

Atlas 200I DK A2 开发者套件
23.0.RC3

避障导航车应用开发指南

文档版本 01
发布日期 2023-11-14



版权所有 © 华为技术有限公司 2023。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

华为技术有限公司

地址： 深圳市龙岗区坂田华为总部办公楼 邮编： 518129

网址： <https://www.huawei.com>

客户服务邮箱： support@huawei.com

客户服务电话： 4008302118

安全声明

漏洞声明

华为公司对产品漏洞管理的规定以“漏洞处理流程”为准，该政策可参考华为公司官方网站的网址：<https://www.huawei.com/cn/psirt/vul-response-process>。

如企业客户须获取漏洞信息，请访问：<https://securitybulletin.huawei.com/enterprise/cn/security-advisory>。

目录

1 样例介绍	1
1.1 外观结构	2
1.2 功能与原理介绍	2
1.3 结构组成	3
1.3.1 智能小车底盘	3
1.3.2 激光雷达模块	4
2 样例组装	5
2.1 准备组件	5
2.2 组装步骤	6
3 运行环境准备	7
3.1 烧录镜像	7
3.2 获取代码	7
3.3 烧录电机代码	8
3.4 测试激光雷达	11
4 快速体验	13
4.1 建图功能	13
4.2 定位功能	20
4.3 导航功能	21
5 代码实现	23

1 样例介绍

自动导航技术已经成为了许多领域的关键应用，如物流、仓储管理、智能家居、机器人服务等。这项技术使得机器和设备能够自主地规划路径、避免障碍物、执行任务，从而提高了效率、降低了成本，并减少了人为操作的需求。华为昇腾Atlas 200I DK A2作为一款强大的开发者套件，为开发者提供了丰富的软硬件资源，可以用于构建智能导航系统。

本样例将使用昇腾Atlas 200I DK A2开发者套件和ROS（机器人操作系统）的nav2（Navigation 2）技术，开发一款自主导航车样例。样例将展示如何借助Atlas 200I DK A2的强大计算能力和AI加速能力，实现机器人或移动设备的自主导航和路径规划。

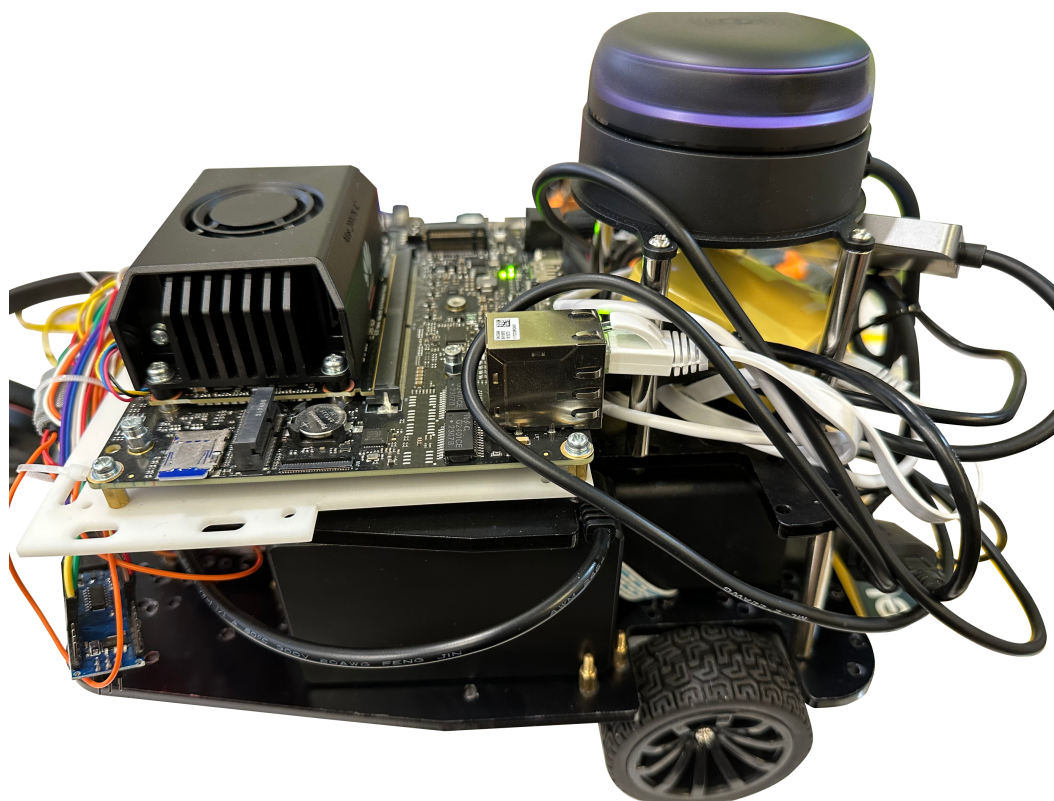
1.1 外观结构

1.2 功能与原理介绍

1.3 结构组成

1.1 外观结构

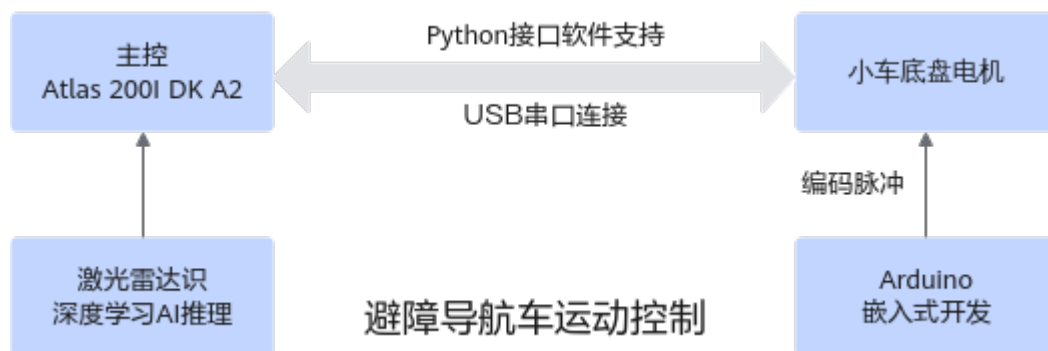
图 1-1 避障导航车外观结构图



1.2 功能与原理介绍

避障导航车首先通过激光雷达在已布置好的障碍物场地中进行地图扫描，模拟出一张带有障碍物标记的地图，然后根据雷达定位自身在地图中的位置，采集环境数据后在开发者套件上进行AI推理，根据推理结果发出指令以及使用Arduino平台可以对其编码电机进行嵌入式开发，控制小车运动的同时避开障碍物形式。

图 1-2 实现原理图



1.3 结构组成

1.3.1 智能小车底盘

三轮小车配备了2个主动橡胶轮和1个从动万向轮，具有轻巧、灵活等优点。规格：203.5mm*145mm、重量0.73kg、负载3kg、轻载速度1.25m/s

图 1-3 智能小车底盘

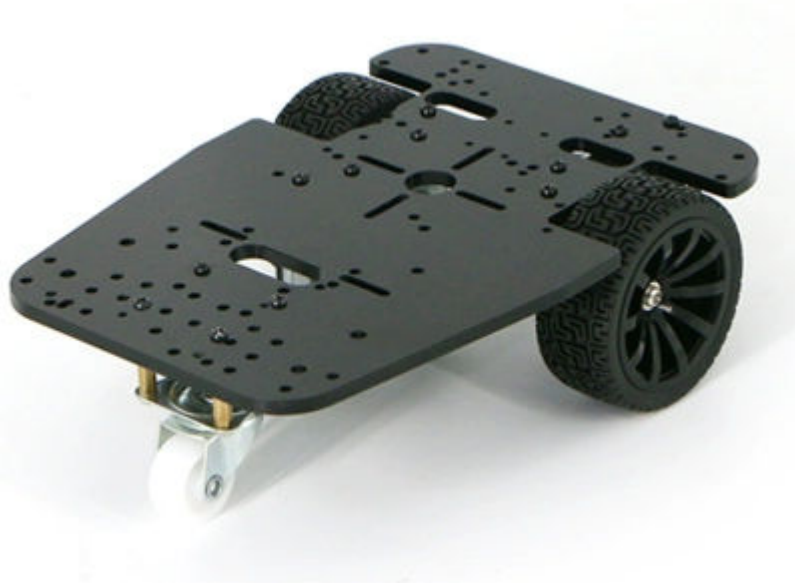


表 1-1 规格说明

规格	说明
长度	203.5mm
宽度	145mm
重量	0.73kg
电机型号	MG513X全金属齿轮电机
车轮	65mm橡胶轮2个、1寸万向轮1个
负载	3kg
负载1kg最大爬坡角度	30度
轻载速度	1.25m/s
编码器	13线霍尔

