

SLAM実習

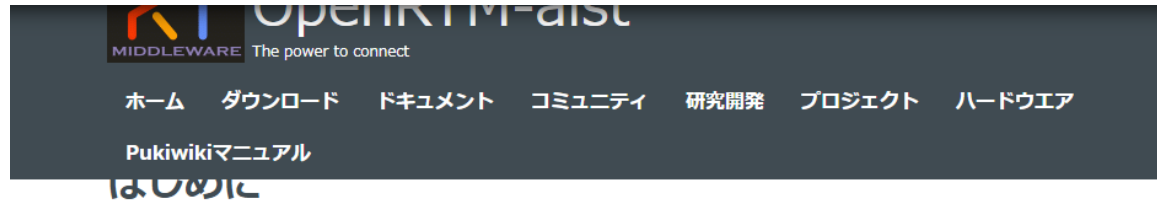
宮本 信彦

国立研究開発法人産業技術総合研究所
インダストリアルCPS研究センター
ソフトウェアプラットフォーム研究チーム

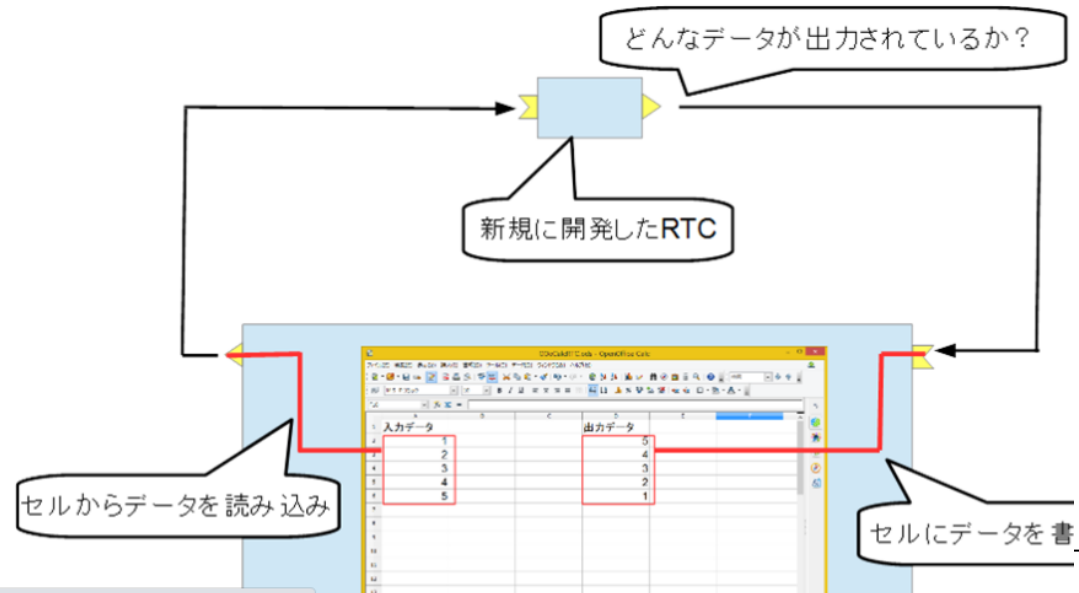


資料

- 「WEBページ」フォルダのHTMLファイルを開く
 - チュートリアル(RTミドルウェア応用実習、Raspberry Pi Mouse) _ OpenRTM-aist.html
- もしくは以下のリンク
 - <https://openrtm.org/openrtm/ja/node/6586>



このページではLibreOffice Calc用RTCによるRTCの動作確認手順について説明します。Calcのセルの値を入力、OutPortの出力した値をセルに表示することで対象RTCの挙動を確認できます。



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とは？

- **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

SLAMライブラリ

- 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搭載版RasPiMouse

- 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反るセンサ
 - 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

