RTAB-Map Mapping

1. Contents

This section explains how to combine a vehicle chassis, LiDAR, and a depth camera to implement RTAB-Map mapping.

1.1 Introduction to RTAB-Map

RTAB-MAP (Real-Time Appearance-Based Mapping) is a vision-based real-time simultaneous localization and mapping (SLAM) algorithm primarily used for mapping and navigation tasks in fields such as robotics and augmented reality. RTAB-Map has the following features.

- **Appearance-based loop closure detection**: Primarily uses visual features for location recognition
- **Real-time performance**: Ensures real-time performance through memory management
- **Multi-sensor support**: Supports RGB-D cameras, stereo cameras, and monocular cameras
- **3D map construction**: Creates dense 3D point cloud maps

1.2. RTAB-Map Working Principle

- Front-end processing
  - Acquires image data from sensors
  - Extracts visual features (such as SIFT, SURF, ORB, etc.)
  - Calculates the pose transformation between the current frame and the previous frame
- Back-end optimization
  - Optimizes the pose graph using graph optimization techniques (such as g2o)
  - Detects loop closures (when the robot returns to a previously visited location)
- Memory management
  - Employs a working memory (WM) and long-term memory (LTM) mechanism
  - Recent data is stored in the WM, while older data is transferred to the LTM
  - When a loop closure is detected, the relevant data is transferred from the LTM back to the WM

This section requires entering commands in a terminal. The terminal you choose to open depends on your board type. This lesson uses the Raspberry Pi as an example. For Raspberry Pi and Jetson-Nano boards, you need to open a terminal on the host computer and enter the command to enter the Docker container. Once inside the Docker container, enter the commands mentioned in this lesson in the terminal. For instructions on entering the Docker container from the host computer, refer to **[01. Robot Configuration and Operation Guide] -- [4.Enter Docker (For JETSON Nano and RPi 5)]**.

For RDKX5 and Orin boards, simply open a terminal and enter the commands mentioned in this lesson.

2. Preparation

Due to performance limitations, the Raspberry Pi 5 and Jetson Nano cannot smoothly run the RTAB-Map algorithm in Docker on the board. Therefore, a virtual machine is required to facilitate this. To enable distributed communication between the car and the virtual machine, two steps are required:

- Both must be on the same local area network. This is most easily achieved by connecting to the same Wi-Fi network.
- Both devices must have the same ROS_DOMAIN_ID. The default ROS_DOMAIN_ID for the car is 30, and the default ROS_DOMAIN_ID for the virtual machine is also 30. If they are different, modify the virtual machine's ROS_DOMAIN_ID. To do this, modify the ~/.bashrc file and change the ROS_DOMAIN_ID value to match the car's. Save and exit the file, then enter the command source ~/.bashrc to refresh the environment variables.
- To verify distributed communication between the two, run the driver program on the board, then enter ros2 node list on the virtual machine. If the underlying data topic appears, the two devices are communicating.

The Orin board can be run directly on the board.

3. Program Startup

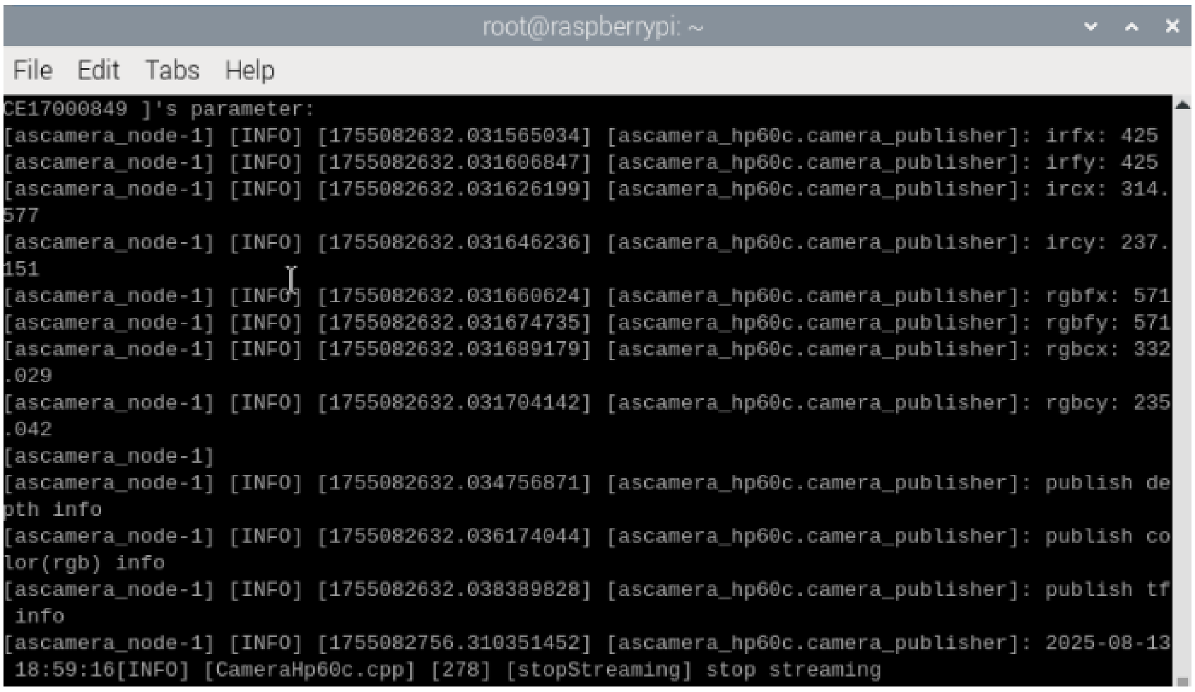
For Raspberry Pi and Jetson-Nano boards, you need to enter the Docker container first. For RDKX5 and Orin controllers, this is not necessary.

