RT コンポーネント操作マニュアル

# 外部入力に適応し光が揺らぐランプ

2012 年 12 月 13 日版

芝浦工業大学

デザイン工学部デザイン工学科

斉藤 文哉, 佐々木 毅

**更新履歴**

2012 年 12 月 13 日版　第 1 版。

# 目次

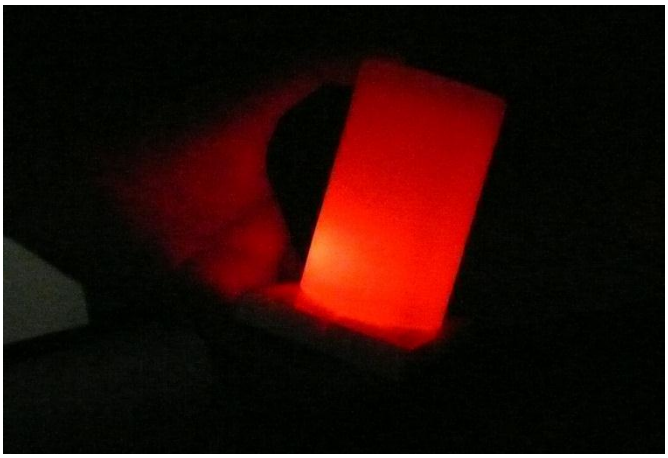
# 1 本コンポーネントの概要

## 1.1 開発の背景

近年、ろうそくの炎を再現した「LED キャンドル」と呼ばれる LED ランプが数多くあり、インテリアとして人気です。しかし、それらは静止したろうそくの炎の揺らぎを再現しているだけであり、外部から力を加えたときに起こる炎の揺れは再現されていません。よって、外部入力に反応する、より本物のろうそくに似た LED ランプを実現することで、LED ランプと言った人工的な物を自然の物の振る舞いに近づけることができ、生活空間の一部としてより溶け込むと考えました。そこで、より本物のろうそくに似た LED ランプを実現する第一歩として、LED ランプに加速度センサを取り付け、外部から加えられた力に反応し、本物のろうそくの炎のようにインタラクティブに光が揺らぐランプを RT ミドルウエアによるコンポーネント指向のシステム構築によって開発することとしました。

## 1.2 本ランプの機能

開発した LED ランプは加速度センサを取り付けていますので、傾ける、動かすといった際にそれを検知し、本物のろうそくのような振る舞いをします。例えば下図のように傾けた場合は、傾けた方向とは逆の LED が強く光るようになっています。

