

AT32F435 & AT32F437入门使用指南

前言

本应用入门指南旨在让用户快速使用 AT32F435xx /AT32F437xx 进行项目开发，AT32F437 相较于 AT32F435 增加了以太网（EMAC）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	AT32F435xx
	AT32F437xx

目录

1	雅特力初步环境准备	7
1.1	搭建 AT32 开发环境.....	7
1.1.1	调试工具及开发板.....	7
1.1.2	烧录工具及软件	7
1.1.3	AT32 开发环境.....	8
1.1.4	快速替代 AT32F403A/407 流程.....	13
1.2	AT32F435 /AT32F437 芯片的增强功能配置.....	13
1.2.1	PLL 时钟设置.....	13
1.2.2	如何打开 FPU 功能（硬件浮点运算单元）	15
1.2.3	AT32F435 /AT32F437 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).....	19
1.2.5	在程序中区分 AT32 与其他 IC 方法.....	22
1.2.6	AT32F435/AT32F437 高级功能.....	23
2	下载编译过程常见问题.....	25
2.1	程序启动进入 Hard Fault Handler	25
2.2	Keil 项目内 Jlink 无法找到 IC	25
2.3	程序下载过程出问题	25
2.3.1	显示 Error: Flash Download failed-“Cortex-M4”问题	25
2.3.2	显示 No Debug Unit Device found 问题	26
2.3.3	显示 RDDI-DAP Error 问题.....	26
2.3.4	ISP 串口下载时卡死问题	26
2.3.5	AT32 恢复下载.....	26
3	安全库区 sLib(Security Library)	27
3.1	概述	27
3.2	应用原理	27
3.3	操作安全库区.....	27

4	文档版本历史	28
---	--------------	----

表目录

表 1. 文档版本历史	28
-------------------	----

图目录

图 1. AT-START-F437 开发板及 AT-Link-EZ 实物图	7
图 2. 雅特力科技官方网站 AT-START-F437 开发板资料包	7
图 3. 雅特力科技官方网站 ICP/ISP/AT-Link-Family 资料包	8
图 4. 雅特力科技官方网站 BSP 资料包	8
图 5. Keil_v5 templates 工程示例	9
图 6. 雅特力科技官方网站 Pack 包.....	9
图 7. 安装 ArteryTek.AT32F435_437_DFP	10
图 8. 安装 Keil4_AT32MCU_AddOn	10
图 9. Keil 中 Pack Installer 图标	10
图 10. 安装 IAR_AT32MCU_AddOn	11
图 11. Keil Debug 选项	11
图 12. Keil Debug 选项 Settings 设置	12
图 13. Keil Utilities 选项.....	12
图 14. IAR Debug 选项.....	12
图 15. IAR CMSIS-DAP 选项	13
图 16. PLL 自动滑顺频率切换配置	14
图 17. ICP 工具编辑用户系统数据选择 SRAM 大小	16
图 18. 用户系统数据设置选择 SRAM 大小.....	16
图 19. ISP 工具编辑用户系统数据选择 SRAM 大小.....	17
图 20. 定义 Extend_SRAM(void)函数修改 SRAM 大小	17
图 21. Keil 启动文件中修改 SRAM 大小.....	18
图 22. IAR 启动文件中修改 SRAM 大小.....	19
图 23. ISP 工具启用访问保护.....	20
图 24. ISP 工具解除访问保护.....	20
图 25. ICP 工具启用擦写保护.....	21
图 26. ICP 工具解除擦写保护.....	22
图 27. 读取 Cortex 型号	22
图 28. 读取 UID,PID	23
图 29. AT-SURF-F437 开发板	24
图 30. 增加开启 FPU 的代码	25

图 31. 下载出现 Flash Download failed-“Cortex- M4”	25
---	----

1 雅特力初步环境准备

雅特力开发环境下载地址:

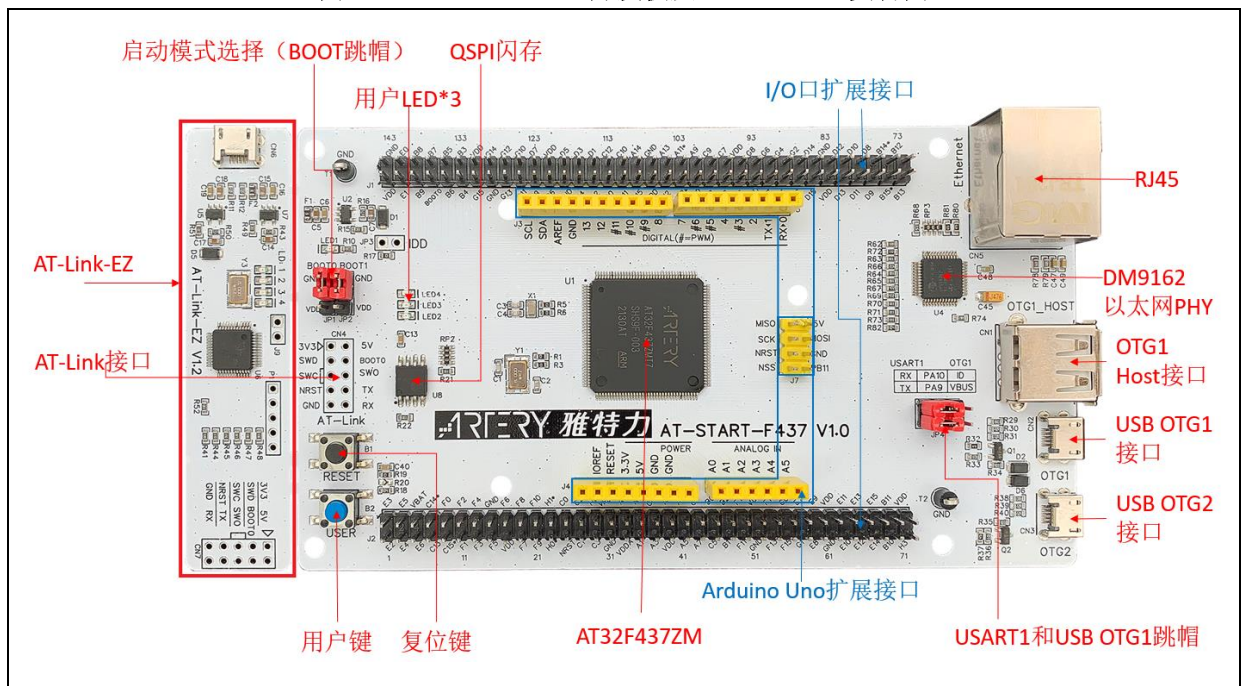
- 雅特力科技官方网站: <http://www.arterytek.com>

1.1 搭建 AT32 开发环境

1.1.1 调试工具及开发板

目前 AT32F435 /AT32F437 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. AT-START-F437 开发板及 AT-Link-EZ 实物图



注意: AT-START 板配备资源的详细说明, 请参考《UM_AT_START_F43x_Vx.x》, 存放路径为[雅特力科技官方网站](http://www.arterytek.com)→产品讯息→高性能 MCU→AT32F4xx 系列下载的 Evaluation Board (开发板)资料包解压后AT_START_F43x_Vx.x\03_Documents。

图 2. 雅特力科技官方网站 AT-START-F437 开发板资料包

Evaluation Board (开发板)		
Download	Description	Version
AT-START-F437	AT32F437开发板并支持Arduino标准接口	V1.0

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→高性能 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→高性能 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→高性能 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

