
TCBでの開始に際して

序説

著者: Marius Nicolae, Microchip Technology Inc.

tinyAVR[®] 0系、tinyAVR 1系、megaAVR[®] 0系のマイクロ コントローラは広い領域の応用を網羅することをそれらに許す強力な計時器を装備しています。タイマ/カウンタB型(TCB:Timer/Counter Type B)は周期的な割り込みから8ビットPWMまたは制限時間超過まで様々な機能と動作形態を提供します。TCBの様々な動作形態は事象システムとの相互関係で使うことができます。この技術概要の範囲は3つの動作形態の構成設定を示すことです。

- **8ビットPWM動作でTCBを使用:**

TCBはこの動作形態に構成設定され、1秒の周期に対して50%のデューティ サイクルを持つPWM信号を生成します。

- **制限時間検査動作でTCBを使用:**

TCBはこの動作形態に構成設定され、入力として構成設定されたGPIO(汎用入出力)ピンによって生成された信号時間(端から端まで)を測定するのに使われます。制限時間(端から端までの周期)を超過する場合に割り込みが生成されます。

- **休止動作でTCBを使用:**

TCBは例えAVR[®]マイクロ コントローラがスタンバイ休止動作でも、周期的割り込みを生成するように構成設定されます。

注: 各例に於いて、GPIOは入力(事象システム入力信号)または出力(例えば、割り込み発生毎でのピン交互切り替え)のどちらかに使われます。コード例はATmega4809 Xplained Pro (ATMEGA4809-XPRO)で開発されましたが、tiny AVR[®] 0系、tinyAVR 1系、megaAVR[®] 0系全体に対して適合します。

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

目次

序説	1
1. 関連デバイス	3
1.1. tinyAVR® 0系統	3
1.2. tinyAVR® 1系統	3
1.3. megaAVR® 0系統	3
2. 概要	4
3. 8ビットPWM動作でTCBを使用	4
4. 制限時間検査動作でTCBを使用	8
5. 休止動作でTCBを使用	11
6. 参照	13
7. 追補	13
Microchipウェブ サイト	18
お客様への変更通知サービス	18
お客様支援	18
Microchipデバイス コード保護機能	18
法的通知	18
商標	19
DNVによって認証された品質管理システム	19
世界的な販売とサービス	20

