

作者：恩智浦半导体

1 简介

通常有三种方法实现 EEPROM 功能。其一是使用真正的 EEPROM，例如具有 256 B EEPROM 的 KEA64。优点是不占用闪存，但缺点是价格昂贵。第二种方法是使用软件实现 EEPROM 功能，如 KEA8 和 KEA128。优点是便宜，但缺点是它占用了闪存，包括用于 EEPROM 实现的额外代码和模拟 EEPROM 占用的闪存。第三种方法是使用固件来实现 EEPROM 功能，例如 S32K1xx。优点在于 EEPROM 功能的实现对客户来说是绝对透明的，并且不需要闪存来存储用于 EEPROM 实现的额外代码。

本应用笔记主要介绍 S32K1xx EEPROM (EEE) 功能特性和用法。S32K1xx EEPROM (EEE) 允许用户将部分片上闪存配置为增强型 EEPROM、额外闪存或两者的组合。

有关本应用笔记中提及的任何参考的详细信息，请参阅 S32K1xx 参考手册和数据手册。

2 S32K1xx EEPROM (EEE) 功能

S32K1xx EEPROM (EEE) 具有许多功能，允许更换外部 EEPROMs 以提高其性能。

S32K1xx EEE 功能包括：

- 自动化 – 无需客户软件开发。
 - 重启：EEE 镜像载入 RAM。
 - 读：直接从 RAM 读取记录。无 EEE 操作。
 - 写：直接向 RAM 写入记录。镜像与 EEE 闪存自动同步。
- 利用最佳实践来优化其可靠性和循环耐久性。
 - 循环负载均衡。
 - 扇区失效。
- 耐掉电。
 - 新的快速写入模式，优先在断电前写入少量数据。
- 已经在生产中得到验证。
 - 已应用超过150万片 C90TFS 在工业和消费领域。

目录

1 简介	1
2 S32K1xx EEPROM (EEE) 功能.....	1
2.1 S32K1xx EEE 工作.....	2
2.2 S32K1xx EEE 如何使用内存 ..	2
3 使用 S32K1xx EEE	4
3.1 S32K1xx EEE 分区	5
3.2 S32K1xx FlexRAM 配置	10
3.3 S32K1xx 命令错误处理	14
3.4 S32K1xx EEE 错误处理	14
3.5 S32K1xx EEE 启动.....	14
3.6 S32K1xx 读取和写入 EEE	15
4 S32K1xx EEE 性能	16
5 S32K1xx 掉电检测	16
6 S32K1xx 新的快速写入模式.....	17
7 S32K1xx EEPROM 耐久性	18
8 软件注意事项	19
8.1 同步操作.....	19
8.2 启用 CSEc 和 EEPROM	19
8.3 上电建议	19
8.4 数据记录检查.....	20
8.5 功耗模式转换.....	20
9 附录 A EEPROM 示例	20
10 修订历史.....	23



2.1 S32K1xx EEE 工作

为了提供增强的 EEPROM 功能，S32K1xx EEE 使用了一块 RAM (FlexRAM)，一块闪存 (FlexNVM) 和 EEE 状态机器。启用 EEE 功能后，FlexRAM 将成为您的 EEE 存储空间。FlexRAM 地址空间是您访问所有 EEE 数据的地方。当访问 EEE 时，EEE 状态机会跟踪数据并将其备份为数据记录，存储在用作 E-flash 的 FlexNVM 的部分空间。使用大容量 E-flash 来备份少量 EEE 数据，使 S32K1xx EEE 的实现提供极高的耐久性。EEE 状态机使用72位记录将数据从 EEE 备份到闪存 (E-Flash) 中。记录的32位用于数据，另外40位是有关数据的地址、状态和奇偶校验信息。按需写入和删除数据记录。这意味着如果 EEPROM 中的某个位置从未被访问过，则不会有该位置的数据记录。这有助于减少需要备份的数据量，并增加内存耐久性。

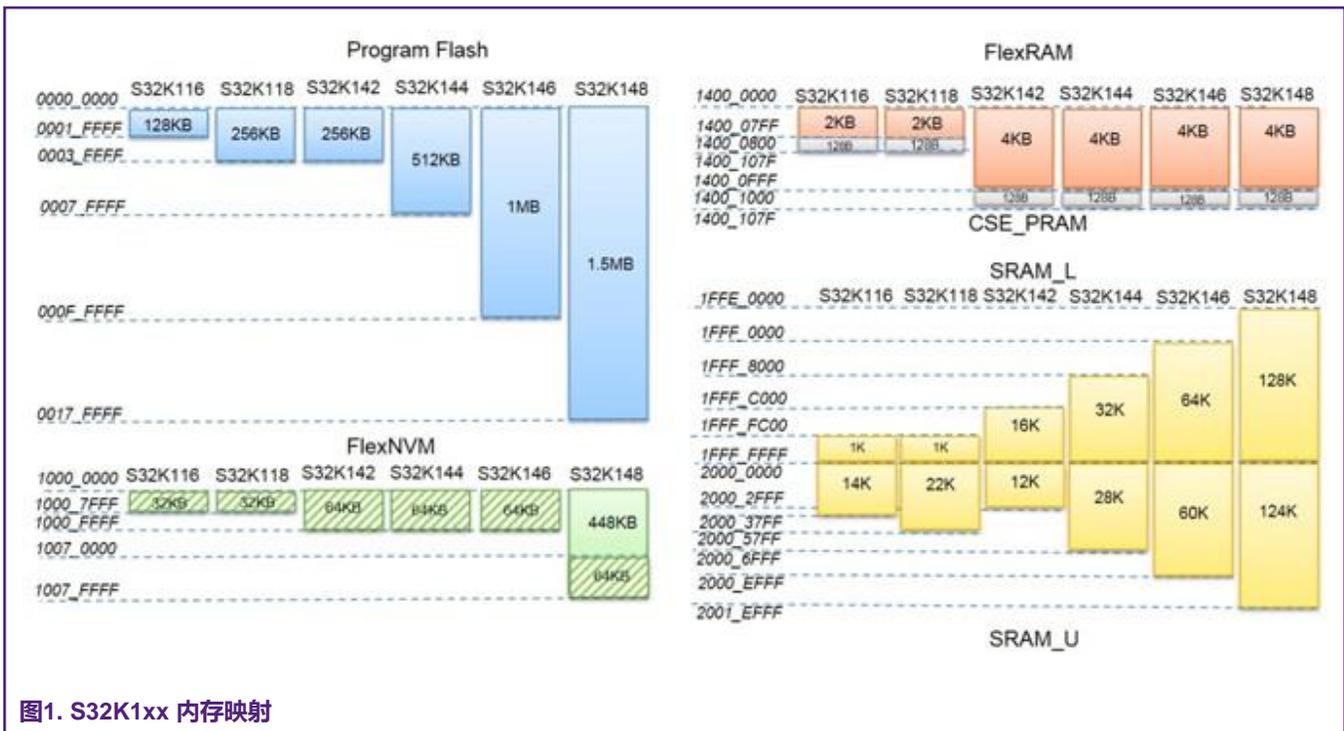
2.2 S32K1xx EEE 如何使用内存

S32K1xx 拥有两块独立的闪存，P-Flash 块和 FlexNVM 块。P-Flash 块用作程序闪存块，但也可用于存储指令和数据。FlexNVM 块是一个可配置的闪存块，可用作额外的闪存空间 (D-flash)、支持增强型 EEPROM 功能的存储器 (E-flash) 或作为两者的组合。

注意

FlexNVM 中不用作 EEE 备份存储器 (E-flash) 的部分称为 D-flash。该闪存通常用于数据存储空间；但是，与 P-flash 一样，D-flash 实际上可以用于指令或数据的存储。

S32K1xx 由 FlexNVM、FlexRAM 和 EEE 状态机组成。这三个模块共同支持 EEE 功能。第2页的图 1 展示了 S32K1xx 内存映射。



使用 S32K148 需要特别注意，其中 FlexNVM 与 P/DFlash 共享 512KB 的地址空间。软件注意事项中描述了这些注意事项的详细信息。

下面的小节使用 S32K144 的内存映射为例，作为不同 EEE 配置的参考。尽管不同的 S32K1xx MCU 有各自的内存空间/地址，但不同的 S32K1xx MCU 适用于相同的原理。

2.2.1 禁用 EEE 功能的 S32K1xx

下图显示了禁用 EEE 功能时 S32K144 的内存分配。

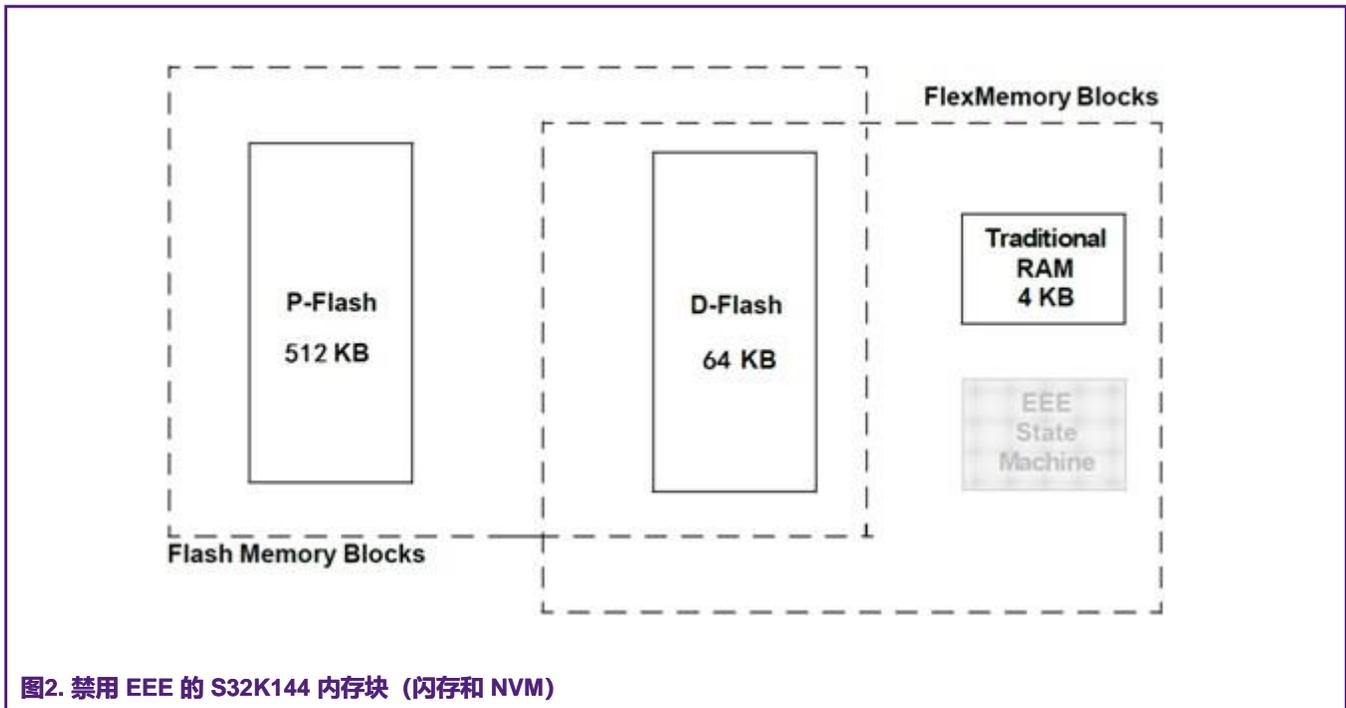


图2. 禁用 EEE 的 S32K144 内存块 (闪存和 NVM)

