ROS: Robot Operating System

EEL 4930/5934: Autonomous Robots

Spring 2023

Md Jahidul Islam

Lecture 2

ECE | Florida
Electrical & Computer Engineering

UF | UNIVERSITY of FLORIDA

ROS: Robot Operating System

⇒ A middleware "OS" for robotics

- Open source *software packages*
  - Components + Tools + Interfaces
- For general-purpose *robot programming + hw/sw interfacing*
  - Actuators: things that move
  - Sensors: things that read the world
  - Control system: robots brain (AI functions!)
- Works best with linux distributions
- Visit ros.org for an introduction

ROS.org

ROS: Getting Started

⇒ **Install ROS *melodic* or *noetic* (ROS 1)**

- Preferred: Linux laptops or Raspberry PI or Jetson Nanos
- Follow the instructions:
  - Getting started: https://www.ros.org/blog/getting-started/
  - Installation: https://wiki.ros.org/ROS/Installation
- Make sure to install the correct distribution for your platform

⇒ **ROS2 documentation:** https://docs.ros.org/

ROS.org

⇒ **Learn basic ROS functionalities**

- ROS Noetic tutorials by Robotics Back-End
- ROS Noetic tutorials by Emil Vidmark
- ROS2 Humble tutorials by Robotics Back-End
- Or browse any other resources!

Lecture Outline

⇒ ROS backgrounds

⇒ Installation (Noetic / Melodic)

- ROS nodes, services, topics, packages
- ROS topics (how to subscribe and how to publish)

⇒ **Bagging**: Saving and playing data

- Managing topics and data formats

⇒ **RViz:** ROS visualizer

- Simulator packages and interfaces

⇒ Working in **Catkin workspaces**

- How to create, build, and run.
- Examples: listener/talker, turtlesim
- Running ROS packages in command line
  - Using **rosrun**
  - Using **roslaunch**

⇒ ***Case study and HH1***

- *Capturing webcam video with ros node*
- *Draw face bounding box*
- *Publish image topic*
- *Use image_view or RViz for visualization*
- *Use rosrun/roslaunch*

ROS: Structure

ROS

https://www.ros.org/

= plumbing + tools + capabilities + community

Life Before ROS

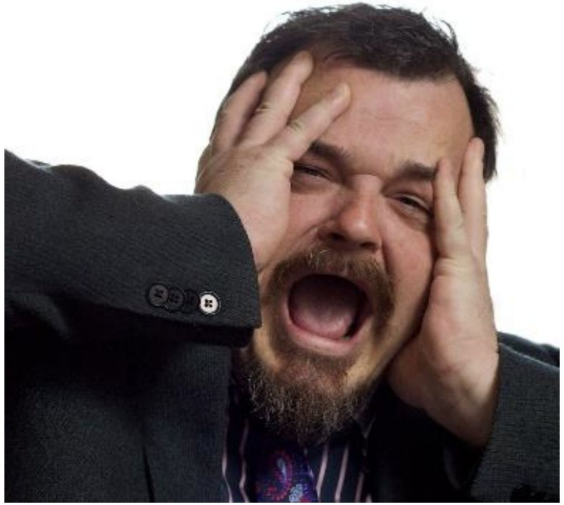
⇒ Lack of standards

⇒ Little code reusability

- Keeping reinventing (or rewriting) device drivers
- Inter-process communication protocols
- Standard algorithms

⇒ New robot in the lab (or in the factory)

- Start re-coding (mostly) from scratch



ROS: History

⇒ Originated by a grad student at **Stanford AI Lab in 2007**.

⇒ Taken up and developed by **Willow Garage**

- A now defunct, but influential, robotics start-up
- Probably the driving influence behind ROS adoption

⇒ 2013: supported by the Open Source Robotics Foundation (OSRF)

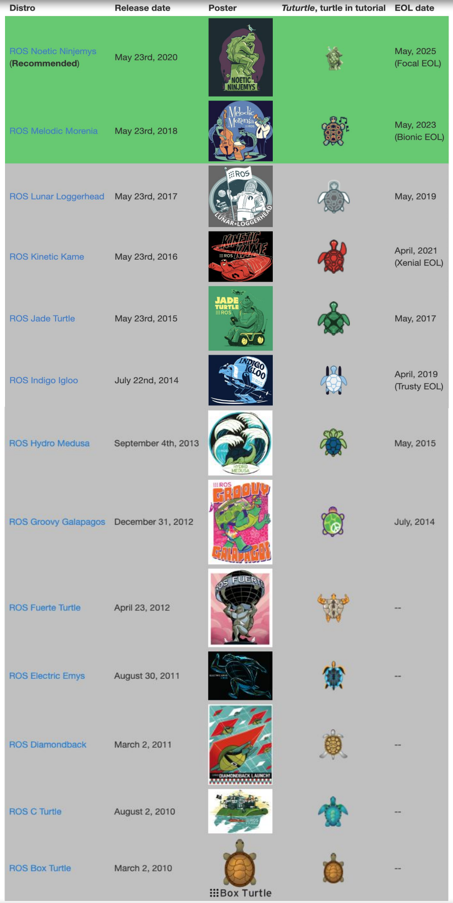
- https://www.openrobotics.org/
- Some Caltech Alums work for/with the foundation

⇒ A series of "releases" define different generations of ROS

⇒ *Read more details here: https://www.theconstructsim.com/history-ros/*

ROS: Distributions

Green: supported release
Grey: unsupported release (End of Life)

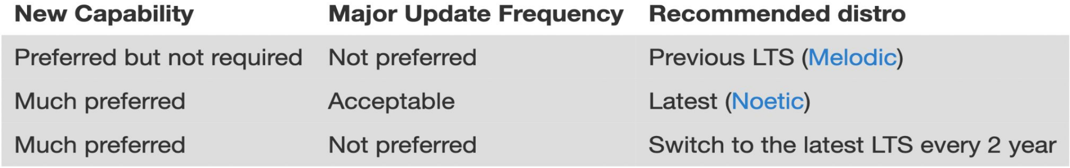


Distro	Release date	Poster	*Tuturtle*, turtle in tutorial	EOL date
ROS Noetic Ninjemys (**Recommended**)	May 23rd, 2020			May, 2025 (Focal EOL)
ROS Melodic Morenia	May 23rd, 2018			May, 2023 (Bionic EOL)
ROS Lunar Loggerhead	May 23rd, 2017			May, 2019
ROS Kinetic Kame	May 23rd, 2016			April, 2021 (Xenial EOL)
ROS Jade Turtle	May 23rd, 2015			May, 2017
ROS Indigo Igloo	July 22nd, 2014			April, 2019 (Trusty EOL)
ROS Hydro Medusa	September 4th, 2013			May, 2015
ROS Groovy Galapagos	December 31, 2012			July, 2014
ROS Fuerte Turtle	April 23, 2012			--
ROS Electric Emys	August 30, 2011			--
ROS Diamondback	March 2, 2011			--
ROS C Turtle	August 2, 2010			--
ROS Box Turtle	March 2, 2010			--

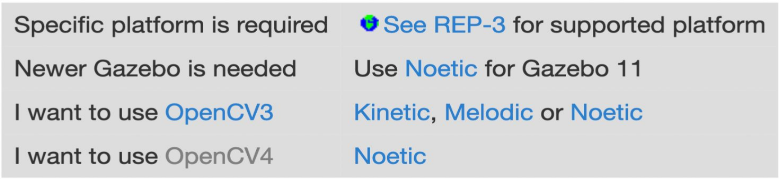
⇒ A versioned set of ROS Packages:

- Like a Linux distribution
- Provide a relatively stable codebase for development
- Primarily for core ROS components
  - User contributed packages must make their own updates

⇒ Which distribution to use:



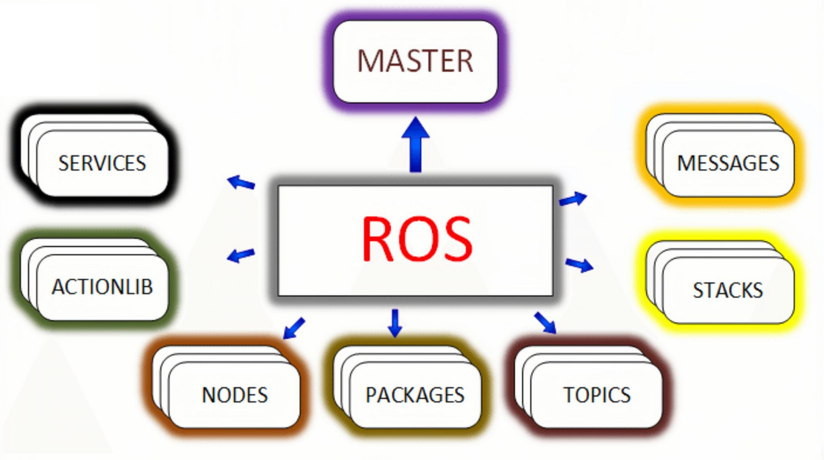
New Capability	Major Update Frequency	Recommended distro
Preferred but not required	Not preferred	Previous LTS (Melodic)
Much preferred	Acceptable	Latest (Noetic)
Much preferred	Not preferred	Switch to the latest LTS every 2 year