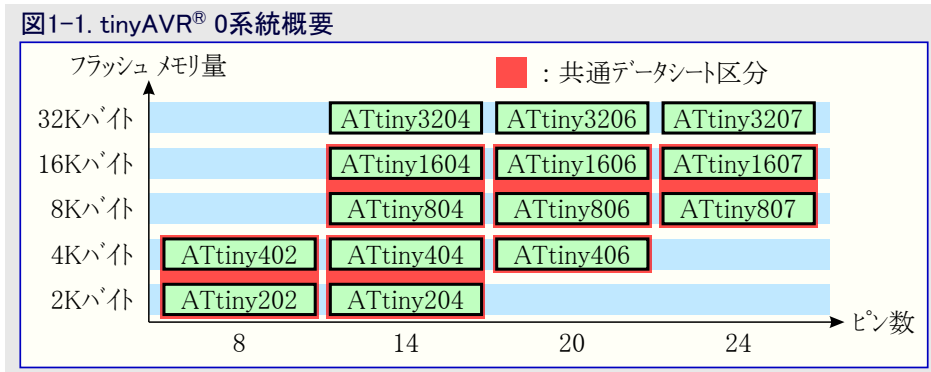
1. 関連デバイス

本章はこの資料に関連するデバイスを一覧にします。

1.1. tinyAVR[®] 0系統

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

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

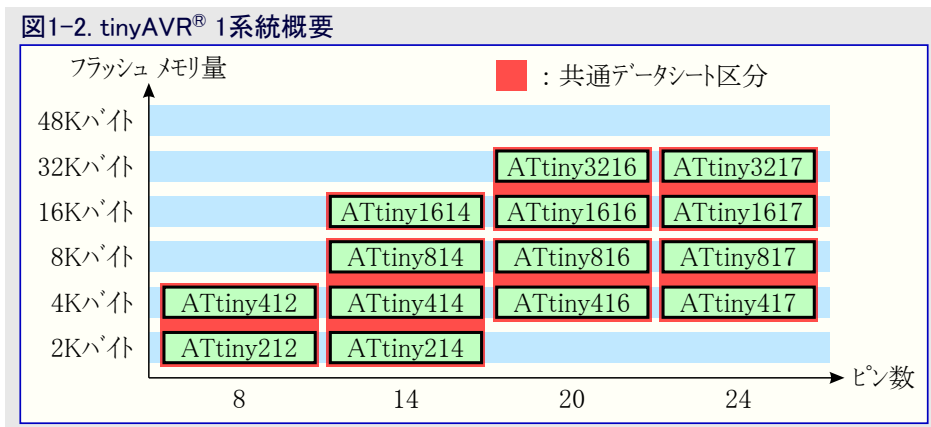


異なるフラッシュメモリ量を持つデバイスは一般的に異なるSRAMとEEPROMの量を持ちます。

1.2. tinyAVR[®] 1系統

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

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



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

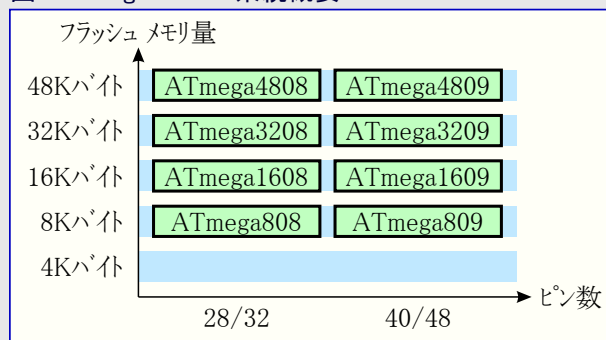
1.3. megaAVR[®] 0系統

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

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

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

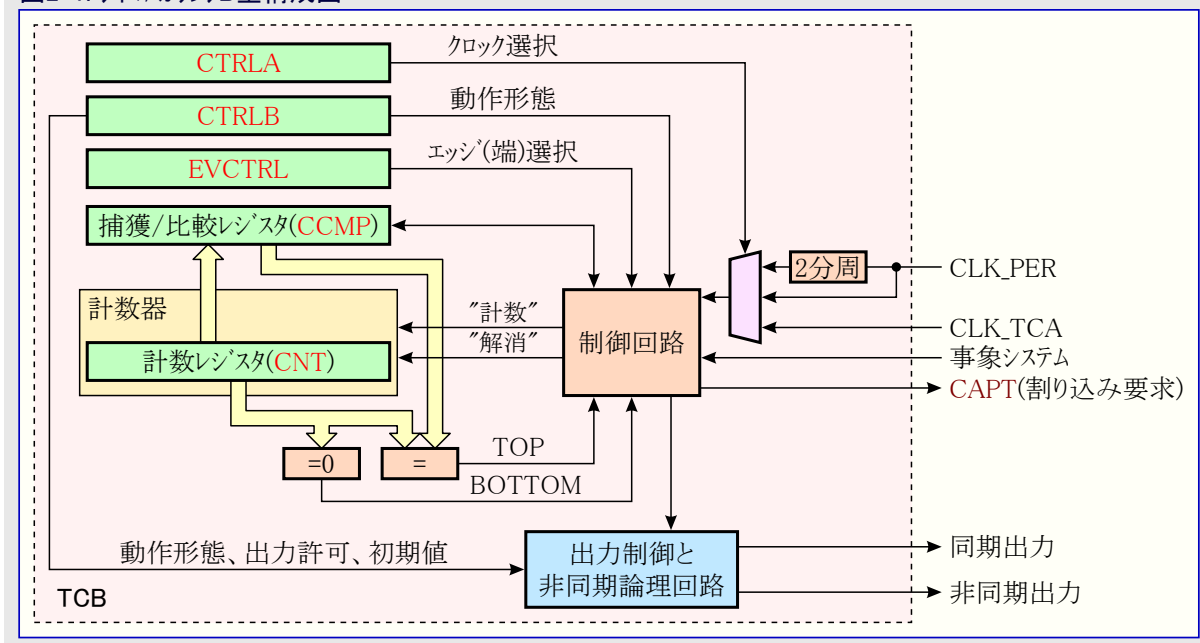
図1-3. megaAVR[®] 0系統概要



2. 概要

16ビット タイマ/カウンタB型(TCB)の能力は周波数と波形の生成、デジタル信号の時間と周波数の測定と共に事象での計数捕獲を含みます。TCBは基本計数器と各動作形態が独特な機能を提供する8つの異なる動作形態の1つに設定することができる制御論理回路から成ります。下で示されるように、基本計数器は任意選択の前置分周を持つ周辺クロックによってクロック駆動されます。

図2-1. タイマ/カウンタB型構成図



3. 8ビットPWM動作でTCBを使用

使用事例説明: TCB(TCB3実体)は8ビットPWM動作に構成設定され、50%のデューティ サイクルで1秒周期のPWM信号を生成します。GP IO(ポートBの5番、PB5)ピンはこの信号を見せるための出力として使われます。

結果: TCBは1秒の周期に対して50%のデューティ サイクルPWM信号を生成します。基板上のLED(PB5ピン)は500ms毎に交互切り替えます。

この計数器は16ビット捕獲/比較(TCBn.CCMPHとTCBn.CCMPL)レジスタのレジスタ対の各々が個別比較レジスタとして使われる8ビットPWM動作で動くように構成設定することができます。計数器は0からCCMPLまで継続的に計数し、出力はBOTTOMで設定(1)され、計数器がCCMPHに達する時に解除(0)されます。

この周辺機能がPWM動作で許可されると、捕獲/比較レジスタの値変更は出力を変更しますが、その遷移は無効な値を出力するかもしれません。故に以下が推奨されます。

1. 周辺機能を禁止してください。
2. 捕獲/比較レジスタ(CCMPHとCCMPL)に値を書いてください。
3. 計数レジスタに\$0000を書いてください。
4. 単位部を再許可してください。

CCMPHは出力がHighに駆動される周期数、一方でCCMPL+1は出力パルスの周期です。

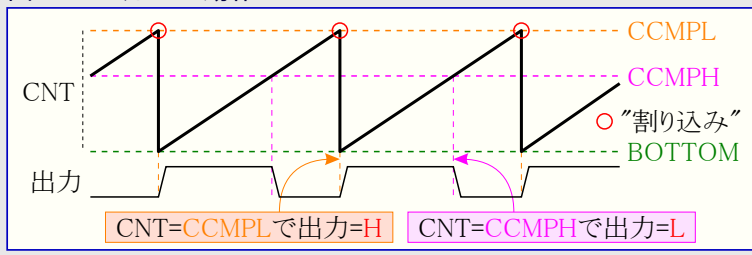
例えば、計数器への入力として扱う256Hzの周波数のクロックに対して、結果は1秒周期です(CCMPHは0~255までの値を持つことができる8ビットレジスタです)。

PWM信号を可視化するため出力としてピンを構成設定

PWMを可視化するため、ピンは出力動作で構成設定されます。以下のコードはLowを出力するようにPB5を設定します。

```
PORTB_DIR |= PIN5_bm;
PORTB_OUT |= PIN5_bm;
```

図3-1. 8ビットPWM動作



システム クロック構成設定

図2-1の構成図に従い、TCBには以下のように2つの主なクロック元があります。

- CLK_PER (主クロックのCLK_MAINから派生される周辺クロック)
- CLK_TCA (CLK_PERから派生することができるTCAクロック)

このTCBの使用事例の目的に対しては、TCBの実体3(TCB3)とCLK_PERクロック元が使われます。1秒周期を得るため、入力クロックは可能な限り低くしなければなりません。以下の構成設定が行われなければなりません。

- CLKSELに対して内部32kHz超低電力発振器が選ばれなければなりません。
- クロック前置分周器(PEN)が許可されなければなりません。
- 最高前置分周(PDIV64)が使われなければなりません。

TCBに対するクロック元として32kHz内部発振器を使うため、使用者は以下のレジスタと以下のレジスタ内のビットまたはビット領域を構成設定しなければなりません。

図3-2. MCLKCTRLBレジスタ構成設定

ビット	7	6	5	4	3	2	1	0
	CLKOUT			PDIV3~0				PEN
アクセス種別	R	R	R	R/W	R/W	R/W	R/W	R/W
リセット値	0	0	0	1	0	0	0	1

● ビット4~1 – PDIV3~0 : 前置分周器分周値 (Prescaler Division)

前置分周器許可(PEN)ビットが'1'を書かれると、これらのビットは主クロック前置分周器の分周比を定義します。

これらのビットは応用の必要条件に適合させるようにシステムのクロック周波数を変えるために走行時の間に書くことができます。

使用者ソフトウェアは結果のCLK_PER周波数が許された最大(デバイスのデータシートの「電気的特性」をご覧ください)を決して超えないような、正しい入力周波数(CLK_MAIN)の構成設定と前置分周器設定を保証しなければなりません。

値	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
分周数	2	4	8	16	32	64	(予約)	(予約)	6	10	12	24	48	(予約)	(予約)	

● ビット0 – PEN : 前置分周器許可 (Prescaler Enable)

前置分周器が許可されるにはこのビットが'1'を書かれなければなりません。許可されると、前置分周器分周値(PDIV)ビット領域によって分周比が選ばれます。

このビットが'0'を書かれると、主クロックはPDIVの値に関わらず、分周なしを通して渡されます(CLK_PER=CLK_MAIN)。

主クロックと前置分周器の構成設定レジスタ(CLKCTRL.MCLKCTRLAとCLKCTRL.MCLKCTRLB)はそれらのレジスタを変更するのに制限時間書き込み手順を使う構成設定変更保護(CCP: Configuration Change Protection)機構によって保護されます。これらのレジスタに書くには、最初に正しい鍵がCPU.CCPレジスタに書かれ、後続して4 CPU命令内に保護されたビットへ書き込みアクセスされなければなりません。

