Multi-car Navigation2 fast relocation navigation

Note: The virtual machine needs to be in the same LAN as the car, and the ROS_DOMAIN_ID needs to be consistent. You can check [Must-read before use] to set the IP and ROS_DOMAIN_ID on the board.

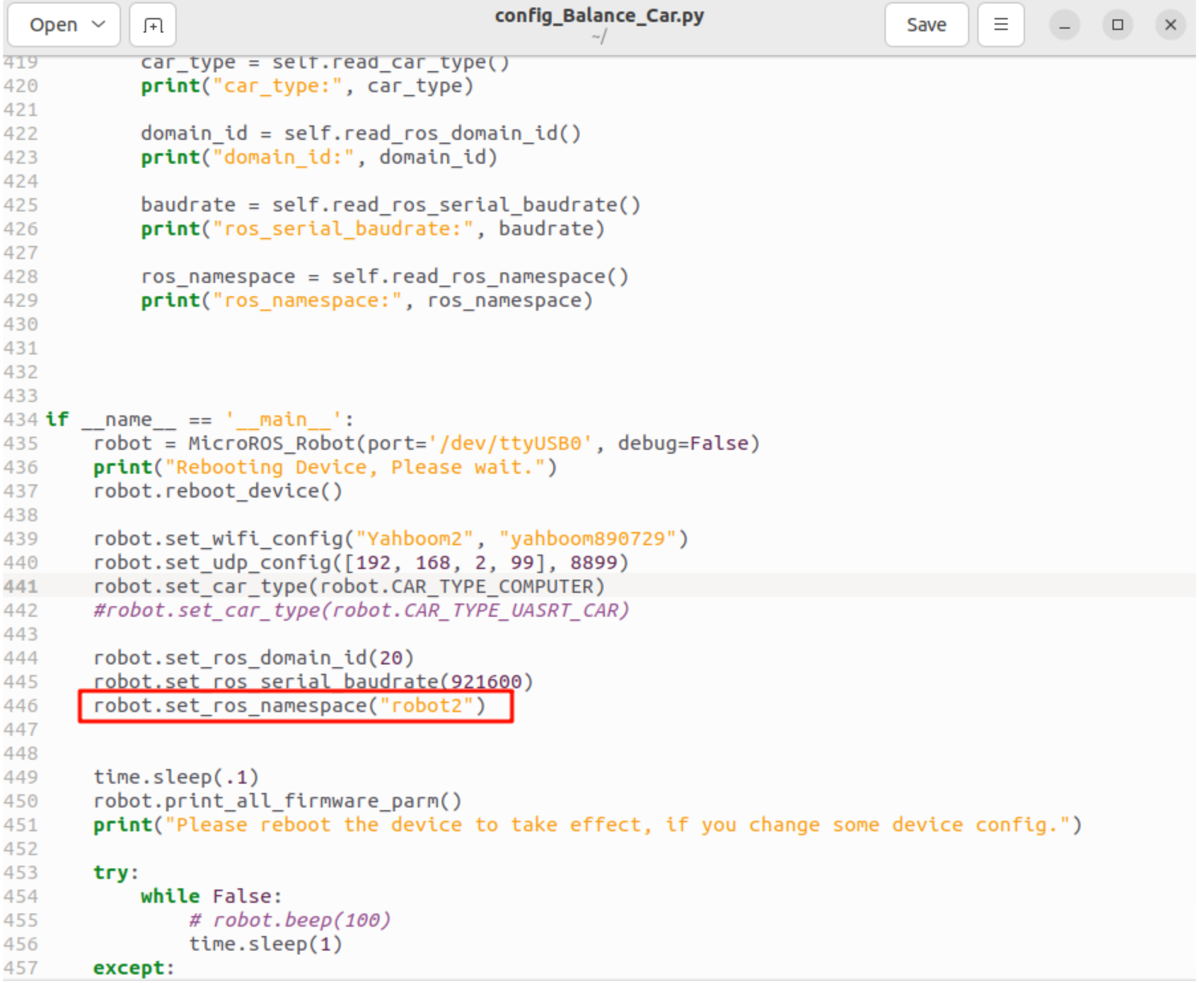
1. Program function description

After the program is started, you can give the target points of the two cars in rviz. After receiving the command, the two cars will calculate the path based on their own posture and move to their destinations.

```
Note: Before running the program, the car needs to be restarted in a standing
position to ensure that all sensors are reset
```

2. Basic settings for multi-machine functions

Take two cars as an example. It is recommended to use two computers with matching virtual machines, change the config_robot.py files respectively, set robot.set_ros_namespace() to robot1 and robot2 respectively; set robot.set_udp_config() to the IP addresses of the two virtual machines respectively, and **the ROS_DOMAIN_ID of the two cars and the ROS_DOMAIN_ID of the virtual machine need to be set to the same**. Then open the terminal in the /home/yahboom directory, enter `sudo python3 config_Balance_Car.py` to run this program (the rest of the programs other than running multiple cars need to be changed back and re-run this program).

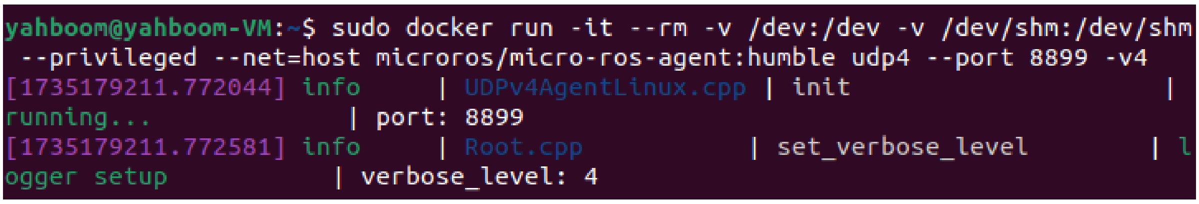


```
        car_type = self.read_car_type()
        print("car_type:", car_type)

        domain_id = self.read_ros_domain_id()
        print("domain_id:", domain_id)

        baudrate = self.read_ros_serial_baudrate()
        print("ros_serial_baudrate:", baudrate)

        ros_namespace = self.read_ros_namespace()
        print("ros_namespace:", ros_namespace)




if __name__ == '__main__':
    robot = MicroROS_Robot(port='/dev/ttyUSB0', debug=False)
    print("Rebooting Device, Please wait.")
    robot.reboot_device()

    robot.set_wifi_config("Yahboom2", "yahboom890729")
    robot.set_udp_config([192, 168, 2, 99], 8899)
    robot.set_car_type(robot.CAR_TYPE_COMPUTER)
    #robot.set_car_type(robot.CAR_TYPE_UASRT_CAR)

    robot.set_ros_domain_id(20)
    robot.set_ros_serial_baudrate(921600)
    robot.set_ros_namespace("robot2")


    time.sleep(.1)
    robot.print_all_firmware_parm()
    print("Please reboot the device to take effect, if you change some device config.")

    try:
        while False:
            # robot.beep(100)
            time.sleep(1)
    except:
```

3. Start and connect the agent

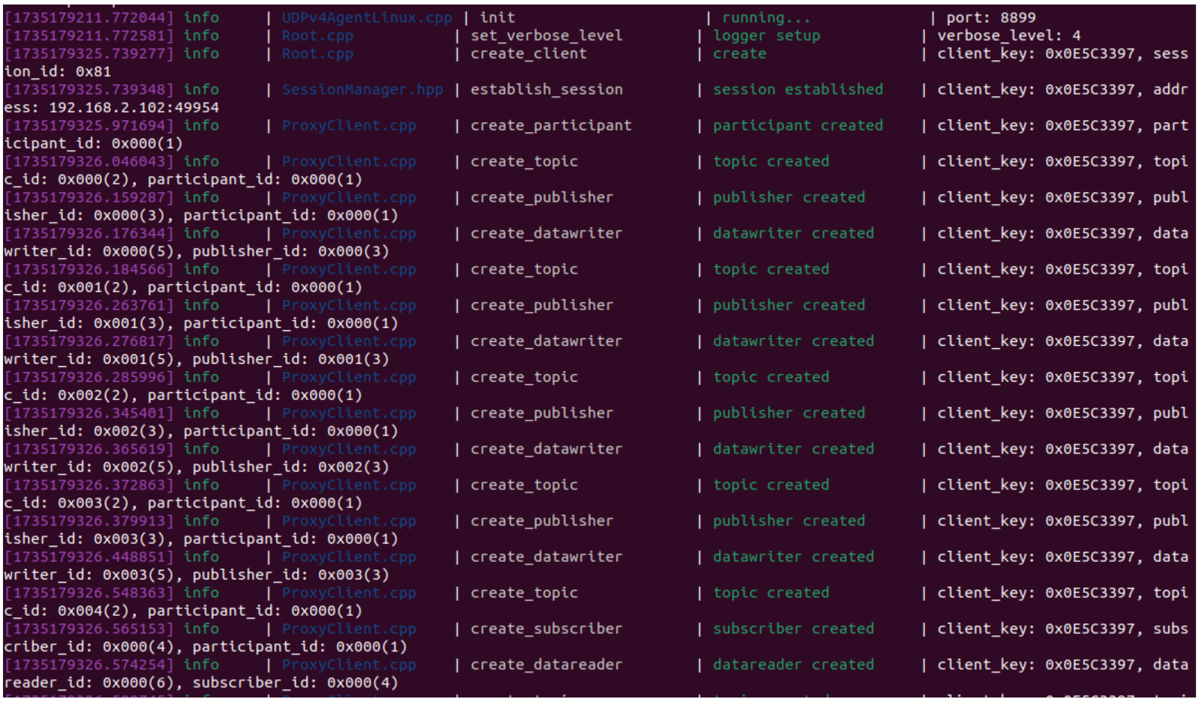
Take the matching virtual machine as an example. In the two virtual machines, enter the following command to start the agent of each car.

```
sudo docker run -it --rm -v /dev:/dev -v /dev/shm:/dev/shm --privileged --
net=host microros/micro-ros-agent:humble udp4 --port 8899 -v4
```



```
yahboom@yahboom-VM:~$ sudo docker run -it --rm -v /dev:/dev -v /dev/shm:/dev/shm
 --privileged --net=host microros/micro-ros-agent:humble udp4 --port 8899 -v4
[1735179211.772044] info     | UDPv4AgentLinux.cpp | init                     |
running...             | port: 8899
[1735179211.772581] info     | Root.cpp            | set_verbose_level        | l
ogger setup              | verbose_level: 4
```

