

# RSE2107A – Lecture 2

Introduction to Python and ROS

# Agenda

01

Python Crash Course

02

ROS Introduction  
1-Hour Video Session

FILTERS



### Python Crash Course For Beginners

514K views · 3 years ago

Traversy Media

In this crash course we will be going over Python programming basics like variables, data types and structures, functions, loops, ...



### Python Tutorial in 30 Minutes (Crash Course for Absolute Beginners)

54K views · 1 year ago

thenevboston

Welcome to my Python Crash Course! In the next 30 minutes, I am going to try and teach you as much Python as I can!



Introduction | Comments | Variables | Data Types | Strings | Booleans | Operators | Lists | Tuples | Set... 17 chapters



### Python for Beginners - Learn Python in 1 Hour

4.9M views · 1 year ago

Programming with Mosh

#Python, #MachineLearning, #WebDevelopment Python Exercises for Beginners: <https://goo.gl/1XnQB1> ☆ My Favorite Python ...



Introduction | What You Can Do With Python | Your First Python Program | Variables | Receiving Inpu... 19 chapters



### Learn Python - Full Course for Beginners [Tutorial]

32M views · 3 years ago

freeCodeCamp.org

☆ Contents ☆ (0:00) Introduction (1:45) Installing Python & PyCharm (6:40) Setup & Hello World (10:23) ...

CC



Introduction | Installing Python & PyCharm | Setup & Hello World | Drawing a Shape | Variables & Dat... 35 chapters

 FILTERS


## ROS Tutorial (ROS1) - ROS Noetic 2H30 [Crash Course]

2.5K views • 2 months ago



Robotics Back-End

This ROS tutorial [Crash Course] will get you started with ROS Noetic in no time. Step by step and hands-on lessons only! You will ...



Install and Setup ROS Noetic | Understand what is a ROS Node | Create and Setup a Catkin...

11 chapters ▾



## ROS Courses (English)

ROBOTIS OpenSourceTeam

Chapter 01 Robot Software Platform • 7:27

Chapter 02 Robot Operating System • 11:48

[VIEW FULL PLAYLIST](#)


## An Introduction to ROS, the Robot Operating System: Intro to ROS (2/6)

36K views • 1 year ago



linuxfestnorthwest

Sid Faber Canonical The Robot Operating System (ROS) is a flexible, open source framework for writing robot software. ROS has ...



## Programming for Robotics (ROS) Course 1

339K views • 5 years ago



Robotic Systems Lab: Legged Robotics at ETH Zürich

Lecturers: Péter Fankhauser, Dominic Jud, Martin Wermelinger Course 1 covers following topics: – ROS architecture & philosophy ...

CC



Course Structure | Exercise Evaluation | Overview Course 1 | What is ROS? | History of ROS | ROS...

27 chapters ▾

# Python Crash Course

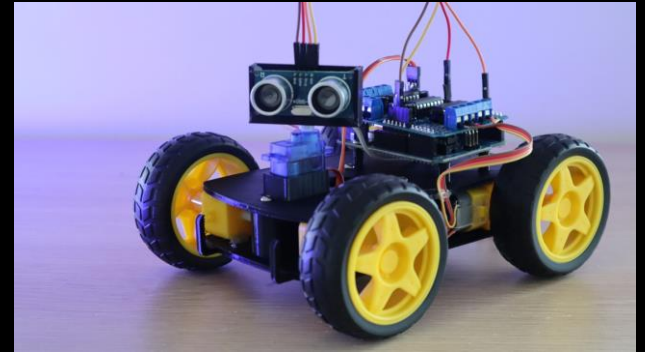
# Python Crash Course

- Python version
- Variables
- Types
- Conditional
- For/While loops
- Functions
- Imports

