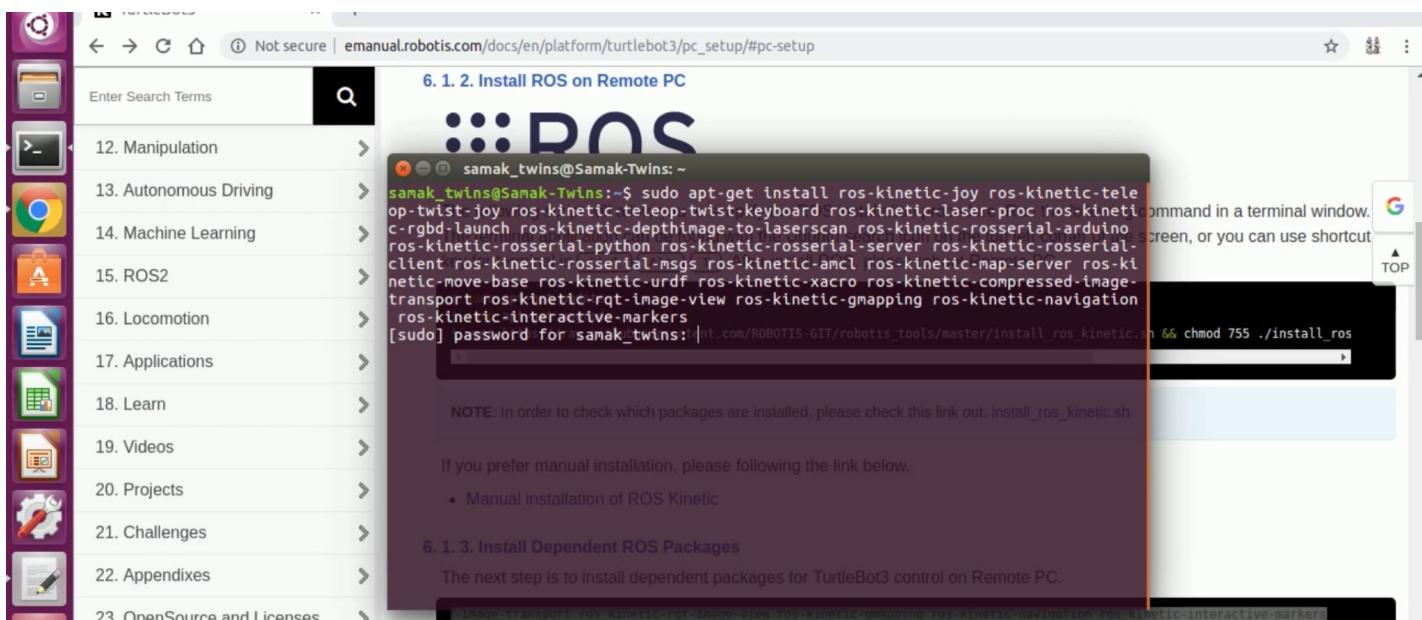
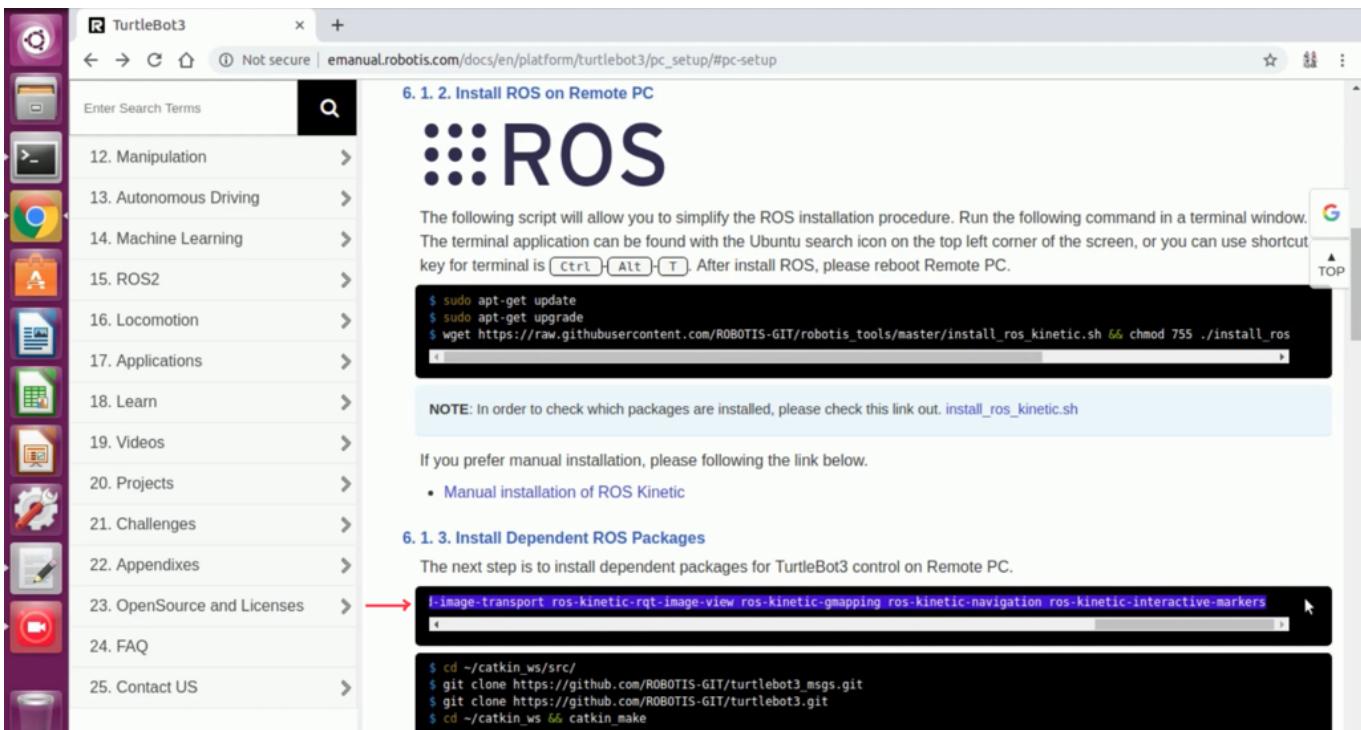


