Gmapping建图

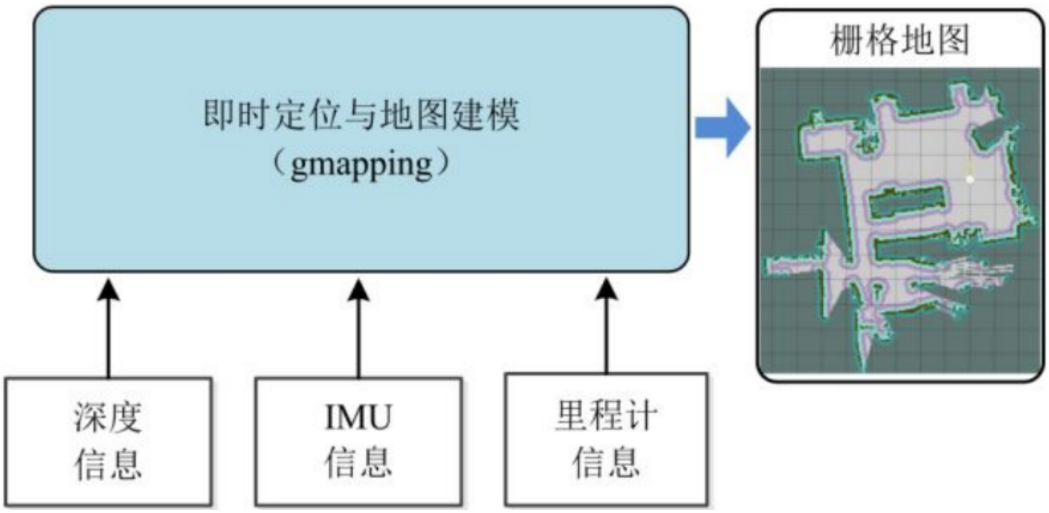
注：虚拟机需要与小车处在同一个局域网下，且ROS_DOMAIN_ID，需要一致，可以查看【使用前必看】来设置板子上的IP和ROS_DOMAIN_ID。

1、Gmapping简介

- gmapping只适用于单帧二维激光点数小于1440的点，如果单帧激光点数大于1440，那么就会出【[mapping-4] process has died】 这样的问题。
- Gmapping是基于滤波SLAM框架的常用开源SLAM算法。
- Gmapping基于RBpf粒子滤波算法，即时定位和建图过程分离，先进行定位再进行建图。
- Gmapping在RBpf算法上做了两个主要的改进：改进提议分布和选择性重采样。

优点：Gmapping可以实时构建室内地图，在构建小场景地图所需的计算量较小且精度较高。

缺点：随着场景增大所需的粒子增加，因为每个粒子都携带一幅地图，因此在构建大地图时所需内存和计算量都会增加。因此不适合构建大场景地图。并且没有回环检测，因此在回环闭合时可能会造成地图错位，虽然增加粒子数目可以使地图闭合但是以增加计算量和内存为代价。