Enter the Docker container (see [Docker course] --- [4. Docker Startup Script] for steps).

All the following commands must be executed within the same Docker container **(see [Docker course] --- [3. Docker Submission and Multi-Terminal Access] for steps).**

First, enter the following command in the car terminal to start the depth camera.

```
ros2 launch ascamera hp60c.launch.py
```



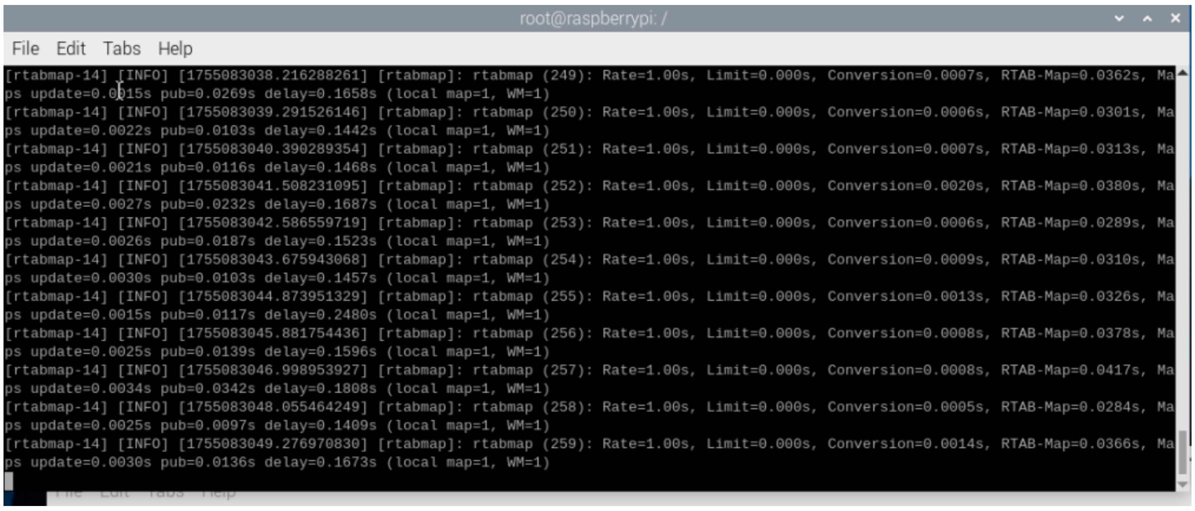
root@raspberrypi: ~

File Edit Tabs Help

```
CE17000849 ]'s parameter:
[ascamera_node-1] [INFO] [1755082632.031565034] [ascamera_hp60c.camera_publisher]: irfx: 425
[ascamera_node-1] [INFO] [1755082632.031606847] [ascamera_hp60c.camera_publisher]: irfy: 425
[ascamera_node-1] [INFO] [1755082632.031626199] [ascamera_hp60c.camera_publisher]: ircx: 314.
577
[ascamera_node-1] [INFO] [1755082632.031646236] [ascamera_hp60c.camera_publisher]: ircy: 237.
151
[ascamera_node-1] [INFO] [1755082632.031660624] [ascamera_hp60c.camera_publisher]: rgbfx: 571
[ascamera_node-1] [INFO] [1755082632.031674735] [ascamera_hp60c.camera_publisher]: rgbfy: 571
[ascamera_node-1] [INFO] [1755082632.031689179] [ascamera_hp60c.camera_publisher]: rgbcx: 332
.029
[ascamera_node-1] [INFO] [1755082632.031704142] [ascamera_hp60c.camera_publisher]: rgbcy: 235
.042
[ascamera_node-1]
[ascamera_node-1] [INFO] [1755082632.034756871] [ascamera_hp60c.camera_publisher]: publish de
pth info
