

Application Note

外部フラッシュでの QSPI 使用



Jojo Arthur

概要

MSPM33C321A マイコン (MCU) は、クワッド シリアル ペリフェラル インターフェイス (QSPI) コントローラを内蔵しており、4 つのデータライン経由で外部シリアル NOR フラッシュ メモリに接続し、最大 20MB/秒の持続的なデータ転送を実現します。このアプリケーション ノートでは、LP-MSPM33C321A LaunchPad™ 開発キットとオンボードの 128Mbit MX25L12833F NOR フラッシュをリファレンス プラットフォームとして使用し、MSPM33C321A 上で外部 QSPI フラッシュの接続および構成を行う方法について説明します。測定結果は、データ ストリーミングの事例で示しています。この事例では、マイコンが外部フラッシュからバルク データを読み取り、1 つまたは 2 つの SPI ポートを介して同時に転送します。この文書は、光通信モジュール、ディスプレイベースのユーザー インターフェイス、オーディオ再生システムなどの最終製品において、アセットまたはデータ ストレージ用の外部メモリを必要とする組込みシステム設計者を対象としています。

目次

1 はじめに.....	2
2 詳細説明.....	3
2.1 QSPI コントローラの概要.....	3
2.2 外部 NOR フラッシュへのハードウェア接続.....	4
2.3 ソフトウェアの設定.....	5
2.4 性能測定値:SPI ストリーミングによるフラッシュ読み取り.....	8
2.5 アプリケーション使用事例.....	10
3 まとめ.....	11
4 参考資料.....	11

商標

すべての商標は、それぞれの所有者に帰属します。

1 はじめに

多くの組み込みアプリケーションでは、オンチップへの保存が実用的でない量の不揮発性ストレージが必要とされています。ディスプレイ用のグラフィカル アセット、光学トランシーバのキャリブレーションおよび設定用のレコード、音声プロンプト システム用の圧縮オーディオ クリップは、それぞれ数 MB を消費する可能性があります。アプリケーションコード自体はオンチップ フラッシュに十分収まります。外部シリアル NOR フラッシュ デバイスは小型パッケージで、このバルク ストレージを低コストで提供し、シンプルな 6 信号インターフェイスを備えています。

MSPM33C321A は、160MHz Arm®Cortex®-M33 コアをベースとするミックスドシグナル マイコンで、最大 1MB のオンチップ フラッシュと 256KB の SRAM (いずれも ECC 付き) を搭載しています。通信ペリフェラルの中には専用の QSPI コントローラが含まれており、4 つのデータライン経由で外部シリアル フラッシュ メモリと接続し、最大 20MB/秒のデータ転送をサポートします。QSPI を 2 つの DMA コントローラ (合計 16 チャネル) と組み合わせることで、CPU の関与を最小限に抑えながら、大きなデータ ブロックを外部フラッシュからオンチップ SRAM に、または直接他のインターフェイスに移動できます。

このアプリケーション ノートでは、3 つのトピックを取り上げています。1 つ目は、QSPI コントローラの機能セットとそのハードウェアから外部 NOR フラッシュへの接続、2 つ目は 1 つまたは 2 つの SPI ポートから外部フラッシュからのデータ読み取りをストリーミングする測定性能例、最後は、光学モジュール、ディスプレイ アプリケーション、オーディオ ストレージでこのデータパスを使用する際の設計上の考慮事項です。

2 詳細説明

2.1 QSPI コントローラの概要

MSPM33C321A の QSPI コントローラは、外部シリアル フラッシュ メモリ用に特別に設計されており、以下の主な機能があります。

- 最高 40MHz のシリアル クロック速度でシングル、デュアル (bi)、クワッド データ動作 ($VDD \geq 2.7V$ 、 $VDD \geq 1.71V$ で最大 20MHz)、クワッド モードで 20MB/秒までのデータ転送に対応
- QSPI モード-0 とモード-3 クロック極性 / 位相をサポートするプログラマブル インターフェイス
- 3 バイト アドレスと 4 バイト アドレスの両方の外部 QSPI フラッシュ メモリをサポート
- 最適化された読み出し動作のための事前設定された読み出しフレーム フォーマットとダミー クロックの自動挿入
- CPU の介在なしで連続的なデータ読み取りを実現する構成可能な自動プリフェッチ機能
- 独立した送受信 FIFO (最大 4 領域の深さ)、DMA をサポート
- 複数の外部メモリ デバイス用の 4 本のチップセレクトライン (QSPI_CS0 ~ QSPI_CS3)