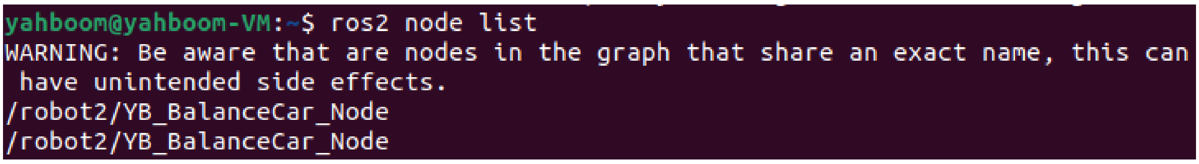
Then, turn on the switches of the two cars and wait for the two cars to connect to their respective agents. If the connection is successful, the terminal display is as shown in the figure below.



```
[1735179211.772044] info     | UDPv4AgentLinux.cpp | init                     | running...             | port: 8899
[1735179211.772581] info     | Root.cpp            | set_verbose_level        | logger setup           | verbose_level: 4
[1735179325.739277] info     | Root.cpp            | create_client            | create                 | client_key: 0x0E5C3397, sess
ion_id: 0x81
[1735179325.739348] info     | SessionManager.hpp | establish_session        | session established    | client_key: 0x0E5C3397, addr
ess: 192.168.2.102:49954
[1735179325.971694] info     | ProxyClient.cpp     | create_participant       | participant created    | client_key: 0x0E5C3397, part
icipant_id: 0x000(1)
[1735179326.046043] info     | ProxyClient.cpp     | create_topic             | topic created          | client_key: 0x0E5C3397, topi
c_id: 0x000(2), participant_id: 0x000(1)
[1735179326.159287] info     | ProxyClient.cpp     | create_publisher         | publisher created      | client_key: 0x0E5C3397, publ
isher_id: 0x000(3), participant_id: 0x000(1)
[1735179326.176344] info     | ProxyClient.cpp     | create_datawriter        | datawriter created     | client_key: 0x0E5C3397, data
writer_id: 0x000(5), publisher_id: 0x000(3)
[1735179326.184566] info     | ProxyClient.cpp     | create_topic             | topic created          | client_key: 0x0E5C3397, topi
c_id: 0x001(2), participant_id: 0x000(1)
[1735179326.263761] info     | ProxyClient.cpp     | create_publisher         | publisher created      | client_key: 0x0E5C3397, publ
isher_id: 0x001(3), participant_id: 0x000(1)
[1735179326.276817] info     | ProxyClient.cpp     | create_datawriter        | datawriter created     | client_key: 0x0E5C3397, data
writer_id: 0x001(5), publisher_id: 0x001(3)
[1735179326.285996] info     | ProxyClient.cpp     | create_topic             | topic created          | client_key: 0x0E5C3397, topi
c_id: 0x002(2), participant_id: 0x000(1)
[1735179326.345401] info     | ProxyClient.cpp     | create_publisher         | publisher created      | client_key: 0x0E5C3397, publ
isher_id: 0x002(3), participant_id: 0x000(1)
[1735179326.365619] info     | ProxyClient.cpp     | create_datawriter        | datawriter created     | client_key: 0x0E5C3397, data
writer_id: 0x002(5), publisher_id: 0x002(3)
[1735179326.372863] info     | ProxyClient.cpp     | create_topic             | topic created          | client_key: 0x0E5C3397, topi
c_id: 0x003(2), participant_id: 0x000(1)
[1735179326.379913] info     | ProxyClient.cpp     | create_publisher         | publisher created      | client_key: 0x0E5C3397, publ
isher_id: 0x003(3), participant_id: 0x000(1)
[1735179326.448851] info     | ProxyClient.cpp     | create_datawriter        | datawriter created     | client_key: 0x0E5C3397, data
writer_id: 0x003(5), publisher_id: 0x003(3)
[1735179326.548363] info     | ProxyClient.cpp     | create_topic             | topic created          | client_key: 0x0E5C3397, topi
c_id: 0x004(2), participant_id: 0x000(1)
[1735179326.565153] info     | ProxyClient.cpp     | create_subscriber        | subscriber created     | client_key: 0x0E5C3397, subs
criber_id: 0x000(4), participant_id: 0x000(1)
[1735179326.574254] info     | ProxyClient.cpp     | create_datareader        | datareader created     | client_key: 0x0E5C3397, data
reader_id: 0x000(6), subscriber_id: 0x000(4)
```

Check the currently started node. In the two virtual machines, randomly select one and open the terminal to enter,

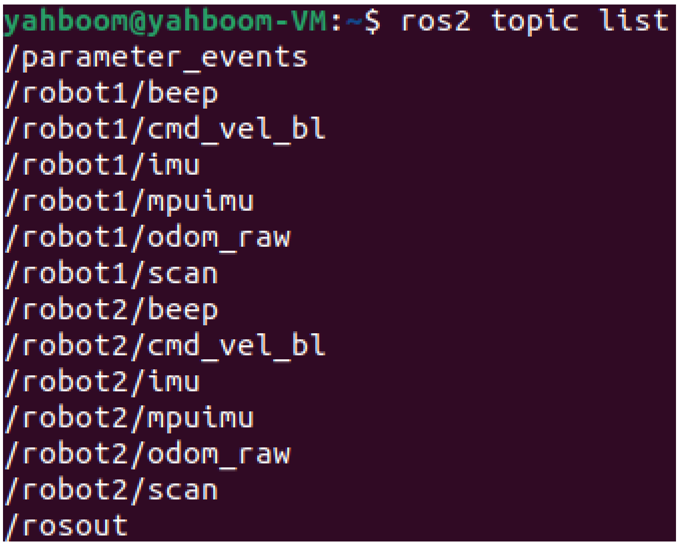
```
ros2 node list
```



```
yahboom@yahboom-VM:~$ ros2 node list
WARNING: Be aware that are nodes in the graph that share an exact name, this can
 have unintended side effects.
/robot2/YB_BalanceCar_Node
/robot2/YB_BalanceCar_Node
```

As shown in the figure above, the nodes of both cars have been started. Query the current topic information, input in the terminal,

```
ros2 topic list
```



```
yahboom@yahboom-VM:~$ ros2 topic list
/parameter_events
/robot1/beep
/robot1/cmd_vel_bl
/robot1/imu
/robot1/mpuimu
/robot1/odom_raw
/robot1/scan
/robot2/beep
/robot2/cmd_vel_bl
/robot2/imu
/robot2/mpuimu
/robot2/odom_raw
/robot2/scan
/rosout
```

