

Kinetis K 和 L 系列 MCU 的生产用 Flash 编程最佳做法

作者: Melissa Hunter

1 简介

本应用说明描述对 Kinetis K 系列和 L 系列器件进行生产用 flash 编程的建议步骤。用户可以选择不同的物理接口，与 MCU 之间通信对 flash 编程，还有多条命令可用于修改 flash 内容。

本应用说明将讨论不同的物理接口、flash 命令，并提供推荐的操作顺序，以尽量减少 flash 编程过程中出现的问题，并缩短 flash 编程时间。本说明适用于开发生产用 flash 编程硬件、生产用 flash 编程软件或用于支持 flash 编程的调试器接口的任何用户。虽然本文档中的某些信息可能对软件控制的现场固件更新有参考价值，但本文档并未明确阐述此主题。

2 Kinetis flash 类型

Kinetis K 和 L 系列器件均使用 Freescale 薄膜存储器(TFS) flash；但是，不同的器件使用的 TFS flash 类型有所不同。根据具体 Kinetis 器件的不同，flash 可能是 FTFA、FTFL 或 FTFE 模块。

下表展示了 flash 模块之间的差异。

表 1. Flash 模块类型的比较

	FTFA	FTFL	FTFE
--	------	------	------

下一页继续介绍此表...

内容

1	简介.....	1
2	Kinetis flash 类型.....	1
3	编程接口.....	2
4	编程步骤.....	3
5	其他编程考虑因素.....	7
6	常见问题.....	8
7	扩展阅读.....	9
8	修订历史记录.....	9

表 1. Flash 模块类型的比较 (继续)

对应器件	Kinetis L 系列	Kinetis K 系列处理器，具有 512KB 或更少的 flash	Kinetis K 系列处理器，具有 512KB 或更多的 flash
编程命令	编程长字（32 位）	编程节和编程长字（32 位）	编程节和编程短语（64 位）
是否支持 EEE?	否	是，部分器件支持	是，部分器件支持
是否支持交换?	否	是，部分器件支持	是，部分器件支持
Flash 总线接口	握手—MCM_PLACR[ESFC]位可用于在 flash 忙时自动停止 flash 读取。	等待状态—读取忙碌时的 flash 数据块时会导致错。当其他数据块正在处理命令时，可以读取空闲的数据块。	等待状态—读取忙碌时的 flash 数据块时会导致错。当其他数据块正在处理命令时，可以读取空闲的数据块。

3 编程接口

有两个主要物理接口可用于 Kinetis 上的生产用 flash 编程—SWD/JTAG 和 EzPort。在同时支持两种接口的器件上，将在处理器上多路复用 SWD/JTAG 引脚和 EzPort 引脚。在复位时将采样 EZIP_CS 信号的状态，以确定引脚上使能的是 SWD/JTAG 引脚功能还是 EzPort 引脚功能。如果在复位时 EZIP_CS 为高电平，则引脚上使能的是 SWD/JTAG 功能。如果在复位时 EZIP_CS 为低电平，则引脚上使能的是 EzPORT 引脚功能。

3.1 SWD/JTAG（串行线调试/联合测试行动组）

Kinetis K 系列器件支持通过 SWD 和 JTAG 接口连接到 ARM 调试访问端口 (DAP)。Kinetis L 系列器件仅支持 SWD 接口。

SWD 和 JTAG 是不同的物理接口，但用于 flash 编程的命令和寄存器访问相同。为简单起见，本文档后续内容中将通篇使用术语 SWD，但描述的操作同时适用于 SWD 和 JTAG。

SWD 是处理器的主要调试接口。除了调试之外，SWD 还可用于 flash 编程。通常，flash 编程算法以及支持所需 flash 命令的软件将通过 SWD 加载到微控制器的片上 RAM 中，然后，SWD 将用于控制擦除 flash 以及从 RAM 中编程可执行代码。