フレーム フォーマットは、QSPI の中心となる構成項目です。QSPICTL0.QSPIFORMAT フィールドは、トランザクションの命令、アドレス、データ フェーズが IO ラインにどのように分配されるかを選択します。各文字がそれぞれのフェーズ (シングル、デュアル、クワッド レーン) を表す 3 文字の表記を使用します。構成フレームおよびステータス フレーム (コマンド書き込み、ステータス読み取り) は SPI または QPI 半二重形式を使用し、データ読み取りフレームは表 1 に示す 6 つの形式をサポートします。QQQ モードは QPI モードと同じであり、クロックの組み合わせ SPO=0/SPH=0 (Mode-0) および SPO=1/SPH=1 (Mode-3) のみがサポートされていることに注意してください。

表 2-1. QSPI データ読み取りフレーム フォーマット (QSPICTL0.QSPIFORMAT)

フォーマット	コード	手順	アドレス	データ	代表的な使用事例
SSS	8h	シングル	シングル	シングル	通常 / 高速読み取り
SSD	9h	シングル	シングル	デュアル	高速読み取りデュアル出力
SDD	Ah	シングル	デュアル	デュアル	高速読み取りデュアル IO
SSQ	Bh	シングル	シングル	クワッド	高速読み取りクワッド出力
SQQ	Ch	シングル	クワッド	クワッド	高速読み取りクワッド IO
QQQ (QPI)	Dh	クワッド	クワッド	クワッド	完全な QPI 動作

データ読み取りフォーマットごとに、コントローラは、選択されたフラッシュ読み取りコマンドで必要な命令オペコード (QSPICMDBYTE)、アドレス幅 (QSPIADDRMODE)、オプションのパフォーマンス バイト (QSPIPERFBYTE)、およびダミー クロック数 (QSPIDUMMYCLK) をプログラムします。ダミー クロックは、要求されたデータを取得するための NOR フラッシュ時間を確保します。シリアル クロックは、SYSCLK から CLKDIV 分周器および CLKCTL プリスケアラを介して 40MHz (最大値) まで生成されます。遅延データ サンプリング オプション (CLKCTL.DSAMPLE) または内部クロック ループバック (QSPITIMING.DATSMPLDY) を使用すると、高クロック レートにおける基板およびフラッシュの出力遅延を補償できます。

QSPI ペリフェラルはベース アドレス 0x4003.2000 (非セキュア) /0x5003.2000 (セキュア) にマップされ、割り込みライン 47 (QSPI0) を使用して、トリガ 18 (QSPI RX) および 19 (QSPI TX) を介して DMA コントローラに接続されます。レジスタレベルの詳細については、『MSPM33C3x 160MHz マイコン テクニカル リファレンス マニュアル』の「QSPI」の章を参照してください。

2.2 外部 NOR フラッシュへのハードウェア接続

LP-MSPM33C321A LaunchPad 開発キットは、QSPI インターフェイスに接続された Macronix MX25L12833F 外部 NOR フラッシュを搭載しており、この資料に掲載されているリファレンス ハードウェアとして使用されています。

MX25L12833F は、256 バイトのページサイズを持つ 128Mbit (16MB) 3V シリアル NOR フラッシュで、3 バイト (24 ビット) アドレッシングで完全にアドレス指定可能なため、QSPIADDRMODE はフルアレイに対して 3 バイト設定のままです。このデバイスは、ステータスレジスタにクワッド イネーブル (QE) ビットを用意しています。このビットは、クワッド レーン動作の前に設定する必要があります。表 2 に、NOR フラッシュと LaunchPad ピンの間の信号マッピングを示します。

