

目 录

1	雅特力初步环境准备	2
1.1	搭建 AT32 开发环境	2
1.1.1	调试工具及开发板	2
1.1.2	烧录工具及软件	2
1.1.3	AT32 KEIL 以及 IAR 开发环境	2
2	AT32F415 芯片的增强功能	4
2.1	预取指令缓冲	4
2.2	PLL 时钟设置	5
2.3	PLL 自动滑顺	5
2.4	加密	6
2.4.1	读保护	6
2.4.2	写保护	9
2.4.3	安全库区 sLib	12
2.5	设定系统存储器为扩展主存	12
3	在程序中区分 AT 与其他 IC 方法	15
4	下载编译过程常见问题	16
4.1	程序下载过程出问题	16
4.1.1	显示“error flash download failed cortex m4”问题	16
4.1.2	ISP 串口下载时卡死问题	16
4.1.3	AT32 恢复下载	16
4.2	升级轩微烧录器固件	18
5	版本历史	19

1 雅特力初步环境准备

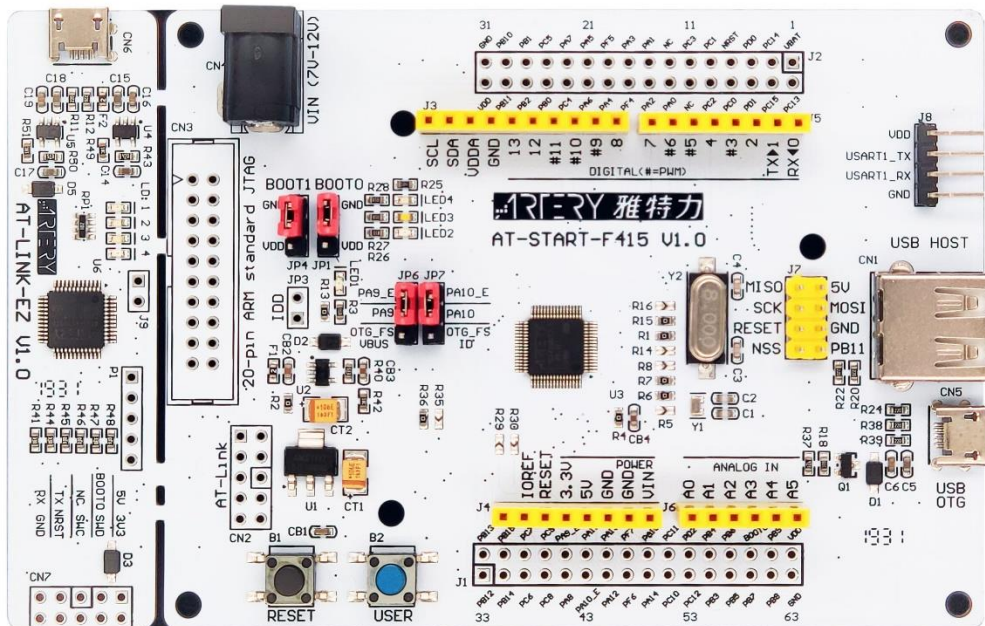
雅特力开发环境下载地址：

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 AT32F415 支持的调试工具有 AT-Link/J-Link，AT-START-F415 和 AT-Link-EZ 如下图所示，左边为 AT-Link-EZ，右边为 AT-START-F415。



