

DSP コース・ワークショップ : Physical Computing

第 6 回 : Processing

小林 茂

2005 年 11 月 24 日*

1 Processing とは何か？

1.1 概要

Processing^{*1}は Ben Fry (Broad Institute) と Casey Reas (UCLA Design | Media Arts) が中心になってオープンソースで開発を続けているプログラミング環境で、2005 年の Ars Electronica の Net Vision 部門で Golden Nica を受賞した。



図 1 Processing のアイコン

Processing の主な特長は次のようになる。

- 複数の OS 上で利用できる
 - Windows XP
 - Mac OS X
 - Linux
- 無償でダウンロードして利用することができる
- サンプルが数多く揃っている
- 様々なメディア処理を行うためのライブラリが豊富
 - Open GL
 - ネットワーク
 - MIDI

- Processing そのものは Java で記述されているが、プログラミングにはオブジェクト指向プログラミングの知識を必ずしも必要としない^{*2}
- アイデアを思いついた時、手軽にプログラムを書いてすぐに試すのに適している
- プログラミング言語としての文法は Java をベースにしているためわかりやすい
- gainer 専用のライブラリが用意されている

1.2 Processing でどのようなものが作れるのか？

Processing のウェブサイトにある Exhibition^{*3}をみると、Processing でどのようなことができるのかを知ることができる。それぞれの作品のページを開くと、それぞれの作品の説明と一緒にスケッチ (sketch) をみることができる様になっているものが多い。

それぞれのスケッチは Java アプレットであるため、Java を利用できる様に設定されているウェブブラウザであれば、その場で動かしてみることができる。また、大抵の場合にはソースコードも公開されているため、どのようなプログラムで実現されているのかを知ることができる。

その作品の中には PC 上の Processing とプロジェクトだけで完結しているものもあるし、様々なセンサやアクチュエータを組み合わせられて構成されているものもある。この様に、Processing はそれ単体だけでなく、他の要素と組み合わせることで幅広い

* 改訂 : 2005 年 12 月 28 日

*1 <http://www.processing.org/>

*2 大規模なプログラムや、ライブラリを開発する際には必要になってくるが、最初から必要としないので、だんだん慣れていけばよい。

*3 <http://www.processing.org/exhibition/>

作品制作に使えるプラットフォームとなるのである。

1.2.1 課題

Exhibitonの中から自分が最も興味深いと思う作品（ただしスケッチとソースコードが紹介されているもの）を選び、ソースコードを読んで自分なりに解説せよ。

2 Processing のインストールと起動

2.1 Processing 本体のセットアップ

Processing は、公式サイトダウンロード・ページ^{*4}から誰でもダウンロードすることができる。Windows XP の場合には 2 種類のパッケージがあるが、Java 環境を含む「with Java」のパッケージをダウンロードするとよい。Mac OS X の場合にはパッケージは 1 種類しかないが、OS 10.4 ではプリインストールされない Stuffit 形式で圧縮されているため、OS 10.4 を使用する場合には Stuffit Expander を入手するなどの対応が必要である。

ダウンロードが終わったら、適当な場所^{*5}に解凍する。インストーラ等は必要とせず、これだけで Processing 本体のインストールは終了する。

Processing フォルダ内のアプリケーション本体（例：Processing 95）をダブルクリックすると、Processing が起動する。問題なく起動することを確認したら、一度終了する。

2.1.1 シリアル関連のセットアップ（OS X のみ）

Mac OS X の場合には、この後にシリアル関連のセットアップが必要である。シリアル関連のセットアップは、次の手順で行う。このセットアップを行わないと、gainer を含む USB シリアル機器は利用できないので必ず行う。

