

# 抜粋版

はじめてのLiDAR開発キット2 (Livox版、ROS2対応)  
～Livox LiDARを使って自動運転, 3D地図, ドローンに活用～  
開発編 (ubuntu22.04, ros2)

LIVOX

ROS2™



スペクトラム・テクノロジー株式会社

<https://spectrum-tech.co.jp>

[sales@spectrum-tech.co.jp](mailto:sales@spectrum-tech.co.jp)

# 開発キット 目次

	ページ
• ubuntu運用マニュアル	
• ubuntuについて	<a href="#">4</a>
• Linux基本コマンド	<a href="#">4</a>
• ubuntu基本操作	<a href="#">6</a>
• 日常運用(ウイルススキャン、更新)	<a href="#">7</a>
• 開発キット 全体像	<a href="#">9</a>
1. ハード概要	<a href="#">10</a>
2. ソフト概要	<a href="#">11</a>
3. 3D地図開発	
① lio-sam mid70,avia	12
① lio-sam mid360,hap	22
② fast lio ros2	24
③ lidarslam_ros2	27
④ li_slam_ros2	30
⑤ Gps	33
⑥ Cloudcompare	36

抜粋版のため本文とページは一致しません

# 開発キット 目次

	ページ
4. Autoware開発	
① Autoware基本	46
② Autoware Launch GUI	54
③ Rosbag replay simulation	55
④ Scenario simulation (ros2版)	57
⑤ Scenario simulation (docker版)	58
⑥ Digital twin simulation:AWSIM	60
⑦ Autoware起動	62

抜粋版のため本文とページは一致しません

# Ubuntu運用マニュアル

## 1. Ubuntuについて

Linuxの中でも一番シェアの高いOSです。2004年にDebian系から派生。

## 2. Linux基本コマンド

### ① システム関係

- 起動: 電源を入れると自動で起動します。
- 再起動: `$ reboot`  
又は、左上のメニューの「ゲストを再起動」
- 終了: `$ shutdown`  
又は、左上のメニューの「ゲストをシャットダウン」
- ログアウト `$ exit`  
ルートからログアウトします
- **日本語／英語の入力切替**: 半角／全角のボタン(ESCボタンの下)

# Ubuntu運用マニュアル

## 2. Linux基本コマンド

### ② ディレクトリ操作、コピー、移動、削除

masa@ubuntu:~\$ **cd** /home/masa/Documents

ディレクトリの切り替え

masa@ubuntu:~/home/masa/Documents\$ **ls** ファイルとディレクトリの表示(表示したら操作したいファイルを右クリックでコピーして操作します)

masa@ubuntu:~\$ **cp** ファイル名 ディレクトリ 配下のディレクトリのファイルを別のディレクトリへコピー

masa@ubuntu:~\$ **mv** ファイル名 ディレクトリ 配下のディレクトリのファイルを別のディレクトリへ移動

masa@ubuntu:~\$ **rm** ファイル名 ファイルの削除

便利な機能  
でのコマンド共通(マイナスを2個とhelp)

rm -help  
コマンドのオプションが分からない場合は、ヘルプで問い合わせる。すべ

### ③ ユーザ権限、プロセス他

masa@ubuntu:~\$ **su** -

スーパーユーザ(root)に切り替え、パスワードを入力

masa@Ubuntu:~\$ **sudo**

ルート権限で各種コマンドを実施します。

masa@ubuntu:~\$ **ps** a

現状の動いているプロセスを表示

masa@ubuntu:~\$ **kill**

特定のプロセスを強制終了

masa@ubuntu:~\$ **apt-get** install pkg

パッケージのインストールなどに使用

masa@ubuntu:~\$ **date**

日付、時間の設定を行います。

masa@ubuntu:~\$ **leafpad** /etc/network/interfaces

インタフェースに記述している内容を変更します。Viよりも使いやすいです。

### ④ モジュール、usb、メモリ、HDDなどの表示

masa@ubuntu:~\$ **lsmod**

linuxのモジュールリスト表示

masa@ubuntu:~\$ **lsusb**

usbのデバイス表示

masa@ubuntu:~\$ **free -mt**

メモリ使用状態表示

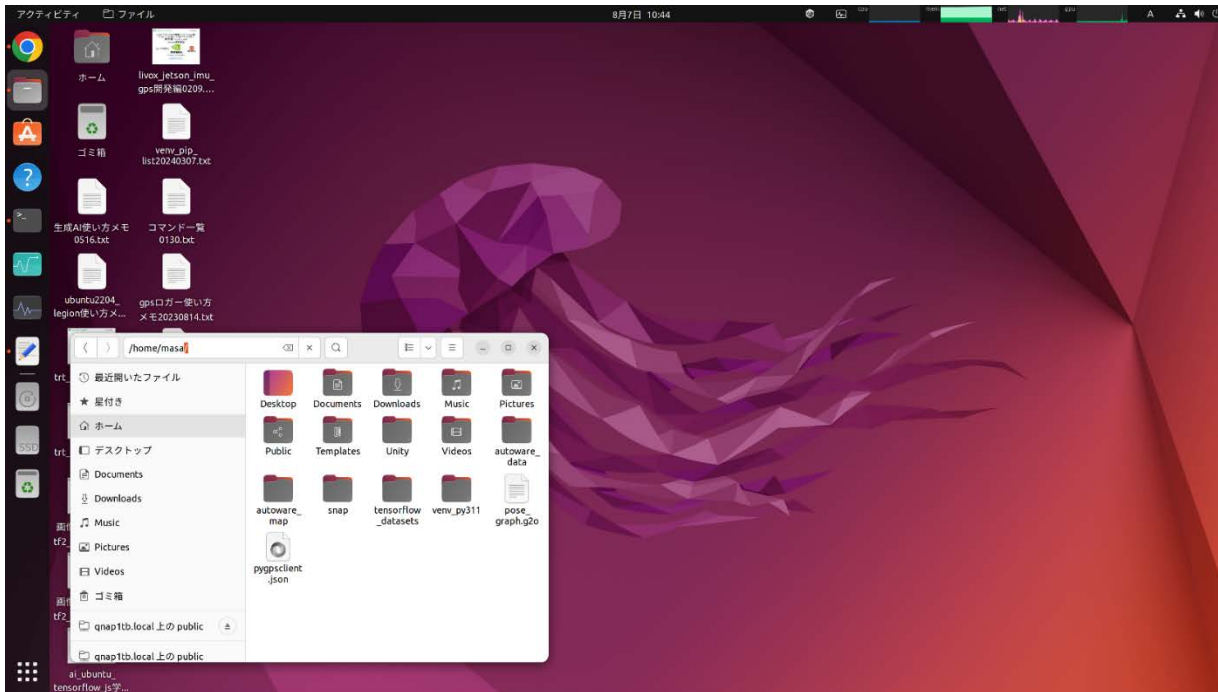
masa@ubuntu:~\$ **df**

HDD(マイクロSD)の使用状態表示

# Ubuntu運用マニュアル

## 3. 基本操作

### ① 表示画面と内容



主に使用するもの

- ・ブラウザ: Chrome
- ・フォルダ: Documents内に必要なファイルがあります。
- ・コマンド: コマンド画面を立ち上げて、python3のプログラムを動作させます。

