
AVR107 : 直列型メモリとAVR®のインターフェース

要点

- EEPROMデバイス: Microchip 25xx256 (25AA256/25LC256)
- DataFlashデバイス: Microchip SST25VFxxxx (SST25VF010A,SST25VF020,SST25VF040B,SST25VF080B)
- 1/2/4/8MビットSPI直列フラッシュとATtiny817を用いるSPI DataFlashドライバ
- 256K SPIバス直列EEPROMとATtiny817を用いるSPI EEPROMドライバ
- 完全な直列型メモリ機能支援
- SST25VFxxxxの特徴:
 - 柔軟な消去能力
 - 高速消去とバイト書き込み
 - 自動アドレス増加(AAI:Auto Address Increment)書き込み
- 25xx256の特徴:
 - 64バイト ページ
 - 塊書き込み保護
 - 組み込み書き込み保護

序説

著者: Rupali Honrao, Microchip Technology Inc.

直列インターフェースメモリは消費者、車載、電気通信、医療、工業、PCに関連する市場の広範囲で使われます。主に個人的な選択データや形態設定/構成設定データの格納に使われる直列メモリは今日利用可能な不揮発性メモリ(NVM)の最も柔軟な形式です。他のNVM解決策に比べて直列メモリデバイスは低電力消費だけでなく、少ないピン数、より小さな外囲器、より低い電圧も提供します。

殆どのAVR®マイクロコントローラは25AA256/25LC256 EEPROMのような直列メモリデバイスとSST25VF010A,SST25VF020,SST25VF040B,SST25VF080BのようなDataFlashデバイスとの接続を許すSPIインターフェースを提供します。

AVRデバイスとのSPI直列メモリ統合を楽にすることで促進するため、効率的なアクセスを提供するように基本ドライバが開発されました。この応用記述はこれらのドライバの機能と基本構造を記述します。この応用記述はAtmel | STARTでソースコードも提供します。

本書は一般の方々の便宜のため有志により作成されたもので、Microchip社とは無関係であることを御承知ください。しおりの[はじめに]での内容にご注意ください。

目次

要点	1
序説	1
1. 直列型メモリ	3
1.1. 直列型メモリとAVRのハードウェア接続	3
1.2. SPI直列型メモリのアクセス方式	3
1.3. SPI周辺機能の扱い	7
1.4. 多忙検出	7
2. Atmel STARTからのソースコード取得	7
3. ソースコード概要	7
3.1. SPIインターフェースの初期化	8
3.2. ホーリング操作動作関数	8
3.3. DataFlash読み込みと書き込みの関数	9
3.4. EEPROM読み込みと書き込みの関数	10
4. 開発環境	10
5. 改訂履歴	11
Microchipウェブ サイト	12
お客様への変更通知サービス	12
お客様支援	12
製品識別システム	12
Microchipデバイスコード保護機能	13
法的通知	13
商標	13
DNVによって認証された品質管理システム	14
世界的な販売とサービス	15

1. 直列型メモリ

後続項はAVRへのメモリ接続、メモリアクセス形式、領域書き込み保護ビット、多忙検出を記述します。より多くの情報についてはデバイスのデータシートを参照してください。

1.1. 直列型メモリのAVRへのハードウェア接続

SPIインターフェースは主装置動作に構成設定されます。SCKとMOSIのポートは出力、MISOのポートは入力です。図1-1をご覧ください。

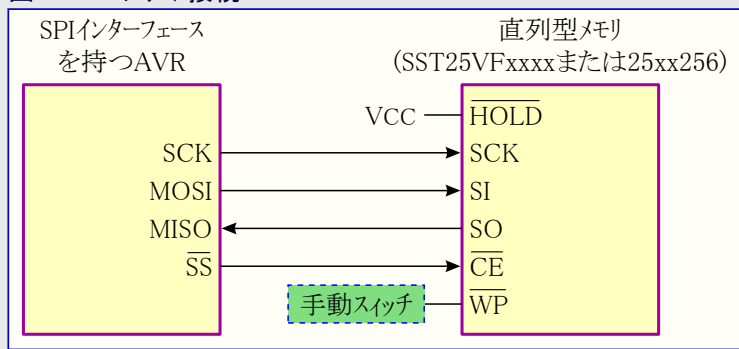
この応用では直列メモリのチップ選択(\overline{CE})ピンを制御するのに従装置選択(\overline{SS})信号が使われることに注意してください。

SPIインターフェースはデータが有効でない場合にクロックを常に停止するため、HOLD信号をその不活性状態のHighに保つことができます。

使用者は書き込み保護(\overline{WP})信号の構成設定を選ぶことができます。

注: EEPROMの動作電圧範囲は1.8~5.5Vで、一方DataFlashデバイスは2.7~3.6Vで動きます。

図1-1. ハードウェア接続



1.2. SPI直列型メモリのアクセス方式

提供されるドライバの基本構造を理解するため、直列メモリアクセスの各種方式を良く調べる必要があります。

全てのアクセスは次の手順に従います。

1. チップ選択線がLowに駆動されます。
2. SCKクロックに同期してデータ線上で幾らかの直列化されたバイトが送られるか、または受け取られます。
3. チップ選択線がHighに駆動されます。

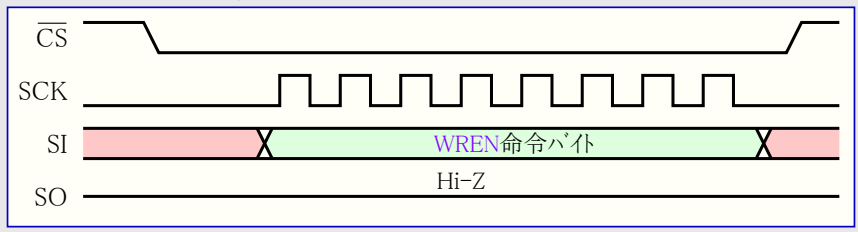
直列化されるバイトの数と値はアクセス形式に依存します。

この応用記述のSPIメモリデバイスはSPI動作種別0('00')とSPI動作種別3('11')の両方を支援します。提唱ドライバでは動作種別0が選ばれます(詳細については直列型メモリデバイスのデータシートをご覧ください)。

