

無人載具技術與應用

ROS-Ubuntu

徐瑋隆

wlhsu304@gmail.com

ROS/ Installation

Available Translations: [German](#) | [Spanish](#) | [French](#) | [Italian](#) | [Japanese](#) | [Korean](#) | [Brazilian Portuguese](#) | [Portuguese](#) | [Русский \(Russian\)](#) | [Thai](#) | [Turkish](#) | [简体中文](#) | [Ukrainian](#) | [Vietnamese](#)

1. See Also:

1. [ROS/Installation](#) (this page)
2. [Distributions](#)
3. [Installation](#)

ROS Installation Options

There is more than one ROS distribution supported at a time. Some are older releases with long term support, making them more stable, while others are newer with shorter support life times, but with binaries for more recent platforms and more recent versions of the ROS packages that make them up. See the [Distributions](#) page for more details. We recommend one of the versions below:

ROS Melodic Morenia

Released May, 2018

LTS, supported until May, 2023

Recommended for Ubuntu 18.04



ROS Noetic Ninjemy

Released May, 2020

Latest LTS, supported until May, 2025

Recommended for Ubuntu 20.04



ROS 2 Documentation

The ROS Wiki is for ROS 1. Are you using ROS 2 (Dashing/Foxy/Rolling)? Check out the [ROS 2 Documentation](#)

Wiki

Distributions

ROS/Installation

ROS/Tutorials

RecentChanges

Page

Immutable Page

Info

Attachments

More Actions:

User

Login

ROS 2 Installation

If you're looking for ROS 2 installation pages please see [docs.ros.org](#)

There are instructions for [Foxy](#), [Galactic](#), and [Rolling](#).

[melodic/ Installation](#)

ROS Melodic installation instructions

These instructions will install the **ROS Melodic Morenia** distribution, which is available for Ubuntu Artful (17.10), Bionic (18.04 LTS) and Debian Stretch, among other platform options.

To install our previous long-term support release, **ROS Kinetic Kame**, please see the [Kinetic installation](#) instructions.

The links below contain instructions for installing **ROS Melodic Morenia** on various operating systems.

Select Your Platform

Supported:



Ubuntu Artful amd64
Bionic amd64 armhf arm64



Debian Stretch amd64 arm64



Windows 10 amd64

[Source installation](#)

Experimental:



Arch Linux Any amd64 i686 arm armv6h armv7h aarch64



Gentoo

[ROS 2 Documentation](#)

The ROS Wiki is for ROS 1. Are you using ROS 2 (Dashing/Foxy/Rolling)? [Check out the ROS 2 Documentation](#)

Wiki

Distributions

ROS/Installation

ROS/Tutorials

RecentChanges

[melodic/Installation](#)

Page

[Immutable Page](#)

Info

Attachments

More Actions:

[Raw Text](#)

User

[Login](#)

3. 创建ROS工作空间

catkin rosbuild

这些操作方法只适用于ROS Groovy及后期版本，对于ROS Fuerte及早期版本请选择rosbuild。

下面我们开始创建一个[catkin 工作空间](#)：

```
$ mkdir -p ~/catkin_ws/src  
$ cd ~/catkin_ws/src
```

即使这个工作空间是空的（在'src'目录中没有任何软件包，只有一个CMakeLists.txt链接文件），你依然可以编译它：

```
$ cd ~/catkin_ws/  
$ catkin_make
```

[catkin_make](#)命令在[catkin 工作空间](#)中是一个非常方便的工具。如果你查看一下当前目录应该能看到'build'和'devel'这两个文件夹。在'devel'文件夹里面你可以看到几个setup.*sh文件。[source](#)这些文件中的任何一个都可以将当前工作空间设置在ROS工作环境的最顶层，想了解更多请参考[catkin](#)文档。接下来首先[source](#)一下新生成的setup.*sh文件：

```
$ source devel/setup.bash
```

要想保证工作空间已配置正确需确保ROS_PACKAGE_PATH环境变量包含你的工作空间目录，采用以下命令查看：

```
$ echo $ROS_PACKAGE_PATH  
/home/<youruser>/catkin_ws/src:/opt/ros/indigo/share:/opt/ros/indigo/stacks
```

到此你的工作环境已经搭建完成，接下来可以继续学习 [ROS文件系统教程](#).

- `mkdir -p ~/catkin_ws/src`
 - `cd catkin_ws/`
 - `catkin_make`
-
- `source /opt/ros/noetic/setup.bash`
 - `source ~/catkin_ws/devel/setup.bash`

Terminal

peter@ubuntu: ~/catkin_ws/devel

peter@ubuntu: ~/catki... x

peter@ubuntu: ~/catki... x

peter@ubuntu: ~/catki... x

+ ▾

```
peter@ubuntu:~$ cd ~/catkin_ws/
peter@ubuntu:~/catkin_ws$ cd devel/
peter@ubuntu:~/catkin_ws/devel$ pwd
/home/peter/catkin_ws/devel
peter@ubuntu:~/catkin_ws/devel$ ls
cmake.lock    lib          local_setup.sh  setup.bash  _setup_util.py
env.sh        local_setup.bash  local_setup.zsh  setup.sh   setup.zsh
peter@ubuntu:~/catkin_ws/devel$ gedit ~/.bashrc
```

```
# sleep 10; alert
alias alert='notify-send --urgency=low -i "$( [ $? = 0 ] && echo terminal || "$(_history|tail -n1|sed -e '\''s/^\s*[0-9]\+\s*//;s/[;&]\s*alert$/'\''")"
# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

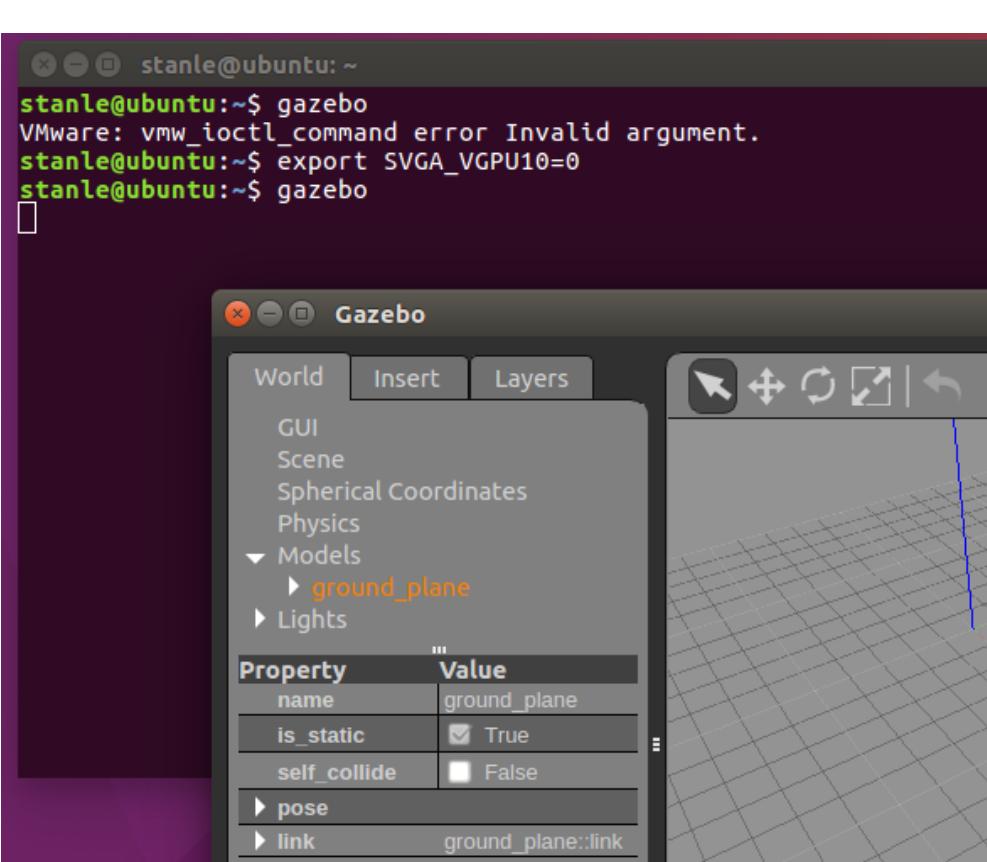
# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
    if [ -f /usr/share/bash-completion/bash_completion ]; then
        . /usr/share/bash-completion/bash_completion
    elif [ -f /etc/bash_completion ]; then
        . /etc/bash_completion
    fi
fi
source /opt/ros/kinetic/setup.bash
source ~/catkin_ws/devel/setup.bash
```

sh ▾ Tab Width: 8 ▾

```
stanle@ubuntu:~$ sudo apt-get install ros-noetic-gazebo*
```

