

電紙オルガンを作ろう

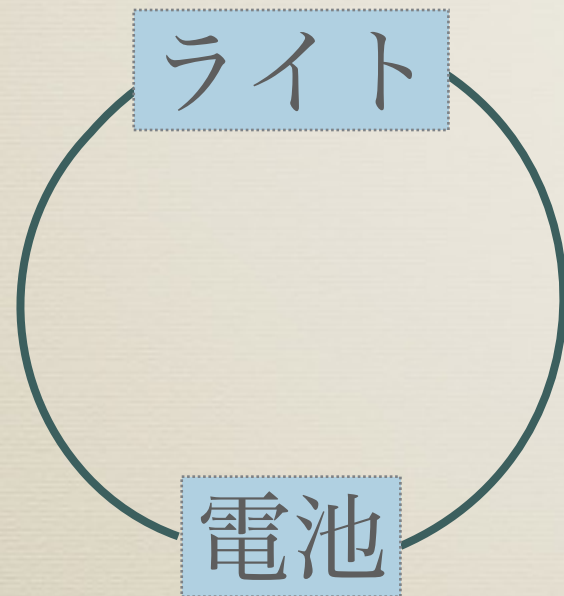
概要

- * 導入
- * デジタル電子工作の基礎知識
- * 電紙オルガンを作ろう
- * まとめ

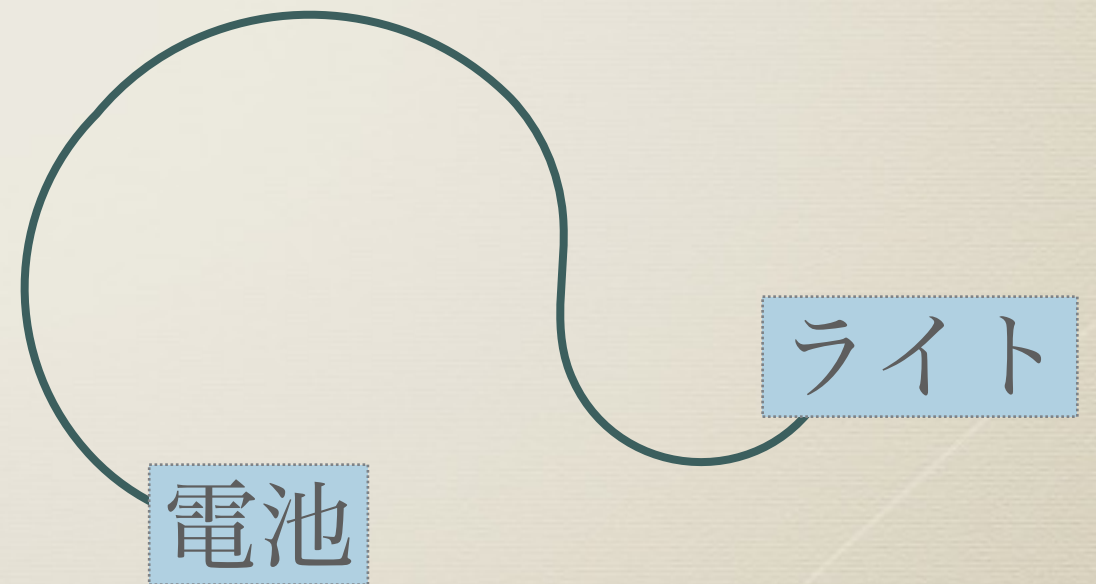
デジタル電子工作に 必要な知識

電気回路/電子回路

- * 回路は回路(circuit)です。全体が輪っかとして繋がっていないと動けません。



- * 回路(circuit)



- * 回路ではない

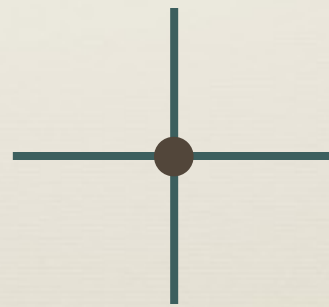
電子部品と回路図

配線

- * 電気を流す線で金銀銅 (ほとんどが銅)が使われます。
- * 他の線と意図しない接触をさせないために被覆されています。
- * 回路図では線として書かれます。
- * 交差点では接続するかしないかの注意が必要です。
- * デジタル回路では何本かをまとめてI本の線として書かれることも多いです。



