Gmapping mapping

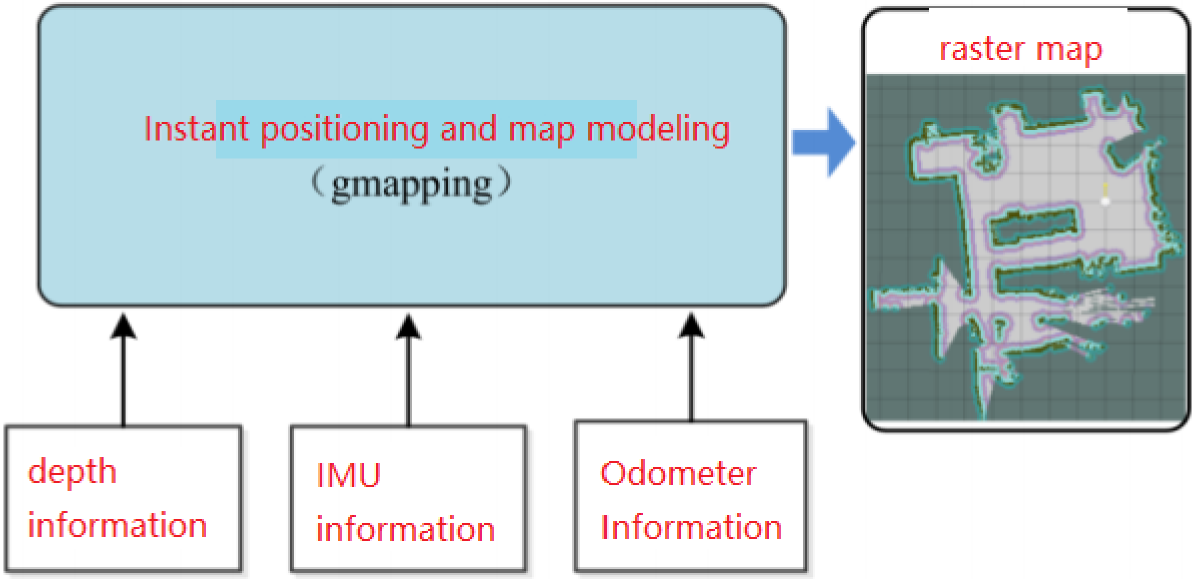
Note: The virtual machine needs to be in the same LAN as the car, and the ROS_DOMAIN_ID needs to be consistent. You can check [Must read before use] to set the IP and ROS_DOMAIN_ID on the board.

1. Introduction to Gmapping

- gmapping is only applicable to points where the number of two-dimensional laser points in a single frame is less than 1440. If the number of laser points in a single frame is greater than 1440, then [[mapping-4] process has died] will occur.
- Gmapping is a commonly used open source SLAM algorithm based on the filtered SLAM framework.
- Gmapping is based on the RBpf particle filter algorithm, which separates the real-time positioning and mapping processes. Positioning is performed first and then mapping is performed.
- Gmapping has made two major improvements on the RBpf algorithm: improved proposal distribution and selective resampling.

Advantages: Gmapping can construct indoor maps in real time. The amount of calculation required to construct small scene maps is small and the accuracy is high.

Disadvantages: As the scene grows, the number of particles required increases because each particle carries a map, so the amount of memory and computation required to build a large map increases. Therefore it is not suitable for building large scene maps. And there is no loop detection, so the map may be misaligned when the loop is closed. Although increasing the number of particles can close the map, it comes at the expense of increased calculations and memory.



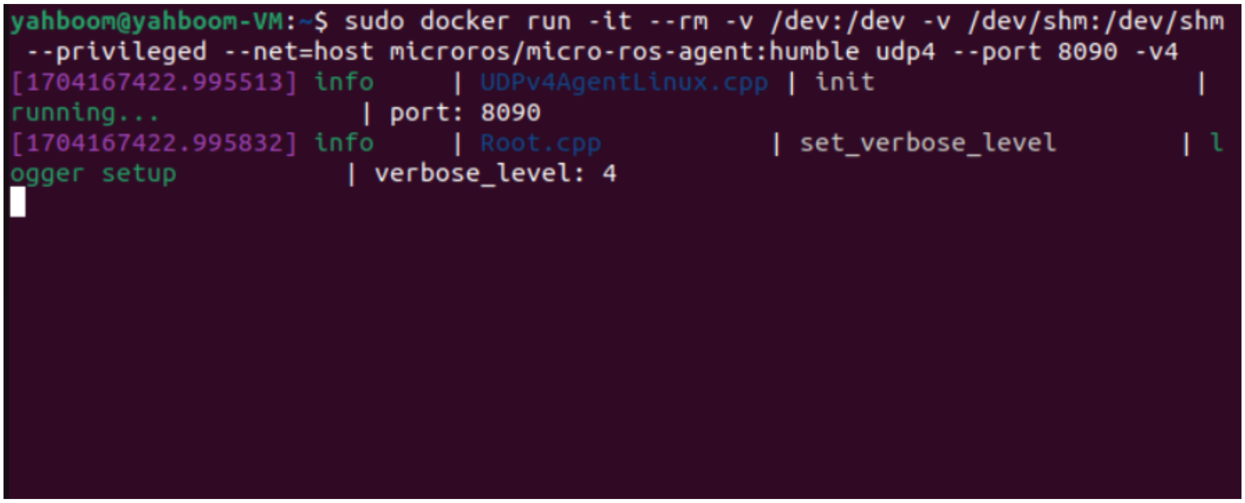
2. Program function description

Connect the car to the agent and run the program. The mapping interface will be displayed in rviz. Use the keyboard or handle to control the movement of the car until the map is completed. Then run the save map command to save the map.

