

簡易照度計V2(S9705)

概要

以前にも、照度/電流変換用のIC(NJL7502L)を使用した、簡易照度計(自動レンジ切り替え)を製作しました。今回は、照度/電流/周波数変換用のIC(S9705)を使用した、簡易照度計を製作しました。

S9705は、フォトダイオードと電流/周波数変換器とを組み合わせたCMOSフォトICで、入射光強度に比例した発振周波数の矩形波を出力します。広いダイナミックレンジを持ち、カウンタ等と組み合わせることにより、容易に光強度を測定することが出来ます。

<S9705の特長>

- 光の強さを周波数に変換
- 広いダイナミックレンジ(5桁)
- 良好なリニアリティ
- 出力タイミングのリセット機能
- 出力はマイコンに直接接続可能
- 4ピンプラスチックパッケージ
- 電源電圧:2.7V~5.5V
- 50kHz(@2856K100lx)



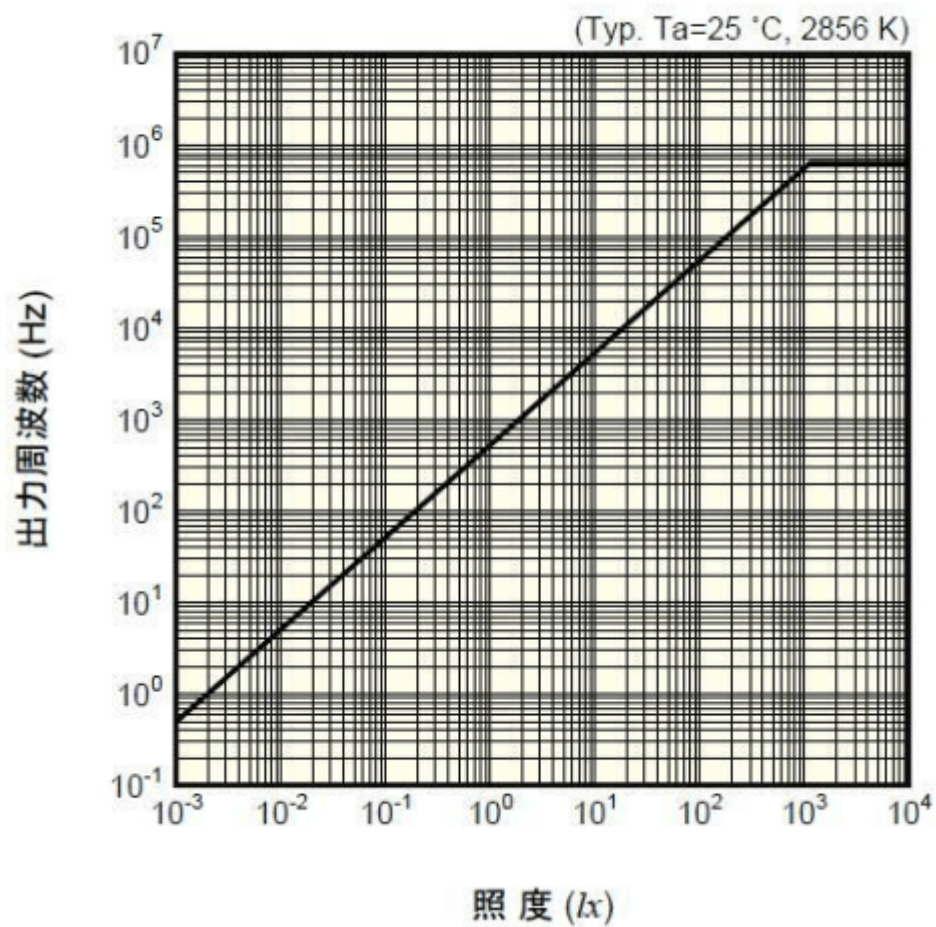
<S9705の概観>

動作原理

S9705は、照度に応じた周波数の矩形波を出力しますので、その周波数をカウントし、照度に変換します。100Luxで50KHzなので、周波数を500で割り算すれば照度を求めることが出来ます。

