8 - SPI インターフェース

注文コード:
SMAR1-SP1-...　SPI + ラインドライバーインクレメンタル出力

8.1 SPI インターフェースの紹介

SPI は、Serial Peripheral Interface の頭字語です。 これは、マイクロコントローラーと、センサー、ADC、DAC、シフトレジスタ、SRAM などの周辺 IC との間で最も広く使用されているインターフェースの 1 つです。 SPI は、同期の全二重マスタースレーブベースのインターフェースです。Lika エンコーダは常にスレーブデバイスであり、SPI モーインターフェースポートします。マスターまたはスレーブからのデータはクロック SCLK の立ち上がりエッジと同期され、SCLK のアイドル極性は重要ではありません。 SPI は事実上の標準になっていますが、それは正当な標準ではありません。 言い換えれば、それは公式に指定されていません。

8.2 インターフェース

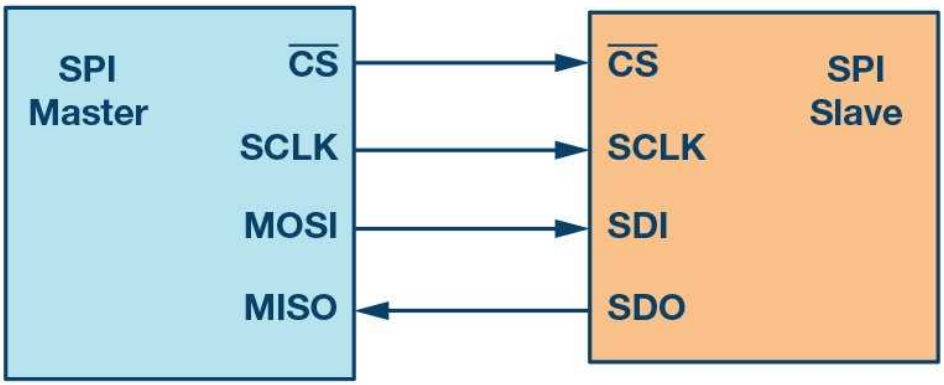

図 8-マスターと 1 つのスレーブを使用した SPI 構成

SPI バスは、4 本の単方向ワイヤで構成されています。
Lika エンコーダは 4 線式 SPI デバイスであり、次の信号を伝送します。

- クロック：SPI CLK、SCLK
- インターフェースの有効化：チップセレクト（CS）
- データ入力：MISO（マスター入力、スレーブ出力）、マスター用 SDI。 スレーブ用 MOSI（マスターOUT、スレーブ IN）
- データ出力：MISO（マスター入力、スレーブ出力）、スレーブ用 SDO。 マスター用 MOSI（マスターOUT、スレーブ IN）

8.2.1 SCLK SPI クロック信号

クロック信号を生成するデバイスはマスターと呼ばれます。 マスターとスレーブ間で送信されるデータは、マスターによって生成されたクロックに同期されます。 前述のように、SPI モード 0 および 3 がサポートされています。つまり、SCLK のアイドルレベルは 0 または 1 であり、データは立ち上がりエッジで受け入れられます。

モードは、送信が開始されたとき（CS がローにプルダウンされたとき）の SCLK レベル（極性（CPOL）と呼ばれることもあります）に依存し、サンプリングエッジ（CPHA）は次の表に示すとおりです。

モード	極性 (CPOL)	相 (CPHA)
0		
3		

SPI インターフェースはマスターを 1 つだけ持つことができ、1 つまたは複数のスレーブを持つことができます。 図 8 は、マスターとスレーブ間の SPI 接続を示しています。

8.2.2 CS チップ選択信号

マスターからの CS チップセレクト信号は、スレーブを選択するために使用されます。 これはアクティブロー信号であり、スレーブを SPI バスから切断するためにハイにプルされます。 複数のスレーブを使用する場合は、マスターからスレーブごとに個別の CS チップセレクト信号が必要です。 Lika エンコーダでは、CS チップセレクト信号は常にアクティブロー信号です。

