Freescale Semiconductor
应用笔记

Document Number: AN4533
Rev 0, 06/2012

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

作者: Maclain Lobdell
汽车、工业化与跨市场解决方案小组

内容

1 简介 1
2 应用要求 1
3 程序 Flash 交换概述 3
4 交换步骤 6
5 交换详情 8
6 多任务应用程序概述 11
7 软件示例 11
8 结论 18
9 参考资料 18

1 简介

Kinetis 微控制器上的程序 Flash 交换功能为远程在线系统软件更新提供了一种稳健途径。本应用笔记将描述如何利用该交换功能。本文将讨论常见问题并提供一个应用程序示例。

本应用笔记面向应用 Flash 交换功能的应用程序的系统开发人员。本应用笔记会提及在线通信的方法，但这不是本文讨论的重点。

关于 Flash 交换的更多重要信息，请参见本应用笔记中所讨论具体器件的参考手册。

2 应用要求

许多应用都需要可靠的在线系统更新方法。这就要求具备一种机制，以便远程更新系统软件/固件来修复漏洞并进行重要软件升级。重要的应用（如电气计量）在更新时要求极短的系统停机时间，必须能耐受通信错误，并且不应存在产生导致系统停止工作的问题的风险。更新过程必须能够适应各种困难环境。许多应用的电源都不够可靠，系统更新过程必须能在可能断电的情况下继续工作。

远程更新系统软件需要通过通信接口（例如：Wi-Fi、Zigbee®、UART、SPI、I2C、USB 和以太网等）传输新代码。在系统上执行的软件上传程序必须能将代码接收并编程到本地存储器中。通常，出厂编程是通过 JTAG 或后台调试模式（BDM）等本地调试连接完成的，但这些连接无

法远程实现。从经济角度考虑，派技术人员到系统所在地点进行现场重编程也是不可行的做法。因此，可靠的在线更新方法必不可少。

2.1 远程固件更新系统

2.1.1 传统的远程更新系统

传统的远程固件更新系统依靠引导加载应用，其在系统复位时执行，然后选择要运行的应用程序或执行应用程序更新程序。远程服务器将新固件传送到系统，并编程到其本地存储器中。

大多数情况下，当系统更新开始后，主应用会暂停运行。随后，主应用代码将被擦除并重新编程。主应用仅有一个副本，因此如果在接收并编程新代码时出现未检测到的错误，系统可能无法正常工作，除非下载新应用，否则可能停止运行。最坏情况是系统无响应，而且无法强制进入用于系统更新的引导加载模式。

优势
- 易于实现
- 无需保存应用的备份副本，因此所需 Flash 存储器空间更少

劣势
- 在更新过程中必须停止主应用
- 无法恢复到已知可以正常工作的应用
- 在更新过程中可能无法承受断电

2.1.2 具有代码备份的系统

在系统的存储器中可以设置完整的软件备份。如果检测到严重错误，系统可以恢复到主应用任务的备份副本。可使用引导加载程序选择要执行的主应用的正确副本。

优势
- 在多任务处理操作系统中，当后台任务在更新应用的新副本时，主应用任务可以继续执行。
- 代码的备份副本。可以恢复到已知可以正常工作的应用程序。

劣势
- 需要额外的存储器空间来存放备份副本。
- 需要引导加载程序（用于选择要运行的应用的启动程序）。

2.1.3 使用 Flash 存储器交换的系统

在具有两个或更多支持交换的内部 Flash 存储块的器件中，每个 Flash 存储块的存储器基地址可进行交换。因此，每个 Flash 存储块的地址位置都会在器件的逻辑存储映射中进行交换。复位后，内置 Flash 交换系统会通过逻辑存储器映射中的 Flash 存储块位置来选择要执行的软件。这使得代码备份系统的编程更为容易。您可以在一个存储块中执行，而同时擦除/编程另一个存储块。在 Kinetis 器件上，Flash 交换系统监视/控制应用程序新旧切换的所有步骤；这进一步确保了系统运行的可靠性，以防在这些过程中发生断电情况。

优势
- 易于编程。应用总是在存储器映射的低地址存储块中执行。
- 可以承受断电。
- 无需引导加载程序。主应用启动无延迟。

- 适用于多任务操作系统。最小化应用停机时间。在多任务处理系统中，当后台任务在更新应用的新副本时，主应用可以继续执行。
- 代码的备份副本。可以恢复到已知可以正常工作的应用程序。

劣势

- 需要额外的 Flash 存储器空间来存放备份副本。

3 程序 Flash 交换概述

Kinetis 器件提供用于程序和数据存储的内部 Flash 存储器。对于具有两个或更多程序 Flash（p-flash）存储块并且具有 p-flash 交换特性的器件，系统编程人员可对 p-flash 空间的逻辑存储器映射进行配置，这样两个物理 p-flash 存储块中的任一个均可存在于相对地址 0x0000 处。Flash 交换特性并非强制。

由于 Kinetis 器件的默认向量表位于地址 0x0000 处，因此这一点尤为有用。当处理器退出复位时，它从地址 0x0000 处取得初始堆栈指针（SP），并从地址 0x0004 处取得程序计数器（PC）。因此，交换两个 p-flash 存储块的基地址使得系统可以从 p-flash 存储块 0 启动，也可以从 p-flash 存储块 1 启动，因为两个存储块均可定位在基地址 0x0000 处。

交换系统由交换控制命令控制，这与 Kinetis 器件上的其他 Flash 命令相似。必须首先初始化交换系统，之后才能执行交换。

3.1 关于 p-flash 交换的重要事实

- 当前位于相对地址 0x0000 的存储块称为有效存储块。也称为低地址存储块。其他存储块为无效存储块或高地址存储块。
- 典型的用法是系统在一个存储块（有效存储块）中执行，同时对另一个存储块（无效存储块）进行重新编程。在大多数应用中，系统应用代码限制在一个 Flash 存储块（有效存储块）中。
- 在重新编程无效存储块之后，且执行就绪时，用户可以完成交换过程以交换存储块。此时，无效存储块变为有效存储块，反之亦然。
- 在系统复位后，代码从有效存储块（位于地址 0x0000 处的存储块）处开始执行。
- 在存储块之间来回交换的过程是相同的。