3. Start and connect to the agent

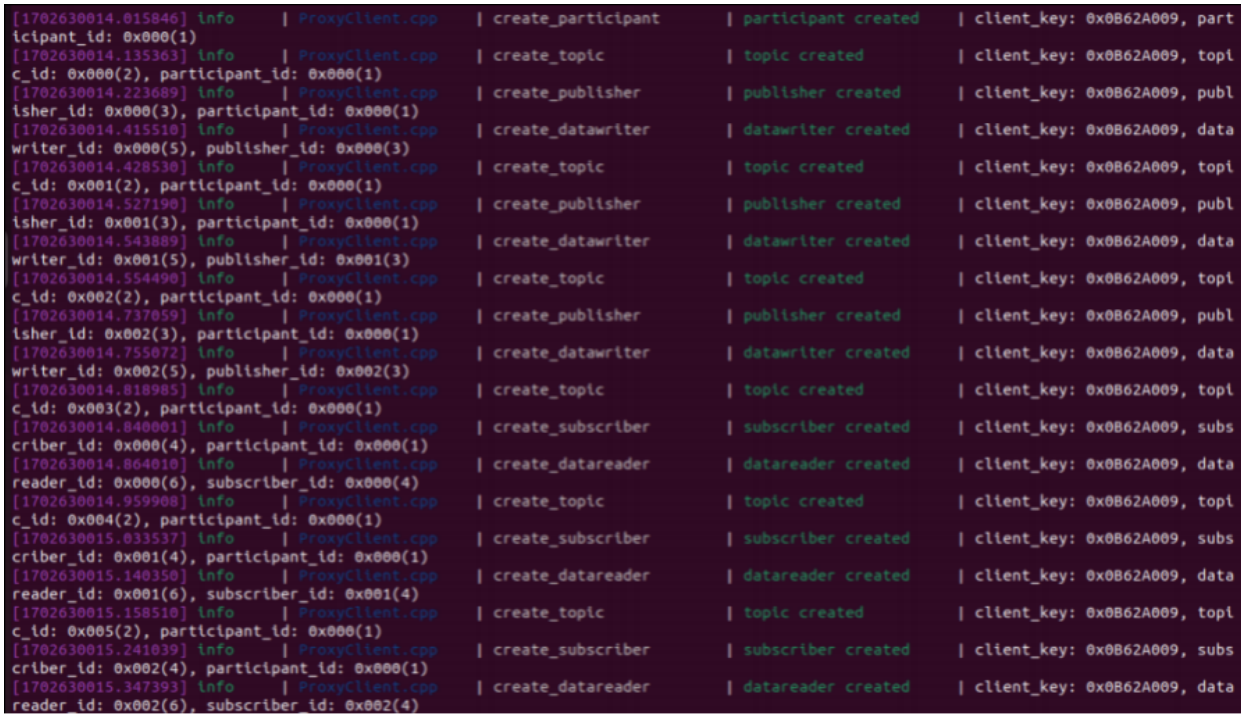
Taking the supporting virtual machine as an example, enter the following command to start the agent:

```
sudo docker run -it --rm -v /dev:/dev -v /dev/shm:/dev/shm --privileged --net=host
microros/micro-ros-agent:humble udp4 --port 8090 -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 8090 -v4
[1704167422.995513] info     | UDPv4AgentLinux.cpp | init                     |
running...              | port: 8090
[1704167422.995832] info     | Root.cpp             | set_verbose_level       | l
ogger setup             | verbose_level: 4
```

Then, turn on the car switch and wait for the car to connect to the agent. The connection is successful, as shown in the figure below.

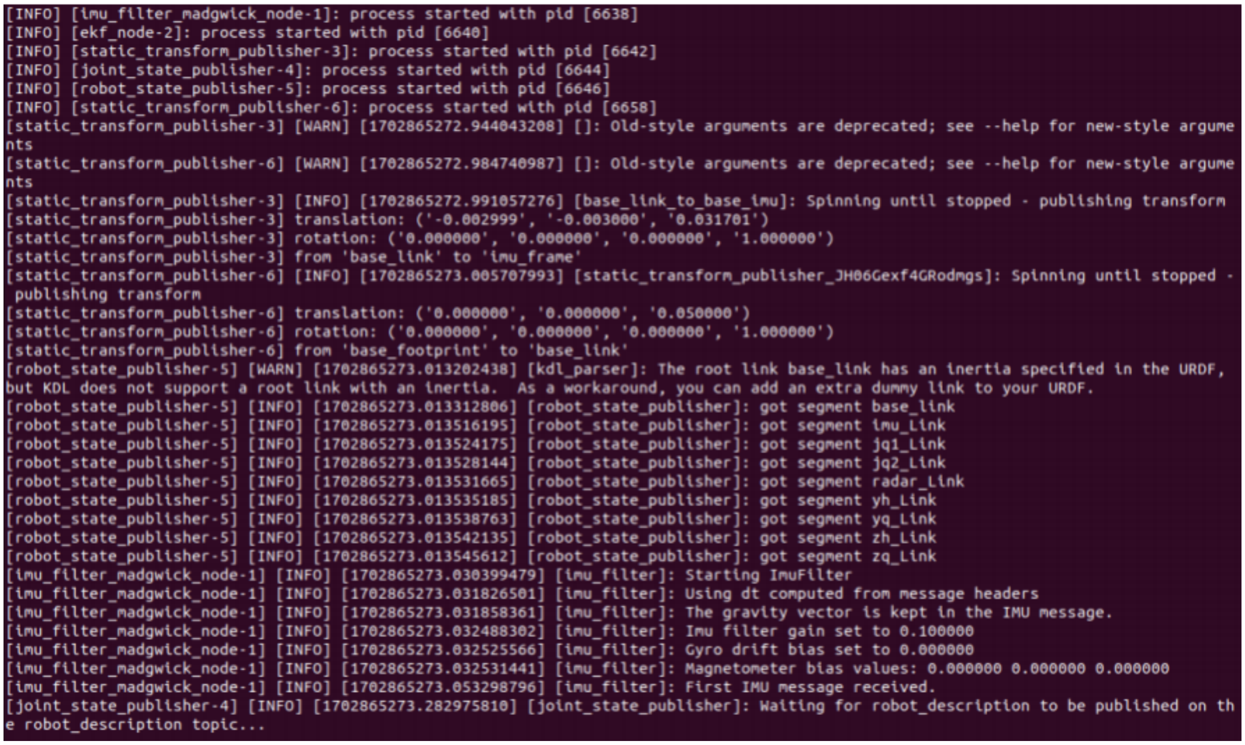


```
[1702630014.015846] info     | ProxyClient.cpp    | create_participant    | participant created    | client_key: 0x0B62A009, part
icipant_id: 0x000(1)
[1702630014.135363] info     | ProxyClient.cpp    | create_topic          | topic created          | client_key: 0x0B62A009, topi
c_id: 0x000(2), participant_id: 0x000(1)
[1702630014.223689] info     | ProxyClient.cpp    | create_publisher      | publisher created      | client_key: 0x0B62A009, publ
isher_id: 0x000(3), participant_id: 0x000(1)
[1702630014.415510] info     | ProxyClient.cpp    | create_datawriter     | datawriter created     | client_key: 0x0B62A009, data
writer_id: 0x000(5), publisher_id: 0x000(3)
[1702630014.428530] info     | ProxyClient.cpp    | create_topic          | topic created          | client_key: 0x0B62A009, topi
c_id: 0x001(2), participant_id: 0x000(1)
[1702630014.527190] info     | ProxyClient.cpp    | create_publisher      | publisher created      | client_key: 0x0B62A009, publ
isher_id: 0x001(3), participant_id: 0x000(1)
[1702630014.543889] info     | ProxyClient.cpp    | create_datawriter     | datawriter created     | client_key: 0x0B62A009, data
writer_id: 0x001(5), publisher_id: 0x001(3)
