

Installation Tutorial (ROS, C++ compiler, Gazebo, SPACiSS)

1) Install ROS with navigation and visualization stack

cf: <http://wiki.ros.org/kinetic/Installation/Ubuntu>
or <http://wiki.ros.org/melodic/Installation/Ubuntu>
or <http://wiki.ros.org/noetic/Installation/Ubuntu>
depending on your Ubuntu version.

Authorize all repositories types

```
sudo add-apt-repository "deb http://archive.ubuntu.com/ubuntu $(lsb_release -sc)
main universe restricted multiverse"
```

```
sudo apt-get update
```

Add sources

```
sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc)
main" > /etc/apt/sources.list.d/ros-latest.list'
```

Add keys

```
sudo apt-key adv --keyserver 'hkp://keyserver.ubuntu.com:80' --recv-key
C1CF6E31E6BADE8868B172B4F42ED6FBAB17C654
```

Install (distrib **kinetic or melodic or noetic**, desktop-full)

```
sudo apt-get update
sudo dpkg --configure -a
sudo apt-get install ros-kinetic-desktop-full
```

List available packages

```
apt-cache search ros-kinetic
```

List installed packages

```
rospack list-names
```

Dependencies for building packages

```
sudo apt install python-rosdep python-rosinstall python-rosinstall-generator
python-wstool build-essential
sudo apt install python-rosdep
```

!! For Noetic !!

```
sudo apt install python3-rosdep python3-rosinstall python3-rosinstall-generator
python3-wstool build-essential
sudo apt install python3-rosdep
sudo ln -s /usr/bin/python3 /usr/bin/python
```

Initialize rosdep

```
sudo rosdep init
rosdep update
```

Add ROS environment variables to your bash session every time a new shell is launched:

```
echo "source /opt/ros/kinetic/setup.bash" >> ~/.bashrc
source ~/.bashrc
```

Check ROS env variables

```
printenv | grep ROS
```

2) C++11 compiler by default

```
sudo gedit ~/.bashrc
Add: alias g++="g++ --std=c++0x"
Compile using C++11 with g++ filename.cpp
```

!! For Noetic !!

```
sudo gedit ~/.bashrc
Add: alias g++="g++ --std=c++14"
Compile using C++14 with g++ filename.cpp
```

3) Install Gazebo

http://gazebo-sim.org/tutorials?tut=install_ubuntu

```
sudo sh -c 'echo "deb http://packages.osrfoundation.org/gazebo/ubuntu-stable
`lsb_release -cs`
main" > /etc/apt/sources.list.d/gazebo-stable.list'
cat /etc/apt/sources.list.d/gazebo-stable.list
```

```
sudo apt install wget
wget https://packages.osrfoundation.org/gazebo.key -O - | sudo apt-key add -
```

```
sudo apt-get update
sudo apt-get install gazebo
sudo apt-get install ros-kinetic-gazebo-ros-pkgs ros-kinetic-gazebo-ros-control
ros-kinetic-
gazebo*
```

Run Gazebo once to load models

```
gazebo
```

4) Install QT

```
sudo apt-get install qt4-default
```

!! For Noetic !!

```
sudo apt-get install qt5-default
```

5) Components for the project (if some are missing)

```
sudo apt-get install ros-kinetic-roslaunch
sudo apt-get install ros-kinetic-rviz
sudo apt-get install ros-kinetic-pcl-conversions ros-kinetic-pcl
sudo apt-get install ros-kinetic-perception
sudo apt-get install ros-kinetic-joint-state-publisher
sudo apt-get install ros-kinetic-robot-state-publisher
```

6) Install catkin

```
wget http://packages.ros.org/ros.key -O - | sudo apt-key add -
sudo apt-get update
sudo apt-get install python-catkin-tools
sudo apt-get install ros-kinetic-cmake-modules
```

!! For Noetic !!

```
sudo apt install python3-catkin-tools python3-osrf-pycommon
```

7) Create ROS catkin workspace

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

*Running it the first time in your workspace, it will create a CMakeLists.txt link in your 'src' folder. Additionally, if you look in your current directory you should now have a 'build' and 'devel' folder. Inside the 'devel' folder you can see that there are now several setup.*sh files. Sourcing any of these files will overlay this workspace on top of your environment.*

Source your new setup.*sh file:

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

Check that ROS_PACKAGE_PATH env variable contains the workspace

```
echo $ROS_PACKAGE_PATH
```

8) For developers with QT Creator

In CMAKE in src folder in CmakeLists.txt add:

```
#Add custom (non compiling) targets so launch scripts and python files show up
in QT
Creator's project view.
file(GLOB_RECURSE EXTRA_FILES */)
add_custom_target(${PROJECT_NAME}_OTHER_FILES ALL WORKING_DIRECTORY $
{PROJECT_SOURCE_DIR} SOURCES ${EXTRA_FILES})
```

9) Installation SPACiSS

```
cd [workspace]/src
git clone https://github.com/maprdhm/Spaciss.git
cd Spaciss
git submodule update --init --recursive
git checkout [the targeted branch depending on your ros version]
cd ../../
catkin_make or catkin build (twice at the first time)
```

Commons errors

Could not find a package configuration file provided by "move_base" with any of the following name..."

→ `sudo apt-get install ros-kinetic-navigation`

*libcurl: (51) SSL: no alternative certificate subject name matches target host name
'api.ignitionfuel.org'*

→ <https://varhowto.com/how-to-fix-libcurl-51-ssl-no-alternative-certificate-subject-name-matches-target-host-name-api-ignitionfuel-org-gazebo-ubuntu-ros-melodic/>

Gtk-Message: Failed to load module "canberra-gtk-module"

→ `sudo apt-get install libcanberra-gtk-module`

10) Test installation

Simple

```
roslaunch experimental_package business_area.launch
```

External control

```
roslaunch experimental_package business_area_external.launch
```

and in an other terminal:

```
rostopic pub /move_base_simple/goal geometry_msgs/PoseStamped '{header: {stamp: now, frame_id: "odom"}, pose: {position: {x: -10.0, y: 10.0, z: 0.0}, orientation: {w: 1.0}}}'
```

With Gazebo

```
roslaunch pedsim_gazebo_plugin scene1.launch
```