Arduinoではじめる
素晴らしきマイコンの世界

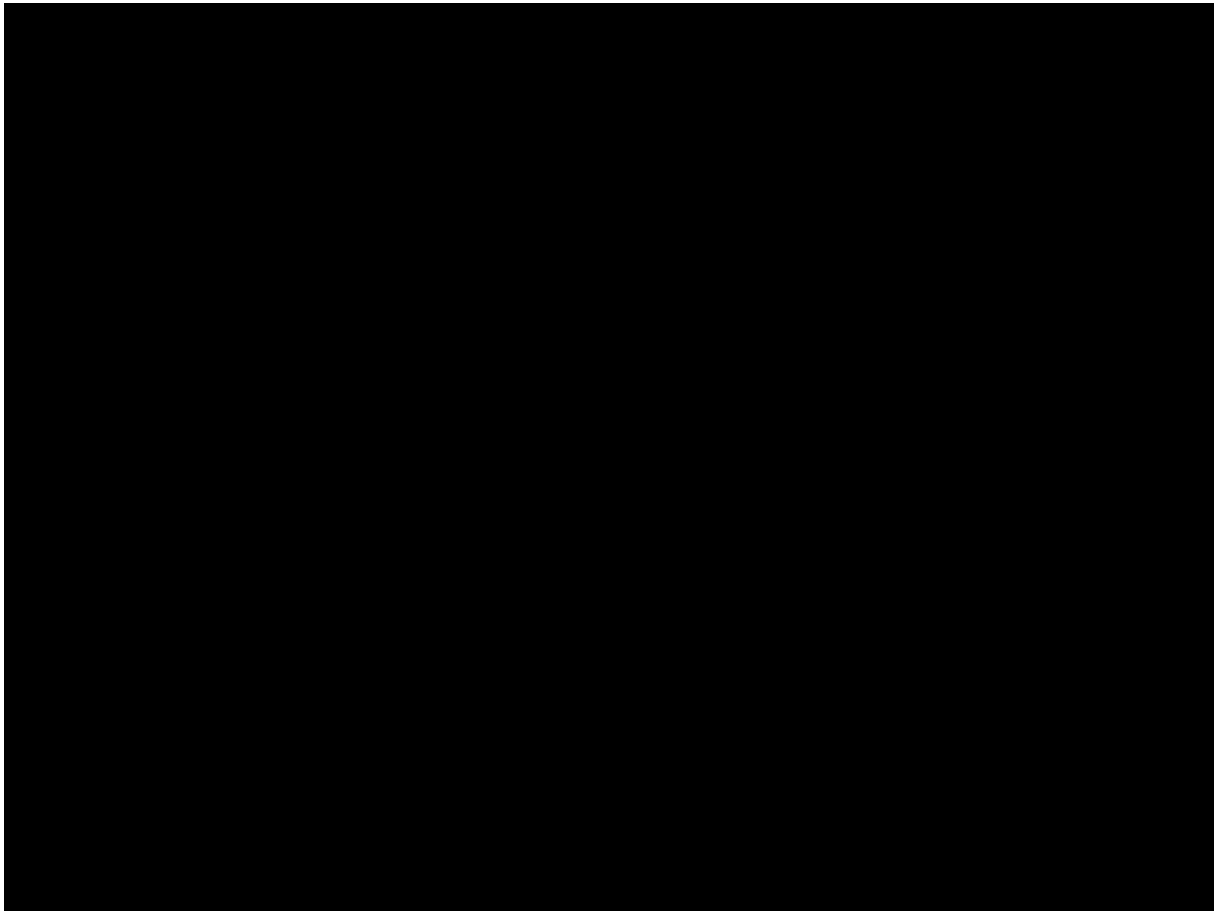
IT Technical Seminar V12

2011.05.20

株式会社オープントーン

根本

今日はこんなことします！



今回の目的

- まずはマイコンを知ってもらう
  - マイコンってなに？
  - Arduinoって一体・・・？
- マイコンって簡単ってことを知ってもらう
  - 基礎講座
  - ちょっとだけ応用
- 素晴らしきマイコンの世界に
  興味を持っていただければ

おらっしゃ
あぁぁ！！！

```
 ∩∧ ∧
ヽ( ﾟДﾟ)
   ＼⊂＼
    o-、 )～
      ∪
```

さっそくですが・・・

マイコンってなんだろ？

CPUやメモリを1つのLSIチップに集積した回路

どんなことに使われている？

- 現代ではいろいろなものにマイコンが利用されている
- どんなものに利用されているかご存じ？

ところでArduinoって？

- アルドゥイーノと発音
- イタリア生まれ
- 入出力ポートを備えた基板
- C言語風のArduino言語とそれの統合開発環境から構成されるシステム

とにかく・・・ **開発環境一式が揃っているため、誰でも簡単にマイコンの開発を始めることが出来る！**

まずは準備を始めます！

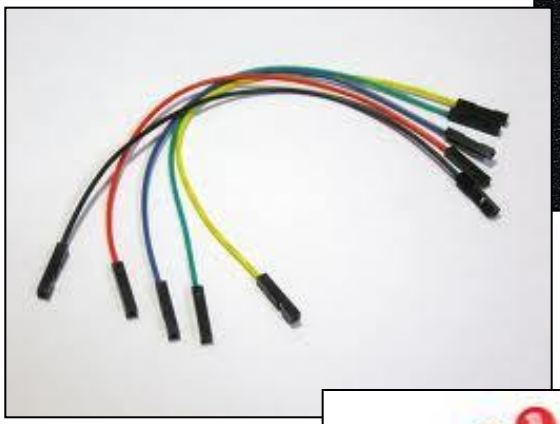
必要なもの

- Arduino UNO
  - 3000円程度
- USBケーブル
  - 580円程度
  - Aオス-Bオス

あったらいいもの

- ブレッドボード
  - 半田付けの必要がない
  - 250円程度

- ジャンパワイヤ
  - 柔軟性がある
  - 10本300円程度
  - オス・メスに注意



- 発光ダイオード
  - 10本300円程度

次は開発環境構築です！

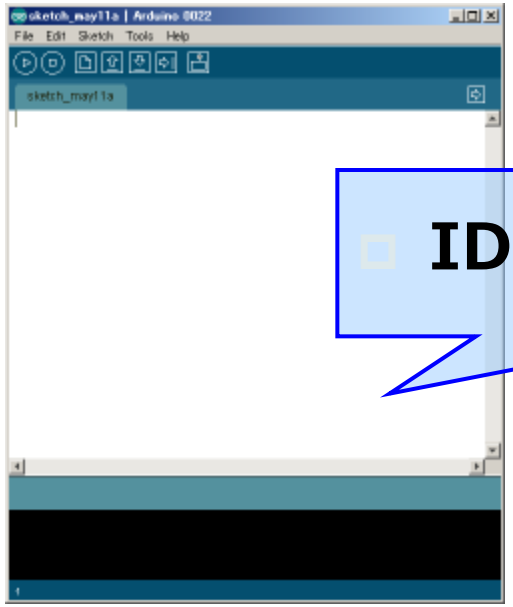
開発環境をつくりましょう！

- http://arduino.cc/en/

Downloadページに進み、使用OSにあったダウンロードリンクをクリック
（2011/5/20時点のバージョンは「Arduino 0022」）

インストール

- Arduino IDEのインストールと設定
  - インストールは、先ほどダウンロードした「arduino-0022.zip」を任意の場所に展開するだけ
  - 展開したファイルの中の「arduino.exe」を実行してIDEを起動する



IDEを起動したところ

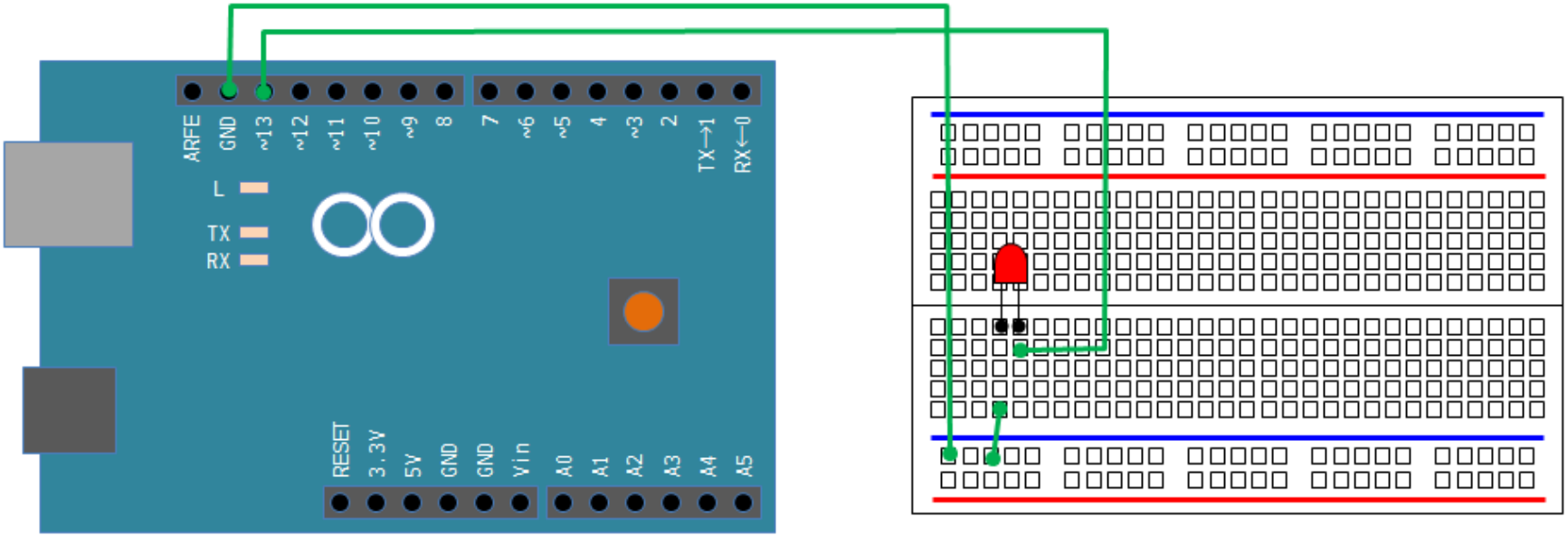
IDEの設定

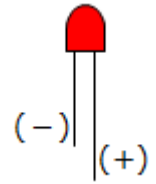
- Arduino IDEのインストールと設定（続き）
  - Tools > Board > Arduino Unoを選択する
  - Tools > Serial Portで、USBケーブルを接続したシリアルポートを指定する

