

タイマ/カウンタA型(TCA)での開始に際して

序説

著者: Catalin Visan, Microchip Technology Inc.

AVR®マイクロ コントローラは信号測定から事象同期と波形生成まで広い範囲の応用を網羅するように設計された強力な計時器が装備されています。

タイマ/カウンタA型(TCA:Timer/Counter type A)はtinyAVR® 0系、tinyAVR® 1系、megaAVR® 0系、AVR® DAデバイスに存在する16ビット計時器です。TCAの背後にある主な見解は複雑で難しい活動だけでなく、簡単な計時器の基本的な機能も実行するために非常に柔軟な計時器が必要とされることです。柔軟性は16ビット計時器を完全に2つの独立した8ビット計時器に分けることの可能性や組み込み波形生成動作のような多数の提供される機能によってもたらされます。別の重要な特性は計時器が走行している間にデューティサイクルが変更された時のPWM信号の予期せぬ動きのように、計時器使用時の一般的な問題を克服するためにTCAが考案されました。TCAは生成した波形を全ての単一状況で予測可能にする、各種レジスタの更新を同期化する2重緩衝されたレジスタを持ちます。

この技術概説の目的はTCAの動作形態のいくつかを読者に馴れさせて、この計時器の特殊性を強調し、初期化コードの断片を提供することです。機能のより深い理解についてはデータシートを調べてください。資料の構成は3つの特定使用事例を網羅します。

- **周期的割り込み動作の使用:**

250ms毎に割り込みを起動するように計時器を初期化し、割り込み処理ルーチンで例のGPIOを交互切り替えます。

- **2傾斜PWM信号の生成:**

GPIOピンで1kHzの周波数と50%のデューティサイクルで2傾斜16ビットPWM信号を生成するように計時器を初期化します。

- **分割動作で2つのPWM信号の生成:**

独立したデューティサイクルと周波数で2つのGPIOピンに於いて2つの1傾斜8ビットPWM信号を生成するように計時器を分割動作で初期化します。

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



GitHubでATmega4809コード例を見てください。

貯蔵庫を閲覧するにはクリックしてください。



GitHubでAVR128DA48コード例を見てください。

貯蔵庫を閲覧するにはクリックしてください。

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

目次

序説	1
1. 関連デバイス	3
2. 概要	4
3. 周期的割り込み動作の使用	4
4. 2傾斜PWM信号の生成	7
5. 分割動作で2つのPWM信号の生成	9
6. 参考文献	11
7. 改訂履歴	11
8. 追補	12
Microchipウェブ サイト	15
製品変更通知サービス	15
お客様支援	15
Microchipデバイス コード保護機能	15
法的通知	15
商標	16
品質管理システム	16
世界的な販売とサービス	17

1. 関連デバイス

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

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

図1-1. tinyAVR[®] 0システム概要

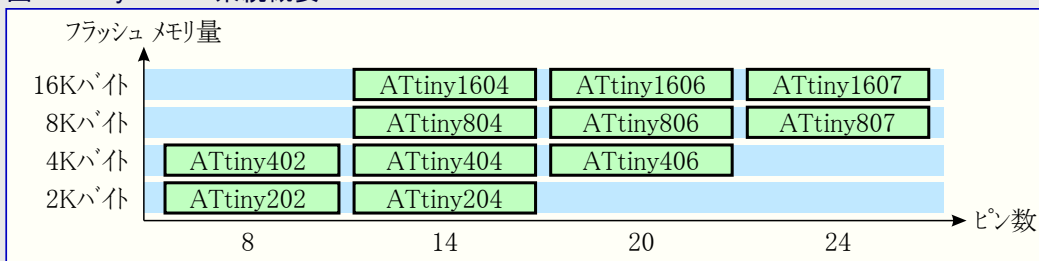


図1-2. tinyAVR[®] 1システム概要

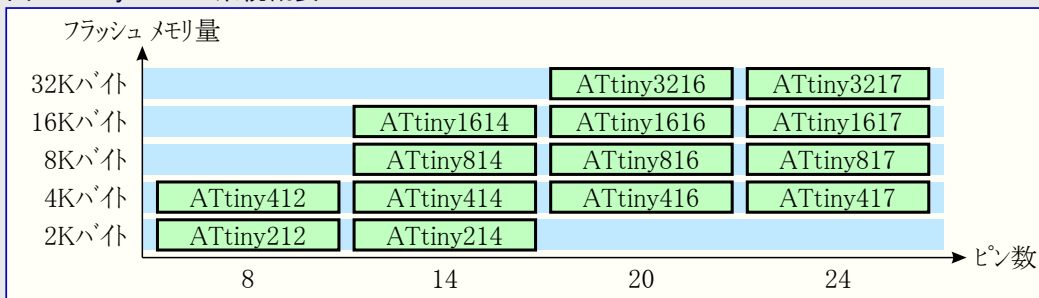


図1-3. megaAVR[®] 0システム概要

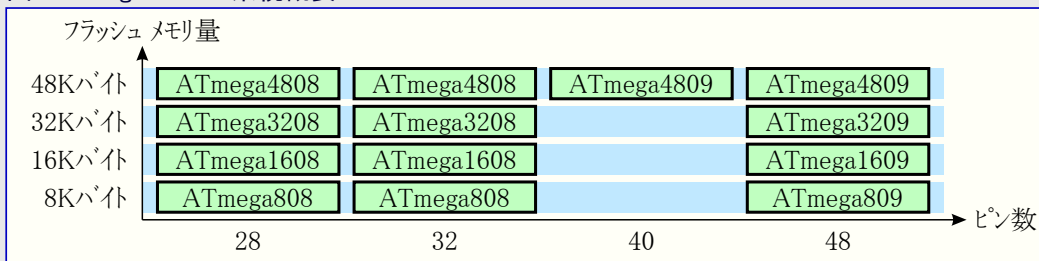
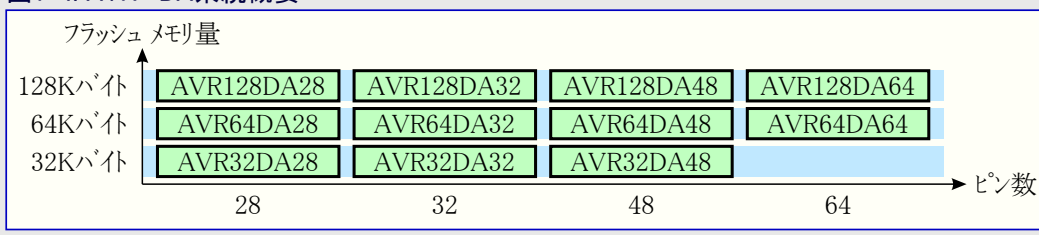


