

電圧計&電流計アダプタV2(バー表示)

概要

以前に製作した電圧計&電流計アダプタをご覧になった、ロシアの方から、電圧及び電流の測定範囲を拡張してほしいとの要望があり製作しました。

<電圧計&電流計アダプタの仕様>

- 電圧測定範囲=0V~25V(100mV単位)
- 電流測定範囲=0mA~2000mA(1mA単位)
- 電流表示のみバー表示(16点バー)

<電圧計&電流計アダプタV2の仕様>

- 電圧測定範囲=0V~55V(100mV単位)
- 電流測定範囲=0A~4.5A(10mA単位)
- 電圧および電流ともにバー表示(50点バー)

動作原理

PICの電源(+5V)よりも高い電圧を測定したり、電流ロスを出れるだけ低く抑えるために、工夫してあります。

- 高電圧の測定(抵抗による分圧方式を採用)
- 電流ロスの軽減(低抵抗+オペアンプによる増幅方式の採用)

またLCDへの表示も、デジタル的な数値表示だけでなく、アナログ的なバー表示を加え、より視覚的に判断できるようにしました。

動作原理(ハードウェア)

電圧検出 100k Ω と10k Ω による分圧を行い、オペアンプのボルテージフォロアでインピーダンス変換し、PICのA/D変換の入力とします。

電流検出 0.1 Ω の電流検出抵抗を使用し、その両端の電圧を、オペアンプで11倍に増幅し、PICのA/D変換の入力とします。

動作原理(ソフトウェア)

電圧の測定と表示

- A/D変換で1000回サンプリングしその平均値(V1)を求めます。
- V1に応じてバーを表示します。
- V1を換算し、出力電圧値を求めます。
- 出力電圧から電流検出抵抗の両端電圧を差し引きます。

- 差し引いた値を更に四捨五入し、最終電圧値とします。
- 最終電圧値をLCDに表示します。

電流の測定と表示

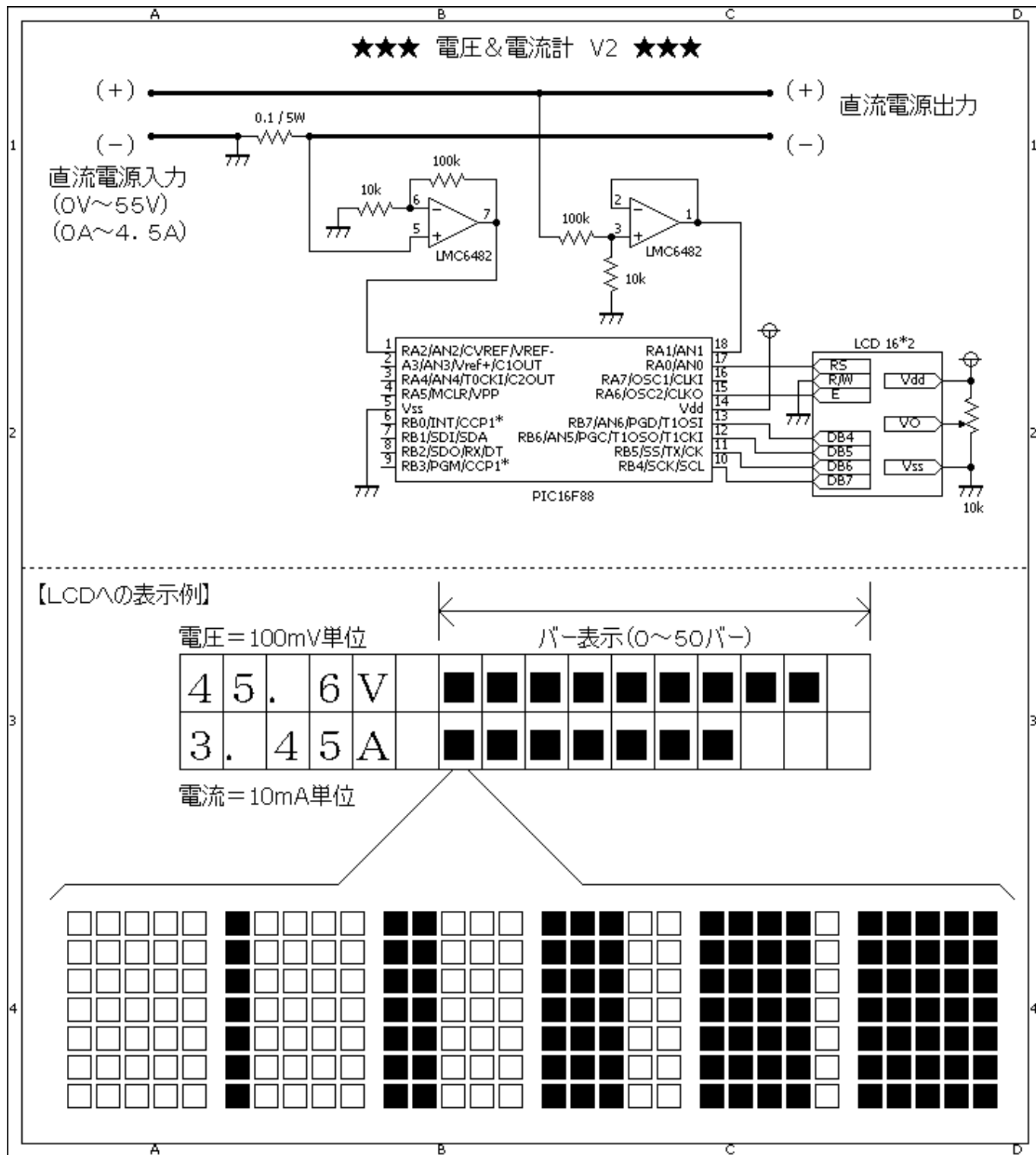
- A/D変換で1000回サンプリングしその平均値(V2)を求めます。
- V2に応じてバーを表示します。
- V2と電流検出抵抗値で、オームの法則より、電流値(I)を求めます。
- 電流値(I)を更に四捨五入し、最終電流値とします。
- 最終電流値をLCDに表示します。

