

Arduino/ Suwano をはじめよう@KYOTO- U No.6

2012 年 5 月 31 日

京都大学 学術情報メディアセンター

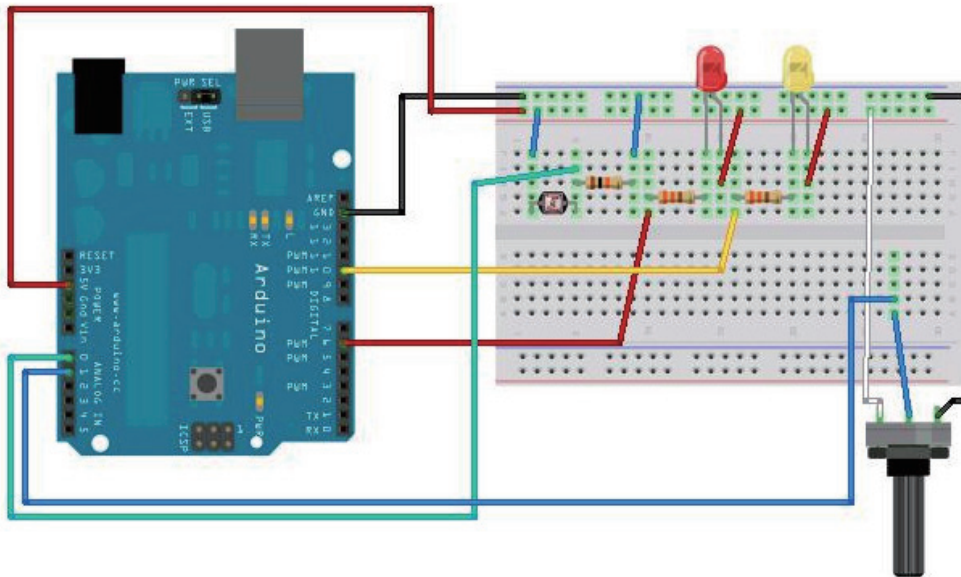
喜多 一

今回の学習目標

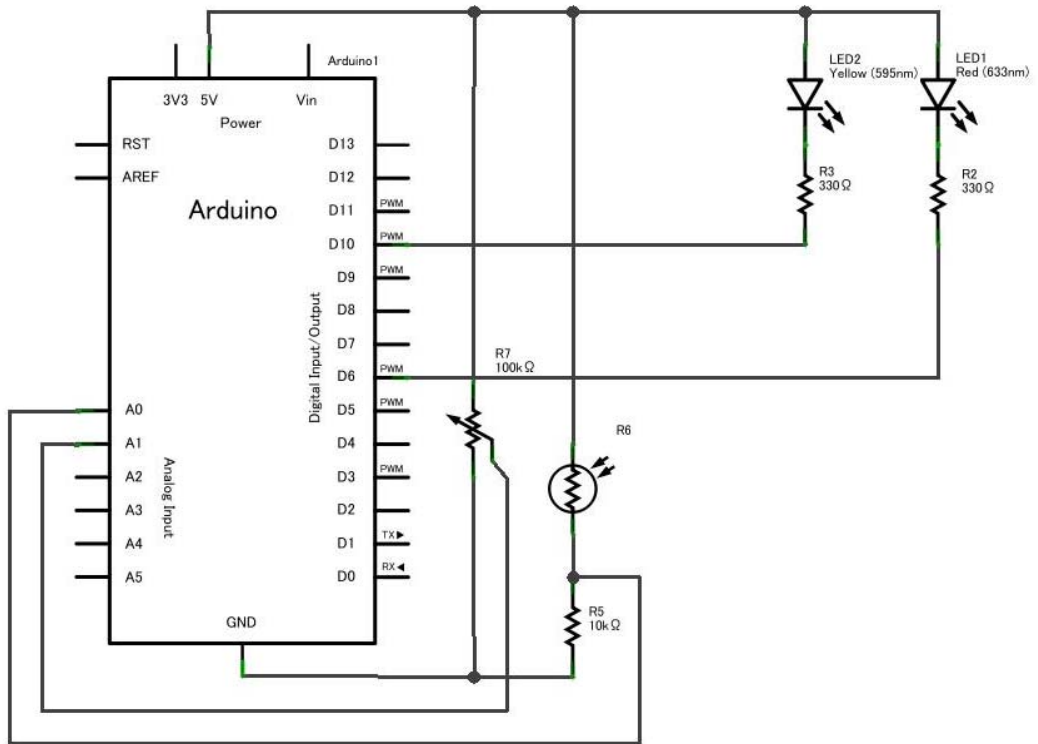
- アナログ出力(PWM)の方法を知る.
- 繰り返し処理の方法を知る.
- 変数とスコープ
- アナログ入力の方法を知る.
- PC へのデータの送出手を行う.
- Arduino Uno のピン配置を知る.

