

AT32F403A & AT32F407入门使用指南

前言

这篇入门手册旨在让用户快速使用AT32F403A，AT32F407 MCU进行项目开发，AT32F407 相较于AT32F403A 增加了以太网（ETH）功能

支持型号列表：

支持型号	AT32F403A
	AT32F407

目录

1 雅特力初步环境准备	4
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	9
1.2 AT32F403A_407芯片的增强功能配置	9
1.2.1 PLL大于72MHz设定	9
1.2.2 如何打开FPU功能（硬件浮点运算单元）	10
1.2.3 AT32F403A_407 零等待/非零等待Flash和内置SRAM大小选择配置说明	11
1.2.4 加密方式（读保护,外部Flash的加密）	17
1.2.4.1 读保护	17
1.2.4.2 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）	19
1.2.5 在程序中区分AT与其他IC方法	22
1.2.5.1 UID/PID 组合方式来识别	22
1.2.5.2 取 32 位作简缩的唯一 UID 码来识别	22
2 下载编译过程常见问题	23
2.1 程序启动进入Hard Fault Handler	23
2.1.1 常见进入Hardfault异常状况	23
2.2 程序下载过程出问题	23
2.2.1 显示 Error: Flash Download failed – “Cortex-M4” 问题	23
2.2.2 显示 No Debug Unit Device found 问题	23
2.2.3 显示 RDDI-DAP Error 问题	23
2.2.4 ISP串口下载时卡死问题	23
2.2.5 AT32恢复下载	24
2.2.5.1 KEIL 环境下的解决方法	24
2.2.5.2 IAR 环境下的解决方法	25
2.3 升级轩微烧录器固件	25
3 版本历史	27

表目录

表 1 .BSP 与 PACK 的选择	8
表 2 . 文档版本历史	27

图目录

图 1. AT32F403A 开发板实物图	6
图 2. Keil Debug 选项	7
图 3. Keil Debug 选项 settings 设置	7
图 4. Keil Utilities 选项	7
图 5. IAR Debug 选项	7
图 6. IAR CMSIS-DAP 选项	8
图 7. SXX Flash 等待位	9
图 8. ARTERY BSP/Pack Keil 环境中选择开启 FPU	10
图 9. SXX BSP / ARTERY Pack Keil 环境中选择开启 FPU	10
图 10. Keil 环境中增加开启 FPU 的代码	10
图 11. IAR 环境中选择开启 FPU	11
图 12. IAR 环境中增加开启 FPU 的代码	11
图 13. ICP 工具选项字节操作选择 SRAM 大小	12
图 14. ICP 工具选项字节操作选择 SRAM 大小	12
图 15. ISP 工具选项字节操作选择 SRAM 大小	12
图 16. ISP Multi-Port Programme 工具选项字节操作选择	13
图 17. 轩微烧录器选项字节操作选择 SRAM 大小	13
图 18. 轩微烧录器选项字节操作选择 SRAM 大小	14
图 19. ISP 启用读保护	17
图 20. ISP 解除读保护	18
图 21. ISP Multi-Port Programmer 启用读保护	18
图 22. ISP Multi-Port Programmer 解除读保护	19
图 23. ICP 工具对外部存储器加密操作	20
图 24. ISP 工具对外部存储器加密操作	20
图 25. ISP Multi-Port Programmer 工具对外部存储器加密	21
图 26. 增加开启 FPU 的代码	23
图 27. 下载出现 Flash Download failed – “Cortex- 4”	23
图 28. KEIL 环境中工具 ConfigJLink_V1.0.0 的操作	24
图 29. KEIL 环境中工具 ConfigJLink_V1.0.0 的操作	24
图 30. IAR 环境中工具 ConfigJLink_V1.0.0 的操作	25

图 31. IAR 环境中工具 ConfigJLink_V1.0.0 的操作	25
图 32. 轩微烧录器固件生成时间	26
图 33. 轩微烧录器固件显示图片	26

1 雅特力初步环境准备

雅特力 MCU 资料下载地址:

