

# rtshell入門

Geoffrey Biggs

産業技術総合研究所

ロボットイノベーション研究センター

ロボットソフトウェアプラットフォーム研究チーム

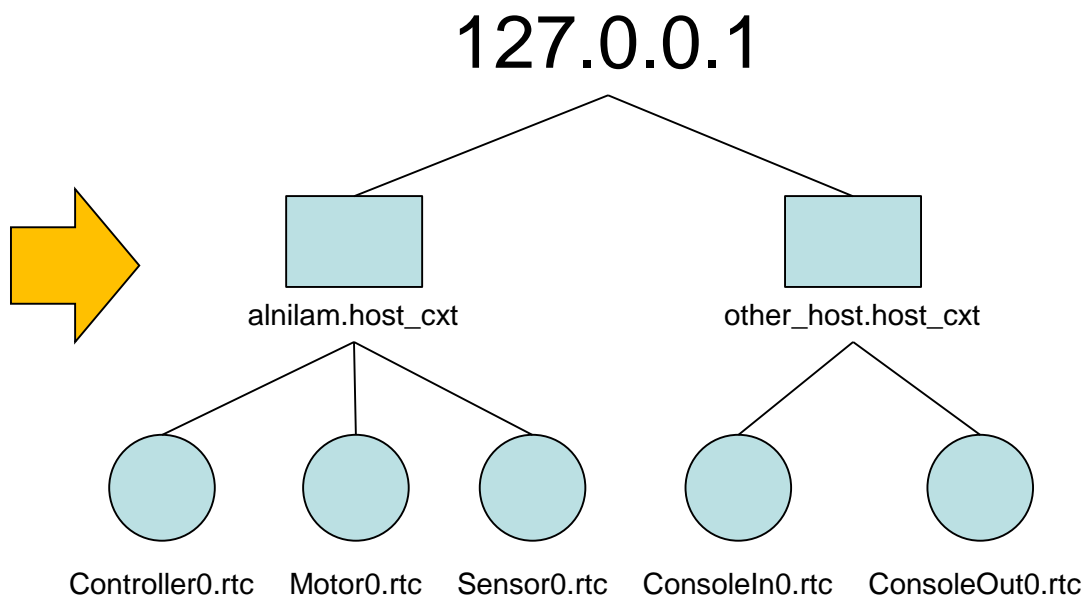
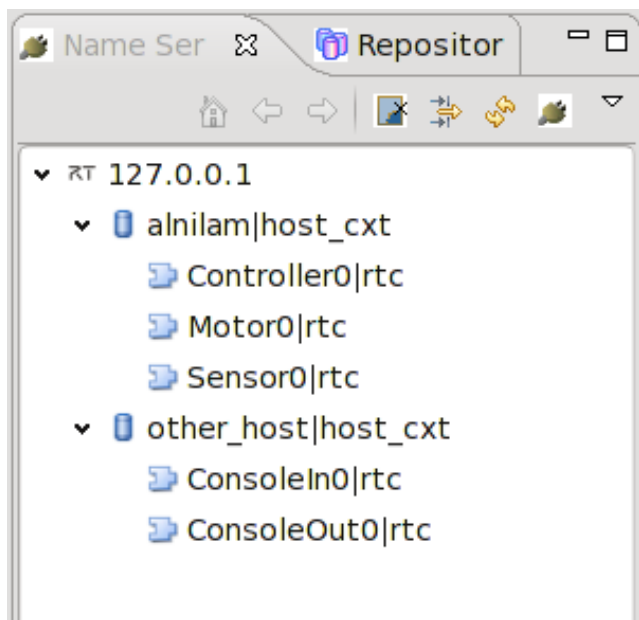
# rtshellって何?