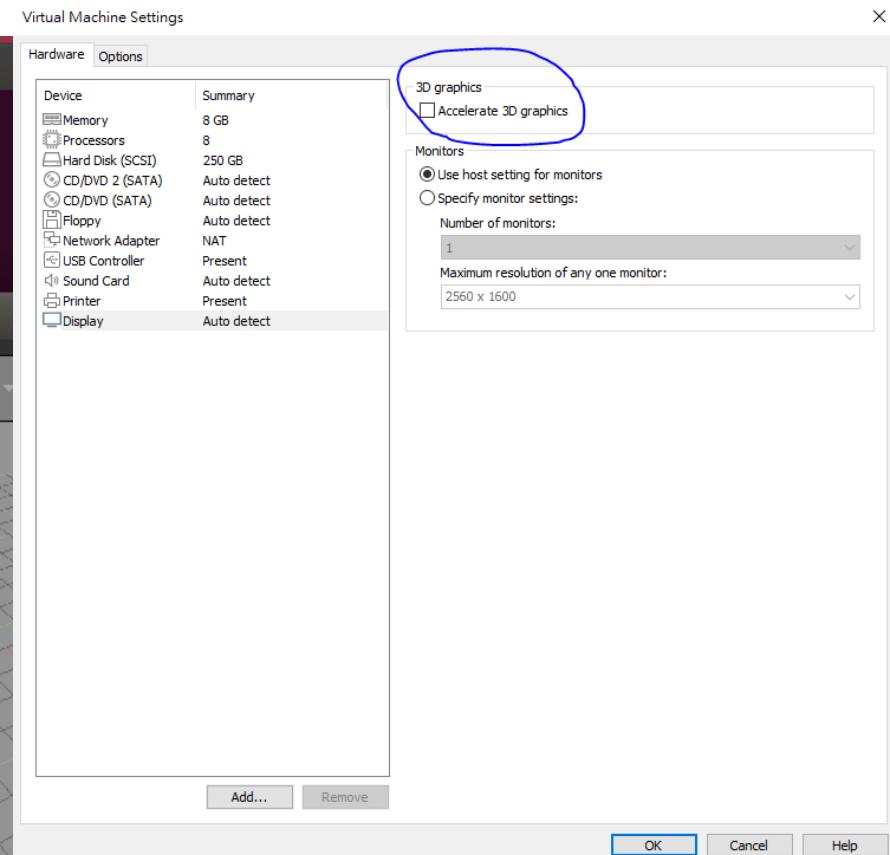
Test that stand-alone Gazebo works

Gazebo

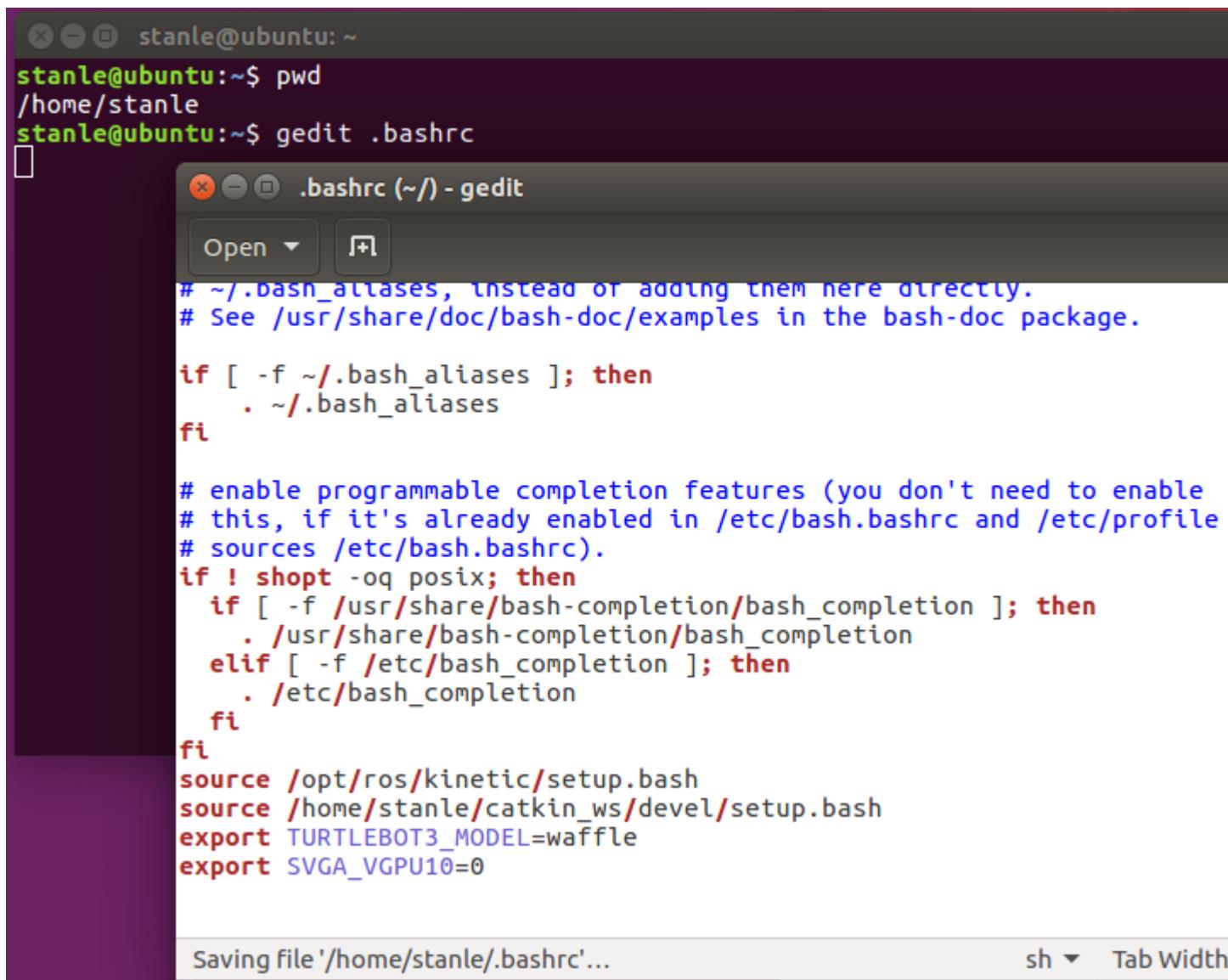


If vmware 無法啟動 gazebo ,

- (1) 关闭虚拟机的3D图形加速 (vm 設定)
\$ export SVGA_VGPU10=0



```
source /opt/ros/noetic/setup.bash  
source ~/catkin_ws/devel/setup.bash  
export TURTLEBOT3_MODEL=waffle  
export SVGA_VGPU10=0
```



The screenshot shows a terminal window titled "stanle@ubuntu: ~" with the command "pwd" outputting "/home/stanle". Below it, the command "gedit .bashrc" is run, opening a Gedit text editor window. The Gedit window has a title ".bashrc (~/) - gedit" and contains the contents of the .bashrc file. The file includes standard bash initialization code, followed by the source commands from the provided text block.

```
# ~/.bash_aliases, instead of adding them here directly.  
# See /usr/share/doc/bash-doc/examples in the bash-doc package.  
  
if [ -f ~/.bash_aliases ]; then  
    . ~/.bash_aliases  
fi  
  
# enable programmable completion features (you don't need to enable  
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile  
# sources /etc/bash.bashrc).  
if ! shopt -q posix; then  
    if [ -f /usr/share/bash-completion/bash_completion ]; then  
        . /usr/share/bash-completion/bash_completion  
    elif [ -f /etc/bash_completion ]; then  
        . /etc/bash_completion  
    fi  
fi  
source /opt/ros/kinetic/setup.bash  
source /home/stanle/catkin_ws/devel/setup.bash  
export TURTLEBOT3_MODEL=waffle  
export SVGA_VGPU10=0
```

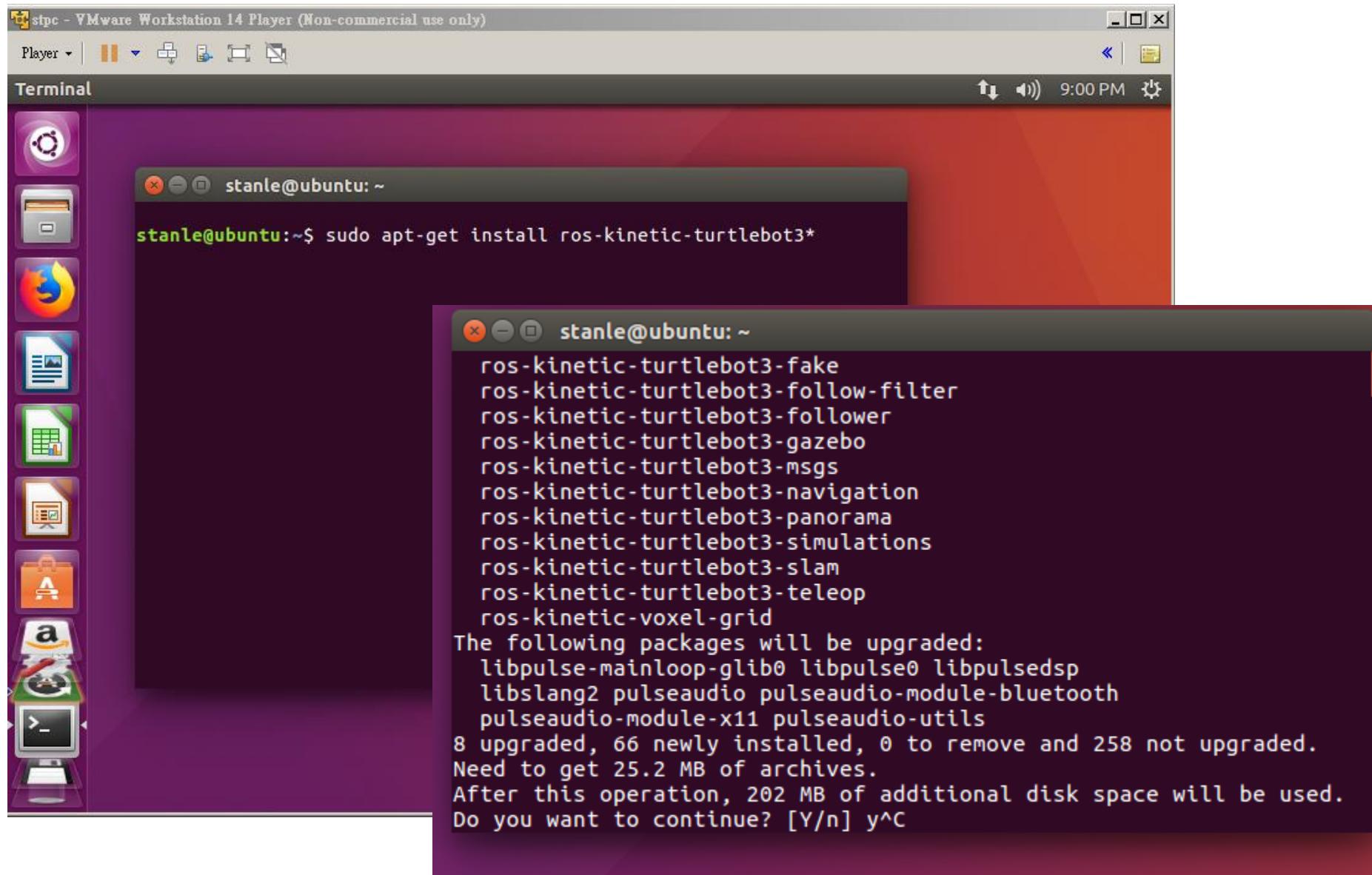
Saving file '/home/stanle/.bashrc'... sh Tab Width:

- source /opt/ros/noetic/setup.bash
 - source ~/catkin_ws/devel/setup.bash
 - export TURTLEBOT3_MODEL=waffle
 - export SVGA_VGPU10=0
-
- export ROS_MASTER_URI="http://192.168.31.188:11311"
 - export ROS_IP=192.168.31.246

```
sudo apt install aptitude
```

```
sudo aptitude search turtlebot3
```

```
sudo apt install ros-noetic-turtlebot3*
```



- \$ sudo apt-get update
- E: 無法將 /var/lib/apt/lists/lock 鎖定 - open
(11: Resource temporarily unavailable)
- E: Unable to lock directory /var/lib/apt/lists/
- sudo rm /var/lib/apt/lists/lock

	Name	Size
Recent	turtlebot3_autorace	1 item
Home	turtlebot3_autorace_camera	6 item
Desktop	turtlebot3_autorace_control	3 item
Documents	turtlebot3_autorace_core	3 item
Downloads	turtlebot3_autorace_detect	6 item
Music	turtlebot3_autorace_bringup	4 item
Pictures	turtlebot3_description	5 item
Videos	turtlebot3_example	6 item
Trash	turtlebot3_fake	4 item
Network	turtlebot3_follower	3 item
Computer	turtlebot3_follow_filter	4 item
Floppy Disk	turtlebot3_gazebo	6 item
Connect to Server	turtlebot3_msgs	3 item
	turtlebot3_navigation	6 item
	turtlebot3_panorama	4 item
	turtlebot3_simulations	1 item
	turtlebot3_slam	6 item
	turtlebot3_teleop	3 item
	turtlesim	5 item

```
stanle@ubuntu: /opt/ros/kinetic/
```

