

rtshell入門

宮本 信彦

国立研究開発法人産業技術総合研究所
インダストリアルCPS研究センター
ソフトウェアプラットフォーム研究チーム



資料

- 配布資料の「WEBpage」のHTMLファイルを開く
 - チュートリアル(rtshell入門、Raspberry Pi Mouse) _ OpenRTM-aist.html
- もしくは以下のリンク
 - <https://openrtm.org/openrtm/ja/node/7097>



チュートリアル(rtshell入門、Raspberry Pi Mouse)

ビュー | 編集 | アウトライン | 軌跡 | 翻訳 | Devel

いいね! Facebookに登録して、友達の「いいね!」を見てみましょう。

Table of contents

- はじめに

はじめに

ここではシミュレータ上のRaspberry Piマウスを操作するRTシステムの起動、終了を自動化するバッチファイル、シェルスクリプトの作成方法について説明します。

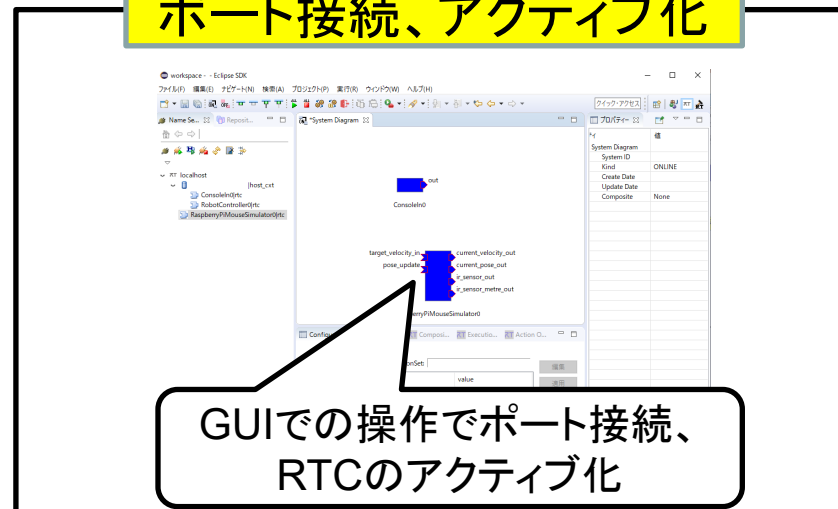
RTシステム起動の自動化

- 先ほどもでのRTシステム起動手順を再起動時も実行するのは手間がかかる

RTC起動



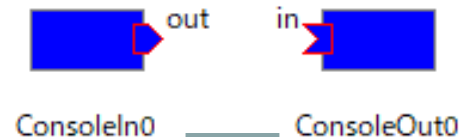
ポート接続、アクティブ化



- これらの処理を自動化するバッチファイル、シェルスクリプトの作成手順を説明する。

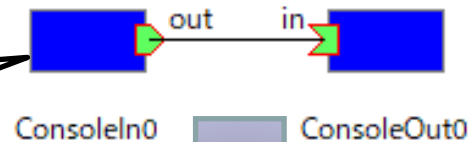
rtshell

- rtshellはコマンドラインでRTコンポーネントやRTシステムを操作するツール



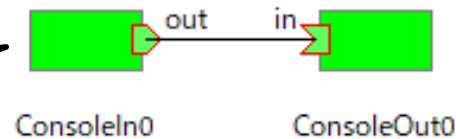
```
> rtcon localhost/ConsoleIn0.rtc:out localhost/ConsoleOut0.rtc:in
```

rtconコマンドで
ポートの接続



```
> rtact localhost/ConsoleIn0.rtc localhost/ConsoleOut0.rtc
```

