

簡易ストロボフラッシャーV2(キセノン管)

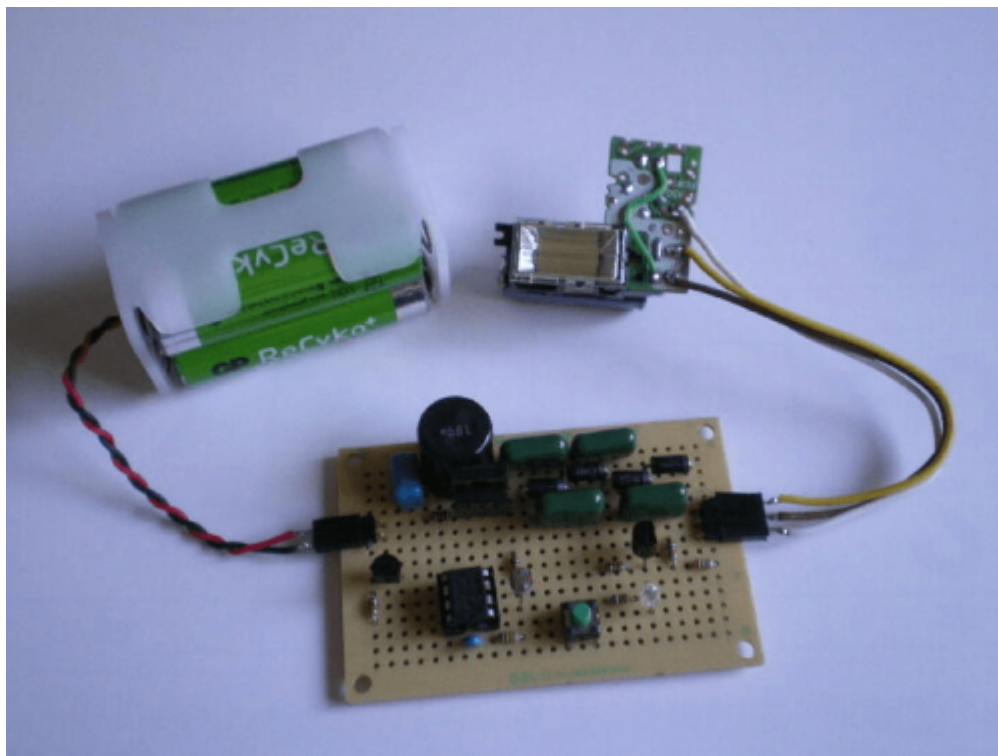
概要

前回、簡易ストロボフラッシャー(キセノン管)を製作しましたが、消費電流が大きいため長時間駆動には向きませんでした。

そこで今回は、消費電流を抑えて長時間駆動が可能な「ストロボフラッシャー」を製作しました。

<仕様>

- 夜間のみ動作させます。(基準照度>現時点の照度)
- 基準照度は、スイッチで任意に設定可能とします。
- ストロボ機構には、安価(500円前後)な使い捨てカメラ「写るんです」の一部の回路を使用します。



動作原理

周囲が暗くなると高電圧を発生させ、電圧が250V以上になるとストロボをフラッシュ(閃光)させます。

動作原理(ハードウェア)

高電圧発生 詳細は、高電圧発生回路を参照してください。

ストロボ機構

- トリガ回路と放電管は、「写るんです」の回路を一部利用します。

詳細は、簡易ストロボフラッシャー(キセノン管)を参照してください。

- SCRでトリガ回路をオンにします。

手持ちのNEC社製のSCR(P2M)を使用しました。SCRは、サイリスタ (Thyristor)とも呼ばれています。秋月電子で販売している「高感度小型サイリスタMCR22-8」でも代用可能です。(8個200円)

