

次世代ロボット知能化技術開発プロジェクト
ロボット知能ソフトウェア再利用性向上技術の開発

機能仕様書

オープンソース移動知能モジュール群
経路計画・軌道追従モジュール編

次世代ロボット知能化技術開発プロジェクト
ロボット知能ソフトウェア再利用性向上技術の開発

機能仕様書

オープンソース移動知能モジュール群

自己位置姿勢推定モジュール編

V e r . 1 . 0

2 0 1 1 年 6 月 3 0 日

R T C 再利用技術研究センター

V e r . 1 . 0

2 0 1 1 年 6 月 3 0 日

R T C 再利用技術研究センター

目次

1. はじめに	1
1. 1. 本書の適用範囲	1
1. 2. 関連文書	1
1. 3. 本書を読むにあたって	1
2. 機能仕様	2
2. 1. 機能概要	2
2. 2. モジュール構成	2
3. RTC 仕様	4
3. 1. PathPlanning (経路計画コンポーネント)	4
3. 1. 1. 機能概要	4
3. 1. 2. 動作環境	4
3. 1. 3. ポート情報	5
3. 1. 4. コンフィグレーション	5
3. 1. 5. 入出力データフォーマット	6
3. 1. 6. 設定ファイル	7
3. 2. Navigation (マップマッチングドライブコンポーネント)	8
3. 2. 1. 機能概要	8
3. 2. 2. 動作環境	8
3. 2. 3. ポート情報	9
3. 2. 4. コンフィグレーション	9
3. 2. 5. 入出力データフォーマット	10
3. 2. 6. 設定ファイル	12
3. 3. PathFollower (マップマッチング/自律移動コンポーネント)	13
3. 3. 1. 機能概要	13
3. 3. 2. 動作環境	13
3. 3. 3. ポート情報	14
3. 3. 4. コンフィグレーション	14
3. 3. 5. 入出力データフォーマット	15
3. 3. 6. 設定ファイル	16
3. 4. PositionInput (Goal 位置入力コンポーネント)	17
3. 4. 1. 機能概要	17
3. 4. 2. 動作環境	17
3. 4. 3. ポート情報	17
3. 4. 4. 入出力データフォーマット	18
3. 4. 1. 設定ファイル	18
3. 5. 経路マップデータ生成ツール (MapManager)	19
3. 5. 1. 機能概要	19

3. 5. 2.	動作環境	19
3. 5. 3.	ユーザインタフェースおよび使用方法	20
4.	特記事項	23

1. はじめに

1. 1. 本書の適用範囲

本書は、ロボット向けミドルウェア OpenRTM 上で車体速度を入力とする対向二輪型の移動ロボットを経路に沿って走行させる際の経路計画、および起動追従機能に関する智能モジュールについて記述した文書である。

本書は「次世代ロボット知能化技術開発プロジェクト」における移動智能モジュール群として構成し、動作確認したものである。本構成以外の利用における有用性、汎用性について保証するものではない。

1. 2. 関連文書

本書は以下に示す移動智能モジュール関連文書の一部である。

表 1-1 関連文書

No.	文書名	備考
1	機能仕様書 自己位置姿勢推定モジュール	
2	機能仕様書 走行系	
3	機能仕様書 経路計画・軌道追従	本書
4	機能仕様書 障害物検知・衝突回避	
5	機能仕様書 オペレータ操作	

1. 3. 本書を読むにあたって

本書は RT ミドルウェア(以下 RTM)、RT コンポーネント(以下 RTC)を用いたロボットシステム開発者を対象に記述されており、RTM、RTC や関連ツールに関する一般的な知識を持つことを前提とする。RT ミドルウェア、RTC については下記を参照のこと。

OpenRTM-aist Official Website :

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

2. 機能仕様

本知能モジュールについての機能仕様、構成を以下に記述する。

2. 1. 機能概要

本知能モジュールは対向二輪型の移動ロボットの経路を生成し、経路に従った走行を実現する。本知能モジュールでは、経路計画、マップマッチングドライブモジュール、経路追従およびゴール位置入力インターフェースが機能として提供されている。本知能モジュールを利用するためには、現在位置を推定するモジュール、車体速度に応じて車体を制御する走行系モジュールが別途必要である。

