

教育用ロボットアームコンポーネント群 操作マニュアル

名城大学メカトロニクス工学科
ロボットシステムデザイン研究室

2016 年 12 月 7 日

内容

1.	はじめに	3
1.1	目的	3
1.2	本書を読むにあたって	3
1.3	動作環境	3
1.4	開発環境	3
2.	シミュレーション, 実機操作の準備	4
2.1	コンポーネントのビルド	4
2.2	dialout にユーザーの追加	4
2.3	rtshell のインストール	4
3.	シミュレータを用いた CRANE+ 操作方法	5
3.1	eclipse の起動	5
3.2	Choreonoid の起動	7
3.3	シミュレータ操作コンポーネントの実行	9
3.4	コンポーネント接続	10
3.5	シミュレーションの実行	11
4.	CRANE+ 実機の制御	13
4.1	CRANE+ 制御コンポーネントの実行	14
4.2	コンポーネント接続	14
4.3	コンポーネントの Activate	14
5.	シェルスクリプトが実行できないとき	17
5.1	シミュレータを用いた CRANE+ 操作	17
5.1.1	ChoreonoidCRANEControllerRTC の実行	17
5.1.2	ManipulatorControlSampleRTC の実行	17
5.2.3	コンポーネント接続	18
5.1.4	シミュレーションの実行	18
5.2	CRANE+ 実機の制御	19
5.2.1	CranepiusRTC の実行	19
5.2.2	ManipulatorControlSampleRTC の実行	19
5.2.3	コンポーネント接続	19

1. はじめに

1.1 目的

本書の目的は教育用ロボットアームコンポーネント群の実行及び操作方法を解説することである。

1.2 本書を読むにあたって

本書は、RT ミドルウェアに関する基礎知識を有した利用者を対象としている。

また、本書を読む前に同ファイル内の Choreonoid 導入マニュアルを参考にして Choreonoid をインストールしておくこと。RTC の詳細な仕様等については同ファイル内の仕様解説マニュアルを参考にすること。

1.3 動作環境

本 RTC の動作確認環境を以下に示す。

OS	Oracle VM VirtualBox Ubuntu14.04
RT ミドルウェア	OpenRTM-aist-1.1.2
Choreonoid	Choreonoid-1.5.0

1.4 開発環境

本 RTC の開発環境を以下に示す。

OS	Oracle VM VirtualBox Ubuntu14.04
言語	C++
RT ミドルウェア	OpenRTM-aist-1.1.2
Choreonoid	Choreonoid-1.5.0

2. シミュレーション，実機操作の準備

2.1 コンポーネントのビルド

ダウンロードしたコンポーネントをビルドする．ダウンロードしたディレクトリ内に `script` というディレクトリがあるので，その中の `CompBuild.sh` を実行する．

```
~$ cd CRANE-simulation/script
/CRANE-simulation/script~$ sh CompBuild.sh
```

コンポーネントのビルドが完了する．

2.2 dialout にユーザーの追加

また，本コンポーネントはシリアル通信を行うためデフォルトの設定では `root` 権限で実行しなければシリアル通信をすることができない．ダウンロードしたシェルスクリプトを用いて実行する場合は `dialout` にユーザーを登録する必要があるため，以下のコマンドを実行する．

```
~$ sudo gpasswd -a ユーザー名 dialout
```

2.3 rtshell のインストール

本マニュアルではシェルスクリプトを使った `RTC` の実行やコンポーネント接続を行う．そのため，`rtshell` をインストールする必要がある．以下のサイトを参考にして `rtshell` をインストールすること．

－Ubuntu14.04 での `rtshell` の環境構築

http://www2.meijo-u.ac.jp/~kohara/cms/technicalreport/ubuntu_rtshell_install

3. シミュレータを用いた CRANE+ 操作方法

3.1 eclipse の起動

①ネーミングサービスを実行する

```
~$ rtm-naming
```

②eclipse を起動する.

③eclipse が立ち上がったたら, 右上のその他から RT System Editor を選択する. (図 1~3)