3.1.1 Kinetis（100 MHz）交换的详细说明

Kinetis 100 MHz 器件具有两个 p-flash 存储块，支持两个存储块的交换。FTFL_FCNFG 寄存器的 SWAP 位指示了位于地址 0x0000 处的 p-flash 存储块。在复位过程中，Flash 模块中的交换系统会将 SWAP 标志的状态置位。

SIM_FCFG2 寄存器中的 SWAPPFLSH 位指示交换系统是否已完成初始化，即可以进行交换。

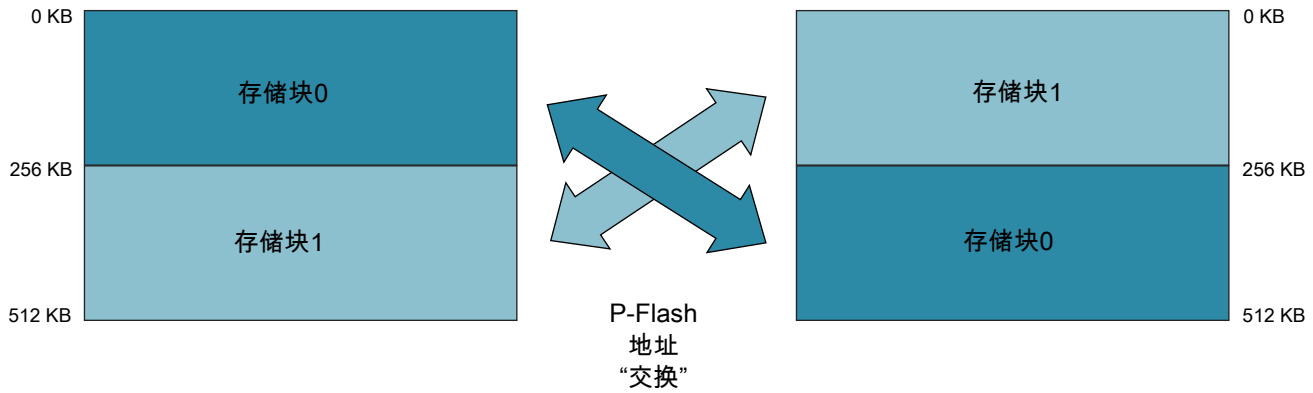


图 1. P-flash 交换（Kinetis 100 MHz）

3.1.2 Kinetis（120/150 MHz）交换的详细说明

Kinetis 120/150 MHz 器件具有两个支持交换的 Flash 配置选项。下面将分别讨论这两个选项。

FTFL_FCNFG 寄存器的 SWAP 位指示了位于地址 0x0000 处的 p-flash 存储块/半存储块。在复位过程中，Flash 模块中的交换系统会将 SWAP 标志的状态置位。

3.1.2.1 512 KB p-flash 选项

具有两个 p-flash 存储块（512 KB p-flash）的 Kinetis 120/150 MHz 器件，支持两个 p-flash 存储块之间的交换。这与 100 MHz Kinetis 器件（512 KB p-flash）相似。

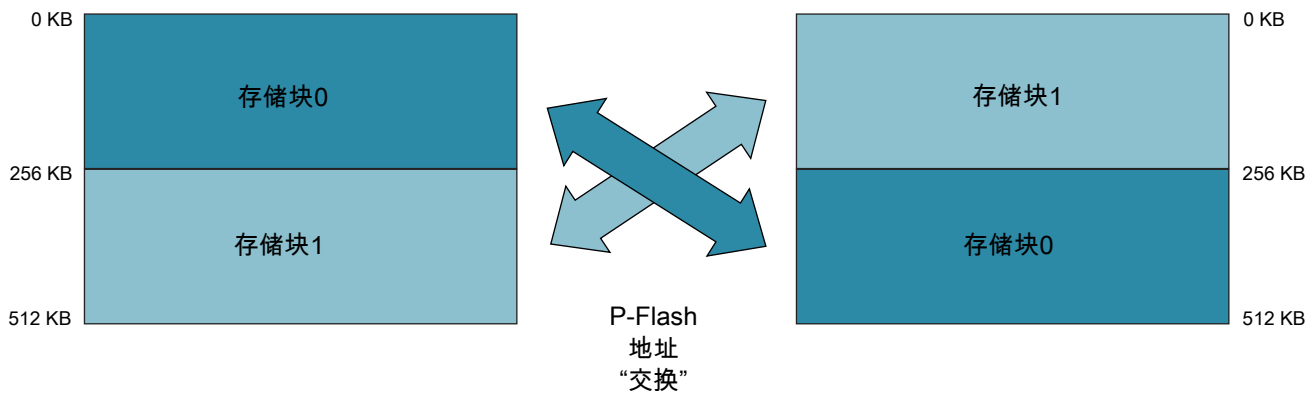


图 2. P-flash 交换（Kinetis 120/150 MHz，512K p-flash）

3.1.2.2 1 MB p-flash 选项

具有四个 p-flash 存储块（1MB）的 Kinetis 120/150 MHz 器件，支持两个双存储块（p-flash 存储块 0-1 和存储块 2-3）之间的交换。