CPU.CCPレジスタに書かれなければならない鍵は下のコードの行に変換される'IOREG'です。より多くの詳細についてはmegaAVR 0系データシートで構成設定変更保護(CCP)を調べてください。

注: CPU_CCPレジスタは1つの読み書き命令だけを許し、その変更は鍵書き込み後に続く4周期内に行われなければなりません。

```
CPU_CCP = CCP_IOREG_gc;
```

望む周波数が可能な最低であるため、前置分周器(PEN)が許可されてPDIVが'64'分周に設定されなければなりません。

```
CLKCTRL.MCLKCTRLB = CLKCTRL_PDIV_64X_gc | CLKCTRL_PEN_bm;
```

図3-3. MCLKCTRLAレジスタ構成設定

ビット	7	6	5	4	3	2	1	0
	CLKOUT			CLKSEL1,0				
アクセス種別	R/W	R	R	R	R	R	R/W	R/W
リセット値	0	0	0	0	0	0	0	0

● ビット1,0 – CLKSEL1,0 : クロック選択 (Clock Select)

このビット領域は主クロック(CLK_MAIN)用の供給元を選びます。

値	0 0	0 1	1 0	1 1
名称	OSC20M	OSCUPL32K	XOSC32K	EXTCLK
説明	20MHz内部発振器	32kHz内部超低電力発振器	32.768kHz外部クリスタル用発振器	外部クロック

CLKSELビット領域が'01'に設定されなければならないことを意味するOSCULP32Kが選ばれなければなりません。これは以下のコードに変換できます。

```
CLKCTRL.MCLKCTRLA = CLKCTRL_CLKSEL_OSCULP32K_gc;
```

図3-4. MCLKSTATUSレジスタ構成設定

ビット	7	6	5	4	3	2	1	0
	EXTS	XOSC32KS	OSC32KS	OSC20MS				SOSC
アクセス種別	R	R	R	R	R	R	R	R
リセット値	0	0	0	0	0	0	0	0

- ビット0 - SOSC : 主クロック発振器変更 (Main Clock Oscillator Changing)

値	0	1
説明	CLK_MAIN用クロック元は切り替えを体験していません。	CLK_MAIN用クロック元は切り替えを体験し、新供給元が安定すると直ぐに変更します。

クロック切り替え処理はSOSCビットによって示されます。プログラムはクロック元の切り替え体験の間、停止されるべきで、故に切り替えが終わるまで待つように実装されるべきです。

```
while (CLKCTRL.MCLKSTATUS & CLKCTRL_SOSC_bm)
{
    ;
}
```

TCB入力クロックと動作形態の構成設定

PB5に8ビットPWM信号を生成するため、以下のレジスタが変更されなければなりません。

- TCBn.CCMP
- TCBn.CTRLA
- TCBn.CTRLB

TCBn.CCMPHとTCBn.CCMPLのレジスタ対は16ビット値のTCBn.CCMPを表します。下位バイト[7~0](接尾辞L)は変位原点でアクセスできます。上位バイト[15~8](接尾辞H)は変位+1でアクセスすることができます。8ビットPWM動作ではTCBn.CCMPHとTCBn.CCMPLは2つの独立した比較レジスタとして働きます。

図3-5. TCBn.CCMPLレジスタ構成設定

ビット	15	14	13	12	11	10	9	8
	CCMP15~8							
アクセス種別	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
リセット値	0	0	0	0	0	0	0	0
ビット	7	6	5	4	3	2	1	0
	CCMP7~0							
アクセス種別	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
リセット値	0	0	0	0	0	0	0	0

- ビット15~8 - CCMP15~8 : 比較/捕獲値上位バイト (Compare/Capture Value high byte)

これらのビットは16ビットの比較/捕獲/TOP値の上位バイトを保持します。

- ビット7~0 - CCMP7~0 : 比較/捕獲値下位バイト (Compare/Capture Value low byte)

これらのビットは16ビットの比較/捕獲/TOP値の下位バイトを保持します。

8ビットPWM動作でTCBを動かす時にTCBn.CCMPLはこの場合で1秒のPWM信号周期を設定されなければなりません。出力パルスの周期がTCBn.CCMPL+1によって定義されるため、TCBn.CCMPLレジスタに設定されなければならない値は0xFF(10進数の255)です。続いて、TCBn.CCMPHは出力がHighに駆動されるクロック周期数を設定されなければなりません。目的は500ms毎にPB5を交互切り替えさせるためにデューティサイクルを50%に設定することです。

```
CCMPH = (CCMPL + 1) × (50/100) = (255+1) × (50/100) = 128    これはTCBn.CCMPH=128=0x80に帰着します。
```

これはTCBn.CCMPがTCBn.CCMPHとTCBn.CCMPLから得られる以下の値を設定されなければならないことを意味します。

```
TCB3.CCMP = 0x80FF;
```

図3-6. TCBn.CTRLAレジスタ構成設定

ビット	7	6	5	4	3	2	1	0
	RUNSTDBY		SYNCPD		CLKSEL1,0			ENABLE
アクセス種別	R	R/W	R	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2,1 – CLKSEL1,0 : クロック選択 (Clock Select)

これらビットの書き込みはこの周辺機能用のクロック元を選びます。

値	0 0	0 1	1 0	1 1
説明	CLK_PER	CLK_PER/2	TCA0からのCLK_TCA	(予約)

● ビット0 – ENABLE : 許可 (Enable)

このビットに'1'を書くことはタイマ/カウンタB型周辺機能を許可します。

TCB3はTCB3.CTRLAレジスタでENABLEビットを'1'に設定することによって許可することができます。これは以下のコードに変換できます。

```
TCB3.CTRLA |= TCB_ENABLE_bm;
```

可能な最低周波数を得るため、TCB3.CTRLAレジスタでCLKSELビット領域を構成設定することによってCLK_PERが更に2分周されます。この場合にCLKSELに対応する値は'01'です。これは以下のコードに変換できます。

```
TCB3.CTRLA |= TCB_CLKSEL_CLKDIV2_gc;
```

図3-7. TCBn.CTRLBレジスタ構成設定

ビット	7	6	5	4	3	2	1	0
		ASYNC	CCMPINIT	CCMPEN		CNTMODE2~0		
アクセス種別	R	R/W	R/W	R/W	R	R/W	R/W	R/W
リセット値	0	0	0	0	0	0	0	0

● ビット4 – CCMPEN : 比較/捕獲出力許可 (Compare/Capture Output Enable)

このビットは比較/捕獲の出力値を許可するのに使われます。

値	0	1
説明	比較/捕獲出力は0です。	比較/捕獲出力は有効な値を持ちます。

● ビット2~0 – CNTMODE2~0 : 計時器動作 (Timer Mode)

