

操作手順書

リファレンスハードウェア移動制御モジュール

V e r . 1.10

2011 年 11 月 30 日

株式会社 東芝

本モジュールのライセンスは Eclipse Public License(EPL)に従います.

改版履歴

Ver.	改版日	改版内容
1.00	2011/11/16	初版作成
1.10	2011/11/30	4.3.4 台車移動の実行 を追加

目次

1	はじめに.....	1
1.1.	概要.....	1
1.2.	関連文書.....	2
1.3.	動作環境.....	2
1.3.1.	関連ハードウェア	2
1.3.2.	関連ソフトウェア	2
1.3.3.	ディレクトリ構成	3
2	環境構築.....	4
2.1.	環境変数の設定.....	4
2.2.	利用ソフトウェアのインストール	5
2.2.1.	C/C++ Development Tools のインストール	5
2.2.2.	RT System Editor のインストール	7
2.2.3.	rtshell のインストール	9
3	ビルド手順	11
3.1.	ソースの展開	11
3.2.	ビルド	12
3.2.1.	ビルド準備	12
3.2.2.	コマンドラインでのビルド	12
3.2.3.	Eclipse でのビルド	13
4	実行手順.....	16
4.1.	RH3 号機との接続	16
4.2.	ネームサーバの起動.....	17
4.3.	RTC の実行.....	17
4.3.1.	実行準備	17
4.3.2.	手動実行	18
4.3.3.	自動実行	31
4.3.4.	台車移動の実行.....	31
4.4.	RTC のディアクティベート.....	32
4.5.	RTC の終了.....	35
4.6.	RH3 号機を人が移動させた場合の対応	36
5	特記事項.....	37

図表目次

図 1-1 リファレンスハードウェア 3 号機	1
図 1-2 ディレクトリ構成	3
図 2-1 プラグイン追加画面	5
図 2-2 ネットワーク設定画面	6
図 2-3 Eclipse Configuration	8
図 3-1 RH 移動制御モジュールのディレクトリ構成	11
図 3-2 パースペクティブ選択画面	13
図 3-3 プロジェクトのインポート画面	14
図 3-4 Eclipse 画面	15
図 4-1 ケーブルの接続	16
図 4-2 電源供給先を指定するスイッチ	16
図 4-3 Debug アイコン	19
図 4-4 パースペクティブ変更確認画面	20
図 4-5 Debug 画面 Resume アイコン	21
図 4-6 Debug 画面 Suspend アイコン/Terminate アイコン	22
図 4-7 Debug 画面 Running 状態	22
図 4-8 Debug 画面 Suspend 状態	23
図 4-9 Debug 画面 Terminated 状態	23
図 4-10 RT System Editor	24
図 4-11 ネームサーバ上にある RTC 一覧	25
図 4-12 Open New System Editor アイコン	26
図 4-13 System Diagram 画面	27
図 4-14 ポートの接続方法	28
図 4-15 ポート接続後の RTC	28
図 4-16 All Activate の選択	29
図 4-17 Activate の選択	30
図 4-18 目標位置姿勢または経路の選択	31
図 4-19 目標位置姿勢の入力	31
図 4-20 移動終了時の表示	32
図 4-21 経路の指定	32
図 4-22 All Deactivate の選択	33
図 4-23 Deactivate の選択	34
図 4-24 Exit の選択	35
図 4-25 人手で RH3 号機を移動する際の流れ	36
表 1-1 関連文書	2

表 1-2 RH3 号機内蔵 PC の環境	2
表 1-3 関連ソフトウェア	2
表 4-1 接続の構成	28

1 はじめに

1.1. 概要

本書では、リファレンスハードウェア（以下、RH）3号機の位置制御モジュールを利用するための手順について記述している。

また、本書は RT ミドルウェア及び RT コンポーネント（以下、RTC），Linux についての基礎知識を有した者を対象者としている。

RT ミドルウェア及び RTC についての詳細は下記サイトを参照のこと。

<http://www.openrtm.org/openrtm/>

RH3 号機の外観は図 1-1 のようになっており，RH 移動制御モジュールは RTC 群により構成されている。



図 1-1 リファレンスハードウェア 3 号機

1.2. 関連文書

本書の関連文書を表 1-1 に示す.

表 1-1 関連文書

No.	文書名	備考
1	機能仕様書 リファレンスハードウェア 移動制御モジュール	本モジュールの機能仕様書

1.3. 動作環境

1.3.1. 関連ハードウェア

RH 移動制御モジュールで利用するハードウェアは RH3 号機を対象としている. また, RH3 号機は PC を内蔵しており, RH 移動制御モジュールは内蔵 PC 上で実行する. 内蔵 PC の環境を表 1-2 に示す.

表 1-2 RH3 号機内蔵 PC の環境

動作 OS	Ubuntu 10.04 LTS
開発言語	C++
コンパイラ	GCC ver4.4.3
RT ミドルウェア	OpenRTM-aist-1.0.0-RELEASE
依存パッケージ	なし

1.3.2. 関連ソフトウェア

RH 移動制御モジュールの実行で利用される関連ソフトウェアを表 1-3 に示す.

表 1-3 関連ソフトウェア

ソフトウェア	バージョン	備考
Eclipse	3.5(Galileo)	
RT System Editor	1.1.0-RC1	
C/C++ Development Tools	6.0.2	
rtshell	3.0.0	
rtctree	3.0.0	
rtspfile	2.0.0	

1.3.3.ディレクトリ構成

RH3 号機内蔵 PC には OpenRTM が既にインストールされている。RH3 号機内蔵 PC にインストールされた OpenRTM 関連のディレクトリ構成を図 1-2 に示す。



図 1-2 ディレクトリ構成

2 環境構築

2.1. 環境変数の設定

OpenRTM を利用するためには、環境変数の設定が必要となる。下記手順にて、環境変数の設定を行う。ファイルの編集には vi エディタを使用する。

- ライブラリの設定

OpenRTM のライブラリを利用するために、ライブラリへのパスの設定を行う。

(1) 下記コマンドを実行し、openrtm.conf を作成する。

```
$ sudo vi /etc/ld.so.conf.d/openrtm.conf
```

(2) 下記の一文を openrtm.conf の末尾に追記し、保存する。

```
/opt/grx/lib
```

(3) 下記コマンドで設定変更を反映する。

```
$ sudo ldconfig -v
```

- PATH の設定

OpenRTM のコマンド群を利用するために、コマンド群へのパスの設定を行う。

(1) 下記コマンドを実行し、profile を開く。

```
$ sudo vi /etc/profile
```

(2) 下記の一文を profile の末尾に追記し、保存する。

```
PATH=${PATH}:/opt/grx/bin
```

(3) コマンド実行後、設定を全体に適用させるために再起動を行う。

2.2. 利用ソフトウェアのインストール

2.2.1. C/C++ Development Tools のインストール

C/C++ Development Tools（以下、CDT）は Eclipse で C/C++の開発を行うための環境であり、Eclipse のプラグインとして提供されている。RH 移動制御モジュールを実行する上でインストールは必須ではなく、以下の場合にのみ必要となる。

- Eclipse から RH 移動制御モジュールをビルドする場合
- RH 移動制御モジュールのソースの変更を効率的に行いたい場合

RH3 号機内蔵 PC には Eclipse 自体は既にインストールされている。そのため、CDT をインストールするためには、Eclipse に CDT のプラグインを追加する必要がある。以下にその手順を示す。

- (1) Eclipse を起動する。
- (2) Eclipse 上部メニューの”Help”から”Install New Software”を選択する。
- (3) “Work with”（図 2-1 赤枠部分）より、”Galileo - http://...”を選択する。
- (4) 図 2-1 青枠部分に表示されたパッケージ一覧の”Programming Languages”の項目から”Eclipse C/C++ Development Tools”を選択する。

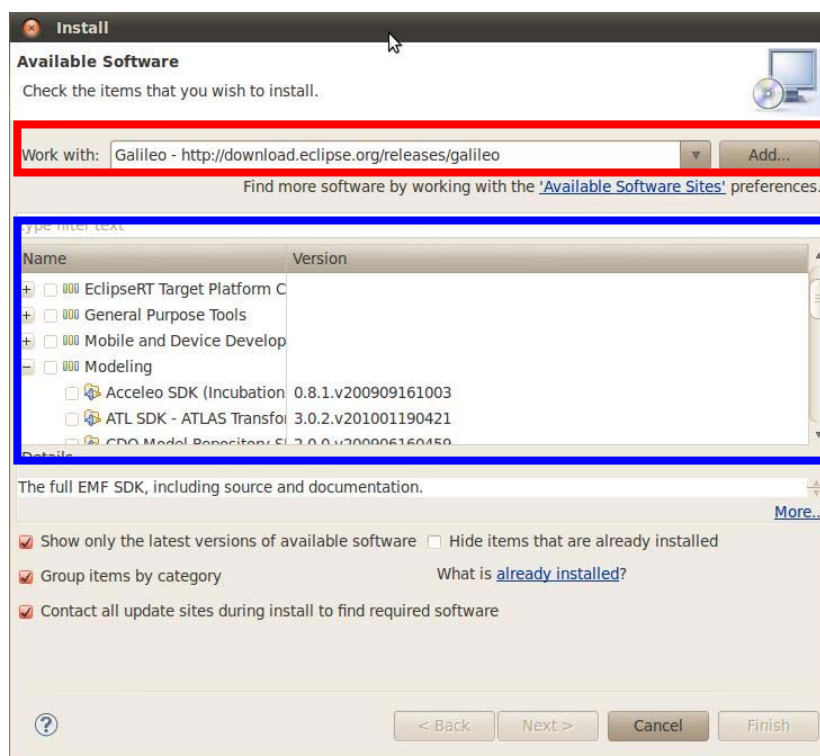


図 2-1 プラグイン追加画面

- (5) “Next”ボタンを押していき、規約に同意した上で”Finish”ボタンを押下するとインストールが開始される。
- (6) インストール完了後、Eclipse の再起動を要求されるので再起動を行う。

なお、インターネットへの接続をプロキシ経由で行う環境の場合、Eclipse にプロキシの設定をする必要がある。その場合は、Eclipse 上部メニューの”Window”から”Preference”を選択、設定画面の”General”内の”Network Connections”を選択するとネットワークの設定画面が表示される。

図 2-2 はネットワークの設定画面であり、図 2-2 赤枠部分の Active Provider を”Manual”に変更することでプロキシの設定が可能となる。プロキシの設定は図 2-2 青枠部分で行い、HTTP/HTTPS のみ対応のプロキシの場合は SOCK の Host ならびに Port は空欄とする。図 2-2 で表記した Host ならびに Port は例であり、環境により適宜変更すること。

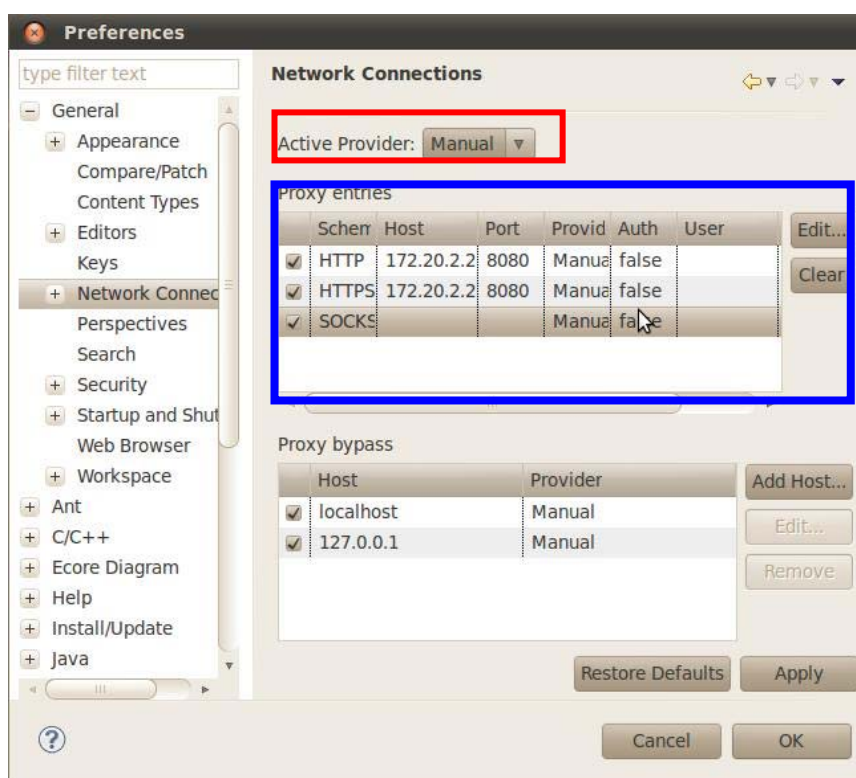


図 2-2 ネットワーク設定画面

2.2.2.RT System Editor のインストール

RT System Editor は Eclipse 上動作し、RTC をグラフィカル操作する機能を持った開発ツールである。Eclipse のプラグインとして提供されている。RTC 同士の接続や、RTC の状態などを変更するために利用する。以下にそのインストールの手順を示す。

