

2.Handheld lidar mapping

This function needs to start the program in the slam_gmapping function package. The source code is located in the [rplidar_ws] source code file.

Here we use the supporting virtual machine to explain how to start the program. If you want to put it on your own motherboard, you need to put [slam_gmapping] and [openslam_gmapping] in the src directory of the workspace to compile.

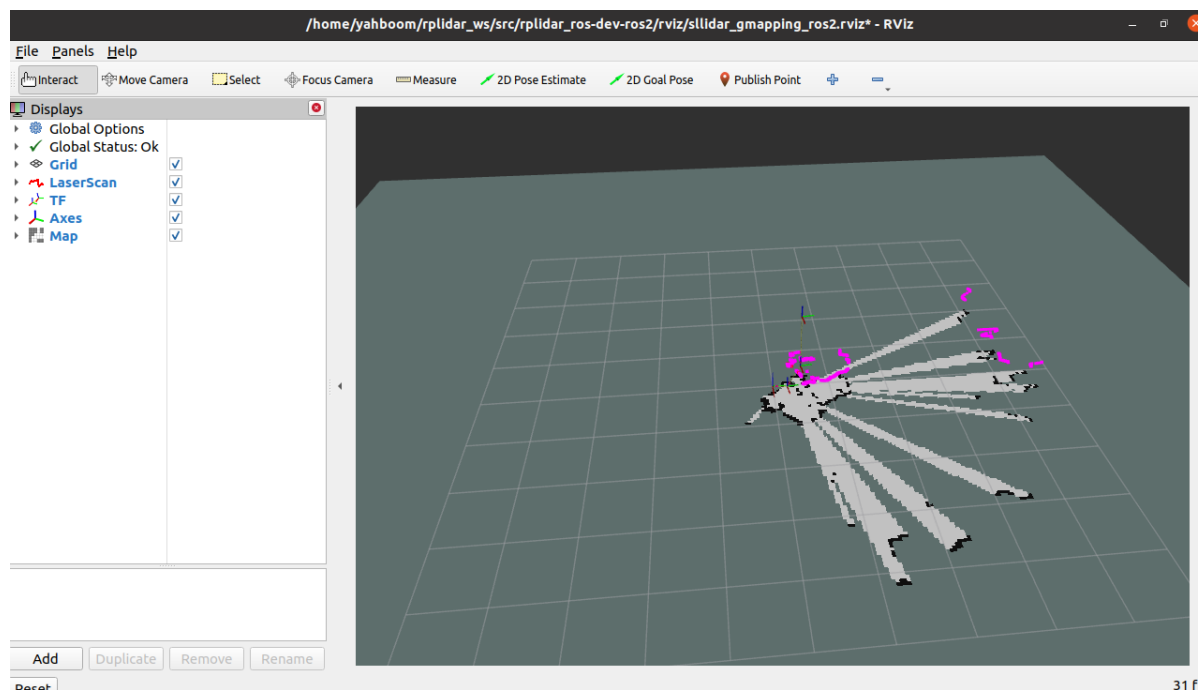
2.1 Start gmapping mapping

Input following command:

```
ros2 launch rplidar_ros test_gmapping.launch.py
```

```
[slam_gmapping-2] m_count 0
[slam_gmapping-2] Registering First Scan
[slam_gmapping-2] [INFO] [1699613716.598416796] [slam_gmapping]: Initialization
complete
[slam_gmapping-2] Laser Pose= 0.1 0.2 0
[rviz2-5] [INFO] [1699613716.648205807] [rviz2]: Trying to create a map of size
384 x 384 using 1 swatches
[rviz2-5] [ERROR] [1699613716.652811756] [rviz2]: Vertex Program:rviz/glsl120/in
dexed_8bit_image.vert Fragment Program:rviz/glsl120/indexed_8bit_image.frag GLSL
link result :
[rviz2-5] active samplers with a different type refer to the same texture image
unit
[slam_gmapping-2] update frame 15
[slam_gmapping-2] update ld=9.3095e-17 ad=0
[slam_gmapping-2] m_count 1
[slam_gmapping-2] Laser Pose= 0.1 0.2 0
[slam_gmapping-2] Average Scan Matching Score=984.978
[slam_gmapping-2] neff= 30
[slam_gmapping-2] Registering Scans:Done
[slam_gmapping-2] update frame 35
[slam_gmapping-2] update ld=0 ad=0
[slam_gmapping-2] m_count 2
[slam_gmapping-2] Laser Pose= 0.1 0.2 0
```

