利用 Armv8-M 架构和 TrustZone 提高嵌入式系统安全性

作者：Jacob Beningo
投稿人：Digi-Key 北美编辑
2018-11-06

确保基于微控制器的物联网应用的安全性可能会很棘手。安全性始于硬件层次，然后扩展到嵌入式软件。为了成功保障软件安全，开发人员要求底层硬件支持以下等关键特性：

- 安全启动
- 存储器保护
- 加密引擎加速器
- 真随机数发生器 (TRG)
- 安全引脚复用
- 软件隔离

虽然 Arm® Cortex®-M 处理器（如 M0+、M3/4/7 系列）支持其中一些特性，但要打造成功的解决方案可能既困难又耗时。

开发人员可以在硬件层次上运用一种新解决方案，即使用基于 Armv8-M 架构的新型 Cortex-M23/33 系列微控制器。这些处理器在设计时就考虑了安全性，包含许多安全特性，例如前面列出的那些特性，包括用于微控制器的 Arm TrustZone®。通过本文，我们将更加熟悉 Armv8-M 架构，并探索如何利用 TrustZone 来提高嵌入式系统安全性。

Armv8-M 架构简介

关于 Armv8-M 架构，首先要知道的是它是 Arm 的最新微控制器架构，面向低成本、深度嵌入的实时嵌入式系统而推出。该系列中加入了三款新型处理器：M23 是低功耗版本，M33 是高性能版本，最近推出的 M35P 是一款高性能、物理安全（防篡改）处理器（图 1）。

