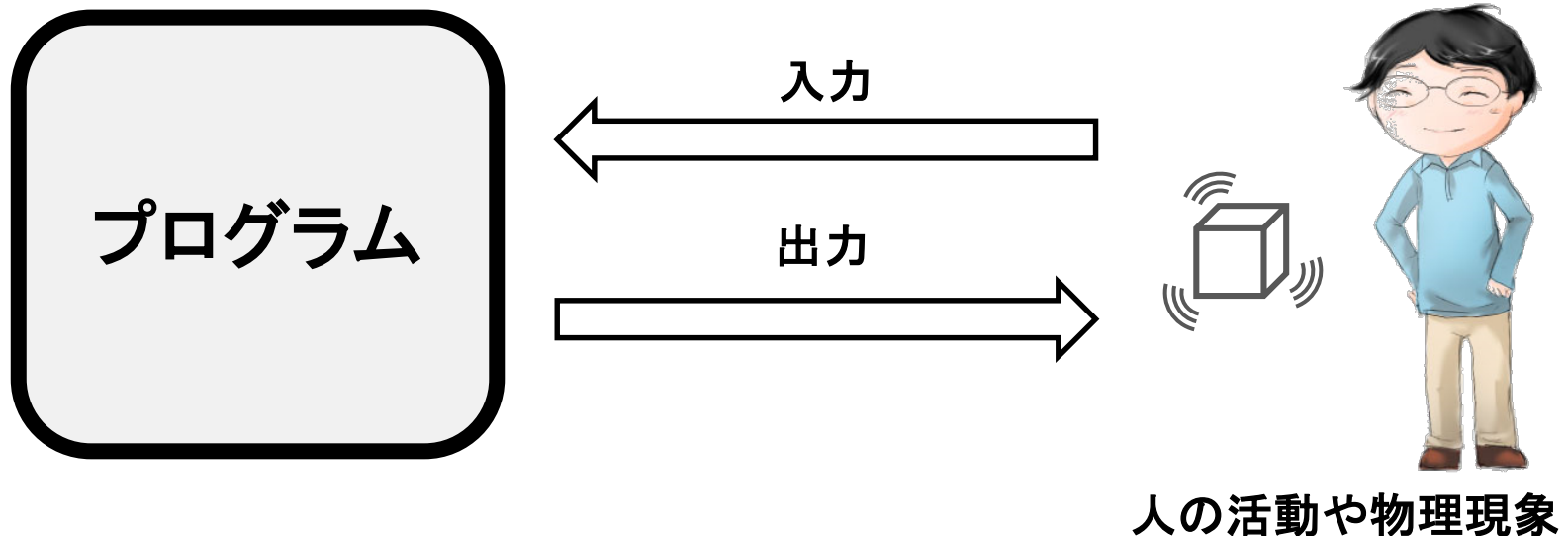

プログラミング演習Ⅱ フィジカルコンピューティング

第1回 Arduinoをはじめよう

中村, 小松, 小林, 鹿喰
(監修: 橋本直)

フィジカルコンピューティングとは

- コンピュータにさまざまな入出力デバイスをつないで実世界との物理的なやりとりを実現する方法
- プログラミングと電子工作の融合
 - センサを使ってプログラムを制御
 - 電氣的な装置をプログラムから操れる！



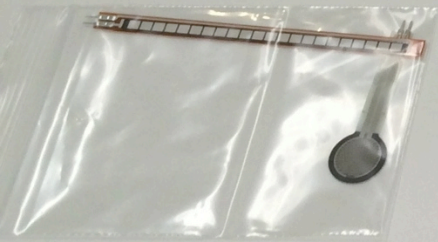
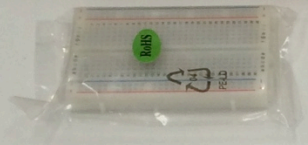
センサとアクチュエータ

- センサ（入力）
 - 光センサ、赤外線センサ、距離センサ、温度センサ、曲げセンサ、マイク、振動センサなどなど
- アクチュエータ（出力）
 - モータやソレノイド（電磁石で動く機構）などなど
- その他の出力デバイス
 - LEDやスピーカなどなど

