

ADUCM3027／ADUCM3029 の SPI バスを介した SD カードとのインターフェース

はじめに

セキュア・デジタル (SD) カードは、携帯機器や組み込みシステムで使用されている最も一般的なフラッシュベースのストレージ・デバイスです。SD カードは小型、低消費電力、簡素、低コストなどの特長があるため、ストレージ条件に対する理想的なソリューションです。SD カードはほとんどの機器と互換性があるため、どのコンピュータを使用しても、カードからデータの容易なアクセスや検索ができ、アプリケーションに応じてさらに進んだ処理を行うことができます。

このアプリケーション・ノートでは、SD カードとインターフェースする ADuCM3027／ADuCM3029 プロセッサのシリアル・ペリフェラル・インターフェース (SPI) の使用方法について説明します。

また、ピン配列、通信規格、設計上の考慮事項など、SD カードに関する一般的な情報を詳述するとともに、信号や波形などの SPI の基本事項や ADuCM3027／ADuCM3029 の SPI インターフェースの機能について説明します。

必要なボード類

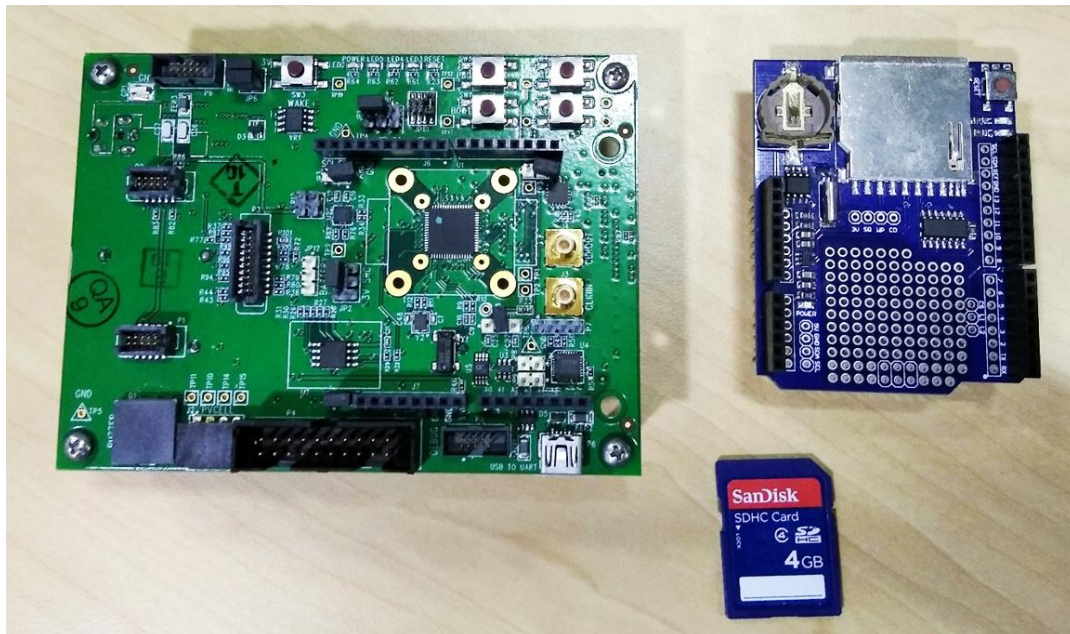


図 1. 必要なボード類：EVAL-ADuCM3029 EZ-KIT 評価用ボード（左）、
Arduino（アルドウィーノ）互換 SD カード・シールド（右上）、SD カード（右下）

アナログ・デバイセズ社は、提供する情報が正確で信頼できるものであることを期していますが、その情報の利用に関して、あるいは利用によって生じる第三者の特許やその他の権利の侵害に関して一切の責任を負いません。また、アナログ・デバイセズ社の特許または特許の権利の使用を明示的または暗示的に許諾するものでもありません。仕様は、予告なく変更される場合があります。本紙記載の商標および登録商標は、それぞれの所有者の財産です。※日本語版資料は REVISION が古い場合があります。最新の内容については、英語版をご参照ください。

