



STM32L4 - SWPMI

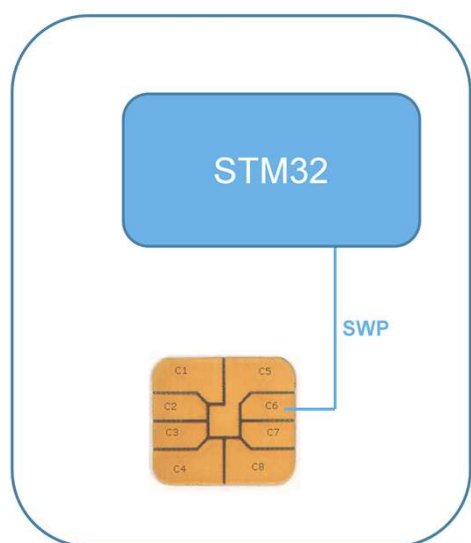
Single **W**ire **P**rotocol **M**aster Interface

Revision 1



Jan - 2016

こんにちは、STM32L4単線プロトコル・マスタ・インタフェース SWPMI(エス・ダブリュー・ピー・エム・アイ)のプレゼンテーションへようこそ。こちらでは、マイクロコントローラにスマートカードを接続するために使用されるこのインタフェースの主な機能について説明します。



- マスタ・モードで、ETSI TS 102 613標準で定義されたシングルワイヤ・プロトコルに従って、全二重シングルワイヤ通信インタフェースを実装

アプリケーションの利点

- SWPトランシーバはSTM32に内蔵
- スマートカードのC6コンタクトでの単線全二重通信は、100キロビット/秒から2メガビット/秒まで



SWPMIIはSTM32の製品の内部に統合され、マスターモードで、ETSI TS102613標準で定義されたシングルワイヤプロトコルに準拠して、全二重シングルワイヤ通信インタフェースを実装しています。

STM32は、SWPトランシーバを実装しています。アプリケーションは、2メガビット/秒までの全二重通信のためのスマートカードへの容易な単ピン接続の恩恵を受ける。

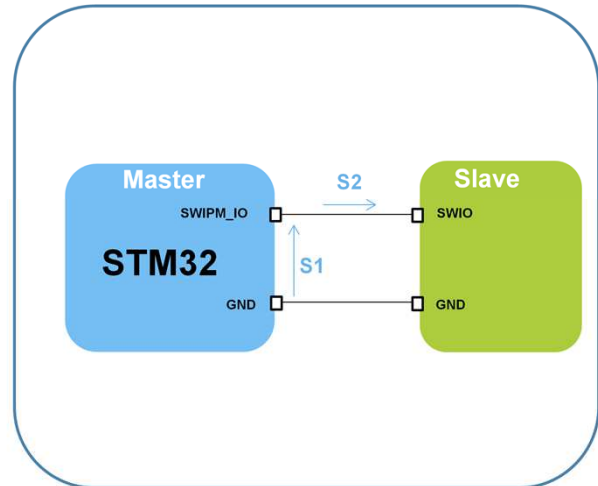
- 3つの送受信モード
 - **No Software Buffer (no DMA)**
 - 割込みまたはポーリング
 - **Single Software Buffer (DMA)**
 - フレームの送信または受信中にソフトウェアの介入なく、最後のフレームのみ必要
 - **Multi Software Buffer (サーキュラーモードでのDMA)**
 - いくつかのフレームは、ソフトウェアの介入なしに処理することができる
- Class B (3 V) と Class C (1.8 V) の電圧を供給



STM32製品内蔵SWPMIは、DMAありなしなど後に詳細を説明する、3つの動作モードを提供しています。STM32は、クラスBおよびクラスCの両方の動作電圧の供給をサポートしています。

単線全二重通信

- S1信号は、マスタからスレーブに、電圧ドメインでデジタル変調(LまたはH)で送信する(0/1.8 V)
- S2信号はスレーブからマスターに、デジタル変調(LまたはH)された電流ドメインで送信する(0/800 μ A)