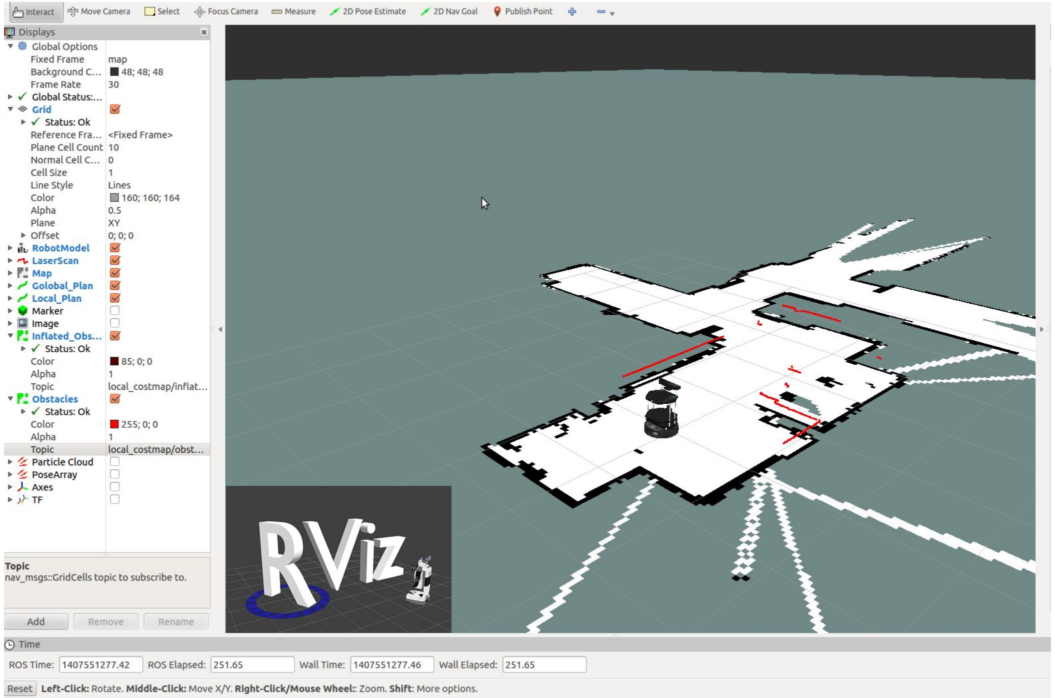


Specific platform is required	See REP-3 for supported platform
Newer Gazebo is needed	Use Noetic for Gazebo 11
I want to use OpenCV3	Kinetic, Melodic or Noetic
I want to use OpenCV4	Noetic

- Noeitc Ninjemys is the final release of ROS 1 by Open Robotics
- Future ROS releases will all be based on ROS 2
  (*visit index.ros.org Releases page*)

Tools To Know For A Roboticist





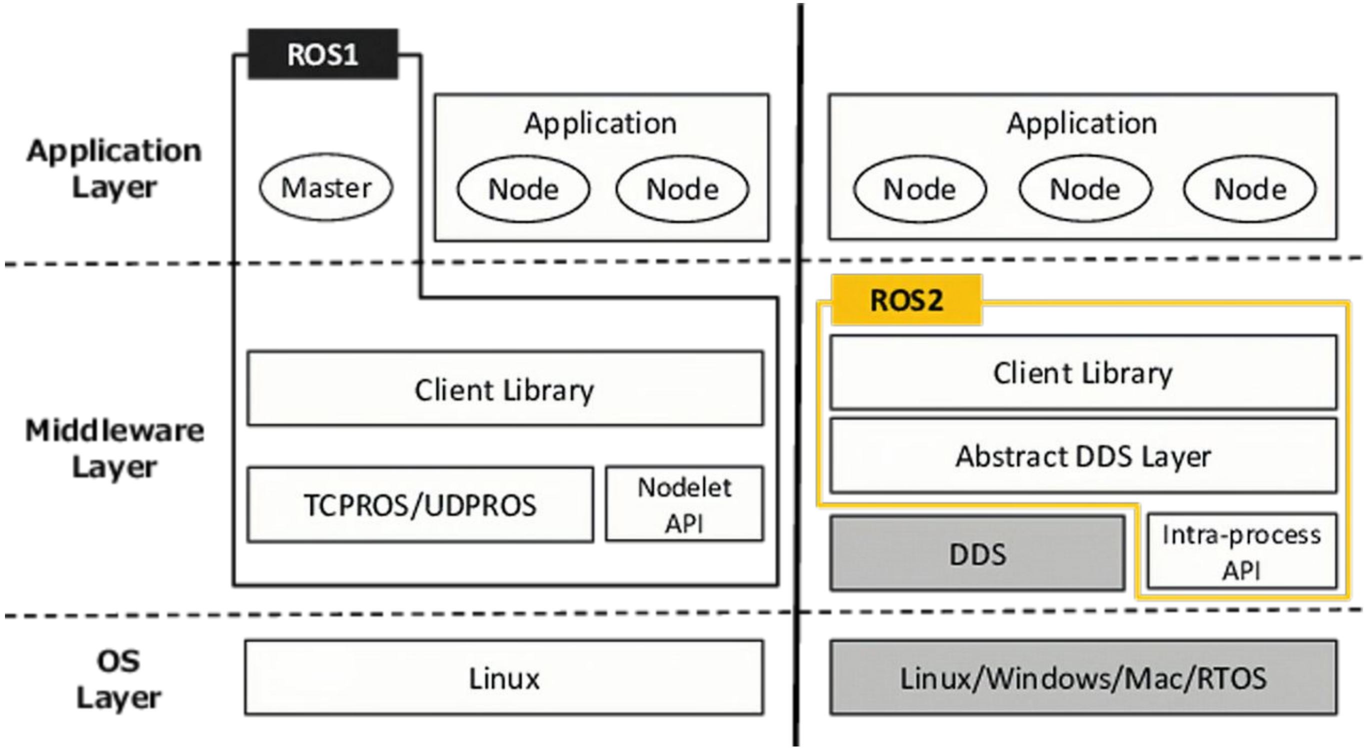
Global Options	
Fixed Frame	map
Background C...	■ 48; 48; 48
Frame Rate	30
✓ Global Status:...	
Grid	☑
✓ Status: Ok	
Reference Fra...	<Fixed Frame>
Plane Cell Count	10
Normal Cell C...	0
Cell Size	1
Line Style	Lines
Color	■ 160; 160; 164
Alpha	0.5
Plane	XY
Offset	0; 0; 0
RobotModel	☑
LaserScan	☑
Map	☑
Golobal_Plan	☑
Local_Plan	☑
Marker	□
Image	□
Inflated_Obs...	☑
✓ Status: Ok	
Color	■ 85; 0; 0
Alpha	1
Topic	local_costmap/inflat...
Obstacles	☑
✓ Status: Ok	
Color	🟥 255; 0; 0
Alpha	1
Topic	local_costmap/obst...
Particle Cloud	□
PoseArray	□
Axes	□
TF	□

Topic
nav_msgs::GridCells topic to subscribe to.

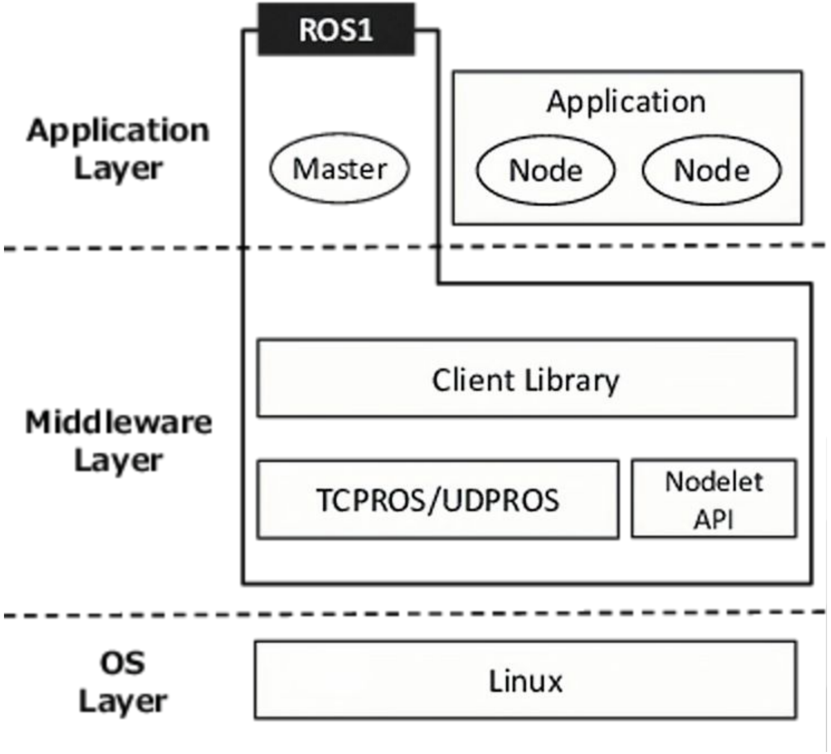
ROS Time: 1407551277.42 ROS Elapsed: 251.65 Wall Time: 1407551277.46 Wall Elapsed: 251.65

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

ROS: Architecture



ROS1: Architecture



⇒ **Low-level device abstraction**

- Joystick
- GPS
- Camera
- Controllers
- Laser Scanners
- …

⇒ **Application building blocks**

- Coordinate system transforms
- Visualization tools
- Debugging tools
- Robust navigation stack (SLAM)
- Arm path planning
- Object recognition
- ...