SWD 也可用于直接写入 flash 寄存器，以针对所需命令对其进行配置。用 SWD 命令发送比在内核上本机执行代码慢，因此通常建议不使用这种方法，因为如果 SWD 直接写入 flash FCCOB 寄存器，flash 编程时间通常会明显延长。

3.2 EzPort

EzPort 是 Kinetis K 系列微控制器上的串行 flash 编程接口。EzPort 唯一的功能是连接至 flash。在 EzPort 模式下，处理器会自动停止，因此当 EzPort 激活时无法执行代码。

有一个可与 flash FCCOB 寄存器直接交互的 EzPort 命令，但对于大多数常用 flash 功能，EzPort 会设置自动管理 FCCOB 寄存器的命令。

有关 EzPort 和受支持的 EzPort 命令的完整详情，请参阅特定 Kinetis K 系列器件的参考手册。

4 编程步骤

以下各节描述编程 flash 的高级过程。

4.1 连接

flash 编程的第一步就是通过所需接口 (SWD 或 EzPort) 建立连接。

4.1.1 SWD 连接步骤

下面列示了通过 SWD 建立到处理器的连接的建议过程。

1. 在支持 EzPort 的器件上，确保 E_{ZP}_CS 为高电平，以保证使能 SWD/JTAG 引脚功能。
2. 给处理器通电，如果已通电，则将 $\overline{\text{RESET}}$ 引脚设为有效，以复位处理器。对于没有 $\overline{\text{RESET}}$ 引脚的器件，请在建立通信之后在 MDM-AP 控制寄存器中写入“系统复位请求”位。
3. 保持复位为低电平，建立与 ARM DAP 的通信。可以读取 MDM-AP ID 寄存器以验证连接是否正常。
4. 读取 MDM-AP 状态寄存器，直至“Flash 就绪”位置位。
5. 读取“系统安全”位以确定是否已使能安全性。如果该位 = 0，则继续操作。如果该位 = 1，如果不发出整体擦除命令或通过其他方法（后门密钥解锁）解除部件安全，将无法与处理器的内部（包括 flash）通信。
6. 写入 MDM-AP 寄存器以置位“调试请求”位。这能防止释放复位引脚时内核试图启动的情况。注意 如果处理器受保护，则无法写入“调试请求”位。
7. 将 $\overline{\text{RESET}}$ 信号置零或清除 MDM-AP 控制寄存器中的“系统复位请求”位。

完成上述步骤之后，即可开始执行调试或 flash 编程。

4.1.2 EzPort 连接步骤

下面列示了通过 EzPort 建立到处理器的连接的建议过程。

1. 确保 E_{ZP}_CS 为低电平，以保证使能 EzPort 引脚功能。
2. 给处理器通电，或者如果已通电，则将 $\overline{\text{RESET}}$ 引脚设为有效，以复位处理器。
3. 释放 $\overline{\text{RESET}}$ 并建立与 EzPort 的通信。
4. 发送“读取状态寄存器”命令以读取“Flash 安全”位 (FS)。如果 FS = 0，则继续操作。如果 FS = 1，如果不发出整体擦除命令或通过其他方法（后门密钥解锁）解除部件安全，将无法与处理器的内部（包括 flash）通信。

完成上述步骤之后，即可开始执行 flash 编程。

4.2 擦除 Flash

Flash 扇区必须首先进行擦除，之后才能进行编程。多种不同的 flash 擦除机制可供使用：

- 整体擦除 (SWD 或 EzPort)
- 擦除所有数据块
- 擦除 Flash 数据块
- 擦除 Flash 扇区

以下各节描述每个命令，以及生产用 flash 烧录器何时可使用各命令。

4.2.1 整体擦除

与上面列示的其他擦除机制不同，整体擦除不是使用 FCCOB 寄存器执行的 flash 命令。而是通过 SWD 或 EzPort 直接请求整体擦除。

即使使能了安全性或者区域受到了保护，调用整体擦除也会导致擦除整个 flash。如果使能了安全性，则必须同时使能整体擦除功能 (FTFx_FSEC[MEEN]不等于 0b10)，否则不会允许整体擦除。如果允许（使能）了整体擦除，将执行以下步骤：

