

前言

本应用入门指南旨在让用户快速使用 AT32F421xx 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	AT32F421 系列
------	-------------

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	AT32 Work Bench.....	13
1.1.5	快速替代 SXX 流程.....	20
1.2	AT32F421 芯片的增强功能配置.....	20
1.2.1	预取指令缓冲	20
1.2.2	PLL 时钟设置.....	21
1.2.3	加密方式	23
1.2.4	设定系统存储器为扩展主存	26
1.2.5	在程序中区分 AT32 与其他 IC 方法.....	29
2	下载编译过程常见问题	32
2.1	程序启动进入 Hard Fault Handler	32
2.2	Keil 项目内 Jlink 无法找到 IC	32
2.3	程序下载过程出问题	32
2.3.1	显示 Error: Flash Download failed-“Cortex-M4”问题.....	32
2.3.2	显示 No Debug Unit Device found 问题	32
2.3.3	显示 RDDI-DAP Error 问题.....	33
2.3.4	ISP 串口下载时卡死问题	33
2.3.5	AT32 恢复下载.....	33
3	安全库区 sLib(Security Library)	35
3.1	概述	35
3.2	应用原理.....	35
3.3	操作安全库区.....	35

4 文档版本历史..... 36

表目录

表 1. 文档版本历史 36

图目录

图 1. AT-START-F421 及 AT-Link-EZ 实物图.....	7
图 2. 雅特力科技官方网站 AT-START-F421 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.AT32F421 _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. 雅特力科技官方网站 AT32 Work Bench 文档与软件	13
图 17. 终端命令安装	14
图 18. 图形方式安装	14
图 19. AT32 Work Bench 引导页	15
图 20. 项目配置界面	15
图 21. “模式和配置”窗口	16
图 22. 外设模式	16
图 23. 外设参数设置	16
图 24. 外设 GPIO 设置.....	17
图 25. 外设 DMA 设置.....	17
图 26. 外设 NVIC 设置.....	17
图 27. 左击和右击效果	18
图 28. 时钟配置界面	18
图 29. 代码预览	19
图 30. “项目管理”窗口.....	19

图 31. 项目文件结构	19
图 32. 闪存性能选择寄存器 (FLASH_PSR) 等待周期	20
图 33. 系统时钟配置函数 system_clock_config	20
图 34. AT32F421 输出 118MHz 的时钟配置	21
图 35. SXX 程序 PLL 自动滑顺频率切换配置	22
图 36. AT32 程序 PLL 自动滑顺频率切换配置	22
图 37. ICP 工具启用/解除访问保护	23
图 38. ISP 工具启用访问保护	24
图 39. ISP 工具解除访问保护	24
图 40. ICP 工具启用擦写保护	25
图 41. ICP 工具解除擦写保护	26
图 42. ICP 工具设定系统存储区 AP 模式	27
图 43. ICP 工具设定系统存储区 AP 模式确认界面	27
图 44. ICP 工具离线烧录系统存储区 AP 模式	28
图 45. ICP 工具离线烧录项目文件设置	29
图 46. ICP 工具离线下载状态监控	29
图 47. 读取 Cortex 型号	30
图 48. 读取 UID,PID	30
图 49. 下载出现 Flash Download failed-“Cortex- M4”	32
图 50. debug 界面修改	33
图 51. CMSIS DAP 界面修改	34

1 雅特力初步环境准备

雅特力开发环境下载地址：

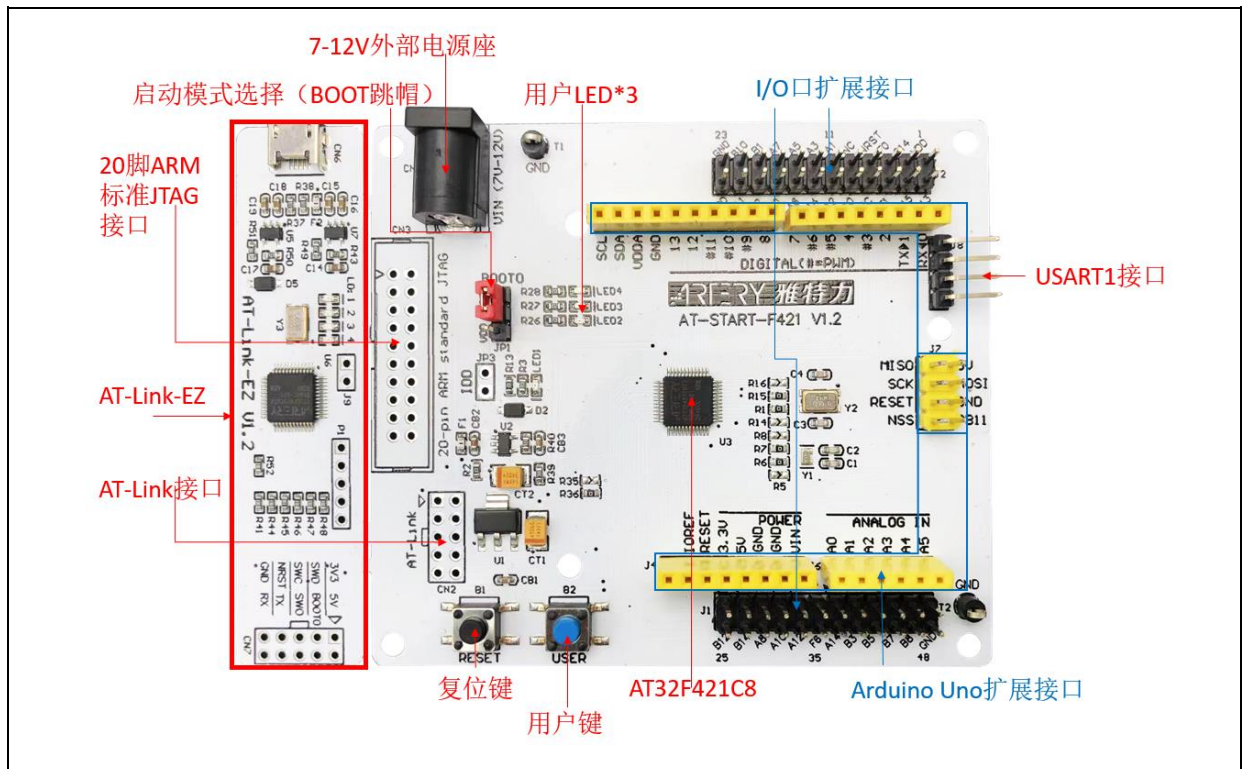
- 雅特力科技官方网站：<http://www.arterytek.com>

1.1 搭建 AT32 开发环境

1.1.1 调试工具及开发板

目前 AT32F421 支持的调试工具有 AT-Link/J-Link，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. AT-START-F421 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_F421_Vx.x》，存放路径为雅特力科技官方网站→产品讯息→超值型 MCU→AT32F4xx 系列下载的 Evaluation Board (开发板)资料包解压后AT_START_F421_Vx.x\03_Documents。

图 2. 雅特力科技官方网站 AT-START-F421 开发板资料包

Evaluation Board (开发板)		
Download	Description	Version
AT-START-F421	AT32F421开发板并支持Arduino标准接口	V1.2

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→超值型 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→超值型 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→超值型 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

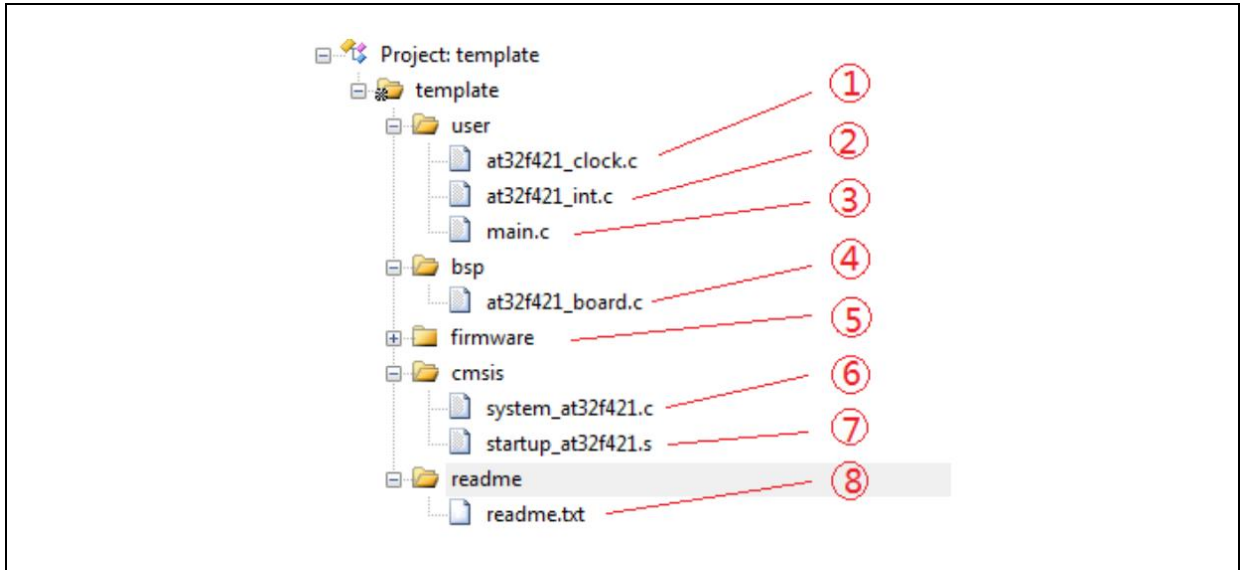
在 ArteryTek 提供的固件库 BSP 中都默认建立好了常用 IDE 的模板工程。BSP 可从[雅特力科技官方网站](#)→产品讯息→超值型 MCU→AT32F4xx 系列获取。

图 4. 雅特力科技官方网站 BSP 资料包

BSP		
Download	Description	Version
Firmware Library	支持AT32F421底层驱动源代码，以固件库(Firmware Library)呈现	V2.0.9

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 AT32F421_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 at32f421_clock.c 时钟配置文件，设置了默认的时钟频率及时钟路径。
- ② at32f421_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 at32f421_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 at32f421_xx.c 是各片上外设的驱动文件。
- ⑥ system_at32f421.c 系统初始化文件。
- ⑦ startup_at32f421.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 AT32F421_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《AT32F421 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→超值型 MCU→AT32F4xx 系列下载的 BSP 解压后AT32F421_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→超值型 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.AT32F421_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.AT32F421 _DFP

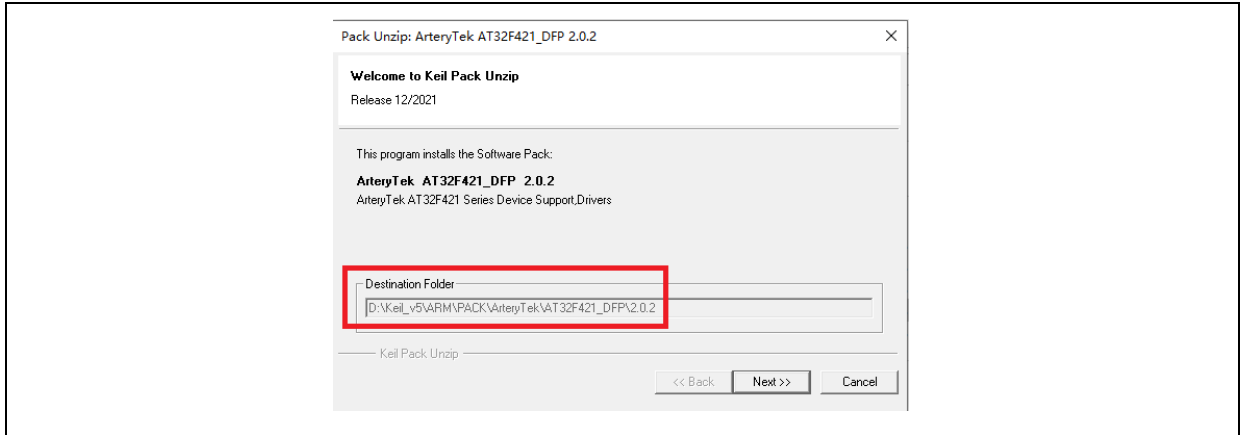
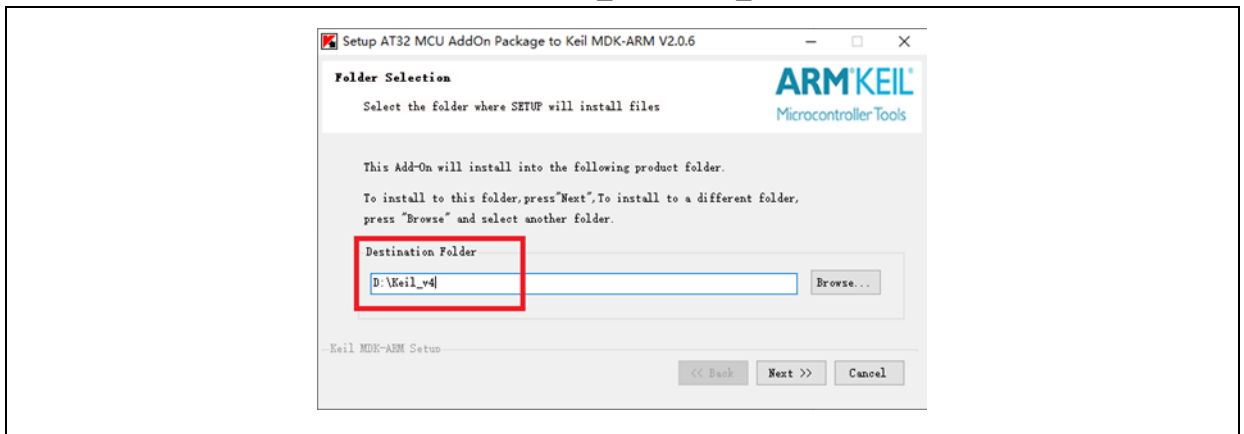
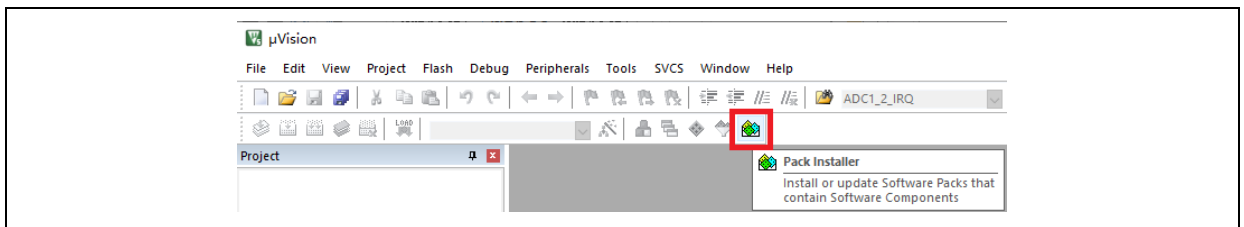


