轮 趣 科 技

WHEELTEC R550 FOR X3 使用手册

推荐关注我们的公众号获取更新资料

版本说明：

版本	日期	内容说明
V1.0	2022/09/27	第一次发布
V2.0	2023/11/30	更新骨架识别功能及 ROS2 源码
V2.1	2023/12/29	更新系统版本为 Humble

网址：www.wheeltec.net

序言

WHEELTEC R550 FOR X3 是基于地平线 X3 派搭建的一款 ROS2 小车，同时支持 ROS 与 TogetherROS 两种机器人开发框架。基于 X3 派强大的 AI 性能，结合全面升级的 TOF 激光雷达与 1080P 相机，可以满足 ROS2 和 AI 入门学习。本文档为 WHEELTEC R550 小车的上手使用教程。

目录

1. ROS2 功能调试

目前 WHEELTEC R550 小车已更新为 ROS2-Humble 版本，以下功能的使用方法与 ROS2-Foxy 版本基本一致。本文档以 ROS2-Humble 版本为例，为用户简述 demo 的使用方法。

在 WHEELTEC R550 小车上调试 ROS2-Humble 功能，需要先确认目前终端处于 Humble 开发框架[默认是 Humble 的环境]，详细步骤请查看 2.2 小节。ROS2-Humble 源码在 wheeltec_ros2 文件夹中，Tros 控制源码在 wheeltec_tros 文件夹中。

1.1 机器人底盘控制

turn_on_wheeltec_robot 功能包主要是启动底盘控制、传感器与 base_footprint 的 TF 静态转换、机器人描述文件以及融合滤波的功能。

turn_on_wheeltec_robot 功能包的启动文件主要有：

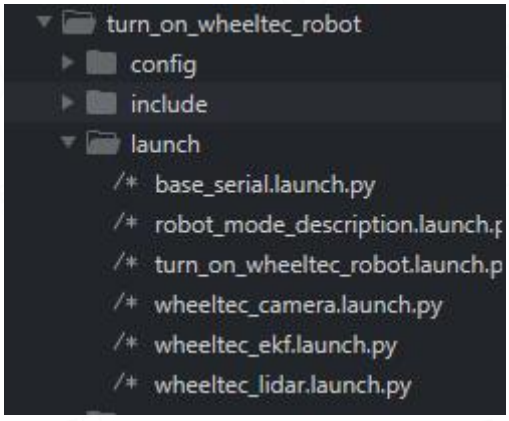

图 1-1-1　turn_on_wheeltec_robot 功能包

其中 base_serial.launch.py 是启动底盘控制节点，实现 ROS 主控与 STM32 的通信；

robot_and_lidar.launch.py 文件是打开底盘控制和雷达节点，主要用于验证传感器是否正常；

robot_mode_description.launch.py 用于导入小车的 urdf 模型，包括传感器和机器人底层的静态 tf 转换；

wheeltec_lidar.launch.py 为 WHEELTEC ROS2 小车的雷达控制节点，用户在这个文件中切换雷达型号，WHEELTEC R550 小车默认使用 n10 雷达，用户不必

修改该文件；

turn_on_wheeltec_robot.launch.py 包含底盘控制、传感器与 base_footprint 的 TF 静态转换、机器人描述文件以及融合滤波的功能。

turn_on_wheeltec_robot 功能包主要用于启动机器人底盘，运行指令为：

```
ros2 launch turn_on_wheeltec_robot turn_on_wheeltec_robot.launch.py
```

正常运行如图所示：

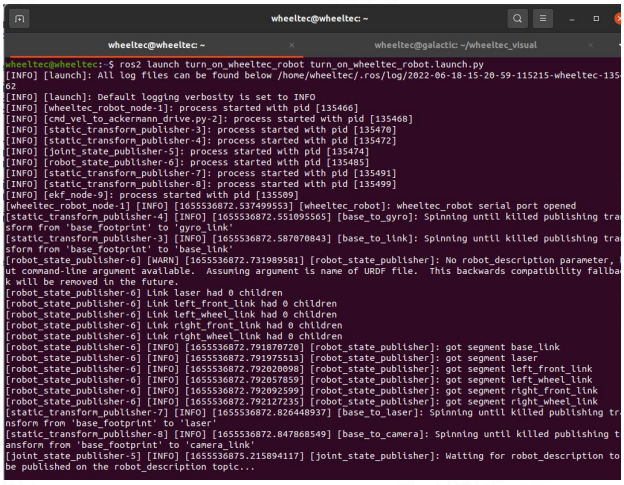

图 1-1-2　运行 turn_on_wheeltec_robot.launch.py

此时打开 RVIZ2 可以看到小车的 urdf 模型：

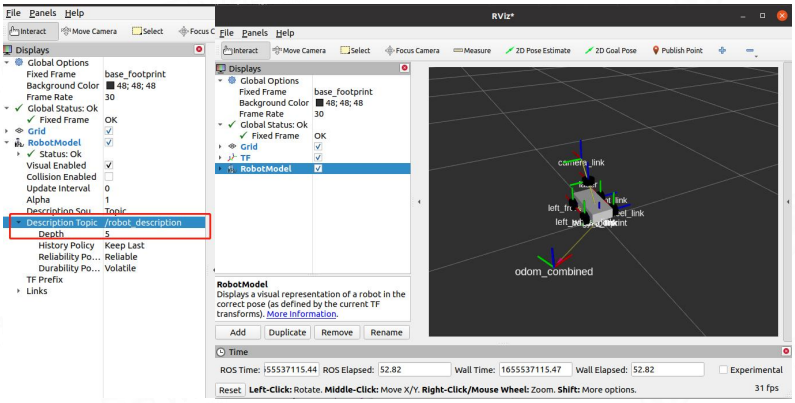

图 1-1-3　rviz2 中的机器人模型

1.2 RGB 视觉巡线

视觉巡线启动文件位于 simple_follower_ros2 功能包中，simple_follower_ros2 为简单跟随功能包，其中包括视觉巡线、雷达、色块跟随【需搭配深度相机使用】功能。

simple_follower_ros2 中的 line_follower.launch.py 文件用于启动视觉巡线功能，该文件包括启动底盘控制和 USB 相机。通过以下指令远程登录到小车端，需要添加-Y 参数打开远程传输图像权限。

```
ssh -Y wheeltec@192.168.0.100
```

运行视觉巡线程序，如图所示：

```
ros2 launch simple_follower_ros2 line_follower.launch.py
```

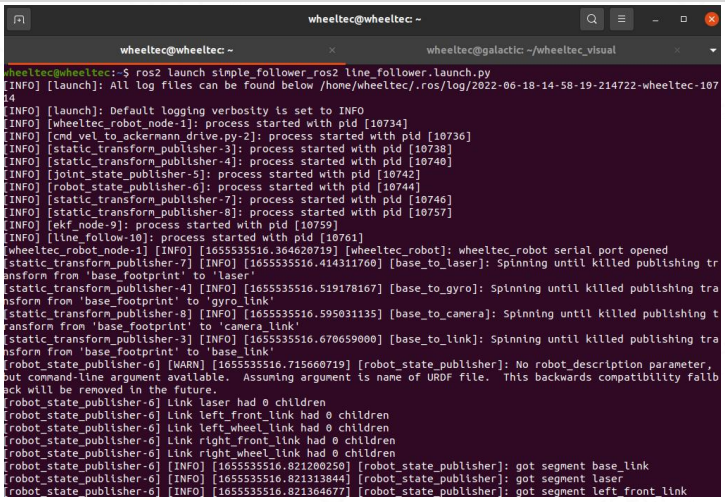

图 1-2-1　运行 line_follower.launch.py

在弹出的弹窗选择小车进行巡线的颜色，以蓝色为例，将滑块滑到【2】，在窗口中白色为识别到指定颜色的结果。

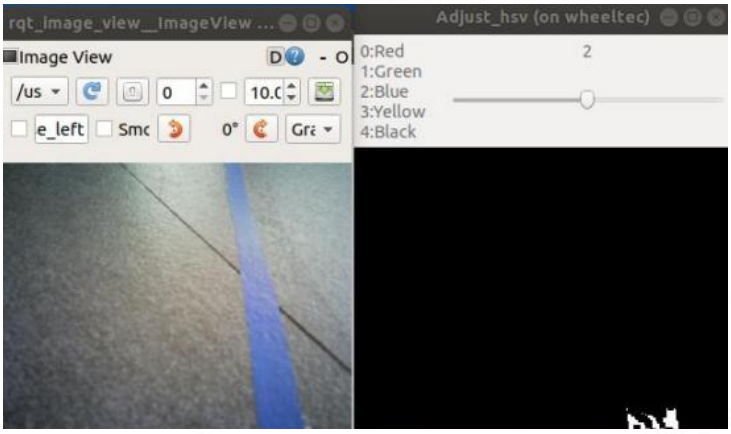

图 1-2-2　运行 line_follower.launch.py 效果

使用 ctrl+c 停止程序后，或会出现小车依然往前走的现象。此时需要用户手动按下 stm32 的复位按键，小车即可停止运动。复位按键在中间层绿色的板子上，如图所示。

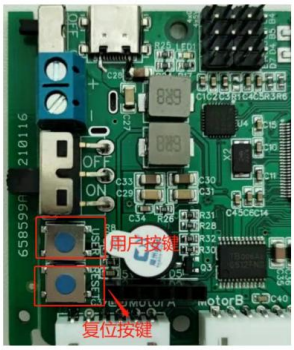

图 1-2-3　stm32 板子上的复位按键

1.3 雷达跟随

雷达跟随启动文件位于 simple_follower_ros2 功能中，laser_follower.launch.py 启动雷达跟随功能，包括打开底盘控制节点和雷达启动节点。启动命令为：

```
ros2 launch simple_follower_ros2 laser_follower.launch.py
```

雷达跟随启动后，小车会不断寻找雷达扫描范围内距离最近的目标，之后找到一个距离最近且合理的目标进行跟随，最终与被跟随物保持适当的距离（即中距值，预设 0.6m），小车与被跟随物体之间夹角为 0 左右。

注意：开启雷达跟随时尽量保证空间不要过于狭小，物与物之间的距离最好保持 1m 及以上。应注意雷达高度，由于雷达只能扫描到与其处于同一水平面上的物体，因此不要将高度较低的物品放置在小车行经路线上，以免发生碰撞。

1.4视觉跟踪

注意：视觉跟踪需要用到深度相机。visual_follower.launch.py 启动视觉跟随功能，包括底盘控制节点和相机启动节点。默认情况下，机器人视觉跟随目标为**红色物体**，它会通过目标物体的位置进行相关计算获得运动速度，从而实现对目标物体的跟随。

镜像默认使用 Astra S 相机运行视觉巡线功能。运行视觉跟踪功能：

```
ros2 launch simple_follower_ros2 visual_follower.launch.py
```