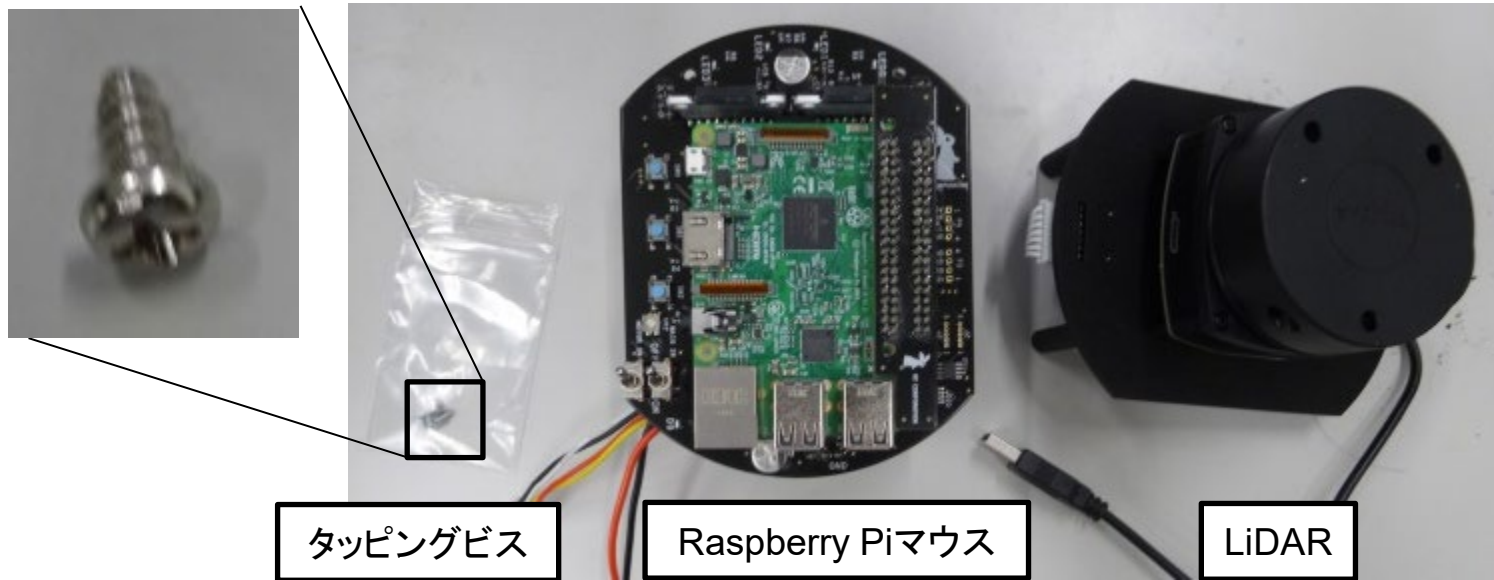
LiDAR

=Laser Imaging Detection and Ranging



LiDARの取り付け

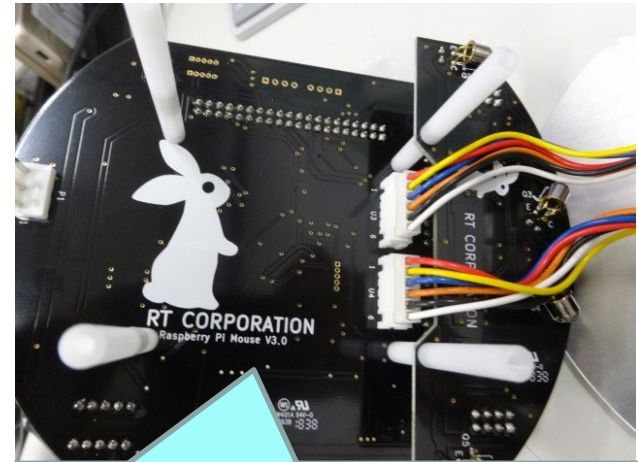
- 以下の部品を組み立てます。



- 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

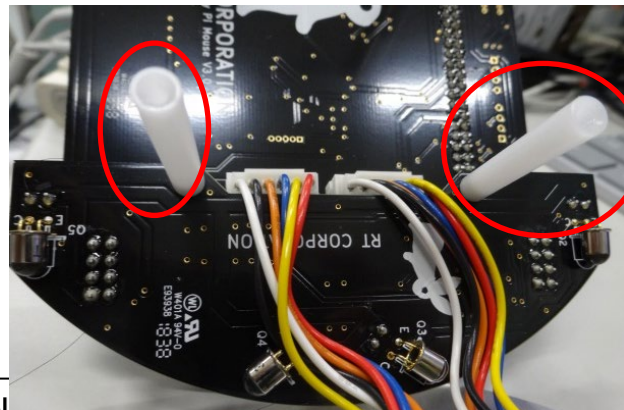
LiDARの取り付け

- Raspberry Piマウス底面のねじをはずす



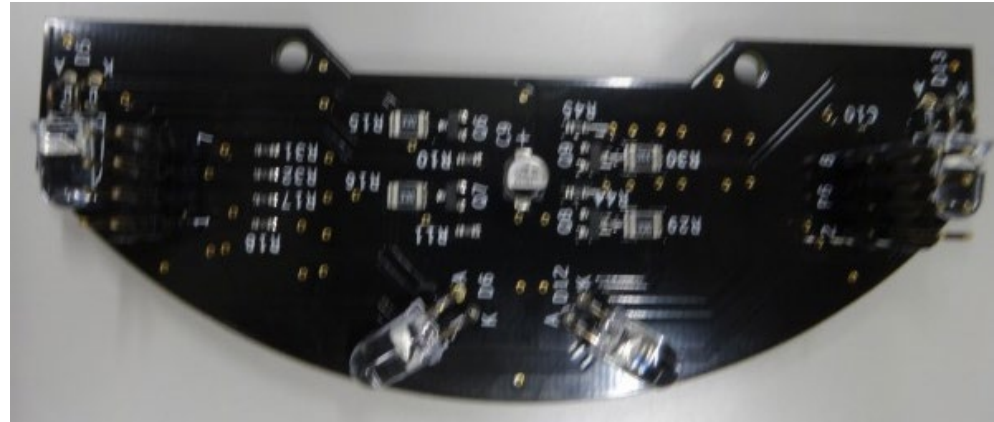
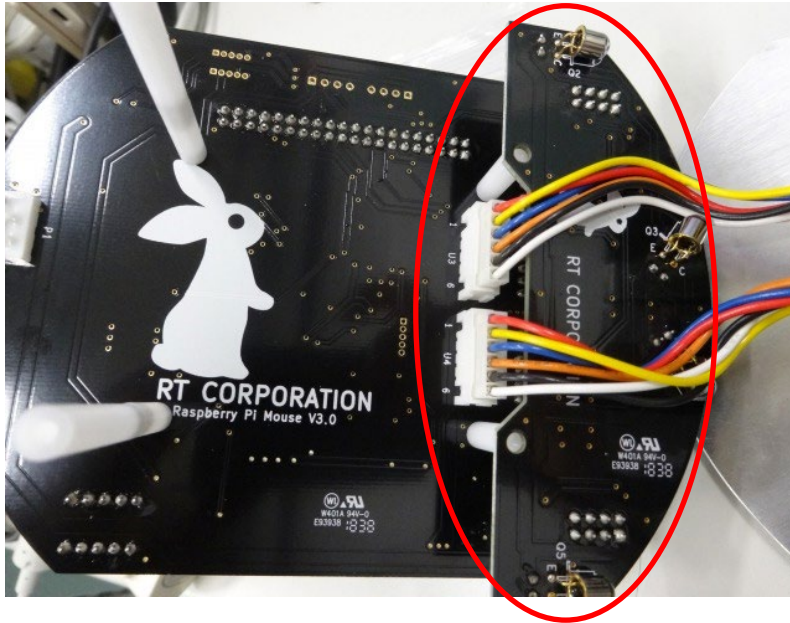
基盤部分をフレームごとずらす

- スペーサーを回して外す



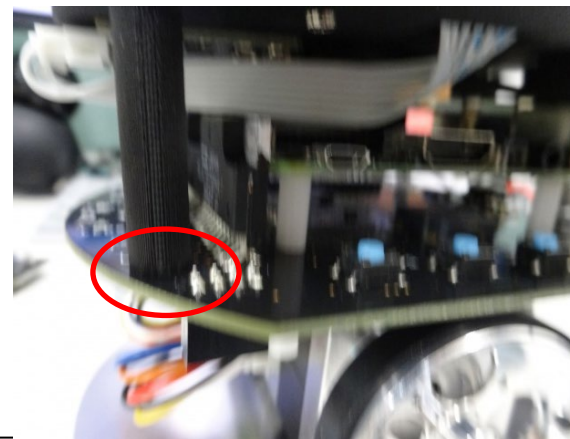
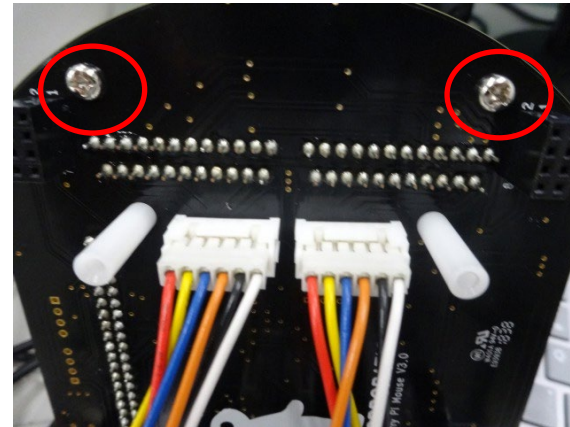
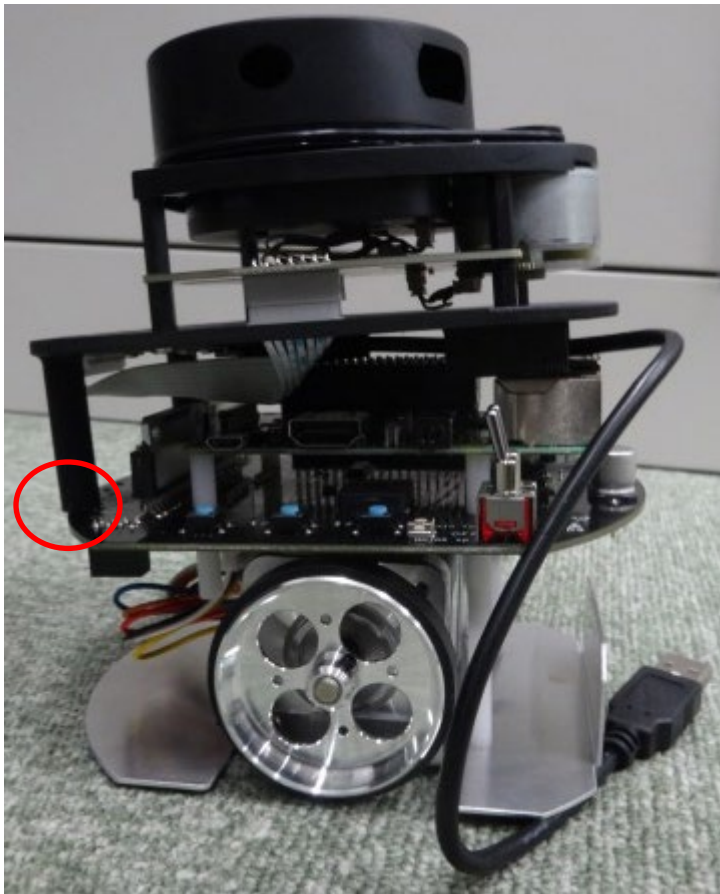
LiDARの取り付け

