

Application Note

外部闪存上的 QSPI 使用



Jojo Arthur

摘要

MSPM33C321A 微控制器 (MCU) 集成了一个四路串行外设接口 (QSPI) 控制器，可通过四条数据线路连接到外部串行 NOR 闪存存储器，提供高达 20MB/s 的持续数据传输。本应用手册介绍了如何使用 LP-MSPM33C321A LaunchPad™ 开发套件及其板载 128Mbit MX25L12833F NOR 闪存作为参考平台，在 MSPM33C321A 上连接和配置外部 QSPI 闪存。将提供数据流用例的测量结果，在该用例中，MCU 从外部闪存读取批量数据并同时通过一个或两个 SPI 端口重新传输。本文档适用于嵌入式系统设计人员，他们需要使用外部存储器在光学通信模块、基于显示的用户界面和音频播放系统等终端设备中进行资产或数据存储。

内容

1 简介.....2

2 详细说明.....3

 2.1 QSPI 控制器概述.....3

 2.2 与外部 NOR 闪存的硬件连接.....4

 2.3 软件配置.....5

 2.4 测出的性能：具有 SPI 流的闪存读取.....8

 2.5 应用用例.....9

3 总结.....10

4 参考资料.....10

商标

所有商标均为其各自所有者的财产。

1 简介

许多嵌入式应用所需的非易失性存储空间超出了芯片内部集成的实际范围。用于光收发器的显示器、校准和配置记录的图形资产或用于语音提示系统的压缩音频剪辑可能各消耗数兆字节，而应用程序代码本身可以很好地置于片上闪存。外部串行 NOR 闪存器件以低成本、小型封装及简单的六信号接口提供这种大容量存储。

MSPM33C321A 是一款基于 160MHz Arm®Cortex®-M33 内核的混合信号 MCU，具有高达 1MB 的片上闪存和 256kB 的 SRAM，两者都具有 ECC。其通信外设中包括专用 QSPI 控制器，该控制器通过四条数据线连接到外部串行闪存，并支持高达 20MB/s 的数据传输。通过与两个 DMA 控制器（共 16 个通道）相结合，QSPI 使器件能够在很少 CPU 参与的情况下将大型数据块从外部闪存移动到片上 SRAM 或直接移动到其他接口。

本应用手册涵盖了三个主题：QSPI 控制器功能集及其与外部 NOR 闪存的硬件连接，从外部闪存读取的数据从一个或两个 SPI 端口流出的测量性能示例，以及在光学模块、显示应用和音频存储中使用此数据路径的实际设计考虑。

2 详细说明

2.1 QSPI 控制器概述

MSPM33C321A 上的 QSPI 控制器专为外部串行闪存而设计，提供以下主要特性：

- 单通道、双通道（双向）和四通道数据操作，串行时钟速率高达 40MHz（ $VDD \geq 2.7V$ ； $VDD \geq 1.71V$ 时高达 20MHz），在四路模式下数据传输速率高达 20MB/s
- 支持 QSPI 模式 0 及模式 3 时钟极性/相位的可编程接口
- 支持 3 字节和 4 字节的地址外部 QSPI 闪存
- 预配置的读取帧格式和自动虚拟时钟插入功能，以优化读取操作
- 可配置自动预取功能，用于在没有 CPU 干预的情况下进行连续数据读取
- 支持 DMA 的独立传输和接收 FIFO（最多 4 个位置深）
- 四条用于多个外部存储器器件的片选线路（QSPI_CS0 至 QSPI_CS3）

帧格式是 QSPI 的中央配置选择。QSPICTL0.QSPIFORMAT 字段使用三字母表示法来选择事务的指令、地址和数据阶段如何跨 IO 线路分布，其中每个字母将一个相位描述为单通道、双通道或四通道。配置和状态帧（命令写入、状态读取）使用 SPI 或 QPI 半双工格式，而数据读取帧支持表 1 中列出的六种格式。请注意，QQQ 模式与 QPI 模式相同，并且仅支持时钟组合 SPO=0/SPH=0（模式 0）和 SPO=1/SPH=1（模式 3）。

