

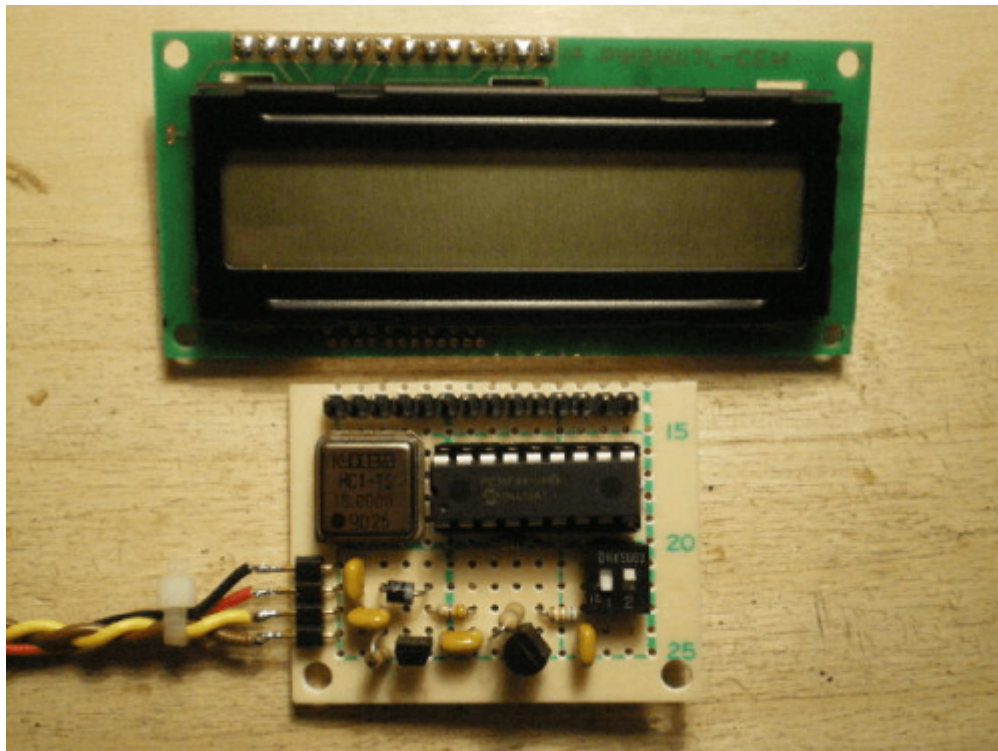
周波数表示ユニット2(真空管ラジオ用)