目次

はじめに.....	1	ファイル・アロケーション・テーブル (FAT) ファイル・システム.....	5
必要なボード類.....	1	SD カードの実装.....	6
改訂履歴.....	2	ADuCM3027 / ADuCM3029 SPI.....	6
SD カード.....	3	ハードウェアの実装.....	6
容量とタイプ.....	3	ソフトウェアの実装.....	7
インターフェースとモード.....	3	SD カードとマイクロコントローラのインターフェース用のサンプル・コード.....	8
SD カードの SPI プロトコル.....	4	参考文献.....	8

改訂履歴

7/2017—Revision 0: Initial Version

SD カード

SD カードは、携帯機器で使用するマルチメディア・カード (MMC) の機能を拡大するため、1999 年に SD アソシエーション (SDA) によって規格が策定されたメモリ・カードです。それ以来、SD カードは広く普及し、業界標準となり、今ではほとんどの携帯機器で画像や音楽などのファイルを格納するために使用されています。

図 2 に示すように、SD カードは、ピン・インターフェース、メモリ・コア、内部レジスタ、そして内部コントローラで構成されています。メモリ・コアはデータ用の記憶装置で、1 MB ~ 2 TB のさまざまな容量のものがああります。内部レジスタは、SD カードの状態を保存します。ピン・インターフェースは、カードとマスター・デバイス (通常はカードを使用するマイクロコントローラ) を接続するものです。

カードのインターフェース・コントローラは、SD カードのコア・メモリを管理します。通常、このコントローラはフラッシュ内のデータの書き込み、読出し、消去、エラー処理、およびフラッシュのウェア・レベリングを行います。このため、SD カードを実装するメインのマイクロコントローラは、一連のデータ・パケットを介してコマンドやデータをカードのインターフェース・コントローラに送信しますが、メモリ・コアの管理には関与しません。

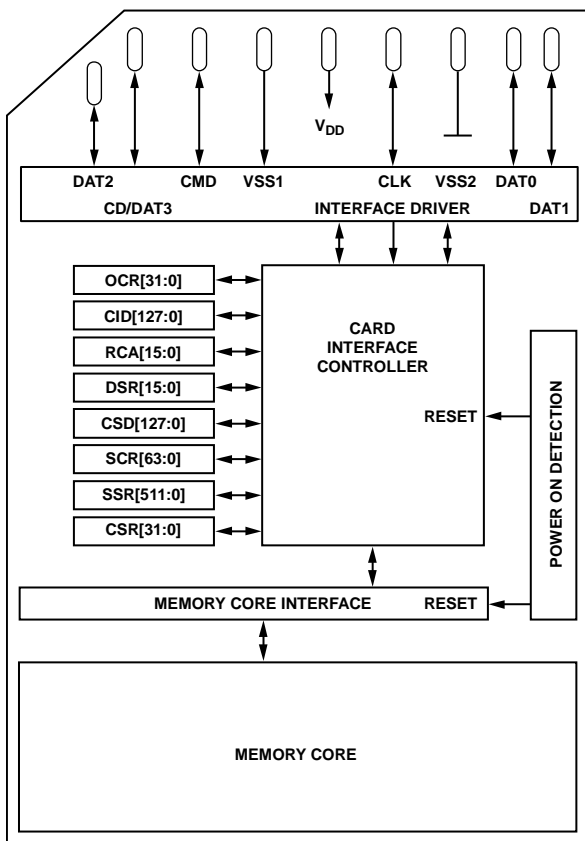


図 2. SD カードの内部構成

容量とタイプ

SD カードには、さまざまなタイプ、大きさ、容量があります。SD カードのタイプは、メモリ容量と準拠する SD 規格によって決まります。

表 1 に、さまざまなタイプの SD カードとそれらの容量を示します。

表 1. SD カードのタイプと容量

Common SD Card Type	Capacity
SD Standard Capacity (SDSC)	1 MB to 2 GB
SD High Capacity (SDHC)	2 GB to 32 GB
SD eXtended Capacity (SDXC)	≥32 GB

