

前言

本应用入门指南旨在让用户快速使用 AT32F403xx 进行项目开发。

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	AT32F403 系列
------	-------------

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	调试工具及开发板.....	6
1.1.2	烧录工具及软件	6
1.1.3	AT32 开发环境.....	7
1.1.4	快速替代 SXX 流程.....	12
1.2	AT32F403 芯片的增强功能配置.....	12
1.2.1	PLL 时钟设置.....	12
1.2.2	如何打开 FPU 功能（硬件浮点运算单元）	14
1.2.3	AT32F403 零等待/非零等待 Flash 和内置 SRAM 大小选择配置说明.....	15
1.2.4	加密方式(访问保护,擦写保护,外部 Flash 的加密).....	20
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	27
2.1	程序启动进入 Hard Fault Handler	27
2.2	Keil 项目内 Jlink 无法找到 IC	27
2.3	程序下载过程出问题	27
2.3.1	显示 Error: Flash Download failed-“Cortex-M4”问题.....	27
2.3.2	显示 No Debug Unit Device found 问题	28
2.3.3	显示 RDDI-DAP Error 问题.....	28
2.3.4	ISP 串口下载时卡死问题	28
2.3.5	AT32 恢复下载.....	28
3	文档版本历史.....	30

表目录

表 1. 文档版本历史 30

图目录

图 1. AT32F403 开发板实物图.....	6
图 2. 雅特力科技官方网站 AT-START-F403 开发板资料包	6
图 3. 雅特力科技官方网站 ICP/ISP/AT-Link-Family 资料包	7
图 4. 雅特力科技官方网站 BSP 资料包	7
图 5. Keil_v5 templates 工程示例.....	8
图 6. 雅特力科技官方网站 Pack 包.....	8
图 7. 安装 ArteryTek.AT32F403_DFP	9
图 8. 安装 Keil4_AT32MCU_AddOn.....	9
图 9. Keil 中 Pack Installer 图标	9
图 10. 安装 IAR_AT32MCU_AddOn.....	10
图 11. Keil Debug 选项.....	10
图 12. Keil Debug 选项 Settings 设置.....	11
图 13. Keil Utilities 选项.....	11
图 14. IAR Debug 选项.....	11
图 15. IAR CMSIS-DAP 选项	12
图 16. 使用 SXX32F103 程序在 AT32F403 输出 200MHz 的时钟配置.....	13
图 17. AT32F403 crm_pll_output_range 参数.....	13
图 18. 雅特力科技官方网站时钟配置工具	13
图 19. Keil 环境中选择开启 FPU	14
图 20. Keil 环境中增加开启 FPU 的代码	14
图 21. IAR 环境中选择开启 FPU	15
图 22. IAR 环境中增加开启 FPU 的代码	15
图 23. ICP 工具编辑用户系统数据选择 SRAM 大小	16
图 24. 用户系统数据设置选择 SRAM 大小.....	16
图 25. ISP 工具编辑用户系统数据选择 SRAM 大小.....	17
图 26. ISP Multi-Port Programmer 工具用户系统数据设置.....	18
图 27. SXX 程序定义 Extend_SRAM(void)函数.....	18
图 28. AT32 程序定义 Extend_SRAM(void)函数.....	19
图 29. Keil 启动文件中修改 SRAM 大小.....	19
图 30. IAR 启动文件中修改 SRAM 大小.....	20

图 31. ISP 工具启用访问保护	21
图 32. ISP 工具解除访问保护	21
图 33. ICP 工具启用擦写保护	22
图 34. ICP 工具解除擦写保护	23
图 35. ICP 工具对外部存储器加密操作	24
图 36. ISP 工具对外部存储器加密操作	24
图 37. 读取 Cortex 型号	25
图 38. 读取 UID,PID	25
图 39. 增加开启 FPU 的代码	27
图 40. 下载出现 Flash Download failed-“Cortex- 4”	27
图 41. debug 界面修改	29
图 42. CMSIS DAP 界面修改	29

1 雅特力初步环境准备

雅特力开发环境下载地址：

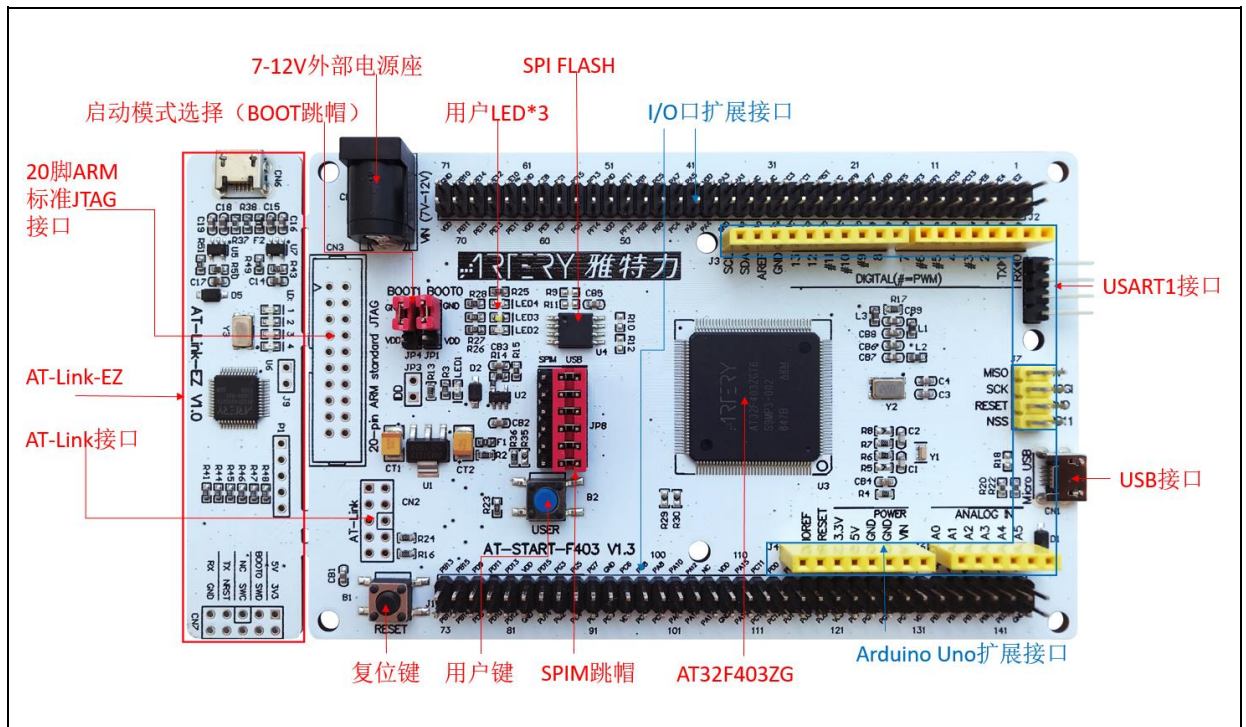
- 雅特力官方网站：<http://www.arterytek.com>

1.1 搭建 AT32 开发环境

1.1.1 调试工具及开发板

目前 AT32F403 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，它也可拆开后可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. AT32F403 开发板实物图



注意：AT-START 板配备资源的详细说明，请参考《UM_AT_START_F40x_Vx.x》，存放路径为雅特力科技官方网站→产品讯息→主流型 MCU→AT32F4xx 系列下载的 Evaluation Board (开发板)资料包解压后AT_START_F40x_Vx.x\03_Documents。

图 2. 雅特力科技官方网站 AT-START-F403 开发板资料包

Evaluation Board (开发板)		
Download	Description	Version
AT-START-F403	AT32F403 开发板并支持Arduino标准接口	V1.3

1.1.2 烧录工具及软件

- AT 烧录工具及软件: AT-Link / AT-Link+ / AT-Link-Pro / AT-Link-ISO / AT-Link-EZ, ICP/ISP。
- 第三方烧录工具: J-Link、安富莱、正点原子、轩微科技、创芯工坊、周立功、迈斯威志、阿莫

烧录器、昂科、永创智能、高勒康达、浦洛、戎象科技、欣扬电子、西尔特、智峰科技等。

注意：上述烧录工具详情请访问[雅特力科技官方网站](#)→技术与开发支持→Hardware Development Tool 页面及第三方编程器（3RD Party Writer）页面。

- ICP 使用说明请参考《UM_ICP_Programmer》，存放路径为[雅特力科技官方网站](#)→产品讯息→主流型 MCU→AT32F4xx 系列下载的 ICP 工具解压后 Artery_ICP_Programmer_Vx.x.xx\Document\UM_ICP_Programmer。
- ISP 使用说明请参考《UM_ISP_Programmer》，存放路径为[雅特力科技官方网站](#)→产品讯息→主流型 MCU→AT32F4xx 系列下载的 ISP 工具解压后 Artery_ISP_Programmer_Vx.x.xx\Document\UM_ISP_Programmer。
- AT-Link 使用说明请参考《UM0004_AT-Link_User_Manual》，存放路径为[雅特力科技官方网站](#)→产品讯息→主流型 MCU→AT32F4xx 系列下载的 AT-Link-Family 工具解压后 AT_Link_CH_Vx.x.x\05_Documents\UM0004_AT-Link_User_Manual_ZH_Vx.x.x。

图 3. 雅特力科技官方网站 ICP/ISP/AT-Link-Family 资料包

Tool		
Download	Description	Version
AT32 IDE_Linux AT32 IDE_Windows	支持AT32 MCU的基于Eclipse开发的跨平台ARM嵌入式系统的软件开发环境	V1.0.02
AT-Link Family	支持AT32 MCU 仿真与在线/离线烧录工具 (包含AT-Link-EZ/AT-Link/AT-Link-Pro/AT-Link-ISO四种工具)	V2.1.1
AT-Link Console	支持AT32 MCU「在电路编程」Console工具	V3.0.02
ICP	支持AT32 MCU「在电路编程」工具	V3.0.05
ISP	支持AT32 MCU「在系统编程」工具	V2.0.06
ISP Multi-Port	支持AT32 MCU一对多设备「在系统编程」工具	V2.0.06

