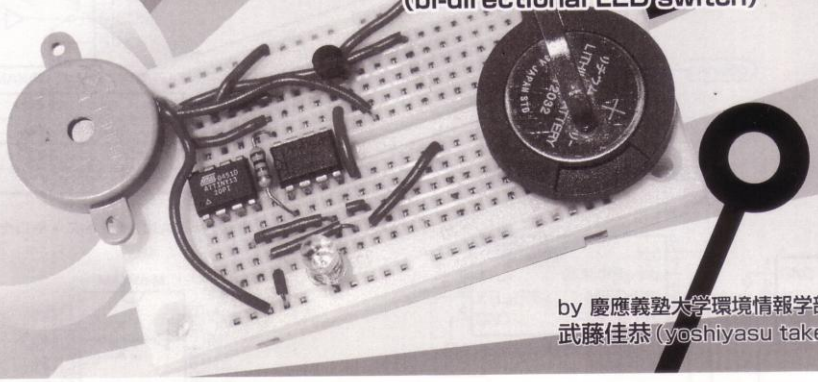
LEDスイッチ

(bi-directional LED switch)



by 慶應義塾大学環境情報学部教授
武藤佳恭(yoshiyasu takefuji)

はじめに

エレキジャックNo.5で紹介した温度差発電に使ったペルチェ素子は，通常，直流電圧を加えて放熱したり・吸熱したりします．このペルチェ素子の両面に温度差を起こすと，逆に直流電圧を起電します．また，2008年1月に実験を再開したJR東日本の発電床プロジェクトでは，スピーカに使うピエゾ素子を使っています．ピエゾ素子に電気信号を与えるとスピーカとして音が発生します．逆に，ピエゾ素子に振動を加えると，交流の電圧が発生します．このように多くのデバイスには，面白い現象として，双方向性(bi-directional)の性質があります．

今回は，LEDの双方向性を利用した，LEDスイッチを紹介します．LED(Light Emitting Diode)は，発光ダイオードのことで，最近の信号機，バスや電車の表示板，車のヘッドライト，その他，いろいろな電気機器の表示に広く使われています．LEDは，今では日常生活に必要不可欠な電子部品の一つになっています．LEDは，ペルチェ素子やピエゾ素子と同様，双方向性の高いデバイスです．

LEDが一躍，日本社会一般に広まったのは，中村修二教授と日亜化学との青色LEDの特許紛争です．青色LEDの発明対価をめぐる訴訟で，東京地裁の一審では，中村教授の請求どおり，日亜化学へ200億円を支払うよう命じていましたが，二審の東京高裁では，和解を勧告し，日亜化学が中村教授に8億4391万円を支払い，中村教授はその他の請求を放棄することとなりました．

ここでは，LEDの双方向性を利用して，一つのLEDを点灯させたり，センサとして利用したりすることによって，比較的簡単に，面白いLEDスイッチ・システムができあがります．LEDに近づくとそのLEDが光ったり，メロディが鳴り始めたりします．

LEDスイッチの基礎

LEDにはいろいろな種類がありますが，今回使うLEDは，超高輝度赤色LEDを使います．実際には，秋月電子通商の18cdの超高輝度赤色LED(5mm)広角(60°)タイプ(1個20円)，あるいは，25cdの超高輝度赤色LED(3mm)(ハイパワー赤色LED：1個30円)を使いました．

