

コンピューター言語 Processing を用いる簡易なプログラム開発

鈴木保任*, 川久保進

山梨大学大学院医学工学総合研究部：400-8510 山梨県甲府市武田 4-4-37

Simple application software development using “Processing” language

Yasutada Suzuki* and Susumu Kawakubo

Interdisciplinary Graduate School of Medicine and Engineering, University of Yamanashi,
4-4-37, Takeda, Kofu, 400-8510, Japan

Open-source programming language, “Processing” is easy to study but it has a lot of useful features to build simple programs for data processing and/or hardware control by ourselves. Processing is free to download and to use, and the program built with Processing can be run on Windows, MacOS, and Linux. In this article, we explain the overview of Processing, how to install, and how to write programs in brief.

Keywords Processing, Java, Programming language

1. はじめに

簡単なデータ処理やハードウェアの制御のためには、プログラムを開発して利用できると便利である。最近、様々なプログラムの開発環境が安価、あるいは無料で提供されるようになっており、研究室内での小規模なプログラム開発が容易になっている。

一方で、OS やハードウェアの高機能化に伴って、従来は簡便なソフトウェア開発に多用されていた Visual Basic が非常に高度な機能を有するようになり、以前のように修得ができなくなっていると感じる。著者らも騙しだまし Visual Basic 6 を用いて開発したアプリケーションを使い続けて来たが、Windows 10 で動作はするものの、プログラムの修正が困難になりつつある。

自作の蛍光検出器をフロー分析に用いるにあたり、フローシグナルを記録するためのプログラムを開発する必要があった。この時、以前開発した吸光度検出器のプログラムを流用しようと試みたが、サポートの切れた Windows XP 上での開発は困難であった。そこで、USB による外部の機器との通信が可能で安価（可能ならフリーソフト）、かつ修得が容易な開発言語を探してみたところ、Java をベースにした Processing という言語があることを知り、開発に利用したところ、検出器との通信、フローシグナルの取得、表示および保存が可能なプログラムを比較的短時間に開発することができた。

Java ベースであることから、Basic とは異なり Java の文法を理解しておく必要がある。しかし、それをマスターしてしまえば、通信やグラフ表示などに関するライブラリー

が充実しているので、目的とする機能を有するプログラムを簡単に作成することができる。本稿では、Processing による開発について、インストールから簡単なプログラムの作成までを通じて紹介する。

2. Processing とは

Processing は 2001 年にマサチューセッツ工科大学の Ben Fry 及び Casey Reas によって開発された、フリーでオープンソースな開発言語である。公式 Web ページにあるように、元々映像を使った芸術作品を作成するために開発され[1]、そのためグラフィックスに関する機能が充実している。また、コンピュータープログラミングを専門としない人にも利用できるよう配慮されている。さらに、Windows, Macintosh, Linux, Raspberry Pi [2] (Linux が動作する小型マイコンボード) 上で開発できるだけでなく、作成したプログラムはいずれの環境でも動作する（ただし、最近のバージョンでは、Mac OS X で動作するプログラムの作成は、Mac OS X 上で行わなければならないようになってしまった）。また、多少の変更が必要な場合もあるが、Android 上で動作するプログラムも開発できる。さらに Android スマートフォンやタブレットに搭載されている GPS や加速度などのセンサー、カメラ、ネットワーク機能などを用いるためのライブラリーも Ketai というプロジェクトで開発されている[3]。

3. ライブラリー

機能拡張のためのライブラリーには、Processing プロジェク

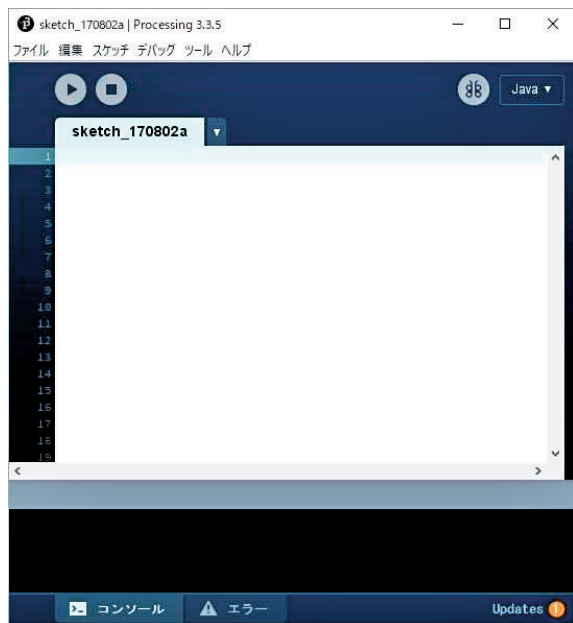


Fig.1 Editor screen of Processing

ト公式で Processing 本体に同梱されているものと、プロジェクト外の人々が開発しているものがある。前者には、PDF ファイルの出力、ネットワーク通信、シリアルポート通信、DXF (CAD 等に用いられる、図形をベクトルデータで表すファイルフォーマット) 出力、ビデオ (カメラからの入力、ビデオの表示、作成)、サウンド、ハードウェア入出力 (Raspberry Pi などデジタル・アナログ信号の入出力を可能にする) が含まれている。