1.2.1. 単一バイト書き込み命令: WREN, WRDI, CHIP ERASE

これらのアクセスは1バイト長です。MOSIデータ線に命令バイト(以降は"op_code"と呼ばれます(訳注)本書では不使用)だけが送られます。

図1-2. WRENタイミング例



1.2.2. 状態レジスタ読み/書き: RDSR/WRSR

これらのアクセスは2バイト長です。命令バイトに書かれる(WRSR)または読まれる(RDSR)バイトが続きます。

注: 状態レジスタ書き込み操作はWREN命令が先行しなければなりません。

図1-3. WRSRタイミング

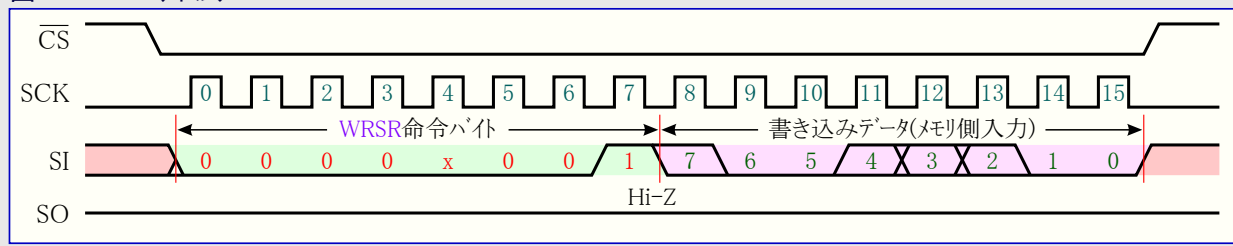
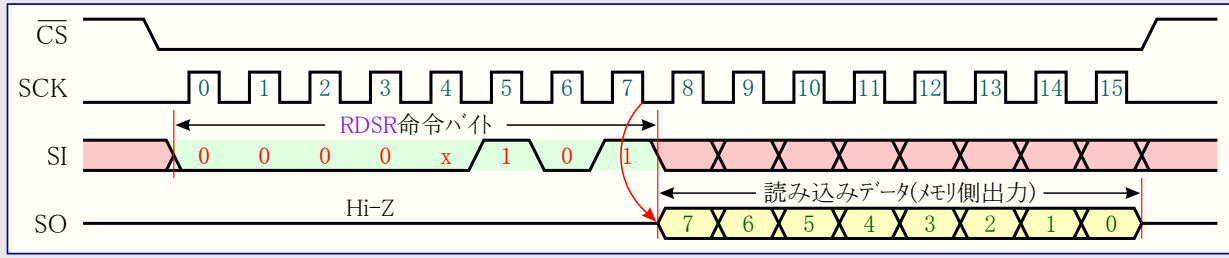


図1-4. RDSRタイミング

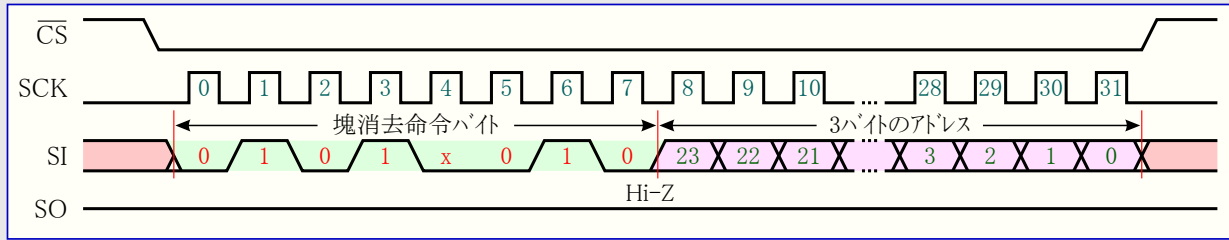


1.2.3. セクタ/塊-消去命令 (SST25VFxxxxデバイスのみ)

このアクセスは1バイトの命令バイトと3つのアドレスバイトから成ります。図1-5は32Kバイト塊消去手順を図解します。32Kバイト塊消去命令は8ビット命令の\$52と後続する3つのアドレスバイトを実行することによって開始されます。

注: どの書き込み命令にも先立って、書き込み許可(WREN)命令が実行されなければなりません。

図1-5. セクタ消去タイミング



1.2.4. データ書き込みとデータ読み込みのアクセス

これらのアクセスは1バイトの命令バイトと後続するアドレス段階とデータ段階から成ります。アドレス段階は25xx256デバイスに対して2バイトのアドレスで、SST25VFxxxxデバイスに対して3バイトのアドレスです。

どのデータ書き込みを試みるのにも先立って、WREN命令を発行することによって書き込み許可ラッチが設定されなければなりません。

1.2.4.1. 25xx256デバイスに対するデータ書き込み

書き込み周回が必要になる前に最大64バイトのデータをデバイスに送ることができます。制限は全てのバイトが同じページに属されなければならないことだけです。ページ書き込み操作は実際に書かれるバイト数に関わらず、単一の物理ページ内でバイトを書くことに制限されます。

図1-6. バイト書き込み手順 (25xx256)

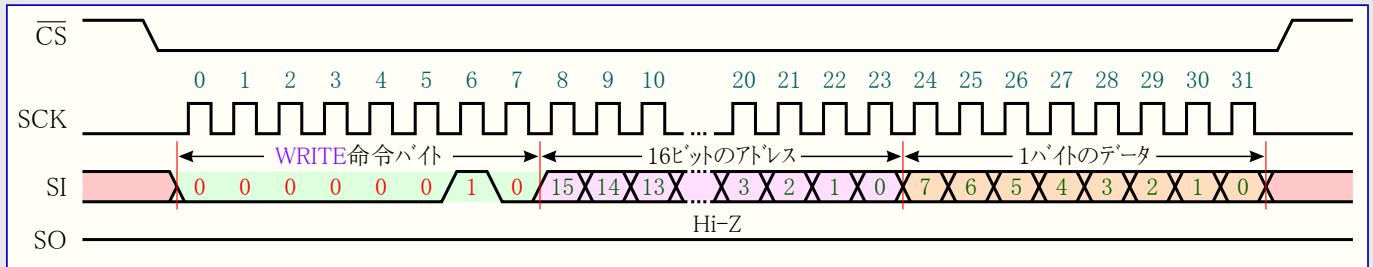
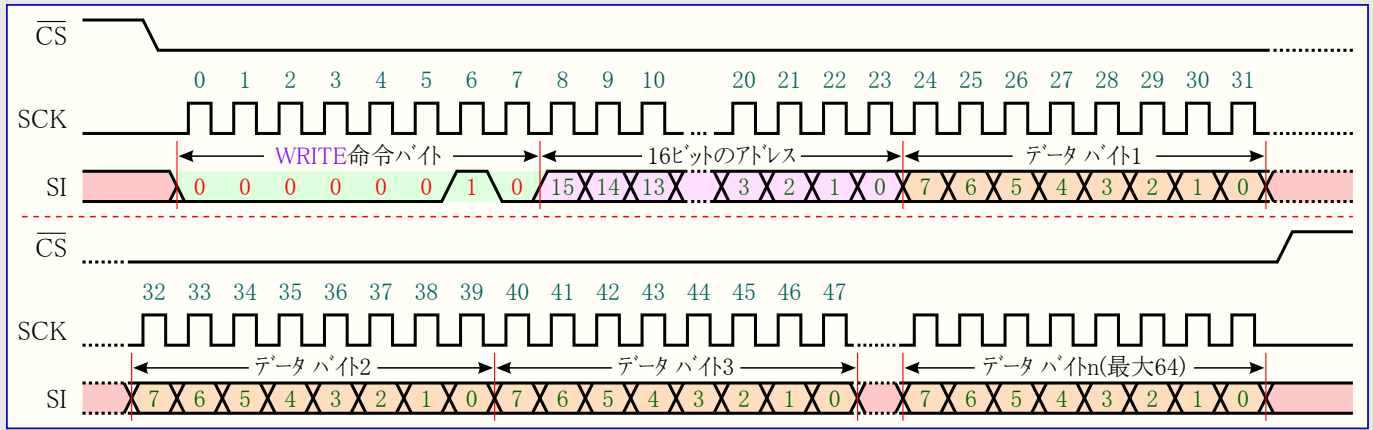


