

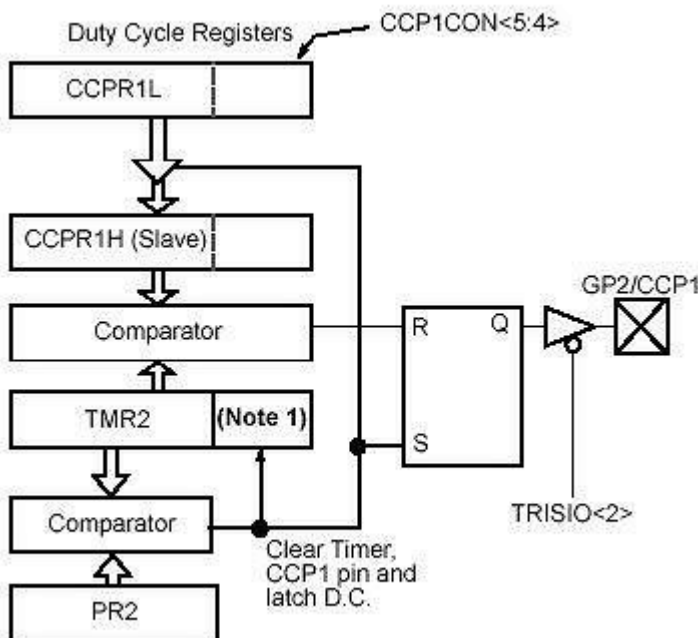
# 簡易信号発生ユニット(PWM)2

## 概要

近々にAMのワイヤレスマイクを作成しようと思っています。その発振部に、コイルとバリコンでは、安定性に欠けるので何か方法は無いものかと思案しておりました。水晶を使う手もあるのですが、必要な周波数分購入しなければなりません。その応用でVXO(Variable-crystal oscillator(可変周波型水晶発振器))はどうかと思いましたが、可変範囲が狭すぎます。あれこれ思案した末、PICが内蔵しているCCPモジュール(Compare/Capture/PWM)のPWMモードが使用できるのではと考えました。

## 動作原理

PICのPWMモードでは、周期とデューティ(duty)が設定できます。



まず、周期が最高どこまで設定可能かを計算し

てみます。

- 周期  $(PR2+1) \times 4T_{osc} \times$  (TMR2のプリスケール値)

クロックが20Mhzの場合には、

- $1667\text{kHz} = (2+1) \times 4 \times 0.00000005 \times$

この時に、PR2の値を2にしたのは、本ユニットの出力は正弦波としますので、デューティは周期の半分となるためです。PR2を変化させていくと、次のような周波数を得ることが出来ます。

