

概述

为了降低软件盗窃的风险，许多芯片都有芯片内部的程序保护。SPC1169 的设计考虑到了程序保护的重要性，为用户提供了强大可靠的程序保护功能，其中包括调试锁定和随机数保护。

目录

1	安全功能.....	7
1.1	随机数据保护	7
2	安全功能的具体实现	9

SPIN TROL

图片列表

图 1-1: 随机数在应用程序中的流程图	8
图 2-1: 实际生产过程中安全功能的具体实现流程图	9

SPIN TROL

表格列表

表 1-1: 配置字的描述 7

SPIN TROL

版本历史

版本	日期	作者	状态	变更
A/0	2023 年 4 月 11 日	CanChai	Released	首次发布。

SPIN TROL

术语或缩写

术语或缩写	描述

SPIN TROL

1 安全功能

当调试或烧录程序时，芯片的调试接口经常被使用。这也意味着通过调试接口可以读取或修改芯片内部存储器的内容。因此，必须在批量生产过程中禁用芯片的调试功能。否则，通过调试接口可以获取芯片内部程序。

通过设置配置字的 `CHIP_SECURITY` 字段，可以锁定 SPC1169 的调试接口。配置字的描述如表 1 所示。一旦 SPC1169 的调试接口被锁定，无法通过调试接口访问内部存储器。此外，通过 ROM 中的引导加载程序读取、编程和扇区擦除内部存储器也被禁用；ROM 中的引导加载程序仅支持擦除 Flash 存储器。

通过以下方法可以解锁 SPC1169 的调试接口：

- 通过 ROM 中的引导加载程序对整个内部 Flash 存储器进行芯片擦除。
- 通过 ROM 中的引导加载程序输入正确的非安全密钥（等于配置字的 `UNSECURITY_KEY` 字段）。此方法临时解锁调试接口。这意味着芯片重新启动时调试接口将保持锁定状态。如果 `UNSECURITY_KEY` 字段的值为 `0x00000000_00000000`，则表示无法通过输入非安全密钥解锁调试接口。

表 1-1: 配置字的描述

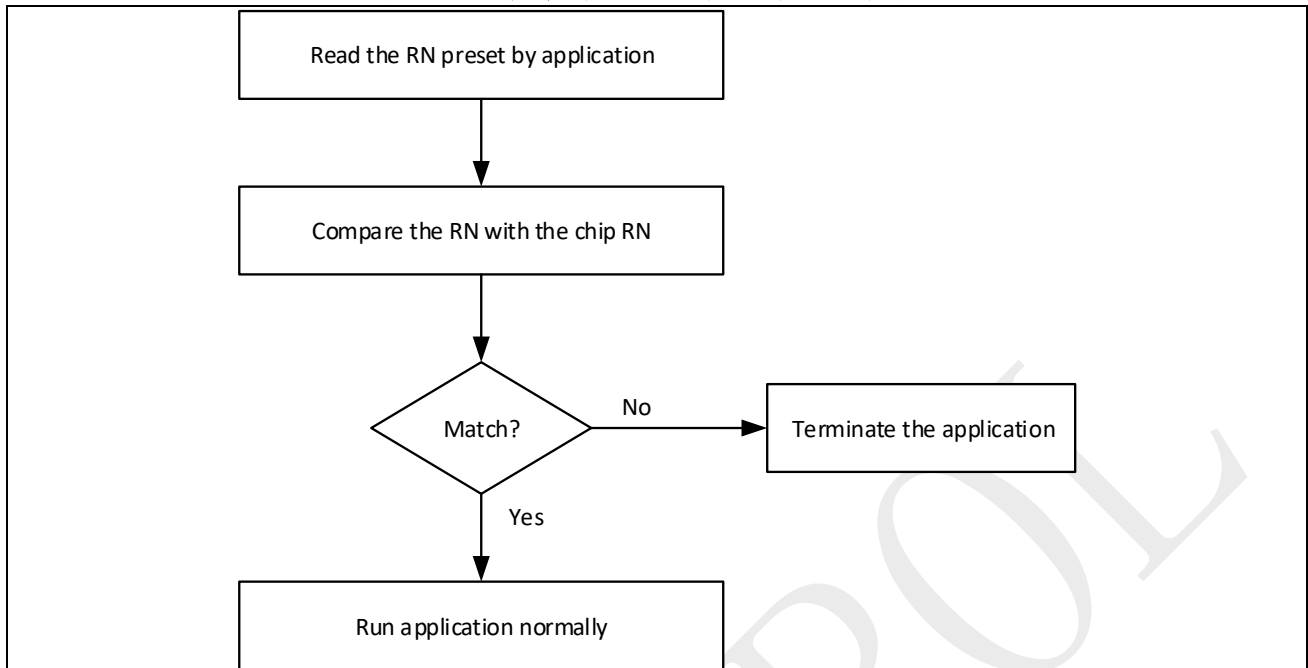
地址	名字	描述
0x1001FFF0	UNSECURITY_KEY	解锁芯片的调试接口需要输入一个 8 字节的密钥 <code>UNSECURITY_KEY</code> ，当调试接口被锁定时，只有正确的密钥才能使调试接口暂时解锁。如果 <code>UNSECURITY_KEY</code> 字段的值全部为 <code>0x00</code> ，则不能通过输入密钥来解锁调试接口。
0x1001FFF8	CHIP_SECURITY	芯片调试接口锁定字 <code>0xFFFFFFFF</code> ：芯片调试接口将不被锁定 其他值：芯片调试接口将被锁定
0x1001FFFC	WDT_ENABLE	Watchdog 使能字 <code>0xFFFFFFFF</code> ：在芯片启动时禁用看门狗 其他：在芯片启动时启用看门狗

1.1 随机数据保护

有些芯片破解者可能有能力使用特定设备去除芯片封装，读取 Flash 内存中的数据，然后将其写入未编程的芯片以复制用户的产品。SPC1169 也针对这种情况具有适当的保护机制。

SPC1169 出厂时被写入一个 8 字节的随机数，该随机数无法修改。客户的应用程序可以在运行时从芯片中读取并验证随机数。如果随机数与应用程序预设的值不匹配，则应用程序将终止。由于芯片的随机数彼此不同，即使芯片破解者从一颗芯片的 Flash 存储器中获取程序数据，然后将程序数据写入其他 SPC1169 芯片，程序也无法正常工作。通过这种方式，客户的产品无法大量复制。

图 1-1: 随机数在应用程序中的流程图



在图 1-1: 随机数在应用程序中的流程图中, 应用程序预设的随机数 (RN) 可以在产品生产过程中通过编程器工具实现。编程器工具读取目标芯片的随机数, 并将其写入目标芯片 Flash 存储器的特定地址。当应用程序启动时, 它从目标芯片 Flash 存储器的特定地址读取预设的随机数, 并将其与目标芯片的随机数进行比较。

2 安全功能的具体实现

对于在现场大规模生产中实现芯片安全性的步骤如图 2-1: 实际生产过程中安全功能的具体实现流程图所示。

图 2-1: 实际生产过程中安全功能的具体实现流程图

