

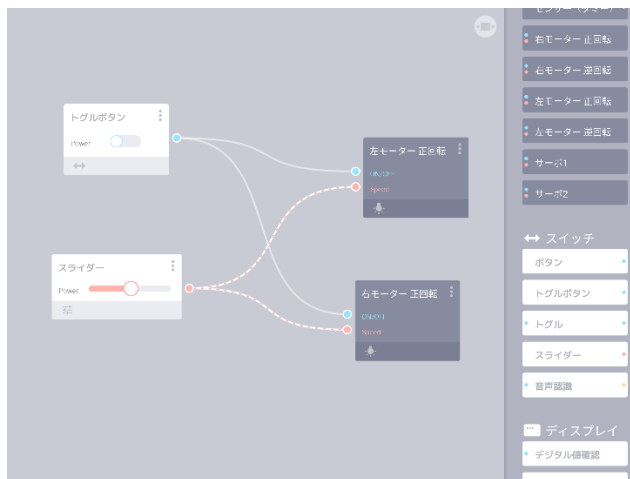
2017/9/12 VIVIWARE BRANCH 作成 Hands-on

0 はじめに…

0.1 VIVIWARE の世界を理解する

・ VIVIPROGRAMMER

Android タブレット上のビジュアルプログラミングアプリ。CORE/BRANCH からデータを受信したり、反対に送信できたりする。



・ VIVIWARE CORE

BLE(無線)ハードウェア。BRANCH（後述）から得た情報を BLE で Android タブレットに送信する、またはタブレットから得た情報を BRANCH へ送る。ただの無線ブロック。BRANCH とはケーブルで接続。

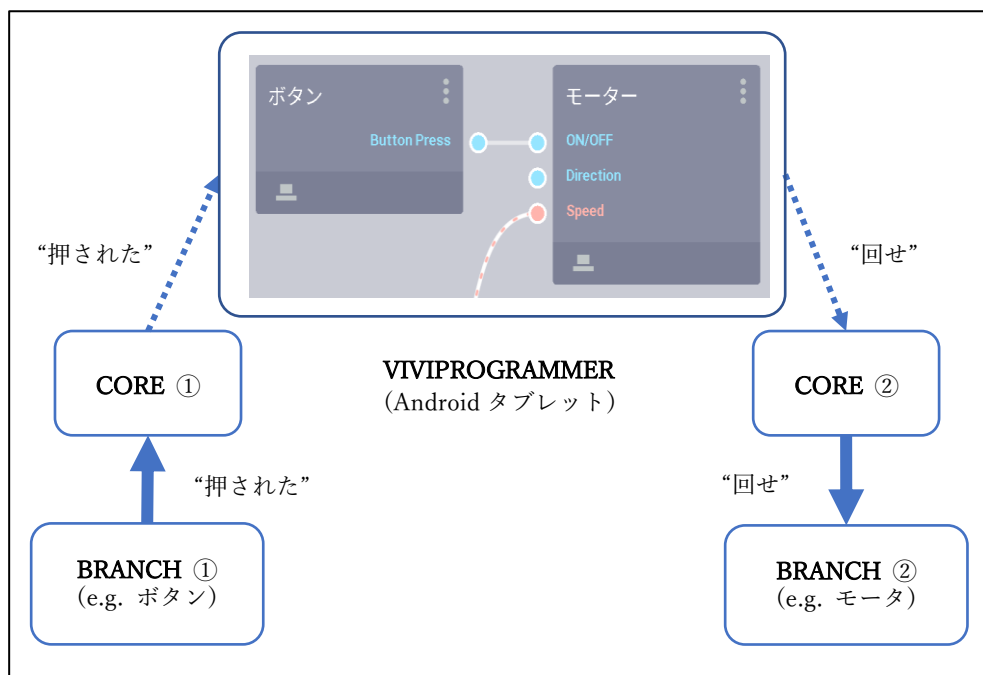


・ VIVIWARE BRANCH

センサーなどのハードウェアから情報を取得して CORE (=タブレット) に送る、または CORE (=タブレット) から受信したデータを使ってモータを動かしたり LED を点灯させたりする機能ブロック。CORE とはケーブルで接続。



・ VIVIWARE 全体のつながり



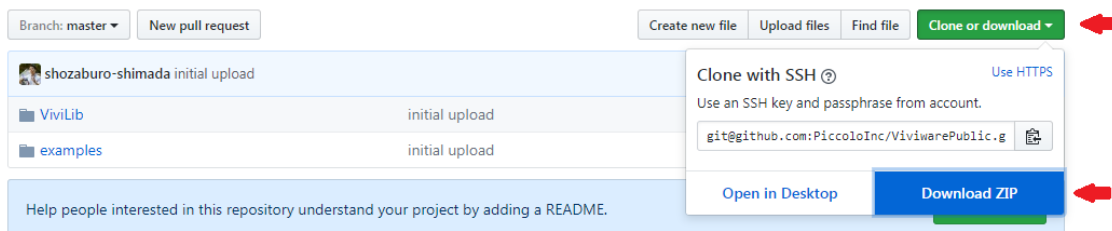
今回はこの図にあるように VIVIWARE 上でコントロールできるハードウェアである BRANCH を作成する。BRANCH の作成には手持ちの Arduino を利用する。作成後はその BRANCH を使って VIVIWARE で遊ぶ。

0.2 （作成前の準備 1）VIVIWARE ライブラリを Arduino にインストールする

- ・ライブラリのダウンロード

下記リンクの“Clone and download”→”Download ZIP”をクリック

<https://github.com/PiccoloInc/ViviwarePublic>



- ・ダウンロードできたら zip を展開する。展開した中に ViviLib というフォルダがあることを確認する。

- ・(Mac の場合) Documents/Arduino/libraries の中に ViviLib というフォルダをフォルダごと入れる

- ・(Windows の場合) C:\Users\<user name>\Documents\Arduino\libraries の中に ViviLib というフォルダをフォルダごと入れる