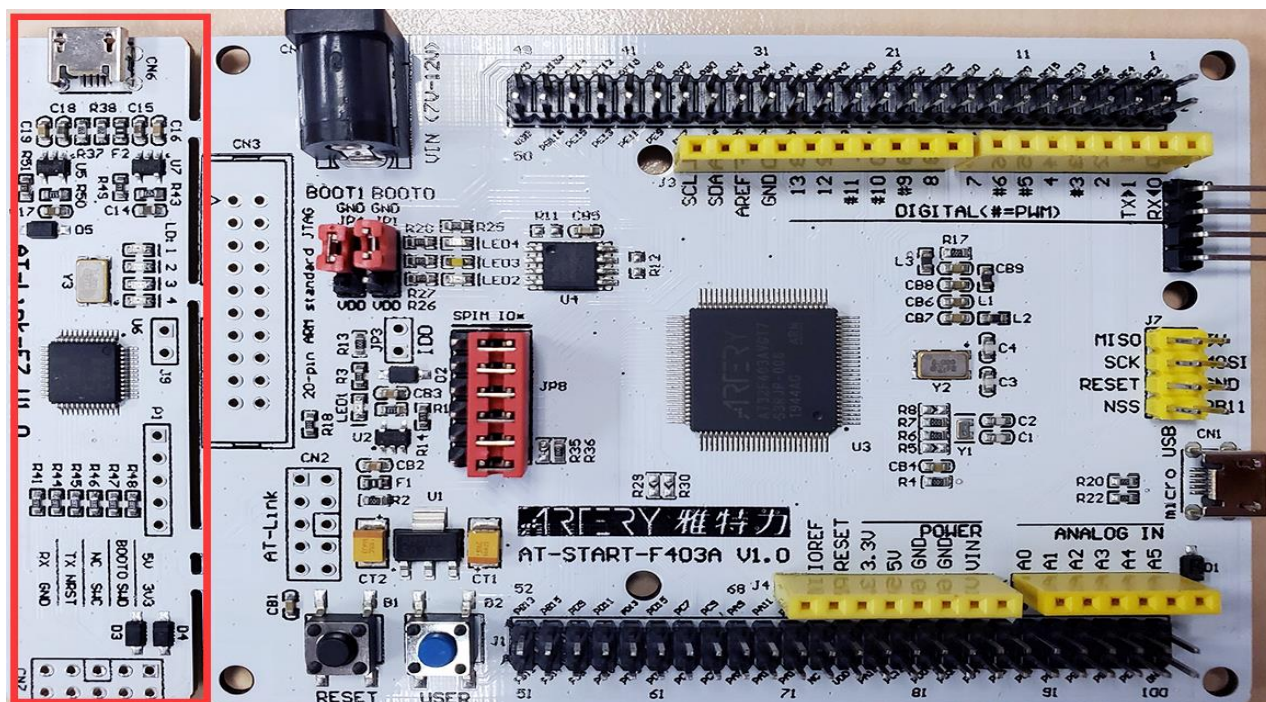
■ 雅特力官方网站: <http://www.arterytek.com>

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 如下图左边红框部分所示, 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



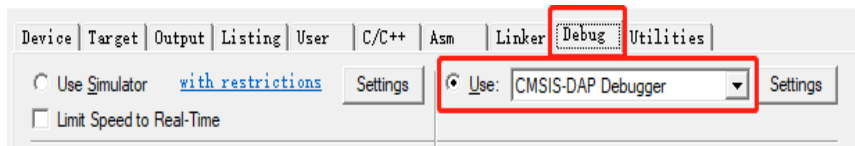
1.1.2 烧录工具及软件

- AT 烧录工具及软件: AT-Link / AT-Link-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

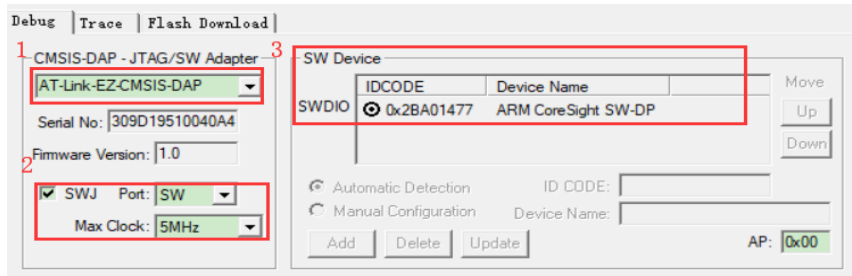
- ① 对于 Keil 编译系统, 建议是 Keil 4.74 或 5.23 以上版本;
在 Keil 环境下使用 AT-Link-EZ, 在 Debug 里选择 CMSIS-DAP 调试器。

图 2. Keil Debug 选项



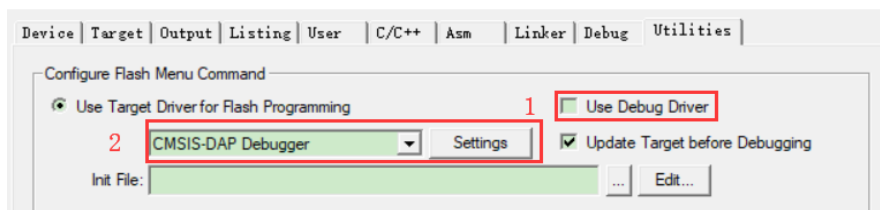
在 Debug 单击 settings 进入 cortex – M Target Driver Setup 界面如下图， 1 先选择的 AT-Link-EZ-CMSIS-DAP
2 Port: 选择 SW，再勾选 SWJ 3 确认已识别到 ARM SWD 调试模块

图 3. Keil Debug 选项 settings 设置



并且在 Utilities 里，先勾掉 1 选项框，选择 2 下拉菜单选择 CMSIS-DAP Debugger，然后再勾选 1（需要先取消再勾选）。

图 4. Keil Utilities 选项



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

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

图 5. IAR Debug 选项

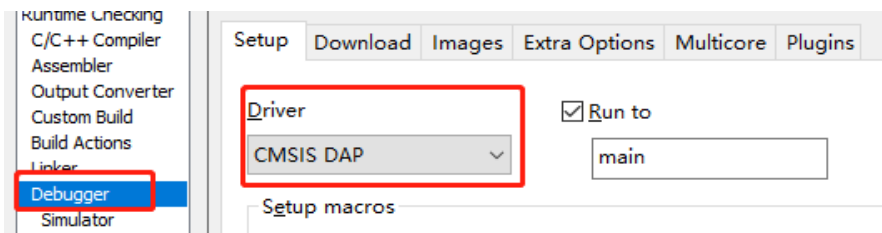
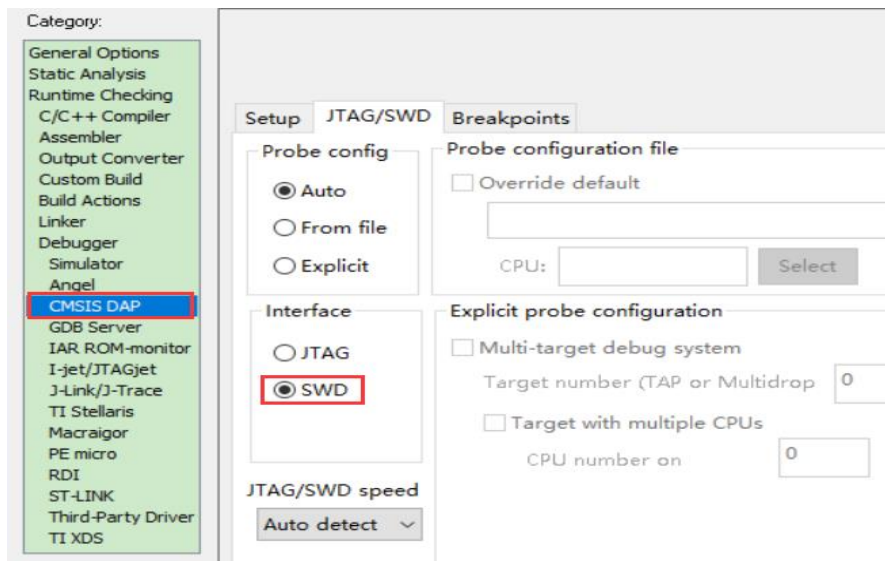