图 8. 安装 Keil4_AT32MCU_AddOn



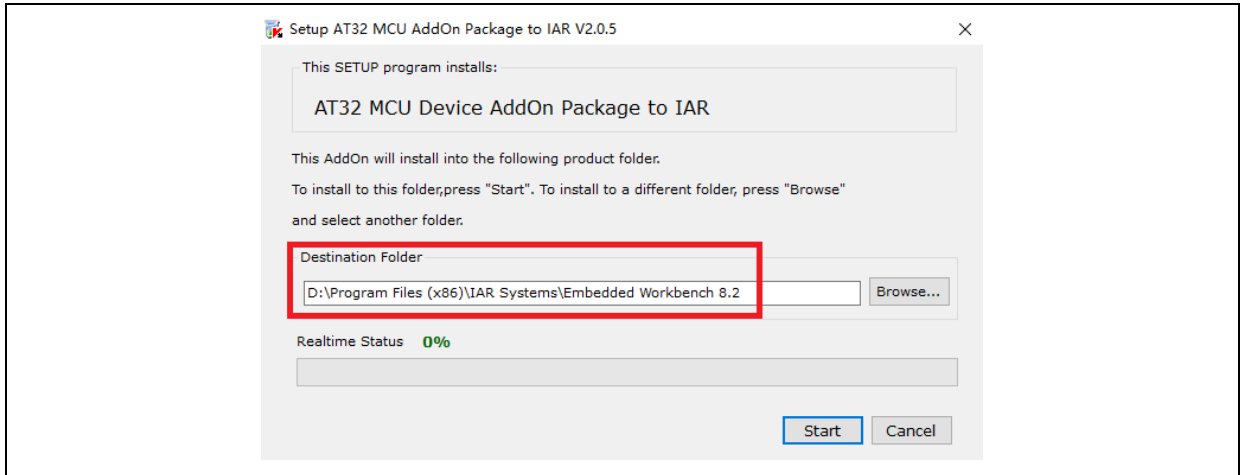
也可打开 keil，点击 Pack Installer 图标，在 Pack Installer 中点击左上角 file，选择 import，导入从 [雅特力科技官方网站](#) 下载好的对应 pack 包完成安装。

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

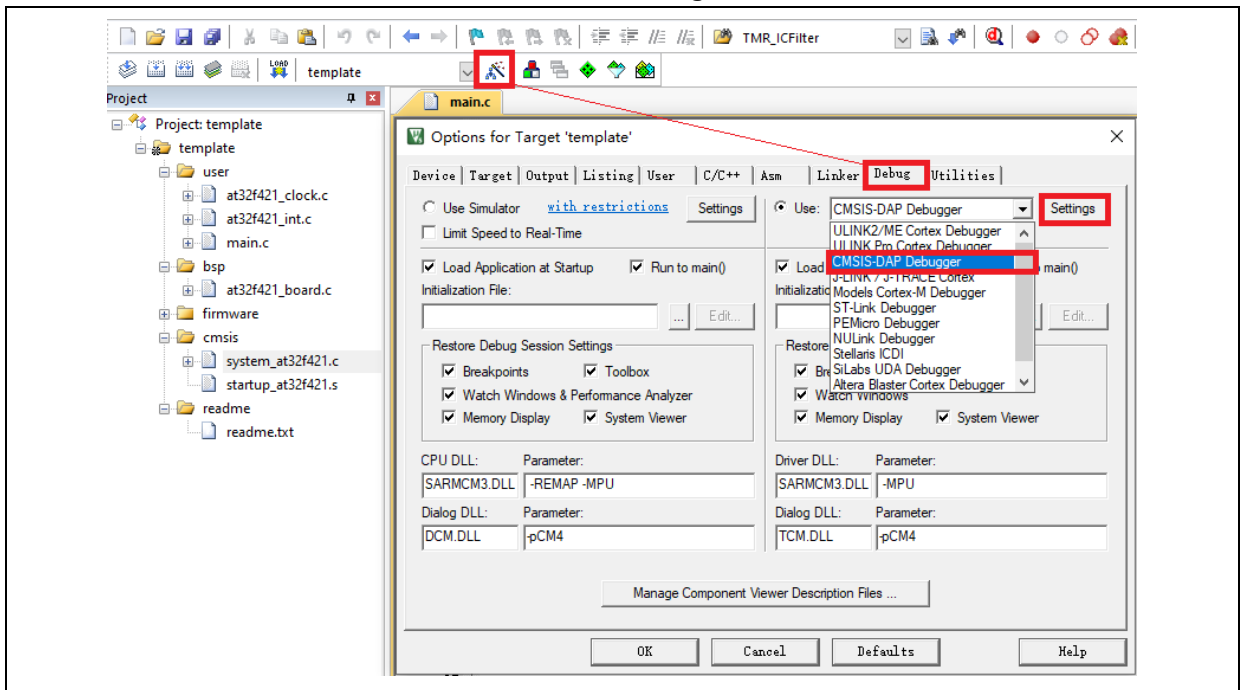


注意：关于 Pack 安装的更多详细说明，请参考《AT32F421 固件库 BSP&Pack 应用指南》的“2 Pack 安装步骤”章节，存放路径为[雅特力科技官方网站](#)→产品讯息→超值型 MCU→AT32F4xx 系列下载的 BSP 解压后 AT32F421_Firmware_Library_Vx.x.x\document。

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



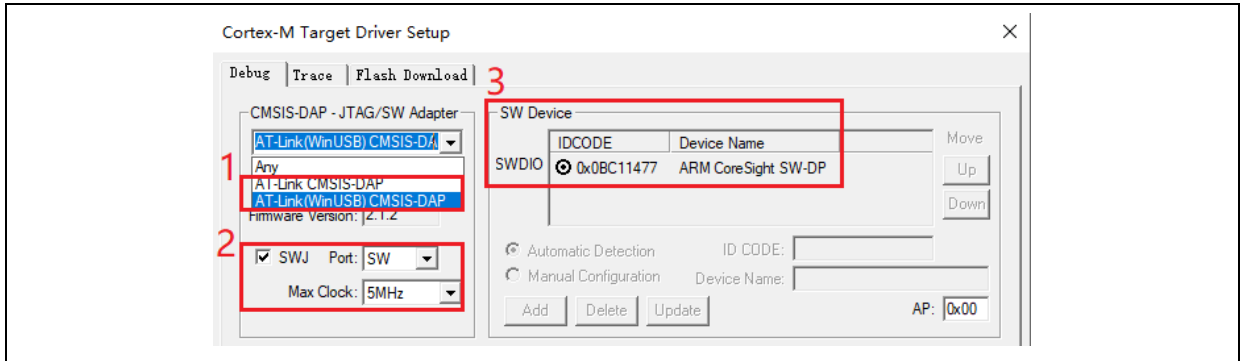
在 Debug 单击 Settings 进入 Cortex-M Target Driver Setup 界面如下图，

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。

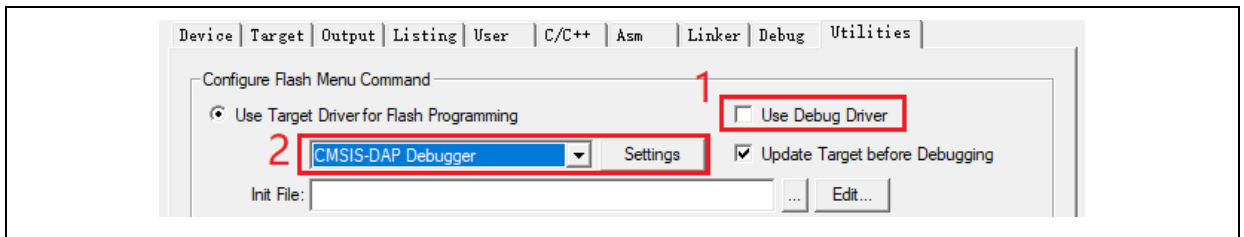
2. Port 选择 SW，再勾选 SWJ；
3. 确认已识别到 ARM SW-DP 调试模块。

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

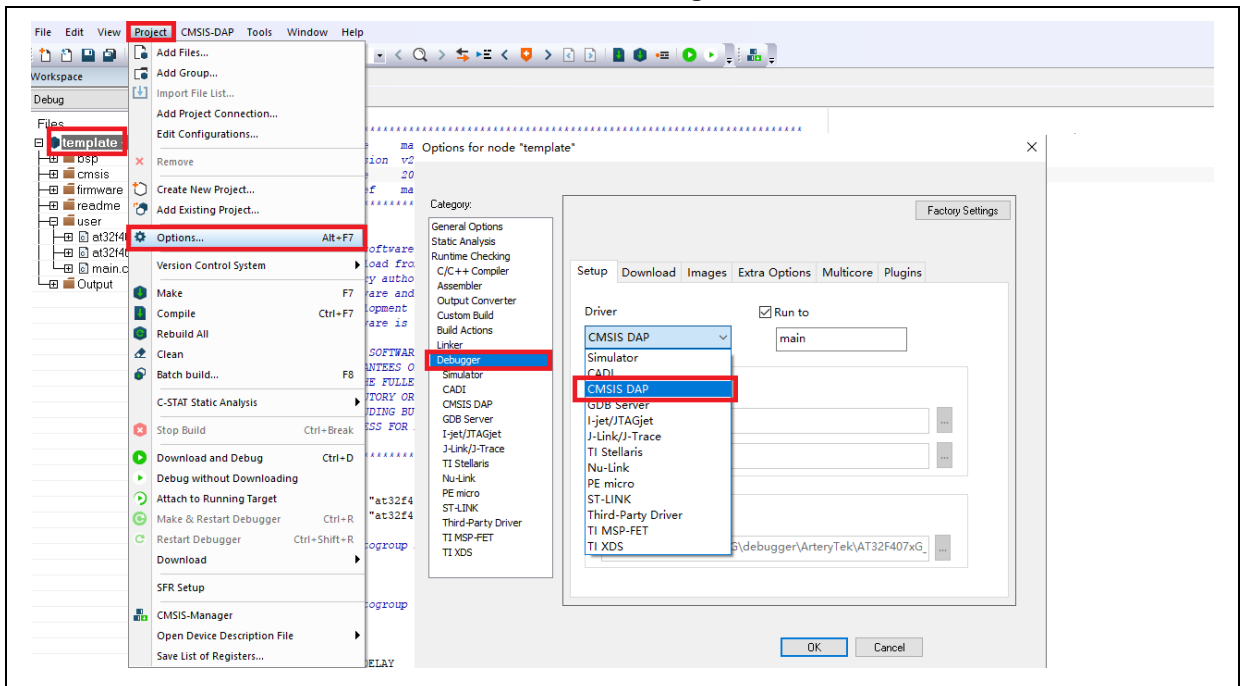
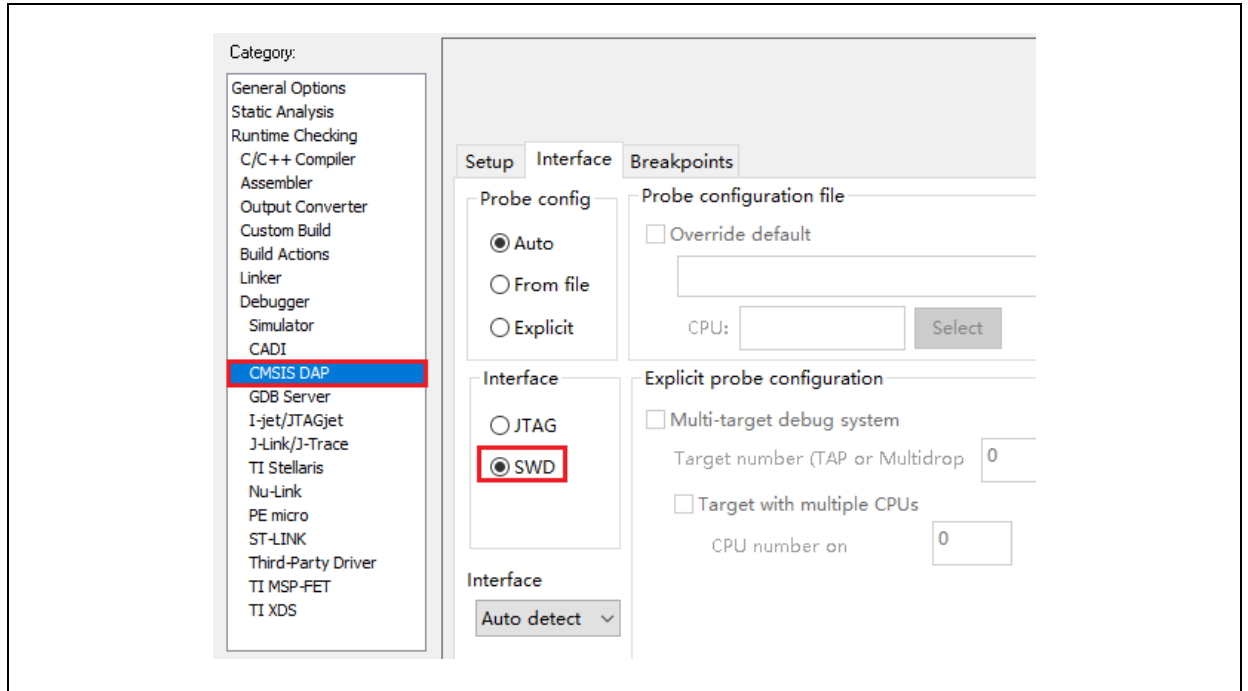


