

AT01616 : WeXタイマ/カウンタ拡張の使い方

Atmel AVR XMEGA E

要点

- タイマ/カウンタ比較チャンネル分配用の出力配列
 - ポートピンに渡って構成設定可能な比較チャンネル出力の分配
 - TC4とTC5間での沈黙時間挿入資源の再分配
- 各々が以下を持つ、4つの沈黙時間挿入(DTI)部
 - 8ビット分解能
 - 独立したHigh側とLow側の沈黙時間設定
 - 2重緩衝された沈黙時間
- 4つの交換(SWAP)部
 - 独立したポート対またはHigh/Low側駆動部の交換
 - 2重緩衝された交換機能
- ポートピンに渡って同期したビット様式を生成する模様型生成部
 - 2重緩衝された模様型生成
- 出力禁止

概要

この応用記述はAtmel[®] XMEGA[®] Eで利用可能なタイマ/カウンタ4/5に対する波形拡張(WeX:Waveform Extension)の様々な機能を記述します。

以前のXMEGAのタイマ/カウンタに対するAWeX拡張に照らして違いと改良を詳述します([AVR[®]1311応用記述](#)をご覧ください)。

代表的な応用でWeXの使用を簡単にするコード例が含まれます。

この料理本で言及される全てのソフトウェア例は[Atmelソフトウェア枠組み](#)(ASF:Atmel Software Framework)で提供されます。

1. 用語集	3
2. 事前要件	3
3. WeX	3
3.1. WeX概要	3
3.2. WeX対AWeX	3
4. 出力配列	4
4.1. Hブリッジ例	4
4.2. 000構成設定	5
4.3. 001構成設定	5
4.3.1. 応用	5
4.4. 010構成設定	5
4.4.1. この動作形態での応用の例	5
4.5. 011構成設定	6
4.5.1. 応用	6
4.6. 100構成設定	6
4.6.1. 応用	6
4.7. レジスタ	6
4.8. 例1 (出力配列制御)	6
4.8.1. ドライバ	6
4.8.1.1. 書き込み関数	6
4.8.1.2. 読み込み関数	6
4.8.2. 例	7
5. 沈黙時間挿入	7
5.1. 概要	7
5.2. 応用目的	8
5.3. WeX DTI改良	8
5.4. レジスタ	9
5.5. 沈黙時間挿入例	9
5.5.1. ドライバ	9
5.5.2. 例	10
6. 交換	10
6.1. 概要	10
6.2. 応用	11
6.2.1. 低速減衰動作	11
6.2.2. 高速減衰動作	11
6.2.3. 混合減衰動作	11
6.3. SWAP WeX改良	11
6.4. レジスタ	12
6.5. SWAP例	12
6.5.1. ドライバ	12
6.5.2. 例	13
7. 模様型生成器	14
7.1. 応用	14
7.2. 模様型生成器WeX改良	14
7.2.1. 模様型生成器緩衝部	14
7.2.2. レジスタ更新	14
7.3. レジスタ	14
7.4. 模様型生成器例	15
7.4.1. ドライバ	15
7.4.2. 例	16
8. 出力無効化禁止	17
8.1. レジスタ	17
8.2. 出力無効化禁止例	17
8.2.1. ドライバ	17
8.2.2. 例	17
9. 改訂履歴	17

1. 用語集

WeX	波形拡張
AWeX	以前のXMEGA A,B,C,Dの新波形拡張
OTMX	出力配列
DTI	沈黙時間挿入
ASF	Atmelソフトウェア枠組み
Atmel Studio	Atmel応用のための統合開発環境(IDE:Integrated Development Environment)
SMPS	スイッチング(動作)電源

2. 事前要件

この資料で検討される解決策は基本的に以下の技量と技術に精通していることが必要とされます。

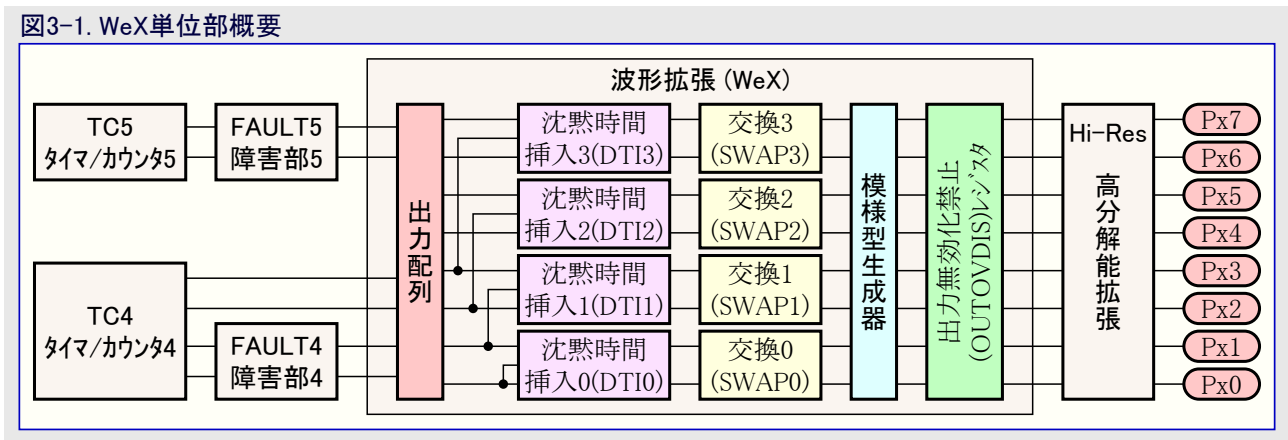
- XMEGA E 手引書
- XMEGA E データシート
- AWeX AVR1311 応用記述
- Atmel Studio 5 または 6
- Atmel デバッグ AVR JTAGICE mkII または JTAGICE3
- Atmel STK[®]600 開始キット

3. WeX

3.1. WeX 概要

波形拡張(WeX)は波形生成(WG:Waveform Generation)動作でタイマ/カウンタに追加機能を提供します。これは主に電動機制御、安定器、LED、Hブリッジ、電力変換器の各種形式と電力制御応用のその他の形式で使われることを意図されます。図3-1.で示されるように、WeXは5つの独立した補助機能から成ります。この概要例は2つのタイマ/カウンタ(TC4とTC5)を持つAtmel ATxmega32E5構成設定です。

図3-1. WeX単位部概要



3.2. WeX対AWeX

WeXはATxmega製品での標準AWeX単位部と比べて以下の改良点を提供します。

新機能:

- 出力配列
- 交換
- 出力禁止

改良された機能:

- 模様型生成器
- 沈黙時間挿入(DTI:Dead-Time Insertion)
- 障害(下をご覧ください。)

他の変更:

- FAULT拡張は標準AWeX説明に含まれる障害機能に対して新しく改良された機能を持つ新しい独立型拡張です。

4. 出力配列

新しい出力配列補助機能は(SMPSのような)電力制御系と照明応用のためにWeXで開発されました。

出力配列(OTMX:Output Matrix)は表4-1.で詳述される構成設定に従ってポートピンに渡ってタイマ/カウンタの波形出力を分配することができます。

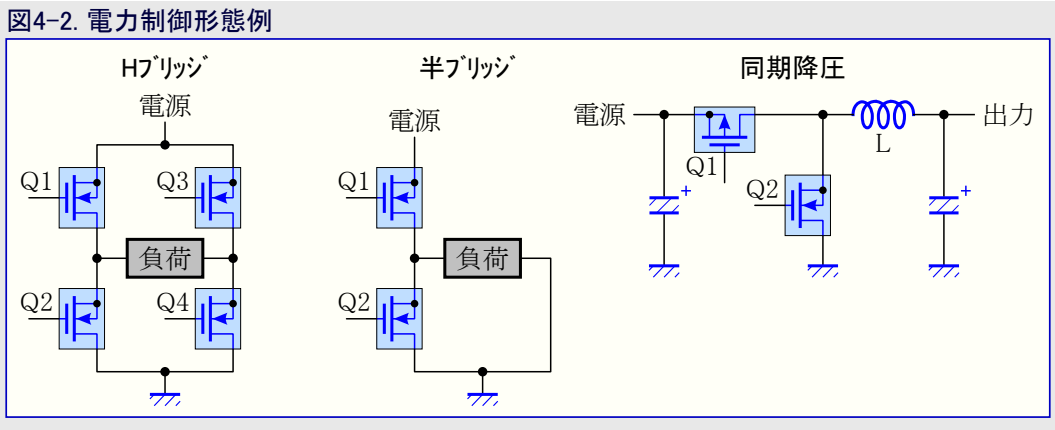
表4-1. タイマ/カウンタ4と5の比較チャネルピンの配線構成設定

OTMX2~0	Px7	Px6	Px5	Px4	Px3	Px2	Px1	Px0	主な応用
0 0 0	-	-	TC5CCB	TC5CCA	TC4CCD	TC4CCC	TC4CCB	TC4CCA	リセット構成設定
0 0 1	TC5CCB	TC5CCA	TC5CCB	TC5CCA	TC4CCD	TC4CCC	TC4CCB	TC4CCA	Hブリッジ/SMPS+電動機制御
0 1 0	TC5CCB	TC5CCA	TC4CCB	TC4CCA	TC5CCB	TC5CCA	TC4CCB	TC4CCA	Hブリッジ/SMPS+電動機制御
0 1 1	TC4CCA	TC4CCA	TC4CCA	TC4CCA	TC4CCA	TC4CCA	TC4CCA	TC4CCA	LED
1 0 0	TC4CCB	TC4CCA	TC4CCA	TC4CCA	TC4CCA	TC4CCA	TC4CCA	TC4CCA	LED

注意: 出力配列がWeX内で最初の補助機能のため、以降の補助機能(DTI、SWAP、模様型生成器、ポート無効化)はこの配列の出力信号に適用されず。

表4-1.はDTI、SWAP、模様型生成器がリセット構成設定の場合にだけ有効です。そうでなければ、TCxCCy出力はそれらが出力ピンに達するまでに補助機能構成設定に従って変換されます。

違う応用形態にWeX出力を適合させるために違う出力配列動作を使うことができます。いくつかの異なる電力制御形態が図4-2.で示されます。



4.1. Hブリッジ例

照明応用は半分のまたは完全なHブリッジを使います。

DC電動機制御応用もHブリッジを制御するのに各種切り替え機構を使います。

いくつかの切り替え機構は表4-2.と図4-3.で一覧にされます。

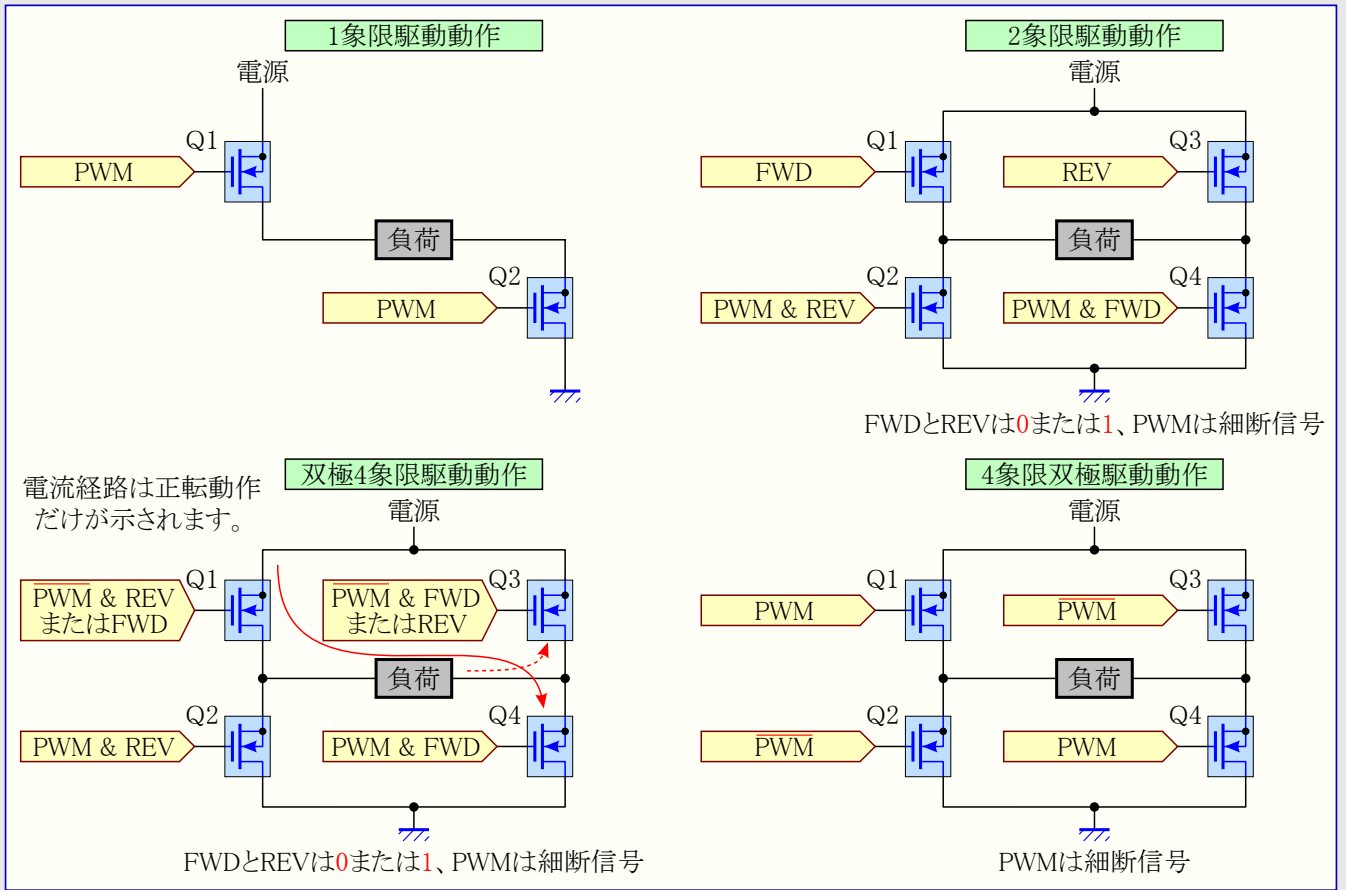
各種出力配列は一般的な制御系をこれら異なる切り替え機構に適合させるのに有用で有り得ます。

表4-2. Hブリッジ制御機構要件

象限	双極/単極	PWM信号	要DTI	模様型生成器使用	計時器動作
1	双極	1 PWM	×	×	-
2	双極	1 PWM	×	×	-
4	双極	2補完PWM	○	○	-
4	単極	2補完PWM	○	×	中央整列動作

(訳注) 表4-1.でOTMX2~0=100のPx7のTC4CCBと6頁の4.6. 100構成設定項の記述でのTC5CCAは明らかに矛盾しています(他に数ヶ所)。本書は原書に従って矛盾したままにしています。どちらが正しいかは不明です。