以上で、開発環境構築は完了です。

いよいよ、部品の組み立てとプログラムです！

ArduinoにLEDを配線

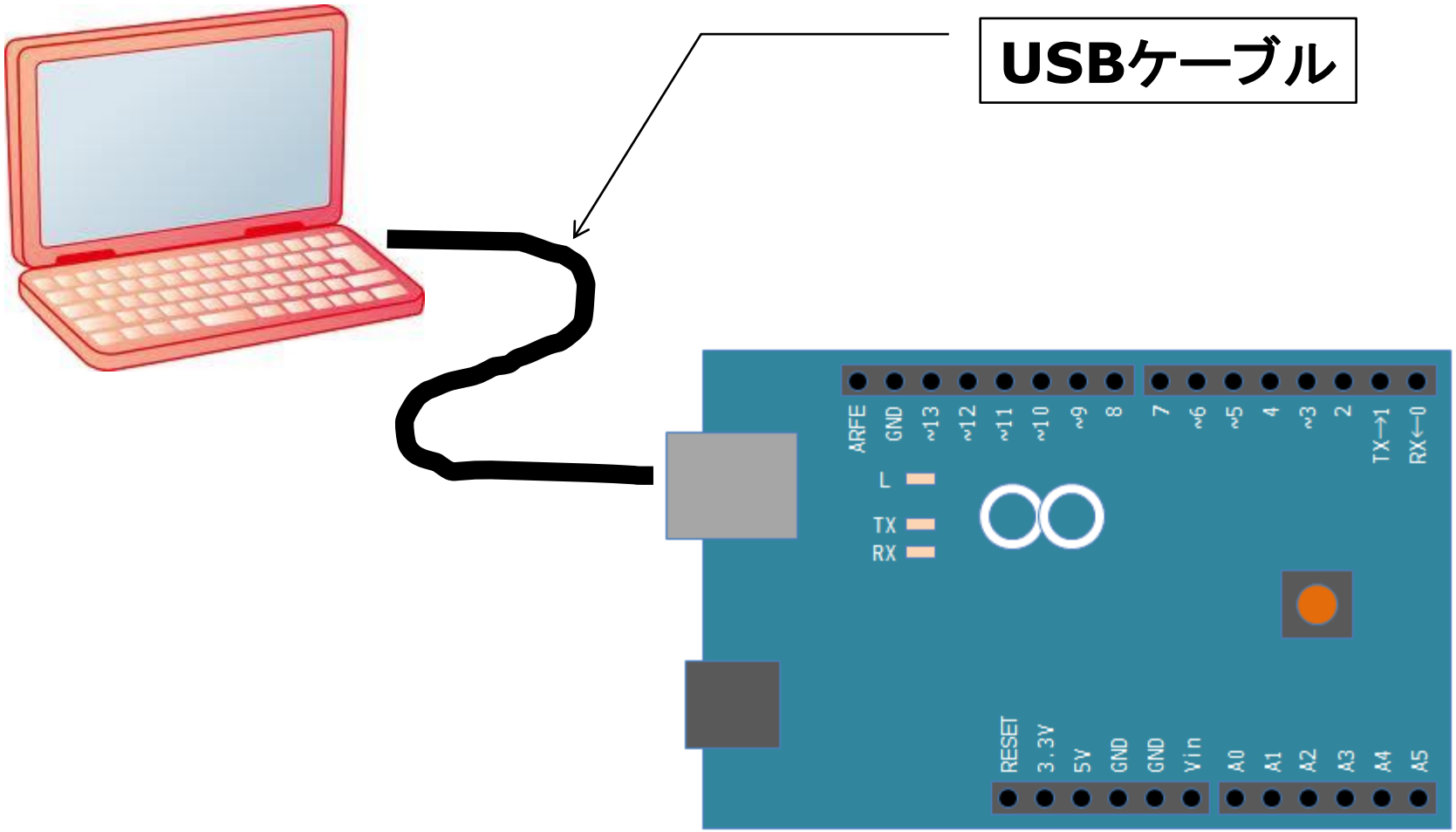




1．LEDのプラスとArduinoのPIN13をジャンパワイヤでつなげる。
2．LEDのマイナスとArduinoのGNDをジャンパワイヤでつなげる。

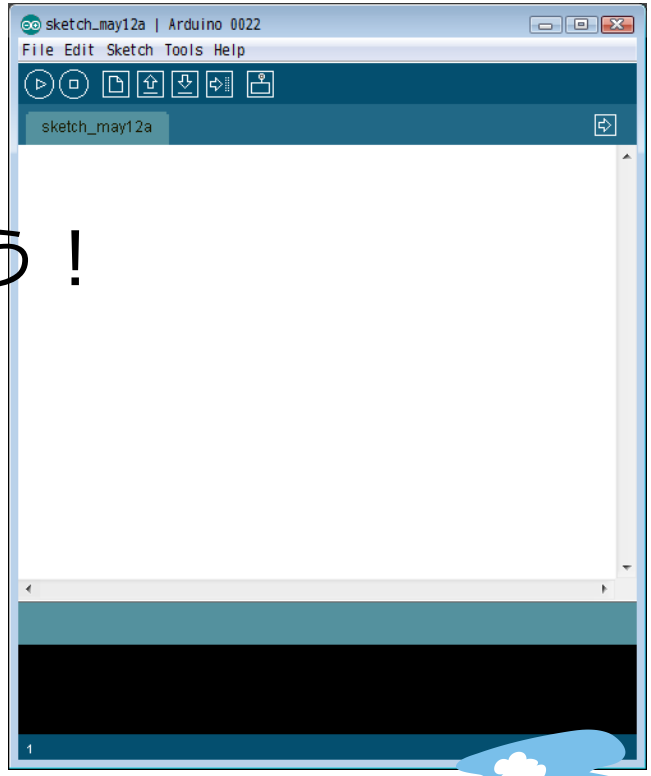
※LEDは足が長いほうがプラスで、短いほうがマイナスです。

配線したArduinoとPCを接続



スケッチしてみよう！

- Arduino IDEを起動

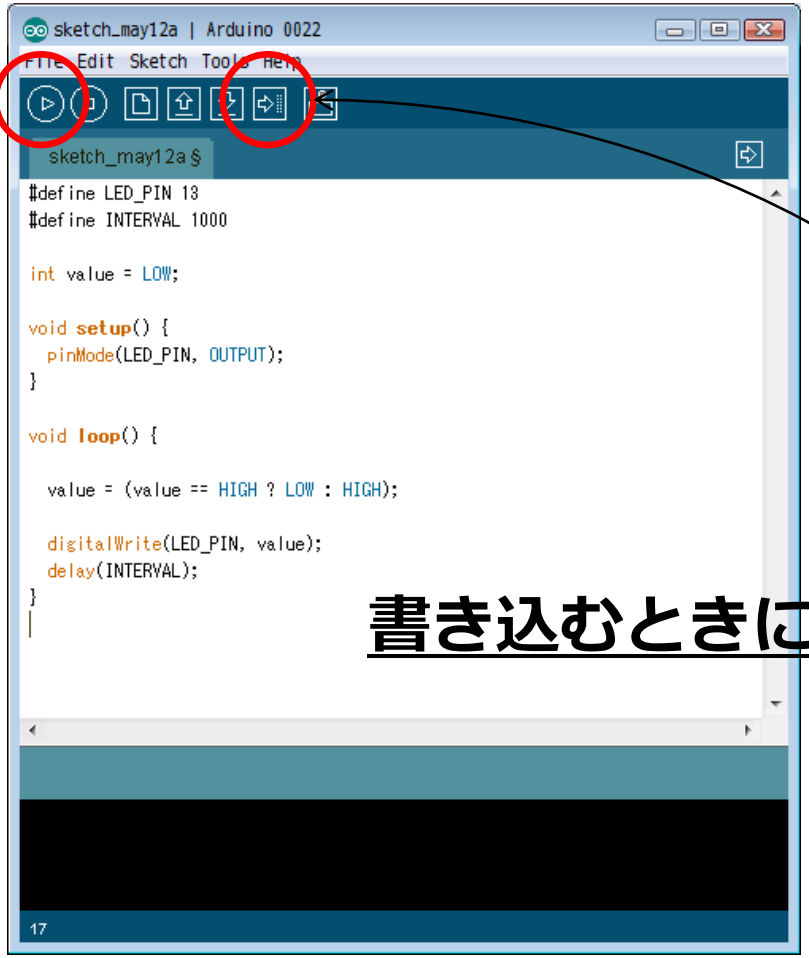


- 次のスケッチを記述してみよう！

```
#define LED_PIN 13
#define INTERVAL 1000

int value = LOW;

void setup() {
  pinMode(LED_PIN, OUTPUT);
}

void loop() {

  value = (value == HIGH ? LOW : HIGH);

  digitalWrite(LED_PIN, value);
  delay(INTERVAL);
}
```

