

第1講 基本的なプログラミングと電子回路の作成

Processing を使って絵を書く。基本的な電子回路を組む。

1 下準備

ノート PC の準備

本実習では、各個人に 1 台のノート PC を割り当てます。次回の授業からは開始までに、あらかじめロッカー内から自分が使用するノート PC を取り出して準備を行なってください。授業終了時には、ノート PC を収められていた箱に収納し、ロッカー内の指定位置まで返却してください。

このノート PC は、他の実習授業でも使用しますので、まず最初に各個人のアカウントを作成します。授業内の作業は、各個人アカウントで行うようにしてください。指示にしたがって、共有アカウントでログインし、個人用アカウントの作成、パスワードの設定を行なってください。

実習で用いる部品



図 1: Arduino UNO

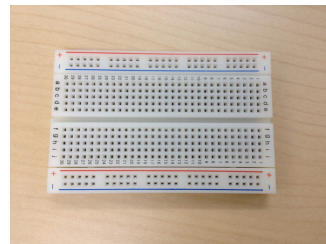


図 2: ブレッドボード



図 3: ジャンプワイヤ



図 4: 【図を差し替える】抵抗、スイッチ、センサ etc.

注意!

Arduino などの電子部品は壊れやすいので取り扱いに注意してください。

- 静電気 (Arduino が壊れてしまいます)
- 電子部品のピンは曲がりやすいので無理に差し込まないように注意! (壊れます)
- PC に刺したまま配線をいじらない (最悪、PC ごと壊れます …)

2 Processing の基礎

本実習では Processing¹ というプログラミング言語を用います。Processing は、Java を単純化し、グラフィック機能に特化した言語です。

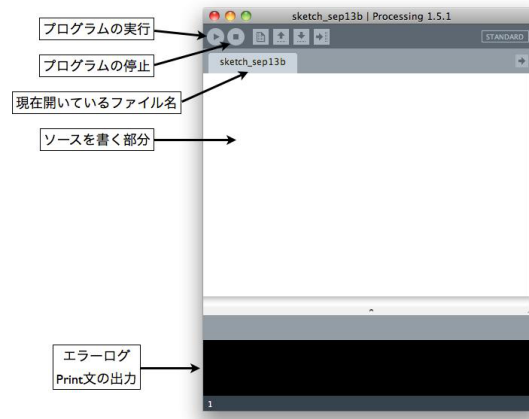


図 5: Processing の IDE (統合開発環境)

基本的な図形を描く

- 四角形を書く: `rect(x, y, w, h);`
 x = x 座標、 y = y 座標、 w = 四角形の幅、 h = 四角形の高さ
- 円を書く: `ellipse(x, y, w, h);`
 x = x 座標、 y = y 座標、 w = 円の幅、 h = 円の高さ
- 線を書く: `line(x1, y1, x2, y2);`
 $x1$ = 始点の x 座標、 $y1$ = 始点の y 座標、 $x2$ = 終点の x 座標、 $y2$ = 終点の y 座標

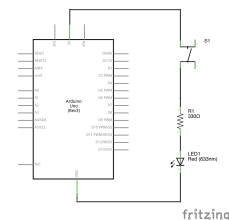
サンプルコード

```
// 画面サイズを幅300px、高さ200pxに設定
size(300, 200);

rect(10, 30, 120, 100); // 四角形を描く

line(10, 10, 290, 190); // 線を描く

ellipse(200, 120, 100, 150); // 円を描く
```



¹<http://processing.org>

図形の色を変える

- 図形に色をつける: `fill(r, g, b);`
r, g, b をそれぞれ 256 段階で指定
- 色を消す: `noFill();`
- 縁取り線に色をつける: `stroke(r, g, b);`
- 縁取り線の幅を調節する: `strokeWeight(weight);`
weight = 線の太さ
- 縁取り線を消す: `noStroke();`
- ウィンドウの背景色を指定する: `background(r, g, b);`

