

UTF8gbsn

HCX Robot Documentation

发行版本 *lastest*

tianbot

2024 年 11 月 28 日

1 使用说明	1
2 节点说明	21
3 整机配置	27

CHAPTER 1

使用说明

1.1 如何连接 HCX 机器人

Contents

- 准备工作
- 网络连接
 - *Rustdesk* 连接
 - 机器人上的 IP 地址
- ROS2 多机通信

1.1.1 准备工作

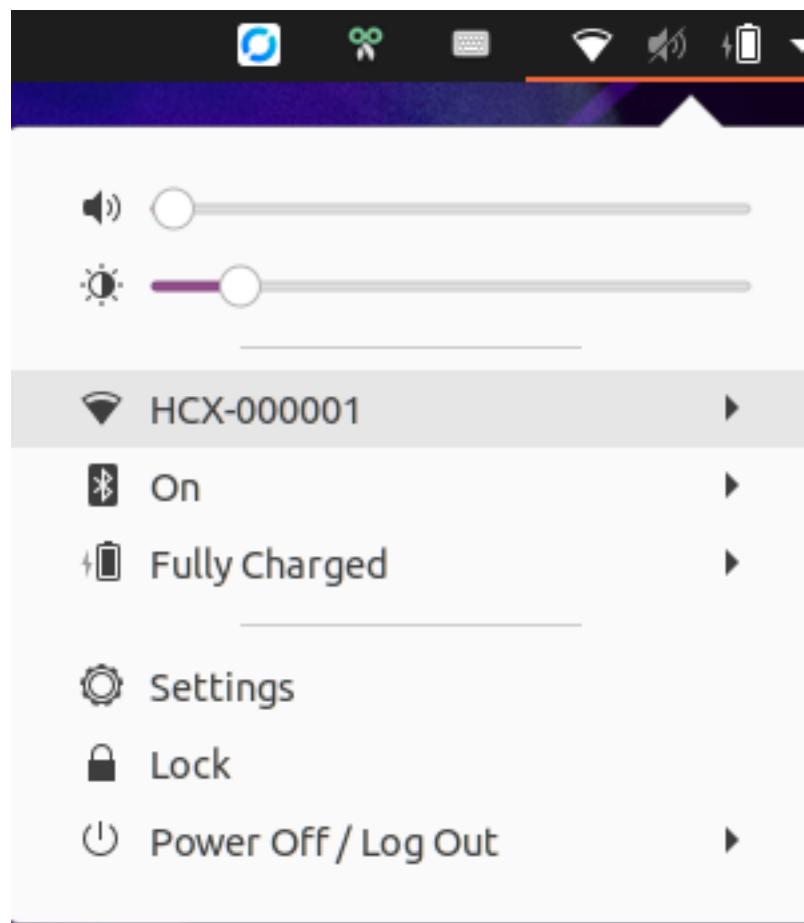
小技巧: 请检查机器人外观，确保没有损坏或丢失任何部件。

- 给机器人充电，确保电池电量充足。
- 给机器人上电，等待机器人启动完成。

1.1.2 网络连接

小技巧: 请确保电脑连接到了机器人自建的 HCX-00000x 热点下。

备注: 热点密码为 12345678。

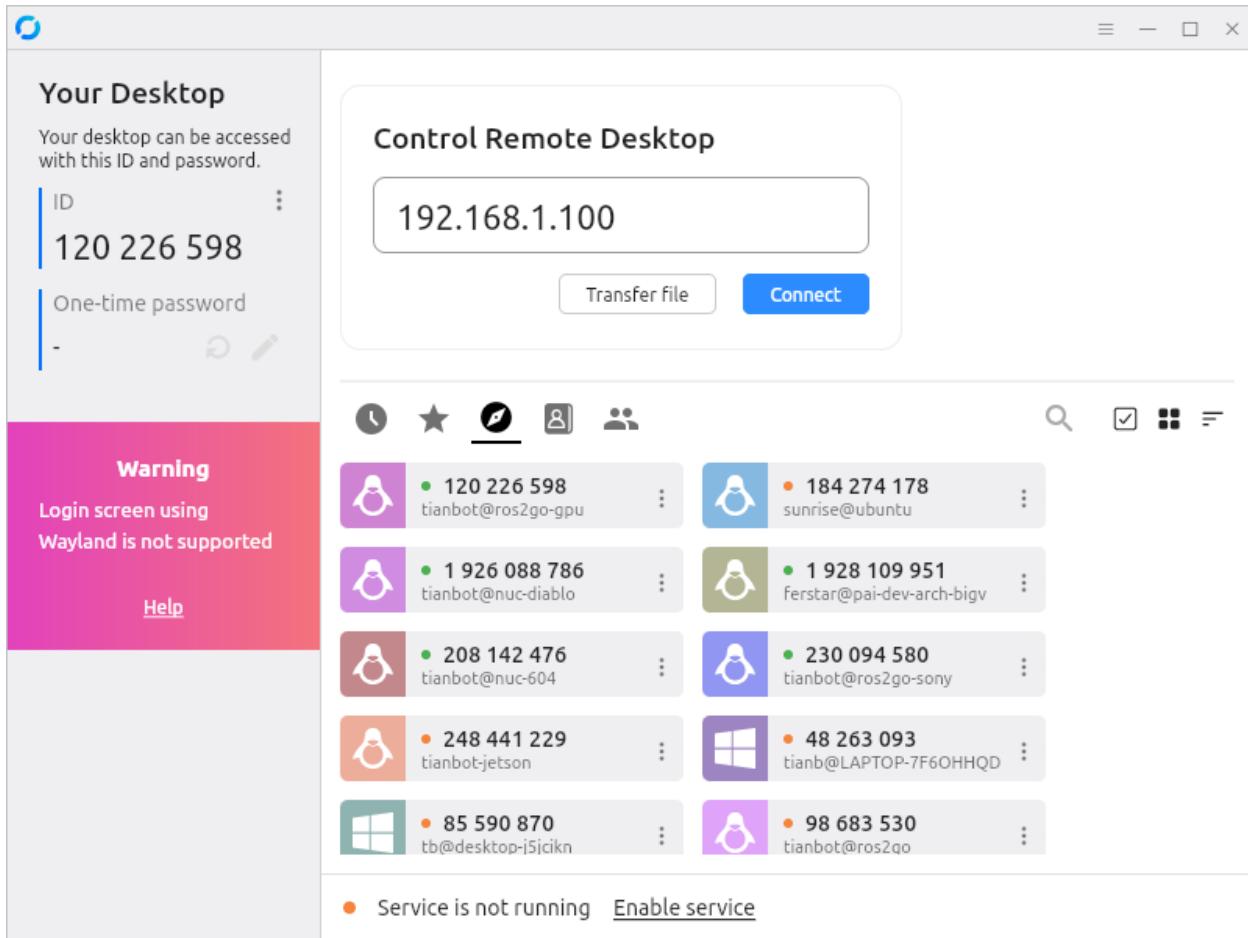


```
### ssh 连接  
然后使用以下命令 ssh 连接到机器人：  
```bash  
ssh tianbot@192.168.1.100 # 密码为 ros
```

## Rustdesk 连接

- 详细流程请参考使用 Rustdesk 远程连接

在窗口中输入 192.168.1.100，然后点击 连接。



---

备注: rustdesk 登录密码 Tianbot\_2016

---



### 机器人上的 IP 地址

机器人上设备的 IP 地址如下表所示：

机器人 IP 地址	对应设备
192.168.1.2	4G 路由器地址, 用于连接机器人
192.168.1.100	机器人 (eth0) IP 地址
192.168.1.1xx	(sn 码的后两位)
192.168.1.201	FPV 网络摄像头 IP 地址

---

**小技巧:** 请确保机器人与电脑在同一网络下，并且能够通过 IP 地址访问。

即电脑连接到机器人自建的 HCX-00000x 热点下，或者连接到机器人所在的局域网。

- 机器人默认用户名: tianbot, 默认密码: ros
- 

### 1.1.3 ROS2 多机通信

---

**小技巧:** 将以下命令添加到 ~/.bashrc 中，以确保您可以通过 LAN 访问 ROS Robots 节点

---

