

Baraja ROS Driver Guide

Installing the ROS Driver

This guide assumes you have a fully installed ROS system (follow the ROS install guidelines here: [ROS Kinetic \(Ubuntu 16.04\)](#) or [ROS Melodic \(Ubuntu 18.04\)](#)).

If you already have a catkin workspace, please skip to step 6.

Create a Catkin workspace:

1. Navigate to where you would like to store your workspace, e.g. ~/
2. In a Terminal, type:
`mkdir -p baraja_ws/src`
3. Move to your workspace directory, in a terminal type
`cd baraja_ws`
4. In the same Terminal
`catkin_make`
5. You should see your workspace build succeed.
6. Source the workspace files; add the following line to ~/.bashrc, modifying `<workspace folder>` to match the full path to your workspace:
`source <workspace folder>/devel/setup.bash`

Build the ROS driver:

7. Clone (or simply copy) the Baraja folder over into the src directory of your catkin workspace (you may have just created this).
8. Make sure you are in your workspace directory then run/re-run catkin_make in a terminal window.

Configuring the Baraja ROS Driver

The Baraja ROS driver can support several parameters depending on your system configuration. These parameters can be easily configured by creating a custom launch file for the driver. A single launch of the Baraja ROS driver can capture point clouds from any number of engines.

Example launch files can be found in your catkin workspace under `src/baraja/Launch/`, for instance, `two_engine_example.launch`, which will publish point clouds from two specified engines.

Do not modify `baraja_base.launch` or `baraja_driver.launch`.

It is recommended that you create a new launch file specific to your hardware setup. Make a copy of one of the example launch files, for instance, `two_engine_example.launch`, and rename it appropriately. For each engine you would like to receive point clouds from:

1. Modify the `engine_ip` default arg value to match the ip address of the engine you wish to receive from.

2. (optional) Modify the **lidar_name** and **child_link** args if necessary, these will give unique names to the published topic.
3. (optional) Modify the **s#_link** default arg values to specify the transformation frame of each sensorhead. This can be used to align multiple sensorheads spatially.

Running the Baraja ROS Driver

To run the Baraja driver:

1. Make sure that you source your setup files (this should be set up with your system when you did your install). Check that lines similar to this appear in your `~/.bashrc` file:

```
source /opt/ros/<kinetic/melodic>/setup.<bash/zsh>
source <workspace folder>/devel/setup.<bash/zsh>
```
2. Open a terminal and start up a roscore:

```
roscore
```
3. Open another (separate) terminal and launch the Baraja Driver by typing:

```
roslaunch baraja <your launch file>.launch, e.g:
roslaunch baraja two_engine_example.launch
```
4. Run your engine, making sure that the Point Cloud UDP Address is set to the ip address of the ROS system.
5. At this stage you should have data being published on the appropriate topics, check this by opening a terminal window and running

```
rostopic list
```
6. You should see the lists of topics, including the sensorheads that are publishing, `baraja_udp` topic and others. If you don't see this, make sure that you have your roscore running and that none of your nodes have crashed (check the terminals that you started up to see if there are errors etc).
7. (optional) Open rviz and add a sensorhead topic to the visualisation, you should see the point cloud displayed (you may need to modify the point size or global transformation frame).