

2026年10月16日

高度ポリテクセンター・能力開発セミナー (E0771)

「RTミドルウェアによるロボットプログラミング技術」

5. 総合演習 (2)



はじめに

- 本日の講習会のWebページ
 - <https://openrtm.org/openrtm/ja/node/7366>
 - 短縮URL
 - <https://x.gd/tjC0y>
- こちらに、必要な情報がまとまっていますので、ブックマークしてください。



概要

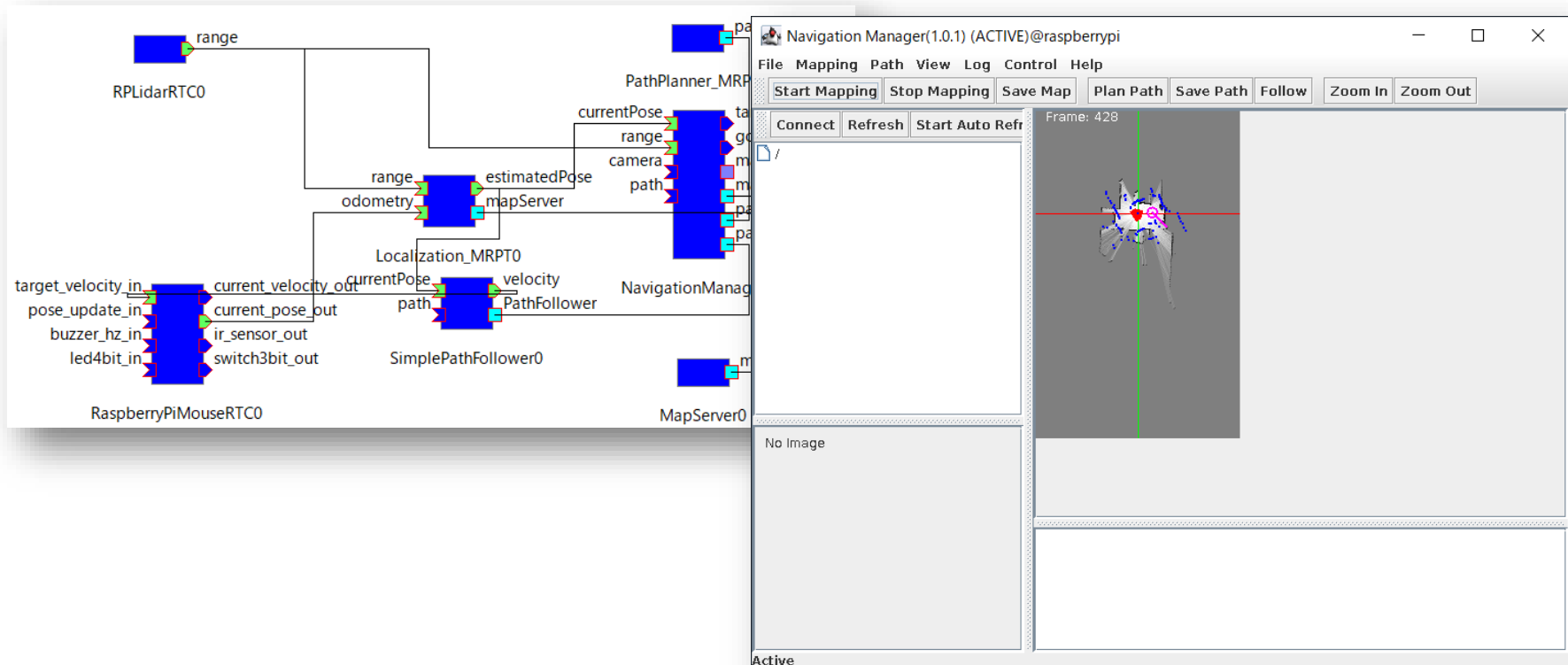
1. SLAMについて
2. 地図作成実習
3. ナビゲーション実習

目標：

1. SLAMとは何かについて学ぶ。
2. SLAMの機能のうち地図作成を実機で行う。
3. SLAMの機能のうち自己位置推定とナビゲーションを実機で行う。

実習内容 (3)

- SLAMによる
 - 地図作成
 - 自己位置推定
 - ナビゲーション



資料

- WEBページ
 - <https://openrtm.org/openrtm/ja/node/7098>
 - 短縮URL
 - <https://x.gd/BYUeB>
- SLAM用RTC等(3.RTCプログラミング演習のファイルと同じ)
 - https://github.com/OpenRTM/RTM_Tutorial/releases/download/tutorial20261015/RTM_Tutorial.zip
 - 短縮URL
 - <https://x.gd/3f5hn>

12:30-16:45	3. RTCプログラミング演習 (1) RTCBuilderによるひな形コードの生成 (2) プログラムの作成とコンパイル (3) シミュレータロボットと接続してテスト (4) 実機ロボットと接続してテスト • チュートリアル(RTコンポーネントの作成入門、Raspberry Pi Mouse、Windows) • チュートリアル(RaspberryPiマウス、Joystick操作) - 資料: 261015-03.pdf - シミュレータ等: RTM_Tutorial.zip
-------------	---

10月16日 (金)	
10:00 -11:45	4. 総合演習 (1) (1) 画像処理コンポーネントの作成 (2) システム構築とテスト • チュートリアル(画像処理実習) - 資料: 261016-04.pdf
11:45 -12:30	昼食
12:30 -16:45	5. 総合演習 (2) (1) SLAMについて (2) 地図作成・ナビゲーション実習 • チュートリアル (MRPT RTC群によるSLAMナビゲーションシステム) - 資料: 261016-05.pdf - SLAM用RTC等(RTCプログラミング演習のRTM_Tutorial.zipと同じ): RTM_Tutorial.zip



はじめに

このページではLiDAR付RaspberryPiマウスを用いてSLAMによるナビゲーションを行います。



SLAMはSimultaneous Localization and Mappingの略で、環境地図作成と自己位置推定を同時に実行することを指し、本チュートリアルでは移動ロボットのナビゲーションに関わるRTコンポーネント群を使用します。

