

AT32 IAP using the CAN

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。本应用笔记使用自定义CAN协议实现IAP功能。

此应用笔记包含如下程序：

- IAP_Programmer.exe上位机软件和
- 嵌入式IAP CAN示例的源代码
- 嵌入式APP CAN示例的源代码
- USB转CAN示例源代码

支持型号列表：

支持型号	AT32F403A/407
------	---------------

目录

1	概述.....	5
1.1	AT32 CAN IAP 快速使用方法	5
1.1.1	硬件资源	5
1.1.2	软件资源	5
1.2	IAP demo 使用.....	6
2	AT32 CAN IAP 程序设置	7
2.1	地址分布	7
2.2	执行流程.....	7
2.3	bootloader project 设置.....	7
2.4	app project 设置.....	8
3	Bootloader CAN 通信协议.....	10
3.1	上位机升级流程	10
3.2	IAP 端下位机通信处理流程.....	10
3.3	升级命令详解.....	11
3.3.1	Get Info 命令.....	11
3.3.2	Upgrade Start 命令	12
3.3.3	Read Mmemory 命令.....	12
3.3.4	Write Memory 命令	12
3.3.5	Get CRC 命令	13
3.3.6	JUMP 命令.....	13
3.3.7	Upgrade Finish 命令.....	13
4	USB 转 CAN 桥接.....	14
4.1	数据转发流程.....	14
4.2	USB CAN 数据转发	14
5	版本历史	16

表目录

表 1. 地址分布	7
表 2. 文档版本历史	16

图目录

图 1 连接示意图.....	5
图 2. IAP demo 上位机.....	6
图 3. 程序执行流程.....	7
图 4. bootloader project 中 address 1 在 Keil 设置.....	8
图 5. bootloader project 中 address 2 在程序中设置.....	8
图 6. app project 中 address 2 在 Keil 设置.....	8
图 7. app project 向量表偏移在程序中设置.....	9
图 8. 上位机升级流程.....	10
图 9. IAP 端下位机命令.....	11
图 10 数据转发流程.....	14

1 概述

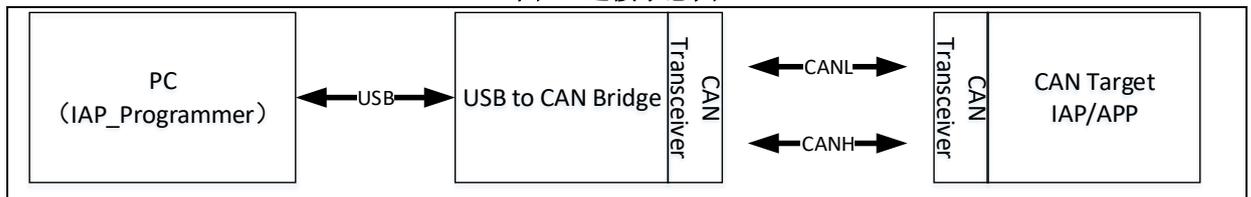
1.1 AT32 CAN IAP 快速使用方法

使用 CAN 进行 IAP 升级，此示例使用 PC 上位软件（IAP_Programmer.exe）通过一个 USB 转 CAN 的桥接设备和目标板进行通信，IAP Demo 和桥接设备默认 CAN 波特率使用 500K。

如下是连接示意图：

- PC 通过 USB 连接 USB to CAN Bridge（使用 AT-START 实现一个转接设备，代码参考 usb_vcp_bridge_can 源代码）
- USB to CAN Bridge 通过 CAN Transceiver 连接 CAN 目标板（注意使用 CAN 必须要使用 CAN Transceiver，CANL 连接 CANL, CANH 连接 CANH）

图 1 连接示意图



1.1.1 硬件资源

1. USB 转 CAN 桥接板（目前使用一块 AT-START-AT32F403A 实现 USB 转 CAN）
 - 1) USB(PA11/PA12)
 - 2) CAN(PB8/PB9)
 - 3) 指示灯 LED2/LED3/LED4
 - 4) AT-START实验板
2. AT-START-AT32F403A CAN IAP 目标板
 - 1) 指示灯LED2/LED3/LED4
 - 2) CAN(PB8/PB9)
 - 3) AT-START实验板
3. CAN Transceiver 转接板