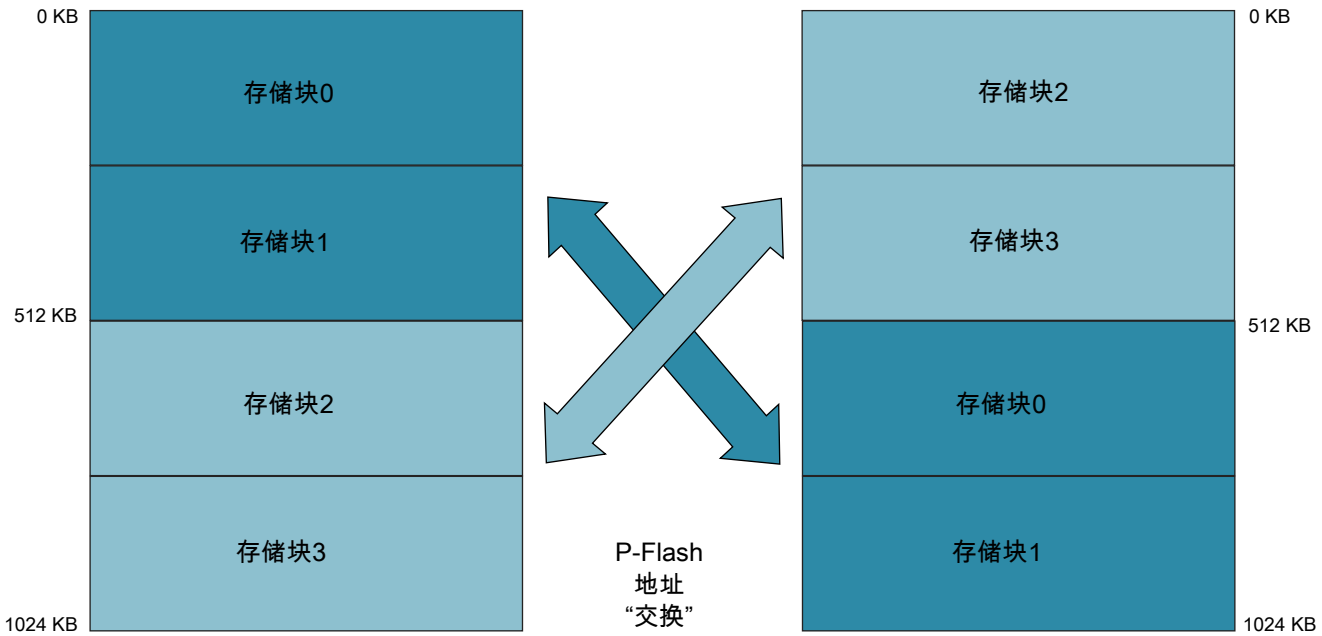


图 3. P-flash 交换（Kinetis 120/150 MHz，1M p-flash）

3.2 交换命令概述

共有四种交换命令。

1. 设置交换指示器地址（初始化交换系统）
2. 在更新（准备）模式中设置交换
3. 在完成模式中设置交换
4. 报告交换状态

设置交换指示器地址命令

该命令用于初始化交换系统。当发出该命令时，您必须提供用于 Flash 交换指示器的地址（将在后续讨论）。该命令设置了交换系统所使用的交换使能字和交换指示器地址。只需要在第一次执行交换时使用该步骤。

设置交换为更新状态

该命令对 Flash 交换指示器值进行编程，指示系统计划更新高地址存储块中的内容。此外，该命令对无效存储块中存放 Flash 交换指示器的扇区取消保护，使其可以进行擦除。在更新（或更新-擦除）状态期间擦除该扇区是交换过程的必须步骤。

设置交换为完成状态

该命令对 Flash 交换指示器值进行编程，告知系统对无效存储块的重新编程已完成，已准备好交换 Flash 存储块。在无效（高地址）存储块中的内容被擦除/重新编程后需要执行该命令，从而更新系统软件。这包括擦除无效存储块中包含 Flash 交换指示器的扇区。

报告交换状态

该命令用于检查交换系统的状态。其返回当前的交换状态、当前的交换存储块状态（当前位于地址 0x0000 的存储块）、下一个交换存储块状态（系统复位后将位于 0x0000 的存储块），以及是否发生错误。

建议将该命令置于在系统复位时执行的系统初始化代码中。这将有助于系统软件识别交换系统中是否存在因交换命令执行过程中断电而导致的错误。

更多详情，请参见错误处理 。

3.2.1 从内部 SRAM 执行交换命令

交换命令必须从 SRAM 程序中执行，以避免发生读取时写入冲突。如果对正在执行代码的 Flash 存储块中进行擦除或编程操作，则可能发生读取时写入冲突。由于 Flash 交换系统会对两个 Flash 存储块的 Flash 交换指示器进行编程，因而当执行交换命令时，有必要保证未在 Flash 中执行代码。

更多详细信息，请参见本应用笔记随附的示例软件。

3.2.2 传递给所有交换命令的地址

所有交换命令都需要由初始化命令设置的有效存储块的交换指示器地址。各交换命令必须使用相同的地址。无论您是从存储块 0 交换到 1，或是从存储块 1 交换到 0。

更多详情，请参见 Flash 交换指示器 。

4 交换步骤

4.1 交换过程

交换 Flash 存储块的过程很简单。在存储块之间来回交换的过程是相同的。

4.1.1 首次交换

1. 通过执行初始化命令对系统进行首次初始化。只需要在第一次执行交换时使用该步骤。在第一次初始化时，交换系统直接进入到更新-擦除状态。
2. 擦除无效（高地址）存储块。
3. 将新软件重新编程到无效（高地址）存储块中。
4. 执行命令将系统设置为完成状态。交换在复位（包括软件复位）后生效。
5. 复位后，存储块进行交换，交换系统变为就绪状态。

4.1.2 其他交换的流程

在首次交换完成后，交换过程从就绪状态开始。

1. 执行命令将系统设置为更新状态。
2. 擦除无效（高地址）存储块。一旦擦除完成，系统将自动进入更新-擦除状态。
3. 在无效存储块中重新编程软件。
4. 执行命令将系统设置为完成状态。
5. 复位微控制器（任意复位，包括软件复位）。
6. 复位后，存储块进行交换，交换系统变为就绪状态。