```
export ROS_DOMAIN_ID=5
```

采用如下操作检查 ROS2 DOMAIN 通信是否正常

```
ros2 topic list
```

---

**小技巧:** 请确保机器人已经连接到网络，并且能够通过 IP 地址访问。

---

## 1.2 启动整机驱动和可视化传感器数据

### Contents

- 启动整机 *ROS* 驱动
- 如何关闭驱动
  - 如何判断程序是否结束
- 在 *rviz* 中查看传感器数据
  - 1. 启动 *rviz*
  - 2. 查看机器人状态

### 1.2.1 启动整机 *ROS* 驱动

- 启动底盘驱动
- 启动传感器驱动
- 启动外设驱动

```
ros2 launch diablo_bringup diablo_bringup.launch.py
```

**备注:** 如何在启动整机驱动的终端中出现如下错误: 请结束驱动后, 重新启动整机驱动。

- [ekf\_node-14] [ERROR] [1732523155.702111265] [ekf\_filter\_node]: Critical Error, NaNs were detected in the output state of the filter. This was likely due to poorly conditioned process, noise, or sensor covariances.
- [pointcloud\_to\_laserscan\_node-15] Error: TF\_NAN\_INPUT: Ignoring transform for child\_frame\_id "base\_footprint" from authority "Authority undetectable" because of a nan value in the transform (-nan -nan -nan) (-nan -nan -nan)
- [pointcloud\_to\_laserscan\_node-15] at line 298 in /tmp/binarydeb/ros-galactic-tf2-0.17.5/src/buffer\_core.cpp
- [pointcloud\_to\_laserscan\_node-15] Error: TF\_DENORMALIZED\_QUATERNION: Ignoring transform for child\_frame\_id "base\_footprint" from authority "Authority undetectable" because of an invalid quaternion in the transform (-nan -nan -nan -nan)
- [pointcloud\_to\_laserscan\_node-15] at line 317 in /tmp/binarydeb/ros-galactic-tf2-0.17.5/src/buffer\_core.cpp
- [ekf\_node-14] Error: TF\_NAN\_INPUT: Ignoring transform for child\_frame\_id "base\_footprint" from authority "Authority undetectable" because of a nan value in the transform (-nan -nan -nan) (-nan -nan -nan)
- [ekf\_node-14] at line 298 in /tmp/binarydeb/ros-galactic-tf2-0.17.5/src/buffer\_core.cpp
- [ekf\_node-14] Error: TF\_DENORMALIZED\_QUATERNION: Ignoring transform for child\_frame\_id "base\_footprint" from authority "Authority undetectable" because of an invalid quaternion in the transform (-nan -nan -nan -nan)
- [ekf\_node-14] at line 317 in /tmp/binarydeb/ros-galactic-tf2-0.17.5/src/buffer\_core.cpp
- [ekf\_node-14] [ERROR] [1732523155.752382101] [ekf\_filter\_node]: Critical Error, NaNs were detected in the output state of the filter. This was likely due to poorly conditioned process, noise, or sensor covariances.
- [ekf\_node-14] Error: TF\_NAN\_INPUT: Ignoring transform for child\_frame\_id "base\_footprint" from authority "Authority undetectable" because of a nan value in the transform (-nan -nan -nan) (-nan -nan -nan -nan)

ROS 驱动成功启动后，LED 氛围灯会常亮，表示驱动启动成功。



## 1.2.2 如何关闭驱动

---

**备注:** 请注意关闭建图、导航等程序也是，后续不再重述

1. 点击运行 `ros2 xxx xxx xxx` 的终端后
  2. 使用 `Ctrl+C` 操作
- 

**小技巧:** 单击运行 `ros2 launch diablo_bringup diablo_bringup.launch.py` 启动整机驱动的终端后，

---

需要按下键盘上的 `Ctrl+C` 按键，即可向程序发送程序结束命令。然后等待程序结束即可。

---

## 如何判断程序是否结束

**备注:** 如在 `ros2 launch diablo_bringup diablo_bringup.launch.py` 启动整机驱动的终端中, 出现以下提示, 则表示程序已经结束。

- ^C[WARNING] [launch]: user interrupted with ctrl-c (SIGINT)
- [ekf\_node-14] [INFO] [1732508443.614860079] [rclcpp]: signal\_handler(signal\_value=2)
- [static\_transform\_publisher-11] [INFO] [1732508443.614760239] [rclcpp]: signal\_handler(signal\_value=2)
- [diablo\_ctrl\_node-6] [INFO] [1732508443.614761786] [rclcpp]: signal\_handler(signal\_value=2)
- [static\_transform\_publisher-9] [INFO] [1732508443.614760294] [rclcpp]: signal\_handler(signal\_value=2)
- [static\_transform\_publisher-10] [INFO] [1732508443.614827033] [rclcpp]: signal\_handler(signal\_value=2)
- [robot\_state\_publisher-13] [INFO] [1732508443.614835078] [rclcpp]: signal\_handler(signal\_value=2)
- [static\_transform\_publisher-8] [INFO] [1732508443.614840605] [rclcpp]: signal\_handler(signal\_value=2)
- [livox\_ros\_driver2\_node-1] [INFO] [1732508443.614852701] [rclcpp]: signal\_handler(signal\_value=2)
- [pointcloud\_to\_laserscan\_node-15] [INFO] [1732508443.614861184] [rclcpp]: signal\_handler(signal\_value=2)
- [odom\_publish\_node-17] [INFO] [1732508443.614853049] [rclcpp]: signal\_handler(signal\_value=2)
- [msg\_convert\_node-16] [INFO] [1732508443.614858505] [rclcpp]: signal\_handler(signal\_value=2)
- [realsense2\_camera\_node-2] [INFO] [1732508443.614794739] [rclcpp]: signal\_handler(signal\_value=2)
- .....

## 1.2.3 在 rviz 中查看传感器数据

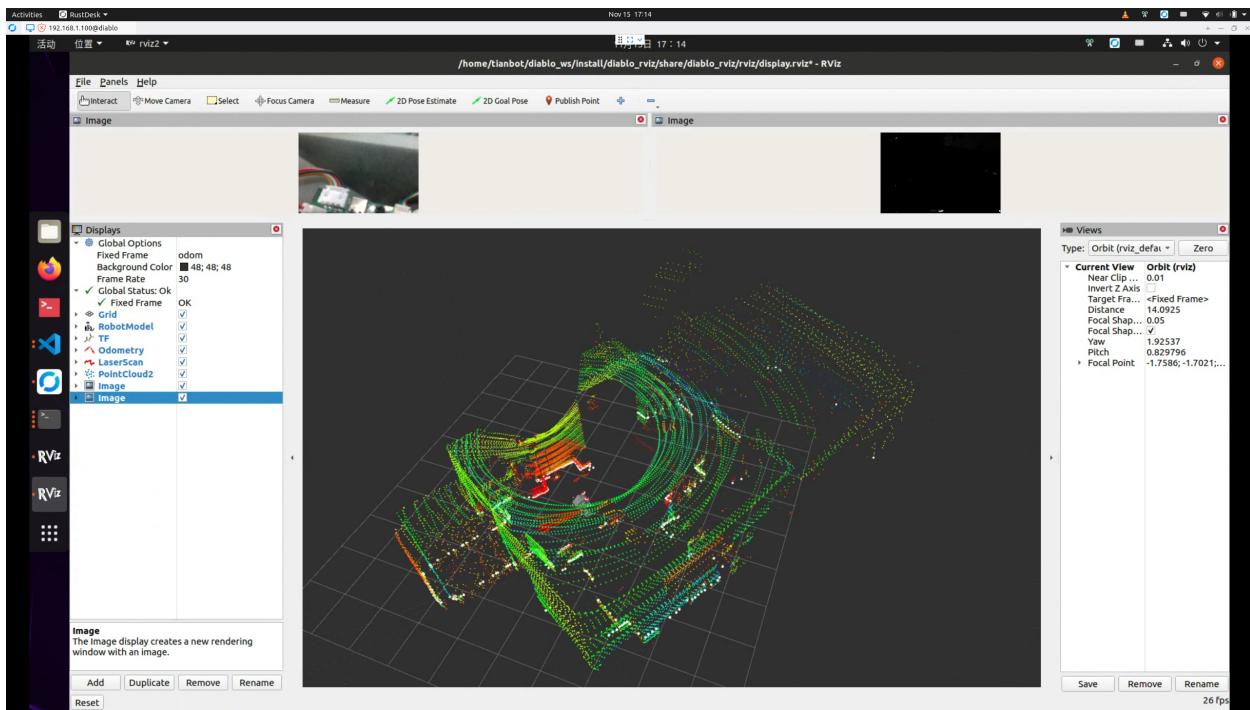
### 1. 启动 rviz

