

# 地図生成自動化プログラム実行手順書



# 1 目次

---

2	使用した環境.....	4
3	地図生成自動化プログラム実行.....	4
4	地図生成自動化プログラムの説明.....	6
5	著作権.....	8

## 2 使用した環境

---

OS : Ubuntu 20.04 LTS

CPU : Intel® Xeon® CPU E3-1505M v6 @ 3.00GHz

メモリ : 16GB

SSD : 256GB

ROS : Noetic Ninjemys

Choreonoid : 1.8 (2022/7/6 のリビジョンを使用)

ナビゲーション用ソフトウェア : 配送システム (2020 年度に日本アドシス様、TIS 様が開発)

使用したロボットモデル : 配送システム開発時に用いた TurtleBot3 を模倣したモデル

環境モデル : LICTiA (会津大学先端 ICT ラボ) 1 階を模倣したモデル

## 3 地図生成自動化プログラム実行

---

以下のコマンドを実行すると、各ノードが起動され Choreonoid による地図生成を実行します。

```
$ cd ~/catkin_ws/src/cnoid_turtlebot_pkgs/cnoid_turtlebot_slam/script
```

```
$ ./create_map_exe.bash
```

## 地図生成自動化プログラム環境構築手順書

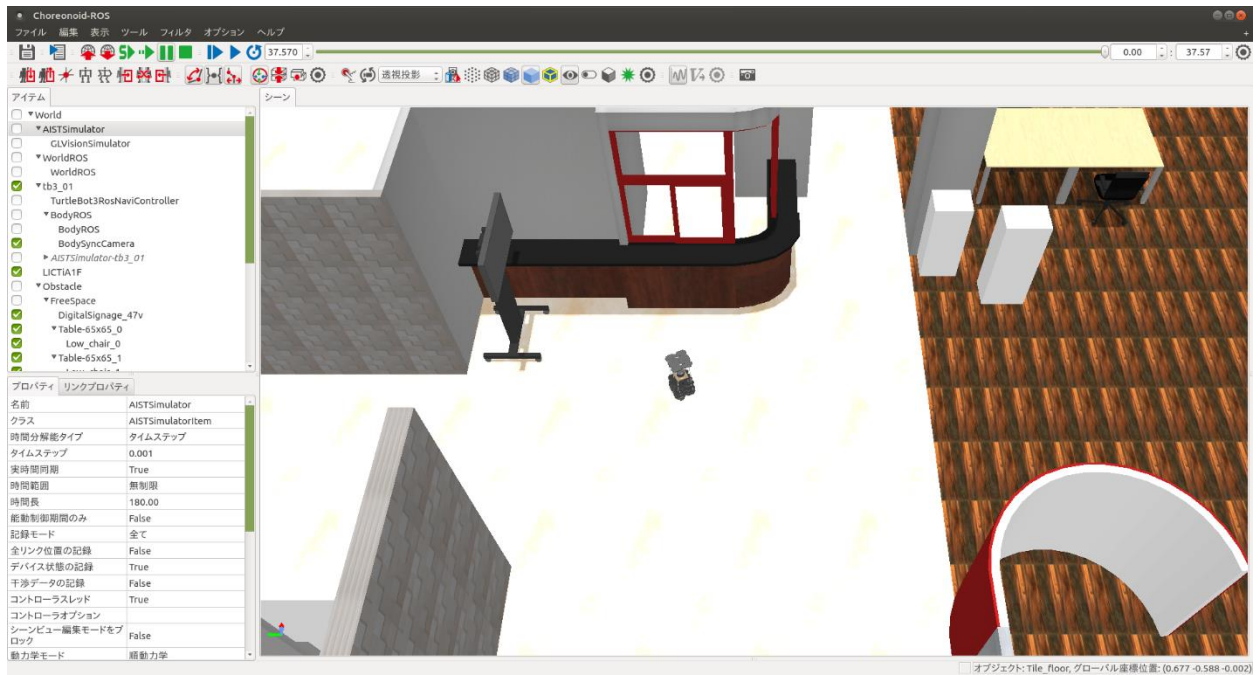


図 1 地図生成自動化プログラム実行時のChoreonoid

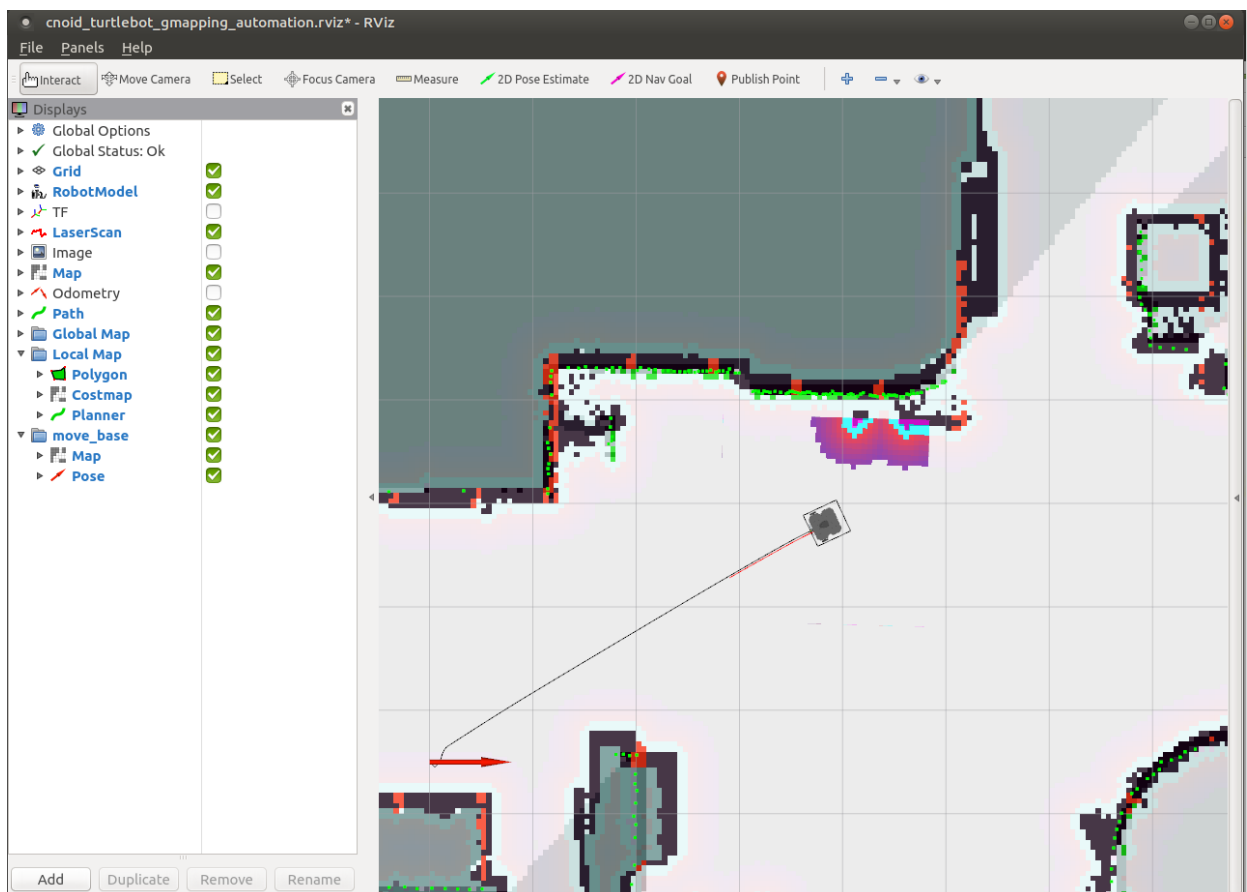


図 2 地図生成自動化プログラム実行時のRviz

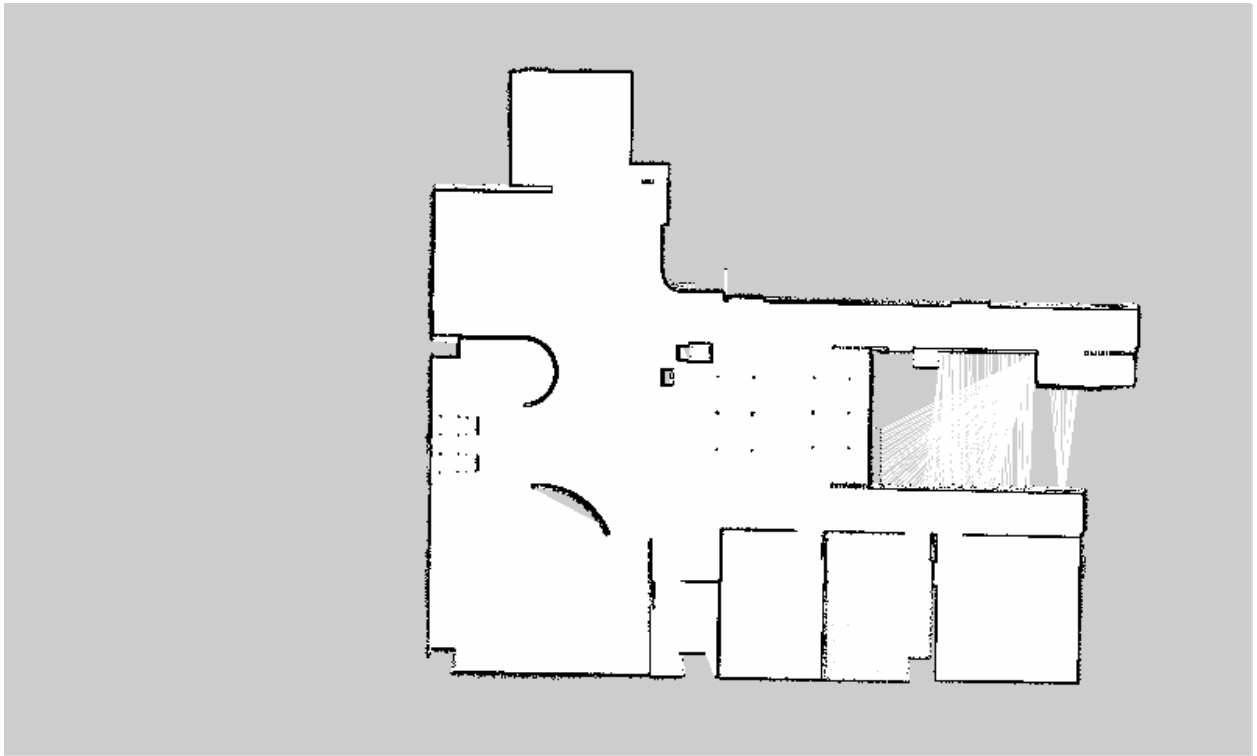


図 3 Choreonoid で生成した地図