1.1.3 AT32 开发环境

1.1.3.1 模板工程介绍

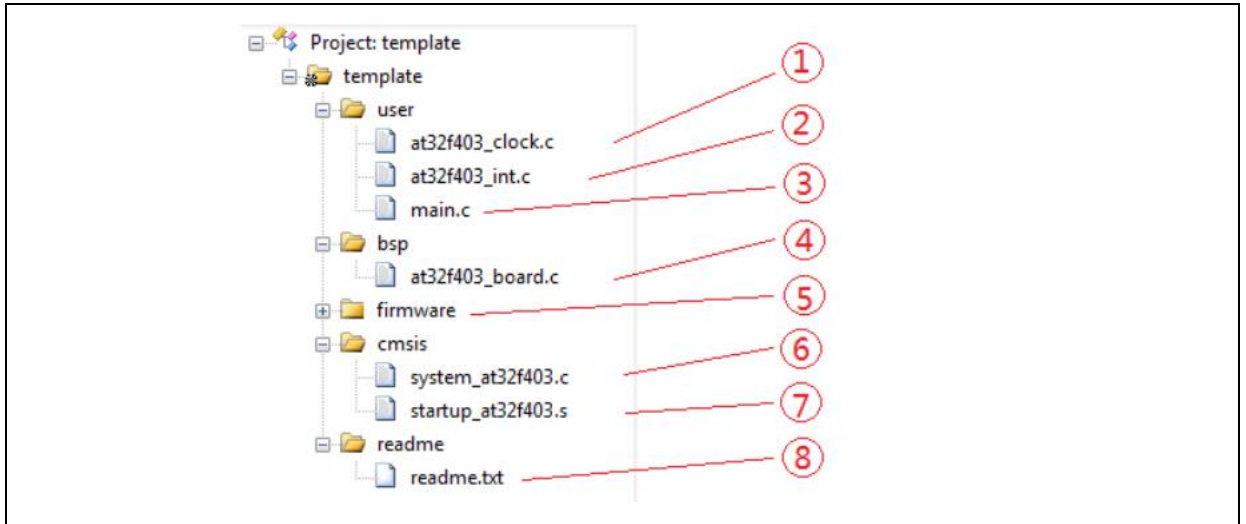
在 ArteryTek 提供的固件库 BSP 中都默认建立好了常用 IDE 的模板工程。BSP 可从[雅特力科技官方网站](#)→产品讯息→主流型 MCU→AT32F4xx 系列获取。

图 4. 雅特力科技官方网站 BSP 资料包

BSP		
Download	Description	Version
Firmware Library	支持AT32F403底层驱动源代码，以固件库(Firmware Library)呈现	V2.0.4

BSP 中创建了 Keil_v5/Keil_v4/IAR_6.10/IAR_7.4/IAR_8.2/eclipse_gcc/at32_ide 的模板工程，存放路径为 AT32F403_Firmware_Library_V2.x.x\project\at_start_f4xx\templates，打开对应工程的文件夹并点击工程文件即可打开对应的 IDE 工程。如下是 Keil_v5 工程示例：

图 5. Keil_v5 templates 工程示例



工程内添加的内容描述如下：

- ① at32f403_clock.c 时钟配置文件，设置了默认的时钟频率及时钟路径。
- ② at32f403_int.c 中断文件，默认编写了部分内核中断函数的代码流程。
- ③ main.c 模板工程的主代码文件。
- ④ at32f403_board.c 板级配置文件，设置了 AT-START 上的按键和 LED 等常用硬件配置。
- ⑤ firmware 下的 at32f403_xx.c 是各片上外设的驱动文件。
- ⑥ system_at32f403.c 系统初始化文件。
- ⑦ startup_at32f403.s 启动文件。
- ⑧ readme.txt 工程的说明文件，记录了模板工程的一些应用功能、设置方式以及关联应用笔记（ApNote）等信息。

除了 templates 外，BSP 还按照外设分类，提供了大量的 examples 示例代码（Keil_v5 工程文件）供用户参考，用户只需要直接打开即可。存放路径为 AT32F403_Firmware_Library_V2.x.x\project\at_start_xxx\examples。

注意：关于 BSP 的更多详细说明，请参考《AT32F403 固件库 BSP&Pack 应用指南》的“4 BSP 使用简述”章节，存放路径为[雅特力科技官方网站](#)→产品讯息→主流型 MCU→AT32F4xx 系列下载的 BSP 解压后 AT32F403_Firmware_Library_Vx.x.x\document。

1.1.3.2 Pack 安装

需要安装 Pack 包在 Keil/IAR 中增加 AT32 MCU 型号，Pack 包可以从[雅特力科技官方网站](#)→产品讯息→主流型 MCU→AT32F4xx 系列下载。

图 6. 雅特力科技官方网站 Pack 包

Download	Description	Version
Keil 4 Keil 5	支持AT32 MCU型号于Keil MDK运行	V2.1.4 V2.1.6
IAR	支持AT32 MCU型号于IAR EWARM 运行	V2.1.2
Segger	支持Segger相关工具可辨识AT32 MCU	V2.0.4

对于 Keil 编译系统，建议 keil4.74 或 5.23 以上版本。Keil_v5 版本需要将 Keil5_AT32MCU_Add On 解压后安装对应 ArteryTek.AT32F403_DFP，对于 Keil_v4 版本需要安装 Keil4_AT32MCU_Add On；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 7. 安装 ArteryTek.AT32F403_DFP

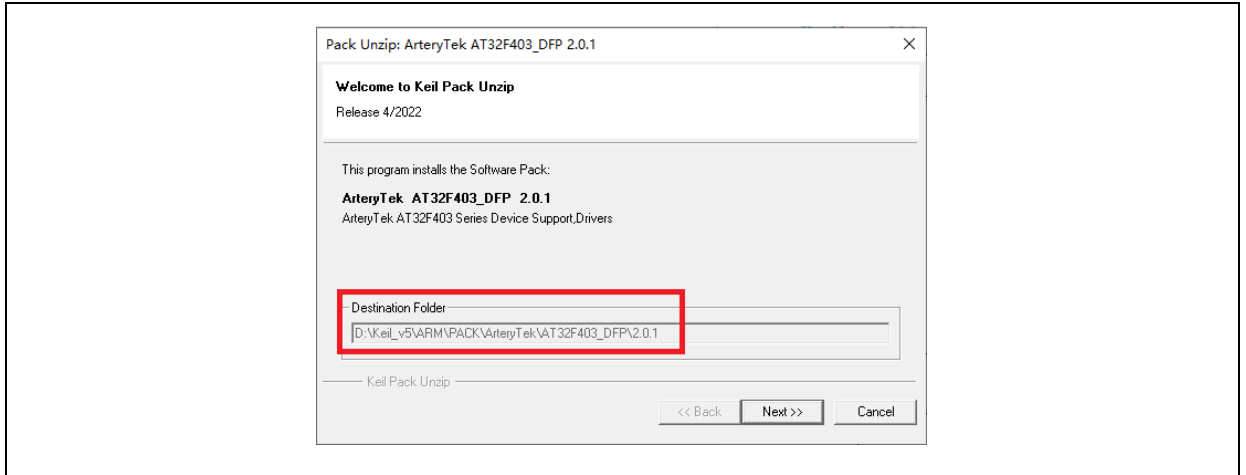
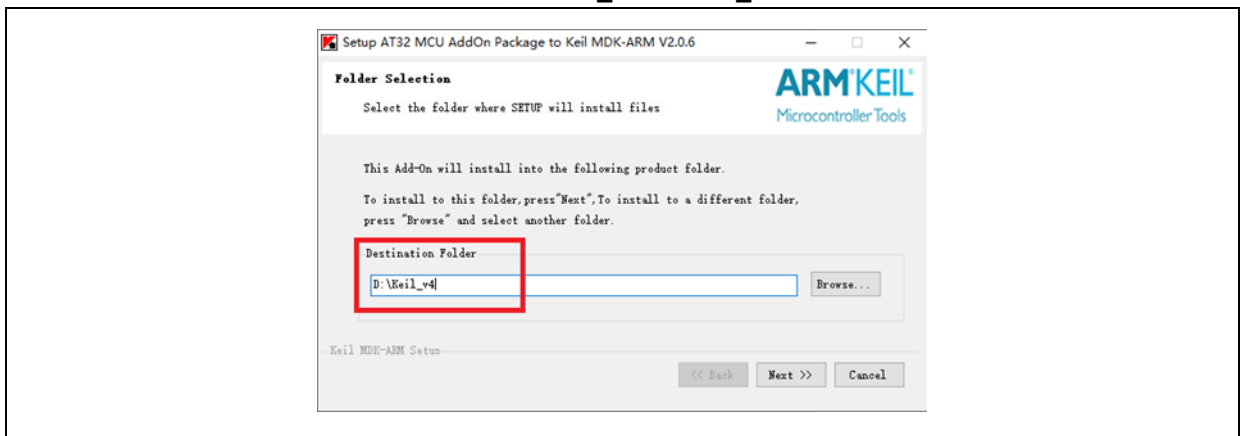
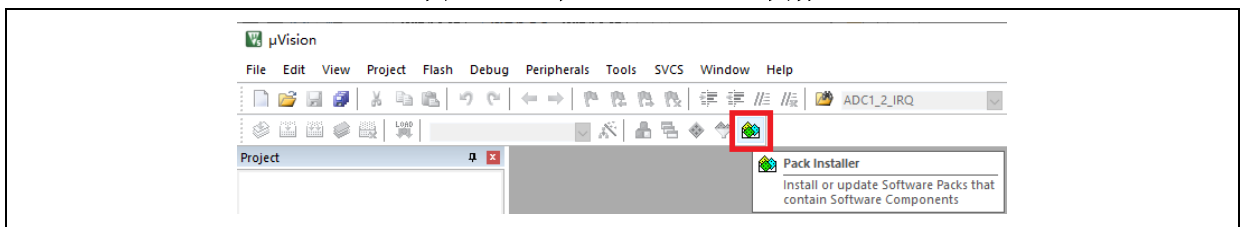