打开 rqt 工具可以查看色块与当前小车的距离：

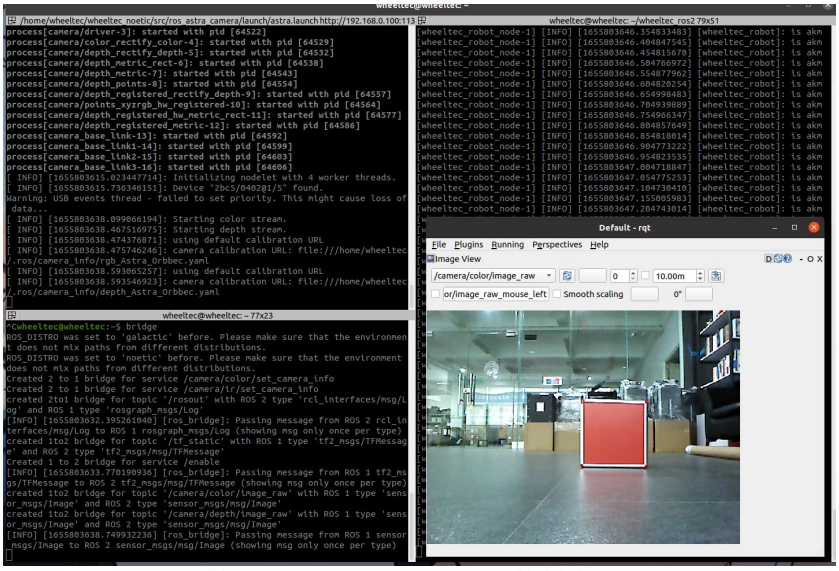

图 1-4-1　运行 visual_follower.launch.py 效果

其中如果使用的是 Gemini 和 Dabai 相机，可以在视觉跟随启动文件中修改 Height 为 400，如图 7-6-7 所示。

```
~/wheeltec_ros2/src/simple_follower_ros2/launch/visual_follower.launch.py
```

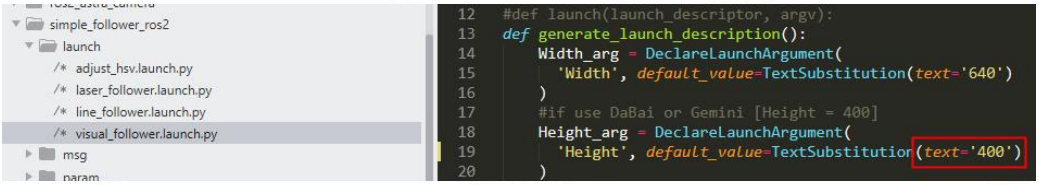

图 1-4-2　运行 visual_follower.launch.py 效果

1.5 2D 建图

wheeltec_robot_slam 功能包主要用于 WHEELTEC ROS2 小车进行 2D 建图。其中包括 gmapping、slam_toolbox 和 cartographer 三种建图算法。运行建图时可以使用蓝牙/航模/键盘控制节点小车运动。

① gmapping 建图

slamgapping 软件包包含用于 OpenSlam 的 Gmapping 的 ROS 包装器，它提供了基于激光的 SLAM（同时定位和映射），作为 slam_gmapping 的 ROS 节点。

使用 Gmapping 算法进行 2D 建图，指令如下。建图页面如图 3-2-3 所示。

```
ros2 launch slam_gmapping slam_gmapping.launch.py
```

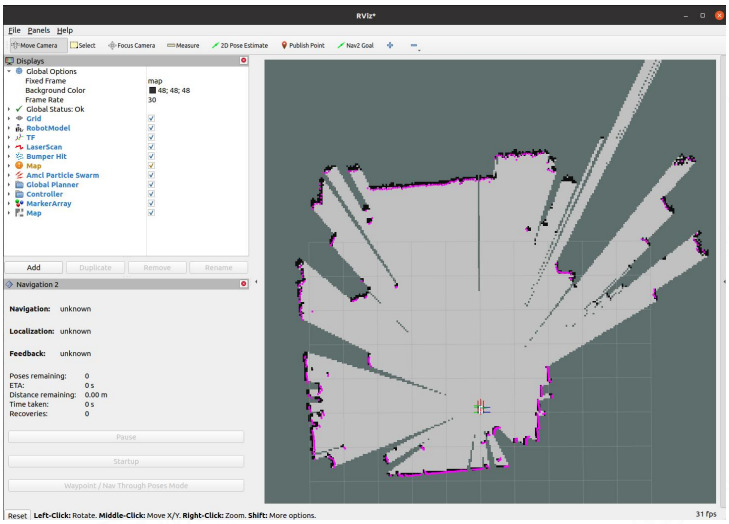

图 1-4-1　gmapping 建图

② slam_toolbox 建图

slam_toolbox 是 2D SLAM 构建的一组工具和功能，该软件包将允许用户完全序列化要重新加载的 SLAM 映射的数据和姿态图，以继续映射，本地化，合并或进行其他操作。这个建图算法支持构建大地图。（该软件包已在 5 倍以上实时高达约 30,000 平方英尺和 3 倍实时高达约 60,000 平方英尺的地图构建方面进行了基准测试。使用的最大面积是 200,000 平方英尺。）关于 slam_toolbox 算法更多详细的介绍可以这里找到。

使用 slam_toolbox 进行 2D 建图，指令如下：

```
ros2 launch wheeltec_slam_toolbox online_sync.launch.py
```

建图界面如图所示

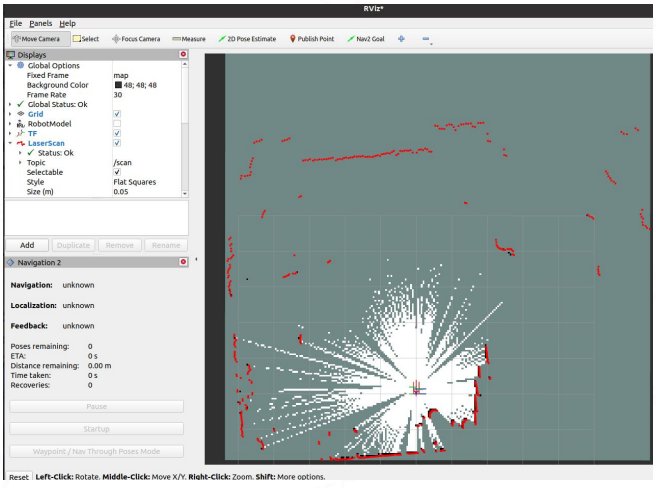

图 1-4-2　slam_toolbox 建图

③ slam_toolbox 工具保存地图

使用 slam_toolbox 建图时，系统可以调用 slam_toolbox 功能包中的 rviz2 插件，实现在 rviz2 中即时保存地图。注意：只针对 slam_toolbox 建图算法。主要步骤为：

Step1：运行 slam_toolbox 建图

```
ros2 launch wheeltec_slam_toolbox slam_toolbox.launch.py
```

Step2：打开 rviz2，点击左上角的 Panels 添加面板，点击 Add New Panel 选项。

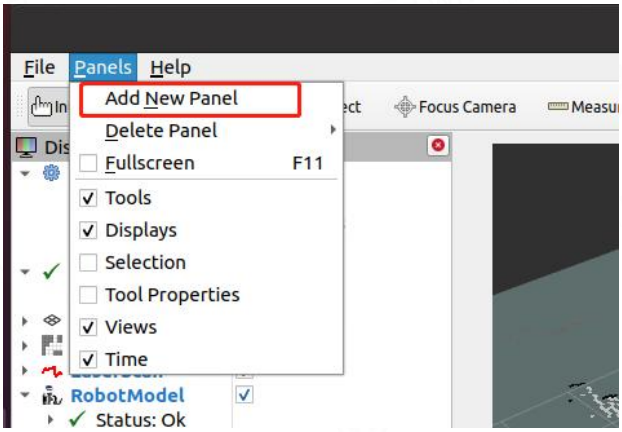

图 1-4-3　添加新的面板(1)

在弹窗中选择 slam_toolbox 一栏下的 SlamToolboxPlugin 插件

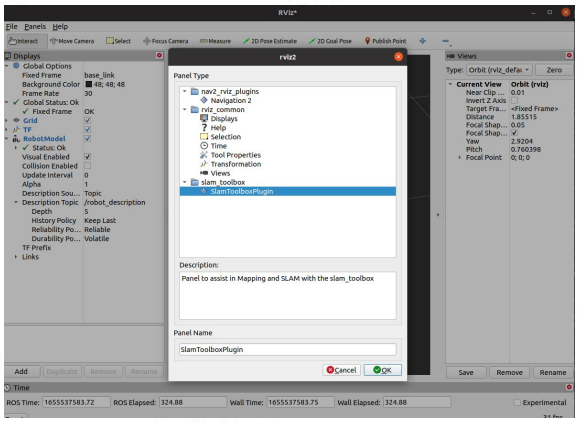

图 1-4-4　添加新的面板(2)

点击 ok，在 rviz2 中出现新的面板，界面如图所示：

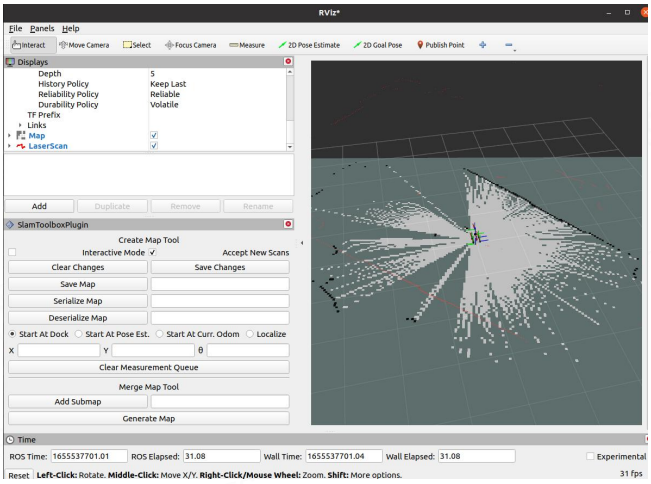

图 1-4-5　添加新的面板(3)

点击 Save Map 即可保存 slam_toolbox 建图，运行 slam_toolbox 建图的终端提示如下：

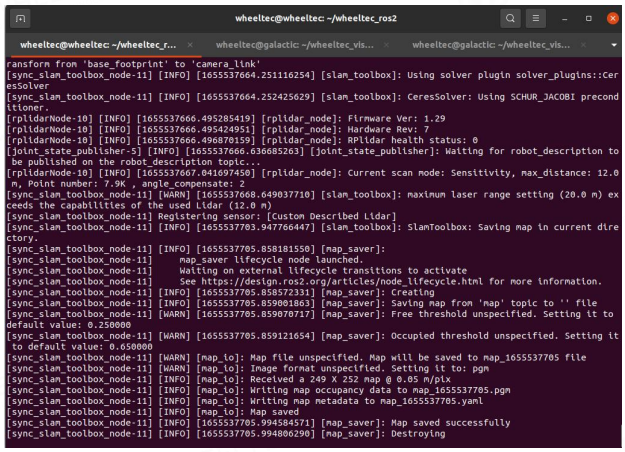

图 1-4-6　保存图片时终端输出

地图保存在运行 slam_toolbox 建图算法的终端所在目录中。

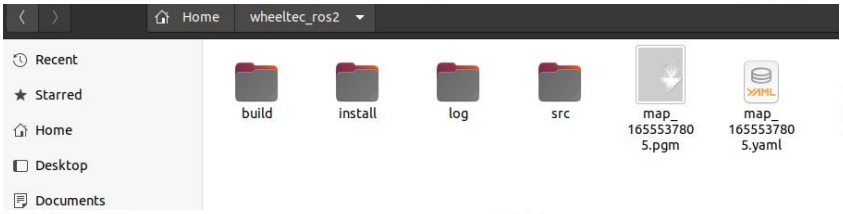

图 1-4-7　地图保存结果

④ cartographer 建图

Cartographer 是一个跨多个平台和传感器配置提供 2D 和 3D 实时同步定位和映射 (SLAM) 的系统。

Cartographer 建图时会发布 odom_combined->base_footprint 的转换，所以运

行 Cartographer 时需要将底层打开的卡尔曼滤波器的 tf 转换关掉。在文件中有相关参数：

```
/wheeltec_ros2/src/wheeltec_robot_slam/wheeltec_cartographer/launch/cartographer.launch.py
```