# Robot Operating System (ROS)

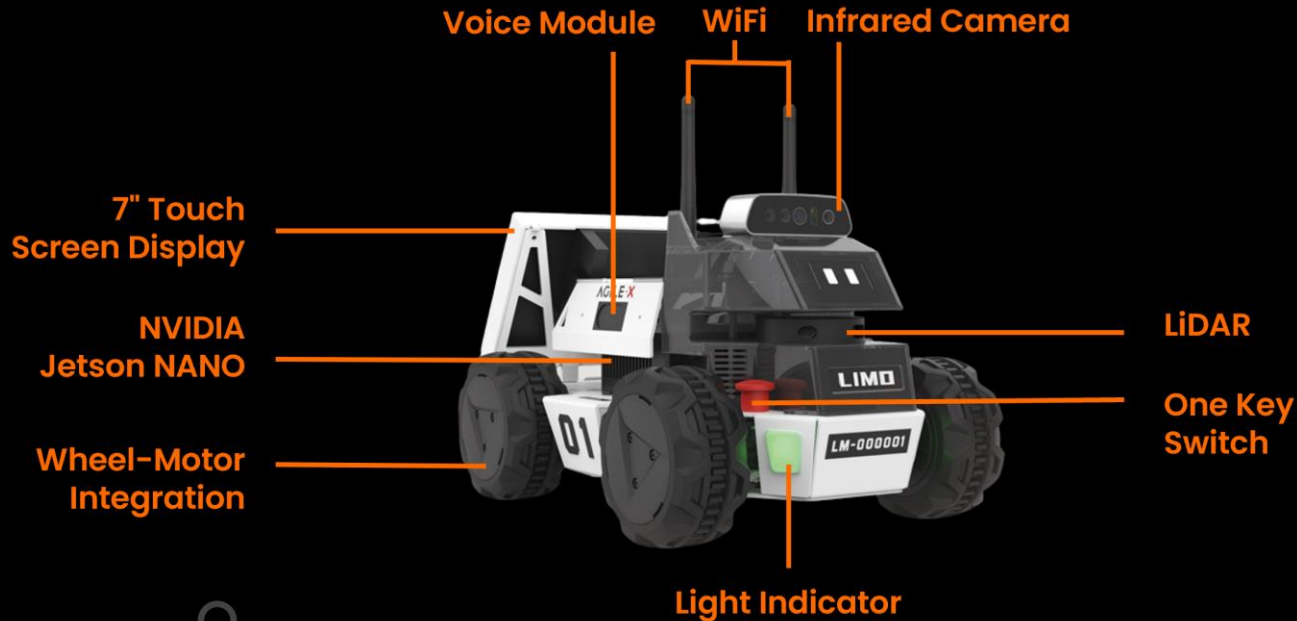
# Why use ROS?

- Arduino based robot
  - Ultrasonic sensor to detect distance from obstacles
  - Robot stops when in the proximity of an obstacle, else it continues moving forward.
  - Does this require ROS?





# Challenges of Doing It All by Yourself



# Robot Evolution

## 1. Microcontroller based

Line follower

Maze solver ...

## 2. Higher level logic that can benefit from PC

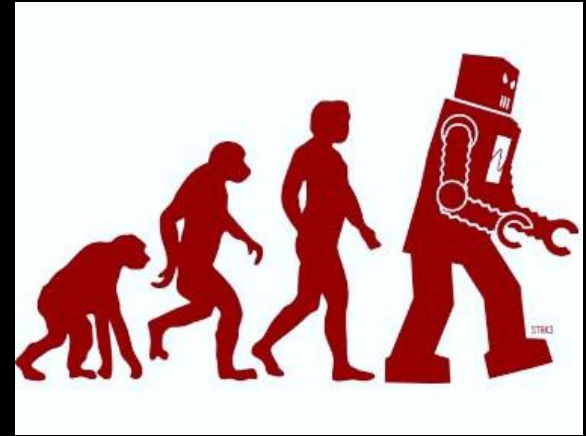
Sensor processing

Robot arm control ...

## 3. Higher level logic requires PC

Path planning

SLAM ...