バー表示のための前準備

- LCDのCGRAMに、バー表示のためのキャラクタデータを登録します。

※CGRAMの詳細な使い方については□CGRAM活用(キャラクタ表示LCD)を参照してください。

回路図



ソースコード

[vi_meter_v2.c](#)

```

//*****
*
/*
『電圧&電流計□□□バー表示』
*/

```

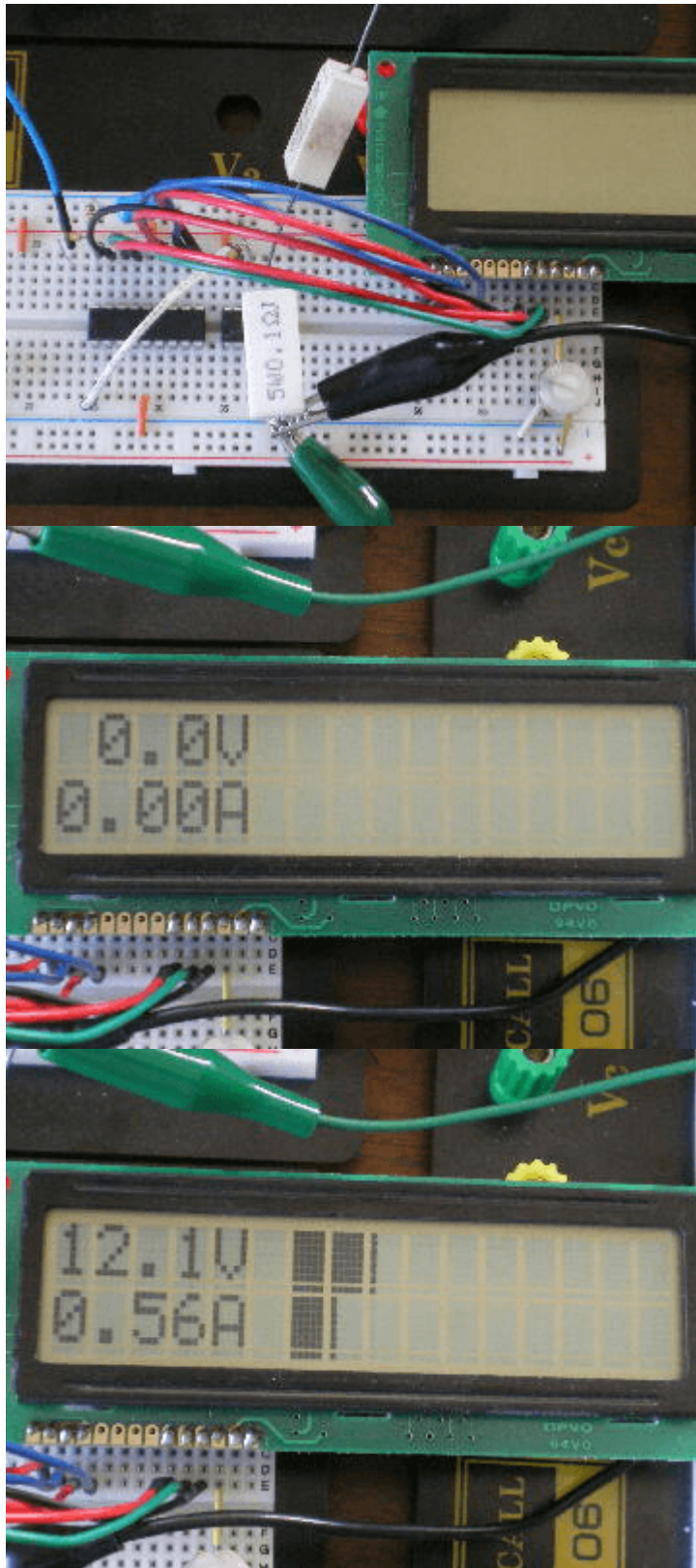
```
//*****  
*  
#define      BYTE      unsigned char  
#define      WORD      unsigned int  
#define      DWORD     unsigned long  
//*****  
*  
const char character0[] = { 0, 0, 0, 0, 0, 0, 0, 0};  
const char character1[] = {16,16,16,16,16,16,16,16};  
const char character2[] = {24,24,24,24,24,24,24,24};  
const char character3[] = {28,28,28,28,28,28,28,28};  
const char character4[] = {30,30,30,30,30,30,30,30};  
const char character5[] = {31,31,31,31,31,31,31,31};  
//  
void      RegistCustomChar()  
{  
    char    i;  
    //  
    LCD_Cmd(64);  
    for (i = 0; i<=7; i++) {  
        LCD_Chr_Cp(character0[i]);  
    }  
    for (i = 0; i<=7; i++) {  
        LCD_Chr_Cp(character1[i]);  
    }  
    for (i = 0; i<=7; i++) {  
        LCD_Chr_Cp(character2[i]);  
    }  
    for (i = 0; i<=7; i++) {  
        LCD_Chr_Cp(character3[i]);  
    }  
    for (i = 0; i<=7; i++) {  
        LCD_Chr_Cp(character4[i]);  
    }  
    for (i = 0; i<=7; i++) {  
        LCD_Chr_Cp(character5[i]);  
    }  
    LCD_Cmd(_LCD_RETURN_HOME);  
}  
//*****  
*  
void      BarDisp(char row, char column, short mode, unsigned int dat)  
{  
    short    i, j, k, cnt;  
    //  
    if (mode == 0) {  
        i = (dat * 10) / 204;  
    } else {  
        i = dat;  
    }  
}
```