表 2-1. QSPI 数据读取帧格式 (QSPICTL0.QSPIFORMAT)

格式	代码	指令	地址	数据	典型用途
SSS	8h	单通道	单通道	单通道	正常/快速读取
固态硬盘	9h	单通道	单通道	双通道	快速读取双路输出
SDD	Ah	单通道	双通道	双通道	快速读取双路 IO
SSQ	Bh	单通道	单通道	四通道	快速读取四路输出
SQQ	Ch	单通道	四通道	四通道	快速读取四路 IO
QQQ (QPI)	Dh	四通道	四通道	四通道	完全 QPI 运行

对于每种数据读取格式，使用所选闪存读取命令所需的指令操作码 (QSPICMDBYTE)、地址宽度 (QSPIADDRMODE)、可选性能字节 (QSPIPERFBYTE) 和虚拟时钟计数 (QSPIDUMMYCLK) 对控制器进行编程；虚拟时钟提供 NOR 闪存时间来检索所请求的数据。串行时钟通过 CLKDIV 分频器和 CLKCTL 预分频器从 SYSCLK 产生，最高可达 40MHz 最大值，延迟数据采样选项 (CLKCTL.DSAMPLE) 或内部时钟环回 (QSPITIMING.DATSMPLDY) 可用于补偿高时钟速率下的电路板和闪存输出延迟。

QSPI 外设映射到基地址 0x4003.2000（非安全）/0x5003.2000（安全），使用中断线 47 (QSPI0)，并通过触发器 18 (QSPI RX) 和 19 (QSPI TX) 连接到 DMA 控制器。有关寄存器级别的详细信息，请参阅 MSPM33C3x 160MHz 微控制器技术参考手册中的“QSPI”一章。

2.2 与外部 NOR 闪存的硬件连接

LP-MSPM33C321A LaunchPad 开发套件包含一个连接到 QSPI 接口的 Macronix MX25L12833F 外部 NOR 闪存，用作本文档的参考硬件。MX25L12833F 是一款 128Mbit (16MB) 3V 串行 NOR 闪存，具有 256 字节的页面大小、可完全通过 3 字节 (24 位) 寻址进行寻址，因此 QSPIADDRMODE 保留其针对整个阵列的 3 字节设置。该器件在其状态寄存器中提供一个四路使能 (QE) 位，该位必须在任何四通道运行之前设置。表 2 列出了 NOR 闪存与 LaunchPad 引脚之间的信号映射。

表 2-2. 外部 NOR 闪存至 QSPI 引脚映射 (LP-MSPM33C321A)

NOR 闪存引脚	引脚功能	LaunchPad 引脚	QSPI 信号
1	CS	PC0	QSPI_CS0
2	IO1	PA14	QSPI_IO1
3	IO2	PA13	QSPI_IO2
4	GND	—	—
5	IO0	PA12	QSPI_IO0
6	SCLK	PB16	QSPI_CLK
7	IO3	PB15	QSPI_IO3
8	VCC	—	—

在使用 LaunchPad 时，值得注意两个板级详细信息。首先，外部 NOR 闪存通过跳线 J12 供电，必须填充该跳线才能为存储器提供 3.3V 电源轨。其次，要通过接头 J11 将 QSPI 信号路由到外部连接，必须焊接电路板背面的五个 0Ω 电阻器 (R77 至 R81)；默认情况下，这些电阻器未组装，以避免 QSPI 信号上产生噪声。

表 2-3 总结了 MX25L12833F 支持用于编程和读取操作的 QSPI 帧格式 (在 LaunchPad 上验证)。该器件支持所有四种读取格式，包括完整的 QQQ (QPI) 操作；对于编程，页面程序采用 SSS、SQQ 和 QQQ 格式，而 SSQ 页面程序不受支持。本文档中的测量和软件流程使用 QPI (QQQ) 模式进行编程及读取。

