

Application Note

MSPM0 和 MSPM33 器件：通过存储体交换保留 IO 状态



Erik Vaughn

摘要

MCU 通常使用 GPIO 将多个信号驱动到 PCB 上的其他 IC。在 MCU 的生命周期内维持这些 GPIO 的电压电平至关重要。大多数 MCU 需要复位才能执行固件更新。这会导致既需要固件更新又需要保持特定 GPIO 电压的应用出现问题。要通过固件更新维持 GPIO 电压，MSPM0 和 MSPM33 器件提供了一种在执行引导过程所需的复位之前锁存 IO 状态的方法。要设置这些锁存器，器件必须进入称为关断的低功耗模式，但这会导致另外一个问题，即如何唤醒器件。本文档介绍了如何触发这些锁存器、如何在关断后自动唤醒 MCU、如何构建软件应用以实现此功能，以及启动过程的用例。

内容

1 简介.....	2
1.1 MSPM0 和 MSPM33 MCU 上的固件更新.....	3
1.2 MSPM0 和 MSPM33 上的功耗模式.....	4
2 IO 锁存工作原理.....	5
2.1 如何锁存 IO 状态.....	5
2.2 在引导过程中保持 IO 状态.....	6
3 围绕关断 IO 功能配置软件.....	7
3.1 配置软件以从关断模式唤醒.....	7
3.2 将状态保持 IO 与引导加载程序结合使用.....	8
3.3 关断存储器.....	9
4 总结.....	9
5 参考资料.....	9

商标

所有商标均为其各自所有者的财产。

1 简介

对于需要在 MCU 生命周期内保持 GPIO 状态的应用（例如 MCU 启用 LDO、偏置模拟信号或打开单独 IC 的应用），在 MCU 通电时保持这些 GPIO 电压状态变得至关重要。通常，有关 GPIO 输出电压和 IOMUX 配置的信息会保留在 MCU 的存储器映射寄存器 (MMR) 中。当 MCU 复位时，这些 MMR 会复位，导致这些外部电压信号丢失。对于 MCU 可能在主动驱动 GPIO 以正确触发固件更新时进行固件更新的用例，MCU 必须发出系统复位命令。此系统复位将导致 MCU 丢失 MMR 信息，从而导致 GPIO 电压丢失。这会导致问题，可通过在电路中添加额外的硬件来解决，而这会增加系统的总体成本。为了帮助防止这种额外成本，MSPM0 和 MSPM33 产品在 IOCell 结构内部包含锁存器，以通过更新保持 IO 状态。

MSPM0 和 MSPM33 器件的每个引脚上都提供 IOMUX 内部的这些锁存器。IOMUX 内部的这些锁存器可保持任何内部上拉/下拉电阻器和 GPIO 输出电压。为了触发这些锁存，MCU 必须进入关断模式和唤醒状态。该过程会强制 MCU 也通过主引导加载程序或次级引导加载程序。通过在更新应用程序映像之前添加这一步，可以根据需要适当地保持 GPIO 电压。