これらビットの書き込みは計時器動作を選びます。

値	0 0 0	0 0 1	0 1 0	0 1 1
説明	周期的割り込み動作	制限時間検査動作	事象での計数捕獲動作	計数捕獲周波数測定動作
値	1 0 0	1 0 1	1 1 0	1 1 1
説明	計数捕獲パルス幅測定動作	計数捕獲周波数/パルス幅測定動作	単発動作	8ビットPWM動作

CCMPENは許可されなければなりません。これは以下のコードに変換できます。

```
TCB3.CTRLB |= TCB_CCMPEN_bm;
```

TCBは8ビットPWM動作に構成設定されなければなりません。これは以下のコードに変換できます。

```
TCB3.CTRLB |= TCB_CNTMODE_PWM8_gc;
```



GitHubで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。



助言: 完全なコード例は「[追補](#)」章でも利用可能です。

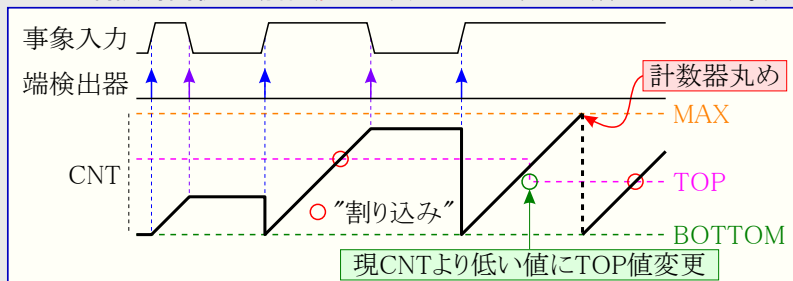
4. 制限時間検査動作でTCBを使用

使用事例説明: TCBを制限時間検査動作に構成設定して(入力として構成設定した)GPIOによって生成される信号時間を測定します。制限時間は2秒に設定され、GPIOによって生成される信号の上昇端と下降端(例えば、実際の応用に於ける押し釘)を検出するのに事象システムが使われます。制限時間超過の場合、別のGPIO(例えば、実際の応用ではLED)が割り込みで交互切り替えされます。

結果: TCBに対する入力信号として扱う事象を生成するのにGPIOを入力として使います。制限時間超過時に割り込みを生成します。

制限時間検査動作では右図で表示されるように、TCBは事象システムとの相互作用を頼ります。この動作では計数器がMAXまで計数して最初に戻ります。初回端で、計数器は再始動され、第2端で停止されます。第2端に先立って計数(TCBn.CNT)レジスタがTOPに達する場合は割り込みが生成されます。凍結状態で、計数器は新規端で再始動します。凍結状態での計数(TCBn.CNT)または捕獲/比較(TCBn.CCMP)のレジスタ読み込み、またはRUNビット(TCBn.STATUSのRUN)書き込みは無効です。

図4-1. 制限時間検査動作 (訳注:本例での事象入力論理は逆です。)



システム クロック構成設定

16ビットTCB計数器に対して2秒周期を得るため、入力周波数は可能な限り低くなければなりません。これ用に、内部32kHz発振器を使うことができます。周波数分周器を使う必要はなく、故にCLK_PER用前置分周器は禁止されます。構成設定処理には以下の3つの段階があります。

1. CLK_PER前置分周器禁止 - 以下のコード断片はCLK_PER前置分周器をどう禁止するかを実演します。

```
CPU_CCP = CCP_IOREG_gc;
CLKCTRL.MCLKCTRLB = 0 << CLKCTRL_PEN_bp;
```

2. 内部32kHz発振器選択 - 以下のコード断片はシステム クロック元を内部32kHz発振器に切り替えます。

```
CPU_CCP = CCP_IOREG_gc;
CLKCTRL.MCLKCTRLA = CLKCTRL_CLKSEL_OSCULP32K_gc;
```

3. クロック切り替え処理完了待機 - 以下のコード断片はクロック元切り替え処理終了をどう待つかを実演します。

```
while (CLKCTRL.MCLKSTATUS & CLKCTRL_SOSC_bm)
{
    ;
}
```

注: 1. MCLKCTRLAとMCLKCTRLBのレジスタはクロック制御レジスタをアクセスするのに先立って、CPU.CCPレジスタに書かれる'IOREG'鍵を必要とします。

2. クロック制御レジスタはこの資料の「8ビットPWM動作でTCBを使用」章で記述されます。

TCBを制限時間検査動作に構成設定

GPIOピンを使って信号を生成するため、それによって事象システムが設定されなければなりません。ポートBの2番(PB2)ピンがこの資料での例として使われます。

以下のレジスタが構成設定されなければなりません。

- TCB0.CCMP
- TCB0.INTCTRL
- TCB0.EVCTRL
- EVSYS.CHANNEL
- EVSYS.USER

1. TCB0.CCMP構成設定

TCB比較/捕獲レジスタは制限時間の値を設定されなければなりません。この資料の目的のため、制限時間用に1秒の値が選ばれました。32.768kHzの入力周波数で16ビット計数器は2秒での完全な周期にします(16ビット計数器の最大値は65535です)。

TCB0.CCMPの再設定値は下で示されるように計算することができます。

$$CCMP = CNT_{max}/2 = 65535/2 = 32767.5$$

32767.5の値は16進数に変換することができず、従って代わりに32767が使われます。

$$CCMP = 32767.5 \approx 32767 = 0x7FFF$$

以下のコード断片は0x7FFF制限時間値をTCB0.CCMPに設定します。

```
TCB0.CCMP = 0x7FFF;
```

2. TCB0.INTCTRL構成設定

図4-2. TCBn.INTCTRLレジスタ構成設定

ビット	7	6	5	4	3	2	1	0
								CAPT
アクセス種別	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 - CAPT : 捕獲割り込み許可 (Capture Interrupt Enable)

このビットに'1'を書くことは捕獲割り込みを許可します。

TCB0.CNTは事象システムのバスで信号端が検出されると直ぐに増加を始めます。制限時間内に信号の補完(逆)端が検出されない場合、TCB0は割り込みを起動します。この点について、TCB0.INTCTRLレジスタで捕獲または制限時間の割り込みが許可されなければなりません。以下のコード断片は割り込みを許可します。

```
TCB0.INTCTRL = TCB_CAPT_bm;
```

注: 全体割り込みも許可されなければなりません。これはソフトウェアプログラムに於いて後の段階で行われます。

3. TCB0.EVCTRL構成設定

図4-3. TCBn.EVCTRLレジスタ構成設定

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

● ビット4 - EDGE : 事象端選択 (Event Edge)

