

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	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
1.1.4 快速替代 SXX 流程	3
1.2 AT32F413 芯片的增强功能配置	4
1.2.1 PLL 大于 72MHz 设定	4
1.2.1.1 PLL 分段设置	4
1.2.1.2 PLL 自动滑顺	4
1.2.2 如何打开 FPU 功能（硬件浮点运算单元）	5
1.2.3 AT32F413 零等待/非零等待 Flash 和内置 SRAM 大小选择设置说明	6
1.2.4 加密方式(读保护,外部 Flash 的加密)	9
1.2.4.1 读保护	9
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)	11
1.2.5 在程序中区分 AT32 与其他 IC 方法	13
2 下载编译过程常见问题	14
2.1 程序启动进入 Hard Fault Handler	14
2.2 程序下载过程出问题	14
2.2.1 显示“Error: Flash Download failed – “Cortex-M4”问题	14
2.2.2 ISP 串口下载时卡死问题	14
2.2.3 AT32 恢复下载	14
2.2.3.1 KEIL 环境下的解决方法	15
2.2.3.2 IAR 环境下的解决方法	15
2.3 升级轩微烧录器固件	16
3 安全库区 sLib(Security Library)	17
3.1 概述	17
3.2 应用原理	17
3.3 操作安全库区	17
4 版本历史	18

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

目前 AT32F413 支持的调试工具有 AT-Link/J-Link，AT-Link 如下图所示。



1.1.2 烧录工具及软件

■ AT 烧录工具及软件：AT-Link、ICP/ISP。

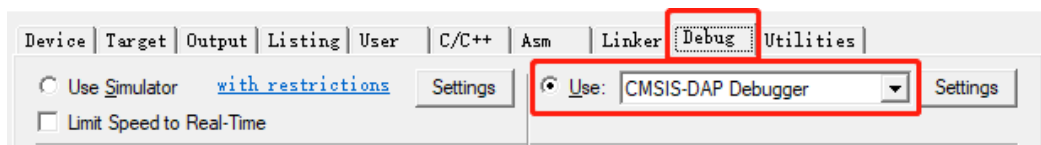
■ 第三方烧录工具：

- 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>

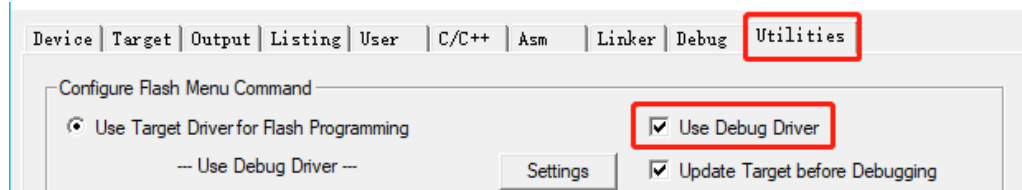
1.1.3 AT32 KEIL以及IAR开发环境

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。

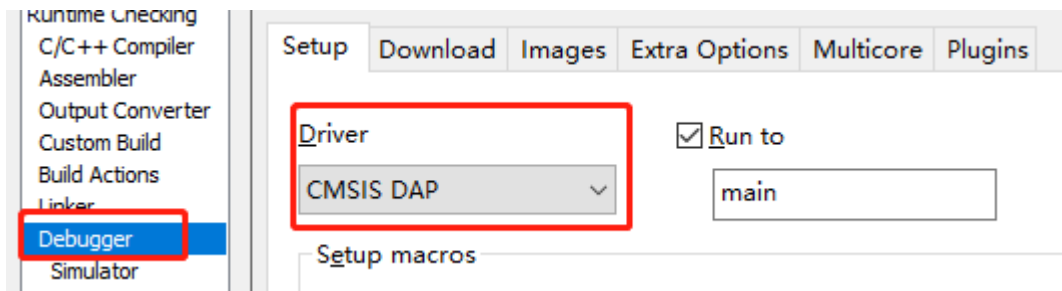


并且在 Utilities 里，下图选项点，需要先取消再勾选一次。



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 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)。