图 6. IAR CMSIS-DAP 选项



- ③ 目前对 IAR 的编译环境 pack 暂不支持 Bank3（通过 SPIM 口外挂 Flash 内执行程序方式）的操作，需要用 bank3 还请在 Keil 环境中编译；
- ④ BSP 与 PACK 的选择(六种情形描述如下)

表 1.BSP 与 PACK 的选择

序号	使用什么 BSP/Pack 开发	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 参考《SXX32F103 对 AT32F403A_407系列 移转手册_V1.0.0》文件修改对应程序
6	SXX32F103 BSP/Pack	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 参考《SXX32F103 对 AT32F403A_407系列 移转手册_V1.0.0》文件修改对应程序

BSP 及 PACK 安装详细操作见《AT32F4xx 标准库 BSP&Pack 应用指南》，在官网下载的《AT32F4xx_StdPeriph_Lib_V1.x.x》文件夹内。

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

图 7. 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，AT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《SXX32F103对AT32F403A_407系列 移转手册_V1.x.x》；
- 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。
- *Note: 由于AT32F403A_407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_407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_V1.0.x.pdf》*

1.2 AT32F403A_407 芯片的增强功能配置

1.2.1 PLL 大于 72MHz 设定

AT32F403A_407 内置的PLL最高可输出240 MHz时钟，时钟高于72MHz时设定略有不同，须根据输出频率设定PLL_RANGE寄存器

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

例如AT32F403A PLL设定程序范例：(HSE = 8 MHz, PLL = 240 MHz)

```
RCC->CFG |= (uint32_t)(RCC_CFG_PLLRC_HSE | RCC_CFG_PLLMULT30 | RCC_CFG_PLLRANGE_GT72MHZ);
```

开启滑顺功能：

```
RCC_StepModeCmd(ENABLE);
```

关闭滑顺功能：

```
RCC_StepModeCmd(DISABLE);
```

开/关滑顺功能函数定义：

```

void RCC_StepModeCmd(FunctionalState NewState)
{
    assert_param(IS_FUNCTIONAL_STATE(NewState));
    if(NewState == ENABLE)
    {
        RCC->MISC2 |= RCC_MISC2_AUTO_STEP_EN;
    }
    else
    {
        RCC->MISC2 &= ~RCC_MISC2_AUTO_STEP_EN;
    }
}

```

注：PLL 等于 72MHz 设定和 SXX32F103 是一样

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

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

AT32F403A PLL 设定程序范例：(HSE = 8 MHz, PLL = 72 MHz)

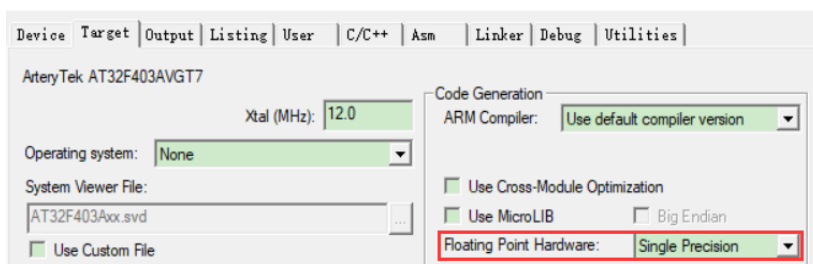
RCC->CFG |= (uint32_t)(RCC_CFG_PLLRC_HSE | RCC_CFG_PLLMULT9);

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 直接选择 Floating Point Hardware 如下图：

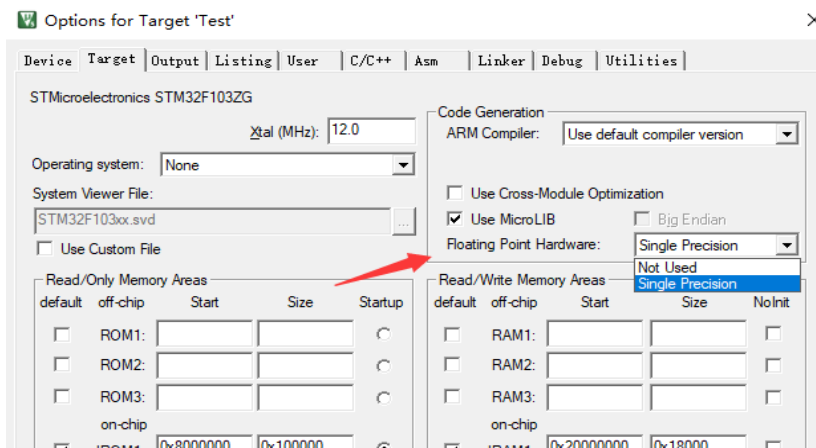
图 8. ARTERY BSP/Pack Keil 环境中选择开启 FPU



- 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 功能需要如下处理：

- 首先需参考《AT32F4xx 标准库 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 9. SXX BSP / ARTERY Pack Keil 环境中选择开启 FPU