图 15. IAR CMSIS-DAP 选项



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《AT32F421 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→超值型 MCU→AT32F4xx 系列下载的 BSP 解压后 AT32F421_Firmware_Library_Vx.x.x\document。

1.1.4 AT32 Work Bench

1.1.4.1 引言

1.1.4 小节用于介绍 AT32 Work Bench 使用。AT32 Work Bench 通过对 MCU 的图形化配置，生成初始化 C 代码和对应 IDE 项目，以减少开发人员的工作量、时间和成本。

■ 环境要求

软件要求：

Windows 系统：需要 Windows 7 及以上操作系统支持。

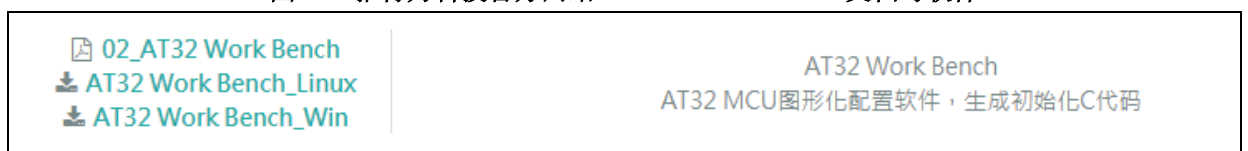
Linux 系统：支持 x86_64 架构的 Ubuntu、Fedora 等发行版。

硬件要求：

建议内存最低 2GB，建议硬盘最低 4GB。

注意：关于 AT32 Work Bench 的详细使用说明，请参阅《UM_AT32_Work_Bench.pdf》文件，其和 AT32 Work Bench 工具一同存放于雅特力官网→技术与开发支持→Tool，链接为 <https://www.arterytek.com/cn/support/index.jsp?index=5>，如下图所示：

图 16. 雅特力科技官方网站 AT32 Work Bench 文档与软件



1.1.4.2 安装

Windows 系统无需安装，只需直接运行可执行程序 AT32_Work_Bench.exe。

Linux 系统目前支持 Ubuntu 16.4 以上系统版本，有两种安装方式：

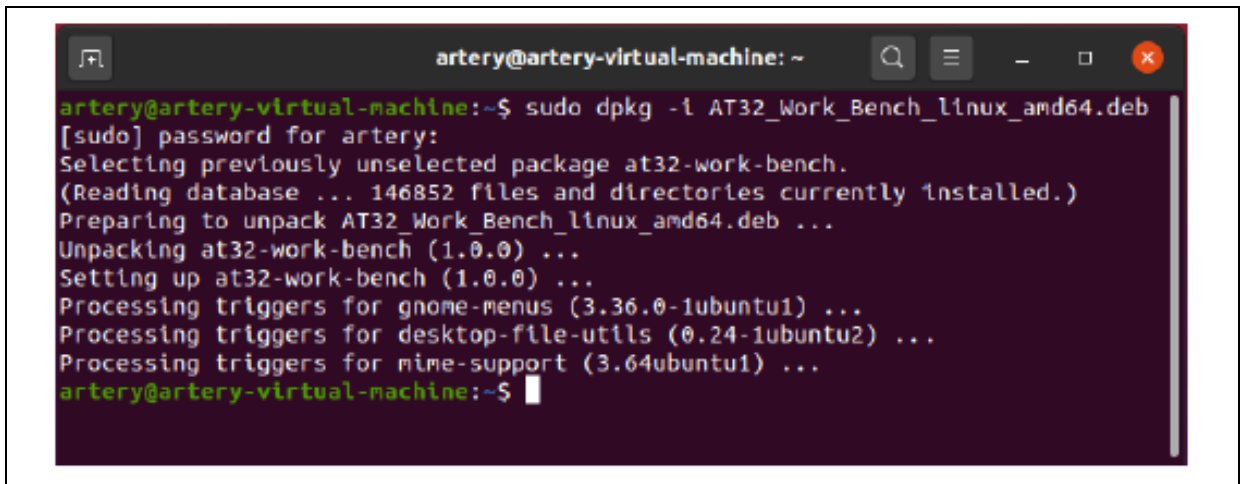
- 终端中使用 dpkg 命令安装

在终端中输入如下命令进行安装：

```
sudo dpkg -i AT32_Work_Bench_Linux-x86_64_Vx.x.xx.deb
```

如下图所示：

图 17. 终端命令安装



- 图形方式安装

拷贝 AT32_Work_Bench_Linux-x86_64_Vx.x.xx.deb 到 Linux 系统中，并双击。在弹出的安装程序中，点击“安装”按钮，系统将本软件自动安装到系统中，如下图所示：

图 18. 图形方式安装



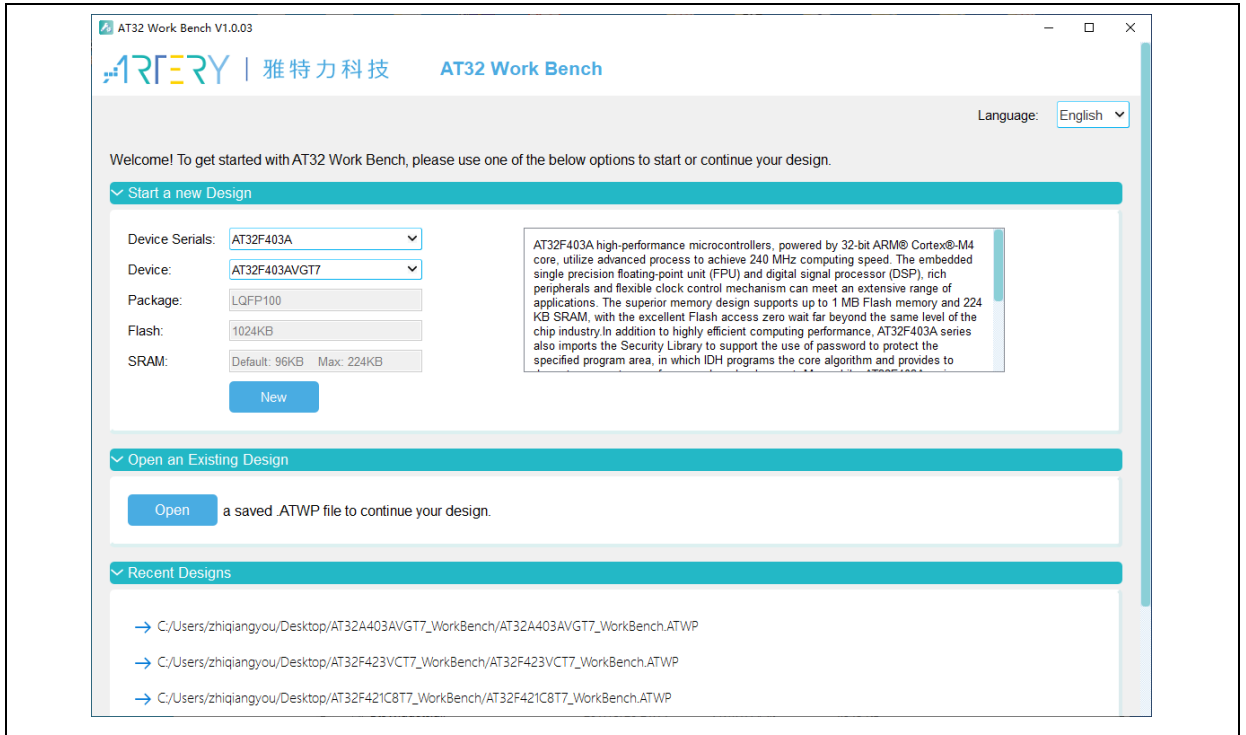
安装完成之后，点击左边任务栏最下方“显示所有程序”按钮，在弹出的程序列表中找到“AT32_Work_Bench”，并点击，将启动 AT32_Work_Bench。

1.1.4.3 项目配置

本小节将以 AT32F403AVGT7 新建一个 USART 工程为例，展示如何使用 AT32_Work_Bench 生成初始化 C 代码和对应 IDE 项目。

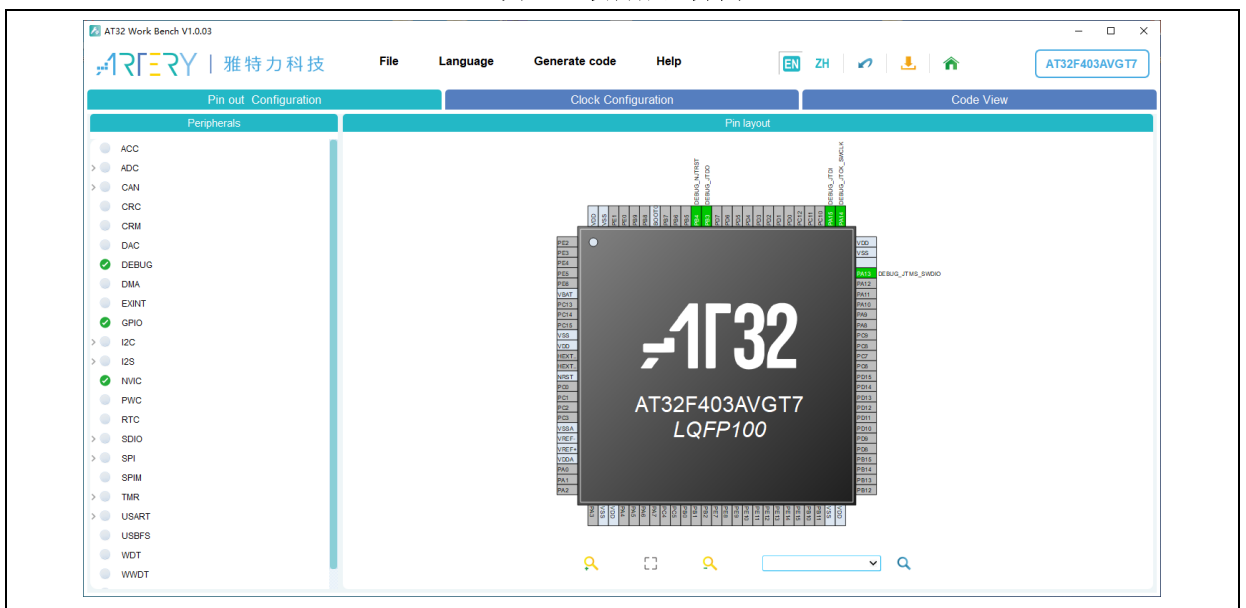
引导页是启动 AT32 Work Bench 时打开的第一个窗口。它提供三个选项：“开始一个新的设计”、“打开一个已有的设计”和“最近使用的设计”，如下图所示：

图 19. AT32 Work Bench 引导页



引导页选择 MCU 的具体信息，然后点击新建进入项目配置界面，项目配置界面可以看到“Pin 布局和配置”，“时钟配置”，“代码预览”三个选项框，这是建立 AT32 Work Bench 工程主要使用到的三个功能，此部分将分（1）（2）（3）三点进行详细说明。项目配置界面如图所示：