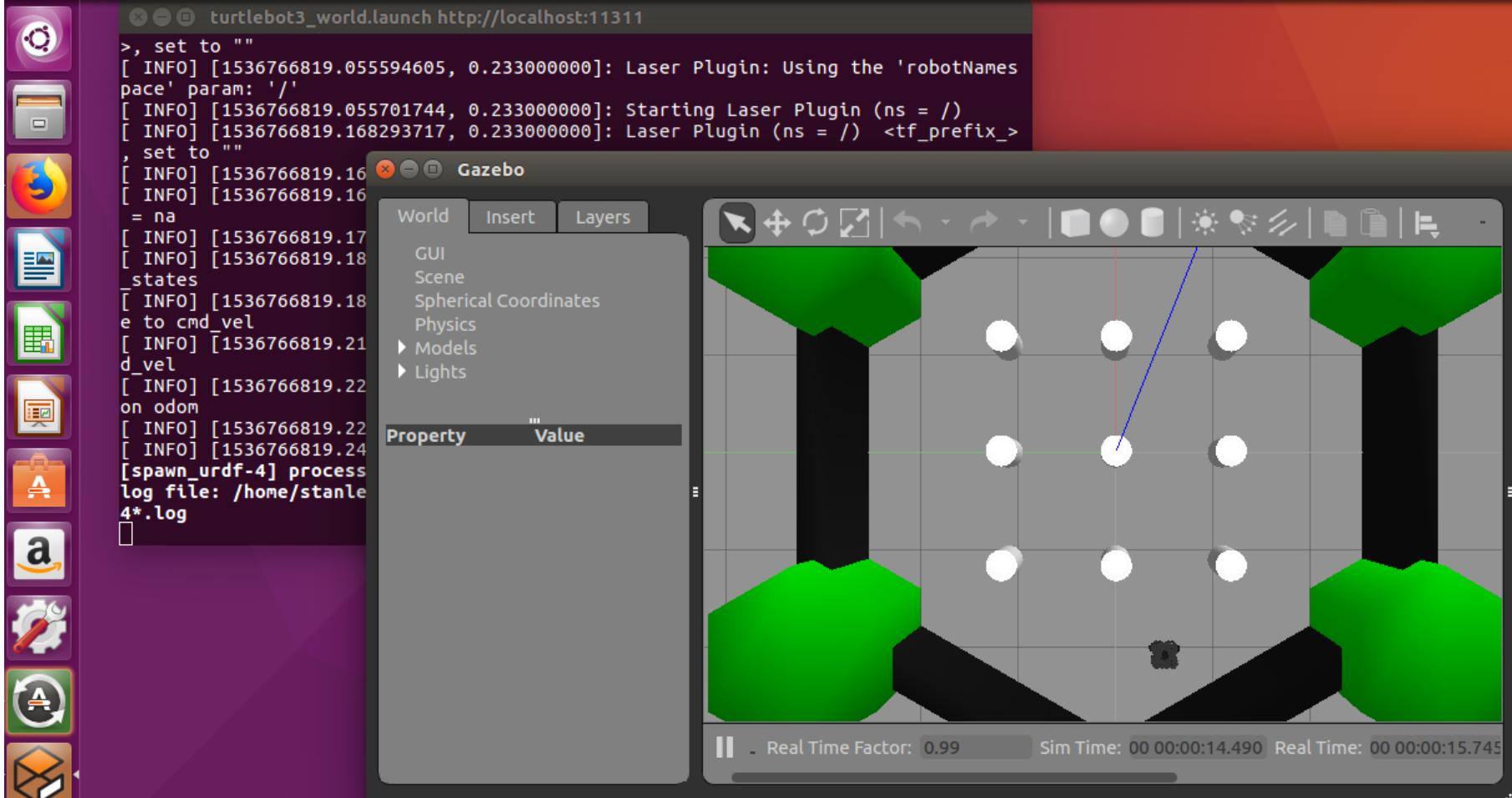
```
turtlebot3
turtlebot3_applications
turtlebot3_applications_msgs
turtlebot3_automatic_parking
turtlebot3_automatic_parking_vision
turtlebot3_autorace
turtlebot3_autorace_camera
turtlebot3_autorace_control
turtlebot3_autorace_core
turtlebot3_autorace_detect
turtlebot3_bringup
turtlebot3_description
turtlebot3_example
turtlebot3_fake
turtlebot3_follower
turtlebot3_follow_filter
turtlebot3_gazebo
turtlebot3_msgs
turtlebot3_navigation
turtlebot3_panorama
turtlebot3_simulations
```

啟動 gazebo turtlebot3 模擬

```
roslaunch turtlebot3_gazebo turtlebot3_world.launch
```

```
stanle@ubuntu:~$ export TURTLEBOT3_MODEL=waffle  
stanle@ubuntu:~$
```

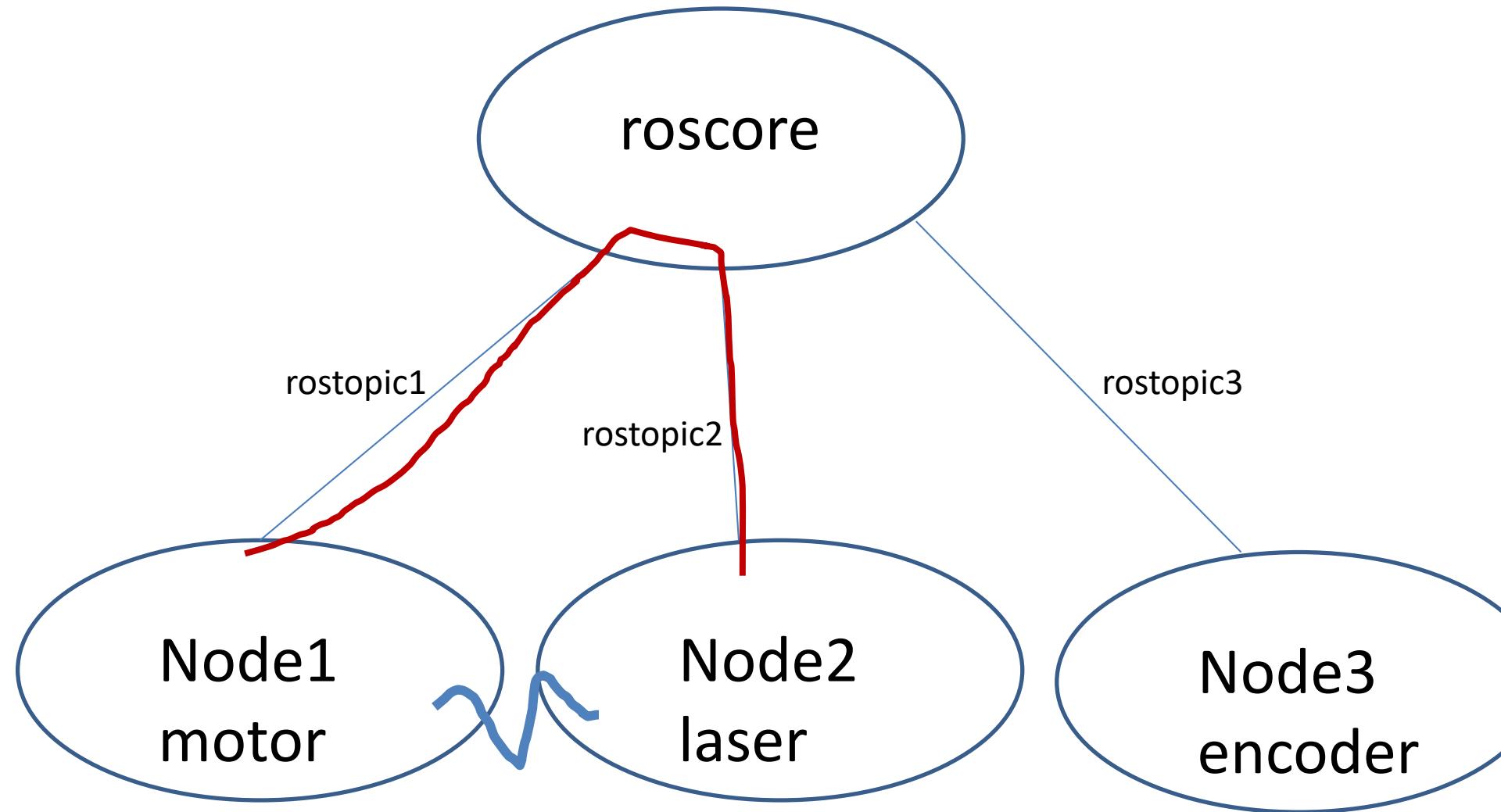
```
stanle@ubuntu: /opt/ros/kinetic/share/turtlebot3_gazebo/launch  
stanle@ubuntu:/opt/ros/kinetic/share/turtlebot3_gazebo/launch$ clear  
  
stanle@ubuntu:/opt/ros/kinetic/share/turtlebot3_gazebo/launch$ ls  
multi_map_merge.launch  
multi_turtlebot3.launch  
multi_turtlebot3_slam.launch  
turtlebot3_autorace.launch  
turtlebot3_autorace_mission.launch  
turtlebot3_empty_world.launch  
turtlebot3_gazebo_rviz.launch  
turtlebot3_house.launch  
turtlebot3_simulation.launch  
turtlebot3_stage_1.launch  
turtlebot3_stage_2.launch  
turtlebot3_stage_3.launch  
turtlebot3_stage_4.launch  
turtlebot3_world.launch  
stanle@ubuntu:/opt/ros/kinetic/share/turtlebot3_gazebo/launch$ roslaunch  
turtlebot3_world.launch
```