1.3.2 激光雷达模块

雷达模块支持同步定位与地图构建（SLAM）、环境扫描与3D重建、障碍物检测与规避安防、多点触碰式人机交互。

图 1-4 激光雷达模块



表 1-2 规格说明

规格	说明
测量半径	<ul style="list-style-type: none">白色物体：12m（米）黑色物体：10m（米）
通讯速率	115200bps
采样频率	16K
扫描频率	5Hz-15Hz
角分辨率	0.9°
机械尺寸	76*41mm
供电电流	450mA-600mA
功耗	2.25W-3W
输出	UART串口（3.3V电平）
工作温度	0摄氏度至40摄氏度
测距精度	<ul style="list-style-type: none">实际距离的1%（≤3m）实际距离的2%（3-5m）实际距离的2.5%（>5m）
ROS支持	ROS1与ROS2

2 样例组装

2.1 准备组件

2.2 组装步骤

2.1 准备组件

基于昇腾Atlas 200I DK A2已经实现了视觉方案的智能小车自动驾驶样例，另外也需要基于激光雷达的智能小车，由此开发了新版的激光雷达建图导航小车，舍弃掉多余的部件，仅选用必要的结构件支撑小车的主体以及开发者套件和激光雷达，以最小的尺寸实现激光雷达的静态建图以及可视化导航运动。

智能小车所需组件见表2-1。

表 2-1 智能小车型号表

模块名称	数量/个	规格	是否需要单独购买
昇腾开发者套件	1	Atlas 200I DK A2	否
智能小车底盘及其配件	1	见实际产品。	是
Arduinio开发板	1	-	是
电机稳压模块	1	-	是
公对公杜邦线	20	20mm	是
思岚A2M12激光雷达	1	A2雷达+转接板支架	是
十字套筒	1	唯一规格	是
UGREEN 绿联USB扩展板	1	唯一规格	是

模块名称	数量/个	规格	是否需要单独购买
3D打印支撑板（下载文件后通过3D打印）	1	唯一规格	是
GL.iNet MT300N-V2迷你路由器	1	唯一规格	是
小车所需额外紧固件 参见表2-2	-	-	是

表 2-2 紧固件数量表

紧固件名称	数量/个
M3*8+6单头六角铜柱	20
M3国标螺母/垫圈	20

2.2 组装步骤

- 步骤1** 首先按照智能小车底盘安装资料（购物时提供）安装好小车的底盘，需使用赠送的加大版的底盘以及编码电机和车轮。
- 步骤2** 使用M3*8+6的单头六角铜柱固定在底盘上之后，将开发者套件的电源卡入六角铜柱围成的区域内。
- 步骤3** 将**3D打印**的DK承载板固定到小车的黑色正常尺寸板材上，并接到底盘上，完成第二层的构建。
- 步骤4** 使用配件中的3*60银色支撑柱将激光雷达支撑在第二层上，连线完成后，即搭建好整个小车。

----结束

3 运行环境准备

- 3.1 烧录镜像
- 3.2 获取代码
- 3.3 烧录电机代码
- 3.4 测试激光雷达

3.1 烧录镜像

参见[一键制卡](#)章节烧录镜像，将SD卡插入开发者套件的卡槽，当前样例仅在Ubuntu OS适配验证过，未在openEuler OS适配验证，推荐烧录Ubuntu OS镜像。

3.2 获取代码

步骤1 远程登录开发者套件，进入“/usr/local”目录运行脚本拉取代码。

```
cd /usr/local
```

步骤2 运行脚本拉取代码。

```
bash E2E_samples_download_tool.sh -d download_destination_path -s source_repository -b branch target_path
```

参数说明：

- -d: 指定代码的下载路径。
- -s: 指定开源仓库的clone url。
- -b: 指定开源仓库分支名称及待下载的项目目录。
- -f: 强制更新下载路径中的目录。当样例目录已删除，但重新下载时提示“Already up to date”时可使用此参数。

命令示例：

```
bash E2E_samples_download_tool.sh -d /home/HwHiAiUser/E2ESamples -s https://gitee.com/HUAWEI-ASCEND/ascend-devkit.git -b master src/E2E-Sample/SLAM/
```

回显如下：

```
Download E2E samples successfully!
```

执行完成后，会在“/home/HwHiAiUser/E2Esamples”目录下生成“src/E2E-Sample/SLAM/”目录。

----结束

3.3 烧录电机代码

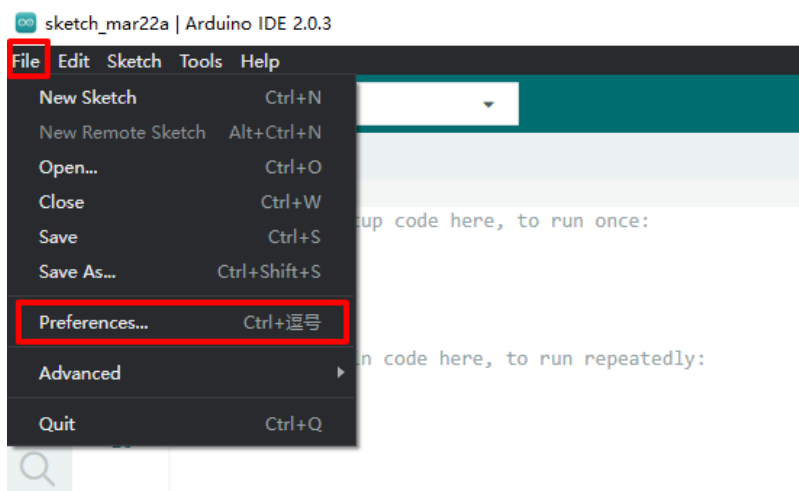
Arduino是一套便捷、灵活、容易上手的硬件开发平台，它包括多种型号的Arduino控制电路板和专用编程开发软件，能帮助用户快速的开发出智能硬件原型。

