

演習：フィジカル・コンピューティング

11/27 3限

演習1 | ブレッドボードに親しむ

演習2 | センサーの状態を「電圧計」から読み取る

11/27 4限

演習3 | Arduino (センサー)

12/04 3～4限

演習4A | Arduino - Processing

演習4B | Unity - Processing

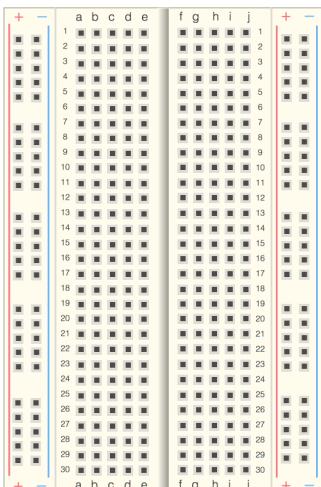
(アプリケーション間通信)

担当：小鷹

ブレッドボードに親しむ

ミニ課題

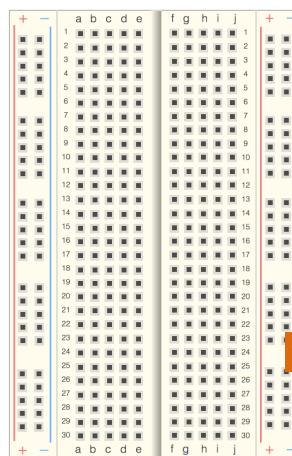
ブレッドボードの通電を調べてみよう。
電気的につながっているところに線を引いてください。