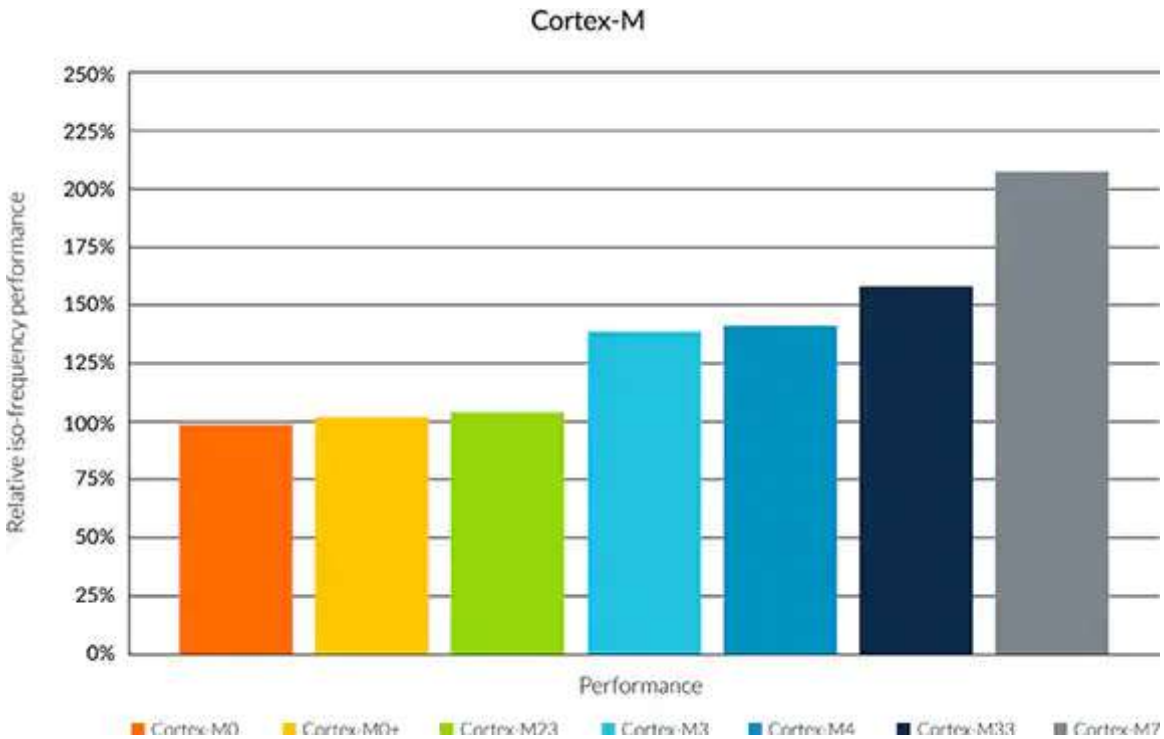

图 1：从性能角度看，新型 Cortex-M23/33 处理器是该系列中 Cortex-M0+ 和 Cortex-M4 处理器的改良型。（图片来源：Arm）

相对于以前的几代架构，Armv8-M 架构改善了性能，值得注意的几项重要改进包括：

- 指令集增强功能

- 灵活的断点配置
- 动态重设中断优先级
- 增强的跟踪支持
- 更简单的存储器保护单元 (MPU) 设置

该架构最大和最有意义的改进是能够使用 Arm TrustZone。TrustZone 是该架构的安全扩展，允许开发人员从物理上隔离执行代码与存储区域，如 RAM、代码空间和硬件中的外设。TrustZone 支持将软件分拆到安全和不安全的区域中，然后在安全或非安全的处理器状态下执行。安全状态允许全面访问处理器的存储器和外设，而非安全状态只能访问非安全区域和故意暴露给非安全代码的安全功能（图 2）。

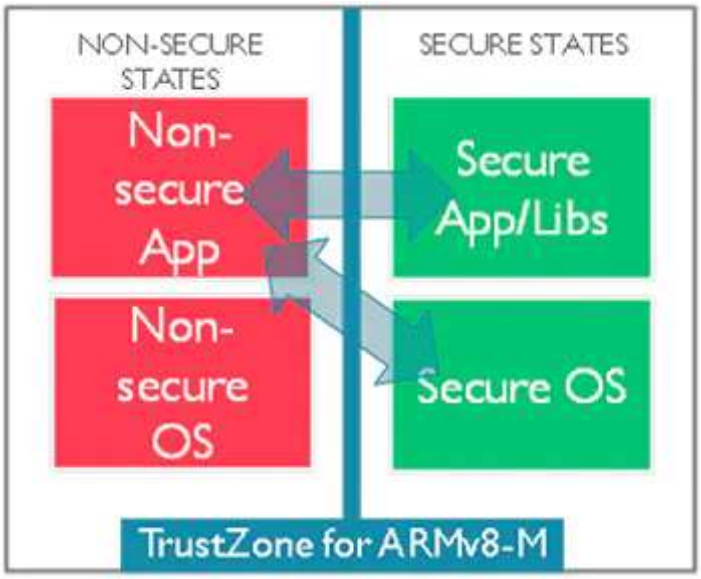

图 2：TrustZone 使用硬件隔离将处理器和应用区分为非安全和安全状态。在非安全状态下执行的代码不能访问或篡改安全存储器或代码。只有在安全状态下运行时才能访问安全存储器和代码。（图片来源：Arm）

开发人员可以选择哪些闪存和 RAM 位置属于安全状态，哪些属于非安全状态。当非安全代码调用安全功能时，非安全和安全状态之间的切换完全是在硬件中以确定的方式处理，在最坏的情况下，切换时间开销为三个时钟周期。CPU 中有几个寄存器为安全状态和非安全状态所共享，但每个状态还有自己的堆栈指针、故障和控制寄存器。M33 甚至还有一个堆栈限值寄存器，可用于检测堆栈溢出。

值得注意的是，TrustZone 是一个处理器扩展，这意味着由器件是否包含 TrustZone 支持要由处理器制造商决定。既然 TrustZone 是可选的，下面我们就来了解目前可用的几款 Armv8-M 处理器以及它们如何处理 TrustZone。

选择支持 TrustZone 的 Armv8-M 处理器

目前有几款处理器支持 Armv8-M 处理器。有意思的是，这些器件是如此新颖，以至于到 2018 年夏末，只有 Microchip Technology 一家制造商在生产。

其他处理器制造商（如 Nuvoton）已宣布即将推出此类器件。我们预计在未来 12 个月内，Armv8-M 器件数量将大幅增加，包括那些支持 TrustZone 的器件。

Microchip 主要生产两个版本的 Armv8-M 架构，即 SAML10 和 SAML11 系列器件。SAML10 版本不包含 TrustZone，而 SAML11 版本则包含。图 3 显示了目前正在生产并供货的 SAML10 和 SAML11 器件的所有型号。这些型号之间的主要区别在于 RAM、闪存、引脚和外设的有无及多少，这些是我们在选择微控制器时会考虑的因素。

