

カラー受光&カラー発光

概要

カラー受光してRGBの値を得ること、またRGBの値を指定してカラー発光させることについては、

- カラー受光 カラーセンサ(S9706)
- カラー発光 フルカラーLED(色設定ユニット)

等で紹介しました。

今回は、これらの受光と発光を組み合わせてみました。つまり、カラーをRGBの値として受光して、その値から、カラーをRGBの値として発光させて見ました。

動作原理

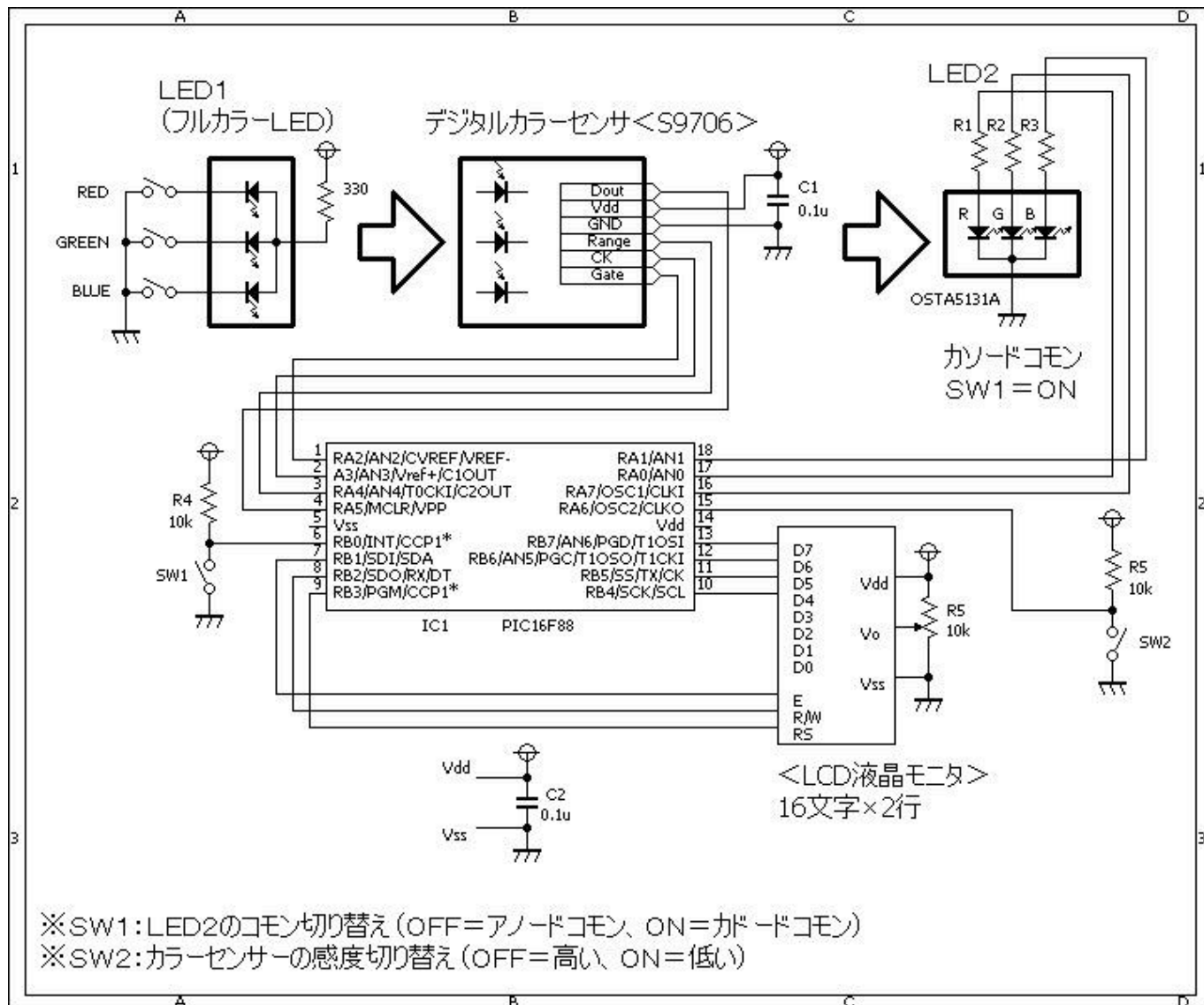
カラー受光の原理については、カラーセンサ(S9706)を参照してください。カラー発光の原理については、フルカラーLED(色設定ユニット)を参照してください。

