Arm® TrustZone®特性面向 STM32L5 和 STM32U5 系列

引言

在 IoT（物联网）应用中，设备很容易受到通过互联网实施的恶意入侵。因此，保护设备和信息并使可信区域和不可信区域彼此隔离，这些安全主题就非常的重要。

STM32L5 和 STM32U5 系列器件（本文档的后面部分称其为 STM32L5、STM32U5 或 STM32L5/U5）基于高性能 Arm® Cortex®-M33 32 位 RISC 内核。该处理器使用 Armv8-M 架构，主要应用于非常注重安全性的环境。

面向 Armv8-M 的 Arm® TrustZone®技术是一种安全扩展，旨在将硬件划分为安全区域和非安全区域。通过 Arm® TrustZone®技术和软件方法，STM32L5/U5 微控制器（MCU）为安全应用程序提供良好的设计灵活性。

本文档介绍 Arm® TrustZone®技术和 STM32L5/U5 器件的各项特性，这些特性允许将 MCU 内存/资源划分为安全区域和非安全区域。

1 概述

本应用笔记适用于基于 Arm® Cortex®内核设备的 STM32L5 和 STM32U5 系列微控制器。

提示 *Arm 是 Arm Limited（或其子公司）在美国和/或其他地区的注册商标。*

arm

参考文档

[1] 参考手册：*基于 Arm®的 STM32L552xx 和 STM32L562xx 高级 32 位 MCU*（RM0438）

[2] 参考手册：*基于 Arm®的 STM32U575xx 和 STM32U585xx 高级 32 位 MCU*（RM0456）

[3] Armv8-M 架构参考手册可从 Arm®网站获得。

[4] 用户手册*采用 STM32L552ZE MCU 的评估板*（UM2597）

[5] 用户手册*采用 STM32U575AII6Q MCU 的评估板*（UM2854）

[6] 用户手册*采用 STM32L562QE MCU 的探索套件*（UM2617）

[7] 用户手册*面向物联网节点的探索套件采用 STM32U585AI*（UM2839）

[8] 用户手册：*STM32L5 Nucleo-144 板*（UM2581）

[9] 用户手册：*STM32U5 Nucleo-144 板*（UM2861）

2 Arm TrustZone 技术

2.1 概述

面向 Armv8-M 的 Arm TrustZone 技术将系统分为两个区域：一个是安全区域，另一个是非安全区域。

安全与非安全区域的划分是基于内存映射的。

所有可用的微控制器资源（包括 Flash 存储器、SRAM、外部存储器、外设和中断）被分配给安全区域或非安全区域。规划了这些资源的安全属性之后，非安全区域只能访问非安全内存和资源，而安全区域可以访问这两个区域中的所有内存和资源，包括安全和非安全资源。

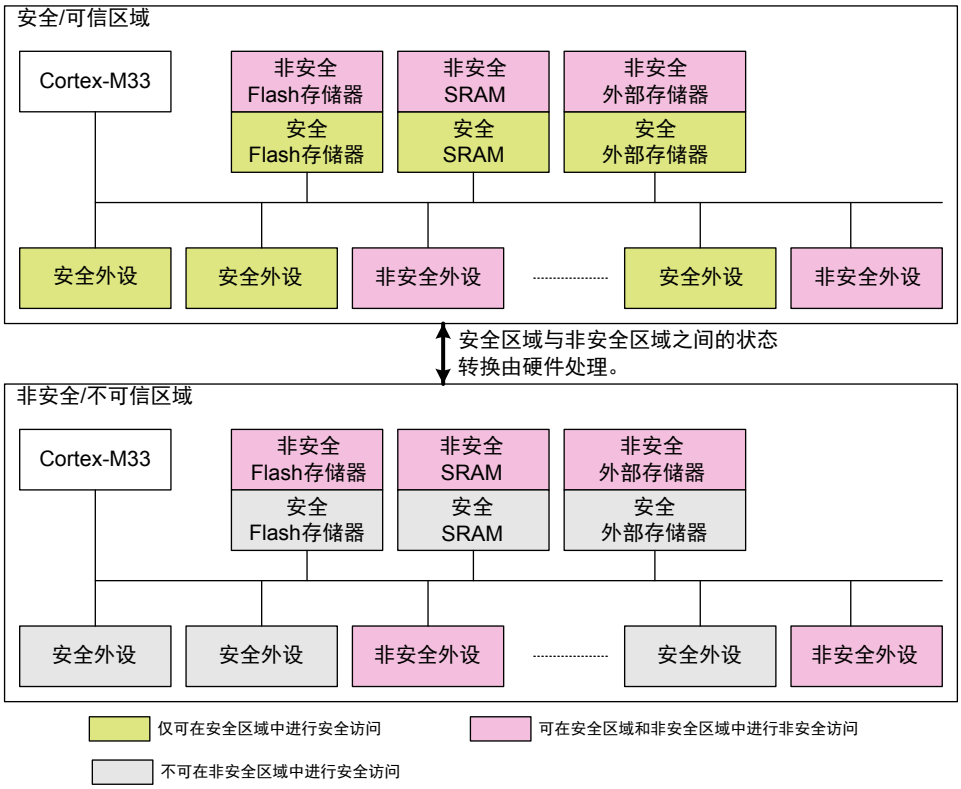
需要保护的重要数据（如加密密钥）必须在安全区域中以安全的方式进行存储和处理。

代码的执行位置定义了其类型：

- 如果代码在安全内存中执行，则称其为安全代码。
- 如果代码在非安全内存中执行，则称其为非安全代码。

安全代码和非安全代码在相同的 STM32L5/U5 器件上运行，如下图所示。

图 1. 在安全区域和非安全区域之间划分资源



2.2 安全状态

在简化视图中，执行的代码地址决定了 CPU 的安全状态，即安全或非安全：

- 如果 CPU 在非安全内存中运行代码，则 CPU 处于非安全状态。
- 如果 CPU 在安全内存中运行代码，则 CPU 处于安全状态。

Armv8-M 技术定义了以下地址安全属性：

- 安全

安全地址用于只能由安全代码或安全 master 访问的内存和外设。安全事务是那些由 master 发起并在安全状态下运行的事务。

- 非安全可调用（NSC）

NSC 是一种特殊类型的安全位置。这种类型的内存是 Armv8-M 处理器允许为其保留 SG（安全门）指令的唯一类型，该指令允许软件从非安全状态转为安全状态。该 SG 指令可用于防止非安全应用程序从分支进入无效入口点。

当非安全代码调用安全端函数时：

– API 中的第一条指令必须是 SG 指令。

– SG 指令必须在 NSC 区域内。

安全代码还提供非安全可调用函数，为非安全代码提供安全服务访问。

- 非安全

非安全地址用于设备上运行的所有软件均可访问的内存和外设。非-安全事务源自以非安全方式运行的 master 或访问非安全地址的安全 master（仅数据事务，非获取指令）。仅允许非安全事务访问非-安全地址。非安全事务不能访问安全地址。

3 在 STM32L5 和 STM32U5 系列器件上实现 TrustZone

3.1 激活 STM32L5 和 STM32U5 TrustZone

在 STM32L5/U5 中，TrustZone 是默认禁用的，可以通过在相应的选项字节中设置 TZEN 选项位进行启用。

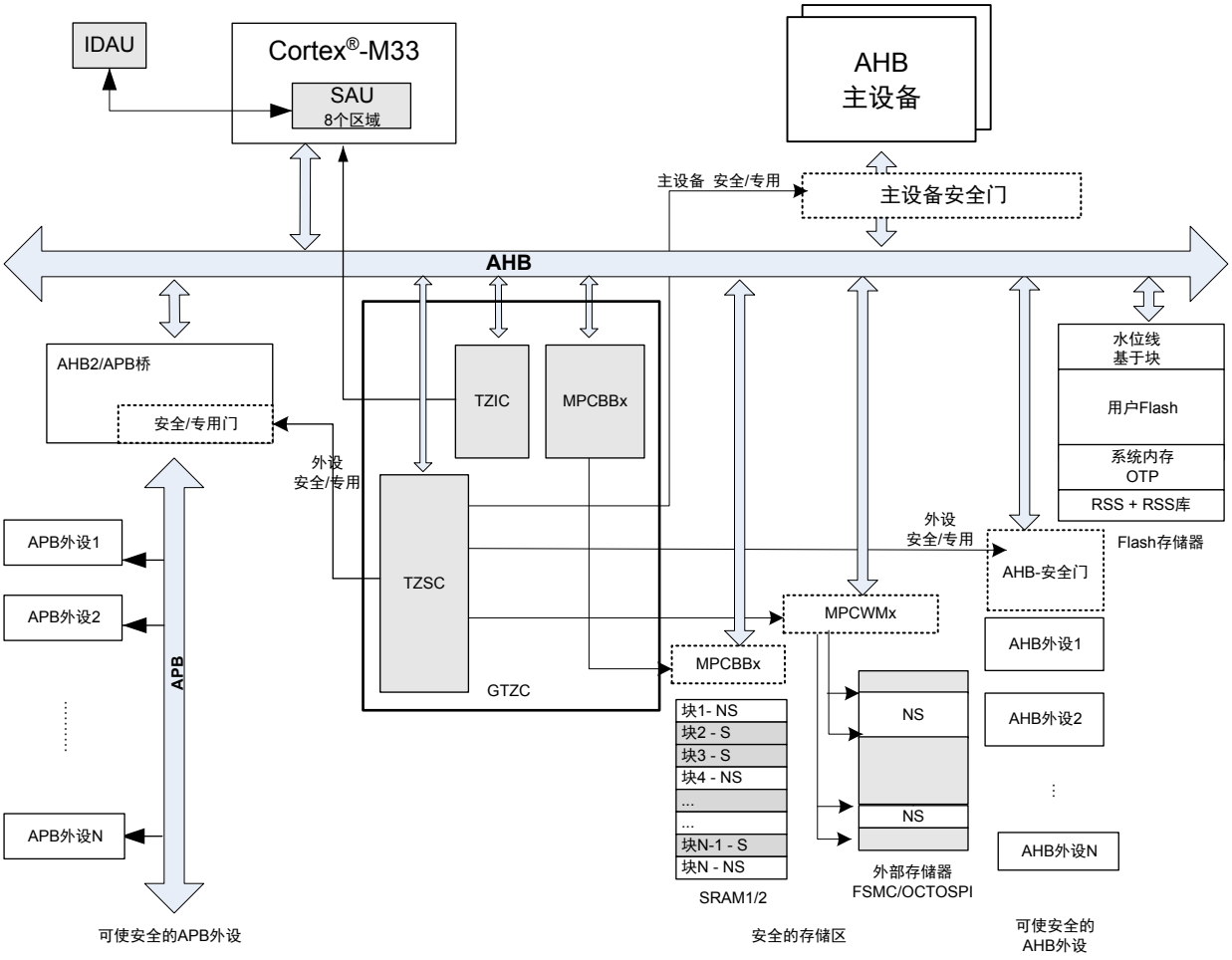
本文档中描述的所有特性适用于启用了 TrustZone 的 STM32L5/U5 器件。