rtactコマンドで
アクティブ化

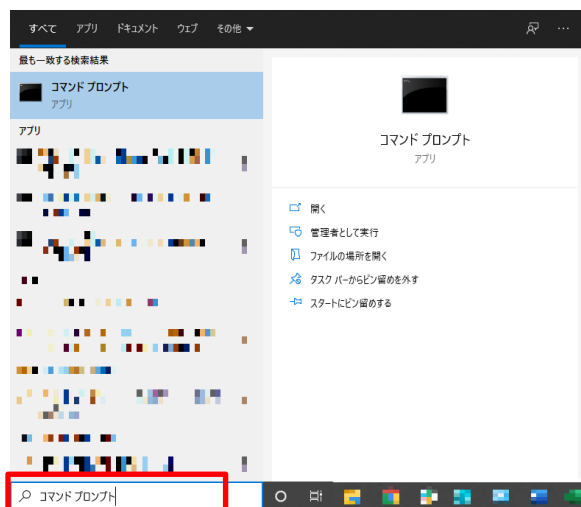


RTシステムの起動、終了を自動化

- 今回開発した「シミュレータ + RobotController」のシステムを起動、終了するための手順は以下のとおりである。
 1. RaspberryPiMouseSimulatorコンポーネント、RobotControllerコンポーネントを起動する。
 2. ポートをコネクタで接続する
 3. RTCをアクティブ化する
 4. RTCを終了する
- 1～4を実行するスクリプトファイル(バッチファイル、シェルスクリプト)を作成し、簡単にRTシステムを起動、終了できるようにする
 - 1については以下のプログラムを実行するコマンドを記述するだけ
 - RaspberryPiMouseSimulatorComp.exe
 - RobotControllerComp.exe
 - 2、3、4についてはrtshellのコマンドを使用する

事前準備

- この実習ではコマンドラインによる操作を行うため、コマンドプロンプト(Windows)、ターミナル(Ubuntu)を起動してください。
 - Windowsの場合は左下の「検索」に「コマンド プロンプト」と入力して検索する。

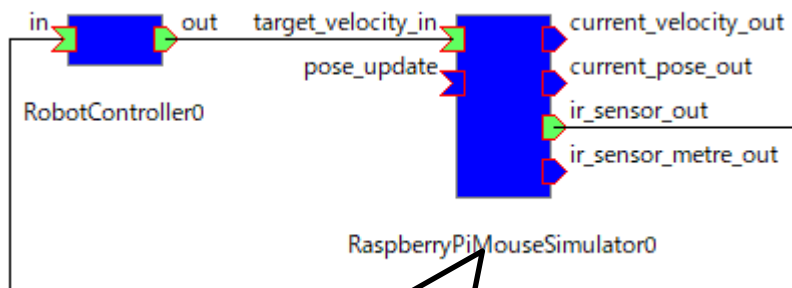


- コマンドプロンプトが起動したら「rtls」を入力してみてください。「rtls」は、内部コマンドまたは外部コマンド、操作可能なプログラムまたはバッチ ファイルとして認識されていません。」と表示された場合、Pythonのインストールフォルダ内のScriptsフォルダが環境変数Pathに設定されていません。
 - 「C:¥Python38¥Scripts」といったフォルダを環境変数Pathに追加する必要がありますが、分からない場合は質問してください。

ポート接続の自動化

1. RTSystemEditor上でデータポートを接続する。
 2. コネクタの接続情報をファイルに保存する。
 3. 再起動時にファイルの情報からコネクタを復元する。
- } 前準備

```
> rtcryo -o C:\work\robotcontroller.xml localhost
```



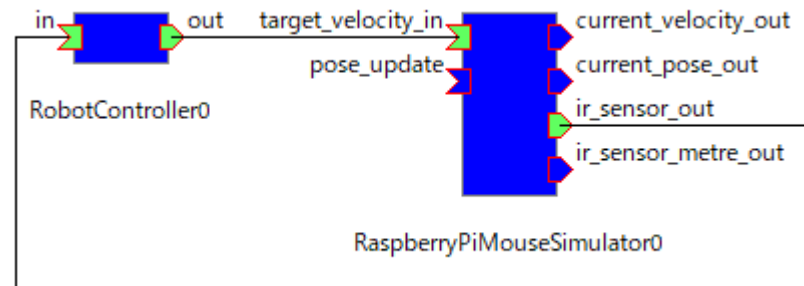
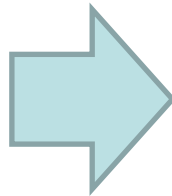
RTSystemEditorで
ポートを接続する



