

電圧&電流計V3(7セグ表示)(PIC16F785)

概要

今までに、電圧や電流を測定するために、

- 電圧計&電流計アダプタ。。。。液晶ディスプレイに、電圧値、電流値を表示
- ミニ電圧&電流計(7セグ表示)。。7セグ(3桁)に、電圧値、電流値をスイッチ切替で表示

を製作しました。今回は、7セグ(3桁)を2個使用し、電圧値、電流値をスイッチ切替無しで表示可能な電圧&電流計を製作しました。

動作原理

<A/D変換>

- 精度を高めるために、基準電圧には、シャントレギュレータを使用し、2.495Vとしました。

(分解能は、 $2.4\text{mV} \approx 2.495\text{V} \div 1024$)

<7セグ(3桁×2個)表示>

- 1msec周期の割り込み処理の中で、1桁分を表示します。従って6桁であれば $\approx 6\text{msec}$ で表示することにな

ります。 ダイナミック点灯の基本的な原理は、ミニ周波数カウンタ(kHz表示)を参照してください。

<電圧の測定>

- 抵抗R1とR2で、1/11に分圧した電圧(V_o)を測定します。

$V = V_o \times 11$ 倍

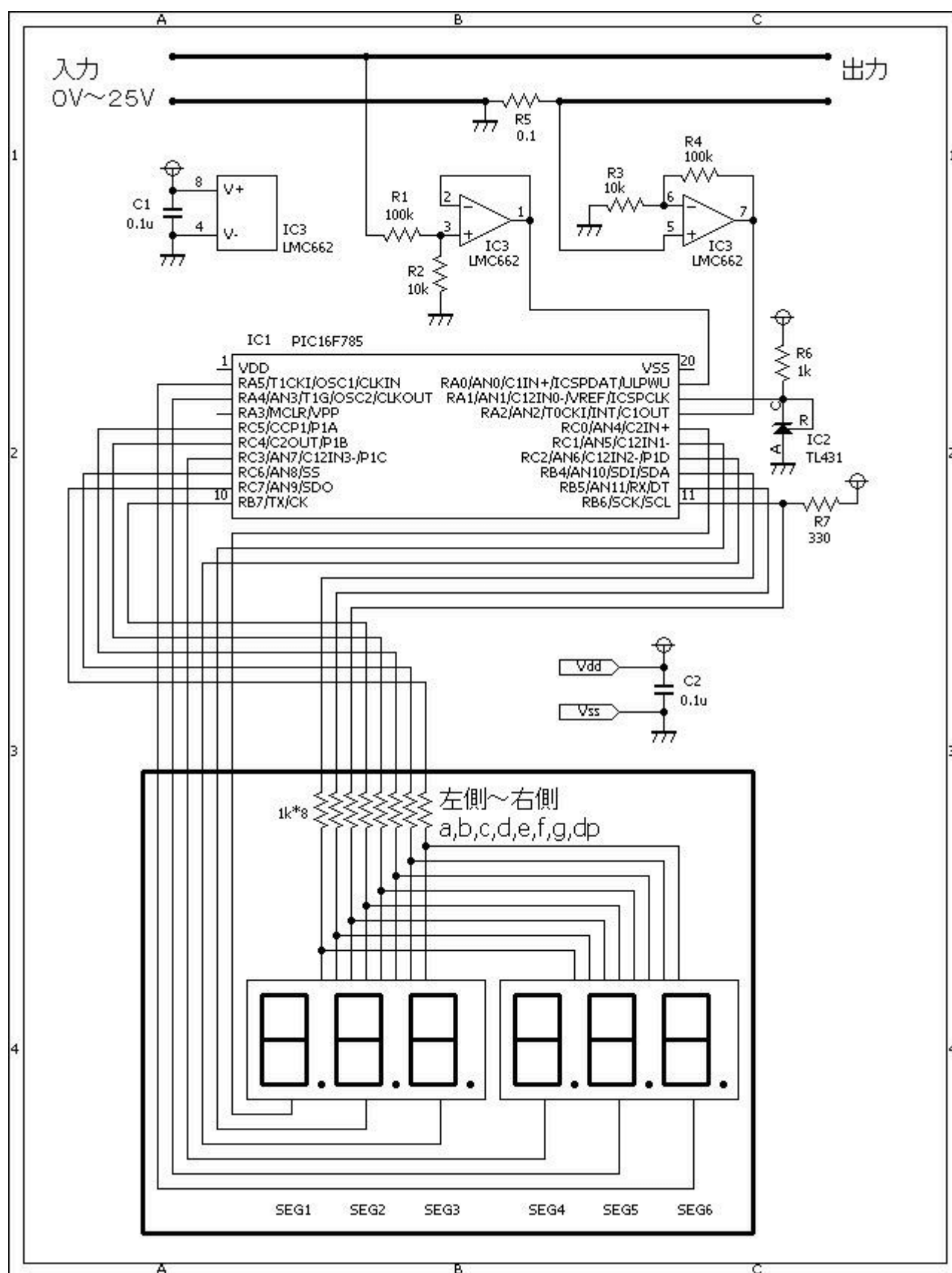
- 分圧の精度を高めるために、オペアンプによるボルテージフォロア(高インピーダンス)で測定します。
- 測定単位を、100mVに設定したので $\approx 100\text{mV}$ 以下は四捨五入しました。