図1-7. ページ書き込み手順 (25xx256)

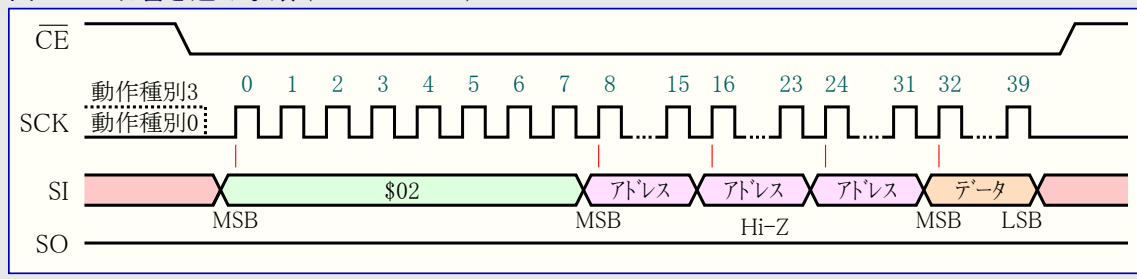


1.2.4.2. SST25VFxxxxデバイスに対するデータ書き込み

バイト書き込み:

バイト書き込み命令は選んだバイトのビットを望むデータに書き込み、[図1-8](#)で図解されます。

図1-8. バイト書き込み手順 (SST25VFxxxx)



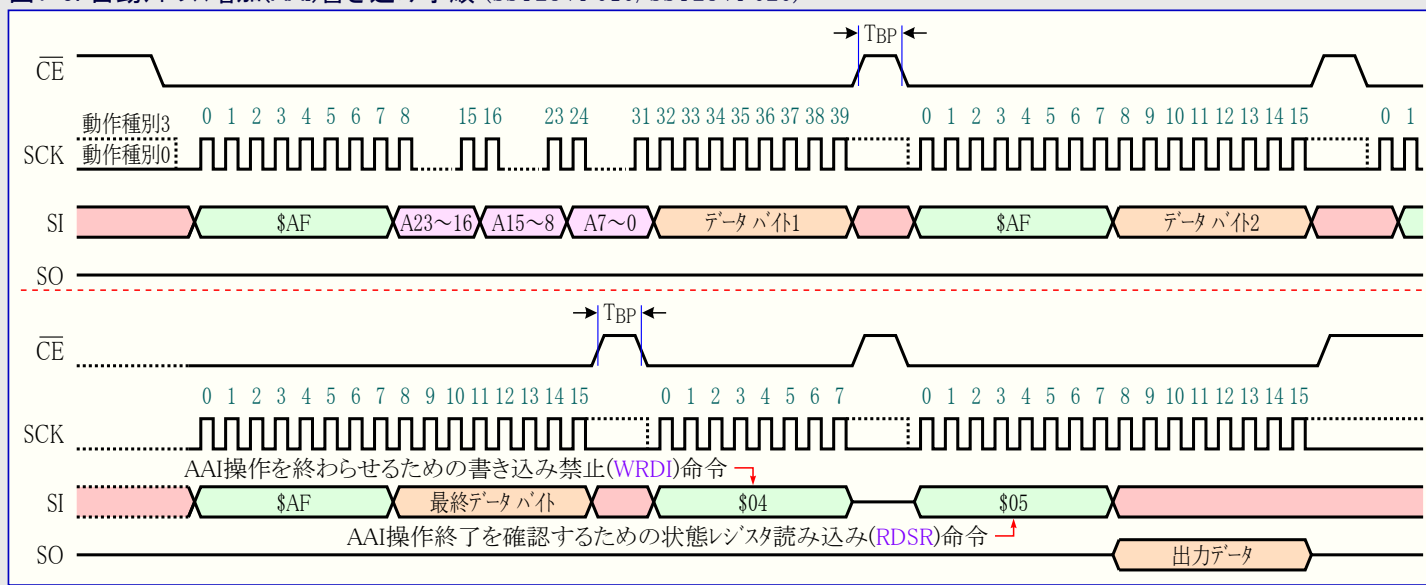
自動アドレス増加(AAI:Auto Address Increment)書き込み:

AAI書き込み命令は次の連続アドレス位置の再発行なしに書かれることを複数バイトのデータに許します。この機能はメモリ配列全体が書かれる時の総書き込み時間を減らします。どの書き込み命令にも先立ち、書き込み許可(WREN)命令が実行されなければなりません。

AAI書き込み命令は8ビット命令の\$AFと後続するアドレスビットのA23~A0を実行することによって開始されます。アドレスに後続して、データがMSB(ビット7)からLSB(ビット0)へ順次入力されます。一旦デバイスがバイト書き込みを完了すると、次に続くアドレスを書き込むことができます。8ビット命令の\$AFと後続する書かれるべきデータを入力してください。望む最後のバイトが書かれてしまった時にAAIを終わるために書き込み禁止(WRDI)命令の\$04を実行してください。WRDI命令の実行後、使用者はデバイスが書き込みを完了したのを確実にするために状態レジスタをポーリングしなければなりません。