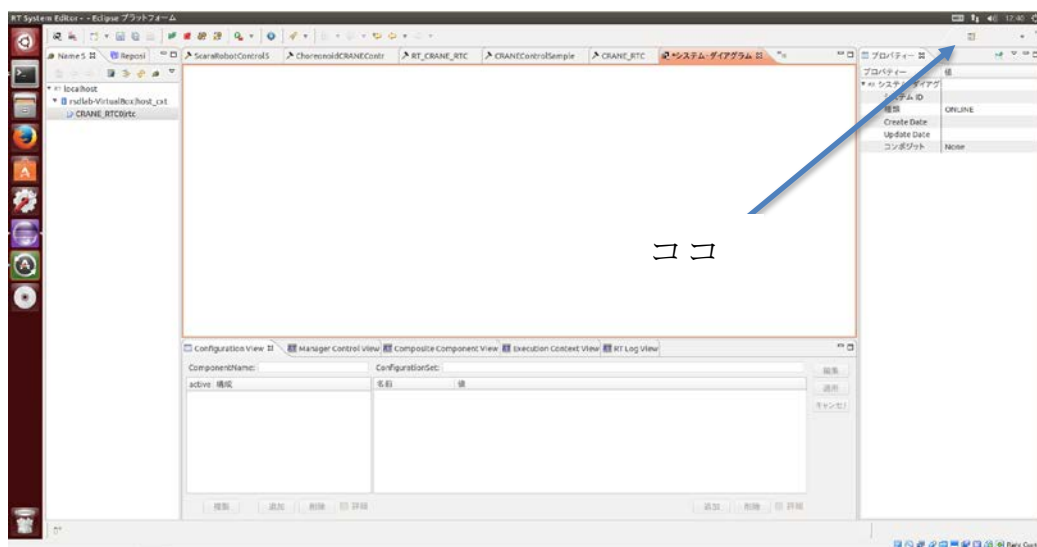


図 1 eclipse

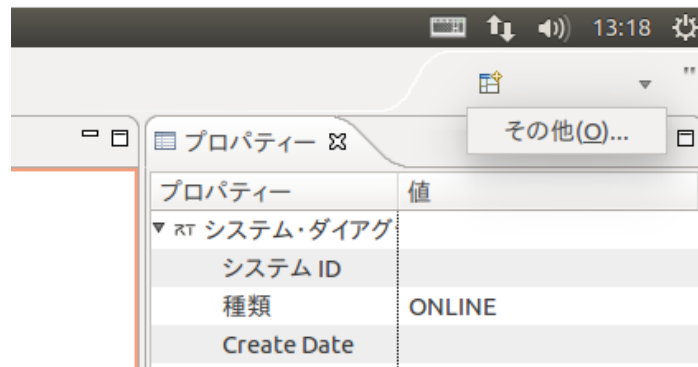


図 2 その他

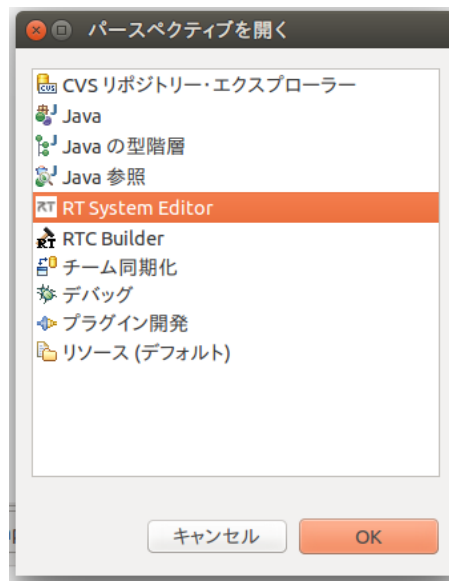


図 3 RT System Editor

⑤ネームサービスに何も無い場合，図 4 のようにネームサーバーの追加から localhost を追加する．（このとき失敗する場合ネーミングサービスが立ち上がっていない→2.の①へ）

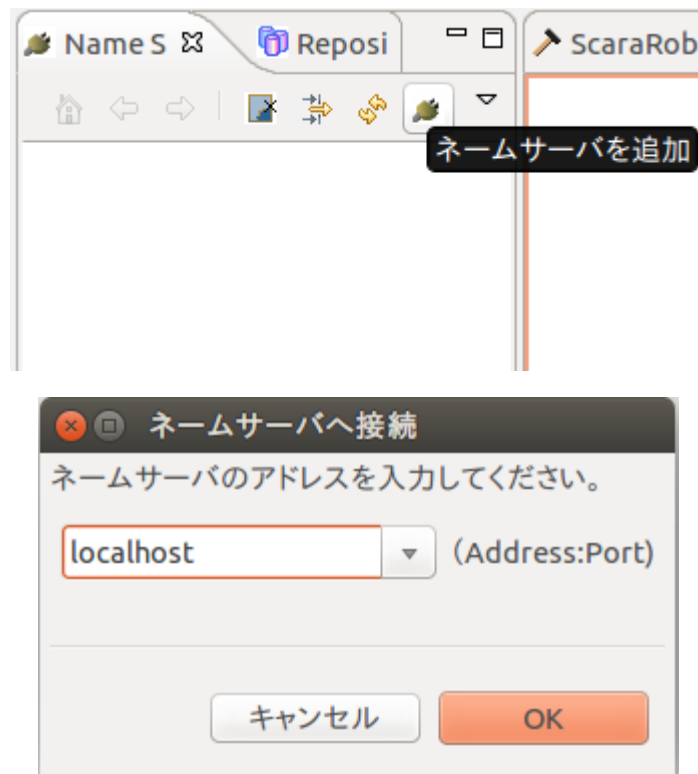


図 4 ネームサーバーの追加

3.2 Choreonoid の起動

①Choreonoid を起動する

```
~$ cd choreonoid-1.5.0/build/bin/  
/choreonoid-1.5.0/build/bin~$ ./choreonoid
```

コレオノイドのファイル→新規からプロジェクトの読み込みを選択し、
CRANE-simulation/model_project /内の CRANEsimulation.cnoidを読み込む(図 5, 図 6)

