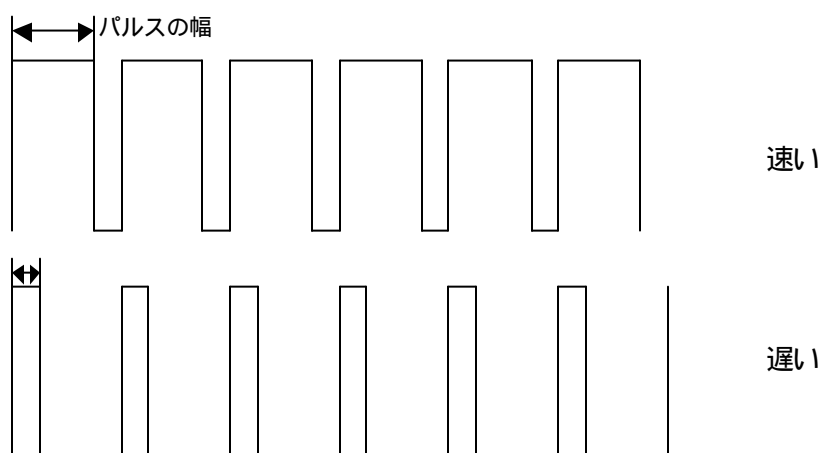


第4章 DCモーターのPWM制御

DCモーターの回転速度は、DCモーターの特性やギヤが内蔵されている場合は、ギヤ比などによって決まります。この回転速度を遅くしたい場合は、モーターに供給する電流を少なくする方法もありますが、今から説明するPWM制御という方法によってコントロールするのが一般的です。

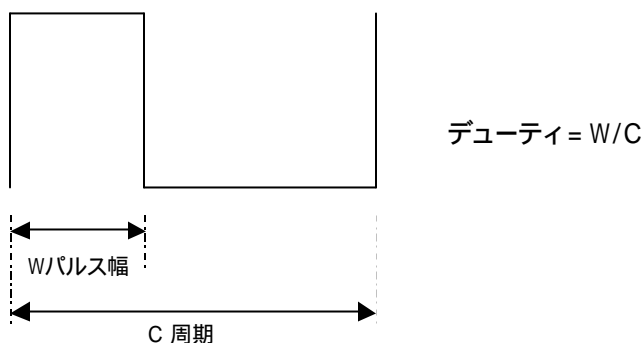
PWM制御は、パルスの幅でもって平均の電力を少なくする方式で、制御するのはパルスの幅ですからデジタルで制御しやすいです。



PWM制御は簡単に言えば、ONとOFFを繰り返す制御です。ONの期間を調整することによってモーターの速度をコントロールします。

PWM制御はモーターに限らず、いろいろなところで使われるマイコン必須の知識です。

さて、上図の2つのパルスを比べると、どちらも周期は同じです。違うのはパルスの幅です。周期に対するパルス幅の割合をデューティといいます。



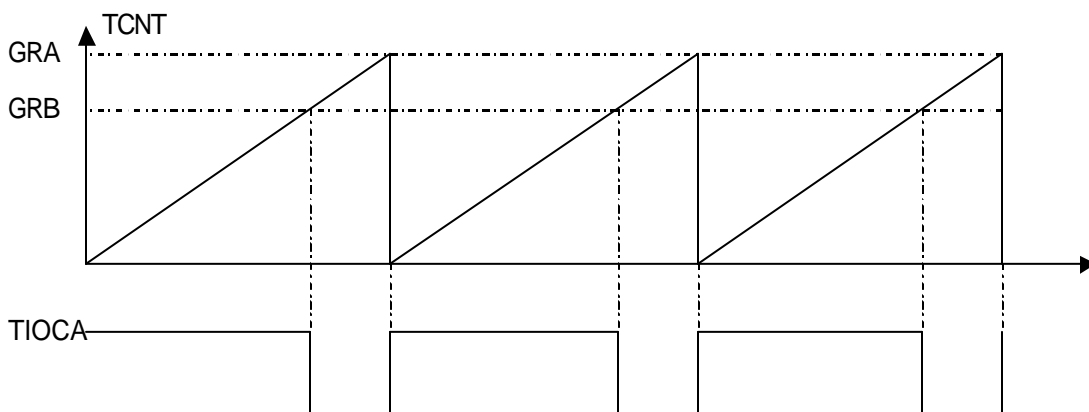
つまりデューティが大きいと、モーターが速く回転します。(最大100%)
 デューティが小さいと、モーターは遅く回転します。(最小0%=停止)

