

# AN0040

应用笔记

## AT32F403A/407 Security Library Application Note

# 前言

这篇应用笔记主要在阐述AT32F403A/407系列安全库区的应用原理、软件使用方法及范例程序。

支持型号列表:

十壮刑口	AT32F403A
又苻至亏	AT32F407

# 目录

1	概述	概述			
2	应用	原理		8	
	2.1	安全库	军区的应用原理	8	
	2.2	如何启	自动安全库区保护		
	2.3	如何角	释除安全库区保护		
	2.4	编排及	及执行安全库区的程序	11	
		2.4.1	不可将中断向量表设置为安全库区	12	
		2.4.2	安全库区代码与用户区代码的关联性	12	
3	安全	库区范	例程序	15	
	3.1	范例需	<b>宗求</b>		
		3.1.1	硬件需求	15	
		3.1.2	软件需求	15	
	3.2	范例椆	既述		
	3.3	安全库	军区保护的代码: FIR 低通滤波器		
	3.4	Projec	ct_L0: 方案商范例	17	
		3.4.1	产生只执行(Execute-only)代码	17	
		3.4.2	编排安全库区的地址	19	
		3.4.3	启用安全库区保护	23	
		3.4.4	Project_L0 执行流程	24	
		3.4.5	产生头文件及符号定义文件	26	
	3.5	Projec	ct_L1:终端用户范例	27	
		3.5.1	建立用户的应用项目		
		3.5.2	在项目中加入符号定义文件		
		3.5.3	调用 SLIB 保护区的函数	29	
		3.5.4	Project_L1 执行流程		
		3.5.5	调试模式下的 SLIB 保护		

4	方案	商和终端用户代码整合及下载操作流程	. 34
	4.1	方案商和终端用户代码分别烧录	. 34
	4.2	方案商和终端用户代码合并烧录	. 37
5	版本	历史	. 40

# 表目录

表 1. AT32F403A/407 各型号闪存大小总表	9
表 2. 文档版本历史	

# 图目录

图 1. 巾有女主岸区的王内伊区叭别	
图 2. 文字池例子(1)	
图 3. 文字池例子(2)	
图 4. 安全库区的函数调用用户区函数的例子	
图 5. 自定义函数范例	14
图 6. 范例流程示意图	
图 7. 应用示意图	
图 8. FIR 低通滤波器	
图 9. Keil 进入 Option 界面	
图 10. Keil 选择 Execute-only Code	
图 11. IAR 进入 Option 界面	
图 12. IAR 设置 C/C++窗口选项	
图 13. 范例程序的主闪存映像及 RAM 分区	
图 14. Keil 设置 Linker 窗口选项	
图 15. Keil scatter 修改	21
图 16. KEIL 代码中 SLIB 使用的 RAM 地址修改	21
图 16. KEIL 代码中 SLIB 使用的 RAM 地址修改 图 17. KEIL 代码中 SLIB 使用的常量地址修改	
图 16. KEIL 代码中 SLIB 使用的 RAM 地址修改 图 17. KEIL 代码中 SLIB 使用的常量地址修改 图 18. icf 文件中 SLIB 地址定义	
图 16. KEIL 代码中 SLIB 使用的 RAM 地址修改 图 17. KEIL 代码中 SLIB 使用的常量地址修改 图 18. icf 文件中 SLIB 地址定义 图 19. icf 文件中地址分配	
<ul> <li>图 16. KEIL 代码中 SLIB 使用的 RAM 地址修改</li> <li>图 17. KEIL 代码中 SLIB 使用的常量地址修改</li> <li>图 18. icf 文件中 SLIB 地址定义</li> <li>图 19. icf 文件中地址分配</li> <li>图 20. icf 文件中 IP-Code 使用的 RAM 修改</li> </ul>	21 21 22 22 22 22
<ul> <li>图 16. KEIL 代码中 SLIB 使用的 RAM 地址修改</li> <li>图 17. KEIL 代码中 SLIB 使用的常量地址修改</li> <li>图 18. icf 文件中 SLIB 地址定义</li> <li>图 19. icf 文件中地址分配</li> <li>图 20. icf 文件中 IP-Code 使用的 RAM 修改</li> <li>图 21. IAR 代码中 SLIB 使用的常量地址修改</li> </ul>	21 21 22 22 22 22 22 22
<ul> <li>图 16. KEIL 代码中 SLIB 使用的 RAM 地址修改</li> <li>图 17. KEIL 代码中 SLIB 使用的常量地址修改</li> <li>图 18. icf 文件中 SLIB 地址定义</li> <li>图 19. icf 文件中地址分配</li> <li>图 20. icf 文件中 IP-Code 使用的 RAM 修改</li> <li>图 21. IAR 代码中 SLIB 使用的常量地址修改</li> <li>图 22. 配置 ICP Programmer</li> </ul>	21 21 22 22 22 22 22 22 22 22 22 23
<ul> <li>图 16. KEIL 代码中 SLIB 使用的 RAM 地址修改</li> <li>图 17. KEIL 代码中 SLIB 使用的常量地址修改</li> <li>图 18. icf 文件中 SLIB 地址定义</li> <li>图 19. icf 文件中地址分配</li> <li>图 20. icf 文件中 IP-Code 使用的 RAM 修改</li> <li>图 21. IAR 代码中 SLIB 使用的常量地址修改</li> <li>图 22. 配置 ICP Programmer</li> <li>图 23. 设置下载选项参数</li> </ul>	21 21 22 22 22 22 22 22 22 22 22 22 23 23 24
<ul> <li>图 16. KEIL 代码中 SLIB 使用的 RAM 地址修改</li> <li>图 17. KEIL 代码中 SLIB 使用的常量地址修改</li> <li>图 18. icf 文件中 SLIB 地址定义</li> <li>图 19. icf 文件中地址分配</li> <li>图 20. icf 文件中 IP-Code 使用的 RAM 修改</li> <li>图 21. IAR 代码中 SLIB 使用的常量地址修改</li> <li>图 22. 配置 ICP Programmer</li> <li>图 23. 设置下载选项参数</li> <li>图 24. Project_L0 执行流程</li> </ul>	21 21 22 22 22 22 22 22 22 23 23 24 25
<ul> <li>图 16. KEIL 代码中 SLIB 使用的 RAM 地址修改</li> <li>图 17. KEIL 代码中 SLIB 使用的常量地址修改</li> <li>图 18. icf 文件中 SLIB 地址定义</li> <li>图 19. icf 文件中地址分配</li> <li>图 20. icf 文件中 IP-Code 使用的 RAM 修改</li> <li>图 21. IAR 代码中 SLIB 使用的常量地址修改</li> <li>图 22. 配置 ICP Programmer</li> <li>图 23. 设置下载选项参数</li> <li>图 24. Project_L0 执行流程</li> <li>图 25. 设置 Keil Misc controls 选项</li> </ul>	21 21 22 22 22 22 22 22 23 23 24 24 25 26
<ul> <li>图 16. KEIL 代码中 SLIB 使用的 RAM 地址修改</li> <li>图 17. KEIL 代码中 SLIB 使用的常量地址修改</li> <li>图 18. icf 文件中 SLIB 地址定义</li> <li>图 19. icf 文件中 地址分配</li> <li>图 20. icf 文件中 IP-Code 使用的 RAM 修改</li> <li>图 21. IAR 代码中 SLIB 使用的常量地址修改</li> <li>图 22. 配置 ICP Programmer</li> <li>图 23. 设置下载选项参数</li> <li>图 24. Project_L0 执行流程</li> <li>图 25. 设置 Keil Misc controls 选项</li> <li>图 26. 修改后的 fir_filter_symbol.txt 内容</li> </ul>	21 21 22 22 22 22 22 22 22 23 23 24 24 25 26
<ul> <li>图 16. KEIL 代码中 SLIB 使用的 RAM 地址修改</li> <li>图 17. KEIL 代码中 SLIB 使用的常量地址修改</li> <li>图 18. icf 文件中 SLIB 地址定义</li> <li>图 19. icf 文件中 NP-Code 使用的 RAM 修改</li> <li>图 20. icf 文件中 IP-Code 使用的 RAM 修改</li> <li>图 21. IAR 代码中 SLIB 使用的常量地址修改</li> <li>图 22. 配置 ICP Programmer</li> <li>图 23. 设置下载选项参数</li> <li>图 24. Project_L0 执行流程</li> <li>图 25. 设置 Keil Misc controls 选项</li> <li>图 26. 修改后的 fir_filter_symbol.txt 内容</li> <li>图 27. 设置 IAR Build Actions 选项</li> </ul>	21 21 22 22 22 22 22 22 23 23 24 25 26 26 26 27
<ul> <li>图 16. KEIL 代码中 SLIB 使用的 RAM 地址修改</li> <li>图 17. KEIL 代码中 SLIB 使用的常量地址修改</li> <li>图 18. icf 文件中 SLIB 地址定义</li> <li>图 19. icf 文件中 UP-Code 使用的 RAM 修改</li> <li>图 20. icf 文件中 IP-Code 使用的 RAM 修改</li> <li>图 21. IAR 代码中 SLIB 使用的常量地址修改</li> <li>图 22. 配置 ICP Programmer</li> <li>图 23. 设置下载选项参数</li> <li>图 24. Project_L0 执行流程</li> <li>图 25. 设置 Keil Misc controls 选项</li> <li>图 26. 修改后的 fir_filter_symbol.txt 内容</li> <li>图 27. 设置 IAR Build Actions 选项</li> <li>图 28. 编辑的 steering_file.txt 内容</li> </ul>	21 21 22 22 22 22 22 22 23 23 24 24 25 26 26 26 27 27
<ul> <li>图 16. KEIL 代码中 SLIB 使用的 RAM 地址修改</li> <li>图 17. KEIL 代码中 SLIB 使用的常量地址修改</li> <li>图 18. icf 文件中 SLIB 地址定义</li> <li>图 19. icf 文件中 SLIB 地址定义</li> <li>图 20. icf 文件中 IP-Code 使用的 RAM 修改</li> <li>图 21. IAR 代码中 SLIB 使用的常量地址修改</li> <li>图 22. 配置 ICP Programmer</li> <li>图 23. 设置下载选项参数</li> <li>图 24. Project_L0 执行流程</li> <li>图 25. 设置 Keil Misc controls 选项</li> <li>图 26. 修改后的 fir_filter_symbol.txt 内容</li> <li>图 27. 设置 IAR Build Actions 选项</li> <li>图 28. 编辑的 steering_file.txt 内容</li> <li>图 29. 修改后的 scatter 文件</li> </ul>	21 21 22 22 22 22 22 23 23 24 24 25 26 26 26 27 27 27 27 28

图 31. 在 Keil 加入 symbol definition file	29
图 32. 修改符号定义文件的类型为 Object file	29
图 33. 在 IAR 中加入符号定义文件	29
图 34. Project_L1 流程图	
图 35. 进入 Show Disassembly at Address	31
图 36. 设置 Show Code at Address	31
图 37. 代码查看	31
图 38. Memory 窗口查看代码	32
图 39. Memory 窗口查看 SLIB_DATA 起始页面	32
图 40. SLIB 写测试	32
图 41. 写保护错误中断	33
图 42. 保存 SLIB 代码	34
图 43. 生成 SLIB 代码部分 bin 文件	35
图 44. ICP 在线烧录 MCU	35
图 45. AT-Link 离线烧录到 MCU	
图 46. 终端用户烧录代码到 MCU	37
图 47. 制作离线项目工程	
图 48. 添加项目文件	39