- (1) Eclipse を起動する。
- (2) Eclipse 上部メニューの”Help”から”Install New Software”を選択する。
- (3) “Work with”より, ”Galileo - <http://...>”を選択する。
- (4) パッケージ一覧の中から、RT System Editor で利用するパッケージを選択する。必要となるパッケージは以下となる。いずれも、パッケージ一覧の”Modeling”の項目から選択する。
 - Graphical Modeling Framework SDK
 - EMF - Eclipse Modeling Framework SDK
 - XSD - XML Schema Definition SDK
- (5) “Next”ボタンを押していき、規約に同意した上で”Finish”ボタンを押下するとインストールが開始される。
- (6) インストール完了後、Eclipse の再起動を要求されるので再起動を行う。
- (7) RT System Editor のバイナリファイルを下記サイトからダウンロードする。

<http://www.openrtm.org/openrtm/ja/node/1736>

サイトには、英語版 (rtmtools110-rc1_en.zip) と日本語版 (rtmtools110-rc1_ja.zip) があるが、どちらをダウンロードしても構わない。本書では英語版を選んだものとして説明を行う。

- (8) ダウンロードしたディレクトリに移動し、下記コマンドにてバイナリファイルの解凍を行う。

```
$ unzip rtmtools110-rc1_en.zip
```

(9) Eclipse を停止し、解凍したディレクトリ内に入っている全ての jar ファイルを、Eclipse のインストールディレクトリ内にある plugins ディレクトリにコピーする。

なお、Eclipse のインストールディレクトリは、下記の場所に記してある。

- Eclipse 上部メニューの”Help”から”About Eclipse”を選択し、表示された画面の”Installation Details”ボタンを押下し、”Configuration”タブより”eclipse.home.location=file:”の値（図 2-3 参照）

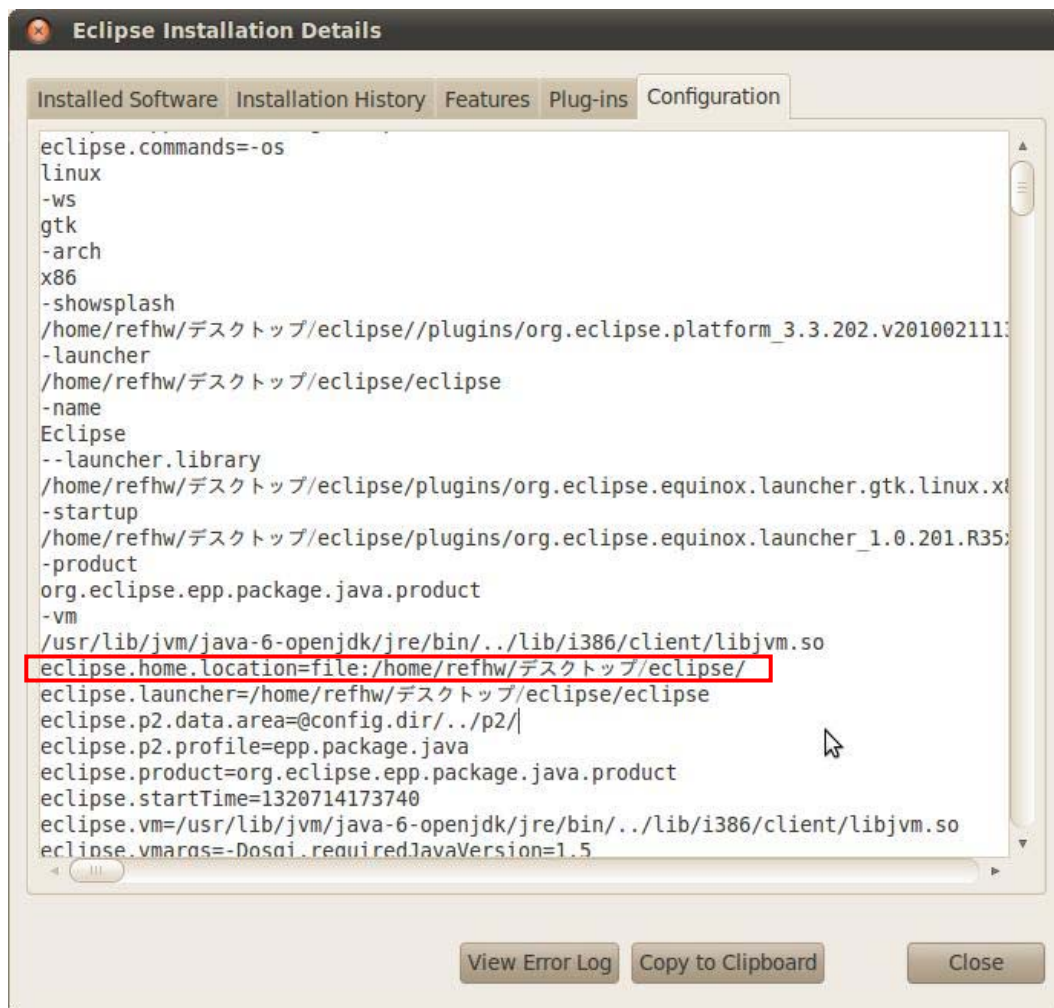


図 2-3 Eclipse Configuration

(10) Eclipse のディレクトリにて下記コマンドを実行し、キャッシュを削除して Eclipse の起動を行う。

```
$ ./eclipse -clean
```

2.2.3.rtshell のインストール

rtshell はネームサーバ上に登録された RTC をコマンドライン上から管理することができるツールである。GUI が利用できない環境において、RT System Editor の代わりに利用することができる。本手順では、RTC の自動実行（RTC の起動及び接続、アクティベート）をするために利用する。起動、接続、アクティベートを手動で行う際にはインストールの必要はない。

下記手順では、rtshell を利用するために必要な rtctree と rtsprofile のインストール、その他必要なモジュールのインストール、各ソフトウェアのダウンロードも行っている。ダウンロード場所は任意ではあるが、本書では下記のディレクトリにダウンロードしたものとして説明を行う。refhw はログインしているユーザ名であり、環境により適宜読み替えること。

```
/home/refhw
```

以下に rtshell のインストールの手順を示す。

- (1) 下記コマンドにて rtshell を使用するために必要なモジュールのインストールを行う。

```
$ sudo apt-get install omniidl4-python  
$ sudo apt-get install python-omniorb  
$ sudo apt-get install python-yaml
```

- (2) 下記サイトから rtctree-3.0.0.tar.gz をダウンロードする。

<http://www.openrtm.org/openrtm/ja/content/rtctreertsprofile>

- (3) 下記コマンドにて、ダウンロードした rtctree-3.0.0.tar.gz を解凍し、インストールを行う。

```
$ cd /home/refhw  
$ tar -xvzf rtctree-3.0.0.tar.gz  
$ cd rtctree-3.0.0  
$ sudo python setup.py install
```

- (4) 下記サイトから rtsprofile-2.0.0.tar.gz をダウンロードする。

<http://www.openrtm.org/openrtm/ja/content/rtctreertsprofile>

- (5) 下記コマンドにて、ダウンロードした rtsprofile-2.0.0.tar.gz を解凍し、インストールを行う。

```
$ cd /home/refhw  
$ tar -xvzf rtsprofile-2.0.0.tar.gz  
$ cd rtsprofile-2.0.0  
$ sudo python setup.py install
```

- (6) 下記サイトから rtshell-3.0.0.tar.gz をダウンロードする。

<http://www.openrtm.org/openrtm/ja/content/rtshell>

(7) 下記コマンドにて，ダウンロードした `rtshell-3.0.0.tar.gz` を解凍し，インストールを行う．

```
$ cd /home/refhw
$ tar -xvzf rtshell-3.0.0.tar.gz
$ cd rtshell-3.0.0
$ sudo python setup.py install
("Generate documentation?"と聞かれるので,"y"を入力する)
```

なお，以上の手順によるインストールで `rtshell` の全てのコマンドが使用できるわけではない．`rtprint`, `rtinject` および `rtlog` のコマンドを使用する場合には別途 `OpenRTM-Python` のインストールが必要となる．後述する `RTC` の自動実行では該当コマンドを使用していないため，今回はインストールを省いている．

3 ビルド手順

3.1. ソースの展開

RH 移動制御モジュールのソースを展開する。

RH 移動制御モジュールのソースの配置場所は任意であるが、本書では下記ディレクトリに置いたものとして説明を行う。refhw はログインしているユーザ名であり、環境に応じて読み替えることとする。

```
|| /home/refhw/workspace
```

下記コマンドにて、ダウンロードしたソースの解凍をすることができる。

```
|| $ cd /home/refhw/workspace
|| $ tar -xvzf positionControlRH3.tar.gz
```

展開したソースのディレクトリ構成を図 3-1 に示す。各ディレクトリにそれぞれ対応する RTC のソースが格納されている。

positionControlRH3	
└ Navigation	Navigation のディレクトリ
└ OdometryRH3	Odometry のディレクトリ
└ OmniTo2WD	OmniTo2WD のディレクトリ
└ PathFollower	PathFollower のディレクトリ
└ PositionInput	PositionInput のディレクトリ
└ RefHard3Base	RefHardRh3 のディレクトリ
└ start.sh	RTC 自動実行用のシェルスクリプト

図 3-1 RH 移動制御モジュールのディレクトリ構成

3.2. ビルド

展開したソースのビルドを行う。ビルドにはコマンドラインから行う方法と **Eclipse** から行う方法があり、以下ではその両方の手順を示す。ビルド結果に差異は出ないため、利用しやすいほうでビルドを行うこと。

3.2.1. ビルド準備

RefHardRh3 のビルド及び実行時に用いるダイナミックリンクライブラリ `libsvm.so` には 64bitCPU 用と 32bitCPU 用が存在する。libsvm.so はソース展開時には 64bitCPU 用のライブラリ `libsvm64.so` と同等のものであるため、32bitCPU で実行する場合には下記の手順を実行し、32bitCPU 用のライブラリ `libsvm32.so` を `libsvm.so` にコピーする必要がある。

```
$ cd /home/refhw/workspace/positionControlRH3/RefHard3Base
$ cp libsvm32.so libsvm.so
```

なお、RH3 号機内蔵 PC は 64bitCPU であるため、上記手順は不要となる。

3.2.2. コマンドラインでのビルド

コマンドライン上で下記コマンドを実行することで、ソースのビルドを行うことができる。

- Navigation のビルド

```
$ cd /home/refhw/workspace/positionControlRH3/Navigation
$ make
```

- PathFollower のビルド

```
$ cd /home/refhw/workspace/positionControlRH3/PathFollower
$ make
```

- Odometry のビルド

```
$ cd /home/refhw/workspace/positionControlRH3/OdometryRH3
$ make
```

- OmniTo2WD のビルド

```
$ cd /home/refhw/workspace/positionControlRH3/OmniTo2WD
$ make
```

- PositionInput のビルド

```
$ cd /home/refhw/workspace/positionControlRH3/PositionInput
$ make
```

- RefHardRh3 のビルド

```
$ cd /home/refhw/workspace/positionControlRH3/RefHard3Base
$ make
```

3.2.3.Eclipse でのビルド

Eclipse からソースのビルドを行う手順について説明する. Eclipse からのビルドを行う場合には, CDT のインストールが必須となる. (2.2.1 参照)

以下に, Eclipse でのビルド手順について示す. 図 3-1 に示したディレクトリごとに CDT のプロジェクトが独立しているため, 下記手順を全てのディレクトリに対し行う必要がある.

- (1) Eclipse を起動する.
- (2) Eclipse 上部メニューの”Window”より”Open Perspective”の”Other”を選択する.
- (3) 表示されたパースペクティブ選択画面から”C/C++”を選択肢, CDT の画面を開く.