また、図 3 に示すように、SD カードには、標準サイズ、ミニ・サイズ、マイクロ・サイズなど、形状や大きさが異なるものがあります。

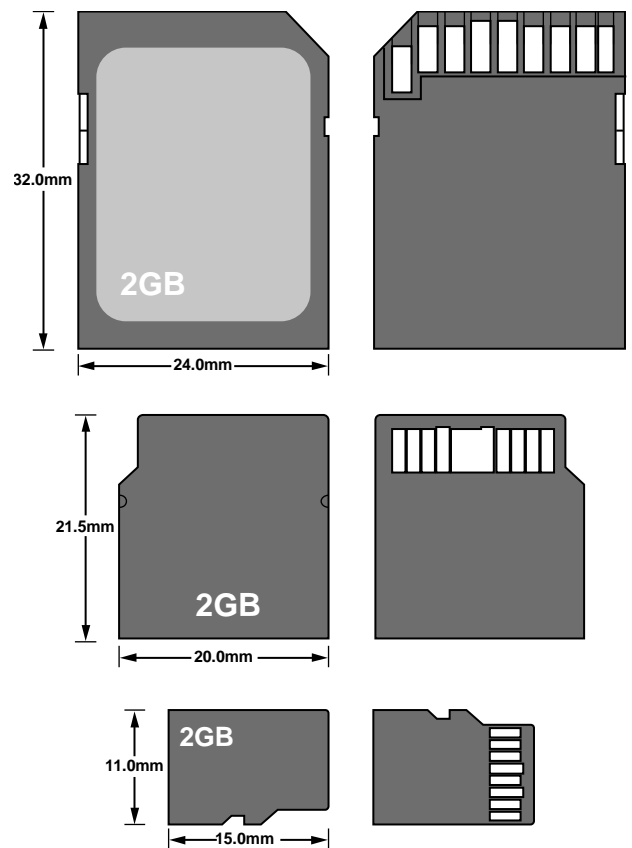


図 3. SD カードの大きさ

インターフェースとモード

SD カードにはピン・インターフェースがあり、使用されている通信モードやカード形状要素に応じて、ピン配列構成が異なります。一般に、このインターフェースは、電源ライン (電源およびグラウンド)、クロック・ライン、データ・ライン、コマンド・ラインで構成されています。最新の高速カードでは、高速で広帯域な転送を実現するために、低電圧差動インターフェースが実装されています。

通常、SD カードとの通信は、SDA で規定されたインターフェースである SD バス・モードで行われます。このモードでは、インターフェースには独立したコマンド・ラインおよびデータ・ライン、ならびにクロック・ラインがあります。転送は独自フォーマットで行われます。ただし、ネイティブ SD インターフェースのない組み込みシステムで、このフォーマットを実装するのは困難です。

ネイティブ SD インターフェースのない組み込みシステムやマイクロコントローラに対応するため、SDA では、SD カードが SPI バス・モードにも対応するように仕様規定しています。このモードでは、SD カードはマイクロコントローラで広く用いられている周知の SPI で動作します。ただし、SPI バス・モードは、SD カードの全標準プロトコルのうち 1 つのサブセットにしかな対応しません。

図 4 に、標準の SD カードのピン配列と micro SD カードのピン配列を示します。

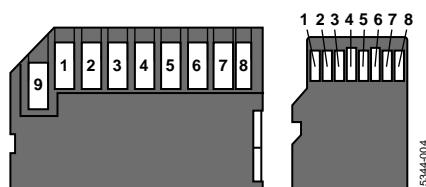


図 4. 標準の SD カードと micro SD カードのピン配列

標準の SD カードのピン機能を表 2 に示します。

表 2. 標準の SD カードの機能

Pin No.	Name	SD Mode	SPI Mode
1	CS/DAT3	Data Line 3	Chip select
2	CMD/DI	Command line	MOSI
3	VSS1	Ground	Ground
4	VDD	Supply voltage	Supply voltage
5	CLK	Clock	Clock (SCK)
6	VSS2	Ground	Ground
7	DAT0	Data Line 0	MISO
8	DAT1	Data Line 1	Unused or IRQ
9	DAT2	Data Line 2	Unused

