

# 理科教育におけるマイコンボード Arduino の活用例

北海道岩見沢農業高等学校  
高木伸雄（時間講師）  
中村拓仁（教諭）  
渡井陽子（実習助手）

Arduino（アルドゥイーノ）は 2005 年にイタリアで教育用に開発されたハード・ソフト共にオープンソースのマイコンボードである。最近多くの利用例や関連するプログラムが公開されている。PC と USB 接続して使用するのが基本である。

ArduinoIDE や PC 側で操作・表示等によく使われるプログラミング言語 Processing も Linux や MacOS にも対応している。今回は Windows で行った。

Arduino と PC を接続して様々な現象を視覚化・数値化することができる。さらに、PC 画面をプロジェクターや TV で拡大映写することにより授業の効果を上げることができる。

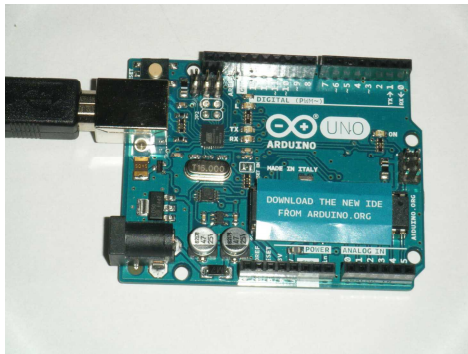
今回実際に行った実験は次に示すものである。

- ①九州工業大学情報工学部で開発した簡易オシロスコープの基板作成
- ②コンデンサー・抵抗回路の放電時電圧変化測定（簡易オシロスコープ使用）
- ③フトリフレクターを使った脈波の検出（簡易オシロスコープ使用）
- ④コンデンサーと抵抗の直列回路に交流を流したときのコンデンサーと抵抗の電圧の位相差測定（簡易オシロスコープ、反転増幅回路・ボルテージフォロア回路使用）
- ⑤ポケットガイガーによる放射線の測定・データの蓄積
- ⑥ pH メーターによる連続測定及び滴定曲線グラフのリアルタイム表示
- ⑦赤外線リモコンの信号パターン表示
- ⑧温度センサーを使った温度の連続測定
- ⑨気圧センサーを使った気圧の測定
- ⑩通過型フォトインタラプタ（フォトセンサ）を使った重力加速度測定

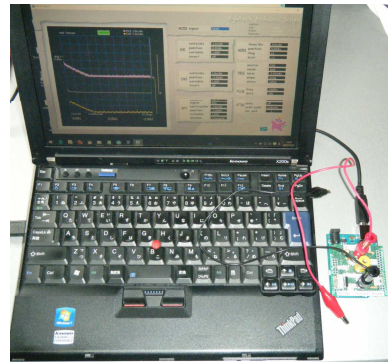
## 1. 簡易オシロスコープ

九州工業大学情報工学部 [https://www.iizuka.kyutech.ac.jp/faculty/physicalcomputing/pc\\_kitscope/](https://www.iizuka.kyutech.ac.jp/faculty/physicalcomputing/pc_kitscope/) が開発した、Arduino と PC を接続して簡易オシロスコープにする大変素晴らしいソフトウェア（Arduino 用及び PC 用）と簡単な回路図が上記ホームページに公開されている。マイコンは Arduino Uno（秋月電子通商で 2940 円）を使用した。

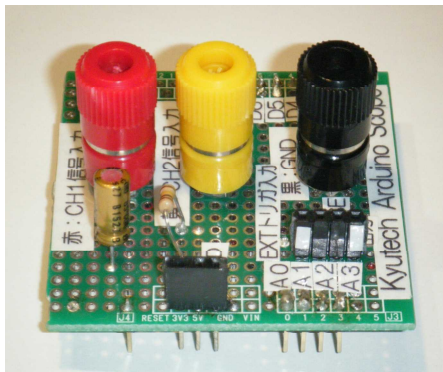
Arduino 用シールド基板に電子部品を載せて配線し半田付けをした。更に、基板を Arduino に載せて接続できるようにいくつかのピンを基板下面に付けた。このようにすると毎回回路を組む必要がなく、いつでも利用することができるので大変便利である。更に、プローブとして両端がバナナチップとミノムシクリップの線を 3 本作った。