表 2-2. 外部 NOR フラッシュから QSPI ピンへのマッピング (LP-MSPM33C321A)

NOR フラッシュ ピン	ピン機能	LaunchPad ピン	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	—	—

LaunchPad を使用するときは、基板レベルの 2 つの詳細に注目してください。まず、外部 NOR フラッシュにジャンパ J12 から給電します。このジャンパ J12 を実装して、メモリに 3.3V レールを供給する必要があります。第 2 に、QSPI 信号をヘッダ J11 を経由して外部接続に配線するには、基板の裏面にある 5 個の 0Ω 抵抗 (R77 ~ R81) を半田付けする必要があります。QSPI 信号のノイズを避けるため、デフォルトではこれらは未実装のままです。

表 2-3 に、LaunchPad で検証済みのプログラムおよび読み取り動作で MX25L12833F がサポートしている QSPI フレームフォーマットを示します。このデバイスは、完全な QQQ (QPI) 動作を含む 4 つの読み取り形式すべてをサポートしています。プログラミング用には、ページプログラムを SSS、SQQ、および QQQ 形式で使用できますが、SSQ ページプログラムはサポートされていません。このドキュメントの測定とソフトウェア フローでは、プログラムと読み取りの両方に QPI (QQQ) モードを使用します。

表 2-3. MX25L12833F がサポートしている QSPI フレーム フォーマット

フレームフォーマット	ページプログラム	読み出し
SSS	✓	✓
SSQ	✗	✓
SQQ	✓	✓
QQQ (QPI)	✓	✓

カスタム基板設計では、QSPI 信号トレースを短くし、長さを一致させます。また、最大 QSPI クロック周波数は電源電圧によって異なります。40MHz は VDD ≥2.7V が必要ですが、1.71V で動作すると、クロックは 20MHz に制限されます。データシートの QSPI タイミング パラメータ (入力セットアップ / ホールド時間と出力有効 / ホールド時間) は、想定される電圧および温度の全範囲にわたり、選択したフラッシュ デバイスと比較してチェックする必要があります。

2.3 ソフトウェアの設定

MSP M33 SDK は、DriverLib QSPI モジュール (DL_QSPI) の上に実装された QSPI フラッシュドライバ (QSPI_FLASH.h) を提供します。SDK のサンプル `qspi_flash_readback_dma` は、ドライバの初期化、フラッシュ識別、クワッドモードの有効化、消去、DMA ベースのページプログラム、DMA ベースの検証付きクワッド読み取りなど、このアプリケーション ノートで説明する一連の流れに対応しています。

`qspi_flash_readback_DMA` サンプルは、MSP M33 SDK の中で、GCC、IAR、TI Clang コンパイラ バリエーションを使用した No-RTOS プロジェクトとして提供されています。このプロジェクトでは、再利用可能な QSPI フラッシュドライバがアプリケーションから分離されます。