0.3 （作成前の準備 2）Arduino UNO に VIVIWARE シールドを挿す



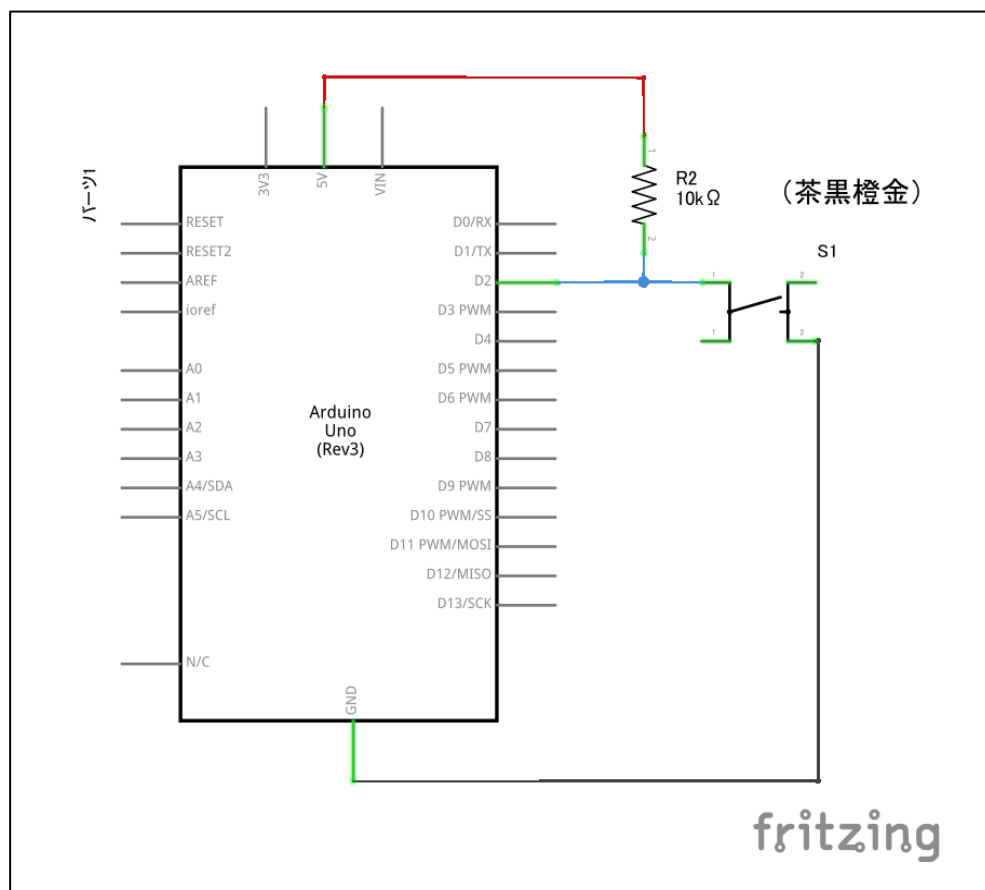
1 ボタンを作る

1.1 ボタン 回路図

用意するもの

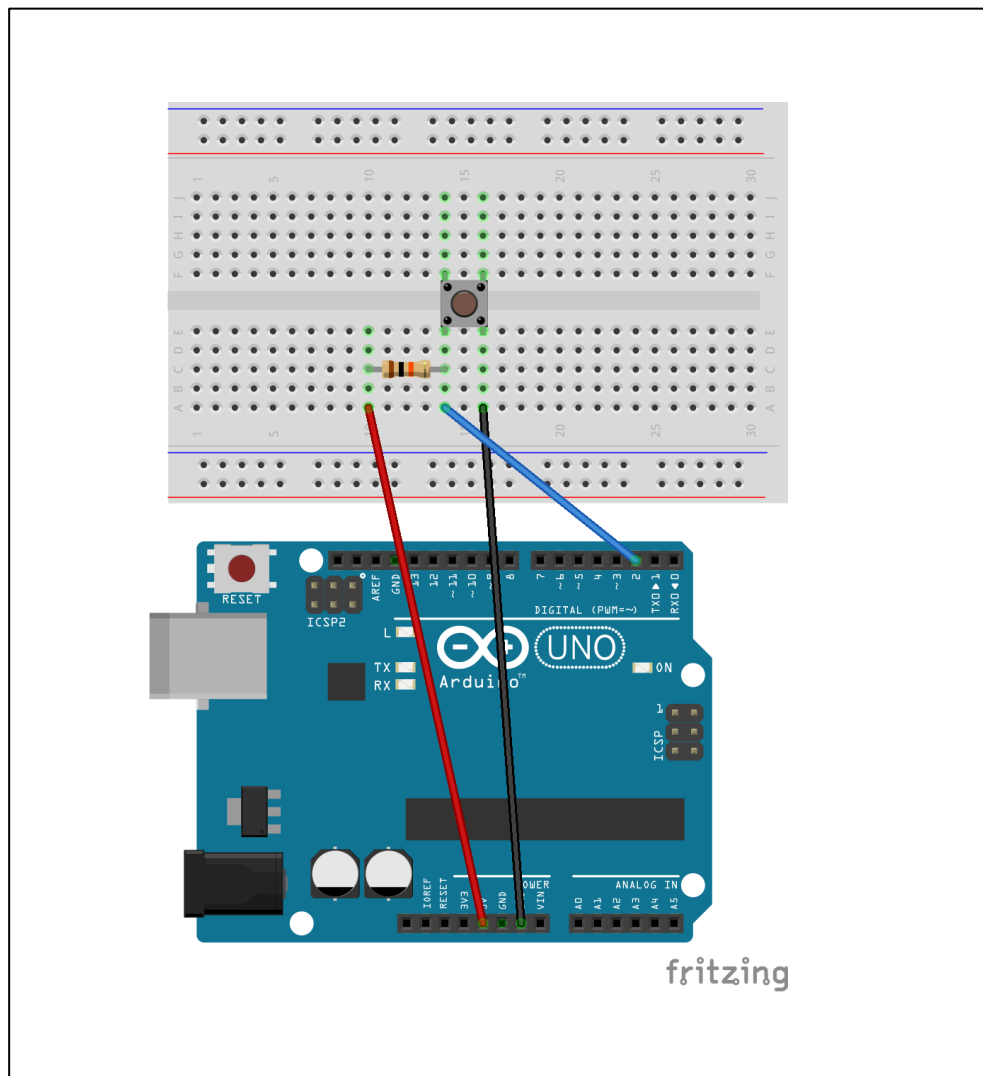
- ・タクトスイッチ
- ・抵抗 10k Ω （茶黒橙金）

この回路では、通常時（ボタン非押下時）は Arduino の D2 ピンが High（5V）、
ボタン押下時は D2 ピンが Low（0V）になる



1.2 ボタン配線図

以下のように配線してください。



1.3 Arduino ソースコード

コードを見ながら自分の Arduino IDE に書き写してください。できたら PC と Arduino を USB ケーブルで接続して”upload”してください。