- トリガトランス経由で、高電圧パルスをキセノン管を囲む反射板に与えます。
- キセノン管は励起され放電(発光)を開始します。

照度の検出

- CDSと抵抗を利用した簡易回路です。
- 周囲が明るくなると電圧が上昇します。
- 周囲が暗くなると電圧が下降します。

動作原理(ソフトウェア)

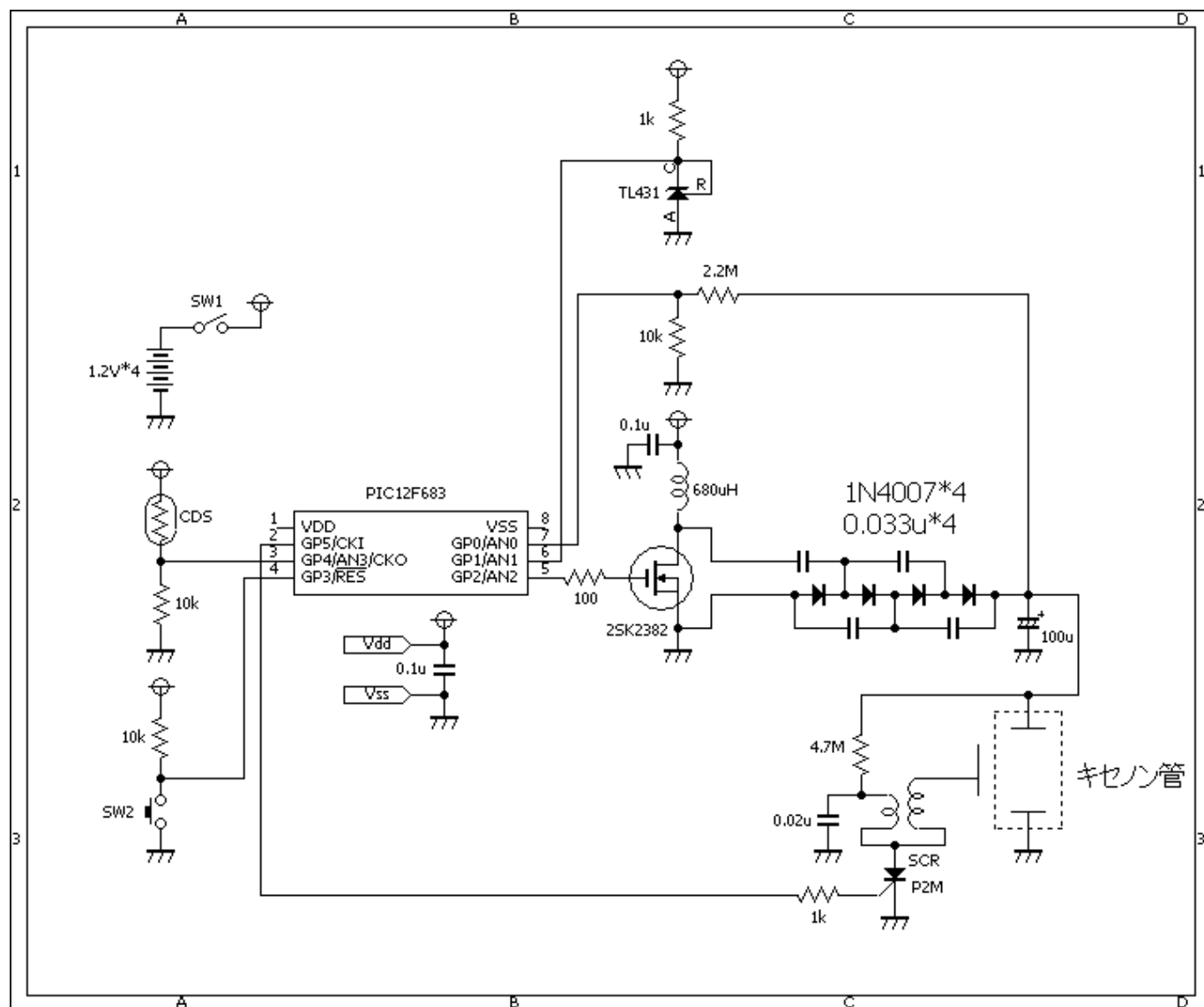
メイン関数(main)

