

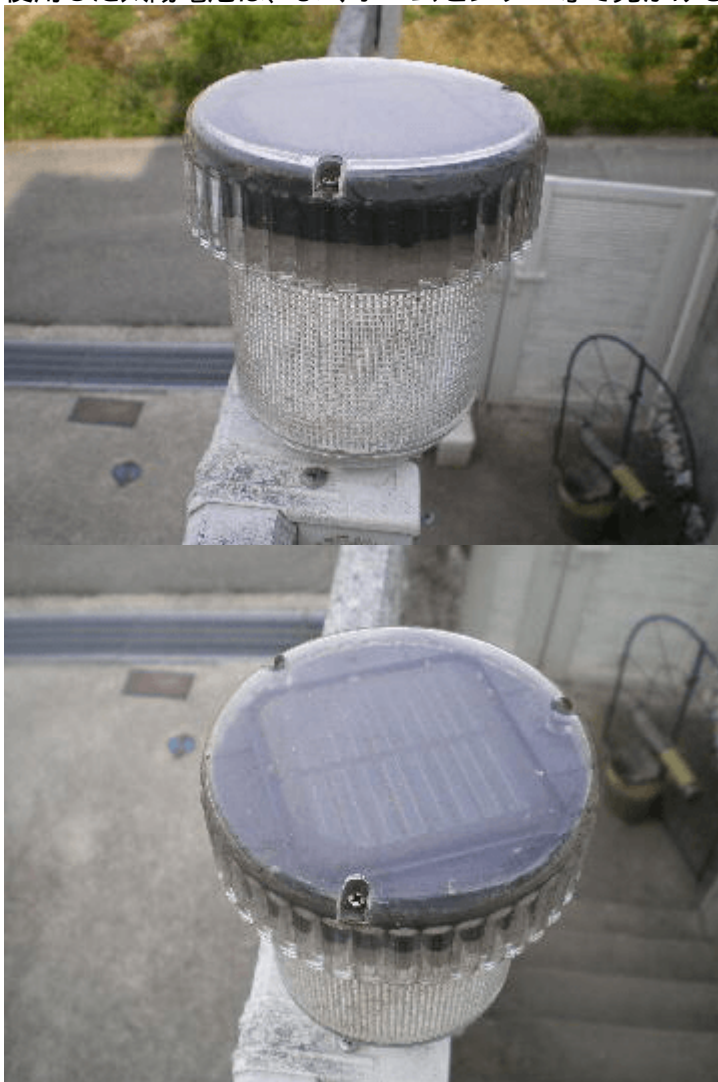
常夜灯V2(太陽電池 + ニッカド電池)

概要

以前にも常夜灯(フルカラーLED)を製作しましたが、乾電池駆動のため電池の交換の手間が必要でした。今回は、乾電池を使用せずに、太陽電池とニッカド電池の組み合わせで、尚且つ、電圧は2V以上で動作可能としました。PIC自身は、2V以下でも動作(PIC最低動作電圧)しますがLEDの点灯がありますので2V以上としました。