1. 无论保护设置如何，flash 的所有数据块都会擦除。这包括程序 flash、数据 flash、程序 flash IFR 交换指示器地址、数据 flash IFR 空间（包括 EEE 分区信息）、EEPROM 备份存储器（E-flash）和 FlexRAM。
2. 读取 flash 验证擦除是否成功完成。
3. 如果擦除验证失败，则置位 FTFx_FSTAT[MGSTAT0]位，且进程停止。如果擦除验证通过，则进程继续。
4. FTFx_FSEC[SEC]寄存器字段设置为 0b10（不安全）。这将立即解除安全性。
5. “Flash 配置”字段中的安全字节也编程为 0xFE（不安全）。在后续复位时，除非修改了“Flash 配置”字段，否则处理器将以不安全模式启动。

4.2.1.1 SWD 整体擦除

下面列示了通过 SWD 接口执行整体擦除的建议过程。

1. 遵循“SWD 连接步骤”一节中描述的步骤执行操作。
2. 读取 MDM-AP 状态寄存器的整体擦除使能位，以确定是否使能了整体擦除命令。如果整体擦除使能 = 0，则禁用了整体擦除，无法擦除处理器或者解除处理器安全。如果整体擦除使能 = 1，则可以使用整体擦除命令。
3. 写入 MDM-AP 控制寄存器以置位“正在执行 Flash 整体擦除”位。这将启动整体擦除进程。
4. 读取 MDM-AP 控制寄存器，直到“正在执行 Flash 整体擦除”位清除。
5. 当“Flash 整体擦除”位清除时，整体擦除已完成。只要整体擦除操作中的整体擦除验证部分成功完成，安全性就会解除。

在整体擦除执行过程中，务必确保处理器不仅未收到新的复位请求，而且未执行可能干扰整体擦除命令操作的任何代码。为了避免此问题，请在整体擦除期间通过保持 RESET 有效或保持“系统复位请求”位置位，以保证处理器处于复位状态。

4.2.1.2 EzPort 整体擦除

下面列示了使用 EzPort 执行整体擦除的建议过程。

1. 遵循“EzPort 连接步骤”一节中描述的步骤执行操作。
2. 发送“读取状态寄存器”命令以读取“批量擦除禁用”位 (BEDIS)。如果 BEDIS = 0，则使能了批量擦除，因此可进行整体擦除。如果 BEDIS = 1，则禁用了整体删除，且无法擦除处理器。
3. 发出“Flash 批量擦除 EzPort”命令。
4. 发送“读取状态寄存器”命令，直到“正在执行写入” (WIP) 位清除。
5. 当 WIP 位清除时，整体擦除已完成。只要整体擦除操作中的整体擦除验证部分成功完成，安全性就会解除。

在整体擦除命令执行过程中，务必确保处理器未收到新的复位请求。

4.2.2 擦除所有数据块

擦除所有数据块是一条执行与 SWD/EzPort 整体擦除相似操作的 flash 命令，但操作并非完全相同。与整体擦除相似，擦除所有数据块将擦除所有用户 flash 内容——即程序 flash、数据 flash、程序 flash IFR 交换指示器地址、数据 flash IFR（包括 EEPROM 分区配置）、EEPROM 备份存储器（E-flash）和 FlexRAM。整体擦除与擦除所有数据块命令之间有两大重要区别：

- 与整体擦除不同，擦除所有数据块考虑了保护设置。如果保护了任何 flash 或 FlexRAM 区域，则擦除所有数据块命令将中止。
- 如果验证完成，擦除所有数据块将通过将 FTFx_FSEC[SEC] 字段设置为不安全来解除安全性，但“flash 配置”字段中的安全字节不会编程为 0xFE。在后续复位时，除非重新编程了“flash 配置”字段中的 FSEC[SEC]，否则处理器将处于安全状态。