3.2 TrustZone 框图

在 STM32L5/U5 中，TrustZone 的实现得益于 SAU（安全属性单元）、IDAU（实现定义的属性单元）、Flash 存储器和 GTZC（全局 TrustZone 安全控制器 ）。

下面的框图详细说明了 TrustZone 的实现过程。

图 2. STM32L5 和 STM32U5 TrustZone 实现概述



3.3 安全属性单元（SAU）和实现定义的属性单元（IDAU）

CPU 看到的内存地址安全状态由内部 SAU（安全属性单元）和 IDAU（实现定义的属性单元）组合控制。

安全属性的最终结果取决于 IDAU 和 SAU 其中较高的安全设置。安全属性的优先级如下：

- 安全具有最高的安全优先级。
- 非安全可调用的安全优先级次之。
- 非安全的安全优先级最低。

下表显示了如何将特定的安全属性（安全、非安全、激活非安全可调用）分配给特定地址。

表 1. 通过 IDAU 和 SAU 配置安全属性

IDAU 安全属性	SAU 安全属性(1)	最终安全属性
非安全	安全	安全
	安全 - NSC	安全 - NSC
	非安全	非安全
安全或 NSC(2)	安全	安全
	非安全	安全 - NSC

1. *定义的区域与 32 字节边界对齐。*
2. *NSC = 非安全可调用。*

3.3.1 STM32L5 和 STM32U5 的 IDAU 和存储器别名使用

STM32L5/U5 内存映射遵循 Arm 的建议实现一个重复的内存映射，一个用于安全视图，另一个用于非安全视图。

这意味着内存映射的每个区域（代码、SRAM、外设）被分成两个子区域，其中内部存储和外设被解码在两个单独的地址位置 - 非安全视图中和安全视图中。IDAU 的实现定义了这些区域的安全属性。

IDAU 内存映射分区不可配置。它由硬件确定。下表显示了 STM32L5/U5 IDAU 定义的内存映射安全属性分区。

表 2. STM32L5 和 STM32U5 器件上的 IDAU 内存映射地址安全属性

区域	地址范围	通过 IDAU 定义安全属性
重新映射时的代码 - 外部存储器	0x0000 0000-0x07FF FFFF（128 Mb）	非安全
代码 Flash 存储器和 SRAM	0x0800 0000-0x0BFF FFFF（64 Mb）	非安全
	0x0C00 0000-0x0FFF FFFF（256 Mb）	非安全可调用
重新映射时的代码 - 外部存储器	0x1000 0000-0x1FFF FFFF（256 Mb）	非安全
SRAM	0x2000 0000-0x2FFF FFFF（256 Mb）	非安全
	0x3000 0000-0x3FFF FFFF（256 Mb）	非安全可调用
外设	0x4000 0000-0x4FFF FFFF（256 Mb）	非安全
	0x5000 0000-0x5FFF FFFF（256 Mb）	非安全可调用
外部存储器(1)	0x6000 0000-0xDFFF FFFF（2 Gb）	非安全

1. *外部存储区域没有别名。*

3.3.2 STM32L5 和 STM32U5 SAU

STM32L5/U5 中有 8 个 SAU 区域。用户通过 SAU 修改所需的安全配置分区，如下表所示。启用 TrustZone 后，SAU 将所有地址默认为安全：所有内存区域均认为安全。

表 3. STM32L5 和 STM32U5 器件上的 SAU 内存映射地址安全属性

区域	地址范围	通过 IDAU 定义安全属性	通过 SAU 定义安全属性	最终安全属性
重新映射时的代码 - 外部存储器	0x0000 0000 - 0x07FF FFFF	非安全	安全 非安全或 非安全可调用	安全 非安全或 非安全可调用
代码 Flash 存储器和 SRAM	0x0800 0000 - 0x0BFF FFFF	非安全	非安全	非安全
	0x0C00 0000 - 0x0FFF FFFF	非安全可调用	安全或 非安全可调用	安全或 非安全可调用
重新映射时的代码 - 外部存储器	0x1000 0000 - 0x1FFF FFFF	非安全	非安全	非安全
SRAM	0x2000 0000 - 0x2FFF FFFF	非安全	非安全	非安全
	0x3000 0000 - 0x3FFF FFFF	非安全可调用	安全或 非安全可调用	安全或 非安全可调用
外设	0x4000 0000 - 0x4FFF FFFF	非安全	非安全	非安全
	0x5000 0000 - 0x5FFF FFFF	非安全可调用	安全或 非安全可调用	安全或 非安全可调用
外部存储器	0x6000 0000 - 0xDFFF FFFF	非安全	安全 非安全或 非安全可调用	安全 非安全或 非安全可调用

示例

外设在两个地址范围被解码：非安全视图中的 0x4000 0000 和安全视图中的 0x5000 0000。

根据 SAU 和 IDAU 的编程，安全代码通过生成安全事务来访问安全视图中的外设，而非安全代码在非安全视图中的另一个地址访问相同的外设。根据 GTZC/TZSC 如何定义外设安全属性，授权或拒绝访问。详情请参见第 4 节 和第 5 节 。

STM32CubeL5 和 STM32CubeU5 中的 SAU 配置

SAU 区域的定义在以下 CMSIS 文件中进行：

- STM32U5：`Device partition_stm32U575xx.h` 和 `partition_stm32U585xx.h`
- STM32L5：`Device partition_stm32l552xx.h` 和 `partition_stm32l562xx.h`

安全项目启用 SAU 并定义 SAU 区域。STM32CubeL5 和 STM32CubeU5 定义下表中列出的默认 SAU 区域（与链接器内存布局文件模板相关联）。

表 4. STM32CubeL5 和 STM32CubeU5 默认 SAU 区域

SAU 区域	STM32L5 地址	STM32U5 地址	STM32Cube SAU
SAU 区域 0	0x0C03 E000 - 0x0C03 FFFF	0x0C0F E000 - 0x0C0F FFFF	安全、非安全可调用
SAU 区域 1	0x0804 0000 - 0x0807 FFFF（256-Kbyte Flash 存储区 2）	0x0810 0000 - 0x081F FFFF（1 Mb Flash 存储区 2）	非安全
SAU 区域 2	0x2001 8000-0x2003 FFFF（SRAM, 160-Kbyte，SRAM1 的后一半 + SRAM2）	0x2004 0000 - 0x200B FFFF (SRAM3)	
SAU 区域 3	0x4000 0000 - 0x4FFF FFFF（外设映射存储器）		
SAU 区域 4	0x6000 0000 - 0x9FFF FFFF（外部存储器）		
SAU 区域 5	0x0BF9 0000 - 0x0BFA 8FFF（系统存储器）		
SAU 区域 6	未使用		
SAU 区域 7			

未被 SAU 区域覆盖的 0x0000 0000-0xDFFF FFFF 中的所有内存空间均处于安全状态。
IDAU 提供的安全属性和 SAU 提供的安全属性组合的结果如下表所示。

表 5. STM32CubeL5 内存安全分区

区域	地址范围	通过 IDAU 定义安全属性	通过 SAU 定义安全属性	最终安全属性
Flash 存储器	0x0804 0000 - 0x0807 FFFF	非安全	非安全	非安全
	0x0C00 0000 - 0x0C03 DFFF	非安全可调用	安全	安全
	0x0C03 E000 - 0x0C03 FFFF	非安全可调用	非安全可调用	非安全可调用
SRAM1	0x3000 0000 - 0x3001 7FFF	非安全可调用	安全	安全
	0x2001 8000 - 0x2002 FFFF	非安全	非安全	非安全
SRAM2	0x2003 0000 - 0x2003 FFFF	非安全	非安全	非安全
外设	0x4000 0000 - 0x4FFF FFFF	非安全	非安全	非安全
	0x5000 0000 - 0x5FFF FFFF	非安全可调用	安全	安全
外部存储器	0x6000 0000 - 0x9FFF FFFF	非安全	非安全	非安全

表 6. **STM32CubeU5 内存安全分区**

区域	地址范围	通过 IDAU 定义安全属性	通过 SAU 定义安全属性	最终安全属性
Flash 存储器	0x0810 0000 - 0x081F FFFF	非安全	非安全	非安全
	0x0C00 0000 - 0x0C0F DFFF	非安全可调用	安全	安全
	0x0C0F E000 - 0x0C0F FFFF	非安全可调用	非安全可调用	非安全可调用
SRAM1	0x3000 0000 - 0x3002 7FFF	非安全可调用	安全	安全
SRAM2	0x3003 0000 - 0x3003 FFFF	非安全可调用	安全	安全
SRAM3	0x2004 0000 - 0x200B FFFF	非安全	非安全	非安全
SRAM4	0x2800 0000-0x2800 3FFF	非安全	非安全	非安全
外设	0x4000 0000 - 0x4FFF FFFF	非安全	非安全	非安全
	0x5000 0000 - 0x5FFF FFFF	非安全可调用	安全	安全
外部存储器	0x6000 0000 - 0x9FFF FFFF	非安全	非安全	非安全

当然，这只是一个例子。用户必须根据应用程序在安全和非安全资源方面的需求调整内存分区。

4 STM32L5 和 STM32U5 系列的安全配置

SAU/IDAU 设置只适用于一个 master：CPU。其他 master（如 DMA）看不到这些策略。这就是外设端需要一个本地安全门的原因。

除了 Cortex-M33 TrustZone 特性，STM32L5/U5 器件还具有一些补充性安全特性，通过在 SAU/IDAU 之上提供第二级安全特性，加强并允许更灵活地划分安全和非安全区域。

4.1 Flash 存储器的安全配置

由于采用了非易失性 Flash 安全水位线和基于易失性块的 Flash 接口寄存器，即使从IDAU/SAU看来不具备安全性，也可以将 Flash 存储器区域配置为安全的。

SAU 和 IDAU 负责授权 CPU 发布的事务，并将 CPU 对互连的访问标记为安全或非安全。Flash 存储器安全水位线和基于块-的寄存器授权来自 CPU/Cortex-M33 和其他主机的事务：

- STM32L5 主机：DMA1、DMA2，以及 SDMMC
- STM32U5 主机：GPDMA1（配备两个主端口的通用 DMA）、DMA2D、SDMMC1，以及 SDMMC2

