

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 11 月 10 日

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-1.1.0
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
RT ミドルウェア	OpenRTM-aist-1.1.0
Choreonoid	Choreonoid-1.5.0

2. シミュレータを用いた CRANE+ 操作方法

準備中

2016/11/18 までに完成予定

図 1~14 まで使用します

3. 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+ を動かすコンポーネントを使うこともできる. CRANE+ 制御コンポーネントを図 14 に示す.

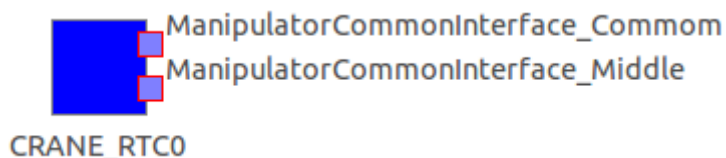


図 14 RT_CRANE_RTCComp

3.1 RT_CRANE_RTC の実行

2.1 eclipse の実行 を行ってから以下の手順を行い, CRANE_RTC を実行する.

本コンポーネントはシリアル通信を使用するため sudo で実行する必要があるので注意すること.

```
~$ cd RT_CRANE_RTC
/RT_CRANE_RTC~$ mkdir build
/RT_CRANE_RTC~$ cd build
/RT_CRANE_RTC/build~$ cmake ..
/RT_CRANE_RTC/build~$ make
/RT_CRANE_RTC/build~$ cd src/
/RT_CRANE_RTC/build/src~$ sudo ./CRANE_RTCComp
```

パスワードを要求されるためユーザー指定のパスワードを入力する.

3.2 ManipulatorControlRTC の実行

RT_CRANE_RTC の動作確認のためのサンプルコンポーネント. RT_CRANE_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.

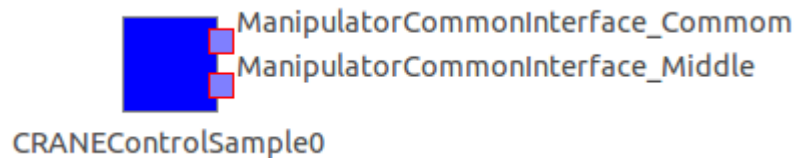


図 15 ManipulatorControlSample

3.2.1 ManipulatorControlSample の実行

①ManipulatorControlSample 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
```

②eclipse で実行されていることを確認し、システムダイアグラムに表示させて以下の図 16 のようにポートを接続する.

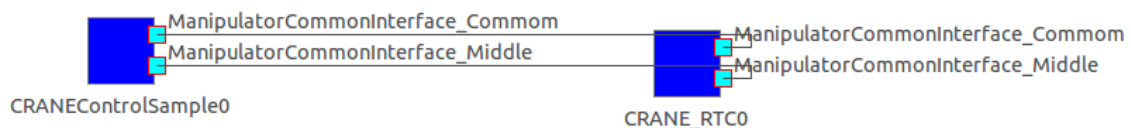


図 16 コンポーネント接続図

③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 17)
- 2) CRANE+とパソコンが接続されている. (図 18)



図 17 kobuki との電源接続



図 18 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 19)

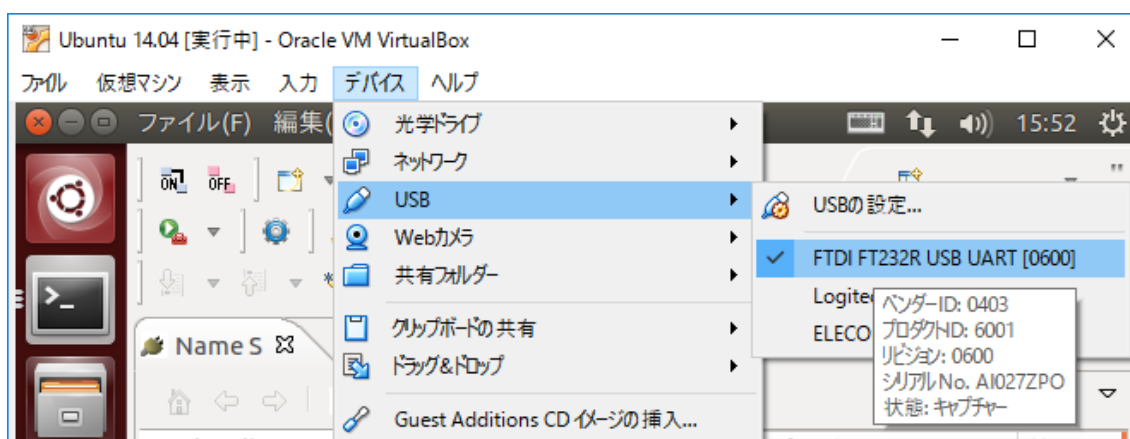


図 19 デバイスの認識

- 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 `RT_CRANE_RTC` 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 20 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。

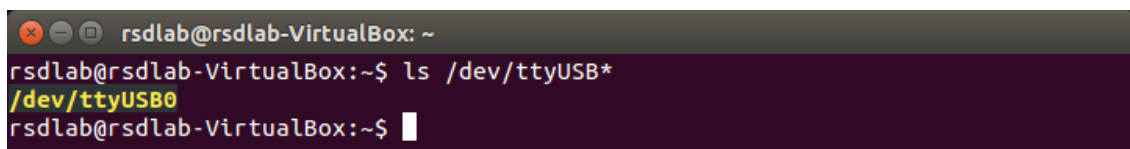


図 20 シリアルポートの確認

- ④以上のことが確認できたら `Activate` で実行する。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、RT_CRANE_RTC を先に Activate してから ManipulatorControlSample を Activate する。Activate に成功すると、CRANEControlSample を実行されている端末で以下の図 21 のように表示される。

```
rsdlab@rsdlab-VirtualBox:~/workspace/CRANEControlSample/build/src$ ./CRANEControlSampleComp
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 : グリッパ開閉
6 : グリッパ閉
7 : グリッパ開
8 : 終了
>>
```

図 21 CRANEControlSample 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。アームを動かす場合はまずこのコマンドを実行しなければ動かない。

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。

x: $-314.5 \sim 314.5$ [mm]

y: $-314.5 \sim 314.5$ [mm]

z: $0 \sim 314.5$ [mm]

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: グリッパ開閉

指定した開閉度[%]でグリッパを開閉させる。

6: グリッパ閉

グリッパを閉じる。

7: グリッパ開

グリッパを開く。

8: 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, `eclipse` でシステム全体を `Deactivate` しておくこと.