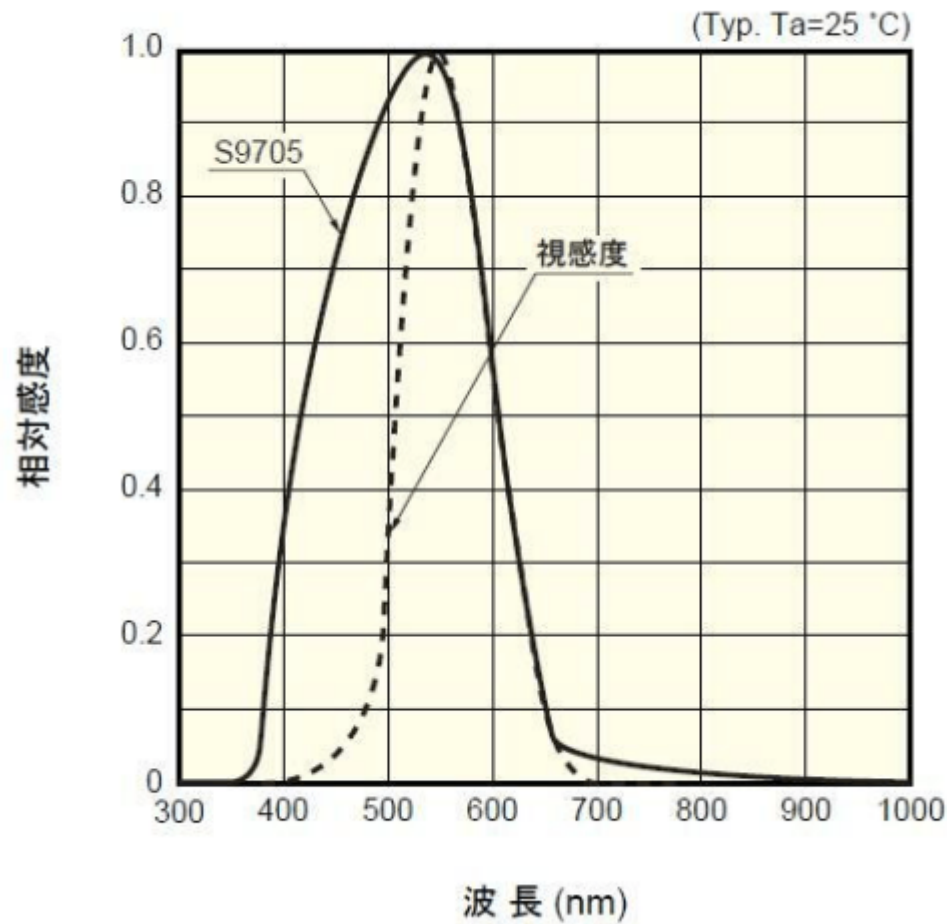
- 周波数をカウントし、LCDに表示する。
- その結果を500で割り算し、LCDに表示する。

