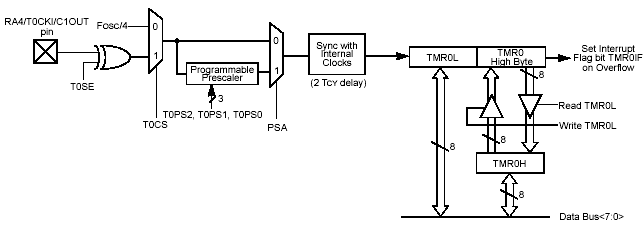
**タイマー０，１，２，３の使い方**

**【タイマー０】**

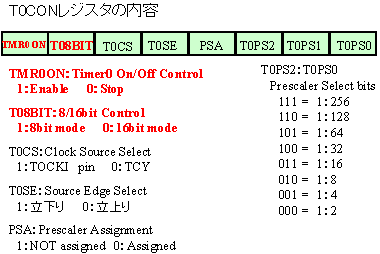
　タイマー０もPIC18シリーズでは強化されています。強化ポイントは下記のように  
なっています。  
  
(1) ８ビットモードと１６ビットモードの両方が用意されソフトウェアで選択できる。  
　　カウンタが長くなったことで、長時間のインターバルタイマも可能になった。  
(2) ８ビットのプリスケーらが専用に用意された。  
　　従来はウォッチドッグタイマと兼用であったが、専用になった。  
(3) １６ビットタイマを確実に読めるようになった。  
　　１６ビットモードの時にも上位バイト、下位バイトを確実に読み出せるように、  
　　ハードウェアで上位バイトをバッファするようにした。

**【タイマー０の構成】**

タイマー０の１６ビットモードの時の構成は下図のようになります。

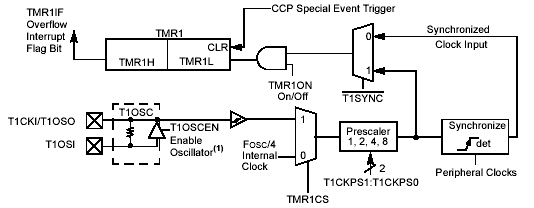
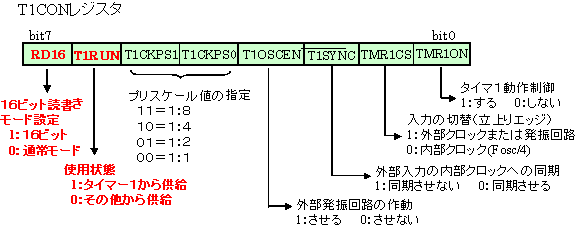


ここで、タイマー０のカウンタの内容は上位、下位の２バイトでRead/Writeします。  
このとき上位バイトは直接は読み書きしないで、TMR0Hレジスタを経由して読み  
書きします。  
この読み書きのタイミングは下位バイトであるTMR0Lレジスタと同期するように  
なっており、読み出しの時には、TMR0Lを読むと、同時に上位バイトがTMR0H  
レジスタにコピーされます。  
このため、従来２度読みしないと下位バイトを読んで、次に上位バイトを読むまで  
の間で、上位バイトがカウントアップしてしまうかもしれないという問題を回避でき  
るようになりました。  
つまりTMR0Lを読み込んでから、次にTMR0Hを読めば、TMR0Lを読み込んだ瞬間  
の時の上位バイトが読み込めることになります。  
  
書込みの時は、あらかじめ上位データをTMR0Hに書き込んでおき、それから下位  
データをTMR0Lに書き込めば、そのタイミングで同時にTMR0Hからカウンタの上位  
バイトに書き込まれます。  
  