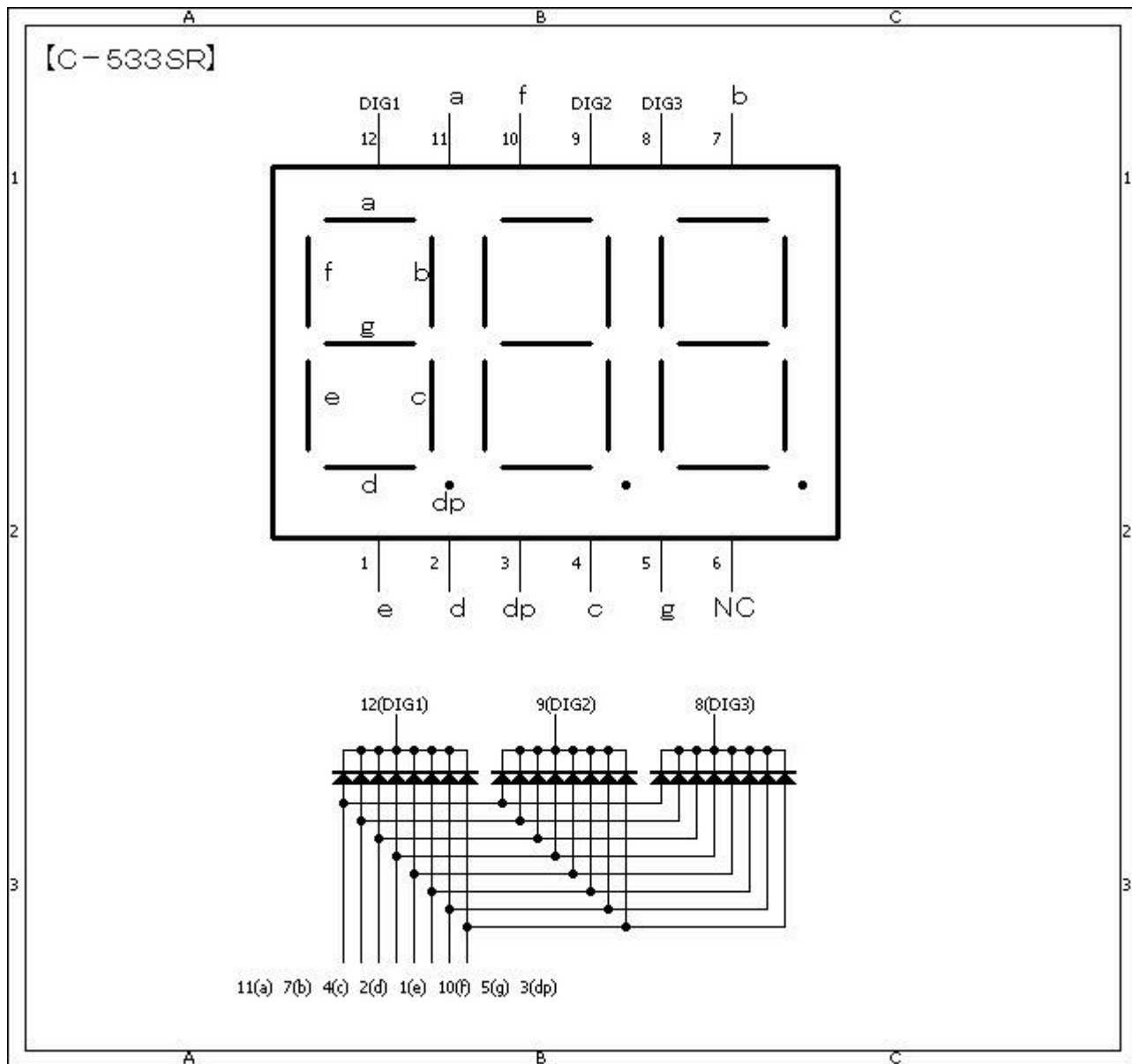
<電流の測定>

- R5の両端電圧(V_o)より電流を測定します。
 $I = V_o \div 0.1\Omega$
- 0.1Ω のワット数は、出来るだけ大きな値のものにして下さい。
- V_o は、とても小さな値になる(例えば $\approx 100\text{mA}$ の電流で、10mV)ので、オペアンプで11倍に増幅して測定します。
- 測定単位を、10mAに設定したので $\approx 10\text{mA}$ 以下は四捨五入しました。