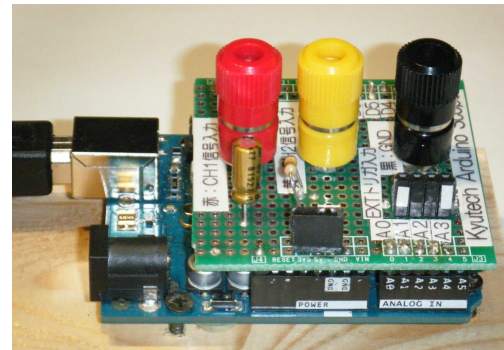
Arduino Uno



PC に USB 接続した Arduino



簡易オシロスコープ用基板



Arduino に接続した基板

以下、上記の簡易オシロスコープを使用した 3 つの測定例を示す。

## 2. CR 回路の放電時の電圧変化

以下の回路を作り、放電時の電圧を測定した。

電源電圧  $E = 3 \text{ V}$

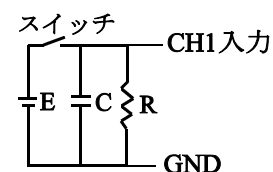
電気抵抗  $R = 100 \text{ k} \Omega$

コンデンサの電気容量  $C = 1 \mu \text{ F}$

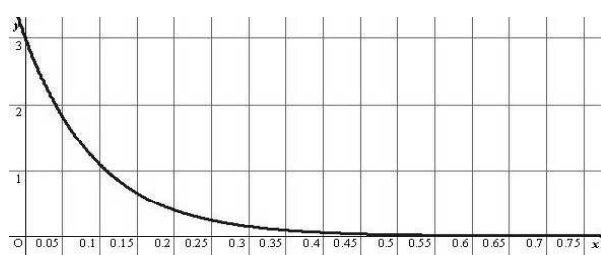
理論値は時間  $t$  に対して

電圧  $V = 3 \times e^{(-10t)}$  になる。

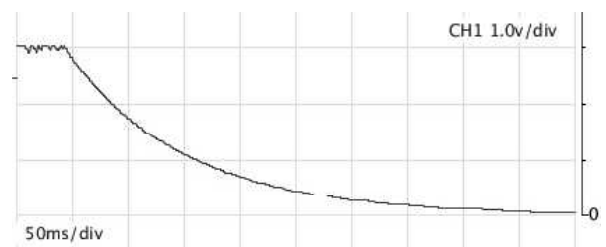
これをグラフ表示ソフト GRAPSE で描いたグラフと測定グラフ（横軸は 1 目盛 50ms）を以下に示す。ほぼ同じであることが分かる。



CR 回路



GRAPSE によるグラフ



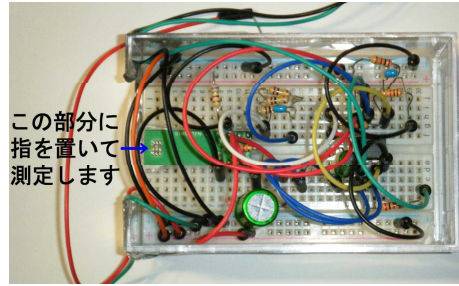
CR 回路放電電圧

## 3. フォトリフレクターを使った脈波の検出

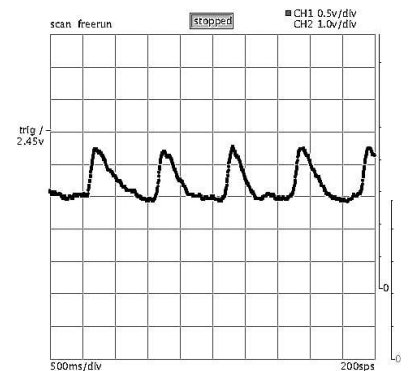
センサー部分には「NJL5501R 搭載 パルスオキシメータ用・反射型センサーモジュール」（秋月で 350 円）を使用した。

回路は秋月の上記モジュールのサイトに出ている「NJL5501R の使用例」の回路を使った。写真は部品を載せて配線したブレッドボードである。

これを Arduino に接続し、簡易デジタルオシロスコープで観察すると図に示す波形が得られた。脈拍は毎分 60 回ほどと読み取れる（横軸は 1 目盛 500ms）。



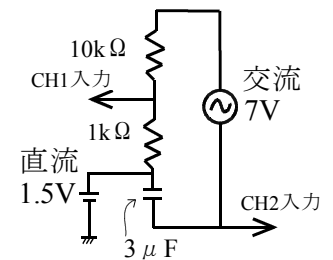
ブレッドボード上に部品組立



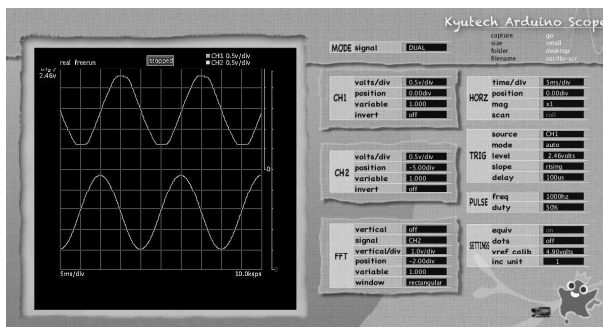
脈波のグラフ（白黒反転）

#### 4. CR 直列接続交流回路における C と R の電圧の位相差

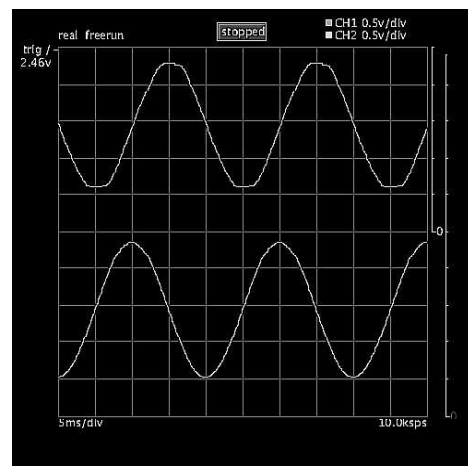
簡易オシロスコープは入力電圧が 0 ～ 5 ボルト（電源電圧）であるため、1.5V の乾電池でバイアスをかけ電圧が負にならないようにした。また、2つの入力チャンネルを同時に使うために CH1 と CH2 は逆位相になっている。これらを考慮して測定グラフを見ると、コンデンサー（CH2, 下の波）の電圧の方が抵抗（CH1, 上の波）の電圧より  $90^\circ$  遅れていることが分かる。



