

APP mapping navigation

1. Content Description

This course explains how to use the ROS Robot app to control a robot for mapping and navigation.

This section requires entering commands in the terminal. The terminal you 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.

1.1. Preparation

You must first download the ROS Rotbot app on your phone. Android/iOS phone users, please scan the QR code to download the remote control software. iOS users can also search and download the ROSRobot map navigation app from the App Store.



The robot and phone must be on the same local area network, which can be achieved by connecting to the same Wi-Fi network.

2. App Map Creation

2.1. 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 [20. Docker course] --- [4. Docker Startup Script] for steps).

All the following commands must be executed within the same Docker container (**see [20. Docker course] --- [3. Docker Submission and Multi-Terminal Access] for steps**).

Enter the following command on the Car terminal to start the Car app map creation.

```
#Start the camera
#nuwa
ros2 launch yahboomcar_depth camera_app.launch.py
#usb
ros2 launch usb_cam camera.launch.py
```

root@raspberrypi: ~

```
File Edit Tabs Help
new publisher info set
[ascamera_node-1] [INFO] [1755085766.784833942] [ascamera_hp60c.camera_publisher]: 2025-08-13
19:49:26[INFO] [as_camera_sdk_api.cpp] [276] [AS_SDK_GetCameraAttrs] pid:26403 v1d13442 bus
num: 1 dev:6 port:2 ports:1.2
[ascamera_node-1] [INFO] [1755085766.784896960] [ascamera_hp60c.camera_publisher]: 2025-08-13
19:49:26[TfTreeFrameIdInfo.cpp] [11] [TfTreeFrameIdInfo] Namespace /ascamera_hp60c
[ascamera_node-1] [INFO] [1755085766.851647919] [ascamera_hp60c.camera_publisher]: 2025-08-13
19:49:26[INFO] [CameraHp60c.cpp] [1830] [getConfigurationParameters] according to the config
uration file.
[ascamera_node-1] [INFO] [1755085766.851727030] [ascamera_hp60c.camera_publisher]: 2025-08-13
19:49:26[INFO] [CameraHp60c.cpp] [1535] [parseConfigFileParameter] moduleName:AS_CAM_HP60C
[ascamera_node-1] [INFO] [1755085766.851802475] [ascamera_hp60c.camera_publisher]: 2025-08-13
19:49:26[INFO] [CameraHp60c.cpp] [1562] [parseConfigFileParameter] Parsing config file /root
/yahboomcar_ross2_ws/software/library_ws/src/ascamera/configurationfiles/hp60c_v2_00_20230704_
configEncrypt.json
[ascamera_node-1] [INFO] [1755085766.851939123] [ascamera_hp60c.camera_publisher]: 2025-08-13
19:49:26[INFO] [CameraHp60c.cpp] [1627] [parseConfigFileParameter] configuration protocol Ve
rsion: v2.x
[ascamera_node-1] [INFO] [1755085766.851962438] [ascamera_hp60c.camera_publisher]: 2025-08-13
19:49:26[INFO] [CameraHp60c.cpp] [1638] [parseConfigFileParameter] configuration Version: v2
.0.0.20230704
[ascamera_node-1] [INFO] [1755085766.855704134] [ascamera_hp60c.camera_publisher]: 2025-08-13
19:49:26[INFO] [CameraHp60c.cpp] [1803] [setParamersAfterOpenCam] Camera confiParaEnable tru
e, setting configuration parameter
```

```
#Choose one of the two mapping algorithms
ros2 launch yahboomcar_nav map_gmapping_app_launch.xml
ros2 launch yahboomcar_nav map_cartographer_app_launch.xml
```