4.2.3 擦除 Flash 数据块

擦除 Flash 数据块命令可用于擦除整个 Flash 数据块。擦除所有 Flash 数据块命令可以擦除多个数据块（如果器件具有多个 flash 数据块），但是擦除 Flash 数据块命令只能擦除单个数据块。如果数据块内的任何区域受保护，擦除就会中止。另外，如果 EEPROM 已分区且要擦除的数据块为数据 flash，擦除也会中止。当前安全设置不受擦除 Flash 数据块命令执行的影响，但如果擦除了“flash 配置”字段，则除非重新编程了“flash 配置”字段中的 FSEC[SEC] 设置，否则在后续复位时处理器将处于安全状态。

注

包含单个 flash 数据块的器件不支持擦除 Flash 数据块命令。如果器件上只有一个 flash 数据块，则擦除所有 Flash 数据块命令提供与未实施的擦除 Flash 数据块命令相同的功能。

4.2.4 擦除 Flash 扇区

顾名思义，擦除 Flash 扇区命令用于擦除整个 Flash 扇区。擦除 Flash 扇区命令每次擦除最少量的存储器。扇区的大小由所用 flash 模块和器件上 flash 数据块配置所决定。请参考 Kinetis 器件参考手册中“芯片配置”一章中的“Flash 存储器配置”节，以确定扇区大小。

如果选择的扇区受保护，则擦除 Flash 扇区命令将中止。如果正在使用交换，则隐含保护包含交换指示器（每半个程序 flash 存储器空间中一个）的扇区，除非交换系统处于更新模式且正在擦除的扇区处于不活动的一半中。

4.2.5 擦除建议

尽管有多个选项用于擦除 Kinetis 上的 Flash，但是大多数 flash 产品程序员只需要实现整体擦除和扇区擦除。EzPort 仅直接支持整体擦除（批量擦除 EzPort 命令）和扇区擦除。可通过使用写入 FCCOB 命令在 EzPort 上实施其他擦除指令，但并非必要。

4.2.5.1 整体擦除建议

整体擦除有一个重要特性，它考虑到了未加密部分。所有 flash 烧录器应该为支持的硬件接口（SWD 或 EzPort）提供整体擦除请求执行支持。如果在连接时检测到器件受保护，将向用户发出提示，询问他们是否想要整体擦除整个 flash 以解除安全性。如果检测到设备处于安全状态，自动发出一个包括整体擦除请求选项的请求，但是 Freescale 建议默认情况下不自动进行整体擦除。这样，用户会收到可能存在问题的警告，如果 flash 中已有需要保留的信息，用户可以选择终止操作。

整体擦除会导致 flash 中的所有用户信息丢失，因此建议 flash 烧录器默认情况下不要使用此选项。如果整体擦除专门用于擦除过程，则无法支持任何类型的多阶段编程（用户在不同时间对多个文件编程的情形）。Freescale 建议默认情况下使用扇区擦除，但为用户提供可以选择是否使用整体擦除的选项。

4.2.5.2 扇区擦除建议

为了避免丢失预编程的信息和配置，生产用 flash 烧录器默认情况下应使用扇区擦除命令。首选方法是在扇区上执行擦除以确保它真正处于已擦除状态，对该扇区编程，然后根据需要移至下一扇区。这样只擦除扇区（如果需要），即使擦除上部扇区或对整个扇区编程时出现导致整个操作中止的错误，也仅有最少量的 flash 经过修改。

