

第5講 様々なセンサを用いる

1 本実習の目標

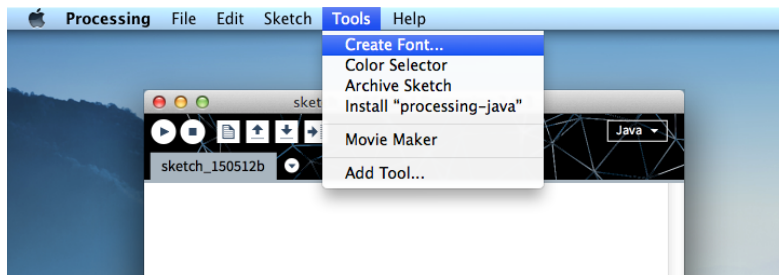
- Processing を用いて画面に文字を表示するプログラムを作成する。
- Arduino の Analog Input を用いて光センサの値を取得する。
- Processing を用いて画面に画像を表示するプログラムを作成する。
- Arduino の Analog Input を用いて圧力センサの値を取得する。

2 Processing の画面に文字を表示する

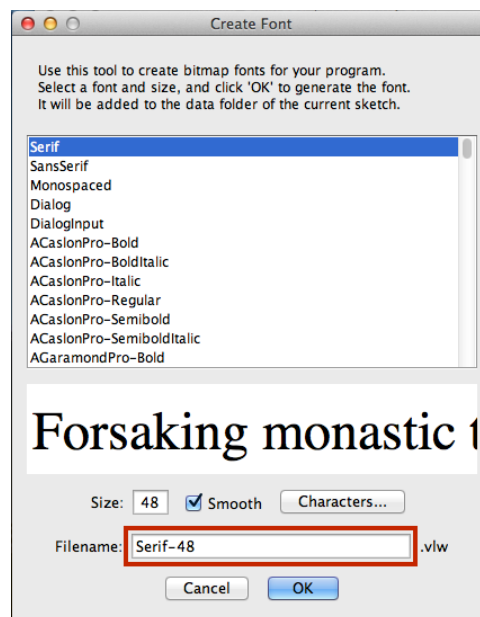
Processing では、矩形や円のような図形だけではなく、文字を描くこともできます。文字を描画するには `text()` を用います。また、PFont と `textFont()` を用いることによってフォントの種類や大きさを変更することができます。

PFont を用いて任意のフォントを使用するためには、フォントファイルを用意しなければなりませんが、Processing には簡単にフォントファイルを用意するための機能が備わっています。

1. メニューから Tools → Create Font を選択する



2. 使いたいフォントを選択する



3. フォントの名前が表示されているのを確認する (例、Serif-48.vlw ← コイツを覚えておくこと!)

4. OK ボタンを押す

これで、data フォルダ¹の中にフォントファイルが生成されたと思います。loadFont() を用いて生成したフォントファイルを読み込むことで、好きなフォントを使うことができます。

```
PFont font;

void setup() {
  size(300, 300);

  // Create Font で作ったフォントを読み込む
  font = loadFont("Serif-48.vlw");
}

void draw() {
  background(255);

  textFont(font, 32); // 使うフォントとその大きさの指定
  fill(0);
  text("Interactive Art", 32, 32); // 文字を描く
}
```

このプログラムを実行すると、Processing のウィンドウの中に Interactive Art の文字列が描画されると思います。できましたか？

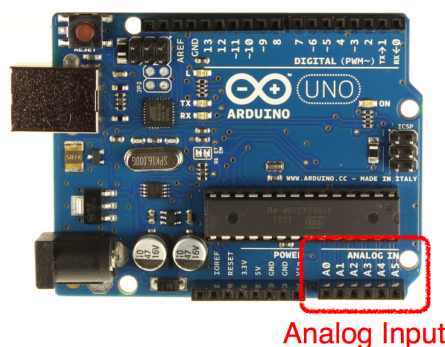
3 Analog Input

Arduino には、Analog Output と同様に Analog Input もあります。Analog Input を用いる場合は A0 ~ A5 ピンに接続してください。

