

AT32F421入门手册

前言

这篇入门手册旨在让用户快速使用AT32F421 MCU进行项目开发

支持型号列表：

支持型号	AT32F421xx
------	------------

目录

1	雅特力初步环境准备.....	5
1.1	调试工具.....	5
1.2	烧录工具及软件.....	5
1.3	AT32 KEIL 以及 IAR 开发环境.....	5
1.4	快速替代 SXX 流程.....	7
2	AT32F421 芯片的增强功能配置.....	8
2.1	PLL 最高可输出 120 MHz.....	8
2.2	安全库区保护.....	8
2.3	系统存储器区域作为主存扩展使用.....	8
2.4	加密方式（读保护）.....	8
2.5	在程序中区分 AT 与其他 IC 方法.....	11
2.5.1	UID/PID 组合方式来识别.....	11
2.5.2	取 32 位作简缩的唯一 UID 码来识别.....	12
3	下载编译过程常见问题.....	13
3.1	程序启动进入 Hard Fault Handler.....	13
3.1.1	常见进入 Hardfault 异常状况.....	13
3.2	程序下载过程出问题.....	13
3.2.1	显示 Error: Flash Download failed – “Cortex-M4” 问题.....	13
3.2.2	显示 No Debug Unit Device found 问题.....	13
3.2.3	显示 RDDI-DAP Error 问题.....	13
3.2.4	ISP 串口下载时卡死问题.....	13
3.3	AT32 恢复下载.....	13
3.3.1	使用 ConfigureJLink.exe 工具的解决方法.....	14
4	版本历史.....	15

表目录

表 1 文档版本历史.....	15
-----------------	----

图目录

图 1. AT32F421 开发板实物图.....	5
图 2. Keil Debug 选项.....	6
图 3. Keil Debug 选项 settings 设置.....	6
图 4. Keil Utilities 选项.....	6
图 5. IAR Debug 选项.....	6
图 6. IAR CMSIS-DAP 选项.....	7
图 7. ISP 启用读保护.....	9
图 8. ISP 解除读保护.....	9
图 9. ISP Multi-Port Programmer 启用读保护.....	10
图 10. ISP Multi-Port Programmer 解除读保护.....	10
图 11. 下载出现 Flash Download failed – “Cortex- 4”.....	13
图 12. 工具 ConfigJLink_V1.0.0 的操作 1.....	14
图 13. 工具 ConfigJLink_V1.0.0 的操作 2.....	14

1 雅特力初步环境准备

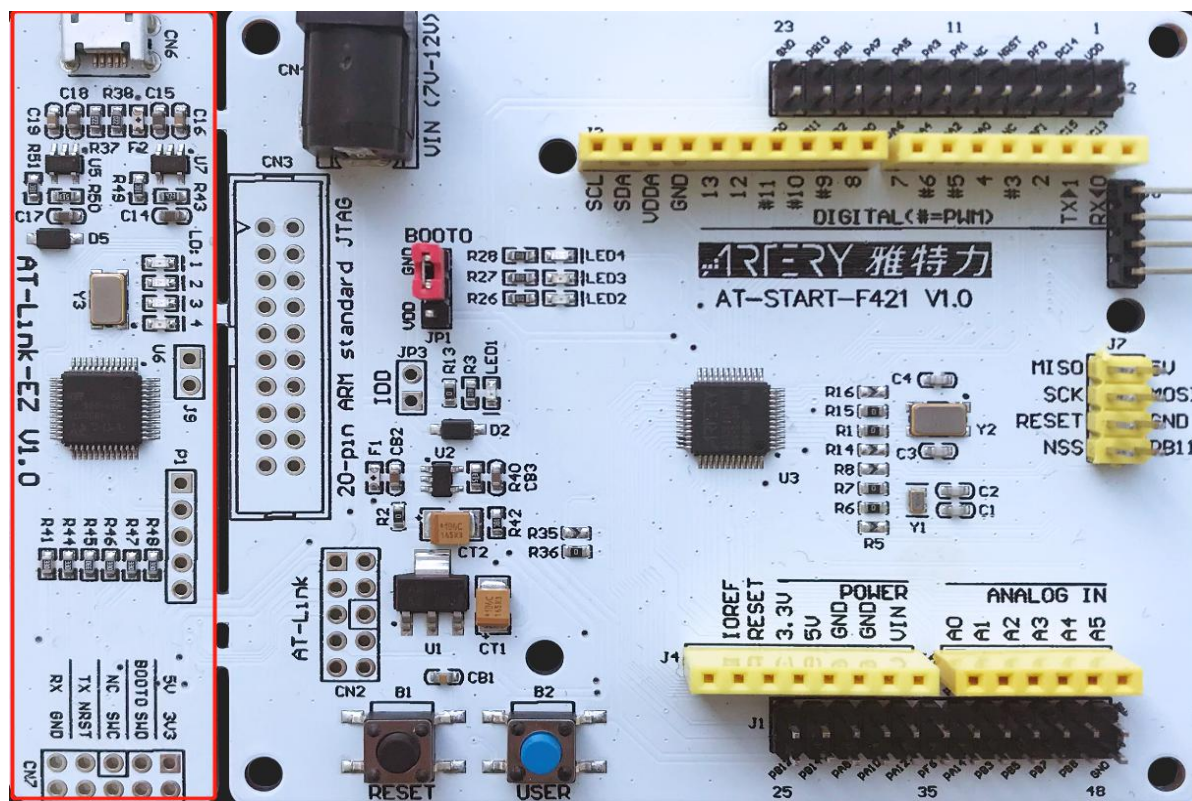
雅特力 MCU 资料下载地址:

