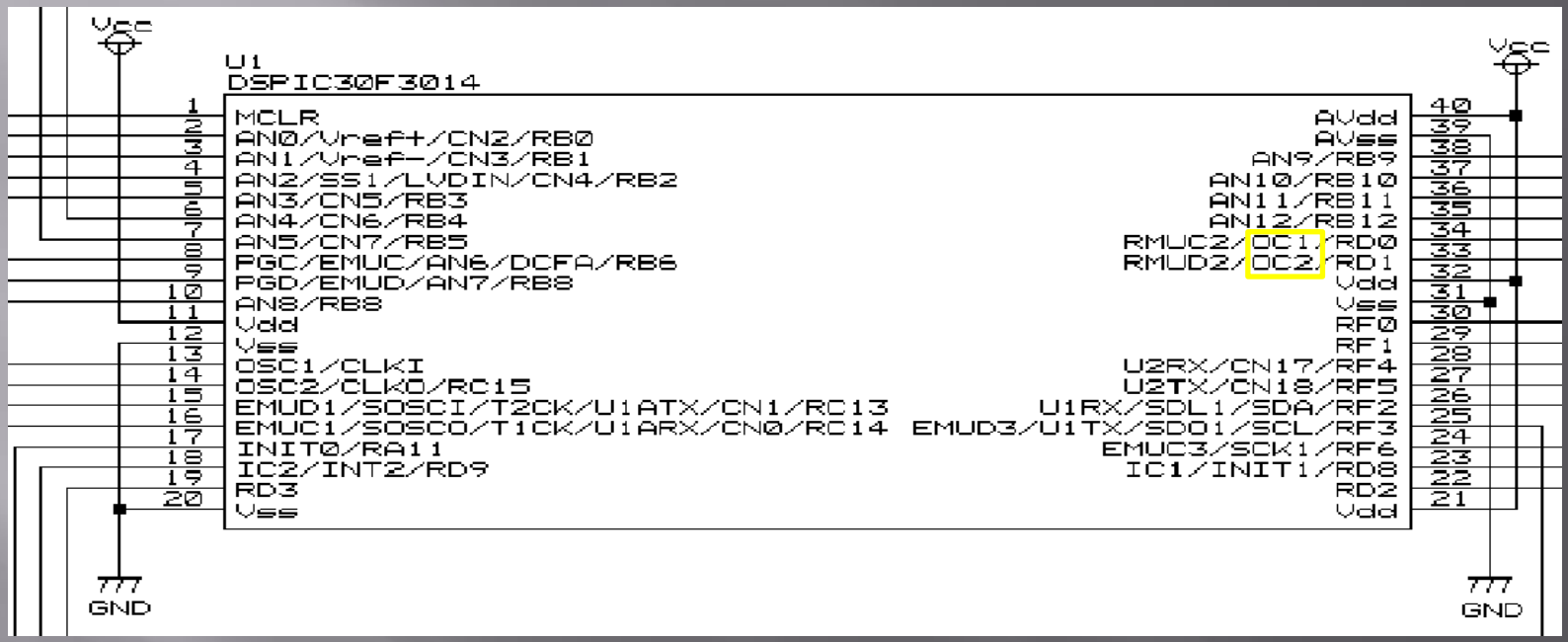
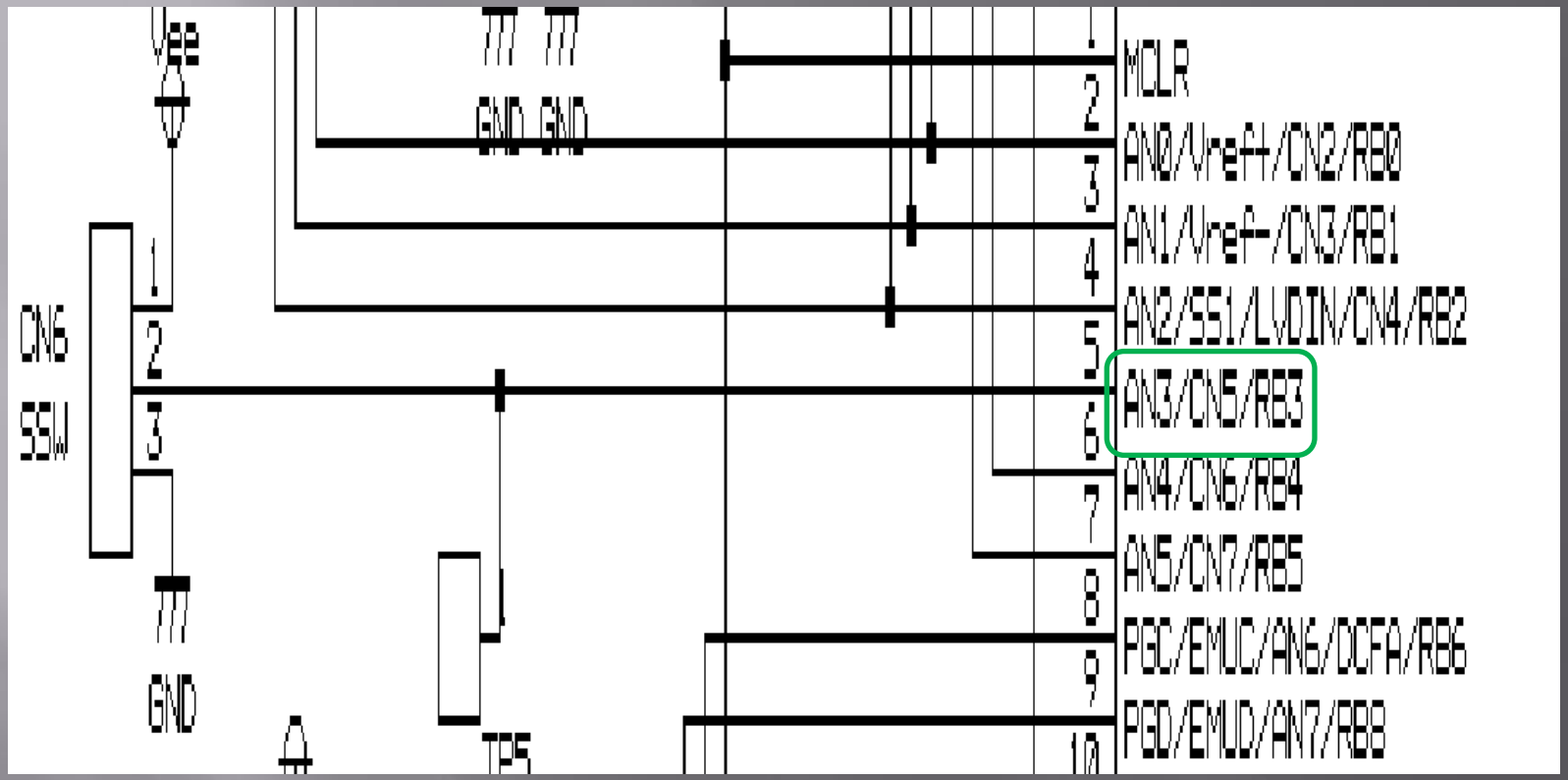
サーボモータの制御方法