配置总结

SAM L10/L11 器件特定特性

器件	闪存 + 数据闪存 (KB)	SRAM (KB)	引脚	SERCOM	ADC 通道	模拟比较器输入	PTC 自电容/互电容通道	I/O 引脚	防篡改引脚	封装
SAML10D14	16+2	4	24	2	5	2	16/64	17	3	VQFN、SSOP
SAML10D15	32+2	8								
SAML10D16	64+2	16								
SAM10E14	16+2	4	32	3	10	4	20/100	25	4	VQFN、TQFP、WLCSP
SAML10E15	32+2	8								
SAML10E16	64+2	16								
SAML11D14	16+2	8	24	2	5	2	16/64	17	3	VQFN、SSOP
SAML11D15	32+2	8								
SAML11D16	64+2	16								
SAML11E14	16+2	8	32	3	10	4	20/100	25	4	VQFN、TQFP、WLCSP
SAML11E15	32+2	8								
SAML11E16	64+2	16								

图 3：Microchip SAML10 和 SAML11 微控制器型号。仅 SAML11 器件包含 Arm TrustZone。（图片来源：Microchip Technology）

对于希望开始使用 Armv8-M 的开发人员，有两种开发套件可供选择。Microchip SAML10 Xplained 评估板包括 SAM L10E14A 微控制器，后者含有 16 KB 闪存、2 KB 数据闪存、4 KB SRAM，并采用 32 引脚封装。Microchip SAML11 Xplained 评估板包括 SAM L11E16A 微控制器，后者含有 64 KB 闪存、2 KB 数据闪存、16 KB SRAM，同样采用 32 引脚封装。除了处理器不同之外，开发板是相同的。图 4 所示为 Xplained 板的图片。

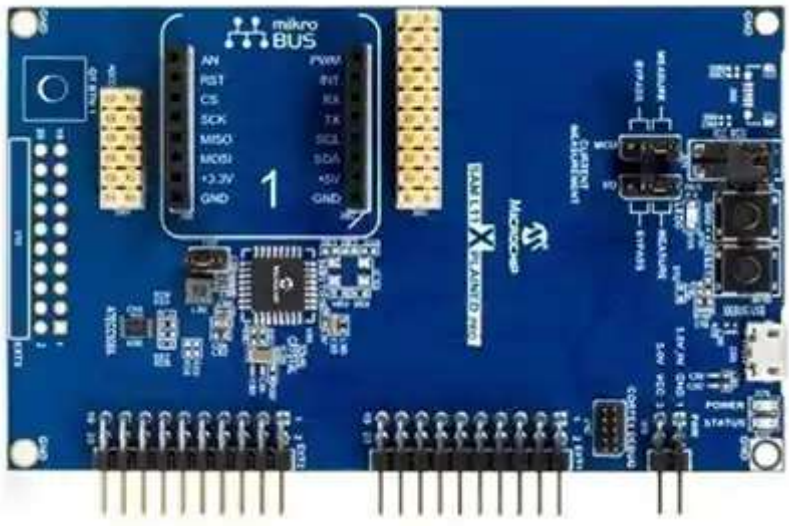

图 4：Microchip SAML10/L11 开发板基于 Armv8 架构。SAML11 版本支持 TrustZone（图像来源：Keil）

TrustZone 应用的工作原理

使用 TrustZone 的开发人员会发现，嵌入式应用的开发方式将发生巨大变化。首先，开发人员需要区分其应用空间，确定哪些代码和库属于安全状态，哪些属于非安全状态。

确定后，开发人员创建两个不同的软件应用：一个用于安全代码，一个用于非安全代码。这可以利用 Keil MDK 之类的编译器/IDE 非常容易地完成。开发人员最终得到的是一个多项目工作空间，其中一个项目是安全代码，另一个是非安全代码（图 5）。

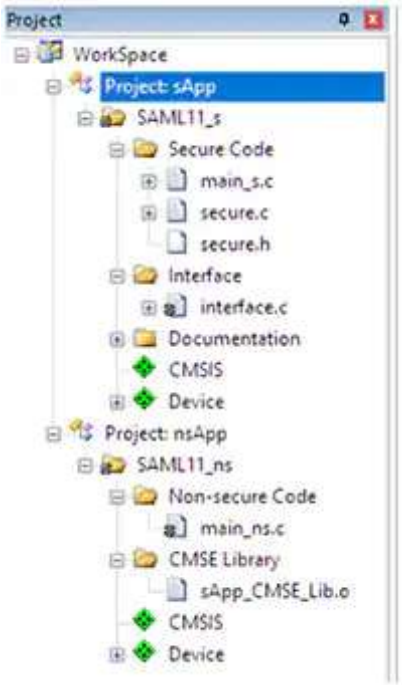

图 5：使用 TrustZone 时，开发人员最终会得到一个多项目工作空间，其中一个项目专门用于安全代码，另一个项目用于用户代码。（图片来源：Keil）

当 TrustZone 应用启动时，代码开始在安全状态下执行。这样，开发人员可以立即建立信任根，由此执行应用的其余部分。系统完成启动后，应用将从安全状态切换到非安全状态，并执行所谓的用户代码。此时，应用就像任何其他嵌入式应用一样执行。主要区别在于非安全代码只能通过安全网关访问安全功能和回调（图 6）。