図1-4. AVR[®] DAシステム概要

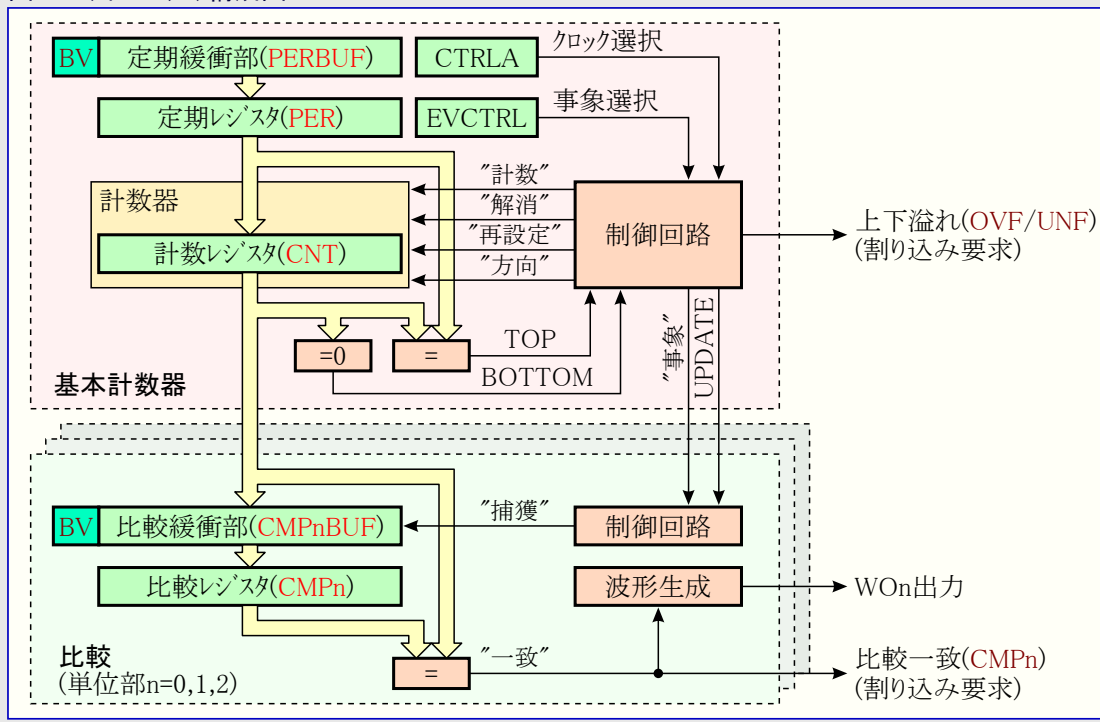


2. 概要

柔軟な16ビット タイマ/カウンタA型(TCA)は正確なプログラム実行タイミング、周波数と波形の生成、指令実行を提供します。

TCA実体は基本計数器と3つの比較チャネルから成ります。使用者はクロック刻時(計時器)または各種事象(計数器)に基づいて上向きまたは下向きに計数するように基本計数器を設定することができます。事象システムは方向制御や同期動作に使うこともできます。周期は特定レジスタだけでなく、各種波形を生成したり、事象を起動するのに使うことができる比較閾値から調節することができます。クロック元を分周するのに前置分周器を使うことができるとTCAがアイドル休止動作で動くことができることも言及する価値があります。

図2-1. タイマ/カウンタ構成図



計数器値は継続的に0と定期値と比較されます。条件の1つが合えば、制御論理回路部は構成設定された動作形態に従って働き、計数器を更新したり、割り込みを生成したりします。計数器は比較レジスタとも比較されます。割り込みを生成すると、波形生成動作が選ばれている場合に波形周期やパルス幅を設定するのにこれらの比較を使ってください。計数器、定期、比較のレジスタとそれらの全ての緩衝部は16ビット幅です。緩衝部は計数レジスタが更新される時にだけ各々のレジスタが更新されるのを保証する仕組みの一部です。各緩衝部は各々のレジスタが更新される必要があるかを判断するために論理回路部によって使われる緩衝部有効(BV)ビットを持ちます。

TCAは事象システムを使うように構成設定することができ、事象信号の上昇端や下降端を計数するのに利用したり、クロック刻時計数を許可するのにそれを使うことができます。また、事象信号の極性は、上向き計数に対するLow信号と下向き計数に対するHigh信号で方向を制御するのに使うことができます。更に、TCAは事象チャネル出力で1周期単発信号を生成することができます。事象チャネルで1周期単発信号を生成するための起動は計数器の溢れ、比較チャネル一致、または分割動作での下溢れにすることができます。

3. 周期的割り込み動作の使用

この計数器の基本的な使用事例はそれが更新される時毎に割り込みを起動するようにそれを設定することです。この動作形態は一まとまりのコードが数ms毎に繰り返して実行されなければならない場合に有用です。使用者は割り込みを許可し、適切なコードを含む割り込み処理ルーチン(ISR:Interrupt Service Routine)を設定しなければなりません。初期化とISRを含む基本例が以下で提供されます。プログラムはTCAの周期的割り込みを使って250ms毎にピンを交互切り替えます。ピンは以下で記述されるように計数器の初期化前に方向レジスタの対応するビットを設定(1)することによって出力として構成設定されなければなりません。この場合はポートAの0番(PA0)ピンが選ばれました。

1. 割り込み制御レジスタの対応するビットの設定(1)がTCAの溢れ割り込みを許可します。

```
TCA0_SINGLE.INTCTRL = TCA_SINGLE_OVF_bm;
```

図3-1. 割り込み制御(INTCTRL)レジスタ

ビット	7	6	5	4	3	2	1	0
		CMP2	CMP1	CMP0				OVF
アクセス	R	R/W	R/W	R/W	R	R	R	R/W
リセット値	0	0	0	0	0	0	0	0

2. この動作では波形が生成されてはならず、故にCTRLBレジスタの波形生成ビット領域はそれによって構成設定されなければなりません。

```
TCA0_SINGLE_CTRLB = TCA_SINGLE_WGMODE_NORMAL_gc;
```

図3-2. CTRLBレジスタ

ビット	7	6	5	4	3	2	1	0
		CMP2EN	CMP1EN	CMP0EN	ALUPD	WGMODE2~0		
アクセス	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
リセット値	0	0	0	0	0	0	0	0

● ビット2~0 – WGMODE2~0 : 波形生成動作 (Waveform Generation Mode)