このビットは事象端を選ぶのに使われます。このビットの影響は制御B(TCBn.CTRLB)レジスタで選んだ計数動作(CNTMODE)に依存します。

計数動作	EDGE	正(上昇)端	負(下降)端
周期的割り込み動作	0	-	-
	1	-	-
制限時間検査動作	0	計数開始	計数停止
	1	計数停止	計数開始
事象での計数捕獲動作	0	計数値を捕獲、割り込み	-
	1	-	計数値を捕獲、割り込み
計数捕獲周波数測定動作	0	計数値を捕獲/解消/再開、割り込み	-
	1	-	計数値を捕獲/解消/再開、割り込み
計数捕獲パルス幅測定動作	0	計数値を解消/再開	計数値を捕獲、割り込み
	1	計数値を捕獲、割り込み	計数値を解消/再開
計数捕獲周波数/パルス幅測定動作	0	第1正端で計数値を解消/再開、後続する負端で捕獲、第2正端で停止と割り込み	-
	1	第1負端で計数値を解消/再開、後続する正端で捕獲、第2負端で停止と割り込み	-
単発動作	0	計数開始	-
	1	計数開始	計数開始
8ビットPWM動作	0	-	-
	1	-	-

● ビット0 - CAPTEI : 捕獲事象入力許可 (Capture Event Input Enable)

このビットに'1'を書くことはTCBに対する事象入力捕獲を許可します。

目標はPB2入力ピンの下降端と上昇端間の信号時間を測定することです(アイドルでピンはHigh状態を保ちます)。これはCAPTEIとEDGEのビットが'1'に設定されることを意味します。以下のコード断片はTCB0.EVCTRLレジスタで両ビットを許可します。

```
TCB0.EVCTRL = TCB_CAPTEI_bm | TCB_EDGE_bm;
```

4. EVSYS.CHANNEL0構成設定

各チャンネルは1つの事象生成部に接続することができます。全ての生成部が全てのチャンネルに接続することができる訳ではありません。どの生成部供給元が各チャンネルに配線することができるか、この配線を達成するためにEVSYS.CHANNEL_nに書かれなければならない生成部の値を見るには下表を参照してください。EVSYS.CHANNEL_nの0x00の値はチャンネルをOFFにします。

図4-4. EVSYS.CHANNEL_nレジスタ構成設定

ビット	7	6	5	4	3	2	1	0
	GENERATOR7~0							
アクセス種別	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
リセット値	0	0	0	0	0	0	0	0

生成部 [2進(16進)]	入力	同期/非同期	CH0	CH1	CH2	CH3	CH4	CH5	CH6	CH7
0000_0001 (\$01)	UPDI	同期	UDPI							
0000_0110 (\$06)	RTC_OVF	非同期	OVF							
0000_0111 (\$07)	RTC_CMP	非同期	CMP							
0000_1000 (\$08)	RTC_PIT0	非同期	DIV64	DIV1024	DIV64	DIV1024	DIV64	DIV1024	DIV64	DIV1024
0000_1001 (\$09)	RTC_PIT1	非同期	DIV128	DIV2048	DIV128	DIV2048	DIV128	DIV2048	DIV128	DIV2048
0000_1010 (\$0A)	RTC_PIT2	非同期	DIV256	DIV4096	DIV256	DIV4096	DIV256	DIV4096	DIV256	DIV4096
0000_1011 (\$0B)	RTC_PIT3	非同期	DIV512	DIV8192	DIV512	DIV8192	DIV512	DIV8192	DIV512	DIV8192
0001_00xx (\$10~\$13)	CCL_LUT _n	非同期	LUT _n							
0010_0000 (\$20)	AC0	非同期	OUT							
0010_0100 (\$24)	ADC0	同期	COMP							
0100_0xxx (\$40~\$47)	PORT0_PIN _n	非同期	PORTA_PIN _n	PORTC_PIN _n	PORTE_PIN _n	-				
0100_1xxx (\$48~\$4F)	PORT1_PIN _n	非同期	PORTB_PIN _n	PORTD_PIN _n	PORTF_PIN _n	-				
0110_0xxx (\$60~\$67)	USART _n	同期	XCK							
0110_1000 (\$68)	SPI0	同期	SCK							
1000_0000 (\$80)	TCA0_OVF	同期	OVF							
1000_0001 (\$81)	TCA0_ERR	同期	ERR							
1000_0100 (\$84)	TCA0_CMP0	同期	CMP0							
1000_0101 (\$85)	TCA0_CMP1	同期	CMP1							
1000_0110 (\$86)	TCA0_CMP2	同期	CMP2							
1010_xxx0 (\$A0~\$AE)	TCB _n	同期	CMP3							

EVSYS.CHANNEL0は以下のコード断片で表されるように、入力事象を生成するためにPB2に対して0x4Aの値を設定されなければなりません。

```
EVSYS.CHANNEL0 = EVSYS_GENERATOR_PORT1_PIN2_gc;
```

5. EVSYS.USER20構成設定

各事象使用部は1つのチャンネルに接続することができます。いくつかの使用部は同じチャンネルに接続することができます。以下の表はそれらの対応する使用部ID番号と共に全ての事象システム使用部を一覧にします。このID番号はUSER_nレジスタ指標に対応し、例えば、ID番号2を持つ使用部はEVSYS.USER2レジスタによって制御されます。

図4-5. EVSYS.USERnレジスタ構成設定

使用部番号	使用部名称	同期/非同期	説明	使用部番号	使用部名称	同期/非同期	説明
0	CCL-LUT0A	非同期	LUT0事象A入力	12	EVOUTD	非同期	事象出力Dピン
1	CCL-LUT0B	非同期	LUT0事象B入力	13	EVOUTE	非同期	事象出力Eピン
2	CCL-LUT1A	非同期	LUT1事象A入力	14	EVOUTF	非同期	事象出力Fピン
3	CCL-LUT1B	非同期	LUT1事象B入力	15	USART0	同期	USART0事象入力
4	CCL-LUT2A	非同期	LUT2事象A入力	16	USART1	同期	USART1事象入力
5	CCL-LUT2B	非同期	LUT2事象B入力	17	USART2	同期	USART2事象入力
6	CCL-LUT3A	非同期	LUT3事象A入力	18	USART3	同期	USART3事象入力
7	CCL-LUT3B	非同期	LUT3事象B入力	19	TCA0	同期	TCA0事象入力
8	ADC0	非同期	ADC起動	20	TCB0	非同期	TCB0事象入力
9	EVOUTA	非同期	事象出力Aピン	21	TCB1	非同期	TCB1事象入力
10	EVOUTB	非同期	事象出力Bピン	22	TCB2	非同期	TCB2事象入力
11	EVOUTC	非同期	事象出力Cピン	23	TCB3	非同期	TCB3事象入力

ビット	7	6	5	4	3	2	1	0
	CHANNEL3~0							
アクセス種別	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7~0 - CHANNEL7~0 : 使用部チャネル選択 (User Channel Selection)

使用部がどの事象システムチャネルに接続するかを示します。

値	0	1~8 (= n)	その他
説明	OFF、チャネルはこの事象システム使用部に未接続	事象使用部はチャネルn-1に接続されます。	(予約)

