

AT32F415 IAP using a USB host connected U disk

Introduction

This document describes how to implement firmware upgrading by connecting AT32F415 USB HOST to USB disk.

Note: the code in this application note is developed based on Artery BSP V2.x.x (version). For other BSP editions, attention should be paid to their differences in use.

Applicable products:

Product	AT32F415 series
---------	-----------------

Contents

1	IAP online upgrade logic.....	5
2	Upgrade information using USB host to U disk	7
2.1	Features	7
2.2	Program design	7
2.2.1	Address space.....	7
2.2.2	BIN file name format (suffix. BIN).....	8
2.2.3	Jump to APP code.....	8
3	Start upgrade operation	9
3.1	Hardware resources	9
3.2	Software resources	9
3.3	Enter IAP mode	9
3.4	Connect USB disk to AT32F415	9
3.5	Auto upgrade.....	9
4	Revision history.....	11

List of tables

Table 1. Address space.....	7
Table 2. Document revision history.....	11

List of figures

Figure 1. IAP code execution flow	5
Figure 2. Serial interface print information	10

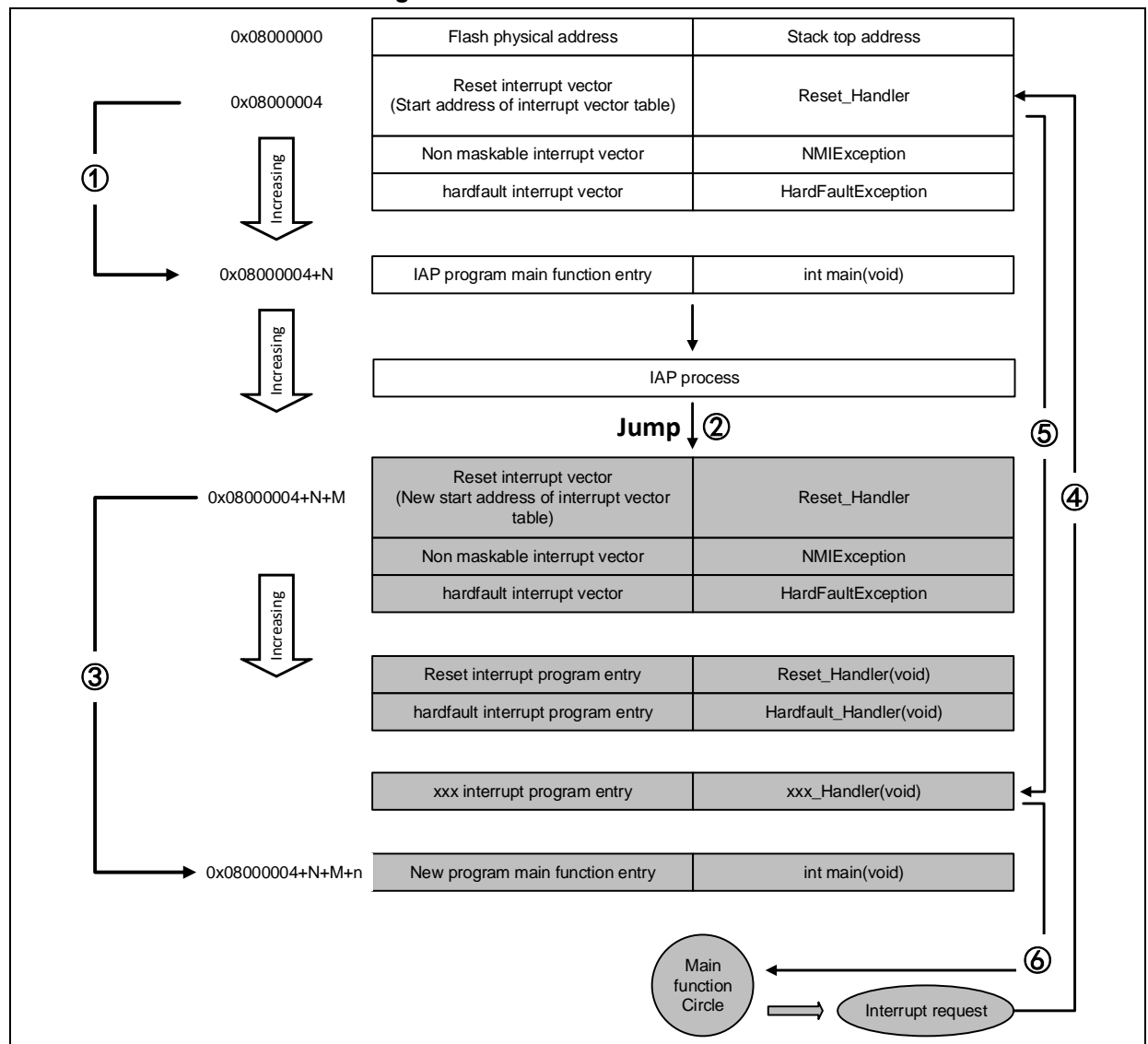
1 IAP online upgrade logic

IAP is an acronym of In Application Programming. It allows user applications to program user Flash area during run time, designed to make it easier for users to upgrade product firmware via reserved communication interfaces after the launch of a product.

