

Crazyflie Software setup:

Following the below paper :

http://act.usc.edu/publications/Hoenig_Springer_ROS2017.pdf

1. Bitcraze Crazyflie PC Client page 5
code to compile PC client for crazyflie :

- Make sure to install pyqt5

sudo apt-get install python3 python3-pip python3-pyqt5 python3-pyqt5.qtsvg

before running below command

```
$ cd ~/crazyflie/crazyflie-clients-python  
$ python3 bin/cfclient
```


Page 8 when install Crazyflie ROS stack, add required packages →
prior to add catkin_make make sure to follow these steps otherwise you will get Cmake errors:

```
$ cd ~/crazyflie_ws/src/crazyflie_ros  
$ git submodule init  
$ git submodule update  
$ cd ROOT
```

More info at :

<https://github.com/USC-ACTLab/crazyswarm/issues/65>

<https://github.com/USC-ACTLab/crazyswarm/blob/master/build.sh#L7-L10>

Now after this step the below code snippet can be handy in case you looking for URI (uniform resource-identifier) of your crazyradio. (make sure you have setup environment by “catkin_make” command) then do :

```
$ source ~/crazyflie_ws/devel/setup.bash  
$ rosrun crazyflie_tools scan
```

```
*****
```

To exit the source :

```
source /etc/init.d/foo &>/dev/null
```

```
echo $?
```

```
https://bash.cyberciti.biz/guide/Source_command
```

```
*****
```

You'll see sth like:

“radio://0/13/250k”

First number “0” can be used as channel index

Second number “13” is your channel number and can be between 0-125

Third one, “250k”, is channel speed per second and is one of these numbers (250k, 1M, 2M)

Next we have to install teleoperation dependency so we can use our joystick to control the CF. On page 9 where it says to install “hector_quadcopter_teleop” I couldn’t use the code snip given as my ROS version is kinetic. So I did it manually:

Install all the packages mentioned in hectorquadrotor.rosinstall and tutorials.rosinstall mentioned in the github repository

The way it’s mentioned in github library you want to add them manually like below

```
$ cd ~/crazyflie_ws/src/  
$ git clone -b kinetic-devel https://...(address).git  
$ git clone -b catkin https://...(address).git
```

Also add qt4 like :

```
$ sudo apt install qt4-default
```

Another needed package for hector to work this would help you avoid error about not finding geography-msgs packages missing :

```
$ sudo apt-get install ros-melodic-geographic-msgs
```

Manually installation for kinetic :

- **geographic-msgs**
- **uuid_msg**
- **Hardware_interface**

Then finally

```
$ cd crazyflie_ws/src
```

```
$ git clone -b kinetic-devel https://github.com/tu-darmstadt-ros-pkg/hector\_quadrotor.git
```

then you have to cd back to workspace and and catkin_make to finish setting up the package.

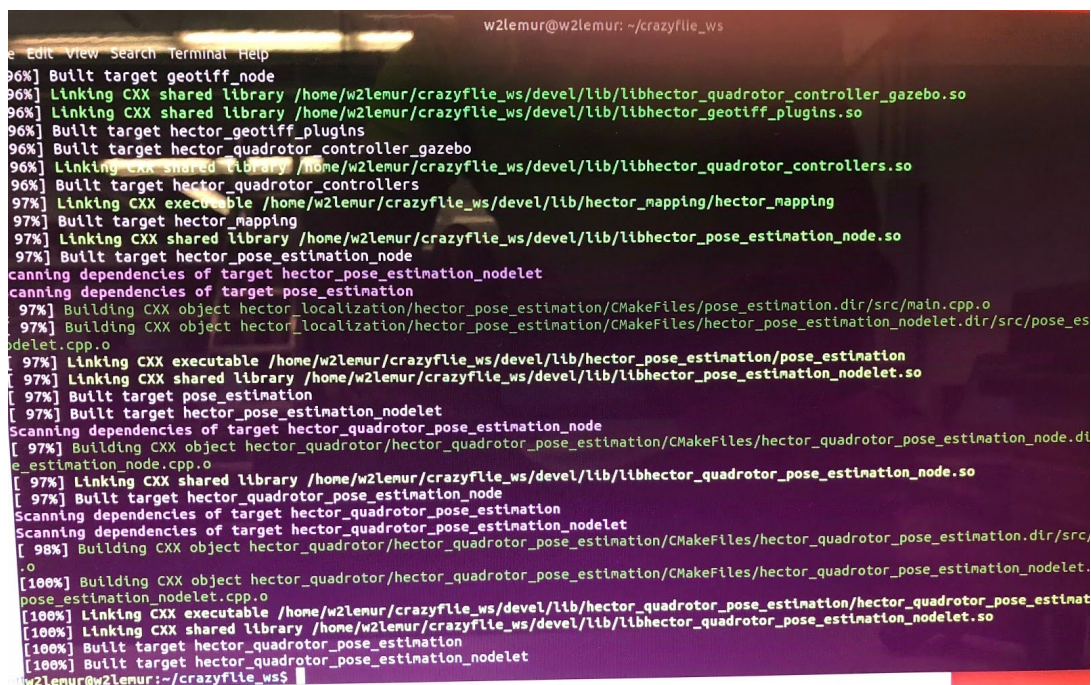
However there would be some packages common in crazyflie_ros and hector_quadcopter

so you can take the crazyflie_ros out of src catkin_make and then put it back in src directory.

```
$ cd ~/crazyflie_ws
```

```
$ catkin_make
```

