# ROSMASTER R2 (ROS1)

## Run Everything on the Jetson Xavier NX (No VM Required)

Updated: December 12, 2025

This document rewrites the Yahboom “Robot terminal + VM terminal” workflow into a single-machine setup: your Jetson Xavier NX runs the robot, ROS master, and RViz locally.

## Table of contents

## 1) One-time setup (single-machine ROS networking)

On the NX, ROS must point to itself. If you previously followed a multi-machine tutorial, your environment variables may still be pointing at a VM IP.

Edit your **~/.bashrc** and set ROS master + host to localhost:

```
gedit ~/.bashrc

# Add/replace with:
export ROS_MASTER_URI=http://localhost:11311
export ROS_HOSTNAME=localhost
# If your setup uses ROS_IP instead:
# export ROS_IP=localhost
```

```
source ~/.bashrc

echo $ROS_MASTER_URI
echo $ROS_HOSTNAME
echo $ROS_IP
```

## 2) Confirm RViz can open on the NX

RViz is a GUI app. You need a graphical desktop session (HDMI monitor/keyboard, or VNC/remote desktop that provides a real display).

From the NX desktop terminal:

```
echo $DISPLAY
rviz
```

## 3) Core bringup (chassis + handle/joystick)

This starts the chassis driver and the standard control stack Yahboom provides.

Run:

```
roslaunch yahboomcar_bringup bringup.launch
```

```
rostopic list
rostopic echo /cmd_vel
rostopic echo /odom
rostopic echo /tf
```

## 4) LiDAR mapping (2D mapping with gmapping)

This is the common Yahboom pattern: sensor bringup + mapping node + a separate RViz 'view' launch. In Yahboom docs that 'view' is often described as the VM step; here it runs on the NX.

```
# Terminal 1
roslaunch yahboomcar_nav laser_bringup.launch
```

```
# Terminal 2
roslaunch yahboomcar_nav yahboomcar_map.launch use_rviz:=false map_type:=gmapping
```

```
# Terminal 3 (RViz view)
roslaunch yahboomcar_nav view_lidar_mapping.launch
```

```
# Drive slowly while mapping
rosrun teleop_twist_keyboard teleop_twist_keyboard.py
```

```
# Save the map
mkdir -p ~/maps
rosrun map_server map_saver -f ~/maps/my_map
```

## 5) LiDAR navigation (drive on your saved map)

Navigation is: sensor bringup + navigation launch (map + planner) + RViz view.

Replace **my_map** with your saved map name.

```
# Terminal 1
roslaunch yahboomcar_nav laser_bringup.launch
```

```
# Terminal 2
roslaunch yahboomcar_nav yahboomcar_navigation.launch use_rviz:=false map:=my_map
```

```
# Terminal 3 (RViz view)
roslaunch yahboomcar_nav view_navigate.launch
```

## 6) Astra Pro Plus / depth camera mapping + navigation (optional)

If you use an Orbbec Astra Pro Plus, Yahboom often provides a dedicated bringup launch that publishes depth data and/or a virtual laser scan.

```
# Mapping
# Terminal 1
roslaunch yahboomcar_nav astrapro_bringup.launch
# Terminal 2
roslaunch yahboomcar_nav yahboomcar_map.launch use_rviz:=false map_type:=gmapping
# Terminal 3
roslaunch yahboomcar_nav view_vision_mapping.launch

# Save
mkdir -p ~/maps
rosrun map_server map_saver -f ~/maps/visual_map
```

```
# Navigation
# Terminal 1
roslaunch yahboomcar_nav astrapro_bringup.launch
# Terminal 2
roslaunch yahboomcar_nav yahboomcar_navigation.launch use_rviz:=false map:=visual_map
# Terminal 3
roslaunch yahboomcar_nav view_navigate.launch
```

## 7) Multi-point goals (record goal poses from RViz)

To record goal poses, echo the goal topic and click goals in RViz. Each click prints a pose you can copy into a YAML or a script.

```
rostopic echo /move_base_simple/goal
```

## 8) Voice multi-point navigation (optional)

Run the same navigation stack as Section 5, then start Yahboom's voice goal publisher script (path may vary by image).

```
# Terminal 1
roslaunch yahboomcar_nav laser_bringup.launch
# Terminal 2
roslaunch yahboomcar_nav yahboomcar_navigation.launch use_rviz:=false map:=my_map
# Terminal 3
roslaunch yahboomcar_nav view_navigate.launch
# Terminal 4
python ~/yahboomcar_ws/src/yahboomcar_voice_ctrl/scripts/voice_Ctrl_send_mark.py
```

## 9) ORB-SLAM2 + Octomap (optional 3D mapping)

If your launch file supports **use_rviz**, set it to true to keep everything on the NX.

```
roslaunch yahboomcar_slam robot_orb_octomap.launch frame_id:=odom use_rviz:=true
```

## 10) Common problems (RViz, TF, map not showing)

**RViz opens but shows nothing:** verify topics and transforms exist.

**‘No transform’ errors:** usually means bringup/mapping/navigation isn’t running, or the wrong Fixed Frame is selected in RViz.

**Quick checks:**

```
rostopic list
rostopic echo /tf
rostopic echo /scan
rostopic echo /map
```

```
rosrun tf view_frames
evince frames.pdf
```

## 11) Switching back to VM visualization (optional)

If you later want RViz on a PC/VM, point both machines at the NX as ROS master. Replace IPs with your real addresses.

```
# NX ~/.bashrc
export ROS_MASTER_URI=http://192.168.68.51:11311
export ROS_HOSTNAME=192.168.68.51

# VM ~/.bashrc
export ROS_MASTER_URI=http://192.168.68.51:11311
export ROS_HOSTNAME=192.168.68.100

# Test
# NX: roscore
# VM: rviz
```