サンプルコード

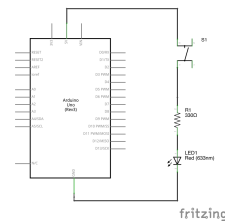
```
size(300, 200);

// 背景を黒に
// background(0, 0, 0); と同じ意味
background(0);

fill(255, 0, 0); // 四角形を赤に
rect(10, 30, 120, 100);

stroke(0, 255, 0); // 線を緑に
line(10, 10, 290, 190);

strokeWeight(3); // 線の太さを3pxに
fill(0, 0, 255); // 円を緑に
ellipse(200, 120, 100, 150);
```



アニメーションを作る

Processing では、プログラムを実行時に `setup` 関数が 1 度実行され (初期化)、その後 `draw` 関数が繰り返される (ループ処理)。

```
void setup(){
  // 初期化の処理をここに書く
  // プログラム開始時に1度だけ実行される
  size(200, 200); // ウィンドウサイズの指定
}

void draw(){
  // 繰り返したい処理をここに書く
  // setup()が実行された後にプログラムが終了するまで繰り返される
  line
}
}
```

- マウスの座標を取得する: mouseX, mouseY
- hoge

3 電子回路の基礎

電気、電流、抵抗のおさらい

よく、電気の流れは水の流れに例えられます。厳密に言えばさまざまな違いはありますが、その性質を大まかにとらえるには、水の流れに例えて理解するのは有効な方法です。

電圧

電流

抵抗

よく出てくる補助単位

電圧、電流、抵抗の単位はそれぞれボルト、アンペア、オームですが、実際にはこれに接頭辞がついた補助単位が用いられる場合があります。以下は、登場する補助単位の例です。

- 1,000 倍を表す単位がキロ (例: 10 k Ω)
- 1,000,000 倍を表す単位がメガ (例: 1 M Ω)
- 1/1,000 を表す単位がミリ (例: 10 mA)

オームの法則

電子部品に電圧をかけすぎたり、電流を流しすぎると壊れてしまう。そのために、電源側に抵抗器を配置し、電流量を調節する必要がある。

オームの法則は、

$$V(\text{電圧}) = I(\text{電流}) \times R(\text{抵抗}) \quad (1)$$

R (抵抗) を求めるために指揮を変形すると、

$$R = \frac{(\text{電源電圧} - LED \text{ にかかる電圧})}{LED \text{ に流したい電圧}} \quad (2)$$

スイッチを押したら LED を点灯させる

Arduino、スイッチ、抵抗、LED を用いて簡単な回路を組む。

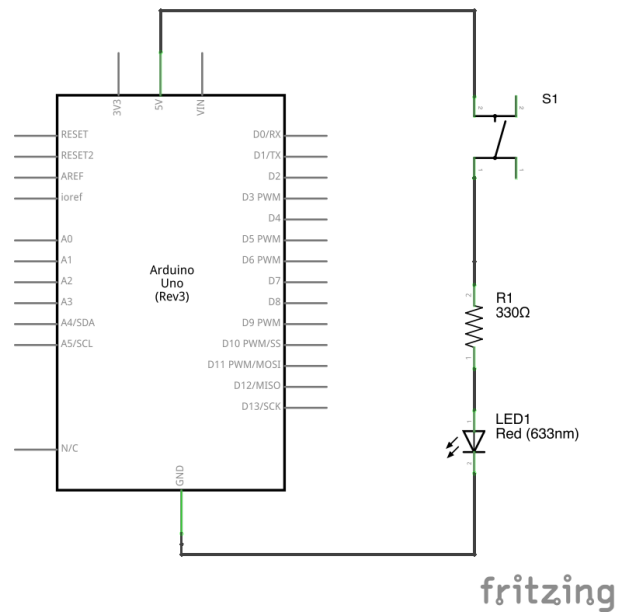


図 6: 回路図

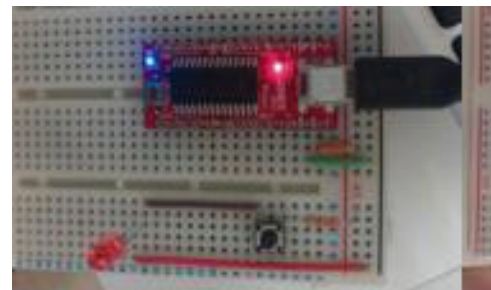


図 7: 配線例: タクトスイッチを離した状態だと LED が消灯し、押した状態だと LED が点灯する