Viviware_Button.ino

```
#include <ViviLib.h>

unsigned char data[2];
char idPacket[] = {0x00, 0x02};
char fw_version = 0x01;
const int buttonPin = 2;
int preValue = 0;

ViviLib vivilib(fw_version, idPacket, 0);

void setup() {
  //シリアル通信開始（デバッグ用）
  Serial.begin(9600);
  Serial.println("Hello");

  //ライブラリの初期化
  vivilib.init();
  data[0] = 0x01;
  data[1] = 0x01;

  //D2 ピンを入力に設定
  pinMode(buttonPin, INPUT);
}

void loop() {
  //VIVIWARE CORE と接続されているか？
  if (vivilib.connected()) {

    //ボタン（D2 ピン）の状態をチェックする
    int curValue = (digitalRead(buttonPin)) ? 0 : 1;

    //前回からボタンの状態が変更されているか？
    if (curValue != preValue) {
      Serial.print("Send: ");
      Serial.println(curValue);

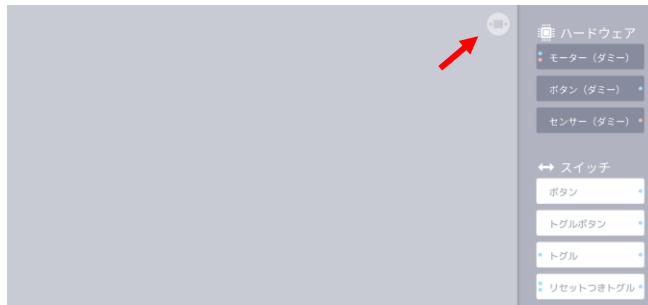
      //ボタンの状態を VIVIWARE CORE に送信
      data[1] = curValue;
      vivilib.sendData(data, 2);

      //今回のボタンの状態を記録しておく
      preValue = curValue;
    }
  }
  delay(50);
}
```

1.4 動作確認する

手順1：タブレットで VIVIPROGRAMMER を起動

手順2：NewProject か既存の Project を選択してプログラミング画面を表示、右上のアイコンをタッチしモジュール一覧を表示した状態で待機する。



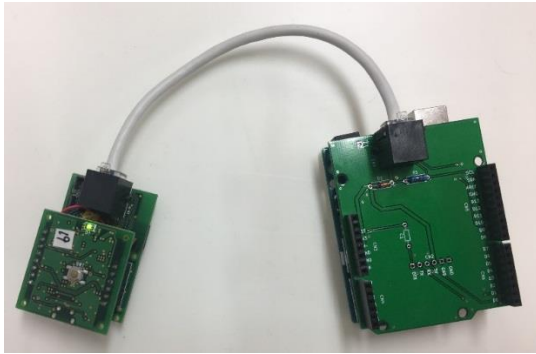
手順3：CORE の電源をいれる（CORE のボタンを長押し、LED が点灯）

手順4：NFC タグをタブレットにかざして CORE をペアリング。ブランクモジュールが表示されていればペアリング成功。



手順5；CORE と作成した BRANCH(Arduino UNO)をケーブルで接続

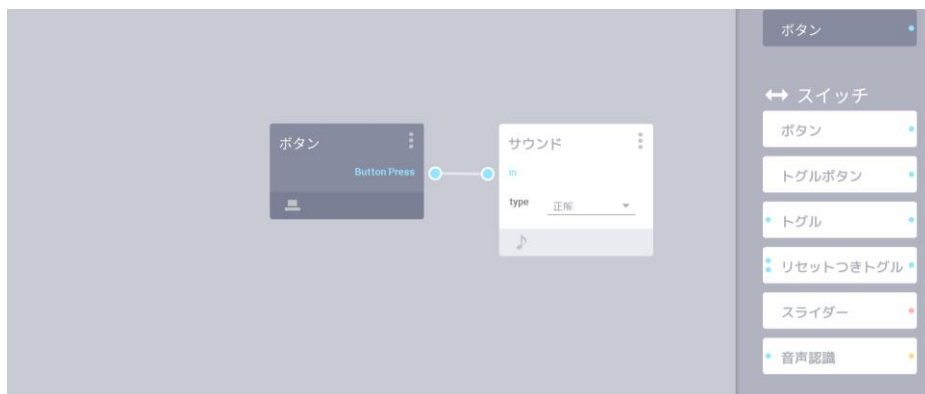
(注：接続前に、PC と接続している USB ケーブルを抜いておく)