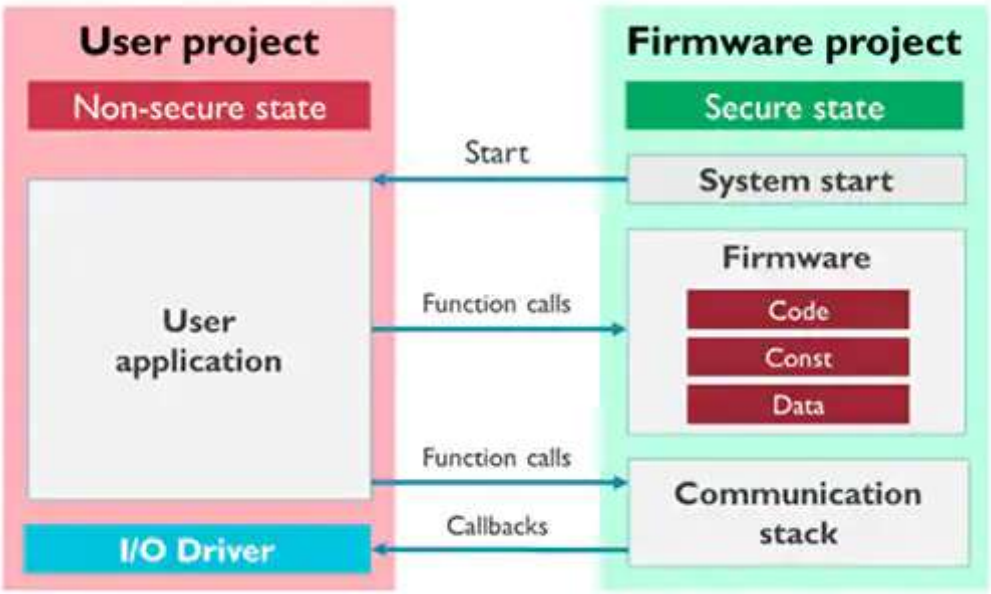

图 6：TrustZone 应用从安全状态开始执行，在建立信任根之后进入非安全状态。非安全状态只能对安全代码中的公开函数进行函数调用，否则会抛出异常。（图片来源：Keil）

如果用户应用试图不通过安全网关访问安全代码、存储器或外设，就会产生异常。毫无疑问，这意味着软件有缺陷，生产环境中有漏洞，或者黑客试图访问系统。此时，代码可以决定如何阻止攻击，例如重启系统以删除任何可能在非安全 SRAM 中运行的注入代码。

利用 TrustZone 确保嵌入式应用安全性的技巧和诀窍

有很多技术可以帮助提高嵌入式系统的安全性。下面几点技巧和诀窍有助于提高开发人员使用 Armv8-M 架构和 TrustZone 的兴趣：

- 在复位期间使用安全区来建立信任根和可信执行环境。
- 将安全关键任务、库和密钥放入安全区。
- 将用户代码放在非安全区中。
- 为了简单起见，将 RTOS 内核放在一个位置，安全区或非安全区均可。
- 在安全区和非安全区中使用 MPU 可以改善进程隔离。
- 尽量减少安全区中的代码有助于尽量减小使安全代码攻击面。
- 启动从安全状态到非安全状态的转换之前，确保安全代码清除任何来自未分组寄存器中的秘密信息。

总结

确保基于微控制器的物联网应用的安全性十分重要，但也很棘手。安全性始于硬件层面，但许多运行 Cortex-M0+、Cortex-M3/4/7 内核的传统微控制器系列可能缺乏保障器件安全所需的特性。开发人员现在可以利用 Cortex-M23 和 Cortex-M33 内核上的新型 Armv8-M 架构，来保障使用越来越多这种架构处理器的嵌入式应用的安全性。