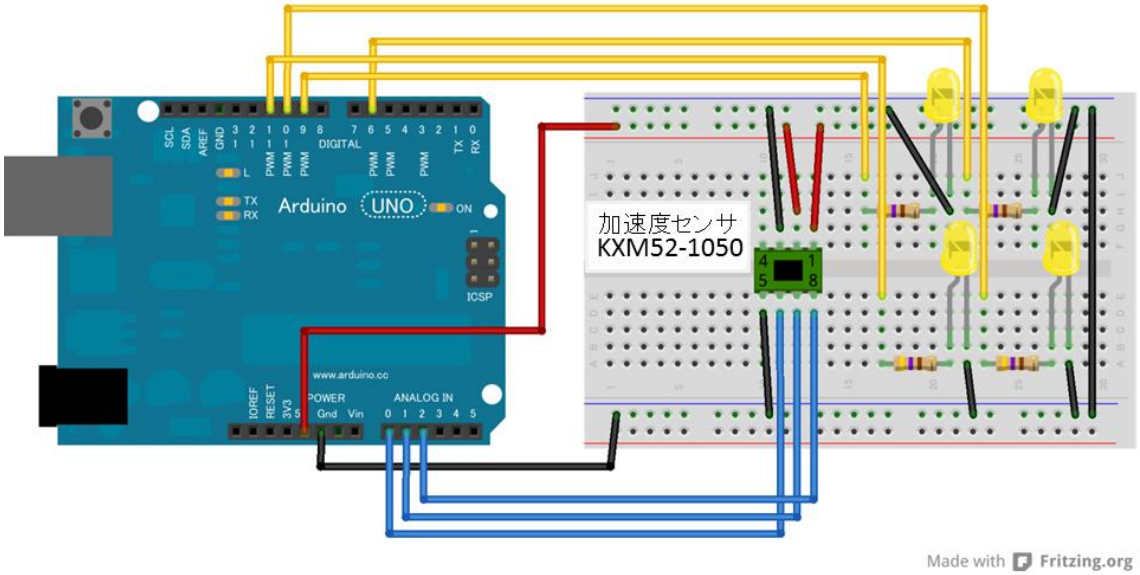


**傾けた時の様子**

また、ろうそくの炎の揺れは「1/f ゆらぎ」という特性を持っていますので、規則性と不規則性が混ざった、ろうそく独特のゆらぐ光り方も再現しました。

## 2 ハードウェアについて

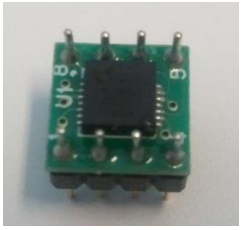
ここではハードウェアの構成について述べます。LED ランプを作る際に参考にしてください。

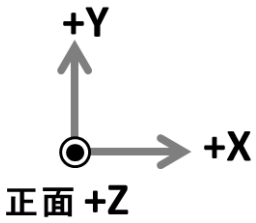



- マイコンボード： Arduino uno
- 加速度センサ：３軸加速度センサモジュール　KXM52-1050
- 抵抗：470Ω

加速度センサは

・6 番ピン：X 軸出力
・7 番ピン：Y 軸出力
・8 番ピン：Z 軸出力






となっております。詳しくは加速度センサに付属の説明書を確認してください。

## 3 ソフトウェアについて

### 3.1 開発環境

本コンポーネントと Arduino のプログラム（スケッチ）は Windows にて動作確認をしております。

開発環境は以下の通りです。

- OS：Windows 7 Professional Service Pack1
- RT ミドルウエア： OpenRTM-aist-1.1.0-RC3 （C++版）
- コンパイラ：Microsoft Visual C++ 2008 Express Edition /Arduino IDE 1.0.2

- CORBA：omniORB 4.1.4
- Eclipse：Eclipse 3.4.2
- CMake：CMake 2.8.10.1

## 3.2 利用しているコンポーネント群

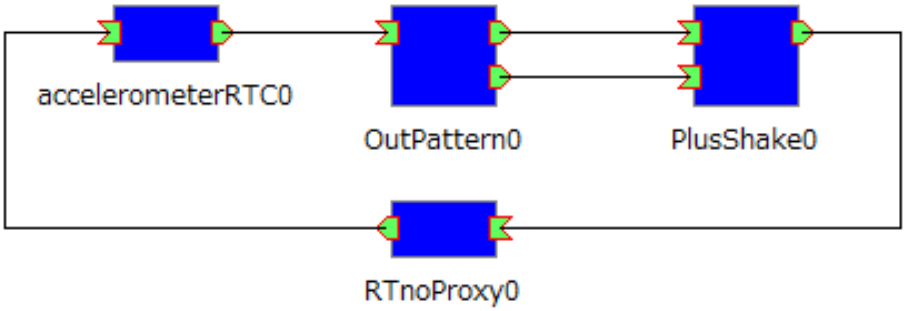
本稿の「外部入力に適応して光が揺らぐ LED ランプ」を実現するために開発したコンポーネント群について紹介します。

- 加速度センサコンポーネント(accelerometerRTC)
  加速度センサの結果を必要な RTC に渡すコンポーネント
- センサ出力変換コンポーネント(OutPattern)
  加速度センサの値によって光り方を変える
- 揺らぎ付加コンポーネント(PlusShake)
  ろうそくの炎のような光のゆらぎを再現する

また、マイコンボードとして Arduino uno を使用しているため、RT コンポーネントとの通信するためのライブラリ群である RTno を使用し、RTnoProxy コンポーネントを通じて通信をしています。