AAI書き込み手順については[図1-9](#)を参照してください。これはSST25VF010AとSST25VF020のデバイスに適用できます。

図1-9. 自動アドレス増加(AAI)書き込み手順 (SST25VF010/SST25VF020)



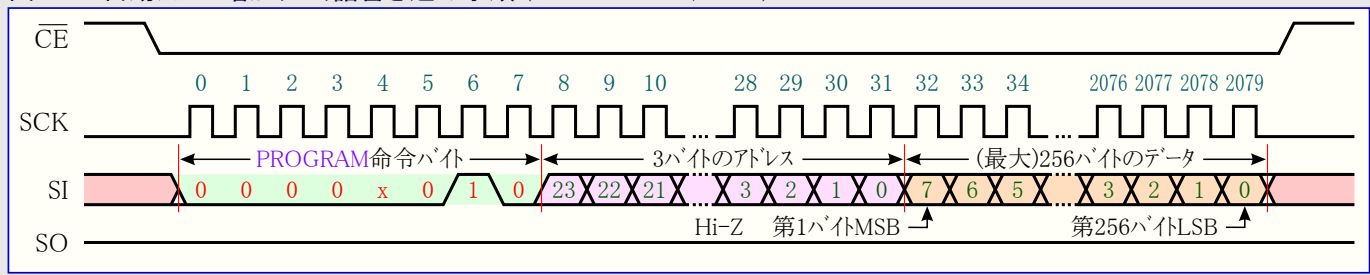
自動アドレス増加(AAI:Auto Address Increment)語書き込み:

自動アドレス増加(AAI)語書き込み手順は以下を除いて上で記述された自動アドレス増加(AAI)書き込み手順と同じです。

- AAI語書き込み用命令符号は\$ADです。
- 2バイトのデータが書かれます。
- ハードウェア書き込み終了検出動作でAAI語書き込みを終わらせる手順は、書き込み禁止(WRDI)命令の\$04と後続する語書き込み中の準備可/多忙状態としてのSO禁止(DBSY)の8ビット命令の\$80を実行してください。

自動アドレス増加(AAI)語書き込み手順については[図1-10](#)を参照してください。これはSST25VF040B/080Bデバイスに適用できます。

図1-10. 自動アドレス増加(AAD)書き込み手順 (SST25VF040B/080B)

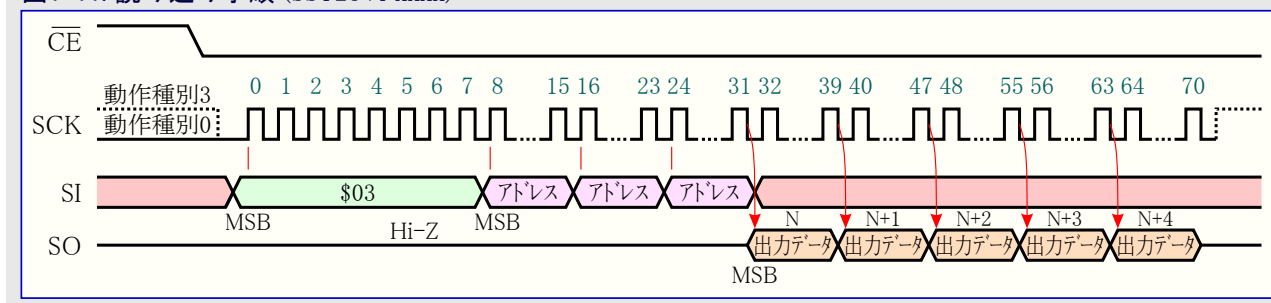


1.2.4.3. データ読み込みアクセス

これらのアクセスは1バイトの命令符号と後続するアドレス段階から成ります。アドレス段階は25xx256デバイスに対して2バイトのアドレス、SST25VFxxxxデバイスに対して3バイトのアドレスです。

デバイスは $\overline{\text{CE}}$ をLowに引くことによって選ばれます。8ビット読み込み(READ)命令と後続するアドレスがデバイスに送出されます。正しいREAD命令とアドレスが送られた後、選ばれたアドレスでメモリに格納されたデータがSOピンで順次出力されます。クロックパルスの提供を続けることによって次のアドレスでメモリに格納されたデータを続けて読むことができます。内部アドレスポインタは各データバイトが順次出力された後により高い次のアドレスへ自動的に増されます。最高アドレス到達時、アドレス計数器は何時までも続く読み込み周回を許すアドレス\$0000へ繰り返します。読み込み操作はCEピンを(Highへ)上げることによって終了されます。

図1-11. 読み込み手順 (SST25VFxxxx)



1.2.5. アクセス形式要約

次表はアクセス形式とそれらに関連する長さを要約します。

表1-1. アクセス形式要約表

命令	命令バイト	アドレス段階長 (バイト)		データ段階長 (バイト)	
		25xx256	SST25VFxxxx	MOSI線	MISO線
WREN	0000 0110 (\$06)	0	0	0	0
WRDI	0000 0100 (\$04)	0	0	0	0
RDSR	0000 0101 (\$05)	0	0	0	1 (注4)
WRSR	0000 0001 (\$01)	0	0	1	0
READ	0000 0011 (\$03)	2	3	0	1~∞ (注4)
WRITE	0000 0010 (\$02)	2	不可 (注1)	1~64	0
BYTE PROGRAM	0000 0010 (\$02)	不可 (注1)	3	1	0
AAI書き込み	1010 1101 (\$AD)	不可 (注1)	3 (注2)	2	0
AAI書き込み	1010 1111 (\$FF)	不可 (注1)	3 (注3)	1	0
4Kバイトセクタ消去	0010 0000 (\$20)	不可 (注1)	3	0	0
32Kバイト塊消去	0101 0010 (\$52)	不可 (注1)	3	0	0
64Kバイト塊消去	1101 1000 (\$D8)	不可 (注1)	3 (注2)	0	0
CHIP ERASE	0110 0000 (\$60) または 1100 0111 (\$C7)	不可 (注1)	0	0	0
RDID	1001 0000 (\$90) または 1010 1011 (\$AB)	不可 (注1)	3	0	1~∞ (注4)

注1: この命令はこのデバイスに対して利用不可です。

注2: この命令はSST25VF040B/080Bデバイスに対して利用可能です。

注3: この命令はSST25VF010A/020デバイスに対して利用可能です。

注4: SST25VFxxxxデバイスに対するデータ段階長は1~∞です。データはSST25VFxxxxデバイスに対してCEでのLowからHighへの遷移によって終了されるまで進行中のクロック周回で継続的に読まれます。