图 1-4-8　修改 cartographer.launch.py 文件

修改文件后进行编译：

```
cd ~/wheeltec_ros2
colcon build --packages-select turn_on_wheeltec_robot
```

打开 Cartographer 建图：

```
ros2 launch wheeltec_cartographer cartographer.launch.py
```

建图效果如图所示：

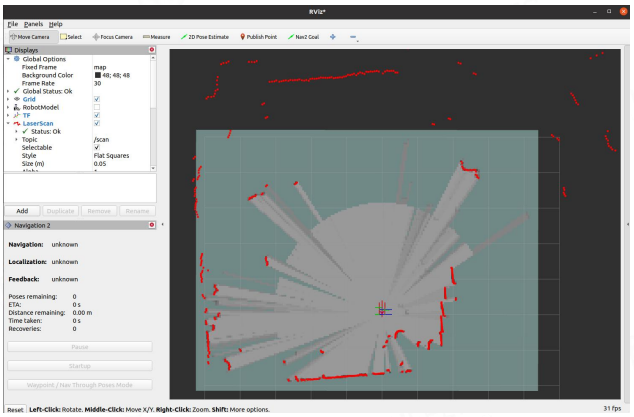

图 1-4-9　cartographer 建图

⑤ 保存地图

Map Server 用于处理堆栈的地图加载请求并托管地图主题的服务器，同时它也是一个地图保存服务器，根据服务请求保存地图。存在一个类似于 ROS 1 的地图保护程序 CLI，用于单个地图保存。

WHEELTEC ROS2 机器人保存地图指令为：

```
ros2 launch wheeltec_nav2 save_map.launch.py
```

或指定保存路径，map_dir 为目标路径，map_name 为保存的地图文件名：

```
ros2 run nav2_map_server map_saver_cli -f <map_dir>/<map_name> --ros-args -p save_map_timeout:=10000
```

运行 save_map.launch.py 文件调用 Map Server 服务，将地图保存在

~/wheeltec_ros2/src/wheeltec_robot_nav2/map 目录下。

建图完成后运行保存地图指令示例：

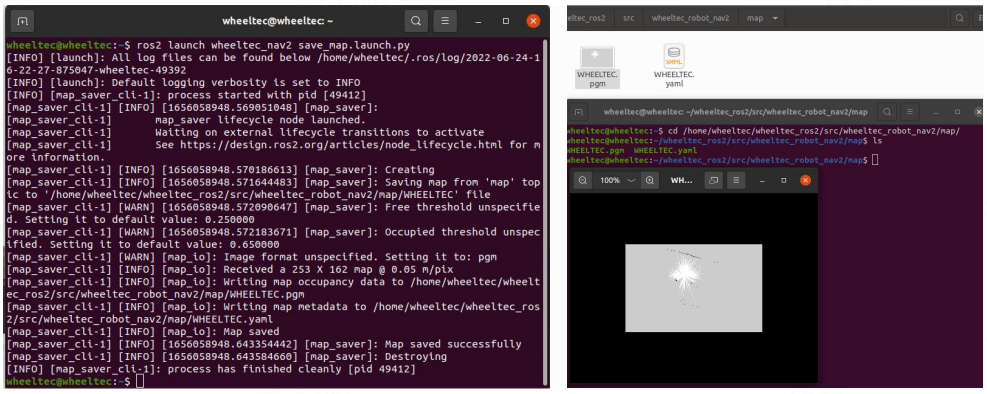

图 1-4-10　保存地图

此时可以在~/wheeltec_ros2/src/wheeltec_robot_nav2/map 目录下看到地图文件 WHEELTEC.pgm 和 WHEELTEC.yaml，双击文件可查看地图图像。

1.6 RRT 自主探索建图[RRT Exploration]

wheeltec_robot_rrt2 基于快速探索随机树 (RRT) 算法实现 WHEELTEC 机器人的自主探索建图功能。RRT（Rapidly-Exploring Random Trees）即快速随机扩展树，它是一种单一查询算法，一边抽样一边建图，通过不断随机探索来实现遍历。RRT 算法概率完备，可用于进行路径规划与地图边界探索。但因为其随机性比较强，所规划的路径并不一定是最优解。在 WHEELTEC ROS2 机器人中 RRT 算法通过将地图边界点作为目标点来实现小车自主导航探索，同时开启建图节点，实现自主建图。

在 WHEELTEC ROS2 机器人中只用到了 rrt_exploration 的全局 RRT 边界点检测器节点、本地 RRT 边界点检测器节点、过滤节点和分配者节点。

① 修改 navigation 导航参数

RRT Exploration 自主探索建图会一边导航一边建图，但不同的车型导航参数不一致，膨胀半径也不同，默认膨胀半径 inflation_radius 的值为 0.55m。用户根据自己的车型修改改参数，需要修改的文件为：

```
~/wheeltec_ros2/src/wheeltec_robot_rrt2/config/nav2_params.yaml
```

如 mini 系列小车修改 inflation_radius 值为 0.25， 文件修改后内容如下

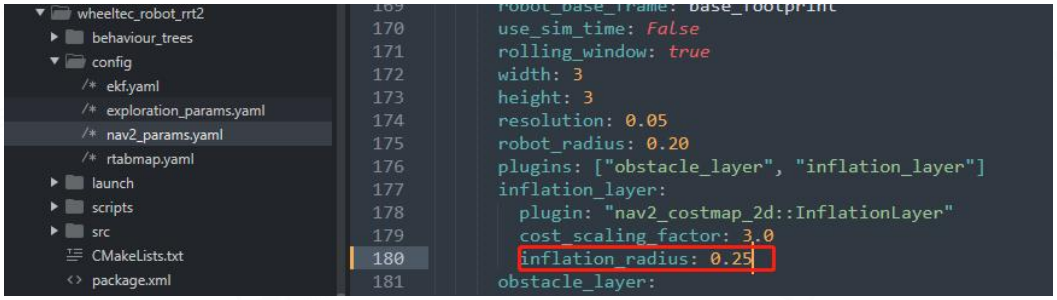

图 7-3-0　nav2_params.yaml 参数文件

```
cd ~/wheeltec_ros2
colcon build --packages-select wheeltec_robot_rrt
```

② 运行 RRT Exploration

在 ROS2 中运行 RRT Exploration 自主探索建图：

先运行 2D 建图：

```
ros2 launch wheeltec_slam_toolbox online_sync.launch.py
```

再打开 RRT 自主探索建图：

```
ros2 launch wheeltec_robot_rrt wheeltec_rrt_slam.launch.py
```

运行效果如图所示：

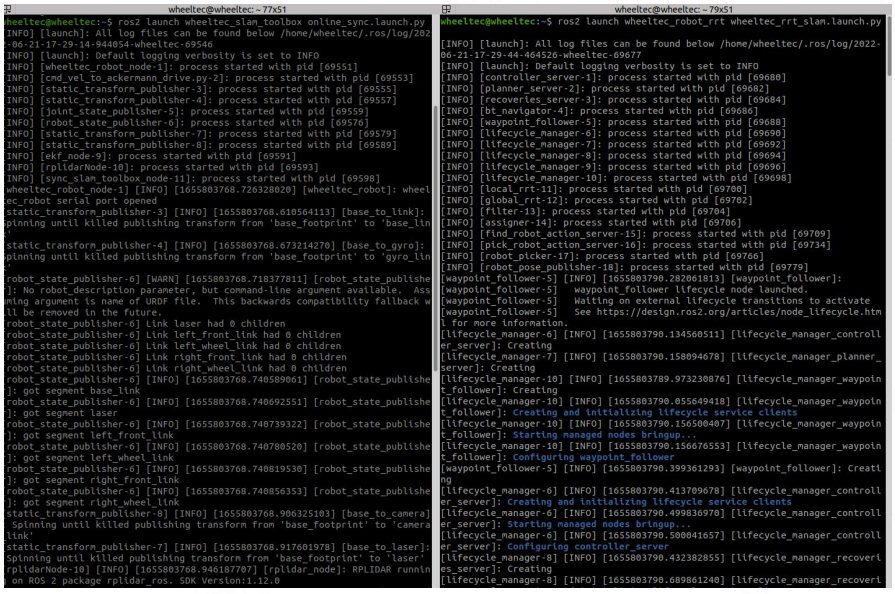

图 7-3-1　终端运行 rrt slam

期间终端会刷一段时间如下信息，这属于正常现象：

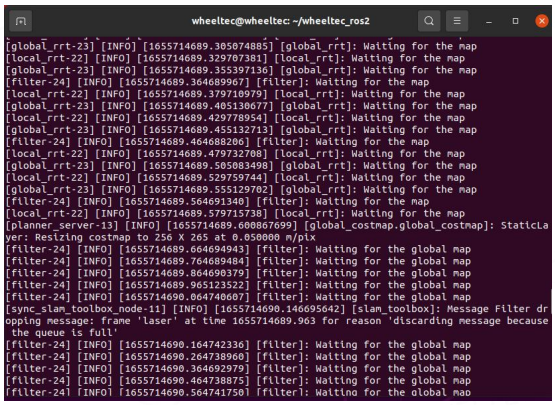

图 7-3-2　终端出现警报

最终终端输入蓝色字体提示后，打开 rviz2

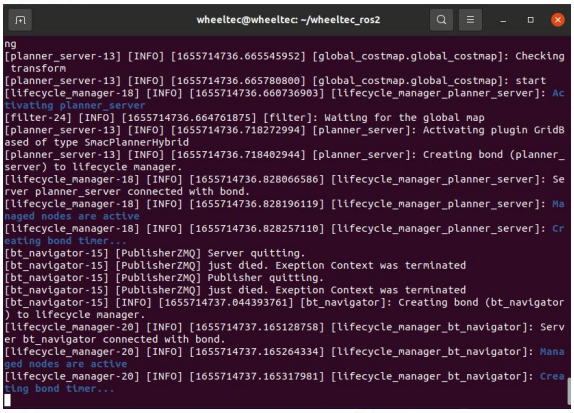

图 7-3-3 终端最终输出

在虚拟机端打开 RVIZ2：

```
ros2 launch wheeltec_rviz2 wheeltec_rviz.launch.py
```

注意：该 rviz 文件是环境中所配置好的，若使用其它 rviz 文件，请添加配置信息：

点击 rviz 界面左下角的 add，在弹窗中选择 By topic，添加图中框选的六个话题。

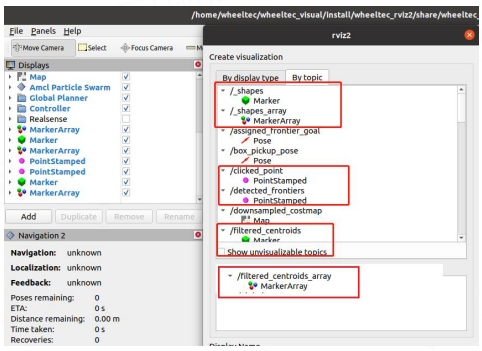

图 7-3-4　rviz2 配置

点击 Publish Point 按顺时针或逆时针的顺序发布建图区域的四个边缘点：

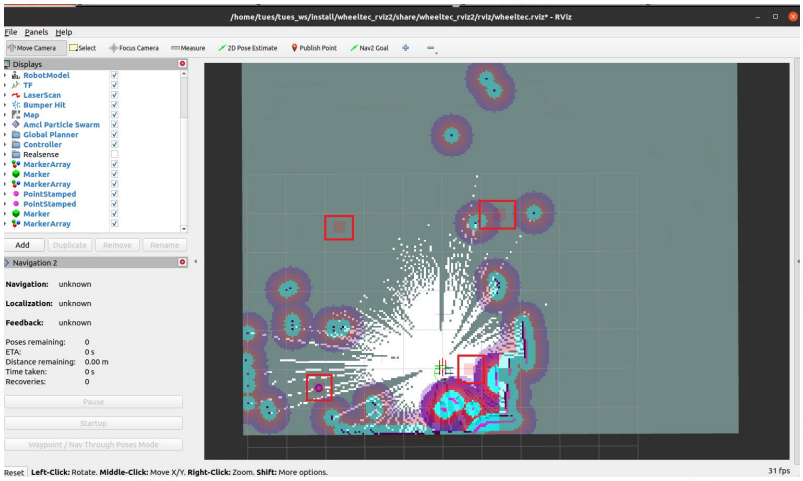

图 7-3-5　发布边缘点

顺时针发布四次Point后，用户还需要定下第五个点（最后一次PublishPoint），这个点是随机树的起点，需要发布在已知地图区域，一般选择定在小车车身上。当雷达扫描范围有部分选择屏蔽时，　需将起点定在距离小车较近处、雷达可扫描范围内。

