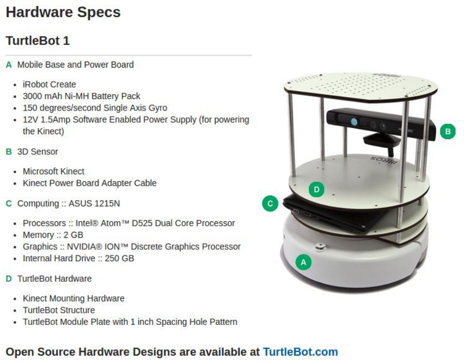
Simulador de Turtlebot

Clase 7
Ing. Alexander López

Turtlebot robot



Hardware Specs

TurtleBot 1

A Mobile Base and Power Board

- iRobot Create
- 3000 mAh Ni-MH Battery Pack
- 150 degrees/second Single Axis Gyro
- 12V 1.5Amp Software Enabled Power Supply (for powering the Kinect)

B 3D Sensor

- Microsoft Kinect
- Kinect Power Board Adapter Cable

C Computing :: ASUS 1215N

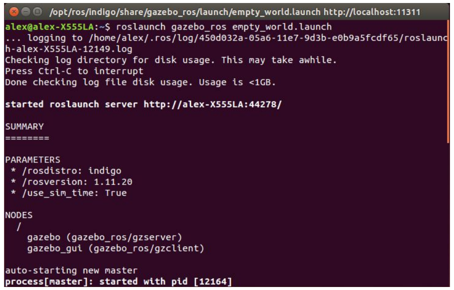
- Processors :: Intel® Atom™ D525 Dual Core Processor
- Memory :: 2 GB
- Graphics :: NVIDIA® ION™ Discrete Graphics Processor
- Internal Hard Drive :: 250 GB

D TurtleBot Hardware

- Kinect Mounting Hardware
- TurtleBot Structure
- TurtleBot Module Plate with 1 inch Spacing Hole Pattern

Open Source Hardware Designs are available at TurtleBot.com

Gazebo



```
/opt/ros/indigo/share/gazebo_ros/launch/empty_world.launch http://localhost:11311
alex@alex-X555LA:~$ roslaunch gazebo_ros empty_world.launch
... logging to /home/alex/.ros/log/450d032a-05a6-11e7-9d3b-e0b9a5fcdf65/roslaunc
h-alex-X555LA-12149.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://alex-X555LA:44278/

SUMMARY
========

PARAMETERS
 * /rosdistro: indigo
 * /rosversion: 1.11.20
 * /use_sim_time: True

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

auto-starting new master
process[master]: started with pid [12164]
```

killall gzserver

Gazebo

Version resumida:

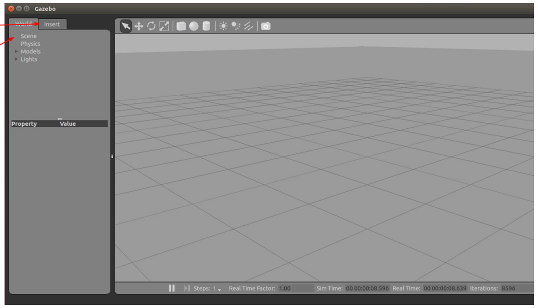
```
<launch>
	<!-- start gazebo with an empty plane -->
	<param name="/use_sim_time" value="true" />
	<node name="gazebo" pkg="gazebo" type="gazebo" args="$(find
gazebo_worlds)/worlds/empty.world" respawn="false" output="screen"/>
</launch>
```

Gazebo

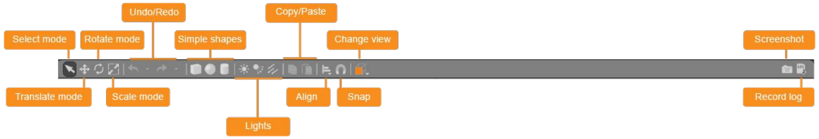
El panel derecho puede utilizarse para interactuar con las partes móviles de un modelo seleccionado (las articulaciones).

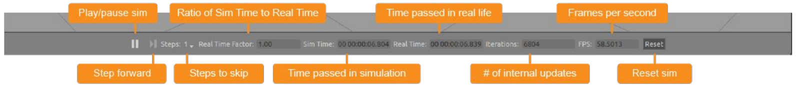
Insertar Tab: Es donde agrega nuevos objetos (modelos) a la simulación.

World Tab: muestra los modelos que se encuentran actualmente en la escena y le permite ver y modificar los parámetros del modelo, como su pose.

