

# 面向 Kinetis E 系列微控制器的 EEPROM 仿真驱动器

作者: 王鹏

可字节编程或字编程且可擦除的电可擦可编程只读存储器 (EEPROM) 在汽车电子控制单元 (ECU) 中经常使用。编程和擦除操作的灵活性使得它适合用于断电时必须保留的以及在运行时需要单独更新的应用变量的数据存储。对于不带 EEPROM 存储器的设备, 可以通过 EEPROM 仿真软件, 使用页面可擦除的 Flash 存储器来仿真 EEPROM。

该演示代码显示了如何在 Flash 中仿真 EEPROM。

## 内容

1. 简介 .....	1
2. 软件架构 .....	2
3. EEPROM 仿真驱动器示例 .....	7
4. 结语 .....	7
5. 参考 .....	8
6. 术语表 .....	8
7. 修订历史记录 .....	8

## 1 简介

Kinetis E 系列微控制器 (KE02 除外) 没有片上 EEPROM, 但是, 这些器件可以使用本应用笔记中所述的软件在片上 Flash 存储器中存储非易失性数据, 因此, 节省了购置外部 EEPROM 的成本。

一个可擦除 Flash 单元相当于一个扇区。由于 Flash 编程只能在已擦除的地址中进行, 因此, 在编程之前必须擦除 Flash 存储器。不使用软件算法直接在 Flash 中编程数据会导致频繁擦除 Flash, 频繁的擦除会缩短 Flash 的寿命, 增加数据写入的时间。

面向 Kinetis E 系列的 EEPROM 仿真驱动器在 Flash 上实施固定长度的数据记录方案, 并具有以下功能:

- 组织数据项
- 初始化 EEPROM
- 检查 EEPROM 状态
- 读取、写入和删除数据项

包含一种用于保存数据的算法, 可避免直接和频繁写入 Flash

## 2 软件架构

图 1 演示了一个简单的 API 函数，供客户从 Flash 存储器中实施 EEPROM 仿真驱动器。

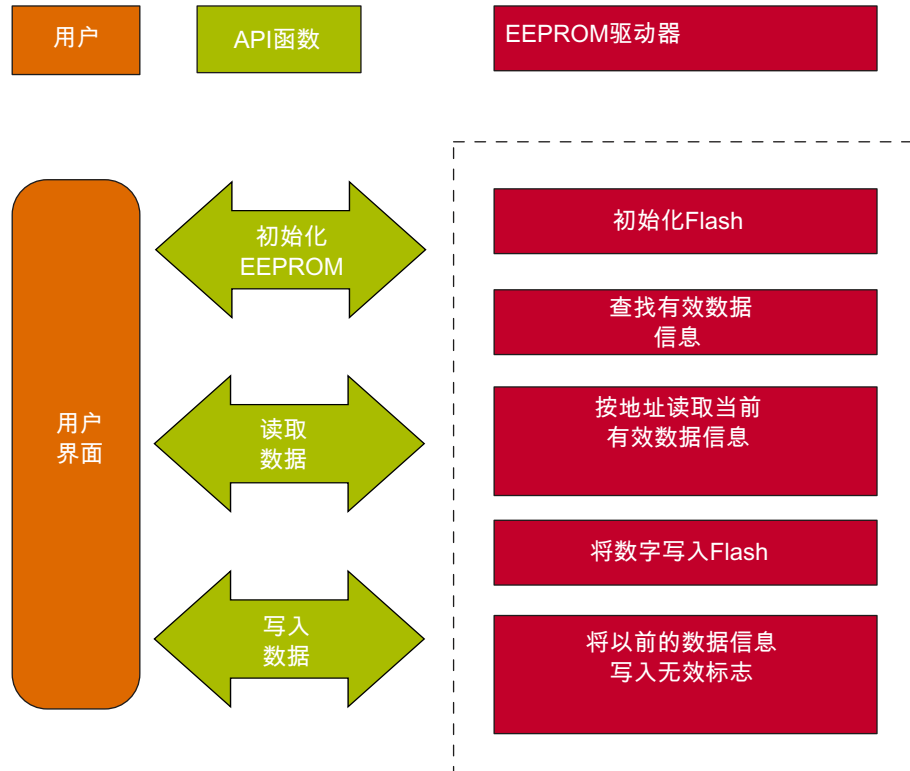


图 1. 时序图元素

下面列出了 API 函数：

```

uint8_t EE_Init(uint32_t *pCurrentAddress, uint32_t u32BusClock);
uint8_t EE_Write(EE_ItemInfoPtr pWrItemInfo, uint32_t *p32CurrentAddress);
uint8_t EE_Read(EE_ItemInfoPtr pRdItemInfo, uint32_t u32CurrentAddress);
uint8_t EE_SearchIndex(uint32_t *pCurrentAddress);
  
```

EEPROM 驱动器中定义了四种状态。在下面的内容以及整个文档中，每种状态称为项：

```

#define EE_ITEM_INFO_NULL0xff
#define EE_ITEM_INFO_PROCESSING0xe7
#define EE_ITEM_INFO_VALID0xa5
#define EE_ITEM_INFO_INVALID0x00
  
```

下面逐个描述了四种状态。使用此信息来实施状态机，以识别有效信息。

- **EE\_ITEM\_INFO\_NULL**  
表示当前数据项为 NULL。该数据项有效，并且可对其编程，以便将新数据写入当前地址。
- **EE\_ITEM\_INFO\_PROCESSING**  
表示当前数据项正在处理中。过程可以是：正在写入当前项、当前项不完整不适合整个更新的序列、向 Flash 写入当前项时断电。
- **EE\_ITEM\_INFO\_VALID**  
表示当前数据项有效。在此状态下，可以正常使用该数据项。
- **EE\_ITEM\_INFO\_INVALID**  
表示当前数据项无效。数据项已变旧，并且已被替换。

## 2.1 存储器映射

此 EEPROM 仿真算法保存数据项。用户还可以定义每个数据项的长度。有关编程说明，请参考文件“ee\_emulation.h”中的以下宏定义。

```
/* 这里确保长度能被 4 整除，使 Flash 操作遵循 4 字节对齐原则 */
#define EE_ITEM_INFO_LENGTH16
#if ((EE_ITEM_INFO_LENGTH%4) != 0 )
#error "please ensure EE information length is align with 4 bytes"
#endif
```

为了节省 Flash 存储空间并简化代码，定义的长度必须能被 4 整除。

一个数据项包含一个结构体，如下所示：

```
typedef struct
{
    uint8_t u8Flag; // 表示项状态
    uint8_t u8InfoBuff[EE_ITEM_INFO_LENGTH - 1]; // 数据项缓冲区
}EE_ItemInfoType,*EE_ItemInfoPtr;
```

