
高周波数入力でTCDを使うPWM信号生成

序説

著者: Marius Nicolae, Microchip Technology Inc.

AVR® DA MCU系統のマイクロ コントローラは使用者に対して利用可能な4つの周波数チャネルによってクロック駆動することができる多用途型のタイマ/カウンタが装備されます。12ビット タイマ/カウンタD型(TCD)に対する入力周波数は超低電力(ULP:Ultra Low-Power)内部または外部の発振器によって提供される32.768kHzから位相固定化閉路(PLL:Phase-Locked Loop)クロック逡倍システムを使うことによって48MHzまで変わります。従って、TCDは広範囲のパルス幅変調(PWM:Pulse-Width Modulation)信号を生成するのに使うことができます。

この技術概説の範囲はTCD動作形態のいくつかを記述して、TCDの特殊性を強調し、初期化コードの断片を提供します。機能のより深い理解についてはAVR DAデータシートを参照してください。

この技術概説は内部高周波数発振器(OSCHF)と(OSCHFからクロック駆動される)PLLのAVR DA高周波数供給元がTCDに対する入力クロックとして使われる2つの事例を記述します。

• 1傾斜動作で2つのPWM信号生成

この例の目的は入力クロックとしてOSCHFを持つ1傾斜動作で構成設定したTCDを使って2つのPWM信号を生成することです。OSCHFは24MHzで動くように構成設定され、PWM信号は出力としてPA4ピンでのWOAとPA5ピンでのWOBに設定されます。WOAでのPWM信号は25%のデューティサイクルを持ち、WOBでの信号は30%のデューティサイクルを持ちます。

• 2傾斜動作で2つのPWM信号生成

この例の範囲は入力クロックとしてPLLを持つ2傾斜動作で構成設定したTCDを使って2つのPWM信号を生成することです。PLLは48MHzで動くように構成設定され、PWM信号は出力としてPA4ピンでのWOAとPA5ピンでのWOBに設定されます。WOAでのPWM信号は10%のデューティサイクルを持ち、WOBでの信号は20%のデューティサイクルを持ちます。

注: 各例ではPWM信号生成に対してPA4とPA5のピンが使われます。コード例はAVR128DA48 Curiosity Nano開発基板を使って開発されました。

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. AVR® DA系概要	3
2. 概要	3
3. 1傾斜動作で2つのPWM信号生成	4
4. 2傾斜動作で2つのPWM信号生成	10
5. 参考資料	12
6. 追補	12
7. 改訂履歴	15
Microchipウェブサイト	16
製品変更通知サービス	16
お客様支援	16
Microchipデバイスコード保護機能	16
法的通知	16
商標	17
品質管理システム	17
世界的な販売とサービス	18

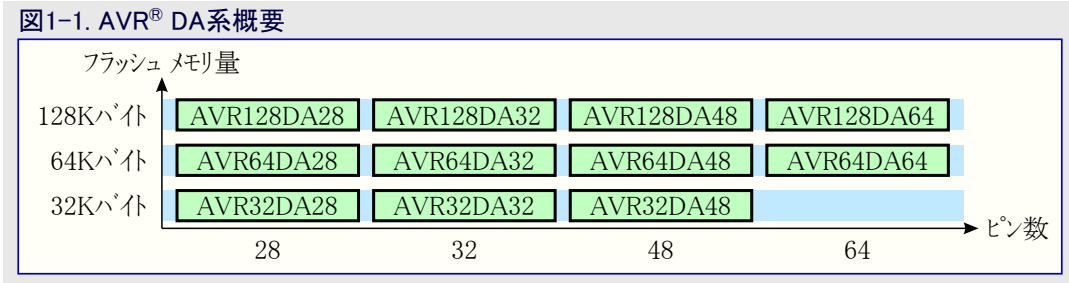
1. 関連デバイス

本章はこの文書に関連するデバイスを一覧にします。

1.1. AVR® DA系概要

下図はピン数の変種とメモリ量を展開してAVR® DAデバイスを示します。

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



異なるフラッシュメモリ量を持つデバイスは一般的にSRAM量も異なります。

2. 概要

12ビット タイマ/カウンタ型(TCD)は非同期計数器、前置分周器、比較、捕獲、制御論理回路から成る高性能な波形制御器です。TCDは周辺機能クロックに対して非同期のクロックで動くことができる計数器を含みます。これは任意選択の沈黙時間を持つ独立した2つの出力を生成することができる比較論理回路を含みます