;;

イ

# 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。

因为这一重要的需求,AT32F403A/407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。

本文档将详细阐述AT32F403A/407系列安全库区的应用原理和软件使用方法。

## 2 应用原理

## 2.1 安全库区的应用原理

- 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\_INSTRUCTION)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\_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\_INSTRUCTION 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,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。

下图是包含安全库区的主闪存区映射示意图,安全库区的程序代码可以很容易地被终端用户调用并执行,但不能直接被读取,因而达到保护的功能。



图 1. 带有安全库区的主闪存区映射

安全库区的范围大小是以扇区(sector)为单位做设定,每一扇区的大小以实际MCU型号为准。表1是AT32F403A/407系列各型号的主闪存大小、每扇区大小及可设置范围。

型号 内部闪存大小 (Byte) 每扇区大小 (Byte) 可设置范围 AT32F403AxC Sector 2 ~ 63 256K 2K AT32F407xC (0x08001000 ~ 0x0801FFFF) AT32F403AxE Sector 2~63 512K 2K AT32F407xE (0x08001000 ~ 0x0801FFFF) AT32F403AxG Sector 2~63 1024K 2K (0x08001000 ~ 0x0801FFFF) AT32F407xG

表 1. AT32F403A/407 各型号闪存大小总表

## 2.2 如何启动安全库区保护

默认状态下,安全库区设定寄存器始终是不可读且被写保护。要想对安全库区设定寄存器进行写操作,首先要对安全库区设定寄存器解锁,对SLIB\_UNLOCK寄存器写入解锁 0xA35F6D24值,通过查看SLIB\_MISC\_STS寄存器的SLIB\_ULKF位确认解锁成功,随 后便允许对安全库区设定寄存器写入设定值。

启动主闪存安全库区的步骤如下:

- 检查FLASH\_STS寄存器的OBF位,以确认没有其他正在进行的闪存操作;
- 对SLIB\_UNLOCK寄存器写入0xA35F6D24,以进行安全库区解锁;
- 检查SLIB\_MISC\_STS寄存器的SLIB\_ULKF位,以确认解锁成功;
- 在SLIB\_SET\_RANGE寄存器设定要保护的区域,包含指令区与数据区的地址;
- 等待OBF位变为'0';
- 在SLIB\_SET\_PWD寄存器设定安全区域密码;
- 等待OBF位变为'0';
- 烧录将存入安全库区的代码;
- 进行系统复位,重装载安全库区设定字;
- 读出SLIB\_STS0/STS1寄存器用于判断安全库区设定结果。

注意事项:

- 只可在主闪存中设置安全库区,实际可设置范围参见表1;
- 安全库区代码必须以扇区为单位进行烧录,且起始地址必须与主闪存地址对齐;
- 中断向量表是数据型态且通常会被放置在闪存的第一扇区(扇区0)内,请勿将闪存的 第一扇区设定为安全库区;
- 要被安全库区保护的程序代码,不可放置在闪存的前4KB内;

关于安全库区设定寄存器的详细说明,请参阅AT32F403A/407系列技术手册。 启动安全库区的程序可参考安全库区应用范例project\_I0中位于main.c中的slib\_enable() 函数。亦可使用雅特力的ICP或ISP刻录工具做设定,后面章节将会有详细的说明。

## 2.3 如何解除安全库区保护

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)。

解除主闪存安全库区的步骤如下:

- 检查FLASH\_STS寄存器的OBF位,以确认没有其他正在进行的编程操作;
- 在SLIB\_PWD\_CLR寄存器写入先前设置的安全区域密码;
- 进行系统复位,重装载安全库区设定字;
- 读出SLIB\_STS0寄存器用于判断安全库区设定结果。

## 2.4 编排及执行安全库区的程序

如前面章节所提到,在指令安全库区(SLIB\_INSTRUCTION)内的的程序代码可以被 MCU经由I-Code总线抓取,但不能经由D-Code总线以读取数据的方式去读出,这样的 保护是全面性的,也就是说在指令安全库区之内的程序代码,也不能读取同样被放置在 指令安全库区之内的数据,例如C程序代码常被编译成的文字池(literal pool)、分支表 (branch table)或常数(constant)等之类当指令被执行时会经由D-Code总线去读取的数 据。

这代表指令安全库区之内只能放置指令,不能放置任何数据。因此用户在编排要放置在 指令安全库区之内的程序代码时,必须配置编译程序(compiler)的设定去产生只执行 (execute-only)的代码以避免上述那些型态的数据产生。

图2及图3是一般常见的文字池跟分支表的例子:

switch()是C程序中常用的跳转指令,此例子中的sclk\_source变量是去读取CRM\_CFG 寄存器,图2可看到编译出来的汇编代码(assembly code) "LDR R7, [PC, #288]",会用 程序计数器(program counter, PC)间接寻址的方式去取得CRM\_CFG寄存器的地址,而 CRM\_CFG的地址会被以常数的方式存放在邻近的指令区(也在指令安全库区之内),因 此执行switch()指令时就会发生数据的读取。如果指令安全库区内有这类的程序代码, 在执行的时候就会产生错误。

第三章的范例程序将会说明如何设定编译程序的配置来避免这样的问题。



图 2. 文字池例子(1)

	137: s	system_core	e_clock = syste	em_core_clock >> div_value;
	0x0800486E	4F06	LDR	r7,[pc,#24] ; @0x08004888
	0x08004870	683F	LDR	r7,[r7,#0x00]
	0x08004872	40F7	LSRS	r7,r7,r6
	0x08004874	F8DFC010	LDR.W	r12,[pc,#16] ; @0x08004888
	0x08004878	F8CC7000	STR	r7,[r12,#0x00]
	138: }			
4	0x0800487C	BDF0	POP	{r4-r7,pc}
	0x0800487E	0000	DCW	0x0000
	0x08004880	1000	DCW	0x1000
	0x08004882	4002	DCW	0x4002

#### 图 3. 文字池例子(2)

## 2.4.1 不可将中断向量表设置为安全库区

中断向量表包含每个中断处理程序的入口点地址,由MCU通过D-Code总线读取。通常,中断向量表位于主闪存第一扇区(sector 0)的起始地址0x08000000,因此在设置指令安全库区时,必须遵守以下的规则:

- 不可将主闪存的第一扇区设置为安全库区
- 要被安全库区保护的程序代码,不可放置在闪存的第一扇区内

## 2.4.2 安全库区代码与用户区代码的关联性

受安全库区保护的程序代码(IP-code)可以从位于用户代码区(安全库区之外的区域)的函数库中调用函数。在这种情形下, IP-Code将会包含这些函数的地址,允许PC(程序计数器)在执行IP-Code时跳转到这些函数。一旦安全库区被启动,这些函数的地址就不能被改变,此时,这些位于用户代码区的函数的地址就必须固定下来,否则PC将跳转到错误的地址而无法正常工作。因此在设置安全库区的时候,应该将所有与IP-Code相关联的函数都一起编排到安全库区之内以避免此情况发生。下图显示出一个被保护的函数Function\_A()调用到用户区内的函数Function\_B()的例子。

**,:17[**7]7[;



图 4. 安全库区的函数调用用户区函数的例子

此外,另一个最常见的情形就是使用到C语言的标准函式库,例如memset()及 memcpy()这类函数。如果IP-Code跟用户区代码都有调用到这类函数,就会有上述问题 的困扰。列举两种常用的解决方法:

- 1) 将其编译到安全库区范围内,具体如何实现可以查看keil或IAR的相关文档。
- 2) 避免在IP-Code内使用C的标准函式库,若非要使用,就必须将用到的函数改写为其他名称,以下是一个范例,在IP-Code 中写一个my\_memset()函数取代原先的memset()。

# 图 5. 自定义函数范例 void\* my\_memset(void \*s, int c, size\_t n); void arm\_fir\_init\_f32( arm\_fir\_instance\_f32 \* S, uint16\_t numTaps, float32\_t \* pCoeffs, float32\_t \* pState, uint32\_t blockSize) ₽{ /\* Assign filter taps \*/ S->numTaps = numTaps; /\* Assign coefficient pointer \*/ S->pCoeffs = pCoeffs; /\* Clear state buffer and the size of state buffer is (blockSize + numTaps \*/ my\_memset(pState, 0, (numTaps + (blockSize - 1u)) \* sizeof(float32\_t)); /\* Assign state pointer \*/ S->pState = pState; void\* my\_memset(void \*s, int c, size\_t n) 曱 { while (n>0) \*( (char\*)s + n-- -1 ) = (char)c; return (s);

# 3 安全库区范例程序

本章节将介绍安全库区的使用范例,并详述完成此范例程序所需的每一个步骤。因为AT32F407的SLIB功能与AT32F403A完全相同,此处就以AT32F403A为范例做说明。

## 3.1 范例需求

## 3.1.1 硬件需求

- 带有AT32F403AVGT7芯片的AT-START-F403A实验板
- AT-Link仿真器,用来调试范例程序

## 3.1.2 软件需求

- Keil® µvision IDE (本范例使用µvision V5.18.0.0) 或 IAR Embedded workbench IDE(本范例使用IAR V8.22.2)
- 雅特力ICP或ISP刻录工具,主要是用来启动或解除安全库区的设置

## 3.2 范例概述

本应用指南提供了两个范例项目,展示了软件开发商开发智权代码(IP-code)给终端用户应用的场景。其中

- Project\_L0为方案商开发算法并编排到安全库区的示例
- Project\_L1为终端用户应用此算法的示例

Project\_L0完成的算法将预先下载刻录到AT32F403A芯片并设置安全库区保护,同时提供下列各项 设定讯息给终端客户应用程序使用:

- 主闪存区块的映像,说明安全库区所占用的区域以及用户可开发程序的区域
- 包含算法函数定义的头文件,让终端用户可以用来调用相关的函数
- 符号定义文件(symbol definition file),此符号文件内含IP-Code的各个函数的实际地址,让终端 用户程序可以正确的调用,下图为此范例的示意图。



### 图 6. 范例流程示意图

软件方案商可以参考Project\_L0范例开发算法代码,并参考Project\_L1提供终端用户使用,下图为应

用示意图。



#### 安全库区保护的代码: FIR 低通滤波器 3.3

本范例使用CMSIS-DSP库提供的FIR低通滤波器(FIR lowpass filter)算法作为被安全库保护的IP-Code,关于FIR低通滤波器算法可详阅CMSIS-DSP的相关文件,这里仅着重在说明如何设置安全库 以保护此算法及如何被终端用户的程序代码调用。

范例中的低通滤波器的输入信号是一个混和了频率各为1KHz及15KHz的两个正弦波的讯号,而低通 滤波器的截止频率约为6KHz。经过低通滤波后,将15KHz的讯号滤除而仅剩下1KHz的正弦波输出。 下图为FIR低通滤波功能的示意图。