2点間の抵抗が
0になるとどこ
を探す？

演習1 ブレッドボードに親しむ

ブレッドボード

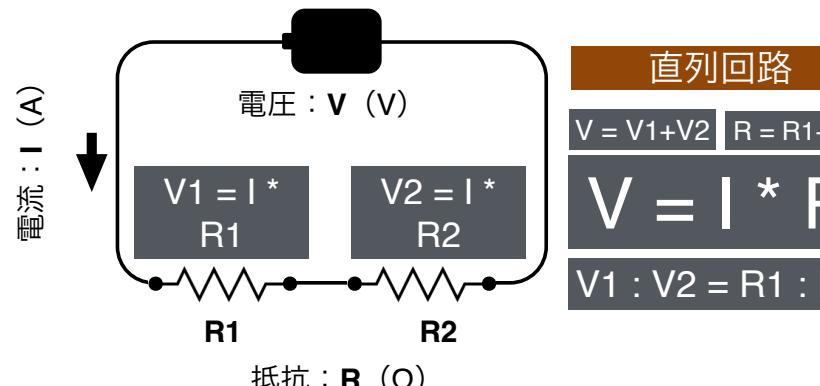


- テスト用の基板を作成するのに最適。
- 実際は、研究制作のあらゆる場面で使用されている。

マルチメーター



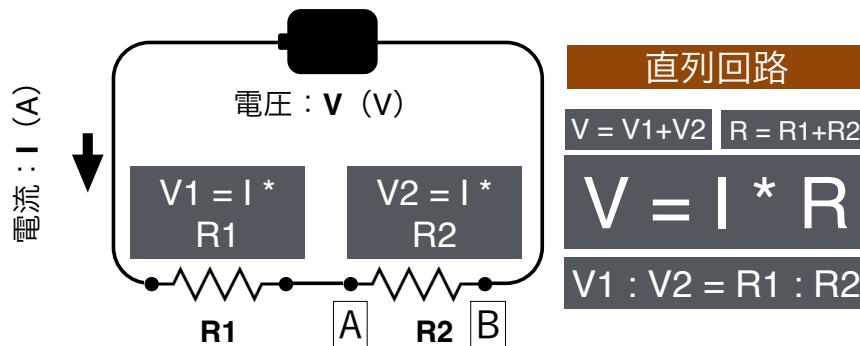
ブレッドボードに親しむ



- 上の回路で、電流は常に一定値をとる。
- 抵抗は仕事を生むところ…抵抗の強い場所で高い電圧が消費される。

ミニ課題

電圧Vと抵抗R1を固定したまま、R2の抵抗のみを変えた時、AB間の電圧・電流値を実際に計測せよ。



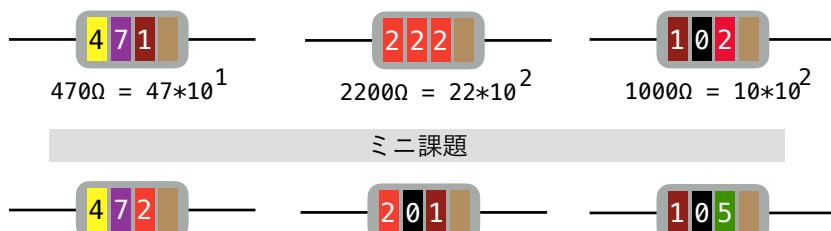
R1	R2	V	V1	V2	I
1000 Ω	470 Ω	V	V	V	mA
	1000 Ω		V	V	mA
	2200 Ω		V	V	mA

抵抗のカラーコード

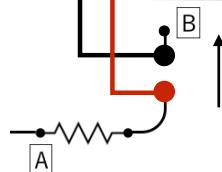
http://www.jarl.org/Japanese/7_Technical/lib1/teikou.htm

数値	色	覚え方	数値	色	覚え方
0	黒	黒い礼 (0) 服	5	緑	五月ミドリ
1	茶	小林 (1) 茶	6	青	徳川無声 (六声)
2	赤	赤いに (2) んじん	7	紫	紫式 (七) 部
3	橙	み (3) かんはダイダイ	8	灰	ハイヤー (8)
4	黄	四季 (黄) の色	9	白	ホワイトク (9) リスマス

抵抗のカラーコードは左の二つが数値を表し左から3つめが乗数を表します。一番右側が許容差を表しています。



ABを通じる電流を測る

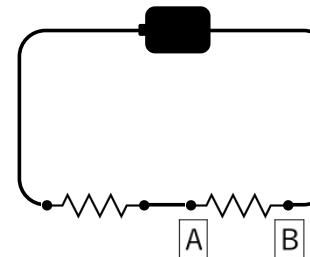


AB間に穴を開けて、電流計を中継させます。

AB間の電圧を測る

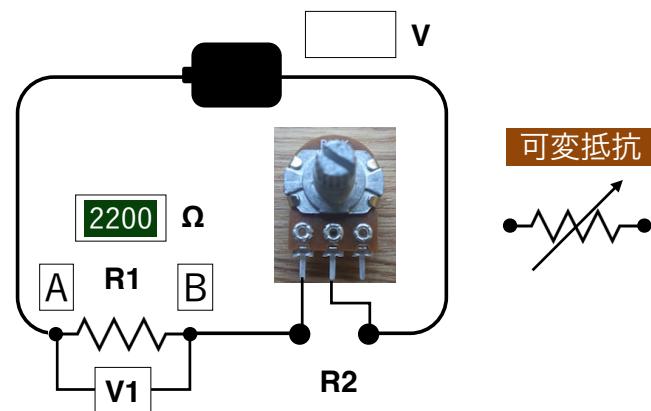


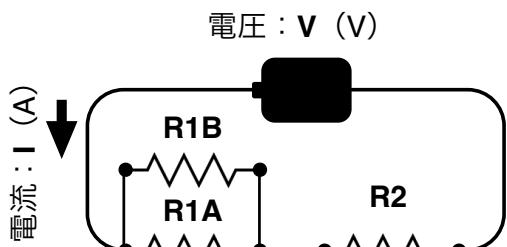
Bを基準としたときのAの電位は、Bをマイナス端子に、Aをプラス端子に接続して計測します。



ミニ課題

抵抗・可変抵抗器を下のように接続し、可変抵抗のつまみをまわした時の、AB間の電圧の変化を観察せよ。さらに、その変化から、可変抵抗の仕様（抵抗がどの範囲で変化するか）を推測せよ。





$$V1 = I * R1 \quad V2 = I * R2$$

並列回路 + (直列)

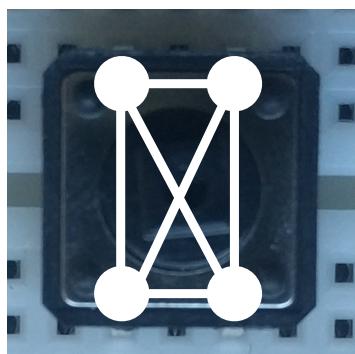
$$\frac{1}{R1A} + \frac{1}{R1B} = \frac{1}{R1}$$