表 2-3. MX25L12833F 支持的 QSPI 帧格式

帧格式	页面程序	读取
SSS	✓	✓
SSQ	✗	✓
SQQ	✓	✓
QQQ (QPI)	✓	✓

对于定制电路板设计，请保持 QSPI 信号布线较短且长度匹配，并注意最大 QSPI 时钟频率取决于电源电压：40MHz 需要 $VDD \geq 2.7V$ ，而工作电压为 1.71V，将时钟限制为 20MHz。应在预期电压及温度范围内根据所选的闪存器件检查数据表 QSPI 时序参数 (输入建立/保持和输出有效/保持时间)。

2.3 软件配置

MSP M33 SDK 提供一个 QSPI 闪存驱动程序 (QSPI_FLASH.h)，该驱动程序分层于 DriverLib QSPI 模块 (DL_QSPI) 之上。SDK 示例 `qspi_flash_readback_dma` 演示了本应用手册中使用的整个生命周期：驱动程序初始化、闪存识别、四模式启用、擦除、基于 DMA 的页面编程以及基于 DMA 的四路读取（带验证）。

MSP M33 SDK 中提供了 `qspi_flash_readback_DMA` 示例，其形式为具有 GCC、IAR 和 TI Clang 编译器变体的 No-RTOS 工程。该工程将可重复使用的 QSPI 闪存驱动程序与应用程序分开：

