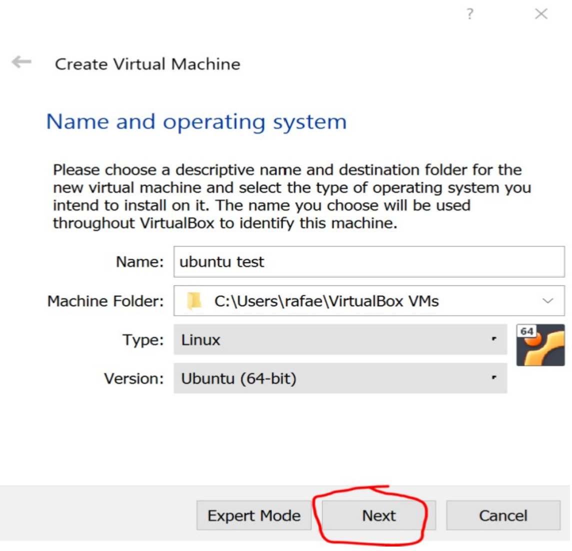
Rafael Rodrigues Control a Robotic Arm using ROS Melodic on Linux 18.04

→ Windows

I use a virtual machine but if you are starting now a dual boot is the best away, in the end the virtual machine will probably be too slow depending on your computer.

1) dual boot
install unebootin/rufos/etcher which one you prefer
download latest ubuntu, in my case 18.04
format a usb stick at least 8gb,
and go to your bios and put boot from usb on top
next normal linux install

2) Virtual machine
download virtual box I used the 6.04
download latest ubuntu, in my case 18.04
create a disk 20 gb, 4 ram , 2 processors if you can do better in the processor do it

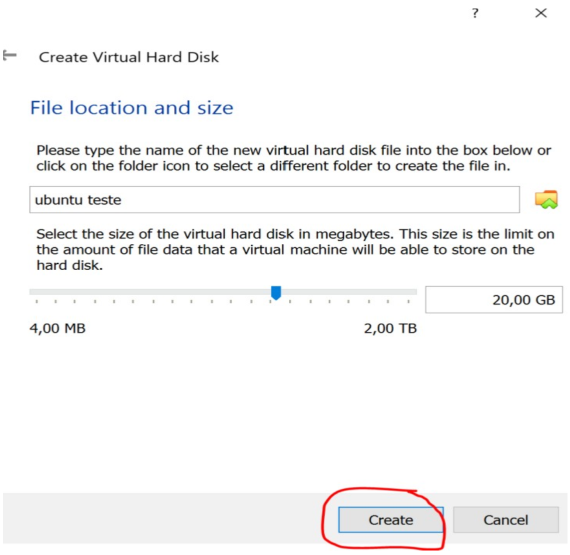


Name and operating system

Please choose a descriptive name and destination folder for the new virtual machine and select the type of operating system you intend to install on it. The name you choose will be used throughout VirtualBox to identify this machine.

Name: ubuntu test
Machine Folder: C:\Users\rafae\VirtualBox VMs
Type: Linux
Version: Ubuntu (64-bit)

Figure 1: VM Name and Type



File location and size

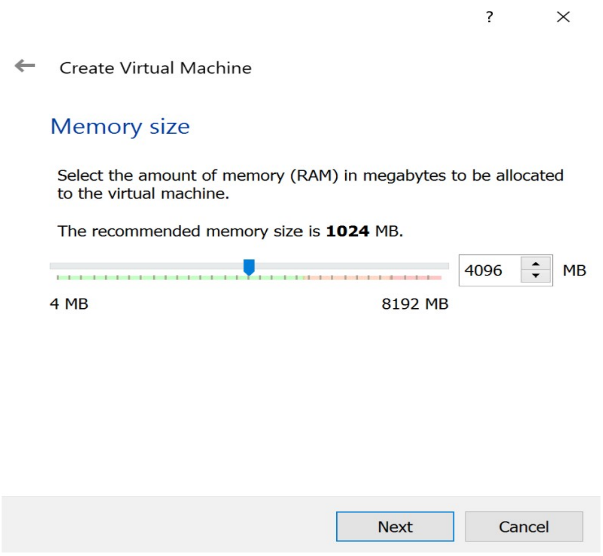
Please type the name of the new virtual hard disk file into the box below or click on the folder icon to select a different folder to create the file in.

ubuntu teste

Select the size of the virtual hard disk in megabytes. This size is the limit on the amount of file data that a virtual machine will be able to store on the hard disk.

20,00 GB

Figure 3: VM Disk



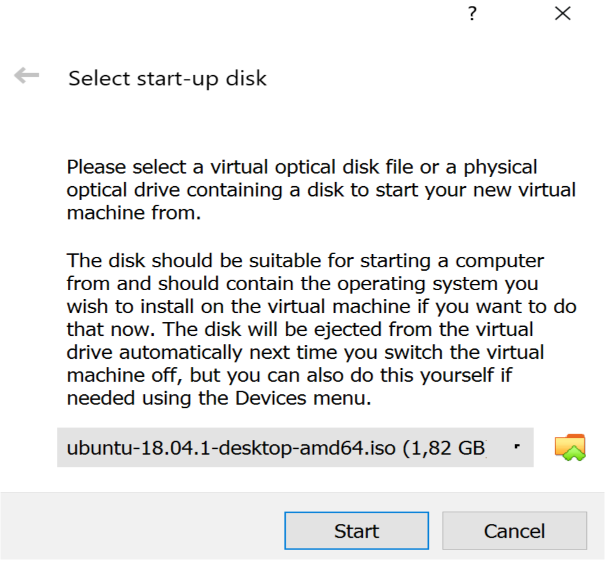
Memory size

Select the amount of memory (RAM) in megabytes to be allocated to the virtual machine.

The recommended memory size is **1024** MB.

4096 MB

Figure 2: VM Ram



Select start-up disk

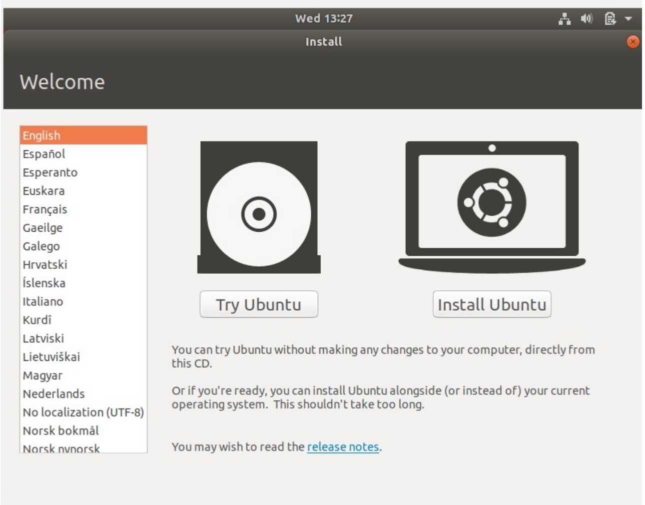
Please select a virtual optical disk file or a physical optical drive containing a disk to start your new virtual machine from.

The disk should be suitable for starting a computer from and should contain the operating system you wish to install on the virtual machine if you want to do that now. The disk will be ejected from the virtual drive automatically next time you switch the virtual machine off, but you can also do this yourself if needed using the Devices menu.

ubuntu-18.04.1-desktop-amd64.iso (1,82 GB

Figure 4: VM Start up

next normal linux instal



You can try Ubuntu without making any changes to your computer, directly from this CD.

Or if you're ready, you can install Ubuntu alongside (or instead of) your current operating system. This shouldn't take too long.

You may wish to read the release notes.

Figure 5: Linux Install

→ Linux

Now we have a fresh linux

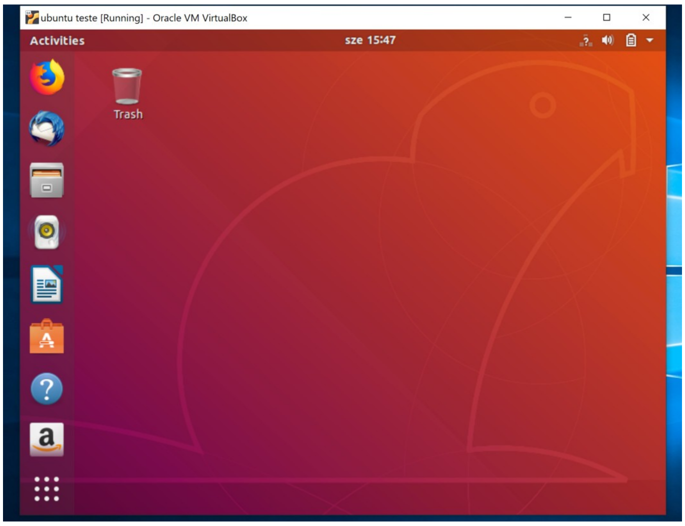
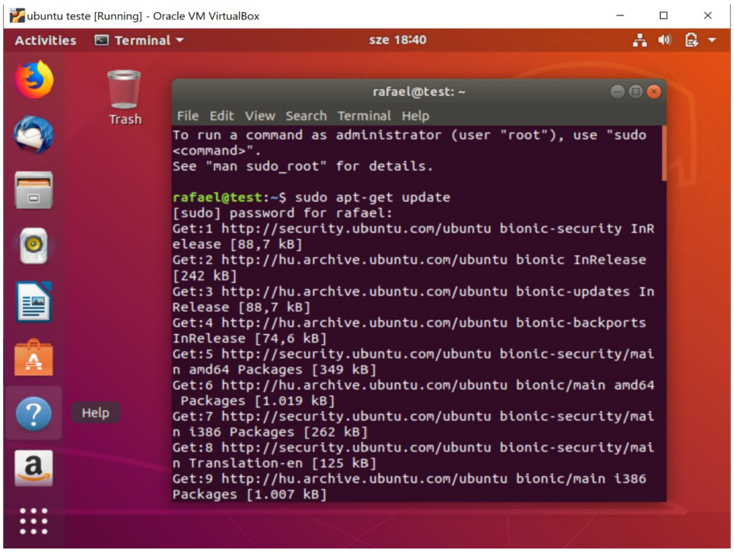


Figure 6: Ubuntu fresh install

you should do a `sudo apt-get update` to verefy if there is any update


ubuntu teste [Running] - Oracle VM VirtualBox
Activities
Terminal
sze 18:40
Trash
Help
rafael@test: ~
File Edit View Search Terminal Help

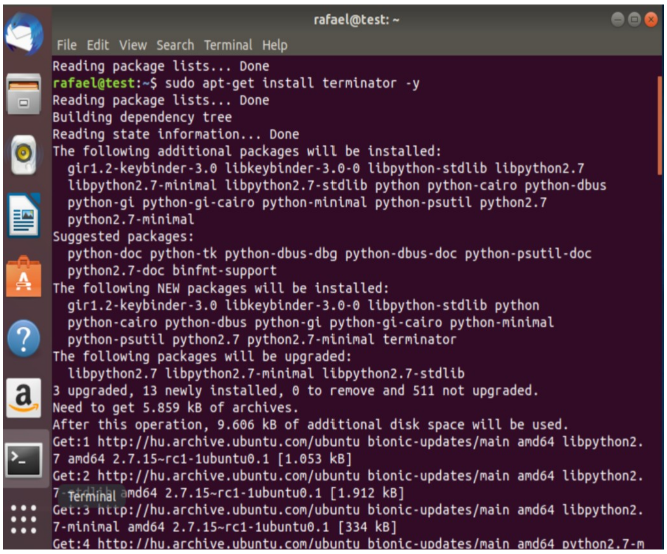
```
To run a command as administrator (user "root"), use "sudo
<command>".
See "man sudo_root" for details.

rafael@test:~$ sudo apt-get update
[sudo] password for rafael:
Get:1 http://security.ubuntu.com/ubuntu bionic-security InR
elease [88,7 kB]
Get:2 http://hu.archive.ubuntu.com/ubuntu bionic InRelease
[242 kB]
Get:3 http://hu.archive.ubuntu.com/ubuntu bionic-updates In
Release [88,7 kB]
Get:4 http://hu.archive.ubuntu.com/ubuntu bionic-backports
InRelease [74,6 kB]
Get:5 http://security.ubuntu.com/ubuntu bionic-security/mai
n amd64 Packages [349 kB]
Get:6 http://hu.archive.ubuntu.com/ubuntu bionic/main amd64
 Packages [1.019 kB]
Get:7 http://security.ubuntu.com/ubuntu bionic-security/mai
n i386 Packages [262 kB]
Get:8 http://security.ubuntu.com/ubuntu bionic-security/mai
n Translation-en [125 kB]
Get:9 http://hu.archive.ubuntu.com/ubuntu bionic/main i386
Packages [1.007 kB]
```