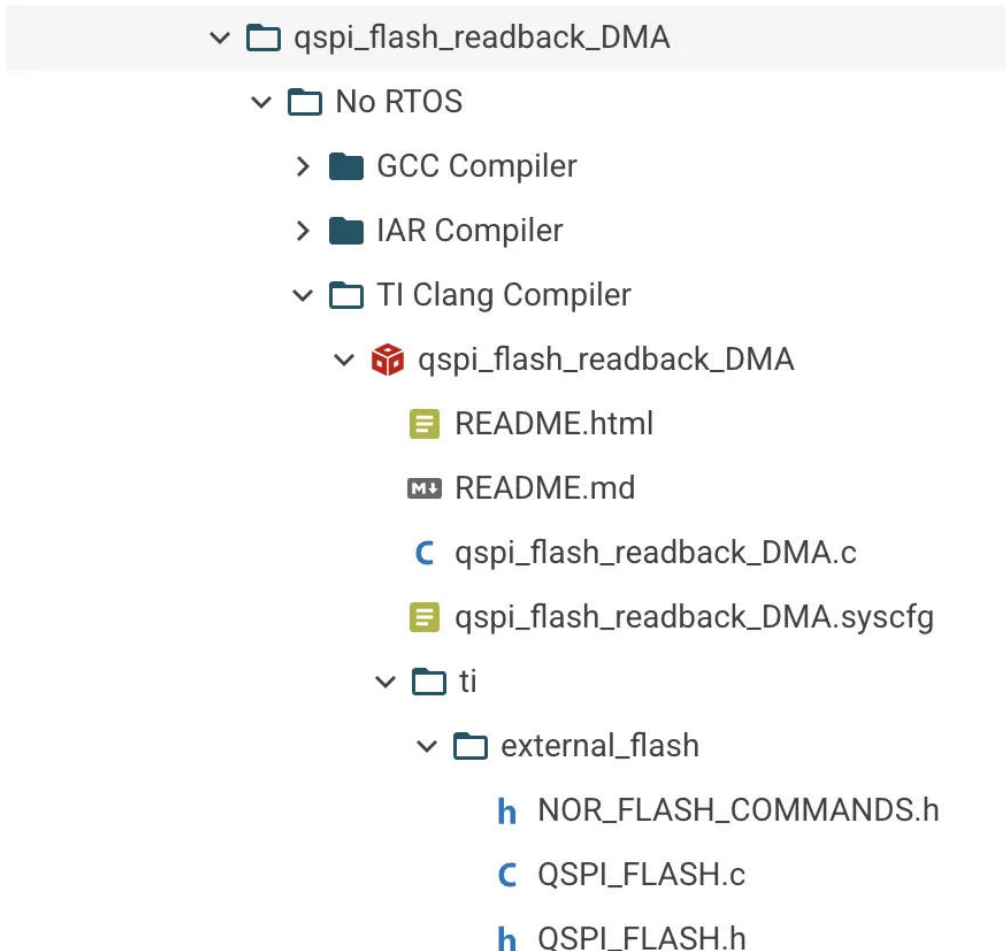


図 2-1. QSPI_flash_readback_DMA プロジェクト構造

`NOR_FLASH_COMMANDS.h` でフラッシュ固有のオペコード (読み取り ID、書き込みステータス、QPI の有効化、セクタ消去、ページプログラム、クワッド I/O 読み取り) とダミーのサイクル数を集約しているため、このサンプルを別の NOR フラッシュにリターゲットする場合、通常は、このヘッダを新しいフラッシュ データシートに合わせて編集するだけで済みます。`QSPI_FLASH.c/h` は、DriverLib (DL_QSPI) の上に QSPIFLASH ドライバを実装しており、`.syscfg` ファイルは SysConfig を使用してピン マルチプレクシング、クロック ツリー、DMA チャネル構成を生成します。

サンプル アプリケーションは次のシーケンスを実行し、各ステップの後に返されるステータスをチェックして、最初の障害時にエラー フラグをラッチします。

1. デバイス (SYSCFG_DL_init) を初期化し、QSPI 割り込みをイネーブルにして、QSPIFLASH ドライバ ハンドルをバインドします
2. SPI 半二重モードで JEDEC ID を読み出し、フラッシュが存在することを確認します
3. フラッシュ ステータス レジスタの QE ビットをセットして、QPI モードをイネーブルにします

4. QPI 半二重モードでターゲット セクタを消去します
5. QSPIFLASH_writeDMA を使用してテスト データを書き込みます (ページ境界の自動処理)
6. クワッド I/O 読み取りコマンドを使用して、QSPIFLASH_readDMA でデータを読み戻します
7. QPI モードをディスエーブルにし、QSPI の初期化を解除します
8. 読み戻しバッファをワード単位で検証し、Launchpad LED に検証結果を報告します

