# 特徴

- 高性能、低消費AVR® 8ビット マイクロ コントローラ
- ・進化したRISC構造
  - 強力な131命令(多くは1周期で実行)
  - 32個の1バイト長汎用レジスタ
  - 完全なスタティック動作
  - 16MHz時、16MIPSに達する高速動作
  - 2周期実行の乗算命令

### • データ メモリと不揮発性プログラム メモリ

- 実装自己書き換え可能な16Kバイト(8K語)フラッシュ メモリ内蔵(10,000回の書き換え可能)
- 個別施錠ビットを持つ任意のブート コード領域 チップ内ブート プログラムによる実装書き換え 真の書き込み中の読み出し動作
- 512 n イトのEEPROM (100,000回の書き換えが可能)
- 1Kバイトの内蔵SRAM
- ソフトウェア保護用の設定可能な施錠機能

### • JTAG (IEEE 1149.1準拠) インターフェース

- JTAG規格に従った境界走査(Boundary-Scan)能力
- 広範囲な内蔵デバッグ機能
- JTAGインターフェース経由でのフラッシュ、EEPROM、ヒュース、施錠ビットのプログラミング

### • 内蔵周辺機能

- 4×25セグメントのLCD制御器/駆動部
- 独立した前置分周器、比較機能付き2つの8ビットタイマ/カウンタ
- 独立した前置分周器、比較、捕獲機能付き1つの16ビットタイマ/カウンタ
- 専用発振器と8ビット タイマ/カウンタによる実時間計数器(RTC)
- 4つのPWM出力
- 差動入力付き8チャネルの10ビット A/D変換器
- 設定可能な直列USART
- 主装置/従装置動作SPI直列インターフェース
- 開始条件検出器付き多用途直列インターフェース
- 設定可能な専用発振器付きウォッチト、ック、タイマ
- アナログ比較器
- ピン変化での割り込みと起動復帰

### 特殊マイクロ コントローラ機能

- 電源ONリセット回路と設定可能な低電圧検出器(BOD)
- 校正付き内蔵RC発振器
- 外部及び内部の割り込み
- アイドル、A/D変換雑音低減、パワーセーブ、パワータヴン、スタンバイ、の5つの低消費動作

# • I/Oと外囲器

- 53ビットの設定可能なI/O
- 64リート TQFP,64ハ ット QFN/MLF

### • 動作速度/電圧

- ATmega169V:

 $0\sim4 \text{MHz}/1.8\sim5.5 \text{V}$ 

 $0 \sim 8 MHz/2.7 \sim 5.5 V$ 

- ATmega169:

 $0 \sim 8 \text{MHz} / 2.7 \sim 5.5 \text{V}$ 

 $0\sim16 MHz/4.5\sim5.5 V$ 

### • 温度範囲

--40~85℃(工業用)

### • 超低消費電力

- 活動動作

 $1MHz/1.8V : 350\mu A$ 

32kHz/1.8V: 20μA(発振器含む) 32kHz/1.8V: 40μA(LCD、発振器含む)

- パワーダウン動作 0.1µA/1.8V



8ビット **AVA**® マイクロ コントローラ 実装書き換え可能な 16 K バイトフラッシュ メモリ内蔵

# ATmega169 ATmega169V

### 诵告

新規設計には推奨されません。

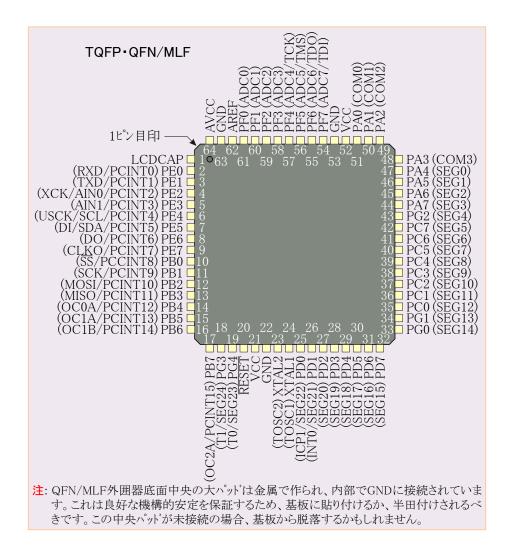
本書は一般の方々の便宜のため有志によって作成されたもので、Atmel社とは無関係であることを御承知ください。しおりの[はじめに]での内容にご注意ください。

Rev. 2514P-07/06, 2514PJ19-09/22





# ピン配置

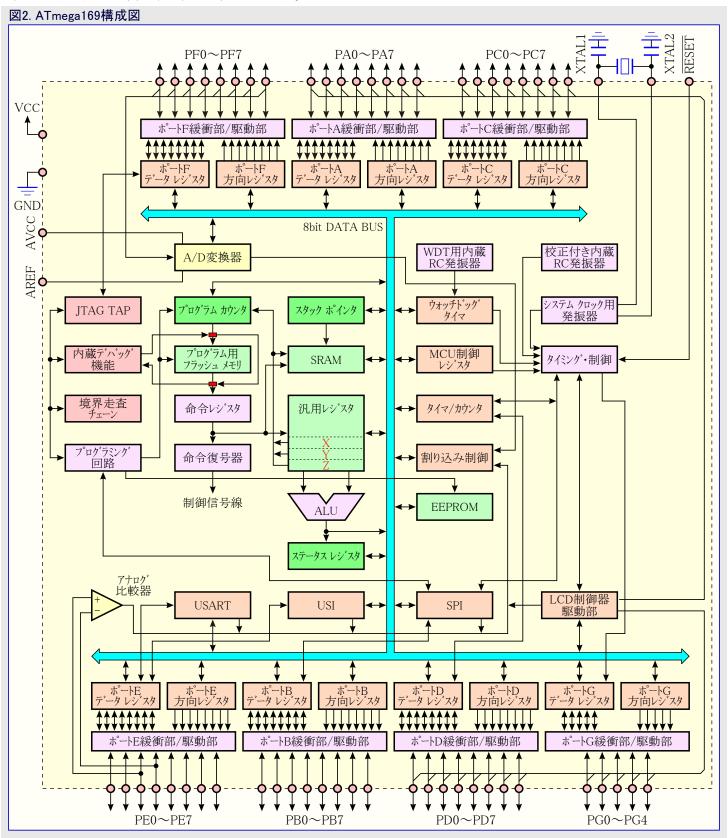


# お断り

本データシート内で示された代表値はシミュレーションと同じ製法技術で製造された他のAVRマイクロ コントローラの特性を基にしています。最小と最大の値はデバイスの特性が記載された後に利用可能になります。

# 概要

ATmega169はAVR RISC構造の低消費CMOS 8ビットマイクロコントローラです。1周期で実行する強力な命令は、MHzあたり1 MIPSにも達し、実行速度対電力消費の最適化が容易に行えます。







AVRは32個の汎用レジスタと豊富な命令群を兼ね備えています。32個の全レジスタはALU(Arithmetic Logic Unit)に直結され、レジスタ間命令は1クロック周期で実行されます。AVR構造は現状のCISC型マイクロコントローラに対し、最大10倍の単位処理量向上効果があります。

ATmega169は次の特徴、書き込み中読める能力を持つ16Kハ・イトの実装書き換え可能なフラッシュメモリと512ハ・イトのEEPROM、1Kハ・イトのSRAM、53本の汎用入出力線、32個の汎用作業レシ・スタ、内蔵デ・ハ・ッグ・とプログラミング・機能、境界走査用のJTAGインターフェース、昇圧回路内蔵の完全なLCD制御器、比較動作を含む柔軟性のある3つのタイマ/カウンタ、内部及び外部割り込み、設定変更可能な直列USART、開始条件検出器付き多用途直列インターフェース(USI)、8チャネルの10ビット A/D変換器、設定変更可能な内部発振器付きウォッチドッグ・タイマ、SPI直列ホート、ソフトウェアで選べる5つの低消費動作機能を提供します。アイドル動作では動作を停止しますが、SRAM、タイマ/カウンタ、SPIホート、割り込み機能は有効で、動作を継続します。ハプーダウン動作ではレシ、スタの内容は保護されますが、発振器が停止するため、以降のハードウェアリセットか割り込みまで他の全機能を禁止(無効に)します。ハプーセーブ動作では非同期タイマとLCD制御器が動作を継続し、デバイスのその他が停止中であっても、基準タイマの継続とLCD表示器の動作を使用者に許します。A/D変換雑音低減動作ではA/D変換中の切り替え雑音を最小とするために、非同期タイマ、LCD制御器、A/D変換器を除く全ての周辺機能とCPUが停止します。スタンバイ動作ではクリスタル発振子/セラミック振動子用発振器が動作し、一方デバイスのその他は休止します。これは低消費電力と非常に速い起動の組み合わせを許します。

本デバイスはAtmelの高密度不揮発性メモリ技術を使って製造されています。内蔵の実装書き換え(ISP)可能なプログラム用フラッシュメモリは規定の不揮発性メモリ書き込み器、SPI直列インターフェース経由、AVRコア上ブートプログラムの実行によって再書き込みができます。プートプログラムは応用領域フラッシュメモリ内の応用プログラムの読み込みにどのインターフェースでも使えます。ブート領域フラッシュメモリ内のプログラムは真の「書き込み中の読み出し可」動作により、応用領域フラッシュメモリ更新中も実行を継続します。モノリシックチップ上の実装書き換え可能なフラッシュメモリと、8ビットRISC型CPUの組み合わせによるAtmel ATmega169は多くの組み込み制御の応用に対して高度な柔軟性と対費用効果をもたらす強力なマイクロコントローラです。

ATmega169 AVRはCコンパイラ、マクロ アセンブラ、デ・バッカ・、シミュレータ、インサーキット エミューレータ、評価キットを含む専用のプログラム及びシステム開発ツールで支援されます。

# ピン概要

### VCC

デジタル電源ピン。

### **GND**

接地ピン。

### PA7~PA0 - (ホ°ートA)

ポートAは(ビット毎に選ばれる)内蔵プルアップ抵抗付き8ビット双方向入出力ポートです。ポートA出力緩衝部は高い吐き出しと吸い込み両方の能力の対称駆動特性です。入力としてプルアップ抵抗が活性(有効)なら、外部的にLowへ引かれたポートAピンは電流を吐き出します。リセット条件が有効になるとクロックが動いていなくてもポートAピンはHi-Zにされます。

ポートAは41頁で一覧されるATmega169の様々な特殊機能も扱います。

### **PB7~PB0** - (ホートB)

ポートBは(ビット毎に選ばれる)内蔵プルアップ抵抗付き8ビット双方向入出力ポートです。ポートB出力緩衝部は高い吐き出しと吸い込み両方の能力の対称駆動特性です。入力としてプルアップ抵抗が活性(有効)なら、外部的にLowへ引かれたポートBピンは電流を吐き出します。リセット条件が有効になるとクロックが動いていなくてもポートBピンはHi-Zにされます。

ポートBは他のポートより優れた駆動能力を持ちます。

ポートBは42頁で一覧されるATmega169の様々な特殊機能も扱います。

### PC7~PC0 - (ポートC)

ポートCは(ビット毎に選ばれる)内蔵プルアップ抵抗付き8ビット双方向入出力ポートです。ポートC出力緩衝部は高い吐き出しと吸い込み両方の能力の対称駆動特性です。入力としてプルアップ抵抗が活性(有効)なら、外部的にLowへ引かれたポートCピンは電流を吐き出します。リセット条件が有効になるとクロックが動いていなくてもポートCピンはHi-Zにされます。

ポートCは44頁で一覧されるATmega169の特殊機能も扱います。

### **PD7~PD0** - (ポートD)

ポートDは(ビット毎に選ばれる)内蔵プルアップ抵抗付き8ビット双方向入出力ポートです。ポートD出力緩衝部は高い吐き出しと吸い込み両方の能力の対称駆動特性です。入力としてプルアップ抵抗が活性(有効)なら、外部的にLowへ引かれたポートDピンは電流を吐き出します。 リセット条件が有効になるとクロックが動いていなくてもポートDピンはHi-Zにされます。

ポートDは45頁で一覧されるATmega169の様々な特殊機能も扱います。

### **PE7~PE0** - (ポートE)

ポートEは(ビット毎に選ばれる)内蔵プルアップ抵抗付き8ビット双方向入出力ポートです。ポートE出力緩衝部は高い吐き出しと吸い込み両方の能力の対称駆動特性です。入力としてプルアップ抵抗が活性(有効)なら、外部的にLowへ引かれたポートEピンは電流を吐き出します。リセット条件が有効になるとクロックが動いていなくてもポートEピンはHi-Zにされます。

ポートEは46頁で一覧されるATmega169の様々な特殊機能も扱います。

### **PF7~PF0** - (ホートF)

ポートFはA/D変換器へのアナログ入力として扱います。

A/D変換器が使われない場合、ポートFは8ビット双方向入出力ポートとしても扱います。ポート ピンは(ビット毎に選ばれる)内蔵プルアップ抵抗を提供できます。ポートF出力緩衝部は高い吐き出しと吸い込みの両能力の対称駆動特性です。入力としてプルアップ抵抗が活性(有効)なら、外部的にLowへ引かれたポートFピンは電流を吐き出します。リセット条件が有効になるとクロックが動いていなくてもポートFピンはHi-Zにされます。JTAGインターフェースが許可されると、リセットが起きてもPF7(TDI)、PF5(TMS)、PF4(TCK)ピンのプルアップ抵抗は活性(有効)にされます。

データを移動出力するTAP状態移行以外、TDOピンはHi-Zにされます。

ポートFはJTAGインターフェース機能も扱います。

### **PG4~PG0** - (ポートG)

ポートGは(ビット毎に選ばれる)内蔵プルアップ抵抗付き5ビット双方向入出力ポートです。ポートG出力緩衝部は高い吐き出しと吸い込み両方の能力の対称駆動特性です。入力としてプルアップ抵抗が活性(有効)なら、外部的にLowへ引かれたポートGピンは電流を吐き出します。リセット条件が有効になるとクロックが動いていなくてもポートGピンはHi-Zにされます。

ポートGは50頁で一覧されるATmega169の様々な特殊機能も扱います。

### RESET

リセット入力。最小パルス幅(25頁の表16.参照)より長いこのピンのLowレベルはクロックが動いていなくてもリセットを生成します。より短いパルスはリセットの生成が保証されません。

### XTAL1

発振器反転増幅器への入力と内部クロック操作回路への入力。

### XTAL2

発振器反転増幅器からの出力。

### **AVCC**

AVCCはポートFとA/D変換器用供給電圧(電源)ピンです。例えA/D変換が使われなくても、外部的にVCCへ接続されるべきです。A/D変換が使われる場合、VCCから低域通過濾波器を通して接続されるべきです。

### **AREF**

AREFはA/D変換器用アナログ基準(電圧)ピンです。

### **LCDCAP**

図98.で示されるように外部コンデンサ(代表値:0.47µF以上)がLCDCAPに接続されなければなりません。このコンデンサはLCD電源(VLC D)用蓄電器として働きます。大容量コンデンサはVLCDのリップルを低減しますが、VLCDが目的値へ達するまでの時間を増加します。

# コード例について

この文書はデバイスの様々な部分の使用法を手短に示す簡単なコート・例を含みます。これらのコート・例はアセンブルまたはコンパイルに先立ってデバイス定義へッタ・ファイルがインクルートされると仮定します。全てのCコンパイラ製造業者がヘッタ・ファイル内にヒット定義を含めるとは限らず、またCでの割り込みの扱いがコンパイラに依存することに注意してください。より多くの詳細についてはCコンパイラの資料で確認してください。

これらのコート・例はアセンブルまたはコンパイルに先立ってデバイス定義ファイルがインクルートされることが前提です。拡張I/O領域に配置したI/Oレジスタに対し、IN, OUT, SBIS, SBIC, CBI, SBI命令は拡張I/O領域へのアクセスを許す命令に置き換えられなければなりません。代表的にはSBRS、SBRC、SBR、CBR命令と組み合わせたLDS、STS命令です。





# AVR CPU 37

### 序説

ここでは一般的なAVRコア構造について説明します。このCPUコアの主な機能は正しいプログラム実行を保証することです。従ってCPU はメモリ アクセス、計算実行、周辺制御、割り込み操作ができなければなりません。

# 構造概要

最大効率と平行処理のため、AVRはプログラムとデータに対してメモリとハンスを分離するハーハート、構造を使います。プログラムメモリ内の命令は単一段のハーイプラインで実行されます。1命令の実行中に次の命令がプログラムメモリから事前取得されます。この概念は全部のクロック周期で命令実行を可能にします。プログラムメモリは実装書き換え可能なフラッシュメモリです。

高速レジスタファイルは1クロック周期アクセスの32個の8ビット長汎用レジスタを含みます。これは1クロック周期ALU(Arithmetic Logic Unit)操作を許します。代表的なALU操作では2つのオペラントがレジスタファイルからの出力で、1クロック周期内でその操作が実行され、その結果がレジスタファイルに書き戻されます。

32個中の6つのレジスタは効率的なアトンス計算ができるデータ空間アトンス指定用に3つの16ビット長間接アトンスポインタ用レジスタとして使用されます。これらアトンスポインタの1つはプログラム用フラッシュメモリ内の定数表参照用アトンスポインタとしても使えます。これら16ビット長付加機能レジスタはX,Y,Zレジスタで、本章内で後述されます。

ALUはレシ、スタ間またはレシ、スタと定数間の算術及び論理操作を支援します。単一レシ、スタ操作もALUで実行できます。算術演算操作後、操作結果についての情報を反映するためにステータスレシ、スタ (SREG)が更新されます。

プログラムの流れは条件/無条件分岐や呼び出し命令によって提供され、全アドレス空間を直接アドレス指定できます。AVR命令の多くは

16ビット語(ワード)形式です。全てのプログラム メモリのアドレスは(訳注:定数のみを除き)16または32ビット長命令を含みます。

プログラム用フラッシュ メモリ空間はブート プログラム領域と応用プログラム領域の2つに分けられます。 どちらの領域にも書き込み禁止や読み書き防止用の専用施錠ビットがあります。 応用フラッシュ メモリ領域内に書き込むSPM命令はブート プログラム領域内に属さ(存在し)なければなりません。

割り込みやサブルーチン呼び出し中、戻りアトレスを示すプログラムカウンタ(PC)はスタックに保存されます。スタックは一般的なデータ用SRAM上に実際には割り当てられ、従ってスタック容量は全SRAM容量とSRAM使用量でのみ制限されます。全ての使用者プログラムはリセット処理ルーチンで(サブルーチン呼び出しや割り込みが実行される前に)、スタック ホーインタ(SP)を初期化しなければなりません。SPはI/O空間で読み書きアクセスが可能です。データ用SRAMはAVR構造で支援される5つの異なるアトレス指定種別を通して容易にアクセスできます。

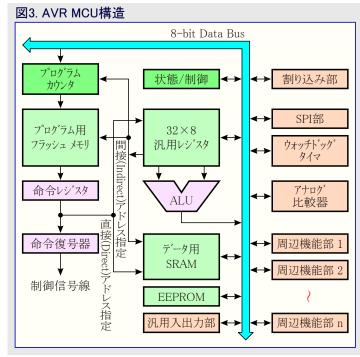
AVR構造に於けるメモリ空間は全て直線的な普通のメモリ配置です。

柔軟な割り込み部にはI/O空間の各制御レシ、スタとステータスレシ、スタ(SREG)の特別な全割り込み許可(I)ビットがあります。全ての割り込みは割り込みへ、クタ表に個別の割り込みへ、クタを持ちます。割り込みには割り込みへ、クタ表の位置に従う優先順があります。下位側割り込みへ、クタアドレスが高い優先順位です。

I/Oメモリ空間は制御レジスタ、SPI、他のI/O機能としてCPU周辺機能用の64アドレスを含みます。I/Oメモリは直接またはレジスタ ファイルの次のデータ空間位置\$20~\$5Fとしてアクセスできます。加えてATmega169にはST/STS/STDとLD/LDS/LDD命令だけ使えるSRAM内の\$60~\$FFに拡張I/O空間があります。

# **ALU** (Arithmetic Logic Unit)

高性能なAVRのALUは32個の全汎用レジスタとの直結で動作します。汎用レジスタ間または汎用レジスタと即値間の演算操作は単一クロック周期内で実行されます。ALU操作は算術演算、論理演算、ビット操作の3つの主な種類に大別されます。符号付きと符号なし両方の乗算と固定小数点形式を支援する乗算器(乗算命令)も提供する構造の実装(製品)もあります。詳細記述については「命令要約」章をご覧ください。



# ステータス レジ スタ (Status Register) SREG

ステータスレジスタは最も直前に実行した演算命令の結果についての情報を含みます。この情報は条件処理を行うためのプログラムの流れ変更に使えます。ステータスレジスタは「命令一式手引書」で詳述したように、全てのALU操作後、更新されることに注目してください。これは多くの場合でそれ用の比較命令使用の必要をなくし、高速でより少ないコードに帰着します。

ステータス レシ スタは割り込み処理ルーチン移行時の保存と割り込みからの復帰時の回復(復帰)が自動的に行われません。これはソフトウェアによって扱われなければなりません。

AVRのステータス レシ、スタ(SREG)は次のように定義されます。

ピット	7	6	5	4	3	2	1	0	_
\$3F (\$5F)	I	Т	Н	S	V	N	Z	С	SREG
Read/Write	R/W								
初期値	0	0	0	0	0	0	0	0	

### • ビット7 - I: 全割り込み許可 (Global Interrupt Enable)

全割り込み許可ビットは割り込みが許可されるために設定(1)されなければなりません。その時に個別割り込み許可制御は独立した制御レジスタで行われます。全割り込み許可ビットが解除(0)されると、個別割り込み許可設定に拘らず、どの割り込みも許可されません。I ビットは割り込みが起こった後にハードウェアによって解除(0)され、後続の割り込みを許可するために、RETI命令によって設定(1)されます。Iビットは「命令一式手引書」で記述されるようにSEIやCLI命令で応用(プログラム)によって設定(1)や解除(0)もできます。

• **ビット6** - T: **ビット変数** (Bit Copy Storage)

ビット複写命令、BLD(Bit LoaD)とBST(Bit STore)は操作したビットの転送元または転送先として、このTビットを使います。レジスタ ファイルのレジスタからのビットはBST命令によってTに複写でき、TのビットはBLD命令によってレジスタ ファイルのレジスタ内のビットに複写できます。

• ビット5 - H: ハーフキャリー フラク (Half Carry Flag)

ハーフキャリー(H)フラグはいくつかの算術操作でのハーフキャリーを示します。ハーフキャリーはBCD演算に有用です。詳細情報については「命令要約」記述をご覧ください。

• ビット4 - S: 符号 (Sign Bit, S= N Ex-OR V)

Sフラグは常に負(N)フラグと2の補数溢れ(V)フラグの排他的論理和です。詳細情報については「命令要約」記述をご覧ください。

• ビット3 - V : 2の補数溢れフラグ (2's Complement Overflow Flag)

2の補数溢れ(V)フラグは2の補数算術演算を支援します。詳細情報については「命令要約」記述をご覧ください。

• **L'yh2** - **N** : **負フラク** (Negative Flag)

負(N)フラグは算術及び論理操作での負の結果(MSB=1)を示します。詳細情報については「命令要約」記述をご覧ください。

• **ビット1 - Z**: **ゼロ フラグ** (Zero Flag)

ゼロ(Z)フラグは算術及び論理操作でのゼロ(0)の結果を示します。詳細情報については「**命令要約**」記述をご覧ください。

• ビット0 - C: キャリー フラク (Carry Flag)

キャリー(C)フラグは算術及び論理操作でのキャリー(またはボロー)を示します。詳細情報については「**命令要約**」記述をご覧ください。





# 汎用レジスタファイル

このレシ、スタファイルはAVRの増強したRISC命令群用に最適化されています。必要な効率と柔軟性を達成するために、次の入出力機構がレシ、スタファイルによって支援されます。

- 1つの8ビット出力オペランドと1つの8ビットの結果入力
- 2つの8ビット出力オペラントと1つの8ビットの結果入力
- 2つの8ビット出力オペランドと1つの16ビットの結果入力
- 1つの16ビット出力オペランドと1つの16ビットの結果入力

図4.はCPU内の32個の汎用作業レジスタの構造を示します。

レジスタファイルを操作する殆どの命令は全てのレジスタに直接アクセスし、それらの殆どは単一周期命令です。

図4.で示されるように各レシ、スタは使用者データ空間の最初の32位置へ直接的に配置することで、それらはデータメモリアト・レスも割り当てられます。例え物理的にSRAM位置として実装されていなくてもX,Y,Zレシ、スタ(ポーペンタ)がレシ、スタファイル内のどのレシ、スタの指示にも設定できるように、このメモリ構成は非常に柔軟なレシ、スタのアクセスを提供します。

図4. AV	R CPU 汎用レジスタ構タ	<b></b>	
	7 0	アドレス	
	R0	\$00	
	R1	\$01	
	R2	\$02	
	}		
	R13	\$0D	
	R14	\$0E	
λπ Œ	R15	\$0F	
汎用 レシブスタ	R16	\$10	
ファイル	R17	\$11	
2 7 17	}		
	R26	\$1A	VI パイカー 下位バイト
	R27	\$1B	Xレジスター・下位バイト 上位バイト
	R28	\$1C	Yレジスタ 下位バイト
	R29	\$1D	上ルハイト
	R30	\$1E	フレジスター 下位バイト
	R31	\$1F	上位バイト

# Xレシ、スタ、Yレシ、スタ、Zレシ、スタ

R26~R31レシ、スタには通常用途の使用にいくつかの追加機能があります。これらのレシ、スタはデータ空間の間接アト・レス指定用の16ビットアト・レスポインタです。3つのX,Y,Z間接アト・レスレシ、スタは**図**5.で記載したように定義されます。

種々のアドレス指定種別で、これらのアドレス レジスタは固定変位、 自動増加、自動減少としての機能を持ちます(詳細については 「**命令一式手引書**」をご覧ください)。

図5. X,Y,Zレシ	バスタ樟	<b></b>			
	15	XH (上位)		XL (下位)	0
X レジスタ	7	R27 (\$1B)	0 7	R26 (\$1A)	0
	15	YH (上位)		YL (下位)	0
Y レジスタ	7	R29 (\$1D)	0 7	R28 (\$1C)	0
	15	ZH (上位)		ZL (下位)	0
Z レシ <sup>・</sup> スタ	7	R31 (\$1F)	0 7	R30 (\$1E)	0

# スタック ホ°インタ (Stack Pointer) SPH.SPL (SP)

スタックは主に一時データの保存、局所変数の保存、割り込みとサブルーチン呼び出し後の戻りアドレスの保存に使われます。スタック ポインタ レジスタは常にこのスタックの先頭(<mark>訳注</mark>:次に使われるべき位置)を指し示します。スタックが高位メモリから低位メモリへ伸長するように実行されることに注意してください。これはスタックへのPUSH命令はスタック ポインタを減少するという意味です。

スタック ポインタはサブルーチンや割り込みのスタックが配置されるデータSRAMのスタック領域を指し示します。データSRAM内のスタック空間はサブルーチン呼び出しの実行や割り込みの許可の何れにも先立ってプログラムによって定義されなければなりません。スタック ポインタは \$100以上を指示するように設定されなければなりません。スタック ポインタは PUSH命令でデータがスタックに格納されると-1され、サブルーチン呼び出しや割り込みで戻りアトレスがスタックに格納されると-2されます。スタック ポインタは POP命令でデータがスタックから引き出されると+1され、サブルーチンからの復帰(RET)命令や割り込みからの復帰(RETI)命令でアトレスがスタックから引き出されると+2されます。

AVRのスタック ポインタはI/O空間内の2つの8ビット レジスタとして実装されます。実際に使われるビット数は(そのデバイス)実装に依存します。SPLだけが必要とされる程に小さいAVR構造の実装(デバイス)のデータ空間もあることに注意してください。この場合、SPHレジスタは存在しません。

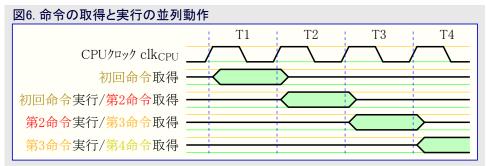
ピット	15	14	13	12	11	10	9	8	_
\$3E (\$5E)	-	-	-	-	-	SP10	SP9	SP8	SPH
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	
ピット	7	6	5	4	3	2	1	0	_
\$3D (\$5D)	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	SPL
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

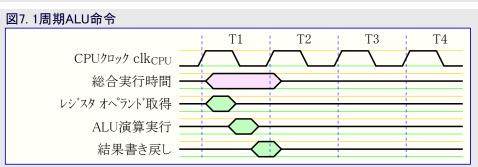
# 命令実行タイミング

本項は命令実行の一般的なアクセス タイミング の概念を記述します。AVR CPUはチップ(デバイス)用に選んだクロック元から直接的に生成したCPUクロック(clkCPU)によって駆動されます。内部クロック分周は使われません。

図6.はハーハート・構造と高速アクセスレジスタファイルの概念によって可能とされる並列の命令取得と命令実行を示します。これは機能対費用、機能対クロック、機能対電源部についての好結果と対応するMHzあたり1 MIPSを達成するための基本的なパイプラインの概念です。

図7.はレシ、スタファイルに対する内部タイミングの概念を示します。単一クロック周期で2つのレジスタオペラント、を使うALU操作が実行され、その結果が転送先レジ、スタへ書き戻されます。





# リセットと割り込みの扱い

AVRは多くの異なる割り込み元を提供します。これらの割り込みと独立したリセット ベクタ各々はプログラム メモリ空間内に独立したプログラム ベクタを持ちます。全ての割り込みは割り込みを許可するために、ステータス レジスタ(SREG)の全割り込み許可(I)ビットと共に論理1が書かれなければならない個別の許可ビットを割り当てられます。BLB02またはBLB12 ブート施錠ビットがプログラム(0)されると、プログラム カウンタ値によっては割り込みが自動的に禁止されるかもしれません。この特質はソフトウェア保護を改善します。詳細については175頁の「メモリ プログラミング」章をご覧ください。

既定でのプログラムメモリ空間の最下位アトレスはリセットと割り込みのヘブクタとして定義されます。ヘブクの完全な一覧は30頁の「割り込み」で示されます。この一覧は各種割り込みの優先順位も決めます。下位側アトレスがより高い優先順位です。リセットが最高優先順位で次が外部割り込み要求の(INTO)です。割り込みヘブクタはMCU制御レシ、スク(MCUCR)の割り込みヘブク選択(IVSEL)ヒ、ットの設定(1)によってブートフラッシュ領域先頭へ移動できます。より多くの情報については30頁の「割り込み」を参照してください。リセットヘブクもBOOTRSTヒューズのプログラム(0)によってブートフラッシュ領域先頭へ移動できます。166頁の「ブートローダ・支援 - 書き込み中読み出し可能な自己プログラミング」をご覧ください。

割り込みが起こると全割り込み許可(I)ビットが解除(0)され、全ての割り込みは禁止されます。使用者ソフトウェアは多重割り込みを許可するため、全割り込み許可(I)ビットへ論理1を書けます。その後全ての許可した割り込みが現在の割り込みルーチンで割り込めます。全割り込み許可(I)ビットは割り込みからの復帰(RETI)命令が実行されると、自動的に設定(1)されます。

根本的に2つの割り込み形式があります。1つ目の形式は割り込み要求フラグを設定(I)する事象によって起動されます。これらの割り込みでは割り込み処理ルーチンを実行するために、プログラム カウンタは対応する現実の割り込みへ、クタを指示し、ハートウェアが対応する割り込み要求フラグを解除(0)します。割り込み要求フラグは解除(0)されるべきフラグのビット位置へ論理1を書くことによっても解除(0)できます。対応する割り込み許可ビットが解除(0)されている間に割り込み条件が起こると、割り込み要求フラグが設定(1)され、割り込みが許可されるか、またはこのフラグがソフトウェアによって解除(0)されるまで記憶(保持)されます。同様に、全割り込み許可(I)ビットが解除(0)されている間に1つまたはより多くの割り込み条件が起こると、対応する割り込み要求フラグが設定(1)されて全割り込み許可(I)ビットが設定(1)されるまで記憶され、その(I=1)後で優先順に従って実行されます。

2つ目の割り込み形式は割り込み条件が存在する限り起動し(続け)ます。これらの割り込みは必ずしも割り込み要求フラグを持っているとは限りません。割り込みが許可される前に割り込み条件が消滅すると、この割り込みは起動されません。

AVRが割り込みから抜け出すと常に主プログラムへ戻り、何れかの保留割り込みが扱われる前に1つ以上の命令を実行します。

ステータス レジスタ(SREG)は割り込みルーチンへ移行時の保存も、復帰時の再設定も自動的に行われないことに注意してください。これはソフトウェアによって扱われなければなりません。





割り込みを禁止するためにCLI命令を使うと、割り込みは直ちに禁止されます。CLI命令と同時に割り込みが起こっても、CLI命令後に割り込みは実行されません。次例は時間制限EEPROM書き込み手順中に割り込みを無効とするために、これがどう使えるかを示します。

アセンブリ言語プログラム例		
IN R16, SREG CLI SBI EECR, EEM SBI EECR, EEW OUT SREG, R16	IWE /E	;ステータス レジスタを保存 ;EEPROM書き込み手順中割り込み禁止 ;EEPROM主書き込み許可 ;EEPROM書き込み開始 ;ステータス レジスタを復帰
C言語プログラム例		
char cSREG; cSREG = SREG; disable_inter EECR  = (1< <eew EECR  = cSREG:</eew 	WE);	/* ステータス レジ・スタ保存変数定義 */ /* ステータス レジ・スタを保存 */ /* EEPROM書き込み手順中割り込み禁止 */ /* EEPROM主書き込み許可 */ /* EEPROM書き込み開始 */ /* ステータス レジ・スタを復帰 */

割り込みを許可するためにSEI命令を使うと、次例で示されるようにどの保留割り込みにも先立ってSEI命令の次の命令が実行されます。

アセンブリ言語プログラム例	
SEI SLEEP	;全割り込み許可;休止形態移行(割り込み待ち)
C言語プログラム例	
enable_interrupt(); sleep();	/* 全割り込み許可 */ /* 休止形態移行 (割り込み待ち) */

注: SLEEP命令までは割り込み禁止、保留割り込み実行前に休止形態へ移行します。

### 割り込み応答時間

許可した全てのAVR割り込みに対する割り込み実行応答は最小4クロック周期です。4クロック周期後、実際の割り込み処理ルーチンに対するプログラム ベクタ アドレスが実行されます。この4クロック周期時間中にプログラム カウンタ(PC)がスタック上に保存(プッシュ)されます。このベクタは標準的に割り込み処理ルーチンへの無条件分岐で、この分岐は3クロック周期要します。複数周期命令実行中に割り込みが起こると、その割り込みが扱われる前に、この命令が完了されます。MCUが休止形態の時に割り込みが起こると、割り込み実行応答時間は4クロック周期増やされます。この増加は選んだ休止形態からの起動時間に加えてです。

割り込み処理ルーチンからの復帰は4クロック周期要します。これらの4クロック周期中、プログラム カウンタ(PC:2バイト)がスタックから取り戻され(ポップ)、スタック ポープンタは増加され(+2)、ステータス レシ、スタ(SREG)の全割り込み許可(I)ビットが設定(1)されます。

# AVR ATmega169のメモリ

この項はATmega169の各種メモリを記述します。AVR構造にはプログラム メモリ空間とデータ メモリ空間の2つの主なメモリ空間があります。加 えてATmega169はデータ保存用EEPROMメモリが特徴です。3つのメモリ空間全ては一般的な直線的アドレスです。

# 実装書き換え可能なプログラム用フラッシュメモリ

ATmega169はプログラム保存用に実装書き換え可能な16Kバイトのフラッシュ メモリをチップ 上に含 みます。全てのAVR命令が16または32ビット幅のため、このフラッシュ メモリは8K×16ビットとして 構成されます。ソフトウェア保護のため、フラッシュ プログラム メモリ空間はブート プログラム領域と応用プ ログラム領域の2つに分けられます。

フラッシュ メモリは最低10,000回の消去/書き込み回数の耐久性があります。ATmega169のプロ グラム カウンタ(PC)は13ビット幅、従って8Kプログラム メモリ位置のアドレス指定です。 ブート プログラム 領域の操作と関係するソフトウェア保護用ブート施錠ビットは166頁の「ブート ローダ支援 - 書き込み 中読み出し可能な自己プログラミング」の詳細で記述されます。175頁の「メモリ プログラミング」は SPI、JTAG、並列プログラミング動作でのフラッシュメモリプログラミングの詳細な記述を含みます。

定数表は全てのプログラム メモリ アトレス空間に配置できます。(LPM命令記述参照)

命令の取得と実行のタイミング図は9頁の「命令実行タイミング」で示されます。

# 図8. プログラム メモリ配置図 \$0000 応用プログラム用 フラッシュ メモリ $(7K \sim 7.875K) \times 16$ ブートプログラム用 フラッシュ メモリ $(128 \sim 1 \text{K}) \times 16$ \$1FFF

# データ用SRAM メモリ

図9.はATmega169のSRAMメモリ構成方法を示します。

ATmega169はINやOUT命令で予約した64位置で支援されるより多くの周辺機能部を持つ 複合マイクロ コントローラです。SRAM(データ空間)内\$60~\$FFの拡張I/O空間に対してLD/LDS/ LDDとST/STS/STD命令だけが使えます。

下位1280データメモリ位置はレシ、スタファイル、I/Oメモリ、拡張I/Oメモリ、データ用内蔵SRAMに充て ます。先頭の32位置はレシブスタファイル、次の64位置は標準I/Oメモリ、その次の160位置は拡張 I/Oメモリ、そして次の1024位置はデータ用内蔵SRAMに充てます。

直接、間接、変位付き間接、事前減少付き間接、事後増加付き間接の5つの異なるアドレス指 定種別でデータ メモリ(空間)を網羅します。レジスタファイル内のレシブスタR26~R31は間接アトブレス指 定ポインタ用レジスタが特徴です。

直接アドレス指定はデータ空間全体に届きます。

変位付き間接動作はYまたはZレシ、スタで与えられる基準アドレスからの63アドレス位置に届きま す。

自動の事前減少付きと事後増加付きのレシ、スタ間接アト、レス指定動作を使う時に(使われる)X, 注: 赤字はI/Oアト、レス Y,Zアドレス レジスタは減少(-1)または増加(+1)されます。

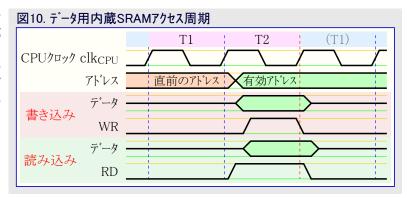
		アトレス
レシ <sup>*</sup> スタ	R0	\$0000
ファイル	∼	~
(32×8)	R31	\$001F
I/O	\$00	\$0020
レシ゛スタ	~	~
(64×8)	\$3F	\$005F
拡張I/O	\$0060	\$0060
レジスタ	~	~
(160×8)	\$00FF	\$00FF
内蔵	\$0100	\$0100
SRAM	~	~
(1024×8)	\$04FF	\$04FF

ATmega169の32個の汎用レジスタ、64個のI/Oレジスタ、160個の拡張I/Oレジスタ、1024小小のデータ用内蔵SRAMは、これら全てのアドレス 指定種別を通して全部アクセスできます。レジスタファイルは8頁の「汎用レジスタファイル」で記述されます。

# データ メモリ アクセス タイミング

この項は内部メモリ アクセスに対する一般的なアクセス タイミングの 概念を記述します。データ用内蔵SRAMアクセスは図10.で記載 されるように2 clkCPU周期で実行されます。

(訳注) 内蔵SRAMのアクセスを含む代表的な命令はT1.T2の 2周期で実行され、T1で対象アドレスを取得/(算出)/ 確定し、T2で実際のアクセスが行われます。後続する (T1)は次の命令のT1です。





# データ用EEPROMメモリ

ATmega169は512小小のデータEEPROMを含みます。それは単一小小が読み書きできる分離したデータ空間として構成されます。EEP ROMは最低100,000回の消去/書き込み回数の耐久性があります。CPUとEEPROM間のアクセスは以降のEEPROMアトレスレジスタ、EE PROMデータレジスタ、EEPROM制御レジスタで詳細に記述されます。

EEPROMへのSPI、JTAG、並列プログラミングの詳細記述については各々185頁、187頁、177頁をご覧ください。

### **EEPROMアクセス**

EEPROMアクセスレジ、スタはI/O空間でアクセス可能です。

EEPROMの書き込み(<mark>訳注</mark>:原文はアクセス)時間は13頁の表1.で与えられます。(書き込みは)自己タイシンが機能ですが、使用者ソフトウェアは次ハーイが書ける時を検知してください。使用者コートがEEPROMに書く命令を含む場合、いくつかの予防処置が取られなければなりません。厳重に濾波した電源では電源投入/切断でVCCが緩やかに上昇または下降しそうです。これはデバイスが何周期かの間、使われるクロック周波数に於いて最小として示されるより低い電圧で走行する原因になります。これらの状態で問題を避ける方法の詳細については14頁の「EEPROMデータ化けの防止」をご覧ください。

予期せぬEEPROM書き込みを防止するため、特別な書き込み手順に従わなければなりません。この詳細については「EEPROM制御レジスタ」の記述を参照してください。

EEPROMが読まれると、CPUは次の命令が実行される前に4クロック周期停止されます。EEPROMが書かれると、CPUは次の命令が実行される前に2クロック周期停止されます。

### EEPROMアドレス レジスタ (EEPROM Address Register) EEARH.EEARL (EEAR)

			O		•	, ,			
ピット	15	14	13	12	11	10	9	8	
\$22 (\$42)	_	-	-	-	-	-	-	EEAR8	EEARH
Read/Write	R	R	R	R	R	R	R	R/W	
初期値	0	0	0	0	0	0	0	不定	
ピット	7	6	5	4	3	2	1	0	
\$21 (\$41)	EEAR7	EEAR6	EEAR5	EEAR4	EEAR3	EEAR2	EEAR1	EEAR0	EEARL
Read/Write	R/W								
初期値	不定								

### • ビット15~9 - Res: 予約 (Reserved)

これらのビットは予約されており、常に0として読みます。

### • ビット8~0 - EEAR8~0: EEPROMアトレス (EEPROM Address)

EEPROMアドレス レジスタ(EEARHとEEARL)は512バイトEEPROM空間のEEPROMアドレスを指定します。EEPROMデータ バイトは0~511間で直線的に配されます。EEARの初期値は不定です。EEPROMがアクセスされるであろう前に適切な値が書かれねばなりません。

# EEPROM T ータレシ スタ (EEPROM Data Register) EEDR

ピット	7	6	5	4	3	2	1	0	_
\$20 (\$40)	(MSB)							(LSB)	EEDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

# • ビット7~0 - EEDR7~0 : EEPROMデータ (EEPROM Data)

EEPROM書き込み操作に対してEEDRはEEPROMアドレスレジスタ(EEAR)で与えたアドレスのEEPROMへ書かれるべきデータを含みます。 EEPROM読み込み操作に対してEEDRはEEARで与えたアドレスのEEPROMから読み出したデータを含みます。

# EEPROM制御レジスタ (EEPROM Control Register) EECR

ピット	7	6	5	4	3	2	1	0	
\$1F (\$3F)	-	-	-	-	EERIE	EEMWE	EEWE	EERE	EECR
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	不定	0	

# • ビット7~4 - Res: 予約 (Reserved)

これらのビットは予約されており、常に0として読みます。

# • ビット3 - EERIE: EEPROM操作可割り込み許可 (EEPROM Ready Interrupt Enable)

EERIEの1書き込みはステータス レジスタ(SREG)の全割り込み許可(I)ビットが設定(1)されているなら、EEPROM操作可割り込みを許可します。EERIEの0書き込みはこの割り込みを禁止します。EEPROM操作可割り込みはEEWEが解除(0)されていると継続する割り込みを発生します。

### • ビット2 - EEMWE: EEPROM主書き込み許可(EEPROM Master Write Enable)

EEMWEL'ットはEEPROM書き込み許可(EEWE)ビットの1設定がEEPROM書き込みの原因となるかどうかを決めます。EEMWEが設定 (1)されるとき、EEWEの1設定は選んだアドレスのEEPROMにデータを書きます。EEMWEが0の場合、EEWEの1設定は無効です。EEMW Eがソフトウェアによって設定(1)されてしまうと、4クロック周期後にハードウェアがこのビットを0に解除します。EEPROM書き込み手順については「書き込み許可(EEWE)ビット」の記述をご覧ください。

### • ビット1 - EEWE: EEPROM書き込み許可(EEPROM Write Enable)

EEPROM書き込み許可信号(EEWE)はEEPROMへの書き込みストローブです。アドレスとデータが正確に設定されると、EEPROMへその値を書くためにEEWEビットは1を書かれなければなりません。論理1がEEWEへ書かれる前にEEPROM主書き込み許可(EEMWE)ビットは1を書かれなければならず、さもなければEEPROM書き込みは行われません。EEPROMを書くとき、次の手順に従うべきです(手順3.と4.の順番は重要ではありません)。

- 1. EEPROM書き込み許可(EEWE)ビットがOになるまで待ちます。
- 2. SPM命令制御/状態レジスタ(SPMCSR)のSPM操作許可(SPMEN)ビットが0になるまで待ちます。
- 3. 今回のEEPROMアドレスをEEPROMアドレスレジスタ(EEAR)に書きます。(任意、省略可)
- 4. 今回のEEPROMデータをEEPROMデータレジスタ(EEDR)に書きます。(任意、省略可)
- 5. EEPROM制御レジスタ(EECR)のEEPROM主書き込み許可(EEMWE)ビットに1を、EEPROM書き込み許可(EEWE)ビットに0を同時に書きます。
- 6. EEMWEL \* ット設定後4クロック周期内にEEPROM書き込み許可(EEWE) と、ットへ論理1を書きます。

CPUがフラッシュ メモリ書き込み中、EEPROMはプログラム(書き込みが)できません。ソフトウェアは新規EEPROM書き込みを始める前にフラッシュ メモリのプログラミングが完了されていることを検査しなければなりません。2.はソフトウェアがフラッシュ メモリをプログラム(書き込みを)することをCPUに許すブート ローダを含む場合だけ関係します。フラッシュ メモリが決してCPUによって更新されないなら、2.は省略できます。ブートプログラミングについての詳細に関しては166頁の「ブートローダ支援 - 書き込み中読み出し可能な自己プログラミング」をご覧ください。

警告: 手順5.と6.間の割り込みはEEPROM主書き込み許可が時間超過するため、書き込み周期失敗になります。EEPROMをアクセスする割り込みルーチンが他のEEPROMアクセスを中断し、EEARかEEDRが変更されると、中断したEEPROMアクセスを失敗させます。これらの問題を避けるため、全ての手順中、ステータスレジ、スタ(SREG)の全割り込み許可(I)ビットは解除(0)されていることが推奨されます。

書き込みアクセス時間が経過されると、EEPROM書き込み許可(EEWE)ビットはハードウェアによって解除(0)されます。使用者ソフトウェアは次バイトを書く前にこのビットをポーリングし、0を待つことができます。EEWEが設定(1)されてしまうと、次の命令が実行される前にCPUは2周期停止されます。

### • ビット0 - EERE: EEPROM読み込み許可 (EEPROM Read Enable)

EEPROM読み込み許可信号(EERE)はEEPROMへの読み込みストローブです。EEARに正しいアトレスが設定されると、EEPROM読み出しを起動するためにEEREはずりは論理しを書かれなければなりません。EEPROM読み出しアクセスは(その)1命令で行われ、要求したデータは直ちに利用できます。EEPROMが読まれるとき、次の命令が実行される前にCPUは4周期停止されます。

使用者は読み込み操作を始める前にEEWEビットをポーリングすべきです。書き込み操作が実行中の場合、EEPROMアドレス レジスタ(EE AR)の変更もEEPROM読み込みもできません。

EEPROMアクセスの時間に校正済み内蔵RC発振器が使われます。表1.はCPUからのEEPROMアクセスに対する代表的な書き込み時間を示します。

表1. EEPROM書き込み時間	1	
項目	校正付き内蔵RC 発振器周期数	代表
EEPROM書き込み(CPU)	27072	3.3ms





次のコート・例はアセンブリ言語とC言語でのEEPROM書き込み関数を示します。本例は(例えば全割り込み禁止によって)割り込みが制御され、これらの関数実行中に割り込みが起きない前提です。本例はソフトウェア内にフラッシュ ブート ロータ・が無い前提でもあります。そのようなコート・が存在する場合、EEPROM書き込み関数は何れかが実行するSPM命令の完了も待たねばなりません。

```
アセンブリ言語プログラム例
                                                   ;EEPROM書き込み完了ならばスキップ
EEPROM_WR:
           SBIC
                 EECR, EEWE
                 EEPROM WR
                                                   ;以前のEEPROM書き込み完了まで待機
           RJMP
           OUT
                 EEARH, R18
                                                   ;EEPROMアドレス上位バイト設定
           OUT
                 EEARL, R17
                                                   ;EEPROMアドレス下位バイ設定
           OUT
                 EEDR, R16
                                                   ;EEPROM書き込み値を設定
                                                   ;EEPROM主書き込み許可ビット設定
                 EECR, EEMWE
           SBT
                 EECR, EEWE
                                                   ;EEPROM書き込み開始(書き込み許可ビット設定)
           SBT
                                                   ;呼び出し元へ復帰
           RET
C言語プログラム例
void EEPROM_write(unsigned int uiAddress, unsigned char ucData)
   while (EECR & (1<<EEWE));
                                                   /* 以前のEEPROM書き込み完了まで待機 */
                                                   /* EEPROMアドレス設定 */
   EEAR = uiAddress;
   EEDR = ucData;
                                                   /* EEPROM書き込み値を設定 */
   EECR \mid = (1 << EEMWE);
                                                   /* EEPROM主書き込み許可*/
   EECR = (1 << EEWE);
                                                   /* EEPROM書き込み開始 */
}
```

次のコート・例はアセンブリ言語とC言語でのEEPROM読み込み関数を示します。本例は割り込みが制御され、これらの関数実行中に割り込みが起きない前提です。

```
アセンブリ言語プログラム例
EEPROM RD:
          SBIC
                 EECR, EEWE
                                                  ;EEPROM書き込み完了ならばスキップ
                 EEPROM RD
                                                  ;以前のEEPROM書き込み完了まで待機
          RJMP
          OUT
                                                  ;EEPROMアドレス上位バイト設定
                 EEARH, R18
                                                  ;EEPROMアドレス下位バイト設定
          OUT
                 EEARL, R17
          SBI
                 EECR, EERE
                                                  ;EEPROM読み出し開始(読み込み許可ビット設定)
          IN
                 R16, EEDR
                                                  ;EEPROM読み出し値を取得
          RET
                                                  ;呼び出し元へ復帰
C言語プログラム例
unsigned char EEPROM_read(unsigned int uiAddress)
   while (EECR & (1<<EEWE));
                                                  /* 以前のEEPROM書き込み完了まで待機 */
                                                  /* EEPROMアドレス設定*/
   EEAR = uiAddress;
   EECR |= (1<<EERE);
                                                  /* EEPROM読み出し開始 */
   return EEDR;
                                                  /* EEPROM読み出し値を取得,復帰 */
```

### パワーダウン休止動作中のEEPROM書き込み

EEPROM書き込み動作が活動中にパワーダウン休止動作へ移行すると、EEPROM書き込み動作が継続し、EEPROM書き込み時間が 過ぎ去ってしまう前に完了します。しかし、書き込み動作が完了されると、発振器が動作を継続し、結果としてデバイスはパワーダウン動作へ完全に移行しません。従ってパワーダウンへ移行する前に、EEPROM書き込み動作が完了される(EEWE=0)ことの確認が推奨されます。(訳補: パワーダウン移行後もEPROM書き込みは正常に完了するが、その後発振器が止まらないことの注意)

# EEPROMデータ化けの防止

低VCCの期間中、正しく動作するための供給電圧がCPUとEEPROMに対して低すぎるためにEEPROMデータが化け得ます。これらの問題はEEPROMを使う基板段階の装置と同じで、同じ設計上の解決策が適用されるべきです。

EEPROMデータ化けは電圧が低すぎる時の2つの状態によって起こされ得ます。1つ目として、EEPROMへの通常の書き込み手順は正しく動作するための最低電圧が必要です。2つ目として、供給電圧が低すぎると、CPU自身が命令を間違って実行し得ます。

EEPROMデータ化けは次の推奨設計によって容易に避けられます。

不充分な供給電源電圧の期間中、AVRのRESETを活性(Low)に保ってください。これは内蔵低電圧検出器(BOD)を許可することによって行えます。内蔵BODの検出電圧が必要とした検出電圧と一致しない場合、外部低VCCリセット保護回路が使えます。書き込み動作実行中にリセットが起こると、この書き込み操作は供給電源電圧が充分ならば(継続)完了されます。

# I/O メモリ (レジスタ)

ATmega169のI/O空間定義は221頁の「レジスタ要約」で示されます。

ATmega169の全てのI/Oと周辺機能はI/O空間に配置されます。全てのI/O位置はI/O空間と32個の汎用作業レシ、スタ間のデータ転送を行うLD/LDS/LDD命令とST/STS/STD命令によってアクセスされます。アドレス範囲\$00~\$1F内のI/Oレシ、スタはSBI命令とCBI命令の使用で直接的にビットアクセス可能です。これらのレシ、スタではSBISとSBIC命令の使用によって単一ビット値が検査できます。より多くの詳細については「命令要約」章を参照してください。I/O指定命令INとOUTを使う時はI/Oアドレス\$00~\$3Fが使われなければなりません。LD命令とST命令を使い、データ空間としてI/Oレシ、スタをアクセスする時はこれらのアドレスに\$20が加算されなければなりません。

ATmega169はINやOUT命令で予約した64位置で支援されるより多くの周辺機能部を持つ複合マイクロコントローラです。SRAM(データ空間)内\$60~\$FFの拡張I/O領域に対してはLD/LDS/LDDとST/STS/STD命令だけが使えます。

将来のデバイスとの共通性を保つため、アクセスされる場合、予約ピットはOが書かれるべきです。予約済みI/Oメモリ アドレスは決して書かれるべきではありません。

状態フラグのいくつかはそれらへ論理1を書くことによって解除(0)されます。CBIとSBI命令は他の多くのAVRの様ではなく、指定ビットだけを操作し、従って状態フラグのようなものを含むレジスタに使えることに注意してください。CBIとSBI命令は(I/Oアドレス)\$00~\$1Fのレジスタでのみ動作します。

I/Oと周辺制御レジスタは以降の項で説明されます。

### 汎用I/Oレジスタ

ATmega169は3つの汎用I/Oレジスタを含みます。これらのレジスタはどの情報の格納にも使え、特に全体変数や状態フラグの格納に有用です。(I/O)アドレス範囲\$00~\$1Fの汎用I/OレジスタはSBI,CBI,SBIS,SBIC命令の使用で直接ビット アクセスが可能です。

# 汎用I/Oレジスタ2 (General Purpose I/O Register 2) GPIOR2

\$2B (\$4B) (MSB) (LSB) <b>GPIC</b>	6 5 4 3 2 1	ピット	0	
		\$2B (\$4B)	(LSB)	GPIOR2
Read/Write R/W R/W R/W R/W R/W R/W R/W	R/W R/W R/W R/W R/W		R/W	
初期値 0 0 0 0 0 0 0 0	0 0 0 0 0	初期値	0	

# 汎用I/Oレジスタ1 (General Purpose I/O Register 1) GPIOR1

ピット	7	6	5	4	3	2	1	0	
\$2A (\$4A)	(MSB)							(LSB)	GPIOR1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

### 汎用I/OレジスタO (General Purpose I/O Register 0) GPIORO

ピット	7	6	5	4	3	2	1	0	
\$1E (\$3E)	(MSB)							(LSB)	GPIOR0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

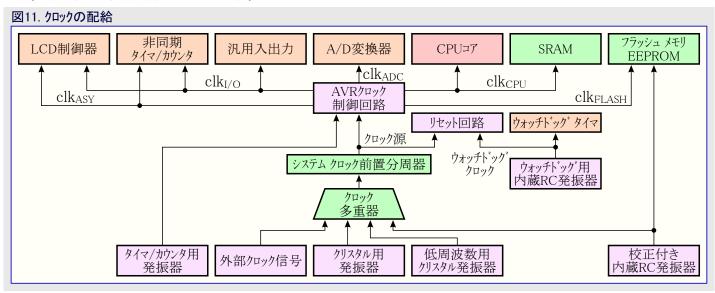




# システム クロックとクロック選択

# クロック系統とその配給

図11.はAVR内の主要なクロック系統とその配給を示します。全てのクロックが与えられた時間有効である必要はありません。消費電力低減のため、21頁の「電力管理と休止形態」で記述される各種休止形態の使用によって、使われない部分のクロックを停止することができます。クロック系統は以下で詳細に示されます。



# CPU クロック - clk<sub>CPU</sub>

CPUクロックはAVRコアの動作と関係する系統の部分に配給されます。このような部分の例は汎用レジスタ ファイル、ステータス レジスタ、スタックポインタを保持するデータ メモリです。CPUクロックの停止はコアが一般的な操作や計算を実行することを禁止します。

### I/O クロック - clk<sub>I/O</sub>

I/Oクロックはタイマ/カウンタ、SPI、USARTのようなI/O部の大部分で使われます。I/Oクロックは外部割り込み部でも使われますが、いくつかの外部割り込みは例えI/Oクロックが停止されても検出されることをこのような割り込みに許す非同期論理回路によって検出されることに注意してください。多用途直列(USI)部の開始条件検出はclk<sub>I/O</sub>が停止されるとき非同期に実行され、全休止形態でUSI開始条件検出を可能とすることにも注意してください。

### フラッシュ クロック - clk<sub>FLASH</sub>

フラッシュ クロックはフラッシュ メモリ インターフェースの動作を制御します。このフラッシュ クロックは常にCPUクロックと同時に活動します。

# 非同期タイマ クロック - clkasy

非同期タイマクロックは外部32kHzクロック用クリスタルから直接的にクロック駆動されるのを非同期タイマ/カウンタとLCD制御器に許します。この専用クロック範囲はデバイスが休止形態の時でも、このタイマ/カウンタの実時間計数器としての使用を許します。これはデバイスの他の部分が休止形態でも、継続出力をLCD制御器にも許します。

# A/D変換クロック - clk<sub>ADC</sub>

A/D変換器には専用のクロック範囲が提供されます。これはデジタル回路によって生成された雑音を低減するためにCPUとI/Oクロックの停止を許します。これはより正確なA/D変換結果を与えます。

# クロック元

このデバイスには右で示されるようにフラッシュ ヒュース ビットによって選択可能な後続のクロック元選択があります。選んだクロック元からのクロックはAVRクロック発生器への入力で、適切な単位部へ配給されます。

各クロック選択に対する各種選択は次項で得られます。CPUがパワーダウンまたはパワーセーブから起動するとき、選んだクロック元は命令実行開始前に安定な発振器動作を保証する起動時間に使われます。CPUがリセットから始まるとき、これらは通常動作開始前に安定電圧へ達するのを電源に許す付加遅延です。ウォッチドッグ発振器はこの起動時間の実時間部のタイミングに使われます。各計時完了に使われるウォッチドッグ(WDT)発振器の周期数は表3.で示されます。198頁の「代表特性」で示されるように、ウォッチドッグ・発振器の周波数は電圧に依存します。

主っ	クロック種別選択	
죠. ४. ४. ४. ४ ४ ४ ४ ४ ४ ४ ४ ४ ४ ४ ४ ४ ४	<b>ソロツソ 4単 万川 1共 4八</b>	

クロック種別	CKSEL3~0
外部クリスタル/セラミック発振子	1111~1000
外部低周波数クリスタル発振子	0111~0110
校正付き内蔵RC発振器	0010
外部クロック信号	0000
(予約)	00x1, 010x

注: 1=非プログラム、0=プログラム、x=0または1

### 表3. WDT発振器の代表的計時完了値、周期数

VCC=3.0V	VCC=5.0V	周期数
4.3ms	4.1ms	4K (4096)
69ms	65ms	64K (65536)

# 既定のクロック元

このデバイスはCKSEL=0010, SUT=10, CKDIV8=プログラム(0)で出荷されます。既定クロック元設定は最長起動時間で初期システム クロック 8 分周の内蔵RC発振器です。この既定設定は全ての使用者が実装または並列書き込み器を使ってそれらを希望したクロック元設定にできることを保証します。

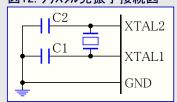
### クリスタル用発振器

XTAL1とXTAL2は**図12**.で示されるように、チップ上の発振器としての使用に設定できる反転増幅器の各々、入力と出力です。クリスタル発振子またはセラミック振動子のどちらでも使えます。

C1とC2はクリスタル発振子とセラミック振動子の両方について常に等しくすべきです。このコンデンサの最適値は使うクリスタル発振子やセラミック振動子、浮遊容量の量、その環境の電磁雑音に依存します。クリスタル発振子使用に対するコンデンサ選択について初期の指針のいくつかは表4.で与えられます。セラミック振動子については製造業者によって与えられたコンデンサ値が使われるべきです。

発振器は示された周波数範囲で最適化された4つの異なる動作ができます。この動作は、表4.で示されたCKSEL3~1ヒューズによって選ばれます。

# 図12. クリスタル発振子接続図



# 表4. クリスタル用発振器動作

CKSEL3~1	周波数範囲	推奨C1,2容量
100 (注1)	0.4~0.9MHz	-
101	0.9~3.0MHz	12∼22pF
110	3.0~8.0MHz	12∼22pF
111	8.0∼ MHz	12∼22pF

注1: 本選択はクリスタル発振子ではなく、セラミック振動子でのみ使われるべきです。

CKSEL0ヒュース はSUT1,0ヒュースと共に、表5.で示される起動遅延時間を選びます。

### 表5. クリスタル発振子/ヤラミック振動子用起動遅延時間選択表

20. 77777	以3.7/7/2017 17 17 17 17 18 31 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1							
CKSEL0	SUT1,0	パワーダウン, パワーセーブ からの起動遅延時間	リセットからの付加遅延 時間 (VCC=5.0V)	推奨使用法				
	0 0	258×CK ( <b>注1</b> )	14×CK+4.1ms	外部セラミック振動子、高速上昇電源				
0	0 1	258×CK ( <b>注1</b> )	$14 \times \text{CK+65ms}$	外部セラミック振動子、低速上昇電源				
U	1 0	1K×CK ( <b>注2</b> )	14×CK	外部セラミック振動子、低電圧検出(BOD)リセット許可				
	1 1	1K×CK ( <b>注2</b> )	14×CK+4.1ms	外部セラミック振動子、高速上昇電源				
	0 0	1K×CK ( <b>注2</b> )	$14 \times \text{CK+65ms}$	外部セラミック振動子、低速上昇電源				
1	0 1	16K×CK	14×CK	外部クリスタル発振子、低電圧検出(BOD)リセット許可				
1	1 0	16K×CK	14×CK+4.1ms	外部クリスタル発振子、高速上昇電源				
	1 1	16K×CK	$14 \times \text{CK+65ms}$	外部クリスタル発振子、低速上昇電源				

**注1**: これらの選択はデバイスの最高周波数付近での動作でないとき、応用にとって起動での周波数安定性が重要でない場合だけ使われるべきです。これらの選択はクリスタル発振子用ではありません。

**注2**: これらの選択はセラミック振動子での使用を意図され、起動での周波数安定性を保証します。デバイスの最高周波数付近での動作でないとき、応用にとって起動での周波数安定性が重要でない場合はクリスタル発振子も使えます。





# 低周波数クリスタル用発振器

デバイスに対するクロック元として時計用32.768kHzクリスタルを使うには、CKSEL3~0ヒューズを'0110','0111'に設定することによって低周波数クリスタル発振器が選ばれなければなりません。クリスタルは図12.で示されるように接続されるべきです。この発振器が選ばれると、起動遅延時間は表6.で示されるSUTヒューズと表7.で示されるCKSEL0ヒューズによって決定されます。

# 表6. 低周波数クリスタル発振器用起動遅延時間選択表

SUT1,0	リセットからの 付加遅延時間 (VCC=5.0V)	推奨使用法		
0 0	14×CK	高速上昇電源またはBOD許可		
0 1	$14 \times \text{CK+4.1ms}$	低速上昇電源		
1 0	$14 \times \text{CK+65ms}$	起動時の周波数安定		
1 1	(予約)			

### 表7. 低周波数クリスタル発振器用起動時間選択表

CKSEL3~0	n°ワーダウン, n°ワーセーブからの 起動遅延時間	推奨使用法
0 1 1 0	1K×CK	(注1)
0 1 1 1	32K×CK	起動時の周波数安定

注1: これらの選択は起動時の周波数安定度が応用に とって重要でない場合だけ使われるべきです。

# 校正付き内蔵RC発振器(訳注:本項は原書に対して修正されています。)

校正された内蔵RC発振器は決められた8.0MHzクロックを供給します。この周波数は3V,25℃での公称値です。8MHz周波数がデバイスの仕様(VCCに依存)を越える場合、起動中に内部周波数を8分周するためにCKDIV8ヒュースがプログラム(0)されなければなりません。このデバイスはCKDIV8ヒュースがプログラム(0)で出荷されます。より多くの詳細については20頁の「システム クロック前置分周器」をご覧ください。このクロックは表8.で示されるように

表8. 校正付き内蔵RC発振器動作CKSEL3~0公称周波数0010(注)8.0MHz注: デバイスはこの選択で出荷されます。

CKSELヒュース'のプログラミングによってシステム クロックとして選べます。選んだなら、外部部品なしで動作します。リセット中、ハートウェアが発振校正レシ、スタ(OSCCAL)に校正値ハイトを設定し、これによってRC発振器を自動的に校正します。この校正は3V,25℃で公称周波数±10%以内の周波数を与えます。atmel.com/avrで利用可能な応用記述に記載された実行時校正法の使用で、与えられたどのVCCと温度でも±2%の精度を達成することができます。この発振器がチップ(システム)クロックとして使われるとき、ウォッチトック・発振器は未だウォッチトック・タイマとリセット付加遅延タイマに使われます。予め設定された校正値のより多くの情報については177頁の「校正ハイト」項をご覧ください。

この発振器が選ばれると、起動時間は**表9**.で示されるようにSUTヒューズによって決定されます。内蔵RC発振器選択はタイマ用発振器ピンとして使われることをXTAL1/TOSC1とXTAL2/TOSCピンに許します。

### 表9 校正付き内蔵RC発振器用起動遅延時間選択表

SUT1,0	パワーダウン, パワーセーブ からの起動遅延時間	リセットからの付加遅延          時間 (VCC=5.0V)					
0 0	6×CK	14×CK	低電圧検出(BOD)リセット許可				
0 1	6×CK	14×CK+4.1ms	高速上昇電源				
10 (注1)	6×CK	14×CK+65ms 低速上昇電源					
1 1		(予約)					

注1: デバイスはこの選択で出荷されます。

# 発振校正レジスタ (Oscillator Calibration Register) OSCCAL (訳注:本項は原書に対して修正されています。)

ピット	7	6	5	4	3	2	1	0	
(\$66)	_	CAL6	CAL5	CAL4	CAL3	CAL2	CAL1	CAL0	OSCCAL
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値	0			゙゚゚゙゙゙゙゙゙゙゙゙゙゙゙゙゙゙゙゙゙゙゙゙゙゙゙゙゙゙゙゙゙゙゙゙゙゙゙	イス固有の校	正値			

• **ビット7 - Res**: **予約** (Reserved) (**訳注**:共通性のため本項追加)

このビットは予約されており、常に0として読みます。

# ● ビット6~0 - CAL6~0:発振校正値 (Oscillator Calibration Value)

このアドレスへの校正バイト書き込みは発振器周波数の偏差処理を省くために内蔵発振器を調整します。これはチップリセット中、自動的に行われます。OSCCALが0の時に最低利用可能周波数が選ばれます。このレジスタへの以外の値を書くことは内蔵発振器の周波数を増加します。このレジスタへの\$7F書き込みは最高使用可能周波数にします。校正付き発振器はフラッシュメモリとEEPROMのアクセス時間に使われます。フラッシュメモリまたはEEPROMが書かれる場合、公称周波数より上へ10%を越えて校正してはいけません。そうでなければ、フラッシュメモリまたはEEPROM書き込みは失敗するかもしれません。この発振器は8.0MHzへの校正を意図したものであることに注意してください。表10.で示されるような他の値への調整は保証されません。

表10. 内蔵RC発振器周波数範囲

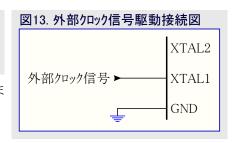
OSCCAL値	公称周波数に対する割合				
USUCAL恒	最小	最大			
\$00	50 %	100 %			
\$3F	75 %	150 %			
\$7F	100 %	200 %			

# 外部クロック信号

外部クロック元からデバイスを駆動するためにXTAL1は 表11 外部クロック信号動作 図13.で示されるように駆動されるべきです。外部ク ロックでデバイスを走行するためにはCKSELヒューズが '0000'にプログラム(設定)されなければなりません。

衣 11. クト 両 ク L ツ ク 1								
CKSEL3~0	周波数範囲							
0 0 0 0	0∼16MHz							

このクロック元が選ばれると、起動時間は表12.で示されるようにSUTヒュース。によって決定されま



### 表12. 外部クロック信号駆動用起動遅延時間選択表

SUT1,0	パワーダウン, パワーセーブ からの起動遅延時間	リセットからの付加遅延 時間 (VCC=5.0V)	推奨使用法						
0 0	6×CK	14×CK	低電圧検出(BOD)リセット許可						
0 1	6×CK	14×CK+4.1ms	高速上昇電源						
1 0	6×CK	14×CK+65ms	低速上昇電源						
11		(予約)							

外部クロックを供給するとき、MCUの安定な動作を保証するために供給したクロック周波数の急な変化を避けることが必要とされます。或 るクロック周期から次への2%より大きな周波数変化は予測されない事態を引き起こします。このようなクロック周波数での変化中、MCUは リセットに保たれるのを保証することが必要とされます。

システム クロック前置分周器が安定な動作を保証しながら、内部クロック周波数の実行時変更の実現に使えることに注意してください。 詳 細については20頁の「システム クロック前置分周器」を参照してください。

# クロック出力緩衝部(外部クロック出力)

CKOUT ヒュース がプログラム(0)されると、システム クロックが $CLKO(ヒ^{\circ})$ に出力されます。この動作はチップのクロックがシステム 上の他の回路を駆 動する時用です。このヒューズがプログラム(0)されると、I/Oピンの通常動作は無視され、このクロックはリセット中も出力されます。CLKOがク ロック出力を扱うとき、校正付き内蔵RC発振器を含む何れのクロック元も選べます。システム クロック前置分周器が使われると、CKOUTヒュー ズがプログラム(0)された時の出力は分周したシステム クロックです。

### タイマ/カウンタ用発振器

ATmega169はタイマ/カウンタ用発振器(TOSC1とTOSC2)ピンとXTAL1,XTAL2ピンを共用します。 これはシステム クロック元として校正付き内 蔵RC発振器が選ばれる時だけタイマ/カウンタ用発振器が使えることを意味します。この発振器は時計用32.768kHzクリスタルでの使用に 最適化されています。クリスタルの接続については17頁の図12.をご覧ください。

非同期状態レシブスタ(ASSR)の外部クロック許可(EXCLK)ビットが論理1を書かれると、TOSC1に外部クロック元の印加が行えます。32kHzクリス タルに代わる入力として外部クロックを選ぶ更なる記述については97頁の「**タイマ/カウンタ2の非同期動作**」をご覧ください。





# システム クロック前置分周器

ATmega169のシステム クロックはクロック前置分周レシ、スタ(CLKPR)の設定によって分周できます。この特徴(機能)は必要とされる処理能力が低い時の消費電力削減に使うことができます。これは全クロック種別で使え、CPUと全同期周辺機能のクロック周波数に影響を及ぼします。clk<sub>CPU</sub>、clk<sub>FLASH</sub>、clk<sub>I/O</sub>、clk<sub>ADC</sub>は表13.で示された値によって分周されます。

clk<sub>ASY</sub>(非同期タイマ/カウンタ用クロック)のクロック周波数は、タイマ/カウンタが同期クロックで駆動される場合だけ分周されることに注意してください(<mark>訳注</mark>:共通性のため本行追加)。

前置分周器設定間を切り替えるとき、システム クロック前置分周器は中間(経過途中)の周波数が直前の設定に対応するクロック周波数または新規設定に対応するクロック周波数のどちらよりも高くなく、クロック系で不具合が起きないことを保証します。

前置分周器として実行するリプルカウンタは分周されないクロック周波数で走行し、CPUのクロック周波数より速いかもしれません。従って例え(カウンタ値が)読めるとしても、前置分周器の状態を決めることはできず、1から他へのクロック分周値切り替えを行う正確な時間は必ずしも予測できません。CLKPS値が書かれる時から新規クロック周波数が活性(有効)になる前にT1+T2~T1+2×T2間かかります。この間で2つの有効なクロック端が生成されます。ここでのT1は直前のクロック周期、T2は新規前置分周器設定に対応する周期です。

予期せぬクロック周波数の変更を防ぐため、CLKPSビットの変更は次の特別な書き込み手順に従わなければなりません。

- 1. クロック分周値変更許可(CLKPCE)ビットに1、CLKPR内の他の全ビットに0を書きます。
- 2. (次からの)4周期以内にCLKPCEビットをOとする欲したCLKPS値を書きます。

前置分周器設定変更時、書き込み手続きが割り込まれないことを保証するため、割り込みは禁止されなければなりません。

# クロック前置分周レシ、スタ (Clock Prescale Register) CLKPR

ピット (\$61)	7 CLKPCE	6	5 -	4	3 CLKPS3	2 CLKPS2	1 CLKPS1	0 CLKPS0	CLKPR
Read/Write	R/W	R	R	R	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	内容参照	内容参照	内容参照	内容参照	

## • ビット7 - CLKPCE: クロック分周値変更許可 (Clock Prescaler Change Enable)

CLKPSビットの変更を許可するためにCLKPCEビットは論理1を書かれなければなりません。CLKPCEビットは同時にCLKPRの他の全 ビットが0を書かれる時だけ更新されます。CLKPCEは書き込み後4クロック周期またはCLKPSビット書き込み時、ハードウェアによって解除(0)されます。この制限時間(4クロック周期)内のCLKPCEビット再書き込みは制限時間の延長もCLKPCEビットの解除(0)も行いません。

### • ビット3~0 - CLKPS3~0: クロック分周値選択 (Clock Prescaler Select Bits 3~0)

これらのビットは選んだクロック元と内部システム クロック間の分周値を定義します。これらのビットは応用の必要条件に合わせた各種クロック周波数を実行時に書けます。分周値が使われると、分周器はMCUへの主クロックを分周し、全ての同期周辺機能の速度が減じられます。分周値は表13.で与えられます。

CKDIV8ヒュース、がCLKPSビットの初期値を決めます。CKDIV8が非プログラム(1)にされると、CLKPSビットは'0000'にリセットされます。CKDI V8がプログラム(0)されると、CLKPSビットは起動時に8分周を与える'0011'にリセットされます。現在の動作条件でデバイスの最高周波数より高い周波数のクロック元を選んだ場合、この機能が使われるべきです。CKDIV8ヒュース、設定に拘らず、どの値もCLKPSビットへ書けることに注意してください。応用ソフトウェアは現在の動作条件でデバイスの最高周波数より高い周波数のクロック元を選んだ場合、充分な分周値が選ばれることを保証しなければなりません。本デバイスはCKDIV8ヒュースがプログラム(0)で出荷されます。

表13. クロック前置分周器選択 CLKPS3 0CLKPS2 0 0 0 CLKPS1 0 0 0 0 CLKPS0 0 0 0 0 0 0 0 分周値(数) 16 64 128 256 (予約)

# 電力管理と休止形態

休止形態は応用でMCU内の未使用部を一時停止することを可能にし、それによって節電します。AVRは応用で必要な消費電力に 仕立てることを使用者に許す様々な休止形態を提供します。

5つの休止形態の何れかへ移行するには休止形態制御レジスタ(SMCR)の休止許可(SE)ビットが論理1を書かれ、SLEEP命令が実行されなければなりません。SMCRの休止種別選択(SM2~0)ビットはSLEEP命令によって活性(有効)にされる休止形態(アイドル、A/D変換雑音低減、パワーダウン、パワーセーブ、スタンバイ)のどれかを選びます。一覧については表14.をご覧ください。MCUが休止形態中に許可した割り込みが起こると、MCUは起動します。その時にMCUは起動時間に加えて4周期停止され、割り込みルーチンを実行し、そしてSLEEP命令の次の命令から実行を再開します。デバイスが休止から起動するとき、レジスタファイルとSRAMの内容は変えられません。休止形態中にリセットが起こると、MCUは起動し、リセット ペクタから実行します。

16頁の図11.はATmega169の各種クロック系統とその配給を示します。この図は適切な休止形態を選ぶ助けになります。

### 休止形態制御レジスタ (Sleep Mode Control Register) SMCR

この休止形態制御レジスタは電力管理用の制御ビットを含みます。

ピット	7	6	5	4	3	2	1	0	_
\$33 (\$53)	-	-	-	-	SM2	SM1	SM0	SE	SMCR
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

### • **ビット3~1 - SM2~0**: 休止種別選択 (Sleep Mode Select Bit 2, 1 and 0)

これらのビットは表14.で示される利用可能な5つの休止形態の1つを選びます。

_	表14. 休	止形態	種別選択	?
	SM2	SM1	SM0	休止形態種別
	0	0	0	アイドル動作
	0	0	1	A/D変換雑音低減動作
	0	1	0	パプーダウン動作
	0	1	1	パプーセーフ・動作
	1	0	0	(予約)
	1	0	1	(予約)
	1	1	0	スタンバイ動作 (注)
	1	1	1	(子約)

**注**: スタンバイ動作は外部クリスタル発振子またはセラミック 振動子でだけ利用できます。

### • ビット0 - SE: 休止許可 (Sleep Enable)

SLEEP命令が実行される時にMCUを休止形態へ移行させるには、休止許可(SE)ビットが論理1を書かれなければなりません。MCUの目的外休止形態移行を避けるため、SLEEP命令実行直前に休止許可(SE)ビットを設定(1)し、起動後直ちに解除(0)することが推奨されます。

# アイドル動作

休止種別選択(SM2~0)ビットが'000'を書かれるとき、SLEEP命令はMCUをアイドル動作へ移行させて、CPUを停止しますが、LCD制御器、SPI、USART、アナログ比較器、A/D変換器、多用途直列インターフェース(USI)、タイマ/カウンタ、ウォッチドッグ、割り込み機構の継続動作を許します。この休止形態は基本的にclkCPUとclkFLASHを停止する一方、他のクロックに走行を許します。

アイドル動作はMCUにタイマ溢れやUSARTの送信完了などの内部割り込みだけでなく、外部で起動された割り込みからの起動も可能にします。アナログ比較器割り込みからの起動が必要とされないなら、アナログ比較器制御/状態レジ゙スタ(ACSR)のアナログ比較器禁止(ACD)ビットを設定(1)することによってアナログ比較器を電源断にできます。これはアイドル動作での消費電力を削減します。A/D変換が許可されるなら、この動作に移行すると変換が自動的に始まります。

### A/D変換雑音低減動作

SM2~0ビットが'001'を書かれるとき、SLEEP命令はMCUをA/D変換雑音低減動作へ移行させ、CPUを停止しますが、A/D変換器、外部割り込み、USIの開始条件検出、タイマ/カウンタ2、LCD制御器、ウォッチドック'の(許可されていれば)継続動作を許します。この休止形態作は基本的にclkg/O, clkCPU, clkFLASHを停止する一方、他のクロックに走行を許します。

これはA/D変換に対する雑音環境を改善し、より高い分解能の測定を可能にします。A/D変換器が許可されている場合、この動作に移行すると、変換が自動的に始まります。A/D変換完了割り込みからの他、外部リセット、ウォッチトック・リセット、低電圧検出(BOD)リセット、LCD制御器割り込み、USIの開始条件検出割り込み、タイマ/カウンタ2の割り込み、SPM/EEPROM操作可割り込み、INTOの外部レヘル割り込み、ピン変化割り込みだけが、A/D変換雑音低減動作からMCUを起動できます。





# パワーダウン動作

休止種別選択(SM2~0)ビットが'010'を書かれると、SLEEP命令はMCUをパワーダウン動作へ移行させます。この動作では外部発振器が停止される一方、外部割り込み、多用途直列インターフェース(USI)の開始条件検出、ウォッチドッグ機能は(許可されていれば)継続して動作します。外部リセット、ウォッチドッグリセット、低電圧検出(BOD)リセット、USI開始条件検出割り込み、INT0の外部レベル割り込み、ピン変化割り込みだけがMCUを起動できます。この休止形態は基本的に生成した全クロックを停止し、非同期部の動作だけを許します。

レヘールで起動した割り込みがハプローダウン動作からの復帰に使われる場合、MCUを起動するためには変更したレヘールが一定時間保持されなければならないことに注意してください。詳細については34頁の「外部割り込み」を参照してください。

ハプワーダウン動作から起動するとき、起動条件が起きてから起動の効果が現れるまで遅延があります。これは停止されてしまっている後の再始動と安定になることをクロックに許します。この起動(遅延)時間は17頁の「クロック元」で記述されるように、リセット遅延時間を定義するのと同じCKSELヒュース、によって定義されます。

### パワーセーブ動作

SM2~0ビットが'011'を書かれると、SLEEP命令はMCUをパワーセーブ動作へ移行させます。この動作は(次の)1つの例外を除いてパワーダウン動作と同じです。

タイマ/カウンタ2またはLCD制御器が許可される場合、それらは休止中も走行(動作)を維持します。ステータスレシ、スタ(SREG)の全割り込み許可(I)ヒ、ットが設定(1)され、タイマ/カウンタ2割り込み許可レシ、スタ(TIMSK2)のタイマ/カウンタ2溢れ割り込み許可(TOIE2)ヒ、ットまたは比較2割り込み許可(OCIE2A)ヒ、ットが設定(1)されるなら、デバイスは対応するどちらの割り込みからでも起動できます。LCD制御器割り込みからも起動できます。

タイマ/カウンタ2とLCD制御器のどちらも走行(動作)しないなら、パワーダウン動作をパワーセーブ動作の代わりにすることが推奨されます。

タイマ/カウンタ2とLCD制御器はパワーセーブ動作で同期と非同期の両方でクロック駆動できます。この2つの単位部に対するクロック元は他の部分と関係なく選べます。タイマ/カウンタ2とLCD制御器のどちらも非同期クロックを使わない場合、休止中、タイマ/カウンタ用発振器は停止されます。タイマ/カウンタ2とLCD制御器のどちらも同期クロックを使わない場合、休止中、そのクロック元は停止されます。例えパワーセーブ動作で同期クロックが走行しても、このクロックはタイマ/カウンタ2とLCD制御器に対してだけ利用可能なことに注意してください。

# スタンバイ動作

外部クリスタル発振子/セラミック振動子クロック種別が選ばれ、SM2~0ビットが'110'のとき、SLEEP命令はMCUをスタンバイ動作へ移行させます。この動作は(外部クリスタル用)発振器が走行(動作)を保たれる例外を除いてパワーダウン動作と同じです。デバイスはスタンバイ動作から6クロック周期で起動します。

表15. 各休止形態に於ける動	作クロック範囲と復帰起動要因
-----------------	----------------

ſ			動作ク	ロック軍	囲		動作夠	Ě振器		復則	帚起動要	医因 (割	り込み)	SPM A/D その		
	休止種別	clk CPU	clk FLASH	clk 10	clk ADC	clk ASY	主クロック供給元	タイマ用 発振器	INTO ピン変化 割り込み	USI 開始条件 検出	LCD 制御器	タイマ/ カウンタ 2	EEPROM	A/D 変換 完了	その他 I/O	
	アイドル			0	0	0	0	2	0	0	0	0	0	0	0	
	A/D変換 雑音低減				0	0	0	2	3	0	2	2	0	0		
I	ハ <sup>°</sup> ワータ <sup>˙</sup> ウン								3	0						
	ハ゜ワーセーフ゛					0		0	3	0	0	0				
	スタンバイ( <mark>注1</mark> )						0		3	0						

**注1**: クロック元として推奨した外部クリスタル発振子またはセラミック振動子が選ばれた場合だけです。

- ② タイマ/カウンタ2またはLCD制御器のどちらかが非同期動作で走行する場合です。
- ③ INTOについてはレベル割り込みのみです。

# 電力削減

電力削減レシ、スタ(PRR)は消費電力を削減するために個別周辺機能へのクロックを停止する方法を提供します。周辺機能は現状で固定化され、I/Oレシ、スタは読み込みも書き込みもできません。クロックを停止している時に周辺機能によって使われていた資源は占有されたままですので、その周辺機能は殆どの場合、クロックを停止する前に禁止されるべきです。周辺機能部の起動は電力削減レシ、スタ(PRR)のビットを解除(0)することによって行い、その周辺機能部を停止前と同じ状態にします。

周辺機能部の停止は全体に亘る重要な消費電力の削減のために活動動作とアイドル動作で使えます。例については202頁の「**周辺機能部供給電流**」をご覧ください。その他の休止形態ではクロックが予め停止されます。

### 電力削減レジスタ (Power Reduction Register) PRR

ピット	7	6	5	4	3	2	1	0	
(\$64)	_	-	-	PRLCD	PRTIM1	PRSPI	PRUSART0	PRADC	PRR
Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

• ビット7~5 - Res: 予約 (Reserved)

これらのビットは予約されており、常に0として読みます。

• ビット4 - PRLCD: LCD電力削減 (Power Reduction LCD)

このビットへの1書き込みはLCD制御器を停止します。停止前にLCD制御器は禁止され、表示器は放電されなければなりません。 LCD制御器禁止方法の詳細については148頁の「LCDの禁止」をご覧ください。

• ビット3 - PRTIM1 : タイマ/カウンタ1電力削減 (Power Reduction Timer/Counter1)

このビットへの1書き込みはタイマ/カウンタ1部を停止します。タイマ/カウンタ1が許可されると、停止前と同様に動作は継続します。

• ビット2 - PRSPI: 直列周辺インターフェース電力削減 (Power Reduction Serial Peripheral Interface)

このビットへの1書き込みはその部分へのクロック停止によって直列周辺インターフェース(SPI)を停止します。SPIの再起動時、SPIは正しい動作を保証するために再初期化されるべきです。

• ビット1 - PRUSARTO: USARTO電力削減 (Power Reduction USARTO)

このビットへの1書き込みはその部分へのクロック停止によってUSARTを停止します。USARTの再起動時、USARTは正しい動作を保証するために再初期化されるべきです。

• ビット0 - PRADC : A/D変換器電力削減 (Power Reduction ADC)

このビットへの1書き込みはA/D変換器(ADC)を停止します。A/D変換器は停止前に禁止されなければなりません。A/D変換器停止時、アナログ比較器はADC入力切替器を使えません。

注: アナログ:比較器は130頁の「アナログ:比較器制御/状態レジスタ(ACSR)」のアナログ:比較器禁止(ACD)と、ットを使って禁止されます。





# 消費電力の最小化

これらはAVRが制御するシステムで消費電力の最小化を試みる時に考慮するためのそれぞれの検討点です。一般的に休止形態は可能な限り多く使われるべきで、休止種別は動作するデバイスの機能が可能な限り少なくなるように選ばれるべきです。必要とされない全ての機能は禁止されるべきです。特に次の機能部は最低可能消費電力の達成を試みるとき、特別な考慮を必要とするでしょう。

# A/D変換器 (ADC)

許可したなら、A/D変換器は全休止形態で許可されます。節電するため、休止形態の何れかへ移行する前にA/D変換器は禁止されるべきです。A/D変換器がOFFそして再びONに切り替えられると、次の(最初の)変換は延長された(初回)変換になります。A/D変換器操作の詳細については132頁の「A/D変換器」を参照してください。

### アナログ比較器

アイドル動作へ移行するとき、アナロゲ比較器は使われないなら、禁止されるべきです。A/D変換雑音削減動作へ移行するとき、アナロゲ比較器は禁止されるべきです。その他の休止形態でのアナロゲ比較器は自動的に禁止されます。しかしアナロゲ比較器が入力として内部基準電圧を使う設定の場合、全休止形態でアナロゲ比較器は禁止されるべきです。さもなければ内部基準電圧は休止形態と無関係に許可されます。アナロゲ比較器設定法の詳細については130頁の「アナロゲ比較器」を参照してください。

# 低電圧検出器 (BOD)

低電圧検出器(BOD)が応用で必要とされないなら、この単位部はOFFにされるべきです。低電圧検出器がBODLEVELヒューズによって許可されていると全休止形態で許可され、故に常時電力を消費します。これはより深い休止形態での総消費電流にとって重要な一因になります。低電圧検出器(BOD)設定法の詳細については26頁の「低電圧検出 (BOD)」を参照してください。

### 内部基準電圧

内部基準電圧は低電圧検出器(BOD)、アナログ比較器、A/D変換器によって必要とされる時に許可されます。これら単位部が上の項で記述されたように禁止されると、内部基準電圧は禁止され、電力を消費しません。再び許可する場合、この出力が使われる前に使用者は基準電圧へ起動(安定時間)を与えなければなりません。基準電圧が休止形態でON保持される場合、この出力は直ちに使えます。起動時間の詳細については27頁の「内部基準電圧」を参照してください。

# ウォッチト゛ック゛タイマ

ウォッチト、ック、タイマが応用で必要とされないなら、この単位部はOFFにされるべきです。ウォッチト、ック、タイマが許可されていると全休止形態で許可され、故に常時電力を消費します。これはより深い休止形態での総消費電流にとって重要な一因になります。ウォッチト、ック、タイマ設定法の詳細については28頁の「ウォッチト、ック、タイマ」を参照してください。

### ホートピン

休止形態へ移行するとき、全てのポート ピンは最小電力使用に設定されるべきです。最も重要なことはその時にピンが抵抗性負荷を駆動しないのを保証することです。I/Oクロック(clk<sub>I/O</sub>)とA/D変換クロック(clk<sub>ADC</sub>)の両方が停止される休止形態ではデバイスの入力緩衝部が禁止されます。これは必要とされない時に入力論理回路によって電力が消費されないことを保証します。いくつかの場合で入力論理回路は起動条件を検出するために必要とされ、その時は許可されます。どのピンが許可されるかの詳細については39頁の「デジタル入力許可と休止形態」を参照してください。入力緩衝部が許可され、入力信号が浮いている状態のままか、またはアナログ信号電圧がVCC/2付近の場合、入力緩衝部は過大な電力を消費するでしょう。

アナログ、入力ピンに対するデジタル入力緩衝部は常に禁止されるべきです。入力ピンでのVCC/2付近のアナログ信号入力は活動動作でも 重要な電流を引き起こし得ます。デジタル入力緩衝部はデジタル入力禁止レジスタ(DIDR0とDIDR1)の書き込みによって禁止できます。 詳細については131頁の「デジタル入力禁止レジスタ1(DIDR1)」と141頁の「デジタル入力禁止レジスタ0(DIDR0)」を参照してください。

## JTAGインターフェースと内蔵デバッグ機能(OCD)

内蔵デバッグ機能がOCDENヒューズによって許可され、チップがパワーダウンまたはパワーセーブの休止形態へ移行すると、主クロック元は許可に留まります。これらの休止形態では、これが総消費電流にとって重要な一因になります。これを避けるには3通りの方法があります。

- OCDENtュースを禁止 (非プログラム(1)設定)
- JTAGENヒュースを禁止 (非プログラム(1)設定)
- MCU制御レシ、スタ(MCUCR)のJTAG禁止(JTD)ビットへの1書き込み

JTAGインターフェースが許可され、JTAG TAP制御器がデータを移動していないと、TDOピンは浮き状態のままにされます。TDOピンに接続したハードウェアが論理レヘ゛ルをプルアップしないなら、消費電力が増加するでしょう。 走査チェーン内の次のデハ・イスのTDIピンがこの問題を避けるプルアップを含むことに注意してください。 MCUCRのJTDビットに1を書くか、またはJTAGENヒュースで非プログラムのままにすることがJTAGインターフェースを禁止します。

# システム制御とリセット

### AVRのリセット

リセット中、全てのI/Oレシブスタはそれらの初期値に設定され、プログラムはリセット、ブクタから実行を開始します。 リセット、ブクタに配置される命令は、きっとリセット処理ルーチンへのJMP(絶対分岐)命令でしょう。 プログラムが決して割り込み元を許可しないなら、割り込みへブクタは使われず、これらの位置に通常のプログラムコートが配置できます。 これはリセット、ブクタが応用領域の一方、割り込みへブクタがブート領域の場合やその逆も同様です。 図14.の回路構成図はリセット論理回路を示します。 表16.はリセット回路の電気的特性を定義します。

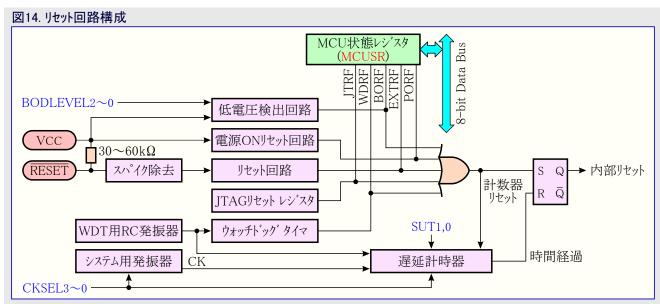
AVRのI/Oポートはリセット元が有効になると直ちにそれらの初期状態にリセットされます。これはどのクロック元の走行も必要ありません。

全てのリセット元が無効にされてしまった後、遅延計数器(タイマ)が始動され、内部リセットを引き伸ばします。これは通常動作開始前に安定電圧へ達することを電源に許します。遅延タイマの遅延時間はCKSELヒューズを通して使用者によって定義されます。この遅延時間についての各種選択は17頁の「クロック元」で示されます。

# リセット元

ATmega169には次の5つのリセット元があります。

- 電源ONリセット・・・・・ 供給電圧が電源ONリセット閾値電圧(VPOT)以下でMCUがリセットされます。
- 外部リセット・・・・・・・ RESET ピンが最小パルス幅以上Lowレベルに保たれると、MCUがリセットされます。
- ウォッチト、ック、リセット・・・ ウォッチト、ック、が許可され、ウォッチト、ック、タイマが終了すると、MCUがリセットされます。
- 低電圧リセット・・・・・ 低電圧検出が許可され、供給電圧(VCC)が低電圧検出電圧(VBOT)以下でMCUがリセットされます。
- JTAG AVR リセット・・ JTAGシステムの走査チェーンの1つとしてリセットレジスタ内に論理1がある間中、MCUがリセットされます。詳細については156頁の「IEEE 1149.1 (JTAG) 境界走査」を参照してください。



### 表16. リセット電気的特性 単位 シンボル 項目 条件 最小 代表 最大 上昇時電源ONリセット閾値電圧 0.7 1.0 1.4 **V**POT TA=-40~85°C 下降時電源ONリセット閾値電圧(注1) 1.3 V 0.6 0.9 $V_{RST}$ RESETt°ン閾値電圧 0.2VCC 0.9VCC VCC=3V RESETt°ン最小パルス幅 2.5 $t_{RST}$ μs

注1: 供給電圧がこの電圧以下にならないと、上昇時の電源ONJセットは動作しません。



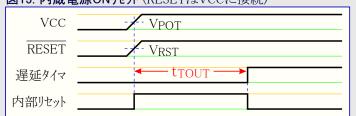


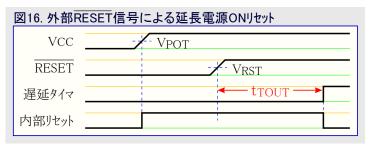
### 電源ONリセット

電源ONJセット(POR)パルスはチップ上の検出回路によって生成されます。検出電圧は表16.で定義されます。POR信号はVCCが検出電圧以下の時は必ず活性(有効)にされます。POR回路は供給電圧異常検出は勿論、始動リセットの起動にも使えます。

電源ONJセット回路はデバイスが電源投入でリセットされることを保証します。電源ONJセット閾値電圧(VPOT)への到達はVCCの上昇後にデバイスがどのくらいリセットを保つかを決める遅延計数器(タイマ)を起動します。VCCがこの検出電圧以下に低下すると、リセット信号はどんな遅延もなく再び有効にされます。

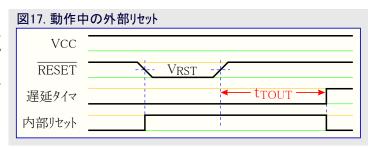






### 外部リセット

外部リセットはRESETt°ンのLowレヘルによって生成されます。例えクロックが動いていなくても、最小ハペルス幅(表16.参照)以上のリセットハペルスはリセットを生成します。短すぎるハペルスはリセットの生成が保証されません。印加された信号の上昇がリセット閾値電圧(VRST)に達すると(遅延タイマが起動され)、遅延タイマは遅延時間(tTOUT)経過後にMCUを始動します。



# 低電圧(ブラウンアウト)検出リセット

ATmega169には固定化された起動(検出)電圧と比較することによって動作中のVCCを監視するチップ上の低電圧検出(BOD)回路があります。BODの起動電圧はBODLEVELtュース。によって選べます。起動電圧はスパイク対策BODを保証するためにヒステリシスを持ちます。検出電圧のヒステリシスはVBOT+=VBOT+VHYST/2、VBOT-=VBOT-VHYST/2と解釈すべきです。

BODが許可され、VCCが起動電圧以下の値に下降すると(**図18**. のVBOT-)、低電圧Jセットが直ちに有効とされます。VCCが起動電圧以上に上昇すると(**図18**.のVBOT+)、(遅延タイマが起動されて)遅延タイマは遅延時間( $t_{TOUT}$ )経過後にMCUを始動します。

BOD回路は電圧が表18.で与えられるtBOD時間より長く起動電圧以下に留まる場合のみ、VCCでの低下を検出します。

# 図18. 動作中の低電圧検出リセット VCC VBOT- VBOT+ 遅延タイマ tTOUT 内部リセット

# 表18. 低電圧検出(BOD)特性

シン	ボル	項目	最小	代表	最大	単位
VH	YST	低電圧検出ヒステリシス電圧		50		mV
$t_{\mathrm{B}}$	OD	最小低電圧検出時間		2		μs

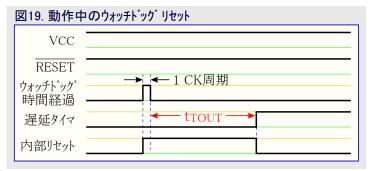
# 表17. BODLEVELヒューズ(VBOT) 設定

BODLEVEL2~0	最小(注1)	代表	最大	単位				
111	低電	王検出(BO	D)リセット禁エ	Ŀ				
1 1 0	1.7	1.8	2.0					
1 0 1	2.5	2.7	2.9	V				
1 0 0	4.1	4.3	4.5					
$0\ 0\ 0\ \sim 0\ 1\ 1$		(予約)						

注1: いくつかのデバイスでVBOTが公称最低動作電圧以下の可能性があります。この状態のデバイスについては製造検査中、VCC=VBOTに落として検査されます。これはマイクロコントローラの正しい動作がもはや保証されない電圧になる前に低電圧検出(BOD)リセットが起きることを保証します。この検査はATmega169Vに対してBODLEVEL=110を使って実行されます。

### ウォッチト ック リセット

ウォッチト・ック・時間経過時、(内部的に)1 CK周期幅の短いリセット パルスを生成します。本パルスの下降端で遅延タイマは遅延時間(t<sub>TOUT</sub>)の計時を始めます。ウォッチト・ック・タイマ操作の詳細については28頁を参照してください。



# MCU状態レジスタ (MCU Status Register) MCUSR

MCU状態レシブスタはどのリセット元がMCUリセットを起こしたかの情報を提供します。

ピット	7	6	5	4	3	2	1	0	
\$34 (\$54)	_	-	-	JTRF	WDRF	BORF	EXTRF	PORF	MCUSR
Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W	
初期値	0	0	0	内容参照	内容参照	内容参照	内容参照	内容参照	

• ビット4 - JTRF: JTAG リセット フラク (JTAG Reset Flag)

このビットはリセットがJTAG命令AVR\_RESETで選ばれたJTAG リセット レジスタ内の論理1によって引き起こされると設定(1)されます。このビットは電源ONリセットまたは、このフラグへの論理0書き込みによってリセット(0)されます。

• ビット3 - WDRF: ウォッチドッグ リセット フラグ (Watchdog Reset Flag)

このビットはウォッチドッグ リセットが起こると設定(1)されます。このビットは電源ONリセットまたは、このフラグへの論理0書き込みによってリセット(0)されます。

• ビット2 - BORF: 低電圧リセット フラク (Brown-Out Reset Flag)

このビットは低電圧リセットが起こると設定(1)されます。このビットは電源ONリセットまたは、このフラグへの論理0書き込みによってリセット(0)されます。

• ビット1 - EXTRF:外部リセット フラク (External Reset Flag)

このビットは外部リセットが起こると設定(1)されます。このビットは電源ONリセットまたは、このフラグへの論理0書き込みによってリセット(0)されます。

• ビット0 - PORF: 電源ONリセット フラク (Power-on Reset Flag)

このビットは電源ONリセットが起こると設定(1)されます。このビットはこのフラグへの論理0書き込みによってのみリセット(0)されます。

リセット条件の確認にリセット フラグを使うため、使用者はプログラム内で可能な限り早くMCUSRを読み、そして解除(0)すべきです。別のリセットが起こる前にこのレジスタが解除(0)されると、そのリセット元はリセットフラグを調べることによって得られます。

### 内部基準電圧

ATmega169は内部基準電圧が特徴です。この基準電圧は低電圧検出(BOD)に使われ、A/D変換やアナログ比較器の入力としても使えます。

# 基準電圧許可信号と起動時間

この基準電圧には使われるべき方法に影響を及ぼすかもしれない起動時間があります。この起動時間は表19.で与えられます。節電のために、この基準電圧は常にONではありません。この基準電圧は次の状態中ONです。

- 1. 低電圧検出リセット許可時 (BODLEVEL ヒュース のプログラム(0)により)
- 2. アナログ比較器基準電圧接続時 (アナログ比較器制御/状態レジスタ(ACSR)の基準電圧入力選択(ACBG)=1)
- 3. A/D変換部動作許可時 (A/D変換制御/状態レジネタA(ADCSRA)のA/D動作許可(ADEN)=1)

従って低電圧検出(BOD)が許可されていないと、ACBGの設定(=1)またはA/D変換部許可(ADEN=1)後、使用者はアナログ比較器またはA/D変換器出力が使われる前に基準電圧へ起動時間を与えなければなりません。パワーダウン動作での消費電力を減らすため、使用者はパワーダウン動作へ移行する前に基準電圧がOFFされるのを保証することで上の3つの状態を避けられます。

### 表19. 内部基準電圧特性

シンホ゛ル	項目	条件	最小	代表	最大	単位
$V_{BG}$	基準電圧		1.0	1.1	1.2	V
t <sub>BG</sub>	起動時間	VCC=2.7V TA=25°C		40	70	μs
$I_{\mathrm{BG}}$	消費電流	1A-25 C		15		μA

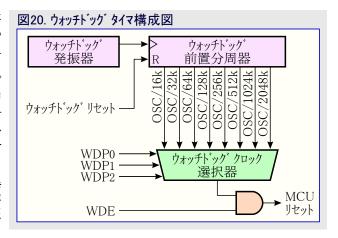




# ウォッチト゛ック゛タイマ

ウォッチト、ック、タイマは1MHzで動く独立したチップ、上の発振器からクロック駆動されます。これはVCC=5Vでの代表値です。他のVCC電圧での代表値については「特性データ」をご覧ください。ウォッチト、ック、タイマ前置分周器の制御により、ウォッチト、ック、リセット間隔は29頁の表21.で示されるように調整できます。ウォッチト、ック、リセット(WDR)命令はウォッチト、ック、タイマをリセットします。ウォッチト、ック、タイマはそれが禁止される時とチップ。リセットが起こる時もリセットされます。このリセット周期を決めるために8つの異なるクロック周期を選べます。別のウォッチト、ック、リセットなしにリセット周期が経過すると、ATmega169はリセットしてリセットへ、クタから実行します。ウォッチト、ック、リセットの詳細タイミンク、については26頁を参照してください。

予期せぬウォッチト、ッグ禁止や予期せぬ計時周期変更を防ぐため、2つの異なる安全基準が表20.で示されるWDTONヒュースによって選ばれます。詳細については29頁の「ウォッチト、ッグ・タイマ設定変更の時間制限手順」を参照してください。



### 表20、WDTONヒューズ設定によるウォッチドッグ機能設定

Ė	WDTON ヒューズ	安全基準	WDT初期状態	WDT禁止方法	計時時間変更方法	
	非プログラム(1)	1	禁止	時間制限	時間制限	
	プ <sup>°</sup> ロク <sup>*</sup> ラム(0)	2	許可	なし(常時許可)	時間制限	

# ウォッチト、ック、タイマ制御レシ、スタ (Watchdog Timer Control Register) WDTCR

ピット (\$60)	7	6	5	4 WDCE	3 WDE	2 WDP2	1 WDP1	0 WDP0	l wdtcr
Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W	WETOK
初期値	0	0	0	0	0	0	0	0	

### • ビット7~5 - Res: 予約 (Reserved)

これらのビットは予約されており、常に0として読みます。

• ビット4 - WDCE: ウォッチドッグ変更許可 (Watchdog Change Enable)

ウォッチト、ック、許可(WDE)ヒ、ットが論理のを書かれるとき、このビットは設定(1)されなければなりません。さもなければ、ウォッチト、ック、は禁止されません。一旦1を書かれると、ハート・ウェアが4クロック周期後、このビットを解除(0)します。ウォッチト、ック、禁止手順についてはWDEビットの記述を参照してください。このビットは前置分周選択ビットを変更する時も設定(1)されなければなりません。29頁の「ウォッチト、ック、タイマ設定変更の時間制限手順」をご覧ください。

• ビット3 - WDE : ウォッチドック 許可 (Watchdog Enable)

ウォッチドッグ許可(WDE)が論理1を書かれるとウォッチドッグ タイマが許可され、WDEが論理0を書かれるとウォッチドッグ タイマ機能が禁止されます。WDEはウォッチドッグ変更許可(WDCE)ビットが論理レベル1を持つ場合だけ解除(0)できます。許可されているウォッチドッグ タイマを禁止するには次の手順に従わなければなりません。

- 1. 同じ操作(命令)内で、ウォッチト、ック・変更許可(WDCE)ヒ、ットとウォッチト、ック・許可(WDE)ヒ、ットに論理1を書きます。例え禁止操作を始める前のWDEが1に設定されていても、論理1がWDEに書かれなければなりません。
- 2. 次からの4クロック周期内で、論理0をWDEに書きます。これがウォッチドック「を禁止します。