4. Load the map program

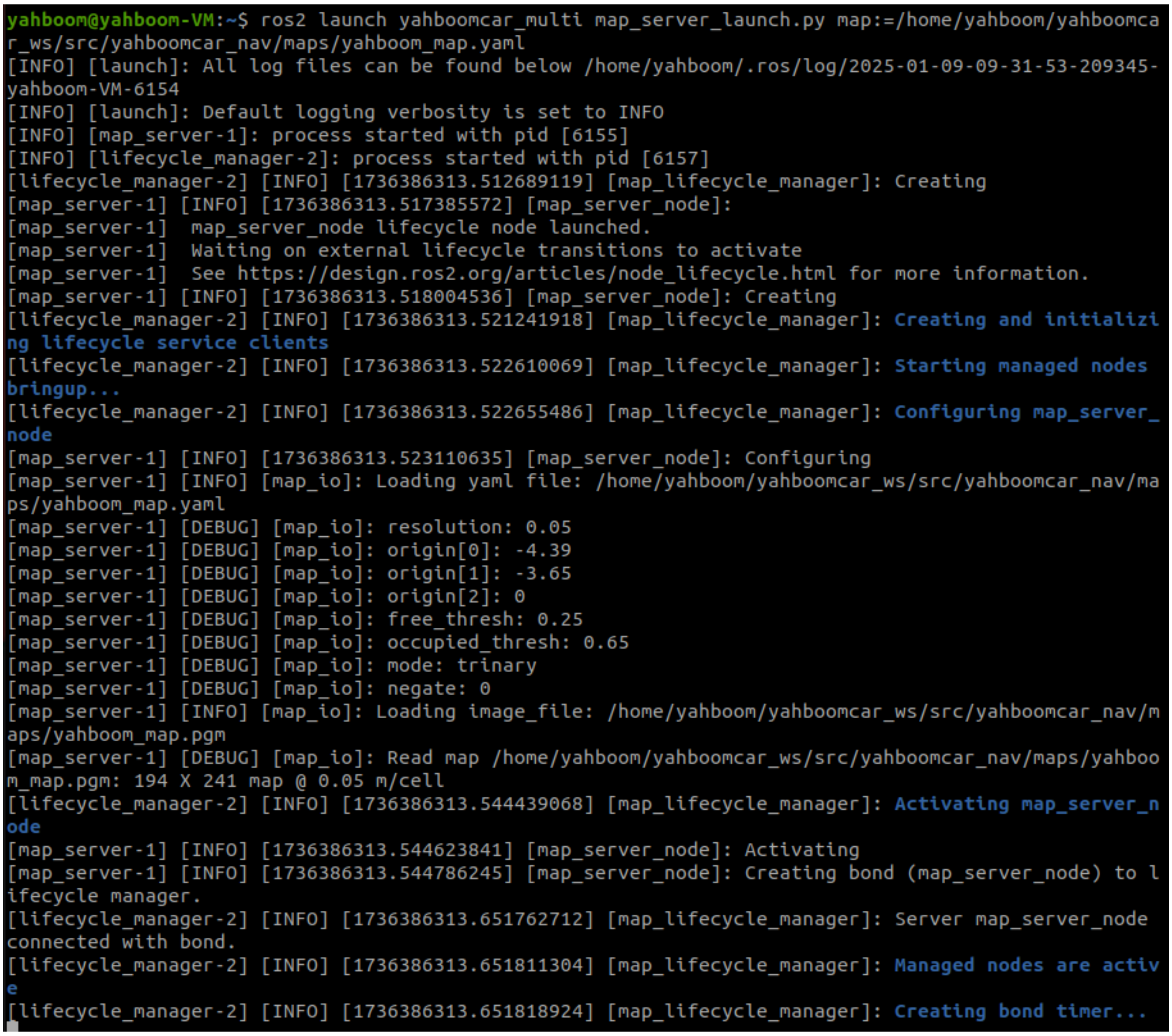
Select one of the two virtual machines at random, open the terminal and input,

```
ros2 launch yahboomcar_multi map_server_launch.py
••map:=/home/yahboom/yahboomcar_ws/src/yahboomcar_nav/maps/yahboom_map.yaml
```

> Parameter description

```
#Load map parameters: (target map can be replaced)
maps:=/home/yahboom/yahboomcar_ws/src/yahboomcar_nav/maps/yahboom_map.yaml
#.pgm file must also be in the same path as .yaml
```

Note: Here, yahboom_map.yaml and yahboom_map.pbstream must be built at the same time, that is, the same map, refer to cartograph mapping algorithm to save the map



```
yahboom@yahboom-VM:~$ ros2 launch yahboomcar_multi map_server_launch.py map:=/home/yahboom/yahboomca
r_ws/src/yahboomcar_nav/maps/yahboom_map.yaml
[INFO] [launch]: All log files can be found below /home/yahboom/.ros/log/2025-01-09-09-31-53-209345-
yahboom-VM-6154
[INFO] [launch]: Default logging verbosity is set to INFO
[INFO] [map_server-1]: process started with pid [6155]
[INFO] [lifecycle_manager-2]: process started with pid [6157]
[lifecycle_manager-2] [INFO] [1736386313.512689119] [map_lifecycle_manager]: Creating
[map_server-1] [INFO] [1736386313.517385572] [map_server_node]:
[map_server-1]  map_server_node lifecycle node launched.
[map_server-1]  Waiting on external lifecycle transitions to activate
[map_server-1]  See https://design.ros2.org/articles/node_lifecycle.html for more information.
[map_server-1] [INFO] [1736386313.518004536] [map_server_node]: Creating
[lifecycle_manager-2] [INFO] [1736386313.521241918] [map_lifecycle_manager]: Creating and initializi
ng lifecycle service clients
[lifecycle_manager-2] [INFO] [1736386313.522610069] [map_lifecycle_manager]: Starting managed nodes
bringup...
[lifecycle_manager-2] [INFO] [1736386313.522655486] [map_lifecycle_manager]: Configuring map_server_
node
[map_server-1] [INFO] [1736386313.523110635] [map_server_node]: Configuring
[map_server-1] [INFO] [map_io]: Loading yaml file: /home/yahboom/yahboomcar_ws/src/yahboomcar_nav/ma
ps/yahboom_map.yaml
[map_server-1] [DEBUG] [map_io]: resolution: 0.05
[map_server-1] [DEBUG] [map_io]: origin[0]: -4.39
[map_server-1] [DEBUG] [map_io]: origin[1]: -3.65
[map_server-1] [DEBUG] [map_io]: origin[2]: 0
[map_server-1] [DEBUG] [map_io]: free_thresh: 0.25
[map_server-1] [DEBUG] [map_io]: occupied_thresh: 0.65
[map_server-1] [DEBUG] [map_io]: mode: trinary
[map_server-1] [DEBUG] [map_io]: negate: 0
[map_server-1] [INFO] [map_io]: Loading image_file: /home/yahboom/yahboomcar_ws/src/yahboomcar_nav/m
aps/yahboom_map.pgm
[map_server-1] [DEBUG] [map_io]: Read map /home/yahboom/yahboomcar_ws/src/yahboomcar_nav/maps/yahboo
m_map.pgm: 194 X 241 map @ 0.05 m/cell
[lifecycle_manager-2] [INFO] [1736386313.544439068] [map_lifecycle_manager]: Activating map_server_n
ode
[map_server-1] [INFO] [1736386313.544623841] [map_server_node]: Activating
[map_server-1] [INFO] [1736386313.544786245] [map_server_node]: Creating bond (map_server_node) to l
ifecycle manager.
[lifecycle_manager-2] [INFO] [1736386313.651762712] [map_lifecycle_manager]: Server map_server_node
connected with bond.
[lifecycle_manager-2] [INFO] [1736386313.651811304] [map_lifecycle_manager]: Managed nodes are activ
e
[lifecycle_manager-2] [INFO] [1736386313.651818924] [map_lifecycle_manager]: Creating bond timer...
```

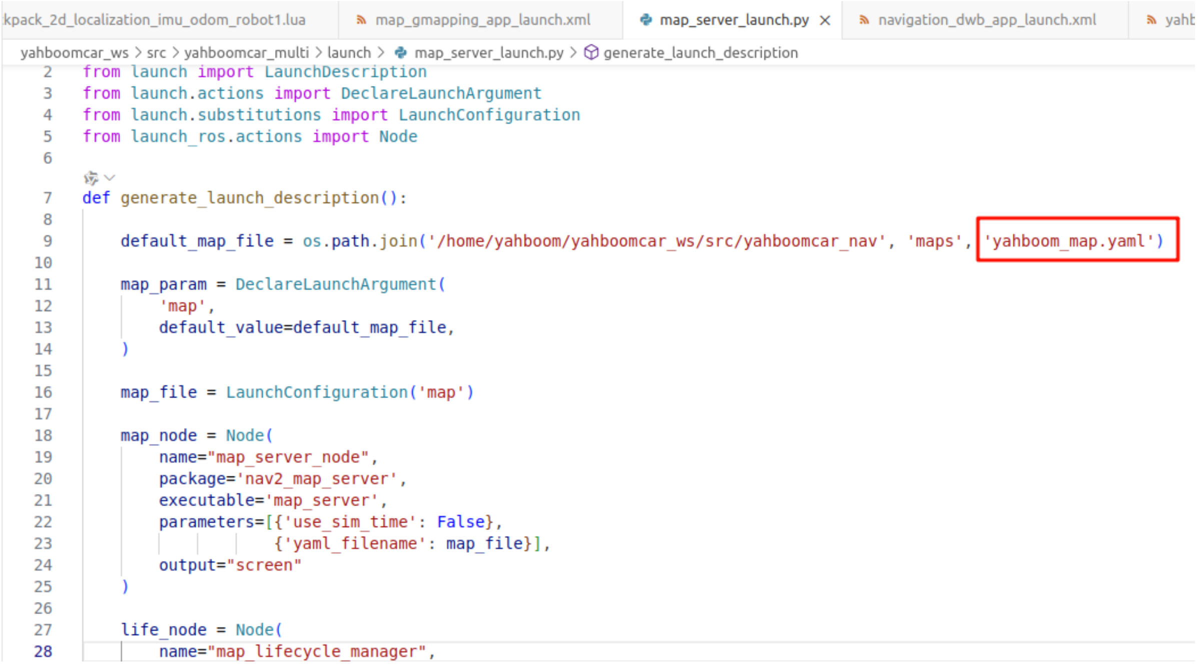
Note: Lifecycle error may occur here, which is caused by the map not being built.

