



# STM32L4 - SPI

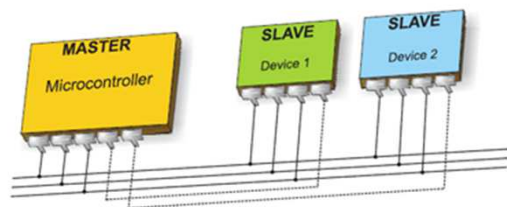
Serial Peripheral Interface

Revision 1



Jan - 2016

こんにちは。STM32シリアル・ペリフェラル・インタフェースSPI  
(エス・ピー・アイ)のプレゼンテーションです。



- 簡単なシリアル通信インタフェース
  - 高度な設定可能
  - 標準の同期プロトコルをサポート

### アプリケーションの利点

- 必要ピンがわずか
- 外付け部品やデバイスとの簡単な統合

内蔵シリアル・ペリフェラル・インタフェースは、マイクロコントローラが、近くの外部デバイスと通信することを可能にする、シンプルな通信インタフェースを提供する。このインタフェースは、多くの同期標準の通信プロトコルをサポートする高度な設定が可能である。

アプリケーションは、わずか数のピンを必要とするコンポーネントに、簡単に直接接続の恩恵を受ける。高度な設定が可能な周辺機器のおかげで、多くのデバイスを単純に既存のプロジェクト内に収納することができる。

- 動作モード
  - マスタまたはスレーブ (マルチマスタ& マルチスレーブ対応)
  - 全二重、単線、半二重
  - モトローラとTIプロトコルサポート
- $f_{PCLK/2}$  まで動作
  - 最小2ワイヤでのインタフェース (スレーブ選択マネージメントオプション)
  - データとクロックフォーマットは設定可能
  - プロトコルレベルでの追加サポート (Tx と Rx FIFOs, DMA, CRC)
  - 割り込み機能を備えた幅広いイベントのフラグ

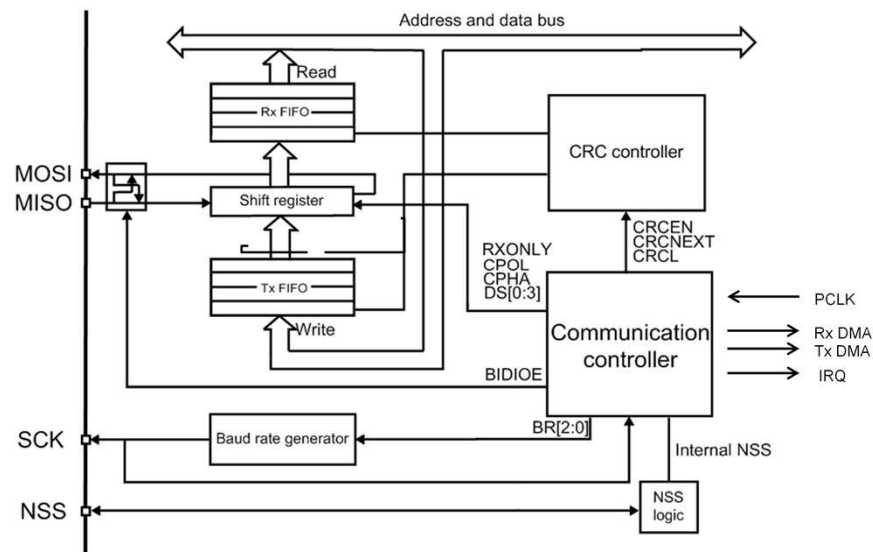


STM32のSPIは、このプレゼンテーションでより詳細に説明される種々の動作モードを提供してい。

通信速度は、内部バス周波数の半分を超えることはできず、最小2本のワイヤは、クロック信号によって同期したシリアル・データを単一方向に送るために必要とされる。オプションとして、ハードウェアスレーブ選択制御信号を付加することができる。データサイズと送信シフト順序並びに、クロック信号の極性及び位相は、設定できる。プロトコルレベルでは、ユーザーは、特定のデータ・バッファをオプションの自動巡回冗長検査CRCと(発音は「シー」「アール」「シー」)で計算、転送データはダイレクトメモリアクセスDMA(発音は「ディー」「エム」「エー」)を通して、使用することができる。割り込み要求を発生することができる幅広いSPIイベントがある。

