

# ピアノ廊下開発手順

## 概要

### ピアノ廊下に使われるもの

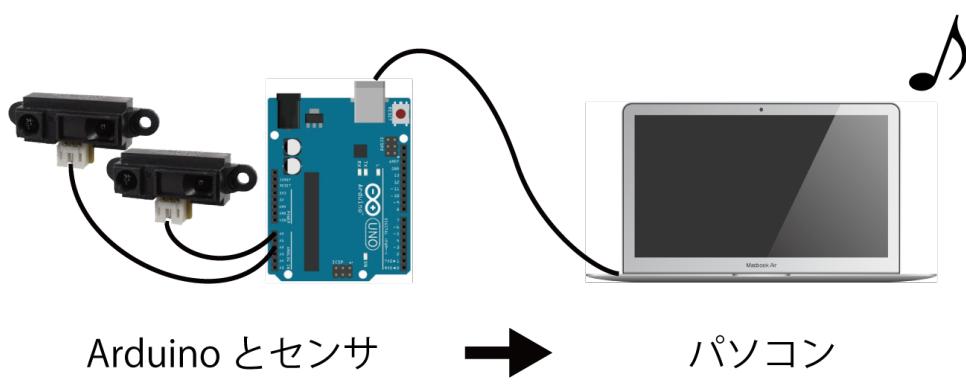
| 品名                                | 数量  | 参考価格(1ヶ) |
|-----------------------------------|-----|----------|
| PC (Win, MacOSX で動作確認済)           | 1   | -        |
| Arduino (今回は MEGA の互換品を使用)        | 1   | 2850     |
| ピアノ廊下のプログラム (配賦)                  | -   | -        |
| Canare 音声用 3芯シールド<br>(GND シールドあり) | 10m | 950      |
| 赤外線測距センサ GP2Y0A21YK (シャープ)        | 8   | 450      |
| 測距センサの固定具 (置き方に合わせて)              | -   | -        |

※Arduino はアナログ入力が最低 8 本あれば、互換品の Arduino を使ってもよい

## 必要な工具

| 名称                                  |
|-------------------------------------|
| ニッパ、ラジオペンチ、はんだごて、<br>フラックス、コードストリッパ |

## 仕組み



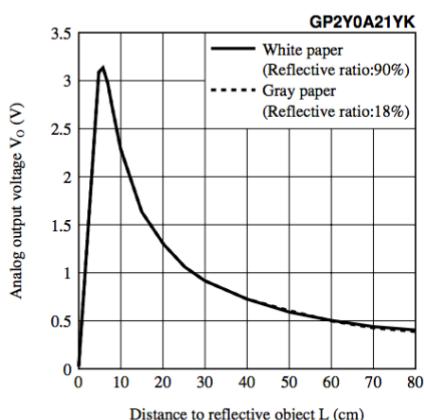
このピアノ廊下は、床の鍵盤を踏むと音がなるというものです。ただ、正確にはピアノのように押して（踏んで）音がなるわけではありません。その代わりに、自動ドアと同じ仕組みの赤外線測距センサを地面すれすれで反応するように設置します。そのため、足で遮ると音がなる、というのが正確な表現になります。今回使うセンサは 80cm までの距離を測れるものです。なので、少し余裕を見て 60cm ぐらいの幅で使えるもの、として今回は製作

しました。以下でもう少し詳しく紹介しておきます。

表 1. 電圧と距離の対応表



図 1. 赤外線測距センサ



センサには赤外線 LED と赤外線の受光部があります。まず、LED で赤外線を発光して、受光部はその返ってくる赤外線の量をみて、それに応じた電圧を出す、という寸法です。今回使うセンサは最大 3V の出力があるものです。80cm より近くに障害物がなければ 0V、近づくにつれて電圧は上がり、いちばん近くで最大 3V の出力を返します。

ここで返ってきた電圧をマイコン（マイクロコンピュータ、小さいコンピュータと考えて結構です）で計り、足があるかを判断します。今回は Arduino を使います。そのプログラムについては、以下の作り方の中で説明しています。

## 作り方

次からピアノ廊下の各要素の作り方の紹介です。要素別に分けると、「Arduino を使ったフットセンサ（もどき）」、「パソコンから音を鳴らす部分」の 3 つに分けられます。