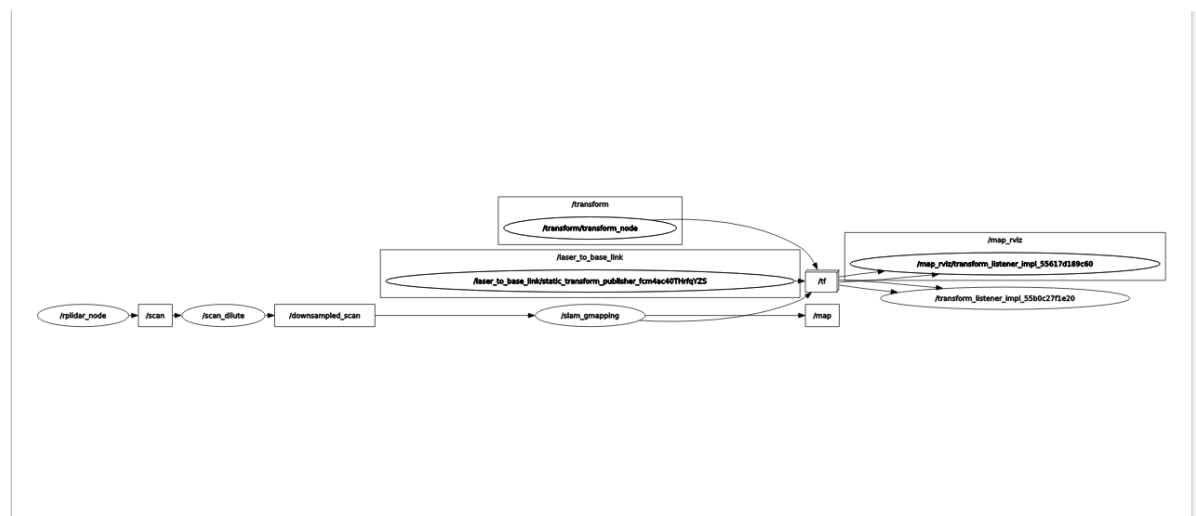
rviz displays as follows:



2.2 View node communication

Input following command:

```
ros2 run rqt_graph rqt_graph
```



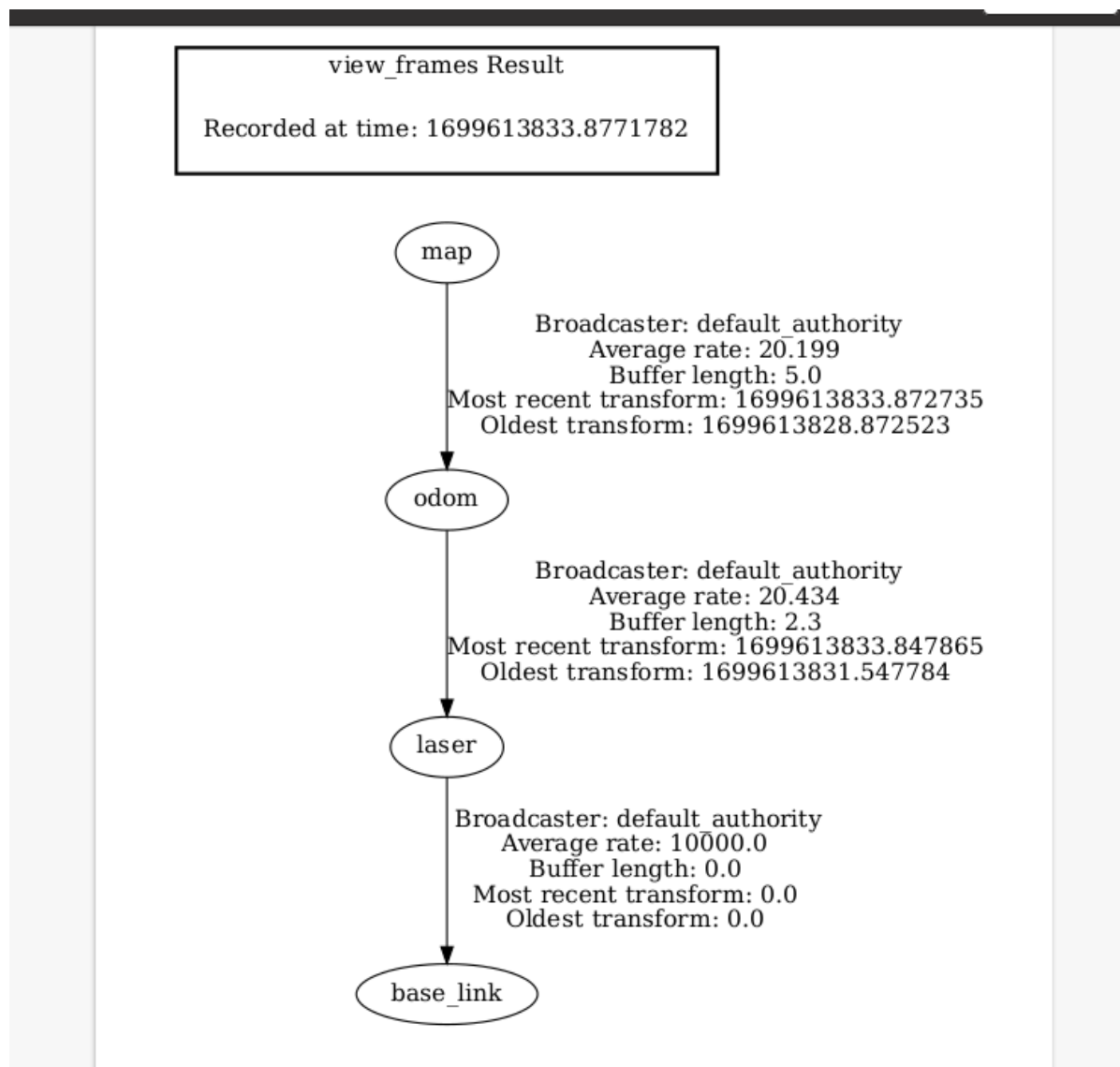
2.3 View TF tree

Input following command:

```
ros2 run tf2_tools view_frames.py
```

```
yahboom@VM:~$ ros2 run tf2_tools view_frames.py
[INFO] [1699500741.610795137] [view_frames]: Listening to tf data during 5 seconds...
[INFO] [1699500746.652013123] [view_frames]: Generating graph in frames.pdf file...
[INFO] [1699500746.65496796] [view_frames]: Result: tf2_msgs.srv.FrameGraphResponse(frame_yaml:'odom: \n parent: 'map'\n broadcaster: 'def
ault_authority'\n rate: 20.95\n most_recent_transform: 1699500746.651993\n oldest_transform: 1699500741.601316\n buffer_length: 5.051\nl
idar: \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\nbase_link: \n parent: 'laser'\n broadcaster: 'default_authority'\n rate: 10000.000\n most_recent_transf
orm: 0.000000\n oldest_transform: 0.000000\n buffer_length: 0.000\nlaser: \n parent: 'odom'\n broadcaster: 'default_authority'\n rate: 2
0.200\n most_recent_transform: 1699500746.604301\n oldest_transform: 1699500741.604318\n buffer_length: 5.000\n")
yahboom@VM:~$
```

In the directory where the command terminal is started, the system will generate a frames.pdf file, which is the generated TF tree.