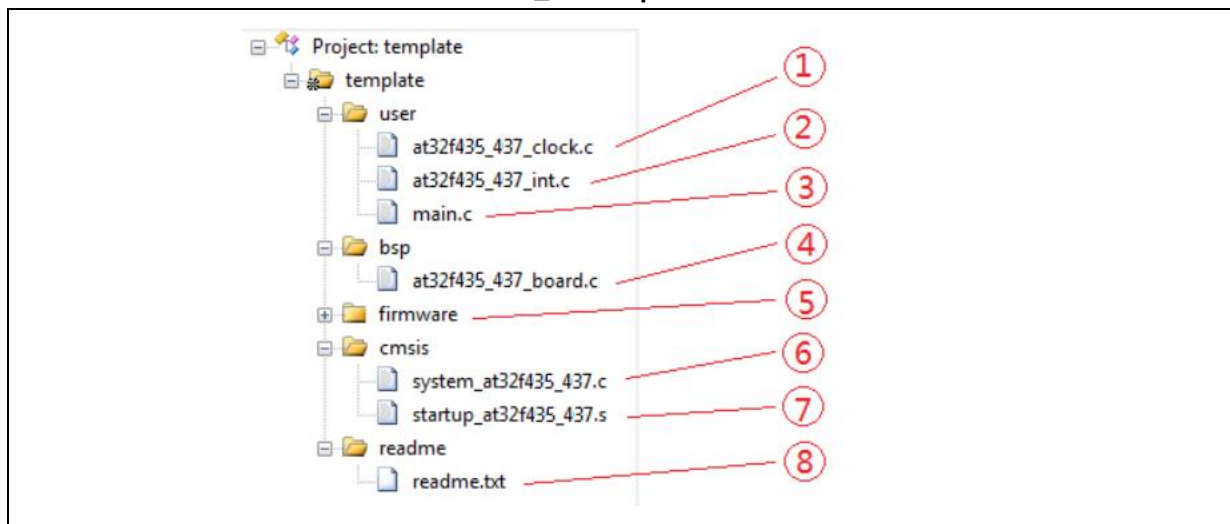
在 ArteryTek 提供的固件库 BSP 中都默认建立好了常用 IDE 的模板工程。BSP 可从[雅特力科技官方网站](#)→产品讯息→高性能 MCU→AT32F4xx 系列获取。

图 4. 雅特力科技官方网站 BSP 资料包

BSP		
Download	Description	Version
Firmware Library	支持AT32F437底层驱动源代码，以固件库(Firmware Library)呈现	V2.1.0

以 AT32F435/437 系列为例，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 AT32F435_437_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 at32f435_437_clock.c 时钟配置文件，设置了默认的时钟频率及时钟路径。
- ② at32f435_437_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 at32f435_437_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 at32f435_437_xx.c 是各片上外设的驱动文件。
- ⑥ system_at32f435_437.c 系统初始化文件。
- ⑦ startup_at32f435_437.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 AT32F435_437_Firmware_Library_V2.x.x\project\at_start_f4xx\examples。

注意：关于 BSP 的更多详细说明，请参考《AT32F435_437 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→高性能 MCU→AT32F4xx 系列下载的 BSP 解压后 AT32F435_437_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→高性能 MCU→AT32F4xx 系列下载。

图 6. 雅特力科技官方网站 Pack 包

Pack		
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_AddOn 解压后安装对应 ArteryTek.AT32F435_437_DFP, 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 7. 安装 ArteryTek.AT32F435_437_DFP

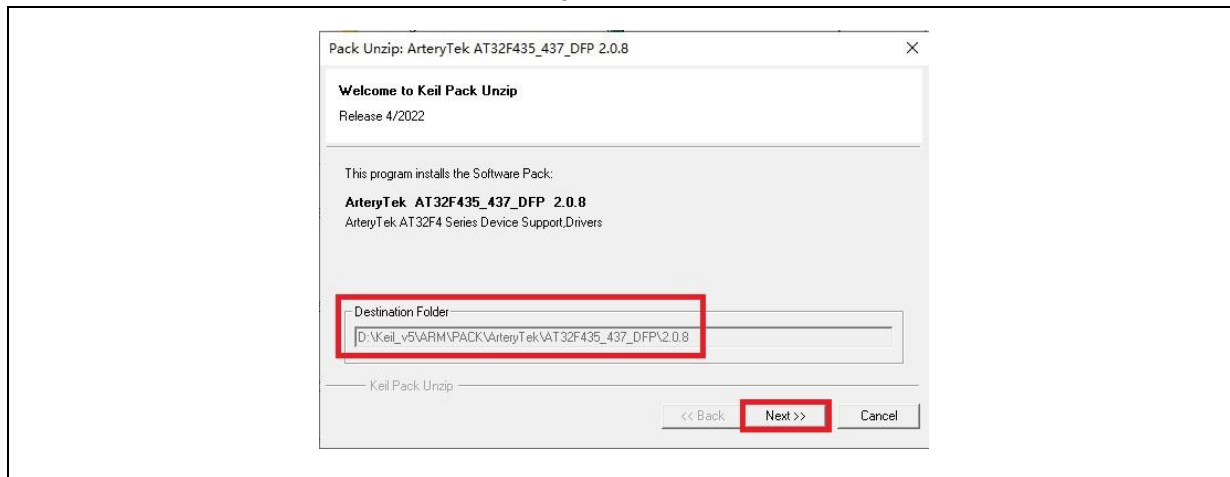
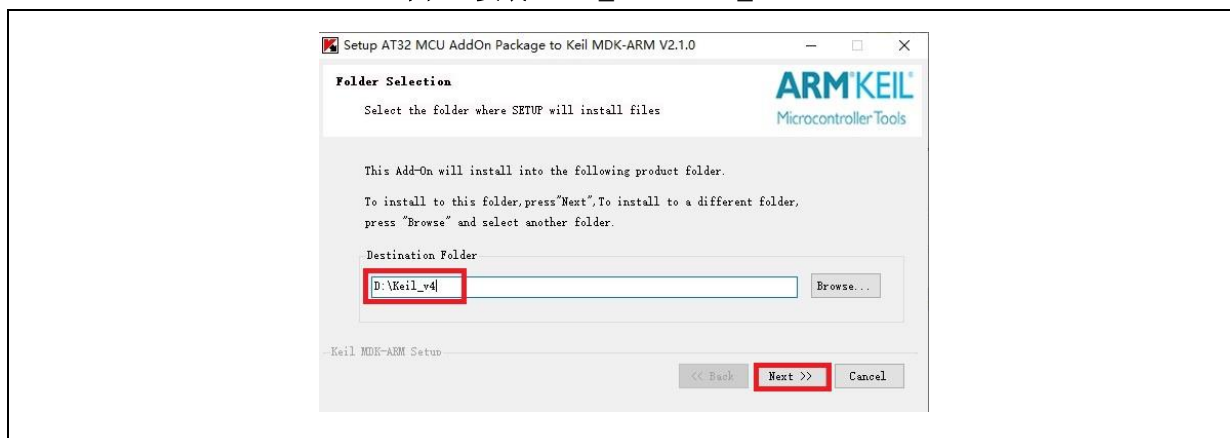
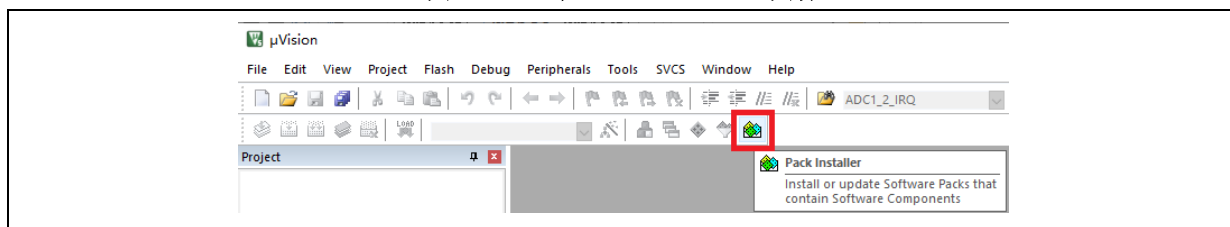