使用部チャネルは(EVSYS.CHANNEL0で)前もって既に構成設定された事象チャネルに繋がられます。以下のコード断片はこれがどう行われ得るかを示します。

```
EVSYS_USERTCB0 = EVSYS_CHANNEL_CHANNEL0_gc;
```



GitHubで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。



助言: 完全なコード例は「[追補](#)」章でも利用可能です。

5. 休止動作でTCBを使用

使用事例説明: 例え、マイクロコントローラがスタンバイ休止動作の時でも、毎秒の溢れ割り込みを生成してGPIO(PB5)ピンを交互切り替えるようにTCBを構成設定します。

結果: TCB溢れ割り込みはマイクロコントローラの走行動作(アイドル/スタンバイ)に関わらず、1秒毎に起動されます。割り込み処理ルーチン(ISR: Interrupt Service Routine)ではPB5に対応する基板上的のLEDが交互切り替えされます。

TCBnは既定でスタンバイ休止動作で禁止され、マイクロコントローラが休止動作へ移行するとすると直ぐにTCBは停止されます。この単位部はTCBn.CTRLAレジスタのスタンバイで走行(RUNSTDBY)ビットが'1'を書かれた場合にスタンバイ休止動作で完全な動作に留まることができます。

休止動作システムクロック構成設定

16ビットTCB計時器に対して1秒の割り込み得るため、入力周波数は可能な限り低くなければなりません。これ用に、内部32kHz発振器を使うことができます。周波数分周器を使う必要はなく、故にCLK_PER用前置分周器は禁止されなければなりません。構成設定処理には以下の3つの段階があります。

1. CLK_PER前置分周器禁止 - 以下のコード断片はCLK_PER前置分周器をどう禁止するかを実演します。

```
CPU_CCP = CCP_IOREG_gc;
CLKCTRL.MCLKCTRLB = 0 << CLKCTRL_PEN_bp;
```

2. 内部32kHz発振器選択 - 以下のコード断片はシステム クロック元を内部32kHz発振器に切り替えます。

```
CPU_CCP = CCP_IOREG_gc;
CLKCTRL.MCLKCTRLA = CLKCTRL_CLKSEL_OSCULP32K_gc;
```

3. クロック切り替え処理完了待機 - 以下のコード断片はクロック元切り替え処理終了をどう待つかを実演します。

```
while (CLKCTRL.MCLKSTATUS & CLKCTRL_SOSC_bm)
{
    ;
}
```

休止動作形態構成設定

休止動作は節電のためにデバイス内の周辺機能とクロック領域を停止するのに使われます。休止制御器(SLPCTRL)は活動と休止の動作間遷移を制御して処理します。ソフトウェアが実行される1つの活性動作と3つの休止動作で4つの動作形態が利用可能です。利用可能な休止動作はアイドル、スタンバイ、パワーダウンです。

割り込みは休止からデバイスを起こすのに使われます。利用可能な割り込み起こし元は構成設定した休止動作に依存します。割り込み発生時、デバイスは起き上がり、**SLEEP**命令に後続する最初の命令から通常のプログラム実行を続けるのに先立って割り込み処理ルーチンを実行します。

図5-1. SLPCTRL.CTRLAレジスタ構成設定

ビット	7	6	5	4	3	2	1	0
						SMODE1,0		SEN
アクセス種別	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2,1 - SMODE1,0 : 休止動作形態 (Sleep Mode)

これらのビット書き込みは休止許可(SEN)ビットが'1'を書かれ、**SLEEP**命令が実行される時に移行される休止動作形態を選びます。

値	0 0	0 1	1 0	1 1
名称	IDLE	STANDBY	PDOWN	-
説明	アイドル休止動作許可	スタンバイ休止動作許可	パワーダウン休止動作許可	(予約)

● ビット0 - SEN : 休止許可 (Sleep Enable)

選択された休止動作にMCUを移行するために**SLEEP**命令が実行される前に、このビットは'1'を書かれなければなりません。

TCB単位部はマイクロ コントローラがスタンバイ休止動作の間に走行することができます。パワーダウン休止動作はTCBを完全に禁止します。

この使用事例の目的に対して、ATmega4809マイクロ コントローラはスタンバイ休止動作で休止へ移行するように構成設定されます。休止動作移行も許可されなければなりません。以下のコード断片は休止操作を許可して**SLEEP**命令が実行された時にスタンバイ休止動作へ移行するようにATmega4809マイクロ コントローラを構成設定します。

```
SLPCTRL.CTRLA = SLPCTRL_SMODE_gm | SLPCTRL_SMODE_STDBY_gc;
```

周期的割り込み動作でTCB構成設定

以下の構成設定は次のレジスタに対して行われなければなりません。

- TCBn.CCMP
- TCBn.CTRLA

TCB捕獲/比較レジスタは割り込みを起動する比較値を設定されなければなりません。この資料の目的のため、周期的割り込み用に1秒の値が選ばれました。32.768kHzの入力周波数で、16ビット計数器は2秒での完全周期にさせます(16ビット数の最大値は65535です)。16ビットTCB計数器の最大値が2秒最大周期に対して65535で達成することができるため、**CCMP**レジスタは最大値-32768の半分(0x7FFF)を設定されなければなりません。以下のコード断片はTCB0.CCMPレジスタに0x7FFFを設定します。

```
TCB0.CCMP = 0x7fff;
```

図5-2. TCBn.CTRLAレジスタ構成設定

ビット	7	6	5	4	3	2	1	0
		RUNSTDBY		SYNCUPD		CLKSEL1,0		ENABLE
アクセス種別	R	R/W	R	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6 – RUNSTDBY** : スタンバイ時走行 (Run Standby)

このビットへの'1'書き込みはスタンバイ休止動作での走行をこの周辺機能に許します。クロック選択(CLKSEL)が'10'(CLK_TCA)に設定される時は適用できません。

前置分周は必要とされず、故に分周値は'0'です。また、計時器は許可されなければなりません。TCBに対してスタンバイ休止動作の間に未だ割り込みを生成するため、RUNSTDBYビットも許可されなければなりません。以下のコード断片はクロック分周器を'0'(前置分周なし)に設定し、計時器を許可してスタンバイ休止動作での走行を計時器に許します。

```
TCB0.CTRLA = TCB_CLKSEL_CLKDIV1_gc | TCB_ENABLE_bm | TCB_RUNSTDBY_bm;
```

周期的割り込みを起動するため、TCB0.INTCTRLレジスタのCAPTビットが許可されなければなりません。以下のコード断片は割り込みを許可します。

```
TCB0.INTCTRL = TCB_CAPT_bm;
```

注: 1. 全体割り込みも許可されなければなりません。これはソフトウェアプログラムに於いて後の段階で行うことができます。

2. パワーダウン休止動作では全ての動作が停止されます。

割り込み発生を可視化するために出力としてピンを構成設定

周期的割り込み発生を可視化するため、PB5ピンは出力動作で構成設定されます。以下のコードはPB5をLow出力として設定します。

```
PORTB_DIR |= PIN5_bm;
PORTB_OUT |= PIN5_bm;
```



