

信号発生ユニット(PLL)

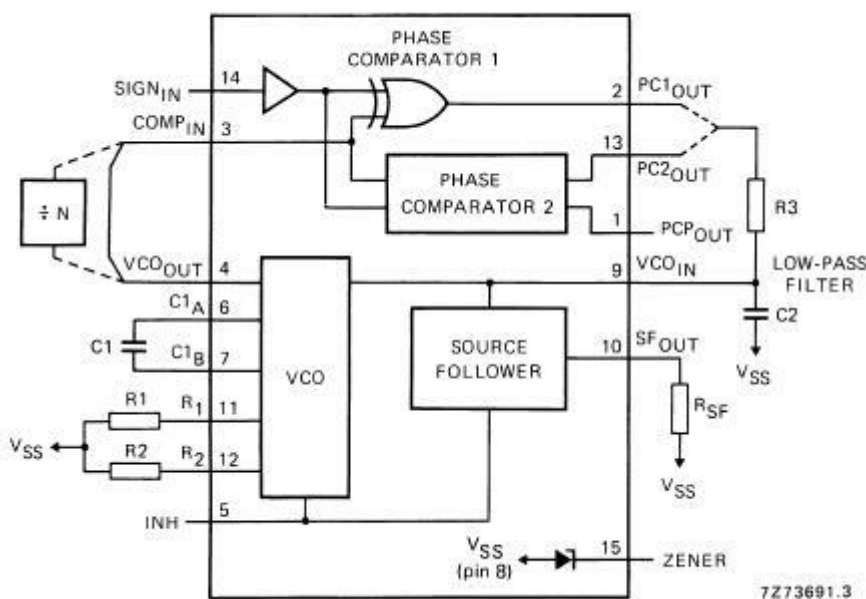
概要

PLLは、電圧に応じて周波数が変化するVCO(電圧制御発振回路)の出力信号と入力(基準周波数)との位相差をVCOにフィードバックすることにより、同期(ロック)させるものです。このときVCO(電圧制御発振回路)の出力信号を分周したものを用いることにより入力信号の周波数を逡倍した信号を得ることが出来ます。基準周波数をN倍(分周の逆)した周波数を得ることが出来るので「掛算器」とも呼ばれます。

動作原理

PLL用のIC(4046BとPIC2個)を使用しました。周波数は、100kHz~1500kHzまでを1kHz単位または10kHz単位でUP/DOWNできます。

4046B(PLL)は、基準周波数と比較周波数の差分を検出する位相比較器、および、デジタルパルスを発生するVCOより構成されています。



基準周波数1kHzの発振部と分周 $\div N$ 部にはPIC12F683を使用しました。発振部のPICではCPPモジュールをコンペアモードで使用し、16MHzのクロックより、精度の高い1kHzの信号を出力します。分周部のPICではCPPモジュールをコンペアモードで使用し、VCOからの出力信号を分周し出力します。その時の分周値は、プッシュスイッチによりUP/DOWNすることが出来ます。起動時は、500kHzに設定されていますUP/DOWNのプッシュスイッチを同時に押すと500kHzに戻ります。

※4046Bは、サトー電子で136円で通販しています。(2007年5月19日時点) ※4046Bよりも高機能(上限約16MHz)な74HC4046Pが693円でした。


```
}

/*****
*****/

void main()
{
    // クロックの設定 → 今回は外付けの16Mhz クロックモジュールを使用する。
    OSCCON = 0b01110000;
    // コンパレータの設定 → 今回は使用しない。
    CMCON0 = 0b00000111;
    // アナログの設定 → 今回は使用しない。
    ANSEL = 0b00000000;
    // ポートの設定
    TRISIO = 0b00101000;
    OPTION_REG.F7 = 0;
    // 入力割り込みの設定 → 今回は使用しない。
    INTCON.INTE = 0;
    INTCON.INTF = 0;
    OPTION_REG.INTEDG = 0;
    // 入力割り込み (変化) の設定 → 今回は使用しない。
    INTCON.GPIE = 0;
    INTCON.GPIF = 0;
    // CCPの設定
    PIE1.CCP1IE = 1;
    PIR1.CCP1IF = 0;
    CCP1CON = 0b00001011;
    CCPR1L = 0xF4; // 0.1sec...10hz... クロックが16Mhzの時
    CCPR1H = 0x01; // 0.1sec...(1÷16000000)*4*8*50000
    // TIMER0の設定 → 今回は使用しない。
    INTCON.T0IE = 0;
    INTCON.T0IF = 0;
    TMR0 = 0;
    OPTION_REG.T0CS = 0;
    OPTION_REG.T0SE = 0;
    OPTION_REG.PSA = 0;
    OPTION_REG.PS0 = 0;
    OPTION_REG.PS1 = 0;
    OPTION_REG.PS2 = 0;
    // TIMER1の設定
    PIE1.TMR1IE = 0;
    PIR1.TMR1IF = 0;
    TMR1L = 0;
    TMR1H = 0;
    T1CON.T1CKPS0 = 1;
    T1CON.T1CKPS1 = 1;
    T1CON.TMR1ON = 1;
    // TIMER2の設定 → 今回は使用しない。
    PIE1.TMR2IE = 0;
    PIR1.TMR2IF = 0;
    T2CON.TMR2ON = 0;
```

```

T2CON.T2CKPS0 = 0;
T2CON.T2CKPS1 = 0;
TMR2 = 0;
// 割り込み(全体)の設定
INTCON.PEIE = 1;
INTCON.GIE = 1;
while (1) {
}
}

/*****
*****/