## LiDAR開発キット2 全体像(Livox版,ROS2対応)

### PCハードウェア

CPU



cpu,32GB RAM

+

GPU

Nvidia RTX40,30  
シリーズ等に対応



USBメモリ



OS



関連ソフト系



Livoxハードウェア

オプションで提供  
します。

Mid-70



Avia

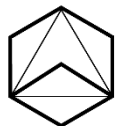


Horizon



自動運転系

ROS 2™



THE  
AUTOWARE  
FOUNDATION

3D地図系

Lio-sam, fast lio  
ros2,  
lidarslam\_ros2,  
li\_slam\_ros2

プログラム言語

python



# 1. ハードウェア概要

## ①必要なハードウェア仕様

ハードウェアの概要です。

区分		プロダクト	メーカー	備考
USBメモリ (livox版, ROS2 対応)		128GB USB3.0 livox関連ソフト		
お客様準備品				
PC本体	cpu	Intel i9, i7, amd Ryzen 9,7など cpu		GPUが搭載できるものに限ります。
	GPU	RTX40,30,20シリーズ	nvidia	
	メモリ	32GB以上		
	SSD	1TB以上		
LiDAR	Mid-70	Detection Range (@ 100 klx):260 m @ 80% reflectivity FOV:70.4° (Circular) Angular Precision (1σ ):< 0.1° Beam Divergence:0.28° (Vertical) × 0.03° (Horizontal) Point Rate:100,000 points/s (first or strongest return)	Livox	弊社からの提供も可能です。 livox製の他の製品も対応可能です。検証は、mid-70で実施しました。
	Avia	Detection Range (@ 0 klx):450 m @ 80% reflectivity FOV:Non-repetitive scanning pattern: 70.4° (Horizontal) × 77.2° (Vertical) Angular Precision (1σ ):< 0.05° Beam Divergence:0.28° (Vertical) × 0.03° (Horizontal) Point Rate:240,000 points/s (first or strongest return)		



## 2. ソフトウェア概要

### ①ソフトウェア一覧

ソフトウェアの概要です。

- 弊社からのマニュアルに基づいてインストールします。
- OS,GPUは、お客様で実施してください。

区分	ソフト名	バージョン	備考
OS	ubuntu	22.04.1LTS	
GPU用	cuDNN	8.9.0+cuda12.0	Gpuに依存
プログラム言語	python3	3.10.12	
Livox関係	Livox-sdk	2.3.0	Mid70,avia,horizon
	Livox-ros2-driver	0.0.1	同上
	Livox-sdk2	1.2.5	Hap,mid360用
自動運転	Livox-ros-driver2	1.2.4	同上
	Autoware	1.0.0	Ros2対応
	ROS2	humble	
関連	AWSIM	1.2.3	unity上で動作
	Docker-ce	>19.03.5	
各種アプリ	unity	2021.1.7f1	Awsim対応
	Lio-sam, fast lio ros2, lidar slam_ros2, li_slam_ros2, cloudcompare,gpsなど多数		

## 3. 3D地図開発

### ① lio-sam デモ

地図作成: SLAM (Simultaneous Localization And Mapping)に使用する地図作成ソフト

<https://github.com/TixiaoShan/LIO-SAM>

起動

source /home/masa/Documents/ws\_livox/install/setup.bash 端末毎に実施のこと

デモ1

ros2 launch lio\_sam run.launch.py

ros2 launch lio\_sam run.demo\_launch.py 上記の改造版

別端末で

ros2 bag play /home/masa/Documents/mapfiles/lio\_sam/walking\_dataset/

録画分で確認のこと

ros2 topic list

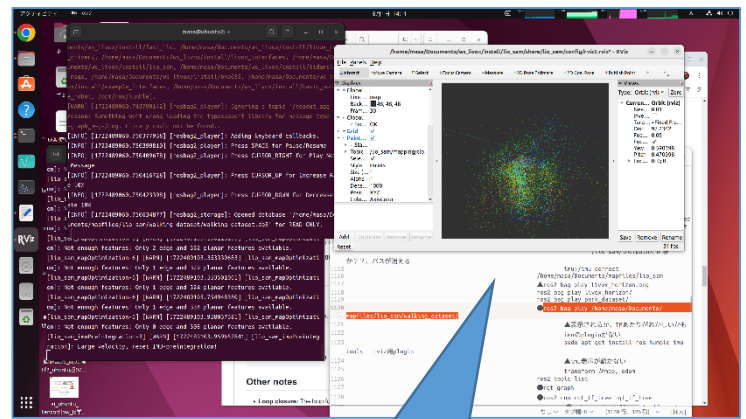
録画データのtopic確認

```

コマンド入力
$ source
/home/masa/Documents/ws_livox/install/setup.bash
$ ros2 launch lio_sam run.launch.py
$ ros2 launch lio_sam run.demo_launch.py
$ ros2 bag play
/home/masa/Documents/mapfiles/lio_sam/walking_dataset/
$ ros2 topic list
    
```

```

masa@ubuntu2: ~
masa@ubuntu2:~$ source /home/masa/Documents/ws_livox/install/setup.bash
masa@ubuntu2:~$ ros2 topic list
/clicked_point
/diagnostics
/events/read_split
/goal_pose
/gx5/gps/fix
/gx5/nav/odometry
/gx5/nav/status
/imu_correct
/imu_raw
/initialpose
/joint_states
/lio_loop/loop_closure_detection
/lio_sam/deskew/cloud_deskewed
/lio_sam/deskew/cloud_info
/lio_sam/feature/cloud_corner
/lio_sam/feature/cloud_info
/lio_sam/feature/cloud_surface
/lio_sam/imu/path
    
```



