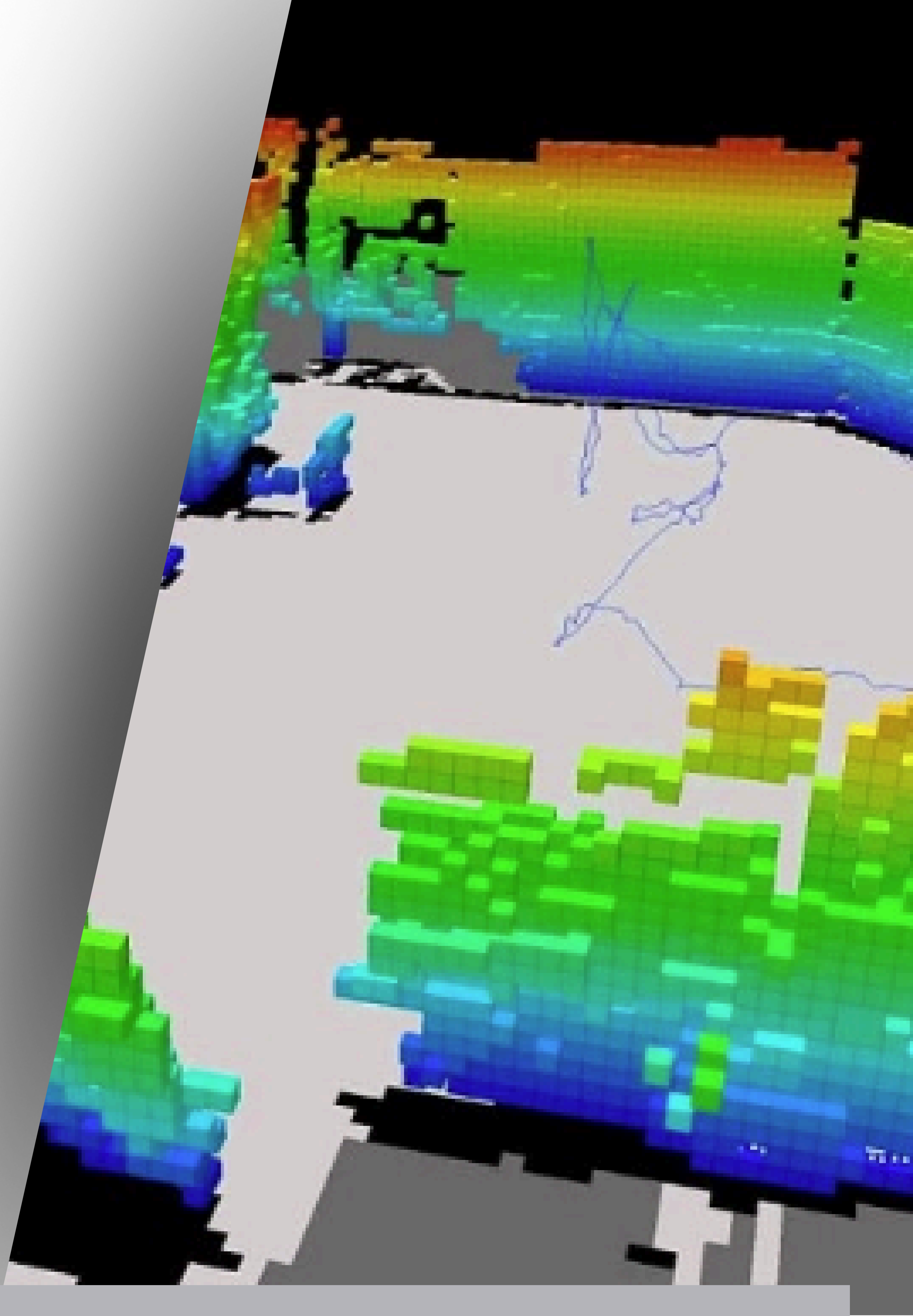
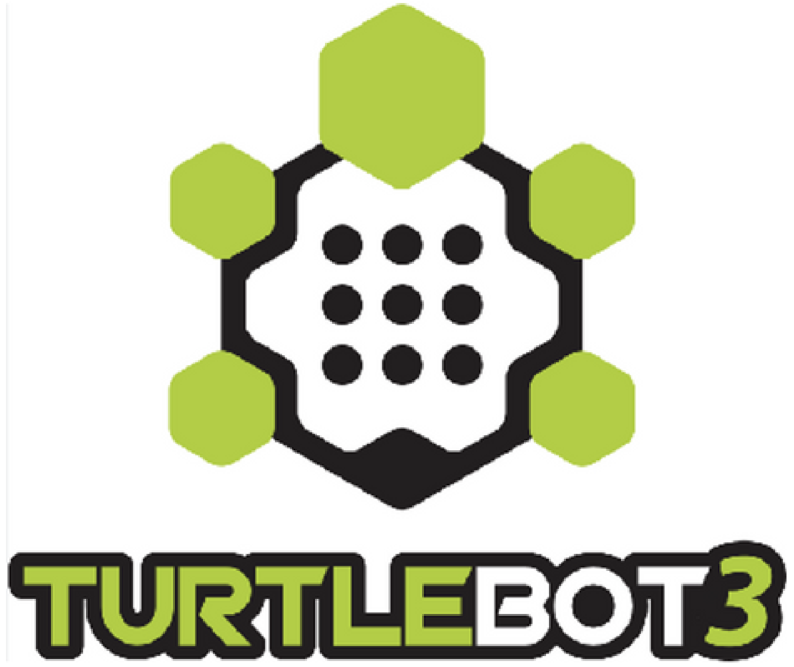
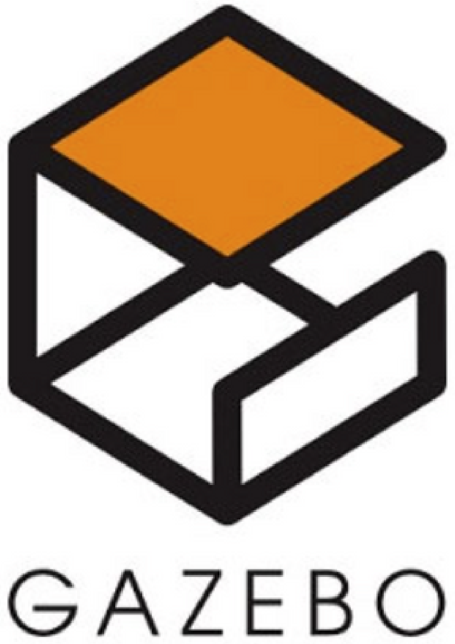
DEGGENDORF INSTITUTE OF TECHNOLOGY

Turtlebot 3 Connection



What are we going to use?





