フィジカルコンピューティングによる
情報工学学習への動機づけ

実習資料

2010 年 8 月 17 日

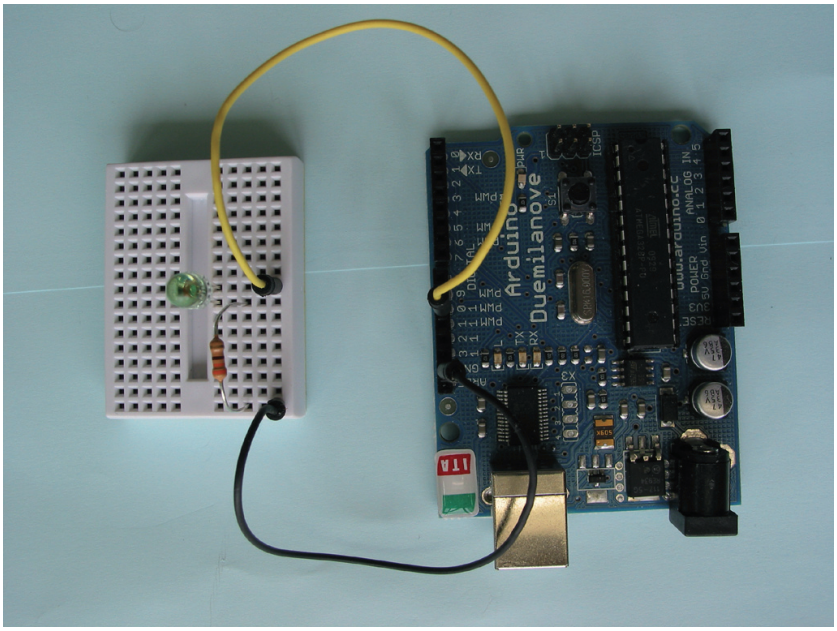
竹内　章

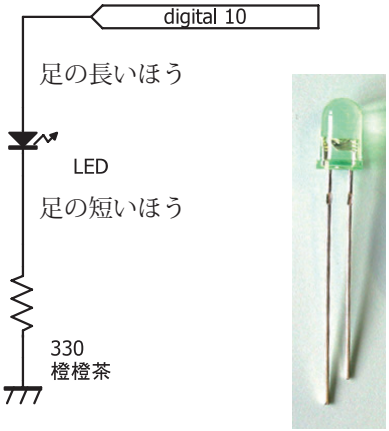
LED 出力

機能

　LED をオン、オフする。

配線





　LED と抵抗を逆順にして、抵抗を digital ピンにつなぎ、LED を抵抗とグラウンドの間に入れてもよい。

プログラム（ファイル名：led）

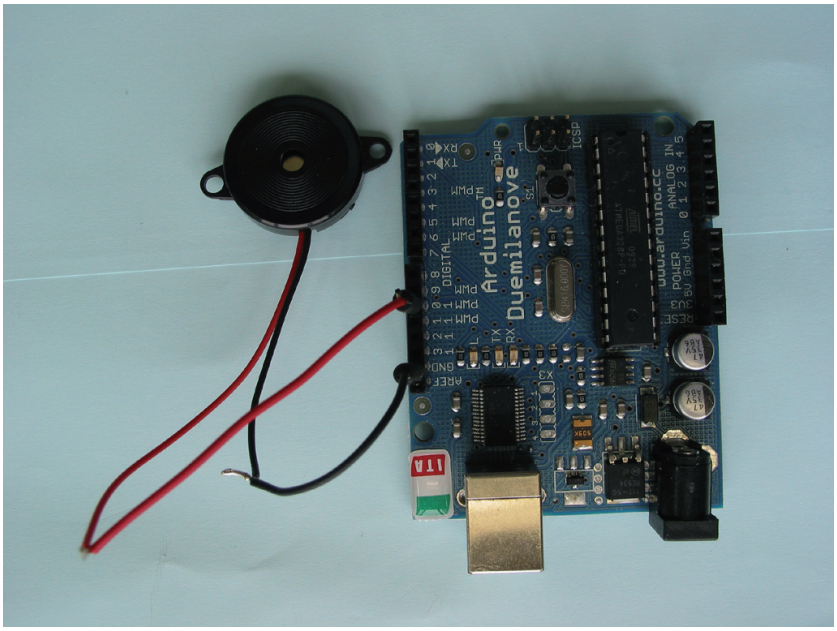
```
int OutPin = 10;                  // LED を接続するデジタルピンの番号。

void setup(){                     // 初期設定
  pinMode(OutPin, OUTPUT);        // OutPin を出力に設定
}
void loop(){                      // 以下のことを繰り返す。
  digitalWrite(OutPin,HIGH);      // スイッチをオンにする。
  delay(1000);                    // 一定時間待つ。単位は ms。
  digitalWrite(OutPin,LOW);       // スイッチをオフにする。
  delay(1000);                    // 一定時間待つ。単位は ms。
}
```