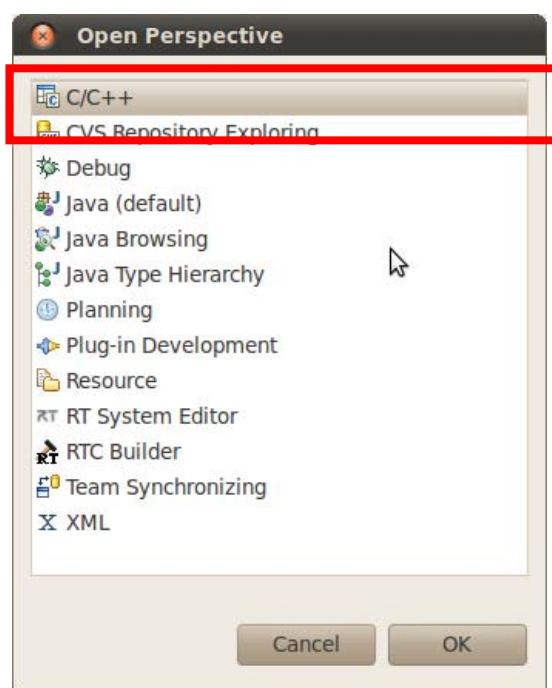


図 3-2 パースペクティブ選択画面

- (4) Eclipse 上部メニューの”File”より”import”を選択する。
- (5) 選択画面より，”General”内の”Exisiting Projects into Workspace”を選択し，”Next”ボタンを押下する。
- (6) 表示されたインポート画面の”Select root directory”（図 3-3 赤枠部分）から，ビルドを行いたい RTC のディレクトリを選択する。

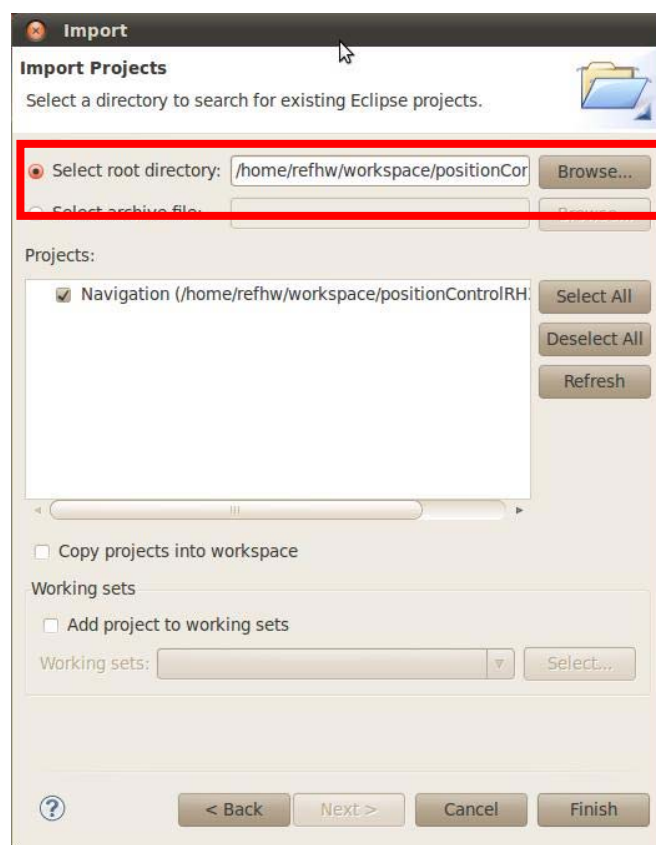


図 3-3 プロジェクトのインポート画面

- (7) インポート画面の”Finish”ボタンを押下すると，選択した RTC のプロジェクトのインポートが完了．ProjectExplorer にプロジェクトが追加される．

- (8) Project Explorer (図 3-4 青枠部分) から、ビルドを行うプロジェクトをクリックして選択する。Project Explorer が表示されていない場合、Eclipse 上部メニュー”Window”の”Show View”から”Project Explorer”を選ぶ。

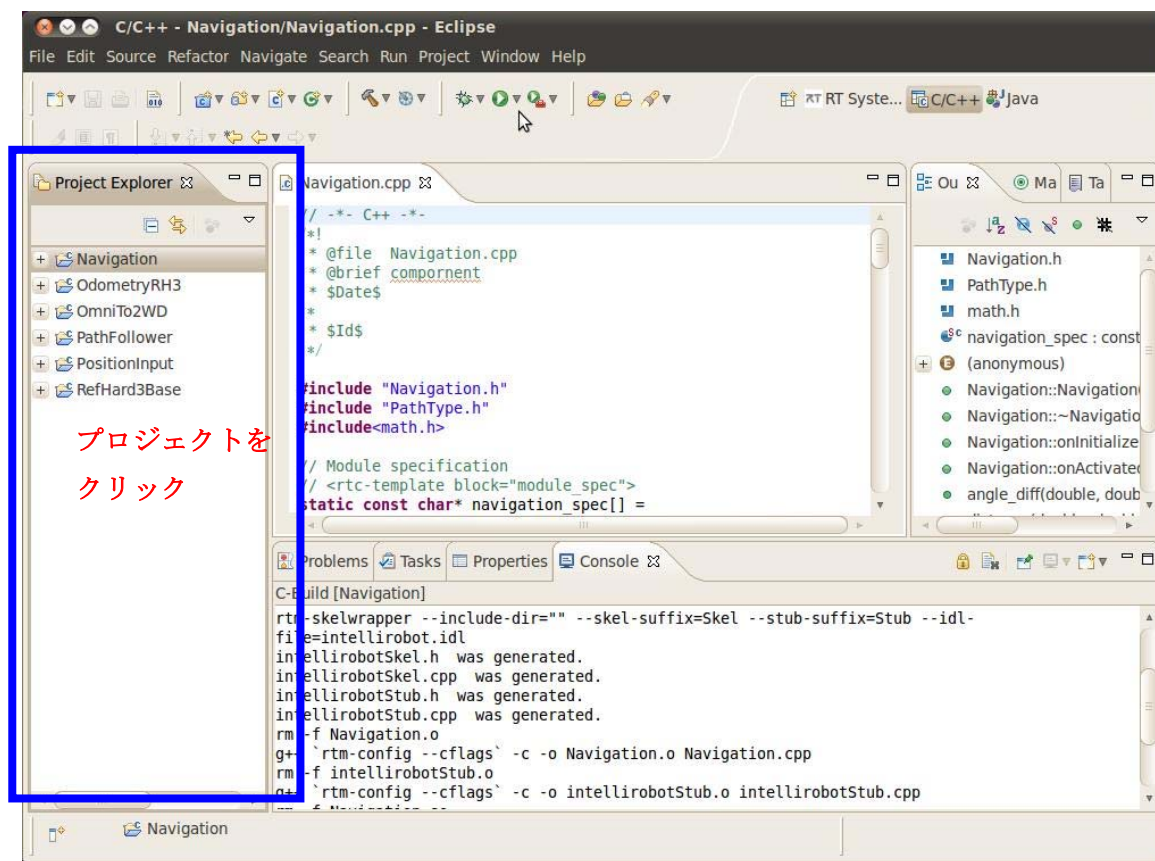


図 3-4 Eclipse 画面

- (9) Eclipse 上部メニューの”Project”から”Build Project”を選択することで、ソースのビルドを行うことができる。

なお、Eclipse 上部メニューの”Project”から”Build All”を選択することで現在 Project Explorer にあるプロジェクト全てをビルドすることが可能である。プロジェクトのディレクトリ内に既にあるオブジェクトファイルを消したい場合は、Eclipse 上部メニューの”Project”から”Clean”を選択することで行える。

4 実行手順

4.1.RH3 号機との接続

RH 移動制御モジュールを実行する際には RH3 号機内蔵 PC と RH3 号機を接続する必要がある。
以下に接続する手順を示す。

- (1) USB-シリアル変換ケーブル 2 本を RH3 号機内蔵 PC から外しておく。
- (2) USB-シリアル変換ケーブルと台車の制御用ケーブルを接続する。もう一本の USB-シリアル変換ケーブルとアームの制御用ケーブルを接続する（図 4-1 参照）。どちらの USB-シリアル変換ケーブルをどちらの制御用ケーブルと接続しても問題はない。
- (3) RH3 号機と、RH3 号機内蔵 PC を接続する。はじめに、台車の制御用ケーブルと接続された USB-シリアル変換ケーブルを、RH3 号機内蔵 PC に接続する。次に、アームの制御用ケーブルと接続された USB-シリアル変換ケーブルを、RH3 号機内蔵 PC に接続する。接続の順番さえ合っていれば、RH3 号機内蔵 PC のどのコネクタに接続しても問題ない。
- (4) RH3 号機の電源供給を外部電源からバッテリーに変更する場合は、PC を起動したまま図 4-2 に示す電源供給先指定スイッチを”バッテリー”側に変更する。



図 4-1 ケーブルの接続



図 4-2 電源供給先を指定するスイッチ

4.2. ネームサーバの起動

RH3 号機内蔵 PC は、スタートアップ時にネームサーバを起動するよう設定してあるため、手動によるネームサーバの起動は不要である。

スタートアップで起動するネームサーバの IP アドレスは”127.0.0.1”，ポート番号はデフォルト (2809) となっている。

別のポートでネームサーバを起動したい場合は、下記コマンドを実行することでネームサーバの起動をすることができる。ポート番号は省略するとデフォルト (2809) となる。

```
$ rtm-naming “ポート番号”
```

また、別 IP アドレスでネームサーバを起動したい場合は、下記コマンドで起動するネームサーバのアドレスを変更することができる。下記コマンド実行後、再度ネームサーバの起動を行うこと。

```
$ export OMNIORB_USEHOSTNAME=”IP アドレス”
```

4.3. RTC の実行

各 RTC を起動、接続し、アクティベートを行う。各工程を手動で行う方法と、シェルスクリプトを用いて自動で行う方法とがある。以下ではその両方の手順について示す。RTC の挙動に差異は出ないため、利用しやすい手順で実行すること。

4.3.1. 実行準備

RefHardRh3 は USB ポートにアクセスするため、実行する際に root 権限が必要となる。root 権限での実行の際にはパスワードが要求されるが、パスワード入力を省くための設定を行う。なお、パスワード入力の省略は自動実行を行う場合には必須であるが、手動実行の際には必須ではないため設定を省いてもよい。省いた場合にはパスワードが要求されるため、適宜入力を行うこと。

- (1) 任意ディレクトリにて下記コマンドを実行し、`sudoers` を起動する。

```
$ sudo visudo
```

- (2) `sudoers` の末尾に下記の一文を追記し、保存する。

```
refhw ALL =NOPASSWD: /home/refhw/workspace/positionControlRH3/RefHard3Base/R  
efHardRh3Comp
```

上記は、以下のような規則に基づいて記述を行っている。そのため、環境に応じて適宜書き換えること。なお、上記一文の”ALL”は、全てのホストのことを表している。RH3 号機を動作させる場合特定のホストを指定する必要がないため、ホストは”ALL”を指定すればよい。

```
“ユーザ名” “ホスト” =NOPASSWD: “コマンド (フルパス)”
```

4.3.2.手動実行

(1) RTC 起動

RTC を起動するためには、コマンドライン上から行う方法と、Eclipse 上から行う方法とがある。Eclipse 上からの RTC を起動する場合の手順は、デバックをする際を想定している。

まず、コマンドライン上から行う方法について示す。下記コマンドを実行することで、各 RTC が起動する。

- Navigation の起動

```
$ cd /home/refhw/workspace/positionControlRH3/Navigation
$ ./NavigationComp
```

- PathFollower の起動

```
$ cd /home/refhw/workspace/positionControlRH3/PathFollower
$ ./PathFollowerComp
```

- Odometry の起動

```
$ cd /home/refhw/workspace/positionControlRH3/OdometryRH3
$ ./OdometryComp
```

- OmniTo2WD の起動

```
$ cd /home/refhw/workspace/positionControlRH3/OmniTo2WD
$ ./OmniTo2WDComp
```

- PositionInput の起動

```
$ cd /home/refhw/workspace/positionControlRH3/PositionInput
$ ./PositionInputComp
```

- RefHardRh3 の起動

```
$ cd /home/refhw/workspace/positionControlRH3/RefHard3Base
$ sudo ./RefHardRh3Comp
```