1. 回路の準備.

ブレッドボードを用いて次図のような回路を作成してください. 光センサの CdS (硫化カドミウムのセル) は中に模様のある丸い部品です. 電極に極性はありません. 片側を 5V に、他方には直列に 10kΩ の抵抗をつないで GND に接続します. また可変抵抗器は両側を 5V と GND に、中央を Analog1 につなぎます.



Fritzingを使用して作成
<http://fritzing.org/download/>



2. アナログ出力の方法

次のプログラムを作成し、実行してみてください。

行番号	ソースコード	説明
1	#define LED1 6	6 番ピンを LED1 と、
2	#define LED2 10	10 番ピンを LED2 とします。
3		
4	int brightness[] =	初期値を持った配列の宣言
5	{0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100};	
6		
7	void setup()	
8	{	
9	pinMode(LED1, OUTPUT);	6, 10 番ピンを出力用にしま
10	pinMode(LED2, OUTPUT);	す。
11	}	
12		
13	void loop() {	
14	for (int i = 0; i<11; i++) {	For 文で変数 i を 0 から
15	int br=brightness[i]*255/100;	10 まで1つずつ増やしなが
16	analogWrite(LED1, br);	ら実行します。
17	analogWrite(LED2, 255-br);	analogWrite は 0 から 255
18	delay(50);	までの段階で PWM 出力を行
19	}	います。LED は 5V 側につな
20	for (int i = 10; i>-1; i--) {	がっている (プルアップされ
21	int br=(brightness[i])*255/100;	ている) ので大きな値になる
22	analogWrite(LED1, br);	ほど暗くなります。
23	analogWrite(LED2, 255-br);	今度は i を 10 から降順に
24	delay(200);	0 まで繰り返します。
25	}	
26	}	

LED1, 2 の明るさは直線的に変化しているはずですが、人間には分かりにくいかもしれません。明るさの変化を指数関数的に変える(例えば 1,2,4,8,16...)とどうなるか試してみてください。

analogWrite() 関数は短い時間のオンオフの繰り返しにより平均値を連続的に変化させる方法(PWM)で出力する関数です。使えるポートは 3, 5, 6, 9, 10, 11 (～が付いている)です。

3. 繰り返し処理のプログラミング

2. 節のプログラムで新しく出てきたプログラムの概念に繰り返し処理があります。

3.1 for 文

動作を繰り返すためによく使う構文が for 文です。For は「～の間」という意味にとればいいでしょう。以下のような表記をします。

```
for (初期化; 継続条件; 繰り返し時の操作) {  
    繰り返す内容  
}
```

for 文は以下のように訳読するとよいでしょう。

「初期化」から初め、「継続条件」が成り立っている間、「繰り返し時の操作」をしながら「繰り返す内容」を繰り返す。

例えば

```
for (int i = 0; i < 11; i++) {  
    ...  
}
```

は

```
for (int i=0;          i<11;          i++){ ... }
```

↓ ↓ ↓
変数 i を 0 から、i が 11 未満の間、i を 1 ずつ増やしながら…を繰り返す

と読めばいいでしょう。ここで「++」は変数の値を 1 だけ増やす演算です。不思議な記号ですがプログラムでは変数を 1 だけ増やすことが多いので、こういう表記が導入されています。1 減らす演算は「--」です。

3.2 変数のスコープ

2 節のプログラムでは配列 brightness は関数 setup() や loop() の外側で宣言されており、「外部変数」とか「グローバル変数」と呼ばれます。外部変数は

- ・ プログラムの実行中は永続的に存在し、値を保持するとともに、
- ・ どの関数からでも読み書きできます。

一方、loop() 関数の中の br や for 文の中の i などは**自動変数**とか**ローカル変数**と呼ばれ、

- ・ 当該の関数が呼ばれている間あるいは for 文が繰り返し動作をしている間だけ存在します。
- ・ また、他の関数からは読み書きできません。

4. 可変抵抗器、CdS セルからの入力

CdS セルは明るさに応答して電気抵抗が変化する（明るいと下がる）素子です。図 1 の回路では明るいとき analog 0 ピンの電圧が高く、暗いと低くなります。CdS セルで明るさを検出してスイッチ代わりに使いましょう。次のプログラムを実行してみてください。

ここでは PC 側とプログラムの実行時にもデータのやりとり（シリアル通信）をします。プログラムのアップロードが終わったら tool → serial monitor を選びます。Arduino 側で検出した A0 番ピンの信号レベル（0 ～ 1023）の値が表示されるはずです。