最初は「Arduino を使ったフットセンサ（もどき）」の作り方紹介です。

### 1. Arduino を使ったフットセンサ（もどき）を作る

必要になる部品

| 品名                                       | 数量  | 価格(1ヶ) |
|--|-----|--------|
| Arduino (今回は互換品を使用)                      | 1   | 2850   |
| ピアノ廊下のプログラム (配賦)                         | -   | -      |
| Canare 音声用 3芯シールド<br>(ノイズ防止用 GND シールドつき) | 10m | 950    |
| 赤外線測距センサ GP2Y0A21YK (シャープ)               | 8   | 450    |
| 測距センサの固定具 (置き方に合わせて)                     | -   | -      |

## 必要な工具

| 名称  |
|---|
| ニッパ、ラジオペンチ、はんだごて、<br>フラックス、コードストリッパ、熱収縮チューブか絶縁テープ |

ここでは、フットセンサ（もどき）を作りましょう。もどき、としているのは正確にはフットセンサではなく、測距センサを踏むセンサとして利用しているためです。

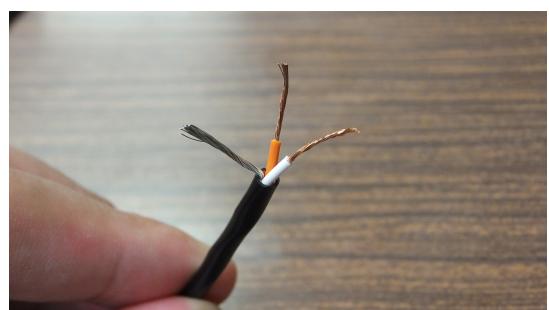
### Step 1. シールドをそれぞれ使う幅に用意する



最初に、センサと Arduino をつなぐコードを用意しましょう。センサは GND、+5V、Vo の 3 本の線が必要になります。そのため、コードは 3 芯のものを選定します。ノイズを拾わないように、できればシールドがついたケーブルがいいです。今回はカナレのオーディオ用のコードを使用しました。

今回、Arduino やパソコンなどを中央に設置し、それぞれの鍵盤を 30cm おきに設置するため、40,80,120,160cm のケーブルを 2 本ずつ作りました。この長さについては実情に応じてといったところです。ただ、コードがながければ長いほどノイズを拾いやすくなってしまうので、なるべく短くしたほうが良さそうです。

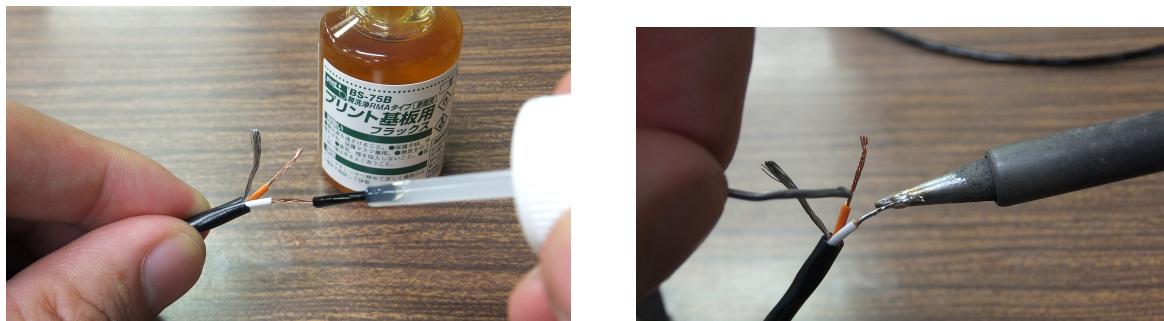
### Step 2. コードを剥き、束ねる



コードストリッパを用いてコードの皮膜を剥きます。コードストリッパがなければ、ニッパで代用もできます。潰さないよう、にぎにぎと軽く握りながら回すのがコツです。3 芯の