```
ros2 launch diablo_rviz view_robot.launch.py
```

**备注:** 这一步需要先正确上电启动机器人, 否则 rviz 无法获取传感器数据

### 2. 查看机器人状态

点击左侧导航栏的“可视化”按钮, 进入可视化界面, 可以看到当前机器人所连接的传感器数据。

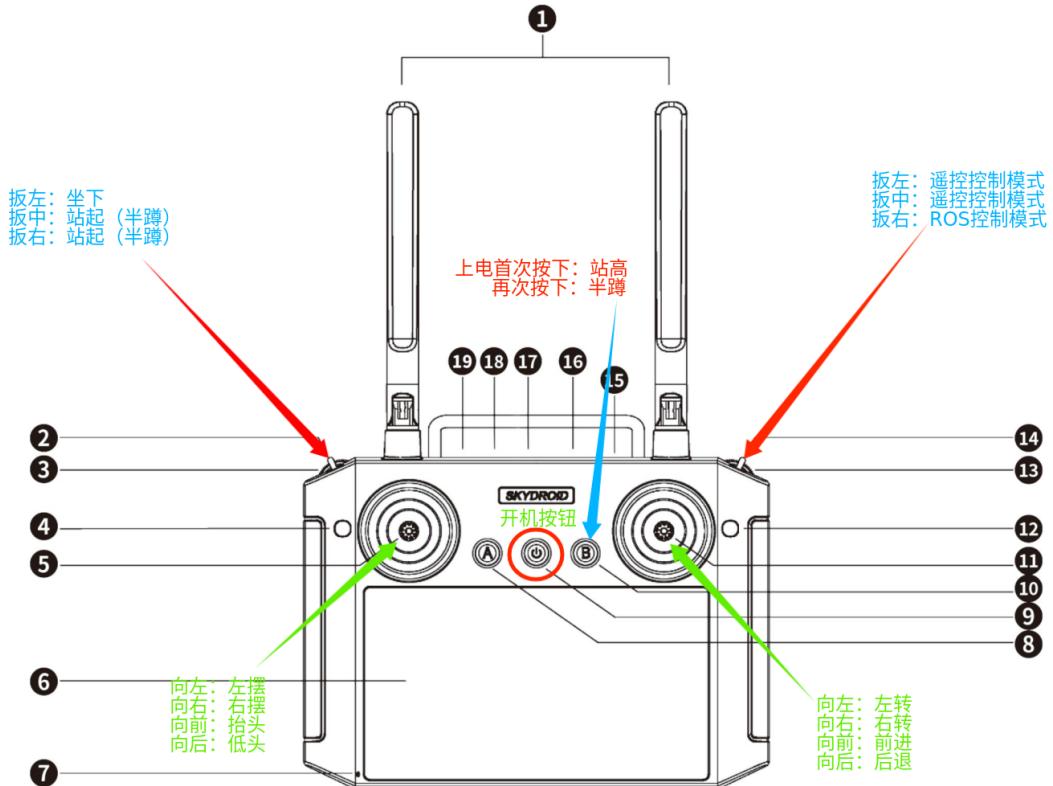


## 1.3 HCX 机器人运动控制

### Contents

- 遥控器控制
- ROS 控制
  - 话题终端发布
  - *rqt* 可视化发布
  - *teleop* 键盘控制

### 1.3.1 遥控器控制



#### 小技巧:

- 更多详细内容可以参考, 刑天用户手册

### 1.3.2 ROS 控制

在 ROS 部分, 我们使用 `geometry_msgs/Twist` 消息来控制机器人的运动, 该消息包含线速度和角速度。

**备注:** 首先请确保运行了如下命令, 正确启动了机器人整机 ROS 驱动

```
ros2 launch diablo_bringup diablo_bringup.launch.py
```

**小技巧:** 发现 rviz 界面全局轨迹正常生成, 但是小车未动, 请检查遥控器上是否禁用遥控模式

**备注:** 一旦发现机器人快要发生碰撞或产生危险时, 请立即将遥控器右拨动开关扳至左侧或中侧以切换到遥控接管模式 (ROS 控制禁用模式)

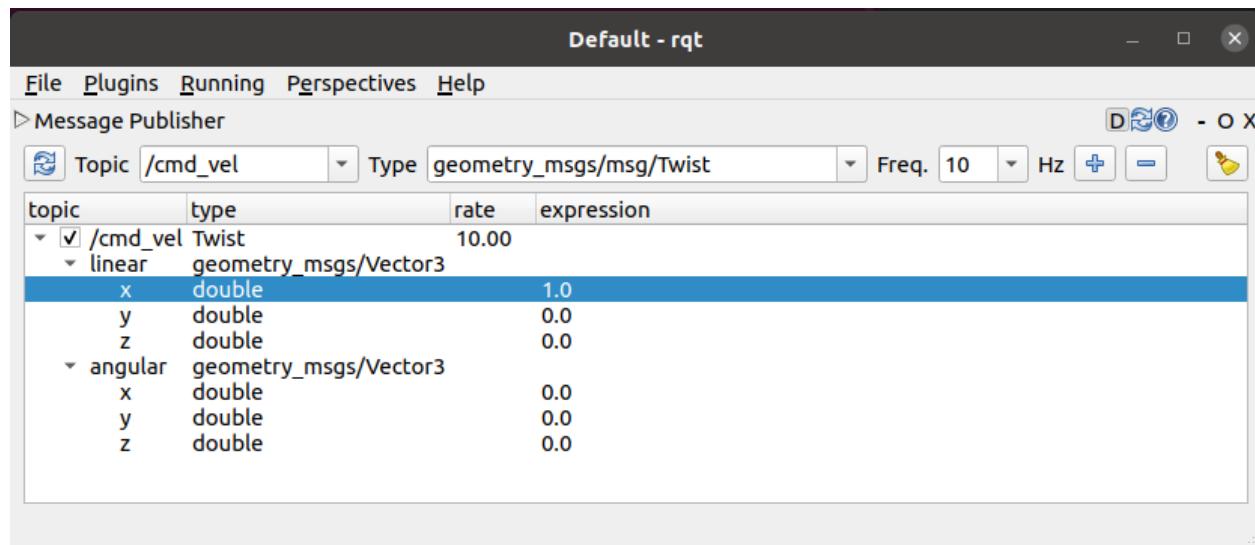
在发布方式上，有如下三种：

### 话题终端发布

```
ros2 topic pub /cmd_vel geometry_msgs/msg/Twist "{linear: {x: 0.1, y: 0.0, z: 0.0},
↳angular: {x: 0.0, y: 0.0, z: 1.0}}" -r 10
```

**小技巧：**与 ROS1 不同，ROS2 的发布消息时自动补全不是很友好，需要手动以类 json 格式输入消息内

### rqt 可视化发布



### teleop 键盘控制

可以使用 teleop 包来控制机器人，运行如下命令即可：