By Nslsmith at en.wikipedia, CC BY-SA 3.0,



By Kazyakuruma - Own work, CC BY-SA 4.0,



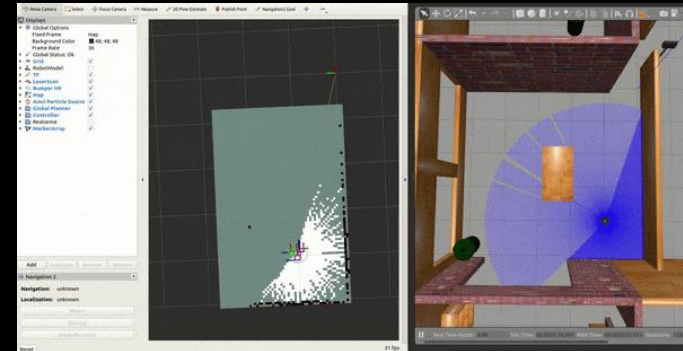


# Weston Robot Challenges of Doing It All by Yourself



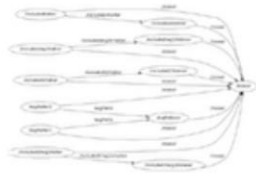
# What is ROS?

- An architecture for distributed inter-process communication
- Multilanguage interface (C++, Python, Lua, Java, etc.)
- Tools for runtime and data analysis
- Packages for common algorithms (software) and drivers (hardware)
- Open source (with some limitations)



