

セクション 13. 出力コンペア

ハイライト

本セクションには以下の主要項目を記載しています。

13.1	はじめに	13-2
13.2	出力コンペアレジスタ	13-3
13.3	出力コンペアの動作	13-5
13.5	省電力モード時の出力コンペアの動作	13-19
13.6	I/O ピンの制御	13-19
13.7	レジスタマップ	13-19
13.8	設計のヒント	13-22
13.9	関連アプリケーション ノート	13-23
13.10	改訂履歴	13-24

13.1 はじめに

本セクションでは、出力コンペア モジュールとその動作モードについて説明します。図 13-1 に、タイマ入力を含む出力コンペア モジュールのブロック図を示します。出力コンペア モジュールは、選択したモードに応じて、タイマの値を 1 つまたは 2 つのコンペアレジスタの値と比較します。タイマ値がコンペアレジスタ値に一致すると、出力ピンの状態を変化させます。出力コンペア モジュールは、コンペアー一致イベント時に出力ピンの状態を変化させる事により、単発の出力パルスまたは連続する出力パルスを生成します。コンペアー一致イベント時に割り込みを発生させる事もできます。

出力コンペア モジュールは下記の動作モードを備えます。

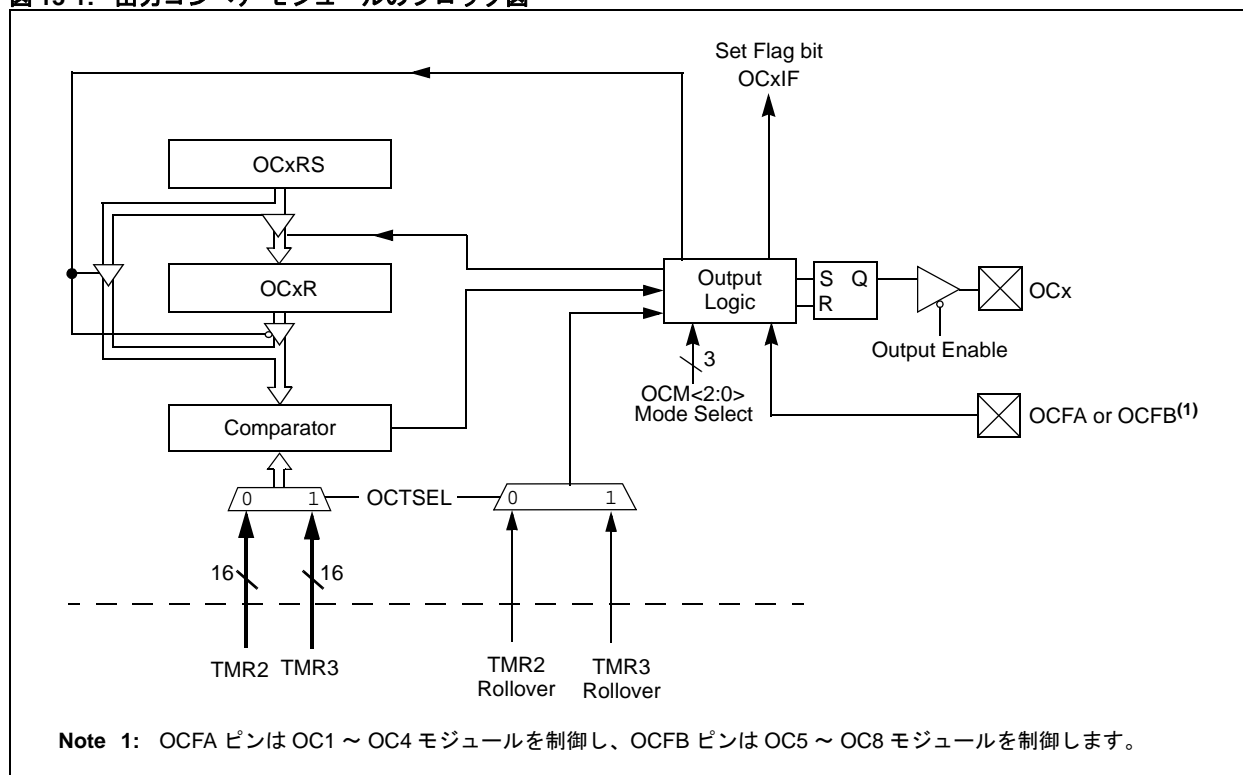
- アクティブ LOW ワンショット モード
- アクティブ HIGH ワンショット モード
- トグルモード
- 遅延ワンショットモード
- 連続パルスモード
- フォルト保護なしの PWM モード
- フォルト保護ありの PWM モード

Note 1: dsPIC33F ファミリの全てのデバイスは、1 つまたは複数の出力コンペア モジュールを備えます。各出力コンペア モジュールのタイムベースには Timer2 または Timer3 を選択できます。詳細は各デバイスのデータシートを参照してください。

2: ピン、制御ステータスビット、レジスタの名前で使用する添え字「x」は、出力コンペア モジュールの番号を表します (x = 1 ~ 8)。

3: レジスタ名で使用する添え字「y」は、タイマの番号を表します (y = 2 または 3)。

図 13-1: 出力コンペア モジュールのブロック図



13.2 出力コンペアレジスタ

各出力コンペア モジュールは下記のレジスタを備えます。

- OCxCON: 出力コンペア制御レジスタ
- OCxR: 出力コンペアレジスタ
- OCxRS: 出力コンペア セカンダリ レジスタ

以下に、これらのレジスタの概要を示します。

レジスタ 13-1: OCxCON: 出力コンペア x の制御レジスタ

U-0	U-0	R/W-0	U-0	U-0	U-0	U-0	R/W-0
—	—	OCSIDL	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	R-0, HC	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	OCFLT	OCTSEL	<OCM2:0>		
bit 7					bit 0		

凡例:	HC = ハードウェア内でクリア
R = 読み出し可能ビット	W = 書き込み可能ビット U = 未実装、「0」として読み出し
-n = POR 時の値	「0」= ビットをクリア

bit 15-14	未実装: 「0」として読み出し
bit 13	OCSIDL: 出力コンペア x のアイドルモード時停止制御ビット 1 = CPU アイドルモード時に出力コンペア x は停止する 0 = CPU アイドルモード時に出力コンペア x は動作を継続する
bit 12-5	未実装: 「0」として読み出し
bit 4	OCFLT: PWM フォルト条件ステータスビット 1 = PWM フォルト条件が発生した (ハードウェアでのみクリア) 0 = PWM フォルト条件は発生していない (このビットは OCM<2:0> = 111 の場合にのみ使用)
bit 3	OCTSEL: 出力コンペア x のタイマ選択ビット 1 = Timer3 を出力コンペア x のクロック源に使用する 0 = Timer2 を出力コンペア x のクロック源に使用する
bit 2-0	OCM<2:0>: 出力コンペア x のモード選択ビット 111 = フォルト保護ありの PWM モード; OCx ピンでの PWM モード、フォルトピン有効 110 = フォルト保護なしの PWM モード; OCx ピンでの PWM モード、フォルトピン無効 101 = 連続パルスモード; OCx ピンを LOW に初期化し、OCx ピンで連続的に出力パルスを発生させる 100 = 遅延ワンショット モード; OCx ピンを LOW に初期化し、OCx ピンで単発パルスを発生させる 011 = トグルモード; コンペアイベントは OCx ピンをトグルする 010 = アクティブHIGHワンショット モード; OCx ピンを HIGH に初期化し、コンペアイベントで OCx ピンを LOW に変更する 001 = アクティブLOWワンショット モード; OCx ピンを LOW に初期化し、コンペアイベントで OCx ピンを HIGH に変更する 000 = モジュール無効; 出力コンペア モジュールは無効

Note: 誤動作を防ぐために、ユーザ ソフトウェアは、出力コンペア制御レジスタへ書き込む際に、対応する出力コンペア モジュールを無効にする必要があります。

dsPIC33F ファミリ リファレンス マニュアル

レジスタ 13-2: OCxR: 出力コンペアレジスタ

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
コンペア値 <15:8>							
bit 15				bit 8			

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
コンペア値 <7:0>							
bit 7				bit 0			

凡例: HC = ハードウェア内でクリア
R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装、「0」として読み出し
-n = POR 時の値 「0」= ビットをクリア

bit 15-0 **コンペア値 <15:0>**

レジスタ 13-3: OCxRS: 出力コンペア セカンダリ レジスタ

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
セカンダリ コンペア値 <15:8>							
bit 15				bit 8			

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
セカンダリ コンペア値 <7:0>							
bit 7				bit 0			

凡例: HC = ハードウェア内でクリア
R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装、「0」として読み出し
-n = POR 時の値 「0」= ビットをクリア

bit 15-0 **セカンダリ コンペア値 <15:0>**

13.3 出力コンペアの動作

13.3.1 タイマの選択

各出力コンペア モジュールのタイムベースには Timer2 または Timer3 を選択できます。タイマの選択には、出力コンペア制御 (OCxCON<3>) レジスタ内の出力コンペアタイマ選択 (OCTSEL) ビットを使用します。

選択したタイマは、タイマ値をゼロからクロックごとに 1 ずつインクリメントします。タイマ値が周期レジスタ (PRV) 内の値に一致すると、タイマはタイマ値をゼロへリセットして再度インクリメントを開始します。タイマへのクロック供給には、内部クロック源 (Fosc/2) または TxCK ピンを介する同期外部クロック源を使用できます。