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

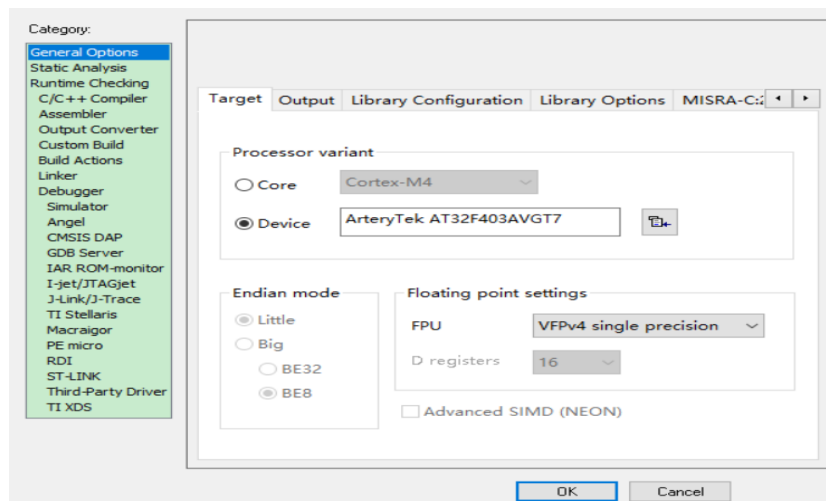
图 10. Keil 环境中增加开启 FPU 的代码

```
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
}
```

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 如下图：

图 11. 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 功能需要如下处理：

- 首先需参考《AT32F4xx 标准库 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 函数中添加以下配置，并将 cm4.h 添加入工程中。