1.3. SPI周辺機能の扱い

直列メモリへのアクセスを実行するにはSPIインターフェース(同期直列通信インターフェース)が使われます。

各SPIクロック周期中、主装置がMOSI線でビットを送って従装置がそれを読み、同時に従装置がMISO線でビットを送って主装置がそれを読みます。この手順は一方方向データ転送が意図されるだけの時でも維持されます。

SPI操作はバイト単位です。バイトが送受信された後、**INTFLAGS**レジスタ内で**SPIF**が設定(1)されます。

SPI周辺機能を扱うための2つの操作方法が想像できます。

1. **SPIF**が設定(1)されたかを検知するために**INTFLAGS**をポーリングし、そして次のSPI転送を実行します。
2. SPI割り込みを許可し、SPI割り込み処理で次のSPI転送を管理します。

注: SPI周辺機能操作のポーリング法は提供されるソースコードで使われます。

1.4. 多忙検出

直列メモリ25xx256デバイスはその状態レジスタ内の**WIP**(書き込み中)ビットでその現在の状態を示します。

直列メモリSST25VFxxxxデバイスは状態レジスタの**BUSY**(多忙)ビットで多忙状態を示します。このビットが'1'に設定される時は書き込みが進行中で、'0'に設定される時は書き込みが進行中ではありません。このビットは読み込み専用です。

2. Atmel | STARTからのソースコード取得

コード例は画像使用者インターフェース(GUI)を通して応用コードの構成設定を許すウェブに基づくAtmel | STARTを通して利用可能です。コードは下の直接コード例リンクまたはAtmel | START先頭頁の**BROWSE EXAMPLES**(例検索)鉤経由Atmel Studio 7.0とIAR Embedded Workbench®の両方に対してダウンロードすることができます。

Atmel | STARTウェブ ページ : <http://start.atmel.com/>

コード例

- SPI DataFlash:
– http://start.atmel.com/#example/Atmel:spi_serial_memory:1.0.0::Application:SPI_DataFlash:
- SPI EEPROM:
– http://start.atmel.com/#example/Atmel:spi_serial_memory:1.0.0::Application:SPI_EEPROM:

例プロジェクトについての詳細と情報に関してはAtmel | STARTで**User guide**(使用者の手引き)を押下してください。**User guide**鉤はAtmel | STARTプロジェクト構成設定部内の一覧画面でプロジェクト名をクリックすることにより、例閲覧部で見つけることができます。

Atmel Studio

DOWNLOAD SELECTED EXAMPLE(選んだ例をダウンロード)をクリックすることにより、Atmel | STARTで例閲覧部からAtmel Studio用**.atzip**ファイルとしてコードをダウンロードしてください。Atmel | START内からファイルをダウンロードするには、**EXPORT PROJECT**(プロジェクトをエクスポート)に続いて**DOWNLOAD PACK**(一括ダウンロード)をクリックしてください。

ダウンロードした**.atzip**ファイルをダブルクリックしてください。プロジェクトがAtmel Studio 7.0に導入されます。

IAR Embedded Workbench

IAR Embedded Workbenchでプロジェクトをインポートする方法の情報についてはAtmel | START使用者の手引きを開き、**Using Atmel Start Output in External Tools**(外部ツールでAtmel START出力を使用)と**IAR Embedded Workbench**を選んでください。Atmel | START使用者の手引きへのリンクは共に頁の右上隅に置かれたAtmel | START先頭頁から**About**(これについて)またはプロジェクト構成設定部内の**Help And Support**(手助けと支援)をクリックすることによって見つけることができます。

3. ソースコード概要

以降の項は周辺機能、SPIインターフェースの初期化、ポーリング動作関数とDataFlash/EEPROM読み書き関数を記述します。

周辺機能:

- CPUクロック : (既定) 3.33MHz
- 使う周辺機能:
– SPI
• PA1 : MOSI • PA2 : MISO • PA3 : SCK • PA4 : SS

Atmel | STARTで構成設定したプロジェクトは周辺機能ドライバとファイルだけでなく全てのドライバを初期化する**main()**関数も生成します。

- 周辺機能ドライバのヘッダとソースのファイルは**src**と**include**のフォルダ内です。
- **atmel_start.c**ファイルで**atmel_start_init()**関数はプロジェクト内のMCU、周辺機能ドライバ、ミドルウェアを初期化します。
- DataFlashドライバ ファイル : **dataflash.c** , **dataflash.h**
- EEPROMドライバ ファイル : **eprom.c** , **eprom.h**

3.1. SPIインターフェースの初期化

`dataflash_int()`関数はDataFlashドライバでSPIを初期化します。

`eeeprom_int()`関数はEEPROMドライバでSPIを初期化します。

- ポート方向が次のように構成設定されます。
 - 出力：PA3/SCK、PA1/MOSI、PA4/SS
 - 入力：PA2/MISO
- SPIインターフェースは制御A(**CTRLA**)レジスタを構成設定することによって主装置として構成設定されます。
- SPIクロック速度については使われるクロック前置分周器が**DIV4**で**CTRLA**レジスタのクロック倍速(**CLK2X**)を用いて倍にされます。
- SPI転送動作は動作種別0として選ばれ、従装置選択は制御B(**CTRLB**)レジスタの従装置選択禁止(**SSD**)ビットを'1'に構成設定することによって禁止されます。
- 従装置選択線はHighに構成設定されます。

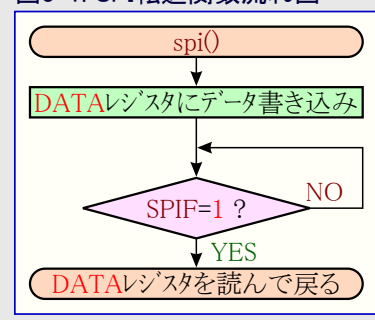
3.2. ホールディング操作動作関数

SPI転送基本関数

`spi()`関数はホールディング動作でSPIインターフェースを扱う基本関数です。これはバイトを送出して受け取ったものを格納します。この関数がチップ選択線のレベルを制御しないことに注意してください。`spi()`は`tx_spi()`と`rx_spi()`から呼ばれます。`tx_spi()`関数はSPIでバイトを書き、`rx_spi()`がバイトを受け取ります。

原型: `char spi(uint8_t data);`

図3-1. SPI転送関数流れ図

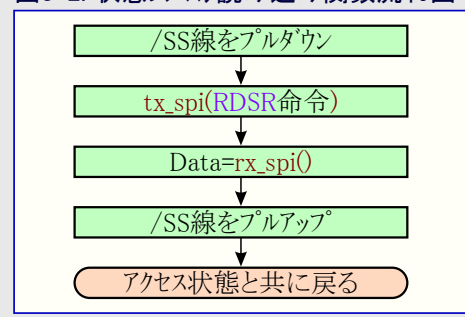


状態レジスタ読み込み関数

`dataflash_read_status_register()`と`eeeprom_read_status_register()`の関数はRDSRアクセスを実行します。これらはアクセス状態符号を返します。