手順6：VIVIPROGRAMMER 右のモジュール一覧に表示されたら成功



手順7；表示されたボタンモジュールとサウンドを接続、ボタンを押すと音が鳴ることを確認する。



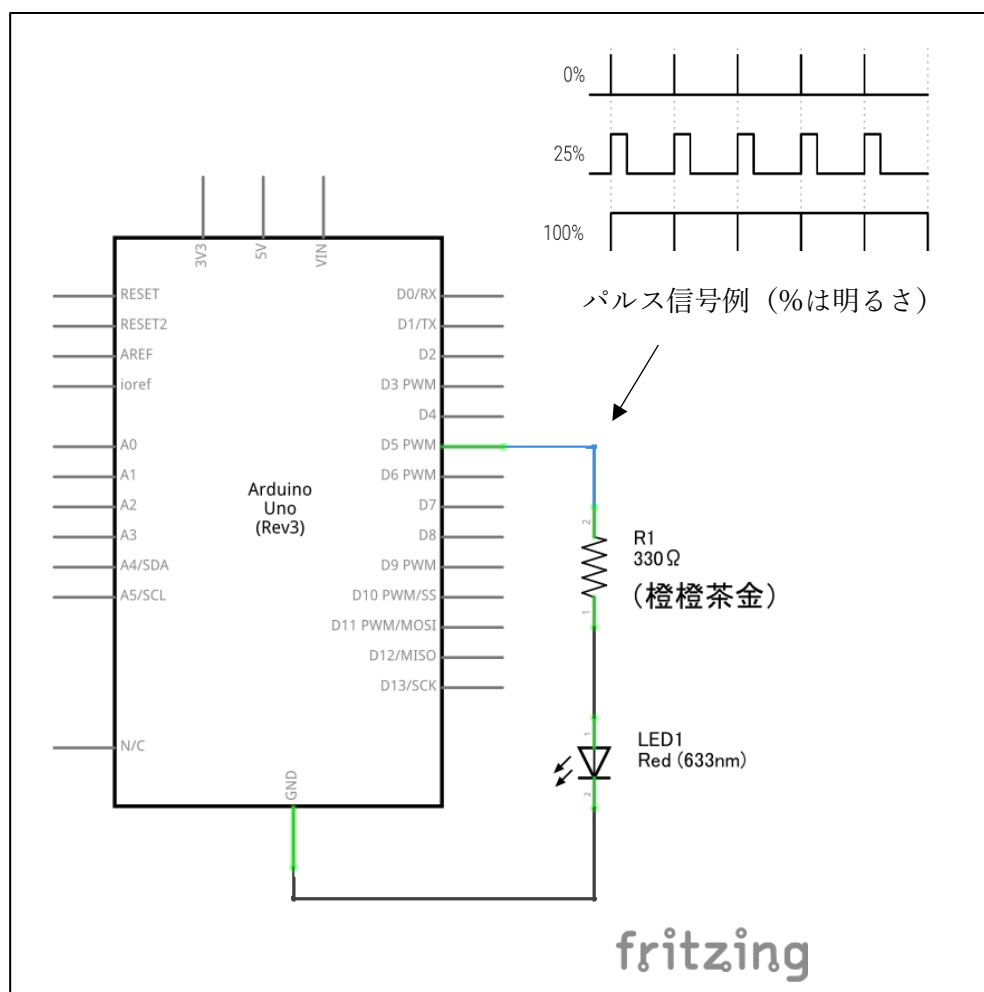
2 LED をつくる

2.1 LED 回路図

用意するもの

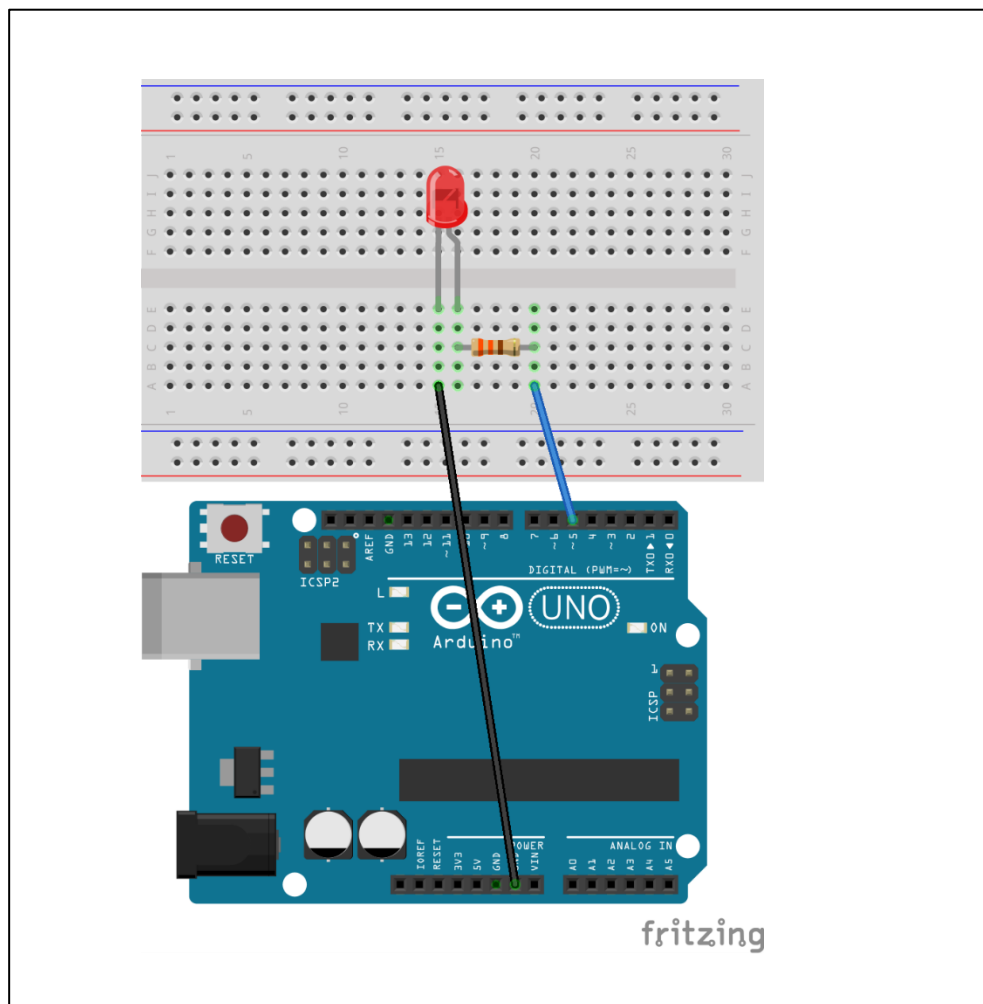
- ・抵抗 330Ω (橙橙茶金)
- ・LED (どれでもよい)

D5 ピンを High(5V)にすると LED が点灯、Low (0V) にすると LED が消灯するが、今回は LED の明るさも変更できるようにしたいので、D5 からはパルス状の信号を出力させる (2.3 Arduino ソースコード参照)



2.2 LED 配線図

LED には向き（足の長さで判断）があるので配線時は注意



2.3 Arduino ソースコード

以下のリンクからコピー

https://github.com/PiccoloInc/ViviwarePublic/blob/master/examples/Viviware_Led/Viviware_Led.ino