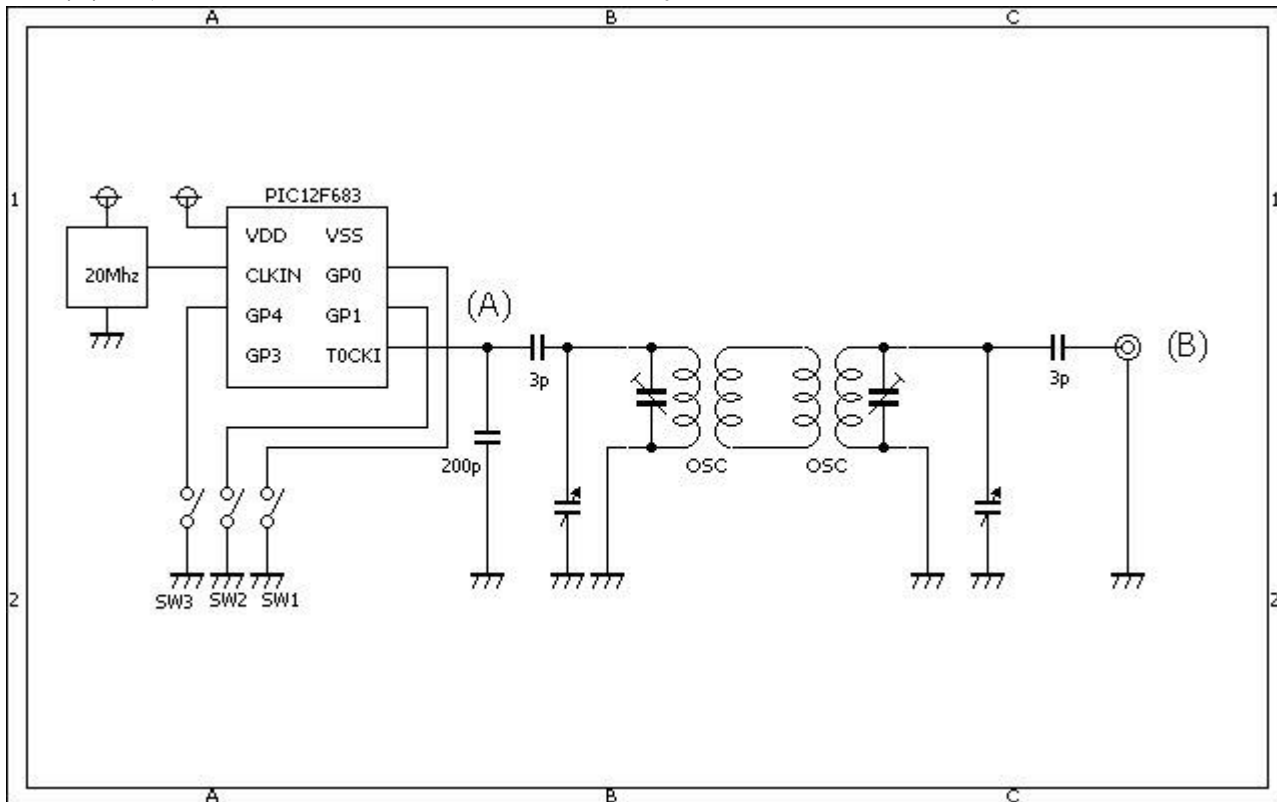
- 2  $\times$  1667Khz
- 3  $\times$  1250Khz
- 4  $\times$  1000Khz
- 5  $\times$  833Khz
- 6  $\times$  714Khz
- 7  $\times$  625Khz

- 8 555Khz
- 9 500Khz

これだけの周波数が設定できればAMラジオの範囲内としては、十分かなと思います。但、PICの出力は、当然のことですが、出力が方形波となってしまいます。そこで、この後に2段の同調回路を設けて正弦波に変換することにしました。