1. Processing をインストールしたフォルダに移動する
2. libraries フォルダに移動する
3. serial フォルダに移動する
4. macosx_setup.command をダブルクリックする

^{*4} <http://www.processing.org/download/index.html>

^{*5} Windows XP であれば Program Files、Mac OS X であれば Applications（アプリケーション）など。

る（図 2）

5. 図 3 のような画面が表示されるため「Y」の後に return を入力して先に進める
6. 数秒間で図 4 のような画面が表示されるのでそれまで待つ
7. Terminal を終了する
8. Apple メニューから「システム終了...」を選択してシステムを終了する
9. 電源が切れたら電源ボタンを押して Mac を起動する

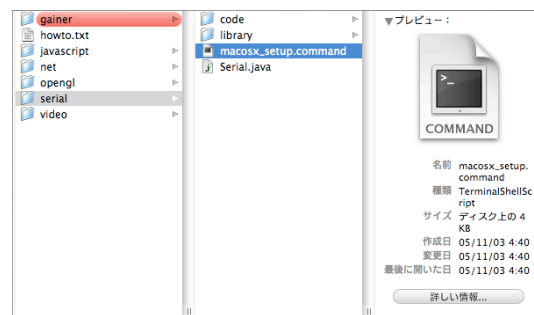


図 2 シリアル関連セットアップ用のコマンド

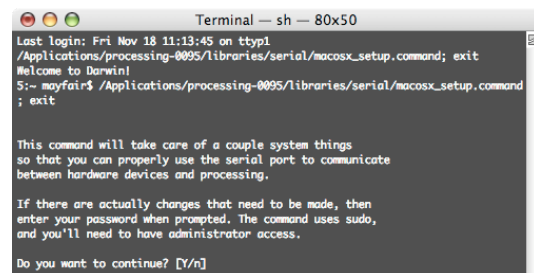


図 3 セットアップ前の確認

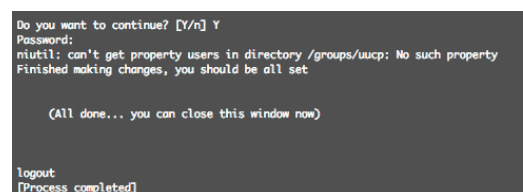


図 4 セットアップ終了時の表示

2.2 gainer 用ライブラリのセットアップ

gainer 用のライブラリは AD1 のワークショップ 関連ファイルが置いてあるサーバから入手する。入手できたら、次の 3 つのフォルダを所定の位置に移動する。なお、以下の説明で (Processing) という部分はインストールしたディレクトリに置き換えて作業すること。

- (Processing)/libraries に gainer のライブラリフォルダ (to_libraries/gainer) を入れる (図 5)
- (Processing)/reference/libraries に gainer のリファレンスフォルダ (to_reference_libraries/gainer) を入れる (図 6)
- Processing のスケッチブックフォルダ*6 に サンプルのフォルダ (to_sketchbook/gainer-examples) を入れる

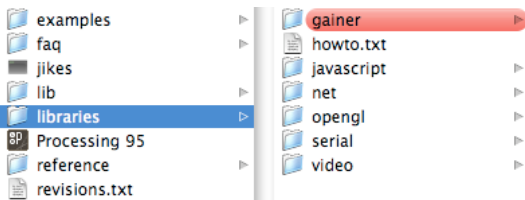


図 5 ライブラリフォルダのコピー先

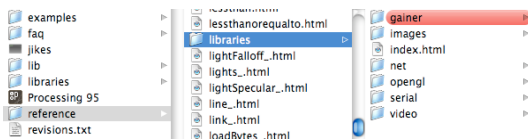


図 6 リファレンスフォルダのコピー先

2.3 gainer と接続した動作確認

ライブラリのセットアップが終わったら、gainer を USB ポートに接続し、次の手順で動作確認を行う。

1. Processing を起動する

*6 Windows XP では My Documents (マイドキュメント) の中にある Processing、Mac OS X では (HOME)/Documents (書類) の中にある Processing がドキュメントフォルダである。

