

4. RT ミドルウェア学習環境を使う

4.1 ネームサーバの起動

「アプリケーション>アクセサリ>端末」で，端末を起動します．端末が起動したら，
\$sudo killall omniNames
と入力して，ネームサーバを安定稼働させるためにすでに起動しているネームサーバを削除します．

次に，

\$rtm-naming 2809

と入力して，ネームサーバを起動します．

4.2 Eclipse の起動

「アプリケーション>アクセサリ>端末」で，4.1 章とは別の端末を起動します．端末が起動したら，3.6.2 章と同じように **Eclipse** を起動します．

4.3 GrxUI を開く

3.6.2 章と同じように，「ウインドウ>パースペクティブを開く>その他」で，**GrxUI** を選択し実行します．

4.4 プロジェクトの読み込み

「newproject0 の上で右クリック>プロジェクトの読み込み」（図 3 4）で，3.6.2 章で保存したシミュレーション環境「**ThreeWheeledRobot.xml**」を選び（図 3 5），3D ビューの左上にある「ルーム」を変更し「上」にします（図 3 6）．これは，ロボットモデルの移動経路を見やすくするためであり，状況に応じて 3D ビューの視点は変更してかまいません．その後，その右にある「ズーム」のボタンをクリックし（図 3 7），画面内にシミュレーション環境がすべて入るようにします．