此处的 u8Flag 表示项的状态。

图 2 演示了以典型方式为非易失性数据存储分配资源的 Kinetis E 系列的默认存储器映射。

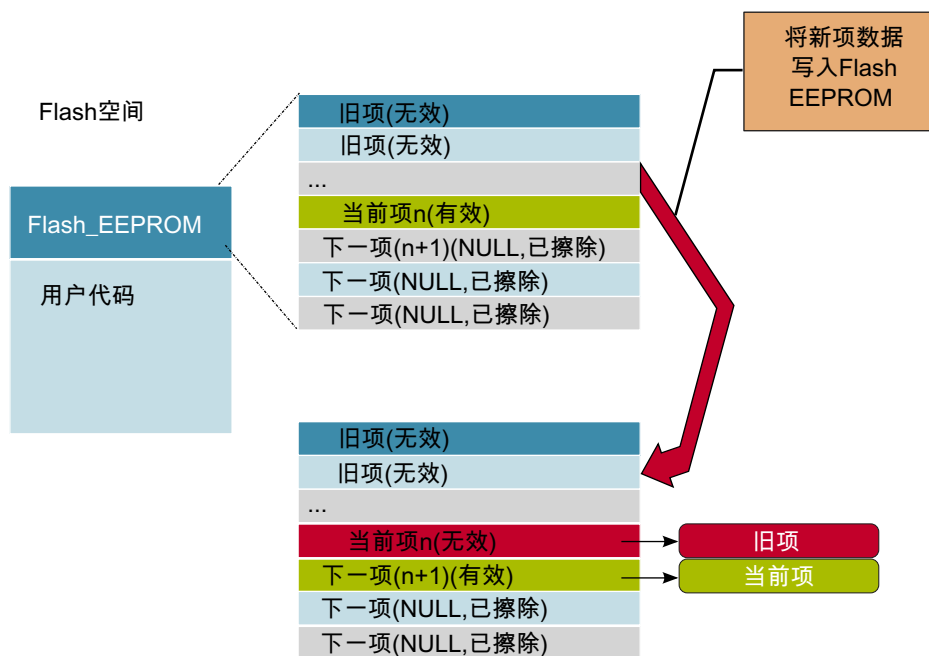


图 2. Flash 中的存储分配

## 2.2 EEPROM 寿命期优化

用户必须指定 EEPROM 仿真驱动器的起始地址和结束地址。还必须为 EEPROM 仿真驱动器、EEPROM 大小和每个数据项的长度保留足够的空间。

```
#define EE_START_ADDRESS0xC000// 起始地址
#define FLASH_PAGE_SIZE512// 每个页面或扇区的字节数
#define EE_PAGE_NUMBER2// EEPROM 的保留页面数
#define EE_END_ADDRESSEE_START_ADDRESS +FLASH_PAGE_SIZE*EE_PAGE_NUMBER-1
```

上面的宏定义了起始地址、用于 EEPROM 的页面数、每个页面（即扇区）的字节数，以及用于 EEPROM 仿真驱动器的扇区和结束地址。

最佳 EEPROM 寿命期是使用公式 1 中的公式估算的。

$$\text{寿命期} = Fc \times EE \text{ 的大小} / \text{项的长度} \tag{公式 1}$$

其中：

Fc—表示闪存编程周期。对于 Kinetis E 系列，典型的周期数为 100K。

EE 的大小—表示为 EEPROM 保留的 Flash 空间。

项的长度—表示每个项的数据长度。

### 2.3 EEPROM 初始化

要使用 EEPROM 仿真驱动器，需要调用“EE\_Init”函数来完成 EEPROM 初始化。此函数将执行 Flash 初始化，并搜索有效的 Flash 地址。它会搜索分配给 EEPROM 的整个 Flash 空间，查找并返回有效数据项的地址，如果找不到有效的数据项，则返回 NULL 数据项地址。

对于 Kinetis E 系列微控制器，需要初始化 Flash 以将 Flash 时钟配置为大约 1 MHz，否则将发生 Flash 操作错误。初始化 Flash 时钟后，可以擦除或编程 Flash。

