



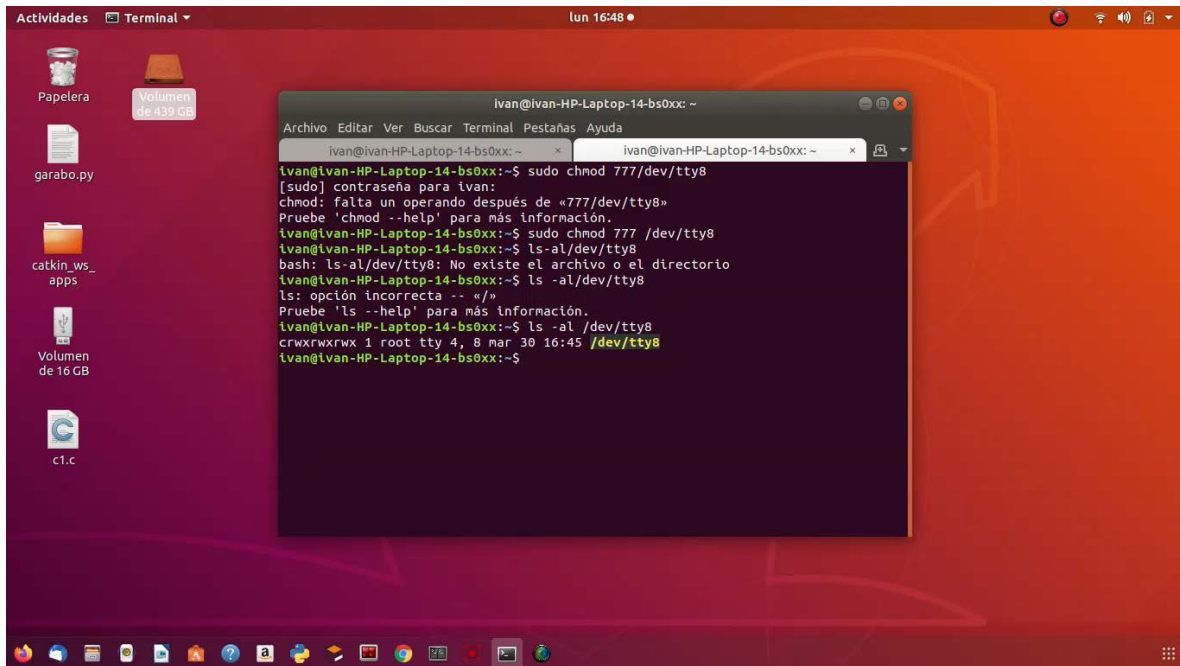
COMUNICACIÓN ROS PSOC

Dinámica de robots

Ingeniería mecatrónica

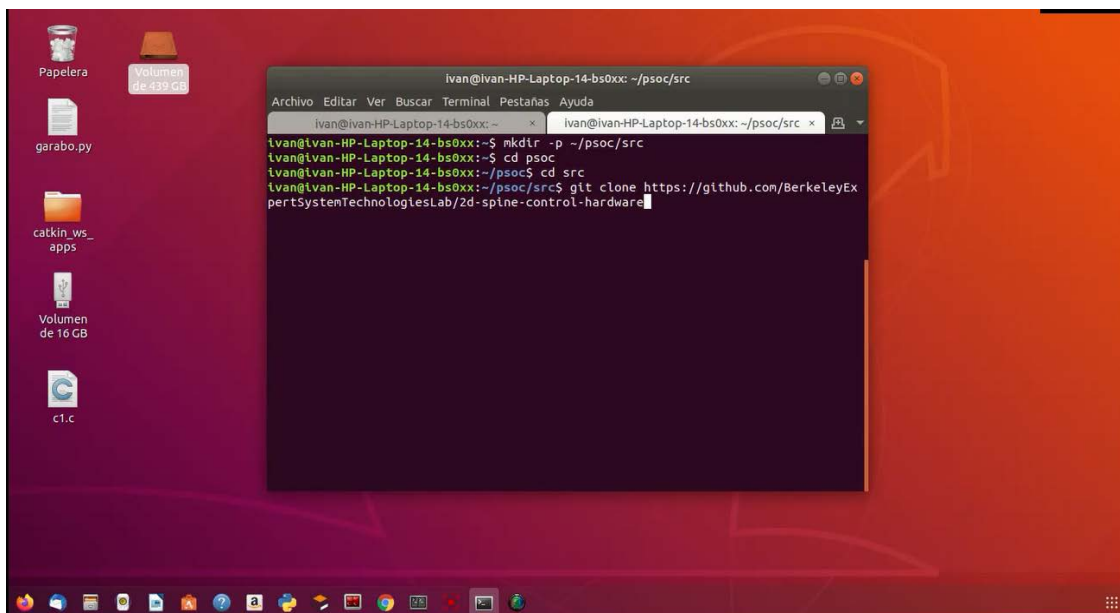
Ivan Alejandro Pasillas Gonzalez

Para esta practica se utilizó el sistema operativo Ubuntu con ros, y git, y a tarjeta psoc para hacer esta practica



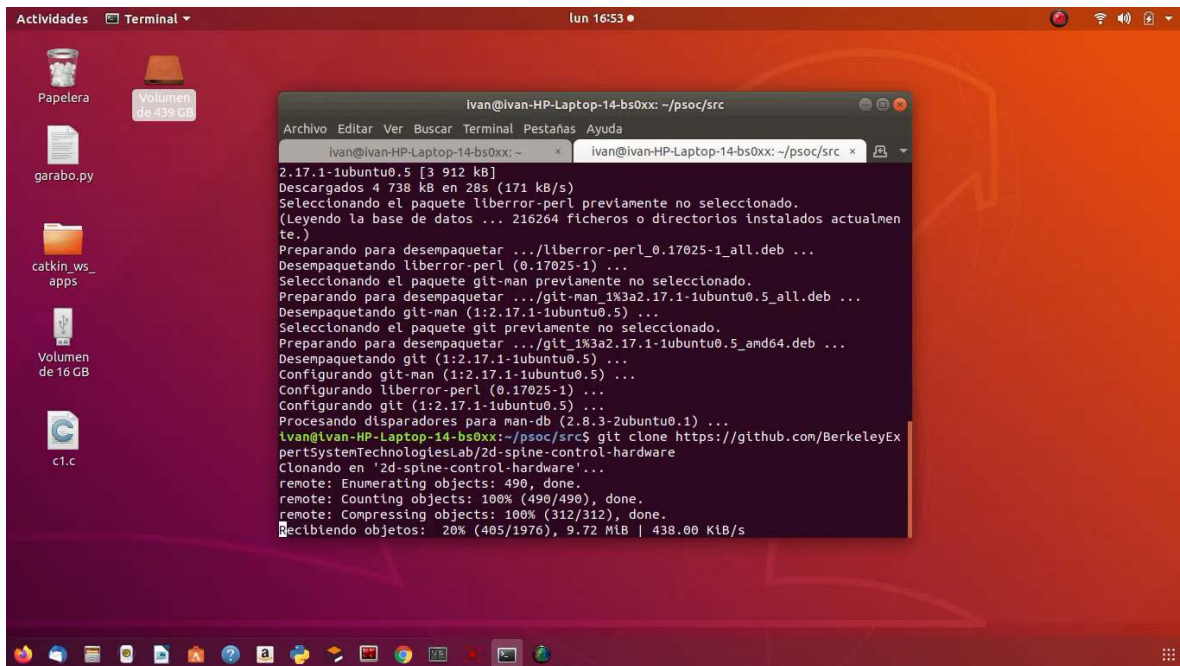
```
Ivan@ivan-HP-Laptop-14-bs0xx: ~  
Archivo Editar Ver Buscar Terminal Pestañas Ayuda  
ivan@ivan-HP-Laptop-14-bs0xx:~  
ivan@ivan-HP-Laptop-14-bs0xx:~$ sudo chmod 777/dev/tty8  
[sudo] contraseña para ivan:  
chmod: falta un operando después de «777/dev/tty8»  
Pruebe 'chmod --help' para más información.  
ivan@ivan-HP-Laptop-14-bs0xx:~$ sudo chmod 777 /dev/tty8  
ivan@ivan-HP-Laptop-14-bs0xx:~$ ls -al/dev/tty8  
bash: ls -al/dev/tty8: No existe el archivo o el directorio  
ivan@ivan-HP-Laptop-14-bs0xx:~$ ls -al/dev/tty8  
ls: opción incorrecta -- «/»  
Pruebe 'ls --help' para más información.  
ivan@ivan-HP-Laptop-14-bs0xx:~$ ls -al /dev/tty8  
crwxrwxrwx 1 root tty 4, 8 mar 30 16:45 /dev/tty8  
ivan@ivan-HP-Laptop-14-bs0xx:~$
```

