ROS COMMANDS OVERVIEW

/##/
Filesystem Commands:

Info:

Packages = software organization unit of ROS code. can contain libraries, executables, scripts, etc.

Manifests = (package.xml) description of a package, defines dependencies btw. packages and captures meta information (version, maintainer, license, etc) of package

Commands:

```
rospack find [package_name]          // to find packages/package locations
roscd <package-or-stack>[/subdir]    // to navigate in ROS-directories
pwd                                  // to show current path/location
rosls <package-or-stack>[/subdir]    // to show ROS-files
```

/##/
Create a Package:

Info: Content of each Package: package.xml, CMakeList.txt, Package folder

Procedure:

```
cd ~/catkin_ws/src
catkin_create_pkg <package_name> std_msgs rospy roscpp
cd ~/catkin_ws
catkin build
```

// Review first-order dependencies:
rospack depends1 <package_name>   //all dependencies will be listed, as stored in the related package.xml

// Review nested dependencies:
rospack depends <package_name>   //all depencencies, even from first- and lower-order dependencies will be listed

// To install missing dependencies:
sudo apt install TBD

rosdep install TBD

// Finally, build your package and re-source:
catkin build
// and close and re-open Terminal to re-source

/##/

```
/#############################################################################/
Classic ROS Commands:

Info: recommended link for more Infos/Troubleshooting:
[ http://wiki.ros.org/ROS/Tutorials/UnderstandingNodes ]
```

Commands:

```
roscore                          // starts the ROS-Master

..................................................................................
rosnode list                     // list all currently running nodes
       info /<node_name>         // shows Publications/Subscriptions , Services & Adress
       ping <node_name>          // to see if node is up & running
..................................................................................
rosrun [package_name] [node_name]  // starts a node (manually)


..................................................................................
rostopic -h                      // Help menu --> typing "-h" @ end opens help on most commands
       bw                        // display bandwidth used by topic
       echo /<topic_name>        // print/read messages to screen
       hz /<topic_name>          // display publishing rate of topic
       list                      // print information about active topics
       pub /<topic_name> [msg_type] [args] // publish/write data to topic
       type /<topic_name>        // print topic type = Messages sent in a topic, prints message_names

rosmsg show <message_name>       //  --> to get details about that message
..................................................................................
/#############################################################################/
```

Communication between nodes:

   1. Pub/Sub (covered under Classic ROS Commands/rostopics)
   2. Services

2. Services:

```
rosservice list                          // print information about active services

rosservice call rosservice type /[service]   // print service type
rosservice call /[service] [args]        // call the service with the provided args
rosservice find                          // find services by service type
rosservice uri                           // print service ROSRPC URI

rosparam list                            // lists all parameters applied to this node
rosparam set /[param_name] [args]        // change/set a parameter
rosparam get /[param_name]               // Note: >..get / <to show content of entire Parameter Server
rosparam dump [file_name] [namespace]    // Save Parameters to YAML File
rosparam load [file_name] [namespace]    // load Parameters e.g. into new namewspace
rosparam delete /[param_name]            // delete parameter


/#############################################################################/
```

/##
Project Launch:

Procedure:

roslaunch robothix_x1_base full.launch