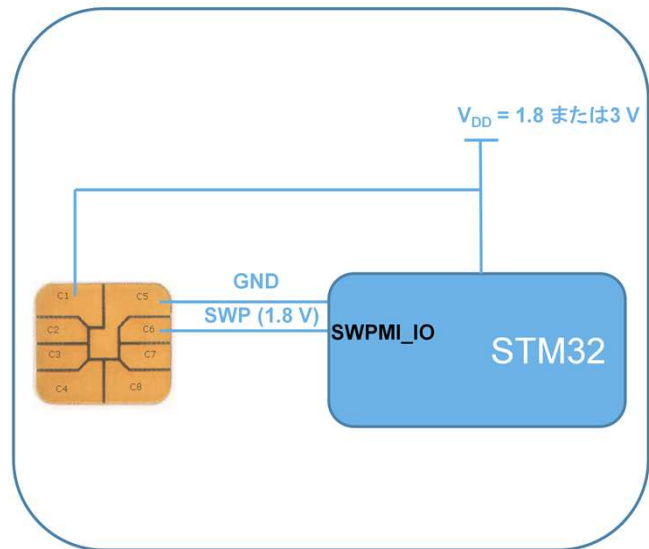


SWPは以下の原理による単線の全二重です。S1信号は、マスタからスレーブへの電圧ドメインで送信されます。S2信号は、スレーブからマスタへ電流ドメインで送信されます。

Class B と Class C

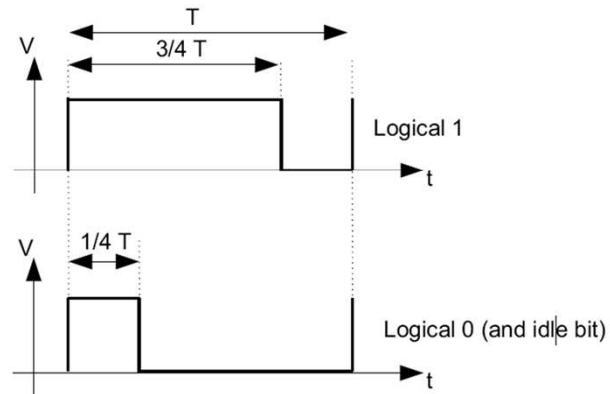
5

- STM32で以下のどちらかを選択する必要がある:
 - Class B: $V_{DD} = 3\text{ V}$
 - Class C: $V_{DD} = 1.8\text{ V}$
- SWPピンの電圧はSTM32によって常に1.8Vになっている
 - クラスBが選択されている場合、SWPMI_IOレジスタ内部レギュレータは、1.8Vに電圧を調整する



電源電圧やクラスは、ソフトウェアの初期化中にSTM32で選択する必要があります。VDDが3 Vである場合、STM32 SWPMI_IOレジスタ内部専用の1.8 Vレギュレータは、SWPの電圧を調整するために使用されます。

- S1 信号: マスター(STM32)からスレーブ(smartcard) へ
- S1はPWMでビットコーディングされている
 - 送信クロック
 - データ



S1信号は、STM32マスタからスマートカードスレーブに、送信されます。S1コード論理0(アイドルビット)に25%のデューティサイクル、S1コード論理1に75%のデューティサイクルを使う。S1信号の周波数は、送信クロックを決定します。

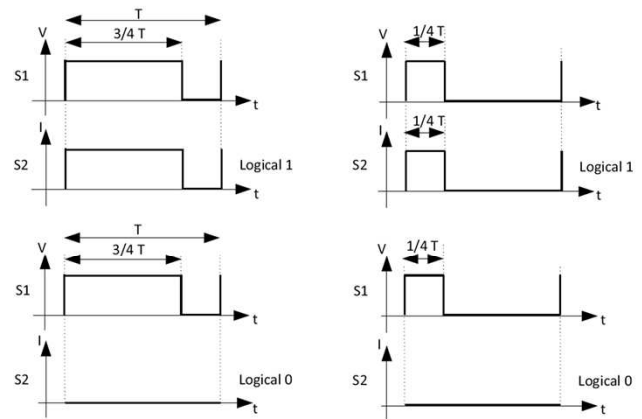
Slide 6

MJM1 I think that "T" (temperature) should be "t" (time) in the diagram.
Mike MILLER, 2015/09/08

