LABORATORY GUIDE

AUTONOMOUS SYSTEMS

Rodrigo Ventura, Luís Custódio, Paulo Nogueira,
Rui Xavier, Marcelo Jacinto, João Luzio

2023/2024

PRE-REQUISITES

- **Personal laptop:**
  - Ubuntu 20.04 LTS / 18.04LTS or 16.04LTS (ROS1 is not supported in 22.04LTS)
  - Windows:
    1. Dual boot (with one of the Ubuntu versions mention above)
    2. Virtual machine with at least 4GB of RAM and 20GB of space (VMware recommended)*
    3. Using ROS 1 on Windows 10 is not recommended
    4. Using Docker and ROS 1 image not recommended (unless you are already familiar with it)
- **Laboratory computers (alternative):**
  - Ubuntu with ROS 1 pre-installed
  - Computers are connected to the deec-robots network (the same as all robots)
    - Username: acsdc
    - Password: acsdclsdc4

- Check https://si.tecnico.ulisboa.pt/software/vmware/ for free student license.
- VirtualBox can also be used, but not recommended.

INSTALLING LINUX AND ROS 1

- Each ROS 1 version is linked to an Ubuntu LTS release

Operating System	ROS 1 Version	Python Version
Ubuntu 16.04LTS	ROS Kinetic	2
Ubuntu 18.04LTS	ROS Melodic	2
Ubuntu 20.04LTS (recommended)	ROS Noetic	3
Ubuntu 22.04LTS	Not Supported	---

- Desktop-Full Install is recommended
- If you have troubles with the official ROS Noetic installation, follow these instructions

RECOMMENDATIONS

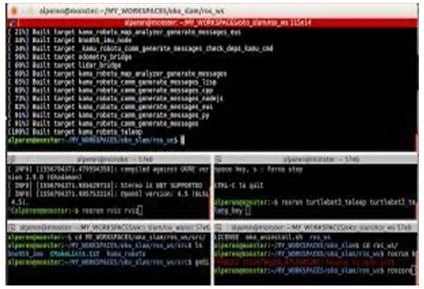
1. USE TERMINATOR

- On the laptop, run

```
sudo apt-get install terminator
```

Tips and Tricks

```
ctrl + shift + C (copy)
ctrl + shift + V (paste)
ctrl + shift + O (split the terminal horizontally)
ctrl + shift + E (split the terminal vertically)
ctrl + shift + Z (focus/unfocus a terminal window)
```



2. DEVELOP YOUR PROJECT IN PYTHON

GETTING STARTED WITH ROS

- “Introduction to the Robot Operating System (ROS)”, from Rodrigo Ventura **[mandatory if the first contact with ROS]**
- Short course videos from Rodrigo Ventura and João Avelino **[complementary]**
  - Part 1: https://www.youtube.com/watch?v=3aVYUAj7sr4&t=1205s&ab_channel=RodrigoVentura
  - Part 2: https://www.youtube.com/watch?v=zqpKWHHlgOA&ab_channel=RodrigoVentura

Short Course on ROS Programming
2020 edition - Part 1
http://bit.ly/3lYxZrC

Rodrigo Ventura João Avelino

Institute for Systems and Robotics
Instituto Superior Técnico
Portugal
{ rodrigo.ventura, javelino }@isr.tecnico.ulisboa.pt

2:04:42

Short Course on ROS Programming
2020 edition - Part 2
http://bit.ly/3lYxZrC

Rodrigo Ventura João Avelino

Institute for Systems and Robotics
Instituto Superior Técnico
Portugal
{ rodrigo.ventura, javelino }@isr.tecnico.ulisboa.pt

4:01:53

THE LABORATORY

- Located at the 5th floor of the North Tower – Room LSDC4

WIFI (Pioneers, Alphabots and Turtlebots):

Network: deec-robots

Password: shakeytherobot

IP range: 192.168.28.<id>

Lab computers:

Username: acsdc

Password: acsdclsdc4

THE LAB COMPUTERS

- Lab computers shall be used **to test code** or **to record bag files** and **NOT for development**.
- Check the Wi-Fi connection of the lab computers to deec-robots network
  (if not working properly, please contact Mr. Manuel Ribeiro).
- ROS_HOSTNAME AND ROS_IP have been added as an alias in the lab computers, they can also be exported by simply running EXPORT_HOST.
  These export commands are needed in each new terminal that will use ROS to communicate with the robot.