- 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



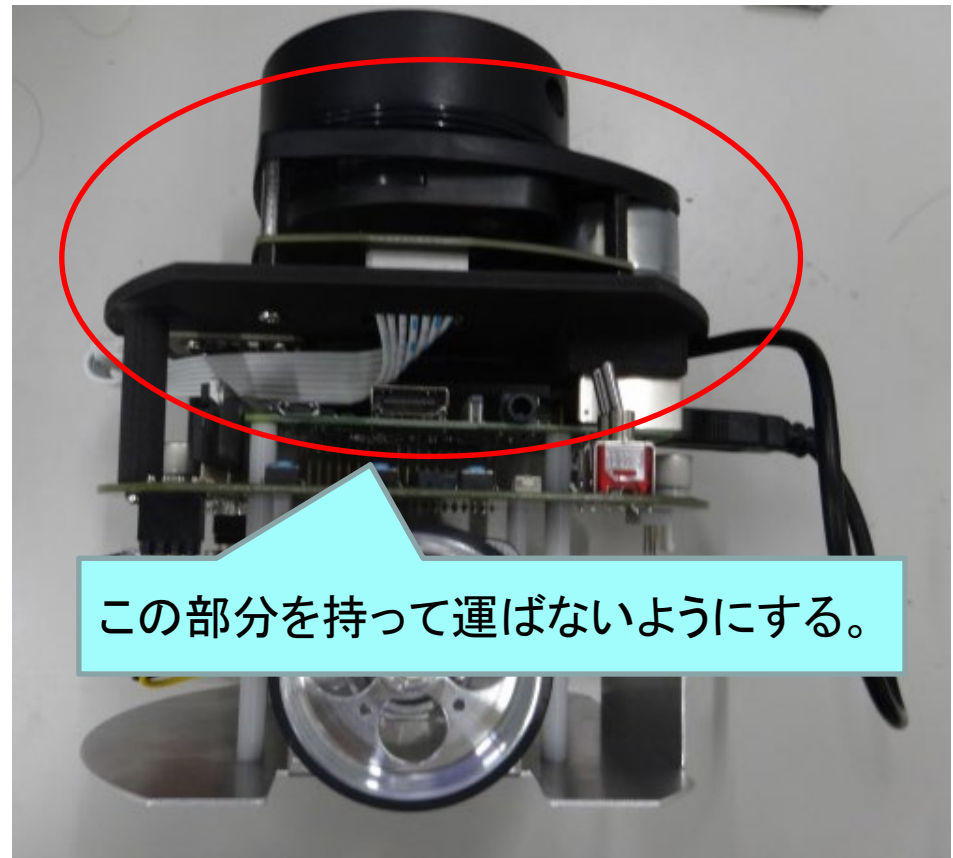
LiDARの取り付け

- 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。



起動済みのRTCの終了

- RaspberryPiMouseRTC、RobotControllerコンポーネントが起動している場合は終了してください。
 - RaspberryPiMouseRTCはWEBブラウザの操作でStopボタンを押す

RaspberryPiMouse with

RT-Components

○ RaspberryPiMouseRTC	[Start]	[Stop]		[Activate]	[Deactivate]
○ RPLidarRTC	[Start]	[Stop]		[Activate]	[Deactivate]
○ Mapper_MRPT	[Start]	[Stop]		[Activate]	[Deactivate]
○ Localization_MRPT	[Start]	[Stop]		[Activate]	[Deactivate]
○ PathPlanner_MRPT	[Start]	[Stop]		[Activate]	[Deactivate]
○ SimplePathFollower	[Start]	[Stop]		[Activate]	[Deactivate]
○ MapServer(Java)	[Start]	[Stop]		[Activate]	[Deactivate]
○ NavigationManager(Java)	[Start]	[Stop]		[Activate]	[Deactivate]

1. 地図作成 (Mapper)

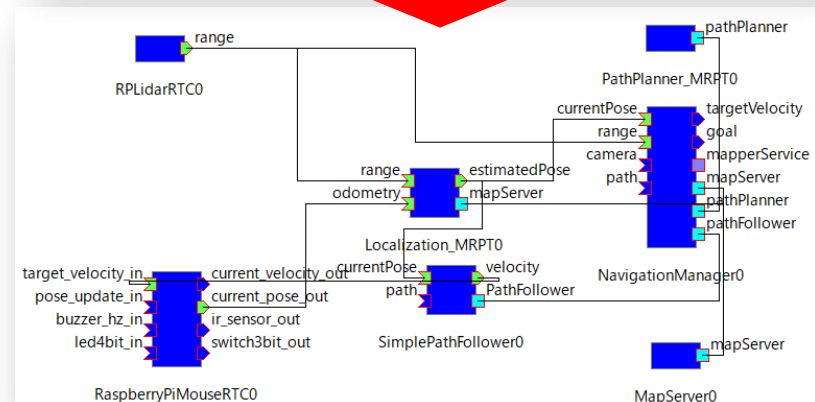
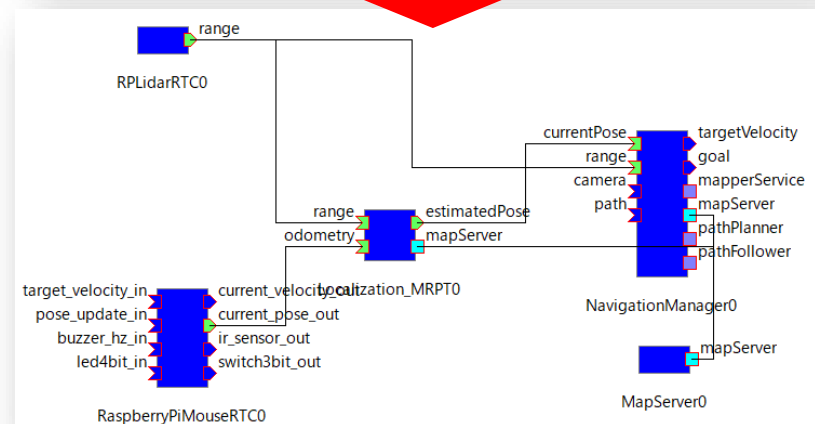
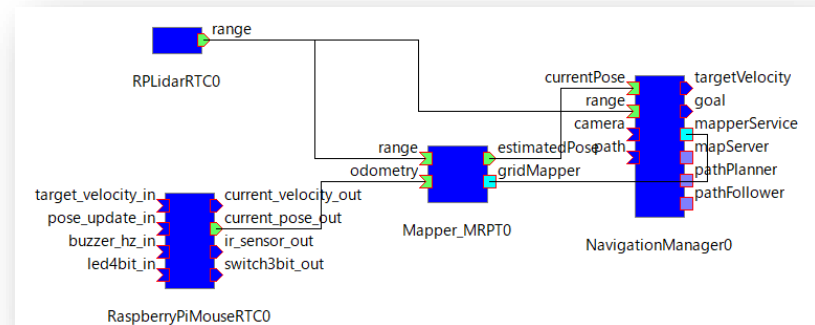
地図作成を行う。地図作成に必要な右のコンポーネントを起動し接続、ジョイスティックでRasPiMouseを移動させ、環境をLiDARで計測し、地図を作成する。

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)

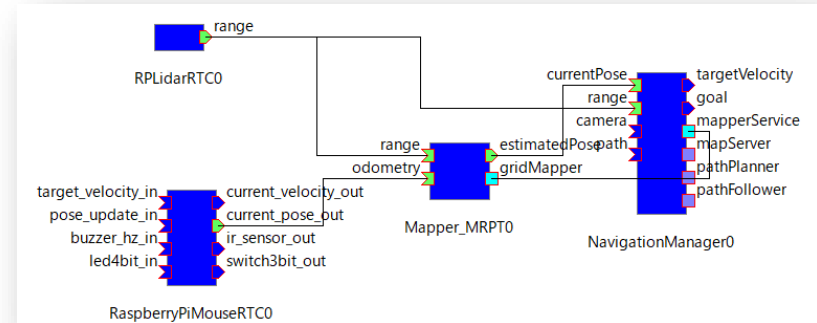
経路生成と移動を行うコンポーネントを新たに起動し、地図内に適当な目的位置を指定、現在の場所から、目的位置までの経路を生成、移動ロボットを移動させる。



地図作成実習

1. 地図作成 (Mapper)

