

AT32F403A & AT32F407入门使用指南

前言

本应用入门指南旨在让用户快速使用 AT32F403Axx /AT32F407xx 进行项目开发，AT32F407 相较于 AT32F403A 增加了以太网（ETH）功能。

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	AT32F403Axx
	AT32F407xx

目录

1	雅特力初步环境准备	6
1.1	搭建 AT32 开发环境.....	6
1.1.1	调试工具	6
1.1.2	烧录工具及软件	6
1.1.3	AT32 KEIL 以及 IAR 开发环境.....	6
1.1.4	快速替代 SXX 流程.....	10
1.2	AT32F403A /AT32F407 芯片的增强功能配置	10
1.2.1	PLL 时钟设置.....	10
1.2.2	如何打开 FPU 功能（硬件浮点运算单元）	12
1.2.3	AT32F403A /AT32F407 零等待/非零等待 Flash 和内置 SRAM 大小选择配置说明 ..	14
1.2.4	加密方式(访问保护,擦写保护,外部 Flash 的加密).....	19
1.2.5	在程序中区分 AT32 与其他 IC 方法.....	24
2	下载编译过程常见问题.....	26
2.1	程序启动进入 Hard Fault Handler	26
2.2	程序下载过程出问题	26
2.2.1	显示 Error: Flash Download failed-“Cortex-M4”问题	26
2.2.2	显示 No Debug Unit Device found 问题	27
2.2.3	显示 RDDI-DAP Error 问题	27
2.2.4	ISP 串口下载时卡死问题	27
2.2.5	AT32 恢复下载.....	27
2.3	升级轩微烧录器固件	29
3	安全库区 sLib(Security Library)	31
3.1	概述	31
3.2	应用原理	31
3.3	操作安全库区	31
4	文档版本历史	32

表目录

表 1. 函数接口表.....	9
表 2. 文档版本历史	32

图目录

图 1. AT32F403A 开发板实物图.....	6
图 2. 安装 ArteryTek.AT32F403A_407_DFP	7
图 3. 安装 Keil4_AT32MCU_AddOn	7
图 4. Keil 中 Pack Installer 图标	7
图 5. Keil Debug 选项.....	7
图 6. Keil Debug 选项 Settings 设置	8
图 7. Keil Utilities 选项.....	8
图 8. 安装 IAR_AT32MCU_AddOn.....	8
图 9. IAR Debug 选项.....	9
图 10. IAR CMSIS-DAP 选项	9
图 11. SXX Flash 等待位.....	10
图 12. 使用 SXX32F103 程序在 AT32F403A /AT32F407 输出 240MHz 的时钟配置.....	11
图 13. AT32F403A /AT32F407 crm_pll_output_range 参数.....	11
图 14. PLL 自动滑顺频率切换配置.....	11
图 15. PLL 自动滑顺频率切换配置.....	12
图 16. Keil 环境中选择开启 FPU	13
图 17. Keil 环境中选择开启 FPU	13
图 18. Keil 环境中增加开启 FPU 的代码	13
图 19. IAR 环境中选择开启 FPU	14
图 20. IAR 环境中增加开启 FPU 的代码	14
图 21. ICP 工具编辑用户系统数据选择 SRAM 大小	15
图 22. 用户系统数据设置选择 SRAM 大小.....	15
图 23. ISP 工具编辑用户系统数据选择 SRAM 大小.....	16
图 24. ISP Multi-Port Programmer 工具用户系统数据设置.....	16
图 25. 定义 Extend_SRAM(void)函数修改 SRAM 大小	17
图 26. 定义 Extend_SRAM(void)函数修改 SRAM 大小	17
图 27. Keil 启动文件中修改 SRAM 大小.....	18
图 28. IAR 启动文件中修改 SRAM 大小.....	18
图 29. ISP 工具启用访问保护.....	20
图 30. ISP 工具解除访问保护.....	20

图 31. ICP 工具启用擦写保护.....	21
图 32. ICP 工具解除擦写保护.....	22
图 33. ICP 工具对外部存储器加密操作	23
图 34. ISP 工具对外部存储器加密操作	23
图 35. ISP Multi-Port Programmer 工具对外部存储器加密	24
图 36. 读取 Cortex 型号	24
图 37. 读取 PID,UID	24
图 38. 增加开启 FPU 的代码	26
图 39. 下载出现 Flash Download failed-“Cortex- 4”	26
图 40. KEIL 环境中工具 ConfigJLink_V1.0.0 的操作	28
图 41. KEIL 环境中工具 ConfigJLink_V1.0.0 的操作	28
图 42. IAR 环境中工具 ConfigJLink_V1.0.0 的操作	29
图 43. IAR 环境中工具 ConfigJLink_V1.0.0 的操作	29
图 44. 轩微烧录器固件显示图片	30

1 雅特力初步环境准备

雅特力开发环境下载地址：

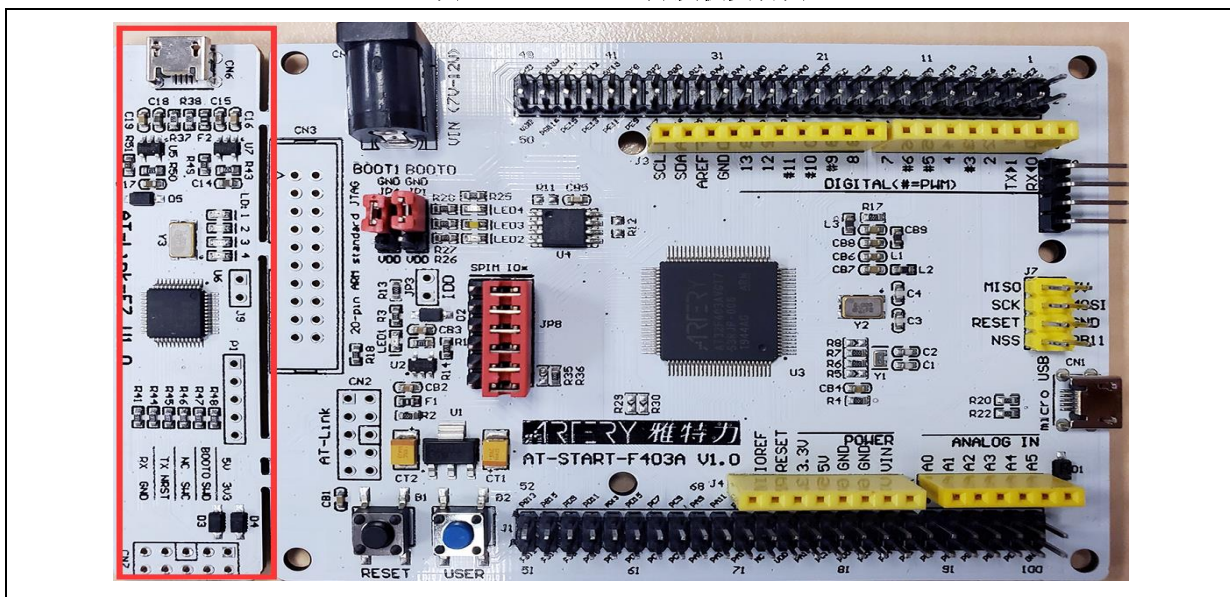
- 雅特力官方网站：<http://www.arterytek.com>

1.1 搭建 AT32 开发环境

1.1.1 调试工具

目前 AT32F403A /AT32F407 开发板都自带 AT-Link-EZ 调试工具，AT-Link-EZ 如下图左边红框所示，它也可拆开后单独搭配其他电路板使用，支持 IDE 在线调试、在线烧录、USB 转串口等功能。

图 1. AT32F403A 开发板实物图



1.1.2 烧录工具及软件

- AT 烧录工具及软件：AT-Link /AT-Link-Pro / AT-Link-ISO /AT-Link-EZ /J-Link、ICP/ISP。
- 第三方烧录工具：
 - 轩微：<https://xuanweikeji.taobao.com>
 - 迈斯威志：www.maxwiz.com.cn
 - 周立功：<http://tools.zlg.cn/tools>
 - 阿莫：<http://www.amomcu.cn>

1.1.3 AT32 KEIL 以及 IAR 开发环境

① 对于 Keil 编译系统，建议是 keil4.74 或 5.23 以上版本；

需要安装 Pack 包在 Keil 中增加 AT32 MCU 型号，Pack 包可以从雅特力官网下载。对于 Keil_v5 版本需要将 Keil5_AT32MCU_AddOn 解压后安装对应 ArteryTek.AT32F403A_407_DFP，对于 Keil_v4 版本需要安装 Keil4_AT32MCU_AddOn；默认情况下，安装时可以自动识别到 Keil 的安装路径，如果识别不到或者不正确，需要手动选择 Keil 的安装路径。