VRML の描画が正常でないとき（図 3 8）は，右クリックをしながら左右にドラッグすることで見えるようになります（図 3 9）．



図 3 4 プロジェクトの読み込み

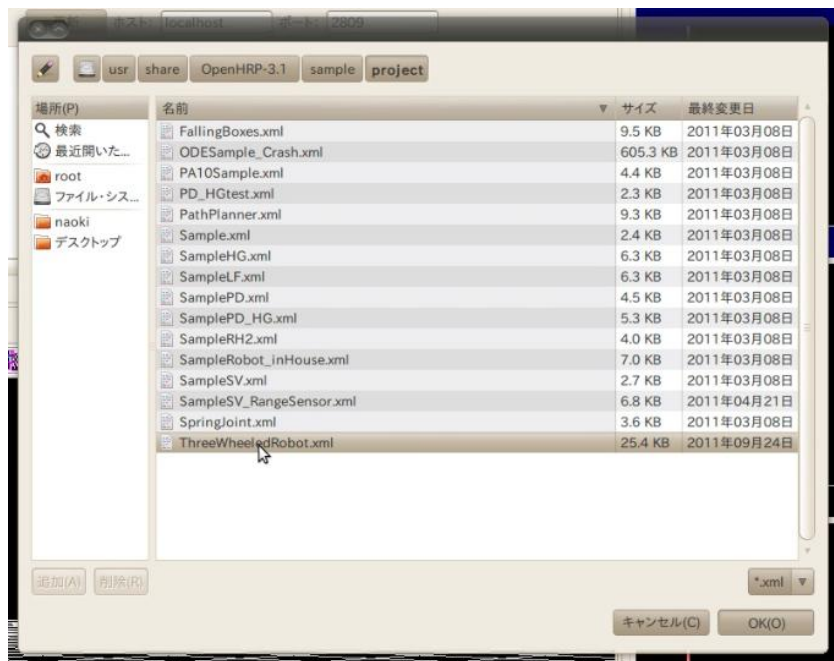


図 3 5 ThreeWheeledRobot.xml の選択

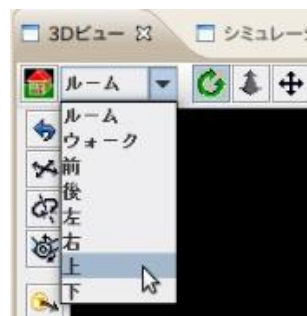


図 3 6 「ルーム」 から「上」へ



図 3 7 「ズーム」をクリック

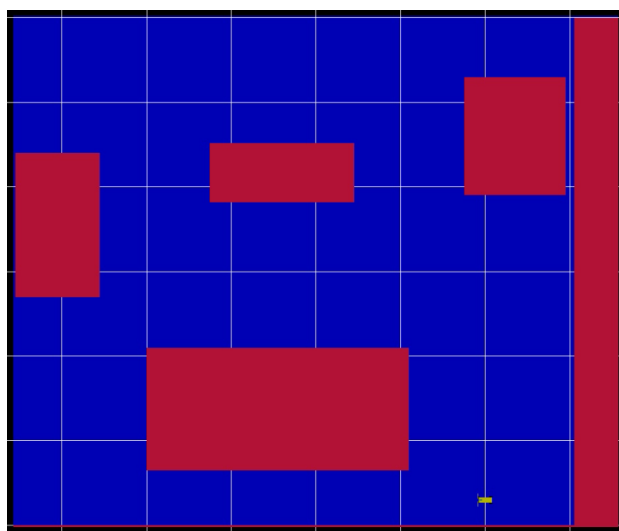


図 3 8 正常でない描画例

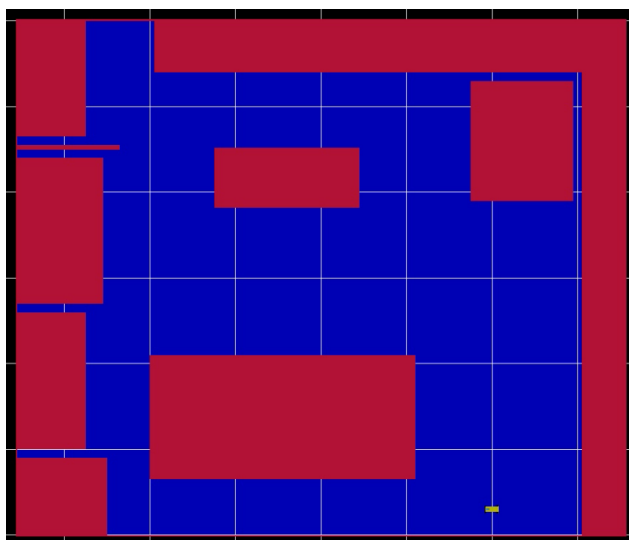


図 3 9 正常な描画例

これで，プロジェクトの読み込みが終了しました．

4.5 RT System Editor を開く

「ウインドウ＞パースペクティブを開く＞その他」から RT System Editor を選択します (図 4 0)．

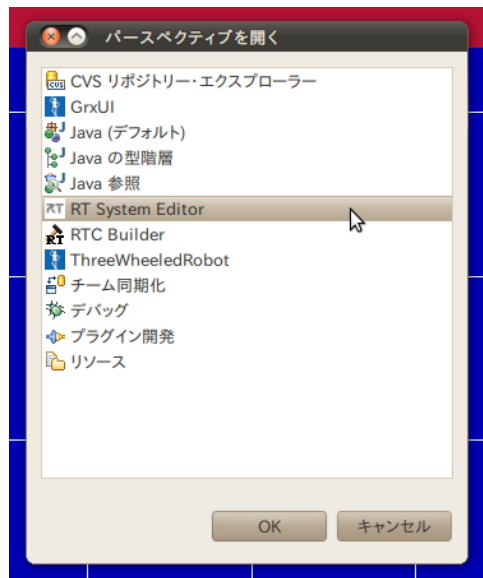


図 4 0 RT System Editor の選択

これで，RT System Editor が開かれます．

4.6 all.sh の実行

3.8 章で作成した all.sh を実行します．端末を開き，all.sh を保存している場所まで移動します．そして，実行します．

(実行例)

```
$cd /home/(アカウント名)/workspace
```

```
$sudo sh all.sh
```

実行すると，来訪者受付システムの移動機能に関する RTC 群のウインドウ（図 4 1 上）と，経路地図のウインドウ（図 4 1 右）が立ち上がります．また，既に開いていた端末（図 4 1 左）には「ready」と表示されていると思います．

このとき，来訪者受付システムの移動機能に関する RTC 群のウインドウのタブひとつひとつ（RTC ひとつひとつのタブ）を見て，「Initialized」のように表示されている文章を確認することで，RTC で何が起きているか知ることができます．RTC の実行中に知りたいことがあった場合は，該当する RTC のウインドウを確認すると良いでしょう．

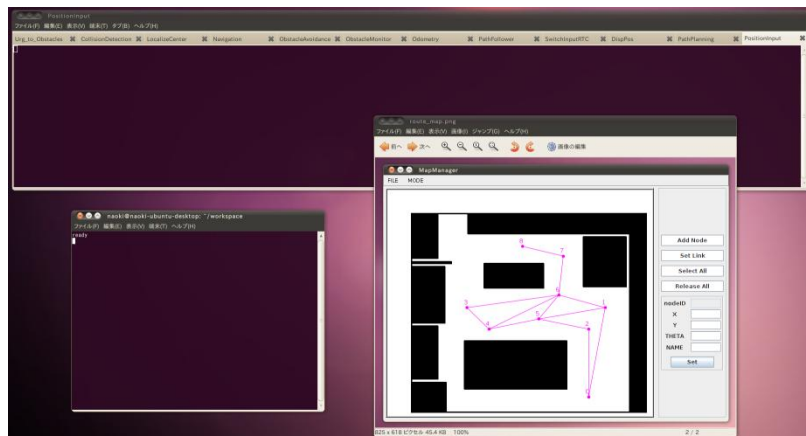


図 4 1 all.sh 実行後のウインドウ例

ここで, RT System Editor の左の Name Service View を見ると, 図 4 2 のように各 RTC が「RTC の名称+0」という形式で起動していることが確認できます. ツリーが折りたたまれている場合は, 図 4 3 のように「+」をクリックしましょう. もし確認できない場合は, 図 4 2 のように更新ボタンをクリックするか, その右にあるコンセントのようなボタンの「ネームサーバを追加」(図 4 4) で, 「localhost:2809」を追加して (図 4 5) と確認しましょう. なお, 図 4 2 の「?」のアイコンは RTC ではありませんのでご注意ください.