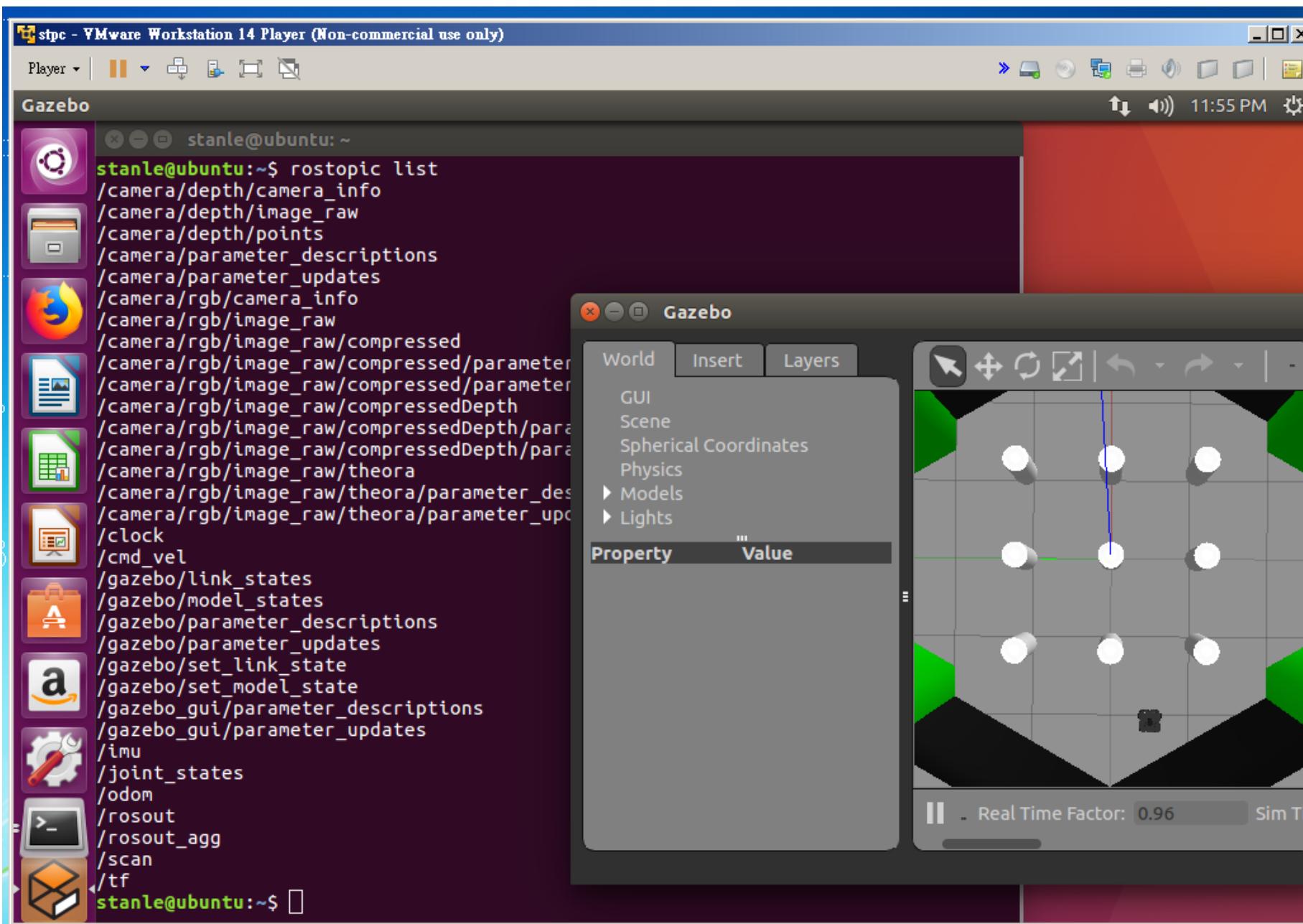
[gazebo-2] process has died 解決方式 (若之前有開 gazebo, 先關閉 gzserver)
> killall gzserver

even if there is no process running and then rosrun package[name] launch file[name].

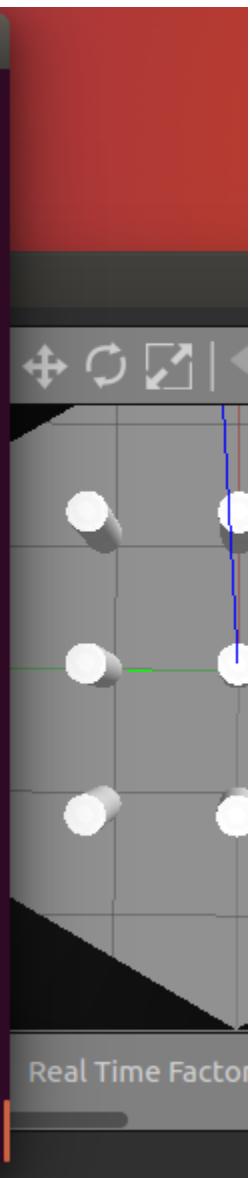
rostopic



rostopic




```
stanle@ubuntu: ~  
stanle@ubuntu:~$ rostopic echo /odom
```



Player | || □ △ □ □

X X X

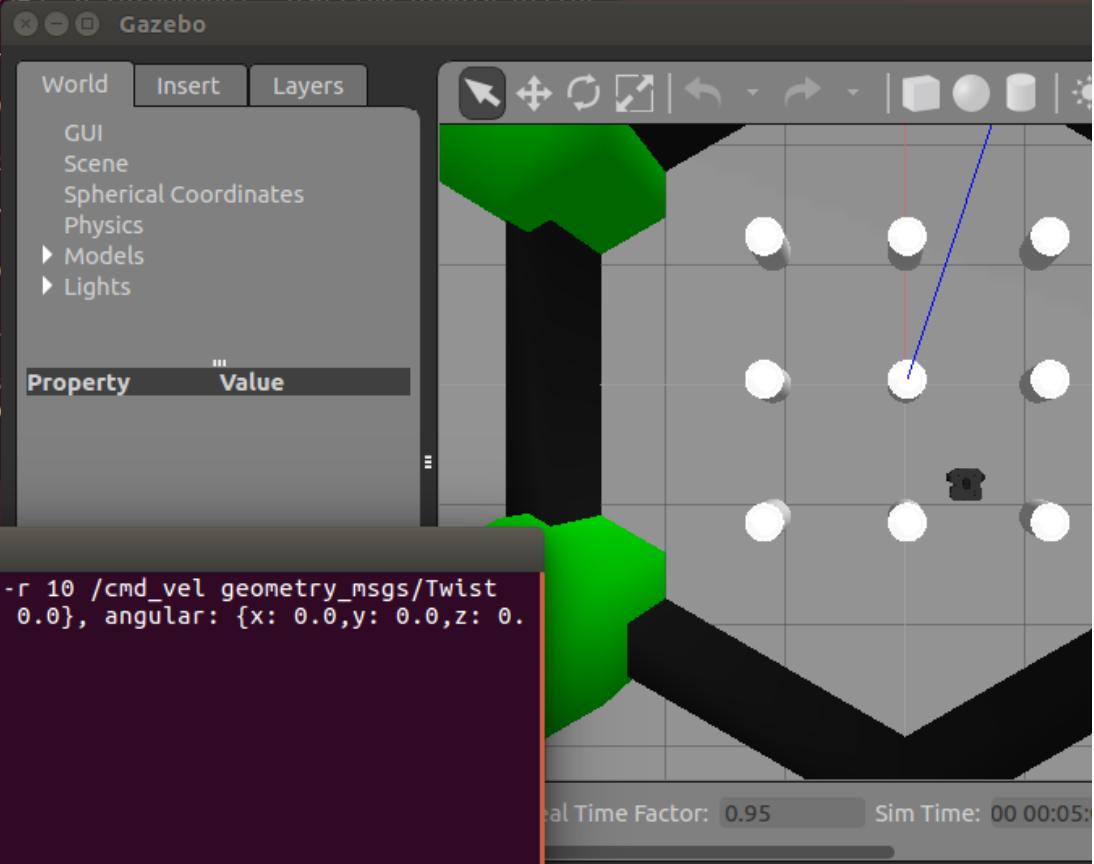
8:43 PM

Terminal



```
turtlebot3_world.launch http://localhost:11311
[ INFO] [1536842262.928974545]: Starting plugin DiffDrive
ive(ns = //)
[ INFO] [1536842262.929224]: osDebugLevel> = na
[ INFO] [1536842262.932349]: f_prefix> =
[ INFO] [1536842262.934442]: vertise joint_states
[ INFO] [1536842262.939407]: y to subscribe to cmd_vel
[ INFO] [1536842262.987939]: bscribe to cmd_vel
[ INFO] [1536842262.990491]: vertise odom on odom
[spawn_urdf-4] process has
log file: /home/stanle/.ro
3/spawn_urdf-4*.log
```

```
stanle@ubuntu:~$ rostopic pub -r 10 /cmd_vel geometry_msgs/Twist
'{linear: {x: 0.1, y: 0.0, z: 0.0}, angular: {x: 0.0,y: 0.0,z: 0.
0}}'
```



```
rostopic pub -r 10 /cmd_vel geometry_msgs/Twist '{linear: {x: 0.1, y: 0.0, z: 0.0}, angular: {x: 0.0,y: 0.0,z: 0.0}}'
```

```
rostopic list
```

```
rostopic echo /scan
```

```
rostopic echo /odom
```

```
rostopic pub -r 1 /cmd_vel  
geometry_msgs/Twist '{linear: {x: 0.1, y:  
0.0, z: 0.0}, angular: {x: 0.0,y: 0.0,z: 0.0}}'
```

Keyboard control

鍵盤控制

[Documentation](#)[Browse Software](#)

teleop_twist_keyboard

[indigo](#)[kinetic](#)[lunar](#)[melodic](#)[Show EOL distros:](#) [Documentation Status](#)

Package Summary

Released

Continuous Integration

Documented

Generic keyboard teleop for twist robots.

- Maintainer status: maintained
- Maintainer: Austin Hendrix <namniart AT gmail DOT com>
- Author: Graylin Trevor Jay
- License: BSD
- Source: git https://github.com/ros-teleop/teleop_twist_keyboard.git (branch: master)

[Github link](#)

1. Installing

```
sudo apt-get install ros-noetic-teleop-twist-keyboard
```

2. Running

```
rosrun teleop_twist_keyboard teleop_twist_keyboard.py
```

3. Controls

See the on-screen instructions:

```
Reading from the keyboard and Publishing to Twist!
-----
Moving around:
  u    i    o
  j    k    l
  m    ,    .

q/z : increase/decrease max speeds by 10%
w/x : increase/decrease only linear speed by 10%
e/c : increase/decrease only angular speed by 10%
anything else : stop

CTRL-C to quit
```

可以直接安裝使用

Github git clone source code 自編譯

The screenshot shows a GitHub repository page for 'ros-teleop/teleop_twist_keyboard'. At the top, there's a navigation bar with links for Features, Business, Explore, Marketplace, Pricing, and a search bar. Below the header, the repository name 'ros-teleop / teleop_twist_keyboard' is displayed, along with metrics: 7 Watch, 32 Star, 128 Fork, and 128 Insights. A prominent 'Join GitHub today' banner is visible. The main content area shows 'Generic Keyboard Teleop for ROS' with 25 commits, 1 branch, 3 releases, and 8 contributors. A pull request from 'trainman419' is listed. On the right side, there's a 'Clone with HTTPS' button, which is highlighted with a blue box. Other options include 'Find file', 'Open in Desktop', and 'Download ZIP'. The repository's README.md file is also partially visible.