配布キット

- 1人1個！
- 自由に使ってOK！
- 自分で管理すること

パーツ	役割
Arduino Leonardo	マイコンボード。電氣的な挙動をプログラミングできる。
USBケーブル	プログラムの書き込み、PCとの通信、電源供給に使う。
ブレッドボード	パーツを挿して、はんだ付けすることなく電子回路を作る。
ジャンプワイヤ(3種)	ブレッドボード上の配線に使う。
電池、電池スナップ	Arduinoを電池駆動するのに使う。
サーボモータ	角度を制御できるモータ
LED	発光ダイオード。光を出す素子。
マトリクスLED	発光ダイオードを8x8に配列した素子。
フルカラーLED	RGB3色の光を出すLED。
圧電スピーカー	ピピピという電子音を出したり、音を感知したりできる。
感圧センサ	圧力の量で抵抗値が変化する素子。
曲げセンサ	曲げの量で抵抗値が変化する素子。
フトリフレクタ	赤外線を発光して、その反射量を計測するセンサ。
タクトスイッチ	小さな押しボタンスイッチ。
CdSセル	光の量で抵抗値が変化する素子。CdSとは硫化カドミウム。
ボリューム	まわすと抵抗値が変化する素子。
抵抗	過電流が流れるのを防いだり、電圧を分圧するのに使う。
ツェナーダイオード	定電圧ダイオード。電圧の安定化に使う。





サーボモータ



曲げセンサ

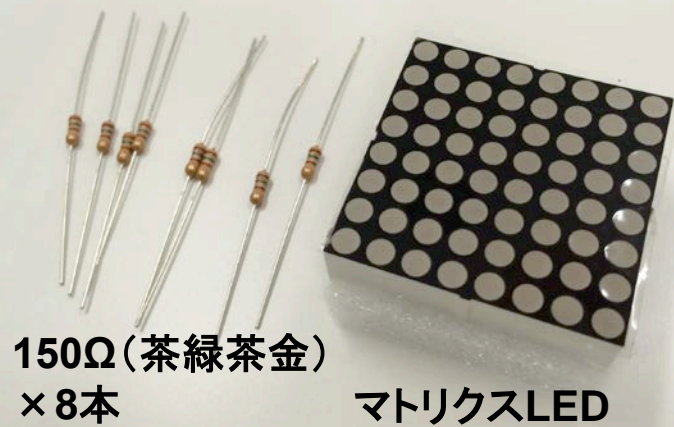
感圧センサ



圧電スピーカ

1MΩ(茶黒緑金)

ツェナーダイオード



150Ω(茶緑茶金)
× 8本

マトリクスLED



LED
赤・緑・青

タクト
スイッチ

可変抵抗
ボリューム

フルカラーLED
拡散キャップ

CdSセ
ル

フォトリフレ
クタ



150Ω 茶緑茶金



330Ω 橙橙茶金



1KΩ 茶黒赤金



10KΩ 茶黒橙金

Arduino (アルドゥイーノ)

- Arduinoはマイコンボード (micro-controller board)
- 電気を制御するプログラムを書き込める
- PCとの通信もできる
- 電池をつなげば単体で動作する



プログラミングと電子回路

- プログラムで電子回路を操るとはということ？
 - **【電圧を読み取る】**
 - センサで計測される物理量は電圧で表現される
 - **【電圧を変える】**
 - LEDやモータを駆動するときの指令値も電圧で表現されるこれら2つの実行手順をプログラムによって記述する
- デジタル情報の表現
 - コンピュータの世界では「0」と「1」で表現される
 - 電子回路の世界では「LOW(0V)」と「HIGH(5V)」で表現される

