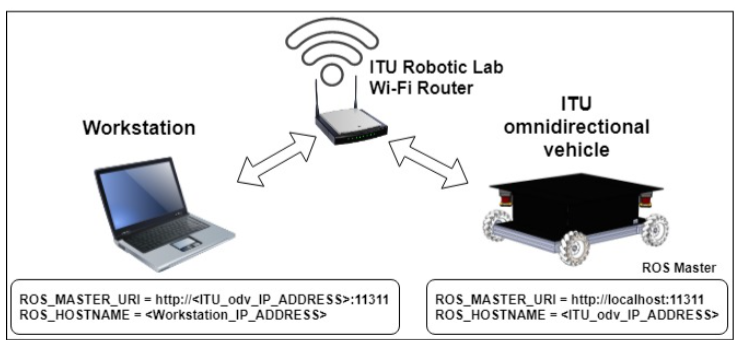
**ITU\_ODV USER MANUAL**

**Author: Ibrahim Dincer NALBANT**

**Vehicle Network Connection:**

* Connect to the Wi-Fi network that you are going to use (from vehicle and workstation both). Find both ITU\_odv and Workstation’s IP addresses.
* To create Secure Shell (SSH) protocol, one must know that ROS master should be running on the ITU\_odv main computer, while the host computer is the Workstation, which sends commands to the vehicle. Follow the below instructions and apply the changes in both of the computers’ .bashrc files as shown in the Figure below.



* Add the lines below to both computers bashrc files (this way, every terminal opened in the Workstation will try to connect to the ITU\_odv via SSH by default. To cancel it and remain in the Workstation, press CTRL+C):
* Workstation:

export ROS\_MASTER\_URI=http://<ITU\_odv\_IP>:11311

export ROS\_HOSTNAME=<Workstation\_IP>

ssh turtlebot@<ITU\_odv\_IP>

* ITU\_odv main computer:

export ROS\_MASTER\_URI=http://localhost:11311

export ROS\_HOSTNAME=<ITU\_odv\_IP>

NOTE\_1: Visit this website for more detailed information: [http://wiki.ros.org/turtlebot/Tutorials /indigo/ Network%20Configuration](http://wiki.ros.org/turtlebot/Tutorials%20/indigo/%20Network%20Configuration)

NOTE\_2: To prevent any synchronization errors (which causes extrapolation errors in the ROS TF package) between ITU\_odv and Workstation computers, install and enable Network Time Protocol (NTP) “ntpdate” to both of them and restart the systems.

**Vehicle Startup**

After successful connection via SSH protocol, the vehicle start process is shown below.

(NOTE: All terminals are started physically in the Workstation computer(ws))(NOTE: tb: turtlebot terminal, ws: workspace terminal)

---------------------**Terminal 0**--------------------------------tb

roscore

Optional--------**Connect Joystick**-----------------------------

Optional-----------**Terminal 1**-------------------------------ws // for PS3 joystick teleoperation

roslaunch itu\_odv\_startup teleop\_start.launch

-----**Connect Arduino to the main computer(USB)**-------

---------------------**Terminal 2**--------------------------------ws

roslaunch itu\_odv\_startup nav\_start.launch

---------------------**Terminal 3**---------------------------------tb

sudo chmod a+rw /dev/ttyACM0

rosrun rosserial\_python serial\_node.py /dev/ttyACM0

---------------------**Terminal 4**--------------------------------ws

roslaunch itu\_odv\_startup vis\_start.launch

-----------------**ITU\_odv switch: ON**--------------------------

------------------**LIDAR switch: ON**---------------------------

--------------------**Terminal 5**----------------------------------tb //create a script file for this

sudo chmod a+rw /dev/ttyACM1

sudo chmod a+rw /dev/ttyACM2

sudo chmod a+rw /dev/ttyUSB0

roslaunch itu\_odv\_startup sensors\_start.launch

---------------------**Terminal 6**--------------------------------ws

//IMPORTANT NOTE: Beware the height level of the obstacles, which cannot be detected by LIDAR sensors.

\*\*\*\*DO: *Press and release the emergency stop button to enable the motors*.

roslaunch itu\_odv\_navigation hector\_slam.launch

🡪 Now that you can drive the ITU\_odv autonomously via RVIZ.

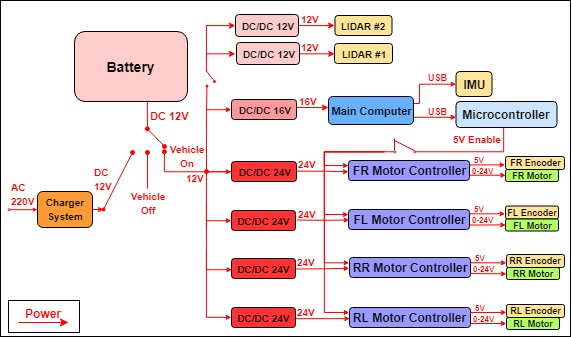
NOTE\_1: Arduino code, library files and ROS workspace folder can be found in this link:

<https://github.com/ITUROBLAB/-itu_odv_ws-idincern->

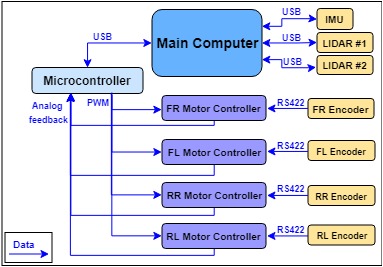
NOTE\_2: This vehicle is produced as a part of M.Sc. thesis in ISTANBUL TECHNICAL UNIVERSITY. For more detailed information about the vehicle, please refer to the thesis “DESIGN AND CONTROL OF ROS BASED OMNIDIRECTIONAL VEHICLE”

**APPENDICES:**

**A. Electrical Power Scheme of the Vehicle** :

****

**B. Wired Communication Scheme of the Vehicle Hardware:**

****