Operating the NEW amr documentation

Table of Contents

[1. IP address 1](#_Toc166052310)

[2. Connecting to raspberry pi 2](#_Toc166052311)

[3. Lidar Sensor 3](#_Toc166052312)

[3.1 Magni\_bringup base.launch 3](#_Toc166052313)

[3.2 Sick\_scan\_xd sick\_tim\_5xx.launch 4](#_Toc166052314)

[4. Mapping 4](#_Toc166052315)

[4.1 teleop\_twist\_keyboard 4](#_Toc166052316)

[4.2 gmapping 4](#_Toc166052317)

[4.3 Rviz 5](#_Toc166052318)

[5. Move base 5](#_Toc166052319)

[5.1 Amcl file 5](#_Toc166052320)

[5.2 Move base custom files 6](#_Toc166052321)

[5.3 Navigation stack files 8](#_Toc166052322)

[5.4 Roslaunch magni\_nav move base custom.launch 9](#_Toc166052323)

[5.5 Rviz 9](#_Toc166052324)

[6. References/tips 10](#_Toc166052325)

# IP address

Raspberry pi wan IP address: 10.42.0.1

Raspberry pi ethernet IP address:192.168.2.130

IPC Wan ip address:192.168.1.120(mMS4.0)/192.168.100.102(vchrome)

IPC ethernet ip address:192.168.2.103

Raspberry pi ethernet IP address for sensor:192.168.42.125

Lidar sensor IP address:192.168.42.100

Use the command “**nc -z -v -w5 192.168.42.100 2112**” in the raspberry pi terminal (change the IP address to the lidar sensor to test the connectivity if it is successful it will be like the image below)



# Connecting to raspberry pi

Using your laptop use the remote desktop connection and use the same wifi as the IPC and enter the ip address of the IPC connected to the wifi.

Use the command **“ssh** [**ubuntu@192.168.2.130**](mailto:ubuntu@192.168.2.130)**”** or the alias command **“bx3”** in the local terminal to connect to the raspberry pi of the AMR

A screenshot of a computer

Description automatically generatedthis will show that you have successfully ssh into the raspberry pi of the robot.

# Lidar Sensor

Go to the bashrc file in the raspberry pi with this command and add in this line “**source /home/ubuntu/catkin\_ws/devel/setup.bash**” in the raspberry pi terminal

A screenshot of a computer

Description automatically generated

It will always source the bashrc file before launch any file.

## Magni\_bringup base.launch

The magni\_bringup base.launch file is a must to be run first before anything else. This launch file sets the robot mode to teleoperation, launches the core setup script, and includes launch files responsible for joystick control and communication with ROS Bridge. It's intended to be launched on boot to bring up the necessary nodes for controlling the Magni robot. To launch the file use the command **“roslaunch magni\_bringup base.launch”** to launch the file.

## Sick\_scan\_xd sick\_tim\_5xx.launch

This launch file will launch/initiate the sensor allowing the other ros nodes to receive the lidar sensor input data. The command is “**roslaunch Sick\_scan\_xd sick\_tim\_5xx.launch**” this command is 2nd and it is to initiate the sensor.

These 2 launch file must always be launched when ssh into the raspberry pi.

# Mapping

## teleop\_twist\_keyboard

To do the mapping we will run the command “**rosrun teleop\_twist\_keyboard teleop\_twist\_keyboard.py**” in another terminal in the raspberry pi, to use the keyboard to move the amr to do mapping.

A computer screen shot of a computer

Description automatically generated

Try to see if the amr is able to move to prove it is working.

## gmapping

After that run the command “**rosrun gmapping slam\_gmapping scan:=scan**” in the raspberry pi terminal to begin mapping.

A computer screen with white text

Description automatically generated

The image above shows that the gmapping is working.

Run the teleop twist first before the gmapping.

## Rviz

Open another terminal and in the local terminal, run the command “**rviz**” or “**mkmap**”, it launches the rviz software to view the amr mapping.