[ascamera_node-1] [INFO] [1755082632.036174044] [ascamera_hp60c.camera_publisher]: publish co
lor(rgb) info
[ascamera_node-1] [INFO] [1755082632.038389828] [ascamera_hp60c.camera_publisher]: publish tf
 info
[ascamera_node-1] [INFO] [1755082756.310351452] [ascamera_hp60c.camera_publisher]: 2025-08-13
 18:59:16[INFO] [CameraHp60c.cpp] [278] [stopStreaming] stop streaming
```

Enter the following command in the car terminal to start mapping, chassis, and radar.

```
ros2 launch yahboomcar_nav map_rtabmap_launch.py
```

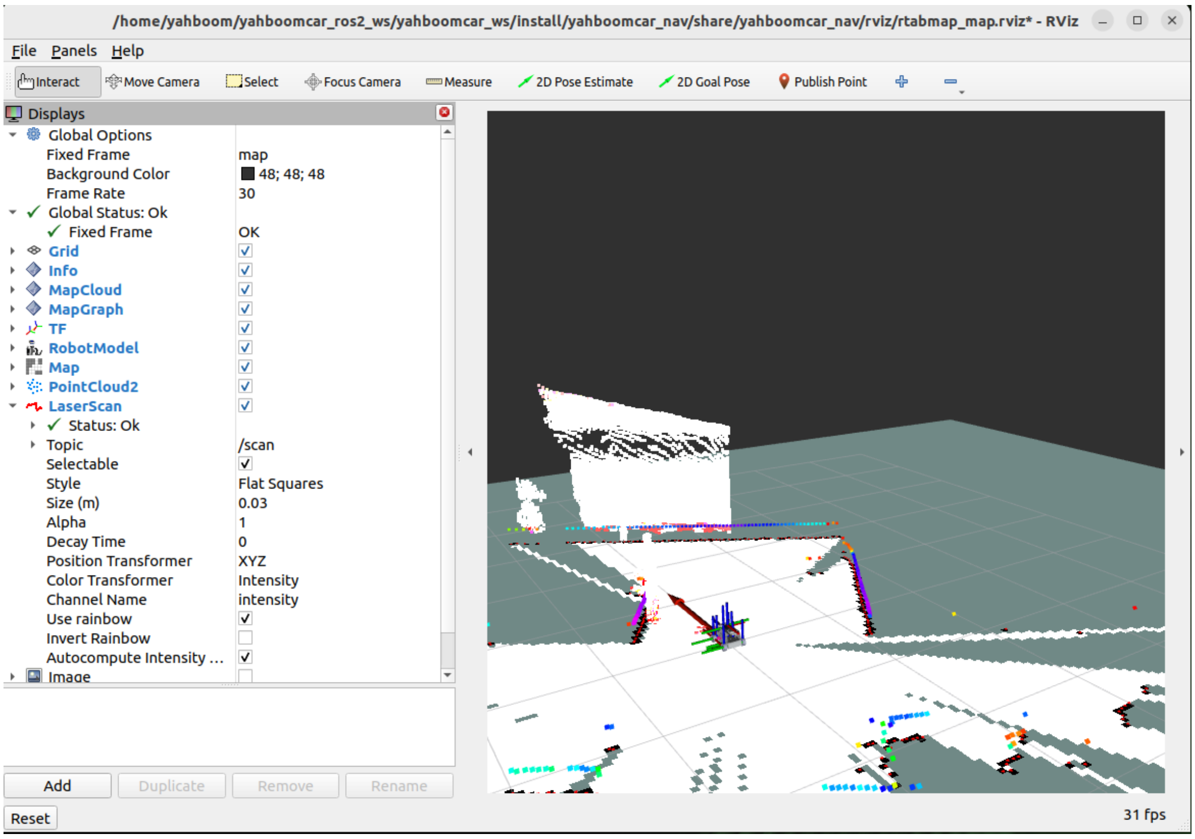


```
[rtabmap-14] [INFO] [1755083038.216288261] [rtabmap]: rtabmap (249): Rate=1.00s, Limit=0.000s, Conversion=0.0007s, RTAB-Map=0.0362s, Ma
ps update=0.0015s pub=0.0269s delay=0.1658s (local map=1, WM=1)
[rtabmap-14] [INFO] [1755083039.291526146] [rtabmap]: rtabmap (250): Rate=1.00s, Limit=0.000s, Conversion=0.0006s, RTAB-Map=0.0301s, Ma
ps update=0.0022s pub=0.0103s delay=0.1442s (local map=1, WM=1)
[rtabmap-14] [INFO] [1755083040.390289354] [rtabmap]: rtabmap (251): Rate=1.00s, Limit=0.000s, Conversion=0.0007s, RTAB-Map=0.0313s, Ma
ps update=0.0021s pub=0.0116s delay=0.1468s (local map=1, WM=1)
[rtabmap-14] [INFO] [1755083041.508231095] [rtabmap]: rtabmap (252): Rate=1.00s, Limit=0.000s, Conversion=0.0020s, RTAB-Map=0.0380s, Ma
ps update=0.0027s pub=0.0232s delay=0.1687s (local map=1, WM=1)