注意事項 RB6はオープンドレインなので、抵抗(330 Ω 前後)でプルアップしてください。

回路図





ソースコード

[vi_meter_7seg_6disp.c](#)

```
//*****
*
/*
    <電圧電流計（7セグ表示）>
*/
//*****
*

#define SEG1 PORTC.F0
#define SEG2 PORTC.F1
#define SEG3 PORTC.F2
#define SEG4 PORTC.F3
#define SEG5 PORTA.F4
```

```
#define      SEG6      PORTA.F5

#define      SEG_a      PORTB.F4
#define      SEG_b      PORTB.F5
#define      SEG_c      PORTB.F6
#define      SEG_d      PORTB.F7
#define      SEG_e      PORTC.F4
#define      SEG_f      PORTC.F5
#define      SEG_g      PORTC.F6
#define      SEG_dp      PORTC.F7

#define      ON      0
#define      OFF      1

//*****
*

void  seg_data_set(char dat)
{
    SEG_a = 0;
    SEG_b = 0;
    SEG_c = 0;
    SEG_d = 0;
    SEG_e = 0;
    SEG_f = 0;
    SEG_g = 0;
    switch (dat) {
    case '0':
        SEG_a = 1;
        SEG_b = 1;
        SEG_c = 1;
        SEG_d = 1;
        SEG_e = 1;
        SEG_f = 1;
        break;
    case '1':
        SEG_b = 1;
        SEG_c = 1;
        break;
    case '2':
        SEG_a = 1;
        SEG_b = 1;
        SEG_d = 1;
        SEG_e = 1;
        SEG_g = 1;
        break;
    case '3':
        SEG_a = 1;
        SEG_b = 1;
        SEG_c = 1;
```

```
        SEG_d = 1;
        SEG_g = 1;
        break;
    case '4':
        SEG_b = 1;
        SEG_c = 1;
        SEG_f = 1;
        SEG_g = 1;
        break;
    case '5':
        SEG_a = 1;
        SEG_c = 1;
        SEG_d = 1;
        SEG_f = 1;
        SEG_g = 1;
        break;
    case '6':
        SEG_a = 1;
        SEG_c = 1;
        SEG_d = 1;
        SEG_e = 1;
        SEG_f = 1;
        SEG_g = 1;
        break;
    case '7':
        SEG_a = 1;
        SEG_b = 1;
        SEG_c = 1;
        SEG_f = 1;
        break;
    case '8':
        SEG_a = 1;
        SEG_b = 1;
        SEG_c = 1;
        SEG_d = 1;
        SEG_e = 1;
        SEG_f = 1;
        SEG_g = 1;
        break;
    case '9':
        SEG_a = 1;
        SEG_b = 1;
        SEG_c = 1;
        SEG_d = 1;
        SEG_f = 1;
        SEG_g = 1;
        break;
    case ' ':
        break;
}
}
```

```
//*****
*

short  seg_flg, data1, data2, data3, data4, data5, data6, dot1, dot2;

void  interrupt()
{