注意：如果没有按照顺时针或逆时针顺序选取范围点，则所选范围点无法围成闭合图形，无法启动自主建图。

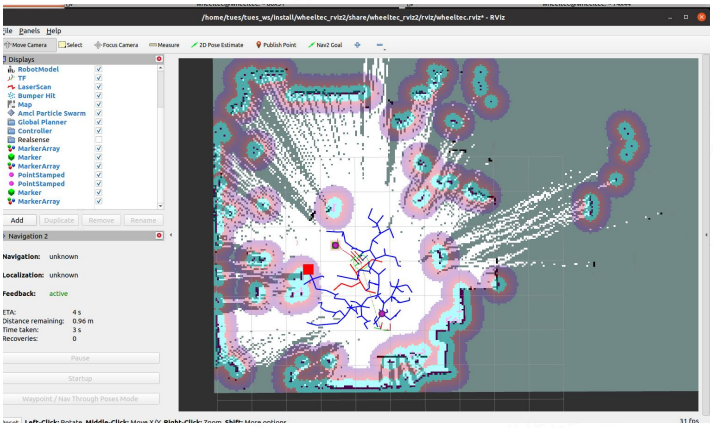

图 7-3-6　开始出现生长树

发布完最后一个点后，随机树开始生长，机器人开始进行导航并对地图边界进行探索。建图过程中蓝色树状为全局树。

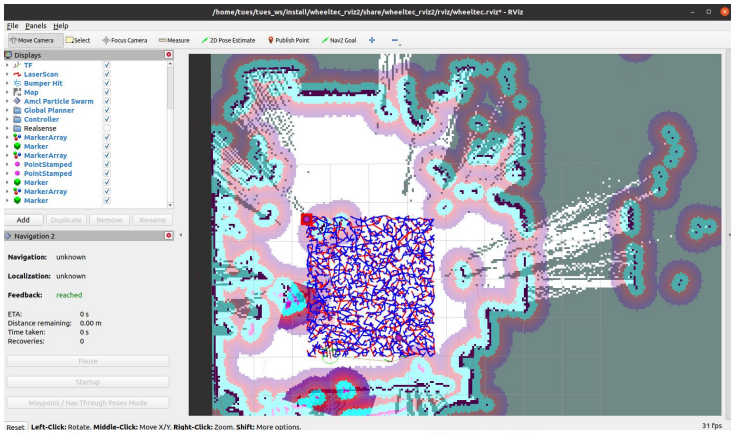

图 7-3-7　密集版生长树

完成探索后，小车将停止运动，此时开始进行计数，若 20 秒内小车都保持静止，则认为自主建图已经完成，启动保存地图节点。

运行 RRT Exploration 时查看 Tf Tree 与节点关系图为：

Tf Tree：

https://gitee.com/tuesdg/wheeltec-ros2-rrt-exploration/blob/master/frames.pdf

节点关系图：

https://gitee.com/tuesdg/wheeltec-ros2-rrt-exploration/raw/master/rosgraph.png

ROS2 中的 RRT Exploration 算法是根据 ROS1 移植的，其原理和 ROS1 一致。

1.7 2D 导航

① Nav2 概述

Nav2 项目是 ROS Navigation Stack 的精神继承者。该项目旨在找到一种让移动机器人从 A 点移动到 B 点的安全方法。它也可以应用于涉及机器人导航的其他应用，例如跟随动态点。这将完成动态路径规划、计算电机速度、避开障碍物和结构恢复行为。

Nav2 的预期输入是符合 REP-105 的 TF 转换，如果使用静态成本地图层，则为地图源，BT XML 文件和任何相关的传感器数据源。然后它将为完整或非完整机器人的电机提供有效的速度命令以跟随。

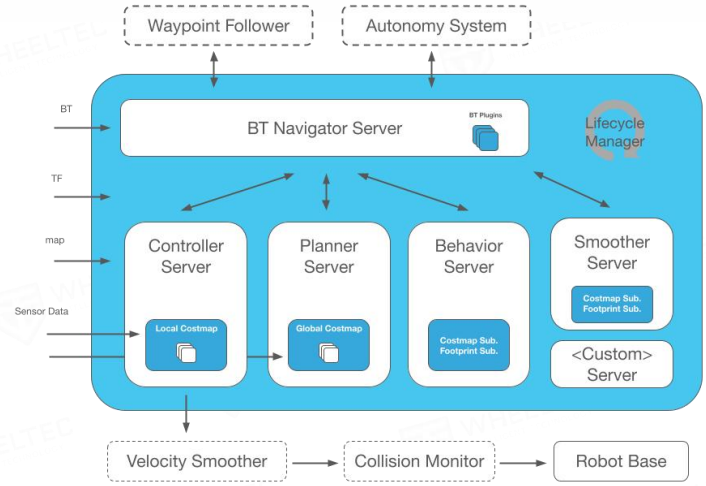

图 17-1　Nav2 结构框架图

Nav2 具有以下工具：

- 加载、服务和存储地图（地图服务器）
- 在地图上定位机器人 (AMCL)
- 围绕障碍物规划从 A 到 B 的路径（Nav2 规划器）
- 按照路径控制机器人（Nav2 控制器）
- 平滑路径计划更加连续和可行（Nav2 Smoother）
- 将传感器数据转换为世界的成本地图表示（Nav2 Costmap 2D）
- 使用行为树构建复杂的机器人行为（Nav2 行为树和 BT Navigator）
- 计算发生故障时的恢复行为（Nav2 Recoveries）
- 跟随顺序航点（Nav2 Waypoint Follower）
- 管理服务器的生命周期和看门狗 (Nav2 Lifecycle Manager)
- 启用用户的自定义算法和行为的插件 (Nav2 Core)

NavFn 使用 A* 或 Dijkstra 算法计算从姿势到目标姿势的最短路径。DWB 将使用 DWA 算法来计算遵循路径的控制结果，并使用其自己的几个插件用于轨迹修正。恢复行为包括：等待、旋转、清除成本图和备份。有一套 BT 插件可以调用这些服务器和计算条件。最后，还有一组 Rviz 插件，用于与堆栈交互并控制生命周期。

② ROS1 To ROS2

从 ROS1 到 ROS2，Navigation 的变化是巨大的。而其中最为显著的变化则是使用行为树来组合各个功能模块，以实现搭乐高积木的开发效果。

ROS1 中的导航主要由 move_base 来组合各个模块的功能。在 move_base 中实现了一个状态机，全局路径规划器和局部路径规划器都是以插件的形式在相应的状态中被调用的。P2P 导航的流程是：先在一个状态里调用全局路径规划器，然后进入到另外一个状态传路径给局部路径规划器，并使其执行该路径。

Nav2 不再是一个单一的整体状态机，而是利用动作服务器和 ROS 2 的低延迟、可靠的通信来分离想法。各个功能的组织则有所不同。全局路径规划功能由各个全局路径规划的插件实现，这些插件被加载到 Planer Server 中。可以通过相应的 action 向 Planer Server 请求全局规划的路径。此时 P2P 导航流程为：行为树上的相应节点会向 Planer Server 请求一条规划到目标点的路径。然后另一个树上的节点与 Controller Server 通信，将规划好的路径发给 Controller Server 去执行。

nav2_bt_navigator 在 move_base 顶层替换为 Action 接口，以使用基于树的操作模型完成导航任务。它使用行为树来实现更复杂的状态机并添加行为作为额外的动作服务器。这些行为树是可配置的 XML。

计划、行为和控制器服务器也是 BT 导航器可以调用来计算的操作服务器。所有 3 个服务器都可以托管许多算法的许多插件，每个插件都可以从导航行为树中单独调用以实现特定行为。提供的默认插件是从 ROS 1 移植而来的，即：DWB、NavFn 以及类似的行为，例如旋转和清除成本图。还添加了等待固定持续时间的新行为。这些服务器通过它们的操作服务器从 BT 导航器调用以计算结果或完成任务。状态由 BT 导航器行为树维护。

所有这些更改使得在启动/运行时用实现相同接口的任何其他算法替换任何这些节点成为可能。

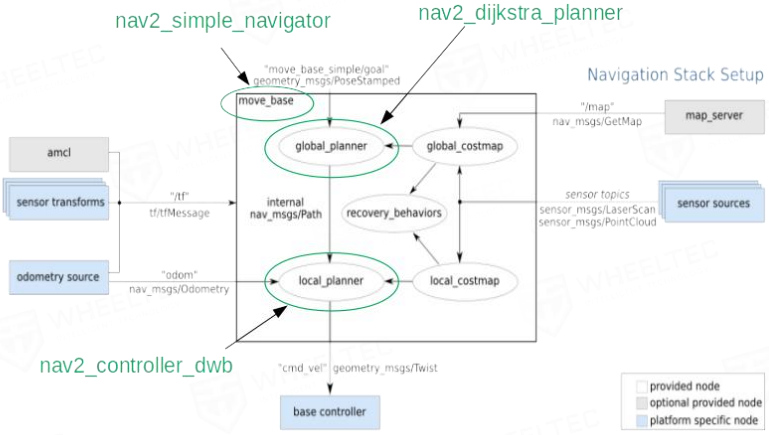

图 1-7-2　Nav2 结构框架图

移植功能包：

amcl：移植到 nav2_amcl

map_server：移植到 nav2_map_server

nav2_planner：替换 global_planner，托管 N 规划器插件

nav2_controller：替换 local_planner，托管 N 控制器插件

Navfn：移植到 nav2_navfn_planner

DWB：替换 DWA 并移植到 nav2_dwb_controller 元包下的 ROS 2

nav_core：移植为 nav2_core 并更新接口

costmap_2d：移植为 nav2_costmap_2d

新功能包：

nav2_bt_navigator：替换 move_base 状态机

nav2_lifecycle_manager：处理服务器程序生命周期

nav2_waypoint_follower：可以通过多个航点来执行复杂的任务

nav2_system_tests: 一组 CI 集成测试和模拟基础教程

nav2_rviz_plugins：一个 rviz 插件，用于控制 Navigation2 服务器、命令、取消和导航

nav2_experimental：深度强化学习控制器的实验（和不完整）工作

navigation2_behavior_trees：行为树库的包装器，用于调用 ROS 动作服务器

③ WHEELTEC NAV2 的路径规划器

WHLLTEC -Nav2 使用 Nav2 Planner 的服务器插件 SmacPlanner 作为规划师，使用 SmacPlannerHybrid 分支，使用 Reeds-shepp 运动模型的 SE2 Hybrid-A* 实现，具有更平滑和多分辨率的查询。支持差速、全向以及阿克曼运动模型。关于 SmacPlanner 的更多信息请查看：

https://github.com/ros-planning/navigation2/tree/main/nav2_smac_planner

局部轨迹规划器使用 RegulatedPurePursuitController 算法计算导航路径，它实现了纯追踪算法的变体来跟踪路径。这种变体我们称之为受监管的纯追踪算法，因为它对碰撞和线速度有额外的监管条款。它还实现了 Adaptive Pure Pursuit 算