# What is ROS?

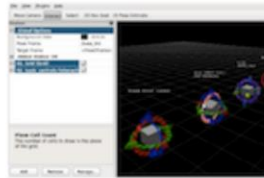
## ROS = Robot Operating System



Plumbing

- Process management
- Inter-process communication
- Device drivers

+



Tools

- Simulation
- Visualization
- Graphical user interface
- Data logging

+



Capabilities

- Control
- Planning
- Perception
- Mapping
- Manipulation

+



ros.org

Ecosystem

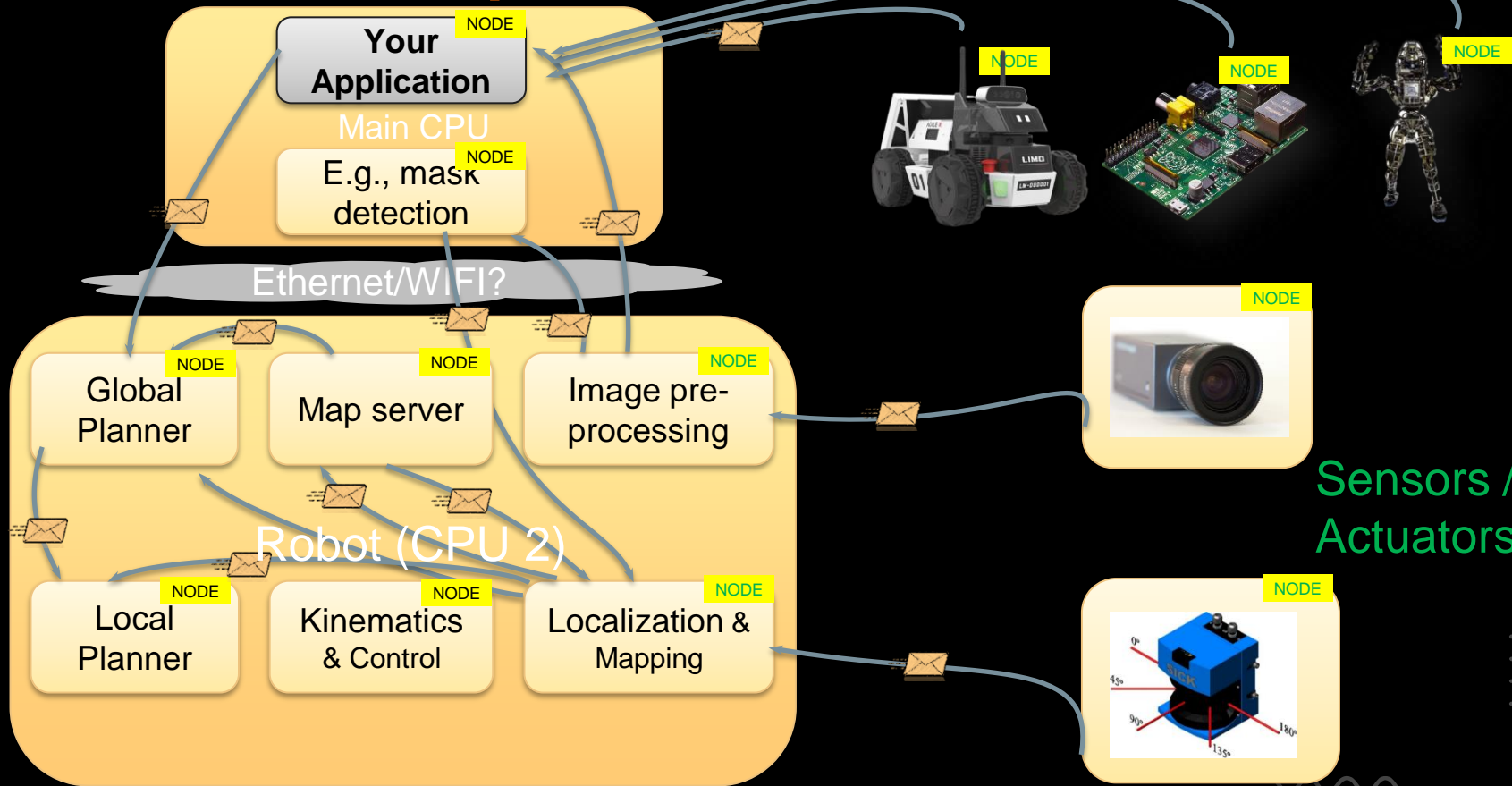
- Package organization
- Software distribution
- Documentation
- Tutorials

# What ROS Is Not?

- An actual operating system
- A programming language
- A development environment or IDE
- A hard real-time architecture

# ROS Concept: Node

*Weston Robot*





# ROS Versions

ROS (1)

Centralized: ROS Master

ROS 2

De-centralized: DDS

Video Session: [https://www.youtube.com/watch?v=yn638LmVwlw&ab\\_channel=RoboticsBack-End](https://www.youtube.com/watch?v=yn638LmVwlw&ab_channel=RoboticsBack-End)

# Agenda

- ROS Distributions
- ROS Nodes
- ROS Topics
- Basic Commands
- ROS Tools

# ROS (1) Distributions

- ROS Kinetic (Ubuntu 16.04)
- ROS Melodic (Ubuntu 18.04)
- ROS Noetic (Ubuntu 20.04)

# ROS Build System

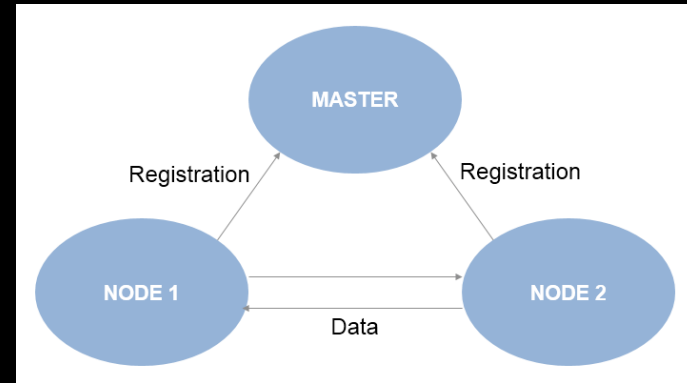
- catkin (just a name):
  - It is a workspace to create your own packages/programs
  - you can create your own (but not recommended now)
- ROS Melodic (Ubuntu 18.04)
- ROS Noetic (Ubuntu 20.04)

# ROS Node

- An executable that is responsible for a specific functionality of the robot.
- Able to communicate with other nodes.
- Nodes are programs often written in C++ or python.