## 回路図

SW1,2,3で、上記の8種類の周波数を切り替えます。



## ソースコード

freq.c

```
//*****  
*  
#define      sw0      GPIO.F0  
#define      sw1      GPIO.F1  
#define      sw2      GPIO.F4  
  
//*****  
*  
void Pwm_Change_DutyEx(unsigned int duty_ratio)  
{
```

```
CCPR1L = duty_ratio >> 2;
CCP1CON.F6 = duty_ratio & 0b00000001;
CCP1CON.F7 = (duty_ratio & 0b00000010) >> 1;
}

//*****
*

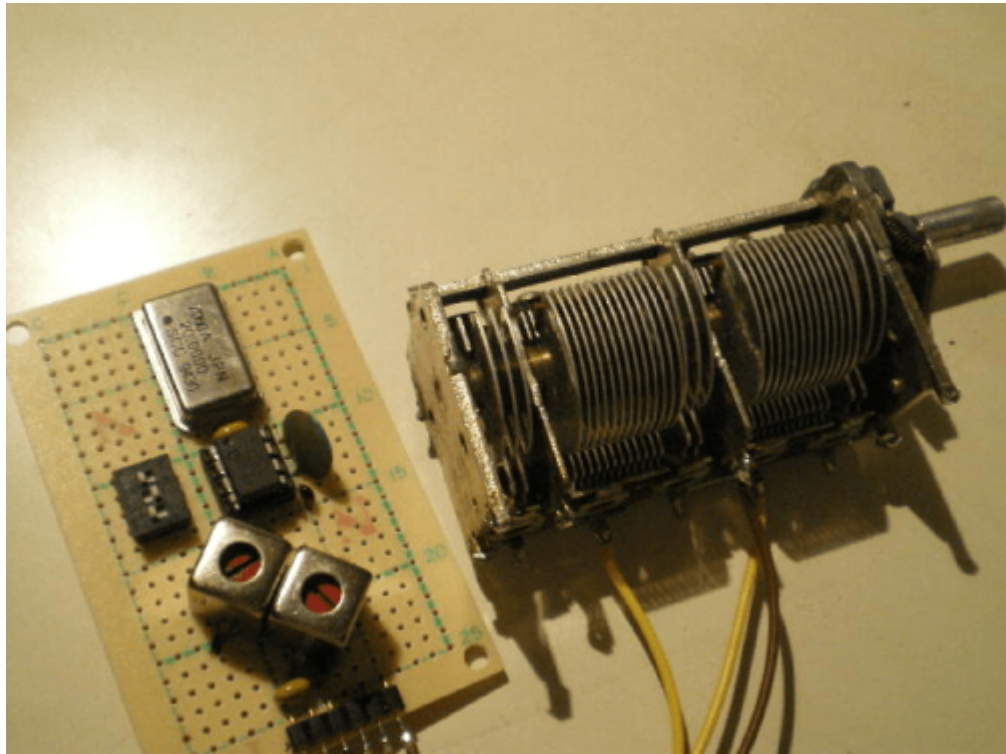
void main()
{
    static    unsigned    char    temp;
    //
    OSCCON = 0b01110000;    // クロックは1Mhz
    CMCON0 = 0b00000111;    // コンパレータは使用しない。
    ANSEL = 0b00000000;    // AN0を使用する。
    TRISIO = 0b00011011;    // ポート設定
    WPU.F0 = 1;            // プルアップ
    WPU.F1 = 1;            // プルアップ
    WPU.F4 = 1;            // プルアップ
    OPTION_REG = 0b00000000;
    //
    Pwm_Init(1000000);    // 1Mhz
    PR2 = 2;
/*
1666Khz 2
1250Khz 3
1000Khz 4
833Khz 5
714Khz 6
625Khz 7
555Khz 8
500Khz 9
*/
    Pwm_Change_DutyEx((PR2 * 4) / 2);
    Pwm_Start();
    //
    temp = GPIO & 0b00010011;
    while (1) {
        if (temp != (GPIO & 0b00010011)) {
            temp = GPIO & 0b00010011;
            switch (temp) {
                case 0b00000000:
                    PR2 = 9;    // 500Khz
                    break;
                case 0b00000001:
                    PR2 = 8;    // 555Khz
                    break;
                case 0b00000010:
                    PR2 = 7;    // 625Khz
                    break;
                case 0b00000011:
```