CAN Transceiver 转接板用于USB to CAN Bridge 和 CAN 目标板的连接

注意：USB转CAN桥接板的波特率和CAN IAP目标板波特率要配置一致

另外要使用CAN通信，必须外接CAN转换芯片。

1.1.2 软件资源

- 1) ToolRelease
 - IAP_Programmer.exe, PC机tool, 用于演示IAP升级流程
 - IAP_Programmer_V2.x.x, PC tool 源代码
- 2) SourceCode\utilities
 - can_iap_demo, IAP源程序, 运行LED2闪烁
 - usb_vcp_bridge_can, USB转CAN桥接源程序
 - can_app_led3_toggle, app1源程序, 运行时LED3闪烁
 - can_app_led4_toggle, app2源程序, 运行LED4闪烁

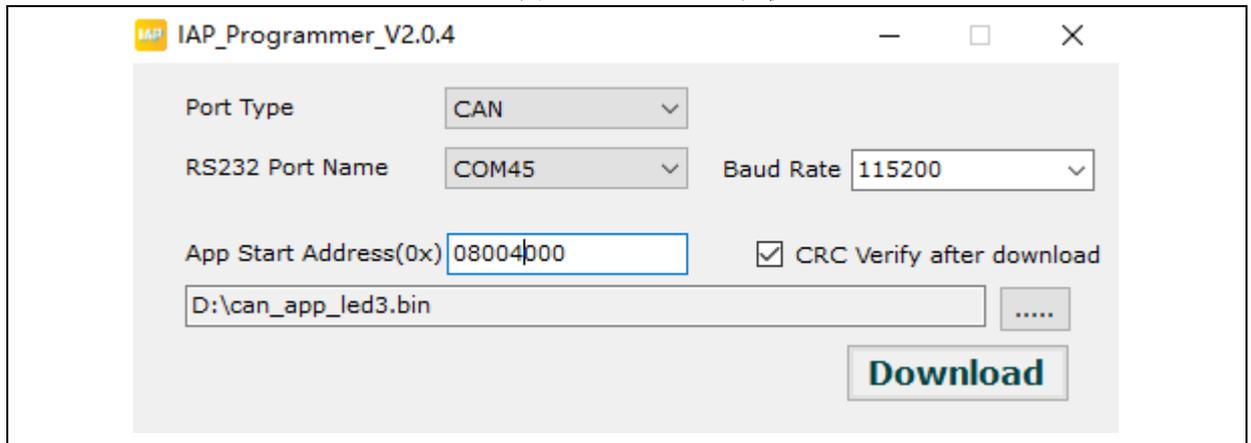
注：示例工程基于keil v5建立，若用户需要在其他编译环境上使用，请参考AT32F403A_407_Firmware_Library_V2.

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can_iap_demo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usb_vcp_bridge_can工程源程序，下载到USB转CAN转接板
- 3) 连接转接板和目标板的CANH/CANL
- 4) 将转接板USB接入到PC，可以看到会有对应的虚拟串口设备COMxx
- 5) 打开IAP_Programmer.exe
- 6) 选择正确的串口、APP下载地址和bin文档，点击Download下载，如下图
- 7) 观察LED2/3/4闪烁，LED2闪烁can IAP工作，LED3闪烁-app1工作，LED4闪烁-app2工作