pathの表示がおかしい。  
Path再生時は、[こちら](#)を使用のこと

## 3. 3D地図開発

### ① lio-sam mid70, avia録画

地図作成: SLAM (Simultaneous Localization And Mapping)に使用する地図作成ソフト

<https://github.com/TixiaoShan/LIO-SAM>

起動

source /home/masa/Documents/ws\_livox/install/setup.bash 端末毎に実施のこと

録画 mid70

ros2 launch lio\_sam run.launch.py

ros2 launch lio\_sam run.launch\_gps.py 上記の改造版でgps対応別端末で

ros2 launch livox\_ros2\_driver livox\_lidar\_launch.py

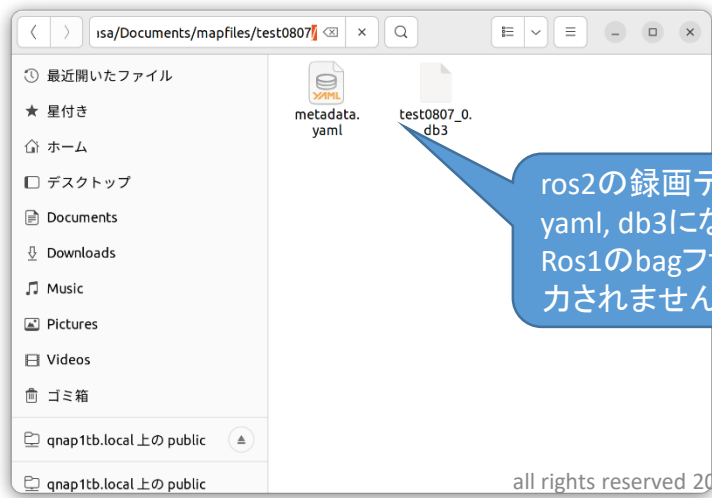
mid70起動 (avia: livox\_lidar\_config.json, broadcast codeを変更)

ros2 bag record -a -o /home/masa/Documents/mapfiles/test0807

録画

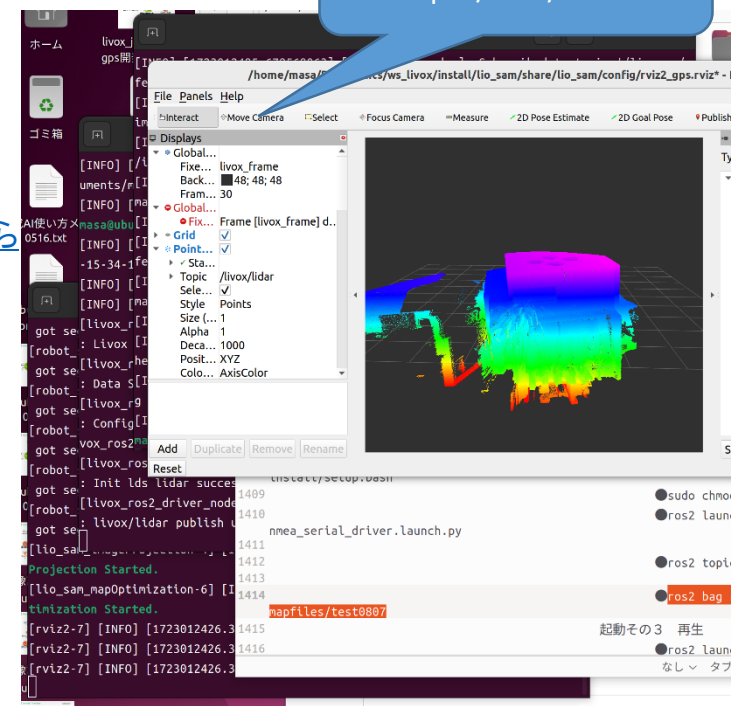
topic:mapにtransformでlivox\_frameを接続できます。方法は[こちら](#)

```
コマンド入力
$ source
/home/masa/Documents/ws_livox/install/setup.bash
$ ros2 launch lio_sam run.launch.py
$ ros2 launch lio_sam run.launch_gps.py
$ ros2 launch livox_ros2_driver livox_lidar_launch.py
$ ros2 bag record -a -o
/home/masa/Documents/mapfiles/test0807
```



ros2の録画データは、yaml, db3になります。Ros1のbagファイルは出力されません

Fixed topic:livox\_frame  
PC2 topic:/livox/lidarへ



## 3. 3D地図開発

### ① lio-sam mid70, avia再生

地図作成: SLAM (Simultaneous Localization And Mapping)に使用する地図作成ソフト

<https://github.com/TixiaoShan/LIO-SAM>

起動

source /home/masa/Documents/ws\_livox/install/setup.bash 端末毎に実施のこと

再生 mid70

ros2 launch lio\_sam run.launch.py

ros2 launch lio\_sam run.launch\_gps.py 上記の改造版でgps対応

別端末で

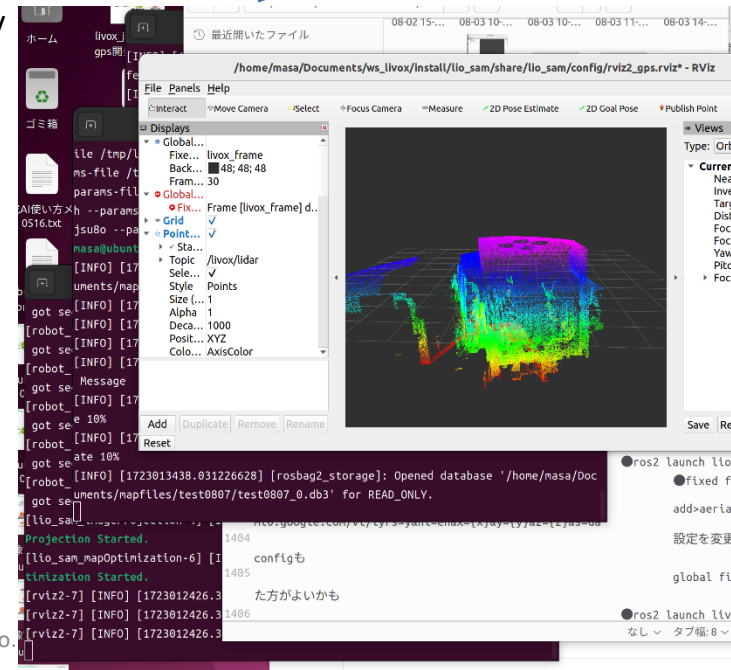
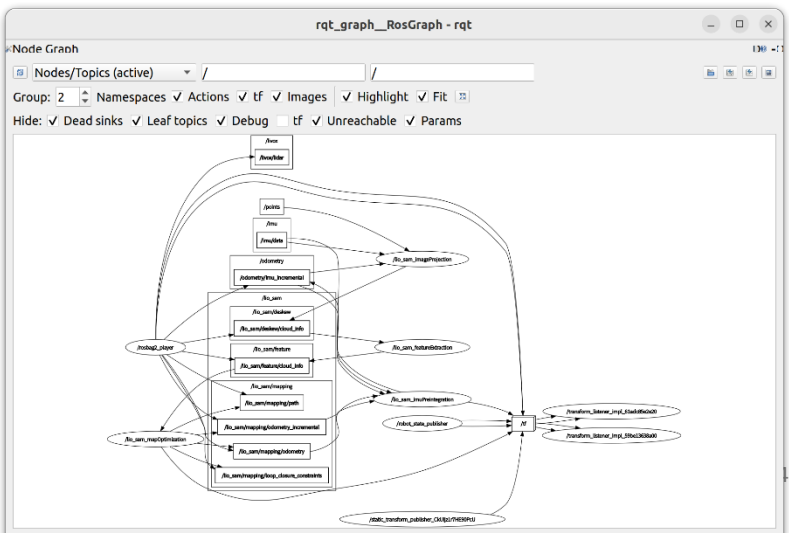
ros2 bag play /home/masa/Documents/mapfiles/test0807/

