



## 简介

本编程手册介绍了如何烧写STM32F101xx、STM32F102xx和STM32F103xx微控制器的闪存存储器。为方便起见，在本文中出特别说明外，统称它们为STM32F10xxx。

STM32F10xxx内嵌的闪存存储器可以用于在线编程(ICP)或在程序中编程(IAP)烧写。

在线编程(In-Circuit Programming – ICP)方式用于更新闪存存储器的全部内容，它通过JTAG、SWD协议或系统加载程序(Bootloader)下载用户应用程序到微控制器中。ICP是一种快速有效的编程方法，消除了封装和管座的困扰。

与ICP方式对应，在程序中编程(In-Application Programming – IAP)可以使用微控制器支持的任何一种通信接口(如I/O端口、USB、CAN、UART、I<sup>2</sup>C、SPI等)下载程序或数据到存储器中。IAP允许用户在程序运行时重新烧写闪存存储器中的内容。然而，IAP要求至少有一部分程序已经使用ICP烧到闪存存储器中。

闪存接口是在AHB协议上实现了对指令和数据的访问，它通过对存储器的预取缓存，加快了存储器的访问；闪存接口还实现了在所有工作电压下对闪存编程和擦除所需的逻辑电路，这里还包括访问和写入保护以及选择字节的控制。

1	概述 .....	4
1.1	特性 .....	4
1.2	闪存模块组织 .....	4
2	读/编写STM32F10xxx内置闪存.....	7
2.1	简介 .....	7
2.2	读操作 .....	7
2.2.1	取指令 .....	7
2.2.2	D-Code接口 .....	7
2.2.3	闪存访问控制器.....	7
2.3	闪存编程和擦除控制器(FPEC).....	8
2.3.1	键值 .....	8
2.3.2	解除闪存锁.....	8
2.3.3	主闪存编程.....	8
2.3.4	闪存擦除 .....	9
2.3.5	选择字节编程 .....	10
2.4	保护.....	12
2.4.1	写保护 .....	12
2.4.2	读保护 .....	12
2.4.3	选择字节块写保护 .....	13
2.5	选择字节说明 .....	13
3	寄存器说明 .....	15
3.1	闪存访问控制寄存器(FLASH_ACR) .....	15
3.2	FPEC键寄存器(FLASH_KEYR).....	16
3.3	闪存OPTKEY寄存器(FLASH_OPTKEYR).....	16
3.4	闪存状态寄存器(FLASH_SR).....	17
3.5	闪存控制寄存器(FLASH_CR).....	18
3.6	闪存地址寄存器(FLASH_AR).....	19
3.7	选择字节寄存器(FLASH_OBR) .....	19
3.8	写保护寄存器(FLASH_WRPR).....	20
3.9	闪存寄存器映像 .....	21

## 术语

下面列出了本文档中所用到的术语和缩写的简要说明：

- ⌘ **小容量产品**是指闪存存储器容量在16K至32K字节之间的STM32F101xx, STM32F102xx 和 STM32F103xx微控制器。
- ⌘ **中容量产品**是指闪存存储器容量在64K至128K字节之间的STM32F101xx, STM32F102xx 和STM32F103xx微控制器。
- ⌘ **大容量产品**是指闪存存储器容量在 256K 至 512K 字 节 之 间 的 STM32F101xx 和 STM32F103xx微控制器。
- ⌘ **Cortex-M3内核**集成了两个调试端口：
  - **JTAG调试接口(JTAG-DP)**提供基于JTAG(Joint Test Action Group 联合测试行动小组)协议的5线标准接口。
  - **SWD调试接口(SWD-DP)**提供基于SWD(Serial Wire Debug 串行线调试)协议的2线标准接口。有关JTAG和SWD协议，请参考*Cortex M3 Technical Reference Manual*。
- ⌘ **字(Word)**: 32位长的数据或指令
- ⌘ **半字(Half Word)**: 16位长的数据或指令
- ⌘ **字节(Byte)**: 8位长的数据或指令
- ⌘ **FPEC(FLASH Program/Erase controller 闪存编程/擦除控制器)**: 内嵌的FPEC负责对内置闪存的写操作。
- ⌘ **IAP(In-Application Programming)**: IAP是在用户程序运行时对闪存微控制器中存储器重新编程。
- ⌘ **ICP(In-Circuit Programming)**: ICP是在芯片安装到用户应用板上后，通过JTAG协议对闪存微控制器中存储器编程。
- ⌘ **I-Code**: 这是连接Cortex-M3核心的指令总线与闪存程序存储器接口的总线，指令预取是在这个总线上实现的。
- ⌘ **D-Code**: 这是连接Cortex-M3核心的D-Code总线(常数和调试访问)与闪存数据存储器接口的总线。
- ⌘ **选择字节**: 存放在闪存中的产品配置位。
- ⌘ **OBL**: 选择字节加载单元。
- ⌘ **AHB**: 先进高性能总线。

# 1 概述

## 1.1 特性

- ⌘ 多达512K字节闪存
- ⌘ 存储器配置：
  - 主存储块：
    - 小容量产品：4K x 64位
    - 中容量产品：16K x 64位
    - 大容量产品：64K x 64位
  - 信息块：258 x 64位

闪存接口的特性：

- ⌘ 带预取缓冲器的读接口(2x64位)
- ⌘ 选择字节加载
- ⌘ 闪存编程/擦除操作
- ⌘ 读出/写入保护
- ⌘ 低功耗模式

## 1.2 闪存模块组织

按照不同容量，存储器组织成32个1K字节/页(小容量)、128个1K字节/页(中容量)、256个2K字节/页(大容量)的主存储器块和一个信息块，见表1、：

表1 闪存模块组织(小容量产品)

块	名称	地址范围	长度(字节)
主存储器	页0	0x0800 0000 – 0x0800 03FF	1K
	页1	0x0800 0400 – 0x0800 07FF	1K
	页2	0x0800 0800 – 0x0800 0BFF	1K
	页3	0x0800 0C00 – 0x0800 0FFF	1K
	页4	0x0800 1000 – 0x0800 13FF	1K
	·	·	·
	·	·	·
·	·	·	
	页31	0x0800 7C00 – 0x0800 FFFF	1K
信息块	启动程序代码	0x1FFF F000 – 0x1FFF F7FF	2K
	用户选择字节	0x1FFF F800 – 0x1FFF F80F	16
闪存存储器接口寄存器	FLASH_ACR	0x4002 2000 – 0x4002 2003	4
	FLASH_KEYR	0x4002 2004 – 0x4002 2007	4
	FLASH_OPTKEYR	0x4002 2008 – 0x4002 200B	4
	FLASH_SR	0x4002 200C – 0x4002 200F	4
	FLASH_CR	0x4002 2010 – 0x4002 2013	4
	FLASH_AR	0x4002 2014 – 0x4002 2017	4
	保留	0x4002 2018 – 0x4002 201B	4
	FLASH_OBR	0x4002 201C – 0x4002 201F	4
	FLASH_WRPR	0x4002 2020 – 0x4002 2023	4

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/558064064061006077>