4.3 编程 flash

擦除了 flash 扇区（或整个 flash）之后，可开始进行编程。器件可能支持三种不同的 flash 编程命令：

- 编程节
- 编程长字（32 位）
- 编程短语（64 位）

仅有部件器件支持编程节命令。处理器将支持编程长字或编程短语命令，但不同时支持。有关更多信息，请参考 flash 模块概述表。

4.3.1 编程节

编程节命令能够编程的区域对比利用常规程序长字命令和程序短语编程的区域更大。节小于 flash 扇区，其大小要么是 FTFL 器件的 FlexRAM/编程加速 RAM 的一半，要么是 FTFE 器件的 FlexRAM/编程加速 RAM 的四分之一。扇区大小和 FlexRAM 大小可能因器件的不同而各异。请参考所用特定 Kinetis 器件的参考手册以确定大小。由于编程节支持对更大的存储器进行编程，所以它是对存储器编程的最高效方法。这是用于生产编程的推荐编程命令（可用时）。

为了通过 SWD 使用编程节命令，将编程的数据写入 FlexRAM/编程加速 RAM（带 FlexMemory 的命令将 RAM 称作 FlexRAM，仅带程序 flash 的 RAM 将 RAM 称作编程加速 RAM）。FlexRAM 必须配置为传统 RAM 模式而不是 EEPROM 模式。数据加载到 FlexRAM 中之后，可以执行编程节命令。

EzPort 将自动在编程中使用编程节命令。进入 EzPort 模式时，EzPort 会将 FlexRAM/编程加速 RAM 配置为传统 RAM 模式。如果 RAM 模式更改为 EEPROM，则在发送节编程 EzPort 命令之前必须将 RAM 改回传统 RAM 模式，否则将不会接受该命令。

4.3.2 编程长字/短语

编程长字或编程短语命令可用于在 flash 中进行 32 位或 64 位编程。与将要编程的数据直接写入 flash FCCOBn 寄存器中的编程节命令不同，这些命令不会在编程中使用 FlexRAM/编程加速 RAM。

FTFA 器件是唯一不支持 EzPort 或编程节命令的部件。在 FTFA 部件上，编程长字命令是用于编程的唯一选项。

4.3.3 编程建议

为了加快编程速度，建议使用编程节命令作为 flash 生产编程方法。而在 Kinetis L 系列上，必须使用编程长字。这可能减慢 flash 编程速度，但由于 Kinetis L 系列器件上的 Flash 密度小于 K 系列部件上的密度，因此总的 Flash 编程时间仍应保持在合理范围内。

执行任何 flash 命令时，均应始终轮询适当的状态寄存器等待命令完成。对于新器件，命令执行时间几乎相同，但依靠固定的延迟循环而非状态可能造成问题。相反，应该轮询 flash FSTAT[CCIF]位或 EzPort 状态寄存器的 WIP 位，以确定命令是否已完成，然后再移至下一命令。

5 其他编程考虑因素

在编程 flash 时，应考虑 flash 内的一些特殊位置以及一些 flash 特点。以下各节提供有关 flash 烧录器应特殊对待的 flash 地址和特点的信息。

5.1 Flash 配置字段

flash 配置字段是 flash 存储器中位于 0x400-0x40F 的 16 字节部分。flash 配置字段中的值在复位时复制到 flash 寄存器，以便为多个 flash 设置设定默认值，最重要的是 flash 保护和系统安全设置。

为了防止器件的意外保护，flash 配置字段需要特殊处理。Freescale 建议，在默认情况下 flash 烧录器将以下值写入 flash 配置字段：

0xFFFF_FFFF

0xFFFF_FFFF

0xFFFF_FFFF

0xFFFF_FFFE

此值让器件处于解除安全状态，不保护任何 flash 区域。

一些用户会希望对 flash 配置字段编程，因此应包括覆盖选项，让用户能够将其他值编程到 flash 配置字段中。请注意，对 flash 配置字段的更改将在下次复位时生效。为了防止更新的 flash 配置字段值保护处理器并防止进一步访问处理器，请在复位处理器之前，确保烧录器完成所有编程步骤以及对写入值的任何验证。