1.1.2 烧录工具及软件

AT 烧录工具及软件：AT-Link、ICP/ISP。

第三方烧录工具：轩微：<https://xuanweikeji.taobao.com>

迈斯威志：www.maxwiz.com.cn

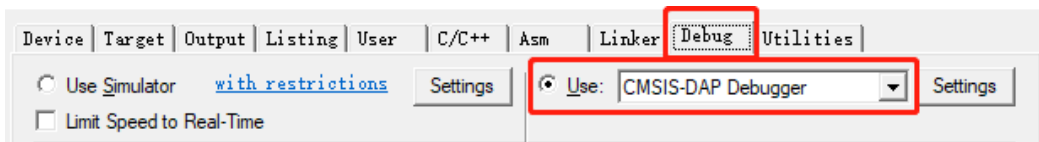
周立功：<http://tools.zlg.cn/tools>

阿莫：<http://www.amomcu.cn>

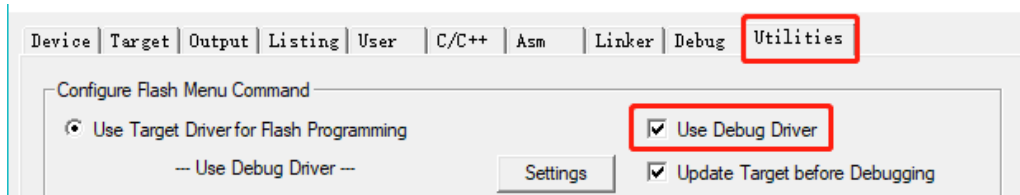
1.1.3 AT32 KEIL 以及 IAR 开发环境

①对于 Keil 编译系统，建议是 keil4.74 或 5.13 以上版本；

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

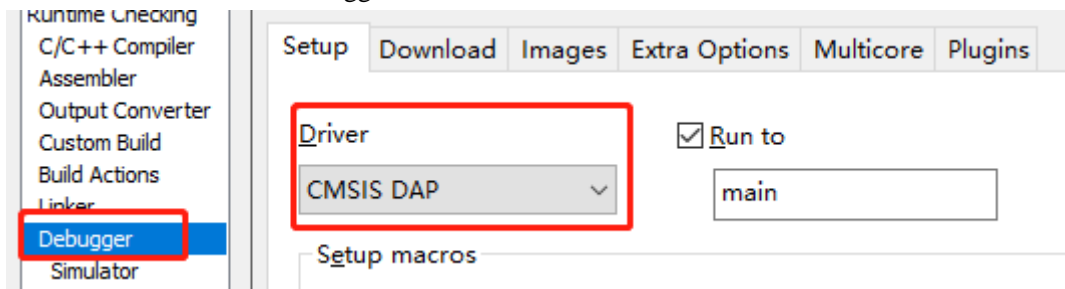


并且在 Utilities 里，下图选项点，需要先取消再勾选一次。



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，在 Debugger 里选择 CMSIS-DAP 调试器。



③ BSP 与 PACK

BSP 及 PACK 安装详细操作见“BSP 和 Pack 应用指南”，在下载开发包的 BSP 文件夹内。

2 AT32F415 芯片的增强功能

2.1 预取指令缓冲

设置预取缓冲器可以使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 SYSCLK 需要设置不同的时延，设置闪存访问控制寄存器（FLASH_ACR）的 bit2~0。

LATENCY: 时延

这些位表示 SYSCLK（系统时钟）周期与闪存访问时间的比例

000: 零等待状态，当 $0 < \text{SYSCLK} \leq 32 \text{ MHz}$

001: 一个等待状态，当 $32 \text{ MHz} < \text{SYSCLK} \leq 64 \text{ MHz}$

010: 两个等待状态，当 $64 \text{ MHz} < \text{SYSCLK} \leq 96 \text{ MHz}$

011: 三个等待状态，当 $96 \text{ MHz} < \text{SYSCLK} \leq 128 \text{ MHz}$

100: 四个等待状态，当 $128 \text{ MHz} < \text{SYSCLK} \leq 150 \text{ MHz}$

AT 库在系统频率配置函数，如 void SetSysClockTo72M (void)里设置如下黑体部分，其他方式找到相同的位置做相同的设置。

```
static void SetSysClockTo72M(void)
{
    ...
    ...
    if (HSEStatus == (uint32_t)0x01)
    {
        #if defined (AT32F415xx)
        /* Enable Prefetch Buffer */
        FLASH->ACR /= FLASH_ACR_PRFTBE;           //启用预取缓冲器
        /* Flash 1 wait state */
        FLASH->ACR &= (uint32_t)((uint32_t)~FLASH_ACR_LATENCY);
        FLASH->ACR /= (uint32_t)FLASH_ACR_LATENCY_2; //两个等待
        #endif
        ...
        ...
    }
    ...
    ...
}
```

注：当 AHB 时钟的预分频系数不为 1 时，必须打开预取缓冲器（FLASH_ACR[4]=1）。

2.2 PLL 时钟设置

设置 AT32F415 的 PLL 有两种方式：一种是使用传统的寄存器 RCC_CFG，一种是使用新增的寄存器 RCC_PLL。使用 RCC_PLL 配置可以配置更多种 PLL 时钟频率，其对应的公式为：PLL 输出时钟=PLL 参考输入时钟 \times PLL_NS / (PLL_MS \times PLL_FR)。

PLL 参考输入时钟：通过 PLL_FREF 设置为与实际使用 HSE 相同的参数或者 HIS

PLL_NS：范围（31~500）

PLL_MS 范围（1~15）

PLL_FR：通过 PLL_FR 设置

AT 库在系统频率配置函数，如 void SetSysClockTo150M (void)里设置如下黑体部分，HSE=8MHz。

```
static void SetSysClockTo150M(void)
{
    if (HSEStatus == (uint32_t)0x01)
    {
        ...
        ...
        /* PLL configuration: PLLCLK = (HSE * 75) / (1 * 4) = 150 MHz */
        RCC->CFG &= RCC_CFG_PLLCFG_MASK;
        RCC->CFG |= (uint32_t)(RCC_CFG_PLLRC_HSE);           //HSE 时钟作为 PLL 输入时钟
        RCC_PLLconfig2(PLL_FREF_8M, 75, 1, PLL_FR_4);       //设置 PLL 输出时钟
        ...
        ...
    }
}
```

