

# AN5421

应用笔记

# STM32L5 系列微控制器和 TrustZone<sup>®</sup> 开发入门

#### 引言

本文档为使用 EWARM 和 MDKARM 软件工具链在 STM32L5 系列微控制器上进行应用开发提供参考。

本应用笔记提供了为 Arm<sup>®</sup> Cortex<sup>®</sup>-m33(Armv8\_M 架构)的器件构建和调试安全和非安全应用程序的基础知识。

本文首先概述 Arm<sup>®</sup> Cortex<sup>®</sup>-M33 和 TrustZone<sup>®</sup>概念。

本应用笔记还描述当通过 TZEN 选项位启用 TrustZone<sup>®</sup>之后,如何使用 EWARM 和 MDKARM 调试 STM32L5 系列微控制器。

1	概述
	本文档适用于 STM32L5 系列单核 Arm <sup>®</sup> 的微控制器。
提示	Arm is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

arm

# 2 Arm<sup>®</sup> Cortex<sup>®</sup>-M33 内核概述

Arm<sup>®</sup> Cortex<sup>®</sup>-M33 是首次完整的集成了 ARMv8 指令集及 TrustZone 安全技术和数字信号处理功能。该处理器支持大量灵活的配置选项,以促进各种应用程序的部署,并提供专用的协处理器接口,用于加速经常使用的计算密集型操作。Cortex<sup>®</sup>-M33 提供性能、功率、安全性和生产效率之间的最佳平衡。

57

# 3 Armv8-M的 TrustZone<sup>®</sup>概念

带有 TrustZone<sup>®</sup>的 Cortex<sup>®</sup>-M33 处理器有两个安全状态(参见图 1)和一些相关的特性:

- 安全状态
- 非安全状态
- 四个堆栈和四个堆栈指针寄存器
- 硬件栈限制检查
- 支持类似于可编程 MPU 的安全属性单元(SAU)
- 系统安全通知接口
- 限定非安全(NS)域只能通过预定义的入口点访问安全代码
- 当切换到非安全时,异常硬件自动保存和清除安全寄存器状态
- 中断或异常控制的扩展存储, SysTick
- 针对每个安全和非安全部分的内存保护单元。





提示 TrustZone<sup>®</sup>被启用后,系统默认在安全状态下启动。

# 4 SAU / IDAU - TrustZone<sup>®</sup>概念

TrustZone<sup>®</sup>安全功能由 FLASH\_OPTR 寄存器中的 TZEN 位激活。TrustZone<sup>®</sup>被启用后, SAU 和 IDAU 根据安全和非安全状态定义访问权限。

- IDAU:将第一个内存分区配置为安全或非安全可调用属性。IDAU内存映射分区是不可配置的,它由硬件配置决定。
- SAU: 8个区域,用于覆盖 IDAU 以设置安全区域和确认非安全区域。
- 首先根据 IDAU 安全属性选择安全状态,然后结合 SAU 安全属性选择安全状态。最终的安全属性是 IDAU 和 SAU 的最高安全设置。
- "安全"安全属性具有最高的安全优先级,非安全可调用的安全优先级次之,非安全属性的安全优先级最低。默 认情况下,任何未定义的区域都是安全的

TrustZone<sup>®</sup>安全被激活之后,默认安全状态如下:

- CPU: Cortex<sup>®</sup>-M33 在复位后处于安全状态。启动地址必须在安全区域中。
- 内存映射: SAU 在复位后是完全安全的。整个内存映射是完全安全的。最多有 8 个 SAU 可配置区域用于安全属性。
- Flash 存储器:
  - Flash 安全区域是由安全水印用户选项定义的。所有闪存都是完全安全的。
  - Flash 块的功能在复位后是非安全的。即使所有的闪存通过 IDAU/SAU 和闪存安全水印选项字节配置都 是非安全的,也可以使用基于闪存块的特性配置易失性安全区域:使用 Flash 块的配置寄存器,任何页 面都可动态编程为安全模式。
  - SRAM: 所有 SRAM 在复位后是安全的。基于内存保护块的控制器(MPCBB)是安全的。
- 非安全内存视图与其他 Cortex<sup>®</sup>-M 内核相同。
- 安全内存空间分为两种内存类型:
  - 安全:包含安全程序代码和数据,如栈区和堆区。
  - 非安全可调用(NSC):包含入口函数(例如 API 的入口点),这是为了防止非安全应用程序分为无效入口点。

#### 5 调试模式

#### 5.1 侵入式调试

侵入式调试定义为一种调试过程-用户控制并观察处理器活动。大多数调试功能被认为是侵入式调试,因为它们允许用户停止处理器并修改其状态。 DBGEN和 SPIDEN 控件都具有侵入式调试权限。

#### 5.2 非侵入式调试

非侵入式调试定义为一种调试过程 - 用户观察处理器但不进行控制。嵌入式跟踪宏单元<sup>™</sup>(ETM)接口和性能监视 器寄存器是非侵入式调试的特性。 NIDEN 和 SPNIDEN 控件都具有非侵入式调试权限。允许侵入式调试的时候总是允许非侵入式调试。

### 6 调试访问

# 6.1 安全调试访问

安全调试访问在所有内存区域和器件外设范围内提供了对所有指令执行的完全可见性。它允许跟踪和调试运行在目标上的安全和非安全软件。 仅在此模式下才能调试安全固件。 在安全状态下运行的代码可以访问安全和非安全信息。

#### 6.2 非安全调试访问

非安全调试视图保护安全内存和外设。在非安全模式下,这些对于调试器是不可见的。调试和跟踪功能仅限于非安 全系统资源。

#### Flash 存储器保护 7

#### 7.1 TrustZone<sup>®</sup>被禁用后的读出保护级别

有三个读出保护级别,如下所示:

- 级别 0: 所有对用户闪存的读取/编程/擦除操作被允许。
- 级别 1: 针对调试器和存储在 RAM 中的潜在恶意代码,对闪存内容进行保护,无法读出。
- 级别 2: 所有的调试功能被禁用,不能再从 SRAM 和系统内存启动。

#### 7.2 TrustZone<sup>®</sup>被禁用后的 RDP 级别转换流程

TZEN 被清除后的 RDP 级别转换流程在图 2 中加以说明。



#### 7.3 TrustZone<sup>®</sup>启用后的读出保护级别

除了前面提到的 RDP 级别设置,有一个新的 RDP 级别名为 0.5,它允许以下功能:

- 所有对非安全闪存的读写操作都是可行的。禁止对安全区域的调试访问。仍然可以调试访问非安全区域。
- 非安全调试模式:当 CPU 处于非安全状态时,可以进行非安全调试。



# **7.4** 当 **TrustZone**<sup>®</sup> 启用后, **RDP** 级别转换流程 设置了 **TZEN** 后的 **RDP** 级别转换流程在图 3 中加以说明。



5TM32CubeProgramme

σ×

# 8 从安全/非安全项目开始

EWARM 和 MDK-ARM 提供相似的方法来支持 STM32L5 系列微控制器。这是通过两个独立的项目(安全和非安全)完成的。

- 第9节 提供 MDK-ARM 项目说明。
- 第 10 节 提供 EWARM 的说明。
- 第 11 节 提供 CubeIDE 的说明。

每个部分都提供逐步说明,以使用 STM32L5 系列微控制器讲解安全和非安全部分的项目设置。

首先,使用 STM32CubeL5 包(STM32Cube\_FW\_L5)中的模板,它由两个子项目组成:一个用于安全应用程序部分,另一个用于非安全应用程序部分。

在开始之前,必须使用 STM32CubeProgrammer 设置选项字节,详情请参见项目的 readme.txt。该工具可从 www.st.com 下载,并在图 4 中加以说明。