```
ros2 run diablo_teleop teleop_node
```

#### 危险：

- w：控制机器人向前移动。(-1.0~+1.0 米/秒)； (-1.6~+1.6 米/秒 Low-speed mode::High-speed mode::)
- s：控制机器人向后移动。(-1.0~+1.0 米/秒)； (-1.6~+1.6 米/秒 Low-speed mode::High-speed mode::)
- a：控制机器人左转。(-5.0~+5.0 弧度/秒) Arbitrarily mode::
- d：控制机器人右转。(-5.0~+5.0 弧度/秒) Arbitrarily mode::
- q：控制机器人向左倾斜。(-0.2~+0.2 弧度/秒) Standing mode::
- e：控制机器人向右倾斜。(-0.2~+0.2 弧度/秒) Standing mode::
- r：将机身倾斜角度调整为水平。Standing mode::

- z: 将机器人切换到站立模式。
- x: 将机器人切换到爬行模式。
- v: 用于提升机器人的控制模式。 (0 ~ 1) Position mode 0;
- b: 用于提升机器人的控制模式。 (-0.25 ~ +0.25 米/秒) Position mode 1;
- n: 用于机器人头部俯仰的控制模式。 (0 ~ 1) Position mode 0;
- m: 用于机器人头部俯仰的控制模式。 (-0.3~ +0.3 弧度/秒) Position mode 1;
- h: 站立模式下的最小高度。Position mode
- k: 站立模式下的中等高度。Position mode
- j: 站立模式下的最大高度。Position mode
- u: 控制机器人上仰。Position mode
- i: 将机身调整为水平。Position mode
- o: 控制机器人下仰。Position mode
- f: 太空步。dance mode
- g: 太空步结束。dance mode
- c: 跳跃模式。Jump mode
- ‘: 退出虚拟遥控器。

## 1.4 HCX 机器人 SLAM 建图

### Contents

- 2D 激光 SLAM 建图
  - Gmapping 建图
  - SLAM\_TOOLBOX 建图
  - Cartographer 建图
- 查看 2D 栅格地图
- 保存并 2D 栅格地图

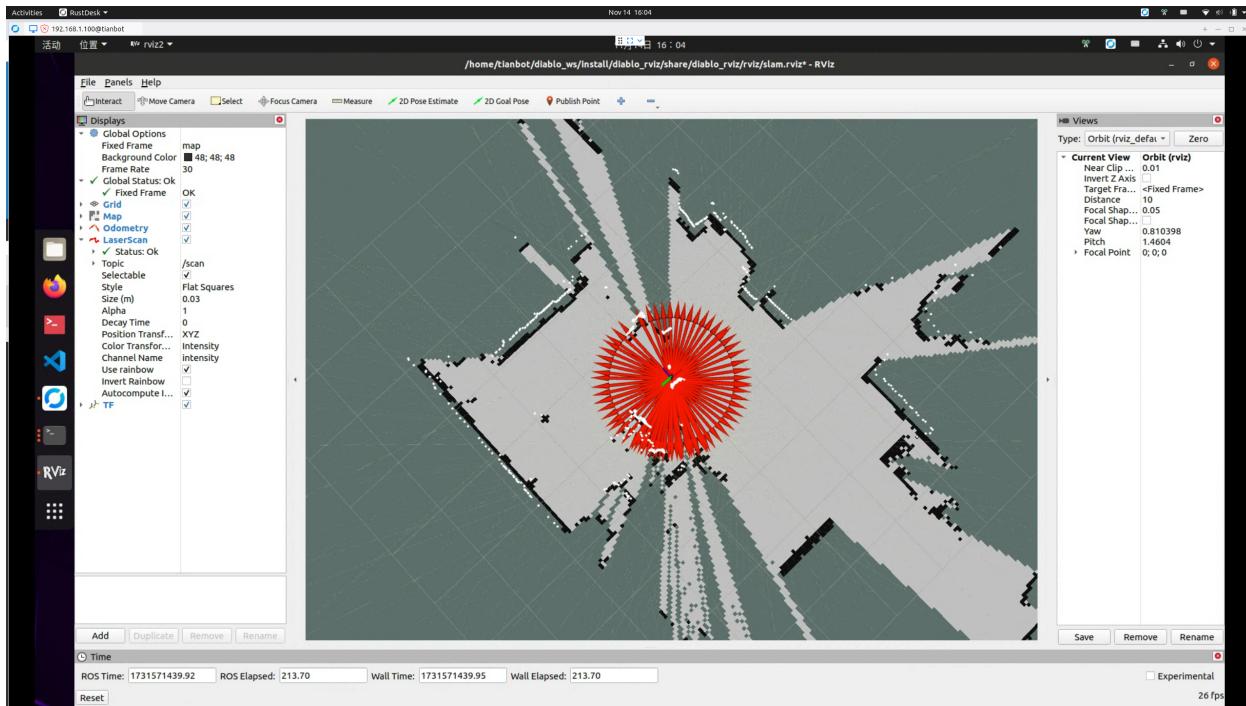
### 1.4.1 2D 激光 SLAM 建图

**备注:** 首先请确保运行了如下命令, 正确启动了机器人整机 ROS 驱动

```
ros2 launch diablo_bringup diablo_bringup.launch.py
```

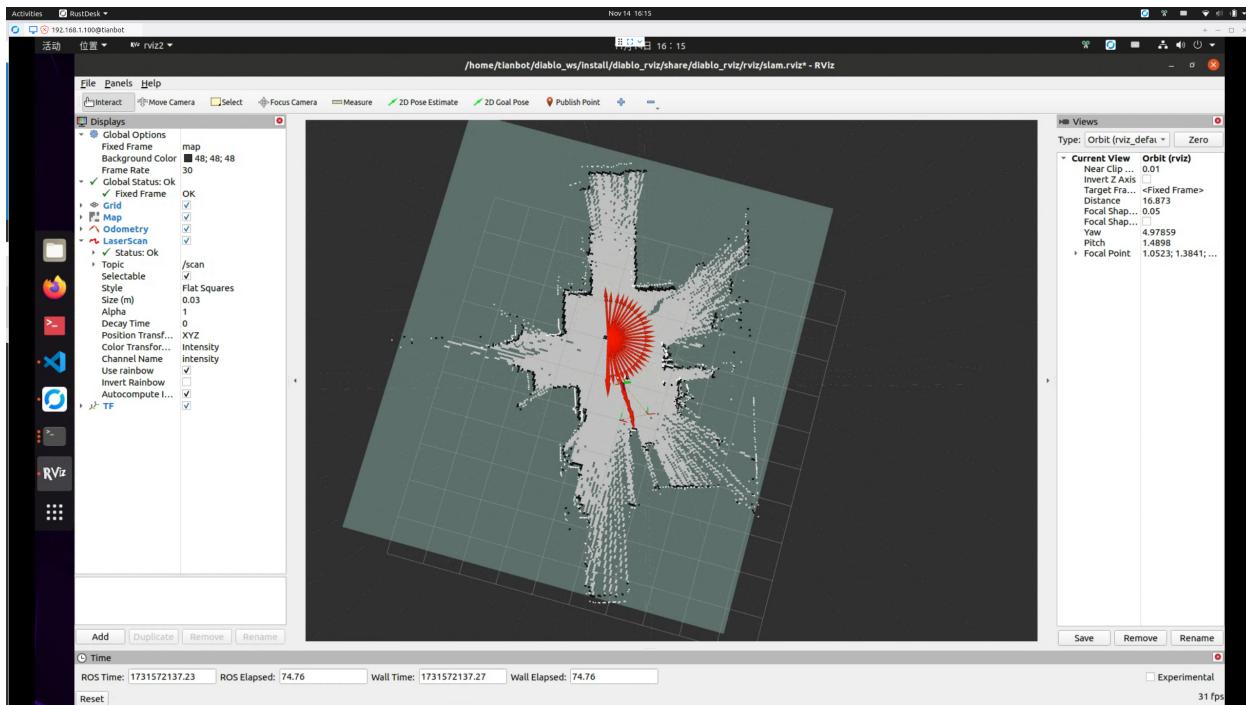
## Gmapping 建图

```
ros2 launch diablo_slam gmapping.launch.py
```



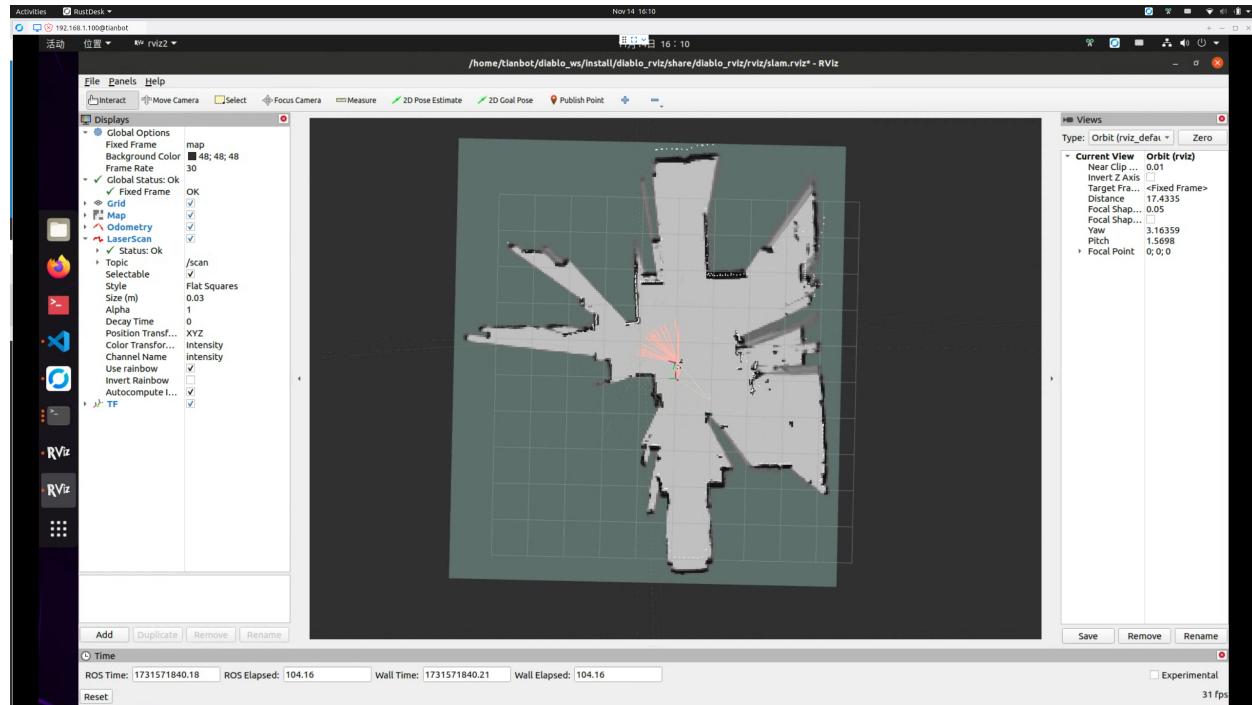
## SLAM\_TOOLBOX 建图