動作原理

使用した太陽電池は、よくホームセンター等で見かける太陽電池式の常夜灯から取り外し使用しました。

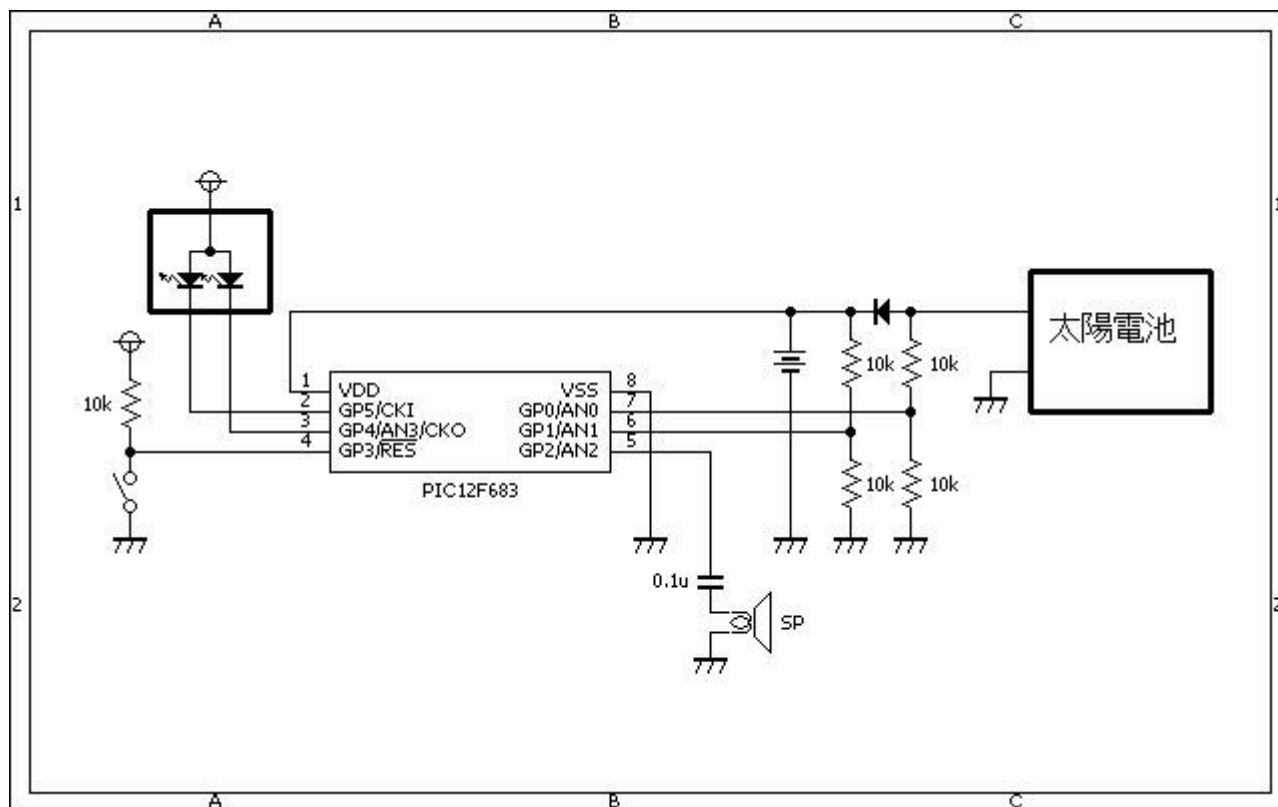


<ニッカド電池の充電> 太陽電池でニッカド電池を充電します。その時にニッカド電池からの逆流を防ぐためにダイオードを間に入れます。今回使用した太陽電池は最大4V近く発電します。<LEDの点灯> 辺りが暗くなるとLED(2色)を点灯させ、ブザー音を鳴らせます。LEDの点灯は、単なるON/OFFでは面白くありませんのでPWM方式により、明暗のレベルを10段階設定可能としました。つまり2色のLEDを組み合わせるので100色に変化します。尚、PWMは、PICのTIMER0モジュールを利用し、割り込み処理の中でソフトウェアで制御しました。<明るさの検出> 辺りが暗くなることを検出する方法は、通常はCDS等のセンサーを使いますが、今回は、太陽電池の電圧とニッカド電池の電圧の電位差により判断します。

- 明るい:太陽電池の電圧>ニッカド電池の電圧
- 暗い:太陽電池の電圧<ニッカド電池の電圧

<ブザー音> ブザー音は、PIC内臓のPWMモジュールを使用しました。単音では面白くありませんので□500Hz□1kHz□2kHz□4kHz□8kHzと0.1秒置きに変化させました。スイッチにより鳴らせるかどうかの切替が可能です。