## 3.3 RT システムの構成

各コンポーネントの繋ぎ方を以下に示します。




Arduino では､加速度センサの各軸の出力結果を accelerometerRTC コンポーネントへ、LED の光らせ方（明度の大きさを決める値と光らせる間隔）を PlusShake コンポーネントから RTnoProxy コンポーネントを通じて通信しています。

# 4 コンポーネントの説明

## 4.1 加速度センサコンポーネント(accelerometerRTC)

加速度センサの結果をArduinoからRTnoProxyを通じて、InPortのaccelerometerValueより読み込み、必要なRTCにOutPortのoutValueより出力するコンポーネントです。

- InPort

| 名称 | 型 | 説明 |
|---|---|---|
| accelerometerValue | TimedLongSeq | 加速度センサの各軸の出力結果。配列の0番目から順にX軸、Y軸、Z軸。 |

- OutPort

| 名称 | 型 | 説明 |
|---|---|---|
| outValue | TimedLongSeq | 加速度センサの各軸の値。配列の0番目から順にX軸、Y軸、Z軸。 |

## 4.2 センサ出力変換コンポーネント(OutPattern)

OutPatternコンポーネントはInPortのinValueから加速度センサの出力を読み込み、その値によってLEDの光らせるパターンを決めるコンポーネントです。まず、受け取った各軸の加速度よりLEDランプが静止しているか動いているかを判別します。静止状態なら傾きを計算、動いている状態なら動き出しの方向を計算し、計算結果より光らせるパターンを決定します。パターンとは、どのLEDをどのくらいの明度で光らせるかのことであり、明度0～255の数値です。決定したパターンとLEDランプの状態をOutPortのmove_flagとoutPatternより出力します。

また、加速度の変化量がコンフィギュレーション変数 Threshold より大きいと動いていると判断し、動いていると判断されている間は同じ光るパターンを出力します。

- InPort

| 名称 | 型 | 説明 |
|---|---|---|
| inValue | TimedLongSeq | 加速度センサの各軸の出力結果。この値を元に傾きや動きの方向を計算する。 |

- OutPort

| 名称 | 型 | 説明 |
|---|---|---|
| outPattern | TimedLongSeq | 各 LED の明度の大きさ。0～255 の値。 |
| move_flag | TimedLong | LED キャンドルの状態を知らせる数値。0：静止状態、1：動いている状態 |

- コンフィギュレーション変数

| 名称 | 型 | 説明 |
|---|---|---|
| slopeThreshold | long | 傾いているか判断する閾値。検出された角度と比べる。<br>デフォルト値：10。<br>0 より大きく 90 未満で、スライダーで 5 ずつ変更可能。 |
| Threshold | long | 動いているか判断する閾値。加速度の変化量と比べる。<br>デフォルト値：5<br>1 以上に変更可能。 |

## 4.3 揺らぎ付加コンポーネント(PlusShake)

PlusShake コンポーネントはろうそくの炎のような揺らぎを再現するコンポーネントです。ろうそくの炎の揺れは「1/f ゆらぎ」という特性を持っているので、間欠カオス法を用いて「1/f ゆらぎ」を生成しています。式は次の通りで、$X(t)$ は $0 \le X(t) \le 1$ であり、最大の明るさを 1 として正規化したものです。

$$X(t+1) = \begin{cases} X(t) + 2X(t)^2 & X(t) < 0.5 \\ X(t) - 2(1 - X(t))^2 & X(t) \ge 0.5 \end{cases}$$

InPort の inValue から読み込んだ各 LED の値を最大値とし、間欠カオス法で値を変化させています。

さらに、光る間隔時間もこの間欠カオス法を用いて一定では無いようにしています。間隔時間はコンフィギュレーション変数 interval_MAX で変更できるようになっており、この値が間隔時間の最大値となります。OutPort の outPattern で変化させた LED の値と間隔時間を出力しており、配列の最後が間隔時間になっています。

また、InPort の inFlag の値によって揺らぎの強さや間隔時間を変化させており、値が 1、つまり LED ランプが動いている状態の時は、揺らぎを激しくし間隔時間も短くなるようになっております。