P-Flash 区始终是 P-Flash。其功能不会因为任何 FlexMemory 配置而改变。因为在这种情况下 EEE 未被使用，整个 FlexNVM 被分配为 D-flash 空间（不需要 E-flash）。FlexRAM 变成了 4kB 的传统 RAM。这意味着它可以被用作额外的内存空间，但请记住，它是以闪存时钟速度运行而不是内核速度运行（紧耦合的内存 RAM，TCM 以内核速度运行）。EEE 状态机存在于设备中，但未处于活动状态。

注意

FlexRAM 作为传统 RAM 使用没有 ECC 校正功能。

2.2.2 EEE 功能启用的 S32K1xx

下图显示了在启用 EEE 功能并且整个 FlexNVM 用于备份 EEE 数据的情况下内存的分配。

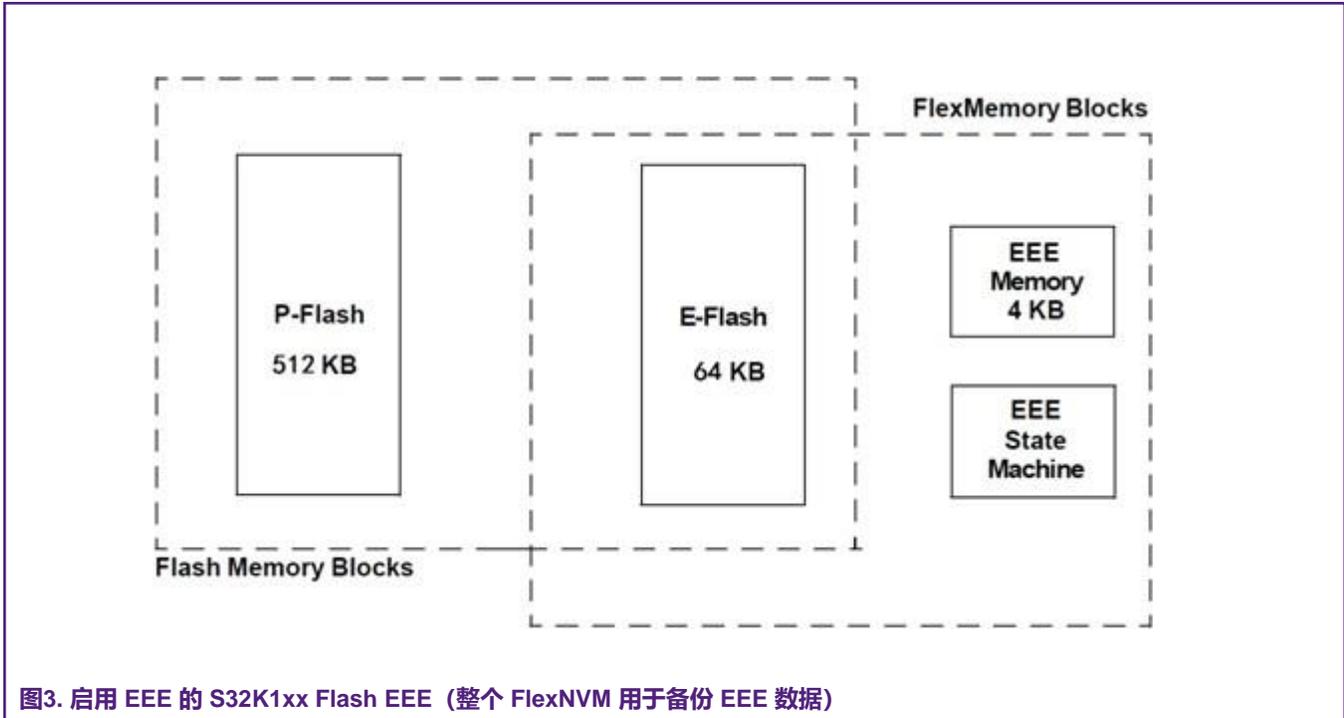


图3. 启用 EEE 的 S32K1xx Flash EEE (整个 FlexNVM 用于备份 EEE 数据)

启用 EEE 功能后，可使用众多配置选项。第4页 图 3 展示了整个 FlexNVM 用作 E-flash 的示例。FlexRAM 作为 EEE 内存空间（多达 4kB）。EEE 数据的任何读写操作都使用这 4kB 存储空间，因为 E-flash 无法直接访问。EEE 状态机自动管理 EEE 内存的所有写入，并按需生成闪存写入及擦除操作到 E-flash。

2.2.2.1 E-flash 及 EEE 内存配置细节

图 3 展示 S32K1xx 存储单元的概况，以及可配置存储区的相应功能。本节更详细地描述了实际使用 E-flash 及 FlexRAM 来创建 EEE 子系统，并提供一些不同配置的示例。S32K1xx 允许很多不同的内存配置选项。EEE数据的大小和用来备份EEE数据的 E-flash 内存大小都是可编程的。这让你可以在 EEE 内存大小和耐久性之间做出权衡。

有两个可变参数来定义你实际的存储空间使用情况，这些参数有：

1. EEE 大小 — 这是 EEE 数据需要的总大小。对于 S32K14x 设备，EEE 的总大小在 0 或 4kB，S32K11x 设备则为 0 或 2kB。
2. FlexNVM 分区 — 该参数定义了 FlexNVM 作为普通闪存 (D-flash) 的数量和用作 EEE 备份的数量 (E-flash)。如果被使用，那么至少 32kB (针对 S32K14x 设备) 或 24kB (针对 S32K11x 设备) 的 FlexNVM 必须被分配为 E-flash。为了更好的获得EEE的耐久性，整个 FlexNVM 可以被用作 E-flash。

FlexNVM 耐久性受 EEE 备份内存和 EEE 大小之间关系的影响。有关 NVM 可靠性规范的详细信息，请参阅 S32K1xx 数据手册。

3 使用 S32K1xx EEE

3.1 S32K1xx EEE 分区

要使用 EEE 的功能，内存必须分区。分区过程告诉状态机将使用多少 EEE 内存和 FlexNVM 闪存备份 EEE。Flash 有一个用于配置 EEE 的特殊分区命令。分区命令用于对 [E-flash and EEE memory configuration details](#) 描述的两个 EEE 配置参数进行编程。这两个参数被编程到闪存模块内的一个特殊位置。由于这是一个非易失性存储位置，因此在设备的整个生命周期中只需进行一次分区。在启动程序分区命令之前，FlexNVM 和 D-flash IFR 必须处于擦除状态。建议将新设备分区作为工厂编程的第一步。

注意

分区只能进行一次。如果将闪存重新分区为不同的配置，则记录的数据将丢失并且您无法保证 EEE 获得预期的耐久性。

注意

如果设备被全擦除（Mass Erase）命令擦除，分区信息、EEE 数据和 EEE 位置信息将丢失。在使用 EEE 的设备上启用安全性时（Flash Secure 使能），强烈建议使用后门密钥。后门密钥允许暂时禁用安全性，而无需执行全擦除命令擦除。如果改为使用全擦除，则无法再保证 EEE 的耐久性。使用了 CSEc 功能的器件需要在全擦除之前执行 DBG_AUTH 命令。

3.1.1 S32K1xx 程序分区命令

程序分区命令使 FlexNVM 块准备以用作数据闪存、模拟 EEPROM 或两者的组合，并初始化 FlexRAM。有关详细信息，请参阅 S32K1xx 的参考手册。下表显示了分区命令所需的参数。

表1. 程序分区命令

FCCOB 编号	FCCOB 内容 [7:0]
0	0x80 (PGMPART)
1	CSEc Key 大小
2	SFE
3	复位时 FlexRAM 加载（只用 bit 0）： 复位序列时 0 - FlexRAM，加载了有效 EEPROM 数据 复位序列时 1 - FlexRAM 未加载
4	EEPROM 数据集大小代码
5	FlexNVM 分区代码

3.1.1.1 程序分区命令要求

Flash 命令通过 FTFC 模块中 FCCOB 寄存器进行配置。每个命令包自己的参数，程序分区命令需要六个参数：

- FCCOB0 定义了所需的命令。0x80 指 PGMPART 命令（程序分区命令）。
- FCCOB1 和 FCCOB2 用于 CSEc 配置。这两个参数在 [AN5401 - Getting Started with CSEc Security module](#)。有深入解释。仅对于 EEPROM 功能，这两个值可以设置为 0x00。
- FCCOB3（仅使用 bit 0）配置在复位期间 FlexRAM 是否加载 EEPROM 数据。如果该位（0x00），则 FlexRAM 在复位期间载入 EEPROM 数据。另一方面，如果该位被设置为（0x01），则 FlexRAM 在复位期间不载入 EEPROM 数据，这意味着在复位后它将作为传统 RAM 运行，直到发出 FlexRAM 配置命令更改其模式，如 [S32K1xx FlexRAM configuration](#) 中描述的那样。

- FCCOB4 表示 EEPROM 数据大小。根据 FlexRAM 大小该选项有两个不同的值。

表2. EEPROM 数据大小

EEERAMSIZE 值 (FCCOB4[3:0])	EEPROM 数据大小 (Bytes)
0xF	0
0x3 ¹	2K
0x2 ²	4K

- 1 有效大小仅适用于有2KB 的FlexRAM的器件 (S32K11x芯片)。
- 2 有效大小仅适用于有 4KB 的FlexRAM的器件 (S32K14x芯片)。

注意

当 FlexNVM 分区代码设置为无 EEPROM 操作时 (例如当 FlexNVM 配置为仅用作 D-Flash 时), EEPROM 数据大小必须设置为 0 字节。

- FCCOB5 表明如何在数据闪存 (D-Flash) 和模拟EEPROM 备份存储 (E-Flash) 之间划分 FlexNVM 块。根据 FlexNVM 大小该选项有不同的值。下表显示了这些选项。

