Gazebo 仿真

```
# 启动 Gazebo 仿真 TurtleBot
roslaunch turtlebot_gazebo turtlebot_empty_world.launch
# 使用键盘控制机器人移动
roslaunch turtlebot_teleop keyboard_teleop.launch
# 启动 rviz
roslaunch turtlebot_rviz_launchers view_robot.launch
```

catkin 工程

```
# 重新编译程序后 (即运行 catkin_make 后) , 需要运行下面的命令
# make sure you have sourced your workspace's setup.sh file after calling catkin_make
# but before trying to use your applications
source ~/catkin_ws/devel/setup.bash

# use the catkin_create_pkg script to create a new package called 'beginner_tutorials'
# which depends on std_msgs, roscpp, and rospy
cd ~/catkin_ws/src
catkin_create_pkg beginner_tutorials std_msgs rospy roscpp
# catkin_create_pkg <package_name> [depend1] [depend2] [depend3]

# first-order dependencies
rospack depends1 beginner_tutorials
# all nested dependencies (include ‘indirect dependencies’)
rospack depends beginner_tutorials

# Building Packages
cd ~/catkin_ws/
catkin_make
```

常用命令

```
rosnode list
rosnode info /rosout
rosnode cleanup

rosrun [package_name] [node_name]
# use a Remapping Argument to change the node's name
rosrun turtlesim turtlesim_node __name:=my_turtle

rostopic list
rostopic list -v
```

rostopic echo shows the data published on a topic
rostopic echo [topic]
rostopic type returns the message type of any topic being published
rostopic type [topic]
rosmsg show [message type]

rostopic pub publishes data on to a topic currently advertised
rostopic pub [topic] [msg_type] [args]

rqt_graph shows the nodes and topics currently running
rosrun rqt_graph rqt_graph
rqt_plot displays a scrolling time plot of the data published on topics
rosrun rqt_plot rqt_plot

rqt_console attaches to ROS's logging framework to display output from nodes
rosrun rqt_console rqt_console
rqt_logger_level allows us to change the verbosity level (DEBUG, WARN, INFO,
and ERROR) of nodes as they run
rosrun rqt_logger_level rqt_logger_level

rosrun rviz rviz

roslaunch starts nodes as defined in a launch file
roslaunch [package] [filename.launch]

TurtleBot 机器人

ssh turtlebot@192.168.1.104 在workstation上使用ssh连接turtlebot
ssh turtlebot@IP_OF_TURTLEBOT 在workstation上使用ssh连接turtlebot

roslaunch turtlebot_bringup minimal.launch 启动 ROS (在turtlebot上运行)

roslaunch turtlebot_bringup 3dsensor.launch 启动 kinect (在turtlebot上运行)

roslaunch turtlebot_teleop keyboard_teleop.launch 启动键盘遥控

rqt -s kobuki_dashboard 启动 dashboard
roslaunch turtlebot_dashboard turtlebot_dashboard.launch 启动 dashboard

roslaunch turtlebot_rviz_launchers view_robot.launch 启动对turtlebot已配置好的rviz

rosrun image_view image_view image:=/camera/rgb/image_color 显示kinect采集的rgb图像

~/.bashrc 配置文件

```
# 编辑修改~/.bashrc
sudo gedit ~/.bashrc

# 文件 ~/.bashrc 中的相关内容
source /opt/ros/hydro/setup.bash                在shell中允许使用ros相关命令
# export EDITOR='emacs -nw'             定义默认的文本编辑器
export ROS_PACKAGE_PATH=$ROS_PACKAGE_PATH:~/catkin_ws/      工作空间
export IP_OF_TURTLEBOT="192.168.1.104"
export IP_OF_WORKSTATION="192.168.1.105"
# export ROS_MASTER_URI=http://$IP_OF_WORKSTATION:11311    用于本机直接连接底盘
export ROS_MASTER_URI=http://$IP_OF_TURTLEBOT:11311      用于本机作为workstation
export ROS_HOSTNAME=$IP_OF_WORKSTATION
```

SLAM Map Building with TurtleBot (Demo)

```
On the TurtleBot
   Bring up the robot
      $ roslaunch turtlebot_bringup minimal.launch
   Run the gmapping demo app
      $ roslaunch turtlebot_navigation gmapping_demo.launch
On the Workstation
   Launch rviz
      $ roslaunch turtlebot_rviz_launchers view_navigation.launch
On the TurtleBot
   Drive the robot around by teleop (keyboard)
   Save the map to file
      $ rosrun map_server map_saver -f /tmp/my_map
   Note: Do not close the gmapping launch until saving the map.
```