8.2.3 MOSI 及び MISO 信号

MOSI と MISO がデータラインです。 MOSI はマスターからスレーブにデータを送信し、MISO はスレーブからマスターにデータを送信します。 データは、8 ビットのパッケージでバイトごとに送信され、msb（最上位ビット）が最初に送信されます。 各データ送信は、SPI マスターによって 1 バイトのオペコードが送信されたときに開始されます（35 ページの「8.2.5 コマンドバイト」を参照）。

8.2.4 Dデータ送信

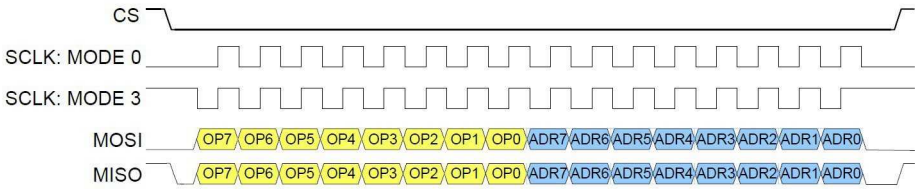

図 9-例としてオペコード読み取りレジスタ（シングル）を使用した SPI 送信 SPI モード 0 および 3

以下に、**READ REGISTER（シングル）コマンド**が送信されると仮定した場合の、SPI データ送信の一般的なシーケンスの説明を示します（図 9 を参照）。

1. SPI 通信を開始するために、マスターは送信を初期化します。マスターは SCLK クロック信号を送信し、CS チップセレクト信号を有効にしてスレーブを選択します。CS チップセレクトはアクティブロー信号です。したがって、マスターはスレーブを選択するためにこの信号でロジック 0 を送信する必要があります。

2. エンコーダは通信レベルを MOSI から MISO に変更します。SPI は全二重インターフェースです。 マスターとスレーブの両方が、それぞれ MOSI ラインと MISO ラインを介して同時にデータを送信できます。 SPI 通信中、データは同時に送信（MOSI / SDO バスにシリアルシフトアウト）および受信（バス-MISO / SDI-のデータがサンプリングまたは読み込まれます）されます。 シリアルクロックエッジは、データのシフトとサンプリングを同期させます。 前述のように、データは 8 ビットのパッケージで送信されます。出力データの長さは 8 ビットの倍数である必要があります。

3. これで、マスターは MOSI を介してオペコード OP とアドレス ADR を送信します。
4. エンコーダは MISO を介して OP と ADR を即座に出力します。
5. エンコーダは、アドレスに従って要求されたデータを送信します。
6. マスターは CS の立ち上がりエッジでコマンドを終了します。
7. 最後に、エンコーダは MISO 出力を 1 に切り替えます。

8.2.5 コマンドバイト

各通信フレームはコマンドバイトで始まります。 これは、操作のタイプを指定する操作コードと 8 ビットアドレスで構成されます。
利用可能な SPI コードは次のとおりです。

コード	説明
0xA6	**SDAD** 送信コマンド (センサーデータ SD)
0xF5	**SDAD** ステータスコマンド (ラッチ無し)
0x97	**REGISTER** (シングル) 読出しコマンド
0xAD	**REGISTER** ステータス/ データコマンド

SDAD 送信コマンド

SMAR1 は、CS がゼロ（REQ / LATCH）のときに、SCLK の最初の立ち上がりエッジで絶対位置をラッチします。 SMAR1 はセンサーデータ（SD）をすぐに出力できるため、マスターは **SDAD 送信コマンド**を直接送信できます。 センサーデータシフトレジスタ（サイズは 8 の倍数で 8～40 ビット）が切り替えられ、MOSI と MISO の間でクロックアウトされます。
無効なデータがシフトレジスタにサンプリングされた場合、エラービットが SPI STATUS バイトに設定されます（ページの **REGISTER status / data コマンド**の表を参照してください）。
38）そして出力データバイトはゼロに設定されます。