図4-3. Hブリッジ制御機構図



タイマ/カウンタ4とタイマ/カウンタ5の6つの出力全てを出力配列の以下の構成設定で再配線または上書きすることができます。

4.2. 000構成設定

000はリセット既定構成設定です。

4.3. 001構成設定

この構成設定は2つの他のピン位置でタイマ/カウンタ5比較チャンネルAとB(TC5CCAとTC5CCB)からの波形出力を2重化します。

OTMX2~0	Px7	Px6	Px5	Px4	Px3	Px2	Px1	Px0	主な応用
0 0 1	TC5CCB	TC5CCA	TC5CCB	TC5CCA	TC4CCD	TC4CCC	TC4CCB	TC4CCA	Hブリッジ/SMPS+電動機制御

故にTC5(CCAとCCB)出力は各々、Px4/6とPx5/7上で同時に利用可能です。

4.3.1. 応用

この動作は例えば、全Hブリッジの4つのトランジスタを制御するのに使うことができます。

4.4. 010構成設定

OTMX2~0	Px7	Px6	Px5	Px4	Px3	Px2	Px1	Px0	主な応用
0 1 0	TC5CCB	TC5CCA	TC4CCB	TC4CCA	TC5CCB	TC5CCA	TC4CCB	TC4CCA	Hブリッジ/SMPS+電動機制御

この動作はタイマ/カウンタ4と5の両方の比較チャンネルAとB(CCAとCCB)からの波形生成出力を2つの他のピン位置に分配します。

4.4.1. この動作形態での応用の例

この配列動作は以下のように使うことができます。

- 使用者が計時器TC5からDTIで比較チャンネルを得ることが必要な場合にDTIが設定されたなら、出力は次のとおりです。

OTMX2~0	Px7	Px6	Px5	Px4	Px3	Px2	Px1	Px0	主な応用
0 1 0	TC5CCB LS	TC5CCA HS	TC4CCB LS	TC4CCA HS	TC5CCB LS	TC5CCA HS	TC4CCB LS	TC4CCA HS	Hブリッジ/SMPS+電動機制御

- 動的制動を必要とする応用では、この動作が同じPWM出力で冗長出力を提供します。この機能はお客様応用での外部論理回路を節約します。
- 2つの全Hブリッジを駆動することが必要とされる応用

4.5. 011構成設定

この動作は共通波形動作と等価です。

タイマ/カウンタ4比較チャンネルA(TC4CCA)からの波形出力を全てのポートピンに分配します。

4.5.1. 応用

この構成設定はステップング電動機を制御するために模様型生成器と共に使うことができます。

照明応用については、複数LED列を制御するのにこの動作が有用で有り得ます。

4.6. 100構成設定

この構成設定はTC5CCAからの波形出力を7番(Px7)ピンへとTC4CCAからの波形出力を他の全ての(Px6~0)ピンに配給します。

4.6.1. 応用

この構成設定を使う応用例は多分以下でしょう。

- 1~7つのLED列の制御
- LED列とLEDに給電するDC/DC変換器の同時制御。この場合、TC5がDC/DC変換器(降圧または昇圧)のトランジスタ制御に使われ、同時にLED電流調整閉路がTC4を制御します。

4.7. レジスタ

ビット +\$00	7	6	5	4	3	2	1	0	CTRL
	UPSEL	OUTMX2~0			DTI3EN	DTI2EN	DTI1EN	DTI0EN	
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6~4 - OTMX2~0 : 出力配列 (Output Matrix)

これらのビットは表4-1.に従ってポートピンに対するタイマ/カウンタ波形生成出力の配線配列を定義します。

4.8. 例1 (出力配列制御)

このプロジェクトは各種動作に対して出力配列を構成設定する方法を提供します。

4.8.1. ドライバ

出力配列を構成設定するためのドライバはASFに含まれます。

`¥src¥asf¥xmega¥drivers¥tcdriver/tc.h`

パラメータ:

- *WEXはWEX単位部(WEXC)へのポインタです。
- `otmx`は出力配列動作です。

4.8.1.1. 書き込み関数

出力配列を構成設定するためのドライバはASFに含まれます。

```
static inline void tc45_WEX_set_otmx(WEX_t *WEX, enum wex_otmx_mode_t otmx)
{
    ((WEX_t *)WEX)->CTRL = (((WEX_t *)WEX)->CTRL & ~WEX_OTMX_gm) | otmx;
}
```

これは指定した出力配列動作でWEXを構成設定します。

```
例: tc45_WEX_set_otmx (&WEXC, WEX_OTMX_DEFAULT);
     tc45_WEX_set_otmx (&WEXC, WEX_OTMX_1);
     tc45_WEX_set_otmx (&WEXC, WEX_OTMX_2);
     tc45_WEX_set_otmx (&WEXC, WEX_OTMX_3);
     tc45_WEX_set_otmx (&WEXC, WEX_OTMX_4);
```

4.8.1.2. 読み込み関数

```
static inline uint16_t tc45_WEX_read_otmx(WEX_t *WEX)
{
    return (((WEX_t *)WEX)->CTRL & WEX_OTMX_gm);
}
```

4.8.2. 例

構成設定

TC4計時器:

比較/捕獲単位部A/B : 比較と1傾斜動作での波形生成

ポートC0/C1とC2/C3の出力はデューティサイクル付き : 周期/4と周期/2

WeXC : OTMX構成設定は変更することができます。

TC5計時器:

比較/捕獲単位部A/B : 比較と1傾斜動作での波形生成

ポートC4/C5の出力はデューティサイクル付き : 周期/8(66MHzとF/4で472 μ s)と周期/6(66MHzとF/4で616 μ s)

WeXC : OTMX構成設定は変更することができます。

OTMX:

OTMXを構成設定するための関数 : `tc45_WEX_set_otmx(..)`が変更されなければなりません。

結果

OTMX 0動作での出力は次のとおりです。

PC0/PC1 : TC4 CCA/CCB

PC2/PC3 : TC4 CCC/CCD

PC4/PC5 : TC5 CCA/CCB

OTMX 1動作での出力は次のとおりです。

PC0/PC1 : TC4 CCA/CCB

PC2/PC3 : TC4 CCC/CCD

PC4/PC5 : TC5 CCA/CCB

PC6/PC7 : TC5 CCA/CCB

OTMX 2動作での出力は次のとおりです。

PC0/PC1 : TC4 CCA/CCB

PC2/PC3 : TC5 CCA/CCB

PC4/PC5 : TC4 CCA/CCB

PC6/PC7 : TC5 CCA/CCB

OTMX 3動作での出力は次のとおりです。

PC0~PC7 : TC4 CCA

OTMX 4動作での出力は次のとおりです。

PC0~PC6 : TC4 CCA

PC7 : TC4 CCB

5. 沈黙時間挿入

5.1. 概要

補完(コンプリメント)出力動作で動くトランジスタの対によって駆動されるシステムでは、同じ側で2つのFETを同時に許可することが完全に禁止されます。これは貫通(電源から接地への回路短絡)になります。

電力出力デバイスは急に切り替えることができないので、補完対の1つのPWM出力のOFF切り替え事象と他のトランジスタのON切り替え事象間に或る量の時間が提供されなければなりません。