- InPort

| 名称 | 型 | 説明 |
|---|---|---|
| inValue | TimedLongSeq | 各 LED の明度の大きさ。0～255 の値。 |
| inFlag | TimedLong | LED キャンドルの状態を知らせる数値。0：静止状態、1：動いている状態 |

- OutPort

| 名称 | 型 | 説明 |
|---|---|---|
| outPattern | TimedLongSeq | 各 LED の明度の大きさと光る間隔時間。明度の大きさは 0～255 の値。<br>配列の最後が光る間隔時間。<br>配列の大きさは、[LED の数+1]。 |

- コンフィギュレーション変数

| 名称 | 型 | 説明 |
|---|---|---|
| interval_MAX | long | 光る間隔時間の値。（単位はミリ秒）<br>この値が最大の間隔時間となる。<br>デフォルト値：225。 |

## 4.4 RTnoProxy

RTnoProxy コンポーネントはマイコンボードとして使用している Arduino-Uno と RT コンポーネントの通信を実現しているコンポーネントです。このコンポーネントを使用するには、市販の組み込みボード用 RT コンポーネント 対応ライブラリの「RTno」が必要です。RTno についての詳しいことは（http://www.openrtm.org/openrtm/en/node/1740）をご覧ください。

- InPort

| 名称 | 型 | 説明 |
|---|---|---|
| inPattern | TimedLongSeq | 各 LED の明度の大きさと光る間隔時間。明度の大きさは 0～255 の値。<br>配列の最後が光る間隔時間。 |

- OutPort

| 名称 | 型 | 説明 |
|---|---|---|
| acceleration | TimedLongSeq | 加速度センサの各軸の出力結果。配列の 0 番目から順に X 軸、Y 軸、Z 軸。 |

# 5 使用方法

## 5.0 ハードウェアの準備

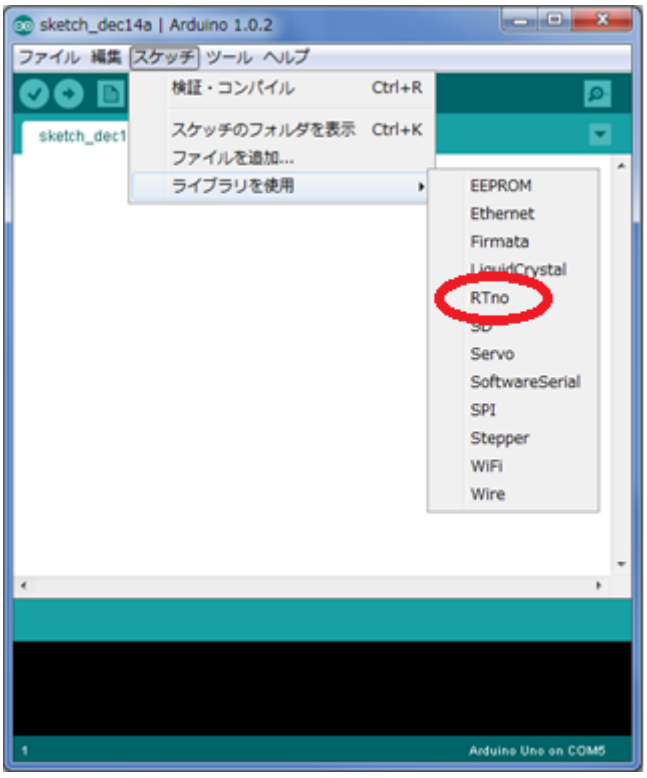
最初に LED ランプの作成をしておいてください。p.1 の「2 ハードウェアについて」を参考にしていただくと作成できると思います。

## 5.1 Arduino と RTno の準備

まず、Arduino と RTno を準備します。Arduino の公式サイト（http://www.arduino.cc/）の Doenload タブから開発環境である Arduino IDE をダウンロードします。