root@raspberrypi: ~/ros

```
File Edit Tabs Help
[static_transform_publisher-14] from 'ascamera_hp60c_camera_link_0' to 'ascamera_hp60c_color_0'
[ydlidar_ros2_driver_node-12] [YDLIDAR] Lidar running correctly! The health status: good
[ydlidar_ros2_driver_node-12] [YDLIDAR] Baseplate device info
[ydlidar_ros2_driver_node-12] Firmware version: 1.2
[ydlidar_ros2_driver_node-12] Hardware version: i
[ydlidar_ros2_driver_node-12] Model: Tmini Plus
[ydlidar_ros2_driver_node-12] Serial: 2025021500090139
[ydlidar_ros2_driver_node-12] [YDLIDAR] Current scan frequency: 10.00Hz
[ydlidar_ros2_driver_node-12] [YDLIDAR] Lidar init success, Elapsed time 1010 ms
[rosbridge_websocket-1] [INFO] [1755085795.402560331] [rosbridge_websocket]: Rosbridge WebSocket server started on port 9090
[joint_state_publisher-4] [INFO] [1755085795.406356805] [joint_state_publisher]: Waiting for robot_description to be published on the r
obot_description topic...
[ydlidar_ros2_driver_node-12] [YDLIDAR] Create thread 0x793E8E0
[ydlidar_ros2_driver_node-12] [YDLIDAR] Succeeded to start scan mode, Elapsed time 2098 ms
[ydlidar_ros2_driver_node-12] [YDLIDAR] Fixed Size: 404
[ydlidar_ros2_driver_node-12] [YDLIDAR] Sample Rate: 4.00K
[ydlidar_ros2_driver_node-12] [YDLIDAR] Succeeded to check the lidar, Elapsed time 0 ms
[ydlidar_ros2_driver_node-12] [2025-08-13 19:49:56][info] [YDLIDAR] Now lidar is scanning...
[slam_gmapping-16] [INFO] [1755085796.103259136] [slam_gmapping]: Laser is mounted upwards.
[slam_gmapping-16] [INFO] [1755085796.104048490] [slam_gmapping]: Initialization complete
[slam_gmapping-16] Laser Pose= 0.0060585 0 -3.14159
[slam_gmapping-16] -maxUrange 5 -maxOrange 8 -sigma 0.02 -kernelSize 1 -lstep 0.05 -lobsGain 3 -astep 0.05
[slam_gmapping-16] -srr 0.1 -srt 0.2 -str 0.1 -stt 0.2
[slam_gmapping-16] -linearUpdate 1 -angularUpdate 0.5 -resampleThreshold 0.5
[slam_gmapping-16] -xmin -10 -xmax 10 -ymin -10 -ymax 10 -delta 0.05 -particles 50
[slam_gmapping-16] update frame 0
[slam_gmapping-16] update ld=0 ad=0
[slam_gmapping-16] m_count 0
[slam_gmapping-16] Registering First Scan
[imu_filter_madgwick_node-8] [INFO] [1755085796.292427945] [imu_filter_madgwick]: First IMU message received.
```

The mobile app displays the following image. Enter the car's IP address, using [zh] for Chinese and [en] for English. Select ROS2. In the Video Target area below, select /camera/rgb/image_raw/compressed. Finally, click [Connect].