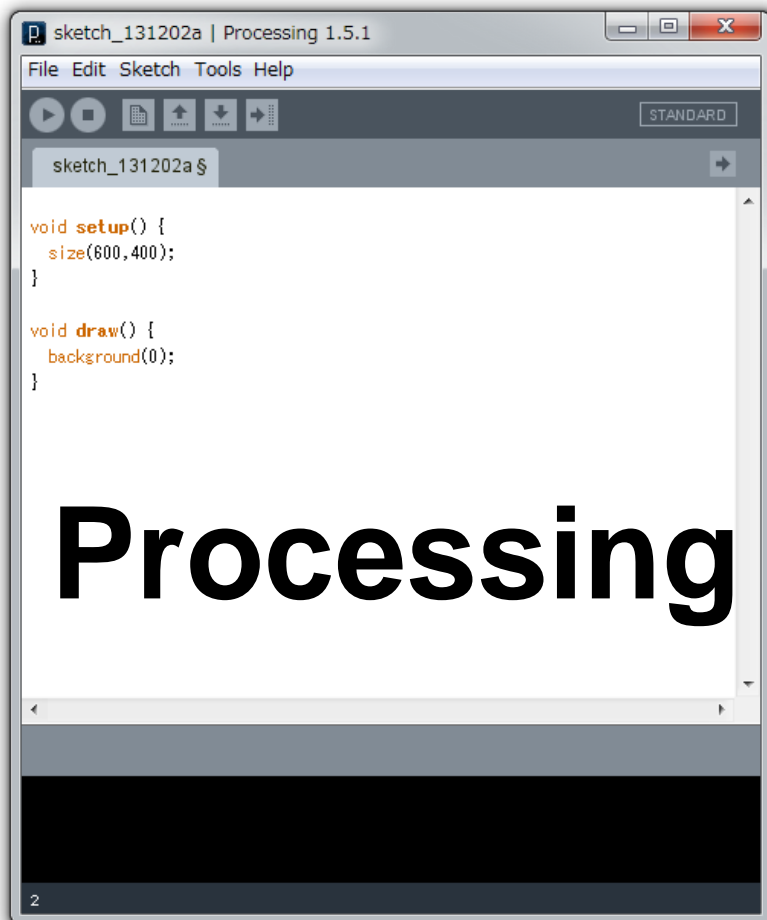
Arduinoをインストールしよう

- Arduino IDE: プログラミングをするツール
 - IDEとは統合開発環境のこと (Integrated Development Environment)
- Arduinoの公式サイトからダウンロード
 - <http://www.arduino.cc/>
 - `arduino-1.0.6-windows.exe` を実行してインストール



ArduinoとProcessingは似てる！（１）

- インタフェースがそっくり！



ArduinoとProcessingは似てる！（2）

- プログラムの構造もそっくり！

Processing

```
void setup() {  
    // 最初の1回だけ実行される  
}  
  
void draw() {  
    // なんども繰り返し実行される  
}
```

Arduino

```
void setup() {  
    // 最初の1回だけ実行される  
}  
  
void loop() {  
    // なんども繰り返し実行される  
}
```

違うのはココ！
↓

Arduinoのプログラミング

- ArduinoはC/C++言語ベース
- Processingとは違う言語だが、演算の記法や基本的な構文 (if, for, while, switch)、コメントアウトなどの作法は同じ
- 配列の書き方がちょっと違うくらい

Processing

```
int[] data;  
data = new int [10];  
  
data[0] = 1;  
data[1] = 2;
```



Arduino

```
int data[10];  
  
data[0] = 1;  
data[1] = 2;
```

Arduino 日本語リファレンス

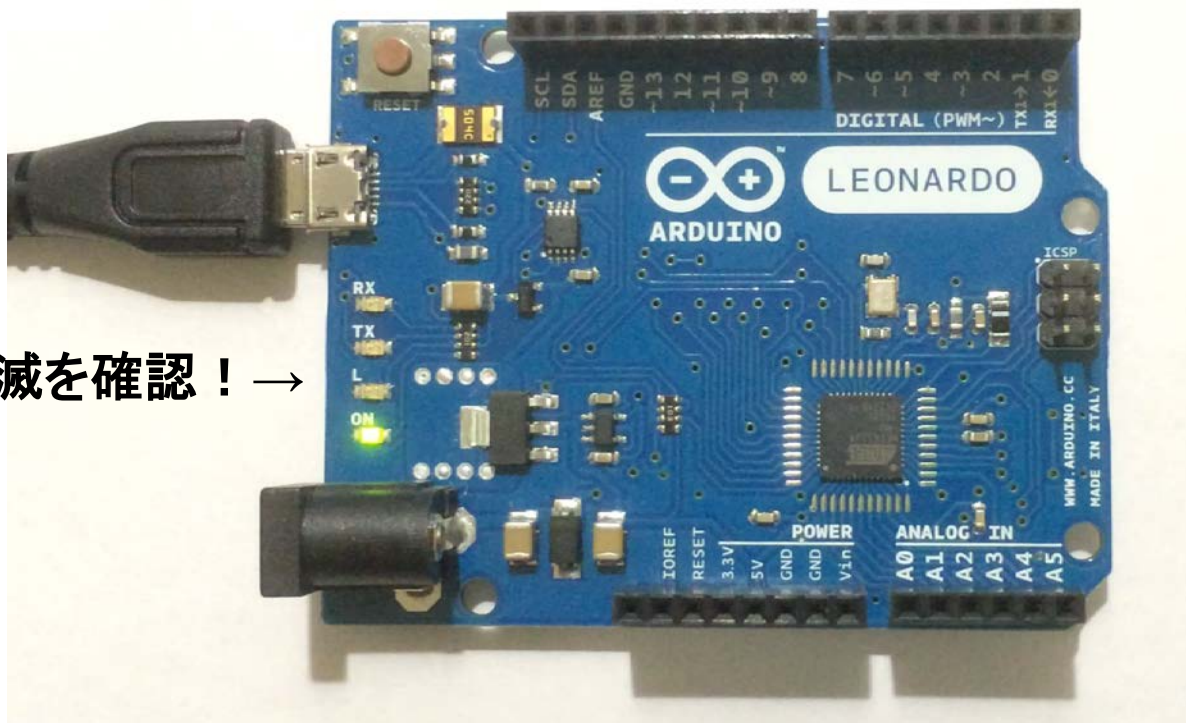
<http://www.musashinodenpa.com/arduino/ref/>