```
ros2 launch diablo_slam slam_toolbox.launch.py
```



## Cartographer 建图

```
ros2 launch diablo_slam cartographer_2d.launch.py
```



## 1.4.2 查看 2D 栅格地图

```
ros2 launch diablo_rviz view_map.launch.py
```

## 1.4.3 保存并 2D 栅格地图

```
ros2 launch diablo_slam map_save.launch.py
```

保存时，终端日志中会显示地图文件的将要保存到的全局路径，例如：

```
tianbot@diablo:~$ ros2 launch diablo_slam map_save.launch.py
[INFO] [launch]: All log files can be found below /home/tianbot/.ros/log/2024-11-14-
→16-17-21-378905-tianbot-483835
[INFO] [launch]: Default logging verbosity is set to INFO
[INFO] [map_saver_cli-1]: process started with pid [483871]
[map_saver_cli-1] [INFO] [1731572241.495636712] [map_saver_cli]:
[map_saver_cli-1] map_saver_cli 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] [1731572241.498539095] [map_saver_cli]: Creating
[map_saver_cli-1] [INFO] [1731572241.499058912] [map_saver_cli]: Saving map from 'map
→' topic to '/home/tianbot/diablo_ws/src/diablo_ros2/diablo_navigation2/maps/map_
→2024-11-14-161721' file
```

(续下页)

(接上页)

```
[map_saver_cli-1] [WARN] [1731572241.499094476] [map_saver_cli]: Free threshold ↴unspecified. Setting it to default value: 0.250000
[map_saver_cli-1] [WARN] [1731572241.499118518] [map_saver_cli]: Occupied threshold ↴unspecified. Setting it to default value: 0.650000
[map_saver_cli-1] [WARN] [map_io]: Image format unspecified. Setting it to: pgm
[map_saver_cli-1] [INFO] [map_io]: Received a 209 X 236 map @ 0.05 m/pix
[map_saver_cli-1] [INFO] [map_io]: Writing map occupancy data to /home/tianbot/diablo_ws/src/diablo_ros2/diablo_navigation2/maps/map_2024-11-14-161721.pgm
[map_saver_cli-1] [INFO] [map_io]: Writing map metadata to /home/tianbot/diablo_ws/src/diablo_ros2/diablo_navigation2/maps/map_2024-11-14-161721.yaml
[map_saver_cli-1] [INFO] [map_io]: Map saved
[map_saver_cli-1] [INFO] [1731572241.580101996] [map_saver_cli]: Map saved ↴successfully
[map_saver_cli-1] [INFO] [1731572241.580286026] [map_saver_cli]: Destroying
[INFO] [map_saver_cli-1]: process has finished cleanly [pid 483871]
```

**备注:** 保存地图时, 请确保机器人已经停止运动, 否则保存的地图可能不完整。保存时, 终端日志中会显示地图文件的将要保存到的全局路径, 例如:

- Saving map from 'map' topic to '/home/tianbot/diablo\_ws/src/diablo\_ros2/diablo\_navigation2/maps/' 目录下。

如果保存的地图文件较大, 保存过程可能需要一些时间, 请耐心等待。有一定可能保存失败, 请重新尝试。

## 1.5 HCX 机器人 2D 导航

### Contents

- Navigation2 导航框架
  - 启动导航节点
  - 设置初始位置
  - 设置导航目标点

### 1.5.1 Navigation2 导航框架

diablo\_navigation 基于 ROS2 的 Navigation2 框架, Navigation2 框架提供了多种导航算法, 包括 A\*、D\*、RRT、RRT\*、PRM 等, 以及相应的路径规划、路径跟踪、避障等功能。

**备注:** 首先请确保运行了如下命令, 正确启动了机器人整机 ROS 驱动

```
ros2 launch diablo_bringup diablo_bringup.launch.py
```

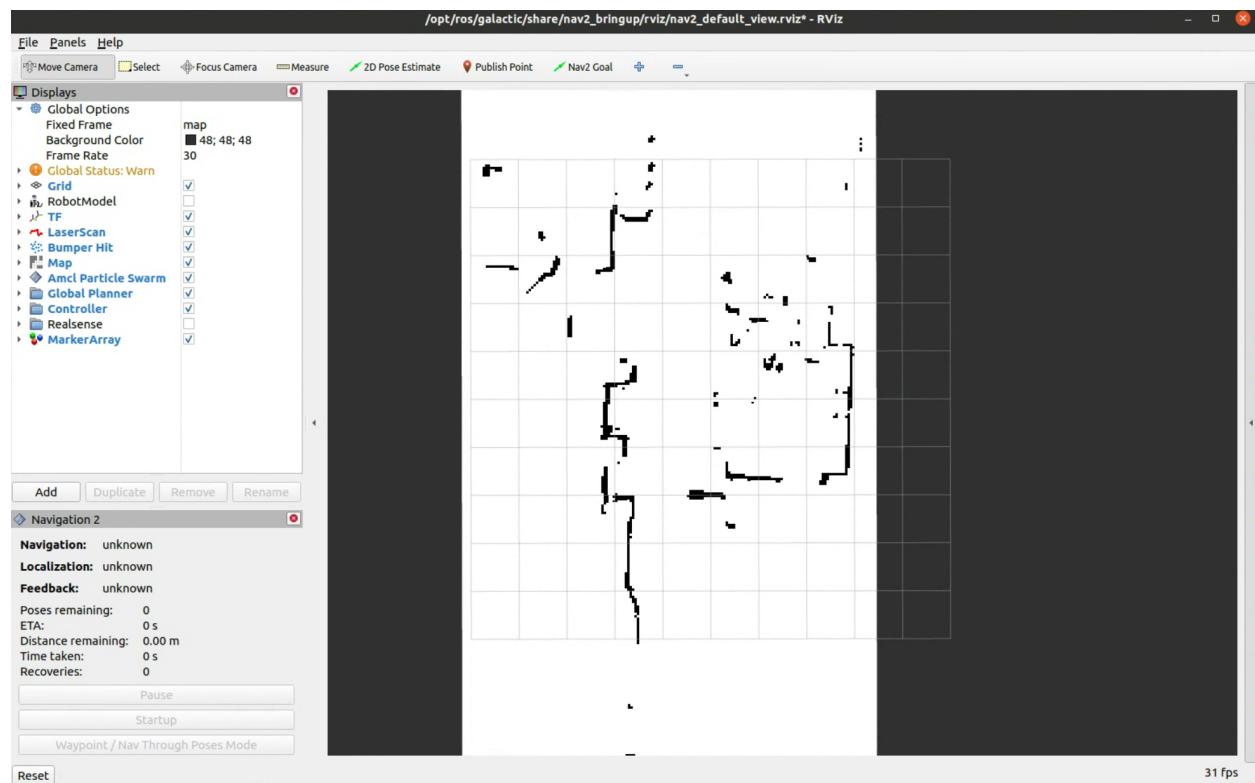
## 启动导航节点