S2 信号

7

- S2 信号: スレーブから(smartcard) マスターへ(STM32)
- マスターがHigh状態としてS1を送信すると、スレーブは、電流を引っ張る(High)または何もしない(Low)のいずれかで、S2信号を送信します。



S2信号はスレーブのスマートカードによって、マスタのSTM32に送信されます。スレーブはS1がHighの間、電流を引っ張ることによって、論理1を送る。もし、S1がHighの間、スレーブが電流を引っ張らなければ、それは論理0です。

Slide 7

MJM2 I think that "T" (temperature) should be "t" (time) in the diagram.
Mike MILLER, 2015/09/08

SWP フレーム構造

8



- Start Of Frame (SOF) = 7Eh = 01111110b
- End Of Frame (EOF) = 7Fh = 01111111b
- ペイロードは1から30のデータ Byteを含む
- ビットスタッフィング
 - 論理値1を持つビットが5つ続いた場合は、値0を持つ追加のスタッフィングビットを挿入する
 - ビットスタッフィングでSOFとEOFを区別することができる
- 16bitのCRC ($X^{16}+X^{12}+X^5+1$) をEOFの前に挿入する

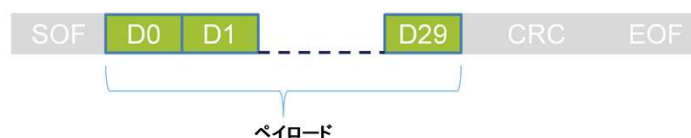


SWPフレームは16進形式で7E Byteで符号化されたフレームの開始フィールドで始まり、16進形式で7F Byteで符号化されたフレームの終端フィールドで終わる。ペイロードは、データの1から30 Byteが含まれている。プロトコルは、ビットスタッフィングを実装しています。余分なビットは、1が5つ連続した場合に挿入されます。これがスタートとエンドフレームがペイロード Byteと区別されることを保証します。データの整合性は、16bitの巡回冗長検査(CRC)によって保証されています。

SWPフレームはSWPMIによってハンドリング

9

ソフトウェアはペイロードフィールドのみ管理する



- SWPMI は以下を自動的に行う

- Start of Frame (SOF) 挿入、削除
- End of Frame (EOF) 挿入、削除
- スタッフィングビットの挿入、削除
- CRC-16 の計算、生成、確認



SWPMIが自動的に、スタートフレーム、ストップフレーム、スタッフィングビット、CRCを処理します。この方法では、ソフトウェアはペイロードデータだけを管理する必要があります。

SWP バスステート

10

- 活性化な遷移



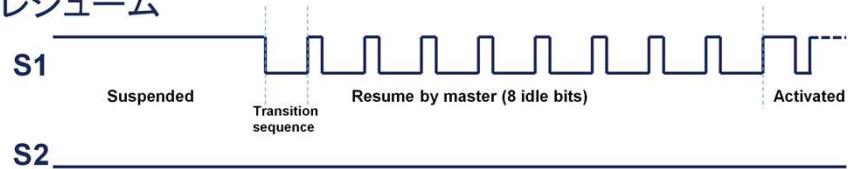
- 不活性化な遷移



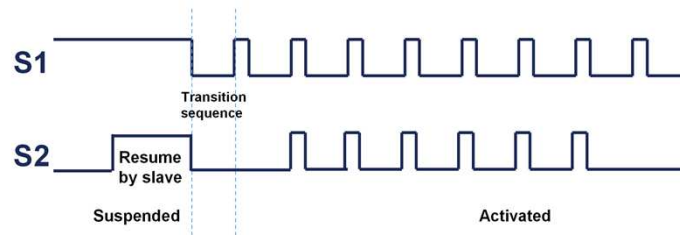
いくつかの状態は、SWPバスのために定義されています。非活性状態では、S1信号がローレベルです。すべての通信を開始する前に、マスターはサスペンド状態でSWPを設定するためにハイレベルにS1信号を上げる必要があります。通信が不要になったら、SWPは、マスターによって不活性化することができます。