步骤1 进入[Arduino官网](#)下载程序安装包“arduino-ide_ version_Windows_64bit.exe”，并按照默认配置安装。

步骤2 双击打开软件，修改语言为中文。

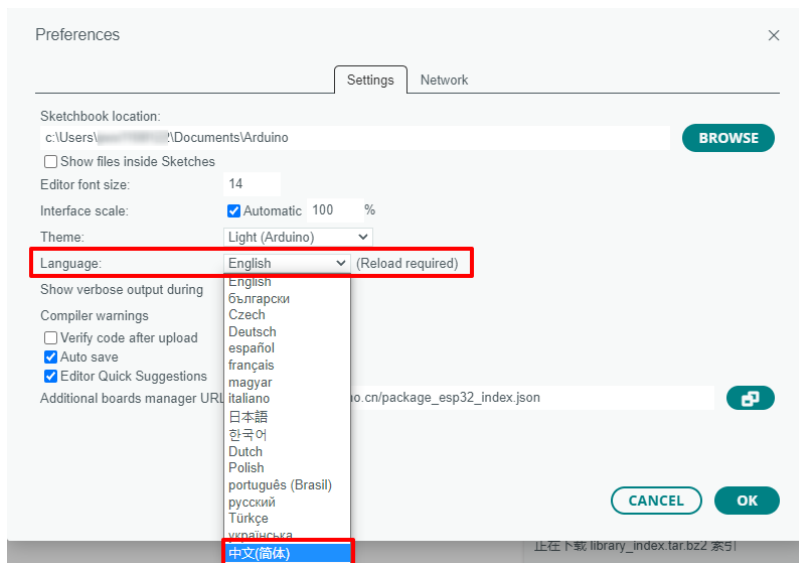
1. 单击“File > Preferences”，如**图3-1**所示。

图 3-1 软件菜单



2. 单击“Language”下拉菜单，选择“中文(简体)”，如**图3-2**所示。单击OK保存后工具会自动重启，重启后为中文界面。

图 3-2 修改语言



步骤3 将开发者套件的第一个USB口（上方）与激光雷达使用USB连接线进行连接，将开发者套件的第二个USB口（下方）与Arduino Nano开发板使用USB连接线进行连接。

步骤4 单击**下载**，获取代码仓并解压。

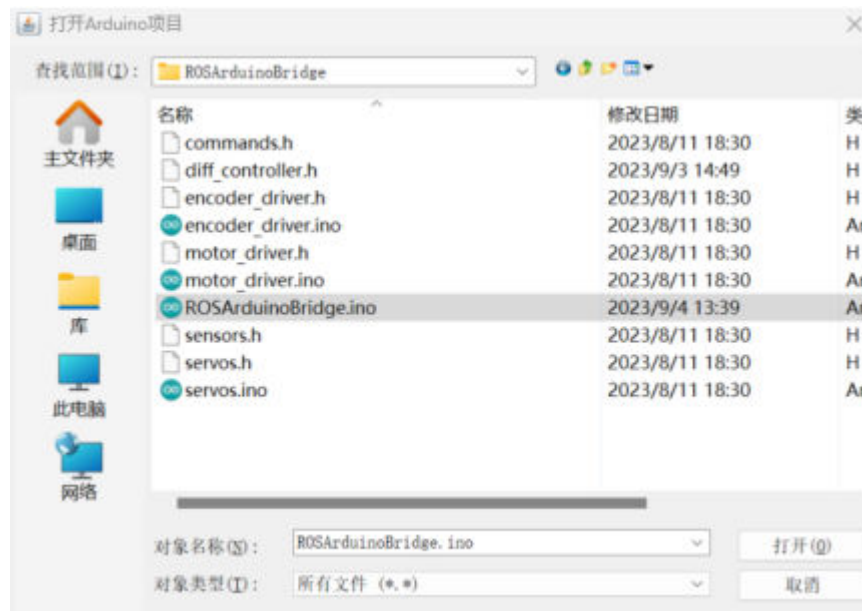
获得src/SLAM/ROSArduinoBridge电机代码。

步骤5 单击“文件”选择“打开”，选择打开**步骤4**目录中的ROSArduinoBridge.ino文件。

图 3-3 打开文件



图 3-4 选择文件



步骤6 进入代码页面后，选择“工具 > 开发板 > Arduino Uno”，处理器选择 ATmega328P(Old Bootloader)。

图 3-5 选择开发板

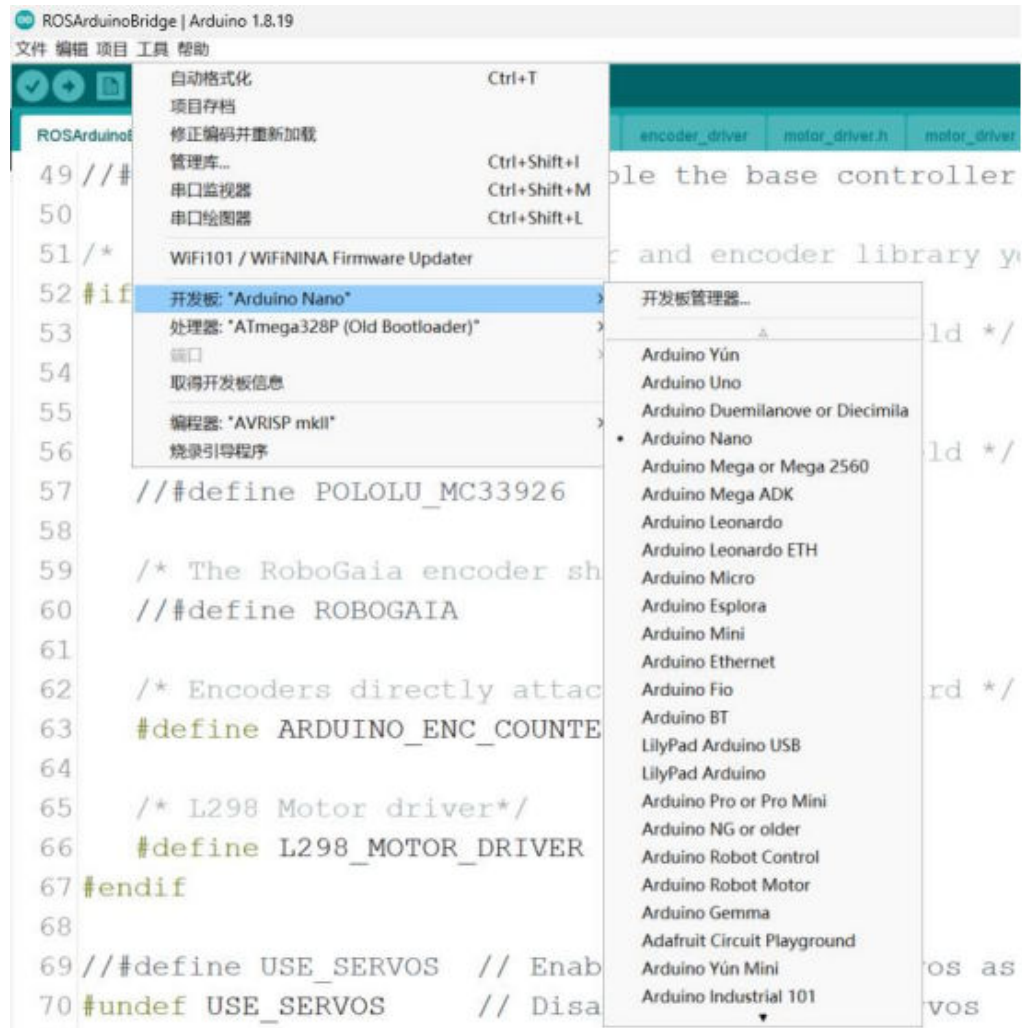
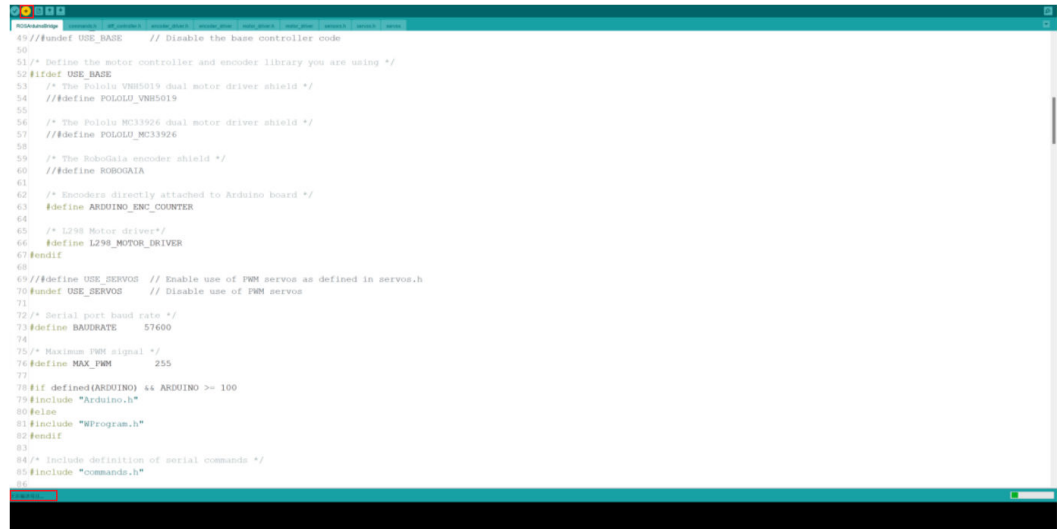