图 2. 安装 ArteryTek.AT32F403A_407_DFP

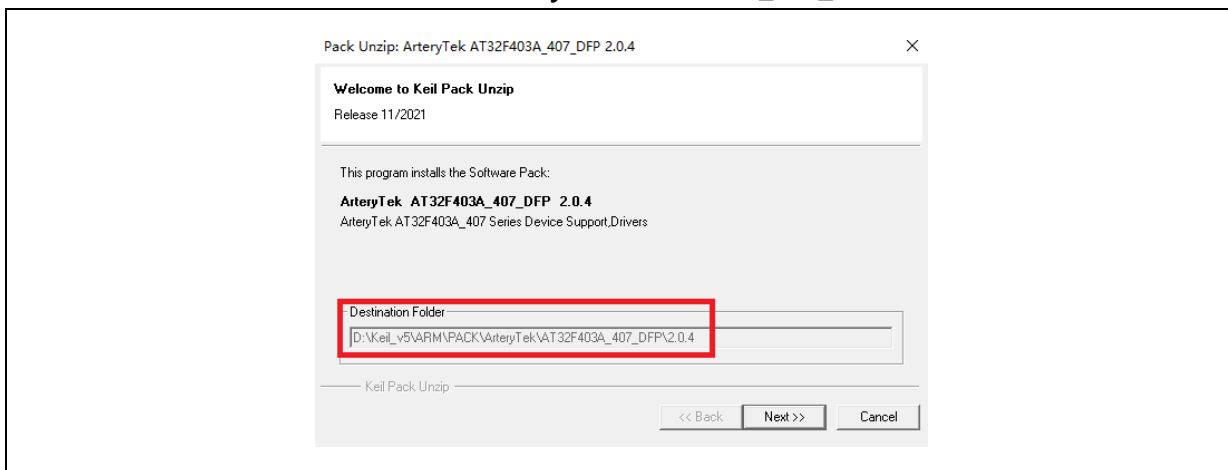
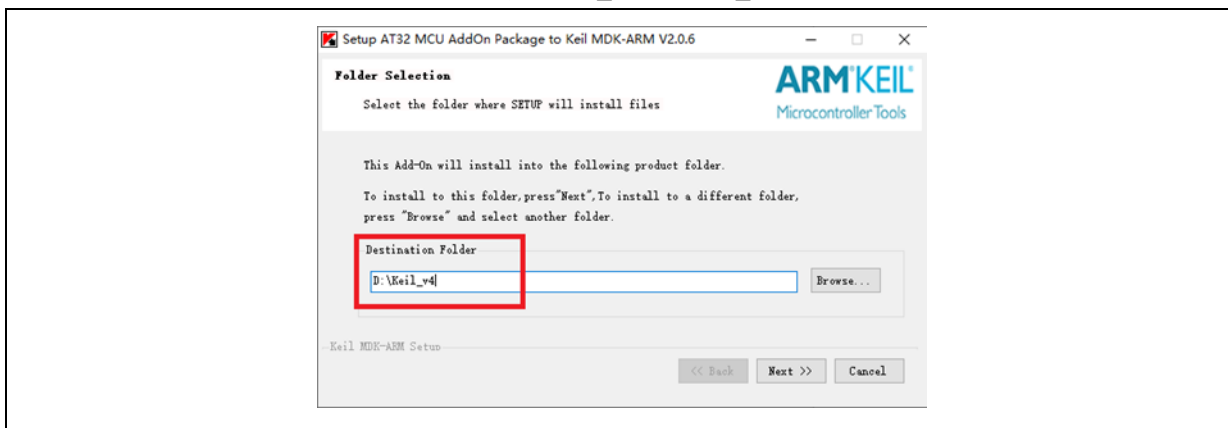
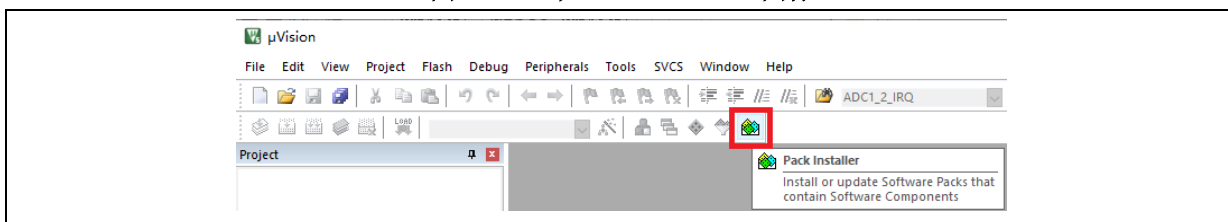


图 3. 安装 Keil4_AT32MCU_AddOn



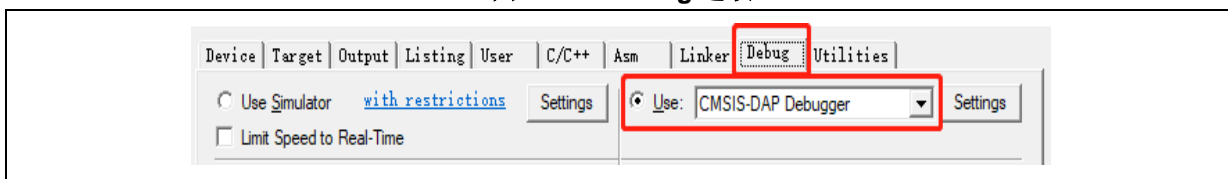
也可打开 keil，点击 Pack Installer 图标，在 Pack Installer 中点击左上角 file，选择 import，导入从雅特力官网下载好的对应 pack 包完成安装。

图 4. Keil 中 Pack Installer 图标



在 Keil 环境下使用 AT-Link，在 Debug 里选择 CMSIS-DAP 调试器。

图 5. Keil Debug 选项

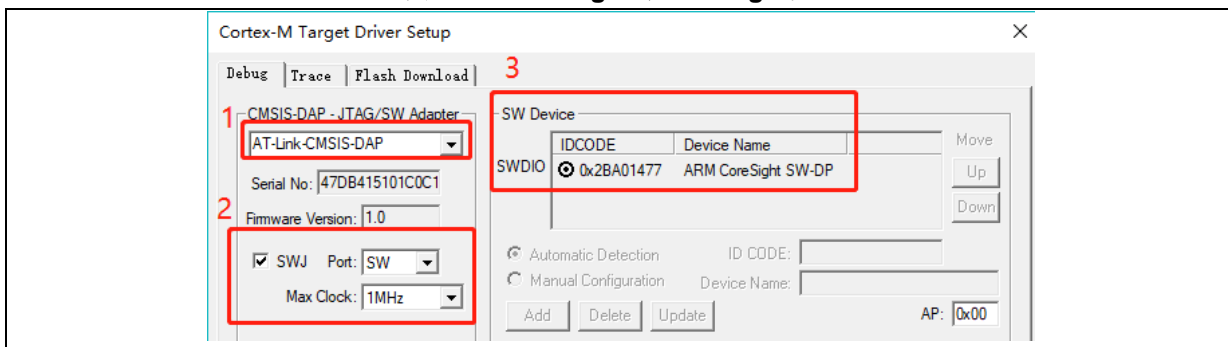


在 Debug 单击 Settings 进入 Cortex-M Target Driver Setup 界面如下图，

1. 先选择 AT-Link-CMSIS-DAP;

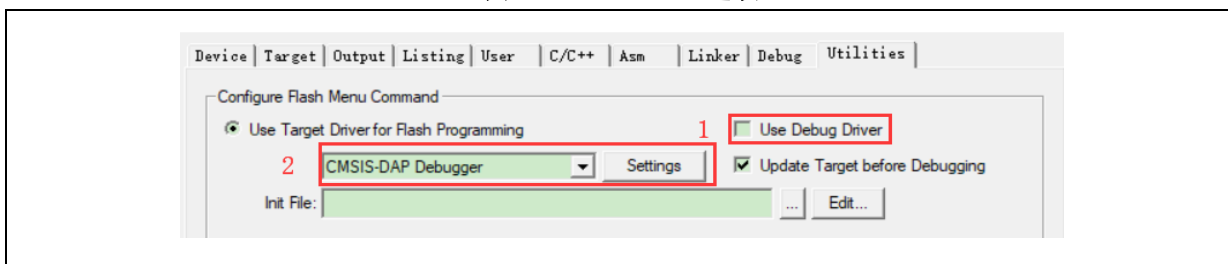
2. Port 选择 SW，再勾选 SWJ；
3. 确认已识别到 ARM SW-DP 调试模块。

图 6. Keil Debug 选项 Settings 设置



并且在 Utilities 里，先勾去下图标示的选项框 1，在选项框 2 下拉菜单选择 CMSIS-DAP Debugger，然后再勾选选项框 1（需要先取消再勾选）。

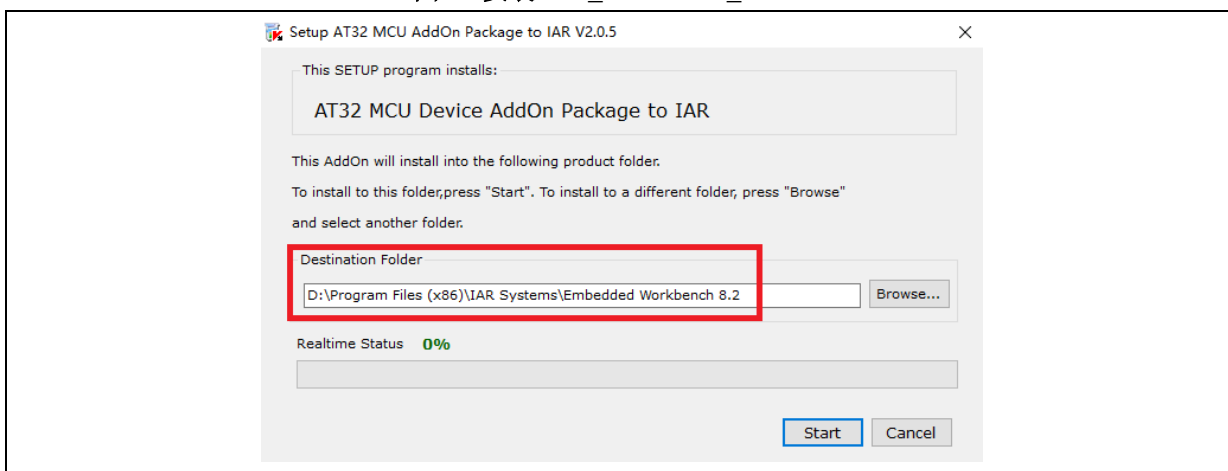
图 7. Keil Utilities 选项



② 对于 IAR 编译系统，建议 IAR7.0 或 IAR6.1 以上版本；

需要安装 Pack 包在 IAR 中增加 AT32 MCU 型号，Pack 包可以从雅特力官网下载。安装 IAR_AT32MCU_AddOn，默认情况下，安装时可以自动识别到 IAR 的安装路径，如果识别不到或者不正确，需要手动选择 IAR 的安装路径。

图 8. 安装 IAR_AT32MCU_AddOn



在 IAR 环境下使用 AT-Link，在 Debugger 里选择 CMSIS-DAP 调试器。

图 9. IAR Debug 选项

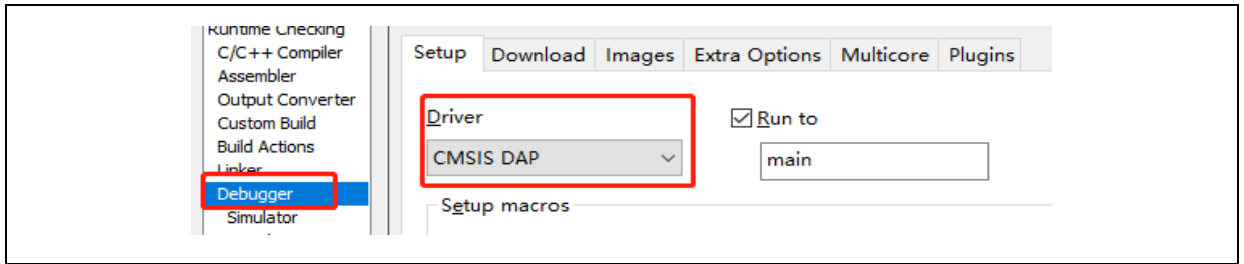
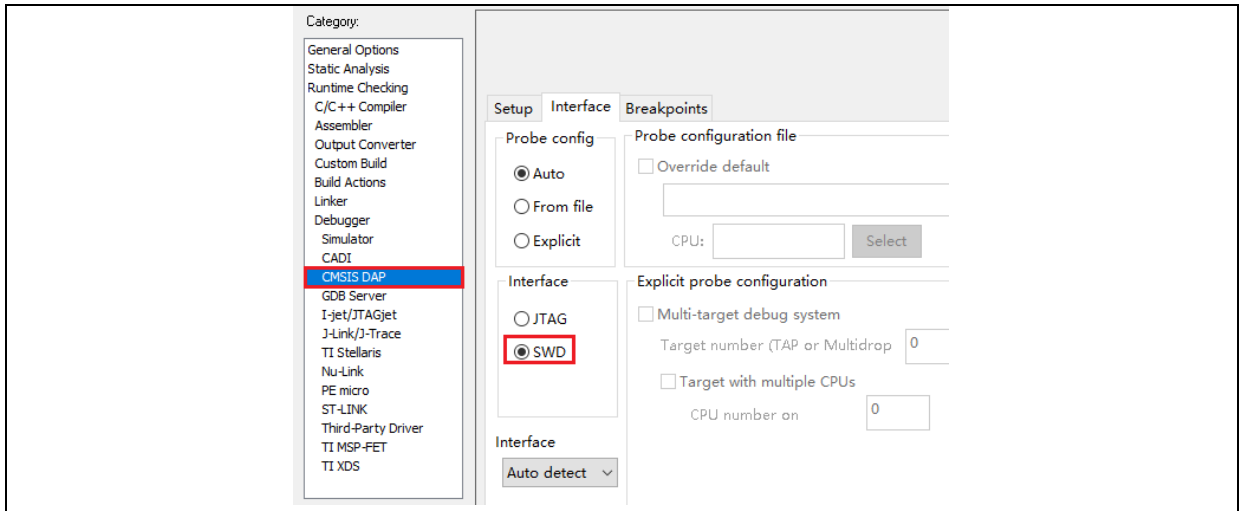