13.3.2 出力コンペアモード

出力コンペアモードの設定には、出力コンペア制御 (OCxCON<2:0>) レジスタ内の出力コンペアモード (OCM<2:0>) ビットを使用します。表 13-1 に、各種ビット設定に対応する出力コンペアモードを示します。図 13-2 に、各種出力コンペア モードの動作を示します。

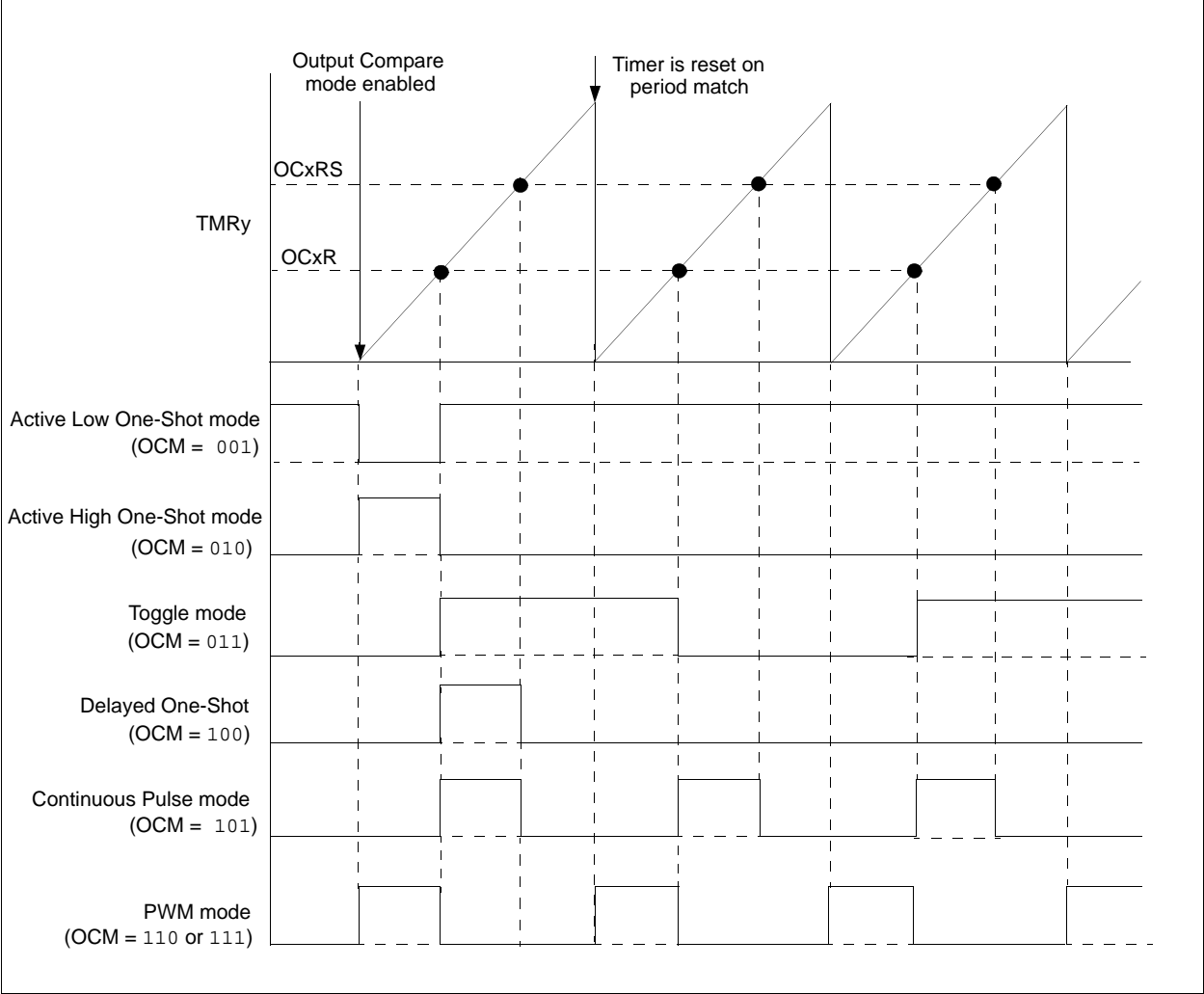
Note: 誤動作を防ぐために、ユーザ ソフトウェアは、出力コンペア制御レジスタへ書き込む際に、対応するタイマを無効にする必要があります。

表 13-1: 出力コンペアの動作モード

OCM<2:0>	モード	OCx ピンの初期状態	OCx 割り込み発生条件
000	モジュールは無効	GPIO レジスタによる制御	—
001	アクティブ LOW ワンショットモード	0	OCx 立ち上がりエッジ
010	アクティブ HIGH ワンショットモード	1	OCx 立ち下がりエッジ
011	トグルモード	直前の状態を維持	OCx 立ち上がりおよび立ち下がりエッジ
100	遅延ワンショットモード	0	OCx 立ち下がりエッジ
101	連続パルスモード	0	OCx 立ち下がりエッジ
110	フォルト保護なしの PWM モード	0、OCxR がゼロの場合 1、OCxR がゼロ以外の場合	割り込みは発生せず
111	フォルト保護ありの PWM モード	0、OCxR がゼロの場合 1、OCxR がゼロ以外の場合	OC1 ~ OC4 に対しては OCFA ⁽¹⁾ ピンの立ち下がりエッジ OC5 ~ OC8 に対しては OCFB ⁽¹⁾ ピンの立ち下がりエッジ

Note 1: OCFA および OCFB ピンは、出力コンペアのフォルト入力ピンです。OCFA ピンは出力コンペア モジュール 1 ~ 4 に対応します。OCFB ピンは出力コンペア モジュール 5 ~ 8 に対応します。

図 13-2: 出力コンペアの動作



13.3.2.1 アクティブ LOW ワンショット モード

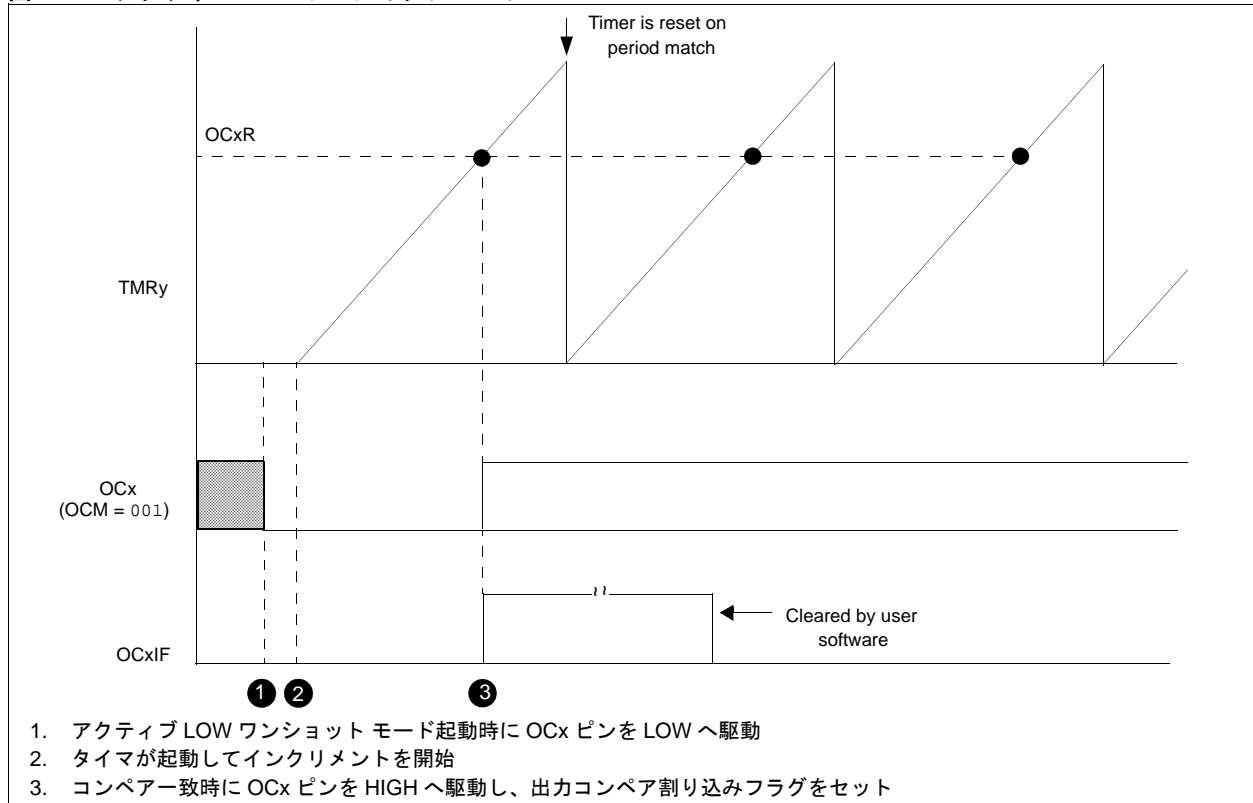
アクティブ LOW ワンショット モードは、単発のアクティブ LOW 出力パルスを生成します。パルスの期間は出力コンペア (OCxR) レジスタで指定します。ワンショット パルスを再度トリガするには、出力コンペア制御 (OCxCON) レジスタへ再度書き込む必要があります。