Figure 7: Ubuntu Update verification

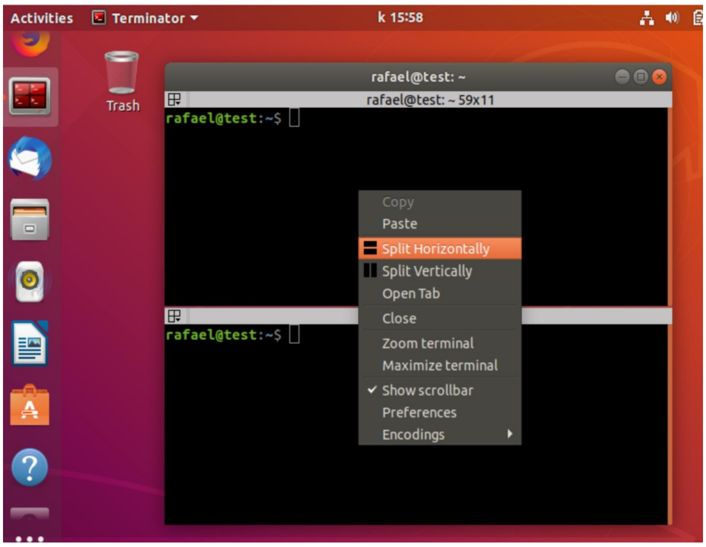
we will be using multiple terminal windows so a goo idea is to install terminator now press "ctrl+alt+t"

```
sudo apt-get install terminator -y
```



```
Reading package lists... Done
rafael@test:~$ sudo apt-get install terminator -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  gir1.2-keybinder-3.0 libkeybinder-3.0-0 libpython-stdlib libpython2.7
  libpython2.7-minimal libpython2.7-stdlib python python-cairo python-dbus
  python-gi python-gi-cairo python-minimal python-psutil python2.7
  python2.7-minimal
Suggested packages:
  python-doc python-tk python-dbus-dbg python-dbus-doc python-psutil-doc
  python2.7-doc binfmt-support
The following NEW packages will be installed:
  gir1.2-keybinder-3.0 libkeybinder-3.0-0 libpython-stdlib python
  python-cairo python-dbus python-gi python-gi-cairo python-minimal
  python-psutil python2.7 python2.7-minimal terminator
The following packages will be upgraded:
  libpython2.7 libpython2.7-minimal libpython2.7-stdlib
3 upgraded, 13 newly installed, 0 to remove and 511 not upgraded.
Need to get 5.859 kB of archives.
After this operation, 9.606 kB of additional disk space will be used.
Get:1 http://hu.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libpython2.
7 amd64 2.7.15~rc1-1ubuntu0.1 [1.053 kB]
Get:2 http://hu.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libpython2.
7-stdlib amd64 2.7.15~rc1-1ubuntu0.1 [1.912 kB]
Get:3 http://hu.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libpython2.
7-minimal amd64 2.7.15~rc1-1ubuntu0.1 [334 kB]
Get:4 http://hu.archive.ubuntu.com/ubuntu bionic-updates/main amd64 python2.7-m
```

Figure 8: Install Terminator



```
rafael@test:~$
```

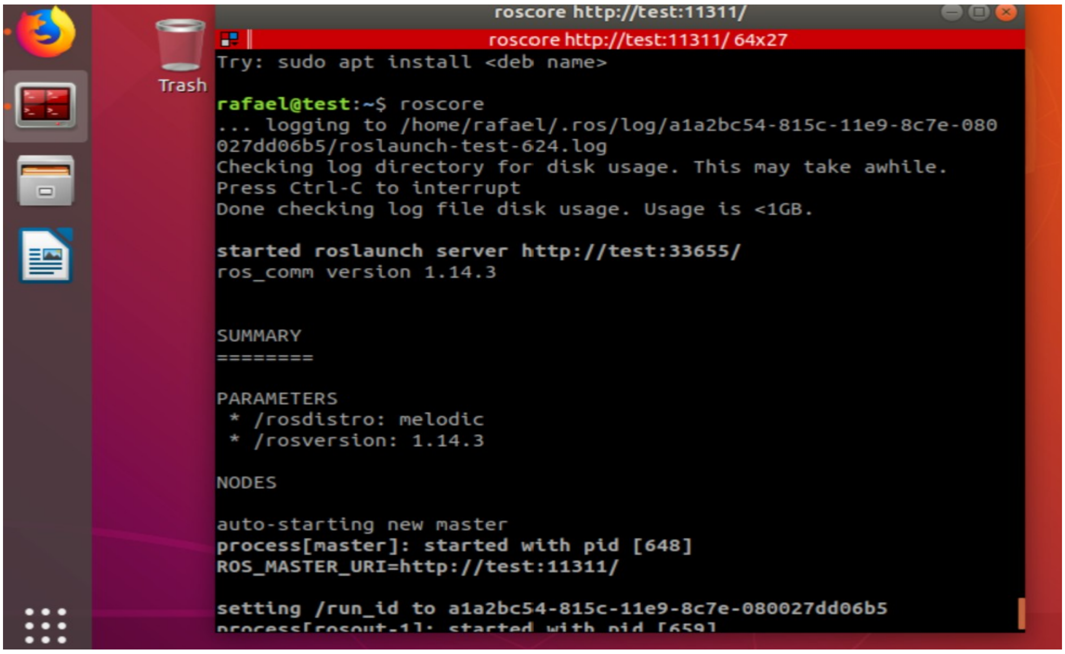
```
rafael@test:~$
```

Figure 9: Terminator in use

→ ROS

install melodic on ubuntu
http://wiki.ros.org/melodic/Installation/Ubuntu
choose desktop full
if you follow all the steps correctly you should now be able to to start roscore

```
roscore
```



```
Try: sudo apt install <deb name>

rafael@test:~$ roscore
... logging to /home/rafael/.ros/log/a1a2bc54-815c-11e9-8c7e-080
027dd06b5/roslaunch-test-624.log
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://test:33655/
ros_comm version 1.14.3


SUMMARY
========

PARAMETERS
 * /rosdistro: melodic
 * /rosversion: 1.14.3

NODES

auto-starting new master
process[master]: started with pid [648]
ROS_MASTER_URI=http://test:11311/

setting /run_id to a1a2bc54-815c-11e9-8c7e-080027dd06b5
process[rosout-1]: started with pid [659]
```

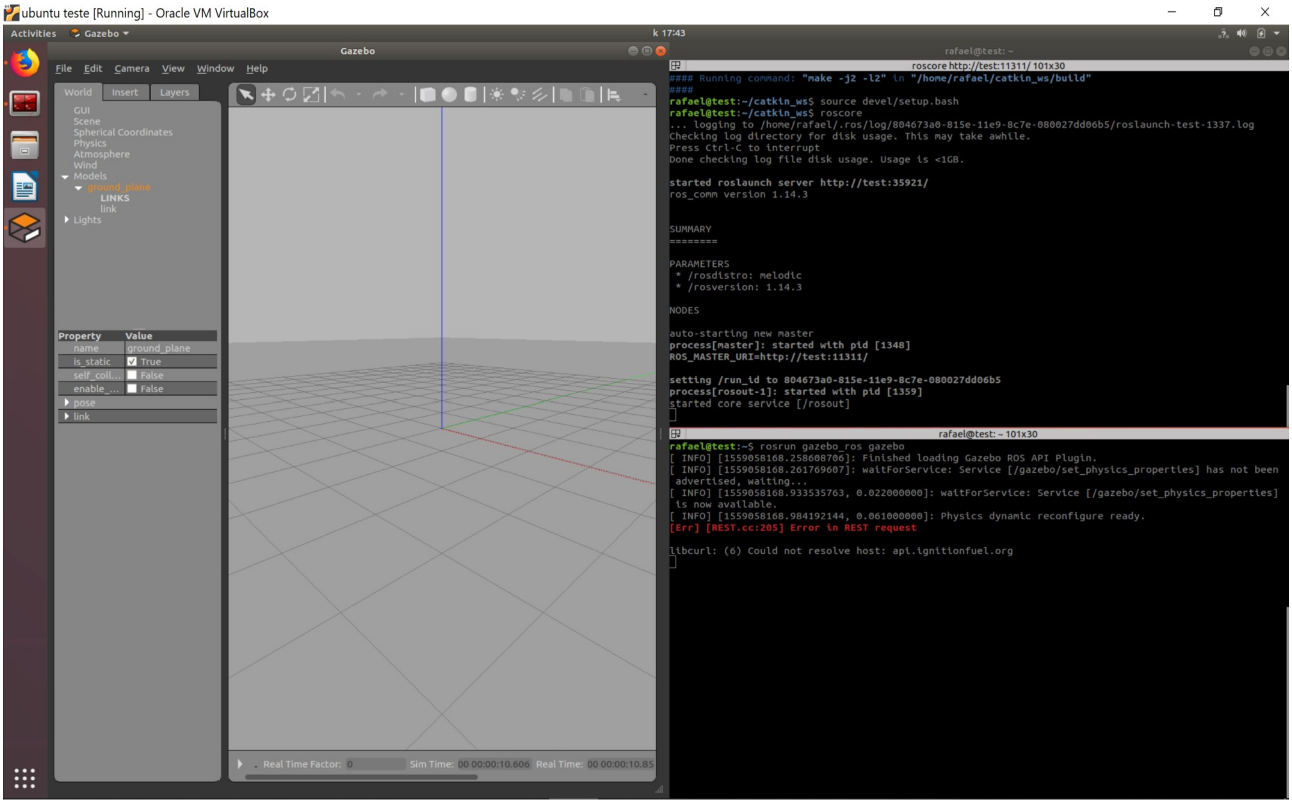
Figure 10: Running roscore command Output

use the following tutorials to get familiar with ROS
try to do until 16
http://wiki.ros.org/ROS/Tutorials

→ Gazebo

http://gazebosim.org/tutorials?tut=ros_installing&cat=connect_ros

```
rosrun gazebo_ros gazebo
```



Property	Value
name	ground_plane
is_static	☑ True
self_coll…	□ False
enable_…	□ False
pose	
link	

Real Time Factor: 0 Sim Time: 00 00:00:10.606 Real Time: 00 00:00:10.85

```
#### Running command: "make -j2 -l2" in "/home/rafael/catkin_ws/build"
####
rafael@test:~/catkin_ws$ source devel/setup.bash
rafael@test:~/catkin_ws$ roscore
... logging to /home/rafael/.ros/log/804673a0-815e-11e9-8c7e-080027dd06b5/roslaunch-test-1337.log
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://test:35921/
ros_comm version 1.14.3


SUMMARY
========

PARAMETERS
 * /rosdistro: melodic
 * /rosversion: 1.14.3

NODES

auto-starting new master
process[master]: started with pid [1348]
ROS_MASTER_URI=http://test:11311/

setting /run_id to 804673a0-815e-11e9-8c7e-080027dd06b5
process[rosout-1]: started with pid [1359]
started core service [/rosout]
```

```
rafael@test:~$ rosrun gazebo_ros gazebo
[ INFO] [1559058168.258608706]: Finished loading Gazebo ROS API Plugin.
[ INFO] [1559058168.261769607]: waitForService: Service [/gazebo/set_physics_properties] has not been
 advertised, waiting...
[ INFO] [1559058168.933535763, 0.022000000]: waitForService: Service [/gazebo/set_physics_properties]
 is now available.
[ INFO] [1559058168.984192144, 0.061000000]: Physics dynamic reconfigure ready.
[Err] [REST.cc:205] Error in REST request

libcurl: (6) Could not resolve host: api.ignitionfuel.org
```