To implement IAP feature (user application upgrade), it is necessary to write two project codes during the design of firmware programs. The first project code is used to update the second code by receiving data or code through communication interfaces such as USB or USART, rather than executing normal functional operations; the second project code is the real functional code. Both codes are programmed in User Flash. After power-on, it is the first code that starts operations as follows:

- 1) Check if the second code needs upgrade or not
- 2) If no need upgrade, jump to step 4
- 3) Perform update operation
- 4) Jump to the second code
- 5) Start code run

Figure 1. IAP code execution flow



As illustrated in Figure 1, after a reset, the AT32 MCU device still takes reset interrupt vector address from the address 0X08000004, enter reset interrupt service routine, and then jump to main function of IAP after the completion of reset interrupt service routine, as marked in ①. Write new APP code into AT32 Flash (gray area above), the reset interrupt vector start address of the new program is 0X08000004+N+M. Then jump to the new reset interrupt vector table, pick up the new reset interrupt vector address, enter new reset interrupt service routine and jump to the main function of new program, as marked in ② and ③. The main function is an endless loop. Besides, it should be noted that there are two interrupt vector tables on different locations of AT32 MUC Flash at this time.

In the process of main function execution, if an interrupt request is received by CPU, the PC pointer still jumps to the interrupt vector table at the address 0X08000004, instead of the new interrupt vector table, as shown in ④; The program then jumps to the new interrupt service routine corresponding to interrupt sources according to the configured offset address of interrupt vector table, as shown in ⑤; After the completion of interrupt service routine, the program returns to main function and continues to run, as shown in ⑥.

Based on the above analysis, we can see that IAP has two requirements to be met:

- 1) The new application must start from certain address with x offset after the IAP program
- 2) The interrupt vector table of the new application must be moved according to an offset of x.

2 Upgrade information using USB host to U disk

The AT32F415 series supports USB host mode, and thus it can be connected to a USB disk. To read BIN files in USB disk through FAQ32 file system can directly upgrade firmware.

Very simple steps:

- Copy BIN files to be upgraded to USB disk
- Connect USB disk to AT32F415 MCU device, and start automatic detection and upgrading of BIN files in USB disk
- After upgrading, jump to APP

2.1 Features

- 32 Kbyte reserved room is required, indicating that APP start address has to be located after the 32KB (0x08008000)
- Only support FAT32 file system
- Only the first FAQ32 file system subarea supports automatic BIN file detection (of several subareas, only the first one with auto BIN detection)
- Only support root directory BIN file detection
- Support several BIN file upgrade
- Automatically jump to APP upon completing upgrade
- APP start address must be page aligned

2.2 Program design

2.2.1 Address space

Table 1. Address space

0x08000000	
~	IAP
0x08008000	
0x08008000	
~	APP available address
....	

2.2.2 BIN file name format (suffix. BIN)

Specify download address (format 1)

File name format: (4Byte) IAP_+(8Byte) address+.BIN

For example, to download a BIN file into the address space with 0x08008000 start address

The file name should be named as: IAP_08008000.BIN

Note 1: It is necessary to ensure that 8Byte address is accessible to APP. Other formats of BIN files would not be upgraded.

Note 2: BIN files must be copied into USB disk root directory.

2.2.3 Jump to APP code

After downloading firmware into Flash, it will automatically jump to the default APP address 0x08008000.

3 Start upgrade operation

3.1 Hardware resources

- 1) LED2/LED3/LED4
- 2) USB(PA11/PA12)
- 3) AT-START-F415 V1.0 test board

3.2 Software resources

- 1) SourceCode
 - AN0030_SourceCode_V2.0.0\utilities\AN0030_demo, IAP source code
 - AN0030_SourceCode_V2.0.0\libraries, AT32 peripheral library
- 2) Doc
 - 《AN0030_AT32F415_IAP_using_a_USB_host_connected_U_disk_ZH_V2.0.0》

Note: All projects are built around keil 5.