以下のサブセクションでは、このフローの主な手順について、サンプルから抜粋したコードを使用して説明します。

2.3.1 ドライバの初期化とフラッシュ識別

SYSCFG_DL_init() が、SysConfig によって生成されたピン、クロック、DMA チャンネルを構成した後、アプリケーションは NVIC の QSPI 割り込みをイネーブルにし、ドライバ ハンドルを QSPI インスタンスにバインドして JEDEC ID を読み取り、フラッシュ デバイスが存在することを確認します。クワッドモードをイネーブルにする前に実行する必要があるコマンドは、SPI 半二重転送モードを使用します。

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

返される 3 バイトの ID (メーカー ID、メモリタイプ、容量) は、新しいボードの起動時に最初に推奨されるチェックポイントです。正しい ID により、クワッドモード構成が試行される前に、表 1 のピン マッピング、J12 電源ジャンパ、およびシング ルビット SPI パスを検証することができます。

2.3.2 クワッド モードの有効化とフラッシュのプログラミング

クワッド動作では、フラッシュ側で 2 つのステップを行う必要があります。QSPIFLASH_setQEBit() でフラッシュ ステータス レジスタのクワッド イネーブル (QE) ビットをセットし、QSPIFLASH_enableQuadMode() でデバイスを QPI モードに切り替えます。この時点から、消去およびプログラム コマンドは QPI 半二重転送モードを使用します。このコマンド構造により、命令オペコード、アドレス サイズ (QSPI コントローラのアドレッシングのサポートにより 3 バイトまたは 4 バイト)、ターゲット アドレスが転送されます。

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

ページ プログラミングでは QSPIFLASH_writeDMA() を使用します。この QSPIFLASH_writeDMA() は、フラッシュ ページ境界を超えてバッファを自動的に分割します。この例では、1 回の呼び出しで 512 バイト (256 バイトのページを 2 つ) をプログラムし、ドライバがページごとに 1 つの page-program コマンドを発行し、転送ごとにアプリケーションの DMA start コールバックを呼び出します。

2.3.3 DMA ベースのクワッド読み取り

このドキュメントのストリーミングの使用例では、読み取りパスがパフォーマンスにとって非常に重要な要因となります。この例では、高速読み出しコマンドに必要なダミー サイクル数を使用してクワッド I/O 読み取りを構成し、16 ビットの FIFO ワード サイズを選択して DMA トランザクション数を半分にし、その転送を次のように QSPIFLASH_readDMA () で処理します。

```
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);
status = QSPIFLASH_readDMA(&handle, readback_data, sizeof(readback_data), &commandConfig,
&dmaConfig);
```

ドライバは DMA チャンネルを所有していません。代わりに、アプリケーションが START コールバックを提供するため、同じコードがどのチャンネル割り当てでも動作します。コールバックはチャンネルを QSPI RX データレジスタにセットし、16 ビットワード単位で転送サイズを設定して、チャンネルをイネーブルにします。

```
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);
}
```

完了は、QSPI DMA_DONE_RX 割り込み (QSPI0、割り込みライン 47) によって通知されます。ハンドラは、ドライバによってポーリングされる完了フラグを設定し、次の転送まで割り込みを無効にします。このパターンでは、割り込みサービスルーチンをいくつかの命令に抑えます。書き込み方向に従って、この構造は、QSPI TX データレジスタをターゲットとした 2 番目の DMA チャンネルにミラーリングされます。

このドライバを使用して構築されたアプリケーションには、この例の 3 つの実践的な注意事項が適用されます。

- 転送の長さが偶数の場合、DL_QSPI_WORD_SIZE_16 を選択します。これにより、DMA が 16 ビットワード移動します。奇数バイト数の場合、8 ビットのワードサイズに戻ります
- 開始コールバック内の DMA_DONE 割り込みを再度有効にし、ハンドラ内で無効にします。これにより、古い割り込みが転送間に発生しないようにします
- ストリーミングが完了すると、QSPIFLASH_disableQuadMode () と QSPIFLASH_deinit () はフラッシュを SPI モードに戻します。その結果、デバイスが標準ツールおよびブートローダで回復可能な状態に維持されます

