

## arduino (2)

ちゃんと理解しよう



1

## 今後の予定について

| 月  | 日  |    | 月  | 日  |         |
|----|----|----|----|----|---------|
| 10 | 6  | 台風 | 12 | 8  | A1      |
|    | 13 | 台風 |    | 15 | 病欠      |
|    | 20 | 概要 |    | 22 | A2      |
|    | 27 | P1 | 1  | 5  | A3      |
| 11 | 10 | P2 |    | 19 | 制作 1    |
|    | 17 | P3 |    | 26 | 制作 2    |
|    | 24 | P4 | 2  | 2  | 制作 3    |
| 12 | 1  | P5 |    | 9  | ワークショップ |

休みに制作物の構想を練る  
M2 は応援する

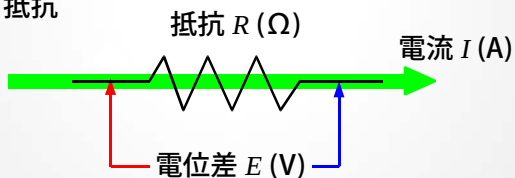
各自、制作物の構想を説明

2

## オームの法則

$$E = I R$$

- $E$ : 電位差、電圧
- $I$ : 電流
- $R$ : 抵抗



3

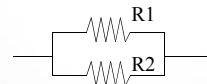
## 応用：合成抵抗

- 抵抗の直列接続



$$\text{合成抵抗 } R = R1 + R2$$

- 抵抗の並列接続



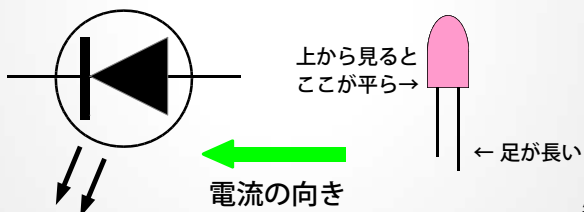
$$\text{合成抵抗 } R = 1 / (1/R1 + 1/R2)$$

4

## LED (発光ダイオード)

電流の流れる向きと、流れない向きがある  
(「極性がある」という)

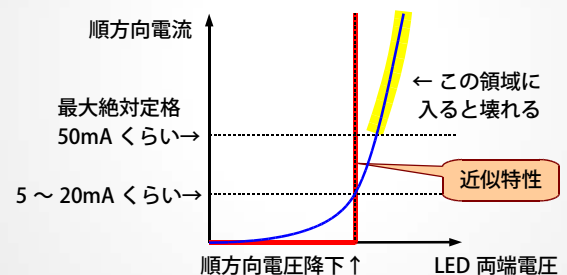
ダイオードといいつつ逆電圧に弱いので注意。  
5V くらいで壊れる。整流には使わない。



5

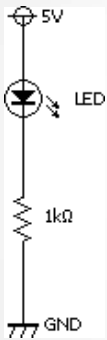
## LED の性質

順方向に電圧をかけたとき、ある電圧を超えると急に電流が流れる。その後は急激に電流量が増える。(「順方向電圧降下」という。白 3.6V、青 2.9V、赤黄緑 2.0V くらい)



6

## 前回の初めての回路



- LED は点灯しているとする。
- LED では 2V の電圧降下が起きる。
- 抵抗の両端の電位差は  $5-2=3V$ 。
- $1k\Omega$  なので R に流れる電流は 3mA。
- 同じだけ LED にも流れる。
- これを「電流制限抵抗」という。