表3. 32kB FlexNVM 设备 (S32K11x) 的 FlexNVM 分区代码

FlexNVM 分区代码 (FCCOB5[3:0])	数据 flash 大小 (Kbytes)	EEPROM-备份大小 (Kbytes)
0x0	32	0
0x3	0	32
0x8	0	32
0x9	8	24
0xB	32	0

表4. 64kB FlexNVM 设备 (S32K142、S32K144、S32K146) 的 FlexNVM 分区代码

FlexNVM 分区代码 (FCCOB5[3:0])	数据 flash 大小 (Kbytes)	EEPROM-备份大小 (Kbytes)
0x0	64	0
0x3	32	32
0x4	0	64
0x8	0	64
0xA	16	48
0xB	32	32
0xC	64	0

表5. 64kB FlexNVM 设备 (S32K148) 的 FlexNVM 分区代码

FlexNVM 分区代码 (FCCOB5[3:0])	数据 flash 大小 (Kbytes)	EEPROM-备份大小 (Kbytes)
0x0	512	0
0x4	448	64
0xF	512	0

在启动程序分区命令之前，数据闪 IFR 必须处于擦除状态。启动程序分区代码后，EEPROM 备份存储器 (E-Flash) 大小和 EEESIZE 值被保存在数据闪存 IFR 区域，并且在系统复位期间从数据闪存 IFR 区域加载 FTFC_FCFG1 寄存器的 DEPART 字段。

注意

SIM_FCFG1[DEPART] 字段保存当前 FlexNVM 分区代码，因此，建议用户在启动程序分区命令之前检查此字段并验证 FlexNVM 尚未分区（未分区 MCU 的 DEPART 字段为 0xF）以避免分区时发生错误。

3.1.1.2 程序分区命令示例

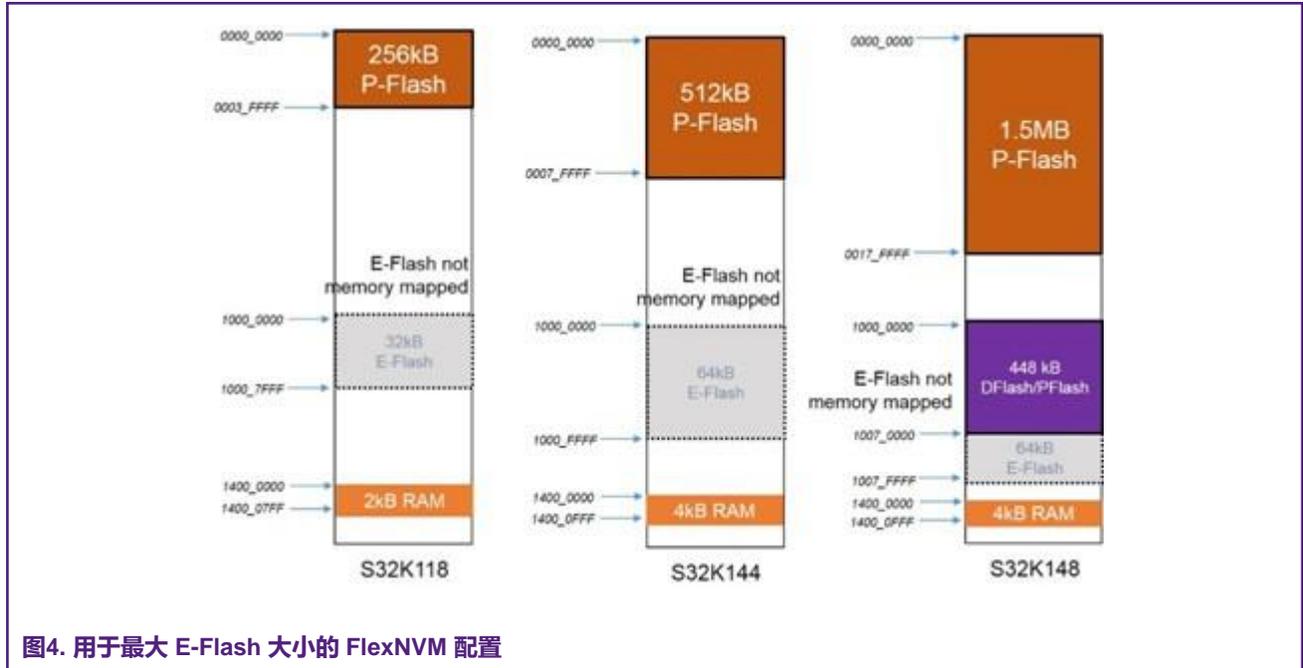
FlexNVM 有多种 EEPROM 备份 (E-Flash) 可用配置。下图展示了 S32K1xx 不同的分区命令和 FlexNVM 配置。注意用户无法使用 (映射) 用于 EEPROM 备份的内存。

注意

FCCOBx 的偏移量与 FCCOB 的索引号不同，这意味着 FCCOB0 不是 FCCOB 数组中的第一个寄存器。有关更多详细信息，请参阅参考手册中 FCCOB 的偏移说明。

- 最大 E-Flash 值。（S32K11x 设备 32kB，S32K14x 设备 64kB）

```
FTFC-> FCCOB [3] = 0x80; /* FCCOB0: Selects the PGMPART command */
FTFC-> FCCOB [2] = 0x00; /* FCCOB1: No CSEc operation */
FTFC-> FCCOB [1] = 0x00; /* FCCOB2: No CSEc operation */
FTFC-> FCCOB [0] = 0x00; /* FCCOB3: FlexRAM loaded with valid EEPROM during reset sequence */
#if S32K11x_DEVICES
FTFC->FCCOB[7] = 0x03; /* FCCOB4: EEPROM data set size code: EEESIZE = 3 (2 kB) */
FTFC->FCCOB[6] = 0x08; /* FCCOB5: FlexNVM Partition code: DEPART = 8 (Data flash: 0, EEPROM
backup: 32 kB) */
#elif S32K14x_DEVICES
FTFC-> FCCOB [7] = 0x02; /* FCCOB4: EEPROM data set size code: EEESIZE = 2 (4 kB) */
FTFC-> FCCOB [6] = 0x04; /* FCCOB5: FlexNVM Partition code: DEPART = 4 (Data flash: 0 kB, EEPROM
backup: 64 kB) */
#endif
flash launchCommand();
```



- D-Flash 和 E-Flash 拆分 FlexNVM (S32K148不可用的选项)

```

FTFC-> FCCOB [3] = 0x80; /* FCCOB0: Selects the PGMPART command */
FTFC-> FCCOB [2] = 0x00; /* FCCOB1: No CSEc operation */
FTFC-> FCCOB [1] = 0x00; /* FCCOB2: No CSEc operation */
FTFC-> FCCOB [0] = 0x00; /* FCCOB3: FlexRAM loaded with valid EEPROM during reset sequence */
#if S32K11x_DEVICES
FTFC->FCCOB[7] = 0x03; /* FCCOB4: EEPROM data set size code: EEESIZE = 3 (2 kB) */
FTFC->FCCOB[6] = 0x09; /* FCCOB5: FlexNVM Partition code: 9 (Data flash 8 kB, EEPROM backup 24 kB) */
*/
#elif S32K14x_DEVICES
FTFC-> FCCOB [7] = 0x02; /* FCCOB4: EEPROM data set size code: EEESIZE = 2 (4 kB) */
FTFC-> FCCOB [6] = 0x03; /* FCCOB5: FlexNVM Partition code: 3 (Data flash 32 kB, EEPROM backup 32 kB) */
*/
#endif
flash_launchCommand();

```

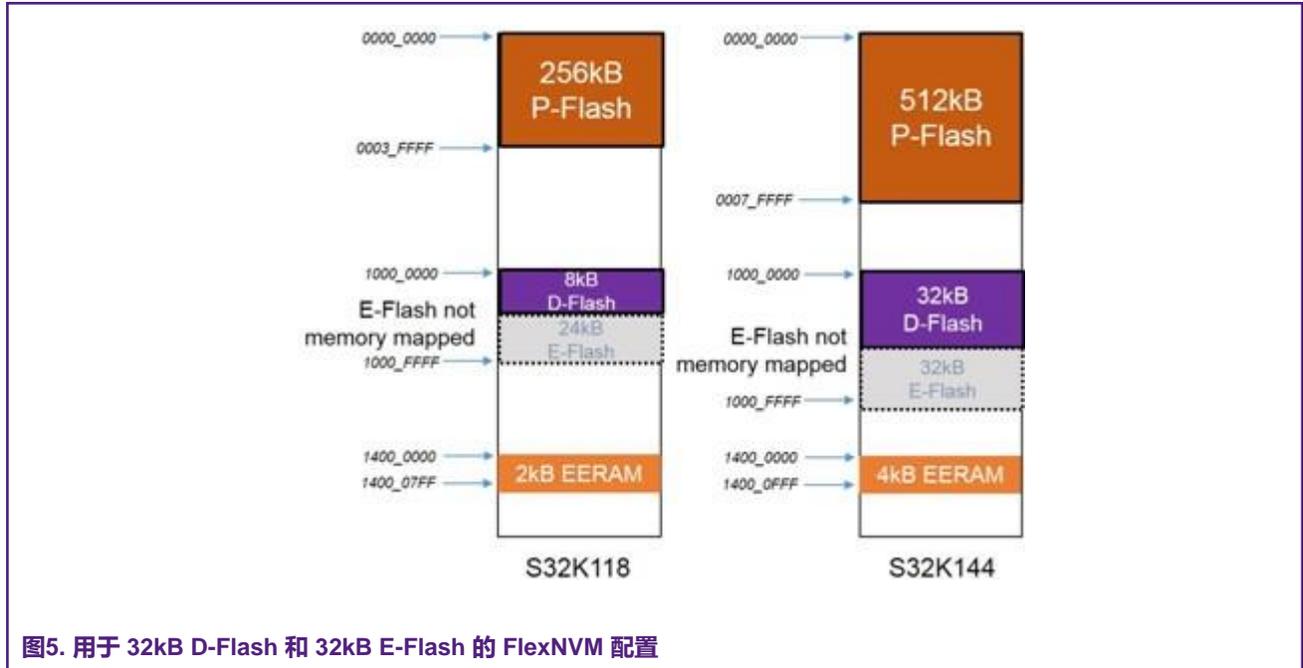


图5. 用于 32kB D-Flash 和 32kB E-Flash 的 FlexNVM 配置

- FlexNVM 配置为 D-Flash 及 0kB 的 E-Flash

```
FTFC-> FCCOB [3] = 0x80; /* FCCOB0: Selects the PGMPART command */
FTFC-> FCCOB [2] = 0x00; /* FCCOB1: No CSEc operation */
FTFC-> FCCOB [1] = 0x00; /* FCCOB2: No CSEc operation */
FTFC-> FCCOB [0] = 0x00; /* FCCOB3: FlexRAM loaded with valid EEPROM during reset sequence */
FTFC-> FCCOB [7] = 0x0F; /* FCCOB4: EEPROM data set size code: EEESIZE = 0x0F (0kB) */
FTFC-> FCCOB [6] = 0x00; /* FCCOB5: FlexNVM Partition code:0 (DFlash 32/64/512 kB, EEPROM backup 0 kB) */
flash_launchCommand();
```