    if (PIR1.CCP1IF == 1) {
        PIR1.CCP1IF = 0;
        //□□□□□桁)点灯処理
        switch (seg_flg) {
            case 0:
                seg_flg = 1;
                SEG6 = OFF;
                seg_data_set(data1);
                SEG_dp = (dot1 == 1) ? 1 : 0;
                SEG1 = ON;
                break;
            case 1:
                seg_flg = 2;
                SEG1 = OFF;
                seg_data_set(data2);
                SEG_dp = (dot1 == 2) ? 1 : 0;
                SEG2 = ON;
                break;
            case 2:
                seg_flg = 3;
                SEG2 = OFF;
                seg_data_set(data3);
                SEG_dp = (dot1 == 3) ? 1 : 0;
                SEG3 = ON;
                break;
            case 3:
                seg_flg = 4;
                SEG3 = OFF;
                seg_data_set(data4);
                SEG_dp = (dot2 == 4) ? 1 : 0;
                SEG4 = ON;
                break;
            case 4:
                seg_flg = 5;
                SEG4 = OFF;
                seg_data_set(data5);
                SEG_dp = (dot2 == 5) ? 1 : 0;
                SEG5 = ON;
                break;
            case 5:
                seg_flg = 0;
                SEG5 = OFF;
```

```

        seg_data_set(data6);
        SEG_dp = (dot2 == 6) ? 1 : 0;
        SEG6 = ON;
        break;
    }
}

//*****
*

unsigned long measurement(unsigned short channel)
{
    static unsigned long dat;
    static unsigned int cnt;
    //
    dat = 0;
    for (cnt = 0; cnt < 1000; cnt++) {
        dat += Adc_Read(channel);
    }
    return (dat);
}

//*****
*

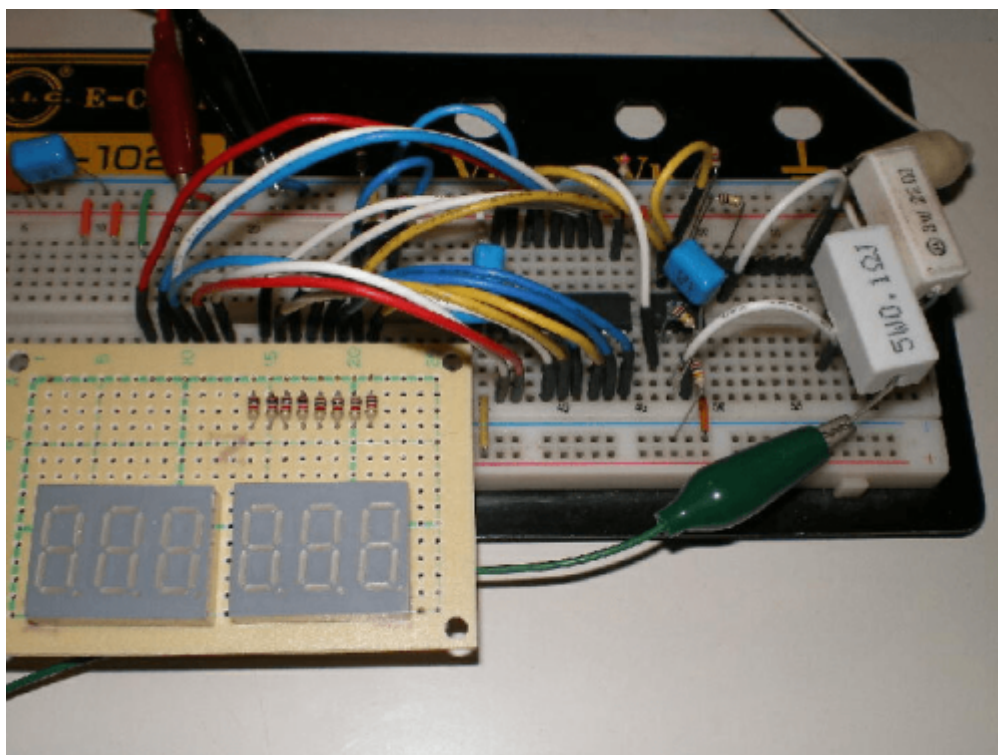
void main()
{
    static char buf[16];
    static double v1, v2, v, i;
    static int tmp;
    //
    TRISA = 0b00001111;
    TRISB = 0b00000000;
    TRISC = 0b00000000;
    OSCCON = 0b01100000; // クロックを4Mhzに設定する。
    ANSEL0 = 0b00000111; // □□□変換を使用する。
    ANSEL1 = 0b00000000; // □□□変換を使用する。
    ADCON0.VCFG = 1;
    // TIMER1の設定
    PIE1.TMR1IE = 0;
    PIR1.TMR1IF = 0;
    T1CON.T1CKPS0 = 1;
    T1CON.T1CKPS1 = 1;
    T1CON.TMR1ON = 0;
    TMR1L = 0;
    TMR1H = 0;
    // CCPの設定
    PIE1.CCP1IE = 1;
    PIR1.CCP1IF = 0;
    CCP1CON.CCP1M3 = 1;