图 10. IAR CMSIS-DAP 选项



③ BSP 与 PACK 的选择(六种情形描述)。

表 1. 函数接口表

序号	使用什么 BSP/Pack	是否使用 AT32F403A _407 新功能	处理方法
1	AT32F403A_407 BSP/Pack	是	直接参考 BSP 相关例程
2	SXX32F103 BSP/Pack	否	直接下载程序运行
3	SXX32F103 寄存器操作	否	直接下载程序运行
4	SXX32F103 BSP + AT32 Pack	否	修改 FPU 设置
5	SXX32F103 寄存器操作	是	<ol style="list-style-type: none"> 如果系统主频设置 108MHz 以上，需要配置自动滑顺频率切换功能 结合《MG0007_从 SXX32F103 移植到 AT32F403A》《MG0009_从 SXX32F107 移植到 AT32F407》修改对应程序
6	SXX32F103	是	<ol style="list-style-type: none"> 如果系统主频设置 108MHz 以上，需要配置自动滑顺频率切

序号	使用什么 BSP/Pack	是否使用 AT32F403A_407 新功能	处理方法
	BSP/Pack		换功能 2. 结合《MG0007_从 SXX32F103 移植到 AT32F403A》 《MG0009_从 SXX32F107 移植到 AT32F407》修改对应程序

BSP 及 PACK 安装详细操作见 《AT32F403A_407 固件库 BSP&Pack 应用指南》，存放路径为官网下载的 BSP 解压后 AT32F403A_407_Firmware_Library_V2.x.x\document。

注意：关于以下两个 Flash 等待位，由于 AT32 与 SXX 的 Flash 机制不同，不需要设置以下两个位：

图 11. SXX Flash 等待位

```
/* Enable Prefetch Buffer */
FLASH->ACR |= FLASH_ACR_PRFTBE;
/* Flash 2 wait state */
FLASH->ACR &= (uint32_t)((uint32_t)~FLASH_ACR_LATENCY);
FLASH->ACR |= (uint32_t)FLASH_ACR_LATENCY_2;
```

此为Sxx在高速下，Flash取指令时间需要等待1-2 clock时钟，AT32F403A /AT32F407无此限制。

1.1.4 快速替代 SXX 流程

- 步骤一：比对外设规格、Flash 容量、SRAM 容量等，解焊 SXX32F103，换成 AT32F403A /AT32F407 对应型号；
- 步骤二：使用雅特力 ICP/ISP 或 KEIL/ IAR 下载 SXX32F103 HEX 文件或 BIN 文件；
- 步骤三：如果有需要，下载 SXX32F103 HEX 文件或 BIN 文件以外的资料或进行系统校正；
- 步骤四：查看程序能否正常运行；
- 步骤五：其他问题快速排查请参考《MG0007_从 SXX32F103 移植到 AT32F403A》《MG0009_从 SXX32F107 移植到 AT32F407》；
- 步骤六：如果经过上述步骤后程序仍无法正常运行，请参考本文件其他章节，或联络代理商及雅特力科技技术支持人员协助解决。

注意：由于AT32F403A /AT32F407采用灵活的内存扩展设计，内部闪存存储器存在非零等待区域，会导致有些 SXX32F103程序在AT32F403A /AT32F407上运行效能不佳。关于如何提高运行效能，请参考AT32提供的应用手册《AN0004_Performance_Optimization》。

1.2 AT32F403A /AT32F407 芯片的增强功能配置

1.2.1 PLL 时钟设置

1.2.1.1 PLL 大于 72MHz 设定

AT32F403A /AT32F407 内置的 PLL 最高可输出 240MHz 时钟，时钟高于 72MHz 时设定略有不同。须根据输出频率设定 PLLRANGE 寄存器，大于 72MHz 时 PLLRANGE=1，小于等于 72MHz 时

PLLRANGE=0。

使用 SXX32F103 BSP 时 PLL 设定程序范例：(HSE=8MHz,PLL=72MHz)

```
RCC->CFGR |= (uint32_t)(RCC_CFGR_PLLSRC_HSE | RCC_CFGR_PLLMULL9);
```

想要使用 SXX32F103 程序在 AT32F403A /AT32F407 输出大于 72MHz 时钟时，需要额外配置 PLLRANGE 位。

例如 PLL=240MHz 时，设置如下：

图 12. 使用 SXX32F103 程序在 AT32F403A /AT32F407 输出 240MHz 的时钟配置

```
#define RCC_CFG_PLLRANGE_GT72MHZ ((uint32_t)0x80000000)

/*!< When PLL frequency is greater than 72MHz */

RCC->CFGR |= (uint32_t)(RCC_CFGR_PLLSRC_HSE | RCC_CFGR_PLLMULL30 |
RCC_CFG_PLLRANGE_GT72MHZ);
```

使用 AT32F403A /AT32F407 BSP 时 PLL 设定程序范例：(HSE=8MHz)

图 13. AT32F403A /AT32F407 crm_pll_output_range 参数

```
typedef enum
{
    CRM_PLL_OUTPUT_RANGE_LE72MHZ= 0x00, /*!< pll clock output range less than or equal to 72mhz */
    CRM_PLL_OUTPUT_RANGE_GT72MHZ= 0x01 /*!< pll clock output range greater than 72mhz */
} crm_pll_output_range_type;
```

例如 PLL=72MHz 时，设置如下：

```
crm_pll_config(CRM_PLL_SOURCE_HEXT_DIV, CRM_PLL_MULT_18, CRM_PLL_OUTPUT_RANGE_LE72MHZ);
```

例如 PLL=240MHz 时，设置如下：

```
crm_pll_config(CRM_PLL_SOURCE_HEXT_DIV, CRM_PLL_MULT_60, CRM_PLL_OUTPUT_RANGE_GT72MHZ);
```

1.2.1.2 PLL 自动滑顺

当 AT32F403A /AT32F407 内置的 PLL 为 108MHz 以上时钟时，需要操作自动滑顺频率切换功能。

使用 SXX32F103 程序时，打开 system_Sxx32f10x.c 找到当前的系统时钟频率配置函数（需经过 1.2.1.1 PLL 配置），在函数 static void SetSysClockToxxM(void) 增加如下斜黑体部分：

图 14. PLL 自动滑顺频率切换配置

```
/* Wait till PLL is ready */
while((RCC->CR & RCC_CR_PLLRDY) == 0)
{
}
```

```
*((unsigned int *)0x40021054) |= (0x30); // 开启自动滑顺频率切换功能

/* Select PLL as system clock source */
RCC->CFGR &= (uint32_t)((uint32_t)~(RCC_CFGR_SW));
RCC->CFGR |= (uint32_t)RCC_CFGR_SW_PLL;
/* Wait till PLL is used as system clock source */
while ((RCC->CFGR & (uint32_t)RCC_CFGR_SWS) != (uint32_t)0x08)
{
}

*((unsigned int *)0x40021054) &=~ (0x30); //关闭自动滑顺频率切换功能
```

使用 AT32F403A /AT32F407 BSP 时 PLL 自动滑顺频率切换程序范例：

图 15. PLL 自动滑顺频率切换配置

```
/* enable auto step mode */
crm_auto_step_mode_enable(TRUE);
/* select pll as system clock source */
crm_sysclk_switch(CRM_SCLK_PLL);
/* wait till pll is used as system clock source */
while(crm_sysclk_switch_status_get() != CRM_SCLK_PLL)
{
}
/* disable auto step mode */
crm_auto_step_mode_enable(FALSE);
/* update system_core_clock global variable */
system_core_clock_update();
```

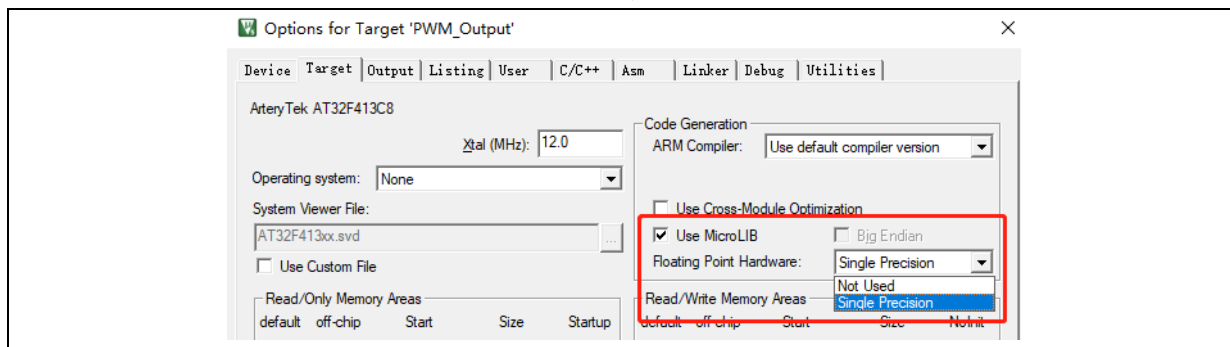
注意：如果开启了自动滑顺频率切换功能，在时钟切换后必须关闭自动滑顺频率切换功能，开启及关闭务必配对使用。

