

# RtcHandle – PythonからRTCを簡単に使う –



末廣 尚士 (産業技術総合研究所)

## 概要:

そこにあるRTCをPython環境から簡単に扱うことができるPythonモジュール

## 特徴:

- ◆Python環境からのRTCの簡単操作
- ◆ロボットシステム構築の支援
- ◆RTCおよびロボットシステムのデバッグツール
- ◆RTCのプロトタイピング
- ◆ロボット作業アプリケーションの開発・実行

## 基本クラス:

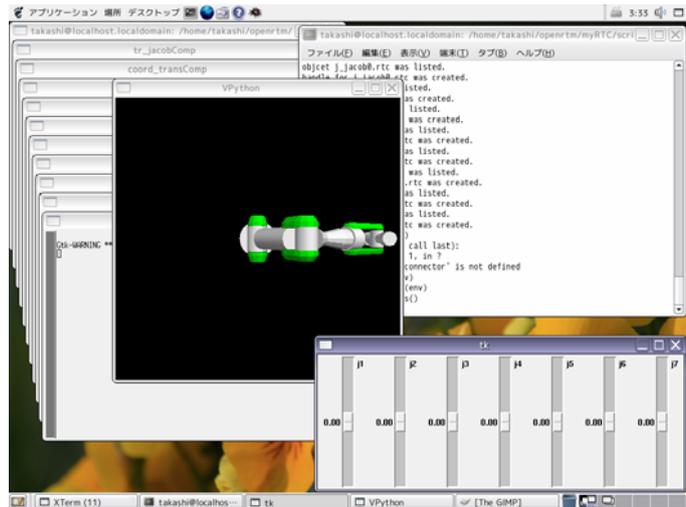
- ◆RtmEnv、NameSpace
- ◆RtcHandle
- ◆Port
- ◆Connector

## 開発環境:

- ◆OpenRTM-aist-python-0.4.1
- Linux, Windows とともに検証済み

## ライセンス (公開条件):

EPL



Pythonスクリプトで11個のRTCを接続して、活性化した例のスクリーンショット

## 連絡先:

独立行政法人産業技術総合研究所  
 知能システム研究部門  
 末廣 尚士  
 〒305-8568 茨城県つくば市梅園1-1-1 中央第2  
 email: t.suehiro <at> aist.go.jp  
 url: http://staff.aist.go.jp/t.suehiro/rtm/



## RtcHandleの基本クラス

- RtmEnv  
orbやnaming serviceなど、OpenRTMの環境情報を保持する。
- NameSpace  
naming serviceとそこに登録されているcorbaオブジェクト、RTCおよび対応するRtcHandleを保持する。
- RtcHandle  
RTCの情報の保持およびその機能へのアクセスを提供する。
- Port  
RTCのPortに対応するクラス。ポートの種類によって以下の3つのサブクラスがある。このクラスを通してポートのサービスや入出力へのアクセスも直接行うことができる。
  - RtcService: サービスポート
  - Rtclnport: 入力ポート
  - RtcOutput: 出力ポート
- Connector  
RTCのPort間の接続情報を管理するConnectorProfileに対応するクラス。接続Portの種類により2つのサブクラスがある。
  - ServiceConnector: サービスポートの接続
  - IOConnector: 入出力ポートの接続

その他、入出力ポートやサービスポートに直接アクセスすることもできる

## 使用例 モジュールの読み込み

```
$ python
...
>>> from rtc_handle import *
```

## OpenRTM環境の構築

```
>>> env =
RtmEnv(sys.argv, ["localhost:9876"])
```

## RtcHandleの取り出し

```
>>> pa10fk =
env.name_space["localhost:9876"].rtc_handles["pa10fk0.rtc"]
```

## Connectorの生成、接続

```
>>> con1 =
IOConnector([pa10fk.outports["frame"],
frm_ctrl.inports["ref_frm"]])
>>> con1.connect()
```

## RTCの活性化

```
>>> pa10fk.activate()
```

RTCLinkを使わずに、Pythonスクリプトで図のようなシステムができる。

