

Instalación Turtleboot

Hector Mauricio Yepez Ponce

11 de marzo de 2018

Para la instalación de librerías con las que funciona el turtleboot se debe previamente instalar ROS (hydro, indigo, kinetic, lunar, etc) en el computador que se encuentran en la pagina oficial de ROS <http://www.ros.org/>, inicialmente se instalo Ubuntu 16.04 LTS compatible con ROS Kinetic pero las librerias del turtleboot no estan actualizadas por lo que en un futuro puede presentar errores si se desea añadir sensores, es por eso que se decidió instalar la version Ubuntu 14.04 LTS de 64 bits y se procedió a la instalación siguiendo los pasos que a continuación se detalla:

- **Instalación de paquetes Ubuntu**

- `sudo apt-get install ros-indigo-turtlebot ros-indigo-turtlebot-apps ros-indigo-turtlebot-interactions ros-indigo-turtlebot-simulator ros-indigo-kobuki-ftdi ros-indigo-rocon-remocon ros-indigo-rocon-qt-library ros-indigo-ar-track-alvar-msgs`

- **Preparación**

- `sudo apt-get install python-rosdep python-wstool ros-indigo-ros`
- `sudo rosdep init`
- `rosdep update`

- **Workspaces**

- `mkdir ~/rocon`
- `sudo rosdep init`
- `rosdep update`
- `cd ~/rocon`
- `wstool init -j5 src https://raw.githubusercontent.com/robotics-in-concert/rocon/release/indigo/rocon.rosinstall`
- `source /opt/ros/indigo/setup.bash`
- `rosdep install --from-paths src -i -y`
- `catkin_make`

- `mkdir /kobuki`
- `cd /kobuki`
- `wstool init src -j5`
- `https://raw.githubusercontent.com/yujinrobot/yujin_tools/master/rosinstalls/indigo/kobuki.rosinstall`

- source ~/rocon/devel/setup.bash
- rosdep install --from-paths src -i -y
- catkin_make

- mkdir ~/turtlebot
- cd ~/turtlebot
- wstool init src -j5
- https://raw.githubusercontent.com/yujinrobot/yujin_tools/master/rosinstalls/indigo/turtlebot.rosinstall
- source ~/kobuki/devel/setup.bash
- rosdep install --from-paths src -i -y
- catkin_make

• Configurar el ambiente

- echo "source /opt/ros/indigo/setup.bash" >> ~/.bashrc
- echo "source ~/turtlebot/devel/setup.bash" >> ~/.bashrc

Una vez que se ha completado la instalación de ROS y de Turtleboot en las dos computadoras, se mostrará los para para comunicar la PC de trabajo al ordenador que porta el robot, ejecutar el nodelet principal que activa todos los nodos de comunicación del hardware del robot, como teleoperar el robot con el teclado de nuestro PC.

• Comunicación PC de trabajo/ PC del robot

Para ello lanzaremos el ROS y sus comandos de forma telematica con nuestro PC a través de SHH en Ubuntu. Para ello debemos crear una red LAN inalámbrica propia a la que deber conectado el robot y el PC con el que se que vaya a a trabajar. Por lo tanto, necesitaremos un router WiFi para crear la red.

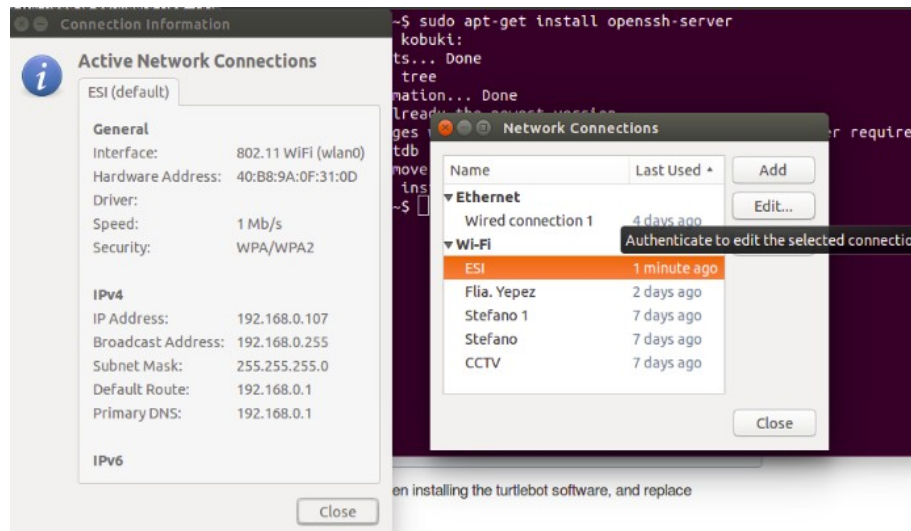
Es necesario instalar SHH server en los dos ordenadores para ello abrimos el terminal y ejecutamos el siguiente comando:

- sudo apt-get install openssh-server

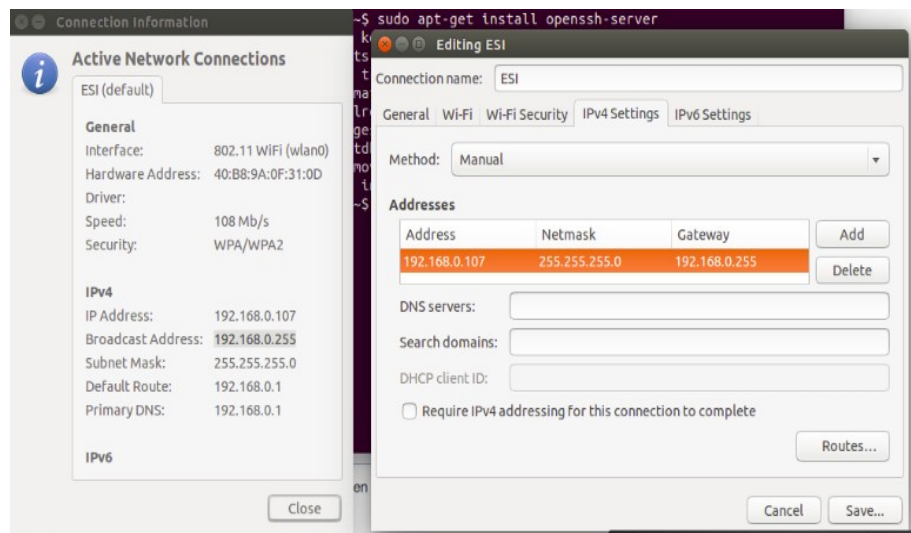
Procedemos averiguar las direcciones IP tanto del PC de trabajo y PC del turtleboot, que permitirá la comunicación entre ambos dispositivos. Para ello ingresamos en la opción conexiones, información de conexión.



Una vez obtenida la información se ingresa a editar conexiones y seleccionamos la red inalámbrica a la cual estamos conectados y presionamos en la opción editar.



Una vez allí escogemos la pestaña de Ajuste de IPv4, en método seleccionamos Manual y a continuación llenamos los campos con la dirección IP para cada dispositivo, guiandonos de la información de conexiones que tenemos desplegada, IP que da acceso a nuestro Router WiFi y en servidor DNS escribimos 8.8.8.8 por ejemplo, aunque en nuestra red no tengamos acceso a Internet, solamente será una conexión inalámbrica por último, clickamos en Guardar. Esto se debe hacer en cada dispositivo.



Ya instalado el SSH server en cada uno de los dispositivos y conectados a la misma red se deberá ejecutar el siguiente comando en el terminal de la PC de trabajo.

- ssh turtle@<TURTLEBOTP.IP>
Donde *turtle* es el nombre de usuario de Ubuntu en la PC del robot, en este caso es **kobuki** y finalmente la dirección IP de la misma que es **192.168.0.107**, quedando de la siguiente manera.
- ssh kobuki@192.168.0.107

Si todo esta bien pedirá la contraseña de la PC del robot, y una vez introducida la contraseña, la ventana del terminal del PC de trabajo se mostrará como si fuera la PC del robot y todo lo que ahí se ejecute será como si se estuviera ejecutando en la PC del robot directamente.

Con la conexión establecida en el terminal de la PC del robot escribimos lo siguiente:

- `echo export ROS_MASTER_URI=http://localhost:11311 >> ~/.bashrc`
- `echo export ROS_HOSTNAME=IP_OF_TURTLEBOT >> ~/.bashrc`
- `echo export ROS_MASTER_URI=http://IP_OF_TURTLEBOT:11311 >> ~/turtlebot/devel/setup.sh`
- `echo export ROS_HOSTNAME=IP_OF_TURTLEBOT >> ~/turtlebot/devel/setup.sh`

Y en *IP_OF_TURTLEBOT* se reemplaza por 192.168.0.107 que es la IP de nuestra PC del robot, quedaría de la siguiente manera:

- `echo export ROS_HOSTNAME=192.168.0.107 >> ~/.bashrc`
- `echo export ROS_MASTER_URI=http://192.168.0.107:11311 >> ~/turtlebot/devel/setup.sh`
- `echo export ROS_HOSTNAME=192.168.0.107 >> ~/turtlebot/devel/setup.sh`