再生

rqt\_graph

```
コマンド入力
$ source
/home/masa/Documents/ws_livox/install/setup.bash
$ ros2 launch lio_sam run.launch.py
$ ros2 launch lio_sam run.launch_gps.py
$ ros2 bag play /home/masa/Documents/mapfiles/test0807/
$ rqt_graph
```

Fixed topic: livox\_frame  
PC2 topic: /livox/lidar



```
コマンド入力
$ source
/home/masa/Documents/ws_livox/install/setup.bash
$ ros2 launch rosbag2_to_pcd rosbag2_to_pcd.launch.xml
```

## 3. 3D地図開発

### ① livox-sam mid70,avia pcd変換

地図作成: SLAM (Simultaneous Localization And Mapping)に使用する地図作成ソフト

<https://github.com/TixiaoShan/LIO-SAM>

起動

source /home/masa/Documents/ws\_livox/install/setup.bash 端末毎に実施のこと

変換 mid70 pcd変換

Ros2の出力ファイルをpcdファイルに変換。

rosbag2 pcd設定

/home/masa/Documents/ws\_livox/install/rosbag2\_to\_pcd/share/rosbag2\_to\_pcd/config/内の

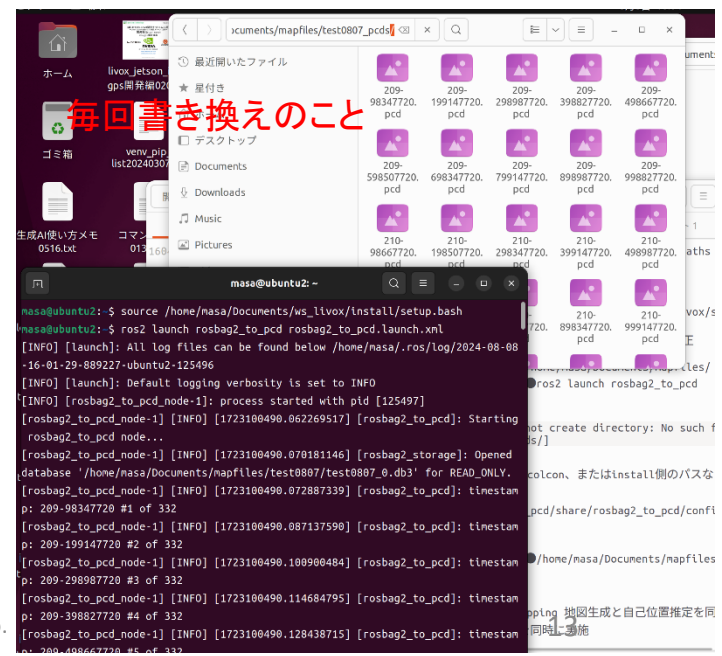
rosbag2\_to\_pcd.param.yaml

path\_bag:変換元rosbag2のフォルダ位置を設定

例: /home/masa/Documents/mapfiles/test0807

ros2 launch rosbag2\_to\_pcd rosbag2\_to\_pcd.launch.xml

Mapfilesにtest0807\_pcdが出力





## 3. 3D地図開発

### ② fast lio ros2 デモ

高速での自己位置推定。lidar,とimuのデータで実現

[https://github.com/Ericksii/FAST\\_LIO\\_ROS2](https://github.com/Ericksii/FAST_LIO_ROS2)

起動

source /home/masa/Documents/ws\_livox/install/setup.bash 端末毎に実施のこと

デモ

ros2 launch fast\_lio mapping.launch.py config\_file:=avia.yaml

別端末で

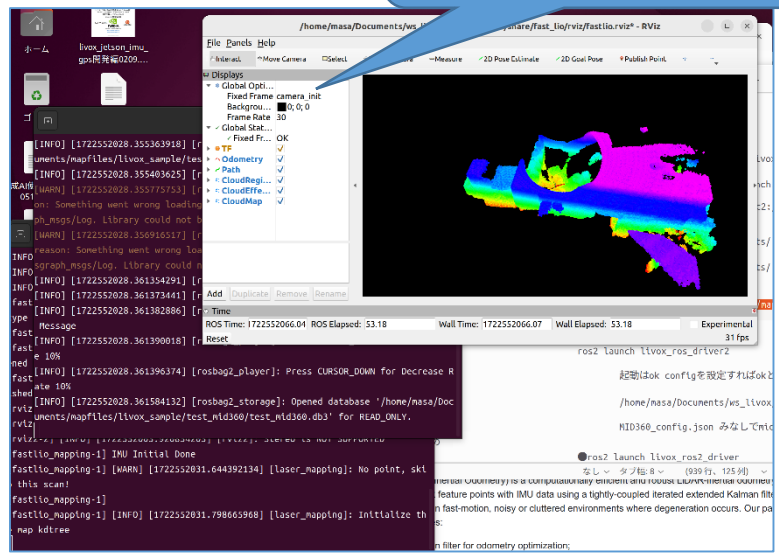
ros2 bag play /home/masa/Documents/mapfiles/livox\_sample/test\_mid360/

録画データで確認

```

コマンド入力
$ source /home/masa/Documents/ws_livox/install/setup.bash
$ ros2 launch fast_lio mapping.launch.py config_file:=avia.yaml
$ ros2 bag play
/home/masa/Documents/mapfiles/livox_sample/test_mid360/
    
```

Fixed topic:camera\_init







## 3. 3D地図開発

### ④ li\_slam\_ros2デモ

lidar slam\_ros2にLIO-SAMのIMU結合したもの,pathも正確に表示

[https://github.com/rsasaki0109/li\\_slam\\_ros2](https://github.com/rsasaki0109/li_slam_ros2)

