



Duckiebot for dummies

Shortcuts

- Abrir consola: Ctrl + Alt + T
- Autocompletado: Tab || Ej: \$ rosc *tab* -> roscore
- Terminar proceso: Ctrl + C
- Copiar (en la consola): Ctrl + Shift + C
- Pegar (en la consola): Ctrl + Shift + V
- Volver a comandos anteriores: Flecha hacia arriba ↑ (no confundir con shift: ⇧)
- Byobucosas:
 - F2: Crear nueva pestaña
 - F3: Moverse a pestaña anterior
 - F4: Moverse a pestaña siguiente
 - Ctrl + D: Cerrar pestaña (también sirve escribir "exit" en la consola)
 - Ctrl + F2: Dividir pestaña verticalmente
 - Shift + F2: Dividir pestaña horizontalmente
 - Shift + F3 / Shift + F4: Cambiar de división

Nota: <archivo> o similares es solo una forma de expresar lo que debe ir. **No** utilice los signos < > al utilizar los comandos. Ej: \$ cat <archivo> -> \$ cat test.py

Shortcuts

- Consolacosas:
 - `$ ls`: Lista de los archivos y carpetas dentro del directorio actual.
 - `$ cd <directorio>`: "Change Directory" cambia al directorio ingresado. (Si se usan dos puntos `..` en el espacio de directorio, se regresará al directorio que contenía al actual. Ej: estamos en el directorio `/duckietown/` si ingreso `"$ cd .. "` me dirigirá al directorio principal del duckiebot ya que ahí se encuentra el directorio/carpeta `/duckietown/.`)
 - `$ pwd`: "Print working directory" Arroja el directorio en el que estamos trabajando actualmente.
 - `$ nano <archivo>`: Edita el archivo con un editor de texto nativo de la consola.
 - `$ chmod +x <archivo>`: Permite que el archivo sea ejecutable
 - `$ cp <archivo1> <archivo2>`: Copia el `<archivo1>` con el nombre del `<archivo2>`.
 - `$ cat <archivo>`: Muestra todo el texto que posee el archivo en la consola.
 - `$ cat <archivo1> > <archivo2>`: Copia el archivo con otro nombre (`archivo2`), el destino se puede reemplazar por un directorio/carpeta `<directorio>/<archivo2>`. PD: En este caso sí se debe utilizar el signo `>` entre `<archivo1>` y `<archivo2>`.

Primeros pasos

- Conectarse al duckiebot: `$ ssh duckiebot@duckiebot.local` (pass: quackquack)
- Normalmente, trabajaremos en la carpeta de duckietown, para acceder a ella se usa el comando "**cd**": `$ cd duckietown`
- Activar el **Hotspot** del duckiebot (Wi-Fi):
 - `$ source vehicle_name.sh`
 - `$./hotspot.sh on`
- Para apagar el duckiebot: `$ sudo shutdown now`
 - Advertencia: Procuren encontrarse en la consola del duckiebot antes de realizar el comando, no queremos que nuestro PC se apague perdiendo todos los archivos y progresos que no guardamos.

ROS

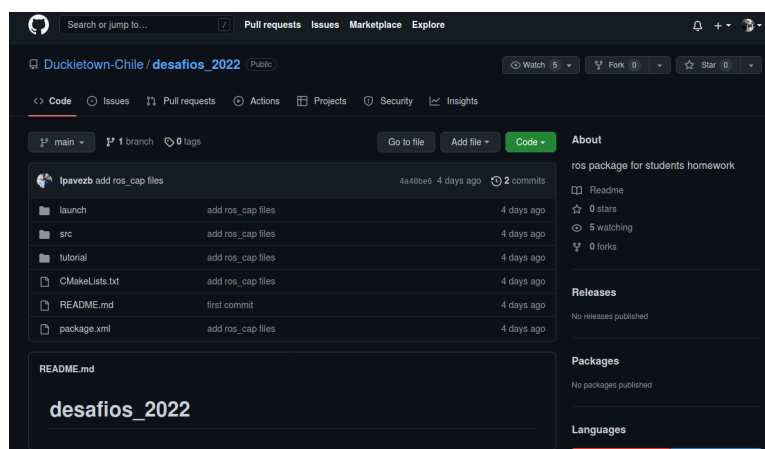
- \$ roslaunch ros_cap duckie_core.launch: Inicia el core de ROS para trabajar con el duckiebot (Importante!) PD: Recuerda que la pestaña quedará ocupada con el roscore por lo que tendrás que abrir otra pestaña.
- \$ rostopic echo <nombre del tópico> Sirve para ver constantemente las actualizaciones de un tópico, por ejemplo, sirve bastante para ver como varían los ejes.
- Para ejecutar scripts de python existen dos formas:
 - \$ rosrund <directorio> <archivo.py>
 - Esta es una forma muy directa de abrir un archivo ya que se puede realizar desde cualquier directorio y funcionará igual. PD: Por las primeras clases, se usará el directorio " ros_cap".
 - Obs: Para que funcione este método, deben hacer **ejecutable** el archivo que quieran usar con el comando "[chmod](#)".
 - \$ python <archivo.py>
 - A pesar de que se vea más simple que el anterior, este comando requiere que el archivo se encuentre en el directorio actual donde estás ejecutando el comando.
Ej: Quiero ejecutar test.py que se encuentra en ros_cap, si me encuentro en el directorio /duckietown/ y ejecuto el comando, éste dará error porque no encontró test.py en /duckietown/.