  
**《T0CONレジスタの構成》**  
　タイマー０の機能強化により制御用のレジスタT0CONも機能追加されて下図の  
ようになりました。



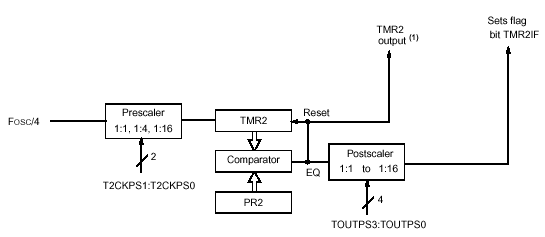
ここでは、TMR0ONとT08BITビットが追加されました。  
T08BITが８ビットと１６ビットのモード切替用の設定ビットです。

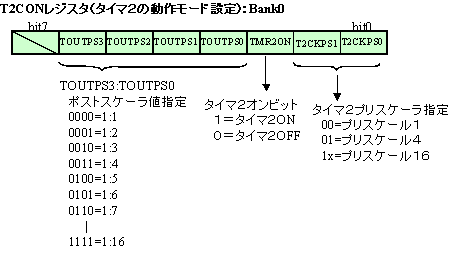
**【タイマー１】**

タイマー１も強化されています。  
(1) 16ビットのカウンタの読み出し、書込みがバッファ経由となって、１回で確実に  
　　読み書きできるようになった。  
  
(2) タイマー１の構成が、下図のように改良されて、上位バイトの扱いがTMR1Hという  
　　バッファ経由となりました。  
　　これで、従来２度読みしたり、書込みには一度停止させたりしなければならな  
　　かったのが、確実に読み書きできるようになりました。  
  
  
  
  
まず読み出しは下位バイトの、TMR1Lを読み出すと、上位バイトが同時にTMR1Hに  
コピーされます。これで後からTMR1Hを読み込めば、下位を読み込んだ瞬間の  
カウンタ値が正確に読み込めます。  
書込み時は、TMR1Hに先に上位データを書き込んでおき、あとから下位データを  
TMR1Lに書き込めば、そのタイミングで同時に上位バイトがTMR1Hからカウンタに  
コピーされます。これで、下位データを書き込む瞬間に１６ビット同時に書き替わる  
ことになります。  
  
この１６ビットモードの追加に伴いT1CONレジスタに１６ビットモード設定用のビット、  
RD16ビットが追加されました。  
  


**【タイマー２】**

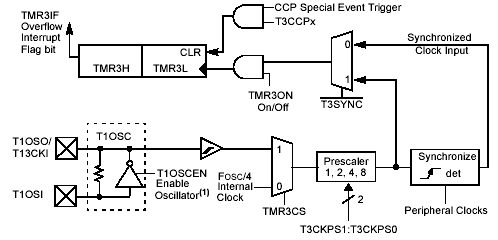
特にこれまでと変更強化部分はありません。  
タイマー２の構成は下図のようになっています。従来同様タイマー２は主に  
CCPのデューティ制御用カウンタとして使われます。

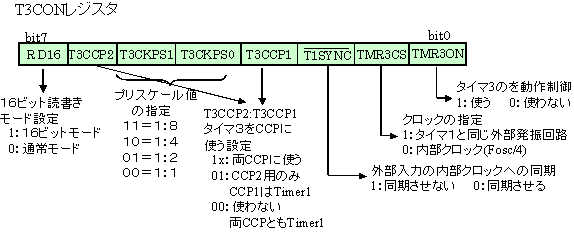


タイマ２の制御ようには下図のT2CONレジスタが使われます。このレジスタは  
PIC16シリーズから特に変更はありません。  
  


**【タイマー３】**

タイマー３はPIC18シリーズで新たに追加されたタイマでCCP用に使えるタイマとして  
強化されました。  
  
(1) タイマー１と同様に８ビットモードと１６ビットモードがあります。  
　　同様に１６ビットモード設定用のビット、RD16がT3CONレジスタにあります。  
  
(2) タイマ－１の発振回路のクロックをタイマー３のクロックとすることが出来るよう  
　　になっています。これによりリアルタイムクロックを２つにすることが可能となります。  
  