图 8. FIR 低通滤波器

使用到的CMSIS DSP库的函数及文件包括:

arm\_fir\_init\_f32() •

此函数的功能是做滤波器函数的初始化设定,包含在arm fir init f32.c文件里

arm\_fir\_f32() •

此函数为滤波器算法的主要部分,包含在arm\_fir\_f32.c文件里

## FIR lowpass filter()

此函数为使用上述两个基本函数写成的FIR低通滤波器全局函数,供终端用户调用,包含在fir filter.c 文件里

### fir coefficient.c

此C文件内含FIR滤波器函数所使用的系数(只读的常数),在范例中会将这些系数放置到数据安全库区

在此范例中,MCU内嵌的FPU及DSP指令会被用来做信号处理以及浮点运算,以达到准确的运算及 正确的输出信号。

## 3.4 Project\_L0: 方案商范例

在此阶段的范例程序,将完成下列几个项目:

- 将算法的相关函数编译成只可执行(execute-only)的代码
- 将算法的程序代码编排放置到主闪存区的扇区2(sector 2)
- 将滤波器函数的系数编排放置到主闪存区的扇区4(sector 4)
- 在主程序中执行FIR\_lowpass\_filter()以验证其正确性
- 验证成功后,将扇区2及扇区3(sector 2/3)设置为指令安全库区,并将扇区4及扇区5(sector 4/5) 设置为数据安全库区,此部分可在范例的主程序中以调用slib\_enable()函数来完成,或使用
   Artery ICP Programmer来完成(建议使用ICP工具完成设置)
- 产出终端用户程序调用低通滤波函数时需用到的头文件及符号定义文件

## 3.4.1 产生只执行(Execute-only)代码

每一种工具链(toolchain)都有自己的设定选项,可以防止编译程序生成文字池(literal pools)和分支表 (branch table)这些在指令执行时会发生读取数据的指令格式,例如"LDR Rn, [PC, #offset]"这类指 令。关于文字池及分支表的例子可参照章节2.4的说明。

以Keil® µvision为例,Keil® µvision有Execute-only Code的选项来做设定,设定的方式如下:

## Keil® µvision: 使用Execute-only Code选项

设置的方式是:

- 选择C文件群组或个别的C文件,范例中是把要保护的相关C文件都放在fir\_filter群组
- 按鼠标右键然后选择对应文件,例如本例程的Option for File 'arm\_fir\_f32.c',如下图



图 9. Keil 进入 Option 界面

● 勾选 C/C++ 窗口里的Execute-only Code选项, 然后--execute\_only命令就会被加到编译过程控制字符串里, 如下图

Properties C/C++		
Preprocessor Symbols		
Define:		
Undefine:		
Language / Code Generation	_	Waminos:
Execute-only Code	I⊻ Strict <u>A</u> NSI C	
Optimization: <default></default>	I I Enum <u>C</u> ontainer always int	
Optimize for Time	Plain Char is Signed	Thum <u>b</u> Mode
Split Load and Store Multiple	Read-Only Position Independent	Mo Auto Includes
✓ One ELF Section per Function	Read-Write Position Independent	C99 Mode
Include Paths		
Misc Controls		
Compiler control string	M4.fp -DMICROLIB -g -O0apcs=interwo Vibraries.cmsis.cm4.core_support -1	rksplit_sections -I\\ A \Vibraries\cmsis
OK	Cancel Defaults	Help

### 图 10. Keil 选择 Execute-only Code

● 本例程中有三个文件位于SLIB\_INSTRUCTION区,分别是arm\_fir\_f32.c、arm\_fir\_init\_f32.c和 fir\_filter.c,这三个文件都需要配置产生为只执行代码。

### IAR : 使用No data read in code memory 选项

设置的方式是:

● 选择fir\_filter群组里对应的文件,按鼠标右键选择Option

图 11. IAR 进入	Option 界面
--------------	-----------

├	
	Options
-⊞ lo fir_coefficient.c	Maka
🖵 🖽 💿 fir_filter.c	

● 如下图,在"C/C++"窗口内勾选Override inherited settings以及No data read in code memory



Options for node "arm_fir_	f32.c*
Exclude from build	
Category:	Override inherited settings     Factory Settings
Static Analysis Runtime Checking	
Custom Build	Preprocessor Diagnostics MISRA-C:2004
	MISRA-C:1998 Encodings Extra Options Language 1 Language 2 Code Optimizations Output List Processor mode Arm Thumb Position-independence Code and read-only data (ropi) Read/write data (rwpi) No dynamic read/write initializati. Mo data reads in code memory
	OK Cancel

图 12. IAR 设置 C/C++窗口选项

● 本例程中有三个文件位于SLIB\_INSTRUCTION区,分别是arm\_fir\_f32.c、arm\_fir\_init\_f32.c和 fir\_filter.c,这三个文件都需要配置产生为只执行代码。

## 3.4.2 编排安全库区的地址

如前面章节提到的,因为主闪存的第一扇区(sector 0)会被用来存放中断向量表,所以本范例选择从 sector 2开始设定为安全库区,其中sector 2及sector 3是指令安全库区,而sector 4跟sector 5则被设 置为数据安全库区。下图为主闪存的映射及RAM的使用分区。RAM的分区主要是为了避免SLIB保护 区的代码与终端用户的代码用到相同的RAM而产生的冲突问题。



图 13. 范例程序的主闪存映像及 RAM 分区

## Keil® µvision的scatter file

步骤如下:

● 到 Project → Optios for Target→Linker 窗口,取消Use memory layout from Target Dialog选 项,然后按Edit按键来开启slib-w-xo.sct文件做修改,如下图

💟 Optic	ns for Target 'at_start_f403a'	
Device	Target       Output       Listing       Vser       C/C++       Asm       Linker       Debug       Utilities         e Memory Layout from Target Dialog       X/O Base:	
S	Atter Velib-w-xo.sct Edit	
co L	Misc -symdefs=fir_filter_symbol.txt	
	TIME DEFAULTS Help	

图 14. Keil 设置 Linker 窗口选项

打开scatter file之后,将需要放到指令安全库区(SLIB\_INSTRUCTION)的代码的目标文件(object file)放到名为LR\_SLIB\_INSTRUCTION的专用加载区,并将标示修改为execute-only (+XO),加载区的起始位置为sector 2,大小占用两个sector,同时也要将SLIB\_Data占用的区域放到名为LR\_SLIB\_DATA的专用加载区,避免编译程序将其他非IP-Code的函数编排到SLIB区内,RW\_IRAM2是将0x20017000到0x20017FFF的区块,指定给算法的函数使用,目的是为了避免终端用户的项目也用到同样的RAM区块,而造成程序执行时发生错误,如下所示

图 15. Keil scatter 修改



 IP-Code 用到的RAM,除了上述的修改scatter file方式之外,也可以使用Keil的 \_\_attribute\_\_((at(address)))描述元将变数放置到固定的地址0x20017000,如下图



```
61 #if defined (__ICCARM__)

62 static float32_t firStateF32[BLOCK_SIZE + NUM_TAPS - 1] @ 0x20017000 ;

63 #elif defined (__CC_ARM )

64 static float32_t firStateF32[BLOCK_SIZE + NUM_TAPS - 1] __attribute__((at(0x20017000)));

65 #endif
```

 数据安全库区的起始位置为sector 4 (0x08002000),要将FIR低通滤波器函数使用到的常数编排 到此地址,除了上述的修改scatter file方式之外,也可以使用Keil的\_\_attribute\_\_((at(address))) 描述元将常数放置到固定的地址,如下图

图 17. KEIL 代码中 SLIB 使用的常量地址修改

### IAR的ICF file

步骤如下:

● 开启\project\_I0\IAR\_V8.2\目录下的icf文件,添加三个新的SLIB加载区,如下所示,其中

**,:17[**27];

AT32F403A/407 Security Library Application Note

SLIB\_RAM区块将0x20017000到0x20017FFF的RAM保留给算法的函数使用

#### 图 18. icf 文件中 SLIB 地址定义

```
/* SLIB INSTRUCTION area */
define symbol __ICFEDIT_region_SLIB_INSTRUCTION_start__ = 0x08001000;
define symbol __ICFEDIT_region_SLIB_INSTRUCTION_end__ = 0x08001FFF;
/* SLIB DATA area */
define symbol __ICFEDIT_region_SLIB_DATA_start__ = 0x08002000;
define symbol __ICFEDIT_region_RAM_start__ = 0x20000000;
define symbol __ICFEDIT_region_RAM_end__ = 0x20000000;
define symbol __ICFEDIT_region_RAM_end__ = 0x20017FFF;
/* SLIB RAM region */
define symbol __ICFEDIT_region_SLIB_RAM_start__ = 0x20017000;
define symbol __ICFEDIT_region_SLIB_RAM_end__ = 0x20017FFF;
```

 在icf文件中,也要将SLIB占用的区域保留起来,避免编译程序将其他非IP-Code的函数编排到 SLIB区内,同时将IP-Code使用的RAM区域保留起来

#### 图 19. icf 文件中地址分配

/* Reserved 0x08001000 ~ 0x08002FFF as SLIB area */ define region ROM_region = mem:[fromICFEDIT_region_ROM_start toICFEDIT_region_ROM_end_]
define region SLIB_INSTRUCTION_region = mem:[fromICFEDIT_region_SLIB_INSTRUCTION_start_ toICFEDIT_region_SLIB_INSTRUCTION_end_];
define region SLIB_DATA_region = mem:[fromICFEDIT_region_SLIB_DATA_start toICFEDIT_region_SLIB_DATA_end];
/* Reserved 0x20017000 ~ 0x20017FFF as RAM used for SLIB code */ define region RAM_region = mem:[fromICFEDIT_region_RAM_start toICFEDIT_region_RAM_end] - mem:[fromICFEDIT_region_SLIB_RAM_start toICFEDIT_region_SLIB_RAM_end];
define region SLIB_RAM_region = mem:[fromICFEDIT_region_SLIB_RAM_start toICFEDIT_region_SLIB_RAM_end];

● IP-Code 用到的RAM,可以使用IAR的@描述元将变数放置到固定的地址0x20017000,或者修改icf文件,如下图

#### 图 20. icf 文件中 IP-Code 使用的 RAM 修改

 数据安全库区的起始位置为sector 4 (0x08002000),要将FIR低通滤波器函数使用到的常数编排 到此地址,除了上述的ICF文件方式之外,也可以使用IAR的@描述元将常数放置到固定的地址,如下图

#### 图 21. IAR 代码中 SLIB 使用的常量地址修改

```
#if defined ( __ICCARM__ )
    const float32_t firCoeffs32[NUM_TAPS] @ 0x08002000 ={
    #elif defined ( __CC_ARM )
    const float32_t firCoeffs32[NUM_TAPS] __attribute__((at(0x08002000))) = {
    #endif
    -0.0018225230f, -0.0015879294f, +0.0000000000f, +0.0036977508f, +0.0080754303f,
    -0.0341458607f, -0.0333591565f, +0.0000000000f, +0.0676308395f, +0.1522061835f,
    +0.1522061835f, +0.0676308395f, +0.000000000f, -0.0333591565f, -0.0341458607f,
    +0.0080754303f, +0.0036977508f, +0.000000000f, -0.0015879294f, -0.0018225230f
    -};
```