R1A	R1B	R2	V1	V2	V
1000 Ω	470 Ω	2200 Ω	V	V	V
	1000 Ω		V	V	
	2200 Ω		V	V	

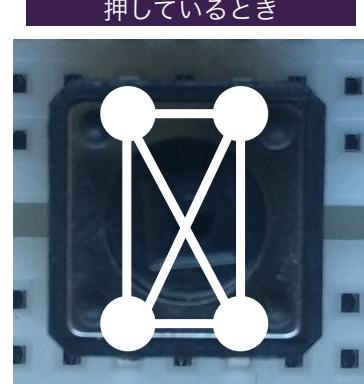
ミニ課題

スイッチを押していない時、スイッチを押している時の導通をチェックせよ。

スイッチを押していない時

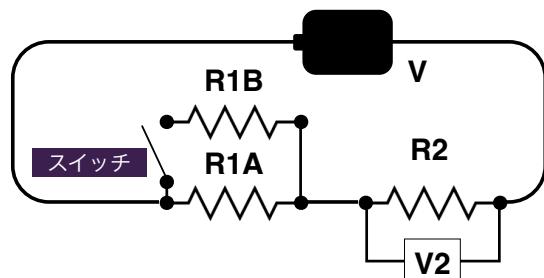


押しているとき



ミニ課題

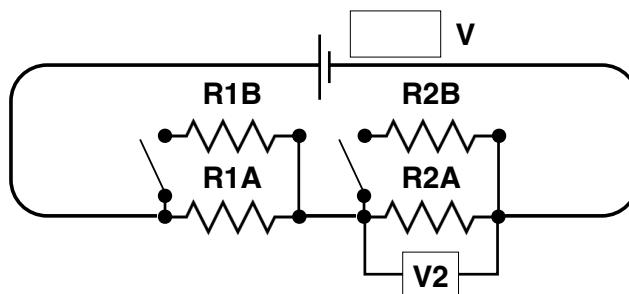
タクトスイッチを使って、下のような回路を構成せよ。さらに、このとき、スイッチのON・OFFによって、V2がどのように変化するかを計測せよ。



R1A	R1B	R2	V2	V
2200 Ω	1000 Ω	2200 Ω	スイッチ ON V	V
			スイッチ OFF V	

ミニ課題

以下のように、二つのスイッチを配置し、スイッチの4つの状態に応じて、V2が違う値を返すように、回路を組んでください。



スイッチ1	スイッチ2	V2
ON	ON	V
ON	OFF	V
OFF	OFF	V
OFF	ON	V

演習：フィジカル・コンピューティング

11/27 3限

演習1 | ブレッドボードに親しむ

演習2 | センサーの状態を「電圧計」から読み取る

11/27 4限

演習3 | Arduino (センサー)

12/04 3～4限

演習4 A | Arduino - Processing

演習4 B | Unity - Processing

(アプリケーション間通信)

担当：小鷹

ミニ課題

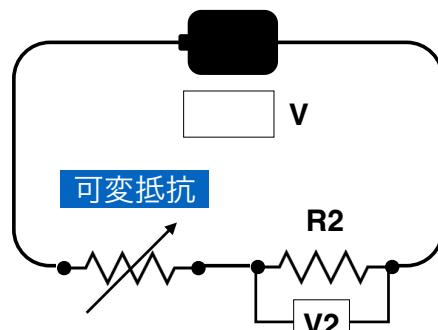
可変抵抗を以下のセンサに差し替えて、センサの状態を変化させた時のV2の値を観測してください。

曲げセンサ

圧力センサ

照度センサ

可変抵抗



傾きセンサ



可変抵抗タイプのセンサ (無電源)