```
ros2 launch diablo_navigation2 nav2.launch.py use_map:=xxxx #_
→启动导航节点，并指定地图文件
```

**备注:** xxxx 为地图文件的名称, 例如: /home/diablo/xxx.yaml, 指定时无需添加.yaml 后缀

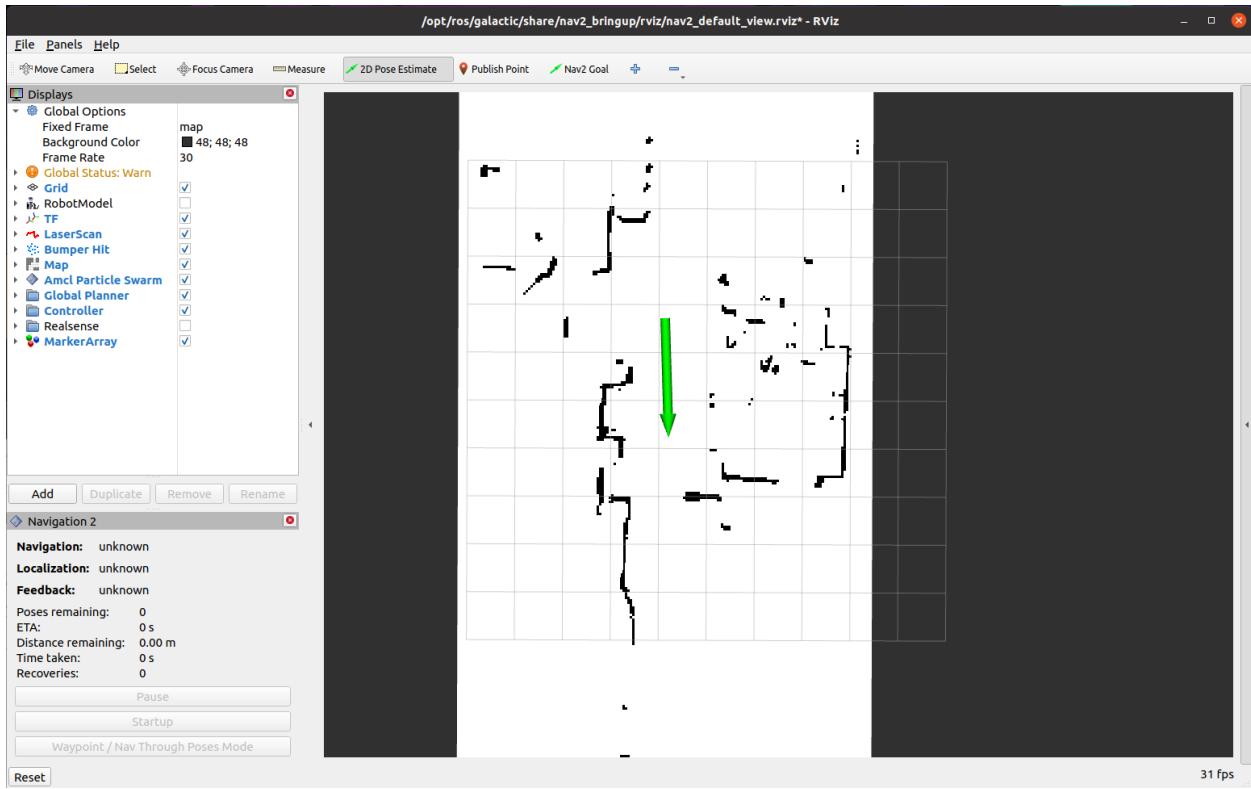
此时可以看到 rviz2 界面中已经加载了地图, 并且机器人暂时还未启动导航模式, 需要手动设置在地图中的初始位置

**备注:** 如果没有加载地图, 请检查 use\_map: = 传输的地图文件路径是否正确, 或者地图文件格式是否真存在



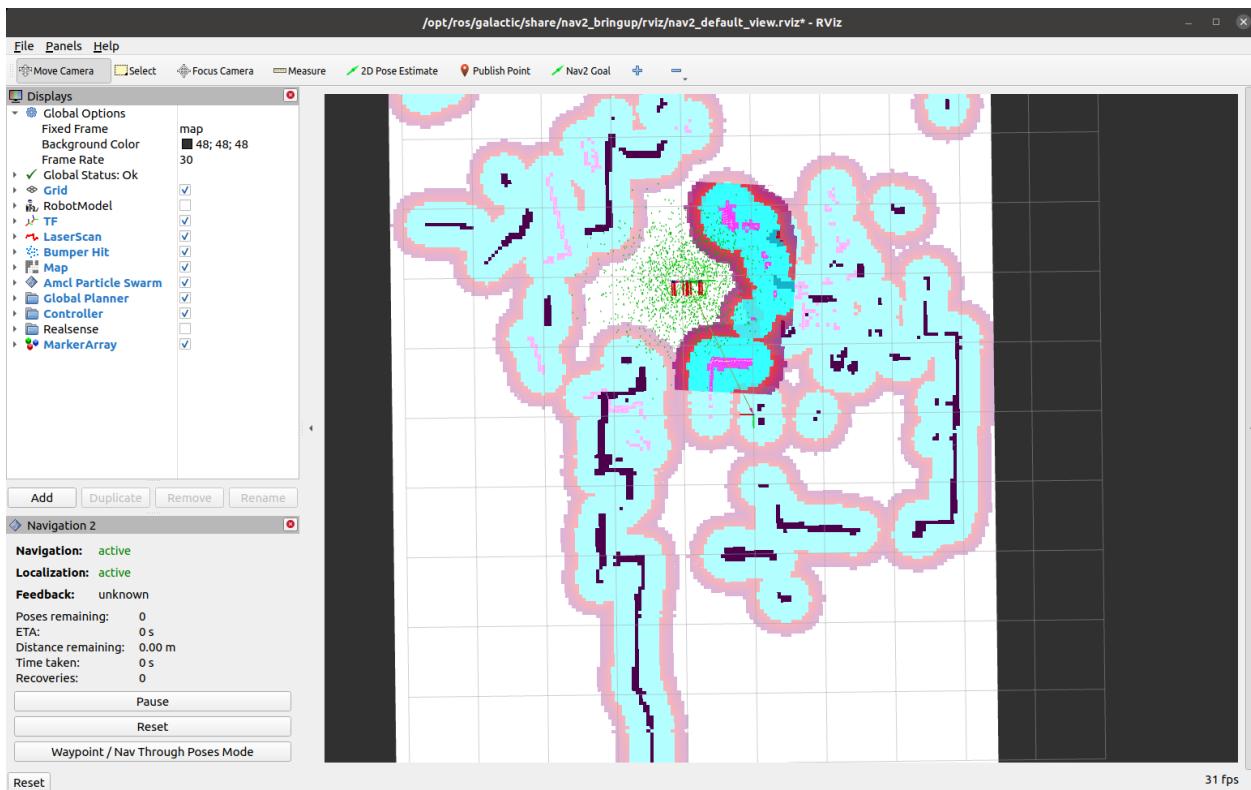
## 设置初始位置

通过 rviz 界面的 2D pose Estimate 工具给定机器人初始位姿



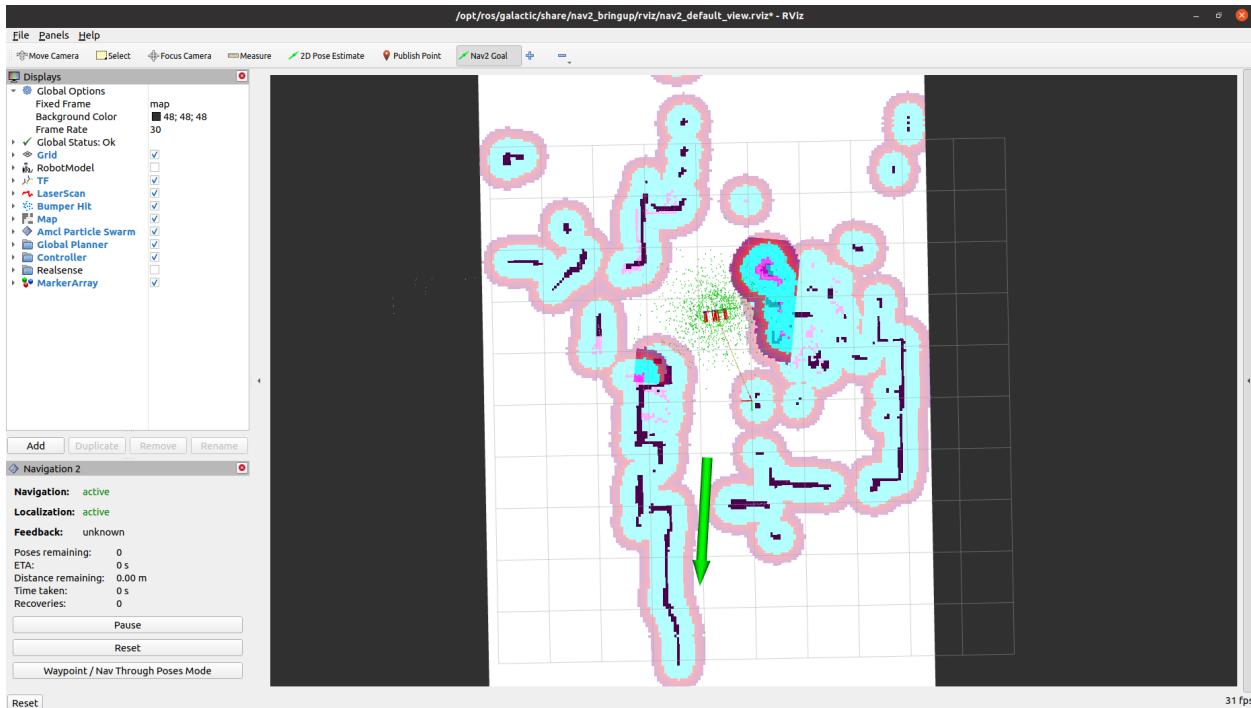
**小技巧:** 在设置初始位置时，需要尽量将初始位置点与机器人的真实位置相对应，姿态上有一点误差也问题不大，可以使用遥控器稍微挪动一下位置，机器人会自动进行修正

给定初始位置后，机器人会激活导航模式，并开始接受目标点并规划全局路径