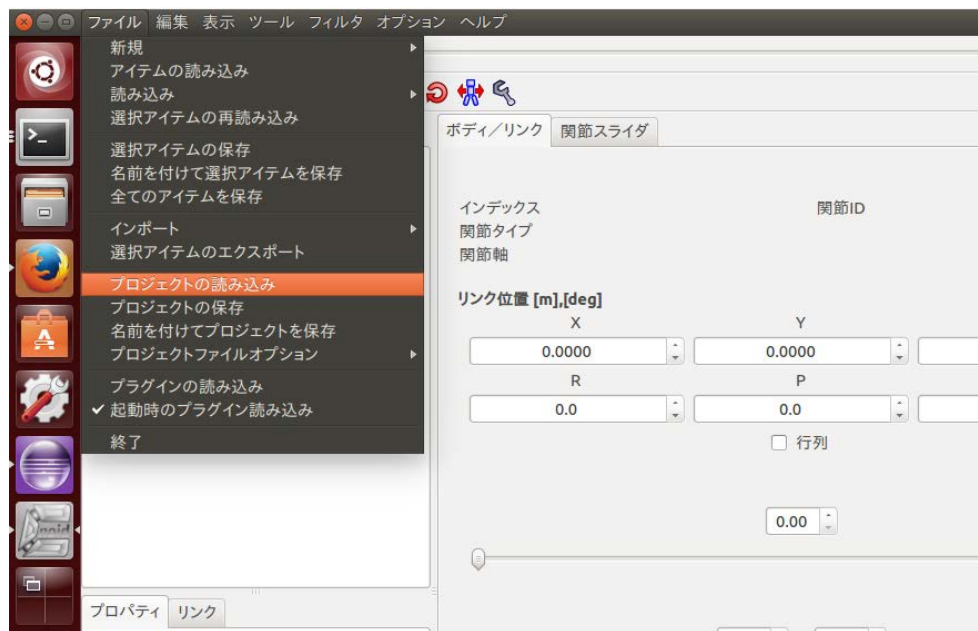


図 5 プロジェクトの読み込み

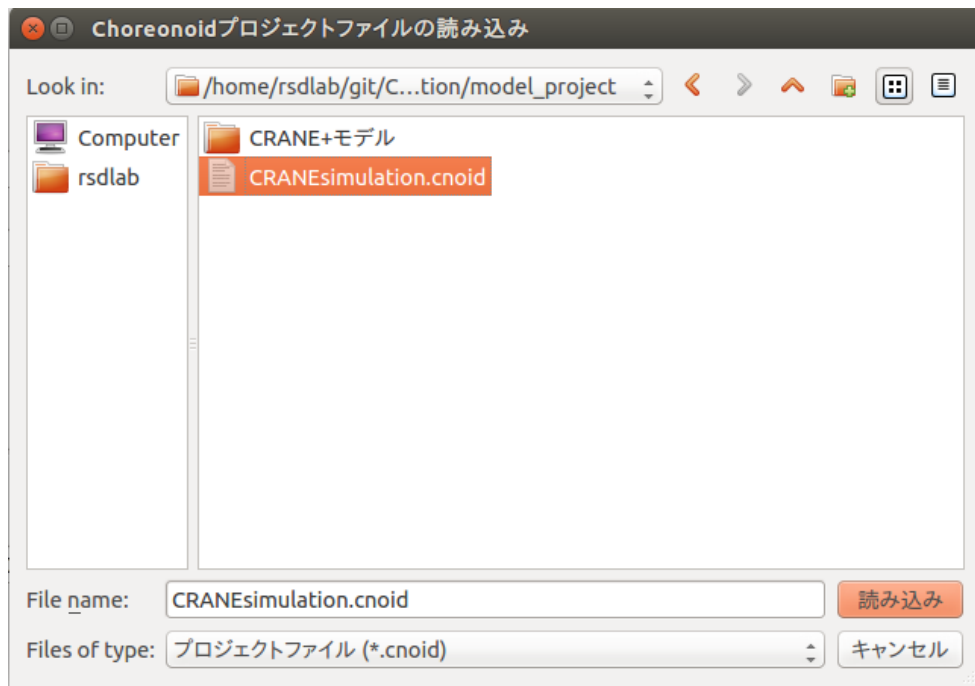


図 6 プロジェクトの読み込み

アイテムツリーが以下の図 7 のように表示されているのを確認する.

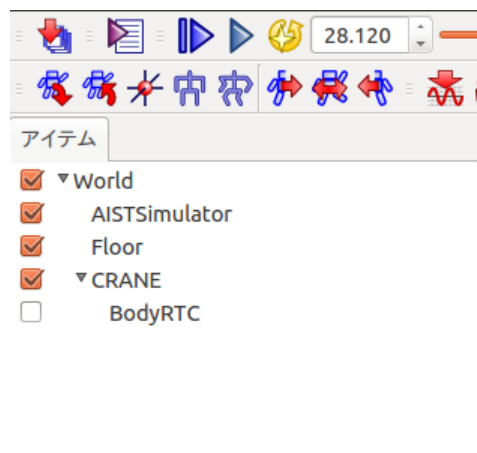


図 7 アイテムツリー

Eclipse のネームサーバーに CRANESimulation が実行されていることを確認する. (図 8)

