

---

---

## 直列周辺インターフェース(SPI)での開始に際して

---

---

### 序説

著者: Alin Stoicescu, Microchip Technology Inc.

この技術概説はtinyAVR<sup>®</sup> 0と1系、megaAVR<sup>®</sup> 0系、AVR<sup>®</sup> DAデバイスの直列周辺インターフェース(SPI:Serial Peripheral Interface)についての情報を提供し、使用者をAVRマイクロコントローラに慣れさせることを意図します。この資料はSPIの応用領域、動作形態、それとハードウェアとソフトウェアの要件を記述します。

資料を通して、周辺機能の構成設定は、SPIピンの位置、ピンの方向、主装置と従装置としてのデバイス初期化方法、システム内部でのデータ交換方法を始め詳細に記述されます。この資料は以下の使用事例を網羅します。

- **主装置SPIとしてデータ送信:**

主装置として構成設定されたデバイスは従装置を制御し、ポーリングと呼ばれる方法を使ってデータを送ります。

- **従装置SPIとしてデータ受信:**

デバイスは従装置として構成設定され、やって来るデータを待ちます。データ受信は割り込みによって起動されます。

- **データ転送形式変更:**

デバイスは主装置として構成設定され、クロック極性とクロック位相と関連してデータを送ります。

**注:** この文書で記述される各使用事例に対して2つのコード例があります。1つはATmega4809で素の状態が開発され、1つはAVR128DA48で開発されたMPLAB<sup>®</sup>コード構成部(MCC)で生成されました。



GitHubでATmega4809コード例を見てください。

貯蔵庫を閲覧するにはクリックしてください。



GitHubでAVR128DA48コード例を見てください。

貯蔵庫を閲覧するにはクリックしてください。

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

序説	1
1. 関連デバイス	3
1.1. tinyAVR® 0系統	4
1.2. tinyAVR® 1系統	4
1.3. megaAVR® 0系統	4
1.4. AVR® DA系概要	4
2. 概要	5
3. SPI主装置としてデータ送信	6
4. SPI従装置としてデータ受信	8
5. データ転送形式変更	10
6. 参考文献	11
7. 追補	12
8. 改訂履歴	14
Microchipウェブサイト	15
製品変更通知サービス	15
お客様支援	15
Microchipデバイスコード保護機能	15
法的通知	15
商標	16
品質管理システム	16
世界的な販売とサービス	17

## 1. 関連デバイス

本章はこの文書に関連するデバイスを一覧にします。以下の図はピン数の変種とメモリ量を展開して各種システムデバイスを示します。

- これらのデバイスが完全にピン互換で同じまたはより多くの機能を提供するため、垂直上方向移植はコード変更なしで可能です。tinyAVR<sup>®</sup> 1系統デバイスでの下方向移植はいくつかの周辺機能のより少ない利用可能な実体のため、コード変更を必要とするかもしれません。
- 左への水平方向移植はピン数、従って利用可能な機能を減らします。
- 異なるフラッシュメモリ量を持つデバイスはまた一般的に異なるSRAMとEEPROMの量を持ちます。

図1-1. tinyAVR<sup>®</sup> 0系統概要

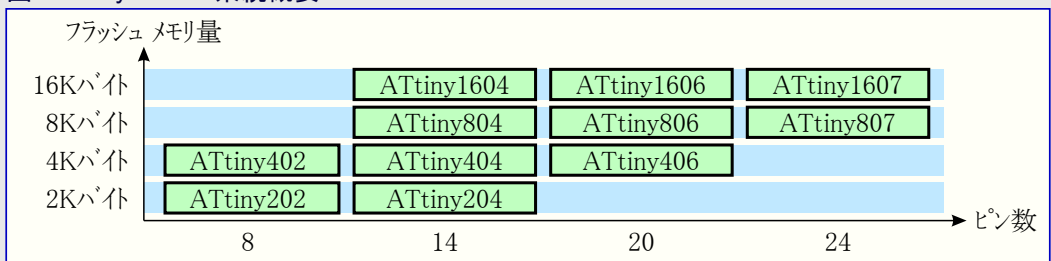


図1-2. tinyAVR<sup>®</sup> 1系統概要

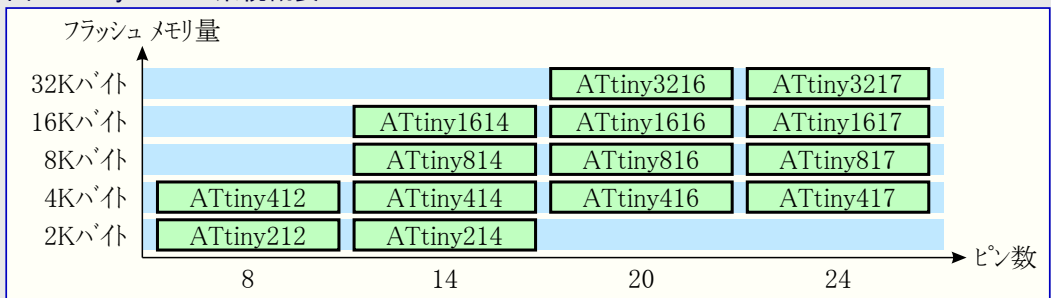


図1-3. megaAVR<sup>®</sup> 0系統概要

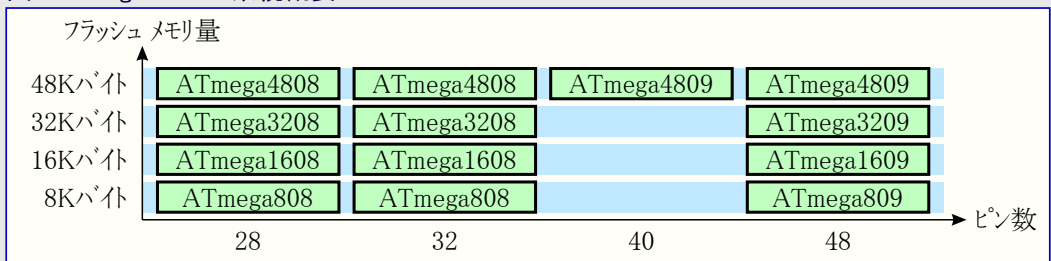
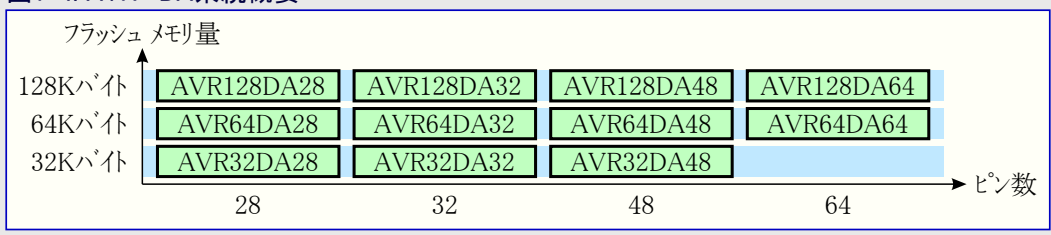