```

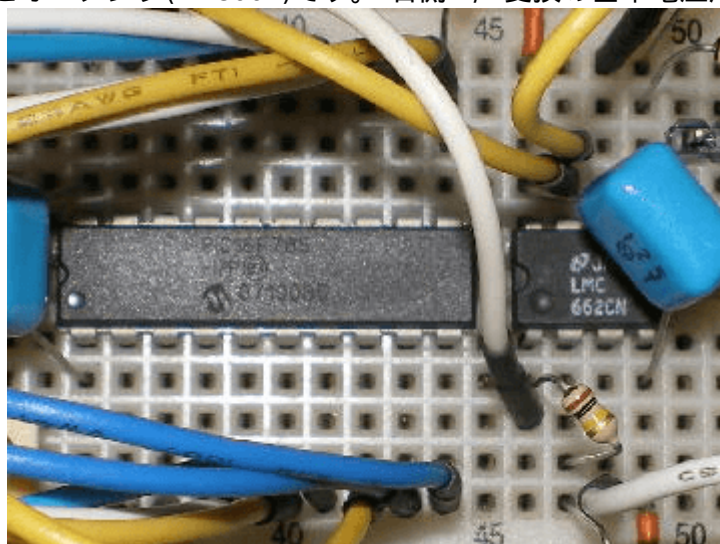
```
CCP1CON.CCP1M2 = 0;
CCP1CON.CCP1M1 = 1;
CCP1CON.CCP1M0 = 1;
CCPR1L = 0x7D;    // 1msec... (1÷4000000)*4*8*125
CCPR1H = 0x00;    //
//
SEG1 = OFF;
SEG2 = OFF;
SEG3 = OFF;
SEG4 = OFF;
SEG5 = OFF;
SEG6 = OFF;
seg_flg = 0;
data1 = ' ';
data2 = ' ';
data3 = ' ';
data4 = ' ';
data5 = ' ';
data6 = ' ';
dot1 = 2;
dot2 = 4;
// 割り込みを許可する。
INTCON.PEIE = 1;
INTCON.GIE = 1;
//
T1CON.TMR1ON = 1;
//
while (1) {
    //電流の測定と表示
    v1 = measurement(2);
    v1 = ((v1 / 1000.0) * 2.4365234375) / 11.0;
//2.495V/1024...TL431
    i = v1 / 0.1;
    tmp = i / 10.0;