曲げセンサ

圧力センサ

位置センサ

照度センサ



傾きセンサ



センサの状態や周辺の環境が変わることで、抵抗値が変わる。

<http://www.switch-science.com/catalog/508/>

<http://www.switch-science.com/catalog/207/>

<http://www.switch-science.com/catalog/270/>

<http://akizukidensi.com/catalog/g/gi-00110/>

曲げセンサ



曲げるとV2は、

上昇する／下降する

つまり、曲げると曲げセンサの抵抗は、

上昇する／下降する

圧力センサ

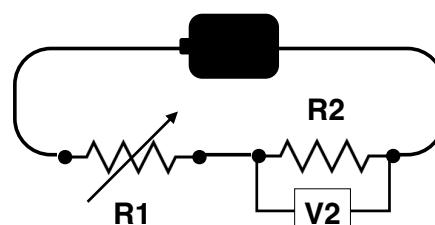


押すとV2は、

上昇する／下降する

つまり、押すと圧力センサの抵抗は、

上昇する／下降する



照度センサ



暗くするとV2は, 上昇する／下降する

つまり, 暗くすると照度センサの抵抗は,

上昇する／下降する

傾きセンサ



傾けるとV2は, 上昇する／下降する

つまり, 傾けると傾きセンサの抵抗は,

上昇する／下降する

可変抵抗



時計回りに回すとV2は, 上昇する／下降する

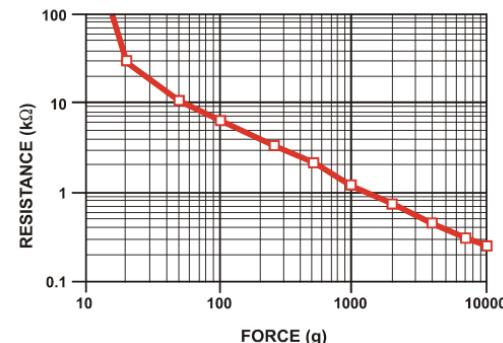
つまり, 回すと可変抵抗の値は,

上昇する／下降する

圧力センサ



圧力センサの仕様



<https://www.sparkfun.com/datasheets/Sensors/Pressure/fsrguide.pdf>

- 押していない場合の抵抗値が100kΩ超.
- 1000gの力で, 抵抗値が1kΩ程度.

D4後半 - フィジカルXR 2024

演習：フィジカル・コンピューティング

11/27 3限

演習1 | ブレッドボードに親しむ

演習2 | センサーの状態を「電圧計」から読み取る

11/27 4限

演習3 | Arduino (センサー)

12/04 3～4限

演習4 A | Arduino - Processing

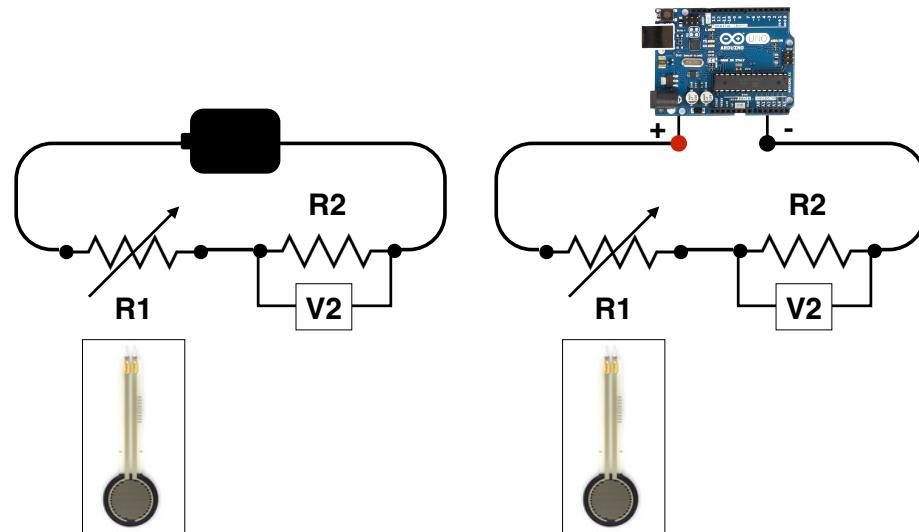
演習4 B | Unity - Processing

(アプリケーション間通信)