## Turtlebot3 installation

- First Install dependent ROS package by running this command in terminal.



- Install Turtlebot3 package by running these commands in terminal

Terminal

```
TurtleBot3 Installation for ROS Kinetic Kame
=====
1. Install Dependent Packages and the TurtleBot3 Package:
Installation Command for Dependent
sudo apt-get install ros-kinetic-j
samak_twins@Samak-Twins:~$ mkdir ~/TurtleBot3
samak_twins@Samak-Twins:~$ mkdir ~/TurtleBot3/src/
samak_twins@Samak-Twins:~$ cd ~/TurtleBot3/src/
samak_twins@Samak-Twins:~/TurtleBot3/src$ git clone https://github.com/ROBOTIS-G
IT/turtlebot3.git
Cloning into 'turtlebot3'...
remote: Enumerating objects: 27, done.
remote: Counting objects: 100% (27/27), done.
remote: Compressing objects: 100% (25/25), done.
remote: Total 3782 (delta 4), reused 20 (delta 2), pack-reused 3755
Receiving objects: 100% (3782/3782), 99.32 MiB | 1.96 MiB/s, done.
Resolving deltas: 100% (2295/2295), done.
Checking connectivity... done.
samak_twins@Samak-Twins:~/TurtleBot3/src$ git clone https://github.com/ROBOTIS-G
IT/turtlebot3_msgs.git
Cloning into 'turtlebot3_msgs'...
remote: Enumerating objects: 2, done.
remote: Counting objects: 100% (2/2), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 131 (delta 0), reused 0 (delta 0), pack-reused 129
Receiving objects: 100% (131/131), 31.59 KiB | 0 bytes/s, done.
Resolving deltas: 100% (56/56), done.
Checking connectivity... done.
If catkin_make command is complete
samak_twins@Samak-Twins:~/TurtleBot3/src$ TurtleBot3 is done.

2. Source the setup.bash file (*Very Important):
source $HOME/TurtleBot3/devel/setup.bash
```

