

3

クックブック

physical × wonderfl で始めるフィジカルコンピューティング

■ 蛭田 直 (信州大学)

■ 秋田純一 (金沢大学)

はじめに

これまでの解説記事では、さまざまなマイコンツールキットの動かし方を説明しました。本稿はクックブックとして紹介したマイコンツールキットで応用できる利用例を Arduino と physical × wonderfl¹⁾ を用いて紹介いたします。

■ クックブックの環境について

クックブックでは、「physical × wonderfl」を PC とツールキットを組み合わせる方法として、よく使う入力と出力の装置の使い方を紹介します。

「physical × wonderfl」(図-1)は、国際情報科学芸術アカデミー (IAMAS) と面白法人カヤックの産学連携プロジェクトとして運営されています。

ActionScript をブラウザのみで制作・共有できる wonderfl と Funnel ライブラリを使用することで、ツールキットを使用したフィジカルコンピューティングを容易に実現することができます。

これまでハードルの高かったフィジカルコンピューティングは Arduino, Gainer, Phidgets, Aniomagic, mbed とさまざまなツールキットが登場したことでハードルは低くなってきました。こうしたツールキットを使用する場合は、ツールキットに書き込みを行いスタンドアロンでの使用を行う場合と、PC と接続して行う方法があります。スタンドアロンで行う場合は、ケーブルの取り回しなどが必要なくなりコンパクトになるため実際の使用を想



図-1 physical × wonderfl Web サイト

定して作業を行うことができます。一方で、ディスプレイやサウンドの使用を想定した制作の場合は、PC 側でこれらの処理を行った方が複雑な処理を行うなどのテストを素早く行うことができます。自分が制作したいものによりどちらの方法で行うかは変わってきますが、いきなりスタンドアロンで実装するよりもツールキットと PC の組合せでテストを行った上で、スタンドアロンにするにはどのツールキットが最適か?などを考慮して制作を行うと効率よく進めることができます。

■ wonderfl について

wonderfl は、ブラウザのみで ActionScript を実現できると同時にコードは公開されます。公開されたコードは、「Fork」することでオリジナルのコードをコピーして新たにコードを編集することができます。



図-2 physical × wonderfl コード実行画面

す(図-2)。この機能により参考となるコードの一部を隠したり改編することで深くコードを理解できたり、コードを引用することで自分のやりたかった機能を素早く実現することができます。

■ Arduino の準備

physical × wonderfl と Arduino を使用して、フィジカルコンピューティングを行うには funnel ライブラリを使用する必要があります。

また、Arduino のドライバも必要となるため funnel.cc (<http://funnel.cc/Main/GettingStarted/>) よりダウンロードを行い、ドライバと Funnel サーバの準備を行います。

Arduino ボードの接続には、Firmata (ファルマータ) を使用しますので、以下の手順でボードにアップロードを行ってください。

Step1. 「File」メニュー→「Examples」→「Firmata」→「Standard Firmata」を選択。

Step2. 「Upload」ボタンを押してアップロード。

■ 動作確認

wonderfl では、複数のネットサービスのアカウントを使用してログインすることができます。ログイン後はユーザ画面からサンプルコードの検索

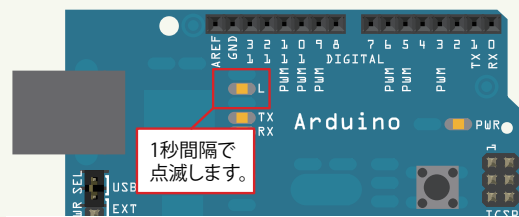


図-3 動作確認

などが行えます。また、wonderfl 内の physical × wonderfl 特設 Web ページよりサンプルコードへアクセスすることができます。

では動作確認を行います。まず初めに「Funnel」サーバを起動させます。次に「physical × wonderfl」に用意されているサンプルにジャンプしましょう。

<http://wonderfl.net/c/uR1R/>

「Play」ボタンを押します。

LED が 1 秒間隔で点滅するか確認してください(図-3)。

LED を光らせる

■ LED について

LED 電球の普及もあり、LED はとても身近なものになりました。また、LED はインジケータランプに使われるようにインタラクションの反応としても身近な装置です。大きな特徴として「消費電力が少ない」「寿命が長い」「発熱量が少ない」などがあります。電圧で発光する半導体素子に、順方向に電圧をかけることで発光します。そのため極性があり接続に注意が必要です。