THE PIONEER ROBOTS

- The Pioneer robots are configured with:
  1. A Raspberry Pi with Ubuntu 18.04LTS and ROS 1 Melodic
  2. The P2OS ROS package

- Relevant topic of P2OS node:
  - Subscribes:
    - **/cmd_vel** (geometry_msgs/Twist)
  - Publishes:
    - **/pose** (nav_msgs/Odometry)
    - **/sonar** (p2os_driver/SonarArray)

- Hokuyo laser rangefinder:
  - Publishes:
    - **/scan** (sensor_msgs/LaserScan)

CREDENTIALS:

Username: pi

Password: acsdclsdc4

Network: deec-robots

Password: shakeytherobot

IP range: 192.168.28.[17…23]

THE PIONEER ROBOTS [Connecting]

1. On a new terminal of the Laptop/Lab computer, and SSH into the RPi of the **Pioneer**

```
ssh pi@192.168.28.[17…23]
```

2. Start a roscore instance inside the RPi of the **Pioneer**

```
roscore
```

3. Open another terminal in the Laptop/Lab computer and SSH into the RPi again

```
ssh pi@192.168.28.[17…23]
rosrun p2os_driver p2os_driver _port:="/dev/ttyUSB0"     (or)
rosrun p2os_driver p2os_driver _port:="/dev/ttyUSB0" _use_sonar:="true"
```

4. Run in the Laptop/Lab computer (or append to you ~/.bashrc file)

```
export ROS_MASTER_URI=http://192.168.28.[17…23]:11311  [Pioneer/RPi IP]
export ROS_HOSTNAME=192.168.[27/28].xxx  [Lab computer IP/Laptop]
export ROS_IP=192.168.[27/28].xxx  [Lab computer IP/Laptop]
```

If `ipconfig` is not available, use
`ip address`
or install net-tools to run
`ifconfig`:

```
apt update
apt install net-tools
```

THE PIONEER ROBOTS [Communications]

5. Test the communications: can the computer see the robot’s topics?

```
rostopic list
```

6. Interesting topics to query: “/pose”, “/sonar”

```
rostopic echo "/pose"
rostopic info "/pose"
rostopic hz "/pose"
```

7. Test motion commands *

```
rostopic pub /cmd_vel geometry_msgs/Twist '[0.0, 0.0, 0.0]' '[0.0, 0.0, 0.5]' (or)
rostopic pub -r 1 /cmd_vel geometry_msgs/Twist '[0.1, 0.0, 0.0]' '[0.0, 0.0, 0.0]'
```

8. To record topics in rosbags **

```
rosbag record -a (for all topics)
rosbag record /<topic name> (for specific topics)
```

* Check that the motors are enabled, the topic `/motor_state` should be 1. If not, press the 'motors' button on the robot. Press again to stop!

** If path unspecified, bags get recorded in the folder where the command is run. See http://wiki.ros.org/rosbag/Commandline#record

THE PIONEER ROBOTS [Operation]

9. Install the teleop package - run in the laptop once **[skip for lab computers]**

```
sudo apt-get install ros-noetic-teleop-twist-keyboard
```

10. Control the robot. On the laptop/Lab computer, run

```
rosrun teleop_twist_keyboard teleop_twist_keyboard.py
```

THE PIONEER ROBOTS [Hokuyo rangefinder]

1. To install on your laptop (alternatively connect directly to the RPi on the robot, and skip this step)

```
sudo apt-get install ros-noetic-urg-node
```

2. Connect the Hokuyo and run

```
chmod a+rw /dev/ttyACM0
```

3. Launch the **urg** node on the device you connected the hokuyo (roscore is assumed to be running)

```
rosrun urg_node urg_node /dev/ttyACM0
```

4. To test with the RVIZ, run on the laptop/Lab computer the following command and see next slide (it is assumed that step 4 from slide 8 was executed)

```
rosrun rviz rviz
```