调用初始化函数后，可以通过“EE\_Read”通知找到了有效数据项，并可以将新数据项写入 Flash。

图 3 提供了详细的初始化流程图。

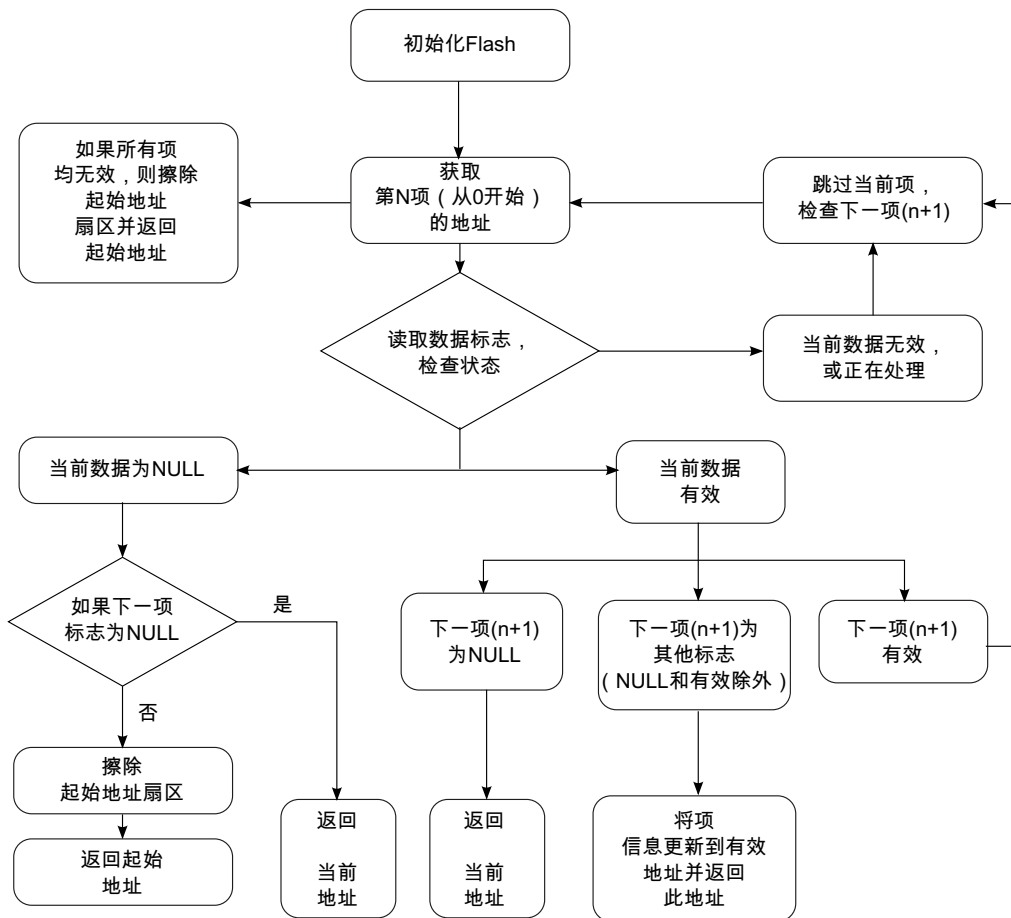


图 3. 初始化流程图

## 2.4 将数据写入 EEPROM

要使用 EEPROM 仿真驱动器更新新的数据项，请调用“EE\_Write”函数。该写入函数会将新数据更新到下一个数据项的地址，然后将当前数据项的标志更改为无效。这可以防止在 Flash 写入过程中，由于意外的原因（例如断电）而导致新数据项损坏。使用状态机提供的算法可以还原数据信息，即旧的有效信息。图 4 演示了如何将数据写入 EEPROM 仿真驱动器。

