

Subsystem Design

# 带闪存的仿真 EEPROM ( B 型 )



## 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 ( B 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。使用闪存存储器可以实现以下功能：

- 意外断电后的数据保留
- 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。

该子系统展示了 B 型的用法。有关 A 型子系统的信息,请参阅 [带闪存的仿真 EEPROM \( A 型 \)](#) 子系统设计。

图 1-1 展示了该子系统的功能图。

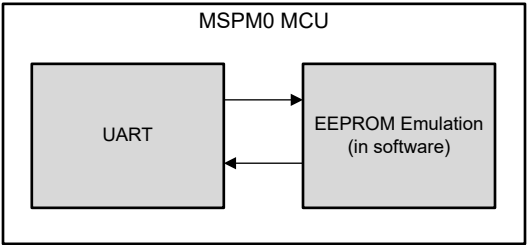


图 1-1. 子系统功能方框图

## 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

1. 添加 EEPROM 仿真库。MSPM0 软件开发套件 (SDK) 包含 EEPROM 仿真库。

#### 备注

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

对于 B 型，需要包含以下文件：

- a. eeprom\_emulation\_type\_b.c
- b. eeprom\_emulation\_type\_b.h

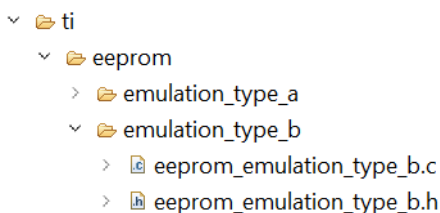


图 3-1. EEPROM 仿真文件

2. 在 eeprom\_emulation\_type\_b.h 代码中添加包含路径。

```
#include <ti/eeprom/emulation_type_b/eeprom_emulation_type_b.h>
```

用户可以修改 eeprom\_emulation\_type\_b.h 中的起始地址、组数（至少两个）和一个组中的扇区数（至少一个）。用于 EEPROM 仿真的默认闪存地址为 0x00001400，默认使用六个扇区（分三组，每组两个扇区），因此占用 0x00001400-0x00002bff。

- a. #define EEPROM\_EMULATION\_ADDRESS (0x00001400)
- b. #define EEPROM\_EMULATION\_GROUP\_ACCOUNT (3)
- c. #define EEPROM\_EMULATION\_SECTOR\_INGROUP\_ACCOUNT (2)

3. 添加 initialize 函数 ( 在 main() 开头 ), 通常在 SYSCFG\_DL\_init() 之后。这样就可以正确格式化相关的闪存区域并正确分配全局变量, 以跟踪活动组和最新数据项。

- EEPROMEmulationState = EEPROM\_TypeB\_init();

```
/* Initialize */
EEPROMEmulationState = EEPROM_TypeB_init();
if (EEPROMEmulationState == EEPROM_EMULATION_INIT_ERROR ||
    EEPROMEmulationState == EEPROM_EMULATION_TRANSFER_ERROR) {
    /* If there is an error, it will stop here */
    __BKPT(0);
}
```

图 3-2. EEPROM 仿真初始化

4. 使用 EEPROM\_TypeB\_write 通过 16 位标识符写入 32 位数据。

- EEPROMEmulationState = EEPROM\_TypeB\_write(var\_id, var\_data);

```
/* Write operation */
EEPROMEmulationState = EEPROM_TypeB_write(var_id, var_data);
if (EEPROMEmulationState != EEPROM_EMULATION_WRITE_OK) {
    /* If there is an error, it will stop here */
    __BKPT(0);
}
```

图 3-3. EEPROM 仿真写入

5. 使用 EEPROM\_TypeB\_readDataItem 根据输入标识符读取数据。

- var\_data\_read = EEPROM\_TypeB\_readDataItem(var\_id);

```
/* Read operation */
var_data_read = EEPROM_TypeB_readDataItem(var_id);
if (gEEPROMTypeBSearchFlag == 0) {
    /* If identifier is not found, it will stop here */
    __BKPT(0);
}
```

图 3-4. EEPROM 仿真读取

6. 根据 gEEPROMTypeBEraseFlag 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 B 型, 在一个组已满后, 设置 gEEPROMTypeBEraseFlag。用户可以根据标志调用 EEPROM\_TypeB\_eraseGroup()。例如, 在步骤 4 中的 EEPROM\_TypeB\_write() 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组。

- EEPROM\_TypeB\_eraseGroup();

```
/* Erase operation */
if (gEEPROMTypeBEraseFlag == 1) {
    /*
     * In this demo, gEEPROMTypeBEraseFlag is checked after write
     * operation. If the group is full, it will be erased immediately
     */
    __BKPT(0);
    EEPROM_TypeB_eraseGroup();
    gEEPROMTypeBEraseFlag = 0;
}
```

图 3-5. EEPROM 仿真擦除

完成步骤一到步骤六后, 就在应用中实现了 EEPROM 仿真 B 型。请参阅节 5 了解流程。

## 4 设计注意事项

1. `eeeprom_emulation_type_b.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 B 型设计](#) 应用说明中的 *应用方面* 部分。
  - a. 组数：至少 2 个
  - b. 一个组中的扇区数：至少 1 个
  - c. 扇区地址

2. 数据项的数量与组中的扇区数量直接相关。

$$\text{Number of data items} = \frac{\text{Sector size}}{\text{Data item size}} \times \text{Number of sectors in one group} - 1 \quad (1)$$

选择正确数量的数据项至关重要。关键是根据应用需要存储的变量数量进行选择。如果变量的数量接近数据项的数量，则在更新这些变量的值时会频繁进行转移。如果变量的数量远远少于数据项的数量，则意味着组的大小相对较大，并会在转移、擦除和搜索等操作中花费额外的时间。