## ブロック図

4



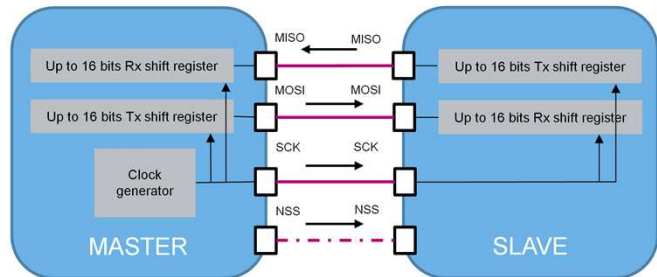
簡略化された、SPIのブロック図は、その基本的な制御機構と機能を示す。SPIペリフェラルに関連付けられた4つのI/O信号がある。すべてのデータは、特定のインタフェースを介して受信送信バッファを通過する。制御ブロックの機能は、構成に応じて有効または無効にしている。

# SPIノードの相互接続

5

## 様々なマスタスレーブ相互接続をサポート

- マスタが常にクロックを提供し、すべてのトラフィックを制御する(通信スレーブは選ばれる)
- データは並列に両方の方向で交換することができる。
- **全二重モード**(双方向通信)両方のマスタとスレーブで送信と同時にデータを受け取る



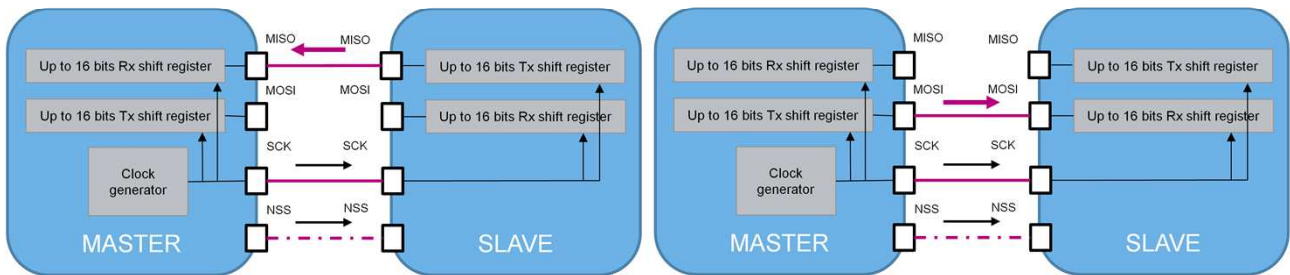
SPIマスタは、常にバストラフィックを制御し、SCKラインを介して専用のスレーブへのクロック信号を提供する。マスタは、オプションのNSS信号を介して通信したいスレーブを選択することができる。専用のシフトレジスタに格納されたデータはMOSI-マスタ出力スレーブ入力とMISO - マスタ入力スレーブ出力を介してマスタとスレーブの間で同期して交換することができる。マスタとスレーブノードが全二重モードで通信するとき、両方のデータラインが使用され、同期データは、同時に両方向に流れる。

# SPIノードの相互接続

6

様々なマスタスレーブ相互接続をサポート

- 単線モード (片方向通信)  
片方のボードは送信専用、もう片方のノードは受信専用



単線モードでは、1つのノードがデータを送信し、他の一つは、それを受信する。データは一方方向のみに流れ、専用データラインは、専らデータ転送のために使用される。未使用のSPIピンは、他の目的に使用することができる。

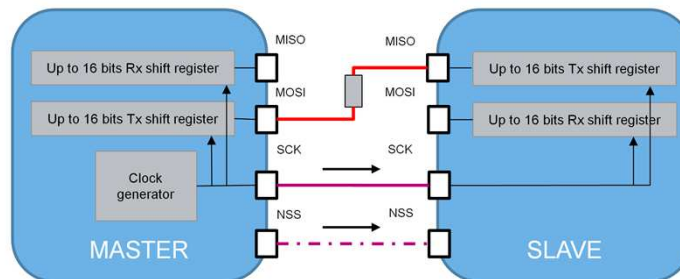
## SPIノードの相互接続

7

### 様々なマスタスレーブ相互接続をサポート

- 半二重モード (準双方向通信)