- ADCを初期化します。
- PWMを初期化します。
- EEPROM内の基準照度(Lux1)を読み込みます。
- 照度(Lux2)を測定します。
- 基準照度(Lux1)と照度(Lux2)を比較し、照度(Lux2)が小さい場合、周囲が暗くなったと判断します。
- フラッシュ点灯関数を呼び出します。
- スイッチ(SW2)が押下されると、照度を測定し、EEPROMに基準照度として書き込みます。

フラッシュ点灯関数(flash_proc)

- 高電圧発生用のPWMをオンします。
- 電圧が250V以上になる迄待ちます。
- ストロボをフラッシュさせます(SCRのゲートに10msec間、電圧を印加します)
- 高電圧発生用のPWMをオフします。

回路図



ソースコード

[flasher_v2_00.c](#)

```
//*****
*
/*
【簡易フラッシャー】
*/
//*****
*
#define BYTE    unsigned    short
#define WORD    unsigned    int
#define DWORD    unsigned    long
//
#define FLASH    GPIO.B5
#define ON        1
#define OFF        0
//
```

```
#define SW      GPIO.B3
//*****
*
extern void    main();
extern void    flash_proc();
extern int     ad_measurement(unsigned short channel);
extern void    PWM1_custom_Set_Duty(WORD duty_ratio);
extern void    led_blink(short cnt);
//*****
*
//■■■メイン関数■■■
void    main()
{
    int      ad, threshold;
    //
    OSCCON = 0b01110000;    //clock=8MHz
    CMCON0 = 0b00000111;
    ANSEL  = 0b00001011;
    TRISIO = 0b00011011;
    FLASH = OFF;
    //ADC初期化
    ADC_Init();
    ADCON0.VCFG = 1;
    //PWM初期化
    PWM1_Init(5000);        //pwm=5kHz
    PR2 = 0xFF;
    PWM1_custom_Set_Duty(150);    //duty=15%
    PWM1_Stop();
    //LED点滅
    led_blink(10);
    //基準照度測定
    //threshold = ad_measurement(3);
    //基準照度読み込み
    threshold = EEPROM_Read(1);
    threshold = (threshold << 8) | EEPROM_Read(0);
    //
    while (1) {
        //照度測定
        ad = ad_measurement(3);
        //基準照度と照度の比較
        if (threshold > ad) {