法背后的基础知识，以根据当前速度改变前瞻距离。这个插件实现了 nav2_core::Controller 接口，允许它在导航堆栈中用作控制器服务器的动作服务器（controller_server）中的本地轨迹规划器。

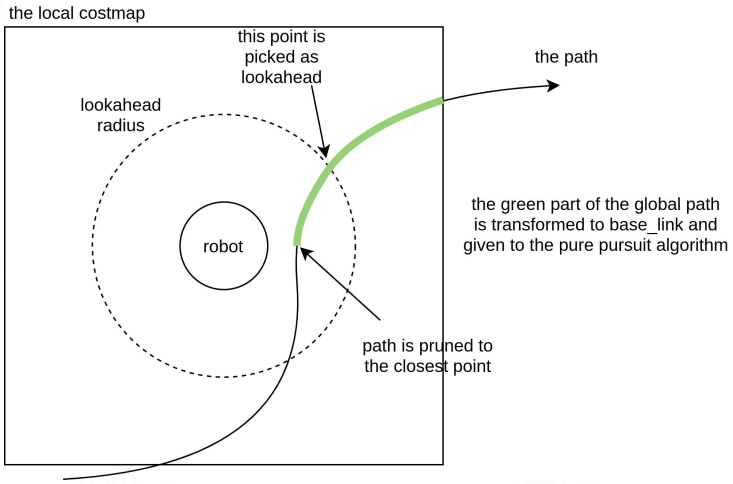

图 1-7-3　本地规划器

该控制器适用于所有类型的机器人，包括差速器、腿式和阿克曼转向车辆。它也可以用于全向平台，但不能充分利用底座的横向运动。如果需要用到横向运动，可以使用 DWB 算法，目前镜像默认使用 RegulatedPurePursuitController 做局部轨迹规划器。

④ WHEELTEC NAV2 功能调试

wheeltec_nav2 功能包的主要作用是在机器人端实现 2D 导航与其它拓展性功能。

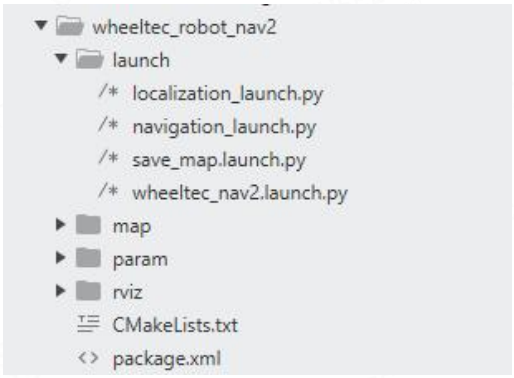

图 1-7-4　wheeltec_nav2 内容

localization_launch.py 是启动机器人在地图中的定位，使用 amcl 蒙特卡洛定位，以及打开 map_server 节点加载地图；

navigation_launch.py 启动导航的一些插件，包括规划器、恢复器和行为树；

在 wheeltec_nav2.launch.py 文件中，除了调用 navigation_launch.py 和 localization_launch.py 文件，还打开了初始化底层控制节点和雷达启动节点，将导航的地图传入 localization_launch.py 中，Nav2 的插件参数和行为树的相关参数传入 navigation_launch.py 文件。在 wheeltec_nav2.launch.py 文件中，如果设置 autostart：= False，则需要单击 RViz 中的开始按钮以初始化节点。同时需要确保 use_sim time 设置为 False，因为我们要使用系统时间而不是 Gazebo 中的模拟时间。

map 文件夹存放导航所需要的地图，所以导航之前需要先构建地图。

param 文件夹存放 Nav2 的插件参数和行为树的 yaml 文件，子文件夹 wheeltec_param 存放 WHLLTEC 不同车型的插件参数。

1) 点到点导航：

在 WHEELTEC 机器人中运行 Navigation2：

```
ros2 launch wheeltec_nav2 wheeltec_nav2.launch.py
```

虚拟机/Ubuntu 电脑中打开 rviz2:

```
ros2 launch wheeltec_rviz2 wheeltec_rviz.launch.py
```

或直接在终端打开 rviz 后，选择提供的 rviz 配置文件。路径为 5.ROS 源码/rviz 配置文件。打开 rviz 后，点击 file->打开 rviz 文件->选择 rviz 配置文件/wheeltec_rviz.rviz 即可。

打开 RVIZ2 后发布导航点如图所示：

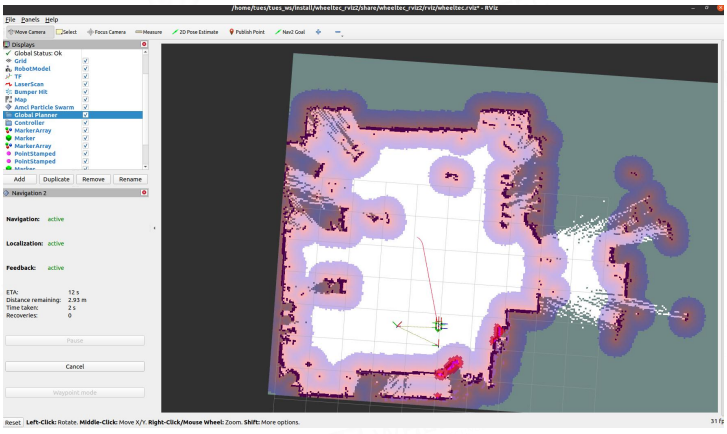

图 1-7-5 运行导航时的 rviz2

在 Nav2 RViz 的左下角方向，用户可以直接通过 RVIZ2 面板找到预计到达时间、距离目标的剩余距离、导航开始后经过的时间以及导航操作期间执行的恢

复次数等信息。

2) 路标跟随演示

路标跟踪是导航系统的基本功能。它告诉我们的系统如何使用导航到达多个目的地。在 2D 单点导航的步骤下，打开 rviz2 后，点击左下角的”start waypoint following”切换成路标跟随模式，如图 3-2-7 所示。

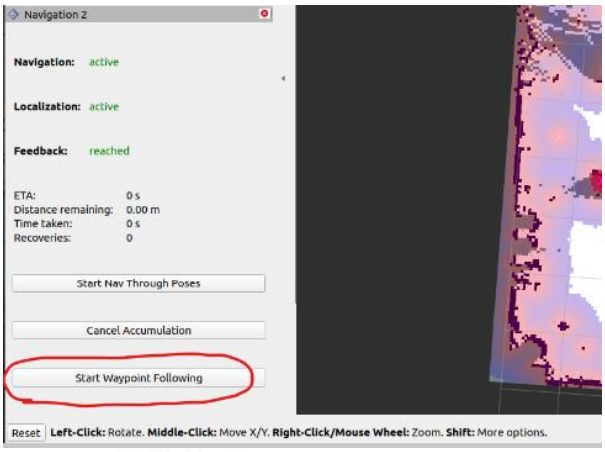

图 1-7-6 waypoint mode 设置页面

到这一步，再点击 rviz2 上方的”Nav2 Goal”开始设定路标位置

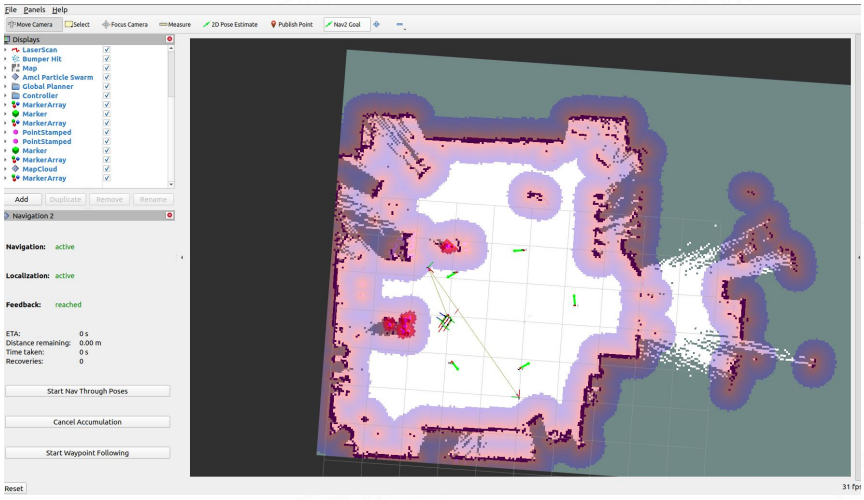

图 1-7-7 waypoint mode 设置路标

标完位置后，点击右下角的”Start Nav Through Poses ”开始实现路标跟随。途中可以点击 Cancel 取消前往目标点。

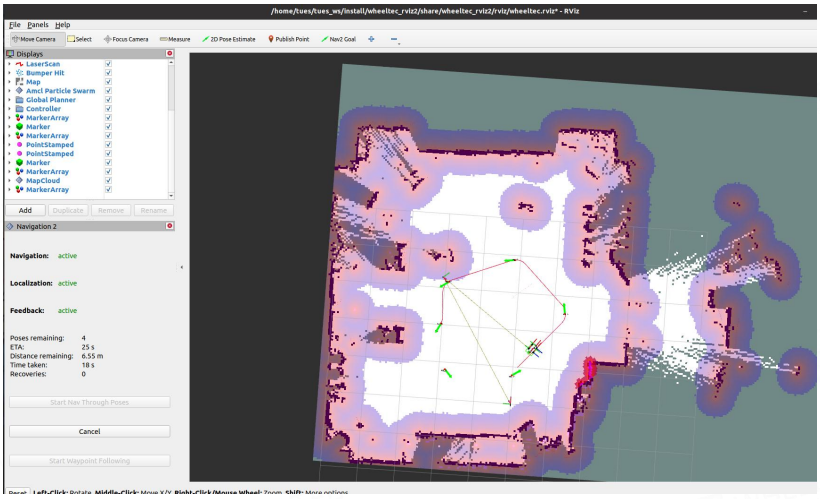

图 1-7-8　waypoint mode 开始导航

1.8 WEB 浏览器显示摄像头图像

wheeltec_web_video 是一个提供网络视频服务器的功能包，它可以实现通过网络进入网页地址查看图像流。

web_video_server 打开一个本地端口并等待传入的 HTTP 请求。一旦通过 HTTP 请求 ROS 图像主题的视频流，它就会订阅相应的主题并创建视频编码器的实例。编码的原始视频数据包提供给客户端。可以通过将附加参数添加到查询字符串来指定参数。

