Ros Basics

# Source:

always sources ROS and your catkin workspace:

* Source /opt/ros/noetic/setup.bash
* Source [catkin workspace]/devel/setup.bash

# Navigation:

* rospack find [package\_name ]
  + To look if a package if installed and give back the install directory
* roscd <package-or-stack>[/subdir]
  + go’s to the package or stack directory
* rosls <package-or-stack>[/subdir]
  + gives the list of the (sub) directory of the package or stack
* roscp <package-or-stack1> <package-or-stack2>
  + copy’s the file or dir form one package to an other

# Creating ros package:

1. make an catkin workspace
   1. go to directory where the workspace need to be
   2. us commando catkin\_make
   3. Remember to source the devel/setup.bash
2. catkin\_create\_pkg <package\_name> [depend1] [depend2] …
   1. create the catkin package with the given name
   2. type the depending packages behind it. Later you can add more packages by editing the CMakeList.txt and the package.xml file in the package directory
3. catkin\_make
   1. this builds the package and later is also builds the programs
   2. after you do catkin\_make don’t forget to source the devel/setup.bash file for updates

# make msg and srv files

the msg (message) files are inside of the msg directory and the srv files are inside of the srv directory. Msg files are used to send messages form one program to another program in the same package. Srv are service files those are used to make communication with a services.

The structure of a msg file is one element per line with first the data type and then the data name like:

string first\_name  
string Last\_name  
uint8 age  
uint32 score

The structure of a srv file is also one line per element with first data type then name but you also have return types those are under 3 dashes (---) line for example:

int64 a  
 int64 b  
 ---  
 int64 sum

After making this file you need to edit the package.xml file. We need to uncomment the

* <build\_depend>message\_generation</build\_depend>
* <exec\_depend>message\_runtime</exec\_depend>

And in the Cmake file we need to

* add message\_generation to the find\_package parameter
* uncomment the add\_message\_file function and include the msg file names
* uncomment the add\_service\_files funcion and include the srv file names
* and uncomment the generate\_message function

After this we need to build is so go to the catkin workspace base directory an execute “catkin\_make”

# Make Cpp program

* make an dir in your package named src
* put all your source files in this directory
* write your cpp program don’t forget to include the ros header file (“ros/ros.h”)
* edit the Cmake file by adding the command to build the executable
  + add\_executable(<name program> <directory main cpp file>)
  + target\_link\_libraries (<name program> ${catkin\_LIBRARIES})
  + add\_dependencies(<name program> (name package)\_generate\_messages\_cpp)
* build wit catkin\_make

# Run Cpp program

If the program does not need to send or receive messages just run it by typing   
rosrun <package name> <program name> if the sends or receives messages don’t forget to start the roscore by opening an extra terminal and use the “roscore” commando

# Playback data

You can record an play back data that is send trough the roscore “broker”. By simpely using the following command

* rosbag record -O <name output file> [topic to subscribe to] [extra topic] […]

you can get info about the file with “rosbag info <file name>” or simulate the data (play in again) with “rosbag play [-r x]<file name>” you can set the speed with -r and x give you the speed (-r 2 is play it as double speed).