图 20. 项目配置界面

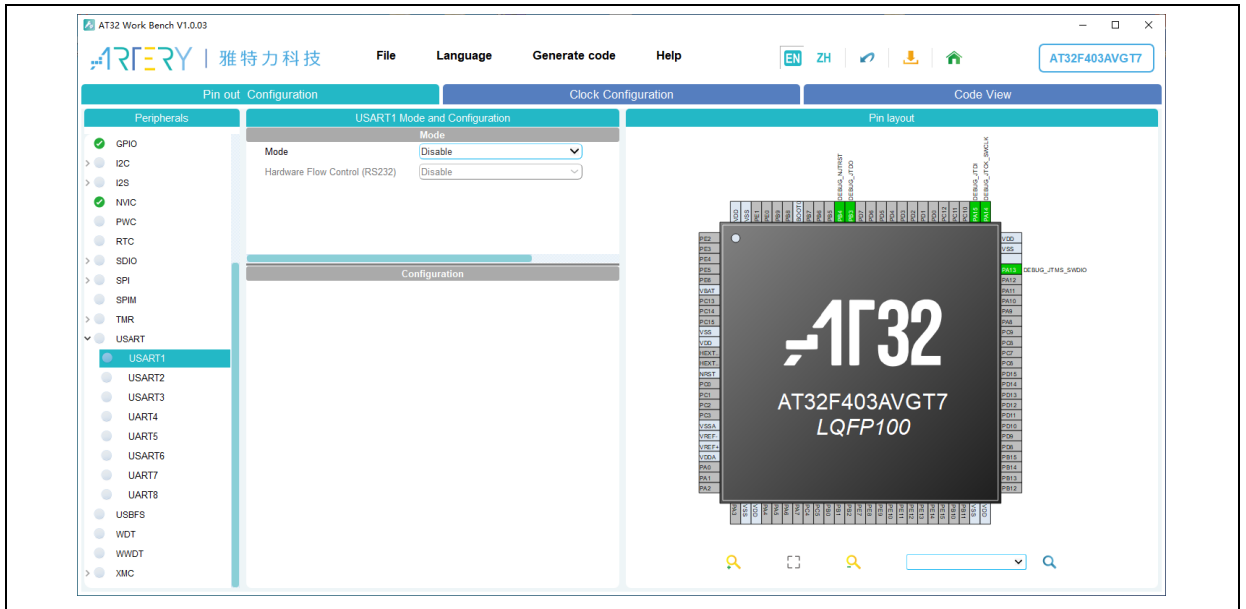


(1) “Pin 布局和配置” 视图主要配置 MCU 使用到的外设和引脚，该视图包含有“外设”、“模式和配置”与“Pin 布局”三个窗口。

从“外设”列表选择一个外设时，将打开所选外设的“模式和配置”窗口，用户可以借此配置其功能模式与参数，模式窗口可选择外设的工作模式和工作模式下使用的 MCU 引脚。

由于 MCU 允许不同外设和多个功能使用相同的引脚，因此选择模式后，会自动配置最适合用户选择的外设的引脚排列，以 USART1 为例，USART1 的“模式和配置”窗口如下图所示：

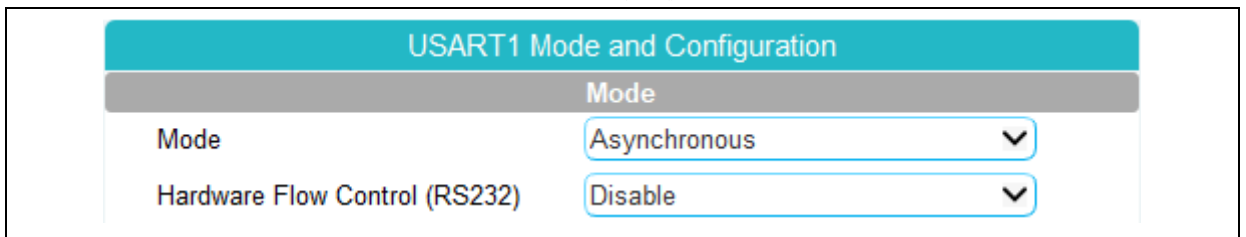
图 21. “模式和配置”窗口



以 USART1 为例，配置 USART1 外设大概分为以下几步：

■ 外设模式

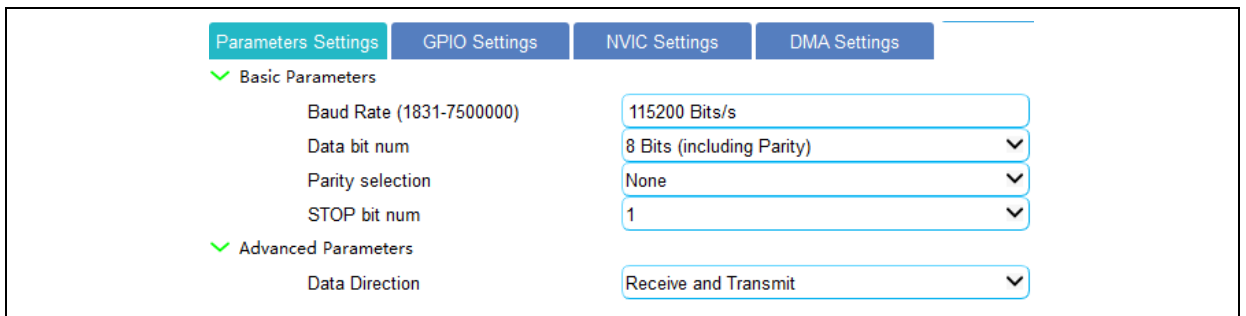
图 22. 外设模式



如上图所示，外设“USART1”的模式选择“异步模式”时，引脚布局的视图中，PA9 与 PA10 将自动映射到信号“USART1_TX”与“USART1_RX”。

■ 外设参数设置

图 23. 外设参数设置



参数配置窗口显示了“USART1”可以配置的参数，如基础参数中的“波特率”、“数据位数”等，

高级参数中的“数据传输方向”等参数。

■ 外设 GPIO 设置

图 24. 外设 GPIO 设置

Pin Name	Signal on Pin	Output level	GPIO type	Pull type	GPIO mode	Driver capability	Label	Modified
PA10	USART1_RX	n/a	n/a	Pull-none	Input mode	n/a		N
PA9	USART1_TX	n/a	Push Pull	Pull-none	Mux function mode	Moderate		N

USART1_RX(PA10)---GPIO Parameters

- Output level: n/a
- GPIO type: n/a
- Pull type: Pull-none
- GPIO mode: Input mode
- Driver capability: n/a
- Label:

外设 GPIO 设置窗口显示了当前外设可以配置的 GPIO，如上图可配置“USART1”中“USART1_TX”和“USART1_RX”的 GPIO 参数。

■ 外设 DMA 设置

图 25. 外设 DMA 设置

DMA Request	Channel	Direction	Priority
USART1_RX	DMA1 Channel 1	Peripheral To Memory	Low
USART1_TX	DMA1 Channel 2	Memory To Peripheral	Low

DMA Request Parameters

- Peripheral Increment: Peripheral Inc Disable
- Memory Increment: Memory Inc Enable
- Peripheral Data Alignment: Byte
- Memory Data Alignment: Byte
- Mode: Normal

外设 DMA 设置窗口可添加当前外设可以配置的 DMA 请求，如上图可配置“USART1”中“USART1_TX”和“USART1_RX”的对应的 DMA 通道以及 DMA 请求的参数。

■ 外设 NVIC 设置

图 26. 外设 NVIC 设置

NVIC Interrupt Table	Enabled	Preemption Priority	Sub Priority
DMA1_Channel1_IRQ	<input checked="" type="checkbox"/>	0	0
DMA1_Channel2_IRQ	<input checked="" type="checkbox"/>	0	0
USART1_IRQ	<input checked="" type="checkbox"/>	0	0

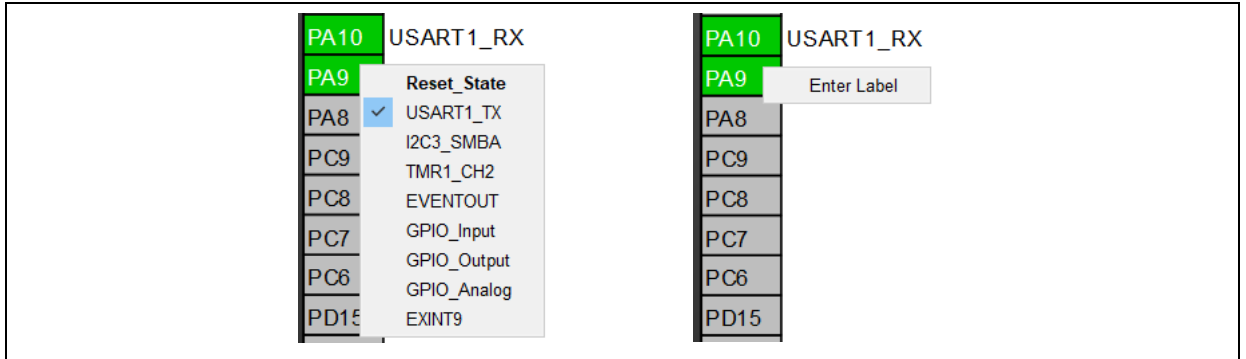
外设 NVIC 设置窗口显示了当前外设可以配置的中断，当外设对应的 DMA 通道开启后，也可配置对应 DMA 通道的中断。“抢占优先级”和“子优先级”的配置需到“NVIC”总配置窗口进行统一配置。

pin 布局窗口以图形方式表示所选封装的引脚布局（例如，LQFP48、QFN32、TSSOP20 等），其中

每个引脚都用其名称（如 PA9）、配置状态和当前信号分配来表示。

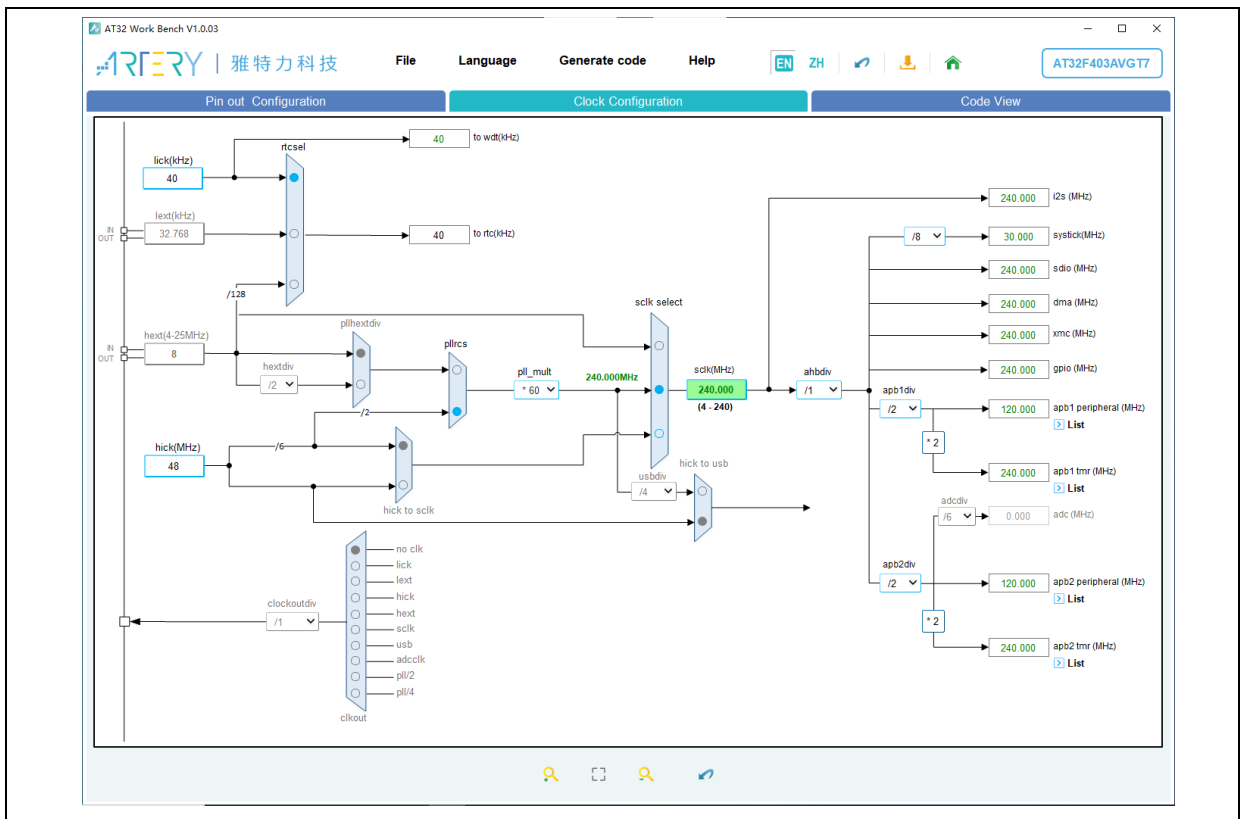
在引脚视图上，左键单击引脚(除固定模式引脚)，将弹出一个右键菜单，显示该引脚可配置的信号；对于已经配置功能的引脚，右键单击将弹出“输入标签”的按钮，可以点击按钮，可以指定此信号的自定义标签，左击和右击效果如下图所示：