图 8. 安装 Keil4_AT32MCU_AddOn



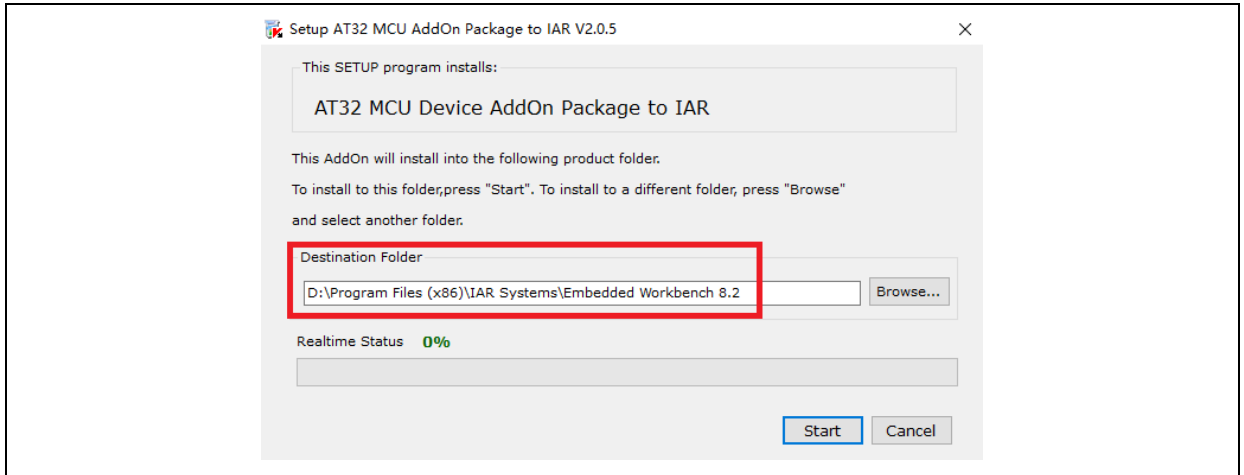
也可打开 keil，点击 Pack Installer 图标，在 Pack Installer 中点击左上角 file，选择 import，导入从 [雅特力科技官方网站](#) 下载好的对应 pack 包完成安装。

图 9. Keil 中 Pack Installer 图标



对于 IAR 编译系统，建议 IAR7.0 或 IAR6.1 以上版本。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 10. 安装 IAR_AT32MCU_AddOn

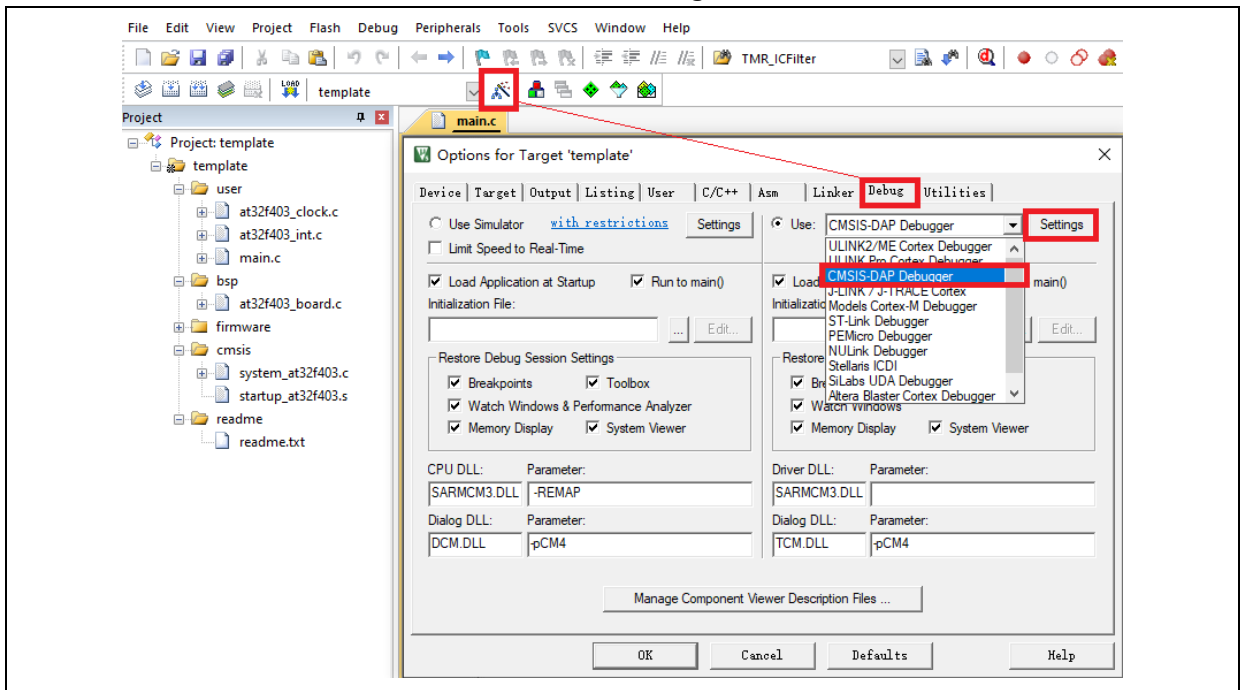


注意：关于 Pack 安装的更多详细说明，请参考《AT32F403 固件库 BSP&Pack 应用指南》的“2 Pack 安装步骤”章节，存放路径为[雅特力科技官方网站](#)→产品讯息→主流型 MCU→AT32F4xx 系列下载的 BSP 解压后 AT32F403_Firmware_Library_Vx.x.x\document。

1.1.3.3 使用 AT-Link 调试及下载

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

图 11. Keil Debug 选项



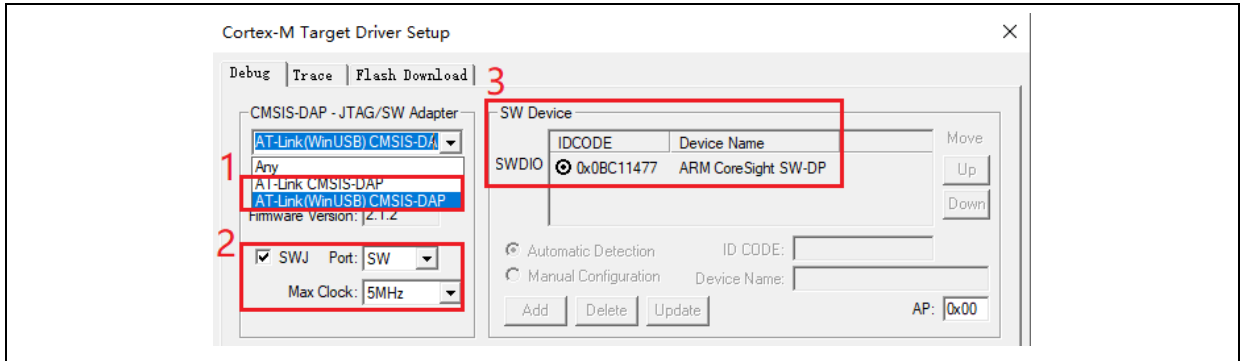
在 Debug 单击 Settings 进入 Cortex-M Target Driver Setup 界面如下图，

1. 先选择 AT-Link(WinUSB)-CMSIS-DAP/AT-Link-CMSIS-DAP；

注意：关于 WinUSB，请参考《FAQ0136_如何使用 AT-LINK 的 WinUSB 功能提升下载速率》，该文档可以从[雅特力科技官方网站](#)→技术与开发支持→FAQ→FAQ0136 下载。

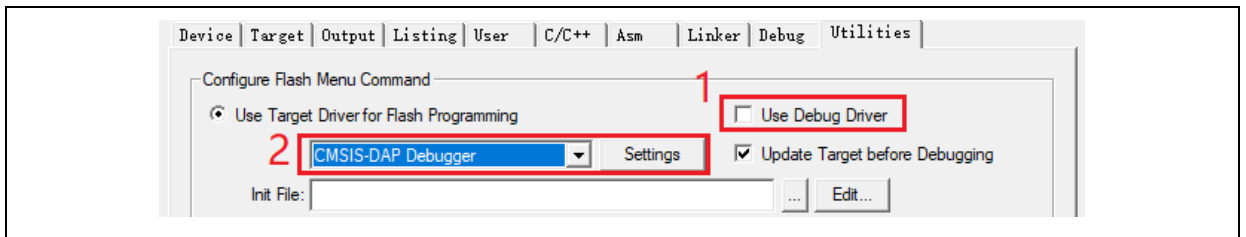
2. Port 选择 SW，再勾选 SWJ；
3. 确认已识别到 ARM SW-DP 调试模块。

图 12. 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 13. Keil Utilities 选项



在 IAR 环境下使用 AT-Link，选中工程，点击 Project，选择 Options，在 Debugger 里选择 CMSIS-DAP 调试器，再在 CMSIS DAP 里选择 SWD。

