

ミニ電圧低下検出ユニット

概要

電源に商用電源(ACアダプタ、トランスなど)を利用している場合は特に気にしなくても良いのですが、電池やバッテリーを利用している場合には、電圧の低下が気になります。

そこで事前に、設定した電圧(閾値)よりも、電圧が下がると、ブザーやリレ \square LEDなどで通知するユニットを製作しました。

<仕様>

- 入力電圧範囲は、0V~27Vとする。
- 閾値は、スイッチで設定(変更)できる。
- ヒステリシス特性(閾値+100mV)を持たせ、制御のバタツキを防止する。
- 電池2本(3V)駆動とする。
- 出力は、ブザーと電圧の2種類とする。

動作原理

対象とする、装置の電源とは別に、本ユニットの電源は電池(2本:3V)で動作させる。

<入力電圧の測定>

1. 抵抗で分圧し、0V~約27Vまでを入力電圧範囲とする。(分解能は約27mV)
2. A/D変換用の基準電圧は、2.5Vとする。
3. PIC内臓のA/D変換モジュールで、電圧をデジタルデータに変換する。
4. デジタルデータを分解能で換算し、入力電圧(V)も求める。

<閾値の設定>

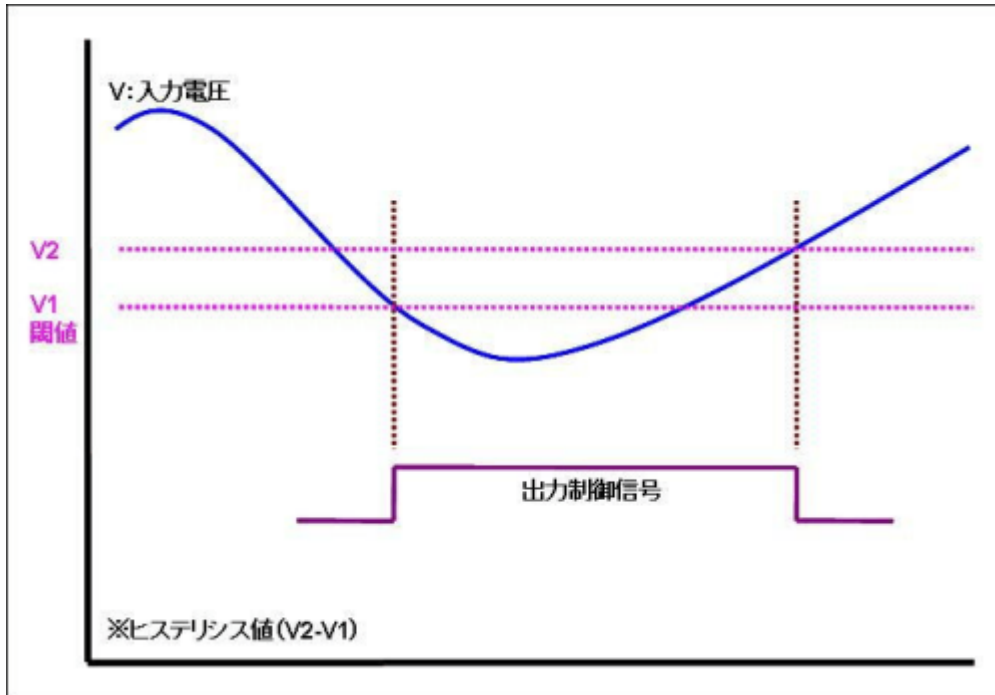
1. 入力電圧(V)を測定する。
2. 設定スイッチが押下されると、入力電圧(V)を閾値(V1)として設定する。
3. この閾値は、PICのEEPROMに記憶し、電源投入後に記憶した値を使用する。

