

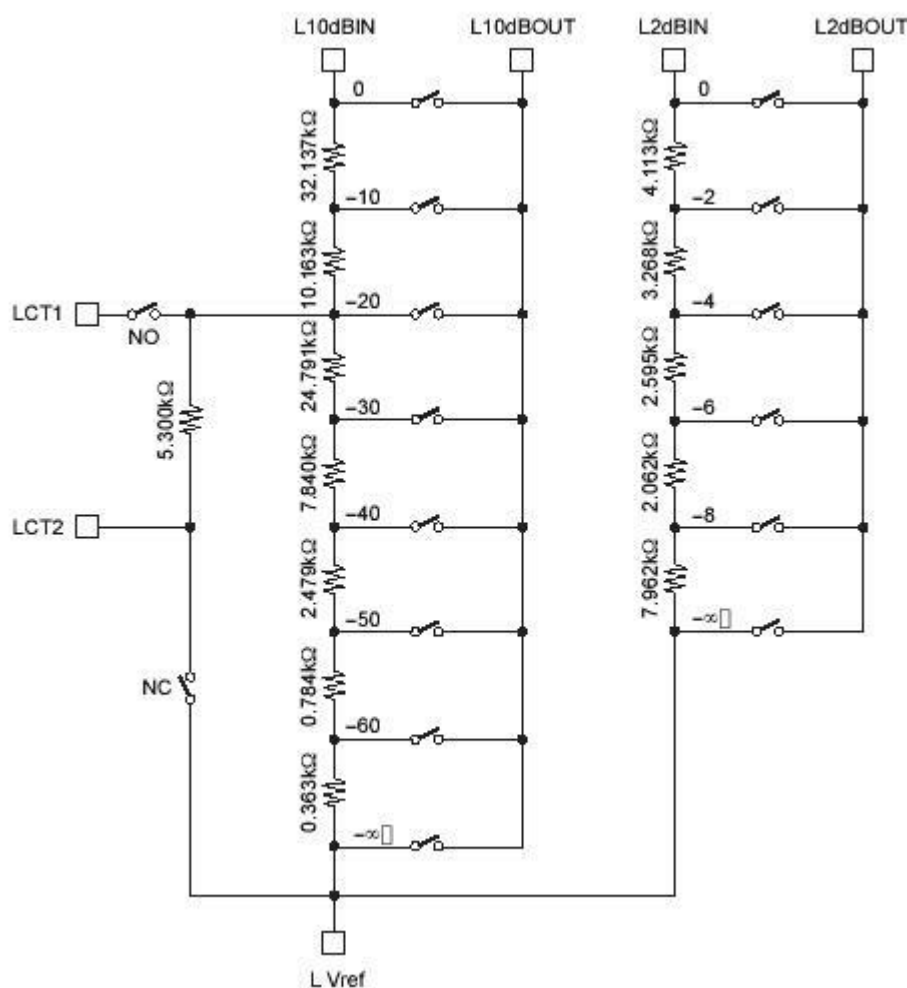
電子ボリューム

概要

SANYOの2チャンネル用電子ボリューム□LC75366□をPICで制御してみました。 千石電商で250円で販売していました□ LC75366は、ボリューム、 バランス、 ラウドネスの各機能をシリアルデータ入力によりコントロールできる 電子ボリュームです□<0dB □-68dB(2dB ステップ)、 36 ポジション>