图 14. IAR Debug 选项

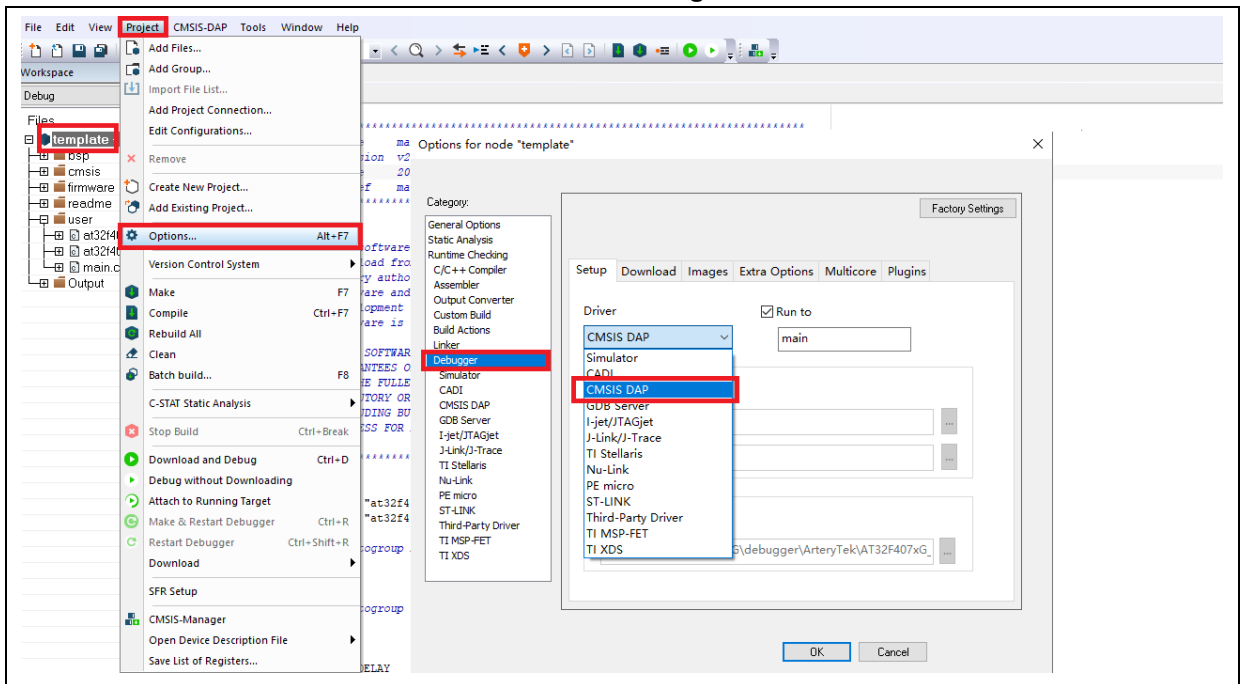
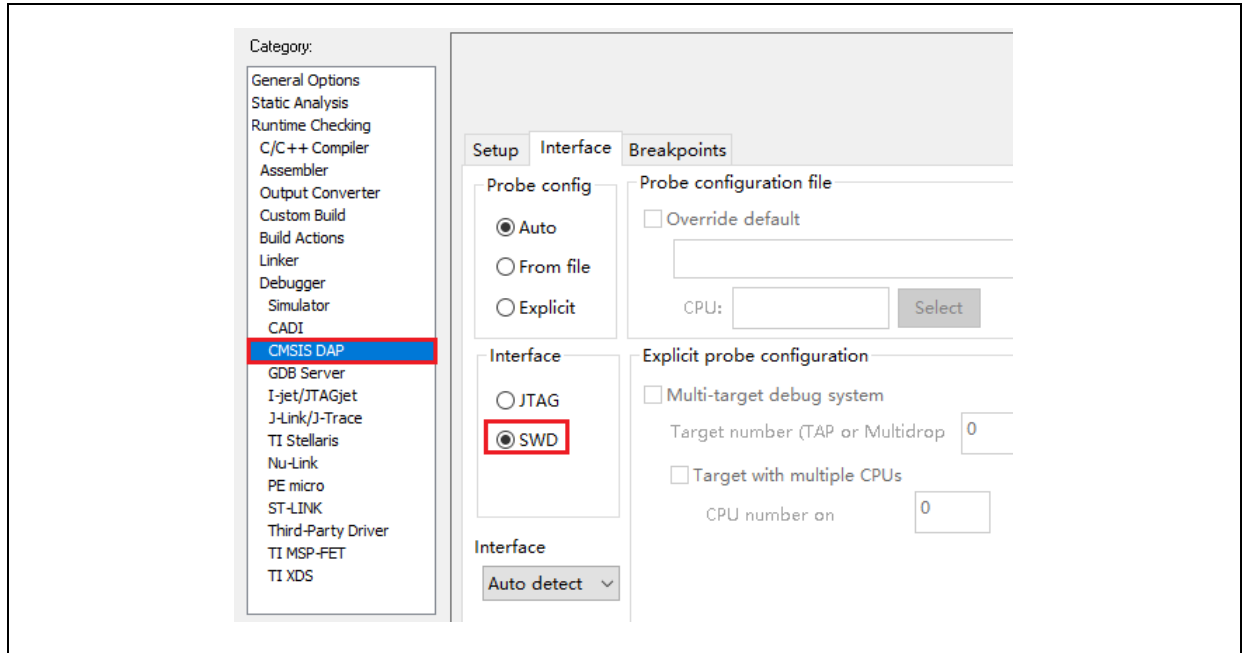


图 15. IAR CMSIS-DAP 选项



注意：关于 Flash 算法文件、MCU 型号切换、Jlink 无法找到 MCU 等的详细说明，请参考《AT32F403 固件库 BSP&Pack 应用指南》相关章节，此处不再赘述。存放路径为[雅特力科技官方网站](#)→产品讯息→主流型 MCU→AT32F4xx 系列下载的 BSP 解压后 AT32F403_Firmware_Library_Vx.x.x\document。

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 AT32F403 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《MG0010_从 SXX32F103 移植到 AT32F403》；
- 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 AT32F403 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 AT32F403 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》，该应用笔记可从[雅特力科技官方网站](#)→技术与开发支持→AP Note→AN0004 获取。

1.2 AT32F403 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

AT32F403 内置的 PLL 最高可输出 200MHz 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 AT32F403 输出大于 72MHz 时钟时，需要额外配置 PLLRANGE 位。
例如 PLL=200MHz 时，设置如下：

图 16. 使用 SXX32F103 程序在 AT32F403 输出 200MHz 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 AT32F403 BSP 时 PLL 设定程序范例：(HEXT=8MHz)

图 17. AT32F403 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=200MHz 时，设置如下：

```
crm_pll_config(CRM_PLL_SOURCE_HEXT_DIV, CRM_PLL_MULT_50, CRM_PLL_OUTPUT_RANGE_GT72MHZ);
```

还可以使用雅特力的时钟配置工具（New Clock Configuration）来快速生成想要的时钟代码并应用到工程，该工具可从[雅特力科技官方网站](#)→产品讯息→主流型 MCU→AT32F4xx 系列获取。

图 18. 雅特力科技官方网站时钟配置工具

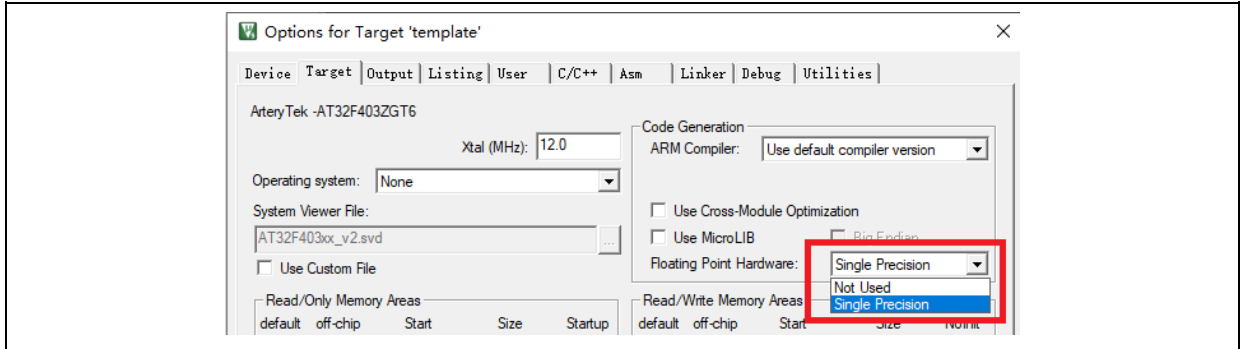
Tool		
Download	Description	Version
AT32 IDE_Linux AT32 IDE_Windows	支持AT32 MCU的基于Eclipse开发的跨平台ARM嵌入式系统的软件开发环境	V1.0.02
New Clock Configuration	支持AT32 MCU的时钟配置	V3.0.03
I2C Timing Configuration	I2C时序配置工具	V2.0.3
CAN BitRate Configuration	AT32 MCU CAN波特率配置	V1.0.00

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 AT32F403 BSP/Pack 或者修改过的 SXX BSP/Pack 直接修改 Floating Point Hardware。

图 19. 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《AT32F403 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. Keil 环境中增加开启 FPU 的代码

```

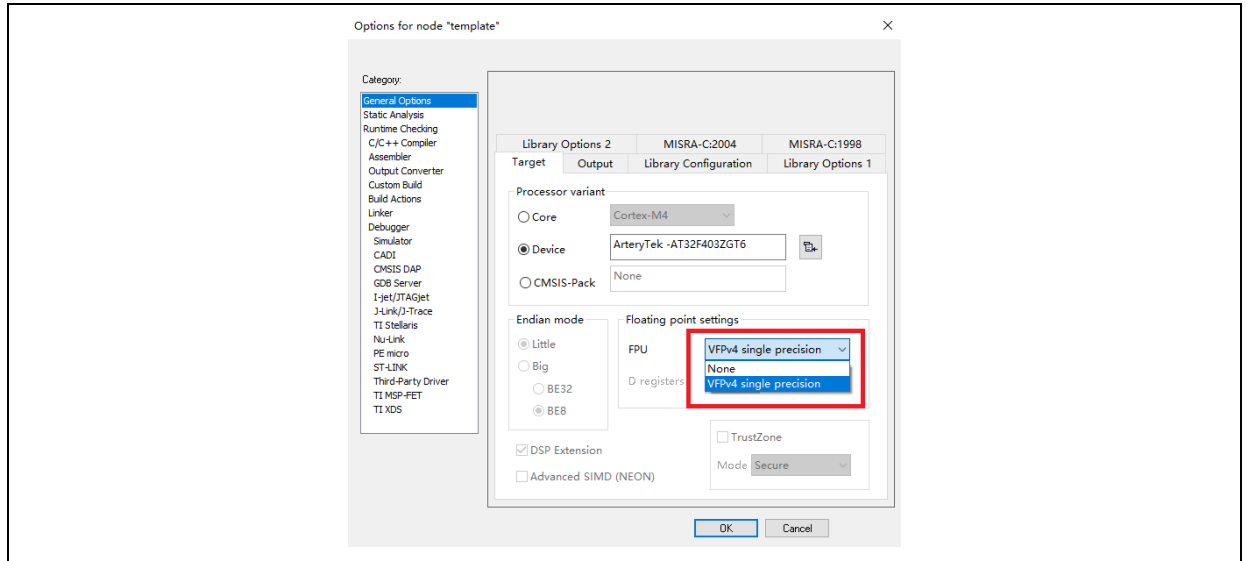
/* Enable FPU*/
#if defined (__FPU_USED) && (__FPU_USED == 1U)
    SCB->CPACR |= ((3U << 10U * 2U) |          /* set CP10 Full Access */
                  (3U << 11U * 2U) );      /* set CP11 Full Access */
#endif

```

IAR 环境中同样分为以下两种情况：

- ① 使用 AT32F403 BSP/Pack 或者修改过的 SXX BSP/Pack 直接修改 Floating Point Hardware。

图 21. 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《AT32F403 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 22. IAR 环境中增加开启 FPU 的代码

```

/* 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

```

更多信息请参考《AN0037_How_to_use_FPU》，该应用笔记可从[雅特力科技官方网站](#)→技术与开发支持→AP Note→AN0037 获取，其中详细描述了如何在 AT32 MCU 上使用 FPU 功能，以及 Keil/IAR 环境下的相关配置。

1.2.3 AT32F403 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 AT32F403ZGT6 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 200MHz，AT32F403 零等待有 256KB 大小，512KB bin 文件的

前面 256KB 可以 200MHz 速率执行，后面 256KB bin 文件存放在非零等待区，执行速率约 80MHz，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：

AT32F403 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。

下文以 AT32F403ZGT6 为例，着重介绍 SRAM 大小选择配置的方法。更多 SRAM 扩展相关的原理说明，请参考《AN0026_Extending_SRAM_in_User's_Program》，该应用笔记可从[雅特力科技官方网站](#)→技术与开发支持→AP Note→AN0026 获取。

