

Using ROS demo

Pre-launch Check

“ **Tip** Please first acquire basic knowledge of `ROS`. You can refer to the [ROS1 Tutorial](#) and [ROS2 Tutorial](#) to understand the common conventions of the `ROS` system.

Please ensure the following are completed:

1. ROS-SDK installation is complete; if it is your first time installing the SDK, it is recommended to **restart** after the first installation.
2. Compilation has been completed without errors.
3. The workspace has been sourced.
4. If using the CAN-COM HUB, ensure the CAN bus is correctly connected and devices can be detected.
 - Use `lsusb` in the terminal to see the following devices.
5. Ensure the remote control is turned off; or if it is on, ensure it is set to CAN mode.

“ For information about the remote control and modes, please refer to the remote control section in the product user manual.

Chassis Devices

“ **Warning** Be cautious, when attempting to control the robot with software for the first time, ensure there are no obstacles within the robot's range of motion, or elevate the robot.

There are two test demos for the chassis:

```
* demo_basic_ctrl.launch
```

```
* demo_key_ctrl.launch
```

`demo_basic_ctrl.launch` is a standard launch file that can be used directly with autonomous driving, controlling the robot's movement via the `/cmd_vel` topic.

`demo_key_ctrl.launch` supports keyboard control, using `turtle_teleop_key` to control the robot with the keyboard. If you have not yet seen the ROS `turtlesim` tutorial, please check it out first.

“ **Tip** Note that the movement is not controlled with the arrow keys, please check the usage tips printed in the terminal and press the forward button indicated.

Successful launch will display the following terminal message

```
[ INFO] [1714662015.789025136]: xnode_vehicle: Vehicle online
[ INFO] [1714662015.789215455]: xnode_vehicle: CAN mode locked
[ INFO] [1714662016.189858534]: xnode_vehicle: mode OK(CAN mode)
[ INFO] [1714662016.789707014]: xnode_vehicle: Vehicle enable
[ INFO] [1714662016.789779160]: xnode_vehicle: Device type = MARK1_DIFF
[ INFO] [1714662016.789801928]: xnode_vehicle: ### Vehicle init finish ###
[ INFO] [1714662017.290447759]: xnode_vehicle: cal_mode = odom ||| model = Di
fferential
[ INFO] [1714662017.290510966]: xnode_vehicle: track_width = 0.520m ||| wheel
_base = 0.410m
[ INFO] [1714662017.290542395]: xnode_vehicle: ### Odom init finish ###
[ INFO] [1714662017.290570052]: xnode_vehicle: ===== VEHICLE READY TO G
O =====
```

For parameter settings and communication `msg` in the launch file, please refer to the `src/drivers/xpkg_vehicle/README.md` file, below are explanations of a few key parameters:

- `calc_speed` sets the odometer calculation method, `true` for using speed, `false` for using motor encoder values, with the default setting recommended as false.
- `mode_can_lock` is the CAN mode lock setting, `true` forces CAN mode, switching to other modes with the remote control will trigger a warning and wait for recovery, setting `false` allows the user to choose the control mode.

!!!note If the chassis cannot be controlled, it may be because `mode_can_lock` is set to `false`, and it has been switched to another mode.

- `rate_x` controls the rate multiplier for input speed, 1 is full speed, it is recommended to set it below 0.2 during testing.

After successfully launching the package, you can view ROS topics through `rostopic list`, below are a few key topics:

- The chassis node sends messages of type `nav_msgs/Odometry` to the topic `/odom`. This message represents the chassis odometer information, including the current pose of the chassis as well as linear and angular velocity information.
 - The chassis node sends messages of type `tf2_msgs/TFMessage` to the topic `/tf`, which include chassis tf transformation information, to be enabled in launch parameters.
 - The chassis node subscribes to messages of type `geometry_msgs/Twist` from the topic `/cmd_vel`. Upon receiving this message, the chassis moves, where `linear.x`/`linear.y` represent linear velocity, and `angular.z` represents rotational speed.
-

#4

□ Yangfan He ## 12 ## 2024 11:05:41

□ Yangfan He ## 12 ## 2024 11:58:13