

RT コンポーネント操作マニュアル

# RT ミドルウェアを用いた LED キャンドル の協調動作の実現

2014 年 10 月 31 日版

芝浦工業大学

ロボティクスシステムデザイン研究室

遠藤太貴 近藤貴大 佐々木毅

## 更新履歷

2014 年 10 月 31 日版 第 1 版。

## 目次

1. 本提案の概要.....	1
1.1 提案の背景.....	1
1.2 開発環境.....	1
1.3 本稿で利用しているコンポーネント群 .....	1
1.4 使用したデバイス .....	2
2. コンポーネントの説明 .....	3
2.1 無線情報取得コンポーネント( w_lan ).....	3
2.2 センサ情報取得コンポーネント( sensor ).....	3
2.3 情報取得コンポーネント( getinfo ).....	3
2.4 LED 制御コンポーネント( led ).....	4
3. お問い合わせ.....	5

# 1. 本提案の概要

## 1.1 提案の背景

近年、無線機器の小型化に伴い様々な製品に通信機器が搭載されるようになり、IoT (Internet of Things)の分野が注目され始めている。IoT は世の中に存在する様々な「モノ」をインターネットにつなぐことにより、モノ同士が情報通信することができる。これにより、ネットワーク上のモノを自律的に連動、協調させることが可能になる。

そこで今回、RTC で動作する自律型デバイスを製作し、状態遷移の変更に外部情報を用いることで協調動作を行い、物理デバイスを用いての動的環境における RTC 間関係の分解・再構築の実現例を示す。今回、自律型デバイスの例として、単純な機能を持ち、人とインタラクションが可能である LED キャンドルを用い、外部情報としては、自律型デバイス間の物理的配置情報を使用する。

## 1.2 開発環境

本コンポーネント群は Linux 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: Ubuntu Linux 12.04LTS (32bit 版)
- RT ミドルウェア: OpenRTM-aist-1.1.0-RELEASE (C++版)
- コンパイラ: gcc 4.6.3-1
- CORBA: omniORB 4.1.6-1 / 4.1.5
- Eclipse: Eclipse 3.4.2 + OpenRTM Eclipse tools 1.1.0-RC4
- CMake: Cmake2.8.7-0

## 1.3 本稿で利用しているコンポーネント群

本稿の「LED キャンドルを用いた協調動作」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。

- 無線情報取得コンポーネント(w\_lan)  
無線ネットワーク上でアクセスポイントの情報を取得するコンポーネント
- センサコンポーネント(sensor)  
音センサと赤外線センサの情報を取得するコンポーネント
- 情報取得コンポーネント(getinfo)  
外部の情報を取得し、ポートの変更を行うコンポーネント
- LED 制御コンポーネント(led)  
LED の点滅を制御するコンポーネント

## 1.4 使用したデバイス

今回開発に用いたデバイスは Gumstix 社で開発販売を行っている CPU ボード Overo 及び拡張ボード Summit です。以下に開発手順を記します。

### 手順1 : Linaro のインストール

今回、Linaro の SD カードへの書き込みには Ubuntu12.04 の PC を使用しました。

①PC に git 及び Linaro Image Tools のインストールを行います。

```
$ sudo apt-get install git
```

```
$ sudo apt-get install linaro-image-tools
```

②ハードウェアパックをダウンロードします。

```
$ sudo clone git://github.com/gumstix/Gumstix-Overo-Linaro.git
```

③PC に SD カードを挿入します。

④以下のコマンドを実行します。

```
$ linaro-media-create --rootfs ext3 --mmc </dev/mmcblk0> --dev overo --hwpack
```

```
<hwpack.tar.gz> --binary <rootfs.tar.gz>
```

<dev/mmcblk0> : SD カードのマウントポイント

<hwpack.tar.gz>, <rootfs.tar.gz> : ハードウェアパックを保存した場所

⑤SD カードを PC から取り出し、Overo に挿入し電源を入れると起動します。

### 手順2 : RT ミドルウェアのインストール

① /etc/apt/source.list に以下を追加します。

```
deb http://openrtm.org/pub/Linux/raspbian/ wheezy main
```

② 以下の手順でインストールします。

```
overo $ sudo apt-get install gcc g++ make uuid-dev
```

```
overo $ sudo apt-get install libomniorb4-dev omniidl omniorb-nameserver
```

```
overo $ sudo apt-get install openrtm-aist openrtm-aist-dev
```

```
openrtm-aist-example
```

```
overo $ sudo apt-get install openrtm-aist-python
```

```
openrtm-aist-python-example
```

```
overo $ sudo apt-get install doxygen
```