GitHubで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。



助言: 完全なコード例は「[追補](#)」章でも利用可能です。

6. 参照

1. ATmega4809製品頁 : <https://www.microchip.com/wwwproducts/en/ATMEGA4809>
2. 'megaAVR® 0系手引書' (DS40002015)
3. 'megaAVR® 0系ATmega3209/4809 - 48ピン データシート' (DS40002016)
4. ATmega4809 Xplained Proウェブ頁 : <https://www.microchip.com/developmenttools/ProductDetails/atmega4809-xpro>

7. 追補

例7-1. TCB 8ビットPWM動作コード例

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

#define TCB_CMP_EXAMPLE_VALUE    (0x80FF)

void CLOCK_init (void);
void PORT_init (void);
void TCB3_init (void);

void CLOCK_init (void)
{
    /* 保護されたレジスタへの書き込みを許可 */
    CPU_CCP = CCP_IOREG_gc;
    /* 前置分周器許可、前置分周器を64分周に設定 */
    CLKCTRL.MCLKCTRLB = CLKCTRL_PDIV_64X_gc | CLKCTRL_PEN_bm;
```

例7-1 (続き). TCB 8ビットPWM動作コード例

```

/* 保護されたレジスタへの書き込みを許可 */
CPU_CCP = CCP_IOREG_gc;
/* 32kHz内部超低電力発振器(OSCULP32K)選択 */
CLKCTRL.MCLKCTRLA = CLKCTRL_CLKSEL_OSCULP32K_gc;

/* システム発振器変更終了待ち */
while (CLKCTRL.MCLKSTATUS & CLKCTRL_SOSC_bm)
{
    ;
}

void PORT_init (void)
{
    PORTB_DIR |= PIN5_bm;
    PORTB_OUT |= PIN5_bm;
}

void TCB3_init (void)
{
    /* PWMの周期とデューティサイクルをCCMPレジスタに設定 */
    TCB3.CCMP = TCB_CMP_EXAMPLE_VALUE;

    /* TCB3許可、CLK_PERを2分周 */
    TCB3.CTRLA |= TCB_ENABLE_bm;
    TCB3.CTRLA |= TCB_CLKSEL_CLKDIV2_gc;

    /* ピン出力許可、TCBを8ビットPWM動作に構成設定 */
    TCB3.CTRLB |= TCB_CCMPEN_bm;
    TCB3.CTRLB |= TCB_CNTMODE_PWM8_gc;
}

int main(void)
{
    CLOCK_init();
    PORT_init();
    TCB3_init();

    while (1)
    {
        ;
    }
}

```

例7-2. TCB制限時間検査動作コード例

```

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

void CLOCK_init (void);
void PORT_init (void);
void EVENT_SYSTEM_init (void);
void TCB0_init (void);

void CLOCK_init (void)
{
    /* 保護されたレジスタへの書き込みを許可 */
    CPU_CCP = CCP_IOREG_gc;
    /* CLK_PER前置分周器禁止 */
    CLKCTRL.MCLKCTRLB = 0 << CLKCTRL_PEN_bp;
}

```

例7-2 (続き). TCB制限時間検査動作コード例

```

/* 保護されたレジスタへの書き込みを許可 */
CPU_CCP = CCP_IOREG_gc;
/* 32kHz内部超低電力発振器(OSCULP32K)選択 */
CLKCTRL.MCLKCTRLA = CLKCTRL_CLKSEL_OSCULP32K_gc;

/* システム発振器変更終了待ち */
while (CLKCTRL.MCLKSTATUS & CLKCTRL_SOSC_bm)
{
    ;
}

void PORT_init (void)
{
    PORTB.DIR |= PIN5_bm;          /* PB5をデジタル出力として構成設定 */
    PORTB.OUT |= PIN5_bm;        /* PB5の初期値を設定 */
    PORTB.DIR &= ~PIN2_bm;       /* PB2をデジタル入力として構成設定 */
    PORTB.PIN2CTRL = PORT_PULLUPEN_bm; /* 内部プルアップ許可 */
}

void EVENT_SYSTEM_init (void)
{
    EVSYS.CHANNEL0 = EVSYS_GENERATOR_PORT1_PIN2_gc; /* 入力事象としてポート1の2番ピンを設定 */
    EVSYS.USERTCB0 = EVSYS_CHANNEL_CHANNEL0_gc;     /* 使用部を事象チャンネル0に接続 */
}

void TCB0_init (void)
{
    /* 制限時間値を比較/捕獲レジスタに設定 */
    TCB0.CCMP = 0x7fff;

    /* TCB0許可、CLK_PERを1分周(前置分周なし)に設定 */
    TCB0.CTRLA = TCB_CLKSEL_CLKDIV1_gc | TCB_ENABLE_bm;

    /* TCBを周期的制限時間動作に構成設定 */
    TCB0.CTRLB = TCB_CNTMODE_TIMEOUT_gc;

    /* 捕獲/制限時間超過割り込みを許可 */
    TCB0.INTCTRL = TCB_CAPT_bm;

    /* 事象入力と事象端を許可 */
    TCB0.EVCTRL = TCB_CAPTEI_bm | TCB_EDGE_bm;
}

ISR(TCB0_INT_vect)
{
    TCB0.INTFLAGS = TCB_CAPT_bm;          /* 割り込み要求フラグ解除(0) */
    PORTB.IN = PIN5_bm;                  /* PB5 GPIO交互切り替え */
}

int main(void)
{
    CLOCK_init();
    PORT_init();
    EVENT_SYSTEM_init();
    TCB0_init();

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

```

例7-2 (続き). TCB制限時間検査動作コード例

```

while (1)
{
    ;
}

```

例7-3. TCB休止動作操作コード例

```

/**
 * ¥file main.c
 *
 * ¥brief Main source file.
 *
 (c) 2018 Microchip Technology Inc.とその子会社

```

これらの条件の遵守を条件として、Microchipソフトウェアと何れの派生物をMicrochip製品に限って使うことができます。Microchipソフトウェアを伴うかもしれない(開放ソースソフトウェアを含む)第三者ソフトウェアの使用に適用される第三者の許諾条件を遵守するのはあなたの責任です。

このソフトウェアはMicrochipの“現状そのまま”によって供給されます。非侵害、商品性、特定目的への適合性のどの暗黙の保証を含め、明示的、暗示的または法令であろうがこのソフトウェアには適用されません。例えMicrochipが可能性を進言されたり、損害が予見できるとしても、原因の如何に関わらず、如何なる場合でもMicrochipはソフトウェアに関連する如何なる種類の間接的、特別、懲罰的、偶発的または必然的な損失、損害、費用または支払いの責任を負いません。法によって許される最大限の範囲まで、このソフトウェアに関連するどの方法での全請求に於いても、microchipの総責任額は、もしあれば、あなたがこのソフトウェアに対してMicrochipへ直接支払った費用の額を超えません。