ROS: Philosophy

- **Peer to Peer**
  - ROS systems consist of many small programs (nodes)
  - Nodes connect to each other and exchange messages
- **Tools-based**
  - There are many small, generic programs that perform tasks
  - Such as visualization, logging, plotting data streams, etc.
- **Multi-lingual**
  - ROS software modules can be written in any language
  - Currently client libraries: C++, Python, LISP, Java, JavaScript, MATLAB, Ruby
- **Thin**
  - The ROS conventions encourage contributors to create stand-alone libraries/packages and then wrap those libraries so they send and receive messages to/from other ROS modules.
- **Free and open source, community-based, repositories**

ROS Installation: Linux

⇒ Check your Ubuntu version first:

Open the terminal and type the following command:

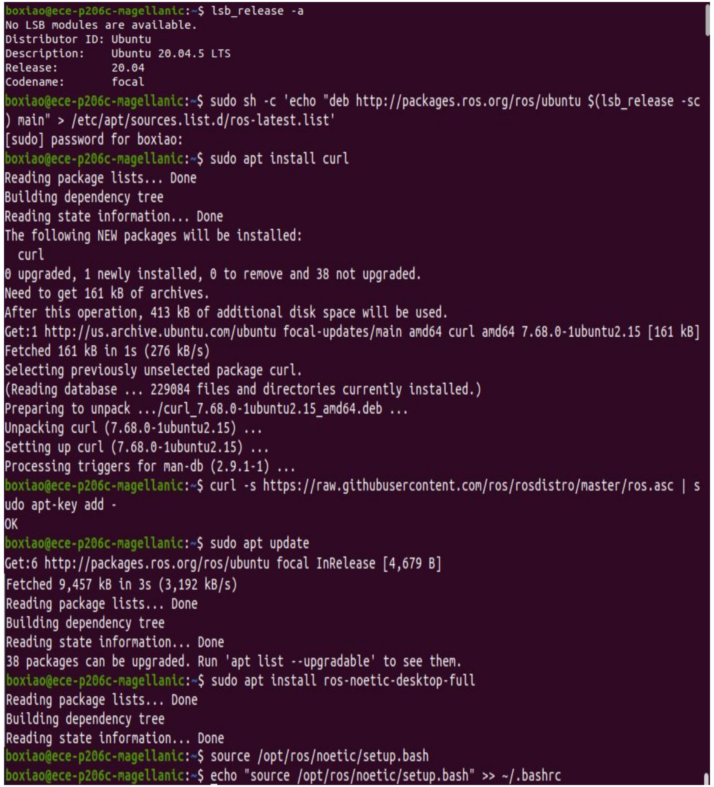
```
$ lsb_release -a
```

⇒ ROS Noetic:

- Primarily targeted at the Ubuntu **20.04** (Focal)
- Follow the installation instruction and reference video to install ROS Noetic step by step

⇒ ROS Melodic:

- Primarily targeted at the Ubuntu **18.04** (Bionic)
- Follow the installation instruction to install ROS Melodic step by step

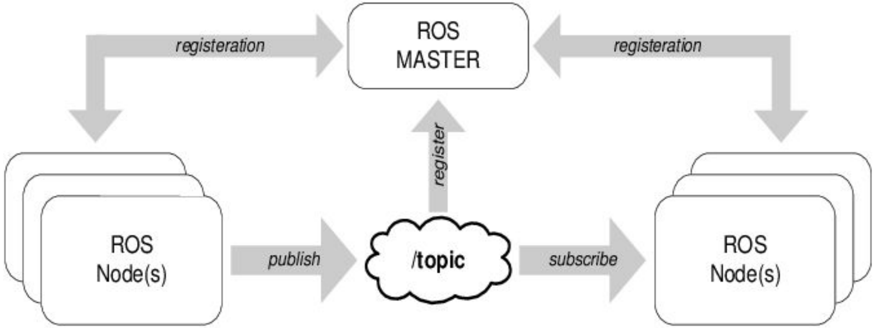


```
boxiao@ece-p206c-magellanic:~$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 20.04.5 LTS
Release:        20.04
Codename:       focal
boxiao@ece-p206c-magellanic:~$ sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc
) main" > /etc/apt/sources.list.d/ros-latest.list'
[sudo] password for boxiao:
boxiao@ece-p206c-magellanic:~$ sudo apt install curl
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  curl
0 upgraded, 1 newly installed, 0 to remove and 38 not upgraded.
Need to get 161 kB of archives.
After this operation, 413 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 curl amd64 7.68.0-1ubuntu2.15 [161 kB]
Fetched 161 kB in 1s (276 kB/s)
Selecting previously unselected package curl.
(Reading database ... 229084 files and directories currently installed.)
Preparing to unpack .../curl_7.68.0-1ubuntu2.15_amd64.deb ...
Unpacking curl (7.68.0-1ubuntu2.15) ...
Setting up curl (7.68.0-1ubuntu2.15) ...
Processing triggers for man-db (2.9.1-1) ...
boxiao@ece-p206c-magellanic:~$ curl -s https://raw.githubusercontent.com/ros/rosdistro/master/ros.asc | s
udo apt-key add -
OK
boxiao@ece-p206c-magellanic:~$ sudo apt update
Get:6 http://packages.ros.org/ros/ubuntu focal InRelease [4,679 B]
Fetched 9,457 kB in 3s (3,192 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
38 packages can be upgraded. Run 'apt list --upgradable' to see them.
boxiao@ece-p206c-magellanic:~$ sudo apt install ros-noetic-desktop-full
Reading package lists... Done
Building dependency tree
Reading state information... Done
boxiao@ece-p206c-magellanic:~$ source /opt/ros/noetic/setup.bash
boxiao@ece-p206c-magellanic:~$ echo "source /opt/ros/noetic/setup.bash" >> ~/.bashrc
```

Example of installing Noetic on ubuntu 20.04

ROS Nodes

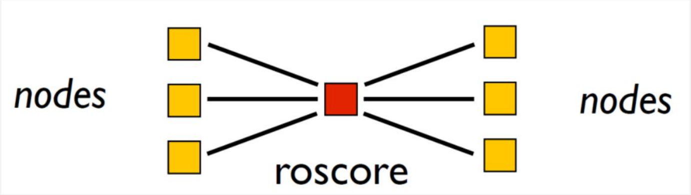
- Single purpose, executable program
  - Can contain many functions, can call other nodes
  - Can subscribe and/or publish topics
- Nodes are assembled into a graph (via communication links)
  - Communication via topics or with a service or with a parameter server
- **Example:** sensor or actuator driver, control loop, motion planning module
- **Programming:** Nodes are developed with the use of a ROS client library
  - *roscpp* – **C++** programs *rospy* – **python** programs



ROS Master

⇒ **Master:** Matchmaker between nodes

- Nodes make be on different cores, different computers, different robots, even different networks.
- This should be transparent to each node’s code
- The “master” service runs on one machine
  - It provides name registration & lookup of nodes and services
- *roscore* starts the master server, parameter server, and logging processes (if any)

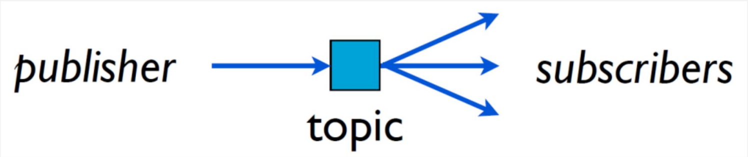


- Every node connects to the master at start-up to register details of the message streams that it publishes
- Also determine its connectivity with the rest of the computation graph via its subscriptions

ROS Topics

⇒ **Topic:** A name for a data stream (TCP or UDP)

- A message bus over which nodes exchange messages
- Example: *lidar* can be the topic that a robot's on-board LiDAR uses to communicate its sensor data
  - The data could be raw, or it could be preprocessed by the lidar sensor node
  - It can send data once, or repeatedly
- Topics are best for unidirectional, streaming communication.
- A request/response model is handled by a service. Fixed data is handled by a parameter server.
- Topic statistics: age of data, traffic volume, # dropped messages



- **Publishing topics:** 1-to-N communication model
- **Subscribing to topics:**
  - Ros Node receives access to the data (bus) published under that topic name

Example: Listener / Talker

Open four terminals, run the following commands in order:

```
Terminal_1:$  roscore
```

# roscore start ROS and create the Master so that nodes can communicate

```
Terminal_2:$  rosrun rospy_tutorials talker
```