VerifyとUpload

データに誤りがないか、規格に従っているかなどを検査する



ArduinoにUploadする

書き込むときにはここをClick！

で、出来たのがこんな感じ

じゃあソースコードを見てみましょう！

スケッチの基本とは？

```
#define LED_PIN 13
#define INTERVAL 1000

int value = LOW;

void setup() {
  pinMode(LED_PIN, OUTPUT);
}

void loop() {

  value = (value == HIGH ? LOW : HIGH);

  digitalWrite(LED_PIN, value);
  delay(INTERVAL);
}
```

setup()
開始時に一度だけ呼ばれる

loop()
この処理が呼ばれ続ける

チカチカ光らせるスケッチ

```
#define LED_PIN 13
#define INTERVAL 1000

int value = LOW;

void setup() {
  pinMode(LED_PIN, OUTPUT);
}

void loop() {

  value = (value == HIGH ? LOW : HIGH);

  digitalWrite(LED_PIN, value);
  delay(INTERVAL);
}
```

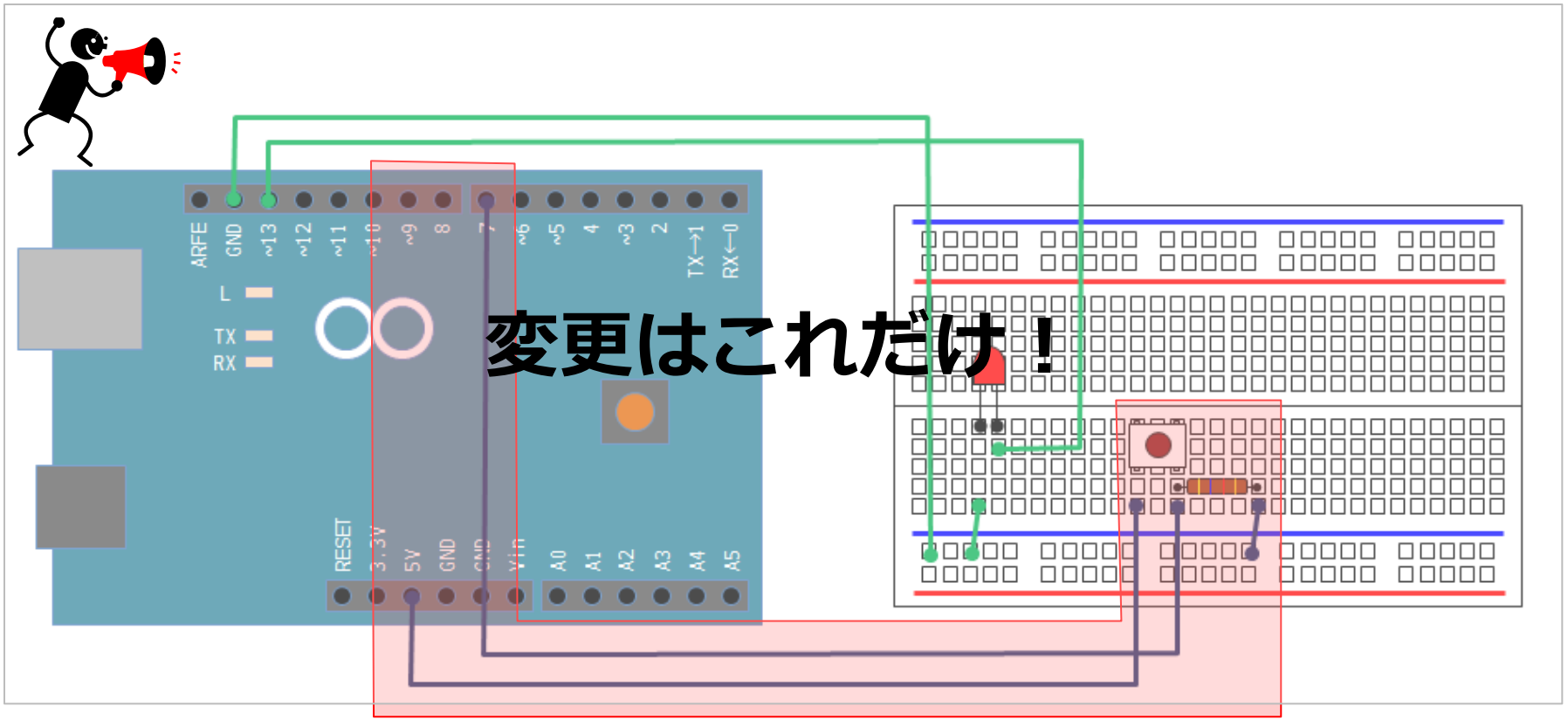
PIN13をOUTPUTとして指定

デジタル出力

指定したミリ秒待機します。

簡単でしょ？

では入力も加えましょう！



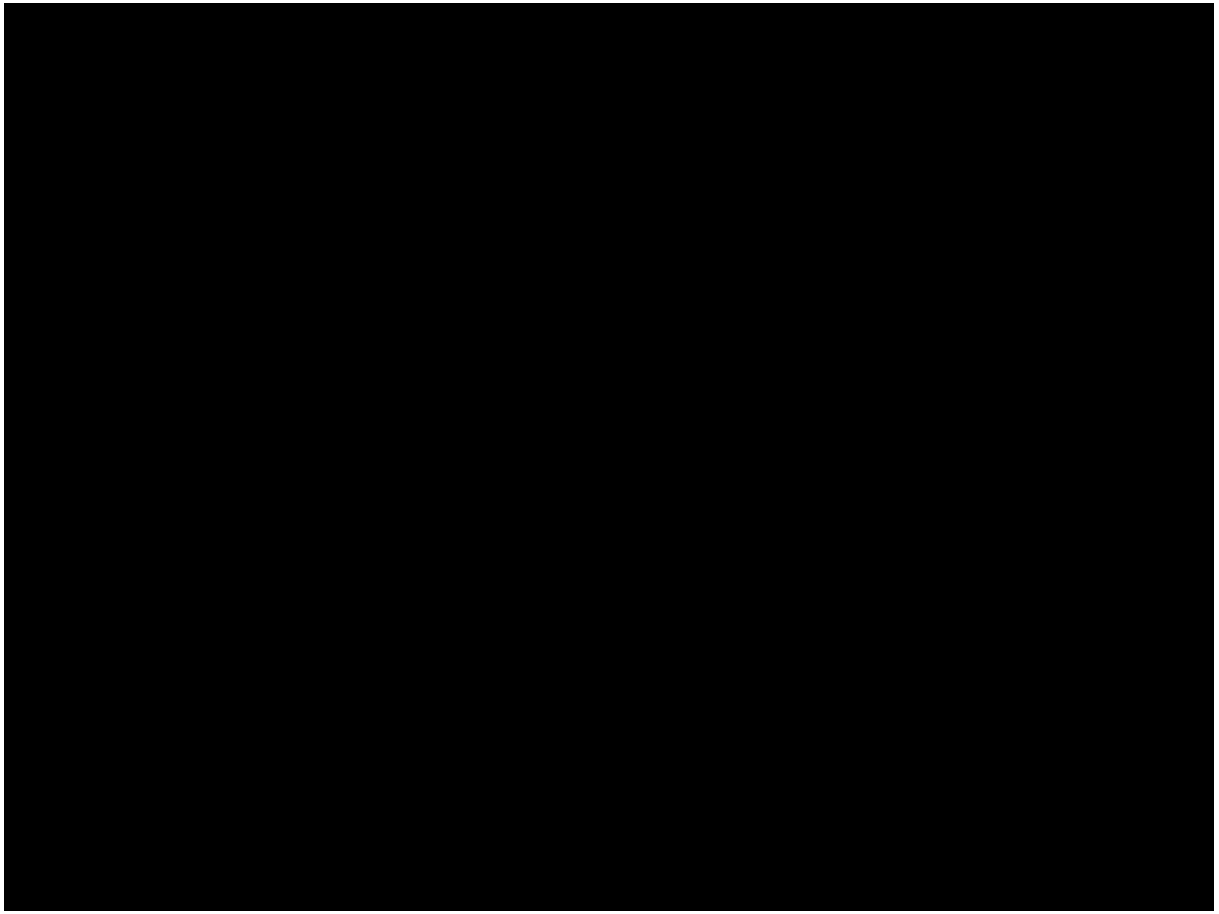
1．スイッチの片足側を、電源（５V）へつなげる。
2．スイッチのもう片足側は抵抗を挟み、入力するPINとなるPIN7へつなげる。
3．抵抗を通した先は、GNDへつなげる。

スケッチも少しだけ修正

```
#define LED_PIN 13
#define BTN_PIN 7
#define INTERVAL 1000

int value = LOW;

void setup() {
  pinMode(LED_PIN, OUTPUT);
  pinMode(BTN_PIN, INPUT);
}

void loop() {

  if (digitalRead(BTN_PIN) == HIGH || value == HIGH) {
    value = (value == HIGH ? LOW : HIGH);

    digitalWrite(LED_PIN, value);
    delay(INTERVAL);
  }
}
```