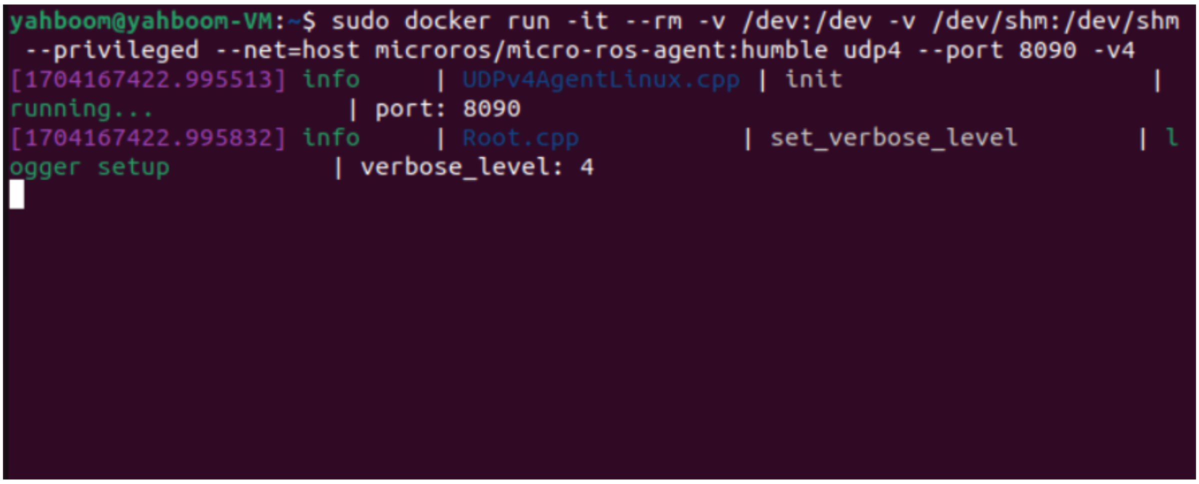
2、程序功能说明

小车连接上代理，运行程序，rviz中会显示建图的界面，用键盘或者手柄去控制小车运动，直到建完图。然后运行保存地图的指令保存地图。

3、启动并连接代理

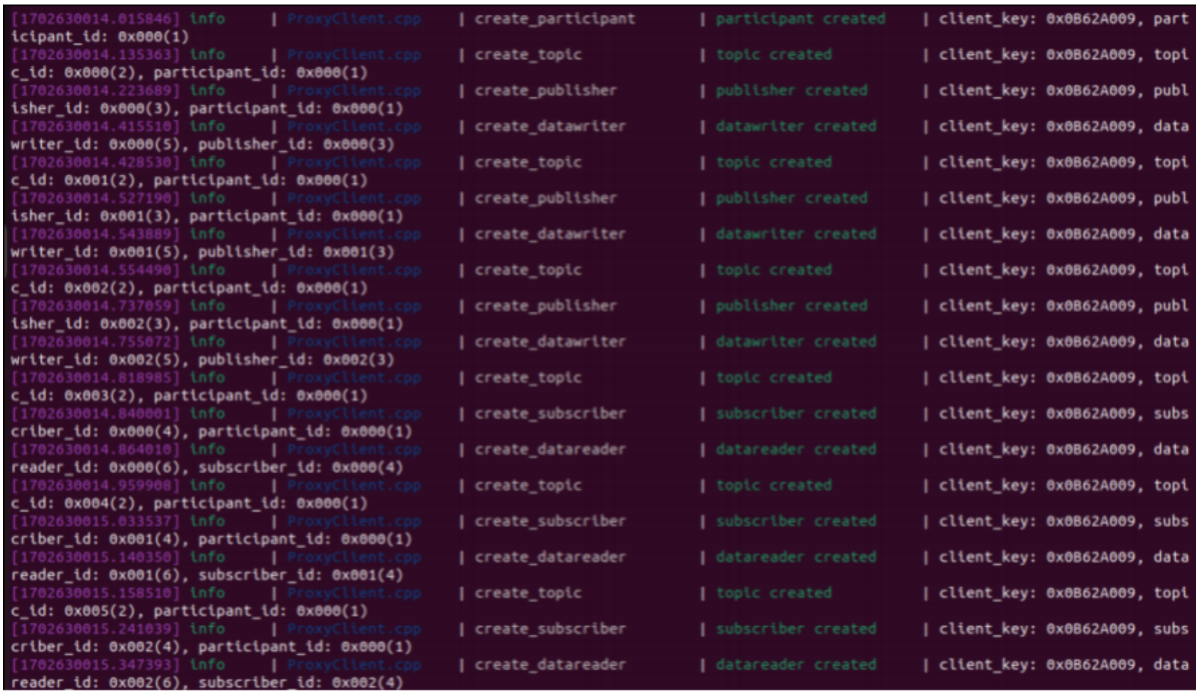
以配套虚拟机为例，输入以下指令启动代理，

```
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
```

然后，打开小车开关，等待小车连接上代理，连接成功如下图所示，



```
[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)
```