Processing から Analog Input を使って値を取得するには、

```
arduino.analogRead(pinNum);
```

を用います。pinNum は 0 ~ 5 (それぞれ、A0 ~ A5 に対応) で指定してください。Digital Input を用いたときのような、PIN モードの設定は必要ありません。



¹作成しているプログラムと同じフォルダの中にある。フォントファイルや画像ファイルなど、プログラムから読み込みたいファイルなどはここに入れておくとい。メニューの Sketch → Show Sketch Folder から作成しているプログラムが入っているフォルダを開ける。

4 光センサを使う

光センサ (図 1) とは、表面に当たる光の量に従って抵抗値が変化する電子部品です。周りが暗いと抵抗値が大きくなり、明るいと抵抗値が小さくなります。テスターを使って抵抗値の変化を測ってみましょう。

実習で使う光センサは CdS セル (CdS: 硫化カドミウム) を主成分とするものです。

- 利点

- 可視光線に対して高感度
- 小型で軽量
- 比較的安価

- 欠点

- 反応速度がやや遅め
- カドミウム = 有害物質 (ゼットイに食うな！)

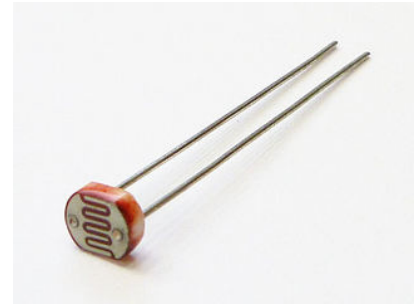


図 1: CdS セル

光センサの値を取得する

光センサの値を取得するためには、Analog Input を用います。Analog Input を使う場合は前述のように A0 ~ A5 ピンを用いる必要があります。接続するピンに注意してください。

回路

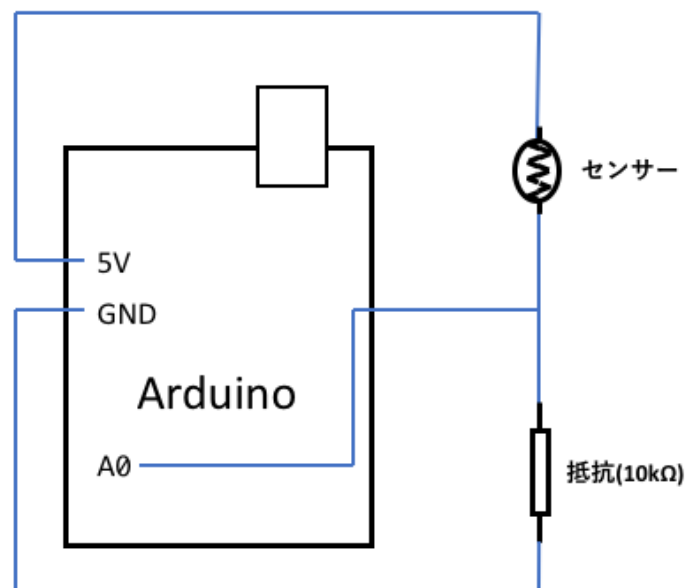


図 2: 光センサの回路図

プログラム

```
import processing.serial.*;
import cc.arduino.*;

Arduino arduino;
int sensorPin = 0; // 光センサを A0 番ピンに接続する場合
PFont font;

void setup() {
    size(300, 300);

    arduino = new Arduino(this, Arduino.list()[3], 57600);
    font = loadFont("Serif-48.vlw");
}

void draw() {
    background(255);

    int sensorValue = arduino.analogRead(sensorPin); // Analog Input

    textFont(font, 32);
    fill(0);
    text("sensor value: " + sensorValue, 32, 32);
}
```

5 Processingで画像を表示する

Processing では、矩形や円などの単純な図形と同様に画像を表示することができます。画像を表示するにはまず、画像を読み込む必要があります。

準備として、data フォルダに使いたい画像ファイルをコピーしてください。わざわざ data フォルダを開かなくても、Processing のウィンドウにドラッグ&ドロップすると勝手にコピーされます。

```
PImage img;

void setup() {
    size(400, 400);

    img = loadImage("hoge hoge.jpg"); // 使いたい画像ファイルの名前を指定
}

void draw() {
    background(255);

    // 画像を表示
    // PImage、x座標、y座標を指定
    image(img, mouseX, mouseY);

    // 幅と高さを指定するとリサイズしてくれる
    image(img, 0, 0, 120, 90);
}
```

