

ATOM-Max Robot Showroom Solution Development Guide

Original Instructions

Issue: V1.0

Date: 2026-03-09

Copyright © SHENZHEN DOBOT CORP LTD 2026. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without the prior written consent of SHENZHEN DOBOT CORP LTD (hereinafter referred to as "DOBOT").

Disclaimer

To the maximum extent permitted by applicable law, the products described (including its hardware, software and firmware, etc.) in this document are provided AS IS, which may have flaws, errors or faults. DOBOT makes no warranties of any kind, express or implied, including but not limited to, merchantability, satisfaction of quality, fitness for a particular purpose and non-infringement of third party rights. In no event will DOBOT be liable for any special, incidental, consequential or indirect damages resulting from the use of our products and documents.

Before using our product, please thoroughly read and understand the contents of this document and related technical documents that are published online, to ensure that the robot is used on the premise of fully understanding the robot and related knowledge. Please use this document with technical guidance from professionals. Even if follow this document or any other related instructions, damages or losses may happen in the using process DOBOT shall not be considered as a guarantee regarding all security information contained in this document.

The user has the responsibility to make sure following the relevant practical laws and regulations of the country, in order that there is no significant danger in the use of the robot.

SHENZHEN DOBOT CORP LTD

Address: Room 1003, Building 2, Chongwen Garden, Nanshan iPark, Liuxian Blvd,
Nanshan District, Shenzhen, Guangdong Province, China

Website: www.dobot-robots.com

Preface

Purpose

This document explains how to use the DOBOT ATOM-Max robot for showroom tours.

Intended audience

This document is intended for:

- Customer
- Sales Engineer
- Installation and Commissioning Engineer
- Technical Support Engineer

Scope of application

DOBOT ATOM-Max

Related documents





Document	Description	Download link
DOBOT ATOM Series Humanoid Robot Quick Start	Introduces how to use the robot.	Contact technical support for access.
DOBOT ATOM Series Robot SDK Application Development Guide	Introduces the methods for robot SDK development.	Contact technical support for access.

Revision history

Date	Version	Revised content
2026-03-09	V1.0	The first release

Symbol conventions

The symbols that may be found in this document are defined as follows.

Sign	NOTE
 DANGER	Indicates a hazard with a high level of risk which, if not avoided, could result in death or serious injury.
 WARNING	Indicates a hazard with a medium level or low level of risk which, if not avoided, could result in minor or moderate injury, robot damage.
 NOTICE	Indicates a potentially hazardous situation which, if not avoided, could result in robot damage, data loss, or unanticipated result.
 NOTE	Provides additional information to emphasize or supplement important points in the main text.

Contents

Preface	ii
1. Product Introduction	1
1.1 Product Overview	1
1.2 Environment Setup	1
1.2.1 Environment Dependencies.....	1
1.2.2 Configure LiDAR IP Address	1
1.2.3 Connect to Control Software.....	3
2. Building SLAM	5
2.1 Mapping.....	5
2.2 Localization.....	5
2.3 Tour Guide	7
2.3.1 Execute Guided Tour	7
2.3.2 Modify Guided Tour	15
2.3.3 Guided Tour Example	17

1. Product Introduction

1.1 Product Overview

The ATOM-Max, developed by DOBOT, is the world's first embodied AI humanoid robot featuring "Dexterous Manipulation + Straight-Knee Walking". The robot features hollow wiring through all joints and a full-scale 1:1 humanoid arm design. It is equipped with 7-degree-of-freedom (DOF) industrial-grade bionic collaborative arms with $\pm 0.05\text{mm}$ repeatability, a humanoid linked head, and a five-finger dexterous hand (optional). Based on the embodied AI operation model and AI edge computing power, it enables "collection, training, and inference" for specific application scenarios.

ATOM-Max is suitable for continuous and repetitive work scenarios with unfixed equipment positions, multi-specification products, high operational similarity, and the need to pass through narrow spaces and make flexible turns. Typical applications include assembly and material handling in car factories, beverage preparation in coffee shops, and nighttime medicine retrieval in chain pharmacies.

1.2 Environment Setup

1.2.1 Environment Dependencies

System Environment

Development is recommended on Ubuntu 20.04; Mac and Windows systems are currently not supported. PC1 runs official services only and does not support development; development can only be performed on PC2.

- **OS:** Ubuntu 20.04 LTS
- **ROS1:** Python3 environment

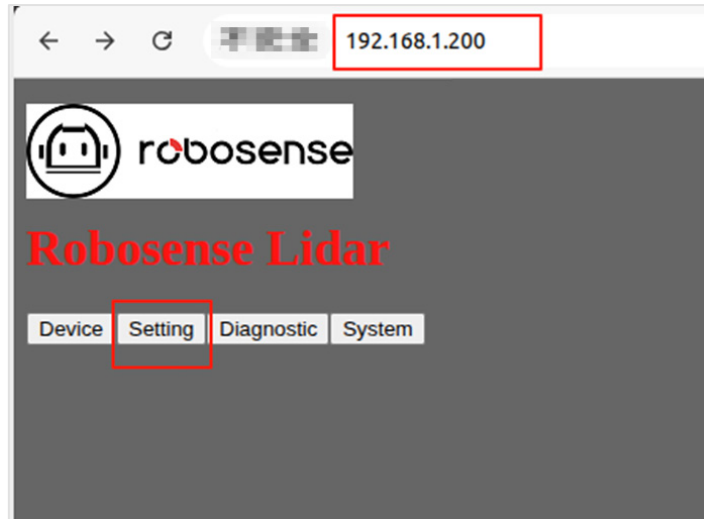
Network Environment

Connect the user's computer and the ATOM-Max switch to the same network. It is recommended that new users connect their computer to the ATOM-Max switch via an Ethernet cable and adapter, and set the network card for robot communication to the "192.168.8" segment. Experienced users can configure the network environment independently.

1.2.2 Configure LiDAR IP Address

The default IP address of the LiDAR is 192.168.1.200. During production, the IP address must be reconfigured to be on the same network segment as the robot, and the destination IP must be set to the robot's IP address to ensure the LiDAR can send data to the robot. Here takes the robot IP address "192.168.8.13" as an example.

Step 1 Set the PC used for configuration to the "192.168.1" segment. Then open a browser, enter "192.168.1.200", and click "Setting".



Step 2 Modify the IP address.