長いリード線がアノード側、短いリード線がカソード側になっています。足を切りそろえてしまった場合は、発光部の中をみて半導体素子のリード線がカソード側になります。

■ 配線

- 5mm 赤色 LED[s1], ジャンプワイヤ

実体配線図を参考にブレッドボードを使用して、デジタルピン 13 番から LED, 抵抗 (330 Ω 程度), GND (グラウンド) とジャンプワイヤを使用して接続します。極性を間違えると発光しません(図-4)。

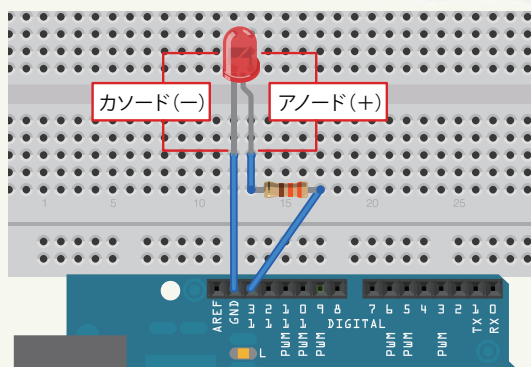


図-4 LED 実体配線図

■ 抵抗の役割と値

電子工作などで使用する砲弾型 LED では、最大の電流が 20mA 程度でこれを超えた電流が流れてしまうと異常発熱により寿命を縮めてしまったり、素子が破壊されてしまいます。このため一般的には抵抗器を使用して電流量を調整する必要があります。

Ω の法則により (電源電圧 - 使用する電圧) / 流したい電流 = 抵抗値 という式になります。

実際にこれを 1.8V の赤色 LED (電流 10mA を流す) で考えると $(5V - 1.8V) / 0.01A = 320 \Omega$ となります。抵抗器には 320 Ω という数値はないので最も近い 330 Ω を使用することになります。

■ 点滅させる

<http://wonderfl.net/c/7xt0/>

接続が完了したら、Funnel サーバを起動して次のコードにアクセスしてください。

```
// ボタンを押して LED オン
squareButton.addEventListener(MouseEvent.CLICK, function mousePressed(e:MouseEvent):void {
    led_pin.value = 1.0;
    squareButton.scaleX = 1.2;
    squareButton.scaleY = 1.2;
});

// ボタンを押して LED オフ
squareButton.addEventListener(MouseEvent.CLICK, function mouseReleased(e:MouseEvent):void {
    led_pin.value = 0.0;
    squareButton.scaleX = 1.0;
    squareButton.scaleY = 1.0;
});
```

「Play」ボタンを押した後、画面の四角ボタンをマウスで押してください。クリックに合わせて、LED が点滅します。

■ 応用編

LEDに限らず、「電圧をかける」か「電流を流す」と動く部品は、基本的にはこの LED と同じように扱うことができます。このときに注意すべき点をあげておきます。

(1) 加える電圧は? : Arduino などのマイコンの I/O ピンは H のときに、通常は電源電圧の電圧が出力されます。接続する部品にこの電圧が加わっても大丈夫か確認をします。Arduino では電源電圧は 5V ですが、心配な場合はテストで測るか、データシートの Supply Voltage (電源電圧) 欄で確認します。もっと低い電圧が必要な場合は、先ほどの LED の電流制限抵抗と同じように抵抗をはさむか、電圧変換インタフェース用の IC を使います。また 1.2V のような連続量 (アナログ) の電圧が必要な場合は、内蔵されている D/A 変換器を使います。またはパルス波で ON の時間の比率を変える (PWM 波) ことで、印加する (平均の) 電圧を制御する方法もよく使われます。