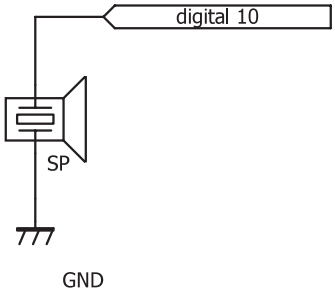
スピーカー出力

機能

　スピーカーを鳴らす。

配線





プログラム（ファイル名：speaker）

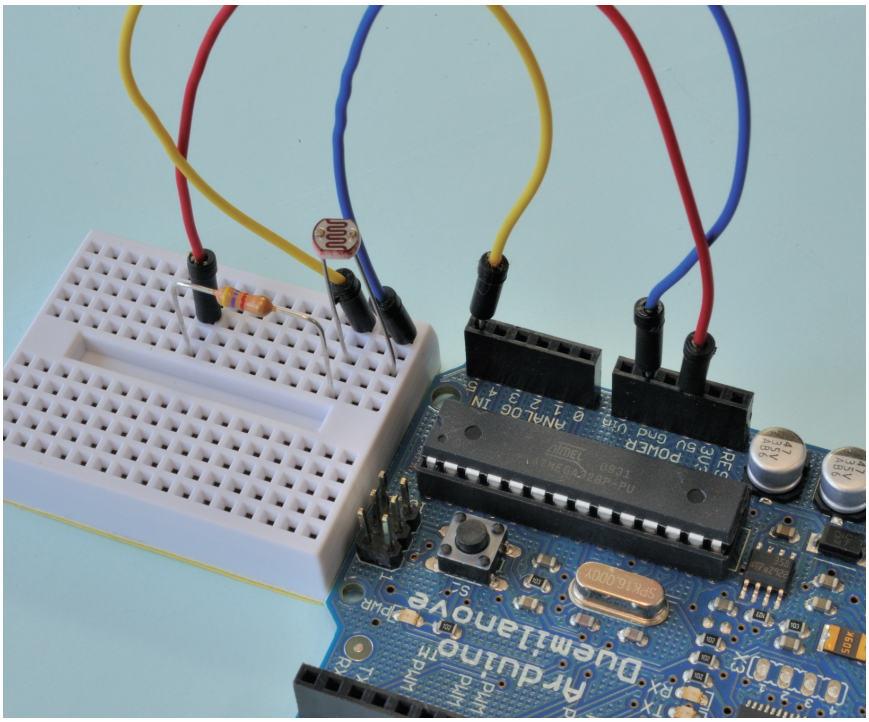
```
int OutPin = 10;                  // スピーカーを接続するデジタルピンの番号。

void setup()                      // 初期設定
{
  pinMode(OutPin, OUTPUT);        // OutPin を出力に設定
}
void loop()                       // 以下のことを繰り返す。
{
  digitalWrite(OutPin,HIGH);      // スイッチをオンにする。
  delayMicroseconds(1000);        // 一定時間待つ。単位はマイクロ秒。
  digitalWrite(OutPin,LOW);       // スイッチをオフにする。
  delayMicroseconds(1000);        // 一定時間待つ。単位はマイクロ秒。
}
```