7

## イタリア人め…

テキスト p.25 で、デジタル 13 番で LED を点灯させるとき、抵抗なしに直結している。

最近の部品は頑丈になったのでとりあえずは壊れずに動くが、いつ壊れても文句は言えないし、一見動いていても痛む。

「もったいない」の精神と、遊びじゃないんだから理論を踏まえて進めよう。

※ 第2版では [NOTE] が追記されている。

8

## 伏線消化の演習

前回の回路と、Processing で培ったプログラミングセンスを駆使して、次のようなシステムを作ろう。

- 13 番 LED が点灯、9 番 LED が消灯
- ボタンを押すたびに点灯消灯が入れ替わる。

### ヒント

Processing(5) p.9 への Pokémon Shock を思い出そう。

9

## プログラム例 (≒Example03B)

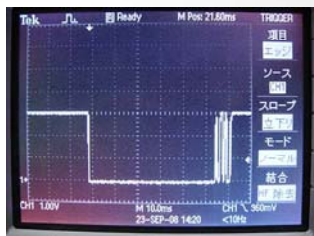
```
int prev = LOW; // 前回のボタン状態
int state = 0; // 0か1のいずれか

void loop() {
  int bs = digitalRead(BUTTON);
  if (prev != bs) {
    prev = bs;
    if (bs == LOW) {
      state = 1 - state; // 0→1, 1→0
    }
  }
  if (state == 1) { ... }
  else { ... }
}
```

10

## チャタリング (bouncing)

- スイッチを1度押したただけなのに、2回動いたりする
- …はずなのだが、最近のスイッチは品質がいいのか、なかなか起きない(※)
- ジャンプワイヤーで直接接触させてスイッチ代わりにしてみる。
- 機械的な接点は、高速な電子回路の視点で見ると、きれいに切り切りしない。



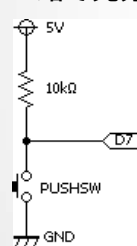
- 時間待ちをしたりして回避する (Example03C)

(※ arduino が遅いだけかも…)

11

## スイッチからの入力

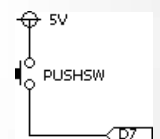
- スイッチの読み取りに、左の回路を組んだ。
- 入力端子の電圧を読み取るだけなら、右の回路で充分では？



- ON では確かに D7 は 5V

- OFF のときは「何も繋がっていない。」

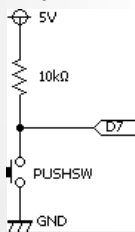
- D7 のワイヤを BB から抜いて、その状況を模擬的に作ると…



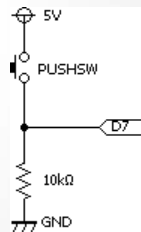
12

## プルアップ / ダウン抵抗

- デジタル入力端子に、常に何らかの電圧をかけるための仕掛け。
- 左はスイッチ OFF 時に高い電圧へ引っ張るので pull-up (OFF で HIGH)



- 右は逆に低い電圧へ引っ張るので pull-down (OFF で LOW)
- 抵抗で消費される電力は基本的に無駄なので、減らすために抵抗値は大きくする



13

## 内蔵プルアップ / ダウン回路

- arduino マイコンは、そもそもデジタル端子を入力にも出力にもできる超高機能
- 端子を入力に設定しつつ出力をかけることで、プルアップ / ダウンの仕掛けをマイコン内部の回路で賄うこともできる！

```
void setup() {  
  pinMode(BUTTON, INPUT);  
  digitalWrite(BUTTON, LOW); // プルダウン  
  //digitalWrite(BUTTON, HIGH); // プルアップ  
}
```

14

## 確かめよう

- 内蔵プルアップ (またはプルダウン) を設定するようにプログラムを改造
- 7 番ピンのジャンプワイヤを引っこ抜いても問題なくなったか確認
- そのワイヤで GND (PD ならば VCC) に触れるとスイッチ ON になることを確認

結論：回路で楽をするとソフトが面倒になる。  
またはその逆。

15

## アナログ (的) 出力

## パルス幅変調

(pulse width modulation)

- 短い時間でデジタル信号の ON/OFF を繰り返すと、平均して中間の値に見える。
- Example01 の delay を小さくする方法では、プログラムは他に何もできない
- UNO の デジ タ ル 端 子 3,5,6,9,10,11 番には自動的に PWM を行う回路がある！



17

## PWM 出力命令

void analogWrite(int, int) は、指定したデジタル端子の PWM 回路で、0 ~ 255 の範囲で指定した比 (分母は 255) の疑似アナログ出力をする。

```
analogWrite( 端子番号, 0 ~ 255 の値 );
```

18

## Example 04

```
const int LED = 9;
void setup() {
  pinMode(LED, OUTPUT);
}
void loop() {
  int i;
  for (i=0; i<255; i++) {
    analogWrite(LED, i);
    delay(10);
  }
  for (i=255; i>0; i--) { ... }
}
```