图 3-6 选择处理器



步骤7 单击界面左上角的上传按钮，等待右下角绿色进度条读条结束即上传成功。

图 3-7 上传代码



---结束

3.4 测试激光雷达

步骤1 以root用户远程登录小车的开发者套件。

步骤2 参考激光雷达模块的使用指导（购物时提供）在Ubuntu22.04环境下，下载好ROS2 Humble，并且按照要求配置好环境，输入如下指令。

```
ros2 launch rplidar_ros view_rplidar_a2m12_launch.py serial_port:=/dev/serial/by-id/usb-Silicon_Labs_CP2102_USB_to_UART_Bridge_Controller_0001-if00-port0 frame_id:=laser_frame  
angle_compensate:=true scan_mode:=Standard baud_rate:=115200
```

说明

需将serial_port参数替换为实际的端口号。

当激光雷达成功运转且输入ros2 topic list有/scan话题显示时表示激光雷达模块测试成功。

----**结束**

4 快速体验

4.1 建图功能

4.2 定位功能

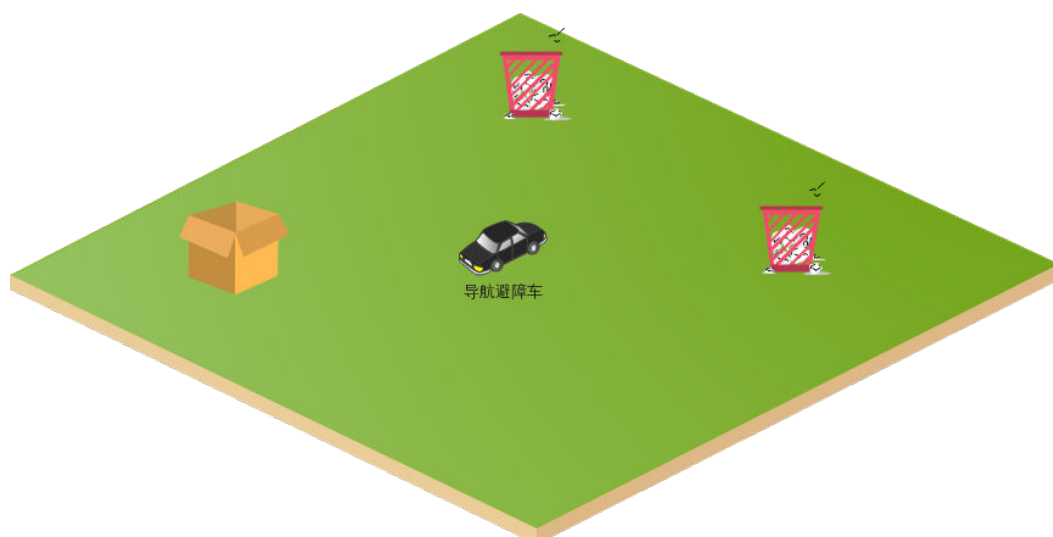
4.3 导航功能

4.1 建图功能

搭建行驶场地

在空旷的场地中，将小车放置在中间并在小车周围放置障碍物，例如：垃圾桶、纸箱等。需要保证两个障碍物之间有充足的空间，保证小车可以自由穿梭，示例环境如下所示。

图 4-1 场地示意图



编译工作空间

步骤1 进入导航车代码目录下。

```
cd /home/HwHiAiUser/E2Esamples/SLAM/ascend_slam_ws/
```


步骤3 再开启一个终端，以root用户登录开发者套件，执行以下命令启动激光雷达。

```
ros2 launch rplidar_ros view_rplidar_a2m12_launch.py serial_port:=/dev/serial/by-id/usb-Silicon_Labs_CP2102_USB_to_UART_Bridge_Controller_0001-if00-port0 frame_id:=laser_frame angle_compensate:=true scan_mode:=Standard baud_rate:=115200
```

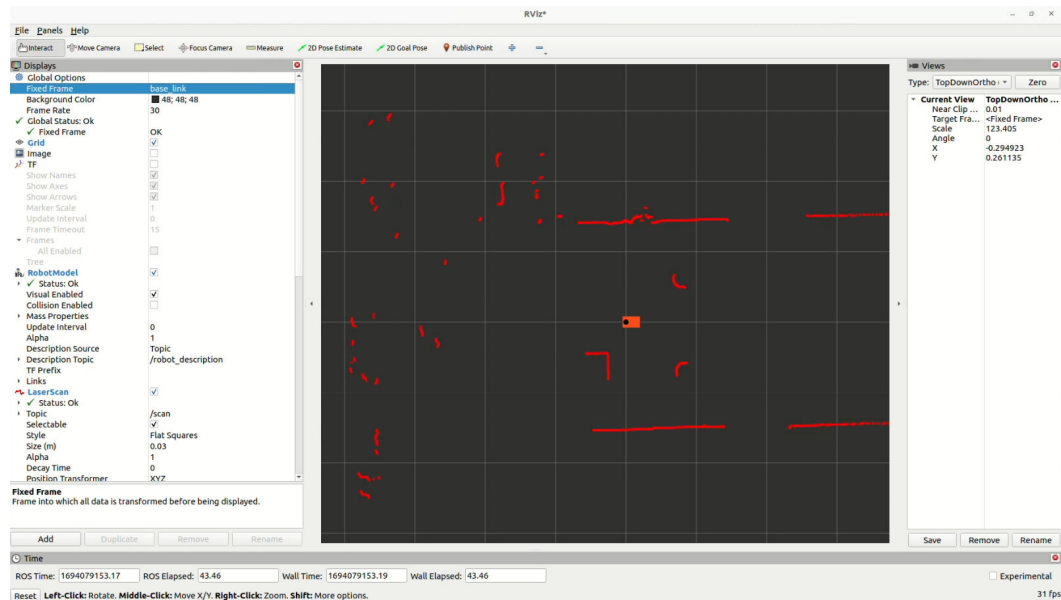
回显如下：

图 4-4 命令回显

```
[base] root@rplidar_ros2:~# ros2 launch sllidar_ros2 sllidar_a2m12_launch.py
[INFO] [launch]: All log files can be found below /root/.ros/log/2023-09-08-01-32-14-885035-davinci-n1n1-87488
[INFO] [launch]: Default logging verbosity is set to INFO
[INFO] [sllidar_node-1]: process started with pid [87489]
[sllidar_node-1] [INFO] [1694107935.151318664] [sllidar_node]: SLLidar running on ROS2 package SLLidar.ROS2 SDK Version:1.0.1, SLLIDAR SDK Version:2.0.0
[sllidar_node-1] [INFO] [1694107935.198895640] [sllidar_node]: SLLidar S/N: 88F899FC9E59AD4C5E59CF7742A3414
[sllidar_node-1] [INFO] [1694107935.200387286] [sllidar_node]: Firmware Ver: 1.32
[sllidar_node-1] [INFO] [1694107935.201690284] [sllidar_node]: Hardware Rev: 6
[sllidar_node-1] [INFO] [1694107935.254579128] [sllidar_node]: SLLidar health status : 0
[sllidar_node-1] [INFO] [1694107935.254732672] [sllidar_node]: SLLidar health status : OK.
[sllidar_node-1] [INFO] [1694107935.773621197] [sllidar_node]: current scan mode: Sensitivity, sample rate: 16 Khz, max_distance: 16.0 m, scan frequency:10.0 Hz,
```