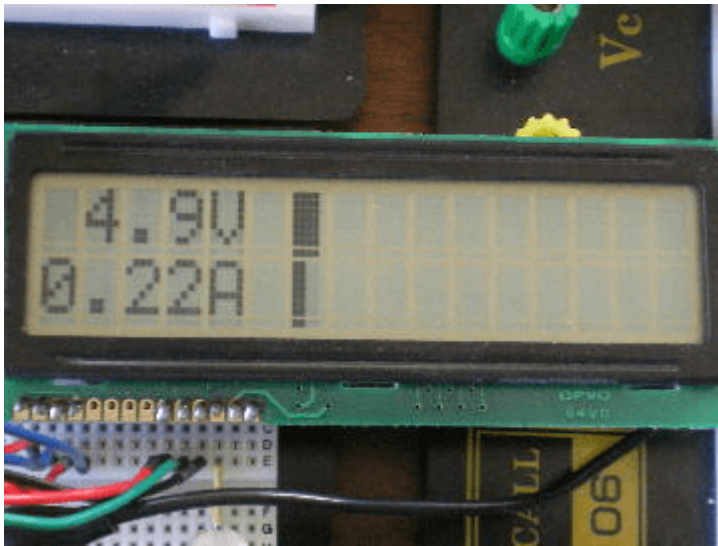
```
    j = i / 5;
    k = i - (j * 5);
    //
    if (row == 1)
        Lcd_Cmd(_LCD_FIRST_ROW);
else
    Lcd_Cmd(_LCD_SECOND_ROW);
    //
    for (cnt = 1; cnt < column; cnt++) {
        Lcd_Cmd(_LCD_MOVE_CURSOR_RIGHT);
    }
    //
    for (cnt = 0; cnt < j; cnt++) {
        Lcd_Chr_Cp(5);
    }
    Lcd_Chr_Cp(k);
    for (cnt++; cnt < 10; cnt++) {
        Lcd_Chr_Cp(' ');
    }
}
//*****
*
sbit LCD_RS at RA0_bit;
sbit LCD_RW at RA7_bit;
sbit LCD_EN at RA6_bit;
sbit LCD_D7 at RB4_bit;
sbit LCD_D6 at RB5_bit;
sbit LCD_D5 at RB6_bit;
sbit LCD_D4 at RB7_bit;
sbit LCD_RS_Direction at TRISA0_bit;
sbit LCD_RW_Direction at TRISA7_bit;
sbit LCD_EN_Direction at TRISA6_bit;
sbit LCD_D7_Direction at TRISB4_bit;
sbit LCD_D6_Direction at TRISB5_bit;
sbit LCD_D5_Direction at TRISB6_bit;
sbit LCD_D4_Direction at TRISB7_bit;
//
void init_lcd()
{
    short cnt;
    //
    LCD_RW_Direction = 0;
    LCD_RW = 0;
    Lcd_Init();
    RegistCustomChar();
    Lcd_Cmd(_LCD_CURSOR_OFF);
    Lcd_Cmd(_LCD_CLEAR);
    Lcd_Out(1, 1, "V&I Meter V2.0");
    Delay_ms(1000);
    Lcd_Cmd(_LCD_CLEAR);
    for (cnt = 0; cnt <= 80; cnt++) {
```

```
        BarDisp(1, 1, 1, cnt);
        Delay_ms(10);
    }
    for (cnt = 0; cnt <= 80; cnt++) {
        BarDisp(2, 1, 1, cnt);
        Delay_ms(10);
    }
    Lcd_Cmd(_LCD_CLEAR);
}
//*****
*
int    measurement(unsigned short channel)
{
    int    cnt;
    long   ad;
    //
    ad = 0;
    for (cnt = 0; cnt < 1000; cnt++) {
        ad += ADC_Get_Sample(channel);
    }
    return (ad /1000);
}
//*****
*
void    main()
{
    char    buf[20];
    double  v1, v2, i;
    long    tmp;
    //
    OSCCON = 0b01110000;
    CMCON  = 0b00000111;
    ANSEL  = 0b00000110;
    TRISA  = 0b11111111;
    TRISB  = 0b00000000;
    //
    ADC_Init();
    init_lcd();
    //
    while (1) {
        v1 = measurement(1);
        v2 = measurement(2);
        //
        BarDisp(1, 7, 0, v1);
        BarDisp(2, 7, 0, v2);
        //
        v1 = (v1 * 11) * 4.8828125;
        v2 = (v2 / 11) * 4.8828125;
        i = v2 / 0.1;
        //
    }
}
```

