

第2講 プログラミングでLEDを制御する

1 本実習の目標

- Processing に Arduino (Firmata) をインストールする。
- Processing から Arduino を制御するためのシリアルポートを把握する。
- Arduino の Digital Output を用いて LED を光らせる。
- Arduino の Digital Input を用いてスイッチの ON/OFF を取得する。

2 Arduino の準備

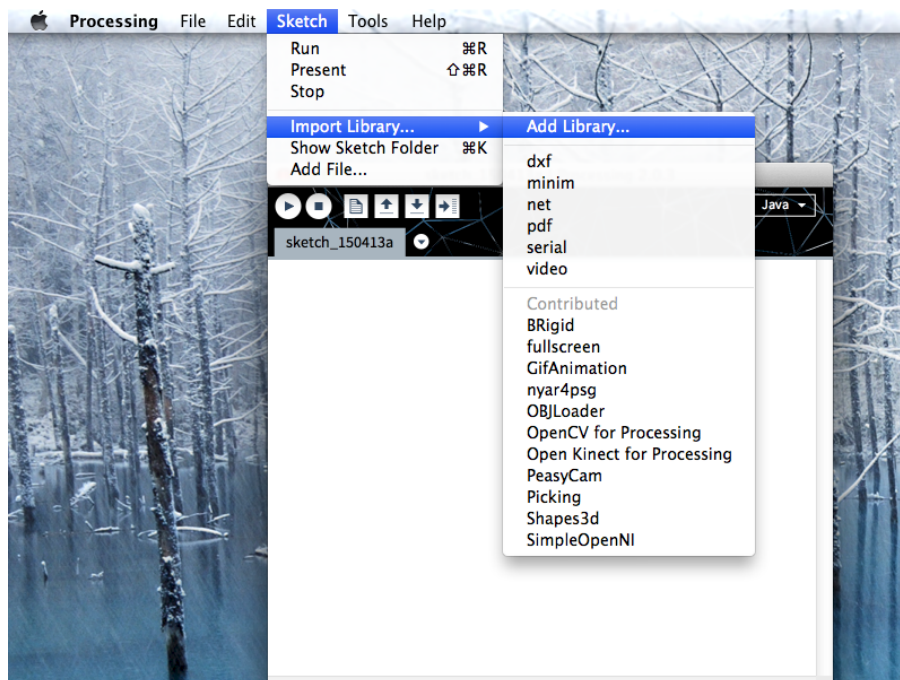
まず、Processing から Arduino を用いるための準備をしましょう。本実習ではスイッチやセンサ、LEDなどをPCから制御するために Arduino¹ というマイコンを用います。

本来、Arduino と Processing を連携させるためには、シリアル通信を用いますが、そのためのプログラムを書くのは少し面倒です。実習では楽をするために、Firmata² を用います。

ライブラリのインストール方法

Processing のメニューから以下のライブラリを追加します。

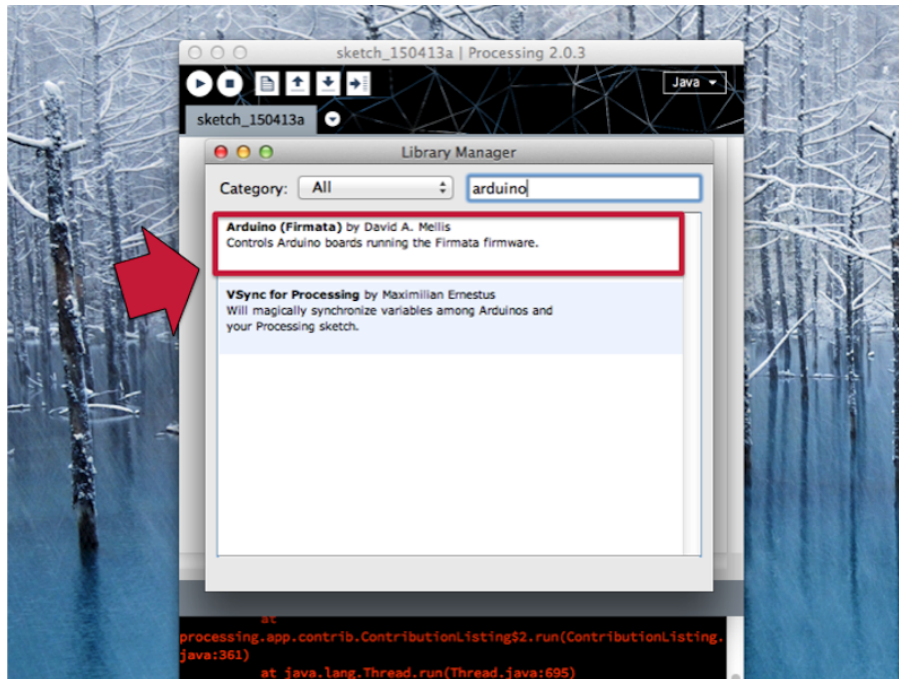
- Sketch Import Library... Add Library...