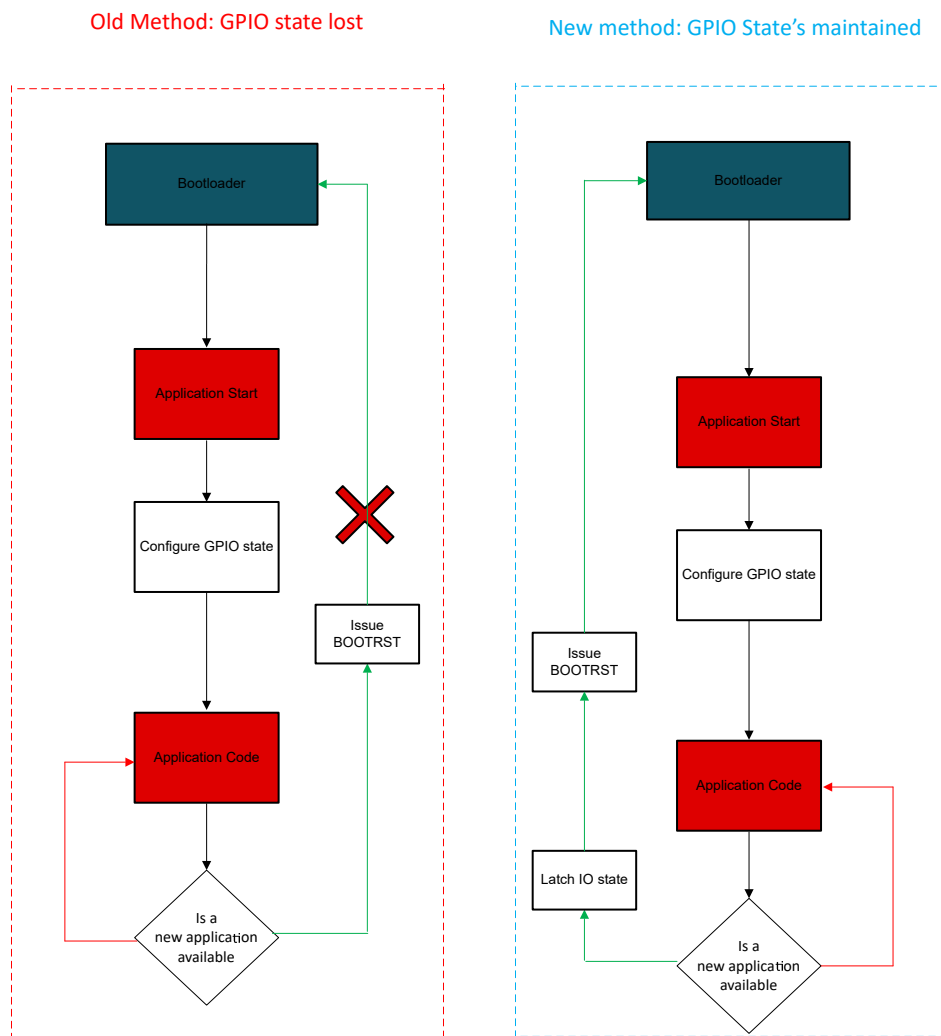


图 1-1. 标准引导过程与维持 GPIO 状态引导过程的比较

1.1 MSPM0 和 MSPM33 MCU 上的固件更新

MSPM0 和 MSPM33 器件有多种方法来支持固件更新过程。通常，这是使用主引导加载程序（有时称为 ROM 引导加载程序或 BSL）或次级引导加载程序（有时称为闪存引导加载程序）来实现的。这些引导加载程序利用一些通信外设从外部器件读取更新，以将新的应用程序映像写入代码存储器。这些引导加载程序中的一些提供安全功能，例如客户安全代码 (CSC)，该代码用于验证新的应用程序映像是否具有正确的签名，以保证新固件来自安全元素。前面提到的所有引导加载程序都遵循类似的流程，该流程要求 MCU 执行复位，以便通过 ROM 引导加载程序或次级引导加载程序正确运行引导加载程序。

MSPM0 和 MSPM33 提供各种复位选项，如 图 1-2 中所示。红色框显示了可能由 MCU 触发的不同类型的复位。图 1-2 图中还突出显示了 BCR 和 BSL。BSL 是引导加载程序，BCR 是引导配置例程，两者均由 ROM 引导加载程序提供。该 ROM 引导加载程序能够触发新的固件更新。然后，我们可以看到 BCR 和 BSL 框上方的任何复位都将触发我们需要的固件更新，例如欠压复位 (BOR)、上电复位 (POR) 和引导复位 (BOOTRST)。