移動ロボットのナビゲーションに関わるRTコンポーネント群はMRPTという自己位置推定、環境地図作成、経路計画などの機能を提供するクロスプラットフォームなライブラリを使用しています。

~~~~~

# SLAMについて

# 移動ロボットの制御

**内界センサ**：車輪の角速度やジャイロ等ロボットの内部の情報を計測するセンサ

**外界センサ**：レーザーや音波、カメラ、GPS等ロボットの外部の情報を取得するセンサ

**SLAM (スラムと読む、Simultaneous Localization and Mapping)**：外界センサを用いて、ロボット周辺のマップを作成しながら同時に自己位置も推定する技術。センサには、レーザー（2次元、3次元）やカメラ、音波などが用いられる。相対位置を比較的安定的に推定できる。

**自己位置推定 (Localization, ローカリゼーション、ローカライゼーション)**：種々のセンサを利用し、ロボットの現在の位置を推定する技術。移動ロボットを制御するために最も基本的かつ必要とされる技術。

**パスプランニング (Path Planning)**：与えられたマップ上で、現在位置から目的地までの経路を計画する方法。

**ナビゲーション(Navigation)**：現在位置を推定しながらロボットを目的地まで移動させること。

**デッドレコニング (Dead Reckoning)**：車輪のエンコーダやジャイロ等内界センサのみ利用する自己位置を推定手法。誤差が蓄積するため長時間使用できない。オドメトリ(Odometry)と呼ばれることもある。

# • SLAM (Simultaneous Localization and Mapping)

- 「自己位置推定とマッピングの同時実行」
  - SLAM=「スラム」と読む
- 外界センサを用いて、ロボット周辺のマップを作成しながら同時に自己位置も推定する技術。
- センサには、レーザー（2次元、3次元）やカメラ、音波などが用いられる。
- 相対位置を比較的安定的に推定できる。

外界の情報を利用するので、オドメトリとは異なり誤差が蓄積せず、大域的に自己位置を正確に推定できる

- MRPT : Mobile Robot Programming Toolkit
  - <https://www.mrpt.org/>
  - 本講習会で使用するライブラリ
  - 様々なSLAMアルゴリズム、マップ形式を包含したSLAMライブラリ集
  - ここでは、2Dレーザセンサを利用したアルゴリズムを仕様



- LiDAR
  - レーザー測距センサ
  - レーザ光を発し対象物までの距離を測定するセンサ
  - RasPiMouseに搭載しているものは、全方向、2次元的に計測可能
- ※小さなチャック付き袋に、LiDAR固定用ねじが2本入っていますので走らせるとき固定してください。