¹<http://arduino.cc>

²<http://firmata.org>

- 図 2 のように検索バーで「arduino」と入力し、「Arduino(Firmata)」を選択します。



これでライブラリのインストールは完了です。

3 シリアルポートの確認

Processing から Arduino にインストールした Firmata を操作するには、使用しているシリアルポートの環境を知る必要があります。まず、下記のコードを Processing に入力してください。

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

void setup() {
  println(Arduino.list());
}
```

すると、Processing の下部のコンソールに以下のようなメッセージが表示されるはずです。

```
/dev/cu.Bluetooth-Incoming-Port
/dev/cu.Bluetooth-Modem
/dev/cu.usbmodem1421
/dev/tty.Bluetooth-Incoming-Port
/dev/tty.Bluetooth-Modem
/dev/tty.usbmodem1421
```

次に、「/dev/...」から始まる記述に対して、それぞれ順番に 0 から番号を付与します。

```
[0] /dev/cu.Bluetooth-Incoming-Port
[1] /dev/cu.Bluetooth-Modem
[2] /dev/cu.usbmodem1421
[3] /dev/tty.Bluetooth-Incoming-Port
[4] /dev/tty.Bluetooth-Modem
[5] /dev/tty.usbmodem1421
```

この中から、「/dev/tty.usbserial...」もしくは「/dev/tty.usbmodem...」から始まる記述の番号(上記の例では5番)をメモしておいてください。Arduino を使うためにはシリアルポートの指定をしなければなりません。続いて、下記のコードの `Arduino.list()[5]` の部分の数字を先ほどメモした番号に変更し、Processing に入力してください。

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

Arduino arduino;

void setup() {
  // Arduino の初期化
  // シリアルポートの指定など
  // Arduino.list()[5] は環境によって変える
  arduino = new Arduino(this, Arduino.list()[5], 57600);
}
```

すると、Processing の下部のコンソールに以下のようなメッセージが表示されるはずです。

```
/dev/tty.usbmodem1421
```

これで Processing から Arduino を制御するためのシリアルポートを知ることができます。これらの命令は今後も Arduino を用いる際に必ず使うので忘れないようにしてください。

4 Digital Output

では、まず簡単なプログラムで動作を確認してみましょう。LED が点灯するプログラムを作成してみましょう。Arduino 側は、Digital Out の13番にLEDを接続しておきます。

4.1 LED を点灯させる

Digital Output を使って LED を点灯させてみましょう。

回路を組む

LED を接続したピンに電圧をかけると LED が点灯する回路を作りましょう。

- 使う部品

- Arduino
- 抵抗
- LED

- ポイント

- 5V LED 抵抗 GND の順に
- LED には極性があるので向きに注意

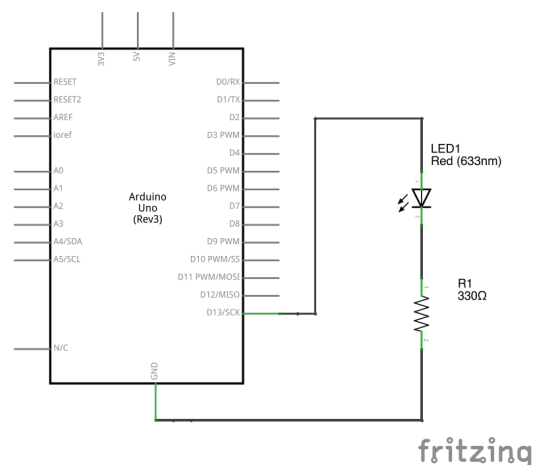


図 1: 回路図

プログラムを書く

LED が点灯するプログラムを書きましょう。

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

Arduino arduino;
int ledPin = 13; // LED を接続したピンの番号

void setup() {
  arduino = new Arduino(this, Arduino.list()[5], 57600);