パルスを作り出す

パルスは、ポートのON/OFFを繰り返すことによって生成できます。しかし、マイコンにはPWM機能が内蔵されていますので、これを使う方が良いです。

マイコンにはタイマーが内蔵されていると説明しましたが、このタイマー機能の一部にPWM機能があるのです。

以前の章でタイマーのTCNTと、GRAレジスタを使って、一定間隔で発生する割り込みの学習をしました。PWM機能を使えば、TCNTとGRA、それとGRAと似たGRBレジスタを使って、任意の周期、デューティのパルスを作り出すことができます。その原理を以下に示します。



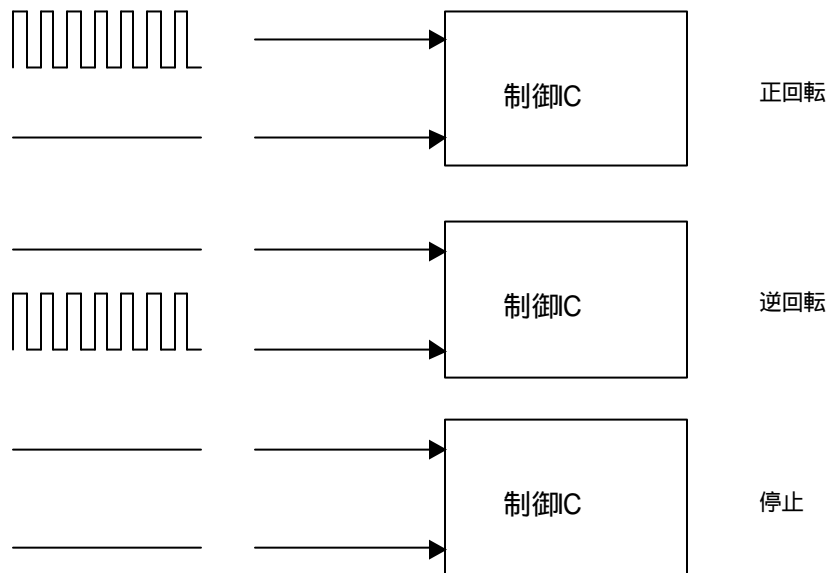
ノコギリ波形の斜めの線は、TCNTがクロックにしたがって、カウントアップしている様子です。TCNTがGRBレジスタの値に一致すると、TIOCAという端子の出力が反転されます。また、

TCNTがGRAレジスタの値に一致すると、TIOCAの出力が元に戻り、TCNTが0にクリアされます。結果的にTIOCA端子からはパルスが出力されます。

この図を見ると、パルスの周期はGRAレジスタの値で決まります。これは、タイマー割り込みのときと同じです。またパルスの幅、つまりデューティはGRBレジスタの値で決まります。

パルスは2つ必要

さて、DCモーターを制御する専用ICは、入力端子が2つあります。ですから、パルスは2系統作する必要があります。ただし、両方同時にパルスを出力する必要はありません。どちらかは、必ず0信号です。



そこで、タイマーのチャンネル0と1を使うことにします。プログラム上ではTCNT0、TCNT1のように、レジスタ名の最後にチャンネル番号を付けて区別します。

レジスタの設定

タイマーのPWM機能を使ってパルスを出力するためのレジスタ設定方法を説明します。

まず、パルスの周波数を決めます。パルスの周波数はモーターや制御ICの特性に合わせて決めますが、弊社での実験の結果、数百Hzが良いようです。周波数が高いと制御ICがついていけないようで、モーターはまったく回転しませんでした。（この制御ICはそもそもPWM制御用ではありません。PWM制御に適したICを使う場合は数KHzが良いようです。）

さて数百Hzというのは、システムクロック(24.576MHz)から比べるとかなり遅い値です。ですから、システムクロックを分周してタイマーに供給する必要があります。

システムクロックの分周はTCRレジスタで行います。タイマー割り込みの章で説明したように最大8分周することができます。

```
TCR = 3; //8分周
```

8分周で24576KHz / 8 = 3072KHzになります。

あとは、GRAレジスタの値によって周期が決まります。約3000000Hzを数百Hzまで落とすわけですからGRAの値はおよそ10000になります。したがって、GRA=10000とします。このときの周波数は

$$3072000\text{Hz} / 10000 = 307\text{Hz}$$

となります。

GRBレジスタの値によってパルス幅が決まります。最大はGRB=10000で、このときデューティ100%、最小はGRB=0、デューティ0です。GRBの値を0から10000まで変化させてDCモーターの回転速度を制御します。