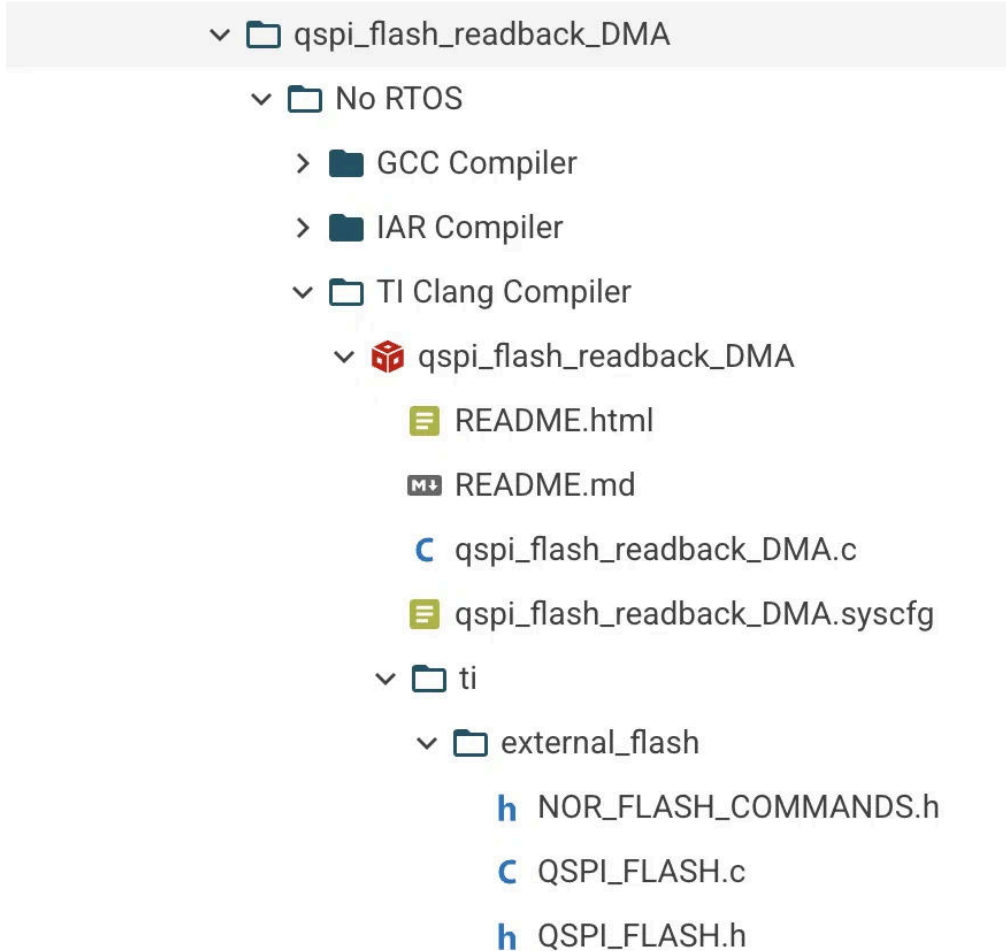


图 2-1. QSPI_flash_readback_DMA 工程结构

`NOR_FLASH_COMMANDS.h` 集中了特定于闪存的操作码（读取 ID、写入状态、启用 QPI、扇区擦除、页面编程、四通道 I/O 读取）和虚拟周期计数，因此将示例重新定向到不同的 NOR 闪存通常只需要根据新的闪存数据表编辑此标头。`QSPI_FLASH.c/h` 在 DriverLib (DL_QSPI) 上实现 QSPIFLASH 驱动程序，并且 `.syscfg` 文件通过 SysConfig 生成引脚多路复用、时钟树和 DMA 通道配置。

示例应用程序执行以下序列，检查每个步骤后返回的状态，并在第一次故障时锁存错误标志：

1. 初始化器件 (SYSCFG_DL_init)，启用 QSPI 中断并绑定 QSPIFLASH 驱动程序句柄
2. 在 SPI 半双工模式下读取 JEDEC ID 以确认闪存存在
3. 设置闪存状态寄存器中的 QE 位，然后启用 QPI 模式
4. 在 QPI 半双工模式下擦除目标扇区
5. 使用 QSPIFLASH_writeDMA（自动页边界处理）对测试数据进行页面编程
6. 使用四路 I/O 读取命令通过 QSPIFLASH_readDMA 读回数据
7. 禁用 QPI 模式并取消 QSPI 初始化

8. 验证 Launchpad LED 上的逐字回读缓冲器并报告通过/失败

以下各部分通过示例中的代码摘录演示了此流程的关键步骤。

2.3.1 驱动程序初始化和闪存识别

在 SYSCFG_DL_init() 配置 SysConfig 生成的引脚、时钟和 DMA 通道后，应用程序会在 NVIC 中启用 QSPI 中断，将驱动程序句柄绑定到 QSPI 实例，并读取 JEDEC ID 以确认闪存器件存在。启用四路模式之前必须运行的命令使用 SPI 半双工传输模式：

```
params.qspiRegs = QSPI ;
状态 = QSPIFLASH_init(&handle, &params) ;
commandConfig.instruction = READ_ID_CMD ;
commandConfig.transferMode = QSPIFLASH_TRANSFER_MODE_SPI_HALF_DUPLEX ;
状态 = QSPIFLASH_readID(&handle, id, sizeof(id), &commandConfig) ;
```

启动新电路板时，建议的第一个检查点是返回的 3 字节 ID (制造商 ID、存储器类型、容量)：正确的 ID 用于确认表 1 中的引脚映射、J12 电源跳线和单位 SPI 路径，然后再尝试任何四路模式配置。

2.3.2 启用四路模式并对闪存进行编程

四路运行需要在闪存侧执行两个步骤：使用 QSPIFLASH_setQEBit() 设置闪存状态寄存器中的四路使能 (QE) 位，然后使用 QSPIFLASH_enableQuadMode() 将器件切换到 QPI 模式。从该时间点起，擦除和编程命令使用 QPI 半双工传输模式。命令结构附带指令操作码、地址大小 (每个 QSPI 控制器的寻址支持为 3 字节或者 4 字节) 和目标地址：

```
commandConfig.instruction = SECTOR_ERASE_CMD ;
commandConfig.addressSize = QSPIFLASH_ADDRESS_SIZE_3BYTE ;
commandConfig.address = ADDRESS ;
commandConfig.transferMode = QSPIFLASH_TRANSFER_MODE_QPI_HALF_DUPLEX ;
状态 = QSPIFLASH_eraseSector(&handle, &commandConfig);
```