次に、Eclipse 上から起動する（デバックを行う）方法について示す。Eclipse に RTC のプロジェクトをインポートしたとして説明を行う。プロジェクトのインポートの仕方については、3.2.3 を参照のこと。

① Eclipse を起動する。

なお、root 権限にて実行する必要がある RefHardRh3 をデバックしたい場合は、Eclipse のインストールディレクトリにて下記コマンドを実行して Eclipse を起動すること。

```
$ sudo ./eclipse
```

② デバックをしたい RTC のプロジェクトを Project Explorer から選択する。

③ Debug アイコン（図 4-3 赤枠部分）を押下する。もしくは、Project Explorer 上の RTC のプロジェクトを右クリック、コンテキストメニューから”Debug as”の”Local C/C++ Application”を選択しても同様の結果となる。

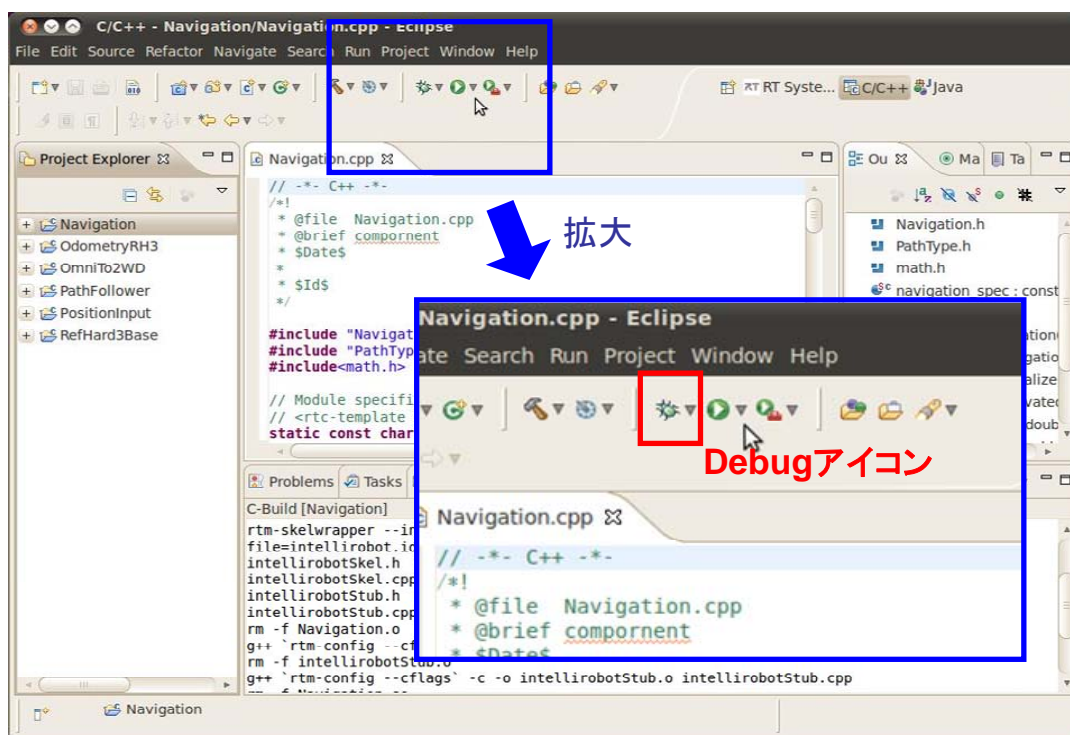


図 4-3 Debug アイコン

④ デバックツールの選択画面が表示されるので”gdb Debugger”を選択し、”OK”ボタンを押下すると、プログラム（RTC）がデバック実行される。二回目の実行からはデバックツールの選択画面は表示されない。

- ⑤ 別パースペクティブとなるデバック画面を開くための確認メッセージが表示されるので、”Yes”を押下してデバック画面を表示する。

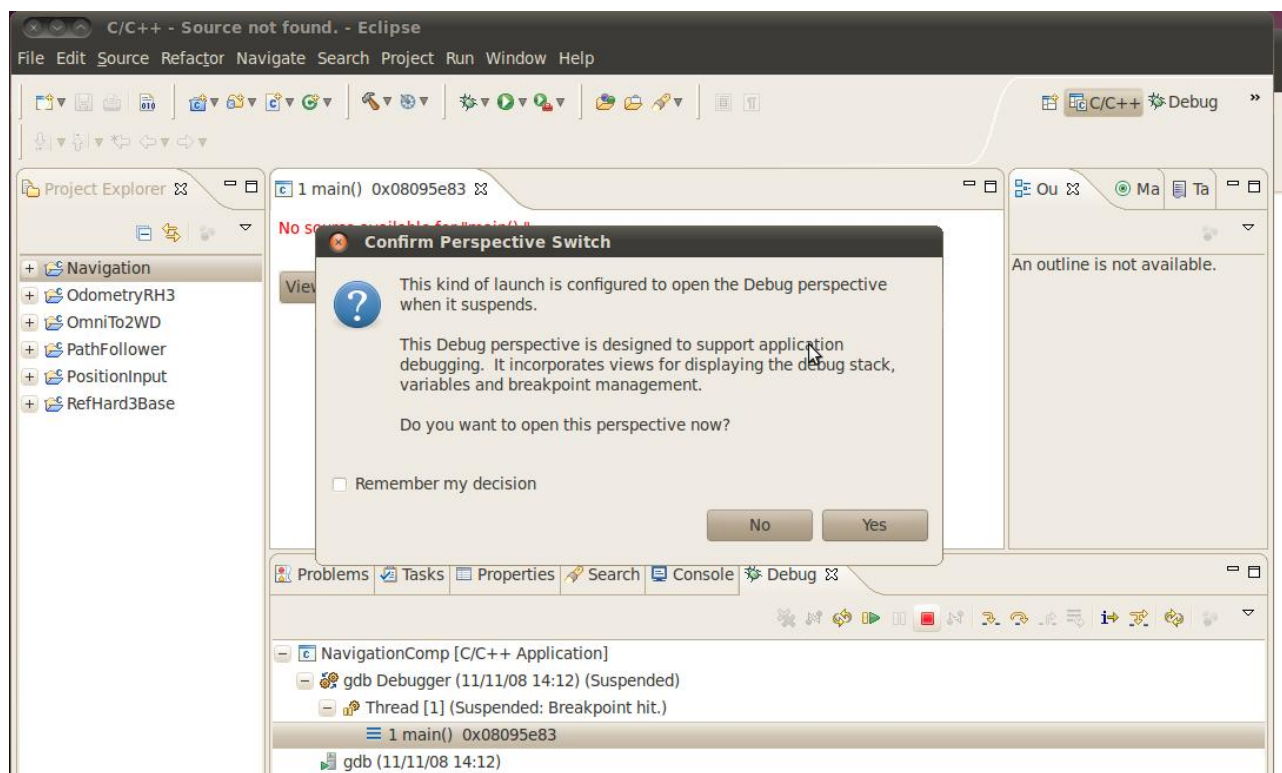


図 4-4 パースペクティブ変更確認画面

⑥ Debug ウィンドウ (図 4-5 赤枠) を操作して、デバックの開始・一時停止・終了を行う。
なお、デバックを開始する際は、Debug ウィンドウ内の”gdb Debugger”を選択した状態で、Resume アイコン (図 4-5 赤枠内の青枠) を押下する。開始操作は一時停止したデバックに対して行うことができる。

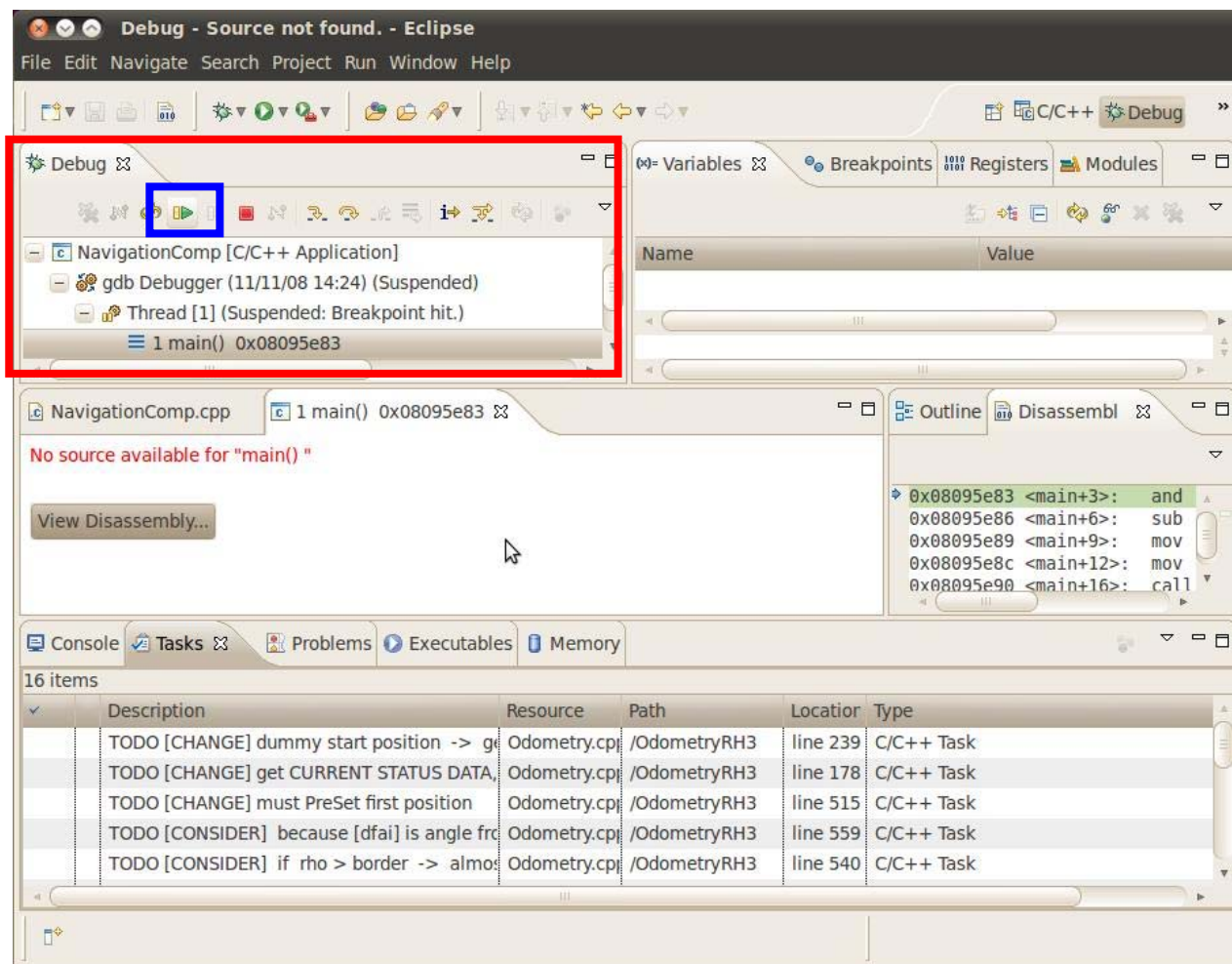


図 4-5 Debug 画面 Resume アイコン

開始しているデバックを一時停止する場合は Debug ウィンドウ内の”gdb Debugger”を選択した状態で Suspend アイコンを押下し、デバック（プログラム）を終了する場合は”gdb Debugger”を選択した状態で Debug ウィンドウ内の Terminate アイコンを押下する。