1.2.2 如何打开 FPU 功能（硬件浮点运算单元）

Keil 环境中分为以下两种情况：

- ① 使用 AT32F403A_407 BSP/Pack 或者修改过的 SXX BSP/Pack 直接修改 Floating Point Hardware。

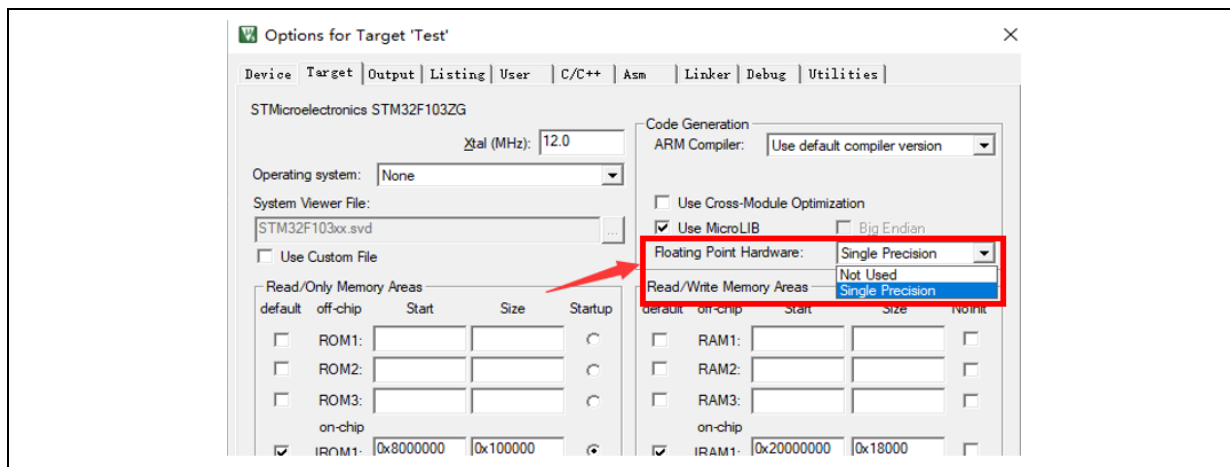
图 16. Keil 环境中选择开启 FPU



② 由于 SXX32F10X 系列是不支持 FPU 功能，客户如需在之前 SXX 库下已开发的工程内打开 FPU 功能需要做如下处理：

- 首先需参考《AT32F403A_407 固件库 BSP&Pack 应用指南》安装 Keil 环境 PACK，修改相关头文件；
- 需选择 AT 对应型号，然后在 Options--Target 中如下图选择：

图 17. Keil 环境中选择开启 FPU



- 最后需在 system_stm32f10x.c 的 SystemInit 函数开头添加以下配置，并将 cm4.h 添加入工程中。

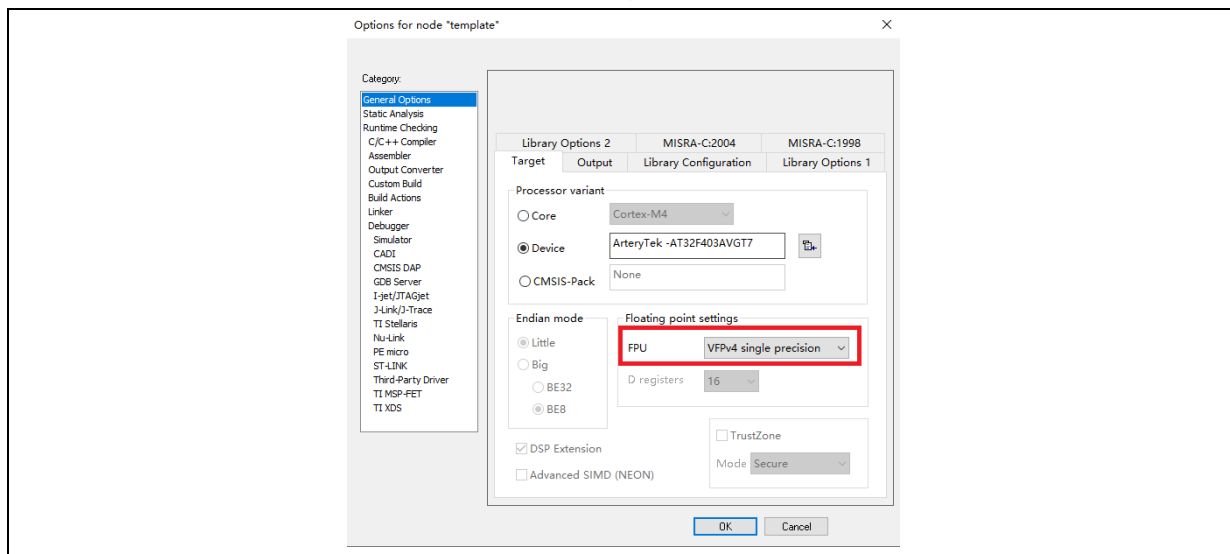
图 18. Keil 环境中增加开启 FPU 的代码

```
/* Enable FPU*/
#ifdef (__FPU_USED) && (__FPU_USED == 1U)
    SCB->CPACR |= ((3U << 10U * 2U) |           /* set CP10 Full Access */
                  (3U << 11U * 2U) );          /* set CP11 Full Access */
#endif
```

IAR 环境中同样分为以下两种情况：

- ① 使用 AT32F403A_407 BSP/Pack 或者修改过的 SXX BSP/Pack 直接修改 Floating Point Hardware。

图 19. IAR 环境中选择开启 FPU



② 由于 SXX32F10X 系列是不支持 FPU 功能，客户如需在之前 SXX 库下已开发的工程内打开 FPU 功能需要做如下处理：

- 首先需参考《AT32F403A_407 固件库 BSP&Pack 应用指南》安装 IAR 环境 PACK，修改相关头文件；
- 需选择 AT 对应型号，然后在 General Options--Target 中选择；
- 最后需在 system_stm32f10x.c 的 SystemInit 函数开头添加以下配置，并将 cm4.h 添加入工程中。

图 20. IAR 环境中增加开启 FPU 的代码

```
/* Enable FPU*/

#ifdef (__FPU_USED) && (__FPU_USED == 1U)

    SCB->CPACR |= ((3U << 10U * 2U) |           /* set CP10 Full Access */
                  (3U << 11U * 2U) );          /* set CP11 Full Access */

#endif
```

1.2.3 AT32F403A /AT32F407 零等待/非零等待 Flash 和内置 SRAM 大小选择配置说明

通过用户系统数据配置支持内部闪存存储器和 SRAM 分配使用。

以 AT32F403AVGT7 为例，内部闪存存储器和 SRAM 可以设置为以下两种配置：

- ZW: 256KB, NZW: 768 KB, SRAM: 96 KB (出厂默认)；
- ZW: 128KB, NZW: 896 KB, SRAM: 224 KB。

内核读取存放在零等待的 Flash 的指令码没有任何延迟，不会因为 CPU 主频太快，Flash 的速度跟不上而要插入等待时钟。假设系统时钟 240MHz，AT32F403A 零等待有 256KB 大小，512KB bin 文件的前面 256KB 可以 240MHz 速率执行，后面 256KB bin 文件存放在非零等待区，执行速率约 96MHz，仍然比 SXX32F10X 最高主频 72MHz 更快。非零等待的执行速率是零等待的 0.4 倍。

内置 SRAM 96KB(默认)/224KB 大小可以通过以下任何方式选择：

AT32F403A SRAM 大小设置涉及 FLASH 用户系统数据说明，通过配置 EOPB0 选择，地址是：0x1FFF_F810。EOPB0=0xFF 表示片上 SRAM 为 96KB，EOPB0=0xFE 表示片上 SRAM 为 224KB。使能 EOPB0 有效务必要掉电或 RESET 一次。

① 使用 ICP/ISP 方式

■ Artery ICP Programmer 工具

设备操作--用户系统数据--选择 96KB/224KB--应用到设备。

图 21. ICP 工具编辑用户系统数据选择 SRAM 大小

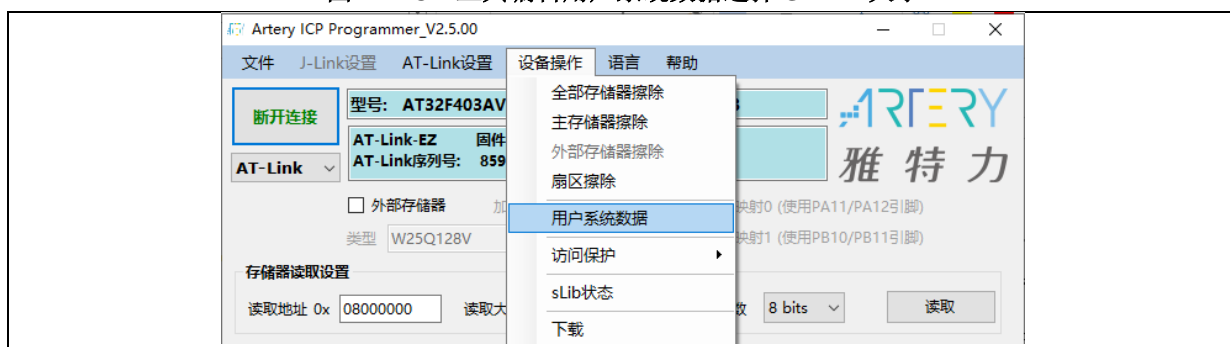
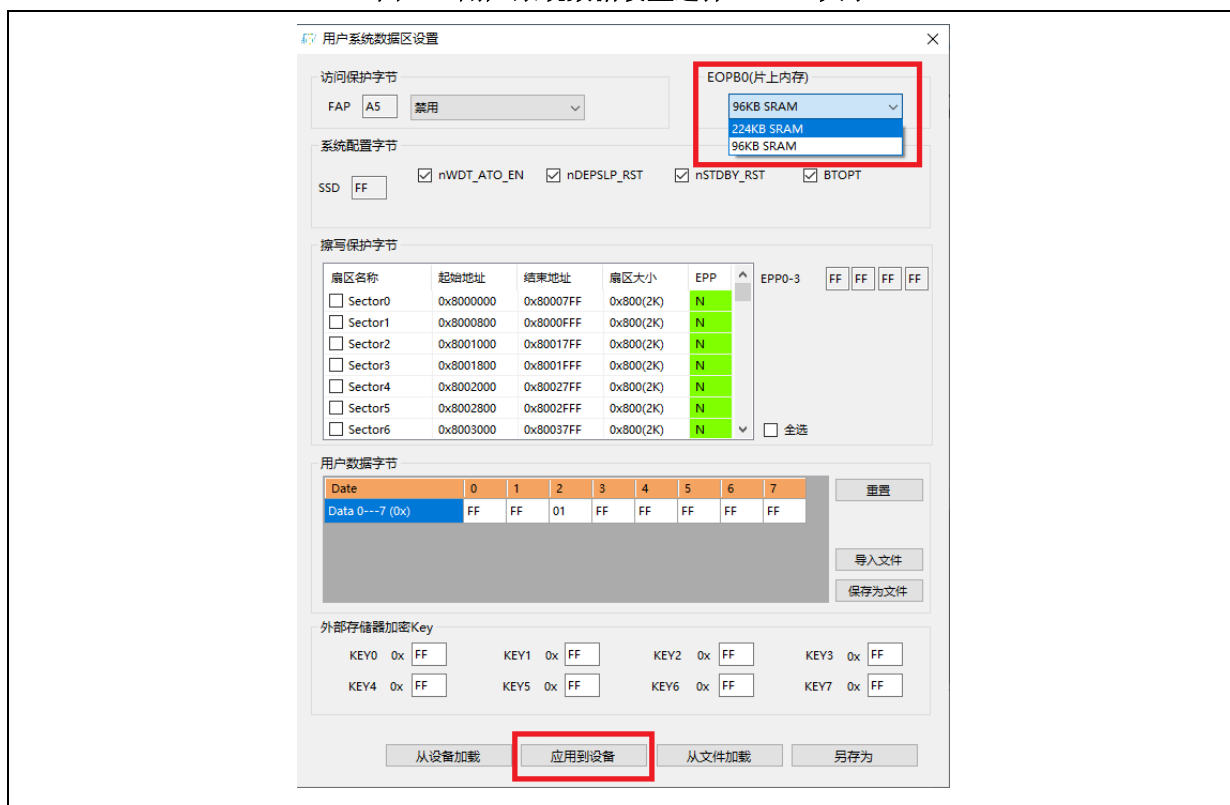