[1702630014.554490] info     | ProxyClient.cpp    | create_topic          | topic created          | client_key: 0x0B62A009, topi
c_id: 0x002(2), participant_id: 0x000(1)
[1702630014.737059] info     | ProxyClient.cpp    | create_publisher      | publisher created      | client_key: 0x0B62A009, publ
isher_id: 0x002(3), participant_id: 0x000(1)
[1702630014.755072] info     | ProxyClient.cpp    | create_datawriter     | datawriter created     | client_key: 0x0B62A009, data
writer_id: 0x002(5), publisher_id: 0x002(3)
[1702630014.818985] info     | ProxyClient.cpp    | create_topic          | topic created          | client_key: 0x0B62A009, topi
c_id: 0x003(2), participant_id: 0x000(1)
[1702630014.840001] info     | ProxyClient.cpp    | create_subscriber     | subscriber created     | client_key: 0x0B62A009, subs
criber_id: 0x000(4), participant_id: 0x000(1)
[1702630014.864010] info     | ProxyClient.cpp    | create_datareader     | datareader created     | client_key: 0x0B62A009, data
reader_id: 0x000(6), subscriber_id: 0x000(4)
[1702630014.959908] info     | ProxyClient.cpp    | create_topic          | topic created          | client_key: 0x0B62A009, topi
c_id: 0x004(2), participant_id: 0x000(1)
[1702630015.033537] info     | ProxyClient.cpp    | create_subscriber     | subscriber created     | client_key: 0x0B62A009, subs
criber_id: 0x001(4), participant_id: 0x000(1)
[1702630015.140350] info     | ProxyClient.cpp    | create_datareader     | datareader created     | client_key: 0x0B62A009, data
reader_id: 0x001(6), subscriber_id: 0x001(4)
[1702630015.158510] info     | ProxyClient.cpp    | create_topic          | topic created          | client_key: 0x0B62A009, topi
c_id: 0x005(2), participant_id: 0x000(1)
[1702630015.241039] info     | ProxyClient.cpp    | create_subscriber     | subscriber created     | client_key: 0x0B62A009, subs
criber_id: 0x002(4), participant_id: 0x000(1)
[1702630015.347393] info     | ProxyClient.cpp    | create_datareader     | datareader created     | client_key: 0x0B62A009, data
reader_id: 0x002(6), subscriber_id: 0x002(4)
```