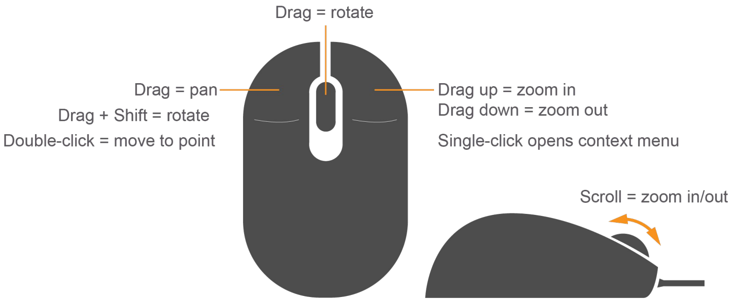


Gazebo

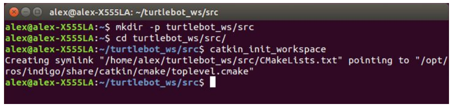




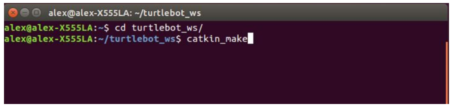
Gazebo



Crear un paquete

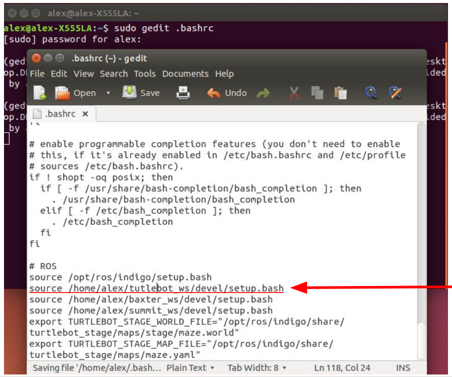


```
alex@alex-X555LA: ~/turtlebot_ws/src
alex@alex-X555LA:~$ mkdir -p turtlebot_ws/src
alex@alex-X555LA:~$ cd turtlebot_ws/src/
alex@alex-X555LA:~/turtlebot_ws/src$ catkin_init_workspace
Creating symlink "/home/alex/turtlebot_ws/src/CMakeLists.txt" pointing to "/opt/
ros/indigo/share/catkin/cmake/toplevel.cmake"
alex@alex-X555LA:~/turtlebot_ws/src$
```



```
alex@alex-X555LA: ~/turtlebot_ws
alex@alex-X555LA:~$ cd turtlebot_ws/
alex@alex-X555LA:~/turtlebot_ws$ catkin_make
```

Crear un paquete



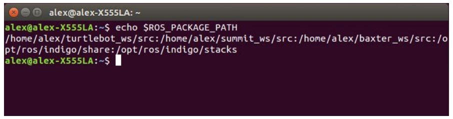
alex@alex-X555LA: ~
.bashrc (~) - gedit
File Edit View Search Tools Documents Help
Open
Save
Undo
.bashrc
Saving file '/home/alex/.bash... Plain Text ▾ Tab Width: 8 ▾ Ln 118, Col 24 INS

```
alex@alex-X555LA:~$ sudo gedit .bashrc
[sudo] password for alex:
```

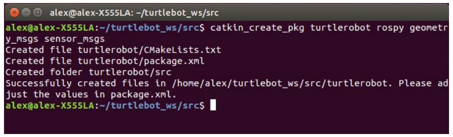
```
# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion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

# ROS
source /opt/ros/indigo/setup.bash
source /home/alex/tutlebot_ws/devel/setup.bash
source /home/alex/baxter_ws/devel/setup.bash
source /home/alex/summit_ws/devel/setup.bash
export TURTLEBOT_STAGE_WORLD_FILE="/opt/ros/indigo/share/
turtlebot_stage/maps/stage/maze.world"
export TURTLEBOT_STAGE_MAP_FILE="/opt/ros/indigo/share/
turtlebot_stage/maps/maze.yaml"
```

Agregar

Crear un nodo



```
alex@alex-X555LA:~$ echo $ROS_PACKAGE_PATH
/home/alex/turtlebot_ws/src:/home/alex/summit_ws/src:/home/alex/baxter_ws/src:/o
pt/ros/indigo/share:/opt/ros/indigo/stacks
alex@alex-X555LA:~$
```



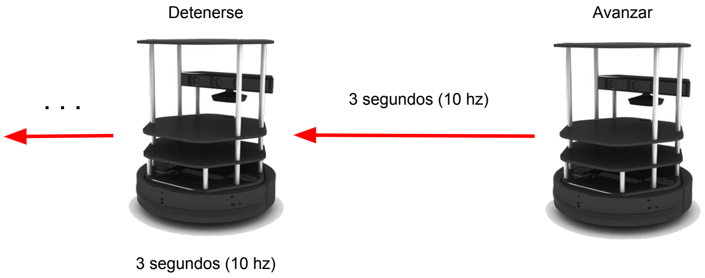
```
alex@alex-X555LA:~/turtlebot_ws/src$ catkin_create_pkg turtlerobot rospy geometr
y_msgs sensor_msgs
Created file turtlerobot/CMakeLists.txt
Created file turtlerobot/package.xml
Created folder turtlerobot/src
Successfully created files in /home/alex/turtlebot_ws/src/turtlerobot. Please ad
just the values in package.xml.
alex@alex-X555LA:~/turtlebot_ws/src$
```

Crear un nodo

¿Que es CMakeLists?

¿Que es package.xml?