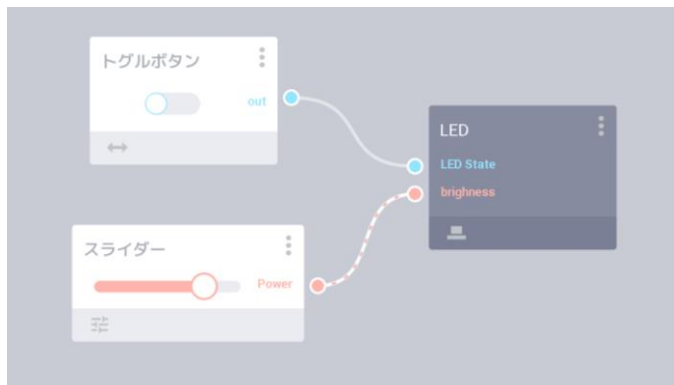
2.4 動作確認

手順1：1.4 の手順を参考にモジュールを表示させるところまで進める

（注：1.4 の手順4 に関しては既にペアリング済みであれば（＝ブランクモジュールが表示済み）であれば、スキップしてください）



手順2：表示された LED モジュールとスライダー、トグルボタンを組み合わせ、LED が点灯・消灯できるか確認



3 感圧センサーをつくる

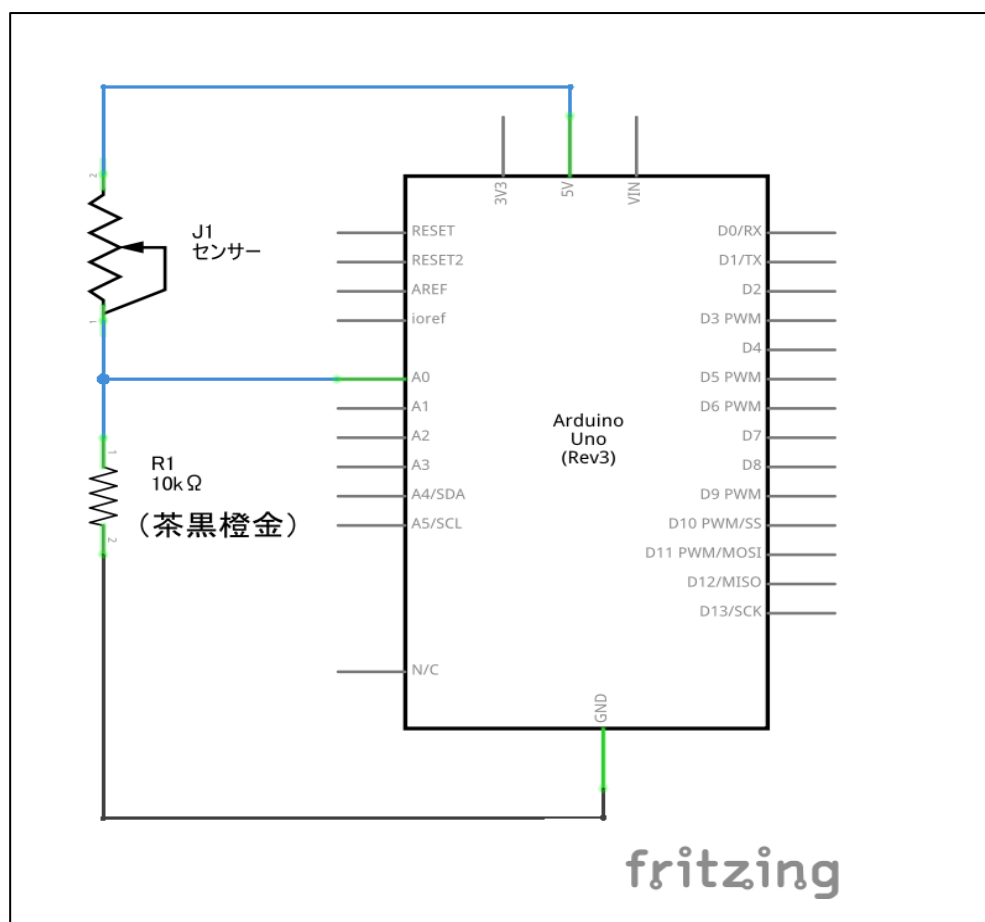
3.1 感圧センサー回路図

用意するもの

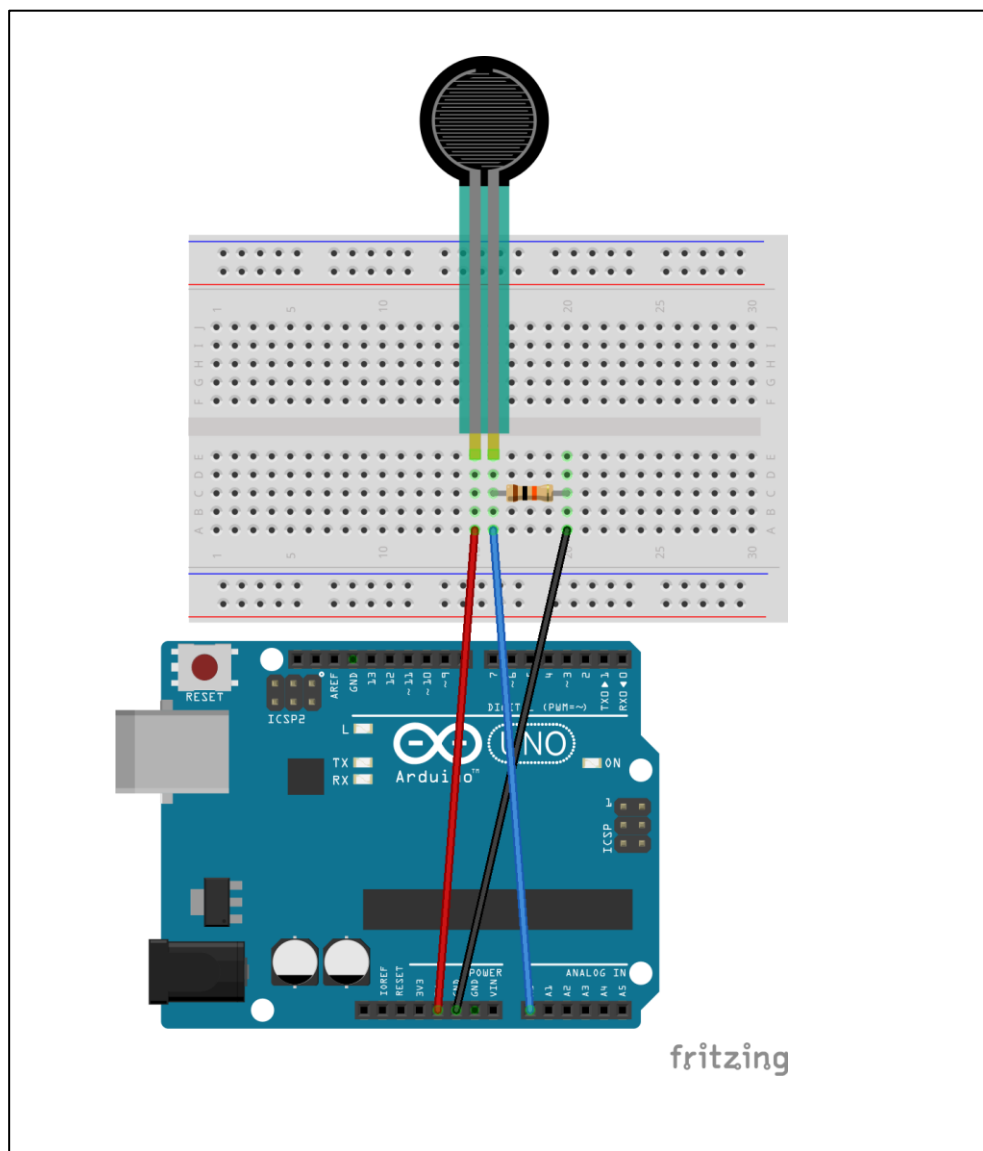
- ・ 感圧センサー (ALPHA-MF02-N-221-A01)
- ・ 抵抗 10k Ω (茶黒橙金)