図1-4. AVR<sup>®</sup> DA系統概要

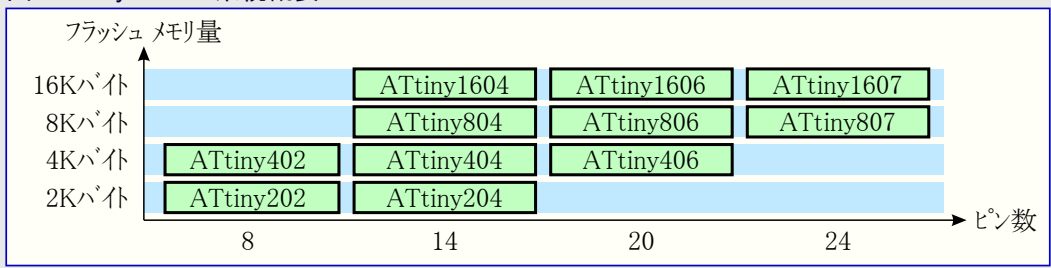


### 1.1. tinyAVR® 0系統

下図はピン数の変種とメモリ量を展開してtinyAVR® 0系統デバイスを示します。

- これらのデバイスが完全にピンと機能が互換のため、垂直方向移植はコード変更なしで可能です。
- 左への水平方向移植はピン数、従って利用可能な機能を減らします。

図1-5. tinyAVR® 0系統概要



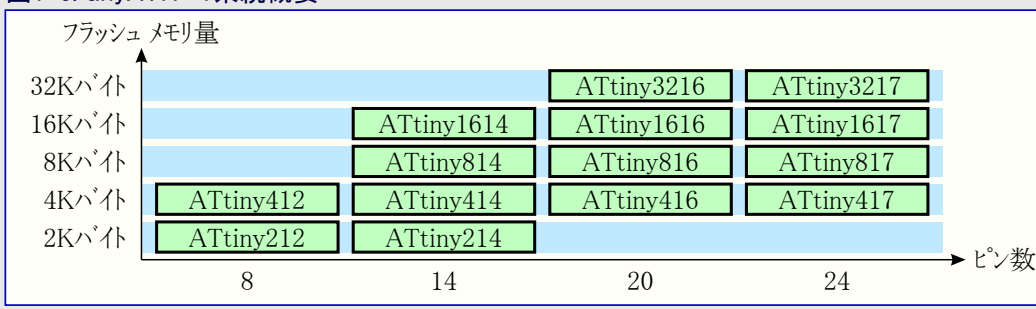
異なるフラッシュメモリ量を持つデバイスは一般的に異なるSRAMとEEPROMの量を持ちます。

### 1.2. tinyAVR® 1系統

下図はピン配置変種とメモリ量を展開してtinyAVR® 1系統デバイスを示します。

- これらのデバイスがピン互換で同じまたはより多くの機能を提供するため、垂直上方向移植はコード変更なしに可能です。下方向移植はより少ない利用可能ないくつかの周辺機能の実体のためにコード変更が必要かもしれません。
- 左への水平方向移植はピン数、従って利用可能な機能を減らします。

図1-6. tinyAVR® 1系統概要



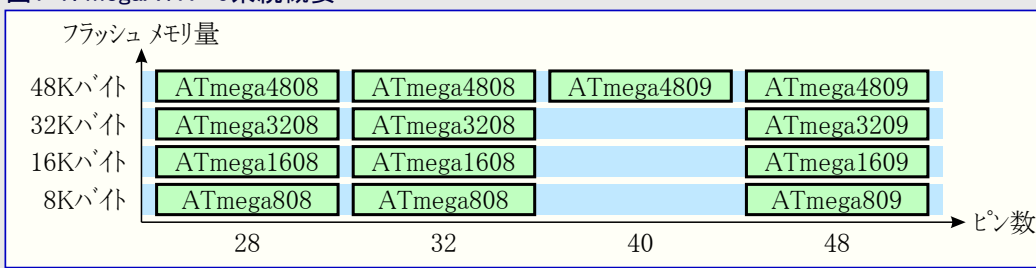
異なるフラッシュメモリ量を持つデバイスは一般的に異なるSRAMとEEPROMの量を持ちます。

### 1.3. megaAVR® 0系統

下図はピン配置変種とメモリ量を展開してmegaAVR® 0系統デバイスを示します。

- これらのデバイスが完全にピンと機能が互換のため、垂直方向移植はコード変更なしで可能です。
- 左への水平方向移植はピン数、従って利用可能な機能を減らします。

図1-7. megaAVR® 0系統概要

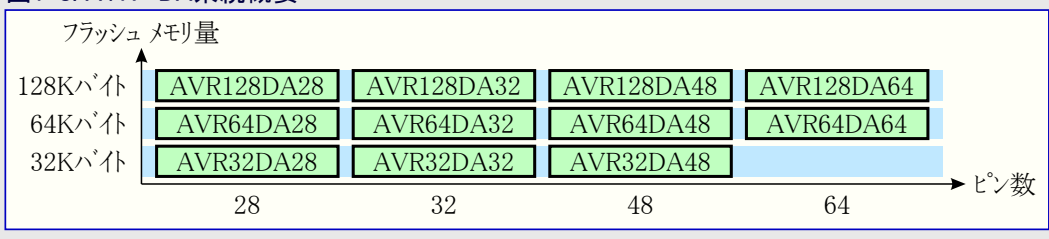


異なるフラッシュメモリ量を持つデバイスは一般的に異なるSRAMとEEPROMの量を持ちます。

### 1.4. AVR® DA系概要

次図はピン配置変種とメモリ量を展開してAVR® DAデバイスを示します。

- これらのデバイスが完全にピンと機能が互換のため、垂直方向移植はコード変更なしで可能です。
- 左への水平方向移植はピン数、従って利用可能な機能を減らします。

図1-8. AVR<sup>®</sup> DAシステム概要

異なるフラッシュメモリ量を持つデバイスは一般的に異なるSRAM量を持ちます。

## 2. 概要

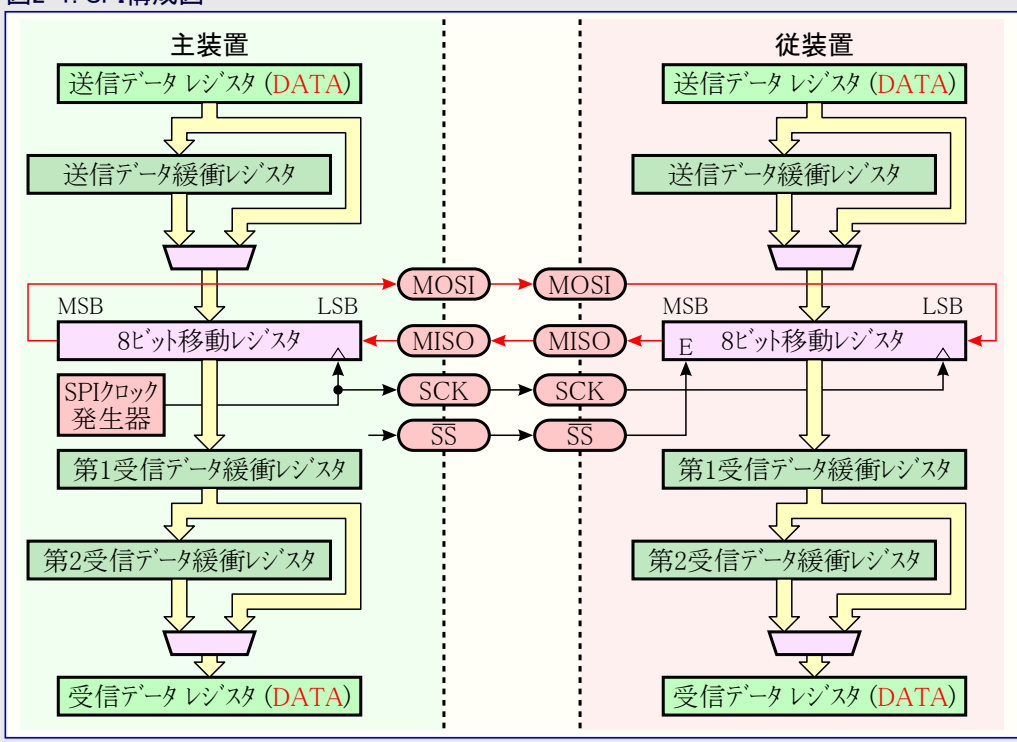
SPIバスは以下の4つの形式の論理信号に基づく同期直列通信インターフェースです。