rtcryoコマンドで接続情報な
どをXMLファイルに保存する

RTシステムの保存、復元

> rtsurrect C:\¥work¥robotcontroller.xml

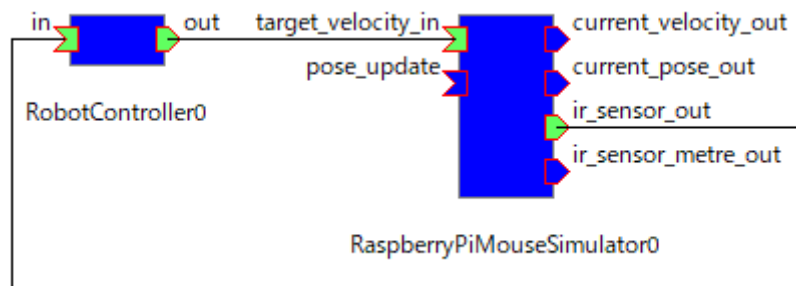


rtsurrectコマンドでXMLファイルの
情報からポートを接続する。

- 作成したXMLファイルからポートの接続情報を読み込み、元のシステムを復元できる。

RTシステムの保存、復元

- rtcryoコマンドを試してみる
 - RTSystemEditorでポートを接続した状態にする。



- rtcryoコマンドでシステムの情報XMLファイルに保存する

```
> rtcryo -o C:\work\robotcontroller.xml localhost
```

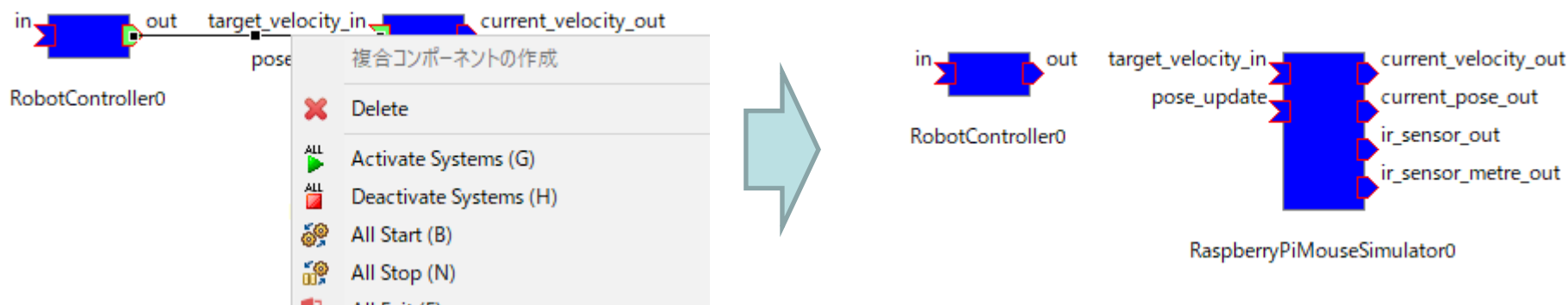
XMLファイルを保存するパスを指定する。適宜パスは分かりやすい場所に変更してください

ネームサーバーを指定する。
今回はlocalhostのみ。

- Pyyamlがインストールされていないとエラーになる
 - pip install pyyaml

RTシステムの保存、復元

- rtsurrectコマンドを試してみる
 1. RTSystemEditorでコネクタを切断した状態にする。
 - コネクタを切断するには、コネクタを選択してDeleteキーを押すか、右クリックしてDeleteを選択する。