# ROS Node

- ROS Master
  - Provides naming and registration services to all other nodes in the ROS system.
  - This helps nodes to coordinate the ROS network.
  - Prerequisite for ROS-based systems.
  - Run 'roscore' in your terminal to launch ROS Master.



# 'roscore' in terminal

```
roscore http://txp-450-231d:11311/101x55
txp@txp-450-231d:~$ roscore
... logging to /home/txp/.ros/log/6d2c2ff0-d0fa-11ec-86a6-48e2446e5af3/roslaunch-txp-450-231d-10717.1
og
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://txp-450-231d:46525/
ros_comm version 1.14.13

SUMMARY
=====

PARAMETERS
* /rostdistro: melodic
* /rosversion: 1.14.13

NODES

auto-starting new master
process[master]: started with pid [10727]
ROS_MASTER_URI=http://txp-450-231d:11311/

setting /run_id to 6d2c2ff0-d0fa-11ec-86a6-48e2446e5af3
process[rosout-1]: started with pid [10738]
started core service [/rosout]
█
```

# ROS Node

- Nodes can be of several types:
  - Publisher
  - Subscriber
  - Service client
  - Service server
- *Nodes can act as one or more of the 4 mentioned types.*



# Publishers & Subscribers

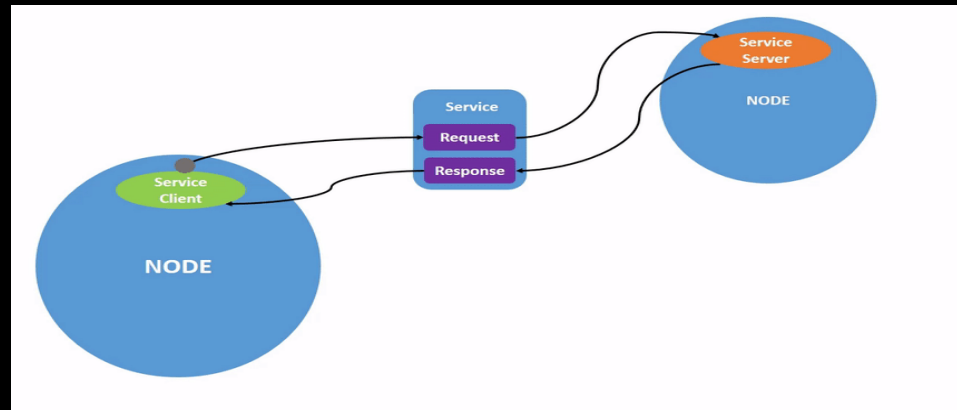
- Publishers
  - Node that publishes at least one type of ROS message to a topic.
- Subscribers
  - “Interested” nodes can then subscribe to these messages.