赤字が追加した箇所です。

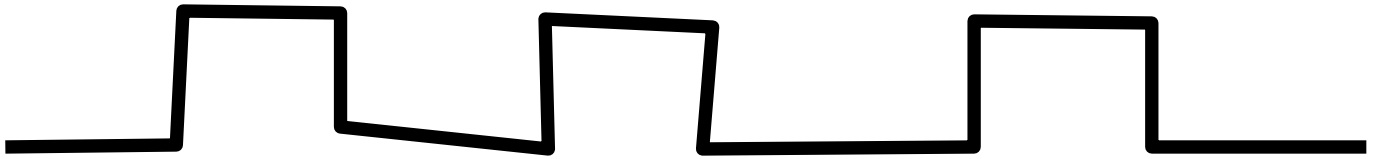
で、出来たのがこんな感じ



簡単でしょ？

今までの件は・・・

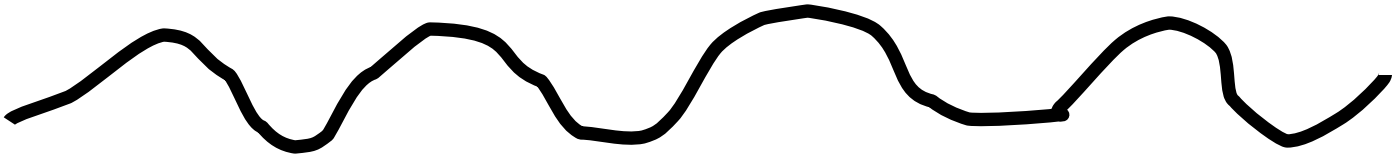
- 俗に言う、デジタル信号の入出力です。



  - こんなイメージです。
  - なので、LEDも付く・付かないのような制御。

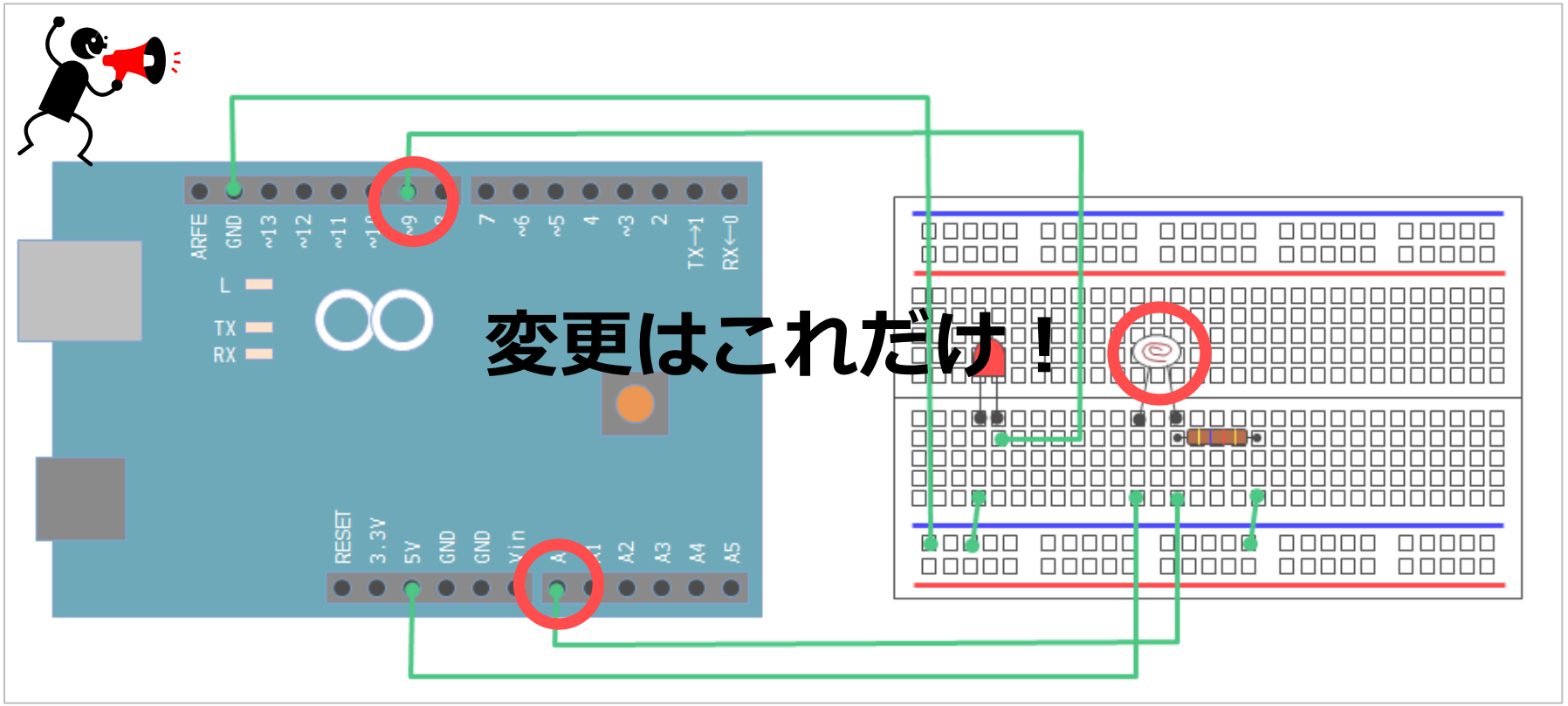
じゃあアナログ信号って？

- アナログ入出力のご紹介



  - こんなイメージです。
  - なので、LEDも強弱をつけて制御できる。

スイッチからセンサーへ！



1．LEDの出力先のPINは、PIN13からPIN9へ変更
2．スイッチをセンサーへ変更
3．センサーの入力PINを、PIN7からPINA0へ変更

ついでに解説

```
#define LED_PIN 9
#define SENSOR_PIN 0
#define INTERVAL 10

int value = 0;

void setup() {
  pinMode(LED_PIN, OUTPUT);
}

void loop() {
  value = analogRead(SENSOR_PIN);
  analogWrite(LED_PIN, value / 4);
  delay(INTERVAL);
}
```