担当：小鷹

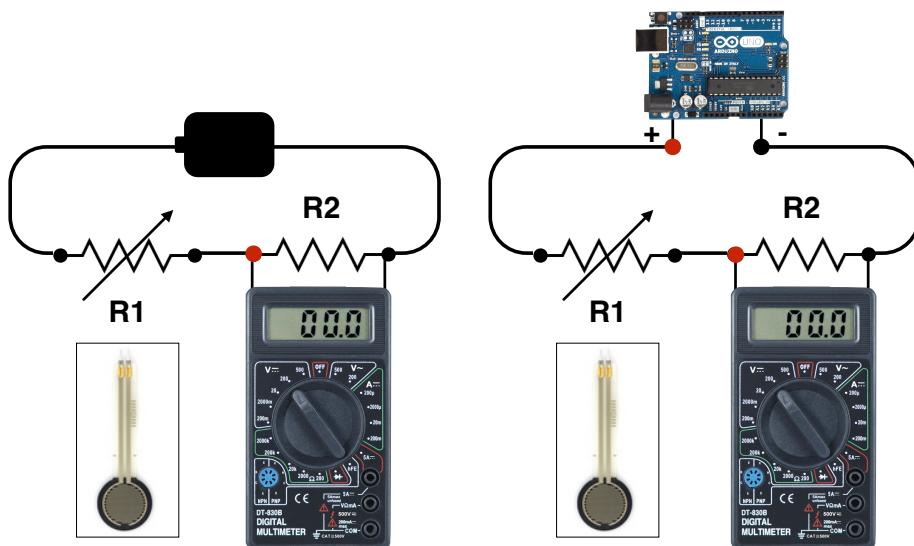
PHYSICAL XR 2023

V2の電圧をマルチメータでなくArduinoに取り込む

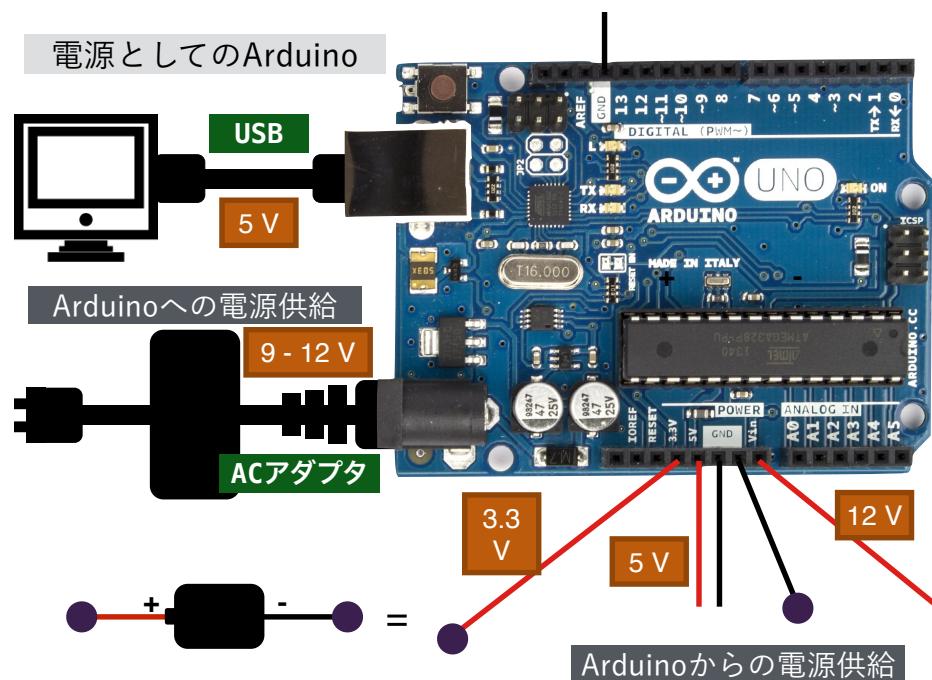


PHYSICAL XR 2023

電圧を電池ではなく, Arduinoから供給してみよう.