注：使用 RCC_PLL 必须满足 500MHz <= PLL 参考输入时钟 \times PLL_NS/PLL_MS <= 1000MHz

2.3 PLL 自动滑顺

当AT32F415的PLL为108MHz以上时钟时，需要操作自动滑顺频率切换功能，AT库在系统频率配置函数，如 void SetSysClockTo150M (void)里设置如下黑体部分，其他方式找到相同的位置做相同的设置。

```
static void SetSysClockTo150M(void)
{
    ...
    ...
    if (HSEStatus == (uint32_t)0x01)
    {
        ...
        ...
        /* Wait till PLL is ready */
        while((RCC->CTRL & RCC_CTRL_PLLSTBL) == 0)
        {
        }
        #if defined (AT32F413xx) || defined (AT32F415xx)
        RCC_StepModeCmd(ENABLE);           // 开启自动滑顺频率切换功能
        #endif
        /* Select PLL as system clock source */
        RCC->CFG &= (uint32_t)((uint32_t)~(RCC_CFG_SYSCLOCKSEL));
        RCC->CFG |= (uint32_t)RCC_CFG_SYSCLOCKSEL_PLL;
        /* Wait till PLL is used as system clock source */
        while ((RCC->CFG & (uint32_t)RCC_CFG_SYSCLOCKSTS) != RCC_CFG_SYSCLOCKSTS_PLL)
        {
        }
        #if defined (AT32F413xx) || defined (AT32F415xx)
        RCC_StepModeCmd(DISABLE);         //关闭自动滑顺频率切换功能
        #endif
    }
    ...
}
```

注：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。

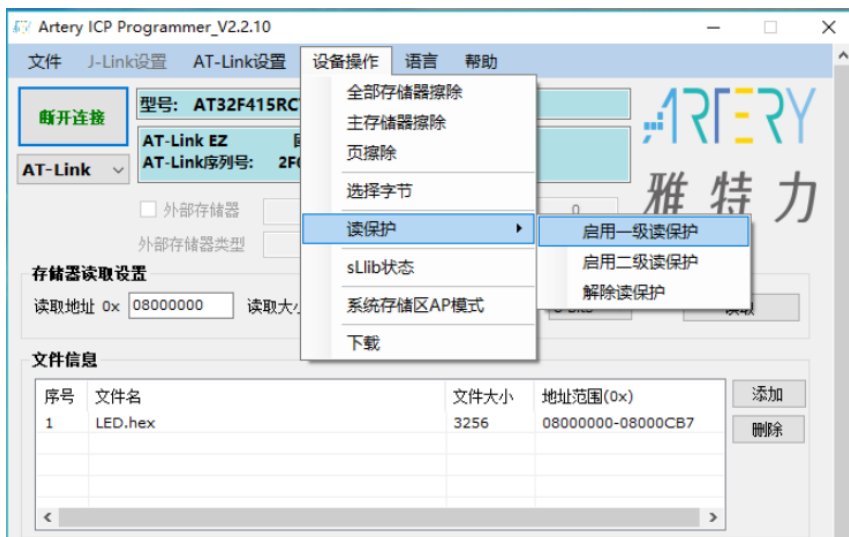
2.4 加密

2.4.1 读保护

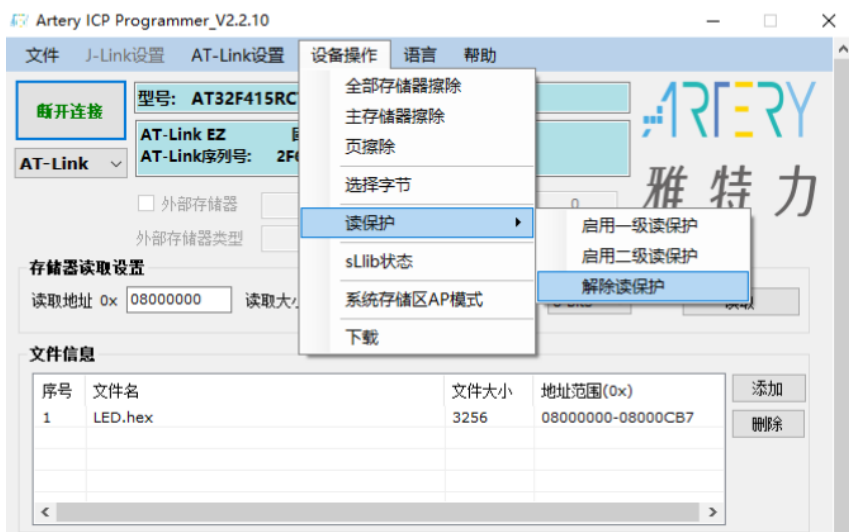
读保护即大家通常说的“加密”，作用于整个 Flash 存储区域。一旦设置了 Flash 的读保护，内置的 Flash 存储区只能通过程序的正常执行才能读出，而不能通过 JTAG 或者 SWD 读出，当使用 ISP/ICP 工具解除读保护时，芯片会对 FLASH 进行擦除操作。可用 ISP/ICP 工具对 IC 进行读保护与解除读保护操作。