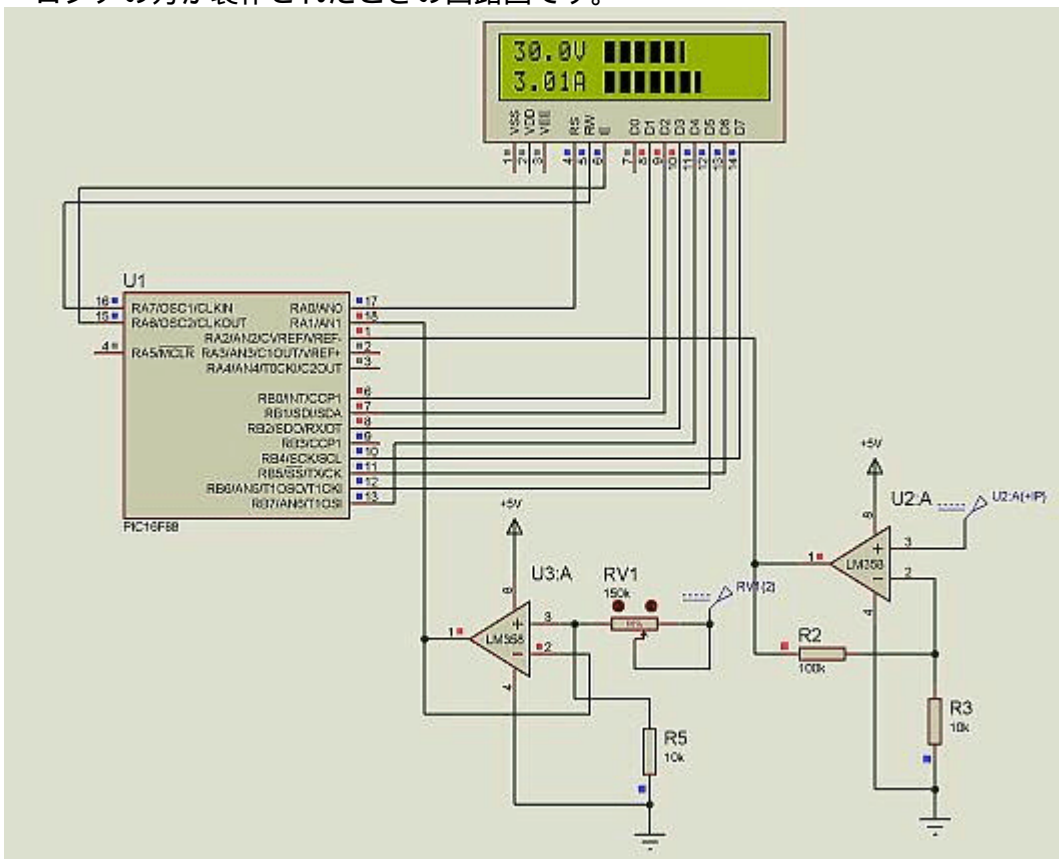
```
tmp = v1 - v2;
if ((tmp % 100) > 50) {
    tmp = (tmp / 100) + 1;
} else {
    tmp = tmp / 100;
}
WordToStr(tmp, buf);
buf[0] = buf[2];
buf[1] = (buf[3] == ' ') ? '0' : buf[3];
buf[2] = '.';
buf[3] = buf[4];
buf[4] = 'V';
buf[5] = 0x00;
Lcd_Out(1, 1, buf);
//
tmp = i;
if ((tmp % 10) > 5) {
    tmp = (tmp / 10) + 1;
} else {
    tmp = tmp / 10;
}
WordToStr(tmp, buf);
buf[0] = (buf[2] == ' ') ? '0' : buf[2];
buf[1] = '.';
buf[2] = (buf[3] == ' ') ? '0' : buf[3];
buf[3] = buf[4];
buf[4] = 'A';
buf[5] = 0x00;
Lcd_Out(2, 1, buf);
}
}
//*****
*
```

動作確認





ロシアの方が製作されたときの回路図です。



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:pic16f88:149&rev=1588228324>

Last update: 2025/10/17 14:28