# 

## 3.4.3 启用安全库区保护

要启用安全库区的保护功能,有以下两种方式:

(1) 使用ICP刻录工具 Artery ICP Programmer(建议用此方式)

要使用ICP Programmer,请参照以下步骤:

- 连接AT-Link仿真器到AT-START-F403A板子上并上电
- 开启ICP Programmer,选择用AT-Link连接,然后添加Project\_L0范例编译后产出的HEX或BIN 文件,如下图

	Davt N	lu una la		AT2	2540	2	GT7		lach	line	1024	VP		1			
Disconnect		1umb	er:	E14/-	2140	10	317	-	lasna	oize:	1024	ND	- 1	;= I	۲1	Ξ.	۲ ۲
AT-Link 🔻	AT-Lir	ik-EZ ik SN:	: 80	21090	C110	08045	54105	17BD	002					雅	1	诗	ナ
		M		FL	ASH_	DA 0	ĸ [	0		Re	emap	0 (Us	e PA1	L1/PA	- 12 pi	ns)	
	Туре						] [ s	elect		🔿 Re	emap	1 (Us	e PB1	LO/PB	11 pi	ns)	
Memory read	l setting	gs															
Address 0x	080000	00	Re	ead s	ize	0x 00	00003	CC	D	)ata b	its	8 bits	-	]		Read	1
											6			,			
-File info																	
																1	
No. File r	ame				File	size	Ad	Idres	s ran	ge(0x	)					A	.dd
No. File r 1 at32	ame 403a_pr	oject_	l0.he	ĸ	File 923	size 2	Ad 08	ldres: 0000	s rang 00-08	ge(0x) 30003	) CB,08	0010	00-08	30011	.17,08	A	dd lete
No. File r 1 at32	ame 403a_pr	oject_	l0.hex	K	File 923	size 2	Ad 08	ldres: 0000	s ran( 00-08	ge(0x) 30003	) CB,08	0010	00-08	30011	.17,08	A De	dd lete
No. File r 1 at32	ame 403a_pr	oject_	<u>10.he</u> y	×	File 923	size 2	Ad 08	ldres: 0000	s ran( 00-08	ge(0x) 30003	) CB,08	0010	00-08	30011	17,08	A De	dd lete
No. File r 1 at32	ame 403a_pr	oject_	l0.hex	×	File 923	size 2	Ad 08	ldres:	s rang 00-08	ge(0x)	) CB,08	0010	00-08	30011	.17,08 ►	A De	dd lete
No. File r 1 at32	ame 403a_pr	oject_	10.hex	x 111	File 923	size 2	Ad	Idres: 0000 Flas	s ran( 00-08	ge(0x 30003	) CB,08	0010	00-08	80011 Fy	.17,08 ► Do	De	dd Hete
No. File r	iame 403a_pr	oject_	l0.hex	×	File 923	size 2	Ad 08	Idress 00000 Flas	s rang 00-08 h CR0	ge(0x) 80003	) CB,08 File	0010 • CRC	00-08	30011 fy	17,08	A De	dd Hete
No. File r 1 at32 < Flash info Fi	ame 403a_pr e:at32f4	oject_ 03a_p	l0.hex	x 	File 923 ex	size 2		Idres: 00000 Flas	s rang 00-08 h CR0	ge(0x) 30003	) CB,08 File	0010 • CRC	00-08 Verif	50011	17,08		dd Hete
No. File r 1 at32 Flash info Fi Address range 0x080020731	ame 403a_pr e:at32f4 :[0x08000 Address	oject_ 03a_p )000 0> range:	10.hex roject	x 111 t_l0.h 03CB] 00300	File 923	size 2 dress r 8004El	Ad 08	Idress 00000 Flas [0x08( checks	s ran( 00-08 h CR( 00100 sum: (	ge(0x) 30003	) CB,08 File 00111 06B92	0010 • CRC 7] A	00-08 verif	50011	.17,08	A De wnLc	dd Hete
No. File r 1 at32 <ul> <li>✓</li> </ul> <li>Flash info Fi Address range 0x080020731 Address</li>	e:at32f4 :[0x08000 Address 0	O3a_p 0000 0x range:	10.hex roject x08000 10x080 2	x t_l0.h 03CBJ 00300 3	File 923 ex Adi 0 0x0	size 2 dress r 8004EI 5	Ad 08	Flas	s rang 00-08 h CR( 00100 sum: (	ge(0x 30003 C 0 0x08 0x000 9	) CB,08 File 001111 26B92 A	0010 CRC 7] A	00-08 verif ddres	5 rang	.17,08 ► Do ge:[0xt	M De wnLc	oad
No.     File r       1     at32	e:at32f4 (0x08000 Address 0 40	oject_ 03a_p 0000 0> range: 1 1B	10.hex roject x08000 (0x080 2 00	t_l0.h 00300 3 20	File 923 ex Adr 0 0x0 4 01	size 2 dress r 8004El 5 30	Ad 08 108	Idres: 00000 Flas (0x084 check: 7 08	s rang 00-08 h CR( 00100 sum: ( 8 1B	ge(0x) 30003 C 0 0x08 0x000 9 36	) CB,08 File 001111 06B92 A 00	0010 • CRC 7] A B 08	00-08 ¢ verif ddres C 13	30011 fy ( s rang 36	► Do	A De 080020 F 08	dd Hete Dad
No.     File r       1     at32       4     at32       4     at32       Address range     at40000000       0x08000000     at4000000       0x08000000     at4000000	e:at32f4 :[0x08000 Address 0 40 17	oject 03a_p 0000 0x range: 1 1B 36	10.hes project x08000 (10x086 2 00 00	x t_l0.h 03CB] 00300 3 20 08	File 923 ex Add 0 0x0 4 01 F3	size 2 dress r 8004E1 5 30 35	Ad 08 range: 871 ( 6 00 00	00000 Flas (0x084 (0x084 7 08 08	s ran 00-08 h CR0 001000 sum: ( 8 1B 99	ge(0x) 00003 00003 00008 00000 9 36 36	) CB,09 File 001111 06B92 A 00 00	0010 • CRC 7] A B 08 08	00-08 c verif ddres C 13 00	80011 fy ( s rang 36 00	► <b>Do</b> (0xt) E 00 00	A De 080020 F 08 00	dd elete Dad 000 At ^ @r

图 22. 配置 ICP Programmer

● 按下载按键,会出现下载选项的页面,此页面会显示SLIB的状态及相关的参数,设定sector 2为 开始扇区、sector 4为数据开始扇区以及sector 5为结束扇区,设定启用密码0x55665566 (可自 定义)并勾选启用SLIB,然后按开始下载,即可完成程序的烧录并启用SLIB,如下图

sLib status: Disable	Remaining usage t	times: 246	
Enable password 0x 55665566	sLib position:	Main Flash 👻	
Disable password 0x	Start sector	Sector 20x08001000 -	
Disable sLib	DATA start sector	Sector 40x08002000 -	
	End sector	Sector 50x08002800 -	
Extra options			
Erase the sectors of file size	▼ Dis	able sLib before download	
Verify Custom encryption key	for verify: 👿 Ena	able sLib	
	🔲 Dis	able FAP before download	
Jump to the user program	Ena	able FAP after download	
Write software serial number(SN)			
Write address Ox 00010000	🔲 But	ton free mode	
Increase step 0x 0000001			
🔲 Write user system data			
User system data file path			

图 23. 设置下载选项参数

关于ICP Programmer的详细说明,请参阅ICP Programmer用户手册。

## (2) 使用范例程序main.c之中的slib\_enable()函数

在低通滤波函数测试正确后执行过一次此函数,就可以启用安全库区的保护功能。要执行此函数,只要在main.c中使能 #define USE\_SLIB\_FUNCTION 即可。

# 3.4.4 Project\_L0 执行流程

在此范例中,FIR低通滤波器会针对混和1KHz及15KHz正弦波的输入信号 testInput\_f32\_1kHz\_15kHz 做计算,计算后输出的1KHz正弦波数据存放到testOutput,然后会跟预 先用MATLAB软件计算好且存放在refOutput中的数据做比对,如果误差值小于预期值(讯噪比SNR大 于预设的门坎),板子上绿色的LED灯会一值闪烁,反之则是红色的LED灯一值闪烁,下图是 Project\_L0的整个流程



图 24. Project\_L0 执行流程

要执行此范例程序,请按照下列步骤:

- (1) 使用Keil® µvision开启\utilities\at32f403a\_407\_slib\_demo\project\_l0\mdk\_v5\目录下的 Project L0项目,并重新编译。
- (2) 在下载代码之前,先检查AT-START-F403A板子上的芯片是否已经有SLIB或读写保护 (FAP/EPP),如果有,就请先用ICP刻录工具将这些保护都解除,然后再下载代码。
- (3) 下载成功后并开始值执行后,会看到板子上的LED3灯持续快速闪烁。
- (4) 按下板子上的USER按键,就会执行低通滤波器的运算。
- (5) 比对运算结果,若结果正确,绿色LED4灯会持续闪烁。反之,则是红色LED2灯持续闪 烁。
- (6) 在比对结果正确的条件下,如果main.c中的USE\_SLIB\_FUNCTION有被定义且芯片未启用过

SLIB的话,就会执行slib\_enable()函数去设置SLIB,若设置失败,红色LED2灯会一直亮着。若 设置成功,绿色LED4灯会点亮约3秒钟然后执行系统重置(system reset)来启动SLIB。然后程序 又回到步骤(3)。

## 3.4.5 产生头文件及符号定义文件

头文件(header file)跟符号定义文件(symbol definition file)是终端客户应用范例Project\_L1在调用FIR 低通滤波函数时需要用到。在范例中,就是main.c中包含的fir\_filter.h 文件。 符号定义文件的产出方法跟使用的工具链(toolchain)相关。

## 使用Keil® µvision 产生符号定义文件

方法如下:

- 进入 Ottions for Target → Linker 设定画面
- 在 Misc controls 这一栏,添加--symdefs=fir\_filter\_symbol.txt 命令,如下图

图	25.	设置	Keil	Misc	controls	选项
---	-----	----	------	------	----------	----

Options for Target 'at_start_f403a'	×
Device   Target   Output   Listing   User   C/C+	+ Asm Linker Debug   Utilities
<ul> <li>□ Use Memory Layout from Target Dialog</li> <li>□ Make RW Sections Position Independent</li> <li>□ Make RO Sections Position Independent</li> <li>□ Don't Search Standard Libraries</li> <li>□ Report 'might fail' Conditions as Errors</li> </ul>	X/O Base:
Scatter File	Edit
Miscsymdefs=fir_filter_symbol.bxt controls Linker -cpu Cottex-M4 fp * o -tibrary_type=microlibstrictscatter ''.\sl	lib-w:xo.sct"
OK	Cancel Defaults Help