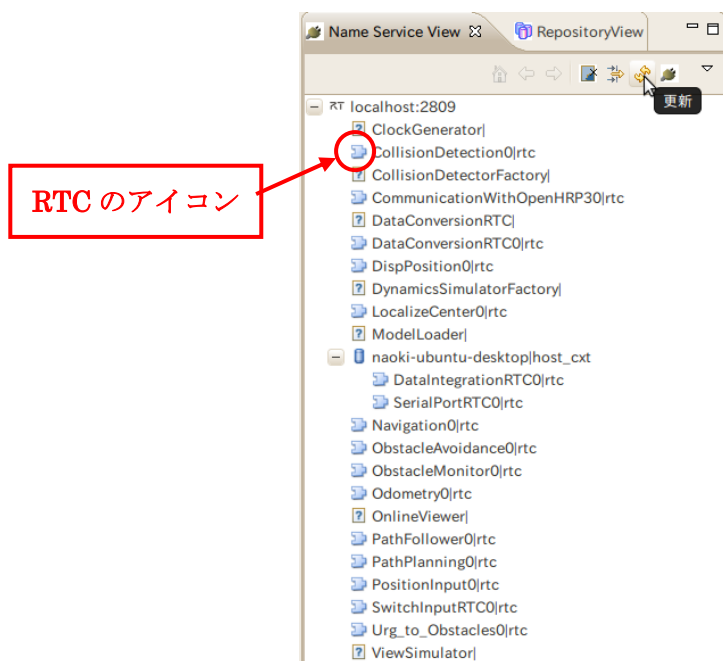


図 4 2 現在起動している RTC

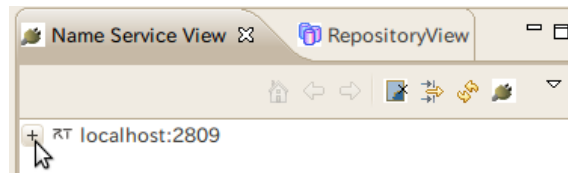


図 4 3 「+」ボタンを押す



図 4 4 ネームサーバを追加のボタン



図 4 5 localhost:2809 の追加

ここで先ほど GrxUI で開いたプロジェクト「ThreeWheeledRobot.xml」に戻ります。「3D ビュー」などのタブに並んでいる、「コントローラ」の設定を行います。

ロボット名・MODEL_CAR を選択し、下の「編集」をクリックします。そして、コントローラ名を「DataConversionRTC」（DataConversionRTC0 ではない）、実行ディレクトリを「/usr/share/OpenHRP-3.1/sample/project/./controller/DataConversionRTC」とし、実行コマンドを「bridge.sh」、コントロール時間[秒]を 0.002 に設定してください。そして左下の「了解」をクリックし、3D ビューに戻ります。ここで、図 4 6 のようにプロジェクトの保存を行なってください。

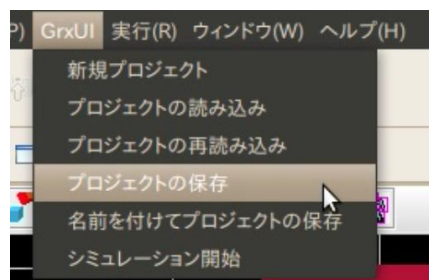


図 4 6 プロジェクトの保存

ここで再度、RT System Editor に戻ります。

4.7 RTC 同士をつなげる

図 4 7 のようにボタンをクリックして、SystemDiagram を開きます。

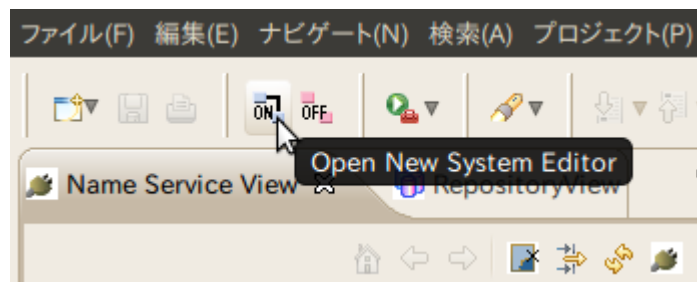


図 4 7 Open New System Editor

そして、Name Service View で起動している RTC をひとつ選択し、図 4 8 のように SystemDiagram 上にドラッグします。握り手のマークとともに RTC を選択できますので、SystemDiagram 上でマウスを離します。すると、System Diagram 上に選択した RTC が現れますので、他の RTC も同様に System Diagram 上に生成させてください。図 4 9 のように RTC を配置させると後々接続しやすいです。