次に RTno の作成者様のサイト（http://ysuga.net/robot/rtm/rtc/rtno）から RTno と PC 側の実行ファイルである RTnoProxy をダウンロードしてください。ページ中段にあるダウンロードのタブから二つともダウンロードできます。RTnoProxy はご自身でお使いの OpenRTM-aist のバージョンに合ったものをインストールしてください。RTno はダウンロードした際のフォルダ名が「ysuga-RTno-XXXX」になっていたら、「RTno」と書き換えてください。

最後にダウンロードして解凍した、arduino-X.X.X フォルダの中の libraries フォルダに先ほどのフォルダ名を変更した RTno フォルダを設置してください。こうすることで arduino.exe を実行した際に「スケッチ→ライブラリを使用」で展開される画面に RTno があると思います。



さらに詳しい RTno と RTnoProxy の準備方法は作成者様のサイトをご覧ください。

## 5.2 Arduino プログラムの編集と書き込み

Arduino プログラムである arduino_ledcandle.ino を Arduino IDE で開いてください。プログラム上部の

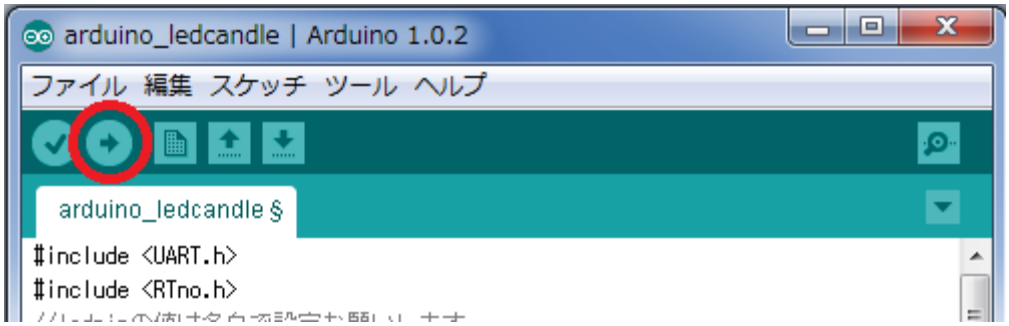
ledpin1、ledpin2、ledpin3、ledpin4

は作成したランプに合わせて設定をお願いします。P.2 の構成図の通りに作成した場合は

ledpin1 = 6、ledpin2 = 9、ledpin3 = 10、ledpin4 = 11

になると思います。

設定が終わったらマイコンボードである Arduino に書き込みをしてください。



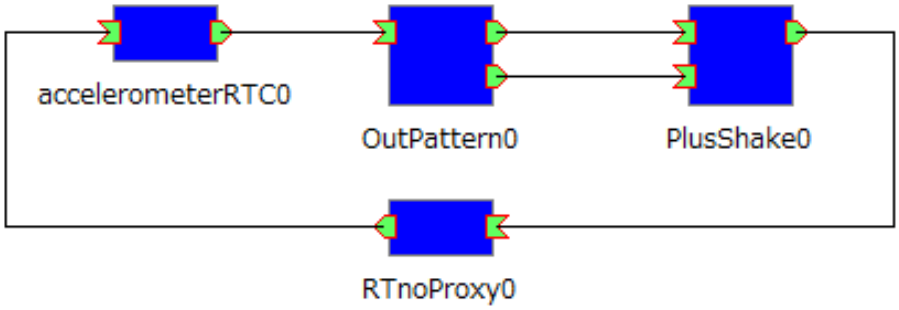

この際に「ツール→マイコンボード」で展開される画面で Arduino Uno が選択されていること、「ツール→シリアルポート」で展開される画面で、Arduino Uno が USB で接続されているポートが正しく選択されていることを確認してから書き込みをしてください。

このプログラムは加速度センサからの出力値を TimedLongSeq として出力し、LED の光らせるパターンと間隔時間を受け取り、実際に LED を analogWrite()で光らせ、delay 関数で光らせる間隔を空けております。

## 5.3 RT System Editor によるシステム構築

RT System Editor のインストール方法と操作方法は OpenRTM-aist のホームページ (http://www.openrtm.org/) を参照してください。