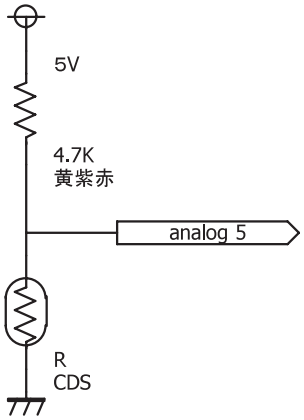
cds で明るさの入力

機能

cds は明るさによって抵抗値が変化するので、それを電圧の変化として入力する。

配線





アナログポートへの電圧入力は０から１０２３までの整数値として読み取られる。標準では、０V の時に０、５V のときに 1023 になる。cds の抵抗値は、明るさによって 1K Ω から 5K Ω程度の範囲で変化するので、上記の配線では 180 から 530 程度の値が読み取れる。

プログラム（部分）

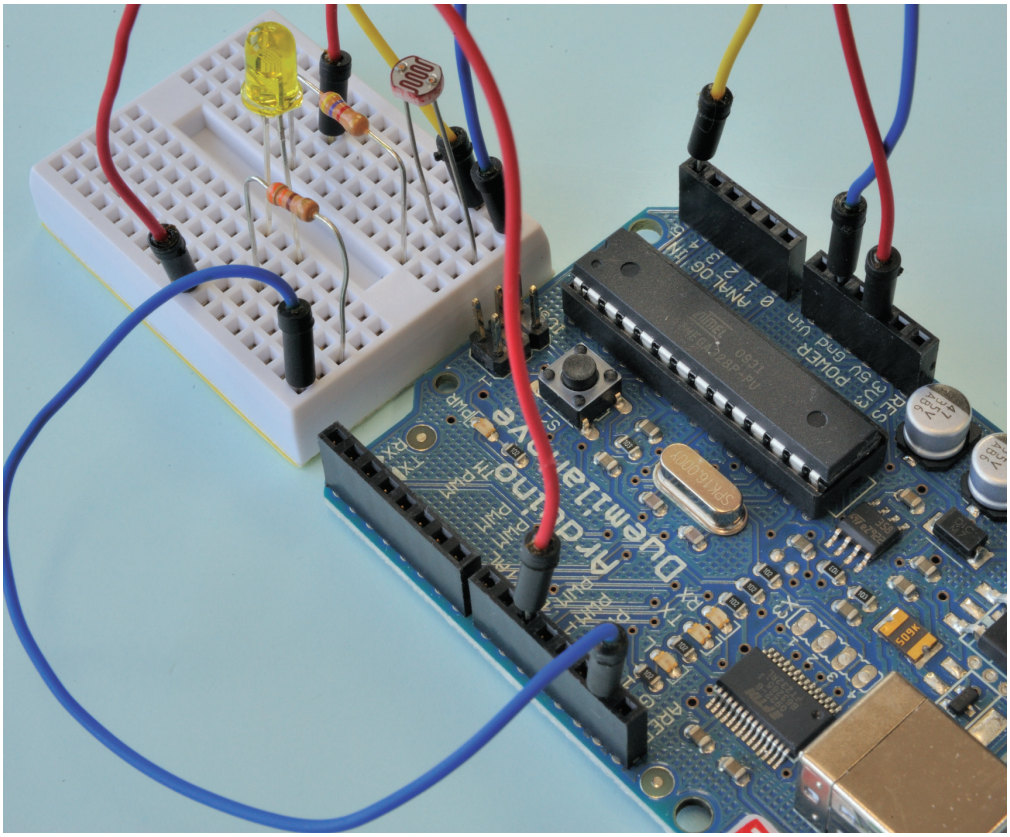
```
int InPin = 5;                  // センサーを接続するアナログ入力ピンの番号。
                                // アナログピンの０から５までのどれを使ってもよく、配線と合わせる。
void setup(){                   // 初期設定
}
void loop(){                    // 以下を繰り返せ。
  int val;                      // センサーからの入力値を格納する整数型の変数宣言。

  val = analogRead(InPin);      // アナログ入力ピン５番の電圧を読み取る。
        ・・・
```