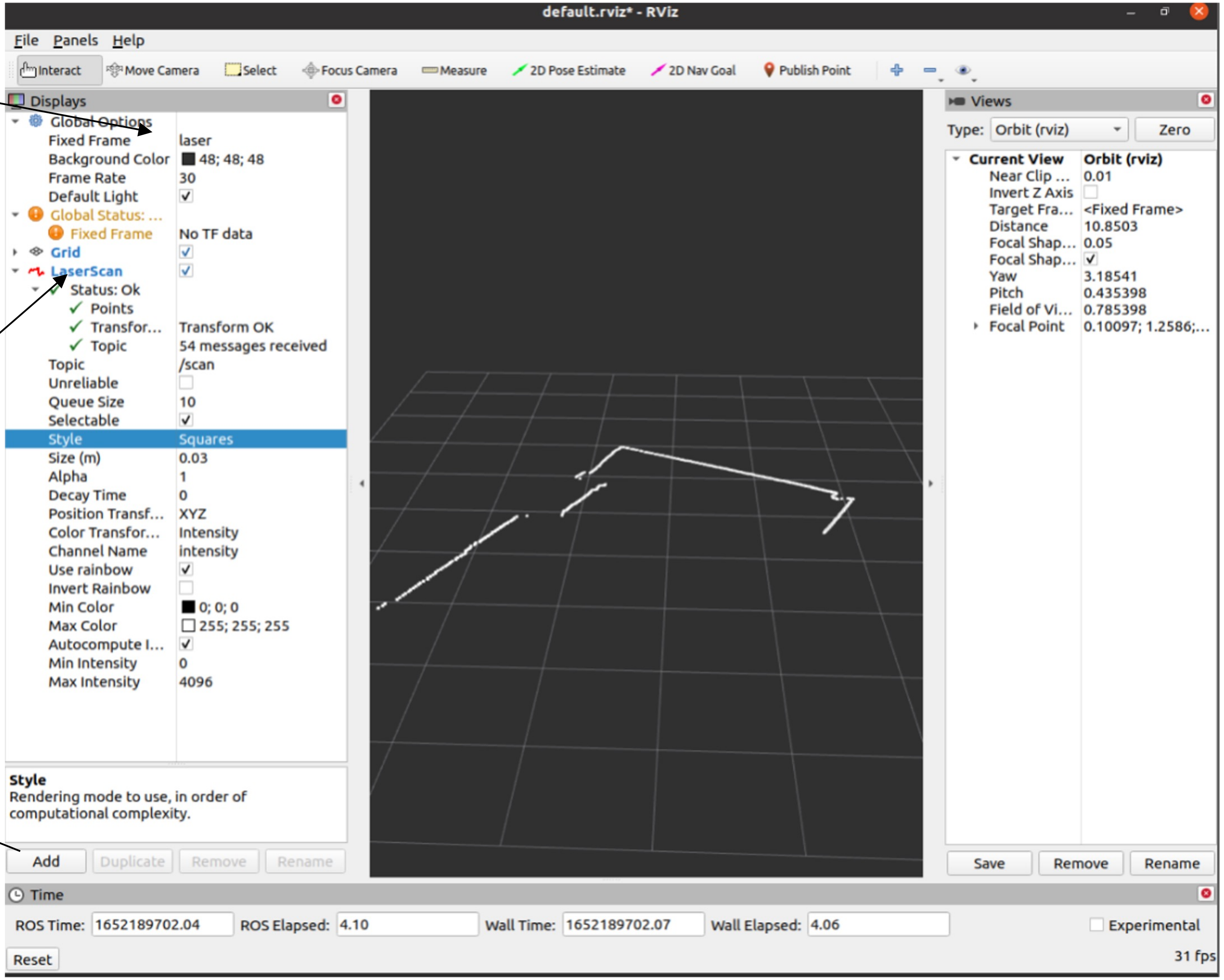
Request access to
Mr. Manuel Ribeiro

See http://wiki.ros.org/urg_node

THE PIONEER ROBOTS [Hokuyo rangefinder]

Write “laser”, since no referential was defined.

Add “LaserScan” node



Global Options	
Fixed Frame	laser
Background Color	■ 48; 48; 48
Frame Rate	30
Default Light	☑
Global Status: ...	
Fixed Frame	No TF data
Grid	☑
LaserScan	☑
Status: Ok	
✓ Points	
✓ Transfor...	Transform OK
✓ Topic	54 messages received
Topic	/scan
Unreliable	□
Queue Size	10
Selectable	☑
Style	Squares
Size (m)	0.03
Alpha	1
Decay Time	0
Position Transf...	XYZ
Color Transfor...	Intensity
Channel Name	intensity
Use rainbow	☑
Invert Rainbow	□
Min Color	■ 0; 0; 0
Max Color	□ 255; 255; 255
Autocompute I...	☑
Min Intensity	0
Max Intensity	4096

Style
Rendering mode to use, in order of computational complexity.

Type: Orbit (rviz)

Current View	Orbit (rviz)
Near Clip ...	0.01
Invert Z Axis	□
Target Fra...	<Fixed Frame>
Distance	10.8503
Focal Shap...	0.05
Focal Shap...	☑
Yaw	3.18541
Pitch	0.435398
Field of Vi...	0.785398
Focal Point	0.10097; 1.2586;...

ROS Time: 1652189702.04 ROS Elapsed: 4.10 Wall Time: 1652189702.07 Wall Elapsed: 4.06 □ Experimental

31 fps

THE PIONEER ROBOTS [Microsoft Kinect]

1. To use the Microsoft Kinect with the Pioneer robots, you must install a few packages on your laptop. Please follow these steps to install them:

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install git-core cmake freeglut3-dev pkg-config build-essential libxmu-dev libxi-dev libusb-1.0-0-dev

cd ~/src
git clone https://github.com/OpenKinect/libfreenect.git
cd libfreenect
mkdir build
cd build
cmake -L ..
Make
sudo make install
sudo ldconfig /usr/local/lib64/

cd ~/catkin_ws/src
git clone https://github.com/ros-drivers/freenect_stack.git
cd ..
catkin buildsource ~/catkin_ws/devel/setup.bash
```

Request access to
Mr. Manuel Ribeiro

THE PIONEER ROBOTS [Microsoft Kinect]

2. To run the node that publishes the images coming from the Kinect, you must first run this command on a terminal, after connecting the USB to your computer and the Kinect to the power outlet or the robots:

```
dmesg
```

3. One of the few last lines will contain similar information to this (check the device number and serial number). In this case, the USB device number is 21.

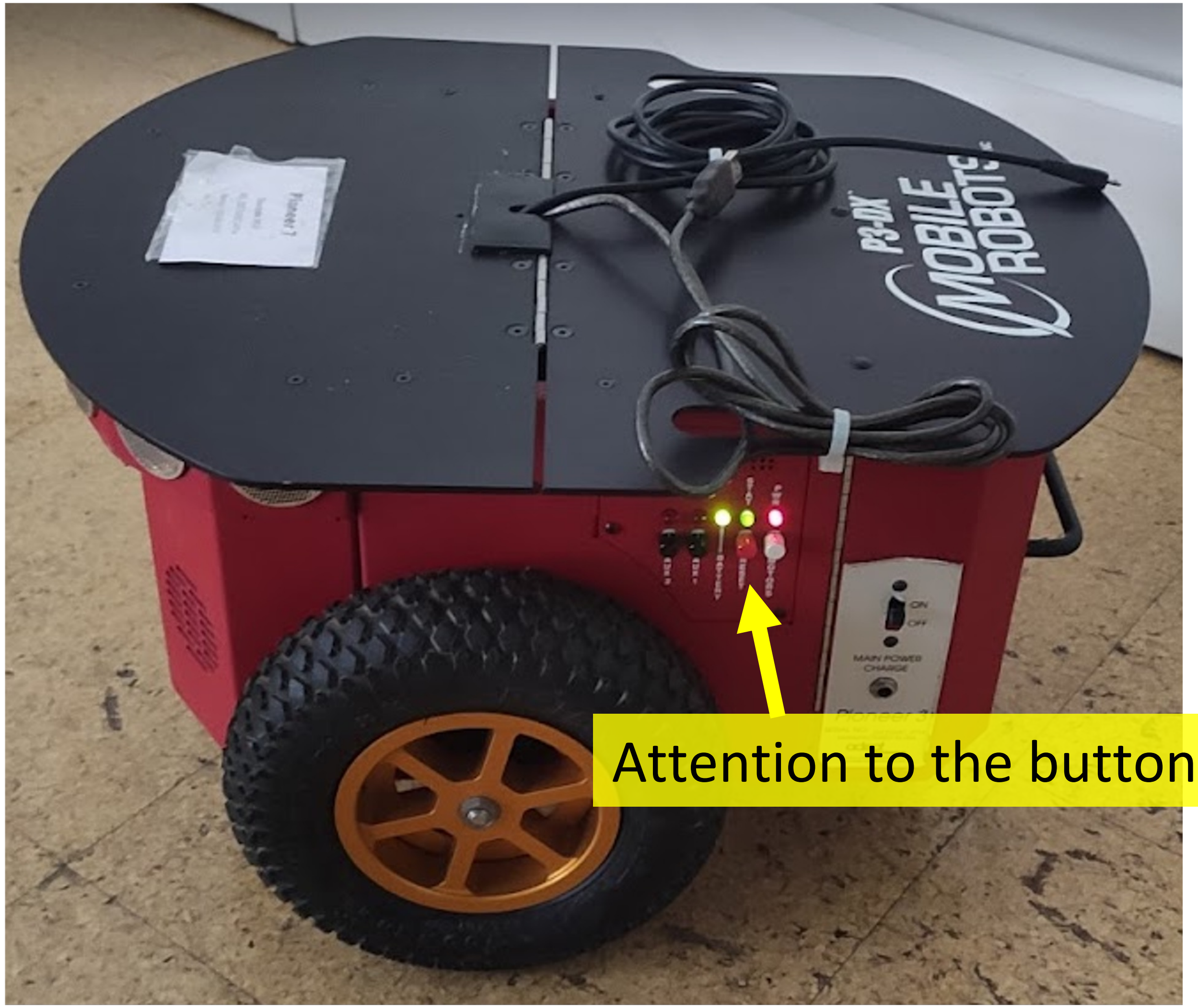
```
[11151.635280] usb 1-2.2: new full-speed USB device number 21 using xhci_hcd
```

4. Now, launch the camera node using this command (replace the *device_id* by the number you got from the **dmesg** command):