アクティブ LOW ワンショット モードの動作は以下の通りです。

1. 出力コンペア (OCx) ピンを即座に LOW へ駆動
2. タイマ値と出力コンペア (OCxR) レジスタ値の一致時に OCx ピンを HIGH へ駆動
3. OCx ピンの立ち上がりエッジで出力コンペア割り込みを生成

図 13-3 に、アクティブ LOW ワンショット モードの動作を示します。例 13-1 に、出力コンペア モジュールをアクティブ LOW ワンショット モードに設定するサンプルコードを示します。

図 13-3: アクティブ LOW ワンショット モード



例 13-1: 出力コンペアをアクティブ LOW ワンショット モードへ設定するサンプルコード

```
// Initialize Output Compare Module
OC1CONbits.OCM = 0b000; // Disable Output Compare Module
OC1CONbits.OCTSEL = 0; // Select Timer 2 as output compare time base
OC1R = 100; // Load the Compare Register Value
IPC0bits.OC1IP = 0x01; // Set Output Compare 1 Interrupt Priority Level
IFS0bits.OC1IF = 0; // Clear Output Compare 1 Interrupt Flag
IEC0bits.OC1IE = 1; // Enable Output Compare 1 interrupt
OC1CONbits.OCM = 0b001; // Select the Output Compare mode

// Initialize and enable Timer2

/* Example code for Output Compare 1 ISR*/
void __attribute__((__interrupt__)) _OC1Interrupt( void )
{
    /* Interrupt Service Routine code goes here */
    IFS0bits.OC1IF = 0; // Clear OC1 interrupt flag
}
```

13.3.2.2 アクティブ HIGH ワンショット モード

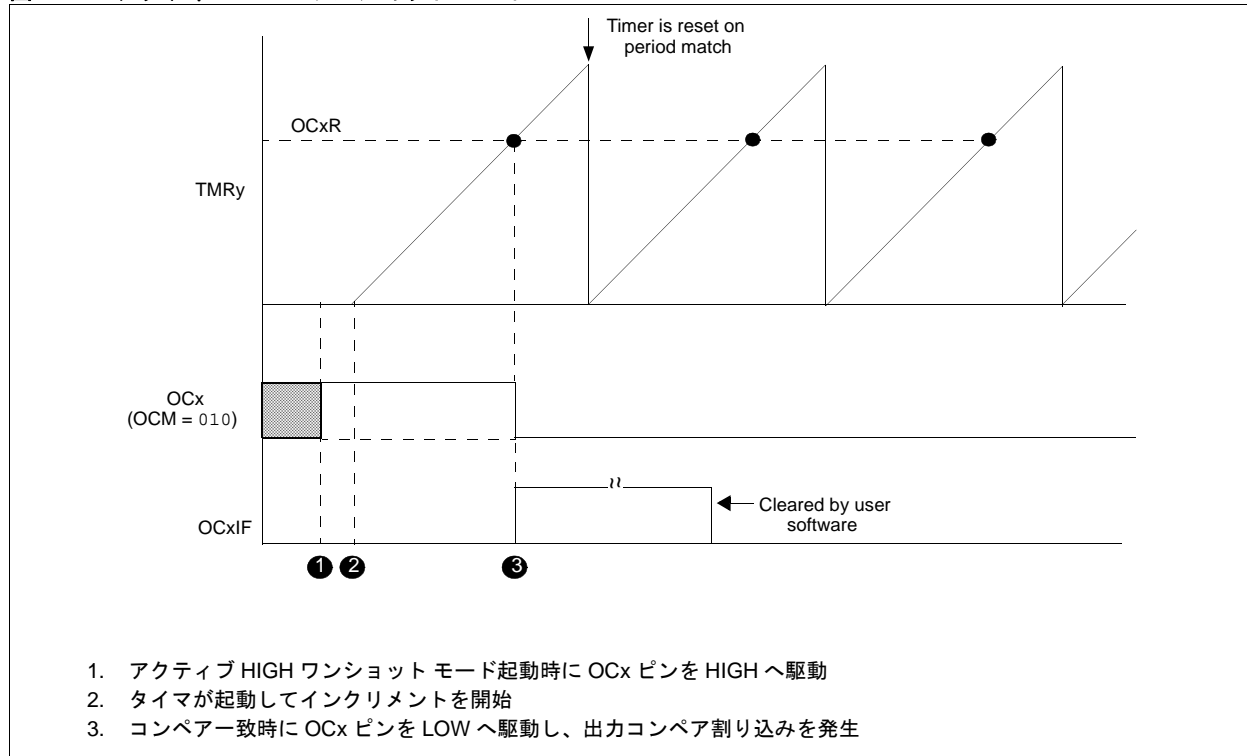
アクティブ HIGH ワンショット モードは、単発のアクティブ HIGH 出力パルスを生成します。パルスの期間は出力コンペア (OCxR) レジスタで指定します。ワンショット パルスを再度トリガするには、出力コンペア制御 (OCxCON) レジスタへ再度書き込む必要があります。

アクティブ HIGH ワンショット モードの動作は以下の通りです。

1. 出力コンペア (OCx) ピンを即座に HIGH へ駆動
2. タイマ値と出力コンペア (OCxR) レジスタ値の一致時に OCx ピンを LOW へ駆動
3. OCx ピンの立ち下がりエッジで出力コンペア割り込みを生成

図 13-4 に、アクティブ HIGH ワンショット モードの動作を示します。例 13-2 に、出力コンペア モジュールをアクティブ HIGH ワンショット モードに設定するサンプルコードを示します。

図 13-4: アクティブ HIGH ワンショット モード



例 13-2: 出力コンペアをアクティブ HIGH ワンショット モードへ設定するサンプルコード

```
// Initialize Output Compare Module
OC1CONbits.OCM = 0b000; // Disable Output Compare Module
OC1CONbits.OCTSEL = 0; // Select Timer 2 as output compare time base
OC1R = 100; // Load the Compare Register Value
IPC0bits.OC1IP = 0x01; // Set Output Compare 1 Interrupt Priority Level
IFS0bits.OC1IF = 0; // Clear Output Compare 1 Interrupt Flag
IEC0bits.OC1IE = 1; // Enable Output Compare 1 interrupt
OC1CONbits.OCM = 0b010; // Select the Output Compare mode

// Initialize and enable Timer2

/* Example code for Output Compare 1 ISR */
void __attribute__((__interrupt__)) _OC1Interrupt( void )
{
    /* Interrupt Service Routine code goes here */
    IFS0bits.OC1IF = 0; // Clear OC1 interrupt flag
}
```


13.3.2.3 トグルモード

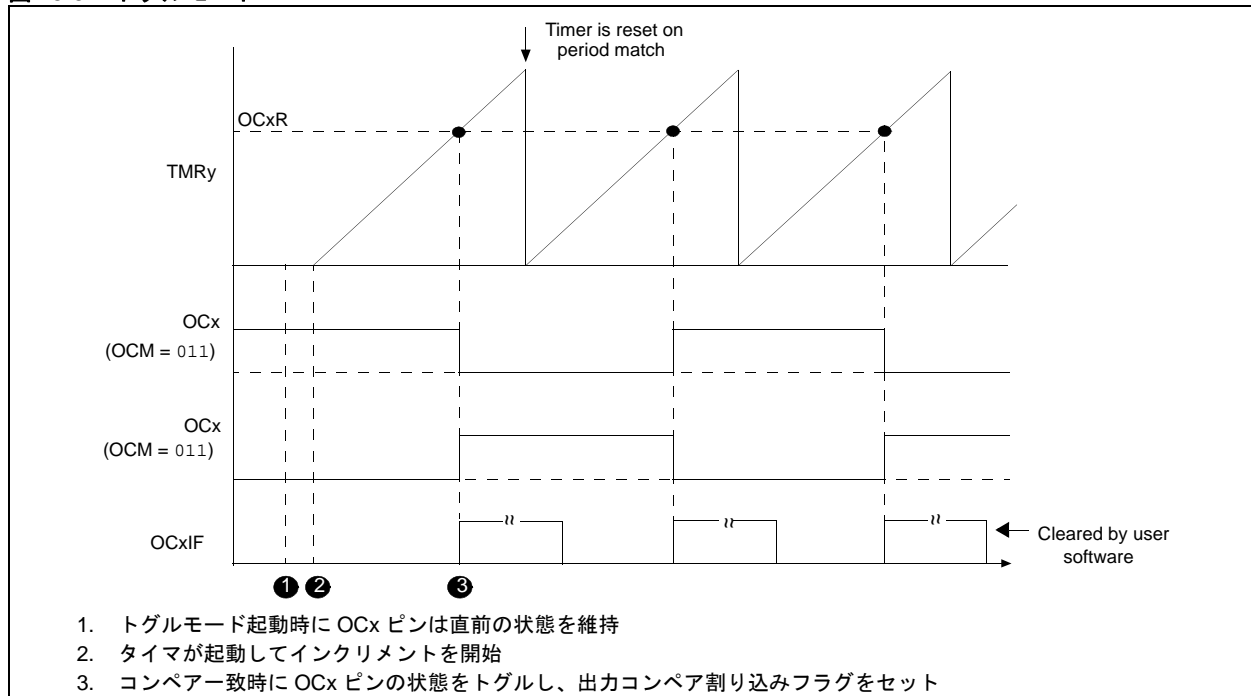
通常トグルモードは、50% デューティの矩形波を生成するために使用します。トグルモードは、出力コンペア モジュールからの内部信号を OCx ピンへ出力します。