#### 图 4. 使用 STM32CubeProgrammer 配置选项字节

STM32 Cube	eProgrammer			1	f 🕒 У 🛧 🏹
	Option bytes				🛑 Not connected
	TZEN		Unchecked : Global TrustZone security disabled Checked : Global TrustZone security enabled Hide anotherine for state anable	ST-LINK	- Connect
<b>.</b>	HDP1EN		Inter protection may area stoke Unchecked : No HDP area 1 Checked : HDP first area is enabled	Serial number	ST-LINK configuration 003500163137511533333639
OB	HDP1_PEND	Value 0x0 Address 0x8000000	End page of first hide protection area	Port	SWD 👻
			Hide protection second area enable	Frequency (kHz)	24000 👻
	HDP2EN		Unchecked : No HDP area 2 Checked : HDP second area is enabled	Mode	Under reset 🔹
	HDP2_PEND	Value 0x0 Address 0x8000000	End page of second hide protection area	Access port	0 -
	NSBOOTADD0	Value 0x10000 Address 0x8000000	Non-secure Boot base address 0	Shared	Hardware reset
	NSBOOTADD1	Value 0x17f20( Address 0xbf90000	Non-secure Boot base address 1	External loader	Usabled •
	SECROOTADDO	Value 0x18000 Address 0xc000000	Service hint hase address 0	Target voltage	3.27 V
	000000000		The boot is always forced to base address value programmed in SECBOOTADD0	Firmware version	V3J3M2
	BOOT_LOCK		Unchecked : Boot based on the pad/option bit configuration Checked : Boot forced from base address memory		Firmware upgrade
	V Serure Area 1				
	Name	Value	Description		
	SECWM1_PSTRT	Value 0x0 Address 0x8000000	Start page of first secure area		
	SECWM1_PEND	Value 0x7f Address 0x803f800	End page of first secure area		
	Write Protection 1				
	V Secure Area 2				
	Name	Value	Description		
	SECWM2_PSTRT	Value 0x1 Address 0x8040800	Start page of second secure area		
	SECWM2_PEND	Value 0x0 Address 0x8040000	End page of second secure area	J	
			Apply Read		<b>.</b>
	Log		Verbosity level 💿 1 💿 2 💿 3	Device	Device information
	16:09:16 : OPTION BYTE PROGRA 16:09:16 : Option Bytes succe 16:09:19 : Disconnected from 16:09:19 : ST-LINK SN : 00350 16:09:19 : ST-LINK FW : V3139 16:09:19 : V01200 : 2720	MMING VERIFICATION: assfully programmed device. DI65137511533333639 2		Type Device ID Flash size	
?				CPU	

# 9 将 MDK-ARM 用于带 Trust Zone 的 Cortex<sup>®</sup>-M33

最新版本的 MDK-ARM (Keil<sup>®</sup>)可从官方 Arm<sup>®</sup> Keil<sup>®</sup>网站下载。MDK-ARM (Keil<sup>®</sup>)默认安装在 PC 本地硬盘上的 C:\Keil 目录下;安装程序将在开始菜单中创建 μVision<sup>®</sup> 5 快捷方式。 本节使用 MDK-ARM v5.27.0.0 和 STM32L562-DK 探索板。

#### 9.1 安全项目设置

本节概述安全项目设置。

1. 打开多项目工作区文件"Project.uvmpw",该文件允许用户同时处理两个项目。打开的项目显示在项目浏览器中,如图 5 中所示。

Project	<b>џ</b>	x
🖃 🚰 WorkSpace		
🖶 🏀 Project: Project_s		
😑 💭 STM32L562E-DK_s		
🕀 🛅 Doc		
🕀 🔜 Drivers/BSP/STM32L562E-Discovery		
🕀 🔜 Drivers/BSP/Components		
🕀 🛅 Drivers/STM32L5xx_HAL_Driver		
🗈 🛅 Drivers/CMSIS		
🗉 🛅 Example/MDK-ARM		
🕀 🦢 Example/User		
🖮 🛠 Project: Project_ns		
🖨 ᇶ STM32L562E-DK_ns		
🗈 🛅 Doc		
🗉 🛅 Drivers/BSP/Components		
🗉 🛅 Drivers/BSP/STM32L562E-Discovery		
🗉 🦾 Drivers/STM32L5xx_HAL_Driver		
🗉 🧰 Drivers/CMSIS		
🗉 应 Example/MDK-ARM		
🖃 🚞 Example/User		

2. 将 project\_s 设为活动项目,参见图 6。



Project		<b>д </b> 🗙
🖃 📅 WorkSpace		
🗄 🔧 Project: Project_s		
🗄 🎕 Project: Project_ns	Set as Active Project	

#### 图 5. MDK-ARM 项目结构

**3**. 选择正确的器件,方法是:打开配置窗口,然后选择: Project / Options for Target / Device,接下来从列表中选择器件(参见图 7)。

Options for Target 'STM32L562E-DK_s'	×
Device Target   Output   Listing   User   C/C++ (AC6)	Asm Linker Debug Utilities
Software Packs       Vendor:       STMicroelectronics       Device:       STM32L562ZETx       Toolset:       ARM       Search:	Software Pack Pack: Keil.STM32L5xx_DFP.1.0.9 URL: <u>http://www.keil.com/pack</u>
STM32L562QEIx STM32L562QEIxQ STM32L562RETx STM32L562VETx STM32L562VETxQ STM32L562ZETx STM32L562ZETxQ I	M Cortex-M33 is the most configurable of all Cortex-M rs. It is a full featured microcontroller class processor based on the ARMv8-M mainline ure with ARM TrustZone security.
ОК Са	ancel Defaults Help

图 7. 设备选择

 $\times$ 

# 4. 从 Project / Options for Target / Target / Code Generation 部分选择"Software Model (软件模型)"为"Secure (安全)"。确保选择正确的内存区域。请参见图 8:

- 安全启动地址:位于 0x0C000000 的闪存:安全闪存
- 安全启动地址: 位于 0x30000000 的 SRAM1: 安全 SRAM

#### 图 8. Project\_s 目标选项

Device Target Output Listing User C/C++ (AC6) Asm Linker Debug Utilities Code Generation STMicroelectronics STM32L552ZE-Q ARM Compiler: V6.11 • Xtal (MHz): 12.0 Software Model: Secure Mode • Operating system: None • System Viewer File: ✓ Use MicroLIB 🗌 Big Endian STM32L5x2.svd Floating Point Hardware: Single Precision Ŧ Use Custom File Read/Only Memory Areas Read/Write Memory Areas default off-chip Start Size Startup default off-chip Start Size Nolnit ROM1:  $\mathbf{C}$ RAM1: ROM2: RAM2: Π Ċ Г ROM3: RAM3: Ċ Г on-chip on-chip 0x8040000 0x40000  $\odot$ 0x20018000 0x18000 IROM1: IRAM1:  $\overline{\mathbf{v}}$ 0xC000000 0x40000 C  $\mathbf{\nabla}$ 0x18000 Г IROM2: IRAM2: 0x30000000 OK Cancel Defaults Help

# Options for Target 'Project\_s'

5. 确保安全的非安全可调用函数(NSC)对象文件"secure\_nsclib.o"在 Misc Controls 部分下的 Project / Options for Target / Linker 中定义。

使用[--import\_cmse\_lib\_out ..\lib\nsclib\_Secure.o]命令创建输出库: nsclib\_Secure.o。 该文件是在编译安全项目期间自动生成的,它包含所有使用前缀