PIN9をOUTPUTとして指定

センサーの値を取得
（アナログ入力）

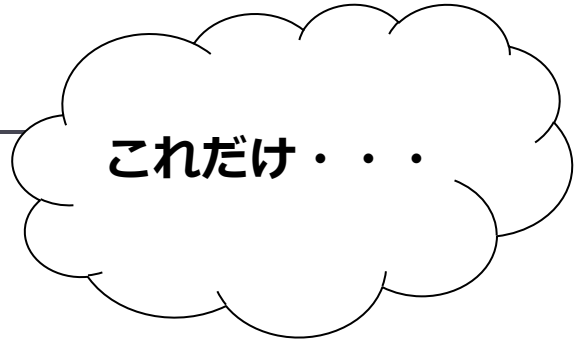
アナログ出力

※補足
analogWrite()でvalueを4で割る理由は、
入力は0～1024の値を取得するが、
出力は0～255の値までの為。

で、出来たのがこんな感じ

簡単でしょ？

この程度です！



- Arduinoが出来ること
  - デジタル入力
  - デジタル出力
  - アナログ入力
  - アナログ出力
  - プログラム実行

でもこれだけ出来ると・・・

こんなことも可能です。

発想次第で夢が広がる（かな）

そんなArduinoで遊ぶ為に・・・

次からは
少～しだけスキルアップ！？

こんなの使ってみます。

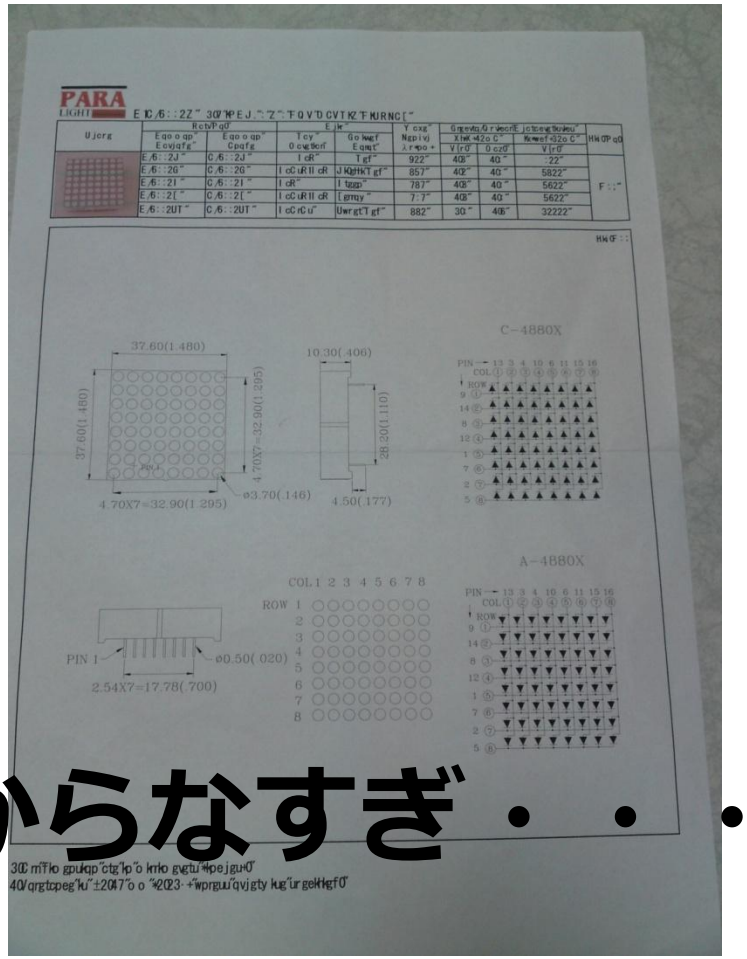
何が出来るかな？？？

電光掲示板！

の子供見たいなもの

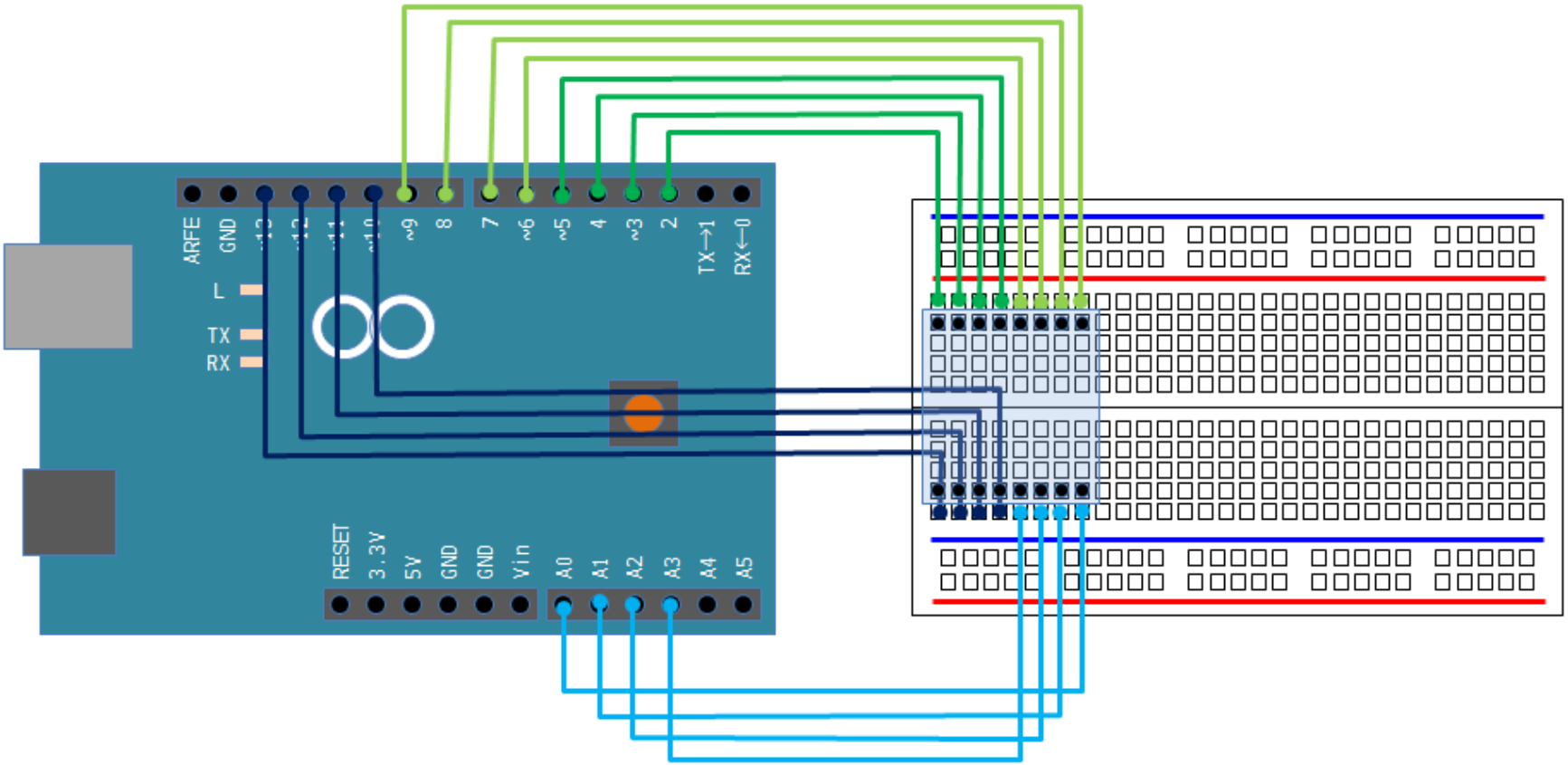
じゃあ早速つかってみましょう

まずはじめに見たのはこれ



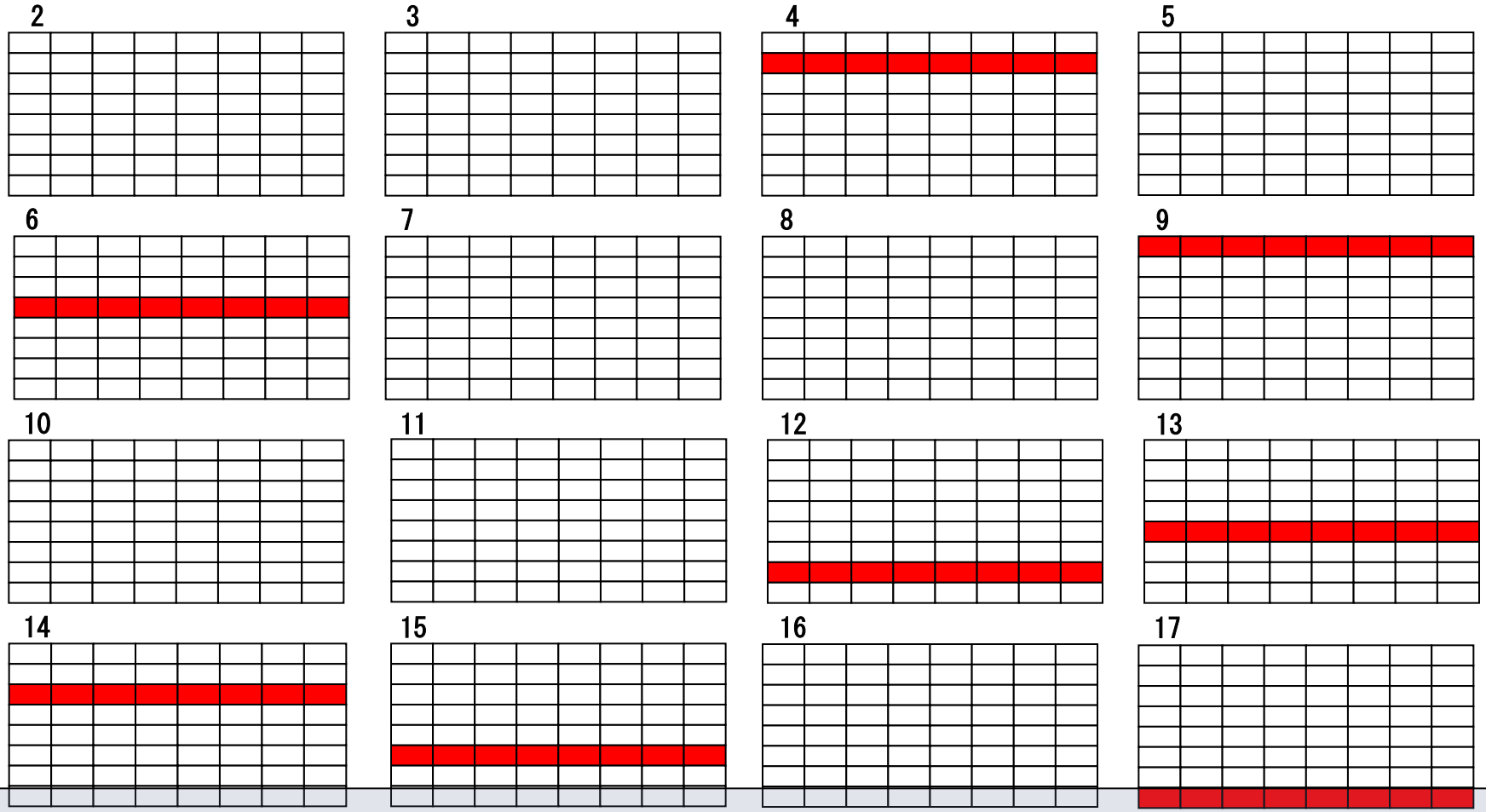
意味がわからなすぎ・・・（絶句）

わからないので・・・



PINとマトリクスのPINをつないでみましょう（汗）

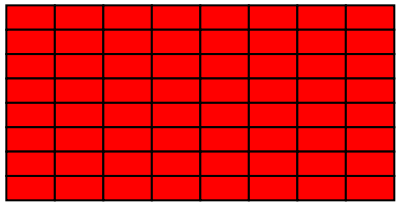
どうなるか確かめる



このPINに信号を送ると、この列がつくんだ！

ってことは！？