両方のマスタおよびスレーブは同期したデータの送受信をひとつおきに行う。ノードは、データ転送のための単一の共通ラインを共有している。



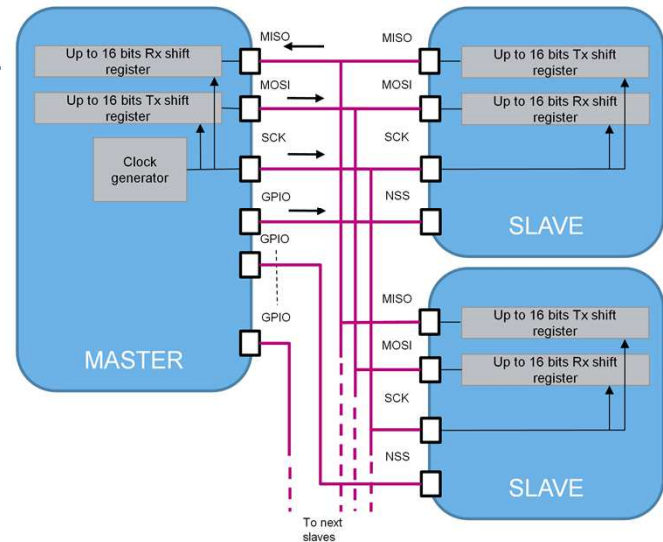
半二重モードは一度に一方向のみに流れるデータ交換およびデータのための単一の行を共有することにより、前の2つのモードを内蔵している。このモードでは、マスタMOSIと、スレーブのMISOピン間の相互接続がある。マスタとスレーブは、この共通データ線を使用するときに、送信と受信の役割を同期して交互する必要がある。これは通常、同期されていないマスタとスレーブノードにおけるデータの方法を再構成するので、一時的な短絡接続を防止するための半二重データラインに直列抵抗を付加することが一般的である。

# SPIノードの相互接続

8

## マルチスレーブネットトポロジーのサポート

- マルチスレーブ: スター型トポロジー
  - 通常、マスタは、データ転送のために単一のスレーブノードを選択する
  - この場合、(GPIOピンによってシミュレートする)別のスレーブ選択信号が必要とされる
  - 異なるクロックおよびデータフォーマットはスレーブノードで使用することができる



SPIネットワークが、複数のスレーブを含む場合、すべてのSCK、MOSIとMISO信号が相互接続されたときに、スター型トポロジーが一般的に使用される。

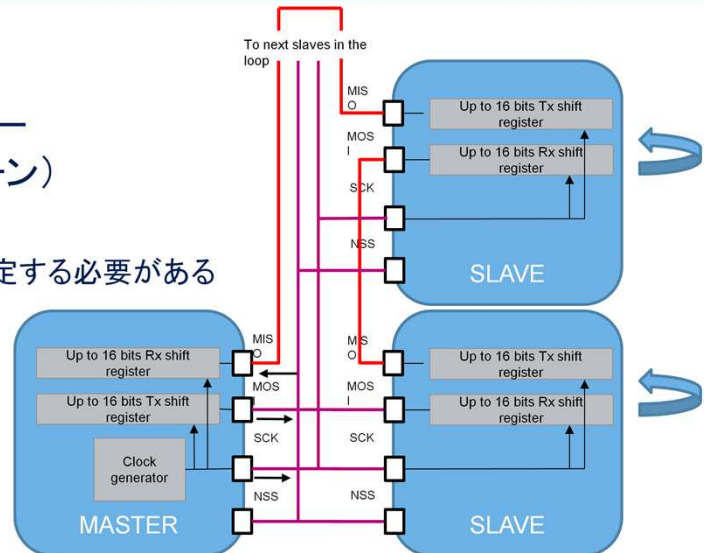
通常、マスタは、ちょうど1スレーブが共通のMISOピンを介してマスタにデータを送信することができるので、一度に一つのスレーブと通信する。この場合には、スレーブからの読出し時は、マスタから異なるスレーブ選択信号を、各スレーブノードに提供しなければならない。マスタは、通信セッションのためにただ1つのスレーブを選択する。別個のスレーブ選択信号が印加されると、別のSPIデータとクロックのフォーマット設定は、スレーブノードに対して、必要に応じて使用することができる。

# SPIノードの相互接続

9

## マルチスレーブネットトポロジーのサポート

- マルチスレーブ: 循環型トポロジー (デジーチェーン)
- データが全てのノードを巡回する
- 全てのノードは同じデータ、クロックで設定する必要がある



別のマルチスレーブ構成は、ノードのすべてのデータの入力と出力が閉じたシリアルチェーンで互いに接続されている円形のトポロジーである。

通信として使用されている一般的なスレーブ選択信号が同時に発生する。すべてのノードが同じデータおよびクロック・フォーマットの設定を維持する必要がある。スレーブSPIノードがマイクロコントローラによって提供される場合、内部での送信と受信のシフトレジスタは、通常、物理的に分離されている。

そのように、それらの間で転送されるデータは、この循環型のモードでソフトウェアによって処理されなければなりません。また、マスタノードは、これらの内部転送を提供するためのデータとの間に十分な時間を提供しなければならない。