- 無くさないよう注意してください

## LiDAR

=Laser Imaging Detection and Ranging



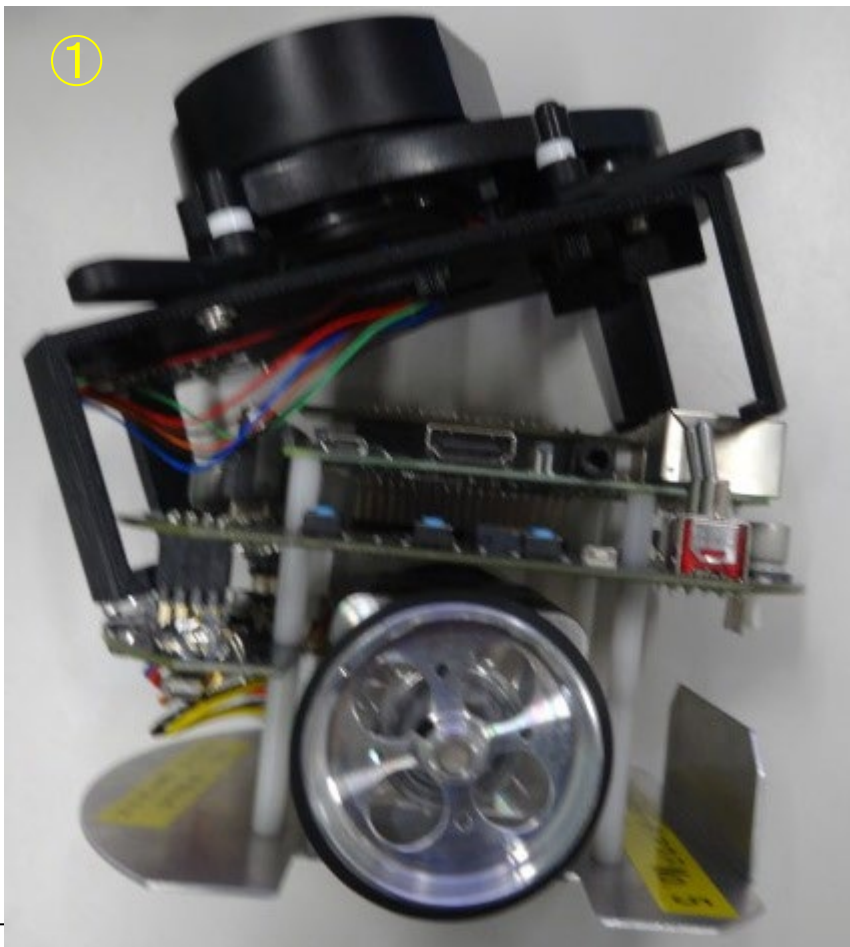
# LiDARの取り付け

- LiDARマウントには2種類ある
  - どちらかを貸し出しているので、これから説明する手順通りに取り付ける
    - 右側の4本のツメで固定するタイプ(マルチLiDARマウント)の取り付け手順を説明する。



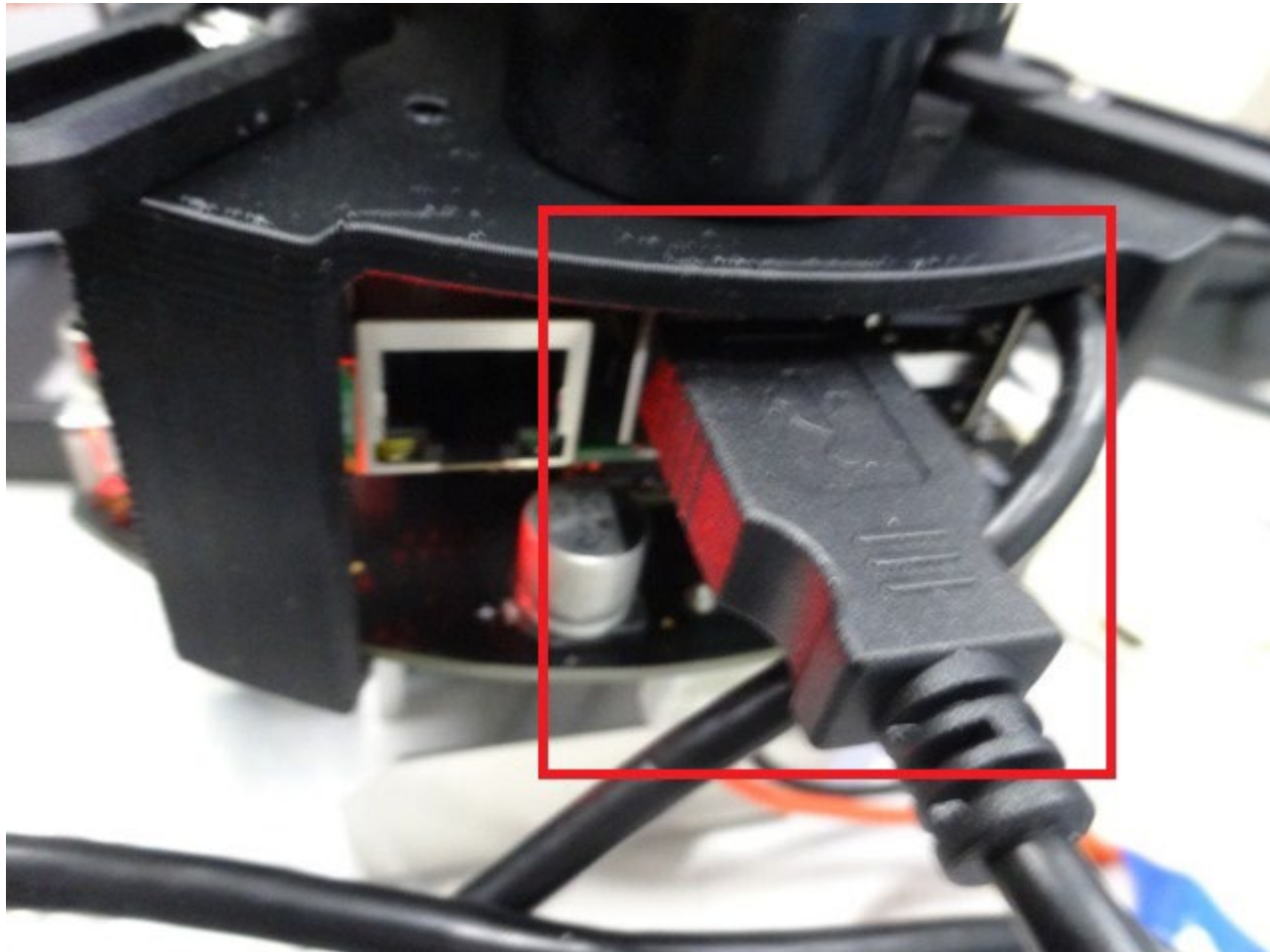
# LiDARの取り付け

- ①のように前方のツメ(長い方のツメ)を引っかける
- 次に②のように後方のツメも引っかける



# LiDARの取り付け

- Raspberry PiのUSBポートに接続する



## 1. 地図作成 (Mapper)

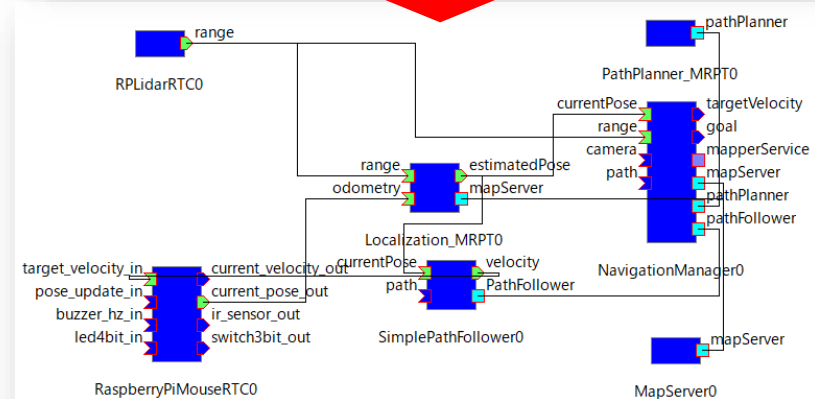
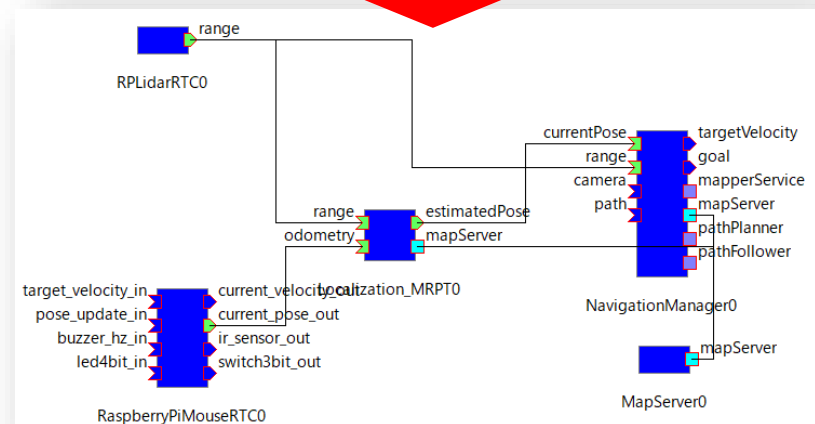
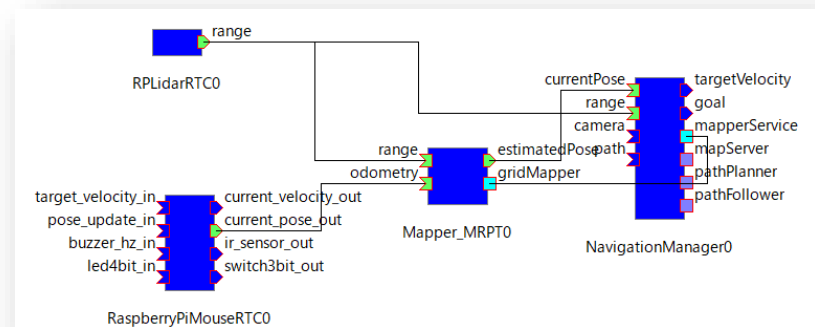
地図作成を行う。地図作成に必要な右のコンポーネントを起動し接続、ジョイスティックでRasPiMouseを移動させ、環境をLiDARで計測し、地図を作成する。

## 2. 自己位置推定 (Localization)

地図の作成が完了したら、システム構成を変更し、自己位置推定を自動的に行う構成に変更する。

## 3. 経路生成・移動 (Navigation)

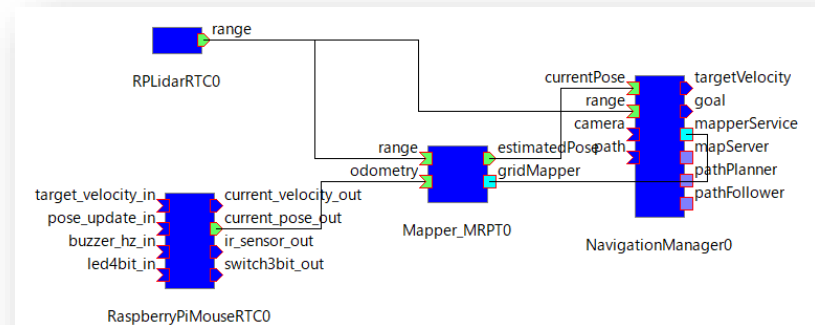
経路生成と移動を行うコンポーネントを新たに起動し、地図内に適当な目的位置を指定、現在の場所から、目的位置までの経路を生成、移動ロボットを移動させる。



# 地図作成実習

## 1. 地図作成 (Mapper)

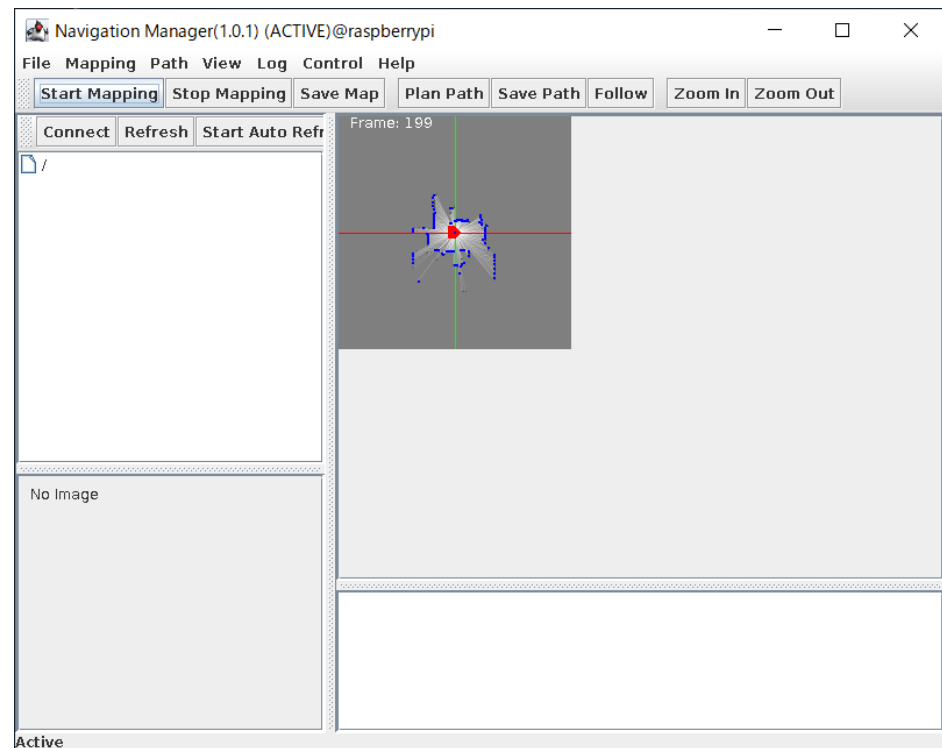
地図作成を行う。地図作成に必要な右のコンポーネントを起動し接続、ジョイスティックでRasPiMouseを移動させる。



右図のように、LiDARで検出したロボット周辺の障害物の地図が現れる。RasPiMouseを移動させ、より広範囲の地図を作成する。



ある程度まで地図ができたなら、地図を保存する。地図はyamlファイルとpngファイルから構成される。このファイルは後述のMapServer RTCの入力となる。

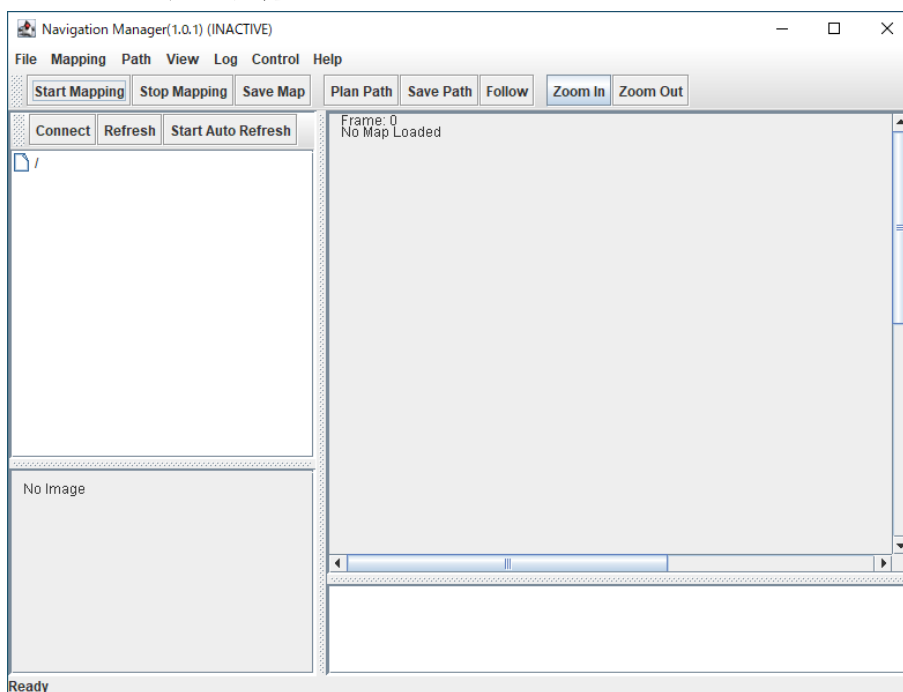


# 作業内容



# RTCの起動、ポートの接続

- NavigationManagerの起動