交差する
(線は繋がっていない)



繋がっている



8本の束

スイッチ

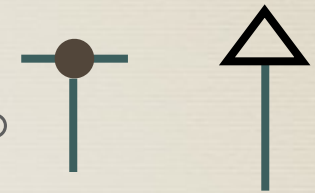
- * 基本は単純に配線をつないだり切ったりする装置です。
(配線を切ったら回路ではなくなります。)
- * 様々なスイッチがあり， 入力センサーとしても良く利用されます。



電源/グランド

- * 電源[V_{cc} , V_{dd}]

- * 複数の電源(電圧)が必要な場合がある。



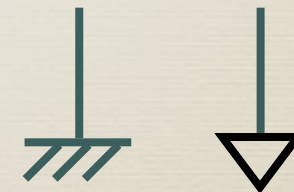
- * 5V, 3.3V, 12V

- * グランド [GRND , V_{ss}]

- * 回路のグランド

- * アース

- * 筐体



抵抗

- * 電流を熱に変換(発熱します)
- * 電流制限や電圧の変更に利用します。
- * 昔と今では記号が違う
- * JISカラーコードについて
 - * 黒い0服, 茶を1杯, 赤い2ンジン, 橙三の男, 黄4恵子, 緑5
 - * 青二才の6でなし, 紫7部, 灰8(や)ー, 白(ホワイト)9リスマス



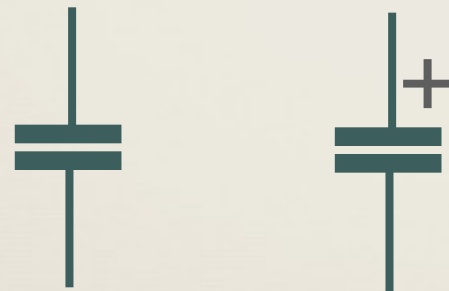
昔



今

コンデンサ

- * デジタル回路では電源関係および、ノイズ除去に使います。



その他

- * ダイオード

- * 一方向にしか電流が流れません。



- * 発光ダイオード

- * 流れた電流が光に変換されます。



オームの法則

$$V = IR$$

- * V 電圧 単位 V (ボルト) 「ボルタ」さんの名前より
- * I 電流 単位 A (アンペア) 「アンペール」さんの名前より
- * R 抵抗 単位 Ω (オーム) 「オーム」さんの名前より

その他

*電力 $W = I \times V$ 単位 W

「ワット」さんの名前より

*静電容量 F (ファラッド)

「ファラデー」さんの名前より

*インダクタンス H (ヘンリー)

「ヘンリー」さんの名前より

*mA とか $K\Omega$ とか $M\Omega$ とかは

*m(ミリ) $1/1000$ を表す

* μ (マイクロ) $1/1000000000$

*n(ナノ) $1/00000000000$

*p(ピコ) $1/000000000000000000$

*K(キロ) 1000 倍を表す

*M(メガ) 10000000 倍を表す

電圧/電流を制御する。

- * 抵抗で分圧する。
- * LEDをつける場合
- * スイッチを入力にする場合

部品の取り扱いについて

- * 部品の定格

- * 通常電流, 最大電流, 通常電圧, 最大電圧

- * 超えると壊れます。

- * ICなど..静電気で壊れます。(人によります)