In Simulation

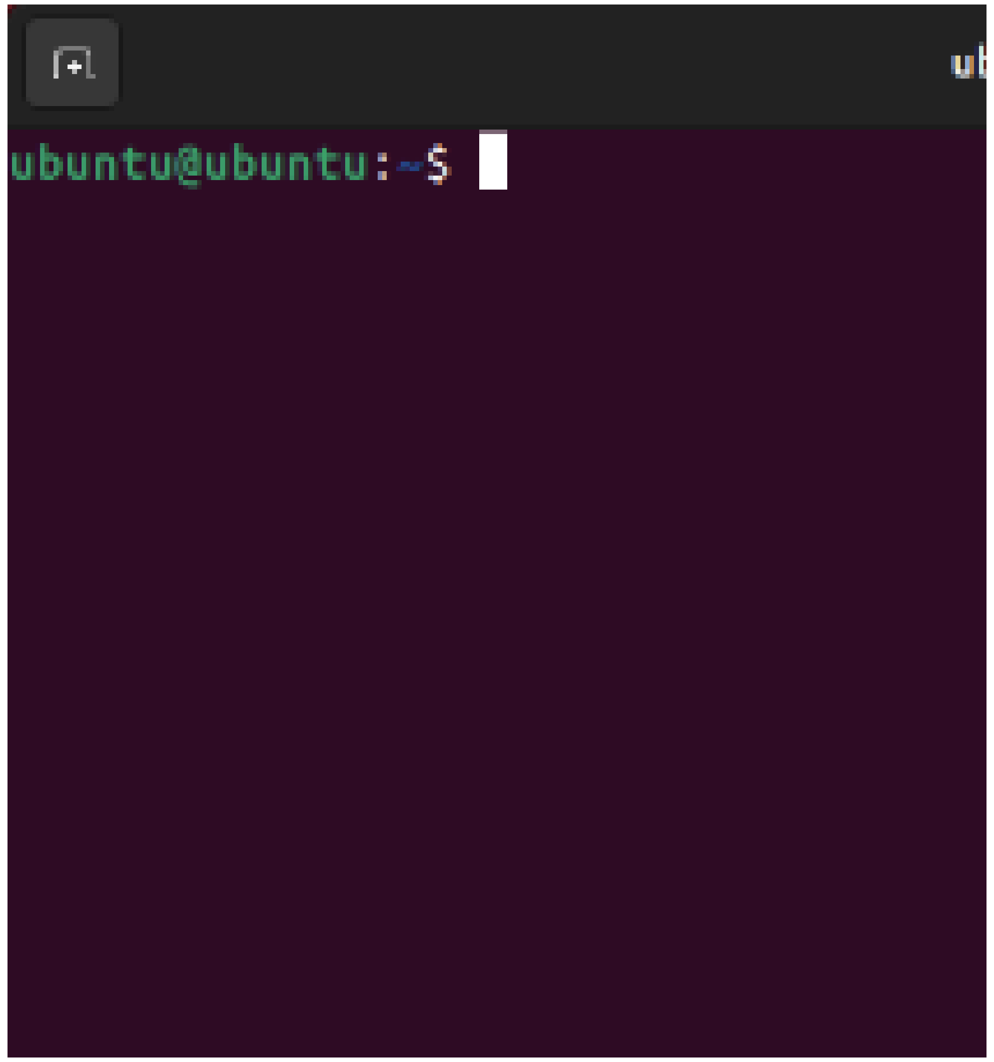
Run Docker Container

Access a new folder **cd** <Folder name>

Check the files within a folder **ls**

Command window comands

1	Access Docker folder	cd ros2_turtlebot3/
2	to access the docker environment	**home/tcc/ros2_turtlebot3$** bash run_docker.sh
3	to access a new tab in the docker environment	**home/tcc/ros2_turtlebot3$** bash into_docker.sh



```
ubuntu@ubuntu:~$
```