# ROS Topic

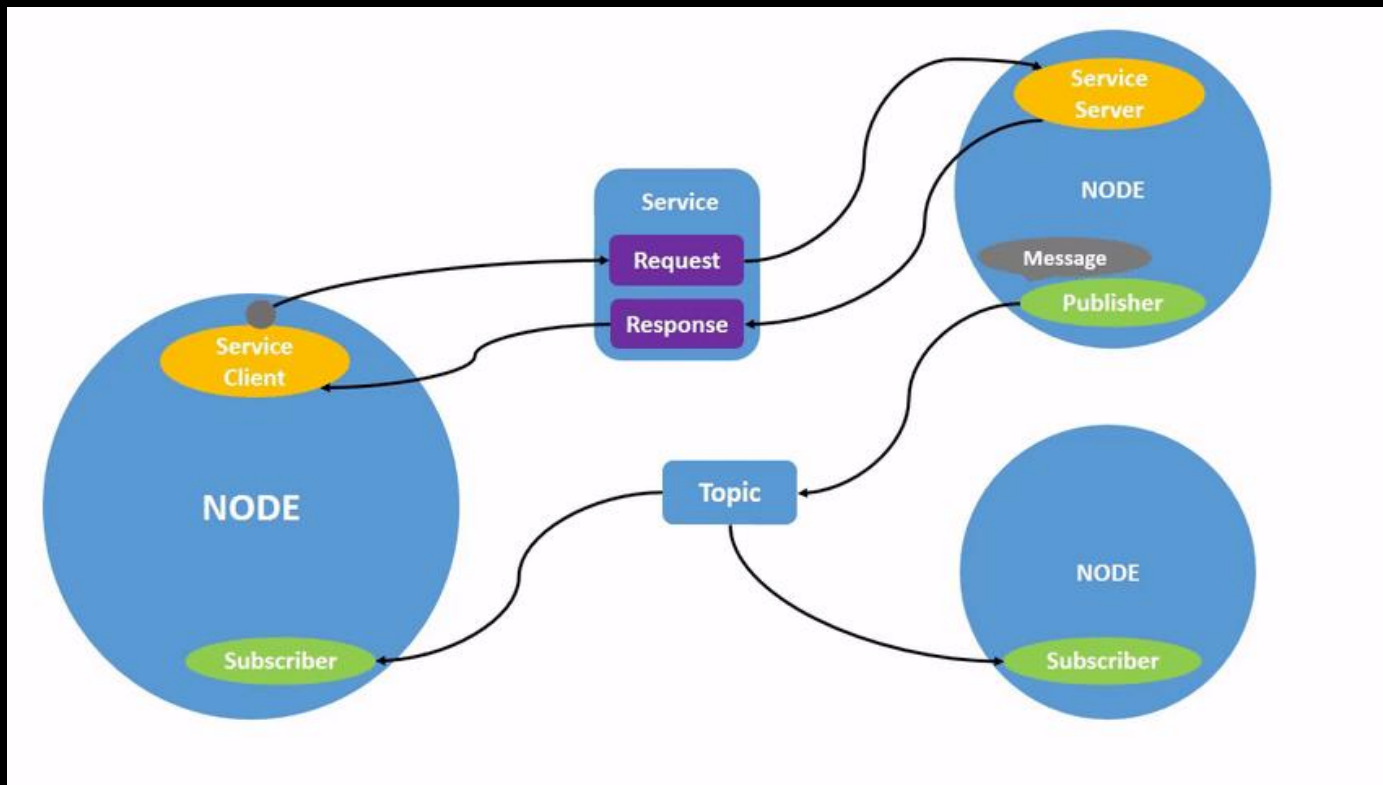
- **Topic:** Named channel over which nodes exchange messages
  - Subscriber nodes must subscribe to a **topic** to receive messages, and Publisher nodes must publish messages to a **topic**.
- **Message:** ROS data type used when subscribing/publishing to topics

# ROS Service

- **Service:** Allows nodes to send a request and receive a response
  - Service client sends a request (message) to a Service server. This prompts the server to execute the request and return a response (message) back to the client