Se ejecuta primero roscore para activar el ros y después el minicon que detecta en puerto com donde se encuentra la psoc conectada

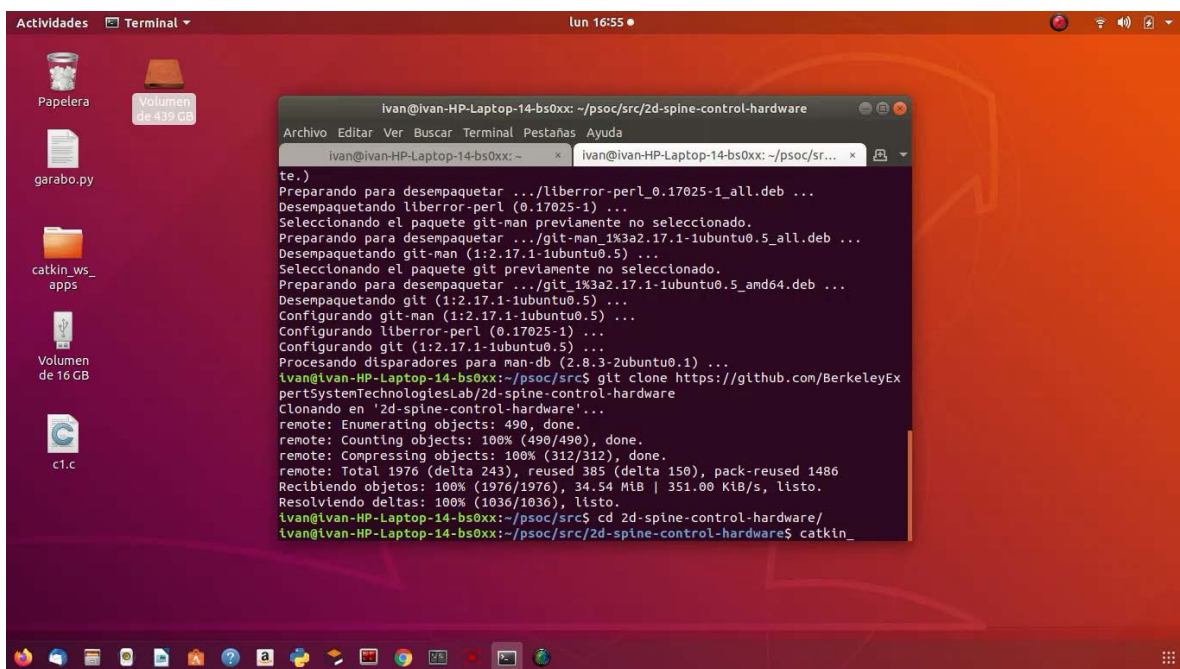


```
Ivan@ivan-HP-Laptop-14-bs0xx: ~/psoc/src  
Archivo Editar Ver Buscar Terminal Pestañas Ayuda  
ivan@ivan-HP-Laptop-14-bs0xx:~  
ivan@ivan-HP-Laptop-14-bs0xx:~$ mkdir -p ~/psoc/src  
ivan@ivan-HP-Laptop-14-bs0xx:~$ cd psoc  
ivan@ivan-HP-Laptop-14-bs0xx:~/psoc$ cd src  
ivan@ivan-HP-Laptop-14-bs0xx:~/psoc/src$ git clone https://github.com/BerkeleyEx  
pertSystemTechnologiesLab/2d-spine-control-hardware
```