# The rosrun command takes the arguments [package name] [node name]

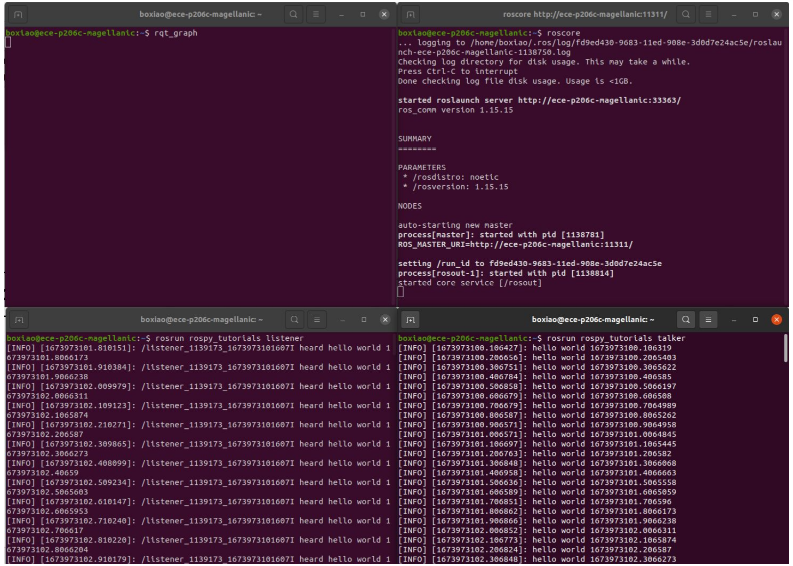
# The "talker" node will broadcast a message on topic "chatter"

```
Terminal_3:$  rosrun rospy_tutorials listener
```

# The "listener" node will receive and print that message

```
Terminal_4:$  rqt_graph
```

# rqt_graph provides a GUI plugin for visualizing the ROS computation graph


boxiao@ece-p206c-magellanic: ~
roscore http://ece-p206c-magellanic:11311/
boxiao@ece-p206c-magellanic: ~
boxiao@ece-p206c-magellanic: ~

```
boxiao@ece-p206c-magellanic:~$ rqt_graph
```

```
boxiao@ece-p206c-magellanic:~$ roscore
... logging to /home/boxiao/.ros/log/fd9ed430-9683-11ed-908e-3d0d7e24ac5e/roslau
nch-ece-p206c-magellanic-1138750.log
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://ece-p206c-magellanic:33363/
ros_comm version 1.15.15


SUMMARY
========

PARAMETERS
 * /rosdistro: noetic
 * /rosversion: 1.15.15

NODES

auto-starting new master
process[master]: started with pid [1138781]
ROS_MASTER_URI=http://ece-p206c-magellanic:11311/

setting /run_id to fd9ed430-9683-11ed-908e-3d0d7e24ac5e
process[rosout-1]: started with pid [1138814]
started core service [/rosout]
```

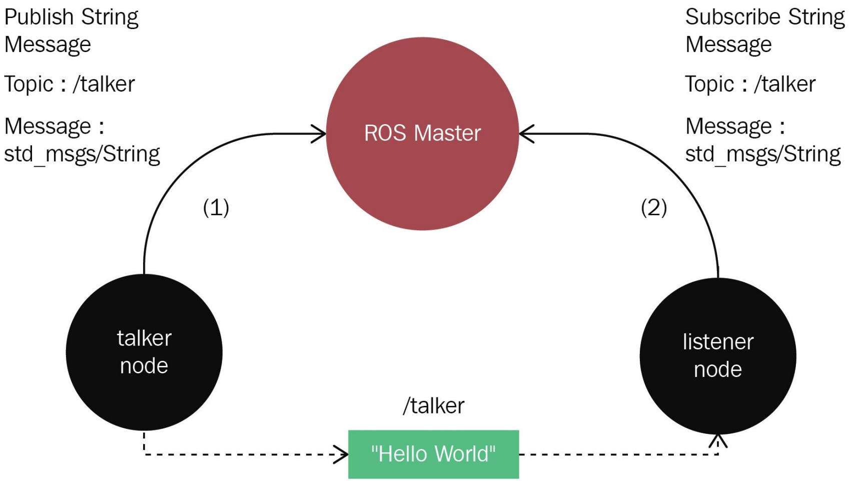
```
boxiao@ece-p206c-magellanic:~$ rosrun rospy_tutorials listener
[INFO] [1673973101.810151]: /listener_1139173_1673973101607I heard hello world 1
673973101.8066173
[INFO] [1673973101.910384]: /listener_1139173_1673973101607I heard hello world 1
673973101.9066238
[INFO] [1673973102.009979]: /listener_1139173_1673973101607I heard hello world 1
673973102.0066311
[INFO] [1673973102.109123]: /listener_1139173_1673973101607I heard hello world 1
673973102.1065874
[INFO] [1673973102.210271]: /listener_1139173_1673973101607I heard hello world 1
673973102.206587
[INFO] [1673973102.309865]: /listener_1139173_1673973101607I heard hello world 1
673973102.3066273
[INFO] [1673973102.408099]: /listener_1139173_1673973101607I heard hello world 1
673973102.40659
[INFO] [1673973102.509234]: /listener_1139173_1673973101607I heard hello world 1
673973102.5065603
[INFO] [1673973102.610147]: /listener_1139173_1673973101607I heard hello world 1
673973102.6065953
[INFO] [1673973102.710240]: /listener_1139173_1673973101607I heard hello world 1
673973102.706617
[INFO] [1673973102.810220]: /listener_1139173_1673973101607I heard hello world 1
673973102.8066204
[INFO] [1673973102.910179]: /listener_1139173_1673973101607I heard hello world 1
```

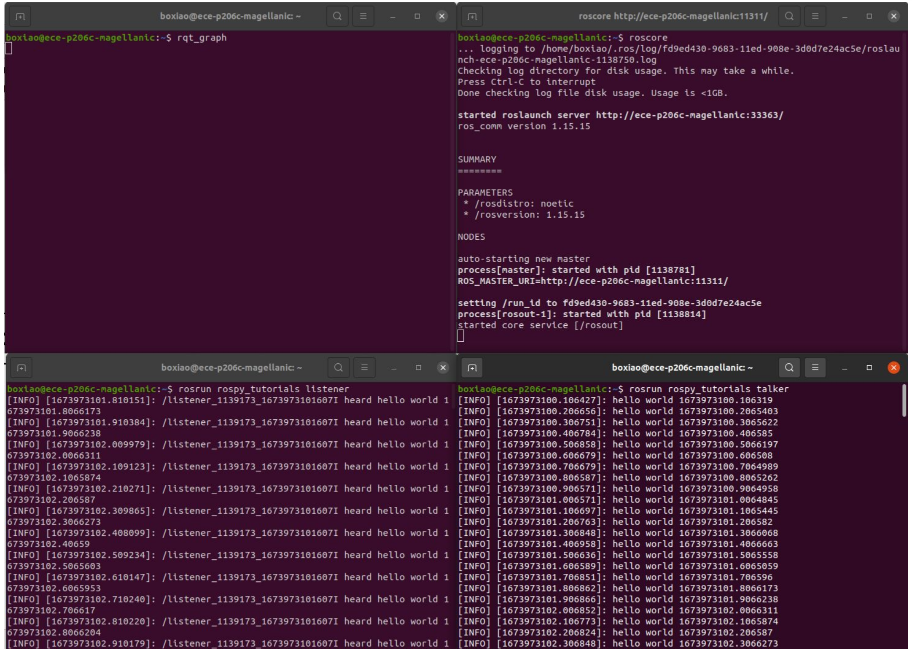
```
boxiao@ece-p206c-magellanic:~$ rosrun rospy_tutorials talker
[INFO] [1673973100.106427]: hello world 1673973100.106319
[INFO] [1673973100.206656]: hello world 1673973100.2065403
[INFO] [1673973100.306751]: hello world 1673973100.3065622
[INFO] [1673973100.406784]: hello world 1673973100.406585
[INFO] [1673973100.506858]: hello world 1673973100.5066197
[INFO] [1673973100.606679]: hello world 1673973100.606508
[INFO] [1673973100.706679]: hello world 1673973100.7064989
[INFO] [1673973100.806587]: hello world 1673973100.8065262
[INFO] [1673973100.906571]: hello world 1673973100.9064958
[INFO] [1673973101.006571]: hello world 1673973101.0064845
[INFO] [1673973101.106697]: hello world 1673973101.1065445
[INFO] [1673973101.206763]: hello world 1673973101.206582
[INFO] [1673973101.306848]: hello world 1673973101.3066068
[INFO] [1673973101.406958]: hello world 1673973101.4066663
[INFO] [1673973101.506636]: hello world 1673973101.5065558
[INFO] [1673973101.606589]: hello world 1673973101.6065059
[INFO] [1673973101.706851]: hello world 1673973101.706596
[INFO] [1673973101.806862]: hello world 1673973101.8066173
[INFO] [1673973101.906866]: hello world 1673973101.9066238
[INFO] [1673973102.006852]: hello world 1673973102.0066311
[INFO] [1673973102.106773]: hello world 1673973102.1065874
[INFO] [1673973102.206824]: hello world 1673973102.206587
[INFO] [1673973102.306848]: hello world 1673973102.3066273
```

Example: Listener / Talker

The application can be divided into two nodes:

- Talker node: responsible of creating the message “Hello World”
- Listener node: subscribes to the *talker* topic and thus receive the messages sent it




boxiao@ece-p206c-magellanic: ~
roscore http://ece-p206c-magellanic:11311/
boxiao@ece-p206c-magellanic: ~
boxiao@ece-p206c-magellanic: ~

```
boxiao@ece-p206c-magellanic:~$ rqt_graph
```

```
boxiao@ece-p206c-magellanic:~$ roscore
... logging to /home/boxiao/.ros/log/fd9ed430-9683-11ed-908e-3d0d7e24ac5e/roslau
nch-ece-p206c-magellanic-1138750.log
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://ece-p206c-magellanic:33363/
ros_comm version 1.15.15


SUMMARY
========

PARAMETERS
 * /rosdistro: noetic
 * /rosversion: 1.15.15

NODES

auto-starting new master
process[master]: started with pid [1138781]
ROS_MASTER_URI=http://ece-p206c-magellanic:11311/

setting /run_id to fd9ed430-9683-11ed-908e-3d0d7e24ac5e
process[rosout-1]: started with pid [1138814]
started core service [/rosout]
```

```
boxiao@ece-p206c-magellanic:~$ rosrun rospy_tutorials listener
[INFO] [1673973101.810151]: /listener_1139173_1673973101607I heard hello world 1
673973101.8066173
[INFO] [1673973101.910384]: /listener_1139173_1673973101607I heard hello world 1
673973101.9066238
[INFO] [1673973102.009979]: /listener_1139173_1673973101607I heard hello world 1
673973102.0066311
[INFO] [1673973102.109123]: /listener_1139173_1673973101607I heard hello world 1
673973102.1065874
[INFO] [1673973102.210271]: /listener_1139173_1673973101607I heard hello world 1
673973102.206587
[INFO] [1673973102.309865]: /listener_1139173_1673973101607I heard hello world 1
673973102.3066273
[INFO] [1673973102.408099]: /listener_1139173_1673973101607I heard hello world 1
673973102.40659
[INFO] [1673973102.509234]: /listener_1139173_1673973101607I heard hello world 1
673973102.5065603
[INFO] [1673973102.610147]: /listener_1139173_1673973101607I heard hello world 1
673973102.6065953
[INFO] [1673973102.710240]: /listener_1139173_1673973101607I heard hello world 1
673973102.706617
[INFO] [1673973102.810220]: /listener_1139173_1673973101607I heard hello world 1
673973102.8066204
[INFO] [1673973102.910179]: /listener_1139173_1673973101607I heard hello world 1
```

```
boxiao@ece-p206c-magellanic:~$ rosrun rospy_tutorials talker
[INFO] [1673973100.106427]: hello world 1673973100.106319
[INFO] [1673973100.206656]: hello world 1673973100.2065403
[INFO] [1673973100.306751]: hello world 1673973100.3065622
[INFO] [1673973100.406784]: hello world 1673973100.406585
[INFO] [1673973100.506858]: hello world 1673973100.5066197
[INFO] [1673973100.606679]: hello world 1673973100.606508
[INFO] [1673973100.706679]: hello world 1673973100.7064989
[INFO] [1673973100.806587]: hello world 1673973100.8065262
[INFO] [1673973100.906571]: hello world 1673973100.9064958
[INFO] [1673973101.006571]: hello world 1673973101.0064845
[INFO] [1673973101.106697]: hello world 1673973101.1065445
[INFO] [1673973101.206763]: hello world 1673973101.206582
[INFO] [1673973101.306848]: hello world 1673973101.3066068
[INFO] [1673973101.406958]: hello world 1673973101.4066663
[INFO] [1673973101.506636]: hello world 1673973101.5065558
[INFO] [1673973101.606589]: hello world 1673973101.6065059
[INFO] [1673973101.706851]: hello world 1673973101.706596
[INFO] [1673973101.806862]: hello world 1673973101.8066173
[INFO] [1673973101.906866]: hello world 1673973101.9066238
[INFO] [1673973102.006852]: hello world 1673973102.0066311
[INFO] [1673973102.106773]: hello world 1673973102.1065874
[INFO] [1673973102.206824]: hello world 1673973102.206587
[INFO] [1673973102.306848]: hello world 1673973102.3066273
```

Example: Turtlesim

Open four terminals, run the following commands in order:

```
Terminal_1:$ roscore
```

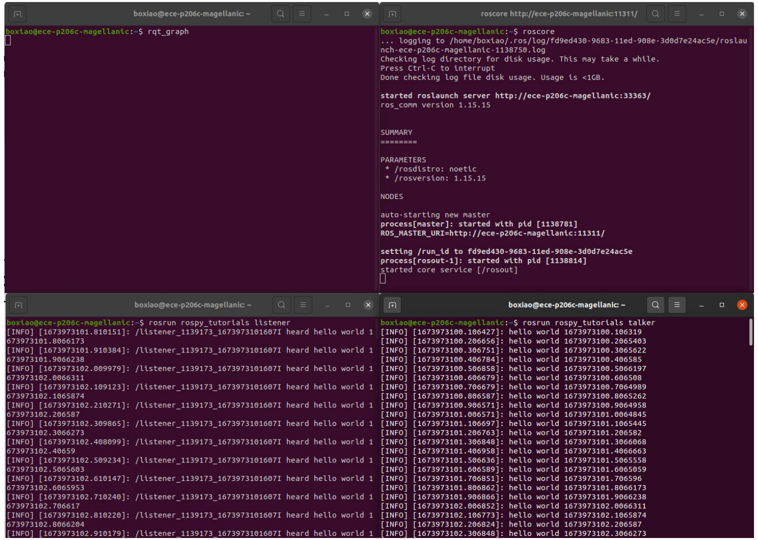
```
Terminal_2:$ rosrun turtlesim turtlesim_node
```

# This node creates the screen image and the turtle

```
Terminal_3:$ rosrun turtlesim turtle_teleop_key
```

# This node allows keyboard control of the turtle

```
Terminal_4:$ rqt_graph
```



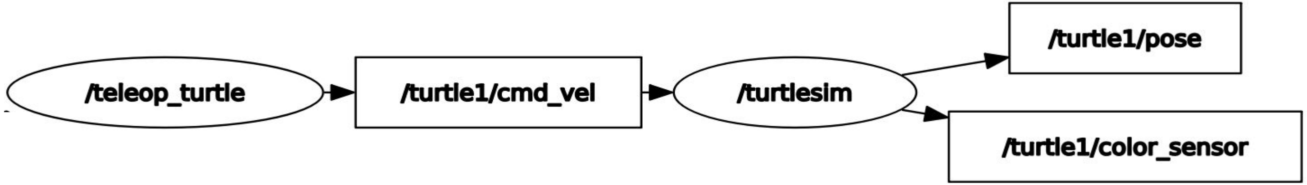
boxiao@ece-p206c-magellanic: ~
roscore http://ece-p206c-magellanic:11311/
boxiao@ece-p206c-magellanic: ~
boxiao@ece-p206c-magellanic: ~

```
boxiao@ece-p206c-magellanic:~$ rqt_graph
```

```
boxiao@ece-p206c-magellanic:~$ roscore
... logging to /home/boxiao/.ros/log/fd9ed430-9683-11ed-908e-3d0d7e24ac5e/roslau
nch-ece-p206c-magellanic-1138750.log
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://ece-p206c-magellanic:33363/
ros_comm version 1.15.15


SUMMARY
========

PARAMETERS
 * /rosdistro: noetic
 * /rosversion: 1.15.15

NODES

auto-starting new master
process[master]: started with pid [1138781]
ROS_MASTER_URI=http://ece-p206c-magellanic:11311/

setting /run_id to fd9ed430-9683-11ed-908e-3d0d7e24ac5e
process[rosout-1]: started with pid [1138814]
started core service [/rosout]
```

