dist_finder.py Source Code

function getRange(data, theta):

$$\theta_i = \theta_0 + i\theta_\Delta$$

where θ_i is the angle at scan point i
θ_0 is the angle_min variable in Ros LaserScan.msg
θ_Δ is the angle_increment variable in Ros LaserScan.msg

to find index of point at 50°:

$\theta_i \rightarrow 50^\circ$

$$50 = \theta_0 + i\theta_\Delta$$

$$\boxed{i = \frac{50 - \theta_0}{\theta_\Delta}}$$

Connecting LIDAR to windows with UrgBenri

go to network settings for usb-ethernet adapter.
goto TCP/IPV4
- set ip address: 192.168.1.15
  subnet mask: 255.255.255.0
  default gateway: 192.168.1.1

*Also disconnect from wifi

Connecting to LIDAR on Jetson

Make sure to Select Hokuyo Connection in networks.

run Command := rosrun urg_node urg_node _ip_address:= "192.168.1.11"

You can display the sensor data by running :=
rostopic echo /scan

Compiling Teensy firmware

In any folder start clean:

1. mkdir catkin_ws
2. cd catkin_ws
3. mkdir src
4. cd src
5. catkin_init_workspace
6. git clone https://github.com/ros-drivers/rosserial.git
7. cd rosserial and git checkout indigo-devel
8. In a completely seperate folder outside this Catkin project do
   git clone https://github.com/mischalkn/Fltenth.git
9. copy the folder race in Fltenth_master/arc_ws/src/ra to catkin_ws/src/rosserial
10. edit the file CMakeLists.txt in catkin_ws/src/rosserial/rosserial_arduino/CMakeLists.txt

in the field add_message_files add the files:

drive_param.msg
drive_values.msg
pid_input.msg

* also copy these message files from nischal's repo to your catkin_ws/src/rosserial/rosserial_arduino/msg

11. go to top level of catkin_ws and run
    catkin_make

12. run catkin_make install

13. Source install/setup.bash

14. Cd <Arduino/libraries>

15. rosrun rosserial_arduino make_libraries.py .

16. Done, now you can compile the teensy firmware.