起動

source /home/masa/Documents/ws\_livox2/install/setup.bash 端末毎に実施のこと

デモ1

rviz2 -d src/li\_slam\_ros2/scanmatcher/rviz/lio.rviz

imu:imu\_correct,path:path、pc2:points\_rawを設定:パスなど表示

別端末で

ros2 launch scanmatcher lio.launch.py

ros2 bag play /home/masa/Documents/mapfiles/lio\_sam/walking\_dataset/

録画データで確認

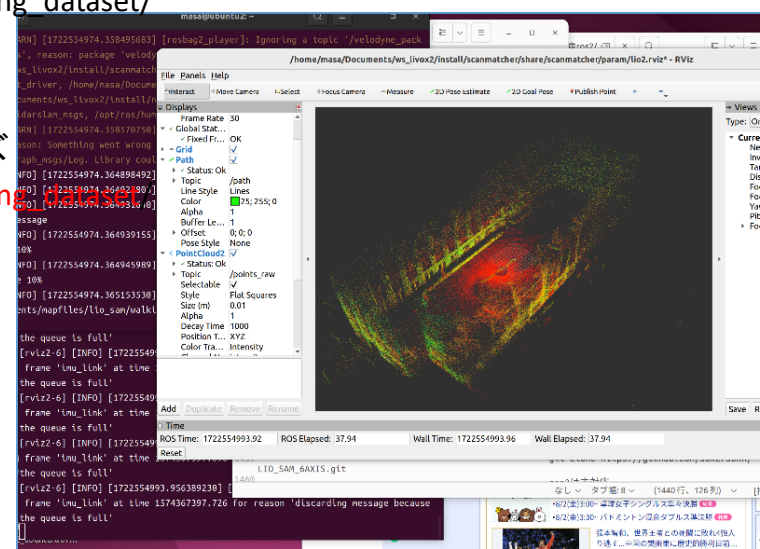
デモ2

ros2 launch scanmatcher lio.launch\_demo.py 弊社カスタマイズ

ros2 bag play /home/masa/Documents/mapfiles/lio\_sam/walking

```

コマンド入力
$ source /home/masa/Documents/ws_livox2/install/setup.bash
$ rviz2 -d src/li_slam_ros2/scanmatcher/rviz/lio.rviz
$ ros2 launch scanmatcher lio.launch.py
$ ros2 bag play
/home/masa/Documents/mapfiles/lio_sam/walking_dataset/
$ ros2 launch scanmatcher lio.launch_demo.py
    
```



## コマンド入力

```
$ source /home/masa/Documents/ws_livox/install/setup.bash  
$ ros2 launch lio_sam run.launch_gps.py  
$ ros2 bag play /home/masa/Documents/mapfiles/test080902
```

## 3. 3D地図開発

### ⑤ Gps 再生

gpsのデータをlidarに連携します。

[https://github.com/ros-drivers/nmea\\_navsat\\_driver](https://github.com/ros-drivers/nmea_navsat_driver)

[https://github.com/nobleo/rviz\\_satellite](https://github.com/nobleo/rviz_satellite)

### 起動

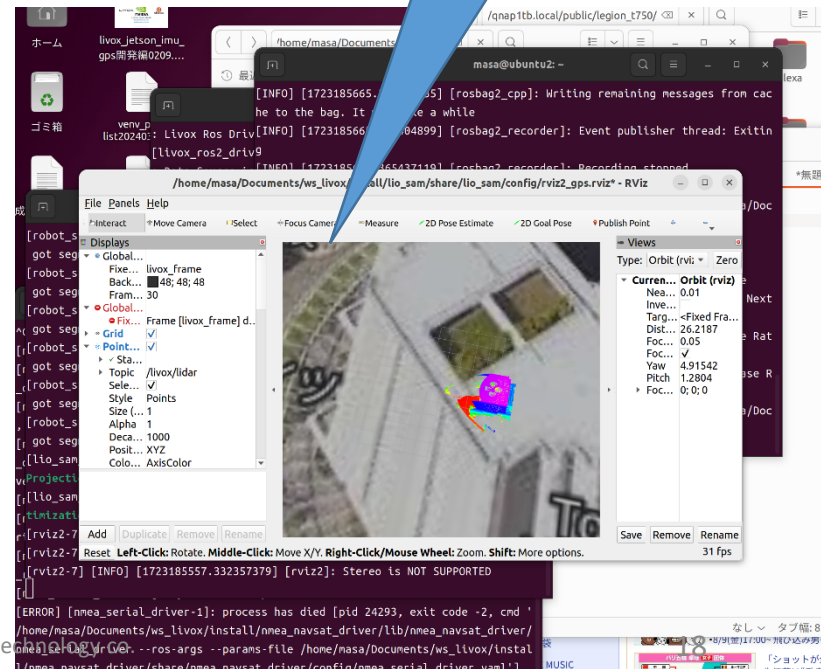
```
source /home/masa/Documents/ws_livox/install/setup.bash
```

### 関係 lio-sam mid70

```
ros2 launch lio_sam run.launch_gps.py
```

```
ros2 bag play /home/masa/Documents/mapfiles/test080902
```

Rviz2のgps表示は1回目のみ。ダイナミックに表示は変わりません。gpsデータはダイナミックに連携



## 3. 3D地図開発

### ⑥ cloudcompare

Pcdの点群データの加工、比較などを行うオープンソース

<https://www.danielgm.net/cc/>

<https://hdtopography.github.io/learning/cloudcompare/CloudCompare.html> 東大の人が解説

起動

Cloudcompareをアプリから起動

File>open: /home/masa/shared\_dir/bag\_to\_pcd\_out:全ファイル選択:例

基本操作

セグメンテーション(切り出し、断面)