Enter robot IP

zh en

Please enter IP address

192.168.11.203

ROS1

ROS2

Video Topic

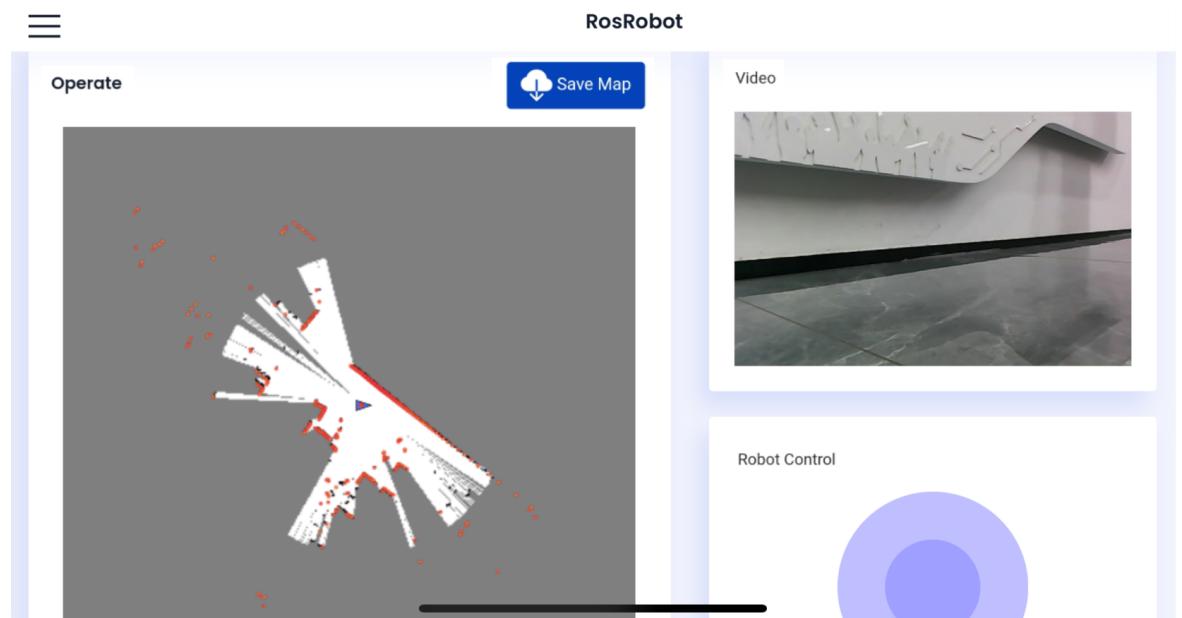
/usb_cam/image_raw/compressed

/camera/rgb/image_raw/compressed

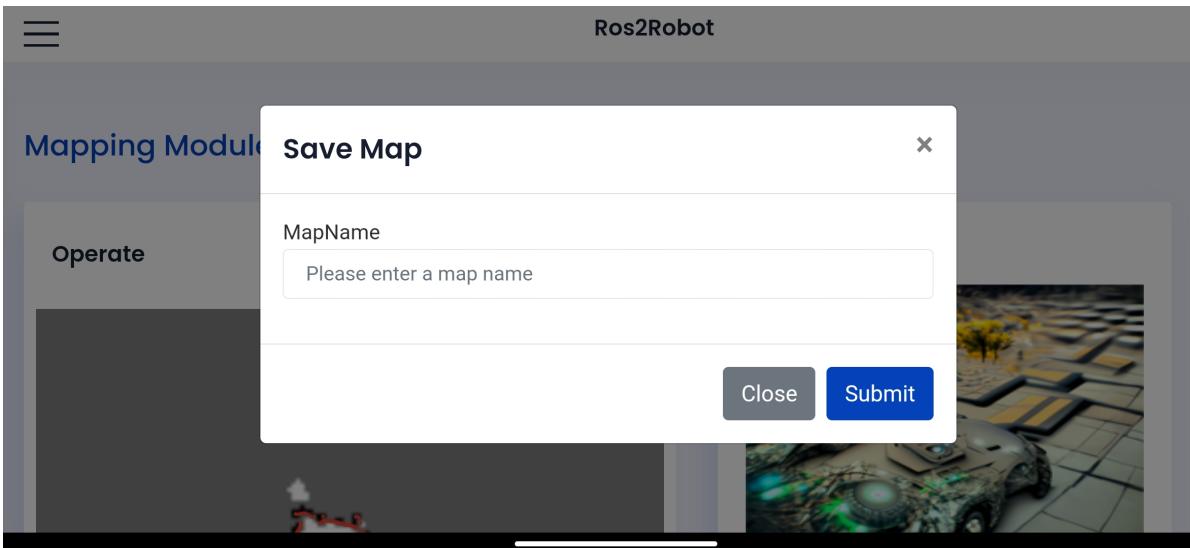
/csi/image_raw/compressed

Connect

After successfully connecting, the following display appears:

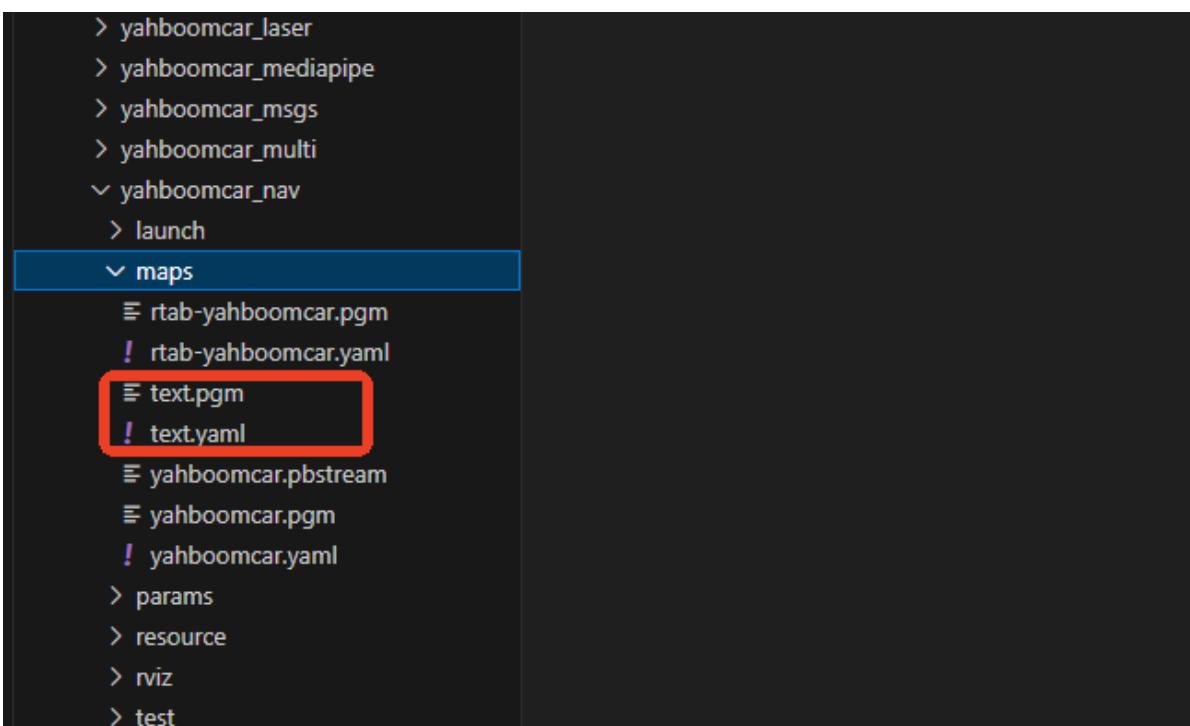


Use the wheel to slowly move the car around the area you want to map. Then click Save Map, enter a map name, and click Submit to save the map.