- トグルモードを有効にする前にアクティブ HIGH ワンショット モードを選択する事により、トグルモードの初期状態を論理「1」に設定できます。
- トグルモードを有効にする前にアクティブ LOW ワンショット モードを選択する事により、トグルモードの初期状態を論理「0」に設定できます。

トグルモードは、タイマ値と出力コンペア (OCxR) レジスタ値が一致するたびに、OCx ピンの状態をトグルします。出力コンペア割り込みは、OCx ピンの立ち上がりおよび立ち下がりエッジの両方で発生します。

図 13-5 に、トグルモードの動作を示します。例 13-3 に、出力コンペア モジュールをトグルモードに設定するサンプルコードを示します。

図 13-5: トグルモード



例 13-3: 出力コンペア モジュールをトグルモードに設定するサンプルコード

```
// Initialize Output Compare Module
OC1CONbits.OCM = 0b000; // Disable Output Compare Module
OC1CONbits.OCM = 0b010; // Define Initial State for OC1 Pin (High if OCM=0b010)
OC1CONbits.OCM = 0b000; // Disable Output Compare Module
OC1CONbits.OCTSEL = 0; // Select Timer2 as output compare time base
OC1R = 100; // Load the Compare Register Value
IPC0bits.OC1IP = 0x01; // Set Output Compare 1 Interrupt Priority Level
IFS0bits.OC1IF = 0; // Clear Output Compare 1 Interrupt Flag
IEC0bits.OC1IE = 1; // Enable Output Compare 1 interrupt
OC1CONbits.OCM = 0b011; // Select the Output Compare mode

// Initialize and enable Timer2

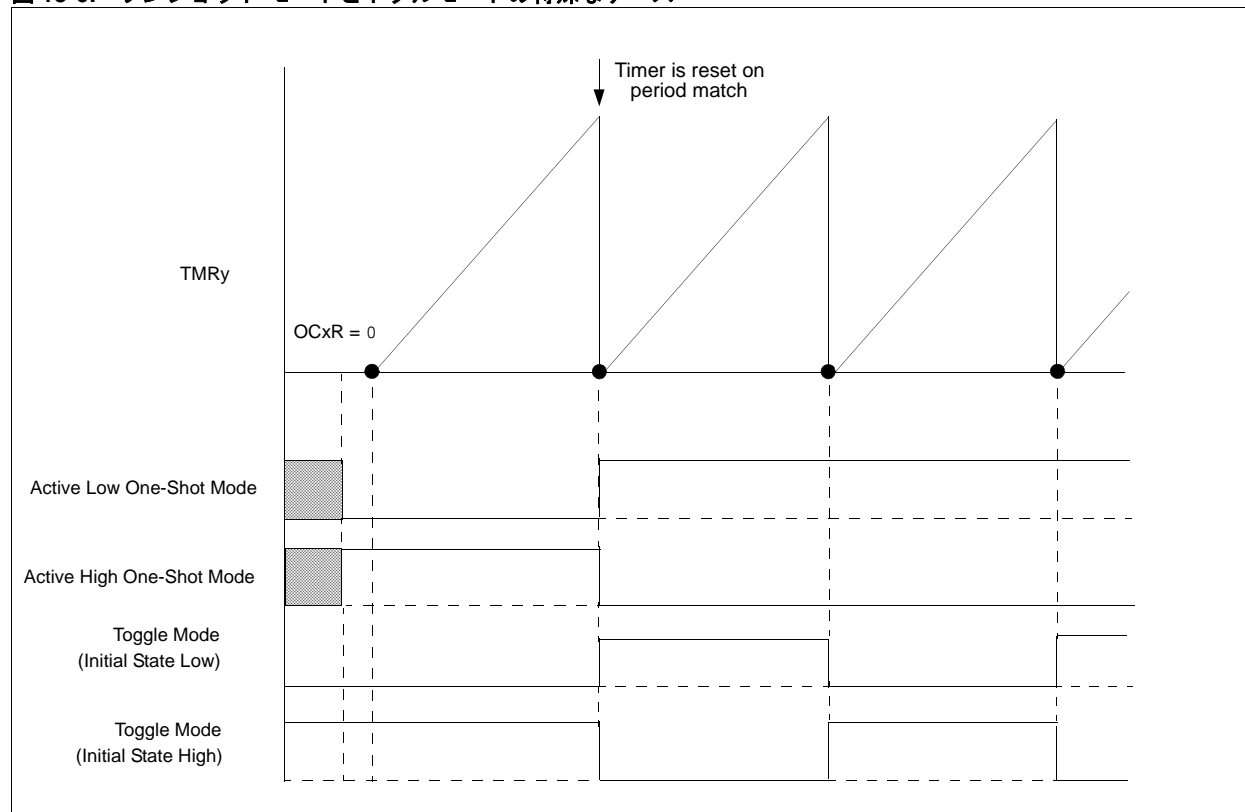
/* Example code for Output Compare 1 ISR */
void __attribute__((__interrupt__)) _OC1Interrupt( void )
{
    /* Interrupt Service Routine code goes here */
    IFS0bits.OC1IF = 0; // Clear OC1 interrupt flag
}
```

13.3.2.4 ワンショットモードとトグルモードの特殊なケース

出力コンペアレジスタの値がゼロ ($OCxR = 0$) の場合、コンペア一致は最初のタイマサイクルでは発生しません。このような場合の下記モードにおける出力コンペアの動作を図 13-6 に示します。

- アクティブ LOW ワンショットモード
- アクティブ HIGH ワンショットモード
- トグルモード

図 13-6: ワンショットモードとトグルモードの特殊なケース



13.3.2.5 遅延ワンショットモード

遅延ワンショットモードは、遅延したアクティブ HIGH の単発パルスを生成します。出力コンペア (OCxR) レジスタの値により、単発パルスの遅延量を制御します。パルス幅は出力コンペア セカンダリ (OCxRS) レジスタで指定します。ワンショット パルスを再度トリガするには、出力コンペア制御 (OCxCON) レジスタへ再度書き込む必要があります。

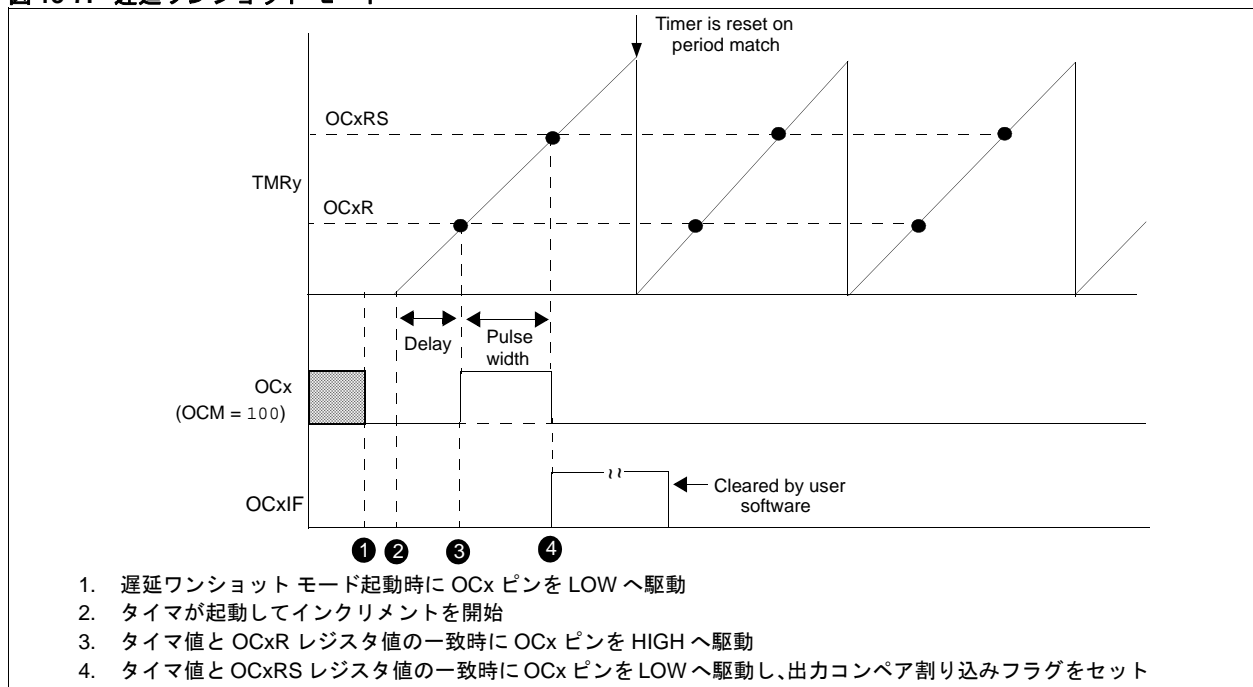
遅延ワンショット モードの動作は以下の通りです。

1. 出力コンペア (OCx) ピンを即座に LOW へ駆動
2. タイマ値と出力コンペア (OCxR) レジスタ値の一致時に OCx ピンを HIGH へ駆動
3. タイマ値と出力コンペア セカンダリ (OCxRS) レジスタ値の一致時に OCx ピンを LOW へ駆動
4. OCx ピンの立ち下がりエッジで出力コンペア割り込みを発生