4,6,9,12,13,14,15,17のPINに信号を送ると、

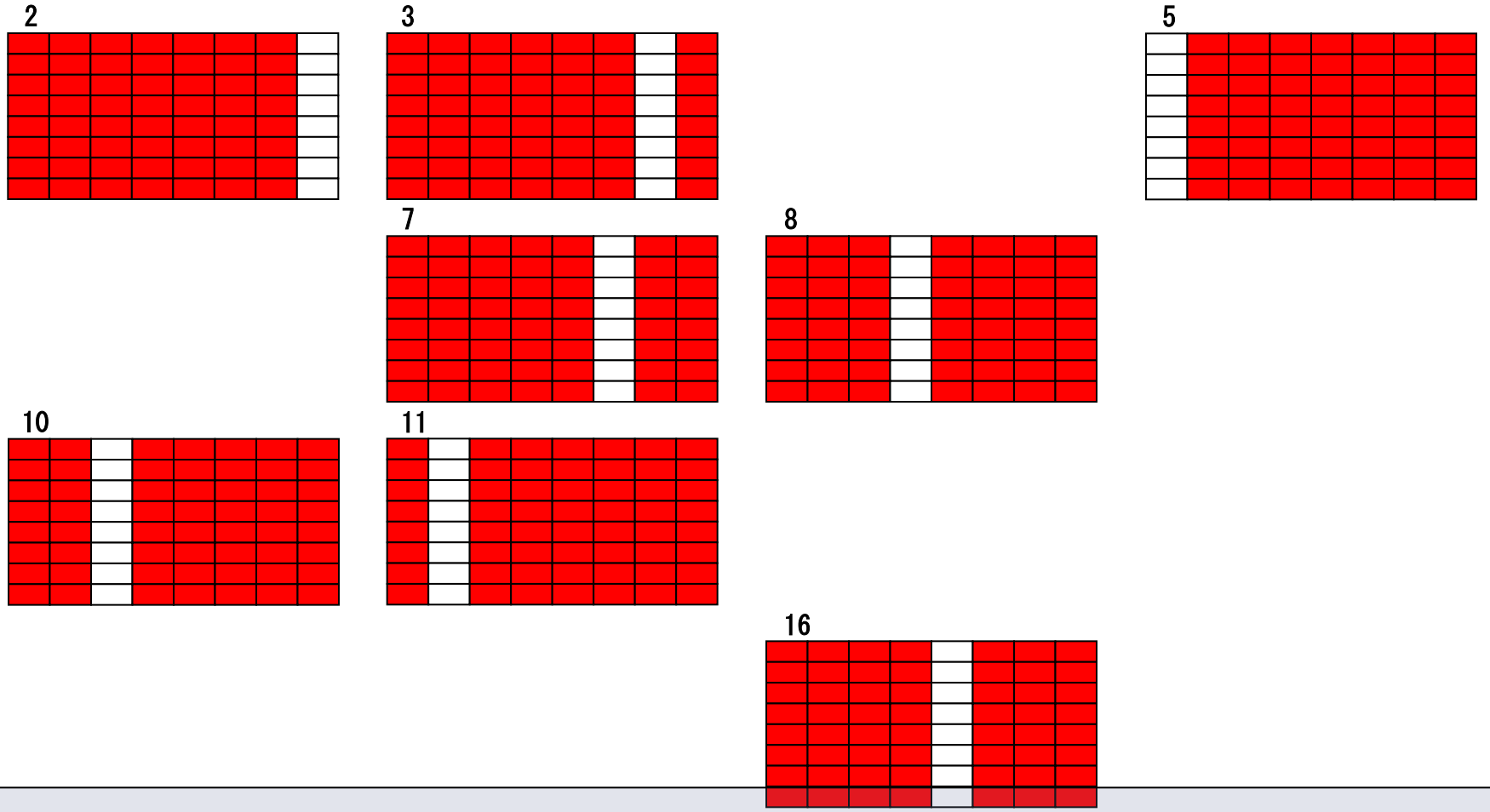


こうなるんだね！

では、

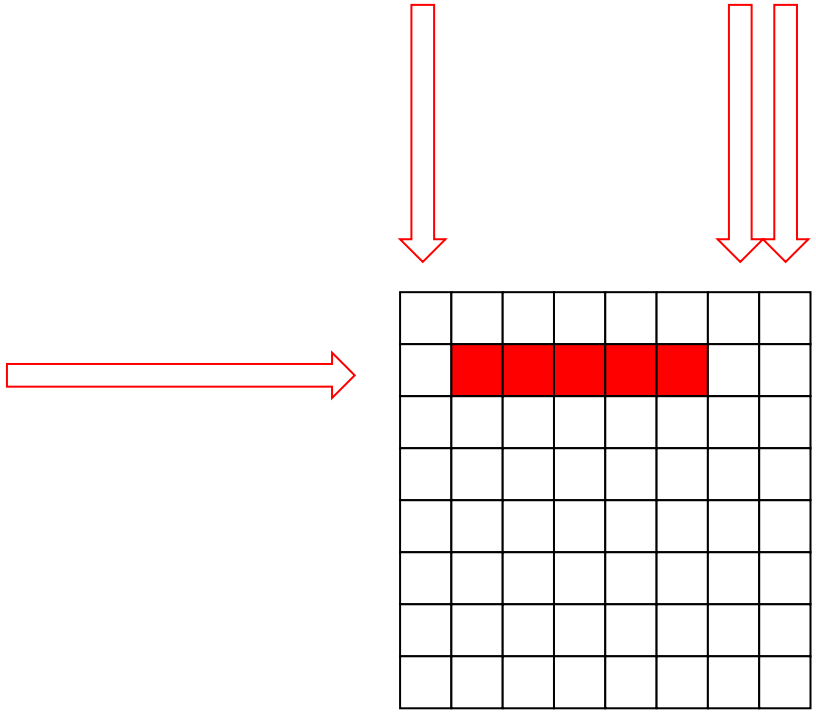
この状態で変化のなかったPINをもう一度試すと・・・

結果はこうなりました。

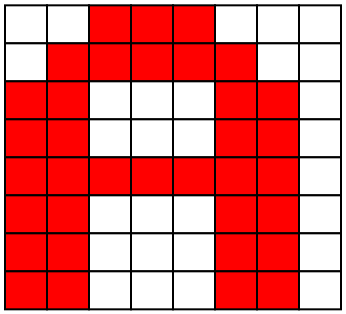


これでPINの役割がわかりました！

例えば・・・



では「A」を表示させたい場合

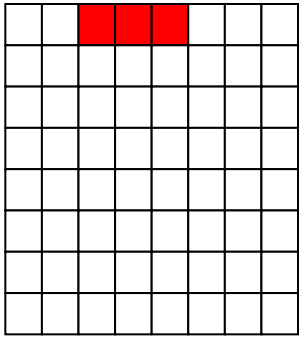


でも・・・

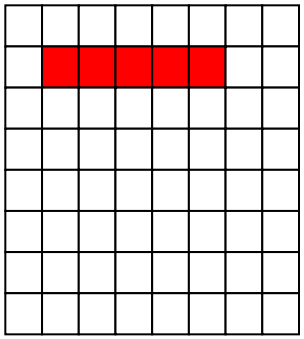
全部で16本のPINで、どう表現しているんだ？？？

実は「A」はこの集合体

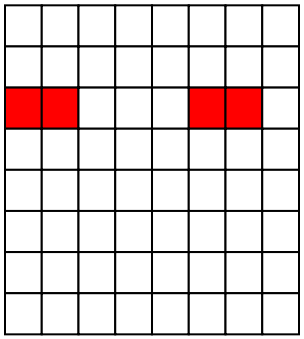
1列目



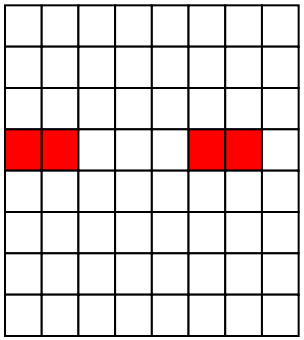
2列目



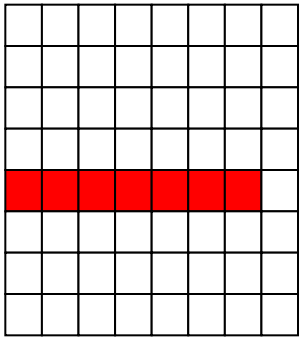
3列目



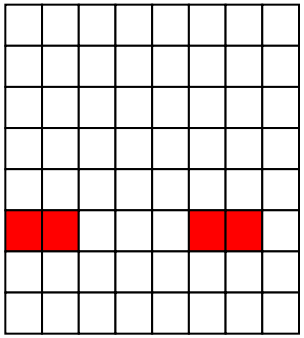
4列目



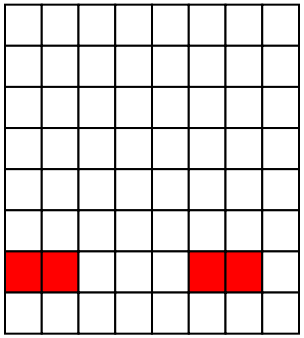
5列目



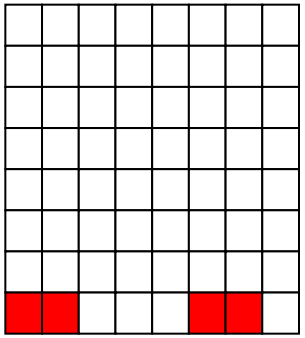
6列目