运行网页视频实时监控功能：

Step1：打开相机设备

打开相机获取图像信息

```
ros2 launch usb_cam usb_cam_launch.py
```

Step2：打开 web_video_server 节点，创建等待 HTTP 请求的服务端。

```
ros2 run web_video_server web_video_server
```

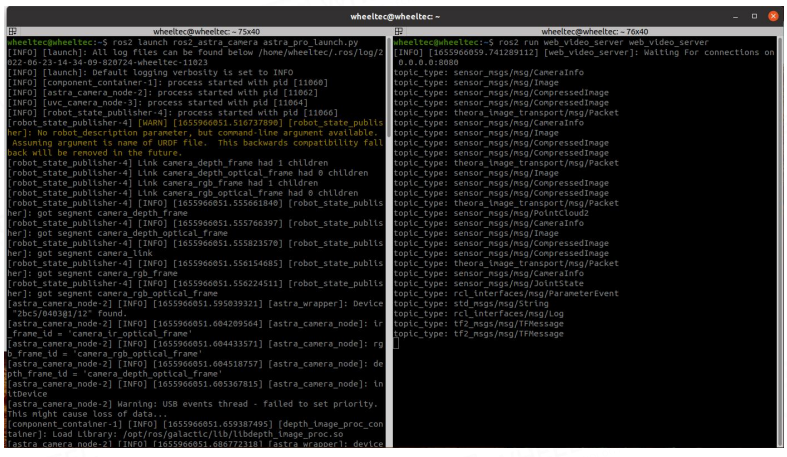

图 1-8-1　打开 web_video_server 节点

Step3：打开客户端（虚拟机）的浏览器并在网页上输入地址 http://192.168.0.100:8080 订阅服务端的图像信息，同时可以通过“ros2 topic list”查看目前已发布的话题列表。

```
主机网页查看：http://localhost:8080/
客户机网页查看：http://192.168.0.100:8080
```

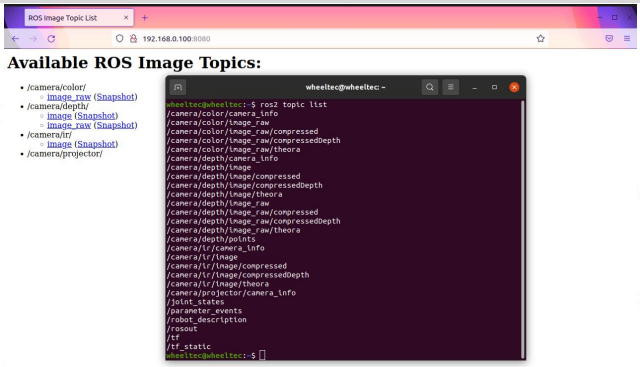

图 1-8-2　在火狐浏览器查看摄像头图像

Step4：在浏览器中点击话题查看相机的实时输出，可以看到 rgb 图的图像流，也可以打开 rqt 可视化工具查看相机输出。

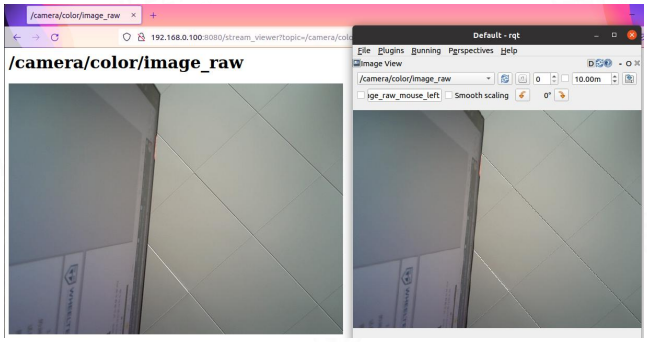

图 1-8-3　查看 RGB 图

注意：建议使用谷歌浏览器和火狐浏览器进行图像查看。

1.9 键盘控制

wheeltec_robot_keyboard 实现使用键盘控制操控机器人的功能。运行键盘控制节点时，用户可以通过按下键盘按键发布/cmd_vel 的速度话题。

一般在 WHEELTEC ROS2 机器人打开键盘控制节点前需要先开启底盘控制，使用键盘控制指令为：

```
#开启底盘控制
ros2 launch turn_on_wheeltec_robot turn_on_wheeltec_robot.launch.py
```

```
#开启键盘控制
ros2 run wheeltec_robot_keyboard wheeltec_keyboard
```

终端输出如下：

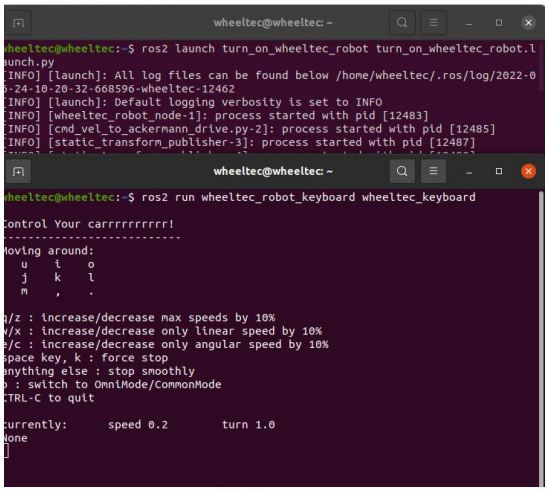

图 1-9-1　终端输出详情

根据终端输出提示，键入[i] [,][j][l]分别代表控制机器人前进后退往左往右，[k]键发布 0 的速度指令使机器人停止运动，符合阿克曼运动模型的机器人无法直接往左或往右运动。

1.10　雷达角度屏蔽

WHEELTEC ROS2 小车除了支持常用的思岚雷达之外，还支持镭神的 M10、N10(网口和串口版)雷达，和乐动雷达 LD14、LD06、LD19 雷达。默认使用思岚的 A1 雷达。若用户使用其它款式雷达，需要在此文件夹下进行更换：

```
wheeltec_ros2/src/turn_on_wheeltec_robot/launch/wheeltec_lidar.launch.py
```

在文件最后一行修改参数，可选参数为：Lsm10P、Lsm10、Lsn10P、Lsn10、Ld14、Ld06。

若使用镭神 LSN10 雷达，则修改参数后文件内容如下：

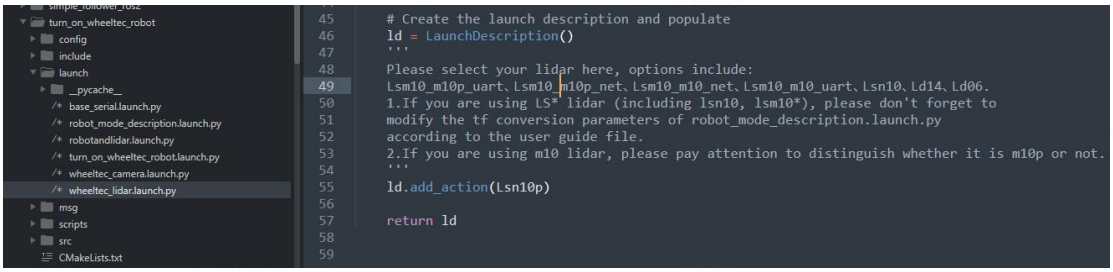

图 1-10-1　修改 wheeltec_lidar.launch.py 文件

修改文件后进行编译：

```
cd ~/wheeltec_ros2
colcon build --packages-select turn_on_wheeltec_robot
```

① 使用镭神雷达注意事项

使用 LSm10、LSn10 除了要修改 wheeltec_lidar.launch.py 文件之外，还需要修改 TF 静态转换的参数，因为 LSm10*和 LSn10*雷达扫描的方向与常用雷达相反，所以需要在 TF 静态转换中把方向调转。

修改文件：

```
wheeltec_ros2/src/turn_on_wheeltec_robot/launch/
robot_mode_description.launch.py
```

选择对应的车型 GroupAction，修改 base_to_laser 节点的 x 方向参数，以 mini_akm 小车为例，则在此处将 3.14 改为 0：

图 1-10-2　修改 robot_mode_description.launch.py 文件

修改文件后进行编译：

```
cd ~/wheeltec_ros2
colcon build --packages-select turn_on_wheeltec_robot
```

② 雷达角度屏蔽

不同雷达角度屏蔽修改的文件不同。WHEELTEC ROS2 中雷达文件均在~/wheeltec_ros2/src/wheeltec_lidar_ros2 目录下。在该目录下，LDlidar 表示乐动雷达，LSlidar 表示镭神雷达。

乐动雷达角度屏蔽只需要修改雷达启动文件即可，乐动 LD06 雷达修改角度屏蔽步骤，修改文件

```
~/wheeltec_ros2/src/wheeltec_lidar_ros2/LDlidar/ldlidar06/launch/ld06.launch.py
```

如要屏蔽乐动 LD06 雷达扫描的角度为[135，225]这一区间，文件内容如图所示。

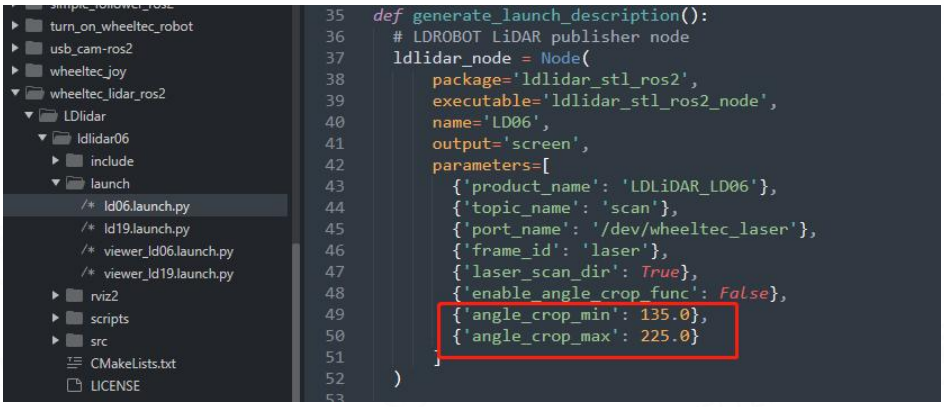

图 1-10-3　修改 ld06.launch.py 文件

文件修改后需要编译

```
cd ~/wheeltec_ros2
colcon build --packages-select ldlidar_stl_ros2
```

乐动 LD14 雷达修改角度屏蔽步骤，修改文件

```
~/wheeltec_ros2/src/wheeltec_lidar_ros2/LDlidar/ldlidar14/launch/ld14.launch.py
```

如要屏蔽乐动 LD14 雷达扫描的角度为[135，225]这一区间，文件内容如图所示。

图 1-10-4　修改 ld14.launch.py 文件

文件修改后需要编译

```
cd ~/wheeltec_ros2
colcon build --packages-select ldlidar_sl_ros2
```

使用镭神雷达进行角度屏蔽，需要修改雷达对应的参数文件。镭神雷达包括 M10（串口版和网口版）、M10P（串口版和网口版）、N10 这几款雷达。雷达文件结构如图所示。