① 使用 ICP/ISP 方式

■ Artery ICP Programmer 工具 (BOOT0=0,BOOT1=0)

设备操作--用户系统数据--选择 96KB/224KB--应用到设备。

图 23. ICP 工具编辑用户系统数据选择 SRAM 大小

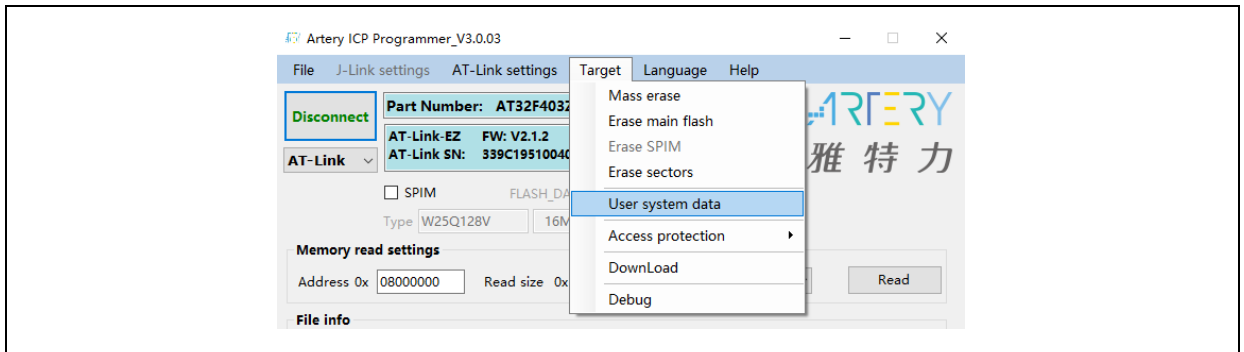
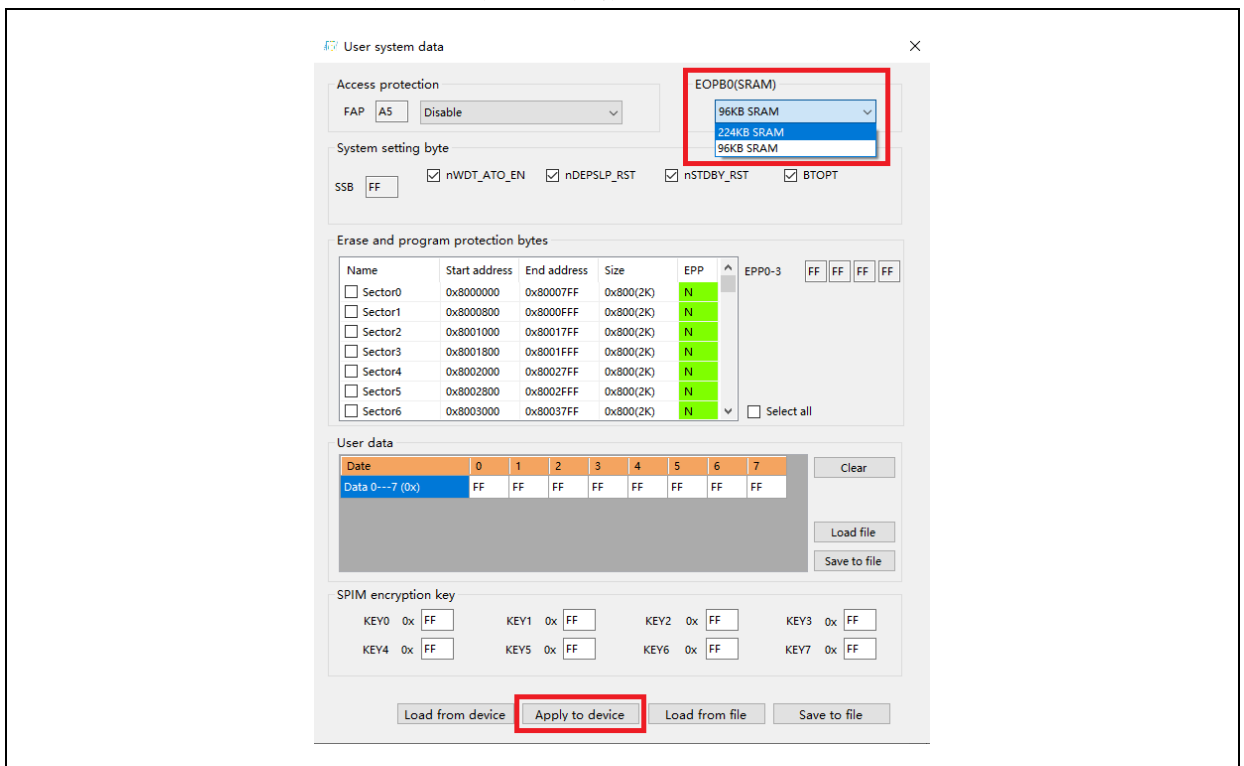
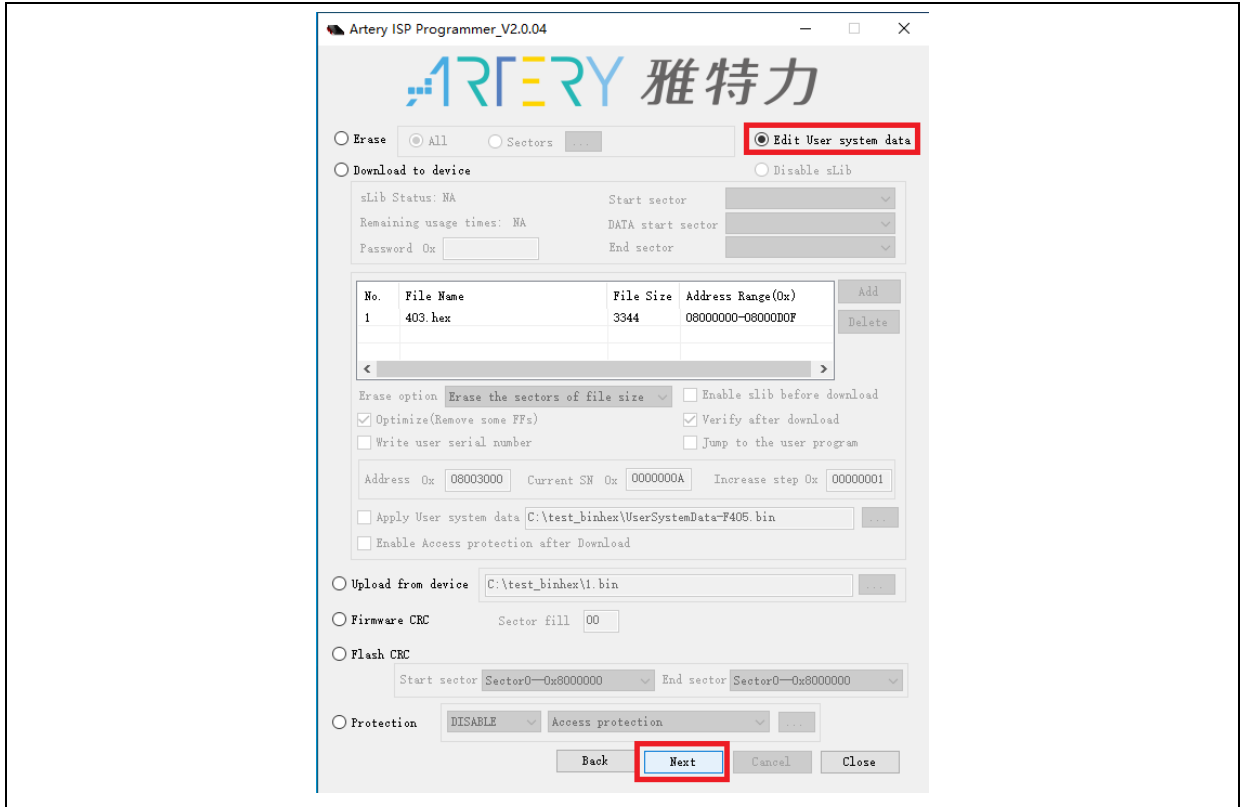


图 24. 用户系统数据设置选择 SRAM 大小



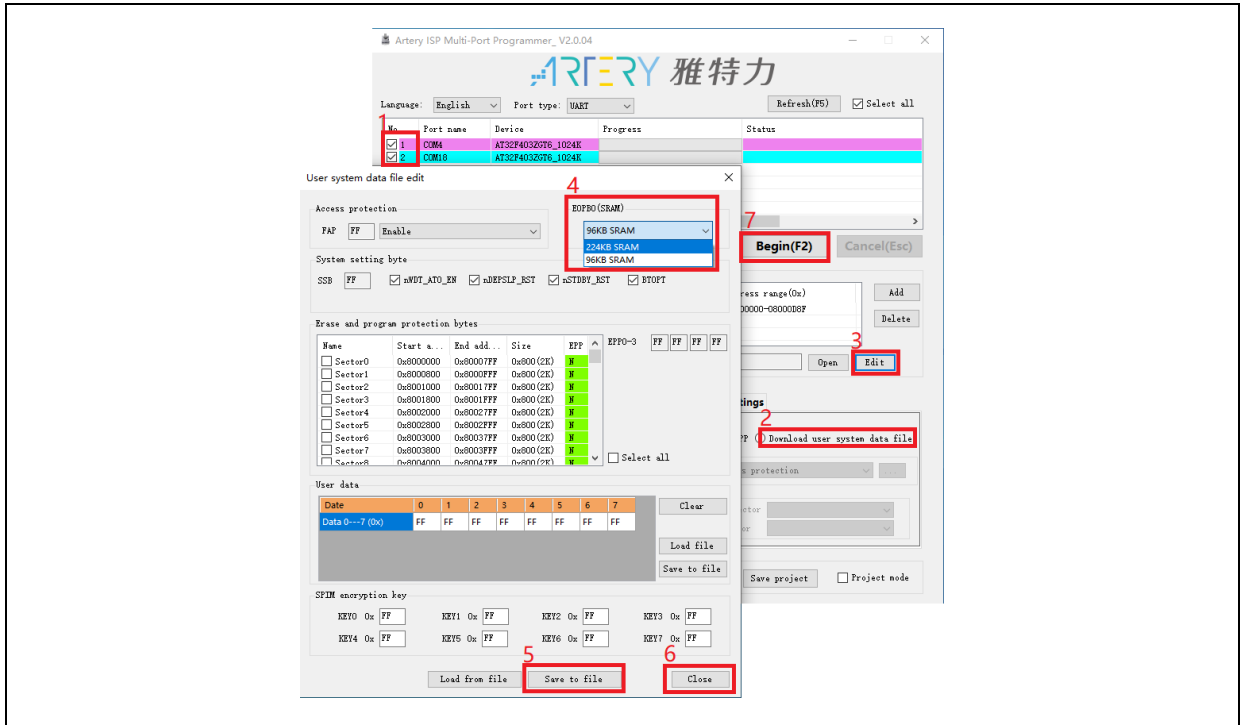
■ Artery ISP Programmer 工具 (BOOT0=1,BOOT1=0)