Figure 11: Testing gazebo

Now we have ROS and Gazebo to exit Gazebo press “ctrl + C”, next step install TurtleBot3

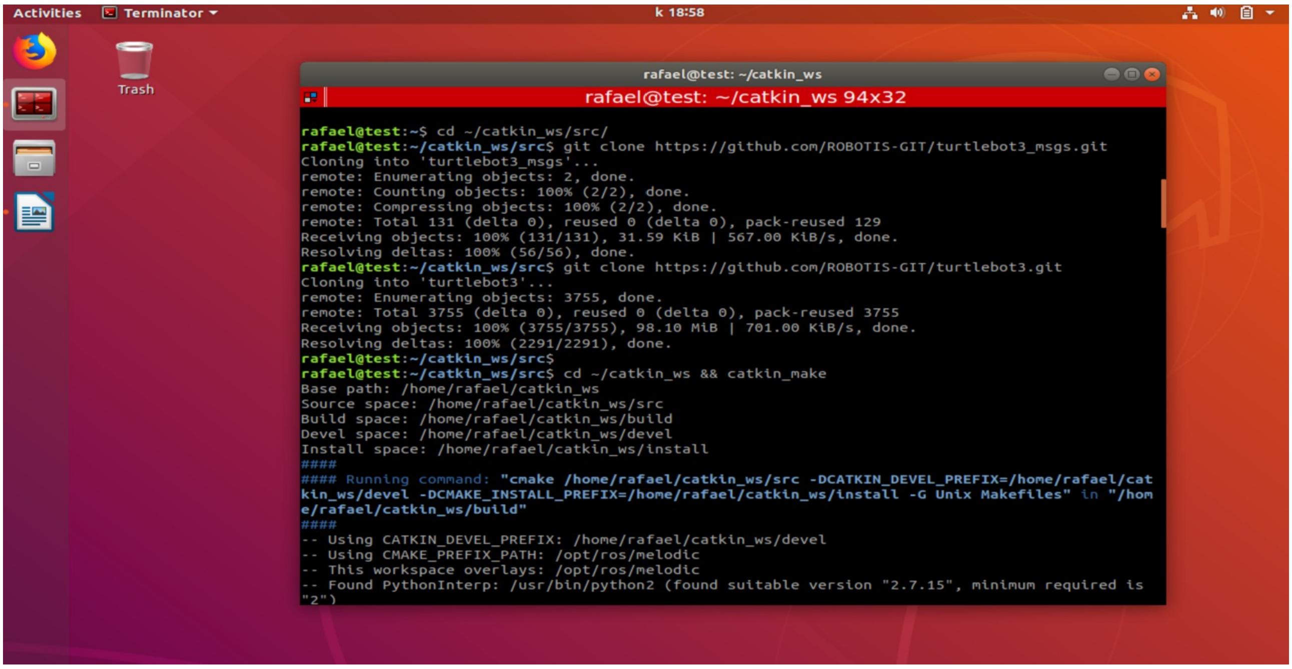
→ TurtleBot3

1) install dependente packages

```
sudo apt-get install ros-melodic-joy ros-melodic-teleop-twist-joy ros-melodic-teleop-twist-keyboard ros-melodic-laser-proc ros-melodic-rgbd-launch ros-melodic-depthimage-to-laserscan ros-melodic-rosserial-arduino ros-melodic-rosserial-python ros-melodic-rosserial-server ros-melodic-rosserial-client ros-melodic-rosserial-msgs ros-melodic-amcl ros-melodic-map-server ros-melodic-move-base ros-melodic-urdf ros-melodic-xacro ros-melodic-compressed-image-transport ros-melodic-rqt-image-view ros-melodic-navigation ros-melodic-interactive-markers
```

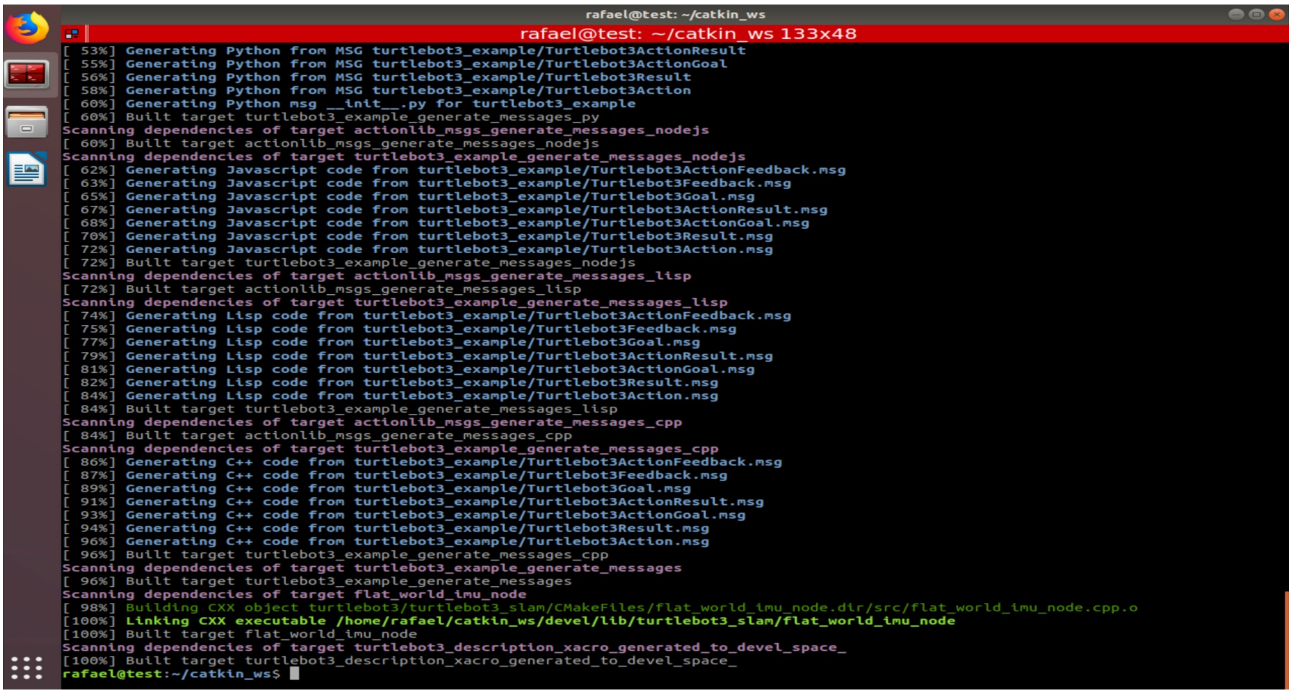
2)

```
cd ~/catkin_ws/src/
git clone https://github.com/ROBOTIS-GIT/turtlebot3_msgs.git
git clone https://github.com/ROBOTIS-GIT/turtlebot3.git
git clone https://github.com/ROBOTIS-GIT/turtlebot3_simulations.git
cd ~/catkin_ws && catkin_make
```



```
rafael@test:~$ cd ~/catkin_ws/src/
rafael@test:~/catkin_ws/src$ git clone https://github.com/ROBOTIS-GIT/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 | 567.00 KiB/s, done.
Resolving deltas: 100% (56/56), done.
rafael@test:~/catkin_ws/src$ git clone https://github.com/ROBOTIS-GIT/turtlebot3.git
Cloning into 'turtlebot3'...
remote: Enumerating objects: 3755, done.
remote: Total 3755 (delta 0), reused 0 (delta 0), pack-reused 3755
Receiving objects: 100% (3755/3755), 98.10 MiB | 701.00 KiB/s, done.
Resolving deltas: 100% (2291/2291), done.
rafael@test:~/catkin_ws/src$
rafael@test:~/catkin_ws/src$ cd ~/catkin_ws && catkin_make
Base path: /home/rafael/catkin_ws
Source space: /home/rafael/catkin_ws/src
Build space: /home/rafael/catkin_ws/build
Devel space: /home/rafael/catkin_ws/devel
Install space: /home/rafael/catkin_ws/install
####
#### Running command: "cmake /home/rafael/catkin_ws/src -DCATKIN_DEVEL_PREFIX=/home/rafael/cat
kin_ws/devel -DCMAKE_INSTALL_PREFIX=/home/rafael/catkin_ws/install -G Unix Makefiles" in "/hom
e/rafael/catkin_ws/build"
####
-- Using CATKIN_DEVEL_PREFIX: /home/rafael/catkin_ws/devel
-- Using CMAKE_PREFIX_PATH: /opt/ros/melodic
-- This workspace overlays: /opt/ros/melodic
-- Found PythonInterp: /usr/bin/python2 (found suitable version "2.7.15", minimum required is
"2")
```