```

*
*/

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

#define TCB_CMP_EXAMPLE_VALUE    (0x7fff)

void CLOCK_init (void);
void SLPCTRL_init (void);
void PORT_init (void);
void TCBO_init (void);

void CLOCK_init (void)
{
    /* 保護されたレジスタへの書き込みを許可 */
    CPU_CCP = CCP_IOREG_gc;
    /* CLK_PER前置分周器禁止 */
    CLKCTRL.MCLKCTRLB = 0 << CLKCTRL_PEN_bp;

    /* 保護されたレジスタへの書き込みを許可 */
    CPU_CCP = CCP_IOREG_gc;
    /* 32kHz内部超低電力発振器(OSCULP32K)選択 */
    CLKCTRL.MCLKCTRLA = CLKCTRL_CLKSEL_OSCULP32K_gc;

    /* システム発振器変更終了待ち */
    while (CLKCTRL.MCLKSTATUS & CLKCTRL_SOSC_bm)
    {
        ;
    }
}

void SLPCTRL_init (void)
{

```


例7-3 (続き). TCB休止動作操作コード例

```
/* 休止動作許可、スタンバイ動作選択 */
SLPCTRL.CTRLA = SLPCTRL_SMODE_gm | SLPCTRL_SMODE_STDBY_gc;
}

void PORT_init (void)
{
    PORTB.DIR |= PIN5_bm;          /* PB5をデジタル出力として構成設定 */
    PORTB.OUT |= PIN5_bm;        /* PB5の初期値を設定 */
}

void TCB0_init (void)
{
    /* 制限時間値を比較/捕獲レジスタに設定 */
    TCB0.CCMP = TCB_CMP_EXAMPLE_VALUE;

    /* TCB0許可、CLK_PERを1分周(前置分周なし)に設定 */
    TCB0.CTRLA = TCB_CLKSEL_CLKDIV1_gc | TCB_ENABLE_bm | TCB_RUNSTDBY_bm;

    /* 捕獲/制限時間超過割り込みを許可 */
    TCB0.INTCTRL = TCB_CAPT_bm;
}

ISR(TCB0_INT_vect)
{
    TCB0.INTFLAGS = TCB_CAPT_bm;  /* 割り込み要求フラグ解除(0) */
    PORTB.IN = PIN5_bm;          /* PB5 GPIO交互切り替え */
}

int main(void) {
    CLOCK_init();
    SLPCTRL_init();
    PORT_init();
    TCB0_init();

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

    while (1)
    {
        sleep_mode();
    }
}
```

Microchipウェブ サイト

Microchipは<http://www.microchip.com/>で当社のウェブ サイト経由でのオンライン支援を提供します。このウェブ サイトはお客様がファイルや情報を容易に利用可能にする手段として使われます。お気に入りのインターネット ブラウザを用いてアクセスすることができ、ウェブ サイトは以下の情報を含みます。

- **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相談役プログラム員一覧
- **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

お客様への変更通知サービス

Microchipのお客様通知サービスは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<http://www.microchip.com/>でMicrochipのウェブ サイトをアクセスしてください。”Support”下で”Customer Change Notification”をクリックし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
- 現場応用技術者(FAE:Field Application 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、または現場応用技術者(FAE)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<http://www.microchip.com/support>でのウェブ サイトを通して利用できます。

Microchipデバイスコード保護機能

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

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

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

法的通知

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

商標

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

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

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

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

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

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

ここで言及した以外の全ての商標はそれら各々の会社の所有物です。

© 2019年、Microchip Technology Incorporated、米国印刷、不許複製

DNVによって認証された品質管理システム

ISO/TS 16949

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

日本語© HERO 2019.

本技術概説はMicrochipのTB3214技術概説(DS90003214A-2019年1月)の翻訳日本語版です。日本語では不自然となる重複する形容表現は省略されている場合があります。日本語では難解となる表現は大幅に意識されている部分もあります。必要に応じて一部加筆されています。頁割の変更により、原本より頁数が少なくなっています。

必要と思われる部分には()内に英語表記や略称などを残す形で表記しています。

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



MICROCHIP

世界的な販売とサービス

米国	亜細亜/太平洋	亜細亜/太平洋	欧州
本社 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-792-7200 Fax: 480-792-7277 技術支援: http://www.microchip.com/support ウェブアドレス: www.microchip.com	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 - 香港特别行政区 Tel: 852-2943-5100 中国 - 南京 Tel: 86-25-8473-2460 中国 - 青島 Tel: 86-532-8502-7355 中国 - 上海 Tel: 86-21-3326-8000 中国 - 瀋陽 Tel: 86-24-2334-2829 中国 - 深圳 Tel: 86-755-8864-2200 中国 - 蘇州 Tel: 86-186-6233-1526 中国 - 武漢 Tel: 86-27-5980-5300 中国 - 西安 Tel: 86-29-8833-7252 中国 - 廈門 Tel: 86-592-2388138 中国 - 珠海 Tel: 86-756-3210040	インド - ハンガロール Tel: 91-80-3090-4444 インド - ニューデリー Tel: 91-11-4160-8631 インド - フネー Tel: 91-20-4121-0141 日本 - 大阪 Tel: 81-6-6152-7160 日本 - 東京 Tel: 81-3-6880-3770 韓国 - 大邱 Tel: 82-53-744-4301 韓国 - ソウル Tel: 82-2-554-7200 マレーシア - クアラルンプール Tel: 60-3-7651-7906 マレーシア - ペナン Tel: 60-4-227-8870 フィリピン - マニラ Tel: 63-2-634-9065 シンガポール Tel: 65-6334-8870 台湾 - 新竹 Tel: 886-3-577-8366 台湾 - 高雄 Tel: 886-7-213-7830 台湾 - 台北 Tel: 886-2-2508-8600 タイ - バンコク Tel: 66-2-694-1351 ベトナム - ホーチミン Tel: 84-28-5448-2100	オーストラリア - ウェルズ Tel: 43-7242-2244-39 Fax: 43-7242-2244-393 デンマーク - コペンハーゲン Tel: 45-4450-2828 Fax: 45-4485-2829 フィンランド - エスポー Tel: 358-9-4520-820 フランス - パリ Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79 ドイツ - ガルピング Tel: 49-8931-9700 ドイツ - ハーン Tel: 49-2129-3766400 ドイツ - ハイムブロン Tel: 49-7131-67-3636 ドイツ - カールスルーエ Tel: 49-721-625370 ドイツ - ミュンヘン Tel: 49-89-627-144-0 Fax: 49-89-627-144-44 ドイツ - ローゼンハイム Tel: 49-8031-354-560 イスラエル - ラーナナ Tel: 972-9-744-7705 イタリア - ミラノ Tel: 39-0331-742611 Fax: 39-0331-466781 イタリア - ハドバ Tel: 39-049-7625286 オランダ - デルフト 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 - マドリード 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
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			