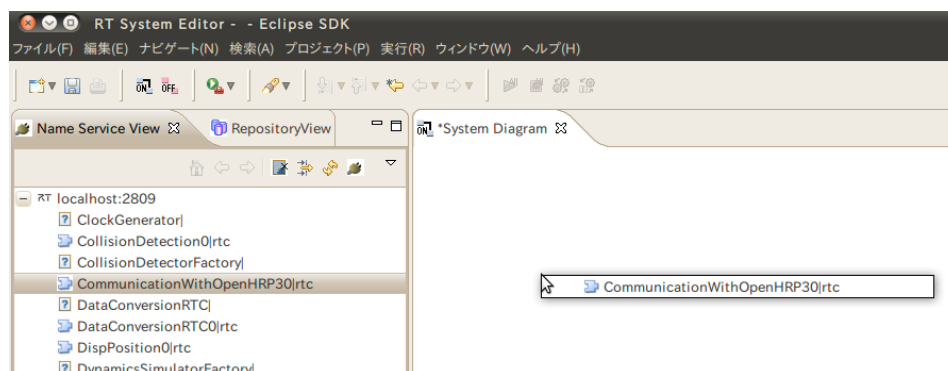


図 4 8 RTC を選択し System Diagram 上へ移動

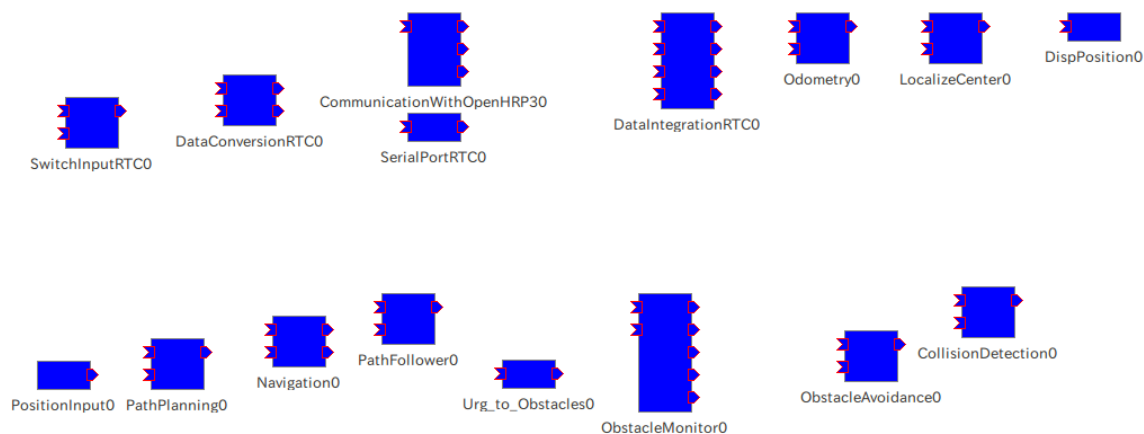


図 4 9 RTC の配置位置

次に, RTC の接続を行います. In Port (RTC 左側にある凹部分, 入力ポート) と Out Port (RTC 右側にある凸部分, 出力ポート) を結ぶために, まず, Out Port にマウスカursorを合わせると, 図 5 0 のようなマークが現れ, ドラッグすることで図 5 1 のような禁止のマークとともに線が出ます. その線を In Port にドラッグしていくと禁止のマークが消え, 図 5 2 のように Out Port のときと同じマークが現れます. このときマウスのボタンを離すことで, 図 5 3 のように Connector Profile が出現します. これはそのまま OK でかまいません. すると, 接続した In Port と Out Port が緑色となり, RTC 同士が接続されます (図 5 4).

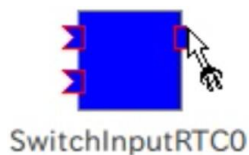


図 5 0 Out Port にマウスカursorを合わせた際の様子例

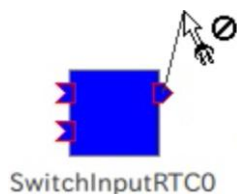


図 5 1 Out Port からドラッグした際の様子例

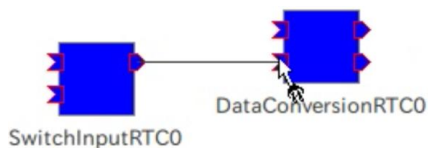


図 5 2 In Port までドラッグした際の様子例

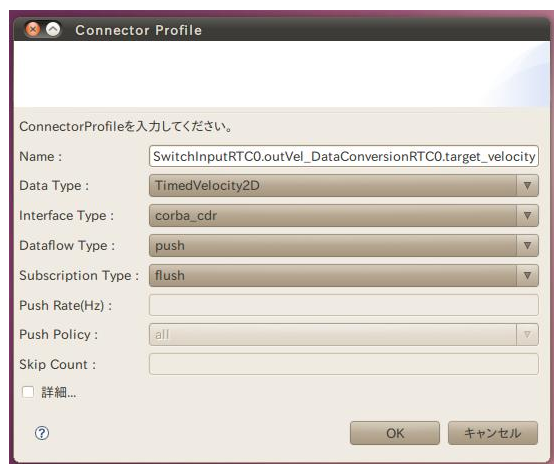


図 5 3 Connector Profile

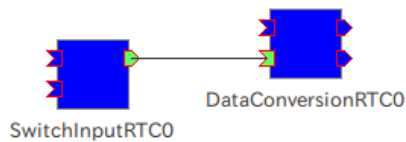


図 5 4 RTC の接続が完了した際の様子

これを図 5 5 のようにすべての RTC で行なってください。

図 5 5 では重なって表示されているため、接続が判断しにくいポートのみ解説します。まず、LocalizeCenter0 からの出力は、DispPosition0, Odometry0, PathPlanning0, Navigation0, PathFollower0 に接続してください。また、ObstacleAvoidance0 と CollisionDetection0 からの出力は SwitchInput0 の In Port に両方接続してください。CommunicationWithOpenHRP30 の上の Out Port は、DataIntegrationRTC0 の上の In Port と、DataConversionRTC0 の上の In Port に接続してください。