General Setting	
Device IP Address:	192.168.8.200
Device IP Mask:	255.255.255.0
Device IP Gateway:	192.168.8.1
Destination IP Address:	192.168.8.13
MSOP Port Number(1025~65535):	6699
DIFOP Port Number(1025~65535):	7788
Return Mode:	Strongest
Time Synchronization Source:	PTP-GPTP
PTP Domain Number(0~127):	0
Respond To PeerDelayRequest:	OFF
No Leap Second:	OFF
Sync Timeout(1~255s):	5
Unlock To Lock Threshold(1~255ms):	1
Lock To Unlock Threshold(1~255ms):	20
Operation Mode:	High-Performance
Phase Lock Setting(0~360):	0 DEG
GPS Baud Rate:	9600
Imu Ctrl:	ON
Imu Port Number(1025~65535):	6688
Imu Output Rate:	200Hz
Accel Range:	[-4g, 4g]
Gyro Range:	[-500, 500]dps
Imu LPF Config(0~255):	0
Restore Default:	OFF
<input type="button" value="Save"/>	

Modify **Device IP Address** to 192.168.8.200.

Modify **Device IP Gateway** to 192.168.8.1.

Modify **Destination IP Address** to 192.168.8.13.

Step 3 After modification, click "Save" to save the settings.

Step 4 Set the PC used for configuration to the "192.168.8" segment.

1.2.3 Connect to Control Software

Please contact DOBOT technical support to obtain the installation software "rsview.zip" for the host computer.

Step 1 Extract it to the home directory on PC2, open a terminal and navigate to the "rsview" directory.

```
cd rsview
```

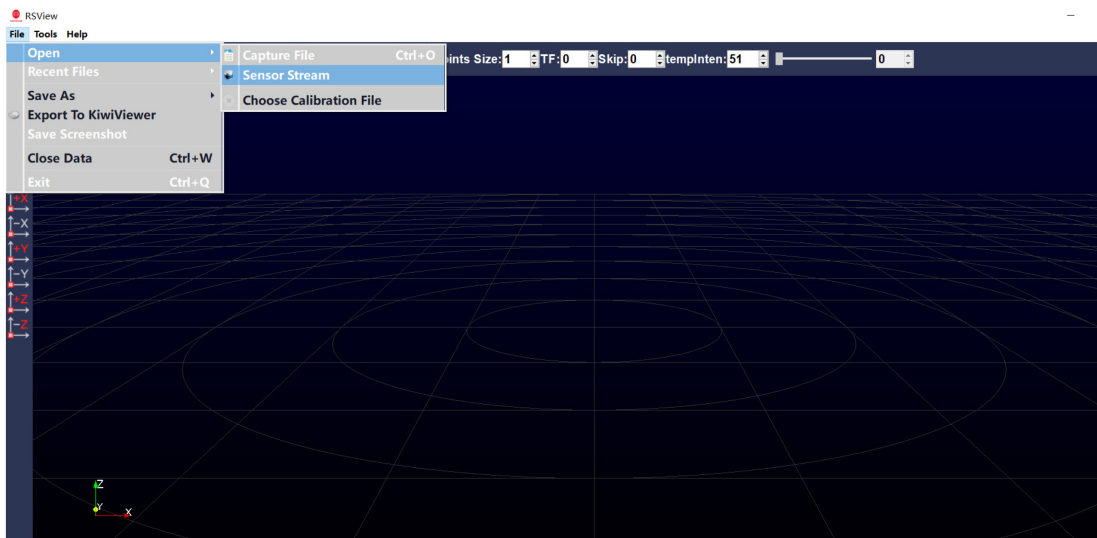
Step 2 Exit the conda environment.

```
conda deactivate
```

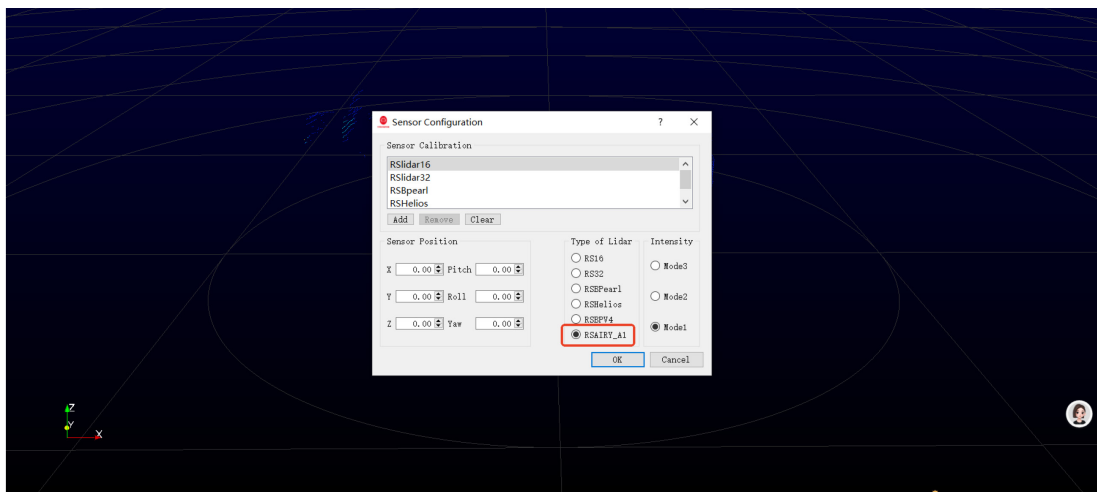
Step 3 Run the control software.

```
./run_rsview.sh
```

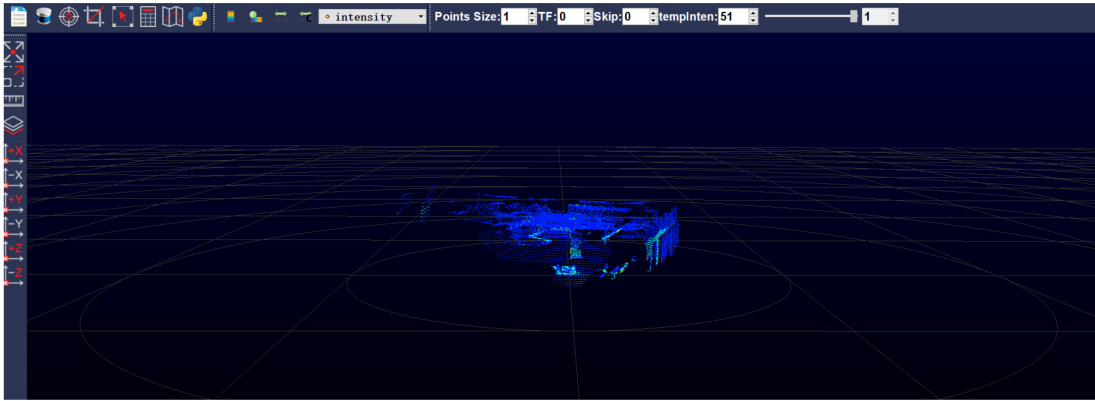
Step 4 After entering the Ubuntu system password, the control software interface will appear. Select "File -> Open -> Sensor Stream".