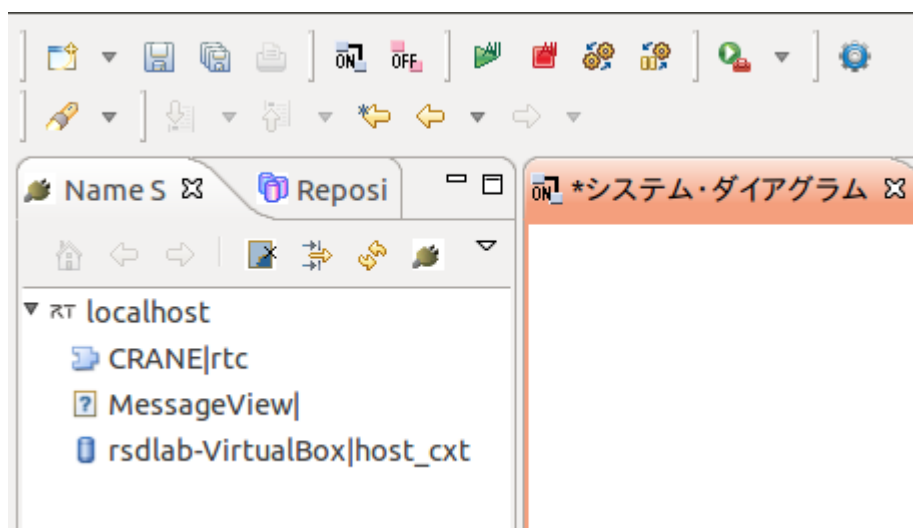


図 8 ネームサーバーの CRANE|rtc の確認

3.3 シミュレータ操作コンポーネントの実行

コレオノイドモデルを制御する ChoreonoidCRANEControllerRTC (図 9) と、サンプルコントローラである ManipulatorControlSample(図 10)を実行する。二つのコンポーネントの仕様を以下に示す。

①ChoreonoidCRANEControllerRTC

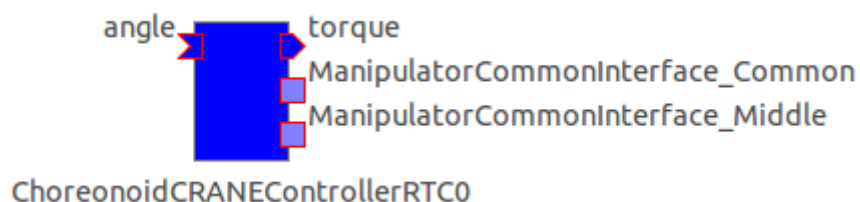


図 9 ChoreonoidCRANEControllerRTC

ChoreonoidCRANEControllerRTC のコンフィグレーションパラメータは以下の通りである。

名称	データ型	デフォルト値	説明
Port_name	string	/dev/ttyUSB0	シリアル通信ポートの指定
Mode	short	0	0：シミュレータのみ 1：実機のみ 2：シミュレータと実機

②ManipulatorControlSample

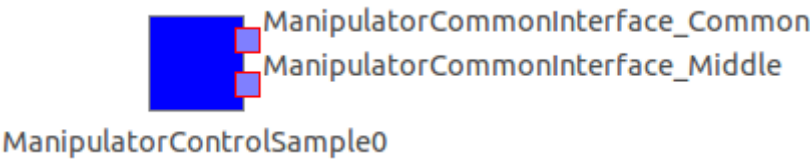


図 10 ManipulatorControlSample

ManipulatorControlSample のコンフィグレーションパラメータは以下の通りである.

名称	データ型	デフォルト値	説明
Speed	int	20[%]	CRANE+の関節速度

二つのコンポーネントを実行するために, `simulationRTCsEXE.sh` を実行する. 端末上で以下のコマンドを実行する.

```
~$ cd CRANE-simulation/script/  
/CRANE-simulation/script ~$ sh simulationRTCsEXE.sh
```

新しい端末が立ち上がり, コンポーネントが実行される.

3.4 コンポーネント接続

Choreonoid のプロジェクトファイルの読み込みと RTC の実行できたら, コンポーネントを接続してシミュレーションの実行をする. `simulationRTCsConnect.sh` を実行する.

```
/CRANE-simulation/script ~$ sh simulationRTCsConnect.sh
```

Eclipse の RTSystemEditor でシステムダイアグラムを ON にし, コンポーネントをシステムダイアグラムに表示し, 以下の図 11 のようにポートが接続されていることを確認する.