- 重新编译整个项目后,在project\_I0\mdk\_v5\Objects 目录下就会产生一个名为 fir\_filter\_symbol.txt的符号定义文件
- 这个符号定义文件包含了整个项目全部的符号定义,所以需要修改,只保留终端用户会调用的低 通滤波函数的定义,删减后的fir\_filter\_symbol.txt显示如下

### 图 26. 修改后的 fir\_filter\_symbol.txt 内容

## 0x08001001 T FIR\_lowpass\_filter

### 使用IAR产生符号定义文件

方法如下:

● 选择 Project→Option→Build Actions

Options for node "project_10" Category: General Options Static Analysis Runtime Checking C/C++ Compiler Assembler Output Converter Custom Build Build Actions Linker Debugger Simulator CADI CMSIS DAP GDB Server I-jet/JTAGjet J-Link/J-Trace TI Stellaris Nu-Link PE micro ST-LINK Third-Party Driver TI XDS	Build Actions Configuration Pre-build command line:  Prost build command line:  \$TOOLKIT_DIR\$\bin\isymexport.exeedit "\$PROJ_DIR\$\st.
---	--

图 27. 设置 IAR Build Actions 选项

- 然后在 Post-build 命令行中输入以下命令
   \$TOOLKIT\_DIR\$\bin\isymexport.exe --edit "\$PROJ\_DIR\$\steering\_file.txt"
   "\$TARGET\_PATH\$" "\$PROJ\_DIR\$\fir\_filter\_symbol.o"
- 此处 fir\_filter\_symbol.o 是要产出的符号定义文件, steering\_file.txt 放在project\_l0\iar\_v8.2目 录下,是用来选择要产生哪些函数的符号,需根据安全库区调用的内容进行手动编辑,内容如 下,其中"show"是用来选择函数的命令

图 28.	编辑的	steering_	_file.txt	内容
-------	-----	-----------	-----------	----

show FIR lowpass filter

## 3.5 Project\_L1:终端用户范例

Project\_L1范例会使用到在Project\_L0中调试好,并已经被刻录到AT32F403A芯片的主闪存中且被 SLIB保护的FIR低通滤波器函数。根据 Project\_L0提供的头文件、符号定义文件以及主闪存区块映 像,终端用户就可以参照Project\_L1做到

- 建立一个应用项目
- 引用Project\_L0提供的头文件及符号定义文件到项目里
- 调用FIR低通滤波器函数
- 开发并调试用户自己的应用程序

### 注意事项:

Project\_L1必须使用跟Project\_L0开发时一样的工具链及相同版本的编译程序,不然有可能会因为版本差异的兼容性问题,而无法使用Project\_L0提供的代码。例如本范例中 Project\_L0使用的是Keil® µvision V5.18.0.0 ,那Project\_L1也要使用同样的这个版本。

## 3.5.1 建立用户的应用项目

因为Project\_LO启用的安全库区已经占用了一些特定的主闪存扇区,Project\_L1的代码必须参照 Project\_L0提供的主闪存区块映像来编排放置的地址。图13为此范例的主闪存区块映射,其中sector 2至sector 5为安全库区所占用,终端用户需使用linker control file将这个区域隔离起来,避免代码在 编译时被编排到这个区域内,方式如下:

## Keil® µ vision 的scatter file

可参照 project\_l1\mdk\_v5\ 目录的end\_user\_code.sct文件,将主闪存空间切成两个区块,中间空出 来的区域就是SLIB保护区。此外,RAM的区域也要保留0x20017000之后的区域。如下图

图 29. 修改后的 scatter 文件

```
LR_IROM1 0x08000000 0x00001000 { : load region size_region
ER_IROM1 0x08000000 0x00001000 { : load address = execution address
*.o (RESET, +First)
*(InRoot$Sections)
.ANY (+RO)
}
RW_IRAM1 0x20000000 0x00017000 { : RW data
.ANY (+RW +ZI)
}
: 0x20017000 ~ 0x20017FFF RAM reserved for SLIB code
}
; 0x08001000 ~ 0x08002FFF is SLIB area
LR_IROM2 0x08003000 0x000FD000 { : load region size_region
ER_IROM2 0x08003000 0x000FD000 { : load region size_region
ER_IROM2 0x08003000 0x000FD000 { : load address = execution address
.ANY (+RO)
}
```

## IAR 的ICF file

可参照 project\_l1\iar\_V8.2\ 目录下enduser.icf文件中如下图的部分

图 30. 修改后的 icf 文件

# 3.5.2 在项目中加入符号定义文件

Project\_L0所产生的符号定义文件fir\_filter\_symbol.txt必须被添加到Project\_L1项目中,才能被正确的 编译并链结到SLIB保护区的代码。

### 在Keil® µ vision中加入符号定义文件

工程中添加fir\_filter\_symbol.txt这个符号定义文件,如下图:



图 31. 在 Keil 加入 symbol definition file

将文件加入fir\_filter群组后,必须将它的文件类型更改为Object文件,而不是原来的文本(text)文件, 修改方式如下

Options for File 'fir_filter_symbol.txt'	×
Properties	1
Path Min filter symbolis	✓ Include in Target Build
Size: 217 bytes	Generate Assembler SRC File
last change.   II May 21 11.14.10 2021	Assemble SRC File
Stop on Exit Code: Not specified	🔽 Image File Compression
Custom Arguments:	
Memory Assignment: Code / Const: Zero Initialized Data: Other Data: Cefault> V	
OK Cancel De	faults Help

图 32. 修改符号定义文件的类型为 Object file

## 在IAR中加入符号定义文件

将fir\_filter\_symbol.o 这个Object文件加到加到fir\_filter群组即可,如下图



图 33. 在 IAR 中加入符号定义文件

## 3.5.3 调用 SLIB 保护区的函数

当filter.h头文件被main.c引用且符号定义文件也正确地加入项目之后,保护区的低通波器函数就可以

被调用。方式如下:

FIR\_lowpass\_filter(inputF32, outputF32, TEST\_LENGTH\_SAMPLES); 其中:

- inputF32:指向包含输入信号数据表的指针
- outputF32: 指向存放输出信号数据表的指针
- TEST\_LENGTH\_SAMPLES: 要被处理的信号样本数

## 3.5.4 Project\_L1 执行流程

Project\_L1的执行流程如下图,说明如下:

- 开始执行后LED3灯会持续闪烁
- 按下AT-START板子上的USER 按键, FIR\_lowpass\_filter()开始做运算
- 如运算结果正确,绿色LED4灯持续闪烁,如运算结果错误则红色LED2灯持续闪烁





## 3.5.5 调试模式下的 SLIB 保护

当终端用户在开发应用程序时,会用到开发工具调试代码,以下将以Keil® μvision为例, 说明在调 适模式下,SLIB如何防止保护区内的代码被以数据的方式读取

- 开启Project\_L1项目并重新编译
- 点击"Start/Stop Debug Session"进入调试模式



● 在"Disassembly"窗口点击数标右键,然后选择"Show Disassembly at Address",如下图

图 35. 进入 Show Disassembly at Address

Disassembly					ņ	L 🗙
. 0x08003E52 4770 BX 94: AT32_Board_Init(); 95:	lr					^
96: /* Configure Flash to gen	erate	$\checkmark$	Mixed Mode		error occur */	
©0x08003E54 2804 CMP	r0,#0		Assembly Mode			
97: Enable_Flash_INT();	UX080		Address Range	•		
99:			Show Disassembly at Address			
100: /* Wait for KEY button t	o be p		Set Program Counter			
0x08003E58 490A LDR	r1,[p	*()	Run to Cursor line	Ctrl+F10		
101: while(AT32_BUTTON_State( 102: (	rı,[r BUTTON		Insert/Remove Breakpoint			
0x08003E5C F0510104 ORRS	r1,r1	0	Enable/Disable Breakpoint	Ctrl+F9		
0x08003E60 4A08 LDR	r2,[p		Incert Tracepoint at '0x00002EE4'			
Ux08003E62 6011 STR	r1,[r		insert nacepoint at 0x00005E54			
104: Delay_ms(300);			Enable/Disable Tracepoint			
106:			Inline Assembly			
107: /* Turn Off LED3 */			Load Hex or Object file			
0x08003E64 E005 B	0x080		In the other Trans			
0w00003760 6000 ThD	~ 1 I P		instruction irace			~
			Execution Profiling	•	, ,	>
i) main.c		P	Insert/Remove Bookmark	Ctrl+F2	-	×
	Call Stack					

● 输入SLIB\_INSTRUCTION起始扇区sector 2的地址0x08001000

#### 图 36. 设置 Show Code at Address

Show Code at Address	×
Address: 0x08001000	Go To

● 可以看到地址0x08001000这里开始,看到的代码都是0xFFFFFFFF

#### 图 37. 代码查看

Disassembly	
Disassenilary	
/b: {	
77: UINCJZ t 1;	0
OXUBUCIOUU FFFFFFF DCD	
OXOCOLOGY FFFFFFF DCD	
79: uint32 t num	Blocks = testlengthcomples/BLOCK SIZE:
80: arm fir instance f	32 G.
81:	
82:	
83: /* Call FIR init fu	unction to initialize the instance structure. */
0x0800100C FFFFFFFF DCD	OXFFFFFFFF
84: arm fir init f32(&	5, NUM TAPS, (float32 t *)&firCoeffs32[0], &firStateF32[0], blockSi
85: /*	
86: ** Call the FIR pro	ocess function for every blockSize samples
87: **	*/
Ox08001010 FFFFFFFF DCD	OxFFFFFFF
Ox08001014 FFFFFFFF DCD	OxFFFFFFFF
OxO8001018 FFFFFFFF DCD	OxFFFFFFF
0x0800101C FFFFFFFF DCD	OXFFFFFFFF
0x08001020 FFFFFFFF DCD	OXFFFFFFF
0x08001024 FFFFFFFF DCD	OXFFFFFFF
0x08001028 FFFFFFFF DCD	OXFFFFFFF
UXU8UU1U2C FFFFFFF DCD	UXFFFFFFF
OXU8001030 FFFFFFF DCD	
UXU8UUIU34 FFFFFFFF DCD	OXFFFFFFF

● 同样地在Memory窗口输入0x08001000的地址,也会看到全部是 0xFF



### 图 38. Memory 窗口查看代码

																				-
Address: 0x0800	1000	1																	0	^
0x08001000:	FF	FF	FF	FF	$\mathbf{F}\mathbf{F}$	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
0x08001013:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
0x08001026:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
0x08001039:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
0x0800104C:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
0x0800105F:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
0x08001072:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
0x08001085:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
0x08001098:	FF	न न	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	~
Call Stack + Lo	ocals		Mer	nory	1	ш м	lemo	ry 2												
•		-		-	_															

● 在Memory窗口,输入SLIB\_DATA起始扇区sector 4的地址0x08002000,因为这个区块允许被 D-Code数据总线读取,所以可以看到原来的数值