この例では、プログラミングされたパターンに対する読み戻しバッファをワード単位で検証し、LaunchPad LED に成功または失敗を報告します。これにより、アプリケーションがテストパターンを実際の資産データに置き換える前に、自己チェックを始めることができます。

2.4 性能測定値: SPI ストリーミングによるフラッシュ読み取り

一般的なシステムアーキテクチャは、MCU をデータ ポンプとして使用します。バルク データは外部 QSPI フラッシュに保存され、実行時に読み出し、SPI 経由で 1 つまたは複数のダウンストリームのコンシューマにストリーミングされます。このパスの特性を評価するため、128KB から 1MB のデータブロックを外部 NOR フラッシュから読み出した後、SPI 経由でデバイスから送信しました (1 つの SPI ポートを使用した送信と 2 つの SPI ポートを使用した同時送信)。受信側では、送信されたバイトは、サンプリング後のハンドシェイクなしで単純にキャプチャおよび廃棄されたため、結果は送信パスの実際のストリーミング能力を表します。表 2-4 に、測定された転送時間を示します。

図 2-2 に、この測定で実行されたデータ パスを示します。データは、外部 NOR フラッシュから QSPI コントローラ経由で DMA 制御下の SRAM バッファに流れ、その後 1 つまたは 2 つの SPI 送信パスを経由して出力され、それぞれが独自の DMA チャンネルで処理されます。Cortex-M33 CPU は転送のみをオーケストレーションします。データをコピーすることはありません。

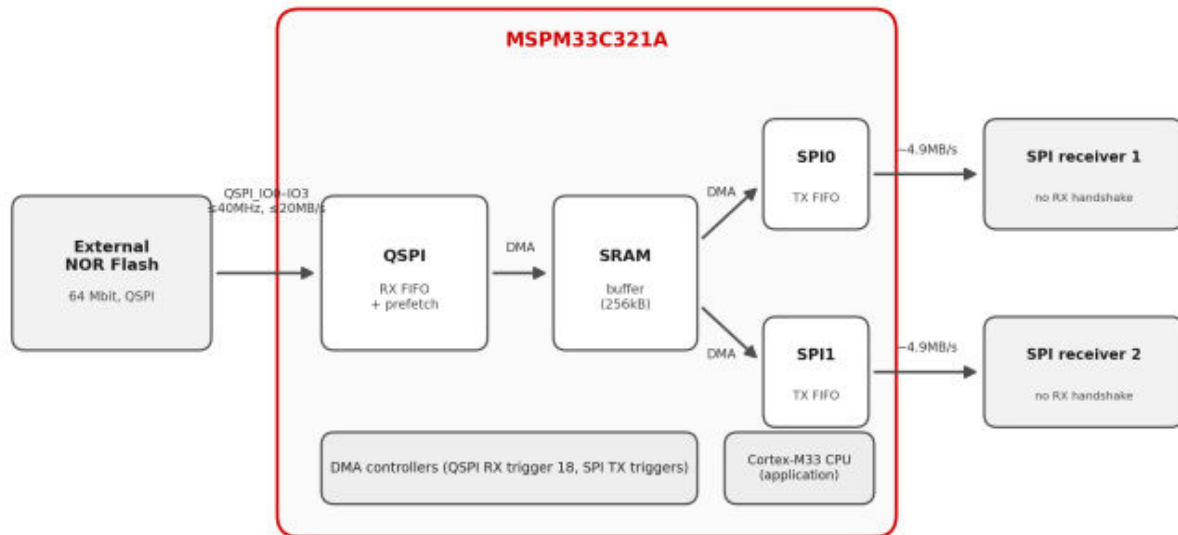


図 2-2. 外部フラッシュからのデュアル SPI ストリーミングパス

表 2-4. 測定された転送時間、外部フラッシュ読み取り + SPI 送信

データ サイズ	1 つの SPI 送信	2 つの SPI 送信
128KB	26.81ms	26.82ms
256KB	52.80ms	52.83ms
512KB	104.79ms	104.84ms
1MB	208.76ms	208.90ms