Plain Text Tab Width: 8 Ln 24. Col 69 INS

Terminal

```
TurtleBot3 Installation for ROS Kinetic Kame
=====
1. Install Dependent Packages and the TurtleBot3 Package:
Installation Command for Dependent
samak_twins@Samak-Twins:~/TurtleBot3
[ 90%] Building CXX object turtlebot3_simulations/turtlebot3_fake/CMakeFiles/tur
tlebot3_fake_node.dir/src/turtlebot3_fake.cpp.o
[ 91%] Generating Python msg __init__.py for turtlebot3_example_generate_messages_py
[ 91%] Built target turtlebot3_example_generate_messages_py
Scanning dependencies of target turtlebot3_msgs_generate_messages
[ 91%] Built target turtlebot3_msgs_generate_messages
Scanning dependencies of target turtlebot3_diagnostics
[ 93%] Building CXX object turtlebot3/turtlebot3Bringup/CMakeFiles/turtlebot3_
diagnostics.dir/src/turtlebot3_diagnostics.cpp.o
[ 95%] Linking CXX executable /home/samak_twins/TurtleBot3/devel/lib/turtlebot3_
slam/flat_world_imu_node
[ 96%] Linking CXX executable /home/samak_twins/TurtleBot3/devel/lib/turtlebot3_
gazebo/turtlebot3_drive
[ 96%] Built target flat_world_imu_node
Scanning dependencies of target turtlebot3_example_generate_messages
[ 96%] Built target turtlebot3_example_generate_messages
[ 96%] Built target turtlebot3_generate_messages_py
[ 98%] Linking CXX executable /home/samak_twins/TurtleBot3/devel/lib/turtlebot3_
bringup/turtlebot3_diagnostics
[ 98%] Built target turtlebot3_diagnostics
[ 98%] Built target turtlebot3_drive
[ 100%] Linking CXX executable /home/samak_twins/TurtleBot3/devel/lib/turtlebot3_
fake/turtlebot3_fake_node
[100%] Built target turtlebot3_fake_node
If catkin_make command is complete
samak_twins@Samak-Twins:~/TurtleBot3$ | or TurtleBot3 is done.

2. Source the setup.bash file (*Very Important):
source $HOME/TurtleBot3/devel/setup.bash
```

- Now Add this to the bashrc file so I have run this command to open it

TurtleBot 3 Installation (~/Documents) - gedit

```
git clone https://github.com/ROBOTIS-GIT/turtlebot3_simulations.git
cd ~/TurtleBot3 && catkin_make

If catkin_make command is completed without any errors, the preparation for TurtleBot3 is done.

=====
2. Source the setup.bash file (*Very Important):
source $HOME/TurtleBot3/devel/setup.bash

Beter add this to the bashrc file.

Run the following command:
→ gedit ~/.bashrc

A gedit (text) file will open. Add the following line to it:
source $HOME/TurtleBot3/devel/setup.bash

Save the file and close it. Run the following Command:
source ~/.bashrc

=====
3. Export TurtleBot3 Model:

TurtleBot3 has three models, burger, waffle, and waffle_pi, so you have to set which model to use before using. To do this, we specify the model to be used with the export command.

export TURTLEBOT3_MODEL=burger
export TURTLEBOT3_MODEL=waffle
export TURTLEBOT3_MODEL=waffle_pi
```

Plain Text ▾ Tab Width: 8 ▾ Ln 44, Col 24 ▾ INS

Terminal

```
# ~/.bashrc: executed by bash(1) for non-login shells.
# see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
# for examples

# If not running interactively, don't do anything
case $- in
  *i*) ;;
  *) return;;
esac

# don't put duplicate lines or lines
# See bash(1) for more options
HISTCONTROL=ignoreboth

# append to the history file, don't overwrite it
shopt -s histappend

# for setting history length see HISTSIZE and HISTFILESIZE in bash(1)
HISTSIZE=1000
HISTFILESIZE=2000

# check the window size after each command and update the values of LINES and COLUMNS
shopt -s checkwinsize

# If set, the pattern "**" used in globbing
# match all files and zero or more directories in a path name
# # update the values of LINES and COLUMNS
# shopt -s globstar

# make less more friendly for non-terminals
[ -x /usr/bin/lesspipe ] && eval "$_LESSOPEN='| /usr/bin/lesspipe %s'$_LESSCLOSE='| /usr/bin/lesspipe %s'$_"

# set variable identifying the chroot you work in (used in the prompt below)
if [ -z "${debian_chroot:-}" ] && [ -r /etc/debian_chroot ]; then
  debian_chroot=$(cat /etc/debian_chroot)
fi

# set a fancy prompt (non-color, unless we know we "want" color)
Loading file '/home/samak_twins/.bashrc'...
```