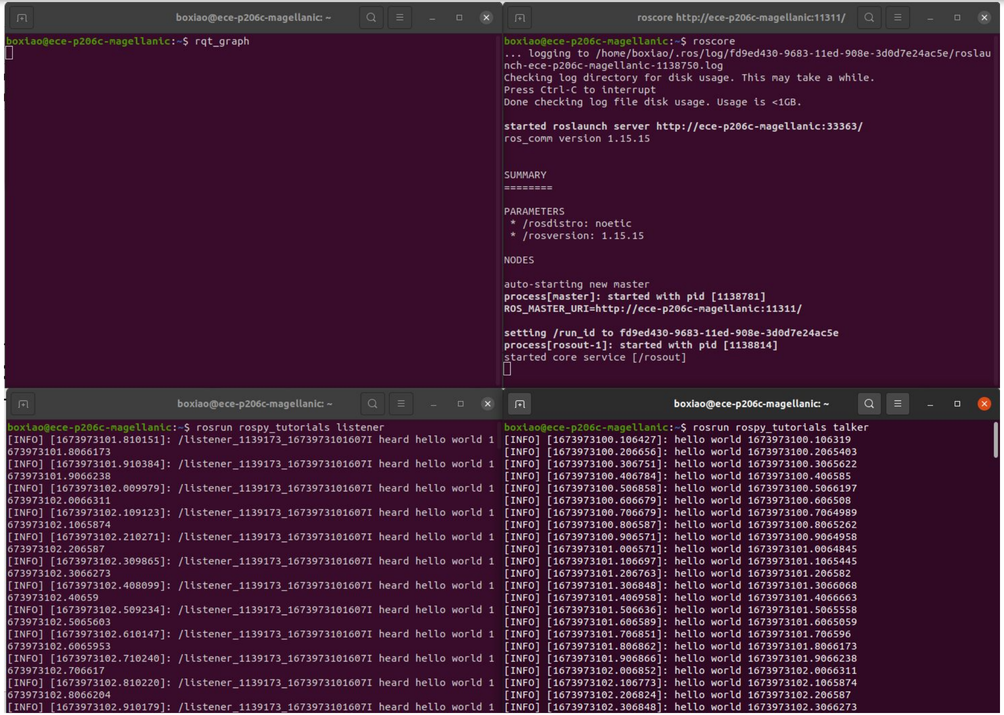
```
boxiao@ece-p206c-magellanic:~$ rosrun rospy_tutorials listener
[INFO] [1673973101.810151]: /listener_1139173_1673973101607I heard hello world 1
673973101.8066173
[INFO] [1673973101.910384]: /listener_1139173_1673973101607I heard hello world 1
673973101.9066238
[INFO] [1673973102.009979]: /listener_1139173_1673973101607I heard hello world 1
673973102.0066311
[INFO] [1673973102.109123]: /listener_1139173_1673973101607I heard hello world 1
673973102.1065874
[INFO] [1673973102.210271]: /listener_1139173_1673973101607I heard hello world 1
673973102.206587
[INFO] [1673973102.309865]: /listener_1139173_1673973101607I heard hello world 1
673973102.3066273
[INFO] [1673973102.408099]: /listener_1139173_1673973101607I heard hello world 1
673973102.40659
[INFO] [1673973102.509234]: /listener_1139173_1673973101607I heard hello world 1
673973102.5065603
[INFO] [1673973102.610147]: /listener_1139173_1673973101607I heard hello world 1
673973102.6065953
[INFO] [1673973102.710240]: /listener_1139173_1673973101607I heard hello world 1
673973102.706617
[INFO] [1673973102.810220]: /listener_1139173_1673973101607I heard hello world 1
673973102.8066204
[INFO] [1673973102.910179]: /listener_1139173_1673973101607I heard hello world 1
```

```
boxiao@ece-p206c-magellanic:~$ rosrun rospy_tutorials talker
[INFO] [1673973100.106427]: hello world 1673973100.106319
[INFO] [1673973100.206656]: hello world 1673973100.2065403
[INFO] [1673973100.306751]: hello world 1673973100.3065622
[INFO] [1673973100.406784]: hello world 1673973100.406585
[INFO] [1673973100.506858]: hello world 1673973100.5066197
[INFO] [1673973100.606679]: hello world 1673973100.606508
[INFO] [1673973100.706679]: hello world 1673973100.7064989
[INFO] [1673973100.806587]: hello world 1673973100.8065262
[INFO] [1673973100.906571]: hello world 1673973100.9064958
[INFO] [1673973101.006571]: hello world 1673973101.0064845
[INFO] [1673973101.106697]: hello world 1673973101.1065445
[INFO] [1673973101.206763]: hello world 1673973101.206582
[INFO] [1673973101.306848]: hello world 1673973101.3066068
[INFO] [1673973101.406958]: hello world 1673973101.4066663
[INFO] [1673973101.506636]: hello world 1673973101.5065558
[INFO] [1673973101.606589]: hello world 1673973101.6065059
[INFO] [1673973101.706851]: hello world 1673973101.706596
[INFO] [1673973101.806862]: hello world 1673973101.8066173
[INFO] [1673973101.906866]: hello world 1673973101.9066238
[INFO] [1673973102.006852]: hello world 1673973102.0066311
[INFO] [1673973102.106773]: hello world 1673973102.1065874
[INFO] [1673973102.206824]: hello world 1673973102.206587
[INFO] [1673973102.306848]: hello world 1673973102.3066273
```



ROS computation graph provided by rqt_graph

Example: Turtlesim

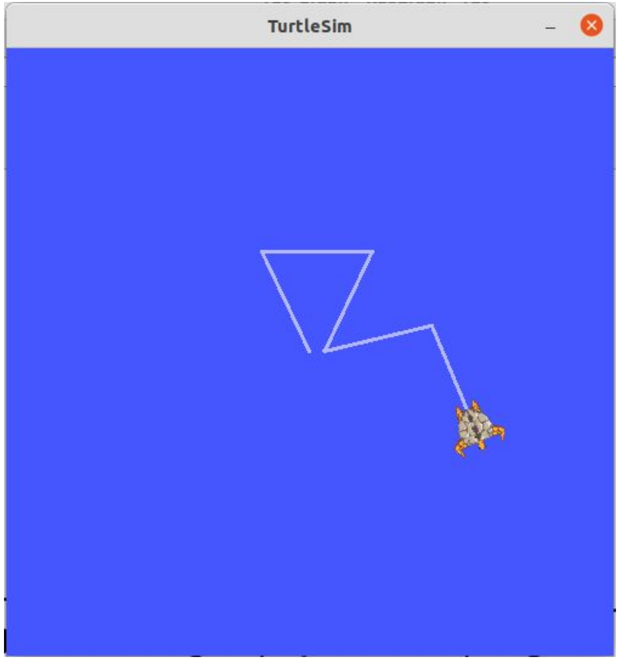

boxiao@ece-p206c-magellanic: ~
roscore http://ece-p206c-magellanic:11311/
boxiao@ece-p206c-magellanic: ~
boxiao@ece-p206c-magellanic: ~

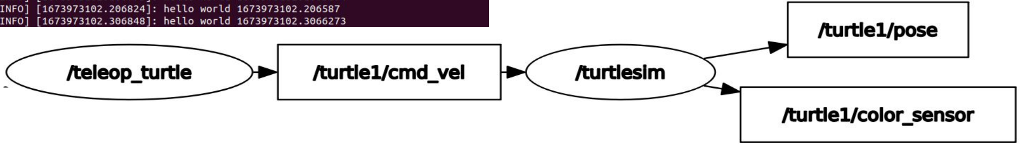
```
boxiao@ece-p206c-magellanic:~$ rqt_graph
```

```
boxiao@ece-p206c-magellanic:~$ roscore
... logging to /home/boxiao/.ros/log/fd9ed430-9683-11ed-908e-3d0d7e24ac5e/roslau
nch-ece-p206c-magellanic-1138750.log
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://ece-p206c-magellanic:33363/
ros_comm version 1.15.15


SUMMARY
========

PARAMETERS
 * /rosdistro: noetic
 * /rosversion: 1.15.15

NODES

auto-starting new master
process[master]: started with pid [1138781]
ROS_MASTER_URI=http://ece-p206c-magellanic:11311/

setting /run_id to fd9ed430-9683-11ed-908e-3d0d7e24ac5e
process[rosout-1]: started with pid [1138814]
started core service [/rosout]
```

```
boxiao@ece-p206c-magellanic:~$ rosrun rospy_tutorials listener
[INFO] [1673973101.810151]: /listener_1139173_1673973101607I heard hello world 1
673973101.8066173
[INFO] [1673973101.910384]: /listener_1139173_1673973101607I heard hello world 1
673973101.9066238
[INFO] [1673973102.009979]: /listener_1139173_1673973101607I heard hello world 1
673973102.0066311
[INFO] [1673973102.109123]: /listener_1139173_1673973101607I heard hello world 1
673973102.1065874
[INFO] [1673973102.210271]: /listener_1139173_1673973101607I heard hello world 1
673973102.206587
[INFO] [1673973102.309865]: /listener_1139173_1673973101607I heard hello world 1
673973102.3066273
[INFO] [1673973102.408099]: /listener_1139173_1673973101607I heard hello world 1
673973102.40659
[INFO] [1673973102.509234]: /listener_1139173_1673973101607I heard hello world 1
673973102.5065603
[INFO] [1673973102.610147]: /listener_1139173_1673973101607I heard hello world 1
673973102.6065953
[INFO] [1673973102.710240]: /listener_1139173_1673973101607I heard hello world 1
673973102.706617
[INFO] [1673973102.810220]: /listener_1139173_1673973101607I heard hello world 1
673973102.8066204
[INFO] [1673973102.910179]: /listener_1139173_1673973101607I heard hello world 1
```

```
boxiao@ece-p206c-magellanic:~$ rosrun rospy_tutorials talker
[INFO] [1673973100.106427]: hello world 1673973100.106319
[INFO] [1673973100.206656]: hello world 1673973100.2065403
[INFO] [1673973100.306751]: hello world 1673973100.3065622
[INFO] [1673973100.406784]: hello world 1673973100.406585
[INFO] [1673973100.506858]: hello world 1673973100.5066197
[INFO] [1673973100.606679]: hello world 1673973100.606508
[INFO] [1673973100.706679]: hello world 1673973100.7064989
[INFO] [1673973100.806587]: hello world 1673973100.8065262
[INFO] [1673973100.906571]: hello world 1673973100.9064958
[INFO] [1673973101.006571]: hello world 1673973101.0064845
[INFO] [1673973101.106697]: hello world 1673973101.1065445
[INFO] [1673973101.206763]: hello world 1673973101.206582
[INFO] [1673973101.306848]: hello world 1673973101.3066068
[INFO] [1673973101.406958]: hello world 1673973101.4066663
[INFO] [1673973101.506636]: hello world 1673973101.5065558
[INFO] [1673973101.606589]: hello world 1673973101.6065059
[INFO] [1673973101.706851]: hello world 1673973101.706596
[INFO] [1673973101.806862]: hello world 1673973101.8066173
[INFO] [1673973101.906866]: hello world 1673973101.9066238
[INFO] [1673973102.006852]: hello world 1673973102.0066311
[INFO] [1673973102.106773]: hello world 1673973102.1065874
[INFO] [1673973102.206824]: hello world 1673973102.206587
[INFO] [1673973102.306848]: hello world 1673973102.3066273
```