サーボモータはPWM制御を使用し簡単に制御することができます。
今回はタイマを使用しPWM信号を生成する制御法を行っています。
タイマを使うのには理由があります。



dsPIC30F3014にはPWMのピンが2つありますが、
それぞれモータで使用しているため使用できません。

そこで、超音波センサ用に用意されていた回路で代用します。



Veeには電源の電圧が直接来ています。
つまりEneloop1.2V×６本　＝　7.2V
がサーボモータに供給されます。

今回使用したサーボモータのスペック表

- 双葉電子工業(株)製 S3003
- 寸法：40.4 x 19.8 x 36 [mm]
- 重量：37.2 [g]
- 動作電源：4.8-6.0 [V]
- 動作スピード：0.23 [sec/60°]（4.8 [V] 時），0.19 [sec/60°]（6.0 [V] 時）
- 出力トルク：3.2 [kg·cm]（4.8 [V] 時），4.1 [kg·cm]（6.0 [V] 時）
- 動作角度：± 60 [deg] 程度*4

しかし今回は7.2Vを供給する

*4 仕様では ±60° となっているが，実際には約 ±90° 程度まで動作可能である．ただし，R/C サーボの回転角度が構造上の可動範囲を超えた場合，サーボ内部のギアが破損する可能性があるので注意せよ．