    if ((i - (tmp * 10)) >= 5) {
        tmp++;
    }
    IntToStr(tmp, buf);
    data4 = buf[3];
    data5 = buf[4];
    data6 = buf[5];
    //電圧の測定と表示
    v2 = measurement(0);
    v2 = ((v2 / 1000.0) * 2.4365234375) * 11.0;
//2.495V/1024...TL431
    v = v2 - v1;
    tmp = v / 100.0;
    if ((v - (tmp * 100)) >= 50) {
        tmp++;
    }
}
```

```
    IntToStr(tmp, buf);  
    data1 = buf[3];  
    data2 = buf[4];  
    data3 = buf[5];  
}  
}  
  
//*****  
*
```

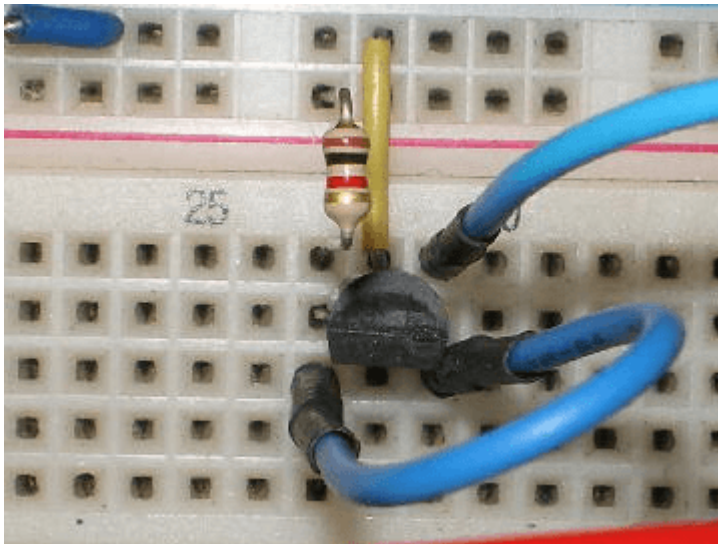
動作確認



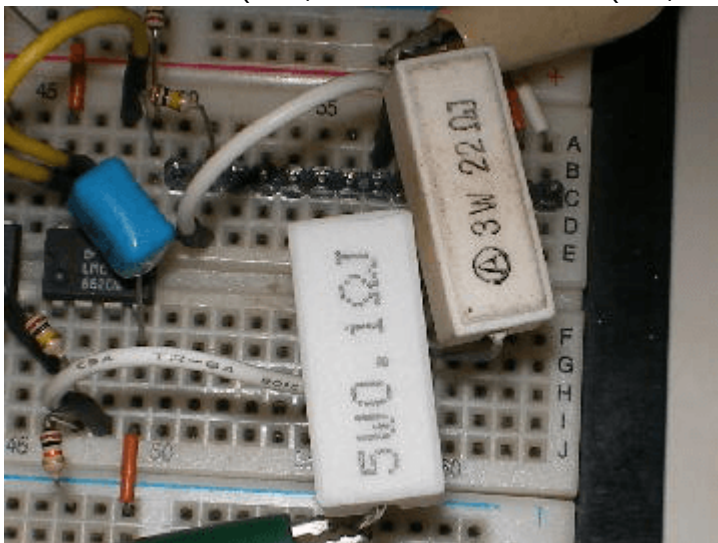
左側:PIC16F785とオペアンプ(LMC662)です。 右側:A/D変換の基準電圧用のシャントレギュレー