動作原理

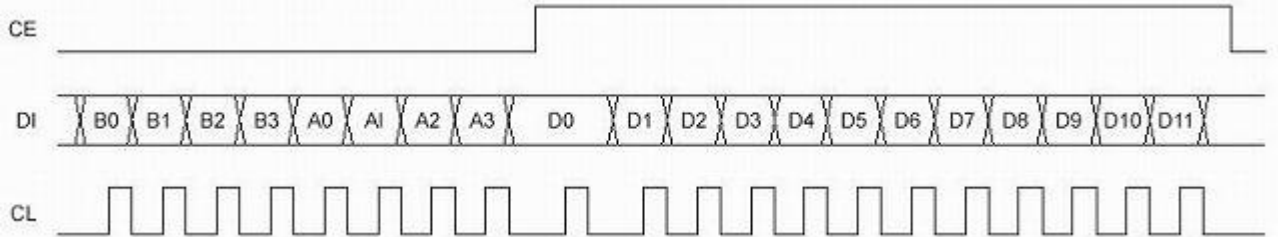
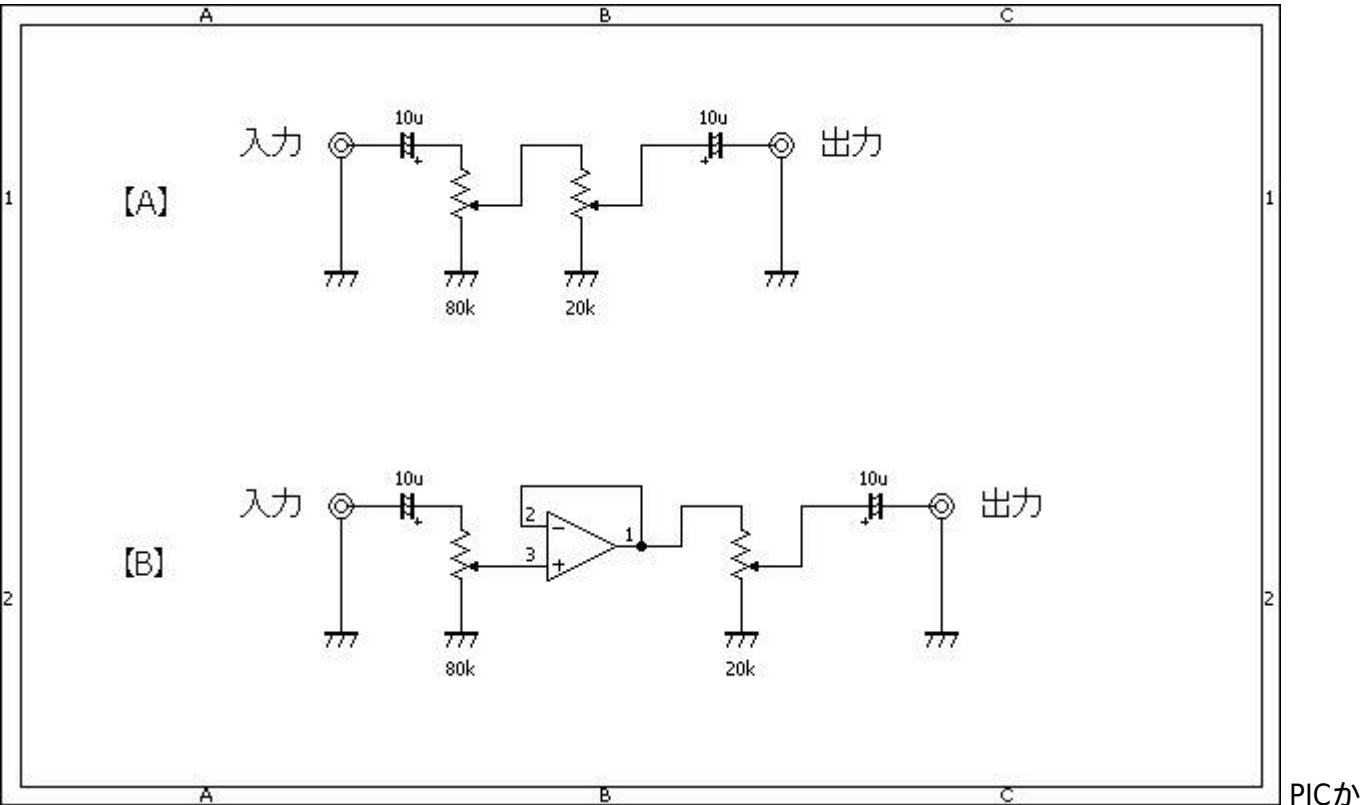
PICにより□10dB□2dBのステップを制御（プッシュSWによりUP/DOWN□します□L10dBのボリュームは、約80k□L2dBのボリュームは、約20kです。



