

Subsystem Design

使用闪存 (A 型) 仿真 EEPROM



1 说明

该子系统演示了如何在应用中实现电可擦除可编程只读存储器 (EEPROM) 仿真 (A 型)。EEPROM 仿真允许器件在闪存存储器中仿真 EEPROM，并使其具有与 EEPROM 等效的耐用性。使用闪存存储器可获得以下功能：

- 意外断电后保留数据
- 适用于不同应用的灵活结构
- 用户配置的擦除操作

下载该示例的代码。

MSPM0 有两种类型的 EEPROM 仿真库。A 型是使用静态随机存取存储器 (SRAM) 缓冲器存储一个大型数据块 (64 字节、128 字节或 256 字节)。有关库的详细信息，请参阅 [EEPROM 仿真 A 型设计](#) 应用说明。B 型是存储许多带有项目标识符的小数据项 (16 位或 32 位)。有关库的详细信息，也请参阅 [EEPROM 仿真 B 型设计](#) 应用说明。

该子系统介绍了 A 型的用法。有关 B 型子系统,请参阅[使用闪存 \(B 型 \) 子系统设计的仿真 EEPROM](#)。

图 1-1 展示了该子系统的功能图。

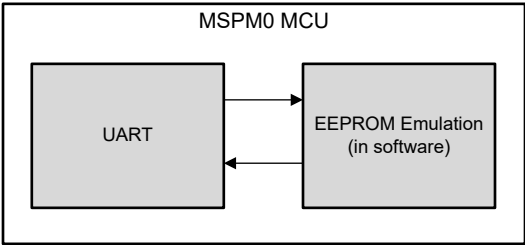


图 1-1. 子系统功能方框图

2 所需外设

此应用需要闪存。

表 2-1. 所需外设

子块功能	外设使用	注释
闪存 API	(1 ×) 闪存	在代码中称为 <i>FLASHCTL</i>

3 设计步骤

- 添加 EEPROM 仿真库。MSPM0 软件开发套件 (SDK) 包括 EEPROM 仿真库。

备注

EEPROM 仿真库基于闪存 API，因此也需要来自 SDK 的 drivelib。

对于 A 型，需要包含以下文件：

- a. eeprom_emulation_type_a.c
- b. eeprom_emulation_type_a.h

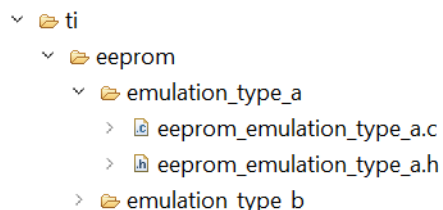


图 3-1. EEPROM 仿真文件

- 在 eeprom_emulation_type_a.h 的代码中添加包含路径。

```
#include <ti/eeprom/emulation_type_a/eeprom_emulation_type_a.h>
```

用户可以在 eeprom_emulation_type_a.h 中修改起始地址、要使用的扇区数和记录大小。用于 EEPROM 仿真的默认闪存地址为 0x00001000，默认使用两个扇区，因此占用 0x00001000-0x000017ff。此外，仿真 EEPROM 的默认大小为 128 字节。

- #define EEPROM_EMULATION_ADDRESS (0x00001000)
- #define EEPROM_EMULATION_SECTOR_ACCOUNT (2)
- #define EEPROM_EMULATION_RECORD_SIZE (128)

- 将全局数组定义为随机存取存储器 (RAM) 中的缓冲器。每次打开系统电源时，都会使用初始函数将仿真 EEPROM 的数据从闪存加载到该缓冲器中。数组大小应等于步骤 2 中的记录大小。

- uint32_t EEPROMEmulationBuffer[EEPROM_EMULATION_DATA_SIZE / sizeof(uint32_t)] ;

4. 在 main() 的开头添加初始函数，通常在 SYSCFG_DL_init() 之后。此操作可正确格式化相关的闪存区域并正确分配全局变量。在步骤 3 中，初始函数 EEPROM_TypeA_init() 还会搜索活动记录并将数据从闪存加载到缓冲器。
 - EEPROM_TypeA_init(&EEPROMEmulationBuffer[0]);

```
/* Initialize the virtual EEPROM */
EEPROMEmulationState = EEPROM_TypeA_init(&EEPROMEmulationBuffer[0]);
if (EEPROMEmulationState != EEPROM_EMULATION_INIT_OK) {
    __BKPT(0);
}
```

图 3-2. EEPROM 初始化

5. 用户可以在初始化后根据需要读取或修改 RAM 中的缓冲器。当需要存储从缓冲器到闪存的数据时，请调用 EEPROM_TypeA_writeData() 在闪存中创建新记录。