4. Start the program

First, start the car to process the underlying data program and enter the terminal.

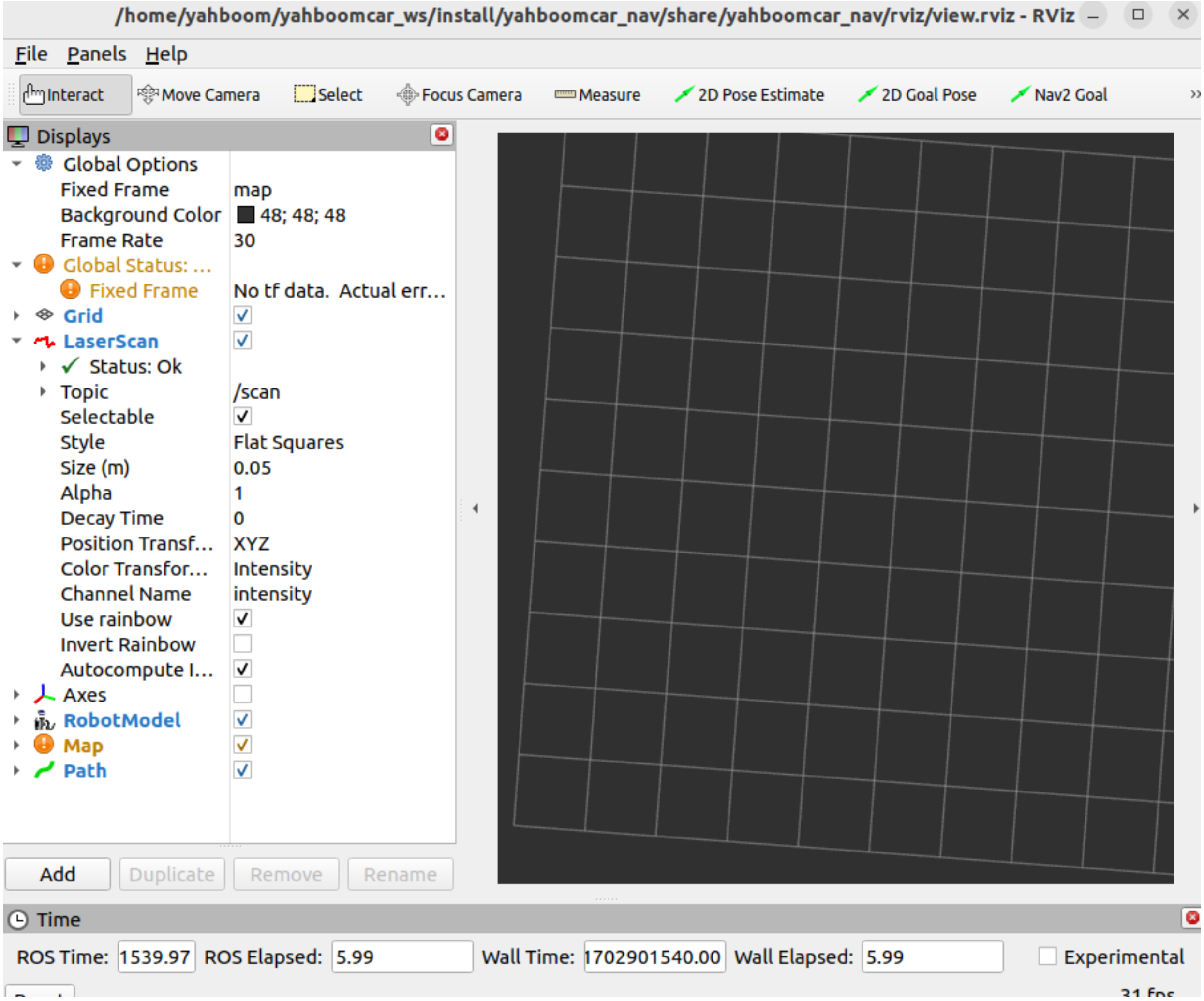
```
ros2 launch yahboomcar_bringup yahboomcar_bringup_launch.py
```



```
[INFO] [imu_filter_madgwick_node-1]: process started with pid [6638]
[INFO] [ekf_node-2]: process started with pid [6640]
[INFO] [static_transform_publisher-3]: process started with pid [6642]
[INFO] [joint_state_publisher-4]: process started with pid [6644]
[INFO] [robot_state_publisher-5]: process started with pid [6646]
[INFO] [static_transform_publisher-6]: process started with pid [6658]
[static_transform_publisher-3] [WARN] [1702865272.944043208] []: Old-style arguments are deprecated; see --help for new-style argume
nts
[static_transform_publisher-6] [WARN] [1702865272.984740987] []: Old-style arguments are deprecated; see --help for new-style argume
nts
[static_transform_publisher-3] [INFO] [1702865272.991057276] [base_link_to_base_imu]: Spinning until stopped - publishing transform
[static_transform_publisher-3] translation: ('-0.002999', '-0.003000', '0.031701')
[static_transform_publisher-3] rotation: ('0.000000', '0.000000', '0.000000', '1.000000')
[static_transform_publisher-3] from 'base_link' to 'imu_frame'
[static_transform_publisher-6] [INFO] [1702865273.005707993] [static_transform_publisher_JH06Gexf4GRodmgs]: Spinning until stopped -
 publishing transform
[static_transform_publisher-6] translation: ('0.000000', '0.000000', '0.050000')
[static_transform_publisher-6] rotation: ('0.000000', '0.000000', '0.000000', '1.000000')
[static_transform_publisher-6] from 'base_footprint' to 'base_link'
[robot_state_publisher-5] [WARN] [1702865273.013202438] [kdl_parser]: The root link base_link has an inertia specified in the URDF,
but 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-5] [INFO] [1702865273.013312806] [robot_state_publisher]: got segment base_link
[robot_state_publisher-5] [INFO] [1702865273.013516195] [robot_state_publisher]: got segment imu_Link
[robot_state_publisher-5] [INFO] [1702865273.013524175] [robot_state_publisher]: got segment jq1_Link
[robot_state_publisher-5] [INFO] [1702865273.013528144] [robot_state_publisher]: got segment jq2_Link
[robot_state_publisher-5] [INFO] [1702865273.013531665] [robot_state_publisher]: got segment radar_Link
[robot_state_publisher-5] [INFO] [1702865273.013535185] [robot_state_publisher]: got segment yh_Link
[robot_state_publisher-5] [INFO] [1702865273.013538763] [robot_state_publisher]: got segment yq_Link
[robot_state_publisher-5] [INFO] [1702865273.013542135] [robot_state_publisher]: got segment zh_Link
[robot_state_publisher-5] [INFO] [1702865273.013545612] [robot_state_publisher]: got segment zq_Link
[imu_filter_madgwick_node-1] [INFO] [1702865273.030399479] [imu_filter]: Starting ImuFilter
[imu_filter_madgwick_node-1] [INFO] [1702865273.031826501] [imu_filter]: Using dt computed from message headers
[imu_filter_madgwick_node-1] [INFO] [1702865273.031858361] [imu_filter]: The gravity vector is kept in the IMU message.
[imu_filter_madgwick_node-1] [INFO] [1702865273.032488302] [imu_filter]: Imu filter gain set to 0.100000
[imu_filter_madgwick_node-1] [INFO] [1702865273.032525566] [imu_filter]: Gyro drift bias set to 0.000000
[imu_filter_madgwick_node-1] [INFO] [1702865273.032531441] [imu_filter]: Magnetometer bias values: 0.000000 0.000000 0.000000
[imu_filter_madgwick_node-1] [INFO] [1702865273.053298796] [imu_filter]: First IMU message received.
[joint_state_publisher-4] [INFO] [1702865273.282975810] [joint_state_publisher]: Waiting for robot_description to be published on th
e robot_description topic...
```

Then, start rviz, visualize the mapping, and enter in the terminal.

```
ros2 launch yahboomcar_nav display_launch.py
```