The map is saved in:

```
~/yahboomcar_ros2_ws/yahboomcar_ws/src/yahboomcar_nav/maps
```



2.2. Launch Command Analysis

Code Path:

```
~/yahboomcar_ros2_ws/yahboomcar_ws/src/yahboomcar_nav/launch/map_gmapping_app_launch.xml
```

Using the `map_gmapping_app_launch.xml` file as an example, we will explain which launch files are launched. The `map_gmapping_app_launch.xml` file is as follows:

```

<launch>
  <arg name="use_rviz" default="false" />
  <include file="$(find-pkg-share
rosbridge_server)/launch/rosbridge_websocket_launch.xml"/>
  <node name="laserscan_to_point_publisher" pkg="laserscan_to_point_publisher"
exec="laserscan_to_point_publish">
    <include file="$(find-pkg-share
yahboomcar_nav)/launch/map_gmapping_launch.py">
      <arg name="use_rviz" value="$(var use_rviz)"/>
    </include>
    <include file="$(find-pkg-share
robot_pose_publisher_ross2)/launch/robot_pose_publisher_launch.py"/>
    <include file="$(find-pkg-share
yahboomcar_app_save_map)/yahboom_app_save_map.launch.py"/>
  </node>
</launch>

```

rosbridge_websocket_launch.xml: Launches the rosbridge WebSocket server for ROS (Robot Operating System)

laserscan_to_point_publisher: Publishes radar point cloud data to the app

map_gmapping_launch: Gmapping mapping-related nodes

robot_pose_publisher_launch.py: Publishes vehicle position information

yahboom_app_save_map.launch.py: Saves the map node

3. App Navigation

3.1. Program Startup

Enter the following command on the car terminal to start the car chassis and radar.

```
root@raspberrypi: ~
ros2 launch yahboomcar_nav laser_bringup_launch.py
```

```

root@raspberrypi: ~
File Edit Tabs Help
[static_transform_publisher-11] translation: ('0.000000', '0.000000', '0.000000')
[static_transform_publisher-11] rotation: ('0.000000', '0.000000', '0.000000', '1.000000')
[static_transform_publisher-11] from 'ascamera_hp60c_camera_link_0' to 'ascamera_hp60c_color_0'
[static_transform_publisher-10] [INFO] [1755086619.567551945] [static_transform_publisher_v3hEzRPM8dFJrqHe]: Spinning until stopped - publishing transform
[static_transform_publisher-10] translation: ('0.006058', '0.000000', '0.149120')
[static_transform_publisher-10] rotation: ('0.000046', '0.000046', '-1.000000', '0.000000')
[static_transform_publisher-10] from 'base_link' to 'laser'
[static_transform_publisher-12] [INFO] [1755086619.878549130] [static_transform_publisher_9woQaiTXA05G8Fjw]: Spinning until stopped - publishing transform
[static_transform_publisher-12] translation: ('0.000000', '0.000000', '0.000000')
[static_transform_publisher-12] rotation: ('-0.500000', '0.500000', '-0.500000', '0.500000')
[static_transform_publisher-12] from 'camera_link' to 'ascamera_hp60c_camera_link_0'
[ydlidar_ros2_driver_node-9] [YDLIDAR] Lidar running correctly! The health status: good
[joint_state_publisher-1] [INFO] [1755086620.131394223] [joint_state_publisher]: Waiting for robot_description to be published on the robot_description topic...
[ydlidar_ros2_driver_node-9] [YDLIDAR] Baseplate device info
[ydlidar_ros2_driver_node-9] Firmware version: 1.2
[ydlidar_ros2_driver_node-9] Hardware version: 1
[ydlidar_ros2_driver_node-9] Model: Tmini Plus
[ydlidar_ros2_driver_node-9] Serial: 2025021500000139
[ydlidar_ros2_driver_node-9] [YDLIDAR] Current scan frequency: 10.00Hz
[ydlidar_ros2_driver_node-9] [YDLIDAR] Lidar init success, Elapsed time 1032 ms
[imu_filter_madgwick_node-5] [INFO] [1755086620.821986491] [imu_filter_madgwick]: First IMU message received.
[ydlidar_ros2_driver_node-9] [YDLIDAR] Create thread 0x82CDE8E0
[ydlidar_ros2_driver_node-9] [YDLIDAR] Successed to start scan mode, Elapsed time 2097 ms
[ydlidar_ros2_driver_node-9] [YDLIDAR] Fixed Size: 404
[ydlidar_ros2_driver_node-9] [YDLIDAR] Sample Rate: 4.00K
[ydlidar_ros2_driver_node-9] [YDLIDAR] Successed to check the lidar, Elapsed time 0 ms
[ydlidar_ros2_driver_node-9] [2025-08-13 20:03:42][info] [YDLIDAR] Now lidar is scanning...

```

Enter the following command on the car terminal to start the camera.

```
#Start the camera
#nuwa
ros2 launch yahboomcar_depth camera_app.launch.py
#usb
ros2 launch usb_cam camera.launch.py
```