Misceláneo

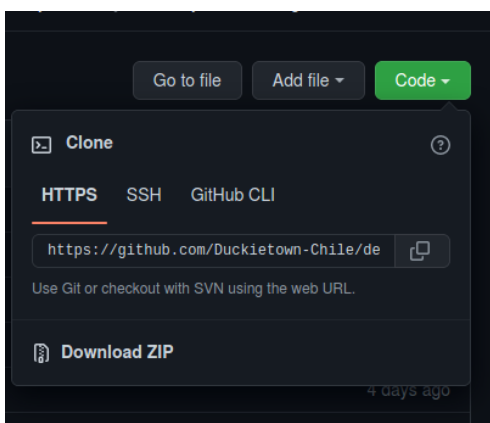
- AYUDA QUIERO VER LAS CARPETITAS DE MI DUCKIEBOT EN EL EXPLORADOR:
 - Abre el explorador de archivos de Ubuntu
 - Click en "Connect to Server" y escribe:
 - sftp://duckiebot@duckiebot.local/
 - Disfruta tus carpetitas :)

Git

- Fork: Hacer una copia de un repositorio externo en nuestro propio repositorio. Basta ir al repositorio de github que queremos *forkear* y, en la esquina superior derecha, se puede encontrar una pestaña que dice *fork*. Basta con hacer clic en ese botón, rellenar los campos requeridos y listo, ahora dentro de mis repositorios se encuentra el repositorio *forkeado*.



- Clone: Para poder tener un repositorio como una carpeta dentro de nuestro computador, se puede usar el comando ***\$git clone [link repositorio]*** en una consola de comandos en la ubicación donde se quiere generar la carpeta con todos los archivos del repositorio. Para encontrar el link, basta ir al repositorio en github, pinchar el botón que dice *code*, y en la sección HTTPS copiar el enlace que se muestra.



- pull: para *descargar* los cambios de github a nuestro repositorio clonado en el computador, basta con abrir una terminal en la dirección del repositorio en nuestro computador y usar el comando ***\$git pull***.
- add/commit/push: si hacemos algún cambio dentro del repositorio en nuestro computador, para poder actualizarlo en la nube de github, en primer lugar se debe usar el comando ***\$git add [nombre de los archivos]***. Si se quieren agregar todos los archivos disponibles, se puede usar ***\$git add -a***. Luego de hacer el *add*, se debe usar el comando ***\$git commit -m"[mensaje descriptivo]"*** para poder actualizar la rama de git con los cambios. Finalmente, para subir los cambios a la nube, se utilizará el comando ***\$git push***.
- status: Con el comando ***\$git status*** se desplegará información general del estado del repositorio.
- Token github: Para trabajar de manera remota con github, se pedirá el nombre de usuario y contraseñas de github, sin embargo, en caso de que se produzca algún error con la contraseña, será necesario crear un token de acceso personal a github. Para hacerlo, en github.com, con la sesión iniciada, pinchar la foto de perfil, ir a settings, a la izquierda buscar la opción de Developer Settings, ir a Personal Access Token, Generate New Token, configurarlo como se prefiera y listo, ahora cada vez que se pida la contraseña al trabajar desde un repositorio remoto, se debe ingresar el token. Es importante mencionar que NO se debe compartir el token, pues dará acceso COMPLETO a la cuenta de github. Además se debe anotar en algún lugar seguro pues, después de generado, no se podrá volver a ver cuál era el token.

Importing a ROS package to the Duckiebot

Para importar un nuevo paquete de ROS, de github, al duckiebot basta con clonarlo en la dirección `/catkin_ws/src/`. Para ello, conectar el duckiebot mediante el hotspot y conectar el duckiebot por cable de red a internet. Así, el duckiebot tendrá acceso a internet y podremos acceder a sus archivos de manera remota, una vez hecha esta conexión se procede a clonar el repositorio en la dirección correspondiente.

Una vez clonado, en la dirección `/catkin_ws/` se debe usar el comando ***\$catkin_make*** para re-compilar nuestro espacio de trabajo. Una vez compilado, se debe hacer un

source al archivo *setup.bash* ubicado en la carpeta *devel*. Para ello, estando en la carpeta *duckietown* hacer: **`$source /catkin_ws/devel/setup.bash`**.

Bugs

- Mi computador no se conecta al duckiebot (ssh duckiebot@duckiebot no funciona).
 - No se sabe por qué ocurre, pero pasa muy rara vez. Para solucionarlo solo se debe activar el hotspot en otro PC y conectarse por Wi-Fi al duckiebot.
 - Revisa que no estés conectado por cable LAN y Wi-Fi al mismo tiempo, sólo conectate al duckiebot por un medio.
- Ya hice mi script de python, se ejecuta bien pero el duckiebot no parece reaccionar ante éste :(
 - Probablemente te falte una letra o escribiste mal algún nodo (suele pasar).