此时查看RVIZ2可视化界面，则可查看点云数据。

图 4-5 RVIZ2 界面



步骤4 再开启一个终端，以root用户登录开发者套件，执行以下命令进行建图初始化。

```
ros2 launch ascend_slam online_async_launch.py params_file:=/home/HwHiAiUser/E2Esamples/SLAM/ascend_slam_ws/install/ascend_slam/share/ascend_slam/config/mapper_params_online_async.yaml use_sim_time:=false
```

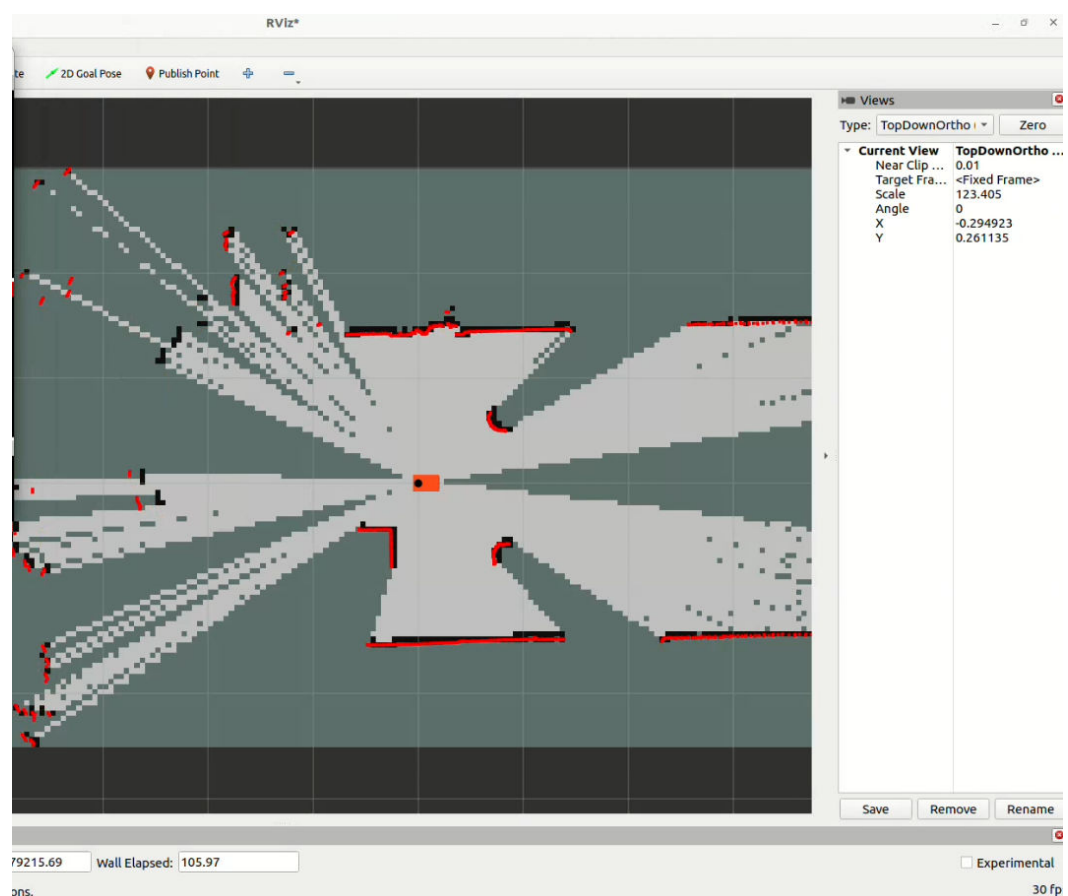
回显如下：

图 4-6 命令回显

```
root@da Vinci:~/ascend_slam_ws# ros2 launch ascend_slam_ws install/ascend_slam_ws/install/ascend_slam/share/ascend_slam/config/mapper_params_online_async.yaml use_sim_time:=false
[ascend_slam_toolbox_node-1] [INFO] [1694107949.868186778] [slam_toolbox]: Using solver plugin solver_plugins:CeresSolver
[ascend_slam_toolbox_node-1] [INFO] [1694107949.868918601] [slam_toolbox]: CeresSolver: Using SCHUR_JACOBI preconditioner.
[ascend_slam_toolbox_node-1] [INFO] [1694107949.997064809] [slam_toolbox]: Message filter dropping message: frame 'laser_frame' at time 1694107949.833 for reason 'discarding message because the queue is full'
[ascend_slam_toolbox_node-1] [WARN] [1694107950.080229552] [slam_toolbox]: maximum laser range setting (20.0 m) exceeds the capabilities of the used Lidar (16.0 m)
[ascend_slam_toolbox_node-1] Registering sensor: [Custom Described Lidar]
```

再次查看RVIZ2窗口，可看到一张初始化的未完成构建的地图出现在小车周围。

图 4-7 RVIZ2 界面



步骤5 开始扫描建图。

1. 执行如下命令，控制小车对当前区域进行建图。

```
ros2 run teleop_twist_keyboard teleop_twist_keyboard
```