2. rtsurrectコマンドでポートの接続情報を復元する

```
> rtsurrect C:\¥work¥robotcontroller.xml
```

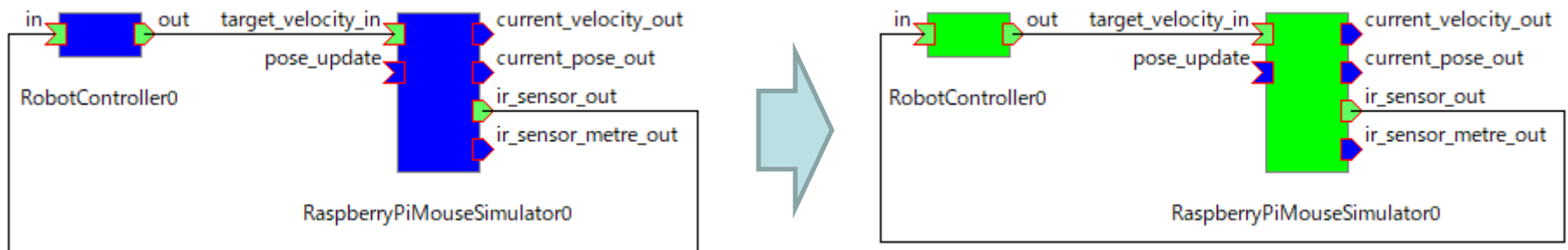
XMLファイルのパスはrtcryoコマンドで保存したパスに変更する

RTCのアクティブ化の自動処理

- rtstartコマンドでXMLファイルに保存したシステムのRTCをアクティブ化する
 - 以下のコマンドを試してみてください

```
> rtstart C:\¥work¥robotcontroller.xml
```

XMLファイルのパスはrtcryoコマンドで保存したパスに変更する

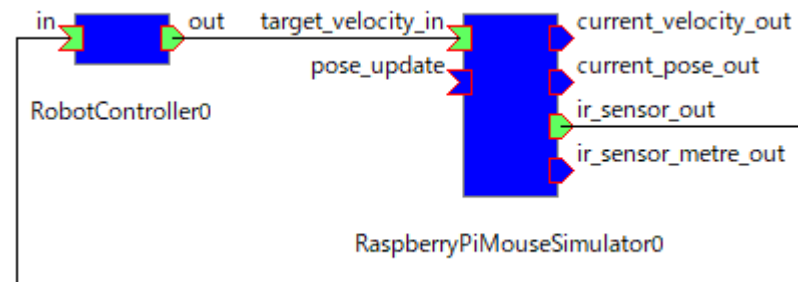
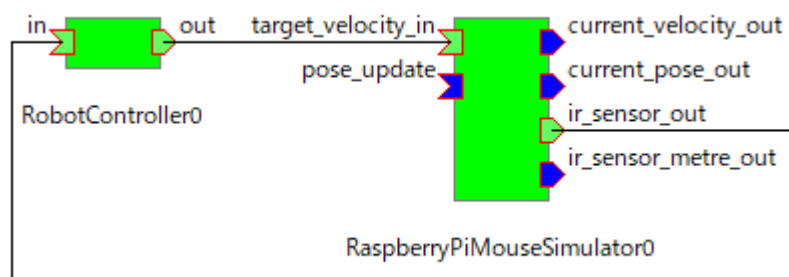


RTCの非アクティブ化

- rtstopコマンドでXMLファイルに保存したシステムのRTCを非アクティブ化する
 - 以下のコマンドを試してみてください

```
> rtstop C:¥work¥robotcontroller.xml
```

XMLファイルのパスはrtcryocommandで保存したパスに変更する

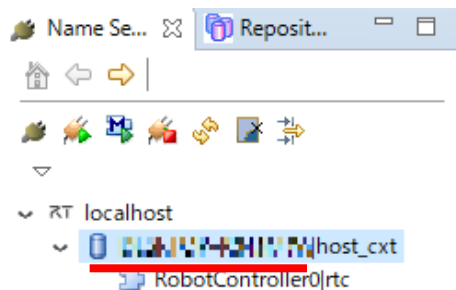


RTCの終了の自動化

- rtxexitコマンドでXMLファイルに保存したシステムのRTCを非アクティブ化する
 - 以下のコマンドを試してみてください

```
> rtxexit localhost/RaspberryPiMouseSimulator0.rtc  
> rtxexit localhost/%COMPUTERNAME%.host_cxt/RobotController0.rtc
```