合成

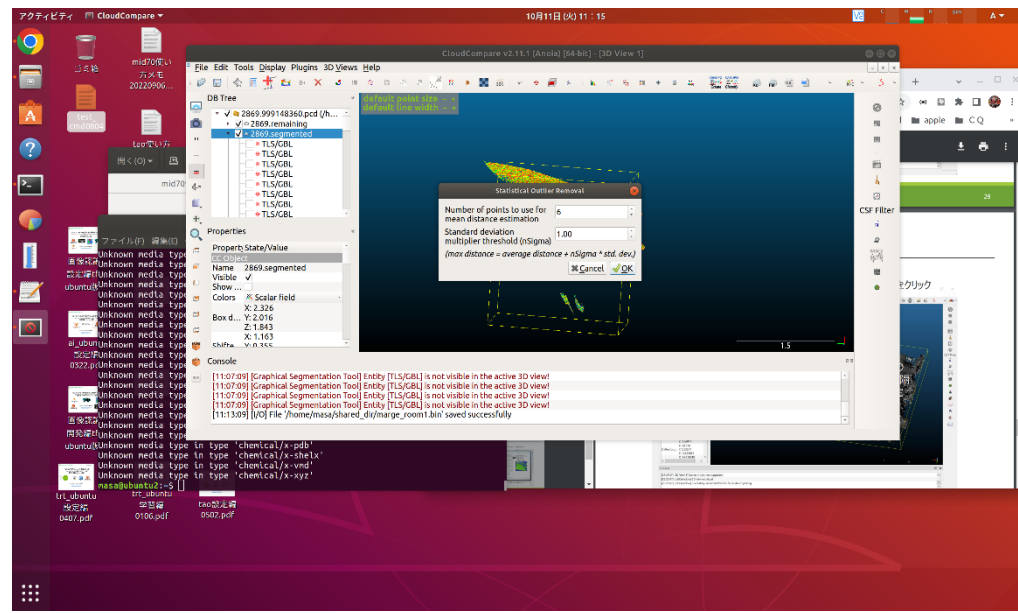
ノイズ除去

リサンプリング

メッシュ化

クラウド間距離

Pcdから他の形式に変換: plyへ



## 4. Autoware開発

### ① Autoware基本: シミュレーション1

<https://autowarefoundation.github.io/autoware-documentation/main/tutorials/>

#### Autoware起動

```
source /home/masa/Documents/autoware/install/setup.bash
ros2 launch autoware_launch planning_simulator.launch.xml
map_path:=$HOME/autoware_map/sample-map-planning vehicle_model:=sample_vehicle
sensor_model:=sample_sensor_kit
```

#### Autoware操作: 車線自動運転

2d pose estimateを押す

最初場所をクリック、その後、車をドラッグさせ方向を右クリックで道路を進む方法に左クリックでドラッグ: 向きが変わる

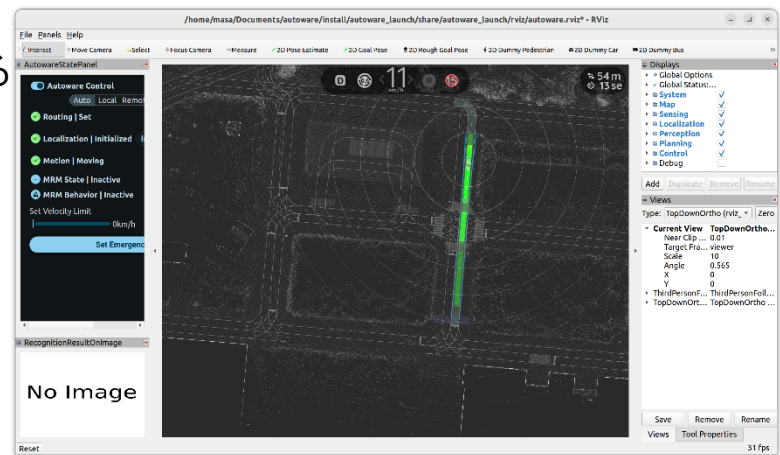
2d goal pose又は2d rough goal pose

ゴールを左クリックしてドラッグ: 道路上に色がでる  
その後、ゴール場所を押す。+が表示されるとok

Autoware control

Autoで自動で車がゴールまで動く

```
コマンド入力
$ source
/home/masa/Documents/autoware/install/setup.bash
$ ros2 launch autoware_launch
planning_simulator.launch.xml
map_path:=$HOME/autoware_map/sample-map-planning
vehicle_model:=sample_vehicle
sensor_model:=sample_sensor_kit
```



## 4. Autoware開発

### ① Autoware基本: シミュレーション7

<https://autowarefoundation.github.io/autoware-documentation/main/tutorials/>

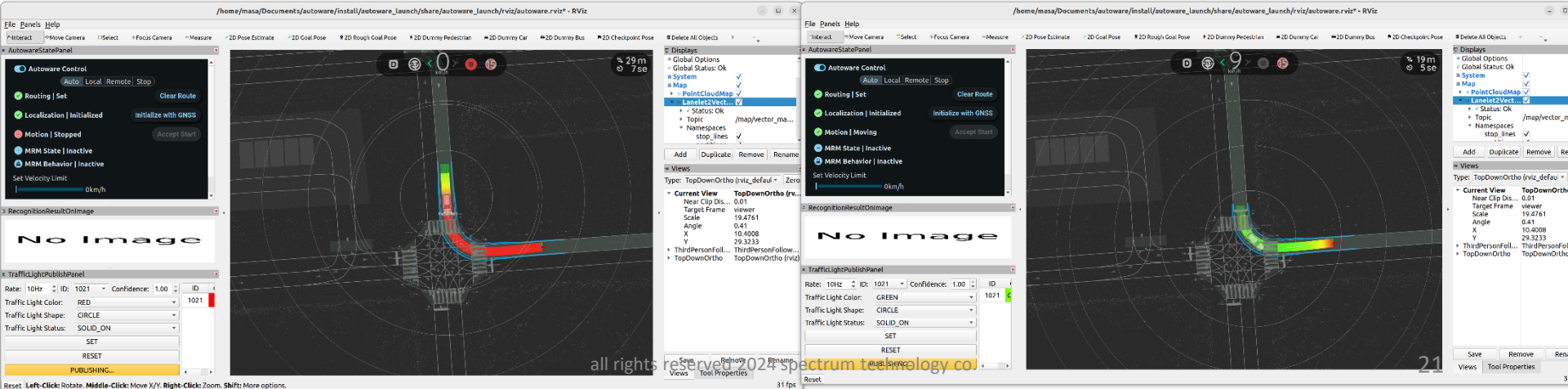
#### Autoware起動

```
source /home/masa/Documents/autoware/install/setup.bash
ros2 launch autoware_launch planning_simulator.launch.xml
map_path:=$HOME/autoware_map/sample-map-planning vehicle_model:=sample_vehicle
sensor_model:=sample_sensor_kit
```

#### Autoware操作: 信号機操作自動運転

- 2d pose estimateを押す
- 2d goal pose又は2d rough goal pose
- TrafficLightPublishPanel
- Autoware control
- Autoで自動で車が、信号で停止。上記信号をgreenにすると移動

```
コマンド入力
$ source
/home/masa/Documents/autoware/install/setup.bash
$ ros2 launch autoware_launch
planning_simulator.launch.xml
map_path:=$HOME/autoware_map/sample-map-planning
vehicle_model:=sample_vehicle
sensor_model:=sample_sensor_kit
```



```

コマンド入力
$ source
/home/masa/Documents/autoware/install/setup.bash
$ autoware-launch-gui
    
```