Step 5 Select "RSAIRY_A1" for "Type of Lidar", and click "OK".

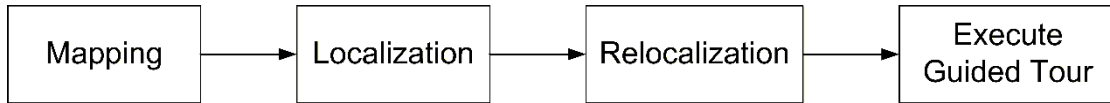


Step 6 The point cloud should now be visible, indicating a successful connection.



2. Building SLAM

The current navigation is developed based on ROS1, with each functional interface on ROS communication mechanisms. The workflow is as follows:



2.1 Mapping

Start Mapping

Communication method	Service
Name	/start_mapping
Message type	std_srvs/Trigger
Message definition	--- bool success #Whether successful string message #No content

Stop Mapping

Communication method	Service
Name	/stop_mapping
Message type	type_define/FinishMapping
Message definition	bool save_map #Whether to save the map string map_name #Map name --- bool success #Whether successful

2.2 Localization

Load the Map

Communication method	Service
Name	/load_map
Message type	type_define/LoadMap
Message definition	string map_name #Map name

	bool success #Whether successful

Initialize Pose (Relocalization)

Communication method	Topic
Name	/initialpose
Message type	geometry_msgs/PoseWithCovarianceStamped
Message definition	std_msgs/Header header #Message header (ignore) geometry_msgs/PoseWithCovariance pose #Robot pose, approximate pose is acceptable

Stop Localization

Communication method	Service
Name	/stop_localization
Message type	std_srvs/Trigger
Message definition	--- bool success #Whether successful string message #No content

Get Robot Pose

Communication method	Topic
Name	/robot_pose_2d
Message type	geometry_msgs/PoseWithCovarianceStamped
Message definition	std_msgs/Header header #Message header geometry_msgs/PoseWithCovariance pose #Robot pose and covariance

2.3 Tour Guide

2.3.1 Execute Guided Tour

Open the terminal and enter the following command to configure environment variables. Subsequent ROS commands can be auto-completed by pressing the Tab key.

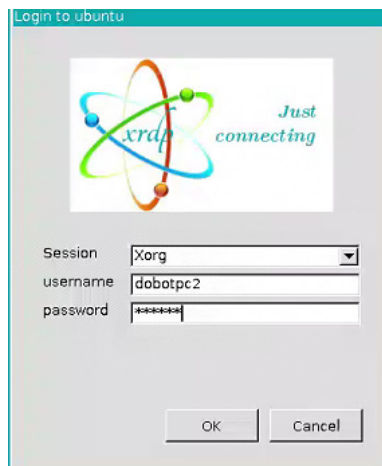
```
echo "source ~/Traverse/install/setup.bash" >> ~/.bashrc
```

Step 1 Access the Control Software

Connect to the control software via WiFi or Ethernet cable as follows:

- WiFi mode (recommended): Connect your computer's WiFi to the robot's WiFi network (SSID: ATOMMX+SN code, default password: 12345678). Go to WiFi settings and manually modify the IP address to 192.168.8.xxx (xxx is user-defined), and change the subnet mask to 255.255.255.0.
- Ethernet connection: Use an Ethernet cable to connect your computer to the LAN1 port on the ATOM-Max back panel. Go to Ethernet settings and manually modify the IP address to 192.168.8.xxx (xxx is user-defined), and change the subnet mask to 255.255.255.0.

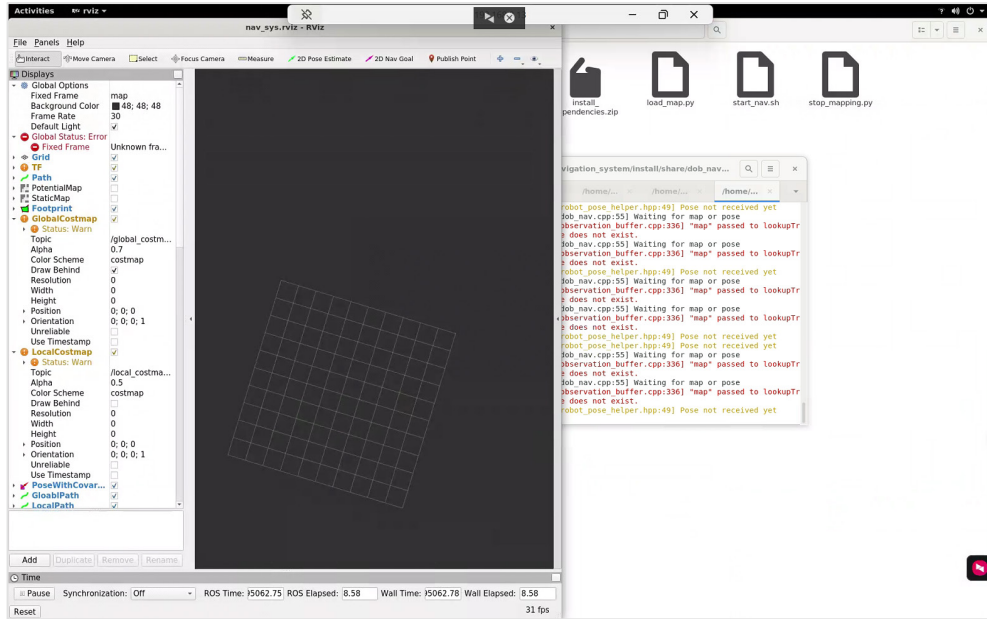
After successfully connecting to the network, open a remote desktop connection on your computer and enter the following address: 192.168.8.13:3390. Once connected, enter username: dobotpc2, password: 123456.