The image shows a Linux desktop environment with several windows open. A terminal window is active in the foreground, displaying the command:

```
stanle@ubuntu:~/catkin_ws/src$ git clone https://github.com/rosteleop/teleop_twist_keyboard.git
```

A context menu is open over the terminal window, with the "Paste" option highlighted by a red box. The menu also includes "Open Terminal", "Copy", "Profiles", "Read-Only", and "Show Menubar".

Below the terminal window, another terminal window shows the output of the git clone command:

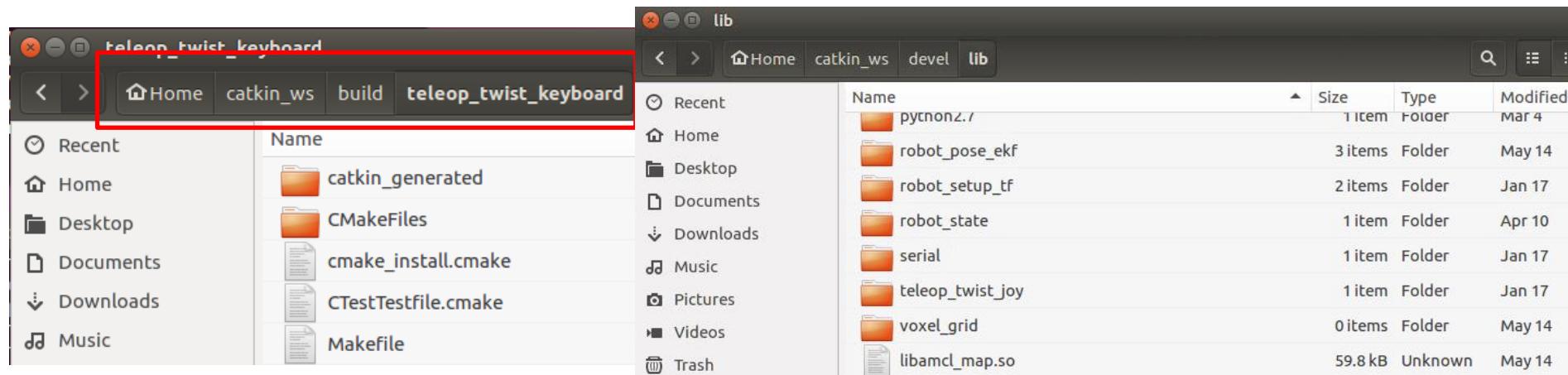
```
stanle@ubuntu:~/catkin_ws/src$ git clone https://github.com/rosteleop/teleop_twist_keyboard.git
Cloning into 'teleop_twist_keyboard'...
remote: Counting objects: 71, done.
remote: Total 71 (delta 0), reused 0 (delta 0), pack-reused 71
Unpacking objects: 100% (71/71), done.
Checking connectivity... done.
stanle@ubuntu:~/catkin_ws/src$
```

At the bottom of the image, a file listing command is shown:

```
stanle@ubuntu:~/catkin_ws/src$ ls
action_tutorials           navigation
CMakeLists.txt              serial
executive_smach             teleop_twist_joy
executive_smach_tutorials   teleop_twist_keyboard
stanle@ubuntu:~/catkin_ws/src$
```

```
stanle@ubuntu: ~/catkin_ws  
stanle@ubuntu:~/catkin_ws/src$ cd /home/stanle  
stanle@ubuntu:~$ cd catkin_ws/  
stanle@ubuntu:~/catkin_ws$ catkin_make
```

```
stanle@ubuntu: ~/catkin_ws  
[ 63%] Built target action_tutorials_generate_messages_py  
Scanning dependencies of target actionlib_msgs_generate_messages_lisp  
[ 63%] Built target actionlib_msgs_generate_messages_lisp  
[ 73%] Built target action_tutorials_generate_messages_lisp  
Scanning dependencies of target actionlib_msgs_generate_messages_nodejs  
[ 73%] Built target actionlib_msgs_generate_messages_nodejs  
[ 83%] Built target action_tutorials_generate_messages_nodejs  
[ 83%] Built target action_tutorials_generate_messages  
[ 86%] Built target DoDishes_client  
[ 88%] Built target cmovebase_server  
[ 91%] Built target DoDishes_server  
[ 94%] Built target cmovebase_client  
[ 97%] Built target teleop_twist_joy  
[100%] Built target teleop_twist_joy_node  
stanle@ubuntu:~/catkin_ws$
```



```
stanle@ubuntu: ~/catkin_ws
```

```
stanle@ubuntu:~/catkin_ws$ rosrun teleop_twist_k  
eyboard teleop_twist_keyboard.py
```

```
stanle@ubuntu: ~/catkin_ws
```

```
stanle@ubuntu:~/catkin_ws$ rosrun teleop_twist_keyboard te  
leop_twist_keyboard.py
```

```
Reading from the keyboard and Publishing to Twist!
```

```
-----  
Moving around:
```

```
 u   i   o  
 j   k   l  
 m   ,   .
```

```
For Holonomic mode (strafing), hold down the shift key:
```

```
-----  
 U   I   O  
 J   K   L  
 M   <   >
```

```
t : up (+z)
```

```
b : down (-z)
```

```
anything else : stop
```

```
q/z : increase/decrease max speeds by 10%
```

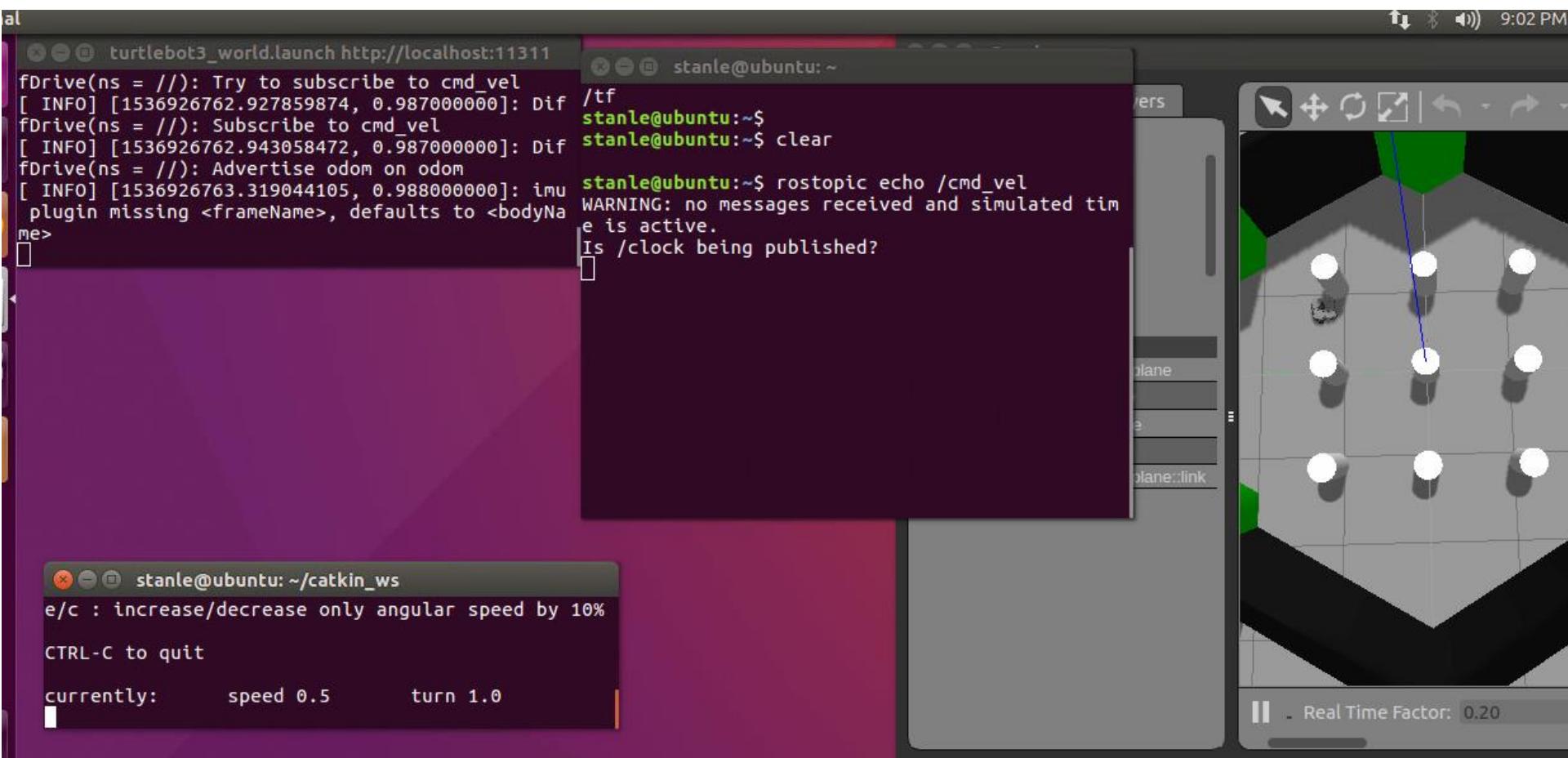
```
w/x : increase/decrease only linear speed by 10%
```