これらのビットは波形生成動作を選び、計数器の計数進行、TOP値、UPDATE条件、割り込み条件、生成される波形の形式を制御します。

標準形態の動作では波形生成が全く実行されません。他の全ての動作形態に対して、対応する比較n許可(CMPnEN)ビットがこれを許可するように設定されていれば、波形生成部からの結果がポートピンに直結されるだけです。ポートピンの方向は出力として設定されなければなりません。

値	000	001	010	011	100	101	110	111
名称	NORMAL	FRQ	-	SINGLESLOPE	-	DSTOP	DSBOTH	DSBOTTOM
説明	動作	標準		1傾斜PWM		2傾斜PWM	2傾斜PWM	2傾斜PWM
	TOP	PER	(予約)	PER	(予約)	PER	PER	PER
	更新	TOP		BOTTOM		BOTTOM	BOTTOM	BOTTOM
	OVF	TOP	TOP	BOTTOM		TOP	TOPとBOTTOM	BOTTOM

3. 計数器が事象ではなくクロック刻時を計数するであろうため、EVCTRLレジスタのCNTEIビットは'0'に設定されなければなりません。これがCNTEIビットの既定値であることは言及する価値があります。

```
TCA0_SINGLE_EVCTRL &= ~(TCA_SINGLE_CNTEI_bm);
```

図3-3. EVCTRLレジスタ

ビット	7	6	5	4	3	2	1	0
						EVACT1,0		CNTEI
アクセス	R	R	R	R	R	R/W	R/W	R/W
リセット値	0	0	0	0	0	0	0	0

4. 定期(PER)レジスタに書かれる値は計数器が開始する時点と割り込みが起動される時点間のクロック刻時数で、そして2つの連続する割り込み間のクロック刻時数も表します。これは以下の式から推定することができます。

注: 定期レジスタに書かれる値は計数が'0'から始まるため望む計数よりも1つ少なくなります。

$$\text{TCA割り込み時間(s)} = \frac{\text{TCA定期} + 1}{\text{TCAクロック(Hz)}}$$

ここでTCA実体のクロックはTCAクロック(Hz)=fCLK(Hz)/TCA前置分周器と周辺機能クロックのfCLK=主クロック/主クロック前置分周器によって定義されます。これらの式を組み合わせると、以下の結果が得られます。

$$\text{TCA割り込み時間(s)} = \frac{(\text{TCA定期} + 1) \times \text{TCA前置分周器}}{f_{\text{CLK}}(\text{Hz})}$$

注: 定期(PER)レジスタは16ビット幅です。従ってTCA前置分周なしで達成可能な最長割り込み周期は以下で一覧にされます。

$$\text{TCA割り込み時間(s)} = \frac{(\text{TCA定期} + 1) \times \text{TCA前置分周器}}{f_{\text{CLK}}(\text{Hz})} = \frac{(0xFFFF + 1) \times 1}{3333333(\text{Hz})} = 19.66 \times 10^{-3}(\text{s})$$

この例に対して目的値を考察すると、

$$\text{TCA定期} = \frac{\text{TCA割り込み時間(s)} \times f_{\text{CLK}}(\text{Hz})}{\text{TCA前置分周器}} - 1 = \frac{250 \times 10^{-3}(\text{s}) \times 3333333(\text{Hz})}{256} - 1 \approx 3254 = 0xCB6$$

```
TCA0_SINGLE_PER = 0xCB6;
```

5. CTRLAレジスタで前置分周器は'256'に設定されます。計数器を開始するため、使用者は同じレジスタの許可ビットを設定(1)しなければなりません。

```
TCA0_SINGLE_CTRLA = TCA_SINGLE_CLKSEL_DIV256_gc | TCA_SINGLE_ENABLE_bm;
```

図3-4. 制御A(CTRLA)レジスタ

ビット	7	6	5	4	3	2	1	0
					CLKSEL2~0			ENABLE
アクセス	R	R	R	R	R/W	R/W	R/W	R/W
リセット値	0	0	0	0	0	0	0	0

● ビット3~1 – CLKSEL2~0 : クロック選択 (Clock Select)

これらのビットはタイマ/カウンタに対するクロック周波数を選びます。

値	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
名称	DIV1	DIV2	DIV4	DIV8	DIV16	DIV64	DIV256	DIV1024
説明 ($f_{TCA} =$)	$f_{CLK_PER}/1$	$f_{CLK_PER}/2$	$f_{CLK_PER}/4$	$f_{CLK_PER}/8$	$f_{CLK_PER}/16$	$f_{CLK_PER}/64$	$f_{CLK_PER}/256$	$f_{CLK_PER}/1024$

6. 計時器が完全に構成設定された後、`sei()`マクロが全体割り込みを許可します。問題を避けるため、常に全体割り込みが禁止される時に周辺機能を構成設定してください。ISRではポートの入力レジスタの対応するビットを設定(1)することによって出力ピンを交互切り替えます。また、溢れ割り込みフラグも解除(0)されます。



助言: 割り込み要求フラグは各々のビット位置に'1'を書くことによってソフトウェアで解除(0)されなければなりません。

ISR (TCA0_OVF_vect)

```
{
  PORTA.OUTTGL = PIN0_bm;
  TCA0.SINGLE.INTFLAGS = TCA_SINGLE_OVF_bm;
}
```

図3-5. INTCTRLレジスタ

ビット	7	6	5	4	3	2	1	0
	CMP2			CMP1	CMP0			OVF
アクセス	R	R/W	R/W	R/W	R	R	R	R/W
リセット値	0	0	0	0	0	0	0	0

割り込み要求は対応する割り込み元が許可される時に生成され、割り込み要求フラグが設定(1)されます。フラグが設定されると直ぐに、マイクロコントローラは使用者によって書かれたISRからのコードの実行を開始します。割り込み要求は割り込み要求フラグが解除(0)されるまで活性(有効)に留まります。ISRのパラメータは割り込みベクタです。従って、使用者はISRが何の割り込み元に対応するかを指定することができます。TCA割り込みベクタの一覧が下で提供されます。プログラミング時、望む割り込みベクタを識別するのにIDEの自動完了機能を使うのが有用です。

図3-6. 標準動作で利用可能な割り込みベクタと供給元

変位	名称	ベクタ説明	条件
+\$00/+\$00	OVF	溢れ割り込み	計数器がTOP値に達して0に丸め
+\$02/+\$04	CMP0	比較チャネル0割り込み	計数器値と比較0レジスタ間的一致
+\$03/+\$06	CMP1	比較チャネル1割り込み	計数器値と比較1レジスタ間的一致
+\$04/+\$08	CMP2	比較チャネル2割り込み	計数器値と比較2レジスタ間的一致