コマンドラインでRTコンポーネントやRTシステムを制御するツール

- RTSystemEditorと同等の機能を持つ
- システム管理(一気に起動等)
- テストやデバッグ用のコマンドを持つ
- RTSystemEditorが使えない場合に
  - ssh環境
  - GUIもネットワークもない
  - バッチファイル等

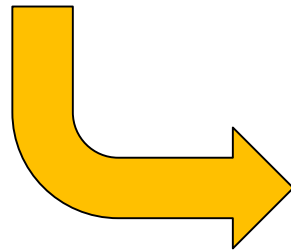
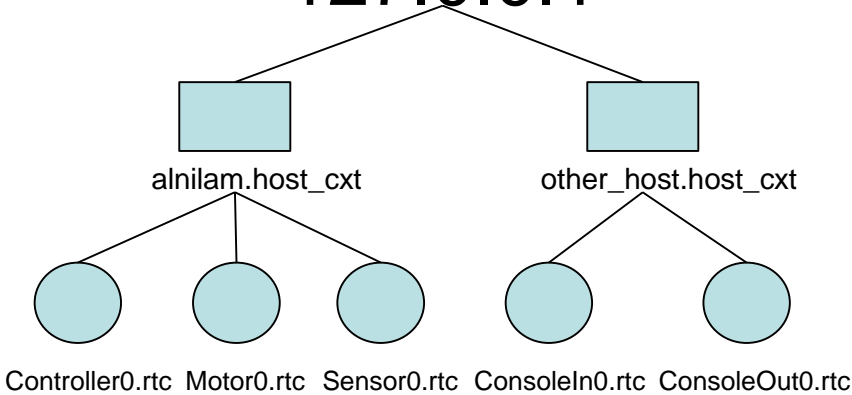
# rtshellのバーチャルファイルシステム

- ファイルシステムのツリーと同じようにRTC等を見る



# rtshellのバーチャルファイルシステム

127.0.0.1



```

$ rtl
Localhost/
$ rtcwd localhost/
$ rtl
alnilam.host_cxt/  other_host.host_cxt/
$ rtcwd alnilam.host_cxt
$ rtl
Controller0.rtc  Motor0.rtc  Sensor0.rtc
$ rtcwd ..
$ rtl
alnilam.host_cxt/  other_host.host_cxt/
$ rtcwd other_host.host_cxt/
$ rtl
ConsoleIn0.rtc  ConsoleOut0.rtc
$ rtpwd
/localhost/other_host.host_cxt/
  
```

# とくに便利な機能

- システム管理
  - Rtshellで全コンポーネントの接続・起動等が一機にできる
  - 半分起動中でも全部起動中にも（一個のコンポーネントがクラッシュしたら便利！）
- システムの確認
  - 全部がちゃんと起動中か？接続されたか？
- コンポーネントデバッグ
  - デバッグ用のコンポーネント作成せずにテスト
- データログ作成
  - ハードウェアなしで開発

# システム管理

- rtshell で全システムを管理する
  - RTSPProfile ファイルを使う
  - RTSystemEditor もRTSPProfile を使う
- **rtresurrect** で復元する
- **rtstart** で起動させる
- **rtstop** で停止する
- **rtteardown** で消す

```
$ rtresurrect my_system.rtsys  
$ rtstart my_system.rtsys
```

# システムのインスペクション

- システムを起動したら確認したい
  - **rtcheck** でライブなシステムをRTSProfile ファイルと比べる
- **rtstodot** でRTSProfile のモデルを可視化する
  - **rtcryo** と組み合わせたらライブなシステムを可視化する
  - (Windowsで利用できない)

# コンポーネントのデバッグ

- rtshell でコンポーネントが送信し受信するデータのインスペクション
  - `rtprint` でアウトポートからのデータを表示する
  - `rtinject` でインポートにデータを送る
- データロギングでテスト

```
$ rtprint /localhost/ConsoleIn0.rtc:out  
  
$ rtinject /localhost/ConsoleOut0.rtc:in  
-c 'RTC.TimedLong({time}, 42)'
```



# データログ作り

- **rtlog** でコンポーネントが送るデータをファイルに記録する
- オフラインで **rtlog** によってファイルからのデータを再生する
  - ハードウェアなしでアルゴリズムのテスト

```
$ rtlog -f log.rtlog  
    /localhost/ConsoleIn0.rtc:out.numbers  
  
$ rtlog -f log.rtlog -p  
    /localhost/ConsoleOut0.rtc:in.numbers
```