  - RTM\_Tutorial内の**Navigation**フォルダ
  - 以下のバッチファイル(シェルスクリプト)を実行する
    - NavigationManager.bat
  - 以下のGUIが起動する。

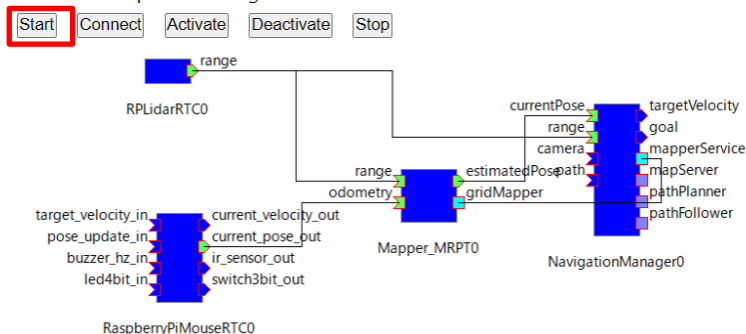


# RTCの起動、ポートの接続

- Mapperシステムの起動
  - WEBブラウザでMapperのStartボタンを押す

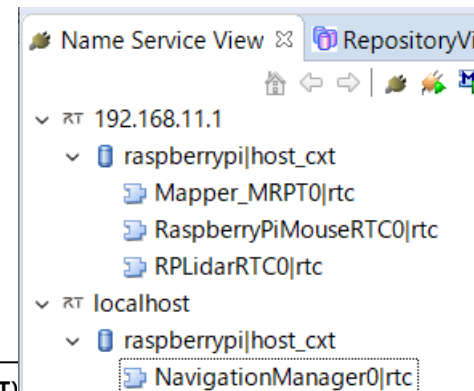
## Mapper

Create a map for a navigation.



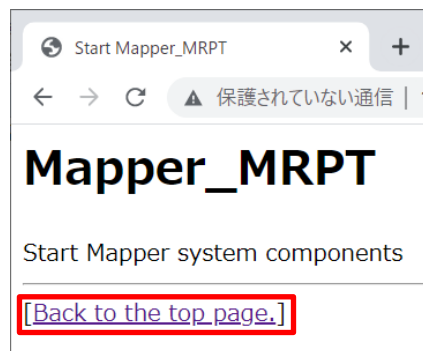
- ネームサーバーに以下のRTCが登録されている状態になる

- Mapper\_MRPT0
- RaspberryPiMouseRTC0
- RPLiderRTC0 or RobotisLDSensor0
- NavigationManager0



# RTCの起動、ポートの接続

- Mapperシステムの起動
  - RTCの起動を確認したら「Back to the top page.」を押す

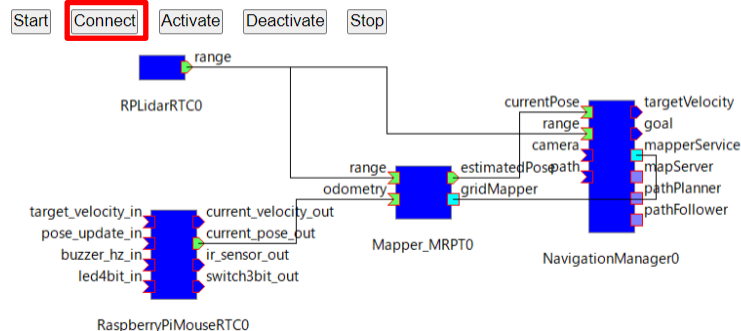


## – ポートの接続

- WEBブラウザからMapperのConnectボタンを押す。

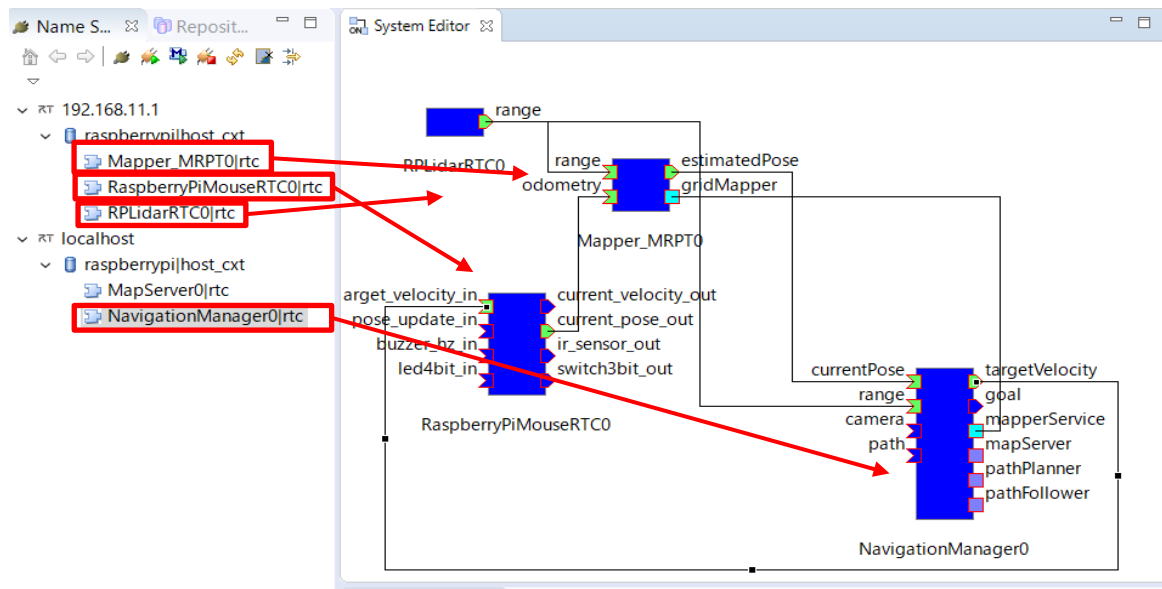
### Mapper

Create a map for a navigation.



# RTCの起動、ポートの接続

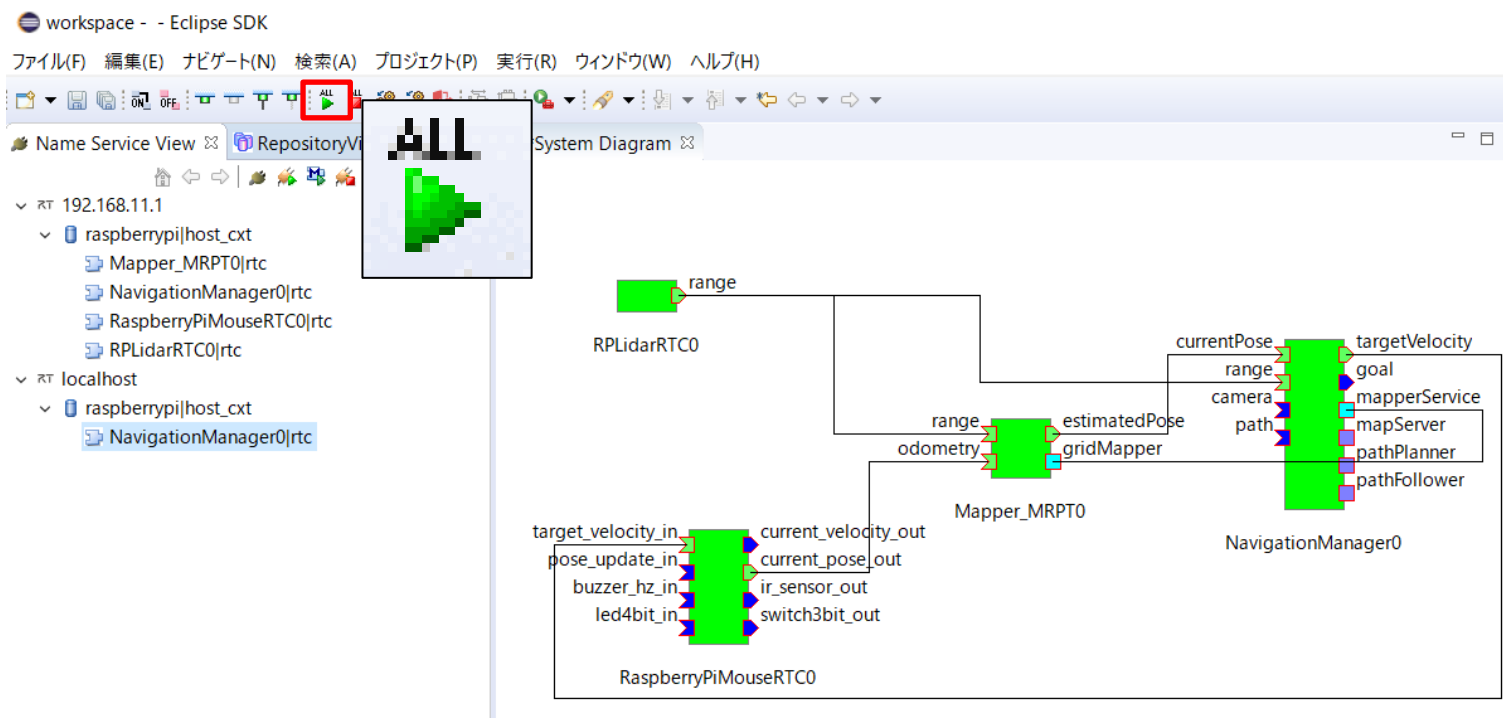
- NavigationManagerのポートを接続する



| コンポーネント名           | ポート名           | コンポーネント名                       | ポート名               |
|--------------------|----------------|--------------------------------|--------------------|
| NavigationManager0 | mapperService  | Mapper_MRPT0                   | gridMapper         |
| NavigationManager0 | currentPose    | Mapper_MRPT0                   | estimatedPose      |
| NavigationManager0 | targetVelocity | RaspberryPiMouseRTC0           | target_velocity_in |