            led_blink(1);
            //
            flash_proc();    //フラッシュ点灯
        }
        //基準照度測定&書き込み
        if (SW == 0) {
            while (SW == 0) {
                led_blink(1);
            }
            led_blink(10);
        }
    }
}
```

```

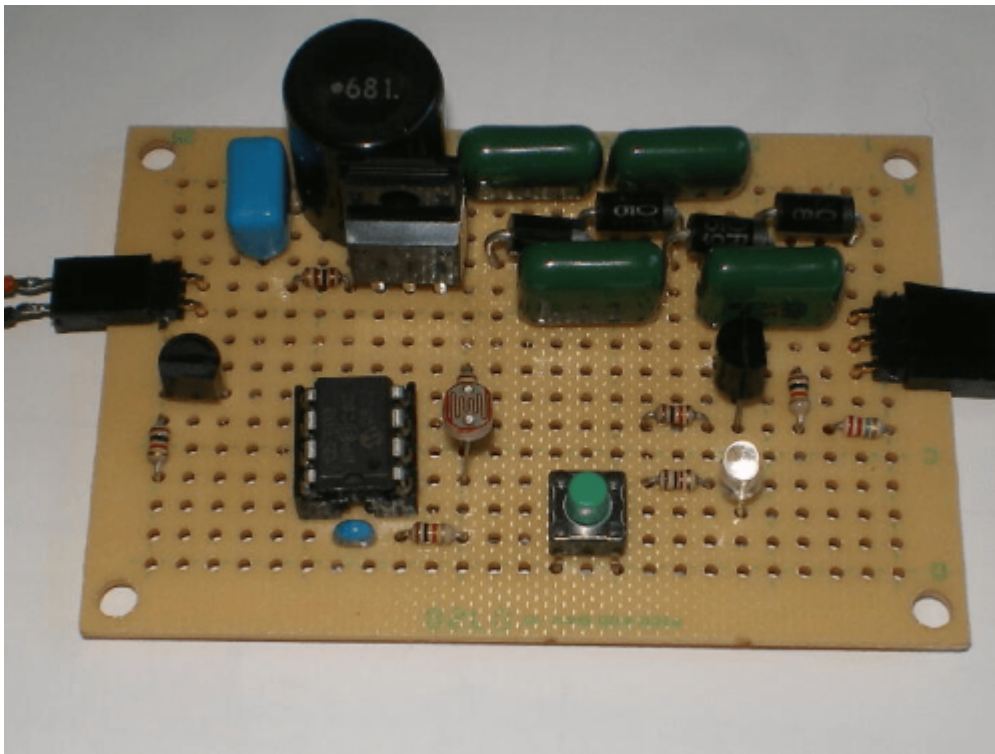
        threshold = ad_measurement(3);
        EEPROM_Write(0, threshold & 0xFF);
        EEPROM_Write(1, (threshold >> 8) & 0xFF);
    }
}

//*****
*
//■■■フラッシュ点灯関数■■■
void    flash_proc()
{
    double  hv;
    //
    PWM1_Start();
    //
    hv = 0.0;
    while (hv < 250000.0) {
        hv = ad_measurement(0);
        hv = hv * 2.4365234375 * 221.0;
    }
    //
    FLASH = ON;
    Delay_ms(10);
    FLASH = OFF;
    //
    PWM1_Stop();
}
//*****
*
//■■■アナログデータ測定関数■■■
int      ad_measurement(unsigned short channel)
{
    long    ad;
    int      cnt;
    //
    ad = 0;
    for (cnt = 0; cnt < 100; cnt++) {
        ad += ADC_Get_Sample(channel);
    }
    return (ad /= 100);
}
//*****
*
//■■■デューティ設定関数■■■
void      PWM1_custom_Set_Duty(WORD duty_ratio)
{
    CCP1L = duty_ratio >> 2;
    CCP1CON.DC1B0 = (duty_ratio & 0b00000001) != 0 ? 1 : 0;
    CCP1CON.DC1B1 = (duty_ratio & 0b00000010) != 0 ? 1 : 0;
}
//*****

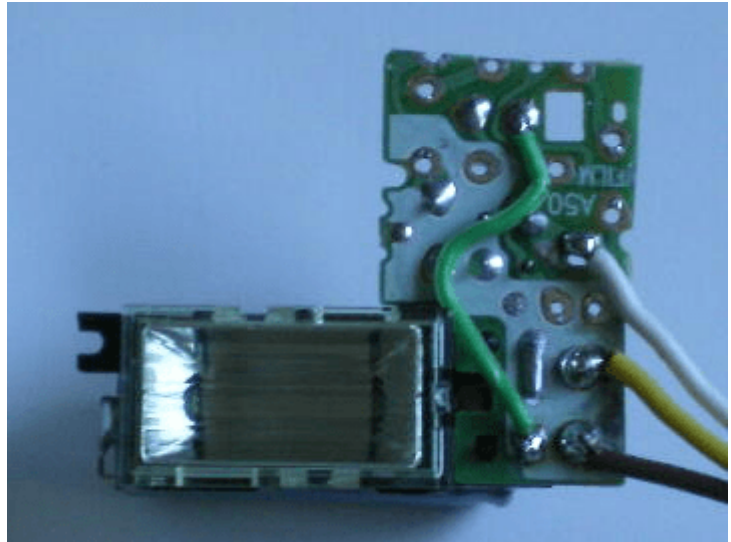
```

```
*
//■■■■点滅関数■■■
void    led_blink(short cnt)
{
    while (cnt > 0) {
        FLASH = ON;
        Delay_ms(100);
        FLASH = OFF;
        Delay_ms(100);
        //
        cnt--;
    }
}
//*****
*
```

