

RGB口ガ(PIC18F2550)

概要

カラーセンサ(S9706)の基本的な動作(RGBのデータを取得し、その値を単純にLCDに表示させる)については、実験室のカラーセンサ(S9706)で確認しました。今回は、LCDに表示させるのではなく、取得したデータを、SDカードに逐次記録するようにしました。

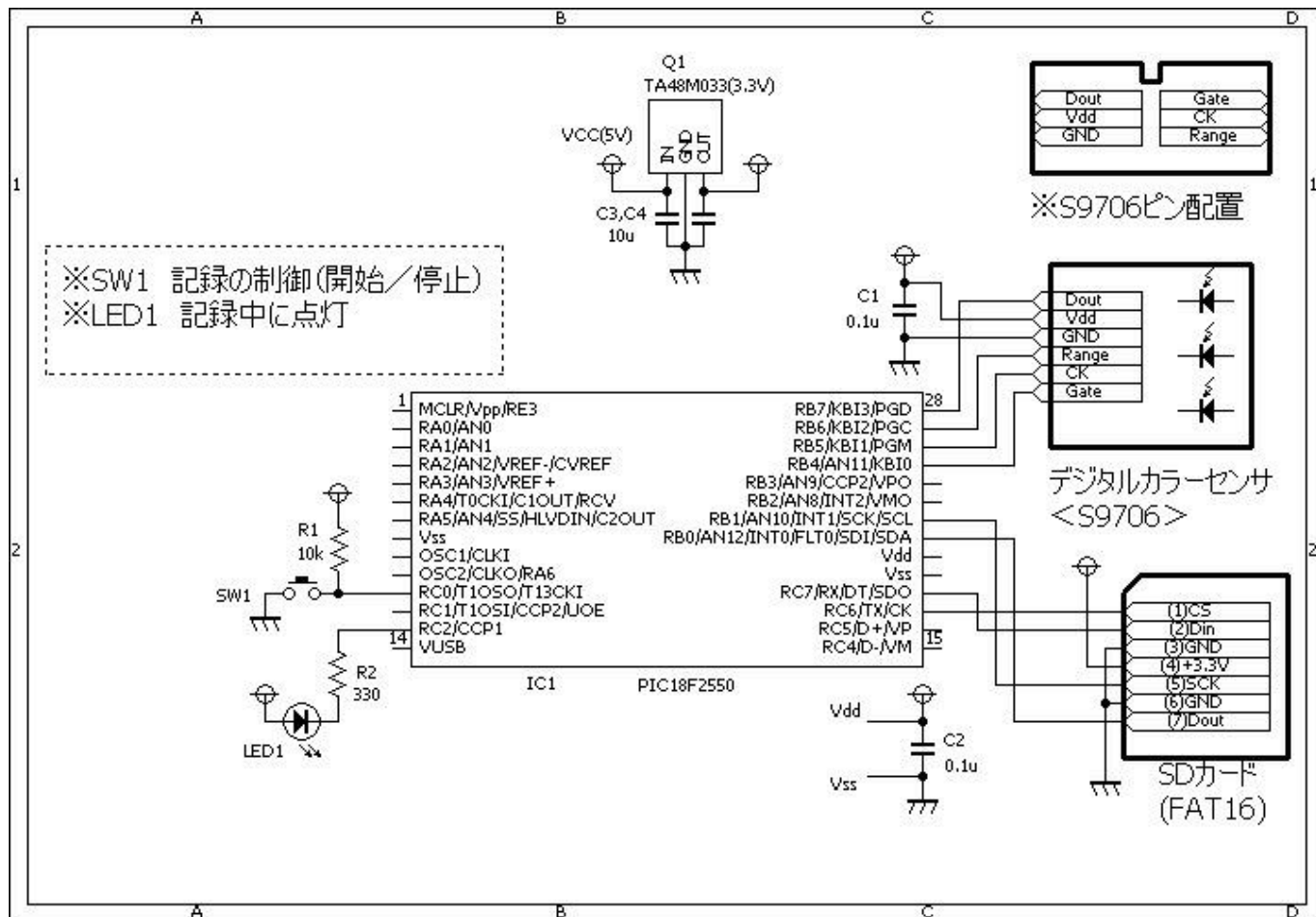
動作原理

カラーセンサ(S9706)の動作原理については、実験室のカラーセンサ(S9706)を参照してください。尚、S8706は感度設定が2段階(高感度、低感度)あるので、両感度のRGB値を取得し記録します。記録速度は、1秒間に約3件分を記録します。

```
<SDカードへの記録フォーマット(ファイル名は、“log.txt"固定)>
一行に、“H,R1,G1,B1,L,R2,G2,B2"の順番に記録します。
(H=高感度(R1,G1,B1) L=低感度(R2,G2,B2))
```

```
$START
H,1234,1234,1234,L,1234,1234,1234
H,1234,1234,1234,L,1234,1234,1234
H,1234,1234,1234,L,1234,1234,1234
:
:
:
$STOP
```

回路図



ソースコード

[rgb_logger.c](#)

```

//*****
*
/*
『ロガー』
デジタルカラーセンサを使用し、のデータをに記録する。
記録データは、高感度低感度の2種類とする。
<記録例>
*/
//*****
*

#define SW PORTC.F0
#define LED PORTC.F2

#define CR 0x0d
#define LF 0x0a

#define ON 0
#define OFF 1

```

```
#define GATE PORTB.F4
#define CK PORTB.F5
#define RANGE PORTB.F6
#define DOUT PORTB.F7

//*****
*

void GetColor(short sensitivity, int addTime, unsigned 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 = RED;
    pColor++;
    //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 = GREEN;
    pColor++;
    //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 = BLUE;
}
```

```
}

//*****
*

void  init_adc()
{
    //変換の設定
    ADCON1.PCFG3 = 1;
    ADCON1.PCFG2 = 0;
    ADCON1.PCFG1 = 1;
    ADCON1.PCFG0 = 0;
}

//*****
*

void  init_sdc()
{
    static  short  cnt;
    //初期化
    Spi_Init_Advanced(MASTER_OSC_DIV64, DATA_SAMPLE_MIDDLE,
CLK_IDLE_LOW, LOW_2_HIGH);
    if (Mmc_Fat_Init(&PORTC, 6)) {
        while (1) {
            LED = ON;
            Delay_ms(100);
            LED = OFF;
            Delay_ms(100);
        }
    }
    Spi_Init_Advanced(MASTER_OSC_DIV16, DATA_SAMPLE_MIDDLE,
CLK_IDLE_LOW, LOW_2_HIGH);
    for (cnt = 0; cnt < 3; cnt++) {
        LED = ON;
        Delay_ms(300);
        LED = OFF;
        Delay_ms(300);
    }
}

//*****
*

void  main()
{
    //変数の定義
    static  char    buf[16];
    static  short   mode;
    static  double  ad;
    static  unsigned int    color[3];
}
```