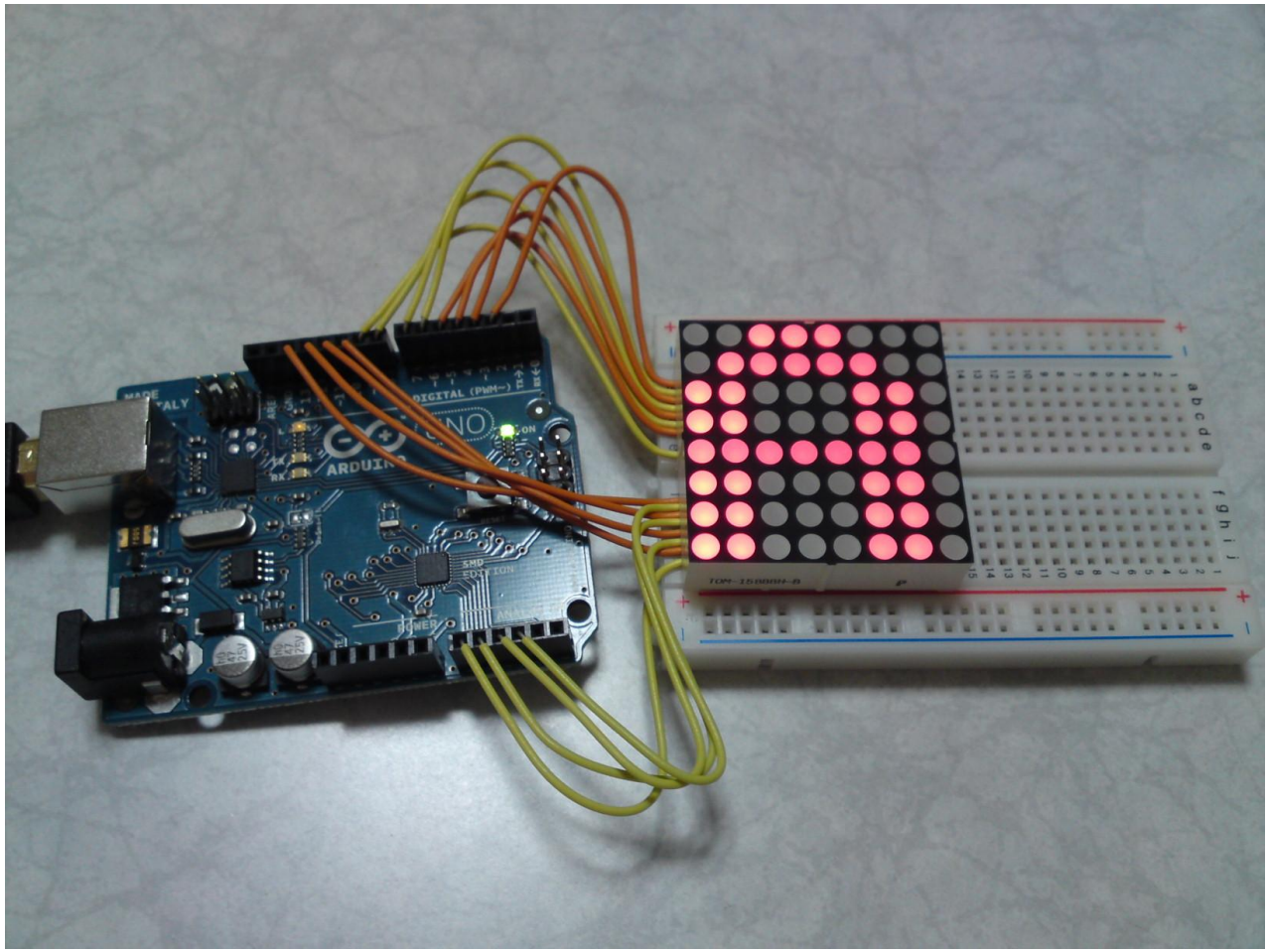
7列目



8列目



これを高速に連続させると



さらにこれを応用すると

じゃあ最後はOpentoneってことで

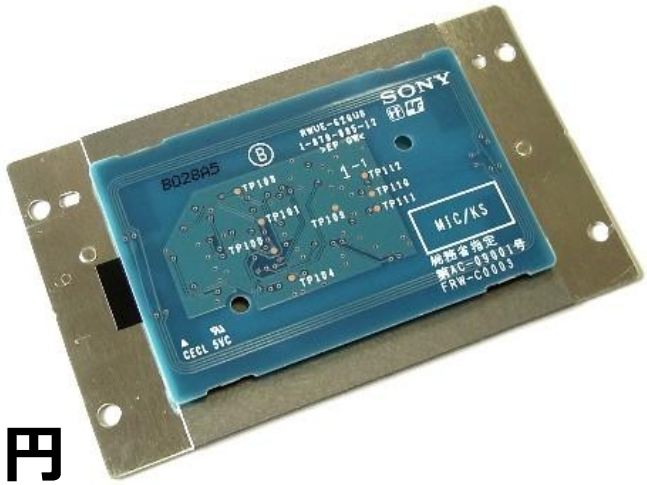
超お手軽に

Felicaを読んじゃう方法です！

必要なものは

- **RC-S620S**
  - **これ１つでFeliCaのリーダー**
  - **SONYさんからライブラリが提供されている**
    - **<u>これが超お手軽な最大の理由！</u>**

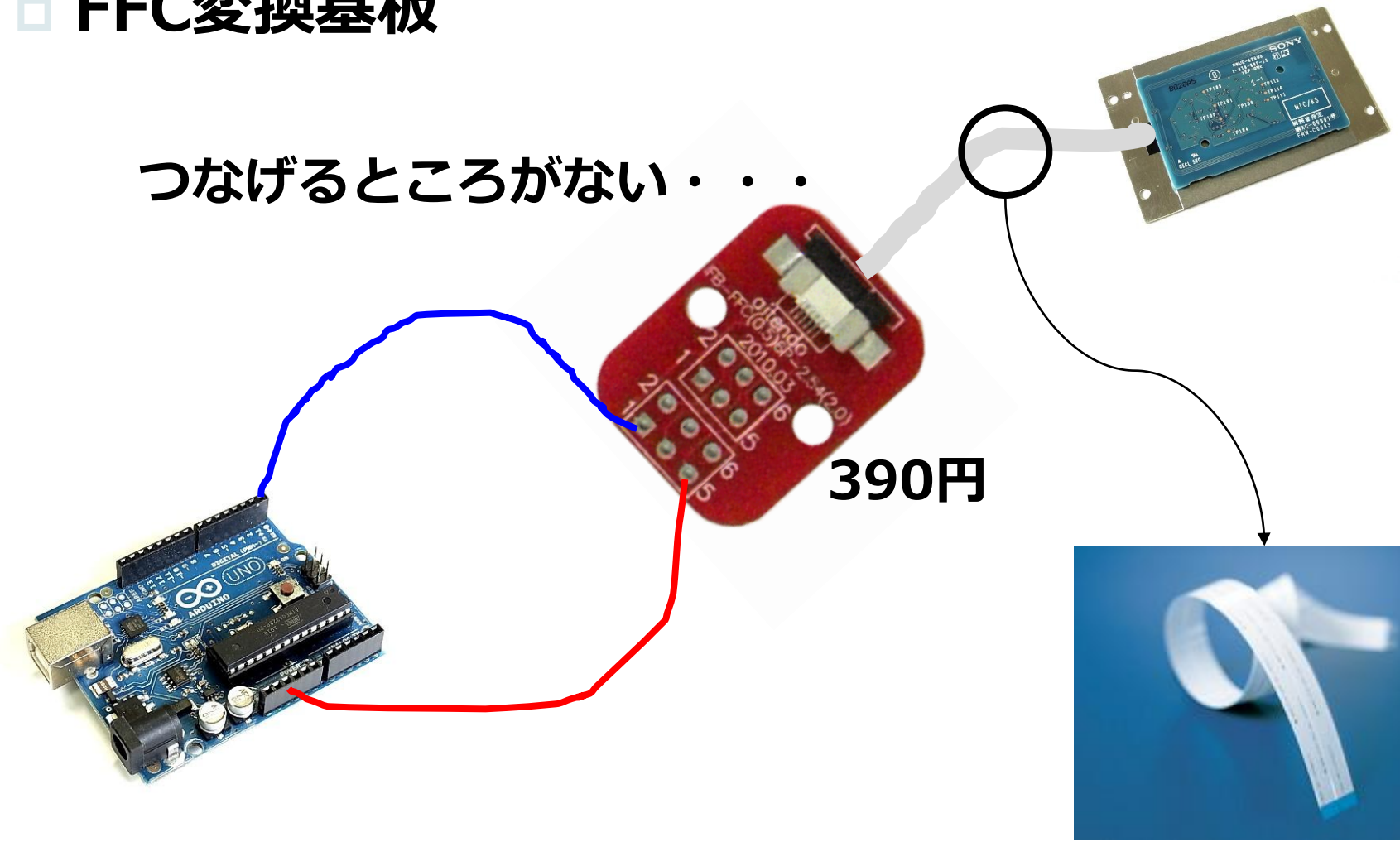
これと一緒



3,500円

実は必要なものがもう一つ

- **FFC変換基板**

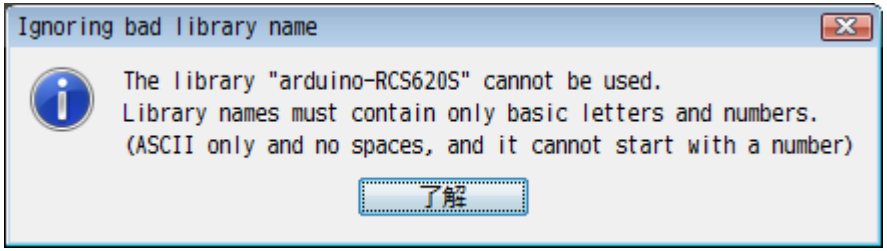


ライブラリをダウンロード

- Arduino向けRC-S620/S制御ライブラリ
  - http://blog.felicalauncher.com/sdk_for_air/?page_id=2699

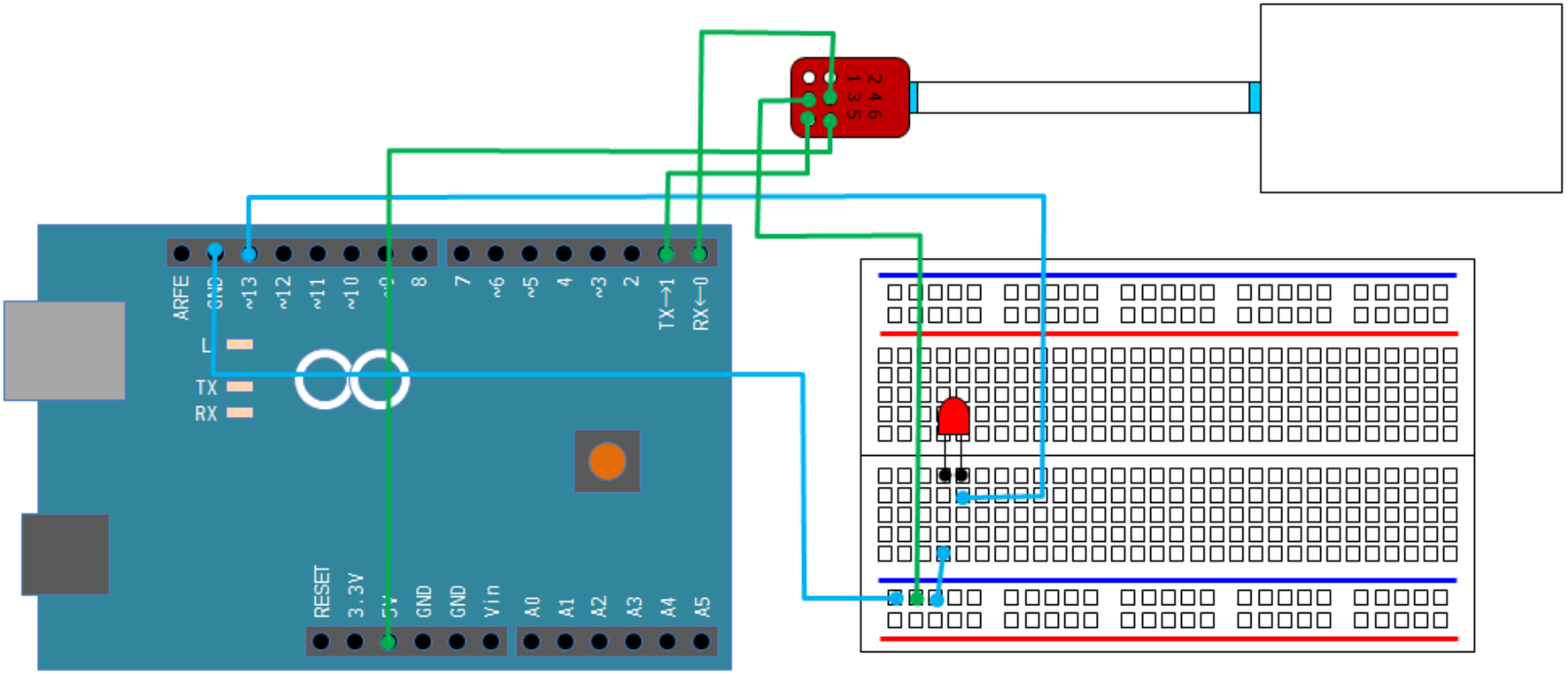
ライブラリ設定

- ダウンロードしたarduino-RCS620S.zipを解凍。
- 解凍したフォルダを[RCS620S]にリネーム
  - フォルダ名が[Arduino-RCS620S]の場合、以下のようなメッセージがArduino IDE起動時に表示されます。