<閾値との比較(1)>

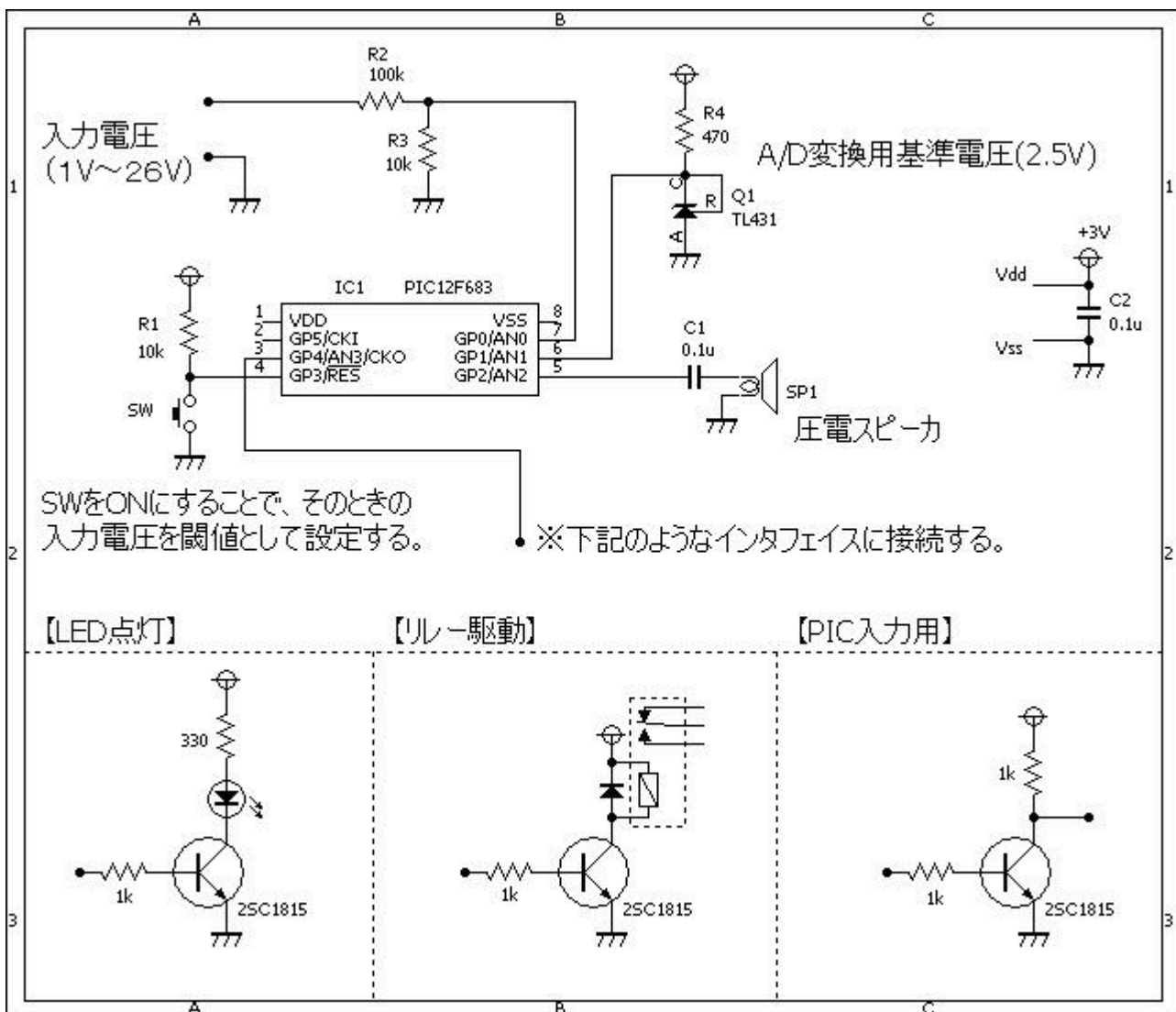
1. 入力電圧(V)を測定する。
2. 閾値(V1)と比較し、 $V < V1$ であれば、出力信号をONにする。
出力信号によって \square LEDの点灯、リレーの駆動 \square PICへの入力などが考えられる。

<閾値との比較(2)>

1. 入力電圧(V)を測定する。
2. (閾値(V1)+100mV)と比較し、 $V > (V1+100mV)$ であれば、出力信号をOFFにする。
このようにヒステリシス特性を持たせることによって、制御のバタツキを防止する。