CDS 入力と LED 出力（P1 と P3 の組み合わせ）

機能

　明るさ応じてスイッチをオン・オフする時間間隔を変化させて、LED の点滅間隔を変える。

配線



プログラム（ファイル名：cds_led）

```
int OutPin = 10;                  // led を接続するデジタルピンの番号
int InPin = 5;                    // センサーを接続するアナログ入力ピンの番号

void setup( ){                    // 初期設定
  pinMode(OutPin, OUTPUT);        // OutPin を出力に設定。
}
void loop( ){                     // 以下を繰り返せ。
  int val;                        // センサーからの入力値を格納する整数型の変数宣言。

  val = analogRead(InPin);
  digitalWrite(OutPin,HIGH);      // スイッチをオン
  delay(val * 2);                 // 値に応じて待つ。単位はミリ秒
  digitalWrite(OutPin,LOW);       // スイッチをオフ
  delay(val * 2);                 // 値に応じて待つ。
}
```

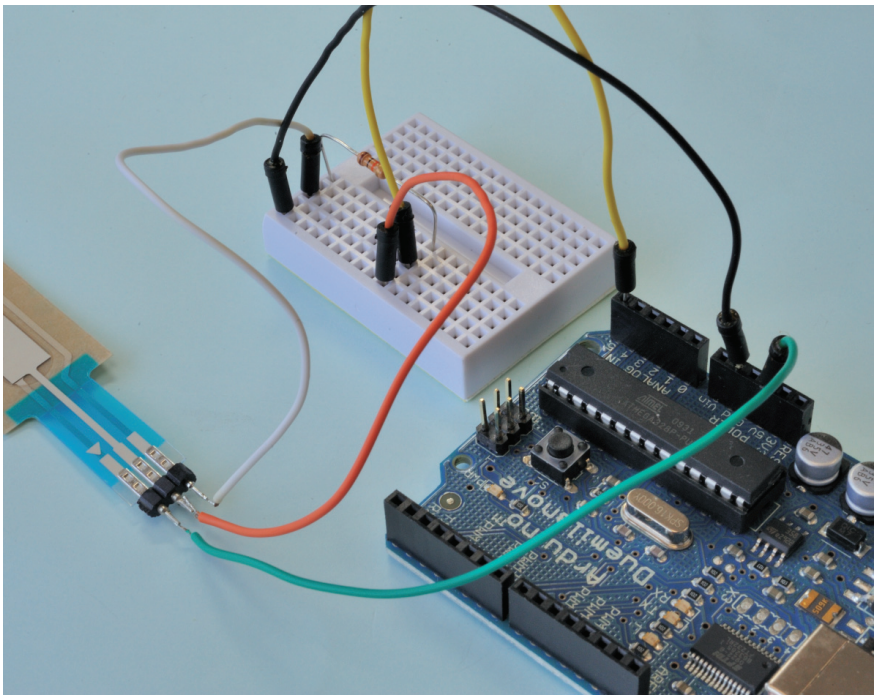
プッシュセンサー入力

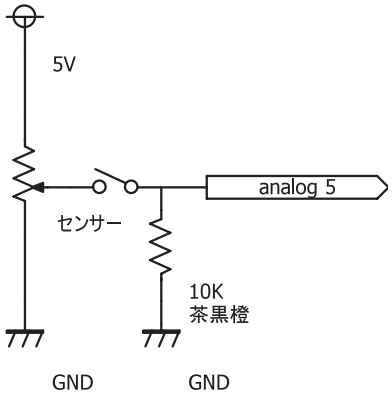
機能

センサーが押されているかと、押されている場所を入力する。

押されている場所によって抵抗値が変化するので、それを電圧の変化として入力する。

配線





アナログポートへの入力は０から１０２３までの整数値として読み取られる。標準では、０Ｖの時に０、５Ｖのときに 1023 になる。上記の配線では、センサーを押していないときの電圧は０Ｖで値は０、センサーを押しているときには電圧が０～５Ｖになるので、０から１０２３の値が読み取れる。