5.2 增强型 EEPROM 支持

带 FlexNVM 的 Kinetis 器件包括 FlexMemory 功能。FlexMemory 让用户能够将某些片上 flash 存储器配置为增强型 EEPROM、附加 flash 存储器或两者的组合。

默认情况下，空白器件将所有 FlexNVM 配置为 flash 存储器。为了让客户能够配置 EEPROM 功能以及可能预编程将存储在 EEPROM 存储器中的起始值，生产用 flash 烧录器将需要包括对配置 EEPROM 的支持。可通过 SWD/JTAG 或 EzPort 接口支持 EEPROM 配置。

5.2.1 FlexMemory 分区

要配置将使用的 EEPROM 存储器数量以及将用于备份 EEPROM 的 flash 数量，必须对 flash 分区。Flash 程序分区命令 (PGMPART) 用于设置存储器分区和配置要使用的 EEPROM。EzPort 不直接支持运行 flash PGMPART 命令，因此 EzPort 的“写入 FCCOB 寄存器”命令将用于执行 PGMPART 命令。

PGMPART 命令执行时会擦除所有 FlexNVM (甚至包括并非用于 EEPROM 支持的扇区)。擦除 FlexNVM 之后，用于 EEPROM 备份的扇区将格式化以便使用。因为在分区之后擦除了 FlexNVM 且 FlexNVM 的一些部分不再可访问，所以如果要使用 EEPROM 功能，EEPROM 配置应是器件编程中的第一步。在配置和可选加载 EEPROM 之后，可以将代码和数据编程到 flash 的其余部分。因为分区影响可用于常规编程的 flash 数量，所以最好从器件读取并保存分区代码，这样，由于 flash 分区，传统 flash 地址不再适用，如果用户对这些地址编程，则发出警告。

注

应只执行一次分区。如果为不同的配置对 flash 反复分区，则记录数据将会丢失，且不保证您能获得预期的耐久性。

注

如果整体擦除器件，分区信息、EEE 数据和 EEE 位置使用信息将会丢失。如果在整体擦除之前使用了 EEE，这可能影响 EEE 的最长耐久性。

5.2.2 编程 EEPROM

对 flash 分区之后，可通过写入从 0x1400_0000 开始的 FlexRAM 存储器空间，对初始 EEPROM 值编程。使用 SWD 时，可以通过直接存储器写入完成此操作。EzPort 包括“写入 FlexRAM”命令，可用于写入 EEPROM 值。

注

当试图对 EEPROM 数据编程时，必须为 EEPROM 模式配置 FlexRAM。在试图使用编程节命令之前，务必将 FlexRAM 模式切换回传统 RAM 模式。

5.3 交换支持

部分 Kinetis 器件包括交换功能，能够交换 flash 数据块的存储器基地址。这让用户能够拥有两个可以启动的代码映像，并让交换系统能够确定哪个映像在执行。

交换功能通常用于稳定更新。理念是将新软件映像下载到不活动的 flash 空间，不触及目前正在运行的软件映像。如果新固件成功下载，则完成交换，处理器下次启动时，新固件将运行。如果在新映像中检测到任何错误，则可忽视它，当前软件不会受到任何影响，仍能继续正常启动。

因为交换的典型功能是应用于固件更新，因此应用于交换的系统通常需要支持在固件内部执行。这意味着大多数生产用 flash 烧录器通常无需直接支持 flash 交换功能。然而，在大多数的交换状态下，如果启用了交换，则隐含地保护 flash 交换指标区间。所以如果编程之前已经配置好准备交换的部分，当试图擦除包含交换的 flash 指标区间时，可能出现特权违规。当对器件重新编程时，可能需要进行整体擦除以清除所有交换配置。

5.4 并行化

为了提高编程吞吐量，可使用 SWD/JTAG 或 EzPort 对多个 Kinetis 器件并行编程。同时连接至多个器件时，可以共享作为处理器输入的控制信号，但应该监控各处理器上每个引脚的输出信号。为了保证烧录器可分别监测每个器件上的命令完成情况，这种措施必不可少。不能盲目假设多个处理器将在同一时间完成命令的运行，因此需要跟踪每个处理器的状态。此外，还要为来自处理器的输出使用唯一的引脚，以便能够在编程之后验证 flash 内容。