■ 出力周波数－照度

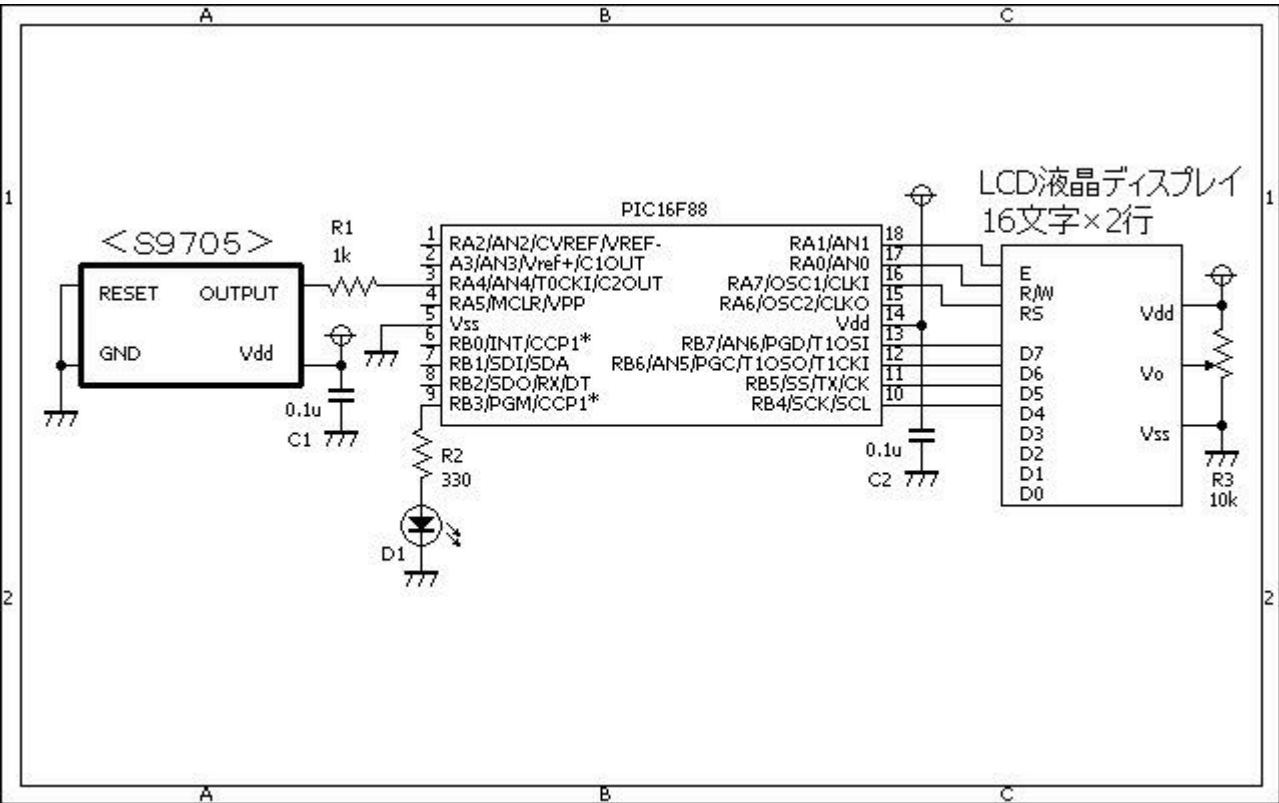


<S9705の特性>

■ 分光感度特性



回路図



ソースコード

プログラムの基本的な仕組みは、周波数カウンタと同じです。

LuxMeterV2.c

```
//*****
*
*
/*
<照度計V2>
■ 照度 周波数変換フォト○○○○○○○○
■ 主な仕様
  浜松ホトニクス製 照度センサ○○
  照度に応じた周波数を出力する○○です。
  周波数をマイコンでカウントするだけなので扱いやすい○○です。
  電源電圧○○○○○○○○○○
○○○○○○○○○○○○○○○○○○○○○○○○○○○○
  ・ 4ピン表面実装パッケージ
  広いダイナミックレンジ
*/
//*****
*

#define          LED          PORTB.F3

//*****
*

void interrupt()
{
  PIR1.TMR1IF = 0;
  //
  TRISA.F4 = 0;          // ゲートを閉める。
  PORTA.F4 = 0;
  T1CON.TMR1ON = 0;      // TIMER1を停止する。
  LED = ~LED;
}

//*****
*

unsigned long FreqMeasurement()
{
  unsigned long freq;
  //
  TRISA.F4 = 0;          // ゲートを閉める。
  PORTA.F4 = 0;
  // TIMER0の設定
  INTCON.T0IF = 0;
  TMR0 = 0;
```

```
// TIMER1の設定
PIR1.TMR1IF = 0;
TMR1L = 0x58;           // 25000=(0.1/8000000) * 4 * 8
TMR1H = 0x9E;           // 0x9E58=65536-25000
//
freq = 0;
// 割り込みを許可する。
INTCON.PEIE = 1;
INTCON.GIE = 1;
// 開始
T1CON.TMR1ON = 1;
// カウント遅延
asm    nop
asm    nop
asm    nop
asm    nop
asm    nop
asm    nop
asm    nop
asm    nop
asm    nop
asm    nop
asm    nop
asm    nop
asm    nop
asm    nop
asm    nop
asm    nop
//
TRISA.F4 = 1;           // ゲートを開ける。
// 測定
while (T1CON.TMR1ON != 0) {
    if (INTCON.T0IF == 1) {
        INTCON.T0IF = 0;
        freq++;
    }
}
if (INTCON.T0IF == 1) {
    INTCON.T0IF = 0;
    freq++;
}
freq = (freq * 256) + TMR0;
return (freq);
}

// *****
*

void main()
{
    float          freq;
    unsigned char  buf[12], cnt;
    //
```