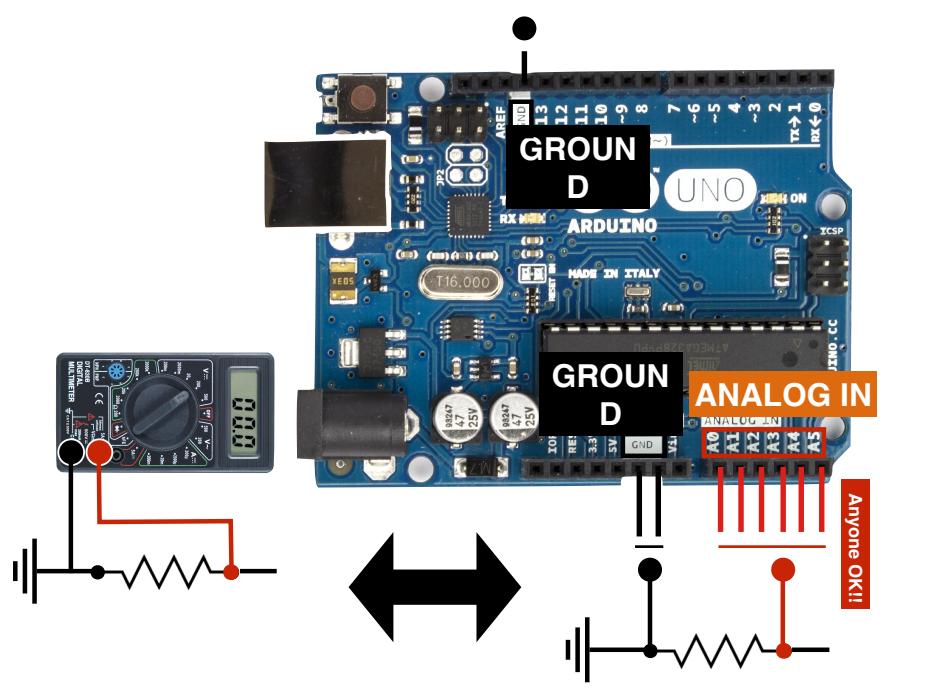
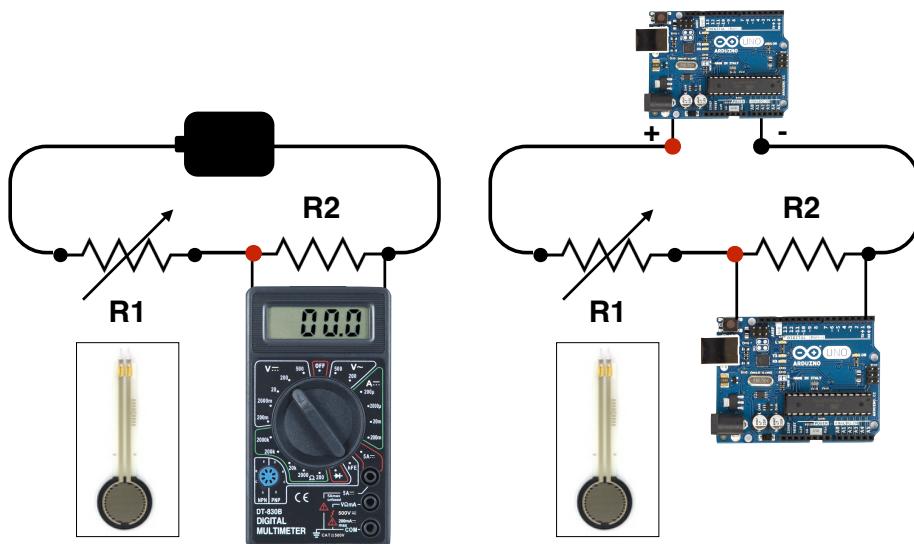


電源としてのArduino



PHYSICAL XR 2023

V2の電圧をマルチメータでなくArduinoに取り込む



Arduinoのインストール

<https://www.arduino.cc/en/software/>

Bring Your Projects to Life with Arduino Software

Arduino IDE 2.3.6
Release notes

The new major release of the Arduino IDE is faster and even more powerful! In addition to a more modern editor and a more responsive interface it features autocompletion, code navigation, and even a live debugger. For more details, check the [Arduino IDE 2.0 documentation](#).

macOS Intel 10.15 Catalina or newer (64-bit) DOWNLOAD

Nightly Builds
Download a preview of the incoming release with the most updated features and bugfixes.

The Arduino IDE 2.0 is open source and its source code is hosted on [GitHub](#).