Step 2 Start the Program

Extract the algorithm package to the "/home/xxx" directory (xxx is the Ubuntu system username), then open the terminal and enter the following command. RViz appearing indicates startup is complete (RViz is enabled by default).

```
cd dob_navigation_system
./start_nav.bash
```

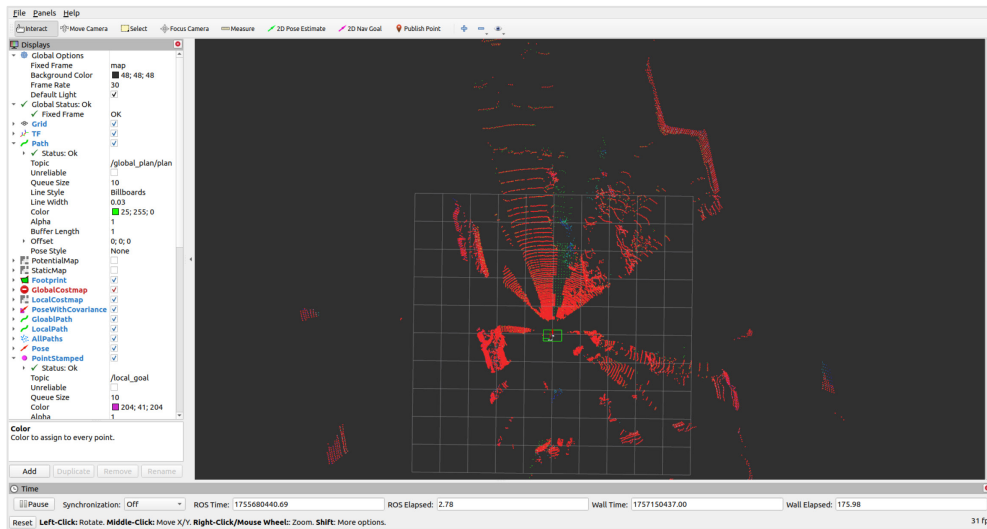


Step 3 Start Mapping

In the terminal where "start_nav.bash" is running, add a new terminal in the upper-left corner and enter the following service command.

```
rosservice call /start_mapping "{}"
```

After the terminal returns a response, the LiDAR will start scanning the forward field of view (360° unobstructed area) in RViz.



NOTICE

If the robot is equipped with a face cover, remove it. Enable the robot and enter the home posture (zero-point posture), ensuring the head is powered on and maintains its posture.

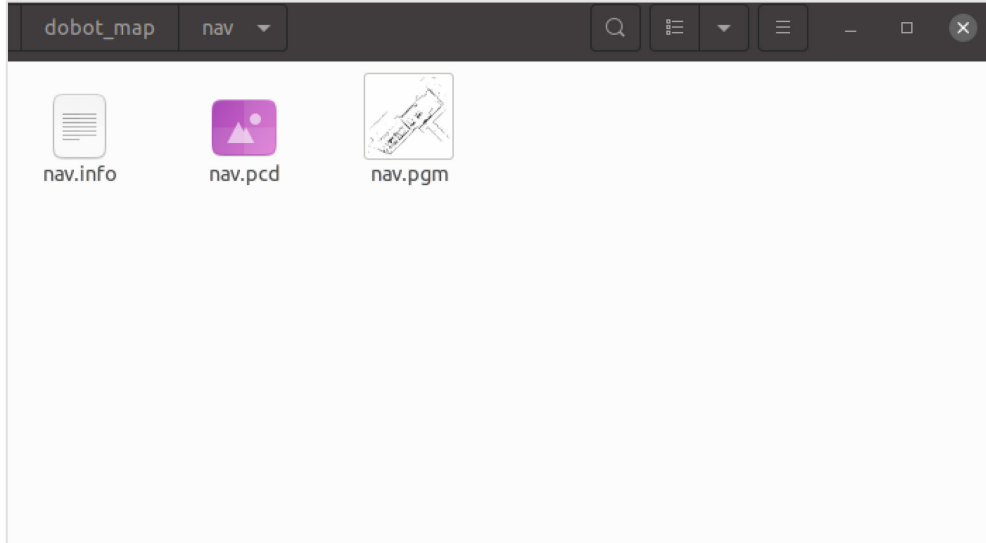
Step 4 Stop Mapping

Enter the following command in the terminal to call the stop-mapping interface, or call it through your own code.

If you need to save the map, set "save_map" to "true" and enter a custom map name for "map_name". The map saving process may take a few seconds to over ten seconds.

```
rosservice call /stop_mapping "save_map: false
map_name: 'your_map_name'"
```

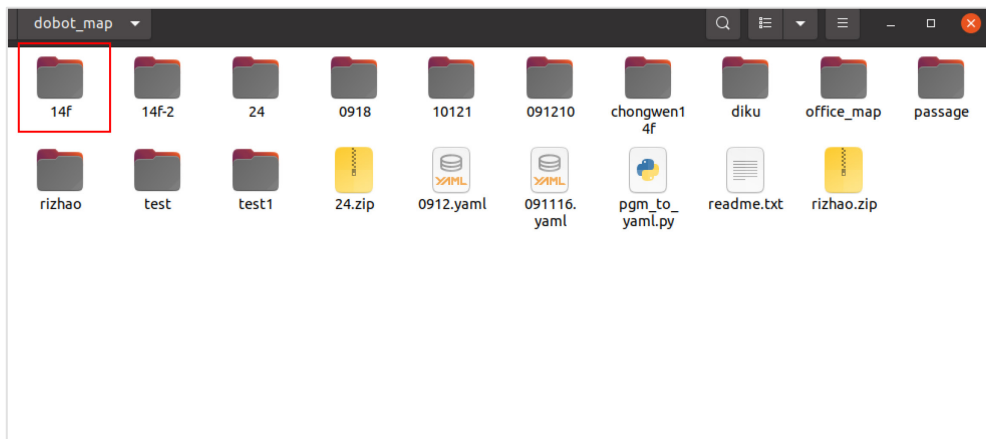
After the map is saved, a folder with the corresponding map name will appear in the "/home/dobotpc2/dobot_map" directory, containing the following three files.



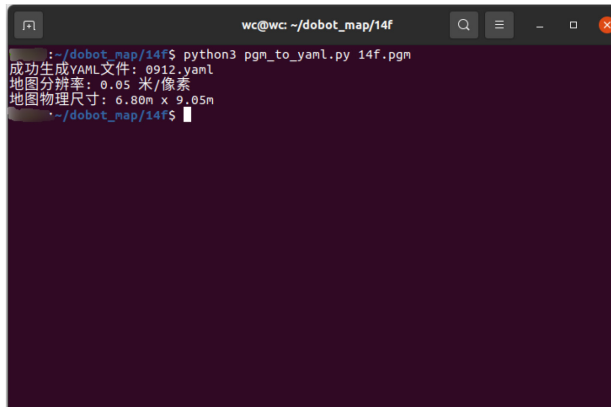
Step 5 Process the Map

Maps saved via "start_mapping" are in ".pgm" format, which cannot be directly read by the navigation algorithm. The map format needs to be processed to be compatible with the navigation algorithm.