後者は、<https://www.processing.org/reference/libraries/> にカテゴリーごとにまとめられている。例えば“Data”カテゴリーには、取得したデータからノイズを取り除く“Signal Filter”というライブラリーがある。また、Processing には GUI のライブラリー (テキストボックスやラジオボタン、スクロールバーなどマウス等を使用してプログラムを制御する機能) が含まれていないが、5 種類のライブラリーから好みのものを追加できる。

後ほど解説するが、自作の計測プログラムにおいてグラフを表示するのは非常に面倒な作業である。画面内に点や線を表示するのはたやすいが、軸に目盛り、目盛り数字を動的に表示させようとする、画面の書き直しを含めて相当量のプログラムを書かなければならない。“Math”のカテゴリーには、XY データとスタイル等を指定するだけで、適切な表示をしてくれるグラフ作成用ライブラリーが含まれていて、測定中のデータを簡単にグラフ化できる。

4. ダウンロードとインストール

Processing 本体のダウンロードは、Processing の Web ページ[1]から行う。Windows の 64 bit 版を例に説明する。ダ

ウンロードしたファイルは ZIP で圧縮されているので、適当なフォルダーに Explorer 等を使用して展開する。インストーラー (自動的に特定のフォルダーにコピーし、スタートメニューやデスクトップなどに登録する機能) はないので、展開してできたフォルダー (2017 年 7 月現在では“processing-3.3.5”) を適当な場所に移動し、その中にある“processing.exe”で右クリックし、“送る”の中の“デスクトップ (ショートカットを作成)”を左クリックすると、デスクトップ上のアイコンから Processing のエディターを起動できるようになる。起動時に Welcome message が表示されるが、チェックボックスをオフにしてから“Get Started”ボタンをクリックすれば、次回以降は表示されなくなる。この時、ドキュメントフォルダーに“Processing”フォルダーが作られ、この中に制作したプログラム等を保存することになる (フォルダーがないというウォーニングが出ることもある)。プログラム作成時に、日本語を入力すると文字化けするが、これは表示フォントが適切でないためである。“ファイル”→“設定”メニューで「エディタとコンソールのフォント」を MS ゴシックなどにすると、日本語が表示できるようになる。なお、古い PC では 3D グラフィクスが動作しないことがある。これは、Processing ではなく PC のグラフィクスライブラリーである OpenGL が対応していないためで、3D グラフィクスを使用しないのであれば問題ない。また、旧バージョンの 1.5.1 を用いれば、3D グラフィクスも動作する。手元の PC では、インテルの Core m5 や ATOM X7-Z8750、同 Z3770 などの比較的最近の CPU を搭載した PC は問題なく動作し、数年前の Core i5-2520 の PC では動作しなかった。PC 本体のビデオドライバーのアップデートで改善する可能性もある。

5. プログラムの作成

Processing を起動すると、Fig. 1 のようなエディター画面が現れる。Processing ではプログラムのことを“Sketch”と呼んでおり、新規作成時には sketch_日付+アルファベットの名前が自動的に付けられる。「名前をつけて保存」をすれば、変更は可能である。この画面から、Processing のサンプルプログラムを読み込み、実行することができる。“ファイル”をクリックし、“サンプル”をさらにクリックすると、“Basics”や“Topics”などのカテゴリーがあり、その中から様々なサンプルプログラムを選択できる。“Demos”カテゴリーのサンプルは、いずれも OpenGL を利用しているため、古い機種では動作しないことがあるが、その他のサンプルは問題ない。読み込むとエディター画面にプログラムが表示され、▶をクリックすると実行ウィンドウが別に表示され、その中でプログラムが実行される。実行時にファイアウォールに関する警告が出た場合は、Java がファイアウォールを越え

```

import controlP5.*; // GUI ライブラリーの読み込み
ControlP5 CP5;      // クラスインスタンスの宣言

void setup() {      // 起動時に最初に実行される
    size(320, 240); // 実行画面の幅と高さ
    CP5 = new ControlP5(this);
                    // インスタンス作成
    fill(0);        // 前景色を黒に
    CP5.addButton("endbutton")
                    // ボタンを追加
        .setLabel("End") // ボタンのラベル
        .setPosition(120, 180)
                    // ボタンの位置
        .setSize(60, 40);
                    // ボタンの幅と高さ
    text("Hello!", 20, 100);
                    // メッセージの表示
}

void draw() {      // 起動中繰り返し実行される
}

void endbutton(){ // ボタンクリック時に実行される
    exit();        // プログラムを終了する
}