图	39.	Memory	窗口查看 SLIB	DATA 起始页面
---	-----	--------	-----------	-----------

A 11 0 000																				^
Address: JUXU8UU	J2001	J																		
0x08002000:	В9	E 1	ΕE	ΒA	12	22	DO	BA	00	00	00	00	F7	55	72	ЗB	CF	4E	04	
0x08002013:	ЗC	58	C2	OB	ЗC	00	00	00	80	9E	85	8E	ВC	88	$\mathbf{DC}$	OB	$\mathtt{BD}$	9C	A3	
0x08002026:	08	ВD	00	00	00	00	OA	82	88	ЗD	FO	DB	1B	ЗE	5F	46	64	ЗE	06	
0x08002039:	41	80	ЗE	5F	46	64	3 E	FO	DB	1B	3 E	OA	82	88	ЗD	00	00	00	00	
0x0800204C:	9C	АЗ	08	ВD	88	DC	OB	$^{\rm BD}$	9E	85	8E	ВC	00	00	00	80	58	C2	OB	
0x0800205F:	ЗC	CF	4E	04	ЗC	F7	55	72	ЗB	00	00	00	00	12	22	DO	ВA	В9	Ε1	
0x08002072:	ΕE	BÅ	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
0x08002085:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
		FF	ਜ਼ਾਜ਼	ਜ਼ਾਜ਼	ਜ਼ਾਜ਼	नन	न न	न न	न न	FF	FF	FF	FF	FF	FF	नन	नन	नन	नन	~

● 在Memory窗口用鼠标双击0x08002000的数据尝试做修改,FLASH\_STS寄存器的EPPERR位置"1"提出警告,显示写保护发生作用

Щ.010-11- <b>(</b>	VWF1	
Property	Value	
PSR	0x0000030	
	0	
USD_UNLOCK	0	
STS	0x0000010	
ODF		
EPPERR	<b>V</b>	
PRGMERR		
 OBF		

图 40. SLIB 写测试

• 如果有使能擦写保护错误中断,继续执行程序就会进入中断程序里面



图 41. 写保护错误中断
115 void FLASH_IRQHandler(void) 116 ₽ {
<pre>117   if(flash_flag_get(FLASH_EPPERR_FLAG)) 118 □ { 119    flash_flag_clear(FLASH_EPPERR_FLAG); 120    delay_ms(500); 121  } 122 }</pre>

# 4 方案商和终端用户代码整合及下载操作流程

方案商和终端用户的代码设计完成后,需要下载到同一个MCU中,这就涉及到各自代码的安全性问题。以下列举两种常用下载操作流程供用户参考,仍然是以上面Project\_L0和Project\_L1为例。操作中涉及到AT-Link的离线下载模式,详细描述可以参考ICP使用文档及AT-Link使用文档。

## 4.1 方案商和终端用户代码分别烧录

方案商先烧录SLIB代码到MCU,然后终端用户再烧录应用代码到MCU,步骤如下:

(1) 方法A: 方案商将编译完成的工程中SLIB部分的代码通过ICP截取保存成BIN或者HEX档: 先将 整个工程下载到MCU(此时不配置SLIB及FAP等信息),然后通过存储器读取功能读取对应SLIB 部分代码(0x08001000~0x08002FFF),文件-存储器数据另存为将其保存为BIN或者HEX,例程 中BIN档命名为slib.bin,如下图。

File	J-Lin	k setting	s A	T-Lir	nk se	tting	s T	arge	t L	angu	lage	He	elp					
_	Save fi	e as			АТЗ	2F40	)3AV	GT7	F	lash	Size:	1024	IKB		.1	זכ	-	2
	Save fla	ash data	as		w:	V1.5	.10							7	- T	XI.		`
	Make e	ncryptio	n file		109	C110	08045	54105	517BC	002					雅	1	恃	5
	Exit				FL	ASH_	DA 0	×	0		@ R	emap	o0 (Us	e PA	11/PA	12 pi	ns)	
		Туре							Select	t	R	emap	o1 (Us	e PB	10/PE	811 pi	ns)	
Me	mory re	ad settin	gs															
	. Γ			_						_	] .	. (			1			
Ado	dress Ox	080010	00	R	lead	size	0x 20	000			Data I	oits (	8 bits	•	J		Read	d
File	info																	
N	- Cl.	name				File	size	٨	Idres	s ran	ae(0)	a						\dd
1.1	o. riie	name							10103	3 i uiii	guioz	<b>y</b>						1uu
1	at3	2f403a_p	roject	l0.he	x	923	2	08	30000	00-08	30003	CB,0	80010	00-00	30011	17,08		alete
1	at3	2f403a_p	roject	_10.he	×	923	2	08	30000	00-08	30003	;; ;CB,0;	80010	00-0	80011	117,08	De	elete
1	at3	2f403a_p	roject	_10.he	×	923	2	08	30000	00-08	ge(0) 30003	;; ;CB,0;	80010	00-0	80011	17,08	De	elete
1	at3	2f403a_p	roject	_10.he	×	923	2	08	30000	00-08	30003	; CB,0	80010	00-0	80011	L17,08 ►	De	elete
1	at3	2f403a_p	roject	_10.he	<b>x</b>	923	2	30	30000	30-00	30003	(CB,0)	80010	00-0	80011	L17,08 ▶	De	elete
1	at3	2f403a_p	roject	_10.he	ex	923	2	30	30000 Flas	00-08	30003	CB,0	80010 e CR0	00-0	80011 fy	L17,08	De	elete
I I Flas	at3	2f403a_p File:at32f4	roject	_10.he	:x  ct_10.ł	923	2	08	Flas	:h CR	30003	CB,01	80010 e CR(	00-0	80011 fy	L17,08 ► <b>Do</b>	De	oad
Flas	h info	2f403a_p File:at32f4 ge:[0x0800	roject 103a_1	_10.he	:x ::: ct_l0.ł )2FFF]	923	2 ecksun	08 08	50000 Flas	00-08	ge(0) 30003	Fil	80010 e CR0	00-0	80011	L17,08	De	oad
Flas Add	h info	File:at32f4 ge:[0x0800	roject 103a_1 1000 (	_10.he	ex (11) (t_l0.h ()2FFF] 3	923 nex che	2 ecksun	08 08 n: 0x0	Flas 01506	00-08 sh CR 2E 8	ge(0) 30003	Fil	80010 e CR0	000-0	80011	L17,08	E De	oad
Flas Add	h info dress rand	File:at32f4 ge:[0x0800 2D	1000 (000 (000 (000 (000 (000 (000 (000	_10.he	ex 	923	2 ecksun 5 46	08 08 n: 0x0 6 0F	Flas 01506 7 46	00-08 ih CR 2E 8 90	9 46	Fil	80010 e CR0 B 25	C veri	fy D EA	E 58	F 19	oad
Flas Add Add 0x08	h info dress range 001000	File:at32f4 ge:[0x0800 2D 47	1000 (0 1 1000 (0 1 1 1000 (0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	_10.he	ex t_l0.ł 2FFF] 3 47 03	923 923 che 4 06 C2	2 ecksun 5 46 F2	08 08 01	Flas 01506 7 46 03	00-08	9 46 F2	Fil	80010 e CR0 B 25 02	C veri C veri C 4F C0	50011 fy D EA F6	E 58 00	F 19 02	A:
Flas Add 0x08 0x08	h info dress randomological de la companya de la co	File:at32f4 ge:[0x0800 2D 47 1D	103a_1 1000 ( 1 E9 F2 21		tri tri tri tri tri tri tri tri tri tri	923 923 chex che 6 6 6 6 6 6 6 00	2 2 5 46 F2 95	08 08 07 07 01 00	Flas 01506 7 46 03 F0	000-08 h CR 2E 8 90 42 58	9 9 46 F2 F8	Fil A 20 00 00	B CR0 B 25 02 24	C veri C veri 4F C0 0C	6 5 6 7 8 0 1 1 8 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1	E 58 00 04	F 19 02 FB	oad
Flas Add Add Ox08 Ox08 Ox08 Ox08 Ox08 Ox08 Ox08 Ox08	h info dress ran 001000 001010 001020 0001030	File:at32f4 ge:[0x0800 2D 47 1D 05	103a_1 1000 0 1 E9 F2 21 F0	2 FF 00 01 07	ex	923 923 hex che 4 06 C2 00 80	2 2 ecksun 5 46 F2 95 02	0x0 0F 01 00 04	Flas 01506 7 46 03 F0 FB	00-08 h CR/ 2E 8 90 42 58 05	9 9 46 F2 F8 F0	Fil A 20 00 00 06	B B 25 02 24 EB	C veri C veri C 0C 80	6 5 5 6 7 6 7 6 7 6 7 6 7 6 7 6 7 7 7 7	► ► ► ► ► ► ► ► ► ► ► ► ► ►	<b>F</b> 19 02 FB 46	A:

图 42. 保存 SLIB 代码

方法B: 方案商将编译完成的工程直接产生bin格式的文件,取其中SLIB区域对应的一段,例如 在KEIL工程中,user选项中添加fromelf.exe --bin --output .\Listings\@L.bin !L,生成对应固件的bin 档,将对应的SLIB段文件添加后缀名.bin格式,本例中改为ER\_SLIB.bin和ER\_SLIB\_DATA.bin,对 应的就是0x08001000地址段的SLIB-INSTRUCTION文件和0x08002000地址段的SLIB-DATA文件, 如下图

Command Items	User Command		Stop on Exi	S	
Before Compile C/C++ File	2				
🗌 🗌 Run #1		2	Not Specified		
Run #2		2	Not Specified		
Before Build/Rebuild					ER IROM1
🔲 Run #1		1	Not Specified		
Run #2		2	Not Specified		ER IROM2
After Build/Rebuild					
🔽 Run #1	fromelf.exebinoutput .\Listings\@L.bin !L	1	Not Specified		ER_SLIB_DATA
Run #2		2	Not Specified		