2. File メニューから Sketchbook → gainer-examples → button を選択する
3. 必要に応じてシリアルポートの名前を変更する (図 7)
4. ウィンドウ左上の Run ボタンをクリックして実行する
5. processing.core.PApplet が起動する (図 8)
6. gainer ボード上のボタンを押す (図 9)

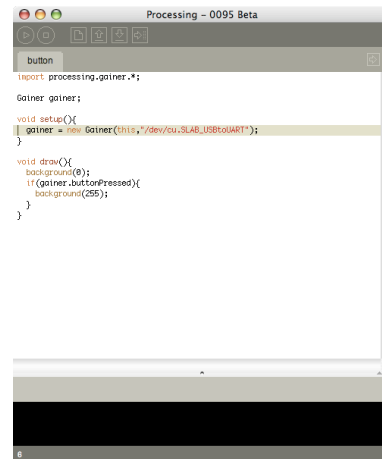


図 7 自分の環境に合わせてシリアルポートの指定を変更する

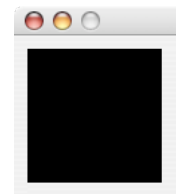


図 8 プログラムを実行した直後の表示

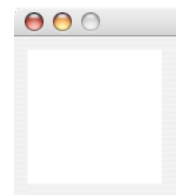


図 9 ボタンを押した時の表示

この状態で、gainer ボード上のボタンを押した時に PC 側の画面が変化すれば動作確認は OK である。もし動作確認ができない場合、Processing のメッセージエリアやテキストエリアに、何かエラーメッセージが表示されていないかどうか確認する。

3 Processing のサンプル紹介

gainer_examples には 12 個のサンプルがある。1 つずつ試していくことにより、Processing から gainer を使用するための方法が理解できるだろう。なお、これらのサンプルを試す際、シリアルポートの名前の変更^{*7}が必要になることに注意。

- analog_get
 - アナログ入力の使用例
 - マウスのボタンを押した時にアナログ入力の値を取得して表示を更新
- analog_seq
 - アナログ入力の使用例
 - アナログ入力の値を常に取得して表示を更新
- analogOut
 - アナログ出力の使用例
 - マウスのボタンを押している間はマウスの Y 方向の値をポート 1 と 2 に出力
 - マウスのボタンを押していない時はマウスの X 方向と Y 方向の値をポート 1 から 4 までに出力
- button
 - ボタンの使用例
 - 背景色をボタンが押されている間は白、押されていない時は黒に変える
- buttonFunction
 - ボタンの使用例 (コールバック型)
 - 背景色をボタンが押されている間は赤、押されていない時は青に変える
- digitalIn_get
 - デジタル入力の使用例

- マウスのボタンを押した時にデジタル入力の値を取得して表示を更新
- digitalIn_seq
 - デジタル入力の使用例
 - デジタル入力の値を常に取得して表示を更新
- digitalOut
 - デジタル出力の使用例
 - マウスの左ボタンで 0x00 を、右ボタンで 0xFF を出力
 - ASCII キーボードによる入力でも制御できる
- digitalOut2
 - デジタル出力の使用例
 - 1 秒ごとにカウントアップし、その値をデジタル出力ポートの 1 から 4 までに 2 進数で出力
- fps
 - gainer との通信速度を計測
- LED
 - LED の使用例
- pong
 - アナログ入力ポートの 1 と 2 に可変抵抗器を接続して使用

4 その他

Processing 自体の文法に関しては公式ウェブサイトの Reference ページ^{*8}が参考になる。それぞれの関数には実際に動作する簡単なサンプルが付属しているため、その動きを見ながら理解していくことができる。また、ソースコードを読みながら使い方の分からない関数にであった場合、大抵の場合にはそのキーワードをダブルクリックして選択し、Find in Reference (ショートカットキー + シフト + f) を実行することでウェブブラウザでリファレンスを見ることができる。

^{*7} Windows XP であれば COM2 など。Mac OS X であれば/dev/cu.SLAB_USBtoUART。

^{*8} <http://www.processing.org/reference/>