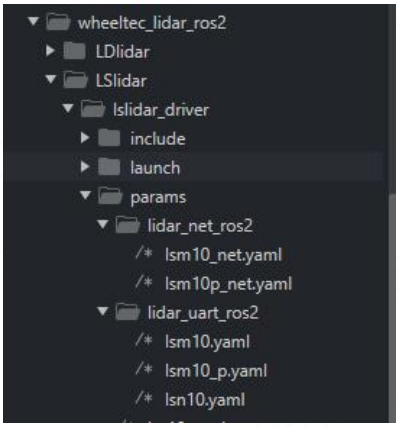

图 1-10-5　LSlidar 文件框架

不同雷达使用的参数文件不同，如 LSN10 雷达修改屏蔽角度，需要修改文件

```
~/wheeltec_ros2/src/wheeltec_lidar_ros2/LSlidar/lslidar_driver/params/lidar_uart_ros2/lsn10.yaml
```

如要屏蔽镭神 LSN10 雷达扫描的角度为[135，225]这一区间，文件内容如图所示。

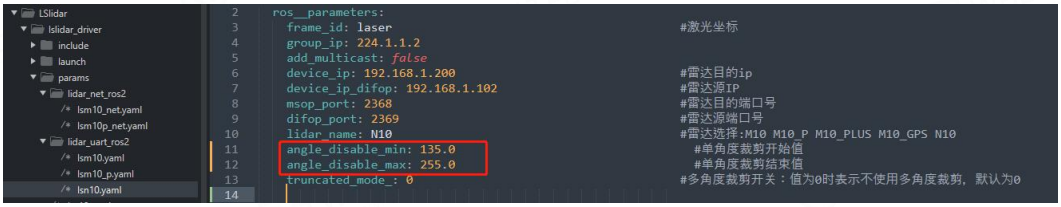

图 1-10-6　lsn10.yaml 文件内容

如 LSM10 串口版雷达修改屏蔽角度，需要修改文件

```
~/wheeltec_ros2/src/wheeltec_lidar_ros2/LSlidar/lslidar_driver/params/lidar_uart_ros2/lsm10.yaml
```

如要屏蔽镭神 LSM10 串口版雷达扫描的角度为[135，225]这一区间，文件内容如图所示。

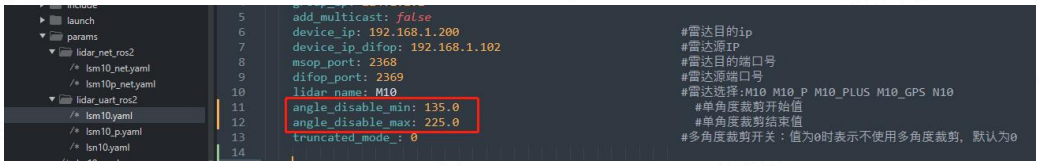

图 1-10-7　lsm10.yaml 文件内容

如 LSM10P 网口版雷达修改屏蔽角度，需要修改文件

```
~/wheeltec_ros2/src/wheeltec_lidar_ros2/LSlidar/lslidar_driver/params/lidar_net_ros2/lsm10p_net.yaml
```

如要屏蔽镭神 LSM10P 网口版雷达扫描的角度为[135，225]这一区间，文件内容如图所示。

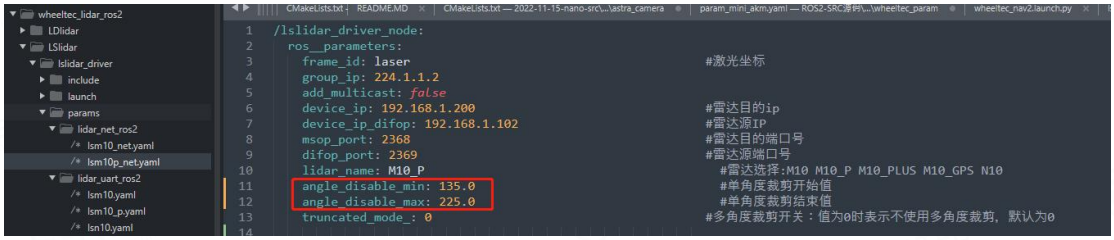

图 1-10-8　lsm10p_net.yaml 文件内容

文件修改后需要编译

```
cd ~/wheeltec_ros2
colcon build --packages-select lslidar_driver
```

2. ROS 与 TogetherROS 环境切换

当用户拿到 R550 小车时，系统中已经搭建好了 ROS2-Humble 与 TogetherROS 开发框架，默认使用 ROS2 Humble 的开发环境。两个框架之间同时使用的时候会发生通信问题。ROS2-Humble 与 TogetherROS 的工作空间是区分开来的，ROS2-Humble 源码在 wheeltec_ros2 文件夹中，TogetherROS 控制源码在 wheeltec_tros 文件夹中。

注意：运行 ROS2-Humble 和 TogetherROS 功能时，python 的版本需要进行切换。ROS2-Humble 的 python 版本需要切换为 python3.10，TogetherROS 的 python 版本需要切换为 python3.8。

2.1 远程登录

WHEELTEC R550 小车可以使用 ssh 进行远程登录调试，具体步骤为如下。连接 X3 小车发出的 wifi，wifi 名为”wheeltec_sunrise”，密码为”dongguan”，连接成功后即可进入虚拟机用 ssh 远程登录到小车上。

图 2-1-1　wifi 连接

终端输入指令远程登录，ssh 登录的密码也是”dongguan”：

ssh -Y wheeltec@192.168.0.100

2.2 切换 ROS2 foxy 系统

在 WHEELTEC R550 小车上调试 ROS2-Humble 功能，需要先确认目前终端处于 Humble 开发框架，(默认使用的环境为 Humble 开发框架)具体步骤为：

修改~/.bashrc 文件，将以下指令屏蔽即可，终端输入 nano ~/.bashrc 进行修改，修改完成后文件如图所示。

```
#source /opt/tros/local_setup.bash
```

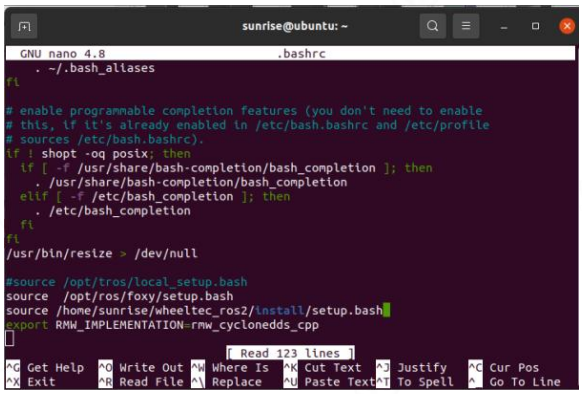

图 2-2-1 修改~/.bashrc 文件

使用 Ctrl+o 保存文件，Ctrl+x 退出编辑，保存修改后运行指令使修改生效。

```
source ~/.bashrc
#将 python 版本设置为 3.10
rm /usr/bin/python3
ln -s /usr/bin/python3.10 /usr/bin/python3
```

2.3 切换 TogetherROS 系统

当用户需要使用 TogetherROS 系统时，将~/.bashrc 文件中的以下内容屏蔽即可，终端输入 nano ~/.bashrc 进行修改，修改完成后文件如图所示。

```
#source /opt/ros/humble/setup.bash
#source /home/wheeltec/wheeltec_ros2/install/setup.bash
#将 python 版本设置为 3.8
rm /usr/bin/python3
ln -s /usr/bin/python3.8 /usr/bin/python3
```

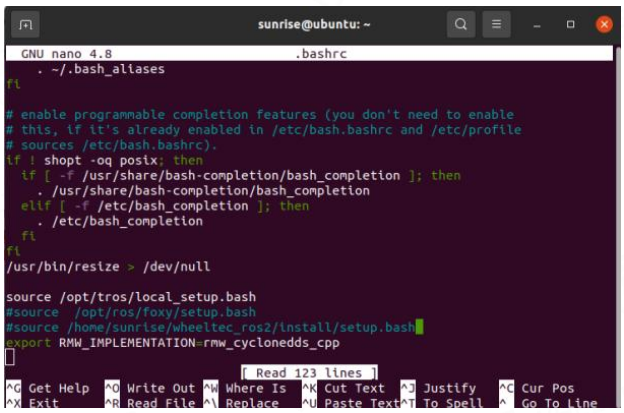

图 2-3-1 修改~/.bashrc 文件

注意：TogetherROS 的例程调用的参数文件存放在根目录中，需要在 root 权限下运行例程。

3. TogetherROS AI 功能调试

TogetherROS 是地平线面向机器人厂商和生态开发者推出的机器人操作系统，旨在释放 AI 在机器人场景的潜能,助力生态开发者和商业客户能够高效、便捷的进行机器人开发，打造具有竞争力的智能机器人产品。

WHEELTEC R550 小车使用 usb-rgb 相机运行 TogetherROS AI 程序，调试 AI 功能前需要确认目前终端处于 TogetherROS 开发环境，并拥有 root 权限：

使用 ssh 远程登录之后，运行指令切换为 root 用户，sudo 密码是”dongguan”：

```
sudo su
```

3.1 手势识别

① 手势识别功能介绍

手势识别算法示例订阅包含人手框、人手关键点信息的 AI msg，利用 BPU 处理器进行 AI 推理，发布包含手势信息的 AI msg。

手势识别 package 订阅人手关键点检测 package 发布的人手关键点检测结果，经过算法推理后发布 hobot AI msg，通过 websocket package 实现在 PC 端浏览器上渲染显示 sensor 发布的图片和对应的 AI 结果。

关于手势识别功能的更多信息请查阅地平线官方手册：

https://developer.horizon.ai/api/v1/fileData/TogetherROS/box/box_adv/gesture_detection.html

② 在 WHEELTEC R550 小车上运行手势识别

WHEELTEC R550 小车运行手势识别步骤为：远程登录到 WHEELTEC R550 小车的 root 用户，密码为“dongguan”确认此时终端为 TogetherROS 环境：

```
ssh -Y root@192.168.0.100
source /opt/tros/local_setup.bash
```

配置 USB 摄像头

```
export CAM_TYPE=usb
#将 python 版本设置为 3.8
rm /usr/bin/python3
ln -s /usr/bin/python3.8 /usr/bin/python3
```

启动手势控制：

```
ros2 launch hand_gesture_detection hobot_hand_gesture_detection.launch.py
```

在虚拟机端/Ubuntu 电脑中打开浏览器，输入 http://192.168.0.100:8080，查看摄像头实时数据，如图所示：

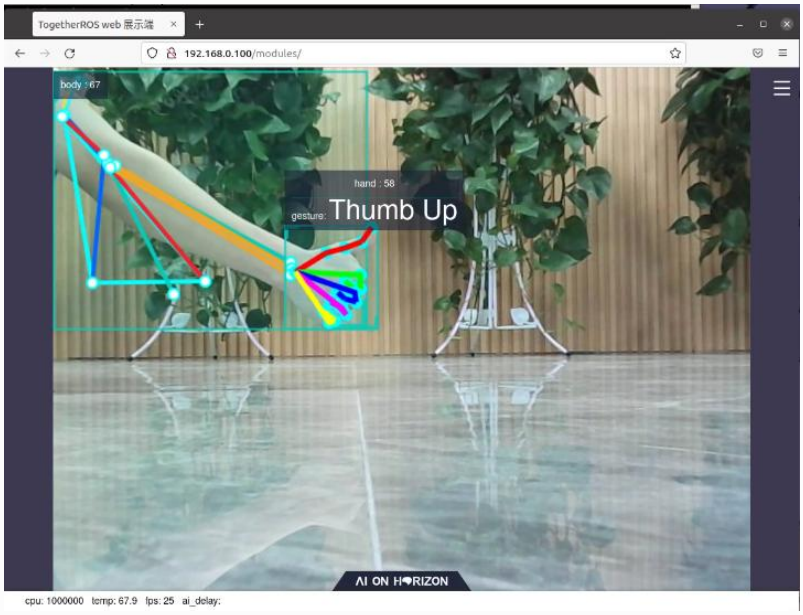

图 3-1-1　WHEELTEC R550 小车运行手势识别

3.2 手势控制

① 手势控制功能介绍

小车手势控制 APP 功能为通过手势控制机器人小车运动，包括左右旋转和前后平移运动。APP 由 MIPI 图像采集、人体检测和跟踪、人手关键点检测、手势识别、手势控制策略、图像编解码、WEB 展示端组成，流程如下图：