micro SD カードのピン機能を表 3 に示します。

表 3. microSD カードの機能

Pin No.	Name	SD Mode	SPI Mode
1	DAT2	Data Line 2	Unused
2	DAT3/CS	Data Line 3	Chip select
3	CMD	Command line	MOSI
4	VDD	Supply voltage	Supply voltage
5	CLK	Clock	Clock (SCK)
6	VSS	Ground	Ground
7	DAT0	Data Line 0	MISO
8	DAT1	Data Line 1	Unused or IRQ

SD カードの SPI プロトコル

SPI バス・モードで使用する SD プロトコルは、SD バス・モードで使用するプロトコルとは若干異なります。SPI バス・モードでの SD カードとの通信は、単純なコマンド応答プロトコルを用いて行われ、マスター・デバイス（マイクロコントローラ）がコマンド・フレームを送信することで開始されます。SD カードがコマンド・フレームを受信すると、このカードはホスト・マイクロコントローラが送信したコマンドに応じて、応答フレームまたはエラーフレームを送信して応答します。

SD カードに送信する SD カード・コマンド・フレームは、6 バイトで構成されています。コマンド・フレームは常にビット 01 で始まり、6 ビットのコマンド番号が後に続きます。最初のバイト・パケットの後に、ビッグ・エンディアン・フォーマットの 4 バイトの引数が続きます。最後のバイトは、7 ビットの巡回冗長検査 (CRC) と 1 ビットのストップ・ビットで構成されています (図 5 参照)。

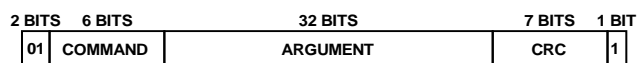


図 5. コマンド・フレームのフォーマット

SD カードは送信されたコマンド・フレームに応じて、異なる応答フレームを用いて各コマンド・フレームに応答します。SPI バス・モードで使用可能な応答タイプは、R1、R3、R7 の 3 通りのみです (図 6 参照)。

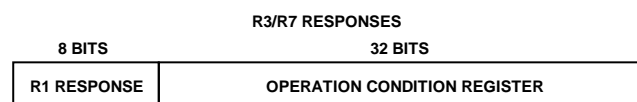
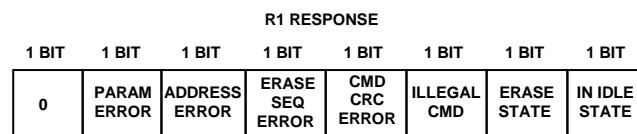


図 6. 応答フレームのフォーマット

次に、SD カードは SEND_STATUS コマンドを除き、すべてのコマンドの後に、R1 応答トークンを送信します。このコマンドは 1 バイト長で、最上位ビット (MSB) は常に 0 に設定されています。その他のビットはエラー表示で、エラーは 1 で示されます。エラー・フラグの意味は次のように規定されています。

- アイドル状態：カードはアイドル状態で、初期化プロセスを実行中です。
- 消去リセット：消去範囲外シーケンス・コマンドを受信したため、消去シーケンスが実行前にクリアされました。
- 不当コマンド：不当コマンド・コードが検出されました。
- 通信 CRC エラー：最後のコマンドの CRC が失敗しました。
- 消去シーケンス・エラー：消去コマンドのシーケンス中にエラーが発生しました。
- アドレス・エラー：ブロック長が一致しない誤ったアドレスがコマンド内で使用されました。

- パラメータ・エラー：コマンドの引数（アドレスやブロック長など）がこのカードの許容範囲外でした。

SD カードの SPI モードでは、SD モードで使用するコマンドの 1 つのサブセットのみが使用されます。SPI モードのコマンドの詳細なリストについては、表 4 を参照してください。コマンド・セットは、カードの初期化、いくつかの重要な詳細情報の検索、およびメモリ・カードとの間のデータ・ブロックの読み書きに限定されています。