Copy "pgm_to_yaml.py" to the custom map folder under "/dobot_map".



```
python3 pgm_to_yaml.py 14.pgm #Where 14 is the custom map name
```



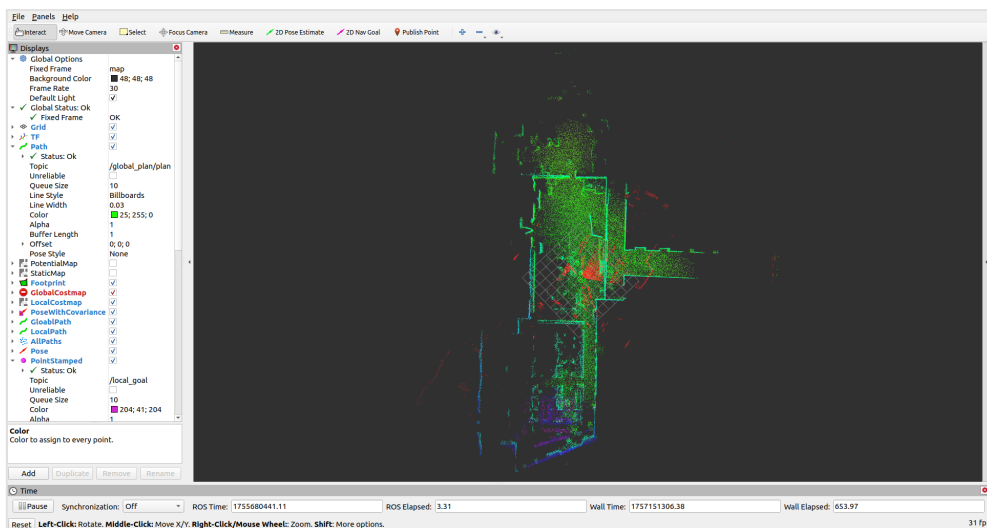
Step 6 Load the Map

The map can be loaded via service or program, as shown in the following steps.

➤ Load the Map (Service Method)

The service for loading the map is "load_map". Enter the desired map name in the "map_name" field.

```
rosservice call /load_map "string_value:: 'your_map_name'"
```



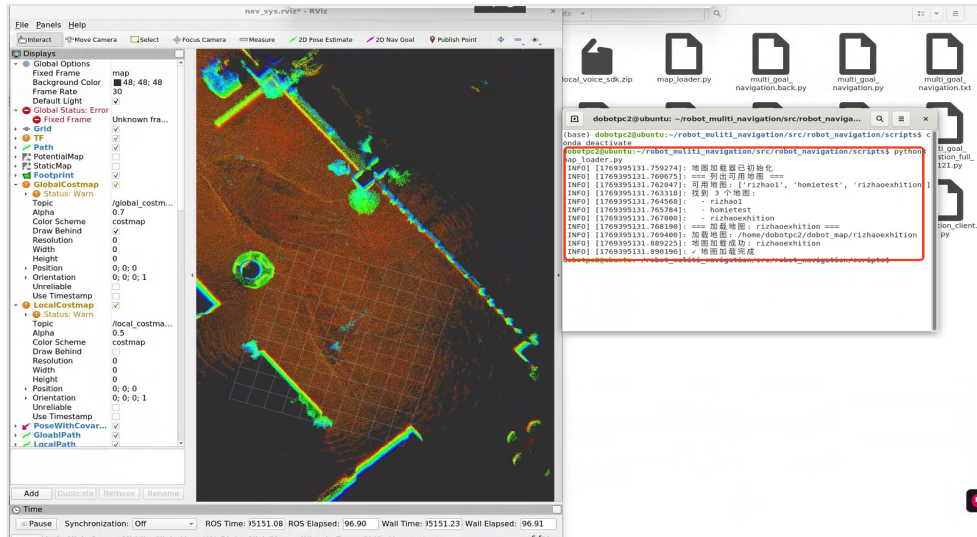
➤ Load the Map (Program Method)

Navigate to the "robot_muliti_navigation" path and run the "map_loader.py" program.

```
conda deactivate #Exit conda
cd robot_muliti_navigation/src/robot_navigation/scripts
python3 map_loader.py
```

⚠ NOTICE

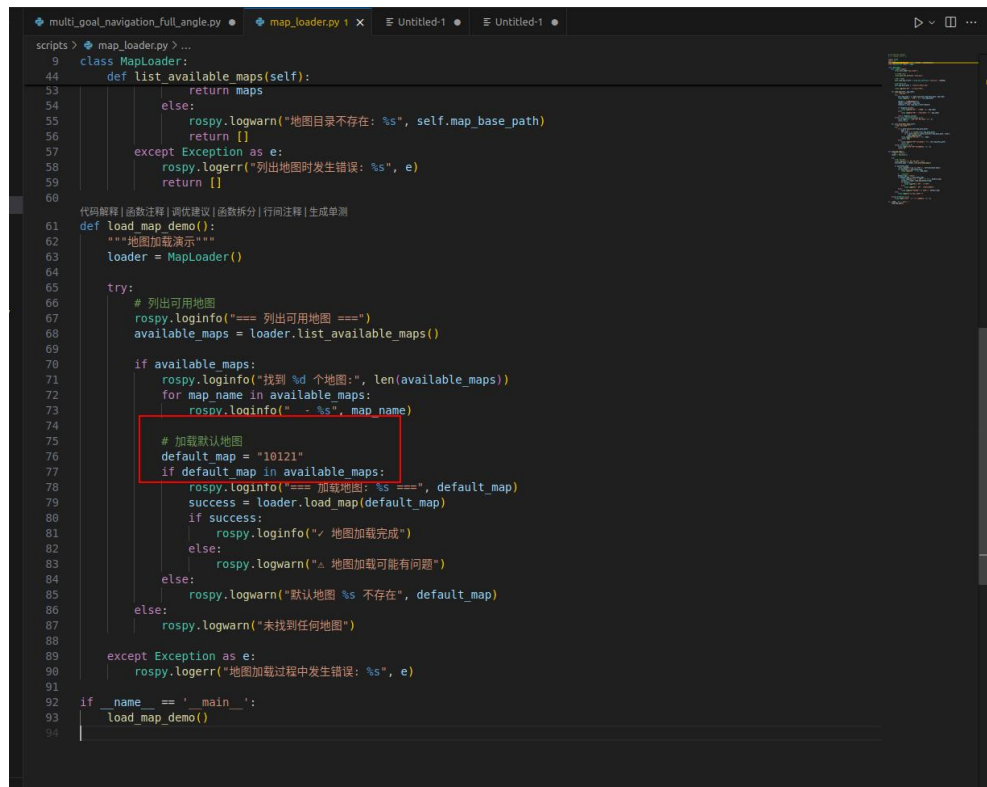
Exiting conda is necessary; otherwise, all Python scripts in this workspace will fail to run.