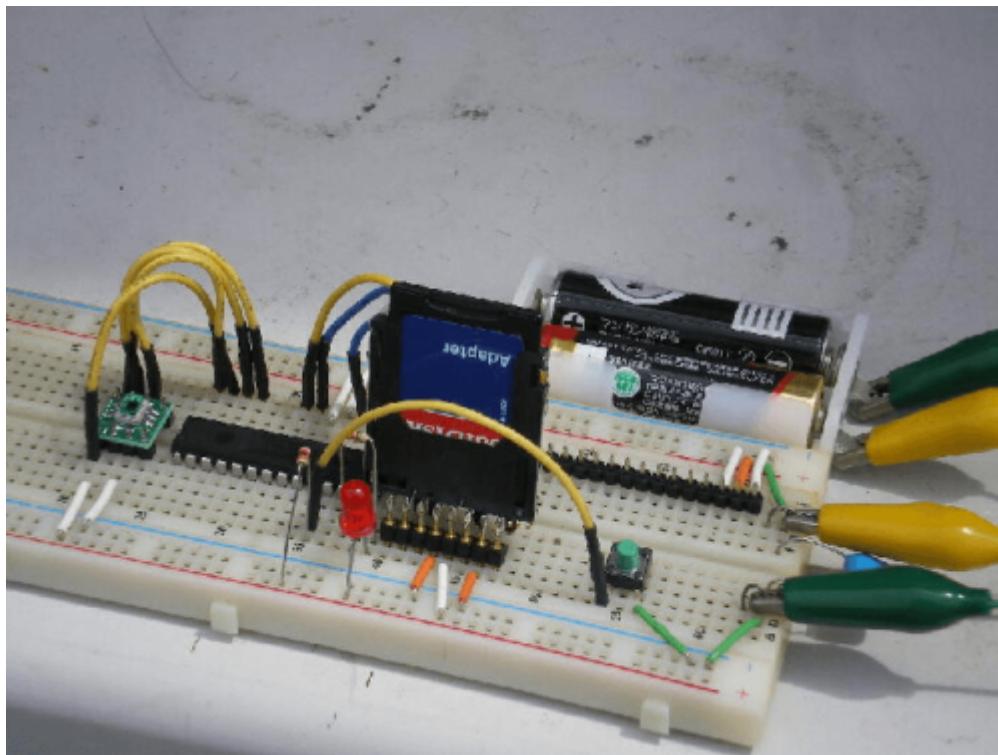
```
//
OSCCON.IRCF2 = 1;
OSCCON.IRCF1 = 1;
OSCCON.IRCF0 = 1;
//ポートの設定
TRISA = 0b01111111;
TRISB = 0b10001111;
TRISC = 0b00000001;
//コンパレータは使用しない。
CMCON = 0b00000111;
//AD変換の設定
init_adc();
//変数の初期化
LED = OFF;
mode = 0;
//SDカードの初期化
init_sdc();
//
while (1) {
    if ((mode == 0) && (SW == ON)) {
        while (SW == ON) {
            Delay_ms(100);
        }
        mode = 1;
        LED = ON;
        //
        Mmc_Fat_Assign("log.txt", 0xA0);
        Mmc_Fat_Rewrite();
        Mmc_Fat_Write("$START\r\n", 8);
    }
    //
    if ((mode == 1) && (SW == ON)) {
        while (SW == ON) {
            Delay_ms(100);
        }
        mode = 0;
        LED = OFF;
        //
        Mmc_Fat_Write("$STOP\r\n", 7);
    }
    //
    if (mode == 1) {