## 4 地図生成自動化プログラムの説明

---

“create\_map\_exe.bash”は、同ディレクトリ内の”launch\_ros\_node.bash”を呼び出しており、複数のターミナル開き以下のコマンドを実行しています。

【ターミナル① ROS マスター】

```
$ roscore
```

【ターミナル② Choreonoid と gmapping による SLAM 関連ノード】

```
$ roslaunch cnoid_turtlebot_slam  
cnoid_turtlebot_lictia_slam_automation.launch
```

【ターミナル③ ナビゲーション関連ノード】

```
$ roslaunch cnoid_turtlebot_navigation  
cnoid_turtlebot_navigation_automation.launch
```

## 地図生成自動化プログラム環境構築手順書

### 【ターミナル④ 地図更新ノード】

```
$ roslaunch map_controller map_organizer_tb3_0_automation.launch
```

### 【ターミナル⑤ ロボット制御統括ノード】

```
$ roslaunch delivery_robot delivery_manager_tb3_0_lictia_automation.launch
```

### 【ターミナル⑥ 経路コストマップ配信ノード】

```
$ roslaunch cnoid_turtlebot_navigation  
map_server_costmap_lictia_automation.launch
```

### 【ターミナル⑦ 地図保存ノード】

```
$ rosrun cnoid_turtlebot_control saveMap
```

### 【ターミナル⑧ 移動指示ノード】

```