图 8. 安装 Keil4_AT32MCU_AddOn



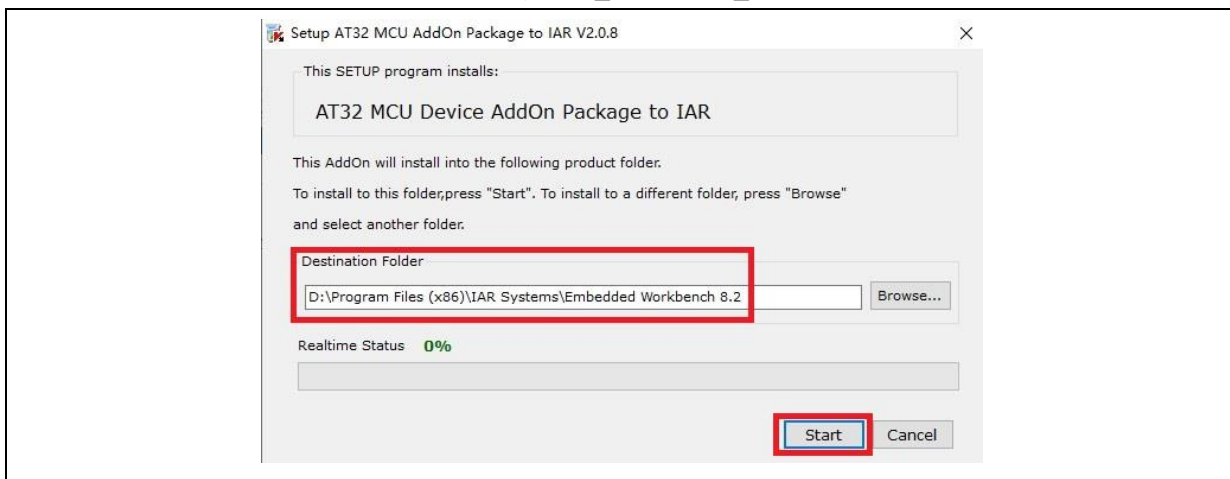
也可打开 keil, 点击 Pack Installer 图标, 在 Pack Installer 中点击左上角 file, 选择 import, 导入从 [雅特力科技官方网站](#) 下载好的对应 pack 包完成安装。

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

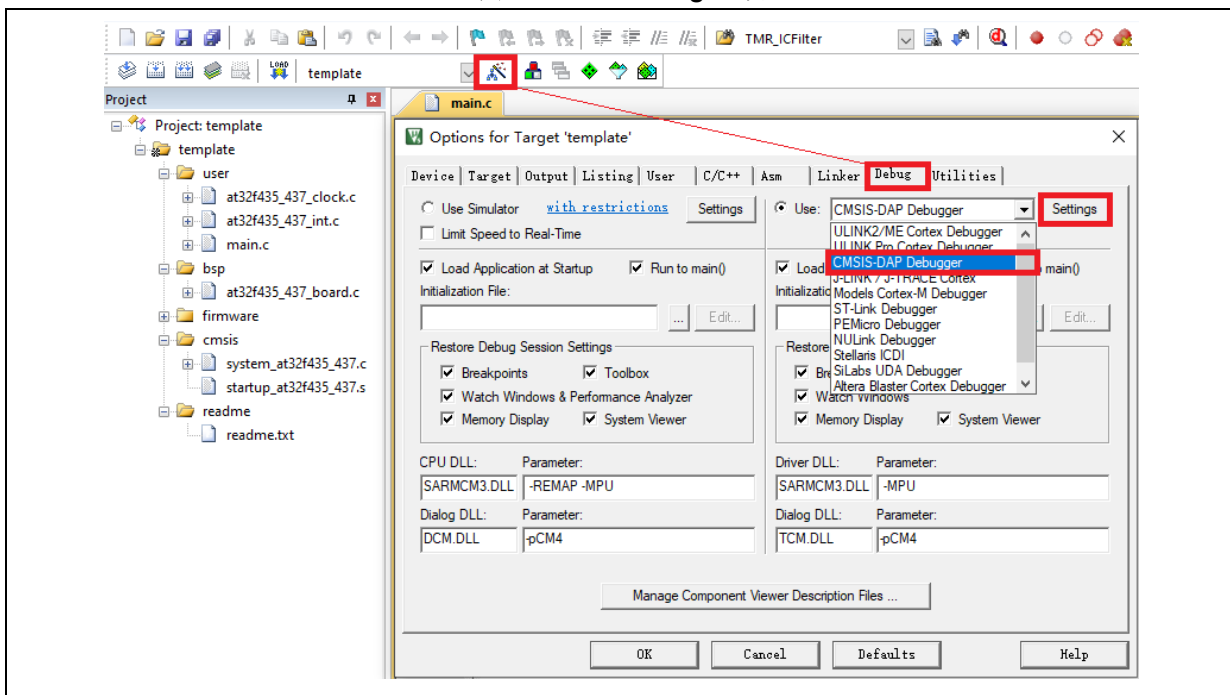


注意：关于 Pack 安装的更多详细说明，请参考《AT32F435_437 固件库 BSP&Pack 应用指南》的“2 Pack 安装步骤”章节，存放路径为[雅特力科技官方网站](#)→产品讯息→高性能 MCU→AT32F4xx 系列下载的BSP 解压后AT32F435_437_Firmware_Library_Vx.x\document。

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



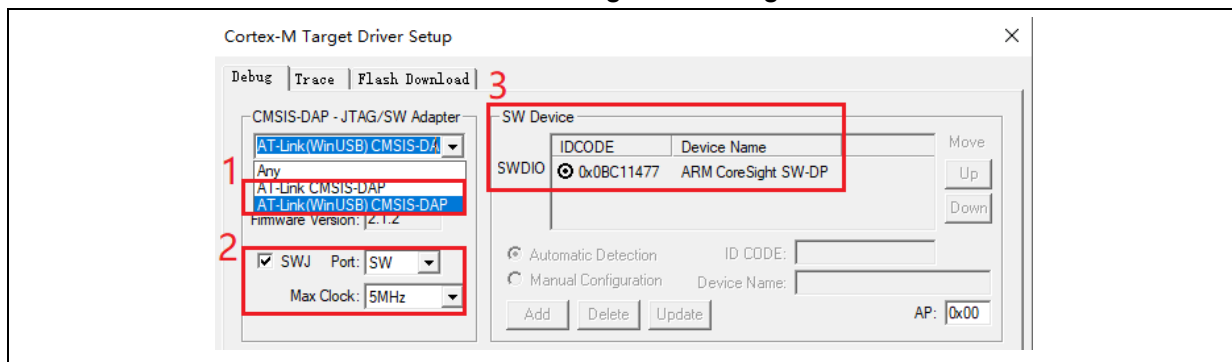
在 Debug 单击 Settings 进入 Cortex-M Target Driver Setup 界面如下图，

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。

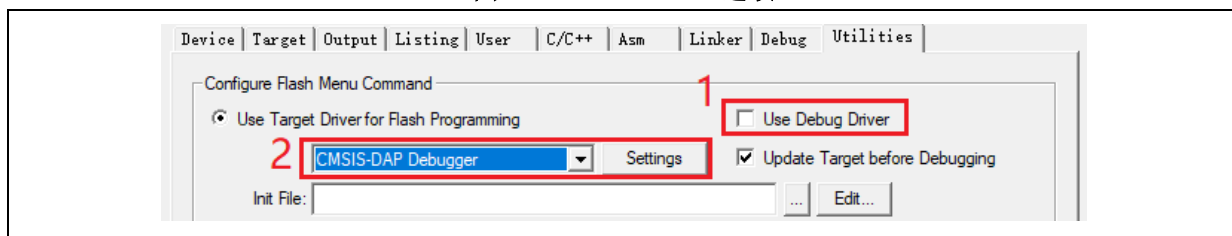
2. Port 选择 SW，再勾选 SWJ；
3. 确认已识别到 ARM SW-DP 调试模块。