2. 2. モジュール構成

本知能モジュールは以下の構成で動作する。移動知能モジュール全体での位置付けは、移動知能モジュール群全体図を参照のこと。

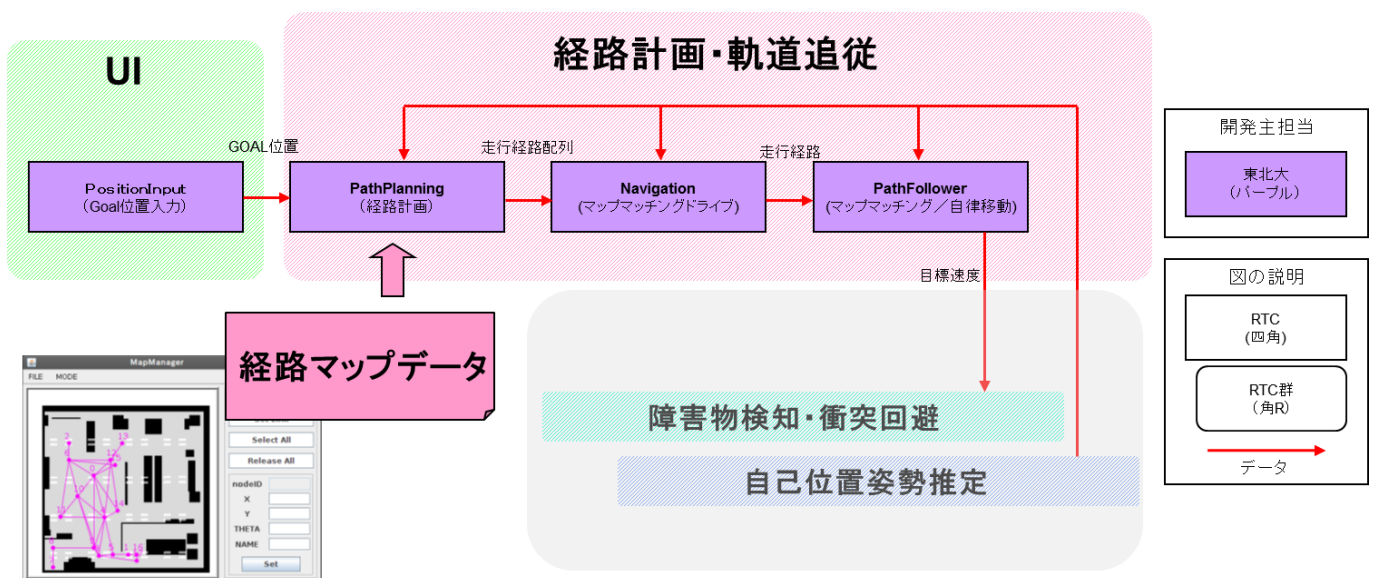


図 2-1 経路計画・起動追従モジュール構成図

3. RTC 仕様

3. 1. PathPlanning（経路計画コンポーネント）

3. 1. 1. 機能概要

本 RTC は、与えられた経路地図を使い、現在位置と目的地を結ぶ最短経路を探索するためのコンポーネントである。経路計画コンポーネントは、下図に示すような経路地図（図中赤線）から、現在位置（図中 start）と、目的位置(図中 goal)を結ぶ、最短経路（図中緑線）をダイクストラ法により探索し、出力する。

本 RTC の利用には、現在位置と目的位置を入力するモジュールが必要である。

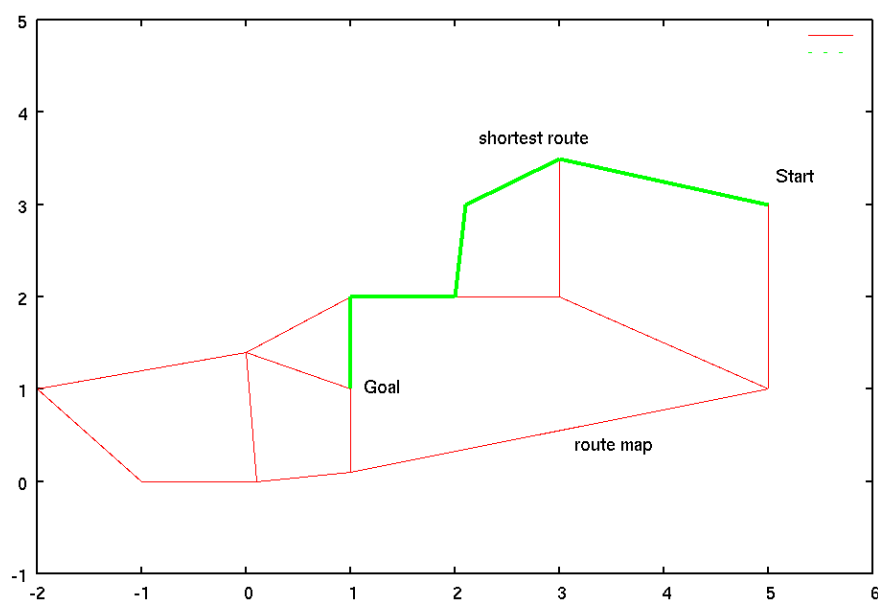


図 3-1 経路計画コンポーネント経路地図

3. 1. 2. 動作環境

本 RTC の動作環境（動作 OS、RT ミドルウェア、開発環境など）について記述する。

動作 OS	Linux (Ubuntu10.04 LTS)
開発言語	C/C++
RT ミドルウェア／バージョン	OpenRTM-aist-1.0.0-RELEASE
依存パッケージ	なし

3. 1. 3. ポート情報

A) データポート (InPort)

名称	型	説明
current_position	IIS::TimedPose2D	現在位置
goal	IIS:: TimedPose2D	目標位置

B) データポート (OutPort)

