

VELODYNE VLP16 DOCUMENTATION: RUNNING WITH ROS INDIGO.

ROS Distribution : INDIGO.

Operating System: Ubuntu 14.04.

Matheus Nascimento.

23/01/2017

Setting up your computer to communicate with the sensor

- * Connect the computer to the interface box with an Ethernet Cable;
- * Apply power to the sensor;
- * For now, disable the WiFi connection on your computer;
- * Configure your computer's IP address¹:
 - Access the Gnome Menu (Super key), type "Networks Connections" then run it;
 - Select the connection's name and click on "edit";
 - Choose the IPV4 Settings tab and change the "Method" field to "Manual" as the figure 1.
 - Click on "add" and set the IP address field to 192.168.1.77 ("77" can be any number except in a range between 1 and 254, except 201);
 - Set the "Netmask" to 255.255.255.0 and the "Gateway" to 0.0.0.0;
 - Check the "Require IPV4 addressing for this connection to complete" field and click on "save";
- * Open your web browser and access the following sensor's network address, 192.168.1.201;
- * If you can't access the page, disconnect and reconnect your Networking.
- * After use and enjoy the Velodyne, remember to reconfigure your network "IPv4 Settings". If you see a bug or issue after that, disconnect and connect your Ethernet Cable.

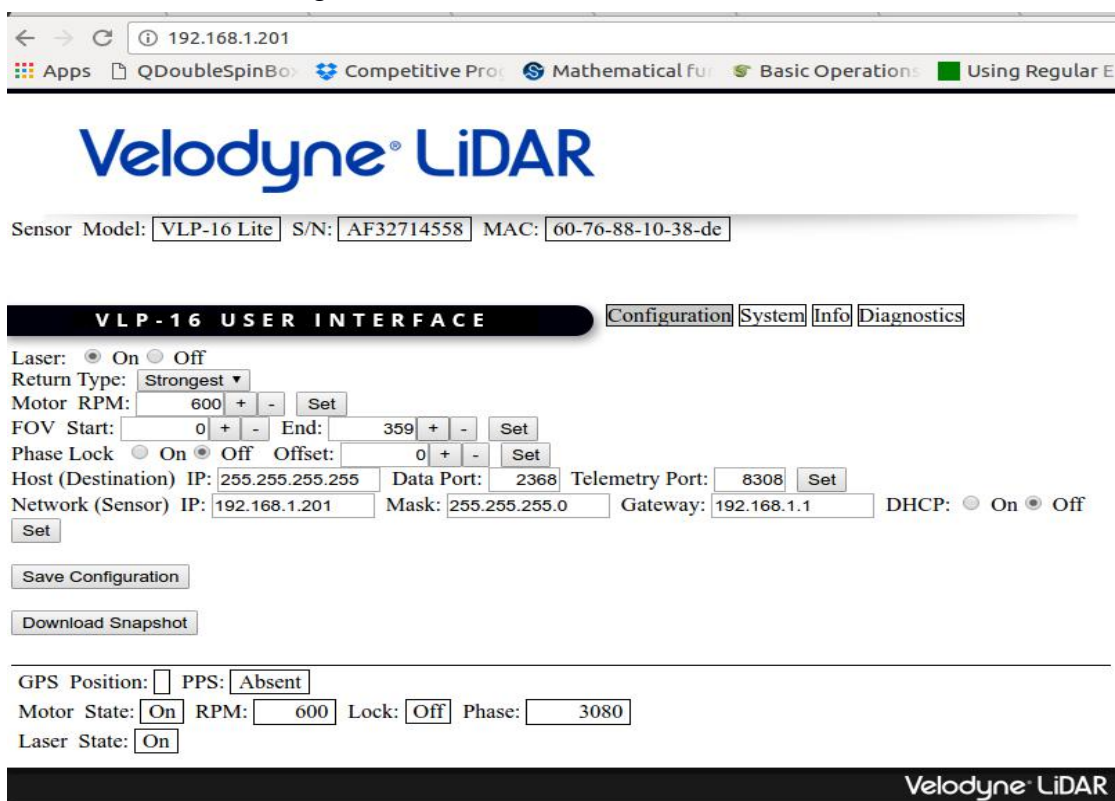
¹ that was the simpler way in Ubuntu 14.04 I've found to do that.

Figure 1 - Configuring the computer's IP address.



* Now, you can see the sensor's webserver interface and set up the wished features as the figure 2 shows.

Figure 2 - Sensor's webserver interface.



Install the VLP16 driver:

Install the Velodyne stack from the repositories by running:

```
"sudo apt-get install ros-indigo-velodyne"
```

On terminal, inside your ROS workspace, find the "src" folder and execute the following command:

```
"$ ~/catkin_ws/src/ git clone https://github.com/ros-drivers/velodyne.git"
```

If everything occurred fine, the following messages are going to show:

```
Cloning into 'velodyne'...
remote: Counting objects: 2983, done.
remote: Total 2983 (delta 0), reused 0 (delta 0), pack-reused 2983
Receiving objects: 100% (2983/2983), 581.33 KiB | 0 bytes/s, done.
Resolving deltas: 100% (1664/1664), done.
Checking connectivity... done.
```

After that, on terminal, inside your workspace, update all dependencies:

```
"$rosdep install --from-paths src --ignore-src --rosdistro indigo -y"
```

If everything occurred fine, the following message is going to show:

```
"#All required rosdeps installed successfully"
```

Then build your workspace:

```
"$ ~/catkin_ws/catkin_make"
```

```
...
```

```
[ 65%] Built target driver_nodelet
[ 70%] Built target velodyne_rawdata
[ 75%] Built target ringcolors_node
[ 80%] Built target ringcolors_nodelet
[ 85%] Built target transform_node
[ 90%] Built target transform_nodelet
[ 95%] Built target cloud_node
[100%] Built target cloud_nodelet
```

After that, your Velodyne packages are going to be ready to run.

Running with RViz

Run the following command on terminal:

```
"$ roslaunch velodyne_pointcloud VLP16_points.launch"
```

Now, the necessary nodes are running. You can check this with the following command:

```
"$ rostopic echo /velodyne_points"
```

You'll can see the messages being published and subscribed in the current topic.

After That, launch rviz, with the "velodyne" frame as the fixed frame.:

```
"$ rosrn rviz rviz -f velodyne"
```

In the "displays" panel, click on "Add" at the screen's lower left, then select the "Point Cloud2" field, then press "OK".

To see the 3D point cloud, in the "Topic" field of the new "Point Cloud2" tab, enter "/velodyne_points" as the figure 3 shows.

Congratulations. Now, your Velodyne is ready to builds the "real" world inside your system. Enjoy it.

Figure 3 - Set the /velodyne_points "Topic".