图 27. 左击和右击效果



(2) 时钟配置界面主要用来进行时钟路径及参数的配置。视图中的下拉菜单和输入框可用于修改实际的时钟树配置，以满足应用需求，时钟配置界面如下图所示：

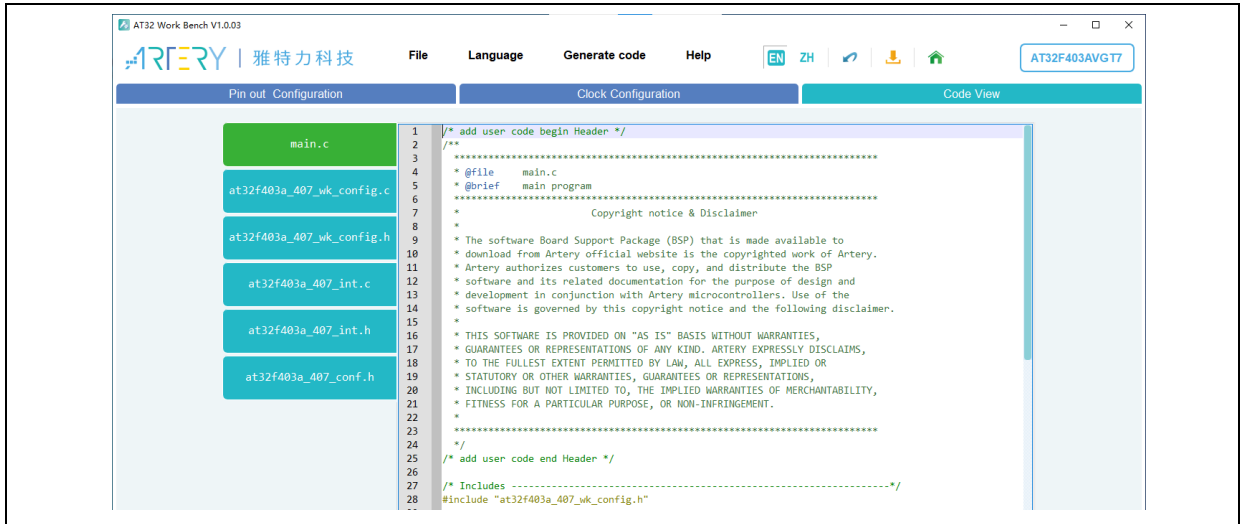
图 28. 时钟配置界面



- 注 1: 需要使 *lck* 为可用状态，需要在外设“CRM”的模式窗口中设置“低速外部时钟”的模式；
- 注 2: 需要使 *hext* 为可用状态，需要在外设“CRM”的模式窗口中设置“高速外部时钟”的模式；
- 注 3: 需要使 *clockout* 为可用状态，需要在外设“CRM”的模式窗口中勾选“时钟输出”。

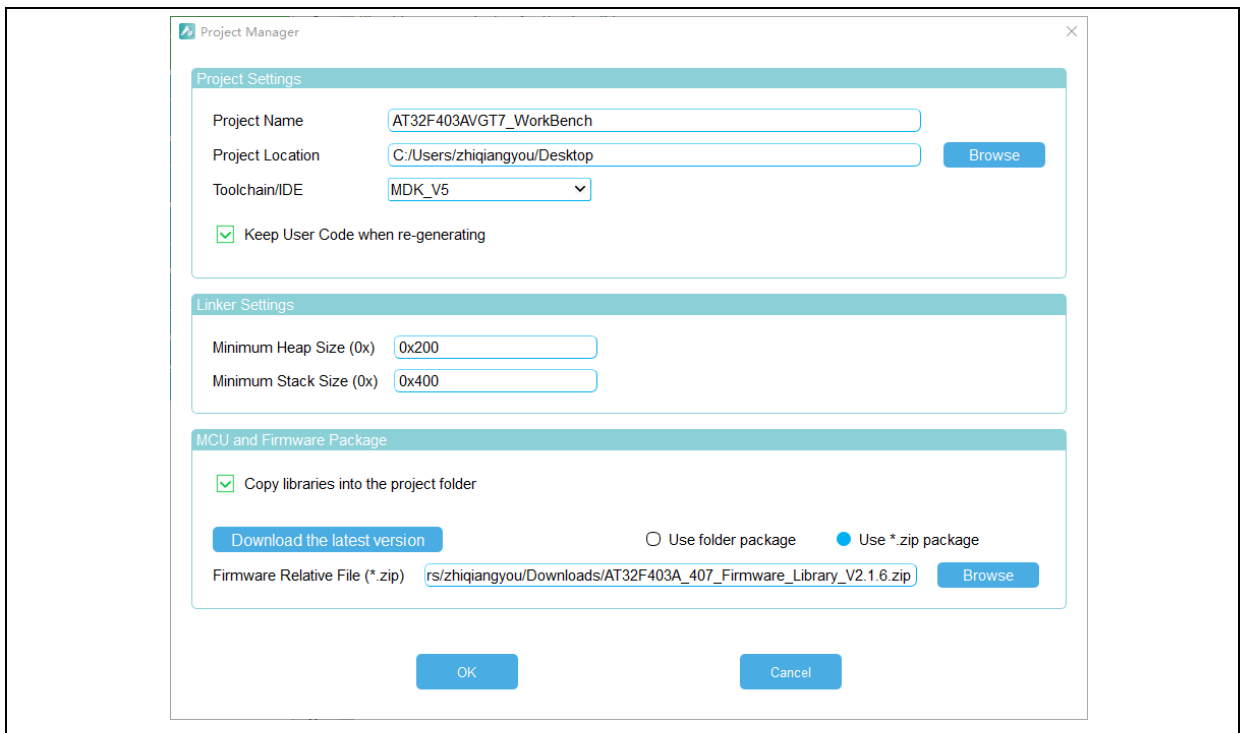
(3) 点击“代码预览”按钮，将自动生成当前配置的代码，并在窗口中显示代码内容。左侧显示了生成的代码文件，右侧显示选中文件的代码内容，如下图所示：

图 29. 代码预览



点击菜单栏或工具栏的“生成代码”按钮，将弹出“项目管理”窗口，如下图所示：

图 30. “项目管理”窗口



对于堆大小和栈大小，建议的默认值分别为 0x200 和 0x400，当应用使用中间件栈时，这些值可能需要增加；选中复制库文件到项目文件夹后，生成代码时将固件包内的库文件自动复制到项目文件夹中；配置完项目选项后，点击“确定”按钮，将自动生成用户代码和选定 IDE 的项目文件。生成的项目文件结构如下图：

图 31. 项目文件结构

名称	修改日期	类型	大小
libraries	2024/1/5 14:44	文件夹	
middlewares	2024/1/5 14:44	文件夹	
project	2024/1/5 14:44	文件夹	
AT32F403AVGT7_WorkBench.ATWP	2024/1/5 14:44	ATWP 文件	3 KB

1.1.5 快速替代 SXX 流程

- 请参考《MG0016_从 SXX32F030 移植到 AT32F421》，该文档可从[雅特力科技官方网站](#)→产品讯息→超值型 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，或联络代理及雅特力科技技术支持人员协助解决。

1.2 AT32F421 芯片的增强功能配置

1.2.1 预取指令缓冲

设置预取指令缓冲器可以使 CPU 更快地执行，CPU 读取一个字的同时下一个字已经在预取缓冲器中等候。预取指令控制器根据预取缓冲器中可用的空间决定是否访问闪存，预取缓冲器中有至少一块的空余空间时，预取控制器则启动一次读操作。

对于不同的系统时钟需要设置不同的时延，设置闪存性能选择寄存器 FLASH_PSR 的 bit2~0 (对应 SXX32 的 Flash 访问控制寄存器 FLASH_ACR)。

图 32. 闪存性能选择寄存器 (FLASH_PSR) 等待周期

Bit	Abbr.	Reset value	Type	Description
Bit 2:0	WTCYC	0x0	rw	Wait states The wait states depends on the size of the system clock, and they are in terms of system clocks. 0: Zero wait state 1: One wait state 2: Two wait states 3: Three wait states The system clock sets the wait state on a 32-MHz basis: Zero wait state for the first 32 MHz One wait state for the second 32 MHz Two wait states on the third 32 MHz Three wait states on the fourth 32 MHz

AT 库在系统时钟配置函数 system_clock_config()进行了相关设置，其他 BSP 请找到相同的位置做相应的设置。

图 33. 系统时钟配置函数 system_clock_config

```

void system_clock_config(void)
{
    /* config flash psr register */
    flash_psr_set(FLASH_WAIT_CYCLE_3);

    /* reset crm */
    crm_reset();

    crm_clock_source_enable(CRM_CLOCK_SOURCE_HICK, TRUE);

    /* wait till hick is ready */
    while(crm_flag_get(CRM_HICK_STABLE_FLAG) != SET)
    {
    }
}
    
```

1.2.2 PLL 时钟设置

1.2.2.1 PLL 的两种设定方式

AT32F421 内置的 PLL 最高可输出 120MHz 时钟，设置 AT32F421 的 PLL 有两种方式：一种是使用传统的寄存器 CRM_CFG （对应 SXX32 寄存器 RCC_CFGR），一种是使用新增的寄存器 CRM_PLL。使用 CRM_PLL 可以配置更多种 PLL 时钟频率，其对应的公式为：

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

使用 SXX32F0xx BSP 时，PLL 设定程序范例：(HSE=8MHz,PLL=48MHz)

```
RCC->CFGR |= (uint32_t)(RCC_CFGR_PLLSRC_HSE | RCC_CFGR_PLLMULL6);
```

想要使用 SXX32F0xx 程序在 AT32F421 输出大于 48MHz 时钟时，可以配置寄存器 CRM_CFG （对应 SXX32 寄存器 RCC_CFGR）。

例如 PLL=120MHz 时，设置如下：

```
RCC->CFGR |= (uint32_t)(RCC_CFGR_PLLSRC_HSE | RCC_CFGR_PLLMULL15);
```

使用 AT BSP 时 PLL 设定程序范例：(HEXT=8MHz,PLL=48MHz)

```
crm_pll_config(CRM_PLL_SOURCE_HEXT, CRM_PLL_MULT_6);
```

使用 AT BSP 时 PLL 设定程序范例：(HEXT=8MHz,PLL=120MHz)

```
crm_pll_config(CRM_PLL_SOURCE_HEXT, CRM_PLL_MULT_15);
```

还可以使用 AT32 新增的 PLL 配置寄存器 CRM_PLL 配置更多的时钟频率，以 AT32F421 BSP 为例展示 PLL 配置程序范例：(HEXT=8MHz, PLL=118MHz)

图 34. AT32F421 输出 118MHz 的时钟配置

```
#define CRM_PLL_NS ((uint16_t)0x3C) /* PLL_NS=59 */
#define CRM_PLL_MS ((uint16_t)0x01) /* PLL_MS=1 */
/* config pll clock resource PLL_FR =4*/
crm_pll_config2(CRM_PLL_SOURCE_HEXT, CRM_PLL_NS, CRM_PLL_MS, CRM_PLL_FR_4);
```

其中，入口参数 CRM_PLL_SOURCE_HEXT 代表选择 HEXT 为外部时钟源，59 为 PLL_NS 值，1 为 PLL_MS 值，CRM_PLL_FR_4（0x02，四分频）为 PLL_FR 值。

更多时钟配置相关信息，请参考《AN0116_AT32F421_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→AN0116 获取，其中详细介绍了 AT32F421 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→超值型 MCU→AT32F4xx 系列获取。

1.2.2.2 PLL 自动滑顺

当 AT32F421 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 SXX32F0xx BSP 时，打开 system_sxx32f0xx.c 找到当前的系统时钟频率配置函数（需经过 1.2.2.1 PLL 配置），在类似函数 static void SetSysClockToxxM(void) 增加如下斜黑体部分：

图 35. SXX 程序 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 AT32F421 BSP 时 PLL 自动滑顺频率切换程序范例：

图 36. AT32 程序 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。

1.2.3 加密方式

注意：AT32F421 的 BOOT1 位于用户系统数据区(0x1FFF F800)，当使用 ISP 工具时，请确认 nBOOT1=1（出厂默认值），使程序由系统启动程序代码区（而不是 SRAM）启动。

1.2.3.1 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。

注意：一旦启动高级访问保护后，将不能被解除，并且禁止用户以任何方式重新擦除以及写入系统数据区。

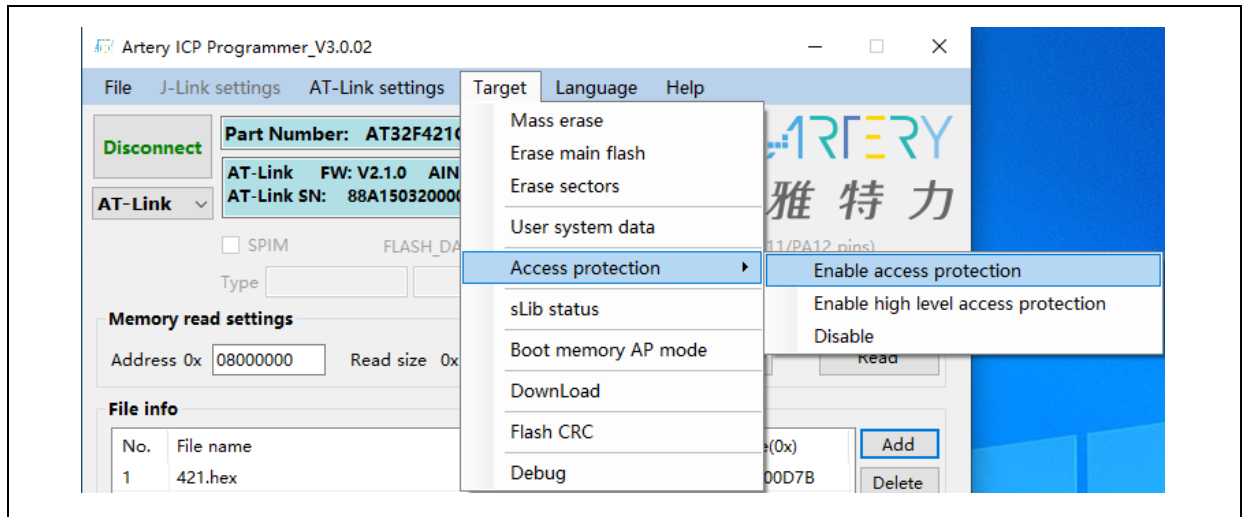
可用 ICP/ISP 工具对 IC 进行访问保护与解除访问保护操作，如下：

■ Artery ICP Programmer 工具（BOOT0=0）

启用访问保护：设备操作--访问保护--启用访问保护/启用高级访问保护。

解除访问保护：设备操作--访问保护--解除访问保护。

图 37. ICP 工具启用/解除访问保护



■ Artery ISP Programmer 工具（BOOT0=1）

启用访问保护：一直点击下一步，直到进入最后界面，选择保护、启用访问保护/高级访问保护--下一步--是，即可将程序加密。

解除访问保护：选择保护、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）

启用访问保护：使能/除能保护、启用访问保护/高级访问保护--开始，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，即可将 Flash 解除加密。