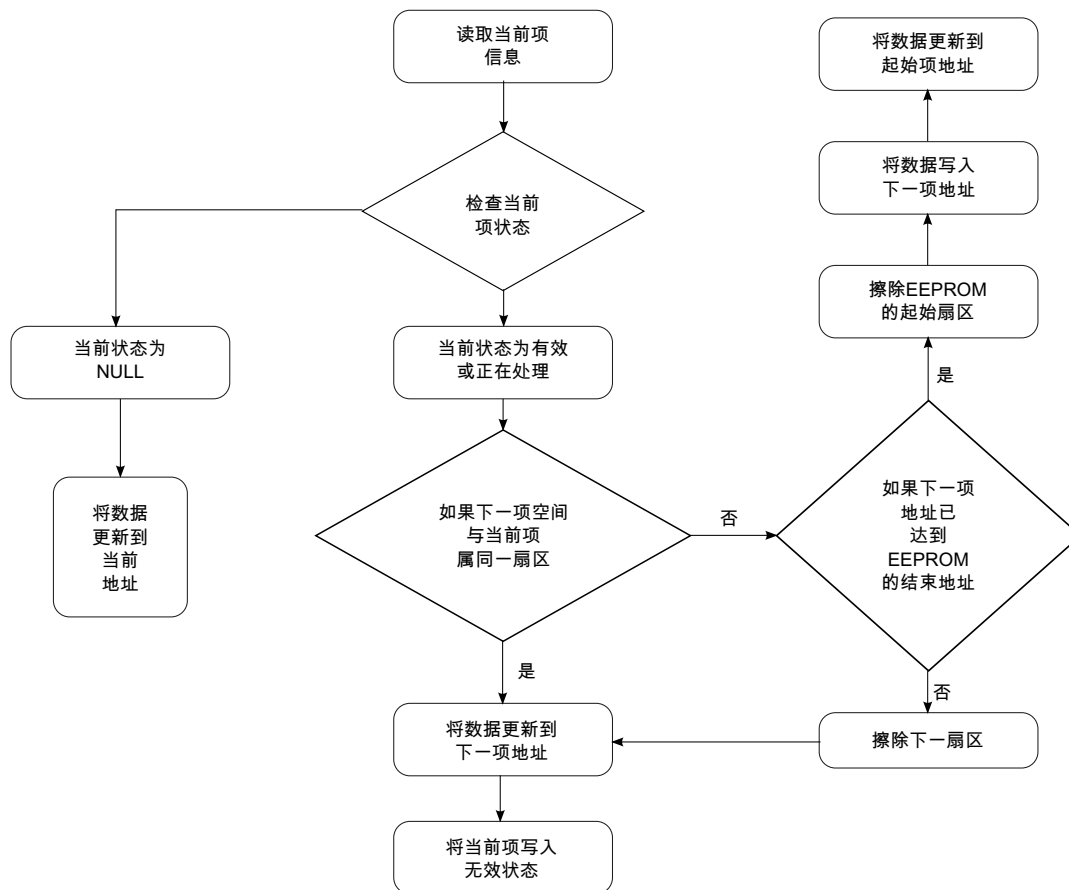


图 4. 写入 EEPROM 流程图

EEPROM 仿真驱动器首先会检查当前数据项的状态，然后确定如何将新数据项更新到下一个地址：

- 如果当前状态为 NULL，则它会直接更新下一个数据项的地址。
- 如果当前状态为 VALID，则它会确定下一个数据项的状态，如果该状态为 NULL，则将新数据项保存到下一个地址，然后将当前数据项的标志更改为 INVALID。

将数据项更新到 Flash 中后，它首先会将标志“EE\_ITEM\_INFO\_PROCESSING”写入新的数据项。数据项更新完成后，它将标志更改为“EE\_ITEM\_INFO\_VALID”。如果在“EE\_ITEM\_INFO\_PROCESSING”更新过程中断电，当前数据项的标志将更新为“EE\_ITEM\_INFO\_INVALID”。电源恢复后，前一数据项的标志将先显示为“EE\_ITEM\_INFO\_VALID”，然后还原到新地址。该驱动器使用“EE\_Init()”将已用数据项的地址更改为“EE\_ITEM\_INFO\_INVALID”。图 5 演示了如何在更新出错后还原数据项的信息。