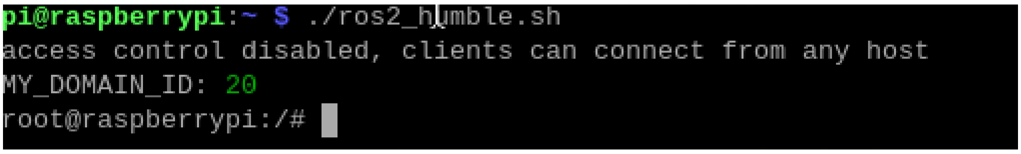
4、启动程序

4.1运行指令

如果是树莓派桌面版本和jetson nano桌面版本，需要提前进入docker，终端输入，

```
sh ros2_humble.sh
```

出现以下界面就是进入docker成功，



```
pi@raspberrypi:~ $ ./ros2_humble.sh
access control disabled, clients can connect from any host
MY_DOMAIN_ID: 20
root@raspberrypi:/#
```

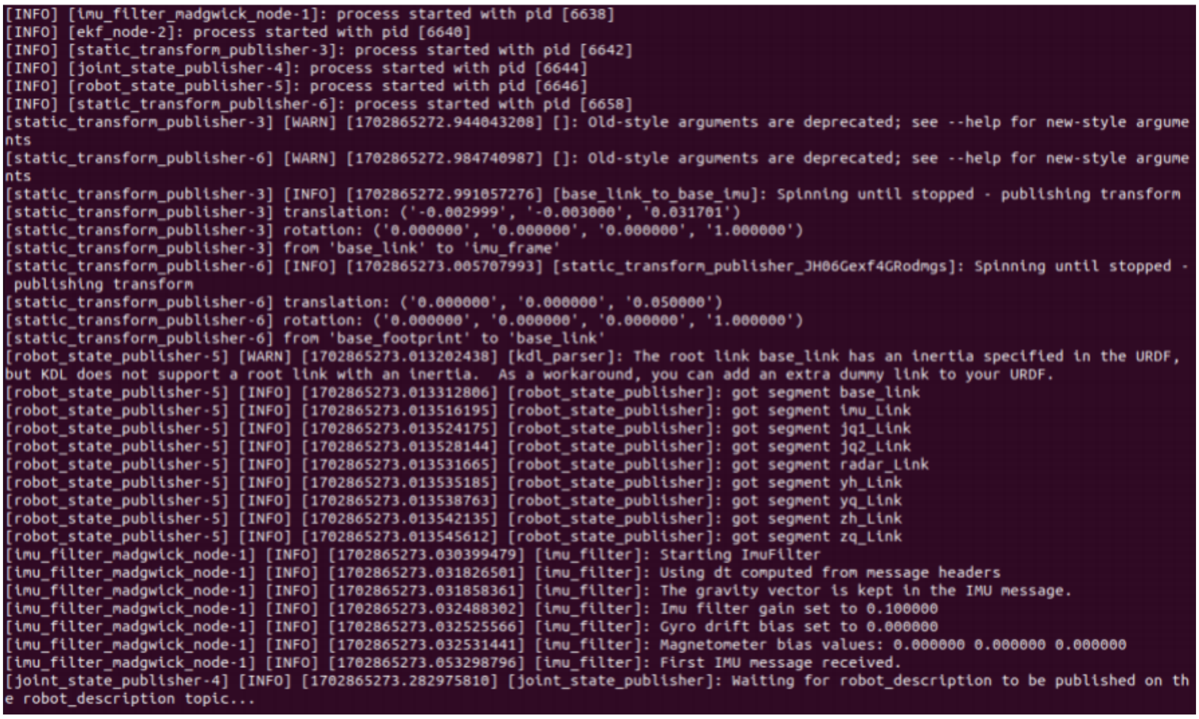
之后在docker里输入，**（查看【docker环境】章节，如何进入同一个docker终端）**

```
ros2 launch yahboomcar_bringup yahboomcar_bringup_launch.py      #底层数据程序
ros2 launch yahboomcar_nav display_launch.py   #启动rviz，可视化建图
ros2 launch yahboomcar_nav map_gmapping_launch.py    #建图节点
#键盘
ros2 run yahboomcar_ctrl yahboom_keyboard
#手柄
ros2 run yahboomcar_ctrl yahboom_joy
ros2 run joy joy_node
#保存地图
ros2 launch yahboomcar_nav save_map_launch.py
```