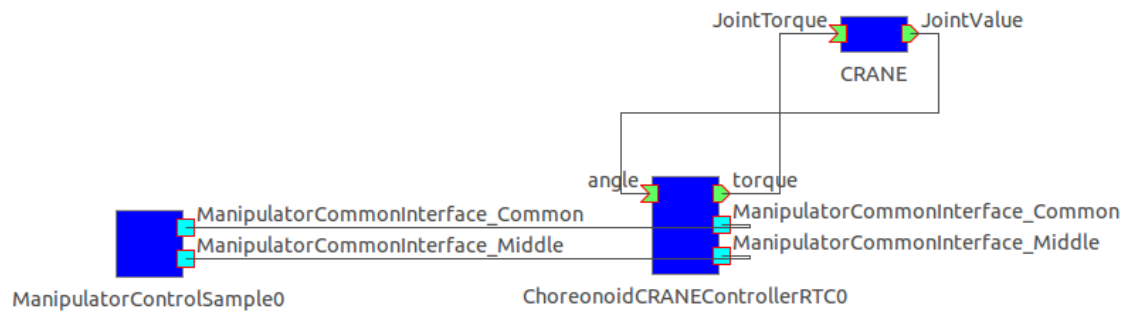


図 11 コンポーネント接続図

3.5 シミュレーションの実行

Choreonoid のシミュレーションの実行をクリックする。(図 12)

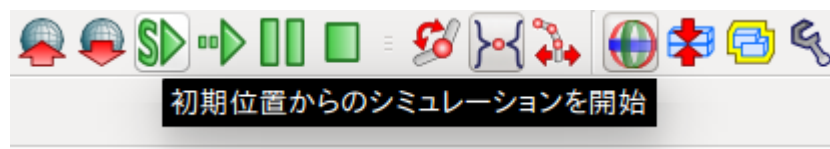


図 12 シミュレーションの開始

シミュレーションが開始されるとコンポーネントが自動的に Activate される。

Activate に成功すると, ManipulatorControlSample を実行している端末で以下の図 13 のように表示される。

```

ManipulatorControlSample
コマンドを選択してください
1 : サーボ ON
2 : サーボ OFF
3 : アーム先端の移動
4 : アーム関節の回転
5 : アーム先端座標の計算
6 : グリッパ開閉
7 : グリッパ閉
8 : グリッパ開
9 : 終了
>>

```

図 13 ManipulatorControlSample 実行画面

このコマンドはそれぞれ以下の機能を持つ。

1: サーボ ON

CRANE+の全モータのトルクを ON にする。シミュレータ上のアームはシミュレーション開始した時に自動的にサーボ ON される。

2: サーボ OFF

CRANE+の全モータのトルクを OFF にする.

3: アーム先端の移動

絶対座標で入力した座標にアームの先端を移動させる. 移動できる座標は以下の通りである. 図 14, 図 15 可動範囲を示す.

円の中心: $(x, y, z) = (14.5, 0, 45.4)$ [mm]

外側リミット半径: 314[mm]

内側リミット半径: 200[mm]

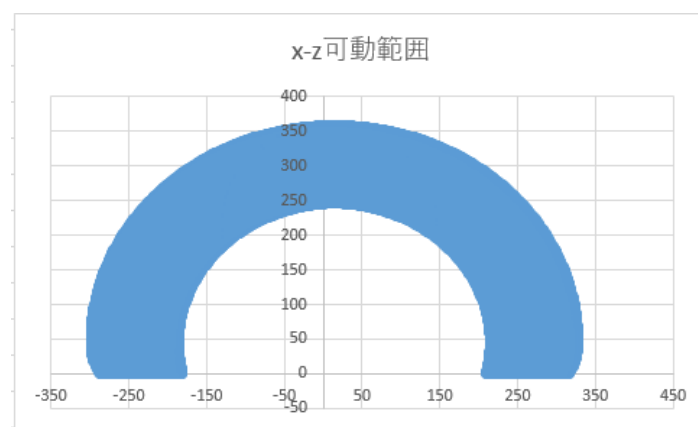


図 14 x-z 平面可動範囲

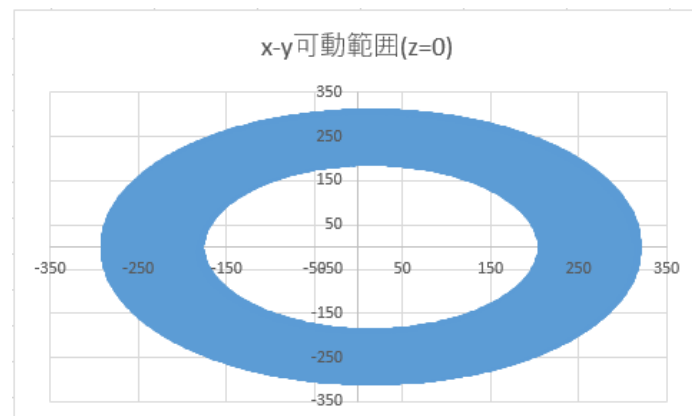


図 15 x-y 平面可動範囲

入力できる値

x 座標: $-300 \sim 329$ [mm]

y 座標: $-314 \sim 314$ [mm]

z 座標: $0 \sim 359$ [mm]

※上記範囲でも原点から先端までの距離が $200 \sim 314$ [mm] の範囲でなければ移動できません.