図1に実際のLEDと電気記号(アノードとカソード)を示します．LEDを点灯させるためには，1.7V以上の電位をアノードとカソードに与え，数

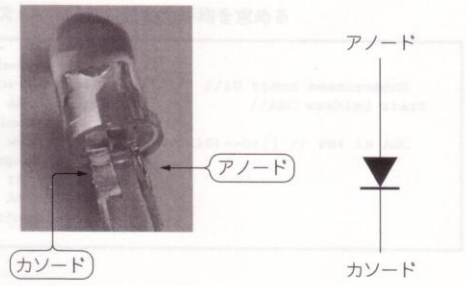


図1　実際のLEDと電気記号（アノードとカソード）

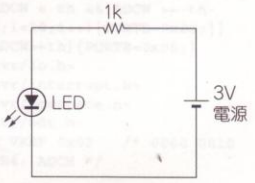


図2　LEDの点灯回路

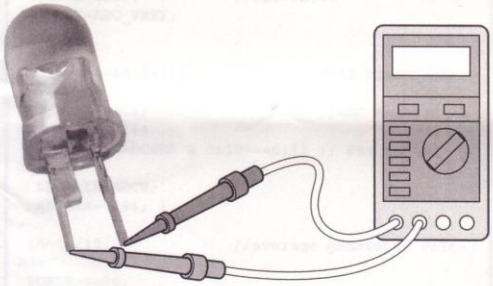


図3　LEDは発電素子

mA以上の電流を流す必要があります．**図2**にLEDの点灯回路を示します．

図3に示すように，LEDの両端をテスタに接続して，電圧を測定してみましょう．驚くことに，LEDの周りの光によって，LEDの両端には直流電圧が生じます．LEDの周りが明るければ，LEDの両端の直流電圧は高くなり，暗くなればLEDの直流電圧は低くなります．

LEDの二つの双方向性の特徴（電圧を加えると点灯，周りの光で発電）をコントロールするために，アナログ・スイッチをここでは使います．

アナログ・スイッチとは

普通のスイッチは手でON/OFFさせ2点間の接続を制御しますが，アナログ・スイッチでは，電気信号（0Vか3V）でアナログ・スイッチをON/OFFさせます．アナログ・スイッチでは，手の換わりに，スイッチON/OFF用の制御信号が必要です．

ここで紹介するアナログ・スイッチはONのとき，両端は接続され，抵抗値が33Ω程度あります．今回用いるMAX4544（SPDT）アナログ・スイッチのスイッチングは，35nsのスピードでスイッチをON/OFFできます．**図7**の"IN"がスイッチの制御信号です．

ここで，**図8**に，今回紹介するLEDスイッチの原理図を示します．**図8**では，SPDTのアナログ・スイッチを使っています．**図8**の"IN"はアナログ・スイッチのコントロール信号で，LEDを点灯モードにするか，LEDの起電圧を"Analog"信号に出力するかをコントロールできます．

図8の原理図を基に，SPDTアナログ・スイッチの"IN"信号をコントロールしながら，アナログ値を読み取ったり，LEDを点灯させたりする**図9**のLEDスイッチ回路を設計しました．

全体の回路を小さくするために，8ピンDIPのAVRチップTINY13を使いました．オプティマイズのパーツ・ショップで，1個約200円で入手

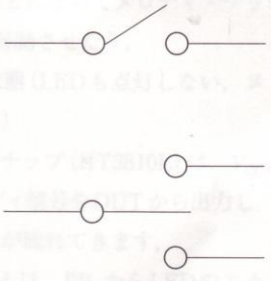
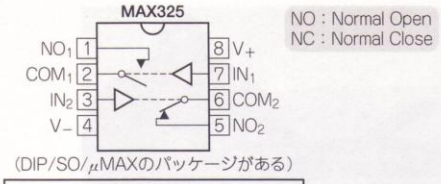


図4
SPST (single pole single throw) **スイッチ**

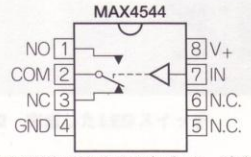
図5
SPDT (single pole double throw) **スイッチ**



(DIP/SO/μMAXのパッケージがある)

MAX325		
LOGIC	SWITCH1	SWITCH2
0	OFF	ON
1	ON	OFF

図6 SPSTアナログ・スイッチ(MAXIM社)



(μMAX/SO/DIP/QFNの各パッケージがある)

MAX4544		
LOGIC	NC	NO
0	ON	OFF
1	OFF	ON

SWITCHES SHOWN FOR"0"INPUT

図7 SPDTアナログ・スイッチ
(MAXIM社)

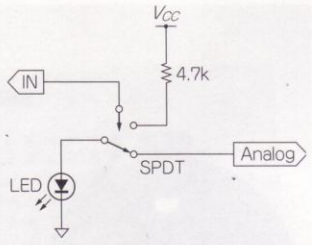


図8 LEDスイッチの原理図

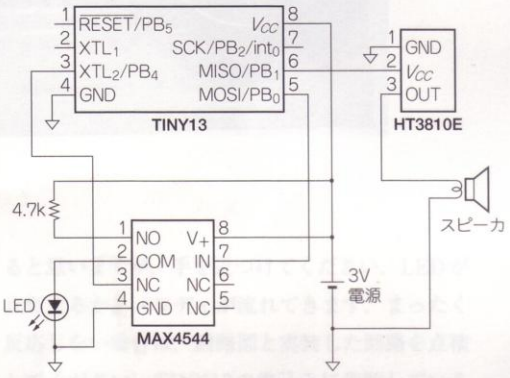


図9 LEDスイッチ回路図

しました.