以配套的虚拟机为例，终端输入，

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

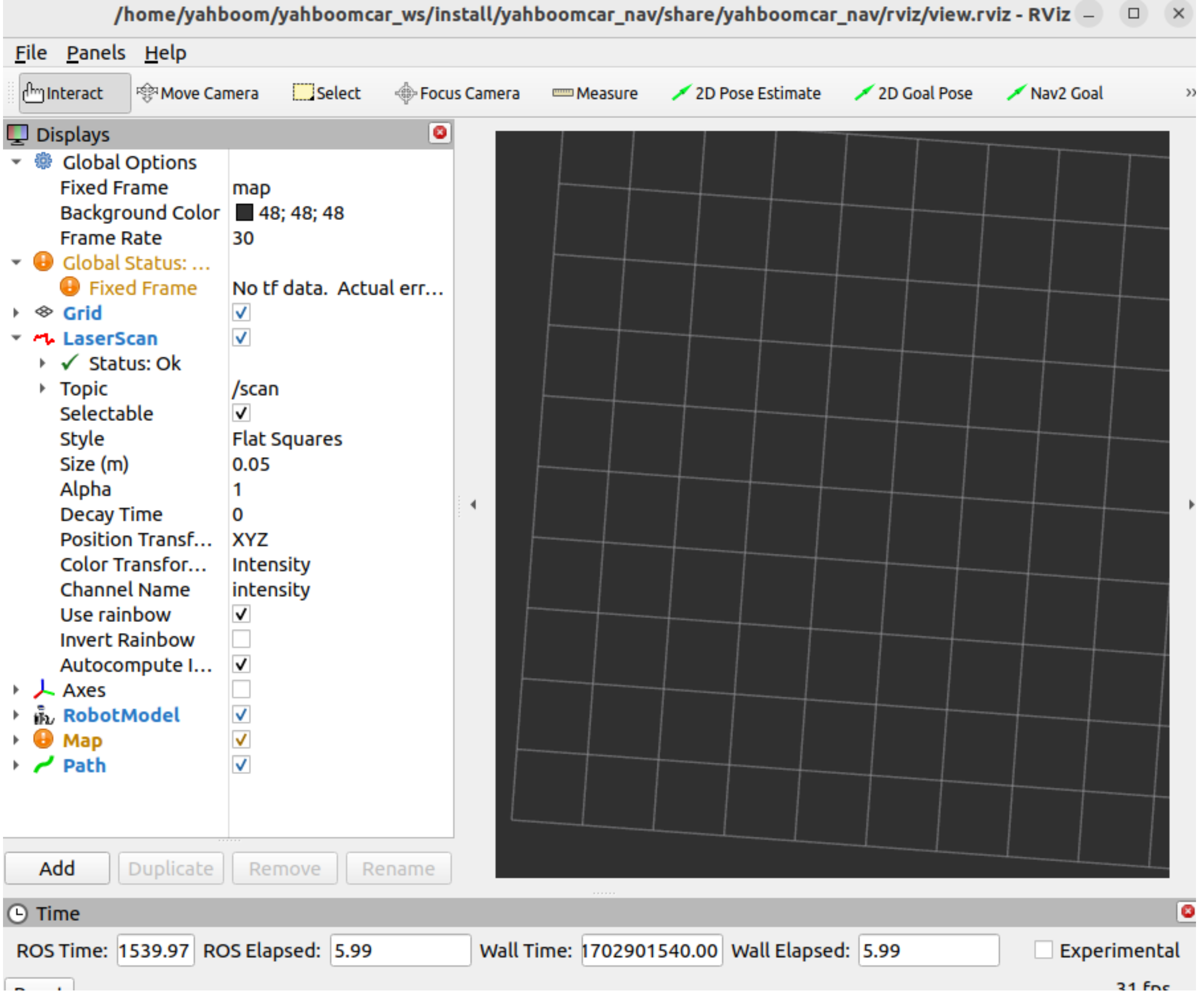
首先启动小车处理底层数据程序，



```
[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...
```