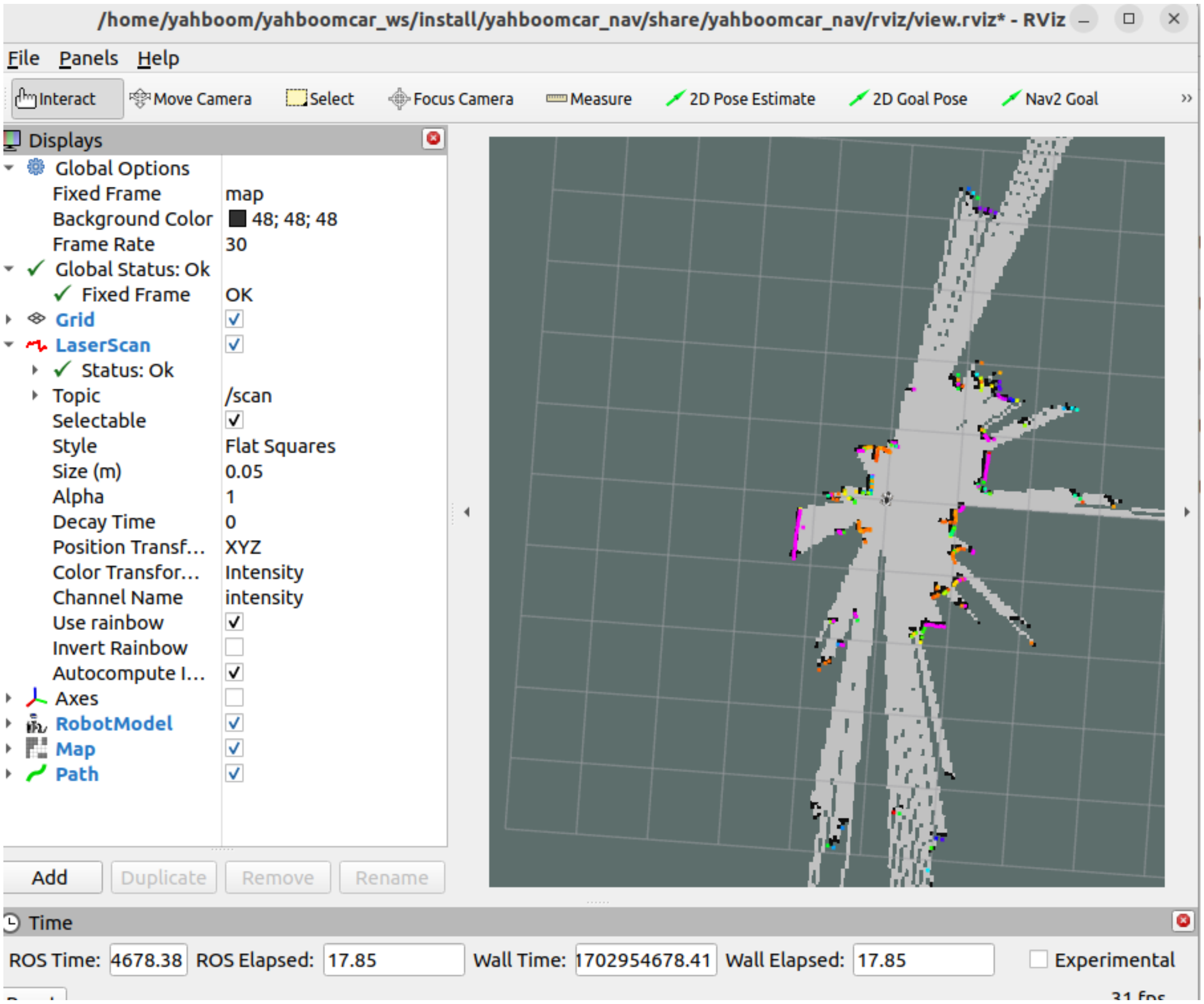


Global Options
Fixed Frame: map
Background Color: 48; 48; 48
Frame Rate: 30
Global Status: ...
Fixed Frame: No tf data. Actual err...
Grid ☑
LaserScan ☑
✓ Status: Ok
Topic: /scan
Selectable ☑
Style: Flat Squares
Size (m): 0.05
Alpha: 1
Decay Time: 0
Position Transf...: XYZ
Color Transfor...: Intensity
Channel Name: intensity
Use rainbow ☑
Invert Rainbow □
Autocompute I... ☑
Axes □
RobotModel ☑
Map ☑
Path ☑

ROS Time: 1539.97 ROS Elapsed: 5.99 Wall Time: 1702901540.00 Wall Elapsed: 5.99 □ Experimental

The mapping node has not been run yet, so there is no data. Next, run the mapping node and enter in the terminal,

```
ros2 launch yahboomcar_nav map_gmapping_launch.py
```



Global Options	
Fixed Frame	map
Background Color	■ 48; 48; 48
Frame Rate	30
✓ Global Status: Ok	
✓ Fixed Frame	OK
Grid	☑
LaserScan	☑
✓ Status: Ok	
Topic	/scan
Selectable	☑
Style	Flat Squares
Size (m)	0.05
Alpha	1
Decay Time	0
Position Transf...	XYZ
Color Transfor...	Intensity
Channel Name	intensity
Use rainbow	☑
Invert Rainbow	☐
Autocompute I...	☑
Axes	☐
RobotModel	☑
Map	☑
Path	☑

ROS Time: 4678.38 ROS Elapsed: 17.85 Wall Time: 1702954678.41 Wall Elapsed: 17.85 ☐ Experimental

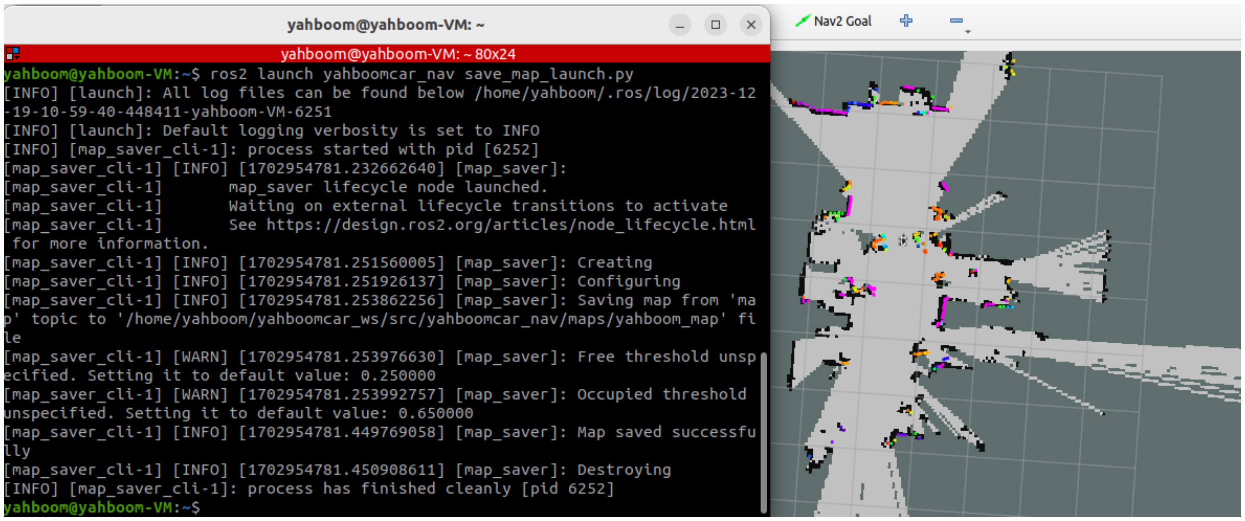
31 fps

Then run handle control or keyboard control, choose one of the two, terminal input,

```
#keyboard
ros2 run yahboomcar_ctrl yahboom_keyboard
#handle
ros2 run yahboomcar_ctrl yahboom_joy
ros2 run joy joy_node
```