```

List 1 Source code for a simple GUI program

てネットワーク接続することを許可する。プログラムの停止は■をクリックする。

では、実際に簡単なプログラムを作成してみよう。“Hello!”を画面に表示するとともに、“End”ボタンをクリックすると終了するプログラムを List 1 に示す。ここでは、GUI の部品であるボタンを表示させるため、ライブラリーを追加する必要がある。ライブラリーの追加は“スケッチ”→“ライブラリをインポート...”を選び、続いて“ライブラリを追加...”をクリックする。非常にたくさんのライブラリーが公開されていて、目的のものを探す際には右上のプルダウンメニューで目的のカテゴリーを選択するか、Filter に目的のライブラリー名の一部を入力するとよい。ここでは“GUI”カテゴリーを選び、その中から“ControlP5”を選択し、“Install”ボタンをクリックする。ダウンロードが始まり、しばらくするとインストールされる。表示が止まったら、引き続き別のライブラリーをインストールするか、×をクリックして画面を消す。ライブラリーはインストールするだけでは利用できず、作成する sketch にインポートする必要がある。先ほどの“ライブラリのインポート”をクリックすると“ControlP5”が追加されているので、これを選択するとエディター画面に

```
import controlP5.*;
```

という行が自動的に追加され、このライブラリーに含まれる機能を利用できるようになる。なお、ライブラリーをインストールすると、新たにライブラリーのサンプルも追加される。

プログラムに含まれる setup() と draw() の 2 つの関数は、どのプログラムでも共通で使われる。実行開始時のみ

```

import grafica.*;
// グラフライブラリーの読み込み
import controlP5.*; // GUI ライブラリーの読み込み

ControlP5 cp5;      // GUI 用インスタンスの宣言
GPlot plot;         // グラフ用インスタンスの宣言
int y, x = -10;     // グラフの X,Y データ用変数

void setup() {
    size(600, 700); // 実行画面のサイズ
    background(255, 200, 0); // 背景色 R,G,B 指定
    cp5 = new ControlP5(this);
                    // GUI インスタンスの作成
    cp5.addButton("drawbutton") // ボタンの追加
        .setLabel("Draw")
        .setPosition(500, 630)
        .setSize(80, 25);
    plot = new GPlot(this);
                    // グラフのインスタンスの作成
    plot.setPos(20, 20); // グラフの位置
    plot.setDim(450, 450); // グラフのサイズ
    plot.getTitle().setText("Graph demo");
                    // グラフタイトル
    plot.getXAxis().getAxisLabel().setText("X axis");
                    // X 軸タイトル
    plot.getYAxis().getAxisLabel().setText("Y axis");
                    // Y 軸タイトル
}

void draw() {
    plot.beginDraw(); // グラフの描画開始
    plot.drawBackground(); // 背景
    plot.drawBox(); // 外枠
    plot.drawXAxis(); // X 軸
    plot.drawYAxis(); // Y 軸
    plot.drawTitle(); // タイトル
    plot.drawGridLines(GPlot.BOTH);
                    // 目盛りの表示
    plot.drawLines(); // データの線を表示
    plot.drawPoints(); // データ点を表示
    plot.endDraw(); // 表示の終了
}

void drawbutton() { // Draw クリック
    y = x * x; // Y=X^2 を計算
    plot.addPoint(x, y); // 新しい点を追加
    x++; // x を 1 加算
}

```

List 2 Source code for a graph plotting program

に呼び出されるのが setup で、プログラム起動中常に行われているのが draw である。setup で画面上の部品の配置や変数の初期化を行い、実行中の状況変化に対応するコードを draw 内に記述する。Java は関数型言語であるので、ユーザ一定義の関数を作成することもできる。このプログラムでは、“End”ボタンをクリックするとプログラムを終了する endbutton() 関数を作成している。関数名は、ボタン作成時の引数と同じにする {CP5.addButton("endbutton")}。