名称	型	データ長	説明
min_path	IIS::TimedPath2DSeq	経路長+1	最短経路



図 3-2 PathPlanning 単体モジュール構成図

3. 1. 4. コンフィグレーション

名称	型	デフォルト値	説明
map_file	stirng	route_map	地図ファイル名

3. 1. 5. 入出力データフォーマット

以下にデータ型詳細および地図ファイルフォーマットについて示す。
移動サブワーキンググループの共通インタフェース仕様については、「移動 SWG 共通 IF 案 101008.pdf」を参照のこと。

1) データ型詳細

- A) 入力 : Current_position、 goal
型 : IIS::TimedPose2D

```
struct TimedPose2D {
    RTC::Time tm;
    sequence<long> id;
    RTC::Pose2D data;
    sequence<double> error;
};
```

要素	説明
data.position.x	X 位置[m]
data.position.y	Y 位置[m]

- B) 出力 : min_path
型 : IIS::TimedPath2DSeq

```
struct TimedPath2DSeq {
    RTC::Time tm;
    sequence<long> id;
    sequence<RTC::Pose2D> pose;
    sequence<RTC::Velocity2D> velocity;
    sequence<double> error;
};
```

要素	説明
id[i]	経路情報列番号
velocity[i].vx	速度 x (設定値 1.0)
velocity[i].vy	速度 y (設定値 0.0)
velocity[i].va	加速度 (設定値 0.0)
pose[i].position.x	経路 X 座標
pose[i].position.y	経路 Y 座標
pose[i].heading	経路姿勢角
error[i]	エラー情報 (設定値 : 0)

※i は経路長 + 1 (末尾にゴール座標が設定されている)

2) 地図ファイルフォーマット

地図ファイルは経路地図情報として、ノードの座標と、どのノードがつながっているかを示すリンク（アーク）が記述されている。コスト値は、0 以下の場合リンクの長さをコストとする。

ノードデータ	識別子: n	x 座標	y 座標	角度
アークデータ	識別子: l	ノード ID	ノード ID	コスト

○ 地図ファイルの例

n 0.00 0.00 0.00

n 1.00 0.00 0.00

n 1.00 1.00 0.00

l 0 1 0

l 1 2 0

ファイル名は route_map で保存し、本 RTC 起動位置に配置する。

※地図生成の詳細については「経路マップデータ生成ツール (MapManager)」を参照のこと。

3. 1. 6. 設定ファイル

本モジュールでは、特殊なドライバ、ライブラリを使用しておらず、使用するにあたって、OpenRTM-aist-1.0.0-RELEASE・RTSystemEditor の標準的な設定で問題はなく、新たにドライバ、ライブラリなどのインストールは必要ない。

3. 2. Navigation（マップマッチングドライブコンポーネント）

3. 2. 1. 機能概要

本 RTC は車体速度を入力として持つ対向二輪型の移動ロボットを、与えられた経路に沿うように走行させるためのコンポーネントである。マップマッチングドライブコンポーネントは、経路地図に合わせ移動ロボットを走行させる動作コマンド列を出力する機能を持つ。

本 RTC の利用には、目的位置を入力するモジュールと、現在位置を推定するモジュール、車体速度に応じてロボットを制御するモジュールが別途必要である。

3. 2. 2. 動作環境

本 RTC の動作環境（動作 OS、RT ミドルウェア、開発環境など）について記述する。

動作 OS	Linux（Ubuntu10.04 LTS）
開発言語	C/C++
RT ミドルウェア／バージョン	OpenRTM-aist-1.0.0-RELEASE
依存パッケージ	なし

3. 2. 3. ポート情報

本 RTC のポート情報を記述する。

A) データポート (InPort)

名称	型	データ長	説明
position	IIS:: TimedPose2D	1	現在位置
min_path	TimedPath2DSeq	経路情報数	経路

B) データポート (OutPort)

名称	型	データ長	説明
path	Path	1	動作コマンドなど
status	TimedState	1	現在状態

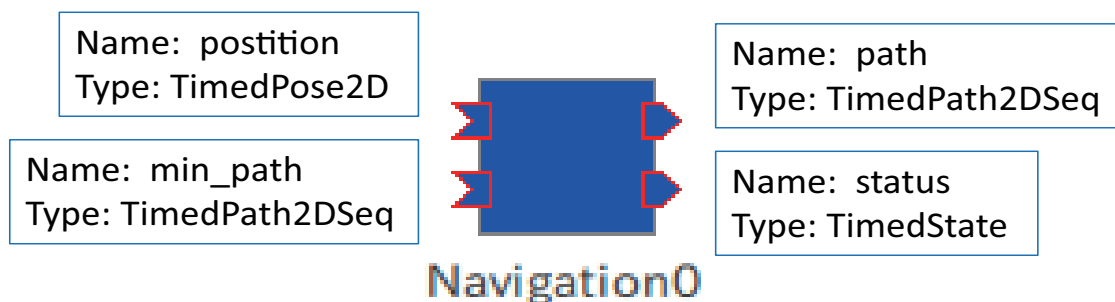


図 3-3 Navigation 単体モジュール構成図

3. 2. 4. コンフィグレーション

コンフィグレーション設定について記載する。

名称	型	デフォルト値	説明
judge_radius	double	0.5	ウェイポイントに到達したと判断する半径
max_vel	double	0.3	走行する最大速度