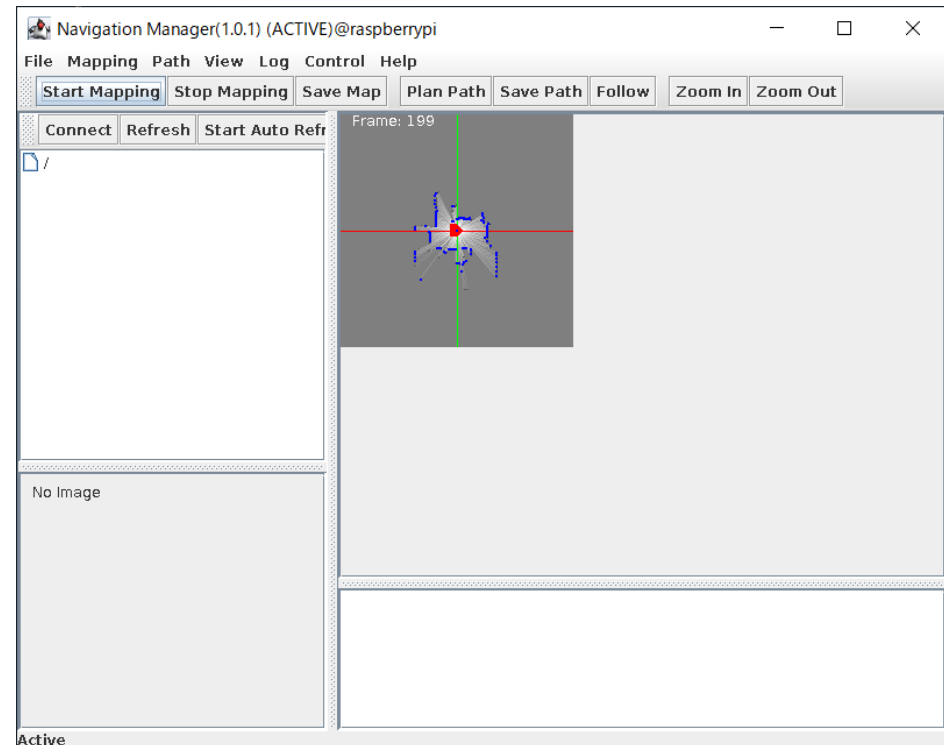
地図作成を行う。地図作成に必要な右のコンポーネントを起動し接続、ジョイスティックでRasPiMouseを移動させる。



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。

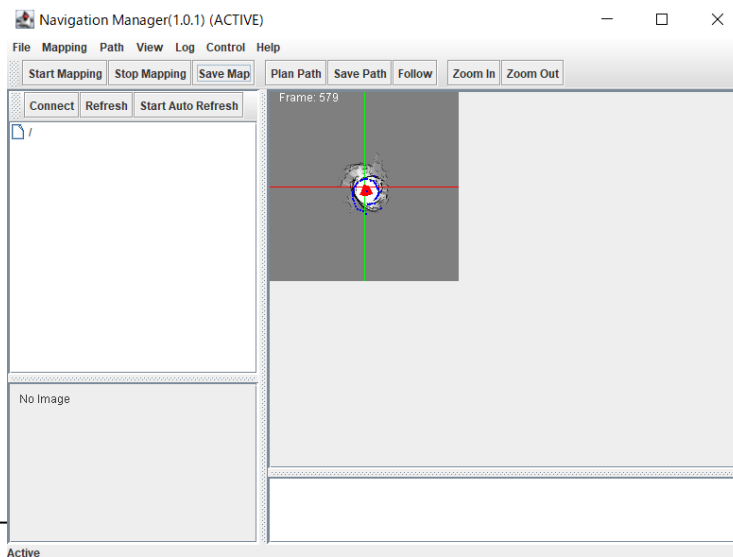


ある程度まで地図ができたなら、地図を保存する。地図はyamlファイルとpngファイルから構成される。このファイルは後述のMapServer RTCの入力となる。



RTCの起動、ポートの接続

- NavigationManagerの起動
 - RTM_Tutorial.zipを展開したフォルダ内の**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(Windows)
 - NavigationManager.sh(Ubuntu)
 - 以下のGUIが起動する。

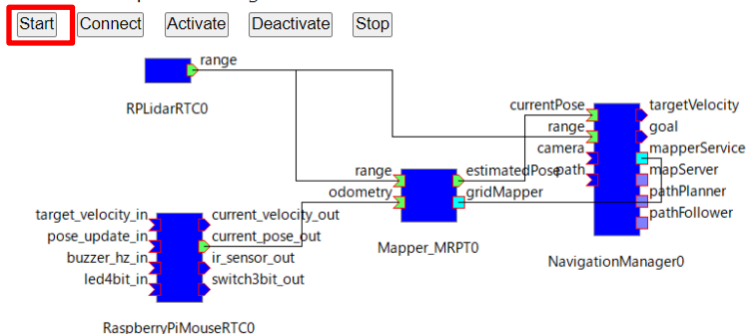


RTCの起動、ポートの接続

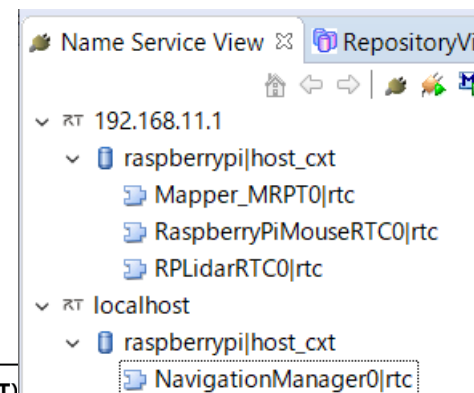
- Mapperシステムの起動
 - WEBブラウザでMapperのStartボタンを押す

Mapper

Create a map for a navigation.

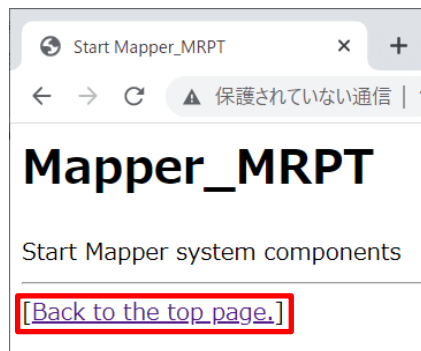


- ネームサーバーに以下のRTCが登録されている状態になる
 - Mapper_MRPT0
 - RaspberryPiMouseRTC0
 - RPLiderRTC0
 - NavigationManager0