| NavigationManager0 | range          | RobotisLDSensor0 (RPLiderRTC0) | range              |

# システムの実行

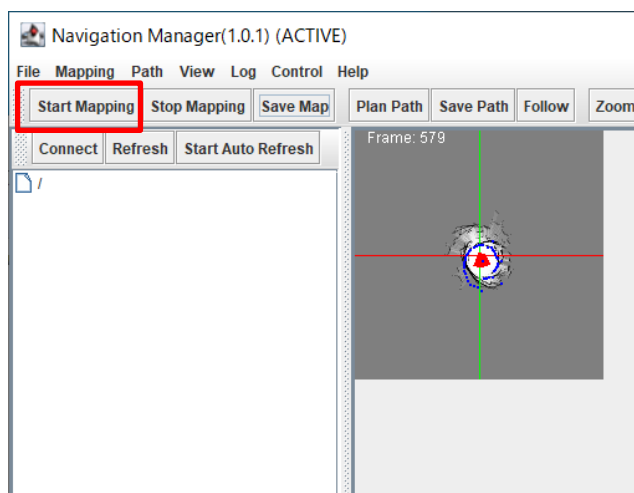
- RTCをアクティブ化する



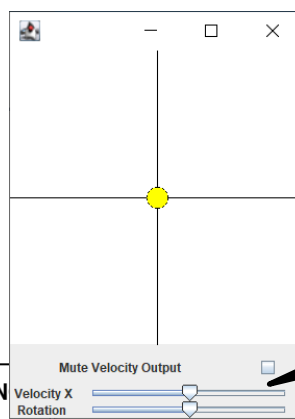
- Raspberry Piマウスのモーター電源をオンにする
  - 2つのトグルスイッチのうちの外側のスイッチ

# システムの実行

- NavigationManagerの**Start Mapping**ボタンを押す。



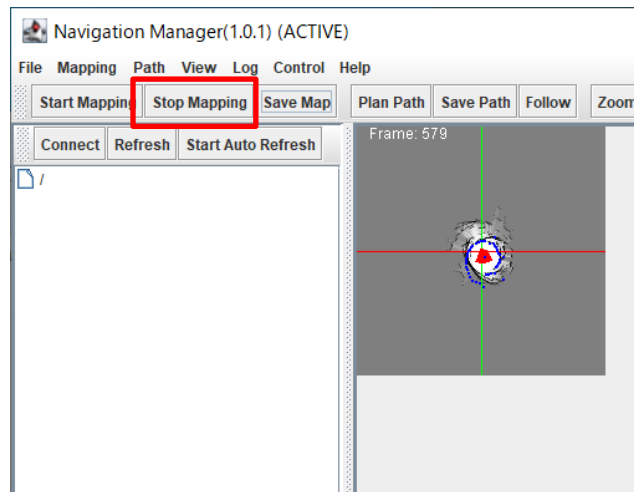
- ジョイスティックパネルを操作することでRaspberry Piマウスが移動し地図生成を行う。



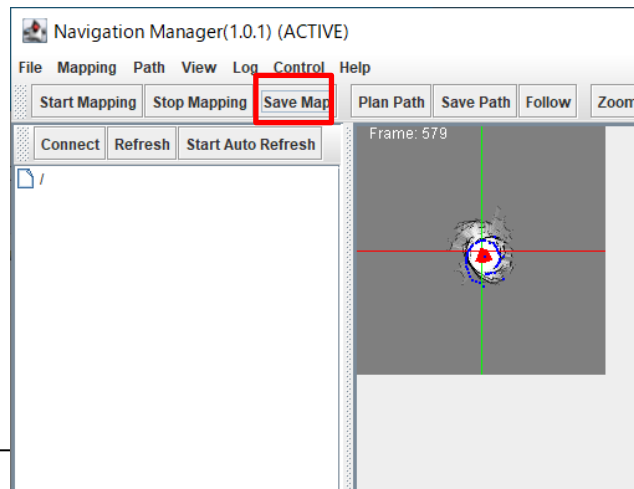
かなりゆっくり動かさないと上手く  
地図作成できません。  
Velocity Xを下げる(スライダーを左  
に動かす)ことで調整してください。

# システムの終了

- NavigationManagerの**Stop Mapping**ボタンを押す。

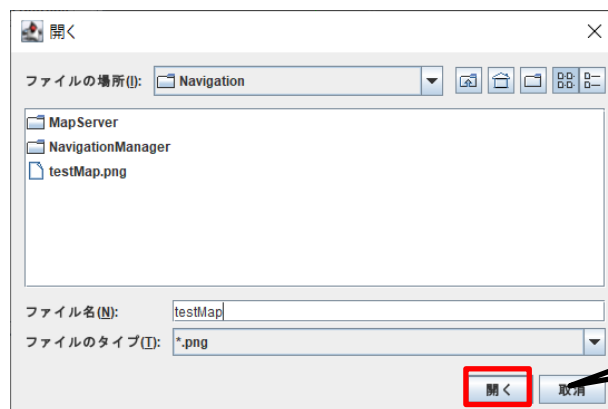


- NavigationManagerの**Save Map**ボタンを押す。



# システムの終了

- 地図情報の保存先としてNavigationフォルダにファイル名testMapで保存する。



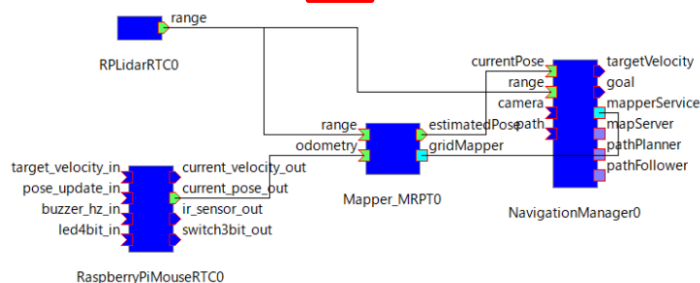
開くボタンを押すと  
保存される。

- WEBブラウザからMapperのStopボタンを押してRTCを終了する

## Mapper

Create a map for a navigation.

Start Connect Activate Deactivate Stop



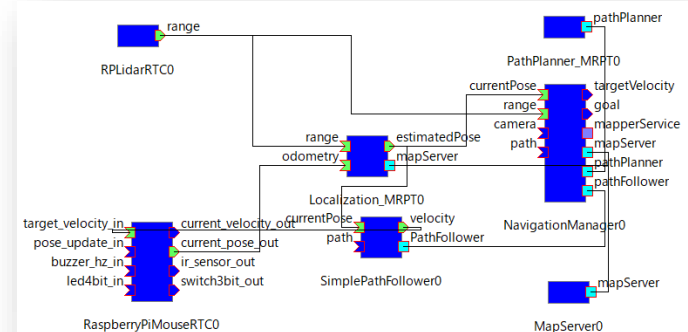
# ナビゲーション実習

## 2. 自己位置推定 (Localization)

地図の作成が完了したら、システム構成を変更し、自己位置推定を自動的に行う構成に変更する。



MapServerに作成した地図（ファイル名）をConfigurationパラメータで与え、自己位置推定に利用する。

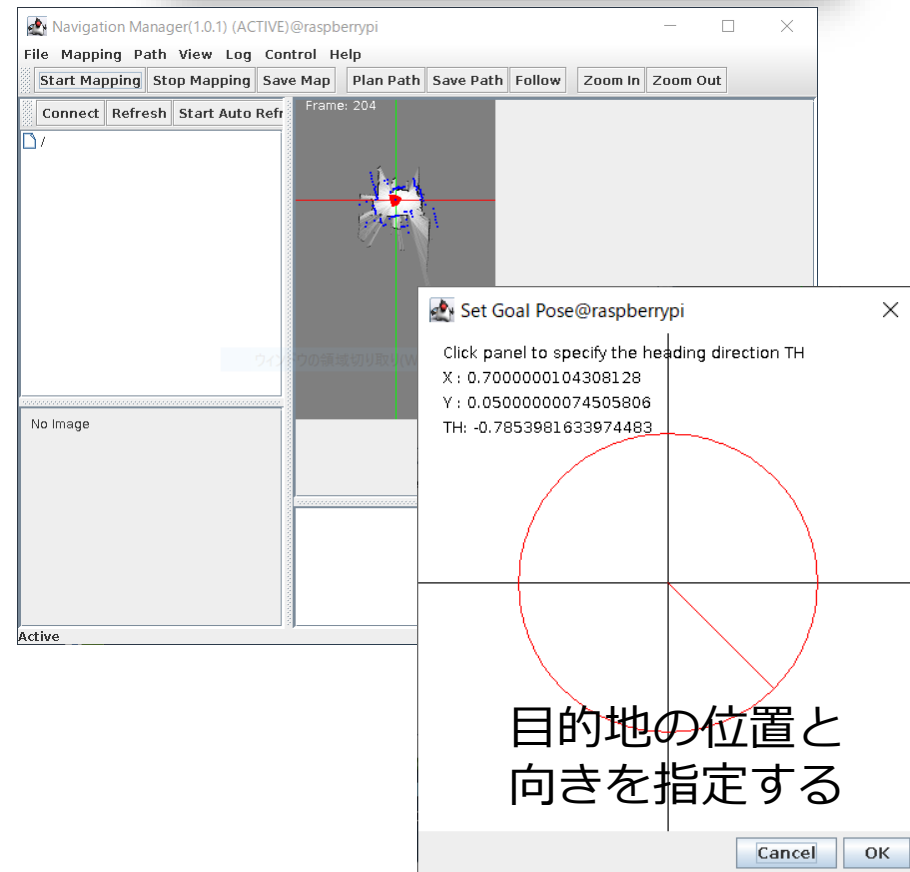


## 3. 経路生成・移動 (Navigation)

経路生成と移動を行うコンポーネントを新たに起動し、地図内に適当な目的位置を指定する（右図）。