CR 直列接続交流回路

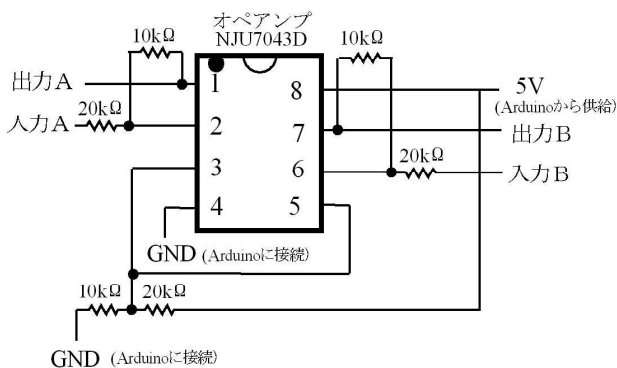


簡易オシロスコープ全体画面



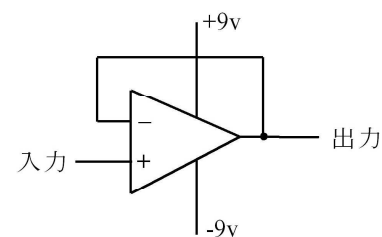
簡易オシロスコープグラフ画面

上記の回路では汎用性に欠ける面があるので、次に示す回路を作成して実験した。

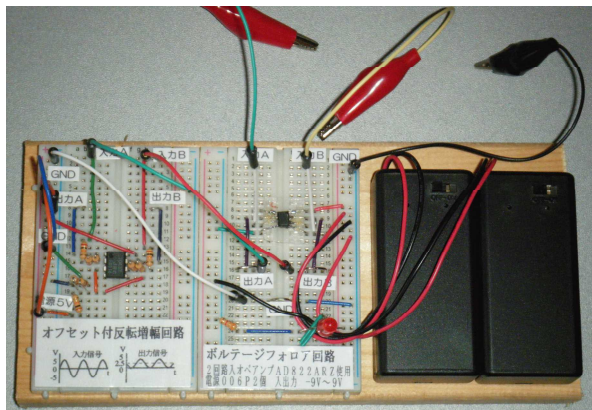


2回路入りオペアンプを使って、-5V～5Vを5V～0Vに変換する回路（オフセット付反転増幅）

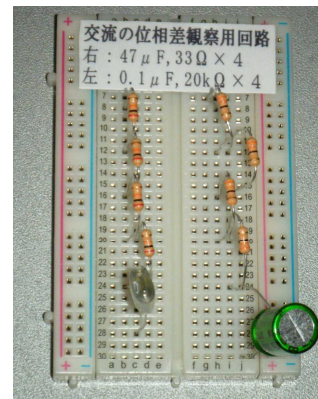
左の回路に両電源対応のオペアンプを使ってボルテージフォロア回路を接続した。



両電源オペアンプを使ったボルテージフォロア

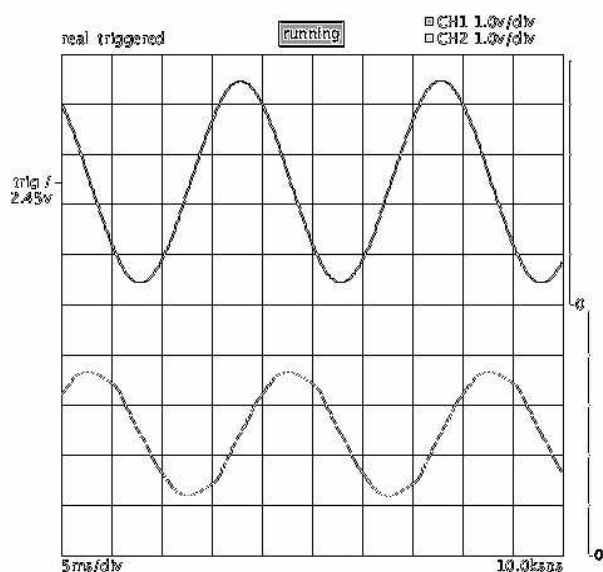


オフセット付き反転増幅回路+ボルテージフォロア回路

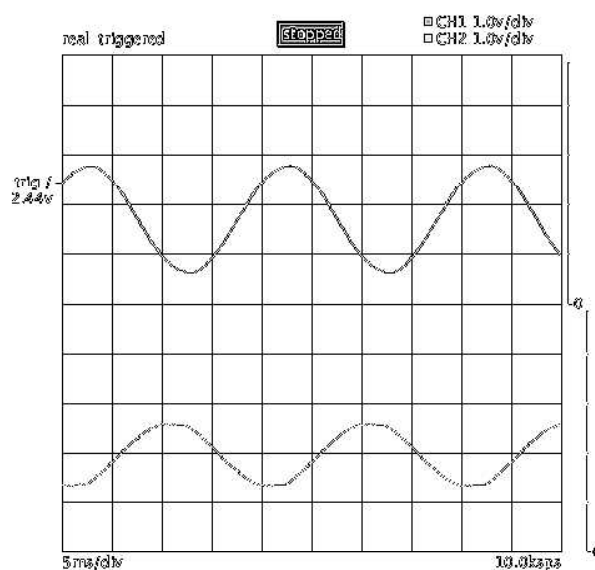


CR 回路 2 種類

リアクタンスが大きな場合 ( $0.1 \mu F$ 、 $40k \Omega$  使用) の位相のずれの測定を示す。  
CH1 (上) はコンデンサーの電圧、CH2 (下) は電気抵抗の電圧