①、Artery ICP Programmer 工具操作步骤

启用读保护：设备操作--读保护--启用一级/二级读保护



解除读保护：设备操作--读保护--解除读保护，设置了读保护不能通过擦除操作来解除读保护。

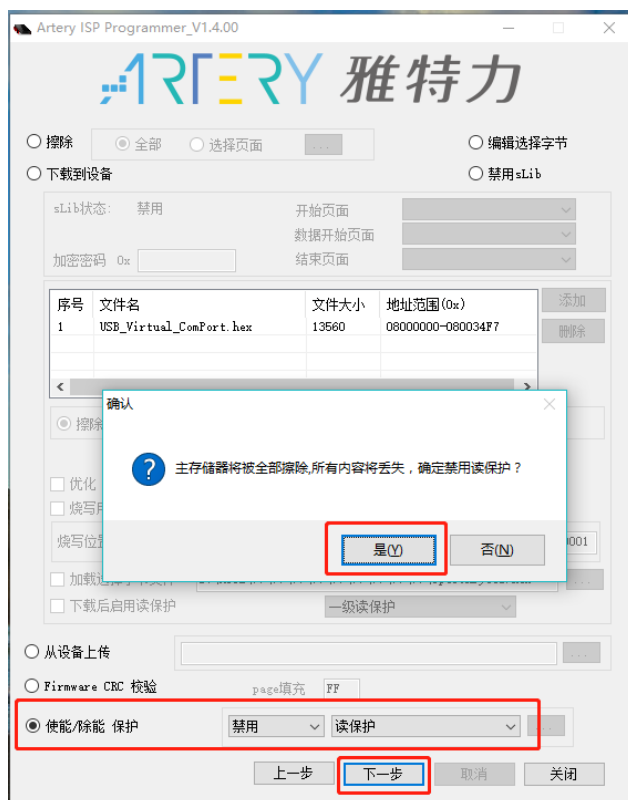


②、Artery ISP Programmer 工具操作步骤

启用读保护：使能/除能保护--启用--一级/二级读保护--下一步--是

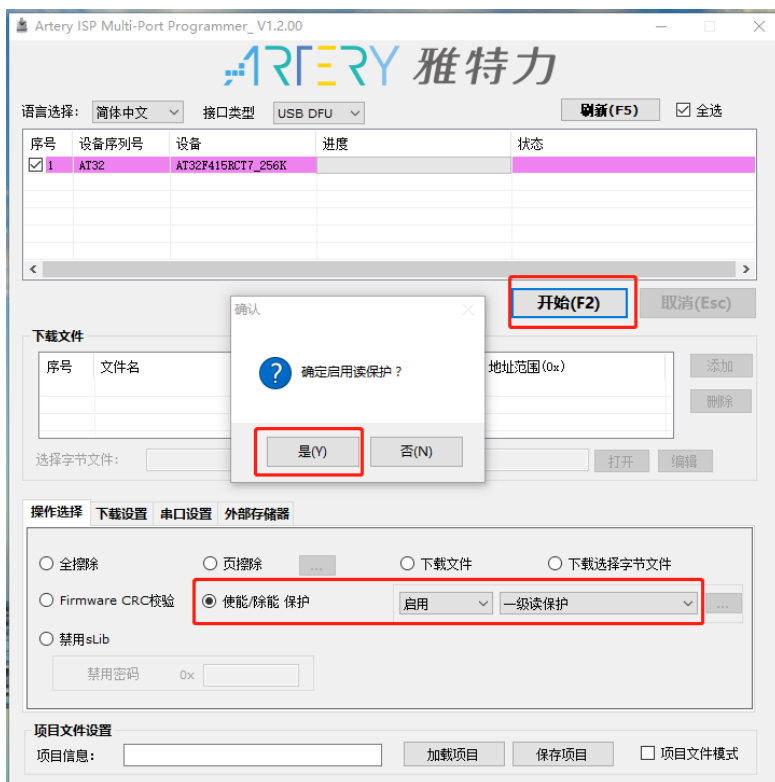


解除读保护：使能/除能保护--禁用--读保护--下一步--是，设置了读保护不能通过擦除操作来解除读保护。