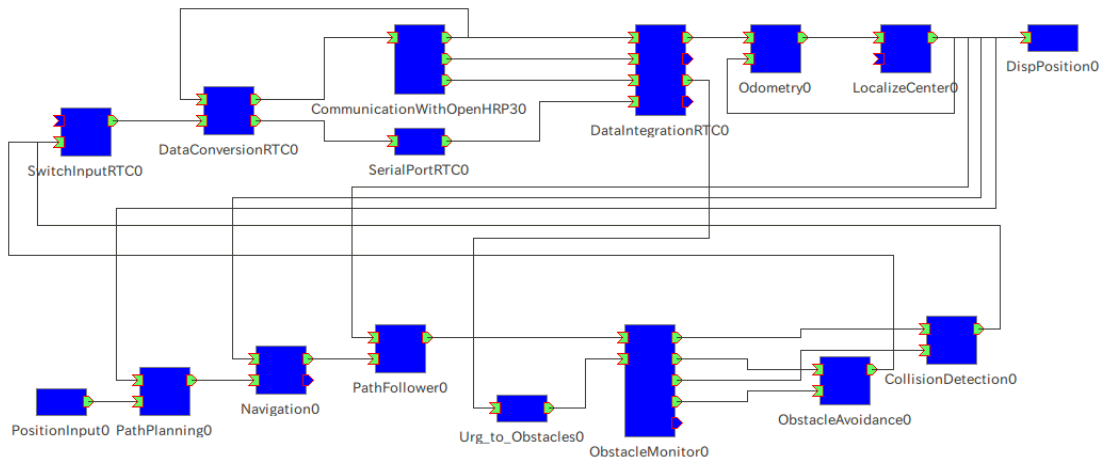


図 5 5 RTC の接続

また、障害物回避の RTC や衝突検知からの停止の RTC など障害物に関連した RTC 群を用いない場合は、図 5 6 のように RTC を接続してください。

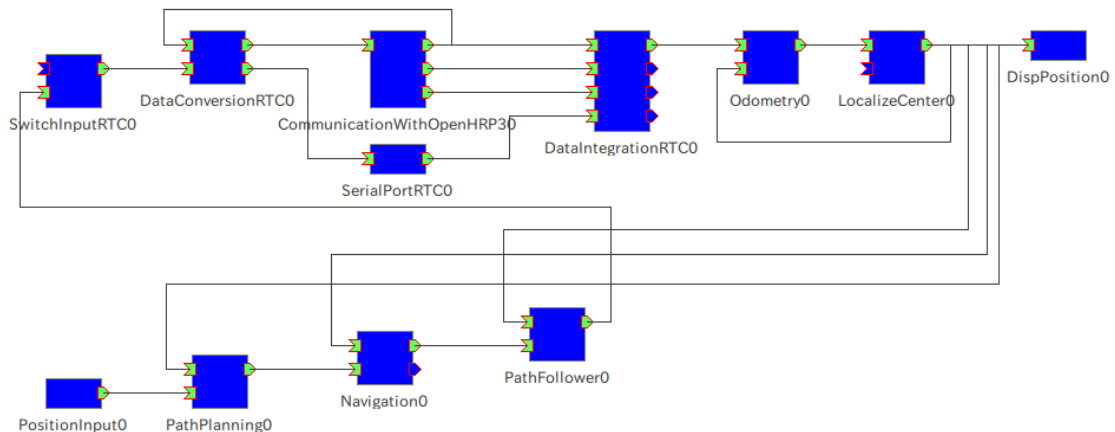


図 5 6 障害物関連の RTC を除いた RTC の接続

4.8 コンフィグレーションの変更

コンフィグレーションの修正を行います。

まず、ラジオボタンの選択を行います。最初はすべて「default」ですが、RTC「Odometry」などでは「ThreeWheeledRobot」などラジオボタンの選択ができます。このような場合は、ラジオボタンの変更後、図57のように適用をクリックします。

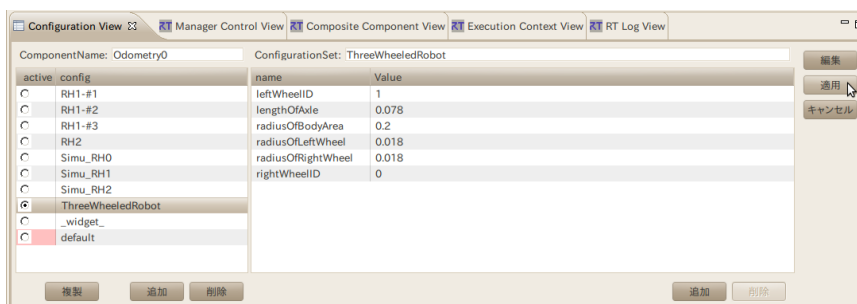


図57 ラジオボタンの変更例

また、ラジオボタンは適切でも、コンフィグレーションの中身である「Value」が適切でない場合があります。この場合は、変更したい Value の数値をクリックし目的の数値に変更した後、図58のように適用をクリックします。

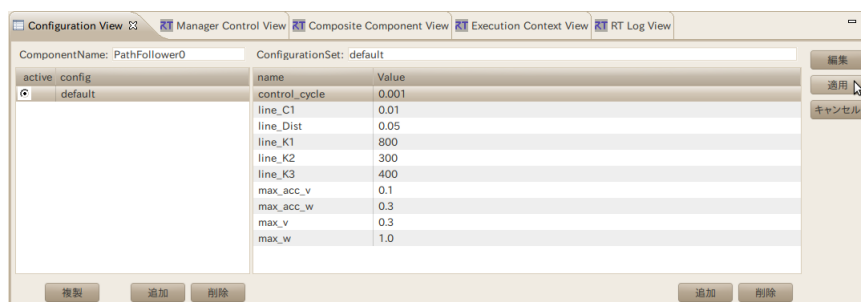


図58 Value の変更例

このようにコンフィグレーションの変更を行います。

本学習環境では、まず、ラジオボタンは以下の通りとします。

なし：CommunicationWithOpenHRP3, DispPosition, SwitchInputRTC,

「default」：PathFollower, Urg_to_Obstacles, ObstacleMonitor, CollisionDetection, ObstacleAvoidance, SerialPortRTC, DataConversionRTC