一直点击下一步，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 25. ISP 工具编辑用户系统数据选择 SRAM 大小**■ Artery ISP Multi-Port Programmer 工具 (BOOT0=1,BOOT1=0)**

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



② 用户也可以使用在 Bootloader program (IAP) 内修改 SRAM 的大小。使用 SXX32F103 程序时，范例如下：

图 27. SXX 程序定义 Extend_SRAM(void)函数

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

static uint32_t F_EOPB0;
F_EOPB0=(uint32_t*)(0x1FFFF810);

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(0x1FFFF810, SRAM_224k);
        /*设置其他用户系统数据*/
        FLASH_Lock();
        NVIC_SystemReset();
    }
}
```

使用 AT32F403 程序时，范例如下：

图 28. AT32 程序定义 Extend_SRAM(void)函数

```

#define SRAM_96k 0xFF
#define SRAM_224k 0xFE

static uint32_t f_eopb0;

f_eopb0=(uint32_t*)(0x1FFFF810);

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(0x1FFFF810, SRAM_224k);

        /*设置其他用户系统数据*/

        flash_lock();

        nvic_system_reset();
    }
}

```

修改用户系统数据前需要擦除用户系统数据区，如果用户系统数据区的其他选项已经设置过，需要将其他设置读出来，擦除后和设置 SRAM 大小一并写入。

③ 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加粗部分代码：

图 29. 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加粗字体

图 30. 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
```

完成以上配置后，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②的写法，定义 `Extend_SRAM(void)` 函数修改 SRAM 大小。

④ 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 (BOOT0=0,BOOT1=0)

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 (BOOT0=1,BOOT1=0)

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 31. ISP 工具启用访问保护

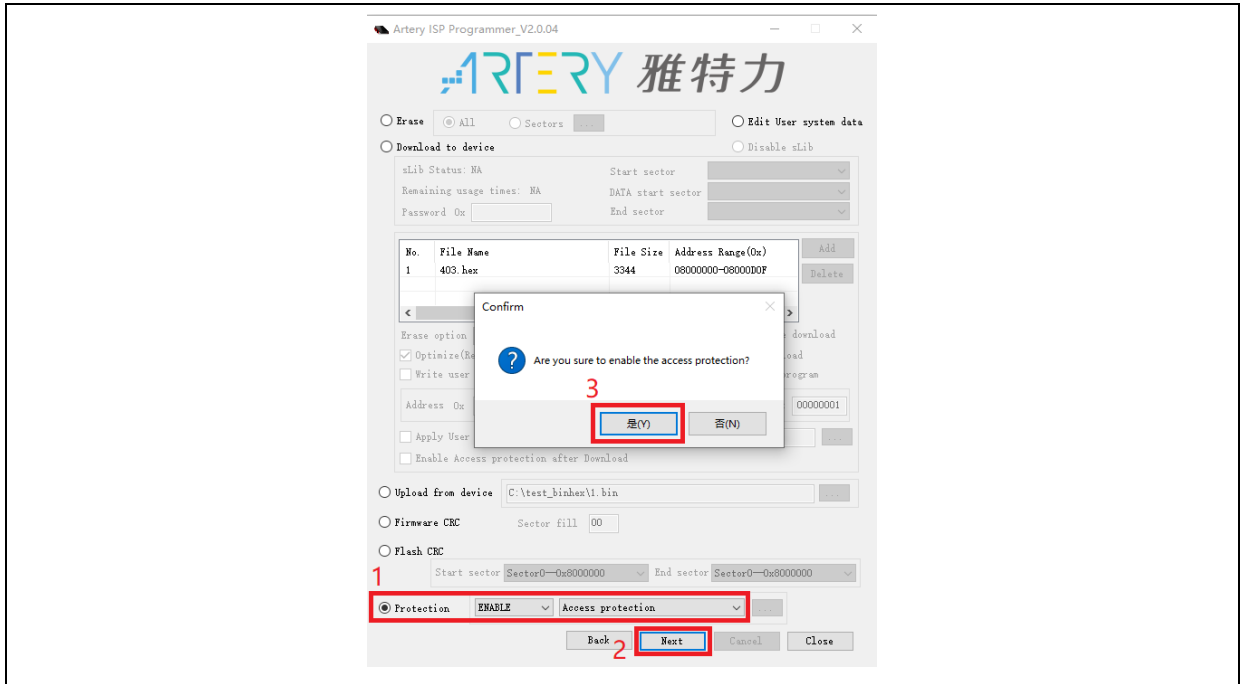
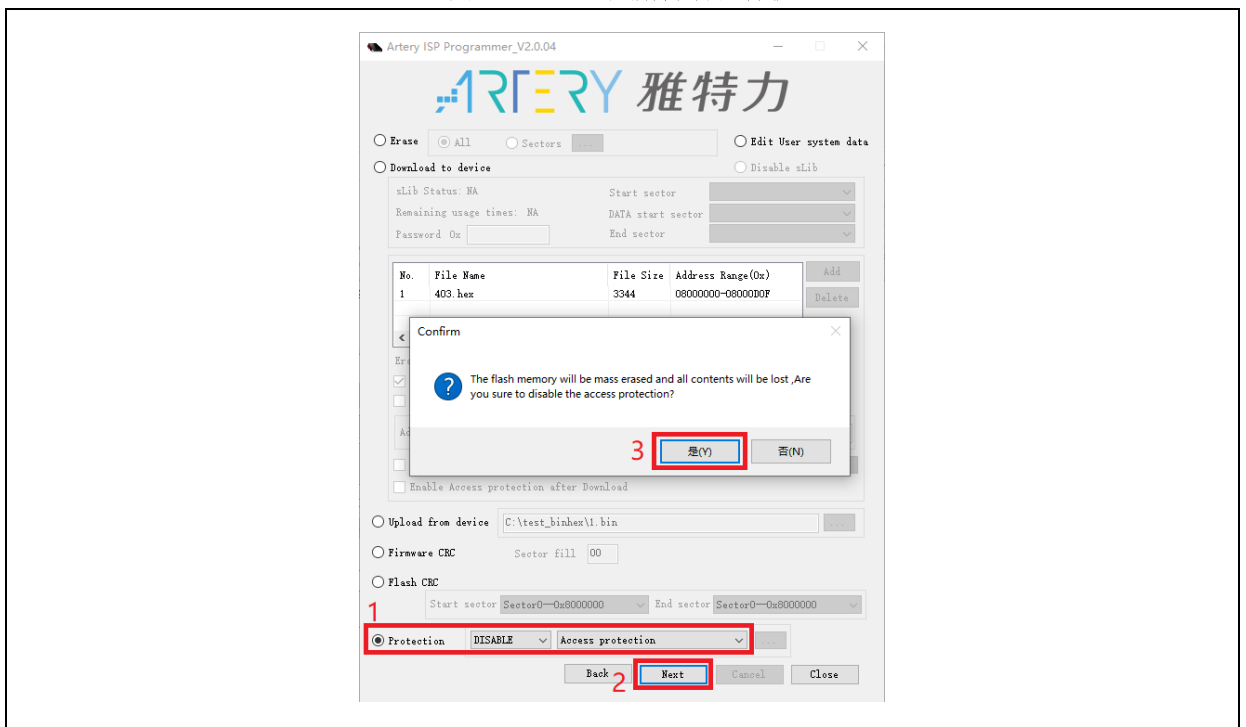


图 32. ISP 工具解除访问保护



■ Artery ISP Multi-Port Programmer 工具 (BOOT0=1,BOOT1=0)

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 进行擦写保护与解除擦写保护操作，如下：

■ Artery ICP Programmer 工具 (BOOT0=0,BOOT1=0)