图 22. 用户系统数据设置选择 SRAM 大小



■ Artery ISP Programmer 工具

BOOT0=1, BOOT1=0, 点击下一步，直到进入最后界面，选择编辑用户系统数据，点击下一步，后续操作与 ICP 相同，选择 96KB/224KB，应用到设备。

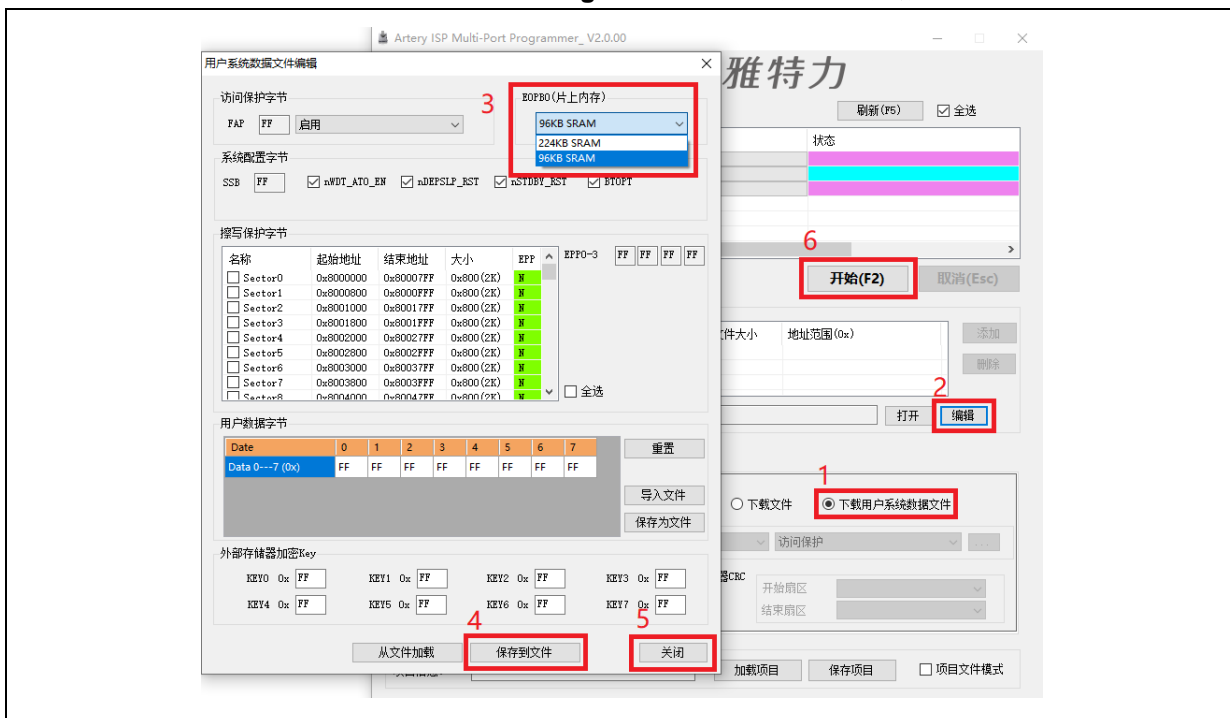
图 23. ISP 工具编辑用户系统数据选择 SRAM 大小



■ Artery ISP Multi-Port Programmer 工具

下载用户系统数据文件--编辑--选择 **96KB/224KB** --保存到文件（新建一个用户系统数据烧录档）--关闭--开始，或者下载用户系统数据文件--打开（已经保存的用户系统数据烧录档）--开始。

图 24. ISP Multi-Port Programmer 工具用户系统数据设置



② 使用脱机烧录器方式

可以使用轩微，Maxwiz 脱机烧录器，设置 OPTION 部分后，保存为 OPTION 文件。客户每次烧录时选择此文件后，直接烧录进 AT32F403A /AT32F407 MCU 来选择 SRAM 大小。

③ 客户也可以使用在 Bootloader program (IAP) 内修改 SRAM 的大小。使用 SXX32F103 程序时，范例如下：

图 25. 定义 Extend_SRAM(void)函数修改 SRAM 大小

```
#define SRAM_96k  0xFF
#define SRAM_224k 0xFE

static uint32_t F_EOPB0;
F_EOPB0=(uint32_t*)(0x1FFFFF810);

void Extend_SRAM(void)
{
    if((F_EOPB0 & 0xFF) != 0xFE) // check if RAM has been set to 224K, if not, change EOPB0
    {
        FLASH_Unlock();
        FLASH_EraseOptionBytes();
        FLASH_ProgramOptionByteData(0x1FFFFF810, SRAM_224k);
        ...                //设置其他用户系统数据
        FLASH_Lock();
        NVIC_SystemReset();
    }
}
```

使用 AT32F403A/ AT32F407 程序时，范例如下：

图 26. 定义 Extend_SRAM(void)函数修改 SRAM 大小

```
#define SRAM_96k  0xFF
#define SRAM_224k 0xFE

static uint32_t f_eopb0;
f_eopb0=(uint32_t*)(0x1FFFFF810);

void Extend_SRAM(void)
{
    if((f_eopb0 & 0xFF) != 0xFE) // check if RAM has been set to 224K, if not, change EOPB0
    {
        flash_unlock();
        flash_user_system_data_erase();
        flash_user_system_data_program(0x1FFFFF810, SRAM_224k);
        ...                //设置其他用户系统数据
        flash_lock();
    }
}
```

```

        nvic_system_reset();
    }
}

```

④ 启动文件中修改 SRAM 的大小。

在运行启动文件时会加载 SRAM，如果程序没有 IAP，而应用程序使用的 SRAM 大于 96KB，那么加载失败就会进入 hardfault，导致应用程序运行不起来。所以可以在启动文件中加载 SRAM 之前将 SRAM 大小配置为 224KB。

在 Keil 编译环境启动文件中增加如下红色部分代码：

图 27. Keil 启动文件中修改 SRAM 大小

```

; Reset handler
Reset_Handler PROC
    EXPORT Reset_Handler                [WEAK]
    IMPORT __main
    IMPORT SystemInit

    IMPORT Extend_SRAM
    MOV32    R0, #0x20001000
    MOV      SP, R0
    LDR      R0, =Extend_SRAM
    BLX      R0
    MOV32    R0, #0x08000000
    LDR      SP, [R0]

    LDR      R0, =SystemInit
    BLX      R0
    LDR      R0, =__main
    BX       R0
ENDP

```

在 IAR 编译环境启动文件中增加如下红色字体

图 28. IAR 启动文件中修改 SRAM 大小

```

; Default interrupt handlers.
    THUMB

    PUBWEAK Reset_Handler
    SECTION .text:CODE:REORDER:NOROOT(2)

    EXTERN  Extend_SRAM
Reset_Handler
    MOV32    R0, #0x20001000
    MOV      SP, R0

```

```
LDR    R0,=Extend_SRAM
BLX    R0
MOV32  R0,#0x08000000
LDR    SP,[R0]

LDR    R0,=SystemInit
BLX    R0
LDR    R0,=__iar_program_start
BX     R0
```

同时需要在应用程序中添加声明和定义 `Extend_SRAM` 函数，参考 1.2.3 中③的写法（图 25/图 26），定义 `Extend_SRAM(void)`函数修改 SRAM 大小。

⑤ 不推荐使用 APP 应用程序修改 SRAM 大小；如果 APP 使用的 SRAM 空间大于修改后的 SRAM 空间，程序会进入 Hardfault。

1.2.4 加密方式(访问保护,擦写保护,外部 Flash 的加密)

1.2.4.1 访问保护

访问保护即大家通常说的“加密”，作用于整个 Flash 存储区域。一旦设置了 Flash 的访问保护，内置的 Flash 存储区只能通过程序的正常执行才能读出，而不能通过 JTAG 或者 SWD 读出，当使用 ICP/ISP 工具解除访问保护时，芯片会对 FLASH 进行擦除操作。