表 2. 并行化引脚连接

接口	共享的引脚	各处理器的每个引脚
SWD	SWD_CLK、SWD_DIO（输入）、复位	SWD_DIO（输出）
JTAG	JTAG_TCLK、JTAG_TMS、JTAG_TDI、复位	JTAG_TDO
EzPort	EZP_CK、EZP_CS、EZP_D、复位	EZP_Q

6 常见问题

下面的列表描述了已知会导致编程问题的的问题。

- 未保护 flash 配置字段以防止意外和/或有意使用错误值（考虑字节序）对 flash 配置字段编程。如果使用使能了安全性、禁用了整体擦除和后门密钥访问的值对 flash 配置字段编程，则无法进行 flash 编程。

- 擦除或编程命令的中断可能导致 flash 内容毁坏。中断可能包括复位、掉电或与处理器上运行的代码冲突。
- 未使用“调试请求”位防止代码在对 flash 编程时执行。如果内核在建立调试连接或修改 flash 时试图运行代码，则可能干扰代码。如果没有编程任何代码（空白部件），则由于内核锁定，处理器将定期复位。如果在器件中编程了修改电源模式、时钟或 SWD/JTAG 引脚的配置的代码，当 flash 命令正在执行时可能防止连接以及/或导致 flash 毁坏。
- 在处理器未通电时给 I/O 引脚施加电压。如果违反引脚 VDIO 或 VIO 规范，处理器可能试图部分通电以及/或使 flash 进入未定义状态。这可能导致 flash 内容毁坏、flash 控制逻辑毁坏或器件配置和调整值毁坏，从而可能导致处理器报告为受保护（锁定的器件）或处理器故障以响应和完成 flash 命令。

7 扩展阅读

可能有用的文档如下：

- 可以通过 <http://www.freescale.com/kinetis> 获取 Kinetis 微控制器的相关文档、软件和工具。
- 芯片配置、Flash 存储器模块以及适用器件的参考手册的 EzPort 章节
- *Using the Kinetis Security and Flash Protection Features* (AN4507)
- 使用 Kinetis 系列中的增强型 EEPROM 功能 (AN4282)
- 利用 Kinetis 微控制器的 Flash 存储器交换功能实现稳健的在线固件更新 (AN4533)

8 修订历史记录

表 3. 修订历史记录

修订版本号	更新描述
0	初始版本
1	更改了对“调试请求”位的核心保持功能的引用。

How to Reach Us:**Home Page:**freescale.com**Web Support:**freescale.com/support

本文档中的信息仅供系统和软件实施方使用 Freescale 产品。本文并未明示或者暗示授予利用本文档信息进行设计或者加工集成电路的版权许可。Freescale 保留对此处任何产品进行更改的权利，恕不另行通知。

Freescale 对其产品在任何特定用途方面的适用性不做任何担保、表示或保证，也不承担因为应用程序或者使用产品或电路所产生的任何责任，明确拒绝承担包括但不限于后果性的或附带性的损害在内的所有责任。

Freescale 的数据表和/或规格中所提供的“典型”参数在不同应用中可能并且确实不同，实际性能会随时间而有所变化。所有运行参数，包括“经典值”在内，必须经由客户的技术专家对每个客户的应用程序进行验证。

Freescale 未转让与其专利权及其他权利相关的许可。Freescale 销售产品时遵循以下网址中包含的标准销售条款和条件：freescale.com/SalesTermsandConditions。

Freescale, the Freescale logo, and Kinetis are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. All other product or service names are the property of their respective owners.

© 2013-2014 Freescale Semiconductor, Inc.

© 2013-2014 飞思卡尔半导体有限公司