图 2. IAP demo 上位机



2 AT32 CAN IAP 程序设置

2.1 地址分布

表 1. 地址分布

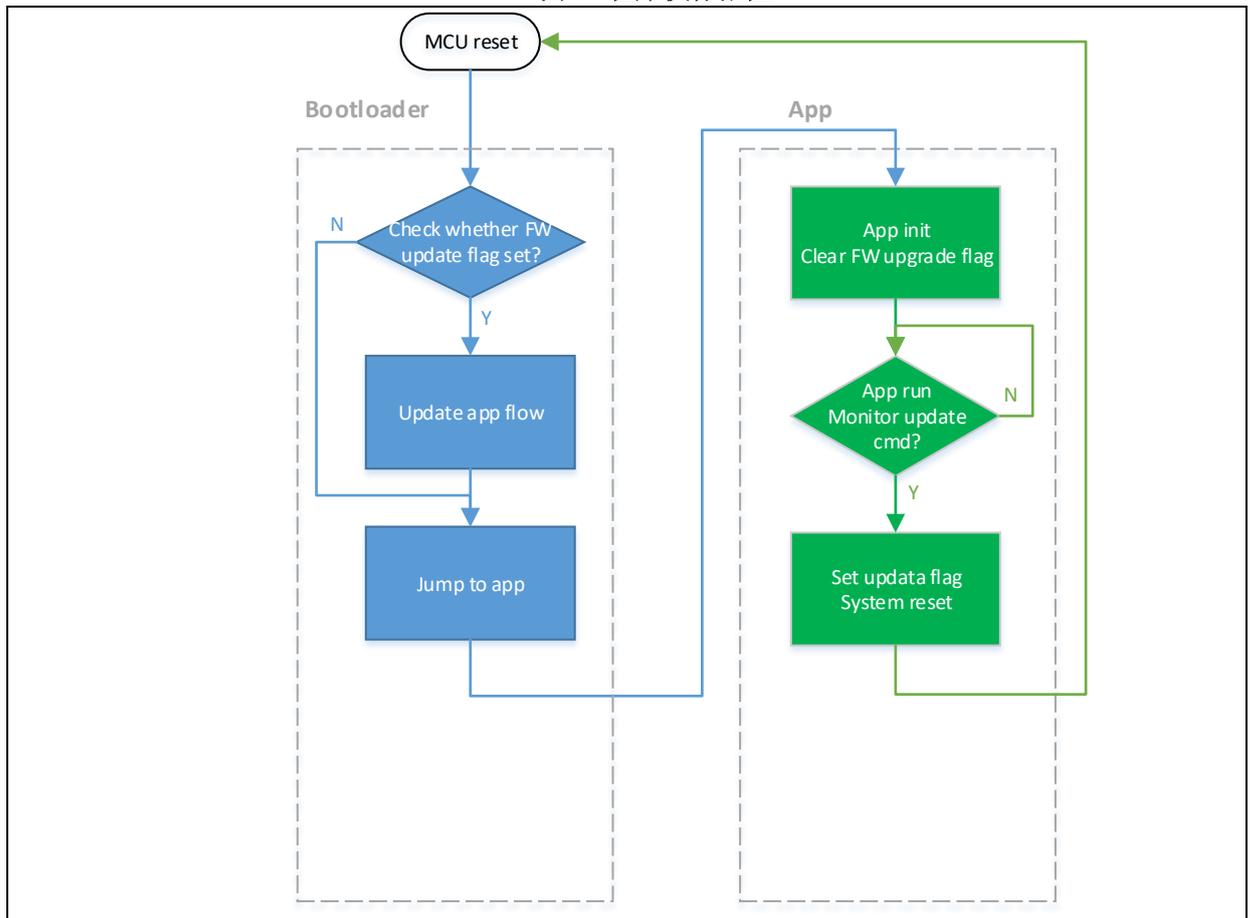
ITEM	Address and Size
app code	address 2: 0x800 4000 size: 0x40000(256K Byte)
bootloader code	address 1: 0x800 0000 size: 0x4000(16K Byte)

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:

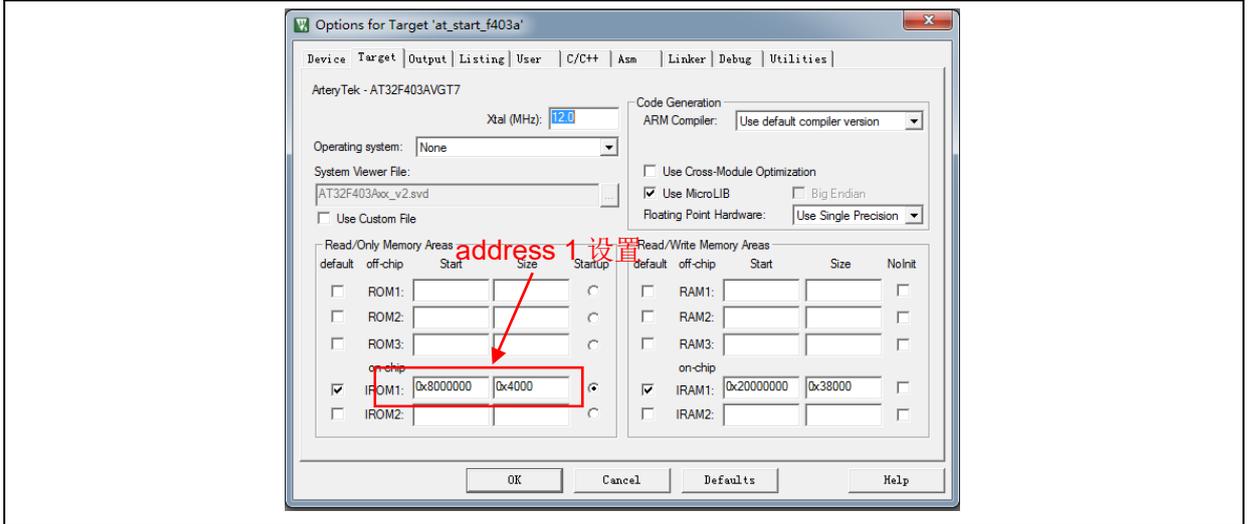
图 3. 程序执行流程



2.3 bootloader project 设置

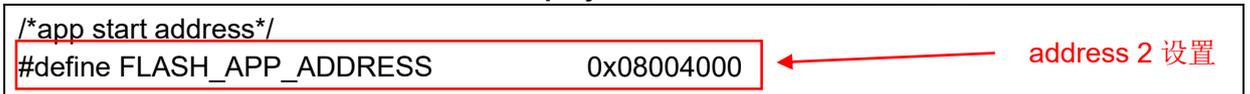
1) Keil设置

图 4. bootloader project 中 address 1 在 Keil 设置



2) bootloader源程序修改lap.h文件中

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

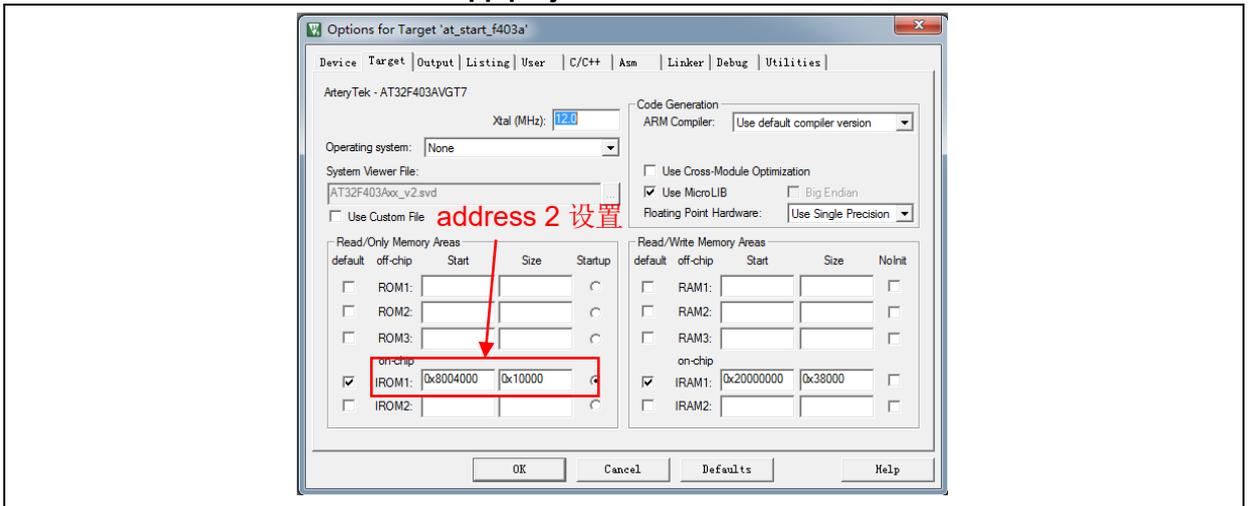


2.4 app project 设置

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

```
/* config vector table offset */  
nvic_vector_table_set(NVIC_VECTTAB_FLASH, 0x4000);
```