可用 ICP/ISP 工具对 IC 进行访问保护与解除访问保护操作，如下：

■ Artery ICP Programmer 工具

启用访问保护：设备操作--访问保护--启用访问保护。

解除访问保护：设备操作--访问保护--解除访问保护。

■ Artery ISP Programmer 工具

启用访问保护：使能/除能保护、启用访问保护--下一步--是，即可将程序加密。

解除访问保护：使能/除能保护、禁用访问保护--下一步--是，即可将 Flash 解除加密。

图 29. ISP 工具启用访问保护



图 30. ISP 工具解除访问保护



■ Artery ISP Multi-Port Programmer 工具

启用访问保护：使能/除能保护、启用访问保护--开始--是，即可将程序加密。

解除访问保护：使能/除能保护、禁用访问保护--开始--是，即可将 Flash 解除加密。

注意：设置了访问保护不能通过擦除操作来解除访问保护。

1.2.4.2 擦写保护

写保护作用于整个 Flash 存储区域或者 Flash 存储区域的某些页。一旦设置了 Flash 的写保护，内置

的 Flash 存储区就不能通过任何方式写入。

可用 ICP/ISP 工具对 IC 进行访问保护与解除访问保护操作，如下：

■ Artery ICP Programmer 工具

启用擦写保护：设备操作--用户系统数据--勾选擦写保护字节扇区--应用到设备。

解除擦写保护：设备操作--用户系统数据--勾除擦写保护字节扇区--应用到设备。

图 31. ICP 工具启用擦写保护

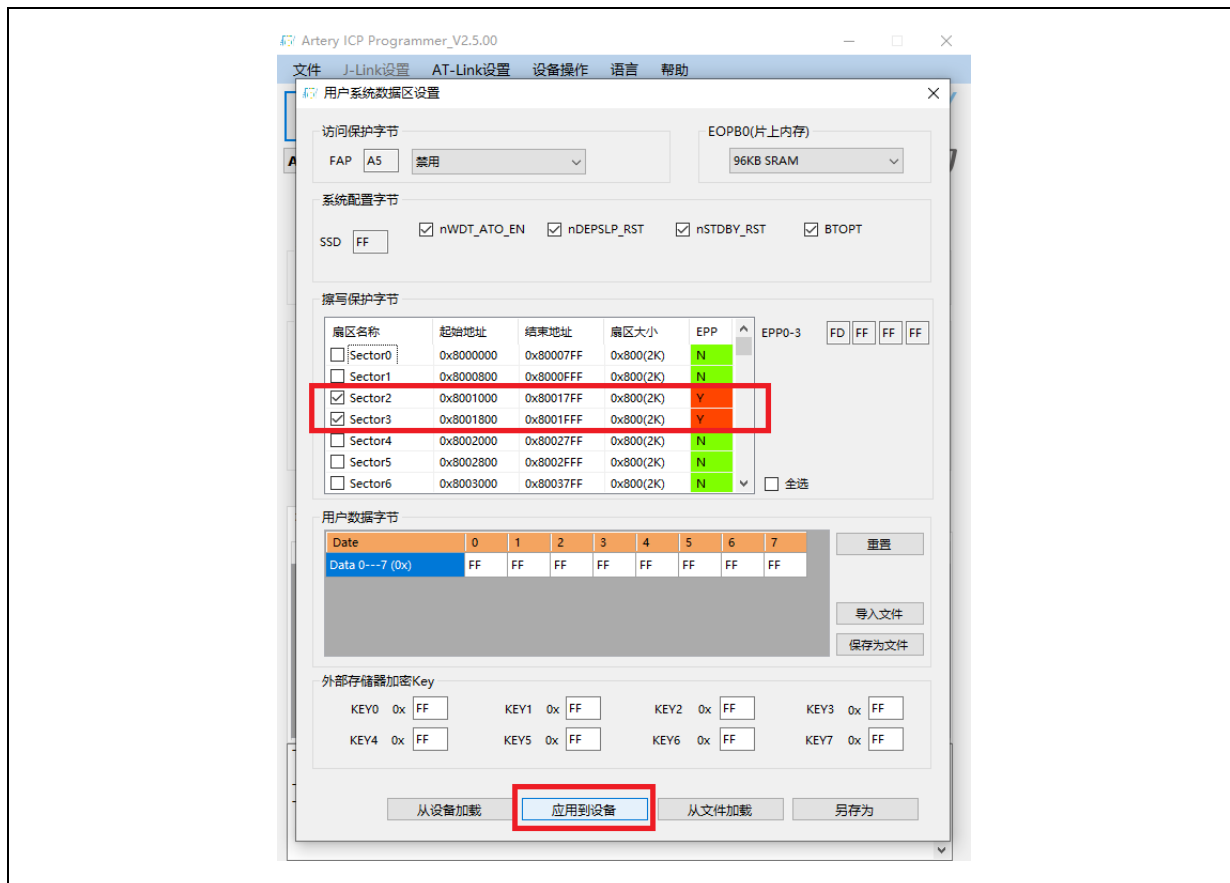
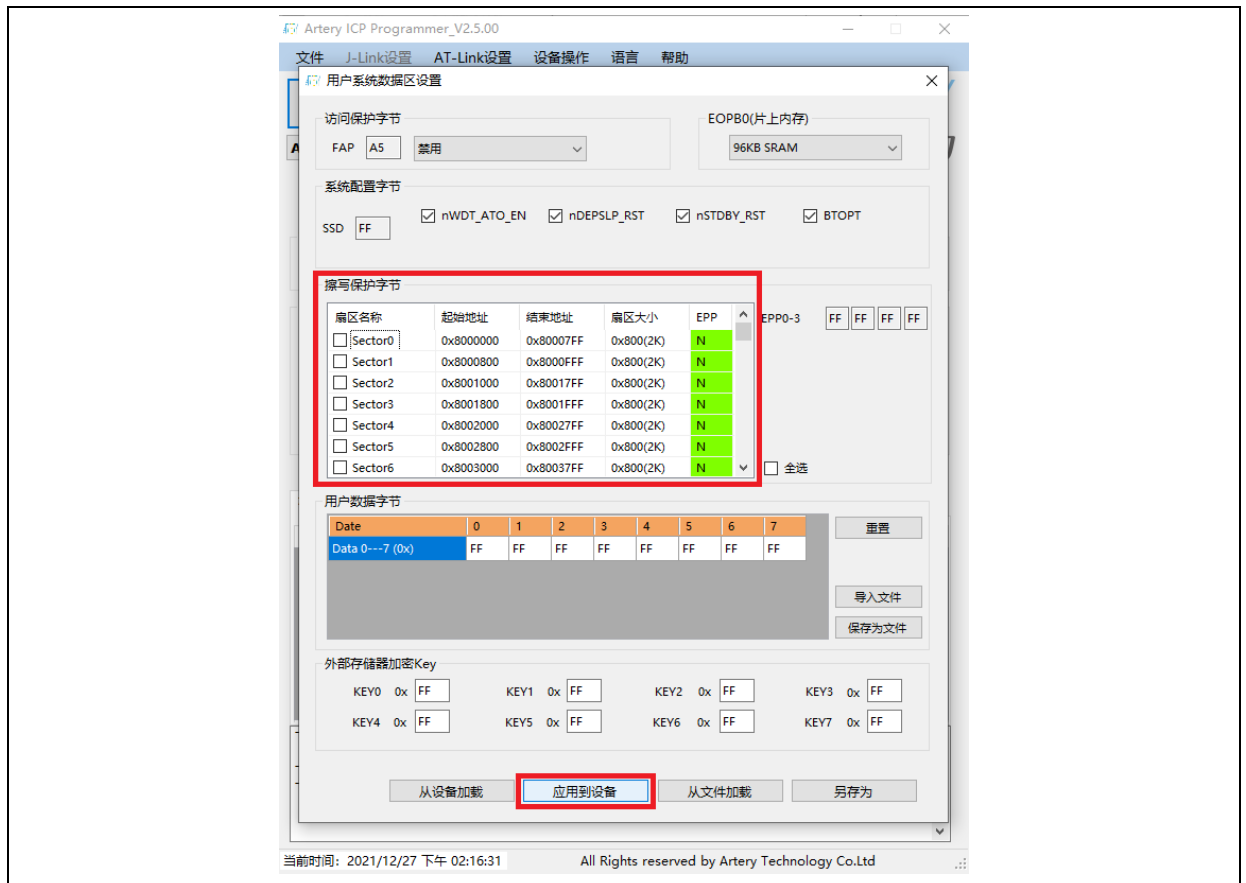


图 32. ICP 工具解除擦写保护



■ Artery ISP Programmer 工具

启用擦写保护：使能/除能保护、启用擦写保护--下一步--是，即可启用写保护。

解除擦写保护：使能/除能保护、禁用擦写保护--下一步--是，即可解除写保护。

■ Artery ISP Multi-Port Programmer 工具

启用擦写保护：使能/除能保护、启用擦写保护--开始--是，即可启用写保护。

解除擦写保护：使能/除能保护、禁用擦写保护--开始--是，即可解除写保护。

注意：设置了擦写保护不能通过擦除操作来解除擦写保护。

1.2.4.3 外部 Flash 的加密(下载和读取外部存储器加密范围内数据的加密)

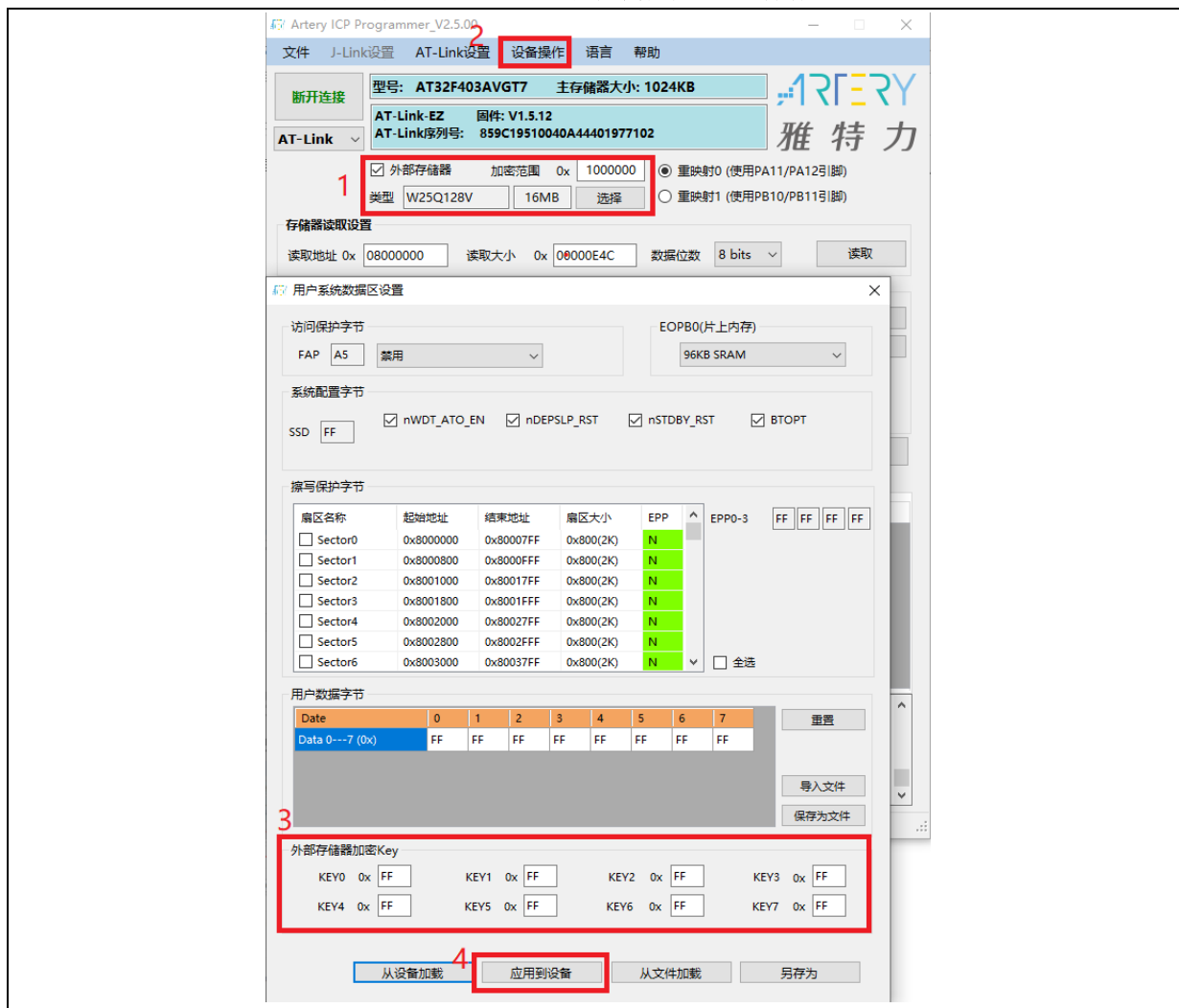
对外部 Flash 加密，需要先进行如下设置加密范围和加密密码后再烧录用户程序，然后开启访问保护。加密范围指的是从 0x08400000 开始需要加密的空间大小。外部存储器加密 Key，如果全部是 0xFF 或者 0x00 则不加密，否则进行加密。解除访问保护会将外部存储器加密 Key 设置为全 0xFF。