ボルテージフォロアあり



ボルテージフォロアなし

(写真は白黒反転して見やすくしてある)

ボルテージフォロアがあると位相差は  $90$  度であるが、ないと大きくずれる。また、電圧も本来の値よりも小さく計測される。

このように、測定対象のリアクタンス等に左右されない測定を行うにはボルテージフォロア回路を入れるのが望ましい。

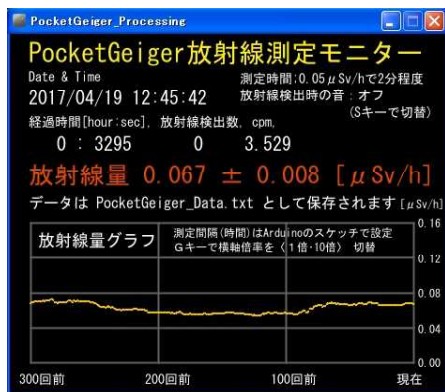
両電源回路のオペアンプは種類が少ないようであるが、2 回路入オペアンプ AD822ARZ SOIC (秋月で 300 円) に 006P 電池を 2 個使用して  $+9V$  から  $-9V$  に対応させた。

## 5. ポケットガイガーによる放射線の測定

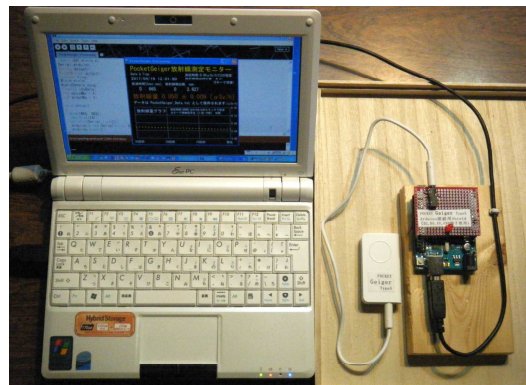
非営利プロジェクト「radiation-watch.org」(石巻市) が安価 (6450 円) で高性能な Pocket Geiger を開発・配布している。Arduino に接続して放射線を簡単に測定することができる「Pocket Geiger Type5」である。PIN フォトダイオードを使用しており、小型軽量で使いやすい。