图 38. ISP 工具启用访问保护

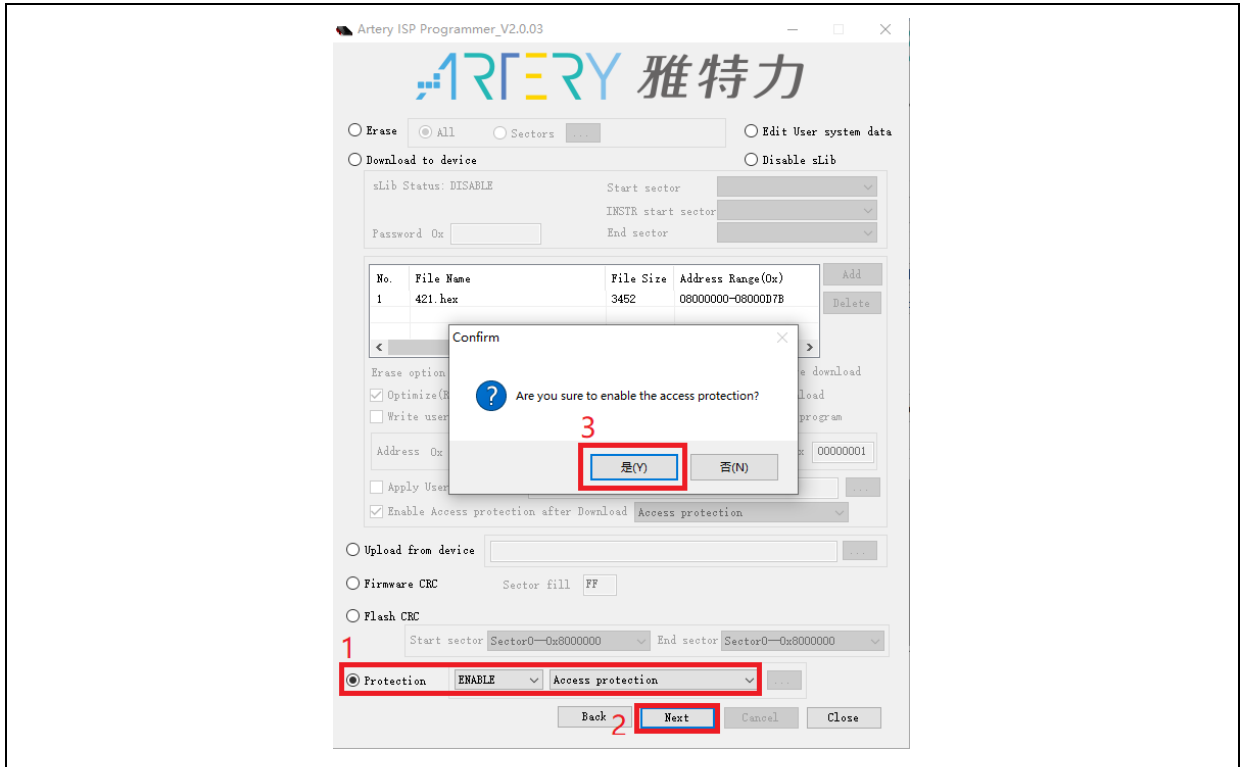
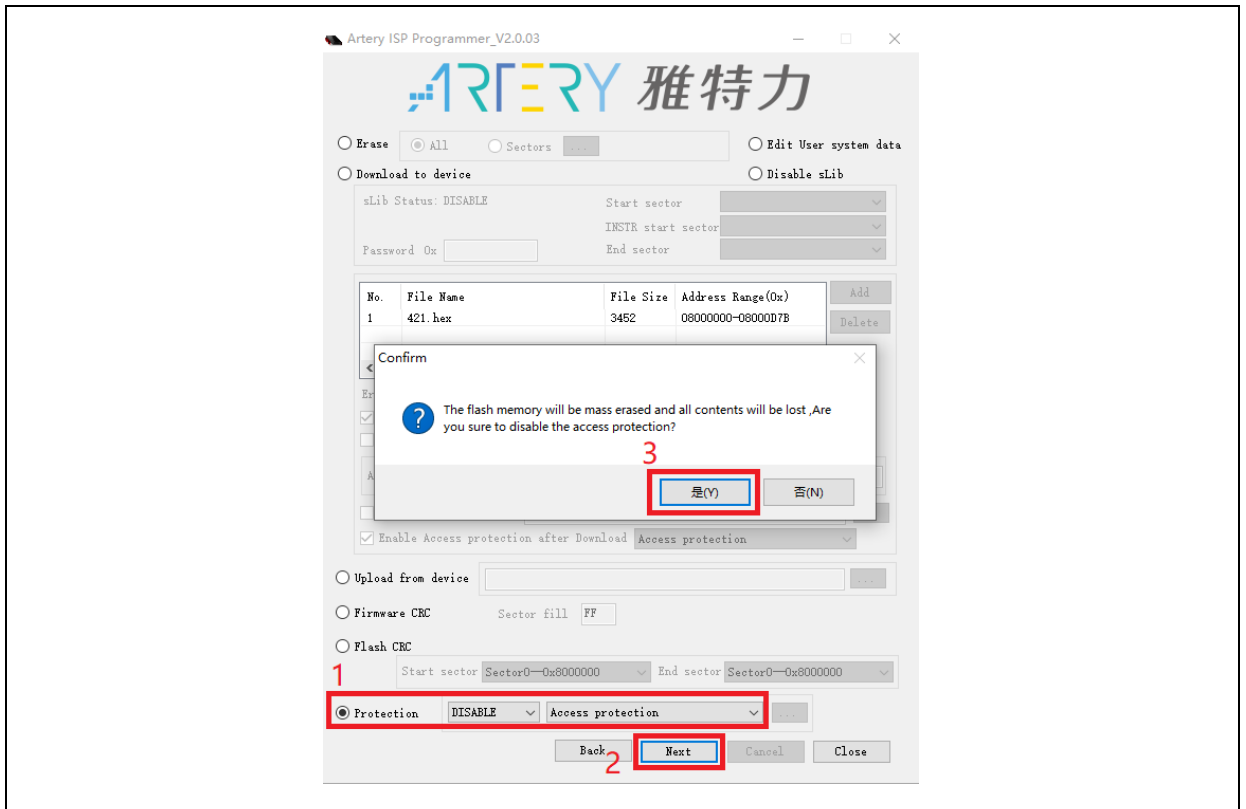


图 39. ISP 工具解除访问保护



注意： 设置了访问保护不能通过擦除操作来解除访问保护。

1.2.3.2 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）

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--勾选擦写保护字节扇区--应用到设备。

■ Artery ISP Programmer 工具（BOOT0=1）

启用擦写保护：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：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）

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。

图 40. ICP 工具启用擦写保护

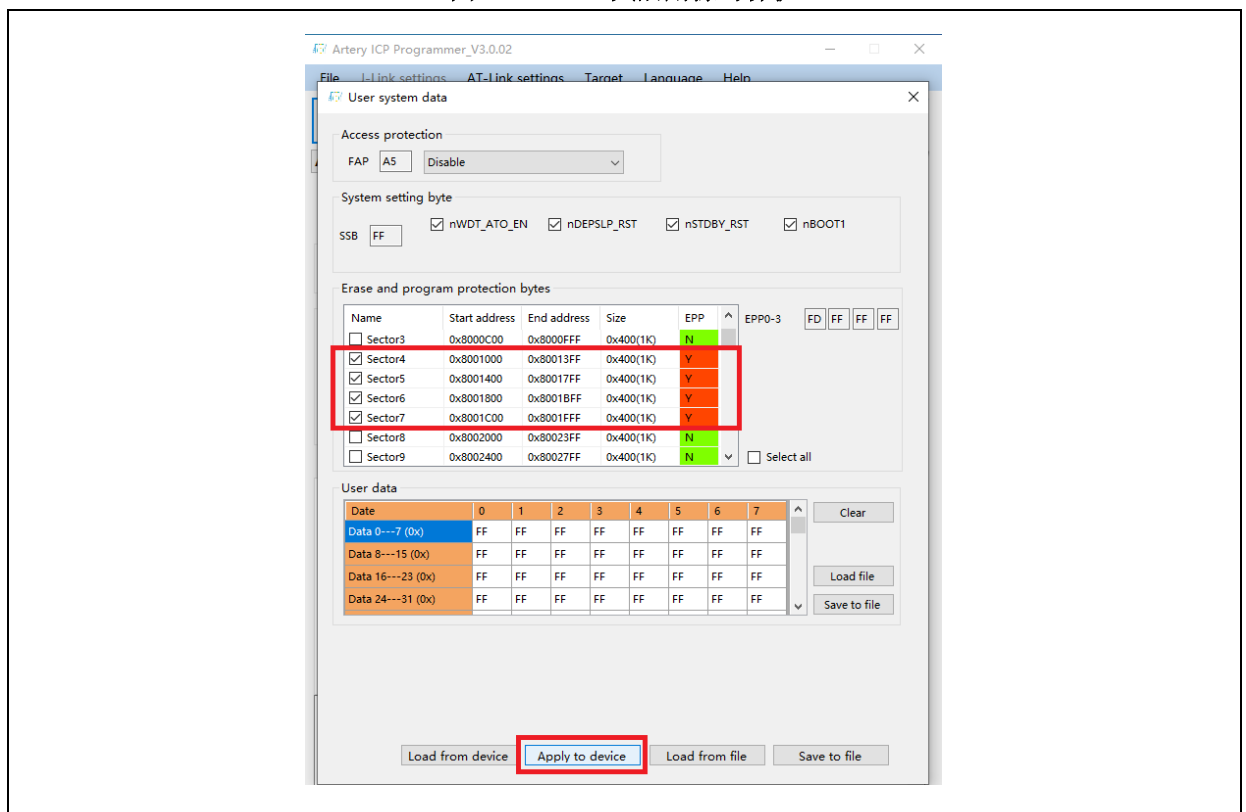
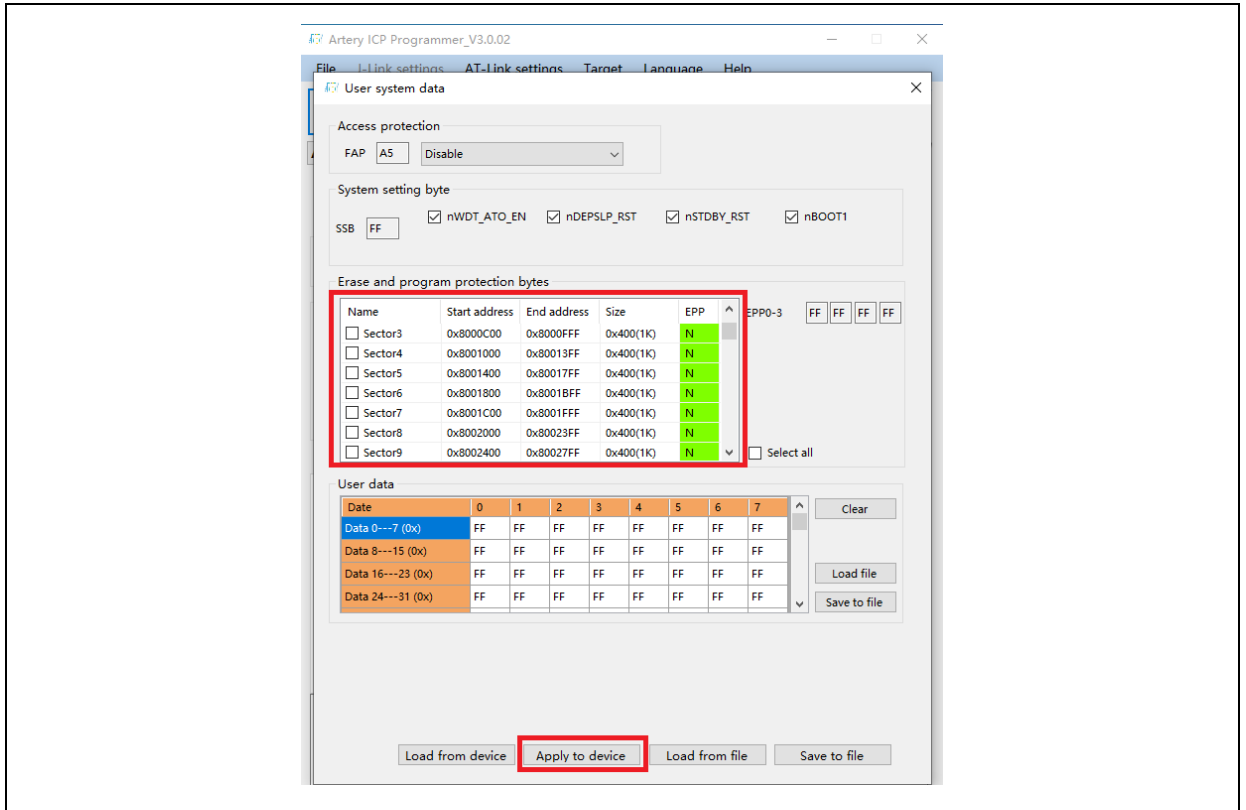