图 12. IAR 环境中增加开启 FPU 的代码

```
#if defined ( _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_407 零等待/非零等待 Flash 和内置 SRAM 大小选择配置说明

通过选择字节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 可以配置为以下两种配置：

- ZW: 256KB, NZW: 768 KB, SRAM: 96 KB (默认配置)；
- 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 大小配置涉及 FMC 选择字节说明，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 有效必须要掉电或 RESET 一次。

① 使用 ICP/ISP 方式

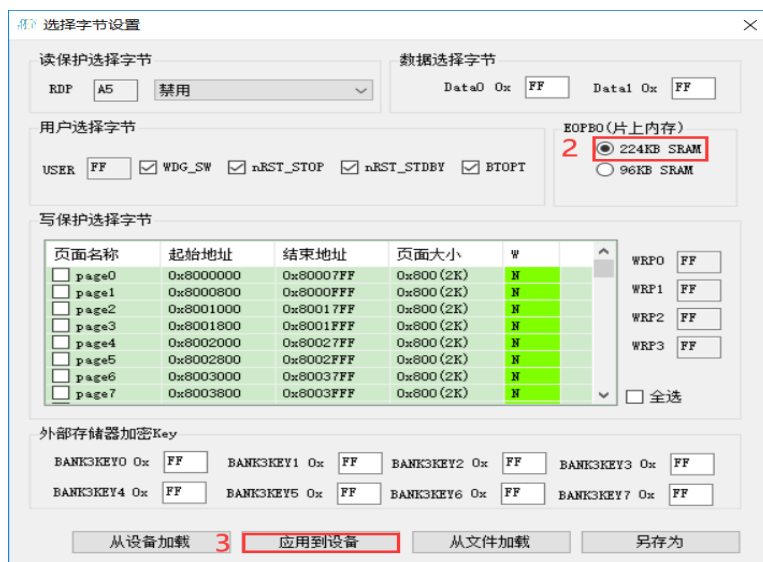
■ ICP 工具

设备操作 1 选项字节 2 选择 96KB/224KB 3 应用到设备。

图 13. ICP 工具选项字节操作选择 SRAM 大小



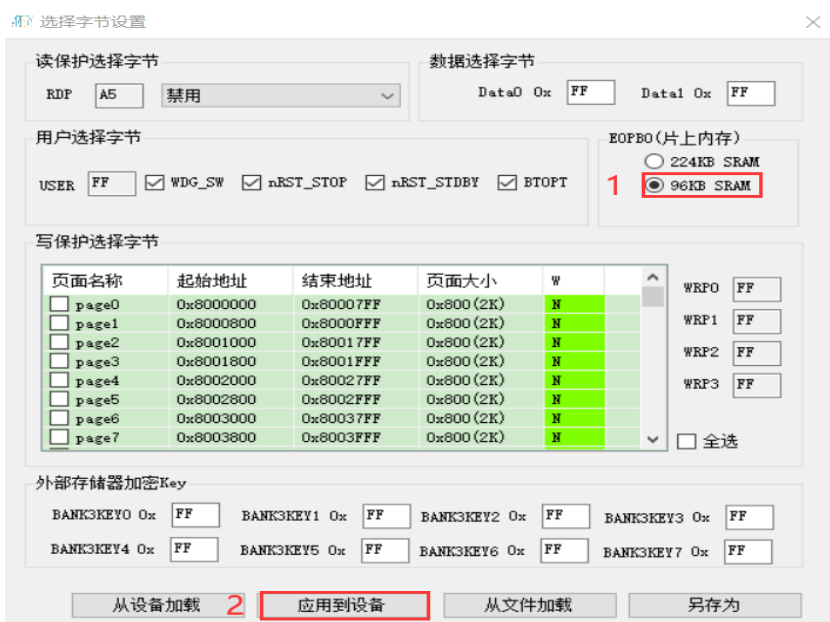
图 14. ICP 工具选项字节操作选择 SRAM 大小



■ Artery ISP Programmer 工具

进入最后界面，1 选择 96KB/224KB，2 应用到设备。

图 15. ISP 工具选项字节操作选择 SRAM 大小

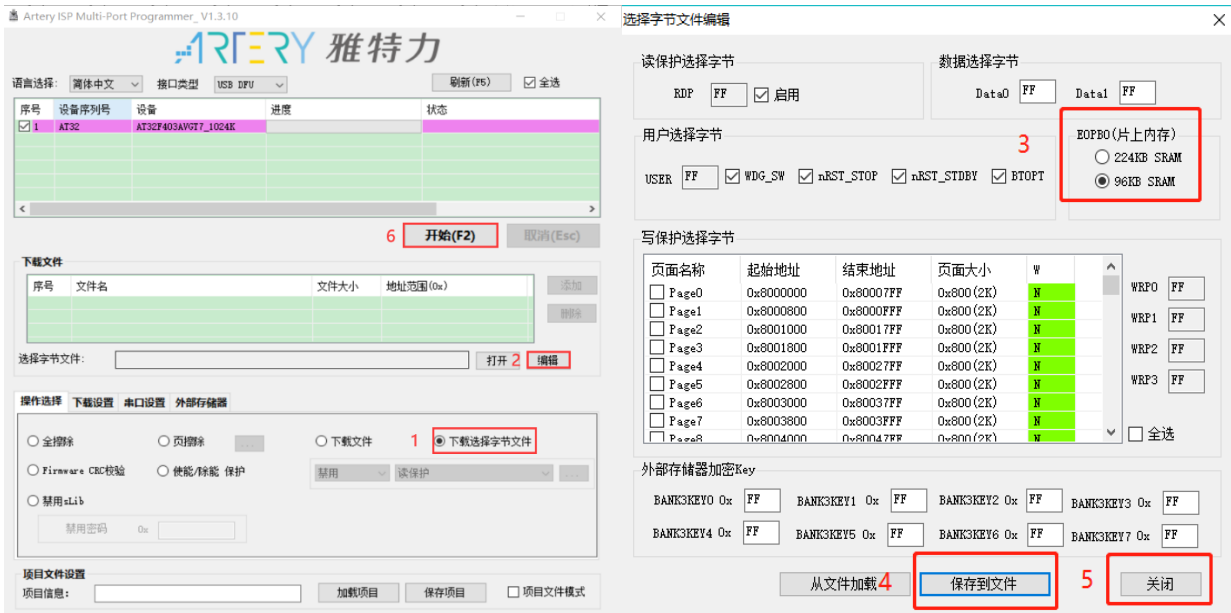


■ Artery ISP Multi-Port Programmer 工具

下载选择字节文件---编辑---选择 96KB/224KB---保存到文件（新建一个选择字节烧录档）---关闭---开始，或者

下载选择字节文件---打开（已经保存的选择字节烧录档）---开始。

图 16. ISP Multi-Port Programmer 工具选项字节操作选择



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 MCU 来选择 SRAM 大小。以轩微为例如下图, 1 先选择 MCU 型号, 2 储存的烧录文件编号, 3 加载要烧录 HEX 或 BIN 档,

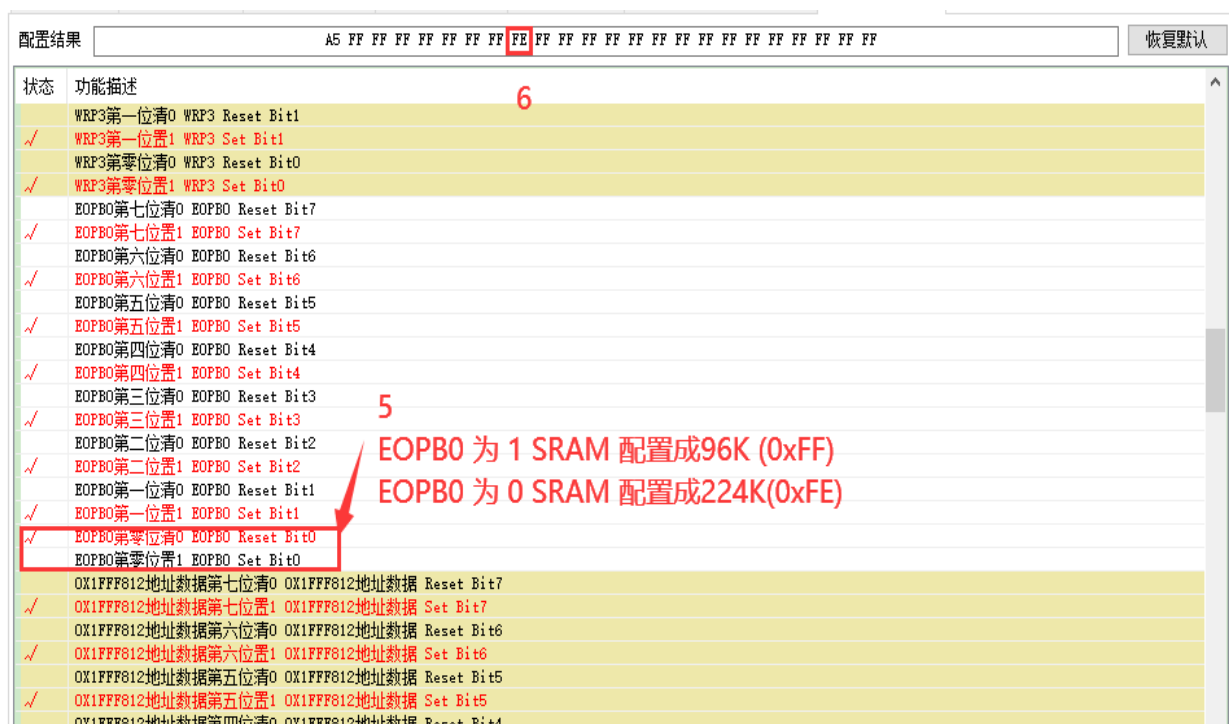
4 编辑选项字节, 5 勾选 EOPB0 第零位置清 0, 6 确认 FF 变成 FE, 7 勾选写并检验选项字节, 8 把配置好的镜像写入脱机编程器

图 17. 轩微烧录器选项字节操作选择 SRAM 大小



选项字节配置 EOPB0=0xFF 表示片上 SRAM 为 96KB，EOPB0=0xFE 表示片上 SRAM 为 224KB

图 18 轩微烧录器选项字节操作选择 SRAM 大小



③ 客户也可以使用在 Bootloader program (IAP) 内修改 SRAM 的大小，函数定义如下，并调用它范例如下：