\_\_attribute\_\_((cmse\_nonsecure\_entry))声明的非安全可调用函数。 请参见图 9。

Options for Target 'Project_s'	×
Device       Target       Output       Listing       User       C/C++ (AC6)       Asm       Linker       Debug       Utilities         Use       Memory Layout from Target Dialog       X/O Base:	
Scatter File Edit	]
Misc controls	
Linker -cpu=Cortex-M33 *.o control -library_type=microlib -strict -scatter ".\Project_s.sct"	
OK Cancel Defaults Help	

图 9. Project\_s 链接器配置

在分散加载文件部分下,检查该文件是否包含正确的地址,如图 9 中所示。 该文件由链接器使用,并决定内存布局的组织方式。分散加载文件示例在图 10 中给出。

#### 图 10. 分散加载文件示例

Project\_s.sct 1 2 ; \*\*\* Scatter-Loading Description File generated by uVision \*\*\* 3 . 4 LR IROM2 0x0C000000 0x00040000 { ; load region size\_region 5 ER IROM2 0x0C000000 0x0003E000 { ; load address = execution address 6 7 \*.o (RESET, +First) 8 \* (InRoot\$\$Sections) 9 .ANY (+RO) 10 .ANY (+XO) 11 12 RW IRAM2 0x30000000 0x00018000 { ; RW data 13 .ANY (+RW +ZI) 14 1 15 } 16 LR\_IROM3 0x0C03E000 0x00002000 { ; load region size\_region
ER\_IROM3 0x0C03E000 0x00002000 { ; load address = execution address 17 18 19 \*(Veneer\$\$CMSE) ; check with partition.h 20 } 21 } 22

6. 从 Project / Options for Target / Debug 选择"ST-LINK Deb ugger"为调试器。请参见图 11。

#### 图 11. 目标选项调试



如果"ST-LINK Debugger"没有显示在列表中:

- a. 前往 C:\Keil install directory
- b. 打开 TOOLS.INI 文件并进行以下更改:
  - i. 查找[ARMADS]:

所有基于 Armv8M 的器件都需要处理器 SARMV8M.DLL。TOOLS.INI 文件包含 CPUDLL3 = SARMV8M.DLL(TDRV2、TDRV13、TDRV14、TDRV15、TDRV16)。 在本例中, ST-Link 驱动程序注册为 TDRV6,且可能根据项目 TDRV6=STLink\ST-LINKIII-KEIL\_SWO.dll("ST-Link Debugger")的不同而变化。

ii. 添加 TDRV6 到 CPUDLL3= SARMV8M.DLL:CPUDLL3 = SARMV8M.DLL(TDRV2、TDRV6、 TDRV13、TDRV14、TDRV15、TDRV16)中的列表。

#### 7. 从"Debug"设置选项卡确保调试器已连接,如图 12 中所示。

- Debug Adapter	SW Dev	vice		
Unit: ST-LINK/V3	SWDIO	IDCODE 0x0BE12477	Device Name ARM CoreSight SW-DP	Move Up
Serial Number:           002A001A3137511533333639           Version: HW: V3         FW: V3J3M2           ✓         Check version on start	⊙ Aut ⊂ Ma Add	tomatic Detection nual Configuration	ID CODE:	Down
		001010	puale in terr	AP: JU
Target Com Port: SW  Clock Req: 1.800 MHz Selected: 1 MHz				AP: JU
Target Com Port: SW  Clock Req: 1.800 MHz Selected: 1 MHz Performance Profile: High Perf Dehus				AP: JU

#### 图 12. 调试配置

 $\times$ 

从"Flash Download"选项卡选择正确的闪存加载程序(参见图 13):

- "Download Function(下载功能)":设置 Flash 操作。
- 用于算法的 RAM: 定义用于载入和执行编程算法的地址空间。通常,地址空间位于片上 RAM 中。
- "Program Algorithm (编程算法)":包含编程 Flash 的定义。

#### 图 13. 闪存加载程序设置

Cortex-M Target Driver Setup

Debug Trace Flash Download				
Download Function C Erase Full Chip Erase Sectors C Do not Erase	<ul> <li>✓ Program</li> <li>✓ Verify</li> <li>✓ Reset and Run</li> </ul>	RAM for Algorithm	Size: 0x0800	
Programming Algorithm     Description     ISTM2215v 512 DupLOC00	Device Size Dev	ice Type Addre		]
	JIZK OIK			1
		State 0-0000000	Size: 0-0000000	
	Add	Remove	Size. Jux00080000	
			OK Cancel	Apply

#### 9.2 非安全项目设置

1. 设置 project\_ns 活动项目(参见图 14)。



Project	
🖃 📴 WorkSpace	
🗄 🔧 Project: Project_s	
🗄 쓚 Project: Project_ns	
	Set as Active Project

#### 2. 打开配置窗口以选择正确的器件:选择 Project / Options for Target (参见图 15)。

Options for Target 'STM32L562E-DK_s'         Device       Target       Output       Listing       User       C/C++ (AC6)       Asm       Linker       Debug       Utilities         Software Packs       Image: Software Pack       Image: Software Pack       Image: Software Pack       Image: Software Pack         Vendor:       STM32L562ZETx       Image: Software Pack       Pack:       Keil.STM32L50x_DFP.1.0.9         Device:       STM32L562QEIx       Image: Pack       Pack:       Image: Media Pack         Search:       Image: STM32L562QEIx       Image: STM32L562QEIx       Image: STM32L562QEIx       Image: STM32L562QEIx         Image: STM32L562QEIx       Image: STM32L562QEIx       Image: STM32L562QEIx       Image: STM32L562VETx       Image: STM32L562VETx         Image: STM32L562VETx       Image: STM32L562VETx       Image: STM32L562VETx       Image: STM32L562VETx       Image: STM32L562VETx         Image: STM32L562VETx       Image: STM32L562VETx       Image: STM32L562VETx       Image: STM32L562VETx       Image: STM32L562VETx       Image: STM32L562VETx         Image: STM32L562VETx       Image: STM32L562VETx       Image: STM32L562VETx       Image: STM32L562VETx       Image: STM32L562VETx       Image: STM32L562VETx       Image: STM32L562VETx       Image: STM32L562VETx       Image: STM32L562VETx       Image: STM32L562VETx       Image: STM		
Device       Target       Output       Listing       User       C/C++ (AC6)       Asm       Linker       Debug       Utilities         Software       Packs <ul> <li>Software</li> <li>Pack</li> <li>Software</li> <li>Pack</li> <li>Pack:</li> <li>Keil.STM32L562ZETx</li> <li>Pack:</li> <li>Keil.STM32L562ZETx</li> <li>URL:</li> <li>http://www.keil.com/pack</li> </ul> Search:         Image: STM32L562QEIx         Image: STM32L562QEIxQ           STM32L562QEIxQ         Image: STM32L562QEIxQ           STM32L562QEIxQ         Image: STM32L562QEIxQ               STM32L562VETx             Image: STM32L562VETx               STM32L562VETx             STM32L562VETxQ               STM32L562ZETx             STM32L562ZETx               STM32L562VETxQ             STM32L562ZETx               STM32L562ZETx             STM32L562ZETx               STM32L562ZETx             STM32L562ZETx               STM32L562ZETx             STM32L562ZETx               STM32L562ZETx             STM32L562ZETx               STM32L562ZETx             STM32L562ZETx               STM32L562ZETx             STM32L562ZETx               STM32L562ZETx </td <td>Options for Target 'STM32L5</td> <td>62E-DK_s'</td>	Options for Target 'STM32L5	62E-DK_s'
Software Packs         Vendor:       STMicroelectronics         Device:       STM32L562ZETx         Toolset:       ARM         Search:       Vendor:         STM32L562QEIx       Image: STM32L562QEIx         STM32L562VETx       Image: STM32L562VETx         STM32L562VETx       STM32L562VETx	Device Target Output Listing	User C/C++ (AC6) Asm Linker Debug Utilities
STM32L562RETx architecture with ARM TrustZone security.	Software Packs Vendor: STMicroelectronics Device: STM32L562ZETx Toolset: ARM Search: STM32L562QElx STM32L562QElxQ	Software Pack Pack: Keil.STM32L5xx_DFP.1.0.9 URL: http://www.keil.com/pack The ARM Cortex-M33 is the most configurable of all Cortex-M processors. It is a full featured microcontroller class processor based on the ARMv8-M mainline
	STM32L562RETx STM32L562VETx STM32L562VETxQ STM32L562VETxQ STM32L562ZETx STM32L562ZETx	architecture with ARM TrustZone security.