```
roslaunch freenect_launch freenect.launch device_id:=21 depth_processing:=false
```

NOTE: If for some reason you can't still see the image from the camera in your computer, then try to replace the device_id by the serial number also shown in **dmesg**.

THE PIONEER ROBOTS [Charging]

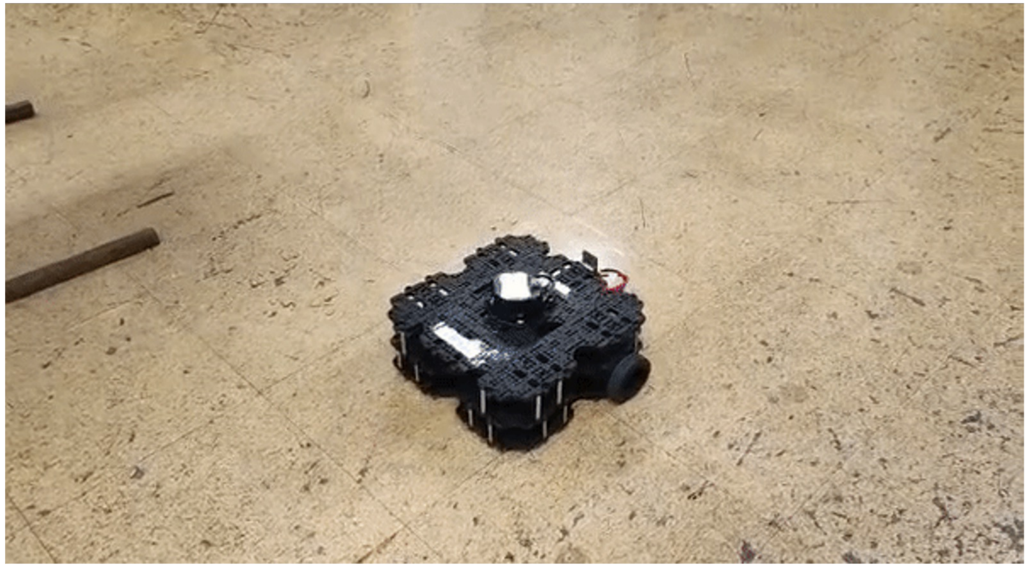


Attention to the buttons!

Charger set

THE TURTLEBOT ROBOTS

- The Turtlebots are configured with a Raspberry Pi with Ubuntu 18.04LTS and ROS 1 Melodic
- Relevant topics:
  - Subscribes:
    - **/cmd_vel** (geometry_msgs/Twist)
    - **/reset** (std_msgs/Empty)
  - Publishes:
    - **/odom** (nav_msgs/Odometry)
    - **/tf** (tf2_msgs/tfMessage)
    - **/scan** (sensor_msgs/LaserScan)
- Additional resources:
  - http://wiki.ros.org/turtlebot3_bringup
  - http://wiki.ros.org/hls_lfcd_lds_driver
  - http://wiki.ros.org/sensor_msgs/Tutorials



CREDENTIALS:

Username: user

Password: user

Network: deec-robots

Password: shakeytherobot

IP range: 192.168.28.[11…15]

THE TURTLEBOT ROBOTS [Connecting]

1. On a new terminal of the Laptop/Lab computer, and SSH into the RPi of the **Turtlebot**

```
ssh user@192.168.28.[11…15]
```

2. Start a roscore instance inside the RPi of the **Turtlebot**

```
roscore
```

3. Open another terminal in the Laptop/Lab computer and SSH into the RPi again

```
ssh user@192.168.28.[11…15]
```

4. Sync the robot time **[not mandatory]**

```
sudo apt-get install ntpdate
sudo ntpdate ntp.ubuntu.com
```

5. Launch the robot drivers

```
roslaunch turtlebot3_bringup turtlebot3_robot.launch
```

THE TURTLEBOT ROBOTS [Laptop Setup]

6. Install the Turtlebot3 packages - run in the laptop once **[skip for lab computers]**.

```
sudo apt-get install ros-noetic-dynamixel-sdk
sudo apt-get install ros-noetic-turtlebot3-msgs
sudo apt-get install ros-noetic-turtlebot3
sudo apt-get install ros-noetic-teleop-twist-keyboard
```

7. Additional configurations. Run in the Laptop/Lab computer (or append to you ~/.bashrc file)

```
export TURTLEBOT3_MODEL=waffle_pi
export TURTLEBOT3_NAME=waffle4                                [TurtleBot3 181818 on the stick]
export TURTLEBOT3_IP=192.168.28.[11…15]                       [TurtleBot IP]
export TURTLEBOT3_NUMBER=[11…15]                              [Last numbers of the TurtleBot3 IP]
export ROS_MASTER_URI=http://192.168.28.[11…15]:11311    [TurtleBot3 IP]
export ROS_HOSTNAME=192.168.[27/28].XXX                       [lab computer / laptop IP]
export ROS_IP=192.168.[27/28].XXX                             [lab computer / laptop IP]
```

If `ipconfig` is not available, use `ip address` or install net-tools to run `ifconfig`:

```
apt update
apt install net-tools
```

THE TURTLEBOT ROBOTS [Communications]

8. Test the communications: can the computer see the robot’s topics?