19

## お題 (1)

Example03 の振る舞いと Example04 の振る舞いを同時に実現する

- 9 番の LED は緩やかな明滅を繰り返す
- 7 番のスイッチを押すたびに、13 番の内蔵 LED が付いたり消えたりする

### ポイント

Example04 そのままだと、9 番が消えた瞬間しかボタンに反応しない機械になってしまう。

20

## お題 (1) 解答例

```
int brt = 0;          // 明るさ 0-255 上昇 256-511 下降
void loop() {
  int bs = digitalRead(BUTTON); //Ex03b と同じ
  if (prev != bs) {
    prev = bs;
    if (bs == LOW) {
      state = 1 - state;
    }
  }
  digitalWrite(LED, state==1?HIGH:LOW);

  brt = (brt + 1) % 512;
  analogWrite(LIGHT, brt<256?brt:511-brt);
  delay(10);          // 速度調整兼チャタリング対策
}
```

21

## お題 (1) 解答例の補足

### 三項演算子

条件式 ? 式 1 : 式 2

条件式が真のとき式 1 に偽のとき式 2 になる式

次の 2 つは同じ結果になる

```
digitalWrite(LED, state==1?HIGH:LOW);
```

```
if (state == 1) {
  digitalWrite(LED, HIGH);
} else {
  digitalWrite(LED, LOW);
}
```

22

## お題 (2)

明るさを操作できる電気スタンドを実現する

- 7 番のスイッチを押すと 9 番の LED が点灯
- 再度短く押すと消灯
- 押し続けると明るさが徐々に変化する

### ヒント

millis() 関数は、プログラム開始からの経過時間をミリ秒単位で返す。(Processing にもあった！)

23

## お題 (2) ヒント続き

- 全体構造はお題 (1) 同様に、loop() 内で for とかしないで終わる形にする
- ボタンが押されてから 0.5 秒以上経過してまだ押されているなら、明るさを変化させるモードに
- それより早く離されたなら消灯に
- ON になった時刻は millis() 関数で調べ、保存しておく
- 答えはテキストの Example 05

24

## 整理

## これまでで制御できたもの

- デジタルなスイッチの読み取り
- 単体 LED の ON/OFF
- 単体 LED の PWM 制御

LED は消費電力が少ないため、マイコンでの制御に非常に有利。

26

## テキストにこの先書いてあること

- 明るさセンサ (CdS) からのアナログ読み取り
- Processing との連携
  - Arduino のセンサ読み取り結果を PC に送り、Processing プログラムがそれに反応する
  - 逆向きや双方向も可能
- モーターの駆動

世の中の大抵のものは LED より電力を必要とするので、arduino に直結しても動かない。

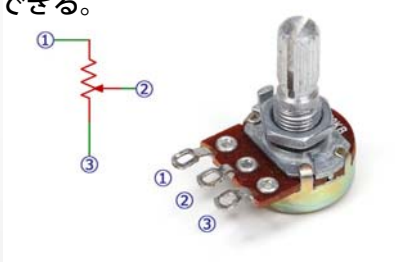
27

## 便利デバイス紹介

(画像は秋月電子より)

## 可変抵抗器

- つまみにより抵抗値を変化させる。次回説明するアナログ読み取りにより、つまみ位置が検出できる。



29

## ジョイスティック

- いわゆるアナログスティック。X と Y の両方に可変抵抗が入っている。



30

## ロータリーエンコーダ

- ジョグダイヤルの、どこまでもグルグル回せる。
- 1 単位回るたびに、回転方向も判るパルス信号が出てくる
- マイコンの中には、このパルス信号を勝手に数えてくれるカウンタがあるはずだが、arduino からは利用できないみたい。



31  
普通の使用例は多いので問題ない。

## 磁気センサ

- 磁石を仕込んだ対象が近づいたことを検出できる。
- アナログ出力
- 地磁気センサは高価。



32

## フォトリフレクタ / ホトインタラプタ

- 赤外線 LED とホトダイオード or ホトトランジスタの組み合わせで、「すぐそこに光を反射する何かがある」ことを検出する
- 赤外なので光ってるのかわかりにくい
- アナログ回路が少し必要