(2) 流す電流は? : Arduino などのマイコンの I/O ピンが部品に流せる電流は、多くても 20mA 程度です (詳細はデータシートの「DC Characteristics (DC 特性)」欄に載っています)。たとえばモータのように、100mA 以上の電流が必要な場合は I/O ピンには直接接続できませんので、専用のドライブ用 IC (モータドライバなど) やトランジスタ、リレーなどを使って制御します。たとえばトランジスタのエミッタ接地増幅回路で、マイコンの I/O ピンから 10k Ω 程度の抵抗を介してトランジスタのベース端子に接続し、大きな電流を流したい対象をコレクタに接続する負荷とすることで、多くの場合は大丈夫です。ただしリレーやモータなど、コイル (インダクタ) 分のあるものを駆動する場合は、OFF 時に大きな誘導起電力が発生してトランジスタを破壊することがあるので、負荷と並列にダイオードを接続します (フライホイールダイオード)。

スイッチを使う

■ タクトスイッチについて

LED が身近なアウトプットであれば、タクトスイッチは最も身近なインプットのための装置です。押したときにだけ通電するという単純な仕組みではありますが、クリック、ダブルクリック、トリプルクリック、長押しのように工夫次第で何種類かの入力考えることができます。タクトスイッチには、色、形、大きさなどとても多くの種類があります。押した状態をホールドするもの、ロック機構が備わったものもあるので、用途に応じて必要なものを選んでください。

■ 配線

- タクトスイッチ [s1]・ジャンプワイヤ

タクトスイッチの配線は至って簡単です。5V からタクトスイッチ、GND とジャンプワイヤを使用して接続します(図-5)。

■ スイッチを使う

<http://wonderfl.net/c/4nko/>

接続が完了したら、Funnel サーバを起動して次のコードにアクセスしてください。

```
// センサに接続したピンのモードを入力にセット
var config:Configuration = Arduino.FIRMATA;
config.setDigitalPinMode(2, IN);
arduino = new Arduino(config);

// センサに接続したピンを初期化してイベントリスナをセット
sensorPin = arduino.digitalPin(2);
sensorPin.addListener(PinEvent.RISING_EDGE, onTouch);
```

「Play」ボタンを押した後、タクトスイッチを押してください。タクトスイッチが押されると音が鳴ります²⁾。

■ 応用編

タクトスイッチは、押していないときには OFF になりますが、このとき Arduino などのマイコンの入力ピンは、どこにも接続されていない(オープン)状態となります。本来はこの状態では、入力ピンの電圧を決める要因がないため、静電気などで電圧が変化し、誤作動や故障の原因となります。そこ

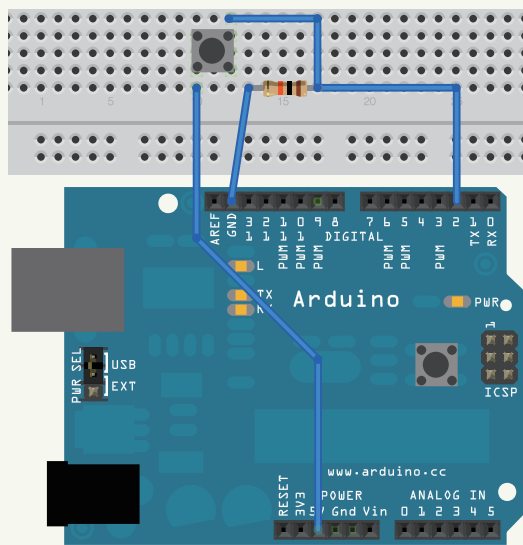


図-5 タクトスイッチ実体配線図

で通常は開放にならないように、10k Ω程度の抵抗を入力ピンと電源(+)端子に接続しておきます。こうすることで、スイッチを押していないときには入力ピンの電圧が高く(1)、押すと低く(0)なります。

また明るさで抵抗値が変わる CdS 素子のような抵抗値の変化を測定したい場合は、測定対象の素子と、それと同じくらいの大きさの抵抗を+電源と GND の間に直列に接続し、その中間の電圧を「スイッチを使う」の例と同じようにアナログ入力として読み取ります。またはその変化を 1/0 で読み取りたい場合は、あまり動作は安定しないかもしれませんが、判定したい抵抗の値に対応する測定電圧が、デジタル入力ピンが 1/0 を決めるしきい値(Threshold)をまたがる電圧となるように設計します。なおこのしきい値は、マイコンのデータシートの DC 特性欄に載っています(通常は電源電圧の 1/2 程度)。

動きを計測する

■ 動きを計測する加速度センサ

加速度センサは、iPhone や Nintendo Wii のコントローラで使用されたことで一般にも広く知られるようになりました。加速度とは、単位時間あたりに対する速度の変化率です。加速度センサは、この変化を電圧の変化などとして出力することができます。