Then control the car and slowly walk through the area that needs to be mapped. After the map is completed, enter the following command to save the map and enter it in the terminal.

```
ros2 launch yahboomcar_nav save_map_launch.py
```



```
yahboom@yahboom-VM:~$ ros2 launch yahboomcar_nav save_map_launch.py
[INFO] [launch]: All log files can be found below /home/yahboom/.ros/log/2023-12
-19-10-59-40-448411-yahboom-VM-6251
[INFO] [launch]: Default logging verbosity is set to INFO
[INFO] [map_saver_cli-1]: process started with pid [6252]
[map_saver_cli-1] [INFO] [1702954781.232662640] [map_saver]:
[map_saver_cli-1]       map_saver lifecycle node launched.
[map_saver_cli-1]       Waiting on external lifecycle transitions to activate
[map_saver_cli-1]       See https://design.ros2.org/articles/node_lifecycle.html
 for more information.
[map_saver_cli-1] [INFO] [1702954781.251560005] [map_saver]: Creating
[map_saver_cli-1] [INFO] [1702954781.251926137] [map_saver]: Configuring
[map_saver_cli-1] [INFO] [1702954781.253862256] [map_saver]: Saving map from 'ma
p' topic to '/home/yahboom/yahboomcar_ws/src/yahboomcar_nav/maps/yahboom_map' fi
le
[map_saver_cli-1] [WARN] [1702954781.253976630] [map_saver]: Free threshold unsp
ecified. Setting it to default value: 0.250000
[map_saver_cli-1] [WARN] [1702954781.253992757] [map_saver]: Occupied threshold
unspecified. Setting it to default value: 0.650000
[map_saver_cli-1] [INFO] [1702954781.449769058] [map_saver]: Map saved successfu
lly
[map_saver_cli-1] [INFO] [1702954781.450908611] [map_saver]: Destroying
[INFO] [map_saver_cli-1]: process has finished cleanly [pid 6252]
yahboom@yahboom-VM:~$
```

A map named yahboom_map will be saved. This map is saved in.

```
/home/yahboom/yahboomcar_ws/src/yahboomcar_nav/maps
```

Two files will be generated, one is yahboom_map.pgm and the other is yahboom_map.yaml. Take a look at the content of yaml.

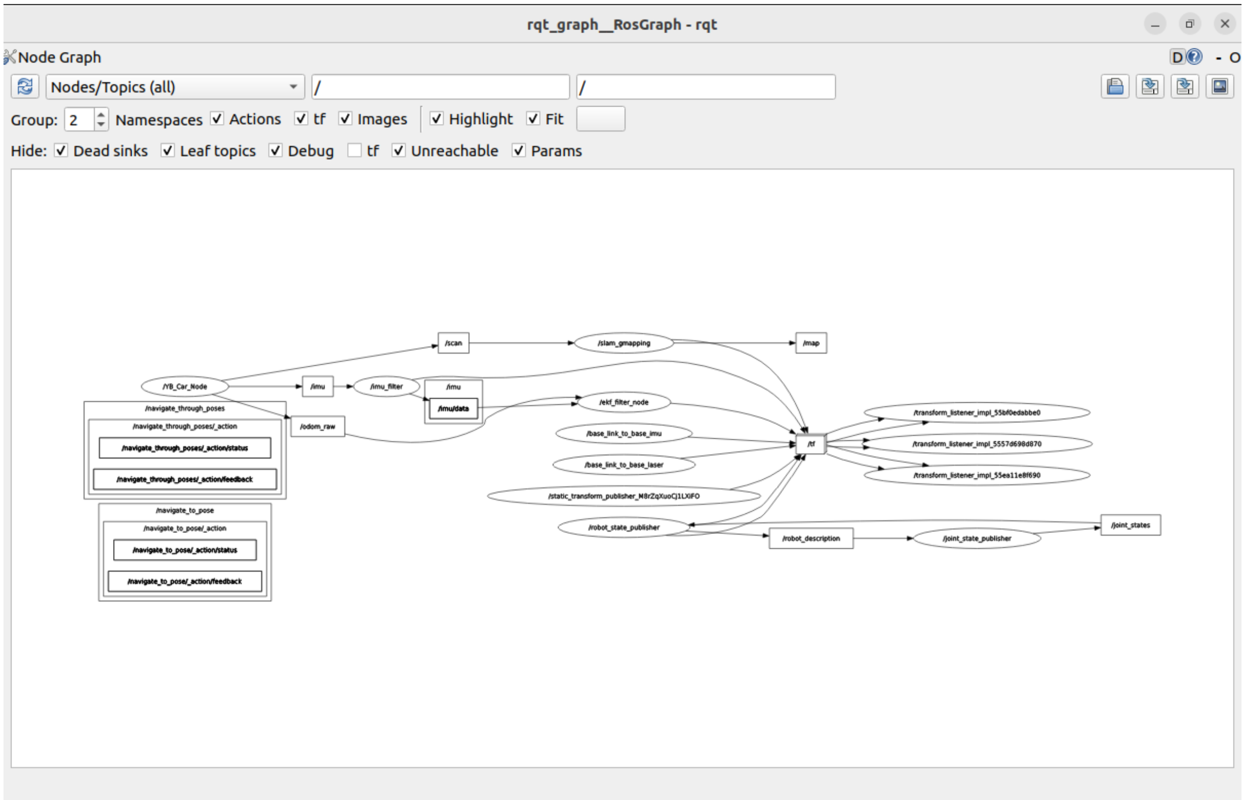
```
image: yahboom_map.pgm
mode: trinary
resolution: 0.05
origin: [-10, -10, 0]
negate: 0
occupied_thresh: 0.65
free_thresh: 0.25
```

- image：The picture representing the map, that is, yahboom_map.pgm
- mode：This attribute can be one of trinary, scale or raw, depending on the selected mode. trinary mode is the default mode.
- resolution：Map resolution, meters/pixel
- origin：The 2D pose (x, y, yaw) in the lower left corner of the map, where yaw is rotated counterclockwise (yaw=0 means no rotation). Many parts of the current system ignore the yaw value.
- negate：Whether to reverse the meaning of white/black and free/occupied (the interpretation of the threshold is not affected)
- occupied_thresh：Pixels with an occupancy probability greater than this threshold will be considered fully occupied.
- free_thresh：Pixels with an occupancy probability less than this threshold will be considered completely free.

5. View the node communication diagram

Terminal input,

```
ros2 run rqt_graph rqt_graph
```


rqt_graph__RosGraph - rqt
Node Graph
D
- O
/YB_Car_Node
/scan
/slam_gmapping
/map
/imu
/imu_filter
/imu
/imu/data
/odom_raw
/ekf_filter_node
/base_link_to_base_imu
/base_link_to_base_laser
/static_transform_publisher_M8rZqXuoCJ1LXiFO
/robot_state_publisher
/robot_description
/joint_state_publisher
/joint_states
/tf
/transform_listener_impl_55bf0edabbe0
/transform_listener_impl_5557d698d870
/transform_listener_impl_55ea11e8f690
/navigate_through_poses
/navigate_through_poses/_action
/navigate_through_poses/_action/status
/navigate_through_poses/_action/feedback
/navigate_to_pose
/navigate_to_pose/_action
/navigate_to_pose/_action/status
/navigate_to_pose/_action/feedback