表 4. SPI モードのコマンド設定

Command	Description
CMD0	Reset card
CMD8	Request for current operational conditions
CMD55	Leading command for application specific commands (ACMDs)
ACMD41	Start card initiation process
CMD58	Request for operation conditions register (OCR)
CMD16	Change block length
CMD17	Read a block of data
CMD24	Write a block of data
CMD32	Set the start block to be deleted
CMD33	Set the end block to be deleted
CMD38	Start block erase set by CMD32 and CMD33

ファイル・アロケーション・テーブル (FAT) ファイル・システム

一般的に使用されている SD カードは、デフォルトで FAT ファイル・システムとしてフォーマットされています。FAT ファイル・システムは、ほぼすべての PC、モバイル・デバイス、組み込みシステムで広くサポートされている伝統的なファイル・システムです。このファイル・システムをサポートするドライバは、小型化、堅牢化、軽量化が可能で、マイクロコントローラや組み込みシステムに容易に実装できます。FAT でフォーマットされたメモリ・カードに保存された任意のファイルは、この基本ファイル・システムに対応可能であるため、ほぼすべての PC で表示や変更が可能です。

SD カードの仕様では、異なる SD カードで容量別に使用できる FAT ファイル・システムのタイプを定めています（追加情報については、参考文献のセクションを参照してください）。

SD カードの実装

ADuCM3027/ADuCM3029 SPI

ADuCM3027/ADuCM3029 マイクロコントローラには、高速センサーやメモリ・デバイスなどのさまざまな SPI 互換デバイスとの通信に使用可能な 3 つの SPI インターフェース (SPI0、SPI1、SPIH) が搭載されています。各 SPI ポートには、4 個の SPI 対応デバイスを制御する 4 つのハードウェア・チップ・セレクト信号があります。SPI バスのペリフェラルには、プログラマブル・ポーレート、クロック位相、クロック極性、ハードウェア・フロー制御のさまざまなメカニズムも含まれており、SPI マスターまたは SPI スレーブとして機能することができます。

3 つの SPI は、接続されている内部バス・インターフェースを除き、SPI のプログラミングとモデルの視点から見ると同一です。SPIH ペリフェラルは、プロセッサ・クロックと同じレートのクロックで動作する高性能なアドバンスト・ペリフェラル・バス (APB) に接続されます。SPI0 と SPI1 はメインの APB に接続されます。ADuCM3027/ADuCM3029 マイクロコントローラのペリフェラルの多くは、APB を使用します。このため、アービタレーションを必要とするモジュール数がきわめて多くなり、遅延が安定しなくなります。

したがって、高データ・レートの場合、SPIH を使用したほうが効率が高く、低遅延でデータを転送できます。

ハードウェアの実装

SD カードと ADuCM3027/ADuCM3029 プロセッサとの間のインターフェース機能を実証するために使用するハードウェアは次のとおりです。

1. EVAL-ADuCM3029 EZ-KIT® ボード
2. Arduino® (アルドゥイーノ) 互換データ・ロギング・シールド
3. 4 GB の SanDisk® SD カード

EVAL-ADuCM3029 EZ-KIT ボード

EVAL-ADuCM3029 EZ-KIT ボードは、ADuCM3027/ADuCM3029 プロセッサ用の評価システムです (図 7 参照)。このボードには、ADuCM3027/ADuCM3029 マイクロコントローラを評価することのできる部品が豊富に搭載されています。さらに、EVAL-ADuCM3029 EZ-KIT には、EI3 インターフェースと Arduino インターフェースも搭載されています。これらのインターフェースは、さまざまなドーター・ボードやシールドに接続可能で、EVAL-ADuCM3029 EZ-KIT ボードの機能を拡張することができます。