序号	使用什么 BSP/Pack	是否使用 AT32F413 新功能	处理方法
1	AT32F413 BSP/Pack	是/否	结合 MG_SXXF103 to AT32F413 的 3.2 外设使用区别 修改对应程序
2	SXX32F103 BSP/Pack	否	结合 MG_SXXF103 to AT32F413 的 3.2 外设使用区别 修改对应程序
3	SXX32F103 寄存器操作	否	结合 MG_SXXF103 to AT32F413 的 3.2 外设使用区别 修改对应程序
4	SXX32F103 BSP + AT32 Pack	否	1. 修改 FPU 设置 2. 结合 MG_SXXF103 to AT32F413 的 3.2 外设使用区别 修改对应程序
5	SXX32F103 寄存器操作	是	1. 需使用处理过的 BSP/Pack 2. 结合 MG_SXXF103 to AT32F413 的 3.2 外设使用区别 修改对应程序
6	SXX32F103 BSP/Pack	是	1. 需使用处理过的 BSP/Pack 2. 结合 MG_SXXF103 to AT32F413 的 3.2 外设使用区别 修改对应程序

BSP 及 PACK 安装详细操作见“AT32F413 BSP 和 Pack 应用指南”，在下载开发包的 PACK 文件夹内。

注：关于以下两个 Flash 等待位，由于 AT32 与 SXX 的 Flash 机制不同，不需要设置以下两个位：

```
/* Enable Prefetch Buffer */
FLASH->ACR |= FLASH_ACR_PREFBE;
/* 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，AT32F413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AT32F413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”[MG_SXXF103 to AT32F413](#)”；
- 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。

- *Note: 由于AT32F413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AT32F413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004_Performance_Optimization.*

1.2 AT32F413 芯片的增强功能配置

1.2.1 PLL大于72MHz设定

1.2.1.1 PLL 分段设置

AT32F413内置的PLL最高可输出200MHz时钟，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。

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

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

AT32F413 PLL 设定程序范例：(HSE=8MHz)

```
#define RCC_CFG_PLLRANGE_LE72MHZ ((uint32_t)0x00000000)
/*!< When PLL frequency is less than or equal to 72MHz */
#define RCC_CFG_PLLRANGE_GT72MHZ ((uint32_t)0x80000000)
/*!< When PLL frequency is greater than 72MHz */
```

例如 PLL=72MHz 时，设置如下：