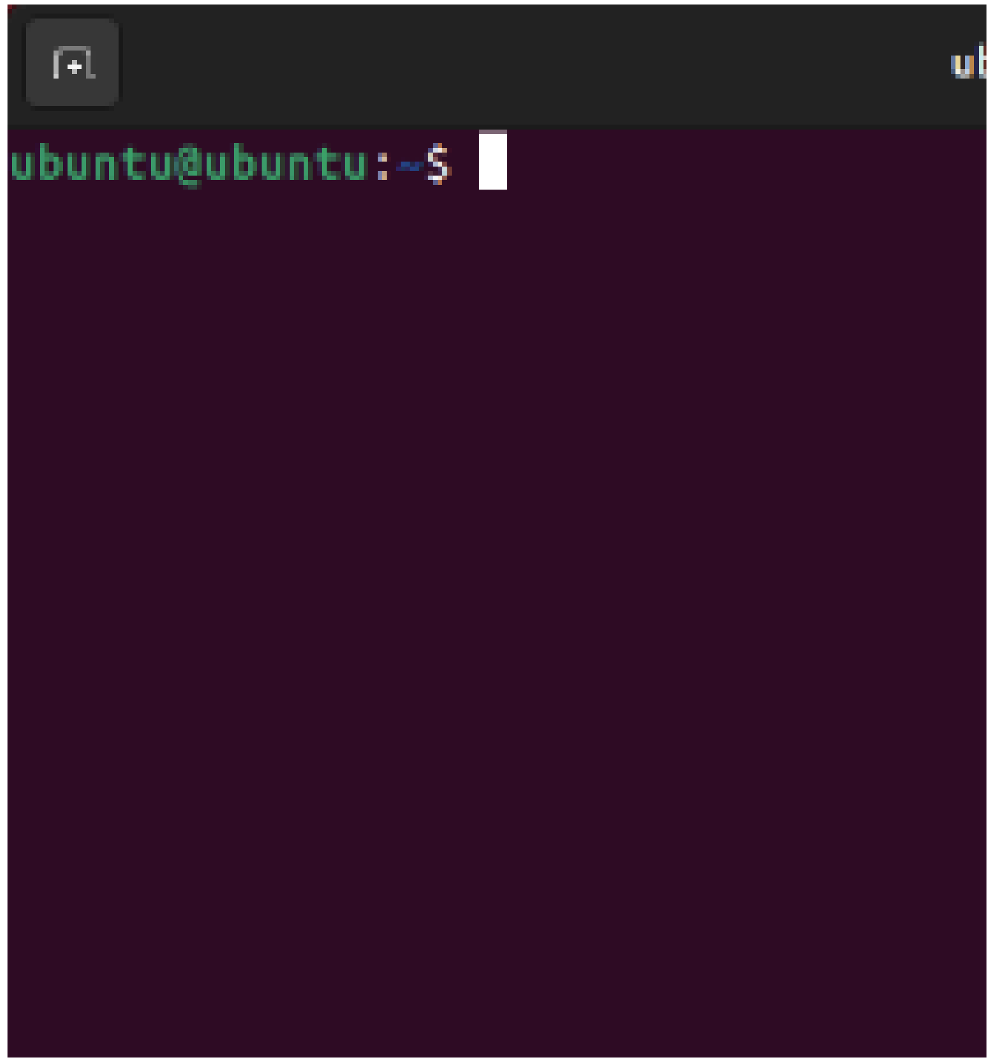
Run turtlebot3 Gazebo simulation

Access a new folder **cd** <Folder name>

Check the files within a folder **ls**

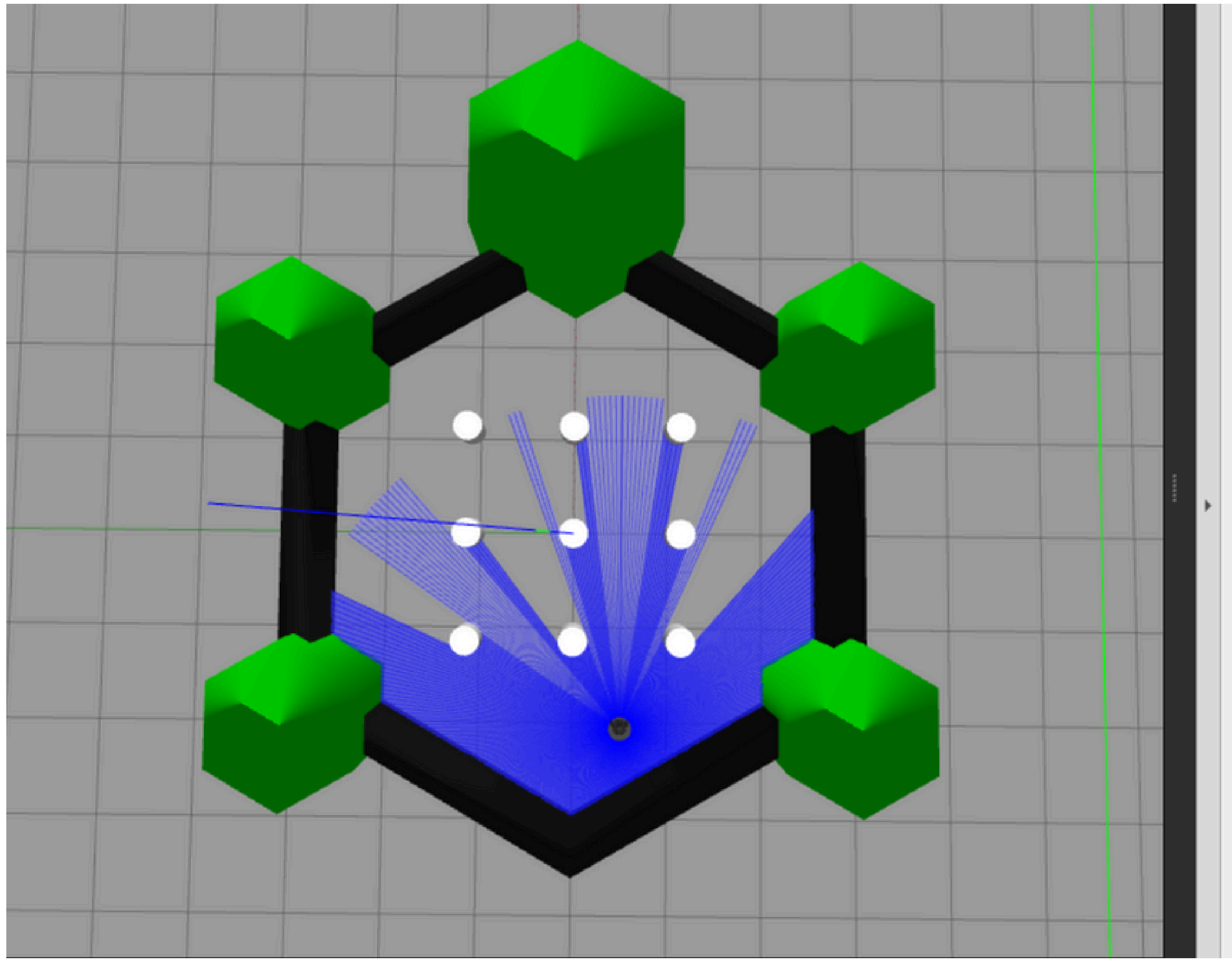
Each commands should be run in a new terminal or new tab of the current terminal

1	to launch Simulation World (inside Docker)	**/ws_slam#** ros2 launch turtlebot3_gazebo turtlebot3_world.launch.py
2	to launch SLAM Node (inside Docker)	**/ws_slam#** ros2 launch turtlebot3_cartographer cartographer.launch.py use_sim_time:=True
3	to launch a control Node (inside Docker)	**/ws_slam#** ros2 run turtlebot3_teleop teleop_keyboard