- SCK : 直列クロック (主装置からの出力)
- MOSI : 主装置出力従装置入力 (主装置からのデータ出力)
- MISO : 主装置入力従装置出力 (従装置からのデータ出力)
- $\overline{SS}$  : 従装置選択 (Low活性、主装置からの出力)

この周辺機能は短距離での高速な通信用で、主に組み込みシステムで使われます。SPI装置はデータの送信用チャンネルと受信用チャンネルを使って全二重動作で通信します。SPIは同時に単一の主装置と1つまたは複数の従装置を持つ主装置-従装置基本構造に基づきます。主装置はクロックを生成することができる唯一のもので、従ってこれがデータ交換の開始者です。SPI主装置は全ての従装置に対して同じSCK、MOSI、MISOのチャンネルを使いますが、通常、従装置の各々に対して個別のSS線を使います。主装置はSS信号をLowに引くことによって望む従装置を選びます。

送るデータはデータレジスタまたは、送信が既に進行中で緩衝動作が有効にされた場合は緩衝レジスタのどちらかに格納されます。データは移動レジスタを使ってMOSIチャンネルで順次送られ、全てのビットはSPIクロック生成部を使って同期化されています。全てのビットが移動出力される間に、従装置から新しいデータがMISOチャンネルで受信されて受信緩衝部、更に受信DATAレジスタに移動されます。既に受信DATAレジスタにデータがあることを意味する受信部が多忙で、緩衝動作が有効にされた場合、データは一時的に第2受信緩衝部に格納されます。緩衝動作はCTRLBレジスタのBUFENビットを'1'に設定することによって有効にされます。

図2-1. SPI構成図



SPI単位部は5つのレジスタを持ちます。1つのレジスタはデータの転送と格納に使われ、2つのレジスタは割り込み要求フラグに使われ、残る2つのレジスタ(CTRLAとCTRLB)は初期化用です。周辺機能を正しく動かすのに必要とされる全ての構成設定はCTRLAレジスタ内のいくつかのビットを変更するように減らされ、一方でCTRLBレジスタは任意選択の各種動作形態に集中されます。レジスタに関するより多くの詳細はデバイスのシステムのデータシートに於いてこの周辺機能章のレジスタ要約で見つけることができます。

図2-2. レジスタ要約 - SPIn

変位	略称	ビット位置	ビット7	ビット6	ビット5	ビット4	ビット3	ビット2	ビット1	ビット0
+\$00	CTRLA	7~0		DORD	MASTER	CLK2X		PRESC1,0		ENABLE
+\$01	CTRLB	7~0	BUFEN	BUFWR				SSD		MODE1,0
+\$02	INTCTRL	7~0	RXCIE	TXCIE	DREIE	SSIE				IE
+\$03	INTFLAGS	7~0		IF	WRCOL					
			RXCIF	TXCIF	DREIF	SSIF			BUFOVF	
+\$04	DATA	7~0	DATA7~0							

### 3. SPI主装置としてデータ送信

主装置は従装置がメッセージの受け手である通信を起動する時を決める装置です。SPI主装置は一般的に高速通信で使われ、焦点は従装置として働く他の装置(例えば、感知器、メモリ、他のMCU)とデータを交換することです。

この使用事例は以下の手順に従います。

- SPIピンの位置を構成設定
- 周辺機能初期化
- ピンの方向を構成設定
- 従装置を制御
- 主装置としてデータ送出

#### SPIピンの位置の構成設定方法

ピンの位置を構成設定する方法は応用目的とSPI動作形態と無関係です。各マイクロコントローラは周辺機能に対して自身の既定物理ピン位置を持ちます。この位置はmegaAVR 0系の系統データシートのPORTMUX周辺機能章で見つけることができます。ATmega4809についてはSPIピンがPA7~4に配置され、PORTMUX単位のTWISPIROUTEAレジスタを使ってPC3~0またはPE3~0に変更することができます。

図3-1. TWISPIROUTEAレジスタ

ビット	7	6	5	4	3	2	1	0
	TWI01,0		TWI01,0		TWI01,0		SPI01,0	
アクセス種別	R	R	R/W	R/W	R	R	R/W	R/W
リセット値	0	0	0	0	0	0	0	0

ピンの順番は群のMOSI、MISO、SCK、 $\overline{SS}$ に従い、MOSIが群の最下位ピン番号を表します。これは使用者がSPIピンの位置をポートCでの任意選択1に変更することができる方法です。

値	0 0	0 1	1 0	1 1
名称	DEFAULT	ALT1	ALT2	NONE
説明	PA7~3	PC3~0	PE3~0	どのピンへも接続なし

これは以下のコードに変換できます。

```
PORTMUX.TWISPIROUTEA = PORTMUX_SPI00_bm;
```

または、ポートEでの任意選択2では、

値	0 0	0 1	1 0	1 1
名称	DEFAULT	ALT1	ALT2	NONE
説明	PA7~3	PC3~0	PE3~0	どのピンへも接続なし

これは以下のコードに変換できます。

```
PORTMUX.TWISPIROUTEA = PORTMUX_SPI01_bm;
```

#### 周辺機能の初期化方法

クロック周波数はマイクロコントローラの主クロックから派生され、前置分周器やSPIハードウェアに存在する分周回路を使って減ら(落と)されます。既定では、主クロックの供給元は20MHz内部発振器で、これは既定値6の前置分周器によって分周され、概ね3.33MHzの主クロック周波数になります。クロックについてのより多くの情報はmegaAVR 0系の系統データシートのクロック制御器章で見つけることができます。

SPIのクロック周波数は主装置動作でだけ動く倍速動作を使って増加することもできます。データ順ビットはデータの並び順(最上位ビットまたは最下位ビット)、チャンネルでどのビットが送信されるかの順番(レジスタから最後または最初のビットで開始)を表します。全ての構成設定はCTRLAレジスタに関連します。

図3-2. CTRLAレジスタ

ビット	7	6	5	4	3	2	1	0
		DORD	MASTER	CLK2X		PRESC1,0		ENABLE
アクセス種別	R	R/W	R/W	R/W	R	R/W	R/W	R/W
リセット値	0	0	0	0	0	0	0	0

次は前の話題で表された既定主クロック元と既定ピン位置で主装置SPI装置をどう構成設定するか例です。倍速動作と16分周でデータを構成設定することによって416kHzの周波数に帰着します。データは最下位ビット(LSB)で開始して移動出力されます。

値	0 0	0 1	1 0	1 1
名称	DIV4	DIV16	DIV64	DIV128
説明	CLK_PER/4	CLK_PER/16	CLK_PER/64	CLK_PER/128

これは以下のコードに変換できます。

```
SPI0.CTRLA = SPI_CLK2X_bm
             | SPI_DORD_bm
             | SPI_ENABLE_bm
             | SPI_MASTER_bm
             | SPI_PRESC_DIV16_gc;
```