■ 雅特力官方网站: <http://www.arterytek.com>

1.1 调试工具

目前 AT32F421 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. AT32F421 开发板实物图



1.2 烧录工具及软件

AT 烧录工具及软件: AT_Link, AT_Link_EZ, AT_Link_Pro, 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>

1.3 AT32 KEIL 以及 IAR 开发环境

对于 Keil 编译系统, 建议是 Keil 4.74 或 5.23 以上版本;《Keil_AT32F4xx_AddOn.exe》的 Pack 支援 Keil_v4,《Keil_AT32F4xx_DFP.1.3.2.pack》的 Pack 支援 Keil_v5, 都采用‘双击’完成一键式安装。

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



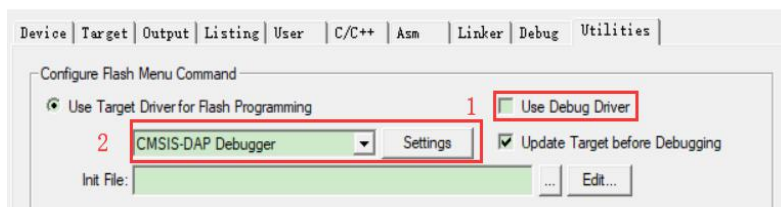
在 Debug 单击 settings 进入 cortex - M Target Driver Setup 界面如下图，1，先选择的 AT-Link-EZ-CMSIS-DAP
2， Port: 选择 SW，再勾选 SWJ 3， 确认已识别到 ARM SWD 调试模块

图 3. Keil Debug 选项 settings 设置



并且在 Utilities 里，勾掉 1 选项框，手动选择 2 下拉菜单选择 CMSIS-DAP Debugger， 或者勾选 1（自动选择）。

图 4. Keil Utilities 选项



对于 IAR 编译系统,建议 IAR7.0 或 IAR6.1 以上版本;《IAR_AT32F4xx_AddOn.exe》的 Pack 支援 IAR_v6 和 IAR_v7, 采用‘双击’完成一键式安装。