その他の設定

タイマー割り込みのところで説明したように、TCNTの値がGRAに一致したらTCNTを0にクリアしなければなりません。この設定は、TCRレジスタで行います。先ほどの分周の設定と合わせて、次のようになります。

$$\text{TCR} = 0x23;$$

次に、タイマーのPWM機能を利用するため、PWMモードにする設定を行います。これは、TMDRというレジスタで行います。このレジスタは初めて出てくるので、レジスタ表を掲載します。

TMDR

ビット	7	6	5	4	3	2	1	0
	-	MDF	FDIR	PWM4	PWM3	PWM2	PWM1	PWM0
初期値	1	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ビット6	説明
MDF	
0	チャンネル2は通常動作
1	チャンネル2は位相計数モード

ビット5	説明
FDIR	
0	TSR2のOVFフラグは、TCNT2がオーバーフローまたはアンダーフローしたときに1にセット
1	TSR2のOVFフラグは、TCNT2がオーバーフローしたときに1にセット

ビット4	説明
PWM4	
0	チャンネル4は通常動作
1	チャンネル4はPWMモード

ビット3	説明
PWM3	
0	チャンネル3は通常動作
1	チャンネル3はPWMモード

ビット2	説明
PWM2	
0	チャンネル2は通常動作
1	チャンネル2はPWMモード

ビット1	説明
PWM1	
0	チャンネル1は通常動作
1	チャンネル1はPWMモード

ビット0	説明
PWM0	
0	チャンネル0は通常動作
1	チャンネル0はPWMモード

今回は、チャンネル0と1の2チャンネルをPWMモードで使用しますので、TMDRの設定値は3になります。

```
TMDR |= 3;
```

TMDRレジスタは、使用するチャンネルだけではなく、他のチャンネルの設定も含まれますから、|=演算を使用して他のチャンネルに影響しないように配慮しています。

最後に、タイマーのカウンタをスタートさせます。これは、タイマー割り込みのところで説明した通りTSTRレジスタで行います。

```
TSTR |= 3;
```

以上まとめると、必要なレジスタの設定は以下のようになります。

```
TCR0 = 0x23;
```

```
TCR1 = 0x23;
```

```
GRA0 = 10000;
```

```
GRA1 = 10000;
```

```
GRB0 = 0 ~ 10000の値
```

```
GRB0 = 0 ~ 10000の値
```

```
TMDR |= 3;
```

```
TSTR |= 3;
```

プログラムの実行方法

プロジェクトは[dc_motor2]フォルダの[dc_motor2.yip]です。

このプログラムを実行すると、モーターは最初停止しています。

テンキーから数字キーを入力すると、その10倍のデューティで回転します。例えば[5]キーを押すと、デューティ50%で回転します。[0]キーでは停止です。

(なお、デューティ10%と100%ではモーターは回転しません。デューティが小さい場合にモーターが回転しないのは、パルスの幅が短すぎて、モーターを回転させるだけの時間電流が流せないものと思われます。また100%では、GRA=GRBとなり、この場合PWMモードが機能しないものと思われます。)

サンプルプログラム解説

```
/*
*****
**** PWM機能を初期化する ****
****      引数   なし      ****
****      戻り値 なし      ****
*****
void pwm_init(void)
{
    TCRO = 0x23;          //CH0 1/8分周 GRA一致でTCNTクリア
    TCR1 = 0x23;          //CH1 1/8分周 GRA一致でTCNTクリア
    GRB0 = 0;             //CH0 デューティ (最初は0)
    GRB1 = 0;             //CH1 デューティ (最初は0)
    GRA0 = 10000;         //CH0 周期 10000/24576000秒
    GRA1 = 10000;         //CH1 周期 10000/24576000秒
    TMDR = 0x03;         //CH0 CH1 はPWMモード
    TSTR |= 0x03;        //カウンタスタート
}
```

この関数は、PWM機能の初期化関数です。この関数実行時点では、まだモーターは回転させませんので、GRB0/GRB1の値は0にします。

```

/*****
**** PWMを使ってDCモータを回転させる ****
****   引数   dir   1 正方向 ****
****                   0 逆方向 ****
****   dty   デューティ ****
*****/
void pwm_dc_motor(int dir, int dty)
{
    if (dir == 0) {
        //逆方向
        GRB0 = 0;           //パルス幅
        GRB1 = dty*100;    //パルス幅
    }
    else if (dir == 1) {
        //正方向
        GRB0 = dty*100;    //パルス幅
        GRB1 = 0;         //パルス幅
    }
}

```

この関数は、任意のデューティ (dty) でモータを回転させる関数です。dirは回転方向で、1のとき正回転で、0のとき逆回転です。デューティはdtyで指定します。