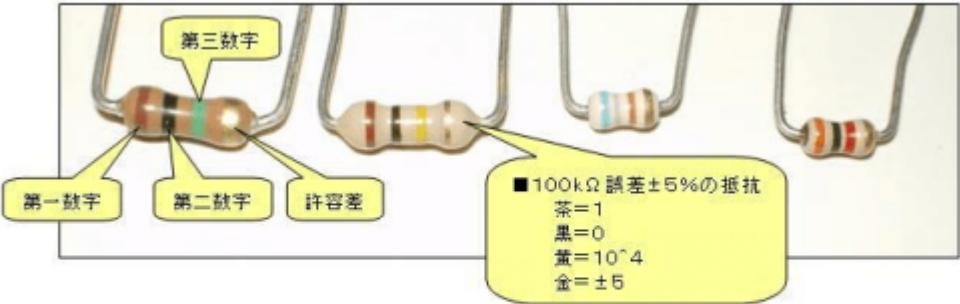


カラーコード値表示ユニット

概要

製作物では、4帯カラーコード(1/4W)の抵抗をよく使います。しかし、よく使う抵抗値(黒~緑)は、長年の経験で直ぐに分かるのですが、あまり使わない抵抗(青~白)は、間違えたり、判断に時間がかかります。

そこで今回は、抵抗のカラーコードをスイッチで設定するだけで、抵抗値を自動計算し結果を表示させるユニットを製作してみました。



色	第一数字	第二数字	第三数字 (乗数)	許容差 (%)
黒	0	0	10 ⁰	—
茶	1	1	10 ¹	±1
赤	2	2	10 ²	±2
橙	3	3	10 ³	±0.05
黄	4	4	10 ⁴	—
緑	5	5	10 ⁵	±0.5
青	6	6	10 ⁶	±0.25
紫	7	7	10 ⁷	±0.1
灰	8	8	10 ⁸	—
白	9	9	10 ⁹	—
金	—	—	10 ⁻¹	±5
銀	—	—	10 ⁻²	±10
無色	—	—	—	±20