页面编程使用 QSPIFLASH_writeDMA()，它会自动跨闪存页边界拆分缓冲区。在该示例中，单次调用对 512 字节 (两个 256 字节页) 进行编程，其中驱动程序每页发出一条页编程命令，并为每次传输调用应用程序的 DMA 启动回调。

2.3.3 基于 DMA 的四路读取

在本文档中，读取路径是流用例的性能关键方向。该示例使用快速读取命令所需的虚拟周期计数配置四路 I/O 读取，选择 16 位 FIFO 字长以将 DMA 事务数量减半，并将传输交给 QSPIFLASH_readDMA()：

```
commandConfig.instruction = QUAD_IO_READ_CMD ;
commandConfig.addressSize = QSPIFLASH_ADDRESS_SIZE_3BYTE ;
commandConfig.address = ADDRESS ;
commandConfig.dummyCycles = DUMMY_CYCLES_QUAD_IO_READ ;
commandConfig.transferMode = QSPIFLASH_TRANSFER_MODE_QPI ;
dmaConfig.startDMA = startDMARxTransfer ;
dmaConfig.dmaCompleteFlag = &gDMARXDataTransferred ;
DL_QSPI_setWordSize(QSPI, DL_QSPI_WORD_SIZE_16) ;
状态 = QSPIFLASH_readDMA(&handle, readback_data, sizeof(readback_data), &commandConfig, &dmaConfig);
```

该驱动程序不拥有 DMA 通道；相反，该应用程序提供启动回调，因此相同的代码可用于任何通道分配。回调将通道指向 QSPI RX 数据寄存器，设置以 16 位字表示的传输大小，并启用通道：

```
void startDMARxTransfer(const void *destAddr, uint32_t sizeBytes)
{
    *(dmaConfig.dmaCompleteFlag) = false ;
    DL_QSPI_enableInterrupt(QSPI_0_INST, DL_QSPI_INTERRUPT_DMA_DONE_RX) ;
    DL_DMA_setSrcAddr(DMA0, DMA0_CH0_CHAN_ID, (uint32_t)(QSPI->RXDATA)) ;
    DL_DMA_setDestAddr(DMA0, DMA0_CH0_CHAN_ID, (uint32_t) destAddr) ;
    DL_DMA_setTransferSize(DMA0, DMA0_CH0_CHAN_ID, sizeBytes / sizeof(uint16_t)) ;
    DL_DMA_enableChannel(DMA0, DMA0_CH0_CHAN_ID) ;
}
```

完成时由 QSPI DMA_DONE_RX 中断发出信号 (QSPI0、中断线 47)。处理程序会设置驱动程序轮询的完成标志并在下一次传输前禁用中断，这种模式将中断服务例程保持为几条指令。写入方向在针对 QSPI TX 数据寄存器的第二个 DMA 通道上镜像了该结构。

该示例中的三个实践说明适用于在此驱动程序上构建的任何应用程序：

- 当传输长度为偶数且 DMA 移动 16 位字时，选择 DL_QSPI_WORD_SIZE_16；回退到 8 位字大小以进行奇数字节计数
- 在开始回调中设置 DMA_DONE 中断并在处理程序中禁用该中断，因此不能在传输之间触发过时中断
- 流式传输完成后，QSPIFLASH_disableQuadMode() 和 QSPIFLASH_deinit() 会将闪存返回到 SPI 模式，从而使器件可通过标准工具和引导加载程序恢复