```

PLLカウンタ部

pll1.c

```

/*****
*****/

void interrupt(){
    if (PIR1.CCP1IF == 1) {
        PIR1.CCP1IF = 0;
        //
        GPIO.F4 = 1;
        asm    nop;
        asm    nop;
        GPIO.F4 = 0;
    }
}

// ****
*

void main()
{
    static    unsigned    int    cnt;
    OSCCON = 0b01110000;    // クロックは8Mhz
    CMCON0 = 0b00000111;    // コンパレータは使用しない。
    ANSEL = 0b00000000;    // AN0を使用しない。
    TRISIO = 0b00100111;    // ポート設定
    OPTION_REG = 0b00000000;
    WPU.F0 = 1;    // プルアップ
    WPU.F1 = 1;    // プルアップ
    WPU.F2 = 1;    // プルアップ
    // CCPの設定
    PIE1.CCP1IE = 1;
    PIR1.CCP1IF = 0;
    CCP1CON = 0b00001011;
    cnt = 500;    // 500kHz(初期値設定)
}

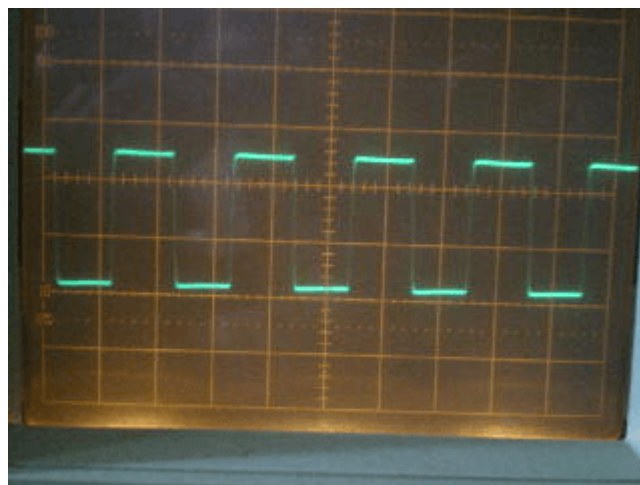
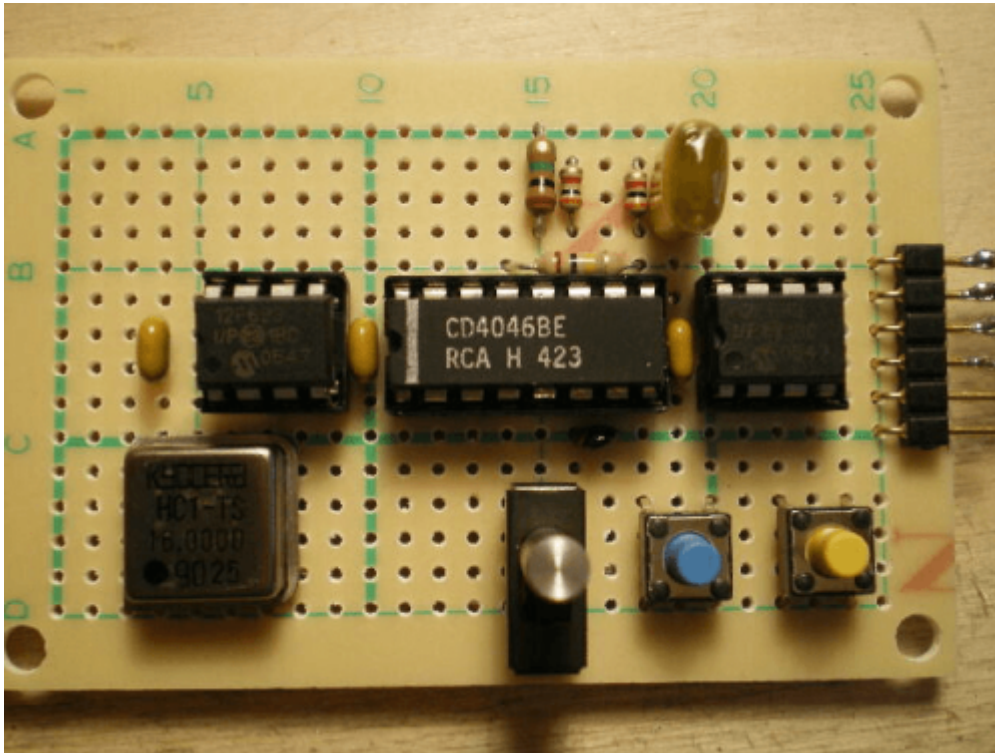
```

```
CCPR1L = cnt & 0xFF;
CCPR1H = (cnt >> 8) & 0xFF;
// TIMER1の設定
PIE1.TMR1IE = 0;
PIR1.TMR1IF = 0;
TMR1L = 0;
TMR1H = 0;
T1CON.T1CKPS0 = 0;
T1CON.T1CKPS1 = 0;
T1CON.TMR1CS = 1;
T1CON.TMR1ON = 1;
// 割り込み(全体)の設定
INTCON.PEIE = 1;
INTCON.GIE = 1;
//
while (1) {
    if ((GPIO.F0 == 1) && (GPIO.F1 == 1))
        continue;
    if ((GPIO.F0 == 0) && (GPIO.F1 == 1)) {
        if (GPIO.F2 == 1)
            cnt++;
        else
            cnt += 10;
    }
    if ((GPIO.F0 == 1) && (GPIO.F1 == 0)) {
        if (GPIO.F2 == 1)
            cnt--;
        else
            cnt -= 10;
    }
    while (1) {
        if ((GPIO.F0 == 0) && (GPIO.F1 == 0)) {
            cnt = 500;
        }
        if ((GPIO.F0 == 1) && (GPIO.F1 == 1))
            break;
        Delay_ms(1);
    }
    if (cnt < 90)
        cnt = 90;
    if (cnt > 1510)
        cnt = 1510;
    CCPR1L = cnt & 0xFF;
    CCPR1H = (cnt >> 8) & 0xFF;
}
}

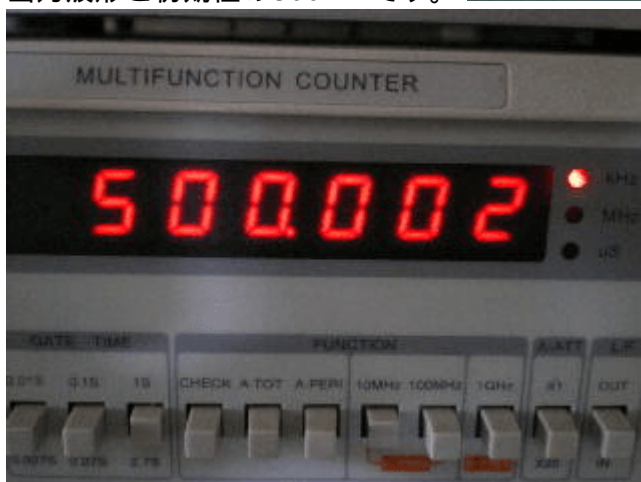
//*****
*
```