6. グラフ表示プログラムの例

2 次元グラフを作成するためのライブラリーは“Grafica”なので、ControlP5 と同様にインストールし、インポートす

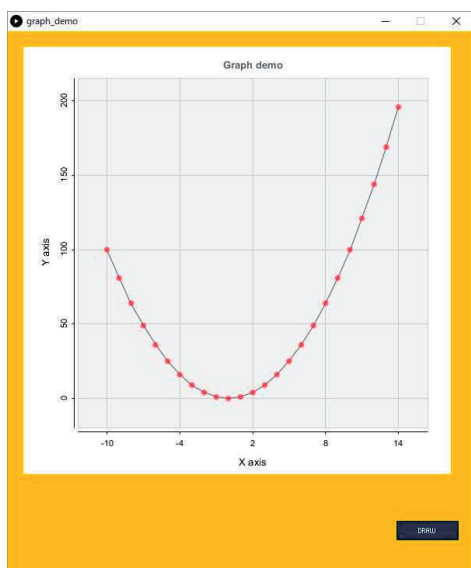


Fig. 2 Screen shot of the graph plotting program shown in List 2

る。先ほどのプログラムと異なるのは、グラフ用のインスタンス及び座標の変数の宣言、`setup` 関数内でグラフのインスタンスを作成するとともに、グラフのタイトル等を設定している、などである。`draw` 関数内ではグラフの各部 (枠, 軸, データ点など) を常に更新している。“Draw” ボタンを押すと、Y の座標値に X の 2 乗値を代入し、X, Y データをグラフに追加し、その後 X の座標値を 1 加算する。プログラムを実行すると、“Draw” ボタンを押すたびに新しいデータ点が追加され、軸の目盛りや数値も含めて更新される。実行時の画面は、Fig. 2 のようになる。

7. 配布用ファイルの作成

エディター画面で動作しているプログラムは、Processing をインストールしてある PC でないと動かないので、別の PC 等で使用する場合には、“ファイル” – “アプリケーションとしてエクスポート” をクリックして配布用ファイルを作成する。表示されるウィンドウでアプリケーションを書きだすプラットフォーム (Windows, Mac OS X, Linux) を選択して“エクスポート”ボタンをクリックすると、Fig. 3 に示すように sketch 本体 (ここでは“graph_demo.pde”) の置いてあるフォルダーに、各プラットフォームに対応したフォルダーが作られ、その中に実行用ファイルが格納される。グラフ表示プログラムをエクスポートし、Linux 上で動作させている様子を Fig. 4 示す。

8. おわりに

機器の制御にはシリアルポートやネットワーク通信が必要であるが、紙面の都合で割愛した。Processing は教育の現

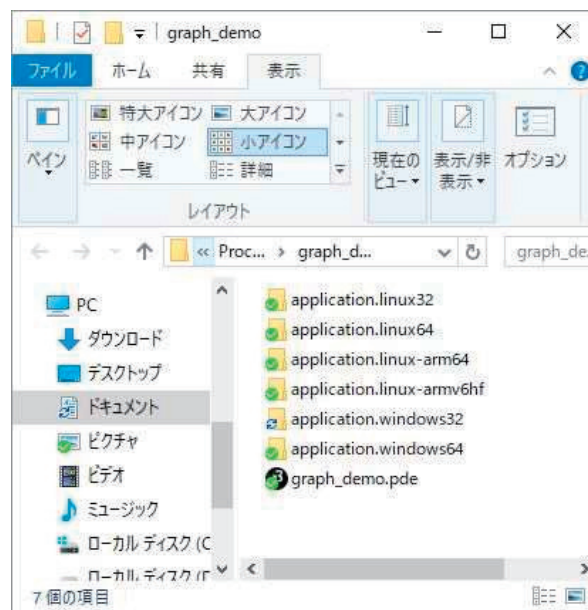


Fig. 3 Exported application files for running on other platforms

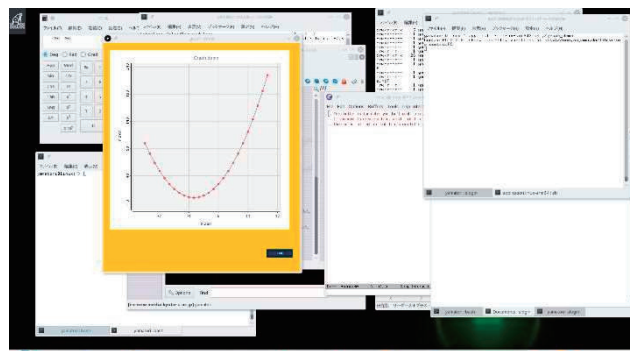


Fig. 4 Screenshot of the exported graph plotting program running on Linux PC

場でも多用されているので、日本語で解説している Web ページも豊富であり、入門には最適である。マイコンボードとの連携については、わかりやすい成書が出版されている [5]。Arduino というボードを例に挙げているが、Arduino の開発環境は Processing と出自が同じで、操作などはよく似ていて修得しやすい。こちらも日本語の資料が豊富であるので、興味があれば検索をお勧めする。

9. 参考文献

- [1] Processing, <http://www.processing.org/> (accessed 2017 7 31).
- [2] Raspberry Pi, <https://www.raspberrypi.org/> (accessed 2017 7 31).
- [3] Ketai Library Home, <http://ketai.org/> (accessed 2017 7 31).
- [4] Processing OpenGL issues, <https://github.com/processing/processing/wiki/OpenGL-Issues> (accessed 2017 7 31).
- [5] 青木直史: “Arduino と Processing ではじめるプロトタイプ入門”, P. 53 (2017), (講談社).