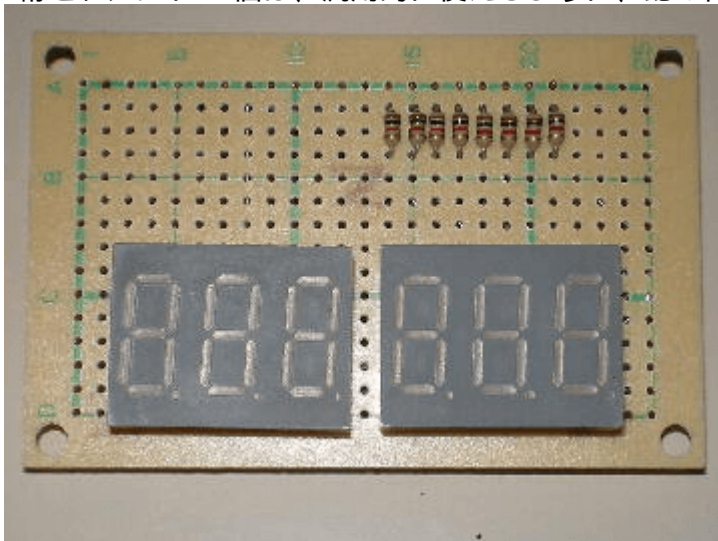
タ(TL431)です。

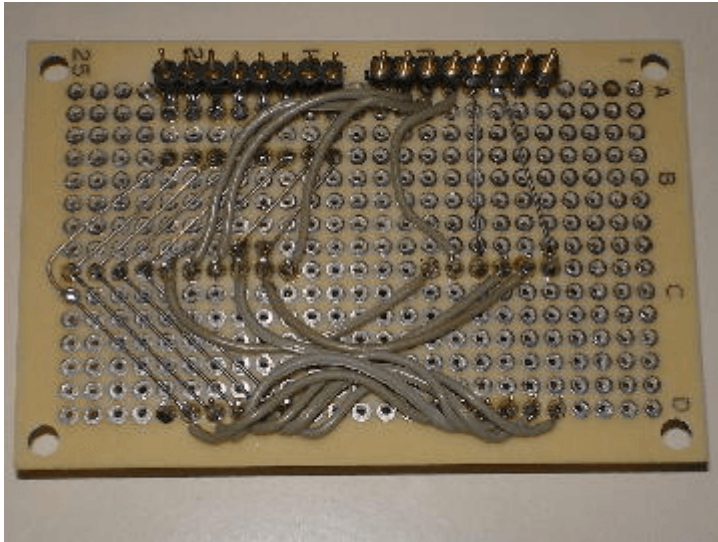


電流検出用の抵抗(0.1Ω)とテスト用の負荷抵抗(22Ω)です。

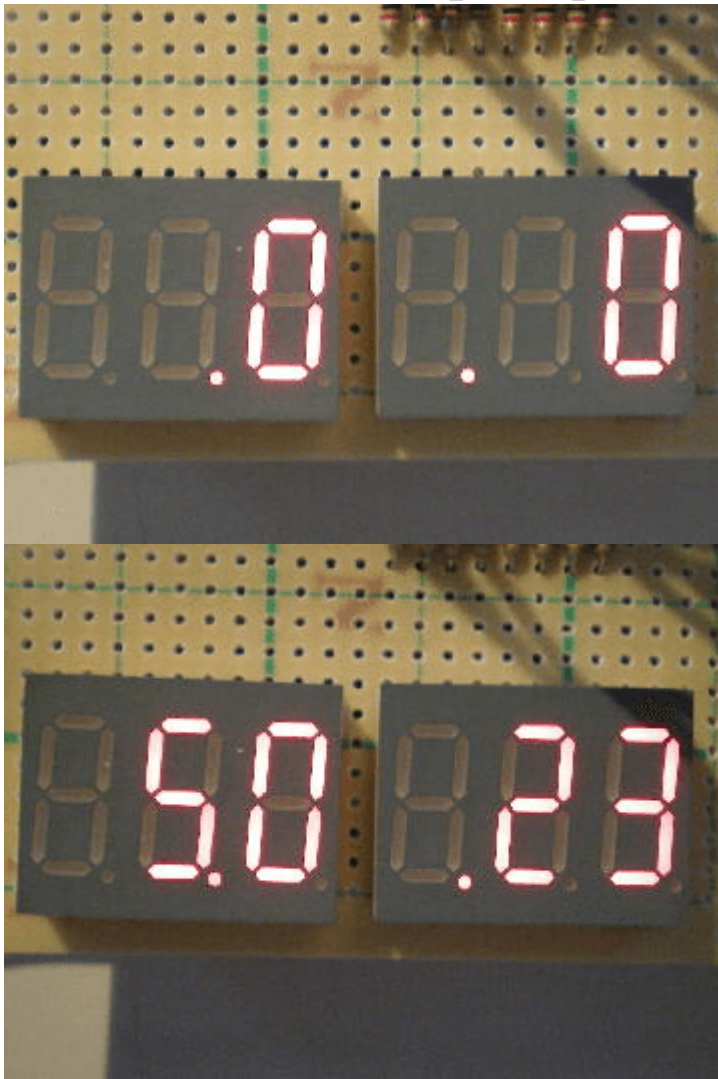


3桁セグメント×2個は、汎用的に使えるように、蛇の目基板に実装しました。

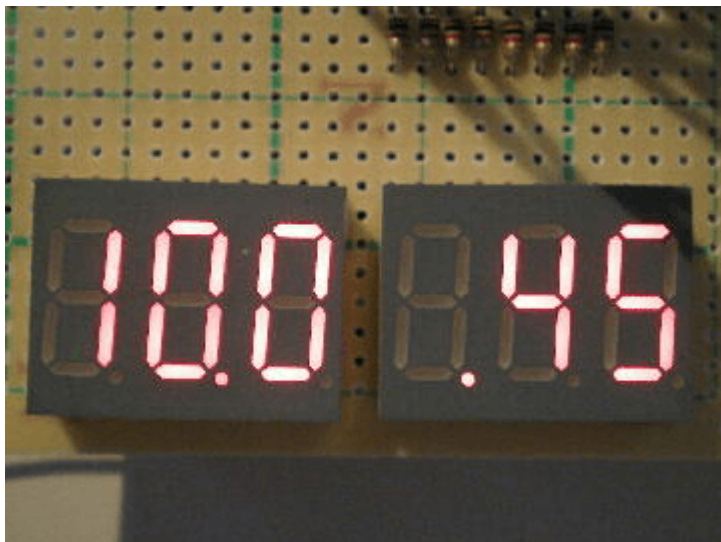




左側:入力電圧0V 右側:入力電圧5V□負荷22Ω□電流230mA(計算値227mA)



左側:入力電圧10V□負荷22Ω□電流450mA(計算値455mA) 右側:入力電圧15V□負荷22Ω□電流680mA(計算



値682mA)



From:

<http://deepsky.jp/wiki/> - うごくといいな

Permanent link:

<http://deepsky.jp/wiki/doku.php?id=elechobby:picdic:otherpic:184&rev=1588246223>

Last update: **2025/10/17 14:27**