**then put the crazyflie_ros back in ~/crazyflie_ws/src*



```
w2lemur@w2lemur: ~/crazyflie_ws
e Edit View Search Terminal Help
[ 96%] Built target geotiff_node
[ 96%] Linking CXX shared library /home/w2lemur/crazyflie_ws/devel/lib/libhector_quadrotor_controller_gazebo.so
[ 96%] Linking CXX shared library /home/w2lemur/crazyflie_ws/devel/lib/libhector_geotiff_plugins.so
[ 96%] Built target hector_geotiff_plugins
[ 96%] Built target hector_quadrotor_controller_gazebo
[ 96%] Linking CXX shared library /home/w2lemur/crazyflie_ws/devel/lib/libhector_quadrotor_controllers.so
[ 96%] Built target hector_quadrotor_controllers
[ 97%] Linking CXX executable /home/w2lemur/crazyflie_ws/devel/lib/hector_mapping/hector_mapping
[ 97%] Built target hector_mapping
[ 97%] Linking CXX shared library /home/w2lemur/crazyflie_ws/devel/lib/libhector_pose_estimation_node.so
[ 97%] Built target hector_pose_estimation_node
Scanning dependencies of target hector_pose_estimation_nodelet
Scanning dependencies of target pose_estimation
[ 97%] Building CXX object hector_localization/hector_pose_estimation/CMakeFiles/pose_estimation.dir/src/main.cpp.o
[ 97%] Building CXX object hector_localization/hector_pose_estimation/CMakeFiles/hector_pose_estimation_nodelet.dir/src/pose_es
nodelet.cpp.o
[ 97%] Linking CXX executable /home/w2lemur/crazyflie_ws/devel/lib/hector_pose_estimation/pose_estimation
[ 97%] Linking CXX shared library /home/w2lemur/crazyflie_ws/devel/lib/libhector_pose_estimation_nodelet.so
[ 97%] Built target pose_estimation
[ 97%] Built target hector_pose_estimation_nodelet
Scanning dependencies of target hector_quadrotor_pose_estimation_node
[ 97%] Building CXX object hector_quadrotor/hector_quadrotor_pose_estimation/CMakeFiles/hector_quadrotor_pose_estimation_node.d
e_estimation_node.cpp.o
[ 97%] Linking CXX shared library /home/w2lemur/crazyflie_ws/devel/lib/libhector_quadrotor_pose_estimation_node.so
[ 97%] Built target hector_quadrotor_pose_estimation_node
Scanning dependencies of target hector_quadrotor_pose_estimation_nodelet
Scanning dependencies of target hector_quadrotor_pose_estimation
[ 98%] Building CXX object hector_quadrotor/hector_quadrotor_pose_estimation/CMakeFiles/hector_quadrotor_pose_estimation.dir/src
.o
[100%] Building CXX object hector_quadrotor/hector_quadrotor_pose_estimation/CMakeFiles/hector_quadrotor_pose_estimation_nodelet.
pose_estimation_nodelet.cpp.o
[100%] Linking CXX executable /home/w2lemur/crazyflie_ws/devel/lib/hector_quadrotor_pose_estimation/hector_quadrotor_pose_estimat
[100%] Linking CXX shared library /home/w2lemur/crazyflie_ws/devel/lib/libhector_quadrotor_pose_estimation_nodelet.so
[100%] Built target hector_quadrotor_pose_estimation
[100%] Built target hector_quadrotor_pose_estimation_nodelet
w2lemur@w2lemur:~/crazyflie_ws$
```

// another method :

https://answers.ros.org/question/244776/is-it-possible-to-run-the-hector_quadrotor-demo-s-in-kinetic/

//

Once done with hector_quadcopter we add the controller. Launch files for both PS3 and Xbox360 controllers are provided so both can be used.

Install joystick package for ps3 controller:

<http://wiki.ros.org/joy>

Also install these ones:

- libusb in case you get error when doing catkin_make
 - sudo apt-get install libusb-dev
 - Libspnav-dev
 - Sudo apt-get install libspnav-dev
 - Bluetooth headers
 - Sudo apt-get install libblueooth-dev
 - Libcwiid
 - sudo apt-get install libcwiid-dev
- Personally feel like PS3 controller was a bit annoying since once plugged in it would mess with your mouse making it do some random movements.

For installing Xbox360 controller drivers :

<https://askubuntu.com/questions/165210/how-do-i-get-an-xbox-360-controller-working>

1. ***\$ sudo apt-get install --install-recommends jstest* joystick xboxdrv***
2. ***Echo "blacklist xpad" | sudo tee -a /etc/modprobe.d/blacklist.conf***
3. ***Sudo rmmod xpad #unload module if already loaded***
4. ***Xboxdrv --silent***
5. ***Then run : \$jstest-gtk to figure out what port is your controller***
6. ***To ensure the xboxdrv is loaded on startup***
 - a. ***Sudoedit /etc/init/xboxdrv.conf***
 - b.
 - c. ***And write following:***
 - Start on filesystem***
 - Exec xboxdrv -D***
 - Expect fork***

Command below can be used to control the crazyflie (use your radio uri and port number → jsx)

```
$ roslaunch crazyflie_demo teleop_ps3.launch uri:=radio://0/100/2M  
joy_dev:=/dev/input/js1
```

For installing VRPN-CLIENT Package please follow below tutorial:

<https://github.com/tuw-cpsg/tuw-cpsg.github.io/tree/master/tutorials/optitrack-and-ros>

- Mocap package is no longer supported by Motive (optitrack software). So don't waste time on that package, only option I found so far is VRPN-client.