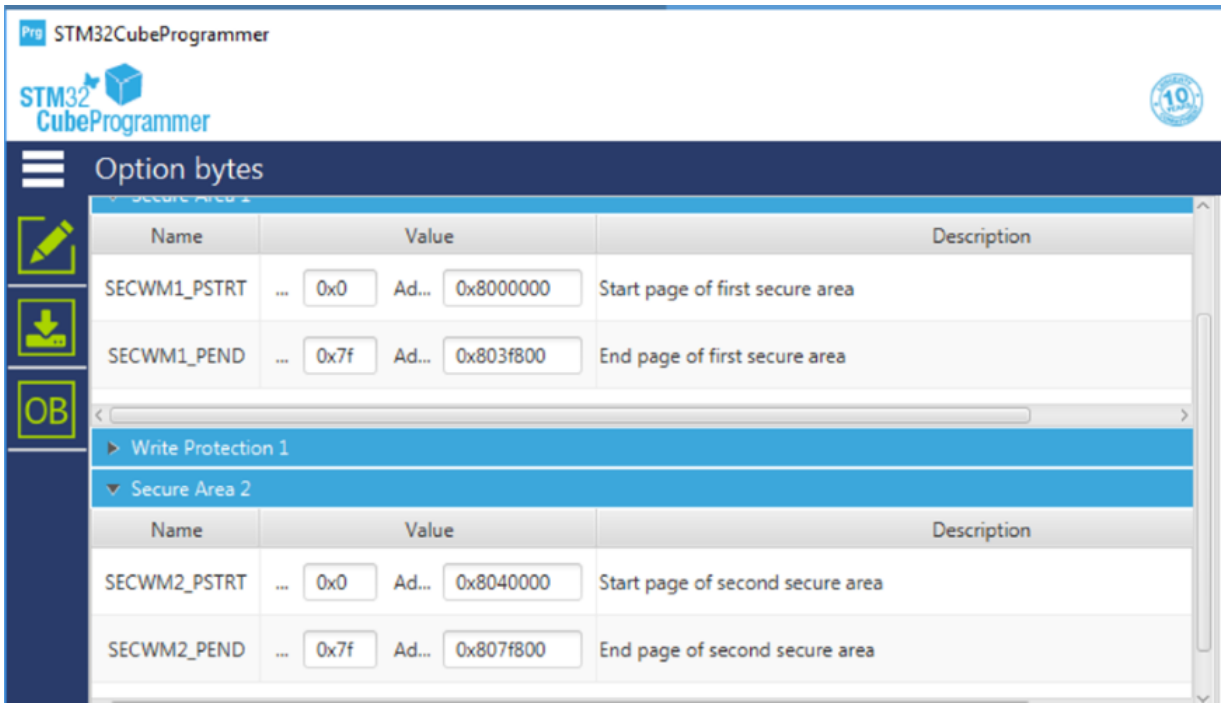
如图 2 所示，由以 Flash 存储器为目标的 Cortex-M33 发布的每个事务首先由 IDAU/SAU 检查，然后由 Flash 安全水位线或基于块的寄存器检查。更多详细信息，请参见图 5。

4.1.1 Flash 存储器的安全水位线

选项字节最多定义两个不同的非易失性安全区域，仅由安全访问进行读写：SECWMx_PSTRT 和 SECWMx_PEND（x = 1，2）。

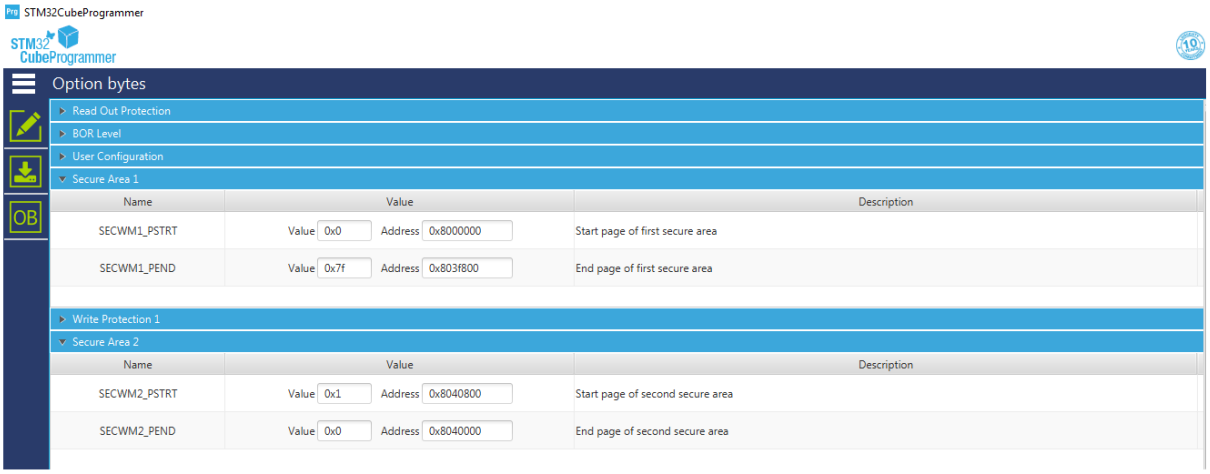
下图显示了设置 TZEN（整个 Flash 存储器为安全）。

图 3. 默认的 Flash 存储器状态，与设置 TZEN 之后的选项字节无关



STM32CubeL5 和 STM32CubeU5 TrustZone 示例假设存储区 1 为安全，而 Bank2 为非-安全。

图 4. 默认的 Flash 存储区安全状态（由 STM32Cube 定义），与选项字节无关



4.1.2 基于 Flash 存储块-的特性

即使整个 Flash 存储器（通过 IDAU/SAU 和 Flash 安全水位线选项字节）都是非安全的，也可以使用基于 Flash 存储块-的特性配置临时安全区域。

使用基于 Flash 接口块-的配置寄存器，任何页面都可配置为安全或非安全模式。

当通过 Flash 安全水位线选项字节将其设置为不安全时，基于块的寄存器可以将页面设置为安全。反之则无法进行：通过 Flash 安全水位线选项字节将页面配置为安全时，不可能使用基于块的寄存器将页面配置为非安全。

4.2 全局 TrustZone 控制器（GTZC）

GTZC 具有以下子块：

- TZSC（TrustZone 安全控制器）允许对以下要素配置安全属性：
  - 外设（参见下面的说明），安全或非安全
  - 外部存储器：通过水位线-存储-保护控制器（MPCWMx，x = 1、2、3）
- MPCBBx（基于块的存储器保护控制器）允许 SRAM 块的安全属性配置如下：
  - STM32L5：可以使用 MPCBB 并基于块-将 SRAM1 和 SRAM2 配置为安全或非安全。基于安全块-的 SRAM 粒度是一个 256 字节的页面。
  - STM32U5：可以使用 MPCBBx 并基于块将 SRAM1、SRAM2、 SRAM3、SRAM4 设置为安全或非安全。基于安全块-的 SRAM 粒度是一个 512 字节的页面。
- TZIC（TrustZone 非法存取控制器）收集系统中所有非法存取事件，并向 NVIC 生成安全中断（GTZC_IRQn）。

提示

当 TrustZone 安全特性激活后，外设被设为“securable”或“TrustZone-aware”：

- *securable：安全属性由 GTZC/TZSC 控制器配置。*
- *TrustZone-aware：使用某些外设安全寄存器配置安全属性。例如，GPIO 是 TrustZone-aware，安全属性是通过 GPIOx_SECCFGR 安全寄存器配置的。*

如需“securable”和“TrustZone-aware”外设列表，请参见文档[1]或[2]中的“TrustZone 外设分类”一节。

5 总体系统安全访问规则

5.1 默认安全状态

当使用 FLASH_OPTR 中的 TZEN 选项位激活 TrustZone 安全特性后，系统默认安全状态如下：

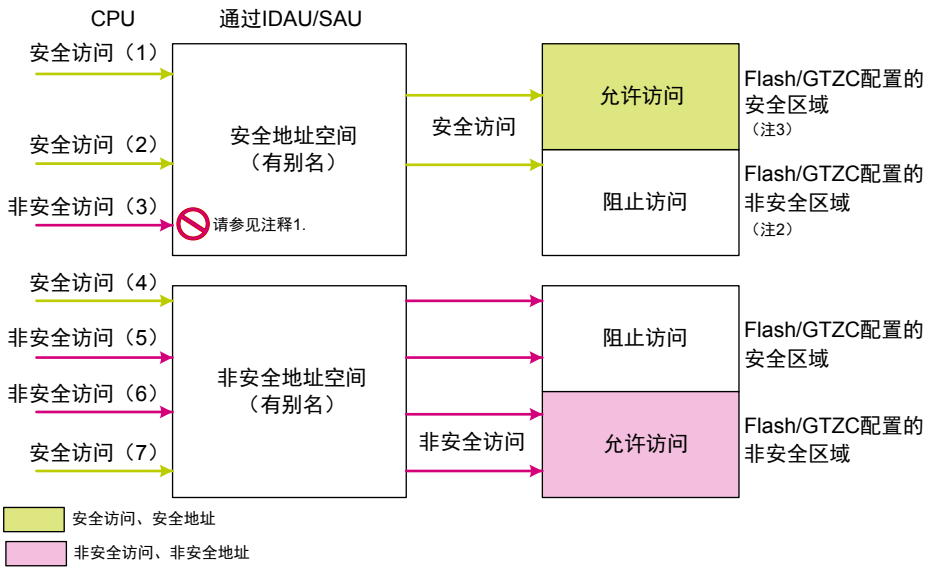
- Cortex-M33 CPU 在重置后处于安全状态。启动地址必须指向一个安全内存区域。
- 所有中断被分配到安全中断控制器。
- 通过 IDAU/SAU，所有内存映射都是完全安全的，直到安全代码启用 SAU 并为非安全资源定义了区域。
- 整个 Flash 存储器均处于安全状态。安全区域由水位线用户选项字节定义，其出厂设置为：
  - SECWMx_PSTRT = 0x00
  - SECWMx_PEND (x = 1,2) = 0x7F
- 所有 SRAM 都是安全的。
- 外部存储器：FSMC 和 OCTOSPIx 存储区处于安全状态。
- 对于 STM32U5，备份 SRAM 处于安全状态。
- 所有外设（除了 GPIO）都是非安全的。
- 所有 GPIO 都是安全的。
- 所有 DMA 通道都是非安全的。
- 备份寄存器都是非安全的。

5.2 内存和外设安全访问规则

CPU 发出的任何事务首先由 SAU 过滤，然后由实现在目标外设（Flash 存储器、SRAM、外部存储器或任何安全外设）附近的安全门过滤。

下图描述了如何根据事务安全属性进行事务过滤。

图 5. 内存和外设数据存取规则汇总



注意：1. 访问被SAU/IDAU阻止，导致安全故障。
2. 安全事务也可以访问外设的非安全寄存器。这与内存访问规则不同。
另外，设置了GTZC_MPCBBx_CR寄存器中的SRWILADIS位之后，允许对非安全SRAM块进行安全读/写访问。
3. “通过Flash / GTZC的安全/非安全属性”指通过Flash安全水位线选项字节、基于Flash存储器块的寄存器、MPCWMx、MPCBB和TZSC_SECCFGRx寄存器确定的安全属性。