(3) タイマー３をタイマー１と同じようにCCPの周期カウンタとして使うことが出来る  
　　ようになっており、いずれか片方を指定して使うことになります。  
  
タイマー３のブロック構成は下図のようになっています。  
図のように発振回路はタイマー１と兼用になっていて、TMR3CSビットにより指定  
して選択します。  
１６ビットのタイマーカウンタは、やはりTMR3Hレジスタでバッファリングされていて  
TMR3Lのリードライトと連動してTMR3Hとタイマの上位８ビットとの間でデータ授受  
を実行します。



タイマ３の制御には下図のT3CONレジスタが使われますが、これは新たに  
追加されたレジスタで、特にタイマ３はCCP用のクロック用として、タイマ１と  
２つが用意されたので、どちらをCCP用に使うかの指定が出来るようになって  
います。  
  


**【タイマー用Cライブラリ関数】**

MPLAB-C18で用意されているタイマー用の関数ライブラリがあります。  
各タイマー毎に用意されていて、タイマー０，１，２，３の数値で区別しています。

**(1) タイマー用C関数ライブラリ一覧**

|  |  |  |
| --- | --- | --- |
| 関数名 | 機能と基本形 | 使用例 |
| CloseTimer0 CloseTimer1 CloseTimer2 CloseTimer3 | 指定タイマーを停止し割込みを禁止する | CloseTimer0(); |
| void CloseTimer0(void); |
| OpenTimer0 OpenTimer1 OpenTimer2 OpenTimer3 | 指定のタイマーの初期設定を行う。割込み、 クロック選択、プリスケーラ値などの設定を含む | OpenTimer0(TIMER\_INT\_OFF & T0\_SOURCE\_INT &  T0\_PS\_1\_32); |
| void OpenTimer0(unsigned char config); |
| ReadTimer0 ReadTimer1 ReadTimer2 ReadTimer3 | 各タイマーのカウンタレジスタの内容を返す ８ビットと１６ビットのものがある 　　Timer0：　int(16-bits)　　TMR0H、TMR0Lレジスタ 　　Timer1：　int(16-bits)　　TMR1H、TMR1Lレジスタ 　　Timer2：　char(8-bits)　　TMR2レジスタ 　　Timer3：　int(16-bits)　　TMR3H、TMR3Lレジスタ | unsigned int result; result = ReadTimer0(); |
| unsigned int ReadTimer1(void); unsigned char ReadTimer2(void); |
| WriteTimer0 WriteTimer1 WriteTimer2 WriteTimer3 | 各タイマのタイマレジスタに値を設定する ８ビットと１６ビットのものがある 　　Timer0：　int(16-bits)　　TMR0H、TMR0Lレジスタ 　　Timer1：　int(16-bits)　　TMR1H、TMR1Lレジスタ 　　Timer2：　char(8-bits)　　TMR2レジスタ 　　Timer3：　int(16-bits)　　TMR3H、TMR3Lレジスタ | WriteTimer0(0); |
| unsigned WriteTimer0(unsigned int timer); unsigned WriteTimer2(unsigned char timer); |