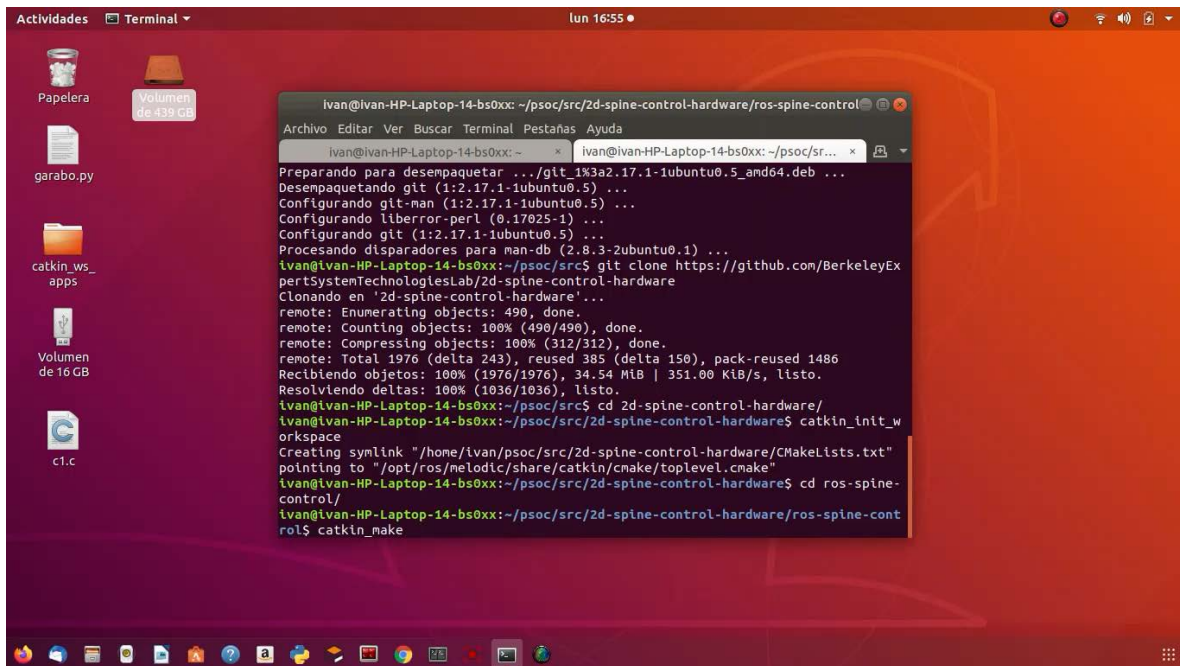
Se crea la carpeta con el comando mkdir una vez hecho accedemos con el comando cd una vez hecho descargamos el archivo git con el comando git clone y esperar que se descargue