然后，启动rviz，可视化建图，终端输入，

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



/home/yahboom/yahboomcar_ws/install/yahboomcar_nav/share/yahboomcar_nav/rviz/view.rviz - RViz

Displays

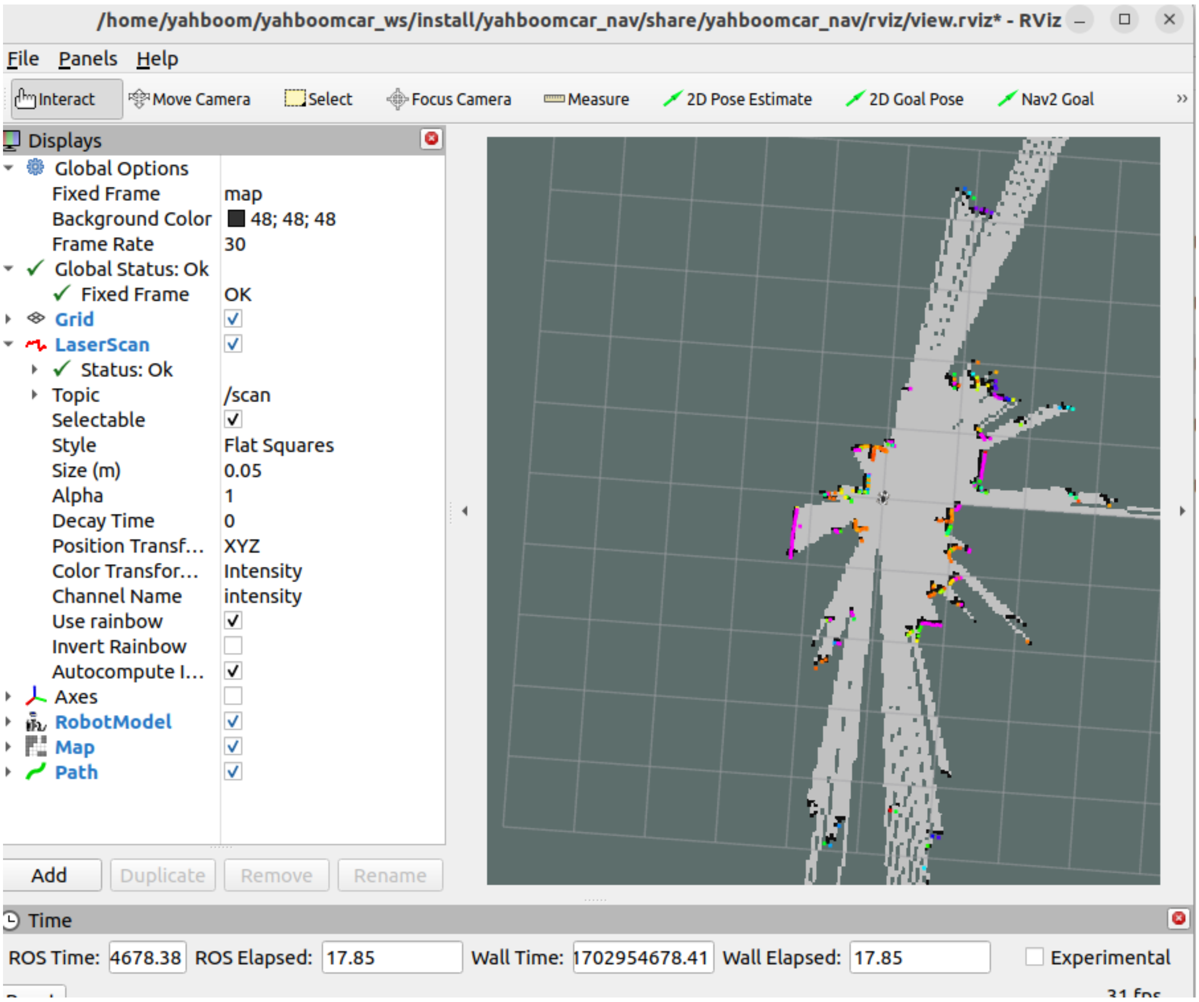
- Global Options
  - Fixed Frame: map
  - Background Color: ■ 48; 48; 48
  - Frame Rate: 30
- Global Status: ...
  - Fixed Frame: No tf data. Actual err...
- Grid ☑
- LaserScan ☑
  - ✓ Status: Ok
  - Topic: /scan
  - Selectable: ☑
  - Style: Flat Squares
  - Size (m): 0.05
  - Alpha: 1
  - Decay Time: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