プログラム（部分）

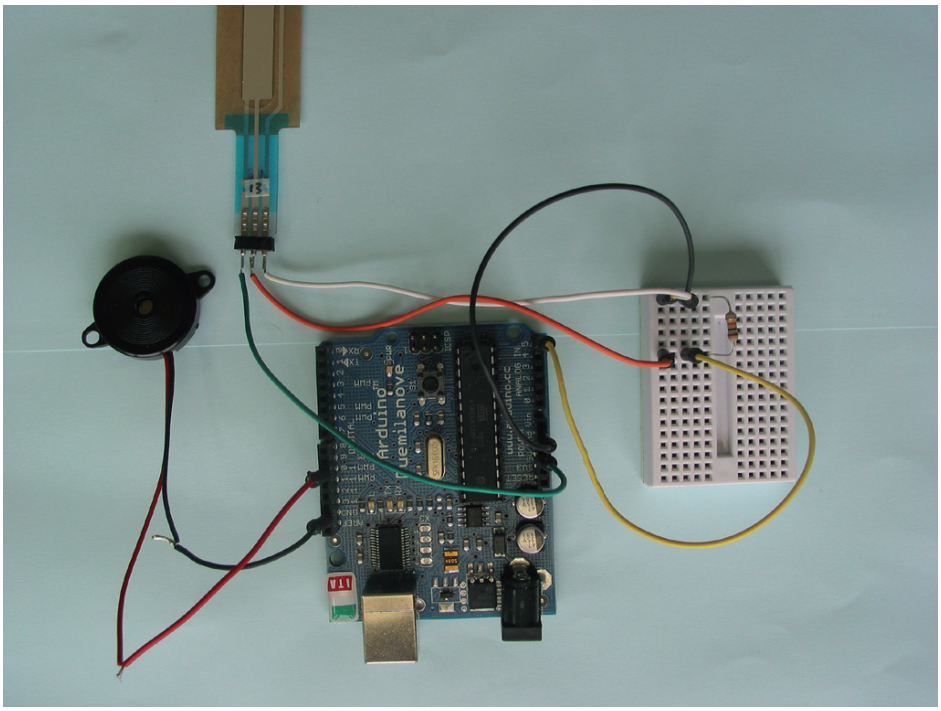
```
int InPin = 5;                  // センサーを接続するアナログ入力ピンの番号。
                                // アナログピンの０から５までのどれを使ってもよく、配線と合わせる。

void setup(){                   // 初期設定
}
void loop(){                    // 以下を繰り返せ。
  int val;                      // センサーからの入力値を格納する整数型の変数宣言。

  val = analogRead(InPin);      // アナログ入力ピン５番の電圧を読み取る。
        ・・・
```

プッシュセンサー入力とスピーカー出力（P2, P5 の組み合わせ）

機能

センサーが押されている場所に応じてスイッチをオン・オフする時間間隔を変化させて、スピーカーの音を変える。

配線



プログラム（ファイル名：push_speaker）

```
int OutPin = 10;                  // スピーカーを接続するデジタルピンの番号
int InPin = 5;                    // センサーを接続するアナログ入力ピンの番号

void setup( ){                    // 初期設定
  pinMode(OutPin, OUTPUT);        // OutPin を出力に設定。
}
void loop( ){                     // 以下を繰り返せ。
  int val;                        // センサーからの入力値を格納する整数型の変数宣言。

  val = analogRead(InPin);
  if (val > 3){                   // ノイズを考慮し、値が３以上であれば、押されていると判断する。
    digitalWrite(OutPin,HIGH);    // スイッチをオン
    delayMicroseconds(100+val);   // 値に応じて待つ。analogRead にかかる時間を考慮して 100 加算。
    digitalWrite(OutPin,LOW);     // スイッチをオフ
    delayMicroseconds(val);       // 値に応じて待つ。単位はマイクロ秒。
  }
}
```

温度センサー入力

機能

温度に応じて出力電圧が変化するので、電圧を読み取る。ここで使用する LM60 の出力電圧は次のとおりである。

出力電圧 = (+6.25mV/℃ × T ℃) + 424mV

したがって、0℃で 424mV、100℃で 1049mV となる。

配線

+3.3V　出力　アース (Gnd)