現在の場所から、目的位置までの経路を生成、移動ロボットが自律的に移動する。



# 作業内容

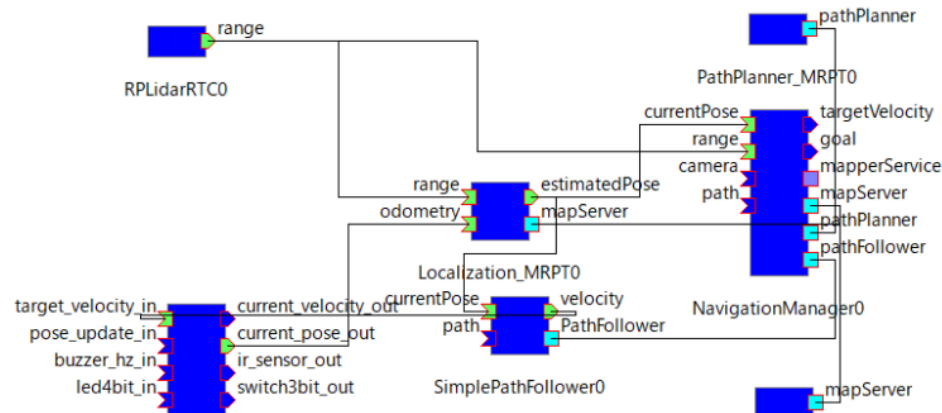


# RTCの起動、ポートの接続

- MapServerの起動(Navigationフォルダ)
  - 以下のバッチファイル(シェルスクリプト)を実行する
    - MapServer.bat
- PathPlanシステムの起動
  - WEBブラウザでPathPlanのStart、Connectボタンを押す

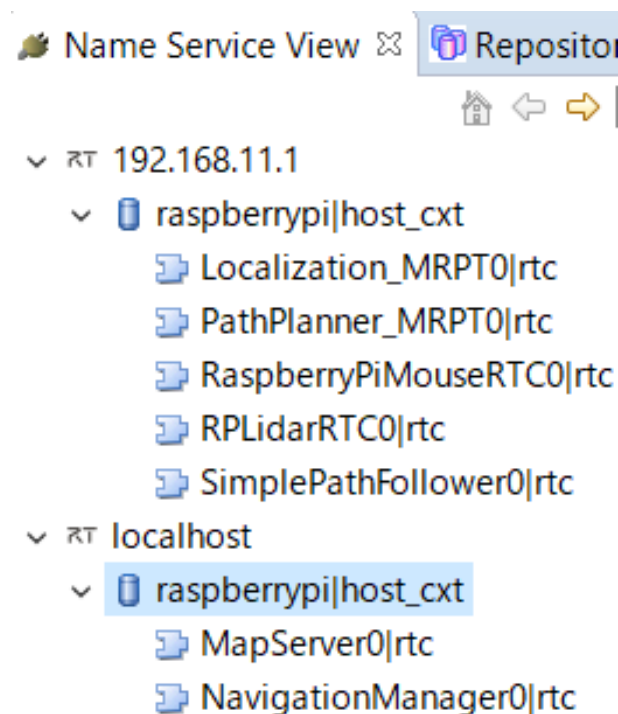
## PathPlan

RasPiMouse navigation by using a path planner using MRPT.



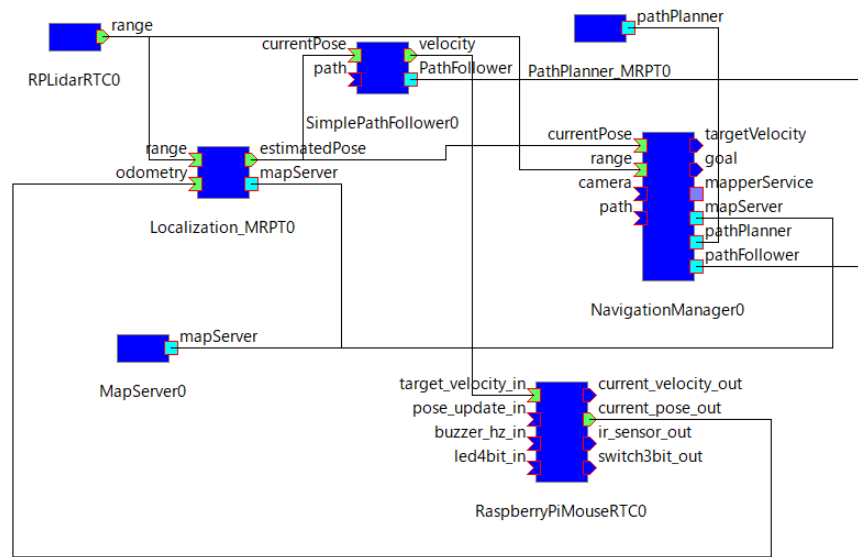
# RTCの起動、ポートの接続

- PathPlanシステムの起動
  - ネームサーバーに以下のRTCが登録されている状態になる
    - Localization\_MRPT0
    - PatgPanner\_MRPT0
    - RaspberryPiMouseRTC0
    - RPLiderRTC0 or RobotisLDSensor0
    - SimplePathFollower0
    - MapServer0
    - NavigationManager0



# RTCの起動、ポートの接続

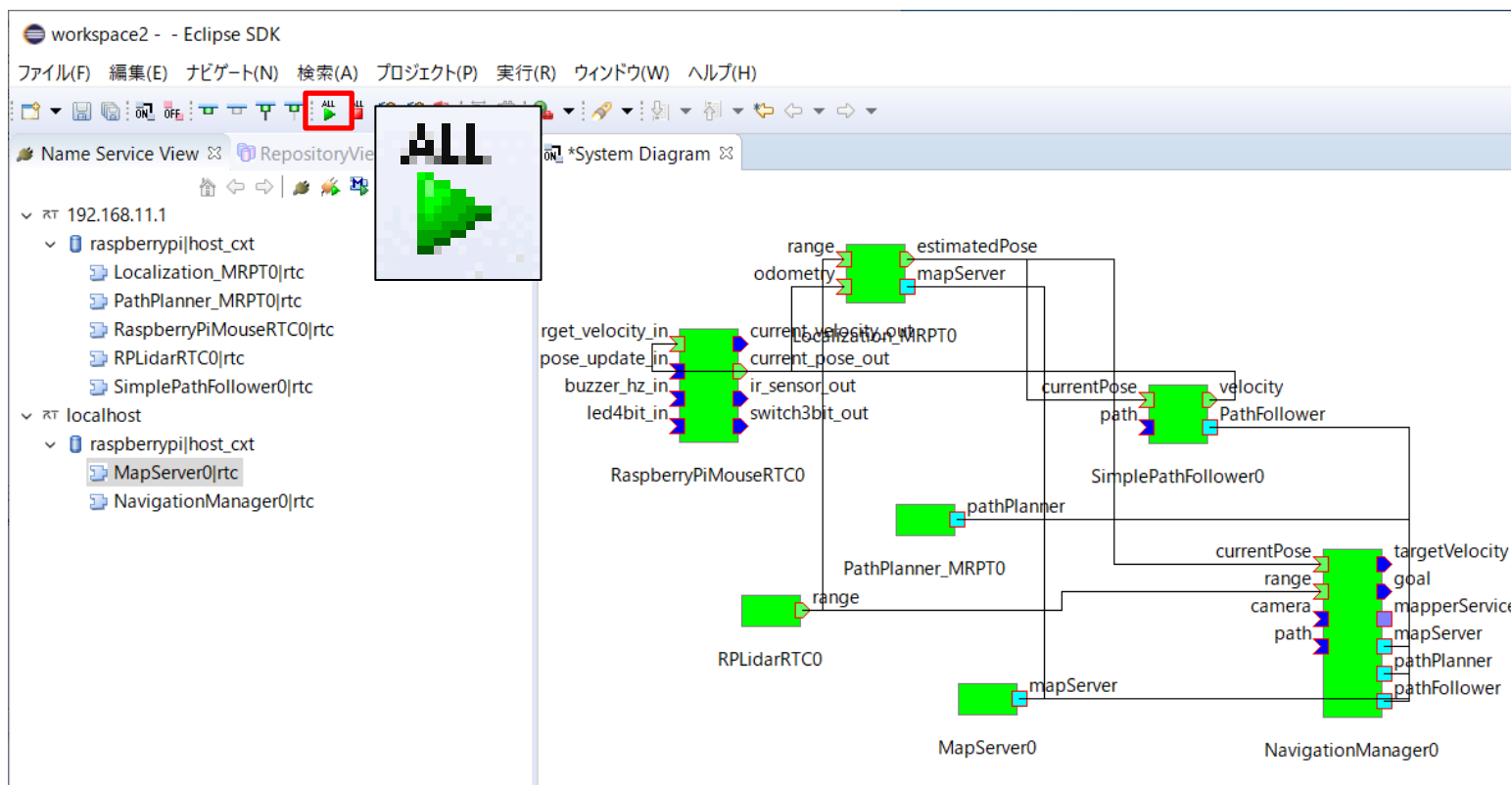
- NavigationManager、MapServerのポートを接続する



| コンポーネント名           | ポート名         | コンポーネント名                          | ポート名          |
|--------------------|--------------|-----------------------------------|---------------|
| NavigationManager0 | mapServer    | MapServer0                        | mapServer     |
| NavigationManager0 | pathPlanner  | PathPlanner_MRPT0                 | pathPlanner   |
| NavigationManager0 | pathFollower | SimplePathFollower0               | PathFollower  |