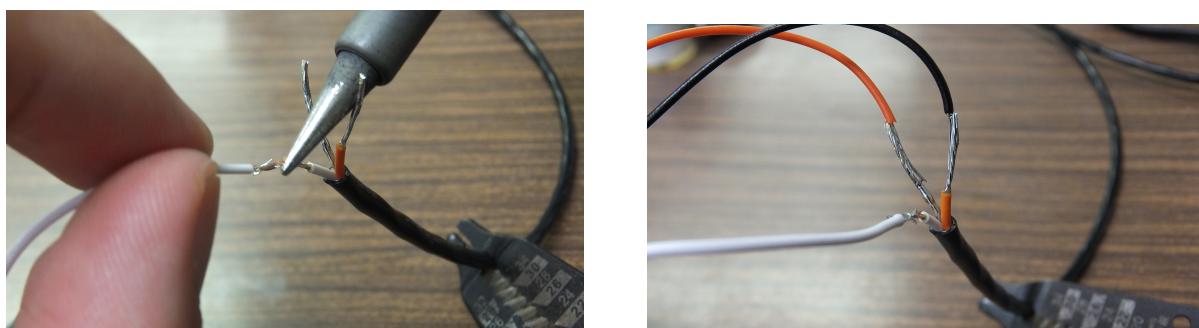
コードでは、外側を剥くと信号線が 2 本とシールド 1 本が出てきます。シールドはボサボサなのでまとめてよってあげましょう。信号線 2 本はまた皮膜があるのでそれぞれ 1cm 強ずつ剥き、被らないよう広げておきましょう。

### Step 3. フラックスを塗り、予備ハンダをする

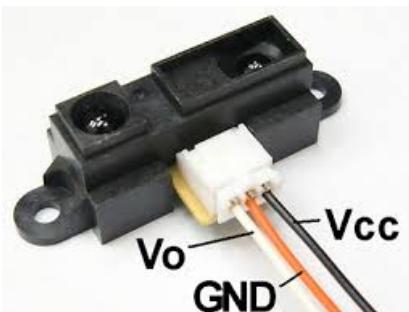


フラックスは、はんだがよく染みこむように塗るものです。ただ、無くとも気持ち多めにはんだをつけることで解決できます。なぜ予備ハンダをしておくかというと、次の作業で測距センサとハンダ付けする際に、はじめから線にハンダが乗っていればハンダ線をのせる手間が省けるためです。

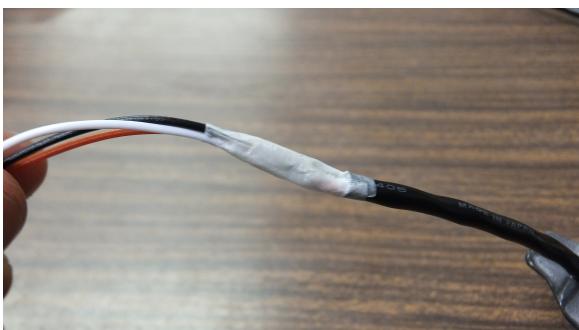
### Step 4. 測距センサの線とハンダ付けする



今回選定した測距センサの線は右図のようになっています。通常、電子工作では GND を黒、赤か橙を Vcc (電源電圧、+) にする事が多いので Vcc と GND の色に注意してください。例ではシールドの先では GND が黒になるよう繋ぎ変えていますのでご注意を。

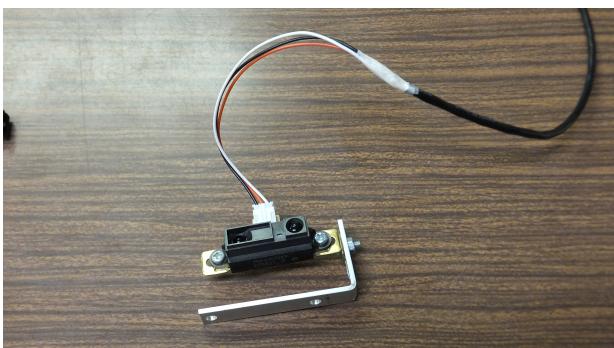


### Step 5. テープで絶縁する



上図のままでは配線した際にショートしてしまうので、一本ずつテープで絶縁します。ショートするとセンサだけでなく、つないでいる Arduino を始めとしたマイコンにも影響が出るので、しっかりと絶縁されている事を確認しましょう。熱収縮チューブでも大丈夫です。最近は百均でも売られています。