これらを組み合わせるためには、レンジ合わせの工夫が少し要ります。 <レンジ合わせ>

- カラー受光において、カラーセンサ(S9706)で得られる値(V1)は、0~4095です。
- カラー発光において、フルカラーLED(色設定ユニット)で指定できる値(V2)は、0~100です。
- そこでV1を41で割ることによりV2のレンジに合わせます。
 $100 \approx 4095 \div 41$

回路図

LED1は、手持ちの適当なフルカラー発光LEDを使用しました。



ソースコード

[Color2Color.c](#)

```
//*****
*
*/
    『カラー カラー』
*/
//*****
*

#define LED_R    PORTA.F0
#define LED_G    PORTA.F7
#define LED_B    PORTA.F1

#define SW1      PORTB.F0

#define ON       0
```

```
#define      OFF      1

#define      GATE      PORTA.F2
#define      CK        PORTA.F3
#define      RANGE     PORTA.F4
#define      DOUT      PORTA.F5

#define      RANGE_SW  PORTA.F6

//*****
*

void  GetColor(short sensitivity, int addTime, int pColor[])
{
    static  unsigned  int      RED, GREEN, BLUE, tmp;
    static  unsigned  short   cnt;
    //
    GATE = 0;
    CK = 0;
    RANGE = sensitivity;
    GATE = 1;
    for (; addTime > 0; addTime--) {
        Delay_ms(1);
    }
    GATE = 0;
    //赤色データの取り込み
    RED = 0;
    for (cnt = 0; cnt < 12; cnt++) {
        CK = 1;
        CK = 0;
        tmp = DOUT << cnt;
        RED |= tmp;
    }
    pColor[0] = RED;
    //緑色データの取り込み
    GREEN = 0;
    for (cnt = 0; cnt < 12; cnt++) {
        CK = 1;
        CK = 0;
        tmp = DOUT << cnt;
        GREEN |= tmp;
    }
    pColor[1] = GREEN;
    //青色データの取り込み
    BLUE = 0;
    for (cnt = 0; cnt < 12; cnt++) {
        CK = 1;
        CK = 0;
        tmp = DOUT << cnt;
        BLUE |= tmp;
    }
}
```

```
pColor[2] = BLUE;
}

//*****
*

static unsigned short pwm_cnt, led_R, led_G, led_B;

void interrupt() // 約0.128msec周期
{
    static short onFlg, offFlg;
    //
    INTCON.T0IF = 0;
    //
    onFlg = (SW1 == 1) ? 0 : 1;
    offFlg = (SW1 == 1) ? 1 : 0;
    //
    LED_R = (pwm_cnt <= led_R) ? onFlg : offFlg;
    LED_G = (pwm_cnt <= led_G) ? onFlg : offFlg;
    LED_B = (pwm_cnt <= led_B) ? onFlg : offFlg;
    //約12.8msec□78Hz□
    if (pwm_cnt < 100)
        pwm_cnt++;
    else
        pwm_cnt = 1;
}

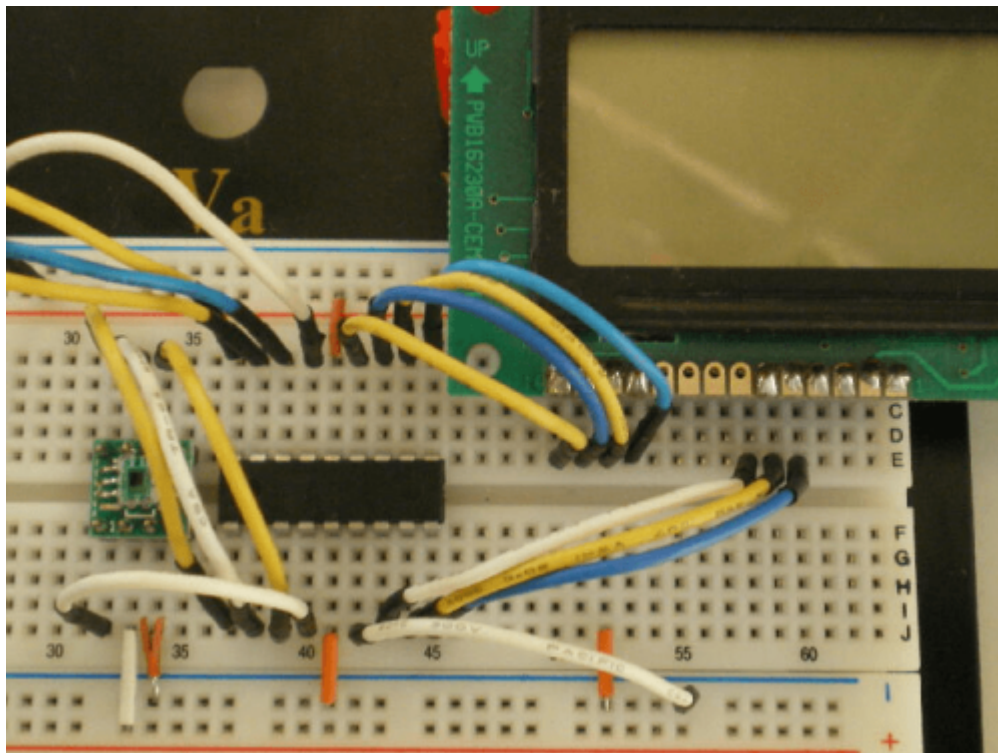
//*****
*

void main()
{
    static char buf[10];
    static unsigned int color[3];
    //
    OSCCON = 0b01110000; // クロックは8Mhz
    CMCON = 0b00000111; // コンパレータは使用しない。
    ANSEL = 0b00000000; // □□□変換を使用する。
    TRISA = 0b01100000;
    TRISB = 0b00000001;
    //□□□の初期化
    Lcd_Config(&PORTB, 3, 1, 2, 7, 6, 5, 4);
    Lcd_Cmd(LCD_CURSOR_OFF);
    Lcd_Cmd(LCD_CLEAR);
    Lcd_Chr(1, 2, 'R');
    Lcd_Chr(1, 6, 'G');
    Lcd_Chr(1, 10, 'B');
    // TIMER0の設定
    INTCON.T0IE = 1;
    INTCON.T0IF = 0;
}
```