7. ポートAの0番(PA0)ピンはそのポートの方向レジスタで対応するビットに'1'を書くことによって出力として設定されます。このGPIOは見ることが出来る出力を得るためだけに構成設定されますが、この動作形態ではTCAそれ自体と関係がありません。

```
PORTA.DIR |= PIN0_bm;
```



助言: 完全なコード例は「[追補](#)」章でも利用可能です。



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

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



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

4. 2傾斜PWM信号の生成

TCBのような他の計時器と比べて時にTCAの最も重要な特性の1つはPWM生成の多様性と精度です。使用者は応用の複雑さに従って様々な構成設定から選ぶことができます。TCAは1傾斜と2傾斜の両PWM生成動作で構成設定することができます。これは定位相(正しい位相のPWM)とより高い最大動作周波数(高速PWM)間の二律背反を許します。また、TCAは不具合なしのPWMを保証する緩衝の仕組みを持ちます。

TCAとTCBの両方は高い最大動作周波数でPWM信号を生成するのに使うことができます。TCAだけは選択可能な方向によって与えられる2傾斜PWM能力のため際どい応用で使うことができます。2傾斜PWMはデューティサイクルが変更される時にパルス中央位置を変更しません。従って、位相が常に一定です。

緩衝の仕組みは各比較レジスタだけでなく、定期レジスタに対する緩衝部も含まれます。これらの緩衝部の使用は回路短絡のために予期せぬ長いパルスをもたらす危険な応用で非常に重要です。更に、これらの緩衝部の存在は同じ計時器だけでも違う比較チャンネルを使う2つの周辺機能間での同期の消失を防ぐことができます。けれども、定期と比較のレジスタを直接更新することができることを考慮すると、使用者によって緩衝の仕組みを避けることができます。右の波形は緩衝と非緩衝の動作間の違いを図解します。

使用者が定期レジスタを直接変更する場合、計時器が渡された新しい閾値を既に持つことが可能で、故に最大値への計数を続けるでしょう。それは更なる問題を引き起こし得る異常な長いパルスを引き起こします。また、2つ以上の比較チャンネルが使われ、それらの1つが更新される場合、起動間の同期が失われるかもしれません。これら全ての有り得る問題を防ぐため、緩衝の仕組みの使用が必要とされます。

下は上で記述した緩衝の仕組みを使って50%のデューティサイクルで1kHzのPWM信号を生成するのにTCA実体をどう設定するか例です。

1. TCAはポート多重器の対応するレジスタでこの単位部の出力を異なるポートに配線するように設定することができ、この場合、既定ポートでもあるポートAが選ばれます。

```
PORTMUX.TCAROUTEA = PORTMUX_TCA0_PORTA_gc;
```

図4-3. TCA用PORTMUX制御

ビット	7	6	5	4	3	2	1	0
						TCA02~0		
アクセス	R	R	R	R	R	R/W	R/W	R/W
リセット値	0	0	0	0	0	0	0	0

- ビット2~0 - TCA02~0 : TCA0出力代替 (TCA0 output)

TCA0用の代替ピン位置を選ぶにはこれらのビットを書いてください。

値	000	001	010	011	100	101	110	111
名称	PORTA	PORTB	PORTC	PORTD	PORTE	PORTF	-	-
説明	PA5~0	PB5~0	PC5~0	PD5~0	PE5~0	PF5~0	(予約)	(予約)

2. CTRLBレジスタは比較チャンネルの許可ビットと波形生成動作を決めるビット領域を含みます。この例では2傾斜PWM動作と共にチャンネル0が使われます。

```
TCA0.SINGLE.CTRLB = TCA_SINGLE_CMPOEN_bm | TCA_SINGLE_WGMODE_DSBOTTOM_gc;
```

図4-1. 緩衝なし2傾斜動作

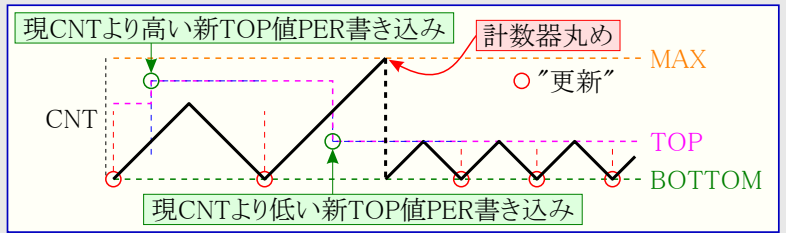


図4-2. 緩衝使用周期変更

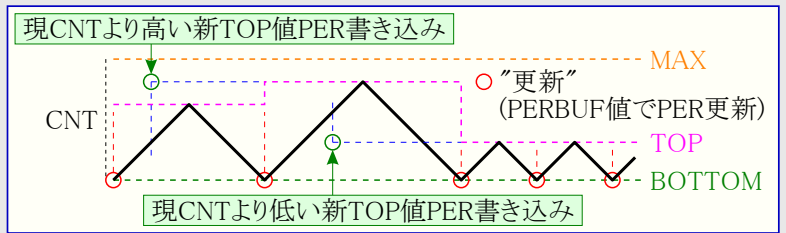


図4-4. CTRLBレジスタ

ビット	7	6	5	4	3	2	1	0
	CMP2EN		CMP1EN	CMP0EN	ALUPD	WGMODE2~0		
アクセス	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
リセット値	0	0	0	0	0	0	0	0

● ビット2~0 – WGMODE2~0 : 波形生成動作 (Waveform Generation Mode)

これらのビットは波形生成動作を選び、計数器の計数進行、TOP値、UPDATE条件、割り込み条件、生成される波形の形式を制御します。

標準形態の動作では波形生成が全く実行されません。他の全ての動作形態に対して、対応する比較n許可(CMPnEN)ビットがこれを許可するように設定されていれば、波形生成部からの結果がポートピンに直結されるだけです。ポートピンの方向は出力として設定されなければなりません。

値	000	001	010	011	100	101	110	111
名称	NORMAL	FRQ	-	SINGLESLOPE	-	DSTOP	DSBOTH	DSBOTTOM
説明	動作	標準	周波数	1傾斜PWM	(予約)	2傾斜PWM	2傾斜PWM	2傾斜PWM
	TOP	PER	CMP0	PER	(予約)	PER	PER	PER
	更新	TOP	TOP	BOTTOM	(予約)	BOTTOM	BOTTOM	BOTTOM
	OVF	TOP	TOP	BOTTOM	(予約)	TOP	TOPとBOTTOM	BOTTOM

3. 事象の代わりにクロック刻時を計数するように計時器を設定するため、EVCTRLレジスタのCNTEIビットは'0'に設定されます。これがCNTEIビットの既定値であることは言及する価値があります。

```
TCA0_SINGLE_EVCTRL &= ~(TCA_SINGLE_CNTEI_bm);
```