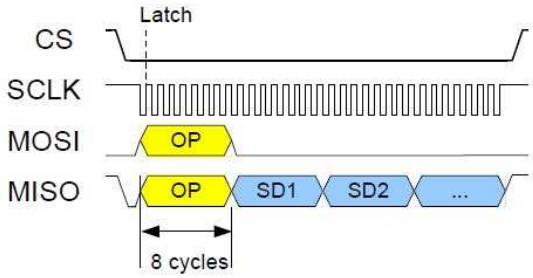

図 10 - SDAD 送信: SD読み取り

SDAD ステータスコマンド

マスターが接続されたスレーブの処理時間を知らない場合、**SDAD status コマンド**を使用してセンサーデータを要求できます。 コマンドの原因は次のとおりです。

1. すべてのスレーブは PACTIVE を介してアクティブ化され、SVALID レジスタを MOSI と MISO の間で切り替えます。
2. 次の SPI 通信の SCLK の最初の立ち上がりエッジで開始されたセンサーデータの次の要求は、スレーブによって無視されます。

変換の終了は、SVALID（SV）によって通知されます。 このコマンドを使用して、マスターは変換の最後までポーリングできます。 センサーデータは、**SDAD 送信コマンド**を介して読み取られます。

SVALID (SV)	
コード	説明
0	センサーデータが無効
1	センサーデータが有効

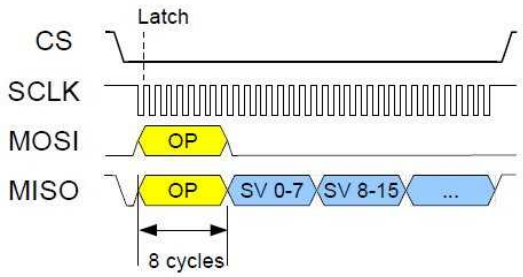

図 11 - SDAD ステータス

図 12 は、2 つの **SDAD ステータスコマンド**と **SDAD 送信コマンド**の相互作用を示しています。 **SDAD ステータスコマンド（1）**で各センサーデータ通信を開始する必要はありません。 SMAR1 には処理時間がないため、有効なセンサーデータを直接出力できます。 このため、コマンドシーケンスは **SDAD 送信コマンド（2）**で開始できます。 これに続いて、**REGISTER status / data コマンド**を実行して、失敗した SPI 通信（通信不良）を検出する必要があります。

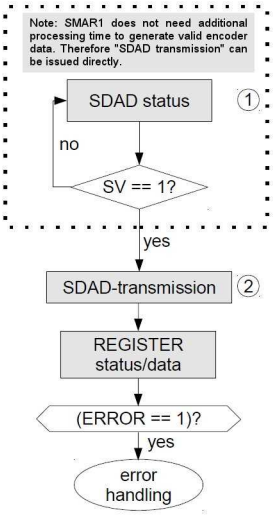

図 12-コマンド SDAD ステータス/ SDAD 送信のシーケンス例

REGISTER（シングル）コマンドの読み取り

このコマンドを使用すると、レジスタデータをスレーブからバイト単位で読み取ることができます。 マスターは、最初に **Read REGISTER（single）**コマンドを送信し、次にアドレス ADR を送信します。 スレーブはすぐにコマンドとアドレスを MISO に出力します。

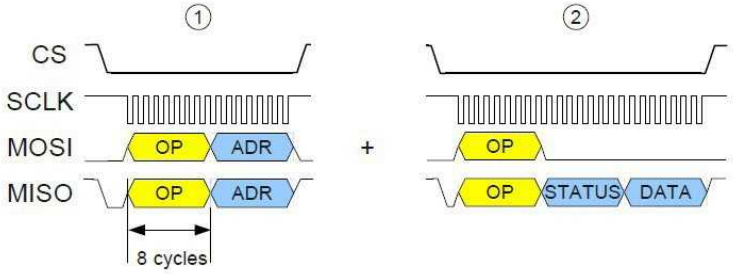

図 13-レジスターの読み取り（シングル）; 読み出しアドレス（1）及びコマンド REGISTER ステータス/データを読み出しデータ（2）に設定

これに続いて、**REGISTER status / data コマンド**を使用して、マスターは SPISTATUS バイトに続く DATA の有効性が SPI STATUS を介して通知されるまでポーリングできます。