RTCの起動、ポートの接続

- Mapperシステムの起動
 - RTCの起動を下記人したら「Back to the top page.」を押す

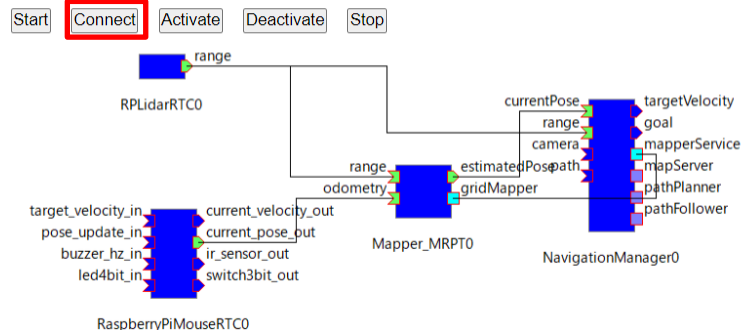


– ポートの接続

- WEBブラウザからMapperのConnectボタンを押す。

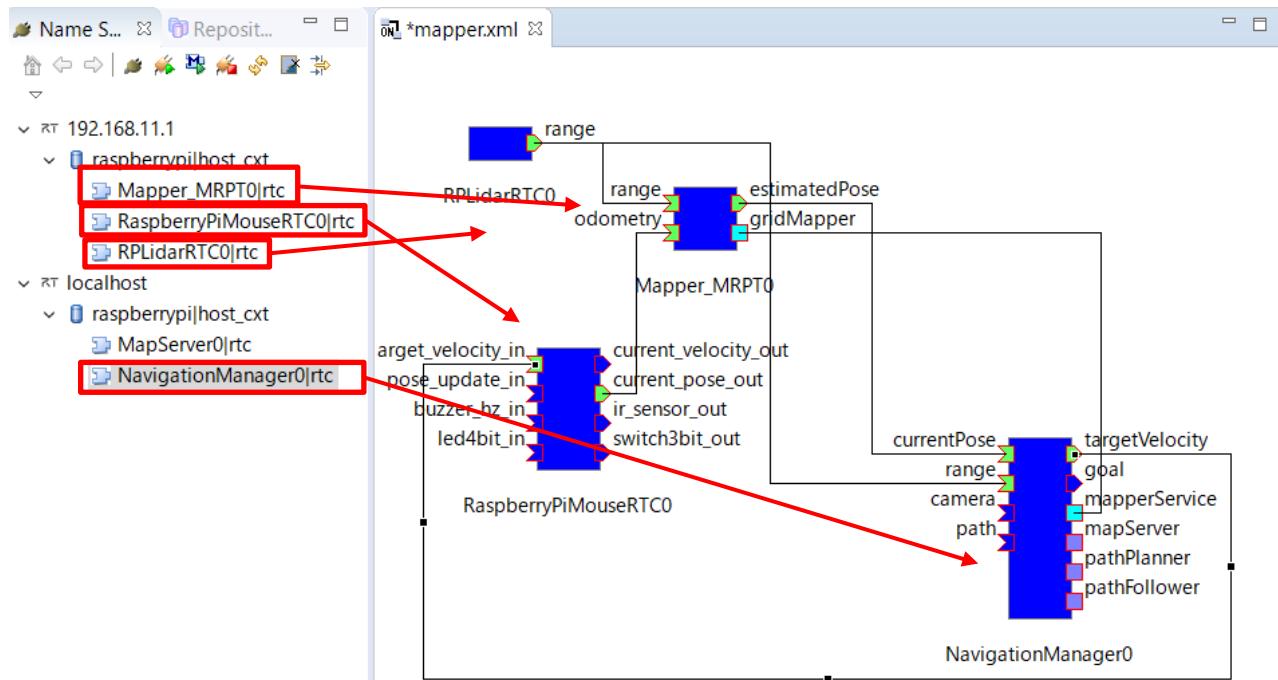
Mapper

Create a map for a navigation.



RTCの起動、ポートの接続

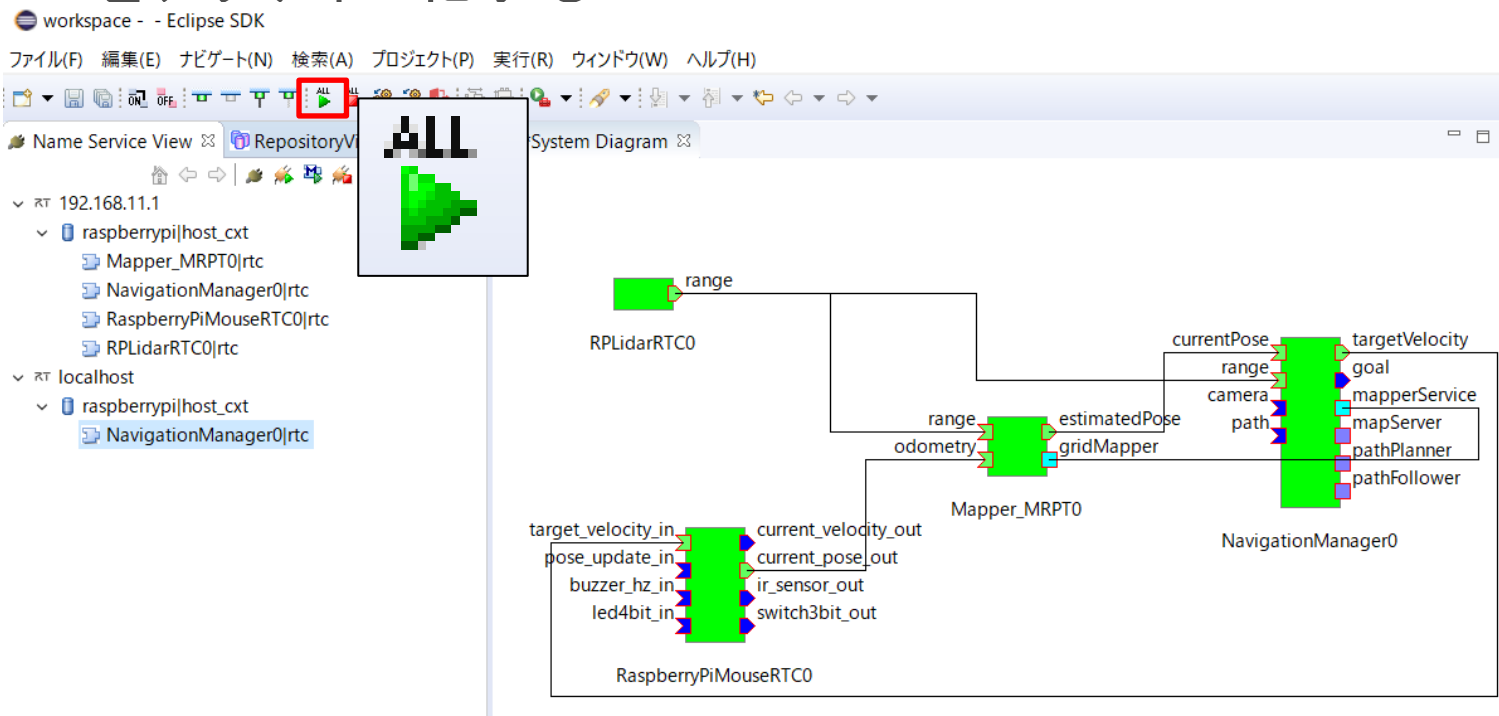
- NavigationManagerのポートを接続する



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	RPLiderRTC0	range

システムの実行

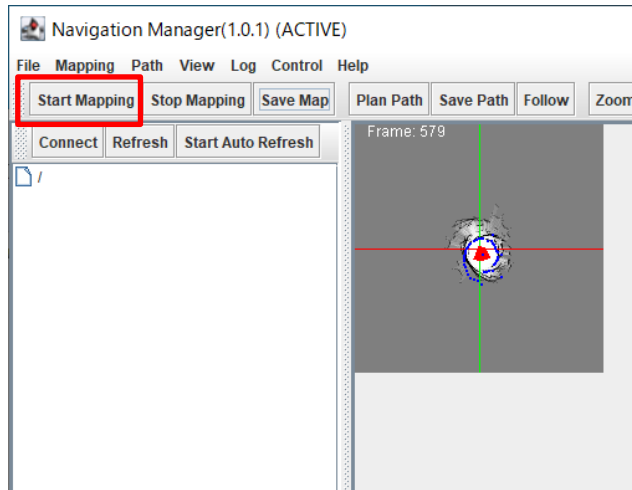
- RTCをアクティブ化する



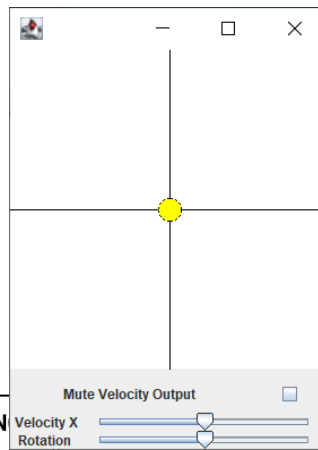
- Raspberry Piマウスのモーター電源をオンにする
 - 2つのスイッチのうちの外側のスイッチ

