

4-3 タイマ/カウンタ

タイマ/カウンタは、コンピュータにとってなくてはならない機能です。とくに、組み込みを目的とするワンチップ・マイコンでは、数多くのタイミングを発生する必要があるため、重要な周辺機能となっています。

タイマ/カウンタの機能で第一に挙げられるのが、ワンショットです。いろいろなところに利用することができますが、ある時点で起動をかけ、一定時間経過後にI/Oヘトリガをかける場合などでよく使用されます。

たとえば、スイッチ入力ではチャタリングが生じるため、正しくスイッチ入力を読むために、一度入ってきた入力をピンからの割り込みで受けて、再度チャタリングが落ち着いたあたりで読み返してスイッチ入力があったことを確認する手法がよくとられます。このときに最初のピンからの割り込みでワンショットをスタートさせ、一定時間経過後上がってくるタイマ割り込みのきっかけでピンを読むことで、チャタリングを避けてデータを取り込むことができます(Column 4-1参照)。

タイマがあるカウント値に達したときにCPUに割り込みをかけ、同時にタイマがリセットされ、再カウントをスタートさせることができます。

一度、レジスタの初期化設定をしておけば、自動的に繰り返しこの動作をさせることができます。この動作をCTC(クリア・タイマ・オン・コンペア・マッチ)モードと呼んでいます。一定時間おきに割り込みをかけられるので、OSなどでスケジューリングをするタイミングのためのシステム・クロックとしたり、一定時間おきに外部のモニタリングをする周辺デバイスに起動をかけるときなどに使用することも可能です。また、出力を取り出す設定をしておけば、CPUの介在なしに一定間隔で出力信号を取り出すことが可能です。

また、タイマを0からスタートし、最大値(TOP)あるいはある設定値に達したときにリセットし、カウントが再開し、設定値でリセット、...これを繰り返すような使い方もあります。タイマのカウント値の途中に閾値を設けておき、この閾値を横切ったとき、カウンタの出力ピンを“L”にし、カウント値がTOPに達したときに出力を“H”にするように設定することができます。閾値の設定を変更することで出力の“H”、“L”の期間を変更できます。

この動作をPWM(パルス幅変調)モードと呼んでいます。たとえば、この信号でLED(あるいはライト)を点灯させれば、明るさをコントロールすることができます。この出力信号をローパス・フィルタに通せば、DC電圧を取り出すことができます。言い換えれば、これはD-Aコンバータに対応しています。

その他、タイマ/カウンタはモータの駆動信号の生成、外部からのパルス信号の計数、外部パルス幅の計測、リモコンなどの送出信号の周波数変調を始め、多くのアプリケーションにとって必須の機能となっています。

4-3-1 8ビット・タイマ/カウンタ0

8ビット・タイマ/カウンタは、システム・クロックをプリスケアラで分周したクロック(またはT₀ピンの外部クロック・ソース)を入力として、それを最大(MAX)、256分の1することができます。16ビット・カウンタが最大65536分の1ですから、あまり分解能は高くありません。しかし、あまり分周比が大きくても周波数も低くなるわけですから、必ずしも大きなカウンタだけが必要というわけではありません。必要に応じて使い分けをします。

また、16ビット・タイマ/カウンタnに搭載されている機能が、すべて8ビット・タイマ/カウンタ0に搭載されているわけではありません。機能によっては、16ビット・タイマ/カウンタを選ばなければならない場合もあります。

この項では、8ビット・タイマ/カウンタ0に搭載されている機能について、順を追って説明します。図4-3-1にカウンタのブロック図を示します。また、表4-3-1には以降で使われるTOP、MAX、BOTTOMについて説明しました。次項では、8ビットにはなく16ビット・タイマ/カウンタnに搭載されている機能を説明していきます。

ノーマル・モード

ノーマル・モードは、タイマ/カウンタ動作の中で一番シンプルな動きをします。タイマ/カウンタ・レジスタTCNT0は、タイマ/カウンタ・クロック、countで1ずつカウント・アップします。このモードでは、カウンタTCNT0は単純にカウント・アップするだけです。Max値の0xFFに達した後、