Arduinoエディタの編集

Arduinoに書き込みを行う。 Arduino Uno

sketch_sensor1.ino グローバル変数の宣言 sketch_sensor1.ino

1 int sensorPin = A0;
2 int sensorValue = 0;
3 float voltValue = 0.0;
4 float maxVolt = 5.0;

センサーピンの番号 (A0-A5)
センサーの値を格納する変数
電圧の値を格納する変数
最大電圧 (5Vにピンを指して計測)

起動時・一回きりの処理 (Processingのsetup()と同じ)

7 void setup() {
8 Serial.begin(19200);
9 pinMode(sensorPin, INPUT);
10 }

Arduino内で数値をモニタするために、シリアル通信を起動。
A0を入力ピンとして設定します。

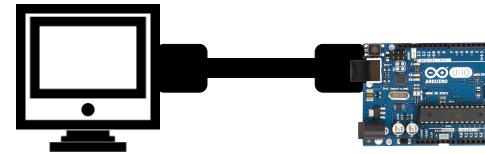
setup関数実行後 に繰り返し行われる処理 (60Hz, Processingのdraw()と同じ)

12
13 void loop() {
14 sensorValue = analogRead(sensorPin);
15 voltValue = maxVolt * sensorValue/1023.0; 電圧に変換
16 Serial.println(sensorValue);
17 Serial.println(voltValue);
18 delay(100);
19 }

センサーの値の読み込み (0-1023).
シリアルモニタに、変数の値をプリント
100ms休む。

Arduino と PC の接続

1. PCとArduinoをUSBケーブルで接続



2. Arduinoを開く。

3. ツールでマイコンボード (Arduino Uno) とシリアルポートを設定

ツール ヘルプ(H)

自動整形
スケッチをアーカイブする
ライブラリを管理...
シリアルモニタ
シリアルプロト

Firmware Updater
SSLルート証明書を書き込み

ボード: "Arduino Uno"
ポート: "/dev/cu.usbmodem1301"
ポート情報を取得

書き込み装置: "Arduino as ISP"
ポート: "/dev/cu.usbmodem1301" /dev/cu.usbmodem1301
ポート情報を取得

浮点数表示: 1.2345
浮点数表示: 1.2345
浮点数表示: 1.2345
浮点数表示: 1.2345

2

3

ツール ヘルプ(H)

自動整形
スケッチをアーカイブする
ライブラリを管理...
シリアルモニタ
シリアルプロト

Firmware Updater
SSLルート証明書を書き込み

ボード: "Arduino Uno"
ポート: "/dev/cu.usbmodem2101"
ポート情報を取得

書き込み装置: "Arduino as ISP"
ポート: "/dev/cu.usbmodem2101 (Arduino Uno)"
ポート情報を取得

シリアルモニタの使い方

シリアルモニタ
を立ち上げる。

Arduino Uno

sketch_dec6a.ino

出力 シリアルモニタ x

メッセージ ('/dev/cu.usbmodem2101'のArduino Unoにメッセージ LFのみ 19200 baud)

3.91
797
3.90 14 Serial.println(sensorValue);
15 Serial.println(voltValue);
789
3.86
784
3.83
783
3.83
783
3.83
784
3.83

シリアル.println関数の引数が表示されます。

bpsを、Serial.begin関数の引数と合わせる。

6 void setup() {
7 Serial.begin(19200);
8 pinMode(sensorPin, INPUT);
9 }

行 17、列 3 Arduino Uno /dev/cu.usbmodem2101の 2 3

シリアルプロッタの使い方

シリアルプロッタを立ち上げる。



```
17 //Serial.println(sensorValue);
18 Serial.println(voltValue);
19 delay(100);
```

一方をコメントアウトすると



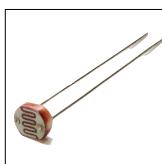
電圧変化を時系列データとして可視化できる。

```
6 void setup() {
7 Serial.begin(19200);
8 pinMode(sensorPin, INPUT);
9 }
```

bpsを, Serial.begin関数の引数と合わせる。

練習

照度センサ



傾きセンサ



タクトスイッチ



圧力センサの例を参考にして、

- 暗くなった時（照度センサ）
- 振った時（傾きセンサ）
- ボタンを押した時（タクトスイッチ）

にLEDが光るようにコードを書いてください。

押すとLEDが光るインタラクションを加えます。

```
1 int sensorPin = A0;
2 int sensorValue = 0;
3 float voltValue = 0.0;
4 float maxVolt = 4.55;
5 int ledPin = 13;
```

```
7 void setup() {
8 Serial.begin(19200);
9 pinMode(sensorPin, INPUT);
10 pinMode(ledPin, OUTPUT);
11 }
```

```
13 void loop() {
14 sensorValue = analogRead(sensorPin);
15 voltValue = 4.55*sensorValue/1023.0;
16 Serial.println(sensorValue);
17 Serial.println(voltValue);
18 delay(50);
19
20 if(voltValue>2.0){
21 digitalWrite(ledPin,HIGH);
22 }else{
23 digitalWrite(ledPin,LOW);
24 }
25
26 }
```