```
void Extend_SRAM(void)
{
    // check if RAM has been set to 224K, if not, change EOPB0
    if(((UOPTB->EOPB0) & 0xFF) != 0xFE)
    {
        /* Unlock Option Bytes Program Erase controller */
        FLASH_Unlock();

        /* Erase Option Bytes */
        FLASH_EraseUserOptionBytes();

        /* Change SRAM size to 224KB */
        FLASH_ProgramUserOptionByteData((uint32_t)&UOPTB->EOPB0,0xFE);
        NVIC_SystemReset();
    }
}
```

④启动文件中修改 AT32F403A_407 SRAM 为 224KB

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：

; 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

; 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
```

在应用程序中添加声明和定义 Extend_SRAM 函数

```
void Extend_SRAM(void)
{
    // check if RAM has been set to 224K, if not, change EOPB0
    if(((UOPTB->EOPB0) & 0xFF) != 0xFE)
    {
        /* Unlock Option Bytes Program Erase controller */
        FLASH_Unlock();

        /* Erase Option Bytes */
        FLASH_EraseUserOptionBytes();

        /* Change SRAM size to 224KB */
        FLASH_ProgramUserOptionByteData((uint32_t)&UOPTB->EOPB0,0xFE);
        NVIC_SystemReset();
    }
}
```

注：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（读保护,外部 Flash 的加密）

1.2.4.1 读保护

读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 的读保护，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 ISP/ICP 工具解除读保护时，芯片会对 Flash 进行擦除操作。