```
RCC->CFG |= (uint32_t)(RCC_CFG_PLLRC_HSE | RCC_CFG_PLLMULT9 | RCC_CFG_PLLRANGE_LE72MHZ);
```

例如 PLL=200MHz 时，设置如下：

```
RCC->CFG |= (uint32_t)(RCC_CFG_PLLRC_HSE | RCC_CFG_PLLMULT25 | RCC_CFG_PLLRANGE_GT72MHZ);
```

1.2.1.2 PLL 自动滑顺

当AT32F413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。

打开 system_sxx32f10x.c 找到当前的系统时钟频率配置函数（需经过 2.1.1 PLL 配置），如 168MHz 函数：

static void SetSysClockTo168(void) 增加如下斜黑体部分：

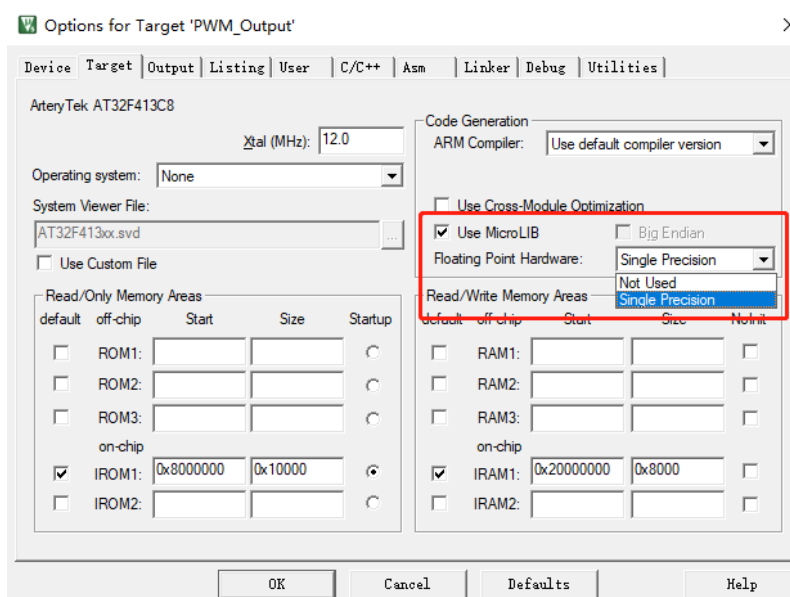
```
/* Wait till PLL is ready */
while((RCC->CR & RCC_CR_PLLRDY) == 0)
{
}
*((unsigned int *)0x40021054) |= (0x30); // 开启自动滑顺频率切换功能
/* Select PLL as system clock source */
RCC->CFGR &= (uint32_t)((uint32_t)~(RCC_CFGR_SW));
RCC->CFGR |= (uint32_t)RCC_CFGR_SW_PLL;
/* Wait till PLL is used as system clock 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
```

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。

1.2.2 如何打开FPU功能（硬件浮点运算单元）

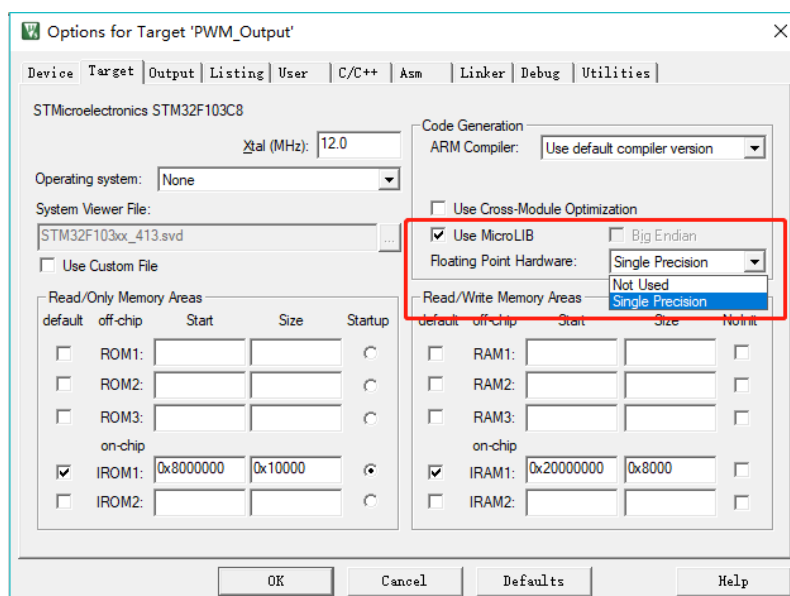
分为以下两种情况:

- ① 使用 AT32F413 BSP/Pack 或者修改过的 SXX BSP/Pack 直接修改 Floating Point Hardware 如下图:



- 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:

- 首先需按照“AT32F413 BSP 和 Pack 应用指南”文件的第 5，第 6 种情况来修改相关头文件;
- 需选择 AT 对应型号，然后在 Options---Target 中如下图选择;



- 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。