Crear un nodo



Crear un nodo

```
#!/usr/bin/env python
import rospy
from geometry_msgs.msg import Twist

cmd_vel_pub = rospy.Publisher('cmd_vel', Twist, queue_size=1)
rospy.init_node('red_light_green_light')

red_light_twist = Twist()
green_light_twist = Twist()
green_light_twist.linear.x = 0.5

driving_forward = False
light_change_time = rospy.Time.now()
rate = rospy.Rate(10)
```

Mensaje tipo Twist, indicado la velocidad de un móvil

El argumento queue_size = 1 le dice a rospy que sólo almacene en un solo búfer mensaje saliente.

Variables tipo Twist se declara con todas sus variables en cero

La variable se cambia su parámetro X para que este avance en el eje correspondiente

Crear un nodo

```
#!/usr/bin/env python
import rospy
from geometry_msgs.msg import Twist

cmd_vel_pub = rospy.Publisher('cmd_vel', Twist, queue_size=1)
rospy.init_node('red_light_green_light')

red_light_twist = Twist()
green_light_twist = Twist()
green_light_twist.linear.x = 0.5

driving_forward = False
light_change_time = rospy.Time.now()
rate = rospy.Rate(10)
```

Variables usada en el condicional para indicar avanzar o detener

Guarda el tiempo actual en una variable

Crear un nodo

```
while not rospy.is_shutdown():
    if driving_forward:
        cmd_vel_pub.publish(green_light_twist)
    else:
        cmd_vel_pub.publish(red_light_twist)
    if light_change_time < rospy.Time.now():
        driving_forward = not driving_forward
        light_change_time = rospy.Time.now() + rospy.Duration(3)
    rate.sleep()
```

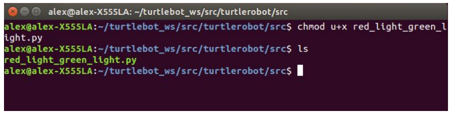
Avanzar

Detenerse

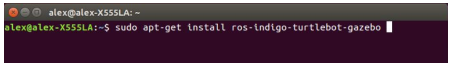
En caso el tiempo se cumpla, se debe aumentar 3 segundos a la cuenta actual

Se agrega 3 segundos a la variable de tiempo

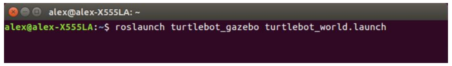
Iniciar Gazebo



```
alex@alex-X555LA:~/turtlebot_ws/src/turtlerobot/src$ chmod u+x red_light_green_l
ight.py
alex@alex-X555LA:~/turtlebot_ws/src/turtlerobot/src$ ls
red_light_green_light.py
alex@alex-X555LA:~/turtlebot_ws/src/turtlerobot/src$
```

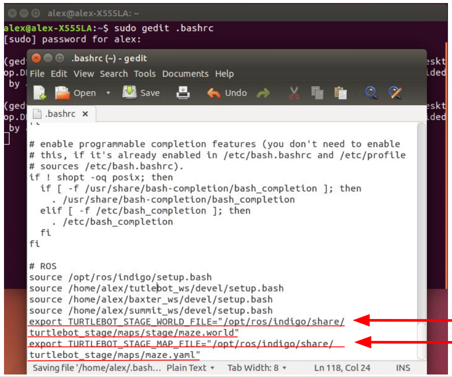


```
alex@alex-X555LA:~$ sudo apt-get install ros-indigo-turtlebot-gazebo
```



```
alex@alex-X555LA:~$ roslaunch turtlebot_gazebo turtlebot_world.launch
```

Iniciar Gazebo


alex@alex-X555LA: ~
.bashrc (~) - gedit
File Edit View Search Tools Documents Help
Open
Save
Undo
.bashrc
Saving file '/home/alex/.bash... Plain Text Tab Width: 8 Ln 118, Col 24 INS

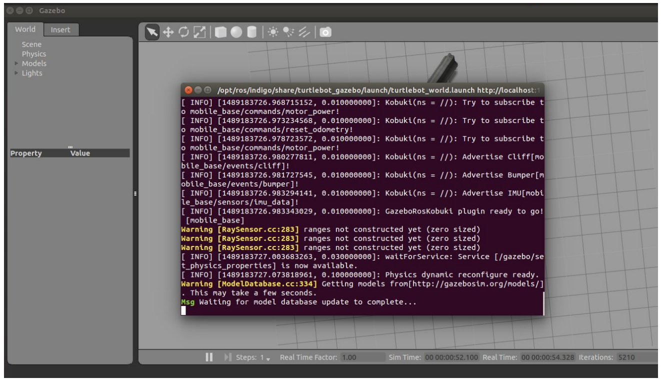
```
alex@alex-X555LA:~$ sudo gedit .bashrc
[sudo] password for alex:
```

```
# 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

# ROS
source /opt/ros/indigo/setup.bash
source /home/alex/tutlebot_ws/devel/setup.bash
source /home/alex/baxter_ws/devel/setup.bash
source /home/alex/summit_ws/devel/setup.bash
export TURTLEBOT_STAGE_WORLD_FILE="/opt/ros/indigo/share/
turtlebot_stage/maps/stage/maze.world"
export TURTLEBOT_STAGE_MAP_FILE="/opt/ros/indigo/share/
turtlebot_stage/maps/maze.yaml"
```

