Load ros (jessie)
*** initial steps
sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu jessie main" > /etc/apt/sources.list.d/ros-latest.list'
wget https://raw.githubusercontent.com/ros/rosdistro/master/ros.key -O - | sudo apt-key add -
sudo apt-get update
sudo apt-get install python-pip python-setuptools python-yaml python-distribute python-docutils python-dateutil python-six
sudo pip install rosdep rosinstall_generator wstool rosinstall
sudo rosdep init
rosdep update

***** need extra libs for urdf**

17 rosinstall_generator ros_comm urg_node freenect_stack --rosdistro indigo --deps --wet-only --exclude roslisp --tar > indigo-custom-wet.rosinstall
rosinstall_generator ros_comm kdl_parser diagnostic_updater urdf std_srvs dynamic_reconfigure geometry common_msgs --rosdistro indigo --deps --wet-only --exclude roslisp --tar > indigo-ros_comm-wet.rosinstall
Hokuyo_node, not urg_node
robot_state_publisher
Xacro
map_server
freenect_stack (requires sudo apt-get install libopencv-dev libopencv-core-dev and libfreenect0.2 libfreenect-demos libfreenect-dev)
18 wstool merge -t src indigo-custom-wet.rosinstall
19 wstool update -t src
20 rosdep install --from-paths src --ignore-src --rosdistro indigo -y -r --os=debian:jessie
21 ../install (runs install command under root; must set setup.zsh for root user)

mkdir ~/ros_catkin_ws/external_src
sudo apt-get install checkinstall cmake
sudo sh -c 'echo "deb-src http://mirrordirector.raspbian.org/raspbian/ testing main contrib non-free rpi" >> /etc/apt/sources.list'
sudo apt-get update

***libconsole-bridge-dev:
cd ~/ros_catkin_ws/external_src
sudo apt-get build-dep console-bridge
apt-get source -b console-bridge
sudo dpkg -i libconsole-bridge0.2*.deb libconsole-bridge-dev_*.deb

***liblz4-dev:
cd ~/ros_catkin_ws/external_src
apt-get source -b lz4
sudo dpkg -i liblz4-*.deb

cd ~/ros_catkin_ws/external_src
git clone https://github.com/ros/urdfdom_headers.git
cd urdfdom_headers
cmake .
sudo checkinstall make install

cd ~/ros_catkin_ws/external_src

sudo apt-get install libboost-test-dev libtinyxml-dev
git clone https://github.com/ros/urdfdom.git
cd urdfdom
cmake .
sudo checkinstall make install

cd ~/ros_catkin_ws/external_src
sudo apt-get install libboost-filesystem-dev libxml2-dev
wget http://downloads.sourceforge.net/project/collada-dom/Collada%20DOM/Collada%20DOM%202.4/collada-dom-2.4.0.tgz
tar -xzf collada-dom-2.4.0.tgz
cd collada-dom-2.4.0
cmake .
sudo checkinstall make install

cd ~/ros_catkin_ws
rosdep install --from-paths src --ignore-src --rosdistro indigo -y -r --os=debian:jessie

sudo mkswap /dev/sda
sudo swapon /dev/sda
swapon -s or top

Load ros (ubuntu)

158 sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu trusty main" > /etc/apt/sources.list.d/ros-latest.list'
159 cat /etc/apt/sources.list.d/ros-latest.list
160 sudo apt-key adv --keyserver hkp://pool.sks-keyservers.net --recv-key 0xB01FA116
161 sudo apt-get update
162 sudo apt-get install ros-indigo-desktop-full
173 sudo apt-get install nfs-kernel-server
174 sudo apt-get install nfs-common
sudo apt-get install xz-utils
530 sudo apt-get install ros-indigo-p2os-launch
sudo apt-get install ros-indigo-rviz
767 sudo apt-get install ros-indigo-p2os-msgs
768 sudo apt-get install ros-indigo-p2os-doc
844 sudo apt-get install liburdfdom-tools
852 sudo apt-get install xmllint
854 sudo apt-get install libxml2-utils
** add unreleased code to ubuntu apt-get ros installation

Add an unavailable package to PC ROS

- mkdir ros_catkin_ws
- cd ros_catkin_ws/
- mkdir src
- cd src
- catkin_init_workspace
- cd ..
- catkin_make
- git clone https://github.com/allenh1/p2os.git
- cd ros_catkin_ws/src/
- cp -r ~/p2os/p2os_urdf/ .
- cd ..