```
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
}
```

1.2.3 AT32F413 零等待/非零等待Flash和内置SRAM大小选择设置说明

除了AT32F413C8T7外，其他型号透过选择字节设置支持内部闪存存储器和SRAM分配使用。

以AT32F413RCT7为例，内部闪存存储器和SRAM可以设置为以下三种配置：

- ZW: 96 KB, NZW: 160 KB, SRAM: 32 KB (出厂默认);
- ZW: 64 KB, NZW: 192 KB, SRAM: 64 KB;
- ZW: 112 KB, NZW: 144 KB, SRAM: 16 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，AT32F413零等待有96KB大小，128KB bin文件的前面96KB可以200MHz速率执行，后面32KB 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 16KB/32KB(默认)/64KB大小可以通过以下任何方式选择：

AT32F413 SRAM大小设置涉及FMC选择字节说明，通过配置EOPB0选择，地址是：0x1FFF_F810。

EOPB0=0x0(0x2)表示片上SRAM为64KB，EOPB0=0x1表示片上SRAM为16KB，EOPB0=0x3表示片上SRAM为32KB。使能EOPB0有效务必要掉电或RESET一次。

① 使用ICP/ISP方式

■ ICP工具

设备操作---选择字节---选择16KB/32KB/64KB---应用到设备。



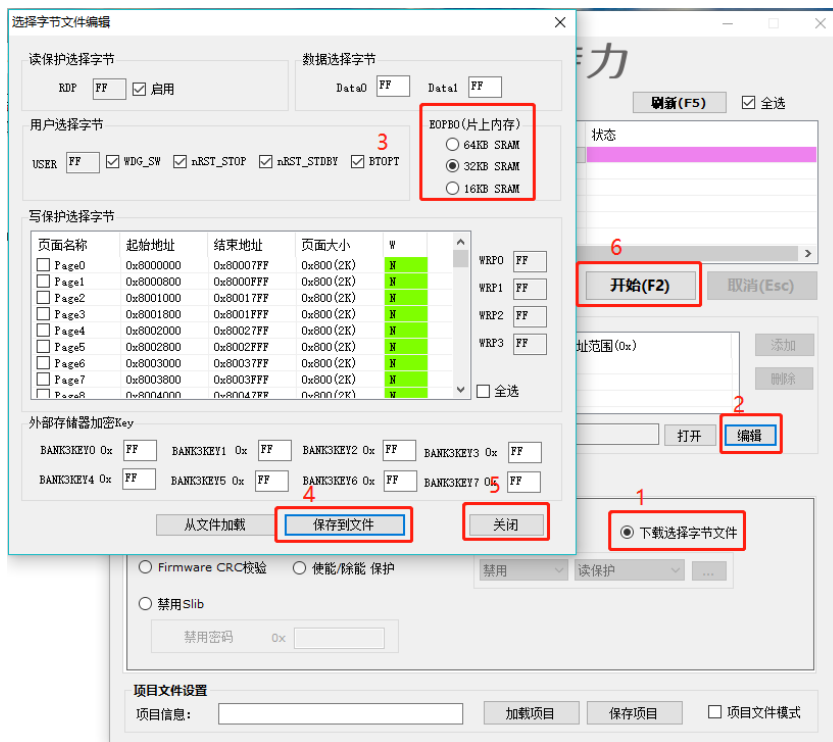
■ Artery ISP Programmer 工具

进入最后界面，选择 **16KB/32KB/64KB**---应用到设备。



■ Artery ISP Multi-Port Programmer 工具

下载选择字节文件---编辑---选择 **16KB/32KB/64KB**---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。



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 AT32F413 MCU 来选择 SRAM 大小。

③ 客户也可以使用在 Bootloader program (IAP) 内修改 SRAM 的大小

范例如下：