回路図



ソースコード

VoltageDrop.c

```
//*****
*
*/
<電圧低下検出ユニット>
*/
//*****
*

#define      SW      GPIO.F3

//*****
*

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

//*****
*

void main()
{
    static    double      ad;
    static    unsigned    int      threshold;
    static    short       cnt, mode;
    //
    TRISIO = 0b00001011;
    OSCCON = 0b01010000; // クロックは2Mhz
    CMCON0 = 0b00000111; // コンパレータは使用しない。
    ANSEL  = 0b00000001; // □□□変換を使用する。
    ADCON0.VCFG = 1;
    //
    GPIO.F4 = 0;
    //
    Pwm_Init(1200);
    Pwm_Change_DutyEx((PR2 * 4) / 2);
    Pwm_Stop();
    // 保存されている閾値の取り込み
    threshold = Eeprom_Read(1);
    threshold = threshold << 8;
    threshold = threshold | Eeprom_Read(0);
    //
    mode = 0;
}
```

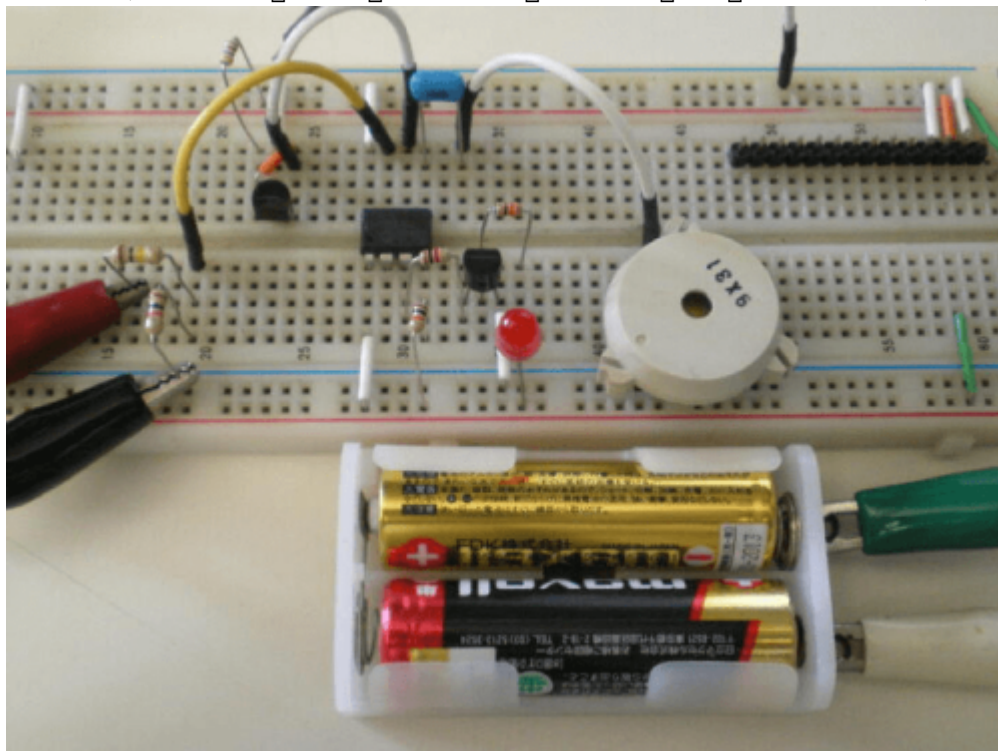
```
GPIO.F4 = 0;
//
for (cnt = 0; cnt < 5; cnt++) {
    Pwm_Start();
    Delay_ms(100);
    Pwm_Stop();
    Delay_ms(500);
}
//
while (1) {
    //電圧を測定する。
    ad = 0.0;
    for (cnt = 0; cnt < 100; cnt++) {
        ad += Adc_Read(0);
    }
    ad = (ad / 100) * 11 * 2.44140625;
    //閾値との比較を行う。
    switch (mode) {
    case 0:
        if (ad < threshold) {
            mode = 1;
            GPIO.F4 = 1;
        }
        break;
    case 1:
        if (ad > (threshold + 100.0)) {
            mode = 0;
            GPIO.F4 = 0;
        }
        break;
    }
    //電圧が低下したらブザーを鳴らす。
    if (mode == 1) {
        Pwm_Start();
        Delay_ms(100);
        Pwm_Stop();
        Delay_ms(500);
    }
    //
    if (SW == 0) {
        threshold = ad;
        // 閾値の保存
        Eeprom_Write(0, (threshold & 0xFF));
        Delay_ms(20);
        Eeprom_Write(1, ((threshold >> 8) & 0xFF));
        Delay_ms(20);
    }
}

//*****
```