ここが重要
検証により-95度~90度動作することがわかった

今回使用するS3003のパルス幅は0.5~2.3msとなっています。
つまり1.8msの間で0~180度動くと考えると

S3003は180度以上回転します。
そのため実際にパルス幅を調べていくと
0.55 ~ 2.35くらいがちょうど
0 ~ 180度の範囲のパルス幅となっています。

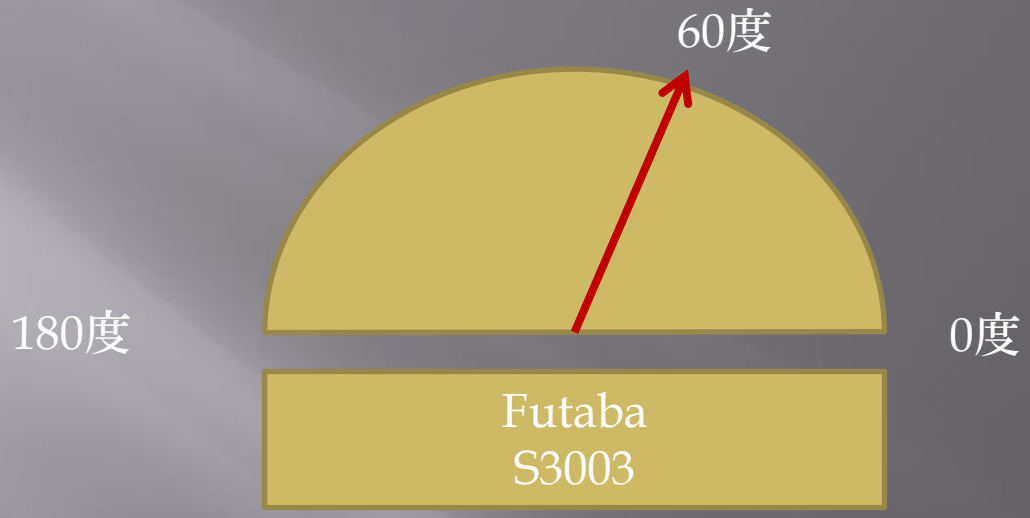
1.8 ÷ 180 = 0.01msで一度動くことがわかる

これをタイマの割り込み周期にすれば
タイマでサーボモータが制御できる

S3003が認識する周期幅が6 ~ 25msの範囲となっています。
なので今回は20msくらいの周期幅でhigh low を切り替えます。

タイマの割り込み周期が0.01msなので

2000カウントで20msとなります。



例として60度にサーボモータを制御する場合を考えます。

0.01×60 = 0.6ms
0.55が0度なので 0.6 + 0.55 = 1.15ms

タイマは0.01ms毎の割り込みなので

115カウント　high
残りの
1885カウント　low
とする周期をサーボに送ることで60度に制御することができる。

同様に
0度に制御したい場合は55カウント分high
残りをlowにすればよい

180度に制御したい場合は255カウント分high
残りをlowにすればよい

1度動かすために変化させるhighのカウント数は1です。

タイマの割り込み周期を0.01msとしたのはこのためです。

```
void ServoInitFunc(void){

        servo(0);
        ConfigIntTimer1(T1_INT_PRIOR_1 & T1_INT_ON);
        //1msec(4/80Mhz×1×200=0.01msec)
        OpenTimer1(T1_ON & T1_GATE_OFF & T1_PS_1_1 &
                T1_SYNC_EXT_OFF & T1_SOURCE_INT,200-1);

}
```

サーボモータ初期化用関数
周期生成にタイマ1を使用するため
0.01msの割り込み周期でタイマ1をオープン

```
void servo(int angle){
    angle += 145;

    ServoTargetValue = angle;

}
```

90 + 55

タイマ1の方でhigh側のカウント上限
に使用するグローバル変数

サーボモータの角度指定用関数
関数内で角度の座標系を0 ~ 180度から
-90 ~ 90度に変更しています。

```
unsigned int TimeCount = 0;

void _ISR _T1Interrupt(void){
        IFS0bits.T1IF = 0;
        TimeCount++;

        if(TimeCount < ServoTargetValue){
                SERVO = 1;
        }else if(TimeCount <= 2000){
                SERVO = 0;
        }else{
                TimeCount = 0;
        }

}
```

サーボの信号端子にPWM信号を出力します。
タイマ資源が余っているとき限定ですが
この方法を使用することですべての汎用入出力ピンにおいて
PWM制御を行うことができるようになります。