```
#define SRAM_16k 0x1
#define SRAM_64k 0x2
static uint32_t F_EOPB0;
F_EOPB0=*(uint32_t*)(0x1FFFFF810);
if((F_EOPB0 & 0x00000003) == 0x3) //请务必读取一次后做判断，这样修改一次后，后续不用再次擦写；
{
    FLASH_Unlock();
    FLASH_EraseOptionBytes();
    FLASH_ProgramOptionByteData(0x1FFFFF810, SRAM_64k);
    ...           //设置其他选择字节
    FLASH_Lock();
}
```

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(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。



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。



■ 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。



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。



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。



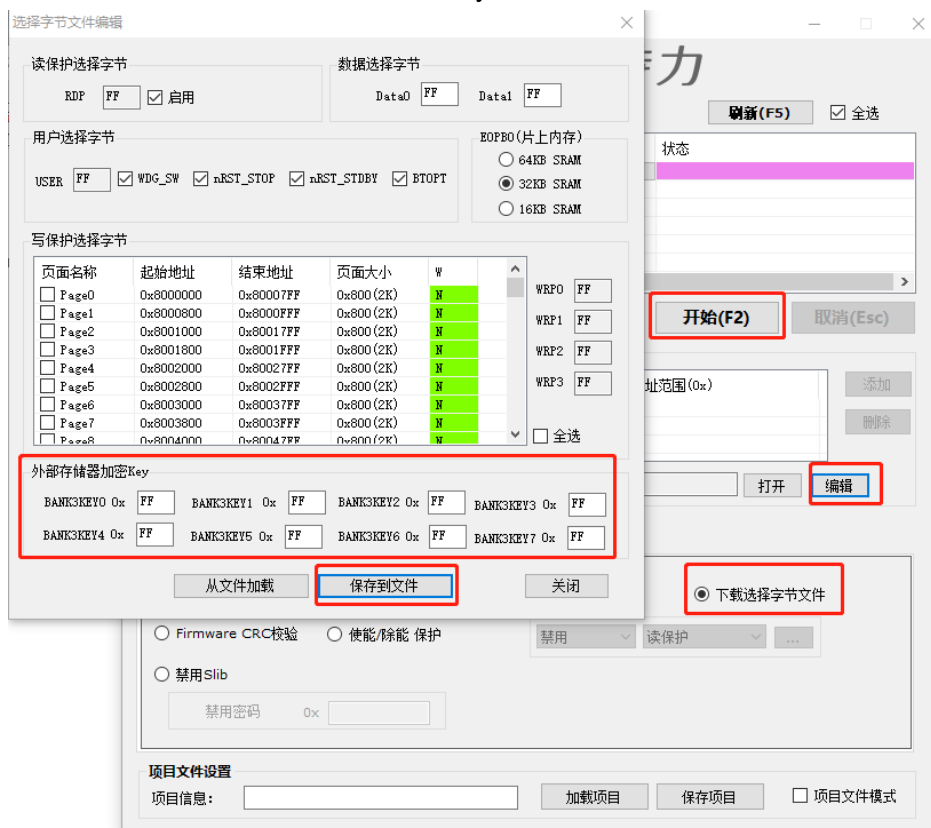
- Artery ISP Programmer 工具

编辑选择字节---下一步---修改外部存储器加密 Key---应用到设备。



■ 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。



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

```
i = *(uint32_t *)0xE000ED00; //读取 PID

if((i & 0xc241) == 0xc241)
{
    printf("This chip is Cortex-M4F.\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 0x1FFFF7F3
#define DEVICE_ID_ADDR2 0xE0042000