该示例根据编程的模式逐字验证回读缓冲区，并在 LaunchPad LED 上报告通过/失败，从而在应用程序用真实资产数据替换测试模式之前提供一个自检起点。

2.4 测出的性能：具有 SPI 流的闪存读取

通用系统架构将 MCU 用作数据泵：批量数据存储在外部 QSPI 闪存中，并在运行时通过 SPI 读出并流式传输到一个或多个下游消费者。为了表征该路径，首先从外部 NOR 闪存读取 128KB 至 1MB 的数据块，然后使用一个 SPI 端口和两个同时传输的 SPI 端口通过 SPI 从器件发出。在接收端，仅捕获和丢弃发送的字节，没有后采样握手，因此结果代表发送路径的原始流能力。表 2-4 列出了测得的传输时间。

图 2-2 显示了该测量使用的数据路径。数据从外部 NOR 闪存通过 QSPI 控制器流向 DMA 控制下的 SRAM 缓冲器，然后通过一条或两条 SPI 发送路径流出，每条路径由其自己的 DMA 通道提供服务。Cortex-M33 CPU 仅安排传输，不会复制数据。

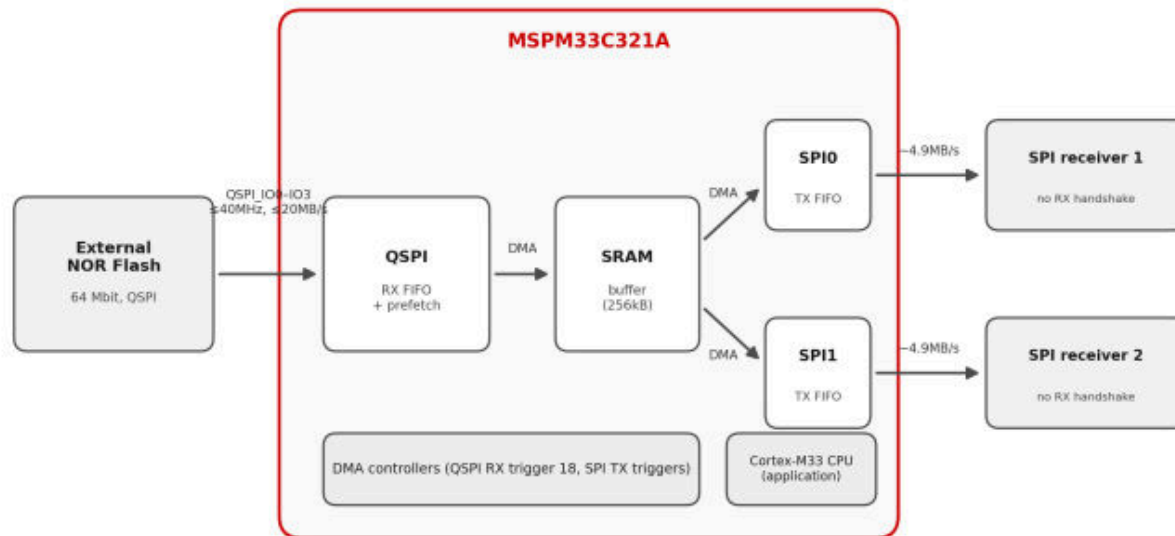


图 2-2. 外部闪存至双 SPI 流路径

表 2-4. 测量的传输时间、外部闪存读取及 SPI 发送

数据大小	1 个 SPI 传输	2 个 SPI 传输
128KB	26.81ms	26.82ms
256KB	52.80ms	52.83ms
512KB	104.79ms	104.84ms
1MB	208.76ms	208.90ms

从这些结果可以观察到两个结果。首先，传输时间随数据大小呈线性变化（将有效载荷加倍会使时间在百分之几以内），这表明完全流水线化，由 DMA 驱动的数据路径，每个块的软件开销可以忽略不计。每个 SPI 端口的有效持续吞吐量约为 4.9MB/s，或线路上约为 39Mbit/s。