注記

エンコーダのステータスとアクティブエラーがあるかどうかを知るには、レジスタ 76 と 77 を読み取る必要があります。詳細については、42 ページの「10.3 ステータスレジスター SPI インターフェイス」を参照してください。

REGISTER（単一）コマンドの書き込み

(現時点では使用されていません)

このコマンドを使用すると、データを 1 バイトずつスレーブに書き込むことができます。マスターは最初に **Write REGISTER（single）コマンド**を送信し、次に ADR とデータ（DATA）をアドレス指定します。 スレーブはすぐにコマンド、アドレス、データを MISO に出力します。

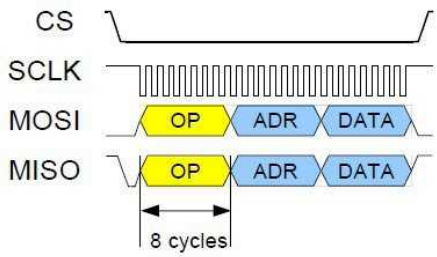

図 14-書き込みレジスタ（シングル）; 書き込みアドレスとデータを設定

REGISTER status / data コマンドを使用して、マスターはレジスタ通信の最後までポーリングできます（SPI STATUS バイトを介して通知されます）。

REGISTERステータス/データコマンド

REGISTER status / data コマンドを使用して、最後のレジスタ通信および/または最後のデータ送信のステータスを要求できます。 SPI ステータスバイトには、次の表に要約されている情報が含まれています。

ビット	名称	ステータスレポートの説明
7	エラー	オペコードが実装されていません、センサーデータが読み出し時に無効でした
6 ... 4	Reserved（予約済み）	
3	DISMISS（解除）	アドレス拒否
2	FAIL（失敗）	データ要求が失敗
1	BUSY（稼働中）	以前のリクエストのためにスレーブがビジー状態です
0	VALID（有効）	DATA is valid

論理レベル: 1 = true（真）; 論理レベル 0 = false（偽）

すべての SPI ステータスビットは、レジスタアクセスごとに更新されます。
ルールの例外は ERROR ビットです。 このビットは、スレーブとの最後の SPI 通信中にエラーが発生したかどうかを示します。

マスターは REGISTER ステータス/データコマンドを送信します。 スレーブはすぐにオペコードを MISO に渡します。 次に、スレーブは SPISTATUS バイトと DATA バイトを送信します。

Read REGISTER（シングル）コマンドと Write REGISTER（シングル）コマンドに続いて、DATA バイトの有効性が VALID ステータスビットで通知されます。
要求されたデータバイトは、READ REGISTER（シングル）コマンドに続いて DATA を介して送り返されます。 Write REGISTER（single）コマンドに続いて、書き込まれるデータが DATA バイトで繰り返されます。 他のすべてのオペコードでは、DATA バイトは定義されていません。
図 15 は、 REGISTER（シングル）読出しコマンド、 REGISTER 書き込み（シングル）コマンド、および REGISTER ステータス/ データコマンドの相互作用を示しています。

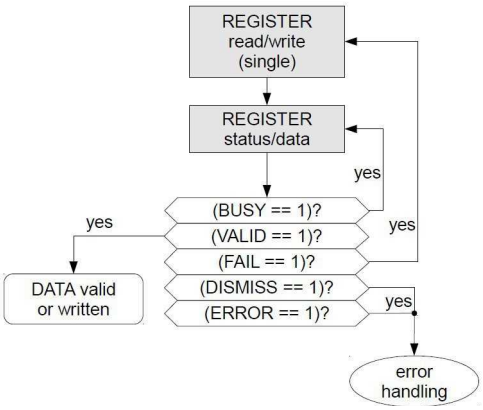

図 15 : **REGISTER** 読出し/書き込み及び **REGISTER** ステータス/データコマンドシーケンス例