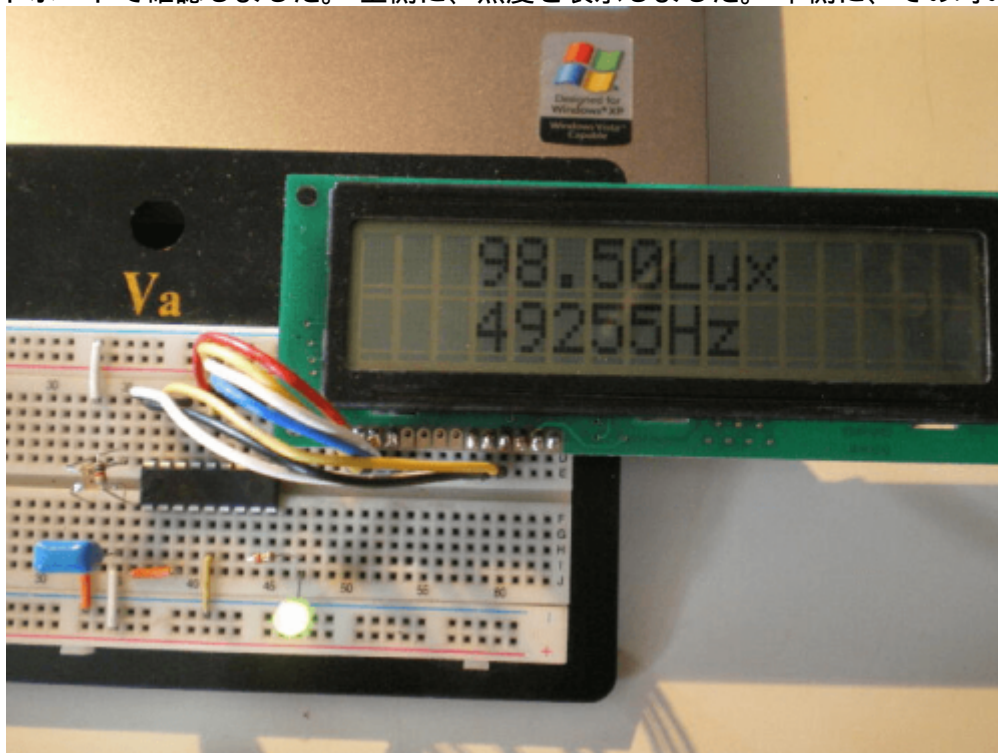
```

OSCCON = 0b01110000;    // 内臓クロックを8Mhzに設定する。
CMCON = 0b00000111;     // コンパレータは使用しない。
ANSEL = 0b00000000;     // A/D変換は使用しない。
TRISA = 0b01111100;
TRISB = 0b00000111;
OPTION_REG.NOT_RBPU = 0;
// TIMER0の設定
INTCON.T0IE = 0;
INTCON.T0IF = 0;
OPTION_REG.T0CS = 1;
OPTION_REG.PSA = 1;
TMR0 = 0;
// TIMER1の設定
PIE1.TMR1IE = 1;
PIR1.TMR1IF = 0;
T1CON.T1CKPS0 = 1;
T1CON.T1CKPS1 = 1;
T1CON.TMR1ON = 0;
//
Lcd_Custom_Config(&PORTB,7,6,5,4,&PORTA,7,0,1);
Lcd_Custom_Cmd(LCD_CURSOR_OFF);
Lcd_Custom_Cmd(LCD_CLEAR);
//
while (1) {
    freq = 0.0;
    for (cnt = 0; cnt < 10; cnt++) {
        freq += FreqMeasurement();
    }
    // 周波数の表示
    LongToStr(freq, buf);
    Lcd_Custom_Out(2, 1, &buf[3]);
    Lcd_Custom_Out(2, 9, "Hz");
    //
    freq = freq / 500.0 * 100.0;
    LongToStr(freq, buf);
    buf[3] = buf[4];
    buf[4] = buf[5];
    buf[5] = buf[6];
    buf[6] = buf[7];
    buf[7] = buf[8];
    buf[8] = '.';
    Lcd_Custom_Out(1, 1, &buf[3]);
    Lcd_Custom_Out(1, 9, "Lux");
}
}

// *****
*
```