右の写真は Pocket Geiger をシールド基板にして Arduino に接続したものである。





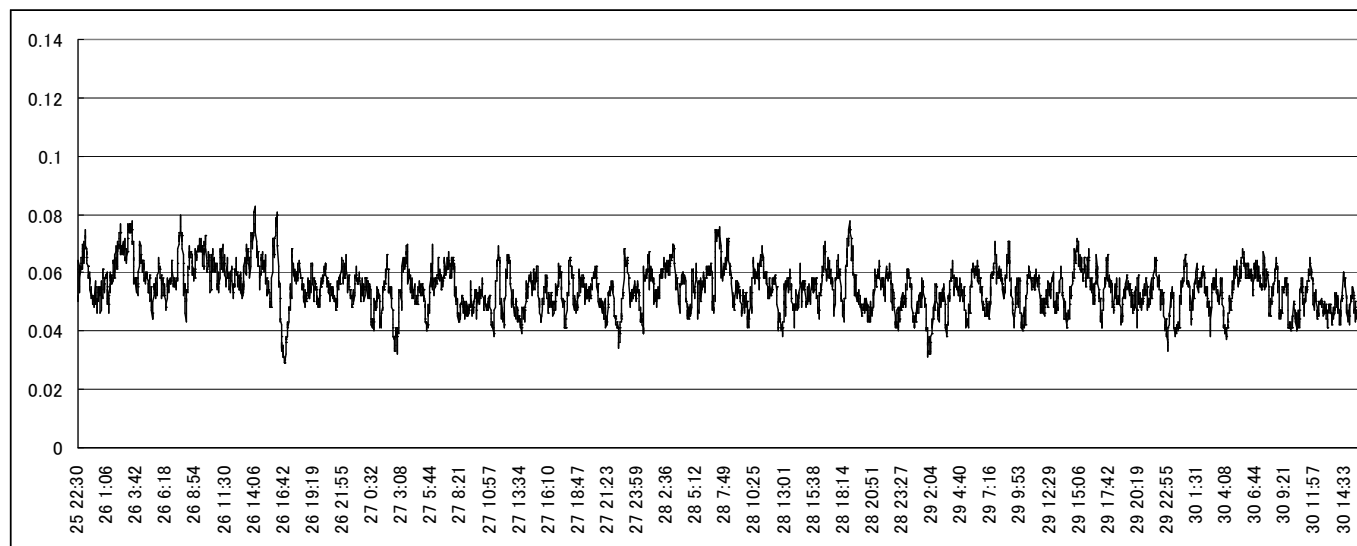
モニター画面



PC・Arduino・Pocket Geiger の接続

radiation-watch.org のホームページから Pocket Geiger Type5 の Arduino 用スケッチ及び PC でモニターできる Windows 用ソフトウェアをダウンロードして使うことができる。このスケッチを一部改変した。更に Processing を使ってデータを表示するモニター画面及び自動的に測定データが蓄積保存されるスケッチを作成した。下はそのデータ（1 分間隔）を Excel でグラフ化したものである。

（測定期間；2016 年 10 月 25 日 22 時 30 分～10 月 30 日 22 時 29 分の 5 日間）



縦軸；空間放射線量率（線量当量率）（ $\mu$  Sv/h） 横軸；経過時間（分）

## 6. pH メーターによる連続測定及び滴定曲線グラフのリアルタイム表示

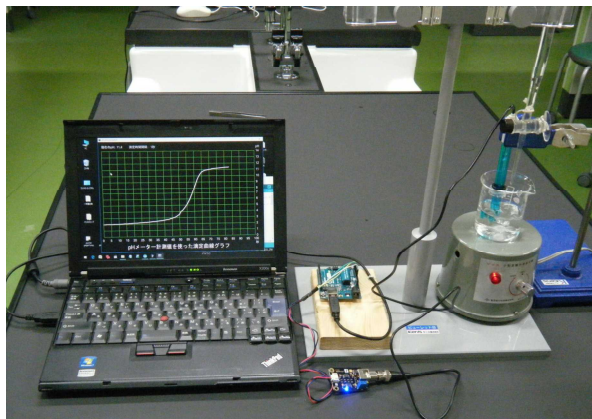
Gravity: Analog pH Sensor / Meter Kit For Arduino（8 千円ほど）を使って滴定曲線を表示していくソフトウェアを Processing を使って作成した。Arduino のスケッチはホームページからダウンロードしたものを改造して使用した。

実験は、ビーカーに入れた 0.01mol/L の塩酸 50ml に 0.1mol/L の水酸化ナトリウムをビュレットを使って毎秒 1.5 滴ほどの割合で滴下した。液はスターラーで攪拌した。（資料フォルダにある実験動画参照）

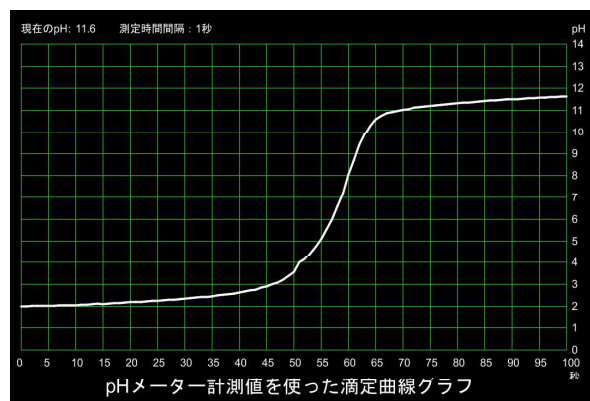
酸とアルカリの中和滴定実験を生徒の目の前で行い、その pH の変化していく状況を同時進行で表示することができるので、pH の理解等に役に立つ。