  // Arduino のピンモードを設定
  // ここでは 13 番ピンを Output 用に設定
  arduino.pinMode(ledPin, Arduino.OUTPUT);
}

void draw() {
  // Arduino の 13 番ピンを HIGH (5V) に
  arduino.digitalWrite(ledPin, Arduino.HIGH);
}
```

これで LED が点灯するはずです。

4.2 LED を点滅させる

Digital Output を使って LED を点滅させてみましょう。

プログラムを書く

LED を接続したピンの電圧を制御するプログラムを書きましょう。

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

Arduino arduino;
int ledPin = 13; // LED を接続したピンの番号

void setup() {
  arduino = new Arduino(this, Arduino.list()[5], 57600);

  // Arduino のピンモードを設定
  // ここでは 13 番ピンを Output 用に設定
  arduino.pinMode(ledPin, Arduino.OUTPUT);
}

void draw() {
  // Arduino の 13 番ピンを HIGH (5V) に
  arduino.digitalWrite(ledPin, Arduino.HIGH);
  delay(500); // 500ミリ秒間待つ

  // Arduino の 13 番ピンを LOW (0V) に
  arduino.digitalWrite(ledPin, Arduino.LOW);
  delay(500);
}
```

```
}
```

これで LED が点滅するはずです。

4.3 Processing の入力に応じて LED を点灯

Processing の画面上でマウスを押すと LED が点灯するプログラムを作成してみましょう。Arduino 側は、Digital Out の 13 番に LED を接続しておきます。

プログラムを書く

Processing の画面上でマウスを押すと LED が点灯するプログラムを書きましょう。

```
/*
 * Arduino - Processingシリアル通信
 * Firmataを使用したサンプル
 * Processing側
 */

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

Arduino arduino;
int ledPin = 13;
color bgColor = color(0);

void setup() {
    size(400, 200);
    arduino = new Arduino(this, Arduino.list()[5], 57600);
    arduino.pinMode(ledPin, Arduino.OUTPUT);
}

void draw() {
    background(bgColor);
}

void mousePressed() {
    arduino.digitalWrite(ledPin, Arduino.HIGH);
    bgColor = color(255,0,0);
}

void mouseReleased() {
    arduino.digitalWrite(ledPin, Arduino.LOW);
    bgColor = color(0);
}
```

画面をクリックすると、LED が点灯するはずです。

5 Digital Input

スイッチの ON/OFF を読み取る

Arduino の Digital Input を用いてスイッチの ON/OFF を取得する。

回路を組む

デジタル回路の場合、入力端子がどこにも接続されていないような状態（オープン）が起こると、電圧が High または Low に定まらず誤動作の原因になります。そのため、回路を安定させるためにプルアップ抵抗/プルダウン抵抗と呼ばれる抵抗を用います。

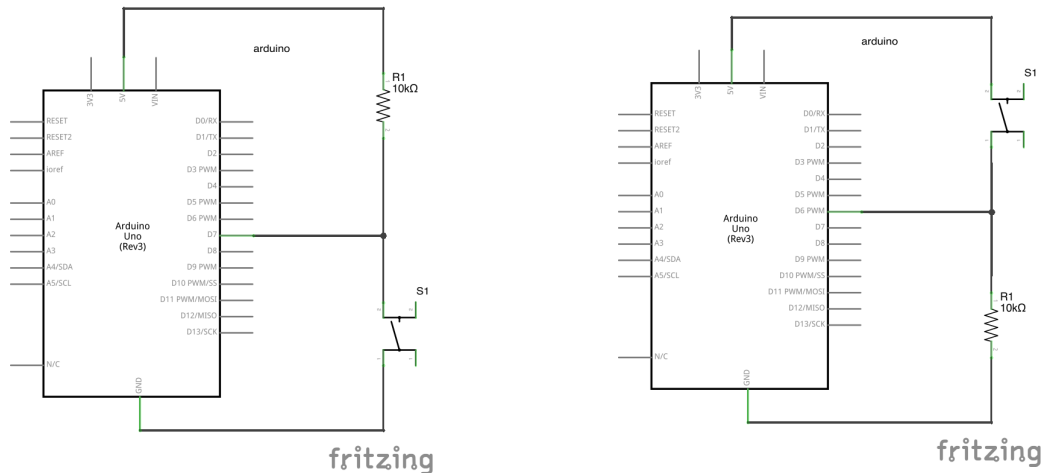


図 2: プルアップ抵抗 (左) とプルダウン抵抗 (右)

プログラムを書く

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

Arduino arduino;
int switchPin = 8; // スイッチを接続したピンの番号

void setup() {
  size(400, 300);
  arduino = new Arduino(this, Arduino.list()[5], 57600);
  arduino.pinMode(switchPin, Arduino.INPUT); // ピンモードを Input に
}

void draw() {
  // 8 番ピンの電圧を取得し、それが HIGH ならば
  if (arduino.digitalRead(switchPin) == Arduino.HIGH) {
    background(255, 0, 0); // 背景を赤に
  } else {
    background(0, 0, 0); // そうでなければ (LOW ならば) 背景を黒に
  }
}
```