• レジューム遷移

• マスターによるレジューム



• スレーブによるレジューム

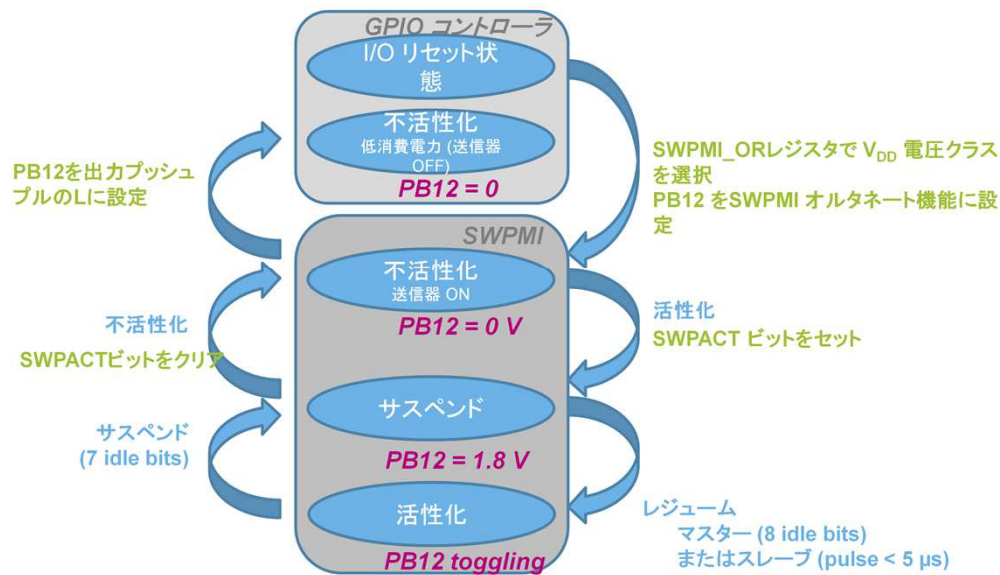


さて、マスタまたはスレーブのいずれかがレジュームシーケンスを送信することにより、通信を開始することができます。マスタによるレジュームシーケンスは、遷移シーケンスと8アイドルビットで構成されます。

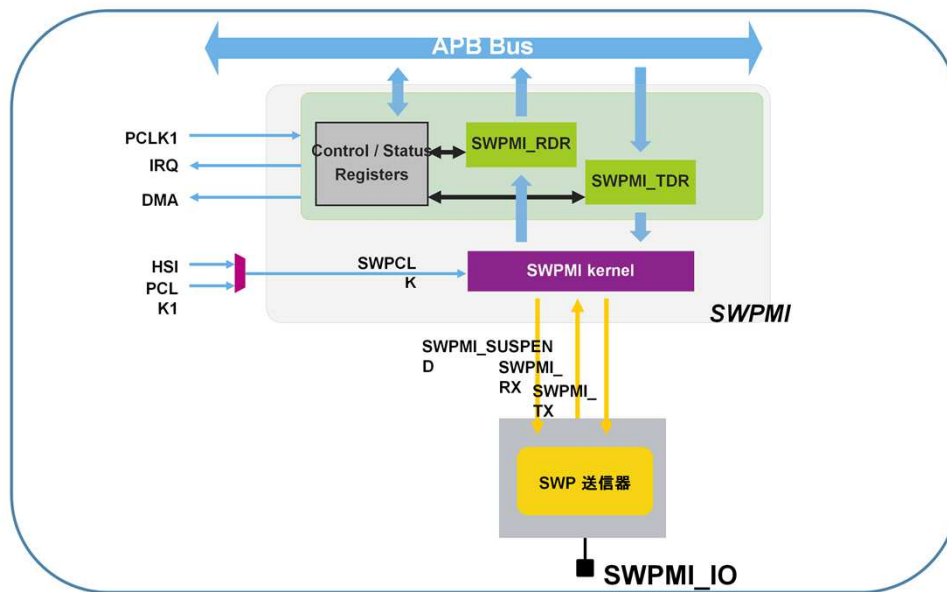
一方、スレーブによるレジュームシーケンスは、マスターが検知するまで電流を引き込むことによりなり、結果として、S1信号がスレーブがデータ送信を開始するためにトグルを開始します。