デフォルトの設定でRobotControllerはネームサーバーで
ホスト名.host_cxtの下に登録される。

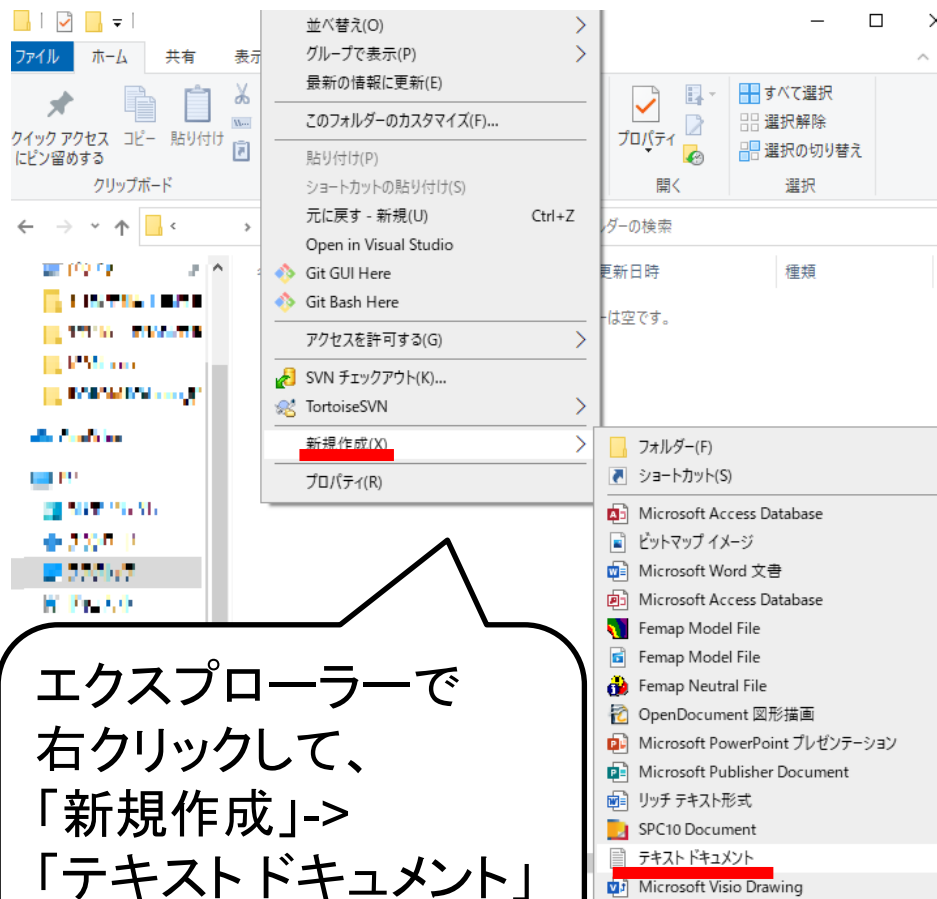


※Ubuntuの場合は「%COMPUTERNAME%」
を「\${HOSTNAME}」に変更する。

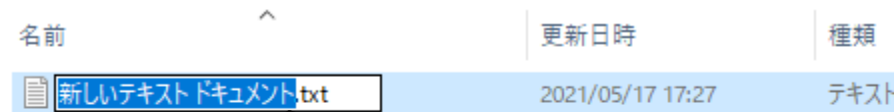
スクリプトファイルの作成

- rtshellのコマンドを用いて、RTシステムの起動、終了を自動化するバッチファイル(Windows)、シェルスクリプト(Ubuntu)を作成する。
 - まずは適当な場所に以下のファイルを作成
 - Windowsの場合はバッチファイル(拡張子.bat)
 - 今回は「robotcontroller_start.bat」、
「robotcontroller_exit.bat」というファイルを作成
 - テキストファイルを新規作成後、名前を変更することで作成する。
 - エクスプローラーで拡張子を非表示にしている場合は注意
 - Ubuntuの場合はシェルスクリプト(拡張子.sh)
 - 今回は「robotcontroller_start.sh」、
「robotcontroller_exit.sh」というファイルを作成