# ROS Model



# Basic Commands

- rosnode
- rostopic
- rosservice
- roslaunch
- rosrun
  - requires *roscore*

To find out more, type `[command] -h` in the terminal  
e.g. `rostopic -h`

# ROS Tools

- Rviz - 3D Visualization tool for ROS
  - helps to visualize the state of the robot
  - `roslaunch rviz rviz --help`

```
/home/txp/catkin_ws/src/limo_ros/limo_description/launch/display_models.launch http://localhost:11311 101x55
txp@txp-450-231d:~$ roslaunch limo_description display_models.launch
... logging to /home/txp/.ros/log/5d9cc140-d0fe-11ec-86a6-48e2446e5af3
roslaunch-txp-450-231d-32121.1
og
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://txp-450-231d:40949/

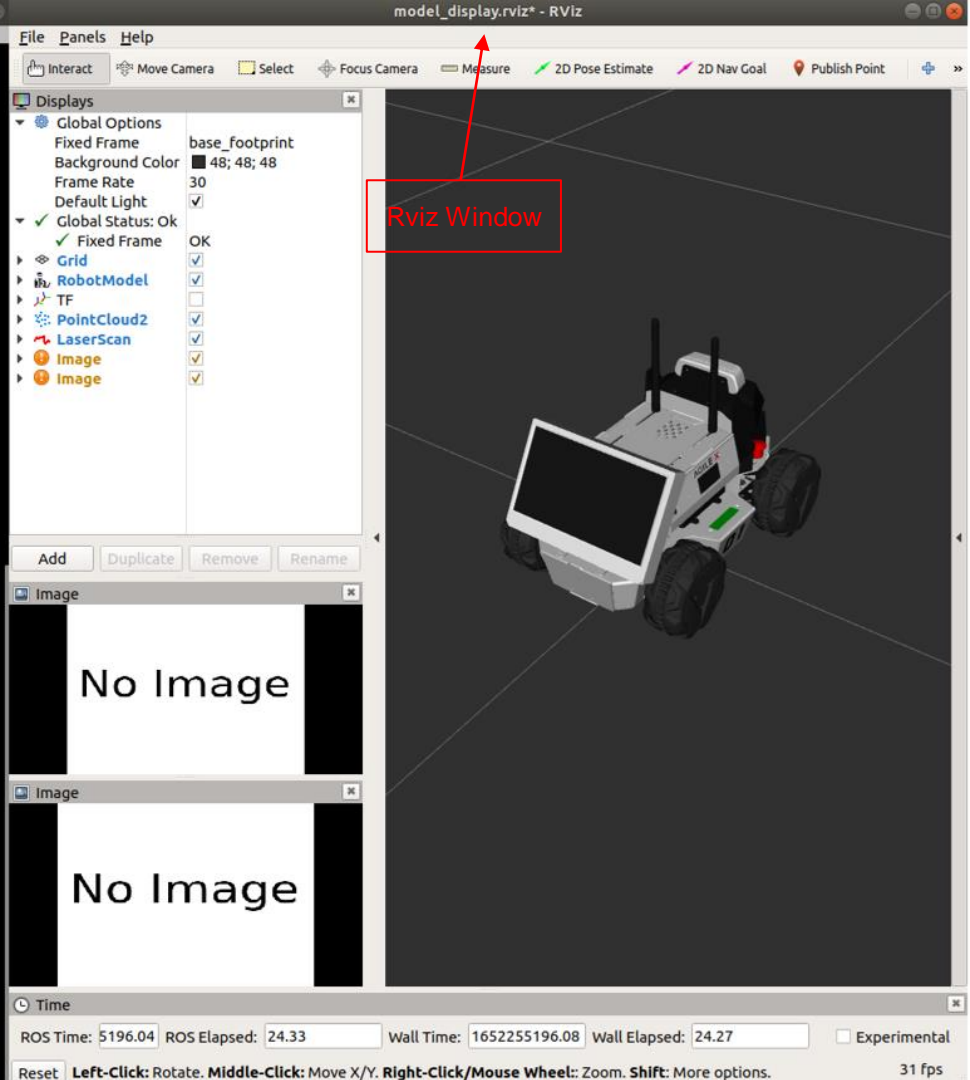
SUMMARY
=====
PARAMETERS
* /robot_description: <?xml version="1...
* /roscpp_core: melodic
* /rosversion: 1.14.13

NODES
/
  joint_state_publisher (joint_state_publisher_gui/joint_state_publisher_gui)
  robot_state_publisher (robot_state_publisher/robot_state_publisher)
  rviz (rviz/rviz)

auto-starting new master
process[master]: started with pid [32134]
ROS_MASTER_URI=http://localhost:11311

setting /run_id to 5d9cc140-d0fe-11ec-86a6-48e2446e5af3
process[rosout-1]: started with pid [32145]
started core service [/rosout]
process[joint_state_publisher-2]: started with pid [32148]
process[robot_state_publisher-3]: started with pid [32153]
process[rviz-4]: started with pid [32154]
```

roslaunch



Rviz Window

ROS Time: 5196.04 ROS Elapsed: 24.33 Wall Time: 1652255196.08 Wall Elapsed: 24.27 Experimental

