

2.Handheld lidar mapping

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

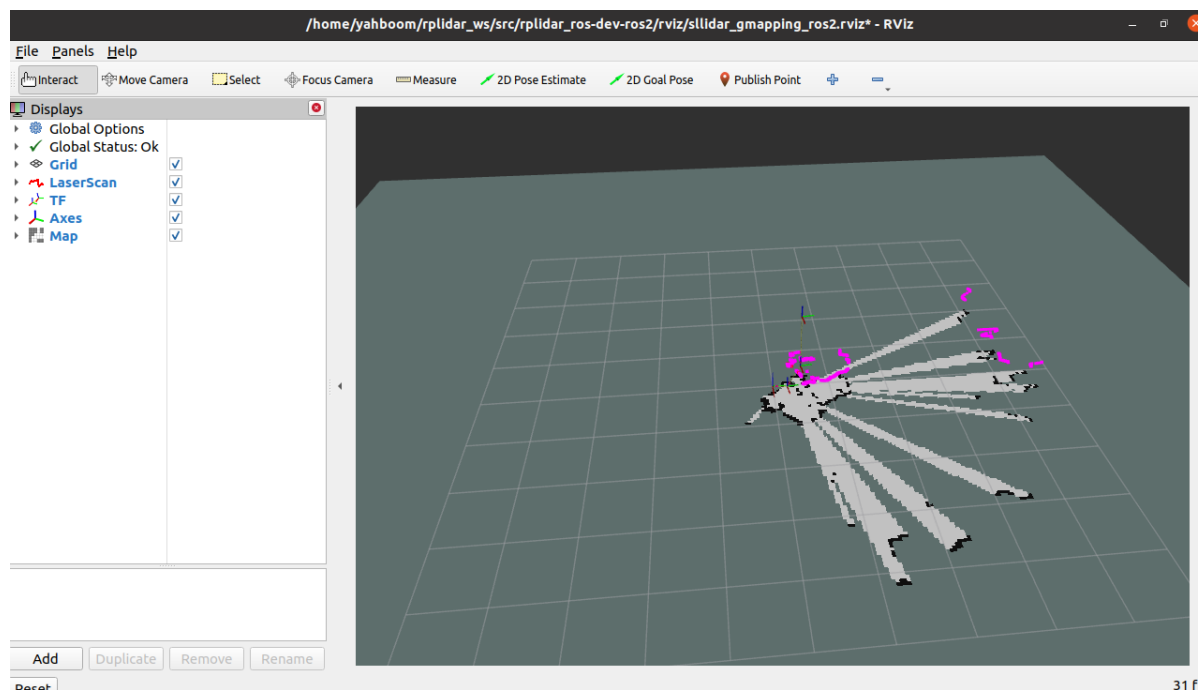
2.1 Start gmapping mapping

Input following command:

```
ros2 launch oradar_lidar ms200_scan.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
```

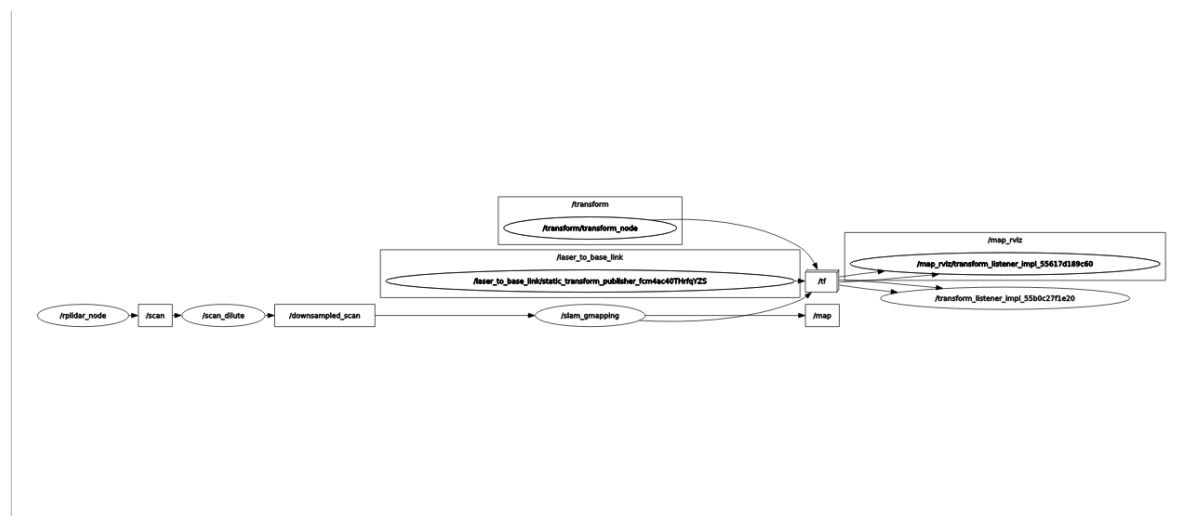
rviz displays as follows:



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@VMH:~$ ros2 run tf2_tools view_frames.py
[INFO] [1699500741.610795137]: [view_frames]: Listening to tf data during 5 seconds...
[INFO] [1699500746.652813123]: [view_frames]: Generating graph in frames.pdf file...
[INFO] [1699500746.652813123]: [view_frames]: Result of msgsrv FramGraph: Response(frame_yaml:'odom':\n parent: 'map'\n broadcaster: 'def\nult authority'\n rate: 20.000000\n oldest_transform: 1699500741.603136\n buffer_length: 5.051\nl\n idar:\n parent: 'base_link'\n broadcaster: 'default authority'\n rate: 10000.0000\n most_recent_transform: 0.000000\n oldest_transf\norm: 0.000000\n buffer_length: 0.000\nbase_link:\n parent: 'laser'\n broadcaster: 'default authority'\n rate: 10000.0000\n most_recent_transf\norm: 0.000000\n oldest_transform: 0.000000\n buffer_length: 0.000\nlaser:\n parent: 'odom'\n broadcaster: 'default authority'\n rate: 2\n0.200\n most_recent_transform: 1699500746.604301\n oldest_transform: 1699500741.604318\n buffer_length: 5.000\n")
yahboom@VMH:~$
```

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

view_frames Result

Recorded at time: 1699613833.8771782