出力コンペア モジュールは、再度トリガされるまで比較動作を停止します。

図 13-7 に遅延ワンショット モードの動作を示します。例 13-4 に、出力コンペア モジュールを遅延ワンショット モードに設定するサンプルコードを示します。

図 13-7: 遅延ワンショット モード



例 13-4: 出力コンペア モジュールを遅延ワンショット モードへ設定するサンプルコード

```
// Initialize Output Compare Module
OC1CONbits.OCM = 0b000; // Disable Output Compare Module
OC1CONbits.OCTSEL = 0; // Select Timer 2 as output compare time base
OC1R = 100; // Load the Compare Register Value for rising edge
OC1RS = 200; // Load the Compare Register Value for falling edge
IPC0bits.OC1IP = 0x01; // Set Output Compare 1 Interrupt Priority Level
IFS0bits.OC1IF = 0; // Clear Output Compare 1 Interrupt Flag
IEC0bits.OC1IE = 1; // Enable Output Compare 1 interrupt
OC1CONbits.OCM = 0b100; // Select the Output Compare mode

// Initialize and enable Timer2

/* Example code for Output Compare 1 ISR*/
void __attribute__((__interrupt__)) _OC1Interrupt( void )
{
    /* Interrupt Service Routine code goes here */
    IFS0bits.OC1IF = 0; // Clear OC1 interrupt flag
}
```

13.3.2.6 連続パルスモード

連続パルスモードは、各タイマサイクルで毎回パルスを出します。このモードは一定デューティサイクルの出力を生成する場合に便利です。連続パルスモードで動作する出力コンペア モジュールのブロック図を図 13-8 に示します。

連続パルスモードの動作は以下の通りです。

1. 出力コンペア (OCx) ピンを即座に LOW へ駆動
2. タイマ値と出力コンペア (OCxR) レジスタ値の一致時に OCx ピンを HIGH へ駆動
3. タイマ値と出力コンペア セカンダリ (OCxRS) レジスタ値の一致時に OCx ピンを LOW へ駆動
4. OCx ピンの立ち下がりエッジで出力コンペア割り込みを発生

図 13-9 に、連続パルスモードの動作を示します。

図 13-8: 連続パルスモードのブロック図

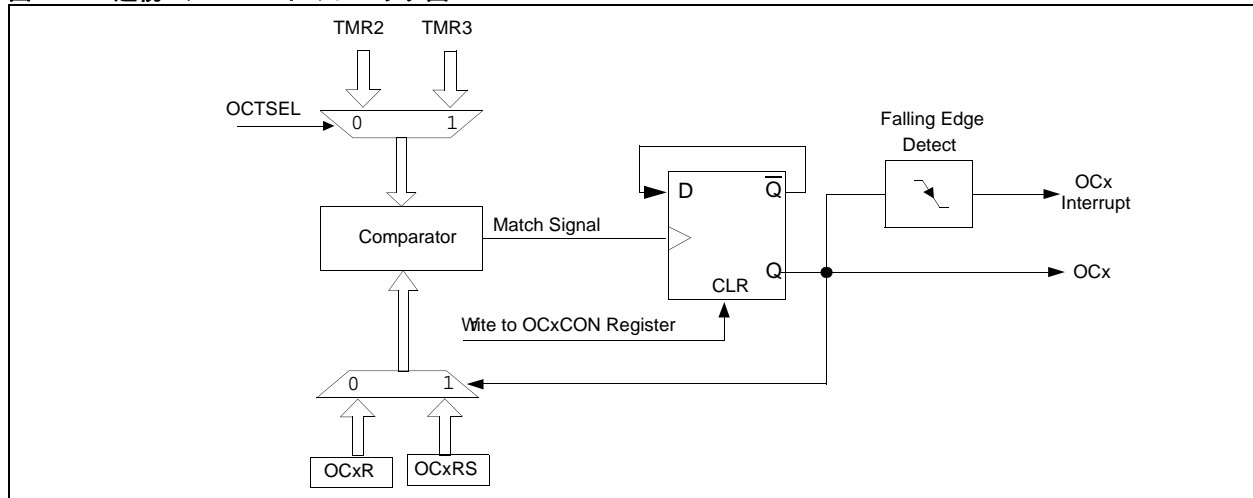
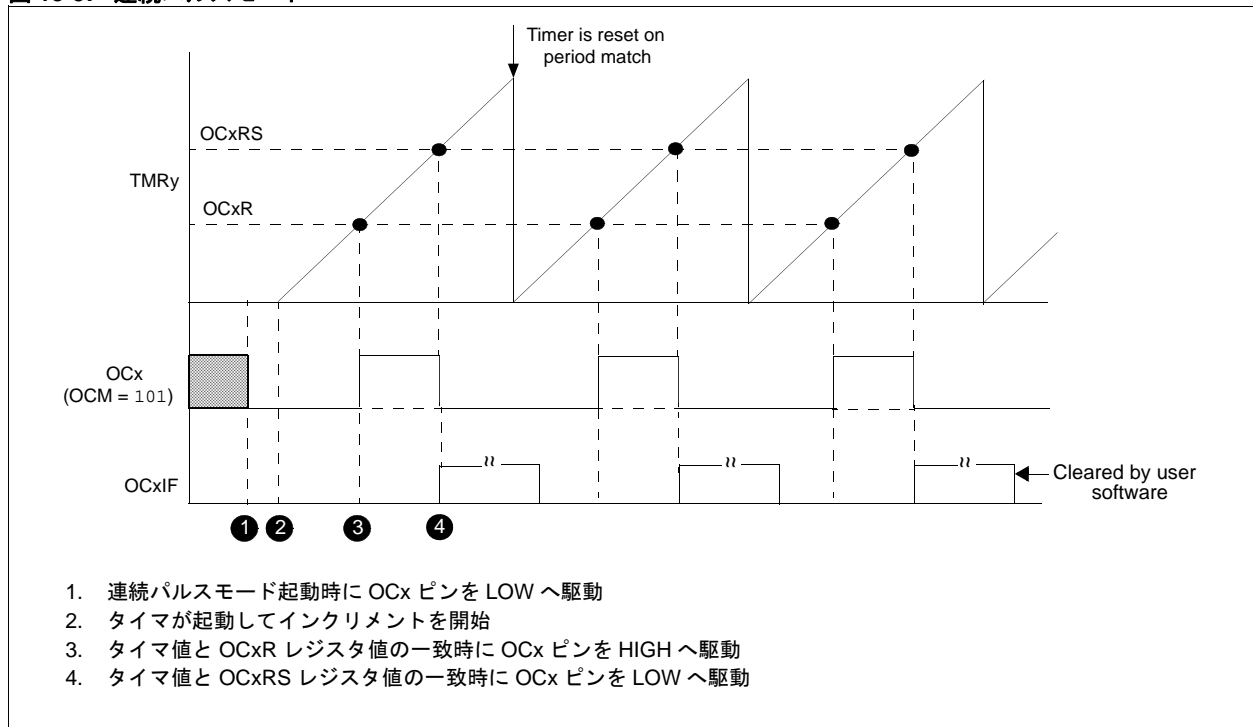


図 13-9: 連続パルスモード



例 13-4 に、出力コンペア モジュールを連続パルスモードに設定するサンプルコードを示します。

例 13-5: 出力コンペア モジュールを連続パルスモードに設定するサンプルコード

```
// Initialize Output Compare Module
OC1CONbits.OCM = 0b000; // Disable Output Compare Module
OC1CONbits.OCTSEL = 0; // Select Timer 2 as output compare time base
OC1R = 100; // Load the Compare Register Value for rising edge
OC1RS = 200; // Load the Compare Register Value for falling edge
IPC0bits.OC1IP = 0x01; // Set Output Compare 1 Interrupt Priority Level
IFS0bits.OC1IF = 0; // Clear Output Compare 1 Interrupt Flag
IEC0bits.OC1IE = 1; // Enable Output Compare 1 interrupt
OC1CONbits.OCM = 0b101; // Select the Output Compare mode

// Initialize and enable Timer2

/* Example code for Output Compare 1 ISR*/
void __attribute__((__interrupt__)) _OC1Interrupt( void )
{
    /* Interrupt Service Routine code goes here */
    IFS0bits.OC1IF = 0; // Clear OC1 interrupt flag
}
```