Nodes/Topics (all) | / | /

Group: 2 Namespaces ☑ Actions ☑ tf ☑ Images ☑ Highlight ☑ Fit

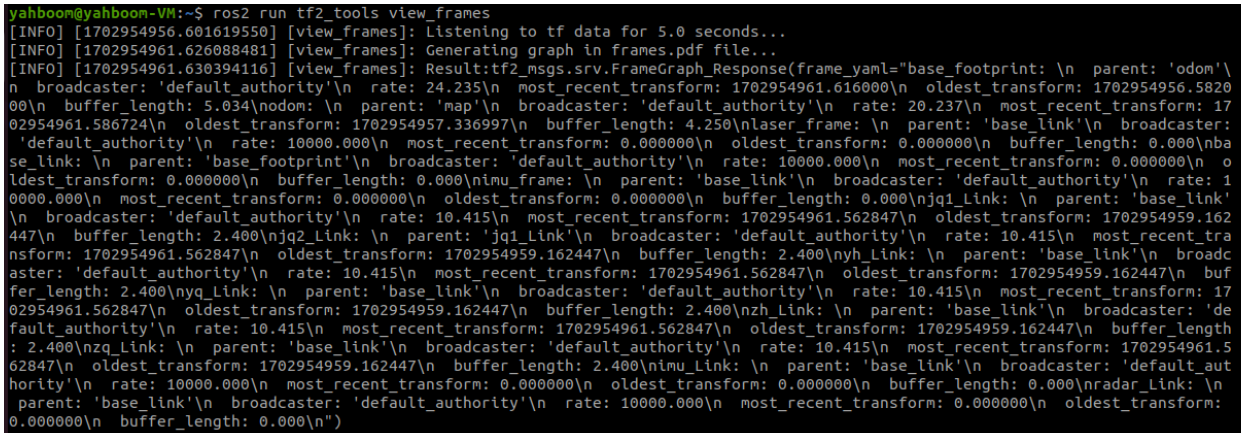
Hide: ☑ Dead sinks ☑ Leaf topics ☑ Debug □ tf ☑ Unreachable ☑ Params

If it is not displayed at first, select [Nodes/Topics(all)], and then click the refresh button in the upper left corner.

6. View TF tree

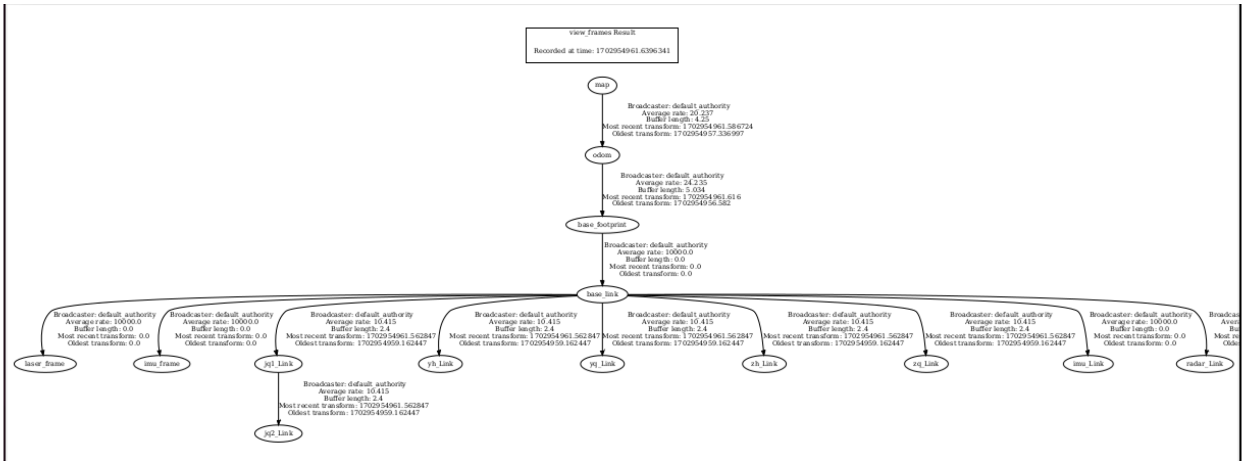
Terminal input,

```
ros2 run tf2_tools view_frames
```



```
yahboom@yahboom-VM:~$ ros2 run tf2_tools view_frames
[INFO] [1702954956.601619550] [view_frames]: Listening to tf data for 5.0 seconds...
[INFO] [1702954961.626088481] [view_frames]: Generating graph in frames.pdf file...
[INFO] [1702954961.630394116] [view_frames]: Result:tf2_msgs.srv.FrameGraph_Response(frame_yaml="base_footprint: \n  parent: 'odom'\
n  broadcaster: 'default_authority'\n  rate: 24.235\n  most_recent_transform: 1702954961.616000\n  oldest_transform: 1702954956.5820
00\n  buffer_length: 5.034\nodom: \n  parent: 'map'\n  broadcaster: 'default_authority'\n  rate: 20.237\n  most_recent_transform: 17
02954961.586724\n  oldest_transform: 1702954957.336997\n  buffer_length: 4.250\nlaser_frame: \n  parent: 'base_link'\n  broadcaster:
 'default_authority'\n  rate: 10000.000\n  most_recent_transform: 0.000000\n  oldest_transform: 0.000000\n  buffer_length: 0.000\nba
se_link: \n  parent: 'base_footprint'\n  broadcaster: 'default_authority'\n  rate: 10000.000\n  most_recent_transform: 0.000000\n  o
ldest_transform: 0.000000\n  buffer_length: 0.000\nimu_frame: \n  parent: 'base_link'\n  broadcaster: 'default_authority'\n  rate: 1
0000.000\n  most_recent_transform: 0.000000\n  oldest_transform: 0.000000\n  buffer_length: 0.000\njq1_Link: \n  parent: 'base_link'
\n  broadcaster: 'default_authority'\n  rate: 10.415\n  most_recent_transform: 1702954961.562847\n  oldest_transform: 1702954959.162
447\n  buffer_length: 2.400\njq2_Link: \n  parent: 'jq1_Link'\n  broadcaster: 'default_authority'\n  rate: 10.415\n  most_recent_tra
nsform: 1702954961.562847\n  oldest_transform: 1702954959.162447\n  buffer_length: 2.400\nyh_Link: \n  parent: 'base_link'\n  broadc
aster: 'default_authority'\n  rate: 10.415\n  most_recent_transform: 1702954961.562847\n  oldest_transform: 1702954959.162447\n  buf
fer_length: 2.400\nyq_Link: \n  parent: 'base_link'\n  broadcaster: 'default_authority'\n  rate: 10.415\n  most_recent_transform: 17
02954961.562847\n  oldest_transform: 1702954959.162447\n  buffer_length: 2.400\nzh_Link: \n  parent: 'base_link'\n  broadcaster: 'de
fault_authority'\n  rate: 10.415\n  most_recent_transform: 1702954961.562847\n  oldest_transform: 1702954959.162447\n  buffer_length
: 2.400\nzq_Link: \n  parent: 'base_link'\n  broadcaster: 'default_authority'\n  rate: 10.415\n  most_recent_transform: 1702954961.5
62847\n  oldest_transform: 1702954959.162447\n  buffer_length: 2.400\nimu_Link: \n  parent: 'base_link'\n  broadcaster: 'default_aut
hority'\n  rate: 10000.000\n  most_recent_transform: 0.000000\n  oldest_transform: 0.000000\n  buffer_length: 0.000\nradar_Link: \n
 parent: 'base_link'\n  broadcaster: 'default_authority'\n  rate: 10000.000\n  most_recent_transform: 0.000000\n  oldest_transform:
0.000000\n  buffer_length: 0.000\n")
```