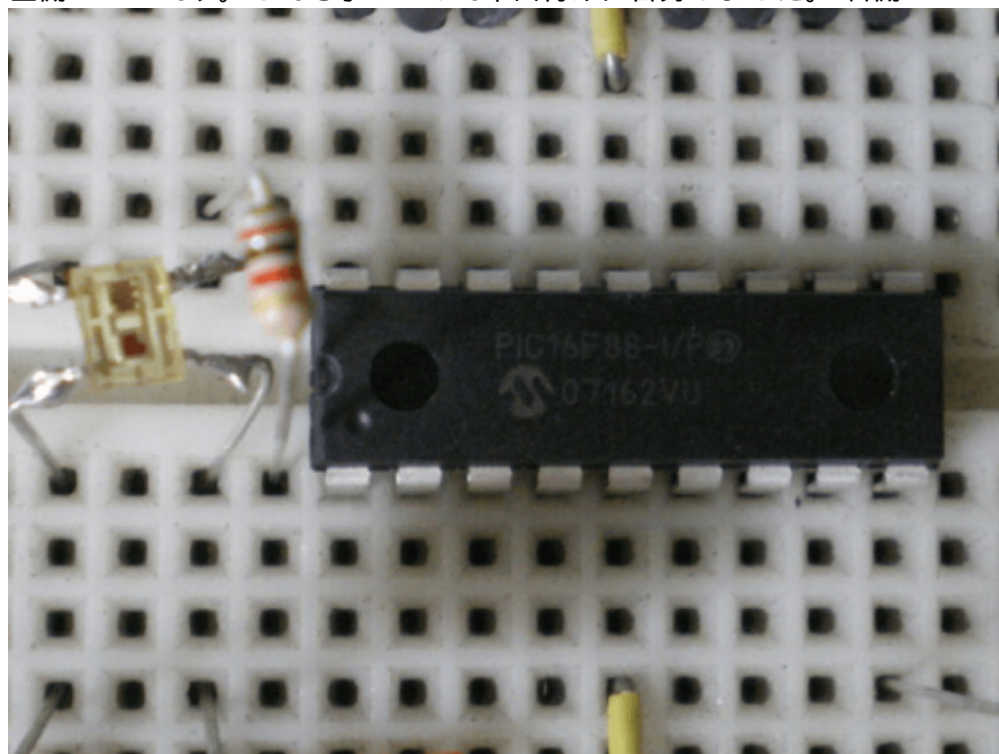
動作確認

いつものブレッドボードで確認しました。上側に、照度を表示しました。下側に、その時の周波数を

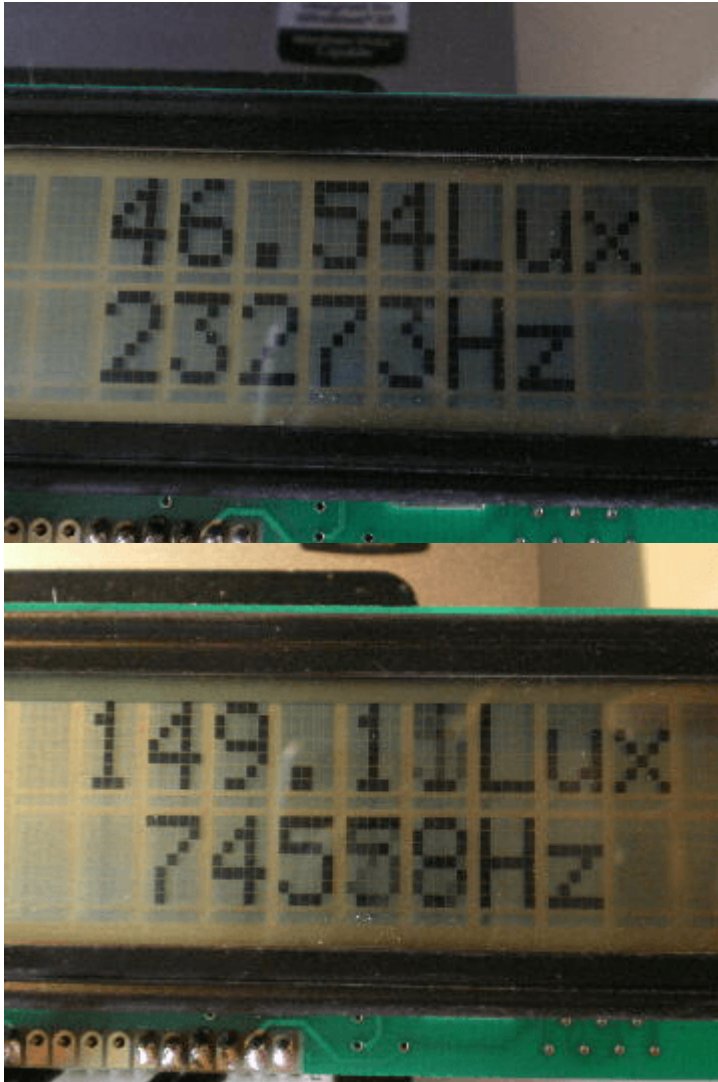


表示しました。

左側:S9705です。とても小さいので半田付けに苦労しました。 右側:PIC16F88です。



小屋の中の照明を、暗くしたり、明るくしたりしてみました。



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://deepsky.jp/wiki/> - うごくといいな

Permanent link:

<http://deepsky.jp/wiki/doku.php?id=elechobby:picdic:pic16f88:93>

Last update: **2020/05/01 18:51**