安全基準2では上記の手順でもウォッチドック、タイマを禁止することができません。29頁の「ウォッチドック、タイマ設定変更の時間制限手順」をご覧ください。

• ピット2~0 - WDP2~0: ウォッチドック゚ タイマ前置分周選択 (Watchdog Timer Prescaler 2,1 and 0)

このWDP2~0ビットはウォッチドッグタイマが許可される時のウォッチドッグタイマの前置分周を決めます。異なる前置分周値と対応する計時完了周期は表21.に示されます。

WDP2	WDP1	WDP0	WDT発振周期数	代表的な計時完了周期		
WDF2	WDF1	WDF0	WUI光派问册致	VCC=3.0V	VCC=5.0V	
0	0	0	16k (16,384)	15.4ms	14.7ms	
0	0	1	32k (32,768)	30.8ms	29.3ms	
0	1	0	64k (65,536)	61.6ms	58.7ms	
0	1	1	128k (131,072)	0.12s	0.12s	
1	0	0	256k (262,144)	0.25s	0.23s	
1	0	1	512k (524,288)	0.49s	0.47s	
1	1	0	1024k (1,048,576)	1.0s	0.9s	
1	1	1	2048k (2,097,152)	2.0s	1.9s	

注: 216頁の図191.もご覧ください。

次のコート・例はウォッチト、ッグ(WDT)をOFFに切り替えるアセンブリ言語とC言語の関数を示します。本例は(例えば全割り込み禁止によって)割り込みが制御され、それ故これらの関数実行中に割り込みが起きない前提です。

```
アセンブリ言語プログラム例
WDT_OFF:
            WDR
                                                         ;ウォッチトック タイマ リセット
                    R16, WDTCR
                                                         ;現WDTCR値を取得
            IN
            ORI
                    R16, (1<<WDCE) | (1<<WDE)
                                                         ;WDCEとWDE論理1値を設定
                                                         ;WDCEとWDEに論理1書き込み
            OUT
                    WDTCR, R16
            LDI
                    R16, (0<<WDE)
                                                         ;WDE論理0値を取得
            OUT
                    WDTCR, R16
                                                         ;ウォッチトック 禁止
                                                         ;呼び出し元へ復帰
            RET
C言語プログラム例
void WDT_off(void)
                                                         /* ウォッチトック タイマ リセット */
      _watchdog_reset();
                                                         /* WDCEとWDEに論理1書き込み */
    WDTCR \mid = (1 << \text{WDCE}) \mid (1 << \text{WDE});
    WDTCR = 0x00;
                                                         /* ウォッチト`ック`禁止 */
注: 5頁の「コート 例について」をご覧ください。
```

# ウォッチドッグタイマ設定変更の時間制限手順

ウォッチドッグ設定変更手順は安全基準間で僅かに異なります。各レベルに対して独立した手順が記述されます。

### 安全基準1

この動作種別ではウォッチト、ック、タイマが初めに禁止されますが、どんな制限もなくウォッチト、ック、許可(WDE)ヒットに1を書くことにより許可できます。ウォッチト、ック、計時完了周期を変更または許可したウォッチト、ック、タイマを禁止する時に時間制限手順が必要とされます。許可したウォッチト、ック、タイマを禁止や、ウォッチト、ック、計時完了周期を変更するには次の手順に従わなければなりません。

- 1. 同じ操作(命令)でウォッチト、ッグ変更許可(WDCE)とWDEに論理1を書きます。WDEビットの直前の値に拘らず、論理1がWDEに書かれなければなりません。
- 2. 次からの4クロック周期内に同じ操作(命令)で欲したWDEとウォッチドッグタイマ前置分周選択(WDP2~0)ビットを書きますが、WDCEビットは解除(0)されてです。

# 安全基準2

この動作種別ではウォッチドッグ タイマが常に許可され、WDEビットは常に1として読みます。時間制限手順はウォッチドッグ計時完了周期を変更する時に必要とされます。ウォッチドッグ計時完了周期を変更するには次の手順に従わなければなりません。

- 1. 同じ操作(命令)でWDCEとWDEに論理1を書きます。WDEビットが常に設定(1)されていても、WDEは時間制限手順を開始するために1を書かれなければなりません。
- 2. 次からの4クロック周期内に同じ操作(命令)で欲したWDP2~0ビットを書きますが、WDCEビットは解除( $\frac{0}{0}$ )されてです。WDEビットに書かれた値は無関係です。





# 割り込み

本章はATmega169によって実行される割り込み操作の詳細を記述します。AVR割り込み操作の一般説明については9頁の「リセットと 割り込みの扱い」を参照してください。

# ATmega169の割り込みべかり

表22. リセットと割り込みのベクタ

ベクタ番号	フ <sup>°</sup> ログラム アト・レス ( <mark>注2</mark> )	発生元	備考		
1	\$0000 ( <b>注1</b> )	リセット	電源ON, WDT, BOD等の各種リセット		
2	\$0002	INT0	外部割り込み要求0		
3	\$0004	PCINTO (PCIO)	ピン変化0群割り込み要求		
4	\$0006	PCINT1 (PCI1)	ピン変化1群割り込み要求		
5	\$0008	タイマ/カウンタ2 COMP2	タイマ/カウンタ2比較一致		
6	\$000A	タイマ/カウンタ2 OVF2	タイマ/カウンタ2溢れ		
7	\$000C	タイマ/カウンタ1 CAPT	タイマ/カウンタ1捕獲発生		
8	\$000E	タイマ/カウンタ1 COMPA	タイマ/カウンタ1比較A一致		
9	\$0010	タイマ/カウンタ1 COMPB	タイマ/カウンタ1比較B一致		
10	\$0012	タイマ/カウンタ1 OVF1	タイマ/カウンタ1溢れ		
11	\$0014	タイマ/カウンタ0 COMP0	タイマ/カウンタ0比較一致		
12	\$0016	タイマ/カウンタ0 OVF0	タイマ/カウンタ0溢れ		
13	\$0018	SPI STC	SPI 転送完了		
14	\$001A	USART RX	USART 受信完了		
15	\$001C	USART UDRE	USART 送信緩衝部空き		
16	\$001E	USART TX	USART 送信完了		
17	\$0020	USI START	USI 開始条件検出		
18	\$0022	USI OVF	USI 計数器溢れ		
19	\$0024	アナログ比較器 ANA_COMP	アナログ比較器出力遷移		
20	\$0026	A/D変換器 ADC	A/D変換完了		
21	\$0028	EEPROM EE_RDY	EEPROM 操作可		
22	\$002A	SPM命令 SPM_RDY	SPM命令操作可		
23	\$002C	LCD	LCDフレーム開始		

注1: BOOTRSTヒューズがブプログラム(0)されると、デハーイスはリセットでフートロータ、アトレスへ飛びます。166頁の「ブートロータ・支援 - 書き込み 中読み出し可能な自己プログラミング」をご覧ください。

注2: MCU制御レシ、スタ(MCUCR)の割り込みへ、クタ選択(IVSEL)ビットが設定(1)されると、割り込みへ、クタはブート フラッシュ領域先頭(部)へ 移動されます。そして各割り込みベクタのアドレスは、この表のアドレスがブート フラッシュ領域の先頭アドレスに加算されます。

表23.はBOOTRST(ヒューズ)とIVSEL(割り込みベクタ 表23. リセットと割り込みベクタの配置 選択ビット)の様々な組み合わせに対するリセットと割 り込みベクタの配置を示します。プログラムが決して 割り込み元を許可しないなら、割り込みベクタは使 われず、これらの位置に通常のプログラムコートを置 けます。これはリセット ヘンクタが応用領域の一方、割 り込みベクタがブート領域、またはその逆の場合で も同様です。

BOOTRST	IVSEL リセット ベクタ アドレス		割り込みベクタ先頭アドレス						
非プログラム(1)	0	\$0000	\$0002						
チ/ ログ /A(1)	1	\$0000	ブート領域先頭アドレス+\$0002						
プログラム(0)	0	ブート領域先頭アトレス	\$0002						
	1	ブート領域先頭アトレス	ブート領域先頭アトンス+\$0002						

**注**: ブート領域先頭アトレスは174頁の**表113**.で示されます。

ATmega169での最も代表的且つ一般的なリセットと割り込みのベクタ アドレス用設定を次に示します。

アト・レス	ラヘ゛ル	命令		注釈
\$0000		JMP	RESET	;各種リセット
\$0002		JMP	EXT_INTO	;外部割り込み要求0
\$0004		JMP	PCINTO	;ピン変化0群割り込み要求
\$0006		JMP	PCINT1	;ピン変化1群割り込み要求
\$0008		JMP	TIM2_COMP	;タイマ/カウンタ2比較一致
\$000A		JMP	TIM2_OVF	;タイマ/カウンタ2溢れ
\$000C		JMP	TIM1_CAPT	;タイマ/カウンタ1捕獲発生
\$000E		JMP	TIM1_COMPA	;タイマ/カウンタ1比較A一致
\$0010		JMP	TIM1_COMPB	;タイマ/カウンタ1比較B一致
\$0012		JMP	TIM1_OVF	;タイマ/カウンタ1溢れ
\$0014		JMP	TIMO_COMP	;タイマ/カウンタ0比較一致
\$0016		JMP	TIMO_OVF	;タイマ/カウンタ0溢れ
\$0018		JMP	SPI_STC	;SPI転送完了
\$001A		JMP	USART_RXC	;USART 受信完了
\$001C		JMP	USART_UDRE	;USART 送信緩衝部空
\$001E		JMP	USART_TXC	;USART 送信完了
\$0020		JMP	USI_STRT	;USI 開始条件検出
\$0022		JMP	USI_OVF	;USI 計数器溢れ
\$0024		JMP	ANA_COMP	;アナログ比較器出力遷移
\$0026		JMP	ADC	;ADC変換完了
\$0028		JMP	EE_RDY	;EEPROM操作可
\$002A		JMP	SPM_RDY	;SPM命令操作可
\$002C		JMP	LCD_SOF	;LCDフレーム開始
;				
\$002E	RESET:	LDI	R16, HIGH (RAMEND)	;RAM最終アドレス上位を取得
\$002F		OUT	SPH, R16	;スタック ポインタ上位を初期化
\$0030		LDI	R16, LOW (RAMEND)	;RAM最終アドレス下位を取得
\$0031		OUT	SPL, R16	;スタック ポインタ下位を初期化
		>		;以下、I/O初期化など

BOOTRSTヒューズが非プログラム(1)、ブート領域容量が2Kバイトに設定され、どの割り込みが許可されるのにも先立ってMCU制御レジスタ (MCUCR)の割り込みペクタ選択(IVSEL)ビットが設定(1)される時の最も代表的且つ一般的なリセットと割り込みのベクタ アドレス用設定を次に示します。

アト・レス	ラヘ゛ル	命令		注釈
\$0000 \$0001 \$0002 \$0003	RESET:	RESET: LDI R16, HIGH (RAMEND) OUT SPH, R16 LDI R16, LOW (RAMEND) OUT SPL, R16		;RAM最終アトンス上位を取得(応用プログラム開始) ;スタック ポインタ上位を初期化 ;RAM最終アトンス下位を取得 ;スタック ポインタ下位を初期化 ;以下、I/O初期化など
		. ORG	\$1C02	;ブート プログラム領域が2Kバイトの場合
\$1C02 \$1C04		JMP JMP	EXT_INTO PCINTO	;外部割り込み要求0 ;ピン変化0群割り込み要求
\$1C2C		JMP	LCD_SOF	;LCDフレーム開始





BOOTRSTヒューズがプログラム(0)、ブート領域容量が2Kバイトに設定される時の最も代表的且つ一般的なリセットと割り込みのベクタ アドレス用設定を次に示します。

アト・レス	ラヘ゛ル	命令		注釈
		. ORG	\$0002	;割り込みベクタ先頭
\$0002		JMP	EXT_INTO	;外部割り込み要求0
\$0004		JMP	PCINTO	;ピン変化0群割り込み要求
\$002C		JMP /	LCD_SOF	;LCDフレーム開始 ;以下、プログラムなど
		. ORG	\$1C00	;ブートプログラム領域が2Kバイトの場合
\$1C00	RESET:	LDI	R16, HIGH (RAMEND)	;RAM最終アドレス上位を取得(プログラム開始)
\$1C01		OUT	SPH, R16	;スタック ポインタ上位を初期化
\$1C02		LDI	R16, LOW (RAMEND)	;RAM最終アドレス下位を取得
\$1C03		OUT /	SPL, R16	;スタック ポインタ下位を初期化 ;以下、I/O初期化など

BOOTRSTヒューズがプログラム(0)、ブート領域容量が2Kバイトに設定され、どの割り込みが許可されるのにも先立ってMCU制御レシ、スタ (MCUCR)の割り込みへ、クタ選択(IVSEL)ビットが設定(1)される時の最も代表的且つ一般的なリセットと割り込みのへ、クタアトレス用設定を次に示します。

アト・レス	ラヘ゛ル	命令		注釈
		. ORG	\$1C00	;ブート プログラム領域が2Kバイトの場合
\$1C00 \$1C02		JMP TMD	RESET EXT INTO	;各種リセット (BOOTRSTヒューズ=0) ;外部割り込み要求0
\$1C02 \$1C04		JMP .TMP	PCINTO	;ピン変化0群割り込み要求
Ψ1001		)	1011110	70 v 久間 vi i i i i i i i i i i i i i i i i i i
\$1C2C ;		JMP	LCD_SOF	;LCDフレーム開始
\$1C2E	RESET:	LDI	R16, HIGH (RAMEND)	;RAM最終アドレス上位を取得(プログラム開始)
\$1C2F		OUT	SPH, R16	;スタック ポインタ上位を初期化
\$1C30		LDI	R16, LOW (RAMEND)	;RAM最終アドレス下位を取得
\$1C31		OUT /	SPL, R16	;スタック ポインタ下位を初期化 ;以下、I/O初期化など

# 応用領域とブート領域間の割り込みベクタ移動

MCU制御レジスタ(MCUCR)は割り込みへ、クタ表の配置を制御します。

# MCU制御レジスタ (MCU Control Register) MCUCR

ピット	7	6	5	4	3	2	1	0	_
\$35 (\$55)	JTD	-	-	PUD	-	-	IVSEL	IVCE	MCUCR
Read/Write	R/W	R	R	R/W	R	R	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

### • ビット1 - IVSEL: 割り込みベクタ選択 (Interrupt Vector Select)

IVSELビットが解除(0)されると、割り込みベクタはフラッシュメモリの先頭に配置されます。このビットが設定(1)されると、割り込みベクタはフラッシュメモリのブート ローダ領域の始まりへ移動されます。ブート フラッシュ領域先頭の実際のアドレスはBOOTSZヒューズによって決定されます。詳細については166頁の「**ブート ローダ支援 - 書き込み中読み出し可能な自己プログラミング**」章を参照してください。割り込みベクタ表の予期せぬ変更を防ぐため、IVSELビットの変更は特別な書き込み手順に従わなければなりません。

- 1. 割り込みベクタ変更許可(IVCE)ビットにlを書きます。
- 2. 4周期内に、欲した値をIVSELに書き、同時に0をIVCEへ書きます。

この手順が実行される間、割り込みは自動的に禁止されます。割り込みはIVCEが設定(1)される周期で禁止され、後続のIVSELに書く命令の後まで禁止されたままです。IVSELが書かれなければ、割り込みは4周期間禁止されたままです。ステータスレジスタ(SREG)の全割り込み許可(I)ビットはこの自動禁止によって影響されません。

注: 割り込みベクタがブート ローダ領域に配置され、BLB02ブート施錠ビットがプログラム(0)されると、応用領域から実行する間中、割り込みが禁止されます。割り込みベクタが応用領域に配置され、BLB12プート施錠ビットがプログラム(0)されると、ブート ローダ領域から実行する間中、割り込みが禁止されます。ブート施錠ビットの詳細については166頁の「ブート ローダ支援 - 書き込み中読み出し可能な自己プログラミング」を参照してください。

### • ビット0 - IVCE: 割り込みベクタ変更許可 (Interrupt Vector Change Enable)

IVCEビットは割り込みベクタ選択(IVSEL)ビットの変更を許可するために論理1を書かれなければなりません。IVCEはIVSELが書かれる時、またはIVCEが書かれた後の4周期後、ハードウェアによって解除(0)されます。上記IVSELで説明されるようにIVCEビットの設定(1)は割り込みを(一時的に)禁止します。以下のコート・例をご覧ください。

```
アセンブリ言語プログラム例
MOVE IVT:
           IN
                  R16, MCUCR
                                                      ;現MCUCR値取得
                                                      ;現MCUCR値複写
           MOV
                  R17, R16
           ORI
                  R16, (1<<IVCE)
                                                      ;IVCE論理1値を取得
                  MCUCR, R16
                                                      ;IVCEに論理1書き込み
           OUT
           ORI
                  R17, (1<<IVSEL)
                                                      ;IVSEL論理1値を取得
                                                      ;ブート領域へ割り込みへづクタを移動
           OUT
                  MCUCR, R17
           RET
                                                      ;呼び出し元へ復帰
C言語プログラム例
void Move_interrupts(void)
                                                     /* 一時定数定義 */
    uchr temp;
                                                     /* 現MCUCR値取得 */
    temp = MCUCR;
    MCUCR = temp | (1 << IVCE);
                                                     /* IVCEに論理1書き込み */
    MCUCR = temp | (1<<IVSEL);</pre>
                                                     /* ブート領域へ割り込みへづりを移動 */
```



# 外部割り込み

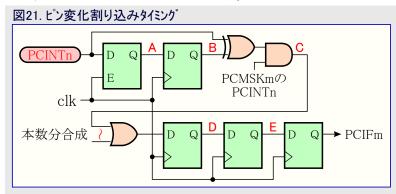
外部割り込みはINT0ピンまたはPCINT0~15ピンの何れかによって起動されます。許可したなら、例えINT0またはPCINT0~15ピンが出力として設定されても、割り込みが起動することに注目してください。この特徴はソフトウェア割り込みを生成する方法を提供します。ピン変化割り込みPCI1は許可したPCINT8~15の何れかが切り替わると起動します。ピン変化割り込みPCI0は許可したPCINT0~7の何れかが切り替わると起動します。ピン変化割り込み許可レジスタn(PCMSK0とPCMSK1)は、どのピンがピン変化割り込み要因となるかを制御します。PCINT0~15でのピン変化割り込みは非同期に検知されます。これはそれらの割り込みがアイドル動作以外の休止形態からもデバイスを起動するのに使えることを意味します。

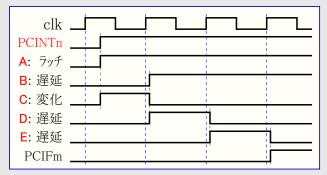
INT0割り込みは上昇端または下降端(含む両端)、またはLowレヘルによって起動できます。これは外部割り込み制御レジスタA(EICRA)の詳述で示される設定です。INT0割り込みがレヘル起動として設定、且つ許可されると、そのピンがLowに保持される限り、割り込みは(継続的に)起動します。INT0の上昇端または下降端割り込みの認知は、16頁の「クロック系統とその配給」で記述されるI/Oクロックの存在を必要とすることに注意してください。INT0のLowレヘル割り込みは非同期に検知されます。これはそれらの割り込みがアイル動作以外の休止形態からも、デバイスを起動するのに使えることを意味します。I/Oクロックはアイドル動作を除く全休止形態で停止されます。

レヘル起動割り込みがハプワーダウン動作からの起動に使われる場合、この必要としたレヘルはレヘル割り込みを起動する完全な起動復帰のため、MCUに対して充分長く保持されなければならないことに注意してください。このレヘルが起動時間の最後に先立って消滅すると、MCUは今までどおり起動しますが、割り込みが生成されません。起動時間は16頁の「システム クロックとクロック選択」で示されるようにSUTヒュース、とCKSELヒュース、で定義されます。

# ピン変化割り込みタイミング

ピン変化割り込みの例は図21.で示されます。





# 外部割り込み制御レジスタA (External Interrupt Control Register A) EICRA

外部割り込み制御レジスタAは割り込み条件制御ビットを含みます。

ピット (\$69)	7 	6 –	5 —	4 –	3	2	1 ISC01	0 ISC00	EICRA
Read/Write	R	R	R	R	R	R	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

# ● ビット1,0 - ISC01,0:外部割り込み0条件制御 (Interrupt Sense Control 0 bit1 and 0)

外部割り込み0はステータスレジスタ(SREG)の全割り込み許可(I)ビットと外部割り込み許可レジスタ(EIMSK)の外部割り込み0許可(INTO)ビットが設定(1)される場合のINTO 外部ピンによって活性(有効)にされます。割り込みを活性にする外部INTOピンの端(エッジ)とレヘルは表24.で定義されます。INTOピンの値は端検知前に採取されています。端または論理変化割り込みが選ばれる場合、1クロック周期より長く留まるパルスは割り込みを生成します。より短いパルスは割り込みの発生が保証されません。Lowレヘル割り込みが選ばれる場合、Lowレヘルは割り込みを生成するために現在実行している命令の完了まで保たれなければなりません。

表24. 外部割り込み0(INT0)割り込み条件							
ISC01	ISC00	INTOピン割り込み発生条件					
0	0	Lowレベル。					
0	1	論理変化(両端)					
1	0	下降端					
1	1	上昇端					

# 外部割り込み許可レジスタ (External Interrupt Mask Register) EIMSK

ピット	7	6	5	4	3	2	1	0	
\$1D (\$3D)	PCIE1	PCIE0	-	-	-	-	-	INT0	EIMSK
Read/Write	R/W	R/W	R	R	R	R	R	R/W	
初期値	0	0	0	0	0	0	0	0	

### • ビット7 - PCIE1: ピン変化1群割り込み許可 (Pin Change Interrupt Enable 1)

ステータスレジスタ(SREG)の全割り込み許可(I)ビットが設定(1)され、PCIE1ビットが設定(1)されると、ピン変化割り込み1が許可されます。許可したPCINT8~15ピンの何れかの変化が割り込みを起こします。このピン変化割り込み要求に対応する割り込みはPCI1割り込みへプタから実行されます。PCINT8~15ピンはピン変化割り込み許可レジスタ1(PCMSK1)によって個別に許可されます。

# • ビット6 - PCIEO: ピン変化0群割り込み許可 (Pin Change Interrupt Enable 0)

ステータスレジスタ(SREG)の全割り込み許可(I)ビットが設定(1)され、PCIE0ビットが設定(1)されると、ピン変化割り込みのが許可されます。許可したPCINT0~7ピンの何れかの変化が割り込みを起こします。このピン変化割り込み要求に対応する割り込みはPCI0割り込みへ、クタから実行されます。PCINT0~7ピンはピン変化割り込み許可レジスタの(PCMSKO)によって個別に許可されます。

# • ビット0 - INTO:外部割り込み0許可 (External Interrupt Request 0 Enable)

ステータスレジスタ(SREG)の全割り込み許可(I)ビットが設定(1)され、INT0ビットが設定(1)されると、INT0外部ピン割り込みが許可されます。外部割り込み制御レジスタA(EICRA)の割り込み条件制御0のビット1と0(ISC01,ISC00)は、この外部割り込みがINT0ピンの上昇端、下降端、両端、またはLowレヘルのどれで活性(有効)にされるかを定義します。例えINT0ピンが出力として設定されても、このピンの動きは割り込み要求を引き起こします。外部割り込み要求のに対応する割り込みはINT0割り込みへ、クタから実行されます。

### 外部割り込み要求フラグレジスタ (External Interrupt Flag Register) EIFR

ピット	7	6	5	4	3	2	1	0	
\$1C (\$3C)	PCIF1	PCIF0	-	-	-	-	-	INTF0	EIFR
Read/Write	R/W	R/W	R	R	R	R	R	R/W	
初期値	0	0	0	0	0	0	0	0	

### • ビット7 - PCIF1 : ピン変化1群割り込み要求フラグ(Pin Change Interrupt Flag 1)

PCINT8~15 ピンの何れかの論理変化が割り込み要求を起動すると、PCIF1が設定(1)になります。ステータスレジスタ(SREG)の全割り込み許可(I)ビットと外部割り込み許可レジスタ(EIMSK)のピン変化1群割り込み許可(PCIE1)ビットが設定(1)なら、MCUは対応する割り込み ^ クタへ飛びます。このフラグは割り込み処理ルーチンが開始されると解除(0)されます。代わりにこのフラグは論理1を書くことによっても解除(0)できます。

### • ビット6 - PCIF0: ピン変化0群割り込み要求フラグ (Pin Change Interrupt Flag 0)

PCINT0~7ピンの何れかの論理変化が割り込み要求を起動すると、PCIF0が設定(1)になります。ステータスレジスタ(SREG)の全割り込み許可(I)ビットと外部割り込み許可レジスタ(EIMSK)のピン変化0群割り込み許可(PCIE0)ビットが設定(1)なら、MCUは対応する割り込みへ、クタへ飛びます。このフラグは割り込み処理ルーチンが開始されると解除(0)されます。代わりにこのフラグは論理1を書くことによっても解除(0)できます。

### • ビット0 - INTFO:外部割り込み0要求フラグ(External Interrupt Flag0)

INT0ピン上の端(エッジ)または論理変化が割り込み要求を起動すると、INTF0が設定(1)になります。ステータスレシ、スタ(SREG)の全割り込み許可(I)ビットと外部割り込み許可レシ、スタ(EIMSK)の外部割り込み0許可(INT0)ビットが設定(1)なら、MCUは対応する割り込みへ、クタへ飛びます。このフラク、は割り込みルーチンが実行されると解除(0)されます。代わりにこのフラク、は論理1を書くことによっても解除(0)できます。INT0がレヘ、ル割り込みとして設定されると、このフラク、は常に解除(0)されます。





# ピン変化割り込み許可レジスタ1 (Pin Change Enable Mask 8~15) PCMSK1

ピット	7	6	5	4	3	2	1	0	
(\$6C)	PCINT15	PCINT14	PCINT13	PCINT12	PCINT11	PCINT10	PCINT9	PCINT8	PCMSK1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

### • ビット7~0 - PCINT15~PCINT8: ピン変化割り込み15~8許可 (Pin Change Enable Mask 15~8)

各PCINT8~15ビットは対応するI/Oピンでピン変化割り込みが許可されるかどうかを選びます。PCINT8~15と外部割り込み許可レジスタ (EIMSK)のPCIE1が設定(1)なら、対応するI/Oピンのピン変化割り込みが許可されます。PCINT8~15が解除( $^{0}$ )されると、対応するI/O ピンのピン変化割り込みは禁止されます。

# ピン変化割り込み許可レジスタO (Pin Change Enable Mask 0~7) PCMSKO

ピット	7	6	5	4	3	2	1	0	
(\$6B)	PCINT7	PCINT6	PCINT5	PCINT4	PCINT3	PCINT2	PCINT1	PCINT0	PCMSK0
Read/Write	R/W								
初期値	0	0	0	0	0	0	0	0	

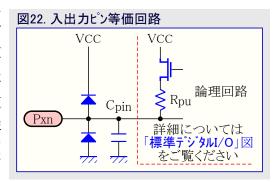
### • ビット7~0 - PCINT7~PCINT0: ピン変化割り込み7~0許可 (Pin Change Enable Mask 7~0)

各PCINT0~7ビットは対応するI/Oピンでピン変化割り込みが許可されるかどうかを選びます。PCINT0~7と外部割り込み許可レジスタ (EIMSK)のPCIE0が設定(1)なら、対応するI/Oピンのピン変化割り込みが許可されます。PCINT0~7が解除(0)されると、対応するI/Oピンのピン変化割り込みは禁止されます。

# 入出力ポート

### 序説

全てのAVRのポートは標準デッタルI/Oポートとして使われる時に真の読み-変更-書き (リート・モディファイライト)を機能的に持ちます。これはSBIとCBI命令で他のどのピンの方向をも無意識に変更することなく、1つのポートピンの方向を変更できることを意味します。(出力として設定されていれば)駆動値を変更、または(入力として設定されていれば)アルアップ抵抗を許可/禁止する時にも同じく適用されます。各出力緩衝部は高い吐き出し(ソース)と吸い込み(シンケ)の両能力で対称的な駆動特性を持ちます。このピン駆動部はLED(表示器)を直接駆動するのに充分な強さです。全てのポートピンは個別に選択可能な、供給電圧で抵抗値が変化しないプルアップ抵抗を持っています。全てのI/Oピンは図22.で示されるようにVCCとGNDの両方に保護ダイオートを持っています。各値の完全な一覧については194頁の「電気的特性」を参照してください。



本章内の全てのレジスタとビットの参照は一般形で記されます。小文字の'x'はポート番号文字、小文字の'n'はビット番号を表します。けれどもプログラム内でレジスタやビット定義に使うとき、正確な形式(例えば、ここで一般に記されたPORTxnがポートBのビット3に対してはPORT B3)が使われなければなりません。物理的なI/Oレジスタとビット位置は51頁の「I/Oポート用レジスタ」で一覧されます。

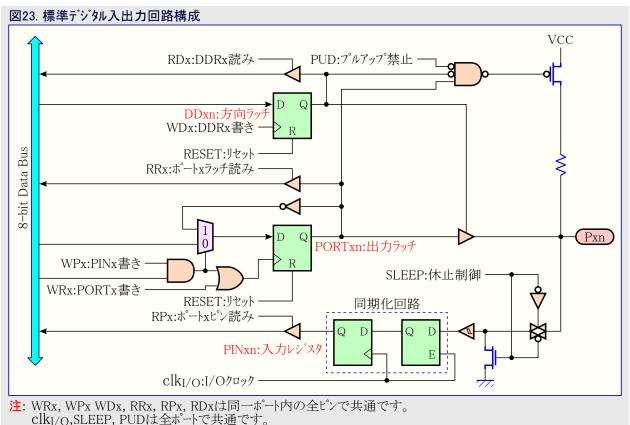
各々1つの出力レジスタ(PORTx)、方向レジスタ(DDRx)、入力レジスタ(PINx)の各ポートに対して、3つI/Oメモリアドレス位置が割り当てられます。入力レジスタのI/O位置は読むだけで、一方出力レジスタと方向レジスタは読み書き(両方)です。けれどもPINxレジスタのビットへの論理1書き込みは、出力レジスタの対応ビット値を(1/0)反転する結果になります。加えてMCU制御レジスタ(MCUCR)のプルアップ禁止(PUD)ビットが設定(1)されると、全ポートで全ビットに対してプルアップ機能を禁止します。

標準デジタルI/OとしてのI/Oポートの使用は次の「標準デジタル入出力としてのポート」で記述されます。多くのポート ピンはデバイスの周辺機能用の交換機能と多重化されます。ポート ピンとの各交換機能のインターフェース法は40頁の「交換ポート機能」で記述されます。交換機能の完全な記述については個別機能部項目を参照してください。

ポート ピンのいくつかの交換機能の許可は、そのポート内の他のピンの標準デジタル入出力としての使用に影響しないことに注意してください。

### 標準デジタル入出力としてのポート

このポートは任意の内部プルアップ付き双方向I/Oポートです。図23.はここで属にPxnと呼ばれるI/Oポートピンの1つの機能説明を示します。





#### ピンの設定

各ポート ピンは3つのレジスタ ビットDDxn、PORTxn、PINxnから成ります。51頁の「I/Oホート用レジスタ」で示されるように、DDxnビットはDDRx I/Oアドレス、PORTxnビットはPORTx I/Oアドレス、PINxビットはPINx I/Oアドレスでアクセスされます。

DDRxレジスタ内のDDxnビットはそのピンの方向を選びます。DDxnが論理1を書かれるとPxnは出力ピンとして設定されます。DDxnが論理0を書かれるとPxnは入力ピンとして設定されます。

そのピンが入力ピンとして設定される時にPORTxnが論理1を書かれると、プルアップ抵抗が活性(有効)にされます。プルアップ抵抗をOFF に切り替えるにはPORTxnが論理0を書かれるか、またはそのピンが出力ピンとして設定されなければなりません。ポートピンは例えクロックが動いていなくても、リセット条件が活性(有効)になるとHi-Zにされます。

そのピンが出力ピンとして設定される時にPORTxnが論理1を書かれると、そのポート ピンはHigh(1)に駆動されます。そのピンが出力ピンとして設定される時にPORTxnが論理0を書かれると、そのポート ピンはLow(0)に駆動されます。

#### ピンの出力交互切り替え

PINxnへの論理1書き込みはDDRxnの値に拘らず、PORTxnの値を反転切り替えします。SBI命令がポート内の1ビットの反転切り替えに使えることに注目してください。

### 入出力間の切り替え

Hi-Z入力(DDxn=0, PORTxn=0)とHigh出力(DDxn=1, PORTxn=1)間の切り替え時、プルアップ許可入力(DDxn=0, PORTxn=1)または Low出力(DDxn=1, PORTxn=0)のどちらかの中間状態が生じるに違いありません。通常、高インピーダンス環境は強力なHigh(吐き出し) 駆動部とプルアップ間の違いに気付かないので、プルアップが許可された状態は十分受け入れられます。この事例でないなら、全ポートの全プルアップを禁止するために、MCU制御レジスタ(MCUCR)のプルアップ禁止(PUD)ビットが設定(1)できます。

プルアップ入力とLow出力間の切り替えは同じ問題を発生します。使用者は中間状態としてHi-Z入力(DDxn=0, PORTxn=0)またはHigh 出力(DDxn=1, PORTxn=1)のどちらかを使わなければなりません。

表25.はピン値に対する制御信号の一覧を示します。

表25. ポート ピンの設定	表25.	ポート	ピンの	設定
----------------	------	-----	-----	----

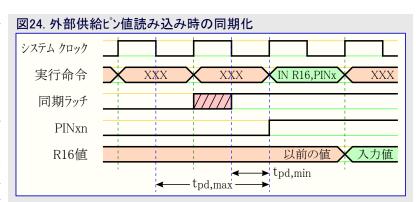
DDxn	PORTxn	PUD (MCUCR)	入出力	プルアップ抵抗	備考
0	0	X	入力	なし	高インヒ゜ーダンス (Hi-Z)
0	1	0	入力	あり	Pxnに外部からLowを入力すると吐き出し電流が流れます。
0	1	1	入力	なし	高インピーダンス (Hi-Z)
1	0	X	出力	なし	Low (吸い込み)出力
1	1	X	出力	なし	High (吐き出し)出力

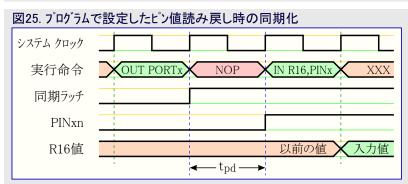
### ピン値の読み込み

DDxn方向ビットの設定に関係なく、ポート ピンはPINxnレシスタ ビットを通して読めます。図23.で示されるようにPINxnレシスタ ビットを通して読めます。図23.で示されるようにPINxnレシスタ ビットと先行するラッチは同期化回路を構成します。これは物理ピンが内部クロックの端(エッジ)付近で値を変える場合の未定義状態(メタ ステーブル)を避けるために必要とされますが、それは遅延も持ち込みます。図24.は外部的に加えられたピン値を読む時の同期化タイミング図を示します。伝播遅延の最小と最大は各々tpd,minとtpd,maxで示されます。

(図24.で)システム クロックの最初の下降端の直後から始まるクロック周期を考察してください。このラッチはクロックがLowの時に閉じ、クロックがHighの時に同期ラッチ信号の斜線部分で示されるように通過(トランスペプレント)となります。この信号値はシステム クロックがLowになる時に保持(ラッチ)されます。それが続くクロックの上昇端でPINxnレジスタに取り込まれます。2つの矢印tpd,minとtpd,maxによって示されるように、ピン上の単一信号遷移は出現時点に依存して0.5~1.5システムクロック周期遅らされます。

ソフトウェアが指定したピン値を読み戻す時は、図25.で示されるようにNOP命令が挿入されなければなりません。OUT命令はシステムクロックの上昇端で同期ラッチを設定します。この場合、同期化回路を通過する遅延時間(tpd)は1システムクロック周期です。





次のコート例はポートBピンの0と1をHigh出力、2と3をLow出力、6と7をプルアップ指定として4~7を入力に設定する方法を示します。結果のピン値が再び読み戻されますが、前記で検討されたように、いくつかのピンへ直前に指定された値を読み戻すことができるようにNOP命令が挿入されます。

```
アセンブリ言語プログラム例
            LDI
                   R16, (1<<PB7) | (1<<PB6) | (1<<PB1) | (1<<PB0)
                                                                ;プルアップとHigh値を取得
                   R17, (1<<DDB3) | (1<<DDB2) | (1<<DDB1) | (1<<DDB0)
                                                                ;出力ビット値を取得
            LDI
                   PORTB, R16
                                                                ;プルアップとHigh値を設定
            OUT
            OUT
                   DDRB, R17
                                                                ;入出力方向を設定
            NOP
                                                                ;同期化遅延対処
                                                                ;ピン値読み戻し
            IN
                   R16, PINB
C言語プログラム例
unsigned char i;
                                                                /* */
    PORTB = (1 << PB7) | (1 << PB6) | (1 << PB1) | (1 << PB0);
                                                                /* プルアップとHigh値を設定 */
    DDRB = (1<<DDB3) | (1<<DDB2) | (1<<DDB1) | (1<<DDB0);
                                                                /* 入出力方向を設定 */
                                                                /* 同期化遅延対処 */
    __no_operation();
                                                                /* ピン値読み戻し*/
    i = PINB;
                                                                /* */
```

**注**: アセンブリ言語プログラムについてはプルアップがピン0,1,6,7に設定されてから、ビット0と1の強力なHigh駆動部としての再定義、ビット2と3のLow駆動部としての定義、方向ビットが正しく設定されるまでの時間を最小とするために2つの一時レジスタが使われます。

### デジタル入力許可と休止形態

図23.で示されるようにデジタル入力信号はシュミットトリカの入力をGNDにクランプできます。この図でSLEEPと印された信号は入力信号のいくつかが開放のまま、またはVCC/2付近のアナログ信号電圧を持つ場合の高消費電力を避けるため、パワータヴン動作、パワーセーブ動作、スタンバイ動作でMCU休止制御器によって設定(1)されます。

SLEEPは外部割り込みピンとして許可されたポート ピンに対しては無視されます。外部割り込み要求が許可されないなら、SLEEPは他のピンについてと同様に有効です。SLEEPは40頁の「**交換ポート機能**」で記載されるように様々な他の交換機能によっても無視されます。

外部割り込みが許可されていない"上昇端、下降端または論理変化(両端)割り込み"として設定された非同期外部割り込みピンに論理1が存在すると、上で言及した休止形態から(復帰)再開する時に、これらの休止形態に於けるクランプが要求された論理変化を生ずるので、対応する外部割り込み要求フラグが設定(=1)されます。

#### 未接続ピン

いくつかのピンが未使用にされる場合、それらのピンが定義されたレベルを持つのを保証することが推奨されます。例え上記のような深い休止形態で多くのデジタル入力が禁止されるとしても、デジタル入力が許可される他の全ての動作(リセット、活動動作、アイドル動作)で消費電流削減のため、浮き状態入力は避けられるべきです。

未使用ピンの定義されたレベルを保証する最も簡単な方法は内部プルアップを許可することです。この場合、リセット中のプルアップは禁止されます。 リセット中の低消費電力が重要なら、外部プルアップまたはプルダウンを使うことが推奨されます。 未使用ピンを直接GNDまたはVCCに接続することは、ピンが偶然に出力として設定されると過電流を引き起こす可能性があるため推奨されません。





### 交換ポート機能

多くのポート ピンには標準デジタル入出力に加え交換機能があります。図26.は単純化された図23.でのポート ピン制御信号が交換機能によってどう重複できるかを示します。この重複信号は全てのポート ピンに存在する訳ではありませんが、この図はAVR マイクロコントローラ系統の全ポート ピンに適用できる一般的な記述として取り扱います。

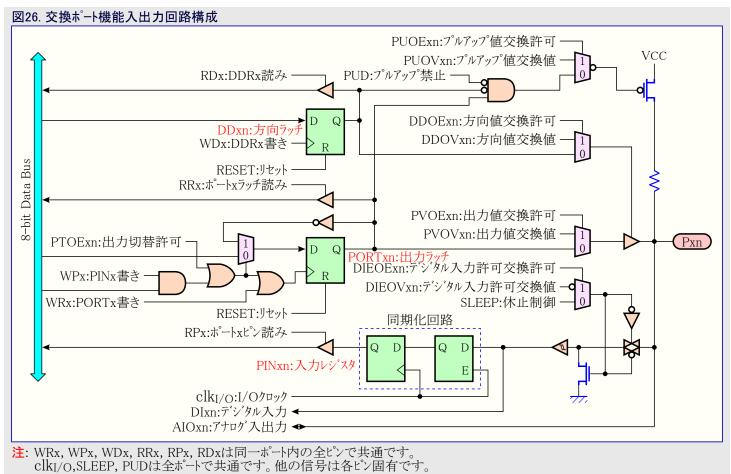


表26.は重複(交換)信号の機能一覧を示します。図26.で示すピンとポートは次表で示されません。重複(交換)信号は交換機能を持つ機能部で内部的に生成されます。

WHITE CLITTING TO SO				
表26. 交換	機能用交換信号の一般定義	義		
信号略名	信号名	意味		
PUOE	プルアップ値交換許可	1で、プルアップ許可はPUOV信号で制御され、0の場合、DDxn=0, PORTxn=1, PUD=0でプルアップが許可されます。		
PUOV	プルアップ値交換値	PUOE=1時、DDxn, PORTxn, PUDの値に関係なく、プルアップの有(1)/無(0)を指定します。		
DDOE	方向値交換許可	1で、出力駆動部はDDOV信号で制御され、0の場合、DDxnレジスタ値で制御されます。		
DDOV	方向値交換値	DDOE=1時、DDxnレジスタ値に関係なく、出力駆動部のON(1)/OFF(0)を制御します。		
PVOE	出力値交換許可	1で出力駆動部がONなら、ポート値はPVOV信号で制御されます。出力駆動部がONで0の場合、ポート値はPORTxnレシブスタ値で制御されます。		
PVOV	出力値交換値	PVOE=1時、PORTxnレジスタ値に関係なく、ポート値を制御(1/0)します。		
PTOE	出力切替許可	PTOE=1時、PORTxnレシブスタ値が反転します。		
DIEOE	デジタル入力許可交換許可	1で、デジタル入力許可はDIEOV信号で制御され、0の場合、MCUの状態(活動動作、休止 形態)によって決定されます。		
OIEOV	デジタル入力許可交換値	DIEOE=1時、MCUの状態(活動動作、休止形態)に関係なく、デジタル入力を許可(1)/禁止(0)します。		
DI	デジタル入力	交換機能用デジタル入力です。この信号は図上でシュミット トリガ出力に接続されていますが、 これは同期化前となります。この信号はクロックとしての使用を除き、各交換機能自身が同 期化します。		
AIO	アナログ入出力	交換機能用アナログ入出力です。この信号はピンに直接接続され、双方向使用ができます。		

以降の項は交換機能に関連する重複(交換)信号と各ポートの交換機能を簡単に記述します。更に先の詳細については交換機能の記述を参照してください。

# MCU制御レジスタ (MCU Control Register) MCUCR

ピット	7	6	5	4	3	2	1	0	
\$35 (\$55)	JTD	-	-	PUD	-	-	IVSEL	IVCE	MCUCR
Read/Write	R/W	R	R	R/W	R	R	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

### • ビット4 - PUD : プルアップ 禁止 (Pull-up Disable)

このビットが1を書かれると、例えDDxnとPORTxnレジスタがプルアップを許可(DDxn=0, PORTxn=1)に設定されていても、I/Oポートのプルアップは禁止されます。この特徴についてより多くの詳細に関しては38頁の「ピンの設定」をご覧ください。

# ポートAの交換機能

ポートAにはLCD制御器のCOM0~3とSEG0~3としての交換機能があります。

#### 表27. ポートAピンの交換機能

ホ°−ト ピン	交換機能
PA7	SEG3 (LCD セグメント3)
PA6	SEG2 (LCD セグメント2)
PA5	SEG1 (LCD セグメント1)
PA4	SEG0 (LCD セグメント0)
PA3	COM3 (LCD 共通電極3)
PA2	COM2 (LCD 共通電極2)
PA1	COM1 (LCD 共通電極1)
PA0	COM0 (LCD 共通電極0)

表28.と表29.はポートAの交換機能を40頁の図26.で示される交換信号に関連付けます。

### 表28. ホートA7~4の交換機能用交換信号

信号名	PA7/SEG3	PA6/SEG2	PA5/SEG1	PA4/SEG0
PUOE	LCDEN	LCDEN	LCDEN	LCDEN
PUOV	0	0	0	0
DDOE	LCDEN	LCDEN	LCDEN	LCDEN
DDOV	0	0	0	0
PVOE	0	0	0	0
PVOV	0	0	0	0
PTOE	-	_	-	-
DIEOE	LCDEN	LCDEN	LCDEN	LCDEN
DIEOV	0	0	0	0
DI	_	_	_	_
AIO	SEG3	SEG2	SEG1	SEG0

# 表29. ポートA3~0の交換機能用交換信号

信号名	PA3/COM3	PA2/COM2	PA1/COM1	PA0/COM0
PUOE	LCDEN•(LCDMUX>2)	LCDEN•(LCDMUX>1)	LCDEN • (LCDMUX>0)	LCDEN
PUOV	0	0	0	0
DDOE	LCDEN•(LCDMUX>2)	LCDEN•(LCDMUX>1)	LCDEN • (LCDMUX>0)	LCDEN
DDOV	0	0	0	0
PVOE	0	0	0	0
PVOV	0	0	0	0
PTOE	-	-	-	-
DIEOE	LCDEN•(LCDMUX>2)	LCDEN•(LCDMUX>1)	LCDEN • (LCDMUX>0)	LCDEN
DIEOV	0	0	0	0
DI	-	-	_	_
AIO	COM3	COM2	COM1	COM0





### ポートBの交換機能

ポートBピンの交換機能は表30.で示されます。

#### 表30 ポートBピンの交換機能

ホ <sup>°</sup> ート ヒ <sup>°</sup> ン	交換機能	ポート ピン	交換機能
PB7	OC2A (タイマ/カウンタ2 比較一致/PWM出力) PCINT15 (ピン変化割り込み15入力)	PB3	MISO (SPI 主側データ入力/従側データ出力) PCINT11 (ピン変化割り込み11入力)
PB6	OC1B (タイマ/カウンタ1 比較B一致/PWM出力) PCINT14 (ピン変化割り込み14入力)	PB2	MOSI (SPI 主側データ出力/従側データ入力) PCINT10(ピン変化割り込み10入力)
PB5	OC1A (タイマ/カウンタ1 比較A一致/PWM出力) PCINT13 (ピン変化割り込み13入力)	PB1	SCK (SPI 直列クロック 主側出力/従側入力) PCINT9 (ピン変化割り込み9入力)
PB4	OC0A (タイマ/カウンタ0 比較一致/PWM出力) PCINT12 (ピン変化割り込み12入力)	PB0	SS (SPI 従装置選択入力) PCINT8 (ピン変化割り込み8入力)

交換ピンの設定は次のとおりです。

• OC2A/PCINT15 - ホートB ビット7: PB7

OC2A: タイマ/カウンタ2の比較A一致出力。PB7ピンはタイマ/カウンタ2の比較A一致用外部出力として扱えます。この機能を扱うため、この ピンは出力として設定(DDB7=1)されなければなりません。このOC2Aピンはタイマ機能のPWM動作用出力ピンでもあります。

PCINT15: ピン変化割り込み15入力。PB7ピンは外部割り込み元としても扱えます。

• OC1B/PCINT14 - ホートB ビット6: PB6

OC1B: タイマ/カウンタ1の比較B一致出力。PB6ピンはタイマ/カウンタ1の比較B一致用外部出力として扱えます。この機能を扱うため、この ピンは出力として設定(DDB6=1)されなければなりません。このOC1Bピンはタイマ機能のPWM動作用出力ピンでもあります。

PCINT14: ピン変化割り込み14入力。PB6ピンは外部割り込み元としても扱えます。

• OC1A/PCINT13 - ホートB ビット5: PB5

OC1A: タイマ/カウンタ1の比較A一致出力。PB5ピンはタイマ/カウンタ1の比較A一致用外部出力として扱えます。この機能を扱うため、この ピンは出力として設定(DDB5=1)されなければなりません。このOC1Aピンはタイマ機能のPWM動作用出力ピンでもあります。

PCINT13: ピン変化割り込み13入力。PB5ピンは外部割り込み元としても扱えます。

• OC0A/PCINT12 - ホ°ートB ビット4 : PB4

OCOA: タイマ/カウンタ0の比較一致出力。PB4ピンはタイマ/カウンタ0の比較一致用外部出力として扱えます。この機能を扱うため、このピンは出力として設定(DDB4=1)されなければなりません。このOCOAピンはタイマ機能のPWM動作用出力ピンでもあります。

PCINT12: ピン変化割り込み12入力。PB4ピンは外部割り込み元としても扱えます。

• MISO/PCINT11 - π°-\B L\*y\3 : PB3

MISO: SPI用の主装置データ入力、従装置データ出力。SPIが主装置として許可されると、このピンはポートB方向レジスタ(DDRB)のDDB3 の設定に拘らず、入力として設定されます。SPIが従装置として許可されると、このピンのデータ方向はDDB3によって制御されます。このピンがSPIによって入力を強制されるとき、プルアップは未だポートB出力レジスタ(PORTB)のPORTB3によって制御できます。

PCINT11: ピン変化割り込み11入力。PB3ピンは外部割り込み元としても扱えます。

• MOSI/PCINT10 - π°-\B L\*ッ\2 : PB2

MOSI: SPI用の主装置データ出力、従装置データ入力。SPIが従装置として許可されると、このピンはポートB方向レシ、スタ(DDRB)のDDB2 設定に拘らず、入力として設定されます。SPIが主装置として許可されると、このピンのデータ方向はDDB2によって制御されます。このピンがSPIによって入力を強制されるとき、プルアップは未だポートB出力レシ、スタ(PORTB)のPORTB2によって制御できます。

PCINT10: ピン変化割り込み10入力。PB2ピンは外部割り込み元としても扱えます。

• SCK/PCINT9 – ポートB ビット1 : PB1

SCK: SPI用の主装置クロック出力、従装置クロック入力。SPIが従装置として許可されると、このピンはポートB方向レジスタ(DDRB)のDDB1 設定に拘らず、入力として設定されます。SPIが主装置として許可されると、このピンのデータ方向はDDB1によって制御されます。このピンがSPIによって入力を強制されるとき、プルアップは未だポートB出力レジスタ(PORTB)のPORTB1によって制御できます。

PCINT9: ピン変化割り込み9入力。PB1ピンは外部割り込み元としても扱えます。

# • SS/PCINT8 - π°-トΒ ビット0 : PB0

SS: SPI従装置選択入力。SPIが従装置として許可されると、このピン/はDDRBのDDB0設定に拘らず、入力として設定されます。従装置としてこのピンがLowに駆動されるとSPI(機能)が活性化(有効に)されます。SPIが主装置として許可されると、このピンのデータ方向はDDB0によって制御されます。このピンがSPIによって入力を強制される時にプルアップは未だPORTBのPORTB0によって制御できます。

PCINT8: ピン変化割り込み8入力。PB0ピンは外部割り込み元としても扱えます。

表31.と表32.はポートBの交換機能を40頁の図26.で示される交換信号に関連付けます。SPI主装置入力とSPI従装置出力がMISOを構成し、一方MOSIはSPI主装置出力とSPI従装置入力に分けられます。

表31. ポートB7~4の交換機能用交換信号

<b> </b>	DD7/OO0A/DOINT15	DDC/OO1D/DOINT14	DDE /OO1A /DOINT10	DD4/OO04/DOINT10
信号名	PB7/OC2A/PCINT15	PB6/OC1B/PCINT14	PB5/OC1A/PCINT13	PB4/OC0A/PCINT12
PUOE	0	0	0	0
PUOV	0	0	0	0
DDOE	0	0	0	0
DDOV	0	0	0	0
PVOE	OC2A許可	OC1B許可	OC1A許可	OC0A許可
PVOV	OC2A	OC1B	OC1A	OC0A
PTOE	-	_	-	_
DIEOE	PCIE1 • PCINT15	PCIE1 • PCINT14	PCIE1 • PCINT13	PCIE1 • PCINT12
DIEOV	1	1	1	1
DI	PCINT15入力	PCINT14入力	PCINT13入力	PCINT12入力
AIO	-	-	-	_

#### 表32. ポートB3~0の交換機能用交換信号

信号名	PB3/MISO/PCINT11	PB2/MOSI/PCINT10	PB1/SCK/PCINT9	PB0/SS/PCINT8
PUOE	SPE•MSTR	SPE•MSTR	SPE•MSTR	SPE•MSTR
PUOV	PORTB3•PUD	PORTB2•PUD	PORTB1•PUD	PORTB0•PUD
DDOE	SPE•MSTR	SPE•MSTR	SPE•MSTR	SPE•MSTR
DDOV	0	0	0	0
PVOE	SPE•MSTR	SPE•MSTR	SPE•MSTR	0
PVOV	SPI従装置出力	SPI主装置出力	SCK出力	0
PTOE	-	-	-	-
DIEOE	PCIE1 • PCINT11	PCIE1 • PCINT10	PCIE1 • PCINT9	PCIE1 • PCINT8
DIEOV	1	1	1	1
DI	SPI主装置/PCINT11入力	SPI従装置/PCINT10入力	SCK/PCINT9入力	SPI SS/PCINT8入力
AIO	_	_	<u>-</u>	_





# ポートCの交換機能

ポートCにはLCD制御器のSEG5~12としての交換機能があります。

# 表33. ポートCピンの交換機能

ホ°−ト ピン	交換機能
PC7	SEG5 (LCD セグメント5)
PC6	SEG6 (LCD セグメント6)
PC5	SEG7 (LCD セグメント7)
PC4	SEG8 (LCD セグメント8)
PC3	SEG9 (LCD セグメント9)
PC2	SEG10 (LCD セグメント10)
PC1	SEG11 (LCD セグメント11)
PC0	SEG12 (LCD セグメント12)

表34.と表35.はポートCの交換機能を40頁の図26.で示される交換信号に関連付けます。

### 表34 ポートC7~4の交換機能用交換信号

次54. 小 「O 」 34の文揆版形用文揆信号						
信号名	PC7/SEG5	PC6/SEG6	PC5/SEG7	PC4/SEG8		
PUOE	LCDEN	LCDEN	LCDEN	LCDEN		
PUOV	0	0	0	0		
DDOE	LCDEN	LCDEN	LCDEN	LCDEN		
DDOV	0	0	0	0		
PVOE	0	0	0	0		
PVOV	0	0	0	0		
PTOE	-	-	-	-		
DIEOE	LCDEN	LCDEN	LCDEN	LCDEN		
DIEOV	0	0	0	0		
DI	-	-	-	-		
AIO	SEG5	SEG6	SEG7	SEG8		

# 表35. ポートC3~0の交換機能用交換信号

信号名	PC3/SEG9	PC2/SEG10	PC1/SEG11	PC0/SEG12
PUOE	LCDEN	LCDEN	LCDEN	LCDEN
PUOV	0	0	0	0
DDOE	LCDEN	LCDEN	LCDEN	LCDEN
DDOV	0	0	0	0
PVOE	0	0	0	0
PVOV	0	0	0	0
PTOE	-	_	-	-
DIEOE	LCDEN	LCDEN	LCDEN	LCDEN
DIEOV	0	0	0	0
DI	-	-	ı	-
AIO	SEG9	SEG10	SEG11	SEG12

# ポートDの交換機能

ポートDピンの交換機能は表36.で示されます。

### 表36. ポートDピンの交換機能

	NOT THE TOP OF NIXIMIE				
ホ <sup>°</sup> ート ヒ <sup>°</sup> ン	交換機能	ホ <sup>°</sup> ート ピン			
PD7	SEG15 (LCD セグメント15)	PD2	SEG20 (LCD セグメント20)		
PD6	SEG16 (LCD セグメント16)	DD1	INT0 (外部割り込み0入力)		
PD5	SEG17 (LCD セグメント17)	PD1	SEG21 (LCD セグメント21)		
PD4	SEG18 (LCD セグメント18)	DDo	ICP1 (タイマ/カウンタ1 捕獲起動入力)		
PD3	SEG19 (LCD セグメント19)	PD0	SEG22 (LCD セグメント22)		

交換ピンの設定は次のとおりです。

• SEG15~20 - ポ-トD ビット7~2 : PD7~2

**SEG15~20**: LCDセグメント15~20用ピンです。

• INT0/SEG21 - ホートD ピット1: PD1

INTO:外部割り込み0入力。PD1ピンはMCUへの外部割り込み元として扱えます。

SEG21:LCDセグメント21用ピンです。

• ICP1/SEG22 - ポートD ビット0 : PD0

ICP1: タイマ/カウンタ1の捕獲起動入力。PD0ピンはタイマ/カウンタ1用捕獲起動入力ピンとして動作できます。

SEG22: LCDセグメント22用ピンです。

表37.と表38.はポートDの交換機能を40頁の図26.で示される交換信号に関連付けます。

### 表37. ポートD7~4の交換機能用交換信号

KON W TO TO CONTRIBUTION TO THE TOTAL TOTAL TO THE TOTAL THE TOTAL TO THE TOTAL TOT					
信号名	PD7/SEG15	PD6/SEG16	PD5/SEG17	PD4/SEG18	
PUOE	LCDEN • (LCDPM>1)	LCDEN•(LCDPM>1)	LCDEN • (LCDPM>2)	LCDEN•(LCDPM>2)	
PUOV	0	0	0	0	
DDOE	LCDEN•(LCDPM>1)	LCDEN•(LCDPM>1)	LCDEN • (LCDPM>2)	LCDEN•(LCDPM>2)	
DDOV	0	0	0	0	
PVOE	0	0	0	0	
PVOV	0	0	0	0	
PTOE	-	-	-	-	
DIEOE	LCDEN • (LCDPM>1)	LCDEN•(LCDPM>1)	LCDEN • (LCDPM>2)	LCDEN•(LCDPM>2)	
DIEOV	0	0	0	0	
DI	_	_	_	_	
AIO	SEG15	SEG16	SEG17	SEG18	

### 表38. ポートD3~0の交換機能用交換信号

信号名	PD3/SEG19	PD2/SEG20	PD1/INT0/SEG21	PD0/ICP1/SEG22
PUOE	LCDEN • (LCDPM>3)	LCDEN•(LCDPM>3)	LCDEN•(LCDPM>4)	LCDEN•(LCDPM>4)
PUOV	0	0	0	0
DDOE	LCDEN•(LCDPM>3)	LCDEN•(LCDPM>3)	LCDEN•(LCDPM>4)	LCDEN•(LCDPM>4)
DDOV	0	0	0	0
PVOE	0	0	0	0
PVOV	0	0	0	0
PTOE	-	-	-	_
DIEOE	LCDEN • (LCDPM>3)	LCDEN•(LCDPM>3)	LCDEN+INT0許可	LCDEN•(LCDPM>4)
DIEOV	0	0	LCDEN+INT0許可	0
DI	_	_	INT0入力	ICP1入力
AIO	SEG19	SEG20	SEG21	SEG22





### ポートEの交換機能

ポートEピンの交換機能は表39.で示されます。

#### 表39. ポートEピンの交換機能

	(00.4) [12270] [10070]					
ホ <sup>°</sup> ート ヒ <sup>°</sup> ン	交換機能	ホ <sup>°</sup> ート ヒ <sup>°</sup> ン	交換機能			
PE7	CLKO (システム クロック出力) PCINT7 (ピン変化割り込み7入力)	PE3	AIN1 (アナログ比較器反転入力) PCINT3 (ピン変化割り込み3入力)			
PE6	DO (3線動作USIデータ出力) PCINT6 (ピン変化割り込み6入力)	PE2	AINO (アナログ・比較器非反転入力) XCK (USART 外部クロック入出力)			
	DI (3線動作USIデータ入力)		PCINT2(ピン変化割り込み2入力)			
PE5	SDA (2線動作USI(TWI)データ入出力) PCINT5 (ピン変化割り込み5入力)	PE1	TXD (USART 送信データ出力) PCINT1 (ピン変化割り込み1入力)			
	USCK (3線動作USIクロック入出力)		, ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,			
PE4	SCL (2線動作USI(TWI)クロック入出力) PCINT4 (ピン変化割り込み4入力)	PE0	RXD (USART 受信データ入力) PCINTO (ピン変化割り込み0入力)			

交換ピンの設定は次のとおりです。

CLKO/PCINT7 – π<sup>°</sup>-トE L<sup>\*</sup>ット7 : PE7

CLKO: システム クロック出力。分周したシステム クロックがPE7ピンに出力できます。分周したシステム クロックはCKOUTヒューズがプログラム(0)されると、PORTE7とDDE7設定に拘らず、出力されます。これはリセット中にも出力されます。

PCINT7: ピン変化割り込み7入力。PE7ピンは外部割り込み元としても扱えます。

• DO/PCINT6 - ポートE ビット6 : PE6

DO: 3線動作USIのデータ出力。

PCINT6: ピン変化割り込み6入力。PE6ピンは外部割り込み元としても扱えます。

• DI/SDA/PCINT5 - ホートE ビット5: PE5

DI: 3線動作USIのデータ入力。

SDA: 2線動作USI(TWI)のデータ入出力。

PCINT5: ピン変化割り込み5入力。PE5ピンは外部割り込み元としても扱えます。

• USCK/SCL/PCINT4 - ホートE ビット4: PE4

USCK: 3線動作USIのクロック入出力。

SCL: 2線動作USI(TWI)のクロック入出力。

PCINT4: ピン変化割り込み4入力。PE4ピンは外部割り込み元としても扱えます。

• AIN1/PCINT3 - π°-\Ε Ε΄ η\β : PE3

AIN1: アナログ比較器の反転入力。このピンはアナログ比較器の反転入力に直接接続されます。

PCINT3: ピン変化割り込み3入力。PE3ピンは外部割り込み元としても扱えます。

• AIN0/XCK/PCINT2 - π°-hE L yh2: PE2

AINO: アナログ比較器の非反転入力。このピンはアナログ比較器の非反転入力に直接接続されます。

**XCK**: USARTの外部クロック入出力。ポートE方向レジスタ(DDRE)のDDE2は、このクロックが入力(DDE2=0)または出力(DDE2=1)のどちらかを制御します。 XCKピンはUSARTが同期種別で動作する時だけ活動します(有効です)。

PCINT2: ピン変化割り込み2入力。PE2ピンは外部割り込み元としても扱えます。

• TXD/PCINT1 - π°-\Ε Ε΄ν\-1 : PE1

TXD:送信データ(USART用データ出力ピン)。USART送信部が許可されると、ピンはDDREのDDE1値に拘らず出力として設定されます。

PCINT1: ピン変化割り込み1入力。PE1ピンは外部割り込み元としても扱えます。

• RXD/PCINT0 - ホートE ビット0: PE0

RXD: 受信データ(USART用データ入力ピン)。 USART受信部が許可されると、このピンはDDREのDDE0の値に拘らず入力として設定されます。 USARTがこのピンを入力に強制するとき、プルアップは未だPORTE0ピットによって制御できます。

PCINTO: ピン変化割り込み0入力。PEOピンは外部割り込み元としても扱えます。

表40.と表41.はポートEの交換機能を40頁の図26.で示される交換信号に関連付けます。

# 表40. ポートE7~4の交換機能用交換信号

信号名	PE7/PCINT7/(CLKO)	PE6/DO/PCINT6	PE5/DI/SDA/PCINT5	PE4/USCK/SCL/PCINT4
PUOE	0	0	2線USI	2線USI
PUOV	0	0	0	0
DDOE	CKOUT	0	2線USI	2線USI
DDOV	1	0	(SDA+PORTE5) • DDE5	(SCL保持+PORTE4)·DDE4
PVOE	CKOUT	3線USI	2線USI·DDE5	2線USI•DDE4
PVOV	clk <sub>I/O</sub>	DO	0	0
PTOE	-	-	0	USITC
DIEOE	PCIE0 • PCINT7	PCIE0 • PCINT6	(PCIE0 • PCINT5)+USISIE	(PCIE0 • PCINT4)+USISIE
DIEOV	1	1	1	1
DI	PCINT7入力	PCINT6入力	DI/SDA/PCINT5入力	USCK/SCL/PCINT4入力
AIO	-	-	_	_

注: CKOUTヒューズがプログラム(0)ならばCKOUTが論理1です。

# 表41. ポートE3~0の交換機能用交換信号

信号名	PE3/AIN1/PCINT3	PE2/AIN0/XCK/PCINT2	PE1/TXD/PCINT1	PE0/RXD/PCINT0
PUOE	0	0	TXEN	RXEN
PUOV	0	0	0	PORTE0.PUD
DDOE	0	0	TXEN	RXEN
DDOV	0	0	1	0
PVOE	0	XCK出力許可	TXEN	0
PVOV	0	XCK	TXD	0
PTOE	-	-	-	_
DIEOE	(PCIE0 • PCINT3)+AIN1D	(PCIE0 • PCINT2)+AIN0D	PCIE0 • PCINT1	PCIE0 • PCINT0
DIEOV	PCIE0 • PCINT3	PCIE0 • PCINT2	1	1
DI	PCINT3入力	XCK/PCINT2入力	PCINT1入力	RXD/PCINT0入力
AIO	AIN1入力	AIN0入力	-	_

注: AINODとAIN1Dは131頁の「デッタル入力禁止レッスタ1(DIDR1)」で記述されます。





#### ポートFの交換機能

ポートFピンには**表42**.に示されるようにA/D変換用アナログ入力としての交換機能があります。ポートFピンのいくつかが出力として設定される場合、A/D変換実行中の時にそれらを切り替えないことが重要です。これは変換結果を不正にするかもしれません。JTAGインターフェースが許可されると、リセットが起きてもPF7(TDI)、PF5(TMS)、PF4(TCK)ピンのプルアップ抵抗は活性(有効)にされます。

### 表42. ポートFピンの交換機能

ホ°ート ヒ°ン	交換機能
PF7	ADC7 (A/D変換アナログ入力チャネル7) TDI (JTAG 検査データ入力)
PF6	ADC6 (A/D変換アナログ入力チャネル6) TDO (JTAG 検査データ出力)
PF5	ADC5 (A/D変換アナログ入力チャネル5) TMS (JTAG 検査種別選択入力)
PF4	ADC4 (A/D変換アナログ入力チャネル4) TCK (JTAG クロック入力)
PF3	ADC3 (A/D変換アナログ、入力チャネル3)
PF2	ADC2 (A/D変換アナログ入力チャネル2)
PF1	ADC1 (A/D変換アナログ入力チャネル1)
PF0	ADC0 (A/D変換アナログ入力チャネル0)

交換ピンの設定は次のとおりです。

• ADC7/TDI - ポートF ビット7 : PF7

**ADC7**: A/D変換チャネル7入力。

TDI: JTAG検査データ入力。命令レジスタまたはデータレジスタ(走査チェーン)に移動入力される直列入力データです。JTAGインターフェースが許可されると、このピンはI/Oピンとして使えません。

• ADC6/TDO - ホートF ビット6: PF6

ADC6: A/D変換チャネル6入力。

TDO: JTAG検査データ出力。命令レジスタまたはデータレジスタから移動出力される直列出力データです。JTAGインターフェースが許可されたなら、このピンはI/Oピンとして使えません。データを移動出力するTAP状態に於いてTDOピンは動的に駆動します。他の状態ではこのピンがHighに引かれます。

• ADC5/TMS – ポートF ビット5 : PF5

**ADC5**: A/D変換チャネル5入力。

TMS: JTAG検査種別選択。このピンはTAP(検査入出力ポート)制御器状態機構を通しての操作に使われます。JTAGインターフェースが許可されると、このピンはI/Oピンとして使えません。

• ADC4/TCK – ポートF ビット4 : PF4

ADC4: A/D変換チャネル4入力。

TCK: JTAGクロック。JTAG操作はTCKに同期します。JTAGインターフュースが許可されると、このピンはI/Oピンとして使えません。

• ADC3 – ポートF ビット3 : PF3

ADC3: A/D変換チャネル3入力。

• ADC2 - ホートF ビット2: PF2

**ADC2**: A/D変換チャネル2入力。

• ADC1 – ポートF ビット1 : PF1

**ADC1**: A/D変換チャネル1入力。

• ADC0 - ポートF ビット0 : PF0

**ADC0**: A/D変換チャネル0入力。

表43.と表44.はポートFの交換機能を40頁の図26.で示される交換信号に関連付けます。

# 表43. ポートF7~4の交換機能用交換信号

信号名	PF7/ADC7/TDI	PF6/ADC/TDO	PF5/ADC5/TMS	PF4/ADC4/TCK
PUOE	JTAGEN	JTAGEN	JTAGEN	JTAGEN
PUOV	1	1	1	1
DDOE	JTAGEN	JTAGEN	JTAGEN	JTAGEN
DDOV	0	IR移動+DR移動	IR移動+DR移動 0	
PVOE	0	JTAGEN	0	0
PVOV	0	TDO	0	0
PTOE	-	-	-	-
DIEOE	JTAGEN	JTAGEN	JTAGEN	JTAGEN
DIEOV	0	0	0	0
DI	_	_	_	_
AIO	ADC7入力/TDI入力	ADC6入力	ADC5入力/TMS入力	ADC4入力/TCK入力

# 表44. ポートF3~0の交換機能用交換信号

信号名	PF3/ADC3	PF2/ADC2	PF1/ADC1	PF0/ADC0
PUOE	0	0	0	0
PUOV	0	0	0	0
DDOE	0	0	0	0
DDOV	0	0	0	0
PVOE	0	0	0	0
PVOV	0	0	0	0
PTOE	-	_	_	_
DIEOE	0	0	0	0
DIEOV	0	0	0	0
DI	-	_	_	_
AIO	ADC3入力	ADC2入力	ADC1入力	ADC0入力





### ポートGの交換機能

ポートGピンの交換機能は表45.で示されます。

### 表45. ポートGピンの交換機能

ホ°−ト ピン	交換機能
PG4	T0 (タイマ/カウンタ0 外部クロック入力) SEG23 (LCD セグメント23)
PG3	T1 (タイマ/カウンタ1 外部クロック入力) SEG24 (LCD セグメント24)
PG2	SEG4 (LCD セグメント4)
PG1	SEG13 (LCD セグメント13)
PG0	SEG14 (LCD セグメント14)

交換ピンの設定は次のとおりです。

• T0/SEG23 - ポートG ビット4 : PG4

T0: タイマ/カウンタ0の外部クロック入力ピンです。

SEG23: LCDセグメント23用ピンです。

• T1/SEG24 - π°-\G L y\3 : PG3

T1: タイマ/カウンタ1の外部クロック入力ピンです。

SEG24: LCDセグメント24用ピンです。

• SEG4 – π°-トG ビット2 : PG2

SEG4: LCDセグメント4用ピンです。

• SEG13 - ポートG ビット1 : PG1

SEG13: LCDセグメント13用ピンです。

• SEG14 – ポートG ビット0 : PG0

SEG14: LCDセグメント14用ピンです。

表46.はポートGの交換機能を40頁の図26.で示される交換信号に関連付けます。

#### 表46. ポートG4~0の交換機能用交換信号

信号名	PG4/T0/SEG23	PG3/T1/SEG24	PG2/SEG4	PG1/SEG13	PG0/SEG14
PUOE	LCDEN• (LCDPM>5)	LCDEN• (LCDPM>6)	LCDEN	LCDEN• (LCDPM>0)	LCDEN• (LCDPM>0)
PUOV	0	0	0	0	0
DDOE	LCDEN• (LCDPM>5)	LCDEN• (LCDPM>6)	LCDEN	LCDEN• (LCDPM>0)	LCDEN• (LCDPM>0)
DDOV	0	0	0	0	0
PVOE	0	0	0	0	0
PVOV	0	0	0	0	0
PTOE	_	-	-	-	_
DIEOE	LCDEN• (LCDPM>5)	LCDEN• (LCDPM>6)	LCDEN	LCDEN• (LCDPM>0)	LCDEN• (LCDPM>0)
DIEOV	0	0	0	0	0
DI	T0入力	T1入力	_	-	_
AIO	SEG23	SEG24	SEG4	SEG13	SEG14

(訳注) 原書の表46.と表47.は表46.として統合しました。

# I/Oポート用レジスタ

# ホ<sup>°</sup>ートA出力レジスタ (Port A Data Register) **PORTA**

ピット	7	6	5	4	3	2	1	0	
\$02 (\$22)	PORTA7	PORTA6	PORTA5	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0	PORTA
Read/Write	R/W								
初期値	0	0	0	0	0	0	0	0	

# ポートA方向レジスタ (Port A Data Direction Register) DDRA

ピット	7	6	5	4	3	2	1	0	
\$01 (\$21)	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	DDRA
Read/Write	R/W								
初期値	0	0	0	0	0	0	0	0	

# ホ°ートA入力レジスタ (Port A Input Address) PINA

ピット	7	6	5	4	3	2	1	0	_
\$00 (\$20)	PINA7	PINA6	PINA5	PINA4	PINA3	PINA2	PINA1	PINA0	PINA
Read/Write	R/W								
初期値	不定								

# ポートB出力レジスタ (Port B Data Register) PORTB

ピット	7	6	5	4	3	2	1	0	
\$05 (\$25)	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	PORTB
Read/Write	R/W								
初期値	0	0	0	0	0	0	0	0	

# ホ°ートB方向レジスタ (Port B Data Direction Register) DDRB

ピット	7	6	5	4	3	2	1	0	
\$04 (\$24)	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	DDRB
Read/Write	R/W								
初期値	0	0	0	0	0	0	0	0	

# ホ°ートB入力レジスタ (Port B Input Address) PINB

ピット	7	6	5	4	3	2	1	0	
\$03 (\$23)	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	PINB
Read/Write	R/W								
初期値	不定								

# ホ°ートC出力レジスタ (Port C Data Register) PORTC

ピット	7	6	5	4	3	2	1	0	
\$08 (\$28)	PORTC7	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	PORTC
Read/Write	R/W								
初期値	0	0	0	0	0	0	0	0	

# ポートC方向レジスタ (Port C Data Direction Register) DDRC

ピット	7	6	5	4	3	2	1	0	
\$07 (\$27)	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	DDRC
Read/Write	R/W								
初期値	0	0	0	0	0	0	0	0	





# ホ°ートC入力レジスタ (Port C Input Address) PINC

ピット	7	6	5	4	3	2	1	0	
\$06 (\$26)	PINC7	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0	PINC
Read/Write	R/W								
初期値	不定								

# ホ°ートD出力レジスタ (Port D Data Register) PORTD

ピット	7	6	5	4	3	2	1	0	
\$0B (\$2B)	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	PORTD
Read/Write	R/W								
初期値	0	0	0	0	0	0	0	0	

# ポートD方向レシスタ (Port D Data Direction Register) DDRD

ピット	7	6	5	4	3	2	1	0	
\$0A (\$2A)	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	DDRD
Read/Write	R/W								
初期値	0	0	0	0	0	0	0	0	

# ホ°ートD入力レジスタ (Port D Input Address) PIND

ピット	7	6	5	4	3	2	1	0	_
\$09 (\$29)	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	PIND
Read/Write	R/W								
初期値	不定								

# ホ<sup>°</sup>ートE出力レジスタ (Port E Data Register) **PORTE**

ピット	7	6	5	4	3	2	1	0	
\$0E (\$2E)	PORTE7	PORTE6	PORTE5	PORTE4	PORTE3	PORTE2	PORTE1	PORTE0	PORTE
Read/Write	R/W								
初期値	0	0	0	0	0	0	0	0	

# ホ°ートE方向レジスタ (Port E Data Direction Register) DDRE

PAD (PAD) DDEC DDEC DDEA					_
\$0D (\$2D) DDE7 DDE6 DDE5 DDE4	DDE3	DDE2	DDE1	DDE0	DDRE
Read/Write R/W R/W R/W R/W	R/W	R/W	R/W	R/W	
初期値 0 0 0	0	0	0	0	

# ホ°ートE入力レジスタ (Port E Input Address) PINE

ピット	7	6	5	4	3	2	1	0	
\$0C (\$2C)	PINE7	PINE6	PINE5	PINE4	PINE3	PINE2	PINE1	PINE0	PINE
Read/Write	R/W								
初期値	不定								

# ホ<sup>°</sup>ートF出力レジスタ (Port F Data Register) **PORTF**

ピット	7	6	5	4	3	2	1	0	
\$11 (\$31)	PORTF7	PORTF6	PORTF5	PORTF4	PORTF3	PORTF2	PORTF1	PORTF0	PORTF
Read/Write	R/W								
初期値	0	0	0	0	0	0	0	0	

# ホ<sup>°</sup>ートF方向レジスタ (Port F Data Direction Register) **DDRF**

ピット	7	6	5	4	3	2	1	0	_
\$10 (\$30)	DDF7	DDF6	DDF5	DDF4	DDF3	DDF2	DDF1	DDF0	DDRF
Read/Write	R/W								
初期値	0	0	0	0	0	0	0	0	

# ホ<sup>°</sup>ートF入力レジスタ (Port F Input Address) **PINF**

ピット	7	6	5	4	3	2	1	0	
\$0F (\$2F)	PINF7	PINF6	PINF5	PINF4	PINF3	PINF2	PINF1	PINF0	PINF
Read/Write	R/W								
初期値	不定								

# ポートG出力レジスタ (Port G Data Register) PORTG

ピット	7	6	5	4	3	2	1	0	
\$14 (\$34)	-	-	-	PORTG4	PORTG3	PORTG2	PORTG1	PORTG0	PORTG
Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

# ホ°ートG方向レジスタ (Port G Data Direction Register) DDRG

ピット	7	6	5	4	3	2	1	0	
\$13 (\$33)	-	-	-	DDG4	DDG3	DDG2	DDG1	DDG0	DDRG
Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

# ホ°ートG入力レジスタ (Port G Input Address) PING

t'yh 7 6 5 4 3 2	0
\$12 (\$32)	G1 PING0 PING
Read/Write R R R R/W R/W R/W R	V R/W
初期値 0 0 0 不定 不定 不定 不	产 不定





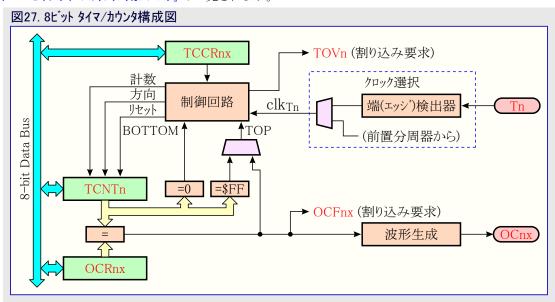
# 8ビット タイマ/カウンタ0 (PWM付き)

タイマ/カウンタのは単一比較部付きの汎用8ビットタイマ/カウンタ部です。主な特徴は次のとおりです。

- 単一比較部付き計数器
- 比較一致でのタイマ/カウンタ解除(自動再設定)
- 不具合なしで正しい位相のパルス幅変調器 (PWM)
- 周波数発生器
- 外部事象カウンタ
- クロック用10ビット前置分周器
- 溢れと比較一致割り込み (TOV0とOCF0A)

#### 概要

この8ビット タイマ/カウンタの簡単化した構成図は**図27**.で示されます。I/Oピンの実際の配置については2頁の「**ピン配置**」を参照してください。CPUがアクセス可能な(I/OビットとI/Oピンを含む)I/Oレジスタは<mark>赤文字(訳注</mark>:原文は太字)で示されます。デバイス仕様のI/Oレジスタとビット位置は62頁の「8ビット タイマ/カウンタ0用レジスタ」で一覧されます。



# 関係レジスタ

タイマ/カウンタ(TCNT0)と比較レジ、スタ(OCR0A)は8ビットのレジ、スタです。割り込み要求信号はタイマ/カウンタ割り込み要求レジ、スタ(TIFR0)で全て見えます。全ての割り込みはタイマ/カウンタ割り込み許可レジ、スタ(TIMSK0)で個別に遮蔽(禁止)されます。TIFR0とTIMSK0はこの図で示されません。

このタイマ/カウンタは内部的、前置分周器経由、またはT0ピンの外部クロック元によってクロック駆動されます。クロック選択論理部はタイマ/カウンタが値を増加(または減少)するのに使うクロック元を制御します。クロック元が選ばれないと、タイマ/カウンタは動きません。クロック選択論理部からの出力はタイマ/カウンタ クロック(clkT0)として参照されます。

2重緩衝化した比較レジスタ(OCR0A)はタイマ/カウンタ値と常に比較されます。この比較結果は比較出力(OC0A)ピンでPWMまたは可変周波数出力を作成するための波形生成器によって使えます。56頁の「比較出力部」をご覧ください。この比較一致発生は比較一致割り込み要求の発生に使える比較一致割り込み要求フラグ(OCF0A)も設定(1)します。

#### 定義

本章でのレシ、スタとヒット参照の多くは一般形で書かれます。小文字の'n'は タイマ/カウンタ番号、この場合は0で置き換えます。小文字のxは比較出力部のチャネル名を表し、この場合はAです。けれどもプログラムでレシ、スタまたはビット定義に使う時は正確な形式が使われなければなりません(例えばタイマ/カウンタ0のカウンタ値のアクセスに対してのTCNT0のように)。

表48.の定義は本文書を通して広範囲に渡って使われます。

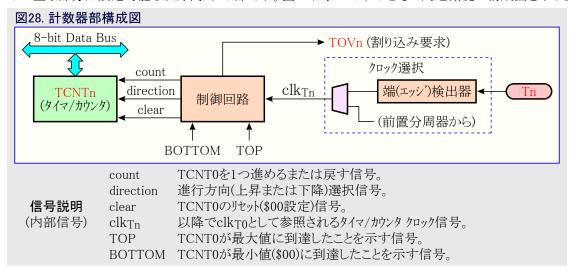
表48. 用語定義								
用語	意味							
BOTTOM	タイマ/カウンタが\$00に到達した時。							
MAX	タイマ/カウンタが\$FF(255)に到達した時。							
TOP	タイマ/カウンタが指定された固定値(\$FF) またはOCROA値に到達した時。この指 定(TOP)値は動作種別に依存します。							

### タイマ/カウンタの クロック

このタイマ/カウンタは内部または外部のクロック元によってクロック駆動できます。このクロック元はタイマ/カウンタ0制御レシ、スタA(TCCR0A)に配置されたクロック選択(CS02~0)ビットによって制御されるクロック選択論理回路で選ばれます。クロック元と前置分周器の詳細については65頁の「タイマ/カウンタ0とタイマ/カウンタ1の前置分周器」をご覧ください。

### 計数器部

8ビット タイマ/カウンタの主な部分は設定可能な双方向カウンタ部です。図28.は、このカウンタとその周辺環境の構成図を示します。



使った動作種別に依存して、カウンタは各タイマ/カウンタ クロック(clkT0)で解除(\$00)、増加(+1)、または減少(-1)されます。clkT0はクロック選択 (CS02~0)ビットによって選ばれた内部または外部のクロック元から生成できます。クロック元が選ばれない(CS02~0=000)と、タイマ/カウンタは停止されます。けれどもTCNT0値はタイマ/カウンタ クロック(clkT0)が存在するしないに拘らず、CPUによってアクセスできます。CPU書き込みは全てのカウンタ解除や計数動作を無視します(上位優先順位を持ちます)。

計数順序(方法)はタイマ/カウンタ制御レジスタA(TCCR0A)に配置された波形生成種別(WGM01,0)ビットの設定によって決定されます。これらはカウンタ動作(計数)方法とOC0A比較出力に生成される方法間の接続に近いものです。進化した計数順序と波形生成についてより多くの詳細に関しては58頁の「動作種別」をご覧ください。

タイマ/カウンタ溢れ(TOV0)フラグはWGM01,0ビットによって選ばれた動作種別に従って設定(1)されます。TOV0はCPU割り込み発生に使えます。





### 比較出力部

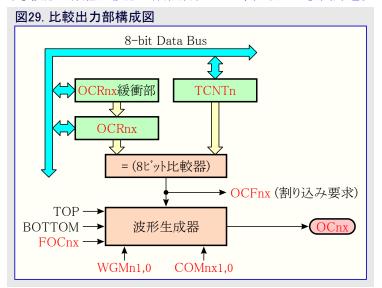
この8ビット比較器はTCNT0と比較レジスタ(OCR0A)を継続的に比較します。TCNT0とOCR0Aが等しければ、比較器は一致を指示します。この一致は次のタイマ/カウンタ クロック周期で比較割り込み要求フラグ(OCF0A)を設定(1)します。許可(I=1, OCIE0A=1)なら、この比較割り込み要求フラグは比較割り込みを発生します。OCF0Aは割り込みが実行されると自動的に解除(0)されます。代わりにOCF0Aはこのビット位置に論理1を書くことによってソフトウェアでも解除(0)できます。波形生成器は波形生成種別(WGM01,0)ビットと比較出力選択

(COM0A1,0)ビットによって設定された動作種別に従った出力を 生成するのにこの一致信号を使います。MAXとBOTTOM信号 は動作種別(58頁の「動作種別」参照)のいくつかで両端値の特 別な場合を扱うため、波形生成器によって使われます。

#### 図29.は比較出力部の構成図を示します。

OCR0Aはパルス幅変調(PWM)のどれかを使う時に2重緩衝化されます。標準動作と比較一致タイマ/カウンタ解除(CTC)動作については2重緩衝動作が禁止されます。2重緩衝動作は計数の流れのTOPまたはBOTTOMのどちらかに対してOCR0Aレジスタの更新を同期化します。この同期化は奇数長や非対称PWMパルスの発生を防ぎ、それによって不具合なしの出力を作成します。

OCROAのアクセスは複雑なように思えますが決してそんなことはありません。2重緩衝動作が許可されるとCPUはOCROA緩衝部をアクセスし、禁止されるとOCROAレジスタを直接アクセスします。



#### 強制比較出力

非PWM波形生成動作での比較器の一致出力は、強制変更(FOC0A)ビットに1を書くことによって強制(変更)できます。比較一致の強制は比較割り込み要求フラグ(OCF0A)の設定(1)やタイマ/カウンタの再設定/解除を行いませんが、OC0Aピンは実際の比較一致が起きた場合と同様に更新されます(COM0A1,0ビット設定がOC0Aピンの設定(1)、解除(0)、1/0交互のどれかを定義)。

### TCNT0書き込みによる比較一致妨害

TCNT0への全てのCPU書き込みは、例えタイマ/カウンタが停止されていても、次のタイマ/カウンタ クロック周期で起こるどんな比較一致をも妨げます。この特質はタイマ/カウンタ クロックが許可されている時に割り込みを起動することなく、TCNT0と同じ値に初期化されることをOCR0Aに許します。

### 比較一致部の使用

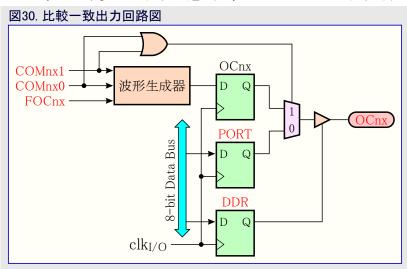
どの動作種別でのTCNT0書き込みでも1タイマ/カウンタ クロック周期間、全ての比較一致を妨げるため、タイマ/カウンタが走行中であるかないかに拘らず、比較出力部を使う場合、TCNT0を変更する時に危険を伴います。TCNT0に書かれた値がOCR0A値と同じ場合に比較一致は失われ(一致が発生せず)、不正な波形生成に終わります。同様にタイマ/カウンタが下降計数のとき、BOTTOMに等しいTCNT 0値を書いてはいけません。

OC0Aの初期設定はポート ピンに対するポート方向レジスタを出力に設定する前に行われるべきです。OC0A値を設定する一番簡単な方法は標準動作で強制変更(FOC0A)ストローブ ビットを使うことです。波形生成動作種別間を変更する時であっても、OC0A(内部)レジスタはその値を保ちます。

比較出力選択(COM0A1,0)と、ットが比較値(OCR0A)と共に2重緩衝されないことに気付いてください。COM0A1,0と、ットの変更は直ちに有効となります。

### 比較一致出力部

比較出力選択(COM0A1,0)ビットは2つの機能を持ちます。波形生成器は次の比較一致での比較出力(OC0A)状態の定義にCOM0A1,0ビットを使います。またCOM0A1,0ビットはOC0Aピン出力元を制御します。図30.はCOM0A1,0ビット設定によって影響を及ぼされる論理回路の簡単化した図を示します。図のI/Oレジスタ、I/Oビット、I/Oピンは赤文字(訳注:原文は太字)で示されます。COM0A1,0ビットによって影響を及ぼされる標準I/Oポート制御レジスタ(PORTとDDR)の部分だけが示されます。OC0Aの状態を参照するとき、その参照はOC0Aピンでなく内部OC0Aレジスタに対してです。システム リセットが起こると、OC0Aレジスタは'0'にリセットされます。



COM0A1,0ビットのどちらかが設定(1)されると、標準I/Oポート機能は波形生成器からの比較出力(OC0A)によって無効にされます。けれどもOC0Aピンの方向(入出力)はポート ピンに対するポート方向レジスタ(DDR)によって未だ制御されます。OC0Aピンに対するポート方向レジスタのビット(DDR\_OC0A)はOC0A値がピンで見えるのに先立って出力として設定されなければなりません。このポートの交換機能は波形生成種別と無関係です。

比較出力ピン論理回路の設計は出力が許可される前のOC0A状態の初期化を許します。いくつかのCOM0A1,0ビット設定が或る種の動作種別に対して予約されることに注意してください。62頁の「8ビット タイマ/カウンタ0用レジスタ」をご覧ください。

### 比較一致出力選択と波形生成

波形生成器は標準、CTC、PWM動作でCOM0A1,0ピットを違うふうに使います。全ての動作種別に対してCOM0A1,0=00設定は次の比較一致で実行すべきOC0Aレジスタの動きがないことを波形生成器へ告げます。非PWM動作での比較出力動作については62頁の表50.を参照してください。高速PWM動作については62頁の表51、位相基準PWMについては62頁の表52.を参照してください。

COM0A1,0ビットの状態変更はこのビットが書かれた後の最初の比較一致で有効になります。非PWM動作について、この動作は強制変更(FOC0A)ストロープビットを使うことによって直ちに効果を得ることを強制できます。





### 動作種別

動作種別、換言するとタイマ/カウンタと比較出力ピンの動作は波形生成種別(WGM01,0)ビットと比較出力選択(COM0A1,0)ビットの組み合わせによって定義されます。比較出力選択ビットは計数順序(動作)に影響を及ぼしませんが、一方波形生成種別ビットは影響を及ぼします。COM0A1,0ビットは生成されるPWM出力が反転されるべきか、されないべきか(反転または非反転PWM)どちらかを制御します。非PWM動作に対するCOM0A1,0ビットは比較一致で出力が解除(0)、設定(1)、1/0交互のどれにされるべきかを制御します(57頁の「比較一致出力部」をご覧ください)。

タイミング情報の詳細については61頁の「**タイマ/カウンタ0のタイミンク**゙」を参照してください(<mark>訳注</mark>:原文中の図番号省略)。

#### 標準動作

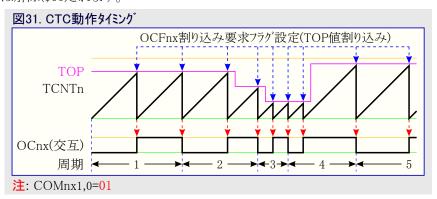
最も単純な動作種別が標準動作(WGM01,0=00)です。本動作種別での計数方向は常に上昇(+)で、カウンタ解除は実行されません。カウンタは8ビット最大値(TOP=\$FF)を通過すると単に範囲を超え、そして\$00(BOTTOM)から再び始めます。通常動作でのタイマ/カウンタ溢れ(TOV0)フラグはTCNT0が\$00になる時と同じタイマ/カウンタ クロック周期で設定(1)されます。この場合のTOV0フラグは設定(1)のみで解除(0)されないことを除いて第9ビットのようになります。けれどもTOV0フラグを自動的に解除(0)するタイマ/カウンタの溢れ割り込みと組み合わせたタイマ/カウンタの分解能はソフトウェアによって増やせます。標準動作での考慮に特別な場合はなく、新しいカウンタ値は何時でも書けます。

比較出力部は与えられた或る時間に割り込みを生成するのに使えます。標準動作で波形を生成するのに比較出力を使うのは、それが大変多くのCPU時間を占有するため推奨されません。

### 比較一致タイマ/カウンタ解除(CTC)動作

比較一致タイマ/カウンタ解除(CTC)動作(WGM01,0=10)ではOCR0Aがカウンタの分解能を操作するのに使われます。CTC動作ではカウンタ (TCNT0)値がOCR0Aと一致すると、カウンタは\$00に解除されます。OCR0Aはカウンタに対するTOP値、従って分解能も定義します。この動作種別はより大きい比較一致出力周波数の制御を許します。それは外部の出来事の計数操作も簡単にします。

CTC動作についてのタイミング図は図31.で示されます。カウンタ(TCNT0)値はTCNT0とOCR0A間で比較一致が起こるまで増加し、そしてその後にカウンタ(TCNT0)は解除(\$00)されます。



OCF0Aフラグを使うことにより、タイマ/カウンタ値がTOP値に達する時毎に割り込みが生成できます。割り込みが許可されるなら、割り込み処理ルーチンはTOP値を更新するのに使えます。けれども前置分周なしまたは低い前置分周値でカウンタが走行している時にBOTTOMと近い値にTOPを変更することは、CTC動作が2重緩衝機能を持たないために注意して行わなければなりません。OCR0Aに書かれた新しい値がTCNT0の現在値よりも低い(小さい)場合、タイマ/カウンタは(その回の)比較一致を失います。その後のカウンタは比較一致が起こるのに先立って最大値(\$FF)へ、そして次に\$00から始める計数をしなければならないでしょう。

CTC動作で波形出力を生成するため、OC0A出力は比較出力選択(COM0A1,0)ビットを交互動作(=01)に設定することによって各比較一致での論理レベル交互切り替えに設定できます。OC0A値はそのピンに対するデータ方向が出力(DDR\_OC0A=1)に設定されない限り ポート ピンで見えないでしょう。生成された波形はOCR0Aが0(\$00)に設定される時に $f_{OC0A}=f_{clk\_I/O}/2$ の最大周波数を得ます。生成波形周波数は次式によって定義されます。

$$f_{\text{OCnx}} = \frac{f_{\text{clk\_I/O}}}{2 \times N \times (1 + \text{OCRnx})}$$

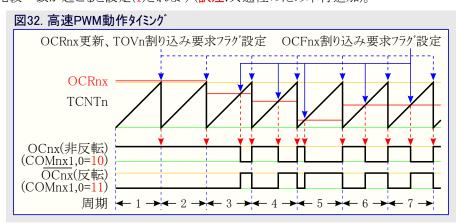
変数Nは前置分周数(1,8,64,256,1024)を表します。

標準動作と同じように、タイマ/カウンタ溢れ(TOV0)フラグはカウンタがMAXから\$00へ計数するのと同じタイマ/カウンタ クロック周期で設定(1)されます。

#### 高速PWM動作

高速 $^{\circ}$ ルス幅変調(PWM)動作(WGM01,0=11)は高周波数PWM波形生成選択を提供します。高速PWMはそれが単一傾斜(鋸波)動作であることによって他のPWM動作と異なります。カウンタはBOTTOMからMAXまで計数し、その後BOTTOMから再び始めます。非反転比較出力動作(COM0A1,0=10)での比較出力(OC0A)は、TCNT0とOCR0A間の比較一致で解除(0)され、BOTTOMで設定(1)されます。反転出力動作(COM0A1,0=11)の出力は比較一致で設定(1)され、BOTTOMで解除(0)されます。この単一傾斜動作のため、高速PWM動作の動作周波数は両傾斜(三角波)動作を使う位相基準PWM動作よりも2倍高くできます。この高い周波数は電力調節、整流、D/A変換に対して高速PWM動作を都合よく適合させます。高い周波数は物理的に小さな外部部品(コイルやコンデンサ)を許し、従ってシステム総費用を削減します。

高速PWM動作でのタイマ/カウンタはタイマ/カウンタ値がMAX値と一致するまで増加されます。そしてタイマ/カウンタは(一致の)次のタイマ/カウンタ クロック周期で解除(\$00)されます。高速PWM動作のタイング図は図32.で示されます。TCNT0値はタイシング図で単一傾斜動作(鋸波)を表す折れ線グラフとして示されます。この図は非反転と反転のPWM出力を含みます。細い赤線はOCR0A値を示し、TCNT0値との交点(接点)がTCNT0とOCR0A間の比較一致を示します(訳注:図補正に伴い本行若干変更)。比較割り込み要求フラグ(OCF0A)はOCR0A=MAXを除いて比較一致が起こると設定(1)されます(訳注:共通性のため本行追加)。



タイマ/カウンタ溢れ(TOV0)フラグはカウンタがMAXに到達する時毎に設定(1)されます。割り込みが許可されるなら、その割り込み処理ルーチンは比較値を更新するのに使えます。

高速PWM動作での比較部はOC0AピンでのPWM波形の生成を許します。COM0A1,0ビットを'10'に設定することは非反転PWM出力を作成し、反転PWM出力はCOM0A1,0を'11'に設定することで生成できます(62頁の表51.をご覧ください)。実際のOC0A値はポート ピンに対するデータ方向(DDR\_OC0A)が出力として設定される場合だけ見えるでしょう。PWM波形はTCNT0とOCR0A間の比較一致でOC0A(内部)レジスタを設定(1)(または解除(0))と、カウンタが解除(\$00、MAXからBOTTOMへ変更)されるタイマ/カウンタ クロック周期でOC0Aレジ、スタを解除(0)または設定(1)することによって生成されます。

PWM出力周波数は次式によって計算できます。

$$f_{\text{OCnxPWM}} = \frac{f_{\text{clk\_I/O}}}{N \times 256}$$

変数Nは前置分周数(1,8,64,256,1024)を表します。

OCR0Aの両端値は高速PWM動作でPWM波形出力を生成する時の特別な場合にあたります。OCR0AがBOTTOM(\$00)と等しく設定されると、出力はMAX+1 タイマ/カケンタ クロック周期毎の狭いスパーイク(パールス)になるでしょう。OCR0AがMAXに等しく設定されると、(COM0A1,0ピットによって設定される出力極性に依存して)定常的なLowまたはHigh出力に終わるでしょう。

(<mark>訳注</mark>:以下は本動作種別がTOP=MAX(\$FF)で固定のため基本的に不適切です。) 高速PWM動作で(デューティ比50%)周波数の波形 出力は比較一致毎に論理反転するOC0A設定(COM0A1,0=01)によって達成できます。生成された波形はOCR0Aが0(\$00)に設定される時にfOC0A=fclk\_I/O/2の最大周波数でしょう。この特性は高速PWM動作で比較出力部の2重緩衝機能が許可されることを除いて、CTC動作でのOC0A交互出力(COM0A1,0=01)と同じです。

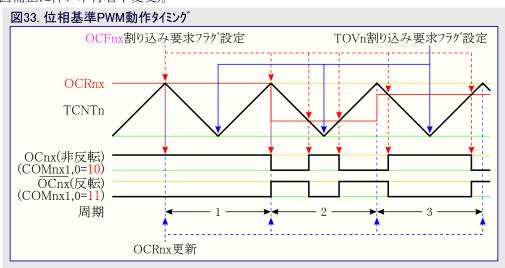




### 位相基準PWM動作

位相基準 $\rat^\circ$ ルス幅変調(PWM)動作(WGM01,0=01)は高分解能で正しい位相のPWM波形生成選択を提供します。位相基準PWM動作は両傾斜(三角波)動作を基準とします。カウンタはBOTTOMからMAXへそして次にMAXからBOTTOMへを繰り返し計数します。非反転比較出力動作(COM0A1,0=10)での比較出力(OC0A)は上昇計数中のTCNT0とOCR0Aの比較一致で解除(0)され、下降計数中の比較一致で設定(1)されます。反転出力動作(COM0A1,0=11)での動作は逆にされます。両傾斜(三角波)動作は単一傾斜(鋸波)動作よりも低い最大動作周波数になります。けれども両傾斜(三角波)動作の対称特性のため、これらの動作種別は電動機制御の応用に好まれます。

位相基準PWM動作のPWM分解能は8ビットに固定されます。位相基準PWM動作でのカウンタはカウンタ値がMAXと一致するまで増加されます。カウンタはMAXに達すると計数方向を変更します。このTCNT0値は1タイマ/カウンタ クロック周期間MAXと等しくなります。位相基準PWM動作のタイミング図は図33.で示されます。TCNT0値はタイミング図で両傾斜動作(三角波)を表す折れ線グラフとして示されます。この図は非反転と反転のPWM出力を含みます。細い赤線はOCR0A値を示し、TCNT0値との交点(接点)がTCNT0とOCR0A間の比較一致を示します(訳注:図補正に伴い本行若干変更)。



タイマ/カウンタ溢れ(TOV0)フラグはタイマ/カウンタがBOTTOMに到達する時毎に設定(1)されます。この割り込み要求フラグはカウンタがBOTT OM値に到達する毎に割り込みを発生するために使えます。

位相基準PWM動作での比較部はOCOAピンでのPWM波形の生成を許します。COM0A1,0ビットを'10'に設定することは非反転PWM 出力を作成し、反転PWM出力はCOM0A1,0ビットを'11'に設定することで生成できます(62頁の表52.をご覧ください)。実際のOC0A値はそのポート ピンに対するデータ方向(DDR\_OC0A)が出力として設定される場合だけ見えるでしょう。PWM波形はカウンタが増加する時のTCNT0とOCR0A間の比較一致でOC0A(内部)レジスタを設定(1)(または解除(0))と、カウンタが減少する時のTCNT0とOCR0A間の比較一致でOC0Aレジスタを解除(0)(または設定(1))によって生成されます。位相基準PWMを使う時の出力に対するPWM周波数は次式によって計算できます。

$$f_{\text{OCnxPCPWM}} = \frac{f_{\text{clk\_I/O}}}{N \times 510}$$

変数Nは前置分周数(1,8,64,256,1024)を表します。

OCR0Aの両端値は位相基準PWM動作でPWM波形出力を生成する時の特別な場合にあたります。非反転PWM動作ではOCR0AがBOTTOM(\$00)に等しく設定されると出力は定常的なLow、MAXに等しく設定されると定常的なHighになるでしょう。反転PWMに対する出力は逆の論理値になります。

図33.の第2周期のその出発点に於いて、例え比較一致がないとしても、OCnxにはHighからLowへの遷移があります。この遷移点はBOTTOMを挟む対称を保証するためです。比較一致なしに遷移を生ずるのは2つの場合です。

- 図33.でのようにOCR0AはMAXからその値を変更します。OCR0A値がMAXのとき、OCnxピン値は下降計数での比較一致の結果と同じです(<mark>訳補: L→H</mark>、直前がHのため、常にH)。BOTTOMを挟む対称を保証するため、(変更直後の)MAX(位置)でのOCnx値は上昇計数での比較一致の結果(H→L)と一致しなければなりません。
- タイマ/カウンタがOCR0A値よりも高い値から数え始め、そしてその理屈のために比較一致、それ故上昇途中で起こされるであろうOCnxの変更を逃します。(訳補: 従って上記同様、MAX位置で(直前がHならば)H→L遷移が生じます。)

### タイマ/カウンタのタイミング

このタイマ/カウンタは同期設計で、従って以下の図でタイマ/カウンタ クロック(clkT0)がクロック許可信号として示されます。この図は割り込みフラケが設定(1)される時の情報を含みます。図34.は基本的なタイマ/カウンタ動作についてのタイミング・データを含みます。この図は位相基準PWM動作以外の全ての動作種別でのMAX値近辺の計数の流れを示します。

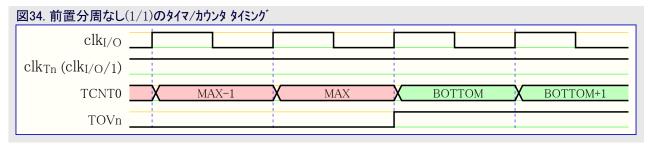


図35.は同じタイミング・データを示しますが、前置分周器が許可されています。

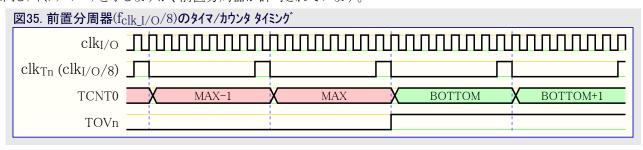


図36.はCTC動作を除く全ての動作種別でのOCF0Aの設定を示します。

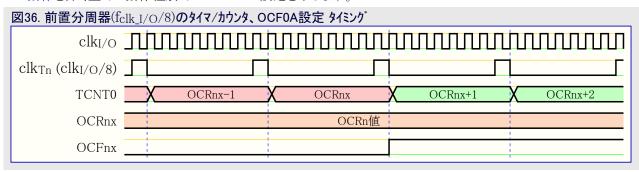
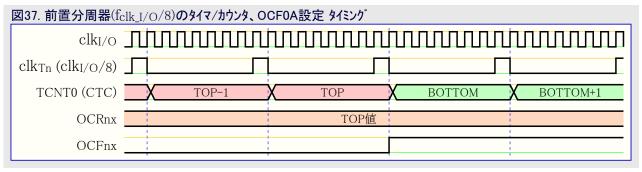


図37.はCTC動作でのTCNT0の解除とOCF0Aの設定を示します。







### 8ビット タイマ/カウンタ0 用レジスタ

タイマ/カウンタ0制御レシ、スタA (Timer/Counter0 Control Register A) TCCR0A

ピット	7	6	5	4	3	2	1	0	_
\$24 (\$44)	FOC0A	WGM00	COM0A1	COM0A0	WGM01	CS02	CS01	CS00	TCCR0A
Read/Write	W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

• ビット7 - FOCOA: OCOA強制変更 (Force Output Compare 0A)

FOC0AビットはWGM01,0ビットが非PWM動作を指示する時だけ有効です。けれども将来のデバイスとの共通性を保証するため、PWM動作で扱う時にTCCR0Aが書かれる場合、このビットは0に設定されなければなりません。FOC0Aビットに論理1を書くと、波形生成部で直ちに比較一致が強制されます。OC0A出力はCOM0A1,0ビット設定に従って変更されます。FOC0Aビットがストローブとして実行されることに注意してください。従って強制した比較の効果を決めるのはCOM0A1,0ビットに存在する値です。

FOC0Aストローブは何れの割り込みの生成もTOPとしてOCR0Aを使う比較一致タイマ解除(CTC)動作でのタイマ/カウンタの解除(\$00)も行いません。

FOC0Atiットは常に0として読みます。

• ビット3.6 - WGM01.0:波形生成種別 (Waveform Generation Mode bit 1 and 0)

これらのビットはカウンタの計数順序(方向)、最大 (TOP)カウンタ値の供給元、使われるべき波形生成のどの形式かを制御します。タイマ/カウンタ部によって支援される動作種別は標準動作、比較 一致タイマ/カウンタ解除(CTC)動作と2形式のパル ス幅変調(PWM)動作です。表49.と58頁の「動作 種別」をご覧ください。