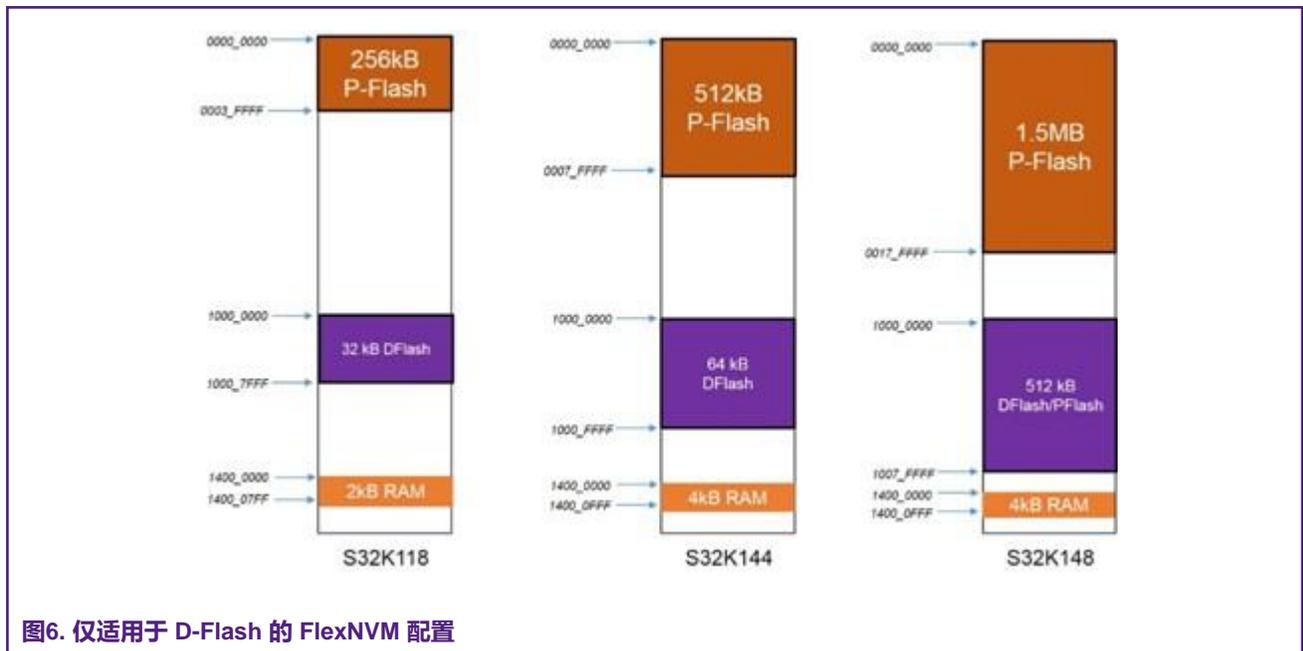


图6. 仅适用于 D-Flash 的 FlexNVM 配置

除配置 FlexNVM 和设置 E-Flash 大小，还需为所需的 EEPROM 操作配置 FlexNVM。

3.2 S32K1xx FlexRAM 配置

设置 FlexRAM 命令可以更改 FlexRAM 的功能:

- 当没有为模拟 EEPROM 进行分区时 (或在分区命令时, 配置 FlexRAM 在复位期间不加载EEPROM的数据 FlexRAM load during reset option field to 0x01) , FlexRAM 通常被用作传统 RAM。 [1]
- 当模拟 EEPROM 进行分区时, FlexRAM 通常存储 EEPROM 数据。

下图展示了 FlexRAM 的配置命令。有关详细信息, 参阅参考手册中的 FTFC 章节。

FCCOB Number	FCCOB Contents [7:0]
0	0x81 (SETRAM)
1	FlexRAM Function Control Code (see Table 32-47)
2	Reserved
3	Reserved
4	Number of FlexRAM bytes allocated for EEPROM quick writes [15:8]
5	Number of FlexRAM bytes allocated for EEPROM quick writes [7:0]
5	Brown-out (BO) Detection Codes <ul style="list-style-type: none"> • 0x00 - No EEPROM issues detected • 0x01 - BO detected before completing EEPROM quick write maintenance • 0x02 - BO detected before completing EEPROM quick writes • 0x04 - BO detected during normal EEPROM write activity
6	Number of EEPROM quick write records requiring maintenance [15:8]
7	Number of EEPROM quick write records requiring maintenance [7:0]
8	EEPROM sector erase count [15:8]
9	EEPROM sector erase count [7:0]

图7. 设置 FlexRAM 功能命令

3.2.1 Set FlexRAM 函数命令要求

配置 FlexRAM 命令用于配置 FlexRAM 的功能和其他一些快速写入设置。这个命令将根据在控制字段写入不同的数值来执行不同的操作。

- FCCOB0 定义所需的命令。0x81 是 SETRAM 命令 (Set FlexRAM 功能命令) 。
- FCCOB1. FlexRAM 功能控制字段用于执行不同的操作。下表展示了不同的控制代码以及每句代码执行的操作:

表6. FlexRAM 功能控制选项

FlexRAM 功能控制代码	动作
0xFF	让FlexRAM 用作RAM: <ul style="list-style-type: none"> — 清除 FCNFG[RAMRDY] 和 FCNFG[EEERDY] 的记号 (flag) — 为 ones to all FlexRAM 位置写一个背景 — 设置 FCNFG[RAMRDY] 记号

表格在下一页继续.....

[1] FlexRAM 的运行速度与 SRAM 不同, 而且 FlexRAM 没有 ECC 可用。

表6. FlexRAM 功能控制选项 (续)

FlexRAM 功能控制代码	动作
0xAA	完成中断的EEPROM 快速写入流程： — 清除FCNFG[EEERDY] 和 FCNFG[RAMRDY] 记号 — 在EEPROM 快速写入操作完成维护 — 设置FCNFG[EEERDY] 记号
0x77	EEPROM 快速写入状态命令： — 清除 FCNFG[RAMRDY] 和 FCNFG[EEERDY] 记号 — 报告仿真的 EEPROM 状态 — 设置 FCNFG[EEERDY] 记号
0x55	让FlexRAM 用于 EEPROM 快速写入： — 清除 FCNFG[RAMRDY] 和 FCNFG[EEERDY] 记号 — 启用仿真的 EEPROM 系统，用于 EEPROM 快速写入 — 设置 FCNFG[EEERDY] 记号
0x00	让FlexRAM用于仿真的EEPROM： — 清除 FCNFG[RAMRDY] 和 FCNFG[EEERDY] 记号 — 为 ones to all FlexRAM 位置写一个背景 — 复制现有的 EEPROM 数据到 FlexRAM — 设置 FCNFG[EEERDY] 记号

- 根据上述所选择的 FlexRAM 功能控制代码，配置剩余的 FCCOB 参数。

随时可以启动 Set FlexRAM 函数命令。因此，用户可以在一些程序将 FlexRAM 用作传统的RAM，然后再切换回部分 E-RAM (用作 EEE 接口的 FlexRAM)，并在两者间切换，这样操作不会丢失 EEPROM 数据。

3.2.1.1 FlexRAM 用作 RAM

如果FlexRAM 需要被用作传统 RAM，则必须在控制代码字段中使用 0xFF。不使用其他 FCCOB 字段。

FTFC 模块将清除 FCNFG[EEERDY] 和 FCNFG[RAMRDY] 标志，用全 1 的背景模式覆盖写入整个 FlexRAM 的内容，并设置 FCNFG[RAMRDY] 标志，表明 FlexRAM 已准备好用作传统 RAM，在这种情况下，可以对 FlexRAM 进行正常的读写访问。注意 FlexRAM 比 SRAM 慢，而且它们之间的地址也不连续。

注意

EPROT 寄存器的状态不会阻止 FlexRAM 被覆盖。

下一段代码展示了 FlexRAM 配置为传统 RAM 时，FlexRAM 功能命令的 FCCOB 配置。

```
FTFC-> FCCOB [3] = 0x81; /* FCCOB0: Selects the SETRAM command */
FTFC-> FCCOB [2] = 0xFF; /* FCCOB1: Make FlexRAM available as RAM */
flash_launchCommand();
```

3.2.1.2 用于普通仿真 EEPROM 的 FlexRAM (无快速写入)

如果 FlexRAM 需要被用作普通的模拟 EEPROM 操作，必须在控制代码字段中使用 0x00。其他 FCCOB 字段不使用。

注意

这是启用 CSEc 的部件在复位后的默认 FlexRAM 模式。因此，无需再配置 FlexRAM 的控制字段为 0x00。

FTFC 将清除 FCNFG[RAMRDY] 和 FCNFG[EEERDY] 标志位，用全 1 的背景模式覆盖为模拟 EEPROM 分配的 FlexRAM 内容，并将现有 EEPROM 数据从 EEPROM 备份空间复制到 FlexRAM 中。完成 EEPROM 复制后，设置 FCNFG[EEERDY] 标志。当 FlexRAM 设置为工作在模拟 EEPROM 时，对 FlexRAM 的正常读写访问可用，但对 FlexRAM 的写入也会调用模拟 EEPROM 的写入。

下一段代码展示了 FlexRAM 用于普通的模拟 EEPROM 时 FlexRAM 命令的 FCCOB 配置。

```
FTFC-> FCCOB [3] = 0x81; /* FCCOB0: Selects the SETRAM command */
FTFC-> FCCOB [2] = 0x00; /* FCCOB1: Make FlexRAM available for emulated EEPROM */
flash_launchCommand();
/* Wait until previous command is finished */
while ((FTFC-> FCNFG & FTFC_FCENFG_EEERDY_MASK) == 0);
```

3.2.1.3 FlexRAM 用于 EEPROM 快速写入

如果 FlexRAM 需要用于 EEPROM 快速写入，必须在控制代码字段中使用 0x55。需要使用 FCCOB4 和 FCCOB5 字段来指示将用于快速写入的字节数。这两个字段代表一个 16 位值 (FCCOB4[15:8] 和 FCCOB5[7:0])。字节数可以从 16 到 512 (0x0010 到 0x0200) 之间的值中选择，但必须能被 4 整除，否则 FlexRAM 功能命令将失效，FSTAT[ACCERR] 标志会被置位。用其他 FCCOB 寄存器不需要配置。