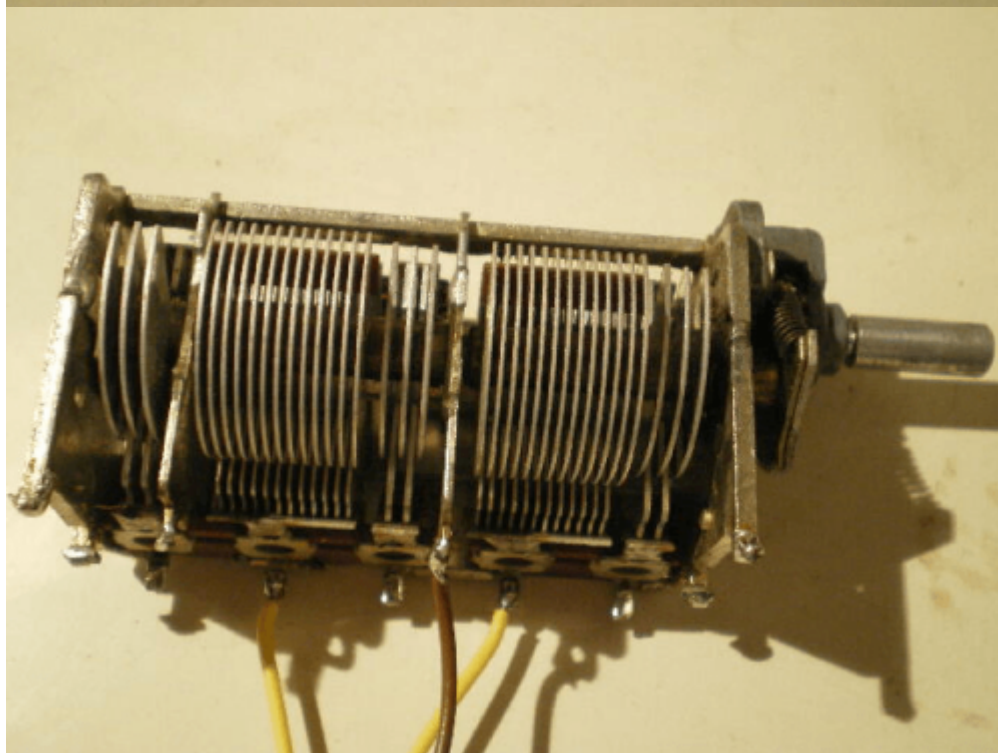
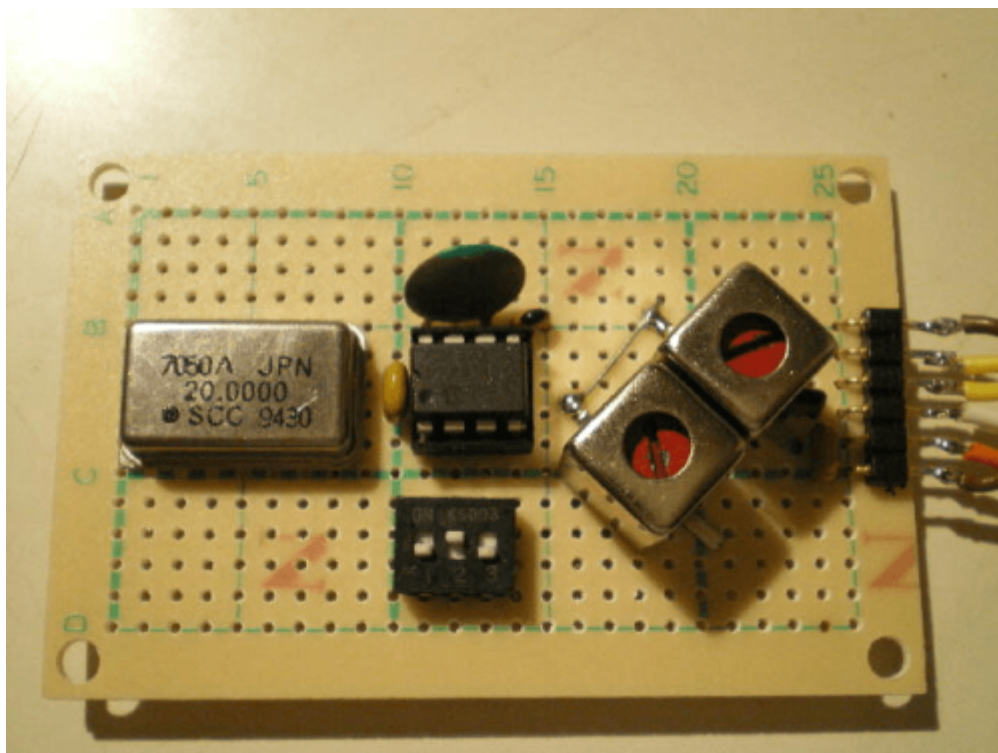
```
        PR2 = 6;    // 714Khz
        break;
    case 0b00010000:
        PR2 = 5;    // 833Khz
        break;
    case 0b00010001:
        PR2 = 4;    // 10000Khz
        break;
    case 0b00010010:
        PR2 = 3;    // 1250Khz
        break;
    case 0b00010011:
        PR2 = 2;    // 1666Khz
        break;
    }
    Pwm_Change_DutyEx((PR2 * 4) / 2);
}
}
}

//*****
*
```