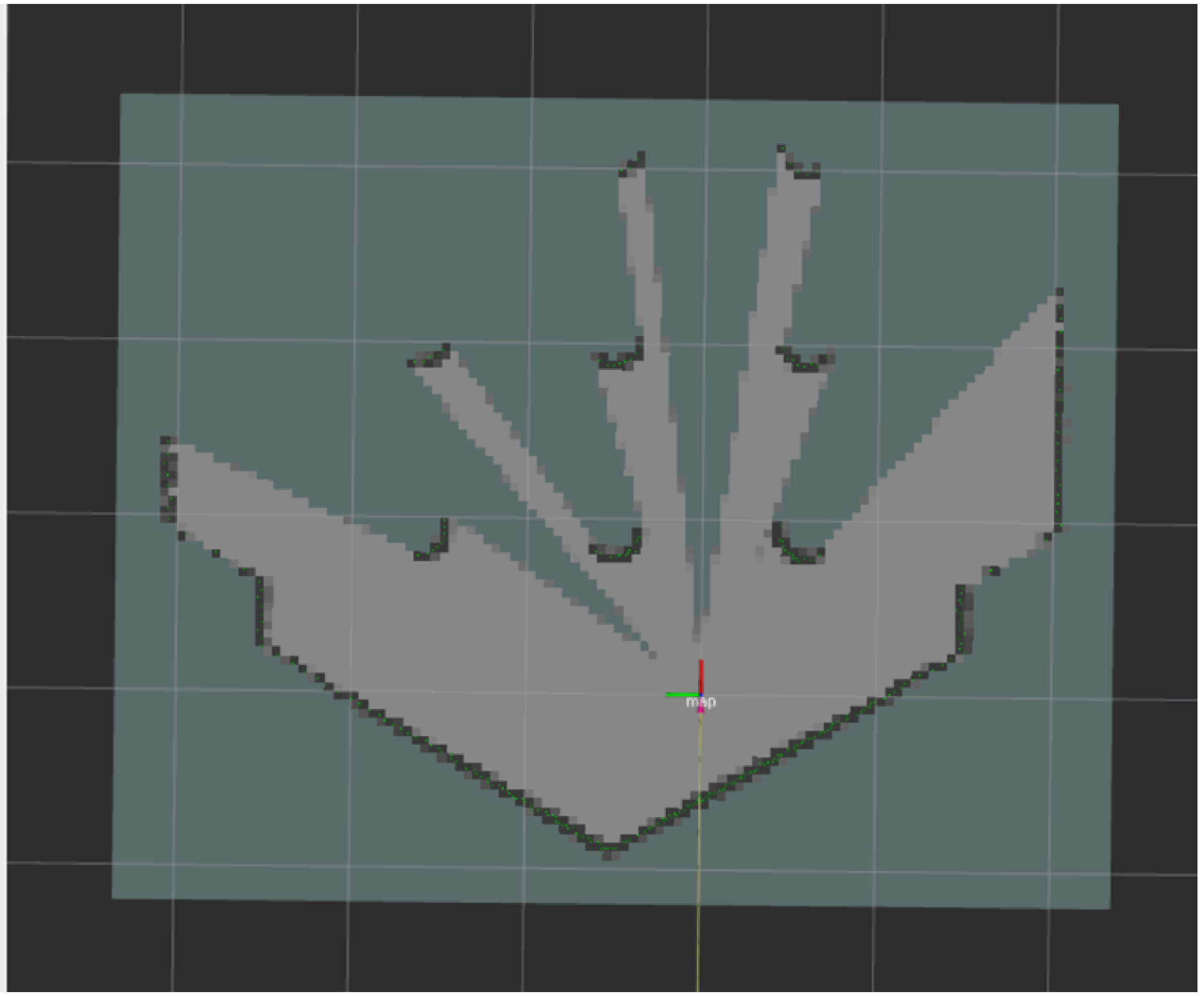


```
ubuntu@ubuntu:~$
```

Windows after launching Gazebo and SLAM nodes



World env



Scanned visualization

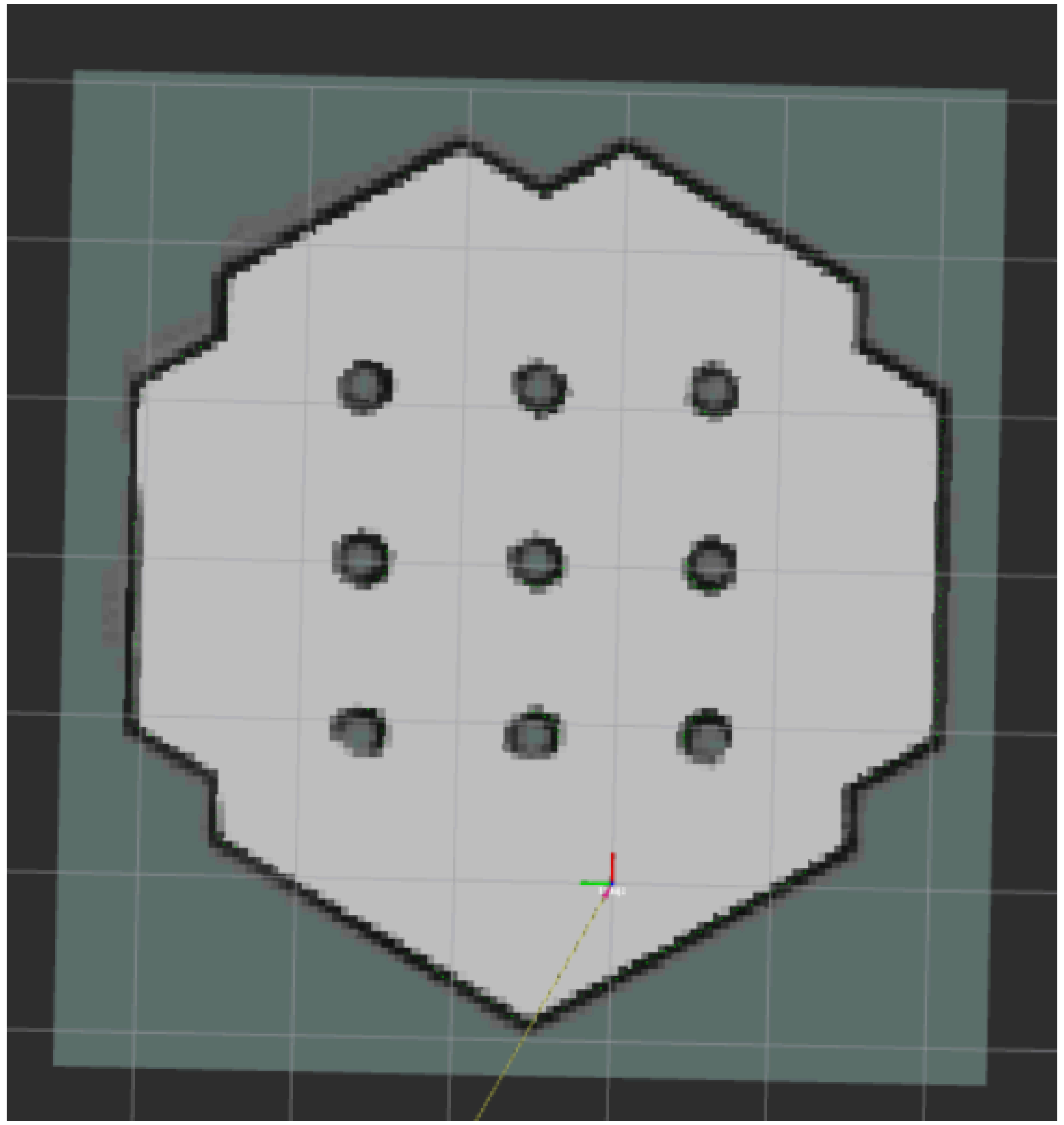
Control Turtlebot to create the map of World environment

```
Control Your Turtlebot3
Moving around

   w
 a s d
   x

w/x : increase/decrease linear velocity
a/d : increase/decrease angular velocity
space key, s : force stop

CTRL-C to quit
```



Until the entire map is created

Save the map

Open a new terminal

```
ros2 run nav2_map_server map_saver_cli -f /ws_slam/map
```

Enjoy navigation

Close the SLAM node

CTRL + C

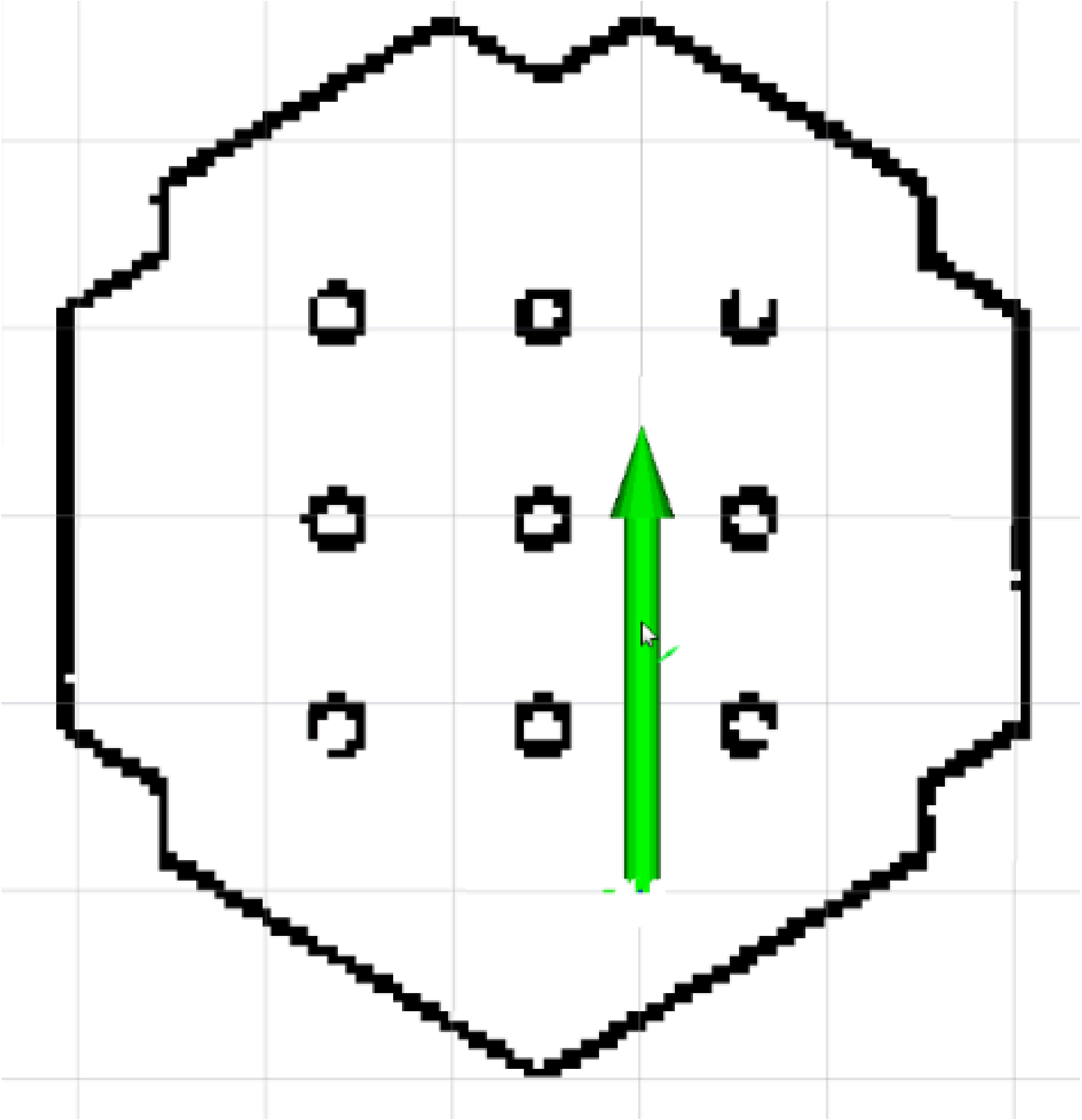
Run Navigation node

```
ros2 launch turtlebot3_navigation2 navigation2.launch.py
use_sim_time:=True map:=/ws_slam/map.yaml
```

Estimate Initial Pose

The goal is to tell the Navigation node where the current pose of the robot on the map