Figure 12: TB3 after Git clone



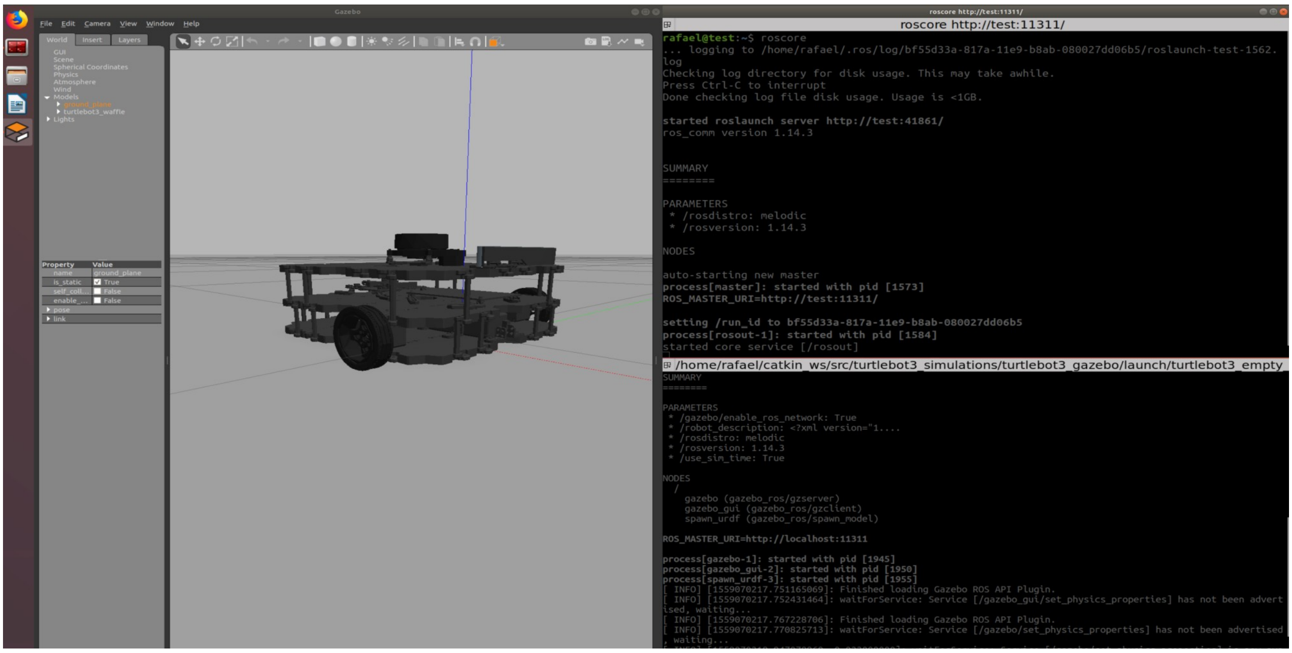
```
[ 53%] Generating Python from MSG turtlebot3_example/Turtlebot3ActionResult
[ 55%] Generating Python from MSG turtlebot3_example/Turtlebot3ActionGoal
[ 56%] Generating Python from MSG turtlebot3_example/Turtlebot3Result
[ 58%] Generating Python from MSG turtlebot3_example/Turtlebot3Action
[ 60%] Generating Python msg __init__.py for turtlebot3_example
[ 60%] Built target turtlebot3_example_generate_messages_py
Scanning dependencies of target actionlib_msgs_generate_messages_nodejs
[ 60%] Built target actionlib_msgs_generate_messages_nodejs
Scanning dependencies of target turtlebot3_example_generate_messages_nodejs
[ 62%] Generating Javascript code from turtlebot3_example/Turtlebot3ActionFeedback.msg
[ 63%] Generating Javascript code from turtlebot3_example/Turtlebot3Feedback.msg
[ 65%] Generating Javascript code from turtlebot3_example/Turtlebot3Goal.msg
[ 67%] Generating Javascript code from turtlebot3_example/Turtlebot3ActionResult.msg
[ 68%] Generating Javascript code from turtlebot3_example/Turtlebot3ActionGoal.msg
[ 70%] Generating Javascript code from turtlebot3_example/Turtlebot3Result.msg
[ 72%] Generating Javascript code from turtlebot3_example/Turtlebot3Action.msg
[ 72%] Built target turtlebot3_example_generate_messages_nodejs
Scanning dependencies of target actionlib_msgs_generate_messages_lisp
[ 72%] Built target actionlib_msgs_generate_messages_lisp
Scanning dependencies of target turtlebot3_example_generate_messages_lisp
[ 74%] Generating Lisp code from turtlebot3_example/Turtlebot3ActionFeedback.msg
[ 75%] Generating Lisp code from turtlebot3_example/Turtlebot3Feedback.msg
[ 77%] Generating Lisp code from turtlebot3_example/Turtlebot3Goal.msg
[ 79%] Generating Lisp code from turtlebot3_example/Turtlebot3ActionResult.msg
[ 81%] Generating Lisp code from turtlebot3_example/Turtlebot3ActionGoal.msg
[ 82%] Generating Lisp code from turtlebot3_example/Turtlebot3Result.msg
[ 84%] Generating Lisp code from turtlebot3_example/Turtlebot3Action.msg
[ 84%] Built target turtlebot3_example_generate_messages_lisp
Scanning dependencies of target actionlib_msgs_generate_messages_cpp
[ 84%] Built target actionlib_msgs_generate_messages_cpp
Scanning dependencies of target turtlebot3_example_generate_messages_cpp
[ 86%] Generating C++ code from turtlebot3_example/Turtlebot3ActionFeedback.msg
[ 87%] Generating C++ code from turtlebot3_example/Turtlebot3Feedback.msg
[ 89%] Generating C++ code from turtlebot3_example/Turtlebot3Goal.msg
[ 91%] Generating C++ code from turtlebot3_example/Turtlebot3ActionResult.msg
[ 93%] Generating C++ code from turtlebot3_example/Turtlebot3ActionGoal.msg
[ 94%] Generating C++ code from turtlebot3_example/Turtlebot3Result.msg
[ 96%] Generating C++ code from turtlebot3_example/Turtlebot3Action.msg
[ 96%] Built target turtlebot3_example_generate_messages_cpp
Scanning dependencies of target turtlebot3_example_generate_messages
[ 96%] Built target turtlebot3_example_generate_messages
Scanning dependencies of target flat_world_imu_node
[ 98%] Building CXX object turtlebot3/turtlebot3_slam/CMakeFiles/flat_world_imu_node.dir/src/flat_world_imu_node.cpp.o
[100%] Linking CXX executable /home/rafael/catkin_ws/devel/lib/turtlebot3_slam/flat_world_imu_node
[100%] Built target flat_world_imu_node
Scanning dependencies of target turtlebot3_description_xacro_generated_to_devel_space_
[100%] Built target turtlebot3_description_xacro_generated_to_devel_space_
rafael@test:~/catkin_ws$
```

Figure 13: TB3 after successful Cmake

if catkin_make finish without errors you are good to go and preparation of TB3 is done

3) Test
New terminal window run `roscore`, in another `roslaunch turtlebot3_gazebo turtlebot3_empty_world.launch`

try to use `export TURTLEBOT3_MODEL='waffle'`
and run it again
if still nothing try
`source /home/”your name”/catkin_ws/devel/setup.bash`



Property	Value
name	ground_plane
is_static	☑ True
self_coll...	□ False
enable_...	□ False

roscore http://test:11311/

```
rafael@test:~$ roscore
... logging to /home/rafael/.ros/log/bf55d33a-817a-11e9-b8ab-080027dd06b5/roslaunch-test-1562.
log
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://test:41861/
ros_comm version 1.14.3


SUMMARY
========

PARAMETERS
 * /rosdistro: melodic
 * /rosversion: 1.14.3

NODES

auto-starting new master
process[master]: started with pid [1573]
ROS_MASTER_URI=http://test:11311/

setting /run_id to bf55d33a-817a-11e9-b8ab-080027dd06b5
process[rosout-1]: started with pid [1584]
started core service [/rosout]
```

/home/rafael/catkin_ws/src/turtlebot3_simulations/turtlebot3_gazebo/launch/turtlebot3_empty

```
SUMMARY
========

PARAMETERS
 * /gazebo/enable_ros_network: True
 * /robot_description: <?xml version="1....
 * /rosdistro: melodic
 * /rosversion: 1.14.3
 * /use_sim_time: True

NODES
  /
    gazebo (gazebo_ros/gzserver)
    gazebo_gui (gazebo_ros/gzclient)
    spawn_urdf (gazebo_ros/spawn_model)

ROS_MASTER_URI=http://localhost:11311

process[gazebo-1]: started with pid [1945]
process[gazebo_gui-2]: started with pid [1950]
process[spawn_urdf-3]: started with pid [1955]
[ INFO] [1559070217.751165069]: Finished loading Gazebo ROS API Plugin.
[ INFO] [1559070217.752431464]: waitForService: Service [/gazebo_gui/set_physics_properties] has not been advert
ised, waiting...
[ INFO] [1559070217.767228706]: Finished loading Gazebo ROS API Plugin.
[ INFO] [1559070217.770825713]: waitForService: Service [/gazebo/set_physics_properties] has not been advertised
, waiting...
```

Figure 14: TB3 Simulation

check this website for more information.
http://emanual.robotis.com/docs/en/platform/turtlebot3/overview/

There is also the possibility to instal from packages this is another method :

```
sudo apt-get install ros-melodic-turtlebot3
sudo apt-get install ros-melodic-turtlebot3-gazebo
```

We now need an arm for our TurtleBot3 so time to add oppenmanipulator-arm

→ Oppenmanipulator-x

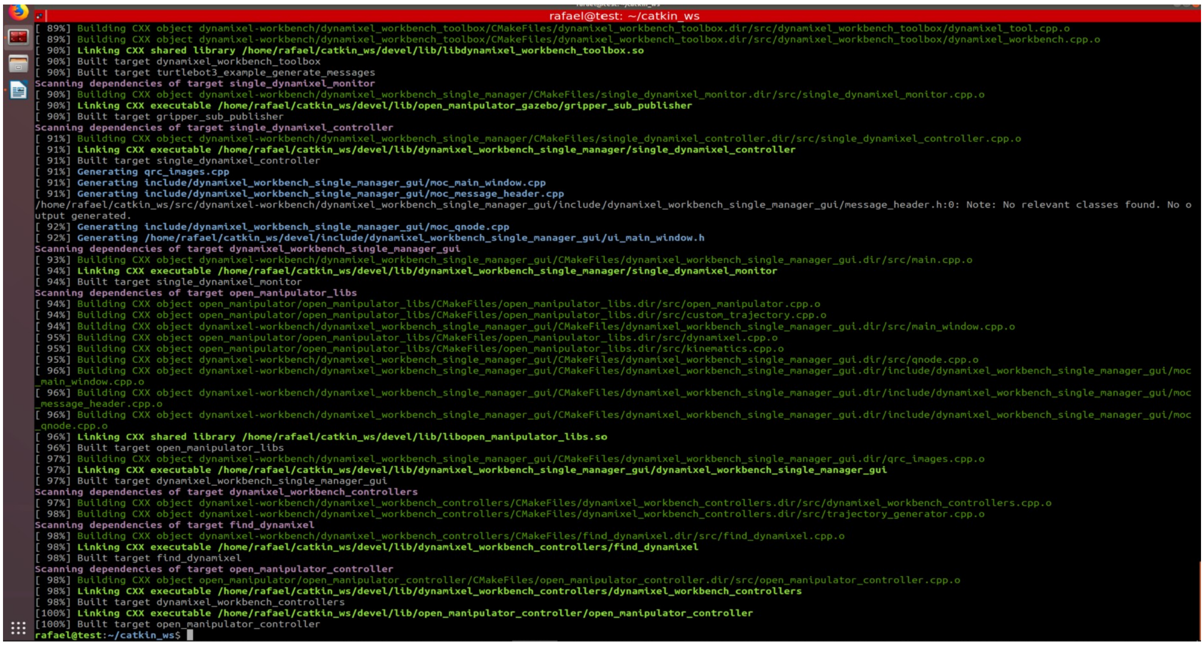
http://emanual.robotis.com/docs/en/platform/openmanipulator_x/overview/

1) Install ROS Packages

```
sudo apt-get install ros-melodic-ros-controllers ros-melodic-gazebo* ros-melodic-moveit* ros-melodic-industrial-core
```