The terminal will load all maps under the "dobot_maps" directory. Successful RViz loading indicates the program ran successfully.

To modify the default map loaded by "map_loader.py", edit it using Vim.

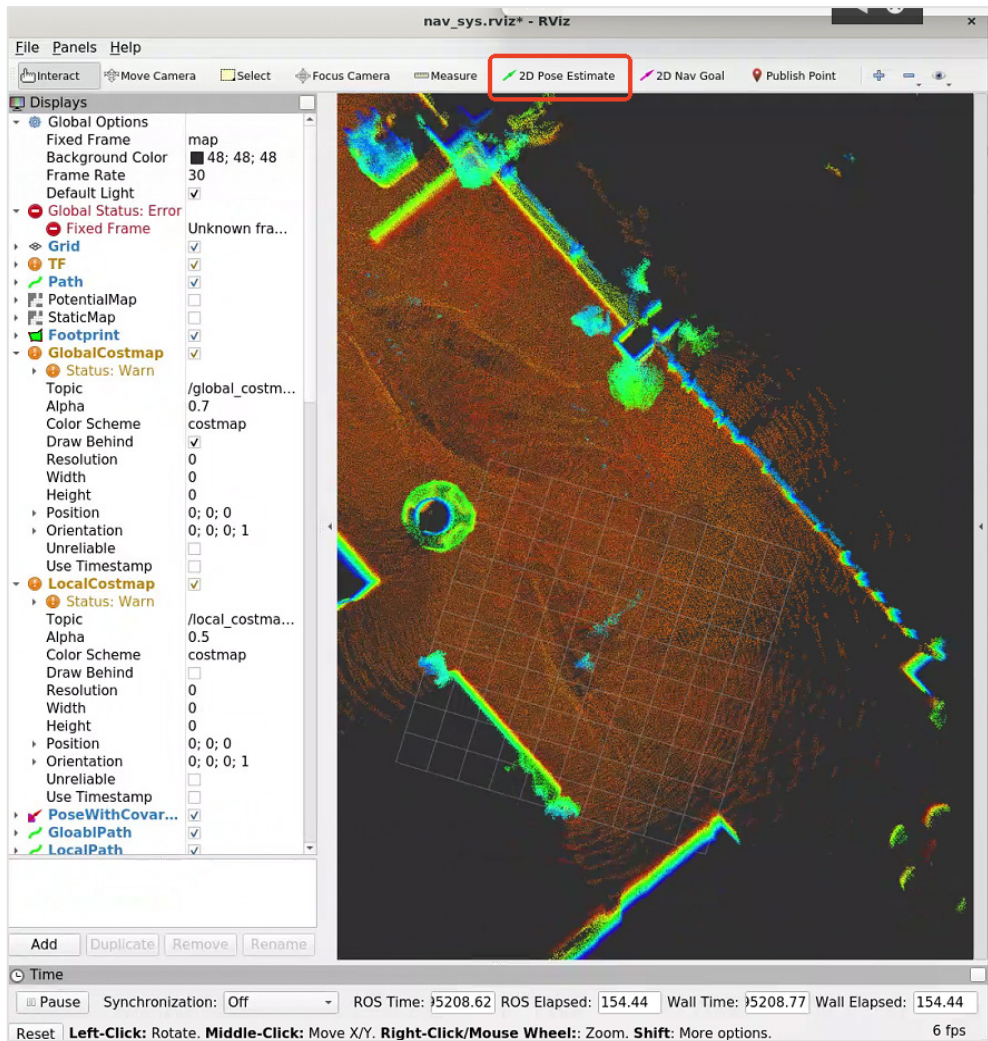
vim map_loader.py



Change the parameter after "default_map" to your custom map name.

Step 7 Relocalization

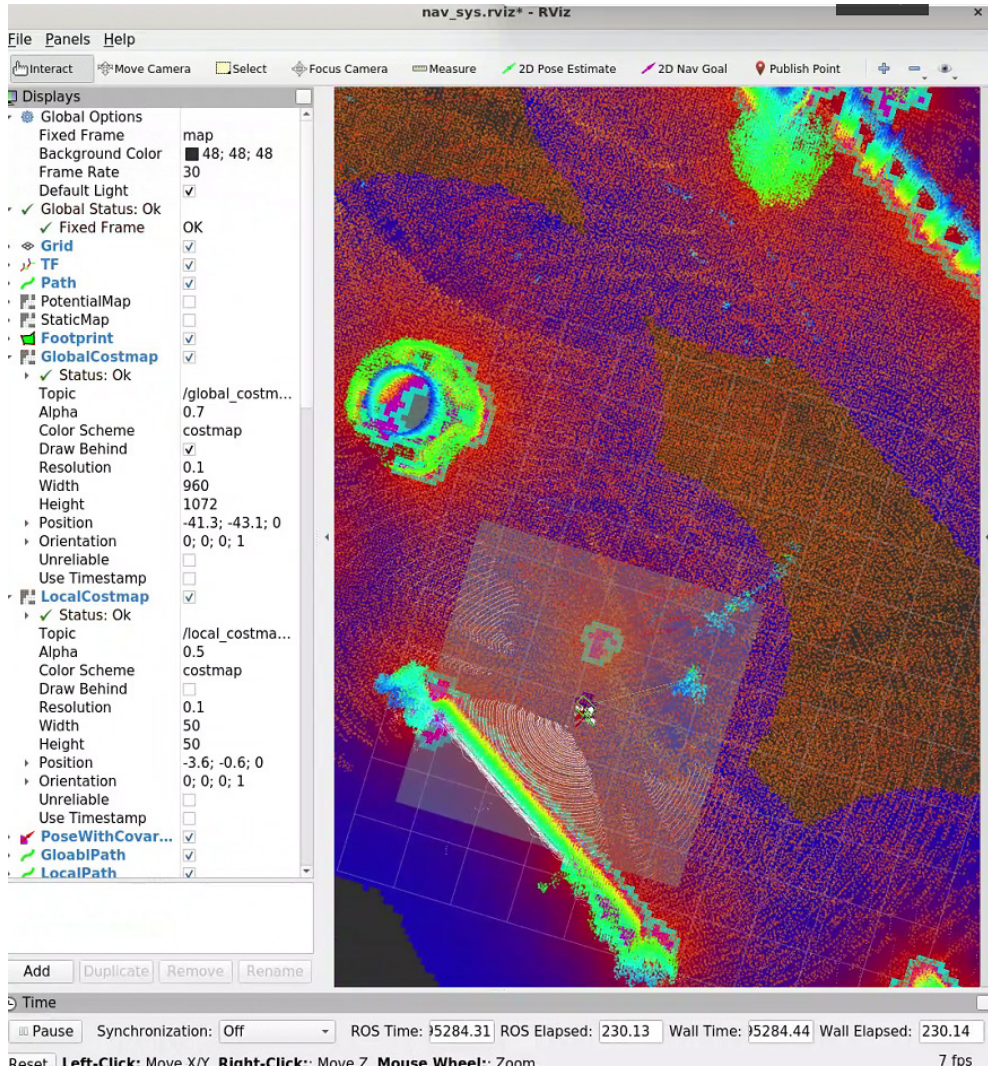
After loading the map, click "2D Pose Estimate" in the RViz status bar to perform relocalization.