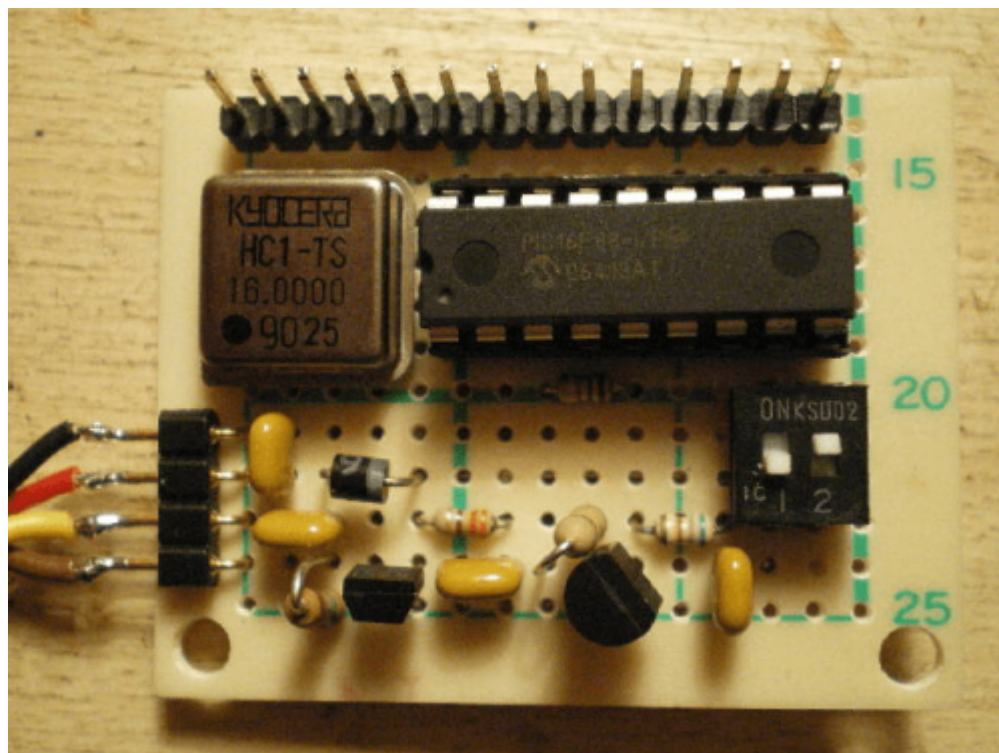
概要

標準的な真空管ラジオ(5球スーパー)に接続して、受信周波数を表示するものです。受信周波数は、局部発振(前段の7極管のカソードより)の周波数より、中間周波数分(455Khz)をマイナス(-) したものを表示します。