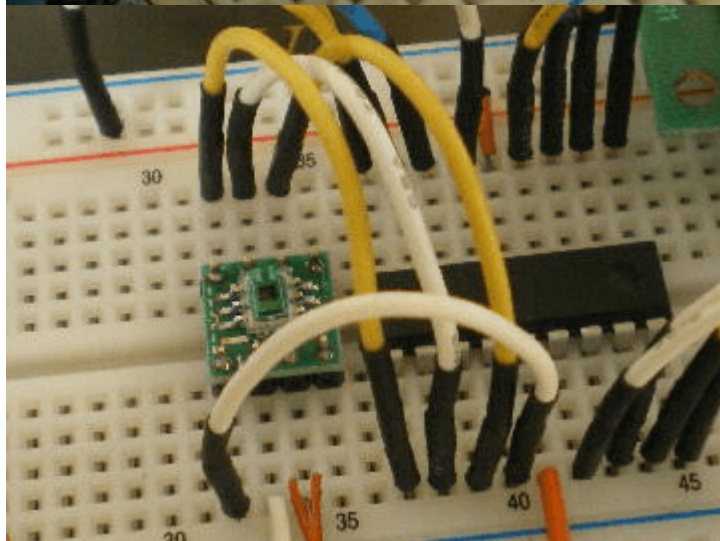
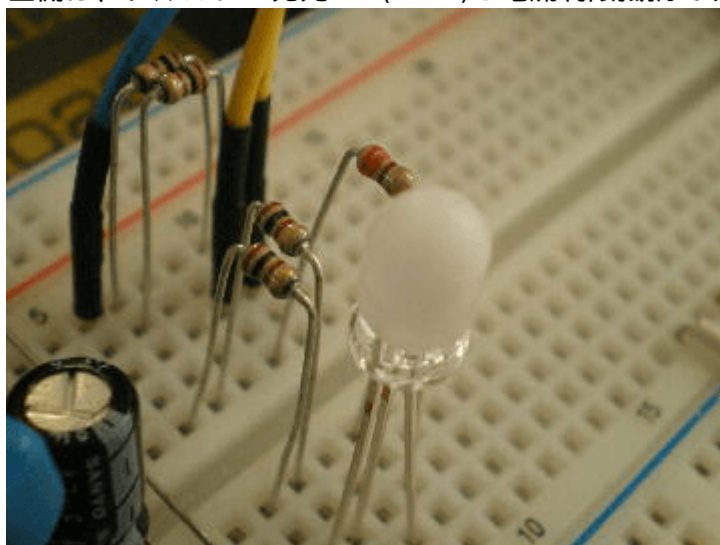
```
OPTION_REG.T0CS = 0;
OPTION_REG.PSA = 1;
OPTION_REG.PS0 = 0;
OPTION_REG.PS1 = 0;
OPTION_REG.PS2 = 0;
TMR0 = 0;
//
pwm_cnt = 1;
led_R = 50;
led_G = 50;
led_B = 50;
// 割り込みを許可する。
INTCON.PEIE = 1;
INTCON.GIE = 1;
//
while (1) {
    //カラーデータの取得
    GetColor(RANGE_SW, 100, color);
    //
    led_R = color[0] / 41;
    led_G = color[1] / 41;
    led_B = color[2] / 41;
    //
    ByteToStr(led_R, buf);
    Lcd_Out(2, 1, buf);
    ByteToStr(led_G, buf);
    Lcd_Out(2, 5, buf);
    ByteToStr(led_B, buf);
    Lcd_Out(2, 9, buf);
}
} //~!

//*****
*
```