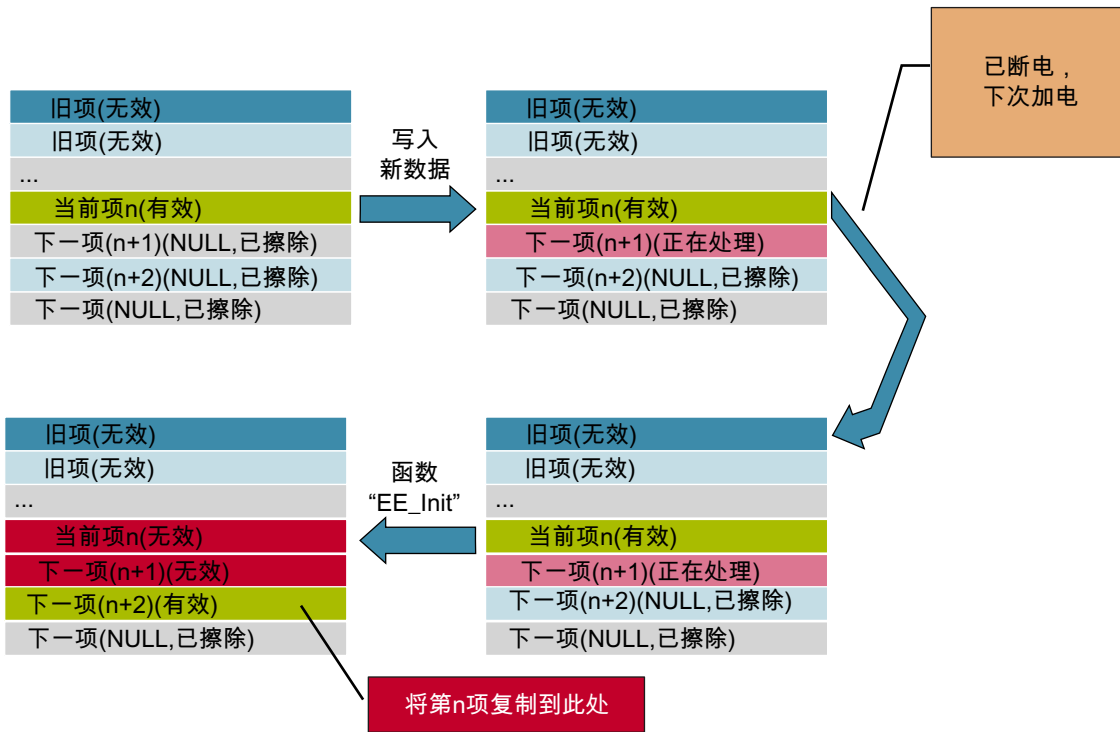


图 5. 从旧数据项中还原数据

## 2.5 从 EEPROM 中读取数据

此 API 函数可以从 Flash 中读取当前数据项，并将其复制到用户指定的 RAM 地址。

```
uint8_t EE_Read(EE_ItemInfoPtr pRdItemInfo, uint32_t u32CurrentAddress)
{
    if( pRdItemInfo->u8Flag == EE_ITEM_INFO_INVALID )
    {
        return FALSE;
    }
    memcpy((void *)pRdItemInfo,
           (uint8_t *)u32CurrentAddress, sizeof(EE_ItemInfoType));
    return TRUE;
}
```

### 3 EEPROM 仿真驱动器示例

图 6 显示了如何在此项目中添加文件，以使用 EEPROM 仿真驱动器。

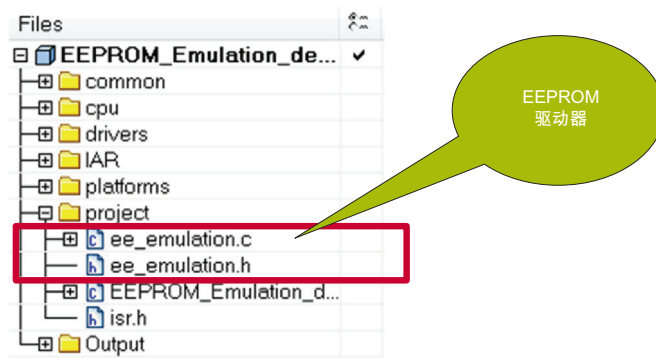


图 6. EEPROM 驱动器

必须同时使用 EEPROM 仿真驱动器和非易失性存储器驱动器。Flash 操作 (包括初始化、编程和擦除) 需要使用以下 API 函数。

```
uint16_t FLASH_Init(uint32_t BusClock);
uint16_t FLASH_Program(uint32_t wNVMTTargetAddress, uint8_t *pData, uint16_t sizeBytes);
uint16_t EEPROM_EraseSector(uint32_t wNVMTTargetAddress);
```