图 15. 设备选择

#### 3. 确保从 Project / Options for Target / Target 选择正确的内存区域:

- 启动地址 0: 位于 0x08040000 的闪存: 非安全闪存
- 启动地址 1:位于 0x20018000 的 SRAM:非安全 SRAM

软件模型必须在非安全模式下设置(参见图 16)。

#### 图 16. 内存配置

#### 🔣 Options for Target 'Project\_ns'

Device Target Output Listing User C/C++ (ACC	) Asm Linker Debug Utilities					
STMicroelectronics STM32L552ZE-Q	Code Generation					
Xtal (MHz): 12.0	Software Model: Non-Secure Mode					
Operating system: None						
System Viewer File:	✓ Use MicroLIB       Big Endian					
STM32L5x2.svd	Floating Point Hardware: Single Precision					
Use Custom File						
Read/Only Memory Areas	Read/Write Memory Areas					
default off-chip Start Size Start	tup default off-chip Start Size Nolnit					
ROM1:	RAM1:					
ROM2:	RAM2:					
ROM3: 0	RAM3:					
on-chip	on-chip					
IROM1: 0x8040000 0x40000 €	IRAM1: 0x20018000 0x18000 □					
□ IROM2: 0xC000000 0x80000 0	□ IRAM2: 0x3000000 0x30000 □					
ОК	Cancel Defaults Help					

 $\times$ 

添加来自安全项目的导入库:此文件在链接时自动包含在非安全项目中。它允许非安全部分调用安全部分的函数(参见图 17)。

☑ Options for Target 'Project_ns'       ×         Device Target Output Listing User C/C++ (AC6) Asm Linker Debug Utilities          ✓ Use Memory Layout from Target Dialog       X/O Base:         Make RW Sections Position Independent       R/O Base:         Make RO Sections Position Independent       R/W Base         Don't Search Standard Libraries       ✓         ✓ Report 'might fail' Conditions as Errors       disable Warnings:         Scatter File       \Project_ns.sct         ✓	图 17. 链接器选项				
Device       Target       Output       Listing       User       C/C++ (AC6)       Asm       Linker       Debug       Utilities         Image: Use Memory Layout from Target Dialog       X/O Base:       Image: Molecular Standard Dialog       X/O Base:       Image: Molecular Standard Dialog         Image: Make RO Sections Position Independent       R/O Base:       Image: Molecular Standard Dialog       Image: Molecular Standard Dialog         Image: Don't Search Standard Libraries       Image: Molecular Standard Dialog       Image: Molecular Standard Dialog       Image: Molecular Standard Dialog         Image: Report 'might fail' Conditions as Errors       disable Warnings:       Image: Molecular Standard Dialog         Scatter       \Project_ns.sct       Image: Molecular Standard Dialog       Image: Molecular Standard Dialog         Misc       /./Secure_nsclib/secure_nsclib.o       Image: Molecular Standard Dialog       Image: Molecular Standard Dialog         Unker       -cpu=Cortex-M33 *.o       Image: Molecular Standard Dialog       Image: Molecular Standard Dialog       Image: Molecular Standard Dialog         Unker       -cpu=Cortex-M33 *.o       Image: Molecular Standard Dialog       Image: Molecular Standard Dialog       Image: Molecular Standard Dialog         OK       Cancel       Defaults       Help	W Options for Target 'Project_ns'	X			
Use Memory Layout from Target Dialog       X/O Base:         Make RW Sections Position Independent       R/O Base:         Make RO Sections Position Independent       R/W Base         Don't Search Standard Libraries       Don't Search Standard Libraries         Report 'might fail' Conditions as Errors       disable Warnings:         Scatter      Project_ns.sct         File          Misc       //Secure_nsclib/secure_nsclib.o         Linker       -copu=Cortex-M33 *.o         controls      ibrary_type=microlib -strict -scatter ".\Objects\Project_ns.sct"         OK       Cancel       Defaults         Help	Device Target Output Listing User C/C++ (AC6) Asm	Linker Debug Utilities			
Misc controls       //Secure_nsclib/secure_nsclib.o         Linker control string       -cpu=Cortex-M33 *.o         -library_type=microlib -strictscatter ".\Objects\Project_ns.sct"         OK       Cancel       Defaults         Help	✓ Use Memory Layout from Target Dialog         Make RW Sections Position Independent         Make RO Sections Position Independent         Don't Search Standard Libraries         ✓ Report 'might fail' Conditions as Errors         Scatter File	X/O Base: 0x08040000 R/W Base 0x20180000 e Warnings:Edit			
String OK Cancel Defaults Help	Misc     .//Secure_nsclib/secure_nsclib.o       Linker    cpu=Cortex-M33 *.o       control    library type=microlibstrictscatter ".\Objects\Proje	ct ns.sct"			
	String OK Cancel	Defaults Help			

在分散加载文件部分下,检查该文件是否包含正确的地址,如图 17 中所示。 该文件由链接器使用,并决定内存布局的组织方式。分散加载文件示例在图 18 中给出。

#### 图 18. 分散加载文件示例

Project\_ns.sct 2 ; \*\*\* Scatter-Loading Description File generated by uVision \*\*\* 4 5 LR\_IROM1 0x08040000 0x00040000 { ; load region size\_region 6 ER\_IROM1 0x08040000 0x00040000 { ; load address = execution address 7 \*.o (RESET, +First) \*(InRoot\$\$Sections) 8 .ANY (+RO) 9 10 .ANY (+XO) 11 } RW\_IRAM1 0x20018000 0x00018000 { ; RW data 12 13 .ANY (+RW +ZI) 14 } 15 }



5. 从 Project / Options for Target / Debug 选择"ST-LINK debugger"(参见图 19)。

图 19.	调试设置
-------	------

Asm	Linker	Debug	Utilities		
•	Use: S	T-Link Del	bugger	-	Settings

- 6. 从 Debug settings / Flash Download 窗口(参见图 20)选择:
  - 下载功能:设置 Flash 操作
  - 用于算法的 RAM: 定义用于载入和执行编程算法的地址空间。通常,地址空间位于嵌入式 RAM 中。
  - 编程算法:包含编程 Flash 的定义。

#### 图 20. FlashLoader 配置

Cortex-M Target Driver Setup	×
Debug Trace Flash Download	
Download Function       C Erase Full Chip       ✓ Program         ⓒ Erase Sectors       ✓ Verify         ⓒ Do not Erase       ⊂ Reset and Run	
Description         Device Size         Device Type         Address Range           STM32L5x_512_NSecure_R         512k         On-chip Rash         08000000H - 0807FFFFH	
Start: 0x08000000 Size: 0x00080000	
Add Remove	
OK Cancel App	ly .