システムの実行

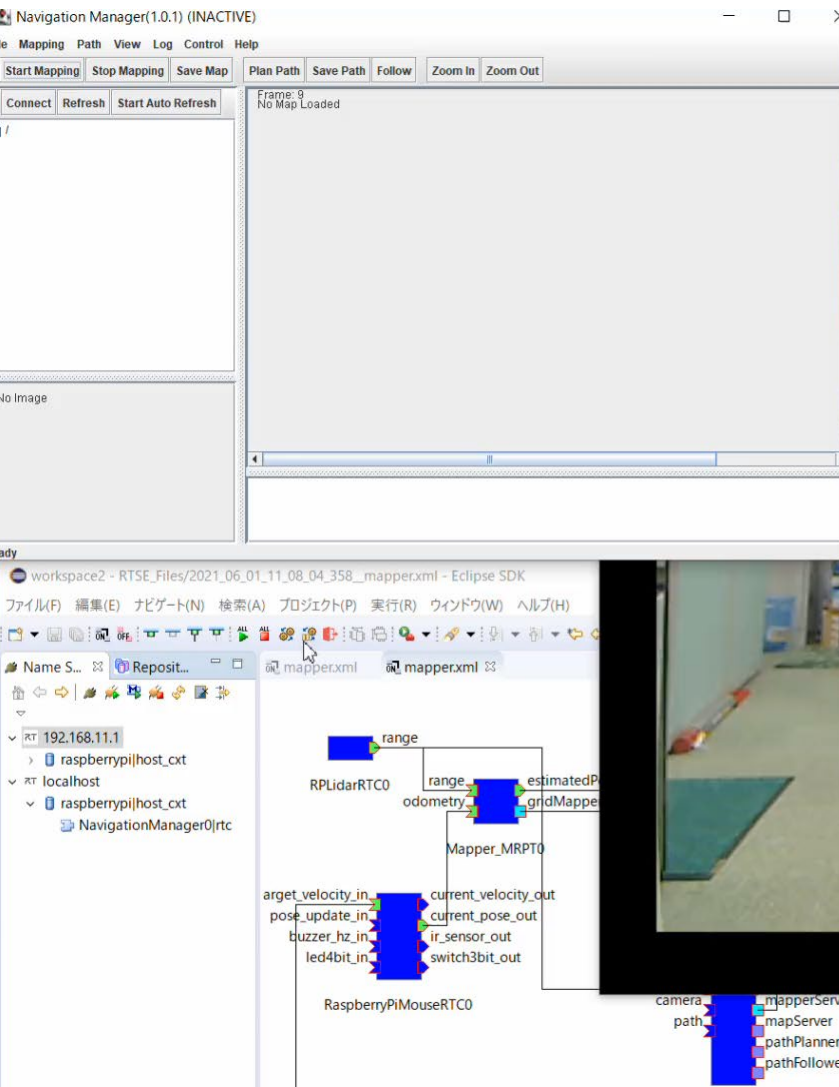
- NavigationManagerの**Start Mapping**ボタンを押す。



- 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。



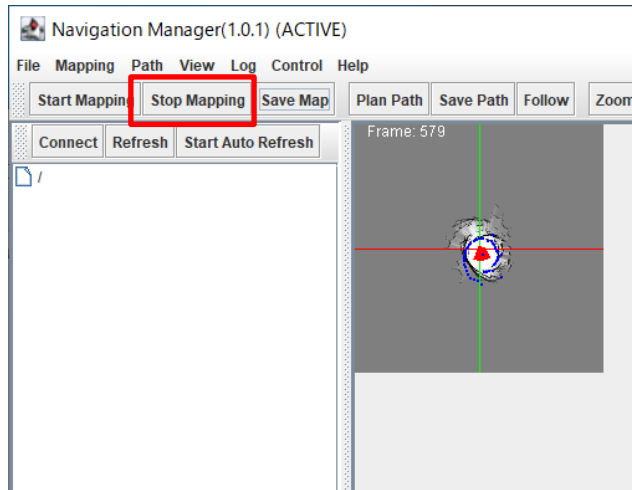
システムの実行



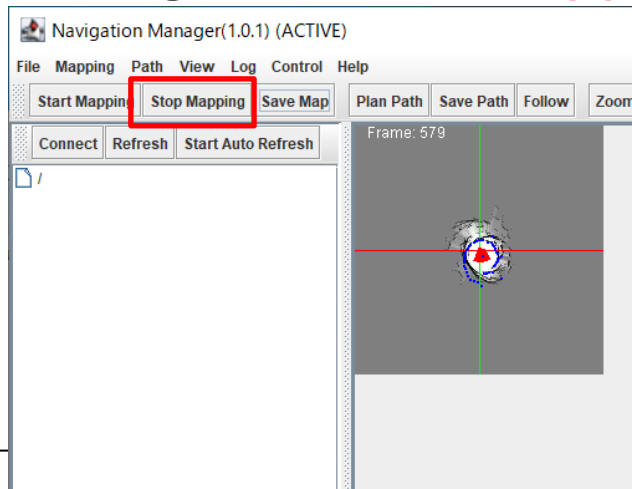
RTSystemEditorで
RTCをアクティブ化します。

システムの終了

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

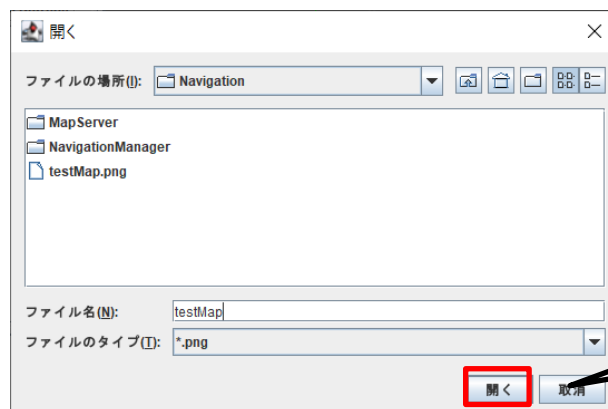


- NavigationManagerの**Save Mapping**ボタンを押す。



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。



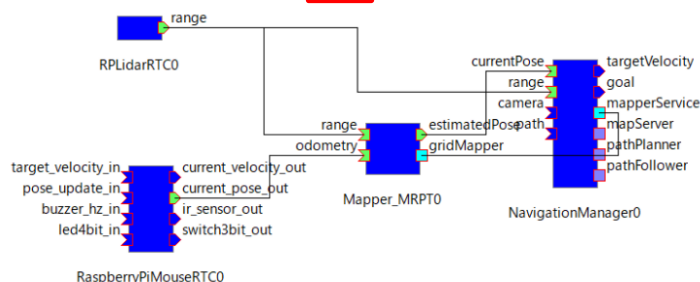
開くボタンを押すと
保存される。

- 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**



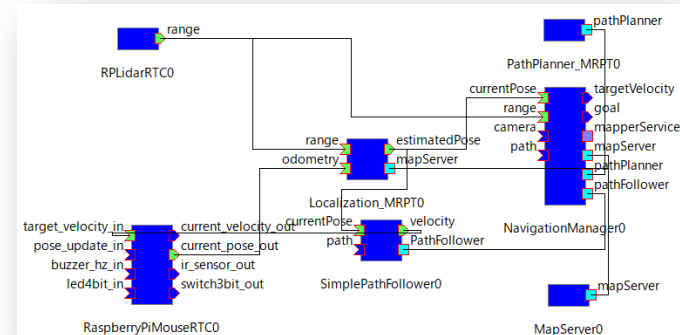
ナビゲーション実習

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。

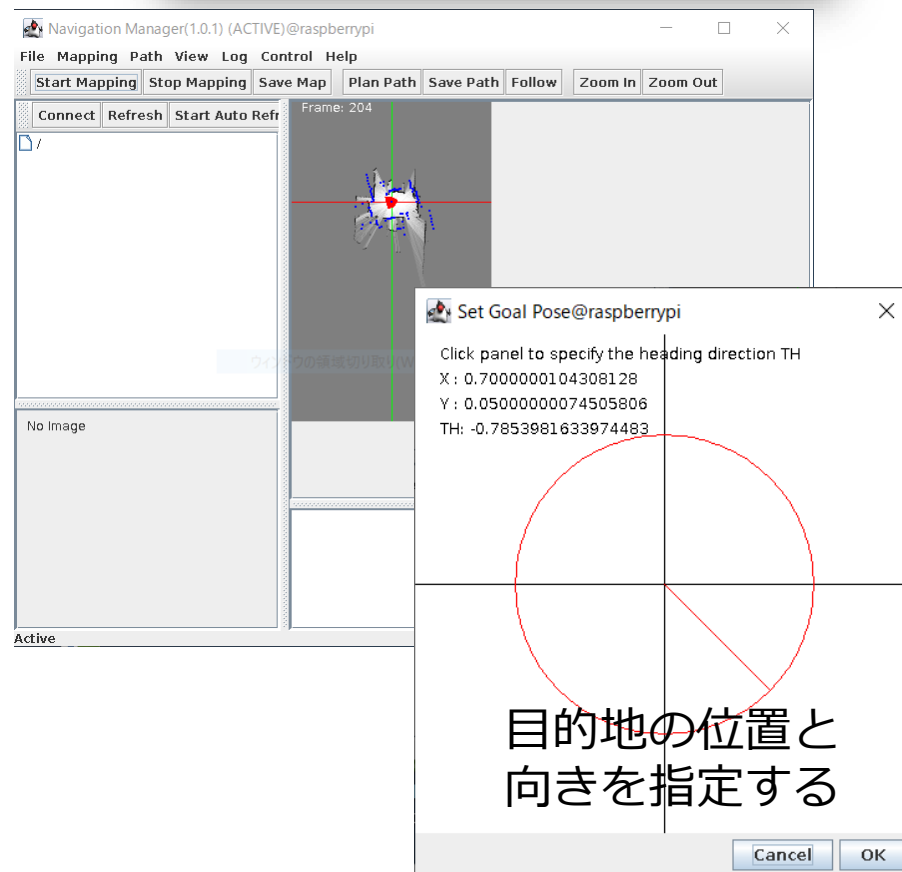


3. 経路生成・移動 (Navigation)