事务带有自己的安全属性。根据这些属性，由 SAU 和 Flash 存储器或 GTZC（面向 SRAM、外部存储器和外设）决定是否授予访问权限。

提示

仅对于 STM32L5 而言，非安全信息块只能由非安全事务访问。信息块是由选项字节、内存保护用户配置、系统内存和 OTP（一次性可编程）区域组成的 Flash 存储区域。特别是 OTP 区域、VREFINT 和温度传感器校准值只能由非安全事务访问。安全应用程序必须规划一个 SAU 区域，并将其配置为非-安全区域。

访问规则如下所列：

- 如果某个地址在 SAU/IDAU 和 Flash/GTZC看来都是安全属性，并对其进行安全访问：则访问是允许的。参见图 5 中的(1)。
- 如果某个地址在SAU/IDAU看来是安全属性，Flash/GTZC看来是非安全属性，并对其进行安全访问，则阻止该访问。参见图 5 中的(2)。
- 通过 SAU/IDAU 对某个安全地址进行非安全访问：无论通过 Flash /GTZC 访问时该地址具有何种安全属性，都应阻止该访问，同时触发 Cortex-M33 安全故障异常。参见图 5 中的 (3)。
- 如果某个地址在SAU/IDAU看来是非安全属性，Flash/GTZC看来是安全属性，并对其进行安全访问：则访问被阻止。参见图 5 中的(4)。
- 如果某个地址在SAU/IDAU看来是非安全属性，Flash/GTZC看来是安全属性，并对其进行非安全访问：则访问被阻止。参见图 5 中的(5)。
- 如果某个地址在SAU/IDAU看来是非安全属性，Flash/GTZC看来是非安全属性，并对其进行非安全访问：则访问被允许。参见图 5 中的(6)。
- 如果某个地址在SAU/IDAU看来是非安全属性，Flash/GTZC看来是非安全属性，并对其进行安全访问：则访问被允许。参见图 5 中的(7)。

访问被阻止后，结果为如下之一：

- RAZ/WI（零读出/忽略写入操作）
- RAZ/WI 和非法访问事件/中断
- 总线错误

例如，对安全 Flash 存储器区域的非安全访问结果是 RAZ/WI，并生成非法访问事件。如果 GTZC_TZIC_IER2（面向 STM32L5）和 GTZC_TZIC_IER4（面向 STM32U5）中的 FLASHIE 启用了非法访问中断，则会产生非法访问中断。

更多详细信息，请参见文档[1]或[2]。

对于指令获取，SAU 的事务输出（安全或非安全）取决于独立于 CPU 状态的目标地址。

表 7. 指令获取规则

CPU 状态	通过 IDAU/SAU 访问时的目标内存地址安全属性	事务
安全或非安全	非安全	非安全
安全或非安全	非安全可调用	安全
安全或非安全	安全	安全

6 启动和根安全服务（RSS）

RSS 嵌入在安全信息块（安全 Flash 存储器区域的一部分）中，并在意法半导体生产期间进行编程。更多详细信息，请参见文档[1]或[2]。

例如，因为有 RSS 扩展固件（RSSe SFI）， RSS 支持安全固件安装（SFI）。当生产被分包给不受信任的第三方时，该特性允许客户保护烧写到 STM32 器件的固件的机密性。参照应用说明*安全固件安装（SFI）概述*（AN4992）获取详细信息。

启动内存地址是通过 SECBOOTADD0[24:0]选项字节设置的。然而，允许的地址空间取决于 Flash 存储器的读出保护（RDP）等级。当 RDP 级别为 0.5 或更高时，如果设置的启动内存地址超出了允许的内存映射区域，则在安全系统 Flash 存储器中强制使用默认启动获取地址。

表 8. 启动空间与 RDP 保护

RDP 级别	启动地址
0	任意启动地址
0.5	仅 RSS 或安全 Flash 存储器
1	
2	

当通过设置 TZEN 选项位启用 TrustZone 时，启动空间必须位于安全区域。SECBOOTADD0[24:0]选项字节用于选择启动安全内存地址。为了提高安全性并建立信任根（RoT），无论采用任何其他启动选项，都必须选择唯一的启动入口选项。这是通过在 FLASH_SECBOOTADD0R 寄存器中设置 BOOT_LOCK 选项位来实现的。该位只能通过安全访问进行设置。

注意：对于 STM32L5，BOOT_LOCK 选项位一经设置便无法清除。唯一的启动入口地址是在 SECBOOTADD0[24:0]选项字节中设置的地址。对于 STM32U5，当 RDP 级别为 0 时，可以清除 BOOT_LOCK。

7 TrustZone 启用后的读取保护（RDP）

TrustZone 启用（TZEN = 1）之后，有四种 RDP 级别（从无保护的级别 0 到最大保护且无调试的级别 2），详情请见下表。

表 9. RDP 保护级别（TrustZone 已启用）

RDP 字节值	RDP 级别
0xAA	0
0x55	0.5
除 0x55、0xAA 或 0xCC 外的任意值	1
0xCC	2

7.1 RDP 级别 1

当 RDP 级别为 1 时，不能访问 Flash 主存储、备份寄存器、备份 RAM（仅 STM32U5）、OTFDEC 区域（可用时）、ICACHE、DCACHE，以及 SRAM。当 CPU 处于安全状态时，如果进行调试访问，则会检测到入侵。

当 CPU 处于不安全状态时，可以通过 JTAG/SWD 连接到目标机以及进行 RDP 回退。除了停止 CPU，任何调试访问均视为入侵。

必须以下述方式中的一种实现 RDP 回归：

- 通过bootloader：必须从 RSS 启动。
  注：因为从 RSS 启动，也可以通过 JTAG/SWD 进行回归。
- 通过 JTAG/SWD，从用户 Flash 存储器启动：CPU 必须处于非安全状态，才能连接到目标。
  在将 RDP 级别设为 1 之前，用户必须确保安全应用程序调用非安全应用程序（以便能够连接到目标），并且能够从 RDP 级别 1 进行回归。

注意: 如果没有非安全代码，则 CPU 始终保持安全状态，且不能以“从用户 Flash 存储器启动”的方式通过 JTAG/SWD实现 RDP 回退。在这种情况下，只能以从 RSS 启动的方式通过 JTAG/SWD/bootloader实现回归。详细信息请参见文档[1]或[2]的“启动配置”一节。在 STM32U5 中，RDP 级别 1 回归可以通过在较低的 RDP 级别下提供的OEM1 和 OEM2 密钥来保护（更多细节请参阅文档[2]）。

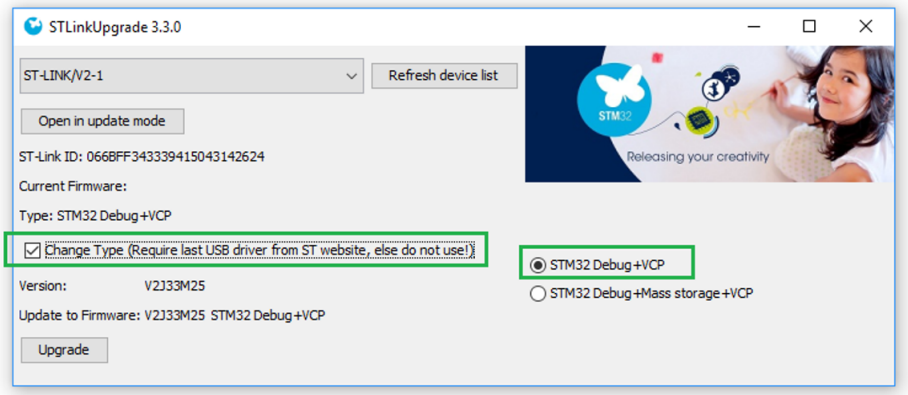
意法半导体的 STM32L5/U5 系列板件（Nucleo 板、评估板和探索板）配备集成式 ST-LINK，可以同时用作电源和进行调试。由于 ST-LINK 的大容量存储接口需要在 ST-LINK 启动时进行目标识别（SWD 连接），所以每次插入 ST-LINK USB 电缆时都会检测到一个调试入侵。用户应用程序永不执行，而 CPU 始终保持 LOCKUP 状态，在发生入侵时不可能执行代码。然后，无法连接目标。

必须使用以下解决方案确保用户应用程序被执行，并通过 JTAG/SWD 连接到目标：

- 仅使用 ST-LINK 进行调试：必须使用另一个电源。为了执行用户应用程序，必须在 ST-LINK USB 电缆已经插入的情况下关闭/打开电源。
- 通过 STLinkUpgrade 应用程序更改固件类型可以禁用 ST-LINK 的大容量存储接口，如图 6 所示。

提示 *只有在 ST-Link/V2 中才能禁用大容量存储。*

图 6. 禁用 ST-LINK 的大容量存储接口



7.2 RDP 级别 0.5

当 RDP 级别为 0.5 时，不能访问安全 Flash 存储、安全备份寄存器、备份 RAM（仅 STM32U5）、OTFDEC 区域（可用时）、ICACHE、DCACHE，以及 SRAM 区域。非安全 Flash 存储器、非安全备份寄存器，以及非安全 SRAM 区域仍然可以访问。

当 CPU 处于安全状态时，不可能通过 JTAG/SWD 连接到目标。

当 CPU 处于不安全状态时，可以通过 JTAG/SWD 连接到目标机以及进行 RDP 回退。

提示 *在 STM32U5 中，如果 RDP 级别为 0.5，则无法请求 RDP 级别 0。需要将 RDP 级别提高到 1，然后回归到 0。*

可使用下述方式中的一种实现 RDP 回归：

- 通过bootloader：必须从 RSS 启动。
  注：因为从 RSS 启动，也可以通过 JTAG/SWD 进行回归。
- 通过 JTAG/SWD 进行回归，从用户 Flash 存储器启动：为了能够连接到目标，CPU 必须处于非安全状态，因为在 RDP 级别 0.5 下禁止安全调试。只有当 CPU 处于非安全状态时，才可能连接到目标。
  在将 RDP 级别设为 0.5 之前，用户必须始终确保安全应用程序调用非安全应用程序（以便能够连接到目标），并且能够从 RDP 级别 0.5 进行回归。

注意: 如果没有非安全代码，则 CPU 始终保持安全状态，且不能以“从用户 Flash 存储器启动”的方式通过 JTAG/SWD实现 RDP 回退。在这种情况下，只能以从 RSS 启动的方式通过 JTAG/SWD/bootloader实现回归。详细信息请参见文档[1]或[2]的“启动配置”一节。

如需详细了解不同读保护级别和访问状态，以及 TZEN = 1 时的保护级别和执行模式，请参考文档[1]或[2]。

提示 *在 RDP 级别 1 和 0.5 时，如果使用 STM32CubeProgrammer，则须在“热插拔”模式下连接到目标，以便在连接到目标期间保持用户应用程序被执行，并避免在 CPU 处于重置状态时进行连接（意味着 CPU 是安全的）。*