/* AT32F413 MCU type table */
const uint64_t AT32_MCU_ID_TABLE[] =
{
    0x0000000470030240, //AT32F413RCT7    256KB    LQFP64
    0x00000004700301C1, //AT32F413RBT7    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");
    }
}
```

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 超过选择字节设置的 SRAM 空间。
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 code 中并没有开启 M4 内核 FPU 寄存器。在 code 中开启 FPU 功能：

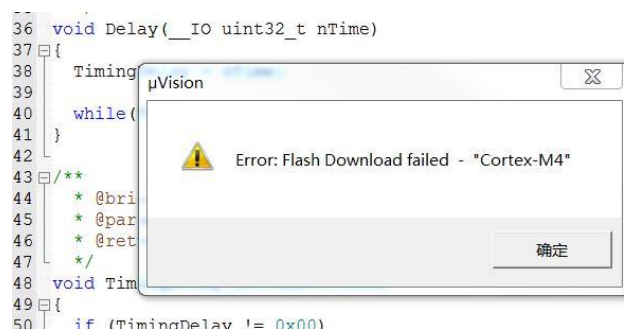
```
void SystemInit (void)
{
    /* Enable FPU*/
    #if defined (__FPU_USED) && (__FPU_USED == 1U)
        SCB->CPACR |= ((3U << 10U * 2U) |          /* set CP10 Full Access */
                       (3U << 11U * 2U) );        /* set CP11 Full Access */
    #endif
}
```

- 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：



出现弹窗的原因可能是以下几种：

- 开启了读保护，先取消 MCU 读保护再下载；
- 选错了 Flash 文件算法，在 Flash Download 处选择正确的 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 从主要闪存存储器启动；
- 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“2.3 AT32 恢复下载”。

2.2.2 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：

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

2.2.3 AT32恢复下载

在使用 AT32F413 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方法

AT-Link 专为 AT32 设计，因此使用 ICP 加 AT-Link 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）支持，本文重点介绍方法 1。

2.2.3.1 KEIL 环境下的解决方法

使用 ARTERY 提供的 ConfigureJLink.exe 工具

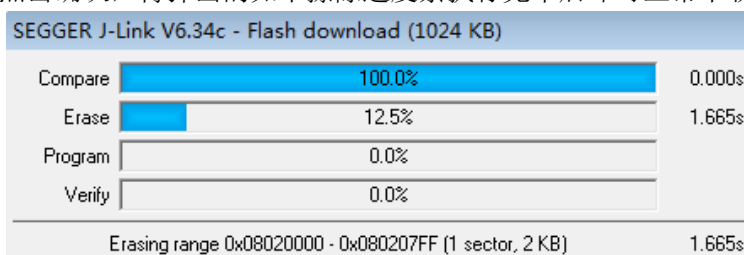
步骤如下：

■ 将ConfigJLink_V1.0.0.exe工具放入工程文件（*.uvprojx）所在目录，

■ 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。



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

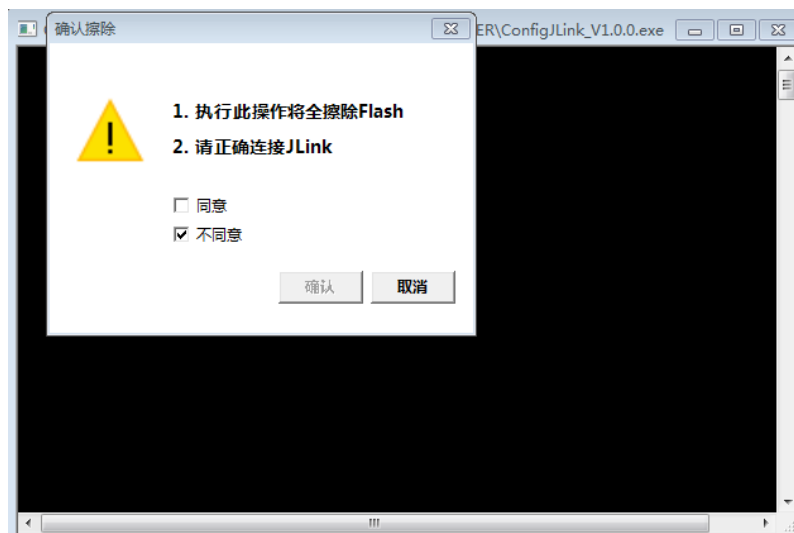
Note4: Keil 环境下，AT32F413 芯片，进入 Standby mode 后，使用该工具解法无效

2.2.3.2 IAR 环境下的解决方法

使用 ARTERY 提供的 ConfigureJLink.exe 工具

步骤如下：

■ 将ConfigJLink_V1.0.0.exe工具放入工程目录下的settings文件夹，然后双击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

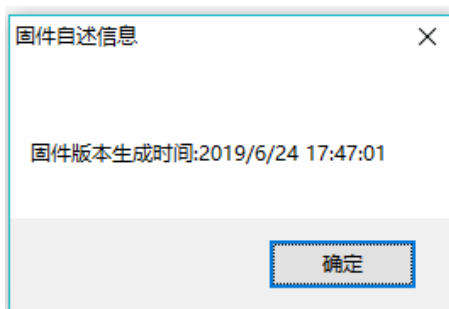
Note2: 若每次下载的程序都会 disable JTAG/SWD PIN 时，每次下载程序前都需要执行一遍上述步骤

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

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；
- 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；
- 自动升级固件，无需任何操作；
- 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。

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，AT32F413 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。

3.3 操作安全库区

详细操作见“安全库区(sLib)应用指南”。

4 版本历史

日期	版本	变更
2019.7.24	1.00	最初版本
2021.1.21	1.01	勘误

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。