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 33. ICP 工具启用擦写保护

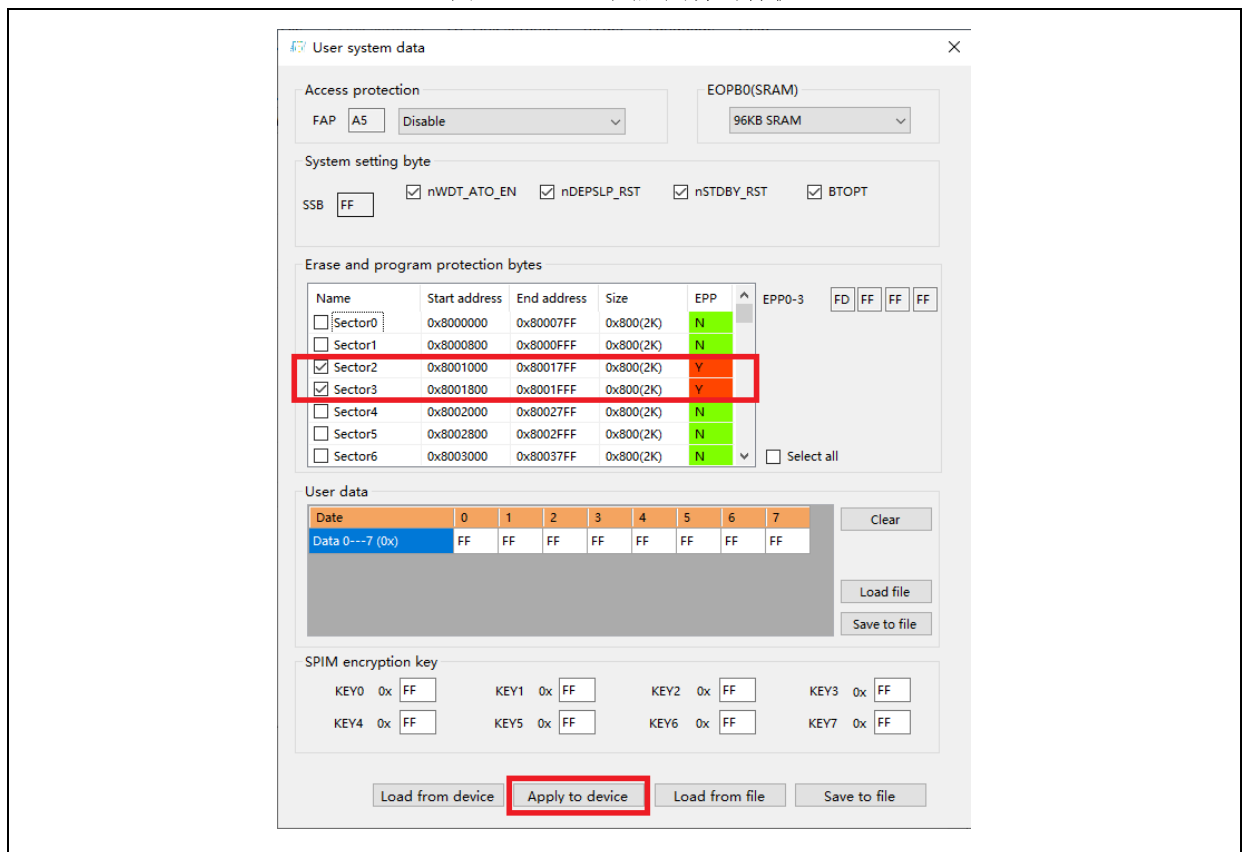
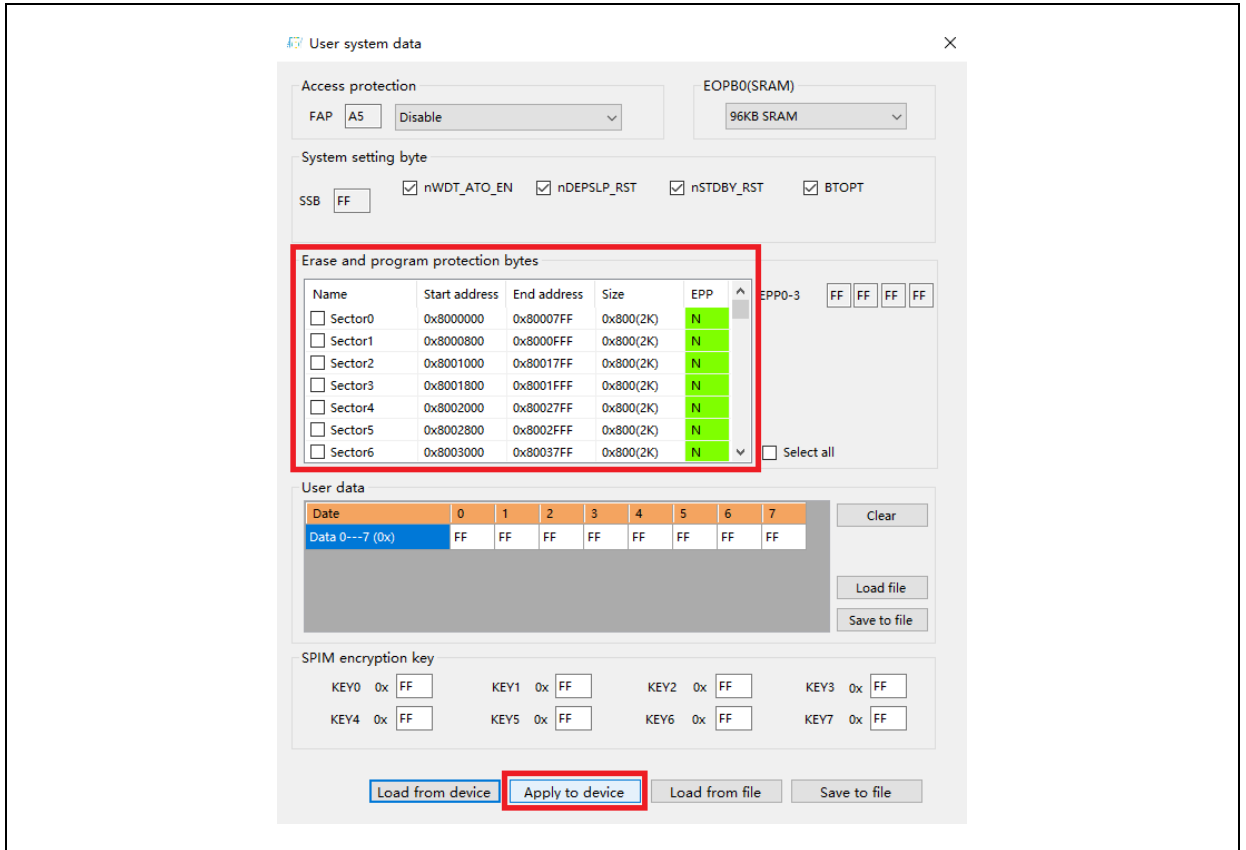


图 34. ICP 工具解除擦写保护



■ Artery ISP Programmer 工具 (BOOT0=1,BOOT1=0)

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 (BOOT0=1,BOOT1=0)

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)

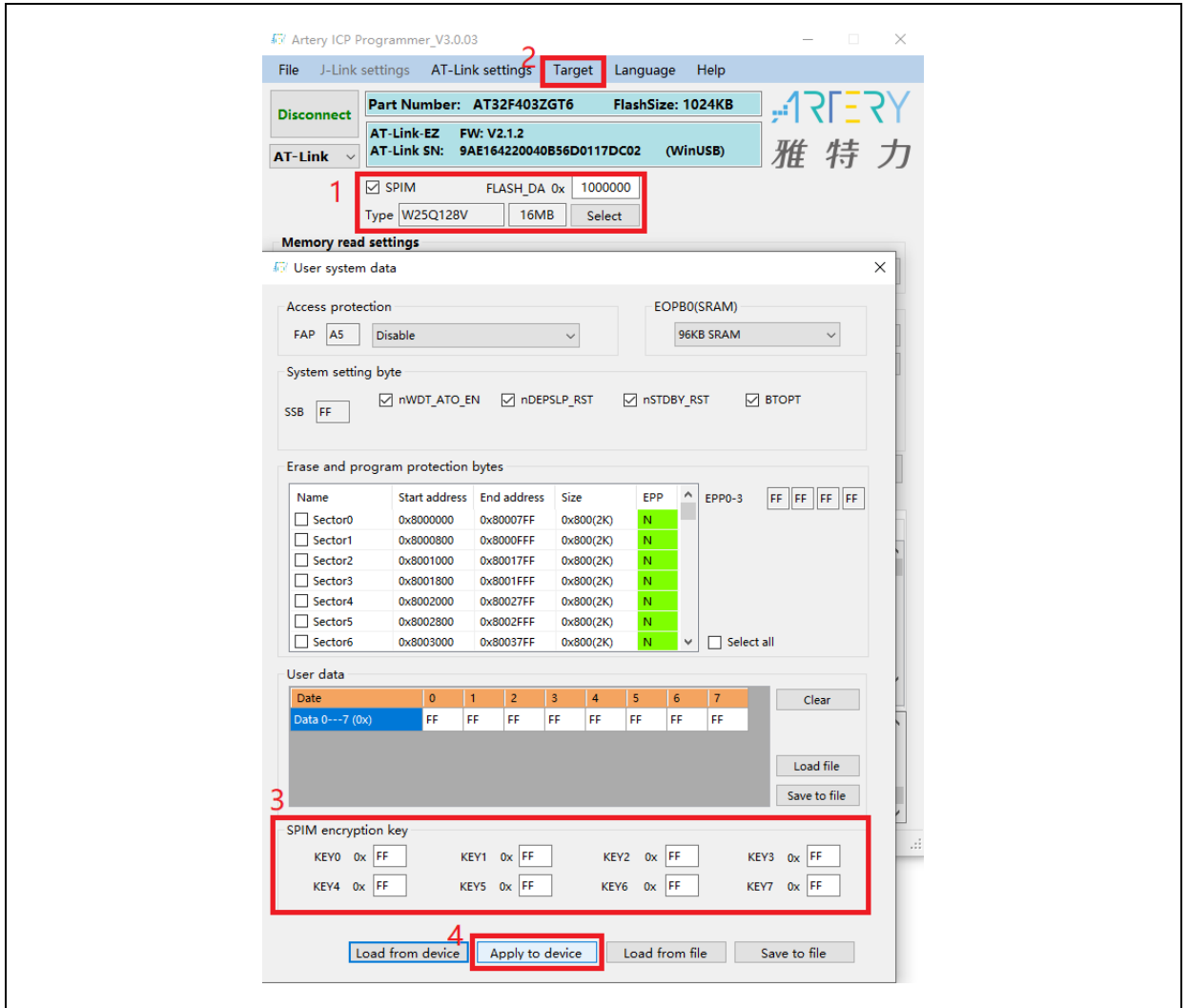
对外部 Flash 加密，需要先进行如下设置加密范围和加密密码后再烧录用户程序，然后开启访问保护。加密范围指的是从 0x08400000 开始需要加密的空间大小。外部存储器加密 Key，如果全部是 0xFF 或者 0x00 则不加密，否则进行加密。解除访问保护会将外部存储器加密 Key 设置为全 0xFF。

以下为 ICP/ISP 工具加密外部存储器的操作说明：

■ Artery ICP Programmer 工具 (BOOT0=0,BOOT1=0)

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。

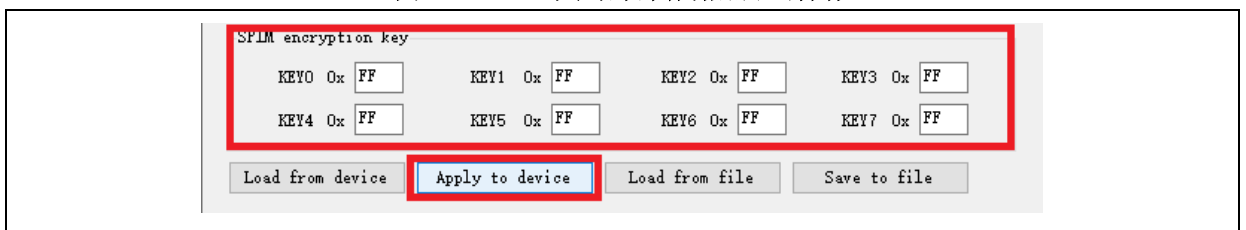
图 35. ICP 工具对外部存储器加密操作



■ Artery ISP Programmer 工具 (BOOT0=1,BOOT1=0)

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 36. ISP 工具对外部存储器加密操作



■ Artery ISP Multi-Port Programmer 工具 (BOOT0=1,BOOT1=0)