3. 2. 5. 入出力データフォーマット

以下に経路走行動作およびデータ型詳細について示す。

移動サブワーキンググループの共通インタフェース仕様については、「移動 SWG 共通 IF 案 101008.pdf」を参照のこと。

1) 経路走行動作

与えられた経路列に対し沿って走行するように、現在位置に応じて直線追従、その場旋回動作を繰り返し出力する。目標位置にたどり着いたかどうかの判定は、目的地に対して、現在位置が `judge_radius` 内にあるかで判定する。

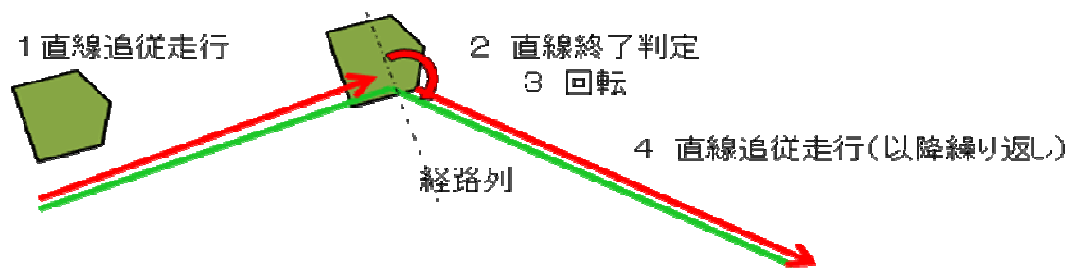


図 3-4 経路走行動作

2) データ型詳細

A) 入力 : position

型 : `IIS::TimedPose2D`

```
struct TimedPose2D {
    RTC::Time tm;
    sequence<long> id;
    RTC::Pose2D data;
    sequence<double> error;
};
```

要素	説明
<code>data.position.x</code>	X 位置[m]
<code>data.position.y</code>	Y 位置[m]
<code>data.heading</code>	姿勢角[rad]

B) 入力 : min_path
型 : TimedPath2DSeq

```
struct TimedPath2DSeq {
    RTC::Time tm;
    sequence<long> id;
    sequence<RTC::Pose2D> pose;
    sequence<RTC::Velocity2D> velocity;
    sequence<double> error;
};
```

要素	説明
pose[i].position.x	経路座標 x
pose[i].position.y	経路座標 y
pose[i].heading	経路上姿勢

※i はデータ数

C) 出力 : path
型 : Path

```
struct Path{
    short type;
    short coordinate;
    double x;
    double y;
    double theta;
    double v;
    double w;
    double r;
    Time tm;
};
```

要素	説明
x	目標位置 x
y	目標位置 y
theta	目標姿勢
v	目標速度

要素	説明
type	走行モード <pre>enum{ RUN_FREE = 0, RUN_STOP, RUN_LINEFOLLOW, RUN_TO_POINT, RUN_CIRCLEFOLLOW, RUN_SPIN, RUN_VEL, RUN_WHEEL_VEL };</pre>

D) 出力 : state

型 : IIS:: TimedState

```
struct TimedState {
    RTC::Time tm;
    long id;
    boolean isStop;
};
```

要素	説明
id	現在状態 <pre>enum{ FINISH = 0, ANGLE_CHECK, TURNING, TURNING2, GOAL_CHECK, SPEED_DOWN, STOPPING };</pre>

3. 2. 6. 設定ファイル

本モジュールでは、特殊なドライバ、ライブラリを使用しておらず、使用するにあたって、OpenRTM-aist-1.0.0-RELEASE・RTSystemEditor の標準的な設定で問題はなく、新たにドライバ、ライブラリなどのインストールは必要ない。

3. 3. PathFollower（マップマッチング/自律移動コンポーネント）

3. 3. 1. 機能概要

本 RTC は車体速度を入力として持つ対向二輪型の移動ロボットを、与えられた経路に沿うように走行させるためのコンポーネントである。マップマッチング／自律移動コンポーネントは、動作コマンドに応じた動作を行うよう、入力された直線、円弧、回転等に沿うような車体速度を生成する機能を持つ。

3. 3. 2. 動作環境

本 RTC の動作環境（動作 OS、RT ミドルウェア、開発環境など）について記述する。

動作 OS	Linux (Ubuntu10.04 LTS)
開発言語	C/C++
RT ミドルウェア／バージョン	OpenRTM-aist-1.0.0-RELEASE
依存パッケージ	なし

3. 3. 3. ポート情報

本 RTC のポート情報を記述する。

A) データポート (InPort)

名称	型	データ長	説明
position	IIS:: TimedPose2D	1	現在位置
target_path	Path	1	目標動作

B) データポート (OutPort)