図3-3. EVCTRLレジスタ

ビット	7	6	5	4	3	2	1	0
						EVACT1,0		CNTEI
アクセス	R	R	R	R	R	R/W	R/W	R/W
リセット値	0	0	0	0	0	0	0	0

4. PERBUFは定期(PER)レジスタの緩衝部です。以下の式を使ってPWM信号の周波数を設定するのに使われます。

$$2\text{傾斜PWM周波数(Hz)} = \frac{f_{CLK}(\text{Hz})}{2 \times \text{TCA前置分周器} \times \text{TCA定期}}$$

この例に対して目的値を考察すると、

$$\text{TCA定期} = \frac{f_{CLK}(\text{Hz})}{2 \times \text{TCA前置分周器} \times 2\text{傾斜PWM周波数(Hz)}} = \frac{3333333}{2 \times 4 \times 1000} \approx 416 = 0x1A0$$

```
TCA0_SINGLE_PERBUF = 0x01A0;
```

5. また、デューティサイクルを設定するためにその緩衝部を使って比較レジスタが更新されます。比較レジスタの値は50%のデューティサイクルが望まれるため、定期レジスタの値の半分です。

```
TCA0_SINGLE_CMPOBUF = 0x00D0;
```

6. CTRLAレジスタのCLKSELビット領域を変更することによって前置分周器を4に設定してください。計数器を開始するため、使用者は同じレジスタの許可ビットを設定(1)しなければなりません。

```
TCA0_SINGLE_CTRLA = TCA_SINGLE_CLKSEL_DIV4_gc | TCA_SINGLE_ENABLE_bm;
```

図4-6. CTRLAレジスタ

ビット	7	6	5	4	3	2	1	0
					CLKSEL2~0			ENABLE
アクセス	R	R	R	R	R/W	R/W	R/W	R/W
リセット値	0	0	0	0	0	0	0	0

● ビット3~1 – CLKSEL2~0 : クロック選択 (Clock Select)

これらのビットはタイマ/カウンタに対するクロック周波数を選びます。

図4-6 (続き). CTRLALレジスタ

値	000	001	010	011	100	101	110	111
名称	DIV1	DIV2	DIV4	DIV8	DIV16	DIV64	DIV256	DIV1024
説明 ($f_{TCA} =$)	$f_{CLK_PER}/1$	$f_{CLK_PER}/2$	$f_{CLK_PER}/4$	$f_{CLK_PER}/8$	$f_{CLK_PER}/16$	$f_{CLK_PER}/64$	$f_{CLK_PER}/256$	$f_{CLK_PER}/1024$

7. その後、ポートAの0番(PA0)ピンはそのポートの方向レジスタで対応するビットに'1'を書くことによって出力として設定されます。

```
PORTA_DIR = PIN0_bm;
```



助言: 完全なコード例は「[追補](#)」章でも利用可能です。



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

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

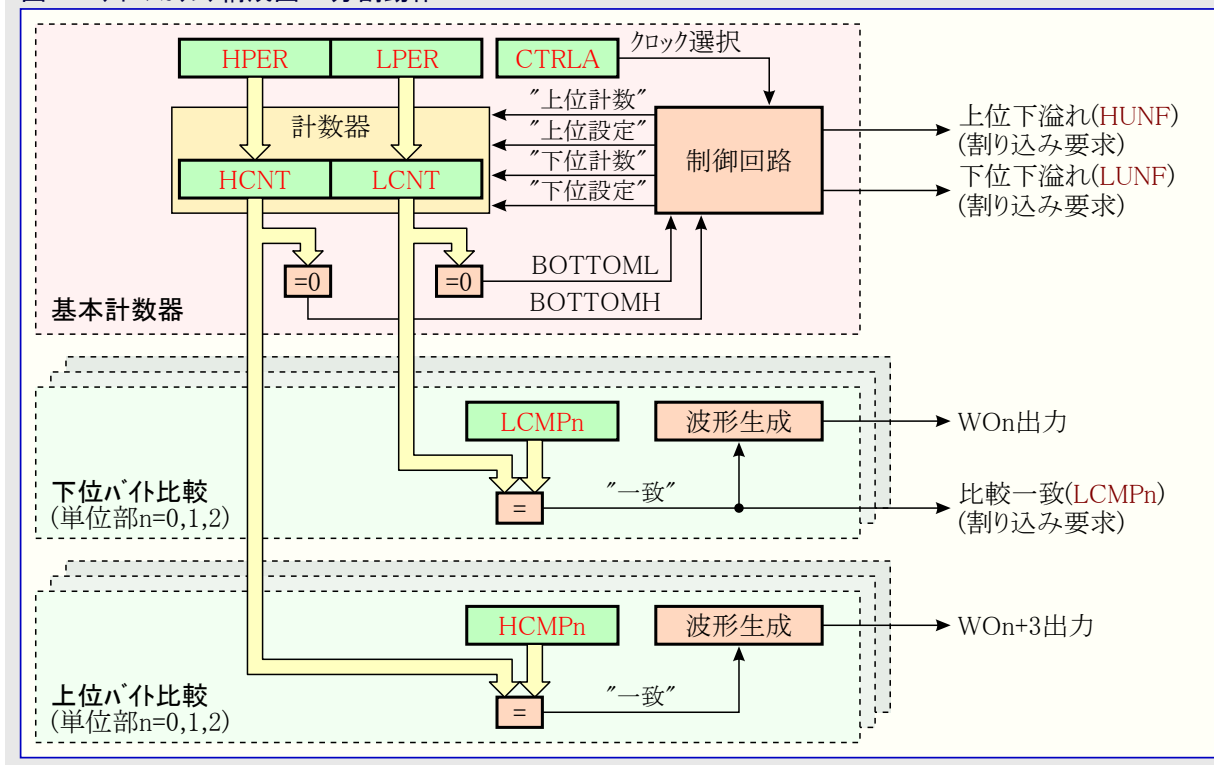


GitHubでAVR128DA48コード例を見てください。
貯蔵庫を閲覧するにはクリックしてください。

5. 分割動作で2つのPWM信号の生成

TCA実体は2つの完全に独立した8ビット計時器に分割することができます。この機能は波形生成応用で極めて役立つ高度な柔軟性を提供します。高精度信号が必要とされる場合を除き、応用の殆どは8ビット信号生成器を使って設計することができ、設計にもう1つの発生器を追加する可能性は大きな利点であり得ます。とは言え、分割動作には計数レジスタ寸法よりも大きな制限があります。最も重要なものは両8ビット計時器が下降計数任意選択しか持たないことで、故に2傾斜PWM動作は利用不可になります。また、緩衝の仕組みも使うことができず、計時器はクロック刻時だけで、もはや事象を計数できません。更に、上位側比較レジスタに対しては割り込みやフラグがありません。これらの制限にも関わらず、分割動作は多数の計時器に対する要求がある時に魅力的であり得ます。分割動作でのTCAの構成図が次で提供されます。