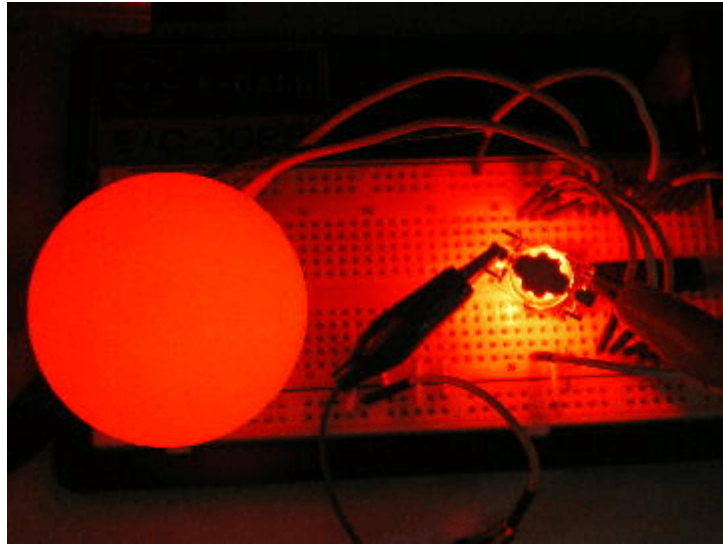
動作確認



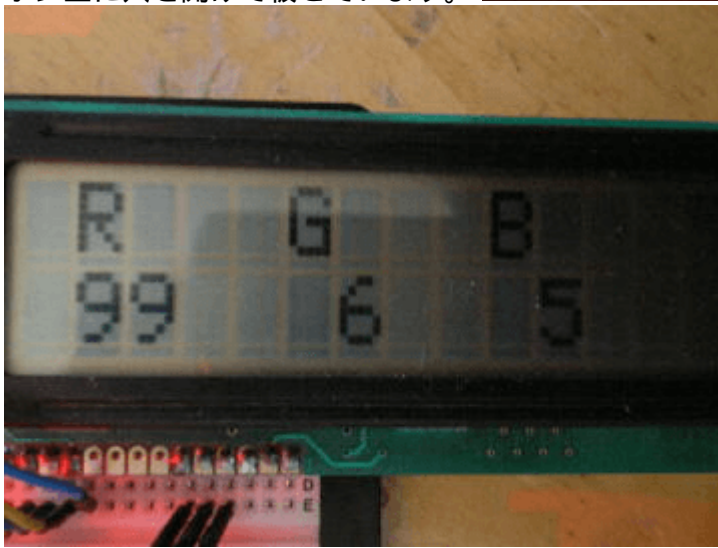
左側は、フルカラー発光LED(LED2)と電流制限抵抗です。右側は、カラーセンサPICLCDです。



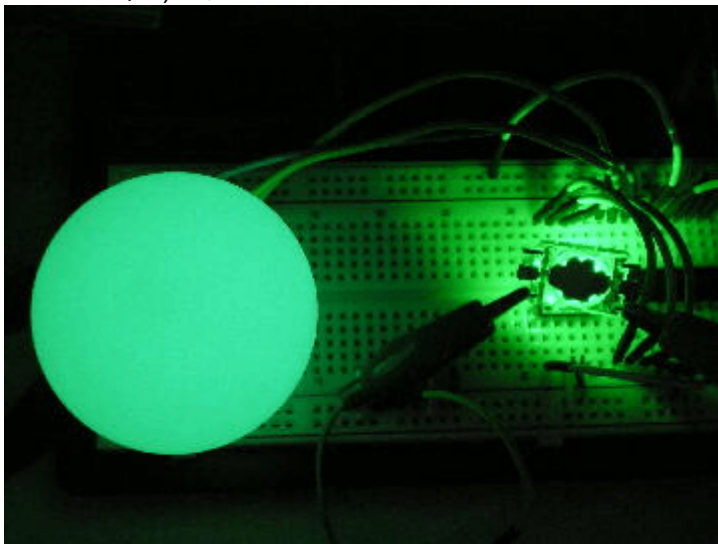
LED1のR(赤)を、カラーセンサーの上で点灯させてみました。LED2には、光を拡散させるために、ピン