Declarar la variable
TURTLEBOT_STAGE_WORD_FILE

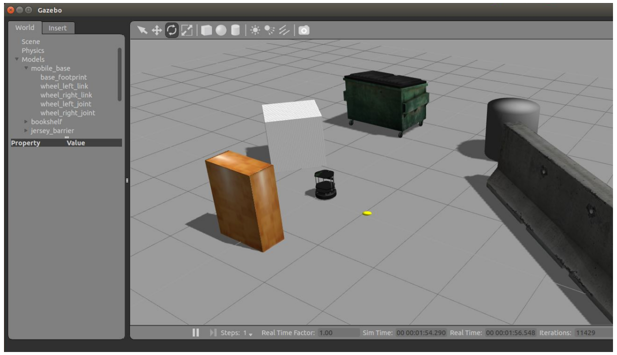
Declarar la variable
TURTLEBOT_STAGE_MAP_FILE

Iniciar Gazebo

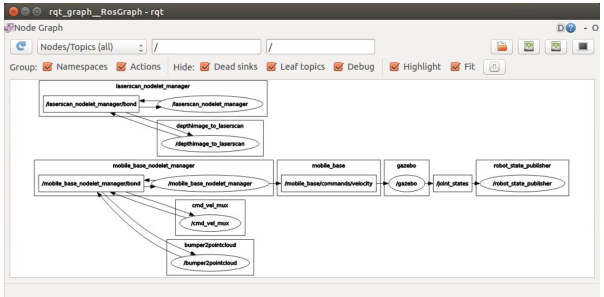


```
/opt/ros/indigo/share/turtlebot_gazebo/launch/turtlebot_world.launch http://localhost:1
[ INFO] [1489183726.968715152, 0.010000000]: Kobuki(ns = //): Try to subscribe t
o mobile_base/commands/motor_power!
[ INFO] [1489183726.973234568, 0.010000000]: Kobuki(ns = //): Try to subscribe t
o mobile_base/commands/reset_odometry!
[ INFO] [1489183726.978723572, 0.010000000]: Kobuki(ns = //): Try to subscribe t
o mobile_base/commands/motor_power!
[ INFO] [1489183726.980277811, 0.010000000]: Kobuki(ns = //): Advertise Cliff[mo
bile_base/events/cliff]!
[ INFO] [1489183726.981727545, 0.010000000]: Kobuki(ns = //): Advertise Bumper[m
obile_base/events/bumper]!
[ INFO] [1489183726.983294141, 0.010000000]: Kobuki(ns = //): Advertise IMU[mobi
le_base/sensors/imu_data]!
[ INFO] [1489183726.983343029, 0.010000000]: GazeboRosKobuki plugin ready to go!
 [mobile_base]
Warning [RaySensor.cc:283] ranges not constructed yet (zero sized)
Warning [RaySensor.cc:283] ranges not constructed yet (zero sized)
Warning [RaySensor.cc:283] ranges not constructed yet (zero sized)
[ INFO] [1489183727.003683263, 0.030000000]: waitForService: Service [/gazebo/se
t_physics_properties] is now available.
[ INFO] [1489183727.073818961, 0.100000000]: Physics dynamic reconfigure ready.
Warning [ModelDatabase.cc:334] Getting models from[http://gazebosim.org/models/]
. This may take a few seconds.
Msg Waiting for model database update to complete...
```

Iniciar Gazebo



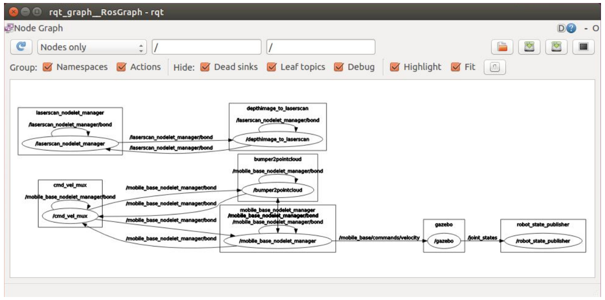
Iniciar Gazebo


rqt_graph__RosGraph - rqt
Node Graph
laserscan_nodelet_manager
/laserscan_nodelet_manager/bond
/laserscan_nodelet_manager
depthimage_to_laserscan
/depthimage_to_laserscan
mobile_base_nodelet_manager
/mobile_base_nodelet_manager/bond
/mobile_base_nodelet_manager
mobile_base
/mobile_base/commands/velocity
gazebo
/gazebo
/joint_states
robot_state_publisher
/robot_state_publisher
cmd_vel_mux
/cmd_vel_mux
bumper2pointcloud
/bumper2pointcloud