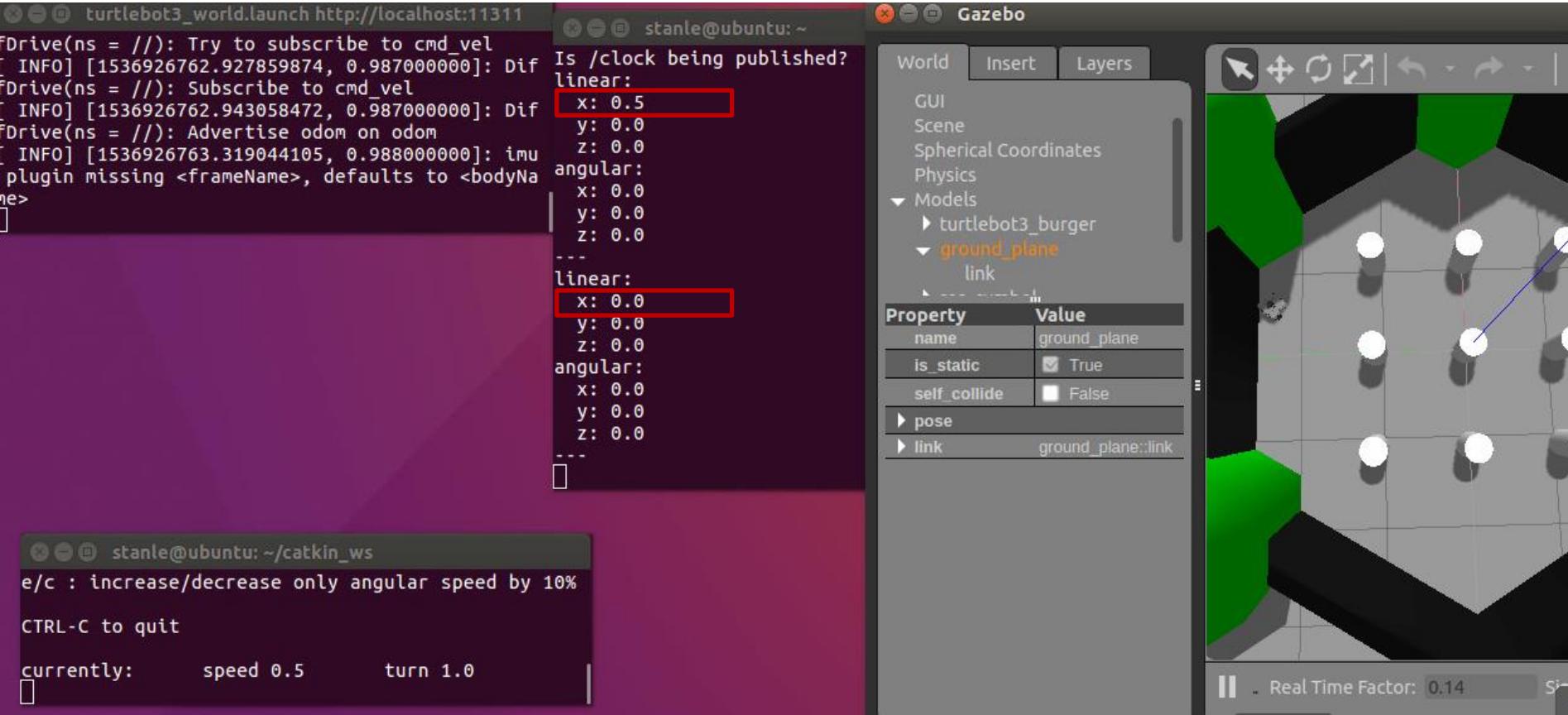
```
e/c : increase/decrease only angular speed by 10%
```

```
CTRL-C to quit
```

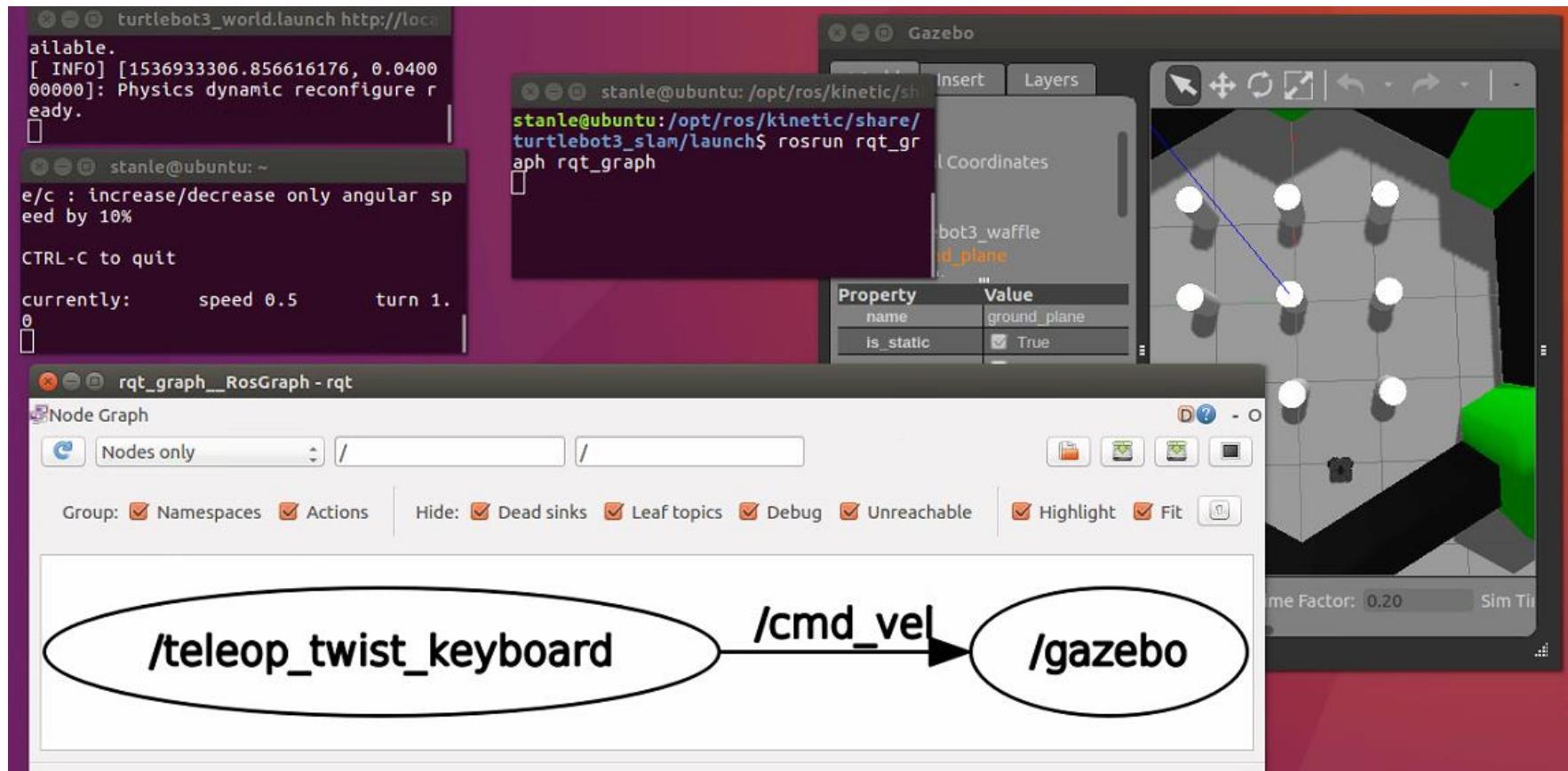
```
currently:      speed 0.5      turn 1.0
```