## 图 43. 生成 SLIB 代码部分 bin 文件

(2) 将ER\_SLIB\_INSTRUCTION.bin和ER\_SLIB\_DATA.bin通过ICP工具,在线烧录到MCU,如下 图

Artery ICP Pr	ogramn	ner_V	2.4.24	ŧ.														DownLoad Form
File J-Link	setting	s A	AT-Lir	nk set	ttings	s T	arge	t l	.angu	age	He	lp						sLib status
-	Part I	Numl	ber:	АТЗ	2F40	)3AV	GT7	F	lashS	ize:	1024	КВ		1	זכ		עכ	SLID status: Disable Remaining usage times: 246
Disconnect	AT-Li	nk-EZ		FW:	V1.5.	.10							- 7	- I	XI.	-	X I	Enable password 0x 55665566 sLib position: Main Flash
AT-Link 🔻	AT-Li	nk SN	1: 8	D109	C110	0804	54105	517BE	002				3	雅	1	<del></del>	カ	Disable password 0x Start sector Sector 20x08001000
	SPI SPI	м		FI	АСН		v 🗆	0		@ R	eman		ο PΔ1	1/24	12 ni	nc)	/ 1	Disable sLib DATA start sector Sector 40x08002000 -
	Type					UA U	Î	Select		© R	eman	1 (Us	se PR1	0/PB	11 pi	ns)		End sector Sector 50x08002800
Memory read	d settin	as									onnap				ar pi			Extra options
											. (	0.1.1				Dee	-	Error the costons of file size
Address 0x	080000	00	H	lead s	size	0x 0	0000	BCC		ata b	oits	8 bit	s 🔻			Kea	a	
File info																		Verify Custom encryption key for verify VEnable sLib
No. File r	name							F	ile siz	e	Addr	ress r	range(	0x)		A	dd	Disable FAP before download
1 ER_S	LIB_INS	TRUC	TION	.bin				2	80		0800	1000	-0800	1117		De	elete	Jump to the user program Enable FAP after download
2 ER_S	LIB_DAT	FA.bin						1	16		0800	2000	-0800	2073				Write software serial number(SN)
															_			Write address 0x 08010000 Button free mode
																		Current CN 0v 0000000
							ſ	Flas	sh CR	c ]	File	e CRO	C verify	y	Do	wnL	oad	
Elash info Ei	e:ER SL	IB DA	ATA.bi	in														Increase step 0x 0000001
Address range	e:[0x0800	2000 0	0x0800	02073]	che	ecksur	n: 0x0	00020	29									Write user system data
Address	0	1	2	3	4	5	6	7	8	9	А	В	С	D	E	F	A: ^	User system data file path
0x08002000	B9	E1	EE	BA	12	22	D0	BA	00	00	00	00	F7	55	72	3B	對	
0x08002010	CF	4E	04	3C	58	C2	0B	3C	00	00	00	80	9E	85	8E	BC	<b>螻:</b> E	
0x08002020	88	DC	0B	BD	9C	A3	08	BD	00	00	00	00	0A	82	8A	3D	塊:	
0x08002030	FO	DB	18	3E	5F	46	64	3E	06	41	80	3E	5F	46	64	3E	疗:	
0x08002040	FO	DB	18	3E	0A	82	8A	3D	00	00	00	00	9C	A3	08	BD	疗:	Start Download Cancel Close
4																		

图 44. ICP 在线烧录 MCU

(3) 或者通过ICP工具配置成离线项目工程保存到AT-Link,然后经过AT-Link离线烧录到MCU,保存 离线项目工程如下图。

I-Link settings AI-Link online config	settings AT	
Offline project		Delete     Creat
Project name slib_dewnload		Device AT32F403A ▼ AT32F403AVGT7 ▼
No. File name	File size	Address range(0x) Storage loca Add
1 ER_SLIB_INSTRUCTION.bin	280	08001000-08001117 Delete
2 ER_SLIB_DATA.bin	116	08002000-08002073
•		•
Frase option Frase the sectors of f	le size	
Lase the sectors of h	ile size	· · · · · · · · · · · · · · · · · · ·
Download times		
Download times		Download interface SWD 🔹
Encryption transmit	Verify	Download interface SWD
Encryption transmit     Write user system data	Verify	Download interface SWD   Reset and run
Encryption transmit     Write user system data     Enable EAP after download	Verify	Download interface SWD   Reset and run
Encryption transmit     Write user system data     Enable FAP after download	] Verify	Download interface SWD   Reset and run
Encryption transmit     Write user system data     Enable FAP after download	] Verify	Download interface SWD   Reset and run
Encryption transmit     Write user system data     Enable FAP after download	Verify	Download interface SWD    Reset and run  Lib settings
Encryption transmit  Write user system data  Enable FAP after download  Software serial number(SN) SPIM  Finable sLib	Verify	Download interface SWD    Reset and run  Lib settings
Encryption transmit  Vrite user system data  Enable FAP after download  Software serial number(SN) SPIM  Enable sLib sLib enable password 0x 55665	Verify	Download interface SWD    Reset and run  Lib settings  Lib position  Main Elash
Encryption transmit     Write user system data     Enable FAP after download      Software serial number(SN) SPIM     Enable sLib     sLib enable password 0x 55665	Verify	Download interface SWD    Reset and run  Lib settings  clib pocition Main Flash  Start sector Sector 20x08001000
Encryption transmit Write user system data Enable FAP after download Software serial number(SN) SPIM Enable sLib sLib enable password 0x 55665 Disable sLib before download	Verify settings S	Download interface SWD    Reset and run  Lib settings  Lib position Main Flash  Start sector Sector 20x08001000  DATA start sector Sector 40x08002000
<ul> <li>Download times</li> <li>Encryption transmit</li> <li>Write user system data</li> <li>Enable FAP after download</li> <li>Software serial number(SN) SPIM</li> <li>Enable sLib sLib enable password 0x 55665</li> <li>Disable sLib before download sLib disable password 0x</li> </ul>	Verify settings S	Download interface SWD    Reset and run  Lib settings  Lib pocition Main Flash    DATA start sector Sector 20x08001000   Lata start sector Sector 50x08002800
<ul> <li>Download times</li> <li>Encryption transmit</li> <li>Write user system data</li> <li>Enable FAP after download</li> <li>Software serial number(SN) SPIM</li> <li>Enable sLib sLib enable password 0x 55665</li> <li>Disable sLib before download sLib disable password 0x</li> </ul>	Verify settings S	Download interface SWD   Reset and run  Lib settings  Start sector Sector 20x08001000   DATA start sector Sector 40x08002000   End sector Sector 50x08002800   Load parameters Save parameters
<ul> <li>Download times</li> <li>Encryption transmit</li> <li>Write user system data</li> <li>Enable FAP after download</li> <li>Software serial number(SN) SPIM</li> <li>Enable sLib</li> <li>sLib enable password 0x 55665</li> <li>Disable sLib before download</li> <li>sLib disable password 0x</li> </ul>	Verify settings 5 566	Download interface SWD    Reset and run  Lib settings  Start sector Sector 20x08001000   DATA start sector Sector 40x08002000   End sector Sector 50x08002800   Load parameters Save parameters
Encryption transmit Write user system data Enable FAP after download Software serial number(SN) SPIM Enable sLib sLib enable password 0x 55665 Disable sLib before download sLib disable password 0x	Verify	Download interface SWD    Reset and run  Lib settings  Start sector Sector 20x08001000  DATA start sector Sector 40x08002800  End sector Sector 50x08002800  Load parameters Save parameters
Encryption transmit Write user system data Enable FAP after download Software serial number(SN) SPIM Enable sLib sLib enable password 0x 55665 Disable sLib before download sLib disable password 0x	Verify settings 566	Download interface SWD   Reset and run  Lib settings  Lib position Main Elash   Start sector Sector 20x08001000   DATA start sector Sector 40x08002000   End sector Sector 50x08002800   Load parameters Save parameters
Encryption transmit Write user system data Enable FAP after download Software serial number(SN) SPIM Enable sLib sLib enable password 0x 556655 Disable sLib before download sLib disable password 0x Open project Save project file	Verify settings 5 5 6	Download interface SWD   Reset and run  Lib settings  Start sector Sector 20x08001000   DATA start sector Sector 40x08002000   Load parameters Save parameters  Save project to AT-Link Close

图 45. AT-Link 离线烧录到 MCU

(4) 经过步骤2或者步骤3,终端用户拿到烧录好SLIB部分的MCU,此时SLIB状态会显示为已启用,终端用户通过在线烧录或者离线烧录应用代码到MCU完成整个过程,在线烧录如下图。



J-LITIK S	ettings	AT-Lir	nk set	tings	Targ	get I	angu	age	Help						sLib	tatus							
	Part Nue	aber	AT33	PE403	AVGT	7 1	lachS	izo: 10	2468		47	٦Г			sLib	status: Enak	ble		Rem	aining usa	age times: 24	5	
isconnect	are the					· ·	Tastia	120. 10	2410	= ;	<b>- 1</b>	71	<u> </u>	Υ	Ena	le password	0x 556	55566	sLib	position:	Main Flash	1	Ŧ
T-link -	AT-Link	=2 5N: 8	FW: 0	11008	, 04541	0517B	D02				<b>##</b>	1	÷	₽∥	Disa	ole password	d Ox		Start	sector	Sector 2	0x08001000	-
	_									- 1	JÆ	7	ч.	~ II			Die	able sLib	DAT	A start sec	ctor Sector 4	0x08002000	-
	SPIM		FLA	SH_D/	4 0x	0		Rem	ap0 (U	se PA1	L1/PA1	L2 pin	is)						End	sector	Sector 5	0x08002800	-
	Туре					Selec	t	Rem	ap1 (U	se PB1	LO/PB1	L1 pin	is)										
Aemory read	settings														Extra	options							
Address 0x (	08001000	F	Read s	ize 0>	0000	0118	D	ata bits	8 bit	s 🔻	) [	F	Read		Er	se the sector	rs of file	size		-	Disable sLib b	efore downlo	ad
1																/erify	Custom	encryption	key for ve	rify: 🔽	Enable sLib		
ile into																					Disable FAP be	efore downlo	ad
No. File na	ame			File si	ze	Addres	s rang	je(0x)	00000		200404	47	Add			ump to the u	iser proc	iram			Enable FAP aft	er download	
1 at5214	iusa_proje	ct_i1.ne	ex	7428		080000	00-08	000388	08003	000-08	300494	47 📘	Delet			anno co uno e					_ Endlore frit are	er aenneaa	
					_							_	Delet			Nrite coffwar	re ceriel	wmber(SN	,				
													belet			Write softwar	re serial i	number(SN	)	-			
1												•	Delet			Write softwar	reserial	number(SN 08010000	)		Button free mo	ode	
•			III									•	Delet			Write softwar Write addr Current SN	re serial i ress 0x [ I 0x [	08010000			Button free mo	ode	
•						Fla	sh CR(		File CR	C verif	fy	) Dow	vnLoa			Write softwar Write addr Current SN Increase st	ressox ( I Ox ( tep Ox (	08010000 00000001 00000001			Button free mo	ode	
۲ lash info File	e:at32f403a	_proje	III ct_l1.he	ex		Fla	sh CR(		File CR	C verif	fy	, Dow	vnLoa			Write softwar Write addr Current SN Increase st	re serial i ress 0x [ I 0x [ tep 0x [	08010000 00000001 00000001			] Button free mo	ode	
<li>1ash info File Address range:</li>	e:at32f403a (0x0800000	_proje	"" ct_l1.hr 003BB]	ex	ess ran	Fla ge:[0x08	sh CR(	) ; ) 0x0800	File CR	C verif	fy	, Dow	vnLoa BAA7			Write softwar Write addr Current SN Increase st Write user sy	re serial i ress 0x [ I 0x [ tep 0x [	oumber(SN 08010000 00000001 00000001 a			] Button free mo	ode	
۲ العلم المراجع مراجع المراجع المراجع المراجع المراجع مراجع المراجع المراجع المراجع المراجع مراجع المراجع المراجع المراجع مراجع م مراجع مراجع	e:at32f403a (0x0800000 0 1	_projec 0 0x0800 2	"" ct_l1.hr 003BB] 3	ex Addr	ess ran 5 6	Fla ge:[0x00	sh CR0 800300	0x0800	File CR 1947] 6 B	C verif checks	fy	, Dow 000AB	vnLoad BAA7 F A			Write softwar Write addr Current SN Increase st Write user sy User system	re serial ress 0x [ l 0x [ tep 0x [ rstem dat data file	ownber(SN 08010000 00000001 00000001 a path			Button free mo	ode	
<ul> <li>Iash info</li> <li>File</li> <li>Address range:</li> <li>Address</li> <li>x08000000</li> </ul>	e:at32f403a (0x08000000 0 1 38 13	1_projet 0 0x0800 2 00	"" ct_l1.hr 003BB] 3 20	ex Addr 4 01 3	ess ran 5 6 10 00	Fla: ge:[0x04 7 0 08	sh CR0 8003000 8 F7	9 0x0800 9 <b>4</b> 35 01	File CR 1947] 0 8 08	C verif checkso C EF	fy um: 0x0	Dow 0000AB	<b>vnLoa</b> 8AA7 <b>F A</b> 08 8:			Write softwar Write addr Current SN Increase st Write user sy User system	re serial i ress 0x [ I 0x [ tep 0x [ rstem dat data file	08010000 00000001 00000001 a path			Button free mo	ode	
<ul> <li>Address</li> <li>Address</li> <li>Address</li> <li>x08000000</li> <li>x0800010</li> </ul>	e:at32f403a (0x0800000 0 1 38 13 F3 35	2 00 00 00	III ct_l1.he 003BB] 3 20 08	Addr Addr 01 S CF S	ess ran 5 6 10 00	Fla ge:[0x00 7 0 08 0 08	sh CR0 800300 8 F7 75	9 0x0800 9 4 35 00 36 00	File CR 1947] 0 8 0 08 0 08	C verif checkso C EF 00	fy um: 0x0	Dow 0000AB	vnLoad 8AA7 F A 08 8: 00 ?.			Write softwar Write addr Current SN Increase st Write user sy User system	re serial i ress 0x ( i 0x ( tep 0x ( rstem dat	08010000 00000001 00000001 a path	)		Button free me	ode	
<ul> <li>ash info</li> <li>File</li> <li>Address range:</li> <li>Address value</li> <li>Address value</li></ul>	e:at32f403a (0x08000000 0 1 38 13 F3 35 00 00	2 00 00 00 00 00 00	t_l1.he 003BB] 20 08 00	ex Addr Addr 01 5 CF 5 00 0	ess ran 5 6 10 00 15 00	Flar ge:[0x08 7 0 08 0 08 0 00	sh CR0 8003000 8 F7 75 00	9 0x0800 9 4 35 00 36 00 00 00	File CR 1947] 0 08 0 08 0 08	C verif checks C EF 00 FB	5y Um: 0x0 35 0 35 0 35 0	Dow     Dow     OOOAB     Coo 0     OO 0	<b>vnLoa</b> 8AA7 <b>F</b> A 08 8: 00 ?. 08			Write softwar Write addr Current SN Increase st Write user sy User system	re serial i ress 0x [ 1 0x [ tep 0x [ rstem dat data file	08010000 00000001 00000001 a path			Button free me	ode	
Iash info     File       Address range:     x0800000       x08000010     x0800010       x08000020     x08000030	e:at32f403a (0×0800000 38 13 F3 35 00 00 D3 35	2 00 00 00 00 00 00 00	III ct_l1.he 003BB] 20 08 00 08 00 08	Addr 4 01 3 CF 3 00 0 00 0	ess ran 5 6 10 00 15 00 10 00	Fla: ge:[0x04 7 0 08 0 08 0 00 0 00	sh CR0 8003000 8 F7 75 00 F9	9 0x0800 9 4 35 00 36 00 00 00 35 00	File CR 1947] 0 8 0 08 0 08 0 00 0 08 0 00 0 08 0 08 0 0 0 0 0	C verif checks EF 00 FB FD	fy 0x0	Dow     000AB     E     00     0     0     0     0	VNLOA 3AA7 F A 08 8: 00 ?. 08 08 08 ?.			Write softwar Write addr Current SN Increase st Write user sy Jser system	re serial i ress 0x [ i 0x [ tep 0x [ data file	08010000 00000001 00000001 a path	)		Button free mo	ode	