[rtabmap-14] [INFO] [1755083042.586559719] [rtabmap]: rtabmap (253): Rate=1.00s, Limit=0.000s, Conversion=0.0006s, RTAB-Map=0.0289s, Ma
ps update=0.0026s pub=0.0187s delay=0.1523s (local map=1, WM=1)
[rtabmap-14] [INFO] [1755083043.675943068] [rtabmap]: rtabmap (254): Rate=1.00s, Limit=0.000s, Conversion=0.0009s, RTAB-Map=0.0310s, Ma
ps update=0.0030s pub=0.0103s delay=0.1457s (local map=1, WM=1)
[rtabmap-14] [INFO] [1755083044.873951329] [rtabmap]: rtabmap (255): Rate=1.00s, Limit=0.000s, Conversion=0.0013s, RTAB-Map=0.0326s, Ma
ps update=0.0015s pub=0.0117s delay=0.2480s (local map=1, WM=1)
[rtabmap-14] [INFO] [1755083045.881754436] [rtabmap]: rtabmap (256): Rate=1.00s, Limit=0.000s, Conversion=0.0008s, RTAB-Map=0.0378s, Ma
ps update=0.0025s pub=0.0139s delay=0.1596s (local map=1, WM=1)
[rtabmap-14] [INFO] [1755083046.998953927] [rtabmap]: rtabmap (257): Rate=1.00s, Limit=0.000s, Conversion=0.0008s, RTAB-Map=0.0417s, Ma
ps update=0.0034s pub=0.0342s delay=0.1808s (local map=1, WM=1)
[rtabmap-14] [INFO] [1755083048.055464249] [rtabmap]: rtabmap (258): Rate=1.00s, Limit=0.000s, Conversion=0.0005s, RTAB-Map=0.0284s, Ma
ps update=0.0025s pub=0.0097s delay=0.1409s (local map=1, WM=1)
[rtabmap-14] [INFO] [1755083049.276970830] [rtabmap]: rtabmap (259): Rate=1.00s, Limit=0.000s, Conversion=0.0014s, RTAB-Map=0.0366s, Ma
ps update=0.0030s pub=0.0136s delay=0.1673s (local map=1, WM=1)
```