**(2) OpenTimer用パラメータ一覧**  
　　OpenTimer関数には各タイマの動作モードを設定するための多くのパラメータが  
　　必要となりますが、これらは下記のようになっています。

|  |  |  |  |
| --- | --- | --- | --- |
| タイマー種別 | 項目種別 | パラメータ名称 | 意味内容 |
| 全部共通 | 割込み許可禁止 | TIMER\_INT\_ON TIMER\_INT\_OFF | タイマー割込み許可 　　　　　　　　　禁止 |
| タイマ０ | モード設定 | T0\_8BIT T0\_16BIT | ８ビットモード指定 １６ビットモード指定 |
| 外部クロックエッジ | T0\_EDGE\_FALL T0\_EDGE\_RISE | 外部クロック立下り 外部クロック立上り |
| クロック選択 | T0\_SOURCE\_EXT T0\_SOURCE\_INT | 外部クロック指定（I/Oピン) 内部クロック指定(Tosc) |
| プリスケーラ値 | T0\_PS1\_1 T0\_PS1\_2 T0\_PS1\_4 T0\_PS1\_8 T0\_PS1\_16 T0\_PS1\_32 T0\_PS1\_64 T0\_PS1\_128 T0\_PS1\_256 | １：１ １：２ １：４ １：８ １：１６ １：３２ １：６４ １：１２８ １：２５６ |
| タイマ１ | モード設定 | T1\_8BIT\_RW T1\_16BIT\_RW | ８ビットモード指定 １６ビットモード指定 |
| クロック選択 | T1\_SOURCE\_EXT T1\_SOURCE\_INT | 外部クロック指定（I/Oピン) 内部クロック指定(Tosc) |
| プリスケーラ値 | T1\_PS\_1\_1 T1\_PS\_1\_2 T1\_PS\_1\_4 T1\_PS\_1\_8 | １：１ １：２ １：４ １：８ |
| 発振回路許可禁止 | T1\_OSC1EN\_ON T1\_OSC1EN\_OFF | 発振回路を使用する 発振回路を使用しない |
| 外部クロック入力 の内部同期の指定 | T1\_SYNC\_EXT\_ON T1\_SYNC\_EXT\_OFF | 内部クロックに同期させる 　　　　　　　　　させない |
| CCP用のクロック指定 | T1\_SOURCE\_CCP T1\_CCP1\_T3\_CCP2 | CCP1,2両方のクロックに使用 CCP１だけのクロックに使用 |
| タイマ２ | プリスケーラ値 | T2\_PS1\_1 T2\_PS1\_4 T2\_PS1\_16 | １：１ １：４ １：１６ |
| ポストスケーラ値 | T2\_POST\_1\_1 T2\_POST\_1\_2 | １：１ １：２ |
| タイマ３ | クロック選択 | T3\_SOURCE\_EXT T3\_SOURCE\_INT | 外部クロック指定（I/Oピン) 内部クロック指定(Tosc) |
| モード設定 | T3\_8BIT\_RW T3\_16BIT\_RW | ８ビットモード指定 １６ビットモード指定 |
| プリスケーラ値 | T3\_PS\_1\_1 T3\_PS\_1\_2 T3\_PS\_1\_4 T3\_PS\_1\_8 | １：１ １：２ １：４ １：８ |
| クロックの選択 | T3\_OSC1EN\_ON T3\_OSC1EN\_OFF | タイマー１のクロックを使う 　　　　　　　　　使わない |
| 外部クロック入力 の内部同期の指定 | T3\_SYNC\_EXT\_ON T3\_SYNC\_EXT\_OFF | 内部クロックに同期させる 　　　　　　　　　させない |
| CCP用のクロック指定 | T3\_SOURCE\_CCP T1\_CCP1\_T3\_CCP2 | CCP1,2両方のクロックに使用 CCP２だけのクロックに使用 |

**【実際の使用例】**

下記のプログラム例は、これらのタイマー関数を使った実際の例です。  
  
《例１》  
　　タイマ０の約２５msec周期の割込みを使って、発光ダイオードを点滅させた例です。  
　　アイドルループでも別の発光ダイオードを点滅させています。  
  
　　　　[★　タイマ０の使用例](http://www.picfun.com/pic18/18ctest7.c)  
  
《例２》  
　　タイマ０とタイマ１を同じ優先順位の割込みで使った例で、タイマ０は約25msec周期  
　　タイマ１はその８倍の周期で割りこみます。  
　　アイドルループでは約０．５秒間隔でLEDを点滅させ、タイマ０では０．１秒間隔で  
　　タイマ１では１秒間隔でそれぞれ別の発光ダイオードを点滅させます。  
  
　　　　[★　タイマ０とタイマ１の同時使用例](http://www.picfun.com/pic18/18ctest9.c)

a33cffde-5ede-44ed-a9a0-b314d4873946

Y2:a33cffde-5ede-44ed-a9a0-b314d4873946