### 图 46. 终端用户烧录代码到 MCU

## 4.2 方案商和终端用户代码合并烧录

方案商的SLIB代码和终端用户的应用代码整合到一个离线项目工程中,通过AT-Link离线烧录一次下载到MCU,步骤如下:

- (1) 方案商将编译完成的工程按照上一章节所述方法处理,得到SLIB部分的BIN档命名为slib.bin。
- (2) 方案商通过ICP制作离线项目工程并保存到PC,可以根据最终需求配置各种参数,比如限制下 载次数、项目文件绑定AT-Link、下载完成后开启FAP等,保存离线项目工程如下图

注意:离线项目工程本身已经经过加密,为进一步提升安全性,方案商还可以将slib.bin制作成加密的slib.benc 文件再添加到离线项目工程中,但此时的离线项目工程只能在对应匹配加密秘钥的AT-Link上才能使用



Offline project	- Delete	Creat	
Project name slib_project           No.         File name           1         ER_SLIB_INSTRUCTION.bin           2         ER_SLIB_DATA.bin           4         Erase option           Erase option         Erase the sectors of           Download times         Encryption transmit           Write user system data         Enable FAP after download	Device AT32F403A  AT32F403AVGT File size Address range(0x) Storage loca 280 08001000-08001117 116 08002000-08002073 "  Tile size Download interface SWD Verify Reset and run	7 V Add Delete	
Software serial number(SN)     SPIN       Image: Software serial number(SN)     SPIN       Image: Software serial number(SN)     Software serial number(SN)       Image: Software serial number(SN)	4 settings       sLib settings         5566       sLib position       Main Flash         5566       Start sector       Sector 20x0800100         DATA start sector       Sector 40x0800200         End sector       Sector 50x0800280         Load parameters       Save project to AT-Link	0 v 10 v 10 v arameters	Image: Straight of the settings       Image: Straight of the specified AT-Link.         Image: Straight of the specified AT-Link SN :       CFD275220040B56D0117C502         Image: Straight of the specified AT-Link SN :       CFD275220040B56D0117C502         Image: Straight of the specified AT-Link AT-Link AIN :       F00FA432D013A913

### 图 47. 制作离线项目工程

(3) 终端用户拿到该离线项目工程,用ICP打开项目文件,通过添加文件功能,可以添加应用代码部 分到该离线项目工程,然后再保存到PC或者直接存储到AT-Link,通过执行离线下载完成操作, 项目文件添加方法如下图。

注意:为防止代码泄露被破解等风险,离线项目工程添加代码文件时其余配置都不可更改,所以需要方案商预 先将最终配置设置好。

Ziiiiie	project		- Delete		Creat
D					
Project	name slib_project	D	AT32F403A	▼ AT32F403AVGT7	×
No.	File name	File size	Address range(0x	) Storage loca	Add
1	ER_SLIB_INSTRUCTION.bin	280	08001000-080011	17	Delete
2	et22f402e project l1 boy	056	08002000-080020	73 DD *	
•		III		• •	
Erase	option Erase the sectors of f	file size			
	vnload times 0		Download interf	ace SWD	
	CONSISTER STOLEN W				<b>W</b>
	austion transmit	7 Vorify		3100	
Enci	ryption transmit	Verify	Reset and run	1	
Enci Krit Enal	ryption transmit te user system data ble FAP after download	∠ Verify	Reset and run	1	
<ul> <li>Encr</li> <li>Writ</li> <li>Enal</li> </ul>	ryption transmit te user system data ble FAP after download are serial number(SN) SPIN	Verify	Reset and run     Lib settings	1	
<ul> <li>□ Encr</li> <li>□ Writ</li> <li>□ Enal</li> <li>Softw</li> <li>□ Er</li> </ul>	ryption transmit te user system data ble FAP after download are serial number(SN) SPIN	Verify	Reset and run     Lib settings     sLib position	Main Flash	•
<ul> <li>Enci</li> <li>Writ</li> <li>Enal</li> <li>Softw</li> <li>Z Er</li> <li>sLib e</li> </ul>	ryption transmit te user system data ble FAP after download tare serial number(SN) SPIN hable sLib enable password 0x *****	Verify Verify A settings	Reset and run <b>Lib settings</b> sLib position     Start sector	Main Flash	× 
<ul> <li>□ Enci</li> <li>□ Writ</li> <li>□ Enal</li> <li>Softw</li> <li>□ V Er</li> <li>sLib e</li> <li>□ Di</li> </ul>	ryption transmit te user system data ble FAP after download mathematical states enable states isable states before download	Verify	Reset and run     Reset and run     Stib settings     sLib position     Start sector     DATA start sector	Main Flash Sector 20x08001000 Sector 40x08002000	
<ul> <li>□ Enci</li> <li>□ Enci</li> <li>□ Writ</li> <li>□ Enal</li> <li>□ Softw</li> <li>□ V Er</li> <li>sLib e</li> <li>sLib e</li> </ul>	ryption transmit te user system data ble FAP after download tare serial number(SN) SPIN hable sLib enable password 0x ***** isable sLib before download disable password 0x	Verify	Reset and run  Lib settings  sLib position Start sector DATA start sector End sector	Main Flash Sector 20x08001000 Sector 40x08002000	
<ul> <li>□ Enci</li> <li>□ Writ</li> <li>□ Enal</li> <li>Softw</li> <li>□ Zin</li> <li>□ Sib e</li> <li>□ Di sLib e</li> </ul>	ryption transmit te user system data ble FAP after download tare serial number(SN) SPIN hable sLib enable password 0x ***** isable sLib before download disable password 0x	Verify Verify	Reset and run	Main Flash Sector 20x08001000 Sector 40x08002000 Sector 50x08002800	
<ul> <li>□ Enci</li> <li>□ Writ</li> <li>□ Enal</li> <li>Softw</li> <li>☑ Er</li> <li>sLib c</li> </ul>	ryption transmit te user system data ble FAP after download tare serial number(SN) SPIN hable sLib enable password 0x ***** isable sLib before download disable password 0x	Verify Verify	Reset and run  Lib settings  sLib position Start sector DATA start sector End sector Load param	Main Flash Sector 20x08001000 Sector 40x08002000 Sector 50x08002800 meters Save par	× 
<ul> <li>□ Enci</li> <li>□ Writ</li> <li>□ Enal</li> <li>Softw</li> <li>✓ Er</li> <li>sLib c</li> </ul>	ryption transmit te user system data ble FAP after download tare serial number(SN) SPIN hable sLib enable password 0x ***** isable sLib before download disable password 0x	Verify  A settings	Reset and run	Main Flash Sector 20x08001000 Sector 40x08002000 Sector 50x08002800 meters Save par	× 
Softw Softw Enal	ryption transmit te user system data ble FAP after download mare serial number(SN) SPIN mable sLib enable password 0x ***** isable sLib before download disable password 0x	Verify  A settings	Reset and run	Main Flash Sector 20x08001000 Sector 40x08002000 Sector 50x08002800 neters Save par	· · · · · · · · · · · · · · · · · · ·

图 48. 添加项目文件

# 5 版本历史

表 2. 文档版本历史

日期	版本	变更
2021.05.21	2.0.0	初始版本
2021.11.01	2.0.1	格式优化
2022.4.13	2.0.2	修改安全库区在flash中的配置范围

#### 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。

© 2022 雅特力科技 (重庆) 有限公司 保留所有权利