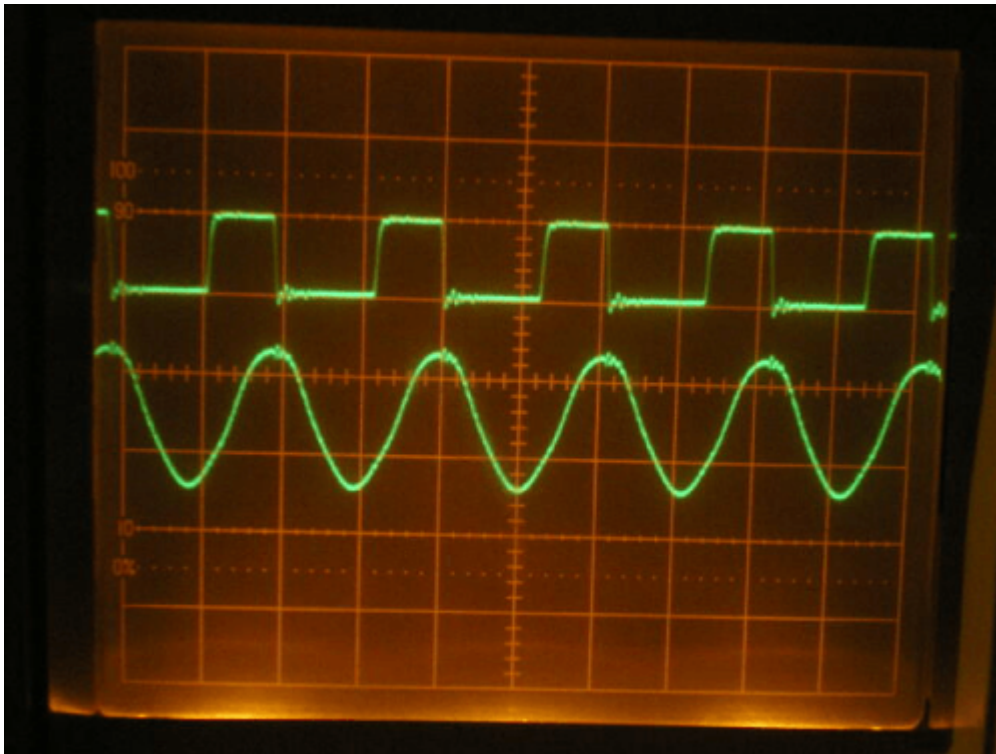
## 動作確認

水晶は手持ちの20.0000Mhzのクロックモジュールを使用、周波数の切り替えは、3個のディップSWで、同調は2個のOSCコイルと2連バリコンで行いました。





上段が(A)PICの出力波形です。下段が(B)2段の同調回路を通った後の波形です。若干、方形波の立下りに歪が発生します。



流石にクロックモジュール

の精度が高いので周波数は結構安定しています□□1Mhz設定時>



安定した周波数をAMラジ

オの範囲で8段階に設定できるのでとても便利です。これをベースにして□AMワイヤレスマイクを作成しようと考えています。

From: <http://www.deepsky.jp/wiki/> - うごくといいな

Permanent link: <http://www.deepsky.jp/wiki/doku.php?id=elechobby:picdic:pic12f683:8&rev=1588058794>

Last update: 2025/10/17 14:27

