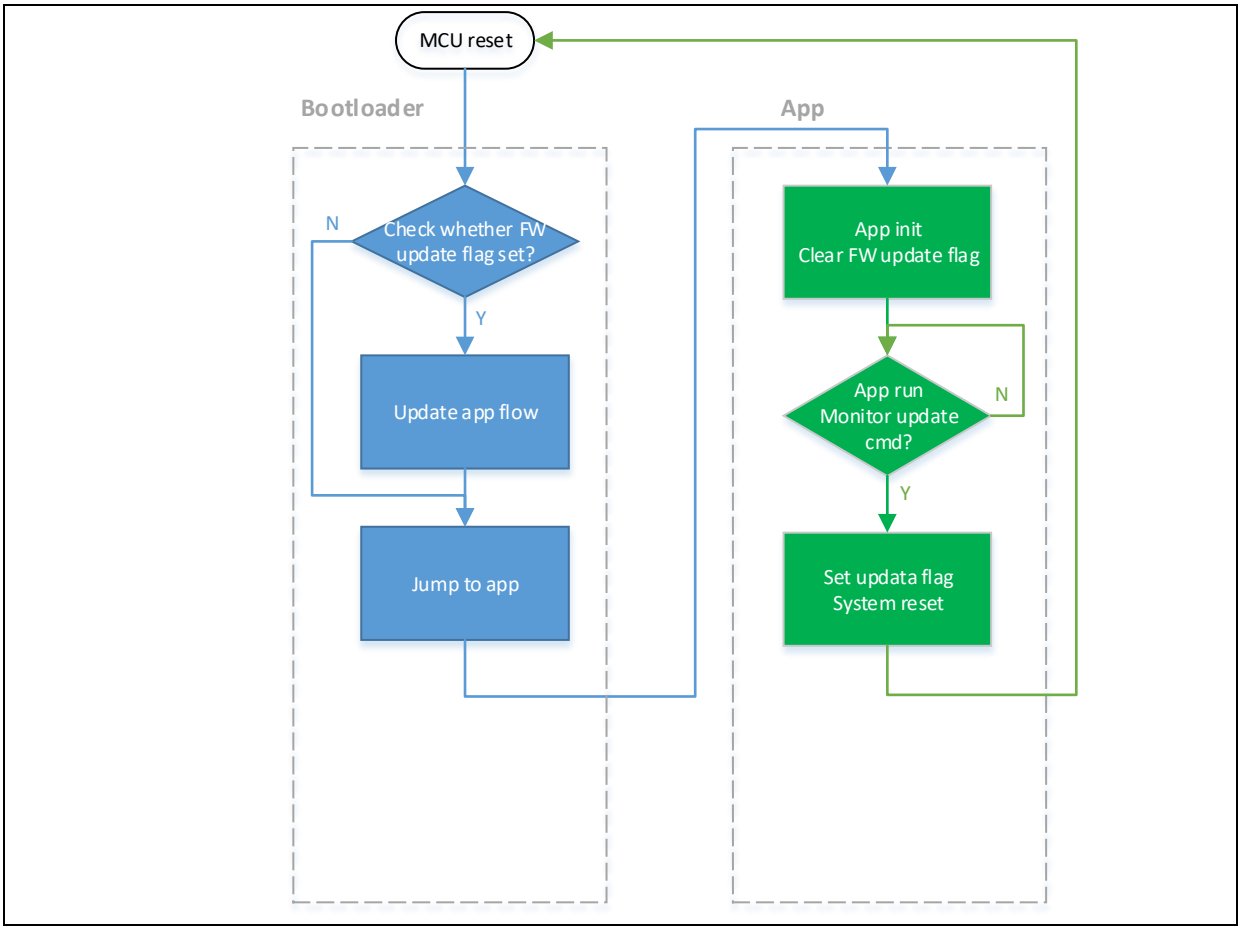
ITEM	Address and Size
app code	address 2: 0x800 4000 size: 0x40000(256K Byte)
bootloader code	address 1: 0x800 0000 size: 0x4000(16K Byte)