注意: 当满足以下条件时，不能进行 RDP 回退：

- BOOT_LOCK 选项位已设置。
- SECBOOTADD0[24:0]是安全用户 Flash 存储器中的地址。
- 不存在非安全代码。CPU 始终保持安全状态，并且不能在 RDP 级别 0.5 下编程非安全 Flash 存储器。

7.3 RDP 级别 2

当 RDP 级别设为 2 时，可以保证保护级别 1。从 SRAM 启动（启动 RAM 模式）和从系统内存启动（bootloader模式）不再可用。只能从主 Flash 存储器或 RSS 启动。

当从主 Flash 存储器或 RSS 启动时，所有操作都允许在主 Flash 存储器上进行。允许通过用户代码对 Flash 存储器和 SRAM 进行读取、擦除和设置访问。

仅对于 STM32U5 而言，除非已提供 OEM2 密钥，否则以下特性适用：

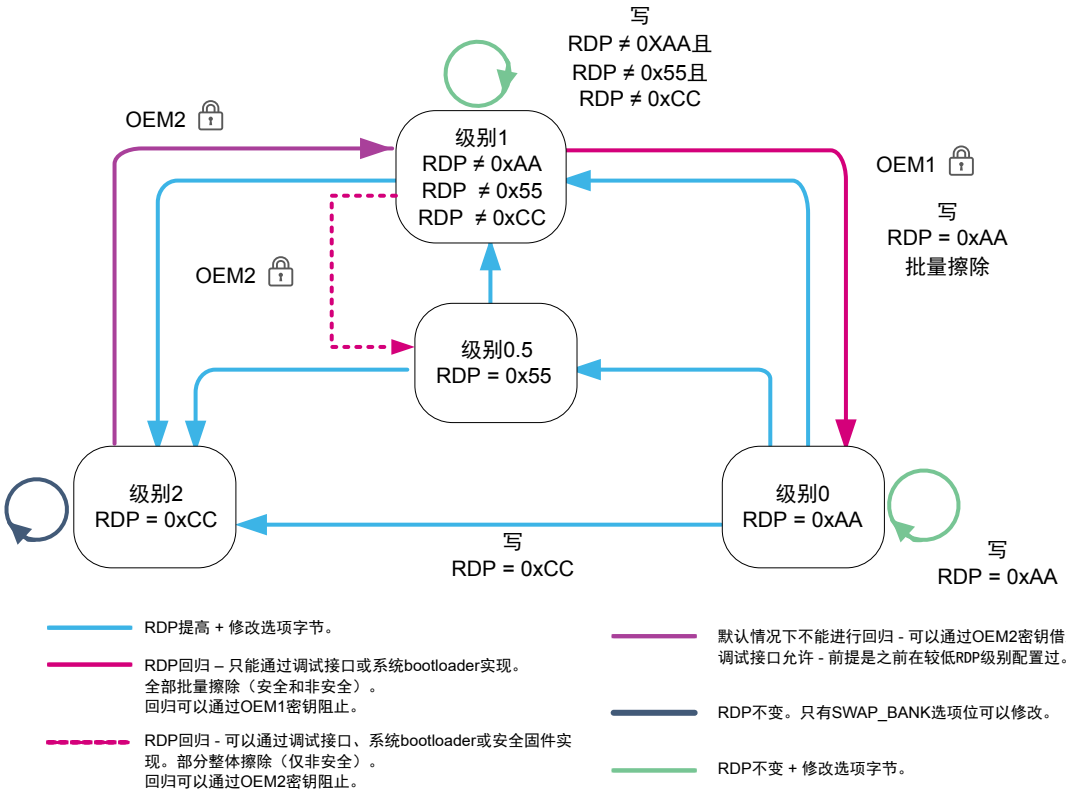
- 除了 SWAP_BANK 选项位，无法编程或擦除选项字节。
- 无法删除 RDP 级别 2（不可撤销的操作）。
- 所有调试功能均禁止。在复位模式下，还将禁用调试。
- 明确禁用 JTAG 和 SWD 输出。在复位模式下，可以使用 JTAG/SWD 实现回归。

提示 *在 STM32U5 中，如果在较低的 RDP 保护级别下提供了 OEM2 密钥，则 JTAG 和 SWD 在复位状态下保持启用状态，仅用于对接 DBGMCU_SR、DBGMCU_DBG_AUTH_HOST 和 DBGMCU_DBG_AUTH_DEVICE 寄存器，以获取设备标识并提供该 OEM2 密钥以请求 RDP 回归。*

7.4 通过 OEM 密钥实现 RDP 转换（仅限于 STM32U5）

下图显示当 TrustZone 已启用（TZEN = 1）时的 RDP 级别转换流程。

图 7. 当 TrustZone 启用后，RDP 级别转换流程



可以定义两个 64 位密钥（OEM1KEY 和 OEM2KEY），用于锁定 RDP 回归（无论有无 TrustZone）：

- 可修改 OEM1KEY：
  - 当 RDP 级别为 0 时
  - 当 RDP 级别为 0.5 或 1 时，前提是 OEM1LOCK 位已清除
- 可修改 OEM2KEY：
  - 当 RDP 级别为 0 或 0.5 时
  - 当 RDP 级别为 1 时，前提是 OEM2LOCK 位已清除

如要执行回归，需要在 DBGMCU_DBG_AUTH_HOST 寄存器中通过 JTAG 或 SWD 先后转变 OEMxKEY[31:0]和 OEMxKEY[63:32]。如果密钥匹配 OEM2KEY，则由硬件启动 RDP2 回归。

更多详细信息，请参见第 10 节 和文档[2]。

8 TrustZone 启用后可用的安全特性

以下特性只有在启用 TrustZone 后才可用：

- GTZC 安全水位线保护
- HDP（隐藏保护）选项字节
- 基于 Flash 存储块的安全保护
- RDP 级别 0.5
- RSS 和 SFI
- BOOT_LOCK
- 安全中断
- GTZC 安全保护

9 TrustZone 禁用

如第 2.1 节 中所述，在所有 STM32L5/U5 器件中，默认情况下都禁用了 TrustZone。设置 TZEN 选项位可以激活 TrustZone。

禁用 TrustZone 必须与 RDP 回退同步进行（参见第 7.2 节 ）。这就要假设系统的 RDP 级别已经为 1 或 0.5（从级别 0.5 进行回归仅适用于 STM32L5）。参见第 7.1 节 和第 7.2 节 了解需要考虑的相关建议。

禁用 TrustZone 后，第 8 节 中提及的所有特性均不再可用，所有安全寄存器都是 RAZ/WI。GTZC 仍然可以用于配置特权存取。

从 TZEN = 1 回归到 TZEN = 0 之后，样品为原始状态，对应于生产状态。

提示 *BOOT_LOCK 选项位一经设置便无法清除（仅适用于 STM32L5）。在清除并再次设置 TZEN 后，BOOT_LOCK 仍然是已设置状态，唯一的启动入口地址是在 SECBOOTADD0[24:0]选项字节中设置的地址。*

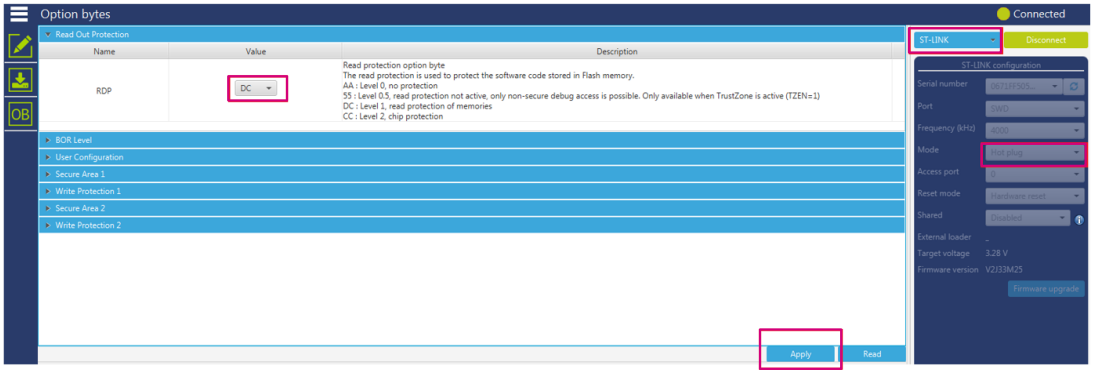
9.1 使用 STM32CubeProgrammer 进行 TrustZone/RDP 禁用演示

9.1.1 通过从用户 Flash 存储器启动实现 TZEN/RDP 回归

在从用户 Flash 存储器启动时执行 TZEN 和 RDP 回退需要以下顺序。

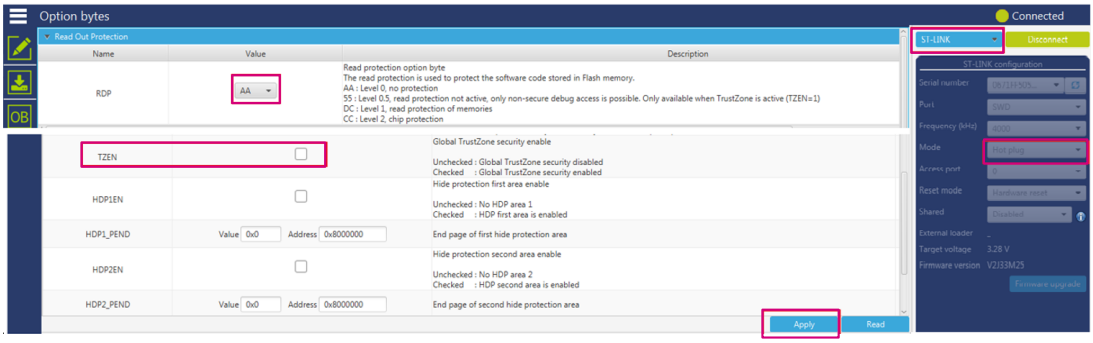
1. 确保安全和非安全应用程序得到正确加载和执行。
2. 通过 STM32CubeProgrammer 将 RDP 级别设为 1（如果是 STM32L5，则将级别设为 0.5) - 发生入侵。然后只能进行‘热插拔’连接。

图 8. 将 RDP 设置为 1 级



3. 请选择下列方案之一从入侵中恢复：
   a. 使用不同于 ST-LINK 的电源（详情请参见第 7.1 节 ），以便能够连接到目标。
   b. 移除 IDD 跳线，然后将其放回原位，即可退出入侵状态。
4. 将 RDP 设为级别 0（选项字节值 0xAA）并取消勾选 *TZEN* 复选框，然后点击*应用*。