PWM制御の沈黙時間機能は、出力生成中の駆動部の動作速度のためにPWMの同じ組(PWMxHとPWMxL)の駆動部が同時にONになることを避けます。

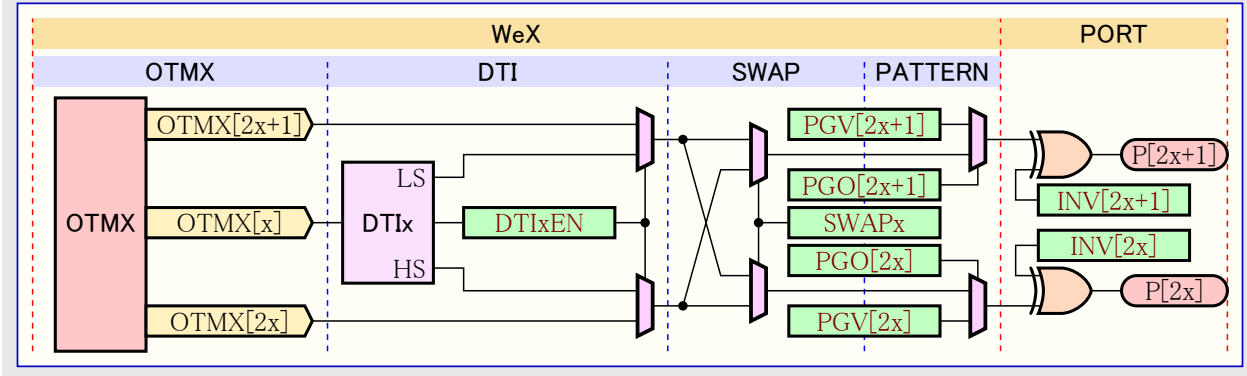
この要件はATxmegaのAWeXでのDTI補助機能を記述する[AVR1311応用記述](#)でも説明されます。

沈黙時間はPWM入出力ピンの対のどれかが補完出力動作で動いている時に挿入されます。

4つの沈黙時間挿入(DTI)機能(DTI0~DTI3)は4つの下位側OTMX出力を制御します([図3-1. WeX単位部概要](#)をご覧ください)。

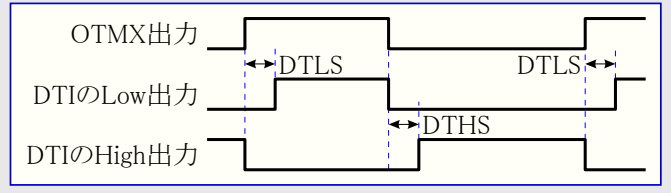
[図5-1.](#)はポートピン対での1つの沈黙時間挿入単位部活動を示します。

図5-1. DTI概要



High側とLow側に対する沈黙時間は各々DTHSとDTLSのレジスタを通して個別に設定することができます(図5-2をご覧ください)。沈黙時間のDTHSとDTLSは全てのOTMX出力に共通です。

図5-2. DTI活動



DTI0はハードウェア沈黙時間を持つ2つの補完PWM出力でOTMX0出力を変形し、出力はポートピンの0と1(Py0とPy1)です。DTI1はハードウェア沈黙時間を持つ2つの補完PWM出力でOTMX1出力を変形し、出力はポートピンの2と3(Py2とPy3)です。DTInはハードウェア沈黙時間を持つ2つの補完PWM出力でOTMXn出力を変形し、出力はポートピンの2xと2x+1(Py2xとPy2x+1)です。近道のため、DTHSとDTLSはDTBOTHレジスタに書くことによって同じ値を設定することができます。

沈黙時間値は主システムクロック周期数で与えられます。沈黙時間に対して許される範囲は従って0~255主システムクロック周期です。沈黙時間長は短すぎるべきではなく、さもなければ同じ駆動部対が未だ同時に活性にされるかもしれません。これは良い効率を保証するために長すぎるべきではありません。正しい沈黙時間長は選ばれた駆動部の速度に従って選ばれるべきです。この沈黙時間は通常、100ns~2.5μs(現実の結果はMOSFETのゲート電圧対入力容量の傾斜のために1.5μs)です。故に最大沈黙時間は8ビットレジスタで構成設定することができます。

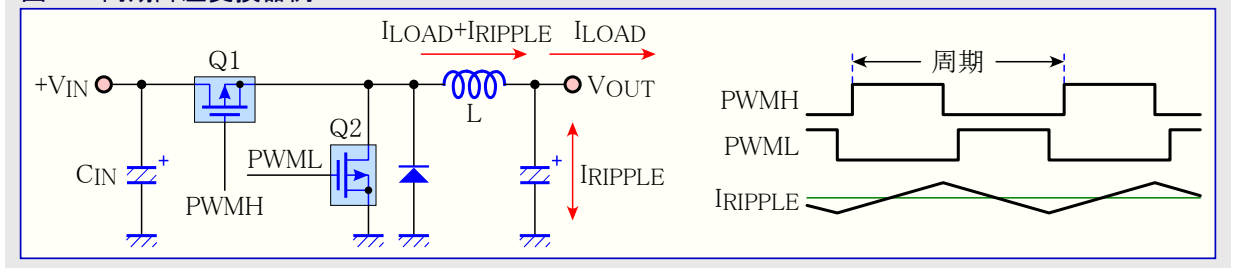
5.2. 応用目的

沈黙時間挿入はトランジスタの対が補完(コンプリメンタリ)出力動作で動いている場合に必須です。

この動作は例えば以下で使われます。

- 半Hブリッジ
- 2象限と4象限の双極制御でのHブリッジ
- 同期降圧変換器(図5-3をご覧ください。)
- 交互積層昇圧変換器

図5-3. 同期降圧変換器例



5.3. WeX DTI改良

AWeXのDTIに比したWeXのDTIの改良はDTIレジスタが模様型生成器補助機能と共用されないことです。

故にWeXでは組み込みDTIを持つPWM信号で模様型生成器を使うことができます。

にも関わらず、DTIレジスタは障害(FAULT)機能での障害消去(Blank)機能と共用されます(XMEGA E手引書の障害(FAULT)拡張をご覧ください)。故にWeXのDTIとFAULTの障害消去は同時に使うことができません。

5.4. レジスタ

ビット	7	6	5	4	3	2	1	0	
+\$00	UPSEL		OUTMX2~0		DTI3EN	DTI2EN	DTI1EN	DTI0EN	CTRL
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3~0 – DTIxEN : 沈黙時間挿入生成器x許可 (Dead-Time Insertion Generator x Enable)

これらのビットのどれかの設定(1)は対応する出力配列に対する沈黙時間挿入生成器を許可します。これは各々Low側とHigh側の波形を持つ出力配列[2x]と[2x+1]を無効にします。このビットは障害消去が許可される場合に0が読めます。

ビット	7	6	5	4	3	2	1	0	
+\$01	DTBOTH7~0								DTBOTH
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 – DTBOTH7~0 : 沈黙時間両側値 (Dead-Time Both Sides bits)

このレジスタへの書き込みは沈黙時間High側(DTHS)レジスタと沈黙時間Low側(DTLS)レジスタを同時に(換言すると同じI/Oアクセス書き込みで)更新します。この読み込みは\$00値を与えます。

ビット	7	6	5	4	3	2	1	0	
+\$02	DTLS7~0								DTLS
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 – DTLS7~0 : 沈黙時間Low側値 (Dead-Time Low Side bits)

このレジスタはLow側沈黙時間に対する周辺機能クロック周期数を保持します。

ビット	7	6	5	4	3	2	1	0	
+\$03	DTHS7~0								DTHS
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 – DTHS7~0 : 沈黙時間High側値 (Dead-Time High Side bits)

このレジスタはHigh側沈黙時間に対する周辺機能クロック周期数を保持します。

5.5. 沈黙時間挿入例

5.5.1. ドライバ

パラメータ:

- *WEXはWEX単位部(WEXC)へのポインタです。

関数

- CCAでの沈黙時間挿入許可

```
static inline void tc45_WEX_enable_cca_deadtime(WEX_t *WEX)
((WEX_t *)WEX)->CTRL |= WEX_DTIOEN_bm;
```

- CCAでの沈黙時間挿入禁止

```
static inline void tc45_WEX_disable_cca_deadtime(WEX_t *WEX)
((WEX_t *)WEX)->CTRL &= ~WEX_DTIOEN_bm;
```

- CCBでの沈黙時間挿入許可

```
static inline void tc45_WEX_enable_ccb_deadtime(WEX_t *WEX)
((WEX_t *)WEX)->CTRL |= WEX_DTI1EN_bm;
```

- CCBでの沈黙時間挿入禁止

```
static inline void tc45_WEX_disable_ccb_deadtime(WEX_t *WEX)
((WEX_t *)WEX)->CTRL &= ~WEX_DTI1EN_bm;
```

- CCCでの沈黙時間挿入許可

```
static inline void tc45_WEX_enable_ccc_deadtime(WEX_t *WEX)
((WEX_t *)WEX)->CTRL |= WEX_DTI2EN_bm;
```

- CCCでの沈黙時間挿入禁止

```
static inline void tc45_WEX_disable_ccc_deadtime(WEX_t *WEX)
((WEX_t *)WEX)->CTRL &= ~WEX_DTI2EN_bm;
```

- CCDでの沈黙時間挿入許可

```
static inline void tc45_WEX_enable_ccd_deadtime(WEX_t *WEX)
((WEX_t *)WEX)->CTRL |= WEX_DTI3EN_bm;
```

- CCDでの沈黙時間挿入禁止

```
static inline void tc45_WEX_disable_ccd_deadtime(WEX_t *WEX)
((WEX_t *)WEX)->CTRL &= ~WEX_DTI3EN_bm;
```

- High側沈黙時間構成設定

パラメータ値：沈黙時間値

```
static inline void tc45_WEX_set_dti_high(WEX_t *WEX, int16_t value)
((WEX_t *)WEX)->DTHS = value;
```

- Low側沈黙時間構成設定

```
static inline void tc45_WEX_set_dti_low(WEX_t *WEX, int16_t value)
((WEX_t *)WEX)->DTLS = value;
```

- 対称沈黙時間構成設定

```
static inline void tc45_WEX_set_dti_both(WEX_t *WEX, int16_t value)
((WEX_t *)WEX)->DTBOTH = value;
```

5.5.2. 例

関数

プログラムは以下の関数でCCx出力の沈黙時間を許可します。

```
tc45_WEX_enable_cca_deadtime(&WEXC);
tc45_WEX_enable_ccb_deadtime(&WEXC);
tc45_WEX_enable_ccc_deadtime(&WEXC);
tc45_WEX_enable_ccd_deadtime(&WEXC);
```

そして以下の関数でCCx出力の沈黙時間を構成設定します。

```
tc45_WEX_set_dti_high(&WEXC, 0x40);
tc45_WEX_set_dti_low(&WEXC, 0x40); /* 0x40=64 故に DT時間=64×1/(Fextern/4) ⇒ DT=64/66×4=3.87μs */
```

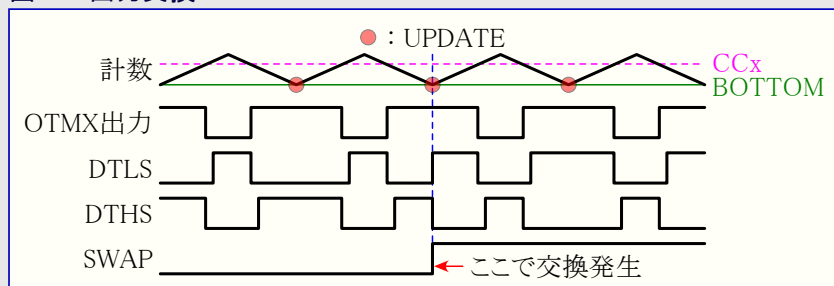
6. 交換

6.1. 概要

この新機能は計時器更新発生時に2つの出力信号を同時に切り替えるのに有用です。

交換操作はUPDATE(更新条件)で達成され、[図6-1.](#)で示されるように非常に短い(数100nsの)タイミング制限を提供することができます。

図6-1. 出力交換



[図3-1.](#)と[図5-1.](#)で示されるように、DTIとSWAPの単位部は4つのポート対部分として見るすることができます。

DTI0/SWAP0の0部分はポートピン(Px0, Px1)に従います。

DTI1/SWAP1の1部分はポートピン(Px2, Px3)に従います。

そしてより一般的には以下です。

DTIn/SWAPnのn部分はポートピン(Px[2n], Px[2n+1])に従います。

6.2. 応用

チャネル交換機能はBLDC電動機制御で非常に有用です。これは相に於いて上側と下側の即時の変更を許します。この機能を使うと、回転子整流(転流)と速度制御を独立した2つのプログラム部分に分けることができます。制御信号の状態はPWM値レジスタの内容を変えることなく、電動機位置(相整流)によって必要とされる時に直ちに更新することができます。これらの更新はPWMデューティサイクル更新に対して非同期に達成することができます。

一旦細断電流閾値に達すると、Hブリッジは以下のような2つの電流循環形態で動き得ます。

- 電流がダイオードを通して循環する場合の非同期動作。使用者は代替経路作成の出来事を制御することができません。
- 代替経路を推進するためにFETを許可/禁止する場合の同期動作。

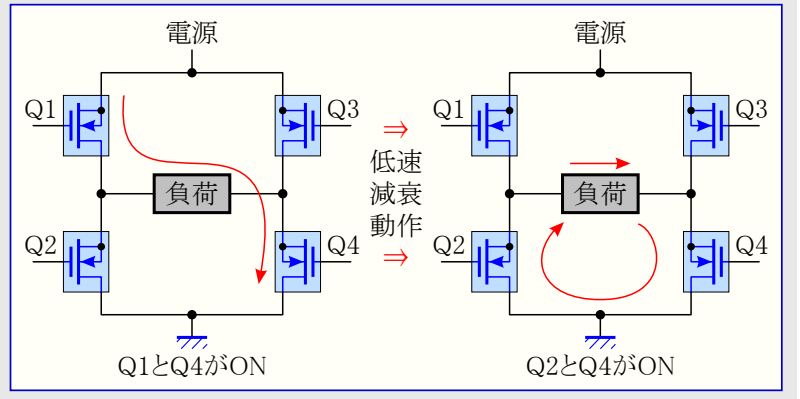
高速減衰と低速減衰の2つの同期動作を使うことができます。高速と低速は電動機速度ではなく、電流減衰動作を示します。これは速度に対して逆です。高速減衰動作では電動機が速度に於いてゆっくり低下する一方で、低速減衰動作では電動機が非常に素早く停止します。

6.2.1. 低速減衰動作

低速減衰動作は同じ上側または下側のHブリッジ区部を使います。大抵は2つの下側FETが使われます。電流指示は2つのFET経路を通して0に減らします。減衰時間はFETのRDS_{ON}に依存します。