## 4. Autoware開発

### ② Autoware Launch GUI

<https://autowarefoundation.github.io/autoware-documentation/main/tutorials/>

#### Autoware Launch GUI起動

```

source /home/masa/Documents/autoware/install/setup.bash
autoware-launch-gui
    
```

#### Autoware Launch GUI 設定

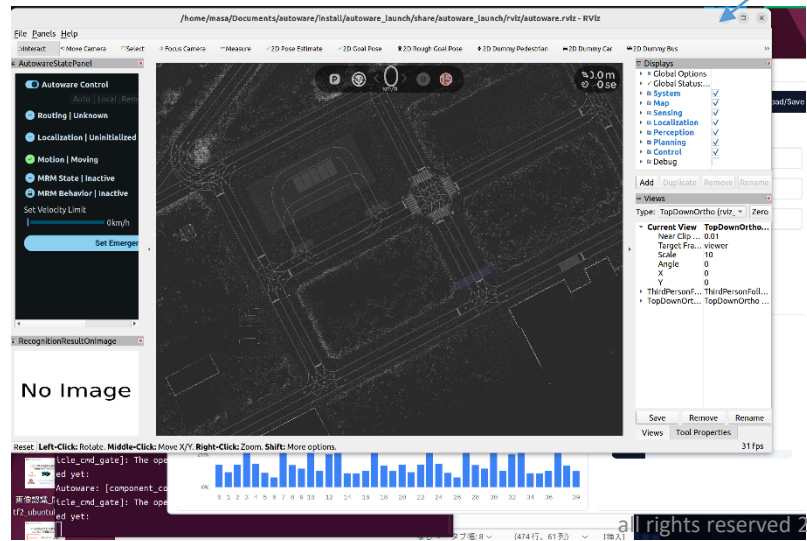
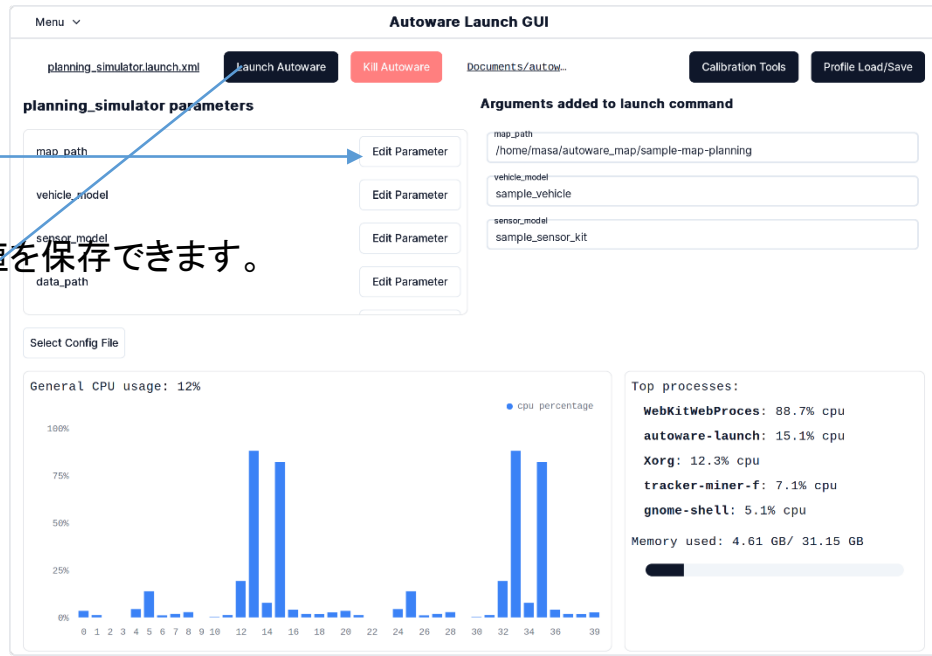
##### 車線自動運転

Map\_path, vehicle\_modelなどを設定

autoware起動

プロファイルの読み出し、保存: 設定値を保存できます。

他のシナリオも同様



# 4. Autoware開発

## ③ Rosbag replay simulation

<https://autowarefoundation.github.io/autoware-documentation/main/tutorials/ad-hoc-simulation/rosbag-replay-simulation/>

Rosbag replay simulation起動

```
source /home/masa/Documents/autoware/install/setup.bash
ros2 launch autoware_launch logging_simulator.launch.xml
map_path:=$HOME/autoware_map/sample-map-rosbag vehicle_model:=sample_vehicle
sensor_model:=sample_sensor_kit
```

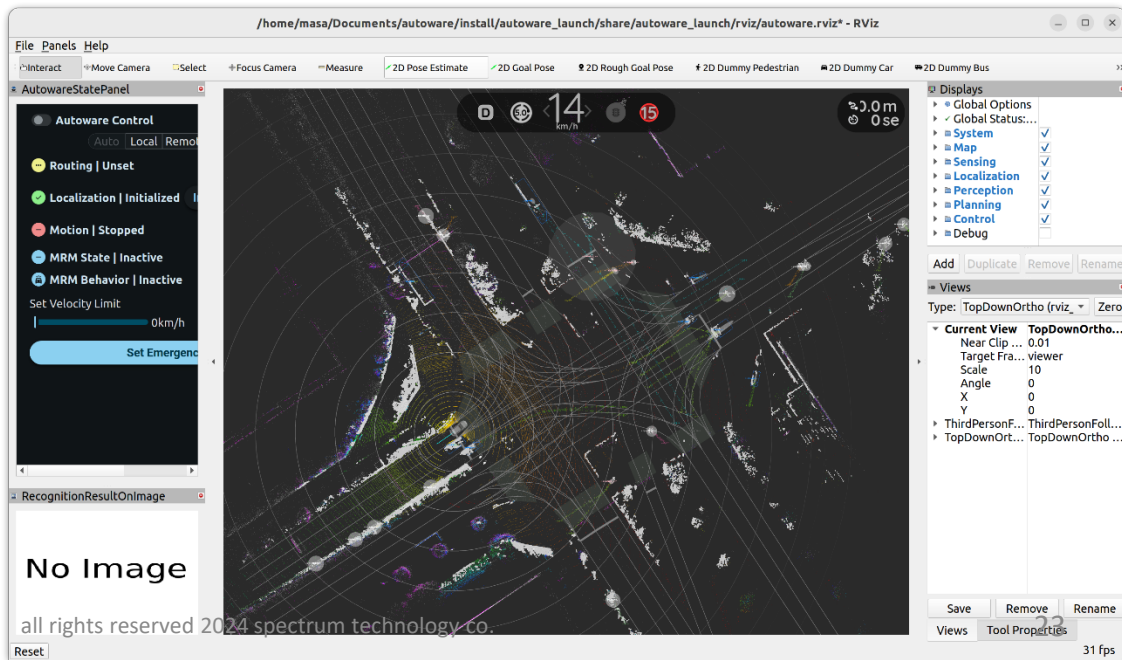
別端末で

```
ros2 bag play ~/autoware_map/sample-rosbag/ -r 0.2 -s sqlite3 デモ再生
```