### Step 6. 台をつけてセンサが完成



最後に固定用の台を付けて完成です。固定用の台は地面に置く、台に固定する等、色々と工夫してみてください。センサは 8 つありますので、同様に後 7 つつくりましょう。根気のいる作業ですが頑張って。

### Step 7. プログラムを書き込む

Arduino にプログラムを書き込みます。まず、Arduino にプログラムを書き込むソフトをダウンロードしてきます。Google などで Arduino と調べると一番に出てくるページです (<https://www.arduino.cc/>)。そのページの「Download」からお手持ちの PC にあったものをダウンロードして下さい。ダウンロードされたらインストールしておきます。

インストールが終わったら、まず起動してみます。

```
sketch_feb01a | Arduino 1.6.5
sketch_feb01a
1 void setup() {
2 // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7 // put your main code here, to run repeatedly:
8
9 }
```

Arduino Mega or Mega 2560, ATmega2560 (Mega 2560) on /dev/cu.usbmodem1451

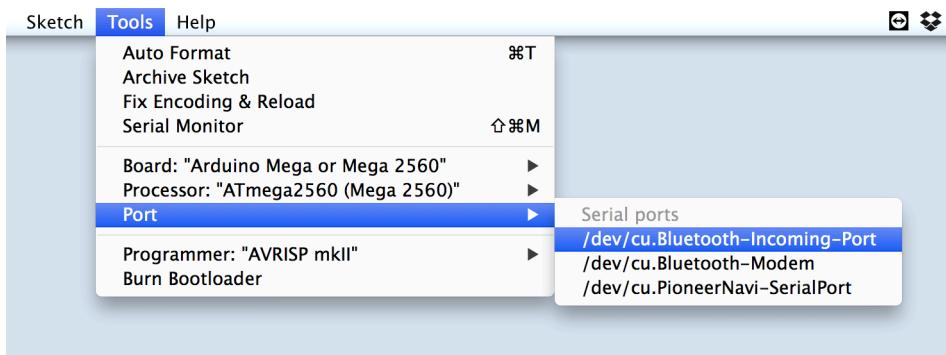
この画面が出たら大丈夫です。次に、プログラムはこちらで用意したものがありますので、それを書き込んでいきます。GitHub からプログラムのファイル群をダウンロードします (<https://github.com/oratake/pianoroad>)。これを「File」>「Open」から開きます。

```
kenban_1_8 | Arduino 1.6.5
kenban_1_8
1 int ledState1 = HIGH;
2 int ledState2 = HIGH;
3 int ledState3 = HIGH;
4 int ledState4 = HIGH;
5 int ledState5 = HIGH;
6 int ledState6 = HIGH;
7 int ledState7 = HIGH;
8 int ledState8 = HIGH;
9
10 // ヒステリシス
11 int volH = 200;
12 int volL = 120;
13
14 // Test LED
15 int pin1 = 38;
16 int pin2 = 39;
17 int pin3 = 40;
18 int pin4 = 41;
19 int pin5 = 42;
20 int pin6 = 43;
21 int pin7 = 44;
22 int pin8 = 45;
23
24 void setup() {
25 delay(5000);
26 Serial.begin(31500);
27 Serial.write(0xC1); //プログラムチェンジ
28 Serial.write(0x00); //楽器
```

Arduino Mega or Mega 2560, ATmega2560 (Mega 2560) on /dev/cu.usbmodem1451

このようなプログラムが表示されるはずです。最後に書き込みです。書き込みの前に Arduino とつなぐための設定をします。USB が Arduino につながったのを確認した上で

「Tools」を選びます。そして Arduino の種類を調べ、それに近い物を Board、Processor から選びます。もし、わからない場合は買った商品名と Board、Processor という単語と一緒に調べれば設定のしかたが出てくるはずです。そして、ポートを選びます。ポートについては /dev/cu.usbmodem14\*\* (\*\*は適当な数字) を選択すれば大丈夫です。