Take the virtual machine as an example, the map loaded here is,

```
/home/yahboom/yahboomcar_ws/src/yahboomcar_nav/maps/yahboom_map.yaml
```

If you need to modify the default loading of other maps, copy the map's yaml file and pgm file to the

`/home/yahboom/yahboomcar_ws/src/yahboomcar_nav/maps/` directory, and then modify the map_server_launch.py program, which is located in `/home/yahboom/yahboomcar_ws/src/yahboomcar_multi/launch`, and modify the following places,


kpack_2d_localization_imu_odom_robot1.lua
map_gmapping_app_launch.xml
map_server_launch.py
navigation_dwb_app_launch.xml
yahb
yahboomcar_ws > src > yahboomcar_multi > launch > map_server_launch.py > generate_launch_description

```
from launch import LaunchDescription
from launch.actions import DeclareLaunchArgument
from launch.substitutions import LaunchConfiguration
from launch_ros.actions import Node

def generate_launch_description():

    default_map_file = os.path.join('/home/yahboom/yahboomcar_ws/src/yahboomcar_nav', 'maps', 'yahboom_map.yaml')

    map_param = DeclareLaunchArgument(
        'map',
        default_value=default_map_file,
    )

    map_file = LaunchConfiguration('map')

    map_node = Node(
        name="map_server_node",
        package='nav2_map_server',
        executable='map_server',
        parameters=[{'use_sim_time': False},
                    {'yaml_filename': map_file}],
        output="screen"
    )

    life_node = Node(
        name="map_lifecycle_manager",
```

Replace the red box with the name of your own map, save and exit, then enter the following command to compile,

```
cd ~/yahboomcar_ws
colcon build
```

Then enter the following command to re-source the environment variable,

```
source ~/.bashrc
```

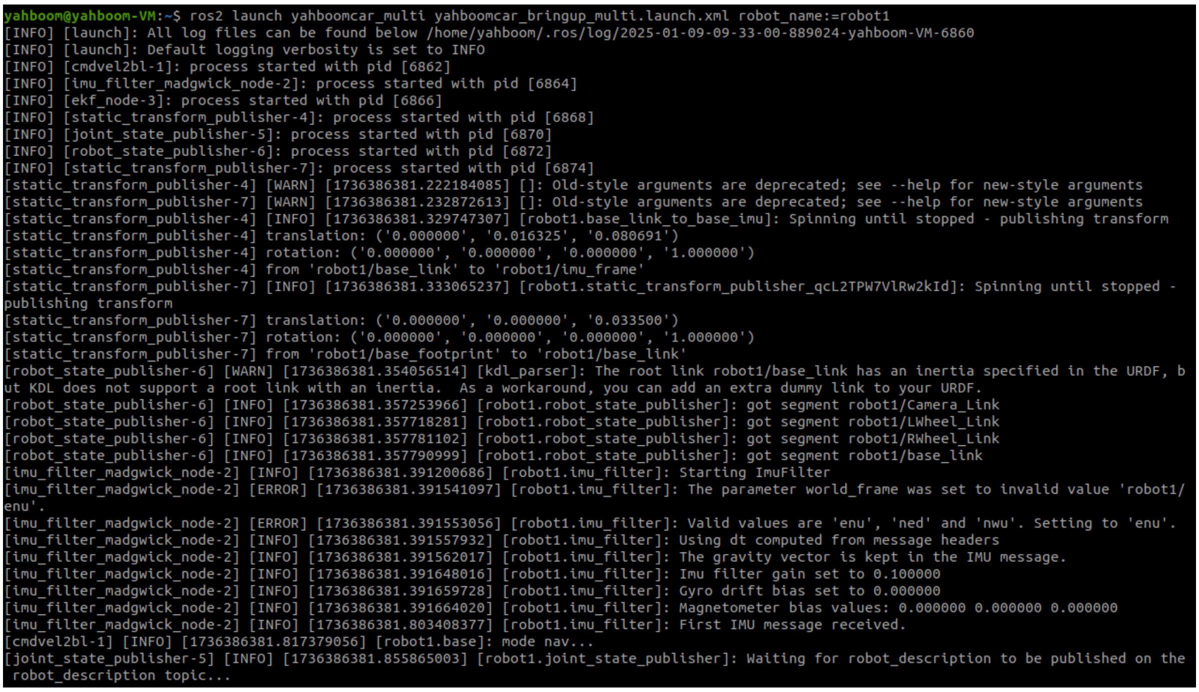
5. Start the car's underlying data processing program

In the virtual machine terminal that starts robot1,

```
ros2 launch yahboomcar_multi navigation_bringup_multi.launch.xml
robot_name:=robot1
```

In the virtual machine terminal that starts robot2,

```
ros2 launch yahboomcar_multi navigation_bringup_multi.launch.xml
robot_name:=robot2
```



```
yahboom@yahboom-VM:~$ ros2 launch yahboomcar_multi yahboomcar_bringup_multi.launch.xml robot_name:=robot1
[INFO] [launch]: All log files can be found below /home/yahboom/.ros/log/2025-01-09-09-33-00-889024-yahboom-VM-6860
[INFO] [launch]: Default logging verbosity is set to INFO
[INFO] [cmdvel2bl-1]: process started with pid [6862]
[INFO] [imu_filter_madgwick_node-2]: process started with pid [6864]
[INFO] [ekf_node-3]: process started with pid [6866]
[INFO] [static_transform_publisher-4]: process started with pid [6868]
[INFO] [joint_state_publisher-5]: process started with pid [6870]
[INFO] [robot_state_publisher-6]: process started with pid [6872]
[INFO] [static_transform_publisher-7]: process started with pid [6874]
[static_transform_publisher-4] [WARN] [1736386381.222184085] []: Old-style arguments are deprecated; see --help for new-style arguments
[static_transform_publisher-7] [WARN] [1736386381.232872613] []: Old-style arguments are deprecated; see --help for new-style arguments
[static_transform_publisher-4] [INFO] [1736386381.329747307] [robot1.base_link_to_base_imu]: Spinning until stopped - publishing transform
[static_transform_publisher-4] translation: ('0.000000', '0.016325', '0.080691')
[static_transform_publisher-4] rotation: ('0.000000', '0.000000', '0.000000', '1.000000')
[static_transform_publisher-4] from 'robot1/base_link' to 'robot1/imu_frame'
[static_transform_publisher-7] [INFO] [1736386381.333065237] [robot1.static_transform_publisher_qcL2TPW7VlRw2kId]: Spinning until stopped -
publishing transform
[static_transform_publisher-7] translation: ('0.000000', '0.000000', '0.033500')
[static_transform_publisher-7] rotation: ('0.000000', '0.000000', '0.000000', '1.000000')
[static_transform_publisher-7] from 'robot1/base_footprint' to 'robot1/base_link'
[robot_state_publisher-6] [WARN] [1736386381.354056514] [kdl_parser]: The root link robot1/base_link has an inertia specified in the URDF, b
ut KDL does not support a root link with an inertia.  As a workaround, you can add an extra dummy link to your URDF.
[robot_state_publisher-6] [INFO] [1736386381.357253966] [robot1.robot_state_publisher]: got segment robot1/Camera_Link
[robot_state_publisher-6] [INFO] [1736386381.357718281] [robot1.robot_state_publisher]: got segment robot1/LWheel_Link
[robot_state_publisher-6] [INFO] [1736386381.357781102] [robot1.robot_state_publisher]: got segment robot1/RWheel_Link
[robot_state_publisher-6] [INFO] [1736386381.357790999] [robot1.robot_state_publisher]: got segment robot1/base_link
[imu_filter_madgwick_node-2] [INFO] [1736386381.391200686] [robot1.imu_filter]: Starting ImuFilter
[imu_filter_madgwick_node-2] [ERROR] [1736386381.391541097] [robot1.imu_filter]: The parameter world_frame was set to invalid value 'robot1/
enu'.
[imu_filter_madgwick_node-2] [ERROR] [1736386381.391553056] [robot1.imu_filter]: Valid values are 'enu', 'ned' and 'nwu'. Setting to 'enu'.
[imu_filter_madgwick_node-2] [INFO] [1736386381.391557932] [robot1.imu_filter]: Using dt computed from message headers
[imu_filter_madgwick_node-2] [INFO] [1736386381.391562017] [robot1.imu_filter]: The gravity vector is kept in the IMU message.
[imu_filter_madgwick_node-2] [INFO] [1736386381.391648016] [robot1.imu_filter]: Imu filter gain set to 0.100000
[imu_filter_madgwick_node-2] [INFO] [1736386381.391659728] [robot1.imu_filter]: Gyro drift bias set to 0.000000
[imu_filter_madgwick_node-2] [INFO] [1736386381.391664020] [robot1.imu_filter]: Magnetometer bias values: 0.000000 0.000000 0.000000
[imu_filter_madgwick_node-2] [INFO] [1736386381.803408377] [robot1.imu_filter]: First IMU message received.
[cmdvel2bl-1] [INFO] [1736386381.817379056] [robot1.base]: mode nav...
[joint_state_publisher-5] [INFO] [1736386381.855865003] [robot1.joint_state_publisher]: Waiting for robot_description to be published on the
 robot_description topic...
```