行番号	ソースコード	説明
1	#define LED1 6	
2	#define LED2 10	
3		
4	void setup()	
5	{	
6	Serial.begin(9600);	PC と通信するための初期設定
7	pinMode(LED1, OUTPUT);	
8	pinMode(LED2, OUTPUT);	
9	digitalWrite(LED1, HIGH);	最初は消灯しておきます
10	digitalWrite(LED2, HIGH);	
11	}	
12		
13	void loop() {	
14	int val1 = analogRead(0);	A0, A1 ピンの電圧を読み込みます。
15	int val2 = analogRead(1);	
16	Serial.print(val1, DEC);	読み込んだ値を PC に 10 進数表記で送ります。
17	Serial.print(" ");	
18	Serial.println(val2, DEC);	読み込んだ値 val1 の大小で LED を点灯、消灯します。閾値の 600 は適宜、調整してください。
19	if (val1>600) {	
20	digitalWrite(LED1, LOW);	
21	} else {	
22	digitalWrite(LED1, HIGH);	
23	}	読み込んだ値 val2 の大小で LED を点灯、消灯します。閾値の 600 は適宜、調整してください。
24	if (val2>600) {	
25	digitalWrite(LED2, LOW);	
26	} else {	
27	digitalWrite(LED2, HIGH);	
28	}	
29	delay(50);	
	}	

4.1 アナログ入力

A0 ~ A5 に入力された電圧(0から5Vの範囲であること)を読み取って 0 ~ 1023 の値で返す関数が `analogRead(pin)` です. ピン番号は 0 から 5 です.

4.2 PC との通信

Arduino から USB ケーブルで接続された PC にデータを送ることができます. Arduino 内部での計算が正しく行われているかどうかはそのままでは確認すべきすべがありません. PC 側に変数の値などを書き出して確認する方法を知っておくことは重要です.

- `Serial.begin(9600);` シリアル通信を開始するための設定です. ()内の数字は通信速度で 9600 は1秒間に 9600 bit 送る(大体 1000 文字程度)というものです.
- `Serial.print(val1, DEC);` `val1` という変数の値を 10 進数(decimal number)表記で書き出します.
- `Serial.print(" ");` " " (2重引用符) で囲まれた文字列を書き出します.
- `Serial.println(val2, DEC);` `val2` という変数の値を 10 進数(decimal number)表記で書き出し、行末で改行します.
- PC 側では Arduino ソフトウェアの Tools → Serial Monitor というメニューを選べば送られてくる内容を確認できます.

5. map() 関数を使う

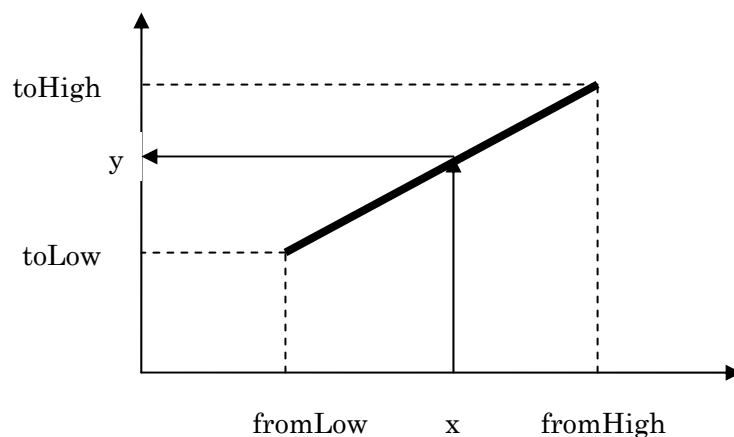
アナログの変数の変域を調整するのに便利な関数が

```
map(value, fromLow, fromHigh, toLow, toHigh)
```

です. 例えば 0 から 1023 までの値をとる変数 `x` (`analogRead()` で得られるものはこの範囲です)の値を 0 から 255 に変換する (`analogWrite()` で許される範囲) に変換するには

```
y = map(x, 0, 1023, 0, 255);
```

とします. 数値の桁あふれをなるべく起さないように long 型の演算が行われます. `fromLow` と `fromHigh`, `toLow` と `toHigh` それぞれの組み合わせで必ずしも Low が High より小さくなければならないという制限はなく範囲の上下を逆転させる使い方もできますが, `fromLow` と `fromHigh` の値が同じだと 0 による割り算でエラーを生じます.



6. Arduino のピン配置

Arduino Uno のピン配置は別紙のようになっています。デジタル入出力は 0 ～ 13 の 14 本ありますが、このうち 0 と 1 は USB での通信用に使われており、13 は予めボード上で LED が配線されています。これらのピンを使うときは要注意です。またこれらのうち 3, 5, 6, 9, 10, 11 は `analogWrite()` 関数での PWM 出力が可能です。