以下为 ICP/ISP 工具加密外部存储器的操作说明：

■ Artery ICP Programmer 工具

勾选外部存储器--选择外部存储器类型--设置外部存储器加密范围--设备操作--用户系统数据--修改外部存储器加密 Key--应用到设备。

图 33. ICP 工具对外部存储器加密操作



■ Artery ISP Programmer 工具

编辑用户系统数据--下一步--修改外部存储器加密 Key--应用到设备。

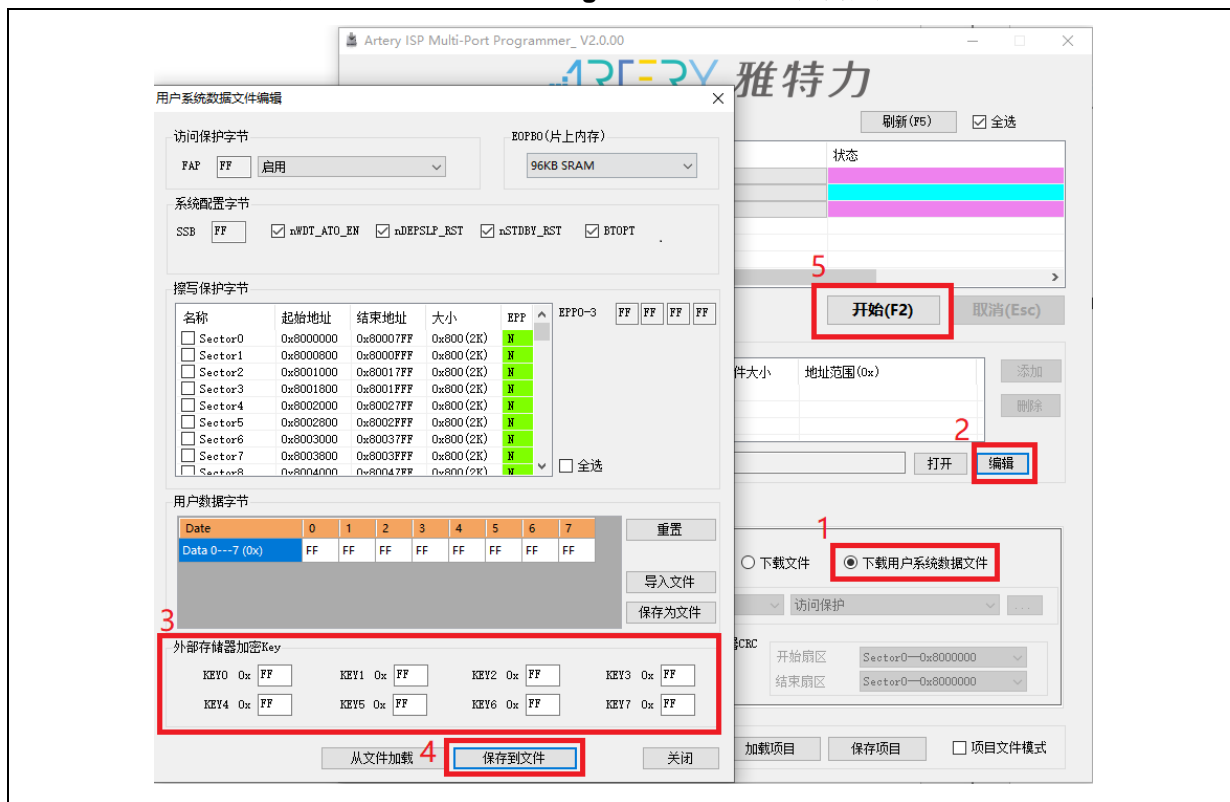
图 34. ISP 工具对外部存储器加密操作



■ Artery ISP Multi-Port Programmer 工具

下载用户系统数据文件--编辑--修改外部存储器加密 Key--保存到文件--开始。

图 35. ISP Multi-Port Programmer 工具对外部存储器加密



1.2.5 在程序中区分 AT32 与其他 IC 方法

■ 读取 Cortex-M 系列 CPU ID 号区分，此方式可以区分出 M0,M3,M4 内核

图 36. 读取 Cortex 型号

```
cortex_id = *(uint32_t *)0xE000ED00; // 读取 Cortex 型号
if((cortex_id == 0x410FC240) || (cortex_id == 0x410FC241))
{
    printf("This chip is Cortex-M4F.\r\n");
}
else
{
    printf("This chip is Other Device.\r\n");
}
```

■ 读取 PID,UID 方式区分

图 37. 读取 PID,UID

```
/* 获取 AT32 MCU 的 PID/UID 的基地址*/
#define DEVICE_ID_ADDR1 0x1FFFF7F3 //定义 Artery MCU 的项目型号
#define DEVICE_ID_ADDR2 0xE0042000 //定义 MCU 的设备型号

/* 用于存放 ID */
uint8_t ID[5] = {0};
```



```

/* AT32F403A MCU type table */
const uint64_t AT32_MCU_ID_TABLE[] =
{
    0x00000000270050242,      //AT32F403ARCT7    256KB    LQFP64
    0x000000002700502CA,      //AT32F403ARET7    512KB    LQFP64
    ...
};

/* 获取 PID/UID */
ID[0] = *(int*)DEVICE_ID_ADDR1;
ID[1] = *(int*)(DEVICE_ID_ADDR2+3);
ID[2] = *(int*)(DEVICE_ID_ADDR2+2);
ID[3] = *(int*)(DEVICE_ID_ADDR2+1);
ID[4] = *(int*)(DEVICE_ID_ADDR2+0);

/* 组合 PID/UID */
AT_device_id =
((uint64_t)ID[0]<<32)|((uint64_t)ID[1]<<24)|((uint64_t)ID[2]<<16)|((uint64_t)ID[3]<<8)|((uint64_t)ID[4]<<0);

/* 判断 AT32 MCU */
for(i=0;i<sizeof(AT32_MCU_ID_TABLE)/sizeof(AT32_MCU_ID_TABLE[0]);i++)
{
    if(AT_device_id == AT32_MCU_ID_TABLE[i])
    {
        printf("This chip is AT32F4xx.\r\n");
    }
    else
    {
        printf("This chip is Other Device.\r\n");
    }
}

```

说明: 在 AT32F4xx 微控制器内部有多个 ID 编码, 将获取到的 ID 信息组装成一个 64bit 的数据, 就可以区分出 MCU 是哪一种型号。更多信息请参考《AN0016_Recognize_AT32_MCU》。

2 下载编译过程常见问题

2.1 程序启动进入 Hard Fault Handler

- 使用 SRAM 超过用户系统数据设置的 SRAM 空间。
请参考 1.2.3 使用 ICP/ISP 或第三方烧录器开启更大 SRAM 空间后烧录程序。
- 在 Keil 或 IAR 上开启了 single precision 功能，在 code 中并没有开启 M4 内核 FPU 寄存器。在 code 中开启 FPU 功能：

图 38. 增加开启 FPU 的代码

```
void SystemInit(void)
{
    /* Enable FPU*/

    #if defined (__FPU_USED) && (__FPU_USED == 1U)

        SCB->CPACR |= ((3U << 10U * 2U) |           /* set CP10 Full Access */
                       (3U << 11U * 2U) );          /* set CP11 Full Access */

    #endif
}
```

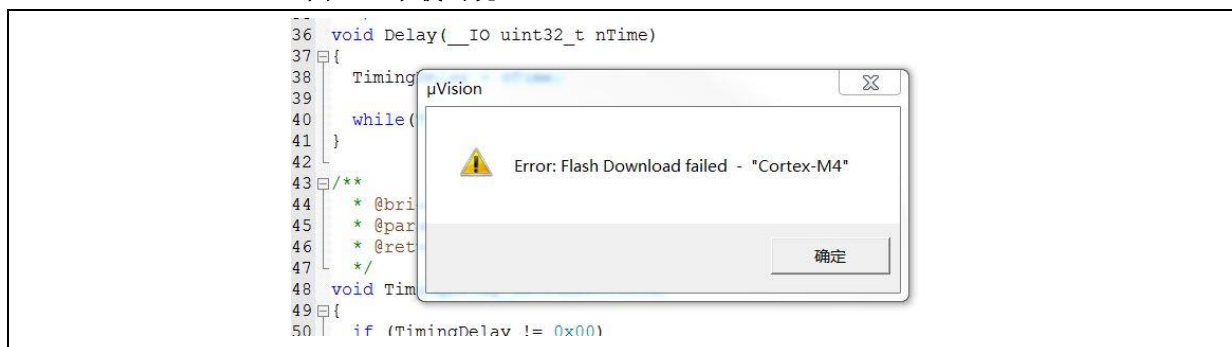
- 访问数据越界。
找到程序中访问越界的问题点，并修改它到正常数据区域内。
- 系统时钟设置超出规格。

2.2 程序下载过程出问题

2.2.1 显示 Error: Flash Download failed-“Cortex-M4”问题

在 KEIL 仿真或下载时弹出：

图 39. 下载出现 Flash Download failed-“Cortex- 4”



出现弹窗的原因可能是以下几种：

- 开启了访问保护，先取消 MCU 访问保护再下载；
- 选错了或者没有选择加载 Flash 文件算法，在 Flash Download 处选择添加正确的 Flash 文件算法；
- BOOT0、BOOT1 选择错误，BOOT0、BOOT1 管脚电平须分别设置为 BOOT0=0、BOOT1=0，使 MCU 从主闪存存储器启动；

- J-Link 驱动版本太低，建议 6.20C 以上版本；
- 在程序中将 JTAG/SWD PIN disable，解决方法参考“2.2.5 AT32 恢复下载”。