図2-1. タイマ/カウンタ型構成図

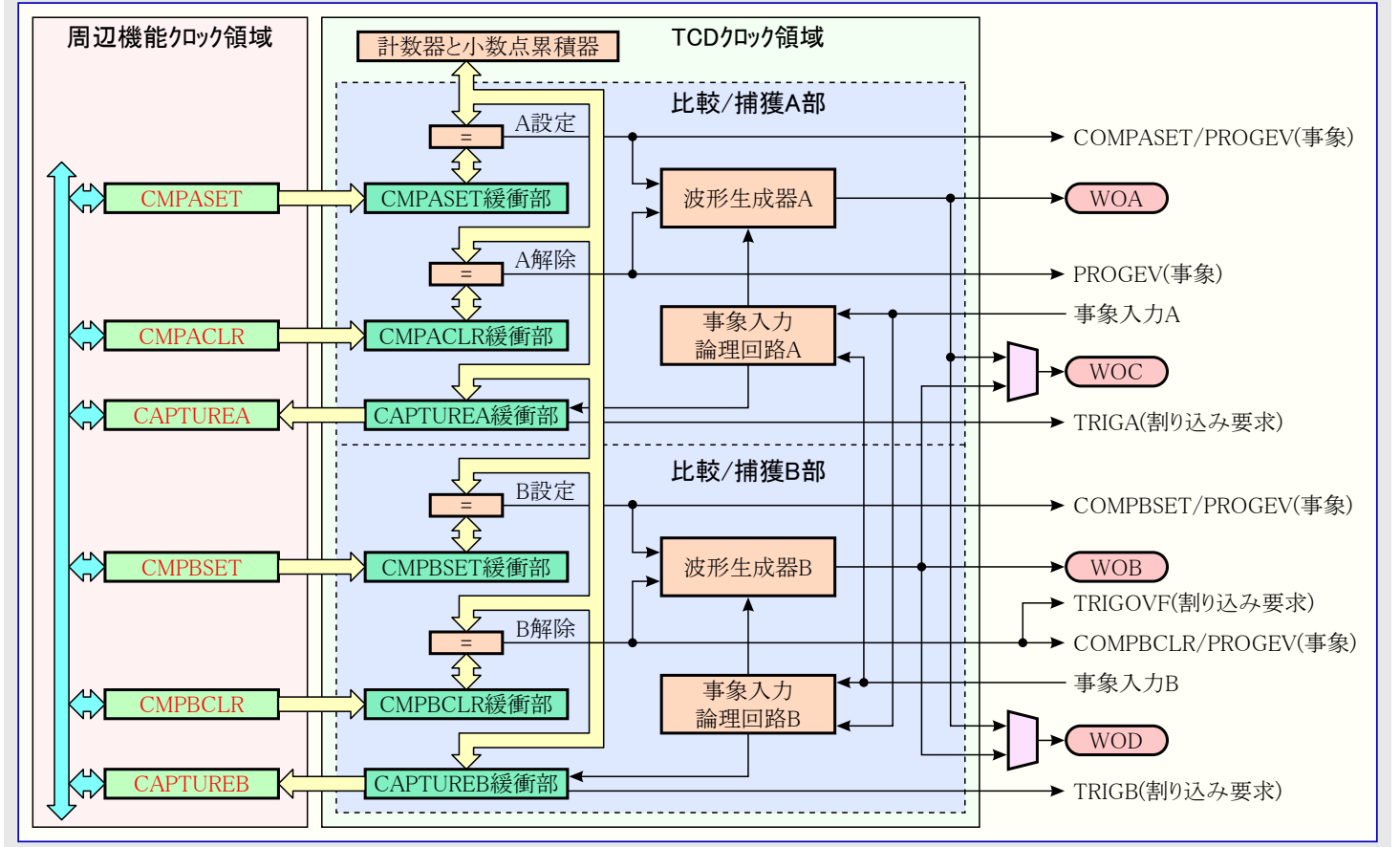
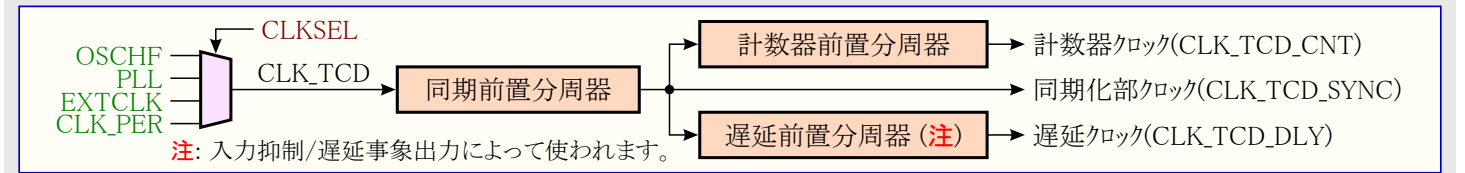


図2-2. クロック選択と前置分周器の概要



3. 1傾斜動作で2つのPWM信号生成

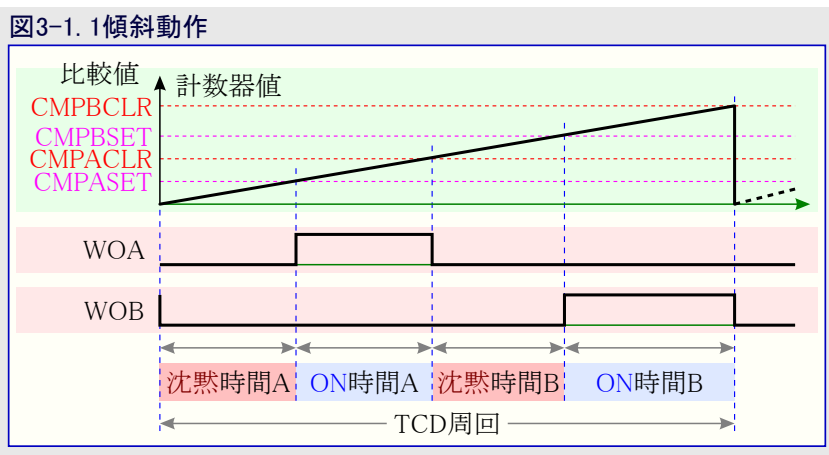
使用事例説明: TCDは24MHzで動く入力クロックとしてOSCHFを持ち、PA4ピンでのWOAとPA5ピンでのWOBで2つのPWM信号を生成する1傾斜動作で動くように初期化されて構成設定されます。WOAのPWM信号は25%のデューティサイクルを持ち、WOBの信号は30%のデューティサイクルを持ちます。

結果: TCDはWOA(PA4ピン)で25%のデューティサイクルを持つ1つのPWM信号とWOBで30%のデューティサイクルを持つ1つのPWM信号を生成します。

TCDはカウンタの値が比較B解除(CMPBCLR)値に達するまで増加する1傾斜動作で動くように構成設定することができます。その後、TCD周回は完了され、カウンタは\$000から再び始めて新しいTCD周回を開始します。TCD周回の周期は以下です。

$$T_{TCD周回} = \frac{(CMPBCLR+1)}{f_{CLK_TCD_CNT}}$$

この構成例では非重複出力が生成され、故に比較A設定(CMPASET) < 比較A解除(CMPACL R) < 比較B設定(CMPBSET) < 比較B解除(CMPBCLR)が使われます。