塩酸にフェノールフタレインを 1 滴入れておくと中和点付近で液が赤紫色になり、視覚効果が増した。

画面の保存、測定時間（60 秒、100 秒、200 秒）の変更、リセットを、割り当てられたキーボードのキーを押すことによって簡単に実行できるようにした。



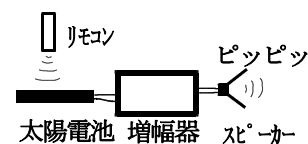
実験装置



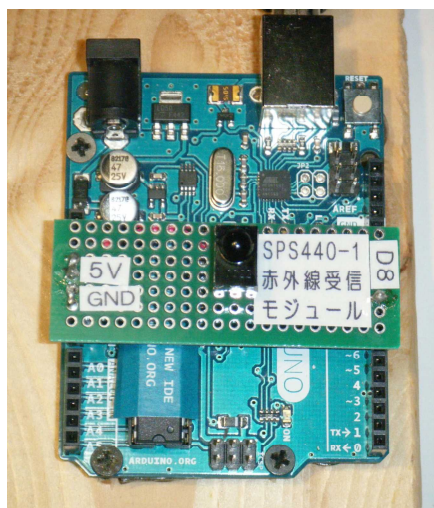
滴定曲線グラフ

## 7. 赤外線リモコンの信号パターン表示

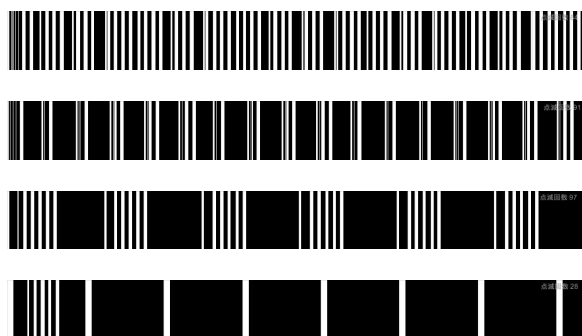
太陽電池、増幅器、スピーカーを組み合わせ、太陽電池にリモコンから赤外線を当てるとピッピッと音が出る。リモコンの別のボタンを押すと、出ている信号は違うはずであるが耳で聞いても分かりづらい。別の機器のリモコンの信号と比べると音の違いを感じやすい。



これらの違いを表示するために Arduino と赤外線受信モジュール（SPS440-1、秋月で 50 円）を使って装置を作った。さらに、Arduino 用スケッチ（プログラム）と Processing 用スケッチを作成した。リモコンの赤外線から出ている信号を可視化して簡単に示すことができるようになった。



Arduino に載せた赤外線受信基板



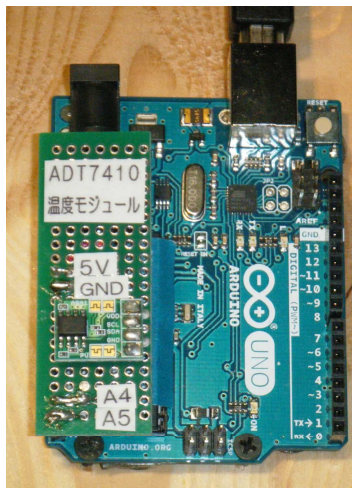
リモコン信号のパターン

上から順にビデオレコーダーのボタン、別のボタン、テレビのボタン、DVD プレイヤーのボタン

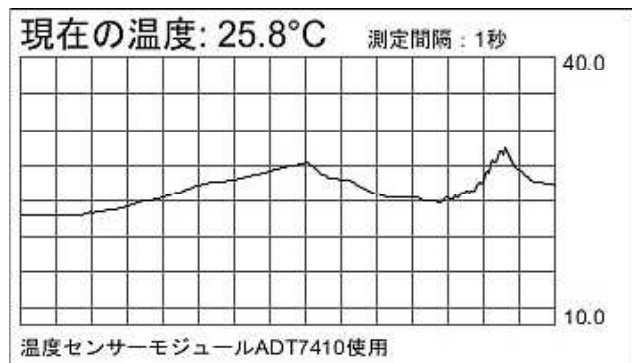
## 8. 温度センサーを使った温度の連続測定

基板にデジタル温度センサー「ADT7410」（秋月で 500 円）と Arduino に接続するピンを半田付けする。この基板を Arduino に載せ PC に接続する。温度センサーと I2C 通信を行うプログラムを PC から Arduino に書き込む。センサーからのデータを Processing で作成したプログラムを使って PC 上でグラフ化する。測定時間間隔は自由に変えることができる。

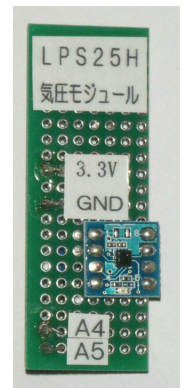
一定温度を超えるとメッセージを発する等、様々な使い道がある。



Arduino に載せた温度モジュール基板



温度センサー測定グラフ  
(センサー部分に手を当てたり、  
息を吹きかけて温度を上げた)



気圧モジュール基板

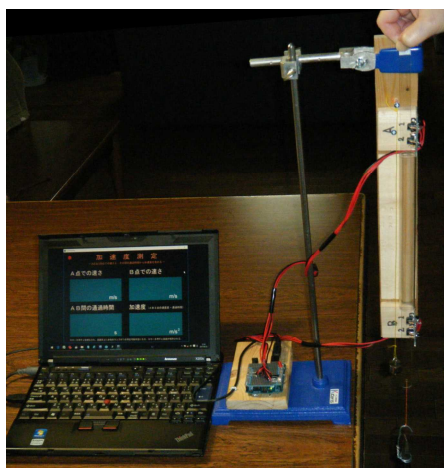
## 9. 気圧センサーを使った気圧の測定

上記の温度センサーと同じように、気圧センサーモジュール「LPS25H」（秋月で 600 円）を使って気圧を連続測定することができる。写真はセンサーモジュールを基板に半田付けしたものである。基板下面に Arduino と接続する 4 本のピンを半田付けしてあり、簡単に使用することができる。