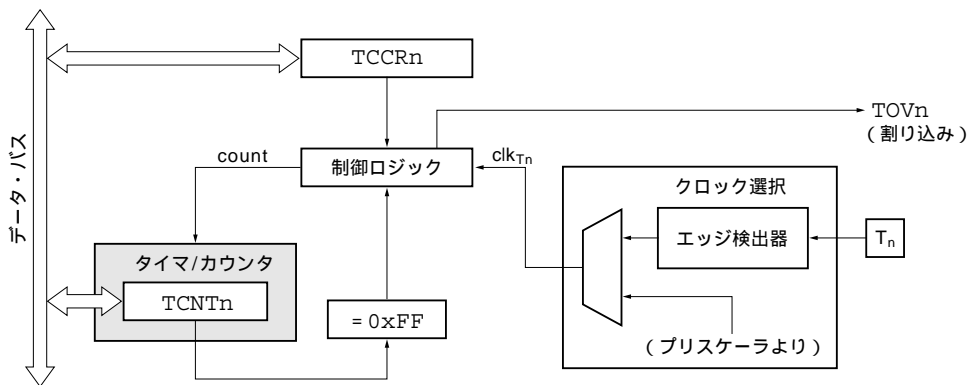


図4-3-1 ベーシック・タイマ/カウンタのブロック図(8ビットの場合)

表4-3-1 TOP、MAX、BOTTOMの定義

TOP	カウンタ値が最大値と等しくなったとき、カウンタはTOPに達したという。TOP値は、固定値に割り当てられたときの0xFF(MAX)か、OCR0Aレジスタの値である。割り当ては、操作モードによって決められる
BOTTOM	カウンタ値が0x00になったとき、BOTTOMに達したという
MAX	カウンタ値が0xFF(10進で255)に達したとき、MAXに達したという

見本

次のクロックでTCNT0はオーバーフローし、'0'になります。この時点でタイマ/カウンタ0インタラプト・フラグ・レジスタTIFR0のオーバーフロー・フラグTOV0がセットされます(図4-3-2参照)。

このフラグTOV0は、プログラムでこのビットに'1'を書くことでクリアすることができます。また、TOV0割り込みをイネーブルにしておくと、オーバーフローが起きたとき、

グローバル・インタラプト・イネーブル

タイマ/カウンタ0インタラプト・フラグ・レジスタTIFR0のTOIE0(タイマ/カウンタ0オーバーフロー・インタラプト0イネーブル)

の両方がセットされていれば、割り込みが上がり、割り込み処理がなされたときに、自動的にクリアされます。

TCNT0は'0'になった後も継続してカウントを続けます。

カウンタ値、TCNT0はいつの時点でも書き換えることが可能です。オーバーフロー割り込みが起きたときにTCNT0の値を更新すると、再度オーバーフローが発生したときに割り込みを発生させることができます。

countでカウント・アップすると簡単に書きましたが、もう少し詳しく見ておきましょう。

AVRでは完全に同期設計を行っているため、 clk_{Tn} はシステムI/Oクロックのイネーブル信号として作用します。プリスケール値が'1'の場合、 clk_{Tn} は常に'1'でI/Oクロックでカウントがアップします。プリスケール値が1/8の場合、8個目のI/Oクロックで clk_{Tn} がセットされ、これがI/Oクロックと論理積をとられたものが、TCNTnカウンタのクロックとなります(図4-3-3参照)。ATmega8のタイマ/カウンタ0の関連レジスタを図4-3-4に示します。

ここで、実際にワンショットを作ってみましょう。

ATmega8でシステム・クロックを1MHzに設定し(内部RCオシレータか外部XTALなどの選択をヒューズ・ビットで設定する。それにあったハードウェアの接続が必要)、16.7msのワンショット