主クロック構成設定

TCDには次のように4つのクロック元があります。

- OSCHF
- PLL
- EXTCLK
- CLK_PER

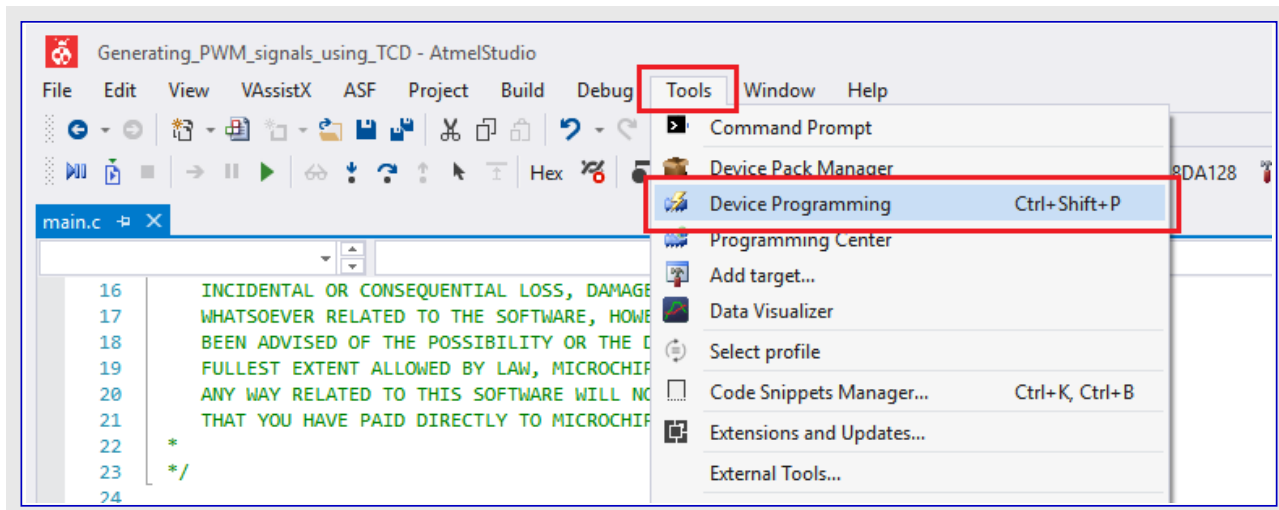
この例でのTCD構成設定はOSCHFクロック元がCLK_MAINとTCDに対する入力です。以下の構成設定は入力クロックとしてOSCHFを持ち24MHzで動くCPUとTCDを持つようにしなければなりません。

「6. 追補」で利用可能なコード例で主クロック初期化はCLK_Init()関数で行われます。

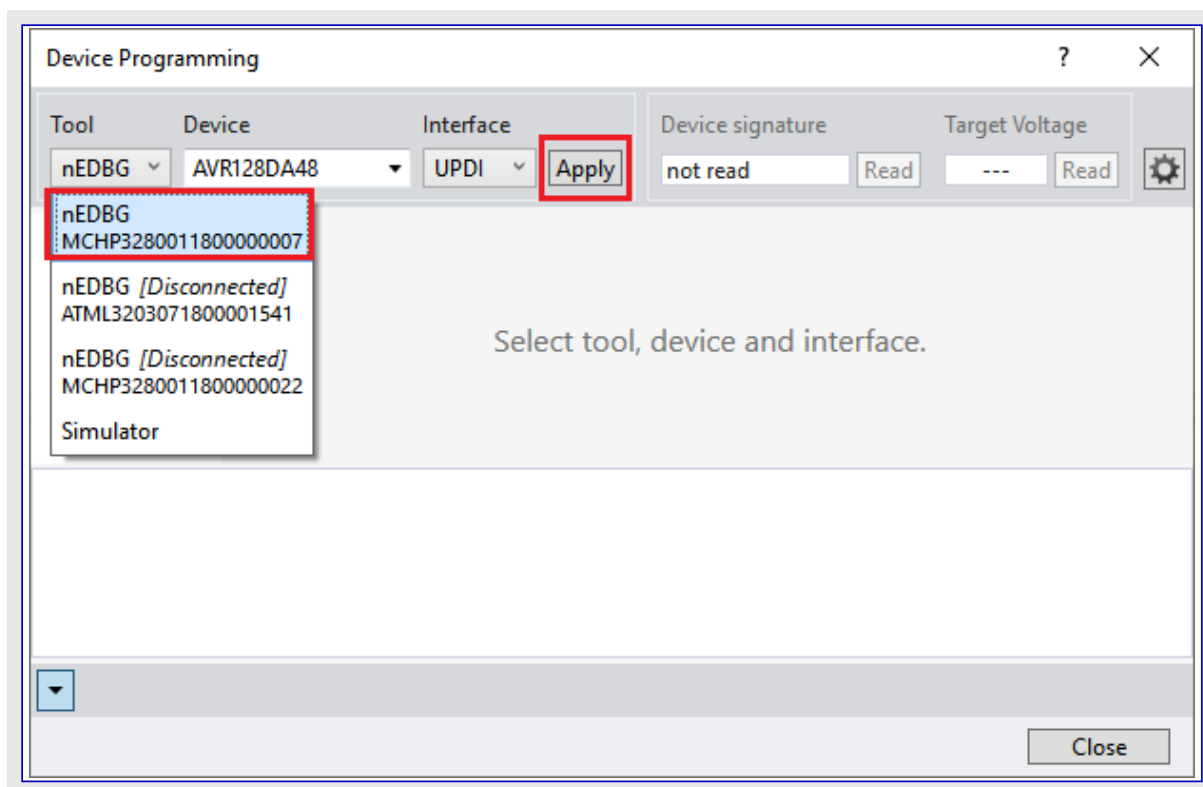
OSCHFに対する既定クロック元設定(任意選択)