動作確認

左のスライドスイッチで、単位□1kHz□10kHz□を切り替えます。 青色プッシュスイッチでUP□黄色プッシュスイッチでDOWNします。



出力波形と初期値の500kHzです。

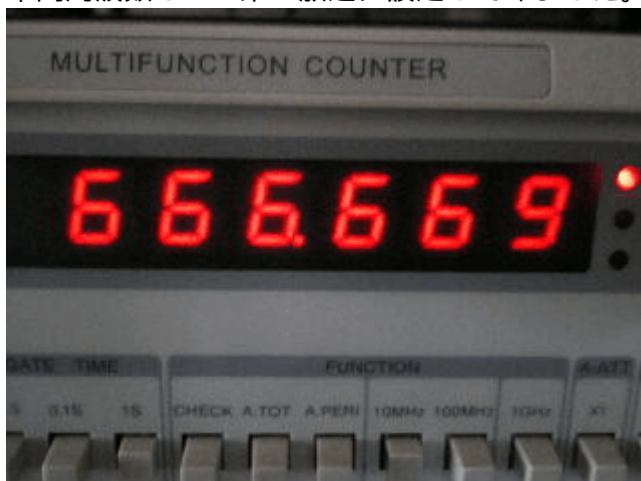




最小値の100kHzと最高値の1500kHzです。



中間周波数とNHK第一放送に設定してみました。



プッシュスイッチで1kHzまたは10kHz単位で設定できますのでとても便利です。 クロックモジュールに もっと精度の高い物を使用すれば、更に精度の高い発振器になりますね。

著作権表示 **copyright notice**

このページは稲崎様の閉鎖したHPのコピーで、著作権は稲崎様にあります。 [詳細](#) This page is a copy of Mr. Inasaki's closed website, and the copyright is held by him. [Details](#)

From:

<http://deepsky.jp/wiki/> - うごくといいな

Permanent link:

<http://deepsky.jp/wiki/doku.php?id=elechobby:picdic:pic12f683:9>

Last update: **2020/05/01 20:19**