以上で、Overo への RT ミドルウェアのインストールは完了です。

## 2. コンポーネントの説明

### 2.1 無線情報取得コンポーネント( w\_lan )

w\_lan は、無線インタフェースを通して周囲の詳細な無線情報を取得する iwlist コマンドを実行し、必要な情報を取得、出力するコンポーネントです。

本コンポーネントはアクティブ化すると iwlist コマンドを内部で実行し、周囲にあるアクセスポイントの情報を一覧で取得します。取得できた中から任意の SSID を持つアクセスポイントを判別し、それらのアクセスポイントの RSSI を比較することで、コンポーネントが動作している LED キャンドルの一番近くに存在する他の LED キャンドルの SSID を LED 制御コンポーネントに出力します。

・ InPort

なし

・ OutPort

名称	型	説明
wl_inf	TimedString	自身が一番近くに存在する LED キャンドルの SSID 情報を出力するポート。

### 2.2 センサ情報取得コンポーネント( sensor )

音センサ及び赤外線センサの情報を取得するコンポーネントです。Summit に搭載されている AD 変換器を通して得られるデジタル値を取得します。音センサの情報は 0～2500 の値で出力します。赤外線センサの情報は、各 LED キャンドルに割り振られている ID を識別し、それら ID 情報を出力します。

・ InPort

なし

・ OutPort

名称	型	説明
sound	TimedLong	音センサの情報を出力するポート。0～2500 の整数値を出力するポート。
infrared	TimedLong	赤外線センサ情報を基に判別した ID を出力するポート。

### 2.3 情報取得コンポーネント( getinfo )

周囲の LED キャンドルの情報及びセンサ情報を基に LED の点滅間隔の決定及び、ポートの繋ぎ替えを行うコンポーネントです。

協調動作を実現するため周囲に自身の状態を伝える必要があります。そのため、まず無線情報取得コンポーネント及び赤外線センサ情報を基に、周囲に存在する LED キャンドルを把握します。それらの情報を基にポートのつなぎ替えを行い、データの送信を行います。

繋ぎ替えは、`rtshell` を用いてシェルスクリプトを実行することにより可能にしました。`rtshell` を実行する際に用いる IP アドレスは `localhost` となっていますが、各デバイスで固定の IP アドレスを指定し、それらを ID と関連させて記述します。その結果、`OutPort` から現在の状態を格納している `mydata` が出力され、`InPort` から周囲の情報として `otherdata` が入力されます。

このようにして得られる周囲の情報と音センサ情報を基に、LED の点滅間隔を決定します。その情報を `led` コンポーネントに出力します。

#### ・ InPort

名称	型	説明
network	TimedString	無線の情報を取得するポート。
sound	TimedLong	音センサの情報を取得するポート。
infrared	TimedLong	赤外線センサの情報を取得するポート。
otherdata	TimedLong	周囲の LED キャンドルの情報を取得するポート。

#### ・ OutPort

名称	型	説明
mydata	TimedLong	現在の状態を周囲の LED キャンドルの <code>getinfo</code> コンポーネントに出力するポート。
led	TimedLong	LED の点滅間隔を出力するポート。[ms]

#### ・ Configuration

名称	型	デフォルト値	説明
id	short	0	ポートの繋ぎ替えの情報の選択を行います。 0:赤外線センサの情報 1:無線ネットワークの情報

## 2.4 LED 制御コンポーネント( led )

LED の制御を行うコンポーネントです。情報取得コンポーネントから出力された点滅間隔のデータを基に、LED の制御を行います。

#### ・ InPort

名称	型	説明
led	TimedLong	LED の点滅間隔の数値を取得するポート。[ms]

・ OutPort

なし

### 3. お問い合わせ

本コンポーネント群につきましては、フィードバックに対し随時修正・開発・公開をしているため、まだまだ展開・改善の余地があるものと考えております。提案・要望・バグ報告・マニュアル記述の不備等に関しましては下記までご連絡ください。

【問い合わせ先】

〒108-8548

東京都港区芝浦 3-9-14,611

芝浦工業大学大学院電気電子情報工学専攻

ロボティクスシステムデザイン研究室

遠藤太貴

Email:ma14022@shibaura-it.ac.jp