Enter the following command on the virtual machine to perform rviz visualization.

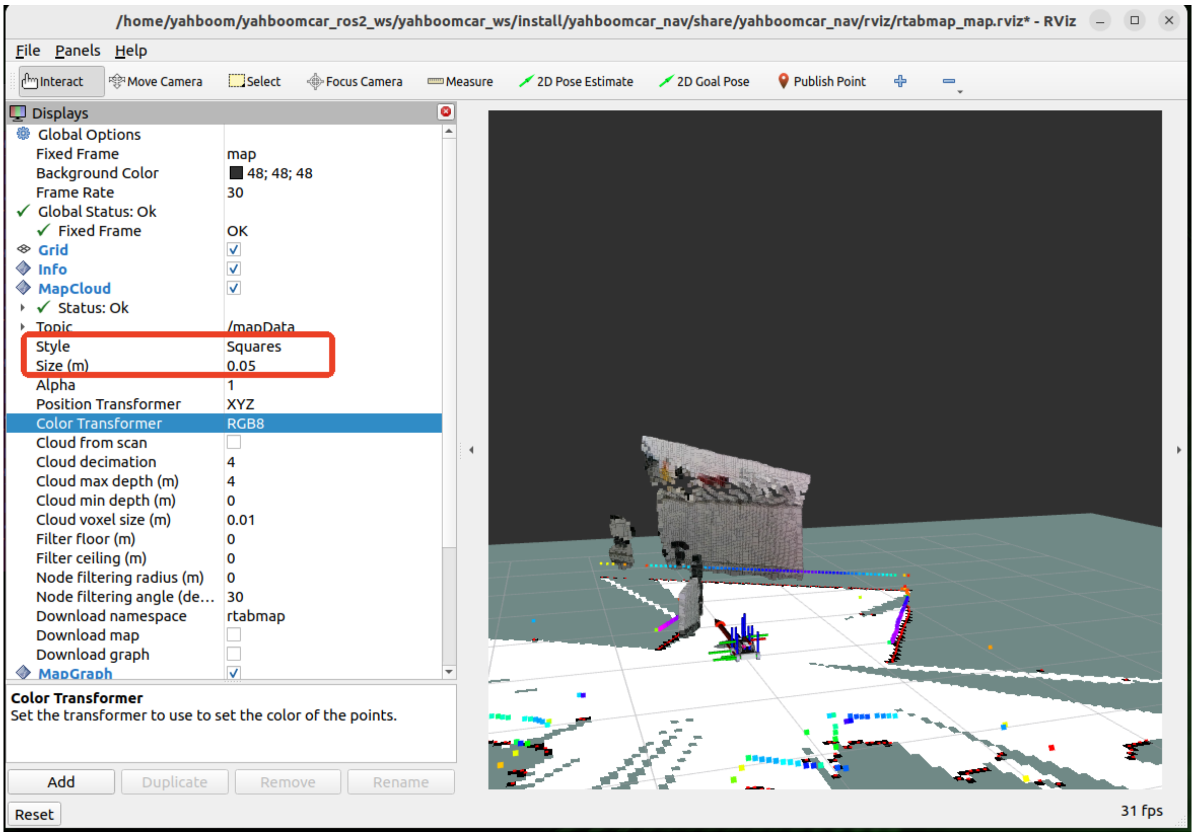
```
ros2 launch yahboomcar_nav display_rtabmap_map_launch.py
```

After successful startup, the image below will appear.



```
Global Options
    Fixed Frame                 map
    Background Color            48; 48; 48
    Frame Rate                  30
✓ Global Status: Ok
    ✓ Fixed Frame               OK
Grid                            ✓
Info                            ✓
MapCloud                        ✓
MapGraph                        ✓
TF                              ✓
RobotModel                      ✓
Map                             ✓
PointCloud2                     ✓
LaserScan                       ✓
    ✓ Status: Ok
    Topic                       /scan
    Selectable                  ✓
    Style                       Flat Squares
    Size (m)                    0.03
    Alpha                       1
    Decay Time                  0
    Position Transformer        XYZ
    Color Transformer           Intensity
    Channel Name                intensity
    Use rainbow                 ✓
    Invert Rainbow              
    Autocompute Intensity ...   ✓
Image                           
```

Modify the settings in rviz to change the point cloud display to RGB, as shown below.