- わからない命令については上記参照
- 新しく覚える命令はそれほど多くない

- USBケーブルでArduinoとPCを接続
- Arduino IDEを起動
- [ツール→マイコンボード]でArduino Leonardoを選択
- [ツール→シリアルポート]でArduinoが接続されているUSBポートを選択(「COM??」等)
- サンプルBlinkを開く(ファイル→スケッチの例→01.Basics)
-  (検証ボタン)を押して文法チェック
-  (マイコンボードに書込むボタン)を押して書込む
- Arduino上で自動的に実行される
- `delay(1000);` の数字を変えて書込んでみよう

ArduinoのLEDがチカチカするか確認！

このLEDの点滅を確認！ →



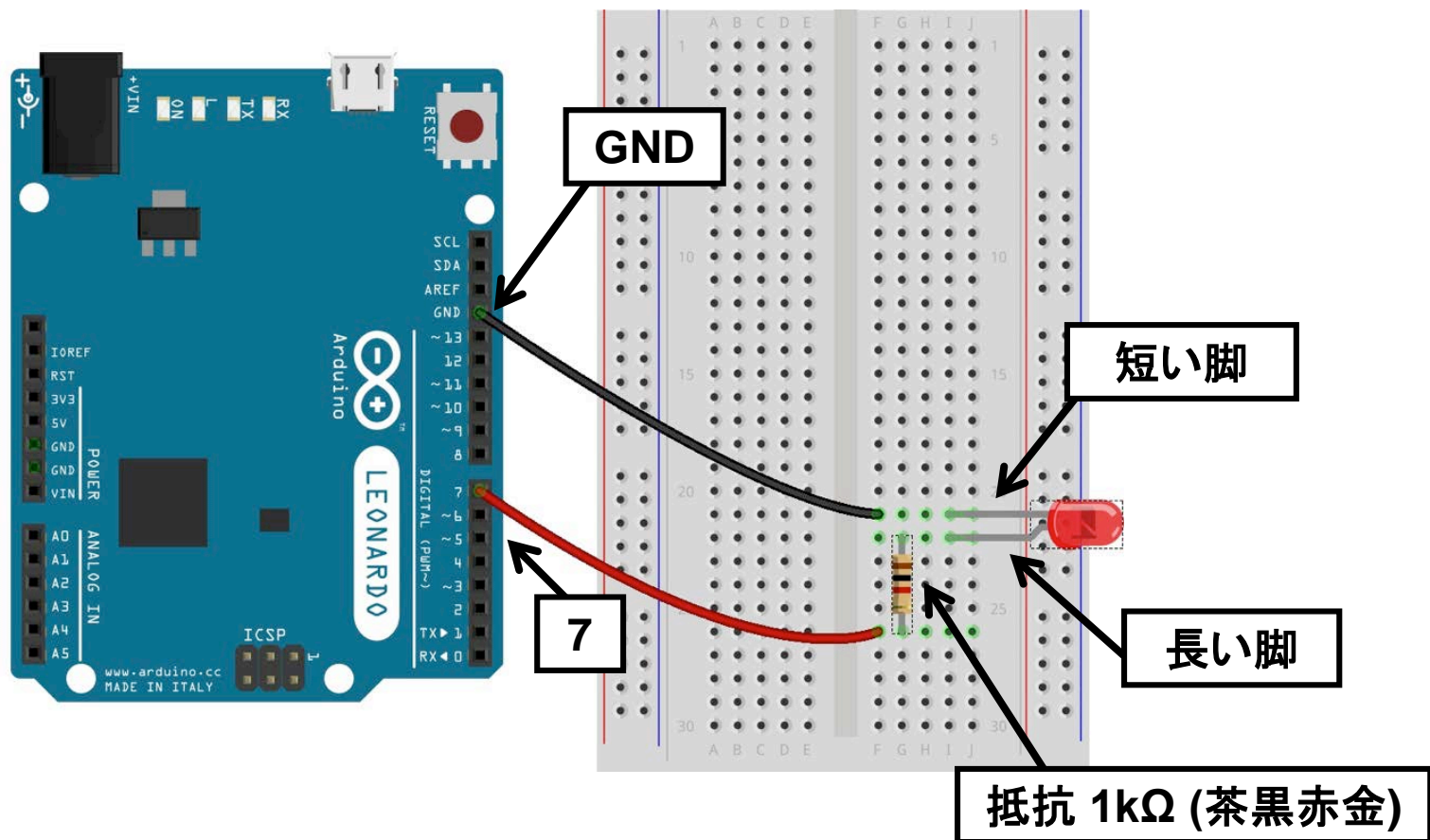


Arduinoでつくる最初の回路

- 電氣的な【入力】と【出力】のプログラミングをやってみよう
- ① LEDを点滅させる回路とプログラムを作る
 - ② スイッチを押したらLEDが点滅する回路とプログラムを作る