yahboomcar_bringup_multi.launch.xml source code path (taking the matching virtual machine as an example),

```
/home/yahboom/yahboomcar_ws/src/yahboomcar_multi/launch/yahboomcar_bringup_multi.launch.xml
```

```
<launch>
    <arg name="robot_name" default="robot1"/>
    <group>
        <push-ros-namespace namespace="$(var robot_name)"/>
        <!--base_node-->
        <node name="base" pkg="yahboomcar_bringup" exec="cmdvel2bl" output="screen">
            <param name="mode" value="nav"/>
            <remap from="/cmd_vel" to="cmd_vel"/>
            <remap from="/cmd_vel_bl" to="cmd_vel_bl"/>
        </node>
        <!--imu_filter_node-->
        <node name="imu_filter" pkg="imu_filter_madgwick" exec="imu_filter_madgwick_node" output="screen">
            <param name="fixed_frame" value="$(var robot_name)/base_link"/>
            <param name="use_mag" value="false"/>
            <param name="publish_tf" value="false"/>
            <param name="world_frame" value="$(var robot_name)/enu"/>
            <param name="orientation_stddev" value="0.00"/>
            <remap from="imu/data_raw" to="imu"/>
        </node>
        <!--ekf_node-->
        <node name="ekf_filter_node" pkg="robot_localization" exec="ekf_node">
            <param from="$(find-pkg-share yahboomcar_multi)/param/ekf_$(var robot_name).yaml"/>
            <remap from="odometry/filtered" to="odom"/>
            <remap from="/odom_raw" to="odom_raw"/>
        </node>
        <node pkg="tf2_ros" exec="static_transform_publisher" name="base_link_to_base_imu"
```

```
          args="0 0.016325 0.080691 0 0 0  $(var robot_name)/base_link $(var robot_name)/imu_frame " />
    </group>
    <include file="$(find-pkg-share yahboomcar_description)/launch/description_multi_$(var robot_name).launch.py"/>

</launch>
```

Here, a pair of tags are used. The command space of all programs in this tag will be robot_name, which is the robot1 or robot2 we defined. Among them, there are some parameter files or topic names, which are also automatically selected and loaded through this robot_name. You can view the content in the code for details.

6. Start the fast relocation program and rviz of the car

Input in the virtual machine terminal that starts robot1, and you can choose any rviz to start.

```
ros2 launch yahboomcar_multi robot1_localization_imu_odom.launch.py load_state_filename:=/home/yahboom/yahboomcar_ws/src/yahboomcar_nav/maps/yahboom_map.pbstream use_rviz:=true
```

> Parameter Description

```
#Select whether to open rviz, true is open, false is not open
use_rviz:=true
#Replaceable target map file
load_state_filename:=/home/yahboom/yahboomcar_ws/src/yahboomcar_nav/maps/testaa.pbstream
```

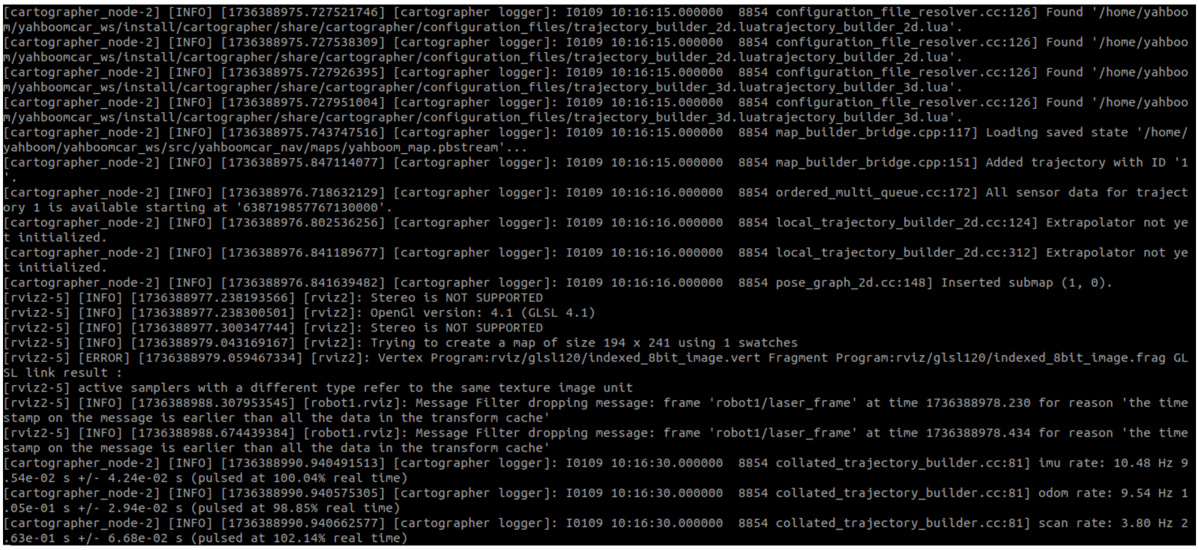
In the terminal of the virtual machine that starts robot2, enter,

```
ros2 launch yahboomcar_multi robot2_localization_imu_odom.launch.py load_state_filename:=/home/yahboom/yahboomcar_ws/src/yahboomcar_nav/maps/yahboom_map.pbstream use_rviz:=false
```

> Parameter description

```
#Select whether to open rviz, true is open, false is not open
use_rviz:=false
#Replaceable target map file
load_state_filename:=/home/yahboom/yahboomcar_ws/src/yahboomcar_nav/maps/testaa.pbstream
```