可用 ISP/ICP 工具对 IC 进行读保护与解除读保护操作，如下：

■ ICP 工具

读保护：设备操作---读保护---启用读保护

解除读保护：设备操作---读保护---解除读保护

■ Artery ISP Programmer 工具

读保护：使能/除能保护、启用读保护---下一步---是，即可将程序加密。

图 19. ISP 启用读保护



解除读保护：使能/除能保护、禁用读保护---下一步---是，即可将 Flash 解除加密。

图 20. ISP 解除读保护



■ Artery ISP Multi-Port Programmer 工具

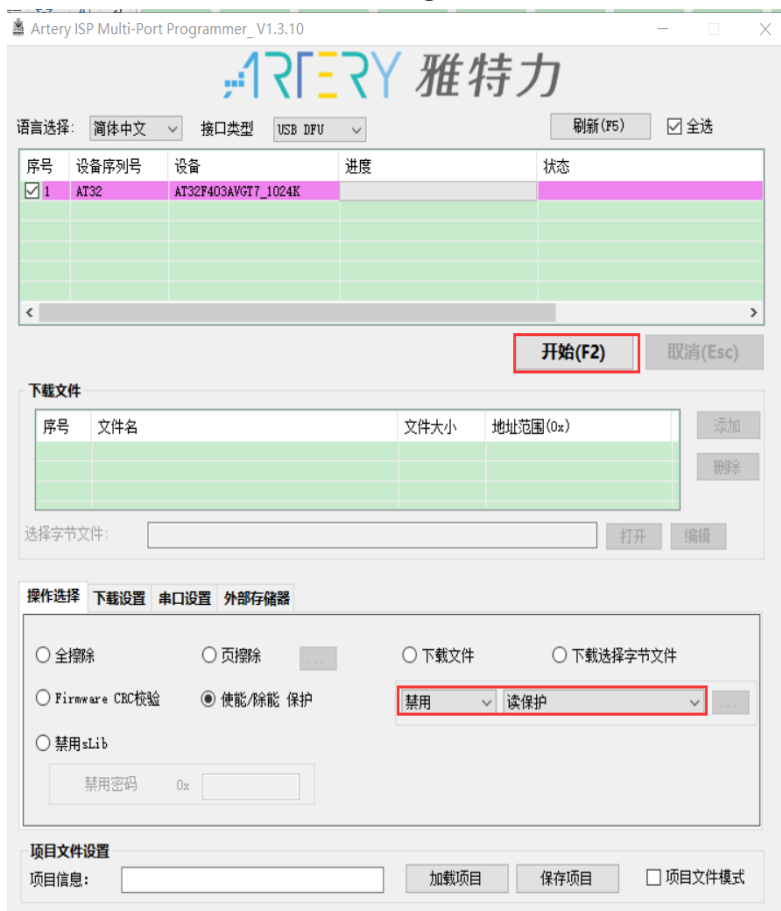
读保护：使能/除能保护、启用读保护---开始，即可将程序加密。

图 21. ISP Multi-Port Programmer 启用读保护



解除读保护：使能/除能保护、禁用读保护---开始，即可将 Flash 解除加密。设置了读保护不能通过擦除操作来解除读保护。

图 22. ISP Multi-Port Programmer 解除读保护



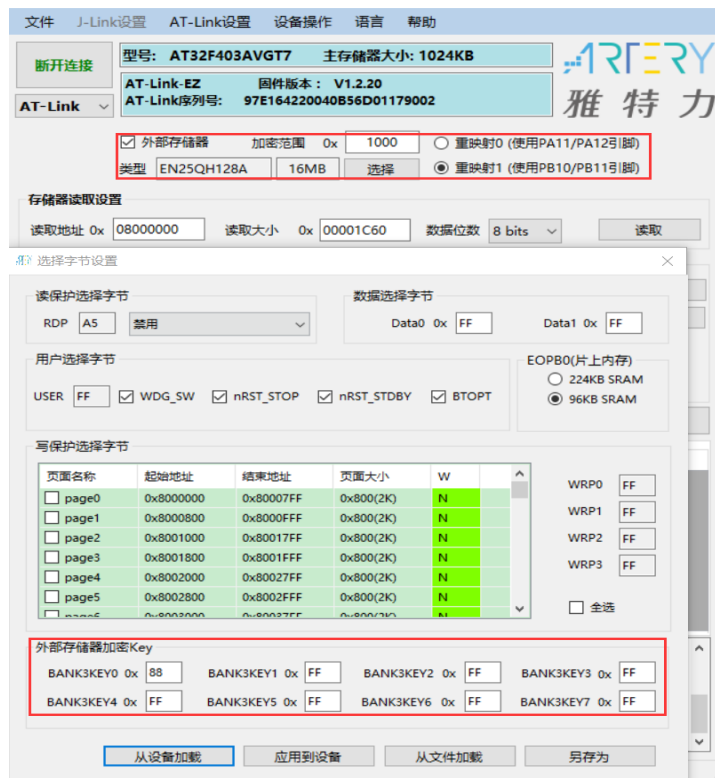
1.2.4.2 外部 Flash 的加密（下载和读取外部存储器加密范围内数据的加密）

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，然后开启读保护。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。解除读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：

■ ICP 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---选择字节---修改外部存储器加密 Key---应用到设备。

图 23. ICP 工具对外部存储器加密操作



■ Artery ISP Programmer 工具

编辑选择字节---下一步---修改外部存储器加密 Key---应用到设备。

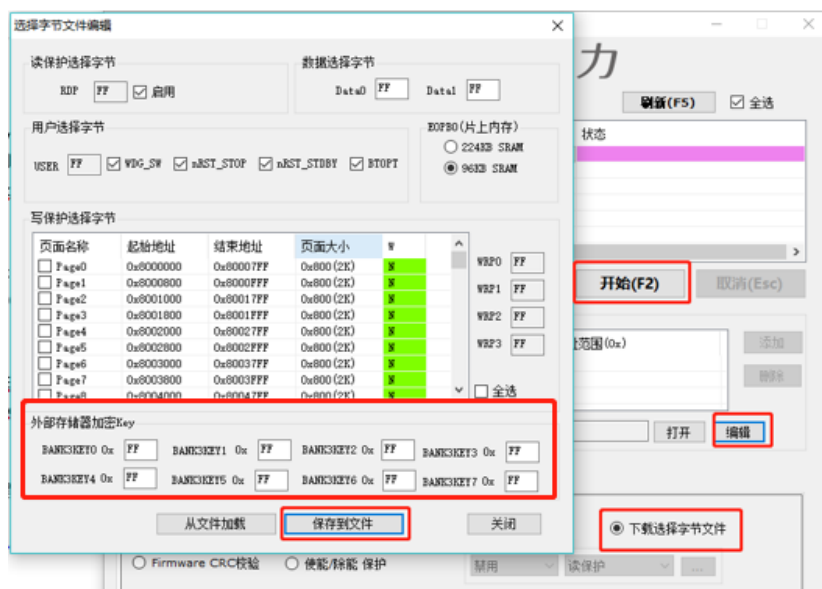
图 24. ISP 工具对外部存储器加密操作



■ Artery ISP Multi-Port Programmer 工具

下载选择字节文件---编辑---修改外部存储器加密 Key---保存到文件---开始。

图 25. ISP Multi-Port Programmer 工具对外部存储器加密



1.2.5 在程序中区分 AT 与其他 IC 方法

1.2.5.1 UID/PID 组合方式来识别

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

```
cortex_id = *(uint32_t *)0xE00ED00;           //读取 cortex 型号

if((cortex_id == 0x410FC240) || (cortex_id == 0x410FC241))
{
    printf("This chip is Cortex-M4.\r\n");
}
else
{
    printf("This chip is Other Device.\r\n");
}
```