4.2 擦除无效的高地址存储块

您可以使用存储块擦除或扇区擦除。只是需要擦除无效存储块中的 Flash 交换指示器扇区。然而，要在无效存储块中更新软件，就必须先进行擦除。一旦擦除完成，系统将自动进入更新-擦除状态。

可以在存储块之间交换而不擦除/重新编程整个无效存储块。您只需擦除具有 Flash 交换指示器（在更新状态下）的扇区。这在交换回无效存储块中已知正常的应用程序时很有帮助。

更多详情，请参见 Flash 交换指示器 。

4.3 步骤的顺序

飞思卡尔建议在交换系统处于更新-擦除状态及系统进入完成状态之前，将新代码上传到无效（高地址）存储块中。这样，如果在过程中发生断电，交换系统将知道断电是在更新途中发生，应当恢复到先前的、已知正常的交换状态。

执行 Flash 交换命令的顺序也很重要。命令必须按照顺序执行，否则会发生错误。详情请参见章节错误处理 。

4.4 典型交换流程小结（首次交换后）

1. 检查状态
2. 就绪->更新
3. 擦除高地址存储块（或仅擦除无效的高地址存储块中 Flash 交换指示器所在的扇区）
4. 更新->更新擦除（自动）
5. 重新编程高地址存储块——检查高地址存储块中的内容（编程时跳过交换指示器位置）
6. 交换更新-擦除->完成
7. 复位
8. 完成->就绪（自动）

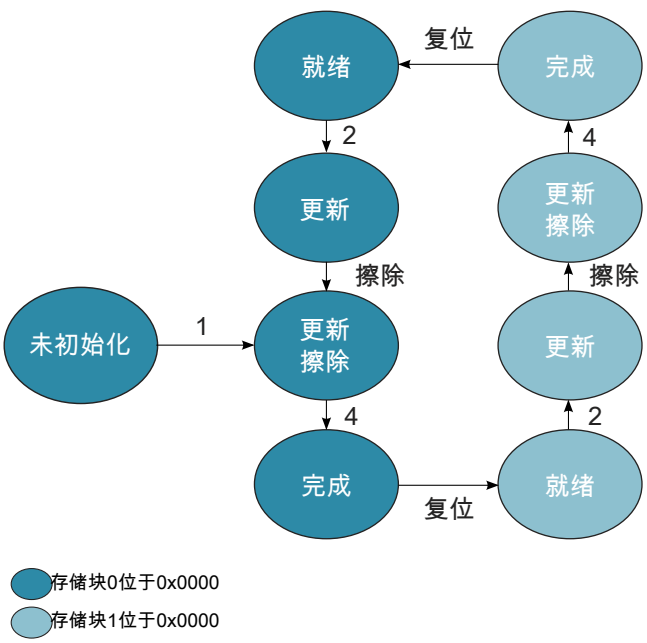


图 4. Flash 交换（Kinetis 120/150 MHz，1M p-flash）

5 交换详情

5.1 Flash 交换指示器

Flash 交换指示器（指示器 0 和指示器 1）位于 p-flash 存储器空间中的两个不同位置。一个位于整个 p-flash 的下半部分，另一个位于上半部分。这两个 Flash 交换指示器所在地址相对于每一半 p-flash 的基地址的偏移量相同。系统编程人员在初始化 Flash 交换系统时会指定 Flash 交换指示器的位置。

Flash 交换指示器的用途是什么？

交换系统使用 Flash 交换指示器来存放系统使用的代码，以确保在交换运行过程中出现潜在的断电情况时也能保持可靠工作。在交换过程的每个步骤中，交换系统采用指定的代码对 Flash 交换指示器进行编程。这为系统提供了一种跟踪机制，从而可以在系统复位时可靠地确定交换状态。

复位时，交换系统会询问交换指示器来确定交换状态，以及交换过程是否中断。您只需要在交换过程的更新状态期间擦除交换指示器即可。

Flash 交换指示器有多大？

每个 p-flash 存储块有 2 个字节

哪些内容应编程到 **Flash** 交换指示器中？

系统编程人员无需编程 Flash 交换指示器或详查其中的内容，Flash 交换系统会自动进行相应的处理。不过，参考手册中的交换状态报告映射表提供了编入交换指示器的代码的详细信息。该表对交换过程中每个步骤的代码以及可能检测到的错误（交换期间可能发生的断电）进行了说明。

Flash 交换指示器应放置在何处？

系统编程人员在初始化 Flash 交换系统时会指定 Flash 交换指示器的位置。可以使用除包含向量和 Flash 配置字段的头两个扇区外的任意 Flash 扇区。根据具体器件，交换指示器必须为 32 位、64 位或 128 位对齐。

例如，器件包含两个 256 KB 的 p-flash 存储块（存储块 0：0x00000-0x3FFFF 和存储块 1：0x40000-0x7FFFF）且扇区大小为 2 KB（0x800）。如果交换指示器放置在 Flash 存储块的最后一个扇区，Flash 交换系统将指定 0x3F800 为存储块 0 的交换指示器位置，指定 0x7F800 为存储块 1 的交换指示器位置。在本示例中，提供给交换系统的地址只有 0x3F800。

注

建议将 Flash 交换指示器置于 p-flash 存储块的最后一个扇区中。这能为应用代码提供最大的连续 Flash 区域。

Flash 交换指示器的地址（偏移量）存储在何处？

Flash 交换指示器的地址（相对于各存储块基地址的偏移地址）存储在称为程序 Flash 存储块 1 IFR 的 Flash 区域中。

可以改变 **Flash** 交换指示器的位置吗？

Flash 交换指示器的地址不能修改，除非执行擦除所有存储块命令，这会使交换系统返回未初始化状态（在支持的器件上可用）。因此，一定不要忘记在交换初始化期间提供给交换系统的地址。