「ThreeWheeledRobot」：PathPlanning, Navigation, Odometry,

目標地点の座標のラジオボタン：PositionInput

出発地点の座標のラジオボタン：LocalizeCenter

次に、コンフィグレーションの変更を行います。表13～表17は、著者の環境のため

に変更したコンフィグレーションとなっております。3.6.1 章で作成したシミュレーション環境や目的の実験条件に合わせて、他のコンフィグレーションも各自の環境に合うように調整してください。また、PositionInput と LocalizeCenter に関しても、目標地点と出発地点を適切に設定してください。

表 1 3 PathFollower のコンフィグレーション値の修正

名称	型	デフォルト値	変更後の値
control_cycle	double	0.02	0.001
max_acc_w	double	0.1	0.3
max_v	double	1.0	0.3
max_w	double	0.3	1.0

表 1 4 Urg_to_Obstacles のコンフィグレーション値の修正

名称	型	デフォルト値	変更後の値
urg_res	double	0.25	0.573
urg_anglefront	int	540	157
urg_pos_x	double	0.0	0.0906

表 1 5 ObstacleMonitor のコンフィグレーション値の修正

名称	型	デフォルト値	変更後の値
body_radius	double	0.1	0.2
interval_weight	double	0.3	0.1
space	double	0.5	0.15
acceleration	double	0.1	1.0
waittime	double	5.0	1.0

表 1 6 CollisionDetection のコンフィグレーション値の修正

名称	型	デフォルト値	変更後の値
body_radius	double	0.1	0.2
acceleration	double	0.1	1.0
space	double	0.5	0.15
interval_weight	double	0.3	0.1

表 1 7 ObstacleAvoidance のコンフィグレーション値の修正

名称	型	デフォルト値	変更後の値
body_radius	double	0.1	0.2
acceleration	double	0.1	1.0
space	double	0.5	0.15
interval_weight	double	0.3	0.1

ここで、System Diagram 上の適当な位置で右クリックして、「Save As」を選択します。すると、図 5 9 のようにウインドウが表示されます。

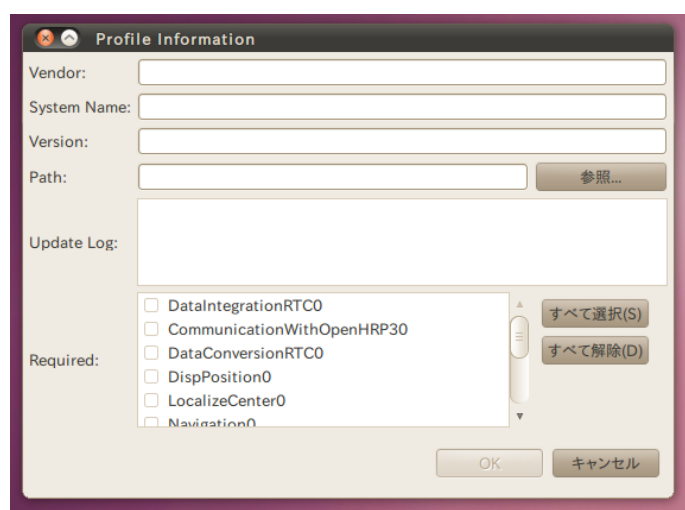


図 5 9 System Diagram の保存

これらを適切に設定すると、Required で選択した RTC を、All.sh で RTC を実行した後 System Diagram 上の適当な位置で右クリックして「Open」を選択することで、図の状態にすることができます。これで少なくとも、Name Service View にある RTC を System Diagram 上へ移動する手間を省略できます。

「Open and Restore」や「Open and Quick Restore」でも開くことはできますが、コンフィグレーションのエラーが発生する場合があります。

(システムエディタ (セーブ編) | OpenRTM-aist :

<http://openrtm.org/openrtm/ja/content/%E3%82%B7%E3%82%B9%E3%83%86%E3%83%A0%E3%82%A8%E3%83%87%E3%82%A3%E3%82%BF%EF%BC%88%E3%82%BB%E3%83%BC%E3%83%96%E7%B7%A8%EF%BC%89>)

4.9 Activate

図のように All Activate のボタンをクリックします。

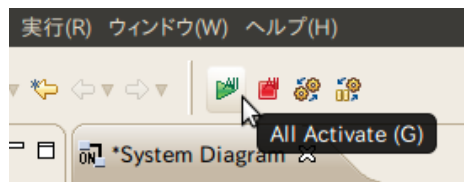


図 6 6 All Activate

すると、図 6 7 のように新しいウインドウが 2 つ現れます。

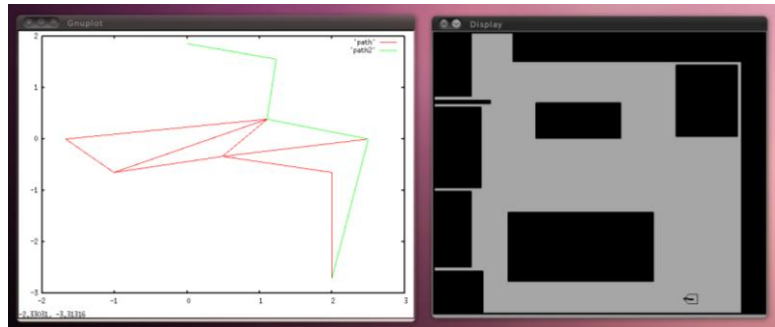


図 6 7 Activate 後に出現するウインドウ群

これで現在, DataConversionRTC と CommunicationWithOpenHRP3 以外の RTC はすべて Activate (緑色) となっています。