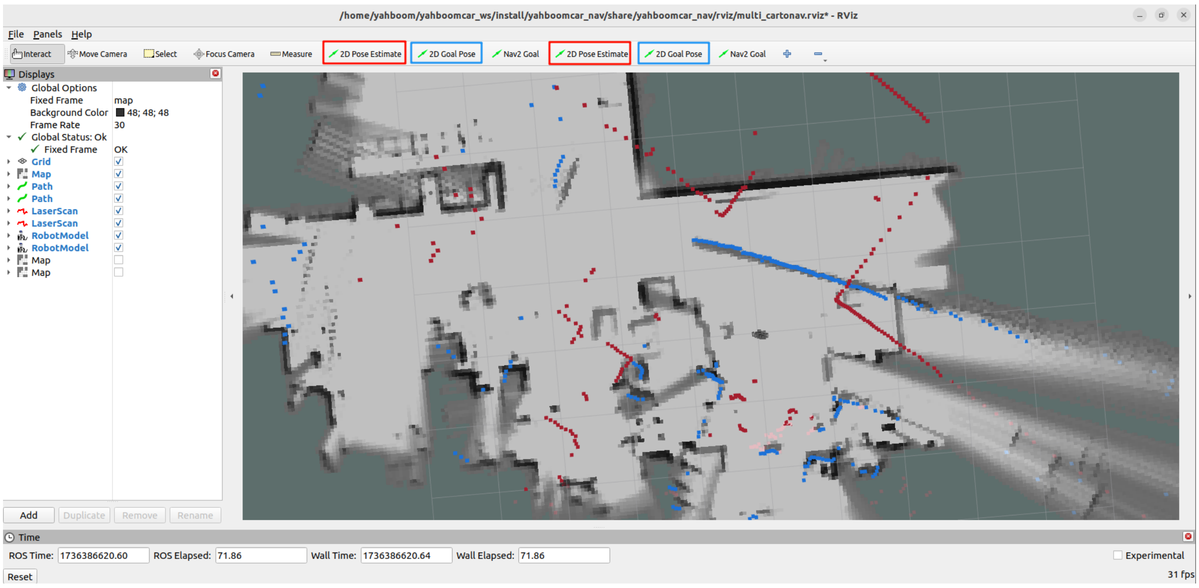
Note: The testaa.pbstream map of robot1 and robot2 must be the same map



```
[cartographer_node-2] [INFO] [1736388975.727521746] [cartographer logger]: I0109 10:16:15.000000  8854 configuration_file_resolver.cc:126] Found '/home/yahboo
m/yahboomcar_ws/install/cartographer/share/cartographer/configuration_files/trajectory_builder_2d.luatrajectory_builder_2d.lua'.
[cartographer_node-2] [INFO] [1736388975.727538309] [cartographer logger]: I0109 10:16:15.000000  8854 configuration_file_resolver.cc:126] Found '/home/yahboo
m/yahboomcar_ws/install/cartographer/share/cartographer/configuration_files/trajectory_builder_2d.luatrajectory_builder_2d.lua'.
[cartographer_node-2] [INFO] [1736388975.727926395] [cartographer logger]: I0109 10:16:15.000000  8854 configuration_file_resolver.cc:126] Found '/home/yahboo
m/yahboomcar_ws/install/cartographer/share/cartographer/configuration_files/trajectory_builder_3d.luatrajectory_builder_3d.lua'.
[cartographer_node-2] [INFO] [1736388975.727951004] [cartographer logger]: I0109 10:16:15.000000  8854 configuration_file_resolver.cc:126] Found '/home/yahboo
m/yahboomcar_ws/install/cartographer/share/cartographer/configuration_files/trajectory_builder_3d.luatrajectory_builder_3d.lua'.
[cartographer_node-2] [INFO] [1736388975.743747516] [cartographer logger]: I0109 10:16:15.000000  8854 map_builder_bridge.cpp:117] Loading saved state '/home/
yahboom/yahboomcar_ws/src/yahboomcar_nav/maps/yahboom_map.pbstream'...
[cartographer_node-2] [INFO] [1736388975.847114077] [cartographer logger]: I0109 10:16:15.000000  8854 map_builder_bridge.cpp:151] Added trajectory with ID '1
'.
[cartographer_node-2] [INFO] [1736388976.718632129] [cartographer logger]: I0109 10:16:16.000000  8854 ordered_multi_queue.cc:172] All sensor data for traject
ory 1 is available starting at '638719857767130000'.
[cartographer_node-2] [INFO] [1736388976.802536256] [cartographer logger]: I0109 10:16:16.000000  8854 local_trajectory_builder_2d.cc:124] Extrapolator not ye
t initialized.
[cartographer_node-2] [INFO] [1736388976.841189677] [cartographer logger]: I0109 10:16:16.000000  8854 local_trajectory_builder_2d.cc:312] Extrapolator not ye
t initialized.
[cartographer_node-2] [INFO] [1736388976.841639482] [cartographer logger]: I0109 10:16:16.000000  8854 pose_graph_2d.cc:148] Inserted submap (1, 0).
[rviz2-5] [INFO] [1736388977.238193566] [rviz2]: Stereo is NOT SUPPORTED
[rviz2-5] [INFO] [1736388977.238300501] [rviz2]: OpenGl version: 4.1 (GLSL 4.1)
[rviz2-5] [INFO] [1736388977.300347744] [rviz2]: Stereo is NOT SUPPORTED
[rviz2-5] [INFO] [1736388979.043169167] [rviz2]: Trying to create a map of size 194 x 241 using 1 swatches
[rviz2-5] [ERROR] [1736388979.059467334] [rviz2]: Vertex Program:rviz/glsl120/indexed_8bit_image.vert Fragment Program:rviz/glsl120/indexed_8bit_image.frag GL
SL link result :
[rviz2-5] active samplers with a different type refer to the same texture image unit
[rviz2-5] [INFO] [1736388988.307953545] [robot1.rviz]: Message Filter dropping message: frame 'robot1/laser_frame' at time 1736388978.230 for reason 'the time
stamp on the message is earlier than all the data in the transform cache'
[rviz2-5] [INFO] [1736388988.674439384] [robot1.rviz]: Message Filter dropping message: frame 'robot1/laser_frame' at time 1736388978.434 for reason 'the time
stamp on the message is earlier than all the data in the transform cache'
[cartographer_node-2] [INFO] [1736388990.940491513] [cartographer logger]: I0109 10:16:30.000000  8854 collated_trajectory_builder.cc:81] imu rate: 10.48 Hz 9
.54e-02 s +/- 4.24e-02 s (pulsed at 100.04% real time)
[cartographer_node-2] [INFO] [1736388990.940575305] [cartographer logger]: I0109 10:16:30.000000  8854 collated_trajectory_builder.cc:81] odom rate: 9.54 Hz 1
.05e-01 s +/- 2.94e-02 s (pulsed at 98.85% real time)
[cartographer_node-2] [INFO] [1736388990.940662577] [cartographer logger]: I0109 10:16:30.000000  8854 collated_trajectory_builder.cc:81] scan rate: 3.80 Hz 2
.63e-01 s +/- 6.68e-02 s (pulsed at 102.14% real time)
```

When it prints (pulsed at 101.80% real time)...etc., it proves that the startup is normal

You can see that the map is loaded, but the positioning of the car is not accurate, then we click [2D Pose Estimate】, give the two cars an initial pose respectively. According to the approximate position of the car in the actual environment, click and drag with the mouse in rviz, and the car can be quickly positioned on the map.



Global Options
Fixed Frame: map
Background Color: ■ 48; 48; 48
Frame Rate: 30
✓ Global Status: Ok
✓ Fixed Frame: OK
Grid ☑
Map ☑
Path ☑
Path ☑
LaserScan ☑
LaserScan ☑
RobotModel ☑
RobotModel ☑
Map □
Map □

ROS Time: 1736386620.60 ROS Elapsed: 71.86 Wall Time: 1736386620.64 Wall Elapsed: 71.86 □ Experimental

Red represents robot1 and blue represents robot2.

The functions of the symbols in Figures 1-4 above are as follows:

[1]: robot1 calibrates the initial pose

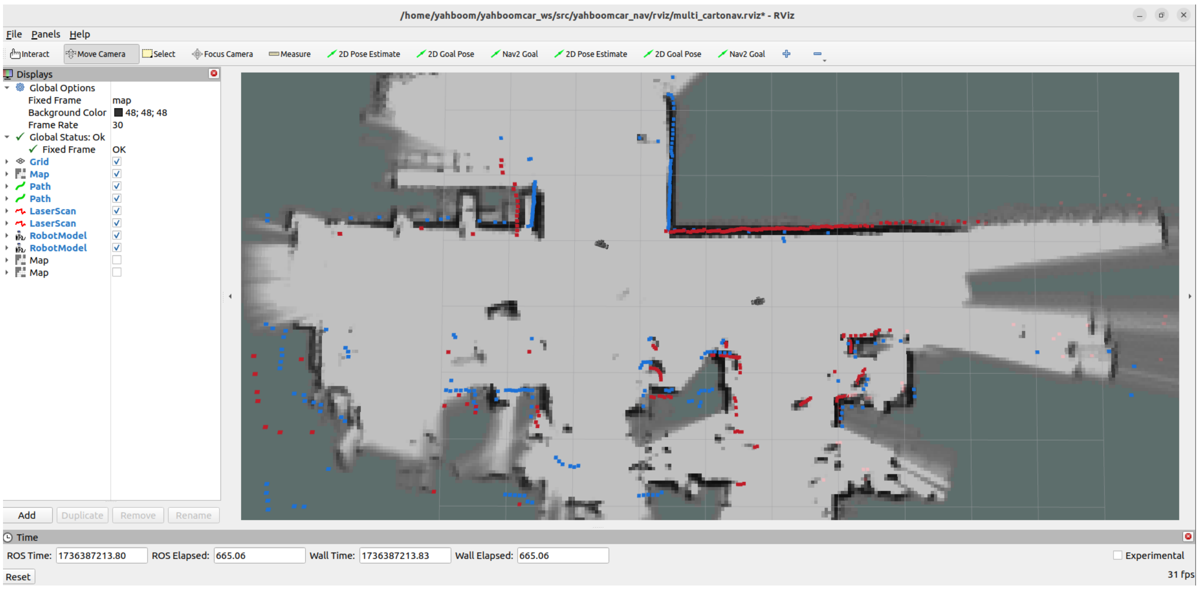
[2]: robot1 gives the target point

[3]: robot2 calibrates the initial pose

[4]: robot2 gives the target point

Note: If positioning is still not possible, it may be that the car is in an obstacle area. Please place it in a place with no obstacles around and start again. You can also keep the terminal open, restart the car directly, and wait for the automatic data recovery to re-position it. It may also be that there are too few map feature points when building the map, and the map construction must be more detailed.

As shown in the figure below, if the area scanned by the radar roughly coincides with the actual obstacle, it means that the posture is accurate.