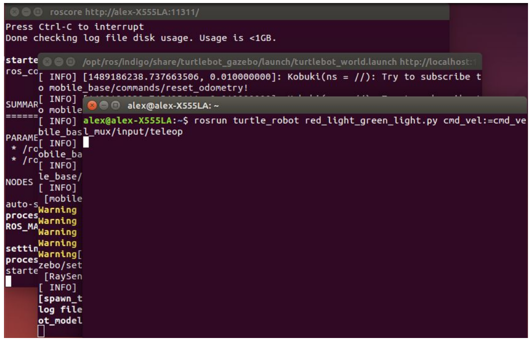
Nodes/Topics (all) / /

Group: ☑ Namespaces ☑ Actions Hide: ☑ Dead sinks ☑ Leaf topics ☑ Debug ☑ Highlight ☑ Fit

Iniciar Gazebo


rqt_graph__RosGraph - rqt
Node Graph
Nodes only
/
/
Group: ☑ Namespaces ☑ Actions
Hide: ☑ Dead sinks ☑ Leaf topics ☑ Debug
☑ Highlight ☑ Fit
laserscan_nodelet_manager
/laserscan_nodelet_manager/bond
/laserscan_nodelet_manager
/laserscan_nodelet_manager/bond
/laserscan_nodelet_manager/bond
depthimage_to_laserscan
/laserscan_nodelet_manager/bond
/depthimage_to_laserscan
bumper2pointcloud
/mobile_base_nodelet_manager/bond
/bumper2pointcloud
cmd_vel_mux
/mobile_base_nodelet_manager/bond
/cmd_vel_mux
/mobile_base_nodelet_manager/bond
/mobile_base_nodelet_manager/bond
/mobile_base_nodelet_manager/bond
/mobile_base_nodelet_manager/bond
mobile_base_nodelet_manager
/mobile_base_nodelet_manager/bond
/mobile_base_nodelet_manager/bond
/mobile_base_nodelet_manager
/mobile_base/commands/velocity
gazebo
/gazebo
/joint_states
robot_state_publisher
/robot_state_publisher

Ejecutar el nodo

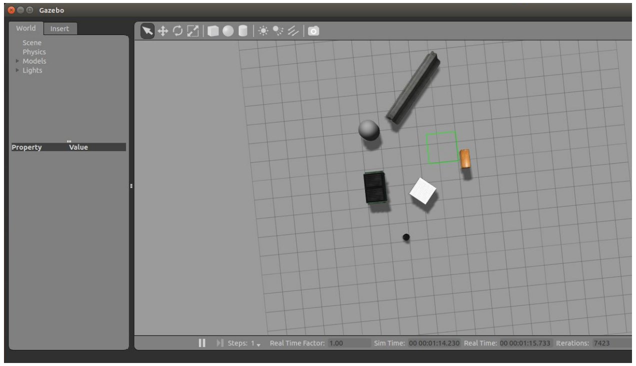

roscore http://alex-X555LA:11311/
/opt/ros/indigo/share/turtlebot_gazebo/launch/turtlebot_world.launch http://localhost:1
alex@alex-X555LA: ~

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

starte
ros_co


SUMMAR
======

PARAME
 * /ro
 * /ro

NODES

auto-s
proces
ROS_MA

settin
proces
starte
```

```
[ INFO] [1489186238.737663506, 0.010000000]: Kobuki(ns = //): Try to subscribe t
o mobile_base/commands/reset_odometry!
[ INFO]
o mobile
[ INFO]
bile_bas
[ INFO]
obile_ba
[ INFO]
le_base/
[ INFO]
 [mobile
Warning
Warning
Warning
Warning
Warning[
zebo/set
 [RaySen
[ INFO]
[spawn_t
log file
ot_model
```

```
alex@alex-X555LA:~$ rosrun turtle_robot red_light_green_light.py cmd_vel:=cmd_ve
l_mux/input/teleop
```

Ejecutar el nodo



Leer data de los sensores

```
#!/usr/bin/env python
import rospy
from sensor_msgs.msg import LaserScan

def scan_callback(msg):
    range_ahead = msg.ranges[len(msg.ranges)/2]
    print “range ahead: %0.1f” % range_ahead

rospy.init_node('range_ahead')
scan_sub = rospy.Subscriber('scan', LaserScan, scan_callback)
rospy.spin()
```

Se importa el tipo de mensaje para Lasers

Se determina la distancia del objeto frente al robot

Suscribirse al tópico Scan y se establece una función callback

Leer data de los sensores

```
def scan_callback(msg):
    range_ahead = msg.ranges[len(msg.ranges)/2]
    print “range ahead: %0.1f” % range_ahead
```

Alternativas:

```
bearing = msg.angle_min + msg.angle_max * i / len(msg.ranges)
```

Ángulo del índice (i) calculado a partir del ángulo máximo y mínimo

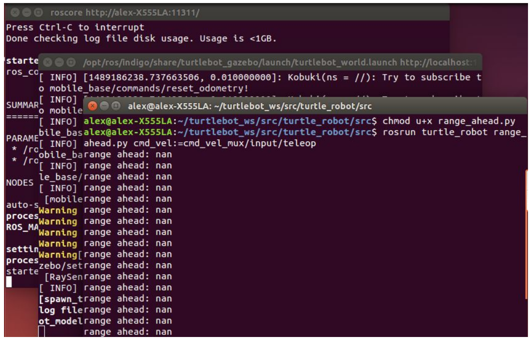
```
range_ahead = msg.ranges[len(msg.ranges)/2]
```

Distancia del robot al objeto más cercano

