Our Process:

* sudo apt-get install ros-hydro-desktop-full ros-hydro-hokuyo-node
* mkdir –p ~/catkin\_ws/src
* cd ~/catkin\_ws/src/
* catkin\_init\_workspace
* cd ~/catkin\_ws/
* catkin\_make
* source devel/setup.bash
* cd ~/catkin\_ws/src
* catkin\_create\_pkg <package\_name\_all\_lowercase> <dependencies

>catkin\_create\_pkg navmap std\_msgs roscpp hokuyo\_node sensor\_msgs

* cd <package\_name\_all\_lowercase>

>cd navmap/

* gedit package.xml (Make all necessary changes, just copy-paste from this:

><?xml version = “1.0”?>

><package>

> <name>navmap</name>

> <version>0.0.0</version>

> <description>The navmap package for UAV Navigation & Mapping </description>

>

> <maintainer email=”alex.t.meyer@vanderbilt.edu”>NavMapLeader</maintainer>

>

> <license>BSD</license>

>

> <author email=”alex.t.meyer@vanderbilt.edu”>Alex Meyer</author>

> <author email=”rebecca.l.turok@vanderbilt.edu”>Rebecca Turok</author>

> <author email=”nolan.m.smith@vanderbilt.edu”>Nolan Smith</author>

> <author email=”john.a.francis@vanderbilt.edu”>John Francis</author>

> <author email=”franklin.s.snider@vanderbilt.edu”>Franklin Snider</author>

> <author email=”david.m.nakayama@vanderbilt.edu”>Michael Nakayama</author>

>

> <buildtool\_depend>catkin</buildtool\_depend>

>

> <build\_depend>hokuyo\_node</build\_depend>

> <build\_depend>roscpp</build\_depend>

> <build\_depend>sensor\_msgs</build\_depend>

> <build\_depend>std\_msgs</build\_depend>

>

> <run\_depend>hokuyo\_node</run\_depend>

> <run\_depend>roscpp</run\_depend>

> <run\_depend>sensor\_msgs</run\_depend>

> <run\_depend>std\_msgs</run\_depend>

>

></package>

* source /opt/ros/hydro/setup.bash
* cd ~/catkin\_ws/
* catkin\_make
* Launch file
* cd ~/catkin\_ws/
* roscd navmap/ (if doesn’t work >source devel/setup.bash
* mkdir launch
* cd launch/
* gedit navmap\_launcher.launch

><launch>

> <!--HOKUYO -->

> <node name=”hokuyo” pkg=”hokuyo\_node’ type=”hokuyo\_node”>

> <!--Set port to connect to here -->

> <param name=”port” type=”string” value=”/dev/ttyACM0”/>

> <param name=”intensity” type=”bool” value=”false”/>

> <param name=”min\_ang” type=”double” value=”-2.08621382713”/>

> <param name=”max\_ang” type=”double” value=”2.08621382713”/>

> </node>

></launch>

* roslaunch navmap navmap\_launcher.launch
* If errors try source /opt/ros/hydro/setup.bash

Catkin\_make

Rosrun navmap laser\_data

Rosrun hokuyo\_node hokuyo\_node

Roslaunch navmap navmap\_launcher.launch

Rosrun rviz rviz

Rostopic echo /scan

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Create ROS Workspace:

* mkdir –p ~/catkin\_ws/src
* cd ~/catkin\_ws/
* catkin\_make
* source devel/setup.bash

Create new Workspace:

* rosws init ~/fuerte\_workspace /opt/ros/fuerte

Create sandbox directory for new packages:

* mkdir ~/fuerte\_workspace/<sandbox\_name>
* rosws set ~/fuerte\_workspace/<sandbox\_name>
* source ~/fuerte\_workspace/setup.bash

Create package in sandbox directory:

* cd ~/fuerte\_workspace/<sandbox\_name>
* roscreate-pkg <pkg\_name\_no\_caps> <dependencies>
* <dependencies>: std\_msgs, rospy (Python library), roscpp (C++ library)
* rospack profile
* rospack find <pkg\_name\_no\_caps>

See first-order package dependencies in specific directory:

* rospack depends1 <pkg\_name\_no\_caps>

Install package dependencies:

* Use rosdep

Building Packages:

* rosmake <pkg\_name\_no\_caps>

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Creating catkin Package:

* cd ~/catkin\_ws/src
* catkin\_create\_pkg <pkg\_name\_no\_caps> <dependencies>

See first-order package dependencies in specific directory:

* rospack depends1 <pkg\_name\_no\_caps>

Customize Catkin Package:

* gedit ~/catkin\_ws/src/<pkg\_name\_no\_caps>/package.xml
* Then update the description tag to describe package, update the maintainer tag to inform people who to contact about the package, update the license tags (can just use BSD), Leave depend tags alone, remove any other unused tags and comments to clean it up

Building catkin package:

* source ~/catkin\_ws/devel/setup.bash
* cd ~/catkin\_ws/
* catkin\_make

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Creating Launch File – used to set values/params in ros and do other stuff:

* cd ~/catkin\_ws
* source devel/setup.bash
* roscd <pkg\_name\_no\_caps>
* mkdir launch
* cd launch
* gedit <launch\_name>.launch

Using roslaunch:

* roslaunch <pkg\_name\_no\_caps> <launch\_name>.launch

Edit files in ROS using rosed:

* rosed <package\_name> <filename>

Default editor is vim, change by editing ~/.bashrc to include:

* export EDITOR=’nano –w’ (for nano)
* export EDITOR=’emacs –nw’ (for emacs)

Creating ROS msg and src:

* <http://wiki.ros.org/ROS/Tutorials/CreatingMsgAndSrv>

Creating C++ Publisher and Subscriber:

* cd ~/catkin\_ws/src/<pkg\_name\_no\_caps>
* mkdir –p ~/catkin\_ws/src/<pkg\_name\_no\_caps>/src
* Create talker and listener depending on which is needed and edit code found on website to needs
* <http://wiki.ros.org/ROS/Tutorials/WritingPublisherSubscriber%28c%2B%2B%29>
* Add necessary lines to the CMakeLists.txt file
* Then do catkin\_make in the catkin workspace

Run Publisher/Subscriber:

* Start a roscore
* Cd ~/catkin\_ws
* source devel/setup.bash

rostopic echo /scan