

ROS CHEAT SHEET

Usage of the Cheat Sheet

| | |
|---------------------------------|--|
| command | Explanation of command, “+” marks options. |
| ⊢ <i>opt.1</i> | Use as written |
| ⊢ ⊢ <i>options</i> | If italics, replace with adequate value for <i>opt.1</i> |
| ⊢ [<i>other</i>] ⁺ | Use one or more <i>other</i> options |

File System

| | |
|--|--|
| catkin_make | Compile a project. |
| ⊢ clean | Remove build, devel & install spaces |
| ⊢ --pkg [<i>pkg</i>] ⁺ | Compile only <i>PKG</i> ... |
| catkin_create_pkg | Creates a new catkin package |
| ⊢ <i>pkg-name</i> | Init package of name <i>pkg-name</i> |
| ⊢ ⊢ <i>deps</i> | Init pkg with specified dependencies |
| source | Executes commands from a file |
| ⊢ devel/setup.bash | <i>Tells</i> the terminal about the compiled workspace. Run it in each terminal. |

Create a New Project

```
mkdir -p project_name/src
catkin_make
```

Common Tools

| | |
|----------------------------------|---|
| roscore | Starts a Master, the Parameter Server and a roscout logging node. A Master must be always up. |
| | |
| rosmmsg/rossrv | Get info of ROS messages/services |
| ⊢ show | Show message/services description |
| ⊢ list | List all messages/services |
| ⊢ package | List messages in a package |
| roslaunch | Used to execute ROS nodes |
| ⊢ <i>PKG EXEC</i> | Execute <i>EXEC</i> of package <i>PKG</i> |
| ⊢ ⊢ [<i>ARGS</i>] ⁺ | Execute with <i>ARGS</i> |
| rostopic | Prints info of ROS nodes |
| ⊢ ping node | Test connectivity of <i>node</i> |
| ⊢ list | Prints all ROS nodes available |
| ⊢ info node | Prints information of <i>node</i> |
| ⊢ machine node | Prints the machine <i>node</i> is running |
| ⊢ kill node | Kills the running <i>node</i> |
| roslaunch | Easily launch multiple nodes. If there is no Master up, it also starts one. |
| | |
| ⊢ <i>pkg file</i> | Launches <i>file</i> from <i>pkg</i> |
| ⊢ ⊢ <i>arg:=val</i> | Use <i>arg</i> with value <i>val</i> |

| | |
|------------------------|--|
| rostopic | Prints info of ROS topics |
| ⊢ echo /topic | Prints message from <i>/topic</i> to screen |
| ⊢ ⊢ -n <i>num</i> | Prints <i>num</i> messages |
| ⊢ hz | display publishing rate of topic |
| ⊢ info /topic | Prints info of <i>/topic</i> |
| ⊢ list | Prints all ROS topics available |
| ⊢ pub /X Y m | Publishes <i>m</i> to <i>/X</i> of type <i>Y</i> |
| ⊢ ⊢ ‘{ <i>field</i> }’ | Create message in-line |
| ⊢ type /topic | Prints the message type of <i>/topic</i> |

| | |
|-----------------------------------|-----------------------------------|
| rosservice | Prints info of ROS services |
| ⊢ args /srv | Print service arguments |
| ⊢ call /srv [<i>args</i>]* | call <i>/srv</i> with <i>args</i> |
| ⊢ info /srv | Print info of service <i>/srv</i> |
| ⊢ list | Print all ROS services available |
| ⊢ type /srv | Prints the type of <i>srv</i> |

| | |
|---------------------|------------------------------------|
| rosparam | Change params of the Param Server |
| ⊢ set par v | Set <i>par</i> with value <i>v</i> |
| ⊢ get par | Get the value of <i>par</i> |
| ⊢ delete par | Delete <i>par</i> |
| ⊢ list | Print all available parameters |

ROS Variables

| | |
|---|---|
| Linux environment variables specific of ROS, set for each terminal. | |
| ROS_ROOT | Path to where ROS is installed |
| ROS_MASTER_URI | Tells nodes where to locate the Master, by default: http://localhost:11311/ |
| | |
| ROS_HOSTNAME | Specifies the host name of the nodes. |
| | It takes precedence over ROS_IP |
| ROS_IP | Specifies the IP of the nodes |
| ROS_DISTRO | Specifies the current ROS distribution |

Linux Commands for Env. Vars.

| | |
|-----------------------|--|
| export var=val | Exports (<i>sets</i>) <i>var</i> to <i>val</i> |
| echo \$var | Prints the value of <i>var</i> |

Connect Multiple Machines

Define where the Master is going to run and get the ip (*master-ip*). In each terminal of all computers:

```
export ROS_HOSTNAME=computer_hostname
export ROS_MASTER_URI=http://master_ip:11311/
```

Useful Packages

| | |
|----------------------------|--|
| rosbag | Tool for using ROSBags |
| ⊢ info bag | Prints the metadata of a rosbag |
| ⊢ record [/topic]* | Records topic data to a bag file |
| ⊢ ⊢ -a | Record all topics |
| ⊢ ⊢ -O <i>name</i> | Save bag file as <i>name.bag</i> |
| ⊢ ⊢ -node= <i>N</i> | Save all topics from node <i>N</i> |
| ⊢ check bag | Determine if a bag is playable |
| ⊢ filter in out exp | Creates <i>out</i> by filtering <i>in</i> using <i>exp</i> |
| ⊢ fix bag | Attempts to fix a corrupted bag |
| ⊢ play bag | Plays <i>bag</i> |
| rqt | Launch the main widget. |
| rqt_bag | GUI for handling rosbags |
| rqt_graph | Visualize nodes, topics & services |
| rqt_image_view | Display images from topics |
| rqt_plot | Plot data from topics |
| rviz | 3D visualization tool for ROS |
| Gazebo | Robot simulator |
| rosterial_python | Connect serial devices, roslaunch required. |
| ⊢ serial.node p | Connect a device in port <i>p</i> with ROS. |

CMakeLists

| | |
|--|--|
| find_package(catkin | Find catkin macros and libraries |
| ⊢ REQUIRED COMPONENTS | find other catkin packages |
| ⊢ ⊢ roscpp | ROS library for c++ |
| ⊢ ⊢ rospy | ROS library for python |
| ⊢ ⊢ std_msgs | ROS standard messages |
| ⊢ ⊢ geometry_msgs | ROS geometry messages |
| ⊢) | |
| include_directories(| Other locations of headers |
| ⊢ include | |
| ⊢ \${catkin_INCLUDE_DIRS} | catkin directories |
| ⊢) | |
| add_executable(| Create an executable |
| ⊢ <i>exec_name</i> | of name <i>exec_name</i> |
| ⊢ <i>file.cpp</i> | from the source <i>file.cpp</i> |
| ⊢) | |
| target_link_libraries(| Tell cmake to link the executable |
| ⊢ <i>exec_name</i> | <i>exec_name</i> against the libraries |
| ⊢ [{ lib_name }] ⁺ | <i>lib_name</i> |
| ⊢) | |

Created by Gary, 2018.
Github Page