- * 配線ケーブル

- * 曲げたり、潰したりすると断線や性能劣化が起こります。

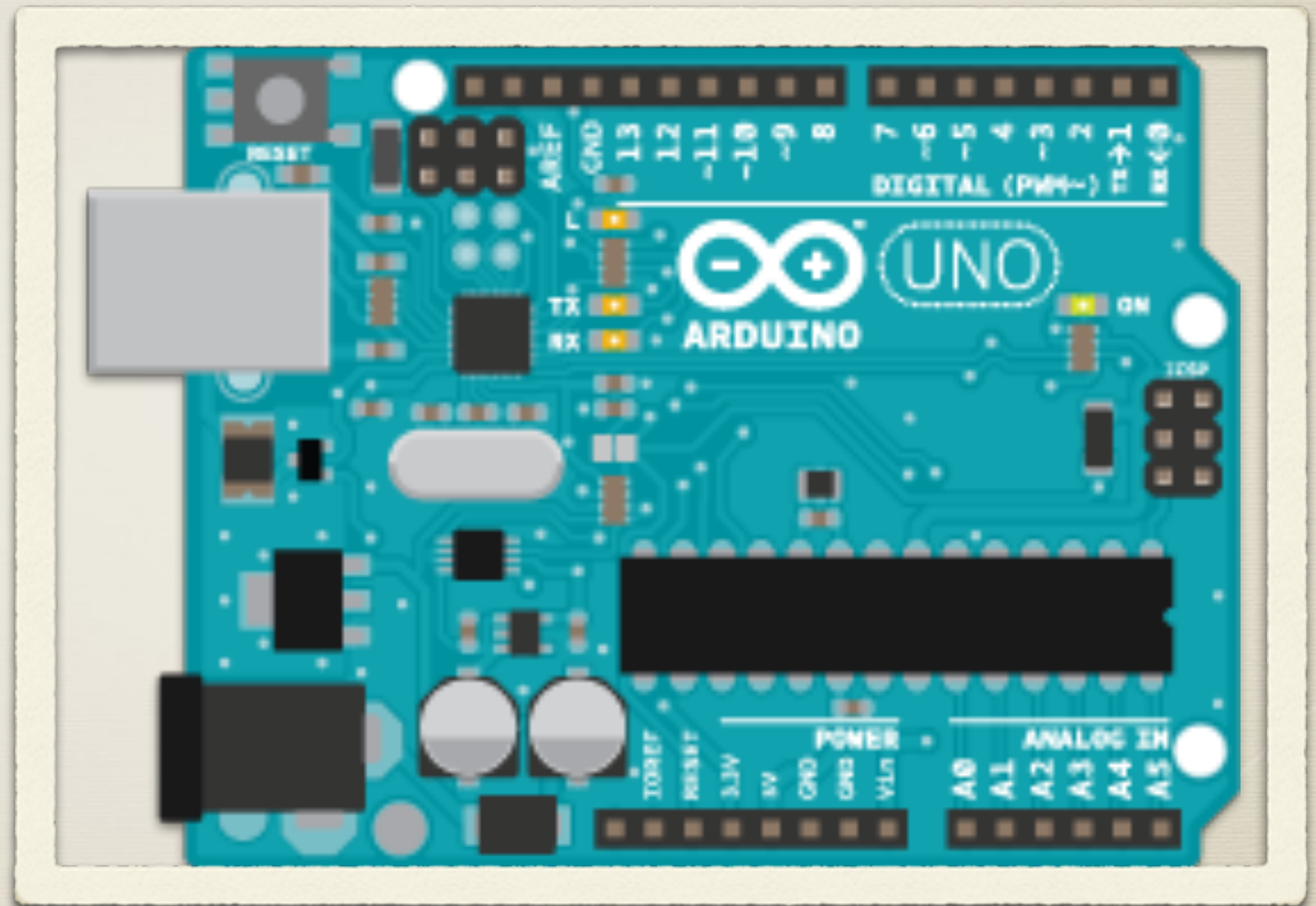
Tips

- * 電源について
 - * 電源電圧と電源容量
- * LEDについて
 - * 壊れます
 - * 物理的破壊, 温度(半田付けの場合), 過電流
 - * arduinoでは直付け(直接GPIOに刺す)しても壊れないようです。
 - * 電流制限 or 電流限界
- * コンデンサについて
 - * デジタル工作ではノイズを防ぐためにICの電源に入れる