SWP バスステート

12



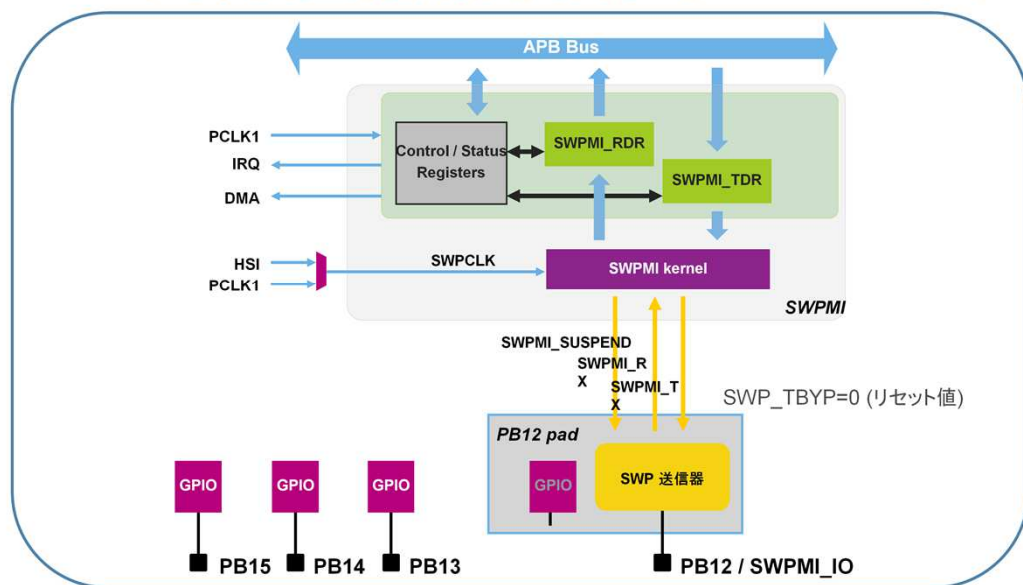
これは、SWPバス状態の、STM32によって管理されている方法の概要です。初期化と活性化の手順の詳細についてはリファレンスマニュアルを参照ことができます。



これはSWPMIペリフェラルのブロック図です。カーネル部分は、内部RC発振器であるHSI、またはAPBバスクロックであるPCLK1の、いずれかによってクロックされています。APBバスとのインターフェースは、SWPMIレジスタへのアクセスがCPUで可能。NVICとDMAへの接続もある。SWPMI_IO信号を介して外部ピンとのインターフェースをする、SWPトランシーバはSTM32に内蔵されています。

内部送信器使用時のブロック図

14



これは内部トランシーバを使用してる、デフォルトの設定です。
SWPMI_IO信号はPB12ピンで提供されています。

15



15

No Software Buffer モード (NSB)

16

- 割り込みまたはポーリングモード
- ソフトウェアの介入は、4 Byteのペイロードデータの送受信 (SWPMI32bitSWPMI_TDR/ SWPMI_RDRデータレジスタの書込み/ 読込み)に必要。



ソフトウェアバッファモード(NSB)で始まる、異なる動作モードを見てみます。このモードでは、データが受信され、ポーリングまたは割り込みモードまたはSWPMIフラグをチェックすることによって送信されます。ソフトウェアの介入は、受信レジスタが一杯になるか、送信データレジスタが空になったときに必要とされます。すなわち、ペイロード内のすべての4つのデータバイトです。

Software Single Buffer モード(SSB)

17

- フレームの送信/受信中に、最後を除いて、ソフトウェアの介入は必要ない
- 送信用のRAM上の32 Byteのフレームバッファ、および受信のために別のものを割り当てる必要がある
- DMA が32bitSWPMI_TDR / SWPMI_RDR レジスタに対して読み書きする
- フレーム長(ペイロードの Byte数)は、RAM上のソフトウェアバッファの最初の Byteに書き込む