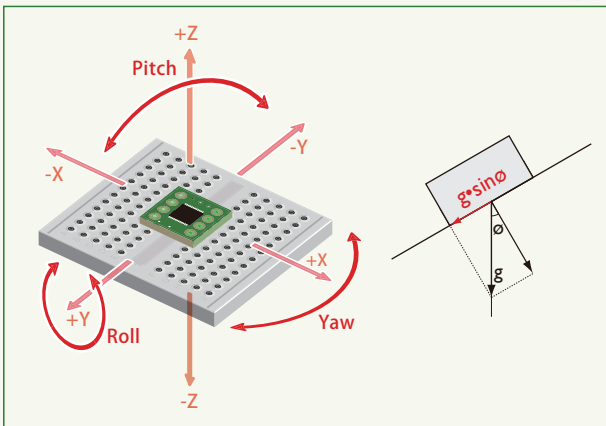


図-6 加速度センサで計測できる動きと角度

加速度には動かしたときに生じる動的加速度と、重力によって生じる静的加速度があります。

動的加速度は、動いた加速に対して生じるので物体が等速に動いている場合の加速度は0となってしまう。そのため、動的加速度は何か動いたという動きの検出に向いています。等速で動き続ける物体の速さなどの検出には向いていません。

一方、静的加速度では角度を求めるためによく使用されます。地球の地表付近にある物体はどんなものでも地球の中心に引っぱられる重力の影響を受けています。その大きさはその物体の質量に(ほぼ)比例しています。この比例定数を重力加速度と呼びます。物体は常に地面方向へ加速しているともいえ、地面方向に反応する軸のセンサは常に一定の加速度に対して反応します。センサの軸が3軸(X, Y, Z)ある場合には、軸の傾きに応じてそれぞれの重力加速度の成分が分配されます。そのため、それぞれの軸の出力を分析することで角度を求めることができます(図-6)。

■ 配線

- 加速度センサ (KXM52-1050) [s1], ジャンプワイヤ

センサの向きを間違えないように、X, Y, Zをそれぞれ A0, A1, A2 に接続します(図-7)。

■ 動きを計測する

<http://wonderfl.net/c/8wmt/>

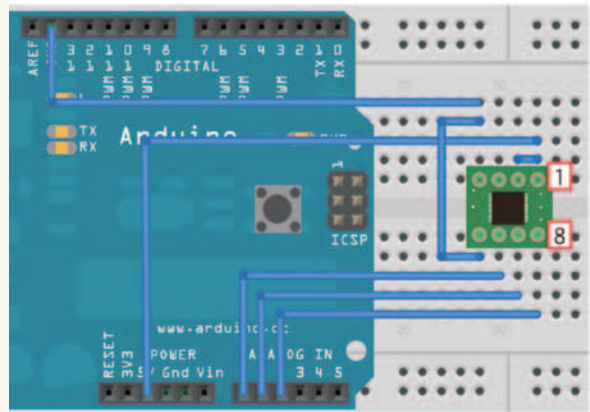


図-7 加速度センサ実体配線図

接続が完了したら、Funnel サーバを起動して次のコードを入力してください。

```
override protected function onRenderTick(e:Event = null):void {
// 立方体の角度をセット
cube.rotationZ
= Math.asin(yAxisPin.value) * 180 / Math.PI;
cube.rotationX
= Math.asin(xAxisPin.value) * 180 / Math.PI;

// シーンをレンダリング
renderer.renderScene(scene, camera, viewport);
}
```

「Play」ボタンを押した後、センサを傾けキューブを動かします³⁾。

■ 応用編

この加速度センサのようにアナログ電圧で得られるセンサは、その値の変化の範囲をデータシートから読み取り、Arduinoなどのマイコンが扱える範囲であればそのままアナログ入力端子に接続して使用できます。その電圧変化が小さすぎる場合はオペアンプの非反転増幅回路などで増幅をしたり、あるいは電圧変化の範囲がマイナスも含む場合には、2本の抵抗で電源電圧の1/2程度の電圧をつくり、そこにコンデンサを通して接続することで高域通過フィルタ(HPF)が形成され、信号の変化の範囲をプラスの範囲に設定することができます。