図5-1. タイマ/カウンタ構成図 - 分割動作



この動作は異なる周波数と異なるデューティサイクルを持つ2つのPWM信号を生成することによって推薦されます。

1. ポート多重器でTCAに対応するレジスタはこの単位部出力を違うポートに配線するように設定することができます。この場合、既定ポートであるポートAが選ばれます。

PORTMUX.TCAROUTEA = PORTMUX_TCA0_PORTA_gc;

図5-2. TCAROUTEAレジスタ

ビット	7	6	5	4	3	2	1	0
						TCA02~0		
アクセス	R	R	R	R	R	R/W	R/W	R/W
リセット値	0	0	0	0	0	0	0	0

- ビット2~0 - TCA02~0 : TCA0出力代替 (TCA0 output)

TCA0用の代替ピン位置を選ぶにはこれらのビットを書いてください。

値	000	001	010	011	100	101	110	111
名称	PORTA	PORTB	PORTC	PORTD	PORTE	PORTF	-	-
説明	PA5~0	PB5~0	PC5~0	PD5~0	PE5~0	PF5~0	(予約)	(予約)

2. CTRLDレジスタの対応するビットを設定(1)することによって分割動作を許可してください。

TCA0.SPLIT.CTRLD = TCA_SPLIT_SPLITM_bm;

図5-3. CTRLDレジスタ

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

- ビット0 - SPLITM : 分割動作許可 (Enable Split Mode)

このビットはタイマ/カウンタを分割動作形態に設定します。その後は2つの8ビットタイマ/カウンタとして動きます。標準16ビット動作と比べてレジスタ割り当てが変わります。

3. CTRLBレジスタは比較チャネルの許可ビットを含みます。この例では計時器のチャネル0の下位バイトと計時器のチャネル0の上位バイトが使われます。

TCA0.SPLIT.CTRLB = TCA_SPLIT_HCMP0EN_bm | TCA_SPLIT_LCMP0EN_bm;

図5-4. CTRLBレジスタ - 分割動作

ビット	7	6	5	4	3	2	1	0
		HCMP2EN	HCMP1EN	HCMP0EN		LCMP2EN	LCMP1EN	LCMP0EN
アクセス	R	R/W	R/W	R/W	R	R/W	R/W	R/W
リセット値	0	0	0	0	0	0	0	0

- ビット4 - HCMP0EN : 上位バイト比較0許可 (High-byte Compare 0 Enable)

LCMP0ENをご覧ください。

- ビット0 - LCMP0EN : 下位バイト比較0許可 (Low-byte Compare 0 Enable)

FRQまたはPWM波形生成動作形態でのHCMPnEN/LCMPnENビット設定(1)は対応するWOnピンに対するポート出力(PORTx.OUT)レジスタを無効にします。

4. この動作形態では、定期(PER)レジスタと比較(CMPn)レジスタが半分に分割されます。定期レジスタの各々半分が各々PWM信号の周波数を決めます。望む周波数値を使い、定期レジスタ値は以下の式から推定することができます。

$$1 \text{ 傾斜PWM周波数(Hz)} = \frac{f_{CLK}(\text{Hz})}{\text{TCA前置分周器} \times (\text{TCA定期} + 1)}$$

この例に対して目的値を考察すると、

$$\text{TCA定期1} = \frac{f_{CLK}(\text{Hz})}{\text{TCA前置分周器} \times 1 \text{ 傾斜PWM 1周波数(Hz)}} - 1 = \frac{3333333}{16 \times 1000} - 1 \approx 207 = 0xCF$$

$$\text{TCA定期2} = \frac{f_{CLK}(\text{Hz})}{\text{TCA前置分周器} \times 1 \text{ 傾斜PWM 2周波数(Hz)}} - 1 = \frac{3333333}{16 \times 3000} - 1 \approx 68 = 0x44$$

これは以下の行のコードに変換できます。

```
TCA0.SPLIT.LPER = 0xCF;
TCA0.SPLIT.HPER = 0x44;
```

5. 比較レジスタの各半分は各々のPWM信号のデューティサイクルを決めます。

```
TCA0.SPLIT.LCMP0 = 0x68;
TCA0.SPLIT.HCMP0 = 0x11;
```

6. CTRLAレジスタで前置分周器が16に設定されます。計数器を開始するため、使用者は同じレジスタで許可ビットを設定(1)しなければなりません。

```
TCA0.SPLIT.CTRLA = TCA_SPLIT_CLKSEL_DIV16_gc | TCA_SPLIT_ENABLE_bm;
```

7. 提供される初期化コードは分割動作でTCAを構成設定する簡単な方法を解説しますが、いくつかの言及が行われなければなりません。1傾斜PWM動作だけが利用可能な波形生成動作です。また、標準動作から分割動作へ切り替える前に計数器を停止してハードリセットを行うことが推奨されます。例が下で提供されます。計数器を停止するため、CTRLAレジスタの許可ビットを解除(0)してください。その後、CTRLSETレジスタの指令ビット領域に使用者がハードリセット指令の符号を書きます。

```
void TCA0_hardReset(void)
{
    TCA0.SINGLE.CTRLA &= ~(TCA_SINGLE_ENABLE_bm);
    TCA0.SINGLE.CTRLESET = TCA_SINGLE_CMD_RESET_gc;
}
```

8. その後、ポートAの0と3(PA0とPA3)はポートの方向レジスタの各対応ビットに'1'を書くことによって出力として設定されます。

```
PORTA.DIR = PIN0_bm | PIN3_bm;
```



助言: 完全なコード例は「[追補](#)」章でも利用可能です。



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

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



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

6. 参考文献

TCA動作形態についてより多くの情報を見つけるには以下のリンクを使ってください。

- ATmega4809製品頁：<https://www.microchip.com/wwwproducts/en/ATMEGA4809>
- megaAVR® 0系手引書 (DS40002015)
- ATmega809/1609/3209/4809 – 48ピン megaAVR® 0系データシート (DS40002016)
- ATmega4809 Xplained Proウェブ頁：<https://www.microchip.com/developmenttools/ProductDetails/atmega4809-xpro>
- AVR128DA48製品頁：www.microchip.com/wwwproducts/en/AVR128DA28
- AVR128DA48 Curiosity Nano評価キットウェブ頁：www.microchip.com/Developmenttools/ProductDetails/DM164151
- AVR128DA28/32/48/64 AVR® DA系データシート (DS40002183)
- AVR® DA系での開始に際して (DS00003429)