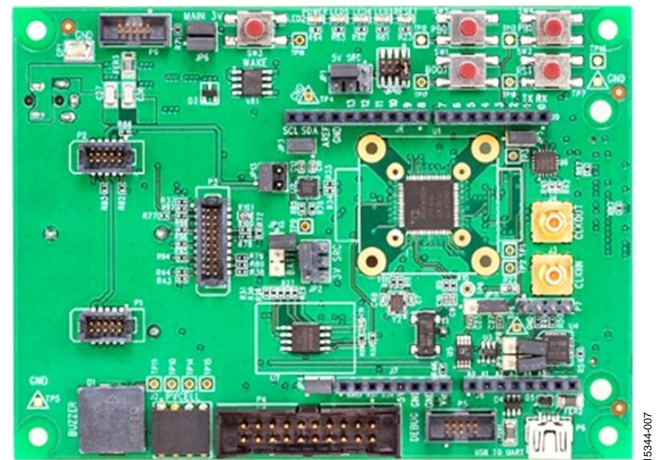


図 7. EVAL-ADuCM3029 EZ-KIT ボード

Arduino 互換データ・ロギング・シールド

使用するデータ・ロギング・シールドは、ほとんどの Arduino ロギング・プロジェクトで一般に使用されているシールドです (図 8 参照)。このロギング・シールドの SPI は、ADuCM3027/ADuCM3029 マイクロコントローラの SPIH ポートに接続されます。

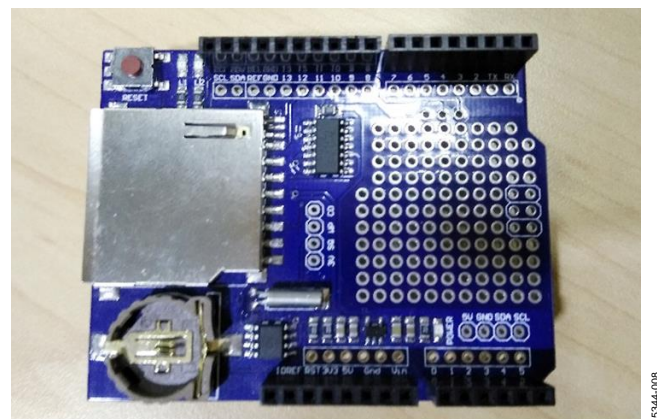


図 8. データ・ロギング・シールド

ほとんどのデータ・ロギング・シールドには、相補型金属酸化膜半導体 (CMOS) バッファ (例えば、CD4050) が搭載されており、3.3 V から 5 V への電圧変換器として機能し、SD カードを損傷から保護します。通常、SD カードのデータ入力、シリアル・クロック、およびチップ・セレクト・ラインは、カードの入力ピンであるため、バッファで保護されています。ただし、SD カードを ADuCM3027/ADuCM3029 マイクロコントローラとインターフェースさせる場合は、マイクロコントローラの出力が SD カードと同じ電圧レベルであるため、この電圧変換器は省略可能です。

SPIH インターフェースと汎用入出力 (GPIO) のポート 1 は、ADuCM3027/ADuCM3029 マイクロコントローラのピンを共有しています。SPIH、SCLK、MOSI、および MISO の信号ラインもマイクロコントローラのピン (ピン P1_02、ピン P1_03、およびピン P1_04) を使用します。このため、これらのピンは未使用のままにしておく必要があります。

このアプリケーション・ノートでは、GPIO のピン P2_01 を SD カード用のチップ・セレクトとして使用し、SPIH 専用のチップ・セレクトは使用していません (図 9 参照)。