注：bootloader区域最后一个扇区，用于存放防止升级过程掉电的flag，用户编译修改bootloader时，要保证不覆盖flag的地址。

2.2 执行流程

IAP 分为 Bootloader 和 App 两部分，应用在 App 中执行，升级过程在 bootloader 中执行。程序执行整体流程框图如下：

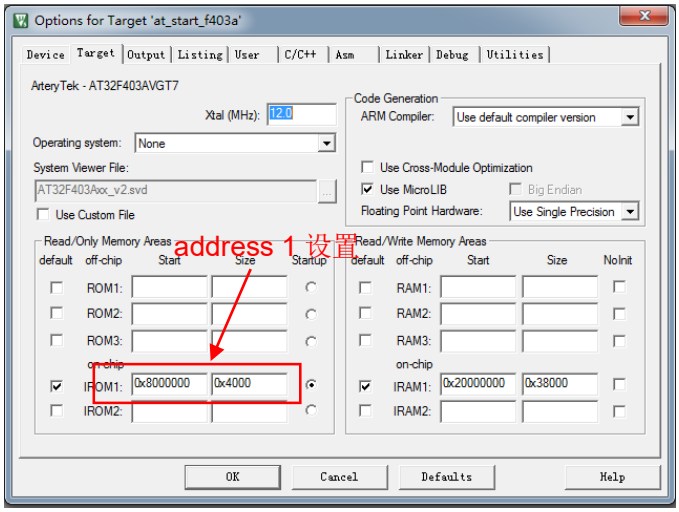
图 3. 程序执行流程



2.3 bootloader project 设置

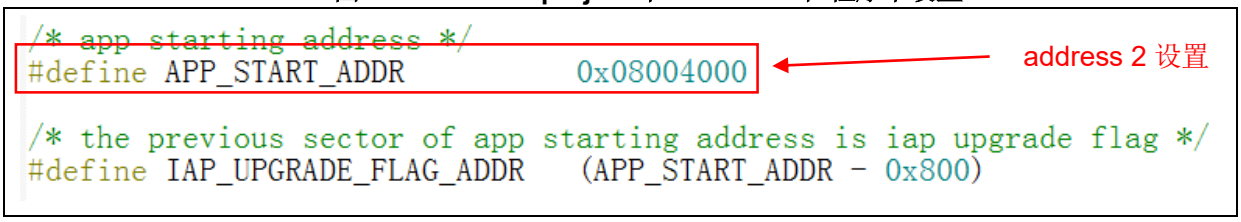
1) Keil设置

图 4. bootloader project 中 address 1 在 Keil 设置



2) bootloader源程序修改Iap.h文件中

图 5. bootloader project 中 address 2 在程序中设置

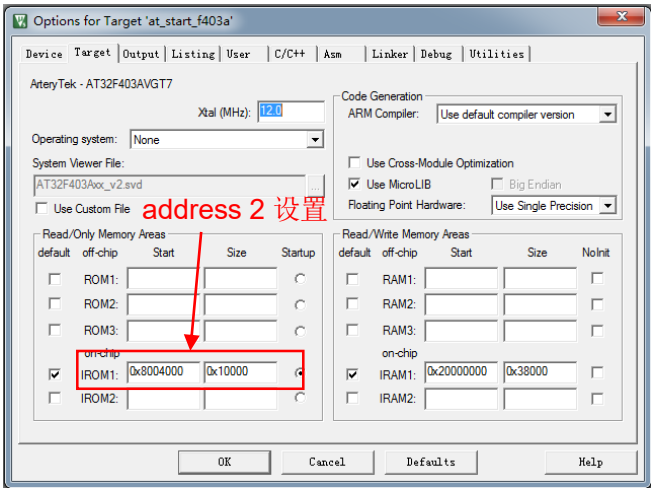


2.4 app project 设置

IAP demo 提供了 2 个 app 程序供测试用，皆以 address 2（0x800 4000）为起始地址。app1 LED3 闪烁，app2 LED4 闪烁。以 app1 为例，设计步骤如下：

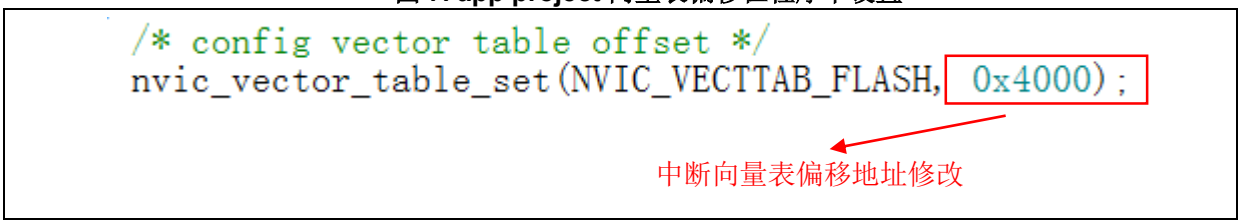
1) Keil工程设置

图 6. app project 中 address 2 在 Keil 设置



2) app1 源程序设置

图 7. app project 向量表偏移在程序中设置



3) 编译生成bin文件

通过 User 选项卡，设置编译后调用 fromelf.exe，根据.axf 文件生成.bin 文件，用于 IAP 更新。通过以上 3 个步骤，我们就可以得到一个.bin 的 APP 程序，通过 bootloader 程序即可实现更新。