名称	型	データ長	説明
velocity	IIS::TimedVelocity2D	1	目標車体速度

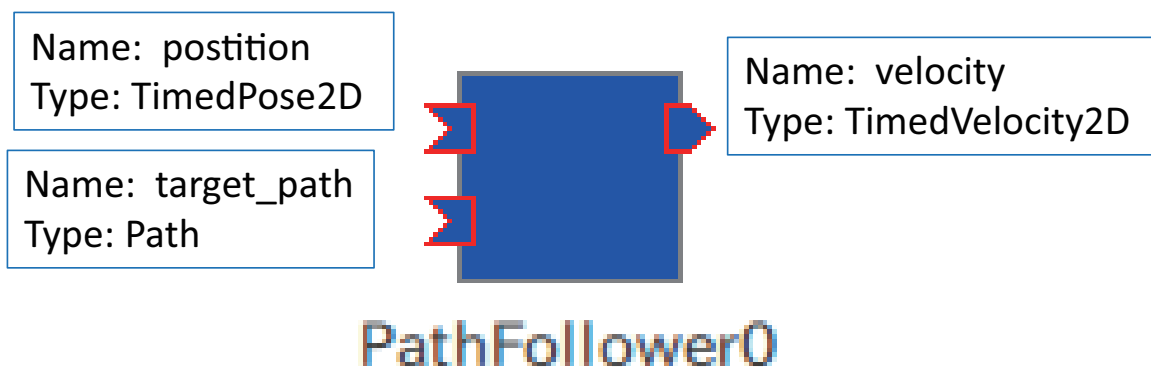


図 3-5 PathFollower 単体モジュール構成図

3. 3. 4. コンフィグレーション

コンフィグレーション設定について記載する。

複数のコンフィグレーションセットを保有する場合は各設定値も記載する。

名称	型	デフォルト値	説明
control_cycle	double	0.02	制御周期[s]
line_C1	double	0.01	経路追従パラメータ
line_Dist	double	0.05	経路追従パラメータ 経路からの距離の最大値
line_K1	double	800	経路追従パラメータ 経路からの距離に関する係数
line_K2	double	300	経路追従パラメータ 経路からの角度に関する係数
line_K3	double	400	経路追従パラメータ 角速度に関する係数
max_acc_v	double	0.1	最大加速度
mac_acc_w	double	0.1	最大角加速度
max_v	double	1.0	最大速度
max_w	double	0.3	最大角速度

3. 3. 5. 入出力データフォーマット

以下にデータ型詳細について示す。移動サブワーキンググループの共通インタフェース仕様については、「移動 SWG 共通 IF 案 101008.pdf」を参照のこと。

1) データ型詳細

A) 入力 : position

型 : IIS::TimedPose2D

```
struct TimedPose2D {
    RTC::Time tm;
    sequence<long> id;
    RTC::Pose2D data;
    sequence<double> error;
};
```

要素	説明
data.position.x	X 位置[m]
data.position.y	Y 位置[m]
data.heading	姿勢角[rad]

B) 入力 : target_path

型 : Path

```
struct Path{
    short type;
    short coordinate;
    double x;
    double y;
    double theta;
    double v;
    double w;
    double r;
    Time tm;
};
```

要素	説明
x	目標位置 x
y	目標位置 y
theta	目標姿勢
v	目標速度

要素	説明
w	目標角速度
type	走行モード enum{ RUN_FREE = 0, RUN_STOP, RUN_LINEFOLLOW, RUN_TO_POINT, RUN_CIRCLEFOLLOW, RUN_SPIN, RUN_VEL, RUN_WHEEL_VEL };

C) 出力 : velocity

型 : IIS::TimedVelocity2D

```
struct TimedVelocity2D {
    RTC::Time tm;
    sequence<long> id;
    RTC::Velocity2D data;
    sequence<double> error;
};
```

要素	説明
data.vx	移動速度[m/s]
data.vy	移動速度[m/s] (設定値 : 0)
data.va	回転速度[rad/s]

3. 3. 6. 設定ファイル

本モジュールでは、特殊なドライバ、ライブラリを使用しておらず、使用するにあたって、OpenRTM-aist-1.0.0-RELEASE・RTSystemEditor の標準的な設定で問題はなく、新たにドライバ、ライブラリなどのインストールは必要ない。

3. 4. PositionInput (Goal 位置入力コンポーネント)

3. 4. 1. 機能概要

本 RTC は車体速度を入力として持つ対向二輪型の移動ロボットを、与えられた経路に沿うように走行させる際に、ゴール位置を経路計画機能に送出するためのユーザインタフェースを含むゴール位置入力コンポーネントである。

3. 4. 2. 動作環境

本 RTC の動作環境（動作 OS、RT ミドルウェア、開発環境など）について記述する。

動作 OS	Linux (Ubuntu10.04 LTS)
開発言語	C/C++
RT ミドルウェア／バージョン	OpenRTM-aist-1.0.0-RELEASE
依存パッケージ	なし