13.3.2.7 遅延ワンショット モードと連続パルスモードにおける特殊なケース

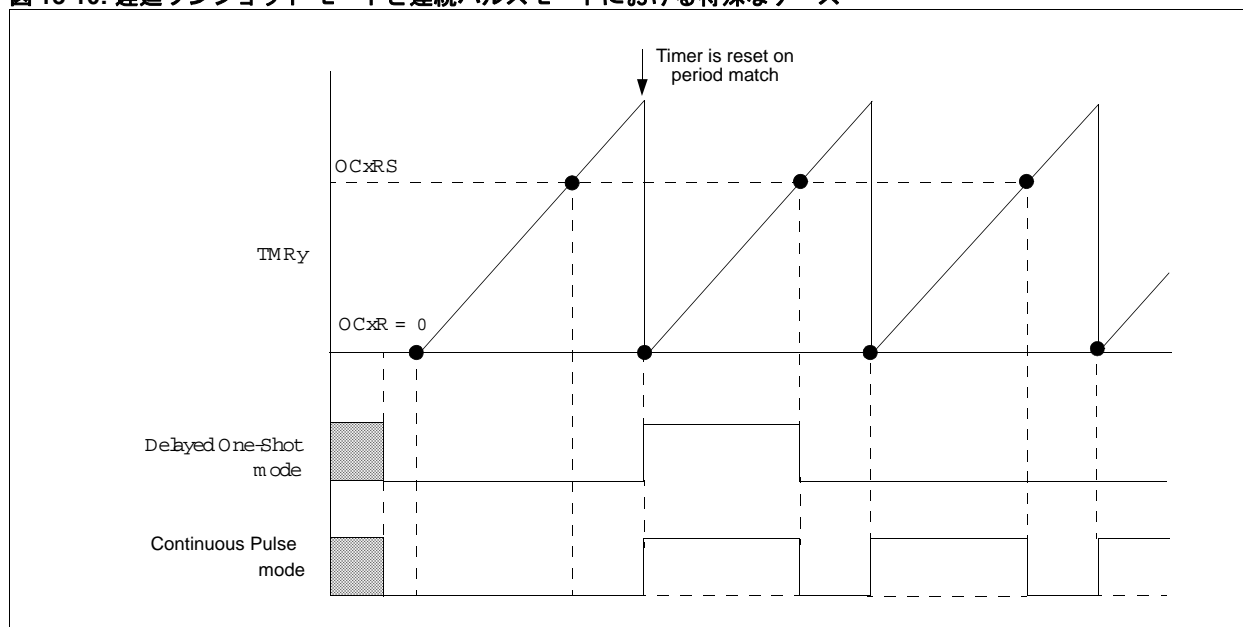
遅延ワンショット モードと連続パルスモードが正しく動作するには、OCxR、OCxRS、PRy 値が下記の関係を満たす必要があります。

- $OCxRS \geq OCxR$
- $PRy \geq OCxRS$

出力コンペアレジスタの値がゼロ ($OCxR = 0$) の場合、コンペアー一致は最初のタイマサイクルでは発生しません。このような場合の下記モードにおける出力コンペアの動作を図 13-6 に示します。

- 遅延ワンショットモード
- 連続パルスモード

図 13-10: 遅延ワンショット モードと連続パルスモードにおける特殊なケース



13.3.2.8 フォルト保護なしの PWM モード

PWM モードは、可変デューティ サイクル出力を生成する場合に使用します。PWM デューティ サイクルは、出力コンペア セカンダリ (OCxRS) レジスタで指定します。このレジスタは、出力コンペア (OCxR) レジスタに対するシャドーレジスタとして機能します。これにより、PWM 出力のグリッチを防ぎます。PWM モードでは、出力コンペア (OCxR) レジスタは読み出し専用です。PWM モードで動作する出力コンペア モジュールのブロック図を図 13-11 に示します。

PWM モード時の OCx ピンの動作は以下の通りです。

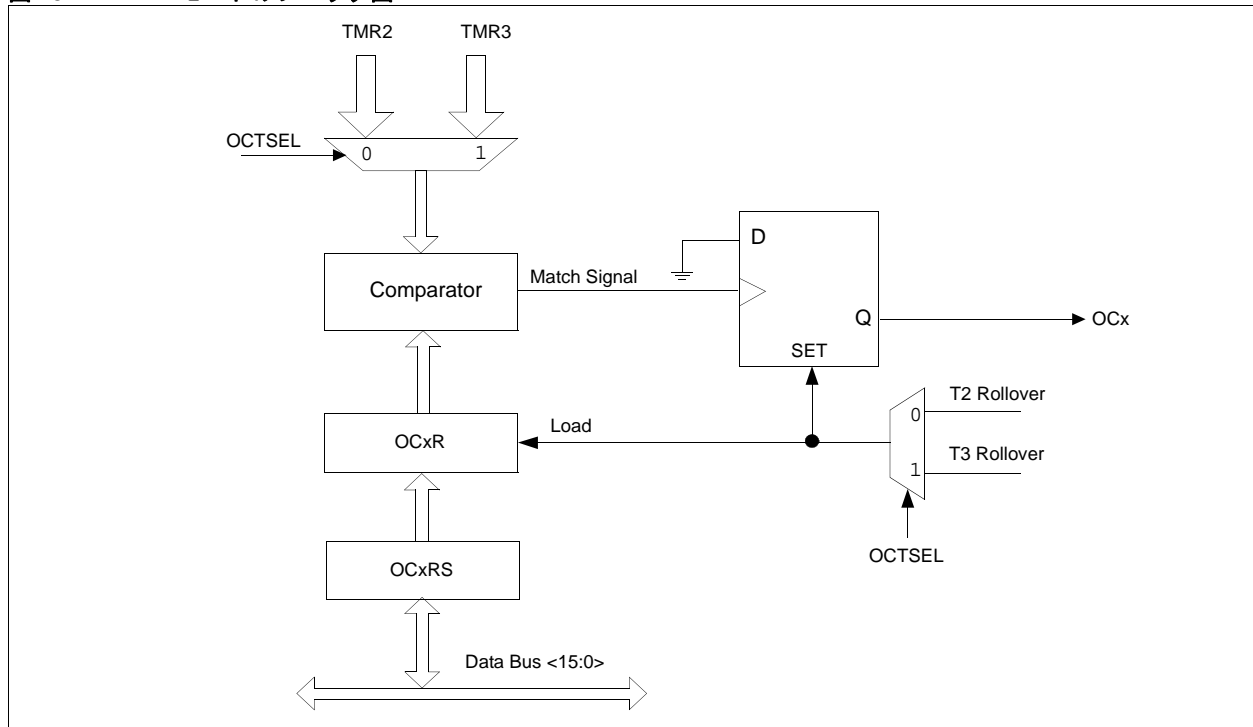
- OCxR レジスタ値がゼロ以外の場合に HIGH へ駆動 (図 13-12 の CASE 1 の場合)
- OCxR レジスタ値がゼロの場合に LOW へ駆動 (図 13-12 の CASE 2 の場合)

選択したタイマが起動し、周期レジスタの値に達するまでインクリメントします。このタイマ値とコンペアレジスタ (OCxR) 値を常時比較します。両値が一致すると OCx ピンを LOW へ駆動します。

タイマのロールオーバー時に OCxRS 値を OCxR レジスタへ転送し、OCx ピンを下記のように駆動します。

- OCxR レジスタ値がゼロ以外の場合は HIGH へ駆動
- OCxR レジスタ値がゼロの場合は LOW へ駆動

図 13-11: PWM モードのブロック図



13.3.2.8.1 PWM 周期

PWM 周期は、PRy (TMRy 周期レジスタ) への書き込みにより指定します。式 13-1 に PWM 周期の計算式を示します。

式 13-1: PWM 周期の計算

$$\text{PWM 周期} = [(PRy) + 1] \cdot Tcy \cdot (\text{TMRy プリスケール値})$$

$$\text{PWM 周波数} = 1/[\text{PWM 周期}]$$

Note: PRy 値が N の場合の PWM 周期は [N + 1] タイムベース サイクルです。例えば PRy レジスタに「7」を書き込んだ場合の周期は 8 タイムベース サイクルです。

13.3.2.8.2 PWM デューティ サイクル

PWM デューティ サイクルは OCxRS レジスタへ書き込む事により指定します。デューティ サイクル値は常時書き込み可能ですが、周期一致によるタイマリセットが発生するまで、その値は OCxR レジスタへラッチされません。これは PWM デューティ サイクルにダブルバッファを提供し、PWM 動作のグリッチを回避しています。PWM モードでは、OCxR レジスタは読み出し専用です。