| NavigationManager0 | currentPose  | Localization_MRPT0                | estimatedPose |
| NavigationManager0 | range        | RobotisLDSensor0<br>(RPLiderRTC0) | range         |
| MapServer0         | mapServer    | Localization_MRPT                 | mapServer     |

# システムの実行

- RTCをアクティブ化する

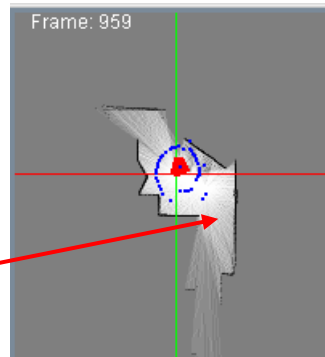


- Raspberry Piマウスのモーター電源をオンにする

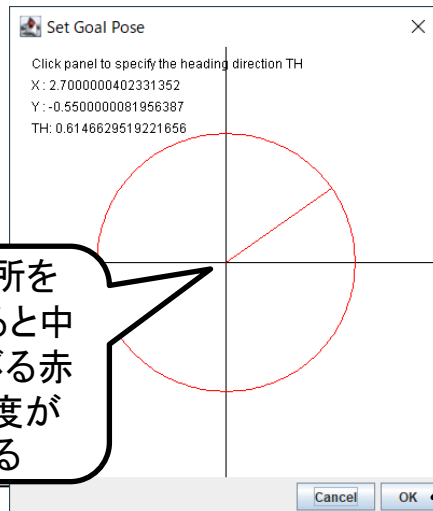
# システムの実行

- Raspberry Piマウスの目標位置、目標姿勢角を設定する
  - NavigationManagerの地図上をクリックする

地図の白い範囲(障害物のない範囲)のどこかをクリックする

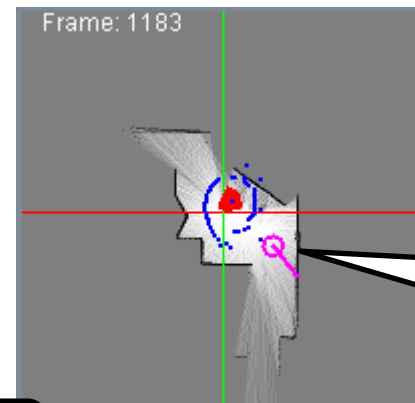


- 以下の画面で目標角度を設定する。



適当な場所をクリックすると中心から延びる赤い線の角度が変化する

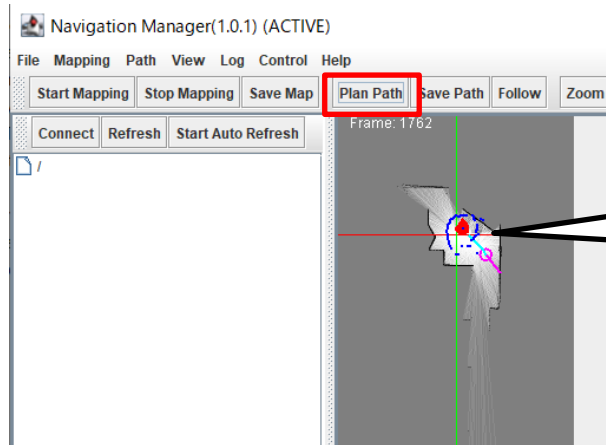
角度を設定したらOKをクリックする



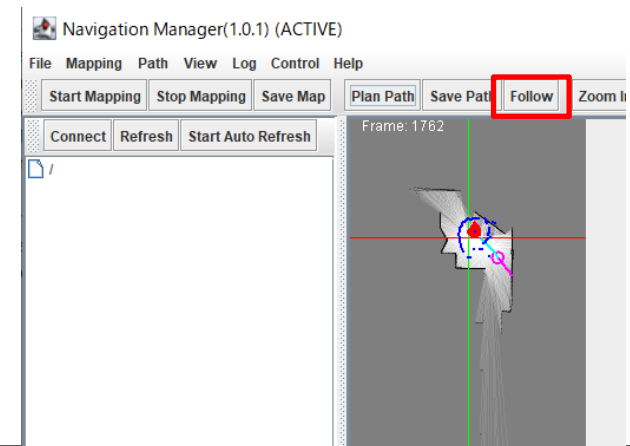
地図上に目標位置が表示される

# システムの実行

- 経路を計画する
  - NavigationManagerのPlan Pathボタンをクリックする

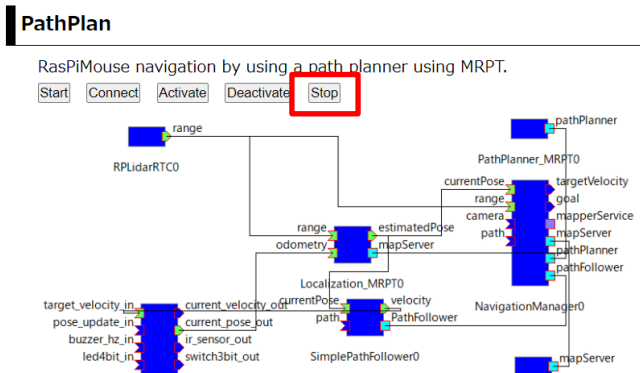


- 経路追従を行う
  - NavigationManagerのFollowボタンをクリックする
  - Raspberry Piマウスが目的地に向かって移動する



# システムの終了

- WEBブラウザからPathPlanのStopボタンを押してRTCを終了する



- モーター電源のスイッチをオフにする
- Raspberry Piをシャットダウンする(真ん中のボタン長押し)
- Raspberry Pi電源のスイッチをオフにする
- LiDARを取り外して元の状態に戻す

# おわりに

- 用意してある迷路や、教室全体の地図を作成し、ナビゲーションしてみましょう
- TeraTermでRaspberryPiにログインして、中を覗いてみましょう。

2日間お疲れさまでした！！

質問などがあれば、Slackからどうぞ。

以下のリンクをクリックしてSlackに参加。