6 曲げセンサを使う

曲げセンサ (図 3) とは、センサ本体を曲げるにより抵抗値が変化するセンサです。プリント面を外側にした状態で曲げると抵抗値が増加します。1 つの曲げセンサだけでは、片方向への曲げしか検出できませんが、2 つを重ねて用いることによって、両方向の曲げを検出できます。

曲げセンサ (や圧力センサなど) は、LED や抵抗、光りセンサなどの部品よりも非常に高価なので壊さないように注意してください！

- ピンが折れやすいので注意すること！
- ブレッドボードに差すときは、ピンに近い部分を持つこと (無理に差さないように)
- フィルム部分を曲げすぎないように

回路

先ほど 4 節で作成した回路とほとんど同じです。光センサを曲げセンサに変えるだけです。

プログラム

```
import processing.serial.*;
import cc.arduino.*;

PImage img;

Arduino arduino;
int sensorPin = 0; // 曲げセンサを A0 番ピンに接続する場合

void setup() {
  size(500,500);

  arduino = new Arduino(this,Arduino.list()[3],57600);
  img = loadImage("hoge hoge.jpg"); // 使いたい画像ファイルの名前を指定
}

void draw() {
  background(255);

  int sensorValue = arduino.analogRead(sensorPin);

  image(img, 0, 0, sensorValue, sensorValue);
}
```

7 様々なセンサ

圧力センサ (図 4) は、感圧面に加重することにより抵抗値が変化するセンサです。タクトスイッチなどの、ON/OFF を検出するスイッチとは違い、押した強さを検出できます。

赤外距離センサ (図 5) は、赤外線を用いてセンサと物体との距離を測定することができるセンサです。(第 6 講で使用します)

圧力センサや赤外距離センサも、光センサや曲げセンサと同様の回路とプログラムで使うことができます。

この他にも、心拍センサや加速度センサ、温度センサなど様々な種類のセンサがあります。

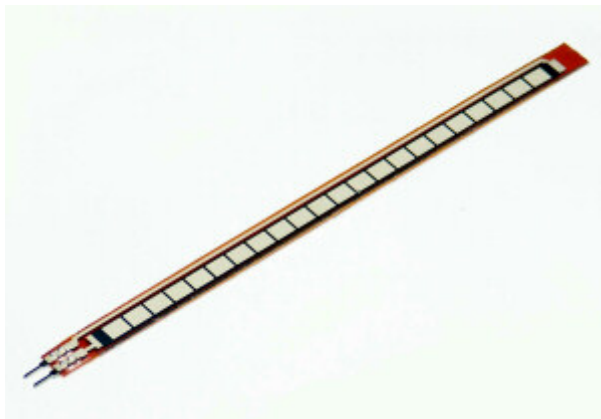


図 3: 曲げセンサ



図 4: 圧力センサ

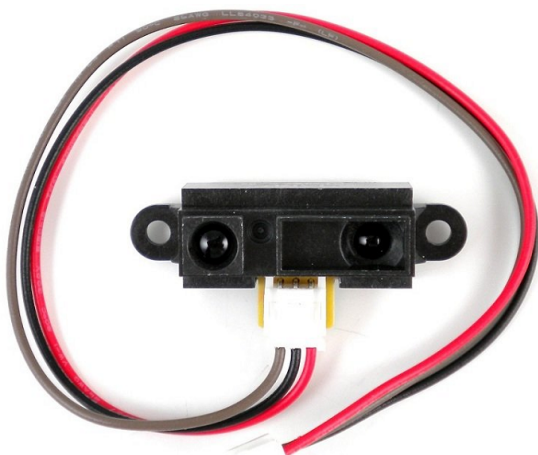


図 5: 赤外距離センサ