回路図



ソースコード

[NigthLightV2.c](#)

```
//*****  
*  
#define LED1 GPIO.F4  
#define LED2 GPIO.F5  
  
#define SW GPIO.F3  
  
#define ON 0  
#define OFF 1  
  
//*****  
*
```

```
static unsigned char LED_cnt, LED1_cnt, LED2_cnt;

void interrupt(){
    if (INTCON.T0IF == 1) {
        INTCON.T0IF = 0;
        GPIO.F1 = ~GPIO.F1;
        //
        if ((LED_cnt <= LED1_cnt) && (LED1_cnt != 0))
            LED1 = ON;
        else
            LED1 = OFF;
        //
        if ((LED_cnt <= LED2_cnt) && (LED2_cnt != 0))
            LED2 = ON;
        else
            LED2 = OFF;
        //
        if (LED_cnt < 9)
            LED_cnt++;
        else
            LED_cnt = 0;
    }
}

//*****
*

void Pwm_Change_DutyEx(unsigned int duty_ratio)
{
    CCP1L = duty_ratio >> 2;
    CCP1CON.F6 = duty_ratio & 0b00000001;
    CCP1CON.F7 = (duty_ratio & 0b00000010) >> 1;
}

//*****
*

void buzzer()
{
    T2CON.T2CKPS0 = 0;
    T2CON.T2CKPS1 = 0;
    Pwm_Init(500);
    Pwm_Change_DutyEx((PR2 * 4) / 2);
    Pwm_Start();
    Delay_ms(100);
    Pwm_Stop();
    //
    T2CON.T2CKPS0 = 0;
    T2CON.T2CKPS1 = 0;
    Pwm_Init(1000);
}
```

```
Pwm_Change_DutyEx((PR2 * 4) / 2);
Pwm_Start();
Delay_ms(100);
Pwm_Stop();
//
T2CON.T2CKPS0 = 0;
T2CON.T2CKPS1 = 0;
Pwm_Init(2000);
Pwm_Change_DutyEx((PR2 * 4) / 2);
Pwm_Start();
Delay_ms(100);
Pwm_Stop();
//
T2CON.T2CKPS0 = 0;