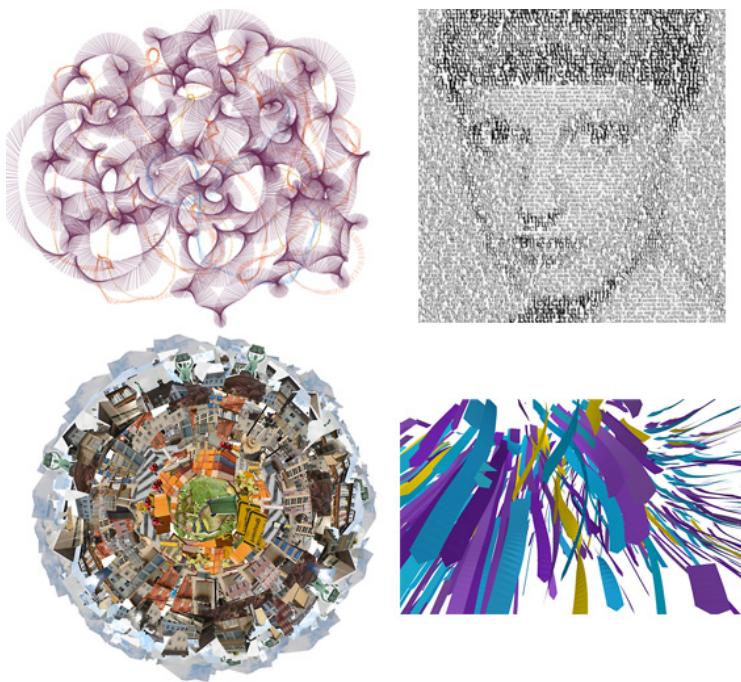


これで Arduino とセンサの連携ができるようになりました。

## 2. Processing を使ってパソコンから音を鳴らす



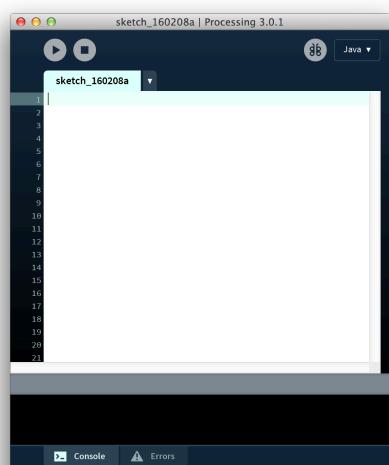
次は Processing を使ってパソコンから音を鳴らす部分の解説です。Processing は 2001 年に作られた MIT ラボ発のビジュアルデザインに特化した、オープンソース統合開発環境です。言語は主に Java を用います。いまいちピンと来ないかもしれません、要するにパソコンでアートをしたい人のための、比較的易しい開発環境です。代表的な使い方としては、アート作品への利用が一番ですが、今回のように、少しライブラリの力を借りて音を操作することも出来ます。



<http://nakayasu.com/lecture/media-programming%E6%A6%82%E8%A6%B3/4968>  
 Nakayasu.com - 講義ノート より

この Processing ですが、今回はビジュアル用途ではなく、音の出し入れに用います。まずは Arduino と同様、Github の筆者のリポジトリ (<http://github.com/oratake/pianoroad>) から Download してきましょう。

それではこのプログラムを開くための Processing をダウンロードして来ましょう (<https://processing.org/>)。ページ左の Download を選択。Donation Ware ですので、使ってみて素晴らしいと思った方は Donation してあげてください。とりあえず使ってみたい方は NoDonation で Download することが出来ます。Download したらインストールして Processing を開きましょう。



このようなウィンドウが表示されれば大丈夫です。それでは先程 Github から落としてきたプログラム (.pde ファイル) を開いてみましょう。

さて、このプログラム、一箇所だけ気をつけねばならないポイントがあります。



The screenshot shows the Processing 3.0.1 IDE interface with the title bar "doremi | Processing 3.0.1". The main area displays the following Java code:

```
13 int ch = 0; //Midi Ch
14 int note = 60; //Note
15 int vel = 127; //Velocity
16 int program = 1; // プログラムチェンジ
17 int dev = 1; //音源の設定
18 int devLength = 0; // デバイスの数
19
20 void setup(){
21     size(256,256);
22
23     // ポート名(Arduinoで選択したもの)とスピードを指定してシリアルポート
24     myPort = new Serial(this, "/dev/cu.usbmodem1411", 9600);
25     // 改行コードが受信されるまで受信し続ける
26     myPort.bufferUntil('\n');
27
28     devLength = RWMidi.getOutputDevices ().length;
29     output = RWMidi.getOutputDevices () [dev].createOutput();
30     output.sendProgramChange (program);
31
32     for (int i = 0; i < devLength; i++) {
33         println ("Output Device " + i + " : " + RWMidi.getOutput
```