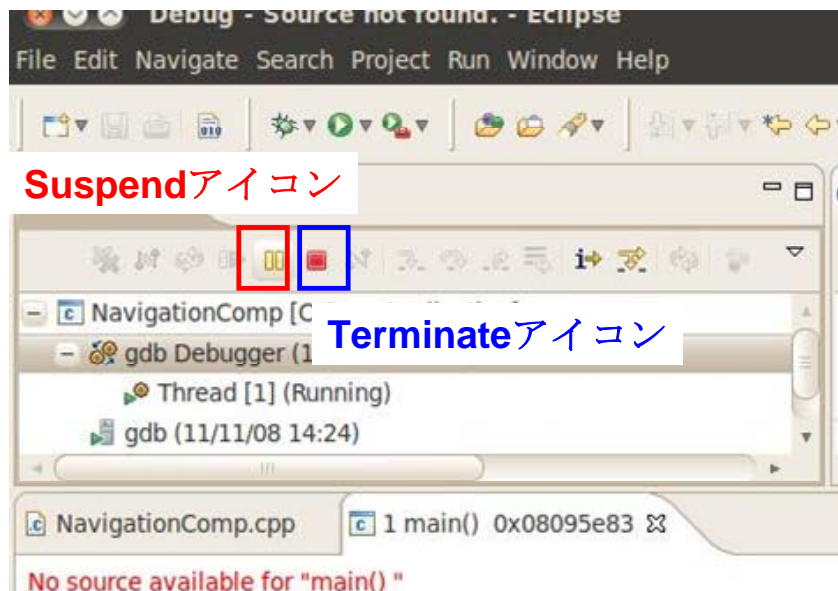


図 4-6 Debug 画面 Suspend アイコン/Terminate アイコン

また、デバックの状態には、開始（Running）・一時停止（Suspend）・停止（Terminated）が存在する。

開始（Running）状態の Debug ウィンドウを図 4-7 に示す。Thread に状態（Running）が表示されている。（図 4-7 赤枠部分）

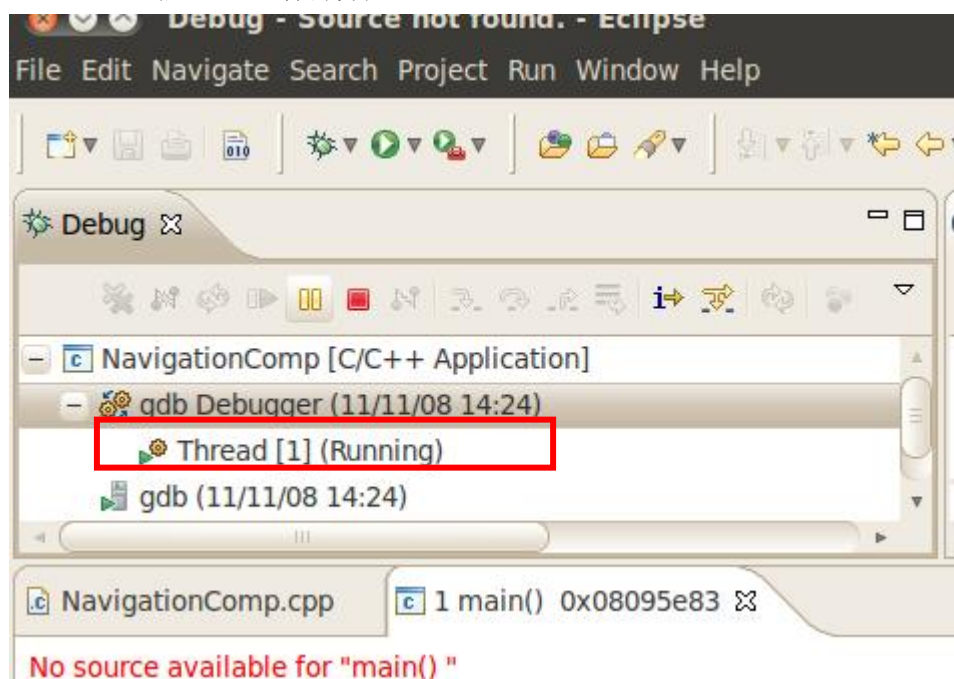


図 4-7 Debug 画面 Running 状態

一時停止 (Suspend) 状態の Debug ウィンドウを図 4-8 に示す. Thread ならびに”gdb Debugger”に状態 (Suspended) が表示されている. (図 4-8 赤枠部分)

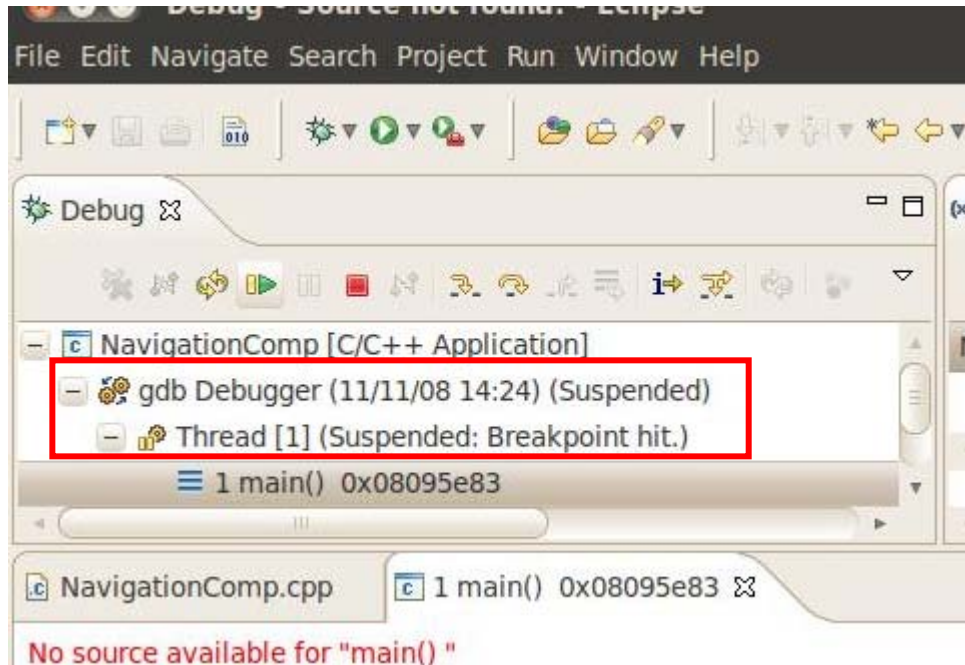


図 4-8 Debug 画面 Suspend 状態

停止 (Terminated) 状態の Debug ウィンドウを図 4-9 に示す. この状態では, デバックならびにプログラムは終了している.

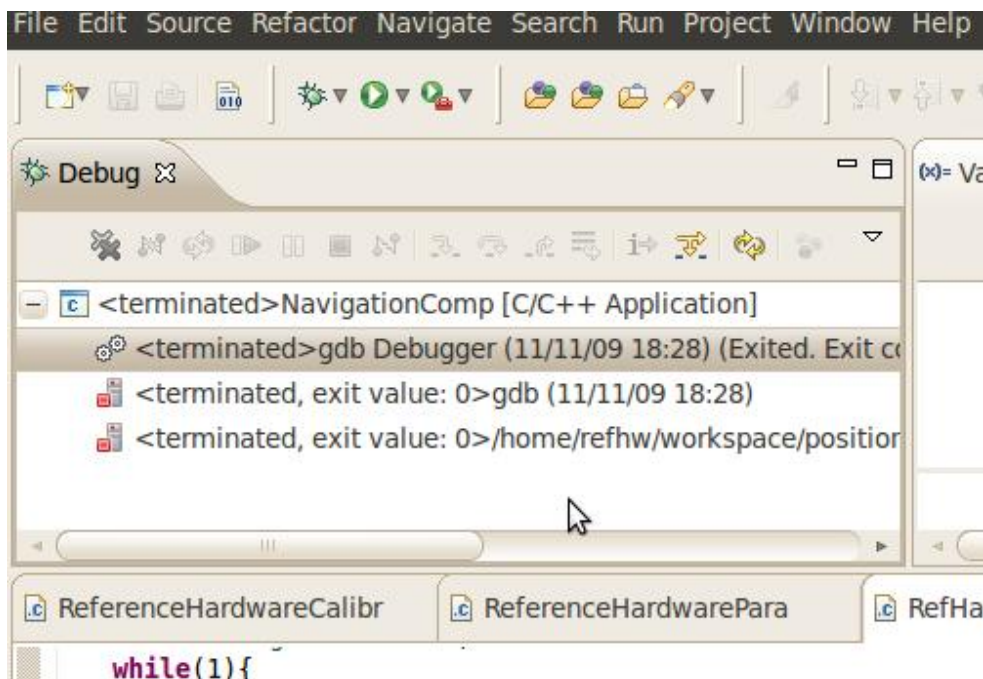


図 4-9 Debug 画面 Terminated 状態

(2) RTC の接続

- RT System Editor の起動
 - ① Eclipse を起動する。
 - ② Eclipse 上部メニューの”Window”から”Open Perspective”内の”Other”を選択する。
 - ③ 表示されたパースペクティブ選択画面から”RT System Editor”を選択し, ”OK”ボタンを押下する。
- ネームサーバの追加
 - ① 表示された Name Service View 内のネームサーバ追加アイコンを押下する。Name Service View が表示されていない場合, Eclipse 上部メニューの”Window”の”Show View”から”Name Service View”を選ぶことで表示される。図 4-10 の青枠が Name Service View, 赤枠がネームサーバ追加アイコンとなっている。

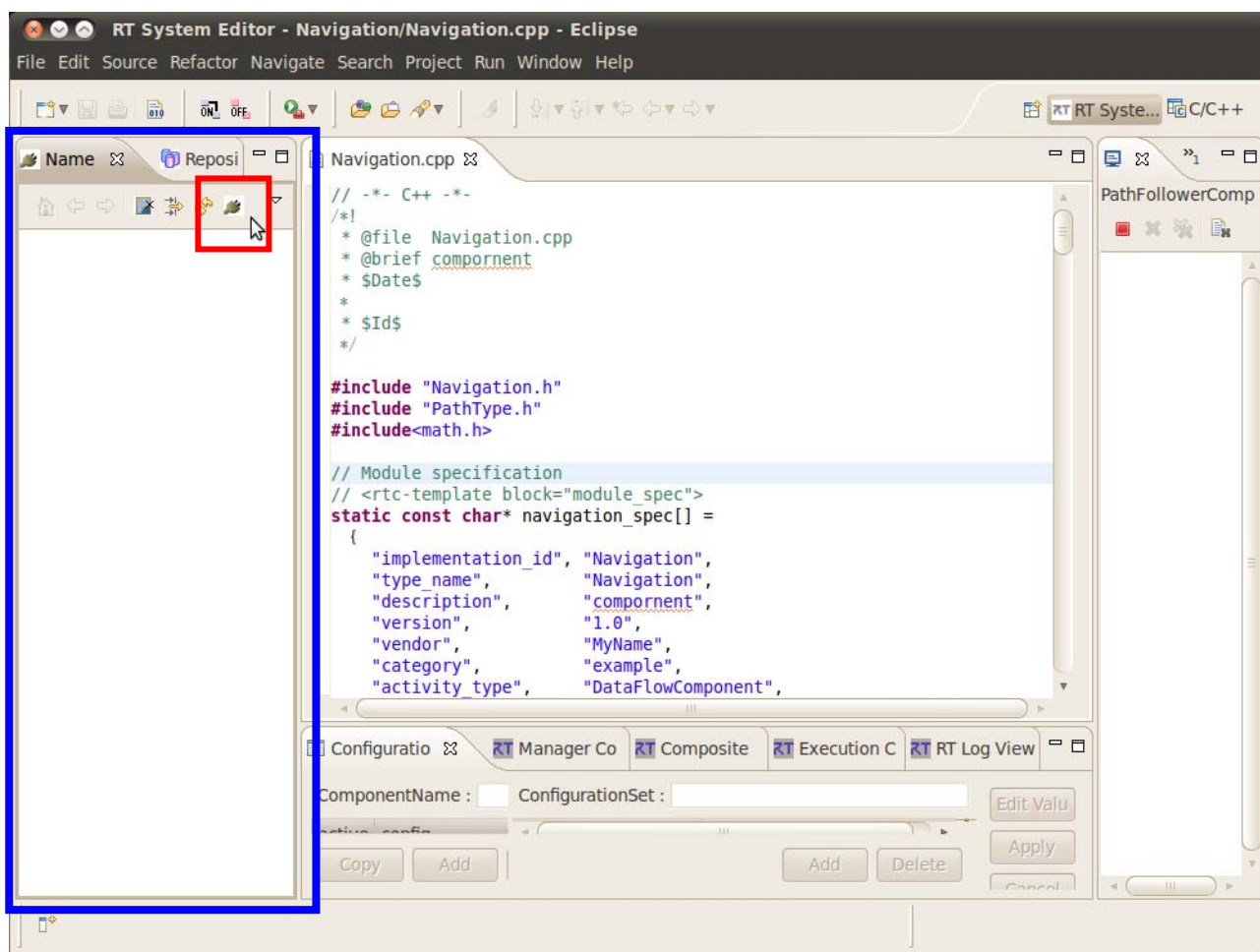


図 4-10 RT System Editor

- ② ネームサーバのアドレス入力画面が表示されるので, "127.0.0.1"と入力して"OK"ボタンを押下する.
- ③ Name Service View に入力したアドレスが表示される. アドレスを展開すると, 現在そのアドレスのネームサーバで起動している RTC が表示される. (図 4-11 参照)