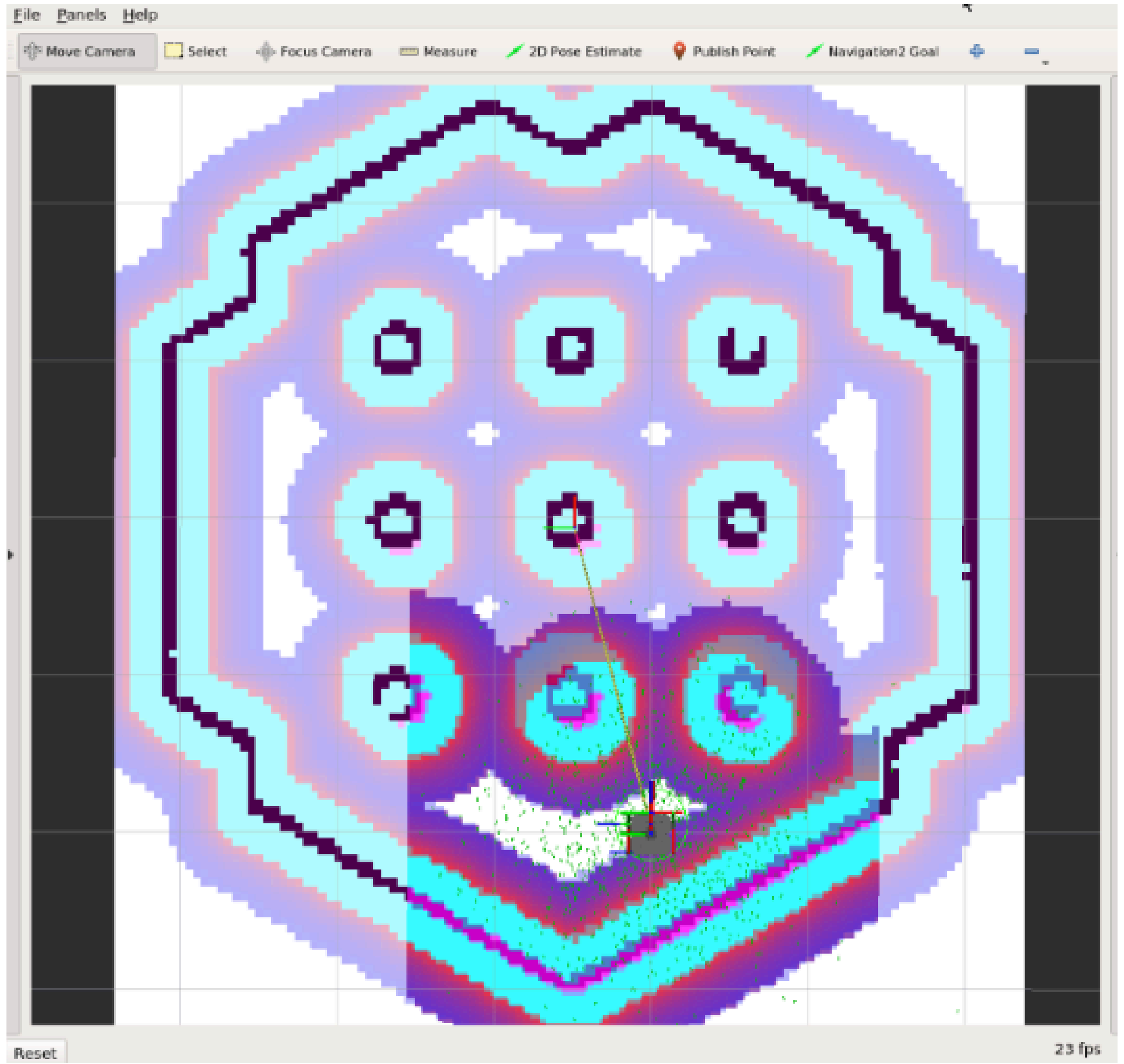
1. Click the **2D Pose Estimate** button in the RViz2 menu
2. Click on the map where the actual robot is located and drag the large green arrow toward the direction where the robot is facing.



Estimate Initial Pose

The goal is to tell the Navigation node where the current pose of the robot on the map

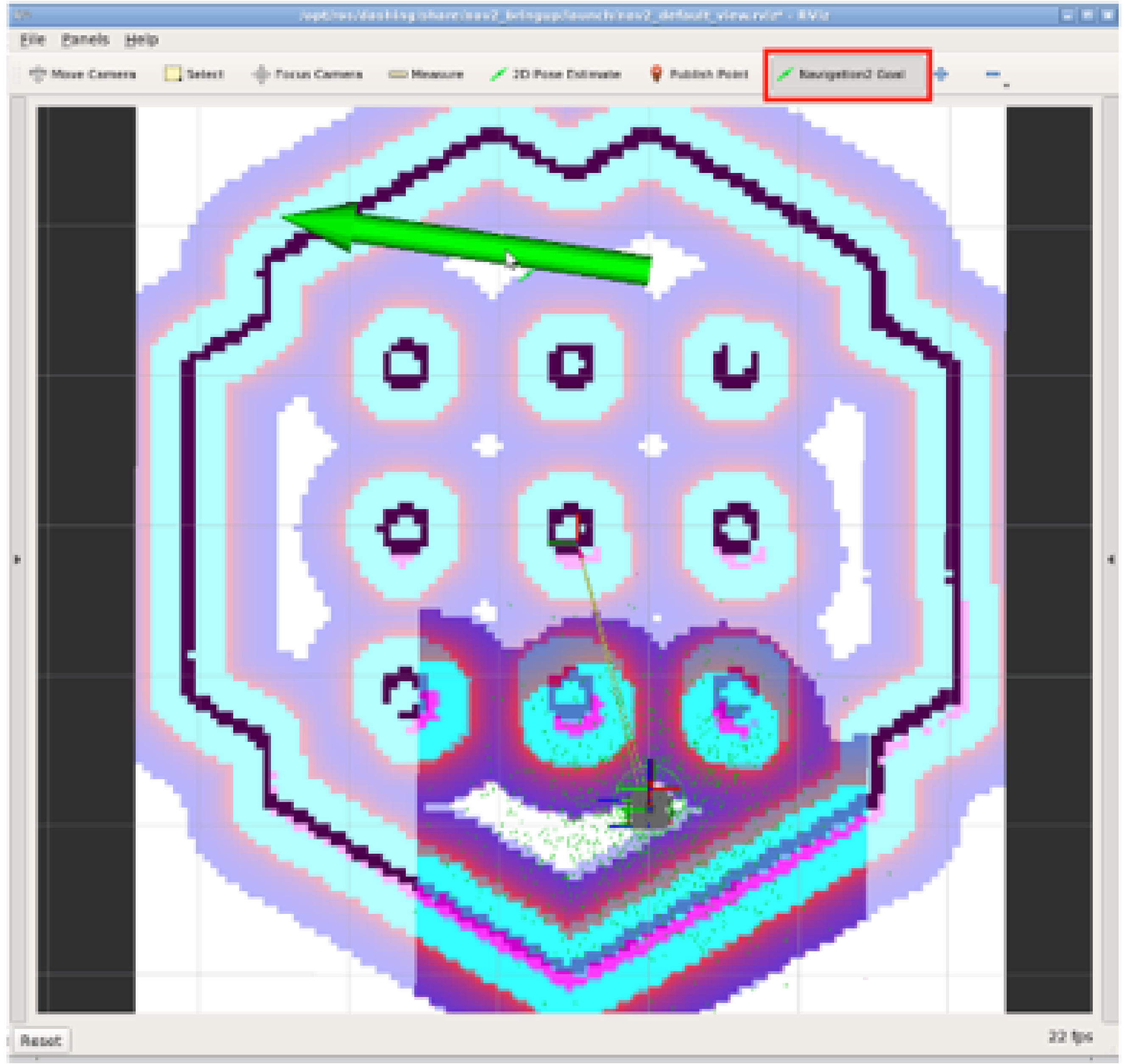
3. Repeat step 1 and 2 until the LDS sensor data is overlayed on the saved map.