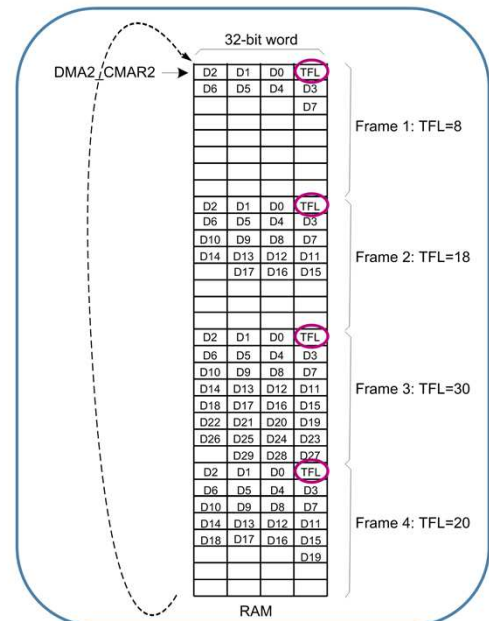


ソフトウェアシングルバッファモード(SSB)は、送信またはソフトウェアの介在なしに全体SWPフレームを受信するために使用されます。フレーム伝送のための32 ByteのソフトバッファがRAMに定義され、SWPMIは自動的にフレームの終わりが受信されるまでは、DMAを通じてSWPMI_TDRレジスタに書き込む。受信のために、32 Byteのソフトバッファは、フレーム受信のためにRAM内に定義され、SWPMI_RDRレジスタのコンテンツは、DMAによってRAMに転送されます。RAMバッファの最初の Byteは、フレームペイロード内の Byte数です。

Software Multi Buffer モード(SMB) / 送信時

18

- ソフトウェアの介入なしに、いくつかのフレームを送信
- 例: RAM上に4つのソフトウェアフレームバッファがある場合
 - サーキュラーモードで設定されたDMAを転送ワード数を32にする
 - バッファの最初の Byteはペイロード数をセット: Transmit Frame Length (TFL).
 - ソフトウェアは、必要に応じて、DMAカウンタと更新フレームバッファを読み出す
 - DMAサーキュラーモードを無効にすることによって、転送は停止されます。場合によっては、RAM内の1つまたは複数のバッファのTFLに0を書き込むことにより停止される(TFL=0はバッファが空であることを意味し、送信がトリガされない)。



最後のモードは、ソフトウェアのマルチバッファモード(SMB)です。このモードは、DMAを使用し、いくつかのSWPフレームは、ソフトウェアの介入なしに扱うことができます。伝送のRAM上の4つフレームバッファの例を見てみる。このモードでは、32 Byteはペイロードサイズに関係なく、常に各フレームのために予約されています。

DMAはサーキュラーモードで設定する必要があり、転送するワード数は32に設定する必要があります。

SSBモードと同様に、各バッファの最初の Byteは、フレームの長さ(これはTFLフィールドと呼ぶ)をセットするために使用されます。ソフトウェアは、DMAカウンタを読み取り、その情報に基づいて各フレームバッファを更新することができます。この例では、3つのフレームは、ソフトウェアの介入なしに送信することができます。送信は、DMAサーキュラーモードを無効にすることによって停止されます。

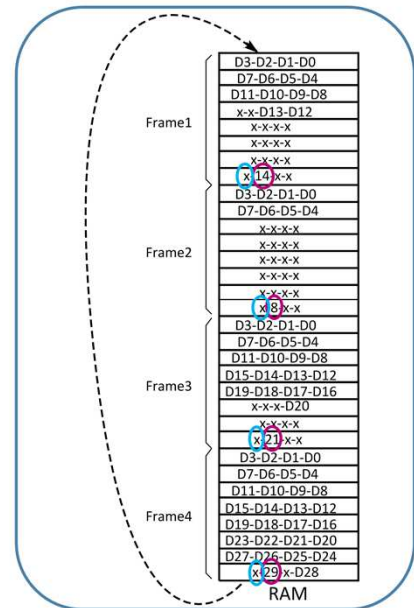
DMAのカウントの終了前に送信を停止する必要がある場合は、TFLフィールドに0をセットする必要があります。この場合、