Time

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

此时还没运行建图节点，所以没有数据。接下来运行建图节点，终端输入，

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



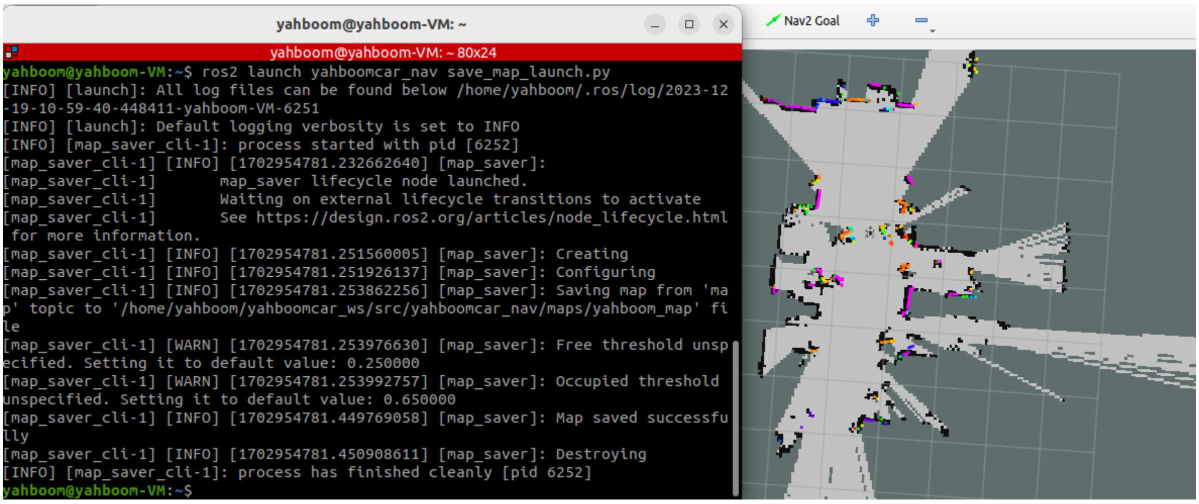
```
Global Options
  Fixed Frame            map
  Background Color       48; 48; 48
  Frame Rate             30
✓ Global Status: Ok
  ✓ Fixed Frame          OK
Grid                     ☑
LaserScan                ☑
  ✓ Status: Ok
  Topic                  /scan
  Selectable             ☑
  Style                  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
```

然后运行手柄控制或者键盘控制，二选一，终端输入，

```
#键盘
ros2 run yahboomcar_ctrl yahboom_keyboard
#手柄
ros2 run yahboomcar_ctrl yahboom_joy
ros2 run joy joy_node
```

然后控制小车，缓慢的走完需要建图的区域，建图完毕后，输入以下指令保存地图，终端输入，

```
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:~$
```