4：アーム関節の回転

グリッパ以外の関節を角度[°]を指定して回転させる。

5：アーム先端座標の計算

アームの各関節からアームの先端座標を計算する。

6：グリッパ開閉

指定した開度[%]でグリッパを開閉させる。

7：グリッパ閉

グリッパを閉じる。

8：グリッパ開

グリッパを開く。

9：終了

コンポーネントを終了する。このコマンドを実行した後、コレオノイド上でシミュレーションの停止ボタンを押すこと。（図 16）

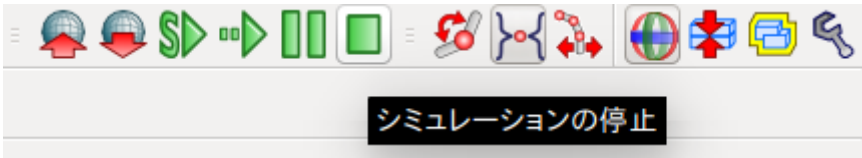


図 16 シミュレーションの停止

4. CRANE＋実機の制御

他のシステムと連携して CRANE＋を動かす場合など、シミュレータを使わずに CRANE＋を動かす CraneplusRTC（図 17）を使うこともできる。CraneplusRTC の仕様を以下に示す。

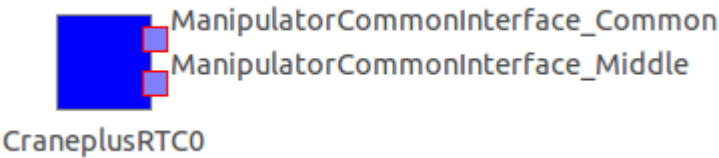


図 17 CraneplusRTCComp

CraneplusRTC のコンフィグレーションパラメータは以下の通りである。

名称	データ型	デフォルト値	説明
Port_name	string	/dev/ttyUSB0	シリアル通信ポートの指定

2.1 コンポーネントのビルド, 2.2 dialout にユーザーの追加, 3.1 eclipse の起動を行ってから以下の操作を行う。本マニュアルではサンプルコントローラである ManipulatorControlSample を使用する。ManipulatorControlSample の詳細については 3.3 シミュレータ操作コンポーネントの実行及び 3.5 シミュレーションの実行を参照すること。

4.1 CRANE+制御コンポーネントの実行

CRANE+ 制御 RTC である CraneplusRTC とサンプルコントローラである ManipulatorControlSample を実行するために、CRANE+RTCsEXE.sh を実行する。端末で以下のコマンドを実行する。

```
~$ cd CRANE-simulation/script/  
/CRANE-simulation/script~$ sh CRANE+RTCsEXE.sh
```

新しい端末が立ち上がり、コンポーネントが実行される。

4.2 コンポーネント接続

RTC が実行できたら、コンポーネントの接続をするために、CRANE+RTCsConnect.sh を実行する。

```
/CRANE-simulation/script ~$ sh CRANE+RTCsConnect.sh
```

Eclipse の RTSystemEditor でシステムダイアグラムを ON にし、コンポーネントをシステムダイアグラムに表示し、以下の図 18 のようにポートが接続されていることを確認する。

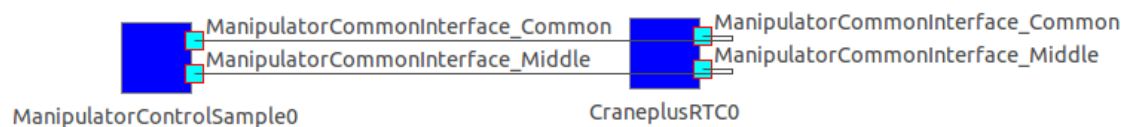


図 18 コンポーネント接続図

4.3 コンポーネントの Activate

ここで、Activate する前に以下の 4 つを確認する。

- 1) CRANE+の電源が kobuki と繋がっており kobuki の電源が入っている。(図 19)
- 2) CRANE+とパソコンが接続されている。(図 20)



図 19 kobuki との電源接続



図 20 CRANE+と PC の接続

- 3) VM などの仮想環境で実行する場合、仮想 OS 側でデバイスを認識する。デバイスを認識するまで通常数秒かかる。(図 21)

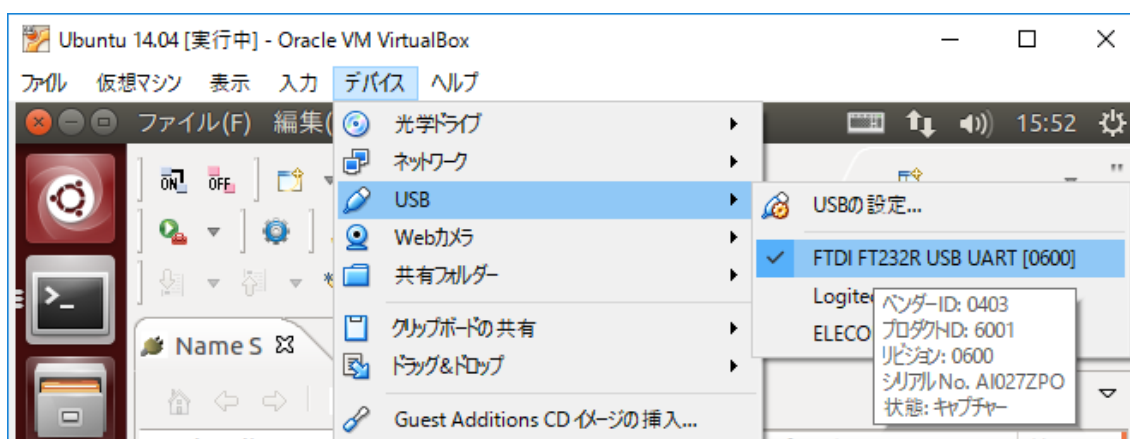


図 21 デバイスの認識(VM の場合)