2)

```
 cd ~/catkin_ws/src/
  git clone https://github.com/ROBOTIS-GIT/DynamixelSDK.git
  git clone https://github.com/ROBOTIS-GIT/dynamixel-workbench.git
  git clone https://github.com/ROBOTIS-GIT/dynamixel-workbench-msgs.git
  git clone https://github.com/ROBOTIS-GIT/open_manipulator.git
  git clone https://github.com/ROBOTIS-GIT/open_manipulator_msgs.git
  git clone https://github.com/ROBOTIS-GIT/open_manipulator_simulations.git
  git clone https://github.com/ROBOTIS-GIT/robotis_manipulator.git
  cd ~/catkin_ws && catkin_make
```



```
[ 89%] Building CXX object dynamixel-workbench/dynamixel_workbench_toolbox/CMakeFiles/dynamixel_workbench_toolbox.dir/src/dynamixel_workbench_toolbox/dynamixel_tool.cpp.o
[ 89%] Building CXX object dynamixel-workbench/dynamixel_workbench_toolbox/CMakeFiles/dynamixel_workbench_toolbox.dir/src/dynamixel_workbench_toolbox/dynamixel_workbench.cpp.o
[ 90%] Linking CXX shared library /home/rafael/catkin_ws/devel/lib/libdynamixel_workbench_toolbox.so
[ 90%] Built target dynamixel_workbench_toolbox
[ 90%] Built target turtlebot3_example_generate_messages
Scanning dependencies of target single_dynamixel_monitor
[ 90%] Building CXX object dynamixel-workbench/dynamixel_workbench_single_manager/CMakeFiles/single_dynamixel_monitor.dir/src/single_dynamixel_monitor.cpp.o
[ 90%] Linking CXX executable /home/rafael/catkin_ws/devel/lib/open_manipulator_gazebo/gripper_sub_publisher
[ 90%] Built target gripper_sub_publisher
Scanning dependencies of target single_dynamixel_controller
[ 91%] Building CXX object dynamixel-workbench/dynamixel_workbench_single_manager/CMakeFiles/single_dynamixel_controller.dir/src/single_dynamixel_controller.cpp.o
[ 91%] Linking CXX executable /home/rafael/catkin_ws/devel/lib/dynamixel_workbench_single_manager/single_dynamixel_controller
[ 91%] Built target single_dynamixel_controller
[ 91%] Generating qrc_images.cpp
[ 91%] Generating include/dynamixel_workbench_single_manager_gui/moc_main_window.cpp
[ 91%] Generating include/dynamixel_workbench_single_manager_gui/moc_message_header.cpp
/home/rafael/catkin_ws/src/dynamixel-workbench/dynamixel_workbench_single_manager_gui/include/dynamixel_workbench_single_manager_gui/message_header.h:0: Note: No relevant classes found. No o
utput generated.
[ 92%] Generating include/dynamixel_workbench_single_manager_gui/moc_qnode.cpp
[ 92%] Generating /home/rafael/catkin_ws/devel/include/dynamixel_workbench_single_manager_gui/ui_main_window.h
Scanning dependencies of target dynamixel_workbench_single_manager_gui
[ 93%] Building CXX object dynamixel-workbench/dynamixel_workbench_single_manager_gui/CMakeFiles/dynamixel_workbench_single_manager_gui.dir/src/main.cpp.o
[ 94%] Linking CXX executable /home/rafael/catkin_ws/devel/lib/dynamixel_workbench_single_manager/single_dynamixel_monitor
[ 94%] Built target single_dynamixel_monitor
Scanning dependencies of target open_manipulator_libs
[ 94%] Building CXX object open_manipulator/open_manipulator_libs/CMakeFiles/open_manipulator_libs.dir/src/open_manipulator.cpp.o
[ 94%] Building CXX object open_manipulator/open_manipulator_libs/CMakeFiles/open_manipulator_libs.dir/src/custom_trajectory.cpp.o
[ 94%] Building CXX object dynamixel-workbench/dynamixel_workbench_single_manager_gui/CMakeFiles/dynamixel_workbench_single_manager_gui.dir/src/main_window.cpp.o
[ 95%] Building CXX object open_manipulator/open_manipulator_libs/CMakeFiles/open_manipulator_libs.dir/src/dynamixel.cpp.o
[ 95%] Building CXX object open_manipulator/open_manipulator_libs/CMakeFiles/open_manipulator_libs.dir/src/kinematics.cpp.o
[ 95%] Building CXX object dynamixel-workbench/dynamixel_workbench_single_manager_gui/CMakeFiles/dynamixel_workbench_single_manager_gui.dir/src/qnode.cpp.o
[ 96%] Building CXX object dynamixel-workbench/dynamixel_workbench_single_manager_gui/CMakeFiles/dynamixel_workbench_single_manager_gui.dir/include/dynamixel_workbench_single_manager_gui/moc
_main_window.cpp.o
[ 96%] Building CXX object dynamixel-workbench/dynamixel_workbench_single_manager_gui/CMakeFiles/dynamixel_workbench_single_manager_gui.dir/include/dynamixel_workbench_single_manager_gui/moc
_message_header.cpp.o
[ 96%] Building CXX object dynamixel-workbench/dynamixel_workbench_single_manager_gui/CMakeFiles/dynamixel_workbench_single_manager_gui.dir/include/dynamixel_workbench_single_manager_gui/moc
_qnode.cpp.o
[ 96%] Linking CXX shared library /home/rafael/catkin_ws/devel/lib/libopen_manipulator_libs.so
[ 96%] Built target open_manipulator_libs
[ 97%] Building CXX object dynamixel-workbench/dynamixel_workbench_single_manager_gui/CMakeFiles/dynamixel_workbench_single_manager_gui.dir/qrc_images.cpp.o
[ 97%] Linking CXX executable /home/rafael/catkin_ws/devel/lib/dynamixel_workbench_single_manager_gui/dynamixel_workbench_single_manager_gui
[ 97%] Built target dynamixel_workbench_single_manager_gui
Scanning dependencies of target dynamixel_workbench_controllers
[ 97%] Building CXX object dynamixel-workbench/dynamixel_workbench_controllers/CMakeFiles/dynamixel_workbench_controllers.dir/src/dynamixel_workbench_controllers.cpp.o
[ 98%] Building CXX object dynamixel-workbench/dynamixel_workbench_controllers/CMakeFiles/dynamixel_workbench_controllers.dir/src/trajectory_generator.cpp.o
Scanning dependencies of target find_dynamixel
[ 98%] Building CXX object dynamixel-workbench/dynamixel_workbench_controllers/CMakeFiles/find_dynamixel.dir/src/find_dynamixel.cpp.o
[ 98%] Linking CXX executable /home/rafael/catkin_ws/devel/lib/dynamixel_workbench_controllers/find_dynamixel
[ 98%] Built target find_dynamixel
Scanning dependencies of target open_manipulator_controller
[ 98%] Building CXX object open_manipulator/open_manipulator_controller/CMakeFiles/open_manipulator_controller.dir/src/open_manipulator_controller.cpp.o
[ 98%] Linking CXX executable /home/rafael/catkin_ws/devel/lib/dynamixel_workbench_controllers/dynamixel_workbench_controllers
[ 98%] Built target dynamixel_workbench_controllers
[100%] Linking CXX executable /home/rafael/catkin_ws/devel/lib/open_manipulator_controller/open_manipulator_controller
[100%] Built target open_manipulator_controller
rafael@test:~/catkin_ws$
```

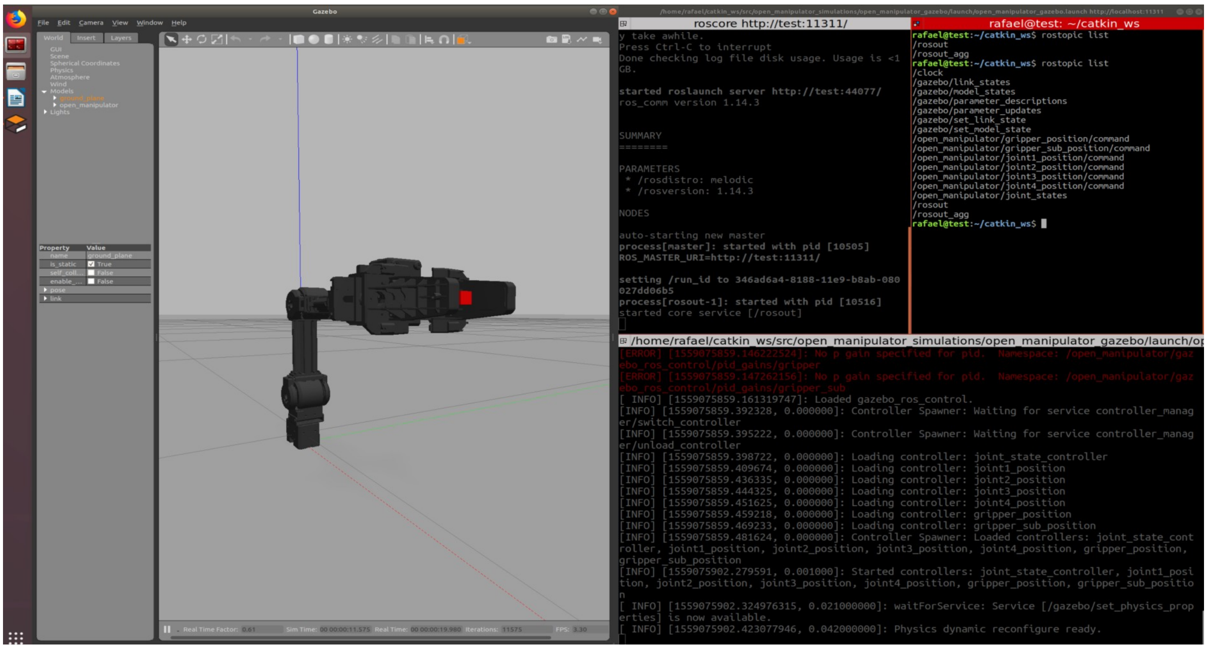
Figure 15: Arm successful Cmake

3)Test → Now try to see if gazebo is working

```
roslaunch open_manipulator_gazebo open_manipulator_gazebo.launch
```

press play in gazebo simulation

in other window use `rostopic list` you should have 3 windows now.


Gazebo
File Edit Camera View Window Help
World Insert Layers
/home/rafael/catkin_ws/src/open_manipulator_simulations/open_manipulator_gazebo/launch/open_manipulator_gazebo.launch http://localhost:11311
roscore http://test:11311/
rafael@test: ~/catkin_ws
/home/rafael/catkin_ws/src/open_manipulator_simulations/open_manipulator_gazebo/launch/op

```
y take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1
GB.

started roslaunch server http://test:44077/
ros_comm version 1.14.3


SUMMARY
========

PARAMETERS
 * /rosdistro: melodic
 * /rosversion: 1.14.3

NODES

auto-starting new master
process[master]: started with pid [10505]
ROS_MASTER_URI=http://test:11311/

setting /run_id to 346ad6a4-8188-11e9-b8ab-080
027dd06b5
process[rosout-1]: started with pid [10516]
started core service [/rosout]
```

```
rafael@test:~/catkin_ws$ rostopic list
/rosout
/rosout_agg
rafael@test:~/catkin_ws$ rostopic list
/clock
/gazebo/link_states
/gazebo/model_states
/gazebo/parameter_descriptions
/gazebo/parameter_updates
/gazebo/set_link_state
/gazebo/set_model_state
/open_manipulator/gripper_position/command
/open_manipulator/gripper_sub_position/command
/open_manipulator/joint1_position/command
/open_manipulator/joint2_position/command
/open_manipulator/joint3_position/command
/open_manipulator/joint4_position/command
/open_manipulator/joint_states
/rosout
/rosout_agg
rafael@test:~/catkin_ws$
```

```
[ERROR] [1559075859.146222524]: No p gain specified for pid.  Namespace: /open_manipulator/gaz
ebo_ros_control/pid_gains/gripper
[ERROR] [1559075859.147262156]: No p gain specified for pid.  Namespace: /open_manipulator/gaz
ebo_ros_control/pid_gains/gripper_sub
[ INFO] [1559075859.161319747]: Loaded gazebo_ros_control.
[INFO] [1559075859.392328, 0.000000]: Controller Spawner: Waiting for service controller_manag
er/switch_controller
[INFO] [1559075859.395222, 0.000000]: Controller Spawner: Waiting for service controller_manag
er/unload_controller
[INFO] [1559075859.398722, 0.000000]: Loading controller: joint_state_controller
[INFO] [1559075859.409674, 0.000000]: Loading controller: joint1_position
[INFO] [1559075859.436335, 0.000000]: Loading controller: joint2_position
[INFO] [1559075859.444325, 0.000000]: Loading controller: joint3_position
[INFO] [1559075859.451625, 0.000000]: Loading controller: joint4_position
[INFO] [1559075859.459218, 0.000000]: Loading controller: gripper_position
[INFO] [1559075859.469233, 0.000000]: Loading controller: gripper_sub_position
[INFO] [1559075859.481624, 0.000000]: Controller Spawner: Loaded controllers: joint_state_cont
roller, joint1_position, joint2_position, joint3_position, joint4_position, gripper_position,
gripper_sub_position
[INFO] [1559075902.279591, 0.001000]: Started controllers: joint_state_controller, joint1_posi
tion, joint2_position, joint3_position, joint4_position, gripper_position, gripper_sub_positio
n
[ INFO] [1559075902.324976315, 0.021000000]: waitForService: Service [/gazebo/set_physics_prop
erties] is now available.
[ INFO] [1559075902.423077946, 0.042000000]: Physics dynamic reconfigure ready.
```