.bashrc

sh ▾ Tab Width: 8 ▾ Ln 1, Col 1 ▾ INS

- Then add the following line to bashrc file

TurtleBot 3 Installation (~/.Documents) - gedit

```

git clone https://github.com/ROBOTIS-GIT/turtlebot3_simulations.git
cd ~/TurtleBot3 && catkin_make

If catkin_make command is completed without any errors, the preparation for TurtleBot3 is done.

=====
2. Source the setup.bash file (*Very Important):
source $HOME/TurtleBot3/devel/setup.bash

Beter add this to the bashrc file.

Run the following command:
gedit ~/.bashrc

A gedit (text) file will open. Add the following line to it:
source $HOME/TurtleBot3/devel/setup.bash

Save the file and close it. Run the following Command:
source ~/.bashrc

=====
3. Export TurtleBot3 Model:
TurtleBot3 has three models, burger, waffle, and waffle_pi, so you have to set which model to use before using. To do this, we specify the model to be used with the export command.

export TURTLEBOT3_MODEL=burger
export TURTLEBOT3_MODEL=waffle

```

Plain Text ▾ Tab Width: 8 ▾ Ln 48, Col 49 ▾ INS

\*.bashrc (~/) - gedit

```

alias fgrep='grep --color=auto'
alias egrep='egrep --color=auto'
fi

# colored GCC warnings and errors
#export GCC_COLORS='error=01;31:warning=01;35:note=01;36:caret=01;32:locus=01:quote=01'

# some more ls aliases
alias ll='ls -alF'
alias la='ls -A'
alias l='ls -CF'

# Add an "alert" alias for long running commands. Use like so:
# sleep 10; alert
alias alert='notify-send --urgency=low -i "$( [ $? = 0 ] && echo terminal || echo error)" "$(history|tail -n1|sed -e '\''$s/\^[[0-9]{1};[0-9]{1}m//g'\'' -e '\''$s/\[\e[0m\]\'')"'"

# 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 -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/TurtleBot3/devel/setup.bash| ↵

```

sh ▾ Tab Width: 8 ▾ Ln 119, Col 41 ▾ INS

Terminal

```

git clone https://github.com/ROBOTIS-GIT/turtlebot3_simulations.git
cd ~/TurtleBot3 && catkin_make
If catkin_make command is completed without any errors, the preparation for TurtleBot3 is done.
=====
2. Source the setup.bash file (*Very important)
source $HOME/TurtleBot3/devel/setup.bash
Beter add this to the bashrc file.
Run the following command:
gedit ~/.bashrc
A gedit (text) file will open. Add the following line:
source $HOME/TurtleBot3/devel/setup.bash
Save the file and close it. Run the following command:
source ~/.bashrc
=====
3. Export TurtleBot3 Model:
TurtleBot3 has three models, burger, waffle, and waffle_pi, so you have to set which model to use before using. To do this, we specify the model to be used with the export command.

export TURTLEBOT3_MODEL=burger
export TURTLEBOT3_MODEL=waffle

```

Plain Text Tab Width: 8 Ln 52, Col 25 INS

- Now run this command again to be able to open the text file

```

case ?*~*:
    *i*;;
    *) return;;
esac

# don't put duplicate lines or lin
# See bash(1) for more options
HISTCONTROL=ignoreboth

# append to the history file, don't
shopt -s histappend

# for setting history length see H
HISTSIZE=1000
HISTFILESIZE=2000

# check the window size after each
# update the values of LINES and C
shopt -s checkwinsize

# If set, the pattern "**" used in
# match all files and zero or more
# shopt -s globstar

# make less more friendly for non-
[ -x /usr/bin/lesspipe ] && eval "

