



Proyecto Práctico: Guía de Configuración de los Turtlebots

Profesor: Arturo Bertomeu Motos

Objetivo

Durante las sesiones prácticas de esta asignatura se pretende desarrollar un proyecto en el que el alumno aprenda el uso de ROS como herramienta de programación de robots aplicando los conceptos aprendidos en teoría. De este modo, se pretende desarrollar un paquete de ROS que sea capaz de hacer salir de un laberinto al robot Turtlebot, disponible en el laboratorio de prácticas, mediante el uso del sensor láser Hokuyo UST-10LX que tiene el propio robot. Además, utilizando el robot se aprenderá a utilizar distintas herramientas de ROS como `rqt` o `RViz`, entre otras.

1. Plataforma Kobuki Turtlebot II

La plataforma robótica que se va a utilizar es la plataforma Kobuki Turtlebot II, un robot móvil de bajo coste ideal para educación e investigación, muy utilizada para la iniciación a ROS y disponible en el laboratorio de prácticas.

Este robot contiene una plataforma móvil *Kobuki* que se compone de un sistema de odometría, un giróscopo, tres bumper de colisión, sensores de caída de rueda, leds, ... También dispone de un sensor láser *Hokuyo UST-10LX*, una cámara *Orbbec Astra* y un brazo robótico *PhantomX*

<https://www.roscomponents.com/es/robots-moviles/9-turtlebot-2.html>

2. Turtlebots Disponibles en el Laboratorio

El laboratorio de robótica dispone de 5 Turtlebots, cada uno con una IP asociada, pudiéndose conectar a ellos a través de *SSH*, utilizando la red disponible en el laboratorio. Para conectarse a ellos simplemente lanzamos la comunicación *SSH*, `ssh <usuarioTurtlebot>@<ipTurtlebot>`.

Si no nos podemos conectar al robot, tendremos que comprobar que realmente el robot está accesible desde nuestro ordenador con el comando *PING*, `ping <ipTurtlebot>`.

Red del Laboratorio		
Labrobot5-1	Labrobot5-2	Labrobot_2-4
Contraseña		
labrobot18wifi		

IP Turtlebot				
Robot1	Robot 2	Robot 3	Robot 4	Robot 5
192.168.1.5	192.168.1.6	192.168.1.7	192.168.1.8	192.168.1.9
Usuario				
turtlebot				tb2
Contraseña				
ros				

3. Configuración del TurtleBot

Una vez conectados al robot, deberemos de comprobar que esté bien configurado el entorno ROS antes de lanzar los **.launch* necesarios. Lo primero que debemos de hacer es comprobar que las variables de entorno, *export ROS_MASTER_URI* y *export ROS_HOSTNAME* estén correctamente añadidas mediante el comando *echo* de Ubuntu, *echo \$<variable>*. Estas variables deben tener los siguientes valores:

- *ROS_MASTER_URI=http://<ipTurtlebot>:11311*
- *ROS_HOSTNAME=<ipTurtlebot>*

Si estos valores no están bien configurados, habrá que añadirlos al fichero **.bashrc* o ejecutar el comando *export <variable>=<valor>* en cada nueva terminal. Si decidimos añadir dichas variables al fichero *.bashrc*, hay que ejecutar el comando *source .bashrc* en dicha terminal para reiniciar las variables de entorno de la consola.

¡¡Hay que tener mucho cuidado al editar el fichero *.bashrc* porque podemos desconfigurar el entorno ROS del Turtlebot!!

Tras ajustar estos parámetros, ya podremos lanzar los dos ficheros **.launch* que arrancarán tanto la base móvil como el sensor láser. Cada comando se debe lanzar desde un terminal independiente.

- *roslaunch turtlebot_bringup minimal.launch*
- *roslaunch turtlebot_bringup hokuyo_ust10lx.launch*

De este modo, ya hemos lanzado todos los nodos que se van a necesitar para la realización del proyecto y ya podremos trabajar desde nuestro ordenador.

4. Configuración del Ordenador Personal

Ahora que ya hemos creado el entorno ROS del proyecto, debemos configurar nuestro ordenador para poder acceder a dicho entorno. Para ello debemos configurar dos variables de entorno en nuestro ordenador, *export ROS_MASTER_URI* y *export ROS_IP* con los siguientes valores:

- *ROS_MASTER_URI=http://<ipTurtlebot>:11311*
- *ROS_IP=<ipOrdenadorPersonal>*

Para conocer la dirección IP de nuestro ordenador utilizaremos el comando *ifconfig*. Esta IP debe tener un valor del tipo *192.168.1.X*.

Una vez exportadas estas variables, si ejecutamos el comando *rostopic list* deberemos ver todos los *topics* disponibles en nuestro entorno, y si ejecutamos *rostopic echo /scan* deberemos ver los valores de ese *topic*.

Si todo esto se cumple, ya podemos empezar a programar nuestro nodo para la realización del proyecto.

5. Simulación Turtlebot

Este robot está muy bien documentado en la *wiki* de ROS, por lo que se puede trabajar en simulación desde casa.

<https://wiki.ros.org/Robots/TurtleBot>