注意

只允许 32 位快速写入，否则会产生访问错误。

FlexRAM 用于 EEPROM 快速写入时，FTFC 将保留 FlexRAM 的内容并准备模拟 EEPROM 系统以进行快速写入活动。设置 FCNFG 寄存器中的 EEERDY 标志后，模拟 EEPROM 系统准备快速写入。当 FlexRAM 设置为接受 EEPROM 的快速写入时，可以对 FlexRAM 进行读写访问，但对 FlexRAM 的每 32 位写入都将调用 EEPROM 快速写入操作。CCIF 和 EEERDY 标志在每 32 位写入时置为无效，并在相关的 EEPROM 快速写入完成后置为有效。由 FCCOB4 和 FCCOB5 的设置指定的对 FlexRAM 的最后 32 位写入后，模拟 EEPROM 系统将为最后一次写入 (~150µs) 创建数据记录，并开始对整个快速写入的数据块进行 EEPROM 的数据维护。CCIF 和 EEERDY 标志将保持无效，直到所有 EEPROM 快速写入维护操作完成。

下一段代码显示了当 FlexRAM 配置为快速写入 EEPROM 时，FlexRAM 命令的 FCCOB 配置。

```
FTFC-> FCCOB [3] = 0x81; /* FCCOB0: Selects the SETRAM command */
FTFC-> FCCOB [2] = 0x55; /* FCCOB1: Make FlexRAM available for emulated EEPROM (quick writes) */
FTFC-> FCCOB [1] = 0x00; /* FCCOB2: Reserved */
FTFC-> FCCOB [0] = 0x00; /* FCCOB3: Reserved */
/* Allocate 512 bytes for quick write operation */
FTFC-> FCCOB [7] = 0x02; /* FCCOB4: Number of bytes allocated for EEPROM quick writes [15:8] */
FTFC-> FCCOB [6] = 0x00; /* FCCOB5: Number of bytes allocated for EEPROM quick writes [7:0] */
flash_launchCommand();
```

[S32K1xx New Quick Write Mode](#) 描述了快速写入模式。

注意

在切换到另一个 FlexRAM 功能之前，必须完成快速写入。如果用户想要在另一个快速写入操作正在进行时开始新的快速写入操作，则同样适用；第一个快速写入操作必须在开始新的写入操作之前完成（无法中止正在进行的操作）。

3.2.1.4 FlexRAM 用于查询 EEPROM 的写状态

如果需要使用 FlexRAM 来查询 FlexRAM 的写入状态，则必须在控制代码字段中使用 0x77。其他 FCCOB 字段不用于启动命令，但是其中部分字段用于从写入操作中检索状态。

命令启动后，FCNFG[EEERDY] 标志被清除，FTFC 将询问模拟 EEPROM 系统并报告 EEPROM 清理要求（在 FCCOB6 和 FCCOB7 上），EEPROM 扇区擦除计数（在 FCCOB8 和 FCCOB9 上）和 FCCOB5 会基于下表所示条件保存掉电检测状态码。

表7. 掉电检测代码选项

掉电代码	条件
0x00	无EEPROM问题
0x04	如果EEPROM的正常写入活动在完成记录前因复位或断电而中断。
0x02	如果EEPROM快速写入活动在写入所有快速写入记录之前被复位操作中断。
0x01	如果中断发生在所有请求的快速写入完成之后，但在快速写入维护完成之前。

CCIF 和 EERDY 标志位将保持无效，直到 EEPROM 状态已加载到 FCCOB 寄存器中。

下一段代码展示了 FlexRAM 查询（快速）写入状态时 FlexRAM 函数命令的 FCCOB 配置以及管理这些状态查询结果的方式。

```
uint8_t brownOutCode;
uint16_t quickWriteMaintenance;
uint16_t sectorEraseCount;

FTFC-> FCCOB [3] = 0x81; /* FCCOB0: Selects the SETRAM command */
FTFC-> FCCOB [2] = 0x77; /* FCCOB1: EEPROM quick write status query */
flash_launchCommand(); /* Launch command and wait until it finishes */
/* FCCOB5: Brown-out (BO) Detection Codes */
brownOutCode = FTFC-> FCCOB [6];
/* FCCOB6,7: Number of EEPROM quick write records requiring maintenance */
quickWriteMaintenance = FTFC-> FCCOB [5] << 8 | FTFC-> FCCOB [4];
/* FCCOB8,9: EEPROM sector erase count */
sectorEraseCount = FTFC-> FCCOB [11] << 8 | FTFC-> FCCOB [10];
```

如果在最后一次快速写入完成之前发生复位，掉电代码将被设置为 0x02，这意味着之前的写入将被忽略并且不会有任何更新生效，维持上次的记录。如果在所有快速写入完成后发生复位（但维护无法完成），掉电代码将设置为 0x01，这些记录可以稍后完成（通过使用 FlexRAM 命令，完成被中断的快速写入过程）。

如果在正常写入活动完成之前发生复位，掉电代码将被设置为 0x04，这意味着之前的写入将被忽略，更新不会生效。

3.2.1.5 FlexRAM 用于完成中断 EEPROM 快速写入过程

如果需要使用 FlexRAM 来完成被中断 EEPROM 写入过程，则必须在控制代码字段中使用 0xAA。其他 FCCOB 字段不使用。

命令启动后，FCNFG[EEERDY] 标志被清除，FTFC 将处理之前未完成的所有 EEPROM 快速写入操作。如果 EEPROM 快速写入未完成写入所有请求的记录，则已写入的记录将被清除，从而有效地恢复先前有效的记录。如果所有 EEPROM 快速写入记录都已写入但维护操作尚未完成，则将对剩余的快写记录进行维护。CCIF 和 FCNFG[EEERDY] 标志将保持否定状态，直到所有 EEPROM 维护完成。

下一段代码展示了用于完成被中断的 EEPROM 操作时，FlexRAM 功能命令的 FCCOB 配置。

```
FTFC-> FCCOB [3] = 0x81; /* FCCOB0: Selects the SETRAM command */
FTFC-> FCCOB [2] = 0xAA; /* FCCOB1: Complete interrupted */
flash_launchCommand(); /* Launch command and wait until it finishes */
```

3.3 S32K1xx 命令错误处理

对于每个与 EEPROM 功能相关的闪存命令（程序分区命令和设置 FlexRAM 功能），都存在导致命令失效的错误条件。S32K1xx 参考手册中的程序分区命令错误处理以及 Set FlexRAM 功能命令错误处理表，展示了这些条件和每个条件的影响标志位。用户需要确保避免这些错误情况，并在每次启动命令时（Launch command）检查错误标志。

注意

设置后，FSTAT 寄存器中的访问错误（ACCER）和闪存保护违规（FPVIOL）标志会阻止启动更多命令或写入 FlexRAM，直到这些错误标志被清除（通过向其写入1）。

3.4 S32K1xx EEE ECC 错误处理

EEE 状态机会自动处理 ECC 错误。单个位的错误会被自动纠正。对于双位的错误，不会产生总线错误或硬件故障。在向下复制期间，如果有效记录存在双位错误，则跳过该记录。如果该有效记录是一个 EEERAM 位置的唯一有效记录，则读取其位置并返回 1。因此，建议不要用 0xFFFFFFFF 初始化 EEPROM 数据，以识别其是 ECC 错误还是初始化数据。

注意

在上电后，固件进行复制时出现双位错误的情况比较少见，建议通过超时复位来响应。

3.5 S32K1xx EEE 启动

复位后，在分区过程中写入的 EEE 配置选项将自动加载。如果启用了 EEE，则状态机在系统启动期间将来自 E-flash 的 EEE 数据加载到 FlexRAM。将数据从 E-flash 复制到 FlexRAM 所需的时间取决于 EEE 的配置大小和需要解析备份的 E-flash 的数量。FTFC_FCNFG[EEERDY] 标志在 EEE 数据加载完成后会被清除，因此软件必须等待 EEERDY 标志置位，然后才能尝试访问 FlexRAM 中的 EEE 数据。如果需要使用中断而不是软件轮询，则可以使用 CCIF 中断而不是轮询查看 EEERDY 标志。

如果复位期间程序分区命令的 FlexRAM Load 字段被设置为 0x01，EEE 数据不会复制到 FlexRAM，因为它在复位后被用作传统 RAM。用户必须在需要时发出 Set FlexRAM 命令以将 FlexRAM 功能更改为 EEERAM。当启用 CSEc 功能时，用户无法获得延迟 EEE 重置选项（用户必须在程序分区命令中将重置选项字段设置为 0x00）。