SWPMIは、もはやDMAリクエストを発行しません。

Software Multi Buffer モード(SMB) / 受信

19

- ソフトウェアの介入なしに、いくつかのフレームを受信
- 例: RAM上に4つのソフトウェアフレームバッファがある場合
 - サーキュラーモードで設定されたDMAを転送ワード数を32にする
 - フレーム長はRAMバッファ内のデータの終わり(31番目、インデックス30)に追加される
 - その他の以下のフラグはフレーム長の次のByte(32番目、インデックス31)にセーブされる:
 - RXBERF (エラー)
 - RXOVRF (オーバーランエラー)
 - RXBFF (バッファレディ)



SMBモードでは、いくつかのフレームは、ソフトウェアの介入なしに受信することができます。RAM上の4つのフレームバッファを受信する例を見てみる。このモードでは、DMAはサーキュラーモードで設定され、転送されるワード数は、32に設定されている必要があります。フレーム長は各ソフトウェアバッファの最後の31番目の Byteにある。各ソフトウェアバッファに格納されたフレームの状態は32番目の Byteにある。これは、エラー、オーバーランエラー、バッファのレディフラグが含まれている。この場合、ソフトウェアは、バッファレディフラグをチェックし、バッファを読んで、32番目の Byteをクリアすることができます。

割込みイベント	説明
Receive buffer full	受信中のフレームの最後のワードがSWPMI_RDRレジスタにある
Transmit buffer empty	送信フレームを送るのにSWPMI_TDRのアップデート必要なし
Receive CRC error	受信フレームでCRCエラーがあった
Receive overrun	ペイロード受信中にオーバーランエラーが発生
Transmit underrun	ペイロード送信中にアンダーランエラーが発生
Receive register not empty	SWPMI_RDRに受信データあり
Transmit register empty	SWPMI_TDRの送信が完了。同レジスタに空きあり
Slave resume	SWPバスがサスペンドのとき、スレーブがレジュームを要求

ここでNVICコントローラ内の割込みをトリガすることができるイベントの概要は次のとおり。

送信および受信バッファ、
送信および受信レジスタ、
エラー（CRC、オーバーランやアンダーラン）、
スレーブによってのレジューム。

割り込みイベント	説明
Transmit request	“Transmit register empty” イベント時、DMA リクエスト発生
Receive request	“Receive register not empty” イベント時、DMA リクエスト発生

- DMAリクエストは、SSBまたはSMB動作モードで動作させるには有効にする必要がある

DMAリクエストは、送受信のためにSWPMIIによって生成されます。SSBおよびSMBモードを使う場合、これらを有効にする必要があります。

Low-power モード

22

モード	説明
Run	アクティブ
Sleep	アクティブ。SWPMIの割込みはデバイスをSleepモードから抜けさせる
Low-power run	アクティブ
Low-power sleep	アクティブ。SWPMIの割込みはデバイスをLow-power Sleepモードから抜けさせる
Stop 1	停止。ペリフェラルレジスタは保持。もしHISがSWPMIのクロックソースのとき、スレーブからのレジューム受信でStop 1モードから抜けさせる
Stop 2	停止。ペリフェラルレジスタは保持
Standby	電源OFF。Standbyモードを抜けるときにペリフェラルの再初期化必要
Shutdown	電源OFF。Shutdownモードを抜けるときにペリフェラルの再初期化必要



すべてのSWPMI割込みがデバイスをスリープモードからウェイクアップすることができます。デバイスはストップモードの場合、スレーブのレジュームイベントによってのみデバイスを起動することができます。

- このペリフェラルにリンクされているこれらのペリフェラルのトレーニングを参照してください
 - Reset and clock control (RCC)
 - SWPMI クロックを有効に
 - SleepモードでのSWPMI クロックを設定
 - SWPMI リセットの制御
 - Nested vectored interrupt controller (NVIC)
 - SWPMI 割込みの設定
 - General-purpose I/Os (GPIO)
 - 内部SWPMI_IOTランシーバを有効にするために、ピンPB12にオルタネート機能を設定



これは、シングルワイヤプロトコルマスタインタフェースに関連するペリフェラルの一覧であり。必要に応じて、詳細については、これらのペリフェラルトレーニングを参照のこと