The terminal window shows the log output for the camera initialization process. It includes messages from the camera node, such as 'image_compressor' starting, camera publisher details, and configuration parameters being loaded from 'ascamera_hp60c_v2_00_20230704_configEncrypt.json'. The log is timestamped from August 13, 2025.

```
root@raspberrypi: ~/ros
File Edit Tabs Help
[camera_app-2] [INFO] [1755086644.069138060] [image_compressor]: 图像压缩节点已启动，正在监听原始图像并发布压缩图像...
[ascamera_node-1] [INFO] [1755086644.968286124] [ascamera_hp60c.camera_publisher]: 2025-08-13 20:04:04[INFO] [CameraSrv.cpp] [158] [onAttached] get model type 9
[ascamera_node-1] [INFO] [1755086644.968536834] [ascamera_hp60c.camera_publisher]: 2025-08-13 20:04:04[INFO] [CameraSrv.cpp] [352] [getConfigFromFile] get file: /root/yahboomcar_ros2_ws/software/library_ws/src/ascamera/configurationfiles/hp60c_v2_00_20230704_configEncrypt.json
[ascamera_node-1] [INFO] [1755086644.968613978] [ascamera_hp60c.camera_publisher]: 2025-08-13 20:04:04[INFO] [as_camera_sdk_api.cpp] [276] [AS_SDK_GetCameraAttrs] pid:26403 vid13442 busnum: 1 dev:6 port:2 ports:1.2
[ascamera_node-1] [INFO] [1755086644.968639366] [ascamera_hp60c.camera_publisher]: usb camera
[ascamera_node-1] [INFO] [1755086644.968661550] [ascamera_hp60c.camera_publisher]: bnum:1
[ascamera_node-1] [INFO] [1755086644.968674660] [ascamera_hp60c.camera_publisher]: dnum:6
[ascamera_node-1] [INFO] [1755086644.968688382] [ascamera_hp60c.camera_publisher]: port_numbers:1.2
[ascamera_node-1] [INFO] [1755086644.968703936] [ascamera_hp60c.camera_publisher]: create a new publisher info set
[ascamera_node-1] [INFO] [1755086644.978030211] [ascamera_hp60c.camera_publisher]: 2025-08-13 20:04:04[INFO] [as_camera_sdk_api.cpp] [276] [AS_SDK_GetCameraAttrs] pid:26403 vid13442 busnum: 1 dev:6 port:2 ports:1.2
[ascamera_node-1] [INFO] [1755086644.978144187] [ascamera_hp60c.camera_publisher]: 2025-08-13 20:04:04[INFO] [TfTreeFrameIdInfo.cpp] [141] [TfTreeFrameIdInfo] Namespace /ascamera_hp60c
[ascamera_node-1] [INFO] [1755086645.047714735] [ascamera_hp60c.camera_publisher]: 2025-08-13 20:04:05[INFO] [CameraHp60c.cpp] [1830] [getConfigurationParameters] according to the configuration file.
[ascamera_node-1] [INFO] [1755086645.047785787] [ascamera_hp60c.camera_publisher]: 2025-08-13 20:04:05[INFO] [CameraHp60c.cpp] [1535] [parseConfigFileParameter] moduleName:AS_CAM_HP60C
[ascamera_node-1] [INFO] [1755086645.047865617] [ascamera_hp60c.camera_publisher]: 2025-08-13 20:04:05[INFO] [CameraHp60c.cpp] [1562] [parseConfigFileParameter] Parsing config file /root/yahboomcar_ros2_ws/software/library_ws/src/ascamera/configurationfiles/hp60c_v2_00_20230704_configEncrypt.json
[ascamera_node-1] [INFO] [1755086645.048005369] [ascamera_hp60c.camera_publisher]: 2025-08-13 20:04:05[INFO] [CameraHp60c.cpp] [1627] [parseConfigFileParameter] configuration protocol Version: v2.x
[ascamera_node-1] [INFO] [1755086645.048026201] [ascamera_hp60c.camera_publisher]: 2025-08-13 20:04:05[INFO] [CameraHp60c.cpp] [1638] [parseConfigFileParameter] configuration Version: v2.0.0.20230704
[ascamera_node-1] [INFO] [1755086645.051425661] [ascamera_hp60c.camera_publisher]: 2025-08-13 20:04:05[INFO] [CameraHp60c.cpp] [1803] [setParamsAfterOpenCam] Camera confiParaEnable true, setting configuration parameter
```

Enter the following command on the car terminal to launch the app navigation.