```

Plain Text Tab Width: 8 Ln 52, Col 25 INS

- Add the following line to it

TurtleBot 3 Installation (~/.Documents) - gedit

Open Save

TurtleBot 3 Installation .bashrc

```
source $HOME/TurtleBot3/devel/setup.bash
```

Save the file and close it. Run the following Command:

```
source ~/.bashrc
```

=====

3. Export TurtleBot3 Model:

TurtleBot3 has three models, burger, waffle, and waffle\_pi, so you have to set which model to use before using. To do this, we specify the model to be used with the export command.

```
export TURTLEBOT3_MODEL=burger
export TURTLEBOT3_MODEL=waffle
export TURTLEBOT3_MODEL=waffle_pi
```

=====

You can run this export command each time in the terminal window. But it is very inconvenient to set it up every time. So, it is recommended to add your default settings in bashrc file. Following is an example for the user of TurtleBot3 Burger model. If you want to use a different model, you just change the value of TURTLEBOT3\_MODEL.

Run the following command:

```
gedit ~/.bashrc
```

A gedit (text) file will open. Add the following line to it:

```
export TURTLEBOT3_MODEL=waffle_pi
```



TurtleBot 3 Installation (~/.Documents) - gedit

Open Save

TurtleBot 3 Installation .bashrc

```
source $HOME/TurtleBot3/devel/setup.bash
```

Save the file and close it. Run the following Command:

```
source ~/.bashrc
```

=====

3. Export TurtleBot3 Model:

TurtleBot3 has three models, burger, waffle, and waffle\_pi, so you have to set which model to use before using. To do this, we specify the model to be used with the export command.

```
export TURTLEBOT3_MODEL=burger
export TURTLEBOT3_MODEL=waffle
export TURTLEBOT3_MODEL=waffle_pi
```

=====

You can run this export command each time in the terminal window. But it is very inconvenient to set it up every time. So, it is recommended to add your default settings in bashrc file. Following is an example for the user of TurtleBot3 Burger model. If you want to use a different model, you just change the value of TURTLEBOT3\_MODEL.

Run the following command:

```
gedit ~/.bashrc
```

A gedit (text) file will open. Add the following line to it:

```
export TURTLEBOT3_MODEL=waffle_pi
```

Save the file and close it. Run the following Command:

```
source ~/.bashrc
```

=====

DONE!

- Now I will test the turtlebot3 by viewing the robot in Gazebo by running these commands in terminal

TurtleBot3 - Google Chrome

TurtleBot3

Not secure | emanual.robotis.com/docs/en/platform/turtlebot3/simulation/

WhatsApp Web Gmail LinkedIn ORCID ResearchGate Google Scholar Robotics Institut... CMU-PhD Ro... Google Play C... YouTube

Enter Search Terms

**11. 2. 1. 6. Virtual SLAM by Multiple TurtleBot3s**

**1) Call Three TurtleBot3s in TurtleBot3 House**

```
$ rosrun turtlebot3_gazebo multi_turtlebot3.launch
```

These loaded turtlebot3s are set initial position and orientation

**2) Execute SLAM**

```
$ ROS_NAMESPACE=tb3_0 rosrun turtlebot3_slam turtlebot3_gmapping.launch set_base_frame:=tb3_0/base_footprint set_...
$ ROS_NAMESPACE=tb3_1 rosrun turtlebot3_slam turtlebot3_gmapping.launch set_base_frame:=tb3_1/base_footprint set_...
$ ROS_NAMESPACE=tb3_2 rosrun turtlebot3_slam turtlebot3_gmapping.launch set_base_frame:=tb3_2/base_footprint set_...
```

**3) Merge Map Data from each TurtleBot3's Map Data**

Before launch this nodes, please make sure arguments for position and orientation of turtlebot3s

```
$ sudo apt-get install ros-kinetic-multirobot-map-merge
$ rosrun turtlebot3_gazebo multi_map_merge.launch
```