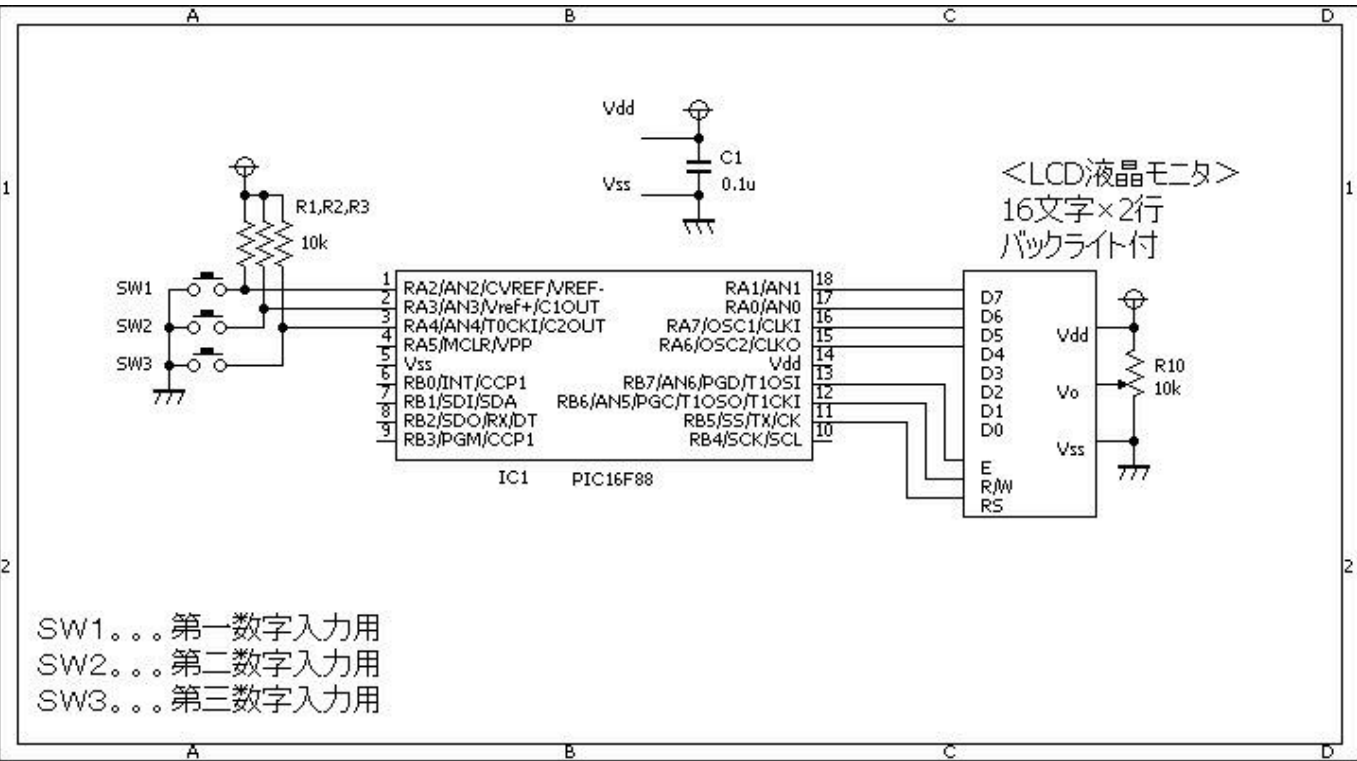
動作原理

3個のプッシュスイッチで、第一数字、第二数字、第三数字を設定します。スイッチを押下する毎に、LCDに表示される色データが、“黒(加)” “~” “白(刈)” と変化していきます。

<表示内容> 加 弌 弌 弌 弌 弌 弌 弌 弌 弌

色が変化する毎に、抵抗値を計算し、LCDへ表示します。計算式は、次のようになります。抵抗=((第一数字×10)+第二数字)×(10^第三数字) LCDへの表示は、全てを、“Ω”にすると桁数が大きくなり直感的ではなくなるので、次の4単位表示としました。“Ω” “KΩ” “MΩ” “GΩ”