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

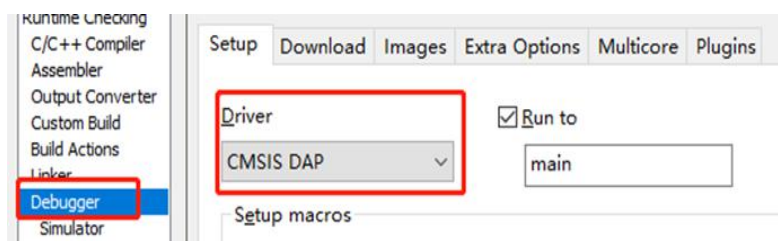
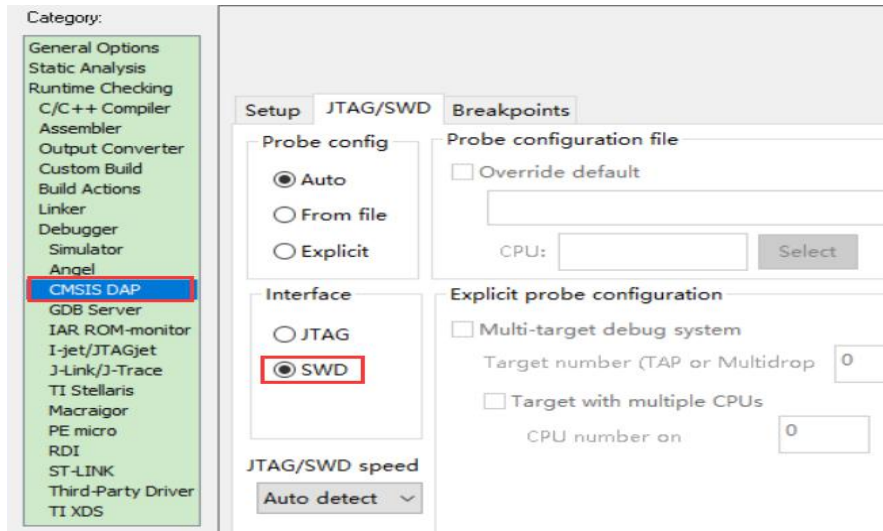


图 6. IAR CMSIS-DAP 选项



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 SXX32F030，换成AT32F421对应型号；
- 步骤二：下载AT32F421 最新的 BSP 和 PACK 文件；
- 步骤三：更新工具启动文件；
 - A. 项目文件：器件连接和 Flash 加载程序。这些文件随支持 AT32F421x 器件的最新版本工具一起提供。有关详细信息，请参见相关工具文档。
 - B. 链接器配置和向量表位置文件：这些文件根据 CMSIS 标准开发并且包含在AT32F421 库安装包内，位置如下：Libraries\CMSIS\CM4\DeviceSupport。
- 步骤四：将 AT32F421 库源文件添加到应用程序源；
 - A. 用 AT32F421 库中提供的 at32f4xx_conf.h 文件替换应用程序的 sxx32f0xx_conf.h 文件。
 - B. 用 AT32F421 库中提供的at32f4xx_it.c/ at32f4xx_it_it.h 文件替换应用程序的现有 sxx32f0xx_it.c/ sxx32f0xx_it.h 文件。
 - C. 用 AT32F421 库中提供的system_at32f4xx.c/ system_at32f4xx.h 文件替换应用程序的现有 system_sxx32f0xx.c/ system_sxx32f0xx.h 文件。
- 步骤五：更新使用 RCC、PWR、GPIO、FLASH、ADC 和 RTC 驱动程序的部分应用程序代码。
- 步骤六：其他问题快速排查请参考《从SXX32F030移植到AT32F421 移转手册_V1.x.x》；
- 步骤七：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: AT32F421 库随附丰富的示例（总共 100 多个），具体演示了使用不同外设的方法（位于 AT32F4xx_StdPeriph_Lib_V1.x.x\Project\AT_START_F421\Examples 下）

2 AT32F421 芯片的增强功能配置

2.1 PLL 最高可输出 120 MHz

AT32F421内置的PLL最高可输出 120 MHz时钟，AT32F421 和 SXX32 F030 系列的时钟源配置过程相同。但是，二者产品的电压范围、PLL 配置、最大频率和 Flash 等待周期配置有所不同。利用 CMSIS 层，可从应用程序代码中隐藏这些区别；用户只需使用 `system_at32f4xx.c` 文件替换 `system_sxx32f0xx.c` 文件。此文件提供了 `SystemInit()` 函数一种实现方法，利用该函数，可在启动时和转到 `main()` 程序之前配置微控制器系统。

详见AT32F421 PLL设定程序范例：

AT32F4xx_StdPeriph_Lib_V1.x.x\Project\AT_START_F421\Templates中`system_at32f4xx.c`的 `static void SetSysClockTo120M(void)`函数

2.2 安全库区保护

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。

为因应这一重要的需求，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。

详见使用范例

■ 请参考AT32F421 Security Library Application Note

■ Demo 请参考 AT32F4xx_StdPeriph_Lib_Vx.x.x\Utilities\AT32F421_SLIB_Demo

2.3 系统存储器区域作为主存扩展使用

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。但是系统存储器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，详见 AN0066__设定系统存储器为主存扩展 (AP模式) 应用指南为主存扩展使用