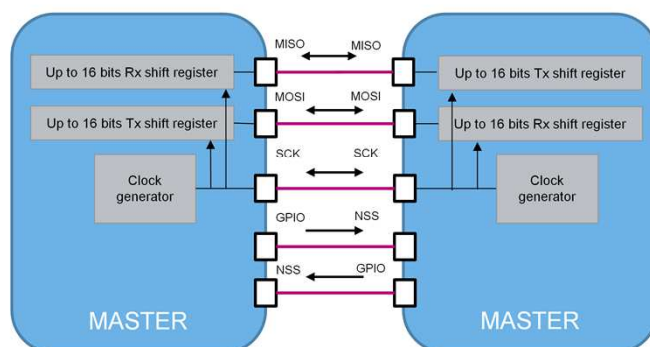
# SPIノードの相互接続

10

## マルチマスタのトポロジーをサポート

- マルチマスタ：マスタ機能を持つ2つのノード

- バスがアイドル状態のとき、ノードは、デフォルトではスレーブモードである
- 通信セッションの開始時に、1つのノードが一時的にバスの制御を取るために、アクティブマスタモードに自身を切り替える
- スレーブ選択ピンは、潜在的なバスの競合を検出するための入力として使用される



SPIネットワークは、マルチマスタ環境で動作することができる。このモードは2つのマスタノードを排他的に接続するために使用される。

デフォルトでは、両方のノードは、ネットワークがアイドル状態のときに、スレーブモードになっている。

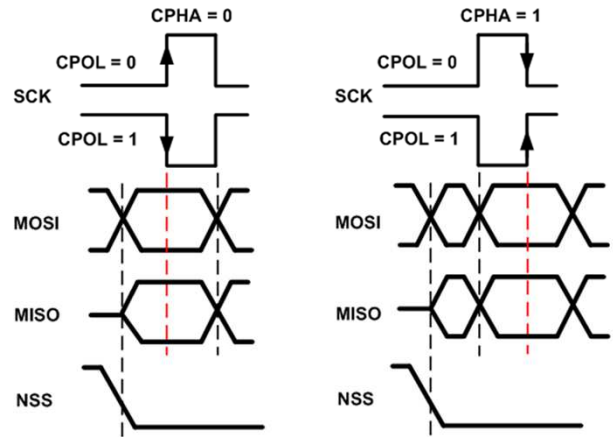
一方のノードがバスの制御を取りたい場合は、マスタモードに切り替わり、GPIOピンを介して他のノード上のスレーブ選択信号をアサートする。スレーブ選択ピンは、一度に一つのマスタがSPIバスを使用できるように、ノード間の潜在的なバス衝突を検出するためのハードウェア入力として動作する。セッションが完了したら、マスタノードはスレーブ選択信号を解放し、次のセッションの開始を待っているパッシブスレーブモードに戻る。

# データフレームフォーマット

11

完全にプログラム可能で柔軟なフォーマット

- データフレームのサイズ
  - 4 から最大16bit
- ビットオーダー
  - MSB または LSB first
- クロック設定 (モード0-3)
  - アイドル時Low または High 極性
  - odd または even エッジでのサンプリング



データ形式を設定するために使用されているいくつかのコントロールがある。

ユーザは、データフレームサイズおよびデータビットの送信順序を定義することができる。

クロックは、モトローラSPI仕様で定義された4つの基本的な構成のいずれかに設定することができる。2つのビットの組み合わせは、クロック信号の極性と位相を制御する。

位相制御ビットがクリアされると、データビットが奇数クロックエッジでサンプリングされ、さらには偶数クロックエッジに同期してデータラインの次のビットをシフトする。

位相制御ビットがセットされている場合、逆になる。

クロック極性ビットクロックの位相設定に応じて、データのサンプリングまたはシフトするために使用されるクロックエッジの、初期クロック信号の状態とアイドルとを定義する。

# データパッキング、FIFO アクセス

12

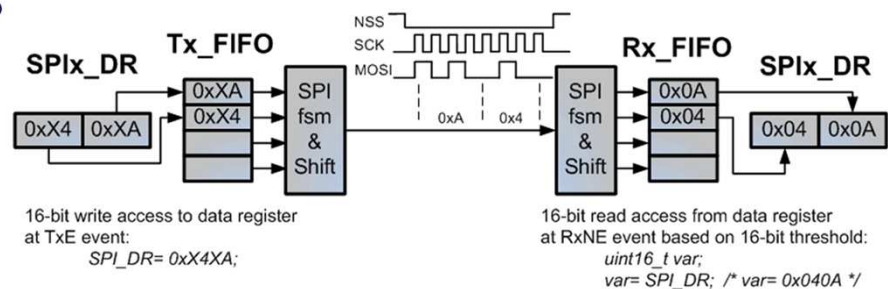
## 高度な低負荷制御

- パッキングモード

- いくつかのデータパターンによるFIFOレジスタのアクセス
- FIFO スレッシュホールドレベルの設定
- DMA アクセス

⇒ イベントの数と要求されるサービスを減らす

⇒ システムの付加を減らす



通信速度が高速でデータフレームが短い場合には、クロック信号が連続的になり、全二重モードを使用する場合、正しいデータフローを確保するのが厳しい要求になる。スレーブノードは、適切にデータオーバーランまたはアンダーラン状態を防止するために、マスタから送信されたすべてのトランザクションにサービスを提供する必要があります。