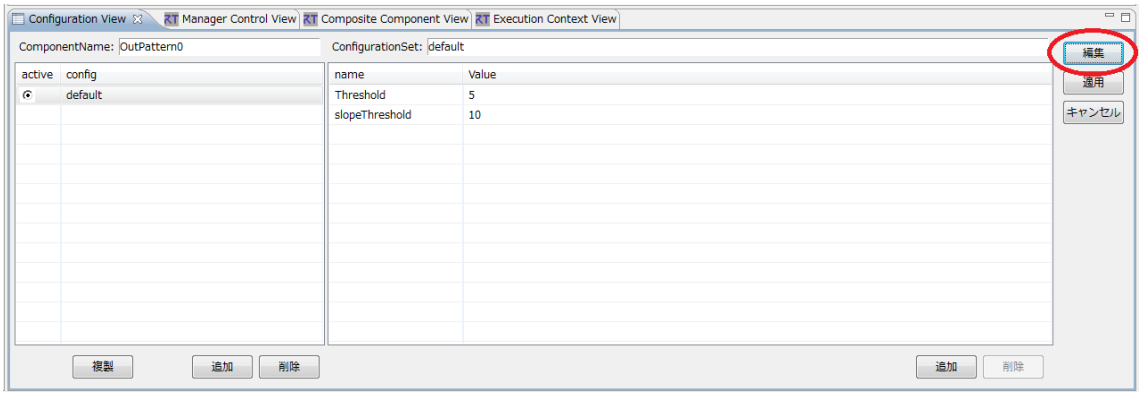
ネームサーバーを起動し接続したら、各コンポーネント同士を接続してください。下図が接続の例です。

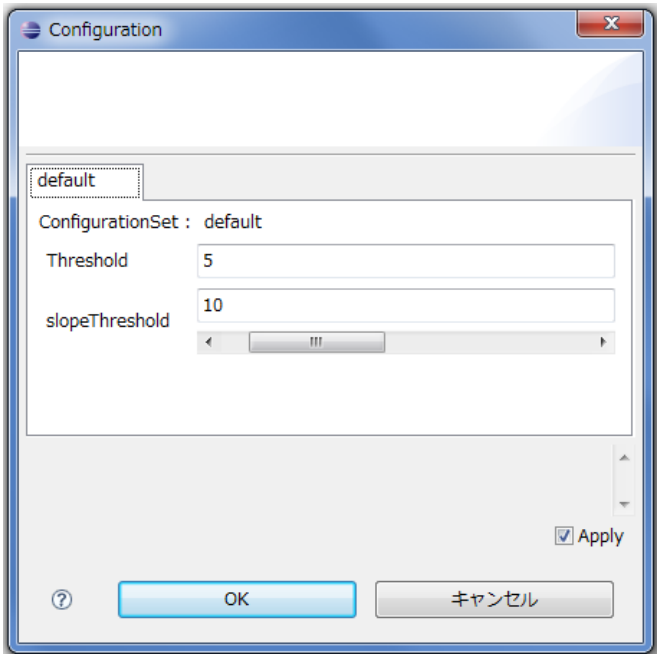


接続が完了したら All Activate を押して実行してください。

## 5.4 コンフィギュレーションの設定

コンフィギュレーション変数の設定は RT System Editor の Configuration View タブの編集ボタンを押して設定画面から設定可能です。




設定を確定する際は、右下の Apply にチェックが入っていることを確認し、OK ボタンを押してください。

各コンポーネントのコンフィグレーション変数に関しては、p.4 の「4 コンポーネントの説明」をご覧ください。

# 6 お問い合わせ

「外部入力に適応し光が揺らぐランプ」は、まだまだ開発途中であり本コンポーネントはバグや改善点などが多々あると思います。報告やご意見、ご要望、本マニュアルの不備などがありましたら、下記まで連絡していただけると幸いです。

お問い合わせ先：
芝浦工業大学 デザイン工学部 デザイン工学科
エンジニアリングデザイン領域メカトロニクスシステム・組込みソフトウェア分野
斉藤 文哉
Email： y09131@shibaura-it.ac.jp