图 41. ICP 工具解除擦写保护



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 设定系统存储器为扩展主存

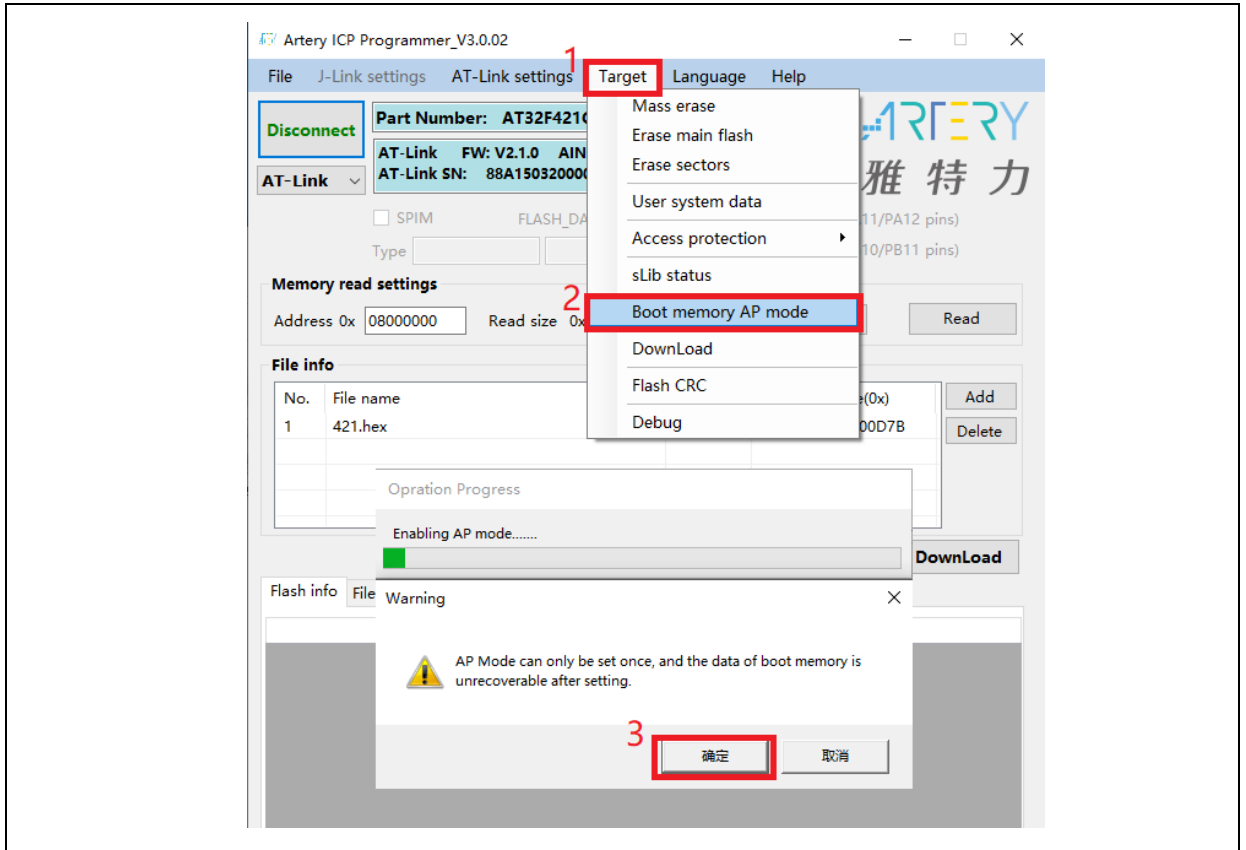
系统存储器 (System Memory) 默认是作为 BOOT 模式存放原厂固化的启动代码。不过，在 AT32F421 系列产品上，添加了新功能，系统存储器也可以选择作为主存的扩展区(AP 模式)用来存放用户自定义代码。

注意： 系统存储区 AP 模式只能设置一次且不可逆，设置后原系统存储器 BOOT 模式功能不可恢复。

产品开发过程中使用 Artery ICP Programmer 将系统存储器作为扩展主存使用方法：

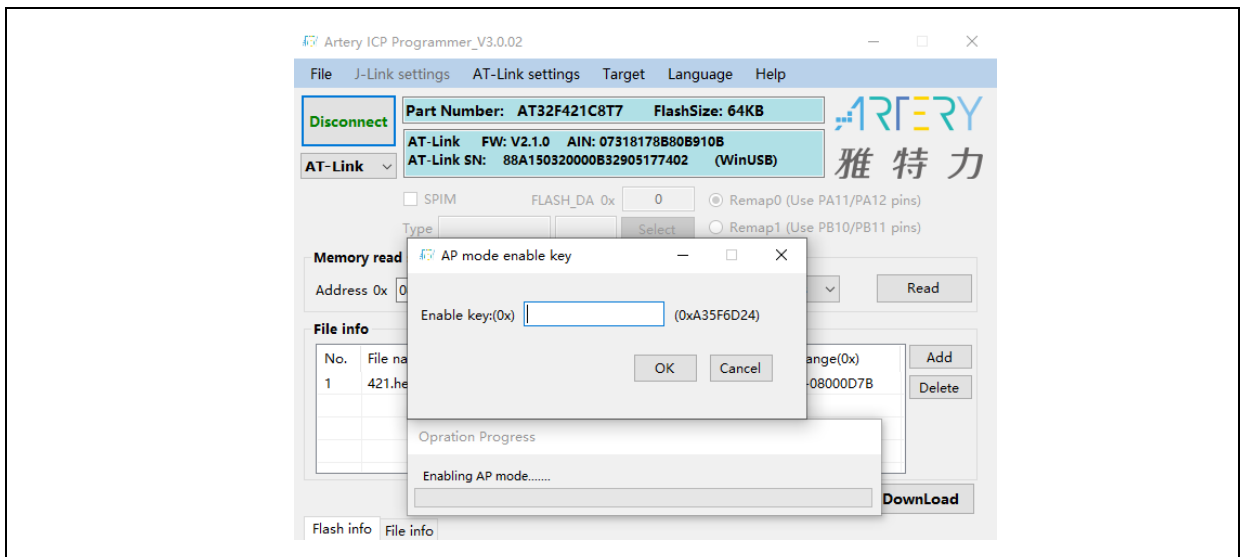
- 连接 AT-Link/J-Link 仿真器到 AT-START-F421 板并上电。
- 开启 Artery ICP programmer，选择用 AT-Link/J-Link 连接。
- 通过菜单栏：设备操作--系统存储区 AP 模式--确定。

图 42. ICP 工具设定系统存储区 AP 模式



- 为防止误操作，需按照提示手动输入启用秘钥 0xA35F6D24，操作后“存储器信息”表格中会有成功或失败的提示信息。

图 43. ICP 工具设定系统存储区 AP 模式确认界面



量产生产过程中使用 Artery ICP Programmer 将系统存储器作为扩展主存使用方法：

- 连接 AT-Link 仿真器到 AT-START-F421 板并上电。

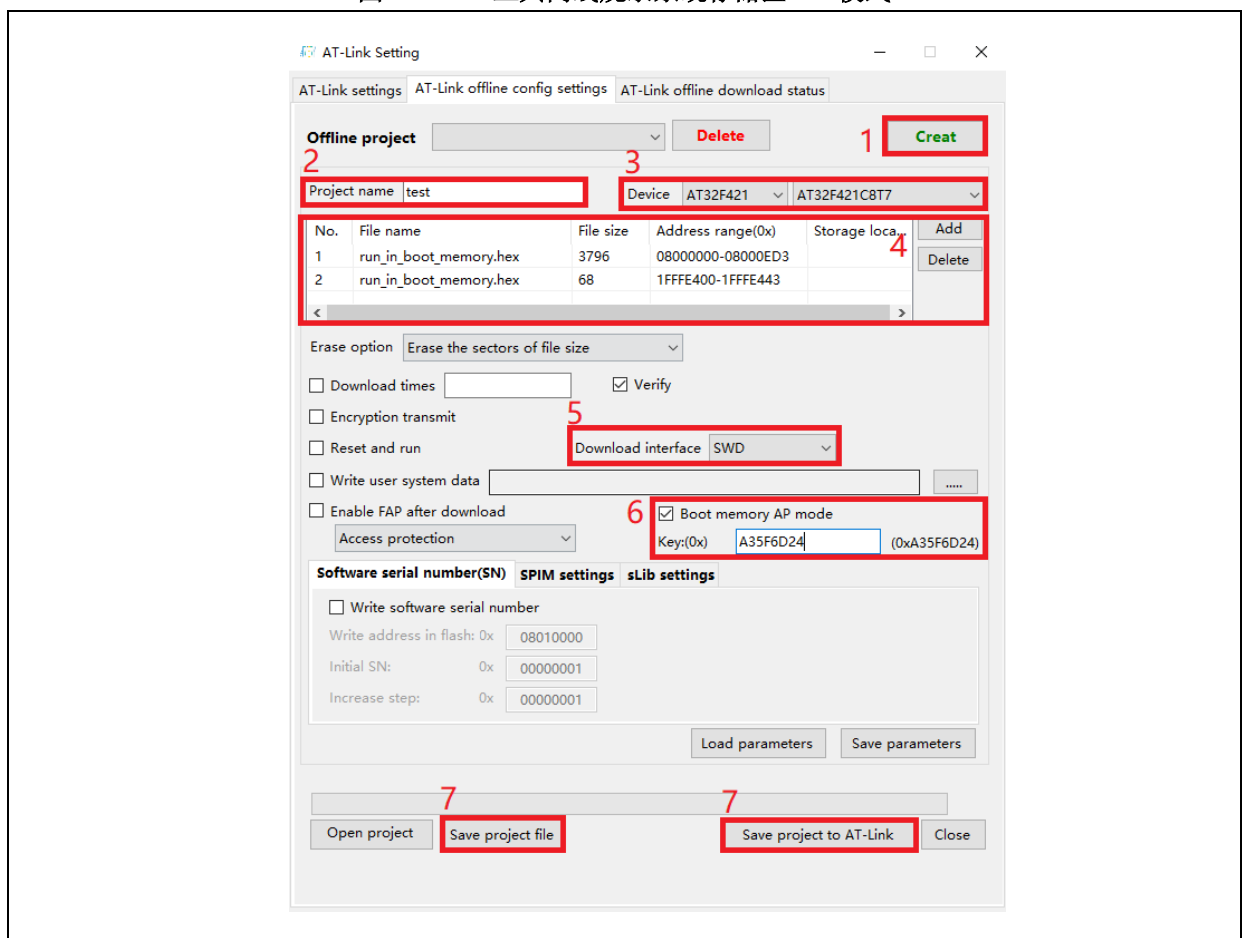
注意：板载的 AT-Link EZ 版本不支持离线烧录，只能选择非 EZ 版本的 AT-Link。

- 开启 Artery ICP programmer，选择用 AT-Link 做连接。

- 通过菜单栏：AT-Link 设置--离线项目配置。
- 生成离线项目的步骤：
 1. 新建项目
 2. 输入项目名称
 3. 选择 MCU 型号
 4. 添加.hex 文件
 5. 下载接口选择 SWD
 6. 勾选系统存储区 AP 模式并输入密钥
 7. 保存项目到 AT-Link 或保存项目文件