# ドキュメント

- 全コマンドで **-h** オプションによってヘルプを取得

```
geoff ~ $ rtinject -h
```

```
Usage: rtinject [options] <path1>:<port1> [<path2>:<port2>...]
```

```
Write a constant value to one or more ports.
```

```
Options:
```

```
--version          show program's version number and exit
-h, --help         show this help message and exit
-c CONST, --const=CONST
                   The constant value to send, as a Python expression. If
                   not specified, values will be read from standard in.
-m MODULES, --mod=MODULES
                   Extra modules to import. If automatic module loading
                   struggles with your data types, try listing the
                   modules here. The module and its __POA partner will be
                   imported.
-n MAX, --number=MAX
                   Specify the number of times to write to the port.
                   [Default: 1]
-p PATHS, --path=PATHS
                   Extra module search paths to add to the PYTHONPATH.
-r RATE, --rate=RATE
                   Specify the rate in Hertz at which to emit data.
                   [Default: 1.0]
-t TIMEOUT, --timeout=TIMEOUT
                   Write data for this many seconds, then stop. This
                   option overrides --number. [Default: -1]
-v, --verbose      Output verbose information. [Default: False]
```

# ドキュメント

- UNIX型では深い説明(使い方の例を含む)がmanページに

RTINJECT(1)

RTINJECT(1)

User commands

**NAME**

rtinject - inject data into ports

**SYNOPSIS**

rtinject [options] <path1:port1> [path2:port2...]

**DESCRIPTION**

Writes constant values to one or more ports. By default, the value is written once. Options are available to write a set number of times, or write regularly for a specified length of time.

A connection will be made to the port using the default connection settings compatible with the port.

**OPTIONS**

**-c** CONST, **--const=CONST**

The constant value to send, as a Python expression. If not specified, values will be read from stdin. Any occurrences of **{time}** in the constant expression will be replaced with the current time.

**-m** MODULES, **--mod=MODULES**

Extra modules to import. If automatic module loading struggles with the constant's data types, try listing the modules here. The module and its **\_\_POA** partner will be imported.

**-n** MAX, **--number=MAX**

lines 1-26

\$ man rtinject

# ドキュメント

- Windows型では深い説明がhtmlページにある



The screenshot shows a web browser window with the following content:

**rtinject**

ポートにデータを送る

Author: Geoffrey Biggs and contributors  
Date: 2010-10-24  
Copyright: EPL-1.0  
Version: 3.0  
Manual section: 1  
Manual group: User commands

**書式**

`rtinject [options] <path1:port1> [path2:port2...]`

**概要**

値を一つ以上のポートに送ります。デフォルトは一回のみ送ります。複数回や定期的に送ることもできます。

<http://www.openrtm.org/pub/OpenRTM-aist/tools/rtshell/3.0/ja/rtshell.html>

# チュートリアル

- openrtm.orgにチュートリアルがある
  - rtshell によるRTシステムの管理

<http://openrtm.org/openrtm/ja/node/5014/>

- rtshell でコンポーネントデータの保存・再生

<http://openrtm.org/openrtm/ja/node/5015/>

- Youtube にスクリーンキャストがある

[www.youtube.com/user/OpenRTM](http://www.youtube.com/user/OpenRTM)

[www.youtube.com/playlist?list=PLE06F481CC7089B9A](http://www.youtube.com/playlist?list=PLE06F481CC7089B9A)

# まとめ

- rtshell で開発効率をあげる
- RTSystemEditor と組み合わせたら全部の力を得る
  - RTSystemEditorでシステム作成
  - Rtshellとバッチファイルでシステム管理
- 説明書に参照してください
  - 全部のコマンドで -h オプションによってヘルプを取得
  - man ページ・HTMLページには深い説明がある
  - 使い方の例もたっぷり