其次，添加第二个 SPI 发送器基本上是自由的：1MB 传输在一个 SPI 端口上需要 208.76ms，在两个端口上需要 208.90ms 来流式传输相同数量的数据，差异小于 0.1%。由于 QSPI 读取侧（高达 20MB/s）比 SPI 发送侧更快，并且每个 SPI 端口都由自己的 DMA 通道提供服务，因此两个输出流并行运行，而不会争用 CPU 时间或总线带宽。这使聚合输出带宽加倍至大约 9.8MB/s。

请注意，这些图是在采样后没有任何接收端握手的情况下测得的。在接收器应用流控制的实际系统中（例如，就绪线路或命令/响应协议），有效吞吐量由接收器设置并将会较低；表 2 中的值表示 MSPM33C321A 发送路径本身的上限。

2.5 应用用例

2.5.1 光学模块

光学收发器和通信模块通常存储大型校准表、每通道调优系数以及超出片上保留容量的固件图像。外部 QSPI 闪存保存此数据，上面测量的流式架构直接映射到模块的内部拓扑：MSPM33C321A 从闪存读取系数块，并通过独立的 SPI 端口同时将其分布到多个 DAC、激光驱动器 IC 或交叉点器件。第二个 SPI 流的成本接近零意味着并行光学通道的配置不会进行序列化，从而减少了模块启动和重新配置时间。该器件的 125°C 工作温度等级及其适合通信模块应用的列表使其非常适合此环境。

2.5.2 显示应用

图形用户界面需要图像资产、字体和动画帧，这些资源很快就会超过片上闪存的增长。将这些资产存储在外部 QSPI 闪存中并将其流式传输到显示控制器是标准模式，由 LaunchPad 开箱即用示例展示，该示例在板载 128×32 OLED 上播放 64Mbit NOR 闪存中的视频。作为尺寸调整基准，128KB 帧或资产块在 27ms 内穿过闪存到 SPI 路径，而大约 4.9MB/s 的持续流式传输支持在中小型 SPI 显示器上提供连续动画。当前一帧仍在进行时钟输出时，可以使用 QSPI 自动预取功能将更大的资产预提取到 SRAM 中。

2.5.3 音频存储及播放

语音提示、提示音和音频片段自然地存储在外部闪存中。MSPM33C321A 将 QSPI 与两个支持全双工 I2S 和 16 时隙 TDM 的数字音频接口配对使用，因此可以从闪存读取压缩或 PCM 音频，并将其传送到传输两侧均具有 DMA 的外部编解码器或放大器。测得的流带宽约为 4.9MB/s，比多通道音频所需的速率高几个数量级（48kHz 时 16 位立体声仅需要 192kB/s），为 CPU 运行应用程序或音频路径通过单独的 QSPI 芯片选择与共享同一外部闪存的显示路径共存留出了足够的余量。

3 总结

MSPM33C321A 上的 QSPI 控制器提供到外部 NOR 闪存的简单高带宽连接，具有高达 20MB/s 的四模式传输、自动预取和完全 DMA 集成。LP-MSPM33C321A LaunchPad 上的测量结果表明，从外部闪存读取的数据可以每个端口以大约 4.9MB/s 的速率通过 SPI 流式传输出去，并且第二个并发 SPI 流基本上不会造成时间损失，因此聚合输出带宽加倍。该闪存至接口数据泵架构可满足各种终端设备的需求，包括光学模块中的系数分布、显示器资产流和音频削波，同时让 Cortex-M33 CPU 可以自由地进行应用处理。

4 参考资料

1. 德州仪器 (TI)，[MSPM33C321x 混合信号微控制器](#) 数据表。
2. 德州仪器 (TI)，[LP-MSPM33C321A 评估模块](#) 用户指南。
3. 德州仪器 (TI)，[MSPM33 C3 系列 160MHz 微控制器](#)，技术参考手册。
4. TI Resource Explorer 中提供了德州仪器 (TI) 的 MSP 软件开发套件 (SDK)、qspi_flash_readback_dma 示例

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