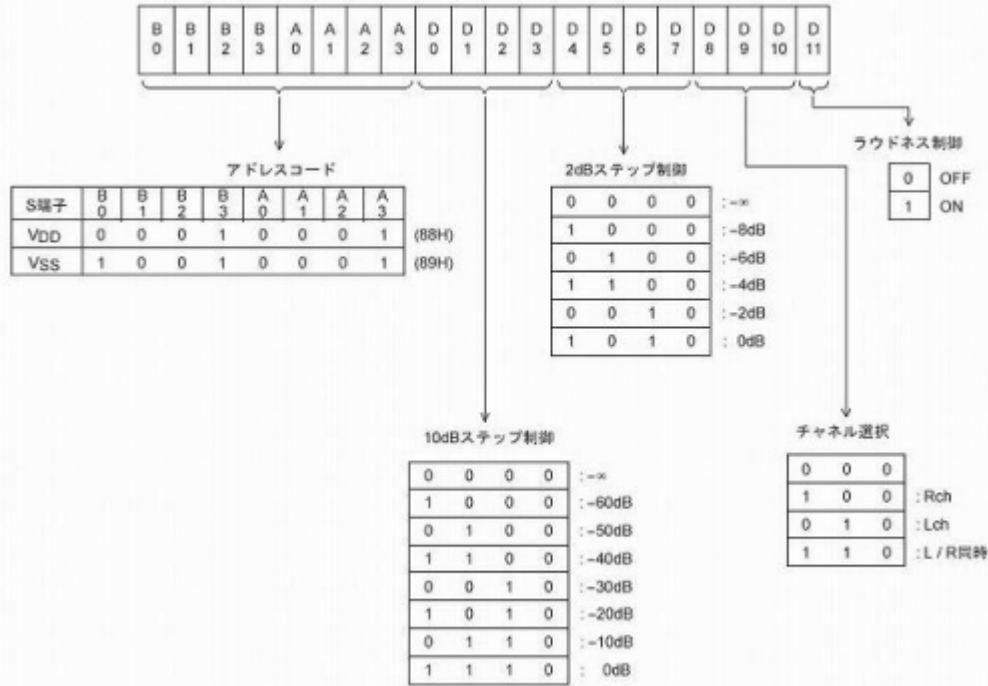
これは下図のような回路に概ね

等しい事になります。

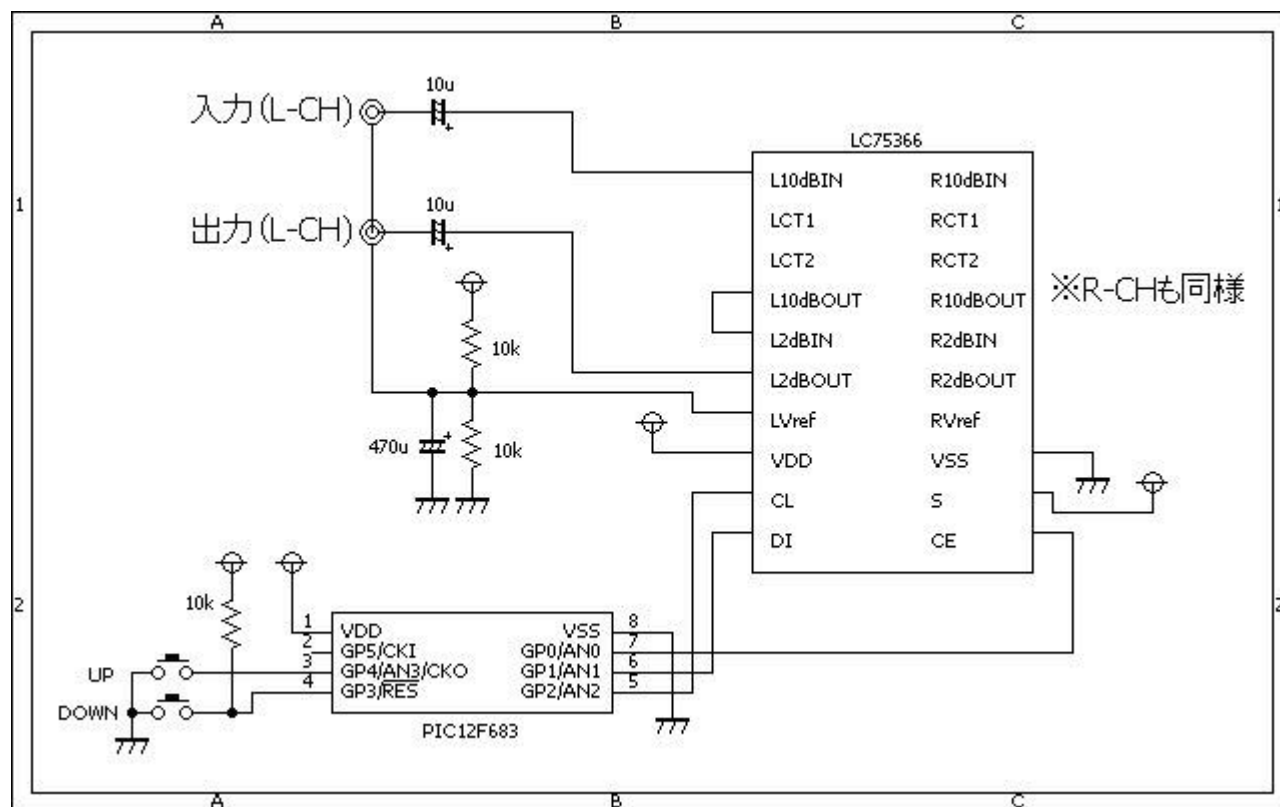
- A□は、L10dBOUTとL2dBINを直接に接続した場合です。
- B□は、L10dBOUTとL2dBINの間にバッファを挿入した場合です。（本来はこれがお勧めです）



このICは、同一バス上に2個接続できます。その個々の認識はアドレスコードにより行います。ラウドネスのON/OFFも出来ませんが、今回はOFFとして使用しました。Rch、Lchは同時に制御することも、個別に制御することも出来ます。これにより左右のバランスを調整することが出来ます。



回路図



ソースコード

lc75366.c

```
//*****
*

#define      CE      GPIO.F0
#define      DI      GPIO.F1
#define      CL      GPIO.F2

#define      swUp     GPIO.F3
#define      swDown   GPIO.F4

#define      LED      GPIO.F5

//*****
*

void interrupt()
{
    if (PIR1.TMR1IF == 1) {
        PIR1.TMR1IF = 0;
        LED = ~LED;
    }
}
```

```
    }  
}  
  
//*****  
*  
  
void    dataOutput(unsigned char di)  
{  
    if (di == 1)  
    {  
        DI = 1;  
        CL = 1;  
        asm    nop;  
        asm    nop;  
        CL = 0;  
    } else {  
        DI = 0;  
        CL = 1;  
        asm    nop;  
        asm    nop;  
        CL = 0;  
    }  
}  
  
void    volumeDataSet(unsigned char vd)  
{  
    unsigned    char    i;  
    CE = 0;  
    dataOutput(0);  
    dataOutput(0);  
    dataOutput(0);  
    dataOutput(1);  
    dataOutput(0);  
    dataOutput(0);  
    dataOutput(0);  
    dataOutput(1);  
    //  
    CE = 1;  
    for (i = 0; i < 8; i++) {  
        if (((vd >> i) & 0x01) == 1)  
            dataOutput(1);  
        else  
            dataOutput(0);  
    }  
    dataOutput(1);  
    dataOutput(1);  
    dataOutput(0);  
    dataOutput(0);  
    CE = 0;  