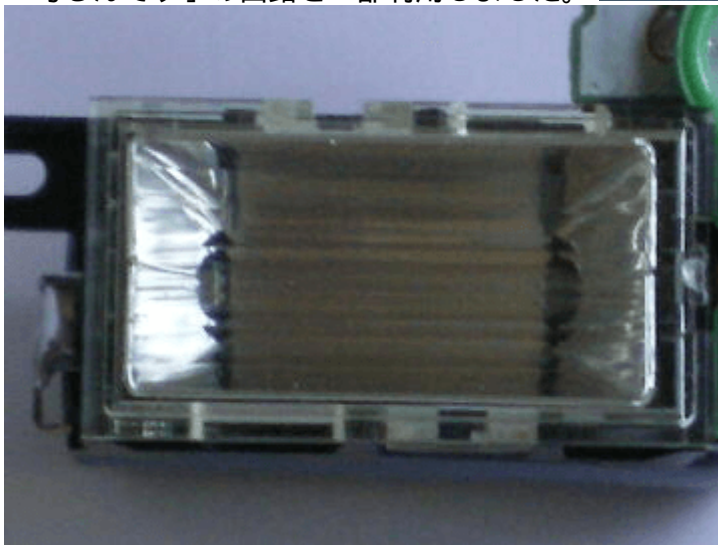
動作確認



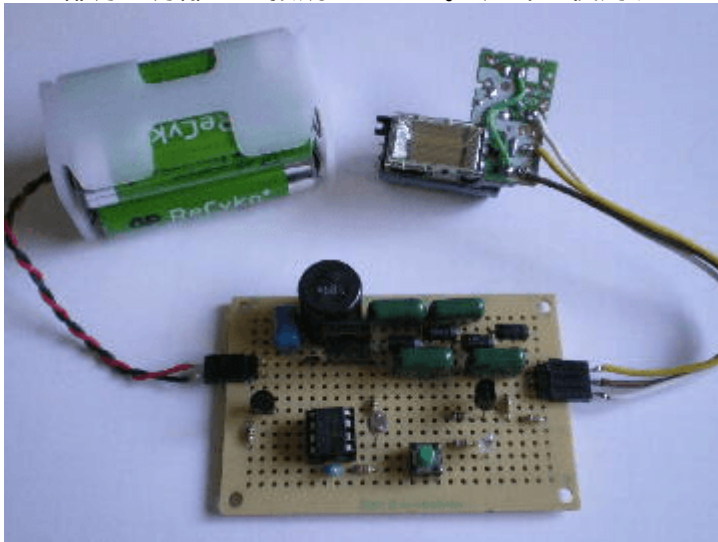
トリガ回路と放電管は、



「写るんです」の回路を一部利用しました。



基板と電池ボックス(単三電池1.2V×4本)とストロボ部分を分離して接続しました。屋外で使用するためにガラス瓶に入れました。





畑に設置して使用しています。お蔭様でスイカも動物被害にあわず無事に収穫することが出来ました!?



高電圧を取り扱いますので、製作時には感電にご注意ください。

From:

<http://deepsky.jp/wiki/> - うごくといいな

Permanent link:

<http://deepsky.jp/wiki/doku.php?id=elechobby:picdic:pic12f683:42&rev=1588141919>

Last update: 2025/10/17 14:27