### ピンの方向の構成設定方法

主装置は伝送を制御して開始するため、MOSI、SCK、 $\overline{SS}$ ピンが出力として構成設定されなければならない、一方でMISOチャンネルは入力としてその既定方向を保ちます。上で説明した既定値、方向、構成設定はここで適用できます。以下の例はSPIピンの既定位置に基づきます。

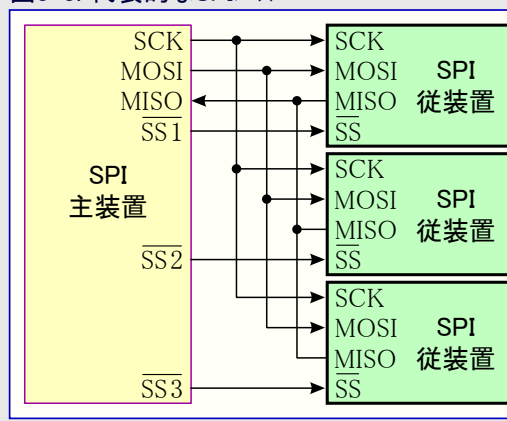
```
PORTA.DIR |= PIN4_bm; // MOSIチャンネル
PORTA.DIR &= ~PIN5_bm; // MISOチャンネル
PORTA.DIR |= PIN6_bm; // SCKチャンネル
PORTA.DIR |= PIN7_bm; //  $\overline{SS}$ チャンネル
```

SPI主装置は1つよりも多くの従装置を制御することができ、従ってより多くの $\overline{SS}$ ピンを必要とします。追加の $\overline{SS}$ チャンネルは単に上の例でのように構成設定することができます。使用者は未使用ピンを選んでその方向を出力として構成設定しなければなりません。

### 従装置の制御方法

主装置は $\overline{SS}$ ピンをLowに引くことによって従装置を制御します。従装置がMISOピンの方向を出力に設定している場合に $\overline{SS}$ ピンがLowの時に従装置の $\overline{SS}$ 駆動部はMISOピンの制御を取って送信DATAレジスタからデータを移動出力します。全ての従装置がメッセージを受け取ることができますが、Lowに引かれた $\overline{SS}$ ピンを持つそれらだけがデータを送り返すことができます。従って、(右の図のような)代表的な接続で複数の従装置を許可することは、それらの全てがメッセージに回答しようとし、1つのMISOチャンネルしかなく、従ってその伝送は書き込み衝突に帰着するため、推奨されません。使用者はINTFLAGSレジスタでWRCOLビットの値を読むことによって衝突の出現を調べることができます。

図3-3. 代表的なSPIバス



### 主装置としてデータを送る方法

前で構成した全ての設定が以下の例で考慮され、フラグ調査にポーリング法が使われます。データを送る前に、使用者は従装置にそれがメッセージの受信であることを知らせるために $\overline{SS}$ 信号をLowに引かなければなりません。

```
PORTA.OUT &= ~PIN7_bm;
```

一旦使用者がDATAレジスタに新しいデータを書くと、ハードウェアが線上にクロックを生成してビット移動出力する新しい転送を開始します。

図3-4. DATAレジスタ

ビット	7	6	5	4	3	2	1	0
	DATA7~0							
アクセス種別	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
リセット値	0	0	0	0	0	0	0	0

```
SPI0.DATA = data;
```

ハードウェアが全ビットの移動を終えると、INTFLAGSレジスタで見つけることができる受信割り込み要求フラグが活性(1)にされます。

図3-5. INTFLAGSレジスタ

ビット	7	6	5	4	3	2	1	0
	IF	WRCOL						
アクセス種別	R/W	R/W	R	R	R	R	R	R
リセット値	0	0	0	0	0	0	0	0

使用者は新しいデータをレジスタに書く前に、割り込みを有効にするか、または定常的にこのフラグの値を読む(ポーリングと呼ばれる方法)のどちらかによってフラグの状態を調べなければならず、そうしなければ書き込み衝突割り込みが起きます。

```
while (!(SPI0.INTFLAGS & SPI_IF_bm))
{
;
}
```

送信するものが残っていない場合、使用者はSSチャネルをHighに引くことができます。

```
PORTA.OUT |= PIN7_bm;
```



助言: 完全なコード例は「[追補](#)」章でも利用可能です。



GitHubでATmega4809コード例を見てください。  
貯蔵庫を閲覧するにはクリックしてください。

本章で記述されたのと同じ機能を持つAVR128DA48用のMCC生成されたコード例は以下のここで見つけることができます。



GitHubでAVR128DA48コード例を見てください。  
貯蔵庫を閲覧するにはクリックしてください。

## 4. SPI従装置としてデータ受信

従装置は通常、機械的な駆動部です。従装置はどんな活動も開始せず、それらは主装置が始める時にだけ必ず働きます。従装置は常に利用可能でなければならず、主装置がSSチャネルをLowに引くまで待たなければなりません。

この使用事例は以下の手順に従います。

- [周辺機能初期化](#)
- [SPI従装置ピン方向構成設定](#)
- [SPI従装置としてデータ受信](#)

### 周辺機能の初期化方法

従装置は主装置からクロック信号を得、故に周辺機能のクロック分周器を変更する点はなく、その変更はSPI従装置の時に影響を及ぼされません。とは言え、周辺機能ハードウェアはMOSIチャネルで受信されるデータを採取しなければなりません。正しく再構成されるデータ信号のため、デバイスの主クロック周波数はSPI SCKチャネルで受信したクロックの最低2倍でなければなりません。

従装置がマイクロコントローラの場合、使用者は周波数要求を考慮して強力なクロック元を構成設定しなければなりません。使用者が従装置のクロック発生器へのアクセスを持たない場合、主装置は従装置の制限を超えないことを保証しなければなりません。これは主装置が同じシステムまたは応用の一部で、周波数を容易に変更することができるマイクロコントローラによって主にSPI周波数または主クロック周波数のどちらかで表現されます。

例をより簡単に理解させるため、「[3. SPI主装置としてデータ送信](#)」章で提示された情報のいくつかもここで適用されます。デバイスは3.33 MHzの主クロック周波数で、データがLSBから始めて移動出力する従装置として構成設定されます。従装置としてのデバイス構成設定は主に単位部を許可してCTRLAレジスタの主装置ビットを無効にすることで再び始めます。

図4-1. CTRLAレジスタ

ビット	7	6	5	4	3	2	1	0
		DORD	MASTER	CLK2X		PRESC1,0		ENABLE
アクセス種別	R	R/W	R/W	R/W	R	R/W	R/W	R/W
リセット値	0	0	0	0	0	0	0	0

```
SPI0. CTRLA = SPI_DORD_bm
              | SPI_ENABLE_bm
              & (~SPI_MASTER_bm);
```

### SPI従装置のピン方向の構成設定

デバイスがSPI従装置動作の時はMOSI、SCK、 $\overline{SS}$ のピンが入力チャネルとして構成設定されることを必要とします。既定で、全ての入出力(I/O)ピンは入力として構成設定され、故にこれらのピンに対して変更される必要はありません。従って、SPIのハードウェア回路はこの周辺機能が許可された場合に送信中にこれらのチャネルの制御を取ります。データをやり返すことが必須でないため、MISOチャネルは出力または入力のどちらにも構成設定することができます。