また値をデジタル値として読み出すセンサもあり、ノイズに強いなどの利点があります。これらは、そのセンサの値の読み出し方(プロトコル)をデータシートで確認し、そのプロトコルに合わせたマイコン側の内蔵インタフェースを使って、制御のプログラムを書きます。主なプロトコルとしては、UART、

I2C, SPI などがあり、いずれも Arduino で扱えます。またその制御は、センサ内のレジスタを、アドレスを指定して読み書きする方法が一般的ですが、そのレジスタの内容は、各センサのデータシートを熟読する必要があります。特に、測定開始のための設定レジスタや測定結果が保持されるレジスタへのアクセスは必須です。

音を鳴らす

■ 圧電スピーカについて

音を鳴らすには圧電スピーカを使用します。薄い圧電素子(ピエゾ素子)と金属板を貼り合わせた構造になっており、電圧を加えることで振動して音を鳴らすことができます。また、加えられた振動で電圧が発生するため簡単な振動を検知するセンサとして使用されます。センサとして使用する際は、大きな衝撃では数十Vの電圧が発生するため注意が必要です。

■ 配線

- 圧電スピーカ [s1]

デジタルピン 13 番と GND に圧電スピーカをつなぎます(図-8)。

■ 音を鳴らす

<http://wonderfl.net/c/7xt0/>

```
// ボタンを押して LED オン
squareButton.addListener(MouseEvent.MOUSE_DOWN,
function mousePressed(e:MouseEvent):void {
    led_pin.value = 1.0;
    squareButton.scaleX = 1.2;
    squareButton.scaleY = 1.2;
}
);

// ボタンを押して LED オフ
squareButton.addListener(MouseEvent.MOUSE_UP,
function mouseReleased(e:MouseEvent):void {
    led_pin.value = 0.0;
    squareButton.scaleX = 1.0;
    squareButton.scaleY = 1.0;
}
);
```

「Play」ボタンを押した後、画面の四角ボタンをマウスで押してください。クリックに合わせて、圧電スピーカから音が鳴ります。

■ 応用編

圧電スピーカは、よく鳴る周波数が比較的狭く、

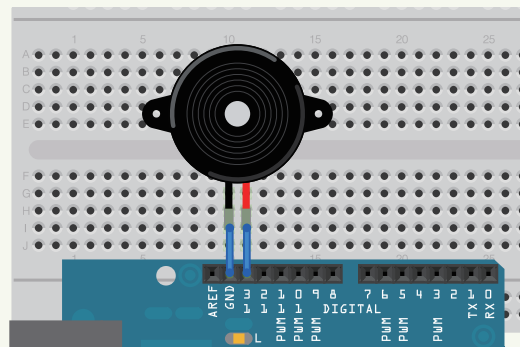


図-8 圧電スピーカ実体配線図

特に低音は得意ではありません。また加える電圧を小さくしても音の大きさはそれに比例しては小さくならないため、きれいなメロディを鳴らすのは得意ではありません。きれいな音を鳴らしたい場合はスピーカやヘッドフォンを使うこととなりますが、これらはインピーダンス(抵抗)が8Ω程度と非常に小さく、多くの電流を流す必要があります。ただし先ほど紹介したトランジスタを用いる方法ではスピーカの音量を細かく制御することが難しいため、スピーカやヘッドフォンのための専用のアンプICを使うのが一般的です。

動きをつくる

■ サーボモータについて

動きをつくるにはさまざまなアクチュエータがありますが、その中でも角度を比較的簡単につくり出せるサーボモータをアクチュエータとして使用します。角度を調整するごとに少しの工夫を加えることで、曲げる、開く、引っぱる、押し出すなどいろいろな動きをつくり出すことができます。

サーボモータは、主にラジコンなどでよく使用されますが、安いものであれば1,000円ぐらいから高いものだと数万円のものまであります。今回は超小型サーボ GWS PICO/STD/F を使用します。

GWS PICO/STD/F は、小型サーボで使用電流が少ないため Arduino 本体からの電源を用いますが、通常サーボモータは負荷がかかったときにはより大きな電流が流れます。そのときには Arduino または、PC の電源保護回路が働いてしまいます。その