33

## 測距モジュール

- 明るさや対象の色に影響されず、距離が正確に測定できる。
- アナログ出力とデジタル出力、距離の範囲によりタイプがある。

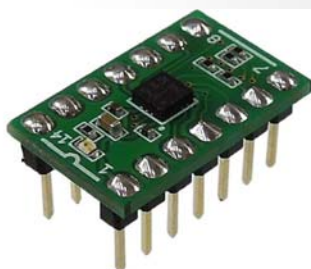


34

## 加速度センサ

p.172 Wire  
p.172 SPI

- X,Y,Z 軸の加速度が取れる
- スマホに内蔵されているそれ。
- アナログ出力タイプと、IIC (I2C), SPI 接続タイプがある。



35

## フルカラー LED

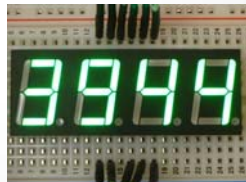
- 1 本に R,G,B の LED が入っている
- 片方の端子は共通なので 4 本足



36

## 7 セグメント LED

- 数字と小数点を表示
- 要はたくさん LED が入ったもの
- arduino のわずかなデジタル端子でも、右の 4 連タイプを制御できる。さらに倍もいける。



37

## ドットマトリクス LED

- 任意の表示が可能
- 通常、デジタル端子が縦 + 横だけ必要
- デジタル端子 22 本で、単色 8x8 ドットを 4 つ駆動した例がある



38

## マイコン内蔵 LED

- RGB 明度をデジタルで指示して点灯させる
- さらに複数個数珠つなぎにしても信号線は 1 本のままでよい



※製作例、Arduino。線材は別売りです

39

## パワー LED

- 100 ルーメン、 $3.3V \times 350mA = 1W$
- まぶしくて目を痛めるレベル
- 直接は制御できないし、外部電源も必要



40

## サーボモータ

p.180 Servo

- 1 回転の範囲で、指定した角度を自動的に維持する装置
- 電源や信号は 4.8V で 5V 系と相性がよく、信号も時間パルスで使え、arduino はライブラリも完備



41

## ステッピングモータ

p.170 Stepper

- ステッピングモータ・ドライバ・シールドと組み合わせて使う
- 絶対量で回転量と方向を指示するとそのように動くので、正確に動かすのは楽
- 力も強い
- 高価
- 重い



42

## DC モータ

- 安価だが、正確な制御は面倒
- とりあえず一定方向に一定電圧で回すレベルから、モータードライバ・シールドを利用した正逆回転・速度制御・ブレーキまで



43

## 液晶ディスプレイ

p.185 LiquidCrystal

- 英数カナ文字を 16 桁 ×2 行くらい表示する
- 大抵のものは互換性があり、ライブラリで使える
- I2C 接続タイプというのも使える



44

## スピーカー

p.142 tone() 関数

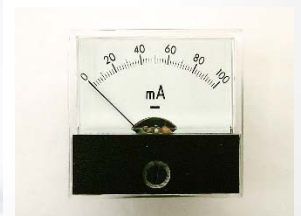
- 圧電タイプは簡単に音が鳴るが、キンキンする。
- ダイナミックタイプは単なるコイルなので、直結すると火を噴く。
- arduino から直接駆動すると結局キンキンするし、音量の動的制御も無理



45

## アナログメータ

- Arduino Cookbook のネタ
- analogWrite() の出力を送り込めば、針をアナログに動かせる



46

## AC アダプタ

- USB 動作では、500mA が限界
- モーターやパワー LEDなどを動かすと PC の USB 端子が焼ける
- 7 ~ 12V φ2.1 センタープラス 容量 1A くらい
- 一般には 9V が推奨、制作物で 5V が大量に必要なら 5V アダプタでも動く



arduino ボード上の三端子レギュレータにパワー系負荷をかけない

47

## 圧電スピーカで振動センサ

- 圧電スピーカは逆に、振動を拾って電気を発生させる。
- これを arduino に取り込むことで、「誰かが触った」「動かした」という情報を取り込むセンサにできる。

48