```
closest_range = min(msg.ranges)
```

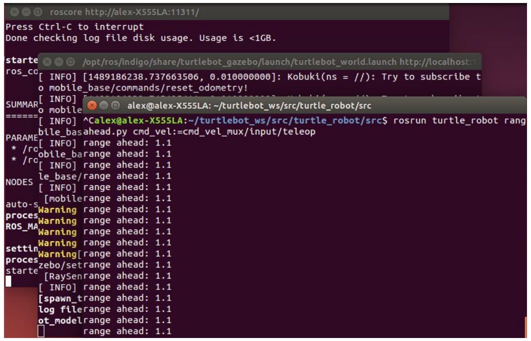
Punto mas cercano al robot

Leer data de los sensores


roscore http://alex-X555LA:11311/
/opt/ros/indigo/share/turtlebot_gazebo/launch/turtlebot_world.launch http://localhost:1
alex@alex-X555LA: ~/turtlebot_ws/src/turtle_robot/src

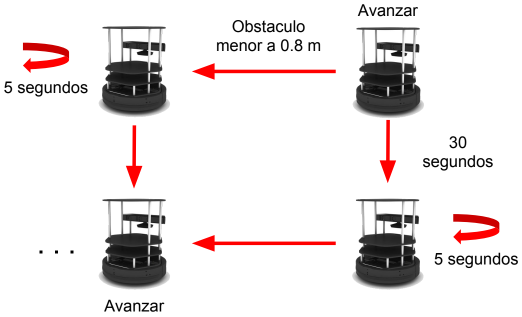
```
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

starte
ros_co[ INFO] [1489186238.737663506, 0.010000000]: Kobuki(ns = //): Try to subscribe t
      o mobile_base/commands/reset_odometry!
SUMMAR[ INFO]
      o mobile
======[ INFO] alex@alex-X555LA:~/turtlebot_ws/src/turtle_robot/src$ chmod u+x range_ahead.py
      bile_basalex@alex-X555LA:~/turtlebot_ws/src/turtle_robot/src$ rosrun turtle_robot range_
PARAME[ INFO] ahead.py cmd_vel:=cmd_vel_mux/input/teleop
 * /roobile_barange ahead: nan
 * /ro[ INFO] range ahead: nan
      le_base/range ahead: nan
NODES [ INFO] range ahead: nan
       [mobilerange ahead: nan
auto-sWarning range ahead: nan
procesWarning range ahead: nan
ROS_MAWarning range ahead: nan
      Warning range ahead: nan
settinWarning[range ahead: nan
procesZebo/setrange ahead: nan
starte [RaySenrange ahead: nan
      [ INFO] range ahead: nan
      [spawn_trange ahead: nan
      log filerange ahead: nan
      ot_modelrange ahead: nan
              range ahead: nan
```

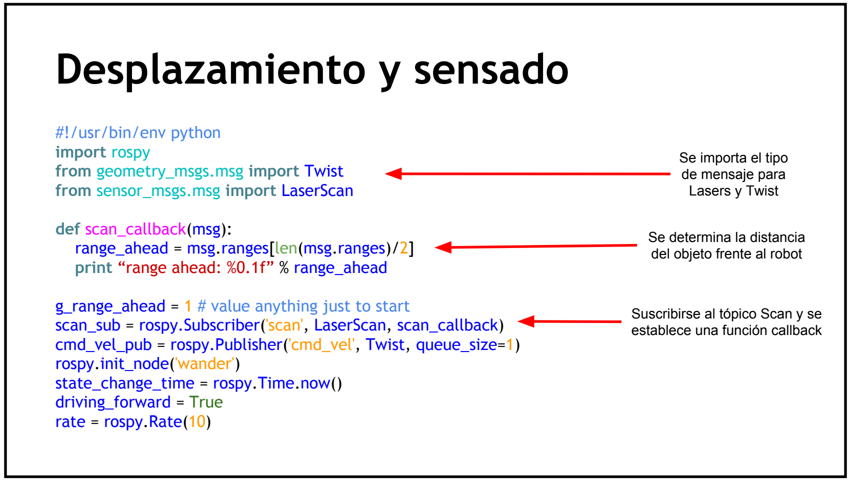
Leer data de los sensores


roscore http://alex-X555LA:11311/
/opt/ros/indigo/share/turtlebot_gazebo/launch/turtlebot_world.launch http://localhost:1
alex@alex-X555LA: ~/turtlebot_ws/src/turtle_robot/src

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

starte
ros_co[ INFO] [1489186238.737663506, 0.010000000]: Kobuki(ns = //): Try to subscribe t
      o mobile_base/commands/reset_odometry!
SUMMAR[ INFO]
      o mobile
======[ INFO] ^Calex@alex-X555LA:~/turtlebot_ws/src/turtle_robot/src$ rosrun turtle_robot rang
      bile_basahead.py cmd_vel:=cmd_vel_mux/input/teleop
PARAME[ INFO] range ahead: 1.1
 * /roobile_barange ahead: 1.1
 * /ro[ INFO] range ahead: 1.1
      le_base/range ahead: 1.1
NODES [ INFO] range ahead: 1.1
       [mobilerange ahead: 1.1
auto-sWarning range ahead: 1.1
procesWarning range ahead: 1.1
ROS_MAWarning range ahead: 1.1
      Warning range ahead: 1.1
settinWarning[range ahead: 1.1
procesZebo/setrange ahead: 1.1
starte [RaySenrange ahead: 1.1
      [ INFO] range ahead: 1.1
      [spawn_trange ahead: 1.1
      log filerange ahead: 1.1
      ot_modelrange ahead: 1.1
              range ahead: 1.1
```