57

#### 9.2.1 编译项目

现在可以同时编译两个项目。从 Project / Batch Setup(项目/批设置)(参见图 21 和图 22)或从菜单栏中可用 的图标转到"批设置"菜单并选择这两个项目。

提示 必须首先编译安全项目,以便为非安全项目创建导入库。为了在编译非安全项目之前编译安全项目,它必须在编译顺序中处于首位。





# Batch Setup X Select Project Targets: Select Project\_s Project\_ns Clean Help Close

然后,从同一个菜单中单击"Batch Build (批编译)"来编译这两个项目(参见图 23)。

🧼 🏥 🛍	😻 🗸 🔜 L🎬 Project_ns	;
Project	😻 Batch Build	<b>д </b>
🖃 📅 WorkS	😻 Batch Rebuild	
🗄 🔧 Pro	Batch Clean	
🕂 🌱 🗘 Pro	🧼 Batch Setup	

#### 图 23. 在一个步骤中编译两个项目

#### 9.3 从安全代码执行到非安全代码

57

下载项目前,必须连接 STM32L562E-DK 探索板,连接方式如下:

1. 连接探索板上的 ST-LINKV3 编程和调试工具,方法是将 USB 电缆插入板件 CN17(ST-LINK USB 链接器)。ST-LINKV3 连接好之后,LD3 亮起为红色,如图 24 中所示。

图 24. STM32L562E-DK 探索板处于连接状态



2. 选择 Project\_ns 项目作为活动项目, 然后加载非安全二进制代码。选择 Project\_s 项目作为活动项目, 然后加载安全二进制代码。图 25 对此进行了说明。



#### 3. 单击工具栏中的"下载和调试"按钮来启动调试会话,如图 26 中所示。



提示

57/

如下所示,系统总是首先在安全代码(main.c)中启动,然后安全应用程序启动非安全应用程序。



图 27. Main.c 示例代码



#### 图 28. 代码切换到非安全代码状态



安全状态由 Keil<sup>®</sup>界面底部的状态栏提供,如图 29 中所示。

#### 图 29. CPU 状态

70   * 8; 71   */ 73 =/** 74   * 8; 75   */ 76   / 77 /***********************************			Į
¢			>
ST-Link Debugger	Debug: Secure CPU: Non-Secure	t1: 0.00000000 sec L:50 C:1	CAP NUM SCRL OVR R/W



# 10 将 EWARM 用于带 Trust Zone<sup>®</sup> 的 Cortex M33

面向 Arm<sup>®</sup>(EWARM)的最新版本 IAR 嵌入式工作台可用从 IAR System 的官方网站下载。 该部分使用 EWARM v8.40.1 和 STM32L562-DK 探索板。

#### 10.1 安全项目设置

要配置安全项目,第一步是打开"多项目"工作区文件: Project.eww,该文件允许用户同时处理两个项目。

1. 打开的项目出现在"Project Explorer (项目浏览器)"视图中,如图 30 中所示。

#### 图 30. EWARM v8.40.1 项目浏览器视图

Workspace	-	φ×
Project_s - STM32L562E-DK_Templates_TrustZone		~
Files	\$	•
🗆 🗂 Project		
- <mark> </mark>	<b>~</b>	
-⊞ ■ Doc		
🛛 🗕 🖬 Drivers		
Example		
U Gutput		
└───── ● Project_ns - STM32L562E-DK_Templates_TrustZone	~	
Example		
🗕 🖵 🔳 Output		
Overview Project_s Project_ns		

#### 2. 将 project\_s-STM32L562E-DK\_Templates\_TrustZone 设为活动项目,如图 31 中所示。

Workspace			-	φ×
Project_ns - STM32L562E-DK_Templates_	TrustZone			~
Files	Options		¢ •	•
Cxample Cxample Cutput Cap ● Project_ns - STM32L Cap ● Doc	Compile Rebuild All Clean		~	
-⊞ ■ Drivers -⊞ ■ Example -⊞ ■ Output	C-STAT Static Analysis Stop Build	>		
	Add	>		
	Remove Rename			
	Version Control System	>		
	Open Containing Folder File Properties			
	Set as Active			

#### 图 31. 将项目设为活动状态

3. 选择 Project-s / Options / General Options 以打开配置窗口,然后从"Processer Variant"部分选择正确的器件。 从"TrustZone"部分确保选择"安全"模式并勾选"TrustZone"复选框,如图 32 中所示。

		图 32. 设	备选择		
Options for node "Project_s"					>
Category: General Options Static Analysis Runtime Checking C/C++ Compiler Assembler Output Converter Custom Build Build Actions Linker Debugger Simulator CADI CMELS DAD	Library O Target Processor O Core O Device O CMSIS	ptions 2 Outpu variant	MISRA t Library C Cortex-M33 ST STM32L5620 None	A-C:2004 Configuration	MISRA-C:1998 Library Options 1
CMSIS DAP GDB Server I-jet J-Link/J-Trace TI Stellaris Nu-Link PE micro ST-LINK Third-Party Driver TI MSP-FET TI XDS	Endian mo Little Big BES BES DSP ES Advance	ade 32 8 <b>xtension</b> ced SIMD	Floating point FPU D registers	settings None - V TrustZo Mode Se	vne cure v
				Oł	< Cancel

 $\times$ 

#### 4. 从 Project-s / Options / Linker / Config"链接器配置文件编辑器"部分(参见图 33):

- a. 单击"编辑"以显示链接器配置文件编辑器。
- b. 检查链接器配置文件,以确保应用程序已链接到正确的地址:
  - 。 安全启动地址: Flash 位于 0x0C000000, 面向安全 Flash
  - 。 安全启动地址: SRAM1 位于 0x3000000, 面向安全 SRAM

#### 图 33. 链接器配置

Options for node "Project\_s"

Jalegoly.				Factory Settings
Seneral Options Static Analysis Runtime Checking	#define Diagnosti	cs Checksum	Encodings	Extra Options
Assembler	Config Library Inp	ut Optimizations	Advanced	Output List
Output Converter Custom Build Build Actions	Linker configuration Override default \$PROJ_DIR\$\s	file tm32l562xx flash s.ic	đ	
Debugger				
Simulator CADI	Linker configuration file	e editor		×
CMSIS DAP GDB Server	Vector Table Memory F	Regions Stack/Heap	Sizes	h
I-jet J-Link/J-Trace TI Stellaris	ROM	Start: 0x0C000000	End: 0x0C03Ff	-FF
Nu-Link PE micro	RAM	0x3000000	0x30017F	FF
SI-LINK Third-Party Driver TI MSP-FET			[	Save
TI XDS				

该.icf 文件包含链接器需要的所有信息。



5. 从 Project / Options / Debugger (项目/选项/调试器) 打开调试器选项卡。从设置部分,在驱动程序字段中选择 ST-LINK 作为调试器 (参见图 34)。

Options for node "Project_s" Category:
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         J-Link/J-Trace         TI Stellaris         Nu-Link         PE micro         ST-LINK         Device description file         Override default         \$TOOLKIT_DIR\$\CONFIG\debugger\ST\STM32L5x2.ddf

图 34. 项目调试器设置

#### 6. 从"下载"选项卡,确保"Use flash loader"已勾选(参见图 35)。



7. 安全项目必须将非安全项目输出文件指定为必须由调试器加载的额外映像。为此,请使用: Project / Options / Debugger / Images 并勾选"Download extra image"复选框(参见图 36)。

Category: General Options Static Analysis	Factory Settings
Runtime Checking	Cature Davidand Images Extra Oations Multicers Division
C/C++ Compiler	Setup Download images Extra Options Multicore Plugins
Output Converter	Download extra image
Custom Build	Path: \$PROJ_DIR\$\STM32L562E-DK_ns\Exe\project_ns.i
Build Actions	
Linker	
Simulator	Download extra image
CADI	Path:
CMSIS DAP	
GDB Server	Offset: Debug info only
I-jet	
J-Link/J-Trace	
Nulliok	Path:
PE micro	Offset: Debug info only
ST-LINK	
Third-Party Driver	
TI MSP-FET	
TI XDS	