T2CON.T2CKPS1 = 0;
Pwm_Init(4000);
Pwm_Change_DutyEx((PR2 * 4) / 2);
Pwm_Start();
Delay_ms(100);
Pwm_Stop();
//
T2CON.T2CKPS0 = 0;
T2CON.T2CKPS1 = 0;
Pwm_Init(8000);
Pwm_Change_DutyEx((PR2 * 4) / 2);
Pwm_Start();
Delay_ms(100);
Pwm_Stop();
}

//*****
*

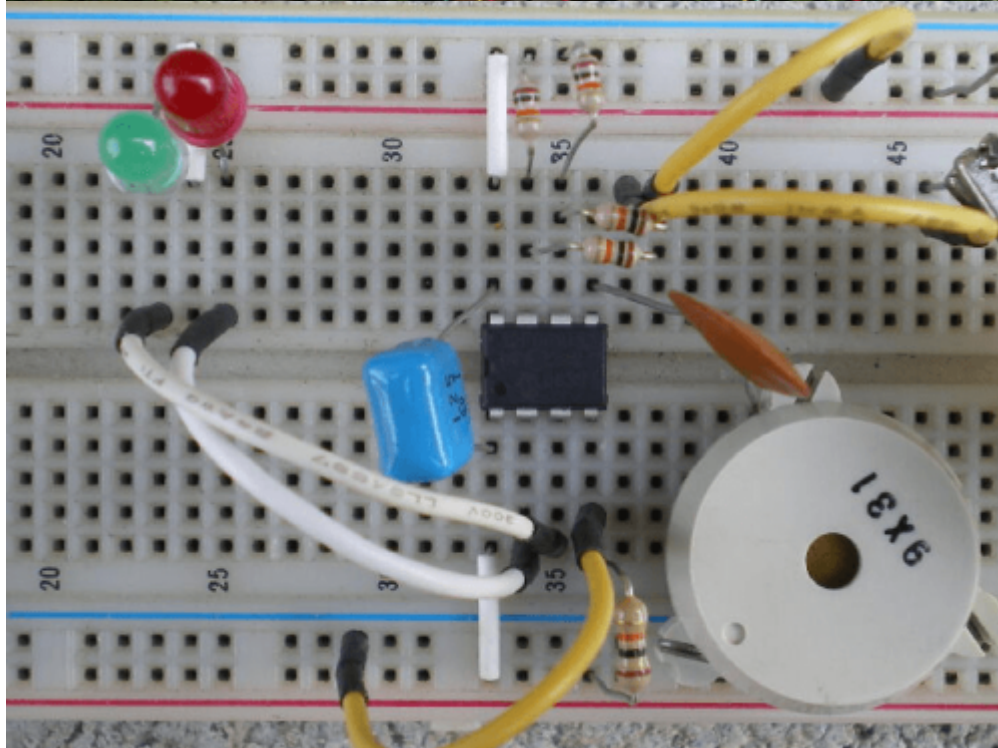
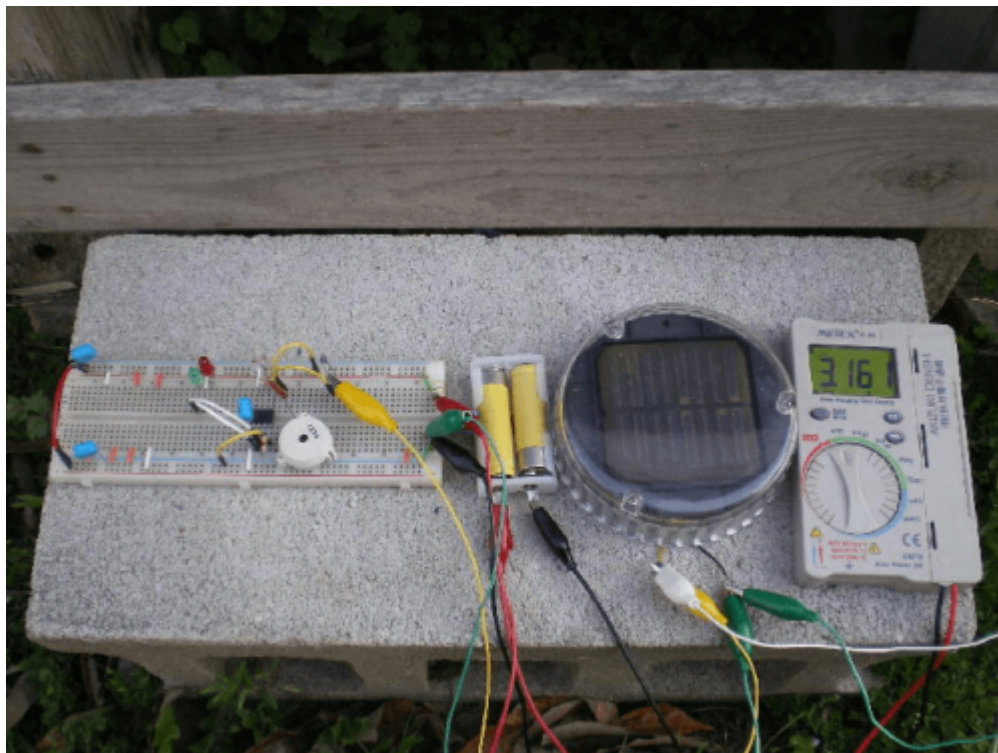
void main()
{
    unsigned int    ad0, ad1, tmp;
    OSCCON = 0b01111000; // クロックは8Mhz
    CMCON0 = 0b00000111; // コンパレータは使用しない。
    ANSEL  = 0b00000011; // AN1を使用する。
    TRISIO = 0b00001011;
    //
    INTCON.T0IE = 1;
    INTCON.T0IF = 0;
    OPTION_REG.T0CS = 0;
    OPTION_REG.PSA = 1;
    OPTION_REG.PS0 = 0;
    OPTION_REG.PS1 = 0;
    OPTION_REG.PS2 = 0;
    //
    LED_cnt = 0;
```

```
LED1_cnt = 0;
LED2_cnt = 0;
LED1 = OFF;
LED2 = OFF;
//
INTCON.PEIE = 1;    // これ以降の処理で割り込みを許可する。
INTCON.GIE = 1;    // これ以降の処理で割り込みを許可する。
//
while (1) {
    ad0 = Adc_Read(0);    // 太陽電池側
    ad1 = Adc_Read(1);    // ニッカド電池側
    if (ad0 > ad1) {
        LED1_cnt = 0;
        LED2_cnt = 0;
        continue;
    }
    //
    tmp = rand();
    LED1_cnt = (((double)tmp) / 32768.0) * 10.0;
    tmp = rand();
    LED2_cnt = (((double)tmp) / 32768.0) * 10.0;
    Delay_ms(1000);
    LED1_cnt = 0;
    LED2_cnt = 0;
    if (SW == 0)
        buzzer();
    Delay_ms(1000);
}

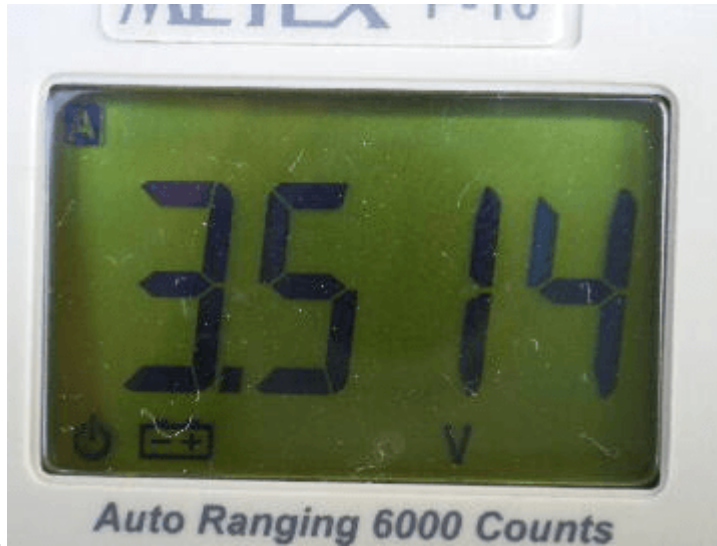
//*****
*
```