③、Artery ISP Multi-Port Programmer 工具操作步骤

启用读保护：使能/除能保护--启用--一级/二级读保护--开始--是



解除读保护：使能/除能保护--禁用--读保护--开始--是，设置了读保护不能通过擦除操作来解除读保护。

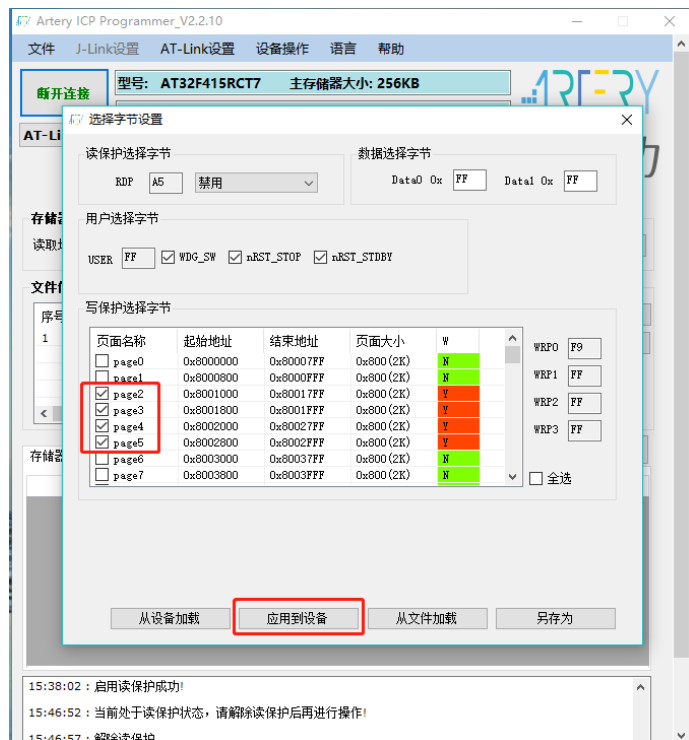


2.4.2 写保护

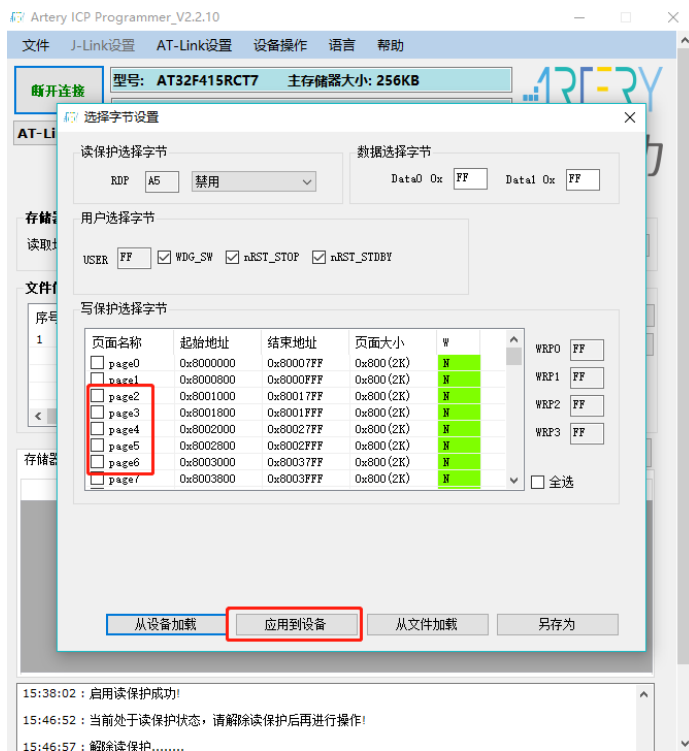
写保护作用于整个 Flash 存储区域或者 Flash 存储区域的某些页。一旦设置了 Flash 的写保护，内置的 Flash 存储区就不能通过任何方式写入。可用 ISP/ICP 工具对 IC 进行启用写保护与解除写保护操作。