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。



RTCの起動、ポートの接続

- 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(Windows)
 - MapServer.sh(Ubuntu)

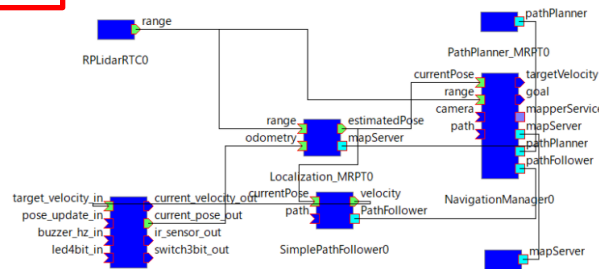
RTCの起動、ポートの接続

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

PathPlan

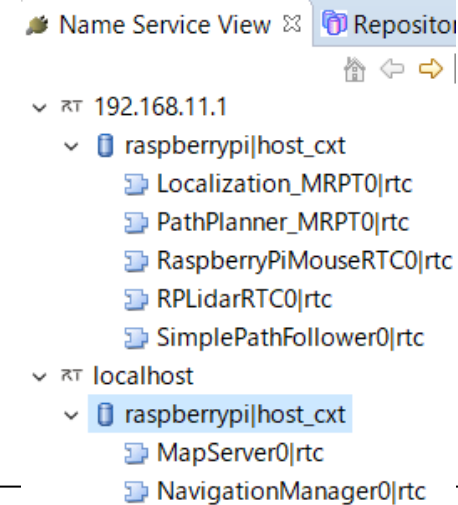
RasPiMouse navigation by using a path planner using MRPT.

Start Connect Activate Deactivate Stop

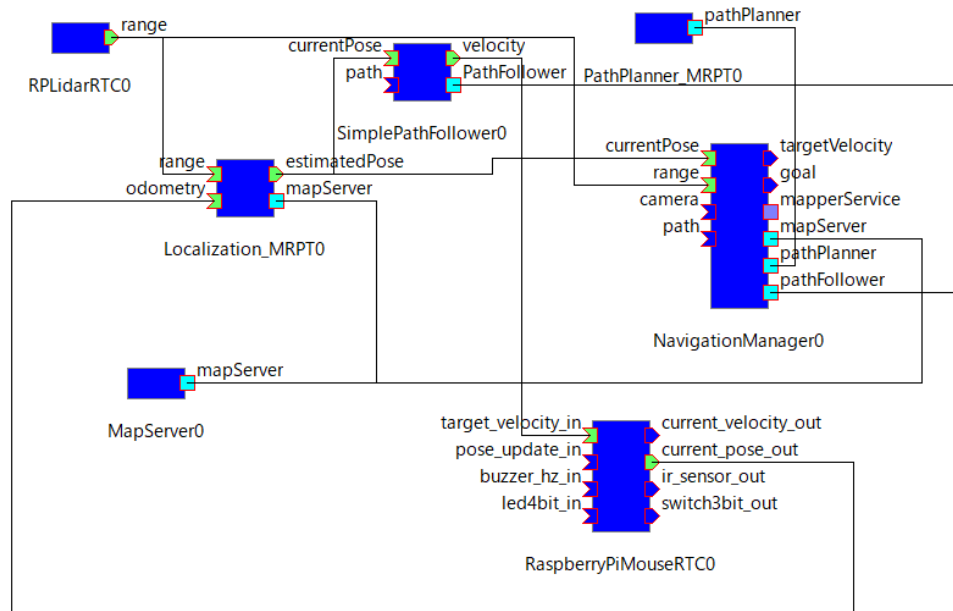


- ネームサーバーに以下のRTCが登録されている状態になる

- Localization_MRPT0
- PathPlanner_MRPT0
- RaspberryPiMouseRTC0
- RPLidarRTC0
- SimplePathFollower0
- MapServer0
- NavigationManager0



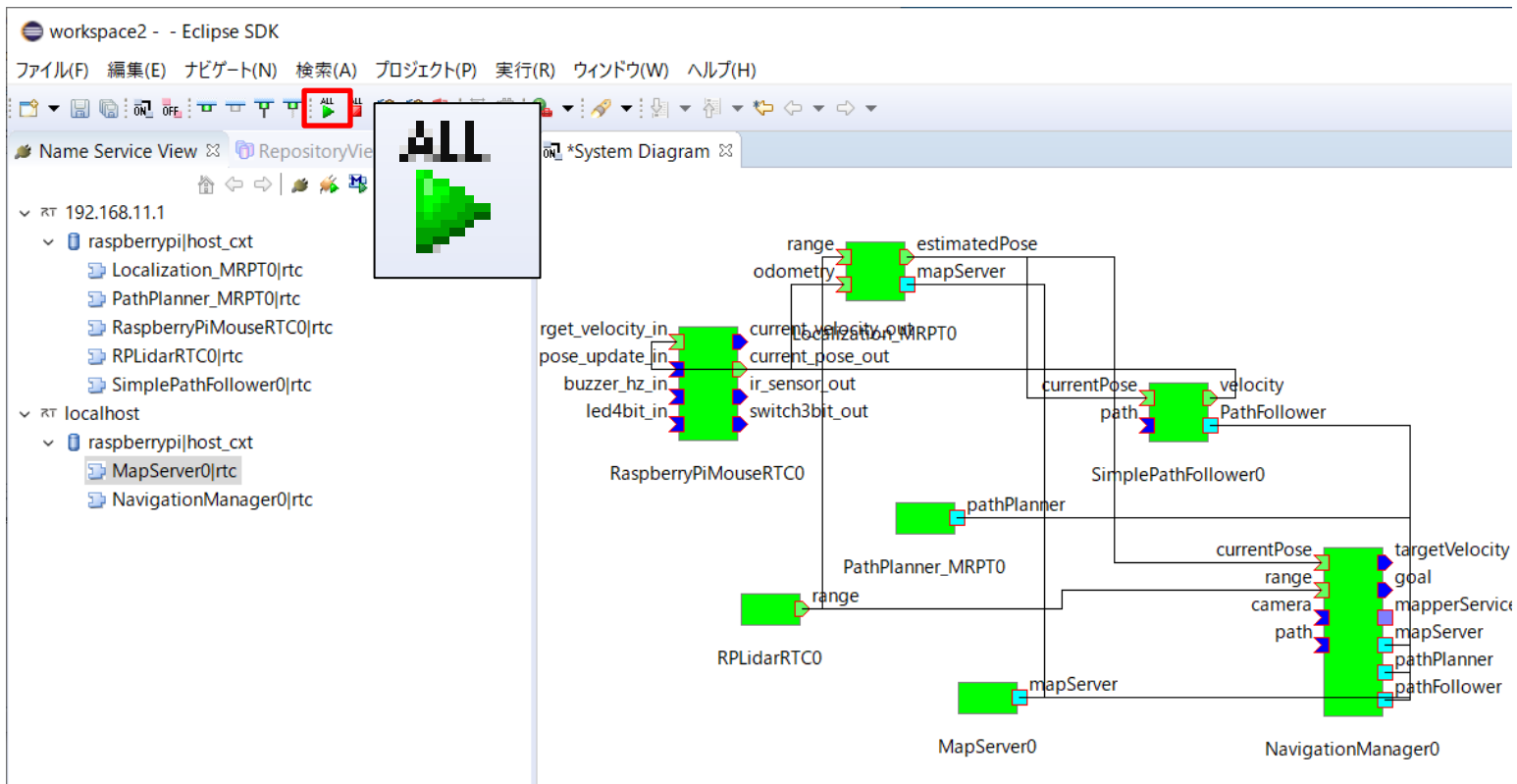
- NavigationManager、MapServerのポートを接続する



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	RPLiderRTC0	range
MapServer0	mapServer	Localization_MRPT	mapServer