これはAtmel Studioを使って以下の手順に従うことによって行うことができます。

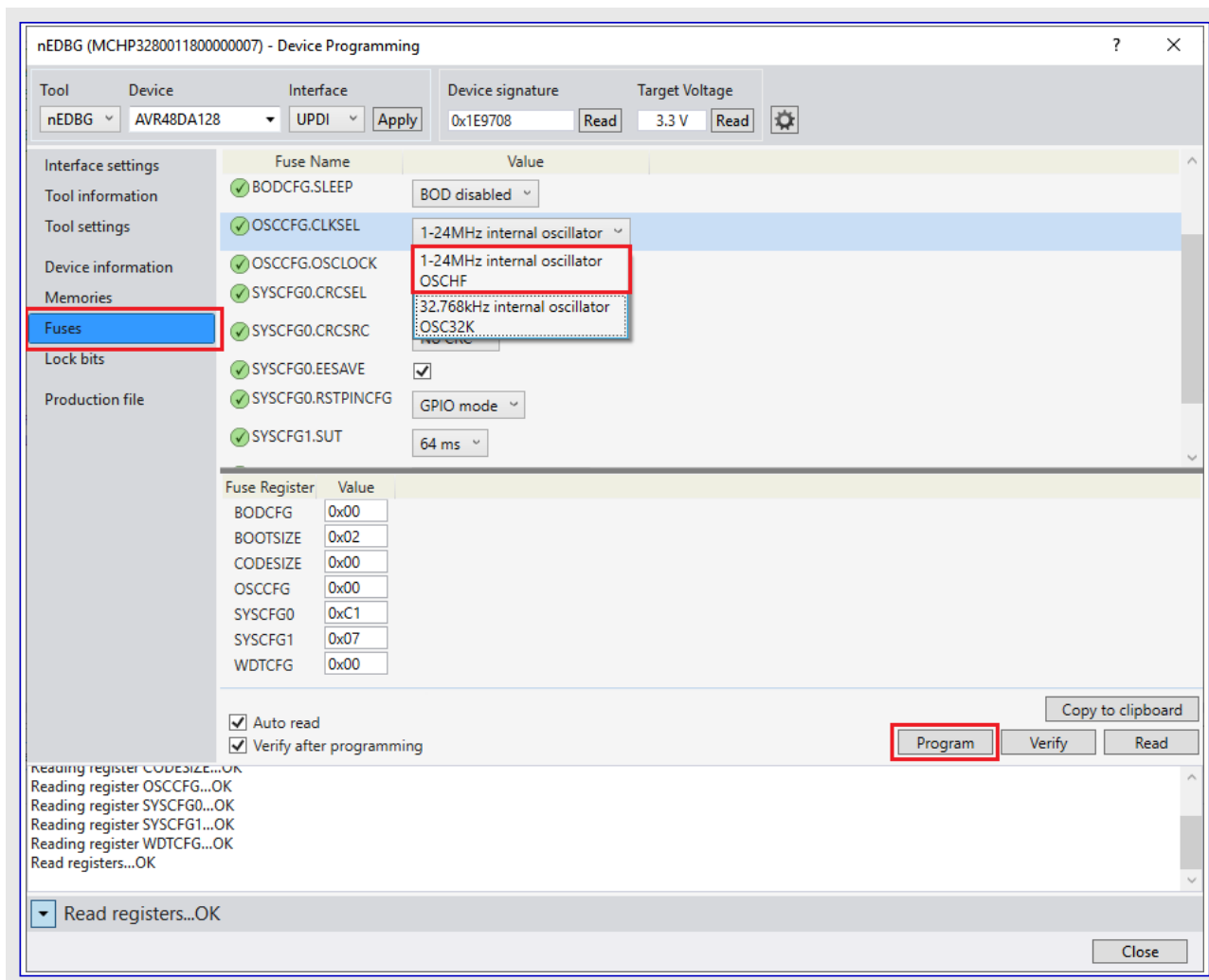
- Tools(ツール)⇒Device Programming(デバイス書き込み)をクリックしてください。



b. 取り付けた開発基板を選んでApply(適用)をクリックしてください。



c. 左側からFuses(ヒューズ)タブを選んでOSCCFG.CLKSELに対して1-24MHz internal oscillator OSCHF(1~24MHz内部発振器 OSCHF)を選び、その後にProgram(書き込み)をクリックしてください。



クロック元をOSCHFに変更して24MHzで動くための構成設定は以下の3つの段階で記述されます。

1. 主クロックに対するクロック元としてOSCHFを設定してください。

図3-2. CLKCTRL.MCLKCTRLAレジスタ構成設定

ビット	7	6	5	4	3	2	1	0
	CLKOUT				CLKSEL3~0			
アクセス種別	R/W	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1,0 – CLKSEL3~0 : 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 0	0 0 0 1	0 0 1 0	0 0 1 1	その他
名称	OSCHF	OSC32K	XOSC32K	EXTCLK	-
説明	内部高周波数発振器	32.768kHz内部発振器	32.768kHz外部クリスタル用発振器	外部クロック	(予約)

主クロック制御A(MCLKCTRLA)レジスタは構成設定変更保護(CCP:Configuration Change Protection)機構によって保護され、レジスタ内容を変更するのに制限時間書き込み手順を必要とします。CCP保護されたレジスタへの書き込みは以下のAPIが使われなければなりません。

```
_PROTECTED_WRITE(register, value);
```

OSCHFが選ばれなければならない、これはクロック選択(CLKSEL)ビット領域が'0000'の値に設定されなければならないことを意味します。これは次のようなコードに変換します。

```
_PROTECTED_WRITE(CLKCTRL.MCLKCTRLA, CLKCTRL_CLKSEL_OSCHF_gc);
```

2. クロック切り替え処理完了を待ってください。

図3-3. CLKCTRL.MCLKSTATUSレジスタ

ビット	7	6	5	4	3	2	1	0
			PLLS	EXTS	XOSC32KS	OSC32KS	OSCHFS	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。

クロック切替処理は主クロック発振器変更(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を経験する間停止しなければならず、故に切替が終わるまでの待機が実装されます。

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

3. 24MHzで動くようにOSCHFを設定してください。

図3-4. CLKCTRL.OSCHFCTRLAレジスタ構成設定

ビット	7	6	5	4	3	2	1	0	
	RUNSTDBY		FRQSEL3~0						AUTOTUNE
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	1	1	0	0	

● ビット5~2 – FRQSEL3~0 : 周波数選択 (Frequency Select)

このビット領域は発振器の出力周波数を選びます。

値	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	その他
名称	1MHz	2MHz	3MHz	4MHz	-	8MHz	12MHz	16MHz	20MHz	24MHz	-
説明 (出力周波数)	1MHz	2MHz	3MHz	4MHz (既定)	(予約)	8MHz	12MHz	16MHz	20MHz	24MHz	(予約)

高周波数発振器制御A(OSCHFCTRLA)レジスタの周波数選択(FRQSEL)ビット領域の既定リセット値は'0011'で、既定周波数値が4MHzであることを意味します。望む24MHzの周波数を得るにはFRQSELビット領域の内容が'1001'に変更されなければなりません。このビット領域はCCP機構を使い、故に保護された書き込みが実行されます。次のコードはOSCHFに対して24MHz出力を選びます。

```
_PROTECTED_WRITE (CLKCTRL.OSCHFCTRLA, CLKCTRL.FREQSEL_24M_gc);
```

PA4とPA5のピンを出力として構成設定

PA4とPA5のピンはWOAとWOBのPWM信号用に出力ピンとして構成設定されなければなりません。次のコード断片はPA4とPA5のピンをHigh出力として設定します。

```
PORTA.DIRSET |= PIN4_bm | PIN5_bm;
PORTA.OUTSET |= PIN4_bm | PIN5_bm;
```

「6. 追補」で利用可能なコード例でピン初期化はPORT_Init()関数で行われます。

TCD入力クロックと動作形態の構成設定

1傾斜動作で構成設定したTCDとクロック入力としてOSCHFを使って2つのPWM信号を生成するには以下のレジスタが変更されなければなりません。

- TCD制御A (TCD0.CTRLA)
- TCD制御B (TCD0.CTRLB)
- TCD比較A設定 (TCD0.CMPASET)
- TCD比較A解除 (TCD0.CMPACLR)
- TCD比較B設定 (TCD0.CMPBSET)
- TCD比較B解除 (TCD0.CMPBCLR)