⚠ NOTICE

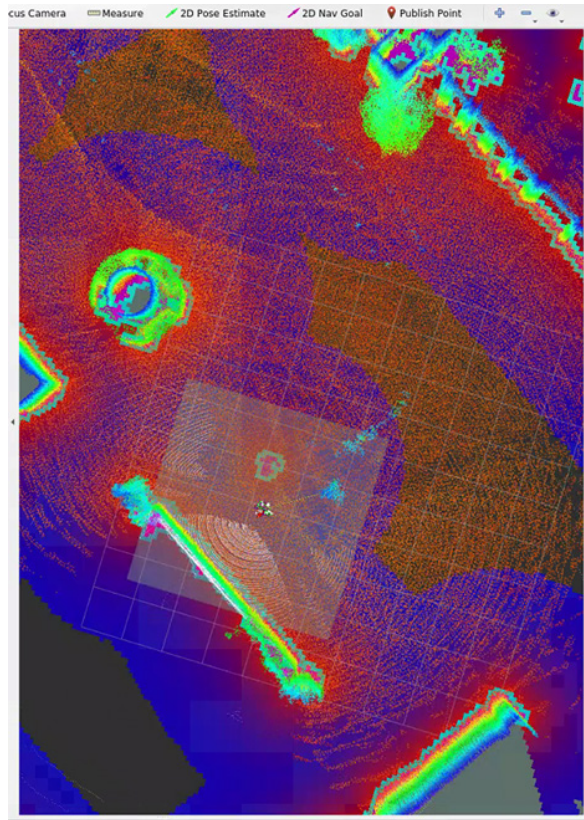
During robot relocalization, avoid laser-penetrable areas (e.g., glass doors) and prioritize locations with distinctive features for pose matching.

After successful relocalization, once the LiDAR is loaded onto the map, the map becomes dynamic, and the LiDAR's forward beam becomes clearly visible.



Step 8 Execute Guided Tour

In the terminal under the path "robot_muliti_navigation/src/robot_navigation/scripts", run "demo.py" (or "xxx.py", the Python file name can be self-defined). When the terminal shows a loading status, it indicates that the tour program is running.



In RViz, you can see the path planned by SLAM for the first point; subsequent points will be executed automatically.

NOTICE

- After the program starts running, you need to press **LT+X** on the remote controller to open the RPC channel. Once the RPC channel is open, the robot is controlled via RPC (remote controller is disabled).
- During the tour, if an accident or emergency occurs, press **LT+Y** on the remote controller to close the RPC channel and switch back to remote controller operation.

2.3.2 Modify Guided Tour

To modify the tour file, refer to the following commands and instructions.

```
vim demo.py
```

```
def main():
    try:
        # 创建导航客户端
        nav_client = MultiGoalNavigation()

        # 定义目标点序列
        waypoints = [
            {
                'x': -4.246,
                'y': 0.344,
                'yaw': 157.475, # 角度制
                'name': '开始区 ',
                'stay_time': 5,
                'navigation_timeout': 60,
                'action_program': 'bridge_test2',
                'upper_limb_enable': True,
                'explanation_text': '尊敬的各位领导, 上午好。我是越疆机器人Atom, 欢迎各位领导莅临越疆科技。请跟我一起走进越疆。',
                'max_retries': 1,
                'retry_delay': 3
            },
            {
                'x': -7.982,
                'y': -11.152,
                'yaw': 78.699, # 角度制
                'name': '大屏幕 ',
                'stay_time': 5,
                'navigation_timeout': 60,
                'action_program': 'bridge_test3',
                'upper_limb_enable': True,
                'explanation_text': '请各位领导移步至大屏幕前, 观看企业宣传片',
                'max_retries': 1, # 新增: 重试配置
                'retry_delay': 3
            }
        ],
```

Parameter definitions:

- **x, y, yaw:** "x, y" refers to the x and y coordinates of the target point, "yaw" refers to the robot's yaw angle (rotation angle). Refer to ["Get x, y, yaw"](#) for these values.
- **name:** Task name for the target point.
- **stay_time:** Dwell time after reaching the target point and completing the explanation thread.
- **navigation_timeout:** Navigation timeout threshold. The program will mark navigation as timed out if $t > \text{timeout}$ (set this value reasonably for your route).
- **action_program:** The explanation action to be executed.
- **explanation_text:** The explanation script/text.
- **max_retry:** The maximum number of navigation retries.
- **retry_try:** The interval between each re-navigation attempt after a navigation timeout is detected.

NOTICE

Do not modify "upper_limb_enable" or "action_program" without technical support approval. Unauthorized changes may damage the robot's motors.

Get x, y, yaw

In the terminal under the SLAM's "dob_navigation_system" directory, use topic to perform queries.

```
rostopic echo /robot_pose_2d
```

When subscribing to the robot's current coordinates via topic, the robot must be moved to the target point. The topic subscription provides real-time coordinate updates.

```
wc@wc: ~/dob_navigation_system
wc@wc:~/dob_navigation_system$ rostopic echo /robot_pose_2d
^Cwc@wc:~/dob_navigation_system$ rostopic echo /robot_pose_2d
x: -1.119944818981664
y: 1.3263611847885315
theta: -0.8180834958583876
---
x: -1.116341653969285
y: 1.3294411360650726
theta: -0.8163932169015724
---
x: -1.121130754341811
y: 1.33190392351715
theta: -0.8185579476465437
---
x: -1.130082431000615
y: 1.3357463212357752
theta: -0.8183903656480919
---
x: -1.1395092763627417
y: 1.3407463979103653
theta: -0.8105463295878826
---
x: -1.142375277189075
y: 1.3525289892225836
```

Unit Conversion

The yaw data obtained from "rostopic echo /robot_pose_2d" is in radians, which cannot be directly read by the "move_base" algorithm. Therefore, the radian data needs to be converted to degrees using the following method.

```
cd robot_muliti_navigation/src/robot_navigation/scripts #Skip if already in this
directory
```

```
python3 rand_to_angle.py
```

Enter the radian value captured from the topic

```
/模板/defeat/src/robot_navigation/scripts
模板/defeat/src/robot_navigation/scripts$ python3 rand_to_angle.py
请输入弧度值: 1.4328
1.4328 弧度 = 82.09 度
~/模板/defeat/src/robot_navigation/scripts$
```

2.3.3 Guided Tour Example

Below is an example with all programs and tour content unmodified.