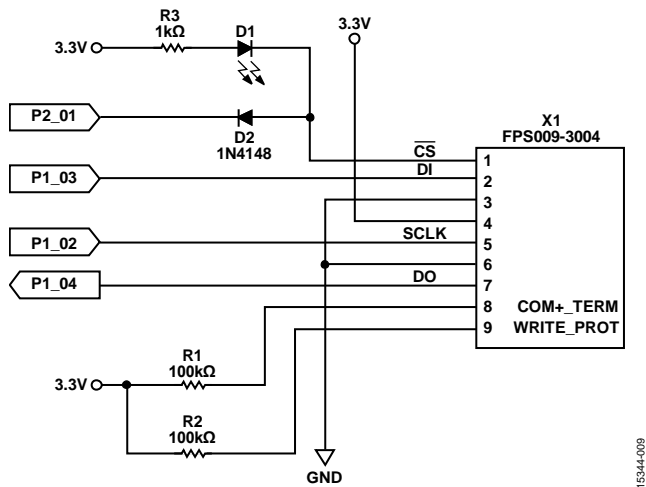


図 9. データ・ロギング・シールドの回路図

ソフトウェアの実装

このアプリケーションをテストするには、次のソフトウェア・ツールを使用します。

- IAR Embedded Workbench 7.60。このソフトウェアをダウンロードするには、EVAL-ADuCM3029 EZ-KIT のウェブページを参照してください。
- IAR 用の ADuCM302x ソフトウェア。EVAL-ADuCM3029 EZ-Kit のウェブページからダウンロード可能。
- ChaN による FatFs ライブラリ R0.12a。FatFs - Generic FAT Filesystem Module のウェブサイトからダウンロード可能。

アプリケーションは、ファイル・システム・コールを処理し、FAT でフォーマットされた SD カード内のファイルの読出しと変更を可能にする FatFs ライブラリを使用します。このファイル・システム・ライブラリを使用するには、下位のハードウェア依存のファンクション・コールを処理するハードウェア抽象化レイヤを実装する必要があります。

SPI インターフェースの初期化

ADuCM3027/ADuCM3029 の SPIH ペリフェラルを初期化するには、以下の手順を実行します。

1. SPIH ペリフェラルの設定を行います。
2. GPIO ピン P2_01 をチップ・セレクトとして設定します。
3. GPIO1 と SPIH の共有ピンについてのピン・マルチプレクスを設定します。

SPIH ペリフェラルの設定

SPIH ペリフェラルの設定は、SPIH ペリフェラルを SD カードと通信するための準備完了状態に設定する複数のコールで構成されています。SPIH ペリフェラルを設定するには、以下の手順を実行します。

1. SPIH ペリフェラルを開きます。

2. ビット・レートを設定します。SPIH のビット・レートの変化は、マイクロコントローラと SD カードとの相互作用の内容に依存することに注意してください。

- マイクロコントローラが SD カードを通信用に初期化する場合、ビット・レートは 100 kHz ~ 400 kHz である必要があります。
- SD カードを SPI モードに設定後は、ハードウェアの設計とマイクロコントローラの容量に応じて、ビット・レートを 20 MHz まで高くすることができます。

3. 連続モード動作に設定します。連続モードでは、SPI ペリフェラルは複数バイトのデータの送受信が実行でき、転送を中止したり中断したりすることはありません。

以下に示すのは、ADuCM3027/ADuCM3029 マイクロコントローラの SPIH を設定するために使用されるコードの例です。

```
static uint8_t SPIMem[ADI_SPI_MEMORY_SIZE];
static ADI_SPI_HANDLE spih_Dev;

// Open the SPI
adi_spi_Open(SPI_DEV_NUM, SPIMem,
             ADI_SPI_MEMORY_SIZE, &spih_Dev);
```

```
// Set the bit rate
adi_spi_SetBaudrate(spih_Dev, 100000);
```

```
// Set the continuous mode
adi_spi_SetContinuousMode(spih_Dev, true);
```

チップ・セレクトの設定

チップ・セレクトは、SPI 専用のチップ・セレクトではなく、GPIO ピン (ピン P2_01) を使用します。カスタマイズされた GPIO ピンを使用すると、チップ・セレクト信号を完全に制御できます。SD カードとマイクロコントローラ間のトランザクションによっては、ソフトウェアがチップ・セレクトを適切に制御するように、チップ・セレクト信号の特別処理が必要となる場合があります。