Global Options
Fixed Frame: map
Background Color: 48; 48; 48
Frame Rate: 30
Global Status: Ok
Fixed Frame: OK

ROS Time: 1736387213.80 ROS Elapsed: 665.06 Wall Time: 1736387213.83 Wall Elapsed: 665.06 □ Experimental

Take the virtual machine as an example, the source code location: /home/yahboom/yahboomcar_ws/src/yahboomcar_multi/launch

robot1_localization_imu_odom.launch.py

```
from launch import LaunchDescription
from launch.actions import DeclareLaunchArgument, GroupAction, LogInfo,Shutdown
from launch.conditions import IfCondition
from launch.substitutions import LaunchConfiguration, TextSubstitution
from launch_ros.actions import Node, PushRosNamespace
from launch_ros.substitutions import FindPackageShare
def generate_launch_description():
    robot_name = 'robot1'

    load_state_filename_arg = DeclareLaunchArgument(
        'load_state_filename',

 default_value='/home/yahboom/yahboomcar_ws/src/yahboomcar_nav/maps/yahboom_map.p
bstream'
    )

    use_rviz_arg = DeclareLaunchArgument(
        'use_rviz',
        default_value='true',
    )

    rviz_node = Node(
        package='rviz2',
        executable='rviz2',
        on_exit=Shutdown(),
        arguments=['-d',
FindPackageShare('yahboomcar_nav').find('yahboomcar_nav') +
'/rviz/multi_cartonav.rviz'],
        parameters=[{'use_sim_time': False}],
        condition=IfCondition(LaunchConfiguration('use_rviz'))
    )
    base_link_to_laser_tf_node = Node(
        package='tf2_ros',
        executable='static_transform_publisher',
        name='base_link_to_base_laser',
        namespace = 'robot1',
        arguments=['0', '0' ,
'0.138','0','0','0','robot1/base_link','robot1/laser_frame']
    )
    group = GroupAction([
        PushRosNamespace(namespace=robot_name),

        Node(
            package='cartographer_ros',
            executable='cartographer_node',
            parameters=[{'use_sim_time': False}],
            arguments=[
                '-configuration_directory',
FindPackageShare('cartographer_ros').find('cartographer_ros') +
'/configuration_files',
                '-configuration_basename',
f'ros1_backpack_2d_localization_imu_odom_{robot_name}.lua',
                '-load_state_filename',
LaunchConfiguration('load_state_filename')],
            remappings=[
                ('imu', 'imu/data'),
                ('scan', 'scan')],
            output='screen'
        ),

        Node(
            package='cartographer_ros',
            executable='cartographer_occupancy_grid_node',
            parameters=[
                {'use_sim_time': False},
                {'resolution': 0.05}],
            remappings=[
                ('map', 'map')]
        ),

        Node(
            package='tf2_ros',
            executable='static_transform_publisher',
            name='base_link_to_base_laser',
            arguments=['0', '0' ,
'0.138','0','0','0','robot1/base_link','robot1/laser_frame']
        ),

        rviz_node,

    ])

    return LaunchDescription([
        # Launch arguments
        load_state_filename_arg,
        use_rviz_arg,
        # Group for robot1
        base_link_to_laser_tf_node,
        group
    ])
```

The following nodes are started here:

- **rviz_node**: Start `rviz2` to visualize the robot status.
- **cartographer_node**: Start `cartographer_node` for SLAM positioning and mapping.
- **cartographer_occupancy_grid_node**: Start `cartographer_occupancy_grid_node` to generate occupancy grid map.
- **base_link_to_laser_tf_node**: Publish the static transformation from chassis to radar.

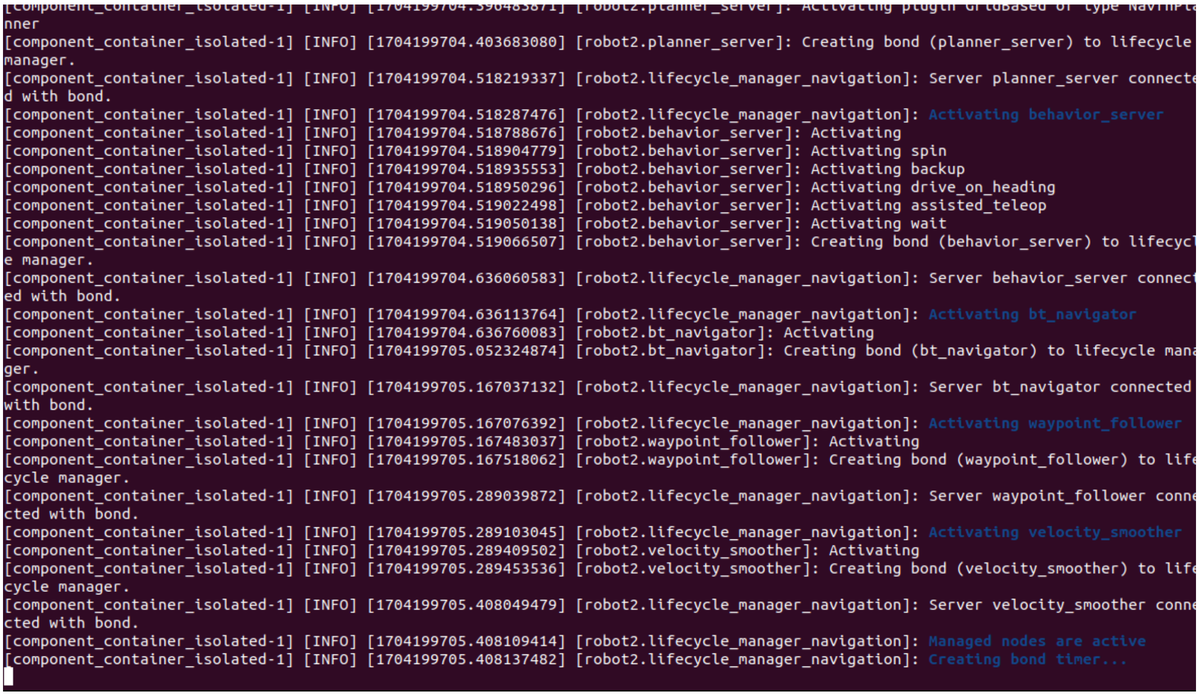
7. Start the car navigation program

In the virtual machine terminal that starts robot1, enter,

```
ros2 launch yahboomcar_multi robot1_navigation_dwb_launch.py
```

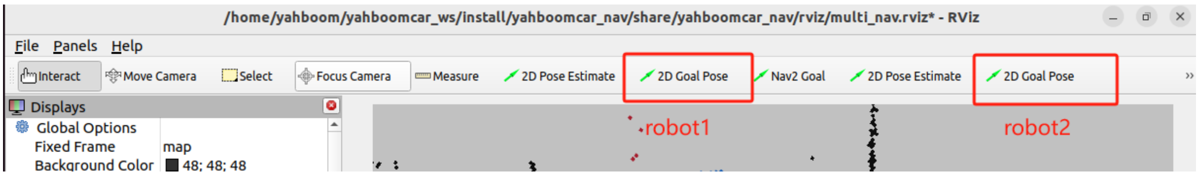
In the virtual machine terminal that starts robot2, enter,

```
ros2 launch yahboomcar_multi robot2_navigation_dwb_launch.py
```



```
[component_container_isolated-1] [INFO] [1704199704.396483871] [robot2.planner_server]: Activating plugin GridBased of type NavfnPla
nner
[component_container_isolated-1] [INFO] [1704199704.403683080] [robot2.planner_server]: Creating bond (planner_server) to lifecycle
manager.
[component_container_isolated-1] [INFO] [1704199704.518219337] [robot2.lifecycle_manager_navigation]: Server planner_server connecte
d with bond.
[component_container_isolated-1] [INFO] [1704199704.518287476] [robot2.lifecycle_manager_navigation]: Activating behavior_server
[component_container_isolated-1] [INFO] [1704199704.518788676] [robot2.behavior_server]: Activating
[component_container_isolated-1] [INFO] [1704199704.518904779] [robot2.behavior_server]: Activating spin
[component_container_isolated-1] [INFO] [1704199704.518935553] [robot2.behavior_server]: Activating backup
[component_container_isolated-1] [INFO] [1704199704.518950296] [robot2.behavior_server]: Activating drive_on_heading
[component_container_isolated-1] [INFO] [1704199704.519022498] [robot2.behavior_server]: Activating assisted_teleop
[component_container_isolated-1] [INFO] [1704199704.519050138] [robot2.behavior_server]: Activating wait
[component_container_isolated-1] [INFO] [1704199704.519066507] [robot2.behavior_server]: Creating bond (behavior_server) to lifecycl
e manager.
[component_container_isolated-1] [INFO] [1704199704.636060583] [robot2.lifecycle_manager_navigation]: Server behavior_server connect
ed with bond.
[component_container_isolated-1] [INFO] [1704199704.636113764] [robot2.lifecycle_manager_navigation]: Activating bt_navigator
[component_container_isolated-1] [INFO] [1704199704.636760083] [robot2.bt_navigator]: Activating
[component_container_isolated-1] [INFO] [1704199705.052324874] [robot2.bt_navigator]: Creating bond (bt_navigator) to lifecycle mana
ger.
[component_container_isolated-1] [INFO] [1704199705.167037132] [robot2.lifecycle_manager_navigation]: Server bt_navigator connected
with bond.
[component_container_isolated-1] [INFO] [1704199705.167076392] [robot2.lifecycle_manager_navigation]: Activating waypoint_follower
[component_container_isolated-1] [INFO] [1704199705.167483037] [robot2.waypoint_follower]: Activating
[component_container_isolated-1] [INFO] [1704199705.167518062] [robot2.waypoint_follower]: Creating bond (waypoint_follower) to life
cycle manager.
[component_container_isolated-1] [INFO] [1704199705.289039872] [robot2.lifecycle_manager_navigation]: Server waypoint_follower conne
cted with bond.
[component_container_isolated-1] [INFO] [1704199705.289103045] [robot2.lifecycle_manager_navigation]: Activating velocity_smoother
[component_container_isolated-1] [INFO] [1704199705.289409502] [robot2.velocity_smoother]: Activating
[component_container_isolated-1] [INFO] [1704199705.289453536] [robot2.velocity_smoother]: Creating bond (velocity_smoother) to life
cycle manager.
[component_container_isolated-1] [INFO] [1704199705.408049479] [robot2.lifecycle_manager_navigation]: Server velocity_smoother conne
cted with bond.
[component_container_isolated-1] [INFO] [1704199705.408109414] [robot2.lifecycle_manager_navigation]: Managed nodes are active
[component_container_isolated-1] [INFO] [1704199705.408137482] [robot2.lifecycle_manager_navigation]: Creating bond timer...
```