データフレームのサイズがバイトに収まる場合には、パッキングモードを使用することができます。その後、複数のデータパターンをFIFO（発音はファイフォ）のレジスタへの単一のアクセスで書き込むまたは読み出すことができます。同時にFIFOスレッシュホールドイベントを適切に設定すると、サービスへのイベントの数は減少し、より良好なデータの流れを制御できる。DMAを併用する場合には、システム上の全体的な負荷が大幅に低減される。





図では、2つの短い4ビットのデータフレームは、専用のFIFOレジスタに1つの16bitアクセスによって読み書きできるというの原理を見ることができる。読出またはデータの書き込みアクセスは、単に1つのイベントを発生させることによって行われる。

# 32bitのRx/Tx FIFO

13

## スレッシュホールドとデータのアクセスとの間のバランス

- 送信と受信のための2つの別々の32bitのFIFO
- 8bitまたは16bitの読出し/書込みアクセス対FIFOのスレッシュホールドと占有フラグ
- 8bitアクセスでのTxとRx FIFOでの異なる機能

Rx & Tx FIFO occupancy			TxLVL	TxNE	RxNE	
16-bit	8-bit				16-bit	8-bit
0		00	1	0	0	0
1/4		01	1	0	1	1
1/2		10	1	1	1	1
>1/2		11	0	1	1	1

\*) Max 3x 8-bit for TxFIFO, 4x 8-bit for RxFIFO



SPIペリフェラルは、データフローを処理するために2つの32bitFIFOを備えている。FIFOは、8bitまたは16bitのデータアクセス命令のいずれかを使用してアクセスすることができる。受信時には、FIFOから生成されるイベントは、スレッシュホールドの設定に依存する。このテーブルでは、イベントフラグの動作が、設定に応じてどのように変化するかを概要を説明する。これは、データの一貫性が失われないスレッシュホールド設定とFIFOへのアクセスのバランスを維持することが重要である。送信時に、FIFOの占有率は、データアクセスに依存する。システムは、8bitの書込みアクセスがFIFOの後半を埋めるために適用されると、FIFOが完全には使用されないため、送信FIFOへの次のアクセスを予測することはできない。この場合、送信FIFOが完全に占有されていなくても、TXEフラグが結果としてクリアされる。

# プロトコルレベルにおけるその他のサポート

14

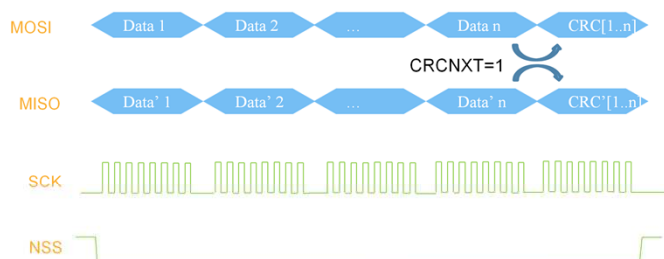
## 拡張 DMA と CRC 管理

- DMA は有利である

- データトランザクションイベントの正確な数
- 転送終了を以下でハンドリング
  - CRC 制御
  - FIFO スレッシュホールド制御

- CRC

- 受信および送信フローの区切りで計算
- CRC は転送終了時に送信:
  - 送信器はCRC結果を直接データシフトレジスタに書き込む
  - 受信器は、RX FIFO内のCRCを蓄え、内部で計算した値を比較する
- プログラマブルなCRC多項式(偶数多項式はサポート外)とCRC長(8bitまたは16bitのCRCフレーム)



プロトコルレベルでの通信中に、DMAは、CRCパターンを適用したり、データの正確な数は、転送された後に正しく設定FIFOスレッシュホールドを変更するために最適に用いることができる。スレッシュホールド制御のために、フレームの数は、パケットサイズと位置合わせされていない場合でも、最後の奇数データフレームが正しくパックされたモードで適用される。CRCが有効になっている場合、分離したCRC計算は、送信機と受信機に使用される。CRC計算の結果は、ソフトウェアにより自動的にDMA転送終了時に適用される。