Set Navigation Goal

The goal is to tell the Navigation node where the destination is

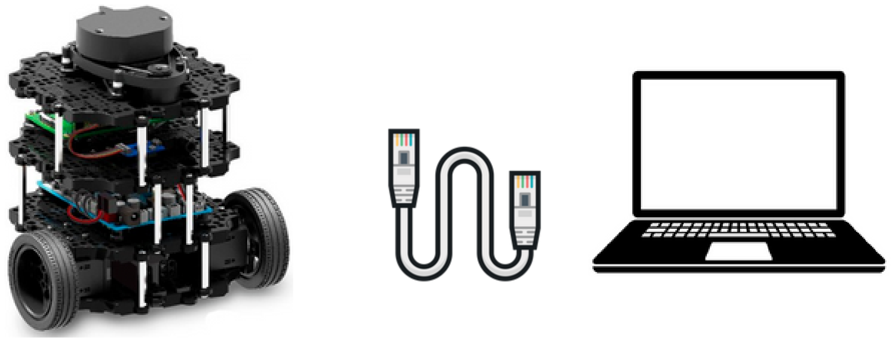
1. Click the Navigation2 Goal button in the RViz2 menu.
2. Click on the map to set the **destination** of the robot and **drag the green arrow** toward the direction where the robot will be facing.



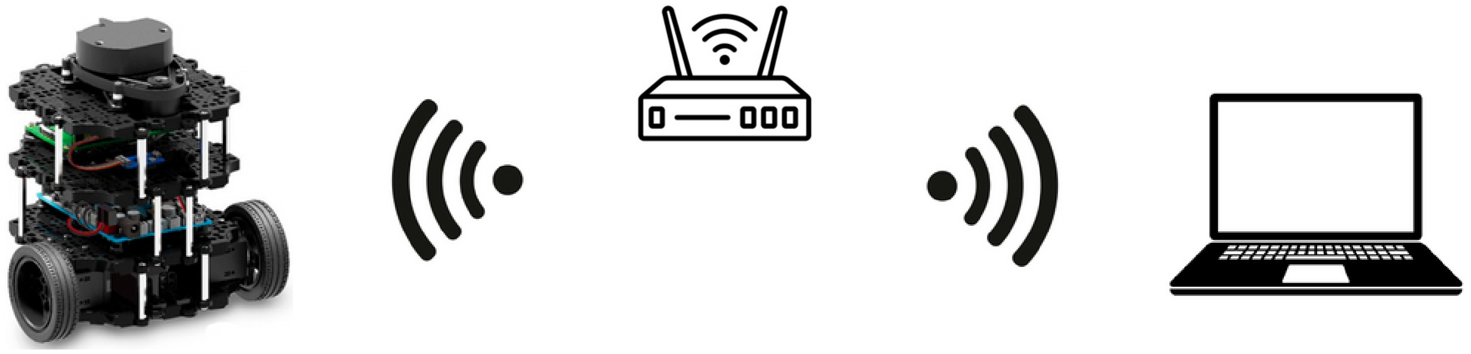
Simulation ends

In real Arena

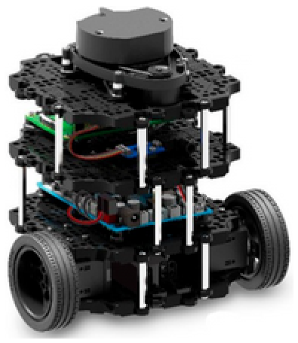
Network Configuration



Connect PC to Turtlebot via Ethernet cable
Configure Turtlebot to connect to WLAN



Connect your PC to “Access Point”
and connect Turtlebot to the common “Access Point”



Check the Turtlebot’s IP Address
using the monitor (Slide 5)

turtlebot’s_IP_Address

Network Configuration
Get into Turtlebot via Ethernet cable

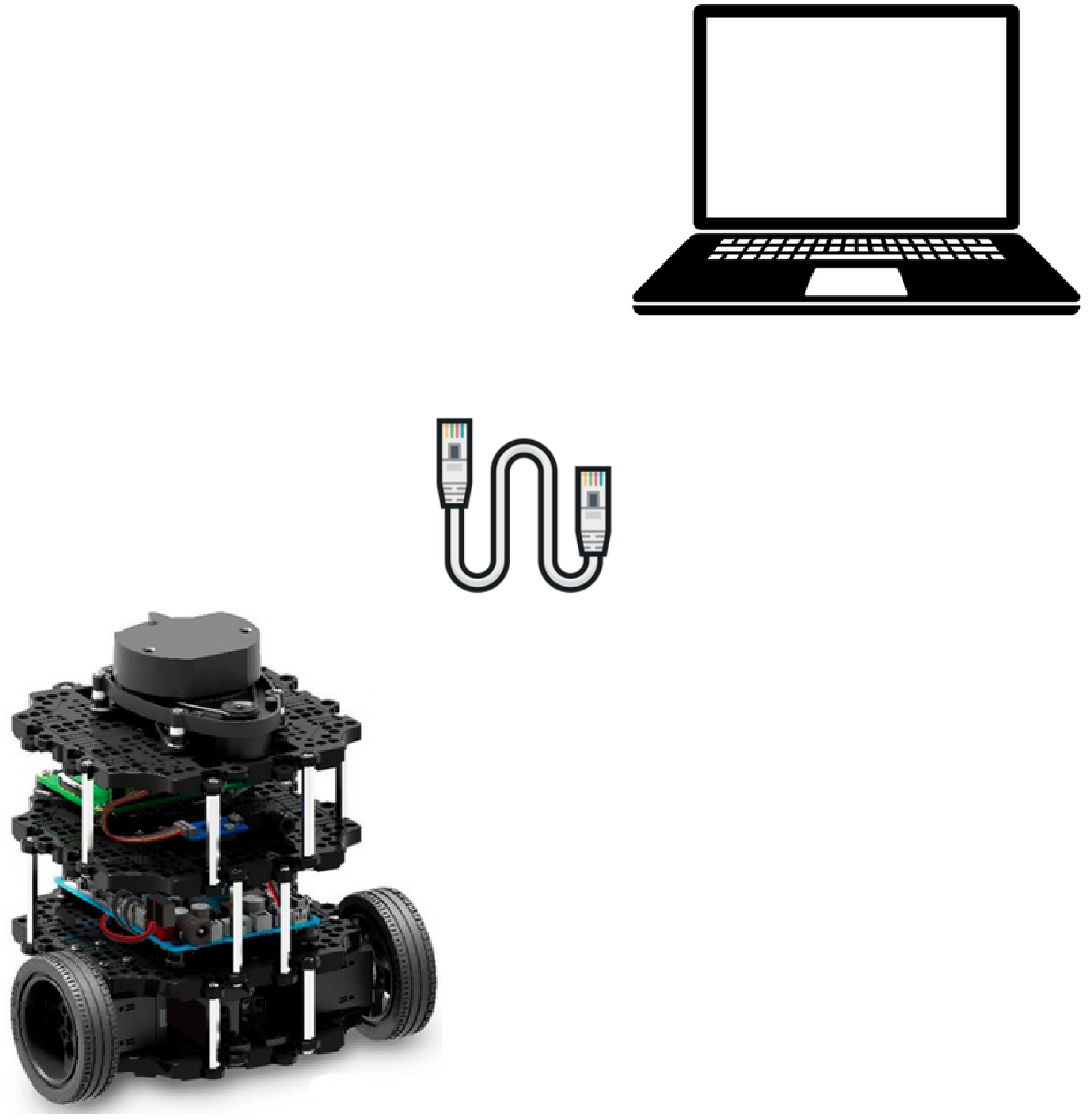
The goal is to configure Turtlebot to connect to WLAN