原型: `uint8_t dataflash_read_status_register(void);`
`uint8_t eeeprom_read_status_register(void);`

図3-2. 状態レジスタ読み込み関数流れ図



単一バイト書き込み命令

`send_cmd()`関数は**WREN**、**WRDI**のような単一バイト命令を送ります。直列メモリはアクセスを実行する前に準備可でなければなりません。

原型: `void send_cmd(uint8_t opcode);`

図3-3. 命令送出流れ図

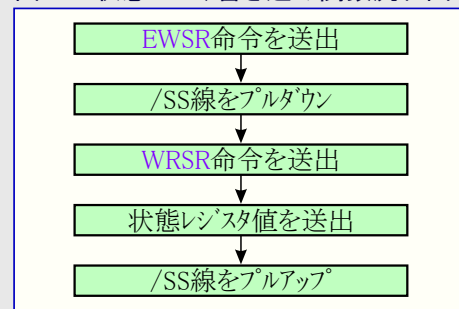


状態レジスタ書き込み関数

この関数はRDSRアクセスを実行します。これはEWSR(状態レジスタ書き込み許可)とWRSR(状態レジスタ書き込み)の命令を送り、その後に状態レジスタに書くデータを送ります。

原型: `void dataflash_write_status_register(uint8_t byte);`

図3-4. 状態レジスタ書き込み関数流れ図



3.3. DataFlash読み込みと書き込みの関数

DataFlash関数は例ソースコードに於いてdataflash.cとdataflash.hで実装、定義されます。

DataFlash単一バイト読み込み:

dataflash_read_byte()関数はDataFlashで与えられたアドレス位置の1バイトを読みます。

原型: `uint8_t dataflash_read_byte(uint32_t addr);`

DataFlash複数バイト読み込み:

複数バイト読み込みは以下の3つの関数を用いて実行されます。これらの関数は例ソースコードのmain.cでread_dataflash()から呼びべます。

原型: `void dataflash_read_multiple_start(uint32_t addr);`
`void dataflash_read_multiple_continue(uint8_t *buff, uint8_t buff_length);`
`void dataflash_read_multiple_stop();`

dataflash_read_multiple_start()は/SS線をプルダウンし、READ命令を送って開始アドレスを送ります。

dataflash_read_multiple_continue()は読んだバイト数が与えられたbuff_lengthと等しくなるまでデータを続けて読みます。データは与えられた緩衝部に格納されます。

dataflash_read_multiple_stop()は読み込み手順を停止するために/SS線をプルアップします。

DataFlash単一バイト書き込み:

dataflash_program_byte()関数は与えられたアドレスに単一バイトを書き込みます。

原型: `void dataflash_program_byte(uint32_t addr, uint8_t byte);`

DataFlash複数バイト書き込み:

複数バイト書き込みは以下の3つの関数を用いて実行されます。これらの関数は例ソースコードのmain.cでwrite_to_dataflash()から呼びべれます。

原型: `void dataflash_program_multiple_start(uint32_t addr);`
`uint8_t dataflash_program_multiple_continue(uint8_t *buff, uint8_t length);`
`void dataflash_program_multiple_stop();`

複数バイト書き込みは次に続くアドレス位置の再発行なしに複数バイトのデータ書き込みを許すAAI書き込み命令を使います。

dataflash_program_multiple_start()はAAI_program_start()関数を呼びべます。これはEBSYとWRENの命令を送り、/SS線をプルダウンしてAAL_WORD_PROG命令を送ります。

dataflash_program_multiple_continue()はデバイス選択に応じて語またはバイトを書くためにAAI_program_continue()関数を呼びべます。SS T25VF040B/SST25VF080Bデバイスに対しては語が書かれ、SST25VF010A/SST25VF020デバイスに対してはバイトが書かれます。AAI_program_continue()関数は書いたバイト数が与えられた長さと同しくなるまで呼びべれます。最後のバイトが書かれると、関数は0を返します。

dataflash_program_multiple_stop()はAAI_program_stop()関数を呼びべます。これは停止手順、WRDIとDBSYの命令を送ります。

注: 命令の記述についてはデバイスのデータシートを参照してください。

DataFlashのデバイスIDと製造者ID読み込み関数:

原型: `uint8_t dataflash_read_id(uint32_t addr);` /* デバイスIDを読むにはアドレスADD_DEV_ID : 0x01を使ってください。*/
`uint32_t dataflash_jedec_id_read();` /* アドレス0x00から製造者IDを読みます。*/

DataFlash消去関数:

```

原型: void dataflash_erase_chip(void);           /* チップ全体消去 */
      void dataflash_erase_sector_4k(uint32_t addr); /* 4Kセクタ消去 */
      void dataflash_erase_block_32k(uint32_t addr); /* 32Kセクタ消去 */
      void dataflash_erase_block_64k(uint32_t addr); /* 64Kセクタ消去 */

```

3.4. EEPROM読み込みと書き込みの関数

EEPROM読み書き関数は例ソースコードのSPI EEPROMに於いてeeprom.cとeeprom.hで実装、定義されます。

EEPROM単一バイト読み込み:

eeprom_read_byte()関数はEEPROMで与えられたアドレス位置の1バイトを読みます。

```
原型: uint8_t eeprom_read_byte(uint16_t addr);
```

EEPROM複数バイト読み込み:

eeprom_program_multiple_read()関数は与えられたアドレスからbuff_lengthまでデータを読んで与えられた緩衝部に格納します。

```
原型: void eeprom_program_multiple_read(uint16_t addr, uint8_t *buff, uint8_t buff_length);
```

この関数での手順は/SS線をプルダウンしてREAD命令を送って開始アドレスを送ります。その後与えられたbuff_lengthと等しい読み込みバイト数までデータを続けて読みます。データが与えられた緩衝部に格納され、その後読み込み処理を停止するために/SS線がプルアップされます。

EEPROM単一バイト書き込み:

eeprom_program_byte()関数は与えられたアドレスに単一バイトを書き込みます。

```
原型: void eeprom_program_byte(uint16_t addr, uint8_t byte);
```

EEPROM複数バイト書き込み:

この関数はbuff_lengthまで与えられた配列から複数バイトを書き込みます。与えられた開始アドレスから書き込みを開始します。この関数ではアドレスとbuff_lengthの検証が行われます。開始アドレスはページアドレスの先頭であるべきで、buff_lengthは限度内であるべきです。検証失敗の場合、関数は何も書かずに戻ります。

アドレスと緩衝部の長さが有効なら、書かれるのに必要な総ページ数が計算され、64バイトまでのページ容量(PAGE_SIZE)にデータを書くeeprom_page_write()関数が呼ばれます。

```

原型: void eeprom_program_multiple_write(uint16_t addr, uint8_t *buff, uint8_t buff_length);
      void eeprom_page_write(uint16_t addr, uint8_t *buff, uint8_t buff_length);

```

EEPROM消去関数:

```

原型: void eeprom_erase_chip();           /* チップ全体消去 */
      void eeprom_erase_page(uint16_t addr); /* ページ消去、アドレスはページアドレスの先頭であるべきです。 */

```

4. 開発環境

付随するドライバコードの開発過程で使われたハードウェアとソフトウェアは次のとおりです。

ハードウェア

- ATtiny817 Xplained Pro基板
- ブレッドボードでの8リードSOICソケット
- SOIC外囲器で半田付けされたAT25AA256/AT25LC256
- SOIC外囲器で半田付けされたSST25VF010A
- SOIC外囲器で半田付けされたSST25VF020
- SOIC外囲器で半田付けされたSST25VF040B
- SOIC外囲器で半田付けされたSST25VF080B

ソフトウェア

- Atmel Studio 7

5. 改訂履歴

資料改訂	日付	注釈
2595A	2005年3月	初版資料公開
2595B	2016年7月	新雛形
2595C	2017年11月	AT25xx256/SST25VFxxxxの直列メモリを用いて更新
DS00002665A	2018年3月	この資料はMicrochipのDS番号を使ってAtmel 2595Cを置き換え

Microchipデバイスコード保護機能

Microchipデバイスでの以下のコード保護機能の詳細に注意してください。

- Microchip製品はそれら特定のMicrochipデータシートに含まれる仕様に合致します。
- Microchipは意図した方法と通常条件下で使われる時に、その製品システムが今日の市場でその種類の最も安全なシステムの1つであると考えます。
- コード保護機能を破るのに使われる不正でおそらく違法な方法があります。当社の知る限りこれらの方法の全てはMicrochipのデータシートに含まれた動作仕様外の方法でMicrochip製品を使うことが必要です。おそらく、それを行う人は知的財産の窃盗に関与しています。
- Microchipはそれらのコードの完全性について心配されているお客様と共に働きたいと思います。
- Microchipや他のどの半導体製造業者もそれらのコードの安全を保証することはできません。コード保護は当社が製品を”破ることができない”として保証すると言うことを意味しません。

コード保護は常に進化しています。Microchipは当社製品のコード保護機能を継続的に改善することを約束します。Microchipのコード保護機能を破る試みはデジタルミレニアム著作権法に違反するかもしれません。そのような行為があなたのソフトウェアや他の著作物に不正なアクセスを許す場合、その法律下の救済のために訴権を持つかもしれません。

法的通知

デバイス応用などに関してこの刊行物に含まれる情報は皆さまの便宜のためにだけ提供され、更新によって取り換えられるかもしれません。皆さまの応用が皆さまの仕様に合致するのを保証するのは皆さまの責任です。Microchipはその条件、品質、性能、商品性、目的適合性を含め、明示的にも黙示的にもその情報に関連して書面または表記された書面または黙示の如何なる表明や保証もしません。Microchipはこの情報とそれの使用から生じる全責任を否認します。生命維持や安全応用でのMicrochipデバイスの使用は完全に購入者の危険性で、購入者はそのような使用に起因する全ての損害、請求、訴訟、費用からMicrochipを擁護し、補償し、免責にすることに同意します。他に言及されない限り、Microchipのどの知的財産権下でも暗黙的または違う方法で許認可は譲渡されません。

商標

Microchipの名前とロゴ、Mchip、AnyRate、AVR、AVR、AVR Freaks、BeaconThings、BitCloud、CryptoMemory、CryptoRF、dsPIC、FlashFlex、flexPWR、Heldo、JukeBlox、KeeLoq、KeeLoq、Kleer、LANCheck、LINK MD、maXStylus、maXTouch、MediaLB、megaAVR、MOST、MOST、MPLAB、OptoLyzer、PIC、picoPower、PICSTART、PIC32、Prochip Designer、QTouch、RightTouch、SAM-BA、SpyNIC、SST、SST、SuperFlash、tinyAVR、UNI/O、XMEGAは米国と他の国に於けるMicrochip Technology Incorporatedの登録商標です。

ClockWorks、The Embedded Control Solutions Company、EtherSynch、Hyper Speed Control、HyperLight Load、IntelliMOS、mTouch、Precision Edge、Quiet-Wireは米国に於けるMicrochip Technology Incorporatedの登録商標です。

Adjacent Key Suppression、AKS、Analog-for-the-Digital Age、Any Capacitor、AnyIn、AnyOut、BodyCom、chipKIT、chipKIT、CodeGuard、CryptoAuthentication、CryptoCompanion、CryptoController、dsPICDEM、dsPICDEM.net、Dynamic Average Matching、DAM、ECAN、EtherGREEN、In-Circuit Serial Programming、ICSP、Inter-Chip Connectivity、JitterBlocker、KleerNet、KleerNet、Mindi、MiWi、motorBench、MPASM、MPF、MPLAB Certified、MPLAB、MPLINK、MultiTRAK、NetDetach、Omniscient Code Generation、PICDEM、PICDEM.net、PICkit、PICtail、PureSilicon、QMatrix、RightTouch、REAL ICE、Ripple Blocker、SAM-ICE、Serial Quad I/O、SMART-I.S.、SQI、SuperSwitcher、SuperSwitcher II、Total Endurance、TSHARC、USBCheck、VariSense、View Sense、WiperLock、Wireless DNA、ZENAは米国と他の国に於けるMicrochip Technology Incorporatedの商標です。

SQTPは米国に於けるMicrochip Technology Incorporatedの役務標章です。

Silicon Storage Technologyは他の国に於けるMicrochip Technology Inc.の登録商標です。

GestICは他の国に於けるMicrochip Technology Inc.の子会社であるMicrochip Technology Germany II GmbH & Co. KGの登録商標です。