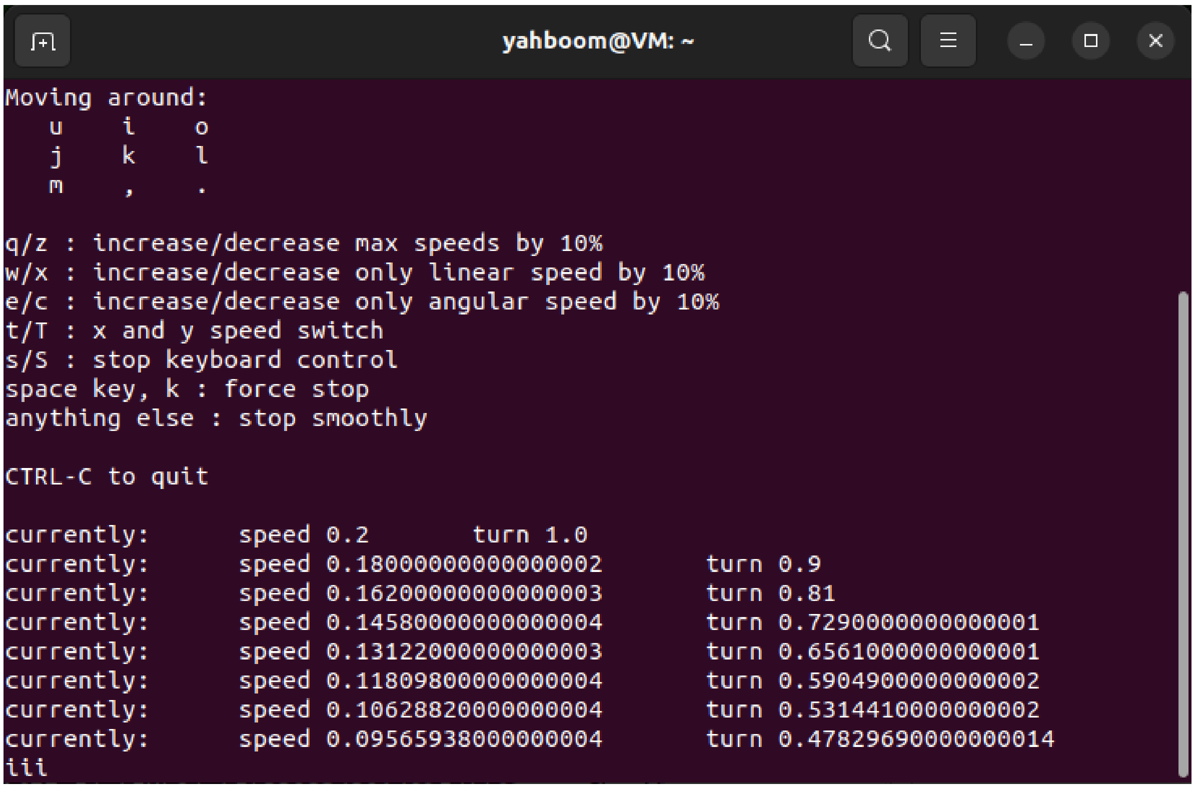
/home/yahboom/yahboomcar_ros2_ws/yahboomcar_ws/install/yahboomcar_nav/share/yahboomcar_nav/rviz/rtabmap_map.rviz* - RViz

```
Displays
Global Options
  Fixed Frame                  map
  Background Color             48; 48; 48
  Frame Rate                   30
Global Status: Ok
  Fixed Frame                  OK
Grid                           ✓
Info                           ✓
MapCloud                       ✓
  Status: Ok
  Topic                        /mapData
  Style                        Squares
  Size (m)                     0.05
  Alpha                        1
  Position Transformer         XYZ
  Color Transformer            RGB8
  Cloud from scan
  Cloud decimation             4
  Cloud max depth (m)          4
  Cloud min depth (m)          0
  Cloud voxel size (m)         0.01
  Filter floor (m)             0
  Filter ceiling (m)           0
  Node filtering radius (m)    0
  Node filtering angle (de...  30
  Download namespace           rtabmap
  Download map
  Download graph
MapGraph                       ✓
```

Color Transformer
Set the transformer to use to set the color of the points.

Finally, open a terminal in the virtual machine and enter the following command to control the car's movement and map creation using the keyboard.