低速減衰動作では回転子が非常に素早く停止します。

図6-2. 低速減衰動作



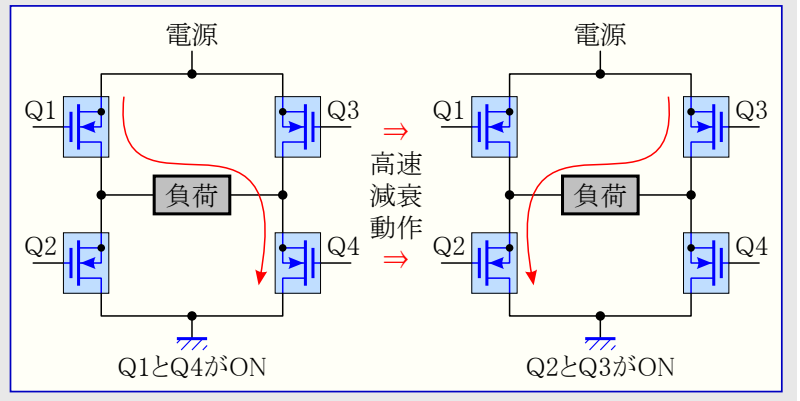
6.2.2. 高速減衰動作

高速減衰動作では、一旦PWM細断電流水準に達すると、Hブリッジは巻線電流に逆方向で流れることを許すために状態を逆にします。Hブリッジ内の電流の高速減少を生じる、流れる電流のための代替経路として逆側のFETが使われます。逆の方法で巻線電流と戦う電圧を印加しています。

巻線電流が0に近づく時に、ブリッジはどの逆電流の流れも防ぐために禁止されます。

高速減衰動作は同期整流とも名付けられ、主にステップング電動機応用(マイクロ ステッピング~)で使われます。高速減衰はFETの内部ダイオードやFETと並列の外部ダイオードに負担を掛けません。

図6-3. 高速減衰動作



6.2.3. 混合減衰動作

第3の減衰動作は混合減衰動作と呼ばれます。これは低速減衰動作と高速減衰動作の混合です。これも主にステップング電動機制御、特にマイクロ ステッピングに使われます。

6.3. SWAP WeX改良

この動作は以前のAWeXで利用不能でした。

交換はタイマ/カウンタ4または5の更新で達成することができます。

(CTRLレジスタの)更新元選択(UPSEL)ビットはタイマ/カウンタ4または5からのUPDATE(更新)事象選択を許します。

6.4. レジスタ

ビット	7	6	5	4	3	2	1	0	
+\$06	–	–	–	–	SWAP3	SWAP2	SWAP1	SWAP0	SWAP
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3~0 – SWAPx : DTI出力対交換 (Swap DTI Output Pair)

これらのビットの設定(1)は沈黙時間挿入(DTI)部出力の[2x]と[2x+1]の出力交換を許可します。沈黙時間挿入生成器x許可(DTIxEN)設定がこの交換操作に影響を及ぼさないことに注意してください。

ビット	7	6	5	4	3	2	1	0	
+\$0A	–	–	–	–	SWAP3BUF	SWAP2BUF	SWAP1BUF	SWAP0BUF	SWAPBUF
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3~0 – SWAPxBUF : DTI出力対交換 (Swap DTI Output Pair)

これらのレジスタビットは交換(SWAP)レジスタビット用の緩衝部です。2重緩衝が使われる場合、UPDATE条件でこれらのビット内の有効な内容が対応するSWAPxビットに複写されます。

ビット	7	6	5	4	3	2	1	0	
+\$04,\$05	–	–	–	–	–	SWAPBUFV	PGVBUFV	PGOBUFV	STATUSCLR STATUSSET
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

● ビット2 – SWAPBUFV : 交換緩衝有効 (SWAP Buffer Valid)

このフラグが設定(1)なら、交換緩衝部(SWAPBUF)が書かれて、次のUPDATE条件で交換(SWAP)レジスタ内に複写される有効なデータを含みます。このビットが0なら、何の動作も行われません。接続されたタイマ/カウンタの更新施錠(LUPD)ビットもSWAPレジスタに対する更新に影響を及ぼします。

ビット	7	6	5	4	3	2	1	0	
+\$00	UPSEL	OUTMX2~0			DTI3EN	DTI2EN	DTI1EN	DTI0EN	CTRL
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 – UPSEL : 更新元選択 (Update Source Selection)

既定によってこれらのレジスタ内容を更新するために交換(SWAP)と模様型生成部によってタイマ/カウンタ4の更新条件が使われます。このビットの設定(1)はレジスタ更新の供給元としてタイマ/カウンタ5の更新条件にします。

6.5. SWAP例

6.5.1. ドライバ

パラメータ:

- *WEXはWEX単位部(WEXC)へのポインタです。

関数

- OTMX 0と1での交換許可

```
static inline void tc45_WEX_enable_swap0(WEX_t *WEX)
    ((WEX_t *)WEX)->SWAP |= WEX_SWAPO_bm;
```

- OTMX 0と1での交換禁止

```
static inline void tc45_WEX_disable_swap0(WEX_t *WEX)
    ((WEX_t *)WEX)->SWAP &= ~WEX_SWAPO_bm;
```

- OTMX 0と1での交換緩衝部許可

```
static inline void tc45_WEX_enable_swap0_buffer(WEX_t *WEX)
    ((WEX_t *)WEX)->SWAPBUF |= WEX_SWAPOBUF_bm;
```

- OTMX 0と1での交換緩衝部禁止

```
static inline void tc45_WEX_disable_swap0_buffer(WEX_t *WEX)
    ((WEX_t *)WEX)->SWAPBUF &= ~WEX_SWAPOBUF_bm;
```

- OTMX 2と3での交換許可

```
static inline void tc45_WEX_enable_swap1(WEX_t *WEX)
    ((WEX_t *)WEX)->SWAP |= WEX_SWAP1_bm;
```

- OTMX 2と3での交換禁止

```
static inline void tc45_WEX_disable_swap1(WEX_t *WEX)
((WEX_t *)WEX)->SWAP &= ~WEX_SWAP1_bm;
```

- OTMX 2と3での交換緩衝部許可

```
static inline void tc45_WEX_enable_swap1_buffer(WEX_t *WEX)
((WEX_t *)WEX)->SWAPBUF |= WEX_SWAP1BUF_bm;
```

- OTMX 2と3での交換緩衝部禁止

```
static inline void tc45_WEX_disable_swap1_buffer(WEX_t *WEX)
((WEX_t *)WEX)->SWAPBUF &= ~WEX_SWAP1BUF_bm;
```

- OTMX 4と5での交換許可

```
static inline void tc45_WEX_enable_swap2(WEX_t *WEX)
((WEX_t *)WEX)->SWAP |= WEX_SWAP2_bm;
```

- OTMX 4と5での交換禁止

```
static inline void tc45_WEX_disable_swap2(WEX_t *WEX)
((WEX_t *)WEX)->SWAP &= ~WEX_SWAP2_bm;
```

- OTMX 4と5での交換緩衝部許可

```
static inline void tc45_WEX_enable_swap2_buffer(WEX_t *WEX)
((WEX_t *)WEX)->SWAPBUF |= WEX_SWAP2BUF_bm;
```

- OTMX 4と5での交換緩衝部禁止

```
static inline void tc45_WEX_disable_swap2_buffer(WEX_t *WEX)
((WEX_t *)WEX)->SWAPBUF &= ~WEX_SWAP2BUF_bm;
```

- OTMX 6と7での交換許可

```
static inline void tc45_WEX_enable_swap3(WEX_t *WEX)
((WEX_t *)WEX)->SWAP |= WEX_SWAP3_bm;
```

- OTMX 6と7での交換禁止

```
static inline void tc45_WEX_disable_swap3(WEX_t *WEX)
((WEX_t *)WEX)->SWAP &= ~WEX_SWAP3_bm;
```

- OTMX 6と7での交換緩衝部許可

```
static inline void tc45_WEX_enable_swap3_buffer(WEX_t *WEX)
((WEX_t *)WEX)->SWAPBUF |= WEX_SWAP3BUF_bm;
```