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

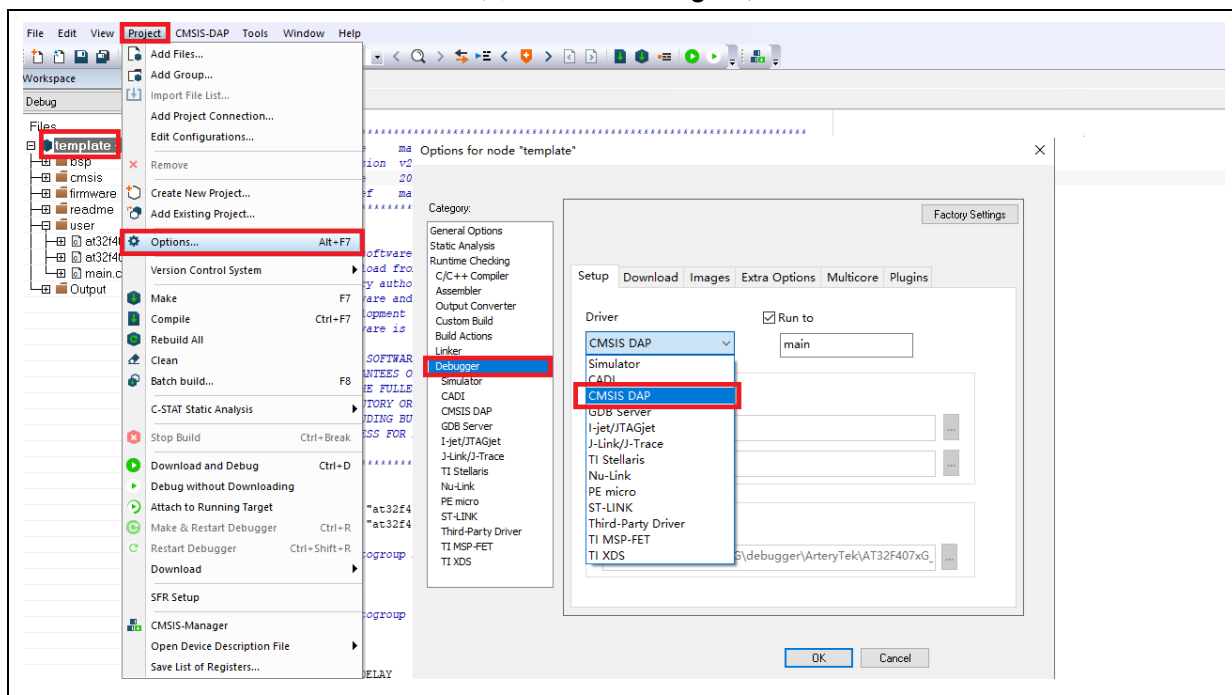
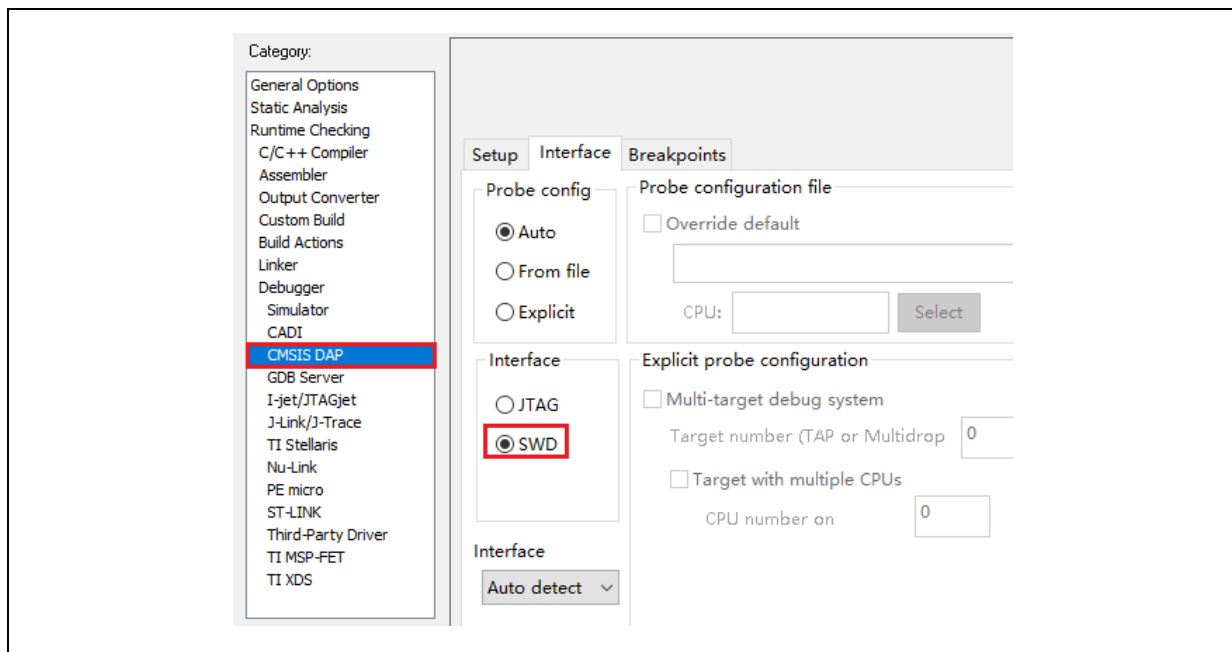


图 15. IAR CMSIS-DAP 选项



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《AT32F435_437 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→高性能MCU→AT32F4xx 系列下载的BSP 解压后AT32F435_437_Firmware_Library_Vx.x\document。

1.1.4 快速替代 AT32F403A/407 流程

- 请参考《MG0018_从 AT32F403A_407 移植到 AT32F435_437》，该文档可从[雅特力科技官方网站](#)→产品讯息→高性能MCU→AT32F4xx 系列获取；
- 如果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。

1.2 AT32F435 /AT32F437 芯片的增强功能配置

1.2.1 PLL 时钟设置

1.2.1.1 PLL 设定方式

AT32F435 /AT32F437 内置的 PLL 最高可输出 288MHz 时钟，须根据输出频率设定 PLL 时钟配置寄存器（CRM_PLLCFG）。

$$\text{PLL 输出时钟} = \text{PLL 参考输入时钟} \times \frac{\text{PLL 倍频系数 PLL_NS}}{\text{PLL 预分频系数 PLL_MS} \times \text{PLL 后分频配置值 PLL_FR}}$$

例如 PLL=288MHz 时(HEXT=8MHz)，设置如下：

```
/*< config pll clock */
crm_pll_config(CRM_PLL_SOURCE_HEXT, 72, 1, CRM_PLL_FR_2);
```

其中，入口参数 CRM_PLL_SOURCE_HEXT 代表选择 HEXT 为外部时钟源，72 为 PLL_NS 值，1 为 PLL_MS 值，CRM_PLL_FR_2（0x01，二分频）为 PLL_FR 值。

更多时钟配置相关信息，请参考《AN0084_AT32F435_437_CRM_Start_Guide》。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→AN0084 获取，其中详细介绍了 AT32F435/437 的时钟源码配置与修改，以及如何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→高性能 MCU→AT32F4xx 系列获取。

1.2.1.2 PLL 自动滑顺

当 AT32F435 /AT32F437 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 AT32F435 /AT32F437 BSP 时 PLL 自动滑顺频率切换程序范例：

图 16. 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。

1.2.1.3 Flash 时钟分频

Flash 时钟分频系数与系统时钟频率相对应，系统时钟频率与 Flash 分频系数对应关系如下：

System Clock Frequency	Flash Clock Division
<= 240 MHz	FLASH_CLOCK_DIV_2
>240 MHz	FLASH_CLOCK_DIV_3

代码实现如下：

```
flash_clock_divider_set(FLASH_CLOCK_DIV_3); /* 设置 flash 分频系数为系统时钟 3 分频 */
```

1.2.2 如何打开 FPU 功能（硬件浮点运算单元）

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 AT32F435 /AT32F437 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。

AT32F435/ AT32F437 的 SARM 大小默认是 384KB，通过修改用户系统数据的扩充系统选项 EOPB 0 来对其 SARM 在最低 128KB 字节到最高 512KB 字节之间动态配置。使能 EOPB0 有效务必要掉电或 RESET 一次。

AT32F435/ AT32F437 的用户系统数据说明如下：

地址	位	内容
0x1FFF_C010	[7: 0]	EOPB0[7: 0]: 扩充的系统选项 不同闪存容量略有不同，见下表扩充的系统选项说明。 <i>注意：位改写只能在安全库区未启动的情况下实现</i>
	[15: 8]	nEOPB0[7: 0]: EOPB0[7: 0]的反码
	[31: 16]	保留不用