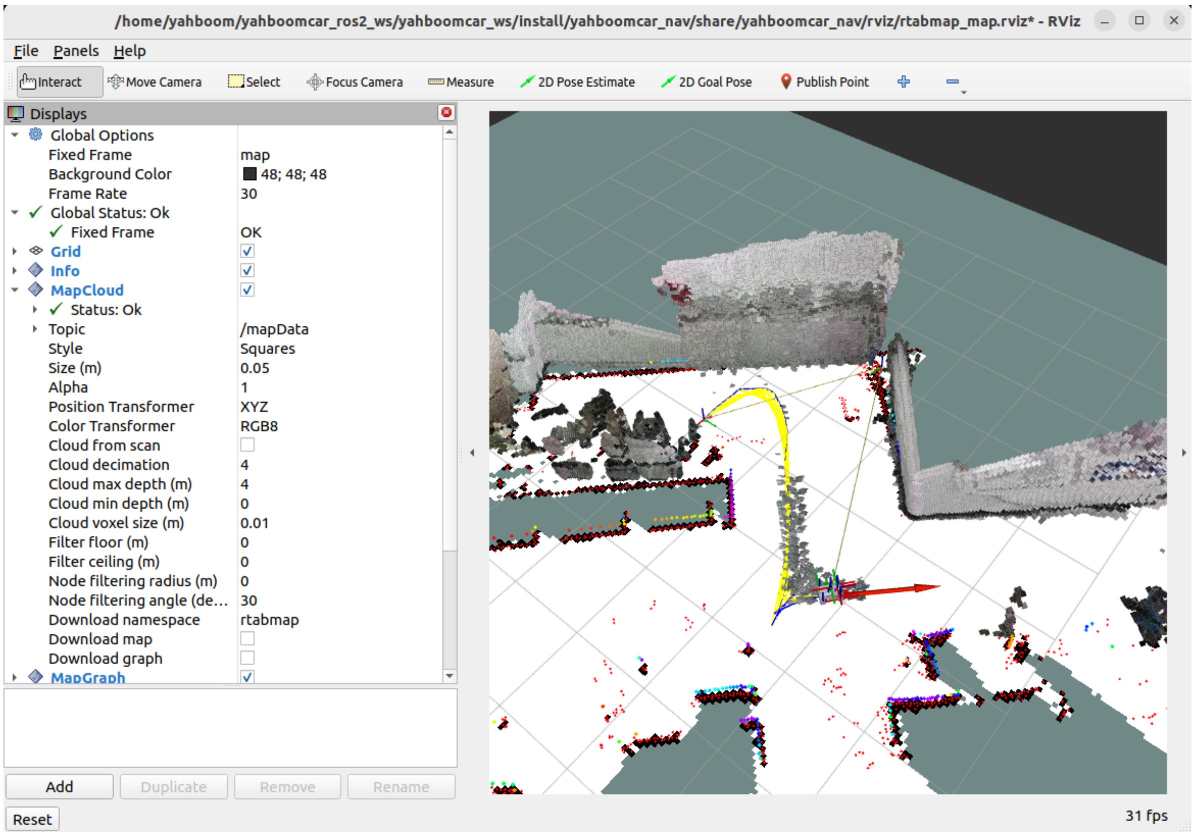
```
ros2 run yahboomcar_ctrl yahboom_keyboard
```



```
Moving around:
   u    i    o
   j    k    l
   m    ,    .

q/z : increase/decrease max speeds by 10%
w/x : increase/decrease only linear speed by 10%
e/c : increase/decrease only angular speed by 10%
t/T : x and y speed switch
s/S : stop keyboard control
space key, k : force stop
anything else : stop smoothly

CTRL-C to quit

currently:      speed 0.2       turn 1.0
currently:      speed 0.18000000000000002       turn 0.9
currently:      speed 0.16200000000000003       turn 0.81
currently:      speed 0.14580000000000004       turn 0.7290000000000001
currently:      speed 0.13122000000000003       turn 0.6561000000000001
currently:      speed 0.11809800000000004       turn 0.5904900000000002
currently:      speed 0.10628820000000004       turn 0.5314410000000002
currently:      speed 0.09565938000000004       turn 0.47829690000000014
iii
```

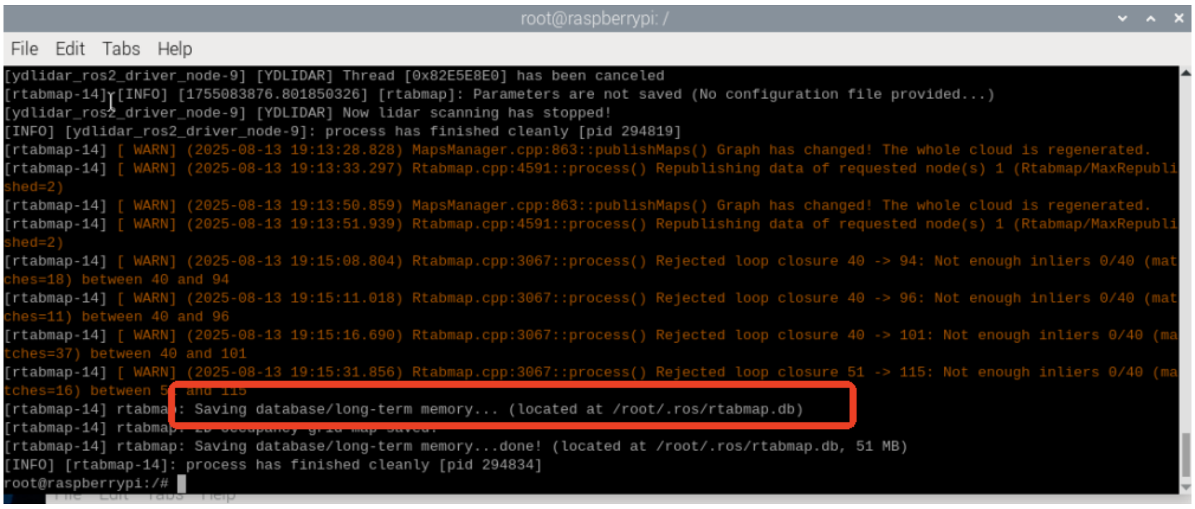
Here, control the car's movement as slowly as possible. Press the [z] key to reduce the linear velocity and deceleration. Then, press the [i] key to move the car forward, the [,] key to move it backward, the [j] key to rotate the car left, and the [l] key to rotate the car right.