	<u>1</u> 2 10.	ルスルノエル	ハモハックマル	`			
	番号	WGM01 (CTC0)	WGM00 (PWM0)	タイマ/カウンタ動作種別	TOP値	OCR0A 更新時	TOV0 設定時
	0	0	0	標準動作	\$FF	即時	MAX
I	1	0	1	位相基準PWM動作	\$FF	TOP	BOTTOM

0	0	0	標準動作	\$FF	即時	MAX
1	0	1	位相基準PWM動作	\$FF	TOP	BOTTOM
2	1	0	比較一致タイマ/カウンタ 解除(CTC)動作	OCR0A	即時	MAX
3	1	1	高速PWM動作	\$FF	BOTTOM	MAX

注: CTC0とPWM0ビット定義名は旧名称です。WGM01,0定義を使ってください。 しかし、これらのビットの機能と位置は旧版のタイマ/カウンタと一致します。

• L'ット5,4 - COMOA1,0:比較OA出力選択 (Compare Match OA Output Mode bit 1 and 0)

これらのビットはOC0A比較出力ピンの動作を制御します。COM0A1,0ビットの1つまたは両方が1を書かれると、OC0A出力はそのI/Oピンの通常ポート機能を無効にし、そのI/Oピンに接続されます。けれども出力駆動部を許可するため、OC0Aピンに対応するポート方向レジスタ(DDR)のビットが設定(1)されなければならないことに注意してください。

OC0Aがピンに接続されるとき、COM0A1,0ビットの機能はWGM01,0ビット設定に依存します。

表50.はWGM01,0ビットが標準動作またはCTC動作(つまりPWM以外)に設定される時のCOM0A1,0ビット機能を示します。

**表51**.はWGM01,0ビットが高速PWM動作に設定される時のCOM0 A1,0ビットの機能を示します。

**表52**.はWGM01,0ビットが位相基準PWM動作に設定される時のC OM0A1,0ビットの機能を示します。

COM0A1	COM0A0	意味
0	0	標準ポート動作 (OC0A切断)
0	1	比較一致でOC0Aピン トグル(交互)出力
1	0	比較一致でOC0Aピン Lowレベル出力
1	1	比較一致でOC0At°ン Highレヘブル出力

#### 表51. 高速PWM動作での比較出力選択

COM0A1	COM0A0	意味
0	0	標準ポート動作 (OC0A切断)
0	1	(予約)
1	0	比較一致でLow、BOTTOMでHighを OCOAピンへ出力(非反転動作)
1	1	比較一致でHigh、BOTTOMでLowを OCOAピン〜出力(反転動作)

注: COM0A1が設定(1)され、OCR0AがTOPと等しい時に特別な状態が起きます。この状態での比較一致は無視されますが、BOTTOMでの設定(1)や解除(0)は行われます。より多くの詳細については59頁の「高速PWM動作」をご覧ください。

#### 表52. 位相基準PWM動作での比較出力選択

COM0A1	COM0A0	意味
0	0	標準ポート動作 (OC0A切断)
0	1	(予約)
1	0	上昇計数時の比較一致でLow、下降計数 時の比較一致でHighをOCOAピンへ出力
1	1	上昇計数時の比較一致でHigh、下降計数 時の比較一致でLowをOCOAt。ンへ出力

注: COM0A1が設定(1)され、OCR0AがTOPと等しい時に特別な状態が起きます。この状態での比較一致は無視されますが、TOPでの設定(1)や解除(0)は行われます。より多くの詳細については60頁の「位相基準PWM動作」をご覧ください。

• ビット2~0 - CS02~0: クロック選択0 (Clock Select0, bit 2,1 and 0)

この3つのクロック選択ビットはタイマ/カウンタ(TCNT0)によって使われるクロック元を選びます。

#### 表53. タイマ/カウンタ0入力クロック選択

CS02	CS01	CS00	意味
0	0	0	停止(タイマ/カウンタ0動作停止)
0	0	1	clk <sub>I/O</sub> (前置分周なし)
0	1	0	clk <sub>I/O</sub> /8 (8分周)
0	1	1	clk <sub>I/O</sub> /64 (64分周)
1	0	0	clk <sub>I/O</sub> /256 (256分周)
1	0	1	clk <sub>I/O</sub> /1024 (1024分周)
1	1	0	T0ピンの下降端 (外部クロック)
1	1	1	T0ピンの上昇端 (外部クロック)

外部ピン(クロック)動作がタイマ/カウンタ0に対して使われる場合、例えT0ピンが出力として設定されても、T0ピンの遷移はカウンタをクロック駆動します。この特徴がソフトウェアに計数制御を許します。

# タイマ/カウンタO (Timer/Counter0) TCNTO

ピット	7	6	5	4	3	2	1	0	_
\$26 (\$46)	(MSB)							(LSB)	TCNT0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

このタイマ/カウンタ レジスタは読み書き両方の操作について、タイマ/カウンタ部の8ビット カウンタに直接アクセスします。TCNT0への書き込みは次のタイマ/カウンタ クロックでの比較一致を妨害(除去)します。カウンタが走行中にカウンタ(TCNT0)を変更することはTCNT0とOCR0A間の比較一致消失の危険を誘発します。

# タイマ/カウンタ0 比較Aレジスタ (Timer/Counter0 Output Compare A Register) OCROA

ピット	7	6	5	4	3	2	1	0	
\$27 (\$47)	(MSB)							(LSB)	OCR0A
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

この比較レシ、スタは継続的にカウンタ(TCNT0)値と比較される8ビットの値を含みます。一致は比較一致割り込みやOC0Aピンでの波形出力を生成するのに使えます。





### タイマ/カウンタ0割り込み許可レジスタ (Timer/Counter 0 Interrupt Mask Register) TIMSKO

ピット	7	6	5	4	3	2	1	0	_
(\$6E)	-	-	-	-	-	-	OCIE0A	TOIE0	TIMSK0
Read/Write	R	R	R	R	R	R	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

• ビット1 - OCIEOA: タイマ/カウンタ0比較A割り込み許可 (Timer/Counter0 Output Compare Match A Interrupt Enable)

OCIEOAビットが1を書かれ、ステータスレジスタ(SREG)の全割り込み許可(I)ビットが設定(1)されると、タイマ/カウンタ0比較A一致割り込みが許可されます。タイマ/カウンタ0で比較一致が起こる、換言するとタイマ/カウンタ0割り込み要求フラケ、レジスタ(TIFR0)で比較割り込み要求フラケ、(OCF0A)が設定(1)されると、対応する割り込みが実行されます。

• ビット0 - TOIE0: タイマ/カウンタ0溢れ割り込み許可 (Timer/Counter0 Overflow Interrupt Enable)

TOIE0ビットが1を書かれ、ステータスレジ、スタ(SREG)の全割り込み許可(I)ビットが設定(1)されると、タイマ/カウンタ0溢れ割り込みが許可されます。タイマ/カウンタ0溢れが起こる、換言するとタイマ/カウンタ0割り込み要求フラケ、レジ、スタ(TIFRO)でタイマ/カウンタ0溢れ割り込み要求(TOVO)フラケが設定(1)されると、対応する割り込みが実行されます。

#### タイマ/カウンタ0割り込み要求フラク・レシ、スタ (Timer/Counter 0 Interrupt Flag Register) TIFRO

ピット	7	6	5	4	3	2	1	0	_
\$15 (\$35)	_	-	-	-	-	-	OCF0A	TOV0	TIFR0
Read/Write	R	R	R	R	R	R	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

• ビット1 - OCFOA: タイマ/カウンタの比較A割り込み要求フラグ (Timer/Conter0, Output Compare A Match Flag)

OCF0Aビットは比較一致がタイマ/カウンタ(TCNT0)と比較レジスタ(OCR0A)間で起こる時に設定(1)されます。対応する割り込み処理ベクタを実行すると、OCF0Aはハードウェアによって解除(0)されます。代わりにこのフラグへ論理1を書くことによってもOCF0Aは解除(0)されます。ステータス レジスタ(SREG)の全割り込み許可(I)ビット、タイマ/カウンタ0割り込み許可レジスタ(TIMSK0)のタイマ/カウンタ0比較一致割り込み許可(OCIE0A)ビット、OCF0Aが設定(1)されると、タイマ/カウンタ0比較一致割り込みが実行されます。

• ビット0 - TOV0: タイマ/カウンタ0溢れ割り込み要求フラグ(Timer/Counter0 Overflow Flag)

TOV0ビットはタイマ/カウンク(TCNT0)溢れが起こる時に設定(1)されます。対応する割り込み処理べクタを実行すると、TOV0はハードウェアによって解除(0)されます。代わりにこのフラグへ論理1を書くことによってもTOV0は解除(0)されます。ステータス レジスタ(SREG)の全割り込み許可(I)ビット、タイマ/カウンタ0割り込み許可レジスタ(TIMSK0)のタイマ/カウンタ0溢れ割り込み許可(TOIE0)ビット、TOV0が設定(1)されると、タイマ/カウンタ0溢れ割り込みが実行されます。位相基準PWM動作ではタイマ/カウンタ0が\$00で計数方向を変える時にこのビットが設定(1)されます。

# タイマ/カウンタ0とタイマ/カウンタ1の前置分周器

タイマ/カウンタ0とタイマ/カウンタ1は同じ前置分周器部を共用しますが、タイマ/カウンタは異なる前置分周器設定ができます。以下の記述はタイマ/カウンタ0とタイマ/カウンタ1の両方に適用されます。

#### 内部クロック元

タイマ/カウンタはシステム クロック(CSn2~0=001設定)によって直接的にクロック駆動できます。これはシステム クロック周波数( $f_{clk\_I/O}$ )と等しいタイマ / カウンタ最大クロック周波数での最高速動作を提供します。選択で前置分周器からの4つのタップの1つがクロック元として使えます。この前置分周したクロックは $f_{clk\_I/O}$ /8,  $f_{clk\_I/O}$ /64,  $f_{clk\_I/O}$ /256,  $f_{clk\_I/O}$ /1024の何れかの周波数です。

#### 前置分周器リセット

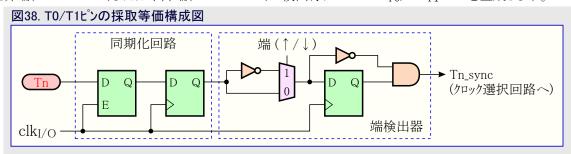
この前置分周器は自由走行で(換言するとタイマ/カウンタのクロック選択論理回路と無関係に動作する)、タイマ/カウンタ0とタイマ/カウンタ1によって共用されます。前置分周器はタイマ/カウンタのクロック選択によって影響を及ぼされないため、前置分周器の状態は前置分周したクロックが使われる状況に対して密接に関係します。タイマ/カウンタが許可され、前置分周器によってクロック駆動される(CSn2~0=5~2)とき、前置分周加工の一例が生じます。タイマ/カウンタが許可される時から最初の計数が起きるまでのシステムクロック周期数はNが前置分周値(8,64,256,1024)とすると、1~N+1システムクロック周期になり得ます。

プログラム実行にタイマ/カウンタを同期することに対して前置分周器リセットを使うことが可能です。しかし、同じ前置分周器を共用する他のタイマ/カウンタも前置分周を使う場合、注意が必要とされなければなりません。前置分周器リセットはそれが接続される全タイマ/カウンタに関する前置分周器周期に影響を及ぼします。

#### 外部クロック元

T0/T1ピンに印加された外部クロック元はタイマ/カウンタ クロック( $f_{clk\_T0}/f_{clk\_T1}$ )として使えます。このT0/T1ピンはピン同期化論理回路によって全てのシステム クロック周期に一度採取されます。この同期化(採取)された信号はその後に端(エッジ)検出器を通して通過されます。図 38.はT0/T1同期化と端検出器論理回路の機能等価構成図を示します。レジスタは内部システム クロック( $f_{clk\_I/O}$ )の上昇端でクロック駆動されます。ラッチは内部システム クロックのHigh区間で通過(Low区間で保持)です。

端検出器は上昇端(CSn2~0=111)または下降端(CSn2~0=110)の検出毎に1つのclkT0/clkT1パルスを生成します。



同期化と端検出器論理回路はT0/T1ピンへ印加された端から計数器が更新されるまでに2.5~3.5システム クロック周期の遅延をもたらします。

クロック入力の許可と禁止はT0/T1が最低1システム クロック周期に対して安定してしまっている時に行われなければならず、さもなければ 不正なタイマ/カウンタ クロック パルスが生成される危険があります。

印加された外部クロックの各半周期は正しい採取を保証するために1システム クロック周期より長くなければなりません。この外部クロックは50%/50%デューティ比で与えられるものとして、システム クロック周波数の半分未満(fextelk/felk\_I/O/2)であることが保証されなければなりません。端検出器が採取を使うため、検出できる外部クロックの最大周波数は採取周波数の半分です(ナイキストの標本化定理)。然しながら、発振元(クリスタル発振子、セラミック振動子、コンデンサ)公差によって引き起こされたシステム クロック周波数やデューティ比の変動のため、外部クロック元の最大周波数はfelk\_I/O/2.5未満が推奨されます。

外部クロック元は前置分周できません。





# 図39. タイマ/カウンタ0とタイマ/カウンタ1の前置分周器部構成図 10ビット前置分周器 clk<sub>I</sub>/O clk<sub>I</sub>/0/256 PSR10 $clk_{I/O}/1024$ clk<sub>I/O/</sub> 同期化/端検出 0 同期化/端検出 CS10 CS00 -タイマ/カウンタ1用 タイマ/カウンタ0用 CS11 CS01 -選択器 選択器 CS12 -CS02 clkT1 clk<sub>T0</sub> タイマ/カウンタ0 クロック タイマ/カウンタ1 クロック

注: 入力ピン(T0/T1)の同期化/端(エッシ)検出論理回路は図38.で示されます。

### 一般タイマ/カウンタ制御レジスタ (General Timer/Counter Control Register) GTCCR

ピット	7	6	5	4	3	2	1	0	
\$23 (\$43)	TSM	-	-	-	-	-	PSR2	PSR10	GTCCR
Read/Write	R/W	R	R	R	R	R	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

#### • ビット7 - TSM: タイマ/カウンタ同時動作 (Timer/Counter Synchronization Mode)

TSMビットに1を書くことはタイマ/カウンタ同期化動作を活性(有効)にします。この動作でPSR2とPSR10へ書かれる値は保持され、従って対応する前置分周器リセット信号の有効を保持します。これは対応するタイマ/カウンタを停止し、設定中にそれらの1つが進行する危険なしに同じ値に設定できることを保証します。TSMビットが0を書かれると、PSR2とPSR10ビットはハードウェアによって解除(0)され、同時にタイマ/カウンタが計数を始めます。

#### • ビット0 - PSR10: タイマ/カウンタ1,0 前置分周器リセット (Prescaler Reset Timer/Counter 1,0)

このビットが1のとき、タイマ/カウンタ0とタイマ/カウンタ1の前置分周器はリセットします。TSMビットが設定(1)されている場合を除き、通常、このビッ トはハードウェアによって直ちに解除(0)されます。タイマ/カウンタ0とタイマ/カウンタ1は同じ前置分周器を共用し、この前置分周器のリセットが両方のタイマ/カウンクに影響を及ぼすことに注意してください。

# 16ビット タイマ/カウンタ1

この16ビット タイマ/カウンタ部は正確なプログラム実行タイミング(事象管理)、波形生成、信号タイミング計測を許します。主な特徴は次に示されます。

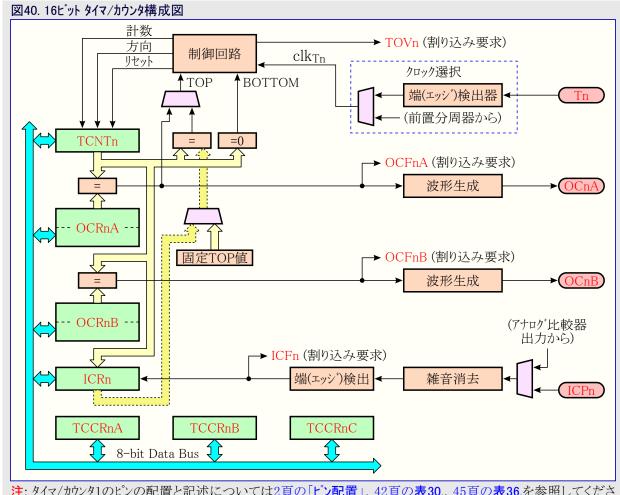
- 真の16ビット設計(換言すれば16ビットPWMの許容)
- 2つの独立した比較出力部
- 2重緩衝の比較レジスタ
- •1つの捕獲入力部
- 捕獲入力雜音消去器
- 比較一致でのタイマ/カウンタ解除(自動再設定)
- 不具合なしで正しい位相のパルス幅変調器 (PWM)
- 可変PWM周期
- 周波数発生器
- 外部事象計数器
- 4つの独立した割り込み (TOV1,OCF1A,OCF1B,ICF1)

#### 概要

本文書でのレシ、スタとヒット参照の多くは一般形で書かれます。小文字の'n'はタイマ/カウンタ番号、小文字の'x'は比較出力部の置き換えです。けれどもプログラムでレシ、スタまたはヒット定義に使う時は正確な形式が使われなければなりません(例えばタイマ/カウンタ1のカウンタ値に対するアクセスのTCNT1のように)。

この16ビット タイマ/カウンタの簡単化した構成図は**図40**.で示されます。実際のI/Oピンの配置については2頁の「ピ<mark>ン配置</mark>」を参照してください。CPUがアクセス可能な(I/OビットとI/Oピンを含む)I/Oレジスタは赤文字(<mark>訳注</mark>: 原文太字)で示されます。デバイス仕様のI/Oレジスタとビット位置は81頁の「16ビット タイマ/カウンタ1用レジスタ」で示されます。

23頁の「電力削減レジスタ(PRR)」のPRTIM1ビットはタイマ/カウンタ1部を許可するために0を書かれなければなりません。



**注**: タイマ/カウンタ1のピンの配置と記述については2頁の「<mark>ピン配置」、42頁の表30.、45頁の表36.</mark>を参照してください。





#### 関係レジスタ

gイマ/カウンタ(TCNT1)、比較レシ、スタ(OCR1A,OCR1B)、捕獲レシ、スタ(ICR1)は全て16ビットレジ、スタです。16ビットレジ、スタをアクセスするとき、特別な手順に従わなければなりません。これらの手順は69頁の「16ビットレジ、スタのアクセス」項で記述されます。タイマ/カウンタ1制御レジ、スタ(TCCR1A,TCCR1B,TCCR1C)は8ビットレジ、スタで、CPUアクセスの制限はありません。割り込み要求信号はタイマ/カウンタ1割り込み要求レジ、スタ(TIFR1)で全て見えます。全ての割り込みはタイマ/カウンタ1割り込み許可レジ、スタ(TIMSK1)で個別に遮蔽(禁止)されます。TIFR1とTIMSK1はこの図で示されません。

このタイマ/カウンタは内部的、前置分周器経由、またはT1ピンの外部クロック元によってクロック駆動されます。クロック選択論理部はタイマ/カウンタが値を増加(または減少)するのに使うクロック元と端(エッシ)を制御します。クロック元が選ばれないと、タイマ/カウンタは動きません。クロック選択論理部からの出力はタイマ/カウンタ クロック(clkT1)として参照されます。

2重緩衝化した比較レシ、スタ(OCR1A,OCR1B)はタイマ/カウンタ値と常に比較されます。この比較結果は比較出力(OC1A,OC1B)ピンで PWMまたは可変周波数出力を生成するための波形生成器によって使えます。73頁の「比較出力部」をご覧ください。この比較一致発生は比較出力割り込み要求の発生に使える比較一致割り込み要求フラグ(OCF1A,OCF1B)も設定(1)します。

捕獲レジスタ(ICR1)は捕獲起動(ICP1)ピンまたはアナログ比較器出力(130頁の「**アナログ比較器**」参照)のどちらかの外部(端で起動された)事象でタイマ/カウンタ値を捕獲(複写)できます。捕獲入力部は尖頭雑音を捕らえる機会を軽減するためにデジタル濾波器(雑音消去器)を含みます。

TOP値または最大947/hが9値は、いくつかの動作種別で、OCR1A、ICR1、または一群の固定値のどれかによって定義できます。 PWM動作でTOP値としてOCR1Aを使うと、OCR1AはPWM出力生成用に使えません。 けれどもこの場合、TOP値は動作中に変更されるのをTOP値に許す2重緩衝化します。 固定的なTOP値が必要とされる場合、ICR1が代わりに使え、PWM出力として使われるべき OCR1Aを開放します。

### 定義

次の定義は本文書を通して広範囲に使われます。

表54. 用語定	表54. 用語定義								
用語	意味								
BOTTOM	カウンタが\$0000に到達した時。								
MAX	カウンタが \$FFFF(65535)に到達した時。								
TOP	カウンタがTOP値に到達した時(計数動作での最大値と等しくなった時)。TOP値は固定値(\$00FF,\$01FF,\$03FF)、OCR1A値、ICR1値の何れか1つを指定できます。この指定は動作種別に依存します。								

#### 互換性

この16ビット タイマ/カウンタは旧版の16ビット AVR タイマ/カウンタから改良更新されてしまっています。この16ビット タイマ/カウンタは次の点に関して以前の版と完全な互換性があります。

- タイマ/カウンタ割り込み関係レジスタを含め、16ビットタイマ/カウンタに関連する全てのI/Oレジスタアドレスの位置。
- タイマ/カウンタ割り込み関係レジスタを含め、16ビットタイマ/カウンタに関連する全てのI/Oレジスタ内のビット位置。
- •割り込みべかり。

次の制御ビットは変更された名前を持ちますが、同じ機能とレジスタ位置を持ちます。

- PWM10はWGM10に変更。
- PWM11はWGM11に変更。
- CTC1はWGM12に変更。

次の制御ビットが16ビット タイマ/カウンタ制御レジスタ内に追加されます。

- FOC1AとFOC1Bが(新設の)TCCR1Cに追加。
- WGM13がTCCR1Bに追加。

この16ビットタイマ/カウンタにはいくつかの特別な状況で互換性に影響を及ぼす改良点があります。

### 16ビット レジスタのアクセス

TCNT1,OCR1A,OCR1B,ICR1は8ビット バス経由でAVR CPUによってアクセスできる16ビット レジスタです。この16ビット レジスタは2回の読みまたは書き操作を使ってバイト アクセスされなければなりません。16ビット タイマ/カウンタは16ビット アクセスの上位バイトの一時保存用に1つの8 ビット レジスタを持ちます。16ビット タイマ/カウンタ内の全ての16ビット レジスタ間で、この同じ一時レジスタが共用されます。下位バイト アクセスが16ビット読み書き動作を起動します。16ビット レジスタの下位バイトがCPUによって書かれると、一時レジスタに保存した上位バイトと書かれた下位バイトは同じクロック周期で両方が16ビット レジスタに複写されます。16ピット レジスタの下位バイトがCPUによって読まれると、16ビット レジスタの上位バイトは下位バイトが読まれるのと同じクロック周期で一時レジスタに複写されます。

全ての16ビットアクセスが上位バイトに対して一時レジスタを使う訳ではありません。OCR1AとOCR1Bの16ビットレジスタ読み込みは一時レジスタの使用に関係しません。

16ビット書き込みを行うために上位バイトは下位バイトに先立って書かれなければなりません。16ビット読み込みについては下位バイトが上位バイトの前に読まれなければなりません。

次のコート・例は割り込みが一時レシ、スタを更新しないことが前提の16ビットタイマ/カウンタレジ、スタのアクセス法を示します。OCR1A, OCR1B, ICR1レジ、スタのアクセスに対して同じ原理が直接的に使えます。C言語を使う時はコンパイラが16ビットアクセスを扱うことに注意してください。



アセンブリ言語コート・例はR17:R16レジスタ対にTCNT1値を戻します。

16ビット レジスタ アクセスが非分断操作であることに注意することが重要です。16ビット レジスタをアクセスする2命令間で割り込みが起き、割り込みコードがその16ビット タイマ/カウンタ レジスタの同じ若しくは他の何れかをアクセスすることによって一時レジスタを更新する場合、割り込み外のその後のアクセス結果は不正にされます。従って主コードと割り込みコードの両方が一時レジスタを更新するとき、主コードは16ビット アクセス中の割り込みを禁止しなければなりません。





次のコード例はTCNT1レジスタ内容の非分断読み込み法を示します。同じ原理を使うことにより、OCR1A,OCR1B,ICR1のどの読み込み も行えます。

```
アセンブリ言語プログラム例
RD_TCNT1:
                                                  ;現全割り込み許可フラグ(I)を保存
           IN
                 R18, SREG
           CLI
                                                  ;全割り込み禁止
                                                  ;TCNT1下位バイト取得(上位バイト⇒一時レジスタ)
                 R16, TCNT1L
           ΙN
                 R17, TCNT1H
                                                  ;TCNT1上位バイ取得(一時レジスタ)
           ΤN
           OUT
                 SREG, R18
                                                  ;全割り込み許可フラグ(I)を復帰
           RET
                                                  ;呼び出し元へ復帰
C言語プログラム例
unsigned int TIM16_Read_TCNT1(void)
   unsigned char sreg;
                                                  /* ステータス レシブスター時保存変数定義 */
   unsigned int i;
                                                  /* TCNT1読み出し変数定義 */
   sreg = SREG;
                                                  /* 現全割り込み許可フラグ(I)を保存 */
                                                  /* 全割り込み禁止 */
   __disable_interrupt();
   i = TCNT1;
                                                  /* TCNT1値を取得 */
                                                  /* 全割り込み許可フラグ(I)を復帰 */
   SREG = sreg;
   return i;
                                                  /* TCNT1値で呼び出し元へ復帰 */
```

#### 注: 5頁の「コート'例について」をご覧ください。

アセンブリ言語コート・例はR17:R16レシブスタ対にTCNT1値を戻します。

次のコート・例はTCNT1レシ、スタ内容の非分断書き込み法を示します。同じ原理を使うことにより、OCR1A,OCR1B,ICR1のどの書き込み も行えます。

```
アセンブリ言語プログラム例
WR TCNT1:
           IN
                 R18, SREG
                                                   ;現全割り込み許可フラグ(I)を保存
                                                   ;全割り込み禁止
           CLI
                                                   ;TCNT1上位バイ設定(一時レジスタ)
           OUT
                 TCNT1H, R17
           OUT
                 TCNT1L, R16
                                                   ;TCNT1下位バイト設定(一時レジスタ⇒上位バイト)
           OUT
                 SREG, R18
                                                   ;全割り込み許可フラグ(I)を復帰
           RET
                                                   ;呼び出し元へ復帰
C言語プログラム例
void TIM16_Write_TCNT1(unsigned int i)
                                                  /* ステータス レシブスター時保存変数定義 */
   unsigned char sreg;
   unsigned int i;
                                                  /* TCNT1書き込み変数定義 */
   sreg = SREG;
                                                  /* 現全割り込み許可フラグ(I)を保存 */
    __disable_interrupt();
                                                  /* 全割り込み禁止 */
                                                  /* TCNT1値を設定 */
   TCNT1 = i;
                                                  /* 全割り込み許可フラグ(I)を復帰 */
   SREG = sreg;
```

#### 注: 5頁の「コード例について」をご覧ください。

アセンブリ言語コート・例はR17:R16レシ、スタ対がTCNT1へ書かれるべき値を含むことが必要です。

#### 上位バイト一時レジスタの再使用

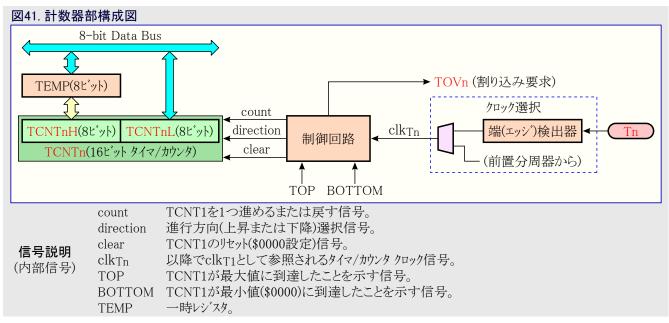
書かれる全レジスタについて上位バイトが同じ複数16ビット レジスタ書き込みなら、上位バイトは1度書かれることだけが必要です。 けれども 直前で記述した非分断操作の同じ規則が、この場合にも適用されることに注意してください。

#### タイマ/カウンタのクロック

このタイマ/カウンタは内部または外部のクロック元によってクロック駆動できます。このクロック元はタイマ/カウンタ制御レシ、スタB(TCCR1B)に配置されたクロック選択(CS12~0)ビットによって制御されるクロック選択論理回路で選ばれます。クロック元と前置分周器の詳細については65頁の「タイマ/カウンタ0とタイマ/カウンタ1の前置分周器」をご覧ください。

### 計数器部

16ビット タイマ/カウンタの主な部分は設定可能な16ビット双方向カウンタ部です。図41.はこのカウンタとその周辺の構成図を示します。



この16ビット カウンタはカウンタの上位8ビットを含むカウンタ上位(TCNT1H)と下位8ビットを含むカウンタ下位(TCNT1L)の2つの8ビット I/Oメモリ位置に配置されます。TCNT1HレジスタはCPUによる間接的なアクセスのみできます。CPUがTCNT1H I/O位置をアクセスするとき、CPUは上位バイトー時レジスタ(TEMP)をアクセスします。この一時レジスタはTCNT1Lが読まれる時にTCNT1H値で更新され、TCNT1Lが書かれる時にTCNT1Hは一時レジスタ値で更新されます。これは8ビット データ バス経由で1クロック周期内での16ビット カウンタ値全体の読み書きをCPUに許します。予測不能な結果を生じる、カウンタが計数中の時のTCNT1書き込みの特別な場合に注意することが重要です。この特別な場合はそれらが重要となる項目で記述されます。

使った動作種別に依存して、カウンタは各タイマ/カウンタ クロック(clkT1)でリセット(\$0000)、増加(+1)、または減少(-1)されます。clkT1はクロック 選択(CS12~0)ビットによって選ばれた内部または外部のクロック元から生成できます。クロック元が選ばれない(CS12~0=000)時にカウンタは 停止されます。けれどもTCNT1値はタイマ/カウンタ クロック(clkT1)が存在するしないに拘らず、CPUによってアクセスできます。CPU書き込 みは全てのカウンタ解除や計数動作を無視します(上位優先順位を持ちます)。

計数順序(方法)はタイマ/カウンタ制御レジ、スタA(TCCR1A)とタイマ/カウンタ制御レジ、スタB(TCCR1B)に配置された波形生成種別(WGM13~0) ビットの設定によって決定されます。これらはカウンタ動作(計数)方法と波形がOC1x比較出力に生成される方法間の接続に近いものです。進化した計数順序と波形生成についてより多くの詳細に対しては76頁の「動作種別」をご覧ください。

タイマ/カウンタ溢れ(TOV1)フラグはWGM13~0ビットによって選ばれた動作種別に従って設定(1)されます。TOV1はCPU割り込み発生に使えます。

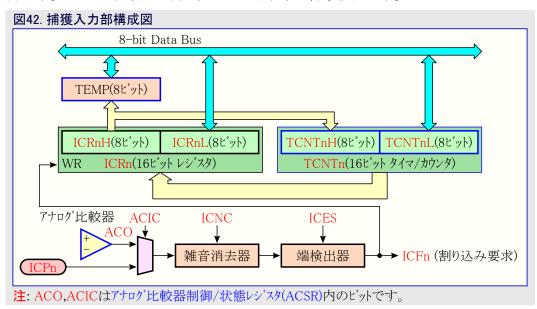




### 捕獲入力部

タイマ/カウンタは外部の出来事を捕獲でき、発生時間を示す時間印(タイマ/カウンタ値)を与える捕獲入力部と合体します。出来事または複数の出来事を示す外部信号はICP1ピンまたは代わりにアナロケ、比較器部経由で印加できます。時間印はその後、周波数、デューティ比、印加された信号の他の特性の計算に使えます。代わりに時間印は出来事の記録作成にも使えます。

捕獲入力部は**図42**.で示される構成図によって図解されます。直接的な捕獲入力部の部分でない構成図の要素は青枠(<mark>訳注</mark>:原文は灰色背景)で示されます。レジスタとビット名での小文字の'n'はタイマ/カウンタ番号を示します。



捕獲起動入力(ICP1)ピン若しくは代わりにアナロケ・比較器出力(ACO)で論理レヘ・ルの変化(出来事)が起き、その変化が端(エッジ)検出器の設定を追認すると、捕獲が起動されます。捕獲が起動されると、カウンタ(TCNT1)の16ビット値が捕獲レジ、スタ(ICR1)に書かれます。捕獲割り込み要求フラケ (ICF1)はTCNT1値がICR1に複写されるのと同じシステム クロックで設定(1)されます。許可(I=1,ICIE1=1)ならば捕獲割り込み要求フラケ は捕獲割り込みを発生します。ICF1は割り込みが実行されると自動的に解除(0)されます。代わりにこのI/Oビット位置に論理1を書くことによってソフトウェアでも解除(0)できます。

捕獲レジスタ(ICR1)の16ビット値読み込みは、初めに下位バイト(ICR1L)、その後に上位バイト(ICR1H)を読むことによって行われます。下位バイトが読まれる時に上位バイトが上位バイト一時レジスタ(TEMP)に複写されます。CPUがICR1H I/O位置を読むと、この一時レジスタをアクセスします。

ICR1はカウンタのTOP値定義にICR1を利用する波形生成種別を使う時にだけ書けます。これらの場合、TOP値がICR1に書かれ得る前に波形生成種別(WGM13~0)ビットが設定されなければなりません。ICR1に書く時は下位バイトがICR1Lに書かれる前に、上位バイトがICR1H I/O位置に書かれなければなりません。

16ビット レジスタ アクセス法のより多くの情報については69頁の「16ビット レジスタのアクセス」を参照してください。

#### 捕獲起動元

捕獲入力部用の主な起動元は捕獲起動入力(ICP1)ピンです。タイマ/カウンタ1は捕獲入力部用起動元としてアナログ比較器出力を代わりに使えます。アナログ比較器はアナログ比較器制御/状態レシ、スタ(ACSR)のアナログ比較器捕獲起動許可(ACIC)ピットの設定(1)によって起動元として選ばれます。起動元を変更することが捕獲を起動し得ることに気付いてください。従って捕獲割り込み要求フラグ(ICF1)は、その変更後に解除(0)されなければなりません。

捕獲起動入力(ICP1)ピンとアナログ比較器出力(ACO)の両入力は、T1ピン(65頁の**図38**.参照)についてと同じ技法を使って採取されます。端検出器も全く同じです。けれども雑音消去が許可されると、付加論理回路が端検出器の前に挿入され、そして遅延を4システムクロック周期増やします。タイマ/カウンタがTOP値定義にICR1を使う波形生成種別に設定されないなら、雑音消去器と端検出器の入力が常に許可されることに注意してください。

捕獲入力はICP1ピンのポートを制御することによりソフトウェアで起動できます。

### 雑音消去器

雑音消去器は簡単なデジタル濾波器機構を使うことによって雑音耐性を改善します。雑音消去器の入力は4採取に渡って監視され、端検出器によって使われる方向転換となる出力を変更するためには4回全てが同じでなければなりません。

雑音消去器はタイマ/カウンタ制御レジスタB(TCCR1B)の捕獲入力雑音消去許可(ICNC1)ビットの設定(1)によって許可されます。許可したとき、雑音消去器は入力に印加した変更からICR1の更新までに4システム クロック周期の追加遅延をもたらします。雑音消去器はシステムクロックを使い、従って前置分周器によって影響されません。

## 捕獲入力の使用

捕獲入力機能を使う主な要求(目的)は入って来る出来事に対して充分なプロセッサ能力を当てがうことです。2つの出来事間の時間が際どいとします。次の出来事が起こる前に捕獲した捕獲レジスタ(ICR1)の値をプロセッサが読めなかった場合、ICR1は新しい値で上書きされます。この場合、捕獲の結果は不正にされます。

捕獲割り込みを使う時にICR1は割り込み処理ルーチンで可能な限り早く読まれるべきです。捕獲割り込みが相対的に高い優先順位であっても、最大割り込み応答時間は他の割り込み要求のどれかを扱うのに必要とされる最大クロック周期数に依存します。

動作中にTOP値(分解能)が積極的に変更されるとき、どの動作種別での捕獲入力部の使用も推奨されません。

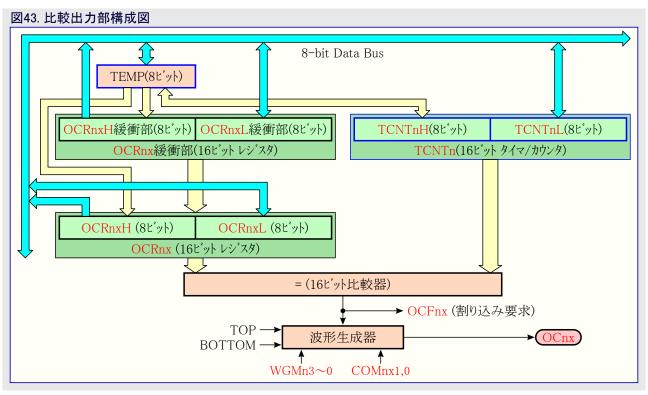
外部信号のデューティ比測定は各捕獲後に起動端が変更されることを必要とします。検出端の変更はICR1が読まれてしまった後に可能な限り早く行われなければなりません。端(エッシ)の変更後、捕獲割り込み要求フラケ(ICF1)はソフトウェア(I/Oビット位置への論理1書き込み)によって解除(0)されなければなりません(訳補:端変更によってICF1が設定(1)されることを想定)。周波数のみの測定については(割り込み処理が使われる場合)、ICF1の解除(0)は必要とされません。

## 比較出力部

この16t'ット比較器はTCNT1と比較レシ'スタ(OCR1x)を継続的に比較します。TCNT1とOCR1xが等しければ、比較器は一致を指示します。この一致は次のタイマ/カウンタ クロック周期で比較割り込み要求フラク'(OCF1x)を設定(1)します。許可(I=1,OCIE1x=1)なら、この比較割り込み要求フラク'は比較割り込みを発生します。OCF1xは割り込みが実行されると自動的に解除(0)されます。代わりにOCF1xはこのI/Ot'ット位置に論理1を書くことによってソフトウェアでも解除(0)できます。波形生成器は波形生成種別(WGM13~0)t'ットと比較出力選択(COM1x1,0)t'ットによって設定された動作種別に従った出力を生成するのにこの一致信号を使います。TOPとBOTTOM信号は動作種別(76頁の「動作種別」参照)のいくつかで両端値の特別な場合を扱うため、波形生成器によって使われます。

比較A出力部の特殊な特性はタイマ/カウンタのTOP値(換言するとカウンタの分解能)定義を許します。カウンタの分解能に加え、TOP値は波形生成器によって生成された波形の周期時間を定義します。

図43.は比較出力部の構成図を示します。レシブスタとビット名での小文字の'n'はタイマ/カウンタ番号(タイマ/カウンタ1に対してはn=1)、小文字の'x'は比較出力部(AまたはB)を表します。直接的な比較出力部の部分でない構成図の要素は青枠(訳注:原文は灰色背景)で示されます。







OCR1xは12種類のパルス幅変調(PWM)のどれかを使う時に2重緩衝化されます。標準動作と比較一致タイマ/カウンタ解除(CTC)動作については2重緩衝動作が禁止されます。2重緩衝動作は計数の流れのTOPまたはBOTTOMのどちらかに対してOCR1xレジスタの更新を同期化します。この同期化は奇数長、非対称PWMパルスの発生を防ぎ、それによって不具合なしの出力を作成します。

OCR1xのアクセスは複雑なように思えますが決してそんなことはありません。2重緩衝動作が許可されるとCPUはOCR1x緩衝部をアクセスし、禁止されるとOCR1xレジ、スタを直接アクセスします。OCR1x(緩衝部またはレジ、スタ)の内容は書き込み操作によってのみ変更されます(タイマ/カウンタはTCNT1やICR1のようにOCR1xを自動的に更新しません)。従ってOCR1xは上位ハイルー時レジ、スタ(TEMP)経由で読まれません。けれども他の16ビットレジ、スタをアクセスする時のように下位ハイトを先に読むのは良い習慣です。OCR1x書き込みは16ビット全ての比較が継続的に行われるため、一時レジ、スタ経由で行われなければなりません。上位ハイト(OCR1xH)が先に書かれなければなりません。上位ハイト(OCR1xH)が先に書かれなければなりません。上位ハイト(OCR1xH)が先に書かれなければなりません。上位ハイト(OCR1xH)が方に多ビットを書かれると、一時レジ、スタ内の)上位ハイトは(下位ハイト書き込みと)同じシステムクロック周期でOCR1x緩衝部またはOCR1xレジ、スタのどちらかに複写されます。

16ビット レジスタ アクセス法のより多くの情報については69頁の「16ビット レジスタのアクセス」を参照してください。

(<mark>訳注</mark>) ここでは比較nxレジスタ全体をOCRnx、OCRnxを構成する緩衝部分をOCRnx緩衝部、実際の比較に使われるレジスタ本体部分をOCRnxレジスタとして記述しています。他の部分での記述でも特に必要がある場合はこの記述方法を適用します。

#### 強制比較出力

非PWM波形生成動作での比較器の一致出力は強制変更(FOC1x)ビットに1を書くことによって強制(変更)できます。比較一致の強制は比較割り込み要求フラケ(OCF1x)の設定(1)やタイマ/カウンタの再設定/解除を行いませんが、OC1xピンは実際の比較一致が起きた場合と同様に更新されます(COM1x1,0ビット設定がOC1xピンの設定(1)、解除(0)、1/0交互のどれかを定義)。

#### TCNT1書き込みによる比較一致妨害

TCNT1への全てのCPU書き込みは、例えタイマ/カウンタが停止されていても、次のタイマ/カウンタクロック周期で起こるどんな比較一致をも妨げます。この特質はタイマ/カウンタクロックが許可されている時に、割り込みを起動することなく、TCNT1と同じ値に初期化されることをOCR1xに許します。

#### 比較一致部の使用

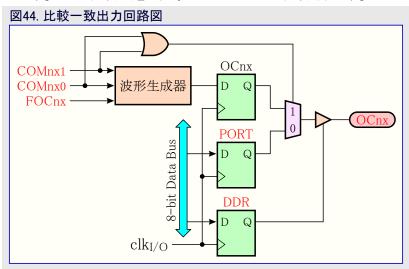
どの動作種別でのTCNT1書き込みでも1タイマ/カウンタ クロック周期間、全ての比較一致を妨げるため、タイマ/カウンタが走行中であるかないかに拘らず、何れかの比較出力部を使う場合、TCNT1を変更する時に危険を伴います。TCNT1に書かれた値がOCR1x値と同じ場合、比較一致は失われ(一致が発生せず)、不正な波形生成に終わります。可変TOP値のPWM動作でTOPに等しいTCNT1を書いてはいけません。(行った場合)TOPに対する比較一致は無視され、カウンタは\$FFFFへ(計数を)続けます。同様にカウンタが下降計数のとき、BOTTOMに等しいTCNT1値を書いてはいけません。

OC1xの初期設定はポート ピンに対するポート方向レジスタを出力に設定する前に行われるべきです。OC1x値を設定する一番簡単な方法は標準動作で強制変更(FOC1x)ストローブ ビットを使うことです。波形生成種別間を変更する時であっても、OC1x(内部)レジスタはその値を保ちます。

比較出力選択(COM1x1,0)ビットが比較値(OCR1x)と共に2重緩衝されないことに気付いてください。COM1x1,0ビットの変更は直ちに有効となります。

## 比較一致出力部

比較出力選択(COM1x1,0)ビットは2つの機能を持ちます。波形生成器は次の比較一致での比較出力(OC1x)状態の定義にCOM1x1,0ピットを使います。次にCOM1x1,0ピットはOC1xピン出力元を制御します。図44.はCOM1x1,0ピット設定によって影響される論理回路の簡単化した図を示します。図のI/Oレジスタ、I/Oピット、I/Oピンは赤文字(訳注:原文は太字)で示されます。COM1x1,0ピットによって影響を及ぼされる標準I/Oポート制御レジスタ(PORTとDDR)の部分だけが示されます。OC1xの状態を参照するとき、その参照はOC1xピンでなく内部OC1xレジスタに対してです。システム リセットが起こると、OC1xレジスタは0にリセットされます。



COM1x1,0ビットのどちらかが設定(1)されると、標準I/Oポート機能は波形生成器からの比較出力(OC1x)によって無効にされます。けれどもOC1xピンの方向(入出力)はポート ピンに対するポート方向レジスタ(DDR)によって未だ制御されます。OC1xピンに対するポート方向レジスタのビット(DDR\_OC1x)はOC1x値がピンで見えるのに先立って出力として設定されなければなりません。このポートの交換機能は一般的に波形生成種別と無関係ですが、いくつかの例外があります。詳細については表55、表56、表57.を参照してください。

比較出力ピン論理回路の設計は出力が許可される前のOC1x状態の初期化を許します。いくつかのCOM1x1,0ビット設定が或る種の動作種別に対して予約されることに注意してください。81頁の「16ビット タイマ/カウンタ1用レジスタ」をご覧ください。

COM1x1.0ビットは捕獲入力部での何の効果もありません。

#### 比較一致出力選択と波形生成

波形生成器は標準、CTC、PWM動作でCOM1x1,0ビットを違うふうに使います。全ての動作種別に対してCOM1x1,0=00設定は次の比較一致で実行すべきOC1xレジスタの動きがないことを波形生成器へ告げます。非PWM動作での比較出力動作については81頁の表55.を参照してください。高速PWM動作については81頁の表56.、位相基準PWMと位相/周波数基準PWMについては81頁の表57.を参照してください。

COM1x1,0ビットの状態変更はこのビットが書かれた後の最初の比較一致で有効になります。非PWM動作について、この動作は強制変更(FOC1x)ストロープビットを使うことによって直ちに効果を得ることを強制できます。





## 動作種別

動作種別、換言するとタイマ/カウンタと比較出力ピンの動作は波形生成種別(WGM13~0)ビットと比較出力選択(COM1x1,0)ビットの組み合わせによって定義されます。比較出力選択ビットは計数順序(動作)に影響を及ぼしませんが、一方波形生成種別ビットは影響を及ぼします。COM1x1,0ビットは生成されたPWM出力が反転されるべきか否か(反転または非反転PWM)のどちらかを制御します。非PWM動作に対してのCOM1x1,0ビットは比較一致で出力が解除(0)、設定(1)、1/0交互のどれにされるべきかを制御します。75頁の「比較一致出力部」をご覧ください。

タイミング情報の詳細については80頁の「タイマ/カウンタ1のタイミング」を参照してください。

#### 標準動作

最も単純な動作種別が標準動作(WGM13~0=0000)です。この動作種別での計数方向は常に上昇(+)で、カウンタの解除は実行されません。カウンタは16ビット最大値(MAX=\$FFFF)を通過すると単に範囲を超え、そして\$0000(BOTTOM)から再び始めます。通常動作でのタイマ/カウンタ溢れ(TOV1)フラケはTCNT1が\$0000になる時と同じタイマ/カウンタ クロック周期で設定(1)されます。この場合のTOV1フラケは設定(1)のみで解除(0)されないことを除いて第17ビットのようになります。けれどもTOV1フラケを自動的に解除(0)するタイマ/カウンタ1溢れ割り込みと組み合わせたタイマ/カウンタの分解能はソフトウェアによって増やせます。標準動作での考慮に特別な場合はなく、新しいカウンタ値は何時でも書けます。

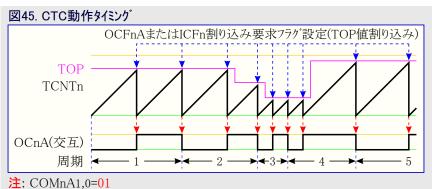
捕獲入力部は標準動作での使用が容易です。けれども外部の事象間の最大間隔がタイマ/カウンタの分解能(16ビット長)を越えてはならないことに気付いてください。事象間の間隔が長すぎる場合、捕獲部に対して分解能を拡張するために、タイマ/カウンタ溢れ割り込みまたは前置分周器が使われなければなりません。

比較出力部は与えられた或る時間に割り込みを生成するのに使えます。標準動作で波形を生成するのに比較出力を使うのは、それが大変多くのCPU時間を占有するため推奨されません。

#### 比較一致タイマ/カウンタ解除(CTC)動作

比較一致タイマ/カウンタ解除(CTC)動作(WGM13~0=0100または1100)ではOCR1AまたはICR1がカウンタの分解能を操作するのに使われます。CTC動作ではカウンタ(TCNT1)値がOCR1A(WGM13~0=4)またはICR1(WGM13~0=12)のどちらかと一致する時にカウンタが\$0000に解除されます。OCR1AまたはICR1はカウンタに対するTOP値、従って分解能も定義します。この動作種別はより大きい比較一致出力周波数の制御を許します。それは外部の出来事の計数操作も簡単にします。

CTC動作についてのタイミング図は**図45**.で示されます。カウンタ(TCNT1)値はOCR1AまたはICR1のどちらかで比較一致が起こるまで増加し、そしてその後にカウンタ(TCNT1)は解除(\$0000)されます。



TOP値を定義するのに使われるレジスタに対してOCF1AまたはICF1のどちらかを使うことにより、カウンタ値がTOP値に到達する時毎に割り込みが生成できます。割り込みが許可されるなら、割り込み処理ルーチンはTOP値を更新するのに使えます。けれども前置分周なしまたは低い前置分周値でカウンタが走行している時にBOTTOMと近い値にTOPを変更するのは、CTC動作が2重緩衝機能を持たないために注意して行わなければなりません。OCR1AまたはICR1に書かれた新しい値がTCNT1の現在値よりも低い(小さい)場合、カウンタは(その回の)比較一致を失います。その後のカウンタは比較一致が起こせるのに先立って、最大値(\$FFFF)へそして次に\$0000から始める計数をしなければならないでしょう。多くの場合でこの特性は好ましくありません。OCR1Aが2重緩衝されるので、代替はTOPを定義するのにOCR1Aを用いる高速PWM動作(WGM13~0=1111)を使うことでしょう。

CTC動作で波形出力を生成するため、OC1A出力は比較出力選択(COM1A1,0)ビットを交互動作(=01)に設定することによって各比較一致での論理レヘル交互切替に設定できます。OC1A値はそのピンに対するデータ方向が出力(DDR\_OC1A=1)に設定されない限り、ポートピンで見えないでしょう。生成された波形はOCR1Aが0(\$0000)に設定される時に $f_{OC1A}=f_{clk\_I/O}/2$ の最大周波数を得ます。生成波形周波数は次式によって定義されます。

$$f_{\text{OCnA}} = \frac{f_{\text{clk\_I/O}}}{2 \times \text{N} \times (1 + \text{OCRnA})}$$

変数Nは前置分周数(1,8,64,256,1024)を表します。

標準動作と同じように、タイマ/カウンタ溢れ割り込み要求(TOV1)フラグはカウンタがMAXから\$0000へ計数するのと同じタイマ/カウンタ クロック周期で設定(1)されます。

#### 高速PWM動作

高速パルス幅変調(PWM)動作(WGM13~0=0101,0110,0111,1110,1111)は高周波数PWM波形生成選択を提供します。高速PWMはそれが単一傾斜(鋸波)動作であることによって他のPWM動作と異なります。カウンタはBOTTOMからTOPまで計数し、その後BOTTOMから再び始めます。非反転比較出力動作(COM1x1,0=10)での比較出力(OC1x)はTCNT1とOCR1x間の比較一致で解除(0)され、BOTTOMで設定(1)されます。反転出力動作(COM1x1,0=11)の出力は比較一致で設定(1)され、BOTTOMで解除(0)されます。単一傾斜動作のため、高速PWM動作の動作周波数は両傾斜(三角波)動作を使う位相基準や位相/周波数基準PWM動作よりも2倍高くできます。この高い周波数は電力調節、整流、D/A変換に対して高速PWM動作を都合よく適合させます。高い周波数は物理的に小さな外部部品(コイルやコンデンサ)を許し、従ってシステム総費用を削減します。

高速PWMのPWM分解能は8,9,10ビットに固定、若しくはOCR1AかICR1のどちらかによって定義できます。許された最小分解能は2 ビット(OCR1AまたはICR1が\$0003設定)、最大分解能は16ビット(OCR1AまたはICR1がMAX設定)です。ビットでのPWM分解能は次式を使うことによって計算できます。

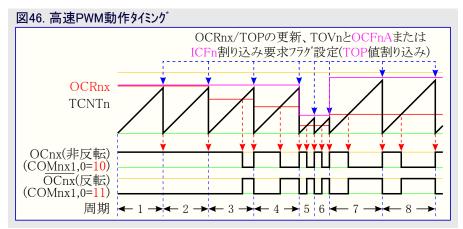
高速PWM動作でのカウンタはカウンタ値が固定値\$00FF,\$01FF,\$03FF(WGM13 $\sim$ 0=0101,0110,0111)、ICR1 値(WGM13 $\sim$ 0=1110)またはOCR1A値(WGM13 $\sim$ 0=1111)のどれかと一致するまで増加されます。そしてカウンタは(一致の)次のタイマ/カウンタ クロック周期で解除(\$0000)されます。高速PWM動作のタイミング図は**図46**.

 $R_{\text{FPWM}} = \frac{\log (\text{TOP} + 1)}{\log 2}$ 

で示されます。本図はOCR1AかICR1がTOPを定義するのに使われる時の高速PWM動作を示します。TCNT1値はタイング図で単一傾斜動作(鋸波)を表す折れ線グラフとして示されます。この図は非反転と反転のPWM出力を含みます。細い赤線はOCR1x値を示し、TCNT1値との交点(接点)がTCNT1とOCR1x間の比較一致を示します(訳注:図補正に伴い本行若干変更)。比較割り込み要求フラグ(OCF1x)は比較一致が起こると設定(1)されます。

タイマ/カウンタ溢れ割り込み要求(TOV1)フラケ はカウンタがTOPに到達する時毎に設定(1)されます。加えて、OCR1AかICR1のどちらかがTOP値を定義するのに使われると、OCF1AまたはICF1割り込み要求フラケ はTOV1が設定(1)されるのと同じタイマ/カウンタクロック周期で設定(1)されます。これらの割り込みの1つが許可されるなら、その割り込み処理ルーチンはTOPと比較値を更新するのに使えます。

TOP値を変更するとき、プログラムは新TOP値が全ての比較レジスタ値と等しいか大きいことを保証しなければなりません。TOP値が何れかの比較レジスタよりも小さな場合、TCNT1とそのOCR1x間で比較一致は決して起きません。固定TOP値を使



う場合、どのOCR1xが書かれる時も、未使用ビットが0で隠(に置換)されることに注意してください。

ICR1がTOP値を定義するのに使われるとき、ICR1を更新する手順はOCR1Aの更新と異なります。ICR1は2重緩衝されません。これは前置分周なしまたは低い前置分周値でカウンタが走行している時にICR1が小さな値に変更される場合、書かれた新しいICR1値がTCNT1の現在値よりも小さくなる危険を意味します。その後の結果はカウンタが(その回の)TOP値での比較一致を失うことです。その後のカウンタは比較一致が起こせるのに先立って、MAX値(\$FFFF)へそして次に\$0000から始める計数をしなければならないでしょう。しかし、OCR1Aは2重緩衝されます。この特徴は何時でも書かれることをOCR1AのI/O位置に許します。OCR1A I/O位置が書かれると、書かれた値はOCR1A緩衝部に置かれます。OCR1A(比較)レジスタはその後にTCNT1がTOPと一致した次のタイマ/カウンタクロック周期にOCR1A緩衝部の値で更新されます。この更新はTCNT1の解除(\$0000)やTOV1の設定(1)と同じタイマ/カウンタクロック周期で行われます。

TOPを定義するのにICR1を使うことは決まったTOP値を使う時に上手くいきます。ICR1を使うことにより、OC1AでのPWM出力を生成するためにOCR1Aが自由に使えます。けれども基準PWM周波数が(TOP値を変更することによって)動的に変更される場合、OCR1Aが2重緩衝機能のため、TOPとしてOCR1Aを使うことは明らかに良い選択です。

高速PWM動作での比較部はOC1xピンでのPWM波形の生成を許します。COM1x1,0ビットを'10'に設定することは非反転PWM出力を作成し、反転PWM出力はCOM1x1,0を'11'に設定することで生成できます。81頁の表56.をご覧ください。実際のOC1x値はそのポートピンに対するデータ方向が出力(DDR\_OC1x=1)として設定される場合にだけ見えるでしょう。PWM波形はTCNT1とOCR1x間の比較一致でOC1x(内部)レジ、スタを設定(1)(または解除(0))と、カウンタが解除(\$0000、TOPからBOTTOMへ変更)されるタイマ/カウンタクロック周期でのOC1xレジ、スタを解除(0)(または設定(1))することによって生成されます。

PWM出力周波数は次式によって計算できます。変数Nは前置分周数(1,8,64,256,1024)を表します。 OCR1xの両端値は高速PWM動作でPWM波形出力を生成する時の特別な場合にあたります。OCR

 $f_{\text{OCnxPWM}} = \frac{f_{\text{clk\_I/O}}}{N \times (1 + \text{TOP})}$ 

1xがBOTTOM(\$0000)に等しく設定されると、出力はTOP+1 タイマ/カケンタ クロック周期毎の狭いスパーイク(パルス)になるでしょう。TOPに等しいOCR1x設定は(COM1x1,0ビットによって設定される出力極性に依存して)定常的なLowまたはHigh出力に終わるでしょう。

高速PWM動作で(デューティ比50%)周波数の波形出力は比較一致毎に論理反転するOC1A設定(COM1A1,0=01)によって達成できます。これはTOP値を定義するのにOCR1Aが使われる(WGM13 $\sim$ 0=1111)の場合にだけ適用されます。生成された波形はOCR1Aが0 (\$0000)に設定される時に $f_{\rm OC1A}=f_{\rm clk\_I/O}/2$ の最大周波数でしょう。この特性は高速PWM動作で比較出力部の2重緩衝機能が許可されることを除いて、CTC動作でのOC1A交互出力(COM1A1,0=01)と同じです。





## 位相基準PWM動作

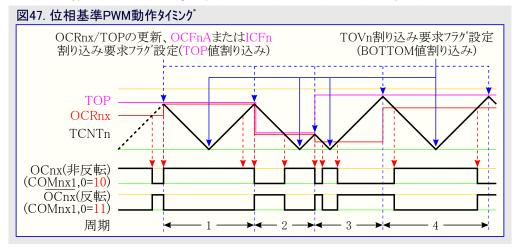
位相基準 $\uppsi^n$  $\uppsi$   $\uppsi$ 

位相基準PWM動作のPWM分解能は8,9,10ビットに固定、若しくはOCR1AかICR1のどちらかによって定義できます。許された最小分解能は2ビット(OCR1AまたはICR1が\$0003設定)、最大分解能は16ビット(OCR1AまたはICR1がMAX設定)です。ビットでのPWM分解能は次式を使うことによって計算できます。

$$R_{\text{PCPWM}} = \frac{\log (\text{TOP} + 1)}{\log 2}$$

位相基準PWM動作でのカウンタはカウンタ値が固定値\$00FF,\$01FF,\$03FF(WGM13~0=0001,0010,0011)、ICR1値(WGM13~0=1010)またはOCR1A値(WGM13~0=1011)のどれかと一致するまで増加されます。カウンタはTOPに到達したその時に計数方向を変更します。このTCNT1値は1タイマ/カウンタクロッ

ク周期間TOPと等しくなります。位相基準PWM動作のタイング図は**図47**.で示されます。本図はOCR1AかICR1がTOPを定義するのに使われる時の位相基準PWM動作を示します。TCNT1値はタイング図で両傾斜動作(三角波)を表す折れ線グラフとして示されます。この図は非反転と反転のPWM出力を含みます。細い赤線はOCR1x値を示し、TCNT1値との交点(接点)がTCNT1とOCR1x間の比較一致を示します(訳注:図補正に伴い本行若干変更)。比較割り込み要求フラグ(OCF1x)は比較一致が起こると設定(1)されます。



タイマ/カウンタ溢れ(TOV1)フラグ はカウンタがBOTTOMに到達する時毎に設定(1)されます。OCR1AかICR1のどちらかがTOP値を定義するのに使われるとき、OCF1AまたはICF1割り込み要求フラグ はOCR1xレジ スタが(TOPに於いて)2重緩衝値で更新されるのと同じタイマ/カウンタ クロック周期によって設定(1)されます。これらの割り込み要求フラグ はカウンタがTOPまたはBOTTOM値に到達する毎に割り込みを発生するのに使えます。

TOP値を変更するとき、プログラムは新TOP値が全ての比較レジスタ値と等しいか大きいことを保証しなければなりません。TOP値が何れかの比較レジスタよりも小さな場合、TCNT1とそのOCR1x間で比較一致は決して起きません。固定TOP値を使う場合、どのOCR1xが書かれる時も、未使用ビットがOで隠(に置換)されることに注意してください。図47.で示される第3周期が図解するようにタイマ/カウンタが位相基準PWM動作で走行中にTOPを積極的に変更するのは、非対称出力で終わることが有り得ます。これに対する理由はOCR1xレジスタの更新時に見出せます。OCR1x更新はTOPで起きるので、PWM周期はTOPで始まりそして終わります。これは下降傾斜長が直前のTOP値によって決定され、一方上昇傾斜長は新しいTOP値で決定されることを意味します。これら2つの値(TOP)が違う時にその周期の2つの傾斜長は異なるでしょう。この長さの相違が出力での非対称な結果を生じます。

タイマ/カウンタが走行中にTOP値を変更する場合、位相基準PWM動作の代わりに位相/周波数基準PWM動作を使うことが推奨されます。一定のTOP値を使う時に2つの動作種別間に現実的な違いはありません。

位相基準PWM動作での比較部はOC1xピンでのPWM波形の生成を許します。COM1x1,0ビットを'10'に設定することは非反転PWM出力を作成し、反転PWM出力はCOM1x1,0を'11'に設定することで生成できます(81頁の表57.をご覧ください)。実際のOC1x値はそのポート ピンに対するデータ方向が出力(DDR\_OC1x=1)として設定される場合にだけ見えるでしょう。PWM波形はカウンタが増加する時のTCNT1とOCR1x間の比較一致でOC1x(内部)レジスタを設定(1)(または解除(0))と、カウンタが減少する時のTCNT1とOCR1x間の比較一致でOC1xレジスタを解除(0)(または設定(1))することによって生成されます。

位相基準PWMを使う時の出力に対するPWM周波数は次式によって計算できます。変数Nは前置分周数(1,8,64,256,1024)を表します。

 $f_{\text{OCnxPCPWM}} = \frac{f_{\text{clk\_I/O}}}{2 \times N \times \text{TOP}}$ 

OCR1xの両端値は位相基準PWM動作でPWM波形出力を生成する時の特別な場合にあたります。非反転PWM動作ではOCR1xがBOTTOMに等しく設定されると出力は定常的なLow、TOPに等しく設定されると定常的なHighになるでしょう。反転PWMに対する出力は逆の論理値になります。TOP値定義にOCR1Aが使われ(WGM13~0=1011)、COM1A1,0=01なら、OC1A出力はデューティ比50%で交互に変化します。

#### 位相/周波数基準PWM動作

位相/周波数基準パルス幅変調(PWM)動作(WGM13~0=1000,1001)は高分解能で正しい位相と周波数のPWM波形生成選択を提供します。位相/周波数基準PWM動作は両傾斜(三角波)動作を基準とした位相基準PWMと似ています。かかりはBOTTOM(\$0000)からTOPへ、そしてその後にTOPからBOTTOMへを繰り返し計数します。非反転比較出力動作(COM1x1,0=10)での比較出力(OC1x)は上昇計数中のTCNT1とOCR1x間の比較一致で解除(0)され、下降計数中の比較一致で設定(1)されます。反転出力動作(COM1x1,0=11)での動作は逆にされます。両傾斜(三角波)動作は単一傾斜(鋸波)動作よりも低い最大動作周波数になります。けれども両傾斜(三角波)PWM動作の対称特性のため、これらの動作種別は電動機制御の応用に好まれます。

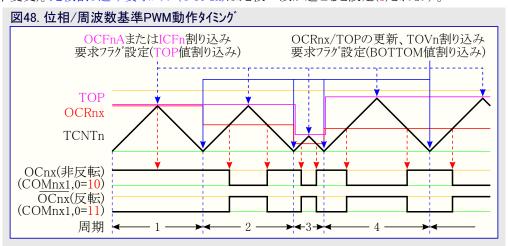
位相基準と位相/周波数基準PWM動作間の主な違いはOCR1xレジスタがOCR1x緩衝部によって更新される時(<mark>訳補</mark>:TOPとBOTTO M)です(図47.と図48.参照)。

位相/周波数基準PWM動作のPWM分解能はOCR1AかICR1のどちらかで定義できます。許された最小分解能は2ビット(OCR1AまたはICR1が\$0003設定)、最大分解能は16ビット(OCR1AまたはICR1がMAX設定)です。ビットでのPWM分解能は次式を使うことによって計算できます。

$$R_{\rm PFCPWM} = \frac{\log (\text{TOP} + 1)}{\log 2}$$

位相/周波数基準PWM動作でのタイマ/カウンタはタイマ/カウンタ値がICR1値(WGM13~0=1000)かOCR1A値(WGM13~0=1001)のどちらかと一致するまで増加されます。カウンタはTOPに到達したその時に計数方向を変更します。このTCNT1値は1タイマ/カウンタクロック周期間、TOPと等しくなります。位相/周波数

基準PWM動作のタイミング図は**図48**.で示されます。この図はOCR1AかICR1がTOPを定義するのに使われる時の位相/周波数基準PWM動作を示します。TCNT1値はタイミング図で両傾斜動作(三角波)を表す折れ線グラフとして示されます。この図は非反転と反転のPWM出力を含みます。細い赤線はOCR1x値を示し、TCNT1値との交点(接点)がTCNT1とOCR1x間の比較一致を示します(訳注:図補正に伴い本行若干変更)。比較割り込み要求フラグ(OCF1x)は比較一致が起こると設定(1)されます。



タイマ/カウンタ溢れ(TOV1)フラグはOCR1xレシ、スタが(BOTTOMに於いて)2重緩衝値で更新されるのと同じタイマ/カウンタ クロック周期で設定(1)されます。OCR1AかICR1のどちらかがTOP値を定義するのに使われるとき、OCF1AまたはICF1割り込み要求フラグはタイマ/カウンタがTOPに到達する時毎に設定(1)されます。これらの割り込み要求フラグはカウンタがTOPまたはBOTTOM値に到達する毎に割り込みを発生するのに使えます。

TOP値を変更するとき、プログラムは新TOP値が全ての比較レン、スタ値と等しいか大きいことを保証しなければなりません。TOP値が何れかの比較レシ、スタよりも小さな場合、TCNT1とそのOCR1x間で比較一致は決して起きません。

図48.が示すように、生成された出力は位相基準PWM動作と異なり、全ての周期で対称です。OCR1xレジスタがBOTTOMで更新されるため、上昇と下降の傾斜長は常に等しくなります。これが対称出力パルス、従って正しい周波数を与えます。

TOPを定義するのにICR1を使うことは決まったTOP値を使う時に上手くいきます。ICR1を使うことにより、OC1AでのPWM出力を生成するためにOCR1Aが自由に使えます。けれども基準PWM周波数が(TOP値を変更することによって)動的に変更される場合、OCR1Aが2重緩衝機能のため、TOPとしてOCR1Aを使うことは明らかに良い選択です。

位相/周波数基準PWM動作での比較部はOC1xピンでのPWM波形の生成を許します。COM1x1,0ピットを'10'に設定することは非反転PWM出力を作成し、反転PWM出力はCOM1x1,0を'11'に設定することで生成できます(81頁の表57.をご覧ください)。実際のOC1x値はそのポート ピンに対するデータ方向が出力(DDR\_OC1x=1)として設定される場合にだけ見えるでしょう。PWM波形はカウンタが増加する時のTCNT1とOCR1x間の比較一致でOC1x(内部)レジスタを設定(1)(または解除(0))と、カウンタが減少する時のTCNT1とOCR1 x間の比較一致でOC1xレジスタを解除(0)(または設定(1))することによって生成されます。

位相/周波数基準PWMを使う時の出力に対するPWM周波数は次式によって計算できます。変数 Nは前置分周数(1,8,64,256,1024)を表します。

 $f_{\text{OCnxPFCPWM}} = \frac{f_{\text{clk\_I/O}}}{2 \times N \times \text{TOP}}$ 

OCR1xの両端値は位相/周波数基準PWM動作でPWM波形出力を生成する時の特別な場合にあたります。非反転PWM動作ではOCR1xがBOTTOMに等しく設定されると出力は定常的なLow、TOPに等しく設定されると定常的なHighになるでしょう。反転PWMに対する出力は逆の論理値になります。TOP値定義にOCR1Aが使われ(WGM13 $\sim$ 0 =1001)、COM1A1,0=01なら、OC1A出力はデューティ比50%で交互に変化します。





## タイマ/カウンタのタイミング

このタイマ/カウンタは同期設計で、従ってタイマ/カウンタ クロック(clkT1)が下図のクロック許可信号として示されます。この図は割り込みフラグが設定(1)される時、そしてOCR1xレジ、スタがOCR1x緩衝値で更新される時(2重緩衝を使う動作種別のみ)の情報を含みます。図49.はOCF1xの設定についてのタイシング図を示します。

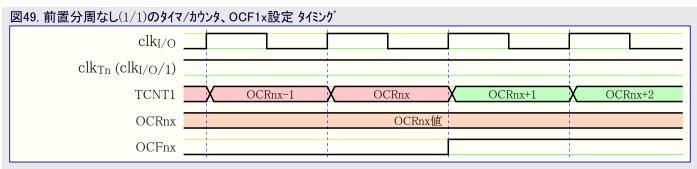


図50.は同じタイミングデータを示しますが、前置分周器が許可されています。

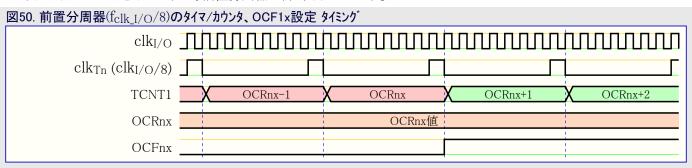


図51.は各動作種別でのTOP近辺の計数手順を示します。位相/周波数基準PWM動作使用時のOCR1xレジスタはBOTTOMで更新されます。タイシング図は同じになりますが、当然TOPはBOTTOMで、TOP-1はBOTTOM+1でなどのように置き換えられます。BOTTOMでTOV1を設定(1)する動作種別についても、同様な名称変更が適用されます。

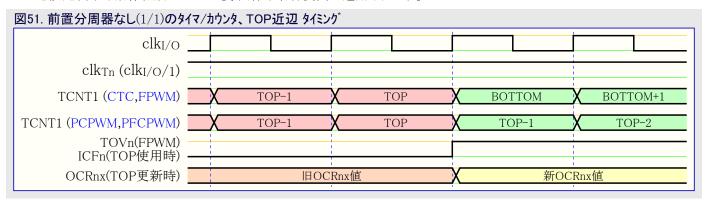
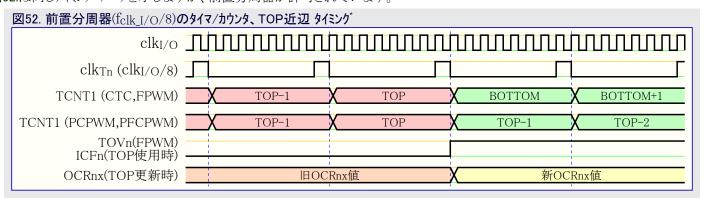


図52.は同じタイミングデータを示しますが、前置分周器が許可されています。



## 16ビット タイマ/カウンタ1 用レジスタ

タイマ/カウンタ1制御レシ、スタA (Timer/Counter1 Control Register A) TCCR1A

ピット	7	6	5	4	3	2	1	0	
(\$80)	COM1A1	COM1A0	COM1B1	COM1B0	-	-	WGM11	WGM10	TCCR1A
Read/Write	R/W	R/W	R/W	R/W	R	R	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

- ビット7,6 COM1A1,0:比較1A出力選択 (Compare Output Mode1A bit 1 and 0)
- ビット5,4 COM1B1,0: 比較1B出力選択 (Compare Output Mode1B bit 1 and 0)

COM1A1,0とCOM1B1,0は各々OC1AとOC1B比較出力ピンの動作を制御します。COM1A1,0ビットの1つまたは両方が1を書かれると、OC1A出力はそのI/Oピンの通常ポート機能を無効にし、そのI/Oピンに接続されます。COM1B1,0ビットの1つまたは両方が1を書かれると、OC1B出力はそのI/Oピンの通常ポート機能を無効にし、そのI/Oピンに接続されます。けれども出力駆動部を許可するため、OC1AまたはOC1Bピンに対応するポート方向レジスタ(DDR)のビットが設定(1)されなければならないことに注意してください。

OC1AまたはOC1Bがピンに接続されるとき、COM1x1,0ビットの機能はWGM13~0ビット設定に依存します。**表55**.はWGM13~0ビットが標準動作またはCTC動作(つまり非PWM)に設定される時のCOM1x1,0ビット機能を示します。

#### 表55. 非PWM動作での比較出力選択(注: xはAまたはB)

COM1x1	COM1x0	意味
0	0	標準ポート動作(OC1x切断)
0	1	比較一致でOC1xピン トグル(交互)出力
1	0	比較一致でOC1xピン Lowレベル出力
1	1	比較一致でOC1xピン Highレベル出力

表56.はWGM13~0ビットが高速PWM動作に設定される時のCOM1x1,0ビット機能を示します。

# 表56. 高速PWM動作での比較出力選択 (注: xはAまたはB, Xは0または1)

COM1x1	COM1x0	意味
0	0	標準ポート動作(OC1x切断)
0	1	WGM13~0=111X : 比較一致でOC1Aピン トグル(交互)出力、OC1Bは標準ポート動作(OC1B切断) WGM13~0上記以外:標準ポート動作(OC1x切断)
1	0	比較一致でLow、BOTTOMでHighをOC1xピンへ出力(非反転動作)
1	1	比較一致でHigh、BOTTOMでLowをOC1xt°ンへ出力(反転動作)

注: COM1x1が設定(1)され、OCR1xがTOPと等しい時に特別な状態が起きます。この状態での比較一致は無視されますが、BO TTOMでの設定(1)または解除(0)は実行されます。より多くの詳細については77頁の「高速PWM動作」をご覧ください。

表57.はWGM13~0ビットが位相基準または位相/周波数基準PWM動作に設定される時のCOM1x1,0ビット機能を示します。

## 表57. 位相基準または位相/周波数基準PWM動作での比較出力選択(注: xはAまたはB, Xは0または1)

COM1x1	COM1x0	意味
0	0	標準ポート動作(OC1x切断)
0	1	WGM13~0=10X1 : 比較一致でOC1Aピン トグル(交互)出力、OC1Bは標準ポート動作(OC1B切断) WGM13~0上記以外:標準ポート動作(OC1x切断)
1	0	上昇計数時の比較一致でLow、下降計数時の比較一致でHighをOC1xピンへ出力
1	1	上昇計数時の比較一致でHigh、下降計数時の比較一致でLowをOC1xピンへ出力

注: COM1x1が設定(1)され、OCR1xがTOPと等しい時に特別な状態が起きます。より多くの詳細については78頁の「位相基準PWM動作」をご覧ください。





#### • ビット1,0 - WGM11,0: 波形生成種別 (Waveform Generation Mode bit 1 and 0)

タイマ/カウンタ制御レジスタB(TCCR1B)で得られるWGM13,2ビットと組み合わせたこれらのビットはカウンタの計数順序(方向)、最大計数(TOP) 値供給元、使われるべき波形生成のどの形式かを制御します(表58.参照)。タイマ/カウンタ部によって支援される動作種別は標準動作(カウンタ)、比較一致タイマ/カウンタ解除(CTC)動作と3形式のハプルス幅変調(PWM)動作です。76頁の「動作種別」をご覧ください。

#### 表58. 波形生成種別選択

番号	WGM13	WGM12 (CTC1)	WGM11 (PWM11)	WGM10 (PWM10)	タイマ/カウンタ動作種別	TOP値	OCR1x 更新時	TOV1 設定時
0	0	0	0	0	標準動作	\$FFFF	即値	MAX
1	0	0	0	1	8ビット位相基準PWM動作	\$00FF	TOP	BOTTOM
2	0	0	1	0	9ビット位相基準PWM動作	\$01FF	TOP	BOTTOM
3	0	0	1	1	10ビット位相基準PWM動作	\$03FF	TOP	BOTTOM
4	0	1	0	0	比較一致タイマ/カウンタ解除(CTC)動作	OCR1A	即値	MAX
5	0	1	0	1	8ビット高速PWM動作	\$00FF	BOTTOM	TOP
6	0	1	1	0	9ビット高速PWM動作	\$01FF	BOTTOM	TOP
7	0	1	1	1	10ビット高速PWM動作	\$03FF	BOTTOM	TOP
8	1	0	0	0	位相/周波数基準PWM動作	ICR1	BOTTOM	BOTTOM
9	1	0	0	1	位相/周波数基準PWM動作	OCR1A	BOTTOM	BOTTOM
10	1	0	1	0	位相基準PWM動作	ICR1	TOP	BOTTOM
11	1	0	1	1	位相基準PWM動作	OCR1A	TOP	BOTTOM
12	1	1	0	0	比較一致タイマ/カウンタ解除(CTC)動作	ICR1	即値	MAX
13	1	1	0	1	(予約)	-	-	-
14	1	1	1	0	高速PWM動作	ICR1	ВОТТОМ	TOP
15	1	1	1	1	高速PWM動作	OCR1A	BOTTOM	TOP

注: CTC1とPWM11,0ビット定義名は旧名です。WGM12~0定義を使ってください。しかし、これらのビットの機能と位置は旧版のタイマ/カウンタと一致します。

#### タイマ/カウンタ1制御レシ、スタB (Timer/Counter1 Control Register B) TCCR1B

ピット	7	6	5	4	3	2	1	0	
(\$81)	ICNC1	ICES1	-	WGM13	WGM12	CS12	CS11	CS10	TCCR1B
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

#### • ビット7 - ICNC1: 捕獲起動入力1雑音消去許可 (Input Capture1 Noise Canceler)

このビットを(1に)設定することが捕獲起動入力雑音消去器を活性(有効)にします。雑音消去器が有効にされると、捕獲起動入力 (ICP1)ピンからの入力が濾波されます。この濾波器機能はそれが出力を更新するのに、連続4回等しく評価されたICP1ピンの採取を必要とします。雑音消去器が許可されると、捕獲入力はそれによって4発振器(システム クロック)周期遅らされます。

## • ビット6 - ICES1: 捕獲起動入力端選択 (Input Capture1 Edge Select)

このビットは出来事での捕獲を起動するのに使われる捕獲起動入力(ICP1)ピンのどちらかの端(エッジ)を選びます。ICES1ビットが0を書かれると起動動作として下降(負)端が使われ、ICES1ビットが1を書かれると上昇(正)端が捕獲を起動します。

捕獲がICES1設定に従って起動されると、カウンタ値が捕獲レシ、スタ(ICR1)に複写されます。この出来事は捕獲入力割り込み要求フラク (ICF1)も設定(1)し、そしてこれは、この割り込みが許可されていれば捕獲入力割り込みを起こすのに使えます。

ICR1がTOP値として使われると(TCCR1AとTCCR1Bに配置された $WGM13\sim0$ じットの記述をご覧ください)、ICP1が切り離され、従って捕獲入力機能は禁止されます。

#### • ビット5 - Res: 予約 (Reserved)

このビットは将来の使用に対して予約されています。将来のデバイスとの共通性を保つため、TCCR1Bが書かれるとき、このビットは<mark>0</mark>を書かれなければなりません。

## • ビット4,3 - WGM13,2: 波形生成種別 (Waveform Generation Mode bit 3 and 2)

TCCR1AのWGM11,0ビット記述をご覧ください。

• ビット2~0 - CS12~0: クロック選択1 (Clock Select1, bit 2,1 and 0)

この3つのクロック選択ビットはタイマ/カウンタ(TCNT1)によって使われるべきクロック元を選びます。図49.と図50.をご覧ください。

表50	カイフ	/ 十 占 い カ 1	7	カカロ	ック選択
イマ リガ.	אור כי	/ ハワンショ	$\sim$		コンフロチャル

CS12	CS11	CS10	意味
0	0	0	停止(タイマ/カウンタ1動作停止)
0	0	1	clk <sub>I/O</sub> (前置分周なし)
0	1	0	clk <sub>I/O</sub> /8 (8分周)
0	1	1	clk <sub>I/O</sub> /64 (64分周)
1	0	0	clk <sub>I/O</sub> /256 (256分周)
1	0	1	clk <sub>I/O</sub> /1024 (1024分周)
1	1	0	T1ピンの下降端 (外部クロック)
1	1	1	T1ピンの上昇端 (外部クロック)

タイマ/カウンタ1に対して外部ピン(クロック)動作が使われる場合、例えT1ピンが出力として設定されても、T1ピンの遷移はカウンタをクロック駆動します。この特性はソフトウェアに計数制御を許します。

## タイマ/カウンタ1制御レジスタC (Timer/Counter1 Control Register C) TCCR1C

ピット (\$82)	7 FOC1A	6 FOC1B	5 -	4	3	2	1	0 -	TCCR1C
Read/Write	W	W	R	R	R	R	R	R	
初期値	0	0	0	0	0	0	0	0	

- ビットフ FOC1A: OC1A強制変更 (Force Output Compare 1A)
- ビット6 FOC1B: OC1B強制変更 (Force Output Compare 1B)

FOC1A/FOC1BビットはWGM13~0ビットが非PWM動作を指示する時だけ有効です。けれども将来のデバイスとの共通性を保証するため、PWM動作で扱う時にTCCR1Cが書かれる場合、これらのビットは0に設定されなければなりません。FOC1A/FOC1Bビットに論理1を書くと波形生成部で直ちに比較一致が強制されます。OC1x出力はCOM1x1,0ビット設定に従って変更されます。FOC1A/FOC1Bビットがストローブとして実行されることに注意してください。それによって強制された比較の効果を決めるのはCOM1x1,0ビットに存在する値です。

FOC1A/FOC1Bストロープは何れの割り込みの生成もTOPとしてOCR1Aを使う比較一致タイマ解除(CTC)動作でのタイマ/カウンタの解除(\$0000)も行いません。

FOC1A/FOC1Bビットは常に0として読みます。

## タイマ/カウンタ1 (Timer/Counter1) TCNT1H,TCNT1L (TCNT1)

ピット	15	14	13	12	11	10	9	8	_
(\$85)	(MSB)								TCNT1H
ead/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	
ピット	7	6	5	4	3	2	1	0	
(\$84)								(LSB)	TCNT1L
ead/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

この2つのタイマ/カウンタ I/O位置(TCNT1HとTCNT1Lを合わせたTCNT1)は、読み書き両方についてタイマ/カウンタ部の16ビット カウンタに直接アクセスします。CPUがこれらのレシ、スタをアクセスする時に上位と下位の両ハ・イトが同時に読み書きされるのを保証するため、このアクセスは8ビット上位ハ・イトー時レジ、スタ(TEMP)を使って実行されます。この一時レジ、スタは他の全ての16ビット レジ、スタによって共用されます。69頁の「16ビット レジ、スタのアクセス」をご覧ください。

カウンタが走行中にカウンタ(TCNT1)を変更することはOCR1xの1つとTCNT1間の比較一致消失の危険を誘発します。

TCNT1への書き込みは全ての比較部に対して次のタイマ/カウンタ クロックでの比較一致を妨害(除去)します。





## タイマ/カウンタ1 比較Aレジスタ (Timer/Counter1 Output Compare Register A) OCR1AH,OCR1AL (OCR1A)

ピット	15	14	13	12	11	10	9	8	_
(\$89)	(MSB)								OCR1AH
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	
ピット	7	6	5	4	3	2	1	0	_
(\$88)								(LSB)	OCR1AL
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

## タイマ/カウンタ1 比較Bレジスタ (Timer/Counter1 Output Compare Register B) OCR1BH,OCR1BL (OCR1B)

ピット (\$8B)	15 (MSB)	14	13	12	11	10	9	8	OCR1BH
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	
ピット (\$8A)	7	6	5	4	3	2	1	0 (LSB)	OCR1BL
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

この比較レシ、スタは継続的にカウンタ(TCNT1)値と比較される16ビット値を含みます。一致は比較一致割り込みやOC1xピンでの波形出力を生成するのに使えます。

この比較レシ、スタは容量が16ビットです。CPUがこれらのレシ、スタへ書く時に上位と下位の両バイが同時に書かれるのを保証するため、このアクセスは8ビット上位バイトー時レシ、スタ(TEMP)を使って実行されます。この一時レシ、スタは他の全ての16ビット レシ、スタによって共用されます。69頁の「16ビット レジ、スタのアクセス」をご覧ください。

## タイマ/カウンタ1 捕獲レジスタ (Timer/Counter1 Input Capture Register) ICR1H,ICR1L (ICR1)

ピット (\$87)	15 (MSB)	14	13	12	11	10	9	8	] ICR1H
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	
ピット	7	6	5	4	3	2	1	0	_
(\$86)								(LSB)	ICR1L
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

この捕獲レシ、スタはICP1ピン(またはタイマ/カウンタ1については任意のアナログ比較器出力)で出来事が起こる毎にカウンタ(TCNT1)値で更新されます。この捕獲レシ、スタはタイマ/カウンタのTOP値を定義するのに使えます。

この捕獲レジスタは容量が16ビットです。CPUがこれらのレジスタをアクセスする時に上位と下位の両バイが同時に読まれるのを保証するため、このアクセスは8ビット上位バイト一時レジスタ(TEMP)を使って実行されます。この一時レジスタは他の全ての16ビット レジスタによって共用されます。69頁の「16ビット レジスタのアクセス」をご覧ください。

## タイマ/カウンタ1割り込み許可レジスタ (Timer/Counter1 Interrupt Mask Register) TIMSK1

ピット	7	6	5	4	3	2	1	0	
(\$6F)	_	-	ICIE1	-	_	OCIE1B	OCIE1A	TOIE1	TIMSK1
Read/Write	R	R	R/W	R	R	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

• ビット5 - ICIE1: タイマ/カウンタ1捕獲割り込み許可 (Timer/Counterl Input Capture Interrupt Enable)

このビットが1を書かれて、ステータス レジスタ(SREG)の全割り込み許可(I)ビットが設定(1)されると、タイマ/カウンタ1捕獲割り込みが許可されます。タイマ/カウンタ1割り込み要求フラグレジスタ(TIFR1)に配置された捕獲割り込み要求フラグ(ICF1)が設定(1)されると、対応する割り込み ^゙クタ(30頁の「割り込み」参照)が実行されます。

• ビット2 - OCIE1B: タイマ/カウンタ1比較B割り込み許可 (Timer/Counterl Output Compare B Match Interrupt Enable)

このビットが1を書かれて、ステータス レジスタ(SREG)の全割り込み許可(I)ビットが設定(1)されると、タイマ/カウンタ1比較B一致割り込みが許可されます。タイマ/カウンタ1割り込み要求フラグレジスタ(TIFR1)に配置された比較1B割り込み要求フラグ(OCF1B)が設定(1)されると、対応する割り込みベクタ(30頁の「割り込み」参照)が実行されます。

• ビット1 - OCIE1A: タイマ/カウンタ1比較A割り込み許可 (Timer/Counterl Output Compare A Match Interrupt Enable)

このビットが1を書かれて、ステーータス レジスタ(SREG)の全割り込み許可(I)ビットが設定(1)されると、タイマ/カウンタ1比較A一致割り込みが許可されます。タイマ/カウンタ1割り込み要求フラグレジスタ(TIFR1)に配置された比較1A割り込み要求フラグ(OCF1A)が設定(1)されると、対応する割り込みベクタ(30頁の「割り込み」参照)が実行されます。

• ビット0 - TOIE1: タイマ/カウンタ1溢れ割り込み許可 (Timer/Counter1 Overflow Interrupt Enable)

このビットが1を書かれて、ステータス レジスタ(SREG)の全割り込み許可(I)ビットが設定(1)されると、タイマ/カウンタ1溢れ割り込みが許可されます。タイマ/カウンタ1割り込み要求フラグレジスタ(TIFR1)に配置されたタイマ/カウンタ1溢れ割り込み要求フラグ(TOV1)が設定(1)されると、対応する割り込みベクタ(30頁の「割り込み」参照)が実行されます。

## タイマ/カウンタ1割り込み要求フラク・レシ、スタ (Timer/Counter1 Interrupt Flag Register) TIFR1

ピット	7	6	5	4	3	2	1	0	1
\$16 (\$36)	_	_	ICF1	_	_	OCFIB	OCFIA	TOVI	TIFR1
Read/Write	R	R	R/W	R	R	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

• ビット5 - ICF1: タイマ/カウンタ1捕獲割り込み要求フラグ(Timer/Conter1, Input Capture Flag)

ICP1ピンに捕獲の事象が起こると、このフラグが設定(1)されます。捕獲レジスタ(ICR1)がWGM13~0によってTOP値として設定されると、ICF1フラグはカウンタがTOP値に到達する時に設定(1)されます。

捕獲割り込みへ、クタが実行されると、ICF1は自動的に解除(0)されます。代わりにこのビット位置へ論理1を書くことによってもICF1は解除(0)できます。

• ビット2 - OCF1B: タイマ/カウンタ1比較B割り込み要求フラグ (Timer/Conter1, Output Compare B Match Flag)

このフラグはカウンタ(TCNT1)値が比較Bレジスタ(OCR1B)と一致した後(次)のタイマ/カウンタ クロック周期で設定(1)されます。

強制的な比較出力(FOC1B)ストローブがOCF1Bフラグを設定(1)しないことに注意してください。

比較B一致割り込みへ、クタが実行されると、OCF1Bは自動的に解除(0)されます。代わりにこのビット位置へ論理1を書くことによってもOCF1Bは解除(0)できます。

• ビット1 - OCF1A: タイマ/カウンタ1比較A割り込み要求フラグ(Timer/Conter1, Output Compare A Match Flag)

このフラグはカウンタ(TCNT1)値が比較Aレジスタ(OCR1A)と一致した後(次)のタイマ/カウンタ クロック周期で設定(1)されます。

強制的な比較出力(FOC1A)ストローブがOCF1Aフラグを設定(1)しないことに注意してください。

比較A一致割り込みベクタが実行されると、OCF1Aは自動的に解除(0)されます。代わりにこのビット位置へ論理1を書くことによっても OCF1Aは解除(0)できます。

• ビット0 - TOV1: タイマ/カウンタ1溢れ割り込み要求フラグ(Timer/Counterl Overflow Flag)

このフラグの(1)設定はWGM13~0ビット設定に依存します。標準またはCTC動作でのTOV1フラグはタイマ/カウンタ1溢れ時に設定(1)されます。他のWGM13~0ビット設定を使用する時のTOV1フラグ動作については82頁の表58.を参照してください。

タイマ/カウンタ1溢れ割り込みベウタが実行されると、TOV1は自動的に解除(0)されます。代わりにこのビット位置へ論理1を書くことによってもTOV1は解除(0)できます。





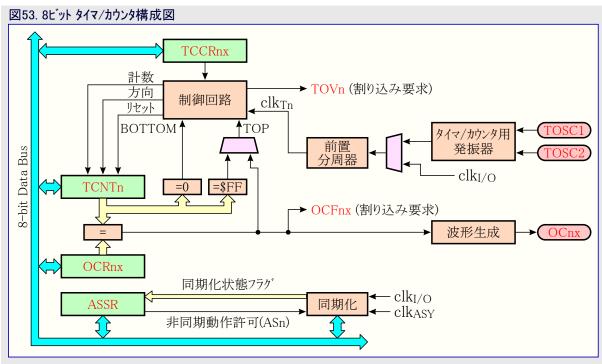
# 8ビット タイマ/カウンタ2 (PWM, 非同期動作付き)

タイマ/カウンタ2は単一比較部付きの汎用8ビット タイマ/カウンタ部です。主な特徴は次のとおりです。

- 単一比較部付き計数器
- 比較一致でのタイマ/カウンタ解除(自動再設定)
- 不具合なしで正しい位相のパルス幅変調器 (PWM)
- 周波数発生器
- クロック用10ビット前置分周器
- 溢れと比較一致割り込み(TOV2とOCF2A)
- I/O(システム)クロックに依存しない時計用外部32kHzクリスタルからのクロック駆動可能

#### 概要

この8ビット タイマ/カウンタの簡単化した構成図は**図53**.で示されます。I/Oピンの実際の配置については2頁の「ピン配置」を参照してください。CPUがアクセス可能な(I/OビットとI/Oピンを含む)I/Oレジスタは赤文字(訳注:原文は太字)で示されます。デバイス仕様のI/Oレジスタとビット位置は94頁の「8ビット タイマ/カウンタ2用レジスタ」で一覧されます。



#### 関係レジスタ

gイマ/カウンg(TCNT2)と比較レジ、スg(OCR2A)は8ビットのレジ、スgです。割り込み要求信号はgイマ/カウンg割り込み要求レジ、スg(TIFR2)で全て見えます。全ての割り込みはgイマ/カウンg割り込み許可レジ、スg(TIMSK2)で個別に遮蔽(禁止)されます。TIFR2とTIMSK2はこの図で示されません。

このタイマ/カウンタは内部的、前置分周器経由、または本章内後ろで詳述されるようにTOSC1/2ピンから非同期にクロック駆動されます。 非同期動作は非同期状態レシ、スタ(ASSR)によって制御されます。 クロック選択論理部はタイマ/カウンタが値を増加(または減少)するのに使う クロック元を制御します。 クロック元が選ばれないと、このタイマ/カウンタは動きません。 クロック選択論理部からの出力はタイマ/カウンタ クロック(clk<sub>T</sub>2)として参照されます。

2重緩衝化した比較レシ、スタ(OCR2A)はタイマ/カウンタ値と常に比較されます。この比較結果は比較出力(OC2A)ピンでPWMまたは可変周波数出力を生成するための波形生成器によって使えます。詳細については88頁の「比較出力部」をご覧ください。この比較一致発生は比較出力割り込み要求の発生に使える比較一致割り込み要求フラグ(OCF2A)も設定(1)します。

#### 定義

本文書でのレシ、スタとヒット参照の多くは一般形で書かれます。小文字の'n' はタイマ/カウンタ番号、この場合は2で置き換えます。けれどもプログラムでレシ、スタまたはヒット定義に使う時は正確な形式が使われなければなりません (例えばタイマ/カウンタ2のカウンタ値のアクセスに対するTCNT2のように)。

表60.の定義は本文書を通して広範囲に渡って使われます。

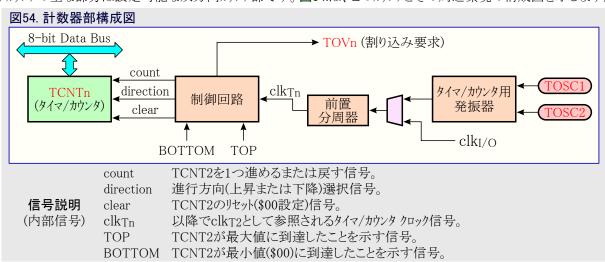
表60. 用語定義							
用語意味							
BOTTOM	タイマ/カウンタが\$00に到達した時。						
MAX	タイマ/カウンタが\$FF(255)に到達した時。						
TOP	タイマ/カウンタが指定された固定値(\$FF) またはOCR2A値に到達した時。この指 定(TOP)値は動作種別に依存します。						

## タイマ/カウンタの クロック

このタイマ/カウンタは内部同期または外部非同期クロック元によってクロック駆動できます。既定のクロック元(clk<sub>T2</sub>)はMCUクロック(clk<sub>I/O</sub>)と同じです。非同期状態レシ、スタ(ASSR)の非同期動作許可(AS2)ビットが論理1を書かれると、クロック元はTOSC1とTOSC2に繋がったタイマ/カウンタ用発振器から取得されます。非同期動作の詳細については97頁の「非同期状態レシ、スタ(ASSR)」をご覧ください。クロック元と前置分周器の詳細については99頁の「タイマ/カウンタ2の前置分周器」をご覧ください。

## 計数器部

8ビット タイマ/カウンタの主な部分は設定可能な双方向カウンタ部です。図54.は、このカウンタとその周辺環境の構成図を示します。



使った動作種別に依存して、カウンタは各タイマ/カウンタ クロック(clkT2)で解除(\$00)、増加(+1)、または減少(-1)されます。clkT2はクロック選択 (CS22~0)ビットによって選ばれた内部または外部のクロック元から生成できます。クロック元が選ばれない(CS22~0=000)時にタイマ/カウンタは停止されます。けれどもTCNT2値はタイマ/カウンタ クロック(clkT2)が存在するしないに拘らず、CPUによってアクセスできます。CPU書き込みは全てのカウンタ解除や計数動作を無視します(上位優先権を持ちます)。

計数順序(方法)はタイマ/カウンタ制御レジスタ(TCCR2A)に配置された波形生成種別(WGM21,0)ビットの設定によって決定されます。これらはカウンタ動作(計数)方法とOC2A比較出力に生成される方法間の接続に近いものです。進化した計数順序と波形生成についてより多くの詳細に関しては90頁の「動作種別」をご覧ください。

タイマ/カウンタ溢れ(TOV2)フラグはWGM21,0ビットによって選ばれた動作種別に従って設定(1)されます。TOV2はCPU割り込み発生に使えます。





## 比較出力部

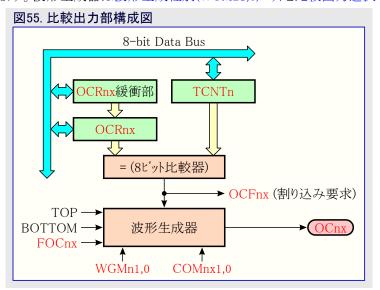
この8ビット比較器はTCNT2と比較レジスタ(OCR2A)を継続的に比較します。TCNT2とOCR2Aが等しければ、比較器は一致を指示します。この一致は次のタイマ/カウンタ クロック周期で比較割り込み要求フラグ(OCF2A)を設定(1)します。許可(I=1, OCIE2A=1)なら、この比較割り込み要求フラグは比較割り込みを発生します。OCF2Aは割り込みが実行されると自動的に解除(0)されます。代わりにOCF2Aはこのビット位置に論理1を書くことによってソフトウェアでも解除(0)できます。波形生成器は波形生成種別(WGM21,0)ビットと比較出力選択

(COM2A1,0)ビットによって設定された動作種別に従った出力を 生成するのにこの一致信号を使います。MAXとBOTTOM信号 は動作種別(90頁の「動作種別」参照)のいくつかで両端値の特 別な場合を扱うため、波形生成器により使われます。

#### 図55.は比較出力部の構成図を示します。

OCR2Aはパルス幅変調(PWM)の何れかを使う時に2重緩衝化されます。標準動作と比較一致タイマ/カウンタ解除(CTC)動作については2重緩衝動作が禁止されます。2重緩衝動作は計数の流れのTOPまたはBOTTOMのどちらかに対してOCR2Aレジスタの更新を同期化します。この同期化は奇数長や非対称PWMパルスの発生を防ぎ、それによって不具合なしの出力を作成します。

OCR2Aのアクセスは複雑なように思えますが決してそんなことはありません。2重緩衝動作が許可されるとCPUはOCR2A緩衝部をアクセスし、禁止されるとOCR2Aレジスタを直接アクセスします。



#### 強制比較出力

非PWM波形生成動作での比較器の一致出力は強制変更(FOC2A)ビットに1を書くことによって強制(変更)できます。比較一致の強制は比較割り込み要求フラケ(OCF2A)の設定(1)やタイマ/カウンタの再設定/解除を行いませんが、OC2Aピンは実際の比較一致が起きた場合と同様に更新されます(COM2A1,0ビット設定がOC2Aピンの設定(1)、解除(0)、1/0交互のどれかを定義)。

## TCNT2書き込みによる比較一致妨害

TCNT2への全てのCPU書き込みは、例えタイマ/カウンタが停止されていても、次のタイマ/カウンタ クロック周期で起こるどんな比較一致をも妨げます。この特質はタイマ/カウンタ クロックが許可されている時に割り込みを起動することなく、TCNT2と同じ値に初期化されることをOCR2Aに許します。

## 比較一致部の使用

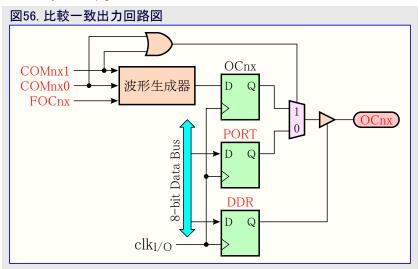
どの動作種別でのTCNT2書き込みでも1タイマ/カウンタ クロック周期間、全ての比較一致を妨げるため、タイマ/カウンタが走行中であるかないかに拘らず、比較出力部を使う場合、TCNT2を変更する時に危険を伴います。TCNT2に書かれた値がOCR2A値と同じ場合に比較一致は失われ(一致が発生せず)、不正な波形生成に終わります。同様にタイマ/カウンタが下降計数のとき、BOTTOMに等しいTCNT 2値を書いてはいけません。

OC2Aの初期設定はポート ピンに対するポート方向レジスタを出力に設定する前に行われるべきです。OC2A値を設定する一番簡単な方法は標準動作で強制変更(FOC2A)ストローブ ビットを使うことです。波形生成動作種別間を変更する時であっても、OC2A(内部)レジスタはその値を保ちます。

比較出力選択(COM2A1,0)と、ットが比較値(OCR2A)と共に2重緩衝されないことに気付いてください。COM2A1,0じ、ットの変更は直ちに有効となります。

## 比較一致出力部

比較出力選択(COM2A1,0)ビットは2つの機能を持ちます。波形生成器は次の比較一致での比較出力(OC2A)状態の定義にCOM2A 1,0ビットを使います。またCOM2A1,0ビットはOC2Aピン出力元を制御します。図56.はCOM2A1,0ビット設定によって影響を及ぼされる論理回路の簡単化した図を示します。図のI/Oレジスタ、I/Oビット、I/Oピンは赤文字(訳注:原文は太字)で示されます。COM2A1,0ビットによって影響を及ぼされる標準I/Oポート制御レジスタ(PORTとDDR)の部分だけが示されます。OC2Aの状態を参照するとき、その参照はOC2Aピンでなく内部OC2Aレジスタに対してです。



COM2A1,0ビットのどちらかが設定(1)されると、標準I/Oポート機能は波形生成器からの比較出力(OC2A)によって無効にされます。けれどもOC2Aピンの方向(入出力)はポート ピンに対するポート方向レジスタ(DDR)によって未だ制御されます。OC2Aピンに対するポート方向レジスタのビット(DDR\_OC2A)はOC2A値がピンで見えるのに先立って出力として設定されなければなりません。このポートの交換機能は波形生成種別と無関係です。

比較出力ピン論理回路の設計は出力が許可される前のOC2A状態の初期化を許します。いくつかのCOM2A1,0ビット設定が或る種の動作種別に対して予約されることに注意してください。94頁の「8ビット タイマ/カウンタ2用レジスタ」をご覧ください。

## 比較一致出力選択と波形生成

波形生成器は標準、CTC、PWM動作でCOM2A1,0ビットを違うふうに使います。全ての動作種別に対してCOM2A1,0=00設定は次の比較一致で実行すべきOC2Aレジスタの動きがないことを波形生成器へ告げます。非PWM動作での比較出力動作については94頁の表62.を参照してください。高速PWM動作については94頁の表63、位相基準PWMについては94頁の表64.を参照してください。

COM2A1,0ビットの状態変更はこのビットが書かれた後の最初の比較一致で有効になります。非PWM動作について、この動作は強制変更(FOC2A)ストローブビットを使うことによって直ちに効果を得ることを強制できます。





## 動作種別

動作種別(換言するとタイマ/カウンタと比較出力ピンの動作)は波形生成種別(WGM21,0)ビットと比較出力選択(COM2A1,0)ビットの組み合わせによって定義されます。比較出力選択ビットは計数順序(動作)に影響を及ぼしませんが、一方波形生成種別ビットは影響を及ぼします。COM2A1,0ビットは生成されるPWM出力が反転されるべきか、されないべきか(反転または非反転PWM)どちらかを制御します。非PWM動作に対するCOM2A1,0ビットは比較一致で出力が解除(0)、設定(1)、1/0交互のどれにされるべきかを制御します(89頁の「比較一致出力部」をご覧ください)。

タイミング情報の詳細については93頁の「タイマ/カウンタ2のタイミング」を参照してください。

#### 標準動作

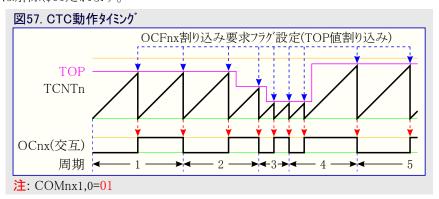
最も単純な動作種別が標準動作(WGM21,0=00)です。本動作種別での計数方向は常に上昇(+)で、カウンタ解除は実行されません。カウンタは8ビット最大値(TOP=\$FF)を通過すると単に範囲を超え、そして\$00(BOTTOM)から再び始めます。通常動作でのタイマ/カウンタ溢れ(TOV2)フラグはTCNT2が\$00になる時と同じタイマ/カウンタ クロック周期で設定(1)されます。この場合のTOV2フラグは設定(1)のみで解除(0)されないことを除いて第9ビットのようになります。けれどもTOV2フラグを自動的に解除(0)するタイマ/カウンタ2溢れ割り込みと組み合わせたタイマ/カウンタの分解能はソフトウェアによって増やせます。標準動作での考慮に特別な場合はなく、新しいカウンタ値は何時でも書けます。

比較出力部は与えられた或る時間に割り込みを生成するのに使えます。標準動作で波形を生成するのに比較出力を使うのは、それが大変多くのCPU時間を占有するため推奨されません。

## 比較一致タイマ/カウンタ解除(CTC)動作

比較一致タイマ/カウンタ解除(CTC)動作(WGM21,0=10)ではOCR2Aがカウンタの分解能を操作するのに使われます。CTC動作ではカウンタ (TCNT2)値がOCR2Aと一致すると、カウンタは\$00に解除されます。OCR2Aはカウンタに対するTOP値、従って分解能も定義します。この動作種別はより大きい比較一致出力周波数の制御を許します。それは外部の出来事の計数操作も簡単にします。

CTC動作についてのタイミング図は図57.で示されます。カウンタ(TCNT2)値はTCNT2とOCR2A間で比較一致が起こるまで増加し、そしてその後にカウンタ(TCNT2)は解除(\$00)されます。