チップ・セレクト・ピンを設定するには、以下のコードを使用して、SPIH チップ・セレクト・オプションを none に設定し、次に、GPIO ピン P2_01 を出力に設定します。

```
adi_spi_SetChipSelect (spih_Dev,
                       ADI_SPI_CS_NONE);
adi_gpio_OutputEnable (SPI_CS_PORT, SPI_CS_PIN,
                       true);
adi_gpio_SetHigh(SPI_CS_PORT, SPI_CS_PIN);
```

マイクロコントローラのピン・マルチプレクサの設定

GPIOx_CFG レジスタは、ADuCM3027/ADuCM3029 マイクロコントローラのピン・マルチプレクサの設定値を保持する設定レジスタです。SPIH では、ピン P1_02、ピン P1_03、ピン P1_04 を SD カードへの接続ピンとして使用します。

これらのピンを SPIH 用に設定するには、REG_GPIO1_CFG レジスタで適切なビット・オプションを設定します。このレジスタの詳細については、マニュアル *ADuCM302x Ultra Low Power ARM Cortex-M3 MCU with Integrated Power Management Hardware Reference* を参照してください。

以下のコードは、ポート設定レジスタを SPIH 用に設定する例です。

```
#define SPI0_SCLK_PORTP1_MUX
((uint32_t) ((uint32_t) 1<<4))
#define SPI0_MISO_PORTP1_MUX
((uint32_t) ((uint32_t) 1<<8))
#define SPI0_MOSI_PORTP1_MUX
((uint32_t) ((uint32_t) 1<<6))

*((volatile uint32_t *) REG_GPIO1_CFG) =
SPI0_SCLK_PORTP1_MUX |
SPI0_MISO_PORTP1_MUX |
SPI0_MOSI_PORTP1_MUX;
```

データの送受信

SPI ペリフェラルを SD カードとの通信用に設定すると、インターフェースは、データ・パケットの送受信と SD カードの起動が可能になります。adi_spi_ReadWrite 関数が SD カードとのデータの送受信を行います。この関数は、データと SPI デバイス・ポインタを保持する構造体命令が必要です。以下に示すのは、このセクションで説明したコードの例です。

```
ADI_SPI_TRANSCEIVER spi_xcv_buff;
spi_xcv_buff.pTransmitter = txbuff;
spi_xcv_buff.pReceiver = rxbuff;
spi_xcv_buff.TransmitterBytes = txsize;
spi_xcv_buff.ReceiverBytes = rxsize;
spi_xcv_buff.nTxIncrement = 1;
spi_xcv_buff.nRxIncrement = 1;

adi_spi_ReadWrite(spih_Dev, &spi_xcv_buff);
```

ADI_SPI_TRANSCEIVER は、このトランザクションで使用されるバッファを保持する構造体命令です。送信、受信、データ・サイズ、およびインクリメント用のバッファが定義される必要があります。

送信バッファと受信バッファは、送信または受信されるデータに応じたサイズの uint8_t の配列です。送信のデータ・サイズは約 6 バイト～8 バイトで、受信のデータ・サイズは 1 バイト～512 バイトです。

adi_spi_ReadWrite 関数はブロッキング関数であり、マイクロコントローラは、トランザクションが成功してはじめて次の命令を実行することに注意してください。

SD カードとマイクロコントローラのインターフェース用のサンプル・コード

このアプリケーション・ノートには、SD カードと ADuCM3027/ADuCM3029 をインターフェースするサンプル・コードが付属しています。サンプル・コードを見るには、このプロジェクトを解凍し、IAR Embedded Workbench 7.6 にインポートしてください。このサンプル・コードは、EVAL-ADuCM3029 EZ-KIT の製品ウェブページからダウンロード可能です。

参考文献

ADuCM302x Ultra Low Power ARM Cortex-M3 MCU with Integrated Power Management Hardware Reference. Analog Devices, Inc. 2016.
SD Specifications Part 1, Physical Layer Simplified Specifications, Version 5.00. SD Card Association. 2014.