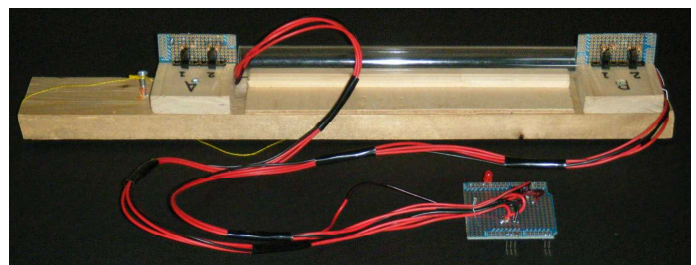
## 10. 通過型フォトインタラプタ（フォトセンサ）を使った重力加速度の測定

通過型フォトインタラプタ（フォトセンサ）CNZ1023（1 個 20 円ほど）を 4 つ Arduino に接続し、「ビースピ」と同じような速度測定機能にタイマー機能を付け足して、 $\text{加速度} = \text{速度変化量} \div \text{時間}$  という原理的な式で加速度を求めることができるようになる。このようにすると PC を使って目の前ですぐに測定値を表示することができる。

実際に授業で使用してみた。PC 画面を大きな TV に出して、目の前ですぐに上下のセンサーを通過したときの速度、センサー間を通過した時間、それらを元に計算した重力加速度の値が表示される。この仕組みは生徒の理解を随分進めた。



装置全体

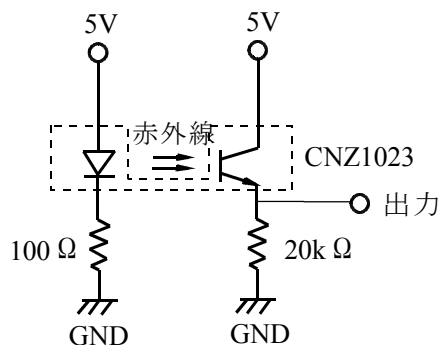


板に付けた測定部分

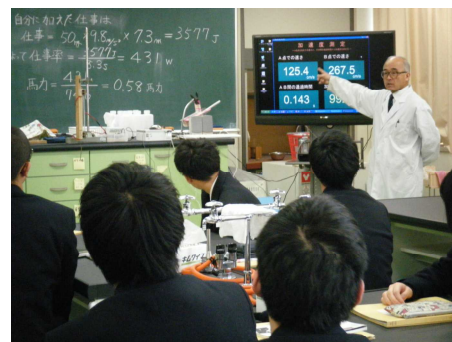


PC の測定値表示画面





CNZ1023 の配線図



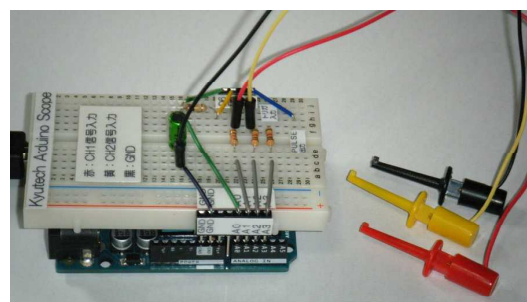
授業で使用

このように、マイコンボード **Arduino** を使って、少しの工夫と努力を行うことによって理科教育の中でいろいろな測定を行うことができる。**Arduino** を利用することにより教育効果を高めていく可能性が大きくなると思われるので、多くの学校等で使われることを希望している。

## 【補足】

### 1. 簡易オシロスコープ用簡易シールド基板（ブレッドボード利用簡易版）

2 頁に掲載してある「簡易オシロスコープ用基板」はユニバーサル基板を使って部品を配置し、配線をして半田付けしているが、作業は結構大変である。しかし、ブレッドボードに部品を載せて配線するのは簡単ではあるが、使うたびに **Arduino** と 8 本の接続コードで結線する必要がある。また、長いコードを使うと手などを引っかけて外してしまう危険性



Arduino に載せた簡易版

もある。そのような点を考慮して、半田付けなし・耐久性あり・簡単に **Arduino** に載せることができる「簡易版」を作成した。ピンソケットとブレッドボードは瞬間接着剤で接着した。強度は十分である。

使用部品；ブレッドボード、ジャンパーワイヤ、リード長 10mm のピンソケット、IC クリップ

参照したサイト；<http://d.hatena.ne.jp/OkibiWorksLabo/20100504/breadboard>

### 2. Arduino 及び Processing の設定

ダウンロードした **Arduino** 及び **Processing** のフォルダを PC のデスクトップ等適当な場所に置く。インストール作業は必要ない（PC のレジストリに書き込まれない。**arduino-1.6.4** にはインストールして使用するバージョンもある。安定している使用できるようにする。）一度実行ファイルをダブルクリックして正常に起動することを確認する。その後は各プログラム（スケッチという）の実行ファイルをダブルクリックすると起動する。