}
```

```
//*****  
*  
void main()  
{  
    static    unsigned    int        ad0;  
    static    unsigned    char       volumeCnt;  
    static    unsigned    char       volumeData[36] = {  
        0b00000000,  
  
        0b00010001,        // -68db  
        0b00100001,        // -66db  
        0b00110001,        // -64db  
        0b01000001,        // -62db  
        0b01010001,        // -60db  
  
        0b00010010,        // -58db  
        0b00100010,        // -56db  
        0b00110010,        // -54db  
        0b01000010,        // -52db  
        0b01010010,        // -50db  
  
        0b00010011,        // -48db  
        0b00100011,        // -46db  
        0b00110011,        // -44db  
        0b01000011,        // -42db  
        0b01010011,        // -40db  
  
        0b00010100,        // -38db  
        0b00100100,        // -36db  
        0b00110100,        // -34db  
        0b01000100,        // -32db  
        0b01010100,        // -30db  
  
        0b00010101,        // -28db  
        0b00100101,        // -26db  
        0b00110101,        // -24db  
        0b01000101,        // -22db  
        0b01010101,        // -20db  
  
        0b00010110,        // -18db  
        0b00100110,        // -16db  
        0b00110110,        // -14db  
        0b01000110,        // -12db  
        0b01010110,        // -10db  
  
        0b00010111,        // -08db  
        0b00100111,        // -06db  
        0b00110111,        // -04db  
        0b01000111,        // -02db
```