After the operation is completed, two files will be generated in the terminal directory, namely .gv and .pdf files. The pdf file is the TF tree.


view_frames Result
Recorded at time: 1702954961.6396341
map
Broadcaster: default_authority
Average rate: 20.237
Buffer length: 4.25
Most recent transform: 1702954961.586724
Oldest transform: 1702954957.336997
odom
Broadcaster: default_authority
Average rate: 24.235
Buffer length: 5.034
Most recent transform: 1702954961.616
Oldest transform: 1702954956.582
base_footprint
Broadcaster: default_authority
Average rate: 10000.0
Buffer length: 0.0
Most recent transform: 0.0
Oldest transform: 0.0
base_link
Broadcaster: default_authority
Average rate: 10000.0
Buffer length: 0.0
Most recent transform: 0.0
Oldest transform: 0.0
Broadcaster: default_authority
Average rate: 10000.0
Buffer length: 0.0
Most recent transform: 0.0
Oldest transform: 0.0
Broadcaster: default_authority
Average rate: 10.415
Buffer length: 2.4
Most recent transform: 1702954961.562847
Oldest transform: 1702954959.162447
Broadcaster: default_authority
Average rate: 10.415
Buffer length: 2.4
Most recent transform: 1702954961.562847
Oldest transform: 1702954959.162447
Broadcaster: default_authority
Average rate: 10.415
Buffer length: 2.4
Most recent transform: 1702954961.562847
Oldest transform: 1702954959.162447
Broadcaster: default_authority
Average rate: 10.415
Buffer length: 2.4
Most recent transform: 1702954961.562847
Oldest transform: 1702954959.162447
Broadcaster: default_authority
Average rate: 10.415
Buffer length: 2.4
Most recent transform: 1702954961.562847
Oldest transform: 1702954959.162447
Broadcaster: default_authority
Average rate: 10000.0
Buffer length: 0.0
Most recent transform: 0.0
Oldest transform: 0.0
laser_frame
imu_frame
jq1_Link
yh_Link
yq_Link
zh_Link
zq_Link
imu_Link
radar_Link
Broadcaster: default_authority
Average rate: 10.415
Buffer length: 2.4
Most recent transform: 1702954961.562847
Oldest transform: 1702954959.162447
jq2_Link

7. Code analysis

Here we only describe map_gmapping_launch.py for mapping. The file path is as follows

```
/home/yahboom/yahboomcar_ws/src/yahboomcar_nav/launch
```

map_gmapping_launch.py

```python
from launch import LaunchDescription
from launch_ros.actions import Node
import os
from launch.actions import IncludeLaunchDescription
from launch.launch_description_sources import PythonLaunchDescriptionSource
from ament_index_python.packages import get_package_share_directory


def generate_launch_description():
    slam_gmapping_launch = IncludeLaunchDescription(
        PythonLaunchDescriptionSource([os.path.join(
        get_package_share_directory('slam_gmapping'), 'launch'),
         '/slam_gmapping.launch.py'])
    )

    base_link_to_laser_tf_node = Node(
     package='tf2_ros',
     executable='static_transform_publisher',
     name='base_link_to_base_laser',
     arguments=['-0.0046412', '0' ,
'0.094079','0','0','0','base_link','laser_frame']
    )

    return LaunchDescription([slam_gmapping_launch,base_link_to_laser_tf_node])
```

A launch file-slam_gmapping_launch and a node for publishing static transformation-base_link_to_laser_tf_node are started here. Take a detailed look at slam_gmapping_launch, the file is as follows

```
/home/yahboom/gmapping_ws/src/slam_gmapping/launch
```

slam_gmapping.launch.py,

```
from launch import LaunchDescription
from launch.substitutions import EnvironmentVariable
import launch.actions
import launch_ros.actions
import os
from ament_index_python.packages import get_package_share_directory

def generate_launch_description():
    return LaunchDescription([
        launch_ros.actions.Node(
            package='slam_gmapping',
            executable='slam_gmapping',
            output='screen',
            parameters=[os.path.join(get_package_share_directory("slam_gmapping"),
"params", "slam_gmapping.yaml")]),
    ])
```

The slam_gmapping node is started here, and the slam_gmapping.yaml parameter file is loaded. The file is located (taking the supporting virtual machine as an example),

```
/home/yahboom/gmapping_ws/src/slam_gmapping/params
```

slam_gmapping.yaml

```
/slam_gmapping:
  ros__parameters:
    angularUpdate: 0.5
    astep: 0.05
    base_frame: base_footprint
    map_frame: map
    odom_frame: odom
    delta: 0.05
    iterations: 5
    kernelSize: 1
    lasamplerange: 0.005
    lasamplestep: 0.005
    linearUpdate: 1.0
    llsamplerange: 0.01
    llsamplestep: 0.01
    lsigma: 0.075
    lskip: 0
    lstep: 0.05
    map_update_interval: 5.0
    maxRange: 6.0
    maxUrange: 4.0
    minimum_score: 0.0
    occ_thresh: 0.25
    ogain: 3.0
    particles: 30
    qos_overrides:
      /parameter_events:
        publisher:
          depth: 1000
          durability: volatile
          history: keep_all
          reliability: reliable
      /tf:
        publisher:
          depth: 1000
          durability: volatile
          history: keep_last
          reliability: reliable
    resampleThreshold: 0.5
    sigma: 0.05
    srr: 0.1
    srt: 0.2
    str: 0.1
    stt: 0.2
    temporalUpdate: 1.0
    transform_publish_period: 0.05
    use_sim_time: false
    xmax: 10.0
    xmin: -10.0
    ymax: 10.0
    ymin: -10.0
```