```
rostopic list
```

9. Interesting topics to query: “/odom”, “/scan”

```
rostopic echo "/odom"
rostopic info "/odom"
rostopic hz "/odom"
```

10. Test motion commands

```
rostopic pub /cmd_vel geometry_msgs/Twist '[0.0, 0.0, 0.0]' '[0.0, 0.0, 0.5]' (or)
rostopic pub -r 1 /cmd_vel geometry_msgs/Twist '[0.1, 0.0, 0.0]' '[0.0, 0.0, 0.0]'
```

11. To record topics in rosbags *

```
rosbag record -a (for all topics)
rosbag record /<topic name> (for specific topics)
```

* If path unspecified, bags get recorded in the folder where the command is run. See http://wiki.ros.org/rosbag/Commandline#record

THE TURTLEBOT ROBOTS [Operation]

12. Control the robot. On the laptop/Lab computer, run

```
roslaunch turtlebot3_teleop turtlebot3_teleop_key.launch
```

Accessing the robot camera

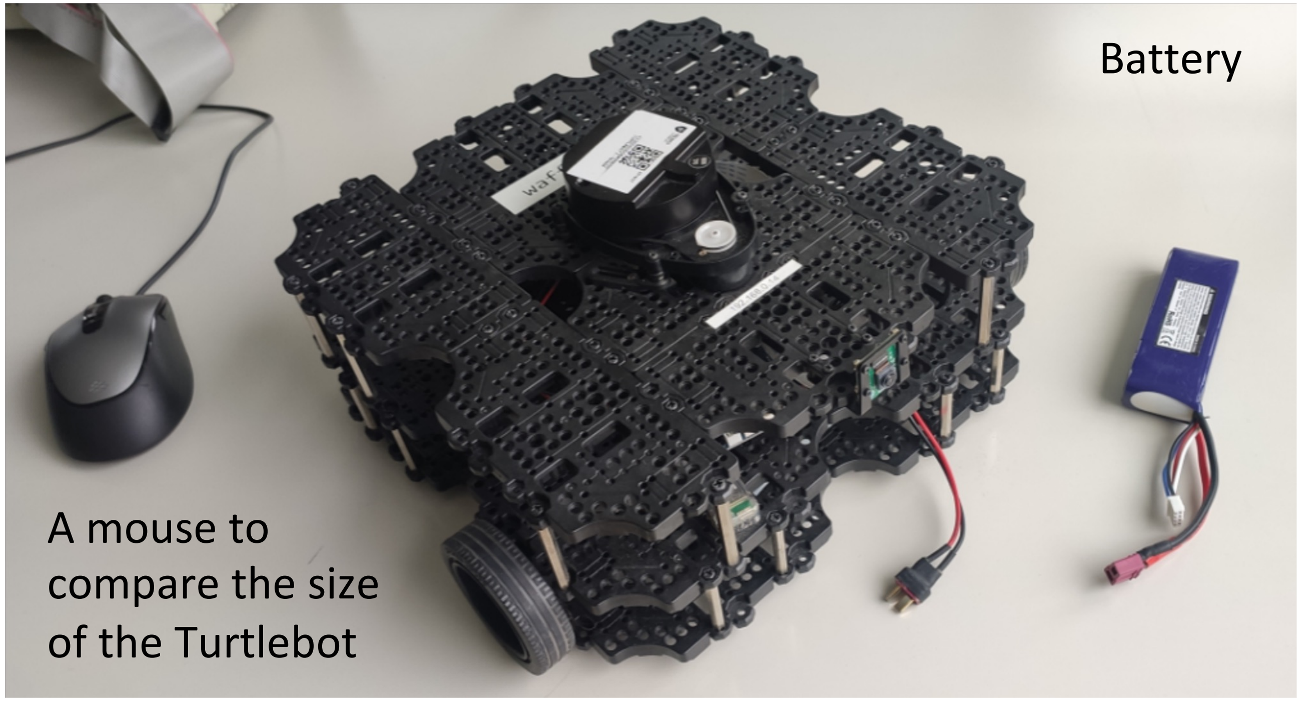
1. Launch the camera node on the robot