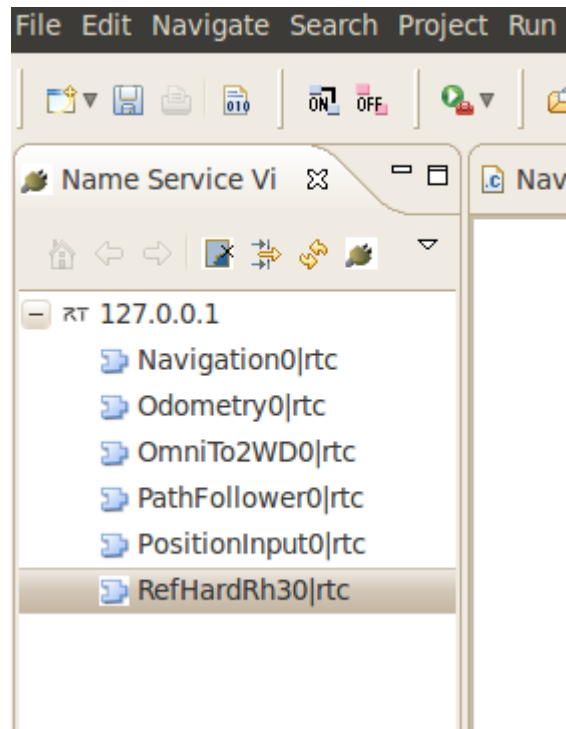


図 4-11 ネームサーバ上にある RTC 一覧

なお, 既に Name Service View に接続したいポートのアドレスが表示されていた場合にはネームサーバの追加は不要である.

RTC を起動しているのにネームサーバに RTC がいなかった場合は, `rtc.conf`を確認すること.

各 RTC のフォルダには `rtc.conf` が配置されており, その設定を元に RTC は起動している. `rtc.conf`に記載されているネームサーバの情報が下記のようにになっているか確認を行う.

```
corba.nameservers:127.0.0.1:2809
```

- ポートの接続

- ① Open New System Editor アイコン (図 4-12 赤枠) を押下し, 新しい System Diagram を開く.

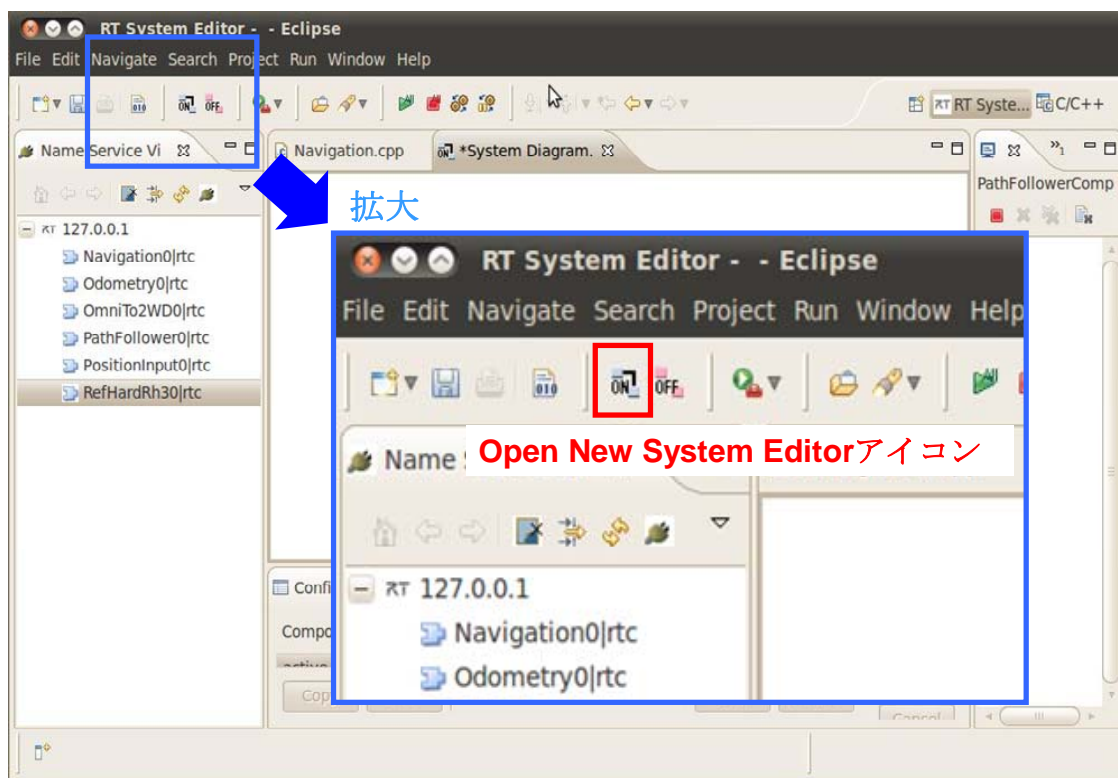


図 4-12 Open New System Editor アイコン

- ② 表示された System Diagram に, Name Service View から各 RTC をドラッグ&ドロップする.

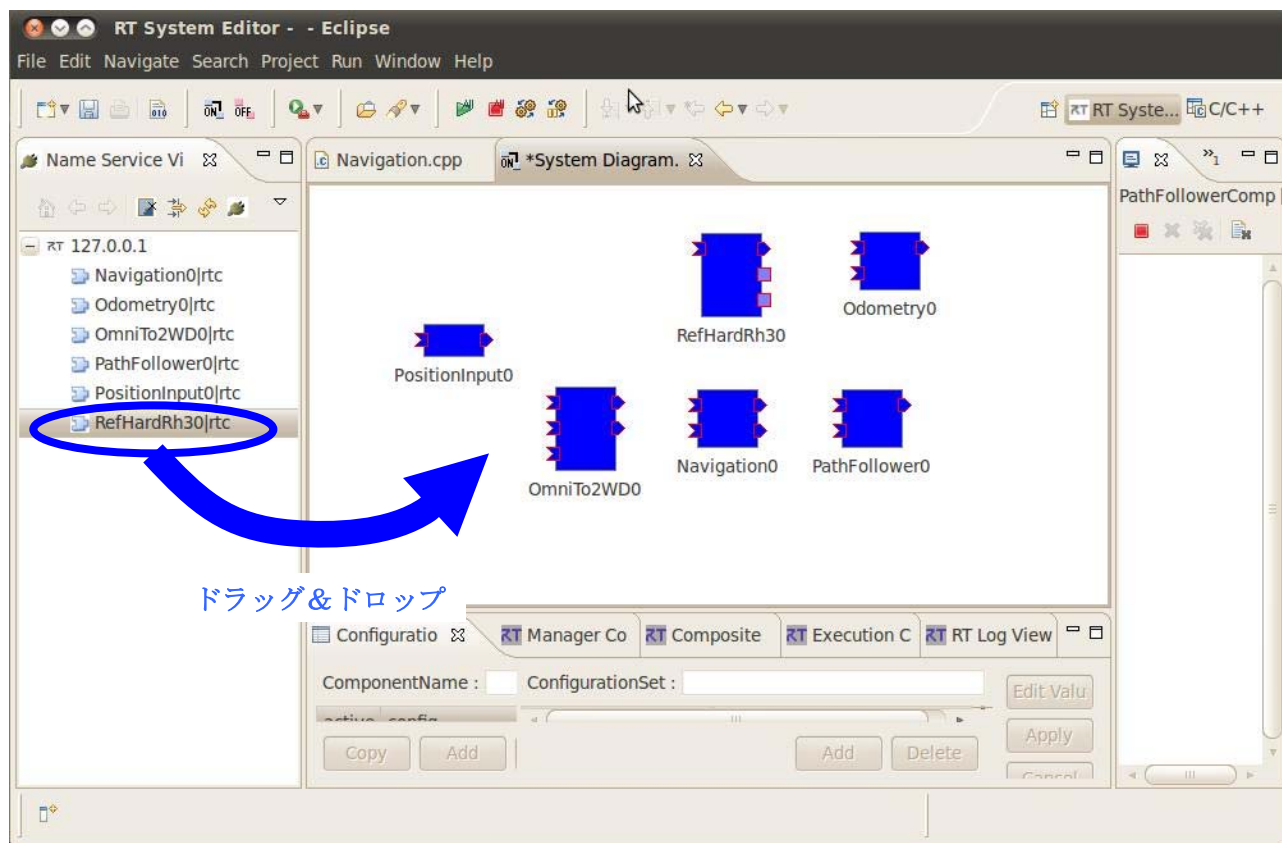


図 4-13 System Diagram 画面

- ③ 各 RTC のポートを接続する．接続は，RTC のポートをドラッグ，接続先のポートにドロップして行う．



図 4-14 ポートの接続方法

接続の構成は，表 4-1 を参照．接続後は図 4-15 のようになる．

表 4-1 接続の構成

出力ポート		接続先入力ポート	
RTC 名	ポート名	RTC 名	ポート名
Navigation0	.path	PathFollower0	.target_path
Navigation0	.status	OmniTo2WD0	statusGet
Odometry0	OdometryPosition	Navigation0	.position
Odometry0	OdometryPosition	OmniTo2WD0	current_position
Odometry0	OdometryPosition	PathFollower0	.position
OmniTo2WD0	min_path	Navigation0	.min_path
OmniTo2WD0	status	PositionInput0	.statusGet
PathFollower0	.velocity	RefHardRh30	InTargetVelocity
PositionInput0	.position	OmniTo2WD0	goal
RefHardRh30	OutWheelAngle	Odometry0	CurrentWheelAngle