- 4) シリアルポートの接続を確認する。USB に接続しているデバイスを CRANE+のみにし、以下のコマンドを打ちこんで表示されるポートに CRANE+が接続されている。このとき、ttyUSB0 以外に接続されている場合 CraneplusRTC のコンフィグレーションパラメータを変更する。

```
~$ ls /dev/ttyUSB*
```

以下の図 22 のように表示されればシリアルポートが接続されている。

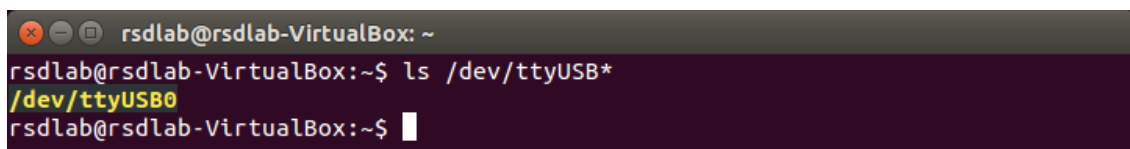


図 22 シリアルポートの確認

- ④以上のことが確認できたら Activate で実行する (図 23)。このとき、

ManipulatorControlSample を先に Activate するとエラーとなるので、必ず同時に Activate するか、CranepiusRTC を先に Activate してから ManipulatorControlSample を Activate する。CRANE+RTCsActivate.sh を実行するか、図 23 のように eclipse で All Activate する。

```
/CRANE-simulation/script ~$ sh CRANE+RTCsActivate.sh
```

または

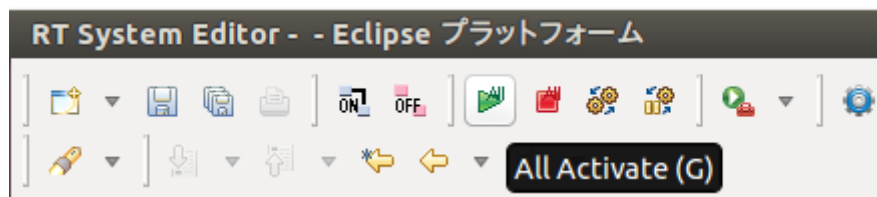


図 23 Activate

Deactivate する際は、ManipulatorControlSample で「9：終了」を選択した後に CRANE+RTCsDeactivate.sh を実行するか、図 24 のように eclipse で All Deactivate する。

```
/CRANE-simulation/script ~$ sh CRANE+RTCsDeactivate.sh
```

または

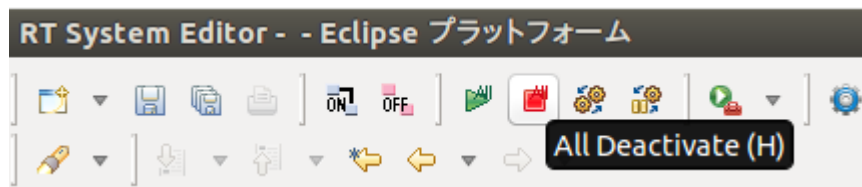


図 24 Deactivate

5. シェルスクリプトが実行できないとき

シェルスクリプトを使ったコンポーネントのビルド，実行，接続ができない場合以下を参考に操作する．

5.1 シミュレータを用いた CRANE+ 操作

3.1 eclipse の起動と， 3.2 Choreonoid の実行を行ってから以下の操作を行う．

5.1.1 ChoreonoidCRANEControllerRTC の実行

ChoreonoidCRANEControllerRTC を実行する．端末上で以下のコマンドを実行する．

```
~$ cd ChoreonoidCRANEControllerRTC
/ChoreonoidCRANEControllerRTC~$ mkdir build
/ChoreonoidCRANEControllerRTC~$ cd build
/ChoreonoidCRANEControllerRTC/build~$ cmake ..
/ChoreonoidCRANEControllerRTC/build~$ make
/ChoreonoidCRANEControllerRTC/build~$ cd src
/ChoreonoidCRANEControllerRTC/build/src~$ ./ChoreonoidCRANEControllerRTC
```

5.1.2 ManipulatorControlSampleRTC の実行

ChoreonoidCRANEControllerRTC の動作確認のためのサンプルコンポーネント．
ChoreonoidCRANEControllerRTC の動作検証を行う場合は，こちらのコンポーネントを利用する．本コンポーネントの詳細については 3.3 シミュレータ操作コンポーネントの実行及び 3.5 シミュレーションの実行を参照すること．端末上で以下のコマンドを実行する．

```
~$ cd ManipulatorControlSample
/ManipulatorControlSample~$ mkdir build
/ManipulatorControlSample~$ cd build
/ManipulatorControlSample/build~$ cmake ..
/ManipulatorControlSample/build~$ make
/ManipulatorControlSample/build~$ cd src
/ManipulatorControlSample/build/src~$ ./ManipulatorControlSample
```