Figure 16: Arm simulation

One more step combine TB3 with open_manipulator

If for some reason the comands above are not wotking for you please try installfrom packages

```
sudo apt-get install ros-melodic-open-manipulator
sudo apt-get install ros-melodic-open-manipulator-gazebo
```

→ TB3 with open_manipulator

https://github.com/ROBOTIS-GIT/open_manipulator_with_tb3
https://www.youtube.com/playlist?list=PLRG6WP3c31_XI3wlvHlx2Mp8BYqgqDURU
http://emanual.robotis.com/docs/en/platform/turtlebot3/manipulation/#manipulation

1)

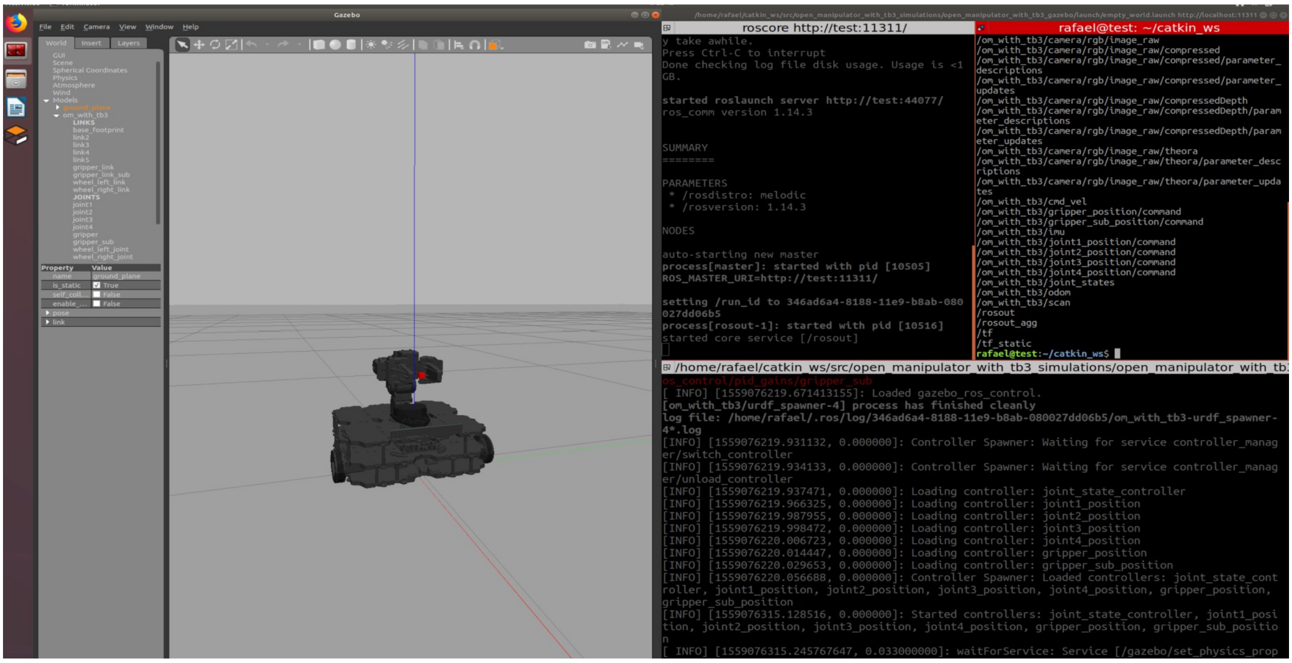
```
cd ~/catkin_ws/src/
git clone https://github.com/ROBOTIS-GIT/open_manipulator_with_tb3.git
git clone
https://github.com/ROBOTIS-GIT/open_manipulator_with_tb3_msgs.git
git clone
https://github.com/ROBOTIS-GIT/open_manipulator_with_tb3_simulations.git
git clone https://github.com/ROBOTIS-GIT/open_manipulator_perceptions.git
sudo apt-get install ros-melodic-smach* ros-melodic-ar-track-alvar ros-
melodic-ar-track-alvar-msgs
cd ~/catkin_ws && catkin_make
```

For this there is also the possibility to install from packages with following command

```
sudo apt-get install ros-melodic-open-manipulator-with-tb3
sudo apt-get install ros-melodic-open-manipulator-with-tb3-gazebo
```

2) Now for simulation in gazebo

```
export TURTLEBOT3_MODEL='waffle'
roslaunch open_manipulator_with_tb3_gazebo empty_world.launch
```



/home/rafael/catkin_ws/src/open_manipulator_with_tb3_simulations/open_manipulator_with_tb3_gazebo/launch/empty_world.launch http://localhost:11311

roscore http://test:11311/

```
y take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1
GB.

started roslaunch server http://test:44077/
ros_comm version 1.14.3


SUMMARY
========

PARAMETERS
 * /rosdistro: melodic
 * /rosversion: 1.14.3

NODES

auto-starting new master
process[master]: started with pid [10505]
ROS_MASTER_URI=http://test:11311/

setting /run_id to 346ad6a4-8188-11e9-b8ab-080
027dd06b5
process[rosout-1]: started with pid [10516]
started core service [/rosout]
```

rafael@test: ~/catkin_ws

```
/om_with_tb3/camera/rgb/image_raw
/om_with_tb3/camera/rgb/image_raw/compressed
/om_with_tb3/camera/rgb/image_raw/compressed/parameter_
descriptions
/om_with_tb3/camera/rgb/image_raw/compressed/parameter_
updates
/om_with_tb3/camera/rgb/image_raw/compressedDepth
/om_with_tb3/camera/rgb/image_raw/compressedDepth/param
eter_descriptions
/om_with_tb3/camera/rgb/image_raw/compressedDepth/param
eter_updates
/om_with_tb3/camera/rgb/image_raw/theora
/om_with_tb3/camera/rgb/image_raw/theora/parameter_desc
riptions
/om_with_tb3/camera/rgb/image_raw/theora/parameter_upda
tes
/om_with_tb3/cmd_vel
/om_with_tb3/gripper_position/command
/om_with_tb3/gripper_sub_position/command
/om_with_tb3/imu
/om_with_tb3/joint1_position/command
/om_with_tb3/joint2_position/command
/om_with_tb3/joint3_position/command
/om_with_tb3/joint4_position/command
/om_with_tb3/joint_states
/om_with_tb3/odom
/om_with_tb3/scan
/rosout
/rosout_agg
/tf
/tf_static
rafael@test:~/catkin_ws$
```

/home/rafael/catkin_ws/src/open_manipulator_with_tb3_simulations/open_manipulator_with_tb3

```
os_control/pid_gains/gripper_sub
[ INFO] [1559076219.671413155]: Loaded gazebo_ros_control.
[om_with_tb3/urdf_spawner-4] process has finished cleanly
log file: /home/rafael/.ros/log/346ad6a4-8188-11e9-b8ab-080027dd06b5/om_with_tb3-urdf_spawner-
4*.log
[INFO] [1559076219.931132, 0.000000]: Controller Spawner: Waiting for service controller_manag
er/switch_controller
[INFO] [1559076219.934133, 0.000000]: Controller Spawner: Waiting for service controller_manag
er/unload_controller
[INFO] [1559076219.937471, 0.000000]: Loading controller: joint_state_controller
[INFO] [1559076219.966325, 0.000000]: Loading controller: joint1_position
[INFO] [1559076219.987955, 0.000000]: Loading controller: joint2_position
[INFO] [1559076219.998472, 0.000000]: Loading controller: joint3_position
[INFO] [1559076220.006723, 0.000000]: Loading controller: joint4_position
[INFO] [1559076220.014447, 0.000000]: Loading controller: gripper_position
[INFO] [1559076220.029653, 0.000000]: Loading controller: gripper_sub_position
[INFO] [1559076220.056688, 0.000000]: Controller Spawner: Loaded controllers: joint_state_cont
roller, joint1_position, joint2_position, joint3_position, joint4_position, gripper_position,
gripper_sub_position
[INFO] [1559076315.128516, 0.000000]: Started controllers: joint_state_controller, joint1_posi
tion, joint2_position, joint3_position, joint4_position, gripper_position, gripper_sub_positio
n
[ INFO] [1559076315.245767647, 0.033000000]: waitForService: Service [/gazebo/set_physics_prop
```

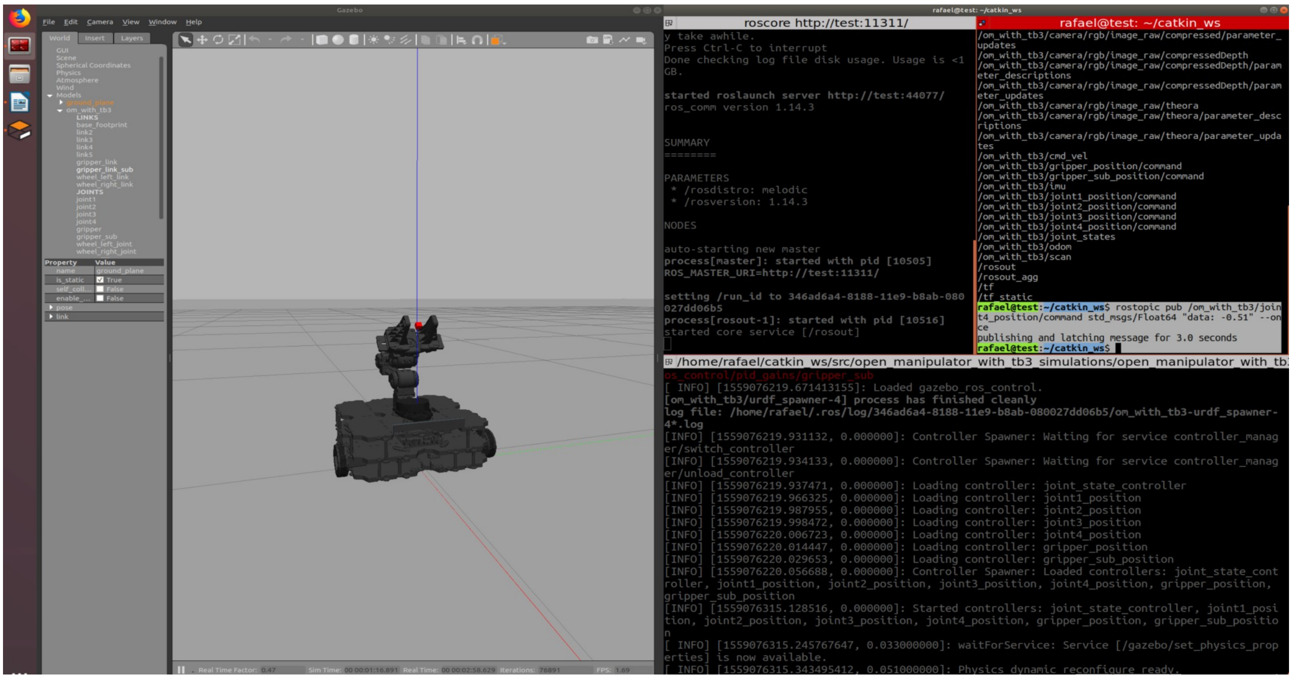
Figure 17: TB3 and arm simulation

press play

in other window open rostopic, and in another use this

```
rostopic pub /om_with_tb3/joint4_position/command std_msgs/Float64
"data: -0.51" --once
```