arduinoの概要

arduinoの概要

- * 入力
- * 演算
- * 出力
- * 独立して動く



入力



- * デジタル入力とアナログ入力
- * 通信によるセンサー入力



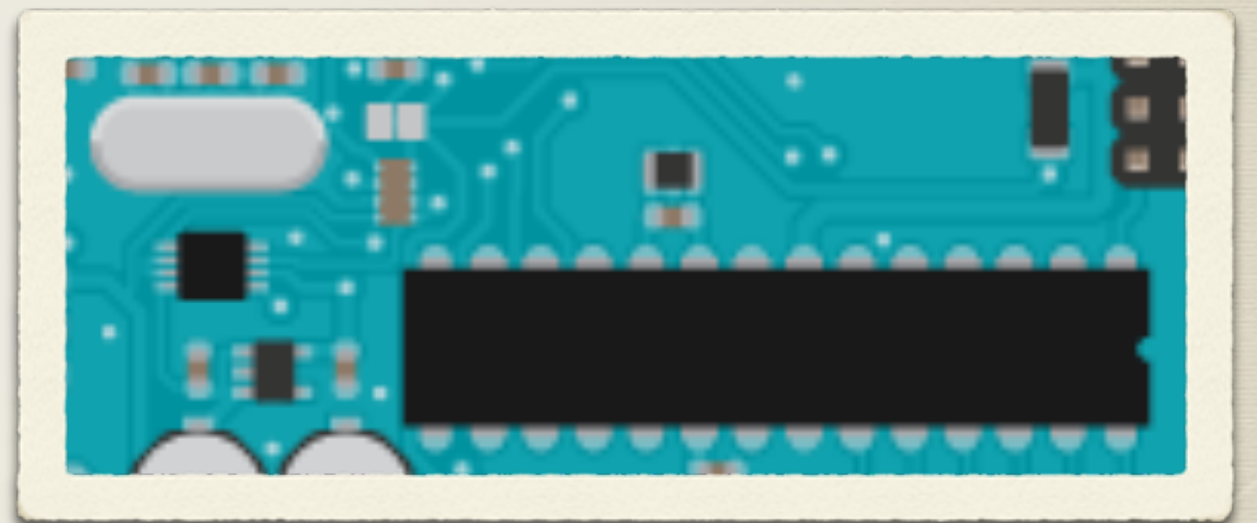
出力

- * デジタル出力(o,I) → (GND, Vcc)
- * PWM(高速にo,Iを繰り返す)
on/offの時間の割合でアナログ的に制御



演算

* 入力を取り込み, 出力する。