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。

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 37. 读取 Cortex 型号

```
cortex_id = *(uint32_t *)0xE00ED00;// 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 UID,PID 方式区分

图 38. 读取 UID,PID

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

/* 用于存放 ID */
uint8_t ID[5] = {0};

/* AT32F403MCU type table */
const uint64_t AT32_MCU_ID_TABLE[] =
{
    0x0000000170050240, //AT32F403ZCT6 256KB LQFP144
    0x0000000170050344, //AT32F403ZGT6 1024KB LQFP144
    ...
};

/* 获取 UID/PID */
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 UID/PID */
AT_device_id = ((uint64_t)ID[0]<<32)|((uint64_t)ID[1]<<24)|((uint64_t)ID[2]<<16)|((uint64_t)ID[3]<<8)|((u
int64_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。更多信息请参考各型号技术手册 RM 的调试（DEBUG）章节以及《AN0016_Recognize_AT32_MCU》，该应用笔记可从[雅特力科技官方网站](#)→技术与开发支持→AP Note→AN0016 获取。

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 39. 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 Keil 项目内 Jlink 无法找到 IC

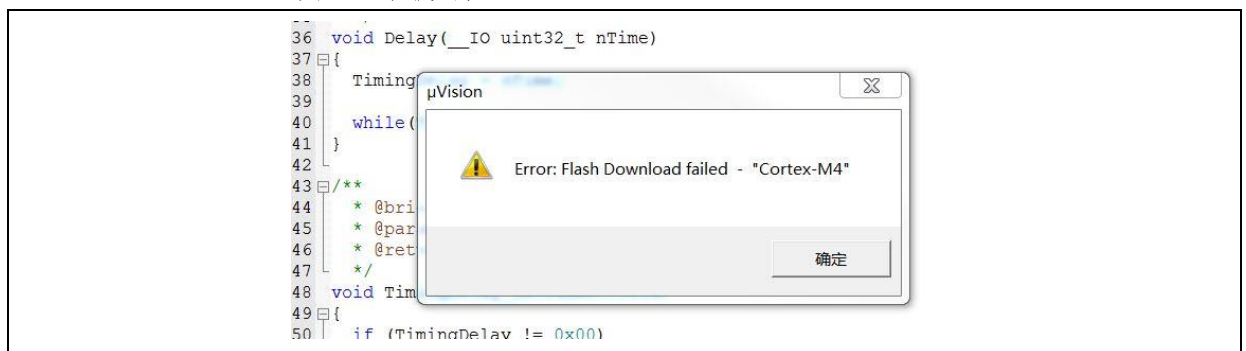
- 请参考《FAQ0008_Keil 项目内 Jlink 无法找到 IC 问题》，该文档可以从[雅特力科技官方网站](#)→技术与开发支持→FAQ→FAQ0008 下载。
- 请参考《FAQ0132_JLink 手动添加 Artery MCU》，该文档及其附件可以从[雅特力科技官方网站](#)→技术与开发支持→FAQ→FAQ0132 下载。

2.3 程序下载过程出问题

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

在 KEIL 仿真或下载时弹出：

图 40. 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”。

2.3.2 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。

2.3.3 显示 RDDI-DAP Error 问题

- 编译器优化等级过高，如 Keil AC6 编译器的默认优化等级-Oz，需要改为-O0/-O1。
- 在程序中将 JTAG/SWD PIN disable，解决方法参考“2.2.5 AT32 恢复下载”。

2.3.4 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。

2.3.5 AT32 恢复下载

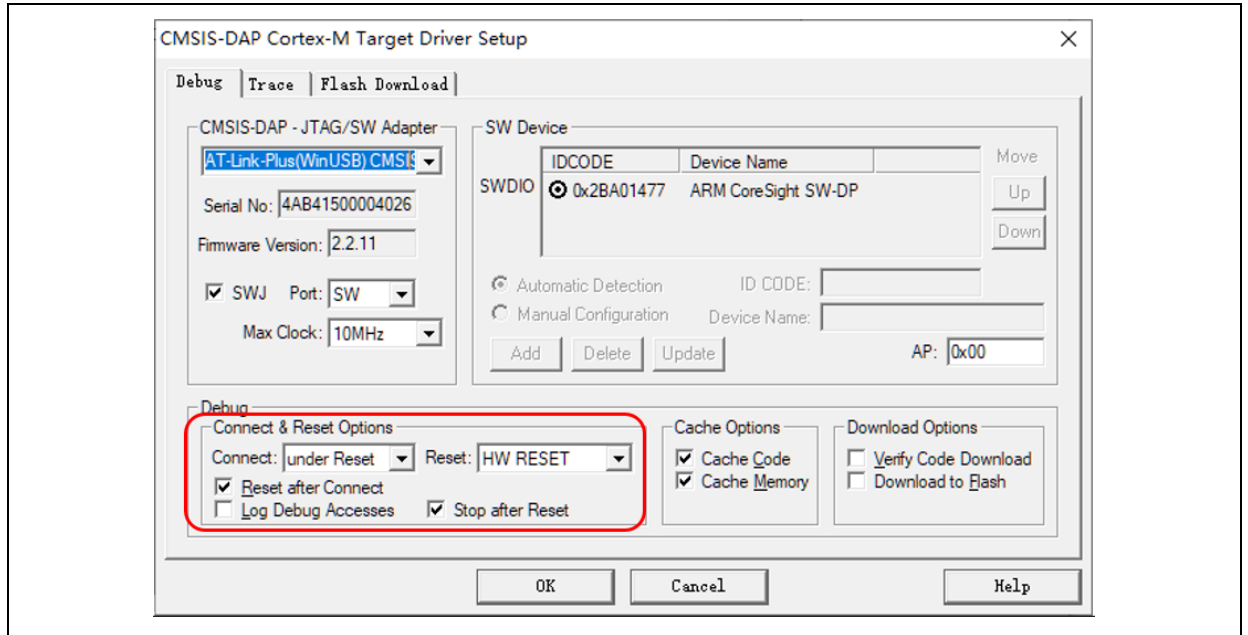
在使用 AT32F403 时，用户可能在以下操作后无法再次下载程序：

- 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 等低功耗模式后，无法下载程序并且找不到 JTAG/SWD device。

解决该问题的原理是在程序还没有运行时使芯片 HALT 住，介绍几种方式如何解决该问题。

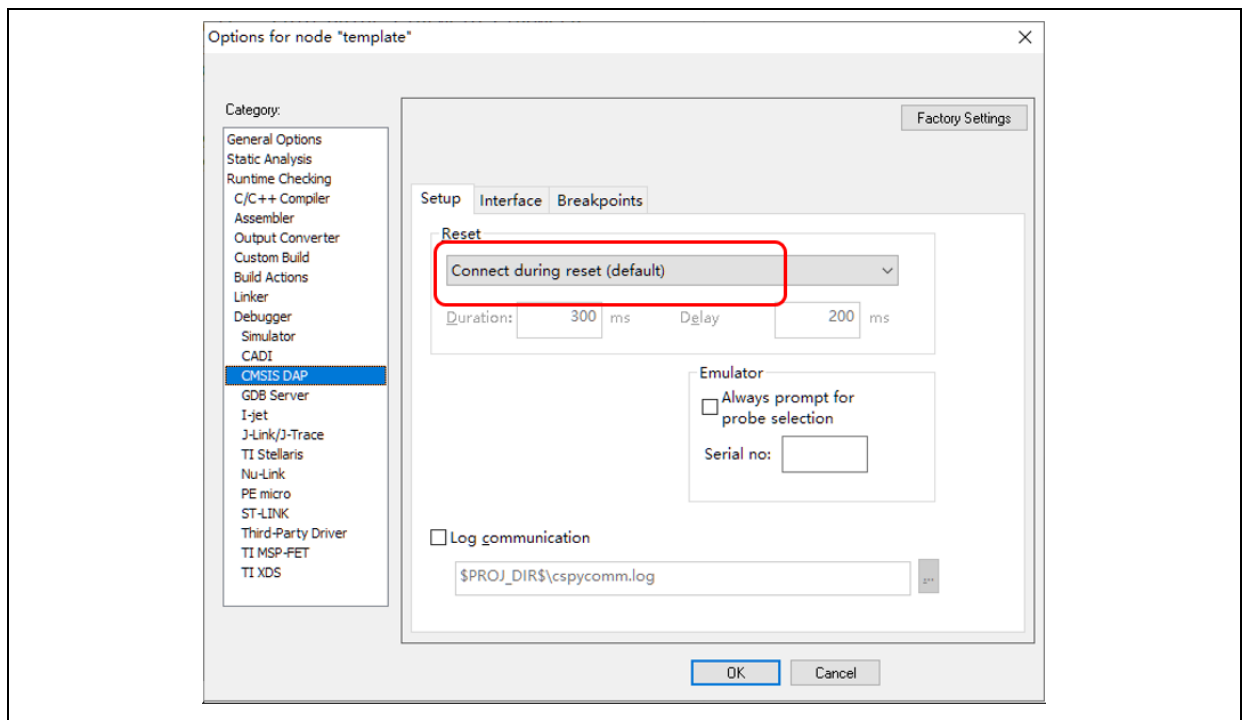
1. 更改芯片 BOOT 模式，改为启动程序存储器启动或者 sram 启动，然后通过复位脚复位芯片，此时就可以擦除程序恢复下载。
2. 使用 ICP 软件工具搭配 AT-Link 调试器，连接 AT-Link RST pin 到芯片的复位脚，ICP 界面上直接点击连接即可正常连接，连接后擦除芯片内程序即可恢复。
3. 使用 Keil 软件搭配 AT-Link 调试器，连接 AT-Link RST pin 到芯片的复位脚，在 Keil 的 debug 界面修改为下图中红框内的选项，就可以擦除程序恢复下载。

图 41. debug 界面修改



4. 使用 IAR 软件搭配 AT-Link 调试器，连接 AT-Link RST pin 到芯片的复位脚，在 IAR 的 CMSIS DAP 界面修改为下图中红框内的选项，就可以擦除程序恢复下载

图 42. CMSIS DAP 界面修改



3 文档版本历史

表 1. 文档版本历史

日期	版本	变更
2022.04.25	2.0.0	最初版本
2022.08.05	2.0.1	优化部分表述，更新第三方烧录工具支持
2022.10.09	2.0.2	增加部分开发环境与文件路径说明
2022.10.21	2.0.3	完善UID,PID表述
2024.01.05	2.0.4	更新“AT32恢复下载”的描述
2024.03.06	2.0.5	格式调整

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、死亡及财产损失的应用。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。

© 2024 · 雅特力科技 · 保留所有权利