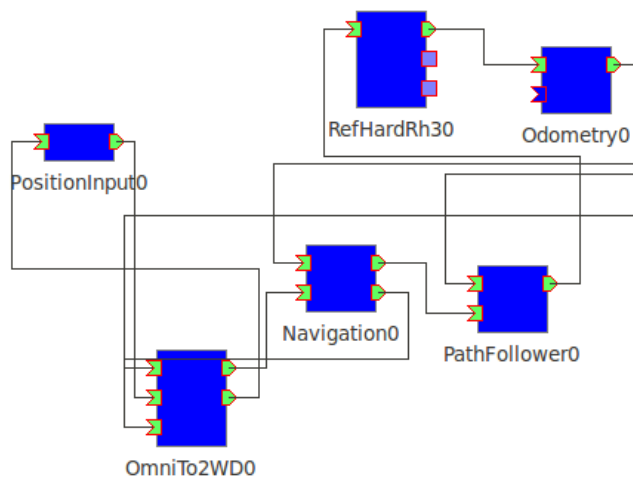


図 4-15 ポート接続後の RTC

(3) RTC のアクティベート

RTC をアクティベートするためには、System Diagram 上を右クリックし、”All Activate”を選択する。

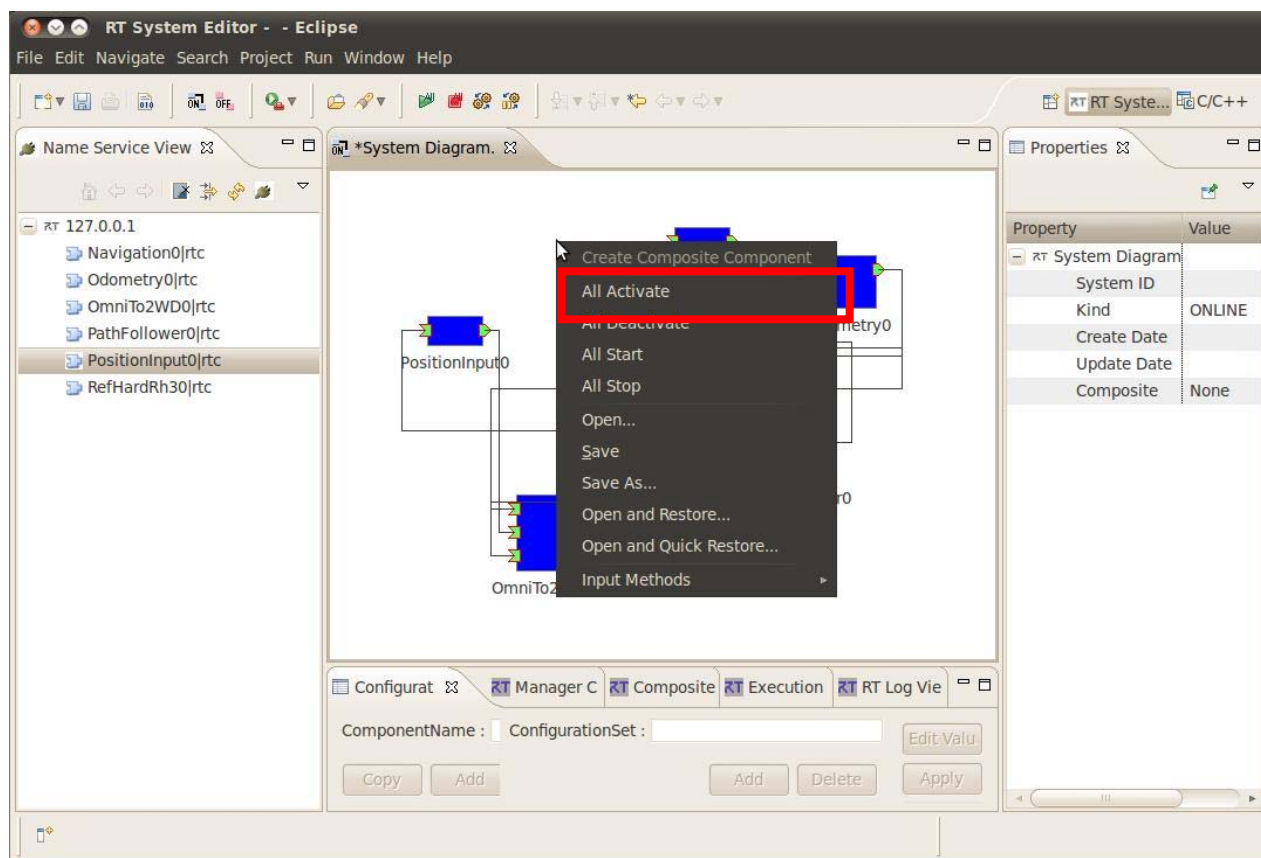


図 4-16 All Activate の選択

なお、上記方法は System Diagram 上の RTC を全てアクティベートする方法であり、一つずつ RTC をアクティベートしたい場合は、System Diagram 上の RTC を右クリックし、"Activate"を選択する。

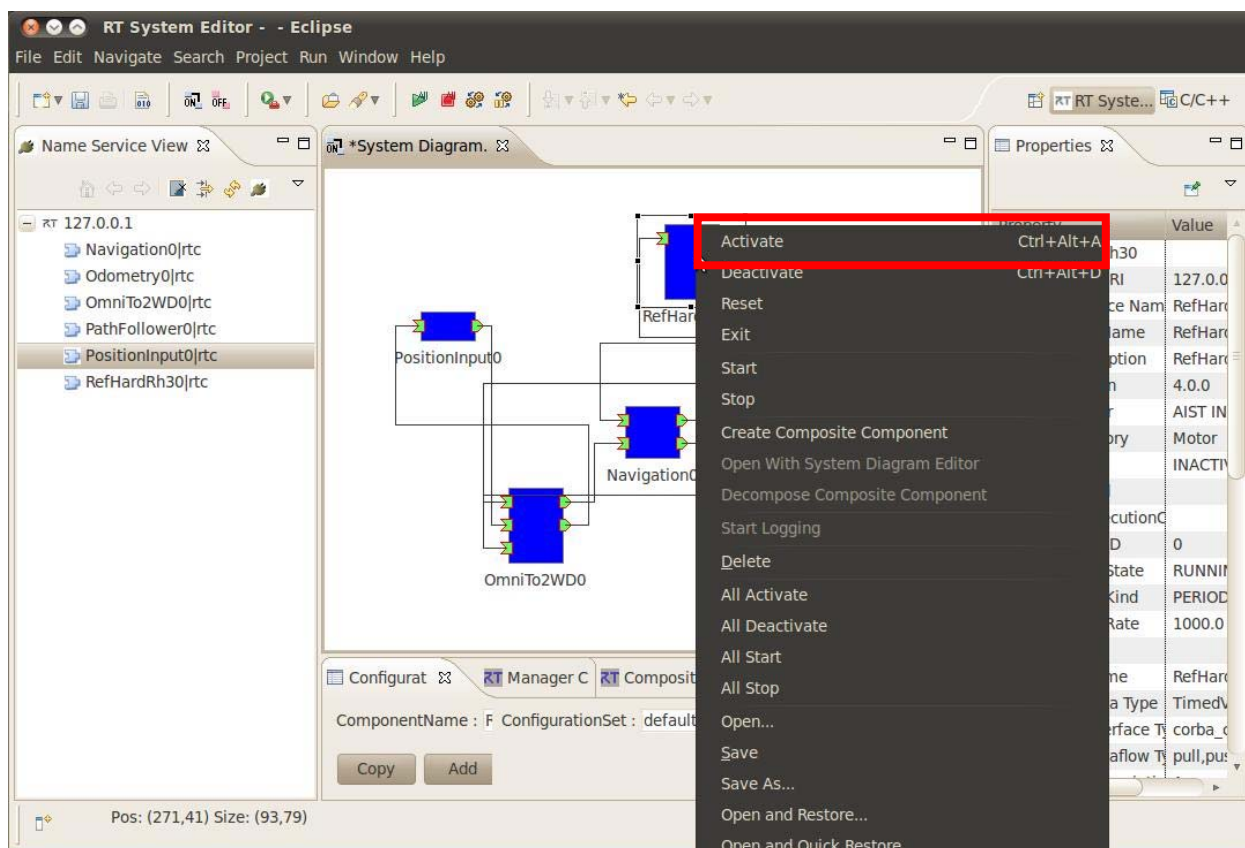


図 4-17 Activate の選択

4.3.3.自動実行

シェルスクリプトを用い自動で RTC の起動, 接続, アクティベートを行う方法について説明する. この方法を用いる場合, rtshell のインストールを行っているものとする. (2.2.3 参照)

下記コマンドでシェルスクリプトを実行することで, 自動実行ができる.

```
$ cd /home/refhw/workspace/potisionControlRH3
$ ./start.sh
```

自動実行後 RefHardRh3Comp を起動しているコンソール画面でパスワード入力が求められている場合, 実行準備が正しく出来ていない可能性があるので 4.3.1 を再度確認すること.

また, シェルスクリプト内で RTC 名, ポート名を固定で指定しているため, 初期状態から名前を変更した場合は適宜シェルスクリプトを変更すること.

4.3.4.台車移動の実行

PositionInput をアクティベートすると, コンソール画面に目標位置姿勢 (goal) か経路 (path) の入力を求められる. 目標位置姿勢を指令する場合は 1 を, 経路を入力する場合は 2 を入力し, エンターキーを押す.



図 4-18 目標位置姿勢または経路の選択

1 を入力した場合は, 目標位置姿勢 (x,y,theta) の入力を求められる. この位置姿勢は, Odometry 内部に保持された座標系に従う. Odometry が保持する位置姿勢の初期値は(0,0,0)である. x,y,theta をそれぞれ入力し, エンターキーを押す.

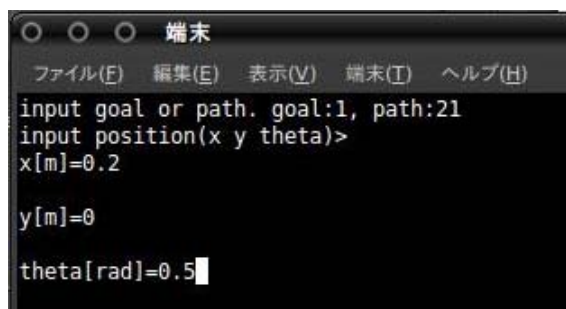


図 4-19 目標位置姿勢の入力

目標位置姿勢を入力すると, 入力結果が「result:」以下に表示され, 台車がその目標位置姿勢へ移動する. 移動が終了した時点でステータス「finish!」が表示される. 一定時間経過しても台車が移動中の場合は, 「status does not come!」が表示される.

次に、指令を続けるか質問されるので、続ける場合は 1 を、終了する場合は 9 を入力し、エンターキーを押す。



```

端末
ファイル(F) 編集(E) 表示(V) 端末(T) ヘルプ(H)
input goal or path. goal:1, path:21
input position(x y theta)>
x[m]=0.2

y[m]=0

theta[rad]=0.5
result:0.2,0,0.5
finish!
quit:9, continue:1

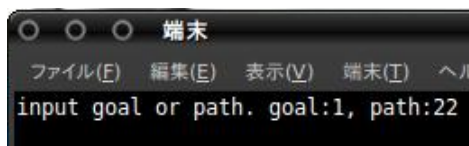
```

図 4-20 移動終了時の表示

1 を押すと、開始時の出力「input goal or path. goal:1, path:2」が再び表示される。9 を押すと、PositionInput の状態がエラー状態に遷移する。

再度目標位置姿勢を指令する際、指令値は現在値からの相対位置姿勢ではなく、Odometry が管理している座標系を基準とした絶対位置姿勢であることに注意する。

次に、経路を指令する手順を述べる。「input goal or path. goal:1, path:2」がコンソール画面に表示された時に、2 を入力し、エンターキーを押す。その後、台車は既定の経路を移動する。



```

端末
ファイル(F) 編集(E) 表示(V) 端末(T) ヘルプ(H)
input goal or path. goal:1, path:22

```

図 4-21 経路の指定

経路を変更したい場合は、PositionInput.cpp の OnInitialize() に記載された defined_path の値を変更する。

4.4. RTC のディアクティベート

RTC をディアクティベートするためには、System Diagram 上を右クリックし、「All Deactivate」を選択する。

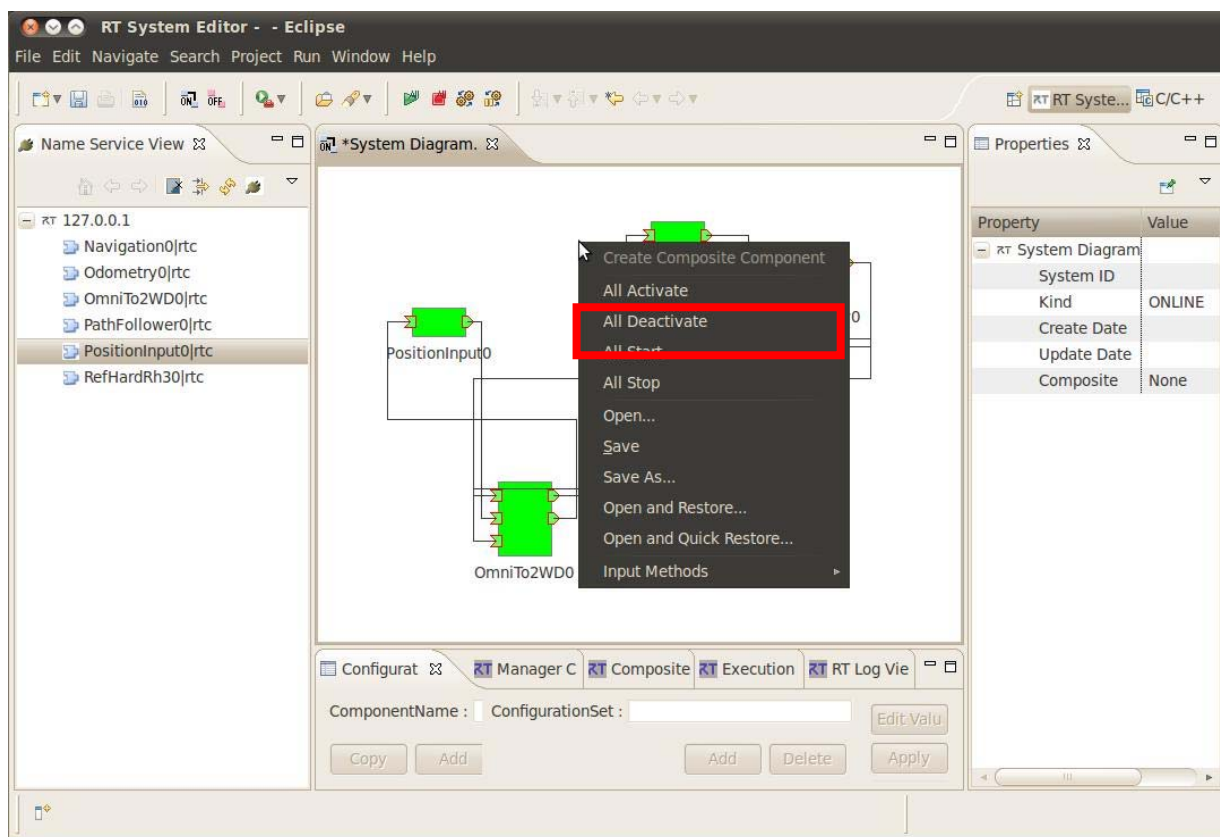


図 4-22 All Deactivate の選択

なお、上記方法は System Diagram 上の RTC を全てディアクティベートする方法であり、一つずつ RTC をディアクティベートしたい場合は、System Diagram 上の RTC を右クリックし、"Deactivate"を選択する。