If you want to use fast relocalization navigation, you must first follow the tutorials [6. LiDAR - 7. Cartographer Mapping] to save a map in pbstream format and a map in yaml format.

```
# Navigation: Choose one of the two options
#1. Normal positioning and navigation
#dwa
ros2 launch yahboomcar_nav navigation_dwa_app_launch.xml
#teb
ros2 launch yahboomcar_nav navigation_teb_app_launch.xml

#2. Fast relocalization and navigation
#2.1 Positioning
#jetson nano, raspberry Pi5
ros2 launch yahboomcar_nav localization_imu_odom.launch.py use_rviz:=false
load_state_filename:=/root/yahboomcar_ros2_ws/yahboomcar_ws/src/yahboomcar_nav/maps/yahboomcar.pbstream
#ORIN
ros2 launch yahboomcar_nav localization_imu_odom.launch.py use_rviz:=false
load_state_filename:=/home/jetson/yahboomcar_ros2_ws/yahboomcar_ws/src/yahboomcar_nav/maps/yahboomcar.pbstream
#RDK X5
ros2 launch yahboomcar_nav localization_imu_odom.launch.py use_rviz:=false
load_state_filename:=/home/sunrise/yahboomcar_ros2_ws/yahboomcar_ws/src/yahboomcar_nav/maps/yahboomcar.pbstream

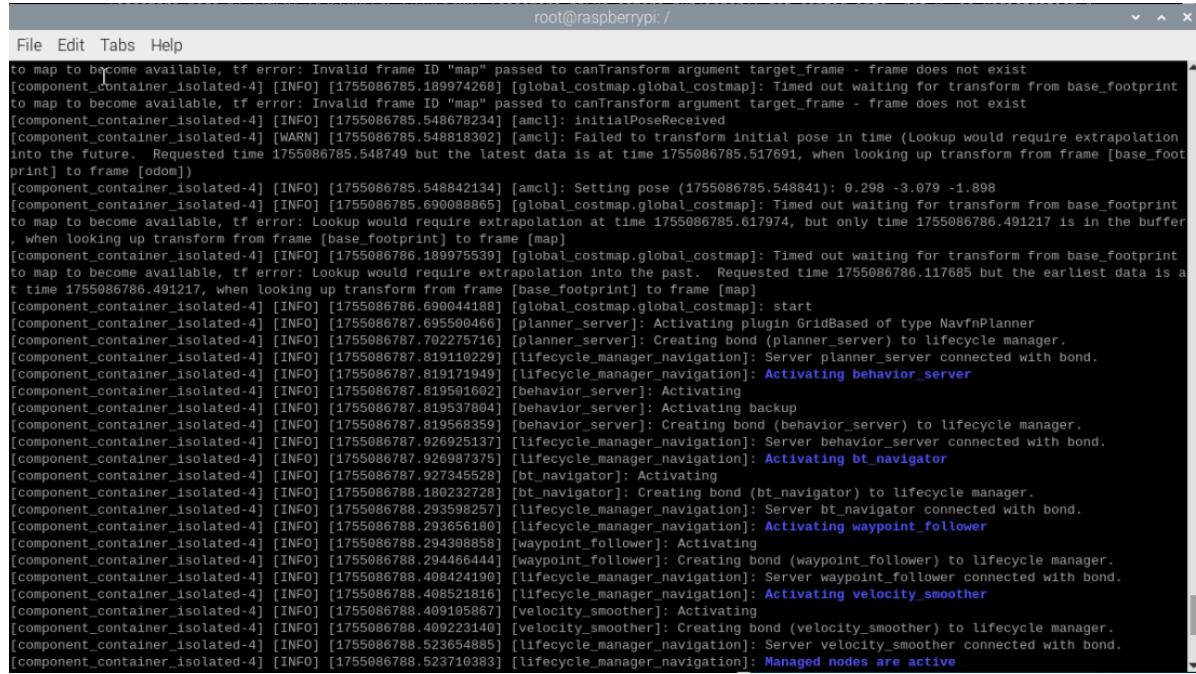
#2.2 Navigation
#jetson nano, raspberry Pi5
```

```

ros2 launch yahboomcar_nav navigation_cartodwb_app_launch.xml
maps:=~/root/yahboomcar_ros2_ws/yahboomcar_ws/src/yahboomcar_nav/maps/yahboomcar.yaml
#ORIN
ros2 launch yahboomcar_nav navigation_cartodwb_app_launch.xml
maps:=~/home/jetson/yahboomcar_ros2_ws/yahboomcar_ws/src/yahboomcar_nav/maps/yahboomcar.yaml
#RDK X5
ros2 launch yahboomcar_nav navigation_cartodwb_app_launch.xml
maps:=~/home/sunrise/yahboomcar_ros2_ws/yahboomcar_ws/src/yahboomcar_nav/maps/yahboomcar.yaml

```

Take Normal positioning navigation as an example.



The screenshot shows a terminal window titled 'root@raspberrypi:/' displaying ROS2 navigation logs. The logs detail the initialization of various components, including the global costmap, planner server, lifecycle manager, and behavior servers. Key messages include:

- [INFO] [1755086785.548841]: Setting pose (1755086785.548841): 0.298 -3.079 -1.898
- [INFO] [1755086785.60008865]: Timed out waiting for transform from base_footprint to map to become available, tf error: Invalid frame ID "map" passed to canTransform argument target_frame - frame does not exist
- [INFO] [1755086785.548878234]: [amcl]: initialPoseReceived
- [WARN] [1755086785.548818302]: [amcl]: Failed to transform initial pose in time (Lookup would require extrapolation into the future. Requested time 1755086785.548749 but the latest data is at time 1755086785.517691, when looking up transform from frame [base_footprint] to frame [odom])
- [INFO] [1755086786.690044188]: [global_costmap.global_costmap]: start
- [INFO] [1755086787.695500466]: [planner_server]: Activating plugin GridBased of type NavfnPlanner
- [INFO] [1755086787.702275716]: [planner_server]: Creating bond (planner_server) to lifecycle manager.
- [INFO] [1755086787.819110229]: [lifecycle_manager_navigation]: Server planner_server connected with bond.
- [INFO] [1755086787.819171949]: [lifecycle_manager_navigation]: Activating behavior_server
- [INFO] [1755086787.819501602]: [behavior_server]: Activating
- [INFO] [1755086787.819537804]: [behavior_server]: Activating backup
- [INFO] [1755086787.819568359]: [behavior_server]: Creating bond (behavior_server) to lifecycle manager.
- [INFO] [1755086787.926025137]: [lifecycle_manager_navigation]: Server behavior_server connected with bond.
- [INFO] [1755086787.926987375]: [lifecycle_manager_navigation]: Activating bt_navigator
- [INFO] [1755086787.927345528]: [bt_navigator]: Activating
- [INFO] [1755086788.180232728]: [bt_navigator]: Creating bond (bt_navigator) to lifecycle manager.
- [INFO] [1755086788.293598257]: [lifecycle_manager_navigation]: Server bt_navigator connected with bond.
- [INFO] [1755086788.293656180]: [lifecycle_manager_navigation]: Activating waypoint_follower
- [INFO] [1755086788.294308858]: [waypoint_follower]: Activating
- [INFO] [1755086788.294466444]: [waypoint_follower]: Creating bond (waypoint_follower) to lifecycle manager.
- [INFO] [1755086788.408424190]: [lifecycle_manager_navigation]: Server waypoint_follower connected with bond.
- [INFO] [1755086788.408521816]: [lifecycle_manager_navigation]: Activating velocity_smoothening
- [INFO] [1755086788.409105867]: [velocity_smoothening]: Activating
- [INFO] [1755086788.409223140]: [velocity_smoothening]: Creating bond (velocity_smoothening) to lifecycle manager.
- [INFO] [1755086788.523654885]: [lifecycle_manager_navigation]: Server velocity_smoothening connected with bond.
- [INFO] [1755086788.523710383]: [lifecycle_manager_navigation]: Managed nodes are active

The mobile app will display the following image. Enter the car's IP address, using [zh] for Chinese and [en] for English. Select ROS2 and select /camera/rgb in the VideoTopic field at the bottom. Finally, click [Connect].