感圧センサーは、加えられた圧力によってその抵抗値が変化する

下の回路ではセンサーの抵抗値が変化すると A0 ピンの電圧が変化するように
なっている、その電圧値の変化を見ることで圧力の変化を検出する



3.2 感圧センサー配線図



3.3 Arduino ソースコード

以下のリンクからコピペ

https://github.com/PiccoloInc/ViviwarePublic/blob/master/examples/Viviware_Pressure/Viviware_Pressure.ino

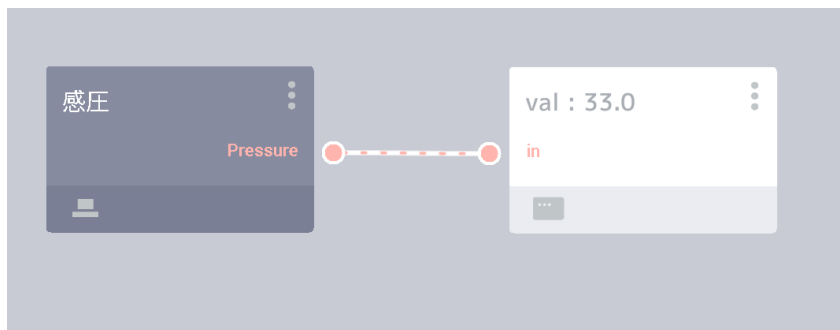
3.4 動作確認

手順1：1.4 の手順を参考にモジュールを表示させるところまで進める

（注：1.4 の手順4 に関しては既にペアリング済みであれば（＝ブランクモジュールが表示済み）であれば、スキップしてください）



手順2：表示された感圧センサーモジュールとアナログ確認モジュールを接続して、センサーを押しながら値が変化していることを確認する



4 Cds（照度）センサーをつくる