#### 图 36. 选择非安全输出文件作为额外映像

调试信息会使调试器只下载调试信息,而不是完整的调试文件。

57

AN5421 - Rev 1

×

- 8. 从 Project / Options / ST-LINK "Setup"选项卡,参见图 37:
  - 选择"ST-LINK debugger"。
  - 选择复位类型:

51

- 。 系统复位:复位内核和外设。
- 内核复位:通过 VECTRESET 位复位内核;外设单元不受影响。
- 。 软件复位:设置 PC 为程序入口地址。
- 。 硬件复位:探针反转 JTAG 连接器上的 nSRST/nRESET 线来复位设备。这种复位通常也会复位 外设单元。
- 。 复位过程中的连接:在保持"Reset(复位)"激活的同时将 ST-LINK 连接到目标。复位被拉为低电 平,并在连接目标时保持低电平。
- 选择通信接口:
  - JTAG: 使用 JTAG 接口。
  - SWD:使用 SWO 接口,该接口使用的引脚数量少于 JTAG。如果要使用串行线输出(SWO)通 信信道,请选择 SWD。
- 选择接入端口:

Options for node "Project\_s"

- Auto(自动):自动将接入端口0应用于 Cortex<sup>®</sup>-M33。
- Manually (手动): 指定要使用的接入端口。

图 37.	项目	设置
-------	----	----

Category: Factory Settings General Options Static Analysis Runtime Checking Setup Communication Breakpoints C/C++ Compiler Assembler Emulator Output Converter ST-LINK/V3  $\sim$ Serial no: Custom Build **Build Actions** Shared mode Always prompt for probe selection Linker Debugger Reset Simulator Connect during reset  $\sim$ CADI CMSIS DAP Interface Access Port GDB Server Interface speed I-jet Auto ○ JTAG J-Link/J-Trace Default  $\sim$ O Specify SWD TI Stellaris Nu-Link PE micro ST-LINK Third-Party Driver TI MSP-FET TI XDS ΟK Cancel

 $\times$ 

#### 10.2 非安全项目设置

57

将 project\_s-STM32L562E-DK\_Templates\_TrustZone 设为活动项目

选择 Project-s / Options/ General Options (项目/选项/一般选项),从而打开配置窗口。在"目标"选项卡中,从处理器部分选择正确的器件(参见图 38)。

从 TrustZone<sup>®</sup>部分,确保选择"非安全"模式,并且勾选 TrustZone<sup>®</sup>复选框。

图 38. 项目设置:一般选项

Options for node "Project\_ns"

Catagory					
Lategory:					
General Options					
Static Analysis					
Runtime Checking					
C/C++ Compiler	Library Opt	ions 2	MISRA	-C:2004	MISRA-C:1998
Assembler	Target	Output	Library C	Configuration	Library Options 1
Output Converter					
Custom Build	Processor v	anant			
Build Actions	○ Core	Cort	ex-M33	$\sim$	
Linker		CT (		05.0	
Debugger	Device	513	5 TMISZEB020	46-04	l≣⊷
Simulator	OCHER	Non	e		
CADI	OCMSIS-	аск	-		_
CMSIS DAP	Endine and				
GDB Server	Endian mod	е н	oating point	settings	
I-jet	Little	F	PU	None	~
J-Link/J-Trace	Big				
TI Stellaris	O BE32	,    C	) registers	- ~	
Nu-Link	© DE02				
PE micro	U BEO				
ST-LINK				✓ TrustZo	ne
Third-Party Driver	DSP Exte	ension		Made N	
TI MSP-FET	Advance	d SIMD (NE	ON)	Mode Nor	n-secure V
TI XDS				L	
				ОК	Cancel
				01	Cancor

 $\times$ 

- 2. 从 Project-s / Options / Linker / Linker configuration (项目/选项/链接器/链接器配置) 文件部分(参见 图 39):
  - 单击"编辑"以显示链接器配置文件编辑器。
  - 检查链接器配置文件,以确保应用程序已链接到正确的地址:
    - 。 启动地址 0: 位于 0x08040000 的闪存(非安全闪存)
    - 。 启动地址 1: 位于 0x20018000 的 SRAM(非安全 SRAM)。

#### 图 39. 项目链接器配置

Options for node "Project\_ns"

57

Acgoly.					Factory 9	Setting
eneral Options						
atic Analysis						
untime Checking			~ I			•
C/C++ Compiler	#define D	liagnostics	Checksum	Encodings	Extra (	Options
Assembler	Config Librar	y Input	Optimizations	Advanced	Output	List
Dutput Converter	Linker config	juration file –				
Custom Build		default				
Build Actions	(ADDO)	DIDA\22	1500 de 1	·		
inker	SPROJ	_DIR\$\stm32	1062XX_flash_ns.	ICT		
Debugger	Lister and					~
Simulator	Linker config	uration file (	editor			
CADI		Memory De		-		
CMSIS DAP	Vector Table	Memory Re	gions Stack/He	ap Sizes		
GDD Server		S	tart:	End:		
1-jet 1-jet	ROM	Ĩ	0x08040000	0x080	7FFFF	
TI Stellaris						
Nulink	RAM		0x20018000	0x200	2FFFF	
PE micro						
ST-LINK						
Third-Party Driver					Sav	-
TI MSP-FET					341	
TI XDS						

57

3. 从"库"中的 Project-s / Options / Linker(项目/选项/链接器)(参见图 40)。 添加从安全项目导入的库。此文件在链接时自动包含在非安全项目中。它允许非安全部分调用安全部分的函 数。

图 40. 链接器库设置

Options for node "Project_ns"							×
Category:						Factory 9	Settings
General Options Static Analysis Runtime Checking							
C/C++ Compiler	#define	Diag	nostics	Checksum	Encodings	Extra (	Options
Assembler	Config	Library	Input	Optimizations	Advanced	Output	List
Output Converter Custom Build Build Actions	Autor	matic run al libraries	time library	v selection			
Linker	SPRO.I		Secure r	sclib/secure ns	clib o		
Debugger Simulator CADI	arnos	_DIN3/	Secure_1	SCIID (SECURE_IIS)		~	,
CMSIS DAP		ride defai	ult progran	n entrv			
I-iet	() E	ntrv svmb	ol	iar program st	art	1	
J-Link/J-Trace	ŎN	o entry s	/mbol				
TI Stellaris							
Nu-Link PE micro ST-LINK Third-Party Driver TI MSP-FET TI XDS							
					OK	0	ancel

4. 其他配置类似于安全项目。

#### 10.3 编译项目

两个项目都准备好进行编译。

1. 选择 Project / Batch Build (项目 / 批编译) 或菜单栏中可用的图标 (参见图 41)。

图 41. 项目批编译 Project - IAR Embedded Workbench IDE - Arm 8.40.1 File Edit View Project ST-Link Tools Window Help 한 🎦 🖬 🗿 Add Files... - < Add Group... Workspace Import File List... Project\_ns - STM32L Add Project Connection... Files Edit Configurations... 🖻 🛄 Project -🖯 🌒 Project\_s Remove - 🗉 蔰 Doc -🗉 🛋 Drivers Create New Project... -🕀 🛋 Examp Add Existing Project... – 🗉 🛋 Output 🕀 🌒 Project\_ stZone Options... ALT+F7 -🕀 🛑 Doc Version Control System ۲ -🕀 🛑 Drivers -🗄 🛋 Examp Make F7 -🕀 🛑 Output Compile CTRL+F7 Rebuild All Clean Batch build... F8 C-STAT Static Analysis ۲ Stop Build CTRL+ATTN 1.0