ここで言及した以外の全ての商標はそれら各々の会社の所有物です。

© 2018年、Microchip Technology Incorporated、米国印刷、不許複製

AMBA、Arm、Arm7、Arm7TDMI、Arm9、Arm11、Artisan、big.LITTLE、Cordio、CoreLink、CoreSight、Cortex、DesignStart、DynamIQ、Jazelle、Keil、Mali、Mbed、Mbed Enabled、NEON、POP、RealView、SecurCore、Socrates、Thumb、TrustZone、ULINK、ULINK 2、ULINK-ME、ULINK-PLUS、ULINKpro、µVision、Versatileは米国や他国でのArm Limited(またはその子会社)の商標または登録商標です。

DNVによって認証された品質管理システム

ISO/TS 16949

Microchipはその世界的な本社、アリゾナ州のチャンドラーとテンペ、オレゴン州グラシャムでの設計とウェハー製造設備とカリフォルニアとインドの設計センターに対してISO/TS-16949:2009認証を取得しました。当社の品質システムの処理と手続きはPIC[®] MCUとdsPIC[®] DSC、KEELOQ符号飛び回りデバイス、直列EEPROM、マイクロ周辺機能、不揮発性メモリ、アナログ製品用です。加えて、開発システムの設計と製造のためのMicrochipの品質システムはISO 9001:2000認証取得です。

日本語© HERO 2021.

本応用記述はMicrochipのAN2665応用記述(DS00002665A-2018年3月)の翻訳日本語版です。日本語では不自然となる重複する形容表現は省略されている場合があります。日本語では難解となる表現は大幅に意識されている部分もあります。必要に応じて一部加筆されています。頁割の変更により、原本より頁数が少なくなっています。

必要と思われる部分には()内に英語表記や略称などを残す形で表記しています。

青字の部分はリンクとなっています。一般的に赤字の0,1は論理0,1を表します。その他の赤字は重要な部分を表します。

世界的な販売とサービス

米国	亜細亜/太平洋	亜細亜/太平洋	欧州
本社 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-792-7200 Fax: 480-792-7277 技術支援: http://www.microchip.com/support ウェブアドレス: www.microchip.com アトランタ Duluth, GA Tel: 678-957-9614 Fax: 678-957-1455 オースチン TX Tel: 512-257-3370 ホストン Westborough, MA Tel: 774-760-0087 Fax: 774-760-0088 シカゴ Itasca, IL Tel: 630-285-0071 Fax: 630-285-0075 ダラス Addison, TX Tel: 972-818-7423 Fax: 972-818-2924 デトロイト Novi, MI Tel: 248-848-4000 ヒューストン TX Tel: 281-894-5983 インディアナポリス Noblesville, IN Tel: 317-773-8323 Fax: 317-773-5453 Tel: 317-536-2380 ロサンゼルス Mission Viejo, CA Tel: 949-462-9523 Fax: 949-462-9608 Tel: 951-273-7800 ローリー NC Tel: 919-844-7510 ニューヨーク NY Tel: 631-435-6000 サンホセ CA Tel: 408-735-9110 Tel: 408-436-4270 カナダ - トロント Tel: 905-695-1980 Fax: 905-695-2078	オーストラリア - シドニー Tel: 61-2-9868-6733 中国 - 北京 Tel: 86-10-8569-7000 中国 - 成都 Tel: 86-28-8665-5511 中国 - 重慶 Tel: 86-23-8980-9588 中国 - 東莞 Tel: 86-769-8702-9880 中国 - 広州 Tel: 86-20-8755-8029 中国 - 杭州 Tel: 86-571-8792-8115 中国 - 香港特別行政区 Tel: 852-2943-5100 中国 - 南京 Tel: 86-25-8473-2460 中国 - 青島 Tel: 86-532-8502-7355 中国 - 上海 Tel: 86-21-3326-8000 中国 - 瀋陽 Tel: 86-24-2334-2829 中国 - 深圳 Tel: 86-755-8864-2200 中国 - 蘇州 Tel: 86-186-6233-1526 中国 - 武漢 Tel: 86-27-5980-5300 中国 - 西安 Tel: 86-29-8833-7252 中国 - 廈門 Tel: 86-592-2388138 中国 - 珠海 Tel: 86-756-3210040	インド - ハンガロール Tel: 91-80-3090-4444 インド - ニューデリー Tel: 91-11-4160-8631 インド - フネー Tel: 91-20-4121-0141 日本 - 大阪 Tel: 81-6-6152-7160 日本 - 東京 Tel: 81-3-6880-3770 韓国 - 大邱 Tel: 82-53-744-4301 韓国 - ソウル Tel: 82-2-554-7200 マレーシア - クアラルンプール Tel: 60-3-7651-7906 マレーシア - ペナン Tel: 60-4-227-8870 フィリピン - マニラ Tel: 63-2-634-9065 シンガポール Tel: 65-6334-8870 台湾 - 新竹 Tel: 886-3-577-8366 台湾 - 高雄 Tel: 886-7-213-7830 台湾 - 台北 Tel: 886-2-2508-8600 タイ - バンコク Tel: 66-2-694-1351 ベトナム - ホーチミン Tel: 84-28-5448-2100	オーストラリア - ウェルズ Tel: 43-7242-2244-39 Fax: 43-7242-2244-393 デンマーク - コペンハーゲン Tel: 45-4450-2828 Fax: 45-4485-2829 フィンランド - エスポー Tel: 358-9-4520-820 フランス - パリ Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79 ドイツ - ガルピング Tel: 49-8931-9700 ドイツ - ハーン Tel: 49-2129-3766400 ドイツ - ハイムブロン Tel: 49-7131-67-3636 ドイツ - カールスルーエ Tel: 49-721-625370 ドイツ - ミュンヘン Tel: 49-89-627-144-0 Fax: 49-89-627-144-44 ドイツ - ローゼンハイム Tel: 49-8031-354-560 イスラエル - ラーナナ Tel: 972-9-744-7705 イタリア - ミラノ Tel: 39-0331-742611 Fax: 39-0331-466781 イタリア - ハドバ Tel: 39-049-7625286 オランダ - デルネン Tel: 31-416-690399 Fax: 31-416-690340 ノルウェー - トロンハイム Tel: 47-7289-7561 ポーランド - ワルシャワ Tel: 48-22-3325737 ルーマニア - ブカレスト Tel: 40-21-407-87-50 スペイン - マドリッド Tel: 34-91-708-08-90 Fax: 34-91-708-08-91 スウェーデン - イェテボリ Tel: 46-31-704-60-40 スウェーデン - ストックホルム Tel: 46-8-5090-4654 イギリス - ウォーキングム Tel: 44-118-921-5800 Fax: 44-118-921-5820