■ 读取 PID,UID 方式区分

```
/* 获取 AT32 MCU 的 PID/UID 的基地址*/
#define DEVICE_ID_ADDR1 0x1FFF7F3
#define DEVICE_ID_ADDR2 0xE0042000

/* AT32F403A MCU type table */
const uint64_t AT32_MCU_ID_TABLE[] =
{
    0x0000000270050242,      //AT32F403ARCT7    256KB    LQFP64
    0x00000002700502CA,      //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

PROJECT ID: 访问地址为 0x1FFF7F3 [7:0]，定义 Artery MCU 的项目型号

DEVICE ID: 访问地址为 0xE0042000 [31:0]，定义 MCU 的设备型号

1.2.5.2 取 32 位作简缩的唯一 UID 码来识别

当客户不想读取 96 位的 UID（基地址：0x1FFF7F7E8 基地址：0x1FFF7F7EC 基地址：0x1FFF7F7F0），只想取其中的 32bit 当作简缩的唯一 UID 码，可以只取[87:79][33:28][16:0] bit 位

2 下载编译过程常见问题

2.1 程序启动进入 Hard Fault Handler

2.1.1 常见进入 Hardfault 异常状况

- 使用 SRAM 超过选择字节设置的 SRAM 空间大小。
请使用 ICP/ISP 或第三方烧录器开启后烧录程序。
- 在 Keil 或 IAR 上开启了 single precision 功能，在代码中并没有开启 M4 内核 FPU 寄存器，需在代码中开启 FPU 功能：

图 26.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
}
```

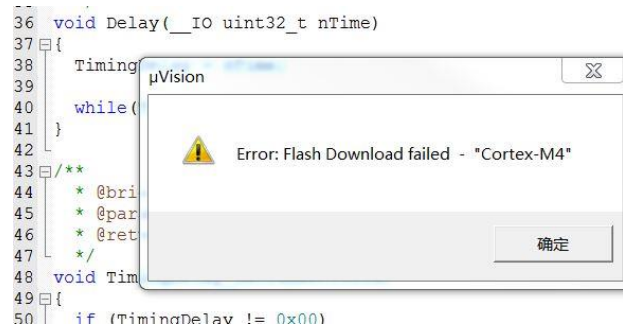
- 访问数据越界。
找到程序中访问越界的问题点，并修改它到正常数据区域内。

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



出现弹窗的原因可能是以下几种：

- 开启了读保护，先取消 MCU 读保护，然后再下载
- 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
- J-Link 驱动版本太低，建议 6.20C 以上版本

2.2.2 显示 No Debug Unit Device found 问题

- 下载端口被占用，比如说 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：

- 电源是否不稳定；
- 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_407 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-EZ方法

AT-Link-EZ 专为 AT32 设计，因此使用 ICP 加 AT-Link-EZ 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-EZ）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 ConfigJLink_V1.0.0.exe 工具

步骤如下：

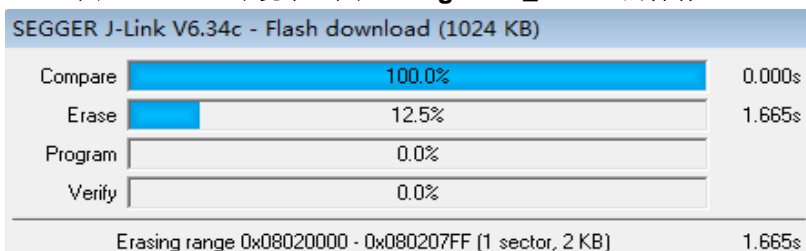
- ① 双击ConfigJLink_V1.0.0.exe，将会弹出如下对话框。

图 28. KEIL 环境中工具 ConfigJLink_V1.0.0 的操作



- ② 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 29. 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 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:

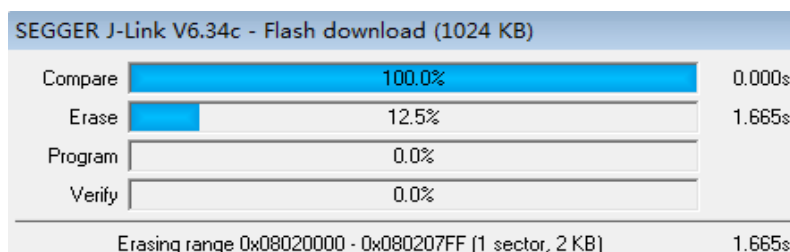
① 双击ConfigJLink_V1.0.0.exe, 将会弹出如下对话框。

图 30. IAR 环境中工具 ConfigJLink_V1.0.0 的操作



② 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 31. 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

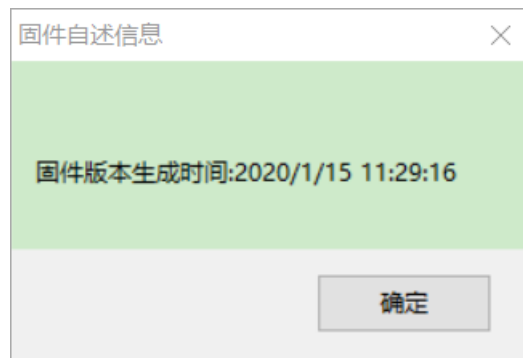
Note3: 若每次下载的程序都会进入 Standby mode 时, 每次芯片上电时都需要执行一遍上述步骤

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不要再按了(这时应是白屏或黑屏状态);
- 上位机软件会弹出升级窗口, 如下图, 点窗口的确定按钮自动升级固件, 无需任何操作(耐心等待提示升级成功大约一分钟);

图 32. 轩微烧录器固件生成时间



- 升级成功后提示固件更新成功，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 33. 轩微烧录器固件显示图片



注：进入显示屏上的固件升级是升级警告，并不是升级操作。

3 版本历史

表 2. 文档版本历史

日期	版本	变更
2020.3.16	1.0.0	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。

© 2020 雅特力科技 (重庆) 有限公司 保留所有权利