备注

断电后，RAM 中的数据会丢失。为实现断电后的数据存储，请至少执行一次该功能。

- EEPROM_TypeA_writeData(&EEPROMEmulationBuffer[0]);

```
/* Store the EEPROMEmulationBuffer to the flash to be a new record */
EEPROMEmulationState =
    EEPROM_TypeA_writeData(&EEPROMEmulationBuffer[0]);
if (EEPROMEmulationState != EEPROM_EMULATION_WRITE_OK) {
    __BKPT(0);
}
```

图 3-3. EEPROM 写入

6. 根据 gEEPROMTypeAEraseFlag 添加擦除函数。再次写入数据之前需要擦除闪存，擦除的最小单位为扇区。对于 EEPROM 仿真，在一个扇区已满后，设置 gEEPROMTypeAEraseFlag。用户可以根据标志调用 EEPROM_TypeA_eraseLastSector()。例如，在步骤 5 中的 EEPROM_TypeA_writeData() 之后添加以下代码。用户还可以根据需要选择一个适当的时间点来擦除整个扇区。
 - EEPROM_TypeA_eraseLastSector();

```
if (gEEPROMTypeAEraseFlag == 1) {
    /* In this demo, when the sector is full, it will be erased immediately */
    EEPROM_TypeA_eraseLastSector();
    gEEPROMTypeAEraseFlag = 0;
}
```

图 3-4. EEPROM 擦除

在完成步骤 1 到步骤 6 之后，将在应用中实现 EEPROM 仿真 A 型。请参阅节 5 了解流程。

4 设计注意事项

1. `eeeprom_emulation_type_a.h` 中有三个用户可配置的参数。可根据应用程序的要求相应地配置这些参数。要设置适当的参数，请参阅 [EEPROM 仿真 A 型设计](#) 应用说明中的 *应用方面* 部分。
 - a. 记录大小：64、128 或 256 字节
 - b. 使用的扇区数：至少 2 个
 - c. 扇区地址
2. 要评估闪存使用情况和耐写次数，请参阅 [EEPROM 仿真 A 型设计](#) 应用说明中的 *应用方面* 部分。
3. 如果在 `EEPROM_TypeA_writeData` 或 `EEPROM_TypeA_eraseLastSector` 期间断电，则可能会损坏数据或标头。

要检测损坏并从损坏中恢复，请执行 `EEPROM_TypeA_init`。加电后立即调用 `EEPROM_TypeA_init` 会检查所有记录的标头，以确认 EEPROM 仿真的数据存储是否正确，并在必要时执行格式修复。

在 EEPROM 仿真的结构中，标头可显示相应记录的状态。一共有四种状态。[节 3](#) 中详细介绍了这四种状态之间的变化。

5 软件流程图

图 5-1 显示了 [EEPROM 仿真 A 型](#) 的代码流程图，其中介绍了如何在应用代码中添加函数来实现 EEPROM 仿真。这里需要三个功能：`EEPROM_TypeA_init`、`EEPROM_TypeA_writeData`、`EEPROM_TypeA_eraseLastSector`。