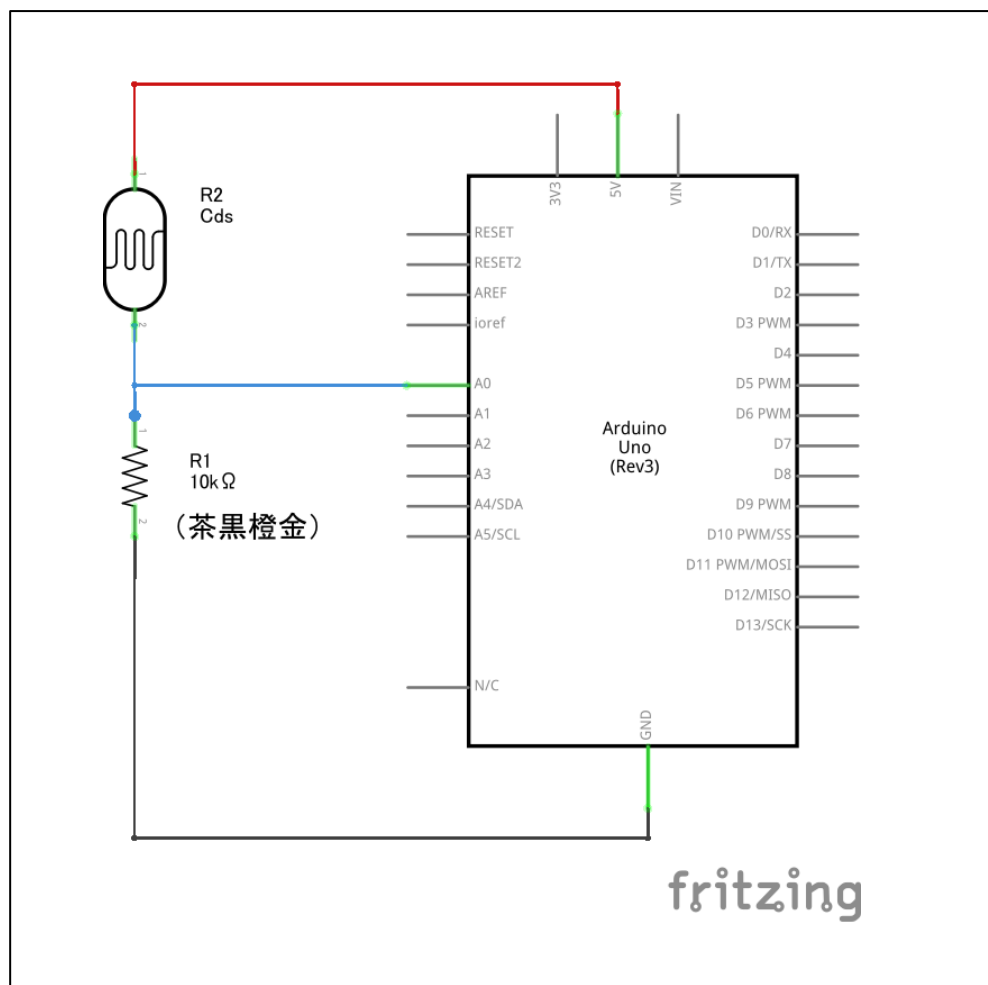
4.1 Cds センサー回路図

用意するもの

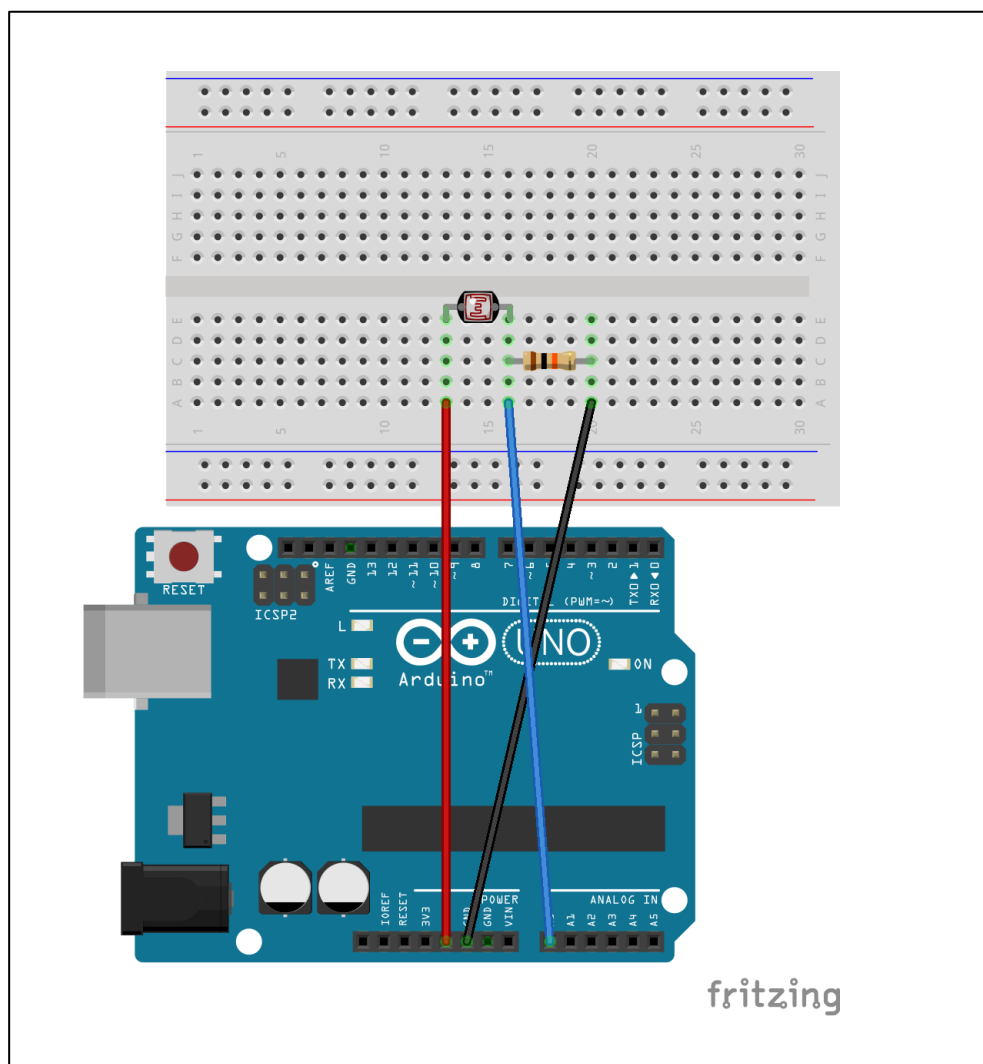
- ・ Cds センサー
- ・ 抵抗 10k Ω （茶黒橙金）

Cds センサーは、当てられた光の明るさでその抵抗値が変化する

従って回路やその検出方法は 4.1 の圧力センサーと同じ



4.2 Cds センサー配線図



4.3 Arduino ソースコード

以下のリンクからコピー

https://github.com/PiccoloInc/ViviwarePublic/blob/master/examples/Viviware_Cds/Viviware_Cds.ino

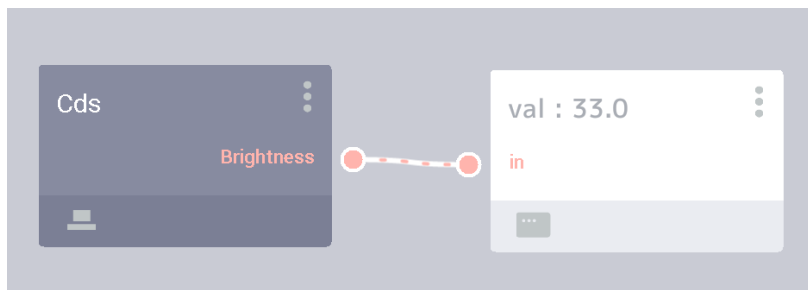
4.4 動作確認

手順1 : 1.4 の手順を参考にモジュールを表示させるところまで進める

(注 : 1.4 の手順4 に関しては既にペアリング済みであれば (= ブランクモジュールが表示済み) であれば、スキップしてください)