在复位期间轮询 FTFC_FCNFG[EEERDY] 和/或 FTFC_FCNFG[RAMRDY] 是获取当前 FlexRAM 模式的好方法。

3.5.1 EEPROM 可用于快速写入时的启动

当 FlexRAM 可用于 EEPROM 快速写入时，建议在上电后检查是否有任何被中断的快速写入操作或维护操作。

程序分区命令定义了两种启动方式。

1. 上电复位后，E-Flash 内容被复制到 FlexRAM (EEERAM) 中。

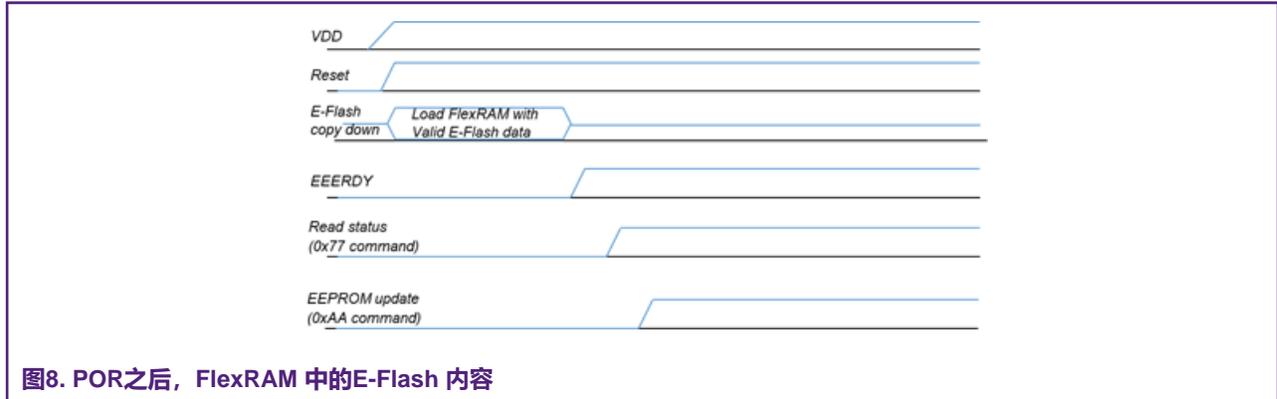


图8. POR之后，FlexRAM 中的E-Flash 内容

2. E-Flash 内容不会被复制到 FlexRAM，因此，FlexRAM 在启动后被用作传统 RAM。（它可以在发出 SetRAM 命令后用作 EEERAM，选择正常/快速写入选项）。

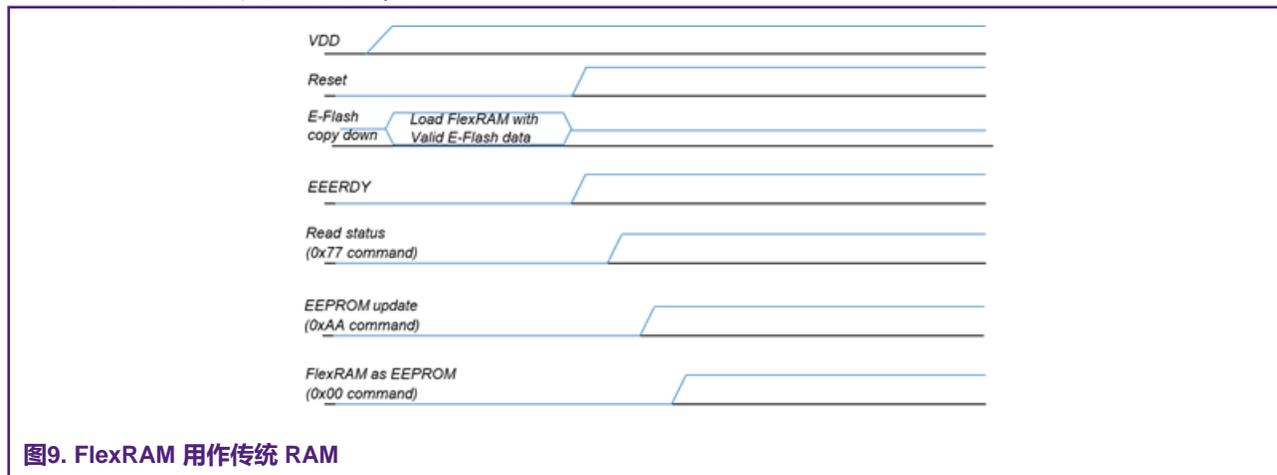


图9. FlexRAM 用作传统 RAM

3.6 S32K1xx 读取和写入 EEE

通过访问 FlexRAM 地址空间来读取和写入 EEE 数据。EEE 空间从 FlexRAM 的开头开始分配。可寻址空间是 FlexRAM 基地址 (0x1400_0000) 到编程的 EEE 大小。

由于 EEE 数据是通过 RAM 访问的，因此可以以任意大小、字节、字或长字对数据进行读写。尽管任何访问大小都是可能的，但用于备份 EEE 的数据记录，使用字大小的数据长度。这意味着字节写入是可能的，但他们对 E-flash 的使用效率较低。

3.6.1 EEE 写入

写入 EEE 空间会启动 EEE 操作以将数据存储到 E-flash 内存中。因为这是闪存编程操作，所以在写入 EEE 空间之前，软件必须测试 CCIF 位以确定是否有任何其他闪存操作正在进行中。由于不允许在一个闪存模块中进行多次并行写入和读并写操作，因此在 EEE 写入完成之前不允许访问 EEE 或 D-flash 空间。检查 CCIF 标志以确定先前的 Flash 或 EEPROM 命令是否已完成。

注意

P-flash 存储器是一个完全独立的逻辑块，因此在 EEE 写入过程中，对 P-flash 的读访问可以继续正常进行。它不适用于与 E-Flash 位于同一内存块的 S32K148 器件中的最后 448 kB P-Flash。

EEE写入不需要按地址顺序写入，实际上用户可以多次写入任何有效位置和/或写入同一位置。

注意

除非设备处于 RUN 模式，否则无法执行 EEE 写入。对于 HSRUN/VLPR模式，只允许 EEE 读取。

3.6.2 EEE 读取

读取 EEE 时，数据由 FlexRAM 提供，因此不会出发闪存操作。但是，在进行 EEE 写入时不允许进行 EEE 读取。在继续访问之前，软件必须在写访问之后等待 CCIF标志。这种情况下 EEE 写入需要特殊函数处理，但 EEE 读取不需要任何特殊软件。这种方法的另一个优点是，如果您有多个 EEE 读访问且中间没有 EEE 写入周期时，则不需要额外的延迟或标志检查。必须考虑的 EEE 读取的特殊情况，是复位后对 EEE 的第一次访问。对于复位后 EEE 的第一次读取，可能需要查看 EEERDY 标志位以确保状态机已完成从 E-flash 到 FlexRAM 的初始数据加载。如果系统启动时间很长，这可以保证在第一次 EEE 读取之前有时间完成初始数据加载，这样可能不需要在第一次读取之前查看 EEERDY 标志。但是，在第一次读取 EEE 之前查看 EEERDY 位会更安全。

注意

某些 CSEc 启动选项（例如，MAC 启动选项）可能会阻止设置 FCNFG[EEERDY] 标志，直到启动过程正确完成。

4 S32K1xx EEE 性能

S32K1xx EEE 除了实现灵活性和高耐久性之外，S32K1xx EEE 比典型的 EEPROM 更快。传统的外部 EEPROM 通常需要大约 5ms 的最长编程时间。相比之下，EEE 可以在 1.5ms 的最坏情况下擦除和写入。也可以通过将 0xFF 写入 EEE 数据位置来预擦除 EEE。预擦除区域有助于减少编程时间，因为它保证不需要擦除周期。通常预擦除数据区域的编程时间约为 100μs。此功能允许在时间紧迫的情况下快速记录数据。一个典型的用例是在检测到即将断电时，需要存储故障数据或操作信息的系统。需要保存的数据量乘以最大写入时间决定了系统中必须提供多少解耦，以维持系统足够长的运行时间，使系统以最小功耗运行以存储数据。通过预擦除数据位置使 EEE 编程时间显著减少，意味着在这种情况下需要更少的去耦并且可以在断电之前存储更多数据。

此外，在检测到掉电的情况下对数据记录使用快速写入模式，确保可以在更短的时间内将关键数据写入 E-Flash，而不是正常写入模式。

5 S32K1xx 掉电检测

EEE 状态机包括可以检测是否有任何 EEE 数据尚未完全写入的逻辑。此功能在文档中称为掉电检测，但这个名称并不完全准确。在写入过程中，任何由于掉电或复位而检测到 EEE 数据未完全写入的情况都被视为一样的。如果复位发生在 EEE 写入过程中，则数据可能会损坏。EEE 状态机会对可能未完全写入的 EEE 数据记录自动测试。如果检测到不完整的记录，状态机会将数据记录标记为不可用，并在下一次 EEE 写入期间将其替换为相关 EEE 地址的之前的有效数据记录。这将保证在写入操作被任何的复位行为中断时，你将得到的是最新的正确的 EEE 写入数据。根据复位发生时的写入操作时间，该值可以是以前的值或新值，但您不会得到损坏的值。

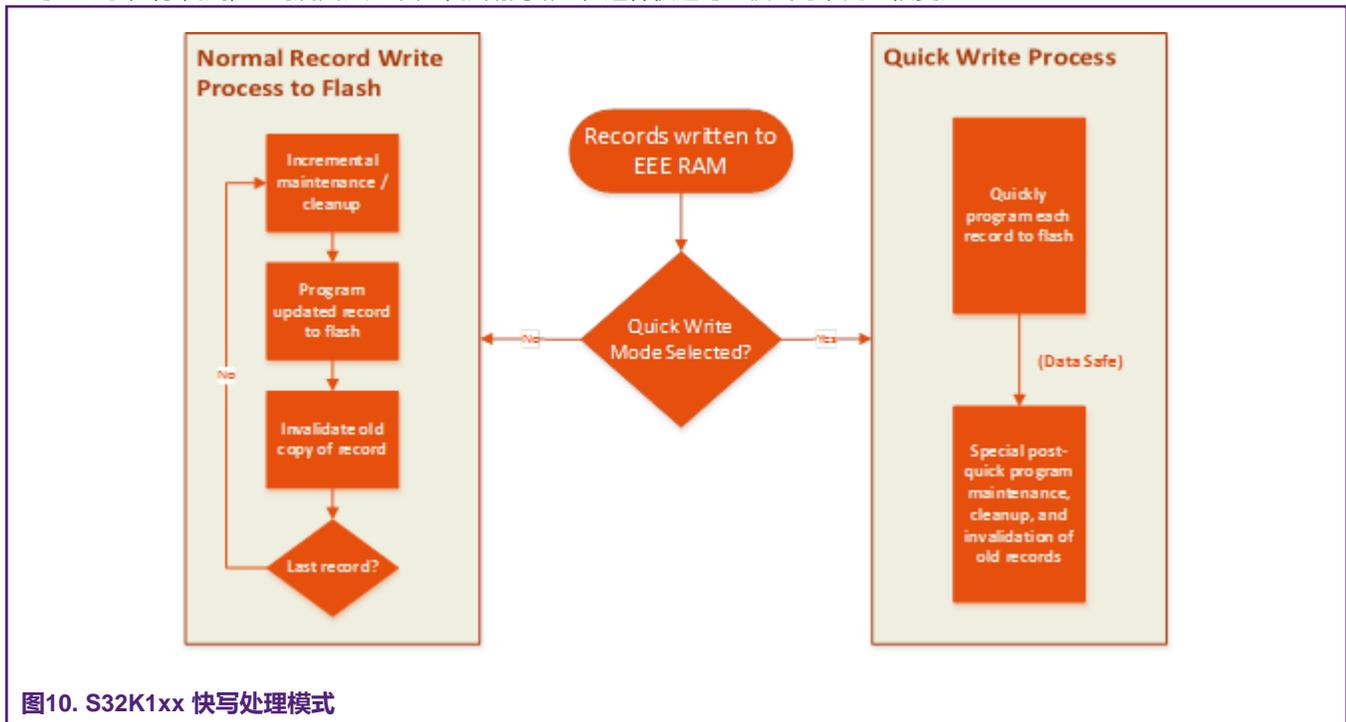
您可以查看 [FlexRAM used for querying write status on EEPROM](#) 部分获取有关掉电检测的更多信息。

为避免 S32K1xx 设备出现掉电情况，有必要正确遵循 [Hardware Design Guidelines](#) 所述的硬件建议，以确保系统在进行任何 EEPROM 备份操作时供电正常。只要保证稳定可靠的电源设计，很少遇到掉电的情况。