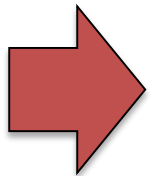
① LEDを点滅させる回路

- 入門では定番の回路。LEDチカチカ(略してLチカ)
- 抵抗の色、端子の番号、LEDの脚の向きに注意



注意

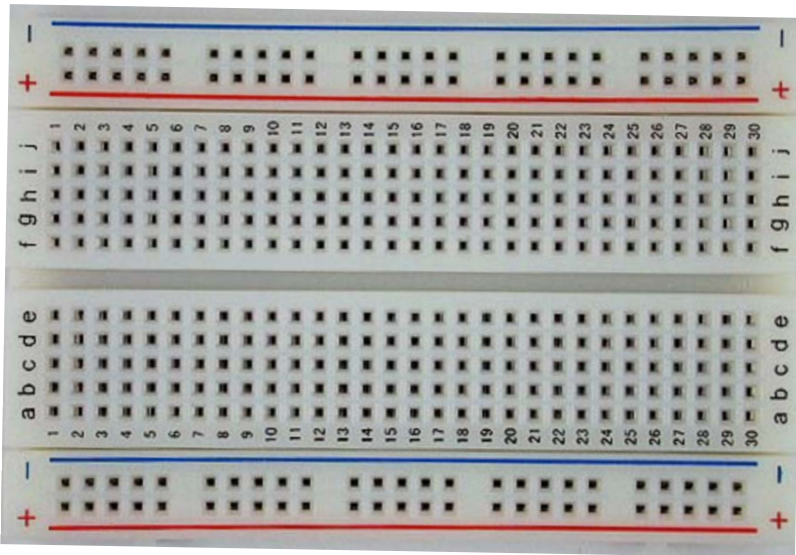
- GNDと、5Vや3.3Vを直接つなげると、ショートして壊れます
- LEDをつなげる時は、保護用に抵抗を直列につなぎます
- 電球やモータを出カピンにつないでも動きません



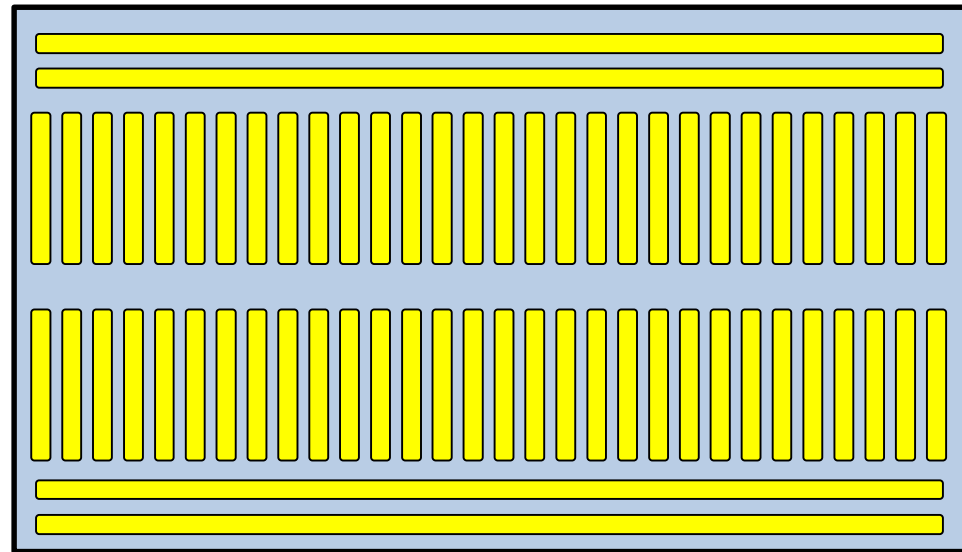
電流が流れ過ぎたり、足りなかつたりするからです。Arduinoに電源をつなげる前に回路を確認しましょう！