PWM デューティ サイクルの重要境界パラメータとして下記が挙げられます。

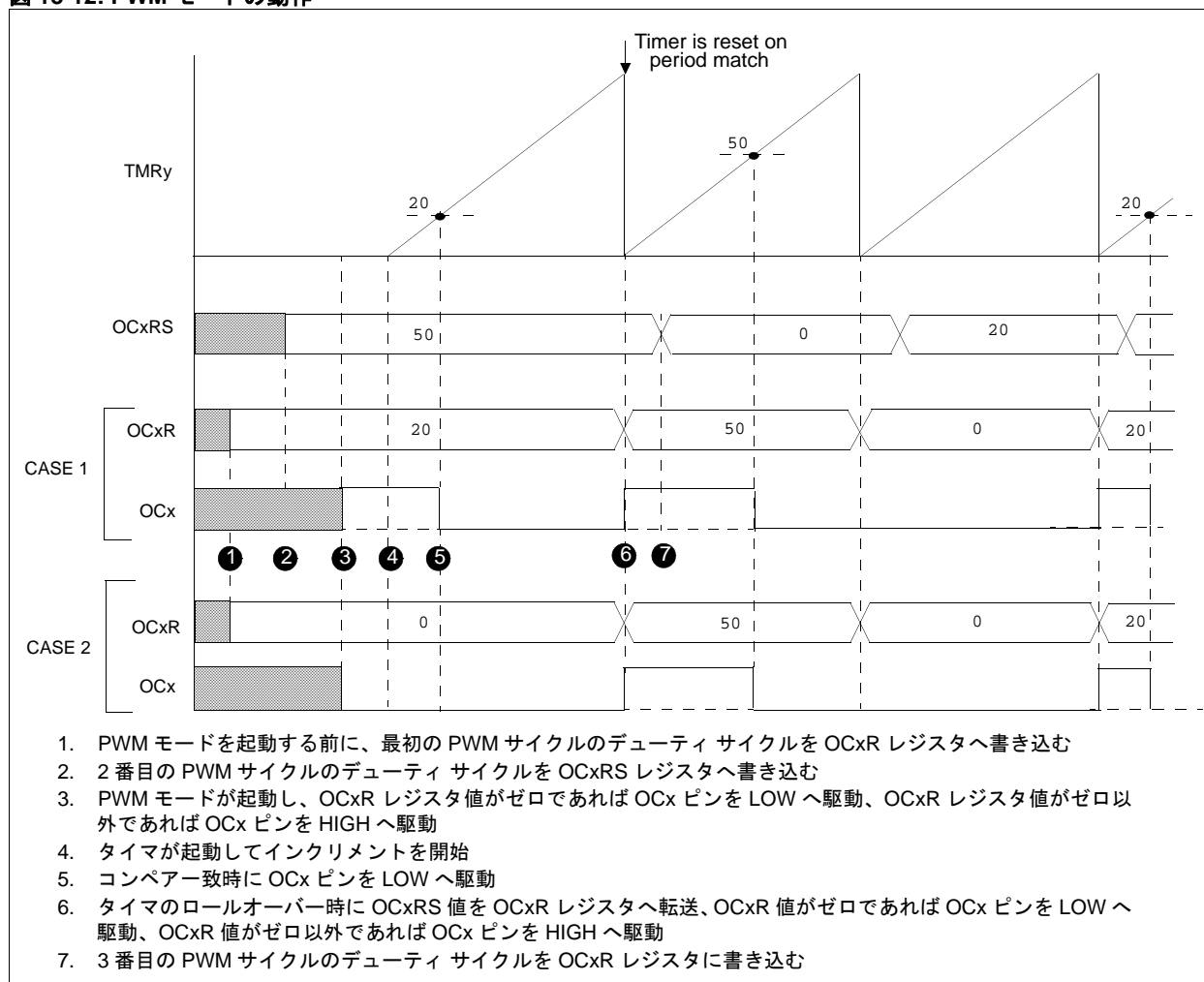
- デューティ サイクルレジスタ OCxR に 0000h を書き込むと、OCx ピンは LOW を維持します (0% デューティ サイクル)。
- OCxR レジスタ値が PRy (タイマ周期レジスタ) 値よりも大きい場合、OCx ピンは HIGH を維持します (100% デューティ サイクル)。
- OCxR レジスタ値が PRy 値に等しい場合、OCx ピンは 1 タイムベースカウント値だけ LOW ですが、後の全てのカウント値では HIGH です。

PWM 分解能は PWM 周波数とタイマクロック周波数によって決まります。タイマクロックは、内部クロック (FCY) をプログラマブル プリスケアラで分周して供給します。詳細はセクション 11.「タイマ」を参照してください。

式 13-2: PWM 分解能の計算式

$$\text{PWM 分解能 (bit)} = \log_2 \left(\frac{\text{PWM 周波数}}{\text{タイマクロック周波数}} \right) \text{ bit}$$

図 13-12: PWM モードの動作



例 13-6 に、出力コンペア モジュールを PWM モードに設定するサンプルコードを示します。

例 13-6: 出力コンペア モジュールを PWM モードに設定するサンプルコード

```
// Initialize Output Compare Module
OC1CONbits.OCM = 0b000; // Disable Output Compare Module
OC1R = 100; // Write the duty cycle for the first PWM pulse
OC1RS = 200; // Write the duty cycle for the second PWM pulse
OC1CONbits.OCTSEL = 0; // Select Timer 2 as output compare time base
OC1R = 100; // Load the Compare Register Value
OC1CONbits.OCM = 0b110; // Select the Output Compare mode

// Initialize and enable Timer2
T2CONbits.TON = 0; // Disable Timer
T2CONbits.TCS = 0; // Select internal instruction cycle clock
T2CONbits.TGATE = 0; // Disable Gated Timer mode
T2CONbits.TCKPS = 0b00; // Select 1:1 Prescaler
TMR2 = 0x00; // Clear timer register
PR2 = 500; // Load the period value

IPC1bits.T2IP = 0x01; // Set Timer 2 Interrupt Priority Level
IFS0bits.T2IF = 0; // Clear Timer 2 Interrupt Flag
IEC0bits.T2IE = 1; // Enable Timer 2 interrupt
T2CONbits.TON = 1; // Start Timer

/* Example code for Timer 2 ISR*/
void __attribute__((__interrupt__)) _T2Interrupt( void )
{
    /* Interrupt Service Routine code goes here */
    OC1RS = 300; // Write Duty Cycle value for next PWM cycle
    IFS0bits.T2IF = 0; // Clear Timer 2 interrupt flag
}
```

13

出力コンペア

13.3.2.9 フォルト保護ありの PWM モード

フォルト保護ありの PWM モードは、フォルト保護なしの PWM モードでの動作に加えて、フォルト条件で PWM 出力を 3 ステートにします。

フォルト保護には OCFA ピンと OCFB ピンを使用します。OCFA ピンは出力コンペア モジュール 1 ~ 4 に対応し、OCFB ピンは出力コンペア モジュール 5 ~ 8 に対応します。

OCFA または OCFB ピンで論理「0」を検出すると、選択した出力コンペアピンを 3 ステートにし、出力コンペアフォルト (OCFLT) フラグ (OCxCON<4>) をセットします。次いで出力コンペア割り込みフラグ (OCxIF) をセットして割り込みを発生します (その割り込みが有効な場合)。

外部フォルト条件を取り除いた後、適切なモードビット OCM<2:0> (OCxCON<2:0>) へ書き込んで PWM モードを再度有効にするまで、出力は 3 ステートにされ、OCxIF ビットはセットされた状態を維持します。

Note: ユーザ アプリケーションは、OCx ピンにプルダウンまたはプルアップ抵抗を追加する事により、フォルト条件発生時のピンの状態を選択できます。

13.4 DMA を使用する出力コンペア動作

dsPIC33F ファミリの一部のデバイスは、ダイレクトメモリ アクセス (DMA) モジュールを備えます。このモジュールを使用すると、データメモリから出コンペア モジュールへ CPU に負荷をかけずにデータを転送できます。デバイスが DMA を内蔵するかどうかは、各デバイスのデータシートを参照してください。詳細はセクション 22.「ダイレクト メモリアクセス (DMA)」を参照してください。

DMA チャンネルは下記のように初期化する必要があります。

- DMA チャンネル周辺モジュール アドレス (DMAxPAD) レジスタの値を、出力コンペア (OCxR) レジスタまたは出力コンペア セカンダリ (OCxRS) レジスタのアドレス値で初期化します。
- DMA 制御 (DMAxCON<13>) レジスタ内の転送方向 (DIR) ビットをセットします。この場合、デュアルポート DMA メモリからデータを読み出して、周辺モジュール特殊機能レジスタへ書き込みます。
- DMA 要求 (DMAxREQ<6:0>) レジスタ内の DMA 要求要因選択 (IRQSEL<6:0>) ビットで、DMA 転送要求要因を選択します。

例 13-7 に、CPU に負荷をかけずに PWM デューティ サイクルを変調するサンプルコードを示します。このコードでは、配列に保存したデューティ サイクル値をタイマ割り込み時に毎回 OCxRS レジスタへ転送します。

例 13-7: CPU の介入を受けずに PWM デューティ サイクルを変調するサンプルコード

```
// Initialize Output Compare Module in PWM mode
OC1CONbits.OCM = 0b000; // Disable Output Compare Module
OC1R=100; // Write the duty cycle for the first PWM pulse
OC1RS=200; // Write the duty cycle for the second PWM pulse
OC1CONbits.OCTSEL = 0; // Select Timer 2 as output compare time base
OC1R= 100; // Load the Compare Register Value
OC1CONbits.OCM = 0b110; // Select the Output Compare mode

// Initialize Timer2
T2CONbits.TON = 0; // Disable Timer
T2CONbits.TCS = 0; // Select internal instruction cycle clock
T2CONbits.TGATE = 0; // Disable Gated Timer mode
T2CONbits.TCKPS = 0b00; // Select 1:1 Prescaler
TMR2 = 0x00; // Clear timer register
PR2 = 500; // Load the period value

// Define a Buffer in DMA RAM to store duty cycle information
unsigned int BufferA[256] __attribute__((space(dma)));

// Setup and Enable DMA Channel
DMA0CONbits.AMODE = 0b00; // Register indirect with post increment
DMA0CONbits.MODE = 0b00; // Continuous, Ping-Pong mode Disabled
DMA0CONbits.DIR = 0; // Peripheral to RAM
DMAOPAD = (int)&OC1RS; // Address of the secondary output compare register
DMAOREQ = 7; // Select Timer2 interrupt as DMA request source
DMAOCNT = 255; // Number of words to buffer.
DMAOSTA = __builtin_dmaoffset(&BufferA);

IFS0bits.DMA0IF = 0; // Clear the DMA interrupt flag
IEC0bits.DMA0IE = 1; // Enable DMA interrupt
DMA0CONbits.CHEN = 1; // Enable DMA channel

// Enable Timer
T2CONbits.TON = 1; // Start Timer

// DMA Interrupt Handler
void __attribute__((__interrupt__)) _DMA0Interrupt(void)
{
    IFS0bits.DMA0IF = 0; // Clear the DMA0 Interrupt Flag
}
```