http://optimize.ath.cx/shopv2_2/chumon.html

ここで紹介する回路に必要な部品は，TINY13チップのほかに，MAX4544のSPDTアナログ・スイッチ，メロディ・チップ，スピーカ，超高輝度LED，3Vのボタン電池などです．すべての部品を，秋月電子通商の150円ブレッドボードに挿します．

MAX4544は，次のサイトからPDIPのサンプルを注文するか，購入してください．

http://japan.maxim-ic.com/quick_view2.cfm?qv_pk=1697&t=or

プログラミングには，GCC(GNU Cコンパイラ)を使います．TINY13はGCCがサポートしているチップです．GCCはオープン・ソースで広く使われている，最強の開発ツールです．GCCは，Windows OS，Mac OS，Linux OSでも使える便利なCコンパイラです．今回は使いませんが，GCCを使うといろいろな複雑なプロトコル・スタックが簡単に利用できます．Linux OSも，GCCで構築されています．

AVR-GCC(AVR専用 GCC Cコンパイラ)をWindows OSにインストールするには，最新のオープン・ソースのWinAVRツールを次のサイトからダウンロードし，ダウンロードしたファイルをダブルクリックすれば，自動的にインストールできます．

http://winavr.sourceforge.net/download.html

リスト1　LEDの起動力平均を求める

```
th=0;
for(i=0;i<10;i++){      //10 times measurement
  ADCSRA=0xC6;              //ADC enabled start
clock/64
  while (( ADCSRA & 0x10)==0){} // PB4 is ADC
input
  th = th+ADCW;
  ADCSRA=0x96; }
 th=th/10;
```

リスト2　TINY13プログラム

```
#include <avr/io.h>
#include <avr/interrupt.h>
#include <avr/pgmspace.h>
#include <avr/wdt.h>
#define ADC_VREF 0x02    /* 0000 0010
vcc:10bit:PB4: ADCH */

int main(void)
{ unsigned char i;
  int th;
  DDRB=0x03;            //PB1 and PB0 are output
  PORTB=0x00;           //PB1=PB0=0
  ADMUX=ADC_VREF;

 th=0;
for(i=0;i<10;i++){              //10 times mea-
surement
  ADCSRA=0xC6;                  //ADC enabled
start clock/64
  while (( ADCSRA & 0x10)==0){} // PB4 is ADC
input
  th = th+ADCW;
  ADCSRA=0x96; }

 th=th/10;              //average generated volt-
age
 PORTB=0x00;

for(;;){
  PORTB=0x02;
  ADCSRA=0xC6;              //ADC enabled start
clock/64
  while (( ADCSRA & 0x10)==0){}
  if(ADCW < th-16
){for(i=0;i<20;i++){PORTB=0x01;}}
  else if(ADCW < th && ADCW >= th-
16){for(i=0;i<20;i++){PORTB=0x02;}}
  else if(ADCW>=th){PORTB=0x00;}
   }
}
```

リスト3　Makefile

```
TARGET = attiny13
COMPILE = avr-gcc -Wall -O2  -I. -mmcu=$(TARGET)
# -DDEBUG_LEVEL=2
OBJECTS =  main.o

all:    main.hex

.c.o:
    $(COMPILE) -c $< -o $@

.S.o:
    $(COMPILE) -x assembler-with-cpp -c $< -o $@

.c.s:
    $(COMPILE) -S $< -o $@

clean:
    rm -f main.hex main.lst main.obj main.cof
main.list main.map main.eep.hex main.bin *.o
main.s

main.bin:    $(OBJECTS)
    $(COMPILE) -o main.bin $(OBJECTS) -Wl,-
Map,main.map

main.hex:    main.bin
    rm -f main.hex main.eep.hex
    avr-objcopy -j .text -j .data -O ihex
main.bin main.hex

cpp:
    $(COMPILE) -E main.c

avrdude:
    avrdude -c usbasp -p $(TARGET) -U
flash:w:main.hex
```