2.4 加密方式（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 7. 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。

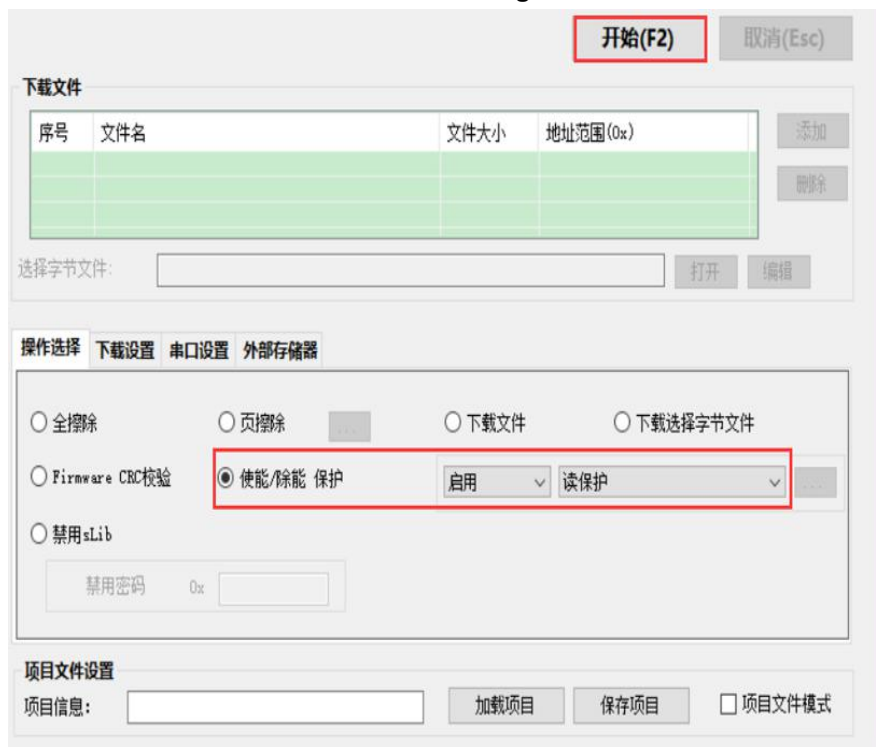
图 8. ISP 解除读保护



■ Artery ISP Multi-Port Programmer 工具

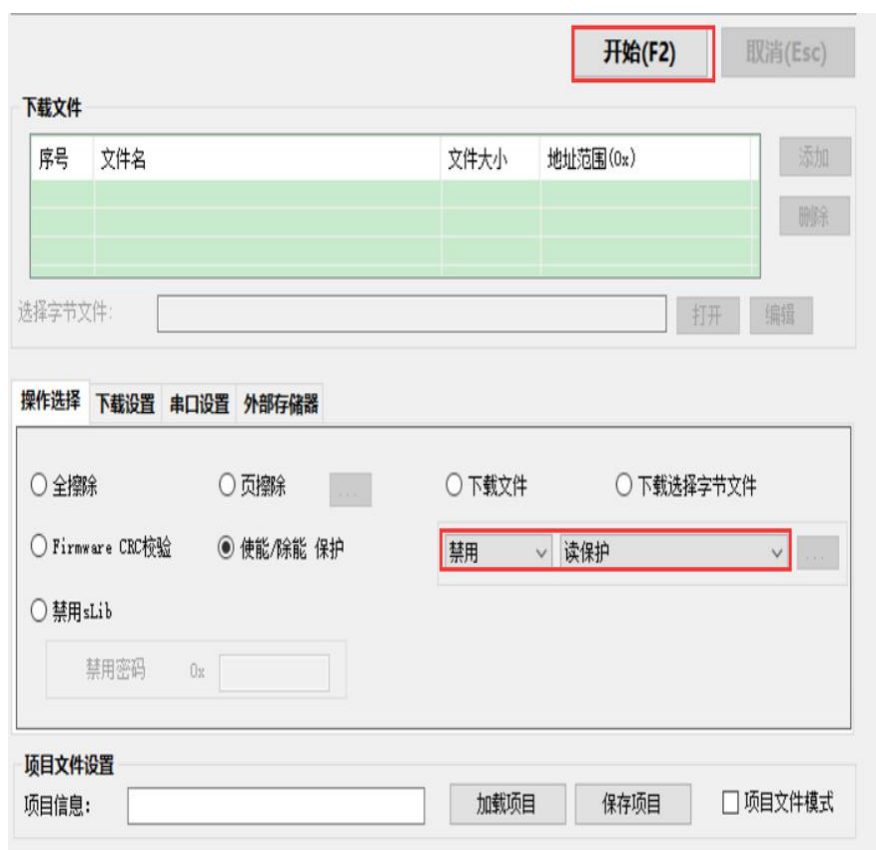
读保护：使能/除能保护、启用读保护---开始，即可将程序加密。

图 9. 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---下一步---是，即可将 Flash 解除加密。