其他选择根据实际需要进行设置。

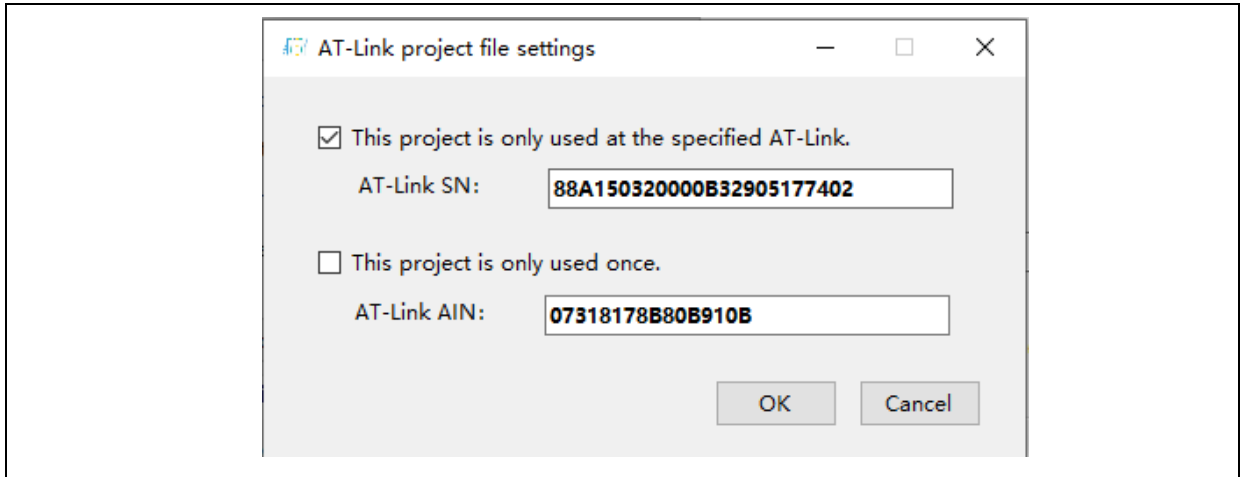
图 44. ICP 工具离线烧录系统存储区 AP 模式



- 若上述步骤 7 选择的是保存项目文件，将会把项目保存为.atcp 格式的文件，方便加载到其他 AT-Link 中使用。

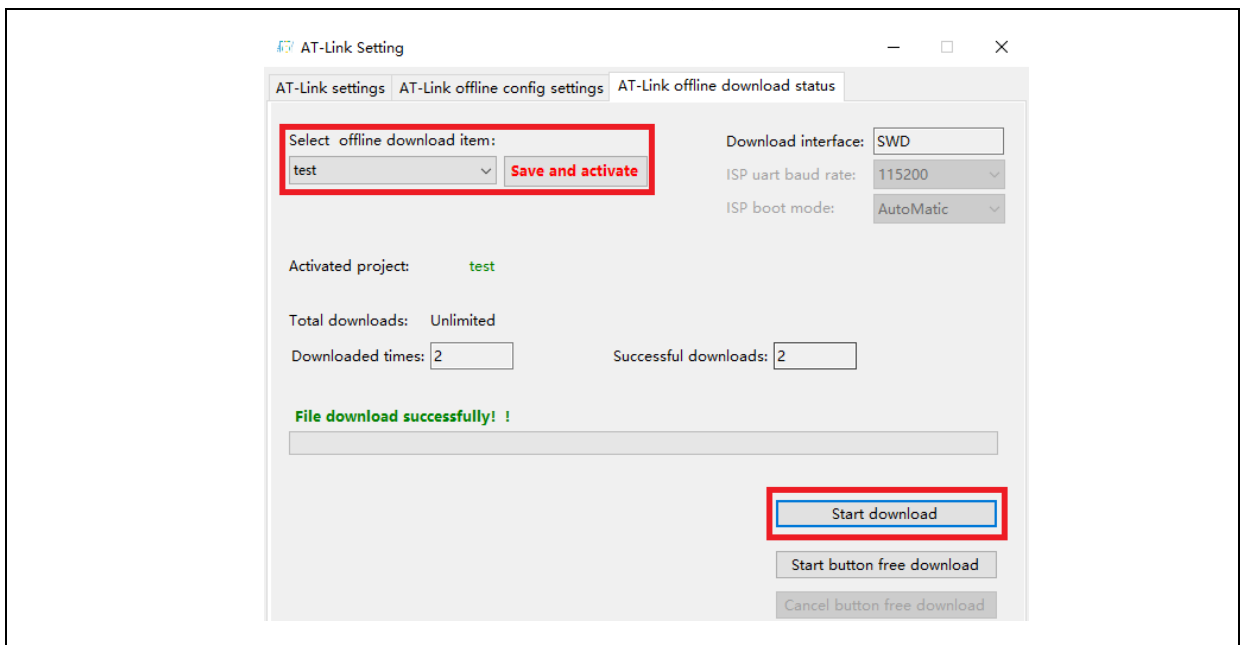
在操作过程中会弹出如下图的窗口。如果选择此项目仅允许在指定 AT-Link 使用，此项目文件与 AT-Link 绑定，只能在绑定的 AT-Link 使用，需设定绑定的 AT-Link 序列号；如果选择此项目仅允许使用一次，此项目文件在同一个 AT-Link 只能使用一次。

图 45. ICP 工具离线烧录项目文件设置



- 若上述步骤 7 保存项目到 AT-Link 成功，在离线下载状态监控窗口里，选择离线下载项目名称--保存并激活，就可以开始烧录了。

图 46. ICP 工具离线下载状态监控



- 关于系统存储器扩展的更多信息请参考《AN0066_config_boot_memory_as_extension_of_main_memory(AP_mode)》，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→AN0066 获取。
- 在系统存储器运行用户程序的 Demo 请参考 BSP，存放路径为[雅特力科技官方网站](#)→产品讯息→超值型 MCU→AT32F4xx 系列下载的 BSP 解压后 AT32F421_Firmware_Library_V2.x.x\utilities\at32f421_boot_memory_ap_demo。

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 47. 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 48. 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};

/* AT32F421 MCU type table */
const uint64_t AT32_MCU_ID_TABLE[] =
{
    0x0000000950010016, // AT32F421PF4P7    16KB    TSSOP20
    0x0000000950020115, // AT32F421PF8P7    64KB    TSSOP20
    ...
};

/* 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

- 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。
- 程序使用 SRAM 超过 MCU SRAM 空间大小。
- 系统时钟设置超出规格。

2.2 Keil 项目内 Jlink 无法找到 IC

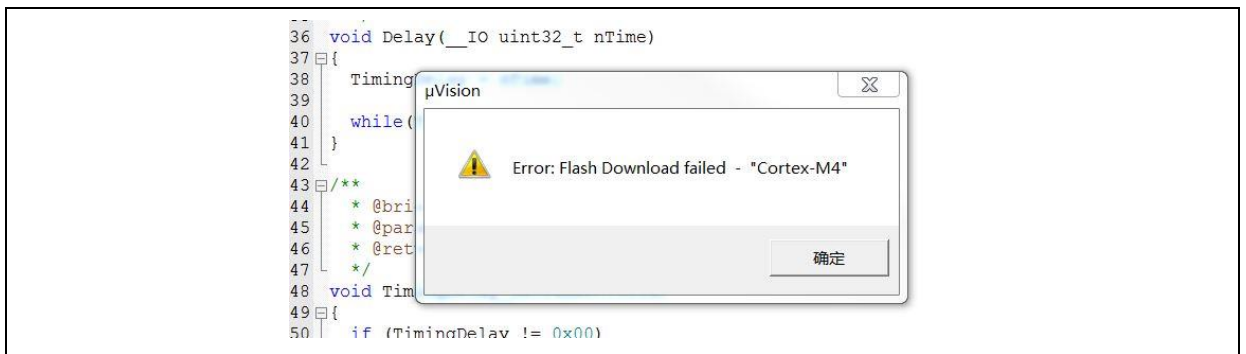
- 请参考《FAQ0008_Keil 项目内 Jlink 无法找到 IC 问题》，该文档可以从[雅特力科技官方网站](#)→技术与开发支持→FAQ→FAQ0008 下载。
- 请参考《FAQ0132_JLink 手动添加 Artery MCU》，该文档及其附件可以从[雅特力科技官方网站](#)→技术与开发支持→FAQ→FAQ0132 下载。

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 49. 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 选择错误，BOOT0 管脚电平须设置为 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

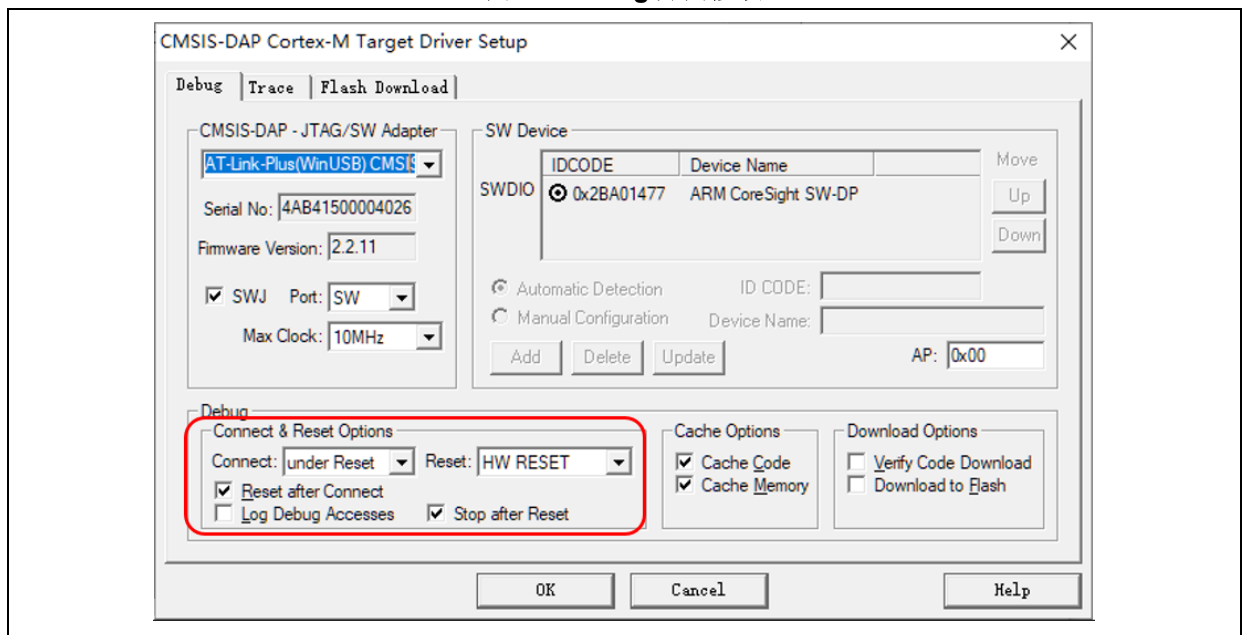
在使用 AT32F421 时，用户可能在以下操作后无法再次下载程序：

- 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。

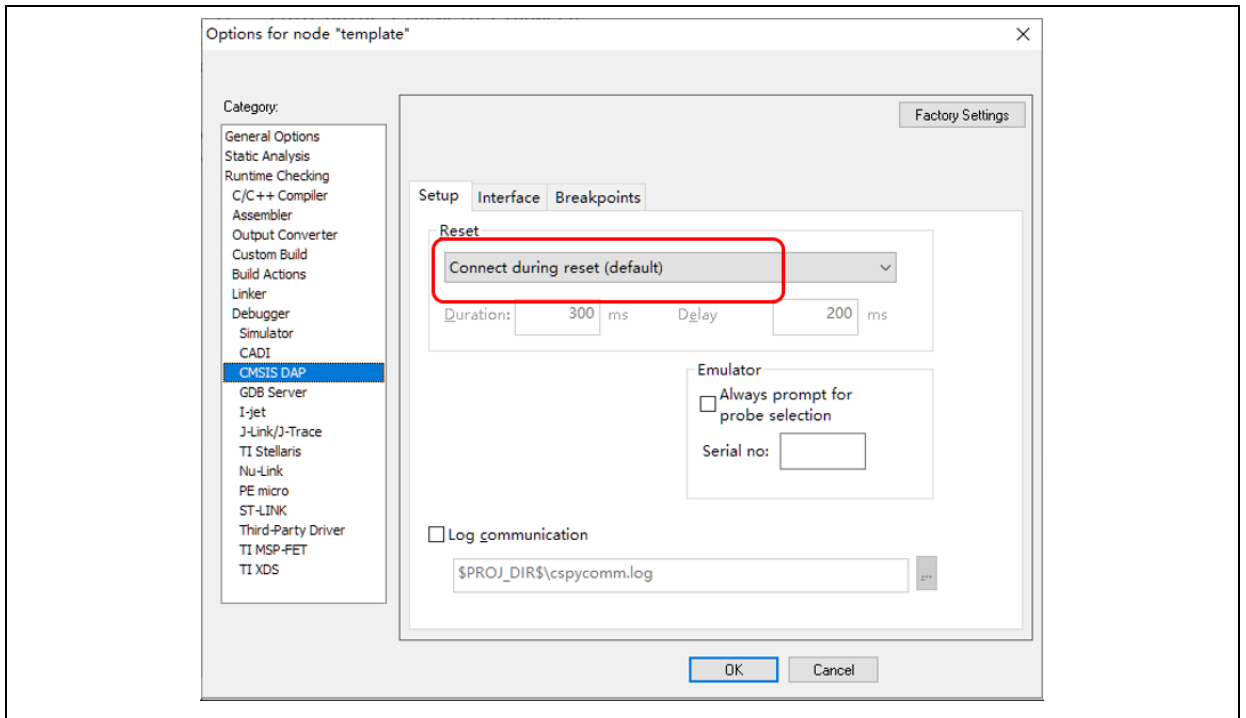
1. 更改芯片 BOOT 模式，改为启动程序存储器启动或者 sram 启动，然后通过复位脚复位芯片，此时就可以擦除程序恢复下载。
2. 使用 ICP 软件工具搭配 AT-Link 调试器，连接 AT-Link RST pin 到芯片的复位脚，ICP 界面上直接点击连接即可正常连接，连接后擦除芯片内程序即可恢复。
3. 使用 Keil 软件搭配 AT-Link 调试器，连接 AT-Link RST pin 到芯片的复位脚，在 Keil 的 debug 界面修改为下图中红框内的选项，就可以擦除程序恢复下载。

图 50. 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 51. CMSIS DAP 界面修改



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。

应这一重要需求，AT32F421 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或者0x00。
- 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《AN0071_AT32F421_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→AN0071 获取。

4 文档版本历史

表 1. 文档版本历史

日期	版本	变更
2021.12.29	2.0.0	最初版本
2022.05.09	2.0.1	新增BOOT1描述, 新增提示高级访问保护不能解除, 优化部分描述。
2022.07.15	2.0.2	优化部分表述
2022.10.10	2.0.3	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	2.0.4	完善UID,PID表述
2024.01.05	2.0.5	更新“AT32恢复下载”的描述, 增加“AT32 Work Bench”的描述
2024.03.06	2.0.6	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