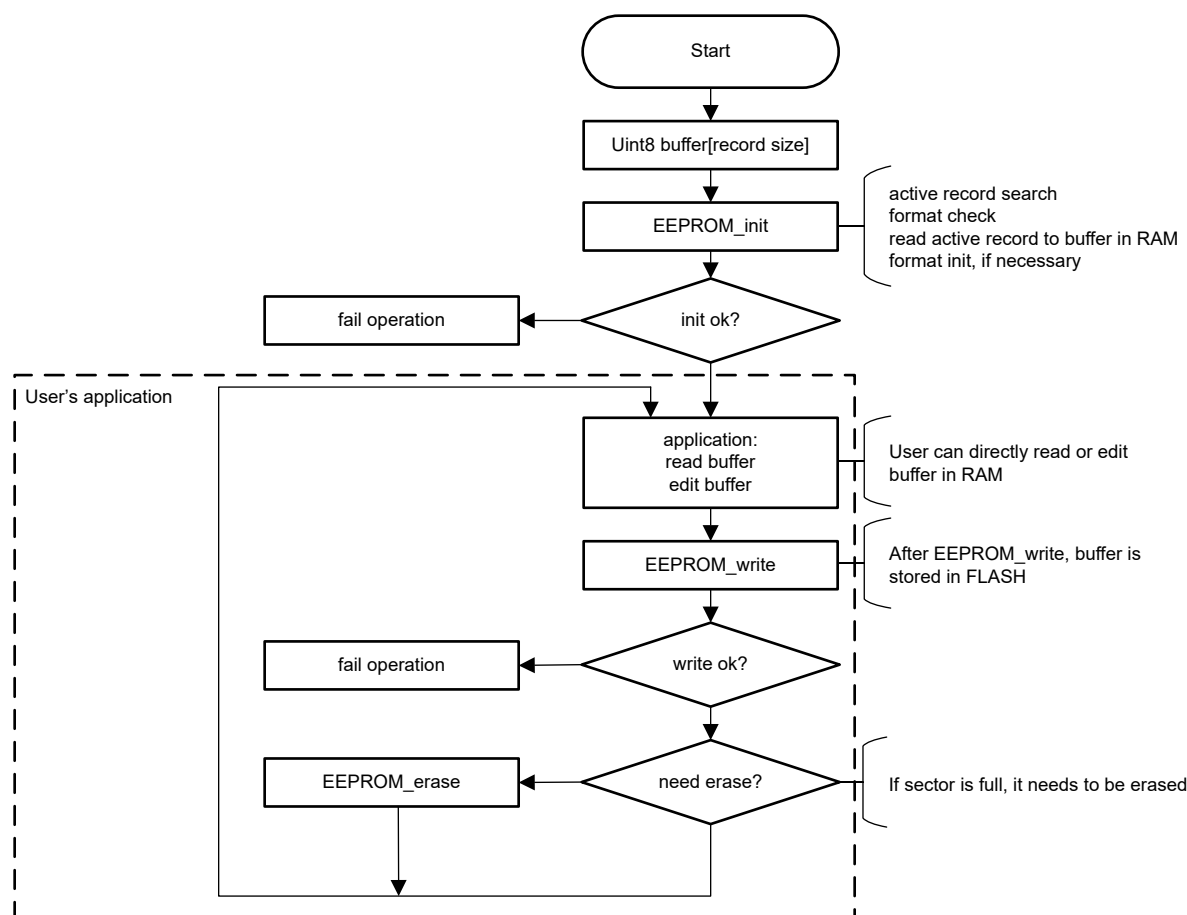


图 5-1. 应用软件流程图

6 应用代码

实现此功能需要六个函数。除 节 5 中提到的三个函数外，其余三个函数主要由 EEPROM_TypeA_init 调用。

- EEPROM_TypeA_init
- EEPROM_TypeA_writeData
- EEPROM_TypeA_eraseLastSector
- EEPROM_TypeA_readData
- EEPROM_TypeA_searchCheck
- EEPROM_TypeA_repairFormat

此外，七个全局变量用于记录 EEPROM 仿真的状态。有四个全局变量用于跟踪活动记录。

- uint32_t gActiveRecordAddress
- uint32_t gNextRecordAddress ;
- uint16_t gActiveRecordNum ;
- uint16_t gActiveSectorNum ;

gActiveRecordAddress 和 gNextRecordAddress 用于存储活动记录相关地址。

gActiveRecordNum 和 gActiveSectorNum 用于跟踪活动记录的位置。

有三个全局变量用于表示标志。

- bool gEEPROMTypeASearchFlag ;
- bool gEEPROMTypeAEraseFlag ;
- bool gEEPROMTypeAFormatErrorFlag ;

当存在活动记录时设置 gEEPROMTypeASearchFlag。

当扇区已满且需要擦除时设置 gEEPROMTypeAEraseFlag。

当发现格式错误时设置 gEEPROMTypeAFormatErrorFlag。

7 其他资源

- 德州仪器 (TI) , [EEPROM 仿真 A 型设计应用说明](#)
- 德州仪器 (TI) , [EEPROM 仿真 B 型设计应用说明](#)
- 德州仪器 (TI) , [使用闪存 \(B 型 \) 子系统设计的仿真 EEPROM](#)
- 德州仪器 (TI) , [下载 MSPM0 SDK](#)
- 德州仪器 (TI) , [详细了解 SysConfig](#)

8 修订历史记录

注：以前版本的页码可能与当前版本的页码不同

Changes from Revision * (February 2025) to Revision A (August 2025)	Page
• 删除了“兼容器件”部分	4

商标

所有商标均为其各自所有者的财产。

重要通知和免责声明

TI“按原样”提供技术和可靠性数据（包括数据表）、设计资源（包括参考设计）、应用或其他设计建议、网络工具、安全信息和其他资源，不保证没有瑕疵且不做任何明示或暗示的担保，包括但不限于对适销性、某特定用途方面的适用性或不侵犯任何第三方知识产权的暗示担保。

这些资源可供使用 TI 产品进行设计的熟练开发人员使用。您将自行承担以下全部责任：(1) 针对您的应用选择合适的 TI 产品，(2) 设计、验证并测试您的应用，(3) 确保您的应用满足相应标准以及任何其他功能安全、信息安全、监管或其他要求。

这些资源如有变更，恕不另行通知。TI 授权您仅可将这些资源用于研发本资源所述的 TI 产品的相关应用。严禁以其他方式对这些资源进行复制或展示。您无权使用任何其他 TI 知识产权或任何第三方知识产权。您应全额赔偿因在这些资源的使用中对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，TI 对此概不负责。

TI 提供的产品受 [TI 的销售条款](#) 或 [ti.com](#) 上其他适用条款/TI 产品随附的其他适用条款的约束。TI 提供这些资源并不会扩展或以其他方式更改 TI 针对 TI 产品发布的适用的担保或担保免责声明。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

邮寄地址：Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
版权所有 © 2025，德州仪器 (TI) 公司