標準動作はこのピンを出力として構成設定することで、ハードウェア回路はデータ交換中にその動きを制御します。このピンが入力として構成設定された場合、普通のI/Oピンとして働き、SPIによって使われません。

DIRレジスタのピン値が値0を持つ時にピンは各々デジタル入力ピン、値1に対してはデジタル出力ピンとして働きます。SPIピンの既定位置が考慮されます。ピンの既定方向値が変更されなかったことを保証するため、全ての必要とされるピンは次のように構成設定されます。

```
PORTA.DIR &= ~PIN4_bm; // MOSIチャネル
PORTA.DIR |= PIN5_bm; // MISOチャネル
PORTA.DIR &= ~PIN6_bm; // SCKチャネル
PORTA.DIR &= ~PIN7_bm; // SSチャネル
```

### SPI従装置としてデータを受信する方法

SPIバスに接続された全ての従装置は主装置によってMOSIチャネルで送られたメッセージを受信します。従装置は $\overline{SS}$ チャネルがLowに引かれていない限り、メッセージに回答することができません。主装置が $\overline{SS}$ ピンをLowに引くと、従装置のSPI周辺機能はMISOの制御を取ってデータを移動出力します。使用者がDATAレジスタに書かない場合、従装置はデータを送らず、周辺機能は0で満たされたバイトを移動出力します。

周辺機能はINTGLAGSレジスタのIFフラグを有効(1)にすることによって新しいデータの受信を合図します。使用者は主装置例で提示されたようなポーリング法によって、または割り込みによってのどちらかでこのビットの値を調べなければなりません。以下の例は、主装置が新しいデータを送る時を知らせる方法がなく、非中断(非完了復帰)のため、このビットの値を確定するのに割り込みを使います。従って、デバイスはアイドルSPI時間中に行わなければならないことは何でも行うことができます。

割り込み使用時、3つの重要なことが考慮に入れられなければなりません。

1. マイクロコントローラに対して割り込みを有効にします。<avr/interrupt.h>のインクルードによってマクロを使うことができます。

```
sei();
```

2. 周辺機能に対する割り込み有効化はINTCTRLレジスタのIEビットを有効(1)にすることによって行うことができます。

図4-2. INTCTRLレジスタ

ビット	7	6	5	4	3	2	1	0
	RXCIE	TXCIE	DREIE	SSIE				IE
アクセス	R/W	R/W	R/W	R/W	R	R	R	R/W
リセット値	0	0	0	0	0	0	0	0

```
SPI0.INTCTRL = SPI_IE_bm;
```

3. ハードウェアによって自動的に解除(0)されない場合、割り込み要求フラグを解除(0)します。新しいデータ受信後、受信完了割り込み要求フラグが有効(1)にされます。これはINTFLAGSレジスタで見つけることができます。

図4-3. INTFLAGSレジスタ

ビット	7	6	5	4	3	2	1	0
	IF	WRCOL						
アクセス	R/W	R/W	R	R	R	R	R	R
リセット値	0	0	0	0	0	0	0	0

割り込み要求フラグの解除(0)は使用者がその応用の目的に基づいてその割り込みルーチンを挿入することもできる、割り込み関数内でこのビットに'1'を書くことによって行われます。

下の例では受信したデータを読み、割り込みを解除(0)してDATAレジスタに書く方法が示されます(受信したデータで何を行うかと主装置に何を書き戻すかは使用者の選択です)。

```
ISR(SPI0_INT_vect)
{
    receiveData = SPI0.DATA;
    SPI0.DATA = writeData;
    SPI0.INTFLAGS = SPI_IF_bm;
}
```



助言: 完全なコード例は「[追補](#)」章でも利用可能です。



GitHubでATmega4809コード例を見てください。  
貯蔵庫を閲覧するにはクリックしてください。

本章で記述されたのと同じ機能を持つAVR128DA48用のMCC生成されたコード例は以下のここで見つけることができます。



GitHubでAVR128DA48コード例を見てください。  
貯蔵庫を閲覧するにはクリックしてください。

## 5. データ転送形式変更

これはデータがクロック生成に関してどう送信されるかを表します。クロック極性とクロック位相がデータ動作種別に対して重要なものです。クロック極性により、1つはアイドル状態の間でLowにすることができる信号の基準を理解することができ、データを送信する時に上昇端で開始し、またはアイドル状態の間でHighにすることができ、データ交換時に下降端で始まります。位相に依存して、データはチャネルでのクロック(上昇端または下降端)に関して生成または採取されます。下図をご覧ください。

図5-1. SPIデータ転送形態

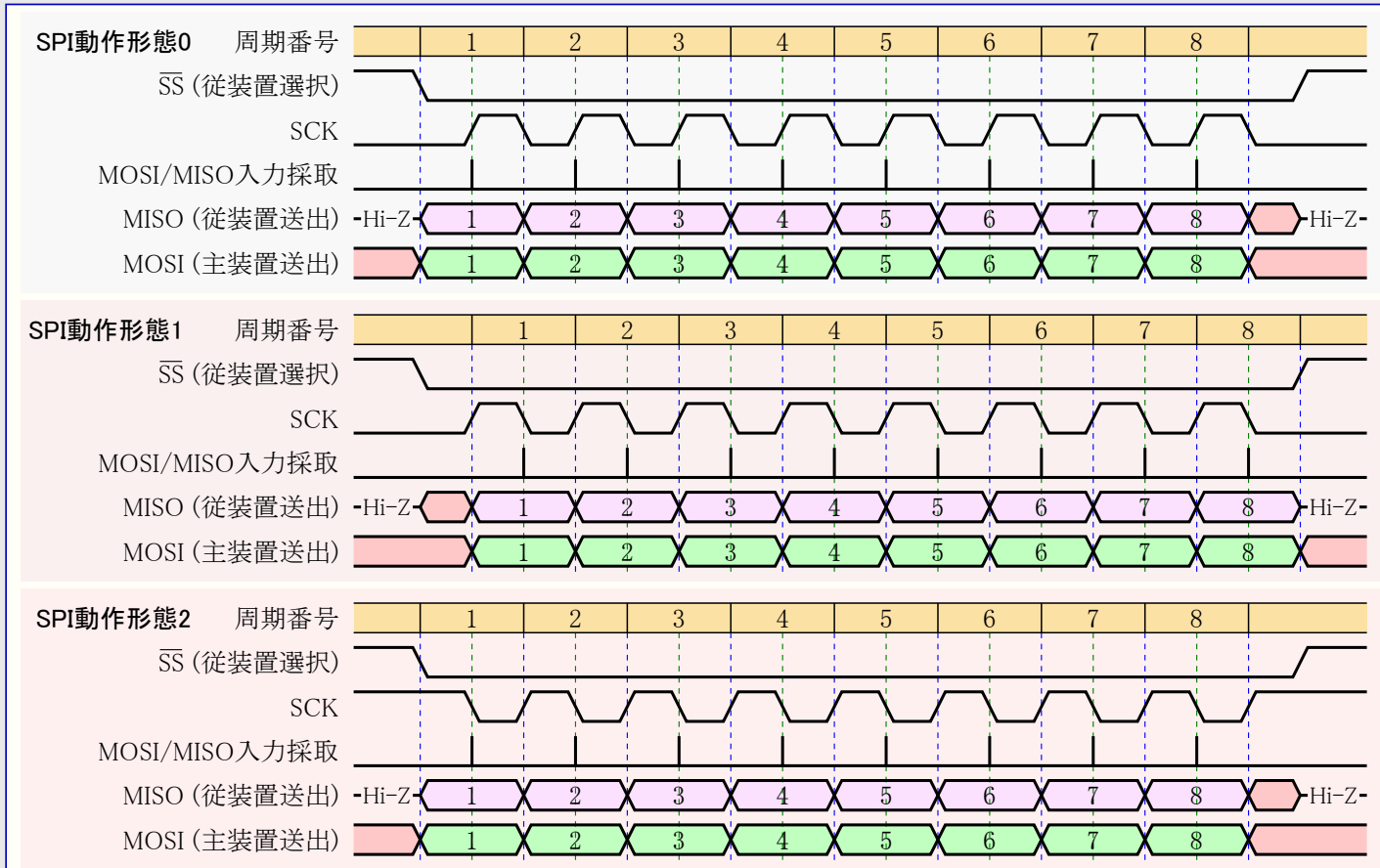
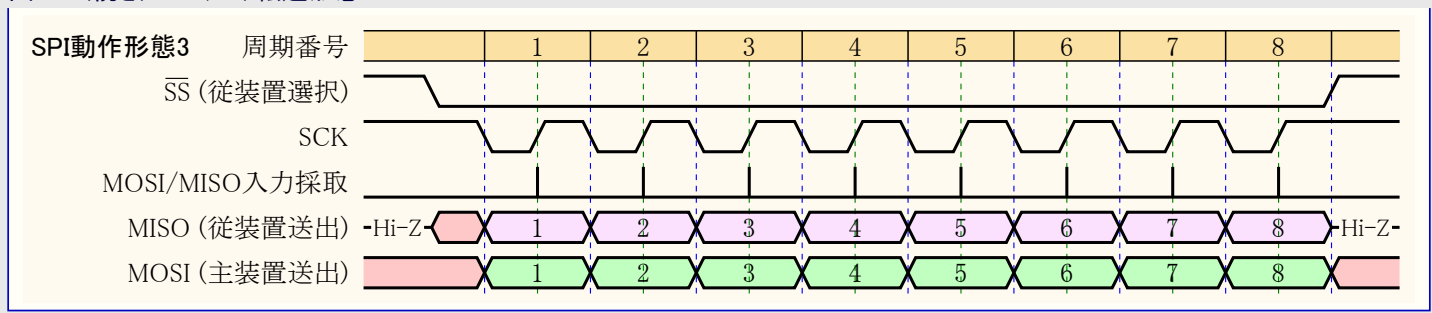