4.10 シミュレーションスタート

先ほど GrxUI で開いたプロジェクト「ThreeWheeledRobot.xml」に戻ります。そして、図 6 8 のように Start Simulation をクリックします。



図 6 8 Start Simulation

すると、図 6 9 のようにログのクリアを確認されます。これは「OK」にしてください。

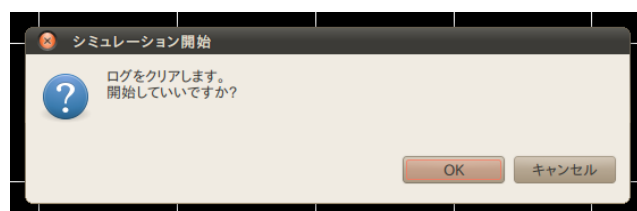


図 6 9 ログのクリアの確認

シミュレーションスタートが 2 度目以降の場合、図 7 0 のようにコントローラの再スタートも確認されます。これも「はい」にしてください。

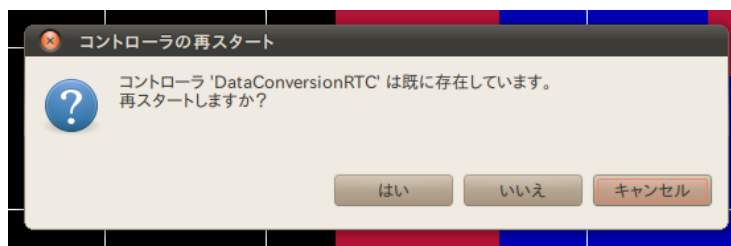


図 7 0 コントローラの再スタートの確認

これでシミュレーションが始まります。図 7 1 のように OpenHRP3 のシミュレーション環境とロボットモデル前方のカメラ画像、DispPosition によるロボットの移動した軌跡と現在位置が変化して表示されます。

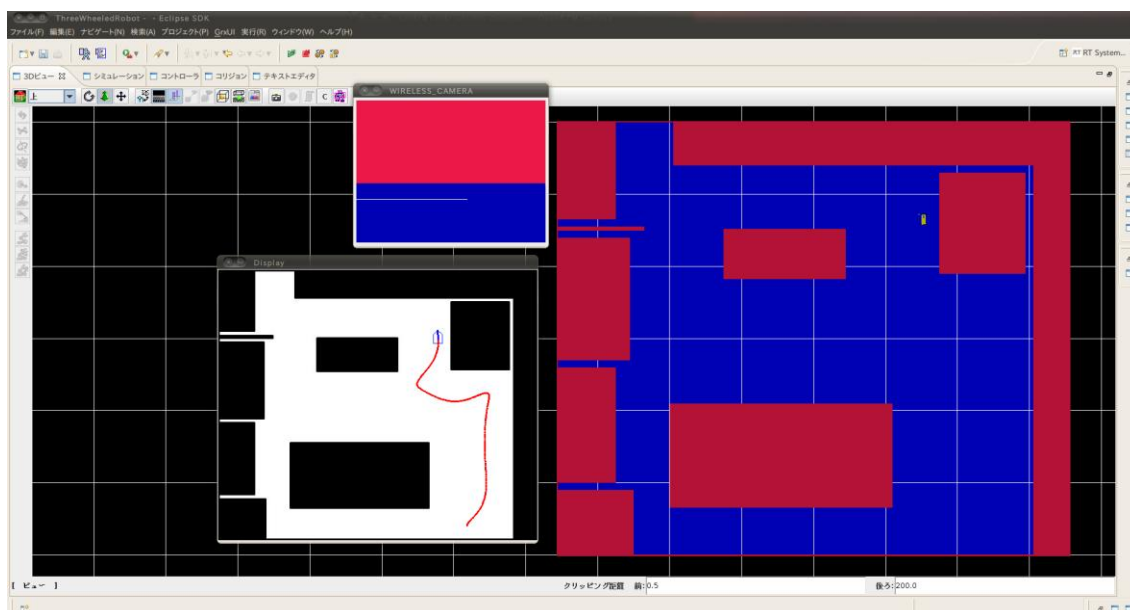


図 7 1 シミュレーション実行中の例

また、シミュレーションをスタートすることで、RT System Editor では、DataConversionRTC と CommunicationWithOpenHRP3 と DataIntegrationRTC が Acticate (緑色) になります。