「6. 追補」で利用可能なコード例でTCD初期化はTCD_Init()関数で行われます。

1. 波形生成動作を選んでTCDを構成設定してください。

図3-5. TCD0.CTRLBレジスタ構成設定

ビット	7	6	5	4	3	2	1	0	
								WGMODE1,0	
アクセス種別	R	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 – WGMODE1,0 : 波形生成動作 (Waveform Generation Mode)

これらのビットは波形生成を選びます。

値	0 0	0 1	1 0	1 1
名称	ONERAMP	TWORAMP	FOURRAMP	DS
説明	1傾斜動作	2傾斜動作	4傾斜動作	2重勾配動作

TCD0を1傾斜動作で使うにはTCD0.CTRLBレジスタの波形生成動作(WGMODE)ビット領域が'00'に設定されなければなりません。次のコード断片はTCD0を1傾斜動作で構成設定します。

```
TCD0.CTRLB |= TCD_WGMODE_ONERAMP_gc;
```

1傾斜動作ではWOA信号に対する'沈黙時間A'と'ON時間A'を設定するのにTCD0.CMPASETとTCD0.CMPACLRLのレジスタが使われ、加えて、WOB信号に対する'沈黙時間B'と'ON時間B'を設定するのにTCD0.CMPBSETとTCD0.CMPBCLRのレジスタが使われます。

TCD0の計数器が継続的に増して溢れるため、CMPASET<CMPACLRL<CMPBSET<CMPBCLR設定はON時間の間に重ならない出力に帰着します。

TCDは12ビット タイマ/カウンタなので\$000~\$FFFに対応する0~4095(4096段階)の範囲です。25%のPWMデューティサイクルについてはWOAが16進形式での\$400に対応する(下でON_TIME_CYCLES_WOAマクロによって定義される)1024クロック周期のON時間を持たなければなりません。

$$ON_TIME_CYCLES_WOA = (25/100) \times 4096 = 1024 = \$400$$

これはTCD0.CMPACLRLとTCD0.CMPASETの値間の差が1024でなければならないことを意味します。この使用事例の実例の目的についてはWOAに対するON時間の開始が1023です(下でON_TIME_START_WOAマクロによって定義されます)。

$$TCD0.CMPASET = ON_TIME_START_WOA = 1023 = \$3FF$$

$$TCD0.CMPACLRL = ON_TIME_START_WOA + ON_TIME_CYCLES_WOA = 1023 + 1024 = 2047 = \$7FF$$

30%のPWMデューティサイクルについてはWOBが16進形式での\$508に対応する(下でON_TIME_CYCLES_WOBマクロによって定義される)1228クロック周期のON時間を持たなければなりません。

$$ON_TIME_CYCLES_WOB = (30/100) \times 4096 = \text{約}1228 = \$508$$

これはTCD0.CMPBCLRとTCD0.CMPBSETの値間の差が1228でなければならないことを意味します。この使用事例の実例の目的についてはWOBに対するON時間の開始が2457です(下でON_TIME_START_WOBマクロによって定義されます)。

$$TCD0.CMPBSET = ON_TIME_START_WOB = 2457 = \$999$$

$$TCD0.CMPBCLR = ON_TIME_START_WOB + ON_TIME_CYCLES_WOB = 2457 + 1228 = 3745 = \$EA1$$

以下のコード断片は25%と30%のデューティサイクルを持つPWM信号を生成するためにTCD0.CMPASET、TCD0.CMPACLRL、TCD0.CMPBSET、TCD0.CMPBCLRを対応する値で初期化します。

```
#define ON_TIME_START_WOA    0x3FF
#define ON_TIME_CYCLES_WOA  0x400
#define ON_TIME_START_WOB   0x999
#define ON_TIME_CYCLES_WOB  0x508
```

```
TCD0.CMPASET = ON_TIME_START_WOA;
TCD0.CMPACLRL = ON_TIME_START_WOA + ON_TIME_CYCLES_WOA;
TCD0.CMPBSET = ON_TIME_START_WOB;
TCD0.CMPBCLR = ON_TIME_START_WOB + ON_TIME_CYCLES_WOB;
```

2. 波形チャネルを出力として許可してください。

図3-6. TCD0.FAULTCTRLレジスタ構成設定

ビット	7	6	5	4	3	2	1	0
	CMPDEN	CMPDEN	CMPDEN	CMPDEN	CMPDEN	CMPDEN	CMPDEN	CMPDEN
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4,5,6,7 - CMPAEN,CMPBEN,CMPDEN,CMPDEN : 比較x許可 (Compare x Enable)

これらのビットはピンでの出力として比較からの波形を許可します。

- ビット0,1,2,3 - CMPA,CMPB,CMPC,CMPD : 比較x値 (Compare Value x)

これらのビットは比較波形出力の既定状態を設定します。

PWM信号を生成するためにWOAとWOBの2つの出力チャネルが許可されなければなりません。加えて、この使用事例を例示するため、2つの波形出力の既定状態はHighです。障害制御(TCD0.FAULTCTRL)レジスタが構成設定変更保護下のため、CMPAEN、CMPBEN、CMPB、CMPAのビットはPROTECTED_WRITE APIを使って書かれなければなりません。

以下のコード断片は出力チャネルを許可して波形出力をHighに設定します。

```
_PROTECTED_WRITE(TCD0.FAULTCTRL, TCD_CMPAEN_bm | TCD_CMPA_bm)
| TCD_CMPBEN_bm | TCD_CMPB_bm);
```


3. TCDが許可する準備が整っているか調べてください。

図3-7. TCD0.STATUSレジスタ

ビット	7	6	5	4	3	2	1	0
	PWMACTB		PWMACTA				CMDRDY	ENRDY
アクセス種別	R/W	R/W	R	R	R	R	R	R
リセット値	0	0	0	0	0	0	0	0

● ビット0 - ENRDY : 許可準備可 (Enable Ready)

この状態ビットは制御A(TCDn.CTRLA)レジスタの許可(ENABLE)値がTCDクロック領域に同期され、再び書かれる準備が整った時を知らせます。

以下の活動がENRDYビットを解除(0)します。

1. TCDn.CTRLAレジスタの許可(ENABLE)ビットへの書き込み
2. 制御E(TCDn.CTRLE)レジスタのTCD周回の最後で禁止発動(DISEOC)の瞬発(ストロブ)
3. デバッグ制御(TCDn.DBGCTRL)レジスタのデバッグ時走行(DBGRUN)ビットが'0'の間で、チップ上デバッグ(OCD)作業で中断に行く