Flash 交换指示器的保护

设定完成后，用户就不能再对 Flash 交换指示器的位置进行编程。此外，也不能擦除有效存储块中包含 Flash 交换指示器的扇区。只有当交换系统处于更新或更新-擦除状态时，才能擦除无效存储块中包含 Flash 交换指示器的扇区。不要在交换指示器上编程，否则会导致错误。

交换状态	可以擦除？	
	有效存储块：交换指示器扇区	无效存储块：交换指示器扇区
未初始化	是	是
就绪	否	否
更新	否	是
更新擦除	否	是
完成	否	否

图 5. 何时允许擦除 Flash 交换指示器扇区

交换状态	可以擦除？	
	有效存储块：所有扇区（交换指示器除外）	无效存储块：所有扇区（交换指示器除外）
未初始化	是*	是
就绪	是*	是
更新	是*	是
更新擦除	是*	是
完成	是*	是

图 6. 何时允许擦除除 Flash 交换指示器扇区以外的扇区

*除 Flash 交换指示器扇区以外，其他所有扇区都可以随时擦除。但是，不能擦除正在执行的存储块。在极少数情况下如果需要擦除有效存储块，可跳转到 RAM 中的子程序并从 RAM 执行。

5.2 交换使能字段（字）

交换使能字段也称为交换使能字，同样存放在程序 Fhasl 存储块 1 IFR 中。在执行交换初始化命令时，交换系统会对该位置进行编程。该位置的内容只是用于告诉交换系统交换已完成初始化，并且已经指定了有效的 Flash 交换指示器地址。

5.3 常见问题

交换为何如此复杂？——断电耐受性

在阅读交换文档后，您可能会产生这样一个疑问：“交换为何如此复杂？”答案就是断电耐受性。在不够稳健的系统中，如果在关键时刻断电，系统可能会崩溃。对于使用 Flash 存储器交换功能的系统，如果在交换过程中断电，内置系统可以检测到错误，并可靠地确定上一个已知的有效 Flash 存储块配置。

使用交换的系统是否仅限于在两个 **Flash** 存储块之一（占总 **Flash** 存储器大小的一半）中执行应用？

在大多数用例中，系统应用将限制在一个 Flash 存储块（有效存储块）中执行。然而，交换系统不会限制代码在无效存储块中执行。因此，代码可以随时在任意一个存储块中执行。

可以使用高级技术将应用大小扩展至大于 p-flash 一半的大小，并且继续使用 Flash 交换功能。这可能会牺牲一些对于问题防护功能，不过为代码提供了更大空间。此话题超出了本应用笔记的讨论范围。

存放 **Flash** 交换指示器的扇区是否可用于存放数据/程序？

通常，用户应指定一个未使用的 Flash 扇区来存放 Flash 交换指示器。然而，交换系统不会干扰包含指示器的 Flash 扇区中的其他数据。因此，该空间可用于存放程序或数据。该空间应相应地在链接器文件中指出。详情请参见重要注意事项 。

5.4 重要注意事项

始终编程向量表和 **Flash** 配置字段

在交换前，您必须为无效的高地址存储块编程中断向量表和 Flash 配置字段（FSEC 和 FOPT 等）。如果不这样做，交换后新的有效存储块中的向量表和 Flash 配置将无法正确设置。这会导致不符合期望的配置，并可能将器件加密锁死。

项目链接器文件的注意事项

代码中没有需要特别注意的问题。只需链接代码，通常就会填入低地址存储块。建议在链接器文件中为有效（低地址）的 Flash 交换指示器位置分配一个部分，以避免应用企图改写。

5.5 错误处理

有两种主要情况需要您检查错误。

5.5.1 在复位后执行交换状态命令选项

在复位后运行交换状态命令选项以报告交换状态，用于检测以下可能的情况。

1. 固件更新/交换期间可能发生了断电：交换状态未就绪

   在固件更新过程中发生复位。如果系统在复位后处于更新或更新-擦除状态，显然就表示出现了这种情况。除非正处于更新固件的过程中，否则系统通常将为就绪状态。

   如果发生了这种情况，则从当前交换状态开始继续交换过程。建议在无效存储块中擦除和重新编程固件，以确保完全编程。

2. 在交换命令期间可能发生了断电：错误标志置位

   在固件更新过程中发生复位，导致 Flash 交换指示器值损坏。在运行交换状态命令选项后，该情况由 FSTAT 寄存器中的 MGSTAT0 错误标志报告。交换系统设计为对该情况完全耐受。然而，需要清除 Flash 交换指示器。这可以通过运行一次交换流程完成。

5.5.2 在执行交换命令后检查错误标志

针对交换控制命令错误处理表中列出的错误来检查 FSTAT 寄存器中的错误标志 ACCERR 和 MGSTAT0，在器件参考手册中可以找到该表。不仅仅是交换命令，建议在执行任何命令后都进行错误检测。

6 多任务应用程序概述

在多任务操作系统中，主任务可以继续运行，而新固件可以通过独立的上传/编程任务进行加载。软件上传任务将交换系统设置为更新模式，接收新应用代码（通过通信接口），将代码编程到无效高地址程序 Flash 存储块中，然后设置交换系统完成交换。系统复位后交换 Flash 存储块，启动新应用。这保证了超短的应用停机时间。只要新应用还具有上传/编程/交换任务，就可以不断重复该过程。对应用的要求是大小小于程序 Flash 存储器的一半，对软件的要求是具有一个可以接收、编程新应用并执行交换的任务。

主应用任务

- 在新固件上传期间继续运行

上传任务

激活后，上传任务执行以下内容:

- 执行命令，将交换系统置于更新状态
- 擦除要重新编程的高地址存储块扇区
- 接收新的应用代码（通过 SCI 和 TCP/IP 等）
- 将新代码编程到高地址 Flash 存储块中
- 检查代码错误
- 执行命令，将交换系统置于完成状态
- 执行软件复位
- 系统复位后，Flash 存储块被交换，并且新应用启动
- 整个过程可以以几乎相同的步骤进行重复