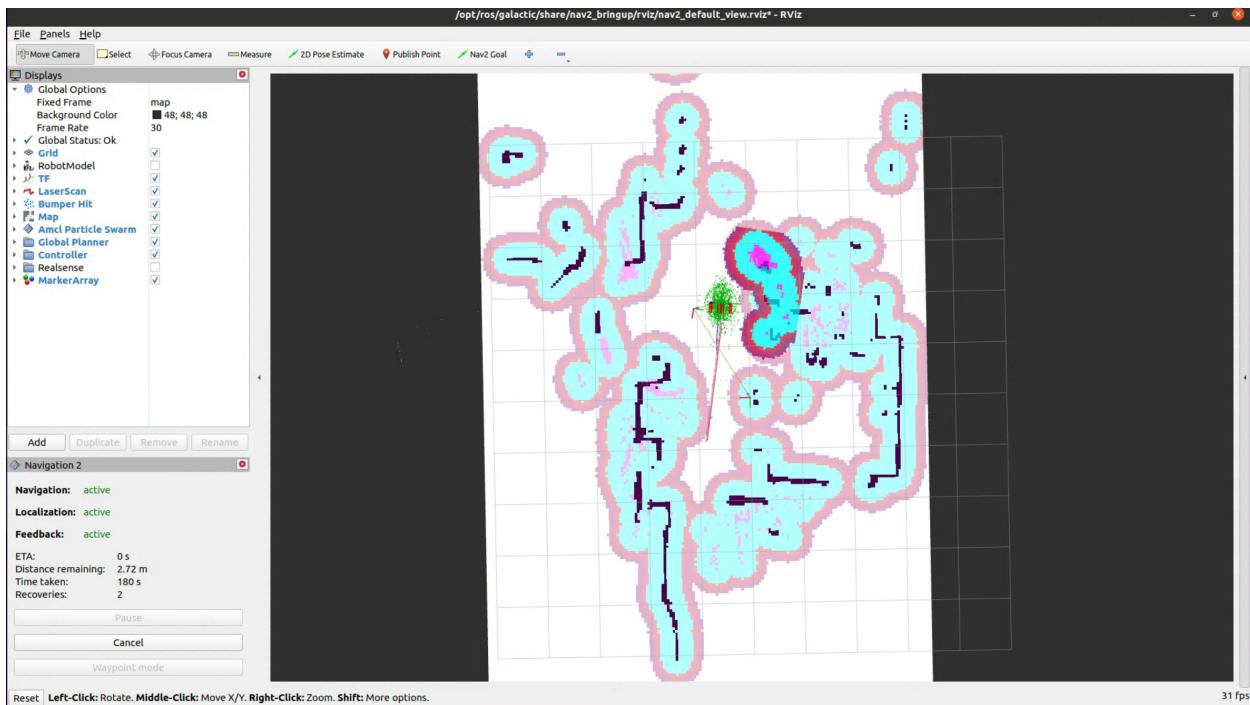


## 设置导航目标点

利用 Navigation2 Goal 按键在地图上给定目标点，单击鼠标左键确认目标点位置后，继续拖动鼠标左键，选择到达目标点处的方向



此时看到，机器人已成功规划出全局路径，并准备开始向目标点移动



此时我们可以将遥控器上右拨动开关扳至右侧以切换到遥控禁用模式（ROS 控制模式），机器人则会自动开始规划路径进行导航

**小技巧：**发现 rviz 界面全局轨迹正常生成，但是小车未动，请检查遥控器上是否禁用遥控模式

**备注：**一旦发现机器人快要发生碰撞或产生危险时，请立即将遥控器右拨动开关扳至左侧或中侧以切换到遥控接管模式（ROS 控制禁用模式）

## 1.6 HCX 机器人驱动自启服务

### Contents

- 使用方法
  - 开启驱动自启服务
  - 查看驱动自启服务状态
  - 重启驱动自启服务
  - 停止驱动自启服务
  - 启动自启服务
  - 禁用自启服务

## 1.6.1 使用方法

### 开启驱动自启服务