```
roslaunch turtlebot3_bringup turtlebot3_rpicamera.launch
```

2. To see the camera feed, run on the laptop/Lab computer

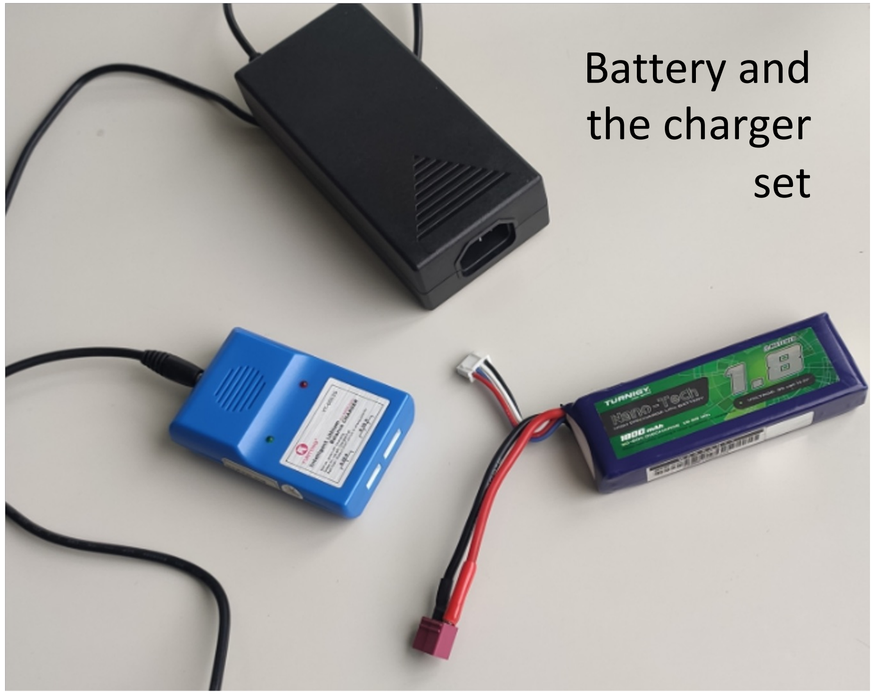
```
rosrun rqt_image_view rqt_image_view
```

THE TURTLEBOT ROBOTS [Charging]



Battery

A mouse to compare the size of the Turtlebot



Battery and the charger set

THE ALPHABOT ROBOTS

- The Alphabots are configured with a Raspberry Pi with Ubuntu 16.04LTS and ROS 1 Kinetic
- Relevant topics:
  - Subscribes:
    - **/cmd_vel** (geometry_msgs/Twist)
  - Publishes:
    - **/camera/compressed** (sensor_msgs/CompressedImage)
- Additional resources:
  - ROS for waveshare Alphabot2
  - Raspicam node
  - Fiducial slam

Warning

These robots do NOT have wheel odometry, IMU or LiDAR

CREDENTIALS:

Username: alphabot2

Password: alphabot2

Network: deec-robots

Password: shakeytherobot

IP range: 192.168.28.[50…63]

THE ALPHABOT ROBOTS [Connecting]

TÉCNICO LISBOA

1. On a new terminal of the Laptop/Lab computer, and SSH into the RPi of the **Alphabot**

```
ssh alphabot2@192.168.28.[50…63]
```

2. Start a roscore instance inside the RPi of the **Alphabot**

```
roscore
```

3. Open another terminal in the Laptop/Lab computer and SSH into the RPi again

```
ssh alphabot2@192.168.28.[50…63]
```

4. Launch the robot drivers

```
roslaunch waveshare_alphabot2 Alphabot2.launch enable_raw:=false
teleop:=false detect_aruco:=false
```

Options:

- enable-raw (default: False): whether to publish raw image or only compressed image topic
- teleop (default: False): to launch the keyboard teleoperation node with the vehicle drivers
- detect_aruco (default: False): to launch the aruco detector with the vehicle drivers

THE ALPHABOT ROBOTS [Communications]

5. Run in the Laptop/Lab computer (or append to you ~/.bashrc file)

```
export ROS_MASTER_URI=http://192.168.28.[50…63]:11311       [Alphabot-RPi IP]
export ROS_HOSTNAME=192.168.28.XXX                     [Lab computer IP/Laptop]
export ROS_IP=192.168.28.XXX                           [Lab computer IP/Laptop]
```

6. Test the communications: can the computer see the robot's topics?

```
rostopic list
```

7. Interesting topics to query: "/camera/compressed"

```
rostopic echo "/camera/compressed"
rostopic info "/camera/compressed"
rostopic hz "/camera/compressed"
```

8. Test motion commands

```
rostopic pub /cmd_vel geometry_msgs/Twist '[0.0, 0.0, 0.0]' '[0.0, 0.0, 0.5]' (or)
rostopic pub -r 1 /cmd_vel geometry_msgs/Twist '[0.1, 0.0, 0.0]' '[0.0, 0.0, 0.0]'
```

THE ALPHABOT ROBOTS [Operation]

9. Install the teleop package - run in the laptop once **[skip for lab computers]**

```
sudo apt-get install ros-noetic-teleop-twist-keyboard
```

10. Control the robot. On the laptop/Lab computer, run

```
rosrun teleop_twist_keyboard teleop_twist_keyboard.py
```

11. To record topics in rosbags *