TCDを許可するには最初に準備が整っているかを調べなければなりません。以下のコード断片はTCDが許可の準備が整うまでの待機を実行します。

```
while (!(TCD0.STATUS & TCD_ENRDY_bm))
{
    ;
}
```

4. 入力クロック元を選んでTCDを許可してください。

図3-8. TCD0.CTRLAレジスタ構成設定

ビット	7	6	5	4	3	2	1	0
		CLKSEL1,0		CNTPRES1,0		SYNCPRES1,0		ENABLE
アクセス種別	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6,5 - CLKSEL1,0 : クロック選択 (Clock Select)

クロック選択ビットはTCDクロックのクロック元を選びます。

値	0 0	0 1	1 0	1 1
名称	OSCHF	PLL	EXTCLK	CLK_PER
説明	内部高周波数発振器	PLL	外部クロック	周辺機能クロック

● ビット0 - ENABLE : 許可 (Enable)

このビットを書くと、自動的にTCDクロック領域に同期されます。

このビットはこのビットの同期化が進行中でない限り変更することができます。状態(TCDn.STATUS)レジスタの許可準備可(ENRDY)ビットをご覧ください。

このビットは許可保護されません。

値	0	1
名称	NO	YES
説明	TCDは禁止されます。	TCDは許可され動きます。

次のコード断片は入力周波数に対してOSCHFを選んでTCDを許可します。

```
TCD0.CTRLA |= TCD_CLKSEL_OSCHF_gc | TCD_ENABLE_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。

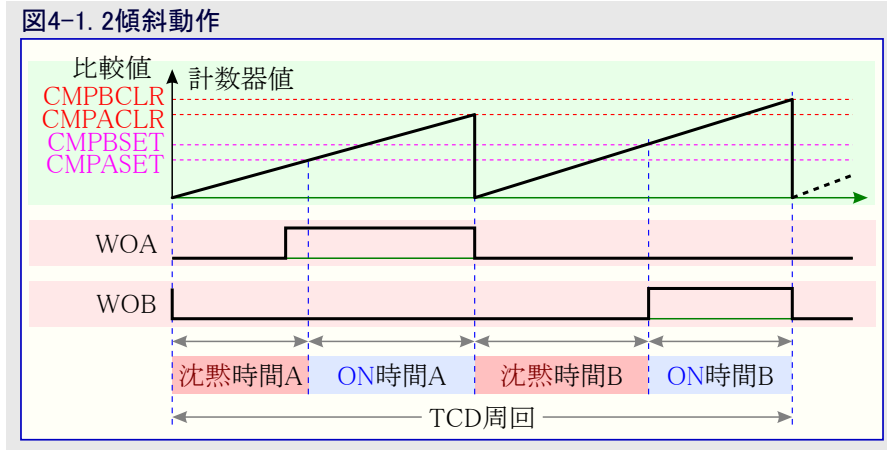


助言: 完全なコード例は「6. 追補」でも利用可能です。

4. 2傾斜動作で2つのPWM信号生成

使用事例説明: TCDは48MHzで動く入力クロックとしてPLLを持ち、PA4ピンでのWOAとPA5ピンでのWOBで2つのPWM信号を生成する2傾斜動作で動くように初期化されて構成設定されます。WOAのPWM信号は10%のデューティサイクルを持ち、WOB(PA5ピン)の信号は20%のデューティサイクルを持ちます。

結果: TCDはWOA(PA4ピン)で10%のデューティサイクルを持つ1つのPWM信号とWOBで20%のデューティサイクルを持つ1つのPWM信号を生成します。



2傾斜動作ではTCDが比較A解除(CMPACLR)値に達するまで上昇計数し、その後にはそれはリセットされて比較B解除(CMPBCLR)値に達するまで上昇計数します。そして、TCD周期が完了し、計数器は\$000から再び開始します。

$$T_{TCD\text{周期}} = \frac{(CMPACLR+1+CMPBCLR+1)}{f_{CLK_TCD_CNT}}$$

主クロック構成設定

PLLからTCD用の最高入力周波数を得るため、OSCHFは最高周波数(24MHz)で動くように構成設定されます。PLLによって達成可能な最高周波数は48MHzで、故にPLLに対して2xの通倍係数が使われます。

更に、OSCHFはCLK_MAIN用のクロック元としても扱います。

「6. 追補」で利用可能なコード例で主クロック初期化はCLK_Init()関数で行われます。

1. 主クロック用クロック元としてOSCHFを設定してください。次のコード断片は主クロックをOSCHF発振器に切り替えます。

```
_PROTECTED_WRITE(CLKCTRL.MCLKCTRLA, CLKCTRL.CLKSEL_OSCHF_gc);
```

2. クロック切り替え処理の完了を待ってください。次のコード断片はクロック元切り替え処理の終了を待つ方法を実演します。

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

3. OSCHFを24MHzで動くように設定してください。次のコード断片はOSCHFの周波数を24MHzに設定します。

```
_PROTECTED_WRITE(CLKCTRL.OSCHFCTRLA, CLKCTRL.FREQSEL_24M_gc);
```

4. PLL設定を構成設定してください。最大PLL周波数が48MHzのため2xの通倍係数が選ばれます。PLLシステムは未だ活動していませんが、TCDによってクロック元として要求されると、始動して生成します。

```
_PROTECTED_WRITE(CLKCTRL.PLLCTRLA, CLKCTRL.MULFAC_2x_gc);
```

PA4とPA5のピンを出力として構成設定

PA4とPA5のピンはWOAとWOBのPWM信号用に出力ピンとして構成設定されなければなりません。次のコード断片はPA4とPA5のピンをHigh出力として設定します。