En la imagen se aprecia como se descarga el git

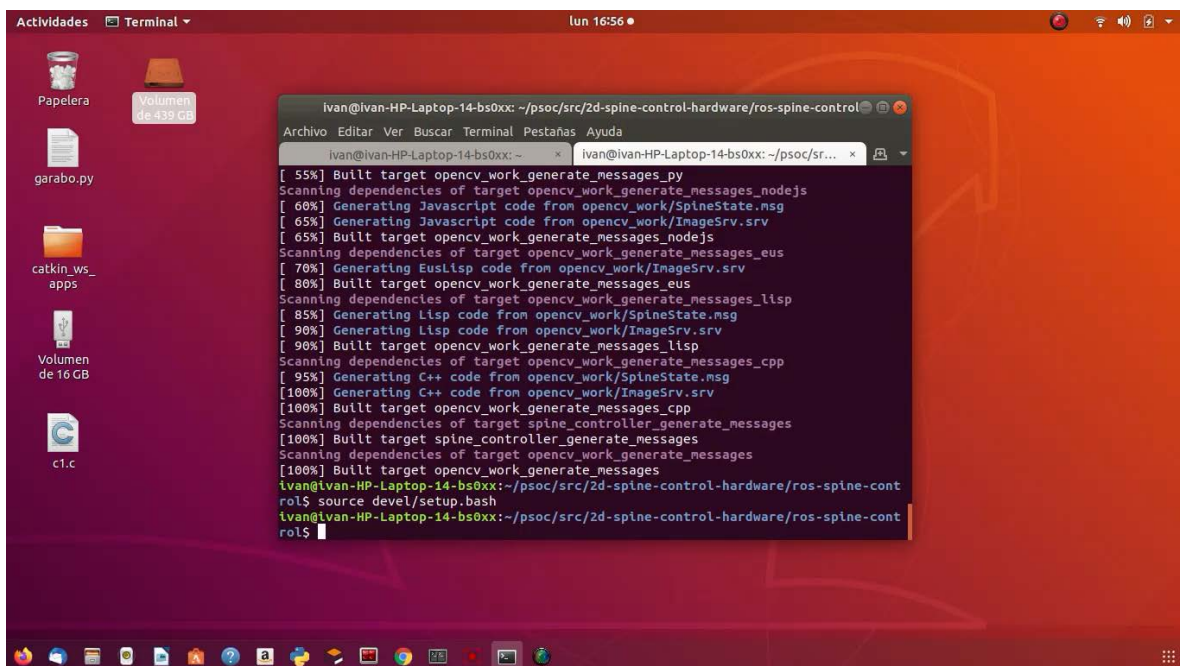


Una vez descargado entramos a la carpeta 2d spine e iniciamos nuestro workspace



```
ivan@ivan-HP-Laptop-14-bs0xx: ~/psoc/src/2d-spine-control-hardware/ros-spine-control
Archivo Editar Ver Buscar Terminal Pestañas Ayuda
ivan@ivan-HP-Laptop-14-bs0xx: ~
Preparando para desempaquetar .../git_1%3a2.17.1-1ubuntu0.5_amd64.deb ...
Desempaquetando git (1:2.17.1-1ubuntu0.5) ...
Configurando git-man (1:2.17.1-1ubuntu0.5) ...
Configurando liberror-perl (0.17025-1) ...
Configurando git (1:2.17.1-1ubuntu0.5) ...
Procesando disparadores para man-db (2.8.3-2ubuntu0.1) ...
ivan@ivan-HP-Laptop-14-bs0xx:~/psoc/src$ git clone https://github.com/BerkeleyEx
pertSystemTechnologiesLab/2d-spine-control-hardware
Clonando en '2d-spine-control-hardware'...
remote: Enumerating objects: 490, done.
remote: Counting objects: 100% (490/490), done.
remote: Compressing objects: 100% (312/312), done.
remote: Total 1976 (delta 243), reused 385 (delta 150), pack-reused 1486
Recibiendo objetos: 100% (1976/1976), 34.54 MiB | 351.00 KiB/s, listo.
Resolviendo deltas: 100% (1036/1036), listo.
ivan@ivan-HP-Laptop-14-bs0xx:~/psoc/src$ cd 2d-spine-control-hardware/
ivan@ivan-HP-Laptop-14-bs0xx:~/psoc/src/2d-spine-control-hardware$ catkin_init_w
orkspace
Creating symlink "/home/ivan/psoc/src/2d-spine-control-hardware/CMakeLists.txt"
pointing to "/opt/ros/melodic/share/catkin/cmake/toplevel.cmake"
ivan@ivan-HP-Laptop-14-bs0xx:~/psoc/src/2d-spine-control-hardware$ cd ros-spine-
control/
ivan@ivan-HP-Laptop-14-bs0xx:~/psoc/src/2d-spine-control-hardware/ros-spine-cont
rol$ catkin_make
```