图 9. 通过用户 Flash 启动从 SWD 实现 TZEN 和 RDP 回退



如果通过用户 Flash 启动实现 TZEN 和 RDP 回退不成功是因为第一步没有完成（安全应用程序没有调用一个非安全应用程序），实现回归的唯一方法就是从 RSS 进行启动，如下一节中所示。

9.1.2 通过 RSS 启动实现 TZEN/RDP 回退

本节讲解如何在意法半导体的 STM32L5/U5 板上更改启动方式。

通过在 BOOT0 引脚上应用一个高电平实现从 RSS 启动：

- 在评估板（STM32L552E-EV 或 STM32U575I-EV）上， 提供开关 SW1 ，用于更改起动方式（参见文档[4]或[5]）。
- 在探索套件（STM32L562E-DK 或 B-U585I-IOT02A）上，必须将其更改为从 RSS 启动（参见文档[6]或[7]）。
- 在 Nucleo 板（NUCLEO-L552ZE-Q 或 NUCLEO-U575ZI-Q）上，必须将 CN11 引脚 5（VDD）和引脚 7（PH3_BOOT0）连接（参见文档[8]或[9]）。

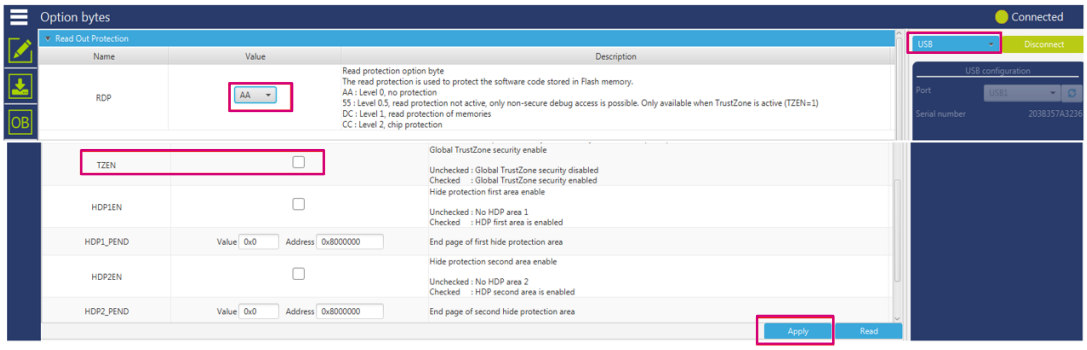
建议按照以下顺序从 RSS 启动：

1. 确保完成以下操作：
   a. 选中 nSWBOOT0 选项字节（BOOT0 取自 PH3/BOOT0 引脚）
   b. 在 PH3/BOOT0 引脚上施加高电压。
   c. NSBOOTADD1 选项字节配置为 0x0BF9 0000 地址（RSS 地址）的 x17F200 值。
   d. 取消选中 BOOT_LOCK 选项字节（启动基于焊盘/选项位配置）。
2. 通过 STM32CubeProgrammer 将 RDP 级别设为 1（入侵发生）。然后只能进行‘热插拔’连接。
3. 使用下列方案之一从入侵中恢复：
   a. 移除 IDD 跳线，然后将其放回原位，即可退出入侵状态。
   b. 使用不同于 ST-LINK 的电源，以便能够连接到目标。
4. 将 RDP 设为级别 0（选项字节值 0xAA）并取消勾选 TZEN 复选框。然后点击*应用*。

可以通过 JTAG/SWD 或bootloader实现回归，详情如下：

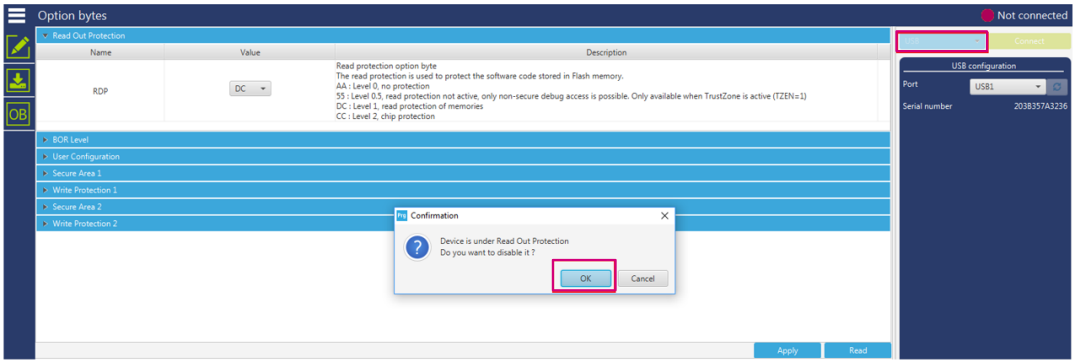
- 通过 JTAG/SWD：将 RDP 设为级别 0（选项字节值 0xAA）并取消勾选 *TZEN* 复选框，然后点击*应用*。
- 通过bootloader：使用一个受支持的通信接口（本例中是 USB 接口）：
  - 如果 RDP 设为级别 0.5，则将 RDP 级别设为级别 0 （选项字节值 0xAA）并取消勾选 *TZEN* 复选框，然后点击*应用*。
    注：只有使用 STM32L5 时，RDP 级别才能从 0.5 回归到 0。如果是 STM32U5, RDP 级别，则须首先从 0.5 提高到 1，然后再从级别 1 回归到 0，同时禁用 TZEN。

图 10. 通过bootloader实现 TZEN 和 RDP 回退（级别 0.5 变为级别 0）



  - 如果 RDP 级别设置为 1，则通过 STM32CubeProgrammer 图形界面实现 RDP 回归如下图所示。

图 11. 通过bootloader实现 RDP 回退（级别 1 变为级别 0）



如果 RDP 级别设为 1，不能使用 STM32CubeProgrammer 图形界面实现 TZEN 回归， 必须使用 STM32CubeProgrammer CLI（命令行指令），应用以下 TZEN 回归命令：

```
> STM32_Programmer_CLI.exe -c port=USB1 -tzenreg
      -------------------------------------------------------------------
                        STM32CubeProgrammer v2.8.0
      -------------------------------------------------------------------
USB speed    : Full Speed (12MBit/s)
Manuf. ID    : STMicroelectronics
Product ID   : DFU in FS Mode
SN           : 207E31953536
FW version   : 0x011a
Device ID    : 0x0482
Warning: Device is under Read Out Protection
Disabling TrustZone...
Disabling TrustZone successfully
```

9.1.3 当 RDP 级别 1 被 OEM1 密钥锁定后的 TZEN/RDP 回归

仅 STM32U5 系列器件提供该功能。设置 OEM1LOCK 位之后，OEM1 RDP 锁定机制被激活，之后阻止从 RDP 级别 1 到 RDP 级别 0 的回归。

可以在示例（OEM1 LSB 密钥 [31:0] = 0xABCDEFAB 且 OEM1 MSB 密钥 [63:32] = 0x12345678）中通过下面的 CLI 指令使用 OEM1 密钥锁定 RDP 级别 1。

```
>STM32_Programmer_CLI.exe -c port=swd  mode=hotplug  -lockRDP1 0xABCDEFAB 0x12345678
      -------------------------------------------------------------------
                        STM32CubeProgrammer v2.8.0
      -------------------------------------------------------------------

    Lock RDP1 password successfully done
```

如要将 RDP 级别提高到 1，可以使用以下 CLI。

```
>STM32_Programmer_CLI.exe -c port=swd  mode=hotplug  -ob rdp=0xDC
      -------------------------------------------------------------------
                        STM32CubeProgrammer v2.8.0
      -------------------------------------------------------------------

    Option Bytes successfully programmed
```

如第 2.1 节 中解释，TZEN 禁用必须在 RDP 级别为 1 时进行。必须提供 OEM1 密钥，以解锁 RDP 级别 1 回归。可以借助下面的 STM32CubeProgrammer CLI，通过 OEM1 密钥实现 TZEN + RDP 级别 1 到级别 0 的回归。

```
> STM32_Programmer_CLI.exe -c port=swd mode=UR -unlockRDP1 0xABCDEFAB 0x12345678 -ob
RDP=0xAA TZEN=0
      -------------------------------------------------------------------
                        STM32CubeProgrammer v2.8.0
      -------------------------------------------------------------------

    Unlock RDP1 password successfully done
    Option Bytes successfully programmed
```

10 演示使用 OEM 密钥进行 RDP 转换（仅 STM32U5 ）

为了达到最佳保护级别，建议激活 TrustZone 并设置 RDP 级别为 2，启用密码认证回归。

RDP 保护 Flash 主存储器、选项字节、备份寄存器、备份 RAM（仅 STM32U5）、OTFDEC 区域（可用时）、ICACHE、DCACHE，以及 SRAM。可以定义两个 64-位密钥（OEM1KEY 和 OEM2KEY），以锁定 RDP 回归。当 TrustZone 被激活之后，CPU 被划分为安全区域和非安全区域，并采用之前章节中描述的一组保护。

本节演示如何在提供 OEM1KEY 和 OEM2KEY 以解锁 RDP 回归时，使用 STM32CubeProgrammer CLI 练习 RDP 级别转换。

提示 *如需详细了解如何运用 OEMxKEY 进行 RDP 转换，请参阅文档[2]。*

在本例中，当 CPU 为安全时，RDP 回归通过 RSS 完成（将 PH3_BOOT0 PIN 连接到板件上的 VDD）。下表总结了第一列中链接的部分中详细描述的转换序列（当 CPU 为非-安全时，所有这些步骤都适用，除了转换到级别 0.5/从级别 0.5 开始转换)。