中断向量表偏移地址修改

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 CAN 通信协议

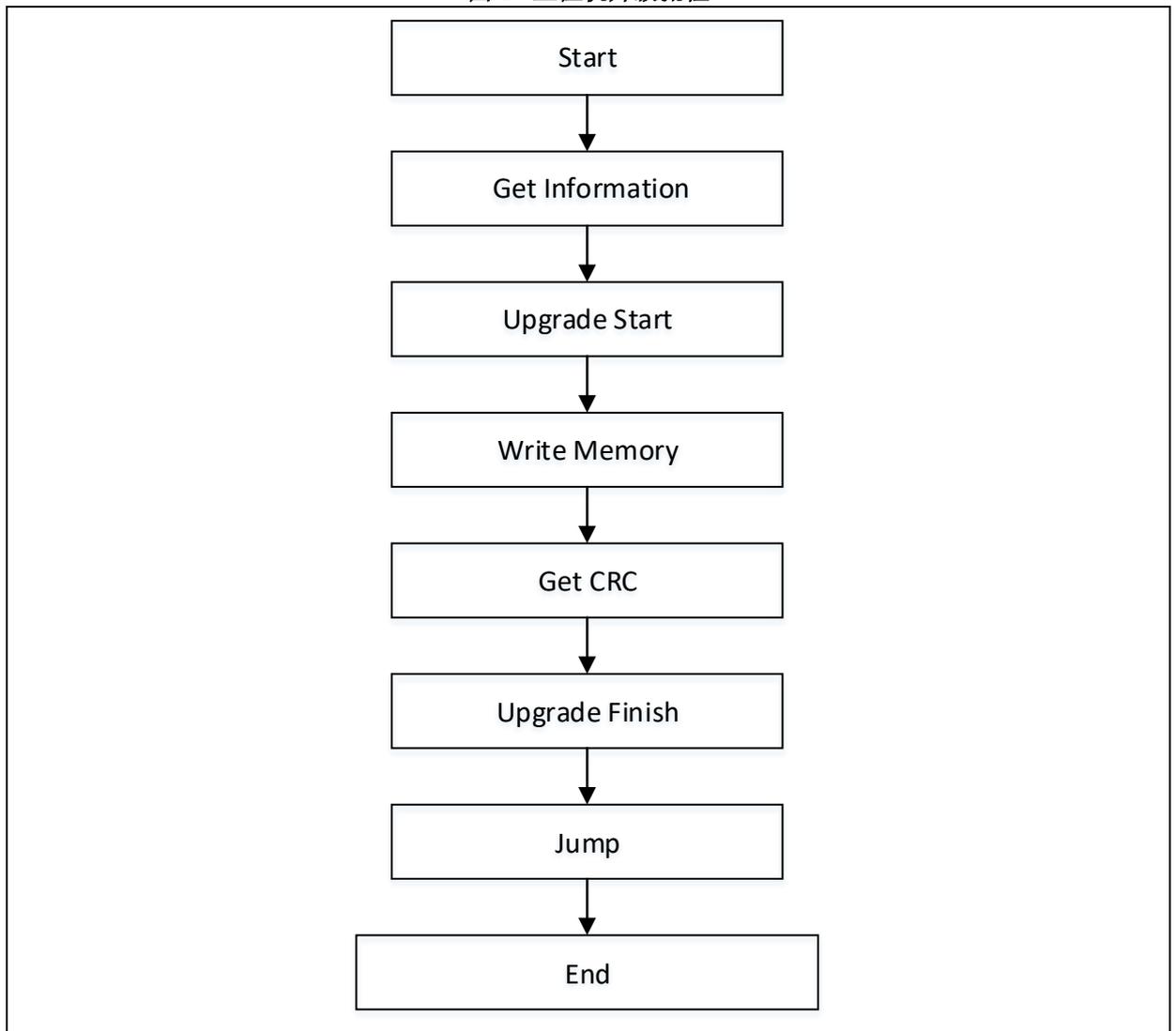
3.1 上位机升级流程

如下下图是上位机升级流程：

- 获取设备信息：包括 app 起始地址，app flash size 大小，sector size 大小
- 发送开始升级命令
- 写存储器
- 获取 CRC 进行校验
- 发送升级完成命令
- 跳转到 APP 执行