- OTMX 6と7での交換緩衝部禁止

```
static inline void tc45_WEX_disable_swap3_buffer(WEX_t *WEX)
((WEX_t *)WEX)->SWAPBUF &= ~WEX_SWAP3BUF_bm;
```

6.5.2. 例

構成設定

CCx割り込みは以下の関数で許可されます。

```
/* CCAとCCBの割り込み発生時に呼ばれる割り込み関数を宣言 */
tc45_set_cca_interrupt_callback(&TIMER_EXAMPLE_C, example_cca_interrupt_callback);
tc45_set_ccb_interrupt_callback(&TIMER_EXAMPLE_C, example_ccb_interrupt_callback);

/* タイマ/カウンタ4のCCA,CCB,CCC,CCD単位部の割り込みレベルを構成設定: Low */
tc45_set_cca_interrupt_level(&TIMER_EXAMPLE_C, TC45_INT_LVL_LO);
tc45_set_ccb_interrupt_level(&TIMER_EXAMPLE_C, TC45_INT_LVL_LO);
tc45_set_ccc_interrupt_level(&TIMER_EXAMPLE_C, TC45_INT_LVL_LO);
tc45_set_ccd_interrupt_level(&TIMER_EXAMPLE_C, TC45_INT_LVL_LO);
```

OTMX動作形態2が使われます。

“未交換動作”での出力は”PC7~0=TC5CCB/TC5CCA/TC4CCB/TC4CCA/TC5CCB/TC5CCA/TC4CCB/TC4CCA”です。

以下の命令はC0/C1の交換と更にC2/C3の交換を許します。

```
tc45_set_cca_interrupt_callback(&TIMER_EXAMPLE_C, example_cca_interrupt_callback);
```

以下の命令はC4/C5の交換と更にC6/C7の交換を許します。

```
tc45_set_ccb_interrupt_callback(&TIMER_EXAMPLE_C, example_ccb_interrupt_callback);
```

関数

● CCx割り込み例

比較/捕獲検出割り込み呼び戻し関数は比較Aチャネル(TIMER_C)で割り込みが起こる時に呼びられます。

それはCC検出指標を増し、従って2回に1回、模様型生成器出力を強制します。CCA割り込み関数の例は以下です(CCB割り込みは同じ割り込み手順を使います)。

```
static void example_cca_interrupt_callback(void)
{
    cca_pwm_index += 1;
    if (cca_pwm_index == 2)
    {
        cca_pwm_index = 0;
        tc45_WEX_enable_swap2(&WEXC); /* C4とC5の交換許可 */
        tc45_WEX_disable_swap3(&WEXC); /* C6とC7の交換禁止 */
    }
    else
    {
        tc45_WEX_disable_swap2(&WEXC); /* C4とC5の交換禁止 */
        tc45_WEX_enable_swap3(&WEXC); /* C6とC7の交換許可 */
    }
}
```

7. 模様型生成器

模様型生成部は一定の論理レベルで同期した出力波形を生成するのに使われます。

模様型生成を使うと、8つの出力のいくつかを一定レベルによって上書きすることができます。

他の2重緩衝されたタイマ/カウンタレジスタのように、レジスタ更新はタイマ/カウンタ波形生成動作によって設定されるUPDATE(更新)条件に同期されます。応用が同期の必要がない場合、応用コードは単純に直接PGO、PGV、PORTxのレジスタをアクセスすることができます。

7.1. 応用

以下の制御に対して模様型生成器を使うことができます。

- ステッピング電動機
- 電力Hブリッジ: 全ての象限構成設定に於いて柔軟な方法でHブリッジを制御することができます。表4-2. Hブリッジ制御機構要件をご覧ください。

7.2. 模様型生成器WeX改良

7.2.1. 模様型生成器緩衝部

新しい模様型生成器の出力許可緩衝部(PGOBUF)と出力値緩衝部(PGVBUF)はDTIと共用されていた以前のAWeXのDTBUFLSとDTBUFHSに比べて今や固有です。このように模様型生成器はDTI信号と共に使うことができます。

7.2.2. レジスタ更新

PGOBUFとPGVBUFの緩衝部の内容での模様型生成器レジスタ(PGOとPGV)の更新はタイマ/カウンタ4または5の計時器更新で達成することができます。

この更新は以前のAWeXでタイマ/カウンタ0のUPDATE(更新)でだけ可能でした。

新しい(CTRLレジスタの)UPSELビットはタイマ/カウンタ4または5からのUPDATE(更新)事象選択を許します。

7.3. レジスタ

ビット	7	6	5	4	3	2	1	0	
+\$07	PGO7~0								PGO
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 - PGO7~0: 模様型生成上書き (Pattern Generation Override)

このレジスタは各出力に対する模様型生成の許可を保持します。1のビット位置は対応する交換(SWAP)出力を各々の模様型生成値(PGV)ビット値で上書きします。

ビット	7	6	5	4	3	2	1	0	
+\$08	PGV7~0								PGV
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 - PGV7~0 : 模様型生成値 (Pattern Generation Value)**

このレジスタは各出力に対する模様型の値を保持します。

ビット	7	6	5	4	3	2	1	0	
+\$0B	PGOBUF7~0								PGOBUF
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 - PGOBUF7~0 : 模様型生成上書き緩衝値 (Pattern Generation Override Buffer)**

このレジスタは模様型生成上書き(PGO)レジスタ用の緩衝部です。2重緩衝が使われる場合、UPDATE条件でこのレジスタ内の有効な内容がPGOレジスタに複写されます。

ビット	7	6	5	4	3	2	1	0	
+\$0C	PGVBUF7~0								PGVBUF
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 - PGVBUF7~0 : 模様型生成値緩衝値 (Pattern Generation Value Buffer)**

このレジスタは模様型生成値(PGV)レジスタ用の緩衝部です。2重緩衝が使われる場合、UPDATE条件でこのレジスタ内の有効な内容がPGVレジスタに複写されます。

ビット	7	6	5	4	3	2	1	0	
+\$04,\$05	-	-	-	-	-	SWAPBUFV	PGVBUFV	PGOBUFV	STATUSCLR STATUSSET
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

● **ビット1 - PGVBUFV : 模様型生成器値緩衝有効 (Pattern Generator Value Buffer Valid)**

このフラグが設定(1)なら、模様型生成値緩衝部(PGVBUF)が書かれて、次のUPDATE条件で模様型生成値(PGV)レジスタ内に複写される有効なデータを含みます。このビットが0なら、何の動作も行われません。接続されたタイマ/カウンタの更新施錠(LUPD)ビットもPGV緩衝部の更新に影響を及ぼします。

● **ビット0 - PGOBUFV : 模様型生成器上書き緩衝有効 (Pattern Generator Overwrite Buffer Valid)**

このフラグが設定(1)なら、模様型生成上書き緩衝部(PGOBUF)が書かれて、次のUPDATE条件で模様型生成上書き(PGO)レジスタ内に複写される有効なデータを含みます。このビットが0なら、何の動作も行われません。接続されたタイマ/カウンタの更新施錠(LUPD)ビットもPGO緩衝部の更新に影響を及ぼします。

ビット	7	6	5	4	3	2	1	0	
+\$00	UPSEL	OUTMX2~0			DTI3EN	DTI2EN	DTI1EN	DTI0EN	CTRL
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 - UPSEL : 更新元選択 (Update Source Selection)**

既定によってそれらのレジスタ内容を更新するために交換(SWAP)と模様型生成部によってタイマ/カウンタ4の更新条件が使われます。このビットの設定(1)はレジスタ更新の供給元としてタイマ/カウンタ5の更新条件にします。