会保存一个命名为yahboom_map的地图，这个地图保存在，

以配套的虚拟机为例代码路径：

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

jetson nano代码路径：

```
/root/yahboomcar_ws/src/yahboomcar_nav/maps
```

树莓派代码路径：

```
/root/yahboomcar_ws/src/yahboomcar_nav/maps
```

会有两个文件生成，一个是yahboom_map.pgm，一个是yahboom_map.yaml，看下yaml的内容，

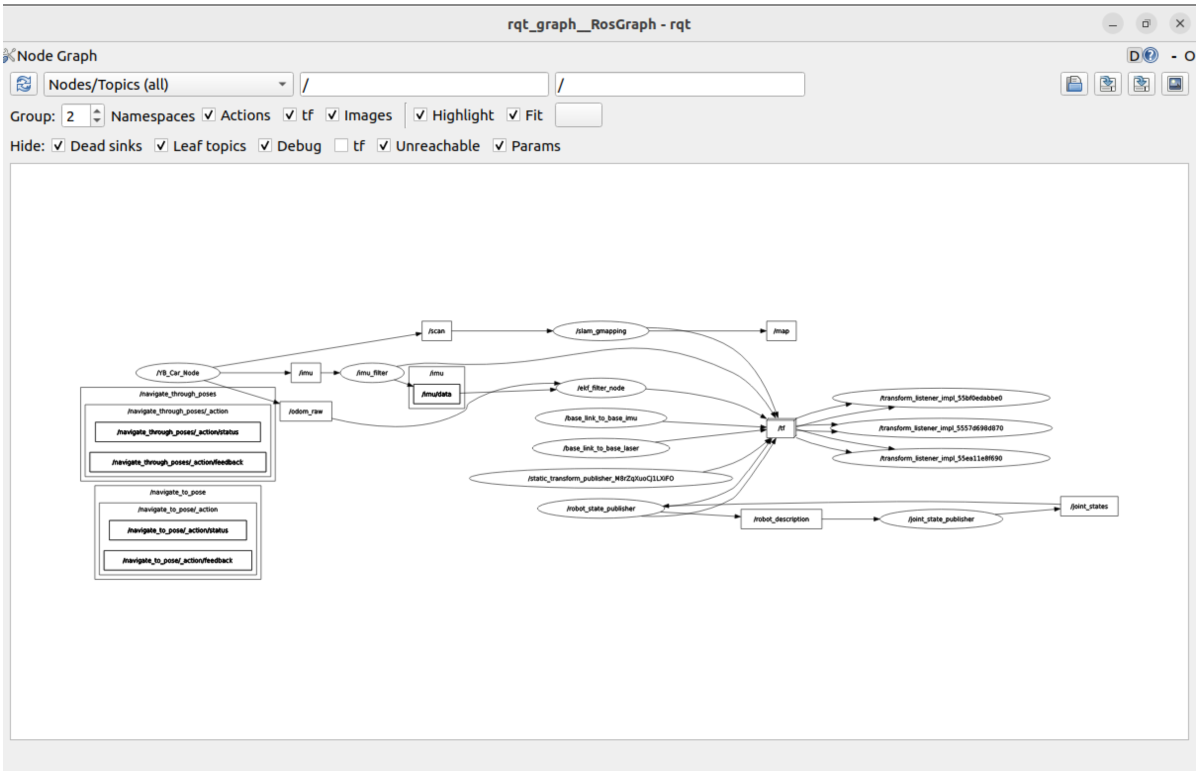
```
image: yahboom_map.pgm
mode: trinary
resolution: 0.05
origin: [-10, -10, 0]
negate: 0
occupied_thresh: 0.65
free_thresh: 0.25
```

- image：表示地图的图片，也就是yahboom_map.pgm
- mode：该属性可以是trinary、scale或者raw之一，取决于所选择的mode，trinary模式是默认模式
- resolution：地图的分辨率， 米/像素
- origin：地图左下角的 2D 位姿(x,y,yaw), 这里的yaw是逆时针方向旋转的（yaw=0 表示没有旋转）。目前系统中的很多部分会忽略yaw值。
- negate：是否颠倒 白/黑 、自由/占用 的意义（阈值的解释不受影响）
- occupied_thresh：占用概率大于这个阈值的的像素，会被认为是完全占用。
- free_thresh：占用概率小于这个阈值的的像素，会被认为是完全自由。