9:02 PM

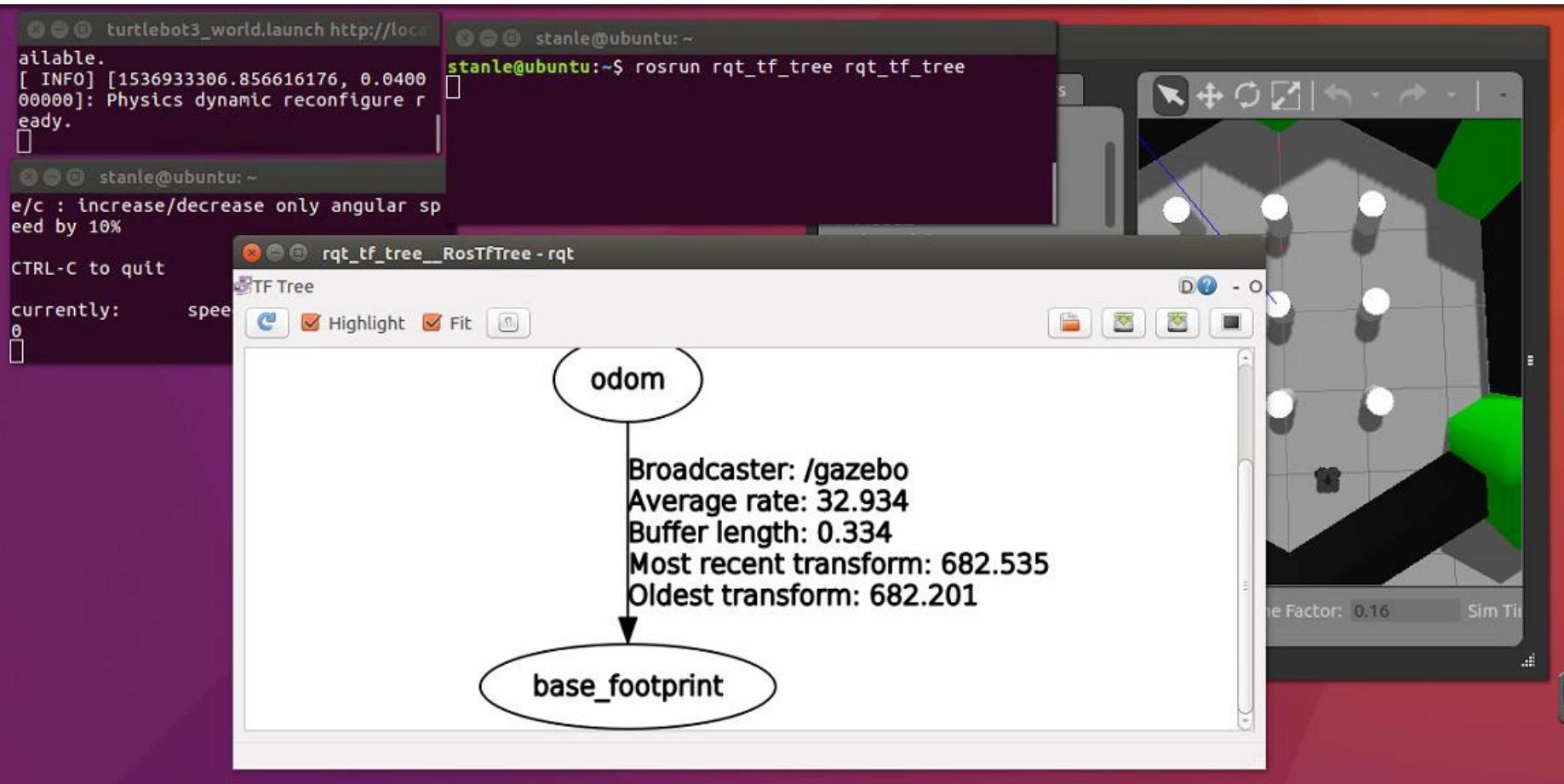




rqt_graph

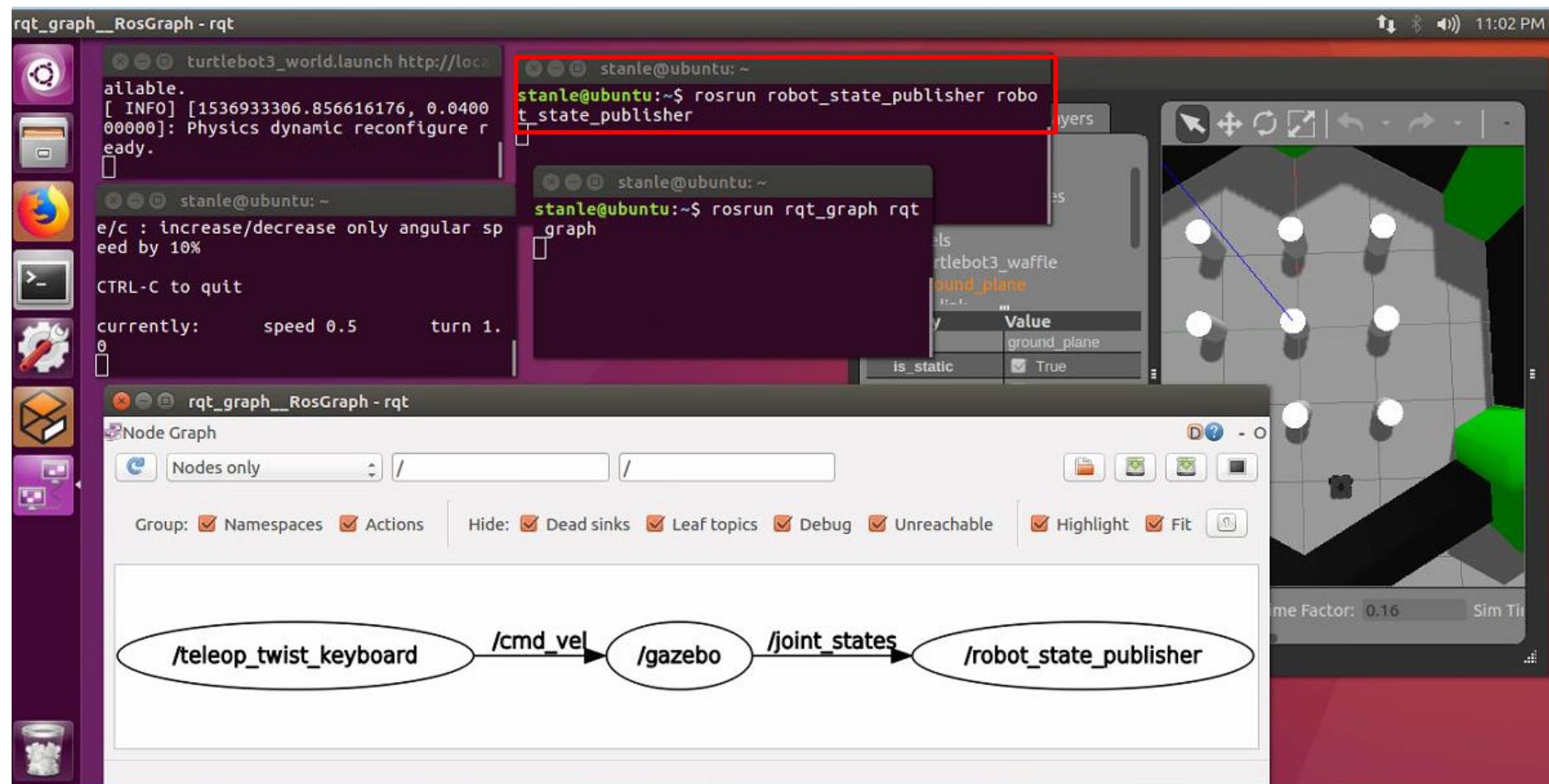


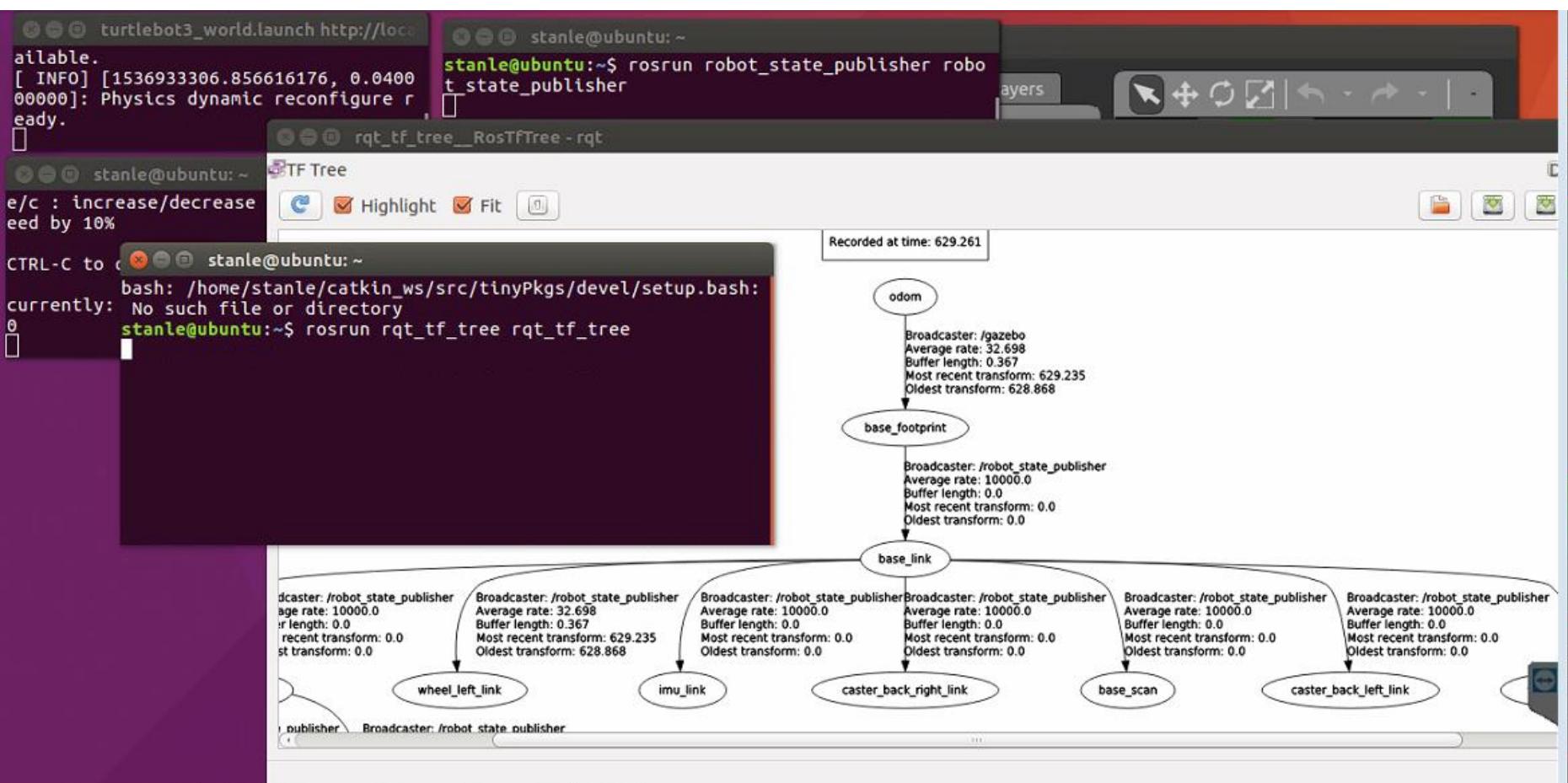
```
rosrun rqt_tf_tree rqt_tf_tree
```



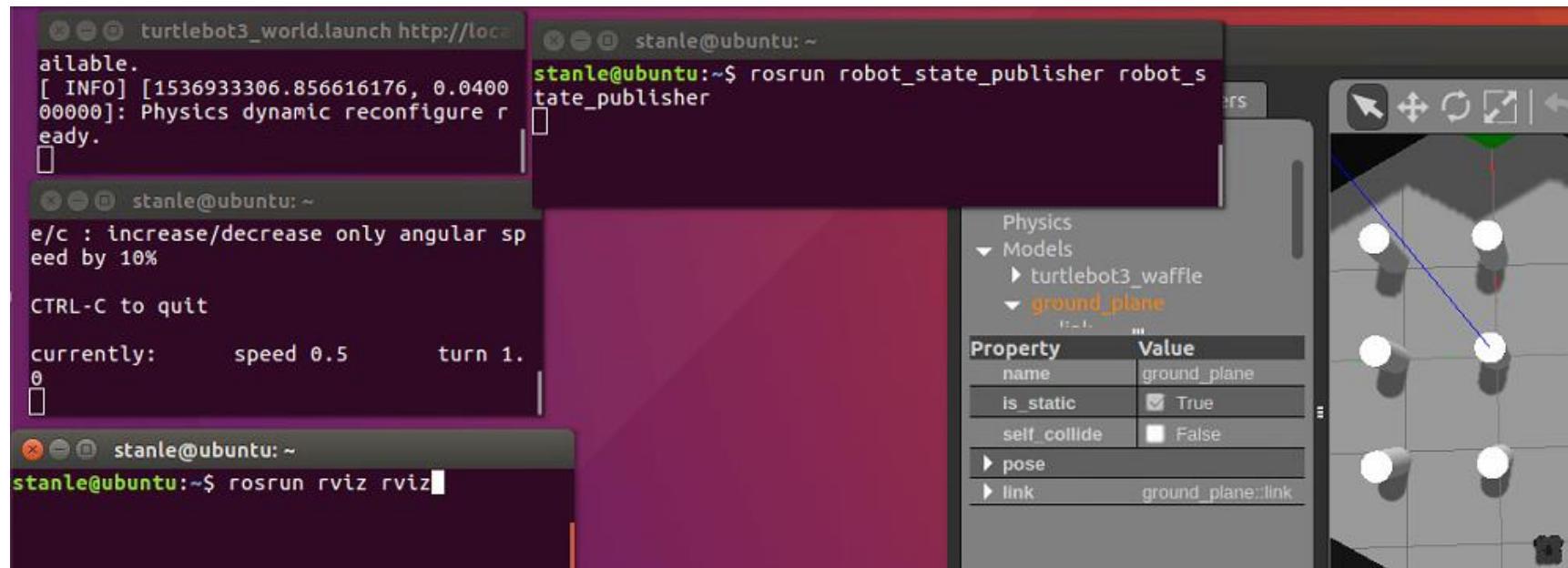
```
roslaunch turtlebot3_gazebo turtlebot3_world.launch  
rosrun robot_state_publisher robot_state_publisher  
Rviz  
rosrun rqt_tf_tree rqt_tf_tree
```