2.2.2 显示 No Debug Unit Device found 问题

- 下载端口被占用，比如 ICP 正在连接目标设备。
- JTAG/SWD 连线错误，或没有连接。

2.2.3 显示 RDDI-DAP Error 问题

- 在程序中将 JTAG/SWD PIN disable，解决方法参考“2.2.5 AT32 恢复下载”。

2.2.4 ISP 串口下载时卡死问题

使用 ISP 串口下载时，偶尔会卡死，卡死之后电脑无法释放串口。
建议处理方式：

- 检查电源是否稳定；
- 更换质量更好的 USB 转串口工具，如 CH340 芯片等。

2.2.5 AT32 恢复下载

在使用 AT32F403A /AT32F407 时，用户可能在以下操作后无法再次下载程序：

- 在程序中将JTAG/SWD PIN disable后，无法下载程序并且找不到JTAG/SWD device；
- 进入Standby mode后，无法下载程序并且找不到JTAG/SWD device。

这里提供 KEIL 和 IAR 环境下的解决方法如下：

- 方法1 使用ARTERY提供的ConfigureJLink.exe工具；
- 方法2 通过切换boot模式；切换boot模式到Boot[1:0]=01b或Boot[1:0]=11b，再按下复位键，即可恢复下载。同理，ISP下载也可恢复下载。
- 方法3 通过ICP tool加AT-Link方法。

AT-Link 专为 AT32 设计，因此使用 ICP 加 AT-Link 可恢复下载。考虑到方法 2 和方法 3 需要相关的电路（Boot PIN）或设备（AT-Link）支持，本文重点介绍方法 1。

2.2.5.1 KEIL 环境下的解决方法

使用 ARTERY 提供的 ConfigureJLink.exe 工具。

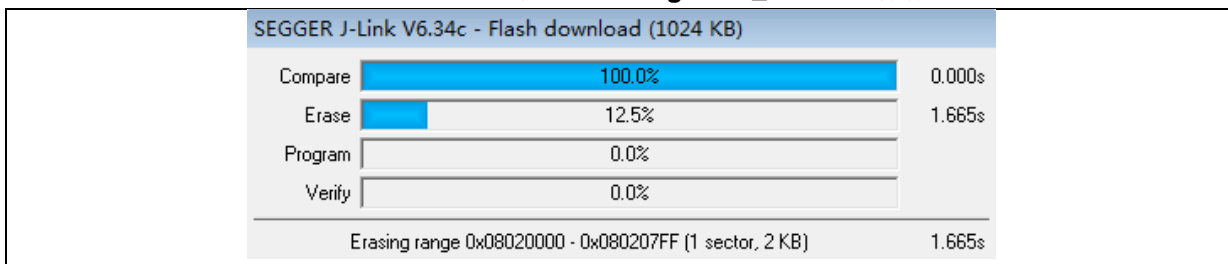
步骤如下：

- 将 ConfigJLink_V1.0.0.exe 工具放入工程文件 (*.uvprojx) 所在目录。
- 双击 ConfigJLink_V1.0.0.exe，将会弹出如下对话框。
- 勾选同意后再点击确认，待弹出的如下擦除进度条执行完毕后即可正常下载程序。

图 40. KEIL 环境中工具 ConfigJLink_V1.0.0 的操作



图 41. KEIL 环境中工具 ConfigJLink_V1.0.0 的操作



Note1: 使用该工具时需要确保 SEGGER J-Link interface DLL 不低于 V6.14。

Note2: 若每次下载的程序都会 disable JTAG/SWD PIN 时，每次下载程序前都需要执行一遍上述步骤。

Note3: 若每次下载的程序都会进入 Standby mode 时，每次芯片上电时都需要执行一遍上述步骤。

Note4: Keil 环境下，AT32F403A /AT32F407 芯片，进入 Standby mode 后，使用该工具解法无效。

2.2.5.2 IAR 环境下的解决方法

使用 ARTERY 提供的 ConfigJLink_V1.0.0.exe 工具

步骤如下：

- 将 ConfigJLink_V1.0.0.exe 工具放入工程目录下的 settings 文件夹。
- 双击 ConfigJLink_V1.0.0.exe，将会弹出如下对话框。
- 勾选同意后再点击确认，待弹出的如下擦除进度条执行完毕后即可正常下载程序。

图 42. IAR 环境中工具 ConfigJLink_V1.0.0 的操作

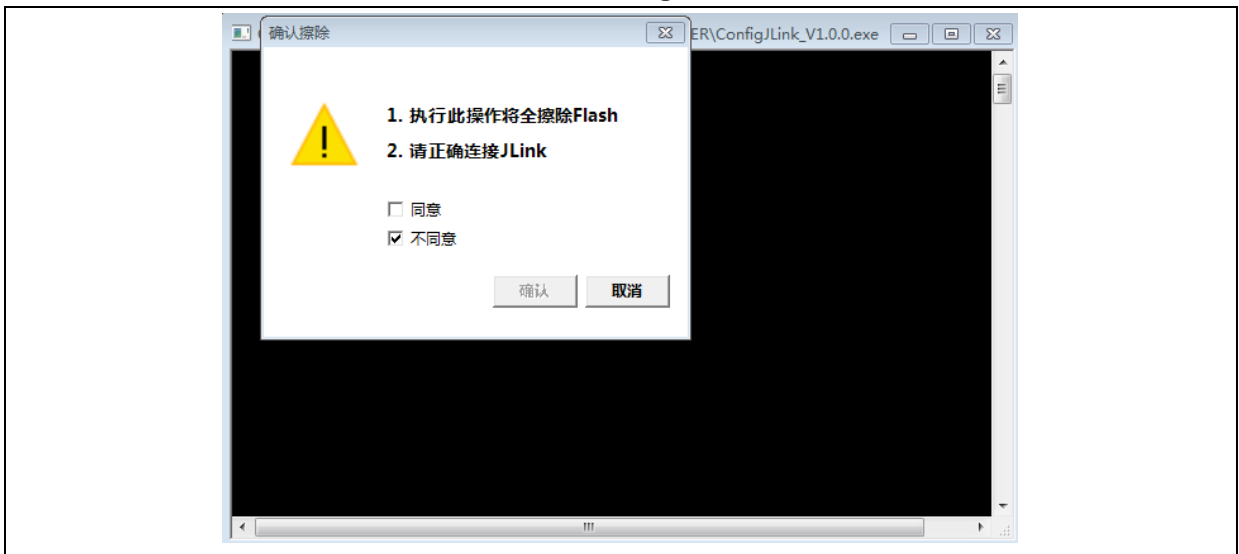
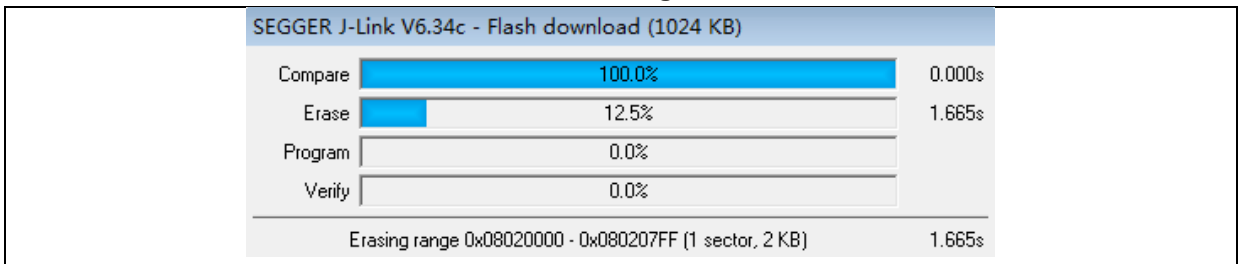


图 43. IAR 环境中工具 ConfigJLink_V1.0.0 的操作



Note1: 使用该工具时需要确保 SEGGER J-Link interface DLL 不低于 V6.14。

Note2: 若每次下载的程序都会 disable JTAG/SWD PIN 时，每次下载程序前都需要执行一遍上述步骤。

Note3: 若每次下载的程序都会进入 Standby mode 时，每次芯片上电时都需要执行一遍上述步骤。

2.3 升级轩微烧录器固件

升级方法步骤：

- 打开最新版本超级脱机编程器上位机软件，需要计算机联网，给设备断电(即拔掉 USB 线)；
- 按下蓝色按键给设备上电，待绿色指示灯常亮后松开按键(这时应是白屏或黑屏状态)；
- 上位机软件会弹出升级窗口，点窗口的确定自动升级固件，无需任何操作（耐心等待提示升级成功大约一分钟)；
- 升级成功提示固件更新时间。在烧录器显示屏的右下角显示更新后的固件版本号，如下图。

图 44. 轩微烧录器固件显示图片



注意：进入显示屏上的固件升级是升级警告，并不是升级操作。

3 安全库区 sLib(Security Library)

3.1 概述

目前越来越多的微控制器(MCU)应用需要使用到复杂的算法及中间件解决方案(middleware solution)，因此，如何保护软件方案商开发出来的核心算法等知识产权代码(IP-Code)，便成为微控制器应用中一项很重要的课题。

应这一重要需求，AT32F403A /AT32F407 系列提供了安全库区(sLib)的功能，以防止重要的 IP-Code 被终端用户的程序做修改或读取，进而达到保护的目的。

3.2 应用原理

- 设定以密码保护主闪存中指定范围的程序区(即安全库区)，软件方案商可将核心算法存放到此区域，以达到保护的功能，其余空白程序区可以提供给终端商客户进行二次开发。
- 安全库区划分为指令安全库区(sLib_Code)及数据安全库区(sLib_Data)，并可选择部分或是整个安全库区存放指令，但不支持整个安全库区存放数据。
- 指令安全库区(sLib_Code)内的程序代码仅能被MCU透过I-Code总线抓取指令(仅能被执行)，不能透过D-Code总线以读取数据的方式读取(包含ICP/ISP调适模式以及从内部RAM启动的程序)，以读取数据的方式去访问sLib_Code时，读到的数值全都是0xFF。
- 数据安全库区(sLib_Data)的数据仅能透过D-Code总线读取，不能写入。
- 安全库区的程序代码及数据，除非输入正确的密码，无法被擦除。在密码不正确时，对安全库区执行写入或擦除，将会在FLASH_STS寄存器的WRPRTFLR位置"1"提出警告。
- 终端用户执行主闪存的整片擦除时，安全库区的程序代码及数据不会被擦除。
- 当安全库区的保护功能被启动后，可以透过在sLib_PSW寄存器写入先前设置的密码来解除保护功能。解除安全库区的保护时，芯片将会执行主闪存的整片擦除(包含安全库区的内容)。因此即使软件方案商设置的密码被泄漏，也不会有程序代码外泄的疑虑。

3.3 操作安全库区

详细操作请参考 《AN0040_AT32F403A_407_Security_Library_Application_Note_ZH_V2.x.x》。

4 文档版本历史

表 2. 文档版本历史

日期	版本	变更
2021.12.27	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，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。

© 2021 雅特力科技 (重庆) 有限公司 保留所有权利