ブレッドボード

- パーツやワイヤを挿して電子回路を作れる道具
- 内部に金具が入っていて、列単位でつながっている



穴にピンや部品を挿して配線

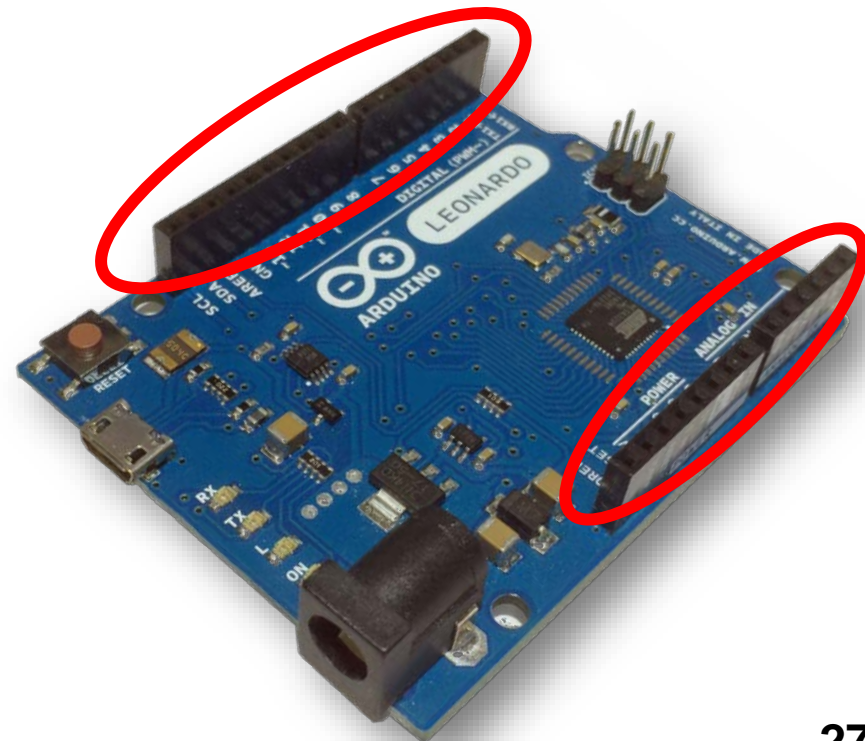


中はこのようにつながっている！

※実験やプロトタイプ用

Arduinoの端子

- 【2】～【13】 デジタル入出力端子
 - プログラムで入力電圧を読みとったり、出力電圧を変えることができる
- 【A0】～【A5】 アナログ入力端子
 - 入力電圧を1024段階で読み取れる
(Leonardoでは、デジタル入力端子の4, 6, 8, 9, 10, 12も使用可能; 【A6】～【A11】に対応)
- 【5V】
 - 5Vの電圧が常時出力されている
 - 電源のプラス端子に相当
- 【GND】
 - グラウンドと読む (groundの略)
 - 電源のマイナス端子 (0V) に相当
 - 3つあるが、どれにつないでも同じ



LED点滅のプログラム

```
void setup() {  
  pinMode( 7, OUTPUT );  
}
```

7番ピンを「出力」に設定

```
void loop() {  
  digitalWrite( 7, HIGH );  
  delay(500);  
  digitalWrite( 7, LOW );  
  delay(500);  
}
```

LEDをONにする

500ms 待つ

LEDをOFFにする

500ms 待つ

命令の意味

- **pinMode(ピン番号, 入出力設定)**
 - 指定したピンの入出力モードを決める
 - 入力にしたければ INPUT、出力にしたければ OUTPUT
- **digitalWrite(ピン番号, 出力状態)**
 - 指定したピンの出力電圧 (5V or 0V) を決める
 - 5V にしたければ HIGH、0V にしたければ LOW
- **delay(時間)**
 - 指定された時間だけ待つ
 - 時間はミリ秒単位で指定