- SW切り替え(SW1)で、マイナスするかしないかの選択が出来ます。
- SW切り替え(SW2)で、Khz表示またはHz表示の選択が出来ます。

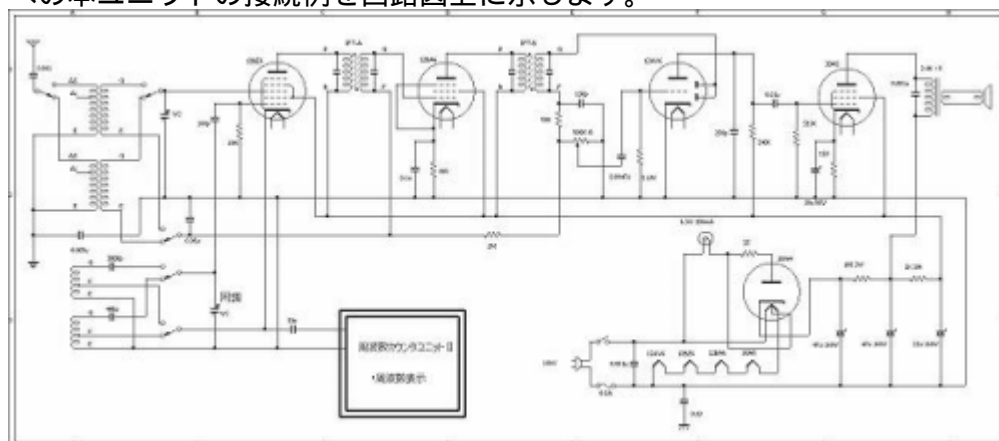
最大25Mhzくらいまでカウントできますので短波帯でも十分使えると思います。



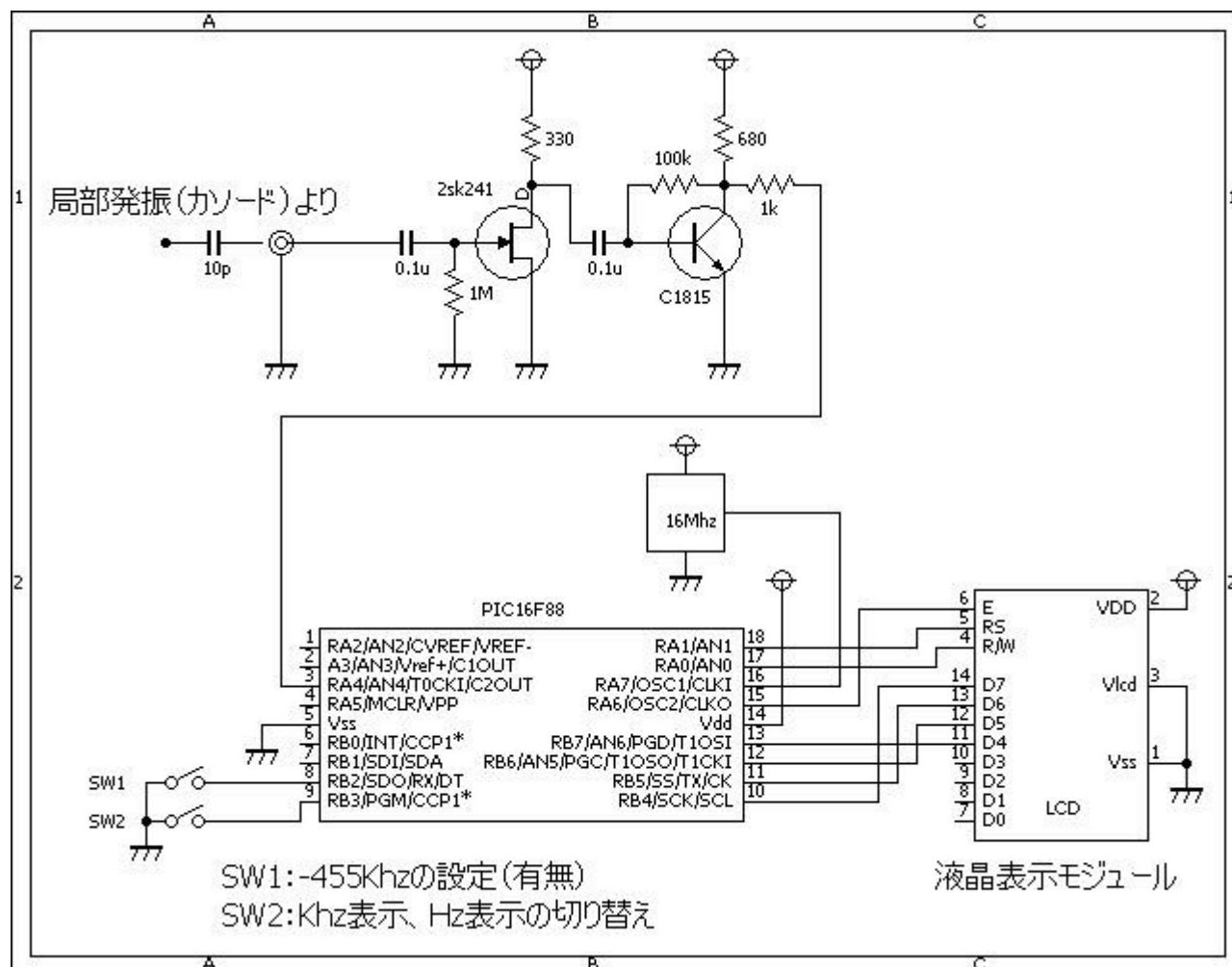


動作原理

標準的な5球スーパーラジオでは、前段の7極管で局部発振と混合を行っています。局部発振ではラジオ放送の周波数よりも中間周波数(455Khz)分だけ高い周波数で発振しています。そこでこの周波数をカウントし、その値から455Khzだけ引いてやればよいことになります。参考までに、標準的な5球スーパーへの本ユニットの接続例を回路図上に示します。



回路図



処理説明

まず、入力信号は、FET(2SK241)およびトランジスタ(2SC1815)で構成されたアンプで増幅されます。増幅された信号は、T0CKI端子に接続されTIMER0でカウントされます。TIMER1とCCP1を使って正確な0.1秒を得ます。(この間だけゲートを開けてカウントします)

- TIMER1を内部クロックで1/8プリスケアラで動作させます。
- するとTIMER1のクロックは、 $16\text{MHz} \div 4 \div 8 = 500\text{kHz}$ となります。
- CCP1をTIMER1とのコンペアモードで使い、一致する値を50000とします。
- すると10Hz周期で一致割り込みが発生し、正確な0.1秒を得ることが出来ます。