7 软件示例

应用笔记提供的软件示例用于演示交换功能。该示例并非基于多任务操作系统。

7.1 软件组件

7.1.1 Flash 驱动程序软件

示例采用针对 Kinetis 微控制器的 Flash 驱动程序软件(C90TFS_FLASH_DRIVER)。驱动程序可以编译到嵌入式应用中，添加 Flash 控制功能。示例采用版本 0.2.9（beta），该版本更新了 Flash 交换支持。

7.1.2 Kinetis 示例代码

本示例基于针对 K60 100 MHz 版本（KINETIS512_SC）的 Kinetis 示例代码。示例中同时提供了 IAR 和 CodeWarrior 工程。在工程中添加了 Flash 驱动程序。

支持的集成开发环境（**IDE**）

- IAR Embedded Workbench for ARM 版本 6.30 或更高版本
- CodeWarrior for MCUs 版本 10.1

7.2 硬件

Kinetis K60 100 MHz MCU 模块（TWR-K60N512），要求版本标记为 4N30D 或更高的 MCU 器件。参见以下附注。

注

需要 Kinetis 100 MHz Rev.1.4 或更新版本。标记在封装上的掩膜组编号为 4N30D。后续修订版可能标记为 5N30D 等。早于 Rev.1.4 的修订版可能不支持交换，或者在交换初始化后禁止“擦除所有存储块”命令。因此，对于早期修订版，依靠“擦除所有存储块”命令来预擦除 Flash 的部分调试器可能无法重新下载到 Flash。此外，对于早期器件，可能无法使交换系统回到未初始化状态。

7.3 软件分析

7.3.1 Flash 驱动程序交换函数

Flash 驱动程序提供两个函数。

7.3.1.1 PFlashGetSwapStatus

位于文件 *PFlashGetSwapStatus.c* 中。该函数返回交换状态详情，并检查在交换状态确定过程中是否检测到任何错误。

7.3.1.2 PFlashSwap

位于文件 *PFlashSwap.c* 中。该函数执行交换控制命令。

7.3.2 回调函数

调用方应用可以向 PFlashSwap 函数传递回调函数。PFlashSwap 驱动程序函数在每次进入下一个交换状态后会调用该回调函数。当 PFlashSwap 函数调用回调函数时，会将交换状态传递回调用应用程序。利用该信息，应用可以决定是否继续进入下一个交换状态。

此外，这为执行固件更新提供了一个绝佳途径。当交换驱动程序向回调函数报告交换系统处于更新状态时，回调函数会下载新的系统软件（固件），并根据需要擦除/重新编程高地址存储块。

7.4 运行演示程序

注

以下说明假设已安装了 IDE 和终端应用程序并具备相关基本知识。详细信息请参见《TWR-K60N512 快速入门指南》，可从 http://www.freescale.com 获取。

7.4.1 说明

1. 下载并安装演示软件。选择一个目录安装。将该目录称为<安装目录>。
2. 在您的计算机与演示板之间插入 USB 线缆，用于供电和连接调试器。

3. 打开《快速入门指南》中所述的 PE 终端实用程序。确保波特率为 115200、无校验且为 8 位数据位。单击“打开串行端口”。
4. 打开 IAR 或 CodeWarrior。
5. 打开交换演示工程

   IAR
   - 打开工作区<安装目录>**\build\iar\swap_demo\swap_demo.eww**

   CodeWarrior
   - 设置工作区为<安装目录>\
   - 导入文件夹<安装目录>**\build\cw\swap_demo**中的工程
6. 选择目标配置

   IAR
   - 选择 **FLASH_512KB_PFLASH** 配置。

   CodeWarrior
   - 选择 **MK60N512VMD100_INTERNAL_FLASH** 配置
7. 编译工程
8. 下载与调试

   IAR
   - 单击“下载与调试”按钮

   CodeWarrior
   - 打开调试配置
   - 选择 **k60-swap_demo_MK60N512VMD100_INTERNAL_FLASH_PnE_OSJTAG**

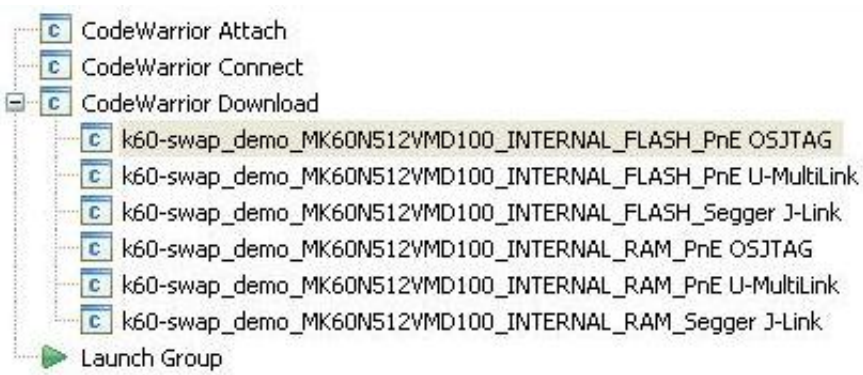


图 7. 调试配置

   - 点击调试
9. 在 PE 终端实用程序中，输出将显示在屏幕上。

   首先，代码会检查哪个存储块位于地址 0x0000。然后，代码执行交换命令来检查交换系统状态。在您第一次执行该代码时，交换系统将报告“未初始化”。

图 8. 终端实用程序输出：首次交换之前

10. 代码执行命令行界面。输入“swap”开始交换演示。交换驱动程序将按顺序让交换系统执行每个交换状态，并调用回调函数，将交换状态报告给调用函数。
11. 当交换系统位于更新状态时，回调函数会模拟固件更新，擦除并重新编程无效（高地址）存储块，复制有效（低地址）存储块内容到无效（高地址）存储块中。