        //カラーデータの取得(高感度)
        GetColor(1, 100, color);
        //赤色データ表示
        WordToStr(color[0], buf);
        Mmc_Fat_Write("H,", 2);
        Mmc_Fat_Write(&buf[1], 4);
        //緑データ表示
        WordToStr(color[1], buf);
        Mmc_Fat_Write(", ", 1);
    }
}
```

```
Mmc_Fat_Write(&buf[1], 4);
//青色データ表示
WordToStr(color[2], buf);
Mmc_Fat_Write(",", 1);
Mmc_Fat_Write(&buf[1], 4);
//カラーデータの取得(低感度)
GetColor(0, 100, color);
//赤色データ表示
WordToStr(color[0], buf);
Mmc_Fat_Write("L", 3);
Mmc_Fat_Write(&buf[1], 4);
//緑データ表示
WordToStr(color[1], buf);
Mmc_Fat_Write(",", 1);
Mmc_Fat_Write(&buf[1], 4);
//青色データ表示
WordToStr(color[2], buf);
Mmc_Fat_Write(",", 1);
Mmc_Fat_Write(&buf[1], 4);
//
buf[0] = CR;
buf[1] = LF;
Mmc_Fat_Write(buf, 2);
}
}
}

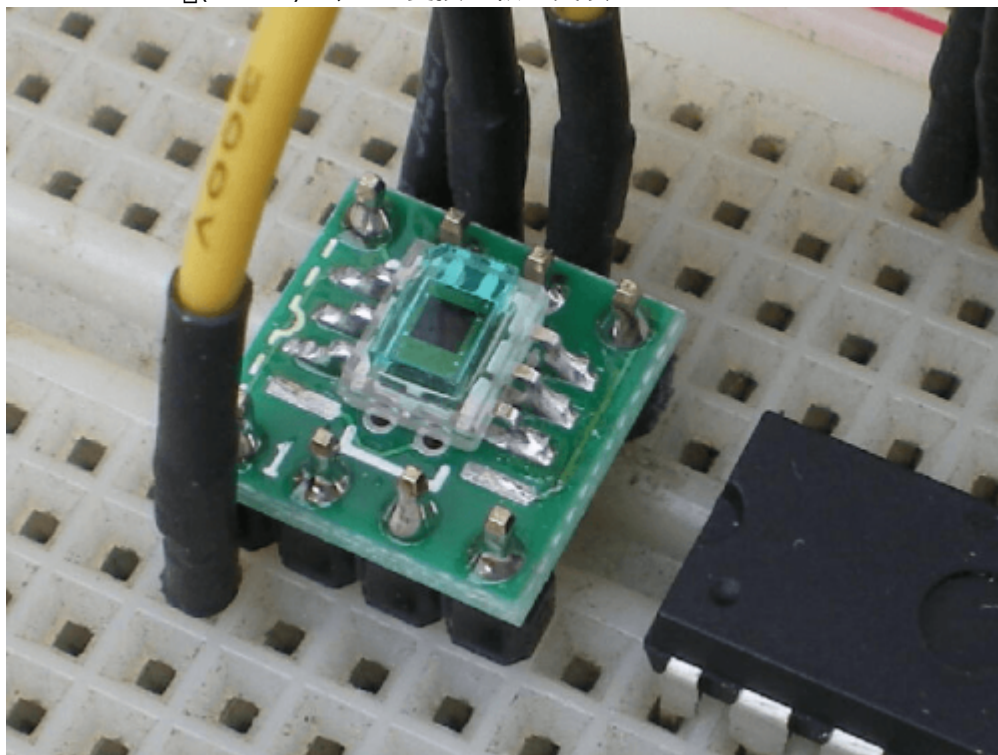
//*****
*
```