Did you see the arm moving?
If yes well done everything is working lets build a gui to control the arm
else No press play in Gazebo!
See output in the next page.



```
y take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1
GB.

started roslaunch server http://test:44077/
ros_comm version 1.14.3


SUMMARY
========

PARAMETERS
 * /rosdistro: melodic
 * /rosversion: 1.14.3

NODES

auto-starting new master
process[master]: started with pid [10505]
ROS_MASTER_URI=http://test:11311/

setting /run_id to 346ad6a4-8188-11e9-b8ab-080
027dd06b5
process[rosout-1]: started with pid [10516]
started core service [/rosout]
```

```
/om_with_tb3/camera/rgb/image_raw/compressed/parameter_
updates
/om_with_tb3/camera/rgb/image_raw/compressedDepth
/om_with_tb3/camera/rgb/image_raw/compressedDepth/param
eter_descriptions
/om_with_tb3/camera/rgb/image_raw/compressedDepth/param
eter_updates
/om_with_tb3/camera/rgb/image_raw/theora
/om_with_tb3/camera/rgb/image_raw/theora/parameter_desc
riptions
/om_with_tb3/camera/rgb/image_raw/theora/parameter_upda
tes
/om_with_tb3/cmd_vel
/om_with_tb3/gripper_position/command
/om_with_tb3/gripper_sub_position/command
/om_with_tb3/imu
/om_with_tb3/joint1_position/command
/om_with_tb3/joint2_position/command
/om_with_tb3/joint3_position/command
/om_with_tb3/joint4_position/command
/om_with_tb3/joint_states
/om_with_tb3/odom
/om_with_tb3/scan
/rosout
/rosout_agg
/tf
/tf_static
rafael@test:~/catkin_ws$ rostopic pub /om_with_tb3/join
t4_position/command std_msgs/Float64 "data: -0.51" --on
ce
publishing and latching message for 3.0 seconds
rafael@test:~/catkin_ws$
```

```
os_control/pid_gains/gripper_sub
[ INFO] [1559076219.671413155]: Loaded gazebo_ros_control.
[om_with_tb3/urdf_spawner-4] process has finished cleanly
log file: /home/rafael/.ros/log/346ad6a4-8188-11e9-b8ab-080027dd06b5/om_with_tb3-urdf_spawner-
4*.log
[INFO] [1559076219.931132, 0.000000]: Controller Spawner: Waiting for service controller_manag
er/switch_controller
[INFO] [1559076219.934133, 0.000000]: Controller Spawner: Waiting for service controller_manag
er/unload_controller
[INFO] [1559076219.937471, 0.000000]: Loading controller: joint_state_controller
[INFO] [1559076219.966325, 0.000000]: Loading controller: joint1_position
[INFO] [1559076219.987955, 0.000000]: Loading controller: joint2_position
[INFO] [1559076219.998472, 0.000000]: Loading controller: joint3_position
[INFO] [1559076220.006723, 0.000000]: Loading controller: joint4_position
[INFO] [1559076220.014447, 0.000000]: Loading controller: gripper_position
[INFO] [1559076220.029653, 0.000000]: Loading controller: gripper_sub_position
[INFO] [1559076220.056688, 0.000000]: Controller Spawner: Loaded controllers: joint_state_cont
roller, joint1_position, joint2_position, joint3_position, joint4_position, gripper_position,
gripper_sub_position
[INFO] [1559076315.128516, 0.000000]: Started controllers: joint_state_controller, joint1_posi
tion, joint2_position, joint3_position, joint4_position, gripper_position, gripper_sub_positio
n
[ INFO] [1559076315.245767647, 0.033000000]: waitForService: Service [/gazebo/set_physics_prop
erties] is now available.
[ INFO] [1559076315.343495412, 0.051000000]: Physics dynamic reconfigure ready.
```

Figure 18: TB3 and Arm send command to joint

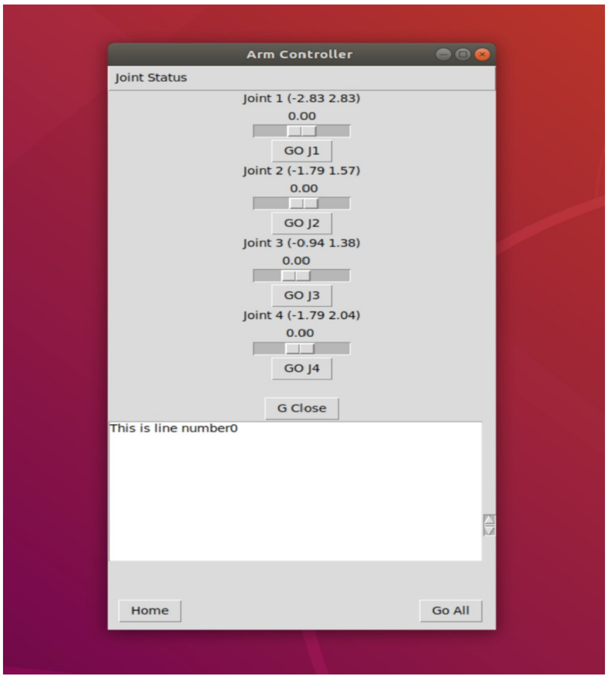
→ Last step build a GUI to control the arm

We are going to use python and you simple need to copy and paste the following code gui.py See Anex1

http://wiki.ros.org/pr2_controllers/Tutorials/Getting%20the%20current%20joint%20angles
http://wiki.ros.org/joint_state_publisher

For the joint_states_listenner.py you can use the Anex2
You need to open 3 windows
1)run `roscore,`
2) run `rosrun joint_states_listener joint_states_listener.py`
If its not working try to put the dir like `nodes/joint_states_listener.py`
Do not forget to make the file executable, right click permission and then check box.
3)please put your dir right and run
`python project/src/joint_states_listener/scripts/gui.py`
If its not working try in 2 and 3
`source /home/”your name”/catkin_ws/devel/setup.bash`

Now you will see this amazing gui!



This is line number0

Figure 19: GUI

→ Final Part all together now!

Finally to run everithing open terminator with 5 windows as show in the picture
1)run `roscore,`

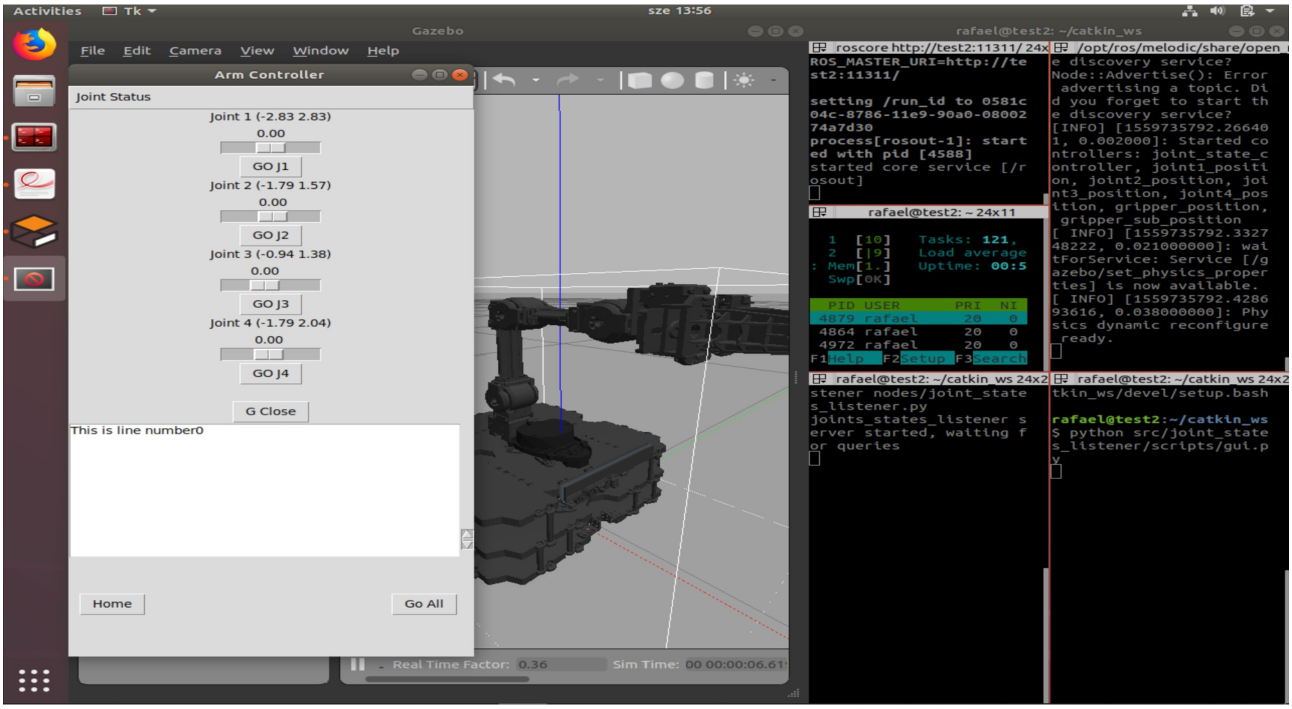
2) optional run `htop`,

3)
```
source /home/rr/project/devel/setup.bash
export TURTLEBOT3_MODEL='waffle'
roslaunch open_manipulator_with_tb3_gazebo empty_world.launch
```
press play!!!

4) run `rosrun joint_states_listener nodes/joint_states_listener.py`

5) please put your dir right and run

```
python project/src/joint_states_listener/scripts/gui.py
```



Joint Status

Joint 1 (-2.83 2.83)
0.00

Joint 2 (-1.79 1.57)
0.00

Joint 3 (-0.94 1.38)
0.00

Joint 4 (-1.79 2.04)
0.00

This is line number0

Real Time Factor: 0.36 Sim Time: 00 00:00:06.61

```
roscore http://test2:11311/ 24x
ROS_MASTER_URI=http://te
st2:11311/

setting /run_id to 0581c
04c-8786-11e9-90a0-08002
74a7d30
process[rosout-1]: start
ed with pid [4588]
started core service [/r
osout]
```

```
rafael@test2: ~ 24x11

  1  [10]    Tasks: 121,
  2  [|9]    Load average
: Mem[1.]    Uptime: 00:5
  Swp[0K]

  PID USER          PRI  NI
 4879 rafael         20   0
 4864 rafael         20   0
 4972 rafael         20   0
F1Help  F2Setup  F3Search
```

```
/opt/ros/melodic/share/open
e discovery service?
Node::Advertise(): Error
 advertising a topic. Di
d you forget to start th
e discovery service?
[INFO] [1559735792.26640
1, 0.002000]: Started co
ntrollers: joint_state_c
ontroller, joint1_positi
on, joint2_position, joi
nt3_position, joint4_pos
ition, gripper_position,
 gripper_sub_position
[ INFO] [1559735792.3327
48222, 0.021000000]: wai
tForService: Service [/g
azebo/set_physics_proper
ties] is now available.
[ INFO] [1559735792.4286
93616, 0.038000000]: Phy
sics dynamic reconfigure
 ready.
```

```
rafael@test2: ~/catkin_ws 24x2
stener nodes/joint_state
s_listener.py
joints_states_listener s
erver started, waiting f
or queries
```

```
rafael@test2: ~/catkin_ws 24x2
tkin_ws/devel/setup.bash

rafael@test2:~/catkin_ws
$ python src/joint_state
s_listener/scripts/gui.p
y
```