5.2.3 コンポーネント接続

Eclipse の RTSystemEditor でシステムダイアグラムを ON にし、コンポーネントをシステムダイアグラムに表示し、以下の図 25 のようにポートを接続する。

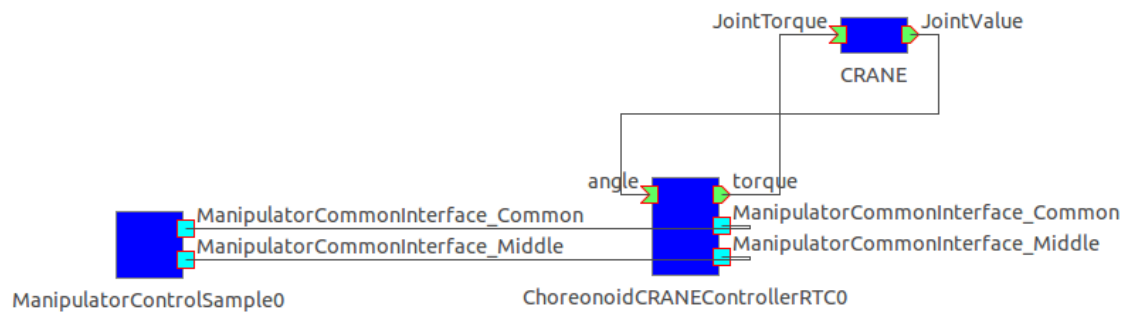


図 25 コンポーネント接続図

5.1.4 シミュレーションの実行

Choreonoid のシミュレーションの実行をクリックする。(図 26)

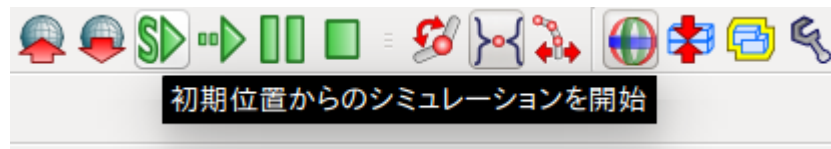


図 26 シミュレーションの開始

シミュレーションが開始されるとコンポーネントが自動的に Activate される。

Activate に成功すると、ManipulatorControlSample を実行している端末で以下の図 27 のように表示される。各コマンドの詳細は 3.5 章を参照すること。

```
ManipulatorControlSample
コマンドを選択してください
1 : サーボ ON
2 : サーボ OFF
3 : アーム先端の移動
4 : アーム関節の回転
5 : アーム先端座標の計算
6 : グリッパ開閉
7 : グリッパ閉
8 : グリッパ開
9 : 終了
>>
```

図 27 ManipulatorControlSample 実行画面

5.2 CRANE+実機の制御

2.2 dialout にユーザーの追加, 3.1 eclipse の起動を行ってから以下の操作を行う.

5.2.1 Craneplus RTC の実行

CraneplusRTC を実行する. 端末上で以下のコマンドを実行する.

```
~$ cd CraneplusRTC
/CraneplusRTC~$ mkdir build
/CraneplusRTC~$ cd build
/CraneplusRTC/build~$ cmake ..
/CraneplusRTC/build~$ make
/CraneplusRTC/build~$ cd src/
/CraneplusRTC/build/src~$ ./CraneplusRTCComp
```

5.2.2 ManipulatorControlSampleRTC の実行

CraneplusRTC の動作確認のためのサンプルコンポーネント. Craneplus RTC の動作検証を行う場合は, こちらのコンポーネントを利用する. 本コンポーネントの詳細については 3.3 シミュレータ操作コンポーネントの実行及び 3.5 シミュレーションの実行を参照すること. 端末上で以下のコマンドを実行する. 5.1.2 をすでに行っている場合は, 一番下の行の実行のみでよい.

```
~$ cd ManipulatorControlSample
/ManipulatorControlSample~$ mkdir build
/ManipulatorControlSample~$ cd build
/ManipulatorControlSample/build~$ cmake ..
/ManipulatorControlSample/build~$ make
/ManipulatorControlSample/build~$ cd src
/ManipulatorControlSample/build/src~$ ./ManipulatorControlSample
```

5.2.3 コンポーネント接続

Eclipse の RTSystemEditor でシステムダイアグラムを ON にし, コンポーネントをシステムダイアグラムに表示し, 以下の図 28 のようにポートを接続する.

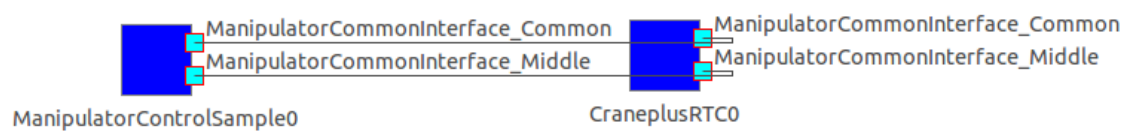


図 28 コンポーネント接続図

コンポーネントの `Activate` については 4.3 コンポーネントの `Activate` を参照すること.