6 S32K1xx 新的快速写入模式

快速写入模式优先写入数据，例如在即将掉电之前。如有必要，可在复位后执行 EEE 记录维护。快速写入模式仅限于32位写入。在快速写入维护完成之前，无法进行正常写入。所有需要写入的字节都必须完成写入操作才能有效。如果写入操作在最后一个字节写入之前被复位或断电中断，则本次所有写入都无效，上一次的记录为有效记录。

下图展示了正常记录写入闪存和快速写入过程的原理。正常写入过程对每个记录写入完执行增量维护。快速写入模式尽可能快地写入记录，将维护推迟到后面。如果在维护期间断电，这种快速写入模式可以完全恢复。



快速写入的时间与正常写入模式相比提高了 66%（但最终必须进行维护）。下图展示了快速写入模式与正常写入模式相比的写入时间。

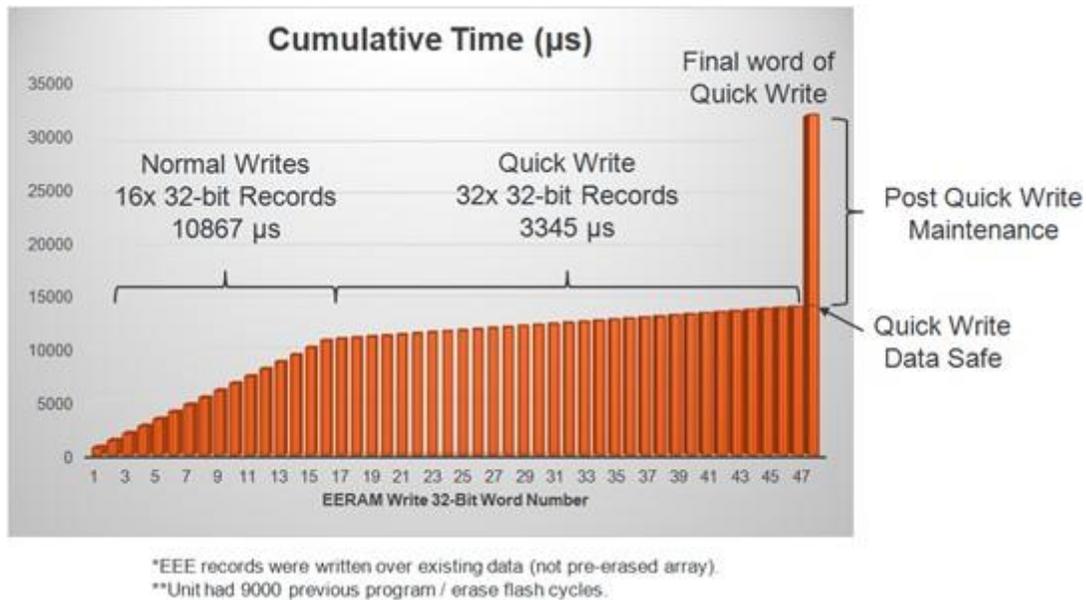


图11. S32K 快速写入过程写入时间表征

注意

这些数字仅供参考，不作为绝对数字。

快速写入过程将所有记录写入 E-Flash 所需的时间更少，但是，总时间（考虑到快速写入后的维护时间）可能比正常写入要长。快速写入的优点是，一旦最后一次快速写入数据已写入（EEE 达到上图所示的快速写入数据安全点），任何掉电事件都不会导致数据丢失，因为可以稍后进行数据维护。但是，如果在写入最后一个数据之前发生掉电，则不会保存新数据。

如果掉电情况发生在快速写入维护完成之前，则可以通过发出 Set FlexRAM 命令来完成中断的快速写入过程，从而在下一个复位周期中恢复数据维护。在 [FlexRAM used for EEPROM quick writes](#) 对其有详细阐述。

根据 EEE 系统的后台数据清理和维护，会出现快速写入似乎比正常写入慢的情况。快速写入比正常写入需要更长的时间，是由于两种模式的记录状态更新和压缩过程不同，但是，快速写入相对于正常写入的优势在于写入新数据的速度更快，因为它在清理过程之前首先将数据保存到 E-Flash 中，而且为防止清理过程由于掉电未完成，那么您可以在下一次上电复位时恢复数据清理过程。

7 S32K1xx EEPROM 耐久性

由于 FlexNVM 分区字段的选择会影响设备的耐久性和数据保留特性，因此根据应用需求选择最佳分区 / EEERAM 大小非常重要。

为了实现 S32K1xx 数据手册中描述的 w/e 循环耐久性，模拟的 EEPROM 备份大小必须至少是 FlexRAM (EEERAM) 中模拟 EEPROM 分区大小的16倍。这可以通过使用所有具有最大的 EEERAM (S32K14x 设备为 4kB, S32K11x 设备为 2kB) 和所有可用 E-Flash (S32K14x 设备为 64kB, S32K11x 设备为 23kB) 来实现。如果用户想要分配少于 64kB 的 E-Flash，例如 S32K14x 设备分配 48kB 的 E-Flash，EEERAM 大小必须少于 4kB (3kB) 以实现数据手册中展示的耐久性。(48kB / 3kB = 16)。

如果需要更高的耐久性，使用更少的记录会增加 RAM 与 NVM 的比率，从而增加 w/e 耐久性。

注意

将 RAM 与 NVM 的比率增加到 16 以上的方法是使用更少的 RAM 地址空间。如果用户仅用 S32K14X 器件 4KB FLEXRAM 中的 2KB，则这个比率从 1:16 变为 1:32，使 W/E 循环耐久性加倍。

恩智浦提供了一种工具来估算在 FlexNVM 上执行的写入/擦除周期数、FlexNVM 更新周期的平均时间以及最坏情况下用户应用程序所需的 EEPROM 周期数。您可以从恩智浦网站 ([FlexMemory Endurance Calculator](#)) 下载该工具以根据您的应用需求计算这些值。

8 软件注意事项

本节重点列出在 S32K1xx 器件上使用模拟 EEPROM 功能时的一些软件注意事项/建议。

8.1 同步操作

由于某些硬件资源由程序闪存 (P-Flash)、数据闪存 (D-Flash) 和 FlexRAM (EEERAM) 共享，因此闪存模块上的某些操作无法同时执行。通常，CCOB 命令不得与任何 CSEc 命令同时执行，反之亦然（有关详细信息，请参阅参考手册中的 CSEc 章节）。

参考手册上允许的针对存储器的同步操作表显示了 P-Flash、D Flash 和 FlexRAM（配置为 EEPROM 和传统 RAM 时）的同时可操作选项。

对于 E-write，只允许同时从 P-Flash 读取。

注意

对于 S32K148，最后 512 kB 存储区由 E/D-Flash 和 P-Flash 共享。在这个单个 bank 中不可能进行读并写访问。在通过对 CSEc、D-Flash、P-Flash 或 EEPROM 的操作进而对该 bank 发出另一个访问之前，请务必轮询 CCIF 标志。有关更多详细信息，请参阅有关 S32K148 的 FlexNVM 操作的单独应用笔记（使用 S32K148 FlexNVM 内存）。

同时访问将被上报和中断，导致 FTFC Read Collision Error Flag (RDCOLERR) 置位，并且之前正在运行的操作也将无效。

8.2 启用 CSEc 和 EEPROM

当 CSEc 操作也被启用时，它将使用多达 512B 的密钥存储，从而 EEPROM 将剩余的大小用于 EEERAM (S32K14x 设备为 3.5kB，S32K11x 设备为 1.5kB)。

此外，如果需要 CSEc，为了使模拟 EEPROM 达到的指定 w/e 耐久性，存储在 FlexRAM 中的记录数量应受到限制，以保持 FlexRAM 与 FlexNVM 的 1:16 比率。为了达到这个比例，建议使用最大的可用 E-Flash 空间。

程序分区命令必须使用最大的可用 E-Flash 大小 (S32K14x 器件为 64kB，S32K11x 器件为 32kB)。

8.3 上电建议

上电时，等待 EERDY 标志被设置，然后再对 EEERAM 进行读/写。此外，强烈建议通过发出一个 set FlexRAM 命令来轮询正常/快速写入状态。

如果 FlexRAM 未配置为在复位期间加载 E-Flash 数据，则用户必须在从 EEPROM 读取/写入之前，通过设置正常或快速写入模式的 FlexRAM 命令来将 FlexRAM 配置为用于 EEPROM。命令启动后，等待 FTFC_FCNFG[EEERDY] 标志被置位。

由于需要清除掉电时写入的无效，复位后的第一次写入时间可能会更长，因此，建议使用 Set FlexRAM 命令来查询状态/未完成过程，如附录部分所示。

8.4 数据记录检查

由于 FlexRAM 不支持 ECC，用户必须执行软件校验（例如使用 CRC）来检测记录的错误。除此之外，对每个安全相关/关键数据备份不止一次（例如每条记录保存 3 次或更多，并且不使用 FlexRAM 中的相邻位置）可以提高数据的可靠性。

如 S32K14x 系列安全手册文档所述，检查模拟 EEPROM 数据的方法必须基于 SETRAM 命令来实现。EEE 驱动将备份数据复制到 EEPROM (FlexRAM) 后，用户必须对 EEPROM 进行完整性检查（如 MISR、CRC），然后执行 SETRAM 切换到系统 RAM，再执行 SETRAM 切换回 EEPROM，一旦数据从 EFlash 被复制到 EEPROM，必须进行另一次完整性检查。如果两次结果相同，那么，复制到 EEPROM 的 EFlash 内容是符合预期的。如果不同，则第一个副本可能由于某种原因损坏了（比如断电等）。有关更多详细信息，请参阅设备安全手册。

8.5 功耗模式转换

由于模拟 EEPROM 的写入只能在器件处于 RUN 模式时执行，如果用户希望切换到任何其他模式（例如以更高频率运行的 HSRUN 模式或以降低功耗值的 VLPR 模式），建议在切换模式前，必须先完成 EEPROM 的写入。如果设备在 HSRUN/VLPR 模式下运行并且需要写入 EEPROM，用户必须切换到 RUN 模式来执行写入，完成后，用户可以切换回 HSRUN/VLPR 模式。

