USEFUL COMMANDS FOR UBUNTU AND BAXTER

open a terminal

```
ctrl-alt-t
```

---

get IP address

```
ifconfig
```

---

change directory to ros workspace and start baxter shell script.   Enter each line one at time

```
cd ros_ws
./baxter.sh   (or ./baxter.sh sim)   for the simulator
```

Any commands or queries that you do regarding Baxter need to be from a terminal where you have entered the above two commands. This is because in order to communicate with and command Baxter, you must establish connection between your workstation and Baxter. The baxter shell script sets up the ROS environment to point your PC to the ROS Master (Baxter) while at the same time registering your IP or Hostname allowing other processes to find your workstation.

Enable Baxter's motors, disable them, or get his state depending on if you use -e -d   or -s

```
rosrun baxter_tools enable_robot.py -e (enable motors)

                                    -d (disable motors)

                                    -s (gives the same info you would get with the command:

                                        rostopic echo /robot/state)
```

Baxter has over 200 topics.

If you want to list the topics use a terminal that is in Baxter's shell

```
cd ros_ws

./baxter.sh (or ./baxter.sh sim)

rostopic list
```

If you want to see what services Baxter has:

```
rosservice list
```

If you want to publish or subscribe to a topic and need the message type use the command rostopic info and then put the topic that you want the information on.

```
rostopic info /robot/xdisplay
```

To get the details of the message use rosmsg show then the message type:

```
rosmsg show sensor_msgs/Images
```

Common Baxter topics you may query

```
rostopic echo /robot/state
```

This tells you if Baxter's motors are enabled and the status of the estop button

```
rostopic echo /robot/joint_states
```

This give you the joints name, position, velocity, and effort.

If you only want to look at one part of the message such as the position, you can type

```
rostopic echo /robot/joint_states/position
```

In your baby_steps package you also have a small program joint_positions.py that subscribes to the /robot/joint_states topic and prints out the joint angles for you.

Additionally, the example program originally called "Golf.py" which uses canny edge , subscribes to /cameras/left_hand_camera/image

If you want to view the data, the command below will open a window with the image

```
rosrun image_view image_view image:=/cameras/left_hand_camera/image
```

To run a program:

```
rosrun "package name" "name of file"
```

To launch a "launch" file (these start multiple nodes at once)

```
roslaunch "package name"  "name of file"
```

Launching the simulator is a good example of using a launch file

```
roslaunch baxter_gazeo baxter_world.launch
```

Using rqt_console and rqt_logger ---- http://wiki.ros.org/ROS/Tutorials/UsingRqtconsoleRoslaunch

This allows you to monitor Baxter's internal messages being passed over topics.

```
rosrun rqt_console rqt_console
```

```
rosrun rqt_logger_level rqt_logger_level
```

rqt_console attaches to ROS's logging framework to display output from nodes. rqt_logger_level allows us to change the verbosity level (DEBUG, WARN, INFO, and ERROR) of nodes as they run.

If you need to set Baxter back to neutral, reset his face screen, make sure the grippers are open, and disable the motors, I included a package called neutral with a program in it called neutral.py. All you need to do is copy the folder to your /ros_ws/src directory

To run the program, you first have to make sure the .py file is executable. Do this by navigating in a terminal to the directory.

```
cd ros_ws/src/neutral/scripts
```

Once in the scripts directory:

```
chmod +x neutral.py
```

Close the terminal

rebuild your workspace by opening a terminal with the command ctrl-alt-t

```
cd ros_ws
```

```
catkin_make
```

```
source  ~/ros_ws/devel/setup.bash
```

close the terminal

open a new terminal   and navigate to either the baxter shell or the simulator shell

```
cd ros_ws
```

```
./baxter.sh sim   (or ./baxter.sh)
```

```
rosrun neutral neutral.py
```

Please let me know if you have any problems with the program and I will help you troubleshoot.