7. 改訂履歴

文書改訂	日付	注釈
A	2018年11月	初版文書公開
B	2021年1月	GitHub貯蔵庫リンク更新。「AVR® DA系概要」追加、「参考文献」、「改訂履歴」章更新。各使用事例に対してAVR128DA48で動くMCC版を追加。その他些細な編集上の修正。
C	2021年7月	周期的割り込み動作の使用の章でTCA周期計算用の式を修正

8. 追補

例8-1. 周期的割り込み動作の使用全コード例

```
#define PERIOD_EXAMPLE_VALUE      (0x0CB6)

#include <avr/io.h>
#include <avr/interrupt.h>
/* 3.33MHz既定クロックを使用 */

void TCA0_init(void);
void PORT_init(void);

void TCA0_init(void)
{
    /* 溢れ割り込み許可 */
    TCA0.SINGLE.INTCTRL = TCA_SINGLE_OVF_bm;

    /* 標準動作設定 */
    TCA0.SINGLE.CTRLB = TCA_SINGLE_WGMODE_NORMAL_gc;

    /* 事象計数禁止 */
    TCA0.SINGLE.EVCTRL &= ~(TCA_SINGLE_CNTEI_bm);

    /* 周期設定 */
    TCA0.SINGLE.PER = PERIOD_EXAMPLE_VALUE;

    TCA0.SINGLE.CTRLA = TCA_SINGLE_CLKSEL_DIV256_gc /* クロック元設定(sys_clk/256) */
                      | TCA_SINGLE_ENABLE_bm;     /* 計時器開始 */
}

void PORT_init(void)
{
    /* ポートAの0(PA0)ピンを出力として設定 */
    PORTA.DIR |= PIN0_bm;
}

ISR(TCA0_OVF_vect)
{
    /* ポートAの0(PA0)ピンを交互切り替え */
    PORTA.OUTTGL = PIN0_bm;

    /* 割り込み要求フラグは手動で解除(0)されなければなりません。 */
    TCA0.SINGLE.INTFLAGS = TCA_SINGLE_OVF_bm;
}

int main(void)
{
    PORT_init();
    TCA0_init();

    /* 全体割り込み許可 */
    sei();

    while (1)
    {
        ;
    }
}
```

例8-2. 2傾斜PWM信号の生成全コード例

```

#define PERIOD_EXAMPLE_VALUE      (0x01A0)
#define DUTY_CYCLE_EXAMPLE_VALUE (0x00D0)

#include <avr/io.h>
/* 3.33MHz既定クロックを使用 */

void TCA0_init(void);
void PORT_init(void);

void TCA0_init(void)
{
    /* ポートAでの波形出力に設定 */
    PORTMUX.TCAROUTEA = PORTMUX_TCA0_PORTA_gc;

    TCA0.SINGLE.CTRLB = TCA_SINGLE_CMPOEN_bm          /* 比較チャンネル0許可 */
                      | TCA_SINGLE_WGMODE_DSBOTTOM_gc; /* 2傾斜PWM動作に設定 */

    /* 事象計数禁止 */
    TCA0.SINGLE.EVCTRL &= ~(TCA_SINGLE_CNTEI_bm);

    /* PWM周波数とデューティサイクル(50%)を設定 */
    TCA0.SINGLE.PERBUF = PERIOD_EXAMPLE_VALUE;
    TCA0.SINGLE.CMPOBUF = DUTY_CYCLE_EXAMPLE_VALUE;

    TCA0.SINGLE.CTRLA = TCA_SINGLE_CLKSEL_DIV4_gc     /* クロック元設定(sys_clk/4) */
                      | TCA_SINGLE_ENABLE_bm;       /* 計時器開始 */
}

void PORT_init(void)
{
    /* ポートAの0(PA0)ピンを出力として設定 */
    PORTA.DIR |= PIN0_bm;
}

int main(void)
{
    PORT_init();
    TCA0_init();

    /* あなたの応用コードで置き換えてください。 */
    while (1)
    {
        ;
    }
}

```

例8-3. 分割動作で2つのPWM信号の生成全コード例

```

#define PERIOD_EXAMPLE_VALUE_L    (0xCF)
#define PERIOD_EXAMPLE_VALUE_H    (0x44)
#define DUTY_CYCLE_EXAMPLE_VALUE_L (0x68)
#define DUTY_CYCLE_EXAMPLE_VALUE_H (0x11)

#include <avr/io.h>
/* 3.33MHz既定クロックを使用 */

void TCA0_init(void);
void PIN_init(void);
void TCA0_hardReset(void);

void TCA0_init(void)
{
    /* ポートAでの波形出力に設定 */
    PORTMUX.TCAROUTEA = PORTMUX.TCA0_PORTA_gc;

    /* 分割動作許可 */
    TCA0.SPLIT.CTRLD = TCA_SPLIT_SPLITM_bm;

    TCA0.SPLIT.CTRLB = TCA_SPLIT_HCMPOEN_bm          /* 上位バイト用比較チャンネル0許可 */
                    | TCA_SPLIT_LCMPOEN_bm;        /* 下位バイト用比較チャンネル0許可 */

    /* PWM周波数とデューティサイクルを設定 */
    TCA0.SPLIT.LPER = PERIOD_EXAMPLE_VALUE_L;
    TCA0.SPLIT.LCMP0 = DUTY_CYCLE_EXAMPLE_VALUE_L;
    TCA0.SPLIT.HPER = PERIOD_EXAMPLE_VALUE_H;
    TCA0.SPLIT.HCMP0 = DUTY_CYCLE_EXAMPLE_VALUE_H;

    TCA0.SPLIT.CTRLA = TCA_SPLIT_CLKSEL_DIV16_gc    /* クロック元設定(sys_clk/16) */
                    | TCA_SPLIT_ENABLE_bm;        /* 計時器開始 */
}

void PIN_init(void)
{
    PORTA.DIR |= PIN0_bm          /* ポートAの0(PA0)ピンを出力として設定 */
              | PIN3_bm;        /* ポートAの3(PA3)ピンを出力として設定 */
}

/* 単一動作から分割動作へ切り替える時に使われなければなりません。 */
void TCA0_hardReset(void)
{
    /* 計時器停止 */
    TCA0.SINGLE.CTRLA &= ~(TCA_SINGLE_ENABLE_bm);

    /* ハードリセット強制 */
    TCA0.SINGLE.CTRLESET = TCA_SINGLE_CMD_RESET_gc;
}

int main(void)
{
    PIN_init();
    TCA0_init();

    while (1)
    {
        ;
    }
}