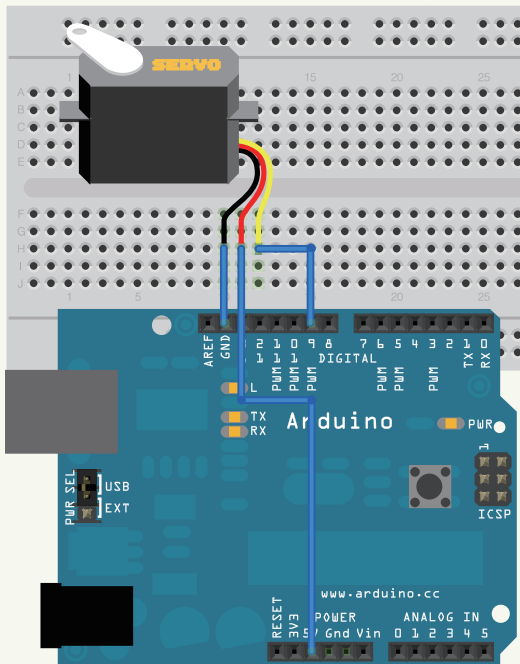


図-9 小型サーボモータ実体配線図

ため、小型サーボ以外では外部からの電源を用います。

■ 配線

- サーボモータ (GWS PICO/STD/F) [s1]
5V と GND に接続した後、入力を D9 に接続します (図-9)。

■ サーボモータを動かす

<http://wonderfl.net/c/54NS/>

接続が完了したら、Funnel サーバを起動して次のコードを入力してください。

```
private function onPulse(e:TimerEvent):void {
  var angle:Number = Math.random() * 180;
  //180 (度) の数値を変えることで、
  //1 秒ごとに指定した角度の範囲でサーボが駆動します。
  servo.angle = angle;
}
```

「Play」ボタンを押すと 1 秒ごとに指定した角度の範囲でサーボモータが駆動します³⁾。

■ 応用編

電流を流すと回転する直流モータには、ホビーユースではマブチモータなどが有名で、タミヤのギヤボックスなどの機構部品とセットで使うと、いろいろと楽しいものを作ることができます。このような

直流モータは、特に回転し始めるときに非常に大きな電流 (ものによっては 1A 近く) が必要となるため、その電流を流す能力がある電源を使う必要があります。特にレギュレータなどの電源回路からモータへの電流を流す場合には注意が必要です。直流モータの制御は、先ほど紹介したトランジスタを用いる回路でも可能ですが、回転方向を変えたい場合には、直流モータに流す電流の向きを変える必要があります。そのための回路には、4 個のトランジスタを使う「Hブリッジ」と呼ばれる回路が使われますが、それがいった専用のドライバ IC が各種ありますので、それらを使うとよいでしょう。モータの回転開始時の電流の変動による電源電圧の変化でマイコンにリセットがかかってしまう場合もありますが、そのような場合は電源を補強するバイパスコンデンサ (パスコン) を追加したり、モータ系とマイコン系の電源を分離したり、モータ駆動信号をフォトカップラなどで電気的に分離したりするなど、いろいろなテクニックがあります。

部品の入手方法

フィジカルコンピューティングを自分で試してみるには、いろいろな部品、特に電子部品を調達する必要があります。昔は、東京・秋葉原や大阪・日本橋、名古屋・大須、などに大きなパーツショップ街があり、買いに行ったり通販で購入するのが一般的でした。今でも部品の入手方法の基本は、やはり店頭か通販ということになりますが、インターネットの普及で、Web から発注・購入ができる店舗も増えてきて、地方在住であっても、以前ほどは部品の調達に困ることはないかと思います。代表的な店舗をいくつか紹介しておきますので、ぜひご自身の目で確かめて、活用してください。また各店舗で、特に品揃えの豊富な製品カテゴリをいくつかあげておきます (もちろんそこに挙げた製品以外の品揃えもあります)。なお筆者の生活圏の関連上、東京・大阪・名古屋の店舗が多いことをご了承ください。

また研究者の方で、校費などで部品を購入したい方も多いかと思います。請求書払いが可能な店舗は

☆印をつけてあります。事前に必要な手続き（債主登録など）が必要な場合もありますので、利用される際は事前にご確認ください。なお、各ショップについている引用番号は他の記事でも引用されていますので参考にしてください。