具体命令操作参考命令详解

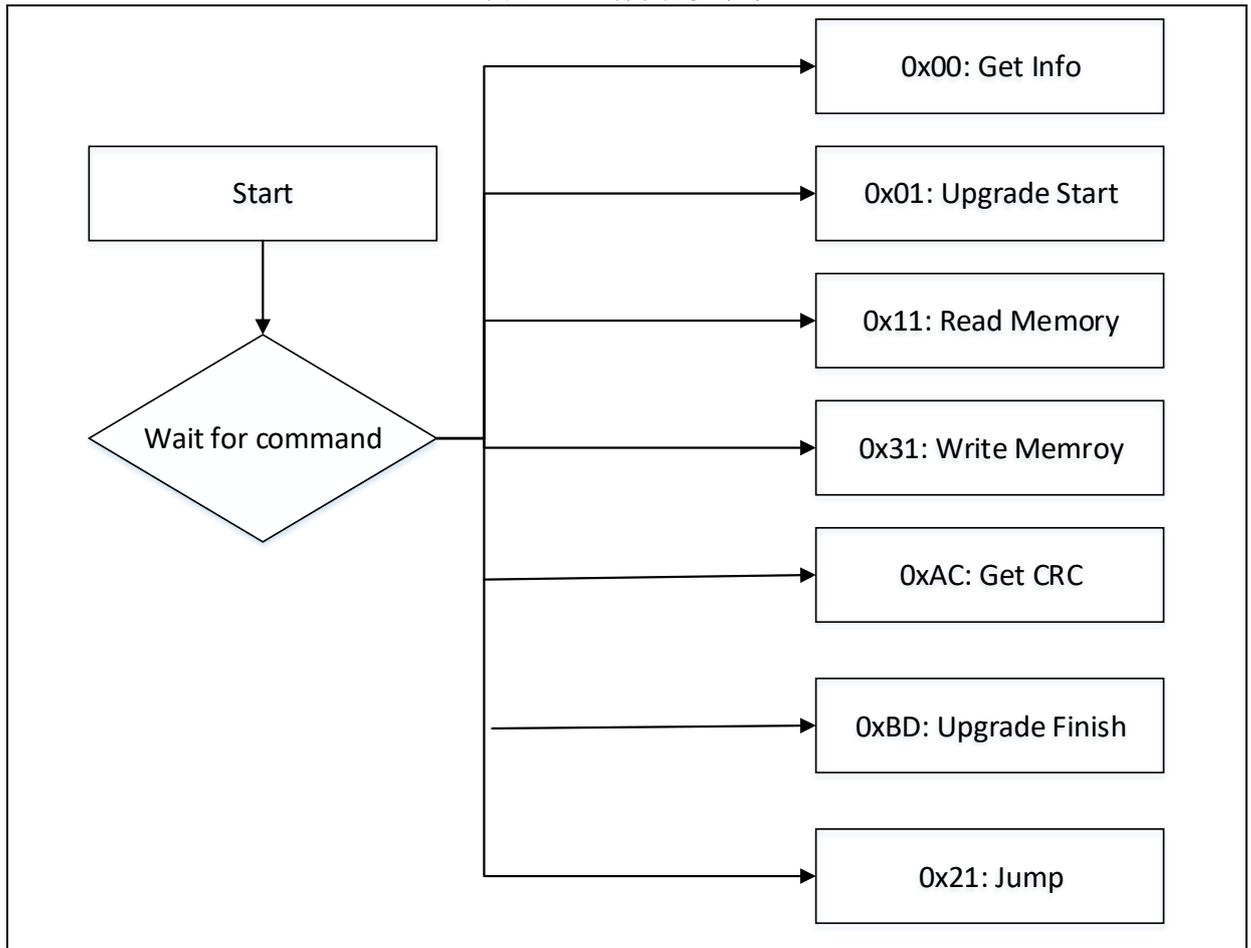
图 8. 上位机升级流程



3.2 IAP 端下位机通信处理流程

IAP 下位机端会通过接收主机端发送的命令做数据处理，包括读写数据，crc 校验等。

图 9. IAP 端下位机命令



注: ACK: 0x79
NACK: 0x1F

3.3 升级命令详解

本节主要说明 CAN IAP 每条协议命令的作用和使用方法，包括主机端和设备端对命令解析的流程。所有命令通过 CAN 标准帧的 ID 来确定，CAN 每一包数据最大 8 字节，例如 Get Info 命令，此时 CAN 标准帧的 ID 为 0。

注意：所有命令都使用标准帧数据帧来传递

如下是命令列表

命令	标准帧 ID	说明
Get Info ¹⁾	0x00	获取 APP 信息，app 起始地址，app flash size, sector size
Upgrade Start	0x01	开始升级
Read Memory	0x11	读取 APP Flash 数据
write memory	0x31	写升级 APP Flash 数据
Jump	0x21	跳转到指定的存储器地址执行程序
Write memory	0x31	写数据到存储器的指定地址
Get CRC	0xAC	获取指定大小数据的 CRC 值
Upgrade Finish	0xBD	升级结束

3.3.1 Get Info 命令

此命令用于获取 APP 信息，包括 APP 起始地址，app flash size 大小，sector size 大小
主机端协议发送流程：

发送消息	接收消息	标准帧 ID	帧长 DLC	数据	描述
1		0x00	-	-	主机发送 Get Info 命令
	1	0x00	1	ACK/NACK	当收到 NACK 时，等待 1s，继续发送 Get info 命令，一直到回 ACK 之后再走后面流程。
	2	0x00	8	1: Address MSB 2: Address 3: Address 4: Address LSB 5: app size MSB 6: app size 7: app size 8: app size LSB	返回 app 地址和 app size
	3	0x00	4	1: Sector size MSB 2: Sector size 3: Sector size 4: Sector size LSB	返回 sector size

3.3.2 Upgrade Start 命令

获取设备信息之后，可以发送此命令开始升级，设备端收到此命令之后会去删除一个 app 的标志，标志在开始升级。

主机端协议发送流程:

发送消息	接收消息	标准帧 ID	帧长 DLC	数据	描述
1		0x01	-	-	主机发送 Upgrade 命令
	1	0x01	1	ACK/NACK	命令结束

3.3.3 Read Mmemory 命令

此命令用于读取 app flash 数据

主机端协议发送流程:

发送消息	接收消息	标准帧 ID	帧长 DLC	数据	描述
1		0x11	6	1: Address MSB 2: Address 3: Address 4: Address LSB 5: Length MSB 6: Length LSB	主机发送 Read Memroy，包含长度和地址
	1	0x11	1	ACK/NACK	当收到 NACK 时，命令结束
	2	0x11	X	Data	目标地址数据
	...	0x11	X	Data	目标地址数据
	n		X	Data	目标地址数据