EOPB0[7: 0]: 扩充的系统选项

256K 闪存容量	位 1: 0	00: 片上 SRAM 512K 字节 + 闪存零等待延迟区域 128K 字节 01: 片上 SRAM 448K 字节 + 闪存零等待延迟区域 192K 字节 10、11: 片上 SRAM 384K 字节 + 闪存零等待延迟区域 256K 字节 <i>注意：位 1~0 的改写只能在安全库区未启动的情况下实现</i>
	位 7: 2	保留不用
1024K 及以上 闪存容量	位 2: 0	000: 片上 SRAM 512K 字节 + 闪存零等待延迟区域 128K 字节 001: 片上 SRAM 448K 字节 + 闪存零等待延迟区域 192K 字节 010: 片上 SRAM 384K 字节 + 闪存零等待延迟区域 256K 字节 011: 片上 SRAM 320K 字节 + 闪存零等待延迟区域 320K 字节 100: 片上 SRAM 256K 字节 + 闪存零等待延迟区域 384K 字节 101: 片上 SRAM 192K 字节 + 闪存零等待延迟区域 448K 字节 110、111: 片上 SRAM 128K 字节 + 闪存零等待延迟区域 512K 字节 <i>注意：位 2~0 的改写只能在安全库区未启动的情况下实现</i>
	位 7: 3	保留不用

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。

下文以 AT32F435ZMT7(闪存容量 4032K)为例，着重介绍 SRAM 从 384KB 修改为 512KB 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。

1.2.3.1 使用 Artery ICP Programmer 工具（BOOT0=0,BOOT1=0）

AT-Link-EZ /AT-Link /J-Link 连接到 MCU → 设备操作 → 用户系统数据 → EOPB0 选择 512KB
（如果有其他设置一并设置好） → 应用到设备，即完成修改。

图 17. ICP 工具编辑用户系统数据选择 SRAM 大小

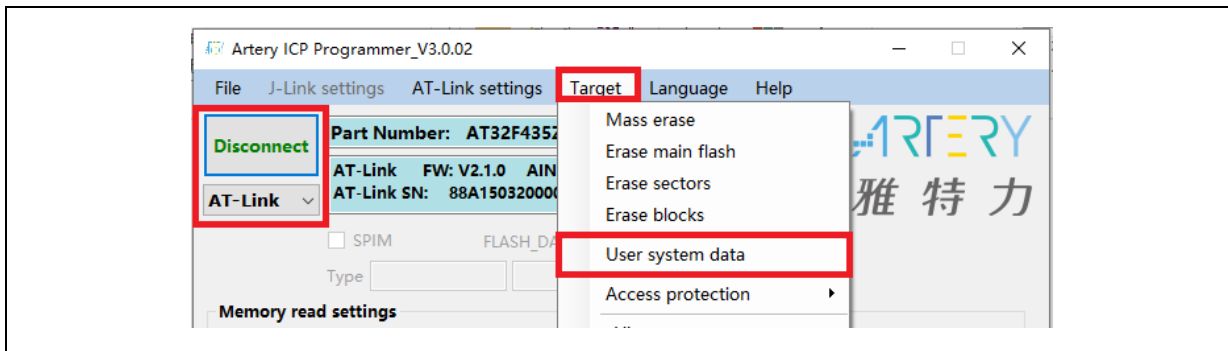
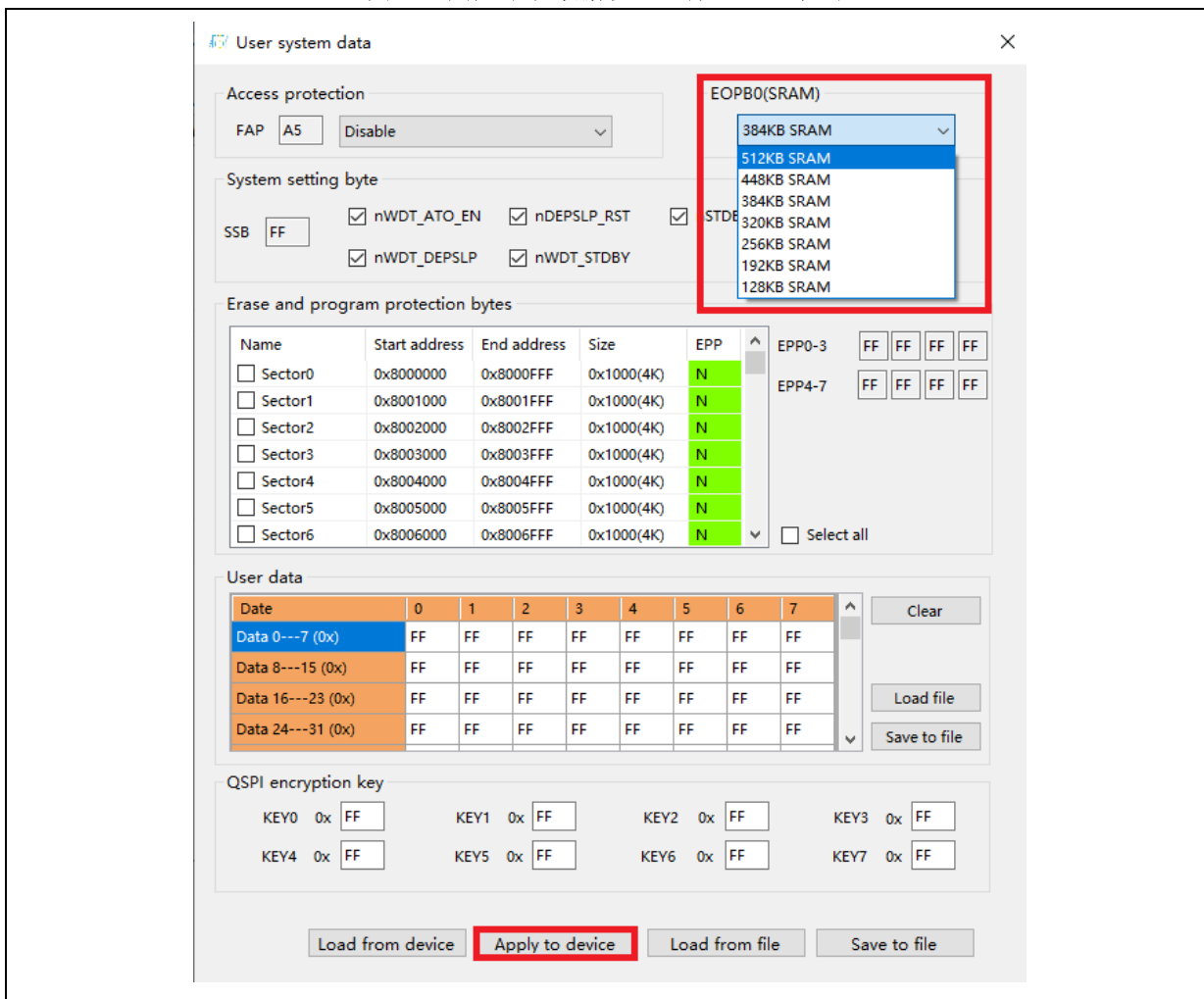