As shown below, the map is created.

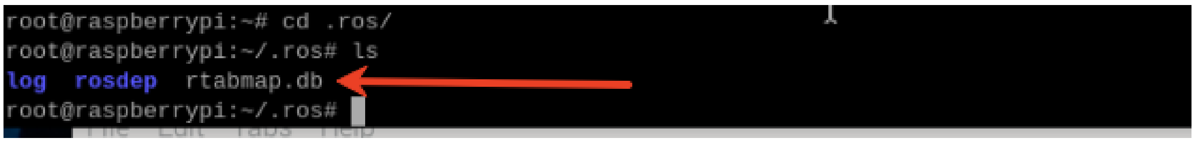


```
Global Options
  Fixed Frame                    map
  Background Color               48; 48; 48
  Frame Rate                     30
✓ Global Status: Ok
  ✓ Fixed Frame                  OK
Grid                             ✓
Info                             ✓
MapCloud                         ✓
  ✓ Status: Ok
  Topic                          /mapData
  Style                          Squares
  Size (m)                       0.05
  Alpha                          1
  Position Transformer           XYZ
  Color Transformer              RGB8
  Cloud from scan                ☐
  Cloud decimation               4
  Cloud max depth (m)            4
  Cloud min depth (m)            0
  Cloud voxel size (m)           0.01
  Filter floor (m)               0
  Filter ceiling (m)             0
  Node filtering radius (m)      0
  Node filtering angle (de...    30
  Download namespace             rtabmap
  Download map                   ☐
  Download graph                 ☐
MapGraph                         ✓
```

After the map is created, press Ctrl+C in the terminal where you launched map_rtabmap_launch.py to close the program. The map will be saved in /root/.ros/rtabmap.db.



```
[ydlidar_ros2_driver_node-9] [YDLIDAR] Thread [0x82E5E8E0] has been canceled
[rtabmap-14] [INFO] [1755083876.801850326] [rtabmap]: Parameters are not saved (No configuration file provided...)
[ydlidar_ros2_driver_node-9] [YDLIDAR] Now lidar scanning has stopped!
[INFO] [ydlidar_ros2_driver_node-9]: process has finished cleanly [pid 294819]
[rtabmap-14] [ WARN] (2025-08-13 19:13:28.828) MapsManager.cpp:863::publishMaps() Graph has changed! The whole cloud is regenerated.
[rtabmap-14] [ WARN] (2025-08-13 19:13:33.297) Rtabmap.cpp:4591::process() Republishing data of requested node(s) 1 (Rtabmap/MaxRepubli
shed=2)
[rtabmap-14] [ WARN] (2025-08-13 19:13:50.859) MapsManager.cpp:863::publishMaps() Graph has changed! The whole cloud is regenerated.
[rtabmap-14] [ WARN] (2025-08-13 19:13:51.939) Rtabmap.cpp:4591::process() Republishing data of requested node(s) 1 (Rtabmap/MaxRepubli
shed=2)
[rtabmap-14] [ WARN] (2025-08-13 19:15:08.804) Rtabmap.cpp:3067::process() Rejected loop closure 40 -> 94: Not enough inliers 0/40 (mat
ches=18) between 40 and 94
[rtabmap-14] [ WARN] (2025-08-13 19:15:11.018) Rtabmap.cpp:3067::process() Rejected loop closure 40 -> 96: Not enough inliers 0/40 (mat
ches=11) between 40 and 96
[rtabmap-14] [ WARN] (2025-08-13 19:15:16.690) Rtabmap.cpp:3067::process() Rejected loop closure 40 -> 101: Not enough inliers 0/40 (ma
tches=37) between 40 and 101
[rtabmap-14] [ WARN] (2025-08-13 19:15:31.856) Rtabmap.cpp:3067::process() Rejected loop closure 51 -> 115: Not enough inliers 0/40 (ma
tches=16) between 51 and 115
[rtabmap-14] rtabmap: Saving database/long-term memory... (located at /root/.ros/rtabmap.db)
[rtabmap-14] rtabmap: 2D occupancy grid map saved.
[rtabmap-14] rtabmap: Saving database/long-term memory...done! (located at /root/.ros/rtabmap.db, 51 MB)
[INFO] [rtabmap-14]: process has finished cleanly [pid 294834]
root@raspberrypi:/#
```



```
root@raspberrypi:~# cd .ros/
root@raspberrypi:~/.ros# ls
log  rosdep  rtabmap.db
root@raspberrypi:~/.ros#
```