OCF2Aフラグを使うことにより、タイマ/カウンタ値がTOP値に達する時毎に割り込みが生成できます。割り込みが許可されるなら、割り込み処理ルーチンはTOP値を更新するのに使えます。けれども前置分周なしまたは低い前置分周値でカウンタが走行している時にBOTTOMと近い値にTOPを変更することは、CTC動作が2重緩衝機能を持たないために注意して行わなければなりません。OCR2Aに書かれた新しい値がTCNT2の現在値よりも低い(小さい)場合、タイマ/カウンタは(その回の)比較一致を失います。その後のカウンタは比較一致が起こるのに先立って、最大値(\$FF)へそして次に\$00から始める計数をしなければならないでしょう。

CTC動作で波形出力を生成するため、OC2A出力は比較出力選択(COM2A1,0)ビットを交互動作(=01)に設定することによって各比較一致での論理レヘール交互切り替えに設定できます。OC2A値はそのピッとは対するデータ方向が出力(DDR\_OC2A=1)に設定されない限りポートピッで見えないでしょう。生成された波形はOCR2Aが0(\$00)に設定される時に $f_{OC2A} = f_{clk_L}/O/2$ の最大周波数を得ます。生成波形周波数は次式によって定義されます。

$$f_{\text{OCnx}} = \frac{f_{\text{clk\_I/O}}}{2 \times N \times (1 + \text{OCRnx})}$$

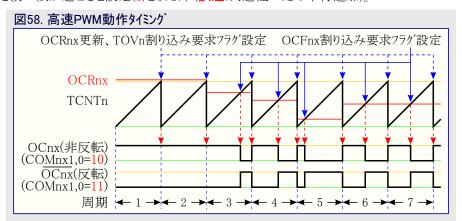
変数Nは前置分周数(1,8,32,64,128,256,1024)を表します。

標準動作と同じように、タイマ/カウンタ溢れ(TOV2)フラグはカウンタがMAXから\$00へ計数するのと同じタイマ/カウンタ クロック周期で設定(1)されます。

#### 高速PWM動作

高速 $^{\circ}$ ルス幅変調(PWM)動作(WGM21,0=11)は高周波数PWM波形生成選択を提供します。高速PWMはそれが単一傾斜(鋸波)動作であることによって他のPWM動作と異なります。カウンタはBOTTOMからMAXまで計数し、その後にBOTTOMから再び始めます。非反転比較出力動作(COM2A1,0=10)での比較出力(OC2A)は、TCNT2とOCR2A間の比較一致で解除(0)され、BOTTOMで設定(1)されます。反転出力動作(COM2A1,0=11)の出力は比較一致で設定(1)され、BOTTOMで解除(0)されます。この単一傾斜動作のため、高速PWM動作の動作周波数は両傾斜(三角波)動作を使う位相基準PWM動作よりも2倍高くできます。この高い周波数は電力調節、整流、D/A変換に対して高速PWM動作を都合よく適合させます。高い周波数は物理的に小さな外部部品(コイルやコンデンサ)を許し、従ってシステム総費用を削減します。

高速PWM動作でのタイマ/カウンタはタイマ/カウンタ値がMAX値と一致するまで増加されます。そしてタイマ/カウンタは(一致の)次のタイマ/カウンタ クロック周期で解除(\$00)されます。高速PWM動作のタイング図は図58.で示されます。TCNT2値はタイシグ図で単一傾斜動作(鋸波)を表す折れ線グラフとして示されます。この図は非反転と反転のPWM出力を含みます。細い赤線はOCR2A値を示し、TCNT2値との交点(接点)がTCNT2とOCR2A間の比較一致を示します(訳注:図補正に伴い本行若干変更)。比較割り込み要求フラグ(OCF2A)はOCR2A=MAXを除いて比較一致が起こると設定(1)されます(訳注:共通性のため本行追加)。



タイマ/カウンタ溢れ(TOV2)フラグはカウンタがMAXに到達する時毎に設定(1)されます。割り込みが許可されるなら、その割り込み処理ルーチンは比較値を更新するのに使えます。

高速PWM動作での比較部はOC2AピンでのPWM波形の生成を許します。COM2A1,0ビットを'10'に設定することは非反転PWM出力を作成し、反転PWM出力はCOM2A1,0を'11'に設定することで生成できます(94頁の表63.をご覧ください)。実際のOC2A値はポート ピン に対するデータ方向(DDR\_OC2A)が出力として設定される場合にだけ見えるでしょう。PWM波形はTCNT2とOCR2A間の比較一致でOC2A(内部)レジスタを設定(1)(または解除(0))と、カウンタが解除(\$00、MAXからBOTTOMへ変更)されるタイマ/カウンタ クロック周期でOC2 A レジスタを解除(0)または設定(1)することによって生成されます。

PWM出力周波数は次式によって計算できます。

$$f_{\text{OCnxPWM}} = \frac{f_{\text{clk\_I/O}}}{N \times 256}$$

変数Nは前置分周数(1,8,32,64,128,256,1024)を表します。

OCR2Aの両端値は高速PWM動作でPWM波形出力を生成する時の特別な場合にあたります。OCR2AがBOTTOM(\$00)と等しく設定されると、出力はMAX+1 タイマ/カケンタ クロック周期毎の狭いスパイク(パルス)になるでしょう。OCR2AがMAXに等しく設定されると、(COM 2A1,0ビットによって設定される出力極性に依存して)定常的なLowまたはHigh出力に終わるでしょう。

(<mark>訳注</mark>:以下は本動作種別がTOP=MAX(\$FF)で固定のため基本的に不適切です。) 高速PWM動作で(デューティ比50%)周波数の波形 出力は比較一致毎に論理反転するOC2A設定(COM2A1,0=01)によって達成できます。生成された波形はOCR2Aが0(\$00)に設定される時にfOC2A=fclk\_I/O/2の最大周波数でしょう。この特性は高速PWM動作で比較出力部の2重緩衝機能が許可されることを除いて、CTC動作でのOC2A交互出力(COM2A1,0=01)と同じです。

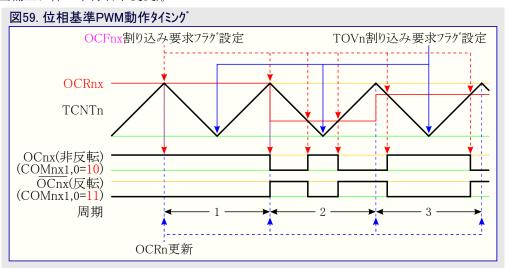




## 位相基準PWM動作

位相基準パルス幅変調(PWM)動作(WGM21,0=01)は高分解能で正しい位相のPWM波形生成選択を提供します。位相基準PWM動作は両傾斜(三角波)動作に基きます。カウンタはBOTTOMからMAXへそして次にMAXからBOTTOMへを繰り返し計数します。非反転比較出力動作(COM2A1,0=10)での比較出力(OC2A)は上昇計数中のTCNT2とOCR2Aの比較一致で解除(0)され、下降計数中の比較一致で設定(1)されます。反転出力動作(COM2A1,0=11)での動作は逆にされます。両傾斜(三角波)動作は単一傾斜(鋸波)動作よりも低い最大動作周波数になります。けれども両傾斜(三角波)動作の対称特性のため、これらの動作種別は電動機制御の応用に好まれます。

位相基準PWM動作のPWM分解能は8ビットに固定されます。位相基準PWM動作でのカウンタはカウンタ値がMAXと一致するまで増加されます。カウンタはMAXに達すると計数方向を変更します。このTCNT2値は1タイマ/カウンタ クロック周期間MAXと等しくなります。位相基準PWM動作のタイミング図は図59.で示されます。TCNT2値はタイミング図で両傾斜動作(三角波)を表す折れ線グラフとして示されます。この図は非反転と反転のPWM出力を含みます。細い赤線はOCR2A値を示し、TCNT2値との交点(接点)がTCNT2とOCR2A間の比較一致を示します(訳注:図補正に伴い本行若干変更)。



タイマ/カウンタ溢れ(TOV2)フラグはタイマ/カウンタがBOTTOMに到達する時毎に設定(1)されます。この割り込み要求フラグはカウンタがBOTT OM値に到達する毎に割り込みを発生するのに使えます。

位相基準PWM動作での比較部はOC2AピンでのPWM波形の生成を許します。COM2A1,0ビットを'10'に設定することは非反転PWM 出力を作成し、反転PWM出力はCOM2A1,0ピットを'11'に設定することで生成できます(94頁の表64.をご覧ください)。実際のOC2A値はそのポート ピンに対するデータ方向(DDR\_OC2A)が出力として設定される場合にだけ見えるでしょう。PWM波形はカウンタが増加する時のTCNT2とOCR2A間の比較一致でOC2A(内部)レジスタを設定(1)(または解除(0))と、カウンタが減少する時のTCNT2とOCR2A間の比較一致でOC2Aレジ゙スタを解除(0)(または設定(1))によって生成されます。位相基準PWMを使う時の出力に対するPWM周波数は次式によって計算できます。

$$f_{\text{OCnxPCPWM}} = \frac{f_{\text{clk\_I/O}}}{N \times 510}$$

変数Nは前置分周数(1,8,32,64,128,256,1024)を表します。

OCR2Aの両端値は位相基準PWM動作でPWM波形出力を生成する時の特別な場合にあたります。非反転PWM動作ではOCR2AがBOTTOM(\$00)に等しく設定されると出力は定常的なLow、MAXに等しく設定されると定常的なHighになるでしょう。反転PWMに対する出力は逆の論理値になります。

図59.の第2周期のその出発点に於いて、例え比較一致がないとしても、OCnxにはHighからLowへの遷移があります。この遷移点はBOTTOMを挟む対称を保証するためです。比較一致なしに遷移を生ずるのは2つの場合です。

- 図59.でのようにOCR2AはMAXからその値を変更します。OCR2A値がMAXのとき、OCnxピン値は下降計数での比較一致の結果と同じです(<mark>訳補: L→H</mark>、直前がHのため、常にH)。BOTTOMを挟む対称を保証するため、(変更直後の)MAX(位置)でのOCnx値は上昇計数での比較一致の結果(H→L)と一致しなければなりません。
- ・タイマ/カウンタがOCR2A値よりも高い値から数え始め、そしてその理屈のために比較一致、それ故上昇途中で起こされるであろうOC nxの変更を逃します。(<mark>訳補</mark>: 従って上記同様、MAX位置で(直前がHならば)H→L遷移が生じます。)

# タイマ/カウンタのタイミング

以下の図は同期動作でのタイマ/カウンタを示し、従ってタイマ/カウンタ クロック(clkT2)が計数許可信号として示されます。非同期動作では clk<sub>I/O</sub>がタイマ/カウンタ用発振器(TOSC)クロックによって置換されるべきです。この図は割り込みフラグが設定(1)される時の情報を含みます。図60.は基本的なタイマ/カウンタ動作についてのタイミング データを示します。この図は位相基準PWM動作以外の全ての動作種別での MAX値近辺の計数の流れを示します。

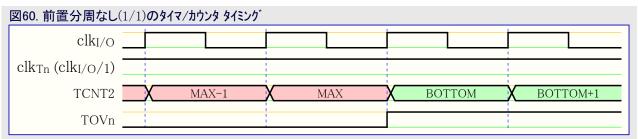


図61.は同じタイミング・データを示しますが、前置分周器が許可されています。

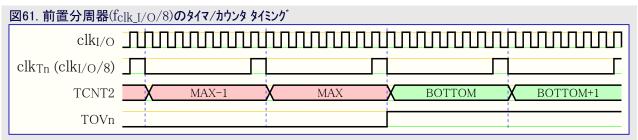


図62.はCTC動作を除く全ての動作種別でのOCF2Aの設定を示します。

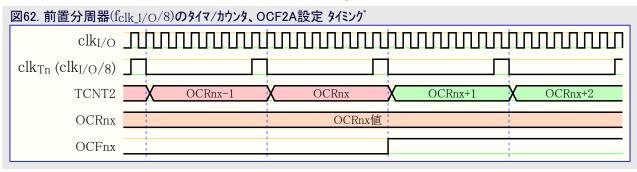
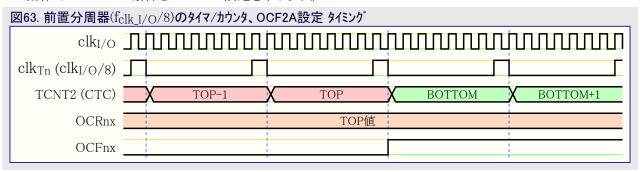


図63.はCTC動作でのTCNT2の解除とOCF2Aの設定を示します。







## 8ビット タイマ/カウンタ2 用レジスタ

タイマ/カウンタ2制御レシ、スタA (Timer/Counter2 Control Register A) TCCR2A

ピット	7	6	5	4	3	2	1	0	_
(\$B0)	FOC2A	WGM20	COM2A1	COM2A0	WGM21	CS22	CS21	CS20	TCCR2A
Read/Write	W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

• ビット7 - FOC2A: OC2A強制変更 (Force Output Compare 2A)

FOC2AビットはWGM21,0ビットが非PWM動作を指示する時だけ有効です。けれども将来のデバイスとの共通性を保証するため、PWM 動作で扱う時にTCCR2Aが書かれる場合、このビットは<mark>0</mark>に設定されなければなりません。FOC2Aビットに論理1を書くと、波形生成部で 直ちに比較一致が強制されます。OC2A出力はCOM2A1,0ビット設定に従って変更されます。FOC2Aビットがストロープとして実行される ことに注意してください。従って強制した比較の効果を決めるのはCOM2A1,0ビットに存在する値です。

FOC2Aストローフ、は何れの割り込みの生成もTOPとしてOCR2Aを使う比較一致タイマ解除(CTC)動作でのタイマ/カウンタの解除(\$00)も行い ません。

FOC2Atiットは常に0として読みます。

• ビット3,6 - WGM21,0: 波形生成種別 (Waveform Generation Mode bit 1 and 0)

これらのビットはカウンタの計数順序(方向)、最大 (TOP)カウンタ値の供給元、使われるべき波形生 成のどの形式かを制御します。タイマ/カウンタ部に よって支援される動作種別は標準動作、比較 一致タイマ/カウンタ解除(CTC)動作と2形式のパル ス幅変調(PWM)動作です。表61.と90頁の「動作 種別」をご覧ください。

番号	WGM21 (CTC2)	WGM20 (PWM2)	タイマ/カウンタ動作種別	TOP値	OCR2A 更新時	TOV2 設定時
0	0	0	標準動作	\$FF	即時	MAX
1	0	1	位相基準PWM動作	\$FF	TOP	BOTTOM
2	1	0	比較一致タイマ/カウンタ 解除(CTC)動作	OCR2A	即時	MAX
3	1	1	高速PWM動作	\$FF	ВОТТОМ	MAX

注: CTC2とPWM2ビット定義名は旧名称です。WGM21,0定義を使ってください。 しかし、これらのビットの機能と位置は旧版のタイマ/カウンタと一致します。

• ビット5,4 - COM2A1,0:比較2A出力選択(Compare Match 2A Output Mode bit 1 and 0)

これらのビットはOC2A比較出力ピンの動作を制御します。COM2A1,0ビットの1つまたは両方が1を書かれると、OC2A出力はそのI/Oピ ンの通常ポート機能を無効にし、そのI/Oピンに接続されます。 けれども出力駆動部を許可するため、OC2Aピンに対応するポート方向レ ジスタ(DDR)のビットが設定(1)されなければならないことに注意してください。

OC2Aがピンに接続されるとき、COM2A1,0ビットの機能はWGM21,0ビット設定に依存します。

表62.はWGM21,0ビットが標準動作またはCTC動作(つまりPWM以 表62. 非PWM動作での比較出力選択 外)に設定される時のCOM2A1,0ビット機能を示します。

表63.はWGM21,0ビットが高速PWM動作に設定される時のCOM2 A1,0ビットの機能を示します。

表64.はWGM21,0ビットが位相基準PWM動作に設定される時のC OM2A1,0ビットの機能を示します。

COM2A1	COM2A0	意味
0	0	標準ポート動作 (OC2A切断)
0	1	比較一致でOC2At°ントケル(交互)出力
1	0	比較一致でOC2At°ン Lowレヘブル出力
1	1	比較一致でOC2At°ン Highレヘブル出力

#### 表63. 高速PWM動作での比較出力選択

COM2A1	COM2A0	意味
0	0	標準ポート動作(OC2A切断)
0	1	(予約)
1	0	比較一致でLow、BOTTOMでHighを OC2Aピンへ出力(非反転動作)
1	1	比較一致でHigh、BOTTOMでLowを OC2Aピンへ出力(反転動作)

注: COM2A1が設定(1)され、OCR2AがTOPと等しい時に特別 な状態が起きます。この状態での比較一致は無視されま すが、BOTTOMでの設定(1)や解除(0)は行われます。より 多くの詳細については91頁の「高速PWM動作」をご覧くだ さい。

#### 表64. 位相基準PWM動作での比較出力選択

COM2A1	COM2A0	意味
0	0	標準ポート動作 (OC2A切断)
0	1	(予約)
1	0	上昇計数時の比較一致でLow、下降計数 時の比較一致でHighをOC2Aピンへ出力
1	1	上昇計数時の比較一致でHigh、下降計数 時の比較一致でLowをOC2Atシ〜出力

注: COM2A1が設定(1)され、OCR2AがTOPと等しい時に特別 な状態が起きます。この状態での比較一致は無視されま すが、TOPでの設定(1)や解除(0)は行われます。より多く の詳細については92頁の「位相基準PWM動作」をご覧く ださい。

• ビット2~0 - CS22~0: クロック選択2 (Clock Select2, bit 2,1 and 0)

この3つのクロック選択ビットはタイマ/カウンタ(TCNT2)によって使われるクロック元を選びます。表65.をご覧ください。

## 表65. タイマ/カウンタ2入力クロック選択

CS22	CS21	CS20	意味
0	0	0	停止(タイマ/カウンタ2動作停止)
0	0	1	clk <sub>T2S</sub> (前置分周なし)
0	1	0	clk <sub>T2S</sub> /8 (8分周)
0	1	1	clk <sub>T2S</sub> /32 (32分周)
1	0	0	clk <sub>T2S</sub> /64 (64分周)
1	0	1	clk <sub>T2S</sub> /128 (128分周)
1	1	0	clk <sub>T2S</sub> /256 (256分周)
1	1	1	clk <sub>T2S</sub> /1024 (1024分周)

## タイマ/カウンタ2 (Timer/Counter2) TCNT2

ピット	7	6	5	4	3	2	1	0	_
(\$B2)	(MSB)							(LSB)	TCNT2
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

このタイマ/カウンタ レジ スタは読み書き両方の操作について、タイマ/カウンタ部の8ビット カウンタに直接アクセスします。TCNT2への書き込みは次のタイマ/カウンタ クロックでの比較一致を妨害(除去)します。カウンタが走行中にカウンタ(TCNT2)を変更することは、TCNT2とOCR2A間の比較一致消失の危険を誘発します。

## タイマ/カウンタ2 比較Aレジスタ (Timer/Counter2 Output Compare Register) OCR2A

ピット	7	6	5	4	3	2	1	0	
(\$B3)	(MSB)							(LSB)	OCR2A
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

この比較レシ、スタは継続的にカウンタ(TCNT2)値と比較される8ビットの値を含みます。一致は比較一致割り込みやOC2Aピンでの波形出力を生成するのに使えます。





## タイマ/カウンタ2割り込み許可レジスタ (Timer/Counter2 Interrupt Mask Register) TIMSK2

ピット	7	6	5	4	3	2	1	0	_
(\$70)	_	-	-	-	-	-	OCIE2A	TOIE2	TIMSK2
Read/Write	R	R	R	R	R	R	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

• ビット1 - OCIE2A: タイマ/カウンタ2比較A割り込み許可 (Timer/Counter2 Output Compare A Match Interrupt Enable)

OCIE2Aビットが1を書かれ、ステータスレシ、スタ(SREG)の全割り込み許可(I)ビットが設定(1)されると、タイマ/カウンタ2比較一致割り込みが許可されます。タイマ/カウンタ2で比較一致が起こる、換言するとタイマ/カウンタ2割り込み要求フラク、レシ、スタ(TIFR2)でタイマ/カウンタ2比較一致割り込み要求フラク、(OCF2A)が設定(1)されると、対応する割り込みが実行されます。

• ビット0 - TOIE2: タイマ/カウンタ2溢れ割り込み許可 (Timer/Counter2 Overflow Interrupt Enable)

TOIE2ビットが1を書かれ、ステータス レジ、スタ(SREG)の全割り込み許可(I)ビットが設定(1)されると、タイマ/カウンタ2溢れ割り込みが許可されます。タイマ/カウンタ2溢れが起こる、換言するとタイマ/カウンタ2割り込み要求フラケ レジ、スタ(TIFR2)でタイマ/カウンタ2溢れ割り込み要求(TOV2)フラケが設定(1)されると、対応する割り込みが実行されます。

## タイマ/カウンタ2割り込み要求フラク レシ スタ (Timer/Counter2 Interrupt Flag Register) TIFR2

ピット	7	6	5	4	3	2	1	0	_
\$17 (\$37)	-	-	-	-	-	-	OCF2A	TOV2	TIFR2
Read/Write	R	R	R	R	R	R	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

• ビット1 - OCF2A: タイマ/カウンタ2比較A割り込み要求フラク (Timer/Conter2, Output Compare A Match Flag)

OCF2Aビットは比較一致がタイマ/カウンタ(TCNT2)と比較レジスタ(OCR2A)間で起こる時に設定(1)されます。対応する割り込み処理ベクタを実行するとき、OCF2Aはハードウェアによって解除(0)されます。代わりにこのフラグへ論理1を書くことによってもOCF2Aは解除(0)されます。ステータス レジスタ(SREG)の全割り込み許可(I)ビット、タイマ/カウンタ2割り込み許可レジスタ(TIMSK2)のタイマ/カウンタ2比較一致割り込み許可(OCIE2A)ビット、OCF2Aが設定(1)されると、タイマ/カウンタ2比較一致割り込みが実行されます。

• ビット0 - TOV2 : タイマ/カウンタ2溢れ割り込み要求フラグ (Timer/Counter2 Overflow Flag)

TOV2ビットはタイマ/カウンク(TCNT2)溢れが起こる時に設定(1)されます。対応する割り込み処理ベクタを実行するとき、TOV2はハードウェアによって解除(0)されます。代わりにこのフラグへ論理1を書くことによってもTOV2は解除(0)されます。ステータス レジスタ(SREG)の全割り込み許可(I)ビット、タイマ/カウンタ2割り込み許可レジスタ(TIMSK2)のタイマ/カウンタ2溢れ割り込み許可(TOIE2)ビット、TOV2が設定(1)されると、タ イマ/カウンタ2溢れ割り込みが実行されます。PWM動作ではタイマ/カウンタ2が\$00で計数方向を変える時にこのビットが設定(1)されます。

## タイマ/カウンタ2の非同期動作

#### タイマ/カウンタ2非同期状態レシ、スタ (Timer/Counter2 Asynchronous Status Register) ASSR

ピット (\$B6)	7	6	5 –	4 EXCLK	3 AS2	2 TCN2UB	1 OCR2UB	0 TCR2UB	ASSR
Read/Write	R	R	R	R/W	R/W	R	R	R	
初期値	0	0	0	0	0	0	0	0	

## • ビット4 - EXCLK:外部クロック信号許可 (Enable External Clock Input)

EXCLKが1を書かれ、非同期クロックが選ばれると、外部クロック入力緩衝部が許可され、32kHzクリスタルの代わりに外部クロックがタイマ発振器1(TOSC1)ピンへ入力できます。EXCLKへの書き込みは非同期動作が選ばれる前に行うべきです。クリスタル発振器はこのビットが0の時にだけ走行(動作)することに注意してください。

#### • ビット3 - AS2: タイマ/カウンタ2非同期動作許可 (Asynchronous Timer/Counter2)

AS2が0を書かれると、タイマ/カウンタ2はI/Oクロック( $clk_{I/O}$ )からクロック駆動されます。AS2が1を書かれると、タイマ/カウンタ2はタイマ発振器 (TOSC1,TOSC2)ピンに接続されたクリスタル発振器からクロック駆動されます。AS2の値が変更されると、タイマ/カウンタ2(TCNT2)、比較2レジスタ(OCR2A)、タイマ/カウンタ2制御レジスタA(TCCR2A)の内容は不正にされるかもしれません。

#### • ビット2 - TCN2UB: タイマ/カウンタ2更新中フラグ(Timer/Counter2 Update Busy)

タイマ/カウンタ2が非同期に動き、タイマ/カウンタ2(TCNT2)が書かれると、このビットが設定(1)になります。TCNT2が一時保存レジスタから更新されてしまうと、このビットはハートヴェアによって解除(0)されます。このビットの論理0はTCNT2が新しい値で更新される用意ができたことを示します。

## • ビット1 - OCR2UB: 比較2レシスタ更新中フラク (Output Compare Register2 Update Busy)

タイマ/カウンタ2が非同期に動き、比較2レシ、スタ(OCR2A)が書かれると、このビットが設定(1)になります。OCR2Aが一時保存レジ、スタから更新されてしまうと、このビットはハート・ウェアによって解除(0)されます。このビットの論理0はOCR2Aが新しい値で更新される用意ができたことを示します。

#### • ビット0 - TCR2UB: タイマ/カウンタ2制御レジスタ更新中フラグ (Timer/Counter2 Control Register Update Busy)

タイマ/カウンタ2が非同期に動き、タイマ/カウンタ2制御レジスタA(TCCR2A)が書かれると、このビットが設定(1)になります。TCCR2Aが一時保存レシ、スタから更新されてしまうと、このビットはハードウェアによって解除(0)されます。このビットの論理0はTCCR2Aが新しい値で更新される用意ができたことを示します。

更新中フラグが設定(1)中に3つのタイマ/カウンタ2 レジスタのどれかに書き込みが実行されると、更新された値は不正にされ、予期せぬ割り込みを起こす原因になるかもしれません。

TCNT2、OCR2A、TCCR2A読み込みについての機構は異なります。TCNT2を読む時は実際のタイマ/カウンタ値が読まれ、OCR2AまたはTCCR2Aを読む時は一時保存レジスタの値が読まれます。





#### タイマ/カウンタ2 非同期動作

タイマ/カウンタ2が非同期で動くとき、いくつかの考慮が成されなければなりません。

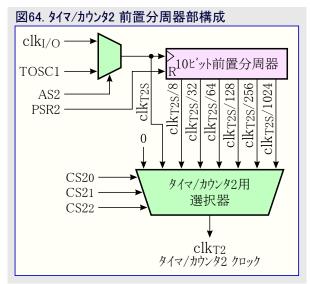
- 警告: タイマ/カウンタ2の同期/非同期クロック駆動間を切り替えると、タイマ/カウンタ2(TCNT2)、タイマ/カウンタ2比較レシ、スタ(OCR2A)、タイマ/カウンタ2制御レシ、スタA(TCCR2A)が不正となるかもしれません。クロック元を切り替える安全な手順を次に示します。
  - 1. タイマ/カウンタ2割り込み許可レジスタ(TIMSK2)のOCIE2AとTOIE2の解除(0)により、タイマ/カウンタ2割り込みを禁止します。
  - 2. 非同期状態レジスタ(ASSR)の非同期動作許可(AS2)設定によってクロック元を適切に選びます。
  - 3. TCNT2,OCR2A,TCCR2Aに新しい値を書きます。
  - 4. 非同期動作へ切り替えるには、TCN2UB、OCR2UB、TCR2UBについて(=0まで)待機します。
  - 5. タイマ/カウンタ2割り込み要求フラク・レシ、スタ(TIFR2)のOCF2AとTOV2フラクを解除(0)します。
  - 6. 必要とされるなら、割り込みを許可します。
- CPU主クロック周波数はタイマ発振器周波数の4倍よりも高くなければなりません。
- TCNT2,OCR2A,TCCR2Aレジスタの1つに書くとき、その値は一時レジスタへ転送され、TOSC1で2つの上昇端後、(実レジスタ)に設定されます。一時レジスタの内容がそれらの転送先へ転送されてしまう前に新しい値を書くべきではありません。記載された3つのレジスタの各々がそれら個別の一時レジスタを持ちます。それは、例えばTCNT2書き込みがOCR2A書き込みの実行を妨げないことを意味します。転送先レジスタへの転送が起きたことを検知するため、非同期状態レジスタ(ASSR)は実装されました。
- TCNT2,OCR2A,TCCR2Aに書いた後でパワーセーブ動作またはA/D変換雑音低減動作へ移行するとき、デバイスを起動するのにタイマ/カウンタ2が使われる場合、使用者は書かれたレジスタが更新されてしまうまで待たなければなりません。さもなければMCUは変更が有効になる前に休止形態へ移行するでしょう。タイマ/カウンタ2比較一致割り込みがデバイスを起動するのに使われる場合、TCNT2またはOCR2A書き込み中の比較機能が禁止されるため、これは非常に重要です。書き込み周期が完了されず、OCR2UBが0に戻る前にMCUが休止形態へ移行すると、デバイスは決して比較一致割り込みを受け取らず、そしてMCUは起動しないでしょう。
- ハプローセーブまたはA/D変換雑音低減動作からデバイスを起動するのにタイマ/カウンタ2が使われる場合、使用者がこれらの動作へ(復帰後)再移行することを欲するなら、予防処置を講じなければなりません。割り込み論理回路はリセットするのに1 TOSC1周期必要です。起動と休止形態再移行間の時間が1 TOSC1周期未満の場合に割り込みが起きず、デバイスは起動に失敗するでしょう。パワーセーブまたは拡張スタンバイ動作再移行前の時間が充分であるかどうか迷うなら、1 TOSC1周期が経過されることを保証するのに次の手順が使えます。
  - 1. TCNT2,OCR2A,TCCR2Aに値を書きます。
  - 2. 非同期状態レシ、スタ(ASSR)の対応する更新中フラグが0に戻るまで待ちます。
  - 3. パワーセーブまたはA/D変換雑音低減動作へ移行します。
- 非同期動作が選ばれると、タイマ/カウンタ2用32.768kHz発振器はパワーダウン動作とスタンハイ動作を除いて常に動作します。電源投入リセット、パワーダウンまたはスタンハイ動作から起動後、この発振器が安定するのに1秒程度かかるかもしれないことを使用者は承知すべきです。電源投入、パワーダウンまたはスタンハイ動作から起動後、タイマ/カウンタ2を使う前に少なくとも1秒待機することが推奨されます。この発振器の使用またはクロック信号がTOSCピンに印加されるかのどちらかに拘らず、起動時の不安的なクロック信号のため、パワーダウンまたはスタンハイ動作からの起動復帰後、タイマ/カウンタ2の全レジスタの内容が失われたと見做されなければなりません。
- タイマ/カウンタ2が非同期でクロック駆動される時のパワーセーブまたはA/D変換雑音低減動作から起動の説明。割り込み条件が合致すると、タイマ/カウンタクロックの次の周期で起動処理が開始され、プロセッサがカウンタ値を読めるのに先立ってタイマ/カウンタは常に最低1進行されます。起動後MCUは4周期停止され、割り込み処理ルーチンを実行し、そしてSLEEP命令の次の命令から実行を再開します。
- パワーセーブ動作から起動直後のTCNT2の読み込みは不正な結果を得るかもしれません。TCNT2が非同期TOSCクロックでクロック駆動されるため、TCNT2読み込みは内部I/Oクロック領域に同期したレジスタを通して行われなければなりません。同期化はTOSCクロックの全上昇端で行われます。パワーセーブ動作から起動し、I/Oクロック(clk<sub>I/O</sub>)が再び活性(有効)になるとき、TCNT2はTOSCクロックの次の上昇端まで以前(休止形態移行前)の値を読むでしよう。パワーセーブ動作から起動後のTOSCクロックの位相は起動時間に依存するため本質的に特定できません。従ってTCNT2読み込みに対する推奨手順は次のとおりです。
  - 1. OCR2AまたはTCCR2Aのどちらかに何か値を書きます。
  - 2. 非同期状態レジスタ(ASSR)の対応する更新中フラグが解除(0)されるまで待ちます。
  - 3. TCNT2を読みます。
- 非同期動作の間中、非同期タイマ用割り込み要求フラグの同期化は3プロセッサ周期+1タイマ周期かかります。従ってプロセッサが割り込み要求フラグ設定の原因となったタイマ値を読めるのに先立って、このタイマは最低1、進行されます。比較出力ピンはタイマクロックで変更され、プロセッサクロックに同期されません。

## タイマ/カウンタ2の前置分周器

タイマ/カウンタ2用クロック元の名前はclkT2Sです。既定でのclkT2Sは主システム I/Oクロック(clkI/O)に接続されます。タイマ/カウンタ2非同期状態レシ、スタ(ASSR)の非同期クロック(AS2)ビットの設定(1)により、タイマ/カウンタ2はTOSC1ピンから非同期にクロック駆動されます。これは実時間計数器(RTC)としてのタイマ/カウンタ2の使用を可能にします。AS2が設定(1)されると、TOSC1とTOSC2ピンは主クロック発振器から切り離されます。クリスタル発振子はタイマ/カウンタ2用の独立したクロック元として扱うため、TOSC1とTOSC2ピン間に接続できます。この発振器は32.768kHzクリスタル発振子で使うために最適化されています。TOSC1に外部クロック信号を印加する場合、ASSRの外部クロック許可(EXCLK)ビットが設定(1)されなければなりません。

gイマ/カウンg2に対して可能な前置分周済み選択はclk $_{T2S}/8$ , clk $_{T2S}/32$ , clk $_{T2S}/64$ , clk $_{T2S}/128$ , clk $_{T2S}/256$ , clk $_{T2S}/1024$ です。加えて0(停止)は 勿論clk $_{T2S}$ も選択可能です。

一般タイマ/カウンタ制御レシ、スタ(GTCCR)のタイマ/カウンタ2 前置分周器リセット(PSR 2)ヒ、ットの設定(1)は前置分周器をリセットします。これは予測可能な前置分周器での操作を使用者に許します。



## 一般タイマ/カウンタ制御レジスタ (General Timer/Counter Control Register) GTCCR

ピット	7	6	5	4	3	2	1	0	
\$23 (\$43)	TSM	-	-	-	-	-	PSR2	PSR10	GTCCR
Read/Write	R/W	R	R	R	R	R	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

• ビット1 - PSR2: タイマ/カウンタ2 前置分周器リセット (Prescaler Reset Timer/Counter2)

このビットが1の時にタイマ/カウンタ2の前置分周器はリセットします。通常、このビットはハードウェアによって直ちに解除(0)されます。タイマ/カウンタ2が非同期動作の時にこのビットが(1を)書かれると、このビットは前置分周器がリセットされてしまうまで1に留まります。TSMビットが設定(1)される場合、このビットはハードウェアによって解除(0)されません。タイマ/カウンタ同期(同時)動作の記載については66頁の「・ビット7 - TSM: タイマ/カウンタ同時動作」の記述を参照してください。





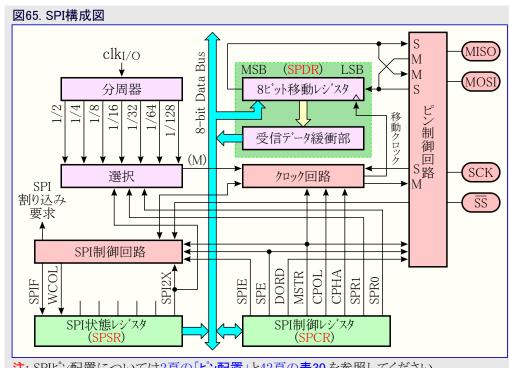
# 直列周辺インターフェース (SPI: Serial Peripheral Interface)

直列周辺インターフェースはATmega169と様々なAVRデバイスや周辺デバイス間の高速同期データ転送を許します。ATmega169のSPIは次の 特徴を含みます。

- 全二重3線同期データ転送
- 主装置/従装置動作
- LSB/MSB先行データ転送
- 設定変更可能な7つのビット速度
- 送信完了割り込み要求フラグ
- 送信上書きフラグ保護
- アイドル動作からの起動
- 倍速(CK/2)主装置SPI動作

23頁の「**電力削減レジスタ**(PRR)」での PRSPIt ットはSPI部を許可するために0 を書かれなければなりません。

SPIでの主装置と従装置のCPU間相 互連結は図66.で示されます。このシス テムは2つの移動レシ、スタと主装置クロック 発生器から成ります。SPI主装置は希 望した従装置のSS(従装置選択)ピンを Lowへ引き込む時に一群の通信を始 めます。主装置と従装置は各々の移 動レジスタで送るべきデータを用意し、主 装置はデータを交換するのに必要なク ロックハ°ルスをSCK信号線に生成しま す。データは常にMOSI(Master Out Slave In)信号線を主装置から従装置



注: SPIピン配置については2頁の「ピン配置」と42頁の表30.を参照してください。

へ、MISO(Master In Slave Out)信号線を従装置から主装置へ移動されます。各データ パケット後、主装置はSS(従装置選択)ピンをHigh へ引き上げることによって従装置と同期を取ります。

主装置として設定されると、SPIインターフェースにはSS信号線の自動制御がありません。これは通信が開始できるのに先立って使用者ソフ トウェアによって操作されなければなりません。これが行われると、SPIデータレシ、スタ(SPDR)へのハイト書き込みがSPIクロック発生器を始動 し、ハート・ウェアが従装置内へ8ビットを移動します。1バイトの移動後、SPIクロック発生器は停止し、SPI状態レジスタ(SPSR)の転送完了フラグ (SPIF)を設定(1)します。SPI制御レジスタ(SPCR)でSPI割り込み許可(SPIE)ビットが設定(1)されていれば割り込みが要求されます。主装 置はSPDR内へ次バイトを書くことによって次バイトの移動を継続、またはSS(従装置選択)信号線をHighへ引き上げることによってパケット の終了を指示することができます。最後の到着バイトはその後の使用のため、緩衝レジスタ内に保持されます。

従装置として設定されると、SPIインターフェースはSSt°ンがHighに駆動される限り、MISOをHi-Zにした休止状態に留まります。この状態で プ゚ログラムはSPIデータ レジスタ(SPDR)の内容を更新できますが、そのデータはSSピンがLowに駆動されるまでSCKピンでの到着クロック パルス によって移動出力されません。1バイトが完全に移動されてしまうと転送完了フラグ(SPIF)が設定(1)されます。SPCRでSPI割り込み許可 (SPIE)ビットが設定(1)されていれば割り込みが要求されます。従装置は受信データを読む前にSPDR内へ送られるべき次のデータの配置 を続けられます。最後の到着バイはその後の使用のため、緩衝レジスタ内に保持されます。

このシステムは送信側で単一緩衝、受信側で2重緩 衝です。これは一連の移動全体が完了される前 に送信されるべきバイトがSPIデータ レジスタ(SPDR) へ書けないことを意味します。しかし、データを受 信するとき、次のデータが完全に移動入力される 前に受信したデータがSPIデータレジスタ(SPDR)から 読まれなければなりません。さもなければ始めの バイトは失われます。

SPI従装置動作では制御論理回路がSCKピンの到 着信号を採取します。このクロック信号の正しい採 取を保証するため、LowとHighの最小周期は以 下であるべきです。

- Low 周期: 2 CPU クロック 周期より長い - High周期: 2 CPUクロック周期より長い

SPIが許可されると、MOSI, MISO, SCK, SSt°ンの データ方向は表66.に従って無視されます。自動的 なポート無視のより多くの詳細については、40頁の 「交換ポート機能」を参照してください。

# 図66. SPI 主装置/従装置の連結



#### 表66. SPIL°ン方向規定

۲	じくと	主装置時の方向規定	従装置時の方向規定
S	SCK	ポートB方向レジスタ(DDRB)の指定	入力
N	MISO	入力	ポートB方向レジスタ(DDRB)の指定
N	MOSI	ポートB方向レジスタ(DDRB)の指定	入力
	SS	ポートB方向レジスタ(DDRB)の指定	入力

注: 使用者定義SPIピンの方向定義方法の詳細記述については42頁の「ポートB の交換機能」を参照してください。

次のコート例は主装置としてSPIを初期化する方法と簡単な送信を実行する方法を示します。例でのDDR\_SPIはSPIピンを制御する実際のポート方向レジスタに置き換えられなければなりません。DD\_MOSI, DD\_MISO, DD\_SCKはこれらのピンに対する実際のポート方向ビットに置き換えられなければなりません。例えばMOSIがPB5ピンに配置されるなら、DD\_MOSIはDDB5、DDR\_SPIはDDRBに置き換えます。

```
アセンブリ言語プログラム例
SPI_M_Init: LDI
                  R17, (1<<DD_MOSI) | (1<<DD_SCK)
                                                      ;MOSI, SCK=出力、他は入力値を取得
                  DDR_SPI, R17
                                                      ;MOSI, SCK=出力、他は入力に設定
            OUT
                  R17, (1<<SPE) | (1<<MSTR) | (1<<SPRO)
                                                      ;SPI許可、主装置、16分周値を取得
            LDI
            OUT
                  SPCR, R17
                                                      ;SPI許可、主装置、16分周に設定
            RET
                                                      ;呼び出し元へ復帰
SPI_M_Tx:
                                                      ; データ(R16) 送信開始
           OUT
                  SPDR, R16
SPI_M_Tx_W: SBIS
                  SPSR, SPIF
                                                      ;転送完了ならばスキップ
                                                      ;転送完了まで待機
            RJMP
                  SPI_M_Tx_W
                                                      ;呼び出し元へ復帰
            RET
C言語プログラム例
void SPI_MasterInit(void)
    DDR\_SPI = (1 << DD\_MOSI) | (1 << DD\_SCK);
                                                      /* MOSI, SCK=出力、他は入力に設定 */
    SPCR = (1 << SPE) \mid (1 << MSTR) \mid (1 << SPRO);
                                                      /* SPI許可、主装置、16分周に設定 */
void SPI_MasterTransmit(char cData)
                                                      /* データ送信開始 */
    SPDR = cData;
    while(!(SPSR & (1<<SPIF)));
                                                      /* 転送完了まで待機 */
```

#### 注: 5頁の「コート'例について」をご覧ください。

**注**: 5頁の「コード例について」をご覧ください。

次のコート・例は従装置としてSPIを初期化する方法と簡単な受信を実行する方法を示します。

```
アセンブリ言語プログラム例
SPI_S_Init: LDI
                  R17, (1<<DD_MISO)
                                                    ;MISO出力、他は入力値を取得
                                                    ;MISO出力、他は入力に設定
           OUT
                  DDR_SPI, R17
           LDI
                  R17, (1<<SPE)
                                                    ;SPI許可値を取得
           OUT
                  SPCR, R17
                                                    ;SPI許可設定
                                                    ;呼び出し元へ復帰
           RET
SPI_S_Rx:
           SBIS
                 SPSR, SPIF
                                                    ;受信(転送)完了ならばスキップ
                                                    ;受信(転送)完了まで待機
           RJMP
                 SPI_S_Rx
           IN
                  R16, SPDR
                                                    ;受信データを取得
                                                    ;呼び出し元へ復帰
           RET
C言語プログラム例
void SPI_SlaveInit(void)
   DDR SPI = (1 << DD MISO);
                                                    /* MISO出力、他は入力に設定 */
   SPCR = (1 << SPE);
                                                   /* SPI許可設定 */
char SPI_SlaveReceive(void)
   while(!(SPSR & (1<<SPIF)));
                                                   /* 受信(転送)完了まで待機 */
   return SPDR;
                                                    /* 受信データと共に復帰 */
```



## SSピンの機能

#### 従装置動作

SPIが従装置として設定されると、従装置選択(SS)ピンは常に入力です。SSがLowに保たれるとSPIは活性に(作動)され、使用者によってそのように設定されていればMISOは出力になります。他の全てのピンは入力です。SSがHighに駆動されると、出力として使用者設定され得るMISOを除く全てのピンは入力、SPIは非活動で、それは到着データを受信しないことを意味します。一旦SSピンがHighに駆動されると、SPI論理回路がリセットすることに注意してください。

このSSピンはパケット/バイト同期に対して、従装置ビット計数器が主装置クロック発生器との同期を保つのに有用です。SSピンがHighに駆動されると、SPI従装置は直ちに送受信論理回路をリセットし、それは移動レジスタ内で部分的に受信したどのデータも取り落とします。

#### 主装置動作

SPIが主装置(SPI制御レジスタ(SPCR)の主装置許可(MSTR)ビット=1)として設定されると、SSピンの方向は使用者が決められます。

SSが出力として設定されると、このピンはSPIシステムに影響を及ぼされない標準出力ピンです。代表的にはこのピンがSPI従装置のSSピンを駆動するでしょう。

SSが入力として設定されると、SPI主装置動作を保証するため、それはHighに保持されなければなりません。SSピンが入力として定義されたSPI主装置として設定される時に周辺回路によってSSピンがLowに駆動されると、SPIシステムは他の主装置が従装置として選んでデータ送信を始めると解釈します。バスの衝突を避けるためにSPIシステムは次の動作を行います。

- 1. SPCRで主/従選択(MSTR)ピットが解除(0)され、SPIシステムは従装置になります。SPIシステムが従装置になる結果としてMOSIとSCKピンが入力になります。
- 2. SPI状態レジスタ(SPSR)でSPI割り込み要求フラグ(SPIF)が設定(1)され、そしてSPI割り込みが許可(SPCRのSPIE=1)され、且つステータスレジスタ(SREG)の全割り込み許可(I)ビットが設定(1)なら、割り込みルーチンが実行されます。

従って割り込み駆動SPI送信が主装置動作で使われ、SSがLowに駆動される可能性があるとき、その割り込み(処理)はMSTRビットが未だ設定(1)されていることを常に検査すべきです。MSTRビットが従装置選択によって解除(0)されてしまっていると、それはSPI主装置動作を再び許可するため、使用者によって設定(1)されなければなりません。

## SPI制御レジスタ (SPI Control Register) SPCR

ピット	7	6	5	4	3	2	1	0	
\$2C (\$4C)	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	SPCR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

#### • ビット7 - SPIE: SPI割り込み許可 (SPI Interrupt Enable)

ステータス レジスタ(SREG)の全割り込み許可(I)ビットが設定(1)されて、SPI状態レジスタ(SPSR)でSPI割り込み要求フラグ(SPIF)が設定(1)されると、このビットがSPI割り込みを実行させます。

#### • ビット6 - SPE: SPI許可 (SPI Enable)

SPEビットが1を書かれるとSPIが許可されます。どのSPI操作を許可するにも、このビットが設定(1)されなければなりません。

#### • ビット5 - DORD : データ順選択 (Data Order)

DORDビットが1を書かれるとデータ語のLSBが最初に転送されます。DORDビットが0を書かれるとMSBが最初に転送されます。

#### • ビット4 - MSTR: 主装置/従装置選択 (Master/Slave Select)

このビットは1を書かれると主装置動作、論理0を書かれると従装置動作を選びます。 $\overline{SS}$ が入力として設定され、MSTRが設定(1)の間に Lowへ駆動されると、MSTRが解除(0)され、SPI状態レジスタ(SPSR)でSPI割り込み要求フラグ(SPIF)が設定(1)になります。その後に使用者はSPI主装置動作を再び許可するためにMSTRを設定(1)しなければなりません。

## • ビット3 - CPOL: SCK極性選択 (Clock Polarity)

このビットが1を書かれると、7仆、nはnはnにSCKはn1のです。n2のと、n3のです。n4のです。n5のについてはn4のです。n6のについてはn6のについてはn6のについてはn6のです。n6のについてはn6のです。n6のについてはn6のです。n6のについてはn6のです。n6のについてはn6のです。n6のについてはn6のです。n6のについてはn6のです。n6のについてはn6のです。n6のについてはn6のです。n6のについてはn6のです。n6のについてはn6のです。n6のについてはn6のです。n6のについてはn6のです。n6のについてはn6のです。n6のについてはn6のです。n6のについてはn6のです。n6のについてはn6のです。n6のについてはn6のです。n6のについてはn6のについいてはn6のについてはn6のについてはn6のについてはn6のについてはn6のについてはn6のについてはn6のについてはn6のについいてはn6のについいてはn6のについいてはn6のについいではn6のについいではn6のについいではn6のについいではn6のについいではn6のにのいいではn6のにのいいではn6のにのいいでのいなのではn6のにのいいでのいなのでなる。n6のにのいいでのいなのでなる。n6のにのいいでのいなのでなる。n6のにのいななる。n6のにのいなのでなる。n6のにのいなのでなる。n6のにのいなのでなる。n6のにのいなのでなる。n6のにのいなのでなる。n6のにのいなのでなる。n6のにのいなる。n6のにのいなる。n6のにのいななる。n6のにのいなる。n6のにのいなる。n6のにのいなる。n6のにのいなる。n6のにのいななる。n6のにのいななる。n6のにのいなる。n6のにのいななる。n6のにのいなる。n6のにのいなる。n6のにのいななる。n6のにのいなる。n6のにのいななる。n6のにのいななる。n6のにのいななる。n6のにのいななる。n6のにの

表67. CI	POL機能動作	
CPOL	SCK先行端	SCK後行端
0	上昇端	下降端
1	下降端	上昇端

#### • ビット2 - CPHA: SCK位相選択(Clock Phase)

このSCK位相選択(CPHA)ビットの設定はデータがSCKの先行(先)端または後行(後)端で採取/(設定)されるかを決めます。例については図67.と図68.を参照してください。CPH A機能は右で要約されます。

表68. CF	PHA機能動作	
СРНА	SCK先行端	SCK後行端
0	入力採取	出力設定
1	出力設定	入力採取

#### • ビット1.0 - SPR1.0: SPIクロック選択 (SPI Clock Rate Select 1 and 0)

これら2ビットは主装置として設定されたデバイスのSCK速度を制御します。従装置でのSPR1とSPR0は無効です。SCKと(システム)発振器クロック周波数fosc間の関連は次表で示されます。

表69. SCK速度選択(fosc=CPUクロック周波数)

<u> </u>									
SPR1	0		(	)	]		1		
SPR0	(	)		1	(	)	1		
SPI2X	1	0	1 0		1 0		1	0	
SCK周波数	fosc/2	fosc/4	fosc/8	fosc/16	fosc/32	fosc	c/64	fosc/128	

## SPI状態レジスタ (SPI Status Register) SPSR

ピット	7	6	5	4	3	2	1	0	
\$2D (\$4D)	SPIF	WCOL	-	-	-	-	-	SPI2X	SPSR
Read/Write	R	R	R	R	R	R	R	R/W	
初期値	0	0	0	0	0	0	0	0	

#### • ビット7 - SPIF: SPI割り込み要求フラグ(SPI Interrupt Flag)

直列転送が完了すると、このSPIFフラケが設定(1)されます。全割り込みが許可(ステータスレジスタ(SREG)の全割り込み許可(I)ビット=1)されて、SPI制御レジスタ(SPCR)でSPI割り込み許可(SPIE)ビットが設定(1)されるなら、割り込みが生成されます。SPIが主装置動作の時にSS ピッか入力でLowに駆動されるなら、これもこのSPIFフラケを同様に設定(1)します。対応する割り込み処理へクタを実行する時にSPIFはハートウェアによって解除(0)されます。代わりにSPIFが設定(1)されたSPI状態レジスタ(SPSR)を始めに読み、その後にSPIデータレジスタ(SPD R)をアクセスすることによってもSPIFフラケは解除(0)されます。

#### • ビット6 - WCOL : 上書き発生フラグ (Write Collision Flag)

データ転送中にSPIデータレシ、スタ(SPDR)が書かれると、このWCOLビットが設定(1)されます。WCOLビット(とSPIFビット)はWCOLが設定(1)されたSPI状態レシ、スタ(SPSR)を始めに読み、その後にSPIデータレシ、スタ(SPDR)をアクセスすることによって解除(0)されます。

#### • ビット5~1 - Res: 予約 (Reserved Bit)

これらのビットは予約されており、常に0として読みます。

#### • ビット0 - SPI2X : SPI倍速許可 (Double SPI Speed Bit)

このビットが論理1を書かれると、SPIが主装置動作の時にSCK速度(SCK周波数)が倍にされます(表69.参照)。これは最小SCK周期が2 CPUクロック周期であることを意味します。SPIが従装置として設定される時に、SPIはfosc(CPUクロック周波数)/4またはそれ以下での動作のみ保証されます。

ATmega169のSPIインターフェースはフラッシュ メモリやEEPROMの書き換え(読み書き)にも使われます。直列プログラミングと照合については185頁をご覧ください。

## SPIデータレジスタ (SPI Data Register) SPDR

ピット	7	6	5	4	3	2	1	0	_
\$2E (\$4E)	(MSB)							(LSB)	SPDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値	不定	不定	不定	不定	不定	不定	不定	不定	

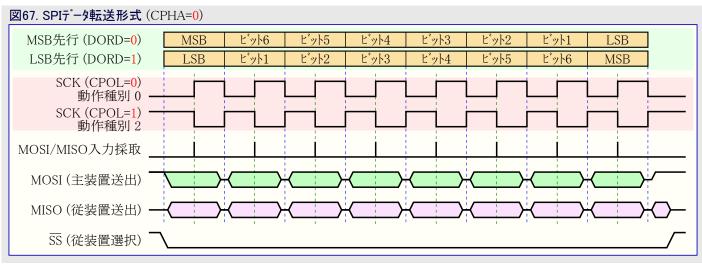
SPIデータレジスタはSPI移動レジスタとレジスタファイル(汎用レジスタ)間のデータ転送に使われる読み書き可能なレジスタです。このレジスタへの書き込みがデータ送信を開始します。このレジスタの読み込みは移動レジスタの受信緩衝部読み出しを引き起こします。

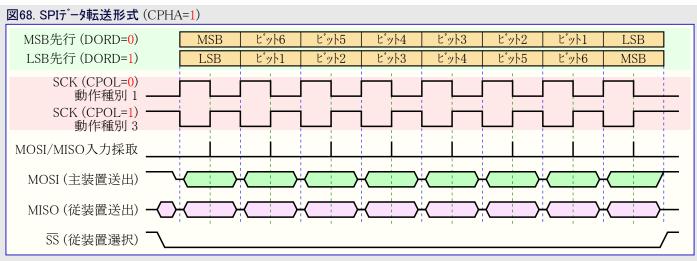


# データ転送形式

直列データに関してはSPI制御レシ、スタ(SPCR)のSCK位相(CPHA)とSCK極性(CPOL)制御ビットによって決定されるSCK位相と極性で4つの組み合わせがあります。このSPIデータ転送形式は図67.と図68.で示されます。データビットは安定のためデータ信号に対して充分な時間を保証するSCK信号の反対端で移動出力と(入力)ラッチが行われます。これは以下で行われるように表67.と表68.を要約することによって明解にされます。

表70. CPOL,CPHA機能動作 SPI動作種別番号 CPOL CPHA SCK先行端 SCK後行端 0 入力採取/上昇端 出力設定/下降端 0 0 出力設定/上昇端 入力採取/下降端 1 出力設定/上昇端 0 入力採取/下降端 3 出力設定/下降端 入力採取/上昇端





#### **USART**

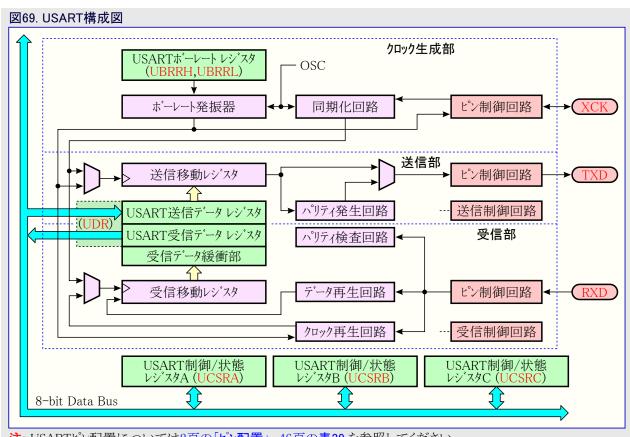
USART(Universal Synchronous and Asynchronous Receiver and Transmitter)は高い柔軟性をもつ直列通信機能です。主な特徴を 次に示します。

- 全二重動作(独立した送受信レジスタ)
- 同期または非同期動作
- 同期クロック駆動された主装置/従装置動作
- 高分解能ボーレート発振器
- 5, 6, 7, 8または9ビット データと1または2停止ビットの直列フレームの支援
- ハート・ウェアによって支援された奇数または偶数パリティの生成と検査
- データ オーバーラン検出
- フレーミング異常検出
- 不正開始ビット検出とデジタル低域通過濾波器を含む雑音濾波器
- 受信完了、送信完了、送信データレジスタ空きの3つの分離した割り込み
- 複数プロセッサ通信機能
- 倍速非同期通信動作

23頁の「電力削減レジスタ(PRR)」でのPRUSART0ビットはUSART部を許可するために0を書かれなければなりません。

#### 概要

USARTの簡単化した構成図は図69.で示されます。CPUがアクセス可能なレジスタとI/Oピンは赤文字(訳注:原文は太字)で示されます。



注: USARTピン配置については2頁の「ピン配置」、46頁の表39.を参照してください。

構成図内の破線はUSARTの3つの主要部分、(上から)クロック生成部、送信部、受信部を分けます。制御レジスタは全部によって共用されます。クロック生成論理部はボーレート発振器と同期従装置動作によって使われる外部クロック入力に対する同期化論理回路から成ります。転送クロック(XCK)ピンは同期転送動作だけで使われます。送信部は単一書き込み緩衝部(UDR)、直列移動レジスタ、ハプリティ発生器、異なる直列フレーム形式を扱うための制御論理回路から成ります。書き込み緩衝部はどんなフレーム間の遅れもなくデータの継続転送を許します。受信部はクロックとデータの再生部のため、USART部の最も複雑な部分です。再生部は非同期データ受信で使われます。再生部に加えて、受信部は2重の受信緩衝部(UDR)、移動レジスタ、ハプティ検査器、制御論理回路を含みます。受信部は送信部と同じフレーム形式を支援し、フレーシング異常、データオーハーラン発生、ハプティ誤りを検知できます。





#### UARTとの互換性

USARTはAVRのUARTと次の項目に関して完全な互換性があります。

- 全てのUSART側レジスタでのビット位置
- ボーレート生成
- 送信操作
- 送信緩衝の動作
- 受信操作

けれども受信緩衝動作にはいくつかの特別な場合で互換性に影響を及ぼす2つの改良点があります。

- 第2受信緩衝部が追加されました。2つのレシ、スタは循環型FIFO緩衝部として動作します。従ってUSARTデータレシ、スタ(UDR)は到着データ毎に一度だけ読まれなければなりません。より重要なのは異常フラグ(FEとDOR)と第9データ ビット(RXB8)が受信緩衝部内のデータと共に緩衝されることの事実です。従って状態ビットは常にUSARTデータレシ、スタ(UDR)が読まれる前に読まれなければなりません。さもなければ緩衝部の状態が失われるため、異常情報も失われます。
- 受信部移動レジスタは第3緩衝段のように動けます。これは緩衝レジスタが一杯の場合、新規開始ビットが検出されるまで直列移動レジスタ(図69.参照)内に留まるのを受信したデータに許すことによって行われます。従ってUSARTはデータオーバーラン(DOR)異常条件により耐えます。

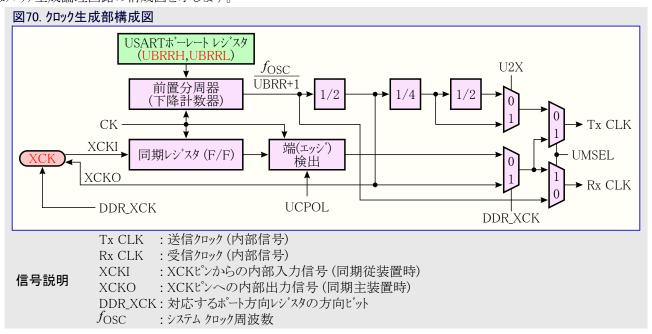
次の制御ビットは名称変更されましたが、機能とレジスタ(ビット)位置は同じです。

- 9ビット選択(CHR9)ビットはデータ長選択ビット2(UCSZ2)へ変更
- オーバーラン発生(OR)フラグはオーバーラン発生(DOR)フラグへ変更
- パリティ誤り(PE)フラケ゛はパリティ誤り(UPE)フラケ、へ変更(訳注:原書で欠落のため追加)

## クロック生成

クロック生成論理回路は送受信部用基準クロックを生成します。USARTは標準非同期、倍速非同期、同期主装置、同期従装置の4つのクロック動作種別を支援します。USART制御/状態レシ、スタC(UCSRC)のUSART動作種別選択(UMSEL)ビットは同期動作と非同期動作のどちらかを選びます。倍速動作(非同期動作のみ)はUSART制御/状態レシ、スタA(UCSRA)にある倍速許可(U2X)ビットによって制御されます。同期動作(UMSEL=1)を使うとき、XCKピンに対する方向制御ビット(DDR\_XCK)はクロック元が内部(主装置動作)または外部(従装置動作)のどちらかを制御します。このXCKピンは同期動作を使う時だけ活性(有効)です。

図70.はクロック生成論理回路の構成図を示します。



#### ボーレート発振器での内部クロック発生

内部クロック生成は非同期と同期主装置動作種別に対して使われます。本項の記述は図70.を参照してください。

USARTボーレートレジスタ(UBRR(UBRRH:UBRRL))と下降計数器は設定変更可能な前置分周器またはボーレート発振器として機能するように接続されます。システム クロック(fosc)で走行する下降計数器は0への下降計数時毎またはUBRRLレジスタが書かれる時にUBRR値で設定されます。1クロック(パルス)はカウンタが0に達する毎に生成されます。このクロックがボーレート発振器出力(=fosc/(UBRR+1))です。送信部は動作種別に依存してボーレート発振器出力を2,8,16分周します。ボーレート発振器出力は受信部クロックとデータ再生部によって直接使われます。しかし、再生部はUSART動作種別選択(UMSEL)、倍速許可(U2X)、DDR\_XCKビットの状態によって設定される動作種別に依存して2,8,16段を使う順次処理回路を使います。

表71.は内部的に生成したクロック元を使う各動作種別に於けるボーレート(bps)とUBRR値の計算式を含みます。

表71. ボーレート レジスタ(UBRR)値計算式		
動作種別	ボーレート計算式	UBRR値計算式
標準速非同期動作 (U2X=0)	$BAUD = \frac{f_{OSC}}{16 \times (UBRR+1)}$	$UBRR = \frac{f_{OSC}}{16 \times BAUD} - 1$
倍速非同期動作(U2X=1)	$BAUD = \frac{f_{OSC}}{8 \times (UBRR+1)}$	$UBRR = \frac{f_{OSC}}{8 \times BAUD} - 1$
同期主装置動作	$BAUD = \frac{f_{OSC}}{2 \times (UBRR+1)}$	$UBRR = \frac{f_{OSC}}{2 \times BAUD} - 1$

注: ボーレートは転送速度(ビット/1秒)で定義されます。

BAUD : ボーレート (bps)

UBRR: UBRRHとUBRRLレシブスタ値(0~4095)

fosc : システム発振器クロック周波数

いくつかのシステム クロック周波数に対するいくつかのUBRR値の例は表79.で得られます(120~121頁参照)。





#### 倍速動作(U2X)

転送速度はUSART制御/状態レジスタA(UCSRA)で倍速許可(U2X)ビットを設定(1)することによって倍にできます。このビットの設定は非同期動作に対してだけ有効です。同期動作を使うとき、このビットは0に設定してください。

このビットの設定(1)は事実上、非同期通信に対する転送速度を倍にするボーレート分周器の分周数を16から8に減じます。けれども受信部がデータ採取とクロック再生に対して半分の(16から8に減じた)採取数をこの場合だけ使い、従ってこの動作種別が使われる時はシステム クロックとボーレート設定の精度がより必要とされることに注意してください。送信部についての低下要因はありません。

#### 外部クロック

外部クロックは同期従装置動作種別によって使われます。本項の記述での詳細については図70.を参照してください。

XCKピンからの外部クロック入力は不確定レベル状態(メタステーブル)の機会を最少とするために同期化レジスタによって採取されます。同期化レジスタからの出力は送受信部で使い得るのに先立って端(エッジ検出器を通過しなければなりません。この処理手順が2 CPUクロック周期の遅延を持ち込み、このため最大外部XCKクロック周波数は次式によって制限されます。

$$f_{\text{XCK}} < \frac{f_{\text{OSC}}}{4}$$

foscがシステム クロック元の安定度に依存することに注意してください。従って周波数変動によるデータ消失の可能性を避けるため、いくらかの余裕分を追加することが推奨されます。

#### 同期クロック動作

同期動作が使われる(UMSEL=1)とき、XCKピンはクロック入力(従装置)またはクロック出力(主装置)のどちらかとして使われます。データ採取またはデータ変更とクロック端間の依存性は同じです。基本原則はデータ出力(TXD)が変更される端と反対のXCKクロック端でデータ入力(RXD)が採取されることです。

USART制御/状態レシ、スタC(UCSRC)のXCK極性(UCPO L)ビットはデータ採取とデータ変更に対してどちらのクロック端が使われるかを選びます。図71.で示されるようにUCPOLがのとき、データはXCKの上昇端で変更され、下降端で採取されます。UCPOLが設定(1)の場合、データはXCKの下降端で変更され、上昇端で採取されます。



#### フレーム形式

1つの直列フレームは複数のデータ ビットと同期ビット(開始ビット、停止ビット)、任意の異常検査用パリティ ビットで定義されます。USARTは有効なフレーム形式として以下の組み合わせ30種全てを受け入れます。

- 1 開始ビット
- 5, 6, 7, 8, 9 tinh Find
- 奇数または偶数パリティビット、またはなし
- 1 または 2 停止ビット

フレームは最下位データ ビット(LSB)が次に続く開始ビットで始まります。その後に次データ ビットが最後の最上位データ ビット(MSB)まで(最大)合計9ビット続きます。許可したなら、パリティ ビットがデータ ビットの後、停止ビットの前に挿入されます。完全なフレームが送信されると、新規フレームによって直ちに後続されるか、または通信線をアイトル状態(high)に設定できます。図72.は組み合わせ可能なフレーム形式を図解します。[]付きビットは任意選択です。



USARTによって使われるフレーム形式はUSART制御/状態レジスタBとC(UCSRB, UCSRC)でデータ長選択(UCSZ2~0)ビット、パリティ選択(UPM1,0)ビット、停止ビット選択(USBS)ビットによって設定されます。受信部と送信部は同じ設定を使います。これらのどのビットの設定変更も、送受信部両方に対して進行中の通信を不正とすることに注意してください。

データ長選択(UCSZ2~0)ビットはフレーム内のデータ ビット数を選びます。 パリティ選択(UPM1,0)ビットはパリティ ビットの許可と種別(奇/偶)を設定します。1または2停止ビットのどちらかの選択は停止ビット選択(USBS)ビットによって行います。受信部は第2停止ビットを無視します。 従ってフレーミング異常(FE)は最初の停止ビットが0(Low)の場合にだけ検出されます。

### パリティビットの計算

パリティ ビットは全データ ビットの排他的論理和(Ex-OR)を行うことによって計算されます。 奇数パリティが使われる場合は排他的論理和の結果が反転されます。 パリティ ビットとデータ ビットの関係は次のとおりです。

```
偶数パリティビット = D0 Ex-OR D1 Ex-OR D2 Ex-OR ~ Ex-OR Dn-1
奇数パリティビット = D0 Ex-OR D1 Ex-OR D2 Ex-OR ~ Ex-OR Dn-1 Ex-OR 1 n:データビット長
```

使った場合、パリティビットは直列フレームの最後のデータビットと最初の停止ビット間に配置されます。

### USARTの初期化

何れかの通信が行えるのに先立ってUSARTは初期化されなければなりません。標準的な初期化手順は使用方法に依存するボーレート設定、フレーム形式設定、送受信部許可から成ります。割り込み駆動USART操作に関して初期化を行うとき、ステータスレジスタの全割り込み許可(I)ピットは解除(0)される(そして全割り込みが禁止される)べきです。

ボーレートまたはフレーム形式の変更を伴う再初期化を行う前には、レシブスタが変更される期間中に進行中の送信がないことを確実にしてください。USART制御/状態レシブスタA(UCSRA)の送信完了(TXC)フラケ は送信部の全転送完了検査に使い、受信完了(RXC)フラケ は受信緩衝部内の未読データ有無検査に使えます。この目的(次送信タイミンケ)にTXCフラケが使われる場合、各々の送信(USARTデータレジブスタ(UDR)が書かれる)前にTXCフラケが解除(0)されなければならないことに注意してください。

次の簡単なUSART初期化コート例が示すアセンフリ言語とC言語の関数は機能的に同じです。この例は固定フレーム形式でポーリングを使う(割り込み不許可)非同期動作と仮定します。ボーレート(UBRR)値は関数の引数として与えられます。アセンブリ言語でのボーレート引数はR17:R16レジスタに格納されると仮定されます。

```
アセンブリ言語プログラム例
USART_Init: OUT
                   UBRRH, R17
                                                        ;ボーレート設定(上位バイト)
                   UBRRL, R16
                                                        ;ボーレート設定(下位バイト)
            OUT
            LDI
                   R16, (1<<USBS) | (3<<UCSZ0)
                                                        ;フレーム形式値を取得
                   UCSRC, R16
                                                        ;フレーム形式設定(8ビット,2停止ビット)
            OUT
            LDT
                   R16, (1<<RXEN) | (1<<TXEN)
                                                        ;送受信許可値を取得
            OUT
                   UCSRB, R16
                                                        ;送受信許可
                                                        ;呼び出し元へ復帰
            RET
C言語プログラム例
#define FOSC 1843200
                                                        /* MCUクロック周波数 */
#define BAUD 9600
                                                        /* 目的USARTボーレート速度 */
#define MYUBRR FOSC/16/BAUD-1
                                                        /* 目的UBRR值 */
void main(void)
USART_Init(MYUBRR);
                                                        /* USART初期化 */
void USART Init (unsigned int baud)
    UBRRH = (unsigned char) (baud>>8);
                                                        /* ボーレート設定(上位バイト) */
    UBRRL = (unsigned char)baud;
                                                        /* ボーレート設定(下位バイト) */
    UCSRC = (1 << USBS) | (3 << UCSZ0) :
                                                        /* フレーム形式設定(8ビット,2停止ビット) */
    UCSRB = (1 << RXEN) \mid (1 << TXEN);
                                                        /* 送受信許可 */
```

#### **注**: 5頁の「コート「例について」をご覧ください。

割り込みの禁止や引数としてフレーム形式を含めるなどで、より進化した初期化ルーチンが作成できます。けれども多くの応用はボーレートや制御レジスタの固定した設定が使われ、これらの応用形式での初期化コートは主ルーチンに直接置けるか、または他のI/Oの初期化コートと併せられます。



# USARTのデータ送信

USART送信部はUSART制御/状態レシ、スタB(UCSRB)で送信許可(TXEN)ヒットを設定(1)することによって許可されます。送信部が許可 されると、TXDピンの標準ピン動作はUSARTによって無視され、送信部の直列出力としての機能を与えられます。何か送信を行う前に 一度はボーレート、フレーム形式、動作種別が設定されなければなりません。同期動作が使われる場合、XCKピンの(受信)クロックは無視さ れ、送信クロックとして使われます。

### 5~8ビットデータフレーム送信

データ送信は送信されるべきデータを送信緩衝部に設定することによって開始されます。CPUはUSARTデータレジスタ(UDR)I/O位置へ 書くことによって送信緩衝部に設定できます。送信緩衝部内のデータは移動レジスタが新規フレームを送る準備が整った時に移動レジスタ へ移されます。移動レジスタはアイドル状態(送信進行中以外)、または直前のフレームの最後の停止ビット送信後、直ちに新規データが設定 されます。移動レシ、スタが新規データを設定されると、ホーレートレシ、スタ(UBRRH:UBRRL)と倍速許可(U2X)ビット、また動作種別によっては XCKピンによって与えられる速度で1つの完全なフレームを転送します。8ビット未満のフレームを使うとき、UDRに書かれた上位ビットは無視 されます。

次のコート・例はUSART制御/状態レジスタA(UCSRA)の送信データレジスタ空き(UDRE)フラケのポーリンケを基準とした簡単なUSART送信関 数を示します。この関数が使われ得る前にUSARTが初期化されなければなりません。アセンブリ言語での送るべきデータはR16レジスタに 格納されると仮定されます。

```
アセンブリ言語プログラム例
USART_Tx:
          SBIS
                 UCSRA, UDRE
                                                 ;送信緩衝部空きでスキップ
          RJMP
                 USART_Tx
                                                 ;送信緩衝部空き待機
          OUT
                 UDR, R16
                                                 ;データ送信(送信開始)
          RET
                                                  ;呼び出し元へ復帰
C言語プログラム例
void USART_Transmit(unsigned char data)
          while (!(UCSRA & (1<<UDRE)));
                                                 /* 送信緩衝部空き待機 */
          UDR = data;
                                                 /* データ送信(送信開始) */
注: 5頁の「コート・例について」をご覧ください。
```

この関数は送信されるべき新規データを設定する前に、UDREの検査によって送信緩衝部が空になるのを単純に待ちます。送信緩衝 部空き割り込みが使われる場合、その割り込み処理ルーチンがデータを緩衝部内に書きます。

### 9ビット データ フレーム送信

9ビット データが使われる場合(UCSZ2~0=111)、データの下位バイトがUSARTデータ レジスタ(UDR)に書かれる前に、第9ビットがUSART制御 /状態レジスタB(UCSRB)の送信データ ビット8(TXB8)ビットに書かれなければなりません。 次のコード例は9ビット データを扱う送信関数を示し ます。アセンブリ言語での送るべきデータはR17:R16レジスタに格納されると仮定されます。

```
アセンブリ言語プログラム例
USART Tx:
           SBIS
                  UCSRA, UDRE
                                                     ;送信緩衝部空きでスキップ。
           R.JMP
                  USART Tx
                                                     ;送信緩衝部空き待機
           CBT
                  UCSRB, TXB8
                                                     ;第9ビットを0に仮設定
           SBRC
                  R17.0
                                                     ;送信すべき第9ビットが0でスキップ
                  UCSRB, TXB8
                                                     ;第9ビットを1に設定
           SBT
                                                     ;データ送信(送信開始)
           OUT
                  UDR, R16
           RET
                                                     ;呼び出し元へ復帰
C言語プログラム例
void USART_Transmit(unsigned int data)
           while (!(UCSRA & (1<<UDRE)));
                                                     /* 送信緩衝部空き待機 */
           UCSRB &= ^{\sim} (1<<TXB8);
                                                     /* TXB8を0に仮設定 */
           if (data \& 0x0100) UCSRB = (1 << TXB8);
                                                     /* 第9ビットをR17からTXB8へ複写 */
           UDR = data;
                                                     /* データ送信(送信開始) */
```

注: これらの送信関数は一般的な機能で書かれています。UCSRBの内容が静的(換言すると、UCSRBのTXB8ビットが初期化後 に使われるだけ)ならば最適化できます。

5頁の「**コード例について**」をご覧ください。

第9ビットは複数プロセッサ通信使用時のアドレス フレーム識別、また例えば同期として扱う他の規約で使うことができます。

### 送信フラグと割り込み

USART送信部には状態を示す2つのフラグ、USARTデータレジスタ空き(UDRE)と送信完了(TXC)があります。両フラグは割り込みを発生するのに使えます。

USARTデータレジスタ空き(UDRE)フラグは送信緩衝部が新規データを受け取る準備ができているかどうかを示します。このビットは送信緩衝部が空の時に設定(1)され、送信緩衝部が移動レジスタに未だ移動されてしまっていない送信されるべきデータを含む時に解除(0)されます。将来のデバイスとの共通性のため、USART制御/状態レジスタA(UCSRA)に書くとき、常にこのビットに0を書いてください。

USART制御/状態レジスタB(UCSRB)でデータレジスタ空き割り込み許可(UDRIE)ビットが1を書かれると、(全割り込みが許可されているならば)UDREフラグが設定(1)されている限り、USARTデータレジスタ空き割り込みが実行されます。UDREはUSARTデータレジスタ(UDR)書き込みによって解除(0)されます。割り込み駆動データ送信が使われるとき、データレジスタ空き割り込みルーチンはUDREを解除(0)するために新規データをUDRに書くか、データレジスタ空き割り込みを禁止するかのどちらかを行わなければならず、さもなければ一旦割り込みルーチンを終了しても新しい割り込みが(継続的に)起こります。

送信完了(TXC)フラグは送信移動レシ、スタ内の完全なフレームが移動出力されてしまい、送信緩衝部に新規データが現在存在しない時に設定(1)されます。TXCフラグは送信完了割り込みが実行されるとき、自動的に解除(0)されるか、またはこのビット位置に1を書くことによっても解除(0)できます。TXCフラグは送信応用プログラムが送信完了後、直ちに通信回線を開放し、受信動作へ移行しなければならない(RS485規格のような)半二重(ハーフデュープレックス)通信インターフェースで有用です。

UCSRBで送信完了割り込み許可(TXCIE)ビットが設定(1)され、(全割り込みが許可されていれば)TXCフラグが設定(1)になる時にUSAR T送信完了割り込みが実行されます。送信完了割り込みが使われるとき、割り込み処理ルーチンはTXCフラグを解除(0)しなくてもよく、これは割り込みが実行されるとき、自動的に行われます。

#### パリティ発生器

パリティ発生器は直列フレーム データに対するパリティ ビットを計算します。 パリティ ビットが許可されると(UPM1=1)、送信部制御論理回路は送られているフレームの最終データ ビットと最初の停止ビット間にパリティ ビットを挿入します。

#### 送信の禁止

送信部の禁止(UCSRBのUSART送信許可(TXEN)ビット=0)は進行中と保留中の送信が完了される(換言すると、送信移動レジスタと送信緩衝レジスタが送信されるべきデータを含まない時)まで有効になりません。禁止されると、送信部はもはやTXDピン(の標準ピン機能)を無効にしません。

### USARTのデータ受信

USART受信部はUSART制御/状態レシ、スタB(UCSRB)で受信許可(RXEN)ビットに1を書くことによって許可されます。受信部が許可されると、RXDビンの標準ビン動作はUSARTによって無視され、受信部の直列入力としての機能を与えられます。何か直列受信が行われ得る前に一度はボーレート、フレーム形式、動作種別が設定されなければなりません。同期動作が使われる場合、XCKビンのクロックは転送クロックとして使われます。

### 5~8ビット データ フレーム受信

受信部は有効な開始ビットを検出する時にデータ受信を開始します。開始ビットに続く各ビットはボーレートまたはXCKクロックで採取され、フレームの最初の停止ピットが受信されるまで受信移動レジスタ内へ移動されます。第2停止ビットは受信部によって無視されます。最初の停止ピットが受信されると(換言すると、受信移動レジスタに完全なフレームが存在すると)、この移動レジスタの内容は受信緩衝部内へ移されます。受信緩衝部はUSARTデータレジスタ(UDR)I/O位置を読むことによって読めます。8ピット未満のフレームを使うとき、UDRから読むデータの上位ビットは0で覆われます。

次のコート・例はUSART制御/状態レシ、スタA(UCSRA)の受信完了(RXC)フラケのポーリンケを基にした簡単なUSART受信関数を示します。 この関数が使われ得る前にUSARTが初期化されなければなりません。

```
アセンブリ言語プログラム例
USART Rx:
           SBIS
                 UCSRA, RXC
                                                   ;受信完了でスキップ
           R TMP
                 USART Rx
                                                   ;受信完了待機
           IN
                 R16, UDR
                                                   ;受信データ取得
           RET
                                                   ;呼び出し元へ復帰
C言語プログラム例
unsigned char USART Receive (void)
   while (!(UCSRA & (1<<RXC)));
                                                   /* 受信完了待機 */
                                                   /* 受信データ取得 */
   return UDR;
注: 5頁の「コート'例について」をご覧ください。
```

この関数は緩衝部を読んで値を戻す前に、RXCフラグの検査によって受信緩衝部にデータが存在するのを単純に待ちます。





### 9ビット データ フレーム受信

9ビット データが使われる場合(UCSZ2~0=111)、USARTデータ レジスタ(UDR)から下位バイトを読む前に、第9ビットがUSART制御/状態レジ スタB(UCSRB)の受信データ ビット8(RXB8)ビットから読まれなければなりません。この規則はフレーミング異常(FE)、オーーバーラン発生(DOR)、パリティ誤り(UPE)状態フラグにも適用します。USART制御/状態レジスタA(UCSRA)から状態情報を読み、その後にUDRからデータを読んでください。UDR I/O位置を読むことが受信FIFO緩衝部の状態を切り替え、その結果、FIFO内に保管されるRXB8,FE,DOR,UPEビット全てが切り替わります。

次のコート、例は9ビットデータと状態ビット両方を扱う簡単なUSART受信関数を示します。

```
アセンブリ言語プログラム例
USART_Rx:
                  UCSRA, RXC
                                                        ;受信完了でスキップ
           SBIS
           RJMP
                  USART_Rx
                                                        ;受信完了待機
           IN
                  R18, UCSRA
                                                        ;状態フラグ取得
                  R17, UCSRB
           IN
                                                        ;受信第9ビット取得
                  R16, UDR
                                                        ;受信データ取得
           IN
                  R18, (1<<FE) | (1<<DOR) | (1<<UPE)
                                                        ;受信異常検査
           ANDI
           BREQ
                  USART Rx V
                                                        ;異常なしで分岐
                  R17, -1
                                                        ;異常で-1値設定
           LDI
                  R16, -1
           LDI
USART_Rx_V: LSR
                  R17
                                                        ;RXB8ビットをビット0位置へ移動
           ANDI
                  R17, $01
                                                        ;RXB8ビットのみ有効
           RET
                                                        ;呼び出し元へ復帰
C言語プログラム例
unsigned int USART_Receive(void)
   unsigned char status, resh, resl;
                                                        /* 一時変数定義 */
   while (!(UCSRA & (1<<RXC)));
                                                        /* 受信完了待機 */
   status = UCSRA;
                                                        /* 状態フラグ取得 */
   resh = UCSRB;
                                                        /* 受信第9ビット取得 */
   res1 = UDR;
                                                        /* 受信データ取得 */
   if (status & ((1<<FE) | (1<<DOR) | (1<<UPE)) ) return -1; /* 受信異常で-1値設定/復帰*/
   resh = (resh >> 1) \& 0x01;
                                                        /* RXB8ビットのみ有効最下位へ */
   return ((resh<<8) | resl);
                                                        /* 結果9ビット データ取得/復帰 */
```

#### 注: 5頁の「コート'例について」をご覧ください。

この受信関数例は何か評価を行う前に、全てのI/Oレジスタをレジスタ ファイルに読みます。これは読まれた緩衝部位置が可能な限り早く新規データを自由に受け入れできるため、最適な受信緩衝部利用になります。

### 受信完了フラグと割り込み

USART受信部には受信部の状態を示す1つのフラグがあります。

USART受信完了(RXC)フラグは未読データが受信緩衝部に存在するかを示します。このフラグは受信緩衝部に未読データが存在する時に1で、受信緩衝部が空の(換言すると、何も未読データを含まない)時に0です。受信部が禁止される場合(RXEN=0)、受信緩衝部が破棄され、その結果としてRXCフラグは0になります。

USART制御/状態レシ、スタB(UCSRB)でUSART受信完了割り込み許可(RXCIE)ビットが設定(1)されると、(全割り込みが許可されていれば)RXCフラケが設定(1)されている限り、USART受信完了割り込みが実行されます。割り込み駆動データ受信が使われるとき、受信完了割り込みルーチンはRXCフラケを解除(0)するためにUSARTデータレジ、スタ(UDR)から受信したデータを読まなければならず、さもなければ一旦割り込みルーチンを終了しても新しい割り込みが(継続的に)起きます。

### 受信異常フラグ

USART受信には3つの異常フラグ、フレーミング異常(FE)、データ オーハーラン発生(DOR)、パリティ誤り(UPE)があります。全てがUSART制御/状態レジ、スタA(UCSRA)を読むことによってアクセスできます。異常フラグに対する共通点は異常状態を示しているフレームと共に受信緩衝部に配置されることです。異常フラグが緩衝されるため、USARTデータレジ、スタ(UDR) I/O位置を読むことが緩衝部読み出し位置を切り替えるので、UCSRAは受信緩衝部(UDR)の前に読まれなければなりません。異常フラグに対するその他の共通点はソフトウェアがこのフラグ位置へ書き込みを行うことによって変更できないことです。しかし、将来のUSART(機能)実装の上位互換性のため、UCSRAが書かれるとき、全てのフラグは0に設定されなければなりません。異常フラグはどれも割り込みを生成できません。

フレーミング、異常(FE)フラグは受信緩衝部に格納された、次に読み込み可能なフレームの第1停止ビットの状態を示します。FEフラグは停止ビットが正しく(Highとして)読まれた時に0で、停止ビットが不正(Low)だった時にFEフラグは1です。このビットは同期外れ状態の検出、中断状態検出、規約での操作に使えます。受信部が最初(第1)以外の全停止ビットを無視するため、FEフラグはUSART制御/状態レジスタC(UCSRC)の停止ビット選択(USBS)ビット設定によって影響を及ぼされません。将来のデバイスとの共通性のため、UCSRAに書くとき、常にこのビットを0に設定してください。

データ オーハーラン発生(DOR)フラグは受信部緩衝部が一杯状態のためのデータ消失を示します。データ オーハーランは受信緩衝部(2フレーム)が一杯で、(次の)新規フレーム データが受信移動レジスタで待っており、(更に次の)新規開始ビットが検出される時に起きます。DORフラグが設定(1)なら、最後にUDRから読んだフレームと次にUDRから読むフレーム間で1つ以上の直列フレームが失われています。将来のデバイスとの共通性のため、UCSRAに書くとき、常にこのビットに0を書いてください。DORフラグは受信されたデータが移動レジスタから受信緩衝部へ正常に移動された時に解除(0)されます。

パリティ誤り(UPE)フラク は受信緩衝部内の次フレームで、受信時にパリティ異常があったことを示します。パリティ検査が許可されていない場合、UPEフラク は常に0が読めます。将来のデバイスとの共通性のため、UCSRAに書くとき、常にこのビットを0に設定してください。より多くの詳細については109頁の「パリティ ビットの計算」と次の「パリティ検査器」をご覧ください。

#### パリティ検査器

ハプリティ検査器はハプリティ種別上位ビット(UPM1)が設定(1)されると活性(有効)になります。実行されるべきハプリティ検査の形式(偶数または奇数)はUPM0ビットによって選ばれます。許可されると、ハプリティ検査器は到着フレーム内のデータ ビットのハプリティを計算し、その結果と(受信)直列フレーム内のハプリティ ビットを比較します。検査の結果は受信データ、停止ビットと共に受信緩衝部に格納されます。その後ハプリティ誤り(UPE)フラグはフレームにハプリティ異常があるかを検査するため、ソフトウェアによって読むことができます。

UPEフラグは受信緩衝部から読まれ得る次フレームで受信時にパリティ異常があり、その時点でパリティ検査が許可されていた(UPM1=1)場合に設定(1)されます。このビットはUSARTデータレジスタ(UDR)が読まれるまで有効です。

#### 受信の禁止

送信部と対照的に受信部の禁止は即時です。従って受信進行中のデータは失われます。禁止されると(換言すると、USART制御/状態レジスタB(UCSRB)のUSART受信許可(RXEN)ビットが0に設定)、受信部はもはやRXDポート ピンの標準機能を無効にしません。受信FIFO緩衝部は受信部が禁止されると破棄されます。緩衝部内の残データは失われます。

### 受信緩衝部の破棄

受信FIFO緩衝部は受信が禁止されると破棄(換言すると、緩衝部は内容を空に)されます。未読データは失われます。例えば異常状態のため、通常動作中に緩衝部が破棄されなければならない場合、USART制御/状態レジスタA(UCSRA)の受信完了(RXC)フラケが解除(0)されるまでUSARTデータレシ、スタ(UDR) I/O位置を読んでください。次のコート例は受信緩衝部の破棄方法を示します。

```
アセンブリ言語プログラム例
USART Flush: SBIS
                                                   ;未読データありでスキップ。
                  UCSRA, RXC
            RET
                                                   ;未読データなしで復帰
                                                   ;データ受信
            IN
                  R16, UDR
                                                   ;未読データなしまで継続
            RJMP
                  USART_Flush
C言語プログラム例
void USART Flush (void)
   unsigned char dummy;
                                                   /* 一時変数定義 */
   while ( UCSRA & (1<<RXC) ) dummy=UDR;
                                                   /* 未読データ読み捨て */
注: 5頁の「コート・例について」をご覧ください。
```





# 非同期受信

USARTは非同期データ受信を扱うためのクロック再生とデータ再生部を含みます。クロック再生論理回路は内部的に生成したボーレートクロックをRXDピンに到着する非同期直列フレームに同期化するのに使われます。データ再生論理回路は到着ビット毎に低域通過の濾波と採取をし、それによって受信部の雑音耐性を改善します。非同期受信動作範囲は内部ボーレートクロック精度、到着フレーム速度、フレーム長のビット数に依存します。

### 非同期クロック再生

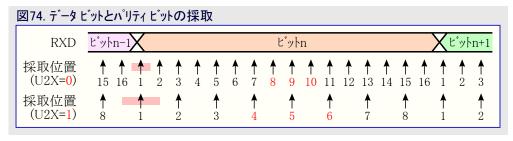
クロック再生論理回路は内部クロックを到着直列フレームに同期化します。**図73.**は到着フレームの開始ビット採取手順を図解します。採取速度は標準速動作でボーレートの16倍、倍速動作で8倍です。赤帯(<mark>訳注</mark>:原文は水平矢印)は採取処理のための同期変量を図示します。倍速動作(U2X=1)を使う時の広い変量時間に注意してください。採取番号0はRXD信号がアイドル(換言すると、通信の動きなし)の時に行われる採取です。



クロック再生論理回路がRXD信号線でHigh(アイドル)からLow(開始)の遷移を検出すると、開始ビット検出手順が開始されます。図で示されるように採取1は最初のLow採取を意味します。その後にクロック再生回路は有効な開始ピットが受信されるかを決めるために、標準速動作に対して採取8,9,10、倍速動作に対して採取4,5,6(図の赤数字(訳注:原文は箱枠内)で示された採取番号)を使います。これら3回で2回以上の採取が論理Highレベルの場合(多数決)、この開始ビットは尖頭雑音として捨てられ、受信部は次のHighからLowの遷移を探し始めます。けれども有効な開始ビットが検出されると、クロック再生論理回路は同期化され、データ再生が開始されます。この同期化手順は各開始ビット毎に繰り返されます。

#### 非同期データ再生

受信部クロックが開始ビットに同期化されるとデータ再生が始められます。データ再生部は標準速動作で16段、倍速動作で8段の順列回路を使います。図74.はデータ ビットとパリティ ビットの採取を示します。各採取は再生部の各段に等しい番号を与えられます。



受信したビットの論理値の決定は受信したビット中央の3採取で論理値の多数決を取ることによって行われます。この中央の3採取は図上の赤文字(訳注:原文は箱枠内)の採取番号によって強調されます。多数決の手順は次のように行われます。2または全3採取がHighレヘルならば受信したビットは論理1が記録されます。2または全3採取がLowレヘルならば受信したビットは論理0が記録されます。この多数決手順はRXDピンの到着信号に対して低域通過濾波器( $p-n^2$ スフィルタ)のように働きます。この再生手順はその後に完全なフレームが受信されるまで繰り返されます。これには最初の(第1)停止ビットを含みます。受信部がフレームの最初の停止ビットだけを使うことに注意してください。

図75.は停止ビットの採取と次フレームの最も早い開始ビット開始の可能性を示します。



フレーム内の他のビットに対して行われるのと同じ多数決が停止ビットにも行われます。停止ビットが論理0値と記録されると、フレーミング異常(FE)フラグが設定(1)されます。

新規フレームの開始ビットを示すHighからLowへの遷移は多数決に使った最後のビット後に実現できます。標準速動作での最初のLowレヘル採取は図75.のA点で有り得ます。倍速動作での最初のLowレヘルはB点に遅れます。C点は完全な長さの停止ビット(の終点)を示します。この早い開始ビット検出は受信部の動作範囲に影響します。

### 非同期での動作範囲

受信部の動作範囲は受信したビット速度と内部的に生成したボーレート間の不一致に依存します。送信部が速すぎるまたは遅すぎるビット速度でフレームを送出したり、内部的に発生した受信部のボーレートが類似した(表72.参照)基準周波数を持たない場合、受信部は開始ビットでフレームを同期できません。

次式は到着データ速度と内部受信部ボーレート間の比率計算に使えます。

 $R_{slow} = \frac{(D+1)\times S}{S-1+D\times S+S_F}$   $R_{fast} = \frac{(D+2)\times S}{(D+1)\times S+S_M}$ 

D: データとパリティのビット数 (5~10)

S: ビットあたりの採取数(標準速=16、倍速=8)

S<sub>F</sub> : 多数決に使う最初の採取番号 (標準速=8、倍速=4) S<sub>M</sub> : 多数決に使う中心の採取番号 (標準速=9、倍速=5)

R<sub>slow</sub>:は受信側ボーレートに対して許容できる最低受信ビット速度の比率です。 R<sub>fast</sub>:は受信側ボーレートに対して許容できる最高受信ビット速度の比率です。

表72.は許容できる最大受信部ボーレート誤差一覧です。標準速動作には、より高いボーレート変動許容力があることに注目してください。

#### 表72. 標準速と倍速での受信部ボーレート推奨最大許容誤差

24.1	(7.5 体中是自己是 50 人口的小 10 个正人联大时日龄是											
D		標	<b>準速動作</b> (U2X= <mark>0</mark> )			f	<b>音速動作</b> (U2X=1)					
U	R <sub>slow</sub> (%)	Rfast(%)	総合許容誤差(%)	推奨許容誤差(%)	R <sub>slow</sub> (%)	Rfast(%)	総合許容誤差(%)	推奨許容誤差(%)				
5	93.20	106.67	-6.80~+6.67	±3.0	94.12	105.66	-5.88~+5.66	±2.5				
6	94.12	105.79	-5.88~+5.79	±2.5	94.92	104.92	-5.08~+4.92	±2.0				
7	94.81	105.11	-5.19~+5.11	±2.0	95.52	104.35	-4.48~+4.35	±1.5				
8	95.36	104.58	-4.54~+4.58	±2.0	96.00	103.90	-4.00~+3.90	±1.5				
9	95.81	104.14	-4.19~+4.14	±1.5	96.39	103.53	-3.61~+3.53	±1.5				
10	96.17	103.78	-3.83~+3.78	±1.5	96.70	103.23	-3.30~+3.23	±1.0				

注: Dはデータ ビット数とパリティ ビットの合計ビット数です。

(訳注) 原書は表72.に標準速、表73.に倍速を記載していますが、比較が容易なように表72.として纏めました。

受信部ボーレートの推奨最大許容誤差は最大総合許容誤差を送信部と受信部で等分割するという仮定の元で作られました。

受信部ボーレート誤差に対して2つの起こり得る原因があります。受信部のシステム クロック(XTAL)は供給電圧範囲と温度範囲に関して常に若干の不安定性があります。システム クロックを生成するのにクリスタル発振子を使う時は殆ど問題ありませんが、(セラミック)振動子でのシステム クロックは振動子偏差に依存して2%を越えて異なるかもしれません。2つ目の誤り原因はより制御可能です。ボーレート発振器は欲したボーレートを得るためにシステム周波数の正確な分周を常に行うことはできません。この場合、可能ならば受け入れ可能な低い誤差を与えるUBRR値が使えます。





### 複数プロセッサ通信動作

USART制御/状態レジスタA(UCSRA)での複数プロセッサ通信動作(MPCM)ビットの設定(1)はUSART受信部によって受信された到着フレームの選別機能を許可します。アドレス情報を含まないフレームは無視され、受信緩衝部に格納されません。これは同一直列バス経由で通信する複数MCUのシステムで、CPUによって扱われなければならない到着フレーム数を効果的に減らします。送信部はMPCMビット設定によって影響されませんが、複数プロセッサ通信動作を利用するシステムの一部の時は違うふうに使われなければなりません。

受信部が5~8データ ビットを含むフレームを受信するように設定されるなら、最初の停止ビットはデータまたはアドレス情報を含むフレームかどうかを示します。受信部が9データ ビットのフレームに設定されるなら、USART制御/状態レジスタB(UCSRB)の受信第9(RXB8)ビットがアドレスとデータのフレームを識別するのに使われます。フレーム種別(最初の停止または第9)ビットが1の時にフレームはアドレスを含みます。フレーム種別ビットが0の時にそのフレームはデータフレームです。

複数プロセッサ通信動作は主MCUからのデータを多くの従MCUで受信することを可能にします。これはどのMCUがアトレス指定されるかを検出するため、最初にアトレスフレームを調べることによって行われます。特定の従MCUがアトレス指定されたなら、そのMCUは後続するデータフレームを通常のように受信し、一方その他の従MCUは他のアトレスフレームが受信されるまで受信したフレームを無視します。

### 複数プロセッサ通信の使用法

主MCUとして動作するMCUは9ビット データ フレーム形式(UCSZ=7)を使えます。UCSRBの送信第9(TXB8)ビットはアドレス フレーム時に設定 (1)、またはデータ フレーム時に解除(0)されて送信されなければなりません。この場合、従MCUは9ビット データ フレーム形式の使用に設定されなければなりません。

複数プロセッサ通信動作でデータを交換するのに次の手順が使われるべきです。

- 1. 全ての従MCUは複数プロセッサ通信動作です(UCSRAの複数プロセッサ通信動作(MPCM)ビットが設定(1))。
- 2. 主MCUはアドレス フレームを送り、全ての従装置がこのフレームを受信し、これを読みます。従CPUでは通常のようにUCSRAで受信完了(RXC)フラグが設定(1)されます。
- 3. 各従MCUはUSARTデータレジ、スタ(UDR)を読み、選ばれたかを判定します。選ばれた場合はUCSRAのMPCMビットを解除(0)し、そうでなければ(非選択の場合は)MPCMビット設定を保ち、次のアドレス フレームを待ちます。
- 4. アト・レス指定されたMCUは新規アト・レス フレームが受信されるまで全データ フレームを受信します。MPCMビットが未だ設定(1)されている他の従CPUはこのデータ フレームを無視します。
- 5. 最後のデータフレームがアドレス指定されたMCUによって受信されると、アドレス指定されたMCUはMPCMビットを設定(1)し、主装置からの新規アドレスフレームを待ちます。以降、手順は2.からを繰り返します。

5~8ビット データ フレーム形式のどの使用も可能ですが、受信側が使うnとn+1ビット データ フレーム形式間を切り替えなければならないため 非実用的です。これは送信部と受信部が同じデータ ビット長設定を使うため、全二重(フルデュープレックス)動作を困難にします。5~8ビット データ フレームが使われる場合、最初の停止ビットがフレーム種別を示すのに使われるので、送信部は2停止ビット使用(USBS=1)に設定されなければなりません。

MPCMビットを設定(1)または解除(0)するのに読み-修正-書き(リード モデファイ ライト)命令(SBIとCBI)を使ってはいけません。MPCMビットは送信完了(TXC)フラグと同じI/O位置を共用しており、SBIまたはCBI命令を使うと偶然に解除(0)されるかもしれません。

(訳注) ATmega169ではUCSRnAのI/OアドレスがSBI,CBI命令適用範囲外のため、上記記述は不適切です。 但し、命令の組み合わせによって同様処理を行う場合に対して、上記注意の本意は適切(有効)です。

### USART用レジスタ

#### USARTデータレシ スタ (USART I/O Data Register) UDR

ピット	7	6	5	4	3	2	1	0	
(\$C6)	R/TXB7	R/TXB6	R/TXB5	R/TXB4	R/TXB3	R/TXB2	R/TXB1	R/TXB0	UDR
Read/Write	R/W								
初期値	0	0	0	0	0	0	0	0	

同じI/Oアトレスを共用するUSART受信データ緩衝レシ、スタとUSART送信データ緩衝レシ、スタはUSARTデータレジ、スタまたはUDRとして引用しました。送信データ緩衝レシ、スタ(TXB)はUDRレシ、スタ位置に書かれるデータの転送先です。UDRレシ、スタ位置読み込みは受信データ緩衝レシ、スタ(RXB)の内容を返します。

5~7ビット データでの上位未使用ビットは送信部によって無視され、受信部によって0に設定されます。

送信緩衝部はUCSRAで送信データレジ、スタ空き(UDRE)フラグが設定(1)される時にだけ書けます。UDREフラグが設定(1)されない時にUDRへ書かれたデータはUSART送信部によって無視されます。送信部が許可されて送信緩衝部にデータが書かれると、送信部は送信移動レジ、スタが空きの時にそのデータを送信移動レジ、スタへ設定(移動)します。その後にデータはTXDピンで直列送信されます。

受信緩衝部は2段のFIFOから成ります。このFIFOは受信緩衝部がアクセスされる毎にその状態を切り替えます。この受信緩衝部の動きのため、この位置に読み-修正-書き(リート・モデファイライト)命令(SBIとCBI)を使えません。ビット検査(SBICとSBIS)命令もFIFOの状態を換えるので、これらの命令を使う時は注意してください。

#### USART制御/状態レジスタA (USART Control and Status Register A) UCSRA

		9	4	3	2	1	0	
(\$C0) RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM	UCSRA
Read/Write R	R/W	R	R	R	R	R/W	R/W	
初期値 0	0	1	0	0	0	0	0	

#### • ビット7 - RXC: USART受信完了フラグ (USART Receive Complete)

このフラグは受信緩衝部に未読データがある時に設定(1)され、受信緩衝部が空の(換言すると、どんな未読データも含まない)時に解除 (0)されます。受信部が禁止されると、受信緩衝部が破棄され、その結果RXCフラグは0になります。RXCフラグは受信完了割り込みを発生するのに使えます(USART制御/状態レジスタB(UCSRB)の受信完了割り込み許可(RXCIE)ビットをご覧ください)。

#### • ビット6 - TXC : USART送信完了フラグ (USART Transmit Complete)

このフラグは送信移動レジスタ内の完全なフレームが移動出力されてしまい、送信緩衝部(UDR)に新規データが現状存在しない時に設定 (1)されます。TXCフラグは送信完了割り込みが実行されるとき、自動的に解除(0)されるか、またはこのビット位置に1を書くことによって解除(0)できます。TXCフラグは送信完了割り込みを発生できます(UCSRBの送信完了割り込み許可(TXCIE)ビットをご覧ください)。

#### • ビット5 - UDRE: USART送信データ レジスタ空きフラグ(USART Data Register Empty)

UDREフラグは送信緩衝部(UDR)が新規データを受け取る準備ができているかどうかを示します。UDREが1ならば緩衝部は空で、従って書かれる準備ができています。UDREフラグは送信緩衝部空き割り込みを発生できます(UCSRBの送信データレジスタ空き割り込み許可(UDRIE)とずりをご覧ください)。送信部が準備できているのを示すため、リセット後のUDREは設定(1)です。

#### • ビット4 - FE : フレーミング異常フラグ(Framing Error)

受信緩衝部の次データが受信した時にフレーミング異常(換言すると、受信緩衝部で次データの最初の(第1)停止ビットがLow)だった場合、このビットが設定(1)されます。このフラグは受信緩衝部(UDR)が読まれるまで有効です。受信したデータの停止ビットがHighの時にFEフラグは0です。UCSRAに書くとき、常にこのビットを0に設定してください。

### • ビット3 - DOR: データ オーバーラン発生フラグ (Data OverRun)

このビットはオーハーラン状態が検出されると設定(1)されます。受信緩衝部(2フレーム分)が一杯で、新規フレームが受信移動レジスタ内で待機中に新規開始ビットが検出されるとデータオーハーランが起こります。UCSRAに書くとき、常にこのビットを0に設定してください。

### • ビット2 - UPE : パリティ誤りフラグ (USART Parity Error)

受信緩衝部の次データが受信した時にパリティ異常があり、その時点でパリティ検査が許可されていれば(UPM1=1)、このビットが設定(1)されます。このフラグは受信緩衝部(UDR)が読まれるまで有効です。UCSRAに書くとき、常にこのビットを0に設定してください。

#### • ビット1 - U2X: 倍速許可 (Double the USART Transmission Speed)

このビットは非同期動作でだけ有効です。同期動作を使うとき、このビットに0を書いてください。

このビットに1を書くことはボーレート分周器の分周値を16から8に減らして事実上、非同期通信の転送速度を倍にします。

### • ビット0 - MPCM:複数プロセッサ通信動作 (Multi-Processor Communication Mode)

このビットは複数プロセッサ通信動作を許可します。MPCMビットが1を書かれると、USART受信部で受信したアドレス情報を含まない全到着フレームは無視されます。送信部はMPCM設定に影響されません。より多くの詳細情報については116頁の「複数プロセッサ通信動作」をご覧ください。





### USART制御/状態レジスタB (USART Control and Status Register B) UCSRB

ピット	7	6	5	4	3	2	1	0	_
(\$C1)	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8	UCSRB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	
初期値	0	0	0	0	0	0	0	0	

#### • ビット7 - RXCIE: 受信完了割り込み許可 (Receive Complete Interrupt Enable)

このビットに1を書くことはUSART制御/状態レジスタA(UCSRA)の受信完了(RXC)フラグでの割り込みを許可します。USART受信完了割り込みはRXCIEビットが1を書かれ、ステータス レジスタ(SREG)で全割り込み許可(I)ビットが1を書かれ、UCSRAでRXCフラグが設定(1)される場合にだけ生成されます。

#### • ビット6 - TXCIE: 送信完了割り込み許可 (Transmit Complete Interrupt Enable)

このビットに1を書くことはUSART制御/状態レジスタA(UCSRA)の送信完了(TXC)フラグでの割り込みを許可します。USART送信完了割り込みはTXCIEビットが1を書かれ、SREGで全割り込み許可(I)ビットが1を書かれ、UCSRAでTXCフラグが設定(1)される場合にだけ生成されます。

#### • ビット5 - UDRIE: 送信データレジスタ空き割り込み許可 (Transmit Data Register Empty Interrupt Enable)

このビットに1を書くことはUSART制御/状態レジスタA(UCSRA)の送信データ レジスタ空き(UDRE)フラグでの割り込みを許可します。USART 送信データ レジスタ空き割り込みはUDRIEビットが1を書かれ、SREGで全割り込み許可(I)ビットが1を書かれ、UCSRAでUDREフラグが設定(1)される場合にだけ生成されます。

### • ビット4 - RXEN: 受信許可 (Receiver Enable)

このビットに1を書くことはUSART受信(部)を許可します。受信部は許可されるとRXDピンの標準ポート動作を無効にします。受信(部)を禁止することは受信緩衝部を破棄し、フレーミング異常(FE)、オーバーラン(DOR)、パリティ誤り(UPE)のフラグを無効にします。

#### • ビット3 - TXEN: 送信許可 (Transmitter Enable)

このビットに1を書くことはUSART送信(部)を許可します。送信部は許可されるとTXDピンの標準ポート動作を無効にします。送信(部)の禁止(TXEN=0書き込み)は進行中と保留中の送信が完了される(換言すると、送信移動レジスタと送信緩衝レジスタが送信されるべきデータを含まない)まで有効になりません。禁止したとき、送信部はもはやTXDポート(の標準I/O機能)を無効にしません。