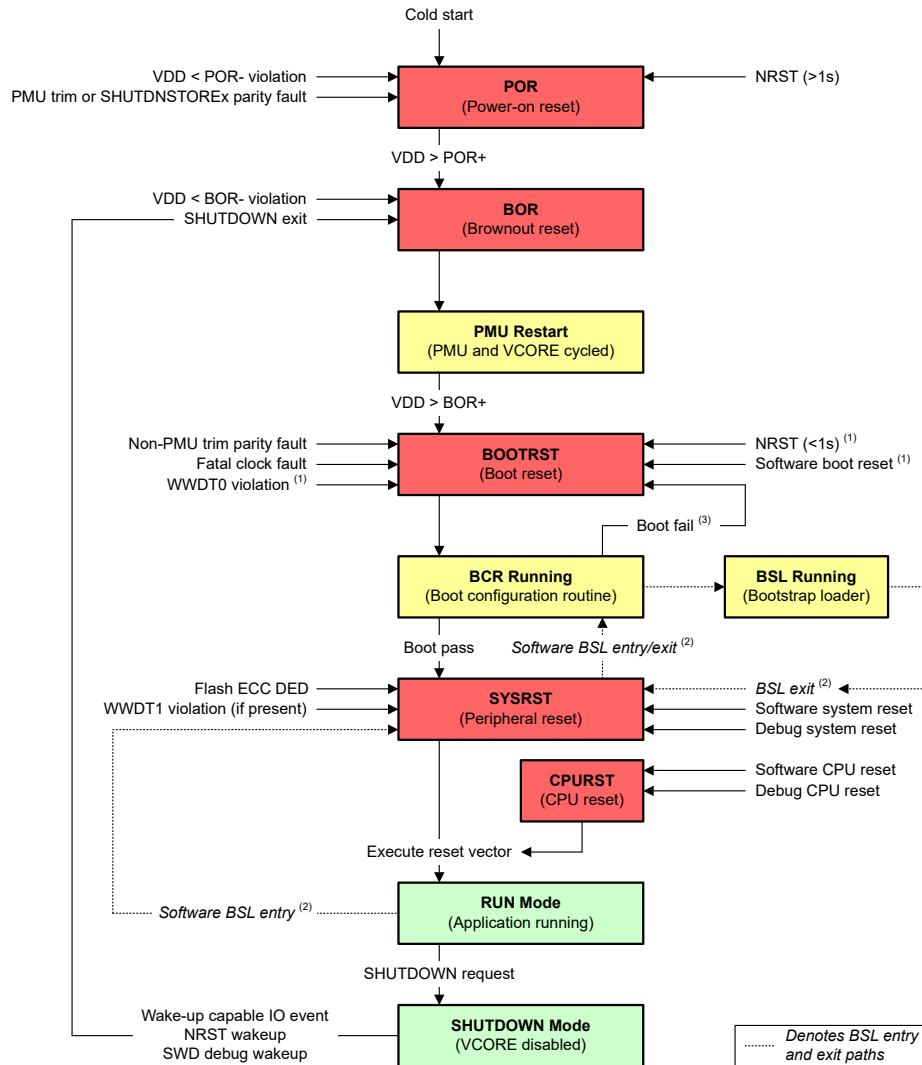


图 1-2. 器件复位流程

1.2 MSPM0 和 MSPM33 上的功耗模式

MSPM0 和 MSPM33 器件上有五种不同的功耗模式：RUN、SLEEP、STOP、STANDBY 和 SHUTDOWN。要触发这些 GPIO 状态，器件必须进入称为关断的低功耗模式状态。与其他模式相比，关断功耗模式的独特之处在于所有外设和 MCU 处于深度睡眠模式。在该模式下，只有配置的外设才能唤醒 MCU。请参阅图 1-3，将其作为不同工作模式的高电平基准。