Please enter IP address

192.168.11.203

ROS1

ROS2

Video Topic

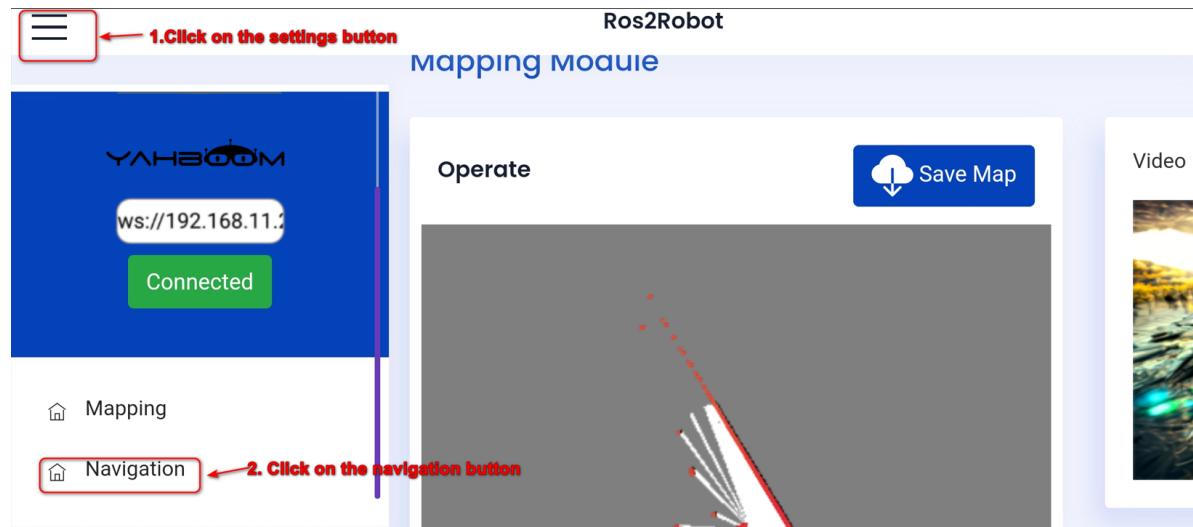
/usb_cam/image_raw/compressed

/camera/rgb/image_raw/compressed

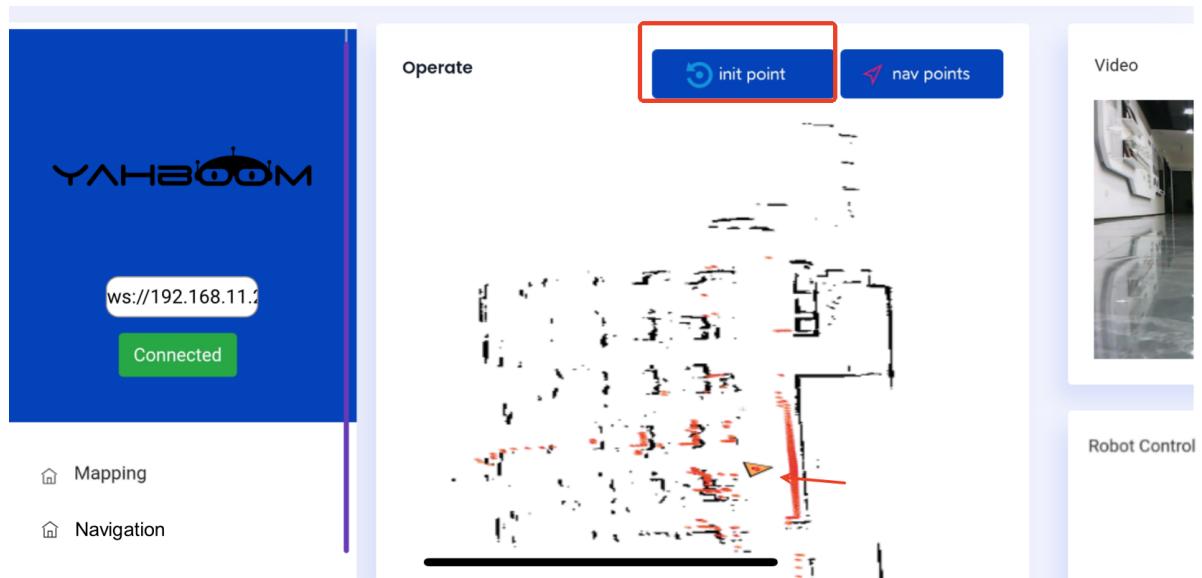
/csi/image_raw/compressed

Connect

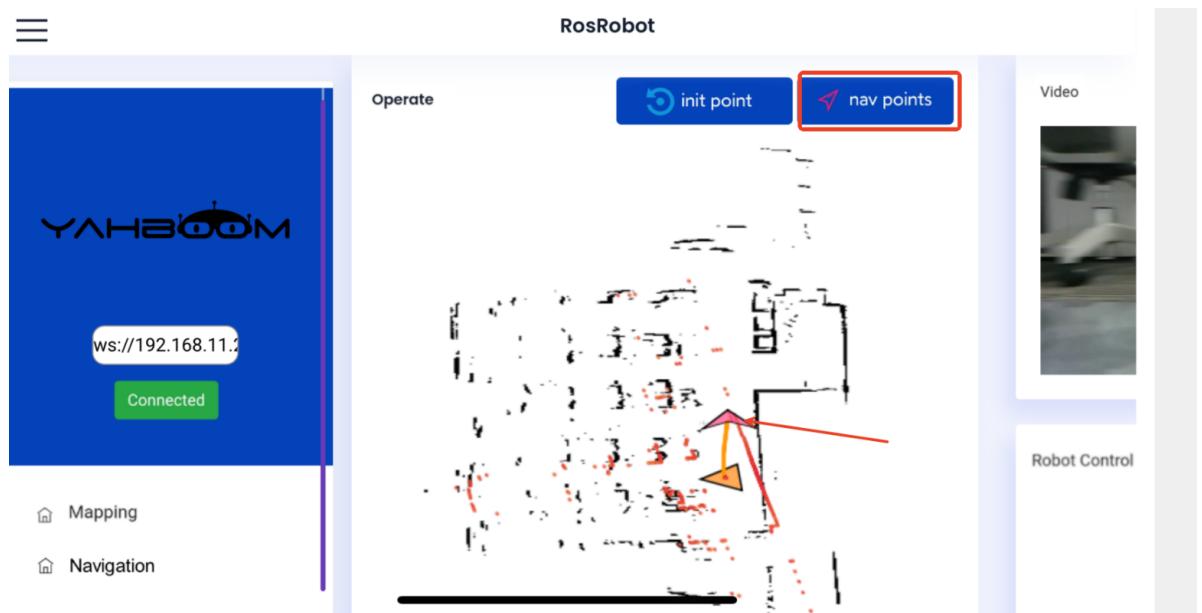
After successfully connecting, select the navigation interface, as shown below.



Then, based on the actual position of the robot, click [Set Initialization Point] to set an initial target point for the robot. If the radar scan area roughly overlaps with the actual obstacle, the position is accurate. As shown in the image below,



Then, click [Set Navigation Point] and give the car a destination. The car will then map a route and follow it to the destination. As shown in the image below,



3.2. Instruction Analysis

Code Path:

```
~/yahboomcar_ros2_ws/yahboomcar_ws/src/yahboomcar_nav/launch/navigation_teb_app_launch.xml
```

Using the `navigation_teb_app_launch.xml` file as an example, we will explain which launch files are launched. The `navigation_teb_app_launch.xml` file is as follows:

```
<launch>
  <include file="$(find-pkg-share
rosbridge_server)/launch/rosbridge_websocket_launch.xml"/>
  <node name="laserscan_to_point_publisher" pkg="laserscan_to_point_publisher"
exec="laserscan_to_point_publish"/>
  <include file="$(find-pkg-share
yahboomcar_nav)/launch/navigation_teb_launch.py"/>
  <include file="$(find-pkg-share
robot_pose_publisher_ros2)/launch/robot_pose_publisher_launch.py"/>
</launch>
```

rosbridge_websocket_launch.xml: Launches the rosbridge WebSocket server for ROS (Robot Operating System)

laserscan_to_point_publisher: Publishes radar point cloud data to the app

robot_pose_publisher_launch.py: Publishes vehicle position information

app_send_goal: Publishes navigation target topic to the app

navigation_teb_launch.py: Navigation-related programs for Navi2