図5-1 (続き). SPIデータ転送形態



主装置と従装置の両方は他が符号化したものを正しく復号できるように同じ方法で構成設定されなければなりません。データ動作形態はCTRLレジスタのMODE1,0ビット領域の値を変更することによって選ぶことができます。

図5-2. CTRLレジスタ

ビット	7	6	5	4	3	2	1	0
	BUFEN	BUFWR				SSD	MODE1,0	
アクセス種別	R/W	R/W	R	R	R	R/W	R/W	R/W
リセット値	0	0	0	0	0	0	0	0

今まで、それらのビットに変更が行われず、ビットの既定値だったため、例はSPI動作形態0に基づいていました。

下はSPI動作形態3を構成設定する方法の例で、「3. SPI主装置としてデータ送信」章で提示された標準/基本主装置SPI初期化動作に基づき、違いはデータ伝送形式の変更だけです。

```
SPI0. CTRLB = SPI_MODE_3_gc;
```

値	0 0	0 1	1 0	1 1
名称	0	1	2	3
説明	先行端：上昇、入力採取 後行端：下降、出力設定	先行端：上昇、出力設定 後行端：下降、入力採取	先行端：下降、入力採取 後行端：上昇、出力設定	先行端：下降、出力設定 後行端：上昇、入力採取



助言：完全なコード例は「追補」章でも利用可能です。



GitHubでATmega4809コード例を見てください。  
貯蔵庫を閲覧するにはクリックしてください。

本章で記述されたのと同じ機能を持つAVR128DA48用のMCC生成されたコード例は以下のここで見つけることができます。



GitHubでAVR128DA48コード例を見てください。  
貯蔵庫を閲覧するにはクリックしてください。

## 6. 参考文献

1. AVR128DA48製品頁：[www.microchip.com/wwwproducts/en/AVR128DA48](http://www.microchip.com/wwwproducts/en/AVR128DA48)
2. AVR128DA48 Curiosity Nano評価キット製品頁：<https://www.microchip.com/Developmenttools/ProductDetails/DM164151>
3. AVR128DA28/32/48/64データシート
4. AVR® DA系での開始に際して
5. ATmega4809製品頁：[www.microchip.com/wwwproducts/en/ATMEGA4809](http://www.microchip.com/wwwproducts/en/ATMEGA4809)
6. megaAVR® 0系データシート
7. megaAVR® 0系ATmega809/1609/3209/4809 - 48ピンデータシート
8. ATmega4809 Xplained Pro製品頁：<https://www.microchip.com/developmenttools/ProductDetails/atmega4809-xpro>

## 7. 追補

例7-1. SPI主装置としてデータ送信全コード例

```

#include <avr/io.h>

void SPI0_init(void);
void clientSelect(void);
void clientDeselect(void);
uint8_t SPI0_exchangeData(uint8_t data);

void SPI0_init(void)
{
    PORTA.DIR |= PIN4_bm;           /* MOSIピン方向を出力に設定 */
    PORTA.DIR &= ~PIN5_bm;         /* MISOピン方向を入力に設定 */
    PORTA.DIR |= PIN6_bm;         /* SCKピン方向を出力に設定 */
    PORTA.DIR |= PIN7_bm;         /* SSピン方向を出力に設定 */

    SPI0.CTRLA = SPI_CLK2X_bm      /* 倍速許可 */
                | SPI_DORD_bm      /* 最初にLSBが送信されます。 */
                | SPI_ENABLE_bm    /* 単位部許可 */
                | SPI_MASTER_bm    /* 主装置SPI単位部 */
                | SPI_PRESC_DIV16_gc; /* 16分周システムクロック */
}

uint8_t SPI0_exchangeData(uint8_t data)
{
    SPI0.DATA = data;

    while (!(SPI0.INTFLAGS & SPI_IF_bm)) /* データが交換されるまで待ち */
    {
        ;
    }

    return SPI0.DATA;
}

void clientSelect(void)
{
    PORTA.OUT &= ~PIN7_bm;         // SSピン値をLowに設定
}

void clientDeselect(void)
{
    PORTA.OUT |= PIN7_bm;         // SSピン値をHighに設定
}

int main(void)
{
    uint8_t data = 0;

    SPI0_init();

    while (1)
    {
        clientSelect();
        SPI0_exchangeData(data);
        clientDeselect();
    }
}