動作確認

単三電池では、少し、電圧が低いのですが、特に問題なく動作しました。



カラーセンサ(S9706)を、DIP変換基板に実装してからブレッドボードに差し込みます。



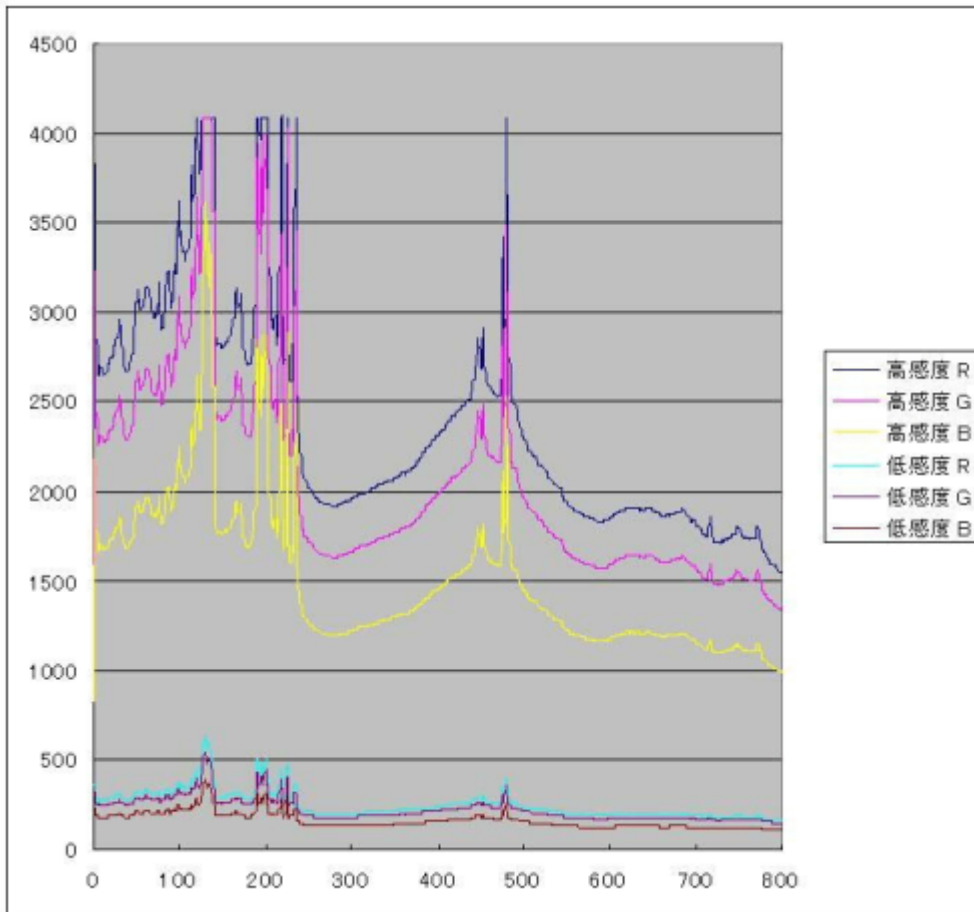
カラーセンサ(S9706)の感度が良過ぎるので、フィルタ(手作り)を取り付けました。