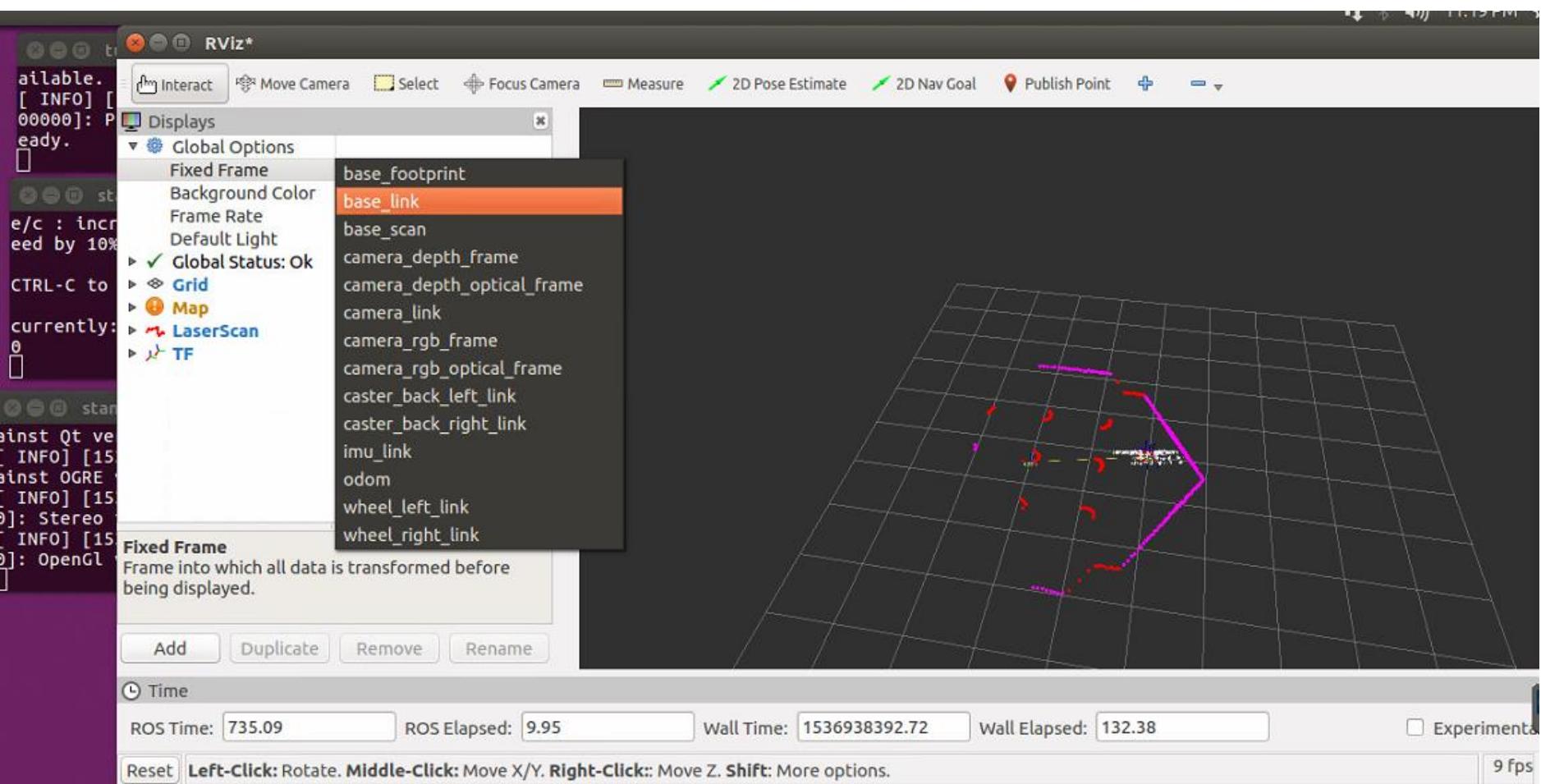
讓 gazebo 的 joint TF 轉換 透過 robot_state_publisher 發佈出來





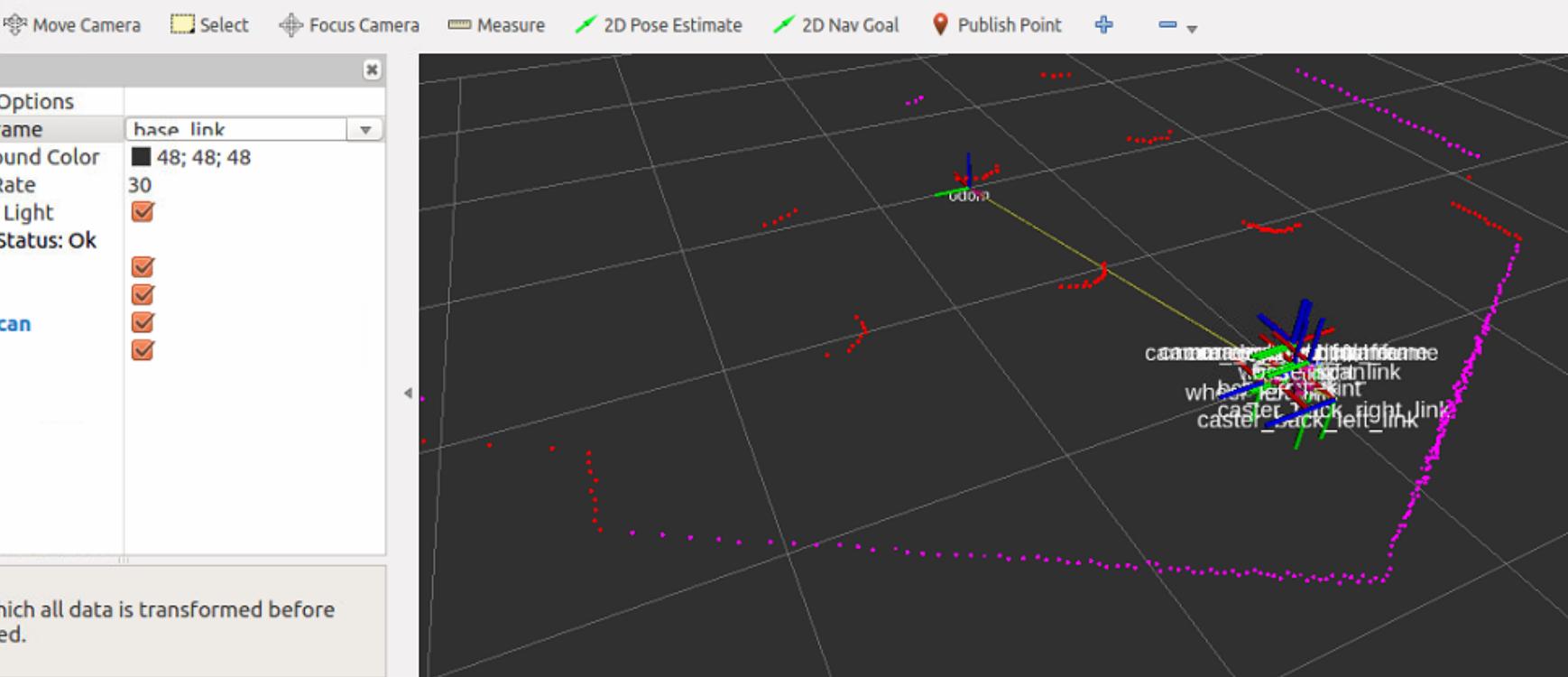
啟動 RVIZ





RViz*

Interact



Displays

Global Options

Fixed Frame **base_link**
Background Color **48; 48; 48**

Frame Rate
30

Default Light

Global Status: Ok

Grid

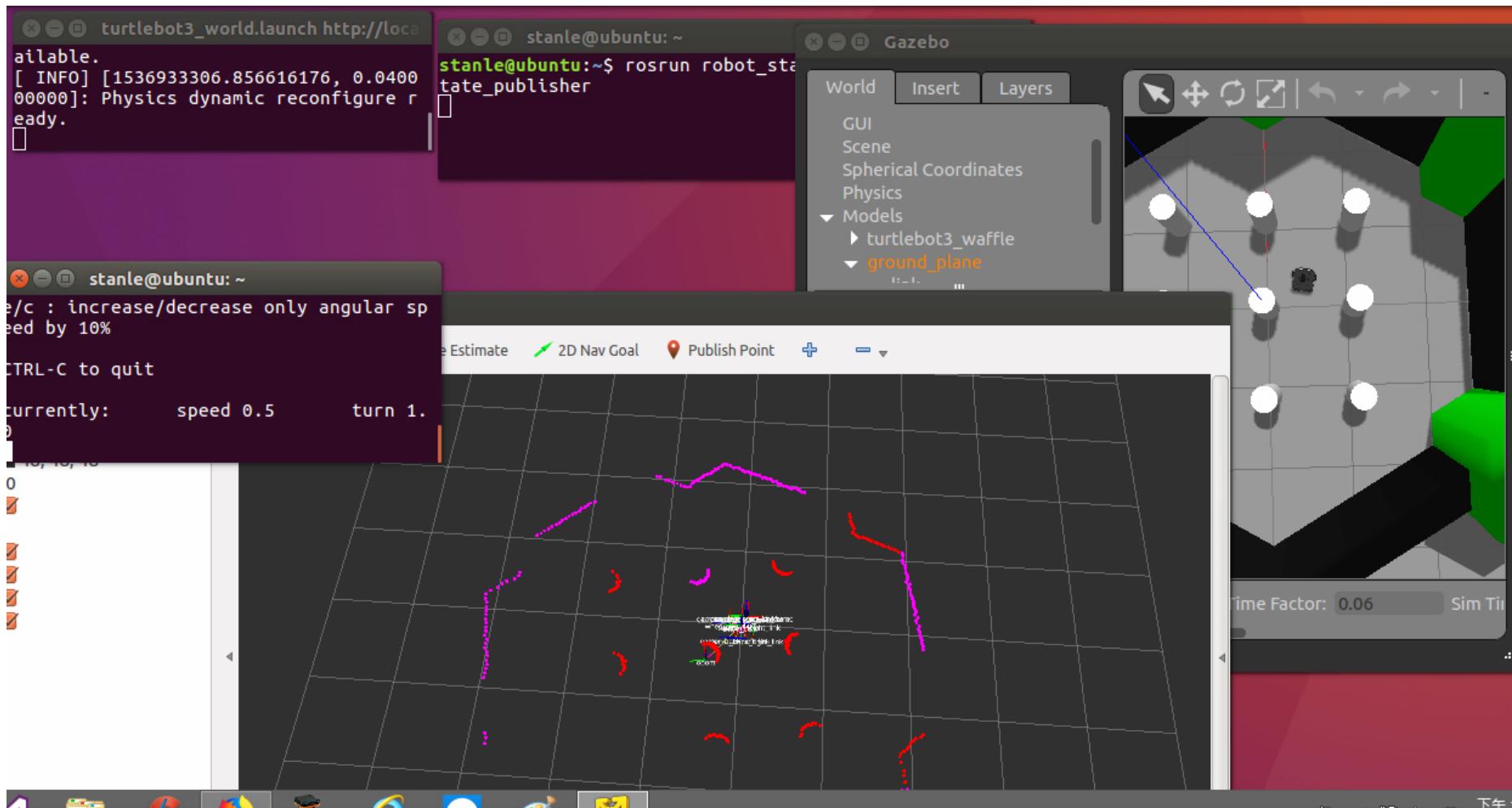
Map

LaserScan

TF

Fixed Frame

Frame into which all data is transformed before being displayed.



關機

`sudo init 0`

`sudo shutdown -h now`

`sudo poweroff`

`sudo halt -f (需要手動關閉最後電源)`