これらの結果から 2 つの観測結果が得られました。第 1 に、転送時間はデータサイズに比例してスケーリングされます (ペイロードを 2 倍にすると、時間は 1% 未満の誤差で 2 倍になります)。これは、ブロックあたりのソフトウェア オーバーヘッドが無視できるほど、完全にパイプライン化された DMA 駆動のデータ パスであることを示しています。実効的な持続スループットは、SPI ポートあたり約 4.9MB/秒、ワイヤ上で約 39Mbit/秒です。

2 番目の SPI トランスミッタの追加に実質的なコストはありません。1MB の転送を 1 つの SPI ポートで行う場合は 208.76ms、2 つのポートでそれぞれ同じ量のデータをストリーミングする場合は 208.90ms で、その差は 0.1% 未満です。QSPI 読み取り側 (最大 20MB/s) は SPI 送信側よりも高速であり、各 SPI ポートは独自の DMA チャンネルで処理されるため、CPU 時間やバス帯域幅に競合することなく、2 つの出力ストリームが並列に動作します。これにより、総出力帯域幅が倍増し、約 9.8MB/秒となります。

これらの値は、サンプリング後に受信側のハンドシェイクなしで測定されたものである点に注意してください。レシーバがフロー制御を適用する実際のシステム (準備完了ラインやコマンド / 応答プロトコルなど) では、実効スループットはレシーバによって設定され、低くなります。表 2 の値は MSPM33C321A 送信パス自体の上限を表しています。

2.5 アプリケーション使用事例

2.5.1 光モジュール

光トランシーバおよび通信モジュールには通常、大きなキャリブレーション テーブル、チャネルごとのチューニング係数、ファームウェア イメージが保存され、オンチップで維持するための経済的な容量を超えることがあります。このデータが外部の QSPI フラッシュに保持され、上記で測定されたストリーミング アーキテクチャがモジュールの内部トポロジに直接マップされます。MSPM33C321A は、係数ブロックをフラッシュから読み取り、独立した SPI ポート経由で複数の DAC、レーザードライバ IC、またはクロス ポイント デバイスに同時に分配します。2 番目の SPI ストリームのコストはほぼゼロであるため、光並列レーンの構成がシリアル化されません。このため、モジュールの起動および再構成時間が短縮されます。デバイスの 125°C 拡張定格温度と、通信モジュール用途でのリストは、この環境に最適です。

2.5.2 ディスプレイ アプリケーション

グラフィカル ユーザー インターフェイスでは、オンチップ フラッシュの容量を超える画像アセット、フォント、アニメーションフレームが必要になります。これらのアセットは、外部 QSPI フラッシュに保存し、ディスプレイ コントローラにストリーミングするのが標準的なパターンです。この LaunchPad に付属のサンプルでは、オンボードの 128×32 OLED 上で 64Mbit NOR フラッシュからビデオを再生します。サイズの目安として、128KB のフレームまたはアセット ブロックがフラッシュから SPI へのパスを 27ms 未満で移動し、約 4.9MB/秒の持続的ストリーミングでは、小型および中型の SPI ディスプレイでの連続アニメーションをサポートします。前のフレームがクロックアウトされている間は、QSPI 自動プリフェッチ機能を使用して、より大きなアセットを SRAM にプリフェッチできます。

2.5.3 オーディオの保存と再生

音声プロンプト、警告トーン、オーディオ クリップは、性質上、外部フラッシュへの保存に適しています。MSPM33C321A は、QSPI と、全二重 I2S と 16 スロット TDM をサポートする 2 つのデジタル オーディオ インターフェイスを組み合わせるため、圧縮または PCM オーディオをフラッシュから読み取り、転送の両側で DMA を使用し、外部コーデックまたはアンプに転送できます。約 4.9 MB/秒の測定されたストリーミング帯域幅は、マルチ チャネル オーディオであっても必要なレートを大幅に上回っているため (48kHz の 16 ビット ステレオに必要なのはわずか 192kB/s)、CPU がアプリケーションを実行するために十分なヘッドルームが残されているほか、別の QSPI チップ セレクトにより、同じ外部フラッシュを共有するディスプレイ パスとオーディオ パスを共存させることができます。