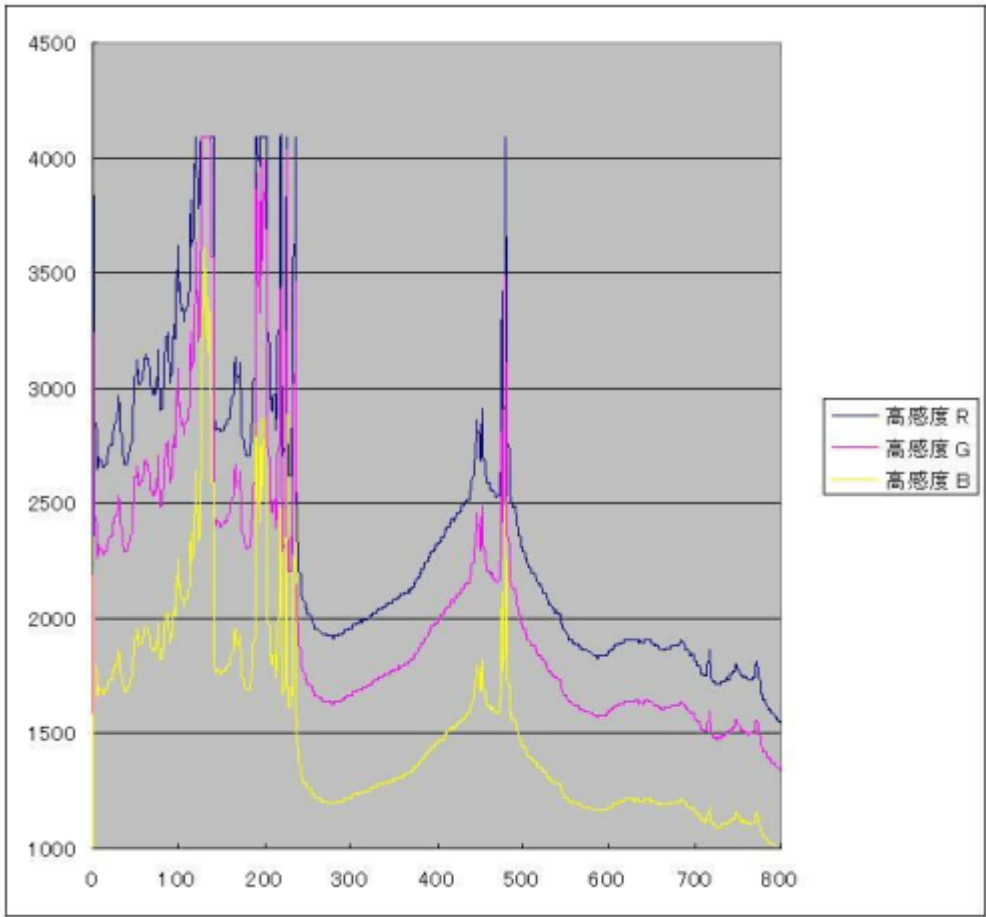
庭に出て測定してみました。測定した時は、曇り空で風が吹いていました。(雲の移動が早い状態) テキストファイル("log.txt")に記録されたデータです。

```
RGB0ガー測定データ - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)
$START
H,1443,1158, 832,L, 356, 312, 221
H,3784,3191,2315,L, 363, 322, 231
H,3044,2599,1901,L, 293, 264, 190
H,2763,2368,1732,L, 285, 256, 185
H,2645,2259,1659,L, 274, 247, 179
H,2699,2318,1703,L, 266, 239, 174
H,2653,2272,1669,L, 279, 251, 182
H,2672,2289,1681,L, 277, 249, 181
H,2689,2302,1692,L, 278, 249, 181
H,2681,2296,1688,L, 276, 248, 180
H,2663,2281,1678,L, 274, 247, 179
H,2653,2274,1673,L, 274, 247, 179
H,2655,2275,1674,L, 275, 247, 179
H,2663,2282,1680,L, 275, 248, 180
H,2675,2293,1687,L, 277, 249, 181
H,2687,2303,1695,L, 278, 250, 182
H,2699,2312,1701,L, 279, 251, 182
H,2713,2324,1709,L, 281, 253, 183
H,2733,2340,1721,L, 283, 254, 184
H,2748,2353,1730,L, 285, 256, 185
H,2759,2362,1736,L, 285, 256, 186
```