# Move base

We created a custom move\_base launch file, same as the old amrs in the directory ~/catkin\_ws/src/ezmap\_pro/magni\_robot/magni\_nav/launch.

Also installed the amcl file for localization, after installation we modify the file slightly below.

## Amcl file

Amcl.launch file of the robot

Original in the files

<param name="kld\_err" value="0.01"/>

<param name="kld\_z" value="0.99"/><!-- kld\_err: 0.005, kld\_z: 0.9 -->

Modified to

<param name="kld\_err" value="0.005"/>

<param name="kld\_z" value="0.9"/><!-- kld\_err: 0.005, kld\_z: 0.9 -->

## Move base custom files

The code of the move\_base\_custom.launch file

<launch>

<!-- Run the map server -->

<arg name="map\_file" default="$(env HOME)/test1-ils.yaml" />

<node name="map\_server" pkg="map\_server" type="map\_server" args="$(arg map\_file)" />

<!--- Run AMCL -->

<include file="$(find magni\_nav)/launch/amcl.launch" />

<!--- Basic TF Transform -->

<node pkg="tf" type="static\_transform\_publisher" name="base\_footprint\_to\_base\_link"

args="0 0 0 0 0 0 base\_footprint base\_link 100" />

<node pkg="tf" type="static\_transform\_publisher" name="base\_link\_to\_cloud"

args="0 0 0 0 0 0 base\_link cloud 20" />

<!--- Move Base Package -->

<node pkg="move\_base" type="move\_base" respawn="false" name="move\_base" output="screen">

<rosparam file="$(find magni\_nav)/param/custom\_param/costmap\_common\_params.yaml" command="lo>

<rosparam file="$(find magni\_nav)/param/custom\_param/costmap\_common\_params.yaml" command="lo>

<rosparam file="$(find magni\_nav)/param/custom\_param/local\_costmap\_params.yaml" command="loa>

<rosparam file="$(find magni\_nav)/param/custom\_param/global\_costmap\_params.yaml" command="lo>

<rosparam file="$(find magni\_nav)/param/custom\_param/base\_local\_planner\_params.yaml" command>

<rosparam file="$(find magni\_nav)/param/custom\_param/costmap\_converter\_params.yaml" command=>

<param name="base\_local\_planner" value="teb\_local\_planner/TebLocalPlannerROS" />

<param name="controller\_frequency" value="10.0" />

</node>

</launch>

A screenshot of a computer screen

Description automatically generated

There are many things to change in the file such as the map file name and location,

The amcl file location, the move base package location and the tf transform.

## Navigation stack files

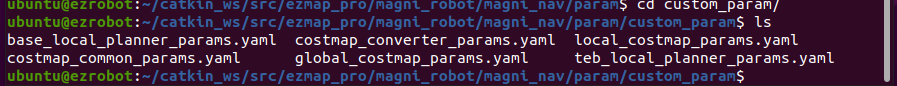
We also created a custom\_param inside param to contain all our navigation stack files

A screenshot of a computer program

Description automatically generated

Inside the custom\_param we added the navigation stack files of the old amr.

To add files use the command “**touch filename**” to add in these files.



## Roslaunch magni\_nav move base custom.launch

Use the command “**roslaunch magni\_nav move\_base\_custom.launch**” or “**mb2**” in the raspberry pi terminal, this will launch the amcl as well as all the navigation stack files and move base, all these files are needed to operate the amr seamlessly.