注

本示例将 Flash 交换指示器置于每个存储块的最后一个扇区中。不允许对 Flash 交换指示器编程，因此当编程无效（高地址）存储块时应避免此区域。

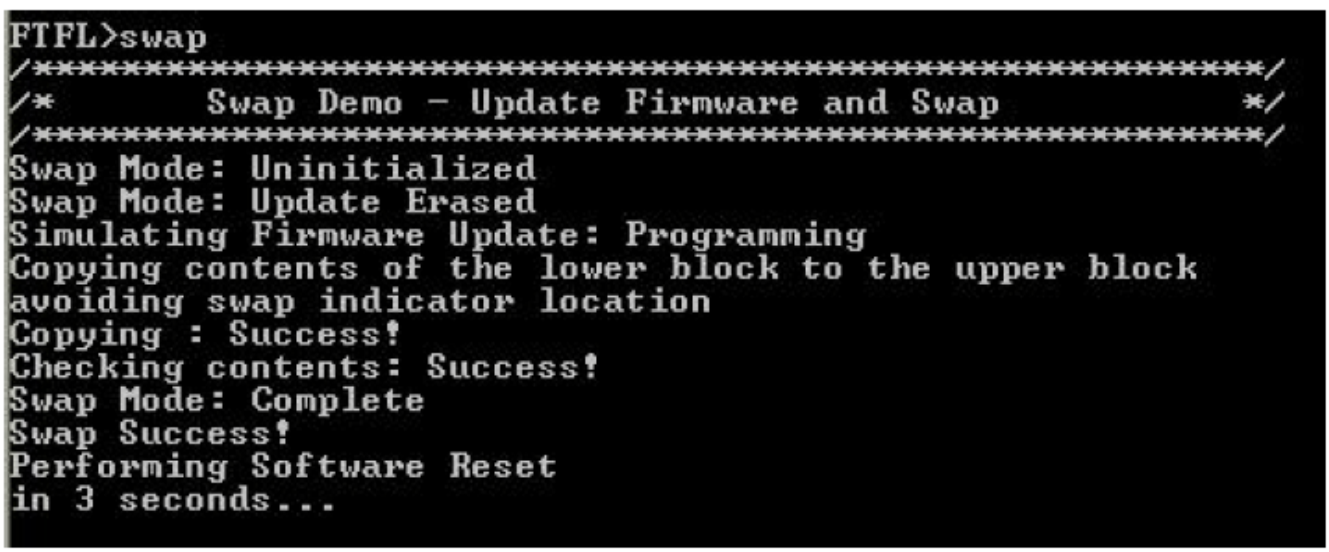


图 9. 终端实用程序输出：首次交换期间

12. 代码在 3 秒内执行软件复位，以完成交换。

图 10. 终端实用程序输出：首次交换之后

13. 当复位发生时，代码将再次启动。Flash 存储块至此交换完成。存储块 1 位于地址 0x0000。现在交换状态为就绪状态。再次输入“swap”命令，交换回存储块 0。

注

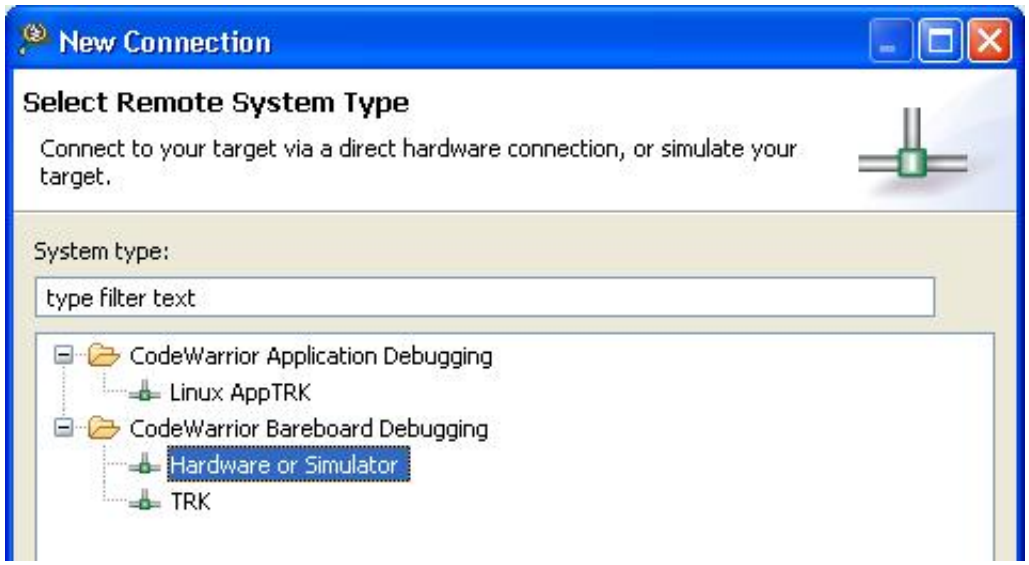
如果软件复位发生时连接着调试器，则调试器将暂停。恢复或断开调试器，然后按下演示板上的复位按钮。