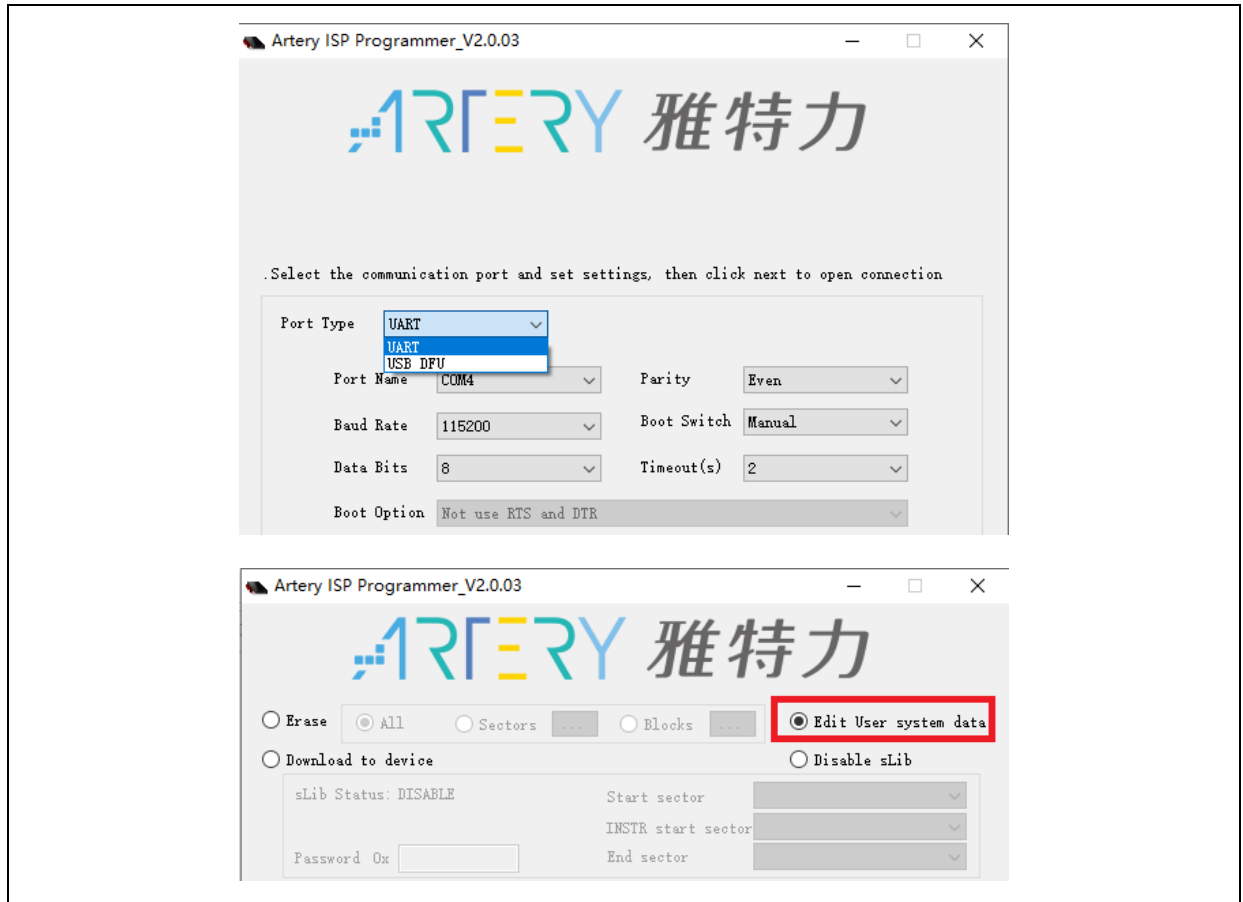
图 18. 用户系统数据设置选择 SRAM 大小



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

UART 或者 USB 连接到 MCU → 一直下一步到如下界面 → 选择编辑用户系统数据 → 下一步 → EOPB0 选择 512KB（如果有其他设置一并设置好） → 应用到设备，即完成修改。

图 19. ISP 工具编辑用户系统数据选择 SRAM 大小



1.2.3.3 在 IAP 中修改

用户也可以使用在 Bootloader program (IAP) 内修改 SRAM 的大小。需要注意的是，编译器里需要将 SRAM 大小设置为与修改后 SRAM 大小一致（参考 AN0026）。

在运行启动文件时会加载 SRAM，如果应用程序使用的 SRAM 大于 384KB，就要在 IAP 修改用户系统数据，可以修改为 512KB，SRAM 大小用户系统数据 EOPB0 的地址是 0x1FFFC010，示例代码如下：

图 20. 定义 Extend_SRAM(void)函数修改 SRAM 大小

```
#define SRAM_384K 0x2
#define SRAM_512K 0x0

static uint32_t f_eopb0;
f_eopb0=(uint32_t*)(0x1FFFC010);
void Extend_SRAM(void)
{
    if((f_eopb0 & 0x07) == SRAM_384K) // check if RAM has been set to 384K, if yes, change EOP
B0
    {
        /* Unlock User System Data Program Erase controller */