- sudo catkin_make install --source src
- cd build
- cd ..
- sudo su
- source /opt/ros/indigo/setup.bash
- sudo catkin_make install -DCMAKE_INSTALL_PREFIX=/opt/ros/indigo
- exit

Compile/test aria

unzip Aria-2.9.0
cd Aria
make (13 minutes)
cd ArNetworking/
make (~10 minutes)
cd ..
export JAVA_HOME=/usr/lib/jvm/jdk-8-oracle-arm-vfp-hflt
export JAVA_INCLUDE=/usr/lib/jvm/jdk-8-oracle-arm-vfp-hflt/include/
make lib/libArNetworkingJava.so
make lib/libAriaJava.so
sudo make install
sudo vi /etc/ld.so.conf
sudo ldconfig
ldconfig -p|grep Aria
cd ../Aria-2.9.0/examples/
./demo

Load/compile p2os (or other unreleased code)

- **Grab code from repository and place in src dir under catkin workspace**

  cd ~
  wget https://github.com/allenh1/p2os/archive/master.zip
  mkdir ros_catkin_ws
  cd ros_catkin_ws
  mkdir src
  cd src
  cp -rf ../../p2os-master/p* .
  cd ..

- must complete build of src

Test pioneer drivers

- rosrun p2os_driver p2os_driver _port:=/dev/ttyUSB0
- rostopic pub /cmd_vel geometry_msgs/Twist -r 1 -- '[0,0,0]' '[0,0,0.3]'

Hokuyo Urg04 ros drivers

(use hokuyo_node drivers, released and can be loaded via apt-get or rosinstall_generator

Run rviz remotely

On robot (Rpi)
```
export ROS_MASTER_URI=http://<RPI_IP>:11311
export ROS_IP=<RPI_IP>
```

Example with RPi address of 10.0.1.9
```
export ROS_MASTER_URI=http://10.0.1.9:11311
export ROS_IP=10.0.1.9
```

On rviz computer
```
export ROS_MASTER_URI=http://<RPI_IP>:11311
export ROS_IP=<PC_IP>
```

Example with RPI ip address of 10.0.1.9 and PC address of 10.0.1.11
```
export ROS_MASTER_URI=http://10.0.1.9:11311
export ROS_IP=10.0.1.11
```

Create python client (key_teleop)

Create c client

Publish message
rostopic pub /cmd_vel geometry_msgs/Twist -r 1 -- '[0,0,0]' '[0,0,0.3]'

Rosrun remap message:

- rosrun key_teleop key_teleopp.py /key_cmd:=/cmd_vel (format=> */existing_published_name:=/new_published_name*)

Important ros commands
Print current topics: *rostopic list*
Look at message contents - rostopic echo <topic>
Displays nodes currently running: *rosnode info*
prints current pub and subscribe messages for a node: rosnode info <node>
Print summary of nodes and messages - rosgraph

Rdist to clean rpi install

Env setup (compiling, remote run, local run)

Create launch files
```
<launch>
    <node pkg="p2os_driver" type="p2os_driver" name="p2os_driver">
        <param name="port" type="string" value="/dev/ttyUSB0" />
    </node>

    <node pkg="hokuyo_node" type="hokuyo_node" name="hokuyo_node">
        <!--
        <param name="serial_port" value="/dev/ttyACM0"/>
```

```
        <param name="angle_min" type="double" value="-2.09"/>
        <param name="angle_max" type="double" value="2.09"/>
        -->
    </node>

    <include file="$(find p2os_urdf)/launch/pioneer3dx_urdf.launch"/>

    <node pkg="tf" type="static_transform_publisher" name="base_link_to_laser" args="0 0 0.1397 0 0 0
base_link laser 100" />

    <node pkg="gmapping" type="slam_gmapping" name="slam_gmapping" args="/scan">
        <param name="delta" type="double" value="0.05" />
        <param name="temporalUpdate" type="double" value="2.5" />
        <param name="xmin" type="double" value="-2" />
        <param name="xmax" type="double" value="2" />
        <param name="ymin" type="double" value="-2" />
        <param name="ymax" type="double" value="2" />
    </node>

    <node pkg="map_server" type="map_server" name="map_server" />
</launch>
```

Read logs form nodes launched by roslaunch

Determine devices for robot/laser

Check disk space overall and in directories

(stage stage_map to capture data and gmapping, rosbag play and rviz)

Create map in real time

Create map offline using rosbag
Record messages: *rosbag record -O outfilename /base_scan /tf*

3 windows with ROS_IP and ROS_MASTER_URI set; rosparam set use_sim_time true

- *roscore*
- *rosrun gmapping slam_gmapping* (*must be before rosbag)
- *rosbag play <bagname>*

Once done, save map : *rosrun map_server map_saver -f <map_name>*

Watch map creation in rviz

- *Ctl-O* to open map cfg
- Map node printing map topic; othro mode then zero to center

Run nav stack/Move to 2d location in map
Move-base and config files
https://github.com/JenJenChung/pioneer_2dnav/archive/master.zip

Apt-get install ros-indigo-move-base ros-indigo-amcl teleop_key map_server

Run stage with pioneer

rosrun stage_ros stageros simple.world (assume simple.world is current dir)

Run gazebo with pioneer

Set/get parameters (underscore vs no underscore global)

- rosparam set _angle_min -2.09
- rosparam list
- rosparam set diagnostic_tolerance .1
- rosparam list