图 10. ISP Multi-Port Programmer 解除读保护



2.5 在程序中区分 AT 与其他 IC 方法

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 *)0xE000ED00;           //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
/* AT32F421 MCU type table */

/* 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");
    }
}

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);
```

F4xx 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 0x1FFF 7F3 [7:0]，定义 Artery MCU 的项目型号

DEVICE ID: 访问地址为 0xE004 2000 [31:0]，定义 MCU 的设备型号

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：0x1FFF F7E8 基地址：0x1FFF F7EC 基地址：0x1FFF F7F0），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

3 下载编译过程常见问题

3.1 程序启动进入 Hard Fault Handler

3.1.1 常见进入 Hardfault 异常状况

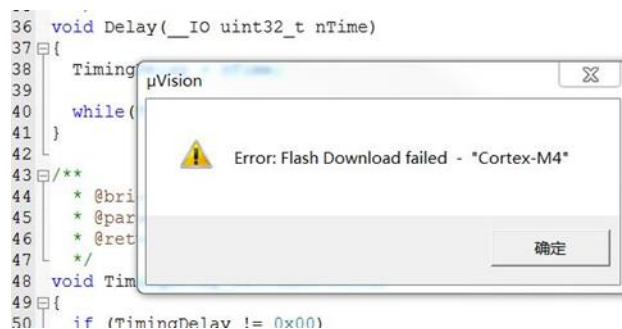
- 程序使用 SRAM 超过 MCU SRAM 空间大小。
- 访问数据越界。
- 系统时钟设置超出规格

3.2 程序下载过程出问题

3.2.1 显示 Error: Flash Download failed – “Cortex-M4” 问题

在 KEIL 仿真或下载时弹出：

图 11. 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
- 启用了 Slib 安全库区，禁用 Slib 安全库区，然后再下载
- J-Link 驱动版本太低，建议 6.20C 以上版本

3.2.2 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

3.2.3 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”

3.2.4 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。

3.3 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 后，无法下载程序并且找不到 JTAG/SWD device，这里提供以下几种解决方法：
 - 方法一：使用 ARTERY 提供的 ConfigureJLink.exe 工具
 - 方法二：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。同理使用 ISP 下载也可恢复下载。
 - 方法三：通过 ICP tool 加 AT-Link/AT-Link-EZ 方法，AT-Link/AT-Link-EZ 专为 AT32 设计，因此使用 ICP 加 AT-Link/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/AT-Link-EZ）支持，本文重点介绍方法一

3.3.1 使用 ConfigureJLink.exe 工具的解决方法

使用 ARTERY 提供的 ConfigJLink_V1.0.0.exe 工具

步骤如下：

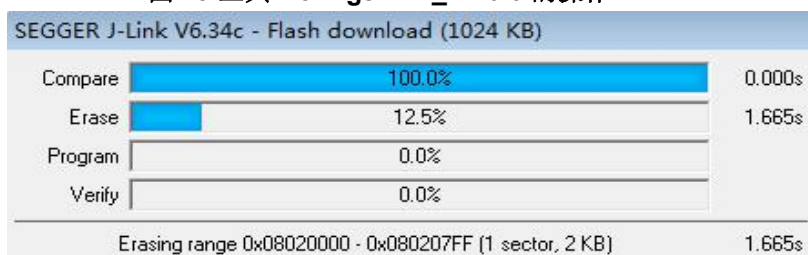
1. 首先连接 J_LINK，双击 ConfigJLink_V1.0.0.exe，将会弹出如下对话框。

图 12.工具 ConfigJLink_V1.0.0 的操作 1



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 13.工具 ConfigJLink_V1.0.0 的操作 2



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

4 版本历史

表 1 文档版本历史

日期	版本	变更
2020.8.17	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