```

## 例7-2. SPI従装置としてデータ受信全コード例

```

#include <avr/io.h>
#include <avr/interrupt.h>

void SPI0_init(void);

volatile uint8_t receiveData = 0;
volatile uint8_t writeData = 0;

void SPI0_init(void)
{
    PORTA.DIR &= ~PIN4_bm;           /* MOSIピン方向を入力に設定 */
    PORTA.DIR |= PIN5_bm;           /* MISOピン方向を出力に設定 */
    PORTA.DIR &= ~PIN6_bm;         /* SCKピン方向を入力に設定 */
    PORTA.DIR &= ~PIN7_bm;         /* SSピン方向を入力に設定 */

    SPI0.CTRLA = SPI_DORD_bm        /* 最初にLSBが送信されます。 */
                | SPI_ENABLE_bm    /* 単位部許可 */
                & (~SPI_MASTER_bm); /* 従装置SPI単位部 */

    SPI0.INTCTRL = SPI_IE_bm;      /* SPI割り込み許可 */
}

ISR(SPI0_INT_vect)
{
    receiveData = SPI0.DATA;

    SPI0.DATA = writeData;

    SPI0.INTFLAGS = SPI_IF_bm;     /* 1書き込みによって割り込み要求フラグ解除(0) */
}

int main(void)
{
    SPI0_init();

    sei();                          /* 全体割り込み許可 */

    while (1)
    {
        ;
    }
}

```

## 例7-3. データ転送形式変更全コード例

```

#include <avr/io.h>

void SPI0_init(void);
void clientSelect(void);
void clientDeselect(void);
uint8_t SPI0_exchangeData(uint8_t data);

void SPI0_init(void)
{
    PORTA.DIR |= PIN4_bm;           /* MOSIピン方向を出力に設定 */
    PORTA.DIR &= ~PIN5_bm;         /* MISOピン方向を入力に設定 */
    PORTA.DIR |= PIN6_bm;         /* SCKピン方向を出力に設定 */
    PORTA.DIR |= PIN7_bm;         /* SSピン方向を出力に設定 */

    SPI0.CTRLA = SPI_CLK2X_bm      /* 倍速許可 */

```

## 例7-3 (続き). データ転送形式変更全コード例

```

        | SPI_DORD_bm           /* 最初にLSBが送信されます。 */
        | SPI_ENABLE_bm       /* 単位部許可 */
        | SPI_MASTER_bm      /* 主装置SPI単位部 */
        | SPI_PRESC_DIV16_gc; /* 16分周システムクロック */

    SPI0.CTRLB |= SPI_MODE_3_gc; /* データ転送形態3 */
}

uint8_t SPI0_exchangeData(uint8_t data)
{
    SPI0.DATA = data;

    while (!(SPI0.INTFLAGS & SPI_IF_bm)) /* データが交換されるまで待ち */
    {
        ;
    }

    return SPI0.DATA;
}

void clientSelect(void)
{
    PORTA.OUT &= ~PIN7_bm; // SSピン値をLowに設定
}

void clientDeselect(void)
{
    PORTA.OUT |= PIN7_bm; // SSピン値をHighに設定
}

int main(void)
{
    uint8_t data = 0;

    SPI0_init();

    while (1)
    {
        clientSelect();
        SPI0_exchangeData(data);
        clientDeselect();
    }
}

```

## 8. 改訂履歴

文書改訂	日付	注釈
A	2019年5月	初版文書公開
B	2021年2月	GitHub貯蔵庫リンク、「参考文献」、使用事例章更新。「AVR® DA系概要」項と「改訂履歴」章追加。各使用事例に対してAVR128DA48で動くMCC版を追加。その他些細な修正。

## Microchipウェブ サイト

Microchipは[www.microchip.com/](http://www.microchip.com/)で当社のウェブ サイト経由でのオンライン支援を提供します。このウェブ サイトはお客様がファイルや情報を容易に利用可能にするのに使われます。利用可能な情報のいくつかは以下を含みます。

- **製品支援** – データシートと障害情報、応用記述と試供プログラム、設計資源、使用者の手引きとハードウェア支援資料、最新ソフトウェア配布と保管されたソフトウェア
- **一般的な技術支援** – 良くある質問(FAQ)、技術支援要求、オンライン検討グループ、Microchip設計協力課程会員一覧
- **Microshipの事業** – 製品選択器と注文の手引き、最新Microchip報道発表、セミナーとイベントの一覧、Microchip営業所の一覧、代理店と代表する工場

## 製品変更通知サービス

Microchipの製品変更通知サービスはMicrochip製品を最新に保つのに役立ちます。加入者は指定した製品系統や興味のある開発ツールに関連する変更、更新、改訂、障害情報がある場合に必ず電子メール通知を受け取ります。

登録するには[www.microchip.com/pcn](http://www.microchip.com/pcn)へ行って登録指示に従ってください。

## お客様支援

Microchip製品の使用者は以下のいくつかのチャネルを通して支援を受け取ることができます。

- 代理店または販売会社
- 最寄りの営業所
- 組み込み解決技術者(ESE:Embedded Solutions Engineer)
- 技術支援

お客様は支援に関してこれらの代理店、販売会社、またはESEに連絡を取るべきです。最寄りの営業所もお客様の手助けに利用できます。営業所と位置の一覧はこの資料の後ろに含まれます。

技術支援は[www.microchip.com/support](http://www.microchip.com/support)でのウェブ サイトを通して利用できます。

## 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、その製品系統が安全であると考えます。
- Microchipデバイスのコード保護機能を破ろうとする試みに使われる不正でおそらく違法な方法があります。当社はこれらの方法が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とされると確信しています。これらのコード保護機能を破ろうとする試みは、おそらく、Microchipの知的財産権に違反することなく達成することはできません。
- Microchipはそのコードの完全性について心配されている何れのお客様とも共に働きたいと思えます。
- Microchipや他のどの半導体製造業者もそのコードの安全を保証することはできません。コード保護は製品が”破ることができない”ことを当社が保証すると言った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

この刊行物に含まれる情報はMicrochip製品を使って設計する唯一の目的のために提供されます。デバイス応用などに関す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。

この情報はMicrochipによって「現状そのまま」で提供されます。Microchipは非侵害、商品性、特定目的に対する適合性の何れの黙示的保証やその条件、品質、性能に関する保証を含め、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。

如何なる場合においても、Microchipは情報またはその使用に関連するあらゆる種類の間接的、特別的、懲罰的、偶発的または結果的な損失、損害、費用または経費に対して責任を負わないものとします。法律で認められている最大限の範囲で、情報またはその使用に関連する全ての請求に対するMicrochipの全責任は、もしあれば、情報のためにMicrochipへ直接支払った料金を超えないものとします。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、Mmicrochipロゴ、Adaptec、AnyRate、AVR、AVRロゴ、AVR Freaks、BesTime、BitCloud、chipKIT、chipKITロゴ、CryptoMemory、CryptoRF、dsPIC、FlashFlex、flexPWR、HELDO、IGLOO、JukeBlox、KeeLoq、Kleer、LANCheck、LinkMD、maXStylus、maXTouch、MediaLB、megaAVR、Microsemi、Microsemiロゴ、MOST、MOSTロゴ、MPLAB、OptoLyzer、PacTime、PIC、picoPower、PICSTART、PIC32ロゴ、PolarFire、Prochip Designer、QTouch、SAM-BA、SenGenuity、SpyNIC、SST、SSTロゴ、SuperFlash、Symmetricom、SyncServer、Tachyon、TimeSource、tinyAVR、UNI/O、Vectron、XMEGAは米国と他の国に於けるMicrochip Technology Incorporatedの登録商標です。