Before turning ON the Turtlebot

Connect

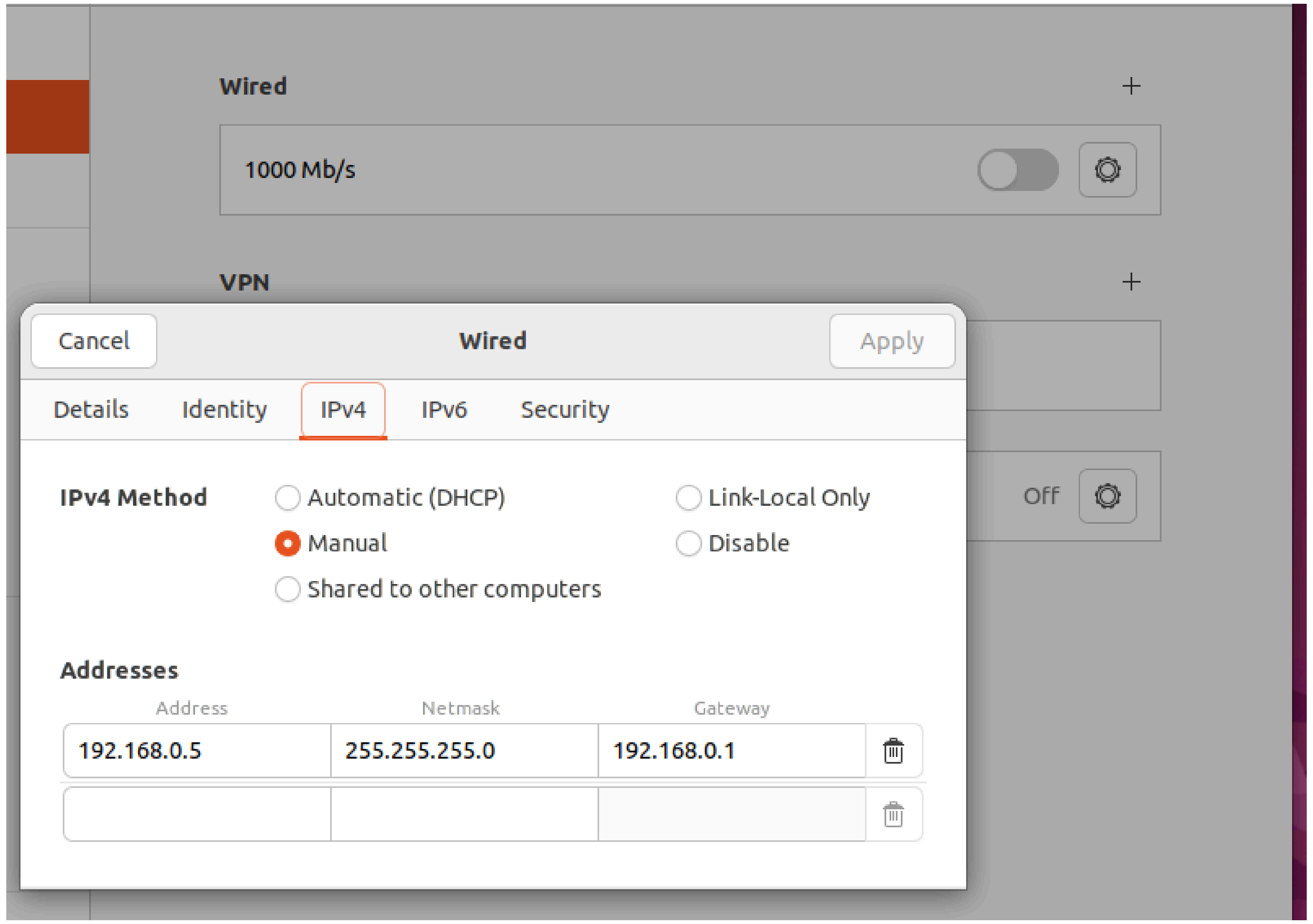
ETHERNET cable: Turtlebot to Laptop

Turn ON Turtlebot
Turn OFF WLAN on Laptop



Network Configuration
Get into Turtlebot via Ethernet cable

1. Click on the network icon in your system tray (on the upper top of the screen).
2. Select **Settings** → **Network**.
3. Choose your **Network** and click the ⚙ gear/settings button at the **Wired** section.
4. Go to the **IPv4** tab.
   - ○ Change the **Method** from Automatic (DHCP) to **Manual**.
   - ○ Under **Addresses**, click Add and enter:
     - ▪ Address: 192.168.0.5
     - ▪ Netmask: 255.255.255.0
     - ▪ Gateway: 192.168.0.1
5. Click **Apply**, then **disconnect** and **reconnect** the wired connection.



Wired

1000 Mb/s

Wired

IPv4 Method

○ Automatic (DHCP) ○ Link-Local Only

● Manual ○ Disable

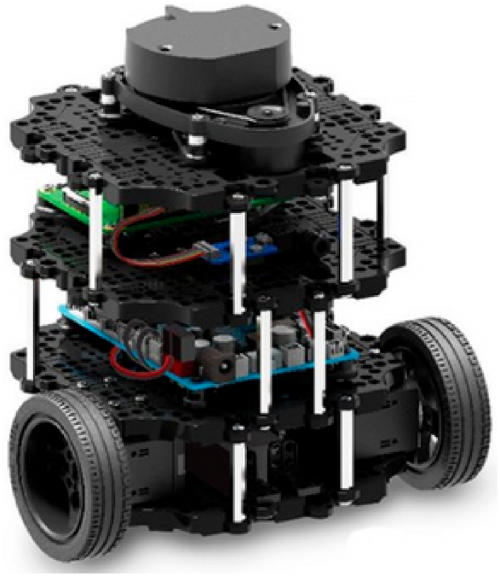
○ Shared to other computers

Addresses

Address	Netmask	Gateway
192.168.0.5	255.255.255.0	192.168.0.1

Network Configuration
Get into Turtlebot via Ethernet cable

The goal is to configure Turtlebot to connect to WLAN



Before turning ON the Turtlebot

Connect

ETHERNET cable: Turtlebot to Laptop

Turn ON Turtlebot
Turn OFF WLAN on Laptop

1 verify IP discovery

```
$ ping 192.168.0.12
```

Until receive

```
PING 192.168.0.12 (192.168.0.12) 56(84) bytes of data.
64 bytes from 192.168.0.12: icmp_seq=1 ttl=117 time=5.92 ms
64 bytes from 192.168.0.12: icmp_seq=2 ttl=117 time=7.07 ms
...
```

2 get into Turtlebot via SSH

```
$ ssh ubuntu@192.168.0.12
```

Login User
Username: ubuntu
Password: turtlebot

Network Configuration

WLAN: **Tututututu**

Password: **12345678Cham**

Connect PC to Turtlebot via Ethernet cable
Configure Turtlebot to connect to WLAN

3 open WLAN configuration via netplan Yaml file

```
ubuntu@192.168.0.12$ sudo nano /etc/netplan/
```

press **Tab**

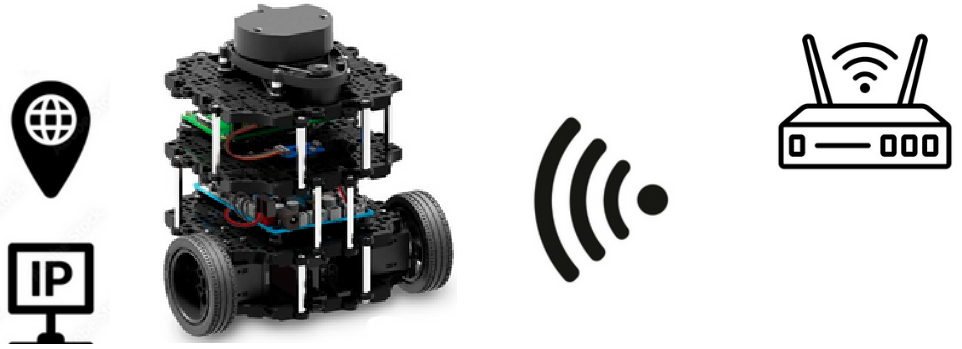
4 after modification save files and reboot

press **Ctrl + S**   press **Ctrl + X**

```
ubuntu@192.168.0.12$ sudo reboot
```

Network Configuration

After rebooting, Get into Turtlebot one more time
Check Turtlebot WLAN IP Address



ubuntu@192.168.0.12$ ip a

```
2: wlan0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
 link/ether AA:BB:CC:DD:EE:FF brd ff:ff:ff:ff:ff:ff
 inet 192.168.XXX.YYY/24 brd 192.168.XXX.255 scope global dynamic noprefixroute wlan0
 valid_lft 53412sec preferred_lft 53412sec
```