プリスケール値は1/8に設定してあります。これで最大約25Mhzまでカウントできますので短波を含む2バンド ラジオへも対応可能となります。 カウントした値から中間周波数(455Khz)分だけを差し引いた値を文字列に変換してLCDへ表示します。

- SW1:-455Khzの設定(有無)
- SW2:Khz表示⇔Hz表示の切り替え

ソースコード

RadioFC R1.21(Khz表示時に1Khz未満の値を切り捨てていました)

RadioFC2.c

```
/*
  <簡易周波数カウンター>

  □■sw1=1□-455Khz
  □■sw1=0□-0Khz
  □■sw2=1□Hz表示
  □■sw2=0□Khz表示

*/

//*****
*

#define sw1 PORTB.F2
#define sw2 PORTB.F3

//*****
*

static short endFlag;

//*****
*

void interrupt(){
    if (PIR1.CCP1IF == 1) {
        PIR1.CCP1IF = 0;
        //
        endFlag++;
        if (endFlag == 1) {
            TRISA.F4 = 0; // ゲートを閉める。
            PORTA.F4 = 0;
            T1CON.TMR1ON = 0; // TIMER1を停止する。
        }
    }
}

//*****
*

void main()
{
    static unsigned long freq; // 0...4294967295
    static unsigned char buf[20];
    static unsigned int ad3;
    static unsigned short cnt;
    // 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。
}
```

```
CMCON = 0b00000111;
// アナログの設定
ANSEL = 0b00001000;
// ポートの設定
TRISA = 0b10111100;
TRISB = 0b00001100;
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.INTE = 0;
INTCON.INTF = 0;
// CCPの設定
PIE1.CCP1IE = 1;
PIR1.CCP1IF = 0;
CCP1CON = 0b00001011;
CCPR1L = 0x50; // 0.1sec...10hz...クロックが16Mhzの時
CCPR1H = 0xC3; // 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 = 1;
OPTION_REG.T0SE = 0;
OPTION_REG.PSA = 0;
OPTION_REG.PS0 = 0;
OPTION_REG.PS1 = 1;
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 = 0;
// 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;
//
Lcd_Custom_Config(&PORTB,4,5,6,7,&PORTA,1,0,6);
Lcd_Custom_Cmd(LCD_CURSOR_OFF);
Lcd_Custom_Out(1, 1, "RadioFC");
Lcd_Custom_Out(2, 1, "R1.21");
Delay_ms(1000);
```

```
Lcd_Custom_Cmd(LCD_CLEAR);  
// 割り込み(全体)の設定  
INTCON.PEIE = 1;  
INTCON.GIE = 1;  
while (1) {  
    // 初期化  
    T1CON.TMR1ON = 0;  
    TRISA.F4 = 0;  
    PORTA.F4 = 0;  
    TMR0 = 0;  
    INTCON.T0IF = 0;  
    TMR1L = 0;  
    TMR1H = 0;  
    PIR1.TMR1IF = 0;  
    endFlag = 0;  
    freq = 0;  
    // 開始  
    T1CON.TMR1ON = 1;  
    Delay_Cyc(3);  
    asm    nop;  
    asm    nop;  
    asm    nop;  
    asm    nop;  
    TRISA.F4 = 1;    //ゲートを開ける。  
    // 測定  
    while (endFlag != 1) {  
        if (INTCON.T0IF == 1) {  
            INTCON.T0IF = 0;  
            freq++;  
        }  
    }  
    freq *= 256;  
    freq += TMR0;  
    // 補正  
    freq *= 8;  
    freq *= 10;  
    if (sw1 == 1)  
        freq -= 455000;    // -455Khz  
    // 表示  
    LongToStr(freq, buf);  
    if (sw2 == 1) {  
        Lcd_Custom_Out(1, 1, &buf[3]);  
        Lcd_Custom_Out(2, 1, "Hz");  
    } else {  
        buf[8] = 'K';  
        buf[9] = 'h';  
        buf[10] = 'z';  
        buf[11] = 0x00;  
        Lcd_Custom_Out(1, 1, &buf[3]);  
        Lcd_Custom_Out(2, 1, " ");  
    }
```

```

    }
    //
/*
    ad3 = Adc_Read(3);
    ad3 = ad3 / 80;
    for (cnt = 0; cnt < ad3; cnt++)
        Lcd_Custom_Chr(2, cnt + 1, 0xFF);
    for (; cnt < 8; cnt++)
        Lcd_Custom_Chr(2, cnt + 1, ' ');
    //
*/
    Delay_ms(500);
}
}

//*****
*
```