■ 電子部品全般

- [s1] 秋月電子((株)秋月電子通商)
秋葉原(秋葉原店), 埼玉県八潮市(八潮店)
<http://akizukidenshi.com/>
マイコン単体・ボード (PIC・H8), IC ソケット・コネクタ, LED, ユニバーサル基板, 高周波コネクタ, 各種センサ単体, キャラクタ液晶モジュール, タクトスイッチ, 圧電スピーカ
- [s2] 千石電商((株)千石電商)☆
秋葉原(秋葉原本店), 大阪(大阪日本橋店)
<http://www.sengoku.co.jp/>
受動部品単体 (抵抗・コンデンサ・インダクタ), コネクタ, 工具, 半導体部品単体 (トランジスタ, ダイオード), サンハヤト製品, 電子部品全般
- [s3] 若松通商((株)若松通商)
秋葉原(本店), 福島県会津若松市(若松店)
<http://www.wakamatsu-net.com/biz/>
半導体部品全般, Arduino 関連.
- [s4] 共立電子(共立電子産業(株))☆
大阪(シリコンハウス, デジット)
<http://eleshop.jp/>
受動部品単体 (抵抗・コンデンサ), 半導体部品単体 (IC・トランジスタ・ダイオード), コネクタ, Arduino 関連, SparkFun 製品
- [s5] マルツ電波(マルツエレクトロニクス(株))☆
仙台, 秋葉原(2店), 静岡, 浜松, 名古屋, 福井(2店), 京都, 大阪, 金沢, 福岡
<http://www.marutsu.co.jp/>
スイッチ・コネクタ, CQ 出版関連製品, Arduino 関連, 電子部品全般
- [s6] タケイ無線
名古屋
半導体部品単体, 秋月電子製品

■ 電子部品全般 (通販のみ)

- [s7] ストロベリー・リナックス☆
<http://strawberry-linux.com/>
オリジナルモジュール部品 (電源など), SparkFun 製品, Arduino 関連,

- [s8] スイッチサイエンス☆
<http://www.switch-science.com/>
Arduino 関連, Gainer 関連, SparkFun 製品
- [s9] RS コンポーネンツ☆
<http://rswww.co.jp/>
電子部品全般, 工具・計測器全般
- [s10] チップワンストップ☆
<http://www.chip1stop.com/>
電子部品全般
- [s11] メカロボショップ☆
電子部品全般, ロボットパーツ関連, 電子手芸関連
<http://www.mecharoboshop.com/>
- [s12] Sparkfun (海外)
<http://www.sparkfun.com/>
オリジナルキット・モジュール部品, センサ単体, 無線通信機
- [s13] Digi-Key (海外)
<http://www.digikey.com/>
電子部品全般
- [s14] Mouser (海外)
<http://jp.mouser.com/>
電子部品全般

■ 番外編

一般的な部品は必ずしも扱っていませんが、掘り出し物が見つかるお店です (すでに閉店している店舗(●印)を含めて紹介しておきます)。

- [s15] 日米商事: 秋葉原. 電子部品全般, ジャンク品.
[s16] 塚口勇商店: 大阪日本橋. モータ, ジャンク品.
[s17] ボントン: 名古屋. スイッチング電源など.
[s18] たんせい●: 秋葉原. ジャンク品.

参考文献

- 1) physical × wonderfl: <http://physical.wonderfl.net/>
- 2) 面白法人カヤック & フォークビッツ: ブラウザで始める ActionScript, (株)ワークスコーポレーション (2009).
- 3) 小林 茂: Prototyping Lab, (株)オライリージャパン (2010). (2011年5月31日受付)

■ 蛭田 直 sunao@shinshu-u.ac.jp

基礎デザインからメディア表現を中心に講義をもち、主な活動として電子玩具から知育玩具などのデザイン、メディア表現の教育教材などの開発を行っています。

■ 秋田純一 (正会員) akita@is.t.kanazawa-u.ac.jp

1998年東京大学大学院工学系研究科電子情報工学専攻修了(博士(工学)). 金沢大学助手, 公立はこだて未来大学講師, 金沢大学講師を経て, 2007年より金沢大学理工学域電子情報学類准教授. 専門は集積回路とその応用システム.