As shown in the figure above, the "Creating bond timer..." appears, indicating that the program is loaded. Then, the corresponding [2D Goal Pose] on riviz can be used to give the target points of the two cars. The cars will generate a path based on their respective postures and surrounding obstacles, and autonomously navigate to their respective destinations.

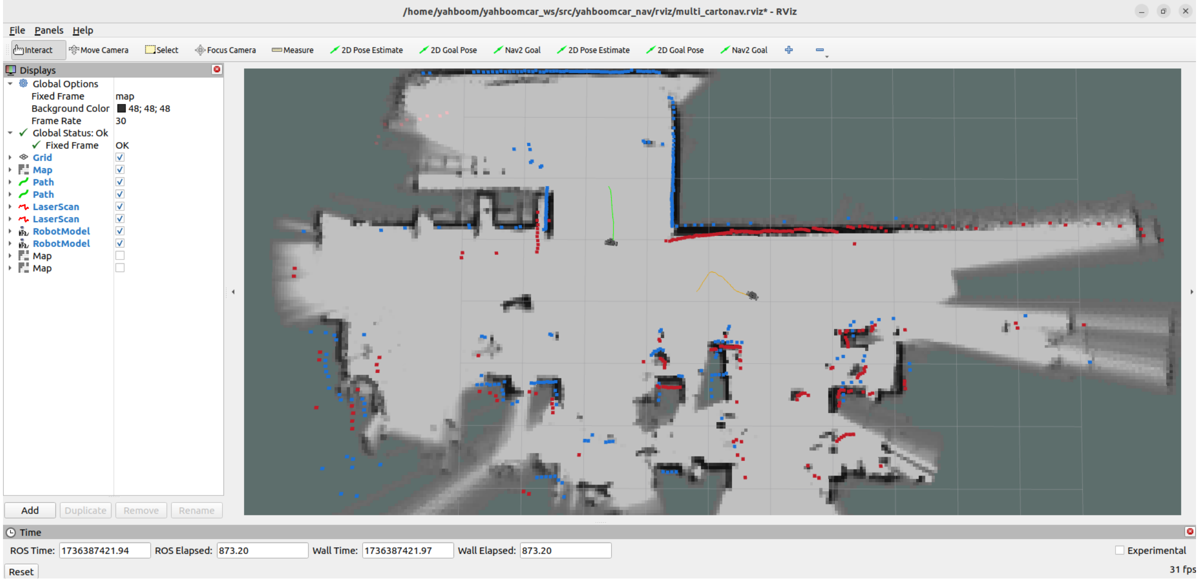


/home/yahboom/yahboomcar_ws/install/yahboomcar_nav/share/yahboomcar_nav/rviz/multi_nav.rviz* - RViz

Fixed Frame map

Background Color 48; 48; 48

The yellow route is the route planned by robot1, and the green line is the route planned by robot2.



Global Options
Fixed Frame map
Background Color 48; 48; 48
Frame Rate 30
✓ Global Status: Ok
✓ Fixed Frame OK
Grid ☑
Map ☑
Path ☑
Path ☑
LaserScan ☑
LaserScan ☑
RobotModel ☑
RobotModel ☑
Map □
Map □

ROS Time: 1736387421.94 ROS Elapsed: 873.20 Wall Time: 1736387421.97 Wall Elapsed: 873.20 □ Experimental

31 fps

8. Multi-car navigation expansion

The tutorial takes two cars as an example. If you want to add other cars, you need to make the following modifications.

8.1. Add a URDF model of the car and add a urdf model loader

Note: The path is,

```
/home/yahboom/yahboomcar_ws/src/yahboomcar_description/launch/description_multi_robot1.launch.py
/home/yahboom/yahboomcar_ws/src/yahboomcar_description/urdf/STM32-V2-V1_robot1.urdf
```

- Add a car model

For reference, `/home/yahboom/yahboomcar_ws/src/yahboomcar_description/urdf/STM32-V2-V1_robot1.urdf`

Change the name and robot1 in the urdf file to other car names, such as robot3.

- Added urdf model loader

For reference, `/home/yahboom/yahboomcar_ws/src/yahboomcar_description/launch/description_multi_robot1.launch.py` Change the name and robot1 in the launch file to other car names. The name needs to be consistent with the newly added urdf.

8.2, Added car ekf parameter table

Note: The path is,

```
/home/yahboom/yahboomcar_ws/src/yahboomcar_multi/param/ekf_robot1.yaml
```

You can refer to `/home/yahboom/yahboomcar_ws/src/yahboomcar_multi/param/ekf_robot1.yaml`, change the name and robot1 in the file to other car names, and the name needs to be consistent with the newly added urdf.

8.3, Added launch file for starting positioning

Note: The path is:

```
/home/yahboom/yahboomcar_ws/src/yahboomcar_multi/launch/robot1_localization_imu_odom.launch.py
```

- Added launch file for positioning

Refer to `/home/yahboom/yahboomcar_ws/src/yahboomcar_multi/launch/robot1_localization_imu_odom.launch.py`, change the name and robot1 in the file to the name of other cars, and the name needs to be consistent with the newly added urdf.

8.4, Added car nav2 parameter table and launch file for starting nav2

Note: The path is:

```
/home/yahboom/yahboomcar_ws/src/yahboomcar_multi/param/robot1_nav_params.yaml
/home/yahboom/yahboomcar_ws/src/yahboomcar_multi/launch/robot1_navigation_dwb_launch.py
```

- Added car nav2 parameter table

Refer to `/home/yahboom/yahboomcar_ws/src/yahboomcar_multi/param/robot1_nav_params.yaml`, change the name and robot1 in the file to the name of other cars, and the name needs to be consistent with the newly added urdf.

- Added launch file for nav2

Refer to `/home/yahboom/yahboomcar_ws/src/yahboomcar_multi/launch/robot1_navigation_dwb_launch.py`, change the name and robot1 in the file to the name of other cars, and the name needs to be consistent with the newly added urdf.

8.5, Added [2D Pose Estimate] and [2D Goal Pose] in the rviz toolbar

Note: The path is:

```
/home/yahboom/yahboomcar_ws/src/yahboomcar_nav/rviz
```

Modify the multi_nav.rviz file, the directory of the file is `/home/yahboom/yahboomcar_ws/src/yahboomcar_nav/rviz/multi_nav.rviz`, find the following part,

```
- Class: rviz_default_plugins/SetInitialPose
  Covariance x: 0.25
  Covariance y: 0.25
  Covariance yaw: 0.06853891909122467
  Topic:
    Depth: 5
    Durability Policy: Volatile
    History Policy: Keep Last
    Reliability Policy: Reliable
    Value: /robot1/initialpose
- Class: rviz_default_plugins/SetGoal
  Topic:
    Depth: 5
    Durability Policy: Volatile
    History Policy: Keep Last
    Reliability Policy: Reliable
    Value: /robot1/goal_pose
...
```

The above are two tools for robot1. You can copy one and put it behind. Change the robot1 that appears to the name of other cars. The name needs to be consistent with the newly added urdf.

After completing the above 5 steps, return to the yahboomcar_ws workspace, compile it using colcon build, and then run the test according to the tutorial. After successful operation, you can add the car model and radar data to display in rviz.