Figure 20: Final Result

Anex 1

```
############################################
#   Author: Rafael Rodrigues               #
#   Professor:                             #
#   Course:                                #
#   Uni:                                   #
############################################

import roslib
roslib.load_manifest('joint_states_listener')
import rospy
import time
import sys
from sensor_msgs.msg import JointState
from std_msgs.msg import Float64
from std_msgs.msg import String
from joint_states_listener.srv import ReturnJointStates
#python2 for python 3 use just tkinter
from Tkinter import *
import Tkinter as tkinter


#################################################
#       Main window Defeniton                   #
#                                               #
#################################################

window = tkinter.Tk()
window.title("Arm Controller ")
window.geometry("400x700")

# icon not working window.wm_iconbitmap('/home/rr/favicon.ico')

#################################################
#   Joint State                                 #
#   Method for calling service                  #
#   Define  Joints from the service             #
#   Getting the position                        #
#   Print Position to the list                  #
#################################################


def call_return_joint_states(joint_names):
    rospy.wait_for_service("return_joint_states")
    try:
        s = rospy.ServiceProxy("return_joint_states", ReturnJointStates)
        resp = s(joint_names)
    except rospy.ServiceException, e:
        print "error when calling return_joint_states: %s"%e
        sys.exit(1)
    for (ind, joint_name) in enumerate(joint_names):
        if(not resp.found[ind]):
            print "joint %s not found!"%joint_name
    return (resp.position, resp.velocity, resp.effort)


#pretty-print list to string
def pplist(list):
    return ' '.join(['%2.2f'%x for x in list])


if __name__ == "__main__":
    joint1 = ["joint1"]
    joint2 = ["joint2"]
    joint3 = ["joint3"]
    joint4 = ["joint4"]

def print1act():
    (position, velocity, effort) = call_return_joint_states(joint1)
    mylist.insert(END, 'Joint 1 position: ' + str(pplist(position)))
    mylist.see(END)


def print2act():
    (position, velocity, effort) = call_return_joint_states(joint2)
    mylist.insert(END, 'Joint 2 position: ' + str(pplist(position)))
    mylist.see(END)


def print3act():
    (position, velocity, effort) = call_return_joint_states(joint3)
    mylist.insert(END, 'Joint 3 position: ' + str(pplist(position)))
    mylist.see(END)


def print4act():
    (position, velocity, effort) = call_return_joint_states(joint4)
    mylist.insert(END, 'Joint 4 position: ' + str(pplist(position)))
    mylist.see(END)

def printall():
    print1act()
    print2act()
    print3act()
    print4act()
```

```
#################################################
#       First Joint                             #
#   Define Label                                #
#   Define Scale                                #
#   Define Method that send entry value         #
#   to the Joint                                #
#   Define the Submit button                    #
#################################################

lbl1 = tkinter.Label(window, text='Joint 1 (-2.83 2.83)')
lbl1.pack()

#first entry if you need free input source
#ent1 = tkinter.Entry(window)
#ent1.pack()

sj1 = tkinter.Scale(window, from_=-2.83, to=2.83, digits=3 ,resolution=0.01,
orient=HORIZONTAL)
sj1.pack()


def joint1send():
    pub = rospy.Publisher('/om_with_tb3/joint1_position/command', Float64,
queue_size=10)
        rospy.init_node('talker', anonymous=True)
        j1_str = float( sj1.get())
        pub.publish(j1_str)
    mylist.insert(END, 'Joint 1 Sended: '+  str(j1_str))
    mylist.see(END)

btn1 = tkinter.Button(window, text='GO J1',command=joint1send)
btn1.pack()

#################################################
#       Second Joint                            #
#   Define Label                                #
#   Define Entry                                #
#   Define Method that send entry value         #
#   to the Joint                                #
#   Define the Submit button                    #
#################################################

#Second label
lbl2 = tkinter.Label(window, text='Joint 2 (-1.79 1.57)')
lbl2.pack()

sj2 = tkinter.Scale(window, from_=-1.79, to=1.57, digits=3 ,resolution=0.01,
orient=HORIZONTAL)
sj2.pack()
```

```
def joint2send():

    pub = rospy.Publisher('/om_with_tb3/joint2_position/command', Float64,
queue_size=10)
        rospy.init_node('talker', anonymous=True) #possible not needed but can
be the first in used
        j2_str = float( sj2.get())
        pub.publish(j2_str)
    mylist.insert(END, 'Joint 2 Sended: '+  str(j2_str))
    mylist.see(END)

# Submit joint 2
btn2 = tkinter.Button(window, text='GO J2',command=joint2send)
btn2.pack()
```

```
#################################################
#       Third Joint                             #
#   Define Label                                #
#   Define Entry                                #
#   Define Method that send entry value         #
#   to the Joint                                #
#   Define the Submit button                    #
#################################################

lbl3 = tkinter.Label(window, text='Joint 3 (-0.94 1.38)')
lbl3.pack()

sj3 = tkinter.Scale(window, from_=-0.94, to=1.38, digits=3 ,resolution=0.01,
orient=HORIZONTAL)
sj3.pack()

def joint3send():

    pub = rospy.Publisher('/om_with_tb3/joint3_position/command', Float64,
queue_size=10)
        rospy.init_node('talker', anonymous=True)
        j3_str = float( sj3.get())
        pub.publish(j3_str)
    mylist.insert(END, 'Joint 3 Sended: '+  str(j3_str))
    mylist.see(END)

btn3 = tkinter.Button(window, text='GO J3',command=joint3send)
btn3.pack()
```

```
################################################
#       Fourth Joint                           #
#   Define Label                               #
#   Define Entry                               #
#   Define Method that send entry value        #
#   to the Joint                               #
#   Define the Submit button                   #
################################################


lbl4 = tkinter.Label(window, text='Joint 4 (-1.79 2.04)')
lbl4.pack()

sj4 = tkinter.Scale(window, from_=-1.79, to=2.04, digits=3 ,resolution=0.01,
orient=HORIZONTAL)
sj4.pack()

def joint4send():

    pub = rospy.Publisher('/om_with_tb3/joint4_position/command', Float64,
queue_size=10)
        rospy.init_node('talker', anonymous=True)
        j4_str = float( sj4.get())
        pub.publish(j4_str)
    mylist.insert(END, 'Joint 4 Sended: '+  str(j4_str))
    mylist.see(END)


btn4 = tkinter.Button(window, text='GO J4',command=joint4send)
btn4.pack()



################################################
#        Gripper Button                        #
#   Send Comand Gripper open/close             #
################################################
lbl5 = tkinter.Label(window, text='')
lbl5.pack()

def update_btn_text():
    global b
    btn5_text.set("G Open")
    pub = rospy.Publisher('/om_with_tb3/gripper_position/command', Float64,
queue_size=10)
        rospy.init_node('talker', anonymous=True)
        gripper_str = float(-0.01)
        pub.publish(gripper_str)
    mylist.insert(END, 'Gripper sended: Close '+  str(gripper_str))
    mylist.see(END)
    b+=1
    if b%2 :
      btn5_text.set("G Close")
      pub = rospy.Publisher('/om_with_tb3/gripper_position/command', Float64, 
queue_size=10)
          rospy.init_node('talker', anonymous=True)
          gripper_str = float(0.01)
          pub.publish(gripper_str)
      mylist.insert(END, 'Gripper sended: Open' + str(gripper_str))
          mylist.see(END)



b=1
btn5_text = tkinter.StringVar()
btn5 = tkinter.Button(window, textvariable=btn5_text, command=update_btn_text)
btn5_text.set("G Close")
btn5.pack()


#################################################
#        Home Button                            #
#   Send All Joints to 0 Position               #
#################################################
def Home():
    pub = rospy.Publisher('/om_with_tb3/joint1_position/command', Float64, 
queue_size=10)
        rospy.init_node('talker', anonymous=True)
        hello_str = float(0)
        pub.publish(hello_str)
    pub = rospy.Publisher('/om_with_tb3/joint2_position/command', Float64, 
queue_size=10)
    rospy.init_node('talker', anonymous=True)
    pub.publish(hello_str)
    pub = rospy.Publisher('/om_with_tb3/joint3_position/command', Float64, 
queue_size=10)
    rospy.init_node('talker', anonymous=True)
    pub.publish(hello_str)
    pub = rospy.Publisher('/om_with_tb3/joint4_position/command', Float64, 
queue_size=10)
        pub.publish(hello_str)



btn6 = tkinter.Button(window, text='Home',command=Home).place(x=10,y=660)
```

```
#################################################
#        ALL Button                             #
#   Send All Joints to X Position               #
#################################################
def All():
    joint1send()
    joint2send()
    joint3send()
    joint4send()


btn7 = tkinter.Button(window, text='Go All',command=All).place(x=320,y=660)
```

```
#################################################
#       Scroll Bar                              #
#   Define scroll bar and list                  #
#################################################


scrollbar = tkinter.Scrollbar(window)
scrollbar.pack(side = RIGHT)
mylist = Listbox(window, yscrollcommand = scrollbar.set )
for line in range(1):
   mylist.insert(END, 'This is line number' + str(line))
   mylist.see(END)
mylist.pack(fill= X )
scrollbar.config( command = mylist.yview )

def a (c):
    mylist.insert(END, 'This is line number' + str(c))

#################################################
#            Menu                               #
#           Joint  Position                     #
#################################################
menubar = tkinter.Menu(window)
jointmenu = tkinter.Menu(menubar, tearoff=0)
jointmenu.add_command(label="Joint1" , command=print1act)
jointmenu.add_command(label="Joint2" , command= print2act)
jointmenu.add_command(label="Joint3" , command= print3act)
jointmenu.add_command(label="Joint4" , command= print4act)
jointmenu.add_command(label="All Joint" , command= printall)

menubar.add_cascade(label="Joint Status", menu=jointmenu)
window.config(menu=menubar)
```

```
"""
Not in use but can be useful
m1 = tkinter.PanedWindow()
m1.pack(fill = BOTH, expand = 1)
left = Entry(m1, bd = 5)
m1.add(left)
m2 = tkinter.PanedWindow(m1, orient = VERTICAL)
m1.add(m2)
top = tkinter.Scale( m2, orient = HORIZONTAL)
m2.add(top)
"""
window.mainloop()
```

Anex2

```
import roslib
roslib.load_manifest('joint_states_listener')
import rospy
from joint_states_listener.srv import *
from sensor_msgs.msg import JointState
import threading

#holds the latest states obtained from joint_states messages
class LatestJointStates:

    def __init__(self):
        rospy.init_node('joint_states_listener')
        self.lock = threading.Lock()
        self.name = []
        self.position = []
        self.velocity = []
        self.effort = []
        self.thread = threading.Thread(target=self.joint_states_listener)
        self.thread.start()

        s = rospy.Service('return_joint_states', ReturnJointStates,
self.return_joint_states)


    #thread function: listen for joint_states messages
    def joint_states_listener(self):
        rospy.Subscriber('/om_with_tb3/joint_states', JointState,
self.joint_states_callback)
        rospy.spin()


    #callback function: when a joint_states message arrives, save the values
    def joint_states_callback(self, msg):
        self.lock.acquire()
        self.name = msg.name
        self.position = msg.position
        self.velocity = msg.velocity
        self.effort = msg.effort
        self.lock.release()


    #returns (found, position, velocity, effort) for the joint joint_name
    #(found is 1 if found, 0 otherwise)
    def return_joint_state(self, joint_name):

        #no messages yet
        if self.name == []:
            rospy.logerr("No robot_state messages received!\n")
            return (0, 0., 0., 0.)
        #return info for this joint
        self.lock.acquire()
        if joint_name in self.name:
            index = self.name.index(joint_name)
            position = self.position[index]
            velocity = self.velocity[index]
            effort = self.effort[index]

        #unless it's not found
        else:
            rospy.logerr("Joint %s not found!", (joint_name,))
            self.lock.release()
            return (0, 0., 0., 0.)
        self.lock.release()
        return (1, position, velocity, effort)


    #server callback: returns arrays of position, velocity, and effort
    #for a list of joints specified by name
    def return_joint_states(self, req):
        joints_found = []
        positions = []
        velocities = []
        efforts = []
        for joint_name in req.name:
            (found, position, velocity, effort) =
self.return_joint_state(joint_name)
            joints_found.append(found)
            positions.append(position)
            velocities.append(velocity)
            efforts.append(effort)
        return ReturnJointStatesResponse(joints_found, positions, velocities,
efforts)


#run the server
if __name__ == "__main__":

    latestjointstates = LatestJointStates()

    print "joints_states_listener server started, waiting for queries"
    rospy.spin()
```