システムの実行

- RTCをアクティブ化する

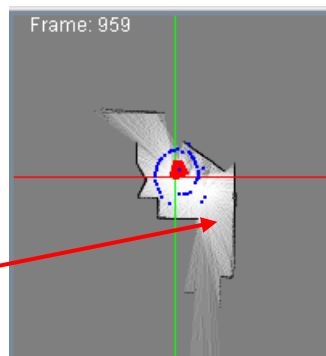


- Raspberry Piマウスのモーター電源をオンにする

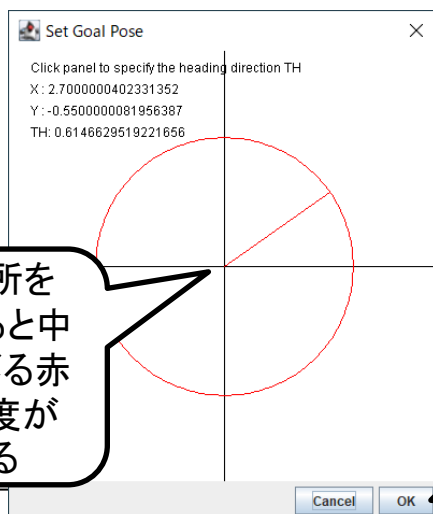
システムの実行

- 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

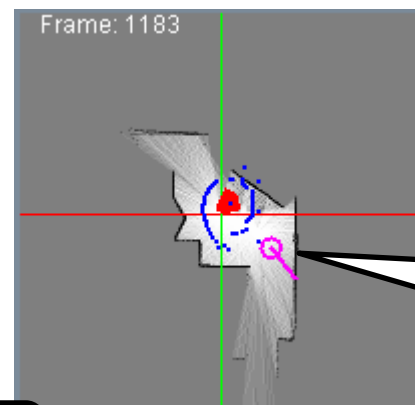


- 以下の画面で目標角度を設定する。



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

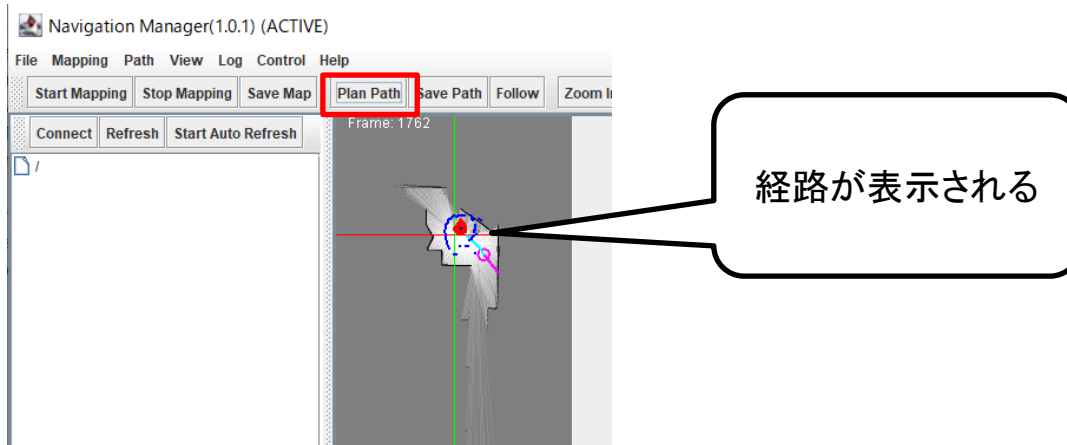
角度を設定したらOKをクリックする



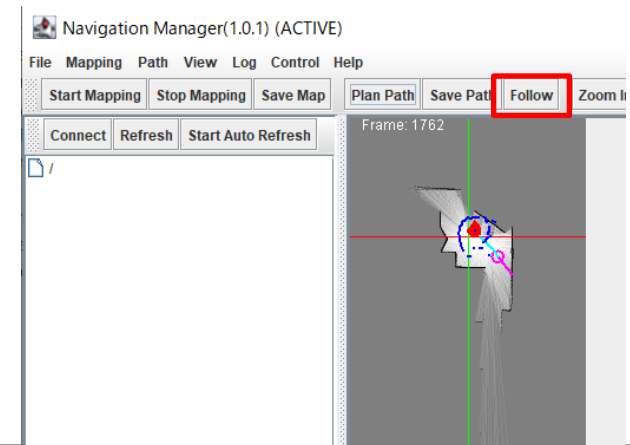
地図上に目標位置が表示される

システムの実行

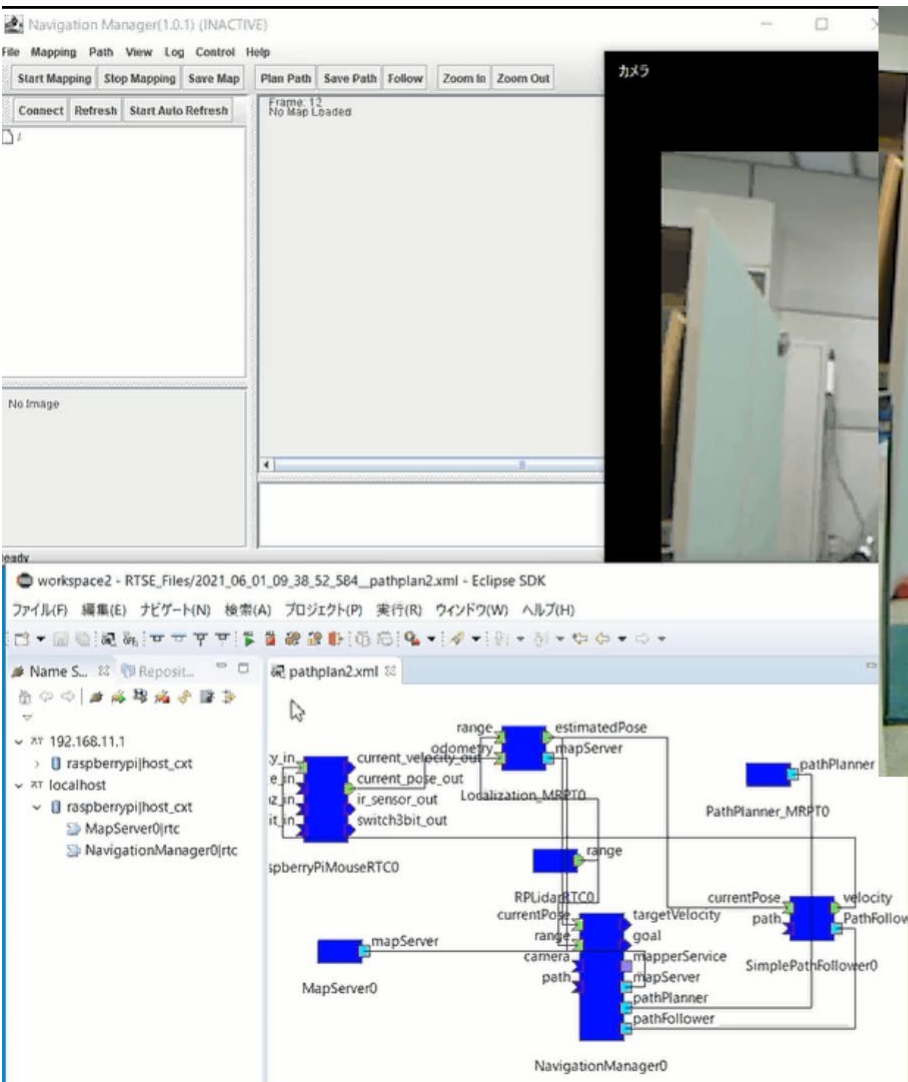
- 経路を計画する
 - NavigationManagerのPlan Pathボタンをクリックする



- 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



システムの実行



RTSystemEditorで
RTCをアクティブ化します

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