送信器CRC演算結果は、レジスタからシフトレジスタ内に直接ロードされ、受信されたCRC値が、受信機のCRC演算レジスタの結果と比較してFIFOに格納されている。計算に使用されるCRC多項式はプログラム可能であり、及びCRCパターンの長さが8bitまたは16bitのいずれかのフレームに設定することができる。

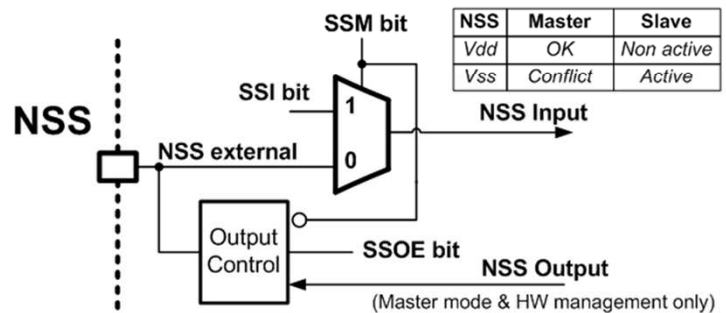
## スレーブ選択信号(NSS)の拡張管理

### • NSS入力

- ハードウェアまたはソフトウェアでの管理
- スレーブモード – アクティブなスレーブを選択
- マスタモード – マスタ間の競合

### • NSS出力

- マスタモード
  - アクティブなスレーブを選択
  - 特別なモード



スレーブ選択信号(NSS)は、一般的にマスタノードによって、通信を行うためのスレーブノードを選択するのに、使用される。一般に、マルチスレーブ・トポロジで使用されているが、単一のマスタ-スレーブのペアのデータフローを同期させるためにも使用することができる。スレーブ選択信号を入力として、または出力のいずれかとして動作することができる。

NSS入力は、外部からハードウェアによって、または内部からSSMとSSIビットに応じてソフトウェアにより管理することができる。

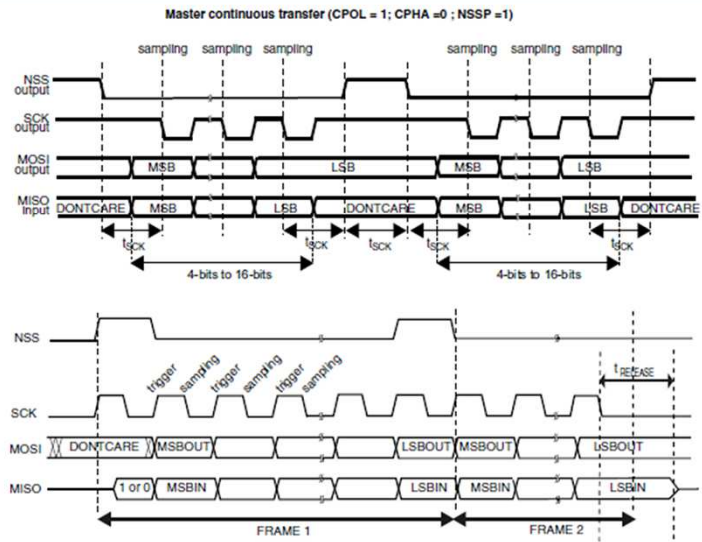
スレーブとして、常に入力として動作し、スレーブ通信を有効にする。マスタ入力として、マルチマスタシステム内のマスタ間の競合を示す。マスタ出力としては、標準または特定の制御モードにおけるハードウェアによって管理される。一般に、標準モードでは、マスタスレーブ選択出力をソフトウェア制御下のGPIOで置き換えることができる。

# 特別な NSS モード

16

## スレーブ選択信号の(NSS)ハードウェア制御とエンハンスメントモード

- NSS パルスモード
  - マスタのみサポート
  - モトローラモード (CPHA = 0 のみ)
- TI モード
  - マスタとスレーブでサポート
  - 固定の CPOL と CPHA 設定
  - スレーブ MISO の HiZ は自動制御



スレーブ選択信号は、特定のハードウェアの制御下にあるときに、いくつかの拡張モードがある。

データの連続転送があるときに、スレーブ選択信号はパルスモードで動作することができる。これは、連続してデータを転送しているとき、マスタはデータフレーム間の、1つのSPIクロック時間にNSSの出力信号にパルスを生成する。

データは、2つのSPIクロック時間によってインターリーブされる。クロック位相は、このモードでは固定されてる。

別の拡張モードは、TIのモードで、データの最後のビットに、マスタによって提供されたNSSパルスによって同期される。クロック極性と位相の設定が固定され、スレーブデータ出力は、自動的にバストラフィックが停止したときにハイインピーダンスに切り替え、特定の設定可能なタイムアウトにされる。