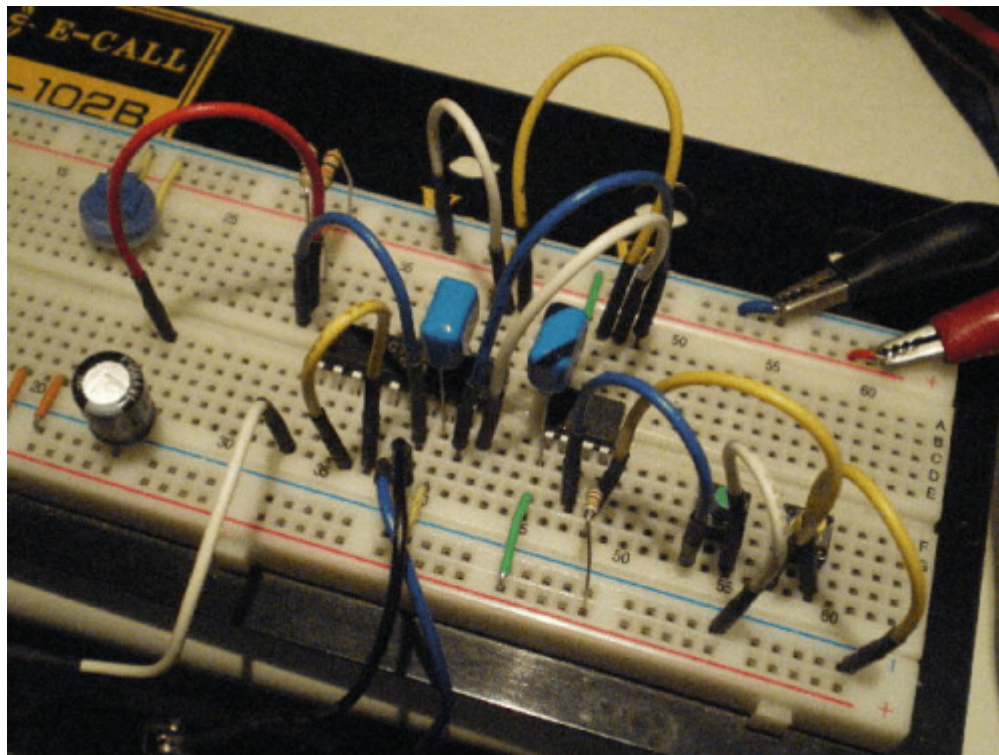
```

    0b01010111,          // -00db
};
//
OSCCON = 0b01110000;    // クロックは8Mhz
CMCON0 = 0b00000111;    // コンパレータは使用しない。
ANSEL = 0b00000000;     // A/Dを使用しない。
TRISIO = 0b00011000;
GPIO = 0b00000000;
OPTION_REG = 0b00000000;
PIE1.TMR1IE = 1;
PIR1.TMR1IF = 0;
T1CON = 0b00000001;
INTCON = 0b01000000;
T2CON.F0 = 0;
T2CON.F1 = 0;
//
WPU.F3 = 1;
WPU.F4 = 1;
//
volumeCnt = 25;          // -20db
volumeDataSet(volumeData[volumeCnt]);
//
INTCON.GIE = 1;          // これ以降の処理で割り込みを許可する。
//
while (1) {
    if (swDown == 0) {
        while (swDown == 0)
            Delay_ms(10);
        if (volumeCnt > 0)
            volumeCnt--;
        volumeDataSet(volumeData[volumeCnt]);
    }
    if (swUp == 0) {
        while (swUp == 0)
            Delay_ms(10);
        if (volumeCnt < 35)
            volumeCnt++;
        volumeDataSet(volumeData[volumeCnt]);
    }
}

//*****
*
```