3. 4. 3. ポート情報

本 RTC のポート情報を記述する。

A) データポート (OutPort)

名称	型	データ長	説明
positon	IIS::TimedPose2D	1	目標位置姿勢

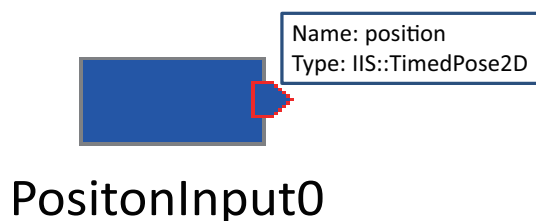


図 3-6 PositionInput 単体モジュール構成図

3. 4. 4. 入出力データフォーマット

本 RTC をアクティベートすると、ターミナル上に「\$ input position(x y theta)>」と表示される。

以下の要素をターミナル上から入力し、エンターにて決定送信する。

X	目標 X 位置、double
y	目標 Y 位置、double
heading	目標姿勢(ラジアン)、double

3. 4. 1. 設定ファイル

本モジュールでは、特殊なドライバ、ライブラリを使用しておらず、使用するにあたって、OpenRTM-aist-1.0.0-RELEASE・RTSystemEditor の標準的な設定で問題はなく、新たにドライバ、ライブラリなどのインストールは必要ない。

3. 5. 経路マップデータ生成ツール（MapManager）

3. 5. 1. 機能概要

MapManager は、本知能モジュールの PathPlanning（経路計画コンポーネント）にて利用する、経路マップデータを作成するツールである。

3. 5. 2. 動作環境

本 RTC の動作環境（動作 OS、RT ミドルウェア、開発環境など）について記述する。

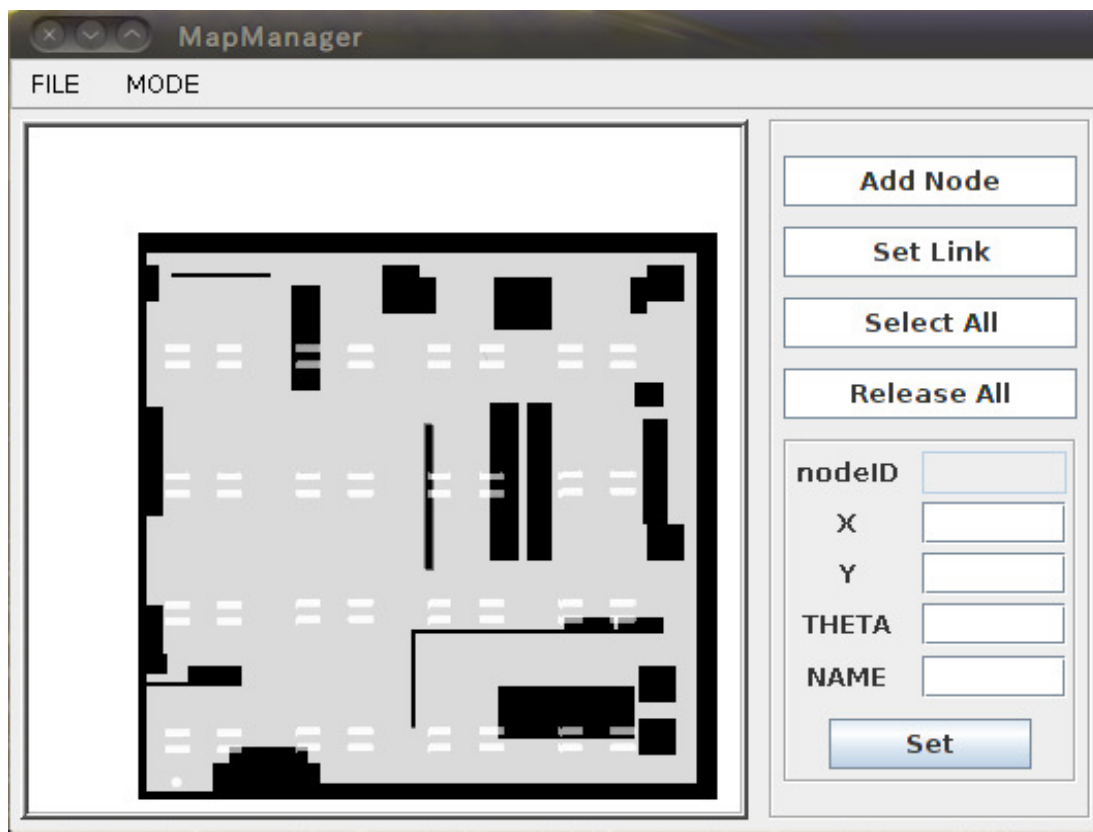
動作 OS	Linux（Ubuntu10.04 LTS）
開発言語	Java
依存パッケージ	なし