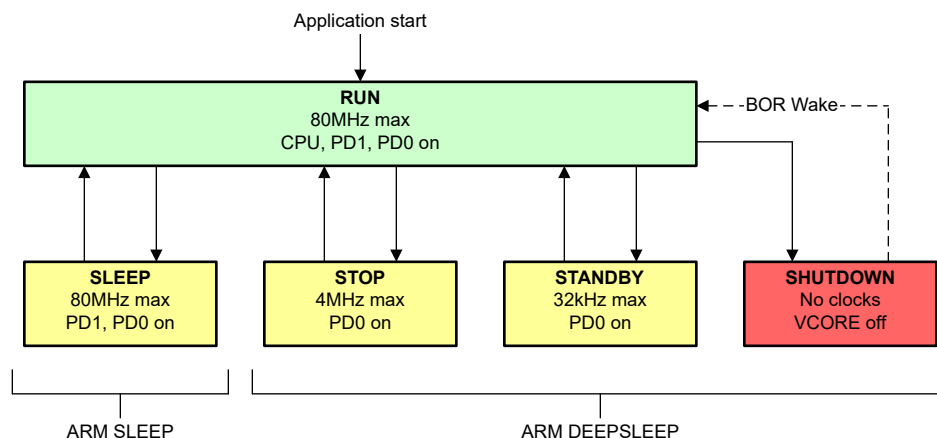


图 1-3. MSPM0G 工作模式

如图 1-3 所示，当退出设备的关断模式时，将触发 BOR。这会导致 MCU 触发复位，并再次启动应用程序代码。如果目标是执行需要在代码开头触发的固件更新，此操作非常方便。然而，在触发 BOR 之前，器件必须从关断模式唤醒。要将 MCU 从关断模式唤醒，请使用支持唤醒功能的 IO、调试连接或 NRST。要自动执行此过程，支持唤醒功能的 IO 是最简单的方法，而不是发出 NRST 或创建调试连接。本文件在节 2 中进一步讨论了这一点。

2 IO 锁存工作原理

要正确了解如何将锁存器用于安全启动过程，首先需要明白这些锁存器的触发过程、用于保持 IO 状态的状态以及如何自动执行唤醒过程。本节讨论了支持锁存 GPIO 电压电平的功能的硬件，如何通过进入关断模式触发这些锁存器，以及如何在进入关断模式后自动唤醒器件。

2.1 如何锁存 IO 状态

触发 IOMUX 锁存

IO 多路复用器内部有一个特定的锁存器可保持 GPIO 的逻辑电平。请参阅下图中的关断锁存器。一旦 MCU 进入关断模式，这些锁存器就会自动设置。锁存器被设置后，GPIO 状态将保持，直到 SHDINIOREL 寄存器中的 RELEASE 位被设置。以下是锁存和解锁 IO 状态所采取的步骤。

1. 当关断信号置为有效时，D-Q 会锁存输出路径捕获 DOUT、DRV (驱动强度)、PIPU (上拉使能) 和 PIPD (下拉使能)。
2. 锁存器保持的状态与 GPIO MMR 无关
3. 释放信号 (通过写入 SHDINIOREL 驱动) 会将锁存器复位，并将 IO 控制返回到正常的 MMR 驱动路径。

此状态通过 BOOTRST、BCR、BSL、SYSRST 和 CPURST (均为 在引导过程中保持 IO 状态 中显示的较低层级上的所有复位) 保持，直到软件通过 SHDINIOREL 释放锁存器。有关更多详细信息，请参阅技术参考手册的 PMCU 章节。