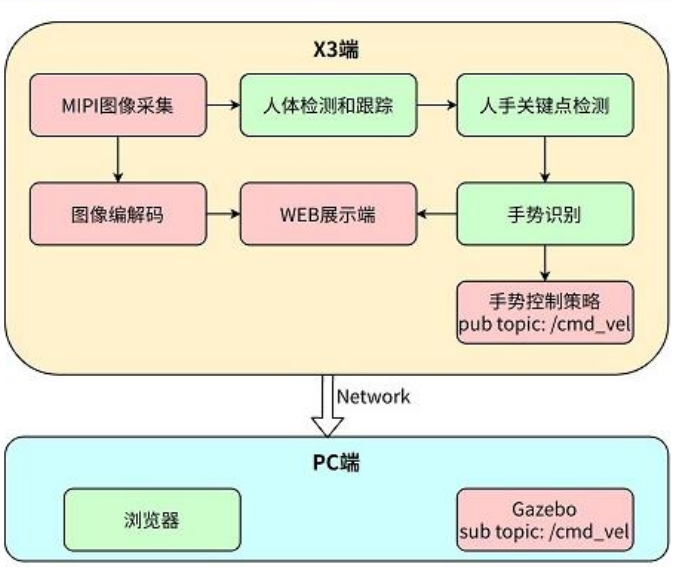

图 3-1-1　手势识别流程

关于手势控制功能的更多信息请查阅地平线官方手册：

https://developer.horizon.ai/api/v1/fileData/TogetherROS/app/car_gesture_control.html

若要更换控制手势，则需要下载手势控制的源码，在工作空间下修改编译再调试。代码仓库：

https://c-gitlab.horizon.ai/HHP/box/hobot_interactions/gesture_control

② 在 WHEELTEC R550 小车上运行手势控制

WHEELTEC R550 小车运行手势控制步骤为：远程登录到 WHEELTEC R550 小车的 root 用户，密码为“dongguan”，确认此时终端为 TogetherROS 环境：

```
ssh -Y root@192.168.0.100
source /opt/tros/local_setup.bash
```

配置 USB 摄像头

```
export CAM_TYPE=usb
#将 python 版本设置为 3.8
rm /usr/bin/python3
ln -s /usr/bin/python3.8 /usr/bin/python3
```

启动手势控制：

```
ros2 launch gesture_control hobot_gesture_control.launch.py
```

打开第二个终端，启动机器人底盘控制：

```
ssh -Y root@192.168.0.100
```

```
#将 python 版本设置为 3.10
rm /usr/bin/python3
ln -s /usr/bin/python3.10 /usr/bin/python3

source /opt/ros/humble/setup.bash
source /home/wheeltec/wheeltec_ros2/install/setup.bash
ros2 launch turn_on_wheeltec_robot turn_on_wheeltec_robot.launch.py
```

运行以上代码后，通过“666 手势/Awesome”手势控制小车前进，“yeah/Victory”手势控制小车后退，“大拇指向右/ThumbRight”手势控制小车右转，“大拇指向左/ThumbLeft”手势控制小车左转。其中左转/右转分别是向人的左/右方向（大拇指的指向）转动。

图 3-1-2　控制时用到的手势

在虚拟机端/Ubuntu 电脑中打开浏览器，输入 http://192.168.0.100:8080，查看摄像头实时数据，如图所示：

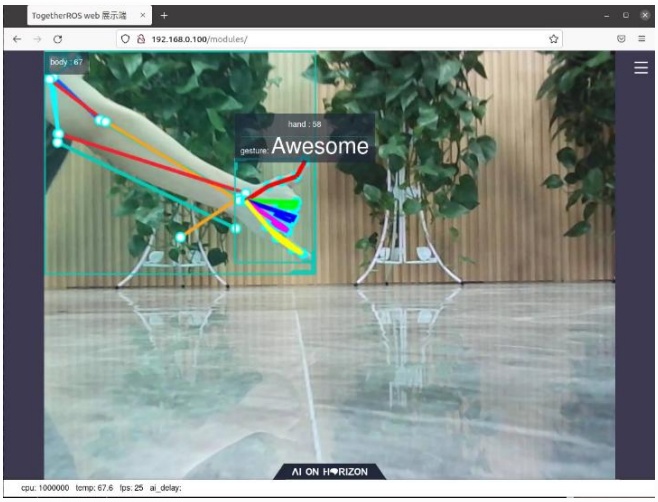

图 3-2-3　WHEELTEC R550 小车运行手势控制

3.3 物体识别

① 物体识别功能介绍

开发板上安装了 FCOS 算法用于测试 USB 摄像头的数据通路，该测试程序会实时读取 USB 摄像头的图像数据，然后运行视觉检测算法，最后把视频和算法结果输出到荧幕中。

FCOS 目标检测算法示例使用图片作为输入，利用 BPU 进行算法推理，发布包含目标类别和检测框的智能 msg。FCOS 是地平线开源的 Onnx 模型，使用 COCO 数据集进行训练，支持的目标检测类型包括人、动物、水果、交通工具等共 80 种类型。代码仓库：https://github.com/HorizonRDK/hobot_dnn

快速运行物体识别功能的详细操作请查阅地平线官方手册：https://developer.horizon.cc/documents_tros/boxs/detection/fcos

图 3-3-1　地平线官方手册

② 在 WHEELTEC R550 小车上运行物体识别

WHEELTEC R550 小车运行手势控制步骤为：远程登录到 WHEELTEC R550 小车的 root 用户，密码为“dongguan”，确认此时终端为 TogetherROS 环境：

```
    ssh -Y root@192.168.0.100
    source /opt/tros/setup.bash
    export CAM_TYPE=usb
    #将 python 版本设置为 3.8
    rm /usr/bin/python3
    ln -s /usr/bin/python3.8 /usr/bin/python3

    ros2 launch dnn_node_example dnn_node_example.launch.py\
dnn_example_config_file:=config/fcosworkconfig.json\
dnn_example_image_width:=480 dnn_example_image_height:=272
```

远程登录时物体识别的图像弹窗无法弹出，此时可以在终端中查看到识别的结果：

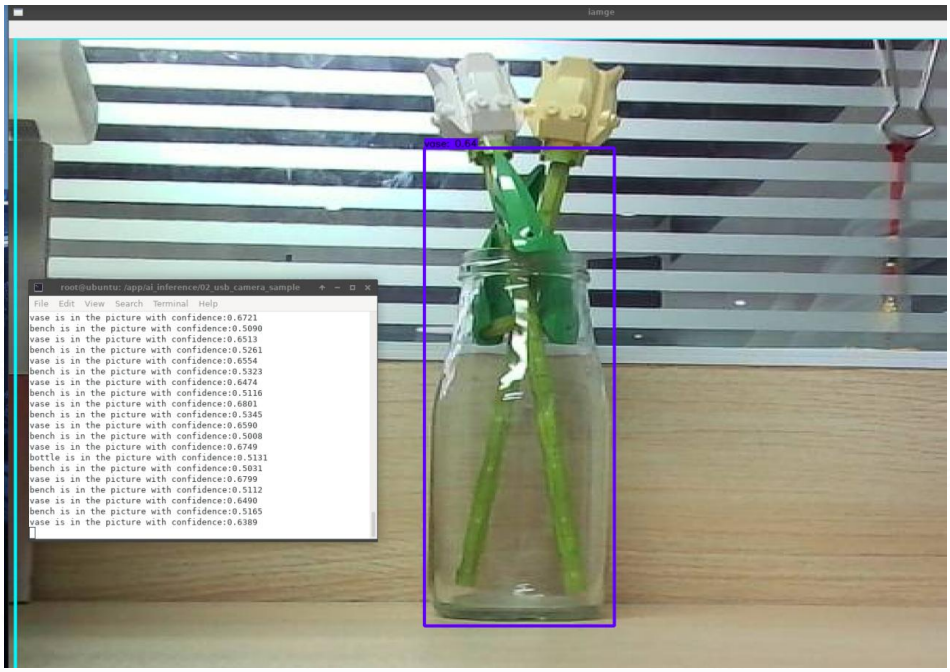

图 3-3-3　WHEELTEC R550 小车运行物体识别

3.4骨架识别

使用地平线 mono2d_body_detection package 功能包可以实现人体骨架识别效果，mono2d_body_detection package 功能包是使用 hobot_dnn package 开发的单目 rgb 人体检测算法示例，在地平线 X3 开发板上使用模型和图像数据利用 BPU 处理器进行模型推理。 检测模型为 fasterRcnn，模型输出包含人体、人头、人脸、人手框和人体关键点检测结果。可以通过以下二进制指令安装：

```
sudo apt update
sudo apt install -y tros-mono2d-body-detection
```

代码仓库：

https://c-gitlab.horizon.ai/HHP/box/hobot_perception/mono2d_body_detection

3.5骨架识别控制

骨架识别控制是轮趣科技使用 RGB 相机基于 mono2d_body_detection package 功能包开发的功能，实现效果为：当识别人体处于相机视野范围内，RDK X3 小车识别人体抬起左脚时，小车后退；当识别人体处于相机视野范围内，RDK X3 小车识别人体抬起右脚时，小车前进。

① 在 WHEELTEC R550 小车上运行骨架识别

WHEELTEC R550 小车运行骨架控制步骤为：远程登录到 WHEELTEC R550 小车的 root 用户，密码为“dongguan”，确认此时终端为 TogetherROS 环境：

```
ssh -Y root@192.168.0.100
source /opt/tros/local_setup.bash
#配置 USB 摄像头
export CAM_TYPE=usb
#启动骨架识别功能(tros):
cp -r /opt/tros/lib/mono2d_body_detection/config/ .
#将 python 版本设置为 3.8
rm /usr/bin/python3
ln -s /usr/bin/python3.8 /usr/bin/python3
ros2 launch mono2d_body_detection mono2d_body_detection.launch.py
```

打开第二个终端(foxy)，启动机器人底盘控制：

```
ssh -Y root@192.168.0.100
#将 python 版本设置为 3.10
rm /usr/bin/python3
ln -s /usr/bin/python3.10 /usr/bin/python3
source /opt/ros/humble/setup.bash
source /home/wheeltec/wheeltec_ros2/install/setup.bash
ros2 launch turn_on_wheeltec_robot turn_on_wheeltec_robot.launch.py
```

打开第三个终端(tros)，启动数据处理控制：

```
ssh -Y root@192.168.0.100
#将 python 版本设置为 3.10
rm /usr/bin/python3
ln -s /usr/bin/python3.10 /usr/bin/python3
source /opt/tros/local_setup.bash
source /home/wheeltec/wheeltec_tros/install/setup.bash
ros2 run wheeltec_bodyreader wheeltec_bodyreader
```

识别效果如图所示

图 3-5-1　WHEELTEC R550 小车运行骨架识别