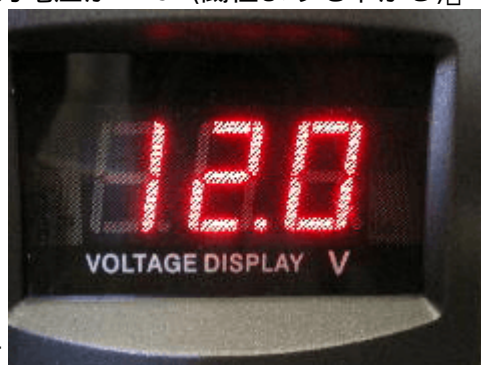
*

動作確認

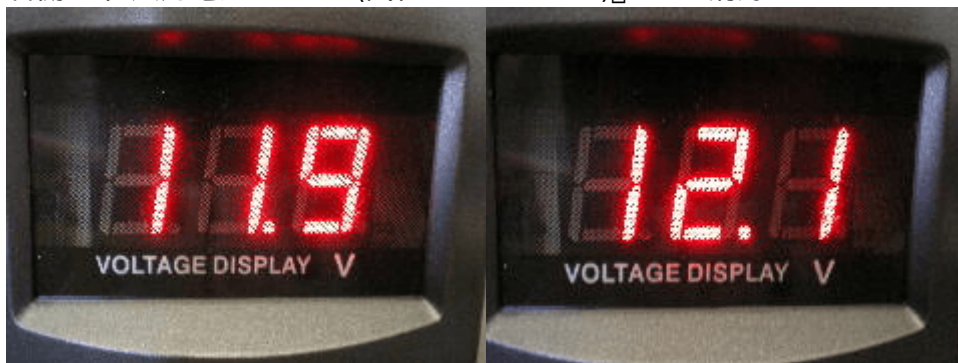
左側より、分圧抵抗、TL431、PIC12F683、2SC1815、LED、圧電スピーカ、電池2本