**Processing** で日本語の表示がうまくいかない場合は、以下を行う。

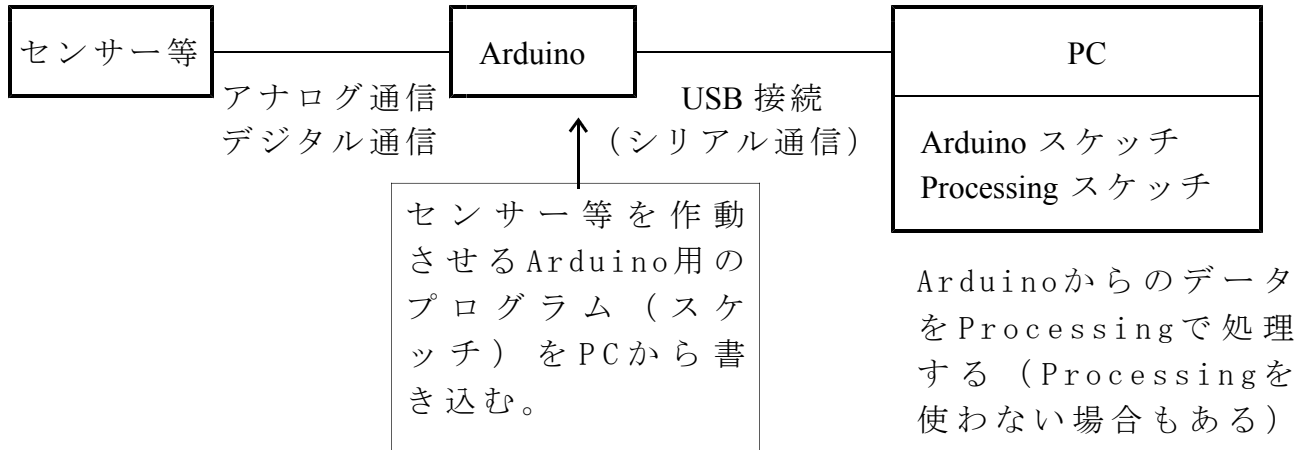
- **Processing** → 環境設定 (OS によっては File → Preferences)
- Editor and Console font を日本フォントにする

(<http://qiita.com/ayaniimi213/items/986c2b567e196029c4a5> 「**Processing 2.X** での日本語の扱い」等の関連サイトを見て設定を変える)



### 3. プログラムの設定・書き込み等の流れ

基本的な関係を以下の図に示す。



#### (1) Arduino

- ① Arduino を PC と USB 接続する
- ② センサー等を Arduino に接続する
- ③ Arduino のプログラム (スケッチ) を起動する
- ④ Arduino IDE の設定をする (マイコンボードの認識、シリアルポートの設定)  
設定の詳細は「Arduino の最初の一步」(下記関連サイト参照)の「3.3 ArduinoIDE の設定をする」に書いてあるので、これを参照して設定する。
- ⑤ 「コンパイル・書き込み」を実行する。

#### (2) Processing

- ① Processing のプログラム (スケッチ) を起動する。
- ② シリアル通信を行うポート (COM1, COM2 等) 及び通信速度を、Arduino で使用するポート及び通信速度と同じにする (自動設定になっているものもある)。  
(下記関連サイトの [physical\\_computing\\_03](#) に詳しく記載してあるので参照)

具体例

Processing スケッチ (void setup() 内で設定されている)

```
void setup() {
  size(1000,100);
  background(0,102,165);
  myPort = new Serial(this, "COM4", 9600); //ポートと通信速度
}
```

Arduino スケッチ (void setup() 内で設定されている)

```
void setup() {
  Serial.begin(9600); // パソコン (ArduinoIDE) とのシリアル通信速度
  pinMode(IRpin, INPUT);
}
```

## 4. 参考

ウェブサイト「Arduino のすすめ」(<http://n.mtng.org/ele/arduino/>) は詳しく書いてあるのでとても参考になる。Arduino の説明、回路やプログラム例、センサーの使い方、半田付けの仕方、参考になるウェブサイト、書籍等が網羅されている。

また、下記の「プログラミング演習Ⅱ フィジカルコンピューティング Arduino をはじめよう」のサイトはプログラミングについて詳しく書いてある。こちらもとても参考になる。

## 5. 関連サイト

### ① Arduino の最初の一步

(<http://n.mtng.org/ele/arduino/tutorial001.html>)

### ② 「プログラミング演習Ⅱ フィジカルコンピューティング Arduino をはじめよう」の3回分の演習ページ

([http://kougaku-navi.net/lecture/fms/archives/programming2/physical\\_computing\\_01.pdf](http://kougaku-navi.net/lecture/fms/archives/programming2/physical_computing_01.pdf))

([http://kougaku-navi.net/lecture/fms/archives/programming2/physical\\_computing\\_02.pdf](http://kougaku-navi.net/lecture/fms/archives/programming2/physical_computing_02.pdf))

([http://kougaku-navi.net/lecture/fms/archives/programming2/physical\\_computing\\_03.pdf](http://kougaku-navi.net/lecture/fms/archives/programming2/physical_computing_03.pdf))

### ③ 「IoT デバイス製作 Step by Step 第7回 デジタルセンサーを使ってみよう」

(<https://www.cloudsquare.info/topic/create-iot-device-vol7-using-digital-sensor>)

## 6. ソフトウェア

上記の実験を行う上で作成した processing 及び Arduino のスケッチ等を「資料」フォルダに入れてあります。自由に改変して、よりよい物ができることを希望しています。それ以外の物（例えば、Processing や九州工業大学の簡易オシロスコープ）は該当するホームページからダウンロードして使ってください。

## 7. 開発環境

ソフトウェアの開発環境は以下です。

PC の OS は Windows XP を使用した。

Arduino IDE は Arduino 1.6.4 を使用した。

Processing IDE は Processing 2.2.1 を使用した