回路図



ソースコード

color_code.c

```
//*****
*
/*
<カラーコード表示>
*/
//*****
*

#define SW1 PORTA.F2
#define SW2 PORTA.F3
#define SW3 PORTA.F4

//*****
*

char *color_code[10] = {"0", "1", "2", "3", "4", "5", "6", "7", "8", "9"};
char buf[32];

void colorCodeDisp(short i, short j, short k)
{
    static short val;
```

```
//
buf[0] = 0x00;
strcat(buf, color_code[i]);
strcat(buf, ",");
strcat(buf, color_code[j]);
strcat(buf, ",");
strcat(buf, color_code[k]);
//
if (strlen(buf) <= 16) {
    Lcd_Custom_Out(1, 1, buf);
} else {
    Lcd_Custom_Out(2, 1, &buf[16]);
    buf[16] = 0x00;
    Lcd_Custom_Out(1, 1, buf);
}
//
val = (i * 10) + j;
ByteToStr(val, buf);
//
switch (k) {
case 0:    //黒色
    Lcd_Custom_Out(2, 9, buf);
    Lcd_Custom_Chr(2, 12, 0xF4);    //Ω
    break;
case 1:    //茶色
    buf[4] = 0x00;
    buf[3] = '0';
    Lcd_Custom_Out(2, 8, buf);
    Lcd_Custom_Chr(2, 12, 0xF4);    //Ω
    break;
case 2:    //赤色
    buf[4] = 0x00;
    buf[3] = buf[2];
    buf[2] = '.';
    Lcd_Custom_Out(2, 8, buf);
    Lcd_Custom_Chr(2, 12, 'K');
    Lcd_Custom_Chr(2, 13, 0xF4);    //Ω
    break;
case 3:    //橙色
    Lcd_Custom_Out(2, 9, buf);
    Lcd_Custom_Chr(2, 12, 'K');
    Lcd_Custom_Chr(2, 13, 0xF4);    //Ω
    break;
case 4:    //黄色
    buf[4] = 0x00;
    buf[3] = '0';
    Lcd_Custom_Out(2, 8, buf);
    Lcd_Custom_Chr(2, 12, 'K');
    Lcd_Custom_Chr(2, 13, 0xF4);    //Ω
    break;
case 5:    //緑色
```

```

        buf[4] = 0x00;
        buf[3] = buf[2];
        buf[2] = '.';
        Lcd_Custom_Out(2, 8, buf);
        Lcd_Custom_Chr(2, 12, 'M');
        Lcd_Custom_Chr(2, 13, 0xF4);    //Ω
        break;
    case 6:    //青色
        Lcd_Custom_Out(2, 9, buf);
        Lcd_Custom_Chr(2, 12, 'M');
        Lcd_Custom_Chr(2, 13, 0xF4);    //Ω
        break;
    case 7:    //紫色
        buf[4] = 0x00;
        buf[3] = '0';