3 まとめ

MSPM33C321A の QSPI コントローラは、外部 NOR フラッシュとシンプルで高帯域幅の接続を確立し、最大 20MB/秒のクアドモード転送、自動プリフェッチ、完全な DMA 統合を実現します。LP-MSPM33C321A LaunchPad で測定された結果を見ると、外部フラッシュからのデータ読み出しでは SPI 経由でポートあたり 4.9MB/秒でストリーミング出力できることを示しています。また、2 番目の同時 SPI ストリームを使用すると、実質的に時間ペナルティがなく、集約した出力帯域幅が 2 倍になることを示しています。このフラッシュ / インターフェイス間データポンプ アーキテクチャにより、Cortex-M33 CPU がアプリケーション処理から解放され、光学モジュールの係数分配、ディスプレイ用のアセットストリーミング、オーディオ クリップの再生など、幅広い最終製品で活用できるようになります。

4 参考資料

1. テキサス インスツルメンツ、『[MSPM33C321x ミックスド シグナル マイコン](#)』、データシート。
2. テキサス インスツルメンツ、『[LP-MSPM33C321A 評価基板](#)』ユーザー ガイド。
3. テキサス インスツルメンツ、『[MSPM33 C3 シリーズ 160MHz マイコン](#)』、テクニカル リファレンス マニュアル。
4. テキサス・インスツルメンツ、MSP ソフトウェア開発キット (SDK)、qspi_flash_readback_dma サンプル。TI Resource Explorer で入手可能

重要なお知らせと免責事項

TI は、技術データと信頼性データ (データシートを含みます)、設計リソース (リファレンス デザインを含みます)、アプリケーションや設計に関する各種アドバイス、Web ツール、安全性情報、その他のリソースを、欠陥が存在する可能性のある「現状のまま」提供しており、商品性および特定目的に対する適合性の黙示保証、第三者の知的財産権の非侵害保証を含みいかなる保証も、明示的または黙示的にかかわらず拒否します。

これらのリソースは、TI 製品を使用する設計の経験を積んだ開発者への提供を意図したものです。(1) お客様のアプリケーションに適した TI 製品の選定、(2) お客様のアプリケーションの設計、検証、試験、(3) お客様のアプリケーションに該当する各種規格や、その他のあらゆる安全性、セキュリティ、規制、または他の要件への確実な適合に関する責任を、お客様のみが単独で負うものとし、TI は一切の責任を拒否します。

上記の各種リソースは、予告なく変更される可能性があります。これらのリソースは、リソースで説明されている TI 製品を使用するアプリケーションの開発の目的でのみ、TI はその使用をお客様に許諾します。これらのリソースに関して、他の目的で複製することや掲載することは禁止されています。TI や第三者の知的財産権のライセンスが付与されている訳ではありません。お客様は、これらのリソースを自身で使用した結果発生するあらゆる申し立て、損害、費用、損失、責任について、TI およびその代理人を完全に補償するものとし、TI は一切の責任を拒否します。

TI の製品は、[TI の販売条件](#)、[TI の総合的な品質ガイドライン](#)、[ti.com](https://www.ti.com) または TI 製品などに関連して提供される他の適用条件に従い提供されます。TI がこれらのリソースを提供することは、適用される TI の保証または他の保証の放棄の拡大や変更を意味するものではありません。TI がカスタム、またはカスタマー仕様として明示的に指定していない限り、TI の製品は標準的なカタログに掲載される汎用機器です。

お客様がいかなる追加条項または代替条項を提案する場合も、TI はそれらに異議を唱え、拒否します。

Copyright © 2026, Texas Instruments Incorporated

最終更新日：2025 年 10 月