life.augmented

AN3429
应用笔记

STM32 专有代码保护概述

前言

软件提供商开发的复杂中间件解决方案 （知识产权 (IP) 代码）需要进行保护。

这种 IP 代码必须能够以终端用户应用程序插件的形式来构建完整的解决方案。全局保护机制会通过专用的应用程序编程接口 (API) 限制对该代码的访问，同时会防止任何读访问。

本应用笔记概括介绍了用于防止专有代码被终端用户代码、调试器工具或 RAM 木马代码读出的机制。该机制提供了全面的 API，因此 IP 代码可由终端用户应用程序轻松调用，并且仍可防止直接访问 IP 代码本身。

建议使用的解决方案基于 MPU 功能以及终端用户应用程序和 IP 代码中的特殊存储器和外设管理机制。

在 STM32 专有代码保护方法中，会使用两种保护级别：

- 全局读出保护（全局 ROP）：通过 STM32 ROP 防止 IP 代码和终端用户代码（被调试器工具或 RAM 木马代码）直接读取。
- IP 代码读出保护 (IP ROP)：通过 MPU 防止 IP 代码被终端用户代码（可能是木马代码）读取。

因为一旦激活了全局 ROP，用户就不再能够完全控制 Flash 编程，因此主应用程序 （IP 代码）必须嵌入 IAP 层。该 IAP 可在不破坏受保护代码区域的同时加载终端用户应用程序。

若需完整解决方案的更详细信息，请联系您本地的 ST 销售代表。

目录

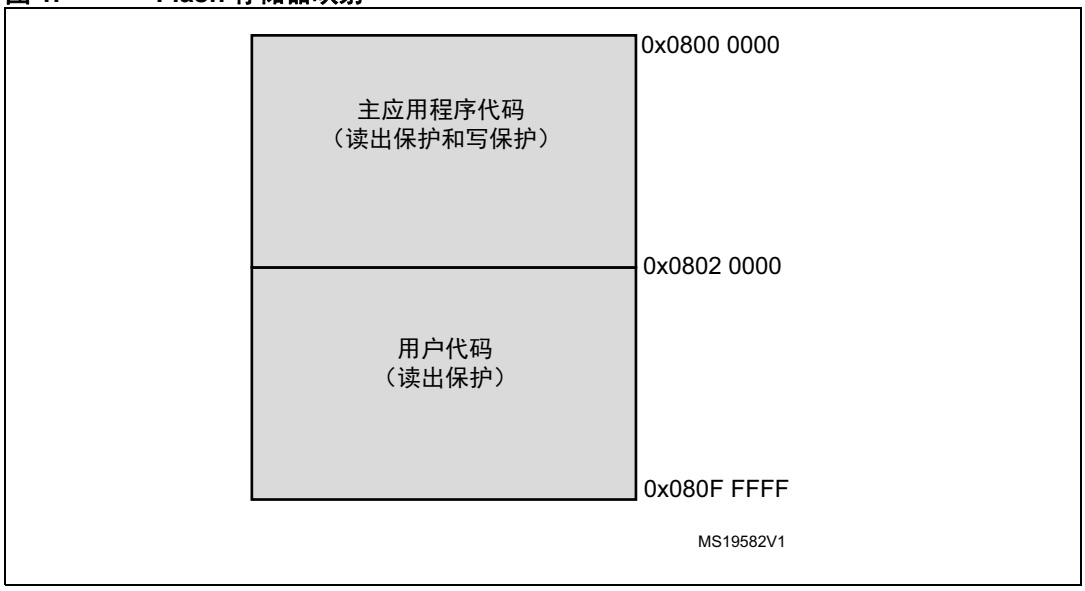
1 代码保护概述

Flash 存储器分为两个主要区域：

- 主应用程序代码: 包含 IP 存储器保护代码、需要保护的 IP 代码以及启用终端用户应用程序加载的 IAP 代码。
- 用户代码：包含终端用户应用程序，使用 IP 存储器保护代码提供的主系统应用程序 API 在保护模式下提供对 IP 代码的访问。

图 1 举例说明了 Flash 存储器映射 （STM32F2 系列）

图 1. Flash 存储器映射



IP 代码的起始地址为 0x0801 0000，在链接器文件中的定义如下：

define region IP_CODE_region = mem:[from 0x08010000 size 0x10000];

place in IP_CODE_region { section IP_Code };

在本例中，会通过 IAR 使用以下编译指示将要保护的函数强制加载到 IP 代码段：

#pragma location="IP_Code"

（函数定义）

终端用户项目会使用包含终端用户应用程序使用的函数 IP 代码 API 的 *exported_api.h* 文件。

2 版本历史

表 1. 文档版本历史

日期	版本	变更
2011 年 7月 19 日	1	初始版本。

表 2. 中文文档版本历史

日期	版本	变更
2016 年 9月 20 日	1	中文初始版本。

重要通知 - 请仔细阅读

意法半导体公司及其子公司 （“ST”）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 ST 产品的最新信息。 ST 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 ST 产品的选择和使用， ST 概不承担与应用协助或买方产品设计相关的任何责任。

ST 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 ST 产品如有不同于此处提供的信息的规定，将导致 ST 针对该产品授予的任何保证失效。

ST 和 ST 徽标是 ST 的商标。所有其他产品或服务名称均为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。