Entramos a la carpeta ros-spine-control y compilamos el workspace



```
ivan@ivan-HP-Laptop-14-bs0xx: ~/psoc/src/2d-spine-control-hardware/ros-spine-control
Archivo Editar Ver Buscar Terminal Pestañas Ayuda
ivan@ivan-HP-Laptop-14-bs0xx: ~
[ 55%] Built target opencv_work_generate_messages_py
Scanning dependencies of target opencv_work_generate_messages_nodejs
[ 60%] Generating Javascript code from opencv_work/SpineState.msg
[ 65%] Generating Javascript code from opencv_work/ImageSrv.srv
[ 65%] Built target opencv_work_generate_messages_nodejs
Scanning dependencies of target opencv_work_generate_messages_eus
[ 70%] Generating EusLisp code from opencv_work/ImageSrv.srv
[ 80%] Built target opencv_work_generate_messages_eus
Scanning dependencies of target opencv_work_generate_messages_lisp
[ 85%] Generating Lisp code from opencv_work/SpineState.msg
[ 90%] Generating Lisp code from opencv_work/ImageSrv.srv
[ 90%] Built target opencv_work_generate_messages_lisp
Scanning dependencies of target opencv_work_generate_messages_cpp
[ 95%] Generating C++ code from opencv_work/SpineState.msg
[100%] Generating C++ code from opencv_work/ImageSrv.srv
[100%] Built target opencv_work_generate_messages_cpp
Scanning dependencies of target spine_controller_generate_messages
[100%] Built target spine_controller_generate_messages
Scanning dependencies of target opencv_work_generate_messages
[100%] Built target opencv_work_generate_messages
ivan@ivan-HP-Laptop-14-bs0xx:~/psoc/src/2d-spine-control-hardware/ros-spine-cont
rol$ source devel/setup.bash
ivan@ivan-HP-Laptop-14-bs0xx:~/psoc/src/2d-spine-control-hardware/ros-spine-cont
rol$
```

Luego establecemos las variables del ambiente que usa ros con el source devel/setup.bash

The screenshot shows a terminal window with the following content:

```
ivan@ivan-HP-Laptop-14-bs0xx: ~/psoc/src/2d-spine-control-hardware/ros-spine-control
roscore http://ivan-HP-Laptop-14-bs0xx:11311
ivan@ivan-HP-Laptop-14-bs0xx: ~/psoc$ rostopic list
/rosout
/rosout_agg
/serial_tx_cmdline
ivan@ivan-HP-Laptop-14-bs0xx: ~/psoc$ rostopic echo /serial_tx_cmdline
caca
data: "caca\n"
---
pasillas
data: "pasillas\n"
---
moy
data: "moy\n"
---

ivan@ivan-HP-Laptop-14-bs0xx: ~/psoc/src/2d-spine-control-hardware/ros-spine-control
rol$ source devel/setup.bash
ivan@ivan-HP-Laptop-14-bs0xx: ~/psoc/src/2d-spine-control-hardware/ros-spine-control
rol$ roslaunch spine_controller serial_tx_cmdline.py /dev/tty8
Running serial_tx_cmdline node with device: /dev/tty8
Unable to register with master node [http://localhost:11311]: master may not be
running yet. Will keep trying.
Opened port. Ctrl-C to stop.
Message to send over serial terminal: caca
Message to send over serial terminal: pasillas
Message to send over serial terminal: moy
Message to send over serial terminal: prueba de comunicaci
```

Por último, usamos el comando `roslaunch` para correr el programa y queda lista la comunicación en la ventana izquierda se introduce el mensaje y en la derecha aparece después de hacer entre en la ventana izquierda

Comandos utilizados

Activar Ros:

```
$roscore
```

Activar minicom:

```
$ sudo chmod 777 /dev/tty8
```

```
$ ls -al /dev/tty8
```

enviador de mensajes

```
$ mkdir -p ~/demo/src
```

```
$ cd ~/demo/src
```

```
$ cd ~/demo/src
```

```
$ git clone https://github.com/BerkeleyExpertSystemTechnologiesLab/2d-spine-control-hardware
```

```
$ cd 2d-spine-control-hardware/
```

```
$ catkin_init_workspace
```

```
$ cd 2d spine/ros-spine-control
```

```
$ catkin_make
```

```
$ source devel/setup.bash
```

```
$ rosrn spine_controller serial_tx_cmdline.py /dev/tty8
```

Receptor de mensajes

En otra ventana:

```
$ cd demo
```

```
$ rostopic list
```

```
$rostopic echo /serial_tx_cmdline
```