#### 2. 同时添加要编译的两个配置(参见图 42)。

必须首先编译安全项目,以便为非安全项目创建导入库。为了在编译非安全项目之前编译安全项目,它必须在编 译顺序中处于首位,如下所示。

- 151		
Edit Batch Build	× Edit Batch Build	×
Name build	Name build	
Available contigurations Project_ns - STM32L562E-DK_Templat Project_s - STM32L562E-DK_Templat >>>  Configurations to build >>> > ( (Drag to order)	Available configurations	Configurations to build Project_ns - STM32L562E-DK_Templat Project_s - STM32L562E-DK_Template Configurations to build
OK Cancel		OK Cancel

#### 图 42. 项目批编译顺序

提示

#### 10.4 从安全代码执行到非安全代码

为了执行任意代码, 它必须下载到板件, 过程如下:

- 1. 下载项目之前,按如下方式连接到 STM32L562E-DK 探索板(参见图 43):
  - 将 USB 电缆插入到探索板的 CN17 ST-LINK USB 连接器,从而将 ST-LINKV3 编程和调试工具连接到探 索板。
  - 当 ST-LINKV3 连接后,LD3 亮起为红色。

# 

#### 图 43. STM32L562E-DK 探索板处于连接状态

选择 Project\_ns 项目作为活动项目,然后加载非安全二进制代码。
 单击工具栏中的下载和调试按钮来启动调试会话,以便对闪存进行编程并开始调试(参见图 44)。

图 44. 下载和调试启动按钮



提示 当尝试加载非安全应用程序时,将显示以下警告信息。

#### 图 45. 加载错误警告信息示例的非安全应用程序

ebu	glog
	Log
	Tue Feb 11, 2020 16:09:31: Warning:
	Tue Feb 11, 2020 16:09:31: Verify error at address 0x080400CF, target byte: 0x00, byte in file: 0x08
	Tue Feb 11, 2020 16:09:31: Warning:
	Tue Feb 11, 2020 16:09:31: Verify error at address 0x080400D0, target byte: 0x00, byte in file: 0xAD
	Tue Feb 11, 2020 16:09:31: Warning:
	Tue Feb 11, 2020 16:09:31: Verify error at address 0x080400D1, target byte: 0x00, byte in file: 0x0F
	Tue Feb 11, 2020 16:09:31: Warning:
	Tue Feb 11, 2020 16:09:31: Verify error at address 0x080400D2, target byte: 0x00, byte in file: 0x04
	Tue Feb 11, 2020 16:09:31: Warning:
	Tue Feb 11, 2020 16:09:31: Verify error at address 0x080400D3, target byte: 0x00, byte in file: 0x08
	Tue Feb 11, 2020 16:09:31: Warning:
	Tue Feb 11, 2020 16:09:31: Verify error at address 0x080400D4, target byte: 0x00, byte in file: 0xB1
	Tue Feb 11, 2020 16:09:31: Warning:
	Tue Feb 11, 2020 16:09:31: Verify error at address 0x080400D5, target byte: 0x00, byte in file: 0x0F
	Tue Feb 11, 2020 16:09:31: Warning:
	Tue Feb 11, 2020 16:09:31: Verify error at address 0x080400D6, target byte: 0x00, byte in file: 0x04
	Tue Feb 11, 2020 16:09:31: Warning:
	Tue Feb 11, 2020 16:09:31: Verify error at address 0x080400D7, target byte: 0x00, byte in file: 0x08
	Tue Feb 11, 2020 16:09:31: Warning:
	Tue Feb 11, 2020 16:09:31: Verify error at address 0x080400D8, target byte: 0x00, byte in file: 0xB5
	Tue Feb 11, 2020 16:09:31: Warning: Too many verify errors, only the first 200 are displayed

这是正常现象,因为在验证阶段,调试器会尝试回读加载的内容,并将其与编译的二进制文件进行比较。 在 SAU配置之前,调试器在非安全区域(@ 0x08040000非安全闪存)生成安全传输。此访问被禁止,并且内容读取为零。

3. 选择 Project\_s 项目作为活动项目, 然后加载非安全二进制文件, 然后启动调试会话。

系统总是首先在安全代码(main.c)中启动,然后安全应用程序启动非安全应用程序

提示

#### 4. 安全状态由 CPU 寄存器下的安全寄存器提供(参见图 46)。

#### 图 46. 安全寄存器位置

Registers 1			▼ Ū X
Find:	~	Group: Current CPU Registers	~
Name	Value	Access	
RO	0x00000000	ReadWrite	
R1	0x00000000	ReadWrite	
R2	0x00000000	ReadWrite	
R3	0x00000000	ReadWrite	
R4	0x00000000	ReadWrite	
R5	0x00000000	ReadWrite	
R6	0x00000000	ReadWrite	
R7	0x00000000	ReadWrite	
R8	0xFFFFFFFF	ReadWrite	
R9	0xFFFFFFFF	ReadWrite	
R10	0xFFFFFFFF	ReadWrite	
R11	0xFFFFFFFF	ReadWrite	
R12	0xFFFFFFFF	ReadWrite	
SP	0x30000818	ReadWrite	
SPLIM	0x00000000	ReadWrite	
LR	0xFFFFFFFF	ReadWrite	
⊞ ¤PSR	0x01000000	ReadWrite	
∃ APSR	0x00000000	ReadWrite	
∃ IPSR	0x00000000	ReadWrite	
± EPSR	0x01000000	ReadWrite	
PC	0x0C000928	ReadWrite	
H PRIMASK	0x00000000	ReadWrite	
<b>BASEPRI</b>	0x00000000	ReadWrite	
H BASEPRI_MAX	0x00000000	ReadWrite	
H FAULTHASK	0x00000000	ReadWrite	
<b>E CONTROL</b>	0x00000000	ReadWrite	
I APSR	0x00000000	ReadWrite	
EAPSR	0x01000000	ReadWrite	
IEPSR	0x01000000	ReadWrite	
SECURE	0x00000001	ReadWrite	
CYCLECOUNTER	U SECURE		
CCTIMERI	U ReadWrite		
CCTIMER2	U Security stat	te la	
CCSTEP	1: Secure		
	Right-click f	or more registers and options	

0=非安全

1=安全



#### 10.5 当 RDP 设为 0.5 时,与 STM32L552ZE-Q 的连接问题

EWARM 能够连接到器件并调试非安全应用程序。要连接到 STM32L552ZE-Q,步骤如下:

- 1. 设置选项字节,如图 47 中所示:
  - TZEN = 1

57

- DBANK = 1
- SECWM2\_STRT = 0x1
- SECWM1\_PEND = 0x0.