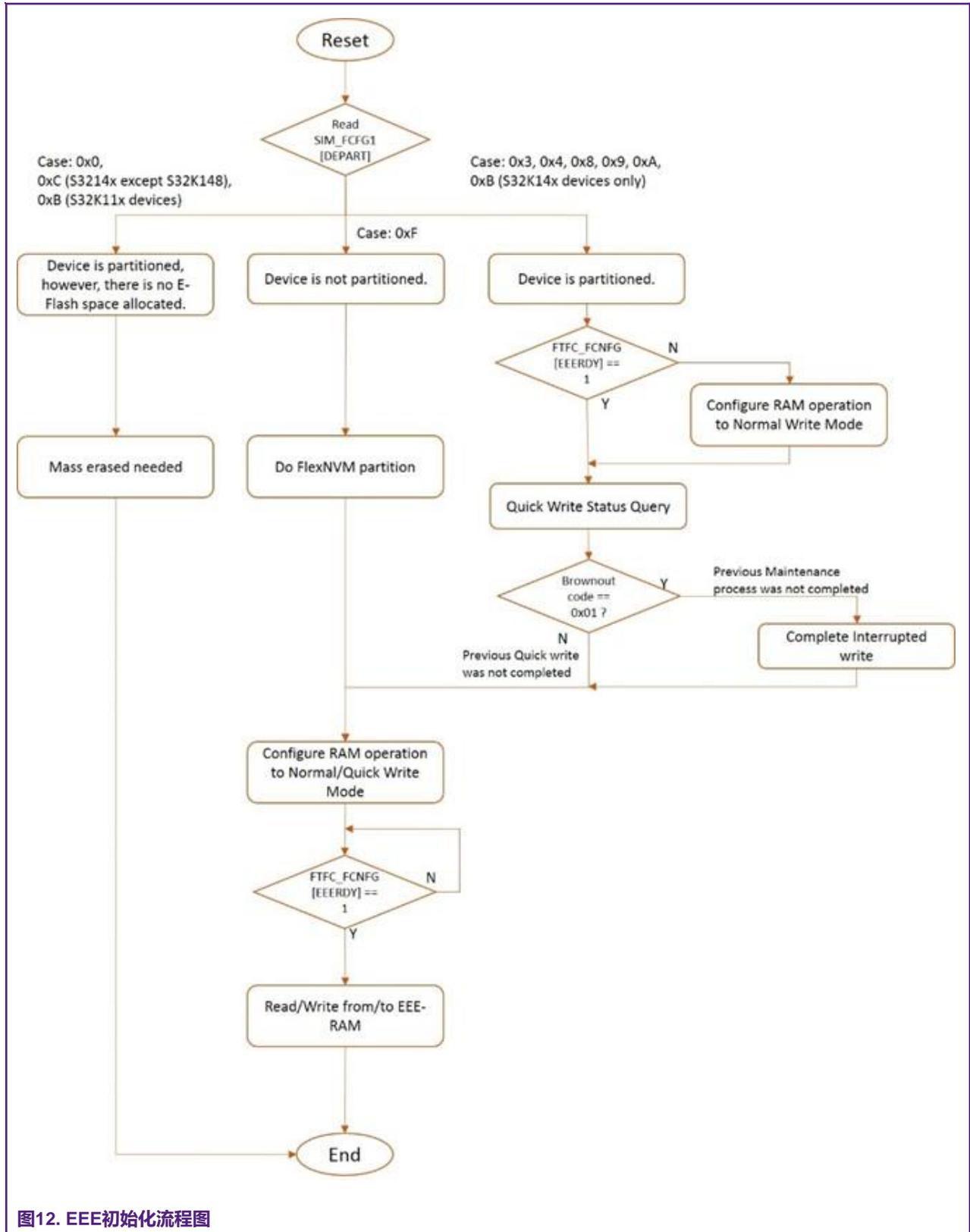
EEPROM 的读取可以在 HSRUN 或 VLPR 模式下执行，如器件参考手册中可用低功耗模式中的哥模块可操作表中所述。

9 附录 A EEPROM 示例

本节重点展示可能涉及 EEPROM 的不同场景以及处理这些场景的方法。它提供了有关不同情况的简要说明，例如 EEPROM 初始化和掉电事件

1. 尽管 EEPROM 分区命令旨在产品生命周期内使用一次，但下图展示了包括 FlexNVM 分区查询操作在内的 MCU 复位事件后的推荐流程图。它显示了如何发出 SetFlexRAM 命令及在多种模式下配置 FlexRAM，例如正常/快速写入模式、快速写入状态的查询或完成被中断的快速写入，其中维护在写入完成前被中断。

这只是使用 FlexNVM 和 EEPROM 功能的一个开始。



2. EEPROM 快速写入用法：由于快速写入模式旨在保存掉电事件之前的关键状态记录，因此用户可以按需正常和快速写入之间切换。由于所有记录都必须在掉电情况发生之前写入FlexRAM，因此强烈建议使用必要的字节数目配置快速写入模式，以保证在掉电之前写入所有字节。

硬件必须保证系统供电的时间，直到最后一个字的写入。所需时间可从数据手册中获得，它取决于写入的字节数和每32位写入所需的时间。

下方流程图显示了一种通过使用低电压检测（LVD）中断程序来处理快速写入过程的方法。有关低电压检测的详细信息，请参阅参考手册中的第39.5节低电压检测系统。

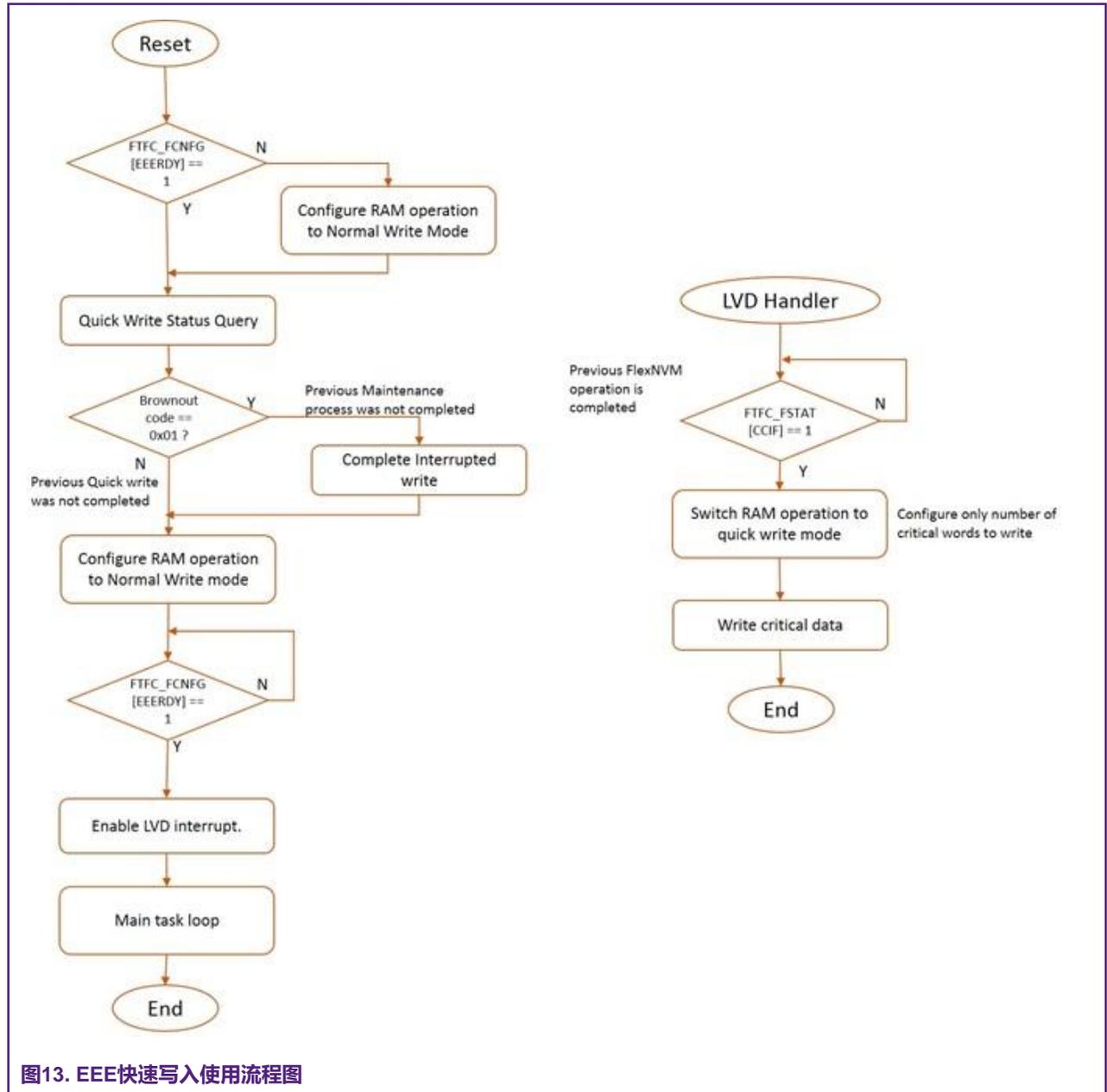


图13. EEE快速写入使用流程图

10 修订历史

修订版本号	日期	重大变更
0	2017年6月	初版发布
1	2018年3月	<ul style="list-style-type: none">更新了上电建议Power-On recommendation，修复了复位顺序加载的FlexRAM的说明更新了程序分区命令要求在程序分区命令示例中添加了注释更新了S32K1xx新的快速写入模式：添加了有关图11的快速写入参考增加了针对EEPROM用例的附录A第20页增加了功耗模式转换在第12页的FlexRAM 用于 EEPROM 快速写入增加了新的注释
2	2019年5月	<ul style="list-style-type: none">在第14页增加了S32K1xx EEE ECC错误处理，并在第15页增加了EEPROM可用于快速写入时启动，用以增加ECC处理和初始化的说明。

How To Reach Us

Home Page:

nxp.com

Web Support:

nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including "typicals," must be validated for each customer application by customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX, EMBRACE, GREENCHIP, HITAG, I2C BUS, ICODE, JCOP, LIFE VIBES, MIFARE, MIFARE CLASSIC, MIFARE DESFire, MIFARE PLUS, MIFARE FLEX, MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TRENCHMOS, UCODE, Freescale, the Freescale logo, AltiVec, C-5, CodeTEST, CodeWarrior, ColdFire, ColdFire+, C-Ware, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Qonverge, Ready Play, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, SMARTMOS, Tower, TurboLink, and UMEMS are trademarks of NXP B.V. All other product or service names are the property of their respective owners. ARM, AMBA, ARM Powered, Artisan, Cortex, Jazelle, Keil, SecurCore, Thumb, TrustZone, and μ Vision are registered trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. ARM7, ARM9, ARM11, big.LITTLE, CoreLink, CoreSight, DesignStart, Mali, mbed, NEON, POP, Sensinode, Socrates, ULINK and Versatile are trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.

© NXP B.V. 2019.

All rights reserved.

For more information, please visit: <http://www.nxp.com>

For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: May 2019

Document identifier: AN11983