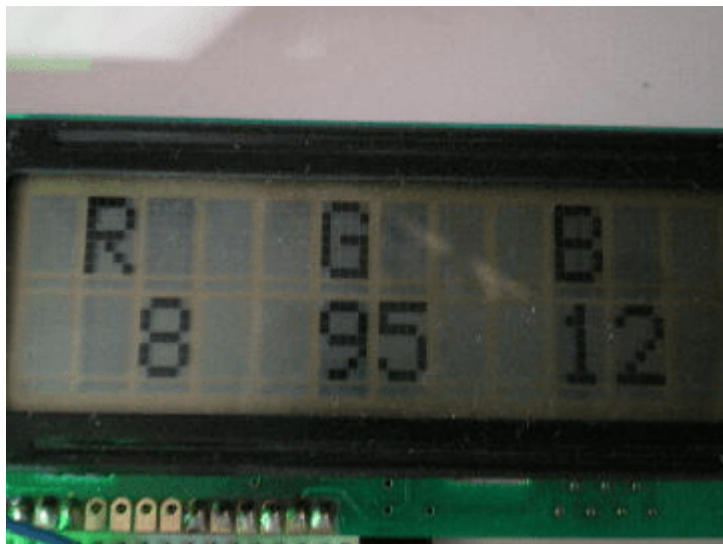


ポン玉に穴を開けて被せています。

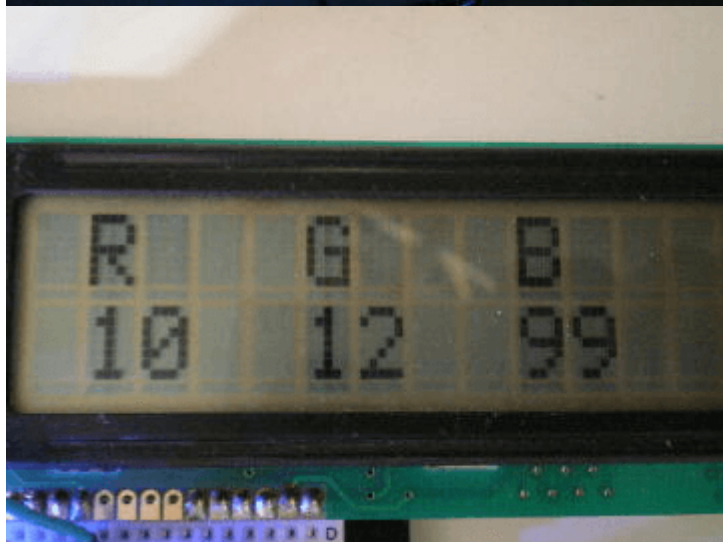
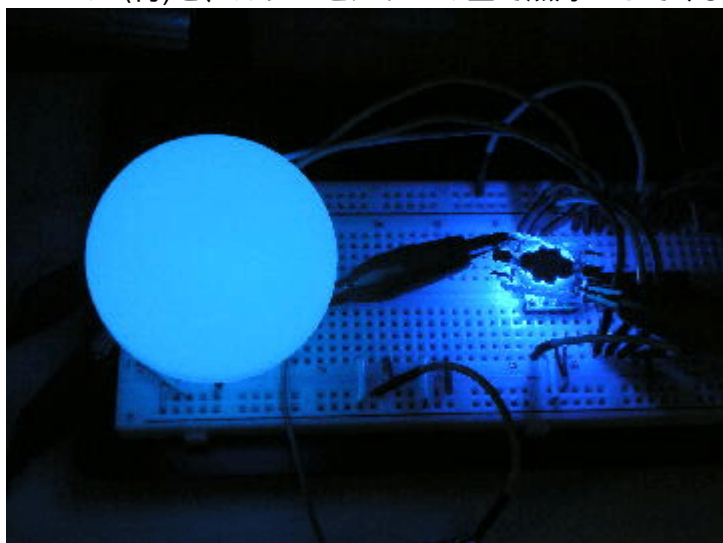


LED1のG(緑)を、カラーセンサーの上で点灯させてみました。





LED1のB(青)を、カラーセンサーの上で点灯させてみました。



如何でしょう? 今回製作した物は、どんな応用が考えられるでしょうか? 例えば、光リピータ。。。光を直角に曲げる!?

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:pic16f88:125&rev=1588214276>

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