建议的标识符数量是最大数据项数的三分之一到一半。

3. 要评估闪存使用情况和耐写次数，请参阅 [EEPROM 仿真 B 型设计](#) 应用说明中的 *应用方面* 部分。
4. 如果在 `EEPROM_TypeB_write` 或 `EEPROM_TypeB_eraseGroup` 期间断电，则可能会损坏数据。

要检测损坏并从中恢复，请执行 `EEPROM_TypeB_init`。上电后立即调用 `EEPROM_TypeB_init`。  
`EEPROM_TypeB_init` 会检查所有组的标头，以确认 EEPROM 仿真的数据存储是否正确。

在 EEPROM 仿真的结构中，标头可显示相应组的状态。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 展示了 *EEPROM 仿真 B 型* 的代码流程图，其中介绍了如何在应用代码中添加函数以实现 EEPROM 仿真。这里需要三个函数：EEPROM\_TypeB\_init、EEPROM\_TypeB\_write、EEPROM\_TypeB\_readDataItem，以及 EEPROM\_TypeB\_eraseGroup。

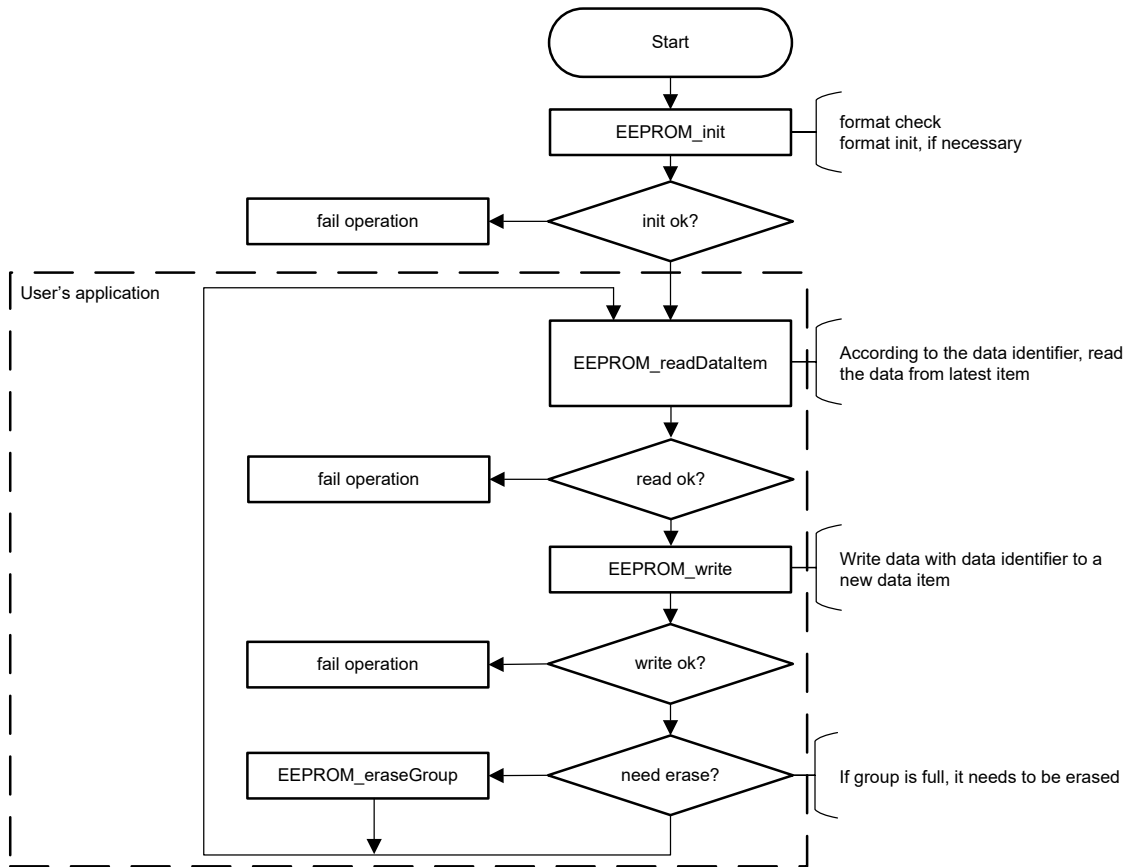


图 5-1. 应用软件流程图

## 6 应用代码

总共有六个函数。前四个函数由用户直接调用。最后两个函数由以下函数调用。

- EEPROM\_TypeB\_init
- EEPROM\_TypeB\_write
- EEPROM\_TypeB\_readDataItem
- EEPROM\_TypeB\_eraseGroup
- EEPROM\_TypeB\_findDataItem
- EEPROM\_TypeB\_transferDataItem

以下两个全局变量用于跟踪活动组：

- uint16\_t gActiveDataItemNum；
- uint16\_t gActiveGroupNum；

gActiveDataItemNum 用于记录数据项的数量。

gActiveGroupNum 用于记录活动组。

有两个全局变量用于表示标志：

- bool gEEPROMTypeBSearchFlag；
- bool gEEPROMTypeBEraseFlag；

当 EEPROM\_TypeB\_readDataItem 根据输入标识符找到数据项时，会设置 gEEPROMTypeBSearchFlag。

当组已满且需要擦除时，会设置 gEEPROMTypeBEraseFlag。

所有全局变量都在 eeprom\_emulation\_type\_b.c 中定义。

## 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)，[带闪存的仿真 EEPROM \(A 型\) 子系统设计](#)
- 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 .....                                                 | 2    |

## 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