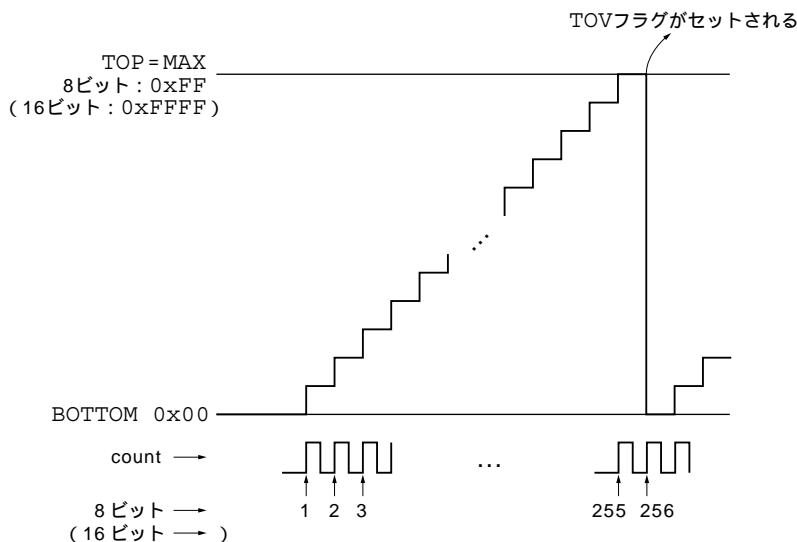


図4-3-2 タイマ/カウンタのノーマル・モードの動作

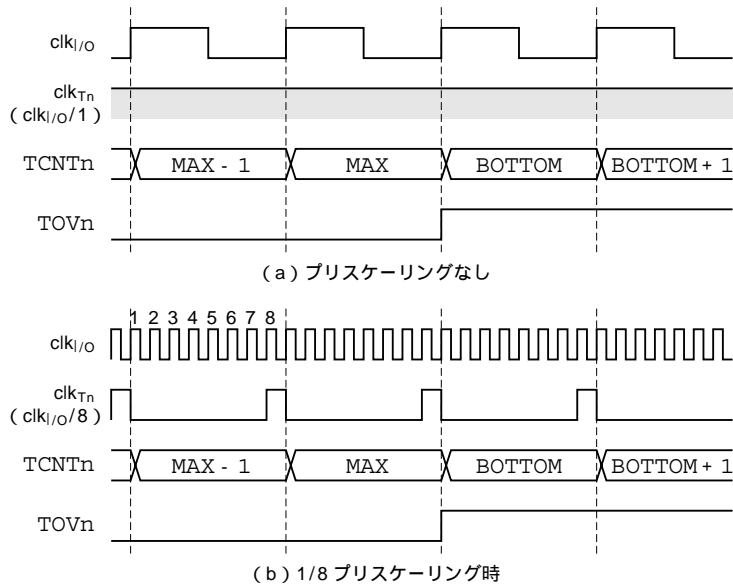


図4-3-3 タイマ/カウンタのタイムチャート

を作るとすると、システム・クロックの1サイクルが1 μ sですから、16700クロックで16.7msになります。

16.7msは周波数に直すと約60Hzです。プリスケール分周比1024で16700を割ると16.3になります。10進の16は0x10ですので、TCNT0に0x100 - 0x10 0xF0をセットしてタイマをスタートすると、約16msでオーバフロー割り込みを発生させることができます。

割り込みが発生したときに、TCNT0に0xF0を設定すれば、約16.7ms後に再度割り込みを発生します。ここでは前提としてノーマル・モード、タイマ/カウンタ・オーバフロー割り込みがイネーブルに設定してあるとしています。

ここで注意しておかなくてははいけないポイントがあります。割り込みが発生して割り込みサービス・プログラムでTCNT0を更新するときに数クロック費やすので、正確な時間を生成しなければならないときは、その時間を加えてTCNT0の値を決めます。

次に示すのが、lccでの初期化プログラムです。

```
void timer0_init(void)
{
    TCCR0 = 0x00; //タイマ停止
    TCNT0 = 0xF0; //タイマ/カウンタ値のロード
    TCCR0 = 0x05; //タイマ・スタート
}
```

割り込みハンドラは次のように書きます。