图 7 演示了 EEPROM 仿真驱动器的基本使用流程。

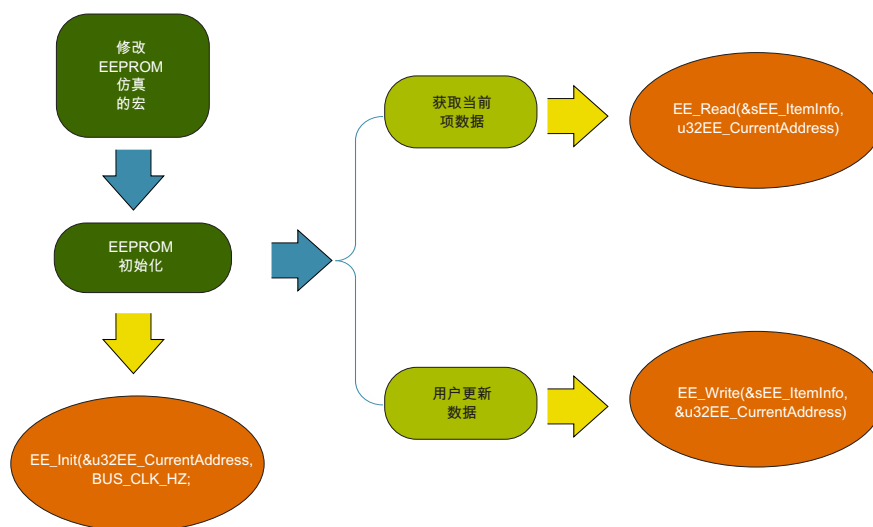


图 7. EEPROM 仿真驱动器的使用流程图

随附的 AN4903SW 中提供了一个软件演示，其中介绍了如何在 Kinetis KE06 上使用 EEPROM 仿真驱动器。软件驱动器在 FRDM-KE06Z 板上运行。该演示展示了一个用于向 EEPROM 写入新数据的简单应用。

### 4 结语

本文档介绍了在 Flash 中实施 EEPROM 的方法。使用此方法，用户可以在 Kinetis E 系列微控制器上快速实施 EEPROM 功能，以及集成软件算法以提高存取速度并优化非易失性存储器的寿命。本文档还介绍了在执行更新操作期间发生断电等意外状况时还原机器的功能。总而言之，客户可以方便地使用 EEPROM 并将它迁移到不同的平台。

## 5 参考

- 面向 Kinetis E 系列微控制器软件的 EEPROM 仿真驱动器 (document number: AN4903SW)
- KE04 子系列参考手册 (document number: MKE04P24M48SF0RM)
- KE06 子系列参考手册 (document number: MKE06P80M48SF0RM)
- KE04 子系列数据手册 (document number: MKE04P24M48SF0)
- KE06 子系列数据手册 (document number: MKE06P80M48SF0)
- 面向 MC9S12C32 的 AN2302 EEPROM 仿真
- 面向 M68HC908 微控制器的 AN3040 EEPROM 仿真驱动器

## 6 术语表

NVM	非易失性存储器
EEPROM	电可擦除可编程只读存储器
FCCOB	Flash 通用命令对象
WDOG	看门狗
MCG	多用途时钟发生器

## 7 修订历史记录

修订版 0 为本文档的初始版本。





**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飞思卡尔产品。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。

飞思卡尔保留对此处任何产品进行更改的权利，恕不另行通知。飞思卡尔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。飞思卡尔数据表和 / 或技术规格中可能提供了“典型”参数，这些参数在不同应用中可能并且确实不同，实际性能也可能会随时间变化。所有操作参数，包括“典型值”在内，必须由客户的技术专家根据每个客户应用进行验证。飞思卡尔未转让与其专利权及其他权利相关的许可。飞思卡尔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](http://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. ARM and the ARM Power logo are the registered trademarks of ARM Limited.

© 2014 Freescale Semiconductor, Inc.