7.4. 模様型生成器例

7.4.1. ドライバ

パラメータ:

- *WEXはWEX単位部(WEXC)へのポインタです。

引数:

- PGO許可
- PGV値

関数

- 模様型生成器出力許可構成設定

```
static inline void tc45_WEX_write_pgo(WEX_t *WEX, int16_t value)
    ((WEX_t *)WEX)->PGO = (((WEX_t *)WEX)->PGO & 0x00) | value;
```

- 模様型生成器緩衝部出力許可構成設定

```
static inline void tc45_WEX_write_pgo_buffer(WEX_t *WEX, int16_t value)
    ((WEX_t *)WEX)->PGOBUF =value;
```

- 模様型生成器出力緩衝部読み込み

戻り値：模様型生成器緩衝部(PGOBUF)値

```
static inline uint16_t tc45_WEX_read_pgo_buffer(volatile void *WEX)
    return (((WEX_t *)WEX)->PGOBUF);
```

- 模様型生成器値構成設定

引数：PGV値

```
static inline void tc45_WEX_write_pgv(WEX_t *WEX, int16_t value)
    ((WEX_t *)WEX)->PGV = (((WEX_t *)WEX)->PGV & 0x00) | value;
```

- 模様型生成器値緩衝部構成設定

引数：PGVBUF値

```
static inline void tc45_WEX_write_pgv_buffer(WEX_t *WEX, int16_t value)
    ((WEX_t *)WEX)->PGVBUF = (((WEX_t *)WEX)->PGVBUF & 0x00) | value;
```

7.4.2. 例

構成設定

模様型生成器値を構成設定

```
tc45_WEX_write_pgv(&WEXC, 0xF0);          /* “1”模様型値書き込み */
tc45_WEX_write_pgv_buffer(&WEXC, 0xF0);
```

タイマ/カウンタ4のCCA,CCB,CCC,CCD単位部の割り込みレベルを構成設定: Low

```
tc45_set_cca_interrupt_level(&TIMER_EXAMPLE_C, TC45_INT_LVL_LO);
tc45_set_ccb_interrupt_level(&TIMER_EXAMPLE_C, TC45_INT_LVL_LO);
tc45_set_ccc_interrupt_level(&TIMER_EXAMPLE_C, TC45_INT_LVL_LO);
tc45_set_ccd_interrupt_level(&TIMER_EXAMPLE_C, TC45_INT_LVL_LO);
```

CCAとCCBの割り込み発生時に呼ばれる割り込み関数を宣言

```
tc45_set_cca_interrupt_callback(&TIMER_EXAMPLE_C, example_cca_interrupt_callback);
tc45_set_ccb_interrupt_callback(&TIMER_EXAMPLE_C, example_ccb_interrupt_callback);
tc45_set_ccc_interrupt_callback(&TIMER_EXAMPLE_C, example_ccc_interrupt_callback);
tc45_set_ccd_interrupt_callback(&TIMER_EXAMPLE_C, example_ccd_interrupt_callback);
```

関数

比較/捕獲検出割り込み呼び戻し関数は比較Aチャネル(TIMER_C)で割り込みが起こる時に呼ばれます。

それはCC検出指標を増し、従って2回に1回、模様型生成器出力を強制します。

CCAチャネル用の例:

```
static void example_cca_interrupt_callback(void)
{
    cca_pwm_index += 1;
    if (cca_pwm_index == 2)
    {
        cca_pwm_index = 0;
        value = tc45_WEX_read_pgo_buffer(&WEXC);
        value_tmx = tc45_WEX_read_otmx(&WEXC);
        if (value_tmx == 0x20) value |= 0x11;    /* 出力配列が010なら、ビット0と4を2重化(TC4CCA) */
        else value |= 0x01;
        tc45_WEX_write_pgo_buffer(&WEXC, value); /* 2回に1回、模様型生成器を許可 */
    }
    else
    {
        value = tc45_WEX_read_pgo_buffer(&WEXC);
        value_tmx = tc45_WEX_read_otmx(&WEXC);
    }
}
```

```

    if (value_tmx == 0x20) value &= 0xEE;
    else
        value &= 0xFE;
    tc45_WEX_write_pgo_buffer(&WEXC, value); /* 他回は模様型生成器を禁止 */
}
}

```

8. 出力無効化禁止

出力無効化禁止部は選択可能なポートピンでの波形出力を禁止するのに使うことができます。

この関数はピンの使い方を最適化することの有益性を提供します。応用が全てのポートピンに渡って波形出力を広げる必要がない時に、選ばれたピンを他の機能的な使用のために自由にさせることができます。

このポート無効化論理回路は全てのタイマ/カウンタ拡張に対して共通です。

WeX拡張は以下の全ての構成設定が存在すると直ぐにポートピンを制御します。

- タイマ/カウンタの比較チャンネルが波形生成用に許可(COMPまたはBOTHCC)
- 比較チャンネルがWeXの補助機能を通して交換または2重化
 - AVR1330:WeXタイマ/カウンタ拡張の使用

8.1. レジスタ

ビット +\$0F	7	6	5	4	3	2	1	0	
	OUTOVDIS7~0								OUTOVDIS
Read/Write	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

- ビット7~0 - OUTOVDIS7~0 : 出力無効化禁止 (Output Override Disable)

これらのビットは対応する出力ポートレジスタ(即ち、ピン位置に対する1ビット毎の関係で)の自動無効化を禁止します。

8.2. 出力無効化禁止例

8.2.1. ドライバ

パラメータ:

- *WEXはWEX単位部(WEXC)へのポインタです。

引数: 出力値

関数

- 出力無効化を構成設定

```

static inline void tc45_WEX_set_output_override(WEX_t *WEX, int16_t value)
{
    ((WEX_t *)WEX)->OUTOVDIS = (((WEX_t *)WEX)->OUTOVDIS & 0x00) | value;
}

```

8.2.2. 例

```

/* 出力禁止構成設定 */
tc45_WEX_set_output_override(&WEXC, 0x0F); /* 1を禁止として:上位側4出力が許可 */
// tc45_WEX_set_output_override(&WEXC, 0xF0); /* 1を禁止として:下位側4出力が許可 */

```

9. 改訂履歴

文書改訂	日付	注釈
42086A	2013年4月	初版文書公開



Enabling Unlimited Possibilities®

Atmel Corporation

1600 Technology Drive
San Jose, CA 95110
USA
TEL (+1)(408) 441-0311
FAX (+1)(408) 487-2600
www.atmel.com

Atmel Asia Limited

Unit 01-5 & 16, 19F
BEA Tower, Millennium City 5
418 Kwun Tong Road
Kwun Tong, Kowloon
HONG KONG
TEL (+852) 2245-6100
FAX (+852) 2722-1369

Atmel Munich GmbH

Business Campus
Parking 4
D-85748 Garching b. Munich
GERMANY
TEL (+49) 89-31970-0
FAX (+49) 89-3194621

Atmel Japan G.K.

141-0032 東京都品川区
大崎1-6-4
新大崎勸業ビル 16F
アトメル ジャパン合同会社
TEL (+81)(3)-6417-0300
FAX (+81)(3)-6417-0370

© 2013 Atmel Corporation. 不許複製 / 改訂:42086A-AVR-04/2013

Atmel®, Atmelロゴとそれらの組み合わせ、AVR®, Enabling Unlimited Possibilities®, STK®, XMEGA®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。

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。たとえば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、または利益と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。

© HERO 2021.

本応用記述はAtmelのAT01616応用記述(Rev.42086A-04/2013)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。