For other compiling environments, refer to “AT32F415_Firmware_Library_CH\project\at_start_f415\templates” (such as IAR6/7/8, keil 4/5) for minor modification.

3.3 Enter IAP mode

If firmware has already been upgraded, just keep pressing User button and then Reset button to enter IAP mode. Otherwise, if firmware has not been upgraded, directly press Reset button to enter IAP mode.

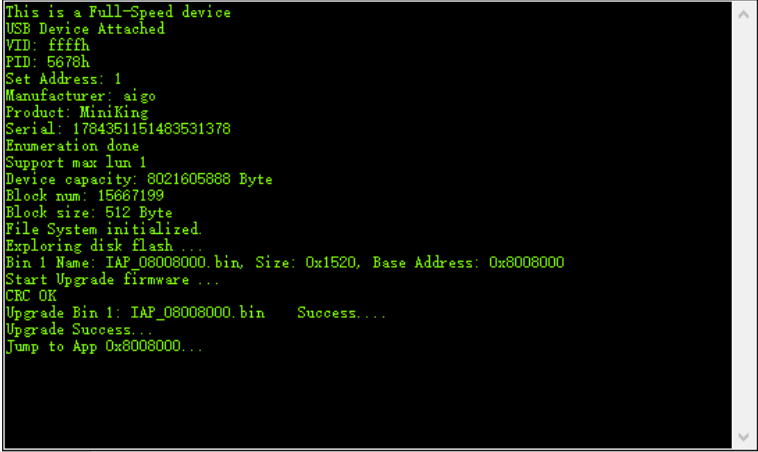
3.4 Connect USB disk to AT32F415

Connect FAT32-format USB disk to the USB standard A socket on AT32F415 board. It is necessary to copy BIN files into USB disk root directory in advance, and file name follows the naming rules described in section 2.2.2.

3.5 Auto upgrade

Demo will output print upgrade status via USART1. It also automatically locate the BIN files under root directory that meet the requirements, before starting auto upgrade. After successful upgrade, jump to the default APP address automatically.

Figure 2. Serial interface print information



```
This is a Full-Speed device
USB Device Attached
VID: ffffh
PID: 5678h
Set Address: 1
Manufacturer: sigo
Product: MiniKing
Serial: 1784351151483531378
Enumeration done
Support max lun 1
Device capacity: 8021605888 Byte
Block num: 15667199
Block size: 512 Byte
File System initialized.
Exploring disk flash ...
Bin 1 Name: IAP_08008000.bin, Size: 0x1520, Base Address: 0x8008000
Start Upgrade firmware ...
CRC OK
Upgrade Bin 1: IAP_08008000.bin  Success....
Upgrade Success...
Jump to App 0x8008000...
```

4 Revision history

Table 2. Document revision history

Date	Revision	Changes
2022.2.10	2.0.0	Initial release

IMPORTANT NOTICE – PLEASE READ CAREFULLY

Purchasers are solely responsible for the selection and use of ARTERY's products and services; ARTERY assumes no liability for purchasers' selection or use of the products and the relevant services.

No license, express or implied, to any intellectual property right is granted by ARTERY herein regardless of the existence of any previous representation in any forms. If any part of this document involves third party's products or services, it does NOT imply that ARTERY authorizes the use of the third party's products or services, or permits any of the intellectual property, or guarantees any uses of the third party's products or services or intellectual property in any way.

Except as provided in ARTERY's terms and conditions of sale for such products, ARTERY disclaims any express or implied warranty, relating to use and/or sale of the products, including but not restricted to liability or warranties relating to merchantability, fitness for a particular purpose (based on the corresponding legal situation in any unjudicial districts), or infringement of any patent, copyright, or other intellectual property right.

ARTERY's products are not designed for the following purposes, and thus not intended for the following uses: (A) Applications that have specific requirements on safety, for example: life-support applications, active implant devices, or systems that have specific requirements on product function safety; (B) Aviation applications; (C) Aerospace applications or environment; (D) Weapons, and/or (E) Other applications that may cause injuries, deaths or property damages. Since ARTERY products are not intended for the above-mentioned purposes, if purchasers apply ARTERY products to these purposes, purchasers are solely responsible for any consequences or risks caused, even if any written notice is sent to ARTERY by purchasers; in addition, purchasers are solely responsible for the compliance with all statutory and regulatory requirements regarding these uses.

Any inconsistency of the sold ARTERY products with the statement and/or technical features specification described in this document will immediately cause the invalidity of any warranty granted by ARTERY products or services stated in this document by ARTERY, and ARTERY disclaims any responsibility in any form.