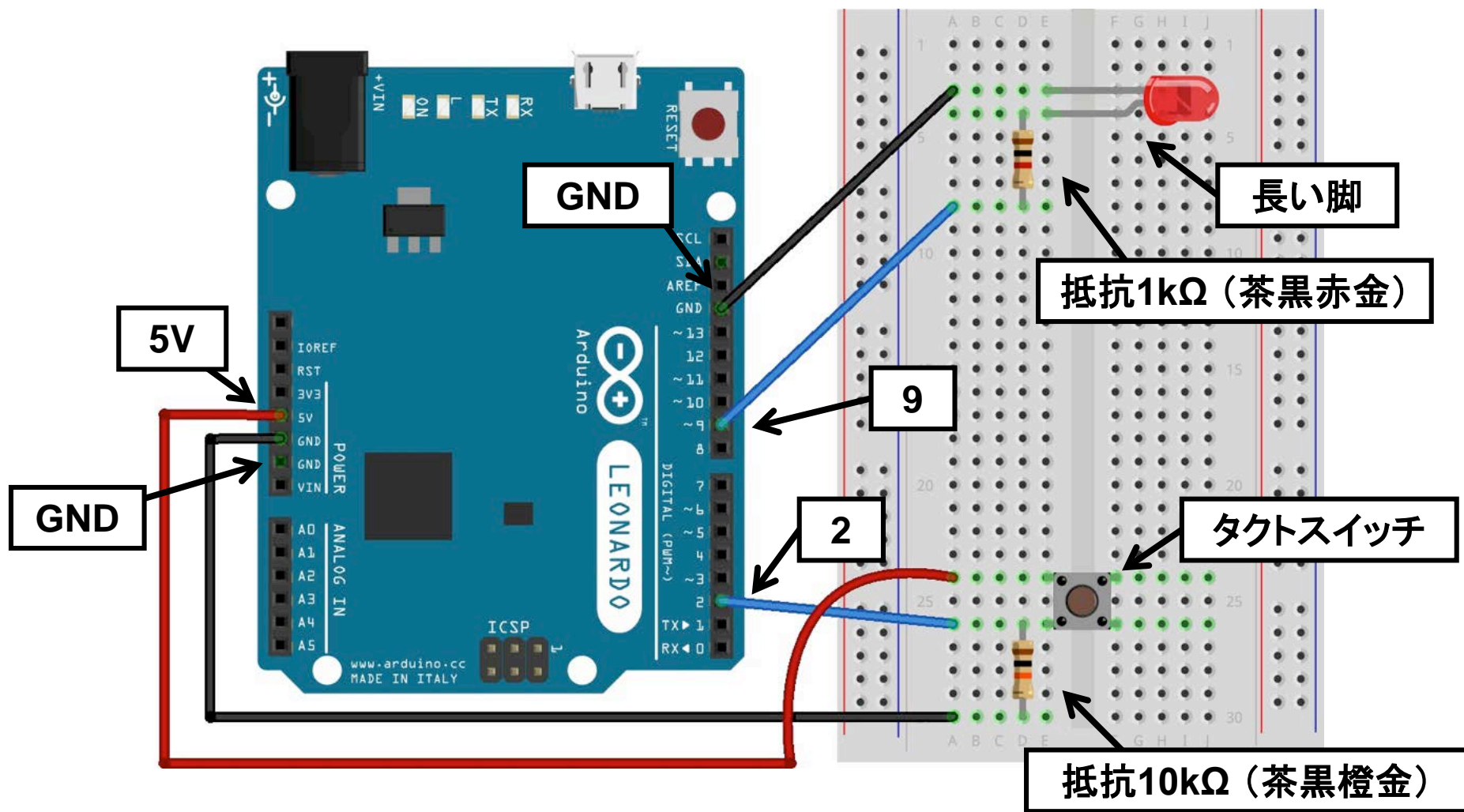
プログラムの保存

- 「ファイル」→「名前を付けて保存」
 - Processingと同様、ここで指定した名前と同名のフォルダが自動的に作られる
- ファイル形式は .ino
 - Processingと同様、プログラムのことをスケッチと呼んでいる

電池駆動にしてみよう

- USBケーブルを取り外す
- 電池スナップを使って9V電池をArduinoに接続する
- 電池を接続すると同時に動きだす

② ボタン入力でLEDが点滅する回路



ボタンを押すとLEDが点滅するプログラム

```
void setup() {
```

```
  pinMode( 2, INPUT );
```

2番ピンを「入力」に設定

```
  pinMode( 9, OUTPUT );
```

9番ピンを「出力」に設定

```
}
```

```
void loop() {
```

```
  int sw = digitalRead( 2 );
```