```

```
flash_unlock();

/* Erase User System Data */
flash_user_system_data_erase();

/* Change SRAM size to 512KB */
flash_user_system_data_program(0x1FFFC010, SRAM_512K);

/*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，需要将其其他设置读出来，擦除后和设置 SRAM 大小一并写入。

1.2.3.4 在启动文件中修改

在运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 384KB，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 大小设置为 512KB。

在 Keil 编译环境启动文件中增加如下加粗部分代码：

图 21. 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部分代码：

图 22. 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 1.2.3.3 中写法，定义 `Extend_SRAM(void)` 函数修改 SRAM 大小。

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)

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。

图 23. ISP 工具启用访问保护

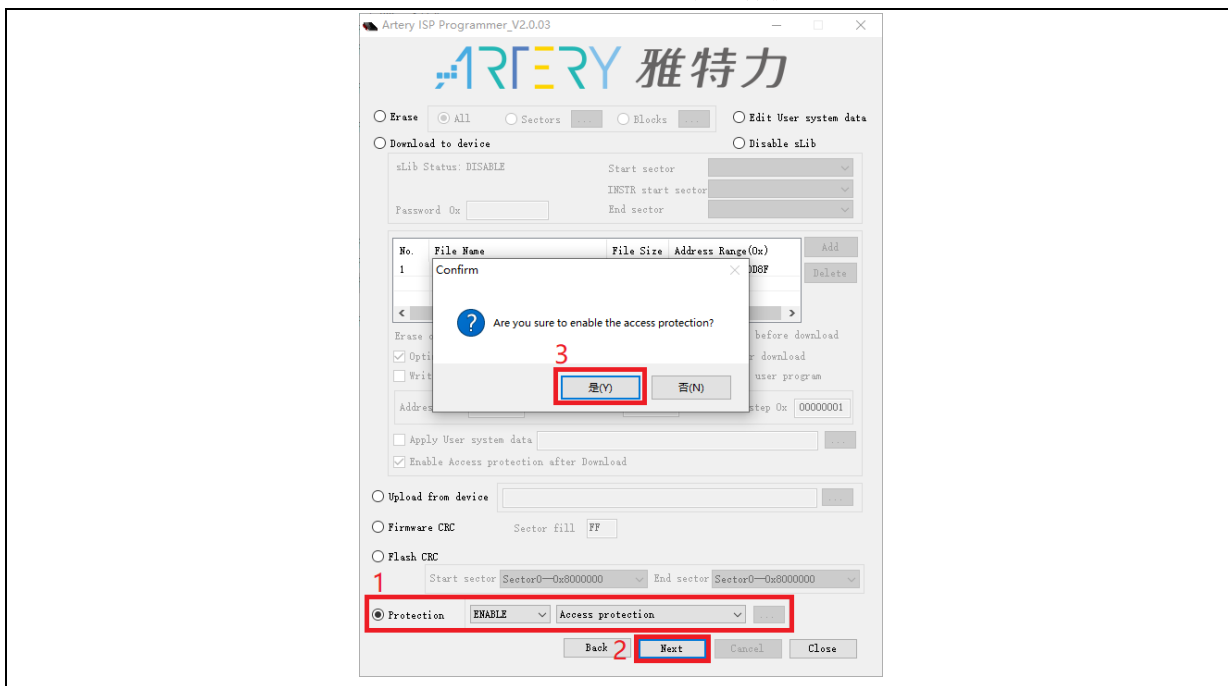
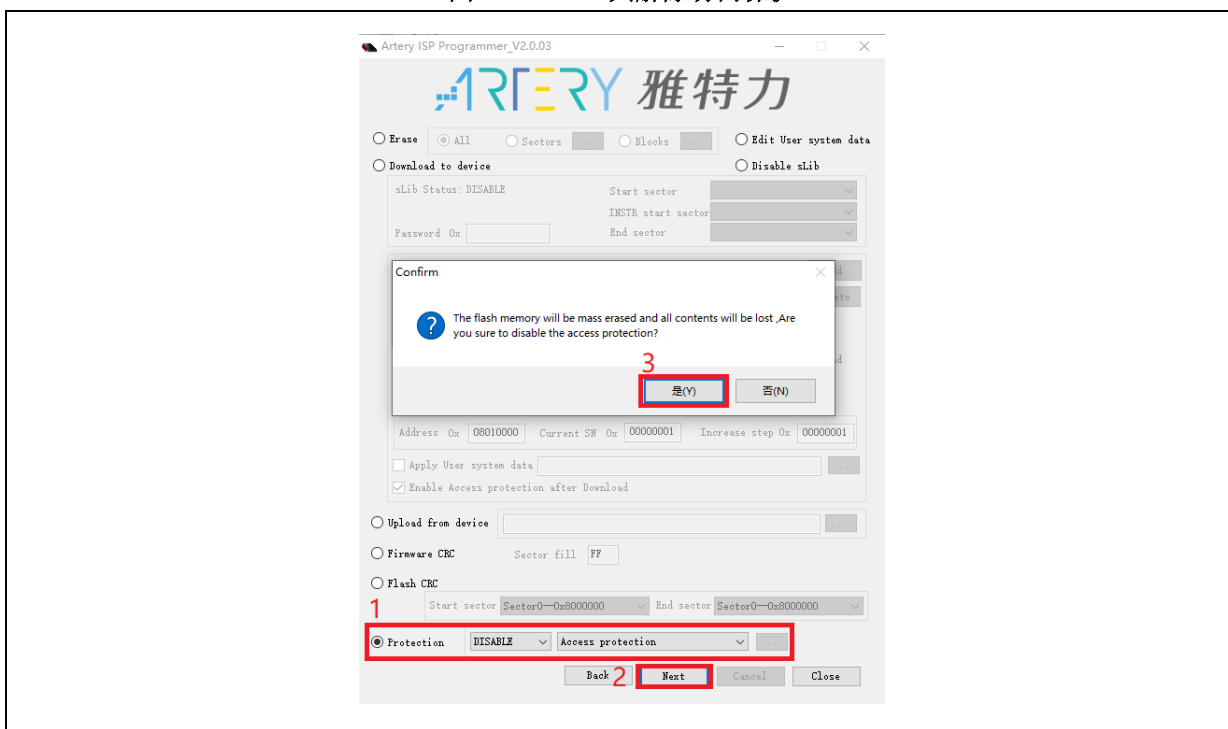


图 24. ISP 工具解除访问保护



■ 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。

■ 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 25. ICP 工具启用擦写保护

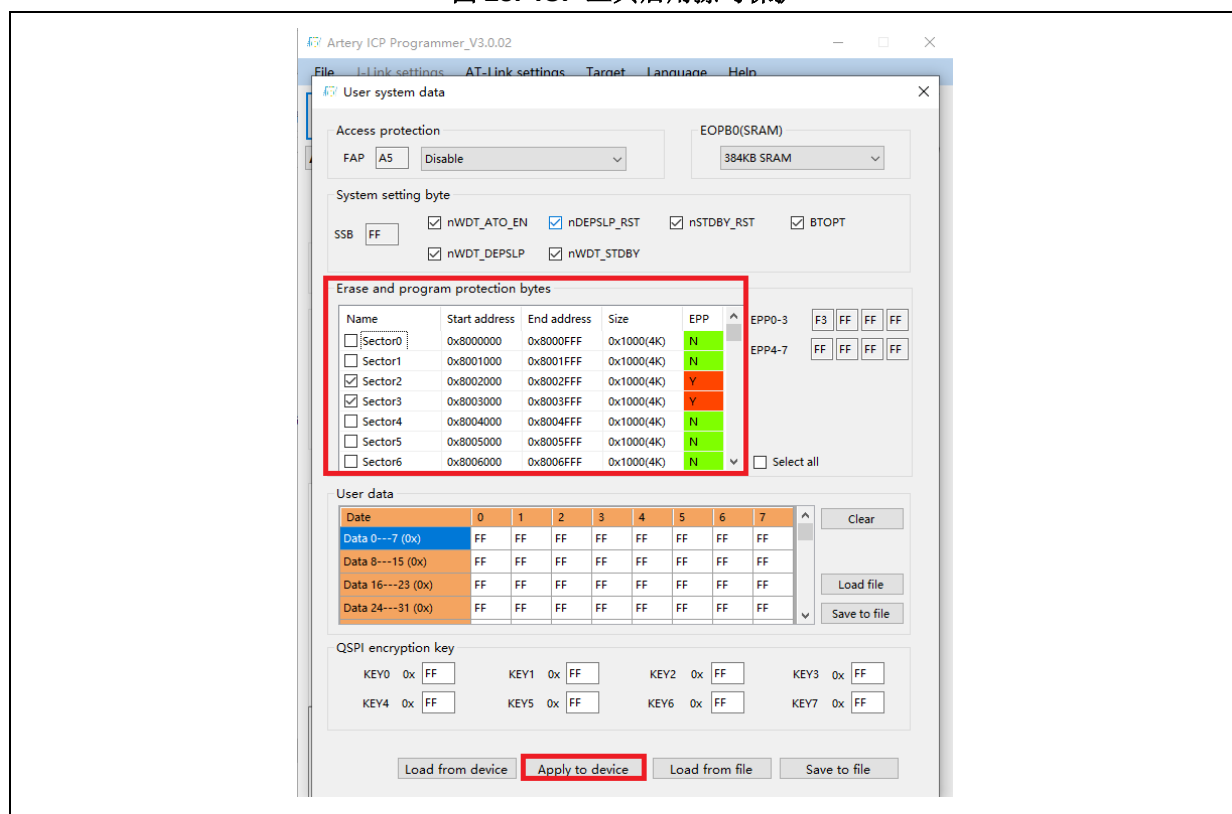
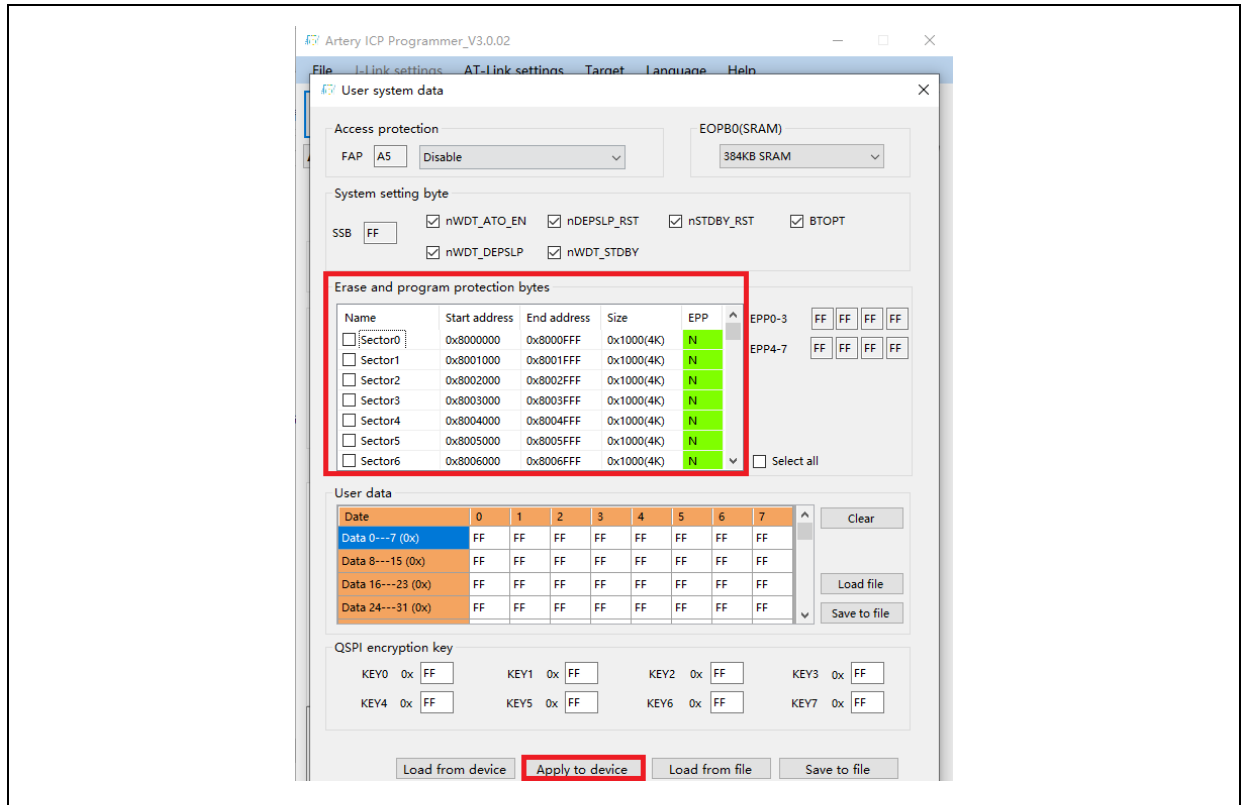


图 26. ICP 工具解除擦写保护



■ 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。

1.2.5 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 27. 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 UID,PID 方式区分

图 28. 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};

/* AT32F435 MCU type table */
const uint64_t AT32_MCU_ID_TABLE[] =
{
    0x0000000D70084540, //AT32F435ZMT7 4032KB LQFP144
    0x0000000D7008454F, //AT32F437ZMT7 4032KB 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)|((uint64_t)ID[4]<<0);

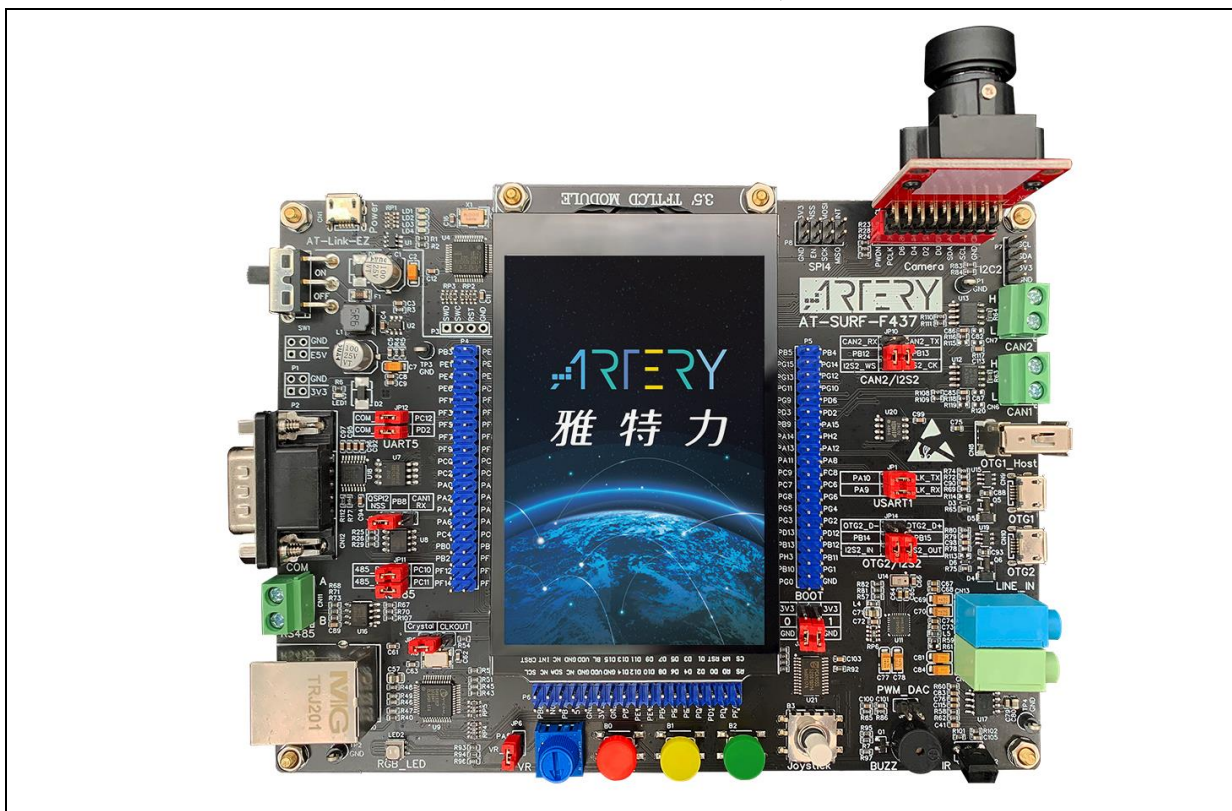
/* 判断 AT32 MCU */
for(i=0;i<sizeof(AT32_MCU_ID_TABLE)/sizeof(AT32_MCU_ID_TABLE[0]);i++)
{
    if(AT_device_id == AT32_MCU_ID_TABLE[i])
    {
        printf("This chip is AT32F4xx.\r\n");
    }
    else
    {
        printf("This chip is Other Device.\r\n");
    }
}
```