```
PORTA.DIRSET |= PIN4_bm | PIN5_bm;
PORTA.OUTSET |= PIN4_bm | PIN5_bm;
```

注: 「6. 追補」で利用可能なコード例でピン初期化はPORT_Init()関数で行われます。

TCD入力クロックと動作形態の構成設定

2傾斜動作で構成設定したTCDとクロック入力としてPLLを使って2つのPWM信号を生成するには以下のレジスタが変更されなければなりません。

- TCD制御A (TCD0.**CTRLA**)
- TCD制御B (TCD0.**CTRLB**)
- TCD比較A設定 (TCD0.**CMPASET**)
- TCD比較A解除 (TCD0.**CMPACLR**)
- TCD比較B設定 (TCD0.**CMPBSET**)
- TCD比較B解除 (TCD0.**CMPBCLR**)

「6. 追補」で利用可能なコード例でTCD初期化はTCD_Init()関数で行われます。

1. 波形生成動作を選んでTCDを構成設定してください。

図4-2. TCD0.CTRLBレジスタ構成設定

ビット	7	6	5	4	3	2	1	0
							WGMODE1,0	
アクセス種別	R	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 – WGMODE1,0 : 波形生成動作 (Waveform Generation Mode)
これらのビットは波形生成を選びます。

値	0 0	0 1	1 0	1 1
名称	ONERAMP	TWORAMP	FOURRAMP	DS
説明	1傾斜動作	2傾斜動作	4傾斜動作	2重勾配動作

TCD0を2傾斜動作で使うにはTCD0.CTRLBレジスタの波形生成動作(WGMODE)ビット領域が'01'に設定されなければなりません。次のコード断片はTCD0を2傾斜動作で構成設定します。

```
TCD0.CTRLB |= TCD_WGMODE_TWORAMP_gc;
```

TCDは12ビット タイマ/カウンタなので\$000~\$FFFに対応する0~4095(4096段階)の範囲です。10%のPWMデューティサイクルについてはWOAが16進形式での\$199に対応する(下でON_TIME_CYCLES_WOAマクロによって定義される)409クロック周期のON時間を持たなければなりません。

```
ON_TIME_CYCLES_WOA = (10/100) × 4096 = 409 = $199
```

これはTCD0.CMPACLRとTCD0.CMPASETの値間の差が409でなければならないことを意味します。この使用事例についてはWOAに対するON時間の開始が1023です(下でON_TIME_START_WOAマクロによって定義されます)。

```
TCD0.CMPASET = ON_TIME_START_WOA = 1023 = $3FF
```

```
TCD0.CMPACLR = ON_TIME_START_WOA + ON_TIME_CYCLES_WOA = 1023 + 409 = 1423 = $598
```

20%のPWMデューティサイクルについてはWOBが16進形式での\$333に対応する(下でON_TIME_CYCLES_WOBマクロによって定義される)819クロック周期のON時間を持たなければなりません。

```
ON_TIME_CYCLES_WOB = (20/100) × 4096 = 819 = $333
```

これはTCD0.CMPBCLRとTCD0.CMPBSETの値間の差が819でなければならないことを意味します。この使用事例についてはWOBに対するON時間の開始が1023です(下でON_TIME_START_WOBマクロによって定義されます)。

```
TCD0.CMPBSET = ON_TIME_START_WOB = 1023 = $3FF
```

```
TCD0.CMPBCLR = ON_TIME_START_WOB + ON_TIME_CYCLES_WOB = 1023 + 819 = 1842 = $732
```

以下のコード断片は10%と20%のデューティサイクルを持つPWM信号を生成するためにTCD0.CMPASET、TCD0.CMPACLR、TCD0.CMPBSET、TCD0.CMPBCLRを対応する値で初期化します。

```
#define ON_TIME_START_WOA    0x3FF
#define ON_TIME_CYCLES_WOA  0x199
#define ON_TIME_START_WOB    0x3FF
#define ON_TIME_CYCLES_WOB  0x333
```

```
TCD0.CMPASET = ON_TIME_START_WOA;
```

```
TCD0.CMPACLR = ON_TIME_START_WOA + ON_TIME_CYCLES_WOA;
```

```
TCD0.CMPBSET = ON_TIME_START_WOB;
```

```
TCD0.CMPBCLR = ON_TIME_START_WOB + ON_TIME_CYCLES_WOB;
```

2. 波形チャネルを出力として許可してください。次のコード断片は出力チャネルを許可して波形出力をHighに設定します。

```
_PROTECTED_WRITE(TCD0.FAULTCTRL, TCD_CMPAEN_bm | TCD_CMPA_bm)
                  | TCD_CMPBEN_bm | TCD_CMPB_bm);
```

3. TCDが許可する準備が整っているか調べてください。次のコード断片はTCDが許可の準備が整うまでの待機を実行します。

```
while (!(TCD0.STATUS & TCD_ENRDY_bm))
{
    ;
}
```

4. 入力クロック元を選んでTCDを許可してください。

図4-3. TCD0.CTRLAレジスタ構成設定

ビット	7	6	5	4	3	2	1	0
	CLKSEL1,0		CNTPRES1,0		SYNCPRES1,0		ENABLE	
アクセス種別	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6,5 – CLKSEL1,0 : クロック選択 (Clock Select)

クロック選択ビットはTCDクロックのクロック元を選びます。

値	0 0	0 1	1 0	1 1
名称	OSCHF	PLL	EXTCLK	CLK_PER
説明	内部高周波数発振器	PLL	外部クロック	周辺機能クロック

● ビット0 – ENABLE : 許可 (Enable)

このビットを書くと、自動的にTCDクロック領域に同期されます。

このビットはこのビットの同期化が進行中でない限り変更することができます。状態(TCDn.STATUS)レジスタの許可準備可(ENRDY)ビットをご覧ください。

このビットは許可保護されません。

値	0	1
名称	NO	YES
説明	TCDは禁止されます。	TCDは許可され動きます。

次のコード断片は入力周波数に対してPLLを選んでTCDを許可します。

```
TCD0.CTRLA |= TCD_CLKSEL_PLL_gc | TCD_ENABLE_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「6. 追補」でも利用可能です。

5. 参考資料

1. AVR128DA28/32/48/64 暫定データシート
2. AVR128DA48 Curiosity Nano 使用者の手引き

6. 追補

例6-1. OSCHFでのTCD 1傾斜動作コード例

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

#define ON_TIME_START_WOA 0x3FF
#define ON_TIME_CYCLES_WOA 0x400
#define ON_TIME_START_WOB 0x999
#define ON_TIME_CYCLES_WOB 0x508

void CLK_Init(void);
void PORT_Init(void);
void TCD_Init(void);
```

```
void CLK_Init(void)
{
    /* 主クロック元としてOSCHFを設定 */
    _PROTECTED_WRITE(CLKCTRL.MCLKCTRLA, CLKCTRL_CLKSEL_OSCHF_gc);

    /* 主クロック発振器変更完了待機 */
    while (CLKCTRL.MCLKSTATUS & CLKCTRL_SOSC_bm)
    {
        ;
    }