7.4.2 设置远程系统（CW 10.1）

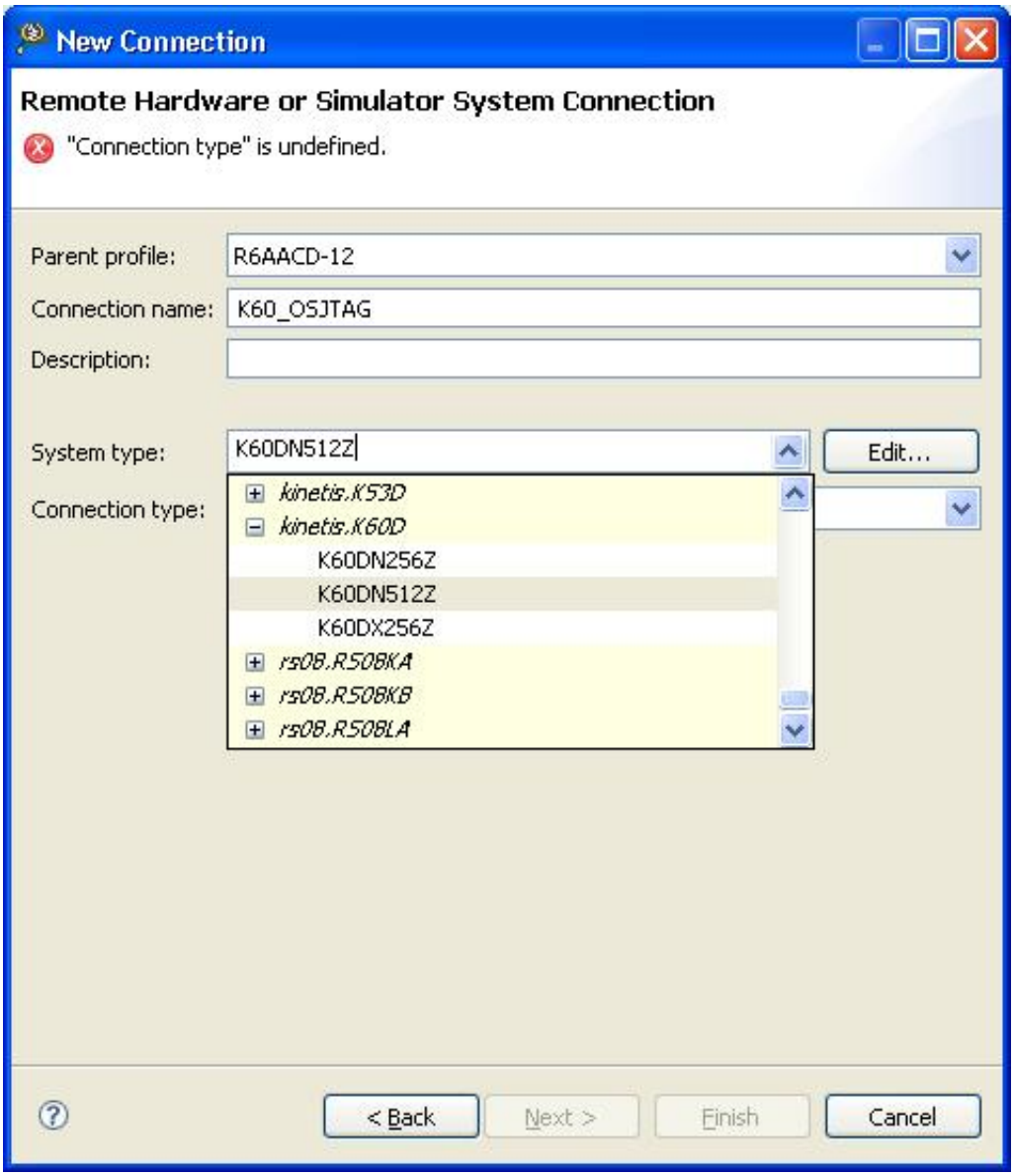
第一次在 CodeWarrior 10.1 上运行此演示程序时，可能需要设置远程系统。最新版本的 CodeWarrior 无需此操作。

1. 打开“Debug Configurations”（调试配置），在“Remote System”（远程系统）下，单击“**New…**”（新建）

2. 这将打开“New Connection”（新连接）窗口。在“CodeWarrior Bareboard Debugging”（CodeWarrior 裸板调试）下，选择“**Hardware or Simulator**”（硬件或仿真器）。



3. 为“**Connection name**”（连接名称）输入一个名称（如 K60_OSJTAG），并选择 **K60DN512Z** 作为“**System**”（系统）类型。单击“**Finish**”（完成）。



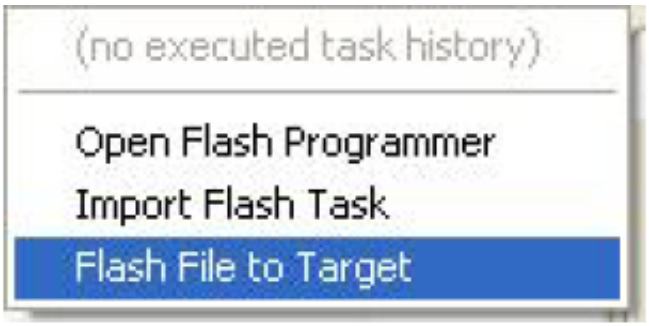
7.4.3 运行演示后的恢复连接问题

存在一个潜在问题，在 Flash 交换演示执行后，CodeWarrior 无法擦除/重新编程器件。这与芯片版本无关。

要解决该问题，请使用 *Flash File to Target* 工具来擦除 Flash。*Flash File to Target* 工具使用“擦除全部”命令，这将擦除整个 Flash 并将交换系统清除为未初始化状态。

1. 单击闪电图标旁的箭头，打开 Flash File to Target 工具。

2. 选择“**Flash File to Target**”



3. 单击“**Erase Device**”（擦除器件）

8 结论

程序 Flash 存储器交换是飞思卡尔 Kinetis 微控制器诸多极具吸引力的功能之一。对于需要极度可靠的固件更新机制，以提供安全备份副本和最短应用停机时间的系统而言，交换功能是一种理想之选。程序 Flash 交换简单易用，能够降低软件复杂性。它非常适合多任务操作系统，如飞思卡尔免费的 MQX RTOS。

如需更多资源，请参见本应用笔记提供的软件示例，以及参考资料 章节。

9 参考资料

1. 可以通过 http://www.freescale.com/kinetis 获取 Kinetis 微控制器的喜相关文档、软件和工具。
2. 用于 Kinetis 微控制器的 Flash 驱动程序软件。在 http://www.freescale.com 上搜索关键字“C90TFS_FLASH_DRIVER”。
3. Kinetis 示例代码。在 http://www.freescale.com 上搜索关键字“KINETIS512_SC”。

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

Document Number AN4533
Revision 0, 06/2012