動作確認

いつものブレッドボードに組み込んで屋外で動作を確認しました。左から、制御部(LED/PIC/圧電スピーカ)、ニッカド電池、太陽電池、テストです。



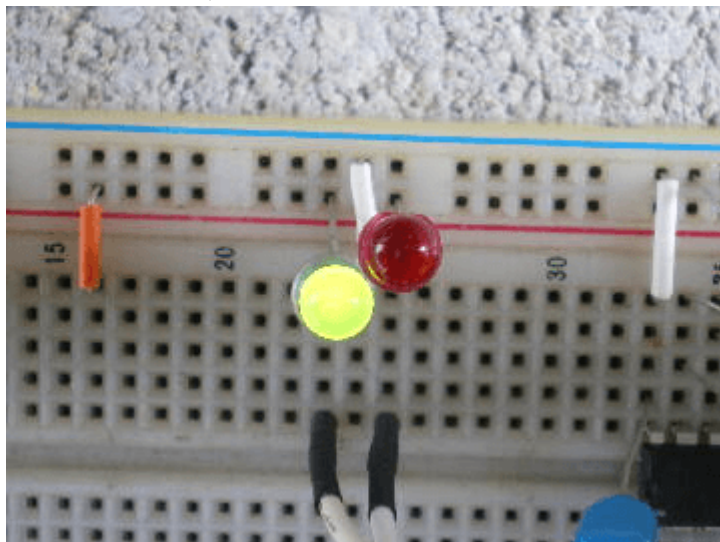
左側:太陽電池の電圧 右側:



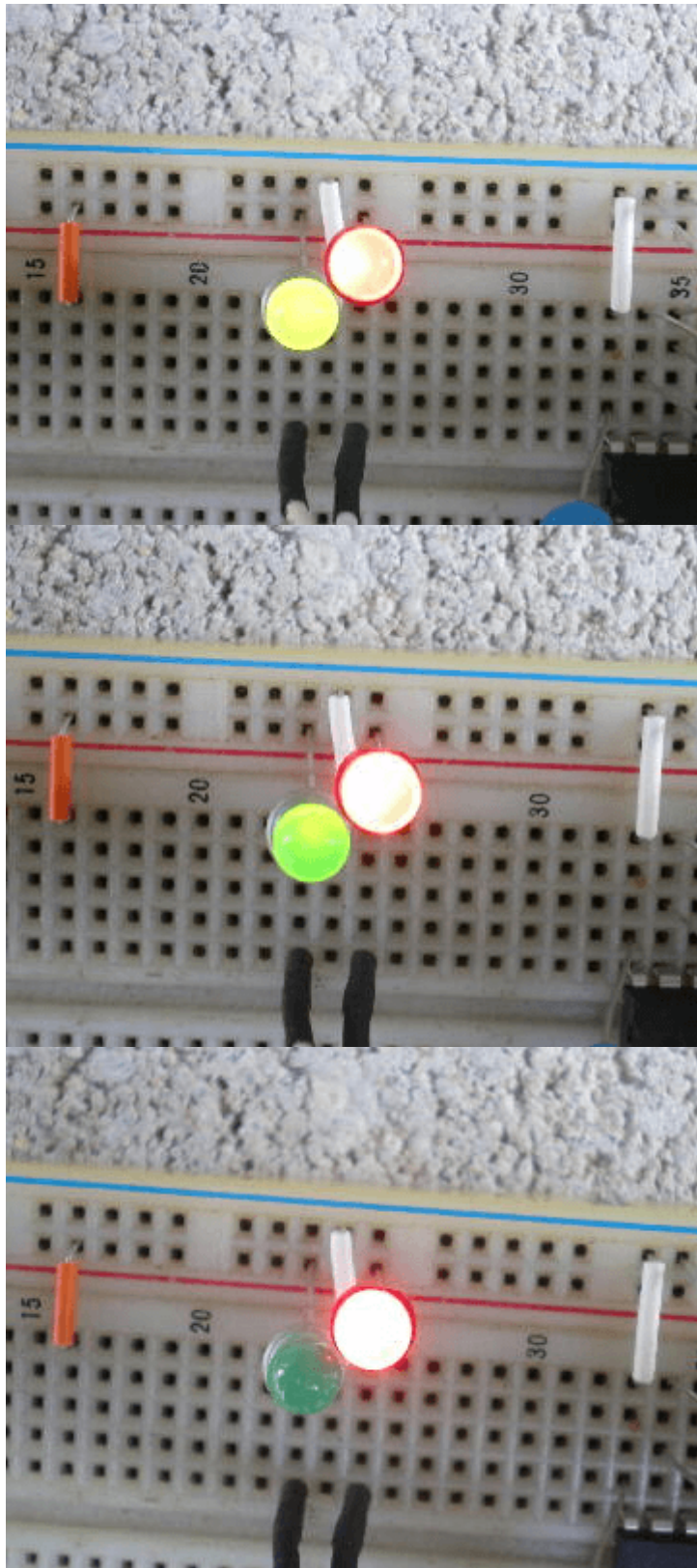
ニッカド電池の電圧

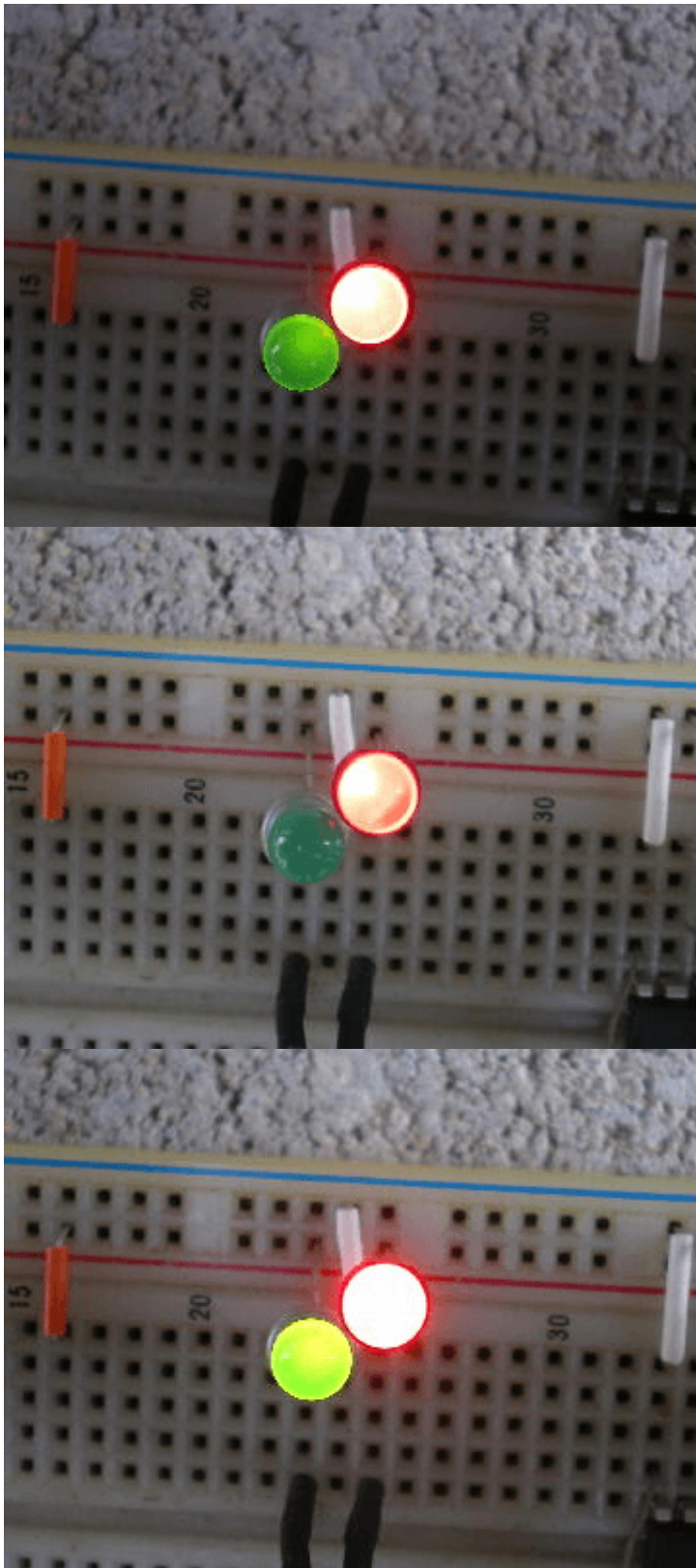


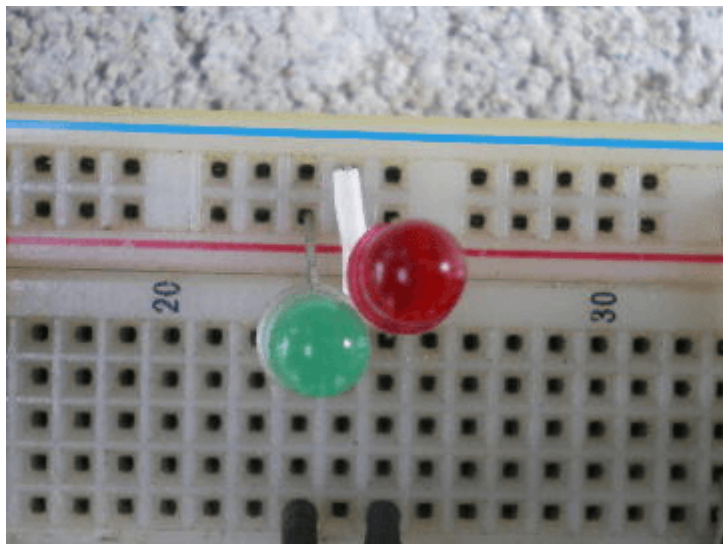
太陽電池に覆いを被せると太陽電池の電圧とニッカド電池の電圧の電位差が逆転しLEDの点灯が始まります。本来は、2色LEDを使用し、光拡散キャップを被せると良いのですが、手持ちがありませんので赤と緑のLEDを使用しました。いろいろな色に変



化します。







如何ですか? これをホームセンターの常夜灯の内部に組み込めば実用レベルに仕上がると思います。{ 😊 }!

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:pic12f683:15&rev=1588065786>

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