sketch_sensor2.ino

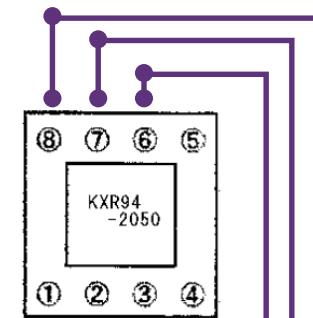


LEDの長い足（+）を13に短い足をGND（-）に指してください。

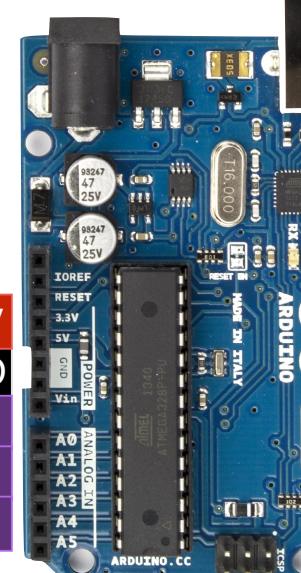


赤で囲った部分を新たに書き加えてください。

電源で駆動するセンサ・モジュール



加速度センサ・モジュール



<http://akizukidenshi.com/catalog/g/gM-05153/>

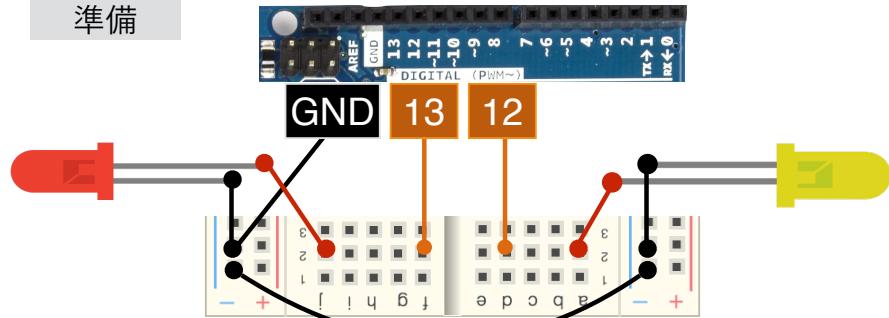
加速度センサの3つの出力をモニタしましょう。

```
sketch_sensor_acc.h
1 int sensorPinX = A0;
2 int sensorPinY= A1;
3 int sensorPinZ = A2;
4
5 int sensorValue = 0;
6 float voltValueX = 0.0;
7 float voltValueY = 0.0;
8 float voltValueZ = 0.0;
9
10 float maxVolt = 4.55;
11
12 int ledPinL = 13;
13 int ledPinR = 12;
```

sketch_sensor_acc.ino

```
15 void setup() {
16   Serial.begin(19200);
17   pinMode(sensorPinX, INPUT);
18   pinMode(sensorPinY, INPUT);
19   pinMode(sensorPinZ, INPUT);
20
21   pinMode(ledPinL, OUTPUT);
22   pinMode(ledPinR, OUTPUT);
23 }
24
25 void loop() {
26   sensorValue = analogRead(sensorPinX);
27   voltValueX = maxVolt*sensorValue/1023.0;
28   sensorValue = analogRead(sensorPinY);
29   voltValueY= maxVolt*sensorValue/1023.0;
30   sensorValue = analogRead(sensorPinZ);
31   voltValueZ = maxVolt*sensorValue/1023.0;
32   Serial.println();
33   Serial.print("X:"); Serial.println(voltValueX);
34   Serial.print("Y:"); Serial.println(voltValueY);
35   Serial.print("Z:"); Serial.println(voltValueZ);
36   delay(500);
37
38   .....
39   .....
40 }
```

準備



課題

下図のように、傾けた側のLEDが光るように、
コードを追加してください。