Abrimos un terminal en la PC de trabajo y escribimos los siguiente comandos:

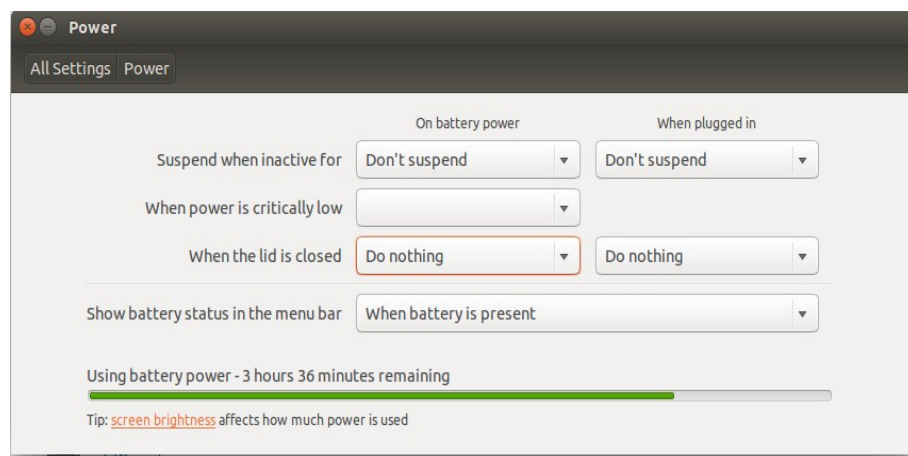
- `echo export ROS_MASTER_URI=http://IP_OF_TURTLEBOT:11311 >> ~/.bashrc`
- `echo export ROS_HOSTNAME=IP_OF_PC >> ~/.bashrc`

De la misma manera en *IP_OF_PC* se reemplaza por la IP de la PC de trabajo en este caso **192.168.0.108**, quedaria de la siguiente manera:

- `echo export ROS_MASTER_URI=http://192.168.0.107:11311 >> ~/.bashrc`
- `echo export ROS_HOSTNAME=192.168.0.108 >> ~/.bashrc`

Una vez hecho todo esto cerramos los terminales abiertos y reiniciamos las dos computadoras.

En la PC del robot se debe configurar que al cerrar la tapa siga funcionando normalmente y no se suspenda el computador para ello, ingresamos a configuracion del sistema, opcion Energia y en la pestaña **Cuando la tapa esta cerrada**, escoger la opción **No hacer nada**.



Ahora encendemos el Kobuki, lo conectamos a la PC del robot, cerramos la tapa de la computadora y lo colocamos en un lugar seguro encima del robot.

– Lanzar los nodos del turtleboot

Para poder enviarle instrucciones al Turtlebot y recibir información de sus sensores y odometría se deben crear nodos de comunicación, con distintos topics y mensajes. Para lanzar los topics correspondientes a cada elemento hardware ejecutamos lo siguiente.

Primero en la PC de trabajo ejecutaremos el comando **roscore** en un nuevo terminal haremos la conexión SSH con la PC del robot.

Una vez establecida la conexión, en la ventana de la PC del robot escribiremos lo siguiente:

- `roslaunch turtlebot_bringup minimal.launch --screen`
- `ls -n /dev | grep kobuki`
- `sudo apt-get install ros-indigo-kobuki ros-indigo-kobuki-core`

Si la vinculación ha sido realizada con éxito el robot producirá un sonido que indicará que los nodos han sido activados y la conexión robot/PC ha sido exitosa.

Ejecutando el comando **rostopic list** se podrá visualizar los nodos que están activos en el turtleboot

Finalmente para teleoperarlo ejecutamos un comando desde la PC del trabajo, ejecutando la librería de teleoperación del turtleboot.

- `roslaunch turtlebot_teleop keyboard_teleop.launch`

```
turtlebot_teleop_keyboard (turtlebot_teleop/turtlebot_teleop_key)
ROS_MASTER_URI=http://localhost:11311
core service [/rosout] found
process[turtlebot_teleop_keyboard-1]: started with pid [3113]

Control Your Turtlebot!
-----
Moving around:
  u    i    o
  j    k    l
  m    ,    .

q/z : increase/decrease max speeds by 10%
w/x : increase/decrease only linear speed by 10%
e/c : increase/decrease only angular speed by 10%
space key, k : force stop
anything else : stop smoothly

CTRL-C to quit

currently:      speed 0.2      turn 1
```