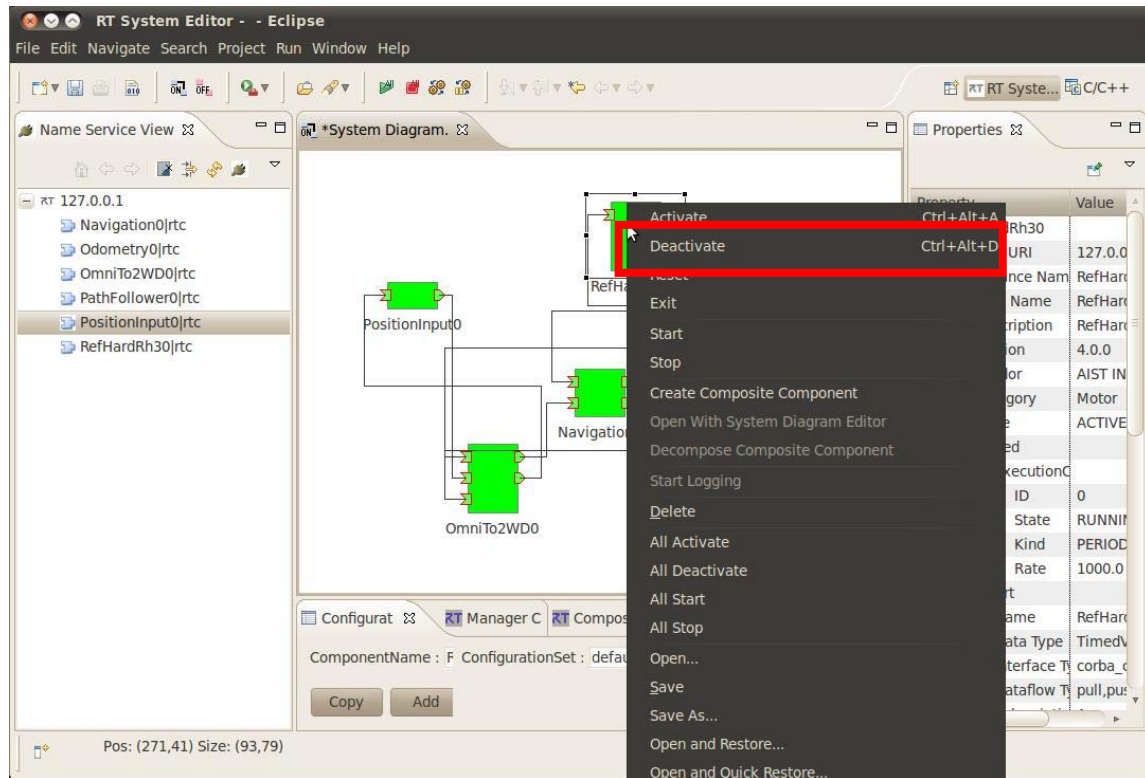


図 4-23 Deactivate の選択

また、ディアクティベート後に再度アクティベートしたい場合は 4.3.2 で RTC をアクティベートしたときと同様に、"All Activate"を選択すること。その際、Odometry は自己位置姿勢が初期値にリセットされるので注意する。

4.5.RTC の終了

RTC を終了するためには、ディアクティベート後 System Diagram 上の RTC を右クリックし、”Exit”を選択する。

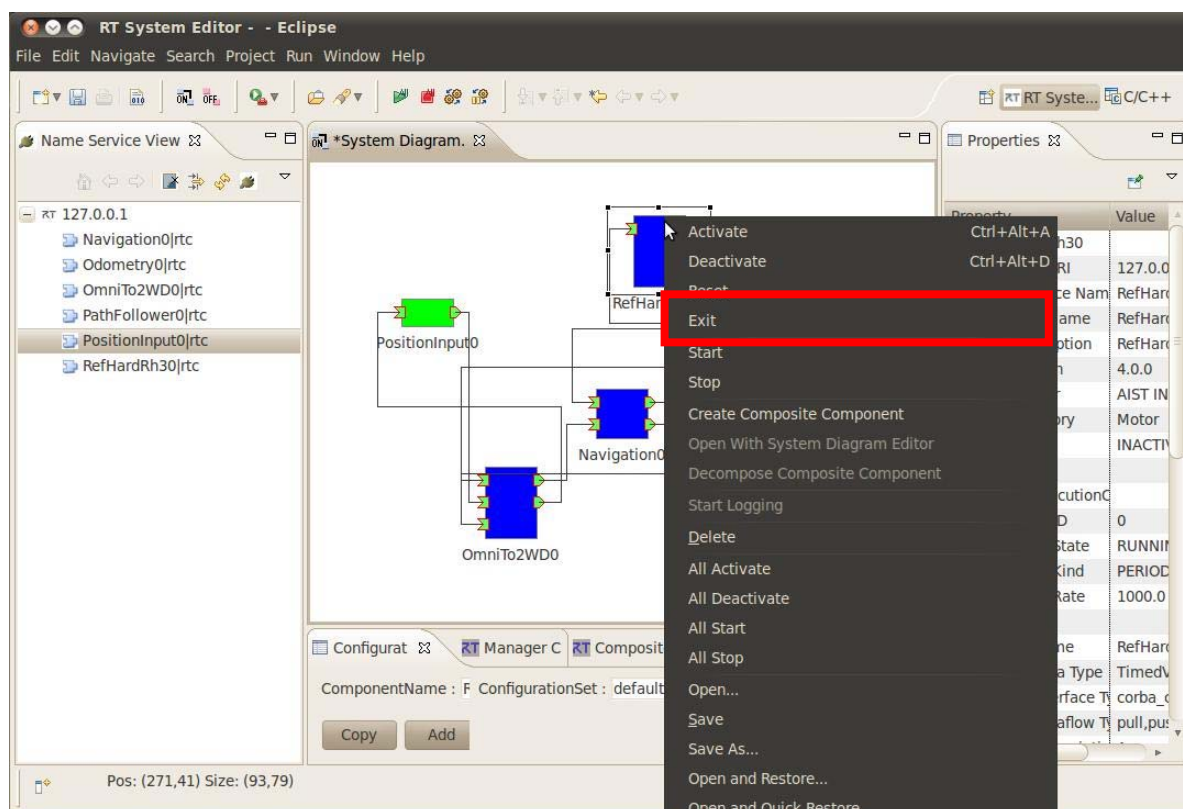


図 4-24 Exit の選択

4.6.RH3 号機を人が移動させた場合の対応

Odometry は RH3 号機の自己位置を内部で把握している．そのため，RH3 号機を人手で移動させてしまった場合，内部で把握した自己位置と実際の位置に差が生じてしまう．そのような場合，内部で把握している自己位置をリセットする必要がある．その手順について，下記に示す．

なお，下記手順では，デモ 1・2 後に人手で RH3 号機を移動させたあと，デモ 3 を行う場合を例にしている．

① デモ 1 開始

全ての RTC を起動，接続，アクティベートする．

② デモ 2 開始

特になし．

③ デモ 2 終了後

全ての RTC をディアクティベート，RefHardRh3 を終了する．

デモ 3 開始地点まで RH3 号機を人手で移動させる．

④ デモ 3 開始

RefHardRh3 を起動し，全ての RTC をアクティベートする．

この際，Odometry 内部に保持された自己位置が初期値にリセットされる．

⑤ デモ 3 終了

全ての RTC をディアクティベートし，終了する．

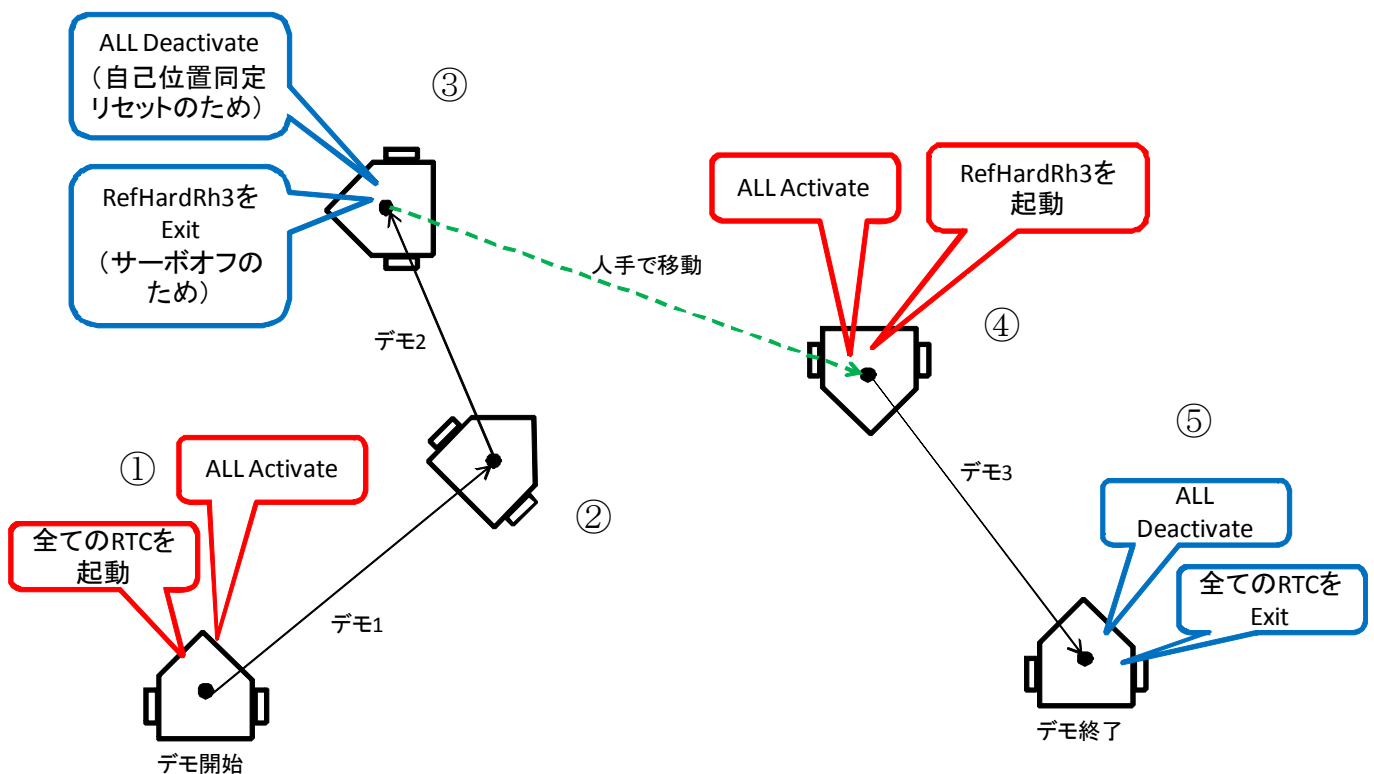


図 4-25 人手で RH3 号機を移動する際の流れ

5 特記事項

本モジュールを利用する場合には、以下の記載事項・条件に同意したものとする。

本モジュールのライセンスは **Eclipse Public License(EPL)**に従います。利用条件の詳細については、下記サイトを参照ください。なお、本モジュールは利用条件に同意した場合にのみ利用可能となっており、本モジュールを利用した時点でライセンス条項に同意したものとみなします。

Eclipse Public License <http://www.eclipse.org/legal/epl-v10.html>