①、Artery ICP Programmer 工具操作步骤

启用写保护：设备操作--选择字节--选择需保护的页--应用到设备

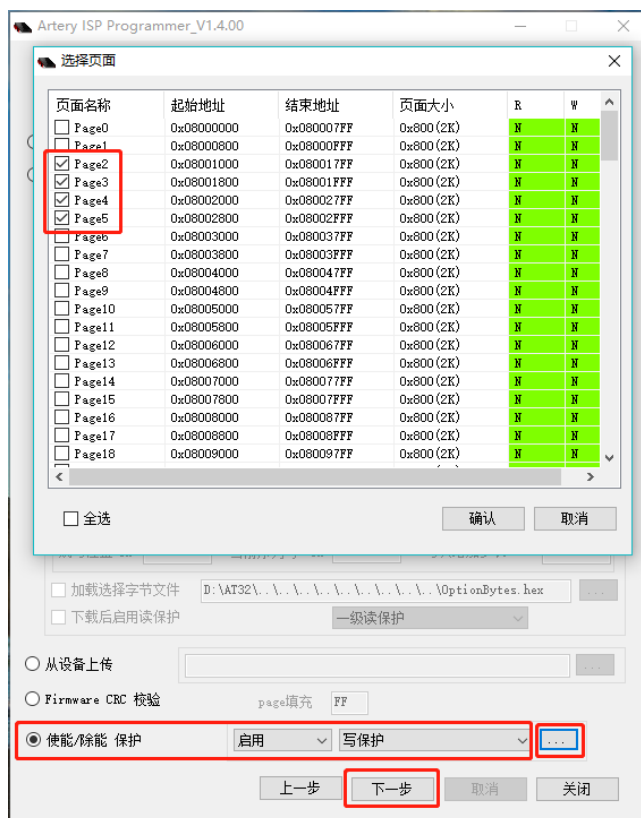


解除写保护：设备操作--选择字节--取消保护的页--应用到设备，设置了写保护不能通过擦除操作来解除写保护。

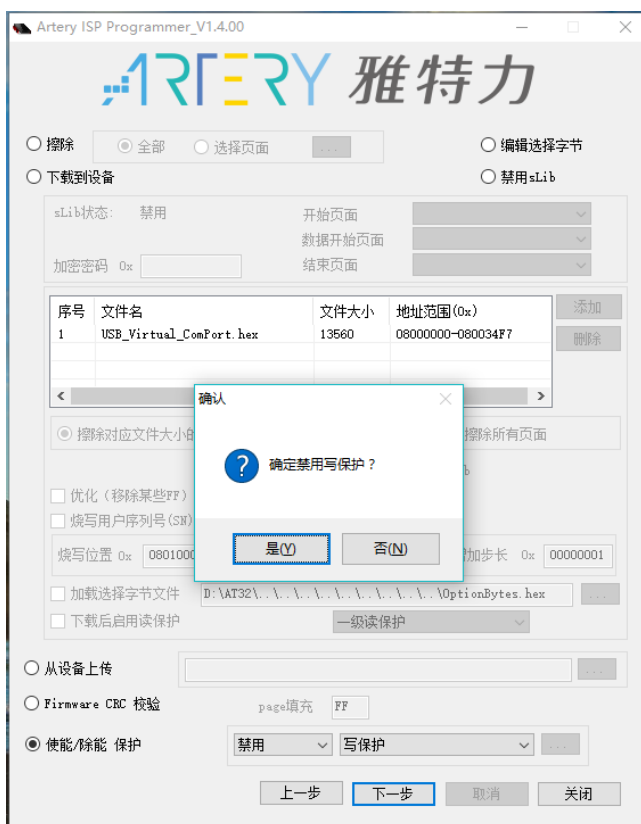


②、Artery ISP Programmer 工具操作步骤

启用写保护：使能/除能保护--启用--写保护--选择需保护的页--确定--下一步--是

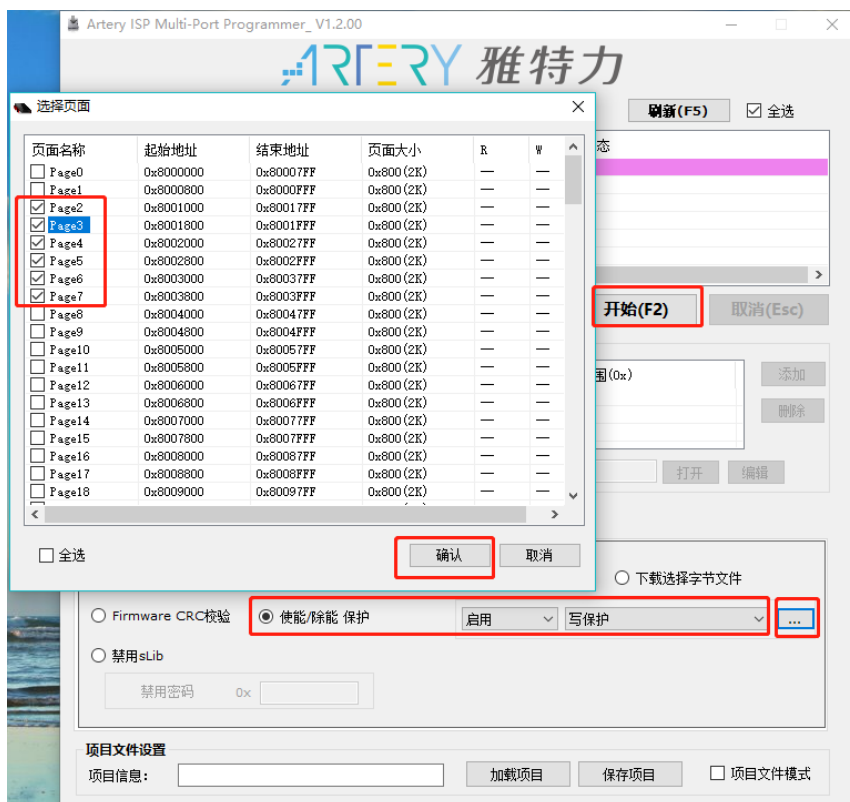


解除写保护：使能/除能保护--禁用--写保护--下一步--是，设置了写保护不能通过擦除操作来解除写保护。



③、Artery ISP Multi-Port Programmer 工具操作步骤

启用写保护：使能/除能保护--启用--写保护--选择需保护的页--确定--开始--是



解除写保护：使能/除能保护--禁用--写保护--开始--是，设置了写保护不能通过擦除操作来解除写保护。



2.4.3 安全库区sLib

● 概述

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

应这一重要需求，AT32F415 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。

● 应用原理

设定以密码保护主闪存中指定范围的程序区(即安全库区)，软件方案商可将核心算法存放到此区域，以达到保护的功能，其余空白程序区可以提供给终端商客户进行二次开发。

安全库区划分为指令安全库区(sLib_Code)及数据安全库区(sLib_Data)，并可选择部分或是整个安全库区存放指令，但不支持整个安全库区存放数据。

指令安全库区(sLib_Code)内的程序代码仅能被MCU透过I-Code总线抓取指令(仅能被执行)，不能透过D-Code总线以读取数据的方式读取(包含ISP/ICP调适模式以及从内部RAM启动的程序)，以读取数据的方式去访问sLib_Code时，读到的数值全都是0xFF。