        Lcd_Custom_Out(2, 8, buf);
        Lcd_Custom_Chr(2, 12, 'M');
        Lcd_Custom_Chr(2, 13, 0xF4);    //Ω
        break;
    case 8:    //灰色
        buf[4] = 0x00;
        buf[3] = buf[2];
        buf[2] = '.';
        Lcd_Custom_Out(2, 8, buf);
        Lcd_Custom_Chr(2, 12, 'G');
        Lcd_Custom_Chr(2, 13, 0xF4);    //Ω
        break;
    case 9:    //白色
        Lcd_Custom_Out(2, 9, buf);
        Lcd_Custom_Chr(2, 12, 'G');
        Lcd_Custom_Chr(2, 13, 0xF4);    //Ω
        break;
    }
}

//*****
*

void main()
{
    static    short    cnt, i, j, k;
    //
    OSCCON = 0b01110000;    // クロックを8Mhzに設定する。
    ANSEL   = 0b00000000;    // □□□変換は使用しない。
    //ポートの設定
    TRISA = 0b11111111;
    TRISB = 0b11111111;
    //
    Lcd_Custom_Config(&PORTA, 1, 0, 7, 6, &PORTB, 5, 6, 7);
    Lcd_Custom_Cmd(LCD_CURSOR_OFF);
    Lcd_Custom_Cmd(LCD_CLEAR);

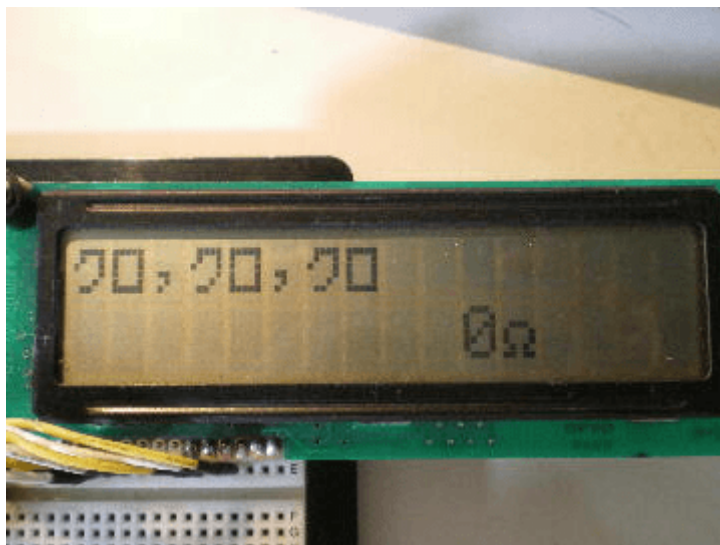
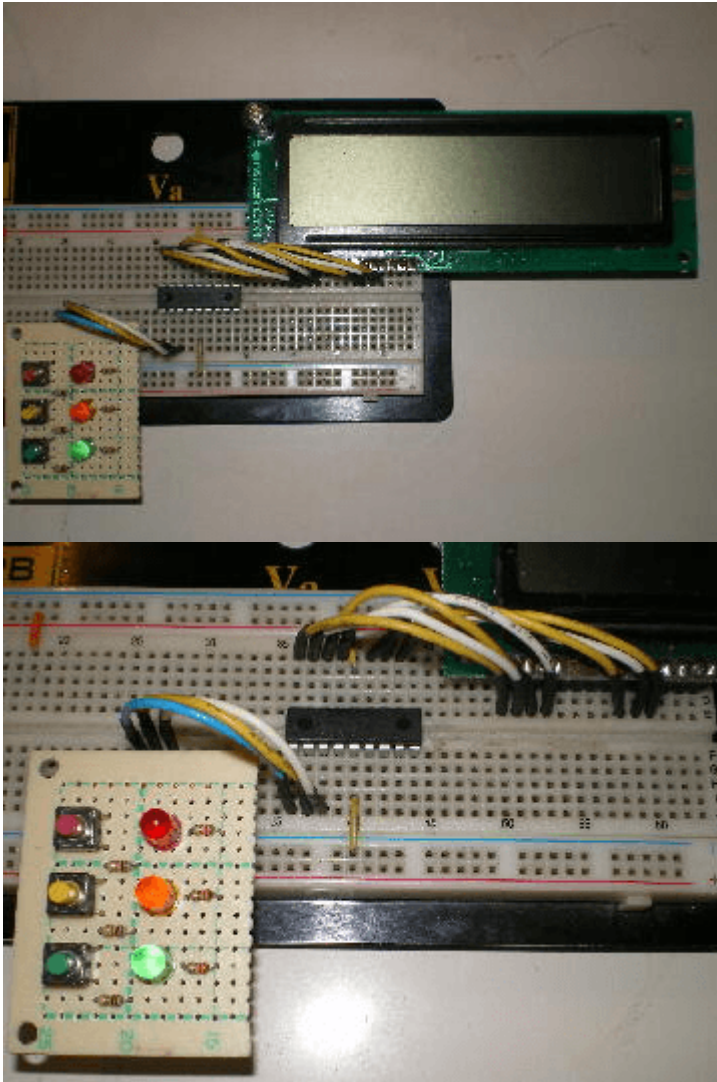
```

```
//
i = 0;
j = 0;
k = 0;
//
colorCodeDisp(i, j, k);
//
while (1) {
    if (SW1 == 0) {
        while (Button(&PORTA, 2, 1, 1) == 0)
            ;
        //
        if (i < 9)
            i++;
        else
            i = 0;
        //
        Lcd_Custom_Cmd(LCD_CLEAR);
        colorCodeDisp(i, j, k);
    }
    if (SW2 == 0) {
        while (Button(&PORTA, 3, 1, 1) == 0)
            ;
        //
        if (j < 9)
            j++;
        else
            j = 0;
        //
        Lcd_Custom_Cmd(LCD_CLEAR);
        colorCodeDisp(i, j, k);
    }
    if (SW3 == 0) {
        while (Button(&PORTA, 4, 1, 1) == 0)
            ;
        //
        if (k < 9)
            k++;
        else
            k = 0;
        //
        Lcd_Custom_Cmd(LCD_CLEAR);
        colorCodeDisp(i, j, k);
    }
}
}

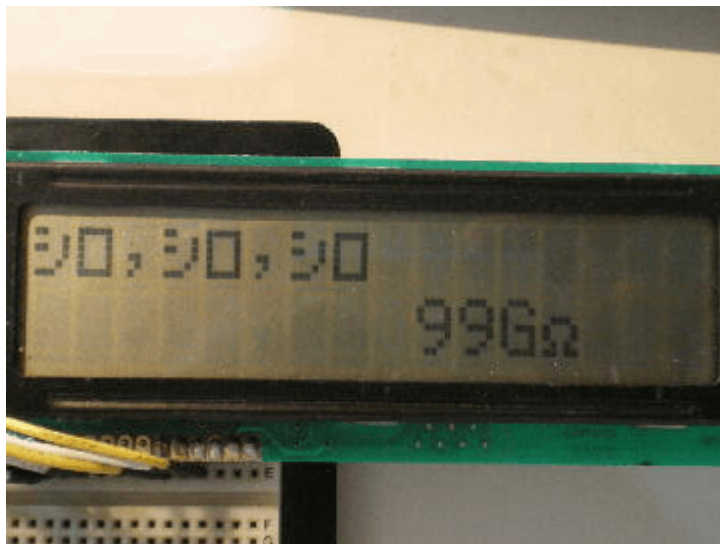
//*****
*
```