Desplazamiento y sensado





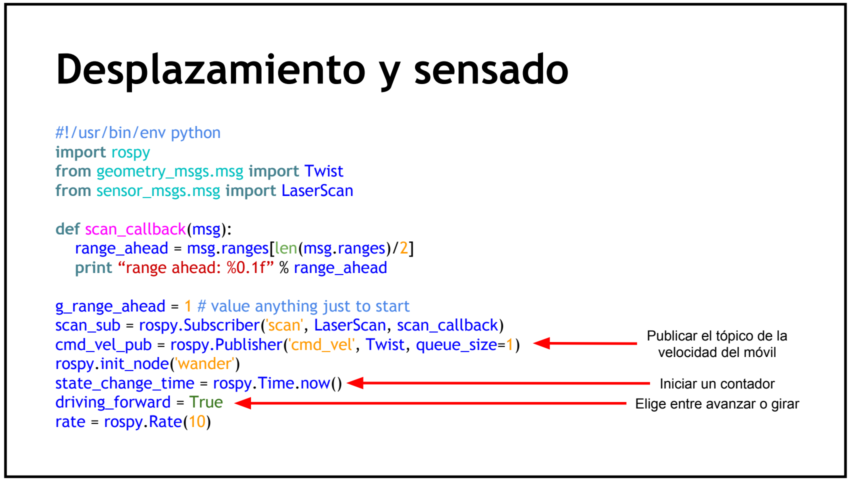
Desplazamiento y sensado

```
#!/usr/bin/env python
import rospy
from geometry_msgs.msg import Twist
from sensor_msgs.msg import LaserScan

def scan_callback(msg):
    range_ahead = msg.ranges[len(msg.ranges)/2]
    print “range ahead: %0.1f” % range_ahead

g_range_ahead = 1 # value anything just to start
scan_sub = rospy.Subscriber('scan', LaserScan, scan_callback)
cmd_vel_pub = rospy.Publisher('cmd_vel', Twist, queue_size=1)
rospy.init_node('wander')
state_change_time = rospy.Time.now()
driving_forward = True
rate = rospy.Rate(10)
```

Se importa el tipo de mensaje para Lasers y Twist

Se determina la distancia del objeto frente al robot

Suscribirse al tópico Scan y se establece una función callback



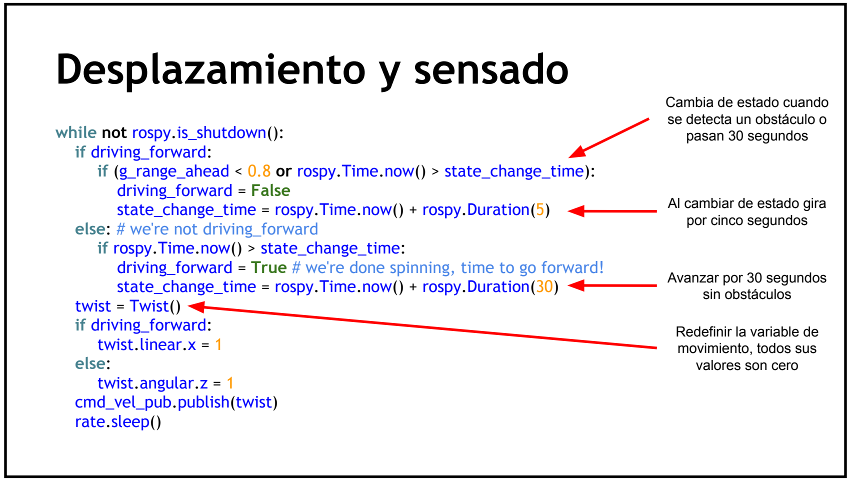
Desplazamiento y sensado

```
#!/usr/bin/env python
import rospy
from geometry_msgs.msg import Twist
from sensor_msgs.msg import LaserScan

def scan_callback(msg):
    range_ahead = msg.ranges[len(msg.ranges)/2]
    print “range ahead: %0.1f” % range_ahead

g_range_ahead = 1 # value anything just to start
scan_sub = rospy.Subscriber('scan', LaserScan, scan_callback)
cmd_vel_pub = rospy.Publisher('cmd_vel', Twist, queue_size=1)
rospy.init_node('wander')
state_change_time = rospy.Time.now()
driving_forward = True
rate = rospy.Rate(10)
```