4) 开启debug app code功能

如果在设计 app code 过程中需要对 app project 进行单独调试，请按照以下操作。

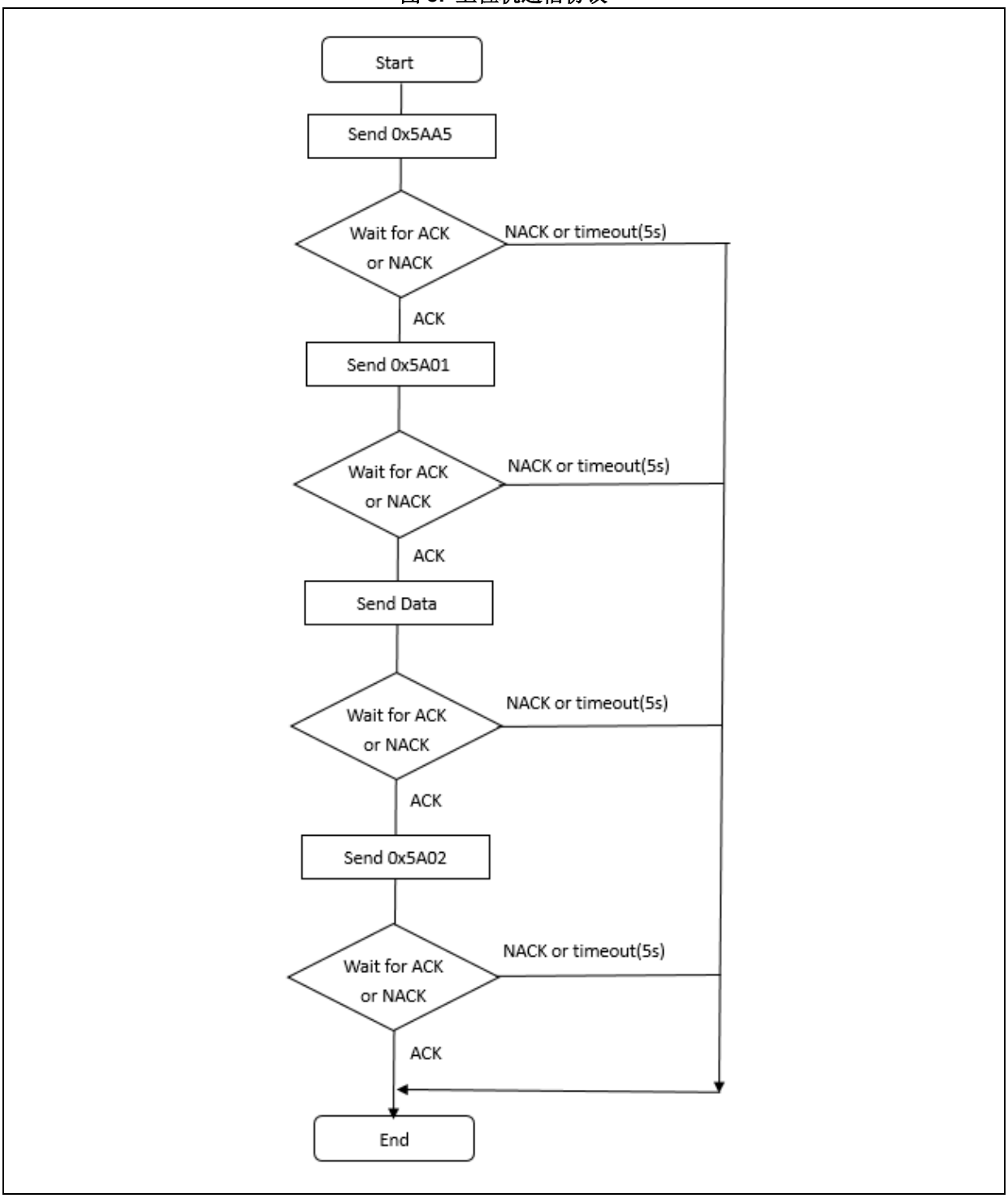
a) 先下载bootloader工程

b) 再调试app工程

3 bootloader/app 与上位机串口通信协议

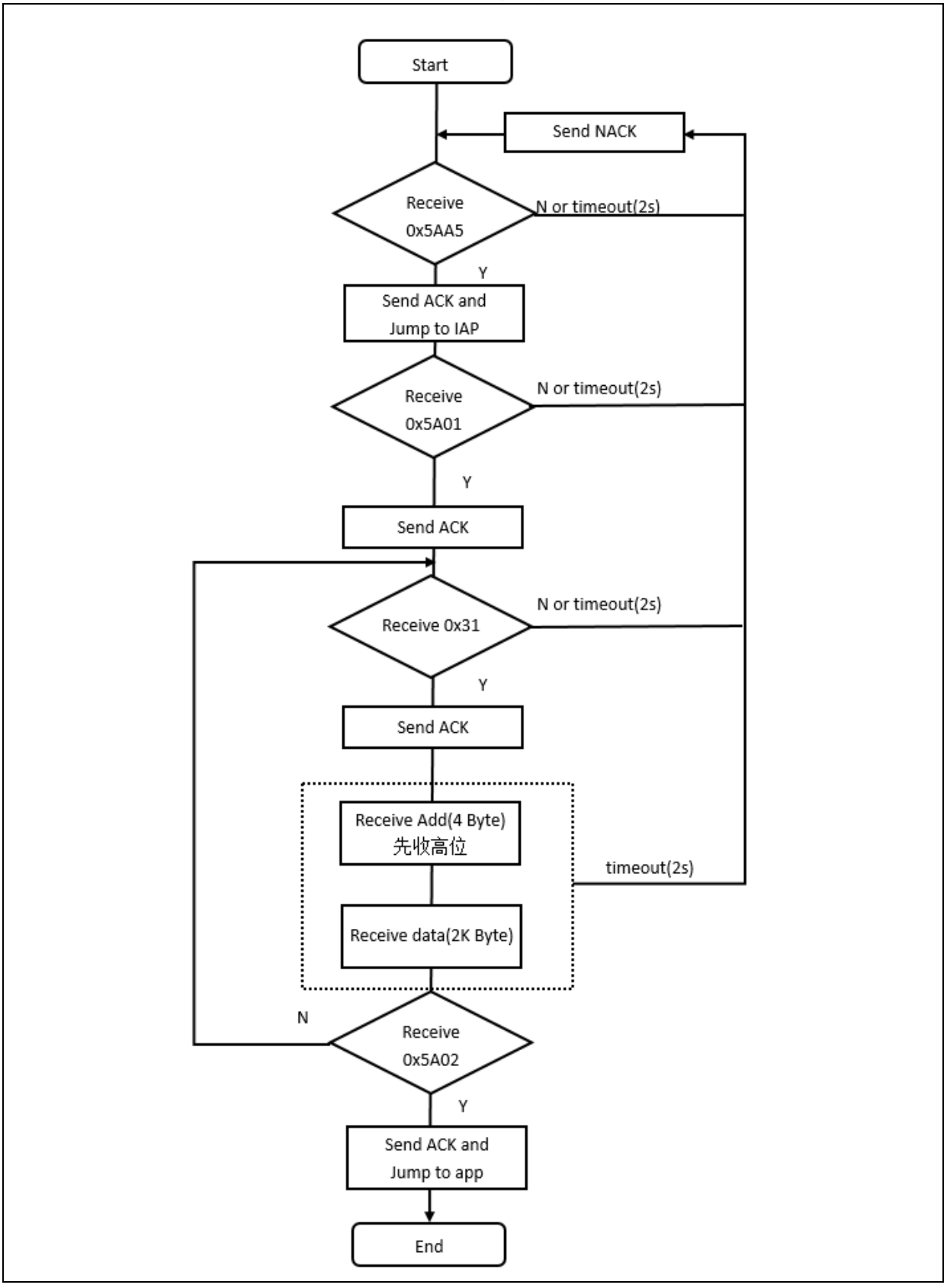
1) 上位机通信协议

图 8. 上位机通信协议



2) IAP 端下位机通信协议

图 9. IAP 端下位机通信协议



注: *ACK: 0xCCDD*

NACK: 0xEEFF

Data: 0x31+ Addr + 数据 + chenksum（1byte）

Addr：4bytes，高位在前

Kbytes，下载数据，不足2K内容填充0xFF

Checksum：1byte，4bytes的Addr + 2KBytes数据的校验和的低八位

4 版本历史

表 2. 文档版本历史

日期	版本	变更
2021.05.21	2.0.0	初始版本
2022.01.19	2.0.1	增加程序执行流程图
2023.02.08	2.0.2	增加 sourcecode 在 BSP 中的路径描述

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，且采购商应独力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。