Reset Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click/Mouse Wheel: Zoom. Shift: More options. 31 fps

# ROS Tools

- rqt - Simplifying tasks and creating a centralized location for ROS tools in the form of plugins
  - Example of plugins
    - rqt\_graph
    - rqt\_topic



```
/home/txp/catkin_ws/src/limo_ros/limo_description/launch/display_models.launch http://localhost:11311
started roslaunch server http://txp-450-231d:45085/

SUMMARY
=====

PARAMETERS
* /robot_description: <?xml version="1...
* /roslistro: melodic
* /rosversion: 1.14.13

NODES
/
  joint_state_publisher (joint_state_publisher_gui/joint_state_publisher_gui)
  robot_state_publisher (robot_state_publisher/robot_state_publisher)
  rviz (rviz/rviz)

auto-starting new master
process[master]: started with pid [3076]
ROS_MASTER_URI=http://localhost:11311

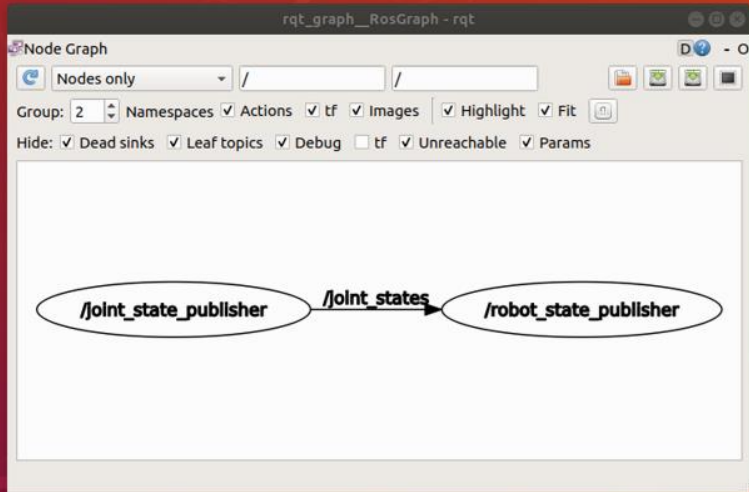
setting /run_id to ef96efc6-d0fe-11ec-86a6-48e2446e5af3
process[rosout-1]: started with pid [3087]
started core service [/rosout]
process[joint_state_publisher-2]: started with pid [3090]
process[robot_state_publisher-3]: started with pid [3095]
process[rviz-4]: started with pid [3096]
```

```
txp@txp-450-231d: ~ 49x27
txp@txp-450-231d:~$ roslaunch rqt_topic rqt_topic

txp@txp-450-231d: ~ 50x27
txp@txp-450-231d:~$ roslaunch rqt_graph rqt_graph
```

rqt\_topic\_\_TopicPlugin - rqt

Topic	Type	Bandwidth	Hz
<input type="checkbox"/> /clicked_point	geometry_msgs/PointStamped		
<input type="checkbox"/> /initialpose	geometry_msgs/PoseWithCovarianceStamped		
<input type="checkbox"/> /joint_states	sensor_msgs/JointState		
<input type="checkbox"/> /move_base_simple/goal	geometry_msgs/PoseStamped		
<input type="checkbox"/> /rosout	rosgraph_msgs/Log		
<input type="checkbox"/> /rosout_agg	rosgraph_msgs/Log		
<input type="checkbox"/> /tf	tf2_msgs/TFMessage		
<input type="checkbox"/> /tf_static	tf2_msgs/TFMessage		



END