$ rosrun delivery_robot edge_node_beta tb3_01 tb3 200
```

なお、本プログラムは、地図生成に一般的な SLAM アルゴリズムである gmapping、自律走行には日本アドシス様が開発した配送システムを使用しており、移動指示ノードでは 200 番を指定することで経由地点を格納している YAML ファイル(waypoints\_list.yaml)を読み込み、指定されている経由地点に移動します。全ての経由地点に到達した場合に自動で地図を保存します。生成した地図データは以下に保存されます。

### 【地図保存場所】

```
~/catkin_ws/src/cnoid_turtlebot_pkgs/cnoid_turtlebot_navigation/maps/mymap_egomap
```

自律走行中にエラーとなった場合、地図は保存せずに地図生成処理を終了します。経由地点は、位置([m])・姿勢([°])を [ x, y, z, roll, pitch, yaw ] のように指定し、カンマ区切りでいくつでも指定可能です。経由地点の記述方法は、以下の「【waypoints\_list.yaml の記述方法】」をご参照ください。

### 【waypoints\_list.yaml の記述方法】

```
waypoints: [ [ x0, y0, z0, roll0, pitch0, yaw0 ], [ x1, y1, z1, roll1, pitch1,  
yaw1 ], . . . [ xn, yn, zn, rolln, pitchn, yawn ] ]
```

## 5 著作権

---

本文書の著作権は公立大学法人会津大学に帰属します。

この文書のライセンスは以下のとおりです。

[クリエイティブ・コモンズ表示 2.1 日本](#)

[Creative Commons — 表示 2.1 日本 — CC BY 2.1 JP](#)