```
rosbag record -a (for all topics)
rosbag record /<topic name> (for specific topics)
```

Accessing the robot camera

1. To see the camera feed, run on the laptop/Lab computer

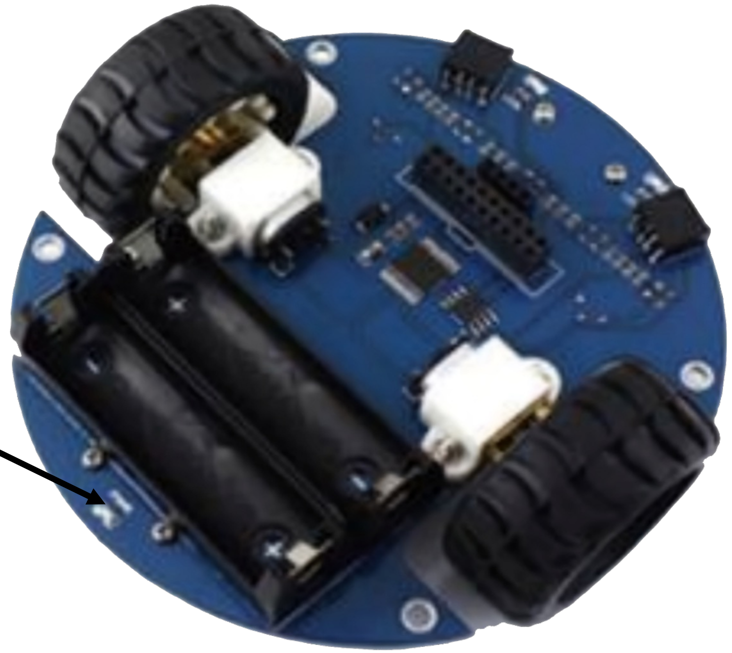
```
rosrun rqt_image_view rosrun rqt_image_view
```

2. You can use the following command to control the pan and tilt of the camera (with rate 1hz)

```
rostopic pub /pan_tilt waveshare_alphabot2/Pan_Tilt "{pan: 0.0, tilt: 1.0}” -r1
```

THE ALPHABOT ROBOTS [Charging]

Charge via micro-USB (5V)



ROS CHEAT-SHEET

Relevant terms to search

Package	Bag
Node	Launch Files
Topic	Parameters
Publisher	TF
Subscriber	RVIZ
Message	Gazebo

Graphical user tools

```
rosrun rviz rviz

rosrun rqt_image_view rqt_image_view

rqt    (can be used to monitor)

rosrun rqt_tf_tree rqt_tf_tree

rosrun rqt_plot rqt_plot
```

Terminal user tools

```
roscore
rosrun <package_name> <node_name>
roslaunch <package_name> <launch_file>
roscd <package_name>

rostopic list
rostopic info <topic_name>
rostopic hz <topic_name>

rosnode list
rosnode info <node_name>

rosbag play <options>
rosbag record <options>

roscreate-pkg <options>

rosservice list
rosservice call <options>

rosmsg list
```

ROS EXTRA NOTES

- Often forgotten – set the environment variables used by ROS

```
source devel/setup.bash
```

- Confirm the definition of the ROS packages path

```
echo $ROS_PACKAGE_PATH
```

- ROS/Linux editors: vim or nano [nano is recommended]
- Make use of the .bashrc file

```
source /opt/ros/noetic/setup.bash
source ~/catkin_ws/devel/setup.bash
export EDITOR='nano -w'
```

IMPORTANT

- Copy a file from the robot to your laptop/Lab computer

```
scp <robot_username>@<robot_ip>:<path_to_file> <location_in_your_computer>
```

ROS EXTRA NOTES [2]

- Explore the use of **`roslaunch`** and launch files
- Bag files may occupy too much space.

  Suggestion: record only the required topics and include compression

```
rosbag record –j <topics>
```

- If problems are detected, run

```
roswtf
```

- To install dependencies of a package, use rosdep

```
sudo apt-get install rosdep
rosdep init
rosdep install <package-name>
```

- ROS log files are located at

```
~/.ros/log
```

ROS EXTRA RESOURCES

- Official ROS website: https://www.ros.org/
- ROS Wiki: http://wiki.ros.org/
- Core ROS Tutorials: http://wiki.ros.org/ROS/Tutorials
  - Beginner Level **[all bullets recommended]**
  - Intermediate Level **[roslaunch tips]**
- TF2 Tutorials: http://wiki.ros.org/tf2/Tutorials/ https://articulatedrobotics.xyz/ready-for-ros-6-tf/
- Robot Model: http://wiki.ros.org/robot_model_tutorials
- Visualization: http://wiki.ros.org/visualization/Tutorials
- Navigation: http://wiki.ros.org/navigation/Tutorials
- MATLAB [**use ROS to record bags and read them in MATLAB**]
  Open and parse rosbag log file (since R2022a supports ROS Noetic): https://www.mathworks.com/help/ros/ref/rosbag.html

- Aruco detector: https://wiki.ros.org/aruco_detect
- Camera Calibrator: http://wiki.ros.org/camera_calibration