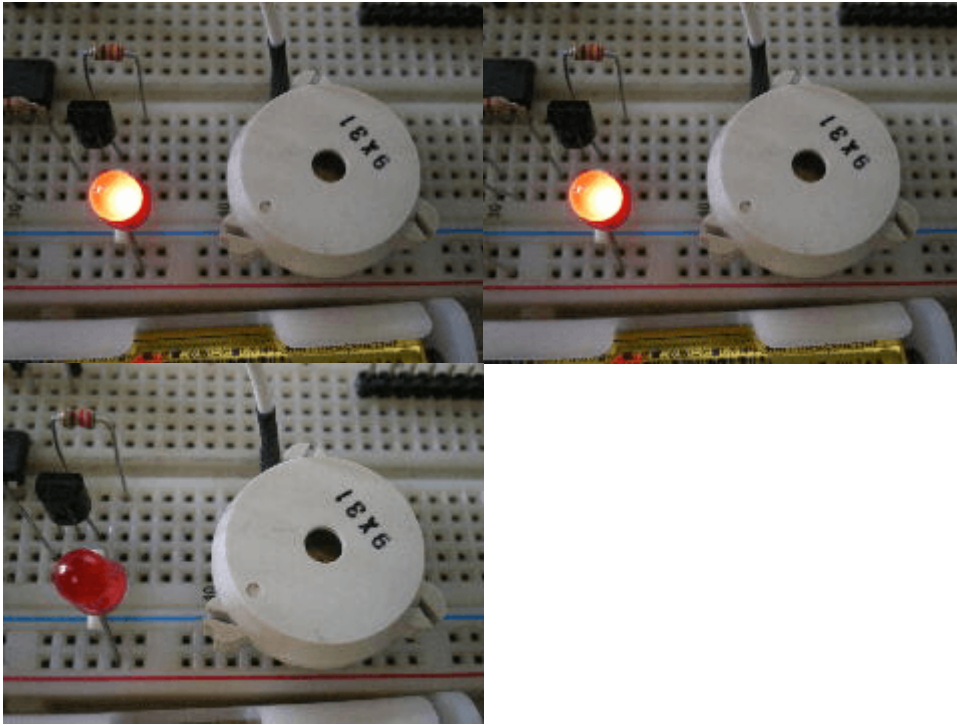


左側上下:閾値を12.0Vに設定するLEDは点灯 中央上下:入力電圧が11.9V(閾値よりも下がる)LEDは点灯

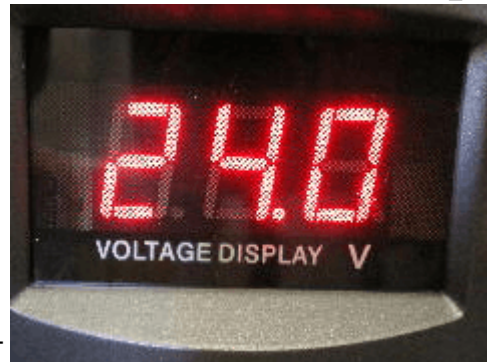


右側上下:入力電圧が12.1V(閾値よりも上がる)LEDは消灯

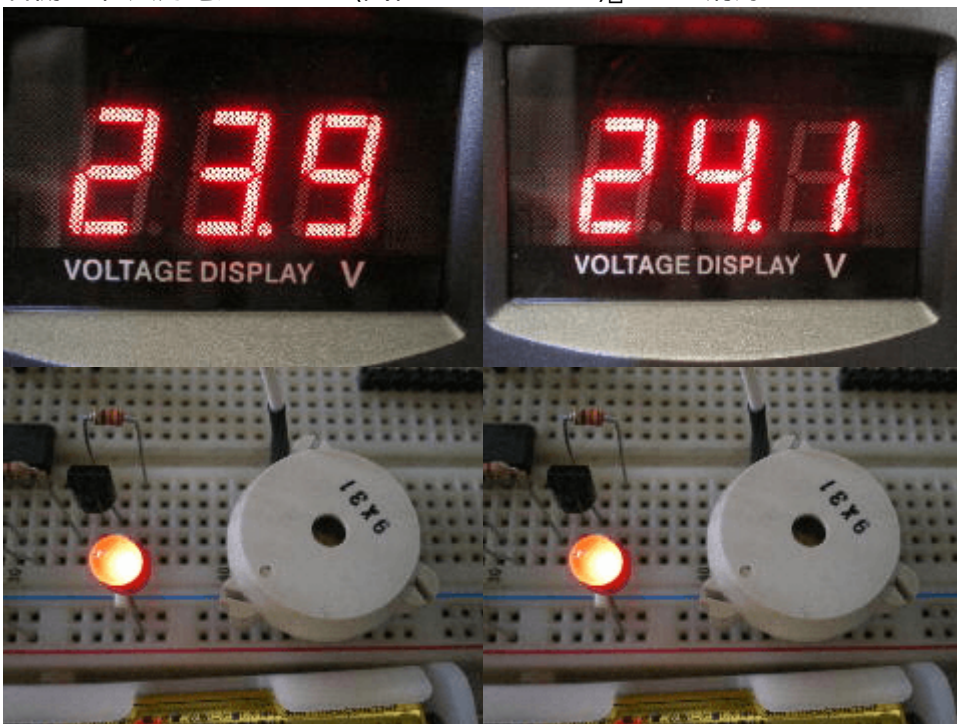


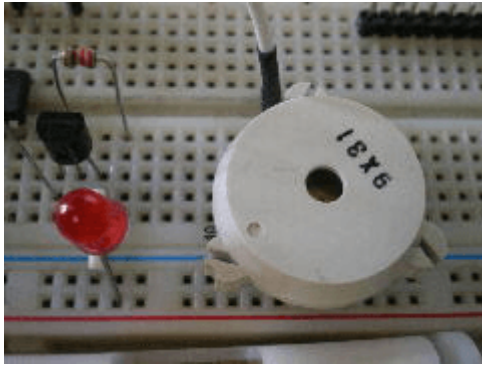


左側上下:閾値を24.0Vに設定するLEDは点灯 中央上下:入力電圧が23.9V(閾値よりも下がる)LEDは点灯

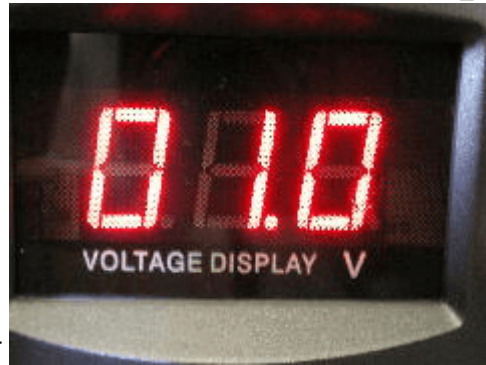