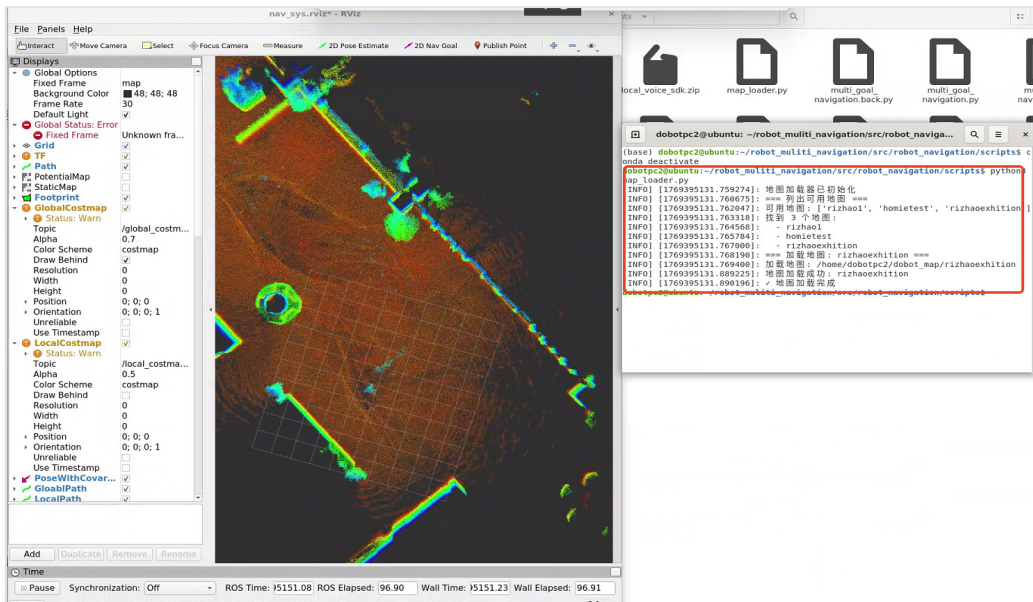
1. Start the Program

Extract the algorithm package to the "home" directory, open a terminal, and enter the following commands. RViz appearing indicates startup is complete (RViz is enabled by default).

```
cd dob_navigation_system
./start_nav.bash
```

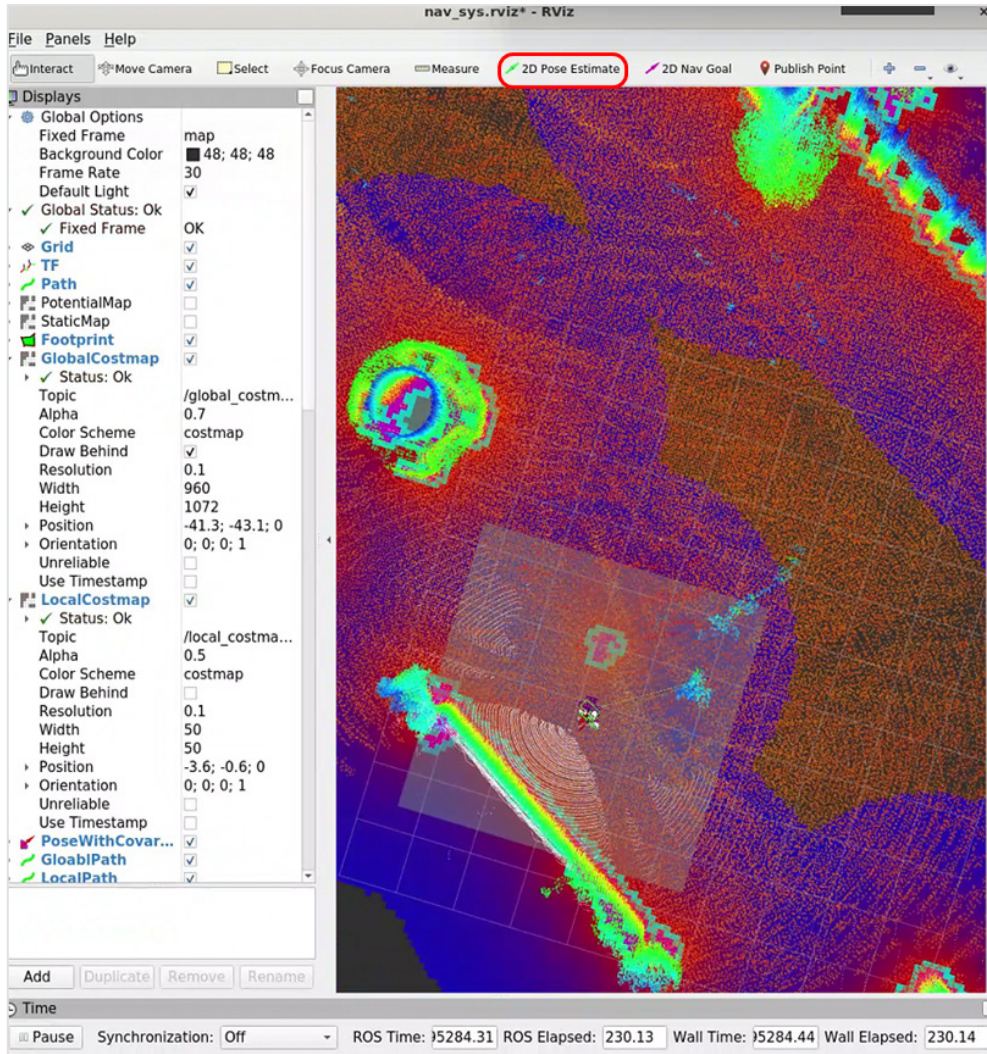
2. Load the Map

```
conda deactivate #Exit conda
cd robot_multit_navigation/src/robot_navigation/scripts
python3 map_loader.py
```



3. Relocalization

In the RViz status bar, click "2D Pose Estimate" to perform relocalization.



4. Run demo.py

```
python3 demo.py
```