4.11 シミュレーションストップ

図 7 2 のように目標地点に到着し、ロボットモデルが到着し次第シミュレーションは終了です。また、その他の条件によりロボットモデルが停止しても終了です。

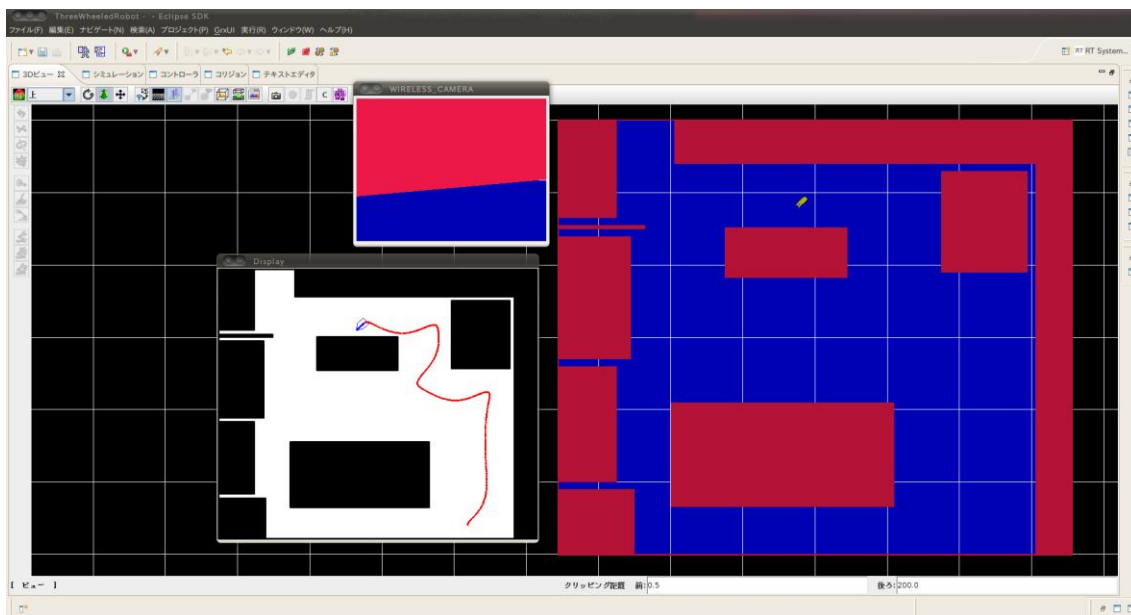


図 7 2 シミュレーション終了時の例

このとき、図 7 3 のようにシミュレーション終了をクリックすることで、図 7 4 のようにシミュレーション時間が表示され終了します。

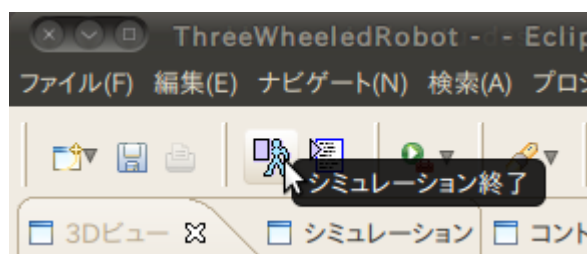


図 7 3 シミュレーション終了のボタン

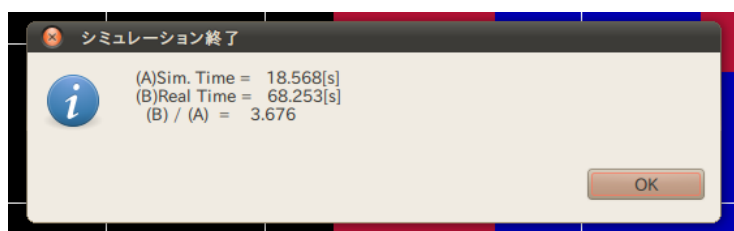


図 7 4 シミュレーション終了時の時間表示の例

4.12 Deactivate

RT System Editor に戻ると、DataConversionRTC, CommunicationWithOpenHRP3, DataIntegrationRTC が Deactivate になっています。他の RTC も Deactivate にするために、図 7 5 のように All Deactivate をクリックします。

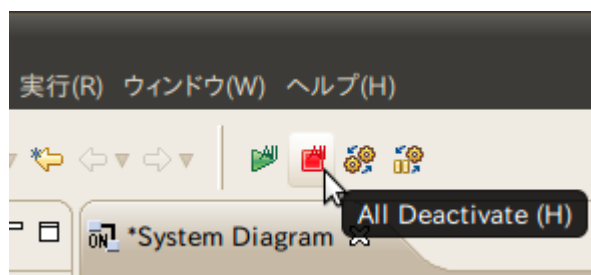


図 7 5 All Deactivate

これで、すべて Deactivate になり、シミュレーションは終了です。

もう一度シミュレーションを実行する場合は、目標地点などのコンフィグレーションを適切に設定した後、4.9 All Activate から再度始めます。

4.13 ログの確認

プログラム内にて、「RTC_DEBUG((“何かしらの情報”))」や「RTC_INFO((“何かしらの情報”))」のような書式で記されているものについては、生成されたログ内で確認することができます。また、それ以外の情報もログ内には記されているため、エラーが発生したときなどの情報の収集に役立ちます。

来訪者受付システムに関しては、`/opt/RH/(それぞれのディレクトリ)/log` にて RTC ごとのログが保存され確認できます。

SerialPortRTC (ProxyRTC), DataConversionRTC, DataIntegrationRTC に関しては、それぞれのディレクトリ内にある「rtc***.log」にてログが確認できます。