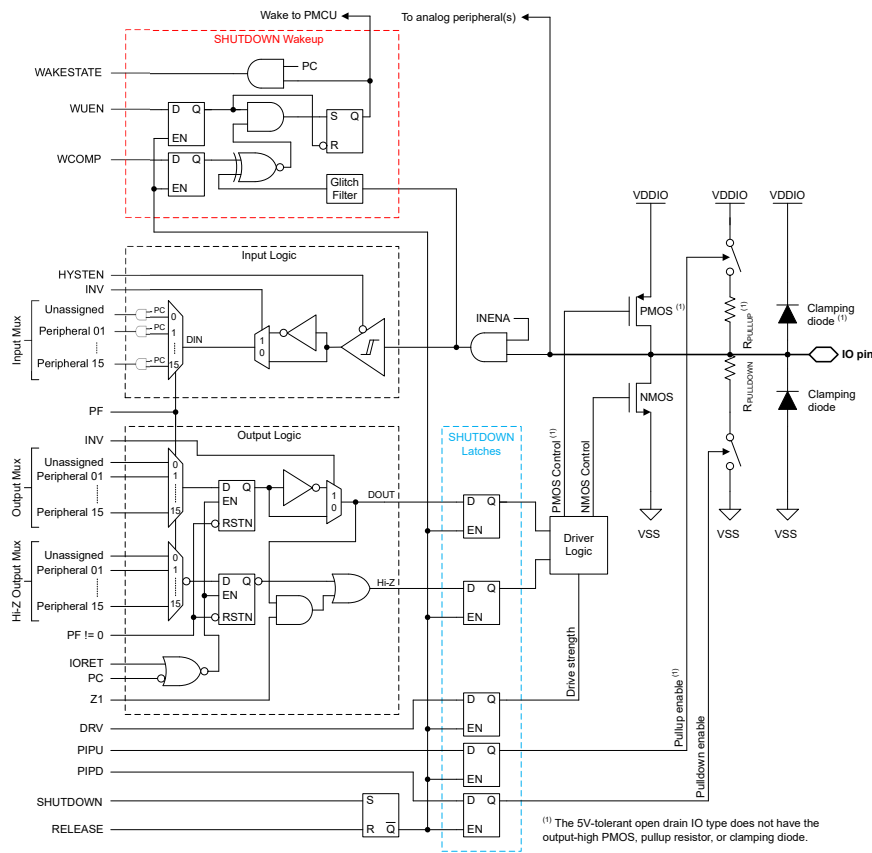


图 2-1. IOMUX 结构

从关断模式唤醒

要正确锁存 IO 状态，MCU 只需进入关断模式。要验证 MCU 可以立即唤醒而不会出现任何问题，可以使用内部上拉电阻器配置具有唤醒功能的单独 IO。如果该唤醒功能的逻辑电平为高电平，则 MCU 会在器件进入关断模式后立即唤醒。

要验证 MSPM0 或 MSPM33 器件是否具有关断功能 IO，请查看器件数据表。下表显示了 MSPM0 或 MSPM33 器件支持的通用 IO 类型以及每种 IO 类型的特性。如果 GPIO 是具有唤醒、ODIO 或 HDIO 的 SDIO，则它支持唤醒关断。

表 2-1. 按 IO 类型分类的数字 IO 功能

缓冲器类型	反转控制	驱动强度控制	迟滞控制	上拉电阻器	下拉电阻器	唤醒逻辑
SDIO (标准驱动)	Y			Y	Y	
带唤醒功能的 SDIO (标准驱动) ⁽¹⁾	Y			Y	Y	Y
HDIO (高驱动)	Y	Y		Y	Y	Y
HSIO (高速)	Y	Y		Y	Y	
ODIO (5V 容限开漏)	Y		Y		Y	Y

2.2 在引导过程中保持 IO 状态

由于 GPIO 寄存器信息丢失，在引导复位后 IO 状态会丢失，因此可以利用前面提到的 IOMUX 内部锁存器来保持 IO 状态，直到应用可以正确重新配置它们。为了触发这些锁存，MCU 必须进入关断模式，在 MCU 从关断模式唤醒后，器件会通过配置的 BCR、BSL 或任何次级引导加载程序。在此过程中，MCU 会正确保持 IO 状态，直到主应用随后可以正确重新配置 IO 状态并释放 IOMUX 锁存器。

要配置应用以锁存 IO 状态，请按如下步骤操作：

1. 将 GPIO 配置为所需状态。
2. 使用内部上拉电阻器将具有从关断状态唤醒功能的单个 GPIO 配置为在高电平时唤醒。
3. 将器件置于关断状态。
4. 唤醒后，在重新配置 MMR 之前，通过写入 SHDINIOREL 来释放 IO 锁存器。