根据以下指令，控制小车在地图中移动，完成建图。

- i: 前进
- j: 左转
- l: 右转
- k: 暂停

控制小车在地图中障碍物附近移动，此时地图会逐渐完整。

图 4-8 移动小车扫描建图

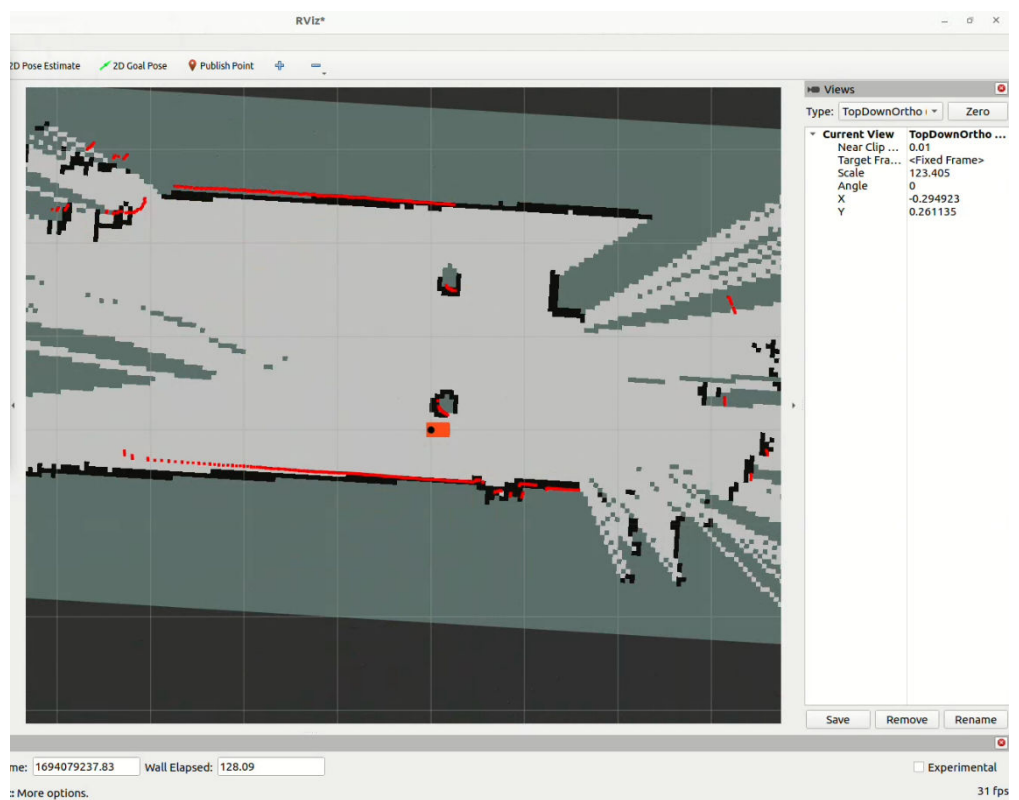
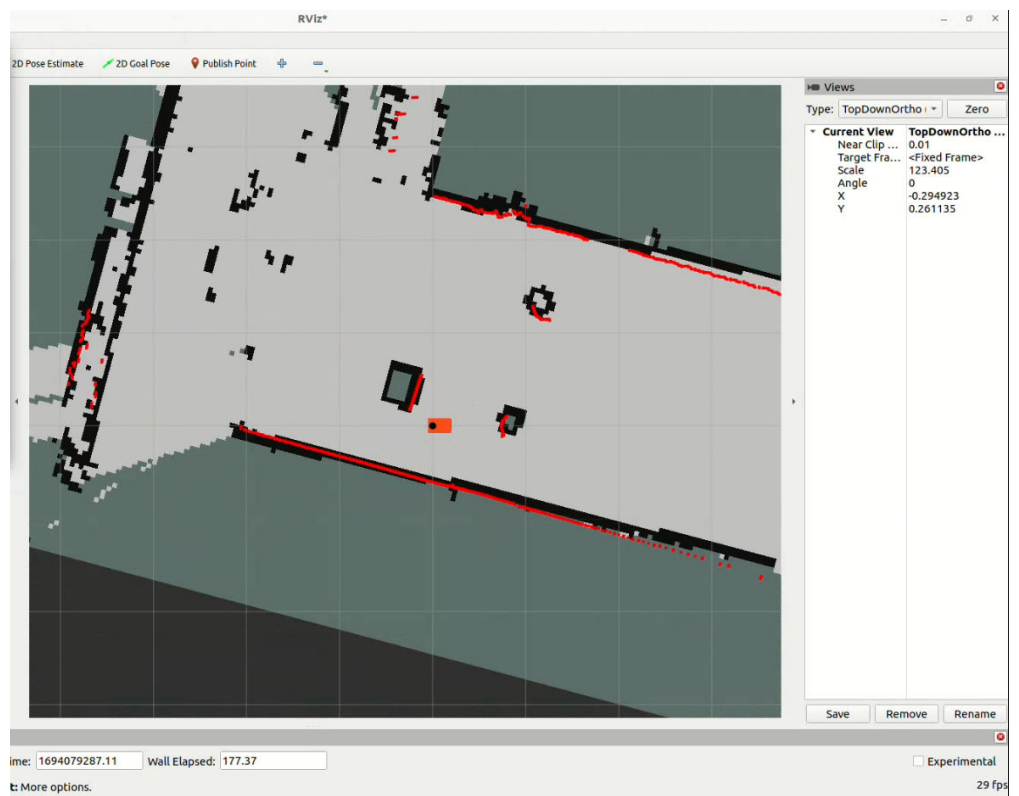


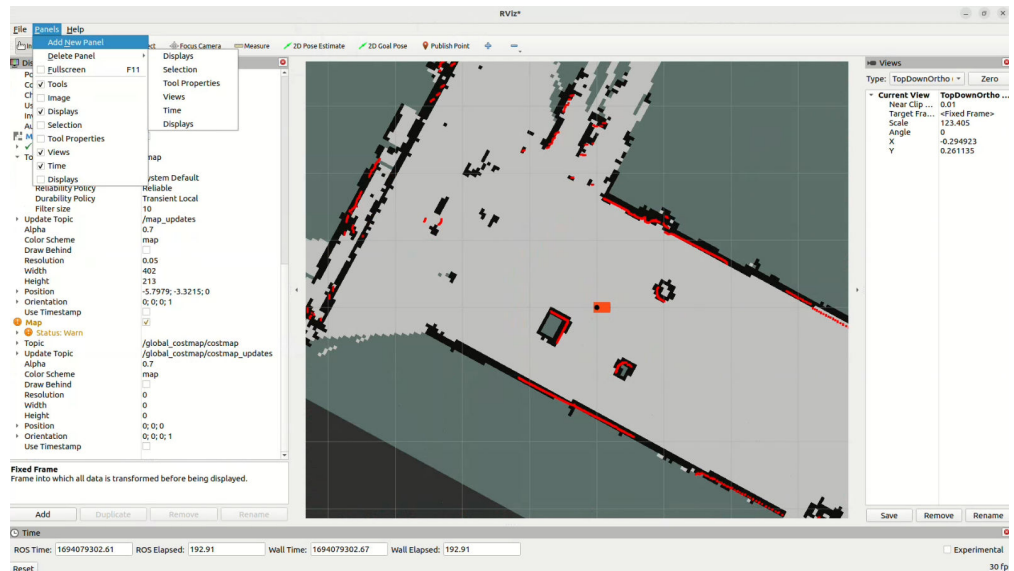
图 4-9 移动小车扫描建图



由上图可见，地图中间由两个黑色边框的椭圆和一个黑色边框的长方体。两椭圆对应两个垃圾桶的位置，而长方体则对应纸箱位置。当三个障碍物在地图中清晰可见后，即可完成建图。