LM60

下から見たピン配置。左から、+3.3V、出力、アース (Gnd)。

出力をアナログ入力ピンにつなぐ。

アナログポートへの入力は０から１０２３までの整数値として読み取られる。この例では、０V の時に０、1.1V のときに 1023 になるように設定している。

プログラム（部分）

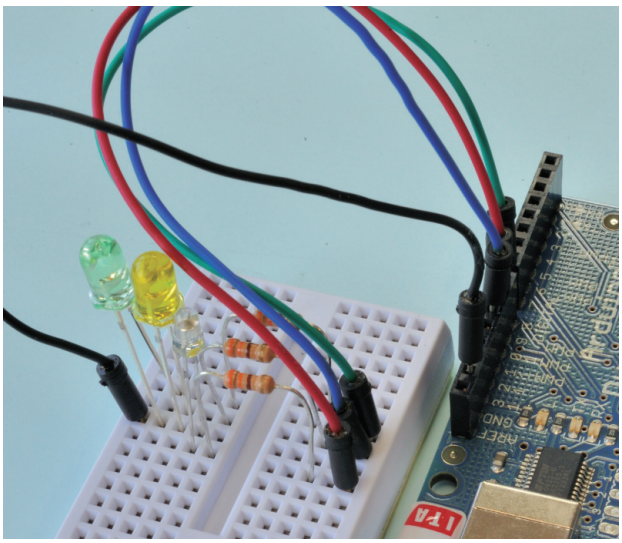
```
int InPin = 5;                  // センサーを接続するアナログ入力ピンの番号。
                                // アナログピンの０から５までのどれを使ってもよく、配線と合わせる。

void setup( ){                  // 初期設定
  analogReference(INTERNAL);    // アナログ入力の最大値を 1.1V に設定する。
}
void loop( ){                   // 以下を繰り返せ。
  int val;                      // センサーからの入力値を格納する整数型の変数宣言。

  val = analogRead(InPin);      // アナログ入力ピン５番の電圧を読み取る。
        ・・・
```

３つの LED に出力

機能

　ある変数の値に応じて、３つの led の内の一つをつける。

配線



プログラム（部分、ファイル名：three_led）

```
int LED1 = 8;                    // led を接続するデジタルピン番号。
int LED2 = 9;                    // ３つを続きのピン番号にしておくこと。
int LED3 = 10;
int old = 0;                     // オンの led を覚えておくための変数。初期値を 0 にする。

void setup( ){
  pinMode(LED1,OUTPUT);          // デジタルピンを出力に設定。
  pinMode(LED2,OUTPUT);
  pinMode(LED3,OUTPUT);
}

void loop( ){
  int level;                     // 新たにオンにする led を指定するための変数。

// level の値を 1 から 3 に設定する処理をこの部分に入れる。次のページ参照。

  if(old != level){              // 現在オンにしてる led とは違う led をオンにするならば、
    if (old != 0) digitalWrite(LED1 + old -1,LOW);   // old が 0 でないなら、対応するデジタルピンをオフにする。
    digitalWrite(LED1 + level - 1,HIGH);  // オンにする led が繋がれているデジタルピンを計算し、オンに。
    old = level;                 // オンにした led を覚えておく。
  }
  delay(50);                     // ５０ミリ秒待つ。
}
```

入力値に応じてレベル分けするプログラムの部分

機能

ある変数の値に応じて何段階かにレベル分けする。

プログラム例１：１入力３レベル分け

ある変数 val の値の範囲に応じて level を３段階にわける。

応用例：センサーからの入力値に応じて、level の値を 1 から 3 に設定する。

```
int val, level;

if (val < 下限値 ) level =1;
else if ( 下限値 < val && val < 上限値 ) level = 2;
else level = 3;
```

プログラム例２：２入力３レベル分け

ある変数 val1 と val2 の値の大小に応じて level を３段階にわける。

応用例：２つのセンサーからの入力の差によって、level の値を 1 から 3 に設定する。例えば、CDS や温度センサーを２つ使って、２地点の明るさの差や温度の差によって、場合分けをする。

```
int val1, val2, level;

if (val2 - val1 > 閾値 ) level =1;
else if (val 1 - val2 > 閾値 ) level = 3;
else level = 2;
```