右側上下:入力電圧が24.1V(閾値よりも上がる)LEDは消灯

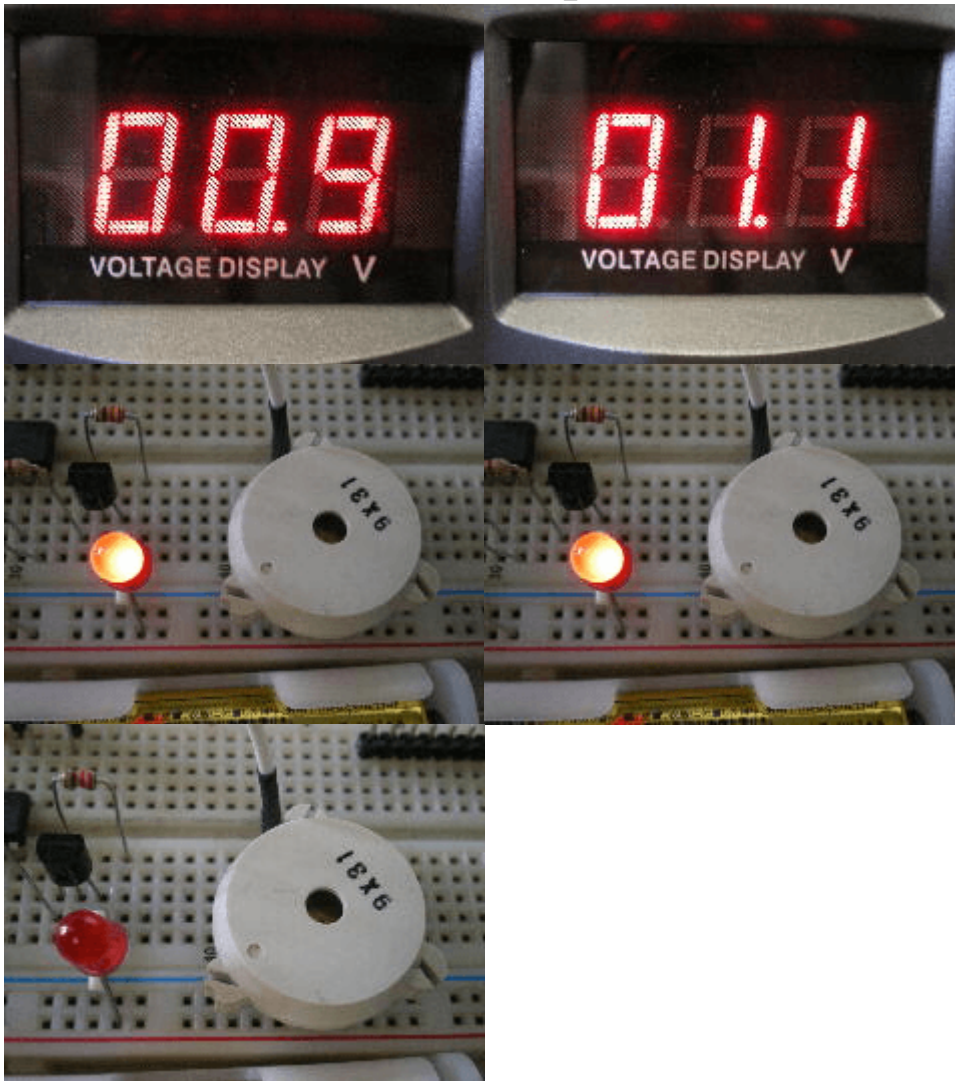




左側上下:閾値を1.0Vに設定する□LEDは点灯 中央上下:入力電圧が0.9V(閾値よりも下がる)□LEDは点灯 右



側上下:入力電圧が1.1V(閾値よりも上がる)□LEDは消灯



如何ですか? これで電池駆動型の製作物も安心して使えますね



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:pic12f683:20>

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