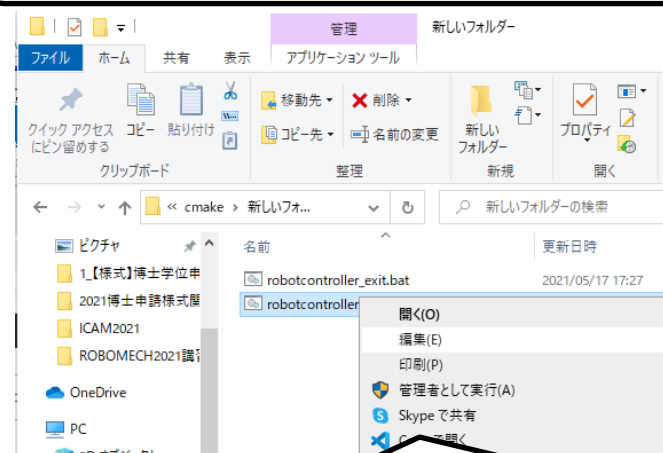
バッチファイル作成,編集



エクスプローラーで
右クリックして、
「新規作成」->
「テキストドキュメント」
を選択する



名前を
「robotcontroller_start.bat」、
「robotcontroller_exit.bat」に
変更する。



ファイルを右クリックして「編集」を選
択する。

起動自動化のスク립トファイルの作成

- まずは「robotcontroller_start.bat」、
「robotcontroller_start.sh」を編集する。
- RaspberryPiMouseSimulator、RobotControllerのプログラムを実行するコマンドを記述する。
 - バッチファイル
 - ファイルのパスは適宜変更する

```
start "" /d C:¥workspace¥RobotController¥build¥src¥Release RobotControllerComp.exe  
start "" /d C:¥work¥RTM_Tutorial¥EXE RaspberryPiMouseSimulatorComp.exe  
timeout 2
```

- シェルスクリプト
 - ファイルのパスは適宜変更する

```
cd ~/workspace/RobotController/build/src/  
./RobotControllerComp&  
cd ~/RasPiMouseSimulatorRTC/build/src  
./RaspberryPiMouseSimulatorComp&  
sleep 2
```


起動自動化のスク립トファイルの作成

- RTシステムを復元、RTCのアクティブ化を実行するコマンドを記述する。
 - XMLファイルのパスは適宜変更する。

```
rtresurrect C:¥work¥robotcontroller.xml
```

```
rtstart C:¥work¥robotcontroller.xml
```

- 記述が完了したら、robotcontroller_start.bat、robotcontroller_start.shを実行してシミュレータが起動するかを確認してください。
 - RTCが起動しない場合、実行ファイルのパスが違う可能性があるため確認してください。

終了自動化のスク립トファイルの作成

- まずは「robotcontroller_exit.bat」、
「robotcontroller_exit.sh」を編集する。
- RTC終了のコマンドを記述する。

– Windows

```
rtexit localhost/RaspberryPiMouseSimulator0.rtc
rtexit localhost/ %COMPUTERNAME%.host_cxt/RobotController0.rtc
```

– Ubuntu

```
HOSTNAME=`hostname`
rtexit localhost/RaspberryPiMouseSimulator0.rtc
rtexit localhost/${HOSTNAME}.host_cxt/RobotController0.rtc
```

- 記述が完了したら、robotcontroller_exit.bat、
robotcontroller_exit.shを実行して以下の事を確認してください。
 - シミュレータが終了する(ウィンドウが消える)。
 - ネームサーバーからRTCが消える。