通过执行上述步骤，器件会立即从关断模式唤醒并重新启动引导序列。从关断模式唤醒会触发 BOR。这会强制器件完成引导复位过程。请参阅下图，了解 BOR 之后发生的情况。

3 围绕关断 IO 功能配置软件

在了解 IO 锁存器背后的机制后，我们可以利用该功能构建一个结构，用于为软件提供通用架构，以将器件置于关断模式，同时保持 GPIO 状态。本节讨论了如何围绕 IOMUX 锁存器构建应用，以便 MCU 可以在保持 GPIO 状态的同时正确地通过引导加载程序。

3.1 配置软件以从关断模式唤醒

如果目标是通过 SYSRST 或其他类型的复位来复位 MCU，同时保持 IO 状态，用户可在 GPIO 上配置具有唤醒功能的内部上拉电阻器，并将 GPIO 设置为高电平时唤醒。完成此步骤后，用户可以将器件置于关断状态，MCU 会在无需其他输入的情况下唤醒。

要配置 MCU 以锁存 IO 状态，需要执行两个主要步骤：

1. 使用内部上拉电阻器将具有从关断模式唤醒功能的 IO 配置为在接收到高电压时唤醒。
2. 关断 MCU

第一步，使用下面的代码示例配置唤醒 IO。

```
DL_GPIO_initDigitalInputFeatures(Wake_IO_PIN_0_IOMUX,  
    DL_GPIO_INVERSION_DISABLE, DL_GPIO_RESISTOR_PULL_UP,  
    DL_GPIO_HYSTERESIS_DISABLE, DL_GPIO_WAKEUP_ON_1);
```

第二步，您只需使用以下 API 即可

```
DL_SYSCTL_setPowerPolicySHUTDOWN();  
__WFI();
```

上述行会将器件置于关断模式。由于具有唤醒功能的 GPIO 配置为在高电平时唤醒，并且有一个上拉电阻连接到 GPIO，因此 MCU 在进入关断模式后立即唤醒。

锁存 IO 后，用户必须释放 IO 状态，才能正确更新 GPIO 功能。为此，可使用以下 API 来直接设置 SHDINIOREL 寄存器中的 RELEASE 位。

```
/* Release shutdown IO latches - call this after MMRs have been re-initialized */  
DL_SYSCTL_releaseShutdownIO();
```

3.2 将状态保持 IO 与引导加载程序结合使用

此功能对于 MCU 引导加载程序非常有用，在该引导加载程序中，IO 状态可能会在引导过程中或存储体交换期间丢失。对于某些 IO 用于偏置信号或在两个应用期间锁定到已知数字状态的用例，您可以使用前面显示的软件来配置 IO 状态。在开发使用引导加载程序的应用时，图 3-1 可用作一般概念，帮助说明如何设计程序以锁存 GPIO 状态。在此示例中，应用 A 检测到有新的固件映像可用并需要执行存储体交换。在进入关断模式之前，它会将其输出 GPIO 配置为所需的保持状态。从关断模式唤醒（会触发 BOR）时，引导加载程序会确定要执行哪个应用。每个应用都会首先重新配置其 GPIO 状态并在运行主应用循环之前通过 SHDINIOREL 释放关断 IO 锁存器。