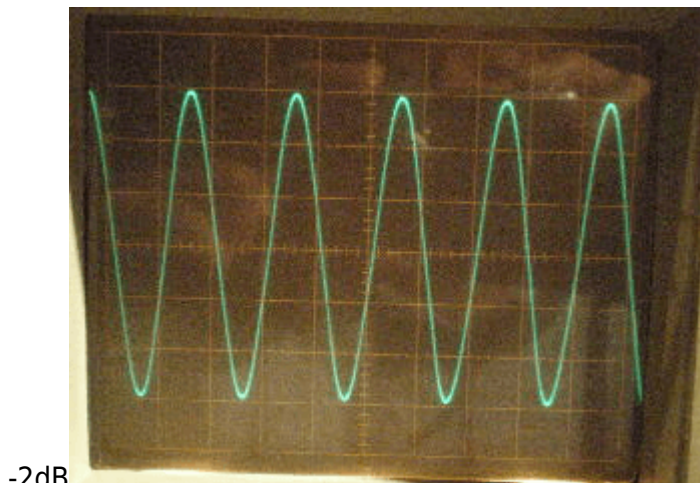
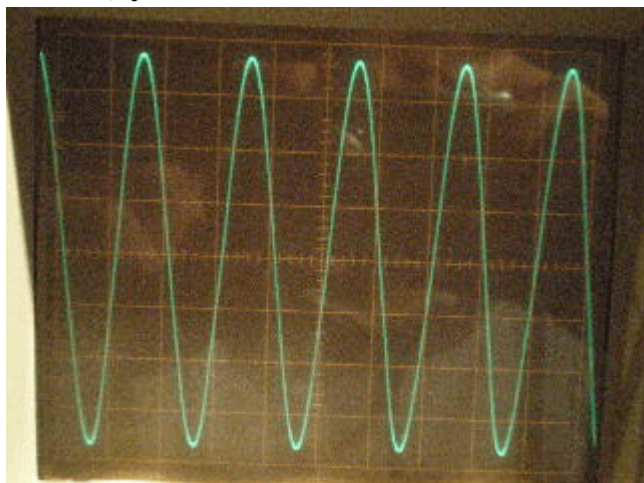
動作確認