2.4 launch file analysis

File path (combined with virtual machine):

```
/home/yahboom/rplidar_ws/src/rplidar_ros-dev-ros2/launch/test_gmapping.launch.py
```

test_gmapping.launch.py

```
from launch import LaunchDescription
from launch_ros.actions import Node
import os
from launch.actions import IncludeLaunchDescription
from launch.conditions import
LaunchConfigurationEquals, LaunchConfigurationNotEquals
from launch.launch_description_sources import
PythonLaunchDescriptionSource, AnyLaunchDescriptionSource
from ament_index_python.packages import get_package_share_directory
from launch.actions import DeclareLaunchArgument
import os

def generate_launch_description():
    LIDAR_TYPE = os.getenv('LIDAR_TYPE')
```

```

    lidar_type_arg = DeclareLaunchArgument(name='lidar_type',
default_value=LIDAR_TYPE, description='The type of lidar')

    lidar_launch =
IncludeLaunchDescription(PythonLaunchDescriptionSource([os.path.join(get_package
_share_directory('rplidar_ros'), 'launch'), '/lidar.launch.py']))

    s2_gmapping_launch =
IncludeLaunchDescription(PythonLaunchDescriptionSource([os.path.join(get_package
_share_directory('slam_gmapping'),
'launch'), '/s2_gmapping_launch.py']), condition=LaunchConfigurationEquals('lidar_
type', 's2'))

    gmapping_launch =
IncludeLaunchDescription(PythonLaunchDescriptionSource([os.path.join(get_package
_share_directory('slam_gmapping'),
'launch'), '/gmapping_launch.py']), condition=LaunchConfigurationNotEquals('lidar_
type', 's2'))

    s2_slam_gmapping_launch =
IncludeLaunchDescription(PythonLaunchDescriptionSource([os.path.join(get_package
_share_directory('slam_gmapping'),
'launch'), '/s2_slam_gmapping_launch.py']), condition=LaunchConfigurationEquals('l
idar_type', 's2'))

    slam_gmapping_launch =
IncludeLaunchDescription(PythonLaunchDescriptionSource([os.path.join(get_package
_share_directory('slam_gmapping'),
'launch'), '/slam_gmapping_launch.py']), condition=LaunchConfigurationNotEquals('l
idar_type', 's2'))

    return LaunchDescription([
        lidar_type_arg,
        lidar_launch,
        s2_gmapping_launch,
        s2_slam_gmapping_launch,
        gmapping_launch,
        slam_gmapping_launch
    ])

```

[lidar_type_arg] Gets the value of [LIDAR_TYPE] set in the environment variable, and starts the corresponding lidar and corresponding mapping instructions based on this value.

[lidar_launch] launch lidar launch file, the file is located at

`/home/yahboom/rplidar_ws/src/rplidar_ros-dev-ros2/launch/lidar.launch.py`

[s2_gmapping_launch] Start S2/S2L lidar gmapping mapping. Here you need to remap the lidar topic into a filtered lidar topic. The file is located

`/home/yahboom/rplidar_ws/src/slam_gmapping/launch/s2_gmapping_launch.py`

[s2_slam_gmapping_launch] Release some necessary static TF transformations and startup rviz for S2/S2L mapping lidar. The file is located at

`/home/yahboom/rplidar_ws/src/slam_gmapping/launch/s2_slam_gmapping_launch.py`

[gmapping_launch] Other lidar gmapping mapping, the file is located in

`/home/yahboom/rplidar_ws/src/slam_gmapping/launch/gmapping_launch.py`

[slam_gmapping_launch] Release some necessary static TF transformations and launch rviz for other lidar mapping. The file is located at

`/home/yahboom/rplidar_ws/src/slam_gmapping/launch/slam_gmapping.launch.py`