※環境表示用の画像は bmp 形式で事前に用意する。

3. 5. 3. ユーザインタフェースおよび使用方法

以下にユーザインタフェースおよびツールの詳細について示す。

1) MapManager を起動する。



ターミナル上にて、

```
$java MapManager
```

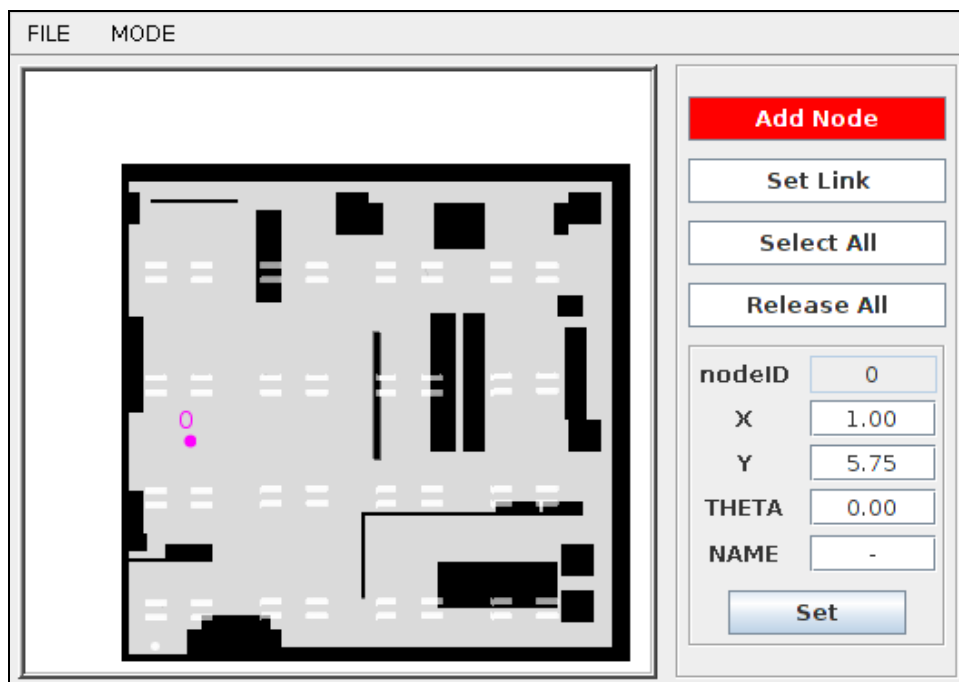
として、MapManager を起動する。事前に環境表示用の画像ファイルは **bmp** にて用意しておく。画像ファイルは MapManager 起動位置に配置する。

以下に、各機能について明記する。

機能名	説明
Add Node	ノード追加機能の ON/OFF
Set Link	2つ以上のノードを選択し、リンクを設定する
Select All	すべての要素を選択する
Release All	すべての選択済み要素を解放する
NodeID,X,Y,THETA,NAME	選択ノードの情報を表示
Set	ノード情報を入力、変更を有効化する