#### • ビット2 - UCSZ2: データ ビット長選択2 (Character Size)

USART制御/状態レシ、スタC(UCSRC)のUCSZ1,0ビットと組み合わせたUCSZ2ビットは送受信部で使うフレームのデータ ビット数(Character size)を設定します。

### • ビット1 - RXB8 : 受信データ ビット8 (Receive Data Bit 8)

RXB8は9ビット データでの直列フレーム操作時に受信したフレームの第9データ ビット(ビット8)です。UDRから下位ビットを読む前に読んでください。

#### • ビット0 - TXB8 : 送信データ ビット8 (Transmit Data Bit 8)

TXB8は9ビット データでの直列フレーム操作時に送信されるべきデータの第9データ ビット(ピット8)です。UDRへ下位ビットを書く前に書いてください。

#### USART制御/状態レジスタC (USART Control and Status Register C) UCSRC

ピット	7	6	5	4	3	2	1	0	
(\$C2)	_	UMSEL	UPM1	UPM0	USBS	UCSZ1	UCSZ0	UCPOL	UCSRC
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	1	1	0	

#### • **ビット7 - Res**: **予約** (Reserved) (**訳注**:共通性のため本項追加)

このビットは将来使用するために予約されています。将来のデバイスとの共通性を保つため、UCSRCが書かれるとき、このビットは<mark>0</mark>が書かれなければなりません。

#### • ビット6 - UMSEL: USART動作選択 (USART Mode Select)

このビットは非同期と同期の動作種別のどちらかを選びます。

表74. USART動作	表74. USART動作選択							
UMSEL	動作種別							
0	非同期動作							
1	同期動作							

### • ビット5,4 - UPM1,0: パリティ選択 (Parity Mode)

これらのビットはパリティの発生と検査の許可と種別を設定します。許可した場合、送信部は各フレーム内での送信したデータ ビットのパリティを自動的に生成して送出します。受信部は到着データからパリティ値を生成し、UPMo設定と比較します。不一致が検出されると、USART制御/状態レジスタA(UCSRA)でパリティ誤り(UPE)フラグが設定(1)されます。

表75. パリティ選択									
UPM0	パリティ動作								
0	禁止								
1	(予約)								
0	偶数パリティ許可								
1	奇数パリティ許可								

#### • ビット3 - USBS : 停止ビット選択 (Stop Bit Select)

このビットは送信部によって挿入される停止ビット数を選びます。受信部はこの設定を無視します(訳補:常に第1停止ビットだけが有効)。

表76. 停止ビット選択	₹
USBS	停止ビット数
0	1ピット
1	2t*y}

### • ビット2,1 - UCSZ1,0: データ ビット長選択 (Character Size)

USART制御/状態レシ、スタB(UCSRB)のUCSZ2ビットと組み合わせたUCSZ1,0ビットは送受信部で使うフレームのデータ ビット数(Character size)を設定します。

+	- " -		· = 255.TU
表 / /	T -4	- 11	卜長選択
4X / /.	, ,		1.18 122.11

ď	ACCOUNT OF BRIDE										
L	UCSZ2∼0	0 0 0	0 0 1	0 1 0	0 1 1	100	101	1 1 0	1 1 1		
	データビット数	5ピット	6Ľ'yŀ	7ピット	8Ľ'yŀ	(予約)	(予約)	(予約)	9ビット		

#### • ビット0 - UCPOL: クロック極性選択 (Clock Polarity)

このビットは同期動作に対してだけ使われます。非同期動作が使われるとき、このビットに0を書いてください。UCPOLビットは同期クロック(XCK)、データ出力変更、データ入力採取間の関係を設定します。

表78. XCKクロック極性選択										
UCPOL	送信データ変更(TXDピン出力)	受信データ採取(RXDピン入力)								
0	XCKの上昇端	XCKの下降端								
1	XCKの下降端	XCKの上昇端								

#### USARTボーレート レジスタ (USART Baud Rate Register) UBRRH, UBRRL (UBRR)

ピット	15	14	13	12	11	10	9	8	
(\$C5)	-	-	-	-	UBRR11	UBRR10	UBRR9	UBRR8	UBRRH
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	
ピット	7	6	5	4	3	2	1	0	
(\$C4)	UBRR7	UBRR6	UBRR5	UBRR4	UBRR3	UBRR2	UBRR1	UBRR0	UBRRL
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

### • ビット15~12 - Res: 予約 (Reserved Bits)

これらのビットは将来使用するために予約されています。将来のデバイスとの共通性のため、UBRRHが書かれるとき、これらのビットはのが書かれなければなりません。

#### • ビット11~0 - UBRR11~0: ボーレート分周値 (USART Baud Rate Register)

これはUSARTのボーレートを含む12ビット レジスタです。UBRRHがUSARTボーレートの上位4ビットを含み、UBRRLが下位8ビットを含みます。ボーレートが変更されると、送受信部で進行中の転送は不正にされます。UBRRL書き込みはボーレート前置分周器の更新を直ちに始めます。





## ボーレート設定例

標準的なクリスタル発振子やセラミック振動子の周波数に対して最も一般的に使われる非同期動作のボーレートは表79.のUBRR設定を使うことによって生成できます。目的のボーレートに対して設定誤差1.5%(標準速:U2X=0)、1.0%(倍速:U2X=1)以上を赤字で示します(訳注:原書の本文は0.5%未満を太字)。より高い誤差率でも受け入れ可能ですが、特に長い直列フレーム(115頁の「非同期での動作範囲」参照)で誤差率が高いと、受信部は雑音耐性が低下します。誤差率は次式を使って計算されます。

誤差率(%) = ( <u>UBRR設定ボーレート(最近似値)</u> - 1) ×100(%)

表79. Xtal、ボーレート対UBRRH.UBRRL設定(UBRR=UBRRH:UBRRL)

12,70. AC	<u> </u>		Hz				2MHz	3 DI (I (L)		2M	lHz			2 457	6MHz	
ホーレート	U2	X= <mark>0</mark>		!X=1	U2	X= <mark>0</mark>		2X=1	U2	2X= <mark>0</mark>		2X=1	U2	X=0		!X=1
(bps)	UBRR	誤差(%)	UBRR	誤差(%)	UBRR	誤差(%)	UBRR	誤差(%)	UBRR	誤差(%)	UBRR	誤差(%)	UBRR	誤差(%)	UBRR	誤差(%)
1200	51	0.2	103	0.2	95	0.0	191	0.0	103	0.2	207	0.2	127	0.0	255	0.0
2400	25	0.2	51	0.2	47	0.0	95	0.0	51	0.2	103	0.2	63	0.0	127	0.0
4800	12	0.2	25	0.2	23	0.0	47	0.0	25	0.2	51	0.2	31	0.0	63	0.0
9600	6	-7.0	12	0.2	11	0.0	23	0.0	12	0.2	25	0.2	15	0.0	31	0.0
14400	3	8.5	8	-3.5	7	0.0	15	0.0	8	-3.5	16	2.1	10	-3.0	20	1.6
19200	2	8.5	6	-7.0	5	0.0	11	0.0	6	-7.0	12	0.2	7	0.0	15	0.0
28800	1	8.5	3	8.5	3	0.0	7	0.0	3	8.5	8	-3.5	4	6.7	10	-3.0
38400	1	-18.6	2	8.5	2	0.0	5	0.0	2	8.5	6	-7.0	3	0.0	7	0.0
57600	0	8.5	1	8.5	1	0.0	3	0.0	1	8.5	3	8.5	2	-11.1	4	6.7
76800	0	-18.6	1	-18.6	1	-25.0	2	0.0	1	-18.6	2	8.5	1	0.0	3	0.0
115.2k	_	_	0	8.5	0	0.0	1	0.0	0	8.5	1	8.5	0	33.3	2	-11.1
230.4k	_	_	0	-45.7	0	-50.0	0	0.0	0	-45.7	0	8.5	0	-33.3	0	33.3
250k	_	_		_	_	_	0	-7.8	_	_	0	0.0	_	_	0	22.9
最高速	62	500	12	25k	115.2k 230.4k			125k 250k			50k	153.6k 307.2k			7.2k	

ボーレート		3.2768MHz U2X=0 U2X=1				3.686	4MHz			4M	lHz		4.608MHz			
(a )	U2	X= <mark>0</mark>	U2	!X=1	U2	X= <mark>0</mark>	U2	!X=1	U2	2X= <mark>0</mark>	U2	2X=1	U2	X= <mark>0</mark>	U2	!X=1
(bps)	UBRR	誤差(%)	UBRR	誤差(%)	UBRR	誤差(%)	UBRR	誤差(%)	UBRR	誤差(%)	UBRR	誤差(%)	UBRR	誤差(%)	UBRR	誤差(%)
1200	170	-0.2	340	0.1	191	0.0	384	0.0	207	0.2	416	-0.1	239	0.0	479	0.0
2400	84	0.4	170	-0.2	95	0.0	191	0.0	103	0.2	207	0.2	119	0.0	239	0.0
4800	42	-0.8	84	0.4	47	0.0	95	0.0	51	0.2	103	0.2	59	0.0	119	0.0
9600	20	1.6	42	-0.8	23	0.0	47	0.0	25	0.2	51	0.2	29	0.0	59	0.0
14400	13	1.6	27	1.6	15	0.0	31	0.0	16	2.1	34	-0.8	19	0.0	39	0.0
19200	10	-3.0	20	1.6	11	0.0	23	0.0	12	0.2	25	0.2	14	0.0	29	0.0
28800	6	1.6	13	1.6	7	0.0	15	0.0	8	-3.5	16	2.1	9	0.0	19	0.0
38400	4	6.7	10	-3.0	5	0.0	11	0.0	6	-7.0	12	0.2	7	-6.3	14	0.0
57600	3	-11.1	6	1.6	3	0.0	7	0.0	3	8.5	8	-3.5	4	0.0	9	0.0
76800	2	-11.1	4	6.7	2	0.0	5	0.0	2	8.5	6	-7.0	3	-6.3	7	-6.3
115.2k	1	-11.1	3	-11.1	1	0.0	3	0.0	1	8.5	3	8.5	2	-16.7	4	0.0
230.4k	0	-11.1	1	-11.1	0	0.0	1	0.0	0	8.5	1	8.5	0	25.0	2	-16.7
250k	_	_	1	-18.1	0	-7.8	1	-7.8	0	0.0	1	0.0	0	15.2	1	15.2
500k	1	1	0	-18.1	_	1	0	-7.8	-	_	0	0.0	0	-42.4	0	15.2
1M	-	1	1	_	-	-	-	_	-	_	_	_		_	0	-42.4
最高速	20	4.8k	40	9.6k	23	0.4k	46	0.8k	2	50k	50	00k	28	38k	5	76k

注: 最高速はUBRR=0、誤差=0.0%です。 以降の周波数は次頁へ続く。

(<mark>訳注</mark>) 原書では頁割された表毎に表番号79.~82.となっていますが、共通性から纏めて表79.としました。 原書に対して数種の発振周波数を追加しました。

表79 (続	き). Xt	al、ボーレ	一b対U	BRRH,U	BRRL∄	<b>殳定</b> (UB	RR=UE	BRRH:UE	BRRL)							
ホーレート		4.915	2MHz			6.144	IMHz			7.372	8MHz			8M	lHz	
(bps)	U2	2X= <mark>0</mark>	U2	2X=1	U2	!X= <mark>0</mark>	U2	!X=1	U2	!X= <mark>0</mark>	U2	!X=1	U2	X= <mark>0</mark>	U2	2X=1
(phs)	UBRR	誤差(%)	UBRR	誤差(%)	UBRR	誤差(%)	UBRR	誤差(%)	UBRR	誤差(%)	UBRR	誤差(%)	UBRR	誤差(%)	UBRR	誤差(%)
1200	255	0.0	511	0.0	319	0.0	639	0.0	383	0.0	767	0.0	416	-0.1	832	0.0
2400	127	0.0	255	0.0	159	0.0	319	0.0	191	0.0	383	0.0	207	0.2	416	-0.1
4800	63	0.0	127	0.0	79	0.0	159	0.0	95	0.0	191	0.0	103	0.2	207	0.2
9600	31	0.0	63	0.0	39	0.0	79	0.0	47	0.0	95	0.0	51	0.2	103	0.2
14400	20	1.6	42	-0.8	26	-1.2	52	0.6	31	0.0	63	0.0	34	-0.8	68	0.6
19200	15	0.0	31	0.0	19	0.0	39	0.0	23	0.0	47	0.0	25	0.2	51	0.2
28800	10	-3.0	20	1.6	12	2.6	26	-1.2	15	0.0	31	0.0	16	2.1	34	-0.8
38400	7	0.0	15	0.0	9	0.0	19	0.0	11	0.0	23	0.0	12	0.2	25	0.2
57600	4	6.7	10	-3.0	6	-4.8	12	2.6	7	0.0	15	0.0	8	-3.5	16	2.1
76800	3	0.0	7	0.0	4	0.0	9	0.0	5	0.0	11	0.0	6	-7.0	12	0.2
115.2k	2	-11.1	4	6.7	2	11.1	6	-4.8	3	0.0	7	0.0	3	8.5	8	-3.5
230.4k	1	33.3	2	-11.1	1	-16.7	2	11.1	1	0.0	3	0.0	1	8.5	3	8.5
250k	1	22.9	2	-18.1	1	-23.2	2	2.4	1	-7.8	3	0.0	1	0.0	3	0.0
500k	0	-38.6	0	22.9	0	-23.2	1	-23.2	0	-7.8	1	-7.8	0	0.0	1	0.0
1M	_	-	0	-38.6	_	_	0	-23.2	_	_	0	-7.8	_	_	0	0.0
最高速	30	7.2k	61	4.4k	38	84k	70	68k	46	0.8k	92	1.6k	50	00k	]	M

ホーレート		9.216	6MHz			9.830	4MHz			101	1Hz			11.059	2MHz	
· /- \	U2	2X= <mark>0</mark>	U2	!X=1	U2	!X= <mark>0</mark>	U2	X=1	U2	2X= <mark>0</mark>	U2	!X=1	U2	2X= <mark>0</mark>	U2	!X=1
(bps)	UBRR	誤差(%)	UBRR	誤差(%)												
1200	479	0.0	959	0.0	511	0.0	1023	0.0	520	-0.0	1041	-0.0	575	0.0	1151	0.0
2400	239	0.0	479	0.0	255	0.0	511	0.0	259	0.2	520	-0.0	287	0.0	575	0.0
4800	119	0.0	239	0.0	127	0.0	255	0.0	129	0.2	259	0.2	143	0.0	287	0.0
9600	59	0.0	119	0.0	63	0.0	127	0.0	64	0.2	129	0.2	71	0.0	143	0.0
14400	39	0.0	79	0.0	42	-0.8	84	0.4	42	0.9	86	-0.2	47	0.0	95	0.0
19200	29	0.0	59	0.0	31	0.0	63	0.0	32	-1.4	64	0.2	35	0.0	71	0.0
28800	19	0.0	39	0.0	20	1.6	42	-0.8	21	-1.4	42	0.9	23	0.0	47	0.0
38400	14	0.0	29	0.0	15	0.0	31	0.0	15	1.8	32	-1.4	17	0.0	35	0.0
57600	9	0.0	19	0.0	10	-3.0	20	1.6	10	-1.4	21	-1.4	11	0.0	23	0.0
76800	7	6.7	14	0.0	7	0.0	15	0.0	7	1.8	15	1.8	8	0.0	17	0.0
115.2k	4	0.0	9	0.0	4	6.7	10	-3.0	4	8.5	10	-1.4	5	0.0	11	0.0
230.4k	2	-16.7	4	0.0	2	-11.1	4	6.7	2	-9.6	4	8.5	2	0.0	5	0.0
250k	1	15.2	4	-7.8	1	-23.2	4	-1.7	2	-16.7	4	0.0	2	-7.8	5	-7.8
500k	0	15.2	1	15.2	0	22.9	1	22.9	0	25.0	2	-16.7	0	38.2	2	-7.8
1M	0	-42.4	0	15.2	0	-38.6	0	22.9	0	-37.5	0	25.0	0	-30.9	0	38.2
最高速	5	76k	1.1	.52M	61	4.4k	1.23	288M	62	25k	1.2	25M	69	1.2k	1.3	824M

ボーレート		14.745	56MHz			161	ИHz			18.432	20MHz			201	ИHz	
/	U2	!X= <mark>0</mark>	U2	2X=1	U2	!X= <mark>0</mark>	U2	X=1	U2	!X= <mark>0</mark>	U2	!X=1	U2	!X= <mark>0</mark>	U2	!X=1
(bps)	UBRR	誤差(%)	UBRR	誤差(%)	UBRR	誤差(%)	UBRR	誤差(%)	UBRR	誤差(%)	UBRR	誤差(%)	UBRR	誤差(%)	UBRR	誤差(%)
1200	767	0.0	1533	0.0	832	0.0	1666	0.0	959	0.0	1919	0.0	1041	0.0	2082	0.0
2400	383	0.0	767	0.0	416	-0.1	832	0.0	479	0.0	959	0.0	520	0.0	1041	0.0
4800	191	0.0	383	0.0	207	0.2	416	-0.1	239	0.0	479	0.0	259	0.2	520	0.0
9600	95	0.0	191	0.0	103	0.2	207	0.2	119	0.0	239	0.0	129	0.2	259	0.2
14400	63	0.0	127	0.0	68	0.6	138	-0.1	79	0.0	159	0.0	86	-0.2	173	-0.2
19200	47	0.0	95	0.0	51	0.2	103	0.2	59	0.0	119	0.0	64	0.2	129	0.2
28800	31	0.0	63	0.0	34	-0.8	68	0.6	39	0.0	79	0.0	42	0.9	86	-0.2
38400	23	0.0	47	0.0	25	0.2	51	0.2	29	0.0	59	0.0	32	-1.4	64	0.2
57600	15	0.0	31	0.0	16	2.1	34	-0.8	19	0.0	39	0.0	21	-1.4	42	0.9
76800	11	0.0	23	0.0	12	0.2	25	0.2	14	0.0	29	0.0	15	1.7	32	-1.4
115.2k	7	0.0	15	0.0	8	-3.5	16	2.1	9	0.0	19	0.0	10	-1.4	21	-1.4
230.4k	3	0.0	7	0.0	3	8.5	8	-3.5	4	0.0	9	0.0	4	8.5	10	-1.4
250k	3	-7.8	6	5.3	3	0.0	7	0.0	4	-7.8	8	2.4	4	0.0	9	0.0
500k	1	-7.8	3	-7.8	1	0.0	3	0.0	1	15.2	4	-7.8	1	25.0	4	0.0
1M	0	-7.8	1	-7.8	0	0.0	1	0.0	0	15.2	1	15.2	0	25.0	1	25.0
最高速	92	1.6k	1.8	432M	1	M	2	2M	1.1	52M	2.3	04M	1.3	25M	2.	5M

注: 最高速はUBRR=0、誤差=0.0%です。





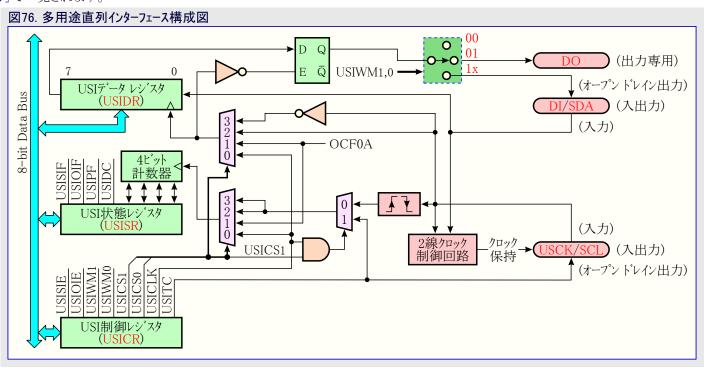
# 多用途直列インターフェース (USI: Universal Serial Interface)

多用途直列インターフェース(USI)は直列通信で必要とされる基本的なハードウェア資源(機能)を提供します。最低限の制御ソフトウェアとの組み合わせで、USIはソフトウェアのみを元とした方法よりも少ないコード量と重要な高い転送レートを許します。プロセッサ負荷を最小とするために割り込みが含まれます。USIの主な特徴は次のとおりです。

- 2線同期データ転送(主装置または従装置)
- 3線同期データ転送(主装置または従装置)
- データ受信割り込み
- アイドル動作からの起動復帰
- 2線動作でのパワーダウン動作を含む全休止形態からの起動復帰
- 割り込み能力を持つ2線動作開始条件検出器

#### 概要

USIの簡単化した構成図は**図76**.で示されます。実際のI/Oピンの配置は2頁の「ピン配置」を参照してください。I/OビットとI/Oピンを含みCPUがアクセス可能なI/Oレジスタは赤文字(訳注:原文は太字)で示されます。デバイス仕様のI/Oレジスタとビット位置は127頁の「USI用レジスタ」で一覧されます。



8ビットの移動レジスタ(USIDR)はデータ バス経由で直接的にアクセス可能で、入って来るデータと出て行くデータを含みます。このレジスタは緩衝部を持たないので、データが失われないことを保証するためにデータは可能な限り早く読まなければなりません。最上位ビットは動作種別設定に依存して2つの出力ピンの1つに接続されます。トランスペアレント(透過型)ラッチは直列(移動)レジスタ出力と出力ピン間に挿入され、データの変更をデータ入力採取と逆のクロック端に遅らせます。直列入力は(種別)設定に拘らず、常にデータ入力(DI)ピンから採取されます。

4ビット計数器はデータ バス経由で読み書き両方ができ、溢れ割り込みを生成できます。この計数器と直列(移動)レジスタは同じクロック元によって同時にクロック駆動されます。これは受信または送信したビット数を数え、転送完了時に割り込みの生成を計数器に許します。外部クロック元が選ばれると、この計数器が両クロック端で計数することに注意してください。この場合、計数器はビット数ではなく、端(エッジ)数を数えます。クロックは3つの異なるクロック元(USCKピン、タイマ/カウンタ0の比較一致、ソフトウェア)から選べます。

2線クロック制御部(回路)は2線バスで開始条件が検出される時に割り込みを発生できます。開始条件検出後、または計数器溢れ後、ク ロック ピンをLowに保持することによって待ち状態を生成することもできます。

### 機能説明

#### 3線動作

USIの3線動作は直列周辺インターフェース(SPI)の動作種別0と1に準拠していますが、従装置選択(SS)ピン機能を持ちません。けれども、この特性(機能)は必要ならばソフトウェアで実現できます。この動作で使われるピン名はDI,DO,USCKです。

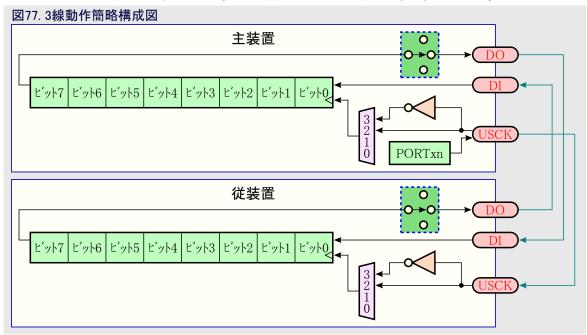
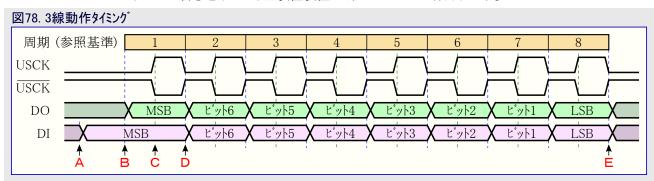


図77.は3線動作での2つのUSI部(一方は主装置、他方は従装置)動作を示します。このような方法で連結された2つの移動レジスタは8 USCKクロック後、各々のレジスタのデータが交換されます。この同じクロック(USCK)がUSIの4ビット計数器も増加(+1)します。従って計数器溢れ割り込み要求フラグ(USIOIF)は転送が完了される時を決めるのに使用えます。このクロックはPORTレジスタ経由USCKピン1/0交互切り替え、またはUSICRのUSITCビットへの1書き込みによる主装置装置ソフトウェアによって生成されます。



3線動作タイミングは図78.で示されます。図の最上部はUSCK周期参照基準です。これらの各周期に対して1ビットがUSI移動レジスタ (USIDR)に移動されます。USCKタイミングは両方の外部クロック動作について示されます。外部クロック動作0(USICS0=0)でのDIは上昇端で採取され、DOは下降端で変更(USIDRが1つ移動)されます。外部クロック動作1(USICS0=1)は外部クロック動作0に対して逆端を使用、換言すると下降端でデータを採取、上昇端で出力を変更します。USIクロック動作種別はSPIデータ動作種別0と1に対応します。このタイミング図(図78.)を参照すると、バス転送は次の手順を含みます。

- 1. 主装置と従装置はデータ出力を設定し、使用規約に依存して出力駆動部を許可します(AとB)。この出力は直列データレジスタへの送信されるべきデータ書き込みによって設定します。出力の許可はポート方向レジスタの対応するビット設定によって行います。A点とB点はどんな特別な順番もありませんが、両方共にデータが採取されるC点よりも最低1/2 USCK周期前でなければならないことに注意してください。これはデータ設定の必要条件を満足させるのを保証するために行わなければなりません。4ビット計数器は0にリセットします。
- 2. 主装置はUSCK線をソフトウェアで2度切り替えることによってクロック パルスを生成します(CとD)。主装置と従装置のデータ入力(DI)ピンの ビット値は最初の端(エッジ)(C)でUSIによって採取され、データ出力は逆端(D)で変更されます。4ビット計数器は両端で計数します。
- 3. レジスタ(バイ)転送完了のために手順2.が8回繰り返されます。
- 4.8クロック パルス(換言すると16クロック端)後、計数器が溢れて転送完了を示します。転送したバイト データは新規転送が開始され得るのに先立って直ぐに処置されなければなりません。溢れ割り込みはプロセッサがアイドル動作に設定されているなら、プロセッサを起動復帰します。使用規約によって、従装置は直ぐに出力をHi-Z設定にできます。





### SPI主装置操作例

次のコートはSPI主装置としてのUSI部使用法を実際に示します。

アセンブリ言語	ラプログラムイ	例	
SPIM:	STS LDI STS LDI	USIDR, R16 R16, (1< <usioif) USISR, R16 R16, (1&lt;<usiwmo) (1<<usiclk)="" (1<<usics1)="" (1<<usitc)<="" th=""  =""><th>;送信データを設定 ;USIOIFビットのみ1値を取得 ;フラグ解除/計数器初期化 ;3線動作クロック生成値を取得</th></usiwmo)></usioif) 	;送信データを設定 ;USIOIFビットのみ1値を取得 ;フラグ解除/計数器初期化 ;3線動作クロック生成値を取得
;			
SPIM_LP:	STS LDS SBRS RJMP	USICR, R16 R16, USISR R16, USIOIF SPIM_LP	;USCKクロック端発生 ;USI状態フラク゚を取得 ;計数器溢れでスキップ ;計数器溢れまで継続
,	LDS RET	R16, USIDR	;受信データを取得 ;呼び出し元へ復帰

このコートは9命令(+RET)のみ使用の容量最適化です。このコート例はDOとUSCKピンがDDREレジスタで出力として許可されると仮定します。この関数が呼び出される前にR16レジスタへ格納した値は従装置に転送され、転送が完了された時に従装置から受信したデータがR16レジスタに格納されて戻ります。

2行目と3行目の命令は計数器溢れ割り込み要求フラグ(USIOIF)を解除(0)し、USI 4ビット計数器値を解除(=0)します。4行目と6行目の命令は3線動作、上昇端移動レジスタ クロック、USITCストロープ計数、USCK出力交互切り替えを設定します。この繰り返しは16回繰り返されます。

次のコート は最高速( $f_{USCK}=f_{CK}/4$ )でのSPI主装置としてのUSI部使用法を実際に示します。

アセンブリ言語	吾フ <sup>°</sup> ロク゛ラム・	例	
SPIM_F:	STS	USIDR, R16	;送信データを設定
	LDI	R16, (1< <usiwmo) (0<<usicso)="" (1<<usitc)<="" td=""  =""><td>;3線動作初期値を取得</td></usiwmo)>	;3線動作初期値を取得
	LDI	R17, (1< <usiwmo) (0<<usicso)="" (1<<usiclk)<="" (1<<usitc)="" td=""  =""><td>;3線動作クロック生成値を取得</td></usiwmo)>	;3線動作クロック生成値を取得
;			
	STS	USICR, R16	;MSB転送
	STS	USICR, R17	;
	STS	USICR, R16	;ビット6転送
	STS	USICR, R17	; ;
	STS	USICR, R16	;ビット5転送
	STS	USICR, R17	;
	STS	USICR, R16	;ビット4転送
	STS	USICR, R17	;
	STS	USICR, R16	;ビット3転送
	STS	USICR, R17	; • • • • • • • • • • • • • • • • • • •
	STS	USICR, R16	;ビット2転送
	STS STS	USICR, R17	, ・レ、、1 1声: ご光
	STS	USICR, R16	;ビット1転送
	STS	USICR, R17 USICR, R16	, ;LSB転送
	STS	USICR, R17	,LSD#公区 ·
	313	USION, NI I	,
,	LDS	R16, USIDR	;受信データを取得
	RET	K10, 001DK	;呼び出し元へ復帰
	IVL I		, 一, 0 四0元 一夜/市

#### SPI従装置操作例

次のコートはSPI従装置としてのUSI部使用法を実際に示します。

アセンブリ言語	吾プログラムイ	例	
init:	LDI	R16, (1< <usiwmo) (1<<usics1)<="" th=""  =""><th>;3線動作,外部クロック値を取得</th></usiwmo)>	;3線動作,外部クロック値を取得
	STS	USICR, R16	;3線動作,外部クロック設定
SPIS:	STS	USIDR, R16	;送信データを設定
	LDI	R16, (1< <usioif)< td=""><td>;USIOIFビットのみ1値を取得</td></usioif)<>	;USIOIFビットのみ1値を取得
	STS	USISR, R16	;フラグ解除/計数器初期化
SPIS_LP:	LDS	R16, USISR	;USI状態フラグを取得
	SBRS	R16, USIOIF	;計数器溢れでスキップ
	RJMP	SPIS_LP	;計数器溢れまで継続
,	LDS RET	R16, USIDR	;受信データを取得 ;呼び出し元へ復帰

このコート、は9命令(+RET)のみ使用の容量最適化です。このコート、例はDDREレシ、スタでDOピンが出力、USCKピンが入力として設定されると仮定します。この関数が呼び出される前にR16レシ、スタへ格納した値は主装置に転送され、転送が完了された時に主装置から受信したデータがR16レシ、スタに格納されて戻ります。

最初の2命令は初期化用だけで、一度だけ実行されるのを必要とすることに注意してください。これらの命令は3線動作と上昇端移動 レジスタ クロックを設定します。この繰り返しはUSI計数器溢れフラグが設定(1)されるまで繰り返されます。

#### 2線動作

USIの2線動作はI<sup>2</sup>C(TWI)バス規約に準拠していますが、出力のスリューレート制限と入力雑音濾波器がありません。この動作で使われるピン名はSCLとSDAです。

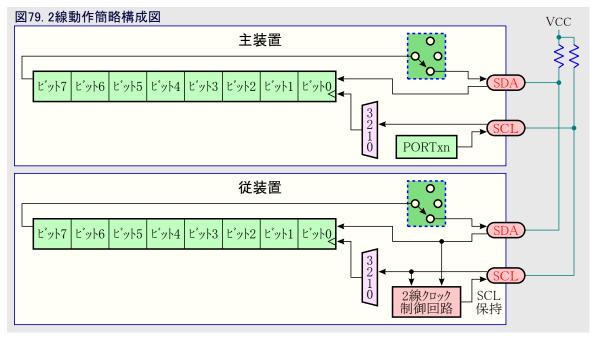


図79.は2線動作での2つのUSI部(一方は主装置、他方は従装置)動作を示します。システム動作が使用通信構造に大きく依存するために示されるのは物理層のみです。この段階での主装置動作と従装置動作間の主な違いは常に主装置によって行われる直列クロック生成と、従装置だけがクロック制御部(回路)を使うことです。クロック生成はソフトウェアで実行されなければなりませんが、移動動作は両装置で自動的に行われます。データの移動に関してこの動作種別で下降端でのクロックだけを実際に使うことに注意してください。従装置はSCLクロックをLowに強制することにより、転送の開始と終了で待ち状態(ウェイトステート)を挿入できます。これは(主装置が)上昇端を生成してしまった後に主装置はSCL線が実際に開放されたかを常に検査しなければならないことを意味します。

このクロックが(4ビット)計数器も増加(+1)するため、計数器溢れ(USIOIF)は転送が完了されたことを示すのに使えます。このクロックは主装置によるPORTEレジスタ経由のSCLピン交互切り替えによって生成されます。

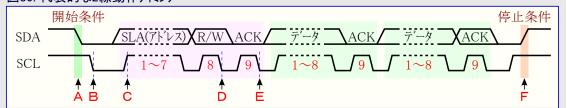
データ方向は物理層によって与えられません。データの流れを制御するためにTWIバスで使ったような或る種の規約が実装されなければなりません。





このタイミンケ 図(図80.)を参照すると、ハス転送は次の手順を含みます。

### 図80. 代表的な2線動作タイミング

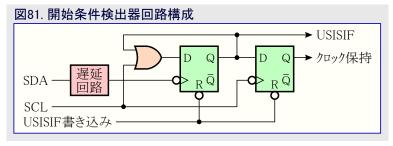


- 1. 開始条件は主装置によるSCL線High中のSDA線Low強制によって生成されます(A)。SDAは移動レジスタのビット7への0書き込みか、またはPORTレジスタで対応ビットを0に設定のどちらかによって強制実行できます。出力が許可されるにはデータ方向(DDR)レジスタのビットが1に設定されなければならないことに注意してください。従装置の開始条件検出器論理回路(図81.参照)が開始条件を検出してUSISIFフラグを設定(1)します。このフラグは必要ならば割り込みを発生できます。
- 2. 加えて開始条件検出器は主装置がSCL線で上昇端を強制してしまった後にSCL線をLowに保持します(B)。これはアドレスを受信するために移動レジスタを設定する前に休止形態から起動復帰または他の処理を完了することを従装置に許します。これ(保持解除)は(4ビット)計数器をリセット(=0)して開始条件検出フラグ(USISIF)を解除(0)することによって行われます。
- 3. 主装置は転送されるべき最初のビットを設定してSCL線を開放します(C)。従装置はSCLクロックの上昇端でデータを採取して移動レジスタへ移動します。
- 4. 従装置アドレスとデータ方向(R/W)を含む8ビットが転送された後、従装置の(4ビット)計数器が溢れてSCL線がLowを強制されます(D)。 従装置は主装置の1つがアドレス指定してしまわない場合、SCL線を開放して新規開始条件を待ちます。
- 5. 従装置がアトンス指定されると、再びSCL線をLowに保持する前の応答(ACK)周期の間中、SDA線をLowに保持します(換言すると D点でSCLを開放する前に(4ビット)計数器レジスタは14に設定されなければなりません)。R/Wビットに依存して主装置か従装置は出力を許可します。このビットが設定(1)の場合、主装置読み込み動作が進行中です(換言すると従装置がSDA線を駆動します)。従装置は応答後(E)にSCL線をLowに保持することができます。
- 6. 主装置によって<mark>停止条件(F)</mark>か新規開始条件(再送開始条件)が与えられるまで全て同方向で複数バイが直ぐに転送できます。 従装置はより多くのデータを受信できない場合、最後に受信したバイトデータに応答しません。主装置が読み込み動作を行うとき、最後のバイルが転送された後の応答ビットをLowに強制することよってこの動作を終了しなければなりません。

#### 開始条件検出器

開始条件検出器は図81.で示されます。SDA線はSCL線の有効な採取を保証するために(50~300nsの範囲で)遅延されます。開始条件検出器は2線動作でだけ許可されます。

開始条件検出器は非同期に動作し、従ってパワーダウン休止動作からプロセッサを起動できます。けれども使う規約がSCL保持時間で制限を持つかもしれません。従ってこの場合にこの特徴(機能)を使う時にCKSELヒュースで設定する発振器起動時間(16頁の「クロック系統とその配給」参照)も考慮内に取り入れられなければなりません。より多くの詳細については127頁のUSISIFフラグ記述を参照してください。



#### クロック速度の考察

SCLとUSCKに対する最大周波数はfcK/4です。これは2線と3線の両動作での最大データ転送と受信速度でもあります。2線従装置動作では従装置が更にデータを受信する準備ができるまで、2線クロック制御部がSCLを保持します。これは2線動作で実データ速度を減らすかもしれません。

USIでの代替使用

USI部を直列通信に使わないなら、柔軟な設計のため、代替処理を行うように設定できます。

半二重非同期データ転送

3線動作で移動レジスタを使うことによってソフトウェアだけよりも簡潔で高性能のUART実装が可能です。

4ビット計数器

4ビット計数器は溢れ割り込みとの自立型計数器として使えます。この計数器が外部的にクロック駆動される場合、両クロック端が増加(+1)を発生することに注意してください。

12ビット計数器

USIの4ビット計数器とタイマ/カウンタ0の組み合わせは12ビット計数器としての使用を許します。

端(エッジ)起動外部割り込み

4ビット計数器を最大値(\$F)に設定することによって追加外部割り込みとして機能できます。溢れフラグと割り込み許可ビットはこの外部割り込みに対して使われます。この特徴(機能)はUSICS1 ビットによって選択(=1)されます。

ソフトウェア割り込み

4ビット計数器の溢れ割り込みはクロック ストローブによって起動するソフトウェア割り込みとして使えます。

### USI用 レジスタ

#### USIデータレシ スタ (USI Data Register) USIDR

ピット	7	6	5	4	3	2	1	0	_
(\$BA)	(MSB)							(LSB)	USIDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

USIは直列(移動)レジスタの緩衝を使いません、換言するとデータレジスタ(USIDR)をアクセスする時に直列レジスタは直接的にアクセスされます。レジスタが書かれるのと同じ周期で直列クロックが起こると、レジスタは書いた値を含み、移動は実行されません。(左)移動動作はUSIC S1,0ビット設定に従って実行されます。この移動動作は外部クロック端、タイマ/カウンタの比較一致により、またはUSICLKストローブ・ビットを使うソフトウェアによって直接的に制御できます。2または3線動作種別が選ばれない(USIWM1,0=00)時でも、外部データ入力(DI/SDA)と外部クロック入力(USCK/SCL)の両方が未だ移動レジスタによって使われ得ることに注意してください。

動作種別に従って使う出力ピン(DOまたはSDA)は出力ラッチを経由してデータレジスタの最上位ビット(ビット7)に接続されます。この出力ラッチは外部クロック元が選ばれる(USICS1=1)時に直列クロック周期の前半中が透過(トランスペプレント)で、内部クロック元が使われる(USICS1=0)時に定常的な透過です。ラッチが透過である限り、新規MSBが書かれると、出力は直ちに変更されます。このラッチは違う(逆の)クロック端でデータ入力が採取され、データ出力が変更されるのを保証します。

移動レジスタからの出力データを許可するために、ピンに対応するデータ方向レジスタ(DDxn)が1に設定されなければならないことに注意してください。

#### USI状態レジスタ (USI Status Register) USISR

ピット	7	6	5	4	3	2	1	0	
(\$B9)	USISIF	USIOIF	USIPF	USIDC	USICNT3	USICNT2	USICNT1	USICNT0	USISR
Read/Write	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

この状態レジスタは割り込み要求フラグ、状態フラグ、計数器値を含みます。

#### • ビット7 - USISIF: 開始条件検出割り込み要求フラグ(Start Condition Interrupt Flag)

2線動作が選ばれるとき、開始条件が検出されると、USISIFフラグが設定(1)されます。出力禁止動作または3線動作が選ばれ、(USICS 1,0=11且つUSICLK=0)または(USICS1,0=10且つUSICLK=0)の時にUSCKピンのどの端(エッジ)もがこのフラグを設定(1)します。

ステータスレジスタ(SREG)の全割り込み許可(I)ビットとUSI制御レジスタ(USICR)で開始条件検出許可(USISIE)ビットが設定(1)されている間中にこのフラグが設定(1)されると、割り込みが生成されます。このフラグはUSISIFビットへの論理1書き込みによってのみ解除(0)されます。このビットの解除(0)は2線動作でのSCLの開始条件保持を解除(開放)します。

開始条件検出割り込みは全ての休止形態からプロセッサを起動します。

### • ビット6 - USIOIF: 計数器溢れ割り込み要求フラグ (Counter Overflow Interrupt Flag)

このフラグは4ビット計数器が溢れると(換言すると15から0への遷移で)設定(1)されます。SREGの全割り込み許可(I)ビットとUSICRで計数器溢れ割り込み許可(USIOIE)ビットが設定(1)されている間に本フラグが設定(1)されると、割り込みが生成されます。本フラグはUSIOIF ビットへの論理1書き込みによってのみ解除(0)されます。本ビットの解除(0)は2線動作でのSCLの計数器溢れ保持を解除(開放)します。計数器溢れ割り込みはアイデル休止動作からプロセッサを起動します。

### • ビット5 - USIPF: 停止条件検出フラグ(Stop Condition Flag)

2線動作が選ばれた時に停止条件が検出されると、USIPFフラケが設定(1)されます。このフラケはこのビットへの1書き込みによって解除 (0)されます。これが割り込み要求フラケでないことに注意してください。この合図は2線バス主装置調停の実装時に有用です。

#### • ビット4 - USIDC: 出力データ衝突フラグ (Data Output Collision Flag)

移動レシ、スタのビット7(MSB)が物理ピン値と異なる時にこのビットは論理1です。このフラグは2線動作が使われる時にだけ有効です。この合図は2線バス主装置調停の実装時に有用です。

#### • ビット3~0 - USICNT3~0: 計数器値 (Counter Value)

これらのビットは現在の4ビット計数器値を反映します。この4ビット計数器値はCPUによって直接的に読み書きできます。

この4ビット計数器は外部クロック端検出器、タイマ/カウンタ0比較一致、USICLKまたはUSITCストローブビットを使うソフトウェアのどれかによって生成した各クロックに対して1つ増加します。このクロック元はUSICS1,0ビット設定に依存します。外部クロック動作に対してUSITCストローブビットへの書き込みによって生成されることをクロックに許す、特別な特徴が付加されます。この特徴は外部クロック元(USICS1=1)設定中のUSICLKへの1書き込みによって許可されます。

出力禁止動作(USIWM1,0=00)が選ばれている時でも、外部クロック入力(USCK/SCL)が未だ計数器によって使われることに注意してください。





### USI制御レジスタ (USI Control Register) USICR

ピット	7	6	5	4	3	2	1	0	
(\$B8)	USISIE	USIOIE	USIWM1	USIWM0	USICS1	USICS0	USICLK	USITC	USICR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	W	W	
初期値	0	0	0	0	0	0	0	0	

この制御レジスタは割ウ込み許可制御、動作種別設定、クロック選択設定、クロック ストローブを含みます。

### • ビット7 - USISIE: 開始条件検出割り込み許可 (Start Condition Interrupt Enable)

このビットを1に設定することが開始条件検出割り込みを許可します。ステータス レジスタ(SREG)の全割り込み許可(I)ビットとUSISIEビットが設定(1)される時に保留割り込みなら、これは直ちに実行されます。より多くの詳細については127頁の開始条件検出割り込み要求フラグ(USISIF)記述を参照してください。

#### ● ビット6 - USIOIE: 計数器溢れ割り込み許可 (Counter Overflow Interrupt Enable)

このビットを1に設定することが計数器溢れ割り込みを許可します。ステータスレジ、スタ(SREG)の全割り込み許可(I)ビットとUSIOIEビットが設定(1)される時に保留割り込みなら、これは直ちに実行されます。より多くの詳細については127頁の計数器溢れ割り込み要求フラケー(USIOIF)記述を参照してください。

#### • ビット5,4 - USIWM1,0: 動作種別選択 (Wire Mode)

これらのビットは使われるべき動作種別(2または3線動作)を設定します。基本的に出力の機能だけがこれらのビットによって影響を及ぼされます。データとクロック入力は選んだ動作種別によって影響を及ぼされず、常に同じ機能を持ちます。従って出力が禁止される時でも移動レジスタと計数器は外部的にクロック駆動でき、入力データが採取されます。USI操作とUSIWM1,0間の関連は表83.で要約されます。

# 表83. USI操作とUSIWM1,0の関係

USIWM1	USIWM0	説明
0	0	出力、クロック保持、開始条件検出器は禁止。ポート ピンは標準として動作。
0	1	3線動作。DO,DI,USCKピンを使用。 データ出力(DO)ピンはこの動作でPORTレジスタの対応ビットを無視します。けれども対応するDDRビットは未だデータ方向を制御します。ポート ピンが入力として設定(DDxn=0)されると、そのピンのプルアップはPORTビットによって制御されます。 データ入力(DI)と直列クロック(USCK)ピンは標準ポート操作に影響を及ぼしません。主装置として動作する時のクロック パルスはデータ方向が出力に設定されている(DDxn=1)間のPORTレジスタの交互切り替えによってソフトウェアで生成されます。USICRのUSITCビットがこの目的に使えます。
1	0	2線動作。SDA(DI)とSCL(USCK)ピンを使用。(注1) 直列データ(SDA)と直列クロック(SCL)ピンは双方向でオープントレイン出力駆動を使います。この出力駆動部は DDRレジスタでSDAとSCLに対応するビットの設定(=1)によって許可されます。 SDAピンに対して出力駆動部が許可されると、出力駆動部は移動レジスタの出力またはPORTレジスタの対応 ピットが0ならばSDA線をLowに強制します。さもなければ、SDA線は駆動されません(換言すると開放されま す)。SCLピン出力駆動部が許可されると、SCL線はPORTレジスタの対応ビットが0なら、または開始条件検出 器によってLowを強制されます。さもなければSCL線は駆動されません。 SCL線は出力が許可され、開始条件検出器が開始条件を検出すると、Lowに保持されます。開始条件検 出フラグ(USISIF)の解除(0)がその線を開放します。SDAとSCLピン入力はこの動作の許可によって影響を及 ぼされません。SDAとSCLピンのプルアップは2線動作で禁止されます。
1	1	2線動作。SDAとSCLピンを使用。 SCL線は計数器溢れが起こる時にもLowへ保持され、計数器溢れフラグ(USIOIF)が解除(0)されるまでLow に保持されることを除いて、上で記述される2線動作と同じ動作です。

注1: DIとUSCKピンは動作種別間での混乱を避けるため、各々直列データ(SDA)と直列クロック(SCL)に改称されます。

#### • ビット3.2 - USICS1.0: クロック選択 (Clock Source Select)

これらのビットは移動レジスタと計数器に対するクロック元を設定します。外部クロック元(USCK/SCL)を使う時にデータ出力ラッチはデータ入力 (DI/SDA)の採取の逆端で出力が変更されるのを保証します。タイマ/カウンタの比較一致またはソフトウェア ストローブが選ばれると、出力ラッチは透過で、従って出力は直ちに変更されます。USICS1,0ビットの解除(=00)がソフトウェア ストローブ選択を許可します。この選択を使う時のUSICLKビットへの1書き込みは移動レジスタと計数器の両方をクロック駆動します。外部クロック元(USICS1=1)に対してUSICLKビットはもはやストローブとして使われませんが、外部クロックとUSITCストロープビットによるソフトウェア クロック間を選びます。

表84.は移動レジスタと4ビット計数器に対して使われるクロック元とUSICS1,0、USICLK設定間の関連を示します。

<b>耒</b> Ω/I	LISTOST	1DISIDIAN	K設定との関係
4X 04.	USIUST.	บกบอเบเ	ハラマル とりりまじか

USICS1	USICS0	USICLK	移動レジスタ クロック元	4ビット計数器クロック元
	0	0	クロックなし	クロックなし
0	U	1	ソフトウェア クロック ストローフ゛(USICLK)	ソフトウェア クロック ストローフ゛(USICLK)
	1	X	タイマ/カウンタ0 比較一致	タイマ/カウンタ0 比較一致
	0	0	外部クロック上昇端	外部クロック両端
1	1	0	外部クロック下降端	タト司3クロツク 呵 s端 
1	0	1	外部クロック上昇端	ソフトウェア クロック ストローフ゛(USITC)
	1	1	外部クロック下降端	// //±/ // // // // (USIIC)

#### • ビット1 - USICLK : クロック ストローブ (Clock Strobe)

ソフトウェア ストローフ、選択が選ばれるように行うことによってUSICS1,0ビットがゼロ(=00)に設定されるなら、このビット位置への1書き込みは移動レジスタを1段移動して計数器を1つ増加します。このクロック ストローフ・が実行される時、直ちに、換言すると同じ命令周期内で出力が替わります。移動レジスタ内に移動した値は直前の命令周期で採取されます。このビットは0として読みます。

外部クロック元が選ばれる(USICS1=1)と、USICLK機能はクロックストローフ・からクロック選択レジ、スタに変更されます。この場合のUSICLKビットの設定(1)は4ビット計数器に対するクロック元としてのUSITCストローフ・を選びます(表84.参照)。

#### • ビット0 - USITC: クロック値切り替え (Toggle Clock Port Pin)

このビット位置に1を書くことはUSCK/SCL値を0から1、または1から0のどちらかへ切り替えます。この切り替えはデータ方向レジスタでの設定と無関係ですが、PORT値がピンで見られるべきなら、DDE4が出力として設定(=1)されなければなりません。この特徴は主装置を実現する時の容易なクロック生成を許します。このビットは0として読みます。

外部クロック元が選ばれ(USICS1=1)、USICLKビットが1に設定されると、USITCストローブビットへの書き込みは直接的に4ビット計数器をクロック駆動します。これは主装置として動作する時に転送が行われる時の早い検出を許します。





# アナログ比較器

アナログ比較器は非反転入力AINOピンと反転入力AIN1ピンの入力値を比較します。非反転AINOピンの電圧が反転AIN1ピンの電圧よりも高い時にACSRのアナログ比較器出力(ACO)ビットが設定(1)されます。この比較器出力はタイマ/カウンタ1の捕獲機能を起動するように設定できます。加えて、この比較器はアナログ比較器専用の独立した割り込みを起動できます。使用者は比較器出力の上昇端、下降端、またはその両方で割り込み起動を選べます。この比較器とその周辺論理回路の構成図は図82.で示されます。

ADC入力(A/D多重器出力)の使用を可能とするには、23頁の「電力削減レジスタ(PRR)」のPRADCビットが論理0を書かれることで(電力削減を)禁止されなければなりません。

#### 図82. アナログ比較器部構成図 内部 VCC 基準電圧 **ACIE** Q **ACBG ←** ACD (AINO) 比較器 割り込み 割り込み 選択 (AIN1) ➤ ACI ACME · ACIS1 ACIS0 ACIC → タイマ/カウンタ1 ADEN 捕獲起動 ACO A/D 多重器出力 選択へ

注: A/D多重器出力については131頁の表86.をご覧ください。 アナログ比較器ピン配置については2頁の「ピン配置」と46頁の表39.を 参照してください。

#### アナログ比較器 制御/状態レジスタ (Analog Comparator Control and Status Register) ACSR

–			0 1			O	,		
ピット	7	6	5	4	3	2	1	0	
\$30 (\$50)	ACD	ACBG	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	ACSR
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
初期値	0	0	不定	0	0	0	0	0	

• ビット7 - ACD: アナログ比較器禁止 (Analog Comparator Disable)

このビットが論理1を書かれると、アナログ比較器への電力がOFFにされます。このビットはアナログ比較器をOFFにするために何時でも設定(1)できます。これは活動動作やアイドル動作で電力消費を削減します。ACDビットを変更する時にACSRのアナログ比較器割り込み許可(ACIE)ピットを解除(0)することによってアナログ比較器割り込みが禁止されなければなりません。さもなければ、このビットが変更される時に割り込みが起こり得ます。

• ビット6 - ACBG: 基準電圧選択 (Analog Comparator Bandgap Select)

このビットが設定(1)されると、内部基準電圧(公称1.1V)がアナログ比較器への非反転入力に置き換わります。本ビットが解除(0)されると、AINOがアナログ比較器の非反転入力に印加されます。内部基準電圧がアナログ比較器への入力として使われるとき、電圧の安定に一定時間を必要とします。安定(待機を)しない場合、不正な比較になるかもしれません。27頁の「内部基準電圧」をご覧ください。

- ビット5 ACO: アナログ比較器出力 (Analog Comparator Output)
- アナログ比較器の出力は同期化され、その後に直接ACOへ接続されます。この同期化は1~2クロック周期の遅延をもたらします。
- ビット4 ACI: アナログ比較器割り込み要求フラグ(Analog Comparator Interrupt Flag)

このビットは比較器出力での出来事がACSRのアナログ比較器割り込み条件(ACIS1,0)ビットによって定義した割り込み方法で起動する時に設定(1)されます。ACSRのアナログ比較器割り込み許可(ACIE)ビットが設定(1)され、ステータス レジスタ(SREG)の全割り込み許可(I)ビットが設定(1)されていると、アナログ比較器割り込みルーチンが実行されます。対応する割り込み処理ベクタを実行すると、ACIはハードウェアによって解除(0)されます。代わりにこのフラグへ論理1を書くことによってもACIは解除(0)されます。

• ビット3 - ACIE: アナログ比較器割り込み許可 (Analog Comparator Interrupt Enable)

ACIEビットが論理1を書かれ、ステータス レジスタ(SREG)の全割り込み許可(I)ビットが設定(1)されていると、アナログ比較器割り込みが活性(有効)にされます。論理0を書かれると、この割り込みは禁止されます。

• ビット2 - ACIC: アナログ比較器捕獲起動許可 (Analog Comparator Input Capture Enable)

論理1を書かれると、このビットはアナログ比較器によって起動されるタイマ/カウンタ1の捕獲機能を許可します。この場合、比較器出力は比較器にタイマ/カウンタ1捕獲割り込みの雑音消去機能と端(エッジ)選択機能を利用させる捕獲入力前置論理回路へ直接的に接続されます。論理0を書かれると、アナログ比較器と捕獲機能間の接続は存在しません。比較器がタイマ/カウンタ1捕獲割り込みを起動するには、タイマ/カウンタ1割り込み許可レジスタ(TIMSK1)の捕獲割り込み許可(ICIE1)ビットが設定(1)されなければなりません。

• ビット1,0 - ACIS1,0: アナログ比較器割り込み条件 (Analog Comparator Interrupt Mode Select)

これらのビットは比較器のどの事象がアナログ比較器割り込みを起動するのかを決めます。各種設定は表85.で示されます。

ACIS1,ACISOビットを変更する時にACSRのアナログ比較器割り込み許可(ACIE)ビットを解除(0)することによってアナログ比較器割り込みが禁止されなければなりません。さもなければ、これらのビットが変更される時に割り込みが起き得ます。

表85. アナログ比較器割り込み条件選択					
ACIS1	ACIS0	割り込み発生条件			
0	0	比較器出力の変移(トグル)			
0	1	(予約)			
1	0	比較器出力の下降端			
1	1	比較器出力の上昇端			

### A/D変換制御/状態レジスタB (ADC Control and Status Register B) ADCSRB

ピット	7	6	5	4	3	2	1	0	
(\$7B)	_	ACME	-	-	-	ADTS2	ADTS1	ADTS0	ADCSRB
Read/Write	R	R/W	R	R	R	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

### • ビット6 - ACME: アナログ比較器多重器許可 (Analog Comparator Multiplexer Enable)

このビットが論理1を書かれ、A/D変換部がOFF(ADCSRAのADENビットが0)にされると、A/D変換の多重器がアナログ比較器への反転入力を選びます。このビットが論理0を書かれると、AIN1がアナログ比較器の反転入力に印加されます。このビットの詳細な記述については次の「アナログ比較器入力選択」をご覧ください。

### アナログ比較器入力選択

アナログ比較器への反転入力を取り替えるのにADC7~0のどれかを選ぶことができます。A/D変換の多重器がこの入力選択に使われ、従ってこの機能を利用するためにA/D変換部がOFF(動作禁止)にされなければなりません。ADCSRBのアナログ比較器多重器許可(ACME)ビットが設定(1)され、A/D変換部がOFF(ADCSRAのADENビットが0)にされていれば、表86.で示されるようにADMUXのチャネル選択(MUX2~0)ビットがアナログ比較器への反転入力を取り替えるための入力ピンを選びます。ACMEが解除(0)、またはADENが設定(1)されると、AIN1がアナログ比較器への反転入力に印加されます。

### 表86. アナログ比較器反転入力選択

ACME	ADEN	MUX2~0	アナログ比較器反転入力	備考
0	X	ххх	AIN1	
	1	ххх	AINI	
		0 0 0	ADC0	
		0 0 1	ADC1	
		0 1 0	ADC2	
1	0	0 1 1	ADC3	
	U	1 0 0	ADC4	
		1 0 1	ADC5	
		1 1 0	ADC6	
		111	ADC7	

#### デジタル入力禁止レジスタ1 (Digital Input Disable Register 1) DIDR1

ピット	7	6	5	4	3	2	1	0	
(\$7F)	_	-	-	-	-	-	AIN1D	AIN0D	DIDR1
Read/Write	R	R	R	R	R	R	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

#### • ビット1,0 - AIN1D,AIN0D: AIN1,AIN0 デジタル入力禁止 (AIN1,AIN0 Digital Input Disable)

このビットが論理!を書かれると、AIN1/0ピンのデジタル入力緩衝部が禁止されます。このビットが設定(1)されると、対応するポート入力レジスタのビット(PINx)は常に0として読みます。AIN1/0ピンにアナログ信号が印加され、そのピンからのデジタル入力が必要とされない時にデジタル入力緩衝部での消費電力を削減するため、このビットは論理!を書かれるべきです。





# A/D変換器

A/D変換部の主な機能を次に示します。

- 10ビット分解能
- 積分非直線性誤差0.5 LSB
- 絶対精度±2 LSB
- 変換時間13~260µs (50kHz~1MHz変換クロック)
- 最大分解能で15kSPS(採取/s)まで (200kHz変換クロック)
- 8チャネルのシングル エント、入力多重器内蔵
- 7チャネルの差動入力チャネル(訳注:共通性から追加)
- A/D変換結果読み出しに対する任意の左揃え
- 0~VCC A/D変換入力電圧範囲
- 選択可能な1.1V A/D変換基準電圧
- 連続と単独の変換動作
- ・割り込み元の自動起動によるA/D変換開始
- A/D変換完了割り込み
- 休止形態雑音低減機能

ATmega169は10ビット逐次比較A/D変換器が特徴です。このA/D変換器はポートFのピンから構成された8つのシングル エンド電圧入力を許す8チャネル アナログ多重器に接続されます。このシングル エンド電圧入力は0V(GND)が基準です。

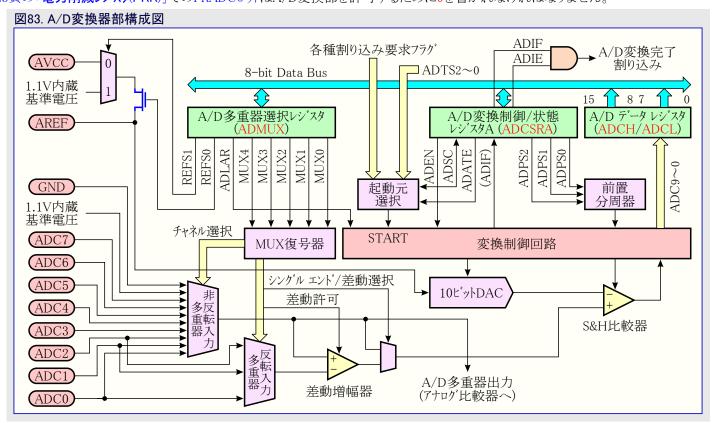
デバイスは16通りの差動電圧入力も支援します。7つの差動アナログ入力は共通反転端子(ADC1)を共用し、一方他のADC入力は非反転入力端子として選べます。(<mark>訳注:</mark>共通性から直前の2行追加)

このA/D変換器はA/D変換器への入力電圧が変換中に一定の値で保持されることを保証する採取&保持(S/H)回路を含みます。 A/D変換部の構成図は図83.で示されます。

A/D変換部には分離されたアナログ電源供給ピン(AVCC)があります。AVCCはVCCから±0.3Vよりも多く違ってはなりません。このピンの接続方法は137頁の「**雑音低減技術**」項をご覧ください。

公称1.1Vの内蔵基準電圧またはAVCCがチップ・上で提供されます。この基準電圧は雑音特性向上のため、コンテンサによってAREFピンで外部的にデカップ(雑音分離)できます。

23頁の「電力削減レシ、スタ(PRR)」でのPRADCビットはA/D変換部を許可するために0を書かれなければなりません。



### 操作

A/D変換部は逐次比較を通してアナログ入力電圧を10tiットのデジタル値に変換します。最小値はGNDを表し、最大値はAREFtevの電圧-1 LSBを表します。A/D多重器選択レジスタ(ADMUX)の基準電圧選択(REFS1,0)tiットへの書き込みにより、任意でAVCCまたは内部 1.1V基準電圧がAREFtevに接続できます。従ってこの内部基準電圧は雑音耐性を改善するためにAREFtevで外部コンデンサによってデカップ(雑音結合減少)ができます。

アナログ入力チャネルはADMUXのチャネル選択(MUX4~0)ビットへの書き込みによって選ばれます。GNDと固定基準電圧(1.1V内蔵基準電圧( $V_{BG}$ ))だけでなく、どのADC入力ピン(ADC7~0)もがA/D変換器のシングルエンパ入力として選べます。A/D変換部はA/D変換制御/状態レジスタA(ADCSRA)のA/D許可(ADEN)ビットの設定(1)によって(動作が)許可されます。基準電圧と入力チャネルの選択はADENが設定(1)されるまで実施しません。ADENが解除(0)されているとA/D変換部は電力を消費しないので、節電をする休止形態へ移行する前にA/D変換部をOFFに切り替えることが推奨されます。

A/D変換部はA/Dデータレジスタ(ADCH,ADCL)で示される10ビットの結果を生成します。既定では、この結果は右揃え(16ビットのビットの 側10ビット)で表されますが、ADMUXで左揃え選択(ADLAR)ビットを設定(1)することにより、任意で左揃え(16ビットのビット)を表せます。

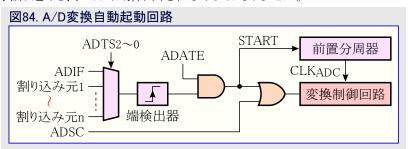
この結果が左揃え補正され、8ビットを越える精度が必要とされない場合はADCHを読むことで足ります。さもなければデータレジスタの内容が同じ変換に属すこと(からの結果)を保証するため、ADCLが初めに、次にADCHが読まれなければなりません。一度ADCLが読まれると、A/D変換器からのA/Dデータレジスタ(ADCH,ADCL)アクセスが阻止されます。これはADCLが読まれてしまい、ADCHが読まれる前に変換が完了すると、どちらのレジスタ(ADCH,ADCL)も更新されず、その変換からの結果が失われることを意味します。ADCHが読まれると、ADCH,ADCLへのA/D変換器アクセスが再び許可されます。

A/D変換部には変換完了時に起動できる自身の割り込みがあります。A/DデータレジスタへのA/D変換器アクセスがADCLとADCHの読み込み間で禁止されている場合、例えその変換結果が失われても割り込みは起動します。

# 変換の開始

単独変換はADCSRAで変換開始(ADSC)ビットに論理1を書くことによって開始されます。このビットは変換が進行中である限り、1に留まり、変換が完了されるとハードウェアによって解除(0)されます。変換が進行中に違う入力チャネルが選ばれると、A/D変換部はそのチャネル変更を実行する前に現在の変換を済ませます。

代わりに、変換は様々な起動元によって自動的に起動できます。自動起動はA/D変換制御/状態レジスタA(ADCSRA)のA/D変換自動起動許可(ADATE)ビットの設定(1)によって許可されます。起動元はA/D変換制御/状態レジスタB(ADCSRB)のA/D変換起動元選択(ADTS2~0)ビットの設定によって選ばれます(起動元の一覧についてはADTSビットの記述をご覧ください)。選んだ起動信号上に上昇端が起きると、A/D変換用前置分周器がリセットし、変換が開始されます。これは一定間隔での変換開始の方法を提供します。変換完了時、起動信号が未だ設定(1)されている場合、新規の変換は開始されません。変換中にこの起動信号上で別の上昇端が起きると、その端(エッジ)は無視されます。指定した割り込みが禁止またはステータスレジスタ(SREG)の全割り込み許可(I)ビットが解除(0)でも、割り込み要求フラケが設定(1)されることに注意してください。従って割り込みを起こさずに変換が起動できます。けれども次の割り込み要因で新規変換を起動するために、割り込み要求フラケは解除(0)されなければなりません。



起動元としてA/D変換完了割り込み要求フラグ(ADIF)を使うことは、A/D変換器に実行中の変換が完了されると直ぐに新規変換を開始させます。そのためA/D変換器は連続動作で動き、継続的な採取(変換)とA/Dデータレジスタ更新を行ないます。最初の変換はAD CSRAでADSCビットに論理1を書くことによって始めなければなりません。この動作でのA/D変換器はA/D変換完了割り込み要求フラグ(ADIF)が解除(0)されるかどうかに拘らず、連続的な変換を実行します。

自動起動が許可されている場合、ADCSRAのADSCビットに論理1を書くことによって単独変換を開始できます。ADSCは変換が進行中かを決めるためにも使えます。ADSCビットは変換がどう開始されたかに拘らず、変換中は1として読みます。

変換はA/D変換雑音低減機能の使用によっても開始され得ます。この機能はアイドル休止動作とA/D変換雑音低減休止動作中に変換を許可します。詳細については137頁の「**雑音低減機能**」をご覧ください。(<mark>訳注</mark>:共通性から2行追加)





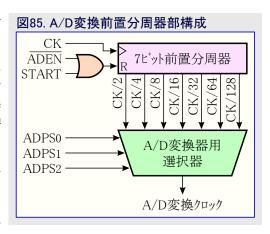
### 前置分周と変換タイミング

既定での逐次比較回路は最大分解能を得るのに50~200kHzの入力クロック周波数を必要とします。10ビットよりも低い分解能が必要とされるなら、A/D変換器への入力クロック周波数はより高い採取速度を得るために200kHzよりも高くできます。

A/D変換部は100kHz以上のどんなCPUクロックからも受け入れ可能なA/D変換クロック 周波数を生成する前置分周器を含みます。この前置分周はA/D変換制御/状態レジスタA(ADCSRA)のA/Dクロック選択(ADPS2~0)ビットによって設定されます。前置分周器はADCSRAでA/D許可(ADEN)ビットの設定(1)によってA/D変換部がONにされた瞬間から計数を始めます。前置分周器はADENビットが設定(1)される限り走行を保ち、ADENが0の時は継続的にリセットします。

ADCSRAのA/D変換開始(ADSC)ビットの設定(1)によってシングルェンド入力の変換を起動すると、その変換は直後の変換クロックの上昇端で始まります。

通常の変換は13変換クロック周期で行われます。A/D変換部がONされる(ADCSRAの ADEN=1)後の最初の変換はアナログ回路を初期化するために25変換クロック周期で行われます。

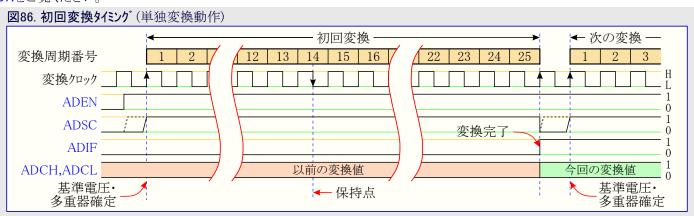


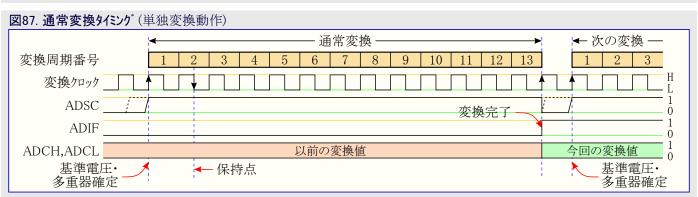
内部基準電圧がA/D変換器への入力として使われるとき、電圧の安定に一定時間を必要とします。安定(待機を)しない場合は初回変換後の最初の読み込み値は不正になるかもしれません。

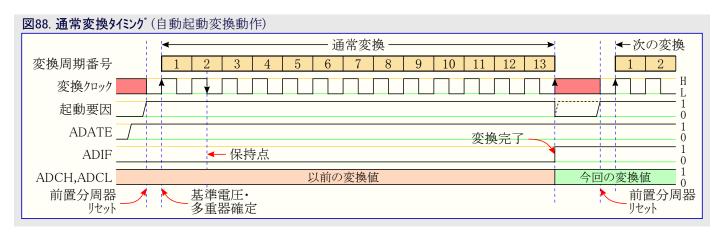
実際の採取&保持(保持開始点)は通常変換の開始後1.5変換クロック周期、初回変換の開始後13.5変換クロック周期で行われます。変換が完了すると、結果がA/Dデータレジ、スタ(ADCH,ADCL)に書かれ、ADCSRAのA/D変換完了割り込み要求フラグ(ADIF)が設定(1)されます。単独変換動作(ADATE=0)では同時にADCSRAのA/D変換開始(ADSC)ビットが解除(0)されます。その後にソフトウェアは再びADSCを設定(1)でき、新規変換は変換クロックの最初の上昇端で開始されます。

自動起動が使われると、前置分周器は起動要因発生時にリセットされます。これは起動要因から変換開始までの一定の遅延を保証します。この動作での採取&保持は起動要因となる信号の上昇後、2変換クロック周期で採取が行われます。同期化論理回路(端(エッジ)検出器)に対して、追加の3 CPUクロック周期が費やされます。A/D変換完了以外の要因からの自動起動を伴う差動動作を使うとき、各変換は25変換クロックを必要とします。これはA/D変換器が毎変換後、禁止そして再許可されなければならないからです。

連続変換動作(ADATE=1)では変換完了後直ちに新規変換が開始され、一方ADSCは1に留まります。変換時間の概要については表87.をご覧ください。







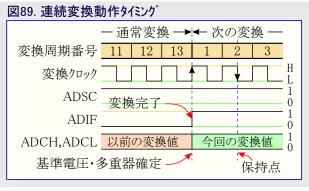


表87. A/D変換時間		
変換種別	保持点	変換時間
初回変換	13.5	25
シングルエント、入力通常変換	1.5	13
自動起動変換	1.5 (2)	13.5
* * ! # ! # ! # ! # ! # ! # ! # !	5 PP 11 2 2 2 2 15	). ).W/ 1.

注: 変換時間を除く各値は変換開始からの変換クロック数です。





### チャネル変更と基準電圧選択

A/D多重器選択レシ、スタ(ADMUX)のチャネル選択(MUX4~0)ピットと基準電圧選択(REFS1,0)ピットはCPUが乱順にアクセスするための一時レジスタを通して単独緩衝されます。これはチャネルと基準電圧の選択が変換中の安全なところでだけ行うのを保証します。チャネルと基準電圧の選択は変換が開始されるまで継続的に更新されます。一旦変換が始まると、A/D変換器に対して充分な採取/変換時間を保証するためにチャネルと基準電圧の選択は固定されます。継続的な更新は変換完了(ADCSRAのADIF=1)前の最後の変換クロック周期で再開します。ADCSRAの変換開始(ADSC)ピットが書かれた後の次の変換クロックの上昇端で変換が始まることに注意してください。従って使用者はADSC書き込み後、1変換クロック周期(経過)まで新しいチャネルまたは基準電圧選択値をADMUXに書かないことが推奨されます。

自動起動が使われる場合、起動要因の正確な時間は確定できません。変換が新規設定によって影響されるように制御するには ADMUXの更新時に特別な注意が祓われなければなりません。

ADCSRAのA/D許可(ADEN)とA/D変換自動起動許可(ADATE)の両方が1を書かれると、何時でも割り込みが起き得ます。この期間でADMUXが変更されると、使用者は次の変換が旧設定または新設定どちらが基準にされるかを知ることができません。ADMUXは次の方法で安全に更新できます。

- ADENまたはADATEが解除(0)されているとき。
- 変換開始後、最低1変換クロック周期経過後の変換中。
- •変換後から、変換起動元として使った割り込みフラグが解除(0)される直前まで。

これら条件の1つでADMUXを更新すると、新設定は次のA/D変換に影響を及ぼします。

JTAGインターフェースが許可されると、PORTF7~4のADC入力チャネル機能が無効にされます。48頁の**表42**.を参照してください。 (<mark>訳注</mark>:共通性のため上記行追加)

#### A/D入力チャネル

チャネル選択を変更する時に使用者は正しいチャネルが選ばれることを保証するために次の指針を守るべきです。

単独変換動作では常に変換を始める前にチャネルを選んでください。チャネル選択はADSCへの1書き込み後、1変換クロック周期で変更されるかもしれません。とは言え、最も簡単な方法はチャネル選択を変更する前に変換が完了するまで待つことです。

連続変換動作では常に最初の変換を始める前にチャネルを選んでください。チャネル選択はADSCへの1書き込み後、1変換クロック周期で変更されるかもしれません。とは言え、最も簡単な方法は最初の変換が完了するまで待ち、その後にチャネル選択を変更することです。既に次の変換が自動的に開始されているので、次の結果は直前のチャネル選択を反映します。それに続く変換は新しいチャネル選択を反映します。

#### A/D変換基準電圧

このA/D変換用の基準電圧(VREF)はA/D変換に対する変換範囲を示します。VREFを越えるシングル エンド入力チャネルは\$3FFで打ち切る符号に帰着します。VREFはAVCC、内部1.1V基準電圧、外部AREFt。ンのどれかとして選べます。

AVCCは受動型スイッチを通してA/D変換部に接続されます。内部1.1V基準電圧は内蔵基準(バンドギャップ)電圧(VBG)から内部増幅器を通して生成されます。どちらの場合でも外部AREFピンは直接的にA/D変換部へ接続され、AREFピンとGND間にコンデンサを接続することにより、基準電圧は雑音耐性をより高められます。VREF(電圧)は高入力インピーダンス電圧計とAREFピンで測定することもできます。VREFは高インピーダンス出力で、容量性負荷のみがシステム内で接続されるべきであることに注意してださい。

使用者がAREFt°ンに接続された固定電圧源にするなら、この外部電圧がその他の内部基準電圧と短絡してしまうため、使用者はこの応用内で他の基準電圧選択を使ってはなりません。外部電圧がAREFt°ンに印加されないなら、使用者は基準電圧選択としてAVCCと内部1.1V基準電圧間の切り替えができます。基準電圧源切り替え後の最初のA/D変換結果は不正確かもしれず、使用者はこの結果を破棄することが推奨されます。

# 雑音低減機能

このA/D変換部はCPUコアと他の周辺I/Oが誘導した雑音を削減するために休止形態中の変換を可能にする雑音低減機能が特徴です。この機能はA/D変換雑音低減動作とアイル動作で使えます。この機能を使うには次の手順が使われるべきです。

- 1. A/D変換部が許可(ADEN=1)され、変換中でない(ADSC=0)ことを確認してください。単独変換動作が選択(ADATE=0)され、且つ A/D変換完了割り込みが許可(ADIE=1)されていなければなりません。
- 2. A/D変換雑音低減(またはアイドル)動作に移行してください。一旦CPUが停止されてしまうと、A/D変換部は変換を始めます。
- 3. A/D変換完了前に他の割り込みが起こらなければ、A/D変換完了割り込みはCPUを起動してA/D変換完了割り込みルーチンを実行します。A/D変換完了前に他の割り込みがCPUを起動すると、その割り込みが実行され、A/D変換完了割り込み要求はA/D変換完了時に生成されます。CPUは新規SLEEP命令が実行されるまで活動動作に留まります。

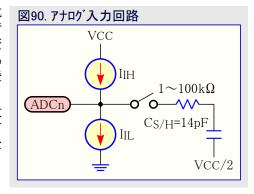
アイドル動作とA/D変換雑音低減動作を除く他の休止形態へ移行する時にA/D変換部が自動的にOFFへ切り替えられないことに注意してください。使用者は余分な消費電力を避けるため、このような休止形態へ移行する前にADENへ0を書くことが推奨されます。

### アナログ入力回路

シングル エント・入力チャネルのアナログ回路は**図90**.で図示されます。ADCnに印加したアナログ(信号)源はそのチャネルがADC入力として選ばれているかどうかに拘らず、ピン容量とそのピンの漏れ電流に左右されます。そのチャネルが選ばれると、(アナログ信号)源は直列抵抗(入力経路の合成抵抗)を通してS/Hコンデンサを駆動しなければなりません。

A/D変換部は概ね10kΩ若しくはそれ以下の出力インピーダンスのアナログ信号用に最適化されています。このような(アナログ信号)源が使われるなら、採取時間は無視してもよいでしょう。より高いインピーダンスの(アナログ信号)源が使われる場合、採取時間は広範囲に変化し得るS/Hコンデンサを充電するために(アナログ信号)源がどれくらいの時間を必要とするかに依存します。必要とされるS/Hコンデンサへの充放電を最小とするため、使用者は緩やかに変化する低インピーダンス(アナログ信号)源だけを使うことが推奨されます。

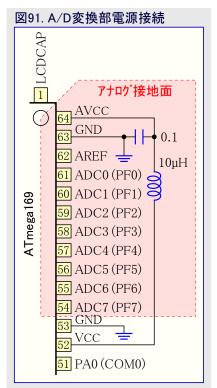
特定できない信号の渦からの歪を避けるために、どのチャネルに対してもナイキスト周波数 (fADC/2)よりも高い信号成分が存在すべきではありません。使用者はADC入力として信号を印加する前に低域通過濾波器(p-n° $\alpha$ , D{ $\mu$ 0)で高い周波数成分を取り除くことが推奨されます。



#### アナログ雑音低減技術

デバイス内外のデジタル回路がアナログ測定の精度に影響を及ぼすかもしれないEMIを発生します。精密な変換精度が必要な場合、次の技法を適用することによって雑音レベルを低減できます。

- 1. アナログ信号経路を可能な限り最短にしてください。アナログ信号線がアナログGND面上を走ることに注意し、高速切り替えデジタル信号線から充分離すことを守ってください。
- 2. デバイスのAVCCピンは図91.で示されるようにLC濾波器を経由してデジタル供給電圧(VCC) に接続されるべきです。
- 3. CPUからの誘導雑音を低減するためにA/D変換の雑音低減機能を使ってください。
- 4. どれかのADCポート ピンがデジタル出力として使われる場合、これらが変換進行中に切り替わらないことが重要です。







### A/D変換の精度定義

シングル エント、入力電圧のnビットA/D変換はGNDとVREF間を2nで直線的に変換します。最低値符号は0として読み、最高値符号は2n-1として読みます。以下の各種パラメータは理想状態からの偏差を表します。

#### • 変位(オフセット)誤差 - 図92.

最初の遷移点(\$000から\$001)で理想遷移点(差0.5 LSB)と比べた偏差です。理想値は0 LSBです。

## • 利得誤差 - 図93.

変位誤差補正後の最後の遷移点(\$3FEから\$3FF)で理想遷 移点(最大差1.5 LSB以下)と比べた偏差です。理想値は0 LSBです。

#### • 積分非直線性誤差 (INL) - 図94.

変位誤差と利得誤差補正後の全ての遷移点で理想遷移点と比べた最大偏差です。理想値は0LSBです。

### • 微分非直線性誤差 (DNL) - 図95.

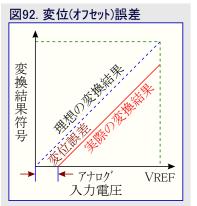
実際の符号の幅(隣接する2つの遷移点間)で理想符号幅(1 LSB)と比べた最大偏差です。理想値は0 LSBです。

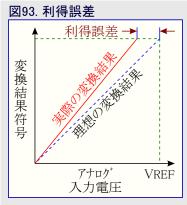
### • 量子化誤差

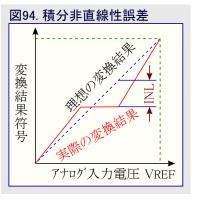
有限数の符号で入力電圧を量子化するため、1 LSB幅となる入力電圧範囲は同じ値の符号になります。この値は常に±0.5 LSBです。

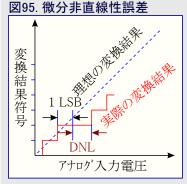
#### • 絶対精度

補正しない全ての遷移点で理想遷移点と比べた最大偏差です。これは、変位誤差、利得誤差、差動誤差、非直線誤差の影響の合成です。理想値は±0.5 LSBです。









### A/D変換の結果

変換完了(ADIF=1)後、変換結果はA/Dデータレジスタ(ADCH, ADCL)で得られます。

シングル エンド入力変換での結果は右式で示されます。 $V_{IN}$ は選んだ入力ピンの電圧で、 $V_{REF}$ は選んだ基準電圧です(139頁の表89.と表90.をご覧ください)。\$000はアナログGNDを表し、\$3FFは選んだ基準電圧-1 LSBを表します。

$$ADC = \frac{V_{IN} \times 1024}{V_{REF}}$$

差動チャネルが使われると、その結果は右式で示されます。VPOSは非反転入力ピンの電圧、VNEGは反転入力ピンの電圧、VREFは選んだ基準電圧です。この結果は-512(\$200)~+511(\$1FF)の2の補数形式で示されます。使用者が結果の符号を素早く検査したいなら、結果のMSB(ADCHの

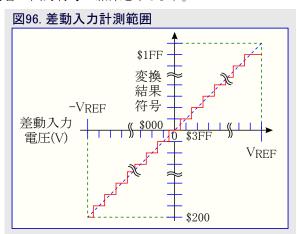
$$ADC = \frac{(V_{POS} - V_{NEG}) \times 512}{V_{REF}}$$

ADC9)を読むことで充分なことに注目してください。このビットが1ならばその結果は一、0ならばその結果は+です。**図96.**は差動入力範囲の符号化を示します。

表88.は差動入力チャネルの組(ADCn-ADCm)が基準電圧(VREF)で選ばれた場合の出力符号の結果を示します。

#### 表88. 入力電圧と出力符号の関係

VADCn	読み出し符号	対応する10進値
VADCm+VREF	\$1FF	511
V <sub>ADCm</sub> +(511/ <sub>512</sub> )V <sub>REF</sub>	\$1FF	511
V <sub>ADCm</sub> +(510/ <sub>512</sub> )V <sub>REF</sub>	\$1FE	510
<b>`</b>	>	>
VADCm+(1/512)VREF	\$001	1
VADCm	\$000	0
V <sub>ADCm</sub> -(1/ <sub>512</sub> )V <sub>REF</sub>	\$3FF	-1
<b>`</b>	>	}
VADCm <sup>-(511/512)</sup> VREF	\$201	-511
VADCm-VREF	\$200	-512



例: ADMUX=\$ED(ADC3とADC2差動入力、V<sub>REF</sub>=1.1V、左揃え)で、ADC3が300mV、ADC2が500mVの場合、A/D変換値は 512×(300-500)÷1100=-93=\$3A3、従って、ADCH=\$D8、ADCL=\$C0、ADLAR=0なら、ADCH=\$03、ADCL=\$A3。

# A/D変換用レジスタ

### A/D多重器選択レジスタ (ADC Multiplexer Select Register) ADMUX

ピット	7	6	5	4	3	2	1	0	
(\$7C)	REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0	ADMUX
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

#### • ビット7,6 - REFS1,0 : 基準電圧選択 (Reference Select Bits 1, 0)

これらのビットは表89.で示されるようにA/D変換器の基準電圧を選びます。これらのビットが変換中に変更されると、その変更は変換が 完了する(ADCSRAのADIF=1)まで実施しません。AREFピンに外部基準電圧が印加される場合、内部の基準電圧が使われてはなりま せん。

表ぬ	A/D 変換部の	基準電圧選択
AVOD.	A/ U & THE BUU	

REFS1	REFS0	基準電圧
0	0	AREFtシの外部基準電圧(AVCCと内部1.1V基準電圧は切り離されます。)
0	1	AVCC (内部1.1V基準電圧は切り離されますが、AREFにデカップ用コンデンサが接続できます。)
1	0	(予約)
1	1	内部1.1V基準電圧(AVCCは切り離されますが、AREFにデカップ用コンデンサが接続できます。)

#### • ビット5 - ADLAR: 左揃え選択 (ADC Left Adjust Result)

ADLARL'ットはA/Dデータレジスタ内の変換結果の配置に影響を及ぼします。結果を左揃えにするにはADLARに1を書いてください。さもなければ結果は右揃えです。ADLARL'ットの変更はどんな進行中の変換にも拘らず、直ちにA/Dデータレジスタの内容に影響を及ぼします。このビットの完全な記述については141頁の「A/Dデータレジスタ」をご覧ください。

#### • ビット4~0 - MUX4~0: A/Dチャネル選択 (Analog Channel Select Bits 4~0)

これらビットの値はA/D変換器にどのアナログ入力の組み合わせが接続されるかを選びます。詳細については**表90**.をご覧ください。これらのビットが変換中に変更される場合、その変更は変換が完了する(ADCSRAのADIF=1)まで実施しません。

表90. アナログ入力チャネル選択

MUX4~0	シングル エンド	差動	入力	MUVALO	シングル エント゛	差動	<u></u> 入力
WUX4~0	入力	非反転入力	反転入力	MUX4~0	入力	非反転入力	反転入力
00000	ADC0			10000		ADC0	ADC1
00001	ADC1			10001		ADC1	ADC1
00010	ADC2			10010		ADC2	ADC1
00011	ADC3			10011		ADC3	ADC1
00100	ADC4			10100		ADC4	ADC1
00101	ADC5			10101		ADC5	ADC1
00110	ADC6			10110		ADC6	ADC1
00111	ADC7			10111		ADC7	ADC1
01000				11000		ADC0	ADC2
01001				11001		ADC1	ADC2
01010				11010		ADC2	ADC2
01011		(利用不可)		11011		ADC3	ADC2
01100		(4,11/11/1/1/1/1/1/		11100		ADC4	ADC2
01101				11101		ADC5	ADC2
01110				11110	1.1V(V <sub>BG</sub> )		
01111				11111	0V(GND)		





### A/D変換 制御/状態レジスタA (ADC Control and Status Register A) ADCSRA

ピット	7	6	5	4	3	2	1	0	
(\$7A)	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	ADCSRA
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

#### • ビット7 - ADEN: A/D許可 (ADC Enable)

このビットにlを書くことがA/D変換部(動作)を許可します。0を書くことによってA/D変換部は(電源が)OFFされます。変換が進行中に A/D変換部をOFFにすることはその変換を(途中)終了します。

#### • ビット6 - ADSC: A/D変換開始 (ADC Start Conversion)

単独変換動作で各変換を始めるにはこのビットへ1を書いてください。連続変換動作で最初の変換を始めるにはこのビットへ1を書いてください。A/D変換部が許可される(ADEN=1)と同時にADSCが書かれるか、またはA/D変換部が許可されてしまった後にADSCが書かれた後の初回変換は、通常の13に代わって25変換クロック周期で行います。この初回変換はA/D変換部の初期化を実行します。ADSCは変換が進行中である限り1として読みます。変換が完了すると0に戻ります。このビットへの0書き込みは無効です。

#### • ビット5 - ADATE: A/D変換自動起動許可(ADC Auto Trigger Enable)

このビットが1を書かれると、A/D変換の自動起動が許可されます。A/D変換器は選んだ起動信号の上昇端で変換を開始します。この起動元はA/D変換制御/状態レジスタB(ADCSRB)のA/D変換起動要因選択( $ADTS2\sim0$ )ビット設定によって選ばれます。

#### • ビット4 - ADIF: A/D変換完了割り込み要求フラグ (ADC Interrupt Flag)

A/D変換が完了し、A/Dデータレジスタが更新されると、このフラケが設定(1)されます。ステータスレジスタ(SREG)の全割り込み許可(I)ビットと A/D変換完了割り込み許可(ADIE)ビットが設定(1)されていれば、A/D変換完了割り込みが実行されます。対応する割り込み処理へクタを実行する時にADIFはハートヴェアによって解除(0)されます。代わりにこのフラケに論理1を書くことによってもADIFは解除(0)されます。 ADCSRAで読み-変更-書き(リート・モディファイライト)を行うと、保留中の割り込みが禁止され得ることに注意してください。これはSBI,CBI 命令が使われる場合にも適用されます(訳注:アトレス範囲外のため、本行は不適切です)。

### • ビット3 - ADIE: A/D変換完了割り込み許可(ADC Interrupt Enable)

このビットが1を書かれ、SREGの全割り込み許可(I)ビットが設定(1)されていると、A/D変換完了割り込みが活性に(許可)されます。このビットが解除(0)されると、この割り込みは禁止されます(訳注:共通性のため本行追加)。

#### • ビット2~0 - ADPS2~0: A/D変換クロック選択 (ADC Prescaler Select Bits)

これらのビットはXTAL(システム)周波数とA/D変換部への入力クロック間の分周値を決めます。

表91. A/D変換クロッ	<b>ク選択</b> (CK=シ	ステム クロック)						
ADPS2	0	0	0	0	1	1	1	1
ADPS1	0	0	1	1	0	0	1	1
ADPS0	0	1	0	1	0	1	0	1
A/D変換クロック	CK/2	CK/2	CK/4	CK/8	CK/16	CK/32	CK/64	CK/128

### A/D変換 制御/状態レジスタB (ADC Control and Status Register B) ADCSRB

ピット	7	6	5	4	3	2	1	0	_
(\$7B)	-	ACME	-	_	-	ADTS2	ADTS1	ADTS0	ADCSRB
Read/Write	R	R/W	R	R	R	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

#### • ビット7,5~3 - Res: 予約 (Reserved Bits)

これらのビットは将来の使用に対して予約されています。将来のデバイスとの共通性のため、ADCSRBが書かれるとき、これらのビットは0が書かれなければなりません。

#### ● ビット2~0 - ADTS2~0: A/D変換自動起動要因選択 (ADC Auto Trigger Source)

A/D変換制御/状態レシ、スタA(ADCSRA)のA/D変換自動起動許可(ADATE) ビットが1を書かれると、これらのビットの値はどの起動元が A/D変換を起動するのかを選びます。ADATEが解除(0)されると、ADTS2~0設定は無効です。変換は選んだ割り込みフラグの上昇端 によって起動されます。解除(0)されている起動元から設定(1)されている起動元への切り替えが、起動信号上に上昇端を生成することに注意してください。ADCSRAのA/D許可(ADEN)ビットが設定(1)されているなら、これが変換を開始させます。連続変換動作(ADTS 2~0=0)への切り替えはA/D変換完了割り込み要求フラグが設定(1)されていても、起動事象を引き起こしません。(訳補:起動元一覧表は次頁参照)

表92. A/[	表92. A/D変換自動起動元選択						
ADTS2	ADTS1	ADTS0	起動元				
0	0	0	連続変換動作				
0	0	1	アナログ比較器				
0	1	0	外部割り込み要求0				
0	1	1	タイマ/カウンタ0比較一致				
1	0	0	タイマ/カウンタ0溢れ				
1	0	1	タイマ/カウンタ1比較B一致				
1	1	0	タイマ/カウンタ1溢れ				
1	1	1	タイマ/カウンタ1捕獲要求				

### A/Dデータレジスタ (ADC Data Register) ADCH,ADCL

	ADLAR=0	時							
ピット	15	14	13	12	11	10	9	8	_
(\$79)	_	-	-	-	-	-	ADC9	ADC8	ADCH
Read/Write	R	R	R	R	R	R	R	R	
初期値	0	0	0	0	0	0	0	0	
ピット	7	6	5	4	3	2	1	0	
(\$78)	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0	ADCL
Read/Write	R	R	R	R	R	R	R	R	
初期値	0	0	0	0	0	0	0	0	
	ADLAR=1	時							
	15	14	13	12	11	10	9	8	_
	ADC9	ADC8	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADCH
	7	6	5	4	3	2	1	0	
	ADC1	ADC0	-	-	-	-	-	_	ADCL

A/D変換が完了すると、その結果がこれら2つのレシ、スタで得られます。 差動チャネルが使われると、結果は2の補数形式で表されます (**訳補**:本行追加)。

ADCLが読まれると、A/DデータレジスタはADCHが読まれるまで更新されません。従ってこの結果が左揃えで、且つ8ビット(差動入力チャネルについては符号+7ビット)を越える精度が必要とされないなら、ADCHを読むことで用が足ります。さもなければADCLが先に、その後にADCHが読まれなければなりません。

A/D多重器選択レジスタ(ADMUX)の左揃え選択(ADLAR)ビットとA/Dチャネル選択(MUX4~0)ビットは本レジスタから結果を読む方法に影響を及ぼします。ADLARが設定(1)ならば結果は左揃えにされます。ADLARが解除(0:既定)ならば結果は右揃えにされます。

• ADC9~0: A/D変換結果 (ADC Conversion result)

これらのビットは138頁の「A/D変換の結果」で詳述されるように変換での結果を表します。

# デッタル入力禁止レシ、スタ0 (Digital Input Disable Register 0) DIDR0

ピット	7	6	5	4	3	2	1	0	
(\$7E)	ADC7D	ADC6D	ADC5D	ADC4D	ADC3D	ADC2D	ADC1D	ADC0D	DIDR0
Read/Write	R/W								
初期値	0	0	0	0	0	0	0	0	

#### • ビット7~0 - ADC7D~ADC0D: ADC7~0 デジタル入力禁止 (ADC7~0 Digital Input Disable)

このビットが論理1を書かれると、対応するADCnピンのデジタル入力緩衝部が禁止されます。このビットが設定(1)されると、対応するポート入力レジスタのビット(PINx)は常に0として読みます。アナログ信号がADCnピンに印加され、そのピンからのデジタル入力が必要とされない時にデジタル入力緩衝部での消費電力を削減するため、そのビットは論理1を書かれるべきです。





# LCD制御器

このLCD制御器/駆動部は4つまでの共通電極と25までのセグメント電極の受動型単色液晶(LCD)用です。

#### 特徴

- 25セグメントと4共通電極の表示能力
- スタティック(1/1), 1/2, 1/3, 1/4 デューティの支援
- スタティック(1/1), 1/2, 1/3 バイアスの支援
- 1つの外部コンデンサのみ必要なチップ上のLCD供給電源
- 低消費電力用のパワーセーブ動作で表示可能
- 低電力波形をソフトウェアで選択可能
- フレーム 周波数の柔軟な選択
- システム クロックまたは外部非同期クロック元間のソフトウェア選択
- LCD寿命を増やす(延ばす)ための等しい吸い込みと吐き出しの(駆動)能力
- 表示値更新または休止形態からの起動用にLCD割り込みが使用可能
- •表示器駆動に不必要なセグメントとコモン ピンは標準I/Oピンとして使用可能
- 表示データのラッチによる完全に自由なレジスタの更新

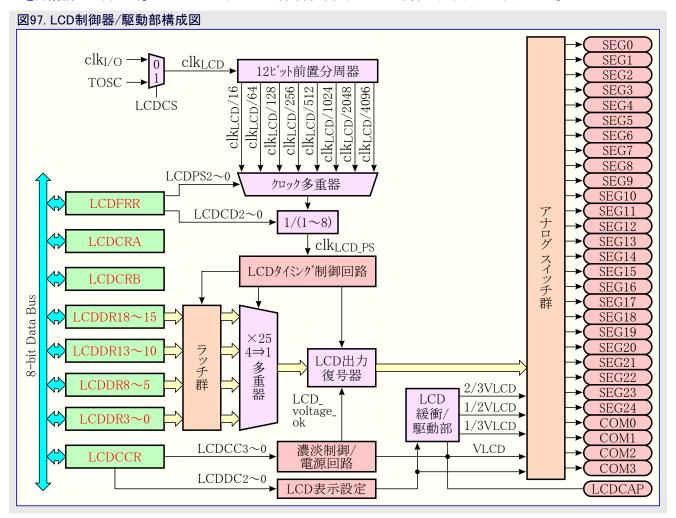
#### 概要

LCD制御器/駆動部の簡単化した構成図は図97.で示されます。I/Oピンの実際の配置については2頁の「ピン配置」を参照してください。

LCDは表示または非表示にできる様々なセグメント(ピクセルまたは完全なシンボル)から成ります。セグメントには2つの電極とその間に液晶があります。 閾値電圧以上の電圧が液晶を横切って印加されると、セグメントは可視(見えるよう)になります。

表示器を品質劣化する液晶の電気泳動効果を避けるため、この電圧は切り替えなければなりません。従ってセグメントを横切る(駆動する)波形にはDC成分があってはなりません。

23頁の「電力削減レジスタ(PRR)」でのPRLCDビットはLCD部を許可するために0を書かれなければなりません。



#### 定義

LCDを説明する時に様々な名称が使われます。**表93**. での定義が本文書を通して使われます。

表93. 用語定義						
用語	意味					
セグメント	ON/OFFできる最も小さな表示要素(ピクセル)。					
コモン	セグメント電極に接続される、いくつかのセグメントを示します。					
デューティ	1/(実際のLCD表示器での共通電極数)。					
バイアス	1/(LCD表示器駆動に使う電圧値数-1)。					
フレーム速度	1秒間にLCDセグメントが選択駆動される回数。					

#### LCDクロック

LCD制御器は内部同期または外部非同期のクロック元でクロック駆動できます。クロック元(clk\_CD)は既定でシステム クロックと等しいclk\_I/Oです。LCD制御レシ、スタB(LCDCRB)でLCDクロック選択(LCDCS)ビットが論理1を書かれると、クロック元はTOSC1ピンから行われます。 このクロックはLCDタイミング、精度、故にLCDセグメントをよぎる最小DC変位(オフセット)電圧を得るために安定でなければなりません。

#### LCD前置分周器

この前置分周器は12ビットのリプルキャリー カウンタと1~8クロック分周器から成ります。LCD前置分周選択(LCDPS2~0)ビットは16,64,128,256,512,1024,2048または4096で分周したclkLCDを選びます。

細かな速度分解能が必要とされるなら、クロックを更に1~8分周するのにLCDクロック分周(LCDCD2~0)ビットが使えます。 このクロック分周器からの出力(clk\_CD ps)は、LCDタイミングに関するクロック元として使われます。

#### **LCD**\

表示メモリは各共通電極毎に群化したI/Oレジスタを通して利用可能です。表示メモリ内のビットが1を書かれると、対応するセグメントが活動(表示駆動:ON)にされ、表示メモリ内のビットが0を書かれると、非活動(非表示駆動:OFF)にされます。

セグメントを活動(表示駆動)にするためには或る閾値以上の絶対電圧が印加されなければなりません。これは対応するCOMピンとSEG ピンで逆相になる出力電圧を出すことによって行われます。複数コモンの表示器では1つ(1/2バイアス)または2つ(1/3バイアス)の追加電圧値が印加されなければなりません。さもなければCOM0の非活動(非表示)セグメントは全ての非選択コモンに対して活動(表示)にされるでしょう。

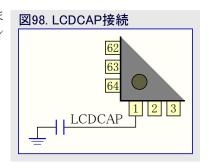
COM0指定は指定しないCOM線に比するCOM0の大振幅出力の逆相駆動によってフレームを開始します。指定したCOM0内の非活動(表示)セグメントは同相で、活動(表示)セグメントは大振幅逆相です。波形図については145頁の「動作種別」を参照してください。LCD データ レジ、スタ3~0(LCDDR3~0)からラッチしたデータは復号器内へ多重化されます。復号器はLCDタイミングで制御され、出力波形を生成するためにアナログ、スィッチを制御する信号を設定します。次にCOM1が指定され、LCDDR8~5からラッチしたデータが復号器に入力されます。この指定はコモン数(デューティ)に従って全てのCOM線が指定されるまで継続します。表示データは新規フレーム開始前にラッチされます。

#### LCD濃淡制御/電源回路

出力波形のピーク値(V<sub>LCD</sub>)がLCDの濃淡を決めます。V<sub>LCD</sub>はVCCと無関係に2.6~3.35V間でソフトウェアによって制御されます。V<sub>LCD</sub>が目標値に達してしまうまで内部信号はLCDへの出力を禁止します。

#### **LCDCAP**

外部コンデンサ(代表値: $>0.47\mu$ F)は**図98**.で示されるようにLCDCAPピンに接続されなければなりません。このコンデンサはLCD電力( $V_{LCD}$ )に対して蓄電器として動作します。大容量は $V_{LCD}$ のリプルを減少しますが、 $V_{LCD}$ が目標値に達するまでの時間を増やします。



# LCD緩衝/駆動部

中間電圧値は緩衝/駆動部から生成されます。緩衝部はLCD濃淡制御レジスタ(LCDCCR)のLCD表示器設定(LCDDC2~0)によって指定された時間分、活性(駆動)です。その後、緩衝部はOFFされ、LCD出力ピンはHi-Zにされます。短い駆動時間は電力消費を削減しますが、高い内部抵抗や容量の表示器は充分な濃淡に達するために長い駆動時間を必要とするかもしれません。





### 表示の必要条件

複数のCOMピンを使う時にLCD駆動部がLCDピンでの各電圧遷移に関して切り替えることができる最大時間は前置分周されたLCDク ロック(clk\_CD\_PS)の50%です。フリッカ(ちらつき)を避けるため、30Hz以上のフレーム速度を保持、従って1/2または1/4デューティ時は約2ms、1/3デューティ時は約2.7msの最大駆動時間を与えることが推奨されます。故に満足な濃淡を得るにはCOMピン数に依存するLCD表示器の全セグメントが2または2.7ms内で完全に充放電できなければなりません。

(<mark>訳補</mark>) 30Hzの1周期は約33.3msです。1/2デューティは1/4デューティの変形と考えます。すると1/4デューティは8分割駆動/1フレーム、1/3デューティは6分割駆動/1フレームなので、1駆動時間は各々4.167msと5.556msになります。この時間は $clk_{LCD\_PS}$ の50%なので各々半分の時間で上記数値表記になります。

### 消費電力の最小化

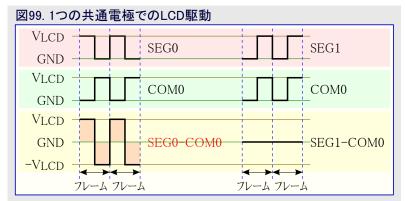
LCD駆動部が最小で切り替わる時間の割合を保つことにより、LCD駆動部の消費電力を最小化できます。これは受け入れ可能な最低アルーム速度を使って可能ならば低電力波形を使うことで達せられます。駆動時間は個々のLCD駆動部と表示器間の組み合わせによる製品に対して、或る視野範囲を許す特定の表示器について満足な濃淡を達成する設定に保たれるべきです。最高LCD電圧設定のいくつかはVCCが2.0V未満の時に高電力消費に終わるかもしれないことに注意してください。推奨最大LCD電圧は2×(VCC-0.2)Vです(訳補:×2はCOMとSEG間の両相駆動(駆動図での±VLCD)のため)。

# 動作種別

# スタティック(1/1)テ゛ューティとスタティック(1/1)ハ゛イアス

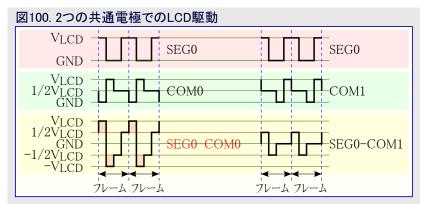
LCD上の全セグメントが1つの共通電極を持つなら、各セグメントは独自(専用)の電極を持たなければなりません。

この表示種別は**図99**.に示される波形で駆動されます。SEG 0-COM0はON(表示)セグメントをよぎる電圧で、SEG1-COM0 はOFF(非表示)セグメントをよぎる電圧です。



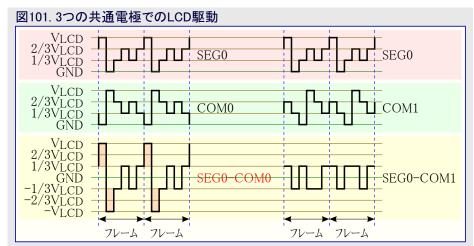
# 1/2デューティと1/2バイアス

2つの共通電極のLCD(1/2デューティ)についてはセグメントを個別に制御するために、より複雑な波形が使われなければなりません。例え1/3バイアスが選べても、それらの表示器に対して1/2バイアスが最も普通です。波形は図100.で示されます。SEG0-COM0はON(表示)セグメントをよぎる電圧で、SEG0-COM1はOFF(非表示)セグメントをよぎる電圧です。



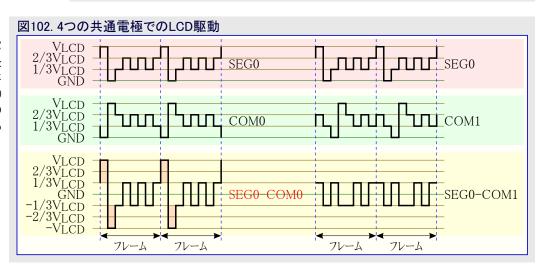
## 1/3デューティと1/3バイアス

1/3バイアスは一般的に3つの共通電極のLCD (1/3デューティ)に対して推奨されます。波形は 図101.で示されます。SEGO-COM0はON(表示)セグメントをよぎる電圧で、SEGO-COM1は OFF(非表示)セグメントをよぎる電圧です。



# 1/4デューティと1/3バイアス

1/3ハイアスは4つの共通電極のLC D表示器(1/4デューティ)に対して任 意選択です。波形は図102.で示さ れます。SEG0-COM0はON(表示) セグメントをよぎる電圧で、SEG0-CO M1はOFF(非表示)セグメントをよぎる 電圧です。

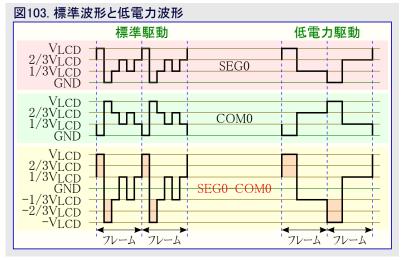






## 低電力波形

切り替えの動き(数)、即ち消費電力を削減するために低電力波形選択(LCDAB)ビットを1に書くことによって低電力波形を選べます。低電力波形はDC電圧ゼロを達成するために同じ表示データで2つの連続フレームを必要とします。従ってデータのラッチと割り込み要求フラケは全ての第2フレームだけで設定されます。既定(標準)波形と低電力波形は1/3デューティ、1/3バイアスに関して図103.で示されます。他のデューティとバイアス選択に対してもその効果は同様です。



## 休止形態でのLCD動作

同期LCDクロックが選択(LCDCS=0)されると、どのクロック元でもアイトル動作とハプワーセーフ・動作でLCD表示は動作します。

システム クロック元として校正付き内蔵RC発振器が選ばれると、LCDCSビットを1に書くことにより、LCDクロックとしてTOSC1からの非同期クロックを選べます。アイドル動作、A/D変換雑音低減動作、パワーセーブ動作でLCDは動作します。