3.3.4 Write Memory 命令

此命令用于写 APP 数据到存储器，注意每次写入的地址必须 sector 对齐（sector 大小可通过 Get Info 获取），数据的长度为 1 个 sector 大小或者小于 1 个 sector。每个 sector 的数据最好是一次写完。

主机端协议发送流程:

发送消息	接收消息	标准帧 ID	帧长 DLC	数据	描述
1		0x31	6	1: Address MSB 2: Address 3: Address 4: Address LSB 5: Length MSB 6: Length LSB	主机发送 Write Memory 命令

发送消息	接收消息	标准帧 ID	帧长 DLC	数据	描述
	1	0x31	1	ACK/NACK	当收到 NACK 时, 命令结束
2		0x31	X	Data	目标地址数据
...		0x31	X	Data	目标地址数据
n		0x31	X	Data	目标地址数据
	2	0x31	1	ACK/NACK	命令结束, 写入完成

3.3.5 Get CRC 命令

此命令用于获取指定地址指定大小数据的 CRC, 地址和大小必须 4 字节对齐

主机端协议发送流程:

发送消息	接收消息	标准帧 ID	帧长 DLC	数据	描述
1		0xAC	8	1: Address MSB 2: Address 3: Address 4: Address LSB 5: Length MSB 6: Length 7: Length 8: Length LSB	主机发送 Get CRC 命令
	1	0xAC	1	ACK/NACK	当收到 NACK 时, 命令结束
	2	0xAC	4	CRC MSB CRC CRC CRC LSB	4 字节的 CRC 值

3.3.6 JUMP 命令

此命令用于跳转到 APP 代码执行

主机端协议发送流程:

发送消息	接收消息	标准帧 ID	帧长 DLC	数据	描述
1		0x21	4	1: Address MSB 2: Address 3: Address 4: Address LSB	主机发送 Jump, 包含地址
	1	0x21	1	ACK/NACK	当收到 NACK 时, 命令结束

3.3.7 Upgrade Finish 命令

此命令用于设置升级完成标志。

主机端协议发送流程:

发送消息	接收消息	标准帧 ID	帧长 DLC	数据	描述
1		0xBD	-	-	主机发送 Upgrade Finish 命令
	1	0xBD	1	ACK/NACK	命令结束