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。

1.2.6 AT32F435/AT32F437 高级功能

雅特力科技提供 AT-SURF-F437 开发板, 配套实例涵盖 AT32F437 大部分高级功能, 且提供很多实用级别的程序 (存放于 BSP 例程\project\at_surf_f437\examples 目录), 详细信息请参考《AN0049_AT_SURF_F437_Board_Application_Note》, 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→AN0049 获取。

图 29. AT-SURF-F437 开发板



注意：系统性能改善是多方面调优共同作用的结果，关于如何提高AT32F435/AT32F437的运行效能，请参考应用手册《AN0004_Performance_Optimization》与《AN0092_AT32F435_437_Performance_Improve》，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/AN0092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 30. 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

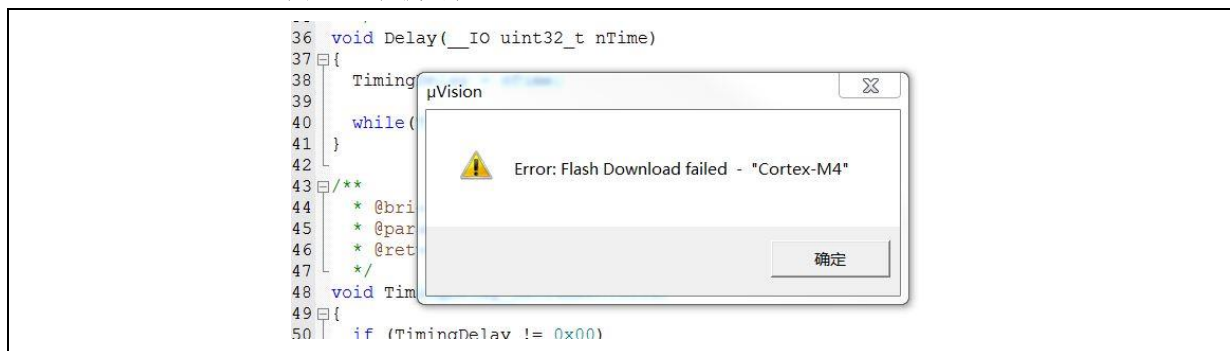
- 请参考《FAQ0008_Keil 项目内 Jlink 无法找到 IC 问题》，该文档可以从[雅特力科技官方网站](#)→技术与开发支持→FAQ→FAQ0008 下载。
- 请参考《FAQ0132_JLink 手动添加 Artery MCU》，该文档及其附件可以从[雅特力科技官方网站](#)→技术与开发支持→FAQ→FAQ0132 下载。

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 31. 下载出现 Flash Download failed-“Cortex- M4”



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 AT32F435 /AT32F437 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 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（注意切回Boot[1:0]=00b）。同理，ISP下载也可恢复。
- 方法2 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。

3 安全库区 sLib(Security Library)

3.1 概述

目前越来越多的微控器(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。

因为这一重要的需求，AT32F435/437 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。
- 安全库区划分为唯读区(SLIB_READ_ONLY)及指令区 (SLIB_INSTRUCTION)，并可选择部分或是整个安全库区存放唯读区或者指令区。
- 唯读安全库区 (SLIB_READ_ONLY)的数据能透过 I-Code和 D-Code总线读取，不能写入。
- 指令安全库区 (SLIB_INSTRUCTION)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 ISP/ICP/调试模式以及从内部 RAM启动的程序），以读取数据的方式去访问 SLIB_INSTRUCTION时，读到的数值全都是0xFF。
- 安全库区的程序代码及数据，除非输入正确的密码，否则无法被擦除。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 EPPERR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_PWD_CLR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《AN0081_AT32F435_437_Security_Library_Application_Note》，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→AN0081 获取。

4 文档版本历史

表 1. 文档版本历史

日期	版本	变更
2022.04.20	2.0.0	最初版本
2022.07.08	2.0.1	优化部分表述
2022.10.11	2.0.2	更新第三方烧录工具支持，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

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。

© 2022 • 雅特力科技 • 保留所有权利