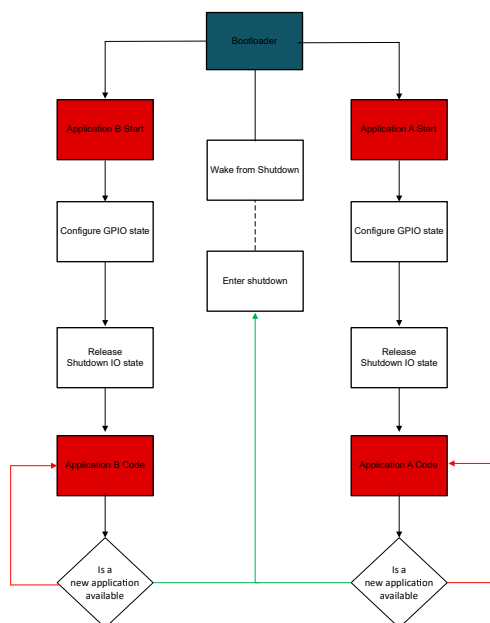


图 3-1. 使用 IO 锁存器功能的引导流程

确定 MCU 是否从关断状态唤醒

如果应用程序映像使用不同的功耗模式并且具有多种不同的方式，则 MCU 可以重新启动其应用程序映像。要确定 MCU 是否已从关断模式唤醒，并且需要设置 SHDINIOREL 寄存器中的 RELEASE 位，则可以读取 RSTCAUSE 寄存器。以下方代码为例。

```

if (DL_SYSCTL_getResetCause() == DL_SYSCTL_RESET_CAUSE_SHUTDOWN) {
    /* Device woke from shutdown - IO latches are active */
    /* Re-initialize peripherals, then release latches */
    DL_SYSCTL_releaseShutdownIO();
}
  
```

如果 MCU 使用在应用程序映像启动之前发出 SYSRST 的 CSC 或其他引导加载程序，则可以将 RSTCAUSE 寄存器与相应的软件触发的 SYSRST ID 值进行比较。有关更多详细信息，请参阅技术参考手册的 PMCU 章节。

3.3 关断存储器

此外，这些 MCU 允许用户在 SYSCTL 外设中存储多达四个字节的数据。您可以将该数据存储在 SHUTDNSTORE0 - SHUTDNSTORE3 寄存器中。只要在进入关断模式之前存储数据，SHUTDNSTORE 寄存器就会保持原来的状态，这种情况类似于 GPIO 状态的保持。

关断存储器软件

要存储此关断存储字节，请使用以下 API。

```
/* Before shutdown: store application state */  
DL_SYSCTL_setShutdownStorageByte(DL_SYSCTL_SHUTDOWN_STORAGE_BYTE_0, myStateValue);
```

从关断模式唤醒后，可使用以下 API 直接读取相应的寄存器

```
/* After wakeup: retrieve stored state */  
uint8_t myStateValue = DL_SYSCTL_getShutdownStorageByte(DL_SYSCTL_SHUTDOWN_STORAGE_BYTE_0);
```

关断存储器用例

对于使用存储体交换或引导加载程序的应用，其中 MMR 和 SRAM 在存储体交换之间丢失，用户可以直接在该存储器中写入它们。当设备在运行时计算出的数据原本只在前一个应用运行时计算出来时，这种方法很有用。这种情况的一个示例：当 I2C 总线上的器件进行通信时，计数器会递增。

4 总结

对于在 MCU 的整个生命周期内必须保持 GPIO 电压的应用，MSPM0 和 MSPM33 MCU 可以在 IO 单元内使用关断 IO 功能来正确锁存电压。通过将器件置于称为关断的低功耗模式，可以设置锁存器，并且 MCU 可以将 GPIO 预配置为上拉电阻器，以立即唤醒 MCU。如果在配置您的软件时考虑这一点，您可以使用前面提到的代码来构建将器件置于关断模式的软件，从而触发引导过程和锁存 IO 状态。利用该功能，可以轻松将软件项目配置为在执行固件更新时不会丢失 IO 电压。

5 参考资料

- 德州仪器 (TI)，[MSPM33 C3 系列 160MHz 微控制器](#) 技术参考手册。

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、与某特定用途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安全、安保法规或其他要求。

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。对于因您对这些资源的使用而对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，您将全额赔偿，TI 对此概不负责。

TI 提供的产品受 [TI 销售条款](#)、[TI 通用质量指南](#) 或 [ti.com](#) 上其他适用条款或 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。除非德州仪器 (TI) 明确将某产品指定为定制产品或客户特定产品，否则其产品均为按确定价格收入目录的标准通用器件。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

版权所有 © 2026，德州仪器 (TI) 公司

最后更新日期：2025 年 10 月