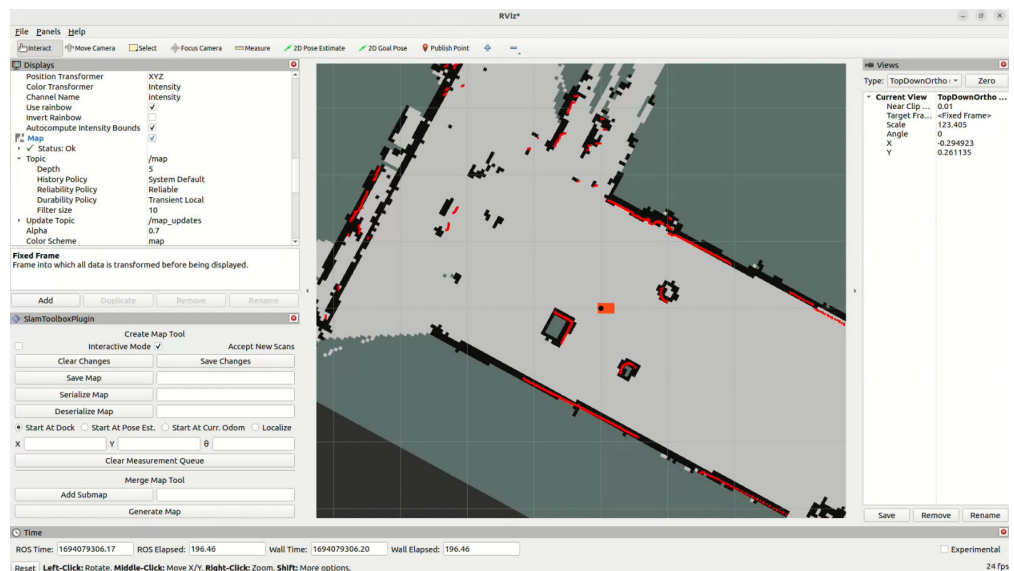
2. 单击左上角的Panels图标，单击“Add New Panel”按钮。

图 4-10 单击 Add New Panel



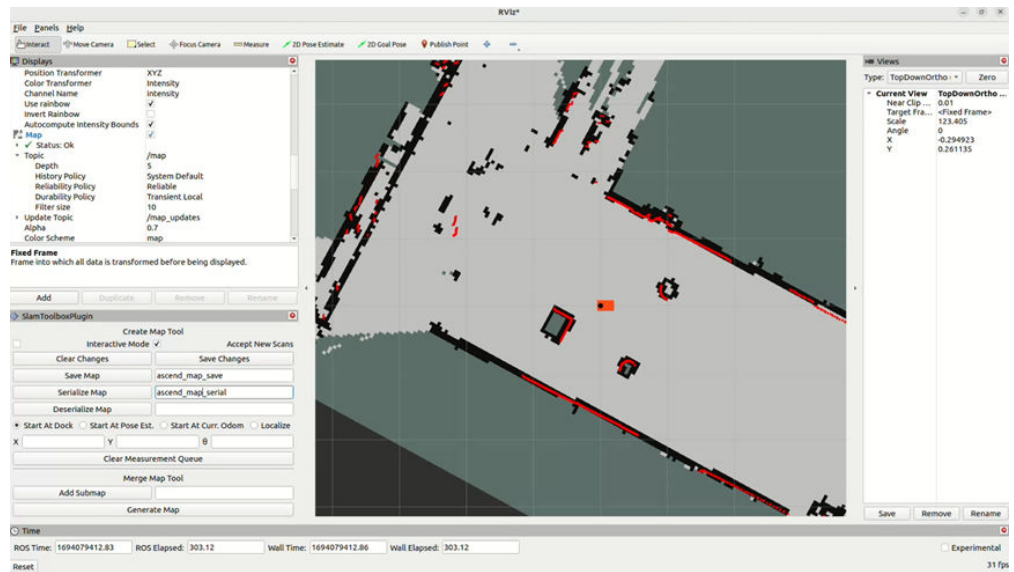
3. 在弹出的窗口中双击最下方的SlamToolboxPlugin，随后RVIZ2界面的左下方即会出现插件界面。

图 4-11 单击 SlamToolboxPlugin

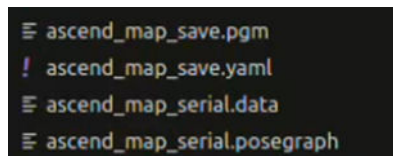


4. 在“Save Map”右侧输入“ascend_map_save”，在“Serialize Map”右侧输入“ascend_map_serial”。

图 4-12 配置地图信息



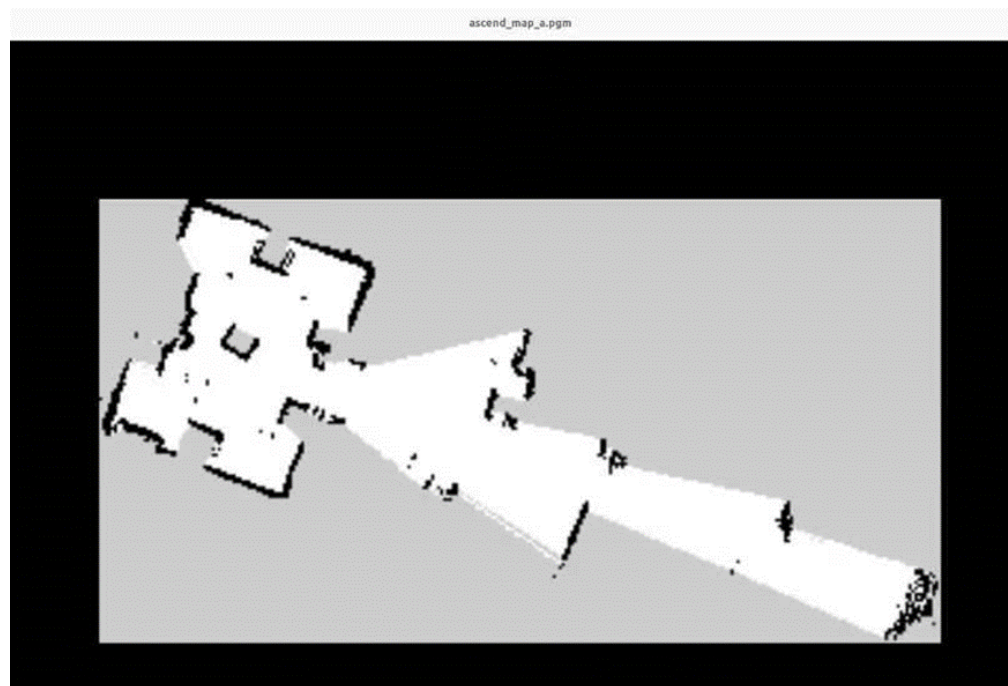
5. 分别单击“Save Map”按钮和“Serialize Map”按钮。
可在“ascend_slam_ws”目录下看到4份地图文件。



说明

因RVIZ2窗口在本机与DK进行通信，如地图文件没及时出现，可多点击几次。
若将这些地图文件下载至本地，可使用看图软件查看ascend_map_save.pgm，可对构建的地图进行进一步查看，示例地图如下。

图 4-13 查看地图



----结束

4.2 定位功能

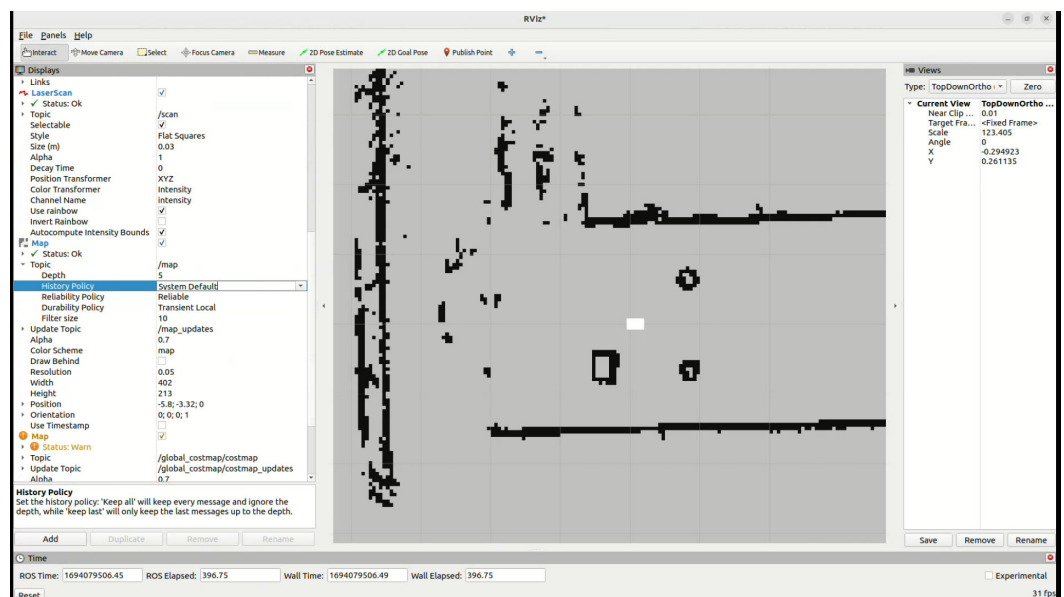
有了地图后，小车需要进一步了解此时此刻在地图中的初始位置。

步骤1 在当前终端，执行如下指令。

```
ros2 launch ascend_slam localization_launch.py map:=./ascend_map_save.yaml use_sim_time:=false
```