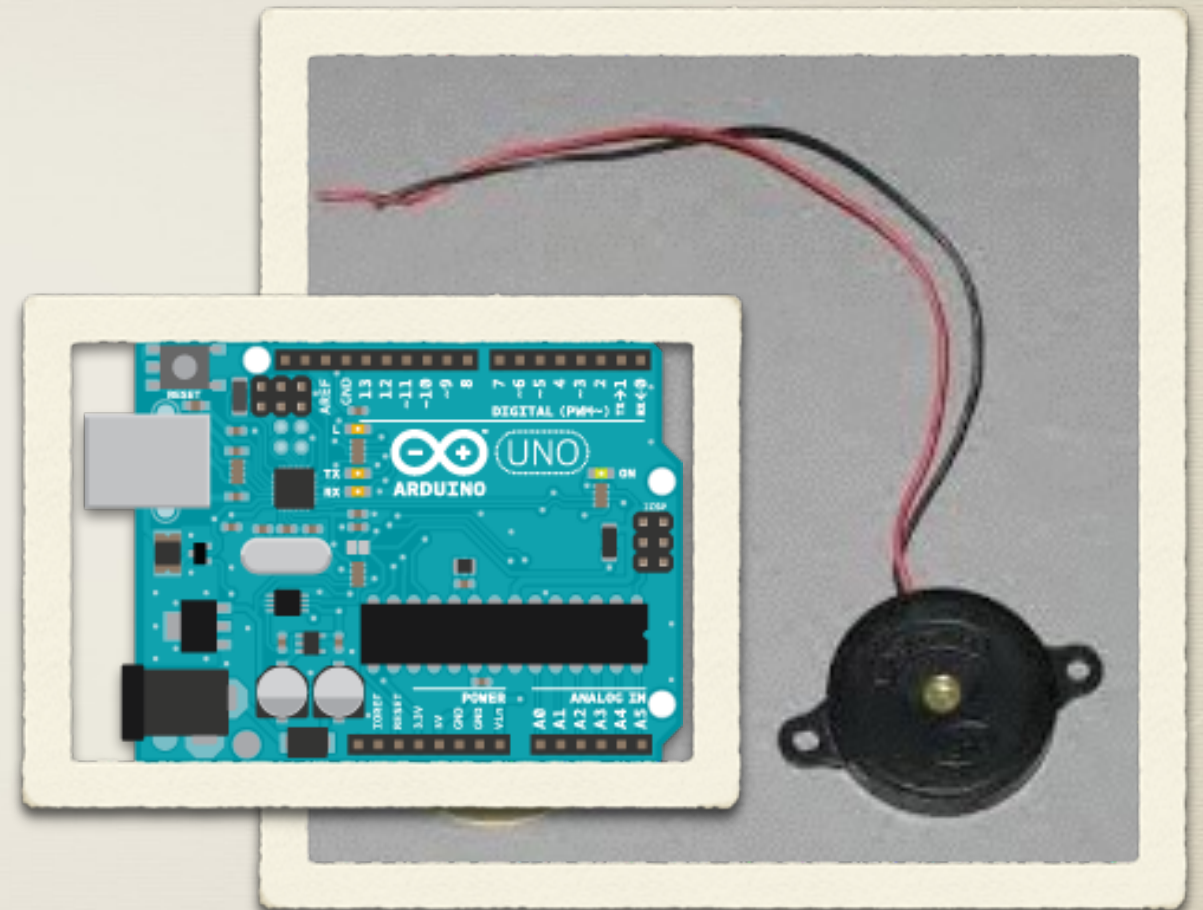
IDEの設定

- * IDEをダウンロードし、インストールする。
(<http://arduino.cc/> のDownloadより)
- * IDEプログラムを立ち上げて最初に行うこと
 - * arduinoの種類の選択 (“Arduino UNO”)
 - * 接続ポートの指定(/dev/cu.xxxxxx)
 - * programmerの指定 (“ArduinoISP”)
 - ※ カッコ内はMac OSの場合の例