数据安全库区(sLib_Data)的数据仅能透过D-Code总线读取，不能写入。

安全库区的程序代码及数据，除非输入正确的密码，无法被擦除。在密码不正确时，对安全库区执行写入或擦除，将会在FLASH_STS寄存器的WRPRTFLR位置"1"提出警告。

终端用户执行主闪存的整片擦除时，安全库区的程序代码及数据不会被擦除。

当安全库区的保护功能被启动后，可以透过在sLib_PSW寄存器写入先前设置的密码来解除保护功能。解除安全库区的保护时，芯片将会执行主闪存的整片擦除(包含安全库区的内容)。因此即使软件方案商设置的密码被泄漏，也不会有程序代码外泄的疑虑。

● 操作安全库区

详细操作见“AT32F415 安全库区(sLib)应用指南”。

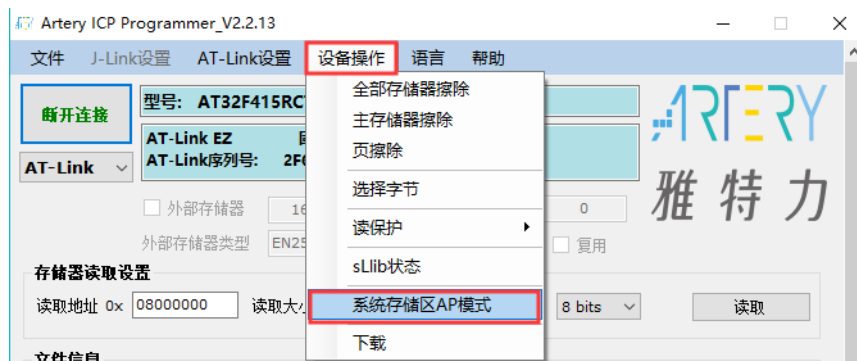
2.5 设定系统存储器为扩展主存

系统存储器 (System Memory) 默认是作为BOOT模式用来存放原厂固化的启动代码。不过，在AT32F415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：系统存储器 **AP** 模式只能设置一次不可逆，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 J-Link 或 AT-Link 仿真器到 AT-START-F415 板子上并上电。
- 开启 ICP programmer，选择用 J-Link 或 AT-Link 做连接。
- 通过菜单栏：设备操作— 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。

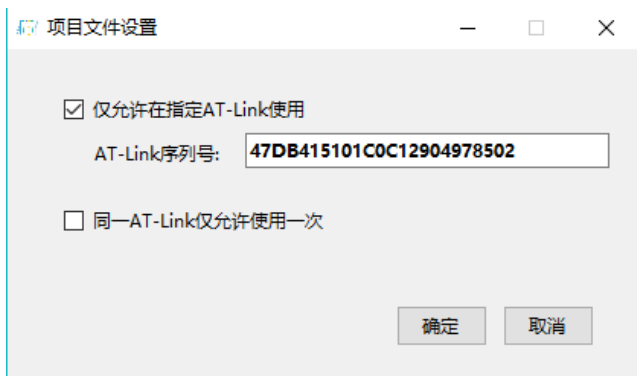


量产生成过程中使用 Artery ICP Programmer 将系统存储器作为扩展主存使用方法：

- 连接 AT-Link（上图中的 AT-Link EZ 版本不支持离线烧录，只能选择非 EZ 版本的 AT-Link）仿真器到 AT-START-F415 板子上并上电。
- 开启 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，开始生成离线项目。
- 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 模式并输入秘钥 — 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。



● 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；如果选择同一 AT-Link 仅允许使用一次，此项目文件在同一个 AT-Link 只能使用一次。



● 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，选择项目名称 — 保存并激活，就可以开始烧录了。



在系统存储器里运行用户程序的 demo 请参考 BSP，存放路径：

AT32F4xx_StdPeriph_Lib_V1.x.x\Utilities\AT32F415_SysMem_AP_Demo.

ICP 使用说明请参考 ICP_Programming_Tool_Vx.x.x\Document\UM_ICP_Programmer.

AT-Link 使用说明请参考 AT-Link_Vx.x.x\User Manual\ AT-Link_User_Manual_SC.