```

Microchipウェブ サイト

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

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

製品変更通知サービス

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

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

お客様支援

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

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

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

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

Microchipデバイスコード保護機能

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

- Microchip製品はそれら特定のMicrochipデータシートに含まれる仕様に合致します。
- Microchipは意図した方法と通常条件下で使われる時に、その製品系統が安全であると考えます。
- Microchipデバイスのコード保護機能を破ろうとする試みに使われる不正でおそらく違法な方法があります。当社はこれらの方法がMicrochipのデータシートに含まれた動作仕様外の方法でMicrochip製品を使うことが必要とされると確信しています。これらのコード保護機能を破ろうとする試みは、おそらく、Microchipの知的財産権に違反することなく達成することはできません。
- Microchipはそのコードの完全性について心配されている何れのお客様とも共に働きたいと思えます。
- Microchipや他のどの半導体製造業者もそのコードの安全を保証することはできません。コード保護は製品が”破ることができない”ことを当社が保証すると言うことを意味しません。コード保護は常に進化しています。Microchipは当社製品のコード保護機能を継続的に改善することを約束します。Microchipのコード保護機能を破る試みはデジタル ミレニアム著作権法に違反するかもしれません。そのような行為があなたのソフトウェアや他の著作物に不正なアクセスを許す場合、その法律下の救済のために訴権を持つかもしれません。

法的通知

この刊行物に含まれる情報はMicrochip製品を使って設計する唯一の目的のために提供されます。デバイス応用などに関する情報は皆さまの便宜のためにだけ提供され、更新によって取り換えられるかもしれません。皆さまの応用が皆さまの仕様に合致するのを保証するのは皆さまの責任です。

この情報はMicrochipによって「現状そのまま」で提供されます。Microchipは非侵害、商品性、特定目的に対する適合性の何れの黙示的保証やその条件、品質、性能に関する保証を含め、明示的にも黙示的にもその情報に関連して書面または表記された書面または黙示の如何なる表明や保証もしません。

如何なる場合においても、Microchipは情報またはその使用に関連するあらゆる種類の間接的、特別的、懲罰的、偶発的または結果的な損失、損害、費用または経費に対して責任を負わないものとします。法律で認められている最大限の範囲で、情報またはその使用に関連する全ての請求に対するMicrochipの全責任は、もしあれば、情報のためにMicrochipへ直接支払った料金を超えないものとします。生命維持や安全応用でのMicrochipデバイスの使用は完全に購入者の危険性で、購入者はそのような使用に起因する全ての損害、請求、訴訟、費用からMicrochipを擁護し、補償し、免責することに同意します。他に言及されない限り、Microchipのどの知的財産権下でも暗黙的または違う方法で許認可は譲渡されません。

商標

Microchipの名前とロゴ、Mmicrochipロゴ、Adaptec、AnyRate、AVR、AVRロゴ、AVR Freaks、BesTime、BitCloud、chipKIT、chipKITロゴ、CryptoMemory、CryptoRF、dsPIC、FlashFlex、flexPWR、HELDO、IGLOO、JukeBlox、KeeLoq、Kleer、LANCheck、LinkMD、maXStylus、maXTouch、MediaLB、megaAVR、Microsemi、Microsemiロゴ、MOST、MOSTロゴ、MPLAB、OptoLyzer、PacTime、PIC、picoPower、PICSTART、PIC32ロゴ、PolarFire、Prochip Designer、QTouch、SAM-BA、SenGenuity、SpyNIC、SST、SSTロゴ、SuperFlash、Symmetricom、SyncServer、Tachyon、TimeSource、tinyAVR、UNI/O、Vectron、XMEGAは米国と他の国に於けるMicrochip Technology Incorporatedの登録商標です。

AgileSwitch、APT、ClockWorks、The Embedded Control Solutions Company、EtherSynch、FlashTec、Hyper Speed Control、Hyper Light Load、IntelliMOS、Libero、motorBench、mTouch、Powermite 3、Precision Edge、ProASIC、ProASIC Plus、ProASIC Plusロゴ、Quiet-Wire、SmartFusion、SyncWorld、Temux、TimeCesium、TimeHub、TimePictra、TimeProvider、Vite、WinPath、ZLは米国に於けるMicrochip Technology Incorporatedの登録商標です。

Adjacent Key Suppression、AKS、Analog-for-the-Digital Age、Any Capacitor、AnyIn、AnyOut、Augmented Switching、BlueSky、BodyCom、CodeGuard、CryptoAuthentication、CryptoAutomotive、CryptoCompanion、CryptoController、dsPICDEM、dsPICDEM.net、Dynamic Average Matching、DAM、ECAN、Espresso T1S、EtherGREEN、IdealBridge、In-Circuit Serial Programming、ICSP、INICnet、Intelligent Paralleling、Inter-Chip Connectivity、JitterBlocker、maxCrypto、maxView、memBrain、Mindi、MiWi、MPASM、MPF、MPLAB Certifiedロゴ、MPLIB、MPLINK、MultiTRAK、NetDetach、Omniscient Code Generation、PICDEM、PICDEM.net、PICkit、PICKtail、PowerSmart、PureSilicon、QMatrix、REAL ICE、Ripple Blocker、RTAX、RTG4、SAM-ICE、Serial Quad I/O、simpleMAP、SimpliPHY、SmartBuffer、SMART-I.S.、storClad、SQI、SuperSwitcher、SuperSwitcher II、Switchtec、SynchronPHY、Total Endurance、TSHARC、USBCheck、VariSense、VectorBlox、VeriPHY、ViewSpan、WiperLock、XpressConnect、and ZENAは米国と他の国に於けるMicrochip Technology Incorporatedの商標です。

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

Adaptecロゴ、Frequency on Demand、Silicon Storage Technology、Symmcomは他の国に於けるMicrochip Technology Inc.の登録商標です。

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

ここで言及した以外の全ての商標はそれら各々の会社の所有物です。

© 2020年、Microchip Technology Incorporated、米国印刷、不許複製

品質管理システム

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

日本語© HERO 2021.

本技術概説はMicrochipのTB3217技術概説(DS90003217C-2021年7月)の翻訳日本語版です。日本語では不自然となる重複する形容表現は省略されている場合があります。日本語では難解となる表現は大幅に意識されている部分もあります。必要に応じて一部加筆されています。頁割の変更により、原本より頁数が少なくなっています。

必要と思われる部分には()内に英語表記や略称などを残す形で表記しています。

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

世界的な販売とサービス

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