手順2 : 表示された Cds センサーモジュールとアナログ確認モジュールを接続して、Cds センサーを手で覆いながら値が変化していることを確認する



5 加速度（今回は傾き）センサーをつくる

5.1 加速度センサー回路図

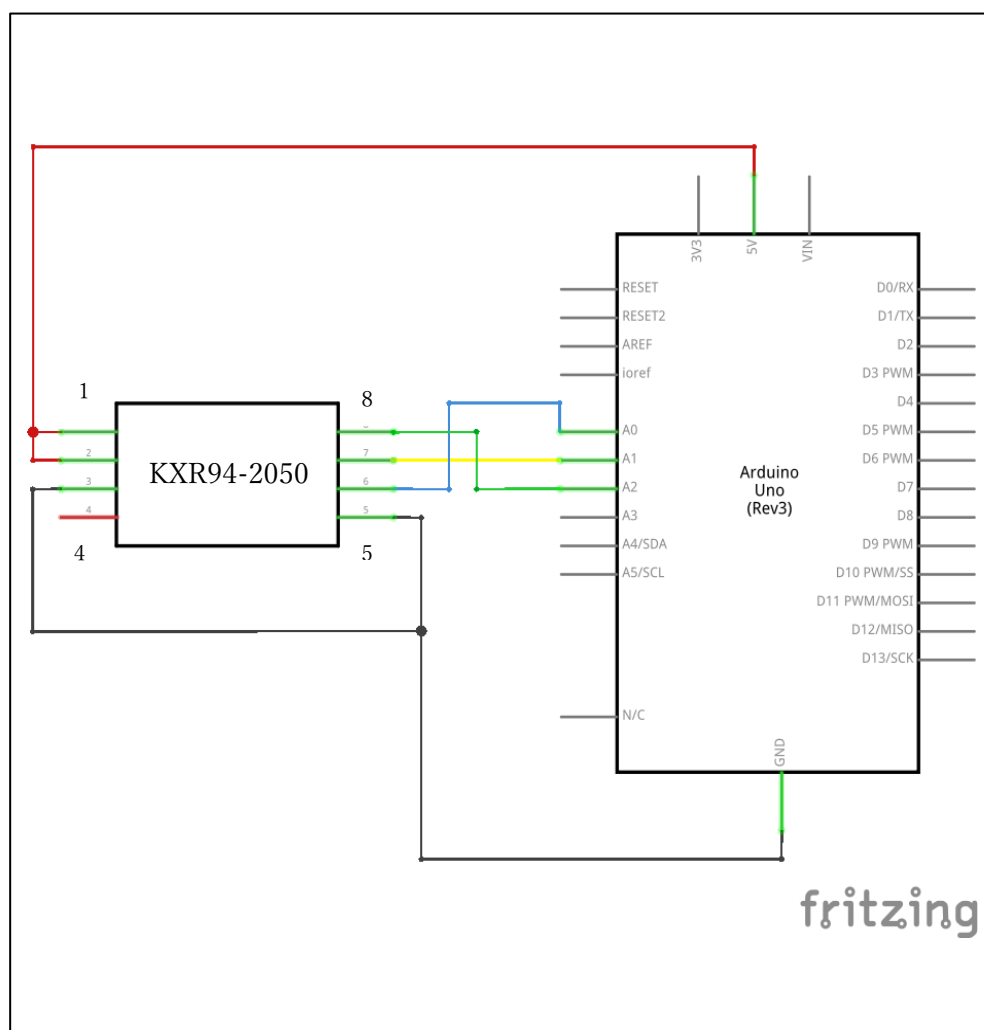
用意するもの

- ・ 加速度センサー（KXR94-2050）

加速度センサーは主に動きを検出する

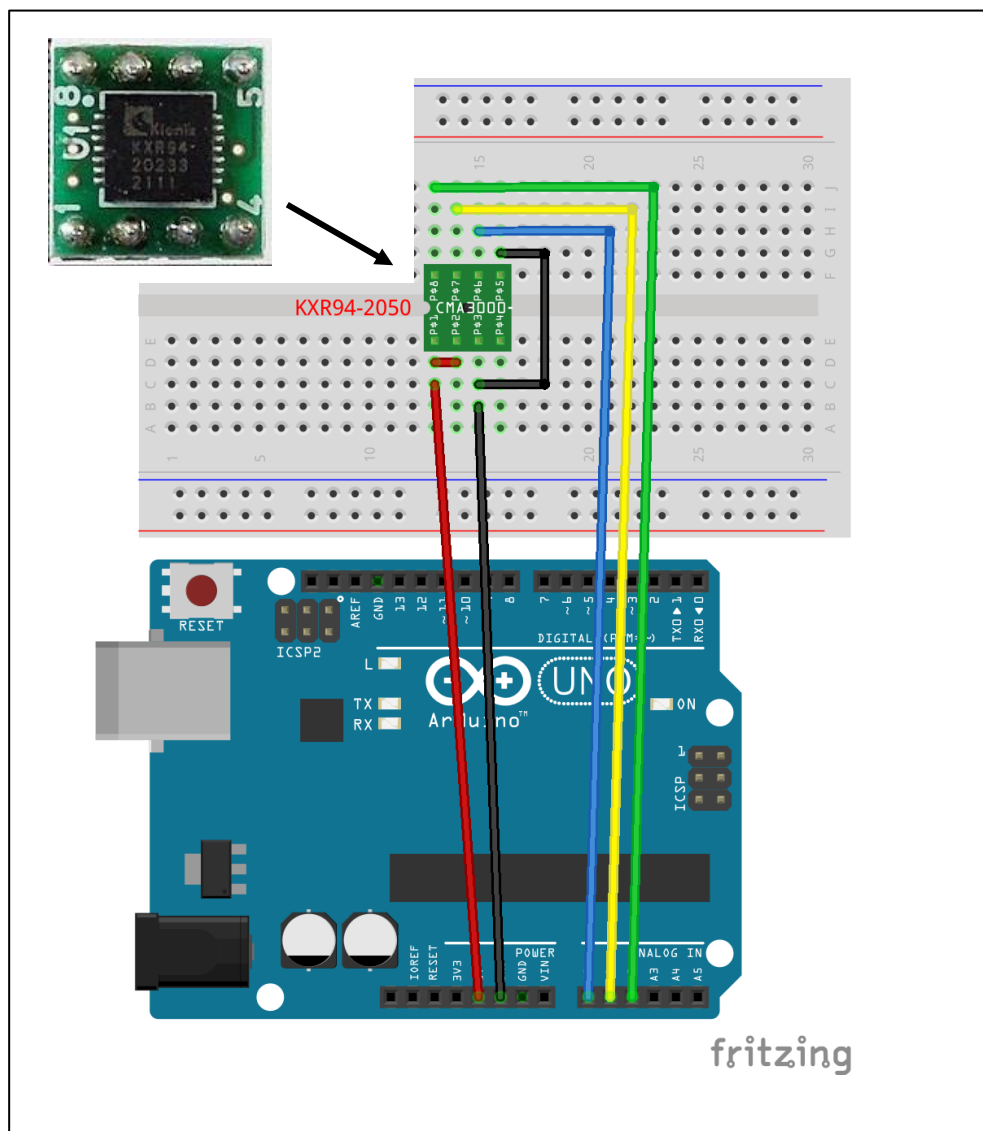
ただしセンサーから取得した値はそのまま使えないことが多い

今回は取得した生データを「傾き」に加工し傾きセンサーとして利用する



5.2 加速度センサー配線図

KXR94-2050 に書いてある 1、4、5、8 の数字が下図のようになるように配置



5.3 Arduino ソースコード

以下のリンクからコピー

https://github.com/PiccoloInc/ViviwarePublic/blob/master/examples/Viviware_Accelerometer/Viviware_Accelerometer.ino

5.4 動作確認

手順1：1.4 の手順を参考にモジュールを表示させるところまで進める

（注：1.4 の手順4 に関しては既にペアリング済みであれば（＝ブランクモジュールが表示済み）であれば、スキップしてください）



手順2：表示された加速度モジュールとアナログ確認モジュールを3つ接続して、加速度センサーを色々な方向に傾けながら値が変化していることを確認する
値は水平面（x-y 平面）に対しての、各軸の傾き。

-90 度 = 0,

0 度 = 50,

+90 度 = 100 という値で表示される。