[https://join.slack.com/t/openrtm-users/shared\\_invite/zt-2utl8xxrp-94XzgNC1BMrep4vE\\_UWN2Q](https://join.slack.com/t/openrtm-users/shared_invite/zt-2utl8xxrp-94XzgNC1BMrep4vE_UWN2Q)

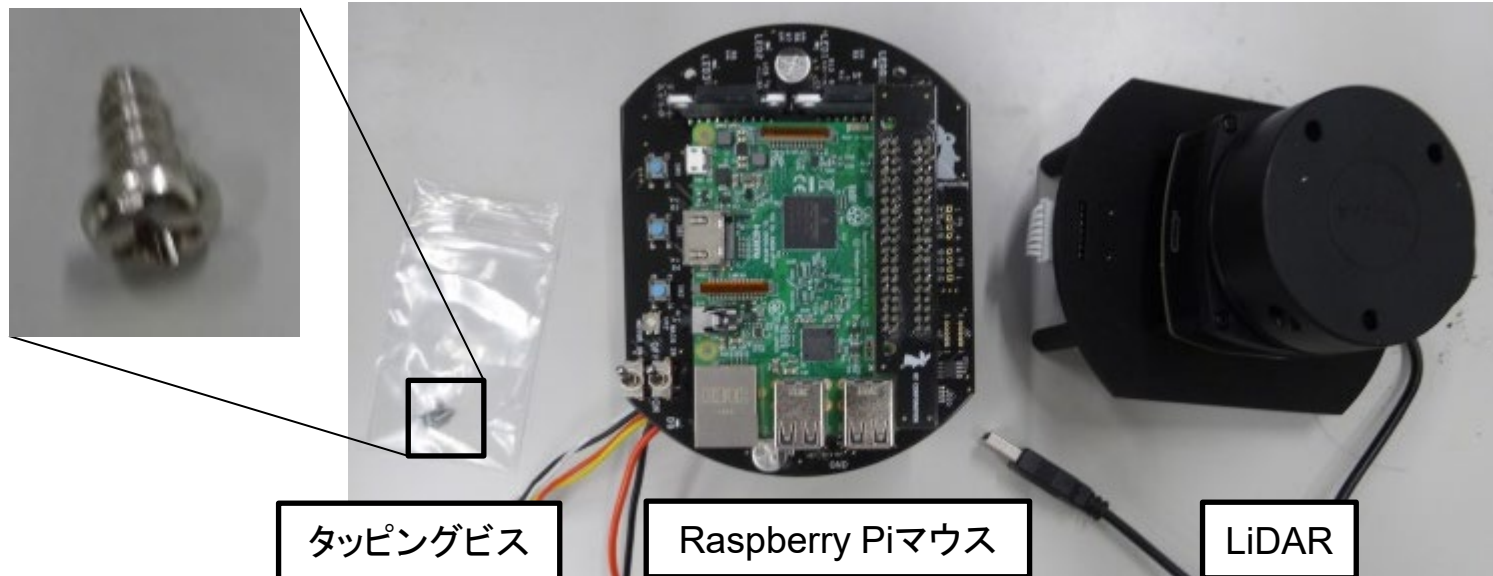
# LiDARの取り付け

- LiDARマウントには2種類ある
  - どちらかを貸し出しているので、これから説明する手順通りに取り付ける
    - 左側の2本の脚が伸びているタイプの手順について説明する



# LiDARの取り付け

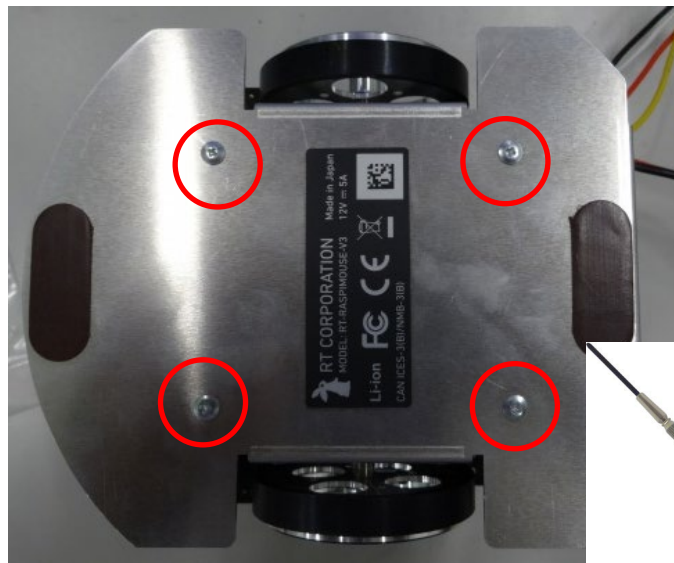
- 以下の部品を組み立てます。



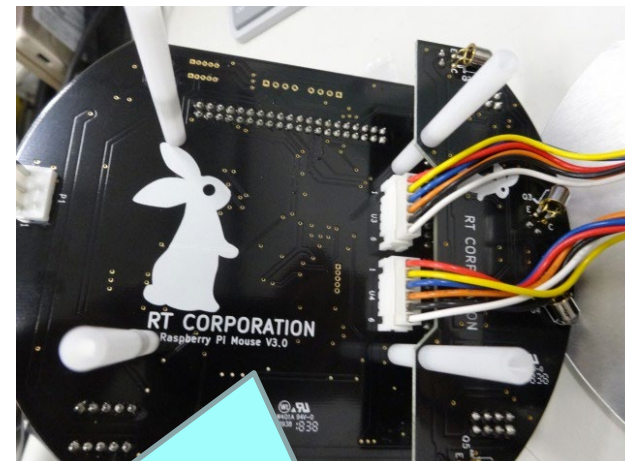
- 一旦Raspberry Piをシャットダウンしてバッテリー(もしくは電源ケーブル)を取り外した状態で作業してください
  - Raspberry Piのシャットダウンは真ん中のボタンを3秒ほど押して30秒ほど待つ。その後スイッチをオフにする。

# LiDARの取り付け

- Raspberry Piマウス底面のねじをはずす

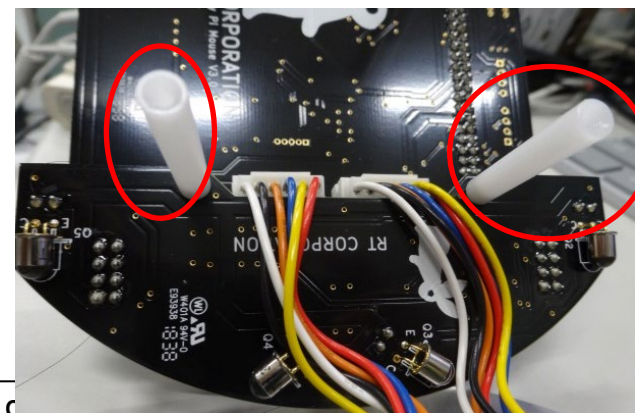


付属のドライバー  
を使用する



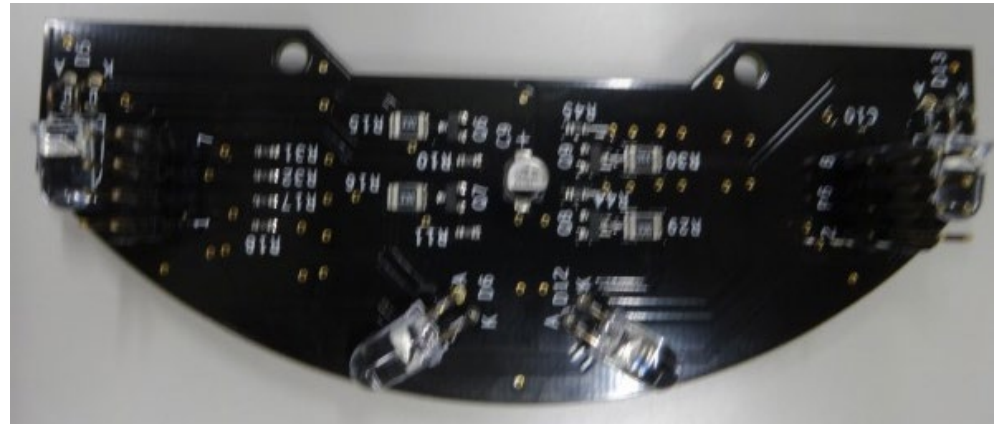
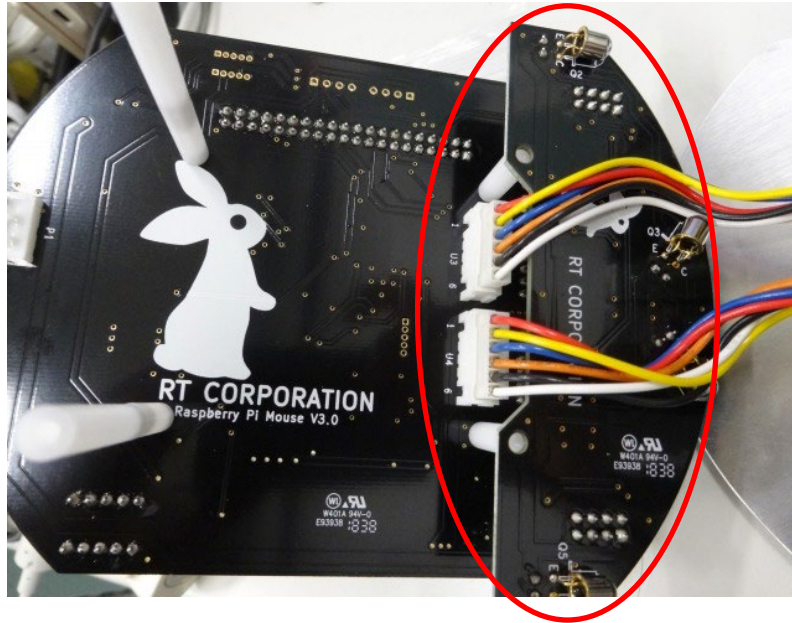
基盤部分をフレームごとずらす

- 前方のスペーサーを2本を回して外す



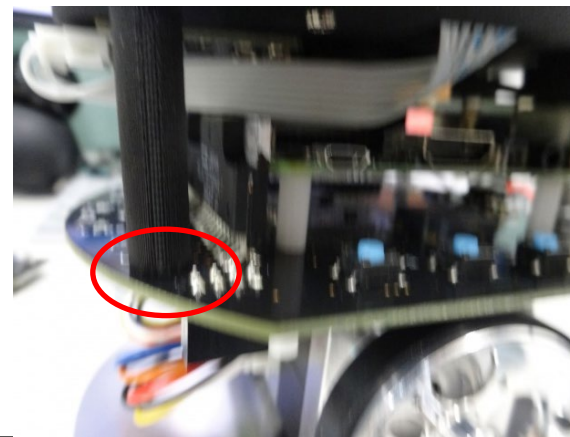
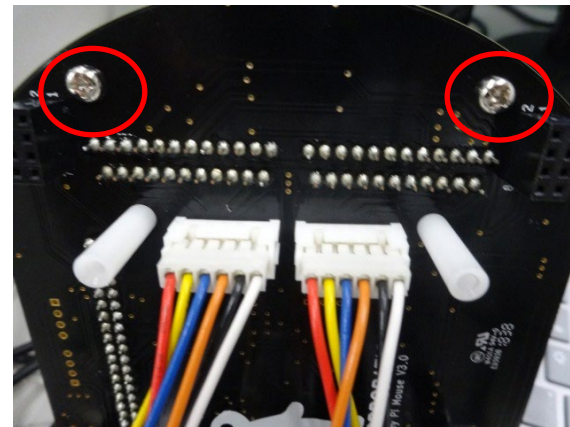
# LiDARの取り付け

- センサ基盤をはずす



# LiDARの取り付け

- LiDARをRaspberry Piマウスの上部分に載せる
- LiDARとRaspberryPiマウスをビスで固定する



# LiDARの取り付け

- Raspberry PiのUSBポートに接続する



# LiDARの取り付け

- 外したパーツを戻せば完成
- ※危ないのでLiDARの土台部分を持たないようにしてください。



- LiDARの取り付け作業が完了したら、RaspberryPiの電源スイッチをオンにしてアクセスポイントに接続してください。

# LiDARの取り付け

- なべタッピングネジ3-8(マルチLiDARマウント固定用)で取り付ける

上の穴からドライバーを  
差し込んで回す