At the bottom of the IDE, there are tabs for "Console" and "Errors".

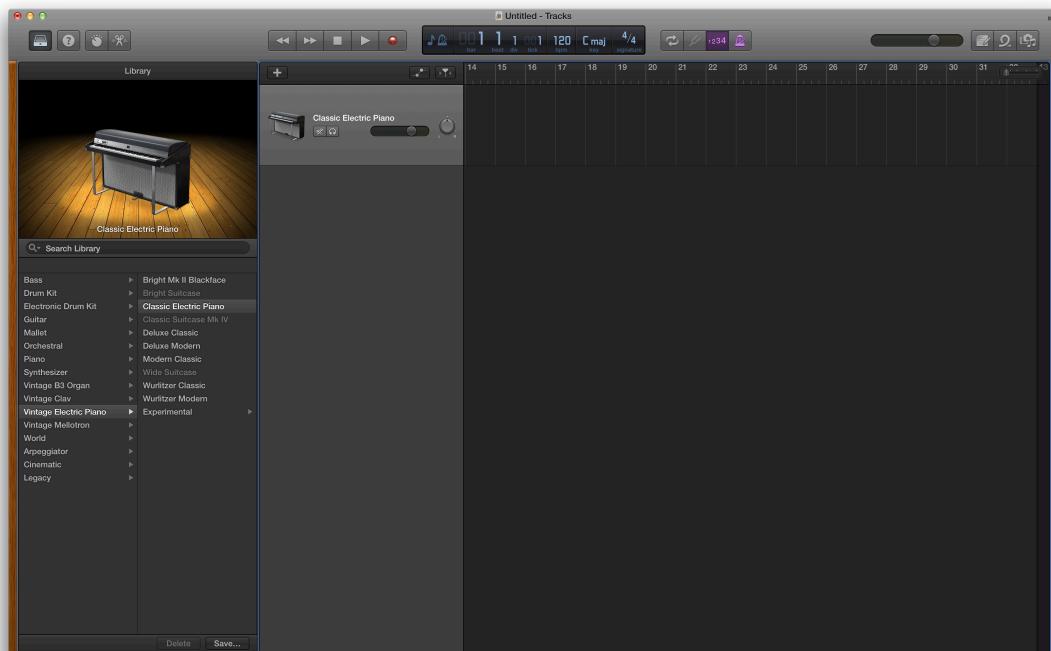
24 行目のコードですが、この”dev/cu.usbmodem1411”というところですが、Arduino のところで書いてあったポートをそのまま書き写してください。パソコンを起動するたびにこのポートは変わるものなので、Arduino のソフトでポートを確認、そのポート名を書き写す事を毎度行います。そして左上の再生ボタン (Run) を押せば準備は完了です。

さて、次は Mac のみ必要ですが、書き写したら今度は midi を流すためのソフトを起動しましょう。なぜ Mac だけかというと、最近の Mac は標準で MIDI をサポートしていないため、MIDI を使うソフトに環境を作つてもらわなくてはなりません。

今回使うのは GarageBand です。無かった方は、App Store にて無料で Download 出来ますので、インストールします。インストールが済んだら起動しましょう。初回は音源のダウンロードで 3 分ほど待ちます。



起動してこの画面になつたら、”Empty Project”を選択。すると下記のような画面が出るので、後は左側から楽器を選ぶだけで様々な楽器の音に変えることが出来ます。



これで、すべての部品が揃います。後は楽器を変えて遊ぶだけ。少し作るのに時間がかかるかもしれません、電子工作に慣れてくれば発展性のあるものです。ぜひ作ってみてください。