    /* OSCHF周波数を24MHzに変更 */
    _PROTECTED_WRITE(CLKCTRL.OSCHFCTRLA, CLKCTRL_FREQSEL_24M_gc);
}

void PORT_Init(void)
{
    /* ポートAの4番ピンと5番ピンをHigh出力に構成設定 */
    PORTA.DIRSET |= PIN4_bm | PIN5_bm;
    PORTA.OUTSET |= PIN4_bm | PIN5_bm;
}

void TCD_Init(void)
{
    /* 1傾斜動作を選択 */
    TCD0.CTRLB |= TCD_WGMODE_ONERAMP_gc;

    /* 比較での設定と解除のレジスタを設定 */
    TCD0.CMPASET = ON_TIME_START_WOA;
    TCD0.CMPACLR = ON_TIME_START_WOA + ON_TIME_CYCLES_WOA;

    TCD0.CMPBSET = ON_TIME_START_WOB;
    TCD0.CMPBCLR = ON_TIME_START_WOB + ON_TIME_CYCLES_WOB;

    /* PWMチャンネルを許可 */
    _PROTECTED_WRITE(TCD0.FAULTCTRL, TCD_CMPAEN_bm | TCD_CMPA_bm);
    _PROTECTED_WRITE(TCD0.FAULTCTRL, TCD_CMPBEN_bm | TCD_CMPB_bm);

    /* TCDを許可する準備が整うのを待機 */
    while (!(TCD0.STATUS & TCD_ENRDY_bm))
    {
        ;
    }

    /* 入力クロックとしてOSCHFを選んでTCDを許可 */
    TCD0.CTRLA |= TCD_CLKSEL_OSCHF_gc | TCD_ENABLE_bm;
}

int main(void)
{
    CLK_Init();
    PORT_Init();
    TCD_Init();

    while (1)
    {
        ;
    }
}
```

例6-2. PLLでのTCD 2傾斜動作コード例

```

#include <avr/io.h>

#define ON_TIME_START_WOA    0x3FF
#define ON_TIME_CYCLES_WOA  0x199
#define ON_TIME_START_WOB    0x3FF
#define ON_TIME_CYCLES_WOB  0x333

void CLK_Init(void);
void PORT_Init(void);
void TCD_Init(void);

void CLK_Init(void)
{
    /* 主クロック元としてOSCHFを設定 */
    _PROTECTED_WRITE(CLKCTRL.MCLKCTRLA, CLKCTRL.CLKSEL_OSCHF_gc);

    /* 主クロック発振器変更完了待機 */
    while (CLKCTRL.MCLKSTATUS & CLKCTRL.SOSC_bm)
    {
        ;
    }

    /* OSCHF周波数を24MHzに変更 */
    _PROTECTED_WRITE(CLKCTRL.OSCHFCTRLA, CLKCTRL.FREQSEL_24M_gc);

    /* PLL用通倍係数を2xに設定 */
    _PROTECTED_WRITE(CLKCTRL.PLLCTRLA, CLKCTRL.MULFAC_2x_gc);
}

void PORT_Init(void)
{
    /* ポートAの4番ピンと5番ピンをHigh出力に構成設定 */
    PORTA.DIRSET |= PIN4_bm | PIN5_bm;
    PORTA.OUTSET |= PIN4_bm | PIN5_bm;
}

void TCD_Init(void)
{
    /* 2傾斜動作を選択 */
    TCD0.CTRLB |= TCD_WGMODE_TWORAMP_gc;

    /* 比較での設定と解除のレジスタを設定 */
    TCD0.CMPASET = ON_TIME_START_WOA;
    TCD0.CMPACLR = ON_TIME_START_WOA + ON_TIME_CYCLES_WOA;
    TCD0.CMPBSET = ON_TIME_START_WOB;
    TCD0.CMPBCLR = ON_TIME_START_WOB + ON_TIME_CYCLES_WOB;

    /* PWMチャンネルを許可 */
    _PROTECTED_WRITE(TCD0.FAULTCTRL, TCD_CMPAEN_bm | TCD_CMPA_bm);
    _PROTECTED_WRITE(TCD0.FAULTCTRL, TCD_CMPBEN_bm | TCD_CMPB_bm);

    /* TCDを許可する準備が整うのを待機 */
    while (!(TCD0.STATUS & TCD_ENRDY_bm))
    {
        ;
    }

    /* 入力クロックとしてPLLを選んでTCDを許可 */
    TCD0.CTRLA |= TCD_CLKSEL_PLL_gc | TCD_ENABLE_bm;
}

```

```

}

int main(void)
{
    CLK_Init();
    PORT_Init();
    TCD_Init();

    while (1)
    {
        ;
    }
}

```

7. 改訂履歴

文書改訂	日付	注釈
A	2020年2月	初版文書公開
B	2020年3月	貯蔵庫リンク更新 最新の商標により、AVR-DAをAVR [®] MCU DA (AVR-DA)に更新
C	2020年5月	最新の商標により、AVR [®] MCU DA (AVR-DA)をAVR [®] DA MCUに、AVR-DAをAVR DAに更新

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
- **Microchip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<http://www.microchip.com/pcn>へ行って登録指示に従ってください。

お客様支援

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

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

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

技術支援は<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、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、TempTracker、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。

APT、ClockWorks、The Embedded Control Solutions Company、EtherSynch、FlashTec、Hyper Speed Control、HyperLight Load、IntelliMOS、Liberio、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、BlueSky、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、MPASM、MPF、MPLAB Certifiedロゴ、MPLAB、MPLINK、MultiTRAK、NetDetach、Omniscient Code Generation、PICDEM、PICDEM.net、PICkit、PICtail、PowerSmart、PureSilicon、QMatrix、REALICE、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。

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<http://www.microchip.com/quality>を訪ねてください。

日本語© HERO 2020.

本技術概説はMicrochipのTB3236技術概説(DS90003236C-2020年5月)の翻訳日本語版です。日本語では不自然となる重複する形容表現は省略されている場合があります。日本語では難解となる表現は大幅に意識されている部分もあります。必要に応じて一部加筆されています。頁割の変更により、原本より頁数が少なくなっています。

必要と思われる部分には()内に英語表記や略称などを残す形で表記しています。

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

世界的な販売とサービス

米国	亜細亜/太平洋	亜細亜/太平洋	欧州
本社 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-792-7200 Fax: 480-792-7277 技術支援: http://www.microchip.com/support ウェブアドレス: http://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: 49-7131-72400 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 - デルネン 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
アトランタ Duluth, GA Tel: 678-957-9614 Fax: 678-957-1455 オースチン TX Tel: 512-257-3370 ホーストン 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			