ASSRでEXCLKが1を書かれ、非同期クロックが選ばれると、外部クロック入力緩衝部が許可され、外部クロックが32kHzクリスタルに代わってタイマ発振器1(TOSC1)ピンで入力できます。より多くの詳細については97頁の「タイマ/カウンタ2の非同期動作」をご覧ください。

同期LCDクロック選択でA/D変換雑音低減動作、パワーダウン動作、スタンバイ動作へ移行する前に使用者はLCDを禁止しなければなりません。148頁の「LCDの禁止」を参照してください。

## 無表示動作

LCD無表示(LCDBL)が1を書かれると、LCDは現在のフレーム完了後、無表示にされます。全てのSEGピンとCOMピンはGNDに接続され、LCDを放電します。表示メモリは保護されます。緩やかな(徐々の)表示消去やセグメントをよぎるDC電圧を避けるため、LCDを禁止する前に表示プランキング(無表示)が使われるべきです。

#### ポートの遮蔽

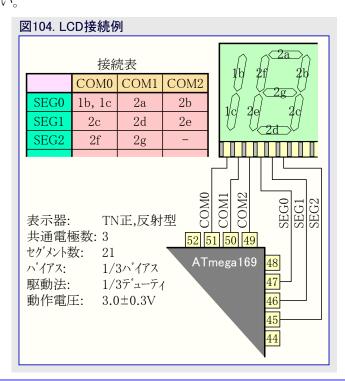
25セグメント電極未満のLCDについては未使用ピンのいくつかを遮蔽し、その代わりに通常ポート ピンとしてそれらを使うことが可能です。詳細については表95.を参照してください。未使用コモン ピンは自動的にポート ピンとして設定されます。

## LCDの使用

次節はLCDを使う方法を記述します。

#### LCDの初期化

LCDを許可する前にいくつかの初期化が実行されなければなりません。通常、この初期化手順はフレーム速度、デューティ、ハーイアス、ポート 遮蔽の設定から成ります。初期LCD濃淡に設定されますが、動作中に調整もできます。 例として次のLCDを考察してください。



```
アセンブリ言語プログラム例
LCD_INI:
            LDI
                   R16, (1<<LCDCS) | (1<<LCDMUX1) | (1<<LCDPM2) ;32kHz/りリスタル発振器、1/3バイアス、1/3デューティ
                                                           ;SEG21~24標準ポート ピン設定
            STS
                   LCDCRB, R16
                   R16, (1<<LCDCD2) | (1<<LCDCD1)
                                                           ;前置16分周、クロック分周7分周により、
            LDI
            STS
                   LCDFRR, R16
                                                           ;フレーム速度49Hz設定
                   R16, (1<<LCDC1) | (1<<LCDCC3) | (1<<LCDCC2) | (1<<LCDCC1)
            LDI
                                                           ;125µs駆動、出力電圧3.3V設定
            STS
                   LCDCCR, R16
                   R16, (1<<LCDEN)
                                                           ;LCD許可、標準波形、割り込み禁止
            LDI
                   LCDCRA, R16
            STS
                                                           ;呼び出し元へ復帰
            RET
C言語プログラム例
void LCD_Init(void);
    LCDCRB = (1<<LCDCS) | (1<<LCDMUX1) | (1<<LCDPM2); /* 32kHz/リスタル発振器、1/3バイアス、1/3デューティ*/
                                                 /* SEG21~24標準ポート ピン設定 */
   LCDFRR = (1 << LCDCD2) | (1 << LCDCD1);
                                                 /* 前置16分周、クロック7分周でフレーム速度49Hz */
    LCDCCR = (1 < LCDC1) | (1 < LCDCC3) | (1 < LCDCC2) | (1 < LCDCC1);
                                                 /* 125µs駆動、出力電圧3.3V設定 */
    LCDCRA = (1 << LCDEN);
                                                  /* LCD許可、標準波形、割り込み禁止 */
}
```

注: 5頁の「コート'例について」をご覧ください。

再初期化が行われる前にLCD制御器/駆動部は禁止されるべきです。





# LCDの更新

表示メモリ(LCDDR0~)、LCD無表示(LCDBL)、低電力波形選択(LCDAB)と濃淡制御(LCDCCR)は全ての新規フレームに先立ってラッチされます。これらのLCDレジ、スタ位置書き込みに制限はありませんが、更新が実行中の間にデータがラッチされると、LCDデータ更新は2つのフレーム間に分かれるかもしれません。これを避けるため、データがラッチされた直後に表示メモリ、LCD無表示、波形選択、濃淡制御を更新するのに割り込みルーチンが使えます。

以下の例でCOM0のSEG4とCOM1のSEG10だけがフレームからフレームで変化したセグメントとの仮定です。データは単純にR20とR21に格納されます。

```
アセンブリ言語プログラム例
LCD UPD:
           STS
                  LCDDRO, R20
                                                       ;表示メモリのみ更新
           STS
                  LCDDR6, R21
           RET
                                                       ;呼び出し元へ復帰
C言語プログラム例
void LCD_update(unsigned char data1, data2);
   LCDDR0 = data1;
                                               /* 表示メモリのみ更新 */
   LCDDR6 = data2;
                                               /* */
}
注: 5頁の「コード例について」をご覧ください。
```

# LCDの禁止

いくつかの応用でLCDを禁止することが必要かもしれません。これはクロック元がない(停止する)パワーダウン動作へMCUが移行する場合です。

LCDは禁止されるのに先立って完全に放電されるべきです。どのセゲメントもDC電圧がよぎったままにされるべきではありません。これを成し遂げる最良の方法は全てのCOMピンとSEGピンをGNDに駆動するLCD無表示(Blanking)機能を使うことです。

LCDが禁止されると(標準)ポート機能が再び活性(有効)にされます。従って使用者はLCD電極に接続したポート ピンがHi-ZまたはLow 出力(シンク:吸い込み)のどちらかであるのを検査しなければなりません。

```
アセンブリ言語プログラム例
LCD_DIS:
           LDS
                  R16, LCDCRA
                                                         ;LCDCRA値を取得
           SBRS
                  R16, LCDIF
                                                         ;割り込み有りでスキップ
                                                         ;割り込み(フレーム開始)まで待機
           RJMP
                  LCD_DIS
                  R16, (1<<LCDEN) | (1<<LCDIF) | (1<<LCDBL)
           LDI
                                                         ;無表示(放電)、割り込みフラグ解除
           STS
                  LCDCRA, R16
LCD D W
           LDS
                  R16, LCDCRA
                                                         ;LCDCRA値を取得
           SBRS
                  R16, LCDIF
                                                         ;割り込み有りでスキップ。
                                                         ;割り込み(フレーム開始)まで待機
                  LCD_D_W
           RJMP
           LDI
                  R16, (0<<LCDEN)
                                                         ;LCD禁止値を取得
           STS
                  LCDCRA, R16
                                                         ;LCD禁止
                                                         ;呼び出し元へ復帰
           RET
C言語プログラム例
void LCD_disable(void);
   while (!(LCDCRA & (1<<LCDIF)));
                                                /* 割り込み(フレーム開始)まで待機 */
   LCDCRA = (1 << LCDEN) | (1 << LCDIF) | (1 << LCDBL);
                                                /* 無表示(放電)、割り込みフラグ解除 */
   while (!(LCDCRA & (1<<LCDIF)));
                                                /* 割り込み(フレーム開始)まで待機 */
   LCDCRA = (0 << LCDEN);
                                                /* LCD禁止 */
```

# LCD制御器/駆動部用レジスタ

# LCD制御/状態レジスタA (LCD Control and Status Register A) LCDCRA

ピット	7	6	5	4	3	2	1	0	
(\$E4)	LCDEN	LCDAB	-	LCDIF	LCDIE	-	_	LCDBL	LCDCRA
Read/Write	R/W	R/W	R	R/W	R/W	R	R	R/W	
初期値	0	0	0	0	0	0	0	0	

#### • ビット7 - LCDEN: LCD許可(LCD Enable)

このビットに1を書くことがLCD制御器/駆動部を許可します。0書き込みによってLCDは直ちに0FFへ切り替えられます。表示器を駆動している間にLCD制御器/駆動部をOFFに変えることは標準ポート機能を許可し、ポートが出力として設定されている場合、DC電圧が表示器に印加され得ます。LCD制御器/駆動部が禁止される場合、表示器を放電するために出力をGND(Low)に駆動することが推奨されます。

#### • ビット6 - LCDAB : LCD低電力波形選択 (LCD Low Power Waveform)

LCDABが論理0を書かれると、既定(標準)波形がLCDピンでの出力です。LCDABが論理1を書かれると、低電力波形がLCDピンでの出力です。このビットが表示動作中に変更される場合、その変更は新規フレームの始めで行います。

#### • ビット5 - Res: 予約 (Reserved)

このビットは予約されており、常に0として読みます。

#### • ビット4 - LCDIF: LCD割り込み要求フラグ(LCD Interruput Flag)

このビットは新規フレームの始めで表示データが更新されるのと同時にハートヴェアによって設定(1)されます。ステータスレジスタ(SREG)の全割り込み許可(I)ピットとLCD割り込み許可(LCDIE)ピットが設定(1)なら、LCDフレーム開始割り込みが実行されます。対応する割り込み処理へクタを実行する時にLCDIFはハートヴェアによって解除(0)されます。代わりにこのフラグへの論理1書き込みがLCDIFを解除(0)します。LCDCRAで読み一修正一書き(リート・モデファイライト)を行うと、保留中の割り込みが禁止され得ることに注意してください。低電力波形が選ばれると、割り込み要求フラグは全ての第2フレームで設定(1)されます。

## • ビット3 - LCDIE: LCD割り込み許可(LCD Interruput Enable)

このビットが1を書かれ、SREGの全割り込み許可(I)ビットが設定(1)なら、LCDフレーム開始割り込みが許可されます。

#### • ビット2.1 - Res: 予約 (Reserved)

これらのビットは予約されており、常に0として読みます。

#### • ビット0 - LCDBL: LCD無表示 (LCD Blanking)

このビットが1を書かれると、(現在の)フレーム完了後に表示器が無表示にされます。全てのCOMピンとSEGピンはGNDに駆動されます。

#### LCD制御/状態レジスタB (LCD Control and Status Register B) LCDCRB

ピット	7	6	5	4	3	2	1	0	
(\$E5)	LCDCS	LCD2B	LCDMUX1	LCDMUX0	-	LCDPM2	LCDPM1	LCDPM0	LCDCRB
Read/Write	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

#### • ビット7 - LCDCS: LCDクロック選択 (LCD Clock Select)

このビットが0を書かれると、システム クロックが使われます。このビットが1を書かれると、外部非同期クロック元が使われます。非同期クロック元はASSRのEXCLKに依存し、タイマ/カウンタ用発振器か外部クロックのどちらかです。より多くの詳細については97頁の「タイマ/カウンタ2の非同期動作」をご覧ください。

#### • ビット6 - LCD2B: LCDバイアス選択 (LCD 1/2 Bias Select)

このビットが<mark>0</mark>を書かれると、1/3バイアスが使われます。このビットが1を書かれると、1/2バイアスが使われます。推奨バイアス選択についてはLCD製品を参照してください。

#### • ビット5,4 - LCDMUX1,0 : LCDデューティ選択 (LCD Mux Select)

このLCDMUX1,0ビットはデューティサイクルを決めます。使われないCOMピンは標準ポート ピンです。各種デューティ選択は**表94**.で示されます。

表94.	LOD	="	_= /	11. 铅气
<b>衣文 54.</b>	LUD	) <u>_</u>	- / 1	迭扒

LCDMUX1,0	デューティ	バイアス(注)	COMピン	標準I/Oピン
0 0	スタティック	スタティック	COM0	COM1~3
0 1	1/2	1/2または1/3	COM0,1	COM2,3
1 0	1/3	1/2または1/3	COM0~2	COM3
11	1/4	1/2または1/3	COM0~3	なし

注: LCD2B=0で1/3バイアス、LCD2B=1で1/2バイアスです。





• ビット3 - Res: 予約 (Reserved)

このビットは予約されており、常に0として読みます。

• ビット2~0 - LCDPM2~0 : LCD ポート遮蔽 (LCD Port Mask)

このLCDPM2~0ビットはセグメント駆動部として使われるべきポート ピン数を決めます。各種の選択は**表95**.で示されます。未使用ピンは標準ポート ピンとして使えます。

### 表95. LCD ポート遮蔽選択

LCDPM2~0	使用SEGピン	最大セグメント数	LCDPM2~0	使用SEGピン	最大セグメント数
0 0 0	SEG0∼12	13	100	SEG0~20	21
0 0 1	SEG0~14	15	101	SEG0~22	23
0 1 0	SEG0∼16	17	1 1 0	SEG0~23	24
0 1 1	SEG0~18	19	111	SEG0~24	25

## LCDフレーム速度レジスタ (LCD Frame Rate Register) LCDFRR

ピット	7	6	5	4	3	2	1	0	
(\$E6)	_	LCDPS2	LCDPS1	LCDPS0	_	LCDCD2	LCDCD1	LCDCD0	LCDFRR
Read/Write	R	R/W	R/W	R/W	R	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

• ビット7 - Res: 予約 (Reserved)

このビットは予約されており、常に0として読みます。

• ビット6~4 - LCDPS2~0: LCD 前置分周選択 (LCD Prescaler Select)

このLCDPS2~0ビットは前置分周器からの引き出し点を選びます。この前置分周器出力はクロック分周(LCDCD2~0)ビット設定によって更に分周できます。各種選択は表96.で示されます。それら(LCDCD)と共に決める前置分周したLCDクロック(clk\_LCD\_PS)がLCD部をクロック駆動します。

# 表96. LCD 前置分周選択

LCDPS2~0	前置分周出力	参考周波数	LCDPS2~0	前置分周出力	参考周波数
0 0 0	clkLCD/16	8.1kHz	100	clk <sub>LCD</sub> /512	260kHz
0 0 1	clk <sub>LCD</sub> /64	33kHz	101	clk <sub>LCD</sub> /1024	520kHz
0 1 0	clk <sub>LCD</sub> /128	66kHz	1 1 0	clk <sub>LCD</sub> /2048	1MHz
0 1 1	clk <sub>LCD</sub> /256	130kHz	111	clk <sub>LCD</sub> /4096	2MHz

**注**: 参考周波数はLCDCD2~0=000, デューティ=1/4, フレーム速度=64Hz時の前置分周した LCDクロック周波数です。

• ビット3 - Res: 予約 (Reserved)

このビットは予約されており、常に0として読みます。

• ビット2~0 - LCDCD2~0 : LCD クロック分周選択 (LCD Clock Divide 2~0)

このLCDCD2~0ビットはクロック分周器の分周比を決めます。様々な選択は**表97**.で示されます。このクロック分周器はフレーム速度選択に付加的な柔軟性を与えます。

# 表97. LCDクロック分周選択

LCDCD2~0	分周値(D)	参考周波数	LCDCD2~0	分周値 (D)	参考周波数
0 0 0	1	256Hz	100	5	51.2Hz
0 0 1	2	128Hz	101	6	42.7Hz
010	3	85.3Hz	110	7	36.6Hz
0 1 1	4	64Hz	111	8	32Hz

**注**: 参考周波数はclk\_CD=32.768kHz, 前置分周=1/16, デューティ=1/4のフレーム速度周波数です。

フレーム周波数は次式によって計算できます。

 $f_{\text{frame}} = \frac{f_{\text{clkLCD}}}{K_{\text{NANCE}}}$ 

N:前置分周値(16, 64, 128, 256, 512, 1024, 2048, 4096)

K:1/3デューティ時は6、その他は8

×N×D D:分周値(**表97**.参照)

これは非常に柔軟な仕組みで、使用者は上の公式から可能なフレーム速度を調査するのに自身の表を計算することが奨励されます。 1/3デューティを使う時にフレーム速度レジスタ(LCDFRR)が一定なら、フレーム速度が33%増やされることに注意してください。フレーム速度計算の例は表98.で示されます。

#### 表98. フレーム 速度計算例

clkLCD	デューティ	K	LCDPS2~0	N	LCDCD2~0	D	フレーム速度
4MH In	1/4	8	1.1.0	2048	0.1.1	1	$4000000 \div (8 \times 2048 \times (1+3)) = 61$ Hz
4MHz	1/3	6	1 1 0	2048	011	4	$4000000 \div (6 \times 2048 \times (1+3)) = 81$ Hz
22 7601-11-	スタティック	8	0 0 0	16	0 0 0	1	$32768 \div (8 \times 16 \times (1+0)) = 256$ Hz
32.768kHz	1/4	8	000	10	100	5	$32768 \div (8 \times 16 \times (1+4)) = 51$ Hz

# LCD濃淡制御レジスタ (LCD Contrast Control Register) LCDCCR

ピット	7	6	5	4	3	2	1	0	
(\$E7)	LCDDC2	LCDDC1	LCDDC0	-	LCDCC3	LCDCC2	LCDCC1	LCDCC0	LCDCCR
Read/Write	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

### • ビット7~5 - LCDDC2~0: LCD表示器設定 (LCD Display Configuration)

LCDDC2~0ビットは共通電極とセグメント ピンの各電圧遷移に関してLCD駆動部がONに切り替えられる時間を決めます。短い駆動時間は低電力消費を引き出しますが、高い内部抵抗や容量の表示器は満足する濃淡に達するのにより長い駆動時間を必要とするかもしれません。例え選んだ駆動時間が長くても、駆動時間は前置分周したLCDクロック周期の1.5倍よりも決して長くならないことに注意してください。無表示(プランキンク゚)またはスタティック パイアス使用時、駆動時間は常に前置分周したLCDクロック周期の1.5倍です。

新規値は直ちに実行され、表示器出力で小さな不具合を起こし得ます。これはLCD制御/状態レジスタ(LCDCRA)のLCD無表示(LCDBL)を設定(1)し、LCDDC2~0変更に先立って次のフレーム開始を待つことよって避けられます。 (訳注:共通性から2行追加)

注: これらのビットはATmega169改訂A~Dでは利用できません。

表99. LCD表7	<b>下</b>
LCDDC2~0	公称駆動時間
0 0 0	300µs
0 0 1	70µs
0 1 0	150µs
0 1 1	450μs
1 0 0	575µs
1 0 1	850µs
1 1 0	1150μs
111	clkLCD_PSの半分

注:この駆動時間は発振器起動時間に対して、より長いでしょう。

# • ビット4 - Res: 予約 (Reserved)

このビットは予約されており、常に0として読みます。

### • ビット3~0 - LCDCC3~0: LCD 濃淡制御 (LCD Contrast Control)

このLCDCC3~0ビットはコモンとセグメント ピン上の最大電圧(VLCD)を決めます。各種選択は**表100**.で示されます。新しい値は全ての新規フレームの始めで実施されます。

表10	nn I	CD	漕等	な 集川	細理	引
42 I U	7U. L	.UU	까 가	ויוח א	ᄬᆟᄣᆘᅔ	マカノ

LCDCC3~0	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
最大電圧(VLCD)	2.60	2.65	2.70	2.75	2.80	2.85	2.90	2.95	3.00	3.05	3.10	3.15	3.20	3.25	3.30	3.35

# LCDメモリの配置

LCDメモリのビットに1を書くと、対応するセグメントが活動(可視)にされます。実際の表示器に対して未使用のLCDメモリビットは保存場所として自由に使えます。

#### LCDデータレジスタ (LCD Data Register) LCDDR0~19

ピット	7	6	5	4	3	2	1	0	
(\$FF)	-	-	-	-	-	-	-	-	LCDDR19 (予約)
(\$FE)	-	_	-	-	-	-	-	SEG324	LCDDR18
(\$FD)	SEG323	SEG322	SEG321	SEG320	SEG319	SEG318	SEG317	SEG316	LCDDR17
(\$FC)	SEG315	SEG314	SEG313	SEG312	SEG311	SEG310	SEG309	SEG308	LCDDR16
(\$FB)	SEG307	SEG306	SEG305	SEG304	SEG303	SEG302	SEG301	SEG300	LCDDR15
(\$FA)	-	1	1	-	1	-	-	-	LCDDR14 (予約)
(\$F9)	-	1	1	_	1	-	-	SEG224	LCDDR13
(\$F8)	SEG223	SEG222	SEG221	SEG220	SEG219	SEG218	SEG217	SEG216	LCDDR12 COM2
(\$F7)	SEG215	SEG214	SEG213	SEG212	SEG211	SEG210	SEG209	SEG208	LCDDR11
(\$F6)	SEG207	SEG206	SEG205	SEG204	SEG203	SEG202	SEG201	SEG200	LCDDR10
(\$F5)	-	1	-	-	-	-	-	-	LCDDR9 (予約)
(\$F4)	_	_	-	-	-	-	-	SEG124	LCDDR8
(\$F3)	SEG123	SEG122	SEG121	SEG120	SEG119	SEG118	SEG117	SEG116	LCDDR7
(\$F2)	SEG115	SEG114	SEG113	SEG112	SEG111	SEG110	SEG109	SEG108	LCDDR6
(\$F1)	SEG107	SEG106	SEG105	SEG104	SEG103	SEG102	SEG101	SEG100	LCDDR5
(\$F0)	_	_	_	_	_	_	-	_	LCDDR4 (予約)
(\$EF)	-	-	-	-	-	-	-	SEG024	LCDDR3
(\$EE)	SEG023	SEG022	SEG021	SEG020	SEG019	SEG018	SEG017	SEG016	LCDDR2
(\$ED)	SEG015	SEG014	SEG013	SEG012	SEG011	SEG010	SEG009	SEG008	LCDDR1
(\$EC)	SEG007	SEG006	SEG005	SEG004	SEG003	SEG002	SEG001	SEG000	LCDDR0
Read/Write	R/W								
初期値	0	0	0	0	0	0	0	0	





# JTAGインターフェースと内蔵デバッグ機能

## 特徴

- JTAGインターフェース (IEEE標準1149.1準拠)
- JTAG規格に従った境界走査(Boundary-Scan)能力
- デバッガは各部のアクセスが可能
  - 全ての内蔵周辺機能
  - 内部及び外部RAM
  - 内蔵レジスタファイル
  - プログラム カウンタ
  - EEPROM及びフラッシュ メモリ
- 中断(BREAK)によって支援される広範囲な内蔵デバッグ機能
  - AVRのBREAK命令
  - プログラムの流れ変更での停止
  - 1命令実行(シングル ステップ)停止
  - プログラム メモリ上の単一アト・レスまたはアト・レス範囲による中断点(ブレークホペイント)
  - データ メモリ上の単一アドレスまたはアドレス範囲による中断点
- JTAGインターフェースを介してのフラッシュ メモリ、EEPROM、ヒュース゛ビット、施錠ビットの読み書き(プログラミング)
- AVR Studioによる内蔵デバッグ機能の支援

#### 概要

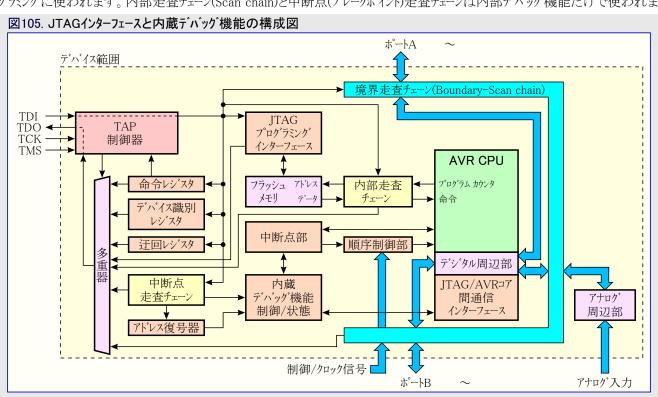
AVRのIEEE標準1149.1準拠JTAGインターフェースは以下について使えます。

- JTAG境界走査(Boudary-Scan)能力による基板などの試験
- 不揮発性メモリ、ヒュース、ヒ・ット、施錠ビットの読み書き(プログラミング)
- 内蔵デバッグ機能による実装デバッグ

簡単な記述は次節で与えられます。JTAGインターフェース経由のプログラミングと境界走査(Boundary-Scan)チェーンに関する詳細な記述は、各々187頁の「JTAGインターフェース経由のプログラミング」と156頁の「IEEE 1149.1(JTAG)境界走査(Boundary-Scan)」で得られます。内蔵デバッグ機能の支援は独自JTAG命令で考慮されており、Atmelと選ばれた業者のみに配布されます。

図105.はJTAGインターフェースと内蔵デバッグ機能の構成図を示します。TAP(検査入出力ポート)制御器はTMSとTCKによって制御される順次回路です。TAP制御器はTDI入力とTDO出力間の走査チェーン(移動レジスタ接続)として、JTAG命令レジスタまたは各種データ レジスタの1つのどれかを選びます。JTAG命令レジスタはデータ レジスタの動きを制御するJTAG命令を保持します。

デバイス識別(Identification)レシブスタ、迂回(Bypass)レシブスタ、境界走査チェーン(Boundary-Scan chain)の各データレシブスタは基板段階の試験に使われます。(実際には現実と仮想の各種データレジスタで構成する)JTAGプログラミングインターフェースはJTAGインターフェース経由での直列プログラミングに使われます。内部走査チェーン(Scan chain)と中断点(ブレークポイント)走査チェーンは内部デバッグ機能だけで使われます。



## 検査入出力ポート (TAP:Test Access Port)

JTAGインターフェースはAVRの4つのピンを通してアクセスされます。JTAG用語ではこれらのピンが検査入出力ポート(TAP)を構成します。

- TMS: 検査種別選択。このピンはTAP制御器順次回路を通しての指示に使われます。
- TCK:検査クロック。JTAG操作はTCKに同期します。
- TDI :検査データ入力。命令レジスタまたはデータレジスタ(走査チェーン)内に移動されるべき直列入力データです。
- TDO: 検査データ出力。命令レシ、スタまたはデータレシ、スタからの直列出力データです。

IEEE標準1149.1では任意TAP信号の検査リセット(TRST)も記載されていますが、この信号は提供されません。

JTAGENヒュース、が非プログラム(1)の時にこれら4つのTAPピンは標準ポート ピンで、TAP制御器はリセットです。プログラム(0)され、MCU制御レジスタ(MCUCR)のJTAGインターフェース禁止(JTD)ビットが解除(0)されると、TAP信号入力は内部的にHighへ引かれ、JTAGは境界走査(Boundary-Scan)とプログラミングに関して許可されます。このデバイスはこのヒューズがプログラム(0)されて出荷されます。

内蔵デバッグ機能ではJTAGインターフェース ピンに加え、外部リセット元が検知できるため、RESETピンがデバッカによって監視されます。応用でリセット信号線にオープン コレクタ(ドレイン)だけが使われるとすれば、デバッカは全システムをリセットするためにRESETピンをLowにすることもできます。

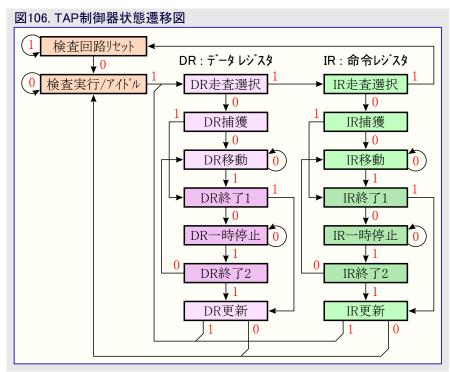
# TAP制御器

TAP制御器は境界走査(Boundary-Scan)回路、JTAGプログラミング回路、内蔵デバッグ機能の働きを制御する16段の無限順次回路です。図106.に描かれた状態遷移はTCKクロックの上昇時の(各状態遷移付近で示される)TMS上に存在する信号に依存します。電源投入リセット後の初期状態は検査回路リセットです。

この文書内の定義として、全ての移動レジスタに関して入出力ともLSBが最初に移動されます。

検査実行/アイドル状態が現在の状態と仮定した、 JTAGインターフェースを使うための典型的な手順を次 に示します。

• 命令レジスタ(IR)移動状態へ移行するためにTC Kの上昇でTMS〜順次1,1,0,0を与えます。この 状態中、TCKの上昇端でTDI入力からJTAG命令レジスタ内に4比ットのJTAG命令を移動します。 TMS入力はIR移動状態に留まるために3 LSB の入力中、Lowに保持されなければなりません。命令のMSBはTMS入力のHigh設定によってこの状態を抜ける時に移動入力されます。命令がTDIピンから移動されている間、捕獲されたIRの状態(\$01)がTDOピンへ移動出力されます。



す。JTAG命令はTDIとTDO間に通す(接続する)特定のデータレジスタを選び、選んだデータレジスタ周辺回路を制御します。

- 検査実行/アイドル状態へ再度移行するためにTMSへ順次1,1,0を与えます。この(JTAG)命令はIR更新状態通過で移動レジスタから並列出力にラッチされます。IR終了1、IR一時停止、IR終了2の各状態は順次回路の誘導操作のためだけに使われます。
- データレジスタ(DR)移動状態へ移行するためにTCKの上昇端でTMSへ順次1,0,0を与えます。この状態中、TCKの上昇端でTDI入力から(JTAG命令レジスタ内の現在のJTAG命令で)選んだデータレジスタに値を移動入力します。DR移動状態に留まるためにMSBを除く全ビットの入力中、TMS入力はLowに保持されなければなりません。データのMSBはTMS入力のHigh設定によってこの状態を抜ける時に移動入力されます。データレジスタがTDIピンから移動入力される間、DR捕獲状態(1,0)でデータレジスタに捕獲した並列入力がTDOピンへ移動出力されます。
- 検査実行/アイドル状態へ再移行するためにTMSへ順次1,1,0を与えます。選んだデータ レシ スタがラッチした並列出力を持つ場合、DR 更新状態でラッチを行います。DR終了1、DR一時停止、DR終了2の各状態は順次回路の誘導操作のためだけに使われます。

状態遷移図で示されるように検査実行/アイドル状態はJTAG命令の選択とデータレジスタの使用間で移行される必要はなく、また、いくつかのJTAG命令は検査実行/アイドル状態で実行されるべき必然的な機能を選ぶかもしれず、それはアイドル状態として不適当になります。

注: TAP制御器の初期状態に関係なく、5 TCKクロック周期に対してTMSをHighに保持することにより、常に検査回路リセット状態へ移行することができます。

JTAG仕様の詳細情報については155頁の「参考文献」に記載された文献を参照してください。





# 境界走査チェーン(Boundary-Scan Chain)の使用

境界走査能力の完全な説明は156頁の「IEEE 1149.1(JTAG)境界走査(Boundary-Scan)」項で与えられます。

# 内蔵デバッグ機能の使用

図105.で示されるように内蔵デバッグに関するハードウェア支援は主に次の部分から成ります。

- 内部AVR CPUコアと内部周辺機能部間インターフェースの走査チェーン(Scan chain)
- 中断点(ブレークポイント)部
- CPUとJTAGシステム間の通信インターフェース

デバッガの実行で必要とされる全ての読み込みまたは変更/書き込みは内部AVR CPU走査チェーン(Scan chain)経由のAVR命令によって行われます。CPUはCPUとJTAGシステム間の通信インターフェース部分が配置されるI/Oメモリ位置に結果を送ります。

中断点部はプログラムの流れ変更での一時停止(Break)、1命令実行(Single Step Break)、2つのプログラム メモリ用中断点、2つの複合中断点を実行します。合わせて4つ中断点は次のどれかとして設定ができます。

- 4つのプログラム メモリ単一中断点
- 3つのプログラム メモリ単一中断点 + 1つのデータ メモリ単一中断点
- 2つのプログラム メモリ単一中断点 + 2つのデータ メモリ単一中断点
- 2つのプログラム メモリ単一中断点 + 1つのプログラム メモリ範囲中断点 (中断点と遮蔽)
- 2つのプログラム メモリ単一中断点 + 1つのデータ メモリ範囲中断点 (中断点と遮蔽)

けれどもAVR Studio®のようなデバッガは最終使用者への柔軟性をより少なくする内部目的にこれら資源の1つまたはそれ以上を使うかもしれません。

内蔵デバック・特殊JTAG命令の一覧は「内蔵デバック・特殊JTAG命令」で与えられます。

JTAG検査入出力ポート(TAP)を許可するにはJTAGENヒューズがプログラム(0)されなければなりません。加えて内蔵デ´バッグ機能で作業するには更にOCDENヒューズがプログラム(0)され、施錠ビットが施錠なしに設定されなければなりません。何れかの施錠ビットが設定(0)されると、保護機能で内蔵デ´バッグ機能が禁止されます。そうでなければ内蔵デ´バッグ機能が保護されたデ´ハ´イスへの裏口を提供することになっていたでしょう(訳補:保護の意味がなくなるの意)。

AVR Studioは使用者にチップ・内蔵デバッグ・能力(機能)、AVRインサーキットエミュレータ、または(AVR Studio)組み込みAVR命令一式シミュレータでAVRデバイスのプログラム実行を完全に制御することを可能にします。AVR StudioはAtmel AVRアセンブラでアセンブルしたアセンブリ言語プログラムと他業者製コンパイラでコンパイルしたC言語プログラムのソースレベル実行を支援します。

AVR StudioはMicrosoft Windows® 98/2000、Windows NT®、Windows XP®、Windows Vista®下で走行します。

AVR Studioの完全な記述についてはAVR Studio使用者の手引きを参照してください。要点だけが本文書で示されます。

全ての必要な実行指令はAVR Studio内のソースレベルと逆アセンブリレベルの両方で利用できます。使用者はプログラムの実行、1命令実行(Single Step)、内側(関数内)実行、外側(関数外)実行、命令行へのカーソル配置と命令行が(カーソル行へ)到達されるまでの実行、一時停止、実行対象のリセットができます。加えて使用者は無制限数のコートの中断点(BREAK命令使用)と、2つまでのデータ メモリの中断点か、代わりに遮蔽による範囲指定(複合)中断点を持てます。

### 内蔵デバッグ特殊JTAG命令

内蔵デバッグの支援は独自JTAG命令で考慮され、Atmelと選ばれた業者だけに配布されます。参考のため、命令コードの一覧が示されます。

• **独自命令0**: \$8 (内蔵デバッグ機能アクセス用独自JTAG命令)

• **独自命令2**: \$A (内蔵デバッグ機能アクセス用独自JTAG命令)

• 独自命令1: \$9 (内蔵デバッグ機能アクセス用独自JTAG命令)

•独自命令3: \$B(内蔵デバッグ機能アクセス用独自JTAG命令)

### 内蔵デバッグに関連するI/Oメモリ内のレジスタ

内蔵デバック レシスタ (On-chip Debug Register) OCDR

ピット	7	6	5	4	3	2	1	0	
\$31 (\$51)	IDRD/MSB							LSB	OCDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

OCDRはマイクロコントローラ内の実行プログラムからディッカーへの通信チャネルを提供します。CPUはこの位置へ書くことによってディッカーへがイトを転送できます。このレジスタが書かれてしまっているのをディッカに示すため、同時に内部フラグIDRD(I/Oディック・レジスタ更新)が設定(1)されます。CPUがOCDRを読むとき、LSB 7ビットがOCDRからで、一方MSBはIDRDビットです。ディッカはこの情報を読む時にIDRDビットを解除(0)します。

いくつかのAVRデバイスに於いてこのレシ、スタは標準I/O位置で共用されます。この場合、OCDRは内蔵デバッグ許可(OCDEN)ヒュースがプログラム(0)された場合にだけアクセスでき、デバッガがOCDRへアクセスできます。他の全ての場合、標準I/O位置がアクセスされます。

このレジスタの使用法のより多くの情報はデバッカの資料を参照してください。

# JTAGプログラミング能力の使用

JTAG経由AVRデバイスプログラミングは4ピンのJTAGポート、TCK,TMS,TDI,TDO経由で実行されます。これらはJTAGプログラミング実行のために(電源ピンに加えて)制御/監視される必要があるピンだけです。外部的な12V印加は必要とされません。JTAG検査入出力ポート (TAP)を許可するにはJTAGENヒューズがプログラム(0)され、MCU制御レジスタ(MCUCR)のJTAGインターフェース禁止(JTD)ビットが解除(0)されなければなりません。

JTAGプログラミング能力は以下を支援します。

- フラッシュ メモリのプログラミングと照合
- EEPROMのプログラミングと照合
- ヒュース゛ ビットのプログラミングと照合
- 施錠ビットのプログラミングと照合

施錠ビット保護は丁度並列プログラミング動作でのようです。施錠ビットLB1またはLB2がプログラム(0)されると、先にチップ消去を行わない限り、OCDENヒューズはプログラム(0)できません。これは保護されたデバイスの内容を読み出す裏口が存在しないことを保証する保護機能です。

JTAGインターフェースを通すプログラミングとプログラミング、特殊JTAG命令の記述は187頁の「JTAGインターフェース経由のプログラミング」」項で与えられます。

# 参考文献

- 一般的な境界走査(Boudary-Scan)についての詳細は次の文献を参照してください。
- IEEE: IEEE Std 1149.1-1990. IEEE Standard Test Access Port and Boundary-Scan Architecture, IEEE, 1993
- Colin Maunder: The Board Designers Guide to Testable Logic Circuits, Addison Wesley, 1992





# IEEE 1149.1 (JTAG) 境界走査(Boundary-Scan)

## 特徴

- JTAGインターフェース (IEEE標準1149.1準拠)
- JTAG規格に従った境界走査(Boundary-Scan)能力
- チップ外接続を持つアナログ回路は勿論、全てのポート機能の完全な走査(Scan)
- 任意IDCODE命令の支援
- AVRリセット用公開AVR\_RESET命令の追加

## 概要

境界走査チェーン(Boundary-Scan chain)にはデジタルI/Oピンは勿論、チップ外接続を持つアナログ回路のアナログとデジタル間境界の論理値を駆動、監視する能力があります。システム上ではJTAG能力を持つ全てのICがTDIとTDOの接続により長い移動レジスタの形態で接続されます。外部制御器がこれらの出力ピンで値を駆動するようにデバイスを設定し、他のデバイスから受け取った入力値を監視します。この制御器は予測される結果と受信した値を比較します。この方法で境界走査(Boundary-Scan)は4つのTAP信号だけを使うことにより、基板上の部品全般と内部結線の検査機構を提供します。

IEEE 1149.1で定義された4つの必須JTAG命令、IDCODE、BYPASS、SAMPLE/PRELOAD、EXTESTだけでなく、AVR特殊公開 JTAG命令のAVR\_RESETも基板検査に使えます。既定JTAG命令がIDCODEなので、データレジスタ経路の初期走査はデバイスの識別 符号(ID-code)を示します。検査動作中にリセット(状態)のAVRデバイスがあることは好都合かもしれません。リセットでない場合、デバイスの入力は走査動作によって決定されるかもしれず、検査動作を終了する時に内部ソフトウェアが未確定状態になってしまうかもしれません。リセットへの移行で、どのポート ピンの出力も直ちにHi-Z状態に移行し、HIGHZ命令が不要になります。必要とされるなら、デバイスを通過する走査チェーンを可能な限り最短とするのにBYPASS命令が発行できます。外部RESETピンをLowに引き込むか、またはリセットデータレジスタの適切な設定によるAVR RESET命令の発行でデバイスをリセット状態に設定できます。

EXTEST命令は外部ピンの採取と出力ピンの値設定に使われます。EXTEST命令がJTAG命令レジスタ(IR)に設定されると、直ちに出力 ラッチからの値がピンで出力駆動されます。このため、最初にEXTEST命令を発行する時に基板の損傷を避けるため、走査チェーン内に 初期値を設定するのにSAMPLE/PRELOAD命令も使われるべきです。SAMPLE/PRELOAD命令は通常動作中デバイスの外部ピンの 高速監視にも使えます。

JTAG検査入出力ポート(TAP)を許可するにはJTAGENヒュース、がプログラム(0)され、MCU制御レシ、スタ(MCUCR)のJTAGインターフェース禁止(JTD)ヒットが解除(0)されなければなりません。

境界走査(Boundary-Scan)にJTAGインターフェースを使う場合、内部チップ。周波数よりも高いJTAG TCKクロック周波数の使用が可能です。チップ。のクロックは走行に必要とされません。

# データ レジスタ

境界走査(Boundary-Scan)に関連するデータレジスタ(DR)を次に示します。

• 迂回(Bypass)レシ、スタ • デバイス識別(Device Identification)レシ、スタ • リセット(Reset)レシ、スタ • 境界走査チェーン(Boundary-Scan chain)

### 迂回 (Bypass) レジスタ

迂回レジスタは1段の移動レジスタから成ります。TDIとTDO間の経路として迂回レジスタが選ばれると、このレジスタはTAP制御器がデータレジスタ(DR)捕獲状態を抜ける時に0ヘリセットされます。迂回レジスタは他のデバイスが検査される時にシステム上の走査チェーン(Scan chain)を短くするのに使えます。

#### テブバイス識別 (Device Identification) レジスタ

図107.はデバイス識別レジスタの構造を示します。

図107. デバイス	図107. デバイス識別レジスタの形式								
ピット	31 28	27	12	11 1	0				
デバイス識別	版番号	部品番号		製造者識別	1				
ピット数	4	16		11	1				

**版番号** 版番号は部品(デバイス)の改訂を示す4ビットです。JTAG版番号はデバイスの改訂版に従います。改訂Aは\$00, 改訂Bは\$01, 以下同様です。

**部品番号** 部品番号は部品(名)を示す16ビットです。ATmega169 のITAG部品番号は表**101**.で一覧されます。

表101. AVR JTAG 部品番号						
部品番号	JTAG部品番号(Hex)					
ATmega169	\$9405					

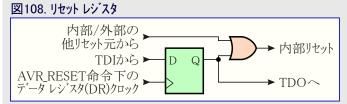
製造者識別 製造者識別は製造業者を示す11ビットです。Atmelの JTAG製造者識別は表102.で示されます。

表102. 製造者ID	
製造業者	JTAG製造者ID番号(Hex)
Atmel	\$01F

### リセット (Reset) レジスタ

リセットレジスタはデバイスのリセットに使われる検査データレジスタです。リセットによってAVRのポートピンがHi-Z状態となるので、未実装の任意 JTAG命令HIGHZ機能の代わりにできます。

リセットレジスタ内の値1は外部RESETt°ンをLowに引っ張るのと同じです。デバイスはリセットレジスタ内の値が1である限り、リセットです。リセットレジスタの開放後、デバイスはクロック選択用ヒュース、ビットの設定に従ってリセット起動遅延時間(17頁の「クロック元」参照)、リセットに留まります。図108.で示されるようにこのデータレジスタからの出力はラッチされず、直ちにリセットが行われます。



# 境界走査チェーン(Boundary-Scan Chain)

境界走査チェーンにはデジタルI/Oピンは勿論、チップ外の接続を持つアナログ回路のアナログとデジタル間境界の論理値上の設定と監視能力があります。

完全な記述については159頁の「境界走査チェーン(Boundary-Scan chain)」をご覧ください。

# 境界走査(Boundary-Scan)用JTAG命令

命令レジスタ(IR)は16種までの命令を支援する4ビット幅です。以下に示した一覧は境界走査(Boundary-Scan)操作に有用なJTAG命令です。全ポート ピンをHi-Z状態にするための任意HIGHZ命令は実装されませんが、AVR\_RESET命令を使うことによって全出力がHi-Z 状態に設定できることに注目してください。

このデータシート内の定義として、全ての移動レジスタに関して入出力ともLSBが最初に移動されます。

各命令の命令コートは命令名下の16進形式で示されます。本文は各命令についてTDIとTDO間の経路として選ばれるデータレジスタを記述します。(訳注:配置構成変更により、内容を一部変更)

# EXTEST - \$0

AVRデバイスに外部回路的検査を行うためにデータレジスタ(DR)として境界走査チェーン(Boundary-Scan chain)を選ぶための必須JTAG命令です。ポートピンは方向、プルアップ、出力値、入力値は走査チェーン(Scan chain)内で全てアクセス可能です。チップ外接続を持つアナログ回路についてはアナログとデジタル論理間のインターフェースが走査チェーン内になります。境界走査チェーンのラッチした出力値はJTAG命令レジスタ(IR)にEXTEST命令が設定されると直ぐに駆動されます。本命令が有効な状態を次に示します。

- DR捕獲:外部ピン上の値が境界走査チェーン内に採取されます。
- DR移動:内部走査チェーンがTCKクロック入力によって移されます。
- DR更新: 走査チェーンからの値が出力ピンに印加(出力)されます。

#### IDCODE - \$1

データレジスタ(DR)として32ビットのデバイス識別(ID)レジスタを選ぶ任意JTAG命令です。デバイス識別レジスタは版番号、デバイス番号と、JED ECによって決めれた製造者符号から成ります。これは電源投入後の既定命令です。本命令が有効な状態を次に示します。

- DR捕獲:デバイス識別レジスタ内の値が境界走査チェーン内に採取されます。
- DR移動:内部走査チェーン(デバイス識別レジスタ値)がTCKクロック入力によって移されます。

## SAMPLE\_PRELOAD - \$2

システム動作に影響を与えずに入出力ピンの状態採取、出力ラッチの事前設定を行うための必須JTAG命令です。けれども出力ラッチはピンに接続されません。データレジスタ(DR)として境界走査チェーン(Boundary-Scan chain)が選ばれます。本命令が有効な状態を次に示します。

- DR捕獲:外部ピン上の値が境界走査チェーン内に採取されます。
- DR移動:境界走査チェーンがTCKクロック入力によって移されます。
- DR更新:境界走査チェーンからの値が出力ラッチに印加(設定)されます。けれども出力ラッチはピンに接続されません。

#### AVR\_RESET - \$C

AVRデバイスを強制的にリセット状態、またはJTAGリセット元から開放するためのAVR特殊公開JTAG命令です。この命令で検査入出力ポート(TAP)制御器はリセットされません。データレシ、スタ(DR)として1ビットのリセットレシ、スタが選ばれます。 リセット チェーン(レシ、スタ)が論理1である限り、リセットが活性(有効)なことに注意してください。このチェーンからの出力はラッチされません。 本命令が有効な状態を次に示します。

• **DR移動**: 走査チェーン(リセット レシ スタ)がTCKクロック入力によって移されます。

### BYPASS - \$F

データレジスタ(DR)に迂回(Bypass)レジスタを選ぶ必須JTAG命令です。本命令が有効な状態を次に示します。

- DR捕獲:迂回レジスタに論理0を設定します。
- DR移動: TDIとTDO間の迂回レジスタ(セル)が移されます。





# 境界走査(Boundary-Scan)に関連するI/Oメモリ内のレジスタ

### MCU制御レシ、スタ (MCU Control Register) MCUCR

MCU制御レシブスタは一般MCU機能制御ビットを含みます。

ピット	7	6	5	4	3	2	1	0	_
\$35 (\$55)	JTD	-	-	PUD	-	-	IVSEL	IVCE	MCUCR
Read/Write	R/W	R	R	R/W	R	R	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

## • ビット7 - JTD: JTAGインターフェース禁止 (JTAG interface disable)

このビットが0の時にJTAGENヒューズがプログラム(0)されていれば、JTAGインターフェースが許可されます。このビットが1の場合、JTAGインターフェースは禁止されます。JTAGインターフェースの予期せぬ許可や禁止を避けるためにこのビットを変更する時は時間制限手順に従わなければなりません。この値を変更するために応用ソフトウェアは4周期以内に希望する値をこのビットへ2回書かなければなりません。内蔵デ バッグ システム使用時、このビットが切り替えられてはならないことに注意してください。

JTAGインタフェースが他のJTAG回路に接続されないままにされる場合、このJTDピットは1に設定されるべきです。この理由はJTAGインターフェースのTDOピンの静止時電流を避けるためです。

# MCU状態レジスタ (MCU Status Register) MCUSR

MCU状態レシ、スタはMCUリセットを起こしたリセット元の情報を提供します。

ピット \$34 (\$54)	7	6	5 -	4 JTRF	3 WDRF	2 BORF	1 EXTRF	0 PORF	MCUSR
Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W	
初期値	0	0	0	内容参照	内容参照	内容参照	内容参照	内容参照	

#### • ビット4 - JTRF: JTAG リセット フラク (JTAG Reset Flag)

このビットはJTAG命令AVR\_RESETによって選んだJTAG リセット レジスタ内の論理1でリセットが起こされると設定(1)されます。このビットは電源ONリセットまたはこのフラグへの論理0書き込みによってリセット(0)されます。

# 境界走査チェーン(Boundary-Scan Chain)

境界走査チェーンにはデジタルI/Oピンは勿論、チップ。外の接続を持つアナログ回路のアナログとデジタル間境界の論理値の設定と監視の能力があります。

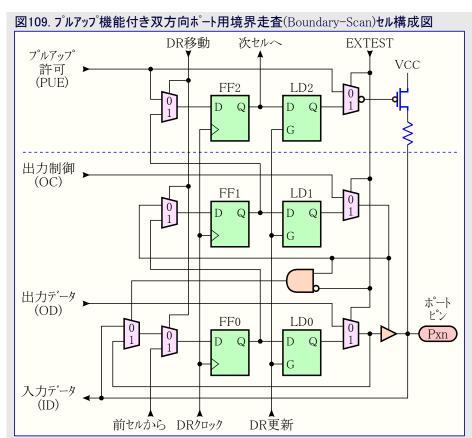
# デジタル ポート ピンの走査

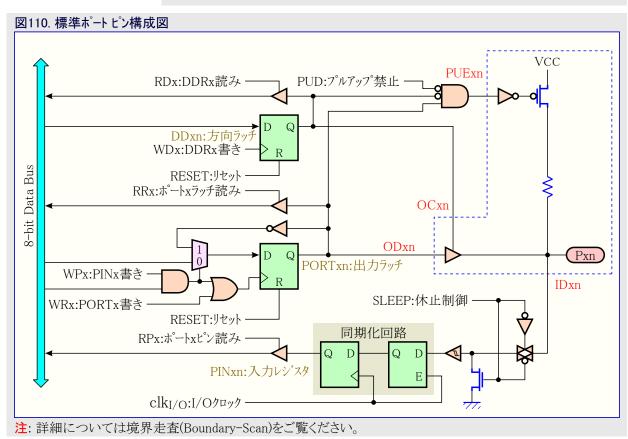
図109.はプルアップ機能付き双方向ポートピン用境界走査(Boundary-Scan)セルを示します。このセルはプルアップ許可(PUExn)用標準境界走査(Boundary-Scan)セルと3つの信号、出力(方向)制御(OCxn)、出力データ(ODxn)、入力データ(IDxn)の組み合わせの双方向ピン用セルの2段の移動レジスタだけから成ります。ポートとピンの添え字は続く記述で使われません。

境界走査(Boundary-Scan)論理回路はデータシート内の図に含まれません。図110.は37頁の「入出力ポート」章で記載される単純なデジタルポートピンを示します。図109.からの境界走査セルの詳細が図110.上の破線内に置き換わります。

交換ポート機能が存在しない場合、入力データ (ID)はPINxnレシ、スタ値(しかしIDは同期化回路を持たない)、出力データ(OD)はPORTxnレシ、スタ、出力制御(OC)はDDxn方向レシ、スタ、プルアップ許可(PUE)は論理合成値(PUD AND DDxn AND PORTxn)に各々対応します。

デジタル交換ポート(ピン)機能は走査チェーン(Scan chain)が実際のピン値を読むために図110.内の破線外側に接続されます。アナログ機能について、それらは外部ピンからアナログ回路へ直結で、走査チェーンはアナログ回路とデジタル論理回路間のインターフェースに挿入されます。



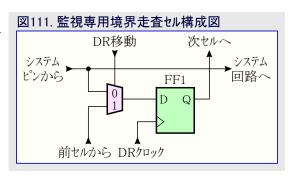






## RESETL°ンの走査

RESETt°ンは標準リセット動作について5V負論理(Low有効)、高電圧並列プログラミングについて12V正論理(High有効)を受け入れます。図111.で示される監視専用セルが5Vリセット信号(RSTT)と12Vリセット信号(RSTHV)の両方に挿入されます。



### クロック ピンの走査

AVRデバイスにはヒューズによって選択可能な多くのクロック種別があります。これらは校正付き内蔵RC発振器、外部RC発振器、外部クロック信号、(高周波数)クリスタル発振子、低周波数クリスタル発振子、セラミック振動子です。

図112.は走査チェーン内で支援される各発振器と外部接続を示します。許可信号は標準境界走査セルで支援され、一方発振器/クロック出力には監視専用セルが付随されます。主クロックに加え、タイマ用発振器も同じ方法で走査されます。校正付き内蔵RC発振器からの出力はこの発振器が外部接続を持たないので走査されません。

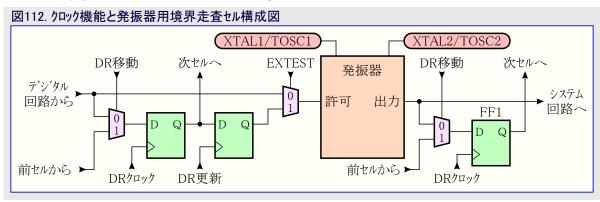


表103.はタイマ用32kHz発振器は勿論、外部クロック ピンXTAL1、XTAL1/XTAL2接続の発振器に対する走査レジスタの一覧です。

#### 表103. 発振器用走査信号

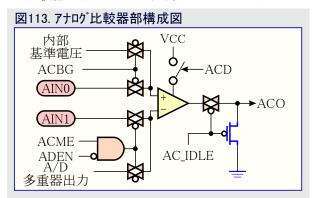
許可信号	走査クロック信号線	クロック種別	未使用時値
EXTCLKEN	EXTCLK(XTAL1)	外部クロック信号	0
OSCON	OSCCK	外部クリスタル発振子 外部セラミック振動子	1
OSC32EN	OSC32CK	低周波数外部クリスタル	1

- 注: 1. 主クロックとして同時に1つよりも多くのクロック元を許可してはいけません。
  - 2. 発振器出力の走査は内部発振器とJTAGのTCKクロック間の周波数偏差のため、予期せぬ結果を与えます。可能なら、外部クロックでの走査が望まれます。
  - 3. クロック設定はヒューズによってプログラミングされます。ヒューズは実行動作時に変更されないので、クロック設定は与えられた応用について考慮して決定されます。使用者は最終システムで使われるのと同じクロック選択で走査することを推奨されます。システム論理回路が休止形態でクロック選択(元)を禁止できるため、許可信号は走査チェーン内で支援され、それ(その禁止)によって(クロックが)供給されない場合、発振器ピンを走査経路から切り離します。

### アナログ比較器の走査

境界走査に関する比較器関連の信号は図113.で示されます。図114.の境界走査セルがこれら信号の各々に付随します。この信号は表104.で記述されます。

この比較器は全てのアナログ入力がデジタルポートピンとしても兼用されるため、純粋な接続試験用に使われる必要がありません。



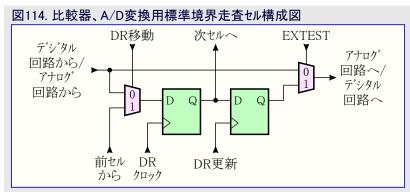
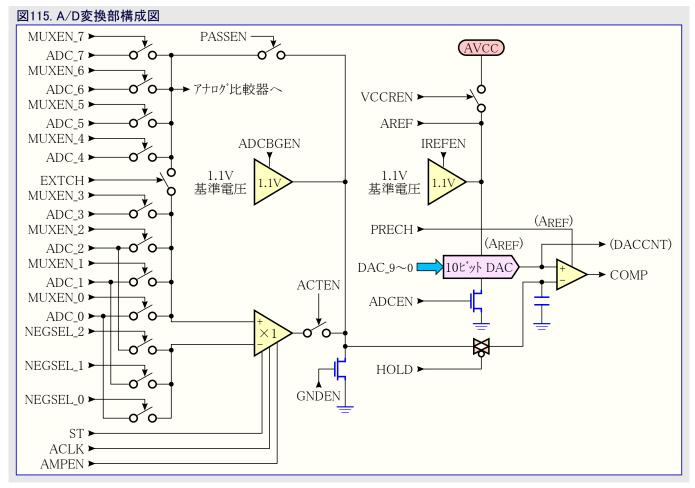


表104. アナログ比較器用境界走査信号

信号名	比較器側での方向	意味	未使用時推奨入力値	推奨入力時出力値
AC_IDLE	入力	真の時にアナログ比較器OFF	1	CPU実行に依存
ACO	出力	アナログ比較器出力	CPU実行時入力	0
ACME	入力	真の時にA/D多重器出力使用	0	CPU実行に依存
ACBG	入力	内蔵基準電圧許可	0	CPU実行に依存

### A/D変換器の走査

図115.はA/D変換部と関連する制御と監視信号の構成図を示します。図114.の境界走査セルがこれら信号の各々に付随されます。このA/D変換器は全アナログ入力がデジタル ポート ピンとしても兼用されるため、純粋な接続試験に使われる必要がありません。



各信号の概要は表105.で示されます。





# 表105. A/D変換用境界走査信号

信号名	A/D変換器側 での方向	意味	未使用時 推奨入力値	推奨入力または 未使用時出力値
COMP	出力	比較器出力	0	0
ACLK	入力	差動段のスイッチトキャパシタのクロック	0	0
ACTEN	入力	差動段⇒比較器接続許可	0	0
ADCBGEN	入力	比較器反転入力へ内部1.1V基準電圧接続	0	0
ADCEN	入力	A/D変換器電源ON信号	0	0
AMPEN	入力	差動段電源ON信号	0	0
DAC_9	入力	D/A変換器デジタル入力ビット9	1	1
DAC_8	入力	D/A変換器デジタル入力ビット8	0	0
DAC_7	入力	D/A変換器デジタル入力ビット7	0	0
DAC_6	入力	D/A変換器デジタル入力ビット6	0	0
DAC_5	入力	D/A変換器デジタル入力ビット5	0	0
DAC_4	入力	D/A変換器デジタル入力ビット4	0	0
DAC_3	入力	D/A変換器デジタル入力ビット3	0	0
DAC_2	入力	D/A変換器デジタル入力ビット2	0	0
DAC_1	入力	D/A変換器デジタル入力ビット1	0	0
DAC_0	入力	D/A変換器デジタル入力ビット0	0	0
EXTCH	入力	ADC3~0の差動段迂回接続	1	1
GNDEN	入力	比較器反転入力GND接続	0	0
HOLD	入力	S&H信号。0=採取、1=保持。差動段使用の場合ACLK =1の時に切り替えなければなりません。	1	1
IREFEN	入力	D/A変換器へのAREFとして内部1.1V基準電圧許可	0	0
MUXEN_7	入力	ADC7 アナログ入力チャネル接続	0	0
MUXEN_6	入力	ADC6 アナログ入力チャネル接続	0	0
MUXEN_5	入力	ADC5 アナログ入力チャネル接続	0	0
MUXEN_4	入力	ADC4 アナログ入力チャネル接続	0	0
MUXEN_3	入力	ADC3 アナログ入力チャネル接続	0	0
MUXEN_2	入力	ADC2 アナログ入力チャネル接続	0	0
MUXEN_1	入力	ADC1 アナログ入力チャネル接続	0	0
MUXEN_0	入力	ADC0 アナログ入力チャネル接続	1	1
NEGSEL_2	入力	ADC2⇒反転差動入力接続	0	0
NEGSEL_1	入力	ADC1⇒反転差動入力接続	0	0
NEGSEL_0	入力	ADC0⇒反転差動入力接続	0	0
PASSEN	入力	差動段迂回路許可	1	1
PRECH	入力	比較器出力ラッチのプリチャーシ゛(負論理)	1	1
ST	入力	AMPEN=1後の最初の2ACLK周期間1ならば、差動段 出力がより早く安定	0	0
VCCREN	入力	D/A変換器へのAREFとしてAVCC許可	0	0

注: 図115.内の不正なスイッチ設定は信号衝突の原因になり、デバイスを損傷するかもしれません。それらは多くの入力を図115.で出力比較器の反転入力のS&Hへ選びます。1つのADCピン、内部基準電圧、GNDのどれか1つの経路だけが選ばれることを確認してください。

走査中にA/D変換器が使われないなら、表105.の推奨入力値が使われるべきです。使用者は走査中に差動段を使わないことが推 奨されます。スィッチトキャパシタを基にした差動段は走査チェーン内で使われる時に達成が難しい高速な操作とタイミング精度が必要です。 従って差動段の操作に関する詳細は提供されません。

AVRのA/D変換部はデジタル論理回路内に実装した逐次比較方式と図115.で示したアナログ回路が基です。境界走査使用時、問題は印加したアナログ電圧が或る範囲内で計測されるのを常に保証することです。これは逐次比較動作を行わずして容易に実行でき、DAC\_9~0デジタル(信号)線に下限値を加え、比較器からの出力がLow(0)であるのを確認し、その後DAC\_9~0デジタル(信号)線に上限値を加え、比較器からの出力がHigh(1)であるのを検証します。

このA/D変換器は全てのアナログ入力がデジタルポートピンとしても兼用されるため、純粋な接続試験に使われる必要がありません。 A/D変換を使う時に次のことに留意してください。

- ADCチャネルで使うポート ピンは信号の衝突を避けるため、プルアップを禁止した入力に設定されなければなりません。
- 通常動作ではA/D変換部許可時に(10回の比較で構成される)擬似変換が実行されます。使用者はA/D変換許可後、A/D変換信号の制御/監視をする前に最低200ns待機するか、または最初の変換結果を使う前に擬似変換の実行を推奨されます。
- DAC値はHOLD信号がLow(0:採取)の時に中間値(\$200)で安定していなければなりません。

例として、電源電圧5VでAREFが外部的にVCCへ接続される時のADCチャネル3で1.5V±5%の入力信号を検証する手順を考察してください。

上限: 1024×1.5V×1.05÷5V=323=\$143 下限: 1024×1.5V×0.95÷5V=291=\$123

表106.の手順内で他の値が与えられる以外は表105.の推奨値が使われます。(表106.には)走査チェーンのDAC値とポート ピン値だけが 示されます。「動作」欄は続く行内の値で境界走査チェーン レジスタを満たす前にどのJTAG命令を使うべきかを記述します。表の同一列 上のデータを走査入力する時に走査出力のデータで検証が行われるべきです。

表106.	A/D変換使用手順								
手順番号	動作	ADCEN	DAC_9~0	MUXEN_7~0	HOLD	PRECH	PF3 データ	PF3 制御	PF3 プルアップ。許可
1	SAMPLE_PRELOAD	1	\$200	\$08	1	1	0	0	0
2	EXTEST	1	\$200	\$08	0	1	0	0	0
3		1	\$200	\$08	1	1	0	0	0
4		1	\$123	\$08	1	1	0	0	0
5		1	\$123	\$08	1	0	0	0	0
6	0となるべきCOMP走査出力の検証	1	\$200	\$08	1	1	0	0	0
7		1	\$200	\$08	0	1	0	0	0
8		1	\$200	\$08	1	1	0	0	0
9		1	\$143	\$08	1	1	0	0	0
10		1	\$143	\$08	1	0	0	0	0
11	1となるべきCOMP走査出力の検証	1	\$200	\$08	1	1	0	0	0

この手法の使用ではHOLD信号の束縛タイミングがTCKクロック周波数を押さえます。手順の第5段階でのHOLDのHigh(1)に保持で、TCKクロック周波数は最大保持時間(thold max)で割られた走査ビット数の最低5倍でなければなりません。





# ATmega169の境界走査(Boundary-Scan)順

表107.は境界走査チェーンがデータ経路として選ばれる時のTDIとTDO間の走査順を示します。ビット0が最下位ビットで走査の最初に入出力されます。走査順は可能な限りピン配置順に従います。このためにポートAのビットは他のポートと違うビット順で走査されます。この規則の例外はアナログ回路が接続される物理ピンに関係なく走査チェーンの最上位を構成するアナログ回路用走査チェーンです。図109.上で、Pxn出力はFF0、Pxn制御はFF1、Pxnプルアップ許可はFF2に各々対応します。ポートFのビット4,5,6,7はJTAG許可時にTAPピンを構成するので走査チェーンには存在しません。

表107. ATmega169境界走查(Boundary-Scan)順

ピット	Almega169境界走信号名	基本単位	ヒット	信号名	基本単位	ピット	信号名	基本単位
197	AC_IDLE	本个平位	144	PE4.Control	<b>坐</b> 本平位	91	PD0.Control	<u> </u>
196	ACO ACO	アナログ	143	PE4.Pullup_Enable		90	PD0.Pullup_Enable	
195	ACME	比較器	142	PE5.Data		89	PD1.Data	
194	ACBG	7072111	141	PE5.Control		88	PD1.Control	
193	COMP		140	PE5.Pullup_Enable		87	PD1.Pullup_Enable	
192	ACLK		139	PE6.Data	ホ°ートE	86	PD2.Data	
191	ACTEN		138	PE6.Control	, ,_	85	PD2.Control	
190	PRIVATE_SIGNAL1	(注)	137	PE6.Pullup_Enable		84	PD2.Pullup_Enable	
189	ADCBGEN		136	PE7.Data		83	PD3.Data	
188	ADCEN		135	PE7.Control		82	PD3.Control	
187	AMPEN		134	PE7.Pullup_Enable		81	PD3.Pullup_Enable	
186	DAC_9		133	PB0.Data		80	PD4.Data	ポ゚ートD
185	DAC_8		132	PB0.Control		79	PD4.Control	
184	DAC_7		131	PB0.Pullup_Enable		78	PD4.Pullup_Enable	
183	DAC_6		130	PB1.Data		77	PD5.Data	
182	DAC_5		129	PB1.Control		76	PD5.Control	
181	DAC_4		128	PB1.Pullup_Enable		75	PD5.Pullup_Enable	
180	DAC_3		127	PB2.Data		74	PD6.Data	
179	DAC_2		126	PB2.Control		73	PD6.Control	
178	DAC_1		125	PB2.Pullup_Enable		72	PD6.Pullup_Enable	
177	DAC_0		124	PB3.Data		71	PD7.Data	
176	EXTCH	A/D変換	123	PB3.Control		70	PD7.Control	
175	GNDEN		122	PB3.Pullup_Enable		69	PD7.Pullup_Enable	
174 173	HOLD IREFEN		121 120	PB4.Data PB4.Control		68 67	PG0.Data PG0.Control	
173	MUXEN_7		119	PB4.Pullup_Enable		66	PG0.Control PG0.Pullup_Enable	
171	MUXEN_6		118	PB5.Data		65	PG1.Data	ホ°ートG
170	MUXEN_5		117	PB5.Control		64	PG1.Control	
169	MUXEN_4		116	PB5.Pullup_Enable		63	PG1.Pullup_Enable	
168	MUXEN_3		115	PB6.Data		62	PC0.Data	
167	MUXEN_2		114	PB6.Control		61	PC0.Control	
166	MUXEN_1		113	PB6.Pullup_Enable		60	PC0.Pullup_Enable	
165	MUXEN_0		112	PB7.Data		59	PC1.Data	
164	NEGSEL_2		111	PB7.Control		58	PC1.Control	
163	NEGSEL_1		110	PB7.Pullup_Enable		57	PC1.Pullup_Enable	
162	NEGSEL_0		109	PG3.Data		56	PC2.Data	
161	PASSEN		108	PG3.Control		55	PC2.Control	
160	PRECH		107	PG3.Pullup_Enable		54	PC2.Pullup_Enable	
159	ST		106	PG4.Data	ポートG	53	PC3.Data	
158	VCCREN		105	PG4.Control		52	PC3.Control	
157	PE0.Data		104	PG4.Pullup_Enable	(45)	51	PC3.Pullup_Enable	ホ°ートC
156	PE0.Control		103	PRIVATE_SIGNAL2	(注)	50	PC4.Data	, , , ,
155	PE0.Pullup_Enable		102	RSTT	リセット回路	49	PC4.Control	
154	PE1.Data		101	RSTHV	(監視専用セル)	48	PC4.Pullup_Enable	
153	PE1.Control		100	EXTCLKEN	→ h= . h / ₹% +L=	47	PC5.Data	
152	PE1.Pullup_Enable	+° 15	99	OSCON	主クロック/発振 器用許可信号	46	PC5.Control	
151	PE2.Data	ホ°ートE	98	RCOSCEN	一番用計刊信号	45	PC5.Pullup_Enable	
150 149	PE2.Control		97 96	OSC32EN EXTCLK(XTAL1)		44 43	PC6.Data PC6.Control	
149	PE2.Pullup_Enable PE3.Data		95	OSCCK	主クロック用	43	PC6.Control PC6.Pullup_Enable	
147	PE3.Control		93	RCCK	入力/発振器	41	PC7.Data	
146	PE3.Control PE3.Pullup_Enable		93	OSC32CK	(監視専用セル)	40	PC7.Control	
145	PE4.Data		92	PD0.Data	ポートD	39	PC7.Pullup_Enable	
_			_	たいとして走杏され		次百へ		

注: PRIVATE\_SIGNAL1とPRIVATE\_SIGNAL2は常に0として走査されるべきです。 次頁へ続く

# 表107 (続き). ATmega169境界走査(Boundary-Scan)順

衣10/	(舵さ)、AImegalog.	<b>児介止宜</b> (DOUI)(	iary-sc	an/ <b>川貝</b>	
ビット	信号名	基本単位	ピット	信号名	基本単位
38	PG2.Data		25	PA4.Control	
37	PG2.Control	ポ゚ートG	24	PA4.Pullup_Enable	
36	PG2.Pullup_Enable		23	PA3.Data	
35	PA7.Data		22	PA3.Control	
34	PA7.Control		21	PA3.Pullup_Enable	
33	PA7.Pullup_Enable		20	PA2.Data	
32	PA6.Data		19	PA2.Control	ホ°ートA
31	PA6.Control	ポートA	18	PA2.Pullup_Enable	
30	PA6.Pullup_Enable	w-PA	17	PA1.Data	
29	PA5.Data		16	PA1.Control	
28	PA5.Control		15	PA1.Pullup_Enable	
27	PA5.Pullup_Enable		14	PA0.Data	
26	PA4 Data		13	PA0 Control	

ピット	信号名	基本単位		
12	PA0.Pullup_Enable	ホ°ートA		
11	PF3.Data			
10	PF3.Control			
9	PF3.Pullup_Enable			
8	PF2.Data			
7	PF2.Control			
6	PF2.Pullup_Enable	ポペートF		
5	PF1.Data	Ψ _LL		
4	PF1.Control			
3	PF1.Pullup_Enable			
2	PF0.Data			
1	PF0.Control			
0	PF0.Pullup_Enable			

# 境界走査記述言語(Boundary-Scan Description Language)ファイル

境界走査記述言語(BSDL)ファイルは境界走査が可能なデバイスを自動検査生成ソフトウェアによって使われる標準形式で記述します。境界走査データレジ、スタ内のビットの順番と機能がこの記述内に含まれます。ATmega169用のBSDLファイルは入手可能です。





# ブートローダ支援 - 書き込み中読み出し可能な自己プログラミング

ブートロータ・支援はMCU自身によるプログラムコート・のダウンロード・とアップロード・用の真の書き込み中の読み出しが可能な自己プログラミング機構を提供します。この特徴はフラッシュメモリに常駐するブートロータ・プログラムを使うMCUによって制御される柔軟な応用ソフトウェア更新を可能にします。 ブートロータ・プログラムはフラッシュメモリ内にコート・を書き(プログラム)、コート・を読み、またはプログラムメモリからコート・を読むのに、利用可能なデータインターフェースと関連する規約のどれもが使えます。 ブートロータ・領域内のプログラムコート・はブートロータ・メモリを含むフラッシュメモリ全体を書く能力を持ちます。 従ってブートロータ・は自身をも変更でき、この機能がそれ以上必要とされないなら、そのコート・から自身を消去することもできます。 ブートロータ・メモリの容量はヒューズ・で設定可能で、ブートロータ・は個別に設定可能な2組の独立したブート施錠ビットを持ちます。 これは異なる保護基準を選ぶ独特な柔軟性を使用者に与えます。

## 特徴

- 書き込み中読める(Read-While-Write)自己プログラミング
- 柔軟性のあるブート ローダ メモリ容量
- 高い安全性(柔軟な保護用の独立したブート施錠ビット)
- リセット ベクタ選択用の独立したヒューズ
- 最適化されたページ容量(注1)
- 効率的なコード手法
- 効率的な読み-変更-書き(リート゛モデファイ ライト)支援

**注1**: ページはプログラミング中に使われる多数のバイトから成るフラッシュ メモリの区画です(177頁の**表121**.参照)。このページ構成は通常動作に影響を及ぼしません。

## フラッシュ メモリの応用領域とブート ローダ領域

フラッシュ メモリは応用領域とブート ローダ領域の2つの主な領域で構成されます(図117.参照)。各領域の容量は174頁の表113.と図117.で示されるようにBOOTSZヒューズによって設定されます。これら2つの領域は個別の施錠ビットの組を持つため、異なる保護基準を持てます。

#### 応用領域

応用領域は応用コートを格納するのに使われるフラッシュ メモリの領域です。応用領域用保護基準は応用ブート施錠ビット(ブート施錠ビット0)によって選べます(168頁の表109.参照)。応用領域から実行される時にSPM命令が禁止されるので、応用領域はどんなブート ロータ コート・も決して格納し得ません。

#### ブート ローダ領域 (BLS)

応用領域が応用コート・格納用に使われるのに対して、SPM命令はBLSから実行する時にだけプログラミングを始められるので、ブートロータ・ソフトウェアはBLSに格納されなければなりません。SPM命令はBLS自身を含む全てのフラッシュメモリをアクセスできます。ブートロータ・領域用保護基準はブートローダ・施錠ビット(フート・施錠ビット))によって選べます(168頁の表110.参照)。

### フラッシュ メモリの書き込み中に読み出し可能な領域と不能な領域

どちらのアト・レスがプログラミングされるかによって、CPUが書き込み中の読み出しを支援するか、ブートローダ・ソフトウェアが更新中にCPUが停止されるかのどちらです。上で記述されるようなBOOTSZヒュース、によって設定可能な2つの領域に加え、フラッシュ メモリは書き込み中読み出し可能な(RWW)領域と書き込み中読み出し不能な(NRWW)領域の2つの固定領域にも分けられます。RWWとNRWW領域間の境界は167頁の図117.と174頁の表114.で与えられます。この2つの領域間の主な違いを次に示します。

- RWW領域側に配置されたヘージを消去または書くとき、NRWW領域はその動作中に読むことができます。
- NRWW領域側に配置されたページを消去または書くとき、その全ての動作中にCPUは停止されます。

ブートロータ・ソフトウェア動作中、使用者ソフトウェアがRWW領域側に配置されるどのコート・も決して読めないことに注意してください。「書き込み中読み出し可能領域」という記述はプログラミング(消去または書き込み)される領域としての引用で、ブートロータ・ソフトウェアが更新中に実際に読まれる領域ではありません。

訳補: 上の記述はNRWW領域からRWW領域をプログラミングするという前提で、消去または書き込みを行う側ではなく、行われる側でこの名称が定義されていることを意味します。即ち、NRWW領域からRWW領域をプログラミングすると、NRWW領域のプログラムは通常通り動作する(即ち読める)ので、プログラミングされる側はRWW領域と名付けられ、この逆ではCPUが停止する(即ち読めない)ので、NRWW領域と名付けられているという意味です。

### RWW - 書き込み中読み出し可能領域

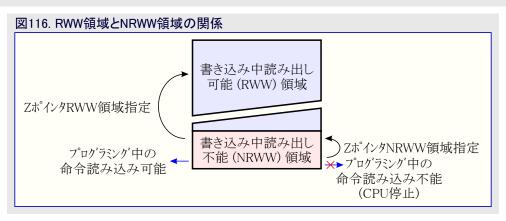
ブートロータ、ソフトウェア更新がRWW領域側のペーシ、をプログラミングする場合、フラッシュメモリからコート、を読むことが可能ですが、NRWW領域に配置されるコート、だけです。プログラミング、実行中、そのソフトウェアはRWW領域が決して読まれないことを保証しなければなりません。使用者ソフトウェアがプログラミング中に(例えば、CALL,JMP,LPM系命令または割り込みによって)RWW領域側に配置されるコートを読もうとすると、そのソフトウェアは未知の状態へ行き着くかもしれません。これを避けるために割り込みは禁止、またはブートロータ、領域へ移動のどちらかにされるべきです。ブートロータ、領域は常にNRWW領域に配置されます。RWW領域が読み出しに対して妨げられている限り、SPM命令制御/状態レシ、スタ(SPMCSR)のRWW領域多忙(RWWSB)と、ットは論理1として読みます。プログラミングが完了した後、RWW領域に配置したコート、を読む前にRWWSBはソフトウェアによって解除(0)されなければなりません。RWWSBを解除(0)する方法の詳細については169頁の「SPM命令制御/状態レジ、スタ(SPMCSR)」をご覧ください。

## NRWW - 書き込み中読み出し不能領域

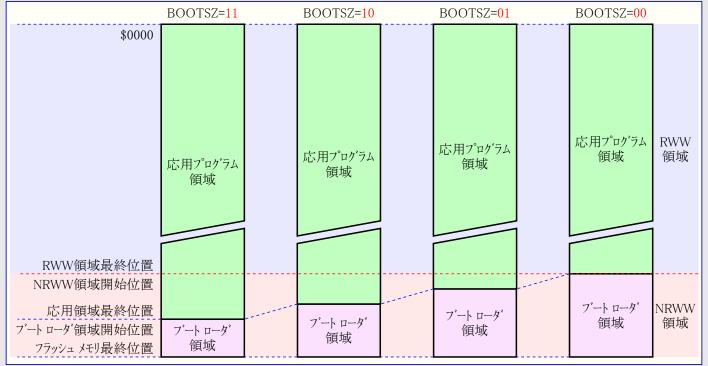
NRWW領域に配置したコードはブートロータ゛ソフトウェアがRWW領域内のヘ゜ーシ゛を更新する時に読めます。 ブートロータ゛コート゛がNRWW領域を更新するとき、全てのヘ゜ーシ゛消去またはヘ゜ーシ゛書き込み動作中にCPUが停止されます。

#### 表108. 書き込み中読み出し可能機能

プログラミング中にZポインタで指定される領域	プログラミング中に読める領域	CPU動作	RWW機能支援
RWW領域	NRWW領域	通常動作	あり
NRWW領域	なし	停止	なし



### 図117. 選択によるプログラム用フラッシュ メモリの領域分割



注: 図内の各パラメータについては174頁の表113.で与えられます。





## ブートローダ施錠ビット

ブート ロータ 能力が必要とされないなら、フラッシュ メモリ全体が応用コート 用に利用可能です。ブート ローダ は個別に設定可能な独立した2組のブート施錠ビットを持ちます。これは異なる保護基準を選ぶ独特な柔軟性を使用者に与えます。

使用者は以下を選べます。

- MCUによって更新するソフトウェアからフラッシュ メモリ全体を保護
- MCUによって更新するソフトウェアからフラッシュ メモリのブート ローダ領域だけを保護
- MCUによって更新するソフトウェアからフラッシュ メモリの応用領域だけを保護
- フラッシュ メモリ全体で更新するソフトウェアを許可

より多くの詳細については表109.と表110.をご覧ください。ブート施錠ビットはソフトウェアと直列または並列のプログラミング動作で設定(0)できますが、これらのビットはチップ消去指令によってのみ解除(1)できます。一般書き込み禁止(LB動作種別2)はSPM命令によるフラッシュメモリのプログラミングを制御しません。同様に、試みられたなら、一般読み書き禁止(LB動作種別3)はLPM命令とSPM命令による読み込みも書き込みも制御しません。(訳補:一般LBはLPM/SPM命令に関して無関係の意)

#### 表109. 応用領域に対する保護種別(0=プログラム、1=非プログラム)

BLB0 動作種別	BLB02	BLB01	保護種別						
1	1	1	LPM, SPM命令が応用領域をアクセスすることに対して制限はありません。						
2	1	0	SPM命令は応用領域に書くことを許されません。						
3	0	0	SPM命令による応用領域への書き込みと、ブートローが領域でのLPM命令による応用領域からの読み込みが許されません。(注)						
4	0	1	ブートローダ領域でのLPM命令による応用領域からの読み込みが許されません。(注)						

注: BLB02=0で、割り込みベクタがブート ローダ領域に配置されていると、応用領域での実行時に割り込みが禁止されます。

#### 表110. ブートローダ 領域に対する保護種別(0=プログラム、1=非プログラム)

BLB1 動作種別	BLB12	BLB11	保護種別
1	1	1	LPM, SPM命令がブートローダ領域をアクセスすることに対して制限はありません。
2	1	0	SPM命令はブートローダ領域に書くことを許されません。
3	0	0	SPM命令によるブートローダ領域への書き込みと、応用領域でのLPM命令によるブートローダ領域からの読み込みが許されません。(注)
4	0	1	応用領域でのLPM命令によるブート ローダ領域からの読み込みが許されません。(注)

注: BLB12=0で、割り込みへ'クタが応用領域に配置されていると、ブートローゲ領域での実行時に割り込みが禁止されます。

### ブートローダブログラムへの移行

ブートロータ、への移行は応用プログラムから分岐(Jump)または呼び出し(Call)によって行います。これはUSARTやSPIインターフェース経由で受信した指令のような起点によって始められるかもしれません。代わりに、リセット後にリセット、グタがブートロータ、領域開始アトンスを指示するようにブートリセット(BOOTRST)ヒュース、をプログラム(0)できます。この場合、ブートロータがリセット後に開始されます。応用コートが設定された(書かれた)後、そのプログラム(ブートロータ、)は応用コートの実行を始めることができます。このヒュースがMCU自身によって変更できないことに注意してください。これは一旦ブートリセットヒュースがプログラム(0)されると、リセット、グタは常にブートロータ、リセットを指示し、このヒュースが直列プログラミング、並列プログラミング、またはJTAGインターフェースを通してのみ変更できることを意味します。

# 表111. ブート リセット ヒュース (O=プロク ラム、1=非プロク ラム)

BOOTRST		リセット後実行開始アトレス (リセット ヘブクタ)
0	ブート ロータ゛リセット	ブートローダ開始アドレス(174頁の表113.参照)
1	応用リセット	\$0000

## SPM命令制御/状態レジスタ (Store Program Memory Control and Status Register) SPMCSR

このレシ、スタはフートローダ、動作を制御するために必要とされる制御ビットを含みます。

ヒ゛ット	7	6	5	4	3	2	1	0	
\$37 (\$57)	SPMIE	RWWSB	-	RWWSRE	BLBSET	PGWRT	PGERS	SPMEN	SPMCSR
Read/Write	R/W	R	R	R/W	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

# • ビット7 - SPMIE: SPM操作可割り込み許可 (SPM Interrupt Enable)

SPMIEL「ットが1を書かれ、ステータスレジスタ(SREG)の全割り込み許可(I)ヒットが設定(1)されているなら、SPM操作可割り込みが許可されます。SPM操作可割り込みはSPMCSRのSPM操作許可(SPMEN)ヒットが解除(0)されている限り実行されます。

## • ビット6 - RWWSB: RWW領域多忙 (Read-While-Write Section Busy)

RWW領域に自己プログラミング(ヘーン)消去またはヘーン・書き込み)操作が開始されると、RWWSBがハートウェアによって設定(1)されます。 RWWSBビットが設定(1)されると、RWW領域はアクセスできません。自己プログラミングが完了された後にRWWSREビットが1を書かれると、このRWWSBビットは解除(0)されます。 同様に、ヘーン・設定操作が開始されると、RWWSBビットは自動的に解除(0)されます。

#### • **ビット5 - Res**: 予約 (Reserved)

このビットは予約されており、常に0として読みます。

#### • ビット4 - RWWSRE: RWW領域読み出し許可 (Read-While-Write Section Read Enable)

RWW領域にプログラミング(ペーシ)消去またはペーシ、書き込み)すると、RWW領域は(RWWSBがハートウェアによって設定(1)され)読み出しに対して防がれます。RWW領域を再許可するために使用者ソフトウェアはプログラミングが完了(SPMENが解除(0))されるまで待たねばなりません。その後、RWWSREtットがSPMENと同時に1を書かれると、次の4クロック周期内のSPM命令がRWW領域を再許可します。ペーシ消去またはペーシ、書き込みでフラッシュメモリが多忙(SPMEN=1)の間、RWW領域は再許可できません。フラッシュメモリが設定されている間にRWWSREtットが書かれると、フラッシュメモリ設定操作は失敗し、設定したデータが失われます。

#### • ビット3 - BLBSET: ブート施錠ビット設定 (Boot Lock Bits Set)

このビットがSPMENと同時に1を書かれると、次の4クロック周期内のSPM命令はR0のデータに従ったブート施錠ビットを設定します。R1のデー タとZポインタのアドレスは無視されます。BLBSETビットは施錠ビットの設定完了で、または4クロック周期内にSPM命令が実行されない場合、自動的に解除(0)されます。

SPMCSRでBLBSETとSPMENが設定(1)された後の3クロック周期内のLPM命令は(ZポインタのZ0によって)ヒューズビットまたは施錠ビットのどちらかを転送先レジスタに読みます。詳細については172頁の「ソフトウェアからのヒューズビットと施錠ビットの読み出し」をご覧ください。

#### • ビット2 - PGWRT : ページ書き込み (Page Write)

このビットがSPMENと同時に1を書かれると、次の4クロック周期内のSPM命令は一時緩衝部に格納したデータでページ書き込みを実行します。 ページ アドレスはZポインタの上位部から取得されます。 R1とR0のデータは無視されます。 PGWRTビットはページ書き込みの完了で、または4クロック周期内にSPM命令が実行されない場合、自動的に解除(0)されます。 NRWW領域がアドレス指定されると、ページ全体の書き込み動作中にCPUは停止されます。

### • ビット1 - PGERS : ページ消去 (Page Erase)

このビットがSPMENと同時に1を書かれると、次の4クロック周期内のSPM命令はページ消去を実行します。 ページアドレスはZポインタの上位部から取得されます。 R1とR0のデータは無視されます。 PGERSビットはページ消去の完了で、または4クロック周期内にSPM命令が実行されない場合、自動的に解除(0)されます。 NRWW領域がアドレス指定されると、ページ全体の消去中にCPUは停止されます。

#### • ビット0 - SPMEN : SPM操作許可 (Store Program Memory Enable)

このビットは次の4クロック周期間のSPM命令を許可します。このビットがRWWSRE、BLBSET、PGWRT、PGERSのどれかと共に1を書かれると、続くSPM命令は特別な意味を持ちます(上の記述をご覧ください)。SPMENだけが書かれると、続くSPM命令はZポインタによってアトレス指定したペーシ、一時緩衝部へR1:R0の値を格納します。Zレジスタの最下位ビットは無視されます。SPMENビットはSPM命令の完了で、または4クロック周期内にSPM命令が実行されない場合、自動的に解除(0)されます。ページ消去とページ書き込み中、SPMENビットはその動作が完了されるまで1に留まります。

下位5ビットに10001, 01001, 00101, 00011, 00001以外のどんな組み合わせを書いても無効です。





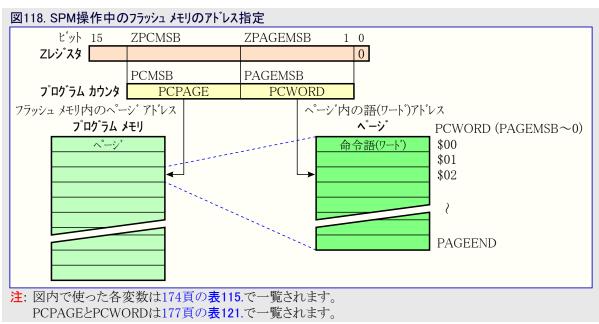
# 自己プログラミング中のフラッシュメモリのアドレス指定

Zポインタ(レジスタ)はSPM命令でのアドレス指定に使われます。

ピット	15	14	13	12	11	10	9	8
ZH (R31)	Z15	Z14	Z13	Z12	Z11	Z10	Z9	Z8
ピット	7	6	5	4	3	2	1	0
ZL (R30)	Z7	Z6	Z5	Z4	Z3	Z2	Z1	Z0

フラッシュ メモリがペーシ で構成されるため(177頁の表121.参照)、プログラム カウンタ(アドレス ポインタ)は2つの違う領域を持つように取り扱われます。1つの領域は下位側ビットから成り、ペーシ 内の語(ワード)をアドレス指定し、一方上位側ビットはそのページをアドレス指定します。これは図118.で示されます。ページ 消去とページ 書き込み操作が個別にアドレス指定されることに注意してください。従ってブートローダ ソフトウェアはページ 消去とページ 書き込み操作の両方で同じページをアドレス指定することが最も重要です。一旦プログラミング操作が開始されると、このアドレスはラッチされ、Zポインタは他の操作に使えます。

Zポインタを使わないSPM操作はブート ローダ施錠ビット設定だけです。この操作でZレジスタの内容は無視され、無効です。LPM命令もアドレスを格納するのにZポインタを使います。この命令はフラッシュ メモリのバイト単位をアドレス指定するので、Zポインタの最下位ビット(Z0)も使われます。



## フラッシュ メモリの 自己プログラミング

プログラム メモリはヘーシ 単位形式で更新されます。ヘーシー 時緩衝部へ格納したデータでヘージを書く前にそのヘージが消去されなければなりません。ヘージー 時緩衝部はSPM命令使用時毎の1語(ワート)で満たされ、この緩衝部はヘージ消去命令前、またはヘージ消去とヘージ書き込み操作間のどちらかで満たすことができます。

#### 手段1(ページ消去前の一時緩衝部格納)

- ページー時緩衝部を満たしてください。
- ペーシ消去を実行してください。
- ページ書き込みを実行してください。

#### 手段2(ページ消去後の一時緩衝部格納)

- ペーシ 消去を実行してください。
- ページ一時緩衝部を満たしてください。
- ページ書き込みを実行してください。

へ°ーシ'の一部の変更だけが必要な場合、消去前にへ°ーシ'の残す部分は(例えばへ°ーシ'一時緩衝部に)保存されなければならず、その後に改めて書かれます。**手段1.**を使う場合、初めにへ°ーシ'を読んで必要な変更を行い、その後に変更したデータを書き戻すことを使用者ソフトウェアに許す効率的な読みー修正ー書き(リート' モデ'ファイライト)機能をブート ロータ'が提供します。**手段2.**が使われる場合、へ°ーシ'が既に消去されているため、格納中の旧データを読むことができません。ヘ°ーシ'一時緩衝部は乱順でアクセスできます。へ°ーシ'消去とへ°ーシ'書き込み操作の両方で使われるへ°ーシ'アト・レスは同じへ°ーシ'をアト・レス指定することが非常に重要です。アセンブリ言語でのコート'例については173頁の「アセンブリ言語による簡単なブートロータ'例」をご覧ください。

## SPM命令によるページ消去の実行

ページ消去を実行するにはZポインタにアドレスを設定してSPM命令制御/状態レジスタ(SPMCSR)に'X0000011'を書き、SPMCSR書き込み後4クロック周期内にSPM命令を実行してください。R1とR0のデータは無視されます。ページ アドレスはZポインタのPCPAGEに書かれなければなりません。この操作中、Zポインタの他のビットは無視されます。

- RWW領域のペーシ消去 : ペーシ消去中、NRWW領域は読めます。
- NRWW領域のページ消去:ページ消去中、CPUは停止されます。

## ページ一時緩衝部の設定(ページ設定)

命令語(ワード)を(ページー時緩衝部に)書くにはZポインタにアドレス、R1:R0にデータを設定してSPMCSRに'00000001'を書き、SPMCSR書き込み後4クロック周期内にSPM命令を実行してください。ZポインタのPCWORDの内容は一時緩衝部のデータのアドレスに使われます。一時緩衝部はページ書き込み操作後、またはSPMCSRのRWWSREビット書き込みによって自動的に消去されます。システム リセット後も消去されています。一時緩衝部を消去せずに各アドレスへ複数回書くことができないことに注意してください。

SPMへージ設定操作の途中でEEPROMが書かれると、設定した全データが失われます。

# ページ書き込みの実行

へ°ーシ`書き込みを行うにはZポインタにアドレスを設定してSPMCSRに'X0000101'を書き、SPMCSR書き込み後4クロック周期内にSPM命令を実行してください。R1とR0のデータは無視されます。ページ アドレスは(Zポインタの)PCPAGEに書かれなければなりません。この操作中にZポインタの他のビットは0を書かれなければなりません。

- RWW領域のページ書き込み : ページ書き込み中、NRWW領域は読めます。
- NRWW領域のページ書き込み:ページ書き込み中、CPUは停止されます。

### SPM操作可割り込みの使用法

SPM操作可割り込みが許可されると、SPMCSRのSPMENt'ットが解除(0)されている時にSPM操作可割り込みが継続的に発生します。これはソフトウェアでSPMCSRをポーリングする代わりにこの割り込みが使えることを意味します。SPM操作可割り込みを使うとき、割り込みが読み出しに対して防がれる時にRWW領域をアクセスするのを避けるために、割り込みへ、クタはブートローダ領域(BLS)へ移動されるべきです。割り込み(へ、クタ)の移動法は30頁の「割り込み」で記述されます。

# ブートローダ領域更新中の考慮

ブート施錠ビット11(BLB11)が非プログラム(1)にされたままとすることによって使用者がブートローダ領域に更新を許す場合、特別な注意が 祓われなければなりません。ブートローダ自身への予期せぬ書き込みはブートローダ全体を不正にし得て、更にソフトウェアの更新が不可能 になるかもしれません。ブートローダ自体の変更が必要ないなら、内部ソフトウェアのどんな変更からもブートローダを保護するためにブート施錠ビット11(BLB11)をプログラム(0)することが推奨されます。

#### 自己プログラミング中のRWW領域読み込みの防止

自己プログラミング中(ページ消去もページ書き込みも)、RWW領域は読み出しに対して常に防がれます。使用者ソフトウェアそれ自身が自己プログラミング操作中にこの領域がアトレス指定されるのを防止しなければなりません。SPMCSRのRWWSBはRWW領域が多忙である限り設定(1)されます。自己プログラミング中の割り込みベクタ表は30頁の「割り込み」で記述されるようにフートローダ領域(BLS)へ移動されるべきか、または割り込みが禁止されなければなりません。プログラミングが完了した後にRWW領域をアトレス指定する前に、使用者ソフトウェアはRWWSREの書き込みによってRWWSBを解除(0)しなければなりません。例については173頁の「アセンブリ言語による簡単なブートローダ例」をご覧ください。

### SPM命令によるブートローダ施錠ビットと一般施錠ビットの設定

ブートローダ・施錠ビットと一般施錠ビットを解除(0)するには希望したデータをR0に設定してSPMCSRに'X0001001'を書き、SPMCSR書き込み後4クロック周期内にSPM命令を実行してください。

ピット	7	6	5	4	3	2	1	0
R0	1	1	BLB12	BLB11	BLB02	BLB01	LB2	LB1

フラッシュ メモリのアクセスに影響を及ぼすブート ローダ施錠ビットの各種設定法については**表109**.と**表110**.をご覧ください。

ROのビット5~0が解除(0)される場合、SPMCSRでSPMENとプート施錠ビット設定(BLBSET)ビットが設定(1)された後4クロック周期内にSPM命令が実行されると、対応する施錠ビットがプログラム(0)されます。この操作中、Zポインタは関係ありませんが、将来との共通性のため、(施錠ビット読み出しに使われるのと同じ)\$0001でZポインタを設定することが推奨されます。将来との共通性のため、施錠ビット書き込み時にROのビット7,6は1に設定することも推奨されます。施錠ビットをプログラミングするとき、この操作中に全てのフラッシュメモリは読むことができます。





## SPM命令での書き込み時のEEPROM書き込みによる妨害

EEPROM書き込み動作がフラッシュ メモリへの全ソフトウェア プログラミングを妨げることに注意してください。ソフトウェアからのヒューズと施錠ビット読み出しもEEPROM書き込み動作中、妨げられます。使用者はEEPROM制御レジスタ(EECR)のEEPROM書き込み許可(EEWE)ビットを検査し、SPM命令制御/状態レジスタ(SPMCSR)へ書く前にこのビットが解除(0)されているのを確認することが推奨されます。

### ソフトウェアからのヒューズビットと施錠ビットの読み出し

ソフトウェアからヒュース」と施錠ビットの両方を読むことが可能です。施錠ビットを読むにはZポインタに \$0001を設定し、SPMCSRのSPMENとブート施錠ビット設定(BLBSET)ビットを設定(1)してください。SPMENとBLBSETビットがSPMCSRに設定された後、3 CPU周期内にLPM命令が実行されると、施錠ビットの値が転送先レシ、スタに格納されます。SPMENとBLBSETビットは施錠ビット読み出しの完了で、または3 CPU周期内にLPM命令が実行されないか、または4 CPU周期内にSPM命令が実行されないか、または4 CPU周期内にSPM命令が実行されない場合、自動的に解除(0)されます。SPMENとBLBSETビットが解除(0)されると、LPMは命令一式手引書で記述されるように動作します。

ピット	7	6	5	4	3	2	1	0
Rd	-	-	BLB12	BLB11	BLB02	BLB01	LB2	LB1

ヒュース、下位ビットを読む手順は上記の施錠ビット読み出しと同様です。ヒュース、下位ビットを読み出すにはZポインタに\$0000を設定してSPM CSRのSPMENとBLBSETビットを設定(1)してください。SPMENとBLBSETビットがSPMCSRに設定された後、3 CPU周期内にLPM命令が実行されると、以下で示されるようにヒュース、下位ビット(FLB)の値が転送先レジスタに格納されます。ヒュース、下位ビットの配置と詳細な記述については176頁の表120.を参照してください。

ピット	7	6	5	4	3	2	1	0
Rd	FLB7	FLB6	FLB5	FLB4	FLB3	FLB2	FLB1	FLB0

同様に、ヒュース・上位ビットを読むにはZポインタに\$0003を設定してください。SPMENとBLBSETビットがSPMCSRで設定(1)された後、3周期内にLPM命令が実行されると、以下で示されるようにヒュース・上位ビット(FHB)の値が転送先レジスタに格納されます。ヒュース・上位ビットの配置と詳細な記述については176頁の表119.を参照してください。

ピット	7	6	5	4	3	2	1	0
Rd	FHB7	FHB6	FHB5	FHB4	FHB3	FHB2	FHB1	FHB0

拡張ヒュース、ヒットを読む時はZポーインタに \$0002を設定してください。SPMENとBLBSETビットがSPMCSRで設定(1)された後、3周期内に LPM命令が実行されると、以下で示されるように拡張ヒュース、ヒット(EFB)の値が転送先レシ、スタに格納されます。拡張ヒュース、ヒットの配置 と詳細な記述については176頁の表118.を参照してください。

ピット	7	6	5	4	3	2	1	0
Rd	1	-	-	-	EFB3	EFB2	EFB1	_

プ゚ログラム(<mark>0</mark>)されたヒューズと施錠ビットは<mark>0</mark>として読みます。 非プログラム(1)にされたヒューズと施錠ビットは<mark>1</mark>として読みます。

#### フラッシュ メモリ データ化けの防止

低VCCの期間中、CPUとフラッシュメモリの正しい動作に対して供給電圧が低すぎるためにフラッシュメモリのプログラムが不正にされ得ます。これらの問題はフラッシュメモリを使う基板段階の装置と同じで、同じ設計上の解決策が適用されるべきです。

フラッシュ メモリのプログラム化けは電圧が低すぎる時の2つの状態によって起こされます。1つ目としてフラッシュ メモリへの通常の書き込み手順は正しく動作するための最低電圧が必要です。2つ目として供給電圧が低すぎると、CPU自身が命令を間違って実行し得ます。 フラッシュ メモリ化けは次の推奨設計によって容易に避けられます(1つは必須)。

- そのシステムでブートローダ更新が必要ない場合、どんなブートローダ、ソフトウェア更新をも防ぐためにブートローダ・施錠ビットをプログラム(0)してください。
- 不十分な供給電源電圧の期間中、AVR RESETを活性(Low)に保ってください。これは動作電圧が検出電圧と一致するなら、内部 低電圧検出器(BOD)を許可することによって行えます。そうでなければ外部低VCCリセット保護回路が使えます。書き込み操作進行 中にリセットが起こると、その書き込み動作は供給電源電圧が充分であれば完了されます。
- 低VCCの期間中、AVRコアをパワーダウン休止動作に保ってください。これはCPUが命令の復号と実行を試みるのを防ぎ、SPMCSR 従ってフラッシュ メモリを予期せぬ書き込みから効果的に保護します。

#### SPM命令使用時のフラッシュ メモリ用プログラミング(書き込み)時間

校正された内蔵RC発振器がフラッシュ メモリ アクセス時間に使われます。**表112**.はCPUからのフラッシュ メモリ アクセスに対する代表的なプログラ ミング・時間を示します。

表112. SPM命令によるフラッシュ メモリのプログラミング時間		
項目	最小	最大
SPM命令によるフラッシュ書き込み(ページ消去、ページ書き込み、施錠ビット書き込み)	3.7ms	4.5ms

## アセンブリ言語による簡単なブート ローダ例

このルーチンはRAMからフラッシュメモリへ1~~)がのデータを書きます。RAM内の最初のデータ位置はYレシ、スタによって指示され、フラッシュメモリ内の最初のデータ位置はZレシ、スタによって指示されます。異常処理は含まれません。このルーチン(少なくともSPMJサブルーチン)はブートロータで領域側に配置されなければなりません。NRWW領域側のコートだけが自己プログラミング(~~)が消去と~~)・書き込み)中に読めます。使用レシ、スタはRO,R1,TMP,CNTL,CNTH,SPMCで、レシ、スタの保存と復帰はこのルーチン内に含まれず、使用レシ、スタはコート・量を犠牲にすれば最適化できます。割り込み表がブートロータ、領域に移動されるか、割り込みが禁止されるかのどちらかが前提です。

ページ内データが256バイル以下の場合は計数器上位が不要になります。また関連する命令も変更になります。これらの部分を赤字で示します(訳注:本行は以下のプログラム補正に対応して追加しました)。

ラヘ゛ル	命令		注釈
	. EQU . ORG	PGSZB = PAGESIZE*2 SMALLBOOTSTART	;PGSZBはページ内のバイト数です。(PAGESIZEは語(ワード)数) ;
WRPG:	LDI CALL	SPMC, (1< <pgers)+(1<<spmen) spmj<="" td=""><td>;[ ^゚ージ消去 ] ; ^゚ージ消去SPMCSR値を取得 ; ^゚ージ消去</td></pgers)+(1<<spmen)>	;[ ^゚ージ消去 ] ; ^゚ージ消去SPMCSR値を取得 ; ^゚ージ消去
	LDI CALL	SPMC, (1< <rwwsre)+(1<<spmen) SPMJ</rwwsre)+(1<<spmen) 	;[RWW領域読み出し再許可] ;RWW領域読み出し許可SPMCSR値を取得 ;RWW領域読み出し許可 ;[RAMからフラッシュ ページ 一時緩衝部へ転送]
WLP:	LDI LDI LD LD LDI CALL ADIW SBIW BRNE SUBI SBCI	CNTL, LOW (PGSZB) CNTH, HIGH (PGSZB) RO, Y+ R1, Y+ SPMC, (1< <spmen) (pgszb)="" (pgszb)<="" 2="" cnth:="" cntl,="" high="" low="" spmj="" td="" wlp="" zh,="" zh:="" zl,=""><td>; ハ・ハ・計数器を初期化 ; (削除) ; RAM上の下位データを取得(ポーペタ進行) ; RAM上の上位データを取得(ポーペタ進行) ; ハ・・ン・一時緩衝部書き込みSPMCSR値を取得 ; 対応語(ワード)データをハ・・シ・一時緩衝部に設定 ; ハ・・ン・一時緩衝部ポーペタ進行 ; 計数器を減数 (SUBI) ; 指定ハ・ハ・トン・トトリー ; [ハ・・・シ・書き込み] ; ハ・・シ・一時緩衝部先頭にポーペタを復帰 ; (削除)</td></spmen)>	; ハ・ハ・計数器を初期化 ; (削除) ; RAM上の下位データを取得(ポーペタ進行) ; RAM上の上位データを取得(ポーペタ進行) ; ハ・・ン・一時緩衝部書き込みSPMCSR値を取得 ; 対応語(ワード)データをハ・・シ・一時緩衝部に設定 ; ハ・・ン・一時緩衝部ポーペタ進行 ; 計数器を減数 (SUBI) ; 指定ハ・ハ・トン・トトリー ; [ハ・・・シ・書き込み] ; ハ・・シ・一時緩衝部先頭にポーペタを復帰 ; (削除)
	LDI CALL LDI CALL LDI	SPMC, (1< <pgwrt)+(1<<spmen) (1<<rwwsre)+(1<<spmen)="" (pgszb)<="" cntl,="" low="" spmc,="" spmj="" td=""><td>; フラッシュ メモリ ページ書き込み ; [ RWW領域読み出し再許可 ] ; RWW領域読み出し許可SPMCSR値を取得 ; RWW領域読み出し許可 ; [ 読み戻し照合 (任意) ] ; バイト計数器を初期化</td></pgwrt)+(1<<spmen)>	; フラッシュ メモリ ページ書き込み ; [ RWW領域読み出し再許可 ] ; RWW領域読み出し許可SPMCSR値を取得 ; RWW領域読み出し許可 ; [ 読み戻し照合 (任意) ] ; バイト計数器を初期化
RLP:	LDI SUBI SBCI LPM LD CPSE JMP	CNTH, HIGH (PGSZB) YL, LOW (PGSZB) YH, HIGH (PGSZB) RO, Z+ R1, Y+ RO, R1 ERROR	; (削除) ; RAMデータ先頭にポインタを復帰 ; ; フラッシュ メモリから1パイト取得(ポインタ進行) ; RAMから1パイトデータを取得(ポインタ進行) ; 値一致でスキップ。 ; 不一致で異常処理へ
;	SBIW BRNE	CNTH:CNTL, 1 RLP	;計数器を減数(SUBI) ;指定以外数分继続
RTN:	IN SBRS RET	TMP, SPMCSR TMP, RWWSB	; [ RWW領域へ復帰 ] ; SPM命令制御/状態レジスタ値を取得 ; RWW領域多忙でスキップ ; 準備可で呼び出し元へ復帰
;	LDI CALL RJMP	SPMC, (1< <rwwsre)+(1<<spmen) SPMJ RTN</rwwsre)+(1<<spmen) 	;[ RWW領域読み出し再許可 ] ;RWW領域読み出し許可SPMCSR値を取得 ;RWW領域読み出し許可 ;RWW領域準備可まで待機へ ;[ SPM命令実行サプハーチン ]
SPMJ:	IN SBRC RJMP	TMP, SPMCSR TMP, SPMEN SPMJ	,[ SPM可で美11リアルーラン] ;SPM命令制御/状態レシブスタ値を取得 ;操作可能(直前のSPM完了)でスキップ ;操作可まで待機
; WAIT:	IN CLI SBIC RJMP	TMP, SREG EECR, EEWE WAIT	; ステータス レシ スタ値を保存 ; 全割り込み禁止 ; EEPROM書き込み中以外でスキップ ; EEPROM書き込み完了まで待機
;	OUT SPM OUT RET	SPMCSR, SPMC SREG, TMP	; SPM動作指定 ; 対応SPM動作実行 ; ステータス レシ スタ値を復帰 ; 呼び出し元へ復帰