RadioFC R1.22(Khz表示時に1Khz未満の値を四捨五入しました)

RadioFC3.c

```

/*
    <簡易周波数カウンター>

    ■sw1=1 - 455Khz
    ■sw1=0 - 0Khz
    ■sw2=1 Hz表示
    ■sw2=0 Khz表示

    <改版履歴>
    ■20070507kHz時に四捨五入機能を追加

*/

//*****
*

#define    sw1    PORTB.F2
#define    sw2    PORTB.F3

//*****
*

static short    endFlag;

//*****
*

void    interrupt()
```

```
{
    if (PIR1.CCP1IF == 1) {
        PIR1.CCP1IF = 0;
        //
        endFlag++;
        if (endFlag == 1) {
            TRISA.F4 = 0;           // ゲートを閉める。
            PORTA.F4 = 0;
            T1CON.TMR1ON = 0;      // TIMER1を停止する。
        }
    }
}

//*****
*

void main()
{
    static unsigned long freq, temp;    // 0...4294967295
    static unsigned char buf[20];
    static unsigned int ad3;
    static unsigned short cnt;
    // 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。
    CMCON = 0b00000111;
    // アナログの設定
    ANSEL = 0b00001000;
    // ポートの設定
    TRISA = 0b10111100;
    TRISB = 0b00001100;
    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.INTE = 0;
    INTCON.INTF = 0;
    // CCPの設定
    PIE1.CCP1IE = 1;
    PIR1.CCP1IF = 0;
    CCP1CON = 0b00001011;
    CCPR1L = 0x50;    // 0.1sec...10hz...クロックが16Mhzの時
    CCPR1H = 0xC3;    // 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 = 1;
```

```
OPTION_REG.T0SE = 0;
OPTION_REG.PSA = 0;
OPTION_REG.PS0 = 0;
OPTION_REG.PS1 = 1;
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 = 0;
// 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;
//
Lcd_Custom_Config(&PORTB,4,5,6,7,&PORTA,1,0,6);
Lcd_Custom_Cmd(LCD_CURSOR_OFF);
Lcd_Custom_Out(1, 1, "RadioFC");
Lcd_Custom_Out(2, 1, "R1.22");
Delay_ms(1000);
Lcd_Custom_Cmd(LCD_CLEAR);
// 割り込み(全体)の設定
INTCON.PEIE = 1;
INTCON.GIE = 1;
while (1) {
    // 初期化
    T1CON.TMR1ON = 0;
    TRISA.F4 = 0;
    PORTA.F4 = 0;
    TMR0 = 0;
    INTCON.T0IF = 0;
    TMR1L = 0;
    TMR1H = 0;
    PIR1.TMR1IF = 0;
    endFlag = 0;
    freq = 0;
    // 開始
    T1CON.TMR1ON = 1;
    Delay_Cyc(3);
    asm    nop;
    asm    nop;
    asm    nop;
    asm    nop;
    TRISA.F4 = 1;    //ゲートを開ける。
    // 測定
```

```
while (endFlag != 1) {
    if (INTCON.T0IF == 1) {
        INTCON.T0IF = 0;
        freq++;
    }
}
freq *= 256;
freq += TMR0;
// 補正
freq *= 8;
freq *= 10;
if (sw1 == 1)
    freq -= 455000;    // -455Khz
// 表示
LongToStr(freq, buf);
if (sw2 == 1) {
    Lcd_Custom_Out(1, 1, &buf[3]);
    Lcd_Custom_Out(2, 1, "Hz");
} else {
    if (buf[8] >= '5') {
        buf[8] = 0x00;
        temp = atol(&buf[3]);
        temp++;
    } else {
        buf[8] = 0x00;
        temp = atol(&buf[3]);
    }
    LongToStr(temp * 1000, buf);
    buf[8] = 'k';
    buf[9] = 'H';
    buf[10] = 'z';
    buf[11] = 0x00;
    Lcd_Custom_Out(1, 1, &buf[3]);
    Lcd_Custom_Out(2, 1, " ");
}
//
/*
    ad3 = Adc_Read(3);
    ad3 = ad3 / 80;
    for (cnt = 0; cnt < ad3; cnt++)
        Lcd_Custom_Chr(2, cnt + 1, 0xFF);
    for (; cnt < 8; cnt++)
        Lcd_Custom_Chr(2, cnt + 1, ' ');
//
*/
Delay_ms(500);
}
}
```