Publicar el tópico de la velocidad del móvil

Iniciar un contador

Elige entre avanzar o girar



Desplazamiento y sensado

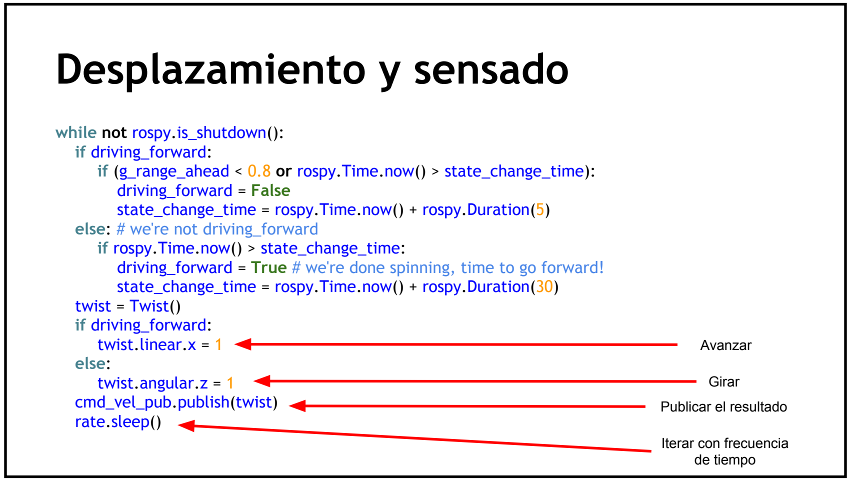
```
while not rospy.is_shutdown():
    if driving_forward:
        if (g_range_ahead < 0.8 or rospy.Time.now() > state_change_time):
            driving_forward = False
            state_change_time = rospy.Time.now() + rospy.Duration(5)
    else: # we're not driving_forward
        if rospy.Time.now() > state_change_time:
            driving_forward = True # we're done spinning, time to go forward!
            state_change_time = rospy.Time.now() + rospy.Duration(30)
    twist = Twist()
    if driving_forward:
        twist.linear.x = 1
    else:
        twist.angular.z = 1
    cmd_vel_pub.publish(twist)
    rate.sleep()
```

Cambia de estado cuando se detecta un obstáculo o pasan 30 segundos

Al cambiar de estado gira por cinco segundos

Avanzar por 30 segundos sin obstáculos

Redefinir la variable de movimiento, todos sus valores son cero



Desplazamiento y sensado

```
while not rospy.is_shutdown():
    if driving_forward:
        if (g_range_ahead < 0.8 or rospy.Time.now() > state_change_time):
            driving_forward = False
            state_change_time = rospy.Time.now() + rospy.Duration(5)
    else: # we're not driving_forward
        if rospy.Time.now() > state_change_time:
            driving_forward = True # we're done spinning, time to go forward!
            state_change_time = rospy.Time.now() + rospy.Duration(30)
    twist = Twist()
    if driving_forward:
        twist.linear.x = 1
    else:
        twist.angular.z = 1
    cmd_vel_pub.publish(twist)
    rate.sleep()
```

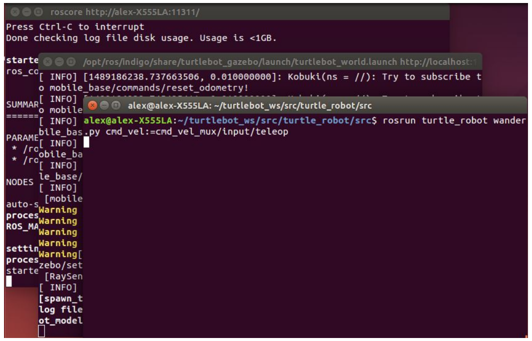
Avanzar

Girar

Publicar el resultado

Iterar con frecuencia de tiempo

Desplazamiento y sensado


roscore http://alex-X555LA:11311/
/opt/ros/indigo/share/turtlebot_gazebo/launch/turtlebot_world.launch http://localhost:1
alex@alex-X555LA: ~/turtlebot_ws/src/turtle_robot/src

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

started
ros_co


SUMMAR
======

PARAME
 * /ro
 * /ro

NODES

auto-s
proces
ROS_MA

settin
proces
starte
```

```
[ INFO] [1489186238.737663506, 0.010000000]: Kobuki(ns = //): Try to subscribe t
o mobile_base/commands/reset_odometry!
[ INFO]
o mobile
[ INFO]
bile_bas
[ INFO]
obile_ba
[ INFO]
le_base/
[ INFO]
 [mobile
Warning
Warning
Warning
Warning
Warning[
zebo/set
 [RaySen
[ INFO]
[spawn_t
log file
ot_model
```

```
alex@alex-X555LA:~/turtlebot_ws/src/turtle_robot/src$ rosrun turtle_robot wander
.py cmd_vel:=cmd_vel_mux/input/teleop
```

¡Gracias!

¡La única pregunta tonta es la que no se hace!