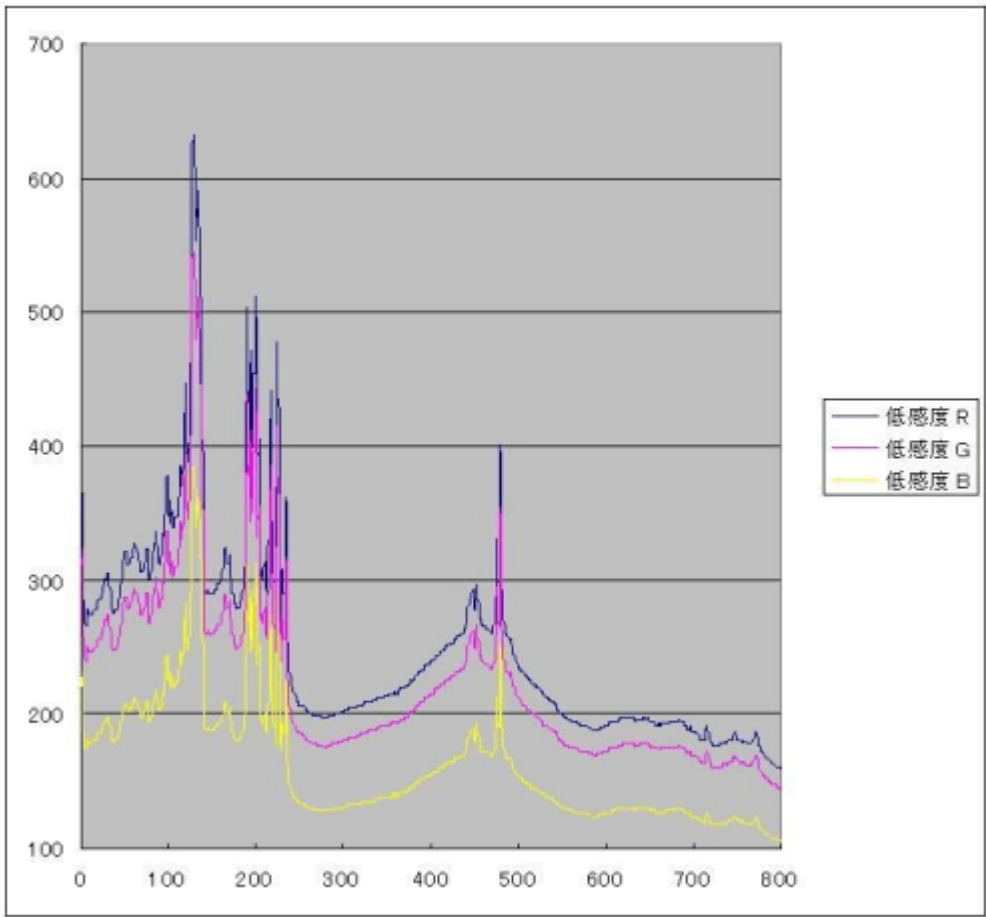
高感度データ(RGB)と低感度データ(RGB)をExcelでグラフ表示してみました。



高感度データ(RGB)のみをExcelでグラフ表示してみました。上側が飽和(サチュレーション(saturation))しています。(最大値は4095で、それ以上では飽和状態です)



低感度データ(RGB)のみをExcelでグラフ表示してみました。



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://www.deepsky.jp/wiki/> - うごくといいな

Permanent link:

<http://www.deepsky.jp/wiki/doku.php?id=elechobby:picdic:otherpic:180>

Last update: **2025/10/17 14:29**