割り込みイベント	説明
Transmit FIFO ready	送信FIFOが新しいデータを受け入れる準備ができたときにセット
Receive FIFO ready	受信FIFOがデータを受信したときにセット
Master mode fault	マルチマスタバス構成で、バス競合が検出された場合にセット
Data overrun error	RX FIFOがフルの時、受信器がその次のデータを受信できなかった
TI frame format error	NSS信号がデータフォーマットに対応していない

- FIFOスレッシュホールドに達したときに、DMA要求は間接モードで生成することができる



これはSPI割り込みイベントの概要である。  
データフローを処理するためにFIFOとエラー検出イベントがある。  
DMA要求は、FIFOスレッシュホールドイベントによって内部的にトリガされる。

# Low-power モード

18

モード	説明
Run	アクティブ
Sleep	アクティブ。ペリフェラル割り込みはデバイスのsleepモードを終了させる原因になる
Low-power run	アクティブ
Low-power sleep	アクティブ。ペリフェラル割り込みはデバイスのLow-power sleepモードを終了させる原因になる
Stop 1	停止。ペリフェラルのレジスタは保持
Stop 2	停止。ペリフェラルのレジスタは保持
Standby	パワーダウン。ペリフェラルはStandbyモード復帰後、再初期化必要
Shutdown	パワーダウン。ペリフェラルはShutdownモード復帰後、再初期化必要



これは、特定の低電力モードでのSPIステータスの概要である。デバイスはがStop、Standby、またはShutdownモードで通信を行うことができない。ペリフェラルがStopまたはパワーダウン・モードに入る前に、すべてのSPIトラフィックが完了したことを確認することが重要である。

- 理論的な通信速度の上限はPCLK/2
- 通信の実際の速度は以下に依存:
  - SPI バスの負荷容量 (いくつかのデバイスが接続されているか、入力容量、ワイヤの長さ)
  - GPIO の内部接続、設定、VDD レベル、周辺温度
  - SPI クロック信号のDuty比
  - データのセットアップとホールド時間
  - ソフトウェアのパフォーマンス
- パフォーマンス
  - 最大速度マスタで 40 MHz
  - 最大速度スレーブで 24 MHz



SPIのパフォーマンスは、ペリフェラルに適用される内部クロックに主に依存する。

ペリフェラルクロックは、達成可能な最大通信周波数の少なくとも2倍にする必要がある。通信の実際の速度は、多くのアプリケーションの因子によって下がっていく。

ユーザは、ノードの数、接続距離、入力容量、ならびにGPIO設定としてSPIバスの負荷を考慮しなければなりません。高速GPIOモードでは、データ信号とクロック信号に適用する必要があります。低い電源電圧、極端な周囲温度はまた、GPIOのエッジが遅くなります。時にはデータのスローなホールドまたはセットアップ時間が、ノード間で尊重されなければならない。

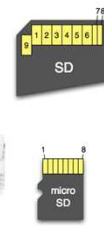
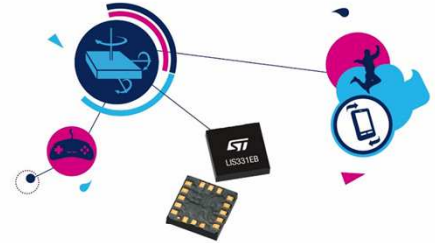
高速なデータの流処理するために、頻繁な割り込みの応答に高速なアプリケーションソフトウェアが必要とされる。

DMAは、システムがより多くのDMAチャンネル、頻繁に割り込みサービスを使用するか、または非割り込み命令(例えばLDMIA)を実行する際にも考慮されなければならない。

# アプリケーション例

20

- ディスプレイ
- スマートセンサ
- メモリ
- MMC/SD カード
- IO エクスパン



Pin	SD	SPI
1	CD/GAT3	CS
2	CMD	DI
3	VSS1	VSS1
4	VDD	VDD
5	CLK	SCLK
6	VSS2	VSS2
7	DAT0	DO
8	DAT1	X
9	DAT2	X

Pin	SD	SPI
1	DAT2	X
2	CD/GAT3	CS
3	CMD	DI
4	VDD	VDD
5	CLK	SCLK
6	VSS	VSS
7	DAT0	DO
8	DAT1	X

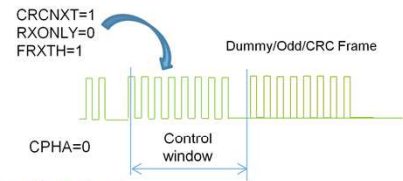


SPIは、複雑な通信プロトコルを必要とすることない単純なデータ転送で、必要とされる広範囲の用途に使用することができる。スマートカードを使用した場合など保護された転送もサポートされている。