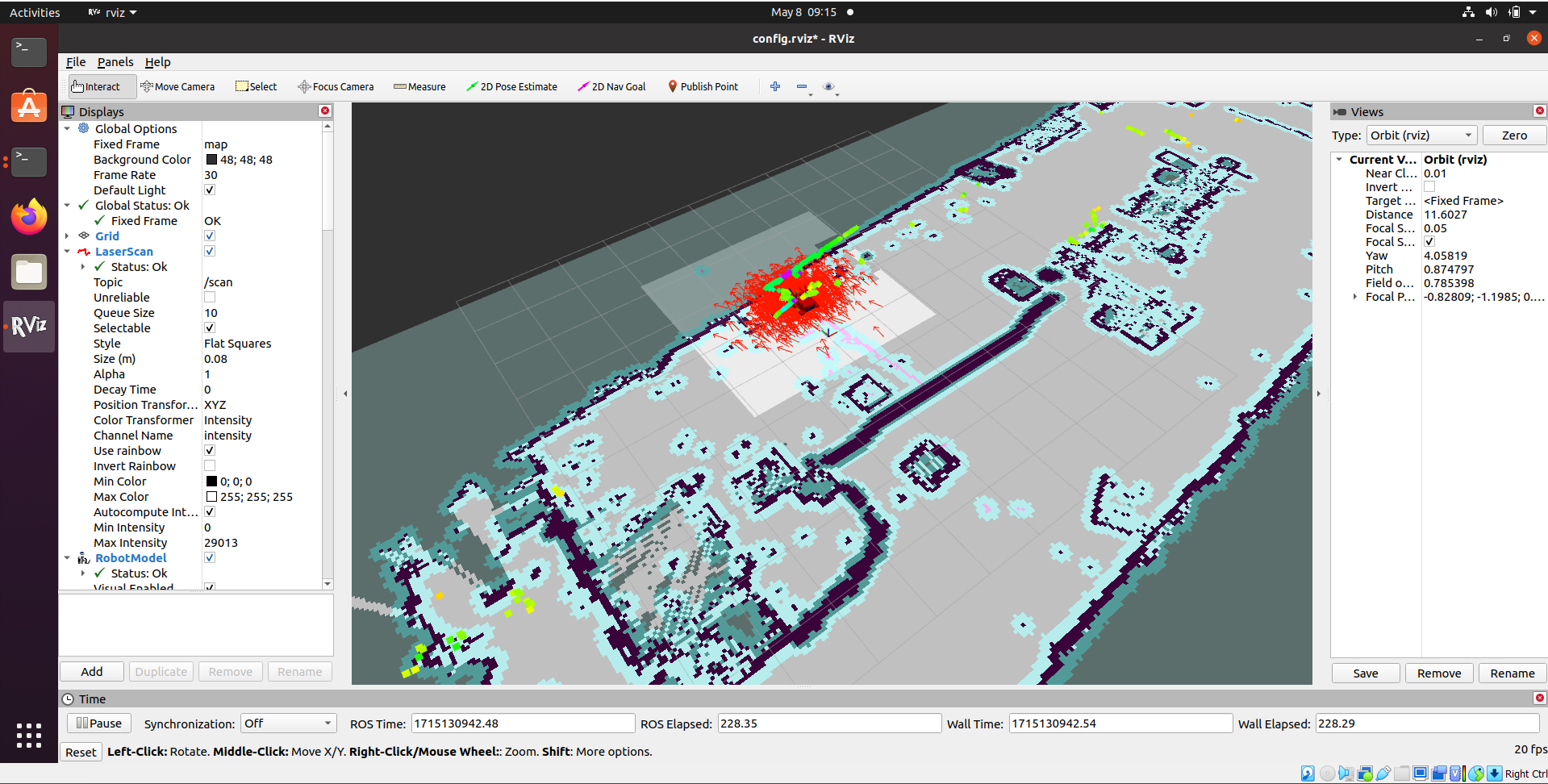
A screenshot of a computer

Description automatically generated

After running move base custom, it should looks something like this.

## Rviz

Also open rviz in the local terminal using the alias command “**rz**” to see the amr and the map.



# References/tips

Should any command above have issue, it okay to stop the programme and run it again as the raspberry pi isn’t able to process all the information smoothly so it takes a few try.

Must run the magni\_bringup first before other launch files.

<https://youtu.be/NW97xLF7CYQ?si=PmyXO-ArB0s8wr_V> (setting static ip of the raspberry pi robot)

<https://github.com/SICKAG/sick_scan> (ip address of lidar connectivity)

<https://wiki.ros.org/navigation> (installing navigation)