**4) Execute RViz**

```
$ rosrun rviz rviz -d `rospack find turtlebot3_gazebo`/rviz/multi_turtlebot3_slam.rviz
```

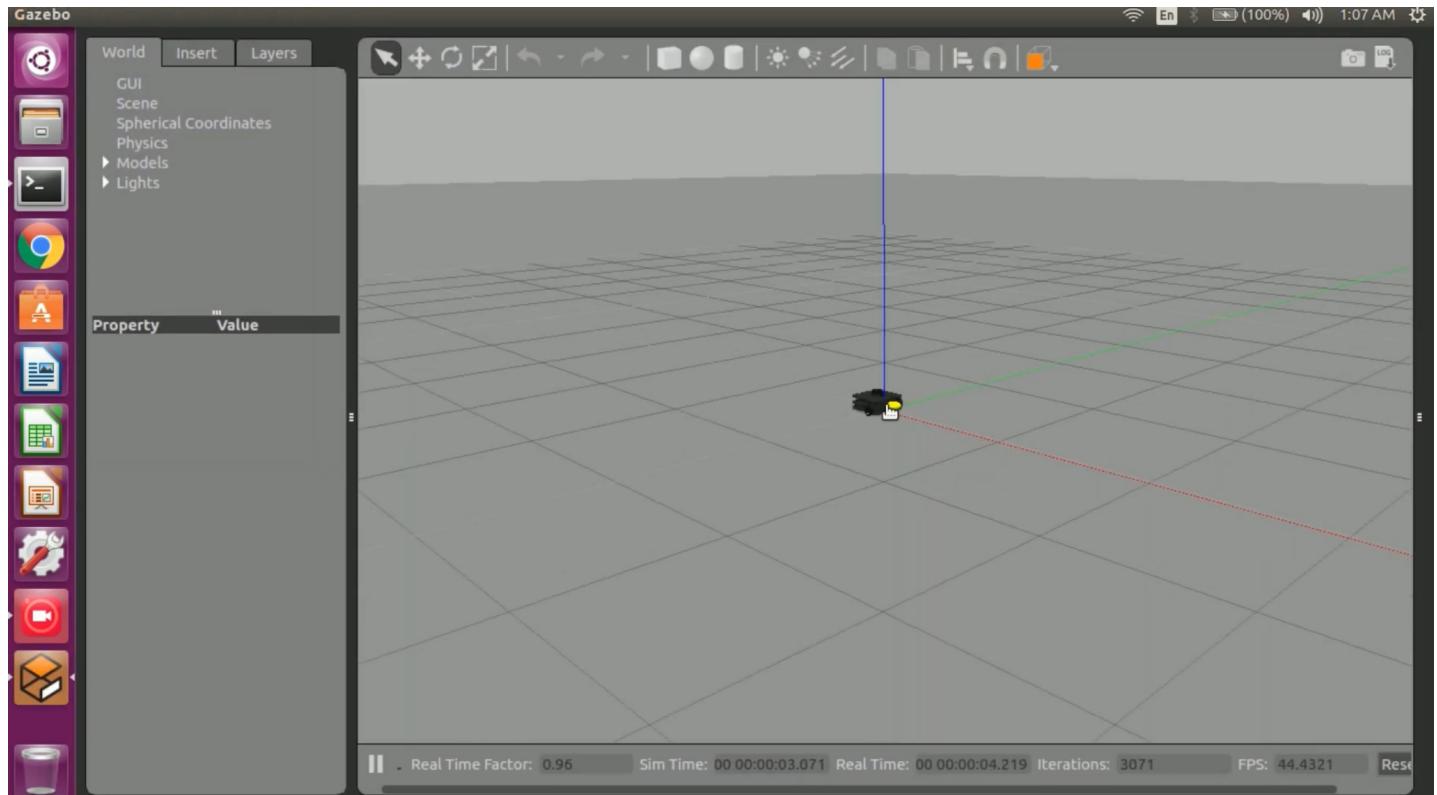
**5) Teleoperation**

```
$ ROS_NAMESPACE=tb3_0 rosrun turtlebot3_teleop turtlebot3_teleop_key
$ ROS_NAMESPACE=tb3_1 rosrun turtlebot3_teleop turtlebot3_teleop_key
```

Terminal

samak\_twins@Samak-Twins: ~

samak\_twins@Samak-Twins:~\$ rosrun turtlebot3\_gazebo turtlebot3\_empty\_world.launch



Thank you.