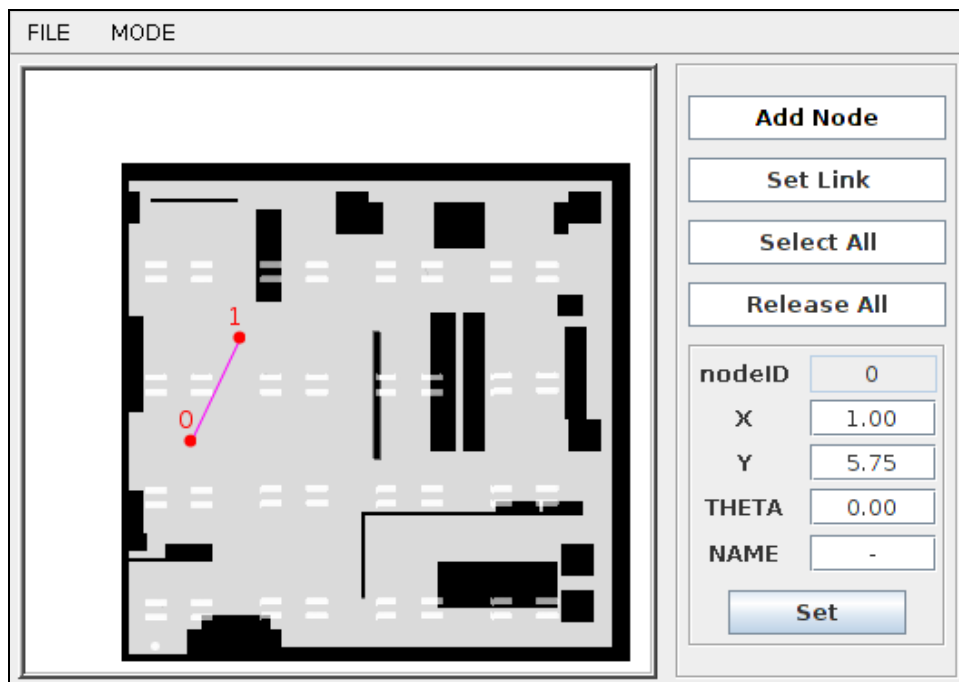
2) 経路上必要な位置にノードを設定する。

Add Node ボタンの押下にて Add Node ボタンが赤色に遷移する。



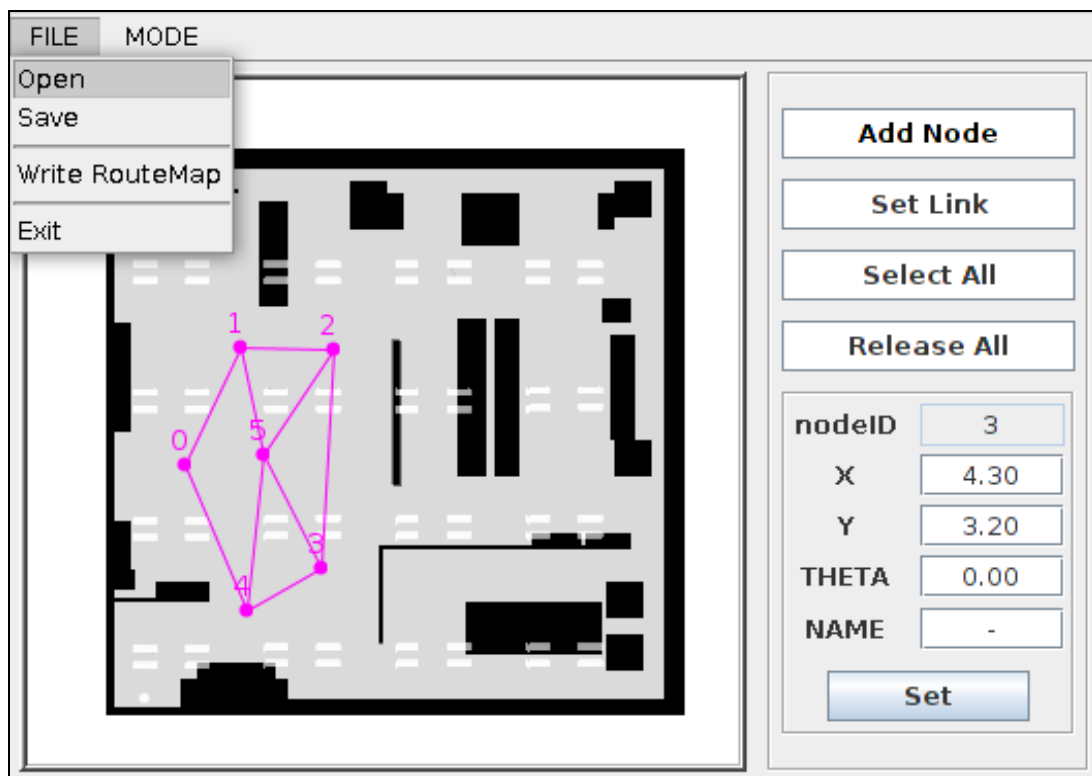
マウス操作によるノードの追加、移動も可能だが、右下のノード情報の変更によっても設定変更が可能である。

3) ノード間にリンクを設定する。



リンクを張るノードを複数選択し、Set Link の押下にて選択ノード間にリンクが設定され、左の表示用画像上にリンクが表示される。

4) 経路マップデータを保存する。



左上メニュー「FILE」から、作成した経路マップデータをファイル出力にて保存することができる。

(a) 「FILE」 - 「Save」 および 「open」

MapManager が利用する dat 形式ファイルの保存および読み込み

(b) 「FILE」 - 「Write RouteMap」

PathPlanning（経路計画コンポーネント）が利用する経路マップデータの保存

(c) 「FILE」 - 「Exit」

MapManager の終了

4. 特記事項

本モジュールをご利用される場合には、以下の記載事項・条件にご同意いただいたものとします。

- 本モジュールは独立行政法人 新エネルギー・産業技術総合開発機構（NEDO 技術開発機構）の「次世代ロボット知能化技術開発プロジェクト」（平成 20 年～平成 23 年度）において、評価を目的として構成されたものである。
- 経路計画コンポーネント、マップマッチングドライブコンポーネント、マップマッチング/自律移動コンポーネント、Goal 位置入力コンポーネントの著作権は東北大学 大学院情報科学研究科 田所研究室が、経路マップデータ生成ツールの著作権は産業技術総合研究所が所有しています。
- ドキュメントに情報を掲載する際には万全を期していますが、それらの情報の正確性またはお客様にとっての有用性等については一切保証いたしません。
- 利用者が本モジュールを利用することにより生じたいかなる損害についても一切責任を負いません。
- 本モジュールの変更、削除等は、原則として利用者への予告なしに行います。また、止むを得ない事由により公開を中断あるいは中止させていただくことがあります。
- 本モジュールの情報の変更、削除、公開の中断、中止により、利用者に生じたいかなる損害についても一切責任を負いません。

【連絡先】

RTC 再利用技術研究センター

〒101-0021 東京都千代田区外神田 1-18-13 秋葉原ダイビル 1303 号室

Tel/Fax : 03-3256-6353 E-Mail : contact@rtc-center.jp