- 電紙オルガンを作ろう -

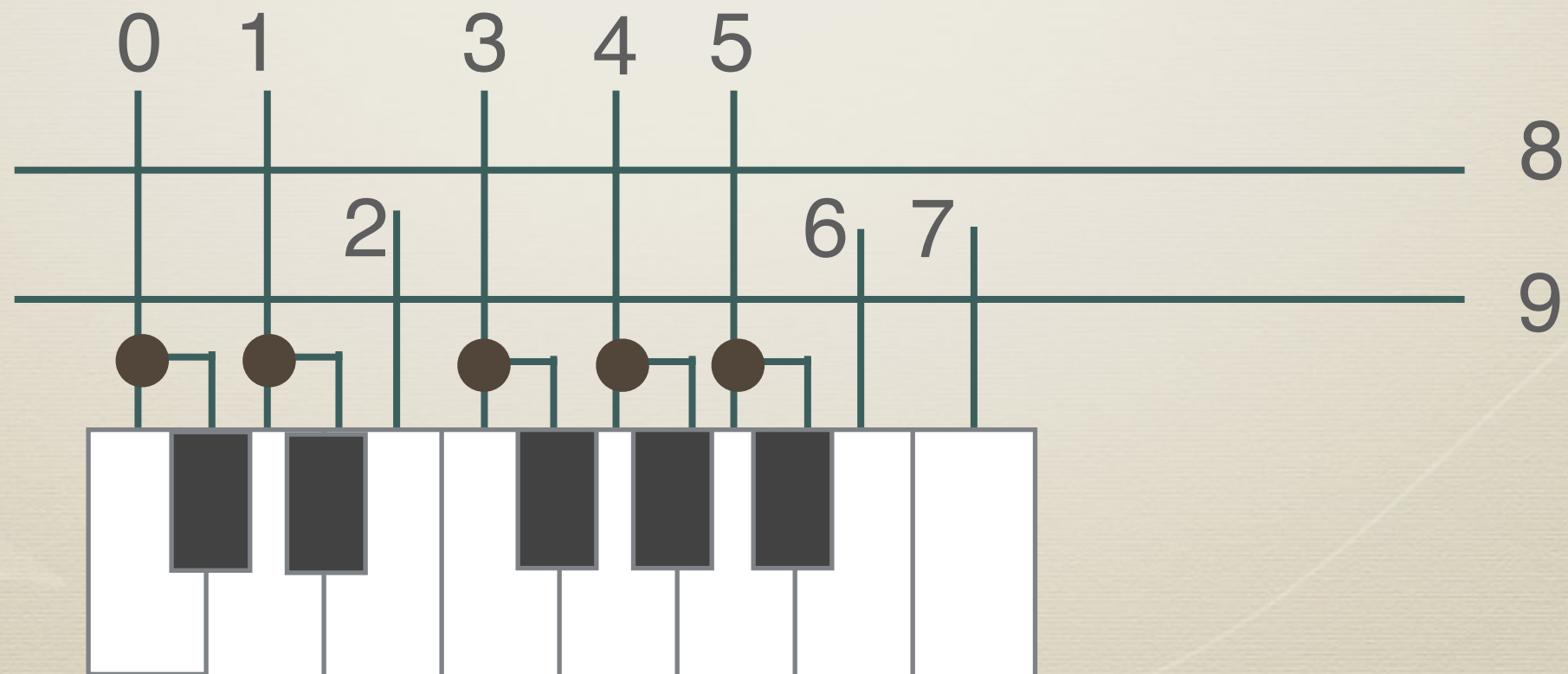
準備するもの

- * arduino 本体
- * キーボード用紙
- * 8pinケーブル
- * 1pinケーブル×2
- * 圧電ブザー



キーボードの回路

- * キーボードは12個
- * GPIO端子が足りない??
- * たくさんのスイッチを検出するためマトリクスを使います。



キーボードを組み立てる。

組み立ての前に確認

- * 印刷の確認
- * 絶縁テープ(セロハンテープ)を貼る

折り曲げと差し込み

* キー折り曲げて白鍵, 黒鍵を順に差し込みます。

キーボードに ケーブルを貼り付ける

- * 銅製の導電テープで黒鍵と白鍵の配線しながらケーブルをテープで貼り付けるます。
- * 配線
 - * 黒 Do - Do#, 茶 Re - Re#, 赤 Mi - Mi#, 橙 Fa
 - * 黄 So - So#, 緑 La - La#, 青 Si , 紫 Do(高音)
 - * 灰色 黒鍵, 白 白鍵,

arduinoにつなぐ

- * キーボードのケーブルをarduinoにつなぎます。
- * 鍵盤8pinケーブルの0～7をGPIO0～7 に差し込む。(C, C#側が0)
- * 黒鍵用の1pinケーブルをGPIO8に差し込む。
- * 白鍵用の1pinケーブルをGPIO9に差し込む。
- * 圧電ブザーの端子(黒)をGNDに端子(赤)をGPIO13に差し込む。

スケッチ

※ スケッチは(<https://github.com/akitam/paperkey>)
よりダウンロードできます。

やること

- * キーが押されていたらそのキーに対応する音を出す。
- * キーが離されたら音を止める
- * 別のキーが押されたら、音を止めて、押されているキーの音を出す

制限事項

- * 今回はtone関数を使う(2音以上は出ません)
- * 押されているキーが変わったら音を変える
- * キーのデジタル入力はプルアップとする。(キーが押されたら0, 離されたら1が返る)

if 文を使う

- * キーが押されているかどうかを順にチェックする。
- * まずは一行目(白鍵)をチェック
 - * 押されていれば, 指定された音階の音を出す。
- * どこも押されていなければ, 二行目(黒鍵)をチェック
 - * 押されていれば, 指定された音階の音を出す。
- * どこも押されていなければ、音を止める

完成

* しましたか?

よく使うアドレス

- * arduino本家 <http://arduino.cc/>
- * <http://www.adafruit.com/>
- * <https://www.sparkfun.com/>
- * <http://akizukidenshi.com/catalog/top.aspx>
- * <http://eleshop.jp/shop/default.aspx>
- * <https://www.switch-science.com/>
- * <http://www.sengoku.co.jp/>