5、查看节点通讯图

终端输入，

```
ros2 run rqt_graph rqt_graph
```

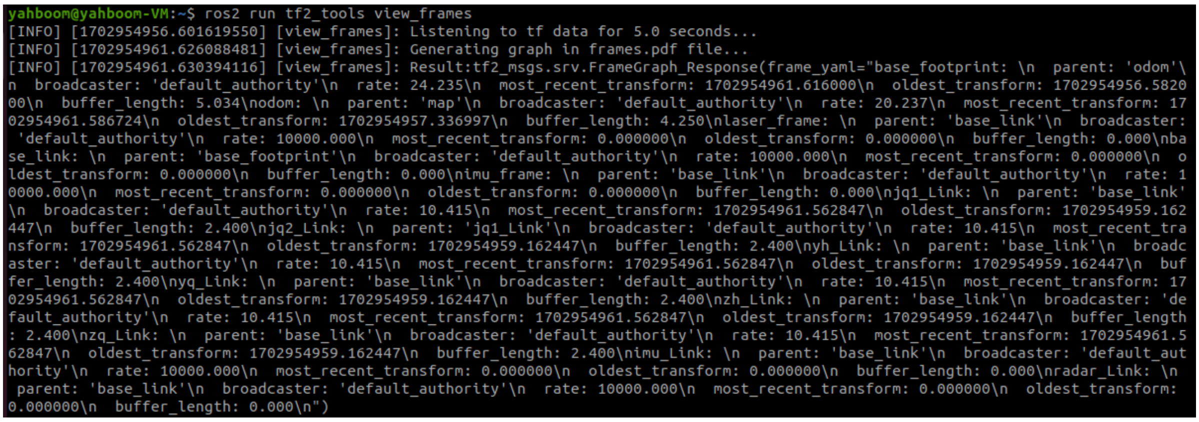


如果一开始没有显示，选择【Nodes/Topics(all)】，然后点击左上角的刷新按钮。

6、查看TF树

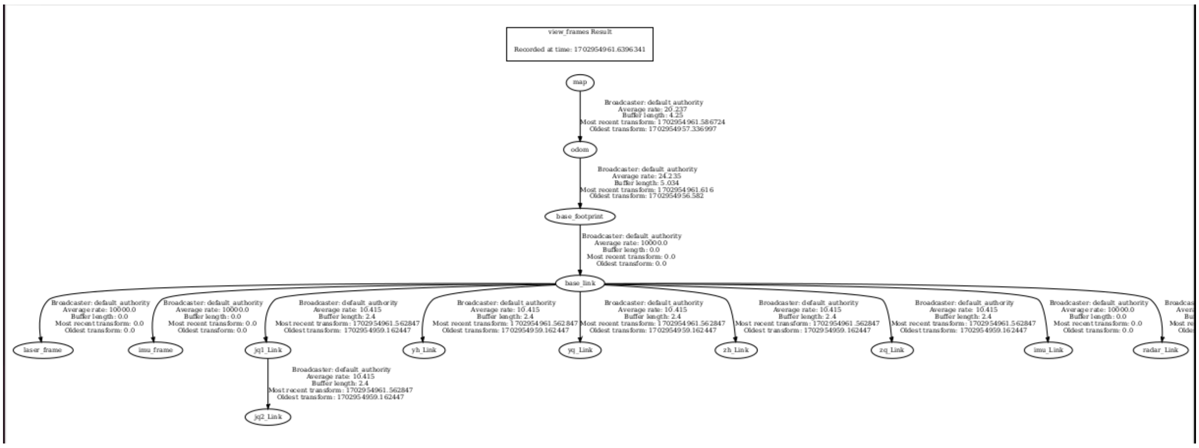
终端输入，

```
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")
```

运行完毕后，会在终端的目录下生成两个文件分别是.gv和.pdf文件，其中的pdf文件就是TF树。



7、代码解析

以虚拟机为例，这里只说明建图的map_gmapping_launch.py，这个文件路径是，

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

jetson nano代码路径：

```
/root/yahboomcar_ws/src/yahboomcar_nav/launch
```

树莓派代码路径：

```
/root/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])
```

这里启动了一个launch文件-slam_gmapping_launch和一个发布静态变换的节点-base_link_to_laser_tf_node。

以虚拟机为例，详细看下slam_gmapping_launch，该文件位于，

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

jetson nano代码路径：

```
/root/gmapping_ws/src/slam_gmapping/launch
```

树莓派代码路径：

```
/root/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")]),
    ])
```

这里启动了slam_gmapping的节点，加载了slam_gmapping.yaml参数文件，

该文件位于（以配套虚拟机为例），

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

jetson nano代码路径：

```
/root/gmapping_ws/src/slam_gmapping/params
```

树莓派代码路径：

```
/root/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
```