13.5 省電力モード時の出力コンペアの動作

13.5.1 スリープモード時の出力コンペア動作

出力コンペア モジュールは、デバイスのスリープモード時に動作しません。OCx ピンは既定初期状態を維持します。各種出力コンペア モード向けの初期状態は表 13-1 に記載しています。

出力コンペア モジュールをフォルト保護ありの PWM モード向けに設定した場合、スリープモード時のフォルト条件は出力を 3 ステートにします。フォルトを検出すると、OCx ピンを 3 ステートにし、OCFLT ビット (OCxCON<4>) をセットします。フォルト発生時の割り込みは保留され、デバイスのウェイクアップ時に割り込みが発生します。

13.5.2 アイドルモード時の出力コンペア動作

デバイスがアイドルモードへ切り換わってもシステムクロック源は動作し続けますが、CPU はコード実行を停止します。アイドルモード時に出力コンペア モジュールが停止するか動作を維持するかは、アイドル時停止 (OCSIDL) ビット (OCxCON<13>) によって決まります。

- OCSIDL = 0 の場合、出力コンペア モジュールはアイドルモード中に動作を続け、完全に機能します。選択したタイマも、アイドルモード時の動作を有効にしておく必要があります。
- PSIDL = 1 の場合、モジュールはアイドルモード時に停止します。この場合、モジュールはスリープモード時と同様に機能します。**セクション 13.5.1 「スリープモード時の出力コンペア動作」**を参照してください。

13.6 I/O ピンの制御

出力コンペア モジュールを有効にした場合、I/O ピンの入出力方向はコンペア モジュールが制御します。コンペア モジュールを無効にした場合、I/O ピンの制御は対応する LAT および TRIS 制御ビットに従います。

「フォルト保護あり」の単純 PWM 入力モードを有効にする場合、対応する TRIS ビットをセットする事により、OCFx フォルトピンを入力向けに設定する必要があります。「フォルト保護あり」PWM モードを有効にするだけでは、OCFx フォルトピンは入力として設定されません。

Note: 一部のデバイスでは、対応する TRIS ビットをクリアする事により、出力コンペアピンの方向ビットを出力向けに設定する必要があります。詳細は各デバイスのデータシートを参照してください。

13.7 レジスタマップ

dsPIC33F の出力コンペア モジュールに関連するレジスタの概要を表 13-2、表 13-3、表 13-4 に示します。

表 13-2: 出力コンペア関連のレジスタマップ

SFR 名	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	全リセット
OC1RS	出力コンペア 1 セカンダリ レジスタ																xxxx
OC1R	出力コンペア 1 レジスタ																xxxx
OC1CON	—	—	OCSIDL	—	—	—	—	—	—	—	—	OCFLT	OCTSEL	OCM<2:0>		0000	
OC2RS	出力コンペア 2 セカンダリ レジスタ																xxxx
OC2R	出力コンペア 2 レジスタ																xxxx
OC2CON	—	—	OCSIDL	—	—	—	—	—	—	—	—	OCFLT	OCTSEL	OCM<2:0>		0000	
OC3RS	出力コンペア 3 セカンダリ レジスタ																xxxx
OC3R	出力コンペア 3 レジスタ																xxxx
OC3CON	—	—	OCSIDL	—	—	—	—	—	—	—	—	OCFLT	OCTSEL	OCM<2:0>		0000	
OC4RS	出力コンペア 4 セカンダリ レジスタ																xxxx
OC4R	出力コンペア 4 レジスタ																xxxx
OC4CON	—	—	OCSIDL	—	—	—	—	—	—	—	—	OCFLT	OCTSEL	OCM<2:0>		0000	
OC5RS	出力コンペア 5 セカンダリ レジスタ																xxxx
OC5R	出力コンペア 5 レジスタ																xxxx
OC5CON	—	—	OCSIDL	—	—	—	—	—	—	—	—	OCFLT	OCTSEL	OCM<2:0>		0000	
OC6RS	出力コンペア 6 セカンダリ レジスタ																xxxx
OC6R	出力コンペア 6 レジスタ																xxxx
OC6CON	—	—	OCSIDL	—	—	—	—	—	—	—	—	OCFLT	OCTSEL	OCM<2:0>		0000	
OC7RS	出力コンペア 7 セカンダリ レジスタ																xxxx
OC7R	出力コンペア 7 レジスタ																xxxx
OC7CON	—	—	OCSIDL	—	—	—	—	—	—	—	—	OCFLT	OCTSEL	OCM<2:0>		0000	
OC8RS	出力コンペア 8 セカンダリ レジスタ																xxxx
OC8R	出力コンペア 8 レジスタ																xxxx
OC8CON	—	—	OCSIDL	—	—	—	—	—	—	—	—	OCFLT	OCTSEL	OCM<2:0>		0000	

凡例: x = リセット時に未知の値、— = 未実装、「0」として読み出し、リセット値は 16 進数で表記

表 13-3: タイマ関連のレジスタマップ

SFR 名	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	全リセット
TMR2	Timer2 レジスタ																xxxx
TMR3HLD	Timer3 ホールドレジスタ (32 ビットタイマ動作専用)																xxxx
TMR3	Timer3 レジスタ																xxxx
PR2	周期レジスタ 2																FFFF
PR3	周期レジスタ 3																FFFF
T2CON	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS<1:0>	T32	—	TCS	—	0000	
T3CON	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS<1:0>	—	—	TCS	—	0000	

凡例: x = リセット時に未知の値、— = 未実装、「0」として読み出し、リセット値は 16 進数で表記

表 13-4: 割り込みコントローラ関連のレジスタマップ

SFR 名	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	全リセット
IFS0	—	—	—	—	—	—	—	—	—	OC2IF	—	—	—	OC1IF	—	—	0000
IFS1	—	—	—	—	—	OC4IF	OC3IF	—	—	—	—	—	—	—	—	—	0000
IFS2	—	—	—	OC8IF	OC7IF	OC6IF	OC5IF	—	—	—	—	—	—	—	—	—	0000
IEC0	—	—	—	—	—	—	—	—	—	OC2IE	—	—	—	OC1IE	—	—	0000
IEC1	—	—	—	—	—	OC4IE	OC3IE	—	—	—	—	—	—	—	—	—	0000
IEC2	—	—	—	OC8IE	OC7IE	OC6IE	OC5IE	—	—	—	—	—	—	—	—	—	0000
IPC0	—	—	—	—	—	OC1IP<2:0>		—	—	—	—	—	—	—	—	—	4444
IPC1	—	—	—	—	—	OC2IP<2:0>		—	—	—	—	—	—	—	—	—	4444
IPC5	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	4444
IPC6	—	—	—	—	—	OC4IP<2:0>		—	OC3IP<2:0>		—	—	—	—	—	—	4444
IPC10	—	OC7IP<2:0>		—	OC6IP<2:0>		—	OC5IP<2:0>		—	—	—	—	—	—	—	4444
IPC11	—	—	—	—	—	—	—	—	—	—	—	—	—	OC8IP<2:0>		—	4444

凡例: x = リセット時に未知の値、— = 未実装、「0」として読み出し、リセット値は 16 進数で表記

13.8 設計のヒント

質問 1: OCSIDL ビット (OCxCON<13>) をセットしていないのに出力コンペアピンが機能を停止します。なぜですか。

回答: 原因としては、対応するタイマ源の TSIDL ビット (TxCON<13>) がセットされている事が考えられます。この状態で PWRSV 命令を実行すると、タイマがアイドルモードに切り換わってしまいます。

質問 2: 32 ビットモードに設定したタイムベースを出力コンペア モジュール向けに使用できますか。

回答: いいえ、使用できません。タイマを出力コンペア モジュール向けに使用する場合、T32 ビット (TxCON<3>) をクリアする必要があります。

13.9 関連アプリケーション ノート

本セクションに関連するアプリケーション ノートの一覧を下に記載します。一部のアプリケーション ノートは dsPIC33F 製品ファミリ向けではありません。ただし概念は共通しており、変更が必要であったり制限事項が存在するものの利用が可能です。出力コンペア モジュールに関連する最新のアプリケーション ノートは以下の通りです。

タイトルアプリケーション ノート番号

環境監視向け I ² C™ ネットワーク プロトコル	AN736
CCP モジュールの使用	AN594
PIC16C924 を使用したクロック アプリケーション	AN649
PWM を使用したアナログ出力の生成	AN538
PIC16F684 によるブラシ付き双方向 DC モータの低コストな制御	AN893
PIC18 マイクロコントローラによる三相誘導モータの速度制御	AN843

Note: dsPIC33F デバイスファミリ関連のアプリケーション ノートとサンプルコードは、マイクロチップ社のウェブサイト (www.microchip.com) でご覧になれます。

13.10 改訂履歴

リビジョン A (2007 年 5 月)

本書の初版

ISBN: 978-1-60932-508-4