動作確認

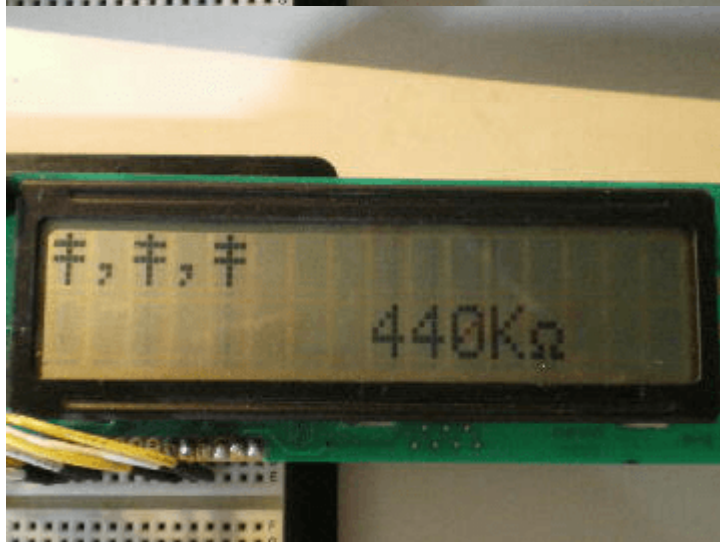
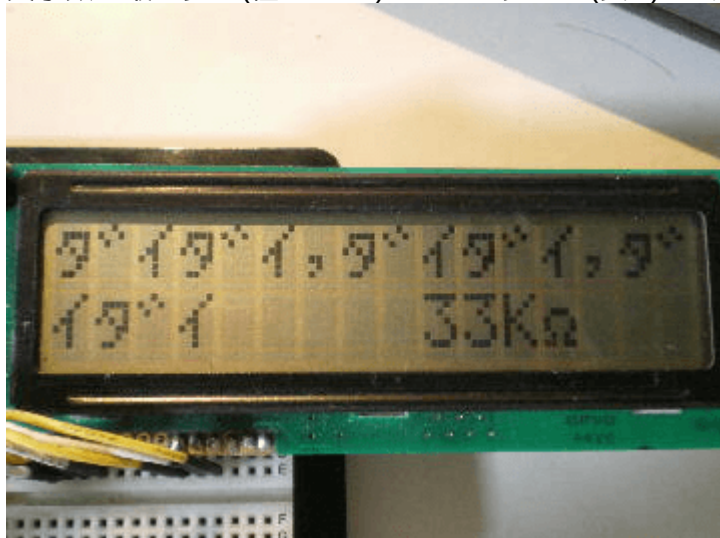
第一数字 赤色プッシュSW 第二数字 黄色プッシュSW 第三数字 緑色プッシュSW LEDは未使用です。



最小値(0Ω)と最大値(99GΩ)です。



文字数の最も多い(橙:ダ イ イ) と少ない(黄:キ) です。





乗数を、“黒(10^0)”～“白(10^9)”まで変化させて見ました。





如何ですか? 手元に1台あると何かと便利ですね 😊

著作権表示 copyright notice

このページは稲崎様の閉鎖したHPのコピーで、著作権は稲崎様にあります。詳細 This page is a copy of Mr. Inasaki's closed website, and the copyright is held by him. [Details](#)

From:

<http://deepsky.jp/wiki/> - うごくといいな

Permanent link:

<http://deepsky.jp/wiki/doku.php?id=elechobby:picdic:pic16f88:117>

Last update: 2020/05/01 17:54