表 10. 演示使用 OEMxKE 实现 RDP 转换的步骤

步骤编号和标题	说明	注释
步骤 1 - 提供 OEM1KEY	配置 OEM1Key，用于解锁 RDP1 到 RDP0 的回归（`OEM1KEY=0x11ABCDEF 0x12ABCDEF`）。	用户可以选择任意 64 位密钥长度，除了全 1 或全 0。如果使用 `0xFFFFFFFF 0xFFFFFFFF`，则 OEMxKEY 被清除。
步骤 2 - 提供 OEM2KEY	配置 OEM2Key（`OEM2KEY=0x21ABCDEF 0x22ABCDEF`）： • 用于批准 RDP2 到 RDP1 的回归 • 用于解锁 RDP1 到 RDP0.5 的回归	
步骤 3 - 检查是否已提供 OEMxKEY	OEM1LOCK 和 OEM2LOCK 位设为 1。	如果 OEMxKEY 没有配置好，则用户必须重复失败的步骤 1 或步骤 2，并再次重新核查
步骤 4 - 设置选项字节 TZEN = 1	将 CPU 设为安全。	TrustZone 已启用
步骤 5 - 设置 RDP 级别 2	将 RDP 级别提高到 2（`-ob rdp=0xCC`）。 这表明，配置 OEM2KEY 之后，从 RDP 级别 2 到级别 1 的回归得到批准。	确保步骤 2 已通过，否则设备将无法访问。
步骤 6 - 通过 OEM2Key 解锁 RDP 级别 2	批准 RDP 级别 2 到级别 1 的回归。 必须提供正确的 OEM2KEY。	• 需要“复位状态下（UR）”模式。 • 当 TZEN = 1 时，需要从 RSS 启动。
步骤 7 - 将 RDP 级别设为 1	现在可以将 RDP 回归到级别 1（由步骤 6 解锁）。	如果不超过，则意味着未能使用正确的 OEM2KEY 解锁 RDP 级别 2 到级别 1 的回归。
步骤 8 - 通过 OEM2 密钥解锁 RDP 级别 1	启用 RDP 级别 1 到级别 0.5 的回归。	确保在 CLI 中，选项`-unlockrdp1` 与 `OEM2KY=0x21ABCDEF 0x22ABCDEF` 一起使用。
步骤 9 - 将 RDP 级别设为 0.5	现在能够将 RDP 级别 1 回归到级别 0.5，因为在步骤 8 中提供了 OEM2Key。	-
第 10.10 节 步骤 10 - 将 RDP 级别从 0.5 提高到 1	-	因为 RDP 级别 0.5 不能回归到级别 0，用户必须先将 RDP 级别提高到 1。
第 10.11 节 步骤 11 - 运用 OEM1Key 解锁 RDP 级别 1	必须提供正确的 OEM1KEY。	-
步骤 12 - 将 RDP 级别设为 0 并复位 TZEN = 0	TZEN + RDP 级别 1 回归到 0	-

10.1 步骤 1 - 提供 OEM1KEY

提供 OEM1KEY（0x11ABCDEF 0x12ABCDEF），以便锁定 RDP 级别 1 到级别 0 的回归。使用以下命令行：

```
> STM32_Programmer_CLI.exe -c port=swd mode=hotplug -lockrdp1 0x11ABCDEF 0x12ABCDEF
Device name : STM32U575/STM32U585
Flash size  : 2 MBytes
Device type : MCU
Device CPU  : Cortex-M33
BL Version  : 0x30
Debug in Low Power mode enabled
Lock RDP1 password successfully done
```

10.2 步骤 2 - 提供 OEM2KEY

提供 OEM2KEY（0x21ABCDEF 0x22ABCDEF），以便授权 RDP 级别 2 到级别 1 的回归，或者锁定 RDP 级别 1 到级别 0.5 的回归。使用以下命令行：

```
> STM32_Programmer_CLI.exe -c port=swd mode=hotplug -lockrdp2 0x21ABCDEF 0x22ABCDEF
Device name : STM32U575/STM32U585
Flash size  : 2 MBytes
Device type : MCU
Device CPU  : Cortex-M33
BL Version  : 0x20
Debug in Low Power mode enabled
Lock RDP2 password successfully done
```

10.3 步骤 3 - 检查是否已提供 OEMxKEY

检查 FLASH_NSSR 寄存器中的 OEM2LOCK 和 OEM1LOCK 位已设置，确保 OEM1KEY/OEM2KEY 已配置并锁定。使用以下命令行：

```
> STM32_Programmer_CLI.exe -c port=swd mode=hotplug -r32 0x40022020 4
Reading 32-bit memory content
  Size          : 4 Bytes
  Address:      : 0x40022020

0x40022020 : 000C0000   --> FLASH_NSSR[19:18]=11
```

使用 GUI 查看地址 0x40022020 或 FLASH_NSSR 寄存器的存储内容。

10.4 步骤 4 - 设置选项字节 TZEN = 1

设置 TZEN 选项字节，使用以下指令启用 TrustZone。

```
>STM32_Programmer_CLI.exe -c port=swd mode=hotplug -ob TZEN=1

UPLOADING OPTION BYTES DATA ...
  Bank          : 0x00
  Address       : 0x50022040
  Size          : 32 Bytes
[##################################################] 100%
  Bank          : 0x01
  Address       : 0x50022060
  Size          : 8 Bytes
[##################################################] 100%
  Bank          : 0x02
  Address       : 0x50022068
  Size          : 8 Bytes
[##################################################] 100%
OPTION BYTE PROGRAMMING VERIFICATION:
Option Bytes successfully programmed
```

10.5 步骤 5 - 设置 RDP 级别 2

将 RDP 级别提高到 2，方法是通过以下指令设置选项字节（-ob rdp=0xCC）：

```
>STM32_Programmer_CLI.exe -c port=swd mode=hotplug -ob rdp=0xCC


UPLOADING OPTION BYTES DATA ...
  Bank           : 0x00
  Address        : 0x50022040
  Size           : 32 Bytes
[##################################################] 100%
  Bank           : 0x01
  Address        : 0x50022060
  Size           : 8 Bytes
[##################################################] 100%
  Bank           : 0x02
  Address        : 0x50022068
  Size           : 8 Bytes
[##################################################] 100%
PROGRAMMING OPTION BYTES AREA ...
  Bank           : 0x00
  Address        : 0x50022040
  Size           : 32 Bytes
[--------------------------------------------------]
Reconnecting...
Error: failed to reconnect after reset !
UPLOADING OPTION BYTES DATA ...
  Bank           : 0x00
  Address        : 0x40022040
  Size           : 32 Bytes
[--------------------------------------------------]
Error: Uploading Option Bytes bank: 0 failed
Error: Reloading Option Bytes Data failed  --> Not possible to reconnect as RDP2
```

10.6 步骤 6 - 通过 OEM2Key 解锁 RDP 级别 2

提供 OEM2KEY，以批准从 RDP 级别 2 转换到 RDP 级别 1。

提示 *建议使用“复位状态下（UR）”模式。*

使用以下指令：

```
>STM32_Programmer_CLI.exe -c port=swd mode=UR -unlockrdp2 0x21ABCDEF 0x22ABCDEF
      -------------------------------------------------------------------
                        STM32CubeProgrammer v2.8.0
      -------------------------------------------------------------------
ST-LINK SN  : 0028003D3038510234333935
ST-LINK FW  : V3J8M3
Board       : NUCLEO-U575ZE
Voltage     : 3.31V
Unlock RDP2 password succefully done!
Error: Cannot connect to access port 0
If you are trying to connect to a device with TrustZone enabled please try to connect with
HotPlug mode
```

如果出现问题：

- 检查系统是否从 RSS 启动，并且板件的 PH3-BOOT0 引脚是否已连接到 VDD。
- 使用以下命令检查当 RDP 级别为 2 时的 DBGMCU 是否可访问（DBGMCU_CR @0xE0044000）：

```
>STM32_Programmer_CLI.exe -c port=swd mode=hotplug -r32 0xE0044104 4
Reconnected with the recommended frequency (3300 kHz)!
Device name : STM32U575/STM32U585
Flash size  : 2 MBytes
Device type : MCU
Device CPU  : Cortex-M33
BL Version  : 0x20
Debug in Low Power mode enabled
Reading 32-bit memory content
  Size          : 4 Bytes
  Address:      : 0xE0044104

0xE0044104 : 292D8E4A
```

如果 DBGMCU 不可访问，则意味着未提供 OEM2KEY。

10.7 步骤 7 - 将 RDP 级别设为 1

使用以下指令启动 RDP 级别 2 到级别 1 的回归（在步骤 6 中由 OEM2KEY 解锁）：

```
> STM32_Programmer_CLI.exe -c port=swd mode=hotplug -ob rdp=0xDC
….
UPLOADING OPTION BYTES DATA ...
  Bank           : 0x00
  Address        : 0x40022040
  Size           : 32 Bytes
[##################################################] 100%
  Bank           : 0x01
  Address        : 0x40022060
  Size           : 8 Bytes
[##################################################] 100%
  Bank           : 0x02
  Address        : 0x40022068
  Size           : 8 Bytes
[##################################################] 100%
OPTION BYTE PROGRAMMING VERIFICATION:
Option Bytes successfully programmed
```

10.8 步骤 8 - 通过 OEM2 密钥解锁 RDP 级别 1

提供 OEM2KEY 的目的是授权 RDP 级别 2 到级别 1 的回归，以及 RDP 级别 1 到级别 0.5 的回归。如果是第二种回归，则须使用下列指令通过 OEM2KEY（0x21ABCDEF 0x22ABCDEF）解锁 RDP 级别 1：

```
> STM32_Programmer_CLI.exe -c port=swd mode=hotplug -unlockrdp1 0x21ABCDEF 0x22ABCDEF
      -------------------------------------------------------------------
                        STM32CubeProgrammer v2.8.0
      -------------------------------------------------------------------
ST-LINK SN  : 0028003D3038510234333935
ST-LINK FW  : V3J8M3
Board       : NUCLEO-U575ZE
Voltage     : 3.31V
SWD freq    : 24000 KHz
Connect mode: Hot Plug
Reset mode  : Software reset
Device ID   : 0x482
Revision ID : Rev B
Reconnecting with the recommended frequency (1000 kHz)!
…
Reconnected with the recommended frequency (3300 kHz)!
Device name : STM32U575/STM32U585
Flash size  : 2 MBytes
Device type : MCU
Device CPU  : Cortex-M33
BL Version  : 0x90
Debug in Low Power mode enabled

Unlock RDP1 password successfully done
```

10.9 步骤 9 - 将 RDP 级别设为 0.5

使用以下指令启动 RDP 级别 1 到级别 0.5 的回归（在步骤 8 中由 OEM2KEY 解锁）：

```
>STM32_Programmer_CLI.exe -c port=swd mode=hotplug -ob rdp=0x55
….
UPLOADING OPTION BYTES DATA ...
  Bank           : 0x00
  Address        : 0x40022040
  Size           : 32 Bytes
[##################################################] 100%
  Bank           : 0x01
  Address        : 0x40022068
  Size           : 8 Bytes
[##################################################] 100%
OPTION BYTE PROGRAMMING VERIFICATION:
Option Bytes successfully programmed
```

10.10 步骤 10 - 将 RDP 级别从 0.5 提高到 1

使用以下命令将 RDP 级别提高到 1，以便能够在下一步达到 RDP 级别 0（不能从 RDP 级别 0.5 转换到级别 0）：

```
> STM32_Programmer_CLI.exe -c port=swd mode=hotplug -ob rdp=0xDC
….
UPLOADING OPTION BYTES DATA ...
  Bank           : 0x00
  Address        : 0x40022040
  Size           : 32 Bytes
[##################################################] 100%
  Bank           : 0x01
  Address        : 0x40022068
  Size           : 8 Bytes
[##################################################] 100%
OPTION BYTE PROGRAMMING VERIFICATION:
Option Bytes successfully programmed
```