- Arduino IDEフォルダ内[libraries]フォルダへ[RCS620S]を移動

接続はこんな感じです。


2 4 6
1 3 5
ARFE
GND
~13
~12
~11
~10
8
7
~6
~5
4
~3
2
TX→1
RX←0
TX
RX
RESET
3.3V
GND
GND
Vin
A0
A1
A2
A3
A4
A5

サンプルプログラムの実行

あ、もう終わりです・・・

FeliCaを読んでるかどうかわからん・・・

ちゃんと読み込んだときに

IDm(Felica製造ID)

PMm(Felica製造パラメータ)

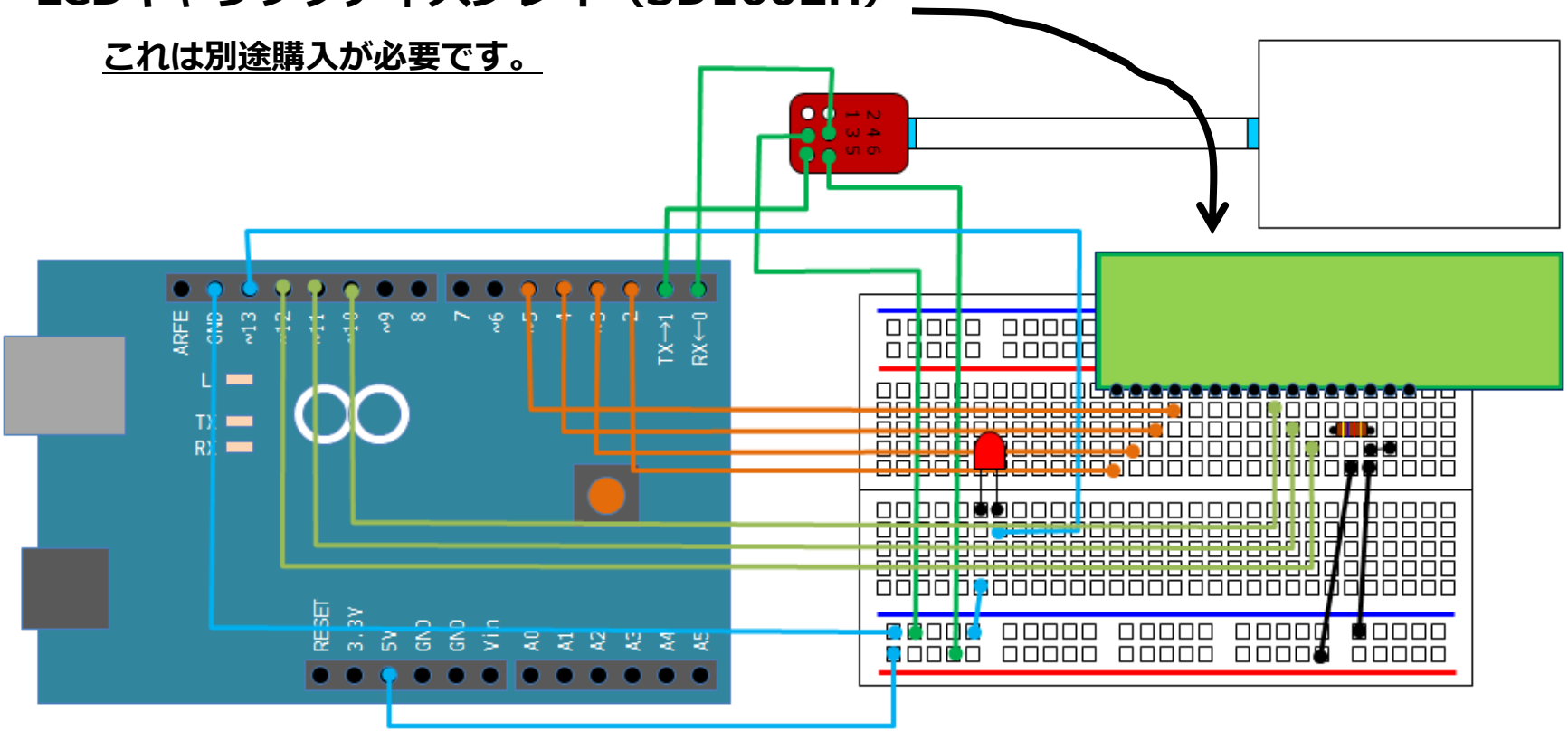
取得しています！

液晶ディスプレイに表示させて確認します

ちょっと複雑ですが

LCDキャラクタディスプレイ（SD1602H）

これは別途購入が必要です。



スケッチはexampleの中で、Felicaを読み込んだタイミングでLCDキャラクタディスプレイに書き込む処理を追加します。

と言われても？？？ですよね

- **簡単に説明します！**
  - キャラクタディスプレイ用ライブラリ（LiquidCrystal.h）をinclude

    #include<LiquidCrystal.h>
  - ライブラリの初期化（グローバルで）

    LiquidCrystal lcd(8, 7, 6, 5, 4, 3, 2);

    PINを指定しています。
    （これは前頁の接続の場合です）
  - LCDの初期化と表示

    lcd.begin(2, 16);

    2列16桁って意味です。

    lcd.clear();

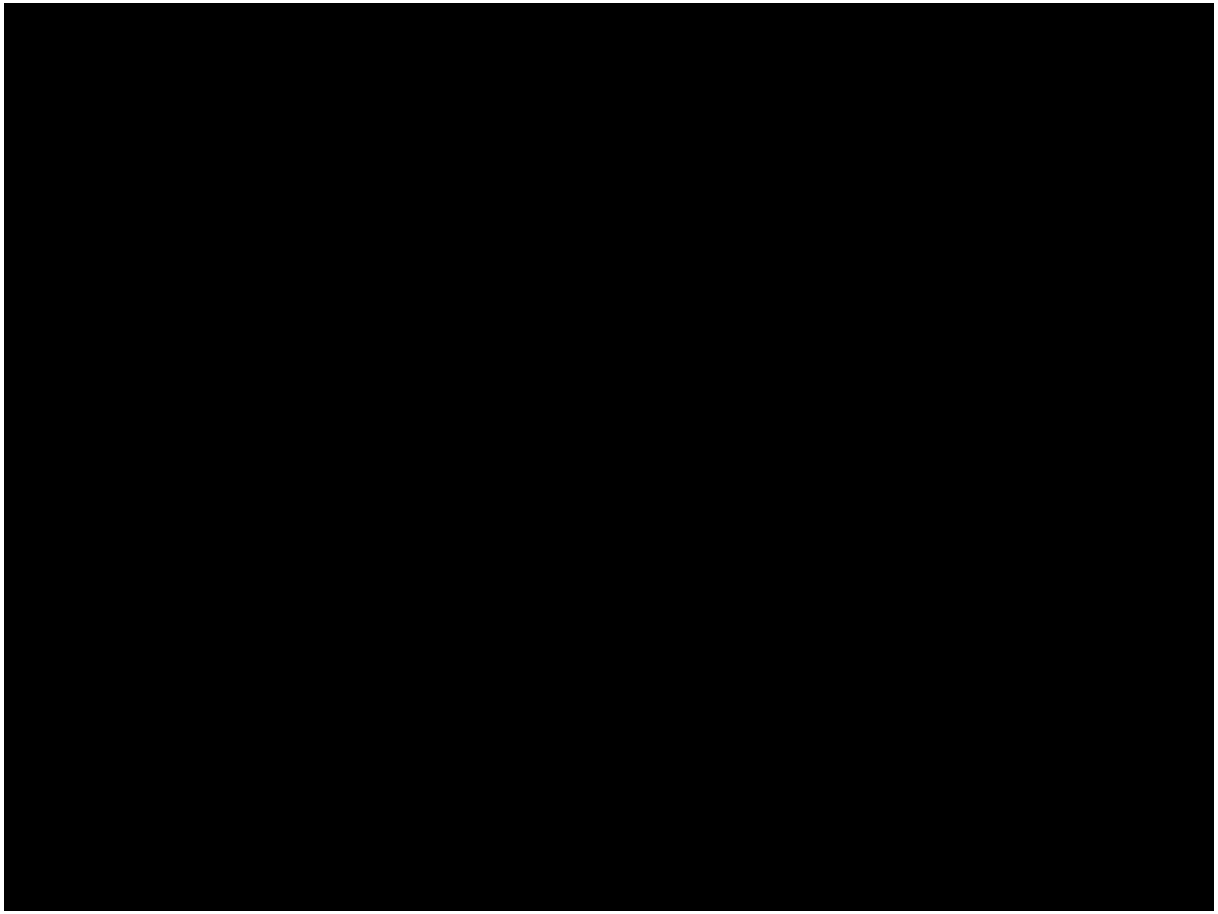
    lcd.setCursor(0, 1);

    引数１、桁番号（0が1番目）
    引数２、列番号（0が1番目）

    lcd.write(0x41);

    表示する文字
    0x41は「A」
  - こんな感じです。
  - 簡単でしたよね？

で、出来たのがこんな感じ



よりお手軽に始めるために

お店によってはエントリーキットなどが販売されていて、

何をしていいのかわからない人でもとっつきやすいかも！

これは若松通商さんのエントリーキットです。

その他のArduino？

- Arduinoはオープンソース

Seeeduino

Netduino

Seeeduino Film

.NET Micro Frameworkを使ってソフトウェアを開発！

最後に

- 日本語リファレンスもある
  - http://www.musashinodenpa.com/arduino/ref/
- 専門家でなくても簡単に実現可能
- 何かを実現するためのインタフェース
- そして、自慢する為に！

ご静聴ありがとうございました。