```
systemctl start diablo_startup.service # 立即开启驱动自启服务
```

### 查看驱动自启服务状态

```
systemctl status diablo_startup.service # 查看驱动自启服务状态
```

### 重启驱动自启服务

```
systemctl restart diablo_startup.service # 重启驱动自启服务
```

### 停止驱动自启服务

```
systemctl stop diablo_startup.service # 停止驱动自启服务
```

### 启动自启服务

```
systemctl enable diablo_startup.service # 开启开机驱动自启服务
```

### 禁用自启服务

```
systemctl disable diablo_startup.service # 禁止开机驱动自启服务
```

本页介绍如何在您将如何通过 ROS2 来使用 HCX 机器人

- 如何连接 HCX 机器人
- 启动整机驱动和可视化传感器数据
- HCX 机器人运动控制
- HCX 机器人 SLAM 建图
- HCX 机器人 2D 导航
- HCX 机器人驱动自启服务



# CHAPTER 2

---

## 节点说明

---

### 2.1 传感器 (ros 端)

#### Contents

- 传感器数据源接口

#### 2.1.1 传感器数据源接口

以下是用户传感器接口，用户可以订阅这些 ROS 话题接口接受消息，以控制机器人传感器数据

传感器名称	话题名称	话题类型/发布者数量	发布频率
RGB-D 相机	/camera/color/camera_info	[sensor_msgs/msg/CameraInfo] 1 publisher	15hz
RGB-D 相机	/camera/color/image_raw	[sensor_msgs/msg/Image] 1 publisher	15hz
RGB-D 相机	/camera/color/metadata	[realsense2_camera_msgs/msg/Metadata] 1 publisher	15hz
RGB-D 相机	/camera/depth/camera_info	[sensor_msgs/msg/CameraInfo] 1 publisher	15hz
RGB-D 相机	/camera/depth/image_rect_raw	[sensor_msgs/msg/Image] 1 publisher	15hz
RGB-D 相机	/camera/depth/metadata	[realsense2_camera_msgs/msg/Metadata] 1 publisher	15hz
RGB-D 相机	/camera/extrinsics/depth_to_color	[realsense2_camera_msgs/msg/Extrinsics] 1 publisher	
RGB-D 相机	/camera/extrinsics/depth_to_depthl	[realsense2_camera_msgs/msg/Extrinsics] 1 publisher	
RGB-D 相机	/camera imu	[sensor_msgs/msg/Imu] 1 publisher	
底盘电池电压	/diablo/sensor/Battery	[sensor_msgs/msg/BatteryState] 1 publisher	2hz
底盘状态	/diablo/sensor/Body_state	[motion_msgs/msg/RobotStatus] 1 publisher	10hz
底盘 imu	/diablo/sensor/Imu	[sensor_msgs/msg/Imu] 1 publisher	30hz
底盘电池电压	/diablo/sensor/ImuEuler	[ception_msgs/msg/IMUEuler] 1 publisher	满电电压 42v 30hz
底盘电机状态	/diablo/sensor/Motors	[motion_msgs/msg/LegMotors] 1 publisher	30hz
运动学底盘轮式里程计	/diablo_odom	[nav_msgs/msg/Odometry] 1 publisher	30hz
	/diagnostics	[diagnostic_msgs/msg/DiagnosticArray] 1 publisher	2hz
底盘电池电压	/joint_states	[sensor_msgs/msg/JointState] 1 publisher	10hz
多线激光雷达	/livox/imu	[sensor_msgs/msg/Imu] 1 publisher	200hz
多线激光雷达	/livox/lidar	[sensor_msgs/msg/PointCloud2] 1 publisher	10hz
mqtt 话题接受端	/mqtt_rx	[std_msgs/msg/String] 1 publisher	3hz
mqtt 话题发送端	/mqtt_tx	[std_msgs/msg/String] 1 publisher	3hz
ekf 底盘轮式里程计	/odom	[nav_msgs/msg/Odometry] 1 publisher	20hz
	/parameter_events	[rcl_interfaces/msg/ParameterEvent] 20 publishers	
机器人 urdf 描述	/robot_description	[std_msgs/msg/String] 1 publisher	
	/rosout	[rcl_interfaces/msg/Log] 22 publishers	3hz
2D 激光雷达	/scan	[sensor_msgs/msg/LaserScan] 1 publisher	10hz
关键部件的坐标变换关系	/tf	[tf2_msgs/msg/TFMessage] 2 publishers	30hz
关键部件的坐标变换关系	/tf_static	[tf2_msgs/msg/TFMessage]	

## 2.2 用户接口 (ros 端)

### Contents

- 用户控制接口
- */mqtt\_txd* 接口说明
  - */led* 接口说明
  - */Stand* 接口说明

### 2.2.1 用户控制接口

以下是用户接口，用户可以向这些 ROS 话题接口发布消息，以控制机器人外设和运动状态

外设接口名称	话题名称	话题类型/发布者数量	发布频率
mqtt 接口	/mqtt_txd	[std_msgs/msg/String]	10hz
速度控制接口	/cmd_vel	[geometry_msgs/msg/Twist]	10hz
站立控制接口	/stand	[std_msgs/msg/String]	10hz
~~	键盘控制接口	/key_control	[motion_msgs/msg/MotionCtrl]
~~	MotionCmd 运动接口	/diablo/MotionCmd	[motion_msgs/msg/MotionCtrl]
照明灯接口	/illumination_led_color	[std_msgs/msg/ColorRGBA]	10hz
氛围灯	/led	[std_msgs/msg/String]	10hz
左氛围灯	/left_led_color	[std_msgs/msg/ColorRGBA]	10hz
中氛围灯	/middle_led_color	[std_msgs/msg/ColorRGBA]	10hz
右氛围灯	/right_led_color	[std_msgs/msg/ColorRGBA]	10hz

### 2.2.2 /mqtt\_txd 接口说明

字段	类型	描述	机器人状态
data	string	settled led_num r g b	控制探照灯、f 氛围灯
data	string	setvel linear_vel angular_vel	控制机器人运动
data	string	setpose num	num 控制 z 位置的状态值，1：站起，0：蹲下
data	string	BATT xxxV	向 mqtt 以 10hz 持续返回电池电量

### /led 接口说明

字段	类型	描述	机器人状态
data	string	0 r g b	开启探照灯
data	string	0 0 0 0	关闭探照灯
data	string	1 r g b	控制右侧氛围灯, x x x 依次为 rgb 的值
data	string	1 0 0 0	关闭右侧氛围灯
data	string	2 r g b	控制中间氛围灯, x x x 依次为 rgb 的值
data	string	2 0 0 0	关闭中间氛围灯
data	string	3 r g b	控制左侧氛围灯, x x x 依次为 rgb 的值
data	string	3 0 0 0	关闭左侧氛围灯

### /Stand 接口说明

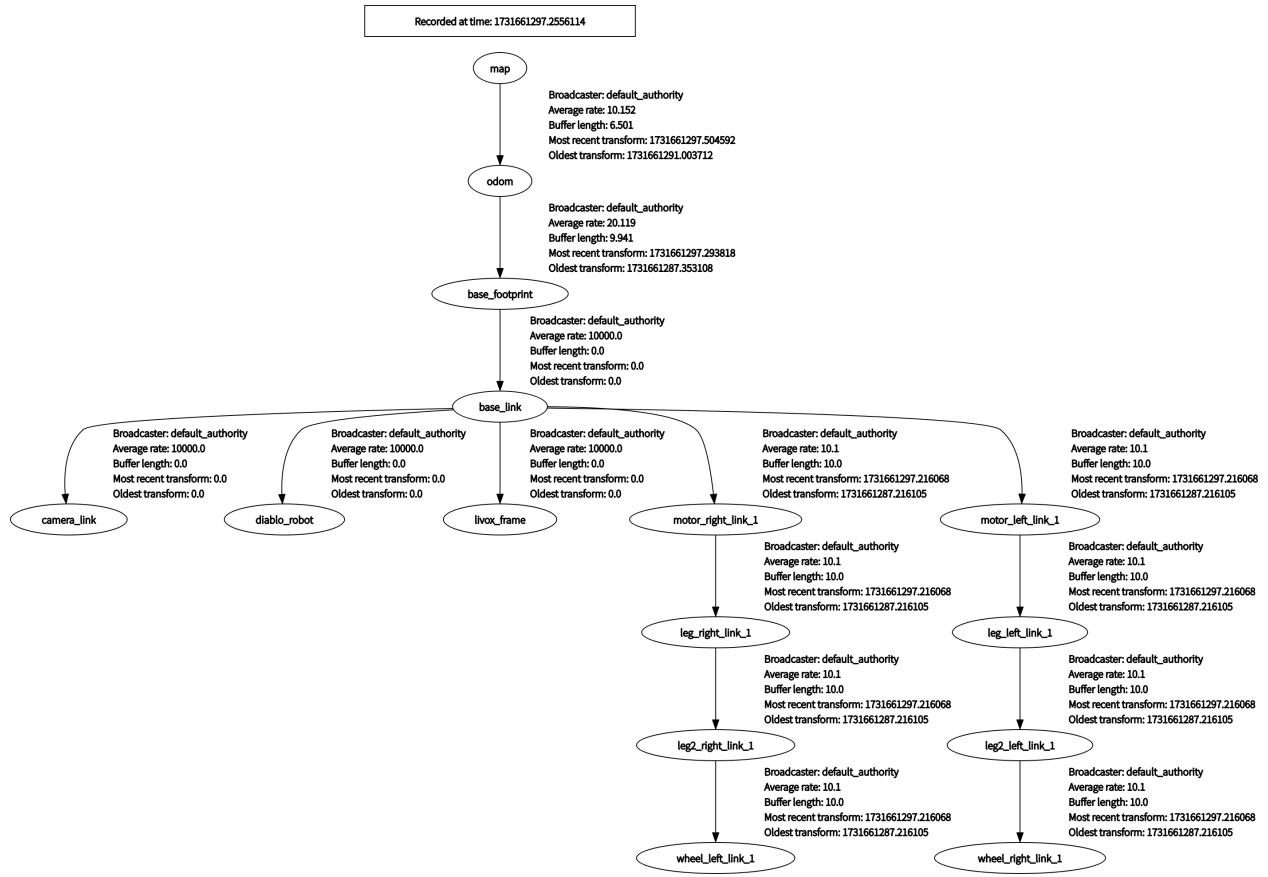
字段	类型	描述	机器人状态
data	string	standup	站立
data	string	sitdown	坐下
data	string	其他 string	保持状态

## 2.3 URDF 描述文件

### Contents

- TF 关系树

### 2.3.1 TF 关系树



## 2.4 可视化

### TODO

本页介绍 HCX 机器人的传感器数据源接口和用户控制接口

- HCX 机器人的传感器数据源
- HCX 机器人的用户控制接口
- HCX 机器人的描述信息
- HCX 机器人 Gazebo 可视化

基于 diablo 平台的 HCX 机器人支持 2D 的室内外自主导航功能，您可以通过 ROS2 快速上手。如果您不准备使用 ROS 进行开发，也可以通过在 ROS 中修改 CMakeLists 的方式仅对源码进行编译。我们将持续更新 HCX 机器人的 ROS2 的功能节点，希望能对您的机器人开发有所帮助。

两轮机器人 - 产品效果图



# CHAPTER 3

---

## 整机配置

---

- 高性能运算平台 NUC
- RGB-D 深度相机 RealSense 系列
- 比肩多线的高性能激光雷达 Livox 系列
- 4G 远程 FPV 摄像头
- 4G 高清数字图传一体远程遥控器
- 独立 4G DTU 模块（自带路由器）