10.11 步骤 11 - 运用 OEM1Key 解锁 RDP 级别 1

必须提供正确的 OEM1KEY，否则无法实现回归。以下指令解锁 RDP 级别 1：

```
> STM32_Programmer_CLI.exe -c port=swd mode=hotplug -unlockrdp1 0x11ABCDEF 0x12ABCDEF
…
Reconnected with the recommended frequency (3300 kHz)!
Device name : STM32U575/STM32U585
Flash size  : 2 MBytes
Device type : MCU
Device CPU  : Cortex-M33
BL Version  : 0xf0
Debug in Low Power mode enabled

Unlock RDP1 password successfully done
```

10.12 步骤 12 - 将 RDP 级别设为 0 并复位 TZEN = 0

TrustZone 启用后，禁用 TZEN 选项字节必须与 RDP 级别 1 到级别 0 的回归同步进行。使用以下指令：

```
> STM32_Programmer_CLI.exe -c port=swd mode=hotplug -ob rdp=0xAA tzen=0
….
UPLOADING OPTION BYTES DATA ...
  Bank           : 0x00
  Address        : 0x40022040
  Size           : 32 Bytes
[████████████████████████████████████████████████] 100%
  Bank           : 0x01
  Address        : 0x40022068
  Size           : 8 Bytes
[████████████████████████████████████████████████] 100%
OPTION BYTE PROGRAMMING VERIFICATION:
Option Bytes successfully programmed
```

提示 *如果 TrustZone 已禁用，则指令如下：*

```
STM32_Programmer_CLI.exe -c port=swd mode=hotplug -ob rdp=0xAA
```

10.13 清除 OEMxKEY

通过在 OEM1KEY 和 OEM2KEY Flash 存储选项字节中写入 0xFFFFFFFF 0xFFFFFFFF 来清除 OEM1Key 和/或 OEM2KEY。

FLASH_NSSR [19:18]中的 OEM1_LOCK 和/或 OEM2_LOCK 位因此被清除。可以按照第 10.3 节 中的描述进行检查（FLASH_NSSR [19:18] = 00）。

注意: 如果 OEM2KEY 已清除或根本没有提供，则 RDP 级别 2 无法回归到级别 1。

当如步骤 1（第 10.1 节 ）和步骤 2（第 10.2 节 ）中所示那样使用密钥初步提供 OEM1 和 OEM 密钥后：

- 可以在 RDP 级别为 0 时清除 OEM1key。
- 可以在 RDP 级别为 1、0.5 或 0 时清除 OEM2Key。

清除 OEM1KEY

在 STM32CubeProgrammer CLI 界面中使用以下指令：

```
> STM32_Programmer_CLI.exe -c port=swd mode=hotplug -lockrdp1 0xFFFFFFFF  0xFFFFFFFF
Device name : STM32U575/STM32U585
Flash size  : 2 MBytes
Device type : MCU
Device CPU  : Cortex-M33
BL Version  : 0x30
Debug in Low Power mode enabled
Lock RDP1 password successfully done
```

清除 OEM2KEY

使用以下指令：

```
> STM32_Programmer_CLI.exe -c port=swd mode=hotplug -lockrdp2 0xFFFFFFFF  0xFFFFFFFF
Device name : STM32U575/STM32U585
Flash size  : 2 MBytes
Device type : MCU
Device CPU  : Cortex-M33
BL Version  : 0x30
Debug in Low Power mode enabled
Lock RDP2 password successfully done
```

用户可以通过读取 FLASH_NSSR 寄存器中的内容来查看 OEM1LOCK 和 OEM2LOCK 位是否被清除，如步骤 3（第 10.3 节 ）所示。

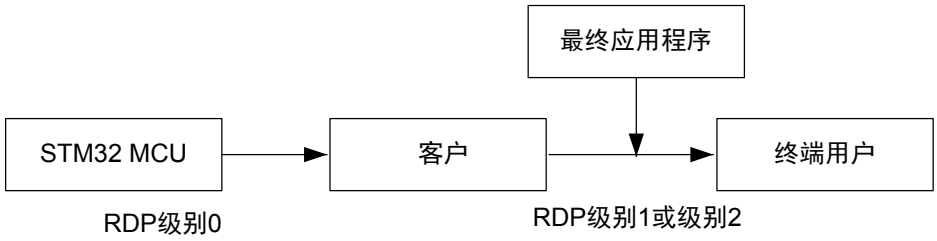
11 使用 TrustZone 的开发建议

11.1 开发方法

有两种开发方法：

- “单人开发”方法：开发人员（客户）负责开发安全和非安全应用程序。可以使用 RDP 级别 1 或 RDP 级别 2 保护用户应用程序。

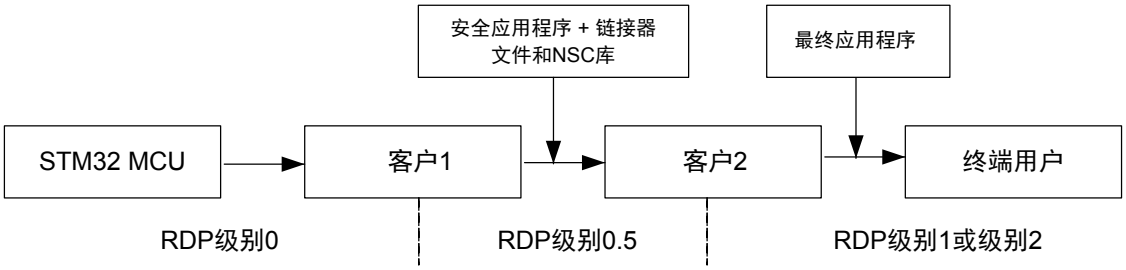
图 12. “单人开发”方法



- “双人开发”方法：第一位开发人员（客户 1）负责开发安全应用程序及其相关的非安全可调用库(`.lib`/`.h`)，并向第二位开发人员（客户 2）提供预定义的链接器文件，后者负责开发非安全应用程序。然后，该安全应用程序被加载到 STM32L5/U5 安全 Flash 存储器中，并使用 RDP 级别 0.5 进行保护，以防止进一步访问设备的安全内存区域。然后，第二位开发人员（客户 2）使用链接器文件和客户 1 提供的非安全可调用库，在预先编程的 STM32L5/U5 上开始进行开发。

  当 RDP 级别设置为 0.5 时，提供安全 Flash 存储器部分的客户 1 还必须考虑在安全端启动后为非安全端（面向客户 2）启用 JTAG/SWD 的方式。因此，客户 1 必须实现一个切换功能，以切换到非安全 Flash 存储器，允许客户 2 开发非安全部分，然后可能锁定设备到 RDP 级别 1 或级别 2。

图 13. “双人开发”方法



有关详情，请参阅文档[1]或[2]中的“产品生命周期”和“软件知识产权保护和协作开发”两节。

11.2 使用非安全外设

当外设被分配给非安全区域时，安全和非安全的应用程序都可以访问外设寄存器。

在非安全区域中，TrustZone for ARMv8-M 注意事项对于开发人员是完全透明的。在安全区域侧，应用程序必须确保外设所需的所有系统资源都是预先配置的，或者对于非安全区域（如 GPIO、NVIC 或 DMA）可用。

11.3 使用安全外设

当将外设分配给安全区域时，只授予安全寄存器访问权限。中断处理应该仅在安全区域中进行管理。根据使用此外设的安全项目和非安全项目之间的软件交互需求，可以采用两种不同的软件开发方法。

当使用不需要与非安全区域进行特定交互的外设时，安全区域将它们作为标准外设进行驱动，而不需要考虑任何特定因素。

如果需要在非安全区域和安全区域之间进行交互以驱动安全外设，则安全应用程序必须提供非安全的可调用 API 和对非安全区域的回调。

12 结论

Arm TrustZone 技术将硬件划分为安全区域和非安全区域。

通过使用固定内存映射定义安全属性的 IDAU，用户可配置的 SAU, 和在 Flash 存储器和 GTZC 中的其他特性，所有 STM32 MCU 资源都可以在安全和非安全区域进行配置，包括内存映射、Flash 存储器、SRAM、外部存储、外设和外围中断。

版本历史

表 11. 文档版本历史

日期	版本	变更
2019 年 10 月 11 日	1	初始版本。
2019 年 10 月 14 日	2	更新了图 6。内存和外设数据存取规则汇总。
2020 年 2 月 10 日	3	更新了： • 引言 • 第 2.1 节 概述 • 第 2.2 节 安全状态 • 第 3.1 节 激活 STM32L5 系列器件的 TrustZone • 第 3.3 节 SAU 和 IDAU • 图 3.地址安全属性 • 第 3.3.1 节 STM32L5 系列的 IDAU 和存储器别名使用 • 表 1.STM32L5 系列器件上的 IDAU 内存映射地址安全属性 • 第 3.3.2 节 STM32L5 系列的 SAU • 第 4 节 STM32L5 系列器件上的安全配置 • 第 4.1 节 Flash 存储器的安全配置 • 第 4.1.1 节 Flash 存储器的安全水位线 • 第 4.1.2 节 基于 Flash 存储器块的特性 • 第 5.1 节 默认安全状态 • 第 5.2 节 内存和外设安全访问规则 • 图 6.内存和外设数据存取规则汇总 • 第 6 节 启动和根安全服务（RSS） • 第 7.1 节 RDP 级别 1 和第 7.2 节 RDP 级别 0.5 • 第 8 节 TrustZone 启用后才可用的安全特性 • 第 9 节 禁用 TrustZone • 第 10.1 节、第 10.2 节，以及第 10.3 节 • 第 11 节 总结 增加了： • 第 1.1 节 参考文档 • 图 5.默认的 Flash 存储区安全状态，与选项字节无关 • 表 4.指令获取规则 • 第 10 节 使用 TrustZone 的开发建议
2020 年 3 月 2 日	4	更新了： • 第 7.1 节 RDP 级别 1 • 第 7.2 节 RDP 级别 0.5 增加了第 9.1 节 使用 STM32CubeProgrammer 进行 TrustZone/RDP 禁用演示。

日期	版本	变更
2021 年 9 月 28 日	5	更新了： • 标题与全文，集成 STM32U5 系列 • “第 1 节 基本信息”中的参考文档 • 第 3.3.2 节 STM32L5 和 STM32U5 的 SAU • 第 4 节 STM32L5 和 STM32U5 系列的安全配置 • 第 7 节 TrustZone 启用之后的读取保护（RDP） • 第 8 节 TrustZone 启用后才可用的安全特性 • 第 9.1.2 节 通过从 RSS 启动实现 TZEN/RDP 回归 • 第 11.1 节 开发方法 增加了： • 第 7.3 节 RDP 级别 2 • 第 7.4 节 使用 OEM 密钥进行 RDP 转换（仅 STM32U5） • 第 9.1.3 节 RDP 级别 1 被 OEM1 密钥锁定后的 TZEN/RDP 回归 • 第 10 章节 演示使用 OEM 密钥进行 RDP 转换（仅 STM32U5）
2022 年 4 月 8 日	6	更新了： • 各种拼写错误

目录

表一览

图一览