# ATmega169用プートローダのパラメータ

自己プログラミングの記述で使ったパラメータは表113.~115.で与えられます。

# 表113. 応用領域とブート ローダ領域の分割設定

ſ	BOOTSZ1	BOOTSZ0	ブートローダ領域		アドレス範囲			
ı	БООТЗЕТ	B001320	容量 (語)	ページ数	応用プログラム領域	ブートローダプログラム領域	ブート リセット アドレス	
I	1	1	128	2	\$0000~\$1F7F	\$1F80~\$1FFF	\$1F80	
I	1	0	256	4	\$0000~\$1EFF	\$1F00~\$1FFF	\$1F00	
I	0	1	512	8	\$0000~\$1DFF	\$1E00~\$1FFF	\$1E00	
I	0	0	1024	16	\$0000~\$1BFF	\$1C00~\$1FFF	\$1C00	

注: 各種BOOTSZヒューズ設定は図117.で示されます。

### 表114. RWW領域とNRWW領域の範囲

領域	ページ数	アドレス範囲
書き込み中読み出し可能(RWW)領域	112	\$0000~\$1BFF
書き込み中読み出し不能(NRWW)領域	16	\$1C00~\$1FFF

注: これら2つの領域についての詳細に関しては167頁の「RWW - 書き込み中読み出し可能領域」と「NRWW - 書き込み中読み出し不能領域」をご覧ください。

# 表115 図118で使った各変数説明とフポインタの配置

<u>秋110. 四110</u>	5115. 図118. で使うに各変数説明と2水12%の配直						
変数名	対原	に値 しゅうしゅう					
多 数 石	PC	Zポインタ (注)	意味				
PCMSB	PC12		プログラム カウンタの最上位ビット。(プログラム カウンタは13ビット、PC12~0)				
PAGEMSB	PC5		1~゚ージ内に使われる語(ワード)アドレスの最上位ビット。 (ページ内の64語(ワード)には6ビット PC5~0が必要)				
ZPCMSB		Z13	PCMSBに配置される(対応する)Zレジスタ内のビット。 (Zoが使われないため、ZPCMSB=PCMSB+1)				
ZPAGEMSB		Z6	PAGEMSBに配置される(対応する)Zレジスタ内のビット。 (Zoが使われないため、ZPAGEMSB=PAGEMSB+1)				
PCPAGE	PC12~6	Z13~7	プログラム カウンタ ヘ゜ーシ゛アト・レス:ヘ゜ーシ゛消去とヘ゜ーシ゛書き込み用のヘ゜ーシ゛選択				
PCWORD	PC5~0	Z6~1	プログラム カウンタ 語(ワード)アドレス : 一時緩衝部格納用語(ワード)選択 (ページ書き込み操作中は0でなければなりません。)				

注: Z15~14: 常に無視されます。

Z0:全てのSPM命令に対してOであるべきで、LPM命令に対するバイ選択です。

自己プログラミング中のZポインタの使用についての詳細に関しては170頁の「**自己プログラミング中のフラッシュ メモリのアドレス指定**」をご覧ください。

# メモリ プログラミング

# プログラム メモリとデータ メモリ用施錠ビット

ATmega169は非プログラム(1)のままか、表117.で一覧される付加機能を得るためにプログラム(0)できる6つの施錠ビットを提供します。この施錠ビットはチップ消去指令でのみ1に消去できます。

表116. 施錠ビット バイの内容

名称	ビット番号	意味	既定值 ( <mark>注</mark> )
_	7		1 (非プログラム)
_	6		1 (非プログラム)
BLB12	5	ブートローダ領域に対する保護用ブート施錠ビット	1 (非プログラム)
BLB11	4	/ 「ドローグ 映域に対する保護用/「N地鉄にが	1 (非プログラム)
BLB02	3	  応用領域に対する保護用ブート施錠ビット	1 (非プログラム)
BLB01	2	心用限域に対する休護用/ 『心臓にが	1 (非プログラム)
LB2	1	フラッシュとEEPROMメモリに対する一般保護用施錠ビット	1 (非プログラム)
LB1	0	フラソンユCEEFNOMPでリ(CA) y る   加又体護用地域に ツト	1 (非プログラム)

注: 0はプログラム、1は非プログラムを意味します。

# 表117. 施錠ビットの保護種別

	メモリ施錠ビット		保護種別	
LB 種別	LB2	LB1	直列、並列、またはJTAG経由プログラミングに対する保護	
1	1	1	メモリ施錠機能は機能しません。	
2	1	0	フラッシュ、EEPROMの更なるプログラミング(書き込み)が禁止されます。ヒュース、ヒットが固定されます。 ( <mark>注1</mark> )	
3	0	0	フラッシュ、EEPROMの更なるプログラミング(書き込み)と照合(読み出し)が禁止されます。 ヒュース・ ヒットとブート施錠ビットが固定されます。 ( <mark>注1</mark> )	
LB0 種別	BLB02	BLB01	フラッシュ メモリの応用プログラム領域に対する保護	
1	1	1	LPM, SPM命令が応用領域をアクセスすることに対して制限はありません。	
2	1	0	SPM命令は応用領域に書くことを許されません。	
3	0	0	SPM命令による応用領域への書き込みと、ブートローダ領域でのLPM命令による応用領域からの読み込みが許されません。(注2)	
4	0	1	ブートロータ、領域でのLPM命令による応用領域からの読み込みが許されません。(注2)	
LB1 種別	BLB12	BLB11	フラッシュ メモリのブート ローダ プログラム領域に対する保護	
1	1	1	LPM, SPM命令がブートローダ領域をアクセスすることに対して制限はありません。	
2	1	0	SPM命令はブート ローダ領域に書くことを許されません。	
3	0	0	SPM命令によるブートローダ領域への書き込みと、応用領域でのLPM命令によるブートローダ 領域からの読み込みが許されません。(注3)	
4	0	1	応用領域でのLPM命令によるブートロータ、領域からの読み込みが許されません。(注3)	

注: 0はプログラム、1は非プログラムを意味します。

注1: 施錠ビットを書く前にヒューズビットとブート施錠ビットを書いてください。

注2: BLB02=0で、割り込みベクタがブート ローダ領域に配置されていると、応用領域での実行時に割り込みが禁止されます。

注3: BLB12=0で、割り込みベクタが応用領域に配置されていると、ブートローダ領域での実行時に割り込みが禁止されます。





# ヒュース゛ヒ゛ット

ATmega169には3つのヒュース、ハ・イトがあります。**表118.~120.**は全てのヒュース、の概略機能とヒュース、ハ・イト内でどのように配置されるかを示します。ヒュース、ログラムされると、論理0として読まれることに注意してください。

### 表118. 拡張ヒューズバイト一覧

名称	ビット	意味	既定值
_	7		1 (非プログラム)
_	6		1 (非プログラム)
_	5		1 (非プログラム)
_	4		1 (非プログラム)
BODLEVEL2	3		1 (非プログラム)
BODLEVEL1	2	低電圧検出(BOD)リセットの制御と検出電圧選択。 (注1)	1 (非プログラム)
BODLEVEL0	1		1 (非プログラム)
(予約)	0	注: 本ビットは決してプログラム(0)されるべきではありません。	1 (非プログラム)

注1: BODLEVELヒューズの符号化については26頁の表17.をご覧ください。

# 表119. ヒューズ上位バイト一覧

名称	ピット	意味	既定值
OCDEN ( <b>注4</b> )	7	内蔵デバッグ機能(OCD)許可。	1 (非プログラム) OCD禁止
JTAGEN ( <b>注5</b> )	6	JTAGインターフェース許可。	<mark>0</mark> (プログラム) JTAG許可
SPIEN (注1)	5	直列プログラミング許可。	0 (プログラム) 直列プログラミング許可
WDTON ( <b>注3</b> )	4	ウォッチドッグタイマ常時有効。	1 (非プログラム) WDTはWDTCRで許可
EESAVE	3	チップ消去からEEPROM内容を保護。	1 (非プログラム) EEPROMは未保護
BOOTSZ1	2	ブートローダ 容量選択。(表113.参照) (注2)	<mark>0</mark> (プ <sup>°</sup> ロク <sup>*</sup> ラム)
BOOTSZ0	1	/ 「	<mark>0</mark> (プ <sup>°</sup> ロク <sup>*</sup> ラム)
BOOTRST	0	リセット ベクタ(応用領域またはブート ローダ領域)選択。	1 (非プログラム) 応用領域

注1: SPIENヒュース は直列プログラミングでアクセスできません。

注2: BOOTSZ1,0既定値は最大ブートローダ容量になります。174頁の表113.をご覧ください。

注3: 詳細については28頁の「ウォッチト・ック、タイマ制御レシ、スタ(WDTCR)」をご覧ください。

注4: 施錠ビットやJTAGENヒューズの設定に拘らず、OCDENヒューズがプログラム(0)された製品の出荷は決してありません。プログラム(0)されたOCDENヒューズは全休止形態でクロック系の一部を許可します。これは消費電力の増加になってしまいます。

**注5**: JTAGインターフェースが未接続のままにされる場合、JTAGENヒューズは可能ならば禁止されるべきです。これはJTAGインターフェースのTDOピンの静止時電流を避けるためです。

## 表120. ヒューズ下位バイト一覧

名称	ビット	意味	既定値
CKDIV8 ( <b>注4</b> )	7	システム クロック 8分周選択。	0 (プログラム) 8分周
CKOUT ( <b>注3</b> )	6	システムクロック出力許可。	1 (非プログラム) 不許可
SUT1	5	起動時間選択。 (注1)	1 (非プログラム)
SUT0	4	<b>应</b> 期时间进入。 (注1)	<mark>0</mark> (プ <sup>°</sup> ロク <sup>*</sup> ラム)
CKSEL3	3		<mark>0</mark> (プ <sup>°</sup> ロク <sup>*</sup> ラム)
CKSEL2	2	クロック種別選択。 (注2)	<mark>0</mark> (プ <sup>°</sup> ロク <sup>*</sup> ラム)
CKSEL1	1	/ / □ / / / (里月) / / (/土2 / )	1 (非プログラム)
CKSEL0	0		<mark>0</mark> (プ゚ログ <sup>゛</sup> ラム)

注1: SUT1,0の既定値は最大起動時間になります。詳細については18頁の表9.をご覧ください。

注2: CKSEL3~0の既定設定は8MHz校正付き内蔵RC発振器になります。詳細については17頁の表2をご覧ください。

注3: CKOUTはポートE7に出力することをシステム クロックに許します。詳細については19頁の「クロック出力緩衝部」をご覧ください。

**注4**: 詳細については20頁の「システム クロック前置分周器」をご覧ください。

ヒューズ ビットの状態はチップ消去によって影響されません。施錠ビット1(LB1)がプログラム(0)されると、ヒューズ ビットが固定されることに注意してください。 施錠ビットをプログラム(0)する前にヒューズ ビットをプログラミング(書き込み)してください。

### ヒュース゛のラッチ

ヒューーズ値はデバイスがプログラミング動作へ移行する時にラッチされ、ヒューズ値の変更はデバイスがプログラミング動作を去るまで無効です。これは一旦プログラム(0)されると直ぐに効果があるEESAVEヒューズには適用されません。ヒューズは通常動作での電源投入でもラッチされます。

# 識票バイ

全てのAtmelマイクロ コントローラはデバイスを識別する3バイトの識票符号を持ちます。この符号は直列と並列の両プログラミング動作で、またデバイスが施錠されていても読めます。この3バイトは分離された空間に存在します。ATmega169の識票バイトを右に示します。

① \$000: \$1E 製造業者Atmelを示します。

② \$001:\$94 フラッシュ メモリ容量16Kバイを示します。

③ \$002:\$05 ②値\$94と合せ、ATmega169を示します。

## 校正バイ

ATmega169は内蔵RC発振器用に1バイトの校正値を持ちます。このバイトは識票アドレス空間でアドレス\$0000の上位バイトにあります。リセット中、校正付き内蔵RC発振器の正しい周波数を保証するために、このバイトが発振校正(OSCCAL)レジスタへ自動的に書かれます。

# ページ容量

#### 表121. フラッシュ メモリのヘージ 数とヘージ の語数

全容量	ページ容量	PCWORD	ページ数	PCPAGE	PCMSB
8K語 (16Kバイト)	64語	PC5∼0	128	PC12~6	12

# 表122. EEPROMメモリのページ数とページの語数

全容量	ページ容量	PCWORD	ページ数	PCPAGE	EEAMSB
512バイト	4バイト	EEA1∼0	128	EEA8∼2	8

# 並列プログラミング

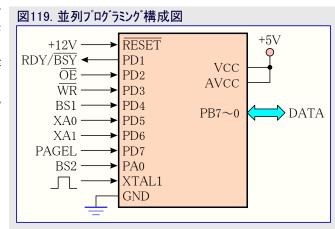
この項はATmega169のプログラム用フラッシュメモリ、データ用EEPROM、メモリの施錠ビット、ヒュース・ビットの並列プログラミングと照合の方法を記述します。特記事項を除いて、パルス幅は最低250nsと仮定されます。

#### 信号名

本章でATmega169のいくつかのピンは並列プログラミング中の機能を表す信号名で参照されます。図119.と表123.をご覧ください。後続の表で示されないピンはピン名によって参照されます。

XA0とXA1ピンはXTAL1ピンが正パルスを与えられる時に実行される動作を決めます。このビット符号化は表125.で示されます。

 $\overline{\text{WR}}$ または $\overline{\text{OE}}$  パルス送出時、設定された指令が決める動作が実行されます。各種指令は表126.で示されます。



#### 表123. 信号名とピン名の関係

信号名	ピン名	入出力	機能
RDY/BSY	PD1	出力	0(Low): 多忙(プログラミング中) 1(High): 準備可(指令受付可)
ŌĒ	PD2	入力	出力許可(負論理)
WR	PD3	入力	書き込みパルス(負論理)
BS1	PD4	入力	上位/下位バイト選択1(0:下位, 1:上位) (一般用)
XA0	PD5	入力	XTAL動作ビット0
XA1	PD6	入力	XTAL動作ビット1
PAGEL	PD7	入力	プログラム メモリとEEPROMデータをヘーシー 一時緩衝部に設定
BS2	PA0	入力	上位/下位バイト選択2 (0:下位, 1:上位) (ヒュース ビット用)
DATA	PB7∼0	入出力	双方向データ バス (OE=Low時出力)





#### 表124. プログラミング動作移行時のピン値

ピン名	シンホ゛ル	値
PAGEL	Prog_enable[3]	0
XA1	Prog_enable[2]	0
XA0	Prog_enable[1]	0
BS1	Prog_enable[0]	0

#### 表125. XA0とXA1の符号化(機能)

XA1	XA0	XTAL1パルス時の動作
0	0	フラッシュまたはEEPROMのアドレス設定(上位/下位はBS1で指定)
0	1	データ設定 (フラッシュの上位/下位はBS1で指定)
1	0	指令設定
1	1	アイドル (動作なし)

#### 表126. 指令バイのビット符号化

指令バイト	指令の機能	指令バイ	指令の機能
\$80 (1000 0000)	チップ消去	\$08 (0000 1000)	識票バイト、校正バイト読み出し
\$40 (0100 0000)	ヒューズビット書き込み	\$04 (0000 0100)	ヒューズビット、施錠ビット読み出し
\$20 (0010 0000)	施錠ビット書き込み	\$02 (0000 0010)	フラッシュ メモリ読み出し
\$10 (0001 0000)	フラッシュ メモリ書き込み	\$03 (0000 0011)	EEPROM読み出し
\$11 (0001 0001)	EEPROM書き込み		

(訳注) 原書本位置の表127.は共通性のため「直列プログラミング」項へ移動しました。

## 並列プログラミング手順

# 並列プログラミング動作への移行

次に示す方法はデバイスを並列プログラミング動作にします。

- ① VCCとGND間に4.5~5.5Vを印加し、最低100µs待ちます。
- ② RESETをLow(0)にし、XTAL1を少なくとも6回、交互に(High/Low)切り替えます。
- ③ 表124.で一覧されるProg enablet°ンを全てLow(0)に設定し、最低100ns待ちます。
- ④ RESETに11.5~12.5Vを印加します。+12VがRESETに印加されてしまった後100ns以内のProg\_enabletプンのどんな動きも、デバイスのプログラミング動作への移行を失敗させます。
- ⑤ 新規指令送出前に少なくとも50µs間待ちます。

## 効率的なプログラミングへの考慮

設定した指令とアドレスはプログラミング中、維持されます。効率的なプログラミングを行うために次を考慮すべきです。

- 複数のメモリ領域を読み書きする時に指令設定は一度だけ必要です。
- チップ消去後のフラッシュ メモリと(EESAVEヒューズがプログラム(0)されている場合を除き)EEPROM全体の内容は\$FFなので、値が\$FFの データ書き込みを飛ばします。
- アトレス上位バイトはフラッシュ メモリで新規256語(ワード)枠、EEPROMで新規256バイト枠の読み書き前に一度だけ必要です。この考慮は 識票バイ・読み出しにも適用されます。

#### チップ消去

チップ消去はフラッシュ メモリ、EEPROM(<mark>注1</mark>)、施錠ビットを消去します。施錠ビットはプログラム メモリが完全に消去されてしまうまでリセット(消去) されません。ヒューズ ビットは変更されません。チップ消去はフラッシュ メモリやEEPROMが再書き込みされる前に実行されなければなりません。

注1: EESAVEヒューズがプログラム(0)されていると、EEPROMはチップ消去中、保護されます。

「チップ消去」指令設定

- (1) XA1をHigh(1)、XA0をLow(0)に設定します。これは指令設定を許可します。
- ② BS1をLow(0)に設定します。
- ③ DATAを\$80(1000 0000)に設定します。これはチップ消去指令です。
- ④ XTAL1に正パルスを与えます。これはチップ消去指令を設定します。
- ⑤ WRに負パルスを与えます。これはチップ消去を開始します。RDY/BSYがLow(0)になります。
- (6) 次の指令を設定する前にRDY/BSYがHigh(1)になるまで待ちます。

# フラッシュ メモリ書き込み (図121.タイミングを参照)

フラッシュ メモリはヘージで構成されます(177頁の**表121**.参照)。フラッシュ メモリヘ書く時にプログラム データはヘージ緩衝部にラッチされます。これは同時に書かれることをプログラム データの1ヘージに許します。次の手順は完全なフラッシュ メモリの書き込み方法を記述します。

### A. 「フラッシュ メモリ書き込み」指令設定

- ① XA1をHigh(1)、XA0をLow(0)に設定します。これは指令設定を許可します。
- **②** BS1をLow(0)に設定します。
- ③ DATAを\$10(0001 0000)に設定します。これはフラッシュ メモリ書き込み指令です。
- ④ XTAL1に正パルスを与えます。これはフラッシュ メモリ書き込み指令を設定します。

#### B. アトレス下位バイト設定

- ① XA1をLow(0)、XA0をLow(0)に設定します。これはアドレス設定を許可します。
- ② BS1をLow(0)に設定します。これは下位アドレス(バイト)を選びます。
- **③** DATAにアドレス下位ハイ(\$00~\$FF)を設定します。
- ④ XTAL1に正パルスを与えます。これはアドレス下位バイを設定します。

#### C. データ下位バイト設定

- ① XA1をLow(0)、XA0をHigh(1)に設定します。これはデータ設定を許可します。
- ② DATAにデータ下位ハイト(\$00~\$FF)を設定します。
- ③ XTAL1に正パルスを与えます。これはデータ下位バイを設定します。

#### D. データ上位バイト設定

- ① BS1をHigh(1)に設定します。これは上位バイを選びます。
- ② XA1をLow(0)、XA0をHigh(1)に設定します。これはデータ設定を許可します。
- ③ DATAにデータ上位ハイト(\$00~\$FF)を設定します。
- ④ XTAL1に正パルスを与えます。これはデータ上位バイを設定します。

### E. 語(ワート)データをヘーン一時緩衝部に設定

- ① BS1をHigh(1)にします。これは上位バイを選びます。
- ② PAGELに正パルスを与えます。これは語データをページー時緩衝部にラッチ(設定)します。
- F. 緩衝部全体が満たされるか、または(必要な)全てのデータが緩衝部内に設定されるまで、B~Eを繰り返し

アト・レス内の下位ビットがページ内の語(ワード)位置を指示する一方、上位ビットがフラッシュ メモリ内のページをアドレス指定します。これは**図120**. で図示されます。 ページ内の語アドレスに8ビット未満が必要とされる場合(ページ容量<256)、アドレス下位バイトの最上位(側)ビットがページ書 き込み実行時のページ アドレスに使われることに注意してください。

#### G. アトレス上位バイ設定

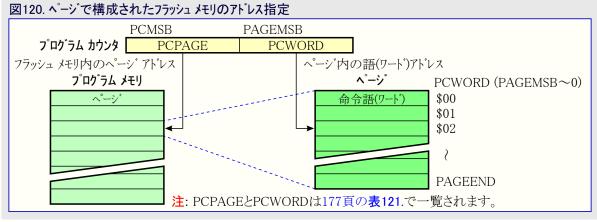
- ① XA1をLow(0)、XA0をLow(0)に設定します。これはアドレス設定を許可します。
- ② BS1をHigh(1)に設定します。これは上位アドレス(バイト)を選びます。
- ③ DATAにアドレス上位ハイト(\$00~\$1F)を設定します。
- ④ XTAL1に正パルスを与えます。これはアドレス上位バイトを設定します。

# H. ページ書き込み

- ① BS1をLow(0)に設定します。
- ② WRに負パルスを与えます。これはデータのページ全体の書き込みを開始します。RDY/BSYがLow(0)になります。
- ③ RDY/BSYがHigh(1)になるまで待ちます。
- I. フラッシュ メモリ全部が書かれるか、または(必要な)全データが書かれてしまうまで、B~Hを繰り返し

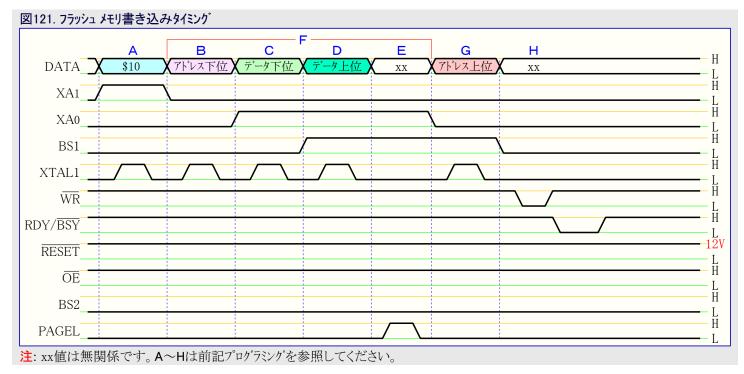
#### J. ページ書き込み終了

- ① XA1をHigh(1)、XA0をLow(0)に設定します。これは指令設定を許可します。
- ② DATAを\$00(0000 0000)にします。これは無操作指令です。
- ③ XTAL1に正パルスを与えます。これは無操作指令を設定し、内部書き込み信号がリセットされます。





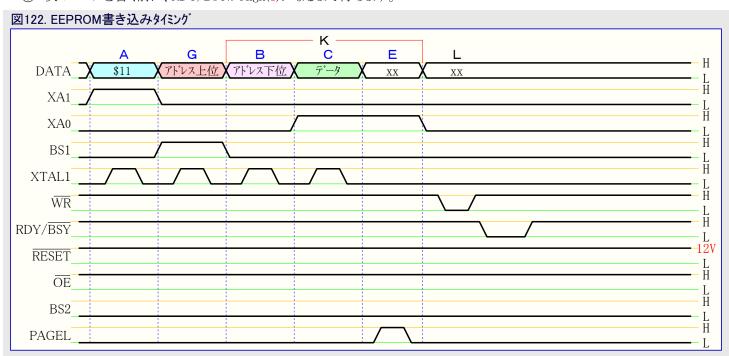




## EEPROM書き込み

EEPROMはヘージで構成されます(177頁の表122.参照)。EEPROMを書く時にデータはヘージ緩衝部にラッチされます。これは同時に書かれることをデータの1ヘージに許します。データ用EEPROMメモリの書き込み方法は次のとおりです。(指令、アトレス、データ設定の詳細については179頁の「フラッシュ メモリの書き込み」を参照。図122.タイミング参照。)

- 1. EEPROM書き込み指令\$11(0001 0001)を設定します。(「フラッシュ メモリ書き込み」のAを参照)
- 2. アドレス上位ハーイト(\$00~\$01)を設定します。(「フラッシュ メモリ書き込み」のGを参照)
- 3. アドレス下位ハイト(\$00~\$FF)を設定します。(「フラッシュ メモリ書き込み」のBを参照)
- **4.** データ バイト(\$00~\$FF)を設定します。(「**フラッシュ メモリ書き込み**」の**C**を参照)
- 5. データをラッチします(PAGELに正パルスを与えます)。(「フラッシュメモリ書き込み」のEを参照)
- K. 緩衝部全体が満たされるまで3~5を繰り返します。
- L. EEPROMページ書き込み
  - ① BS1をLow(0)に設定します。
  - ② WRに負パルスを与えます。これはEEPROMページ書き込みを開始します。RDY/BSYがLow(0)になります。
  - ③ 次のページを書く前に、RDY/BSYがHigh(1)になるまで待ちます。



## フラッシュ メモリ読み出し

フラッシュ メモリの読み出し方法は次のとおりです。(指令とアトンス設定の詳細については179頁の「**フラッシュ メモリの書き込み**」を参照)

- 1. フラッシュ メモリ読み出し指令\$02(0000 0010)を設定します。(「フラッシュ メモリ書き込み」のAを参照)
- 2. アドレス上位ハーイ(\$00~\$1F)を設定します。(「フラッシュ メモリ書き込み」のGを参照)
- 3. アドレス下位ハーイト(\$00~\$FF)を設定します。(「フラッシュ メモリ書き込み」のBを参照)
- 4. BS1をLow(0)、OEをLow(0)に設定します。フラッシュ メモリ語(ワート)の下位バイが直ぐにDATAで読めます。
- 5. BS1をHigh(1)に設定します。フラッシュメモリ語(ワート)の上位バイトが直ぐにDATAで読めます。
- 6. OEをHigh(1)に設定します。DATAはHi-Zになります。

#### EEPROM読み出し

データ用EEPROMの読み出し方法は次のとおりです。(指令とアドレス設定の詳細については179頁の「フラッシュメモリの書き込み」を参照)

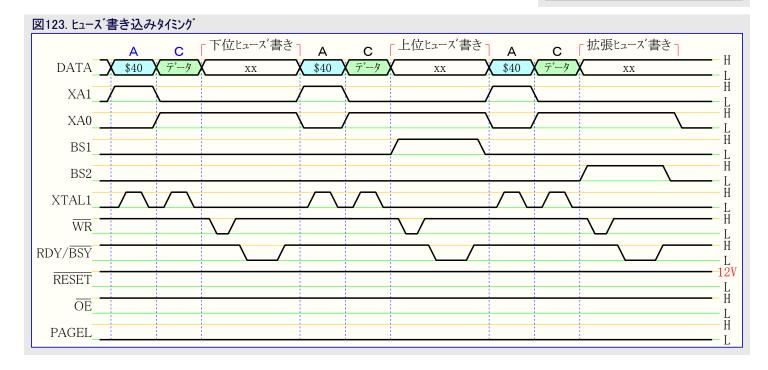
- 1. EEPROM読み出し指令\$03(0000 0011)を設定します。(「フラッシュ メモリ書き込み」のAを参照)
- 2. アドレス上位ハイト(\$00~\$01)を設定します。(「フラッシュ メモリ書き込み」のGを参照)
- 3. アドレス下位バイト(\$00~\$FF)を設定します。(「フラッシュ メモリ書き込み」のBを参照)
- 4. BS1をLow(0)、OEをLow(0)に設定します。EEPROMのハイトデータが直ぐにDATAで読めます。
- 5. OEをHigh(1)に設定します。DATAはHi-Zになります。

#### **ヒューズビット書き込み**(訳注:原書での拡張/上位/下位ヒューズ項を1つに纏めました。)

各ヒューズ、バ・仆の書き込み方法は次のとおりです。(指令とデータ設定の詳細については179頁の「フラッシュ メモリの書き込み」を参照)

- 1. ヒュース゚ ビット書き込み指令\$40(0100 0000)を設定します。(「**フラッシュ メモリ書き込み**」の**A**を参照)
- 2. データ下位バイトを設定します。0=プログラム,1=非プログラム(消去)です。(「フラッシュ メモリ書き込み」のCを参照)
- 3. BS1とBS2を右表の目的バイトに対応する設定にします。
- 4. WRに負パルスを与え、RDY/BSYがHighになるまで待ちます。
- 5. 3.でHighに設定したBS1、BS2をLow(0)に戻します。これはデータ下位バイを選びます。

表A. ヒューズバイト対応BS1,BS2設定						
ヒュース゛ハ゛イト	BS1	BS2				
拡張バイ	Low(0)	High(1)				
上位バイ	High(1)	Low(0)				
下位バイ	Low(0)	Low(0)				





#### 施錠ビット書き込み

施錠ビットの書き込み方法は次のとおりです。(指令とデータ設定の詳細については179頁の「フラッシュ メモリの書き込み」を参照)

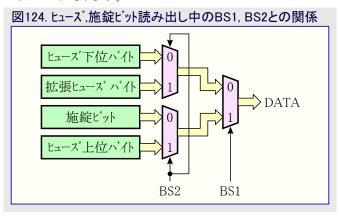
- 1. 施錠ビット書き込み指令\$20(0010 0000)を設定します。(「フラッシュ メモリの書き込み」のAを参照)
- 2. データ下位ハイト(としてデータ)を設定します。 0=プログラム,1=無変化です。LB保護種別3が設定(LB1とLB2がプログラム(0))されると、どの外部的なプログラミング動作種別によってもブート施錠ビットはプログラミングできません。 (「フラッシュ メモリの書き込み」のCを参照)
- 3. WRに負パルスを与え、RDY/BSYがHighになるまで待ちます。

施錠ビットはチップ消去の実行によってのみ解除(1)できます。

## ヒューズビットと施錠ビットの読み出し

ヒュース、ビットと施錠ビットの読み出し方法は次のとおりです。(指令設定の詳細については179頁の「フラッシュ メモリの書き込み」を参照)

- 1. ヒュース ビットと施錠ビットの読み出し指令\$04(0000 0100)を設定します。(「フラッシュ メモリの書き込み」のAを参照)
- 2. BS1とBS2をLow(0)、OEをLow(0)に設定します。ヒュース・下位ビットの状態が直ぐにDATAで読めます。(0=プログラム)
- 3. BS1とBS2をHigh(1)、OEをLow(0)に設定します。ヒュース、上位ビットの状態が直ぐにDATAで読めます。(0=プログラム)
- **4.** BS1をLow(0)、BS2をHigh(1)、OEをLow(0)に設定します。拡張ヒュース・ヒットの状態が直ぐにDATAで読めます。(0=プログラム)
- 5. BS1をhigh(1)、BS2をLow(0)、OEをLow(0)に設定します。施錠ビットの状態が直ぐにDATAで読めます。(0=プログラム)
- 6. OEをHigh(1)に設定します。DATAはHi-Zになります。



## 識票バイ読み出し

識票バイトの読み出し方法は次のとおりです。(指令とアドレス設定の詳細については179頁の「**フラッシュ メモリの書き込み**」を参照)

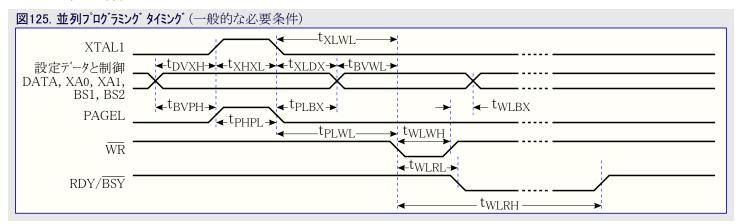
- 1. 識票バイト読み出し指令\$08(0000 1000)を設定します。(「フラッシュ メモリの書き込み |のAを参照)
- 2. アドレス下位バイト(\$00~\$02)を設定します。(「フラッシュ メモリの書き込み」のBを参照)
- **3.** BS1をLow(0)、 OEをLow(0)に設定します。 選んだ識票へいが直ぐにDATAで読めます。
- 4. OEをHigh(1)に設定します。DATAはHi-Zになります。

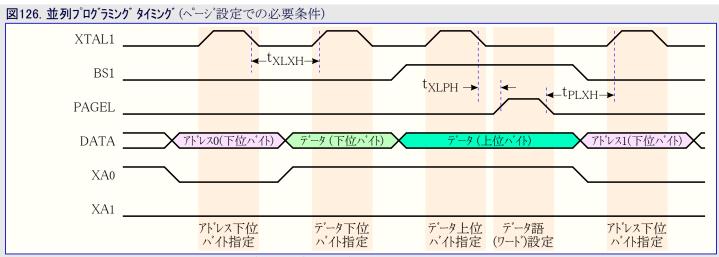
## 校正バイ読み出し

校正バイの読み出し方法は次のとおりです。(指令とアドレス設定の詳細については179頁の「フラッシュ メモリの書き込み」を参照)

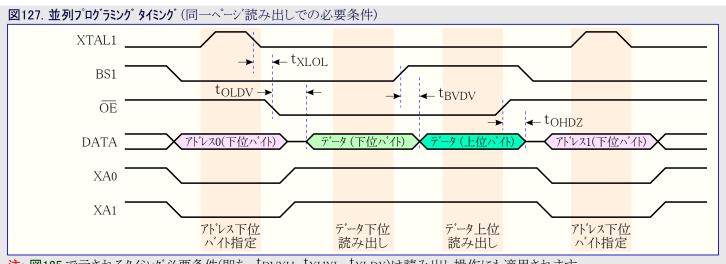
- 1. 校正バイト読み出し指令\$08(0000 1000)を設定します。(「フラッシュ メモリの書き込み |のAを参照)
- 2. アドレス下位バイトに\$00を設定します。(「フラッシュ メモリの書き込み」のBを参照)
- 3. BS1をHigh(1)、OEをLow(0)に設定します。校正バイが直ぐにDATAで読めます。
- 4. OEをHigh(1)に設定します。DATAはHi-Zになります。

#### 並列プログラミング特性





注: 図125.で示されるタイミング必要条件(即ち、<sup>t</sup>DVXH、<sup>t</sup>XHXL、<sup>t</sup>XLDX)は設定操作にも適用されます。



注: 図125.で示されるタイミング必要条件(即ち、<sup>t</sup>DVXH、<sup>t</sup>XHXL、<sup>t</sup>XLDX)は読み出し操作にも適用されます。





表128. 並列プログラミング特性 (VCC=5V±10%)

シンホ゛ル	項目	最小	代表	最大	単位
Vpp	プログラミング許可電圧	11.5		12.5	V
Ipp	プログラミング許可電流			250	μA
t <sub>DVXH</sub>	XTAL1↑に対するデータと制御の準備時間	67			
t <sub>XLXH</sub>	XTAL1↓から次XTAL1↑までの待機時間	200			
txHXL	XTAL1 Highハ゜ルス幅	150			
txldx	XTAL1パルス↓後のデータと制御の保持時間	67			
t <sub>XLWL</sub>	XTAL1パルス↓後のWR↓待機時間	0			
txlph	XTAL1パルス↓後のPAGELパルス↑待機時間	0			
t <sub>PLXH</sub>	PAGELパルス↓後のXTAL1パルス↑待機時間	150			n.,
t <sub>BVPH</sub>	PAGELパルス↑に対するBS1準備時間	67			ns
tPHPL	PAGEL Highパルス幅	150			
t <sub>PLBX</sub>	PAGELパルス↓後のBS1保持時間	67			
t <sub>WLBX</sub>	WRパルス↓後のBS1,BS2保持時間	67			
t <sub>PLWL</sub>	PAGELパルス↓後のWRパルス↓待機時間	67			
t <sub>BVWL</sub>	WRパルス↓に対するBS1準備時間	67			
twlwh	WR Lowパルス幅	150			
twlrl	WRパルス↓後のRDY/BSY↓遅延時間	0		1	μs
twlrh	書き込み時間(WR↓からRDY/BSY↑) (注1)	3.7		4.5	m a
twlrh_ce	チップ。消去時間 (WR↓からRDY/BSY↑) (注2)	7.5		9	ms
t <sub>XLCL</sub>	XTAL1パルス↓後のOE↓待機時間	0			
tBVDV	BS1有効からのDATA遅延時間	0		250	no
toldv	OE ↓後のDATA出力遅延時間			250	ns
toHDZ	OE↑後のDATAフローティング遅延時間			250	

注1: フラッシュ メモリ、EEPROM、ヒュース゛ビット、施錠ビット書き込み指令に対して有効です。

注2: チップ消去指令に対して有効です。

## 直列プログラミング

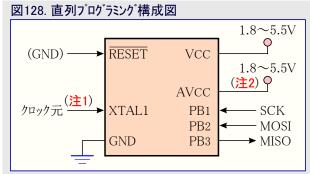
フラッシュ メモリとEEPROMの両方はRESETがGNDに引かれている間に直列SPIハ、スを使ってプログラミングできます。この直列インターフェースはSCK入力、MOSI入力、MISO出力から成ります。RESETがLowに設定された後、書き込み/消去操作が実行され得るのに先立ってプログラミング許可命令が初めに実行されることを必要とします。注意、表127.でSPIプログラミング用のピン配置が一覧されます。全てのデハイスが内部SPIインターフェースに対する専用SPIピンを使うとは限りません。

## 直列プログラミング用ピン配置

信号名	ピン名	入出力	機能
SCK	PB1	入力	直列クロック
MOSI	PB2	入力	直列データ入力
MISO	PB3 出力 直列		直列データ出力

EEPROMをプログラミングする時に自動消去周期が自動書き込み動作内に組み入れられ(直列プログラミングのみ)、チップ消去命令を初めに実行する必要がありません。チップ消去操作はプログラム(フラッシュ メモリ)とEEPROM両方の全てのメモリ位置の内容を\$FFにします。

CKSELヒュース による有効なクロックが与えられなければなりません。直列クロック(SCK)入力のLowとHighの区間の最小値は次のように定義されます。



注1: デバイスが内蔵発振器で動作する場合、XTAL1 ピンにクロック元を接続する必要はありません。

**注2**: VCC-0.3V〈AVCC〈VCC+0.3Vですが、AVCCは常に1.8~5.5V内にすべきです。

 $f_{\rm CK} < 12$ MHz: Low区間 > 2 CPUクロック周期  $f_{\rm CK} < 12$ MHz: High区間 > 2 CPUクロック周期  $f_{\rm CK} \ge 12$ MHz: Low区間 > 3 CPUクロック周期  $f_{\rm CK} \ge 12$ MHz: High区間 > 3 CPUクロック周期

## 直列プログラミング手順

ATmega169に直列データを書く時にデータはSCKの上昇端で行われ、ATmega169から読む時にデータはSCKの下降端で行われます。タイシグの詳細については図129.をご覧ください。

直列プログラミング動作でのATmega169のプログラミングと照合は次の手順が推奨されます(表130.の命令一式参照)。

1. 電源投入手順:

RESETとSCKがLow(0)に設定されている間にVCCとGND間へ電源を印加します。いくつかのシステムで電源投入中、SCKがLowに保持されるのを書き込み器が保証できません。この場合、SCKがLow(0)に設定されてしまった後、RESETは最低2 CPUクロック周期幅の正パルスを与えられなければなりません。

- 2. 最低20ms待ち、MOSIピンにプログラミング許可命令を送ることによって直列プログラミングを許可してください。
- 3. 通信が同期を外していると、直列プログラミング命令は動作しません。同期していると、プログラミング許可命令の第3小小送出時に第2 ハイト(\$53)が戻ります。この戻りが正しいかどうかによらず、命令の4ハイ全てが送信されなければなりません。\$53が戻らない場合、RESETに正パルスを与え、新規プログラミング許可命令を行ってください。
- 4. フラッシュ メモリは1ペーシ、単位で書かれます。ページ、容量は177頁の表121.で得られます。このメモリページ、はページ、設定命令と共にアドレスの下位6+1ビットとデータを供給することによって1パイトづつ設定されます。ページが正しく設定されるのを保証するため、与えられたアドレスにデータ上位パイトが適用される前にデータ下位パイトが設定されなければなりません。プログラム メモリのページ はアドレスの上位7ビットを含むページ書き込み命令の設定によって(フラッシュ メモリに)格納されます。ポーリング(BSY/RDY)が使われない場合、使用者は次のページを行う前に最低twD\_FLASH(表129.参照)待たなければなりません。フラッシュ書き込み操作完了前の(ポーリング以外の)直列プログラミング インターフェースでのアクセスは不正な書き込み結果になり得ます。
- 5. EEPROMはペーシー単位かい、
  小単位のどちらかでプログラミングできます。

**バイト単位**: EEPROMは適切なEEPROM書き込み命令と共にアドレスとデータを供給することによって1バイト単位で書かれます。EEP ROMのメモリ位置は新規データが書かれるのに先立って始めで自動的に消去されます。ポーリング(BSY/RDY)が使われない場合、次のバイトを行う前に最低twp eeprom(表129.参照)待たなければなりません。

**ページ単位**: EEPROMの1ページはEEPROMページ設定命令と共にアドレスの下位2ビットとデータを供給することによって1バイトづつ設定されます。EEPROMページはアドレスの上位7ビットを含むEEPROMページ書き込み命令によって(EEPROMに)格納されます。EEPROMページ アクセス使用時、EEPROMページ 設定命令で設定したバイト位置だけが変更されます。残りの位置は無変化で留まります。ポーリング(BSY/RDY)が使われない場合、次ページ(表122.参照)を行う前に最低twd\_EEPROM(表129.参照)待たなければなりません。チップ消去されたデバイスでの\$FFデータは書かれる必要がありません。

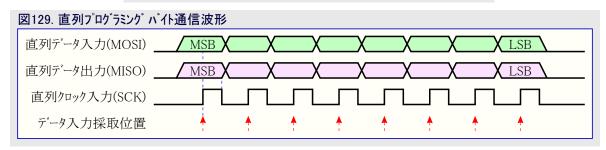
- 6. どのメモリ位置も選んだアドレスの内容を直列出力MISOに読み戻す読み出し命令を使うことによって照合できます。
- 7. プログラミング作業終了時、RESETは通常動作を開始するため、High(1)に設定できます。
- 8. 電源OFF手順(必要とされるならば)
  - ・ RESET を High(1)に 設定します。
  - VCC電源をOFFにします。





表129. ヒューズ, フラッシュ, EEPROM次位置書き込み前の待機時間

シンホ゛ル	最低待機時間	備考					
two_fuse	4.5ms	ヒュース゛書き込み					
two_flash	4.5ms	フラッシュ メモリ書き込み					
twd_eeprom	9.0ms	EEPROM書き込み					
two_erase	9.0ms	チップ消去					



## 表130. 直列プログラミング命令一式

命令		命令	形式	≨h //⊏			
叩巾	第1バイト	第2バイト	第3バイト	第4バイト	動作		
プログラミング許可	1010 1100	0101 0011	XXXX XXXX	XXXX XXXX	RESET=Low後のプログラミン	グを許可します。	
チップ消去	1010 1100	100x xxxx	XXXX XXXX	XXXX XXXX	フラッシュ メモリ,EEPROM,施釒	淀ビットを消去します。	
フラッシュメモリ読み出し	0010 P000	000Н НННН	LLLL LLLL	RRRR RRRR	アドレスH:LのP(H/L)バイトを	読み出します。	
フラッシュ ヘ゜ーシ゛設定	0100 P000	0000 xxxx	xxLL LLLL	WWWW WWWW	緩衝部アドレス <mark>L</mark> のP(H/L)バ	仆に書き込みます。	
フラッシュページ書き込み	0100 1100	000Н НННН	LLxx xxxx	XXXX XXXX	アトレスH:Lのペーシ、に書き込	しみます。	
EEPROM読み出し	1010 0000	000x xxxH	LLLL LLLL	RRRR RRRR	アドレスH:Lのバイトを読み出	します。	
EEPROMバイト書き込み	1100 0000	000x xxxH	LLLL LLLL	WWWW WWWW	アドレスH:Lのバイトに書き込みます。		
EEPROMへ゜ージ設定	1100 0001	0000 0000	0000 00LL	WWWW WWWW	緩衝部アドレスLのバイトに書き込みます。		
EEPROMへ゜ージ書き込み	1100 0010	00xx xxxH	LLLL LL00	XXXX XXXX	アドレスH:Lのページに書き込みます。		
施錠ビット読み出し	0101 1000	0000 0000	XXXX XXXX	xxRR RRRR		を読み出します。	
施錠ビット書き込み	1010 1100	111x xxxx	XXXX XXXX	11WW WWWW	(175頁の表116.参照)	に書き込みます。	
ヒューズ下位読み出し	0101 0000	0000 0000	XXXX XXXX	RRRR RRRR	ヒュース・下位ビット	を読み出します。	
ヒュース・下位書き込み	1010 1100	1010 0000	XXXX XXXX	WWWW WWWW	(176頁の表120.参照)	に書き込みます。	
ヒューズ上位読み出し	0101 1000	0000 1000	XXXX XXXX	RRRR RRRR	ヒュース・上位ビット	を読み出します。	
ヒューズ上位書き込み	1010 1100	1010 1000	XXXX XXXX	WWWW WWWW	(176頁の <b>表119</b> .参照)	に書き込みます。	
拡張ヒューズ読み出し	0101 0000	0000 1000	XXXX XXXX		拡張ヒューズビット	を読み出します。	
拡張ヒューズ書き込み	1010 1100	1010 0100	XXXX XXXX		(176頁の表118.参照)	に書き込みます。	
識票バイト読み出し	0011 0000	000x xxxx	xxxx xxLL	RRRR RRRR	アドレスLの識票バイトを読み出します。		
校正バ小読み出し	0011 1000	000x xxxx	0000 0000	RRRR RRRR	校正バイを読み出します。		
多忙/準備可検査	1111 0000	0000 0000	XXXX XXXX	xxxx xxxR	R=1で多忙、他の操作前に	10まで待機します。	

H=アドレス上位バイトのビット L=アドレス下位バイトのビット P=0=下位バイト、1=上位バイト

R = 読み出しデータ (MCU出力) W = 書き込みデータ (MCU入力) x = 0または1 (無意味/不定)

## 直列プログラミング特性

SPI部の特性については196頁の「SPIタイミング特性」を参照してください。

## JTAGインターフェース経由のプログラミング

JTAGインターフェースを通すプログラミングは4つのJTAG仕様ピン、TCK,TMS,TDI,TDOの制御が必要です。RESETピンとクロック ピンの制御は必要とされません。

JTAGインターフェースが使われ得るにはJTAGENヒュース、がプログラム(0)されなければなりません。本デハイスは既定でこのヒュース、がプログラム(0)されて出荷されます。更にMCU制御レジ、スタ(MCUCR)のJTAGインターフェース禁止(JTD)ビットが解除(0)されなければなりません。JTD ビットが設定(1)の場合、代わりに外部リセットを強制的なLowにできます。その後、2チップ(CPU)クロック後にJTDビットが解除(0)され、JTAG ピンはプログラミングに利用できます。これはJTAGインターフェース経由実装書き込みを許すにも拘らず、一方、実行動作で標準ポート ピンとしてJTAGピンを使う手段を提供します。境界走査(Boundary-Scan)や内蔵デハ、ッグ機能にJTAGピンを使う時にこの手法が使えないことに注意してください。これらの場合のJTAGピンはその目的専用にされなければなりません。

プログラミング・中、TCK入力のクロック周波数はチップの最高周波数よりも低くなければなりません。システム クロック前置分周器はTCKクロック入力を充分に低い周波数へ分周するのには使えません。

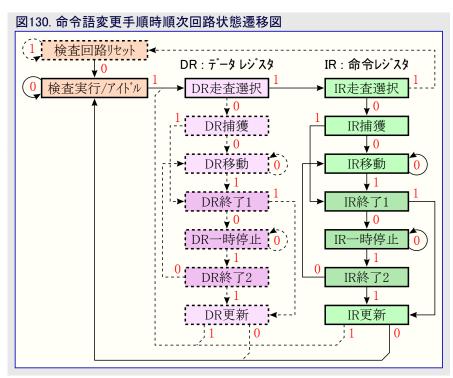
この文書内の定義では全ての移動レジスタについて入出力ともLSBが最初に移動されます。

#### プログラミング特殊JTAG命令

命令レジスタ(IR)は4ビット長で、16種類までの命令を支援します。以下に示される一覧はプログラミングに有用なJTAG命令です。

各命令の命令コートは命令名後の16進形式で示されます。本文は各命令についてTDIとTDO間の経路として選ばれるデータレシブスタを記述します。

検査入出力ポート(TAP)制御器の検査実行/アイドル 状態は内部クロックの発生に使われます。また、JTA G手順間のアイドル状態としても使えます。命令を切 り替えるための順次回路の順番は図130.で示され ます。



#### AVR\_RESET (\$C)

AVRデバイスをリセット状態へ、またはリセット状態から開放するためのAVR固有公開JTAG命令です。この命令によって検査入出力ポート (TAP)制御器はリセットされません。1ビットのリセットレシ、スタがデータレシ、スタ(DR)として選ばれます。 リセット チェーンに論理1がある限り、 リセットが 有効であることに注意してください。 このチェーンからの出力はラッチされません。 本命令が有効な状態を次に示します。

• DR移動: リセットレジスタがTCKクロック入力によって移されます。

## PROG\_ENABLE (\$4)

JTAGポート経由のプログラミングを許可するためのAVR固有公開JTAG命令です。16ビット長のプログラミング許可レジスタがデータ レジスタ(DR) として選ばれます。本命令が有効な状態を次に示します。

• DR移動: プログラミング許可識別子がデータレジスタに移動入力されます。

• DR更新: プログラミング許可識別子が正規値と比較され、識別子が有効なら、プログラミング動作に移行されます。

#### PROG COMMANDS (\$5)

JTAGポート経由のプログラミング命令移行用のAVR固有公開JTAG命令です。15ビット長のプログラミング命令レジスタがデータ レジスタ(DR)として選ばれます。本命令が有効な状態を次に示します。

• DR捕獲: 直前の命令の結果がデータレジスタに設定されます。

• DR移動: データレジスタがTCK入力により移され、直前の命令の結果を移動出力し、新規命令を移動入力します。

• **DR更新** : プログラミング命令がフラッシュ メモリ入力に適用されます。

• 検査実行/**アイドル**:1つのクロック周期が生成され、適用された命令を実行します(常に必要とされる訳でなく、以降の表131.をご覧ください)。





## PROG\_PAGELOAD (\$6)

JTAGポート経由でフラッシュメモリのページデータを直接設定するためのAVR固有公開JTAG命令です。8ビット フラッシュ バイト データ レジスタがデータ レジスタ(DR)として選ばれます。これは物理的にプログラミング命令レジスタの下位8ビットです。活性(有効)状態は以下です。

• **DR移動** : フラッシュ バイト データ レシ スタが TCK クロック 入力によって移動されます。

• DR更新 : フラッシュ バイト データ レジスタの内容が一時レジスタに複写されます。11 TCK周期で開始される書き込み(設定)手

順が一時レジスタの内容をフラッシュページ緩衝部に設定します。AVRはPROG\_PAGELOAD命令移行後出会った最初のDR更新に対して下位バイで始め、新規DR更新状態毎に上位/下位バイト書き込み間を自動的に切り替えます。プログラムカウンタは最初の書き込みバイトを除き、下位バイト書き込み前に予め増加(+1)されます。これは最初のデータがPROG\_COMMANDSによって設定したアドレスに書かれ、ページ緩衝部の最終位置設定がプログラ

ムカウンタを次ペーシーへ増加しないことを保証します。

#### PROG\_PAGEREAD (\$7)

JTAGポート経由でフラッシュメモリ内容を直接捕獲するためのAVR固有公開JTAG命令です。8ビット フラッシュバイト データ レジスタがデータ レジスタ(DR)として選ばれます。これは物理的にプログラミング命令レジスタの下位8ビットです。活性(有効)状態は以下です。

• DR捕獲 : 選んだフラッシュ メモリ ハー・イトの内容がフラッシュ ハー・イト データ レシ スタに捕獲されます。 AVRはPROG\_PAGEREAD命令

移行後に出会った最初のDR捕獲に対して下位バイヤで始め、新規DR捕獲状態毎に上位/下位バイ・読み込み間を自動的に切り替えます。プログラムカウンタは最初の読み込みバイトを含め、各上位バイト読み込み後に増加(+1)されます。これは最初のデータがPROG\_COMMANDSによって設定した先頭アドレスから捕獲され、ページの最

終位置読み込みがプログラム カウンタを次ペーシ へ増加(進行)することを保証します。

• DR移動: フラッシュ バイト データ レジスタがTCKクロック入力によって移動されます。

## データ レジスタ

データ レジスタ(DR)は187頁の「プログラミング特殊JTAG命令」項で記載されたJTAG命令レジスタ(IR)によって選ばれます。プログラミング操作に関連するデータ レジスタを次に示します。

リセット(Reset)レシ、スタ

- プログラミング許可(Programing Enable)レジスタ
- プログラミング命令(Programing Command)レジスタ
- フラッシュ バイト データ(Flash Data Byte)レジスタ

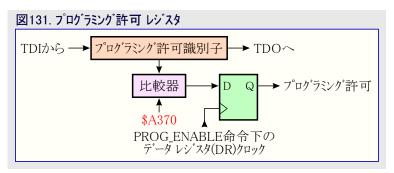
#### リセット (Reset) レシ スタ

リセットレジスタはプログラミング中、デバイスをリセットするのに使われる検査データレジスタです。プログラミング動作への移行に先立ってデバイスを リセットするのに必要とされます。

リセット レシ、スタ内の値1は外部RESETをLowに引き込むことに相当します。リセット レシ、スタ内の値が1である限り、デバイスはリセットされます。 リセット レシ、スタの開放後、デバイスはクロック種別ヒューズ。設定に従ってリセット起動遅延時間(17頁の「クロック元」参照)分リセットを維持します。 157頁の図108.で示されるようにこのデータ レシ、スタからの出力はラッチされず、それ故リセットが直ちに起こります。

# プ<sup>°</sup>ロク<sup>\*</sup>ラミンク<sup>\*</sup>許可 (Programing Enable) レシ<sup>\*</sup>スタ

プログラミング許可レジスタは16ビット長のレジスタです。このレジスタの内容はプログラミング許可識別子\$A370(1010 0011 0111 0000)と比較されます。このレジスタの内容がプログラミング許可識別子と一致すると、JTAGポート経由のプログラミングが許可されます。このレジスタは電源ONリセットで0にリセットされ、プログラミング動作を抜ける時には常にリセットされるべきです。



#### プログラミング 命令 (Programing Command) レジスタ

プログラミング命令レジスタは15ビット長のレジスタです。このレジスタは プログラミング命令を連続的に移動入力し、直前のプログラミング命 令の結果を連続的に移動出力するのに使われます。JTAGプログラミング命令一式は表131.で示されます。プログラミング命令を 移動入力する時の状態順は図133.で図解されます。



表131.	JTAG7	゚ロゲラ	ミングで	合合一	式(	手順)
201.	01/14/	-, ,	<b></b> -	13 13		( J //US/

		命令	TDI入力	TDO出力	備考
£			010 0011 1000 0000	XXX XXXX XXXX XXXX	
ァー	1 ①	<b>チつ°沙生士</b>	011 0001 1000 0000	XXX XXXX XXXX XXXX	
ップ	1①	チップ消去	011 0011 1000 0000	XXX XXXX XXXX XXXX	
肖去			011 0011 1000 0000	XXX XXXX XXXX XXXX	
4	12	チップ消去完了検査	011 0011 1000 0000	xxx xx <mark>S</mark> x xxxx xxxx	(注2)
	2①	フラッシュ書き込み移行	010 0011 0001 0000	XXX XXXX XXXX XXXX	
	22	アドレス上位バイト設定	000 0111 НННН НННН	XXX XXXX XXXX XXXX	(注10)
2	23	アドレス下位バイト設定	000 0011 LLLL LLLL	XXX XXXX XXXX XXXX	
ラッシュ	24	データ下位バイ設定	001 0011 WWWW WWWW	XXX XXXX XXXX XXXX	
ا خ	2 (5)	データ上位バイト設定	001 0111 WWWW WWWW	XXX XXXX XXXX XXXX	
			011 0111 0000 0000	XXX XXXX XXXX XXXX	(注1)
Ė	2 ⑥	データ ラッチ	111 0111 0000 0000	XXX XXXX XXXX XXXX	
メモリ書き込み			011 0111 0000 0000	XXX XXXX XXXX XXXX	
書   き			011 0111 0000 0000	XXX XXXX XXXX XXXX	(注1)
ᇗ	2⑦	ページ書き込み	011 0101 0000 0000	XXX XXXX XXXX XXXX	
<del>آ</del>	20		011 0111 0000 0000	XXX XXXX XXXX XXXX	
			011 0111 0000 0000	XXX XXXX XXXX XXXX	
	2 (8)	ページ書き込み完了検査	011 0111 0000 0000	xxx xx <mark>S</mark> x xxxx xxxx	(注2)
2	3①	フラッシュ読み出し移行	010 0011 0000 0010	XXX XXXX XXXX XXXX	
フト	3②	アドレス上位バイト設定	000 0111 НННН НННН	XXX XXXX XXXX XXXX	(注10)
ラッシュ売出	3 ③	アドレス下位バイト設定	000 0011 LLLL LLLL	XXX XXXX XXXX XXXX	
売│			011 0010 0000 0000	XXX XXXX XXXX XXXX	
#	3 4	データ下位/上位バイト取得	011 0110 0000 0000	xxx xxxx RRRR RRRR	下位バイト
٠			011 0111 0000 0000	xxx xxxx RRRR RRRR	上位バイト
	4①	EEPROM書き込み移行	010 0011 0001 0001	XXX XXXX XXXX XXXX	
	42	アドレス上位バイト設定	000 0111 HHHH HHHH	XXX XXXX XXXX XXXX	(注10)
┋╽	43	アドレス下位バイト設定	000 0011 LLLL LLLL	XXX XXXX XXXX XXXX	
5	4 4	データバイ設定	001 0011 WWWW WWWW	XXX XXXX XXXX XXXX	
EIEPROS書き込み		_, _	011 0111 0000 0000	XXX XXXX XXXX XXXX	(注1)
וע	4 (5)	データ ラッチ	111 0111 0000 0000	XXX XXXX XXXX XXXX	
#			011 0111 0000 0000	XXX XXXX XXXX XXXX	
<u> </u>			011 0011 0000 0000	XXX XXXX XXXX XXXX	(注1)
<u>~</u>	4 6	EEPROM書き込み	011 0001 0000 0000	XXX XXXX XXXX XXXX	
			011 0011 0000 0000	XXX XXXX XXXX XXXX	
	4 (7)	PEDDOM書もはも今マ松木	011 0011 0000 0000	XXX XXXX XXXX XXXX	( <del>}</del> 0)
	4 ⑦	EEPROM書き込み完了検査	011 0011 0000 0000	xxx xxSx xxxx xxxx	(注2)
	5①	EEPROM読み出し移行 アドレス上位バイ・設定	010 0011 0000 0011 000 0111 HHHH HHHH	XXX XXXX XXXX XXXX	(計10)
₹	5 3	アドレス下位バイト設定 アドレス下位バイト設定	000 0111 HHHH HHHH		(注10)
ROV 売出	0 0	アレスト位ハ 17 設定	011 0011 LLLL LLLL	XXX XXXX XXXX XXXX	
売	5 4	EEPROM読み出し	011 0011 EEEE EEEE		
栉│	2 ( <del>1</del> )	EEF KOM 別の大田C	011 0010 0000 0000	xxx xxxx xxxx xxxx xxxx xxxx	
	6①	ヒューズ書き込み移行	010 0011 0000 0000		
	62	データ下位バイ設定	001 0011 0100 0000 0000 0001 0011 WWWW WWWW	XXX XXXX XXXX XXXX	(注3) ビット配置は176頁の表118.参照
	0 2		011 1011 0000 0000	XXX XXXX XXXX XXXX	(注1)
			011 1011 0000 0000	XXX XXXX XXXX XXXX	()
	63	拡張ヒューズ書き込み	011 1011 0000 0000	XXX XXXX XXXX XXXX	
_			011 1011 0000 0000	XXX XXXX XXXX XXXX	
ニューズ	6 4	拡張ヒューズ書き込み完了検査	011 1011 0000 0000	XXX XXXX XXXX XXXX	(注2)
7"	6 5	データ下位バイ設定	001 0011 WWWW WWWW	XXX XXXX XXXX XXXX	(注3) ビット配置は176頁の表119.参照
^		. , ,	011 0111 0000 0000	XXX XXXX XXXX XXXX	(注1)
ッ	0.0	I //mb	011 0101 0000 0000	XXX XXXX XXXX XXXX	
ビット書き込み	6 6	上位側ヒューズ書き込み	011 0111 0000 0000	XXX XXXX XXXX XXXX	
き			011 0111 0000 0000	XXX XXXX XXXX XXXX	
ᇗ	6 ⑦	上位ヒューズ書き込み完了検査	011 0111 0000 0000	xxx xxSx xxxx xxxx	(注2)
4	68	データ下位バイト設定	001 0011 WWWW WWWW	XXX XXXX XXXX XXXX	(注3) ビット配置は176頁の表120.参照
			011 0011 0000 0000	XXX XXXX XXXX XXXX	(注1)
		T H= /mb	011 0001 0000 0000	XXX XXXX XXXX XXXX	
	0.00				
	6 (9)	下位側ヒューズ書き込み	011 0011 0000 0000	xxx xxxx xxxx xxxx	
	6 (9)	ト位側にュース書さ込み	011 0011 0000 0000 011 0011 0000 0000	XXX XXXX XXXX XXXX	

次頁へ続く





## 表131 (続き). JTAG プログラミング 命令一式

		命令	TDI入力	TDO出力	備考
施	7①	施錠ビット書き込み移行	010 0011 0010 0000	XXX XXXX XXXX XXXX	
錠	7②	データ下位バイ設定	001 0011 11 <mark>WW WWWW</mark>	XXX XXXX XXXX XXXX	(注4) ビット配置は175頁の表116.参照
ーツ			011 0011 0000 0000	XXX XXXX XXXX XXXX	(注1)
Ţ	7.00	施錠ビット書き込み	011 0001 0000 0000	XXX XXXX XXXX XXXX	
施錠ビット書き込み	73	地鉄に外書さ込み	011 0011 0000 0000	XXX XXXX XXXX XXXX	
认			011 0011 0000 0000	XXX XXXX XXXX XXXX	
み	7 4	施錠ビット書き込み完了検査	011 0011 0000 0000	xxx xxSx xxxx xxxx	(注2)
	8①	ヒューズ/施錠ビット読み出し移行	010 0011 0000 0100	XXX XXXX XXXX XXXX	
	82	拡張ヒューズ読み出し	011 1010 0000 0000	XXX XXXX XXXX XXXX	(*****) ビット配置は176頁の表118.参照
닏	0 2	拡張は、水説が田し	011 1011 0000 0000	xxx xxxx RRRR RRRR	(注5) と外配直は170頁の及116.参照
一十	83	上位側ヒューズ読み出し	011 1110 0000 0000	XXX XXXX XXXX XXXX	( ) と、と、か配置は176頁の表119.参照
ズ	0 0	工団関ロスが配り口し	011 1111 0000 0000	xxx xxxx RRRR RRRR	(注5) [(注5)]
ヒューズ/施錠ビッ	8 (4)	下位側ヒューズ読み出し	011 0010 0000 0000	XXX XXXX XXXX XXXX	   とうない とうない とうない とうない とうない とうない とうない とうない
錠	0 4		011 0011 0000 0000	xxx xxxx RRRR RRRR	(注5) 「外配置は170員の数720.参照
ード	8 ⑤	施錠ビット読み出し	011 0110 0000 0000	XXX XXXX XXXX XXXX	   と
上	0 0	加速に 月 前67日 0	011 0111 0000 0000	xxx xxxx xxRR RRRR	(注5) (注5)
ト読み出			011 1010 0000 0000	XXX XXXX XXXX XXXX	
出出			011 1110 0000 0000	xxx xxxx RRRR RRRR	拡張ヒューズ
し	8 6	ヒューズ/施錠ビット読み出し	011 0010 0000 0000	xxx xxxx RRRR RRRR	上位ヒュース ( <b>注5</b> )
			011 0110 0000 0000	xxx xxxx RRRR RRRR	ト位ピュース
	_		011 0111 0000 0000	xxx xxxx RRRR RRRR	施錠ビット
識票バイト	9①	識票バイ読み出し移行	010 0011 0000 1000	XXX XXXX XXXX XXXX	
ボバ	92	アドレス バイト設定	000 0011 LLLL LLLL	XXX XXXX XXXX XXXX	
7	93	識票バイト読み出し	011 0010 0000 0000	XXX XXXX XXXX XXXX	
			011 0011 0000 0000	xxx xxxx RRRR RRRR	
校	10①	校正バイ・読み出し移行	010 0011 0000 1000	XXX XXXX XXXX XXXX	
校正バ	10②	アドレス バイト設定	000 0011 LLLL LLLL	XXX XXXX XXXX XXXX	
17	103	校正バ仆読み出し	011 0110 0000 0000	XXX XXXX XXXX XXXX	
-	100	7 11 100 7 1100	011 0111 0000 0000	xxx xxxx RRRR RRRR	
	11①	無操作設定	010 0011 0000 0000	XXX XXXX XXXX XXXX	
	11 (	AND THE BOX	011 0011 0000 0000	XXX XXXX XXXX XXXX	

注1:この命令は直前の命令手順によって上位7ビットが正しく設定されている(通常の)場合、必要ではありません。

注2:Sが1になるまで繰り返します。(待機)

注3:設定値0でプログラム、1で非プログラムです。

注4:設定値0でプログラム、1では施錠ビットが変化しません。

**注5**:読み出し値0でプログラム、1で非プログラムです。

(訳注) 原書での注6~9は表内備考欄に直接記載しました。

注10: PCMSB(表121.参照)とEEAMSB(表122.参照)を越えるアドレス ビットは無効です。

注11:全てのTDIとTDO手順は2進数で示されています。

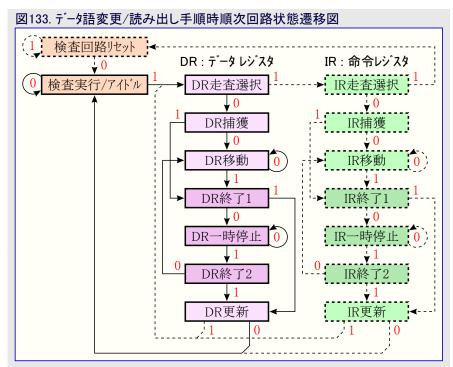
## フラッシュ バイト データ (Flash Data Byte) レジスタ

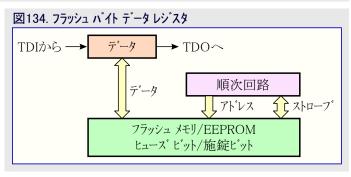
フラッシュ バイト データ レジスタはページ書き込み実行前のフラッシュ メモリ ページ緩衝部全体設定、またはフラッシュ メモリ内容の読み出しや照合の効率的な方法を提供します。順次回路はフラッシュ メモリへの制御信号を設定し、フラッシュ メモリからのストローブ信号を判断し、従って移動入出力されるためのデータ語だけが必要です。

実際のフラッシュ ハイト データレジスタは8ビットの走査チェーンと8ビットの一時レジスタから成ります。ページ設定中、DR更新状態は走査チェーンによる内容を一時レジスタへ複写し、11 TCK周期内で一時レジスタの内容をフラッシュメモリページ緩衝部へ設定する書き込み(設定)手順を始めます。AVRはPROG\_PAGELOAD命令移行後に出会った最初のDR更新に対して下位ハイトで始め、新規DR更新毎に上位/下位ハイト書き込み間を自動的に切り替えます。プログラムカウンタは最初の書き込みハイトを除き、下位ハイト書き込み前に予め増加(+1)されます。これはPROG\_COMMANDSによって設定したアトレスに最初のデータが書かれ、ページ緩衝部の最終位置設定がプログラムカウンタを次ページに増加しないことを保証します。

ヘーシ、読み込み中、選んだフラッシュメモリのハイトの

内容はDR捕獲状態中にフラッシュ バイト データ レジスタ内に捕獲されます。AVRはPROG\_PAGEREAD命令移行後に出会った最初のDR捕獲に対して下位バイトで始め、新規DR捕獲毎に上位/下位バイト読み込み間を自動的に切り替えます。プログラム カウンタは最初の読み込みバイトを含め、各上位バイト読み込み後に増加(+1)されます。これは最初のデータがPROG\_COMMANDSによって設定した先頭アドレスから捕獲され、ページの最終位置読み込みがプログラム カウンタを次ページに増加(進行)することを保証します。





フラッシュ バイト データ レジスタを制御する順次回路はTCKによってクロック駆動されます。通常動作中内でフラッシュ メモリの各バイトに対して8 ビットが移動され、TAP制御器を通して誘導するために必要としたこのクロック周期はフラッシュ バイト データ レジスタに対し、それが使用者にとって明白に操作を完了するための充分なクロック パルス数で順次回路を自動的に送ります。けれどもページ設定中に各DR更新状態間で更に数ビットが移動される場合、各DR更新状態間の最小11 TCK周期を保証するためのいくつかのTCK周期に対し、TAP制御器は検査実行/アイドル状態に留まるべきです。

#### プログラミング手法

1(1)、1(2)形式のような以下の全ての参照は表131を参照してください。

#### プログラミング動作への移行

- 1. JTAG命令AVR\_RESETを入力し、リセット レジスタに1を移動します。
- 2. PROG\_ENABLE命令を入力し、プログラミング許可レシブスタに\$A370(1010 0011 0111 0000)を移動します。

#### プログラミング動作からの抜け出し

- 1. JTAG命令PROG COMMANDSを入力します。
- 2. 無操作命令(11①参照)により、全てのプログラミング命令を禁止します。
- 3. PROG\_ENABLE命令を入力し、プログラミング許可レシ、スタに \$0000(0000 0000 0000 0000)を移動します。
- 4. JTAG命令AVR RESETを入力し、リセットレジスタに0を移動します。

#### チップ消去の実行

- 1. JTAG命令PROG\_COMMANDSを入力します。
- 2. チップ消去命令(11)参照)を使い、チップ消去を開始します。
- 3. チップ消去完了検査(12)参照)を使い、完了までポーリングするか、twl.RH\_CE(184頁の表128.参照)時間待ちます。





## フラッシュ メモリの書き込み

フラッシュ メモリへ書き込む前にチップ消去が実行されなければなりません。191頁の「**チップ消去の実行**」をご覧ください。

- 1. JTAG命令PROG\_COMMANDSを入力します。
- 2. フラッシュ書き込み移行命令(2①参照)を使い、フラッシュ メモリ書き込みを許可します。
- アトレス上位設定命令(2②参照)を使い、書き込みアトレス上位バイトを設定します。
- 4. アドレス下位設定命令(23)参照)を使い、書き込みアドレス下位バイトを設定します。
- 5. データ設定命令(24), 25, 26参照)を使い、書き込みデータを設定します。
- 6. ページ内の全語(ワード)数分4.と5.を繰り返します。
- 7. ページ書き込み命令(2⑦参照)を使い、ページをフラッシュ メモリに書き込みます。
- 8. ページ書き込み完了検査(28)参照)を使い、完了までポーリングするか、twlrh\_Flash(184頁の表128.参照)時間待ちます。
- 9. 全データ書き込みまで3.~8.を繰り返します。

より効率的なデータ転送はPROG PAGELOAD命令を使うことで達せられます。

- 1. JTAG命令PROG\_COMMANDSを入力します。
- 2. フラッシュ書き込み移行命令(2①参照)を使い、フラッシュ メモリ書き込みを許可します。
- 3. アドレス設定命令(2②, 2③参照)を使い、ページアドレスを設定します。 ページ内アドレスにはPCWORD(177頁の表121.参照)が使われ、これらのビットは0として書かれなければなりません。
- **4.** JTAG命令PROG PAGELOADを入力します。
- 5. 先頭命令のLSBから始まって最終命令のMSBで終わるページ内にバイ単位で全命令語を移動入力することによってページ全体を設定します。フラッシュ バイト データ レジスタの内容をフラッシュ ページ位置に複写し、各新規語(ワート)前にプログラム カウンタを自動増加するのにDR更新を使います。
- 6. JTAG命令PROG\_COMMANDSを入力します。
- 7. ページ書き込み命令(2⑦参照)を使い、ページをフラッシュ メモリに書き込みます。
- 8. ページ書き込み完了検査(2⑧参照)を使い、完了までポーリングするか、twlRH\_FLASH(184頁の表128.参照)時間待機します。
- 9. 全データ書き込みまで3.~8.を繰り返します。

#### フラッシュ メモリの読み出し

- 1. JTAG命令PROG\_COMMANDSを入力します。
- 2. フラッシュ読み出し移行命令(3(1)参照)を使い、フラッシュメモリ読み出しを許可します。
- 3. アドレス設定命令(32)、33参照)を使い、読み出しアドレスを設定します。
- 4. データ読み出し命令(34)参照)を使い、データを読み出します。
- 5. 全データ読み出しまで3.~4.を繰り返します。

より効率的なデータ転送はPROG\_PAGEREAD命令を使うことで達せられます。

- 1. JTAG命令PROG\_COMMANDSを入力します。
- 2. フラッシュ読み出し移行命令(3①参照)を使い、フラッシュ メモリ読み出しを許可します。
- 3. アドレス設定命令(3②, 3③参照)を使い、ペーシ、アドレスを設定します。ペーシ、内アドレスにはPCWORD(177頁の表121.参照)が使われ、これらのビットは0として書かれなければなりません。
- 4. JTAG命令PROG\_PAGEREADを入力します。
- 5. 先頭命令のLSBから始まって最終命令のMSBで終わるページ(またはフラッシュ)内で全命令語を移動出力することで、ページ(またはフラッシュ)全体を読みます。DR捕獲状態はフラッシュメモリからのデータ捕獲と各語(ワート)が読まれた後にプログラム カウンタも自動増加します。DR移動に先立ってDR捕獲が起きることに注意してください。従って移動出力される先頭バイトは有効なデータを含みます。
- 6. JTAG命令PROG\_COMMANDSを入力します。
- 7. 全データ読み出しまで3.~6.を繰り返します。

## EEPROMの書き込み

EEPROMへ書き込む前にチップ消去が実行されなければなりません。191頁の「チップ消去の実行」をご覧ください。

- 1. JTAG命令PROG\_COMMANDSを入力します。
- 2. EEPROM書き込み移行命令(4①参照)を使い、EEPROMメモリ書き込みを許可します。
- 3. アドレス上位設定命令(42)参照)を使い、書き込みアドレス上位バイトを設定します。
- 4. アドレス下位設定命令(43)参照)を使い、書き込みアドレス下位バイを設定します。
- 5. データ設定命令(44), 45参照)を使い、書き込みデータを設定します。
- 6. ページ内の全バイ数分4.と5.を繰り返します。
- 7. EEPROM書き込み命令(46)参照)を使い、データをEEPROMメモリに書き込みます。
- 8. EEPROM書き込み完了検査(47)参照)を使い、完了までポーリングするか、twl.RH(184頁の表128.参照)時間待ちます。
- 9. 全データ書き込みまで3.~8.を繰り返します。

PROG\_PAGELOAD命令がEEPROM書き込み時に使えないことに注意してください。

## EEPROMの読み出し

- 1. JTAG命令PROG COMMANDSを入力します。
- 2. EEPROM読み出し移行命令(5①参照)を使い、EEPROMメモリ読み出しを許可します。
- 3. アトンス設定命令(5<sup>2</sup>), 5<sup>3</sup>参照)を使い、読み出しアトンスを設定します。
- 4. データ読み出し命令(54)参照)を使い、データを読み出します。
- 5. 全データが読み出されてしまうまで3.~4.を繰り返します。

PROG PAGEREAD命令がEEPROM読み出し時に使えないことに注意してください。

## ヒュース ビットの書き込み (訳注:本項は修正してあります。)

- 1. JTAG命令PROG\_COMMANDSを入力します。
- 2. ヒュース 書き込み移行命令(6①参照)を使い、ヒュース 書き込みを許可します。
- 3. 拡張データ設定命令(62)参照)を使い、拡張ヒュース・値を設定します。(0=プログラム、1=非プログラム)
- 4. 拡張ヒューズ書き込み命令(63)参照)を使い、拡張ヒューズを書き込みます。
- 5. ヒュース、書き込み完了検査(6④参照)を使い、完了までポーリングするか、tWLRH(184頁の表128.参照)時間待ちます。
- 6. 上位データ設定命令(65)参照)を使い、上位側ヒュース、値を設定します。(0=プログラム、1=非プログラム)
- 7. 上位ヒューズ書き込み命令(66)参照)を使い、上位側ヒューズを書き込みます。
- 8. ヒュース・書き込み完了検査(6⑦参照)を使い、完了までポーリングするか、tWLRH(184頁の表128.参照)時間待ちます。
- 9. 下位データ設定命令(6⑧参照)を使い、下位側ヒューズ値を設定します。(0=プログラム、1=非プログラム)
- 10. 下位ヒュース、書き込み命令(69)参照)を使い、下位側ヒュース、を書き込みます。
- 11. ヒューズ書き込み完了検査(6⑩参照)を使い、完了までポーリングするか、tWLRH(184頁の表128.参照)時間待ちます。

#### 施錠ビットの書き込み

- 1. JTAG命令PROG\_COMMANDSを入力します。
- 2. 施錠ビット書き込み移行命令(7①参照)を使い、施錠ビット書き込みを許可します。
- 3. データ設定命令(72)参照)を使い、施錠ビット値を設定します。(0=プログラム、1=無変化)
- 4. 施錠ビット書き込み命令(73)参照)を使い、施錠ビットに書き込みます。
- 5. 施錠ビット書き込み完了検査(7個参照)を使い、完了までポーリングするか、tWLRH(184頁の表128.参照)時間待ちます。

#### ヒュース・/施錠ビットの読み出し(訳注:本項は修正してあります。)

- 1. JTAG命令PROG COMMANDSを入力します。
- 2. ヒューズ/施錠ビット読み出し移行命令(8①参照)を使い、ヒューズ/施錠ビット読み出しを許可します。
- 3. 全てのヒューズと施錠ビットを読むにはヒューズ/施錠ビット読み出し命令(8億参照)を使います。 拡張ヒューズ・ビットのみを読み出す場合は拡張ヒューズ・読み出し命令(8億参照)を使います。 上位側ヒューズ・ビットだけを読むには上位ヒューズ・読み出し命令(8億参照)を使います。 下位側ヒューズ・ビットだけを読むには下位ヒューズ・読み出し命令(8億参照)を使います。 施錠ビットだけを読むには施錠ビット読み出し命令(8億多無)を使います。

#### 識票が小の読み出し

- 1. JTAG命令PROG COMMANDSを入力します。
- 2. 識票読み出し移行命令(9①参照を使い、識票読み出しを許可します。
- 3. アトンス設定命令(92)参照)を使い、読み出しアトンス\$00を設定します。
- 4. データ読み出し命令(93)参照)を使い、識票バイを読み出します。
- 5. 第2、第3 $^{1}$ 小を読むためにアドレスを $^{1}$ 02として各々 $^{1}$ 2.~4.を繰り返します。

## 校正バイの読み出し

- 1. JTAG命令PROG\_COMMANDSを入力します。
- 2. 校正バイト読み出し移行命令(10(1)参照)を使い、校正バイト読み出しを許可します。
- 3. アドレス設定命令(10<sup>2</sup>)参照)を使い、読み出しアドレス\$00を設定します。
- 4. データ読み出し命令(10(3)参照)を使い、校正バイトを読み出します。





# 電気的特性

# 絶対最大定格 (警告)

#### (警告)

絶対最大定格を超える負担はデバイスに定常的な損傷を与えます。 絶対最大定格は負担の定格を示すためだけのもので、この値また は、この仕様書の動作特性で示された値を超える条件で動作する ことを示すものではありません。長時間の最大定格での使用はデバイスの信頼性を損なう場合があります。

# DC特性

TA=-40°C~85°C, VCC=1.8V~5.5V (特記事項を除く)

シンホ゛ル	項目		条件	最小	代表	最大	単位
VIL	Lowレベル入力電圧		VCC=1.8∼2.4V	-0.5		0.2VCC ( <b>注1</b> )	
VIL	(XTAL1,RESETを除く)		VCC=2.4~5.5V	-0.5		0.3VCC ( <b>注1</b> )	
V <sub>IL1</sub>	Lowレベル入力電圧 (XTAL1)		VCC=1.8∼5.5V	-0.5		0.1VCC ( <b>注1</b> )	
VIL2	Lowレベル入力電圧 (R)	ECET)	VCC=1.8∼2.4V	-0.5		0.1VCC ( <b>注1</b> )	
VIL2	LOWV・ハバハ 見儿 (IV.	ESET)	VCC=2.4~5.5V	-0.5		0.2VCC ( <b>注1</b> )	
VIH	Highレベル入力電圧		VCC=1.8∼2.4V	0.7VCC ( <b>注2</b> )		VCC+0.5	
VIH	(XTAL1,RESETを除く)	)	VCC=2.4~5.5V	0.6VCC ( <b>注2</b> )		VCC+0.5	
V <sub>IH1</sub>	Highレヘブル入力電圧 (X	TAL 1)	VCC=1.8∼2.4V	0.8VCC ( <b>注2</b> )		VCC+0.5	
VIHI	Tilgiiv、ルハハル 电圧(A	(IALI)	VCC=2.4~5.5V	0.7VCC ( <b>注2</b> )		VCC+0.5	V
VIH2	Highレベル入力電圧(R	ESET)	VCC=1.8∼5.5V	0.9VCC ( <b>注2</b> )		VCC+0.5	V
Vol		ポートB以外	IOL=10mA, VCC=5V			0.7	
VOL	   Lレヘブル出力電圧( <mark>注3</mark> )	W _LDFY\	IOL=5mA, VCC=3V			0.5	
V <sub>OL1</sub>		ホ°ートB	IOL=20mA, VCC=5V			0.7	
VOLI		W . LD	IOL=10mA, VCC=3V			0.5	
Vон		ポートB以外	IOH=-10mA, VCC=5V	4.2			
VOH	   Hレヘ・ル出力電圧( <mark>注4</mark> )	W LDSY	IOH=-5mA, VCC=3V	2.3			
VOH1		ー ボートB	IOH=-20mA, VCC=5V	4.2			
		·	IOH=-10mA, VCC=3V	2.3			
IIL	I/OビンLowレベル入力》		VCC=5.5V			1	μA
IIH	I/OビンHighレベル入力		確実なH/L範囲			1	μ. ι
RRST	RESETピン プルアップ抵	抗		30		60	kΩ
Rpu	I/Oピン プルアップ抵抗			20		50	17.52
	活動動作消費電流		VCC=2V, 1MHz			0.44	
	「電力削減レジスタ(PRR)	全ビット=1)	VCC=3V, 4MHz			2.5	
	(.0)		VCC=5V, 8MHz			9.5	mA
ICC	アイル動作消費電流		VCC=2V, 1MHz			0.2	1112 1
100	(電力削減レジスタ(PRR)	·全ビット=1)	VCC=3V, 4MHz			0.8	
	(18) 3 13 4 12 12 12 13 13 13 13 13 13 13 13 13 13 13 13 13		VCC=5V, 8MHz			3.3	
	パワーダウン動作消費電流		VCC=3V, WDT有効		<8	10	μA
			VCC=3V, WDT禁止		<1	2	pr i
	アナログ比較器入力変位		VCC=5V, Vin=VCC/2		<10	40	mV
IACLK	アナログ比較器入力漏れ	に電流		-50		50	nA
TACPD	アナログ比較器伝播遅延	£時間	VCC=2.7V		750		ns
VACED		=- 4 164	VCC=4.0V		500		110

注1: Lowレベルの認識が保証される最高電圧です。 注2: Highレベルの認識が保証される最低電圧です。

(注3)~(注4)は次頁を参照してください。

**注3**: 各I/Oポートは安定状態(非過渡時)に於いて検査条件(ポートBはVCC=3Vで10mA,VCC=5Vで20mA、他のポートはVCC=3Vで5mA,VCC=5Vで10mA)よりも多くの吸い込み電流を流すことができますが、次の条件を厳守しなければなりません。

- 1. 全ポートのIOLの合計が400mAを超えるべきではありません。
- 2. ポートA7~0、C7~4、G2のIOLの合計が100mAを超えるべきではありません。
- 3. ポートB7~0、E7~0、G5~3のIOLの合計が100mAを超えるべきではありません。
- **4.** ポートC3~0、D7~0、G1~0のIOLの合計が100mAを超えるべきではありません。
- 5. ポートF7~0のIOLの合計が100mAを超えるべきではありません。

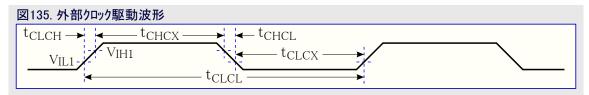
IOLが検査条件を超える場合、VOLも仕様書での値を超えます。表の検査条件よりも大きな吸い込み電流を流すことは保証されません。

注4: 各I/Oポートは安定状態(非過渡時)に於いて検査条件(ポートBはVCC=3Vで10mA,VCC=5Vで20mA、他のポートはVCC=3Vで5mA,VCC=5Vで10mA)よりも多くの吐き出し電流を流すことができますが、次の条件を厳守しなければなりません。

- 1. 全ポートのIOHの合計が400mAを超えるべきではありません。
- 2. ポートA7~0、C7~4、G2のIOHの合計が100mAを超えるべきではありません。
- 3. ポートB7~0、E7~0、G5~3のIOHの合計が100mAを超えるべきではありません。
- 4. ポートC3~0、D7~0、G1~0のIOHの合計が100mAを超えるべきではありません。
- 5. ポートF7~0のIOHの合計が100mAを超えるべきではありません。

IOHが検査条件を超える場合、VOHも仕様書での値を超えます。表の検査条件よりも大きな吐き出し電流を流すことは保証されません。

# 外部クロック特性



#### 表132. 外部クロック特性

シンホ゛ル	ボル 項目		8 <b>∼</b> 5.5∨	VCC=2.	7 <b>~</b> 5.5V	VCC=4.	5 <b>∼</b> 5.5∨	単位
フクホ ル	· · · · · · · · · · · · · · · · · · ·	最小	最大	最小	最大	最小	最大	中位
1/tclcl	クロック周波数	0	1	0	8	0	16	MHz
tclcl	クロック周期	1000		125		62.5		
tchcx	Highレヘル時間	400		50		25		ns
tclcx	Lowレヘル時間	400		50		25		
tclch	上昇時間		2.0		1.6		0.5	110
tchcl	下降時間		2.0		1.6		0.5	μs
⊿tclcl	隣接クロック周期間の変化率		2		2		2	%

注: 詳細については19頁の「外部クロック信号」を参照してください。

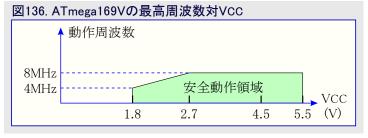
## 最高速と動作電圧

最高周波数は動作電圧に依存します。**図136**.と**図137**.で示されるように最高周波数対動作電圧曲線は1.8~4.5V間で直線です。本区間で与えられた電圧での最高周波数を計算するには①式を使ってください。与えられた周波数に対して必要な電圧を計算するには②式を使ってください。

## ①式····周波数= $a \times ($ 電圧-Vx)+Fy ②式···電圧= $b \times ($ 周波数-Fy)+Vx

3Vでの周波数は $8\div1.8\times(3-2.7)+8$ でVCC=3V時の最高周波数は9.33MHzです。6MHzでの電圧は $1.8\div8\times(6-4)+1.8$ で、最高周波数6MHzにはVCC=2.25Vが必要です。

表133. 最高速対動作電圧計算に使われる定数							
電圧,周波数範囲	а	b	Vx	Fy			
2.7~4.5V,8~16MHz	0 /1 0	1.8/8	2.7	8			
1.8~2.7V,4~8MHz	8/1.8		1.8	4			

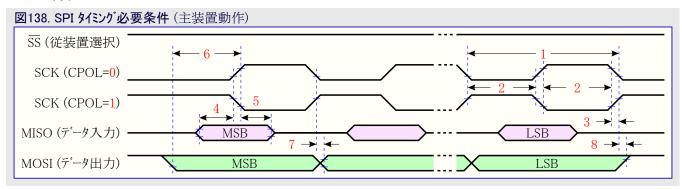


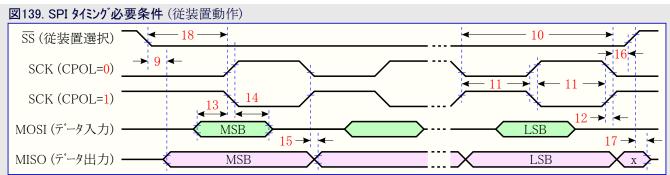






# SPIタイミング特性





## 表134. SPI タイミング 特性

番号	項目	動作種別	最小	代表	最大	単位
1	SCK周期	主装置		表69.参照		
2	SCK High/Low期間	主装置		50%デューティ比		
3	SCK上昇/下降時間	主装置		3.6		
4	入力データ 準備時間	主装置		10		
5	入力データ 保持時間	主装置		10		
6	出力からSCK変移時間	主装置		0.5t <sub>SCK</sub>		ns
7	SCKからの出力遅延時間	主装置		10		
8	SCKからのHigh出力時間	主装置		10		
9	SS↓からの出力遅延時間	従装置		15		
10	SCK周期	従装置	4tcK			
11	SCK High/Low期間 (注)	従装置	2tck			
12	SCK上昇/下降時間	従装置		1.6		μs
13	入力データ 準備時間	従装置	10			
14	入力データ 保持時間	従装置	tcĸ			
15	SCKからの出力遅延時間	従装置		15		ne
16	SCKからのSS↑遅延時間	従装置	20			ns
17	SS↑からの出力Hi-Z遅延時間	従装置		10		
18	SS↓からのSCK遅延時間	従装置	20t <sub>CK</sub>			

注: SPIプログラミングでの最小SCK High/Low期間は、2tCLCL(tCK < 12MHz)、3tCLCL(tCK ≥ 12MHz)です。

# A/D変換器特性 (暫定)

表1	35	Δ/Γ	李	鱼!	持性

シンホ゛ル	項目		条件	最小	代表	最大	単位	
	/\An Ah	シングル エントブ	力変換		10		ا د قوا	
	分解能	差動入力変換			8		ピット	
	絶対精度		変換クロック=200kHz		2	2.5		
	(積分非直線性誤差、	入力変換 変	変換クロック=1MHz		4.5			
	微分非直線性誤差、 量子化誤差、利得誤差、		変換クロック=200kHz 雑音低減動作		2			
	変位(オフセット)誤差を含む)	VREF=4V	変換クロック=1MHz 雑音低減動作		4.5		LSB	
	積分非直線性誤差	3			0.5			
	微分非直線性誤差	シングル エントプ VCC=4V、VR			0.25			
	利得誤差	VCC-4V、VK  変換クロック=20			2			
	変位(オフセット)(セ´ロ)誤差	· 炎·沃/-// 20	70KI 12		2			
	変換時間	連続変換動作	F	13		260	μs	
	変換クロック周波数			0.05		1	MHz	
AVCC	アナログ供給電圧			VCC-0.3		VCC+0.3		
Vref	基準電圧	シングル エントブ	力変換	1.0		AVCC		
VKEF	本中电圧	差動入力変換	<b>A</b>	1.0		AVCC-0.5	V	
Vin	入力電圧	7. 力電圧	シングル エントブ	力変換	GND		VREF	
VIIN	八八电圧	差動入力変換		0		AVCC (注1)		
	入力周波数帯域	か共城・シング・ルエント・入力変換			38.5		kHz	
	八刀叫仮数衎坳	差動入力変換			4		КПХ	
VINT	内蔵基準電圧			1.0	1.1	1.2	V	
RREF	基準電圧入力インピーダンス				32		kΩ	
Rain	アナログ入力インピーダンス				100		ΜΩ	

注1: VDIFF(差動入力差電圧)はVREF以下でなければなりません。

# LCD制御器特性(暫定)

表136. LCD制御器特性

シンホ゛ル	項目	条件	最小	代表	最大	単位
ILCD	LCD駆動部駆動電流	全COM,SEGt°ン合計		100		μА
RLCD	LCD駆動部出力抵抗	各COM,SEGt°ン		10		kΩ





# 代表特性

以下の図は代表的な特性を示します。これらの図は製造中に検査されていません。全ての消費電流測定は全I/Oピンを入力として設定した内部プルアップ許可で行われています。電源幅振幅の方形波発振器がクロック源として使われています。

活動動作とアイ・ル動作の全ての消費電流測定は電力削減レシ、スタ(PRR)の全ビットが設定(1)で行なわれ、従って関係周辺機能部はOFFにされます。これらの測定中にアナログ・比較器も禁止されます。表137.と表138.はPRRによって制御される全周辺機能部に関して活動動作とアイドル動作のICCに対する追加消費電流を示します。詳細については23頁の「電力削減レジ、スタ(PRR)」をご覧ください。パワーダウン動作での消費電力はクロック選択と無関係です。

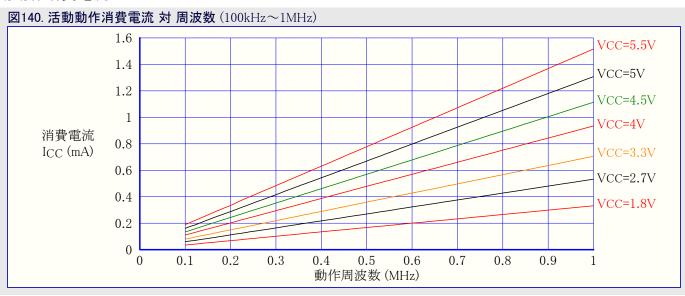
消費電流は動作電圧、動作周波数、I/Oピンの負荷、I/Oピンの切り替え速度、命令実行、周囲温度のような様々な要素の関数です。 支配的な要素は動作電圧と動作周波数です。

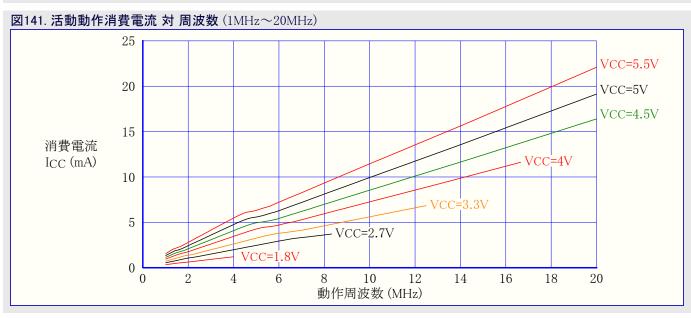
容量性負荷の $\mathfrak{l}^{\circ}$ ンの引き込み電流は(1つの $\mathfrak{l}^{\circ}$ ンに対して) CL(負荷容量)×VCC(動作電圧)×f( $I/O\mathfrak{l}^{\circ}$ ンの平均切り替え周波数) として推測できます。

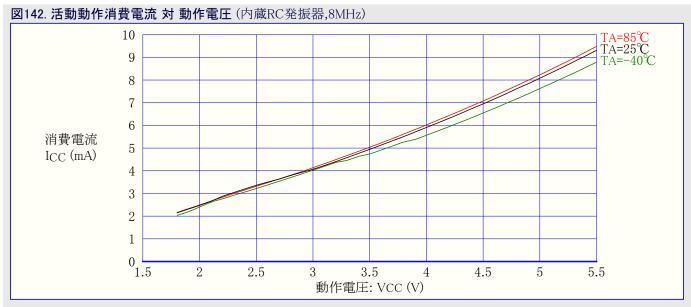
デバイスは検査範囲よりも高い周波数特性を示します。デバイスは注文番号が示す周波数よりも高い周波数での機能特性を保証されません。

ウォッチト、ック、タイマ許可のハ。ワータ・ウン動作での消費電流とウォッチト、ック、タイマ禁止のハ。ワータ・ウン動作での消費電流間の違いは、ウォッチト、ック、タイマによって引き込んだ(消費した)差電流を表します。

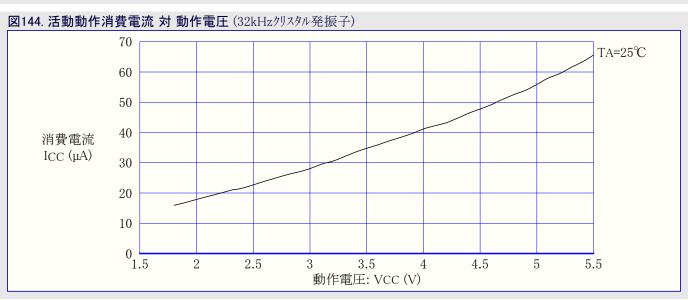
## 活動動作消費電流







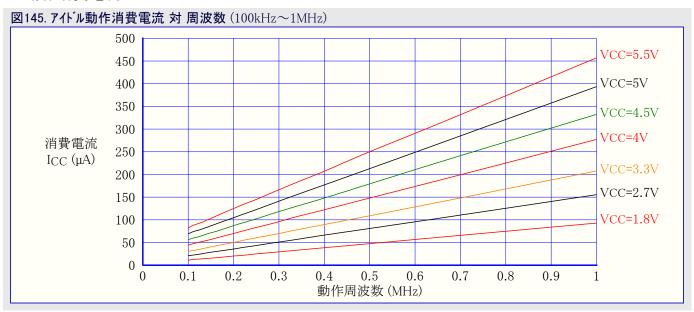


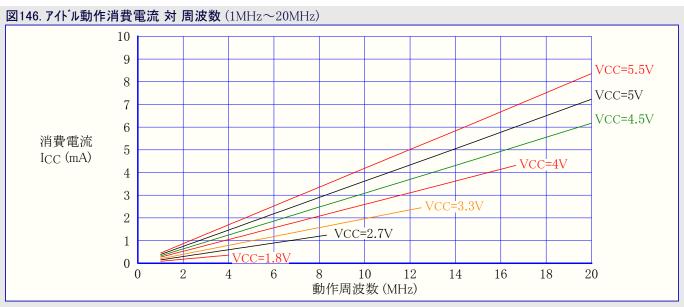




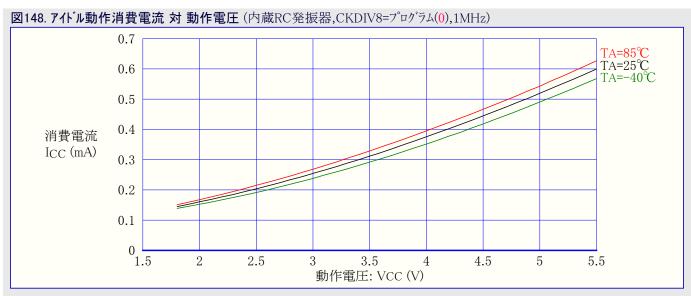


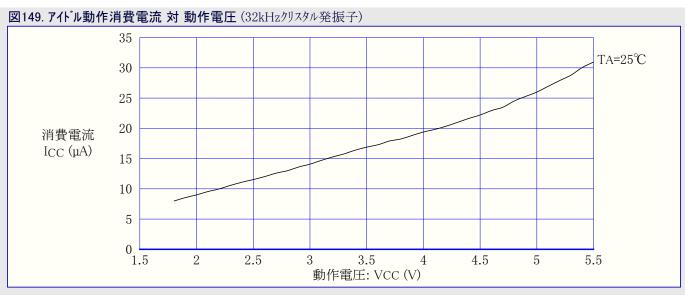
# アイドル動作消費電流















## 周辺機能部供給電流

以下の表と式は活動動作とアイドル動作で個別周辺機能部対する追加消費電流の計算に使えます。周辺機能部の許可や禁止は電力削減レジスタによって制御されます。詳細については23頁の「電力削減レジスタ(PRR)」をご覧ください。

表137. 各部追加消費電流(絶対値:uA)

Z · · · · I II / C / M / M / M / M / M / M / M / M / M				
PRR内ビット	1MHz,2V	2MHz,4V	8MHz,5V	
PRADC	18	116	495	
PRUSART0	11	79	313	
PRSPI	10	72	283	
PRTIM1	19	117	481	
PRLCD	19	124	531	

表138. 各部追加消費電流(相対値:%)

PRR内ビット	活動動作(図140,図141)	ア仆 ル動作(図145,図146)
PRADC	5.6	18.7
PRUSART0	3.7	12.4
PRSPI	3.2	10.8
PRTIM1	5.6	18.8
PRLCD	5.9	19.9

表137.で一覧される以外のVCCと周波数設定については表138.からの数値を元に代表的な消費電流を計算できます。

**例1**: VCC=3V, *f*=1MHzでUSART0,SPI,タイマ/カウンタ1が許可されたア仆ル動作での予測される消費電流を計算します。**表138**.のア仆ル 動作列からUSART0が12.4%、SPIが10.8%、タイマ/カウンタ1が18.6%追加する必要を知ります。**図145**.を読み、VCC=3V, *f*=1MHzでのア仆ル動作消費電流が約0.18mAである事を得ます。USART0,SPI,タイマ/カウンタ1許可のア仆ル動作での総消費電流を得ます。

総消費電流=0.18mA×(1+0.124+0.108+0.186) ⇒ 0.26mA

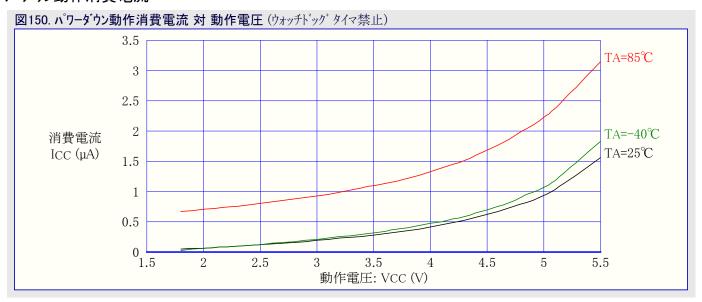
**例2**: 例1と同じ条件ですが、代わりに活動動作です。**表138**.の活動動作列からUSART0が3.7%、SPIが3.2%、タイマ/カウンタ1が5.6%追加する必要を知ります。**図140**.を読み、VCC=3V, *f*=1MHzでの活動動作消費電流が約0.6mAであることを得ます。USART0,SPI, タイマ/カウンタ1許可の活動動作での総消費電流を得ます。

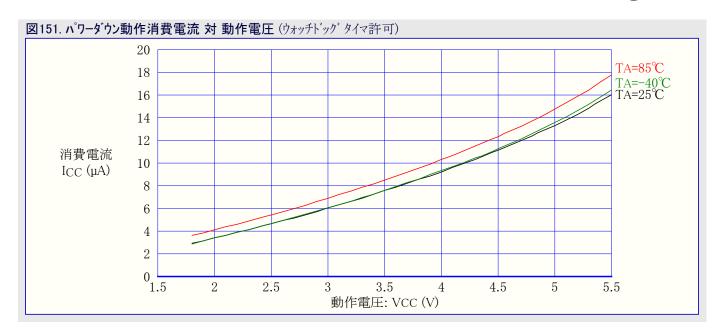
総消費電流=0.6mA×(1+0.037+0.032+0.056)≒0.68mA

**例3**: VCC=3.3V, f=10MHzで全周辺機能部許可の活動動作での予測消費電流を計算します。周辺機能部を除く活動動作消費電流 が約5.6mAであることを得ます(図141.より)。そして表138.の活動動作列からの数値を使い、総消費電流を得ます。

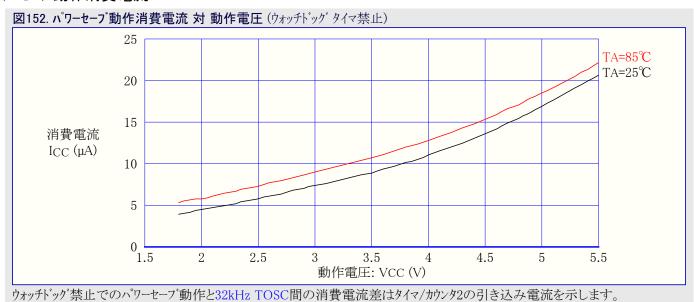
総消費電流=5.6mA×(1+0.056+0.037+0.032+0.056+0.059) ⇒ 6.9mA