# アプリケーションのTipsやトリック

21

- 一般的なTips:
  - SPI(またはそのクロック)を無効にする前に、BSYとFIFO占有状態をチェック
  - 具体的な制御が必要な場合に、DMAを使用する(CRC、RX FIFOのスレッシュホールド、受信専用モードの終了)
  - データがバイトに収まるときにパケットモードを使用すべき
  - NSSのハードウェアの管理は利益をもたらす
- 特別なもの:
  - CRC情報が受信FIFOにロードされ、ユーザがそれをフラッシュしなければならない
  - BSYの動作は、連続したデータランザクションの間、マスタおよびスレーブモードで異なっている
  - 受信、オーバーランフラグが送信動作中にも設定される(これらは無視してください)
  - CRC使用時のDMA tips
    - 全二重モードでのDMA転送のデータサイズ:
      - 送信時 = 送信データ数 (CRC長は含まない)
      - 受信時 = 受信データ数 + CRC長 (Rx-only モードの時 = 受信データ数)



ここではいくつかの有用なTipsがある:

- ユーザーは、DMAランザクションが完了するか、送信FIFOが空になった場合であっても、バス上のトラフィックがまだ進行中であることを認識しなければならない。
- このため、ユーザーが慎重にペリフェラルのステータスを確認し、SPIを無効にしたり、ストップモードにする前に提案した手順に従う必要がある。
- CRC処理を必要とする特定のコントロールをする場合は、DMAを使用し、または受信FIFOスレッシュホールド制御を行ったり(データ数がパケットモードに整列されていない場合)、または受信専用モードでデータの正確な量を受信する。
- このようなコントロールは、以下のすべてのランザクションが適切に処理されるように特定の時間の間に排他的フレームランザクション内で適用する必要がある。
- DMAおよびデータパッキングの使用は、システム全体のパフォーマンスを向上させることができる。これらの機能は、データフレームが短く、高速で連続的な通信フローが要求される場合に役立つ。

- スレーブ選択信号のハードウェア管理は、単一マスタ/単一スレーブペアで特に必要はありませんが、データの流れを同期させ、マルチマスタシステムでの競合を防ぐことができる。

SPIネットワークを設計する際に考慮されるべきいくつかの追加がある。

受信機は常に受信FIFOにCRC情報をロードする。ユーザーは、バッファにCRCがあることを考慮し、それをフラッシュする必要がある。

ビジーフラグは、任意のデータ処理に使用されるべきではなく継続的なトラフィックをチェックする。BSYビットは、マスタデータの連続トランザクション中にデータフレームの間も設定されたままになる。通信は、連続的であるかどうかは関係なく、データ・フレームとの間の少なくとも1つのSPIクロックサイクルLowになる。

ノードがデータを送信のみするときも、受信フローはアクティブなまま。ユーザーは、この場合のすべての受信および関連するオーバーランイベントを無視する必要がある。

CRCは、モードに依存しているなどの場合のデータサイズは、DMAによって処理される。データサイズは、受信専用モードよりも全二重モードでデータを受信または送信するときに異なるように構成されなければならない。

# STM32L4 インスタンス機能一覧

22

SPI 機能	SPI1	SPI2	SPI3
Hardware CRC 計算	Yes	Yes	Yes
Rx & Tx FIFO	Yes	Yes	Yes
NSS パルスモード	Yes	Yes	Yes
TI モード	Yes	Yes	Yes



STM32L4内には3つのSPIインスタンスがあり、それぞれ、我々が議論してきたすべての機能をサポートしている。

- これらの他の周辺機器を参照のこと:
  - Reset と clock 制御 (SPI clock イネーブル, Sleep モード時のクロック制御, Reset)
  - Interrupt (FIFO とエラーイベント)
  - General-purpose inputs/outputs (speed 制御, GPIO 設定)



これは、SPIに関連する周辺機器の一覧である。必要に応じて、これらのペリフェラルトレーニングを参照のこと。

- 詳細については、以下を参照のこと
  - AN4286 - SPI protocol used in the STM32 bootloader
  - AN3364 - Migration and compatibility guidelines for STM32 microcontroller applications
  - Web (connection examples, available monitoring tools)



いくつかの専用のSPIアプリケーションノートがある。一般的に、SPIの詳細については、SPIのトピックおよびSPIバスの監視ツールを議論する多くのWebページがある。多くのデジタル・オシロスコープは、SPIバスの直接のデコードをサポートしている。