#### 图 47. 使用 STM32CubeProgrammer v2.2.0 配置选项字节

Pro STN	//32CubeProgrammer				- 🗆 ×
STM32 Cube	Programmer			f 🕨	× 🖅
	Option bytes				Connected
	▼ Read Out Protection			ST-LINK	Disconnect
	Name	Value	Description		
<b>.</b>	RDP	BB 🔻	Read protection option byte The read protection is used to protect the software code stored in Flash memory. AA : Level 0, no protection ES - Level 0, and protection	ST-LI	NK configuration
OB			D3 : Ever I (J), read protection for active, only non-secure debug access is possible. Only available when in B8 : Ever I (J), read protection of memories CC : Ever I 2, chip protection	Port	SWD 🗸
	N BOR Level		>	Frequency (kHz)	24000 -
1	User Configuration			Mode	Under reset 👻
	▶ Secure Area 1			Access port	0 🔻
ł	► Write Protection 1			Reset mode	Hardware reset 🔹
	Secure Area 2			Shared	
	► Write Protection 2				
				External loader	-
			Apply Read	Target voltage	3.27 V

- 2. 加载非安全二进制(位于 0x08040000),然后加载安全二进制(位于 0x0C000000),如上节所述。
- 3. 使用 STM32CubeProgrammer 设置 RDP=0x55,以减少对非安全的调试(参见图 48)。

Prg STN	132CubeProgrammer				– 🗆 ×
STM32	Programmer			F 🕒 🦉	* 🛪 🏹
	Option bytes				Connected
	▼ Read Out Protection			ST-LINK	▼ Disconnect
	Name	Value	Description		
<b>.</b>	RDP	BB 👻	Read protection option byte The read protection is used to protect the software code stored in Flash memory. AA : Level 0, no protection S5: Level 0, read protection not active only non-secure debug access is possible. Only available when T	ST-LIN	IK configuration 002A001A
OB			Bs: Level 1, read protection of memories CC: Level 2, chip protection	Port	swD 🗸
	BOR Level			Frequency (KHz)	24000 👻
	<ul> <li>User Configuration</li> </ul>			Mode	Under reset 👻
	Secure Area 1			Access port	0 -
ł	Write Protection 1			Reset mode	Hardware reset 💌
2	Secure Area 2			Shared	Disabled
:	Write Protection 2			Eutomal loader	
			Apply Read	Target voltage	- 3.27 V

#### **48.** RDP=0.5

将复位模式改为软件复位:"设置"选项卡中的 Project options / ST-LINK,从"Setup"字段选择"Software", 4. 如图 49 中所示。

	图 49. 复位模式选择	
Options for node "Project_ns"		×
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 J-Link/J-Trace TI Stellaris Nu-Link PE micro ST-LINK Third-Party Driver TI MSP-FET TI XDS	Setup       Communication       Breakpoints         Emulator       Emulator         ST-LINK/V3       Serial no:         Shared mode       Always prompt for probe s         Shared mode       Always prompt for probe s         Reset       Software         Interface       Access Pot         JTAG       Default       Specify	election
	ОК	Cancel



#### 5. 在热插拔模式下连接器件: Project / Attach to the Running Target (参见图 50)。

提示 IDE 不支持在 RDP level 0.5 下进行非安全 flash 再编程,只有 STM32Cubeprogrammer 允许。

# 11 将 CubeIDE 用于带 Trust Zone<sup>®</sup>的 Cortex<sup>®</sup>-M33

本部分将在 CubelDE 中的 STM32 开发入门 (AN5394) 中说明,后者可从 www.st.com 上获取。

#### 版本历史

#### 表 1. 文档版本历史

日期	版本	变更
2020 年 2 月 21 日	1	初始版本。

# 57

# 目录

1	概述							
2	Arm®	Arm <sup>®</sup> Cortex <sup>®</sup> -M33 内核概述3						
3	Armv8-M 的 TrustZone <sup>®</sup> 概念							
4	SAU	/ IDAU - TrustZone <sup>®</sup> 概念						
5	调试	模式						
	5.1	侵入式调试						
	5.2	非侵入式调试						
6	调试	访问7						
	6.1	安全调试访问						
	6.2	非安全调试访问						
7	Flash	存储器保护						
	7.1	TrustZone <sup>®</sup> 被禁用后的读出保护级别 8						
	7.2	TrustZone <sup>®</sup> 被禁用后的 RDP 级别转换流程						
	7.3	TrustZone <sup>®</sup> 启用后的读出保护级别						
	7.4	当 TrustZone <sup>®</sup> 启用后, RDP 级别转换流程9						
8	从安	全/非安全项目开始						
9	将 M	DK-ARM 用于带 Trust Zone 的 Cortex <sup>®</sup> -M33						
	9.1	安全项目设置						
	9.2	非安全项目设置						
		<b>9.2.1</b> 编译项目						
	9.3	从安全代码执行到非安全代码 <b>23</b>						
10	将 E'	WARM 用于带 Trust Zone <sup>®</sup> 的 Cortex M33						
	10.1	安全项目设置						
	10.2	非安全项目设置						
	10.3	编译项目						
	10.4	从安全代码执行到非安全代码39						
	10.5	当 RDP 设为 0.5 时,与 STM32L552ZE-Q 的连接问题42						
11	将 C	ubeIDE 用于带 Trust Zone <sup>®</sup> 的 Cortex <sup>®</sup> -M33						

版本历史	 	

# 图一览

图	1.	<b>Armv8-M</b> 中的安全状态	. 4
图	2.	TrustZone <sup>®</sup> 被禁用(TZEN = 0)后的 RDP 级别转换流程	8
图	3.	TrustZone <sup>®</sup> 被禁用(TZFN = 1)后的 RDP 级别转换流程	9
图	4.	使用 STM32CubeProgrammer 配置洗项字节	10
图	5.	MDK-ARM 项目结构	11
<b>图</b>	6.	洗择安全项目....................................	11
图	7.	设备选择	12
图	8.	Project s目标洗项	13
图	9.	Project s 链接器配置	14
图	10.	分散加载文件示例	15
图	11.	目标选项调试	15
图	12.	调试配置	16
图	13.	闪存加载程序设置	17
图	14.	选择 Project ns 非安全项目	17
图	15.	设备选择	18
图	16.	内存配置	19
图	17.	链接器选项	20
图	18.	分散加载文件示例	20
图	19.	调试设置	21
图	20.	FlashLoader 配置	21
图	21.	项目批设置	22
图	22.	项目编译顺序	22
图	23.	在一个步骤中编译两个项目	22
图	24.	STM32L562E-DK 探索板处于连接状态	23
图	25.	加载非安全二进制代码	23
图	26.	下载和调试按钮	24
图	27.	Main.c 示例代码	24
图	28.	代码切换到非安全代码状态	25
图	29.	CPU 状态	25
图	30.	EWARM v8.40.1 项目浏览器视图	26
图	31.	将项目设为活动状态	27
图	32.	设备选择	28
图	33.	链接器配置	29
图	34.	项目调试器设置	30
图	35.	FlashLoader 选择	31
图	36.	选择非安全输出文件作为额外映像	32
图	37.	项目设置	33
图	38.	项目设置:一般选项	34
<b>图</b>	39.	项目链接器配置	35
图	40.	链接器库设置	36
图	41.	项目批编译	37
图	42.	项目批编译顺序	38
图	43.	STM32L562E-DK 探索板处于连接状态	39
图	44.	下载和调试启动按钮	40
图	45.	加载错误警告信息示例的非安全应用程序	40
图	46.	安全寄存器位置	41
图	47.	使用 STM32CubeProgrammer v2.2.0 配置选项字节	42
图	48.	RDP=0.5	42
图	49.	复位模式选择	43
图	50.	附加于运行中的目标选项	44



#### 重要通知 - 请仔细阅读

意法半导体公司及其子公司("意法半导体")保留随时对 ST 产品和/或本文档进行变更、更正、增强、修改和改进的权利, 恕不另行通知。买方在订货之前应获 取关于意法半导体产品的最新信息。意法半导体产品的销售依照订单确认时的相关 ST 销售条款。

买方自行负责对意法半导体产品的选择和使用,意法半导体概不承担与应用协助或买方产品设计相关的任何责任。

意法半导体不对任何知识产权进行任何明示或默示的授权或许可。

转售的意法半导体产品如有不同于此处提供的信息的规定,将导致意法半导体针对该产品授予的任何保证失效。

ST 和 ST 标志是意法半导体的商标。关于意法半导体商标的其他信息,请访问 www.st.com/trademarks。其他所有产品或服务名称是其各自所有者的财产。 本文档中的信息取代本文档所有早期版本中提供的信息。

#### © 2020 STMicroelectronics - 保留所有权利