3 在程序中区分 AT 与其他 IC 方法

①、读取 Cortex-M 系列 CPU ID 号区分，此方式可以区分出 M0,M3,M4 内核；

```
i = *(uint32_t *)0xE000ED00;//读取 PID
if((i & 0xc241) == 0xc241)
    printf("This chip is Cortex-M4.\r\n");
else
    printf("This chip is Other Device.\r\n");
```

②、读取 PID,UID 方式区分：

```
/* 获取 AT32 MCU 的 PID/UID 的基地址*/
#define DEVICE_ID_ADDR1 0x1FFFF7F3
#define DEVICE_ID_ADDR2 0xE0042000
/* 用于存放 ID */
uint8_t ID[5] = {0};

/* AT32 mcu type table */
const uint64_t AT32_MCU_ID_TABLE[] = /* 415 mcu */
{
    0x0000000570030240, //AT32F415RCT7    256KB    LQFP64
    0x00000005700301C1, //AT32F415RBT7    128KB    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);

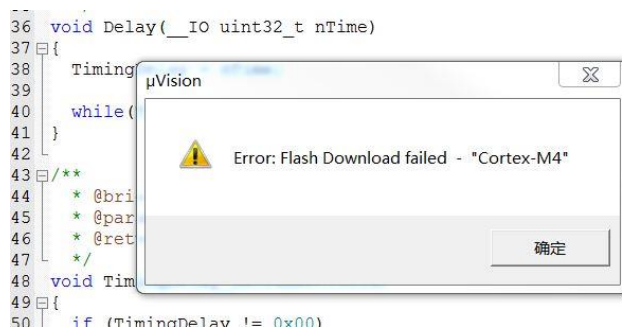
/* 判断 AT32 MCU */
for(i=0;i<sizeof(AT32_MCU_ID_TABLE)/sizeof(AT32_MCU_ID_TABLE[0]);i++)
{
    if(AT_device_id == AT32_MCU_ID_TABLE[i])
        printf("This chip is AT32F4xx.\r\n");
    else
        printf("This chip is Other Device.\r\n");
}
```

4 下载编译过程常见问题

4.1 程序下载过程出问题

4.1.1 显示“error flash download failed cortex m4”问题

在 KEIL 仿真或下载时弹出：



出现弹窗的原因可能是以下几种：

- A: 开启了读保护，先取消 MCU 读保护再下载；
- B: 选错了 FLASH 文件算法，在 FLASH DOWN 处选择正确的 FLASH 文件算法；
- C: Boot0、Boot1 选择错误，Boot0、Boot1 管脚电平须分别设置为 Boot0=0、Boot1=0，使 MCU 从主要闪存存储器启动；
- D: J-Link 驱动版本太低，建议 6.20C 以上版本；
- E: 在程序中将 JTAG/SWD PIN disable，解决方法参考“4.1.3 AT32 恢复下载”。

4.1.2 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：

- A: 是否电源不稳定；
- B: 更换质量更好的 USB 转串口工具，如 CH340 芯片等。

4.1.3 AT32恢复下载

在使用 AT32F415 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。
- 方法3 通过切换boot模式
切换boot模式到Boot[1:0]=01b，在系统存储器启动，使用ISP下载恢复下载。

方法 1:

使用 ARTERY 提供的 ConfigureJLink.exe 工具，存放在 Migration Guide 内。

步骤如下：

- 双击ConfigJLink_V1.0.0.exe，将会弹出如下对话框。



- 勾选同意后再点击确认，待弹出的如下擦除进度条执行完毕后即可正常下载程序。

SEGGER J-Link V6.34c - Flash download (1024 KB)			
Compare	100.0%	0.000s	
Erase	12.5%	1.665s	
Program	0.0%		
Verify	0.0%		
Erasing range 0x08020000 - 0x080207FF (1 sector, 2 KB)			1.665s

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

方法 2:

使用 AT-Link 连接到 AT32F415，如下图，在 ICP 进行擦除或者重新烧录操作。



方法 3:

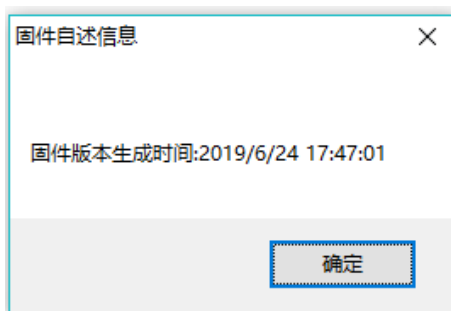
使用 UART 或者 USB DFU 方式连接到 AT32F415，在 ISP 进行擦除或者重新烧录程序。



4.2 升级轩微烧录器固件

升级方法步骤:

- A、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；
- B、按下蓝色按键，插入 Micro_USB2.0 接口，待绿色指示灯常亮后松开按键；
- C、自动升级固件，无需任何操作；
- D、升级成功提示固件更新时间，如下图。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。



注：进入显示屏上的固件升级是升级警告，并不是升级操作。

5 版本历史

日期	版本	变更
2019.08.23	1.00	最初版本
2019.11.22	1.01	增加设定系统存储器为扩展主存使用说明，修改AT32恢复下载

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。

雅特力的产品不得应用于武器。此外，雅特力产品也不是为下列用途而设计并不得应用于下列用途：(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) 汽车应用或汽车环境，且/或(D) 航天应用或航天环境。如果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，采购商仍将独自承担因此而导致的任何风险，雅特力的产品设计规格明确指定的汽车、汽车安全或医疗工业领域专用产品除外。根据相关政府主管部门的规定，ESCC、QML 或 JAN 正式认证产品适用于航天应用。

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。

© 2019 雅特力科技 (重庆) 有限公司 保留所有权利