2番ピンの電圧の状態を読み取る

```
  if ( sw == HIGH ) {
```

スイッチが押されていればHIGHが入力される

```
    digitalWrite( 9, HIGH );
```

LEDをONにする

```
    delay(100);
```

100ms待つ

```
    digitalWrite( 9, LOW );
```

LEDをOFFにする

```
    delay(100);
```

100ms待つ

```
}
```

```
}
```

命令の意味

- `digitalRead(ピン番号)`
 - 指定したピンの入力電圧を読む
 - int値 (LOW または HIGH) を返す

※端子に3V以上の電圧がかかっている場合はHIGHと見なされる

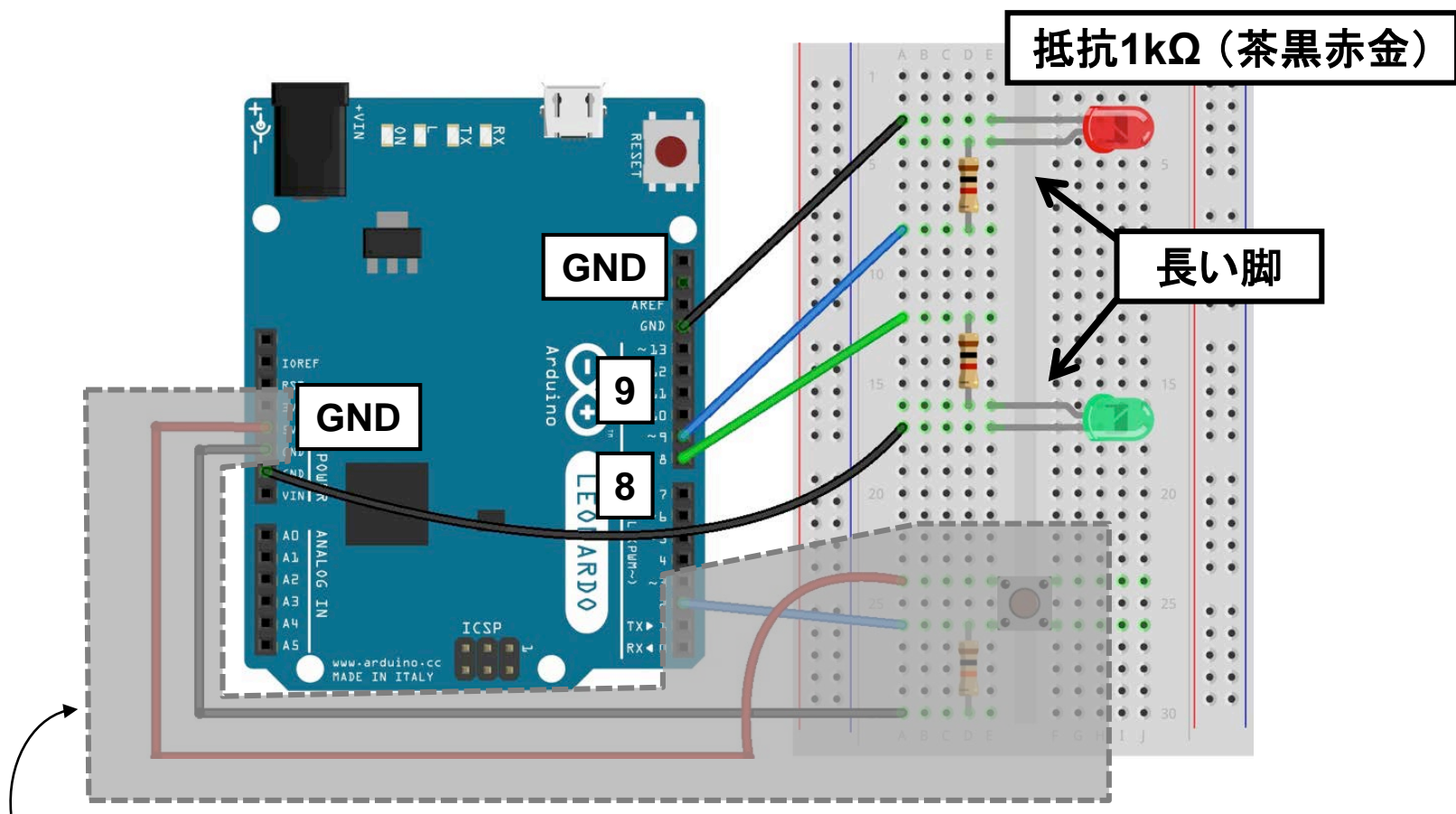
練習問題1

- スイッチを押すたびに、LEDのON/OFFが交互に切り替わるプログラムを作りなさい
- スイッチを3回押したらLEDが光りっぱなしになるプログラムを作りなさい

【ヒント】 状態を保持するフラグ用の変数を作ろう。Arduinoでも論理型 (boolean型) が使える。使い方はProcessingと同じで、true/falseで判定できる。

練習問題2

- 2個のLEDを使った回路を作って、プログラムで交互に点灯させなさい（以下は回路例）



この部分は前の回路の残り(外してしまってもOKだが、スイッチを使うプログラムを書いてみるのもよい) 36

おすすめ教科書 & 参考書

- Arduinoをはじめよう (Massimo Banzi・船田巧、オライリー・ジャパン)
- Prototyping Lab — 「作りながら考える」ためのArduino実践レシピ (小林茂、オライリー・ジャパン)



-
- Processing をインストールしておく