## パワーダウン動作消費電流

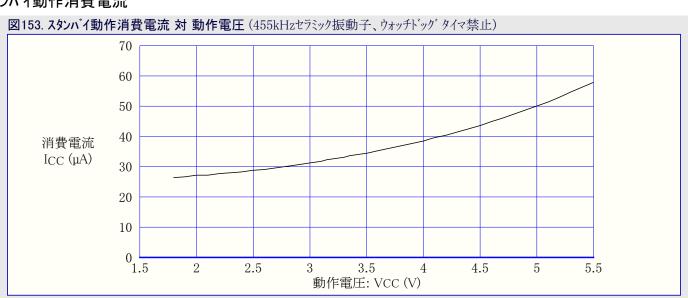




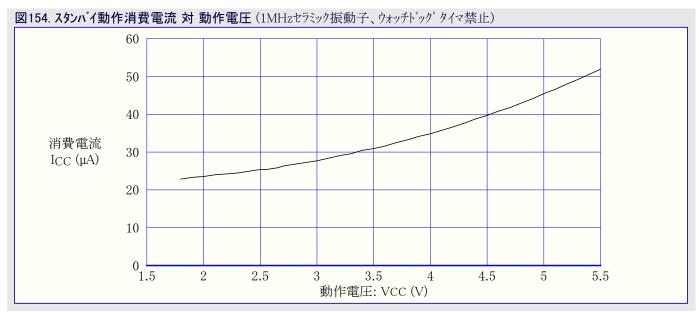
# パワーセーブ動作消費電流

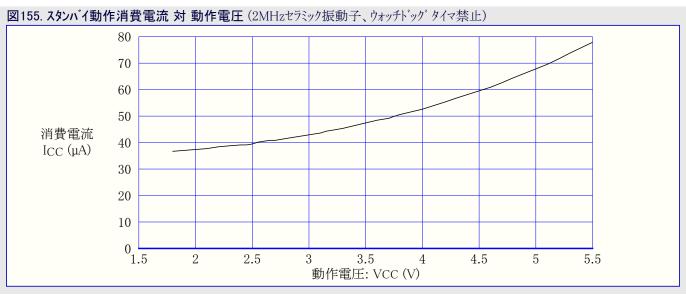


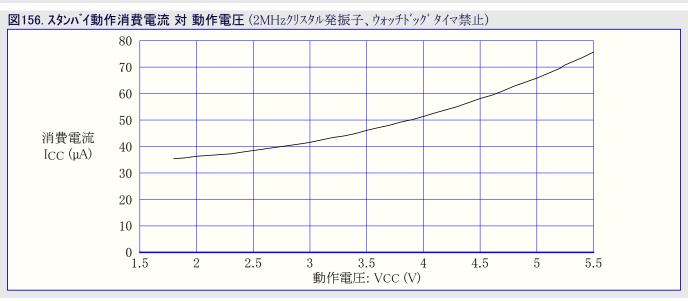
# スタンバイ動作消費電流

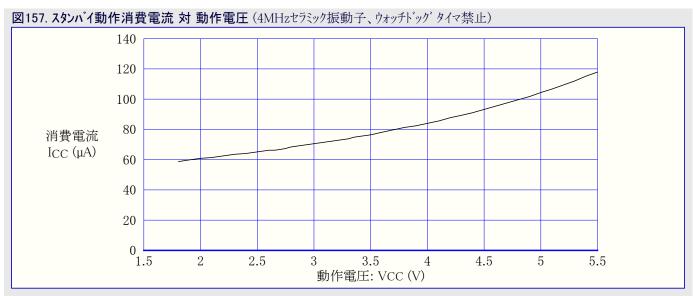


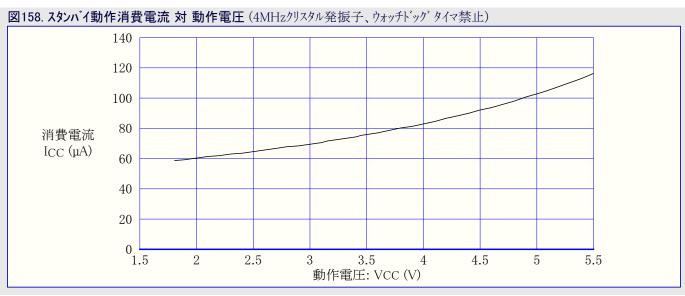


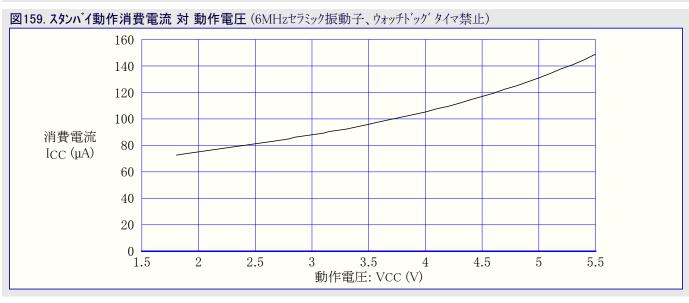






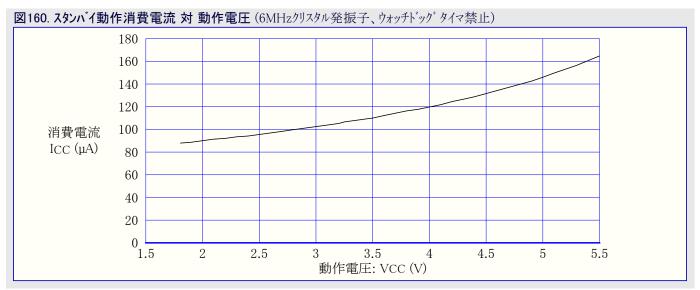




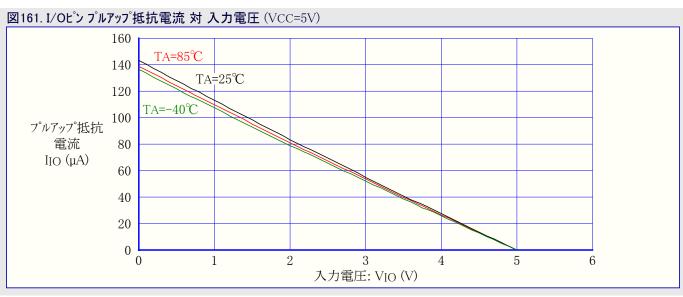


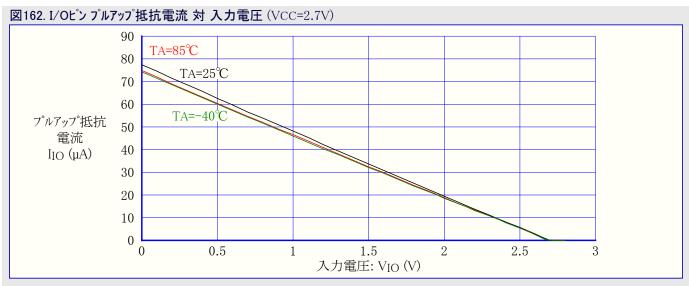


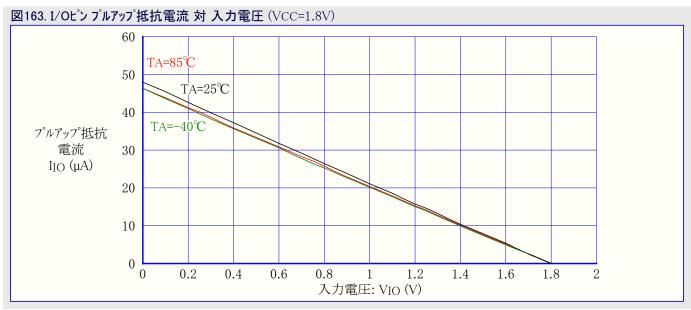


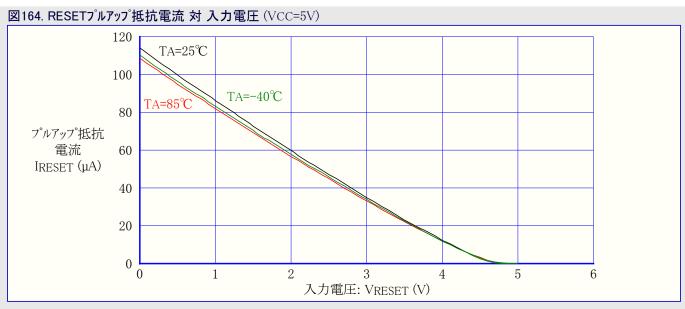


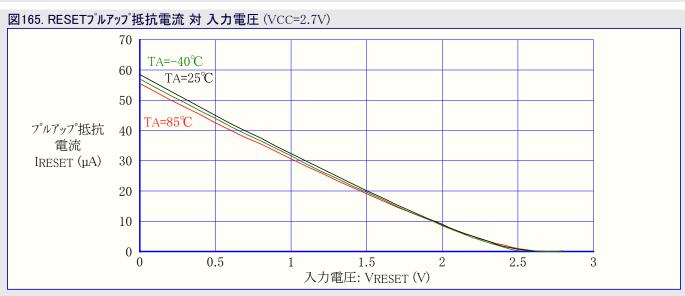
# ピン プルアップ



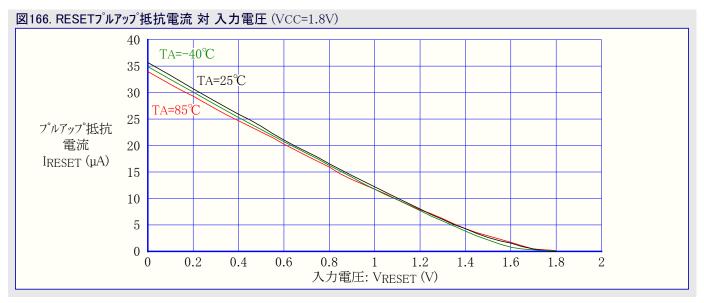




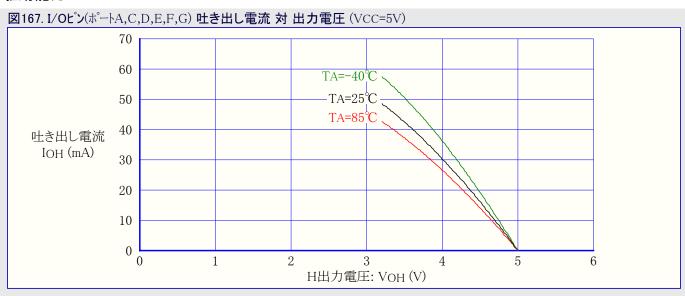


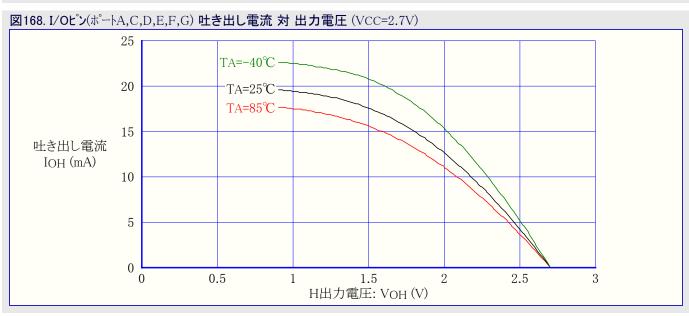


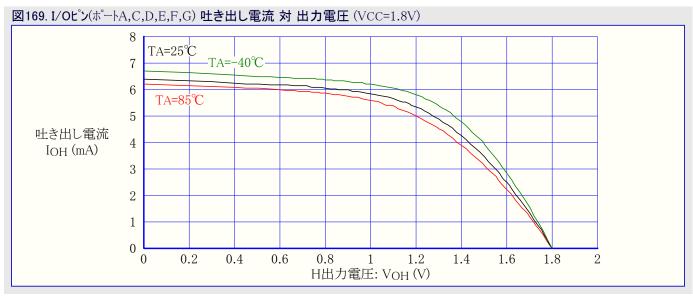


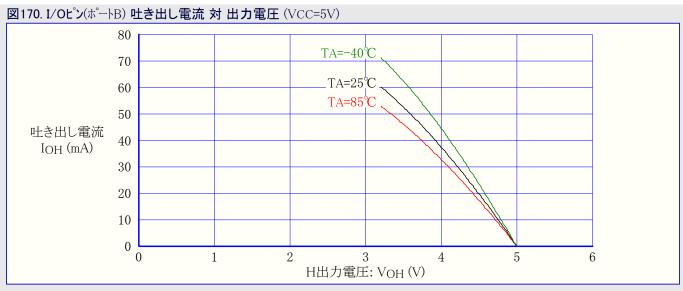


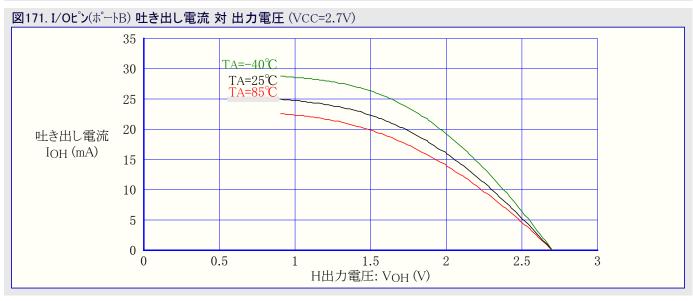
# ピン駆動能力





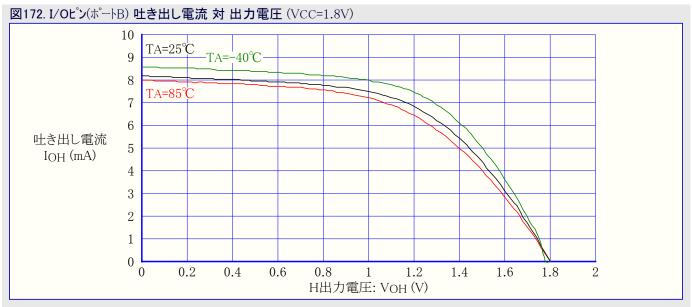


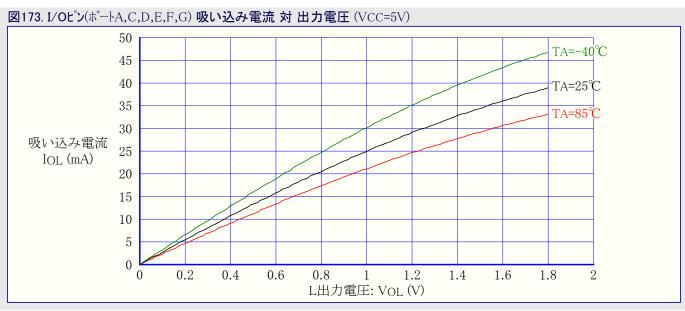


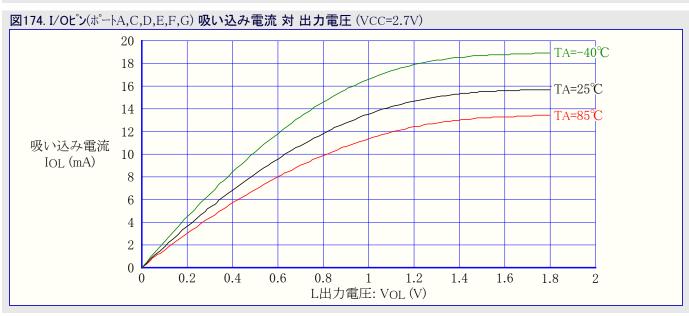


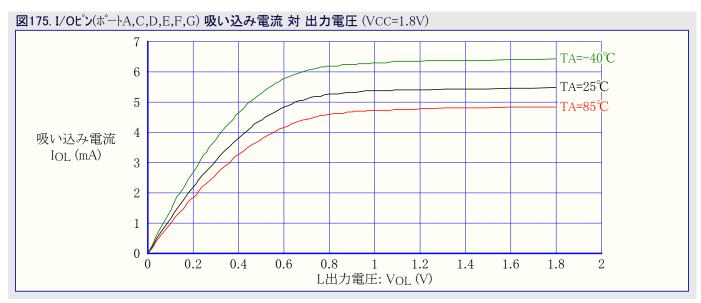


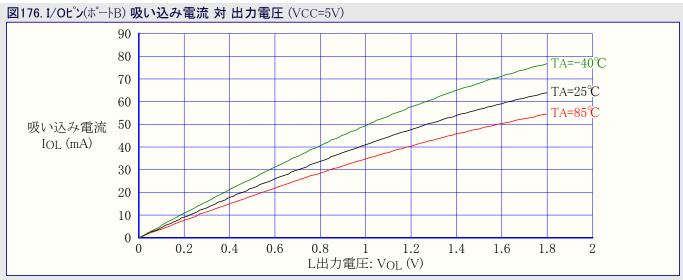


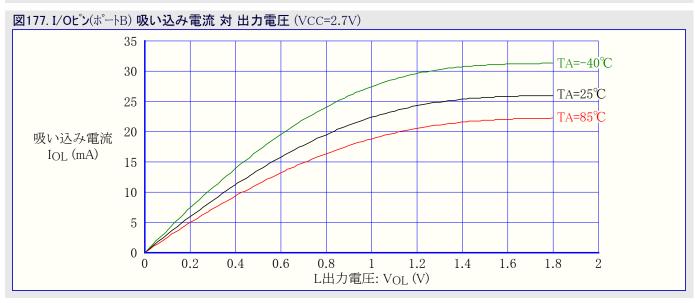






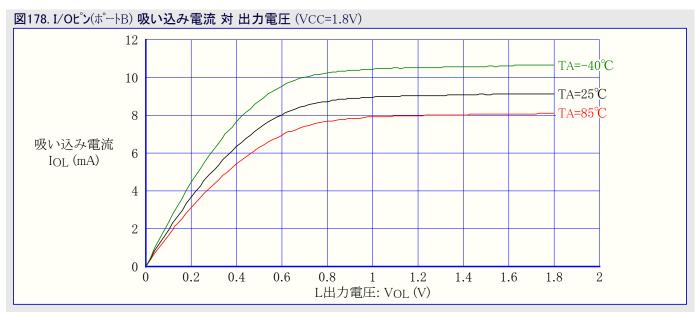




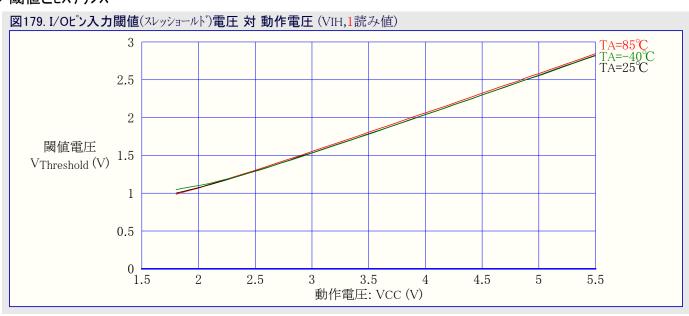


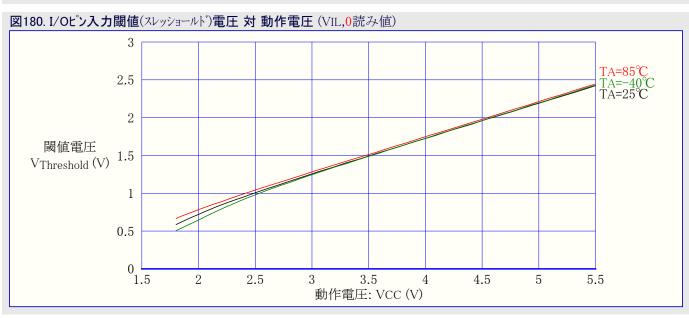


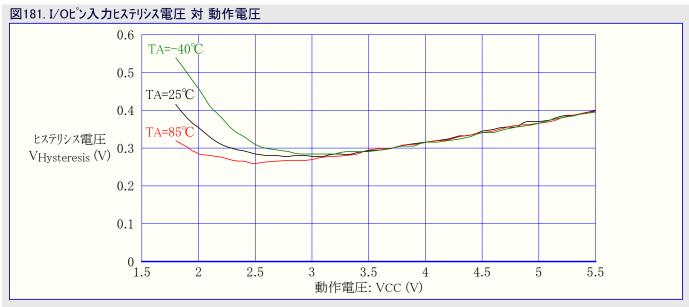


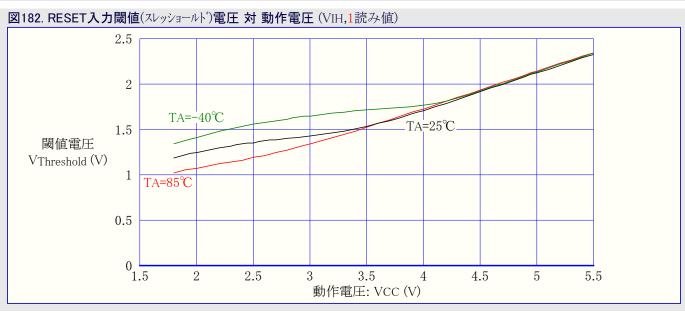


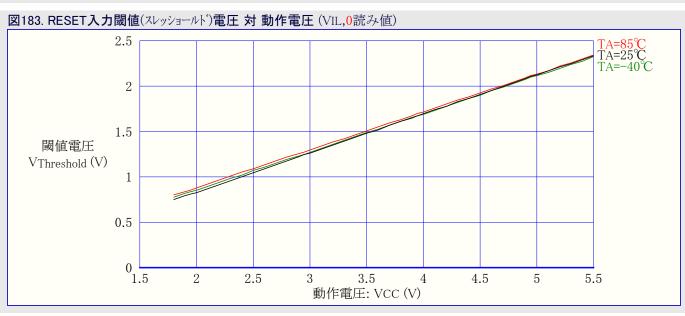
# ピン 閾値とヒステリシス





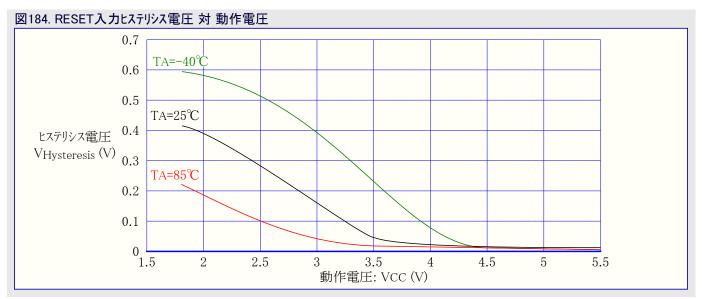




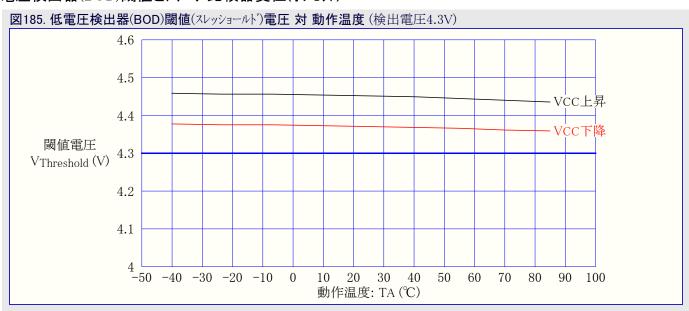


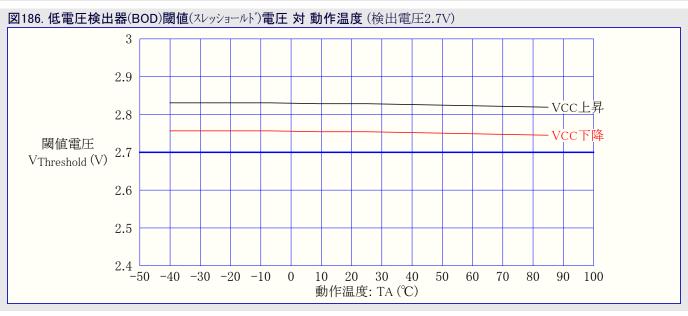


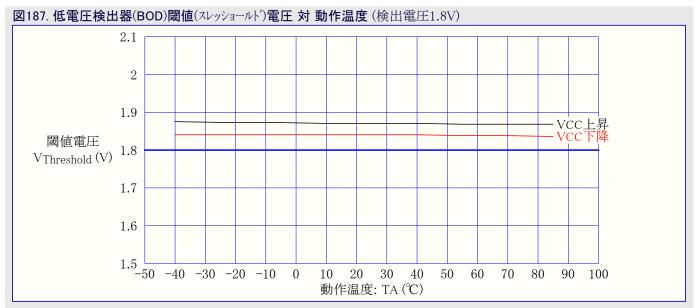


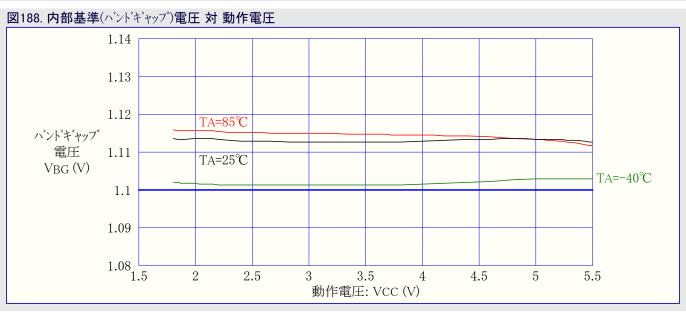


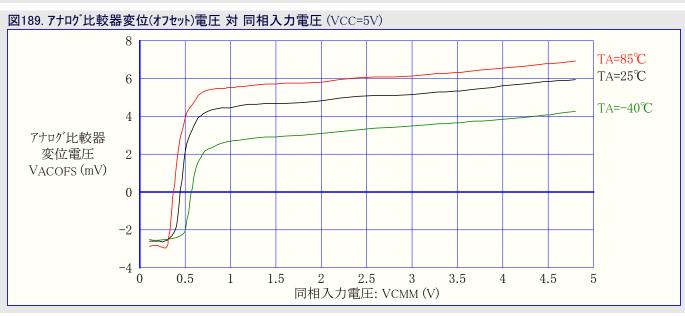
# 低電圧検出器(BOD)閾値とアナログ比較器変位(オフセット)





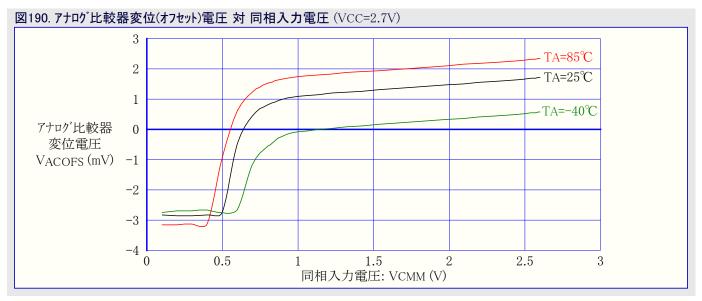




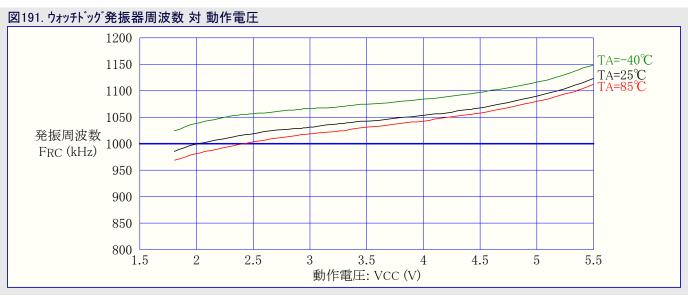


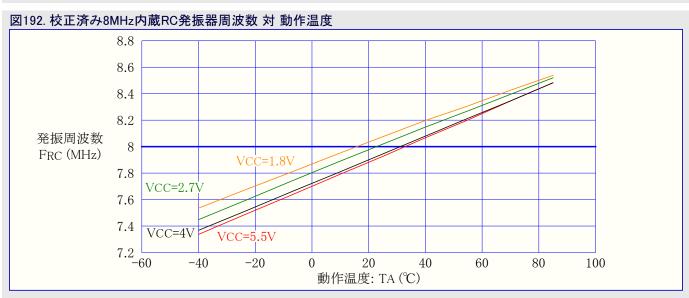


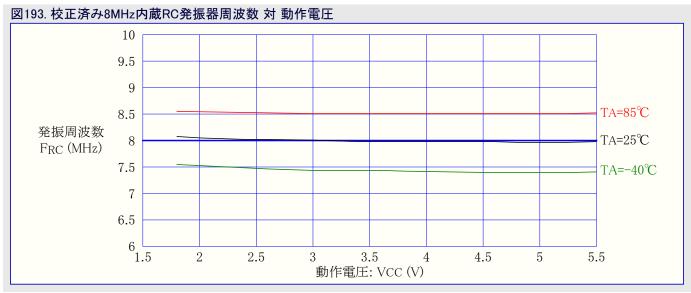


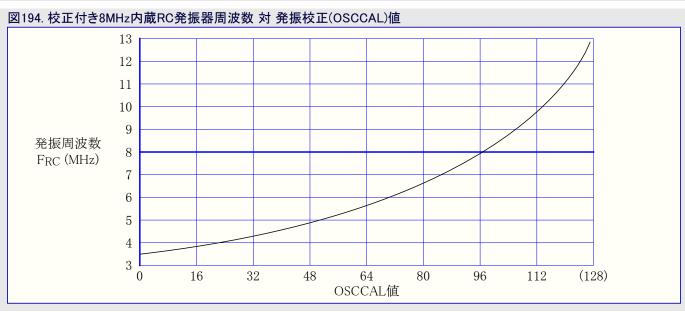


# 内部発振器周波数

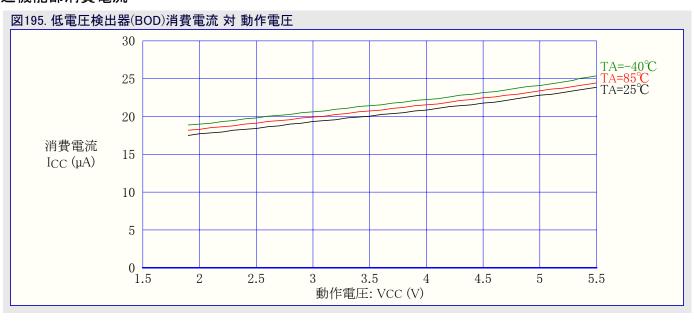






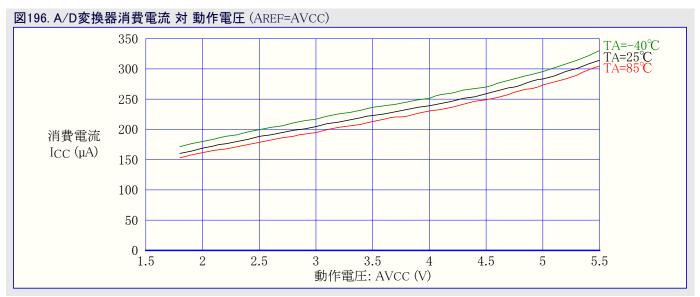


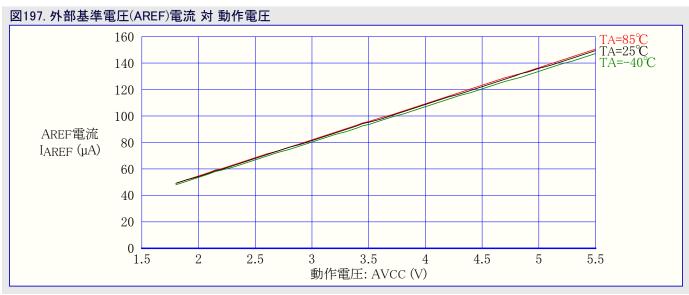
## 周辺機能部消費電流

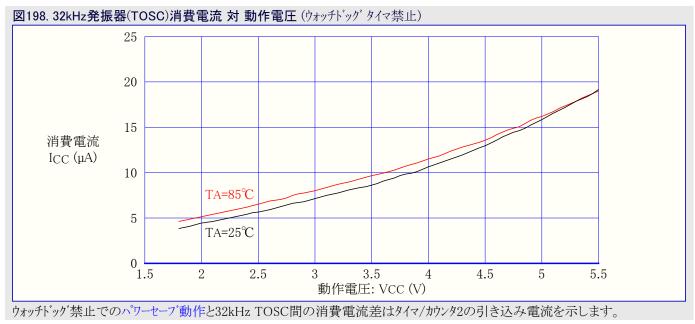


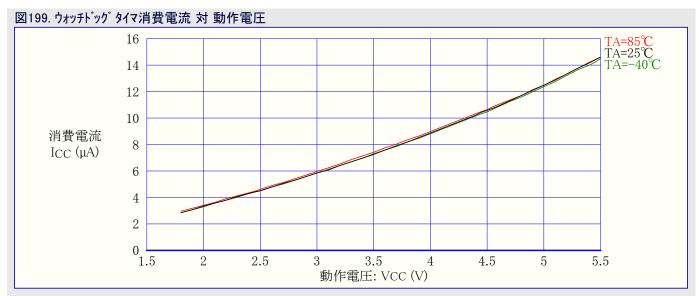


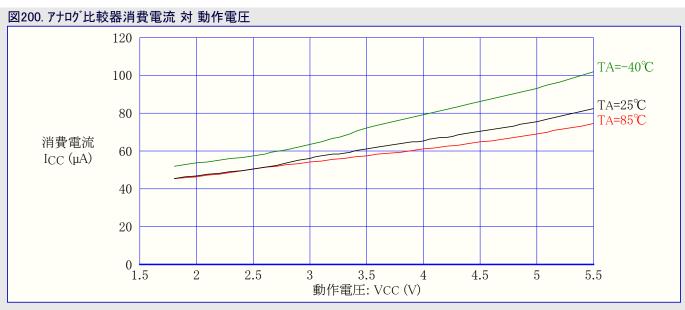


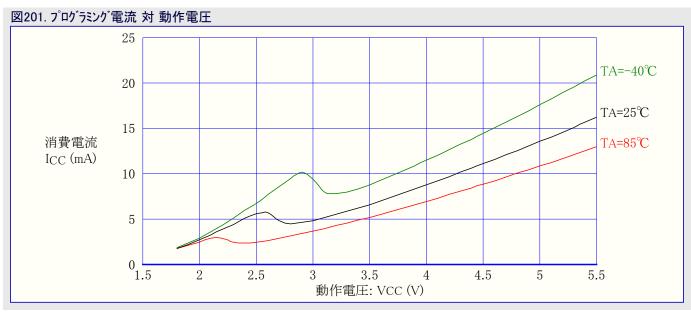








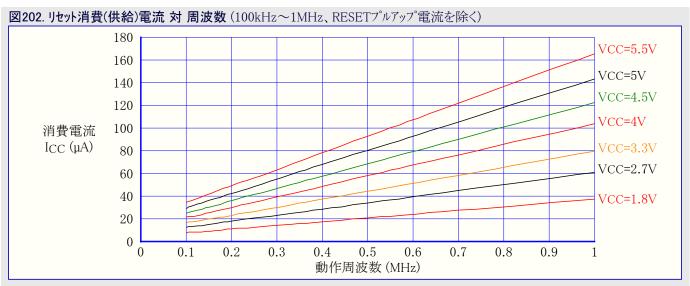


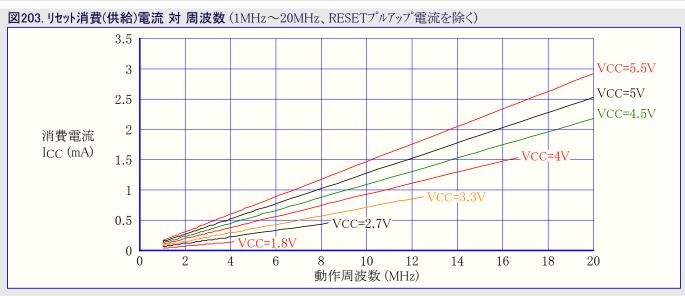


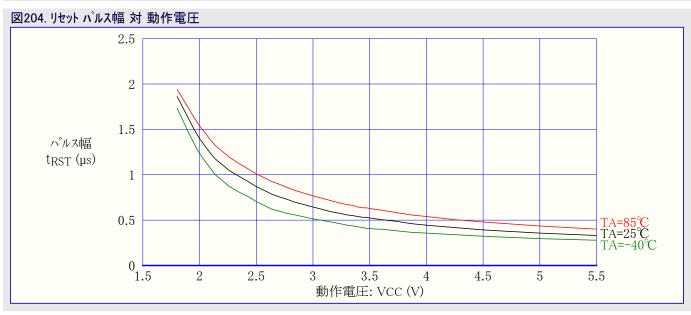




## リセット消費電流とリセットパルス幅







# レジスタ要約

拡張I/Oレシ	<b>バスタ領域</b> (1/	(3)								
アト・レス	レジスタ略称	ピット7	ヒ゛ット6	ヒ゛ット5	ピット4	ピット3	ピット2	ピット1	ピット0	頁
(\$FF)	予約									
(\$FE)	LCDDR18	-	-	-	-	-	-	-	SEG324	151
(\$FD)	LCDDR17	SEG323	SEG322	SEG321	SEG320	SEG319	SEG318	SEG317	SEG316	151
(\$FC)	LCDDR16	SEG315	SEG314	SEG313	SEG312	SEG311	SEG310	SEG309	SEG308	151
(\$FB)	LCDDR15	SEG307	SEG306	SEG305	SEG304	SEG303	SEG302	SEG301	SEG300	151
(\$FA)	予約								CDC004	151
(\$F9)	LCDDR13	- CEC000	- CEC000	- CEC001	- CEC000	- CEC010	- CEC010	SEG217	SEG224	151
(\$F8) (\$F7)	LCDDR12 LCDDR11	SEG223 SEG215	SEG222 SEG214	SEG221 SEG213	SEG220 SEG212	SEG219 SEG211	SEG218	SEG217 SEG209	SEG216 SEG208	151
(\$F7) (\$F6)	LCDDR11	SEG215 SEG207	SEG214 SEG206	SEG213 SEG205	SEG212 SEG204	SEG211 SEG203	SEG210 SEG202	SEG209 SEG201	SEG208 SEG200	151 151
(\$F5)	予約	3EG201	3EG200	3EG200	3EG204	3EG203	3EG202	3EG201	3EG200	101
(\$F4)	LCDDR8	_	_	_	_	_	_	_	SEG124	151
(\$F3)	LCDDR7	SEG123	SEG122	SEG121	SEG120	SEG119	SEG118	SEG117	SEG116	151
(\$F2)	LCDDR6	SEG115	SEG114	SEG113	SEG112	SEG111	SEG110	SEG109	SEG108	151
(\$F1)	LCDDR5	SEG107	SEG106	SEG105	SEG104	SEG103	SEG102	SEG101	SEG100	151
(\$F0)	予約									
(\$EF)	LCDDR3	-	-	-	-	-	-	-	SEG024	151
(\$EE)	LCDDR2	SEG023	SEG022	SEG021	SEG020	SEG019	SEG018	SEG017	SEG016	151
(\$ED)	LCDDR1	SEG015	SEG014	SEG013	SEG012	SEG011	SEG010	SEG009	SEG008	151
(\$EC)	LCDDR0	SEG007	SEG006	SEG005	SEG004	SEG003	SEG002	SEG001	SEG000	151
(\$EB)	予約									
(\$EA)	予約									
(\$E9) (\$E8)	<u>予約</u> 予約									
(\$E0) (\$E7)	LCDCCR	LCDDC2	LCDDC1	LCDDC0	-	LCDCC3	LCDCC2	LCDCC1	LCDCC0	151
(\$E6)	LCDFRR	-	LCDPS2	LCDPS1	LCDPS0	-	LCDCD2	LCDCD1	LCDCD0	150
(\$E5)	LCDCRB	LCDCS	LCD132		LCDMUX0	_	LCDPM2	LCDPM1	LCDPM0	149
(\$E4)	LCDCRA	LCDEN	LCDAB	-	LCDIF	LCDIE	-		LCDBL	149
(\$E3)	予約									
(\$E2)	予約									
(\$E1)	予約									
(\$E0)	予約									
(\$DF)	予約									
(\$DE)	予約									
(\$DD)	予約									
(\$DC)	予約									
(\$DB) (\$DA)	<u>予約</u> 予約									
(\$DA)	予約									
(\$D8)	予約									
(\$D7)	予約									
(\$D6)	予約									
(\$D5)	予約									
(\$D4)	予約									
(\$D3)	予約									
(\$D2)	予約									
(\$D1)	予約									
(\$D0)	予約									
(\$CF)	予約									
(\$CE) (\$CD)	<u>予約</u> 予約									
(\$CD)	予約									
(\$CB)	予約									
(\$CA)	予約									
(\$C9)	予約									
(\$C8)	予約									
(\$C7)	予約									
(\$C6)	UDR				USART デ	ータ レシ・スタ				117
(\$C5)	UBRRH	-	-	-	-		ボーレートレシ゛	スタ上位(UB	RR11∼8)	119
(\$C4)	UBRRL			USART	ボーレート レシ	スタ下位 (UE	RR7~0)			119
(\$C3)	予約									
(\$C2)	UCSRC	-	UMSEL	UPM1	UPM0	USBS	UCSZ1	UCSZ0	UCPOL	118
(\$C1)	UCSRB	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8	118
(\$C0)	UCSRA	RXC	TXC	UDRE	FE	DOR	UPE	U2X	MPCM	117





拡張I/Oレシ	バスタ領域 (2/	(3)								
アトレス	レジスタ略称	ピット7	ヒ゛ット6	ヒ゛ット5	ピット4	ピット3	ピット2	ピット1	ピット0	頁
(\$BF)	予約									
(\$BE)	予約									
(\$BD)	予約									
(\$BC)	予約									
(\$BB)	予約									
(\$BA)	USIDR		'	,		タレジスタ				127
(\$B9)	USISR	USISIF	USIOIF	USIPF	USIDC	USICNT3	USICNT2	USICNT1	USICNT0	127
(\$B8)	USICR	USISIE	USIOIE	USIWM1	USIWM0	USICS1	USICS0	USICLK	USITC	128
(\$B7)	予約									
(\$B6)	ASSR	-	-	-	EXCLK	AS2	TCN2UB	OCR2UB	TCR2UB	97
(\$B5)	予約									
(\$B4)	予約					11 41	-			
(\$B3)	OCR2A			+	タイマ/カウンタ2	比較Aレンス	<del>y</del>	ı———		95
(\$B2)	TCNT2		1	1	タイマ/ )	カウンタ2	l			95
(\$B1)	予約	FOCOA	IVC) 100	COMOA	COM040	WCM01	CCCC	CCO1	CCOo	0.4
(\$B0)	TCCR2A	FOC2A	WGM20	COM2A1	COM2A0	WGM21	CS22	CS21	CS20	94
(\$AF)	予約									
(\$AE) (\$AD)	<u>予約</u> 予約									
(\$AC)	予約									
(\$AC) (\$AB)	予約									
(\$AA)	予約									
(\$A9)	予約									
(\$A8)	予約									
(\$A7)	予約									
(\$A6)	予約									
(\$A5)	予約									
(\$A4)	予約									
(\$A3)	予約									
(\$A2)	予約									
(\$A1)	予約									
(\$A0)	予約									
(\$9F)	予約									
(\$9E)	予約									
(\$9D)	予約									
(\$9C)	予約									
(\$9B) (\$9A)	予約									
(\$9A) (\$99)	<u>予約</u> 予約									
(\$98)	予約									
(\$97)	予約									
(\$96)	予約									
(\$95)	予約									
(\$94)	予約									
(\$93)	予約									
(\$92)	予約									
(\$91)	予約									
(\$90)	予約									
(\$8F)	予約									
(\$8E)	予約									
(\$8D)	予約									
(\$8C)	予約			ļ , , .	(1.10.20.11.11	D	* * * *			
(\$8B)	OCR1BH				カウンタ1 比較					84
(\$8A)	OCR1BL				カウンタ1 比較					
(\$89)	OCR1AH		-		カウンタ1 比較					84
(\$88) (\$87)	OCR1AL ICR1H		-	717/	カウンタ1 比較 /カウンタ1 捕豕	エスプラカ しん	L/\1\			
(\$87)	ICR1H ICR1L		-	717	/カリンタ1 拥加 /カウンタ1 捕狐	ラレン ヘク 上位 ギレンバフカ 下 <i>に</i>	ハイト			84
(\$86)	TCNT1H		1	717		<u> </u>	, 11,	-		
(\$84)	TCNT1H TCNT1L		+	+	+ タイマ / カリング	<u>1 上位ハイト</u> 1 下位バイト	-	-		83
(\$83)	予約				7711/1/1927	T 1 1777, 711,				
(\$82)	TCCR1C	FOC1A	FOC1B	-	-	-	-	-	-	83
(\$81)	TCCR1B	ICNC1	ICES1	-	WGM13	WGM12	CS12	CS11	CS10	82
(\$80)	TCCR1A	COM1A1	COM1A0	COM1B1	COM1B0	-	-	WGM11	WGM10	81
(+00)										- *

拡張I/Oレシ	<b>、スタ領域</b> (3/	3)								
アト・レス	レジスタ略称	ピット7	ピット6	ヒ゛ット5	ピット4	ヒット3	ピット2	ピット1	ピット0	頁
(\$7F)	DIDR1	_	-	-	-	-	_	AIN1D	AIN0D	131
(\$7E)	DIDR0	ADC7D	ADC6D	ADC5D	ADC4D	ADC3D	ADC2D	ADC1D	ADC0D	141
(\$7D)	予約									
(\$7C)	ADMUX	REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0	139
(\$7B)	ADCSRB	-	ACME	-	-	-	ADTS2	ADTS1	ADTS0	131,140
(\$7A)	ADCSRA	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	140
(\$79)	ADCH		. A/	/Dテ゛ータ レシ゛	な上位バイト	(ADC9~8ま	たはADC9~	(2)		141
(\$78)	ADCL		. A	Dテータ レシン	な下位バイト	(ADC7~0ま	たはADC1~	(0)		141
(\$77)	予約									
(\$76)	予約									
(\$75)	予約									
(\$74)	予約									
(\$73)	予約									
(\$72)	予約									
(\$71)	予約									
(\$70)	TIMSK2	_	_	-	-	-	-	OCIE2A	TOIE2	96
(\$6F)	TIMSK1	_	_	ICIE1	-	-	OCIE1B	OCIE1A	TOIE1	85
(\$6E)	TIMSK0	_	_	-	-	-	-	OCIE0A	TOIE0	64
(\$6D)	予約									
(\$6C)	PCMSK1	PCINT15	PCINT14	PCINT13	PCINT12	PCINT11	PCINT10	PCINT9	PCINT8	36
(\$6B)	PCMSK0	PCINT7	PCINT6	PCINT5	PCINT4	PCINT3	PCINT2	PCINT1	PCINT0	36
(\$6A)	予約									
(\$69)	EICRA	-	-	-	-	-	-	ISC01	ISC00	34
(\$68)	予約									
(\$67)	予約									
(\$66)	OSCCAL	_			内蔵RC発掘	辰器 発振校	正値レジスタ			18
(\$65)	予約									
(\$64)	PRR	-	-	-	PRLCD	PRTIM1	PRSPI	PRUSART0	PRADC	23
(\$63)	予約									
(\$62)	予約									
(\$61)	CLKPR	CLKPCE	-	-	-	CLKPS3	CLKPS2	CLKPS1	CLKPS0	20
(\$60)	WDTCR	-	-	-	WDCE	WDE	WDP2	WDP1	WDP0	28

- 注意: 将来のデバイスとの共通性のため、アクセスされる場合の予約ビットは0を書かれるべきです。予約したI/Oメモリ アドレスは決して書かれるべきではありません。
  - アドレス範囲\$00~\$1F内のI/OレジスタはSBIとCBI命令を使う直接ビット アクセスが可能です。これらのレジスタではSBISとSBIC命令を使うことによって単一ビット値が検査できます。
  - いくつかの状態ビットはそれらへ論理1を書くことによって解除(0)されます。他の多くのAVRと異なり、CBIとSBI命令は指定ビットだけ操作し、故にこのような状態フラグを含むレジスタで使えます。CBIとSBI命令は\$00~\$1Fのレジスタだけで動作します。
  - I/O指定命令INとOUTを使う時はI/Oアドレス\$00~\$3Fが使われなければなりません。LDとST命令を使ってデータ空間として I/Oレジスタをアドレス指定する時はこれらのアドレスに\$20が加算されなければなりません。ATmega169はINとOUT命令で予約した64位置で支援できるよりも多くの周辺部(機能)の複合マイクロ コントローラです。SRAM(データ空間)内の拡張I/O空間はST/STS/STDとLD/LDS/LDD命令だけが使えます。





マレ゛コ	しょごっち四々また	۲°۱ ت	<b>▶</b> " <b>!</b> ∩	▶"L.E	<b>▶</b> * <b>!</b> 4	۲°۱۵	ب"ا ۵	<b>▶</b> *l 1	F,""  0	<b>-</b>
アトレス	レジスタ略称	<u> ビットフ</u>	<u> ピット6</u>	<u> ビット5</u>	<u>L'y\4</u>	ピット3	<u> </u>	<u> ビット1</u>	ピット0	<u>頁</u>
\$3F (\$5F)	SREG	- -	T	H -	S -	- V	N	Z	C	7
3E (\$5E)	SPH						SP10	SP9	SP8	8
3D (\$5D)	SPL	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	
3C (\$5C)	予約									
3B (\$5B)	予約									
3A (\$5A)	予約									
\$39 (\$59)	予約									
\$38 (\$58)	予約									
\$37 (\$57)	SPMCSR	SPMIE	RWWSB	-	RWWSRE	BLBSET	PGWRT	PGERS	SPMEN	169
\$36 (\$56)	予約									
\$35 (\$55)	MCUCR	JTD	-	-	PUD	-	ı	IVSEL	IVCE	158,38,33
\$34 (\$54)	MCUSR	-	-	-	JTRF	WDRF	BORF	EXTRF	PORF	158,27
33 (\$53)	SMCR	ı	-	-	-	SM2	SM1	SM0	SE	21
\$32 (\$52)	予約									
\$31 (\$51)	OCDR	IDRD/	OCDR6	OCDR5	OCDR4	OCDR3	OCDR2	OCDR1	OCDR0	154
		OCDR7								
\$30 (\$50)	ACSR	ACD	ACBG	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	130
32F (\$4F)	予約									
32E (\$4E)	SPDR				SPI デー	タレジスタ				103
2D (\$4D)	SPSR	SPIF	WCOL	-	-	-	-	-	SPI2X	103
2C (\$4C)	SPCR	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	102
2B (\$4B)	GPIOR2					レシ スタ2		-		15
2A (\$4A)	GPIOR1				汎用I/C	レシ スタ1				15
\$29 (\$49)	予約									
\$28 (\$48)	予約									
327 (\$47)	OCR0A			5	イマ/カウンタ0		7			63
326 (\$46)	TCNT0				タイマ/フ	カウンタ0				63
325 (\$45)	予約									
324 (\$44)	TCCR0A	FOC0A	WGM00	COM0A1	COM0A0	WGM01	CS02	CS01	CS00	62
\$23 (\$43)	GTCCR	TSM	-	-	-	-	-	PSR2	PSR10	99,66
\$22 (\$42)	EEARH	-	-	-	-	-	-	-	EEAR8	12
\$21 (\$41)	EEARL			EEPROM	アドレス レシブスタ		EEAR7 $\sim$ 0)	<u> </u>		
\$20 (\$40)	EEDR		<u> </u>		EEPROM7	・・ータ レシ・スタ				12
31F (\$3F)	EECR	-	-	-	-	EERIE	EEMWE	EEWE	EERE	12
31E (\$3E)	GPIOR0				. 汎用I/C	シンシンタ0				15
1D (\$3D)	EIMSK	PCIE1	PCIE0	-	-	-	-	-	INT0	35
1C (\$3C)	EIFR	PCIF1	PCIF0	-	-	_	1	-	INTF0	35
31B (\$3B)	予約									
1A (\$3A)	予約									
\$19 (\$39)	予約									
\$18 (\$38)	予約									
\$17 (\$37)	TIFR2	-	-	-	-	-	-	OCF2A	TOV2	96
316 (\$36)	TIFR1	-	-	ICF1	-	-	OCF1B	OCF1A	TOV1	85
315 (\$35)	TIFR0	-	-	-	-	-	П	OCF0A	TOV0	64
314 (\$34)	PORTG	-	-	-	PORTG4	PORTG3	PORTG2	PORTG1	PORTG0	53
313 (\$33)	DDRG	-	-	_	DDG4	DDG3	DDG2	DDG1	DDG0	53
312 (\$32)	PING	-	-	_	PING4	PING3	PING2	PING1	PING0	53
\$11 (\$31)	PORTF	PORTF7	PORTF6	PORTF5	PORTF4	PORTF3	PORTF2	PORTF1	PORTF0	52
\$10 (\$30)	DDRF	DDF7	DDF6	DDF5	DDF4	DDF3	DDF2	DDF1	DDF0	53
30F (\$2F)	PINF	PINF7	PINF6	PINF5	PINF4	PINF3	PINF2	PINF1	PINF0	53
0E (\$2E)	PORTE	PORTE7	PORTE6	PORTE5	PORTE4	PORTE3	PORTE2	PORTE1	PORTE0	52
0D (\$2D)	DDRE	DDE7	DDE6	DDE5	DDE4	DDE3	DDE2	DDE1	DDE0	52
0C (\$2C)	PINE	PINE7	PINE6	PINE5	PINE4	PINE3	PINE2	PINE1	PINE0	52
0B (\$2B)	PORTD	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	52
0A (\$2A)	DDRD	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	52
309 (\$29)	PIND	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	52
308 (\$28)	PORTC	PORTC7	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	51
307 (\$27)	DDRC	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	51
\$06 (\$26)	PINC	PINC7	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0	52
\$05 (\$25)	PORTB	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	51
\$04 (\$24)	DDRB	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	51
303 (\$23)	PINB	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	51
\$02 (\$22)	PORTA	PORTA7	PORTA6	PORTA5	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0	51
201 (401)	DDRA	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	51
301 (\$21) 300 (\$20)	PINA	PINA7	PINA6	PINA5	PINA4	PINA3	PINA2	PINA1	PINA0	51

(訳注) 原書本位置の注意は前頁に移動しました。

## 命令要約

ニーモニック	オペラント゛		動作	フラク゛	クロック
			理演算命令		
ADD	Rd,Rr	汎用レシブスタ間の加算	$Rd \leftarrow Rd + Rr$	I,T,H,S,V,N,Z,C	
ADC	Rd,Rr	キャリーを含めた汎用レジスタ間の加算	$Rd \leftarrow Rd + Rr + C$	I,T,H,S,V,N,Z,C	
ADIW	Rd,K6	即値の語(ワード)長加算	RdH:RdL ← RdH:RdL + K6	I,T,H,S,V,N,Z,C	2
SUB	Rd,Rr	汎用レジスタ間の減算	$Rd \leftarrow Rd - Rr$	I,T,H,S,V,N,Z,C	1
SUBI	Rd,K	汎用レジスタから即値の減算	$Rd \leftarrow Rd - K$	I,T,H,S,V,N,Z,C	1
SBIW	Rd,K6	即値の語(ワード)長減算	RdH:RdL ← RdH:RdL - K6	I,T,H,S,V,N,Z,C	
SBC	Rd,Rr	キャリーを含めた汎用レジスタ間の減算	$Rd \leftarrow Rd - Rr - C$	I,T,H,S,V,N,Z,C	1
SBCI	Rd,K	汎用レジスタからキャリーと即値の減算	$Rd \leftarrow Rd - K - C$	I,T,H,S,V,N,Z,C	1
AND	Rd,Rr	汎用レジスタ間の論理積(AND)	Rd ← Rd AND Rr	I,T,H,S,0,N,Z,C	
ANDI	Rd,K	汎用レジスタと即値の論理積(AND)	Rd ← Rd AND K	I,T,H,S,0,N,Z,C	1
OR	Rd,Rr	汎用レジスタ間の論理和(OR)	Rd ← Rd OR Rr	I,T,H,S,0,N,Z,C	1
ORI	Rd,K	汎用レジスタと即値の論理和(OR)	Rd ← Rd OR K	I,T,H,S,0,N,Z,C	
EOR	Rd,Rr	汎用レジスタ間の排他的論理和(Ex-OR)	Rd ← Rd EOR Rr	I,T,H,S,0,N,Z,C	
COM	Rd	1の補数(論理反転)	Rd ← \$FF - Rd	I,T,H,S,0,N,Z,C	
NEG	Rd	2の補数	Rd ← \$00 - Rd	I,T,H,S,V,N,Z,C	
SBR	Rd,K	汎用レジスタの(複数)ビット設定(1)	$Rd \leftarrow Rd \ OR \ K$	I,T,H,S,0,N,Z,C	1
CBR	Rd,K	汎用レジスタの(複数)ビット解除(0)	$Rd \leftarrow Rd \ AND \ (\$FF - K)$	I,T,H,S,0,N,Z,C	1
INC	Rd	汎用レジスタの増加(+1)	$Rd \leftarrow Rd + 1$	I,T,H,S,V,N,Z,C	
DEC	Rd	汎用レジスタの減少(-1)	$Rd \leftarrow Rd - 1$	I,T,H,S,V,N,Z,C	1
TST	Rd	汎用レジスタのセゴロとマイナス検査	Rd ← Rd AND Rd	I,T,H,S,0,N,Z,C	1
CLR	Rd	汎用レジスタの全0設定(=\$00)	Rd ← Rd EOR Rd	I,T,H,0,0,0,1,C	1
SER	Rd	汎用レジスタの全1設定(=\$FF)	Rd ← \$FF	I,T,H,S,V,N,Z,C	1
MUL	Rd,Rr	符号なし間の乗算	$R1:R0 \leftarrow Rd \times Rr$ $(U \times U)$	I,T,H,S,V,N,Z,C	2
MULS	Rd,Rr	符号付き間の乗算	$R1:R0 \leftarrow Rd \times Rr$ (S×S)	I,T,H,S,V,N,Z,C	2
MULSU	Rd,Rr	符号付きと符号なしの乗算	$R1:R0 \leftarrow Rd \times Rr$ (S×U)	I,T,H,S,V,N,Z,C	2
FMUL	Rd,Rr	符号なし間の固定小数点乗算	$R1:R0 \leftarrow (Rd \times Rr) << 1 \qquad (U \times U)$	I,T,H,S,V,N,Z,C	2
FMULS	Rd,Rr	符号付き間の固定小数点乗算	$R1:R0 \leftarrow (Rd \times Rr) << 1 \qquad (S \times S)$	I,T,H,S,V,N,Z,C	2
FMULSU	Rd,Rr	符号付きと符号なしの固定小数点乗算	$R1:R0 \leftarrow (Rd \times Rr) << 1 \qquad (S \times U)$	I,T,H,S,V,N,Z,C	2
			支命令	_	
RJMP	k	相対無条件分岐	$PC \leftarrow PC + k + 1$	I,T,H,S,V,N,Z,C	2
IJMP		Zレジスタ間接無条件分岐	PC ← Z	I,T,H,S,V,N,Z,C	2
JMP	k	絶対無条件分岐	PC ← k	I,T,H,S,V,N,Z,C	3
RCALL	k	相対サブルーチン呼び出し	$STACK \leftarrow PC, PC \leftarrow PC + k + 1$	I,T,H,S,V,N,Z,C	3
ICALL		Zレシ、スタ間接サフ・ルーチン呼び出し	$STACK \leftarrow PC, PC \leftarrow Z$	I,T,H,S,V,N,Z,C	3
CALL	k	絶対サブルーチン呼び出し	STACK ← PC, PC ← k	I,T,H,S,V,N,Z,C	4
RET		サブルーチンからの復帰	PC ← STACK	I,T,H,S,V,N,Z,C	4
RETI		割り込みからの復帰	PC ← STACK	1,T,H,S,V,N,Z,C	4
CPSE	Rd,Rr	汎用レシ、スタ間比較、一致でスキップ	Rd=Rrなら、PC ← PC + 2or3	I,T,H,S,V,N,Z,C	1/2,3
CP	Rd,Rr	汎用レジスタ間の比較	Rd - Rr	I,T,H,S,V,N,Z,C	1
CPC	Rd,Rr	キャリーを含めた汎用レジスタ間の比較	Rd - Rr - C	I,T,H,S,V,N,Z,C	1
CPI	Rd,K	汎用レジスタと即値の比較	Rd - K	I,T,H,S,V,N,Z,C	1
SBRC	Rr,b	汎用レシ、スタのヒ、ットが解除(0)でスキップ。	Rr(b)=0なら, PC ← PC + 2or3	I,T,H,S,V,N,Z,C	1/2,3
SBRS	Rr,b	汎用レシ、スタのヒ、ットが設定(1)でスキップ	Rr(b)=1なら, PC ← PC + 2or3	I,T,H,S,V,N,Z,C	1/2,3
SBIC	P,b	I/Oレシ、スタのヒ、ットが解除(0)でスキップ。	P(b)=0735, PC ← PC + 2or3	I,T,H,S,V,N,Z,C	
SBIS	P,b	I/Oレシ、スタのヒットが設定(1)でスキップ°	P(b)=1なら, PC ← PC + 2or3	I,T,H,S,V,N,Z,C	
BRBS	s,k	ステータス フラグが設定(1)で分岐	SREG(s)=17\$6, PC ← PC + K + 1	I,T,H,S,V,N,Z,C	
BRBC	s,k	ステータス フラグが解除(0)で分岐	SREG(s)=0なら、PC ← PC + K + 1	I,T,H,S,V,N,Z,C	
BREQ	k	一致で分岐	Z=1766, PC ← PC + K + 1	I,T,H,S,V,N,Z,C	
BRNE	k	不一致で分岐	Z=07\$\$, PC ← PC + K + 1	I,T,H,S,V,N,Z,C	
BRCS	k	キャリー フラグが設定(1)で分岐	C=1なら, PC ← PC + K + 1	I,T,H,S,V,N,Z,C	
BRCC	k	キャリー フラグが解除(0)で分岐	C=07\$\$, PC ← PC + K + 1	I,T,H,S,V,N,Z,C	_
BRSH	k	符号なしの≧で分岐	C=0なら, PC ← PC + K + 1	I,T,H,S,V,N,Z,C	
BRLO	k	符号なしのくで分岐	C=1なら, PC ← PC + K + 1	I,T,H,S,V,N,Z,C	
BRMI	k	-(マイナス)で分岐	N=1なら, PC ← PC + K + 1	I,T,H,S,V,N,Z,C	
BRPL	k	+(プラス)で分岐	N=07\$6, PC ← PC + K + 1	I,T,H,S,V,N,Z,C	
BRGE	k	符号付きの≧で分岐	(N EOR V)=0なら, PC ← PC + K + 1	I,T,H,S,V,N,Z,C	
BRLT	k	符号付きのくで分岐	(N EOR V)=1&6, PC ← PC + K + 1	I,T,H,S,V,N,Z,C	_
BRHS	k	ハーフキャリー フラグが設定(1)で分岐	H=1なら, PC ← PC + K + 1	I,T,H,S,V,N,Z,C	
BRHC	k	ハーフキャリー フラグが解除(0)で分岐	H=0735, PC ← PC + K + 1	I,T,H,S,V,N,Z,C	
BRTS	k	一時フラグが設定(1)で分岐	T=17x6, PC ← PC + K + 1	I,T,H,S,V,N,Z,C	
BRTC	k	一時777 が解除(1)で分岐	T=07&6, PC ← PC + K + 1	I,T,H,S,V,N,Z,C	
BRVS	k	2の補数溢れフラグが設定(1)で分岐	V=17x6, PC ← PC + K + 1	I,T,H,S,V,N,Z,C	
BRVC	k	2の補数溢れフラグが解除(0)で分岐	$V=0$ 7 $c$ 5, $PC \leftarrow PC + K + 1$	I,T,H,S,V,N,Z,C	
BRIE	k	割り込み許可で分岐	V-0/46, PC ← PC + K + 1 I=1なら, PC ← PC + K + 1	I,T,H,S,V,N,Z,C	
BRID	k	割り込み許可で分岐	I=0/36, PC ← PC + K + 1	I, T, H, S, V, N, Z, C	
סואם			I=0/より, FC 、 FC + K + I		1/4

K6, K: 6, 8ビット定数 P: I/Oレジスタ Rd, Rr: 汎用レジスタ(R0~R31) X, Y, Z: X, Y, Zレジスタ

b: ビット(0~7) k: アドレス定数(7,12,16ビット) q: 符号なし6ビット定数(変位) s: ステータス フラグ(C,Z,N,V,X,H,T,I)





ニーモニック	オペラント゛	意味	動作	フラク゛	クロック
_ L_//	1/1 //	<del></del>		///	7477
MOV	Rd,Rr	汎用レジスタ間の複写	Rd ← Rr	I,T,H,S,V,N,Z,C	1
MOVW	Rd,Rr	汎用レジスタ対間の複写	Rd+1:Rd ← Rr+1:Rr	I,T,H,S,V,N,Z,C	1
LDI	Rd,K	即値の取得	$Rd \leftarrow K$	I,T,H,S,V,N,Z,C	1
LD	Rd,X	Xレジスタ間接での取得	$Rd \leftarrow (X)$	I,T,H,S,V,N,Z,C	2
LD LD	Rd,X+ Rd,-X	事後増加付きXレジスタ間接での取得 事前減少付きXレジスタ間接での取得	$Rd \leftarrow (X), X \leftarrow X + 1  X \leftarrow X - 1, Rd \leftarrow (X)$	I,T,H,S,V,N,Z,C I,T,H,S,V,N,Z,C	2 2
LD	Rd, X	Yレジスタ間接での取得	$Rd \leftarrow (Y)$	I,T,H,S,V,N,Z,C	2
LD	Rd,Y+	事後増加付きYレジスタ間接での取得	$Rd \leftarrow (Y), Y \leftarrow Y + 1$	I,T,H,S,V,N,Z,C	2
LD	Rd,-Y	事前減少付きYレジスタ間接での取得	$Y \leftarrow Y - 1$ , $Rd \leftarrow (Y)$	I,T,H,S,V,N,Z,C	2
LDD	Rd,Y+q	変位付きYレジスタ間接での取得	$Rd \leftarrow (Y + q)$	I,T,H,S,V,N,Z,C	2
LD	Rd,Z	Zレジスタ間接での取得	$Rd \leftarrow (Z)$	I,T,H,S,V,N,Z,C	2
LD	Rd,Z+	事後増加付きZレジスタ間接での取得	$Rd \leftarrow (Z), Z \leftarrow Z + 1$	I,T,H,S,V,N,Z,C	2 2
LD LDD	Rd,-Z Rd,Z+q	事前減少付きZレジスタ間接での取得 変位付きZレジスタ間接での取得	$Z \leftarrow Z - 1$ , $Rd \leftarrow (Z)$ $Rd \leftarrow (Z + q)$	I,T,H,S,V,N,Z,C I,T,H,S,V,N,Z,C	2
LDS	Rd,k	データ空間(SRAM)から直接取得	$Rd \leftarrow (k)$	I,T,H,S,V,N,Z,C	2
ST	X,Rr	Xレジスタ間接での設定	$(X) \leftarrow Rr$	I,T,H,S,V,N,Z,C	2
ST	X+,Rr	事後増加付きXレシブスタ間接での設定	$(X) \leftarrow Rr, X \leftarrow X + 1$	I,T,H,S,V,N,Z,C	2
ST	-X,Rr	事前減少付きXレジスタ間接での設定	$X \leftarrow X - 1, (X) \leftarrow Rr$	I,T,H,S,V,N,Z,C	2
ST	Y,Rr	Yレジスタ間接での設定	(Y) ← Rr	I,T,H,S,V,N,Z,C	2
ST	Y+,Rr	事後増加付きYレジスタ間接での設定	$(Y) \leftarrow Rr, Y \leftarrow Y + 1$	I,T,H,S,V,N,Z,C	2
ST STD	-Y,Rr Y+q,Rr	事前減少付きYレジスタ間接での設定 変位付きYレジスタ間接での設定	$Y \leftarrow Y - 1, (Y) \leftarrow Rr$ $(Y + q) \leftarrow Rr$	I,T,H,S,V,N,Z,C	2 2
STD	Z,Rr	変位付さなレンスタ間接での設定   Zレジスタ間接での設定	$(Y + q) \leftarrow Rr$ $(Z) \leftarrow Rr$	I,T,H,S,V,N,Z,C I,T,H,S,V,N,Z,C	2
ST	Z+,Rr	事後増加付きZレジスタ間接での設定	$(Z) \leftarrow Rr$ $(Z) \leftarrow Rr, Z \leftarrow Z + 1$	I,T,H,S,V,N,Z,C	2
ST	-Z,Rr	事前減少付きZレジスタ間接での設定	$Z \leftarrow Z - 1$ , $(Z) \leftarrow Rr$	I,T,H,S,V,N,Z,C	2
STD	Z+q,Rr	変位付きZレジスタ間接での設定	$(Z + q) \leftarrow Rr$	I,T,H,S,V,N,Z,C	2
STS	k,Rr	データ空間(SRAM)へ直接設定	$(k) \leftarrow Rr$	I,T,H,S,V,N,Z,C	2
LPM		プログラム領域からZレジスタ間接での取得	$RO \leftarrow (Z)$	I,T,H,S,V,N,Z,C	3
LPM	Rd,Z	同上(任意のレジスタへ)	$Rd \leftarrow (Z)$	I,T,H,S,V,N,Z,C	3
LPM SPM	Rd,Z+	同上(事後増加付き) プログラム領域へZレジスタ間接での設定	$Rd \leftarrow (Z), Z \leftarrow Z + 1$ $(Z) \leftarrow R1:R0$	I,T,H,S,V,N,Z,C	3
IN	Rd,P	I/OVジスタからの入力	(Z) ← R1:R0 Rd ← P	I,T,H,S,V,N,Z,C I,T,H,S,V,N,Z,C	1
OUT	P,Rr	I/Oレジスタへの出力	P ← Rr	I,T,H,S,V,N,Z,C	1
PUSH	Rr	汎用レジブスタをスタックへ保存	STACK ← Rr	I,T,H,S,V,N,Z,C	2
POP	Rd	スタックから汎用レジスタ〜復帰	Rd ← STACK	I,T,H,S,V,N,Z,C	2
			係命令		_
SBI	P,b	I/Oレシブスタのビット設定(1)	$I/O(P,b) \leftarrow 1$	I,T,H,S,V,N,Z,C	2
CBI LSL	P,b Rd	I/Oレジスタのビット解除(0)   論理的左ビット移動	$I/O(P,b) \leftarrow 0$ $Rd(n+1) \leftarrow Rd(n), Rd(0) \leftarrow 0$	I,T,H,S,V,N,Z,C I,T,H,S,V,N,Z,C	2
LSR	Rd	論理的右ビット移動	$Rd(n) \leftarrow Rd(n+1), Rd(7) \leftarrow 0$	I,T,H,S,V,N,Z,C	1
ROL	Rd	キャリーを含めた左回転	$Rd(0) \leftarrow C$ , $Rd(n+1) \leftarrow Rd(n)$ , $C \leftarrow Rd(7)$	I,T,H,S,V,N,Z,C	1
ROR	Rd	キャリーを含めた右回転	$Rd(7) \leftarrow C$ , $Rd(n) \leftarrow Rd(n+1)$ , $C \leftarrow Rd(0)$	I,T,H,S,V,N,Z,C	1
ASR	Rd	算術的右ビット移動	$Rd(n) \leftarrow Rd(n+1), n=0 \sim 6$	I,T,H,S,V,N,Z,C	1
SWAP	Rd	ニブル(4ビット)上位/下位交換	$Rd(7\sim4) \Leftrightarrow Rd(3\sim0)$	I,T,H,S,V,N,Z,C	1
BSET	S	ステータス レジスタのビット設定(1)	$SREG(s) \leftarrow 1$	1,T,H,\$,Y,N,Z,C	1
BCLR	S Dr. l-	ステータス レシブスタのビット解除(0)	$SREG(s) \leftarrow 0$	0,0,0,0,0,0,0	1
BST BLD	Rr,b Rd,b	汎用レジスタのビットを一時フラグへ移動   一時フラグを汎用レジスタのビットへ移動	$T \leftarrow Rr(b)$ $Rd(b) \leftarrow T$	I, <b>T</b> ,H,S,V,N,Z,C I,T,H,S,V,N,Z,C	1
SEC	ra,b	キャリー フラグを設定(1)	$C \leftarrow 1$	I,T,H,S,V,N,Z,C	1
CLC		キャリー フラグを解除(0)	$C \leftarrow 0$	I,T,H,S,V,N,Z, <b>0</b>	1
SEN		負フラグを設定(1)	N ← 1	I,T,H,S,V, <b>1</b> ,Z,C	1
CLN		負フラグを解除( <mark>0</mark> )	$N \leftarrow 0$	I,T,H,S,V,0,Z,C	1
SEZ		ゼロ フラグを設定(1)	Z ← 1	I,T,H,S,V,N,1,C	1
CLZ		ゼロフラグを解除(0)	$Z \leftarrow 0$	I,T,H,S,V,N,Ø,C	1
SEI CLI		全割り込み許可 全割り込み禁止	I ← 1 I ← 0	1,T,H,S,V,N,Z,C 0,T,H,S,V,N,Z,C	1
SES		全割り込み景正   符号フラグを設定(1)	$S \leftarrow 1$	I,T,H,\$,V,N,Z,C	1
CLS		符号フラグを解除(0)	S ← 0	I,T,H,0,V,N,Z,C	1
SEV		2の補数溢れフラグを設定(1)	V ← 1	I,T,H,S,¥,N,Z,C	1
CLV		2の補数溢れフラグを解除(0)	V ← 0	I,T,H,S, <b>0</b> ,N,Z,C	1
SET		一時フラグを設定(1)	T ← 1	I,T,H,S,V,N,Z,C	1
CLT		一時フラグを解除( <u>0</u> )	$T \leftarrow 0$	I,0,H,S,V,N,Z,C	1
SEH		ハーフキャリー フラケ を設定(1)   ハーフキャリー フラケ を解除(0)	H ← 1	I,T,H,S,V,N,Z,C	1
CLH			<u>  H ← 0                                 </u>	I,T, <b>0</b> ,S,V,N,Z,C	1
NOP		■ MICO	17 hh. Hi 17	I,T,H,S,V,N,Z,C	1
SLEEP		休止形態開始	休止形態参照	I,T,H,S,V,N,Z,C	1
WDR		ウォッチト゛ック゛ タイマ リセット	ウォッチドッグタイマ参照	I,T,H,S,V,N,Z,C	1
BREAK		一時停止	内蔵デバッグ機能専用	I,T,H,S,V,N,Z,C	N/A

## 注文情報

速度(MHz)	電源電圧	注文符号	外囲器	動作範囲	
		ATmega169V-8AI	64A		
8 ( <b>注3</b> )	1.8~5.5V	ATmega169V-8AU ( <b>注2</b> )	04A		
○ ( <u>/±</u> 3)	1.8~5.50	ATmega169V-8MI	C 4N 11	工業用 (-40℃~85℃)	
		ATmega169V-8MU ( <b>注2</b> )	64M1		
	2.7~5.5V	ATmega169-16AI	64A		
16 ( <b>注3</b> )		ATmega169-16AU ( <b>注2</b> )	04A		
10 (,±3)		ATmega169-16MI	64M1		
		ATmega169-16MU ( <b>注2</b> )	041/11		

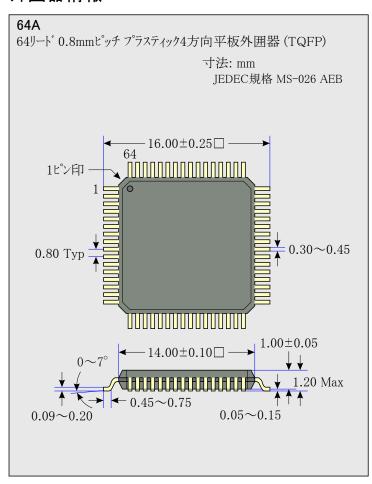
注: このデバイスはウェハー(チップ単体)形状でも供給できます。最低数量と詳細な注文情報については最寄のAtmel営業所へお問い合わせください。

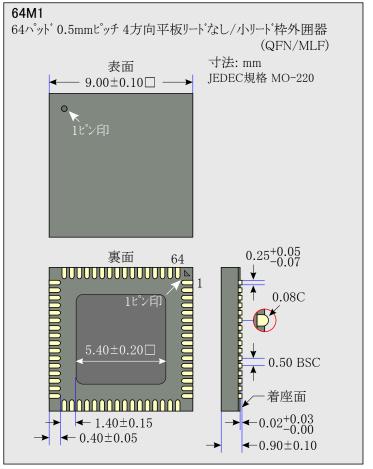
注2: 有害物質使用制限に関する欧州指令(RoHS指令)適合の鉛フリー製品。またハロゲン化合物フリーで完全に安全です。

注3: 速度と電源電圧の関係については図136.と図137.をご覧ください。

	外囲器形式
64A	64リート 1.0mm厚 プラスティック4方向平板外囲器 (TQFP)
64M1	64パッド9×9×1mm 0.5mmピッチ 4方向平板リードなし/小リード枠外囲器 (QFN/MLF)

## 外囲器情報









## データシート改訂履歴

この章内の参照頁番号はこの文書が参照されていることに注意してください。この章内の改訂番号は文書の改訂番号を参照してください。

## 2514A - 2002年8月から2514B - 2002年9月へ変更

1. フラッシュ メモリの書き換え可能回数(寿命)を1,000から10,000に変更

## 2514B - 2002年9月から2514C - 2002年11月へ変更

- 1. 231頁に「障害情報」を追加
- 2. 227頁の「注文情報」と「外囲器情報」に64ピンQFN/MLF外囲器についての情報を追加
- 3. 227頁「注文情報」の工業用注文番号削除(付随して温度範囲の変更)

## 2514C - 2002年11月から2514D - 2003年1月へ変更

- 1. 187頁の「JTAGインターフェース経由のプログラミング」にTCK周波数制限を追加
- 2. 192頁の「フラッシュ メモリ書き込み」と「EEPROM書き込み」に最初の手順としてチップ消去を追加
- 3. 39頁に「未接続ピン」項を追加
- 4. 24頁の「内蔵デバッグ機能」にOCDの禁止方法項目を追加
- 5. ADCとANA\_COMP割り込みアドレスが入れ替わっていたのを修正
- 6. 187頁「直列プログラミング特性」の図表を削除、196頁の「SPIタイミング特性」表を改善
- 7. 171頁「**^゚ージ書き込みの実行**」の未使用Zポインタ ビットに対する「無効」を「0書き込み」に変更
- 8. 149頁のLCDCRAレシ、スタ記述で、「LCDフレーム完了」を「LCDフレーム開始」に修正
- 9. USIコード例でOUTをSTS、INをLDSに変更と、f<sub>SCKmax</sub>を修正 USIのI/Oレジスタは拡張I/O空間で、そのためINとOUTは使えません。LDSとSTSは実行時に1周期多く必要とし、そのために f<sub>SCKmax</sub>は対応して変更されました。
- 10. TOSKONとTOSCK、G10とG20は境界走査に存在しないため、各々160頁の表103、161頁の図115、162頁の表105から削除
- **11**. デバイスの「特徴」一覧でMIPS/MHzを4から16に変更
- **12**. 5頁「**ピン概要**」の「AVCC」でポートAからポートFに修正
- **13**. 120頁の「ボーレート設定例」で230.4Mbpsを230.4kbpsに修正
- 14. 119頁の表78. 「UCPOLL'ット設定」でXCKの上昇端と下降端の位置を修正
- 15. 存在しない「多目的発振器」、「32kHz/JJスタル発振器」応用記述の参照を削除
- 16. 13頁の表1.で校正付きRC発振器周期数を8448から67584に修正
- 17. タイマ/カウンタ1で各種の微小修正
- 18. タイマ/カウンタ0と2にPWMの対称性についての情報を追加
- 19. DIDR0とDIDR1の内容を修正
- 20. 既に存在する2桁の前にCOM番号の付加により、LCDDR内の全ビットに固有名(例:SEG304)を作成
- 21. 98頁の「タイマ/カウンタ2 非同期動作」下の拡張スタンハイ動作をA/D変換雑音低減動作に変更
- 22. ポートBが他のポートよりも優れた駆動能力を持つことについての注意を追加表もその結果、194頁の「DC特性」のように修正
- **23**. 171頁の「**^゚ージー時緩衝部の設定 (^゚ージ設定)**」下にSPMぺージ設定中のEEPROM書き込みについての注意を追加
- 24. ADHSMを完全に削除
- **25**. 227頁の「**外囲器情報**」を更新

### 2514D - 2003年1月から2514E - 2003年2月へ変更

- 1. 1頁の「特徴」項でATmega169とATmega169Lに関する情報を更新
- **2.** 2頁の「ピン配置」、3頁の図2、5頁の「PG4~0 (ポートG)」、50頁の「ポートGの交換機能」、53頁の「I/Oポート用レジスタ」でPG5ピンの全引用を削除
- 3. 176頁の表118.「拡張ヒュース゛バイト一覧」を更新
- 4. 231頁の「**障害情報**」に重要なデータシートの変更を含む'ATmega169改訂C'を追加
- 5. 227頁の「注文情報」でATmega169Lの新速度範囲と新16MHz ATmega169を含む更新

#### 2514E - 2003年2月から2514F - 2003年4月へ変更

1. 文書全体でICPをICP1へ名称変更

- 2. 17頁の「**クリスタル用発振器**」の表で注:を削除
- 3. 18頁の「校正付き内蔵RC発振器」項に「XTAL1/XTAL2がタイマ用発振器ピンとして使える」を記述
- 4. 20頁の「切り替え時間」で前置分周器設定間の切り替え項を追加
- 5. 194頁の「**電気的特性**」項でDC特性とA/D特性を更新 25頁の**表16**.、27頁の**表19**.、197頁の**表134**.からTBDを削除
- 6. Write PINxレジスタに関して37頁の図23、40頁の図26、159頁の図110.を更新
- 7. ITAGに関して48頁の「ポートF**の交換機能**」を更新
- **8**. 122頁の「**多用途直列インターフェース (USI)**」でタイマ/カウンタ0溢れをタイマ/カウンタ0比較一致に置換、また126頁の「**開始条件検出器**」と 128頁の「**USI制御レジネタ**(USICR)」も更新
- 9. 132頁の「A/D変換器」で「特徴」を、138頁の表88.を更新
- 10. 170頁の図118.と176頁の表118.に注:を追加

## 2514F - 2003年4月から2514G - 2003年4月へ変更

- 1. 198頁の「代表特性」を更新
- 2. 227頁の「注文情報」で誤植を更新
- 3. 76頁の図46.、26頁の表18.、156頁の表100.を更新

## 2514G - 2003年4月から2514H - 2003年5月へ変更

1. 図146、図166、図193.で誤植を更新

#### 2514H - 2003年5月から2514I - 2003年9月へ変更

- 1. 「新製品情報」を削除
- 2. 3頁の図2.でAGNDを削除、16頁の図11.に「システム クロック前置分周器」を追加
- 3. 25頁の表16.、26頁の表17.、27頁の表19.、47頁の表41.を更新
- 4. 24頁の「内蔵デバック、機能」を「JTAGインターフェースと内蔵デバック、機能」へ変更と更新
- 5. 62頁の「タイマ/カウンタ0制御レシ、スタA(TCCR0A)」でCOM01,0をCOM0A1,0へ、94頁の「タイマ/カウンタ2制御レシ、スタA(TCCR2A)」でCOM21,0をCOM2A1,0へ更新
- 6. JTAGENに関する152頁の「検査入出力ポート(TAP)」を更新
- 7. 158頁のJTDビットに対する説明を更新
- 8. 176頁の表119.にJTAGENtュースに関する注意を追加
- 9. 194頁「電気的特性」の「絶対最大特性」と「DC特性」を更新
- 10. 231頁の「障害情報」更新、JTAG命令IDCODEに関する問題解決案を追加

#### 2514I - 2003年9月から2514J - 2003年12月へ変更

1. 18頁「校正付き内蔵RC発振器」を更新

#### 2514J - 2003年12月から2514K - 2004年4月へ変更

- 1. 8頁の「スタック ポインタ」で位置を\$60から\$100~変更
- 2. 26頁の表17.、29頁の表21.、174頁の表115.を更新
- 3. 18頁「校正付き内蔵RC発振器」を更新
- 4. 23頁に「**電力削減レジスタ**」を、例を201頁の「**周辺機能部供給電流**」に新規追加
- 5. 132頁の「A/D変換器」でポート記述の誤植を修正
- 6. 142頁からの「LCD制御器」旧版を削除、新版を追加
- 7. 194頁「電気的特性」を更新
- 8. 198頁からの「代表特性」を更新
- 9. 227頁の「**注文情報**」を更新 ATmega169LをATmega169とATmega169Vによって置換

## 2514K - 2004年4月から2514L - 2005年3月へ変更

- 1. MLF外囲器を"4方向平板リートなし/小リート、枠外囲器(QFN/MLF)"に代替変更
- 2. 25頁の表16.、81頁の表56.と表57.、150頁の表98.、151頁の表99.と表100.、186頁の表130.を更新
- 3. 34頁に「ピン変化割り込みタイミング」を追加
- **4**. 109頁の「USART初期化」でCコートが例を更新





- 5. 23頁の「**電力削減レジスタ**(PRR)」と151頁の「**LCD濃淡制御レジスタ**(LCDCCR)」に注を追加
- **6**. フラッシュ ヘーシ 構成表とEEPROMへーン 構成表を177頁の「**ヘージ 容量**」へ移動
- 7. 185頁の「**直列プログラミング手順**」を更新
- 8. 198頁の「代表特性」を更新
- 9. 202頁の「**電力削減レジスタの使用**」を「**周辺機能部供給電流**」に改名
- 10. 221頁の「レジスタ要約」を更新
- 11. 227頁の「注文情報」を更新
- 12. 132頁の図83.,137頁の図91.,181頁の図123.を更新

## 2514L - 2005年3月から2514M - 2005年5月へ変更

1. このデータシートはATmega169/Vの改訂A~Eを網羅します。改訂F以降はwww.atmel.comで得られる"doc2597.pdf"データシートで網羅されます。

## 2514M - 2005年5月から2514N - 2006年3月へ変更

- 1. 「新規設計で推奨されません」を追加
- 2. 改訂A~EとF以降に関する通告文を削除
- 3. 26頁の表17.を更新

## 2514N - 2006年3月から2514O - 2006年3月へ変更

- 1. 汎用I/Oピン数を53から54に更新(<mark>訳注</mark>:この更新が矛盾するため本書では無視)
- 2. 100頁の「**直列周辺インターフェース** (SPI)」を更新

## 25140 - 2006年3月から2514P - 2006年7月へ変更

- 1. 77頁の「**高速PWM動作**」を更新
- 2. 122頁の「**多用途直列インターフェース (USI)**」内で特徴を更新
- 3. 126頁に「**クロック速度の考察**」を追加
- 4. 130頁で「ビット6 ACBG: 基準電圧選択」を更新
- 5. 62頁の表49.,表51.,81頁の表56.,表57.,82頁の表58.,94頁の表61.,表63.を更新
- 6. 134頁の「**前置分周と変換タイミンク**゙」を更新
- 7. 142頁の「LCD制御器」内で特徴を更新
- 8. 231頁の「障害情報」を更新

## 障害情報

この章の改訂番号はATmega169/Vデバイスの改訂版を参照してください。

・直列プログラミング動作移行時の不具合	В
・仕様よりも低精度なA/D変換計測	В
・USART送信禁止移行時の動作	В
・外部クロック設定の不具合	В
・LCD濃淡電圧が不正確	В
・内蔵RC発振器が4MHzで動作	В
・LCD消費電流	С
・いくつかのデータの組み合わせがLCDのセグメントを薄暗くし得る	С
・LCD濃淡制御	С
・JTAGENプログラム(0)時、パワーダウン動作での高消費電流	С
・IDCODEがTDI入力からのデータを遮蔽	D
・LCD表示器内の直列高抵抗がセグメントを薄暗くし得る	D
・非同期タイマでタイマレジスタ書き込み時に割り込みが失われる可能性	B,C,D,E

#### 1. 直列プログラミング移行時の不具合(B)

直列プログラミング動作移行時、第2小小が直列プログラミング手順で示されるように返答されません。

#### 対策/対処

デバイスがプログラミング動作かを保証するには第3バイトの返答を検査してください。(これは改訂Cで修正されます。)

#### 2. 仕様よりも低精度なA/D変換計測(B)

A/D変換は意図したように動作していません。結果として正の変位(オフセット)があります。

#### 対策/対処

これは改訂Cで修正されます。

#### 3. USART送信禁止移行時の動作(B)

送信中にTXENへ0を書くと、送信が急に止められます。TXENに0が書かれると、USARTを停止する前に送信が完了することをデータシートの記述は示します。

#### 対策/対処

TXENに0を書く前に送信が完了することを保証します。(これは改訂Cで修正されます。)

#### 4. 外部クロック設定の不具合(B)

外部発振器(クロック信号)はデータシートに記載された設定で動作しません。

#### 対策/対処

他のクロック元を使ってください。(これは改訂Cで修正されます。)

## 代替処置

XTAL1にプルダウンを追加することが発振器を起動させます。

### 5. LCD濃淡電圧が不正確(B)

1.8~3.1V間のLCD濃淡電圧は正しくありません。VCCが1.8~3.1V間の時にLCD濃淡電圧は約0.5V低下します。この間の消費電流は予想される値よりも高くなります。

#### 対策/対処

濃淡は正しくありませんが表示は見えるので、濃淡制御レシ、スタを使うことで部分的に補償できます。(これは改訂Cで修正されます。)

#### 6. 内蔵RC発振器が4MHzで動作(B)

内蔵RC発振器は示された8MHzの代わりに4MHzで動作します。従って、全てのフラッシュメモリやEEPROMの書き込み時間は示されるよりも長く、2倍です。これにはチップ消去、バイト書き込み、ページ書き込み、ヒューズ書き込み、施錠ビット書き込み、CPUからのEEPROM書き込み、フラッシュメモリの自己書き込みも含みます。

この訳は改訂B試供品がCKDIV8ヒューズ非プログラム(1)で出荷されるためです。

#### 対策/対処

8MHz動作が必要とされる場合、外部クロックを適用します。(これは改訂Cで修正されます。)





#### 7. LCD消費電流(C)

VCCがVLCD-0.2VからVLCD+0.4Vの範囲内の間でLCD消費電流は予期したよりも3倍まで高くなります。これはLCD動作でのパックーセーブ動作だけの問題で、LCD電流は他の全ての動作種別での全体的な消費電力と比べて無視できるほどです。

#### 対策/対処

対策はありません。

#### 8. いくつかのデータの組み合わせがLCDのセグメントを薄暗くし得る(C)

データの或る組み合わせが表示されると、共通電極に接続した全セグメントが薄暗く(低い濃淡)なるかもしれません。

#### 対策/対処

既定波形 : 何れかの未使用セグメント ピンがあれば、それらの1つへの1000pFとそのセグメントへの常時'0'書き込みがこの問題

を無くします。

低電力波形 : 各共通(コモン)ピンに1000pFコンデンサを付加してください。

#### 9. LCD濃淡制御(C)

LCDクロックに同期(チップ)クロックを使い、チップ クロックが125kHzまたはより速い場合、濃淡制御は正しく動きません。

#### 対策/対処

低チップ クロック周波数(32kHz)を使うか、またはLCDCAPt ンに外部電圧を印加してください。

## 10. JTAGENプログラム(0)時、パワーダウン動作での高消費電流(C)

JTAGENがプログラム(0)されると、TDO(PF6)の入力緩衝部が常に許可され、プルアップが常に禁止されます。これは浮き状態のままにし得ます。

#### 対策/対処

PF6に外部プルアップを付加してください。

最終製品出荷前にJTAGENtュースを非プログラム(1)にしてください。

## 11. IDCODEがTDI入力からのデータを遮蔽(D)

IDCODEは正しく動作しません。DR更新の間中、後続するデバイスへのデータは全て1に置換されます。

#### 問題の修正と対策

- ATmega169が走査チェーン内で唯一のデバイスなら、この問題は見られません。
- ATmega169のデバイスIDレジスタとおそらくは走査チェーンの後続するデバイスからのデータ内容を読むために、IDCODE命令を実行するか、またはTAP制御器の検査回路リセット状態へ移行することのどちらかにより、ATmega169のデバイスIDレジスタを選んでください。 境界走査チェーンの先行デバイスのデバイスIDレジスタを読む間中、ATmega169へBYPASS命令を実行してください。
- 境界走査チェーン内の全デバイスのデバイスIDが同時に捕獲されなければならない場合、ATmega169はチェーンの先頭デバイスでなければなりません。

#### 12. LCD表示器内の直列高抵抗がセグメントを薄暗くし得る(D)

硝子部側に直列高抵抗( $>20k\Omega$ )がある形式のいくつかの表示器はLCDのセグメントが薄暗くなり得ます。

## 対策/対処

各共通(コモン)ピンとGND間に1000pF(470~1500pF)コンデンサを付加してください。

## 13. 非同期タイマ/カウンタでタイマ/カウンタ レジスタ書き込み時に割り込みが失われる可能性 (B,C,D,E)

非同期タイマ/カウンタ2のクロックに同期化されるタイマ/カウンタ2のレシ、スタの1つが溢れ割り込みが起こる前の周期で書かれると、その割り込みは消失するかもしれません。

#### 対策/対処

タイマ/カウンタ2制御レジブスタ(TCCR2)またはタイマ/カウンタ2比較レジブスタ(OCR2)を書く前に、タイマ/カウンタ2が値\$FFでないのを常に調べてください。

非同期動作でタイマ/カウンタ2のレシブスタの何れかへ書くための安全な時間は、比較レシブスタが\$FFでない比較一致割り込み処理ルーチン内、または比較レシブスタが\$FFなら、割り込みの開始から最低1非同期クロック周期後だけです。

# 目次

特徴 ・・・・・・・	可效命中
<b>ピン配置 ・・・・・・・・・</b> 2	比較出力部5
お断り ・・・・・・ 2	<b>比較一致出力部 ······</b> 5
概要 •••••• 3	動作種別 ・・・・・・・・・・ 5
	<b>タイマ/カウンタのタイミング・・・・・・・・・・</b> 6
構成図 ・・・・・・・・・・・・・ 3	8ビット タイマ/カウンタ0用レジスタ ・・・・・・・・・・・ 6:
ピン説明 ・・・・・・・・・・・・・・・・ 4	タイマ/カウンタ0とタイマ/カウンタ1の前置分周器・・・ 65
コート 例について・・・・・・・・・・・・・・・・・ 5	
AVR CPU <b>17</b> · · · · · · · 6	<b>16ビット タイマ/カウンタ1 ・・・・・・・・・・・</b> 67
<b>序説 ・・・・・・・・・・・・・・・・・・・・・・</b> 6	<b>概要 ・・・・・・・・・・・・・・・・・・・・</b> 6'
	<b>16ビット レジスタのアクセス ・・・・・・・・・・</b> 69
構造概要 ・・・・・・・・・・・・・・・・・・・・・・・・ 6	<b>タイマ/カウンタのクロック ・・・・・・・・・</b> 7
ALU (Arithmetic Logic Unit) •••• 6	計数器部 ••••••
ステータス レジスタ ・・・・・・・・・・・・・・・・・・・ 7	<b>捕獲入力部 ·······</b> 7
<b>汎用レジスタファイル・・・・・・・・・・・・・・</b> 8	比較出力部 ・・・・・・・・・・・・・・・・・ 7:
スタック ホ°インタ ・・・・・・・・・・・・・・・ 8	比較一致出力部 •••••• 7
<b>命令実行タイミング・・・・・・・・・・・・・・・・・</b> 9	動作種別 ・・・・・・・・・・・・ 7
リセットと割り込みの扱い ・・・・・・・・・・・ 9	
AVR ATmega169のメモリ・・・・・・・・11	タイマ/カウンタのタイミング・・・・・・・・・・・8
	16ビット タイマ/カウンタ1用レジスタ ・・・・・・・・8
実装書き換え可能なプログラム用フラッシュメモリ・・・11	8ビット タイマ/カウンタ2 (PWM,非同期動作) ・・・・ 86
データ用SRAMメモリ・・・・・・・・・・ 11	概要 ••••••80
<b>データ用EEPROMメモリ ・・・・・・・</b> 12	<b>タイマ/カウンタのクロック ・・・・・・・・・・・・・・</b> 8'
I/Oメモリ (レジスタ) ・・・・・・・・15	計数器部8
システム クロックとクロック選択 ・・・・・・・・・ 16	比較出力部8
<u>クロック系統とその配給・・・・・・・・・・・1</u> 6	比較一致出力部 ・・・・・・・・・・・・・・・・・・・・・ 8
クロック元 ・・・・・・・・・・・・・・・・・・・・・・・・17	
既定のクロック元 ・・・・・・ 17	動作種別
クリスタル用発振器・・・・・・・・・・・17	タイマ/カウンタのタイミング・・・・・・・・・・・・・・・・・・・・・・・・9:
低周波数列ス外用発振器・・・・・・・・・・18	8ビットタイマ/カウンタ2用レジスタ・・・・・・・・・・・9
	<b>タイマ/カウンタ2の非同期動作 ・・・・・・・・</b> 9
校正付き内蔵RC発振器・・・・・・・・・・・18	タイマ/カウンタ2の前置分周器 ・・・・・・・・・9
<b>外部クロック信号・・・・・・</b> 19	SPI (直列周辺インターフェース)・・・・・・・ 100
システム クロック出力緩衝部 ・・・・・・・19	SSピンの機能 · · · · · · · · 10:
<b>タイマ/カウンタ用発振器 ・・・・・・・・・</b> 19	データ転送形式・・・・・・・・・・・・・・・・・10.
システム ウロック前置分周器 ・・・・・・・・20	USART 109
電力管理と休止形態 ・・・・・・・・ 21	
アイドル動作 ・・・・・・・・・・・・・・・・・・・・・・・・21	概要 10
A/D変換雑音低減動作 ························· 21	<b>クロック生成 ・・・・・・・・・・・</b> 10′
パワーダウン動作 ・・・・・・・・・・・・・・・・・ 22	フレーム形式 ・・・・・・・・・・・・108
ハ , , , , , , , , , , , , , , , , , , ,	USARTの初期化 ・・・・・・ 109
	<b>USARTのデータ送信 ・・・・・・・・・・・・・・・・</b> 110
スタンバイ動作・・・・・・・・・・・・・・・・・・22	USARTのデータ受信 ・・・・・・・・・・・・・ 11
電力削減(電力削減レジスタ)・・・・・・・・・・23	<b>非同期受信 ••••••</b> 114
消費電力の最小化 · · · · · · · 24	複数プロセッサ通信動作・・・・・・・・・・・110
<u>システム制御とリセット・・・・・・・・・・・・・・・・・25</u>	USART用レジスタ 11'
内部基準電圧 ・・・・・・・・・・ 27	
<b>ウォッチドッグ タイマ ・・・・・・・・・・</b> 28	ポーレート設定例・・・・・・・・・・・・・・・・・・・・・・・・ 12・
ウォッチトック、タイマ設定変更の時間制限手順・・・・ 29	多用途直列インターフェース (USI) ・・・・・・・ 122
_	概要 ・・・・・・・・・・・・ 12/
割り込み 30	<b>機能説明 ・・・・・・・・・・・・</b> 12:
ATmega169の割り込みベクタ ・・・・・・ 30	<b>USIでの代替使用 ・・・・・・・・・・・・・</b> 120
<u>外部割り込み ・・・・・・・・・・・・・ 34</u>	USI用レジスタ ······ 12'
ピン変化割り込みタイミング・・・・・・ 34	アナログ比較器・・・・・・・・・・・ 130
入出力ポート · · · · · · · · · · · · · · · 37	
序説	7 ナログ 比較器入力選択 ・・・・・・・・ 13
	A/D変換器 ······ 132
標準デジタル入出力としてのポート ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	<b>特徴 ······</b> 133
交換ポート機能 ・・・・・・・・・・・・・・・・ 40	<b>操作 ······</b> 133
I/Oポート用レジスタ ・・・・・・・ 51	変換の開始 ・・・・・・・・・・・ 13:
8ビット タイマ/カウンタO (PWM) ・・・・・・・ 54	前置分周と変換タイミング・・・・・・・・・・・13-
概要 ・・・・・・・・・ 54	チャネル変更と基準電圧選択 · · · · · · 13
タイマ/カウンタのクロック ・・・・・・・・・ 55	雑音低減機能 ・・・・・・・・・・ 13
/ 1 · / · · / · · / · · / · / · / · / ·	ΨΕ ΕΙ ΙΕΝΙΙΑ/ΙΝΧ ΠΕ 13





	<b>A/D変換の結果・・・・・・・・</b> 138
	<b>A/D変換用レジスタ ・・・・・・・・・</b> 139
LCD#	<b>刊御器 ·····</b> 142
	A/D変換の結果 ・・・・・・・・・・ 138 A/D変換用レジスタ ・・・・・・・・・ 139 制御器 ・・・・・・・・・・・ 142 特徴 ・・・・・・・・・・・・・・・ 142
	概要
	動作種別
	LCD使用法 ······· 146
	LCD制御器用レジスタ・・・・・ 149
ITAC	
JIAG	
	特徴 · · · · · · · · · · · · · · · · · · ·
	102
	検査入出力ポート (TAP:Test Access Port) ・・ 153
	TAP制御器 ····································
	境界走査チェーン(Boundary-Scan Chain) 154
	の使用 154
	内蔵デバッグ機能の使用 ・・・・・・・・・ 154
	内蔵デバッグ特殊JTAG命令 ・・・・・・・・ 154
	内蔵デバッグ関連エ/Oメモリ内のレジスタ・・・・・・ 154
	JTAGプログラミング能力の使用 ・・・・・・・ 155
	参考文献
IEEE1	<b>149</b> .1(JTAG)境界走査 · · · · · · · · 156
	特徴 ······ 156 概要 ····· 156
	概要 ・・・・・・・・・ 156
	<del>データレジスタ ・・・・・・・・・・156</del>
	境界走査(Boundary-Scan)用JTAG命令 · · · · 157
	境界走査関連I/Oメモリ内のレジスタ ・・・・・ 158
	境界走査チェーン(Boundary-Scan Chain) ・・・・ 159
	ATmega169の境界走査順 ・・・・・・・・ 164
	境界走査記述言語(BSDL)ファイル ・・・・・・ 165
ブート	コーダ支援 - RWW 自己プログラミング・・・ 166
	特徴 ····································
	フラッシュ メモリの応用領域とブート ローダ領域 ・・・ 166
	書き中に読みが可能な領域と不能な領域・・166
	ブートローダ・施錠ビット・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・
	ブートローダ プログラムへの移行・・・・・・・・・168
	自己プログラミングでのフラッシュ アトレス指定・・・・・ 170
	フラッシュ メモリの自己プログラミング・・・・・・・・・・・170
J = 11 =	
メモリノ	『ロクプラミンク』・・・・・・・・・・・・・・ 175
	プログラム メモリとデータ メモリ用施錠ビット・・・・・・ 175
	tュース・ヒット・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・
	識票バイト ・・・・・・・・・・・・ 177
	校正バイト ・・・・・・・・・・・ 177
	ページ 容量・・・・・・・・・・・・・177
	业列ノログラミング ・・・・・・・・・・・・・ 1//
	並列プログラミング手順 ・・・・・・・・・・・ 178
	直列プログラミング・・・・・・・・・・185
	直列プログラミング手順・・・・・・・・・・・・185
	JTAGインターフェース経由プログラミング・・・・・・・187
電気	<mark>ク特性・・・・・・・・・・・・・・・・・・</mark> 194 絶対最大定格 ・・・・・・・・・・・・ 194
	<b>絶対最大定格 ······</b> 194
	DC特性 · · · · · · · · · · · · · · · · · · ·
	<b>外部クロック特性 ・・・・・・・・・</b> 195
	最高速と動作電圧・・・・・・・ 195
	SPI タイミング 特性 · · · · · · · 196
	A/D変換器特性 (暫定) · · · · · · 197
	LCD制御器特性 (暫定) · · · · · · · 197
代表特	<b>寺性 ·····</b> 198
	活動動作消費電流 •••••• 198
	ア仆 ル動作消費電流・・・・・・・・・・・200

周辺機能部供給電流 ・・・・・・・・・ 202
<b>パワーダウン動作消費電流 ・・・・・・・・・</b> 202
<b>パワーセーブ動作消費電流 ・・・・・・・・・・・・</b> 203
スタンバイ動作消費電流 ・・・・・・・ 203
<b>ピン プルアップ・・・・・・・・・・・・・・・</b> 206
t <sup>2</sup> ン駆動能力 ······ 208
<b>ピン 閾値とヒステリシス・・・・・・・・・・・・</b> 212
低電圧検出器(BOD)閾値と214
プテログ 比較 畚 変位(オノセット)
内部発振器周波数 ・・・・・・・・・ 216
周辺機能部消費電流 ・・・・・・・・・ 217
リセット消費電流とリセット パルス幅 ・・・・・・・ 220
レシ <sup>、</sup> スタ要約・・・・・・・221
<mark>命令要約 ••••••</mark> 225
· · · · · · · · · · · · · · · · · · ·
デ <b>ータシート改訂履歴 ・・・・・・・・・・・・・・</b> 228
<b>障害情報 ••••••</b> 231



#### 本社

## Atmel Corporation

2325 Orchard Parkway San Jose, CA 95131, USA TEL 1(408) 441-0311 FAX 1(408) 487-2600

## 国外営業拠点

#### Atmel Asia

Unit 1-5 & 16, 19/F BEA Tower, Millennium City 5 418 Kwun Tong Road Kwun Tong, Kowloon Hong Kong TEL (852) 2245-6100 FAX (852) 2722-1369

#### Atmel Europe

Le Krebs
8, Rue Jean-Pierre Timbaud
BP 309
78054 Saint-Quentin-en-Yvelines
Cedex
France
TEL (33) 1-30-60-70-00
FAX (33) 1-30-60-71-11

#### Atmel Japan

104-0033 東京都中央区 新川1-24-8 東熱新川ビル 9F アトメル ジャパン株式会社 TEL (81) 03-3523-3551 FAX (81) 03-3523-7581

#### 製造拠点

#### Memory

2325 Orchard Parkway San Jose, CA 95131, USA TEL 1(408) 441-0311 FAX 1(408) 436-4314

#### Microcontrollers

La Chantrerie

2325 Orchard Parkway San Jose, CA 95131, USA TEL 1(408) 441-0311 FAX 1(408) 436-4314

BP 70602 44306 Nantes Cedex 3 France TEL (33) 2-40-18-18-18 FAX (33) 2-40-18-19-60

#### ASIC/ASSP/Smart Cards

Zone Industrielle 13106 Rousset Cedex France TEL (33) 4-42-53-60-00 FAX (33) 4-42-53-60-01

FAX (44) 1355-242-743

1150 East Cheyenne Mtn. Blvd. Colorado Springs, CO 80906, USA TEL 1(719) 576–3300 FAX 1(719) 540–1759

Scottish Enterprise Technology Park Maxwell Building East Kilbride G75 0QR Scotland TEL (44) 1355–803–000

#### RF/Automotive

Theresienstrasse 2 Postfach 3535 74025 Heilbronn Germany TEL (49) 71-31-67-0 FAX (49) 71-31-67-2340

1150 East Cheyenne Mtn. Blvd. Colorado Springs, CO 80906, USA TEL 1(719) 576-3300 FAX 1(719) 540-1759

#### **Biometrics**

BP 123 38521 Saint-Egreve Cedex France TEL (33) 4-76-58-47-50 FAX (33) 4-76-58-47-60

Avenue de Rochepleine

#### 文献請求

www.atmel.com/literature

お断り:本資料内の情報はAtmel製品と関連して提供されています。本資料またはAtmel製品の販売と関連して承諾される何れの知的所有権も禁反言あるいはその逆によって明示的または暗示的に承諾されるものではありません。Atmelのウェブサイトに位置する販売の条件とAtmelの定義での詳しい説明を除いて、商品性、特定目的に関する適合性、または適法性の暗黙保証に制限せず、Atmelはそれらを含むその製品に関連する暗示的、明示的または法令による如何なる保証も否認し、何ら責任がないと認識します。たとえAtmelがそのような損害賠償の可能性を進言されたとしても、本資料を使用できない、または使用以外で発生する(情報の損失、事業中断、または利益の損失に関する制限なしの損害賠償を含み)直接、間接、必然、偶然、特別、または付随して起こる如何なる損害賠償に対しても決してAtmelに責任がないでしょう。Atmelは本資料の内容の正確さまたは完全性に関して断言または保証を行わず、予告なしでいつでも製品内容と仕様の変更を行う権利を保留します。Atmelはここに含まれた情報を更新することに対してどんな公約も行いません。特に別の方法で提供されなければ、Atmel製品は車載応用に対して適当ではなく、使用されるべきではありません。Atmel製品は延命または生命維持を意図した応用での部品としての使用に対して意図、認定、または保証されません。

© Atmel Corporation 2006. 不許複製 Atmel®、ロゴとそれらの組み合わせ、AVR®とその他はAtmel Corporationの登録商標または商標またはその付属物です。他の用語と製品名は一般的に他の商標です。

#### © HERO 2022.

本データシートはAtmelのATmega169英語版データシート(改訂2514P-07/06)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訳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。

汎用入出力ポートの出力データレジスタとピン入力は、対応関係からの理解の容易さから出力レジスタと入力レジスタで統一表現されています。一部の用語がより適切と思われる名称に変更されています。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。