TINY13のプログラムの概要

LEDをまず，センサとして用いて周りの明るさを測定します．ここでは，10回ほど測定した平均値を計算します．その値を変数“th”に代入し，刻々と変化するLEDのセンサ値と比較し，LEDを点灯させたり/消灯させたり，メロディを奏でたりさせます．

具体的には，**図9**のPB0を‘0’にすると，アナログ・スイッチはLEDの起電圧をPB_4に接続し，PB_4ポートからアナログ電圧を計測できます．PB_4ポートで読み込まれたアナログ値によって，

(1) PB0=‘1’にすることによってLEDを点灯させたり，

(2) PB_1=‘1’にすることによってメロディ・チップ(HT3810E)を起動させたり，

(3) 何も反応しない状態(LEDも点灯しない，メロディも奏でない)

にします．メロディ・チップ(HT3810E)は，V_{CC}に3Vを与えるとメロディ信号をOUTから出力し，スピーカからメロディが流れてきます．

リスト1のプログラムは，PB_4からLEDのアナ

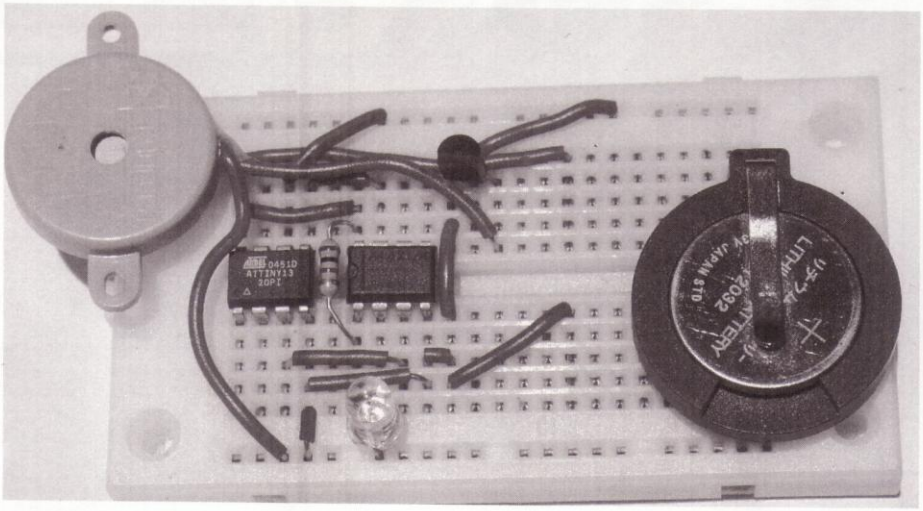


図12　完成したLEDスイッチ

ログ起電圧を10回測定し，その平均値（“th”）を計算します．

次のリスト2のプログラムでは，平均値と刻々と変化するLEDの起電圧値と比較することによって，先ほどの三つの状態の一つを起動させます．ADCWの変数には，LEDの起電圧値が代入されています．

main.hexファイルを生成するには，WinAVRのProgrammers Notepadを起動し，main.c（リスト2）とMakefile（リスト3）を読み込み，Toolsメニューのmakeを起動します．

main.hexファイルを，TINY13のフラッシュ・メモリにプログラム・ライタを使って，書き込んでください．筆者は，自作のUSBライタ，usbasp（800円程で完成できる）を作り，WinAVRの“avrdude”の書込みプログラムでAVRチップに書き込んでいます．

完成したLEDスイッチ回路のテスト方法

ボタン電池を回路に接続すると，LEDが一瞬光ると思いますが，手を近づけてください．LEDが点灯するか，メロディが流れてきます．まったく反応しない場合は，回路図と実装した回路を点検してください．TINY13の書込みに失敗している場合は，プログラム・ライタ書き込み時に，ユーザに知らせてくれます．

リスト2のTINY13プログラム中，次の行の“16”を別の値に変えると3状態の閾値が変わり，光への反応を変えることが簡単にできます．

```
if(ADCW < th-16 ){for(i=0;i<20;i++){PORTB=0x01;}}
  else if(ADCW < th && ADCW >= th-
16){for(i=0;i<20;i++){PORTB=0x02;}}
```

今回紹介した回路とプログラムをベースに，面白いガジェットを作ってください．