//*****

*

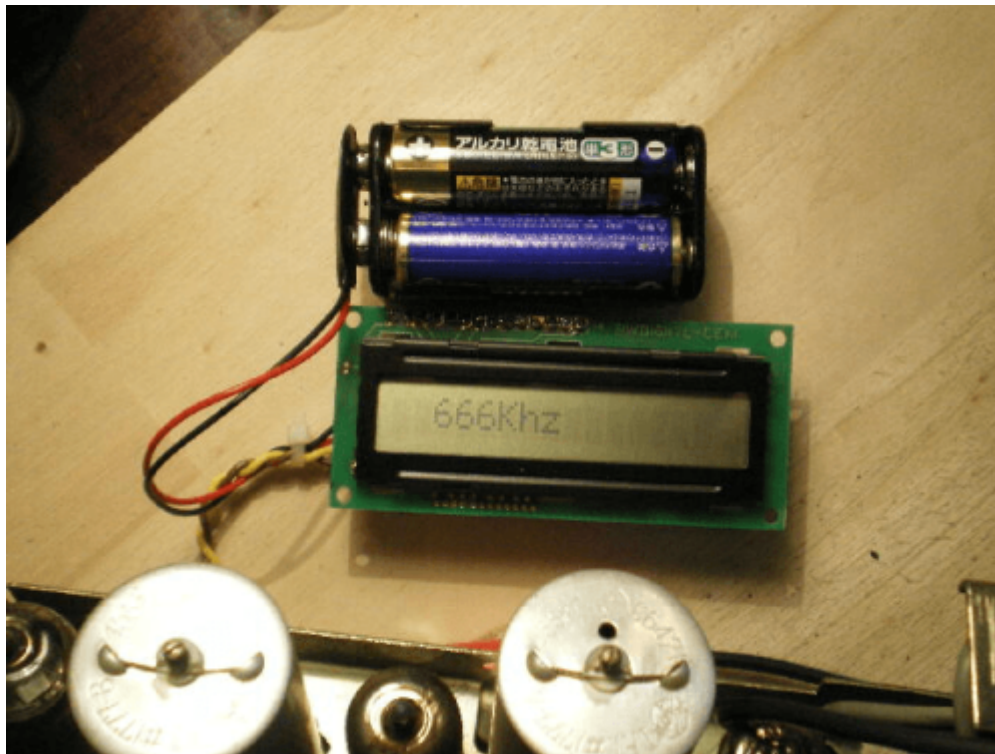
動作確認

ナショナルの真空管ラジオ(BM-400)に接続して動作を確認しました。 ケースが壊れたので裸のまま使

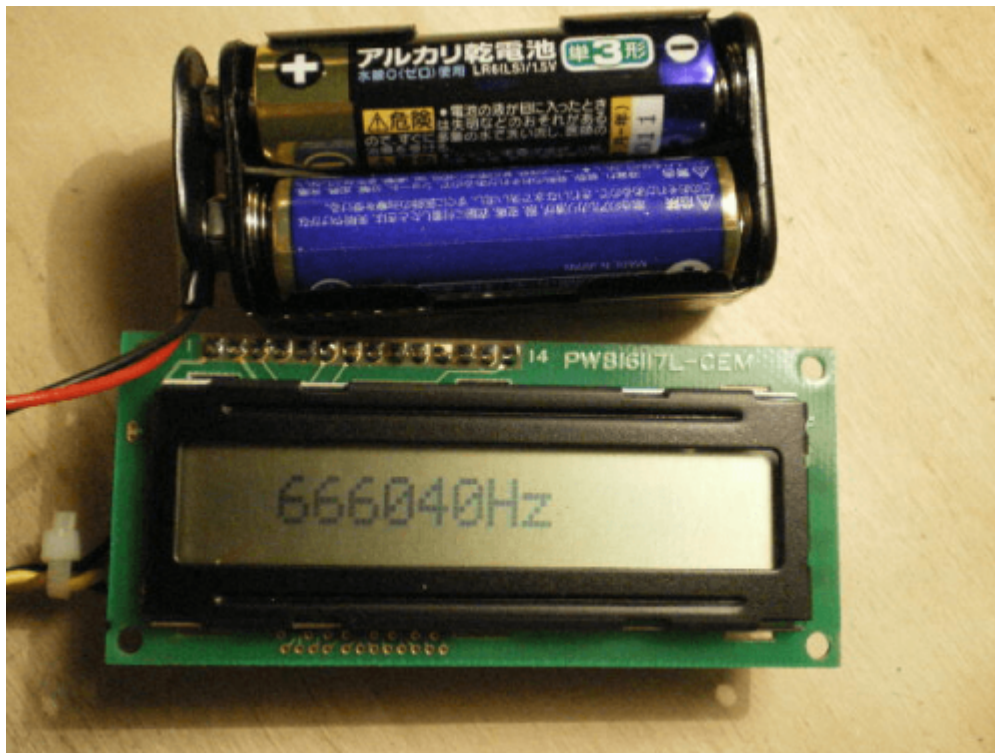


用しています。
放送(666Khz)を受信しているところです。

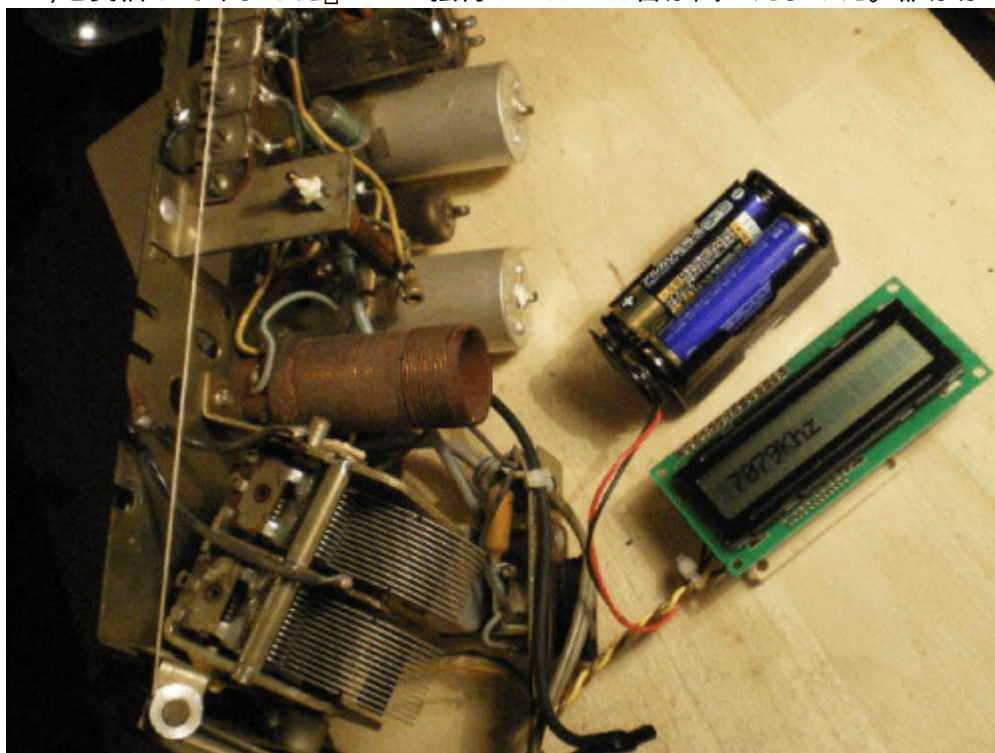
NHK第一



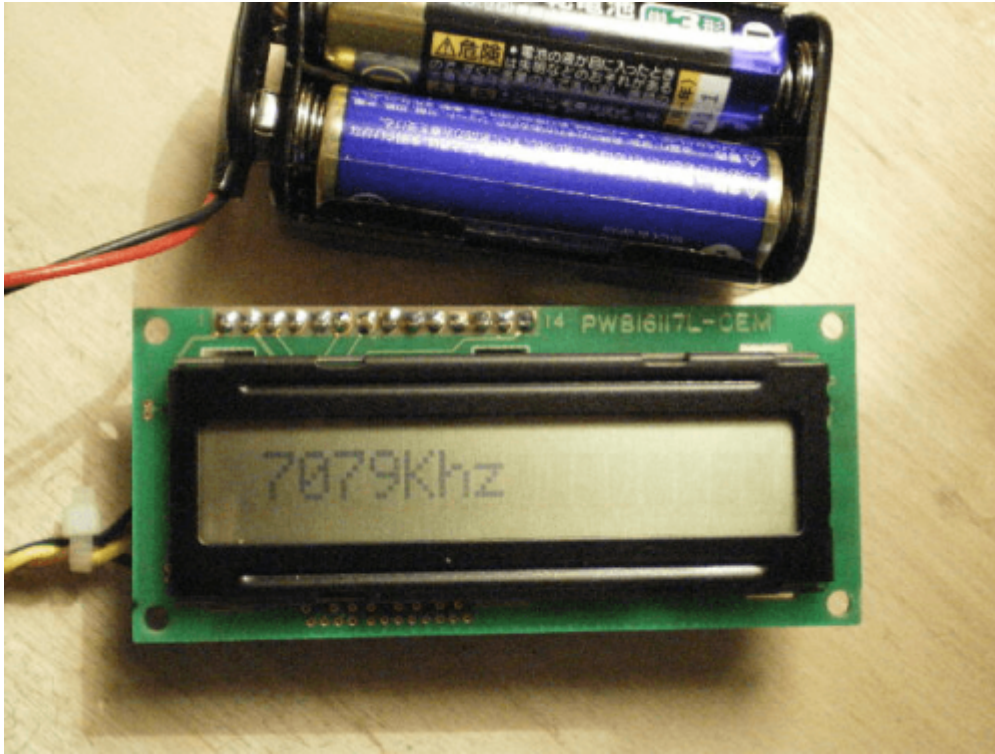
Hz単位で表示してみまし



た。アマチュア無線バンド(7Mhz)を受信してみました□SSBの独特のモガモガ音が聞こえました。誰かがCQを出しているようで



す。



From:

<http://deepsky.jp/wiki/> - うごくといいな

Permanent link:

<http://deepsky.jp/wiki/doku.php?id=elechobby:picdic:pic16f88:48&rev=1588144629>

Last update: **2025/10/17 14:28**