再次查看RVIZ2可视化界面，可以看到当前小车变为白色，表示我们需要在当前地图中，给小车一个初始位置，如下图所示。

图 4-14 初始化小车位置



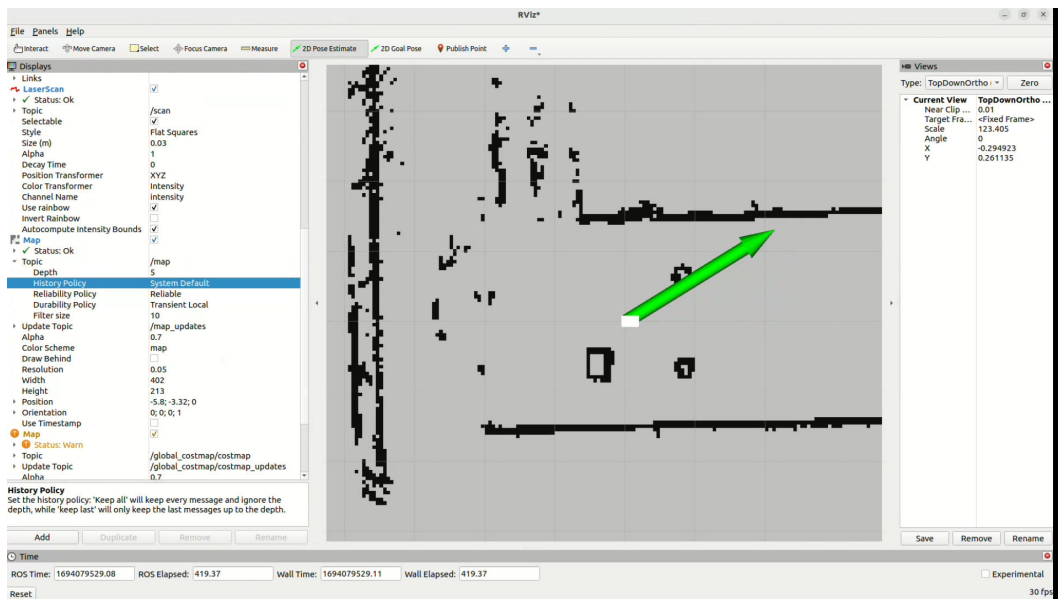
步骤2 点击上方的“2D Pose Estimate”按钮，根据小车当前位置，判断在地图中的估计位置。

图 4-15 点击按钮



步骤3 单击地图，按住鼠标左键并拉动一个车头所对准的方向，松开鼠标。随后小车会根据箭头的位置和方向，出现在箭头指向的位置。

图 4-16 初始化小车位置



说明

如小车未出现在指定位置或未移动，可能由于多机通信网络原因，尝试重新单击“2D Pose Estimate”按钮并完成相应操作。

----结束

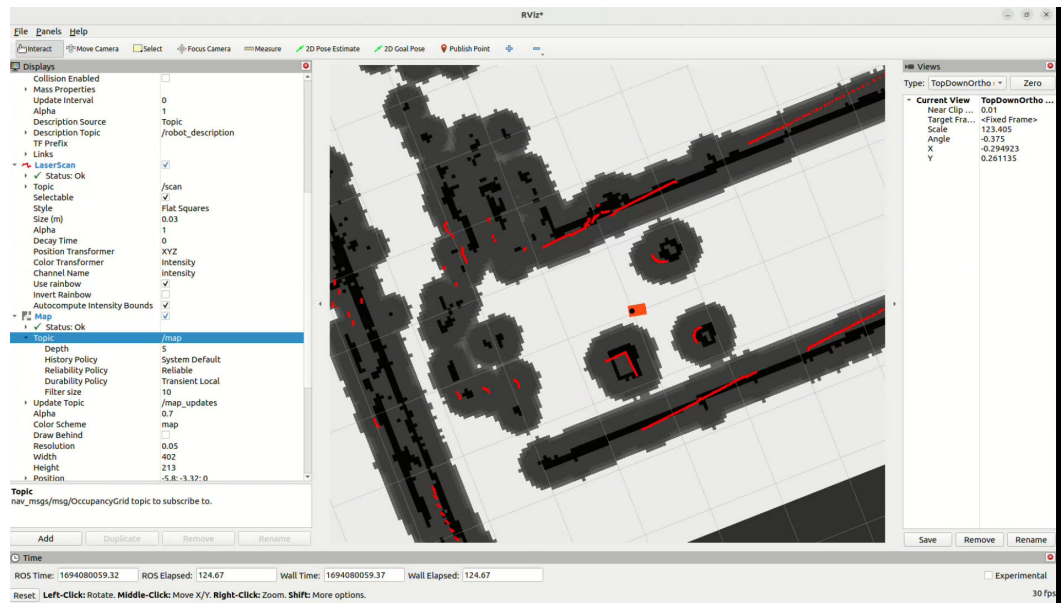
4.3 导航功能

步骤1 保持4.2 定位功能中所有终端开启并重新开启一个终端，以root用户登录开发者套件，执行以下命令。

```
ros2 launch ascend_slam navigation_launch.py use_sim_time:=false map_subscribe_transient_local:=true
```

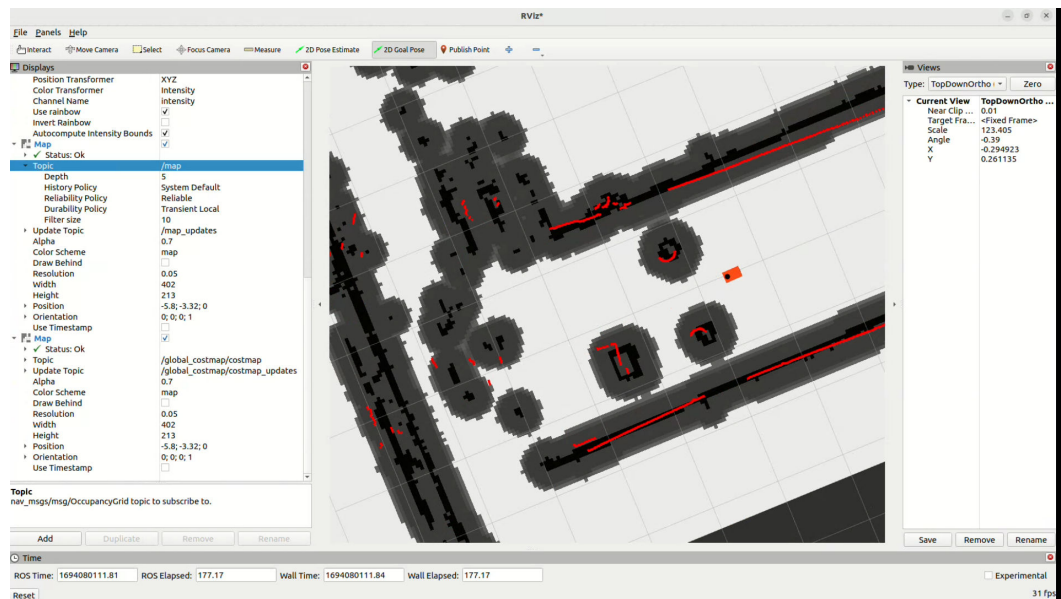
在RVIZ2中此时可以看到膨胀后的地图。

图 4-17 RVIZ2 界面



步骤2 单击上方的“2D Goal Pose”按钮，单击地图，按住鼠标左键并拉动一个车头所对准的方向，松开鼠标。随后小车会根据箭头的位置和方向，出现在箭头指向的位置。随后可看到小车会自行规划路径，在避开障碍物的同时，行驶至指定区域内。

图 4-18 自动避障导航行驶



----结束

5 代码实现



```

├── encoder_driver.ino
├── motor_driver.h
├── motor_driver.ino
├── ROSArduinoBridge.ino
├── sensors.h
├── servos.h
└── servos.ino
    
```

表 5-1 文件介绍

文件（夹）名称	说明
asend_slam_ws/src/ascend_slam/CMakeLists.txt	工作空间编译文件。
asend_slam_ws/src/ascend_slam/config	避障导航车配置文件。
asend_slam_ws/src/ascend_slam/description	避障导航车描述文件。
asend_slam_ws/src/ascend_slam/launch	避障导航车功能代码。
asend_slam_ws/src/ascend_slam/rviz	rviz工具配置文件。
asend_slam_ws/worlds	世界描述文件。
ROSArduinoBridge /	电机代码。