Turtlebot3_IP_Address: **192.168.XXX.YYY**

Network Configuration

Connect LAPTOP to the same WLAN

WLAN: **Tutututu**

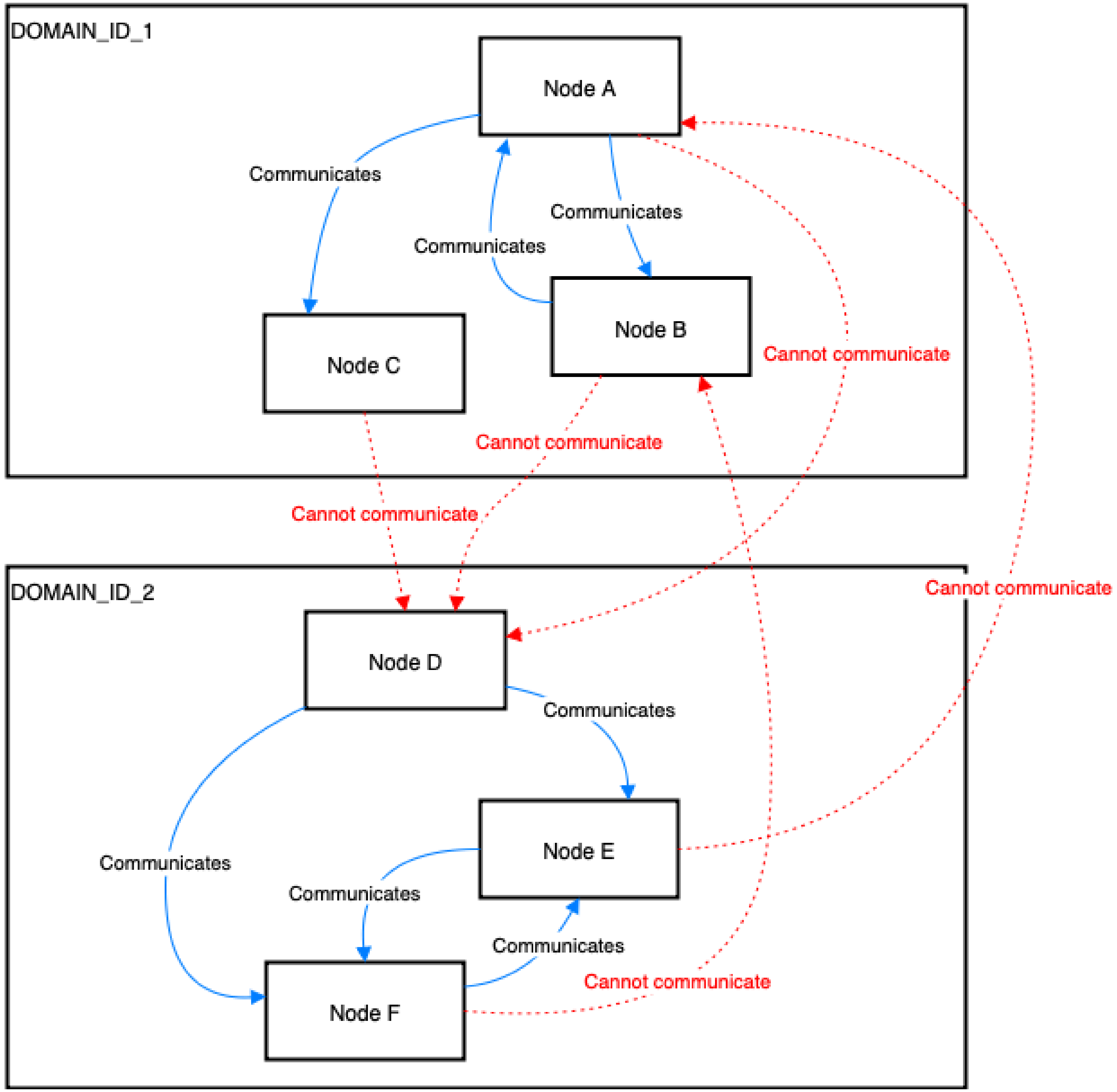
Password: **12345678Cham**

DOMAIN ID in ROS2

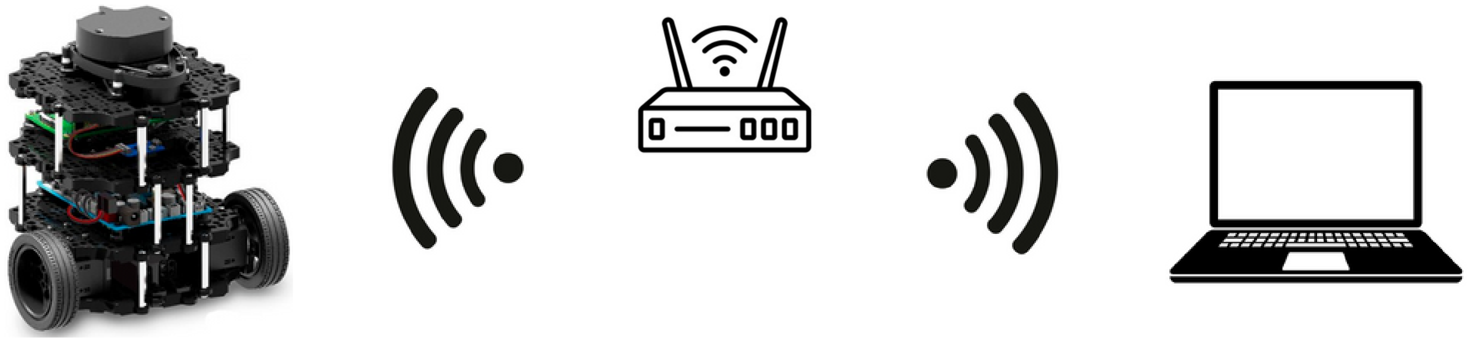
ROS2 communication between Nodes requires them to be in the same **DOMAIN ID**.

Nodes could be either Robots or Computational Unit

The **DOMAIN ID** is used to compute the UDP ports that will be used for discovery and communication



Get into Turtlebot via WLAN using Turtlebot3_IP_Address



1 verify IP discovery

```
$ ping 192.168.XXX.YYY
```

Until receive

```
PING 192.168.0.12 (192.168.0.12) 56(84) bytes of data.
64 bytes from 192.168.0.12: icmp_seq=1 ttl=117 time=5.92 ms
64 bytes from 192.168.0.12: icmp_seq=2 ttl=117 time=7.07 ms
...
```

2 get into Turtlebot via SSH

```
$ ssh ubuntu@192.168.XXX.YYY
```

Login User

Username: ubuntu
Password: turtlebot

Define ROS Domain ID in Turtlebot

```
$ nano ~/.bashrc
```

A Note file will open

The following lines must be added for setting the communication

```
$ export ROS_DOMAIN_ID=<Your_desired_ID>
```

```
$ source ~/.bashrc
```

After Saving and Closing, the file needs to be sourced

NOTE: Should be different from your colleagues

Define IP Addresses within Docker

```
$ gedit ~/.bashrc
```

A Note file will open

The following lines must be added for the set communication

```
$ export ROS_DOMAIN_ID=<Your_desired_ID>
```

```
$ source ~/.bashrc
```

After Saving and Closing, the file needs to be sourced

NOTE: Should be different from your colleagues

Bringup Turtlebot

```
$ ssh ubuntu@Turtlebot3_IP_Address
Example: ssh ubuntu@198.168.122.18
```

The command window will show the next message

password: turtlebot

Then

```
ros2 launch turtlebot3_bringup robot.launch.py
```

SLAM and Navigation

Follow the same procedure as in Simulation from **Page 4**

DEGGENDORF INSTITUTE OF TECHNOLOGY