プログラム例３：１入力８レベル分け

ある変数 val の値に応じて、下限値から等間隔の境界を設定して、level を０から７の８段階に分ける。

応用例：センサーからの入力値によって、level の値を０から７に設定する。たとえば、気温に応じて３色 LED の色を変えるための場合分けができる。

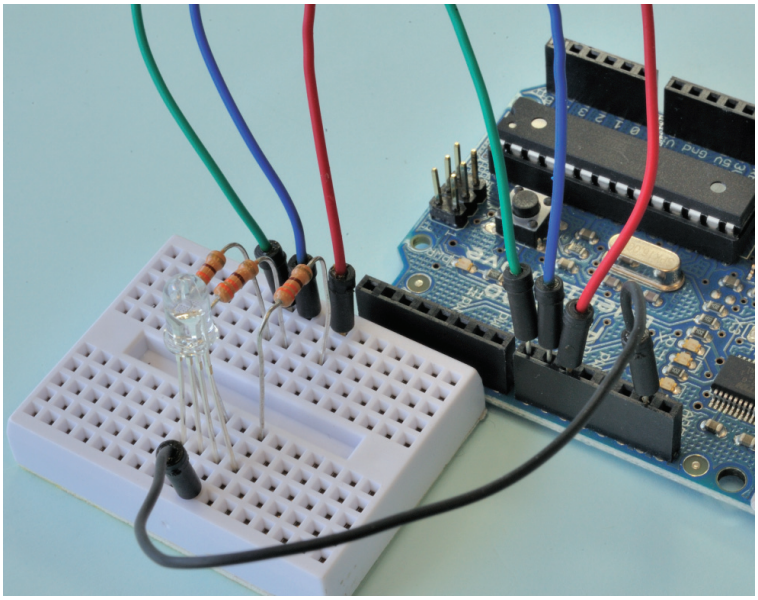
```
int val,  level;

level = 0;                          // level を０に初期化する。
border = 下限値 ;                   //border を下限値に初期化する。
while (val > border && level < 7){  // val が border より大きく、level が７よりも小さい間、以下を繰り返す。
  level++;                          //       level を 1 増やす。
  border += 増分 ;                  //       border を増分だけ増やす。
}                                   // 繰り返しの範囲はここまで。
```

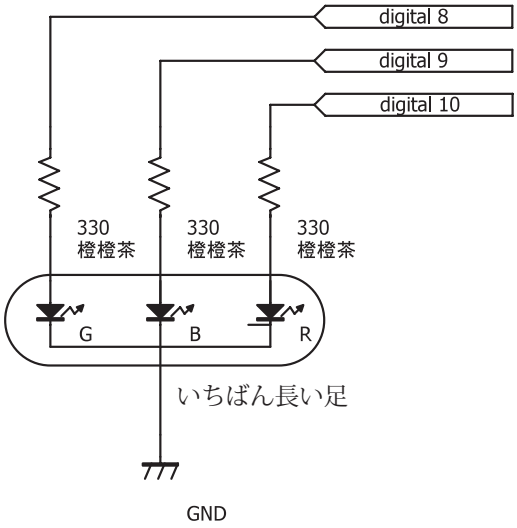
３色 LED の色を組み合わせて７色の出力

機能

３色 LED を利用して、３原色の組み合わせにより、７色に光らせる。消灯を含め、８状態を区別するのに利用できる。

配線





プログラム（ファイル名：led3c）

```
int LED1 = 8;                   // デジタルピン８，９，１０に３色 LED をつなぐ。
int LED2 = 9;
int LED3 = 10;
int level;                      // 色を指定する変数

void setup(){                   // 初期設定
  pinMode(LED1,OUTPUT);         // LED1,2,3 のデジタルピンを出力に設定。
  pinMode(LED2,OUTPUT);
  pinMode(LED3,OUTPUT);
  level = 0;                    // レベルの初期値を０にする。
}

void loop(){                    // 以下を繰り返す。
  digitalWrite(LED1,level & 1); // level の最下位ビットが１なら LED1 をオン、０ならオフにする。
  digitalWrite(LED2,level & 2); // level の下から２ビット目が１なら LED2 をオン、０ならオフにする。
  digitalWrite(LED3,level & 4); // level の下から３ビット目が１なら LED3 をオン、０ならオフにする。
  level++;                      // level の値を１増やす。
  delay(500);                   // ５００ミリ秒待つ。
}
```