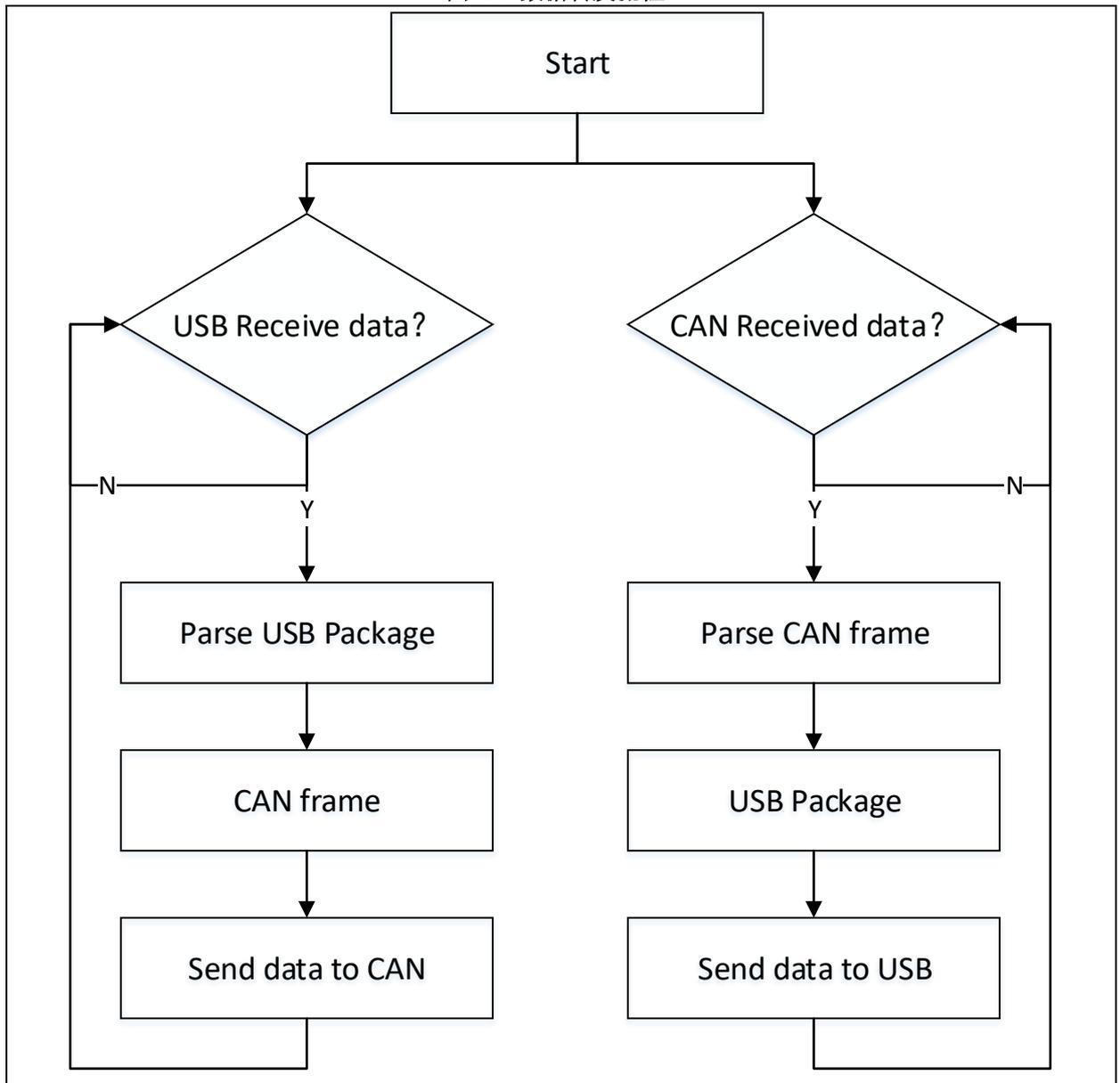
4 USB 转 CAN 桥接

USB 转 CAN 桥接使用一块 AT32F403A 的 AT-START 开发板实现，USB 端实现一个虚拟串口设备，转接板通过 USB 与 PC 上位机通信，然后通过 CAN 与下位机进行通信，实现 USB 与 CAN 数据的透传功能。注意转接板的 CAN 波特率与下位机的 CAN 波特率要配置相同。

4.1 数据转发流程

如下是数据转发流程

图 10 数据转发流程



4.2 USB CAN 数据转发

USB 转 CAN 桥接按照 CAN 协议进行封装，封装格式：
标准帧 ID(1 字节)+帧长 (1 字节)+数据 (数据最大 8 字节)

USB 数据转到 CAN 总线：

USB 到 CAN 按照一包 USB 数据转成一帧 CAN 数据。

收到一包 USB 数据如：0x01 0x02 0x11 0x22

0x01：表示标准帧 ID

0x02：表示数据长度

0x11,0x22：表示具体数据

转换成 CAN 帧后：

ID=0x01

DLC=0x02

DATA0=0x11

DATA1=0x22

CAN 总线数据转到 USB：

CAN 同样按照一帧数据转换成一包 USB 数据：

收到一帧 CAN 数据：ID=0x02, DLC=0x03, DATA0=0x11, DATA1=0x22, DATA2=0x33

转换成 USB 包之后：0x02 0x03 0x11,0x22,0x33

0x02：表示标准帧 ID

0x03：表示数据长度

0x11,0x22,0x33：表示具体数据

以 Write Memory 为例写 32 字节数据到 0x08004000

主机端流程：

发送 1：0x31 0x06 0x08 0x00 0x40 0x00 0x00 0x20 （ID+长度+6 字节数据）

接收 1：0x31 0x01 0x79 （ID+长度+ACK）

发送 2：0x31 0x08 0xAA 0xAA 0xAA 0xAA 0xAA 0xAA 0xAA 0xAA 0xAA (ID+长度+8 字节数据)

发送 3：0x31 0x08 0xAA 0xAA 0xAA 0xAA 0xAA 0xAA 0xAA 0xAA 0xAA (ID+长度+8 字节数据)

发送 4：0x31 0x08 0xAA 0xAA 0xAA 0xAA 0xAA 0xAA 0xAA 0xAA 0xAA (ID+长度+8 字节数据)

发送 5：0x31 0x08 0xAA 0xAA 0xAA 0xAA 0xAA 0xAA 0xAA 0xAA 0xAA (ID+长度+8 字节数据)

5 版本历史

表 2. 文档版本历史

日期	版本	变更
2023.03.15	2.0.0	初始版本

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，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) 汽车应用或汽车环境；(D) 航天应用或航天环境，且/或(E) 武器。因雅特力产品不是为前述应用设计的，而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，风险由购买者单独承担，并且独力负责在此类相关使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。

© 2023 雅特力科技 (重庆) 有限公司 保留所有权利