AgileSwitch、APT、ClockWorks、The Embedded Control Solutions Company、EtherSynch、FlashTec、Hyper Speed Control、Hyper Light Load、IntelliMOS、Libero、motorBench、mTouch、Powermite 3、Precision Edge、ProASIC、ProASIC Plus、ProASIC Plusロゴ、Quiet-Wire、SmartFusion、SyncWorld、Temux、TimeCesium、TimeHub、TimePictra、TimeProvider、Vite、WinPath、ZL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、Augmented Switching、BlueSky、BodyCom、CodeGuard、CryptoAuthentication、CryptoAutomotive、CryptoCompanion、CryptoController、dsPICDEM、dsPICDEM.net、Dynamic Average Matching、DAM、ECAN、Espresso T1S、EtherGREEN、IdealBridge、In-Circuit Serial Programming、ICSP、INICnet、Intelligent Paralleling、Inter-Chip Connectivity、JitterBlocker、maxCrypto、maxView、memBrain、Mindi、MiWi、MPASM、MPF、MPLAB Certifiedロゴ、MPLIB、MPLINK、MultiTRAK、NetDetach、Omniscient Code Generation、PICDEM、PICDEM.net、PICkit、PICKtail、PowerSmart、PureSilicon、QMatrix、REAL ICE、Ripple Blocker、RTAX、RTG4、SAM-ICE、Serial Quad I/O、simpleMAP、SimpliPHY、SmartBuffer、SMART-I.S.、storClad、SQI、SuperSwitcher、SuperSwitcher II、Switchtec、SynchronPHY、Total Endurance、TSHARC、USBCheck、VariSense、VectorBlox、VeriPHY、ViewSpan、WiperLock、XpressConnect、and 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。

Adaptecロゴ、Frequency on Demand、Silicon Storage Technology、Symmcom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。

© 2021年、Microchip Technology Incorporated、米国印刷、不許複製

## 品質管理システム

Microchipの品質管理システムに関する情報については[www.microchip.com/quality](http://www.microchip.com/quality)を訪ねてください。

日本語© HERO 2021.

本技術概説はMicrochipのTB3215技術概説(DS90003215B-2021年2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。



**MICROCHIP**

## 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
<b>本社</b> 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-792-7200 Fax: 480-792-7277 技術支援: <a href="http://www.microchip.com/support">www.microchip.com/support</a> ウェブアドレス: <a href="http://www.microchip.com">www.microchip.com</a> <b>アトランタ</b> Duluth, GA Tel: 678-957-9614 Fax: 678-957-1455 <b>オースチン TX</b> Tel: 512-257-3370 <b>ボストン</b> Westborough, MA Tel: 774-760-0087 Fax: 774-760-0088 <b>シカゴ</b> Itasca, IL Tel: 630-285-0071 Fax: 630-285-0075 <b>ダラス</b> Addison, TX Tel: 972-818-7423 Fax: 972-818-2924 <b>デトロイト</b> Novi, MI Tel: 248-848-4000 <b>ヒューストン TX</b> Tel: 281-894-5983 <b>インディアナポリス</b> Noblesville, IN Tel: 317-773-8323 Fax: 317-773-5453 Tel: 317-536-2380 <b>ロサンゼルス</b> Mission Viejo, CA Tel: 949-462-9523 Fax: 949-462-9608 Tel: 951-273-7800 <b>ローリー NC</b> Tel: 919-844-7510 <b>ニューヨーク NY</b> Tel: 631-435-6000 <b>サンホセ CA</b> Tel: 408-735-9110 Tel: 408-436-4270 <b>カナダ - トロント</b> Tel: 905-695-1980 Fax: 905-695-2078	<b>オーストラリア - シドニー</b> Tel: 61-2-9868-6733 <b>中国 - 北京</b> Tel: 86-10-8569-7000 <b>中国 - 成都</b> Tel: 86-28-8665-5511 <b>中国 - 重慶</b> Tel: 86-23-8980-9588 <b>中国 - 東莞</b> Tel: 86-769-8702-9880 <b>中国 - 広州</b> Tel: 86-20-8755-8029 <b>中国 - 杭州</b> Tel: 86-571-8792-8115 <b>中国 - 香港特別行政区</b> Tel: 852-2943-5100 <b>中国 - 南京</b> Tel: 86-25-8473-2460 <b>中国 - 青島</b> Tel: 86-532-8502-7355 <b>中国 - 上海</b> Tel: 86-21-3326-8000 <b>中国 - 瀋陽</b> Tel: 86-24-2334-2829 <b>中国 - 深圳</b> Tel: 86-755-8864-2200 <b>中国 - 蘇州</b> Tel: 86-186-6233-1526 <b>中国 - 武漢</b> Tel: 86-27-5980-5300 <b>中国 - 西安</b> Tel: 86-29-8833-7252 <b>中国 - 廈門</b> Tel: 86-592-2388138 <b>中国 - 珠海</b> Tel: 86-756-3210040	<b>インド - ハンガロール</b> Tel: 91-80-3090-4444 <b>インド - ニューデリー</b> Tel: 91-11-4160-8631 <b>インド - フォン</b> Tel: 91-20-4121-0141 <b>日本 - 大阪</b> Tel: 81-6-6152-7160 <b>日本 - 東京</b> Tel: 81-3-6880-3770 <b>韓国 - 大邱</b> Tel: 82-53-744-4301 <b>韓国 - ソウル</b> Tel: 82-2-554-7200 <b>マレーシア - クアラルンプール</b> Tel: 60-3-7651-7906 <b>マレーシア - ペナン</b> Tel: 60-4-227-8870 <b>フィリピン - マニラ</b> Tel: 63-2-634-9065 <b>シンガポール</b> Tel: 65-6334-8870 <b>台湾 - 新竹</b> Tel: 886-3-577-8366 <b>台湾 - 高雄</b> Tel: 886-7-213-7830 <b>台湾 - 台北</b> Tel: 886-2-2508-8600 <b>タイ - バンコク</b> Tel: 66-2-694-1351 <b>ベトナム - ホーチミン</b> Tel: 84-28-5448-2100	<b>オーストラリア - ウェルズ</b> Tel: 43-7242-2244-39 Fax: 43-7242-2244-393 <b>デンマーク - コペンハーゲン</b> Tel: 45-4485-5910 Fax: 45-4485-2829 <b>フィンランド - エスポー</b> Tel: 358-9-4520-820 <b>フランス - パリ</b> Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79 <b>ドイツ - ガルヒング</b> Tel: 49-8931-9700 <b>ドイツ - ハーン</b> Tel: 49-2129-3766400 <b>ドイツ - ハイムブロン</b> Tel: 49-7131-72400 <b>ドイツ - カールスルーエ</b> Tel: 49-721-625370 <b>ドイツ - ミュンヘン</b> Tel: 49-89-627-144-0 Fax: 49-89-627-144-44 <b>ドイツ - ローゼンハイム</b> Tel: 49-8031-354-560 <b>イスラエル - ラーナナ</b> Tel: 972-9-744-7705 <b>イタリア - ミラノ</b> Tel: 39-0331-742611 Fax: 39-0331-466781 <b>イタリア - パドバ</b> Tel: 39-049-7625286 <b>オランダ - デルフト</b> Tel: 31-416-690399 Fax: 31-416-690340 <b>ノルウェー - トロンハイム</b> Tel: 47-72884388 <b>ポーランド - ワルシャワ</b> Tel: 48-22-3325737 <b>ルーマニア - ブカレスト</b> Tel: 40-21-407-87-50 <b>スペイン - マドリッド</b> Tel: 34-91-708-08-90 Fax: 34-91-708-08-91 <b>スウェーデン - イェテボリ</b> Tel: 46-31-704-60-40 <b>スウェーデン - ストックホルム</b> Tel: 46-8-5090-4654 <b>イギリス - ウォーキングム</b> Tel: 44-118-921-5800 Fax: 44-118-921-5820