いつものようにブレッドボードで動作確認です。見えにくいですが、緑色のSWがDOWN□黄色のSW

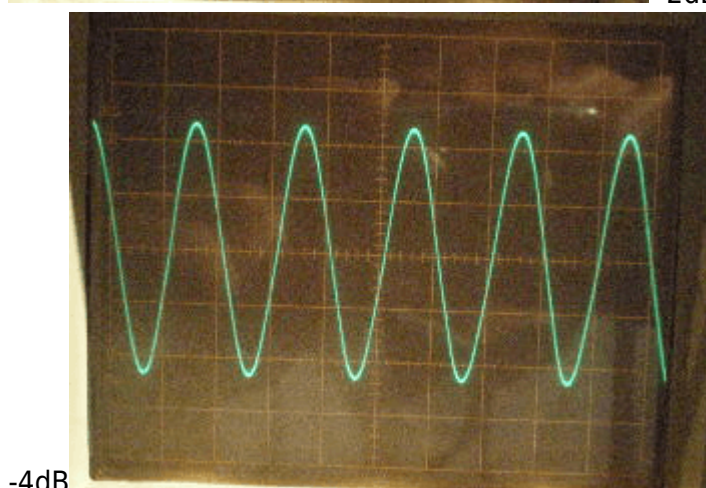


がUPです。

0dB



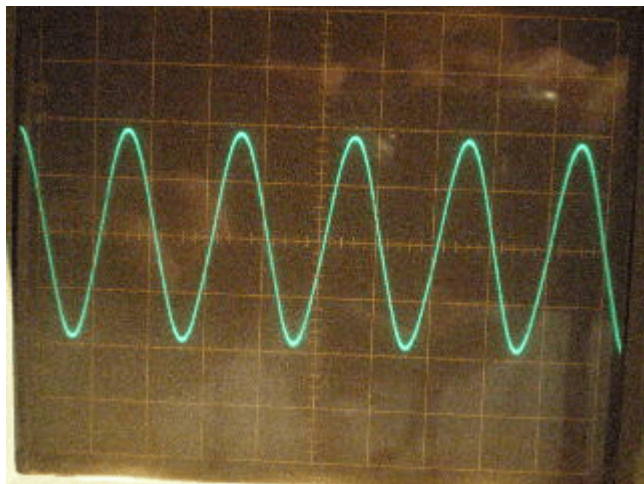
-2dB



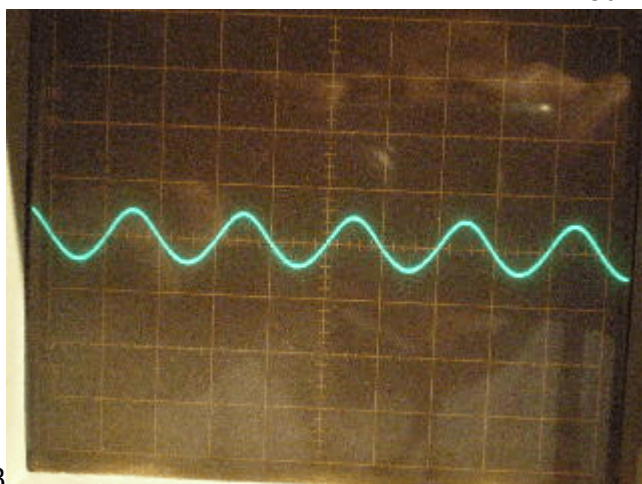
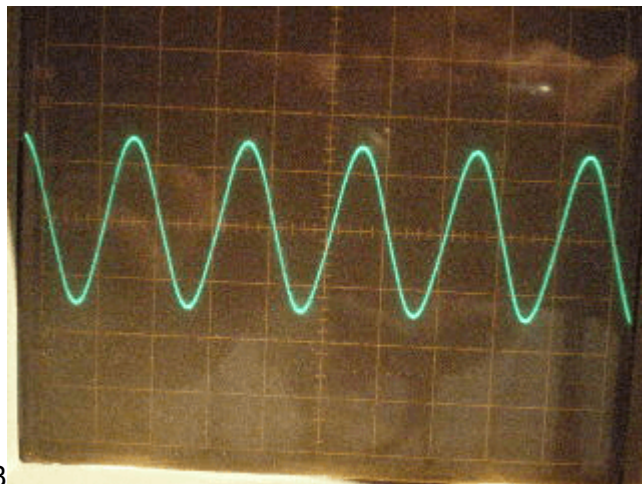
-4dB



-6dB



-8dB



-10dB

-10dBで極端に低下しているのはバッファを

省略したからです。つまり10dBのボリュームに2dBのボリューム(20kΩの負荷)が影響を与えてしまうためです。

外付け部品も少なく、価格も安いのでちょっとしたラジオ、アンプのボリュームには適当かと思います。

From:

<http://deepsky.jp/wiki/> - うごくといいな

Permanent link:

<http://deepsky.jp/wiki/doku.php?id=elechobby:picdic:pic12f683:6&rev=1588057535>

Last update: 2025/10/17 14:27