Autoware表示

自動でデモ表示1

```
コマンド入力
$ source
/home/masa/Documents/autoware/install/setup.bash
$ ros2 launch autoware_launch
logging_simulator.launch.xml
map_path:=$HOME/autoware_map/sample-map-rosbag
vehicle_model:=sample_vehicle
sensor_model:=sample_sensor_kit
$ ros2 bag play ~/autoware_map/sample-rosbag/ -r 0.2 -s
sqlite3
```



## 4. Autoware開発

### ⑤ Scenario simulation (docker版)

[https://tier4.github.io/scenario\\_simulator\\_v2-docs/user\\_guide/RunWithDocker/](https://tier4.github.io/scenario_simulator_v2-docs/user_guide/RunWithDocker/)

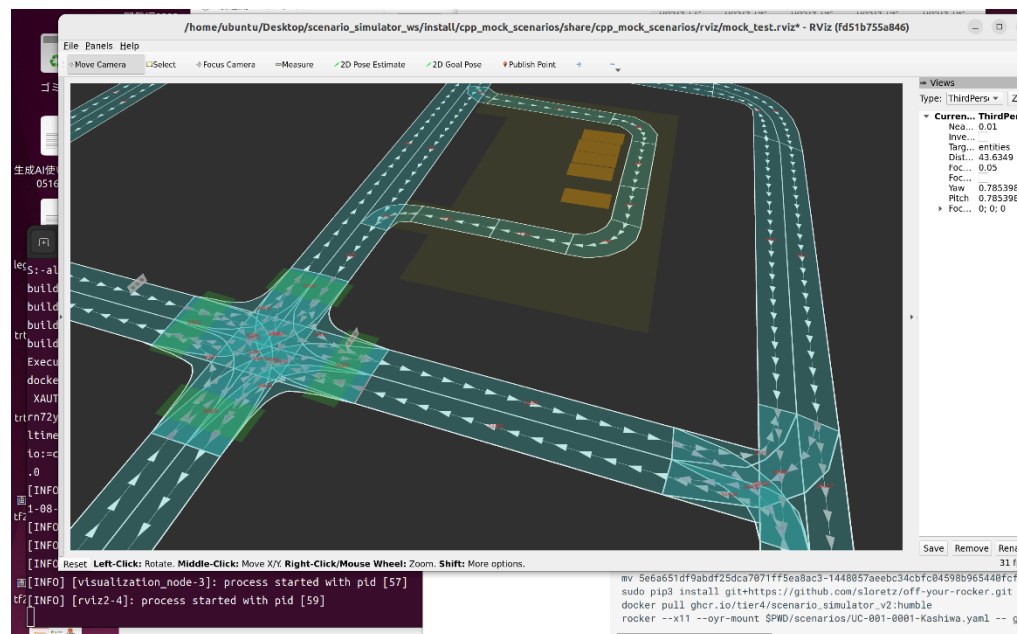
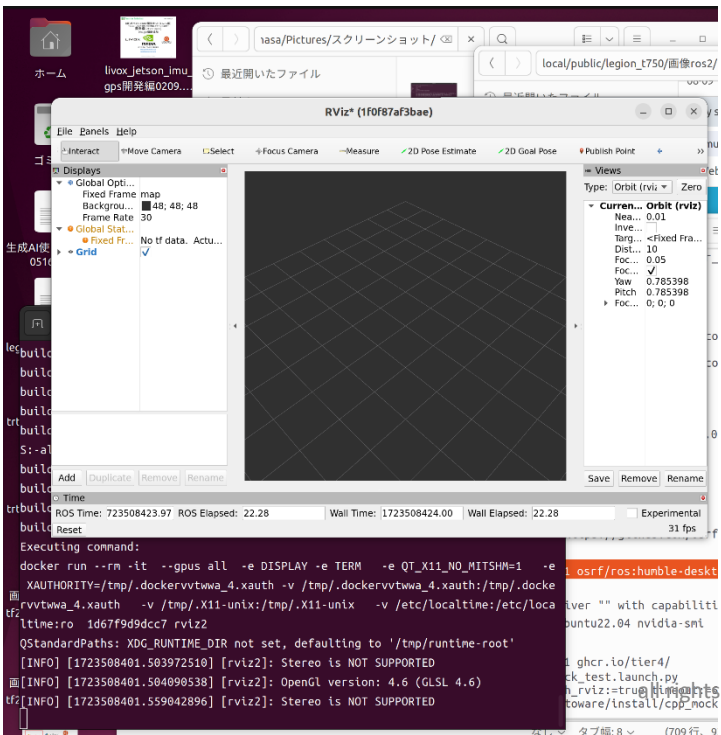
rviz2起動

```
rocker --nvidia --x11 osrf/ros:humble-desktop rviz2
```

scenario\_testデモ表示

```
rocker --nvidia --x11 ghcr.io/tier4/scenario_simulator_v2:humble ros2 launch cpp_mock_scenarios mock_test.launch.py scenario:=crashing_npc scenario:=traffic_simulation_demo launch_rviz:=true timeout:=60.0
```

```
コマンド入力
$ rocker --nvidia --x11 osrf/ros:humble-desktop rviz2
$ rocker --nvidia --x11
ghcr.io/tier4/scenario_simulator_v2:humble ros2 launch
cpp_mock_scenarios mock_test.launch.py
scenario:=crashing_npc scenario:=traffic_simulation_demo
launch_rviz:=true timeout:=60.0
```





## 4. Autoware開発

### ⑥ Digital twin simulation:AWSIM

<https://tier4.github.io/AWSIM/GettingStarted/QuickStartDemo/>

#### AWSIMデモ起動

```
cd /home/masa/Documents/mapfiles/AWSIM_v1.2.3  
./AWSIM.x86_64
```

#### autoware起動

```
source /home/masa/Documents/autoware/install/setup.bash  
ros2 launch autoware_launch e2e_simulator.launch.xml vehicle_model:=sample_vehicle  
sensor_model:=awsim_sensor_kit  
map_path:=/home/masa/autoware_map/nishishinjuku_autoware_map
```

#### コマンド入力

```
$ cd /home/masa/Documents/mapfiles/AWSIM_v1.2.3  
$ ./AWSIM.x86_64  
$ source  
/home/masa/Documents/autoware/install/setup.bash  
$ ros2 launch autoware_launch e2e_simulator.launch.xml  
vehicle_model:=sample_vehicle  
sensor_model:=awsim_sensor_kit  
map_path:=/home/masa/autoware_map/nishishinjuku_autoware_map
```