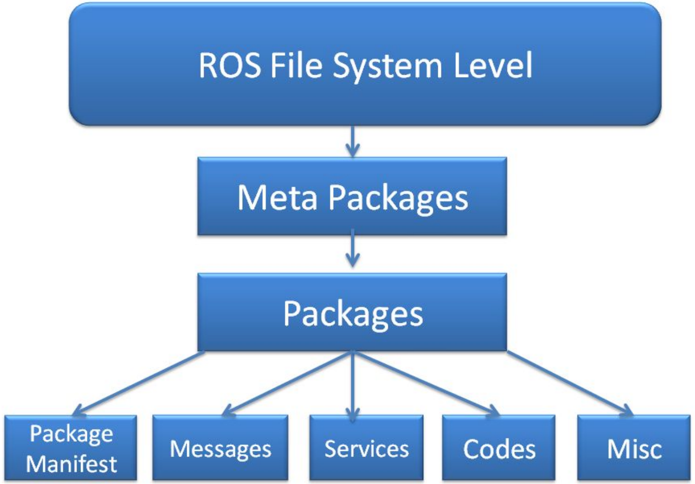


ROS computation graph provided by rqt_graph

ROS Packages

⇒ **Packages:** Basic organizational unit of ROS

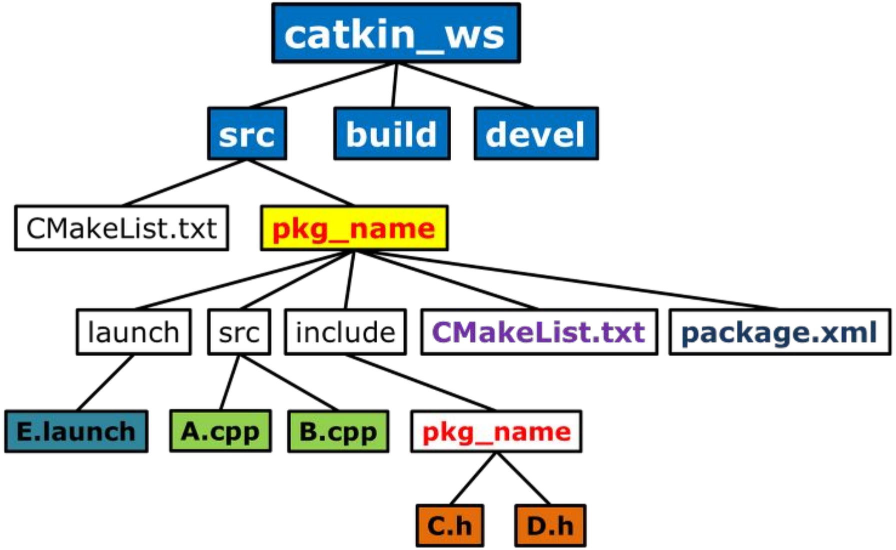
- Contains one or more nodes
- Provides a ROS interface (via messages, services)
- Typically implements a well defined function
  - Example: making a map from sensory data
- Organized into a self-contained directory (specific structure)
  - Contains source code for nodes
  - Message definitions, services, etc



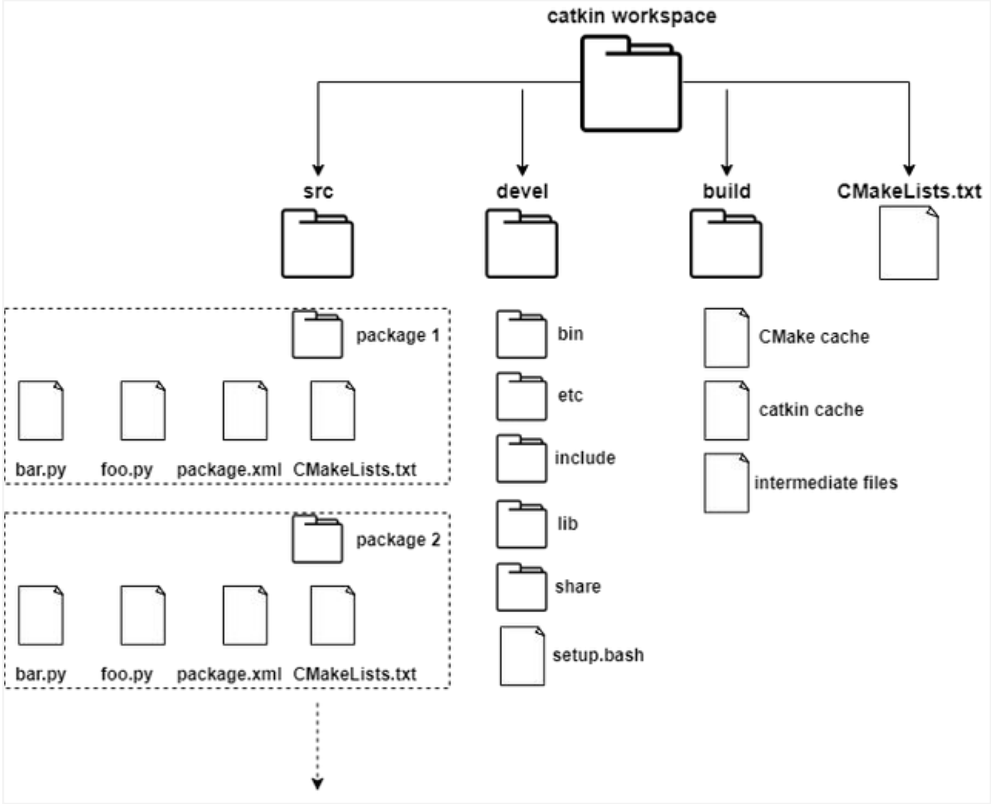
Catkin Workspace

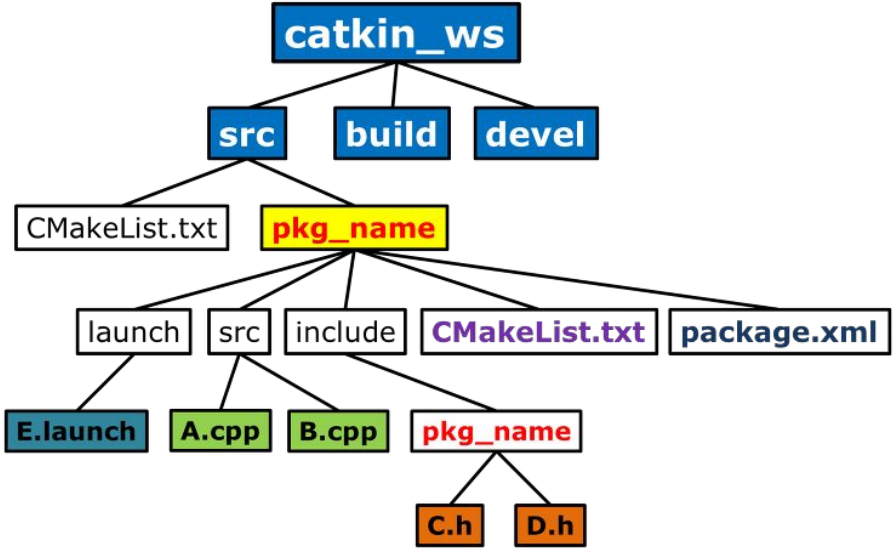
⇒ **Catkin workspace:**

- A set of directories in which a set of related ROS code/packages live
  - Catkin ~ ROS build system
  - CMake + Python scripts
- It's possible to have multiple workspaces
  - Only one-at-a-time can be active
- A ROS package is a directory inside a catkin workspace that has a package.xml file in it



Catkin Workspace

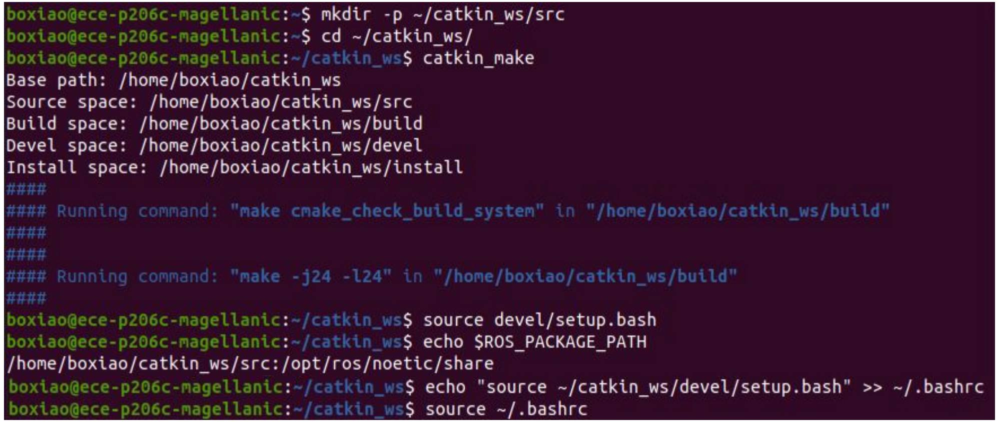




Setup A Catkin Workspace

⇒ Create and setup a Catkin workspace:

- Follow the CreateWorkspace Tutorial and reference video to create and setup Catkin workspace

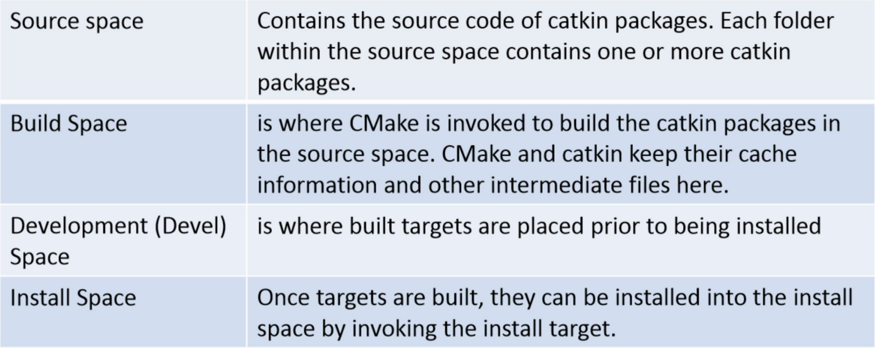


```
boxiao@ece-p206c-magellanic:~$ mkdir -p ~/catkin_ws/src
boxiao@ece-p206c-magellanic:~$ cd ~/catkin_ws/
boxiao@ece-p206c-magellanic:~/catkin_ws$ catkin_make
Base path: /home/boxiao/catkin_ws
Source space: /home/boxiao/catkin_ws/src
Build space: /home/boxiao/catkin_ws/build
Devel space: /home/boxiao/catkin_ws/devel
Install space: /home/boxiao/catkin_ws/install
####
#### Running command: "make cmake_check_build_system" in "/home/boxiao/catkin_ws/build"
####
####
#### Running command: "make -j24 -l24" in "/home/boxiao/catkin_ws/build"
####
boxiao@ece-p206c-magellanic:~/catkin_ws$ source devel/setup.bash
boxiao@ece-p206c-magellanic:~/catkin_ws$ echo $ROS_PACKAGE_PATH
/home/boxiao/catkin_ws/src:/opt/ros/noetic/share
boxiao@ece-p206c-magellanic:~/catkin_ws$ echo "source ~/catkin_ws/devel/setup.bash" >> ~/.bashrc
boxiao@ece-p206c-magellanic:~/catkin_ws$ source ~/.bashrc
```

Example of Catkin workspace setup

⇒ Catkin workspace folders:

- Source space: *workspace_folder/src*
- Build space: *workspace_folder/build*
- Development space: *workspace_folder/devel*
- Install space: *workspace_folder/install*



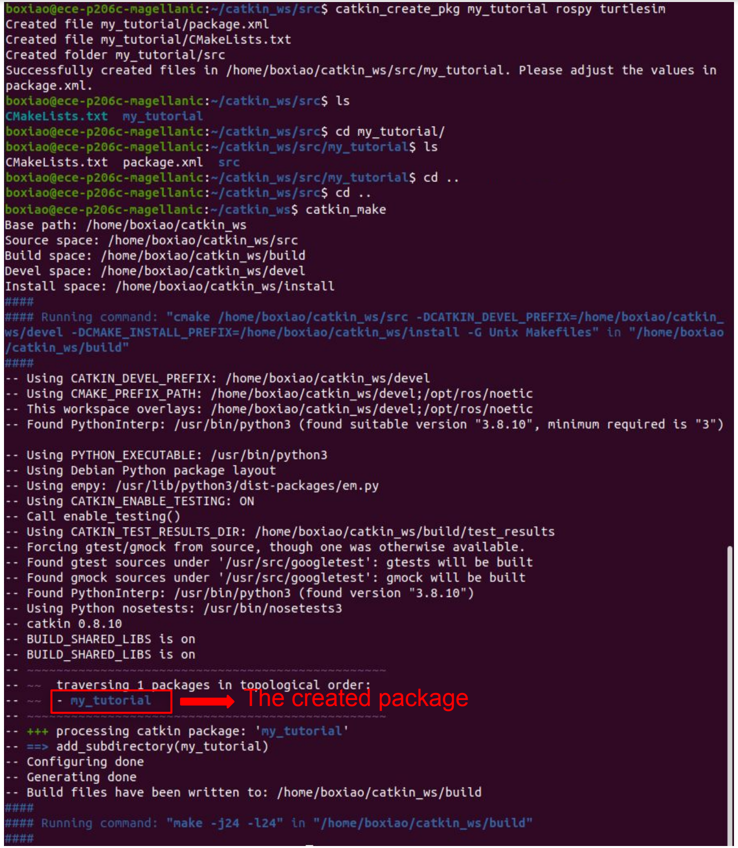
Source space	Contains the source code of catkin packages. Each folder within the source space contains one or more catkin packages.
Build Space	is where CMake is invoked to build the catkin packages in the source space. CMake and catkin keep their cache information and other intermediate files here.
Development (Devel) Space	is where built targets are placed prior to being installed
Install Space	Once targets are built, they can be installed into the install space by invoking the install target.

Create ROS Package In Catkin

⇒ Create a ROS package:

- Follow the CreatingPackage Tutorial and reference video to create ROS package in Catkin workspace
- Useful command:

```
$ catkin_create_pkg <package_name> [depend]
```



```
boxiao@ece-p206c-magellanic:~/catkin_ws/src$ catkin_create_pkg my_tutorial rospy turtlesim
Created file my_tutorial/package.xml
Created file my_tutorial/CMakeLists.txt
Created folder my_tutorial/src
Successfully created files in /home/boxiao/catkin_ws/src/my_tutorial. Please adjust the values in
package.xml.
boxiao@ece-p206c-magellanic:~/catkin_ws/src$ ls
CMakeLists.txt  my_tutorial
boxiao@ece-p206c-magellanic:~/catkin_ws/src$ cd my_tutorial/
boxiao@ece-p206c-magellanic:~/catkin_ws/src/my_tutorial$ ls
CMakeLists.txt  package.xml  src
boxiao@ece-p206c-magellanic:~/catkin_ws/src/my_tutorial$ cd ..
boxiao@ece-p206c-magellanic:~/catkin_ws/src$ cd ..
boxiao@ece-p206c-magellanic:~/catkin_ws$ catkin_make
Base path: /home/boxiao/catkin_ws
Source space: /home/boxiao/catkin_ws/src
Build space: /home/boxiao/catkin_ws/build
Devel space: /home/boxiao/catkin_ws/devel
Install space: /home/boxiao/catkin_ws/install
####
#### Running command: "cmake /home/boxiao/catkin_ws/src -DCATKIN_DEVEL_PREFIX=/home/boxiao/catkin_
ws/devel -DCMAKE_INSTALL_PREFIX=/home/boxiao/catkin_ws/install -G Unix Makefiles" in "/home/boxiao
/catkin_ws/build"
####
-- Using CATKIN_DEVEL_PREFIX: /home/boxiao/catkin_ws/devel
-- Using CMAKE_PREFIX_PATH: /home/boxiao/catkin_ws/devel;/opt/ros/noetic
-- This workspace overlays: /home/boxiao/catkin_ws/devel;/opt/ros/noetic
-- Found PythonInterp: /usr/bin/python3 (found suitable version "3.8.10", minimum required is "3")

-- Using PYTHON_EXECUTABLE: /usr/bin/python3
-- Using Debian Python package layout
-- Using empy: /usr/lib/python3/dist-packages/em.py
-- Using CATKIN_ENABLE_TESTING: ON
-- Call enable_testing()
-- Using CATKIN_TEST_RESULTS_DIR: /home/boxiao/catkin_ws/build/test_results
-- Forcing gtest/gmock from source, though one was otherwise available.
-- Found gtest sources under '/usr/src/googletest': gtests will be built
-- Found gmock sources under '/usr/src/googletest': gmock will be built
-- Found PythonInterp: /usr/bin/python3 (found version "3.8.10")
-- Using Python nosetests: /usr/bin/nosetests3
-- catkin 0.8.10
-- BUILD_SHARED_LIBS is on
-- BUILD_SHARED_LIBS is on
-- ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
-- ~~  traversing 1 packages in topological order:
-- ~~  - my_tutorial
-- ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
-- +++ processing catkin package: 'my_tutorial'
-- ==> add_subdirectory(my_tutorial)
-- Configuring done
-- Generating done
-- Build files have been written to: /home/boxiao/catkin_ws/build
####
#### Running command: "make -j24 -l24" in "/home/boxiao/catkin_ws/build"
####
```

The created package

Example of package creation

Most Useful Commands

```
$ roscore
```

# roscore command start ROS and create the Master so that nodes can communicate

```
$ rosrun <package_name> <node_name>
```

# rosrun command allows you to run an executable in an arbitrary package from anywhere

```
$ roslaunch <package_name> <file.launch>
```

# Many ROS packages come with “launch files”, roslaunch command reads the .launch/XML format

```
$ rqt_graph
```

# rqt_graph command provides a GUI plugin for visualizing the ROS computation graph

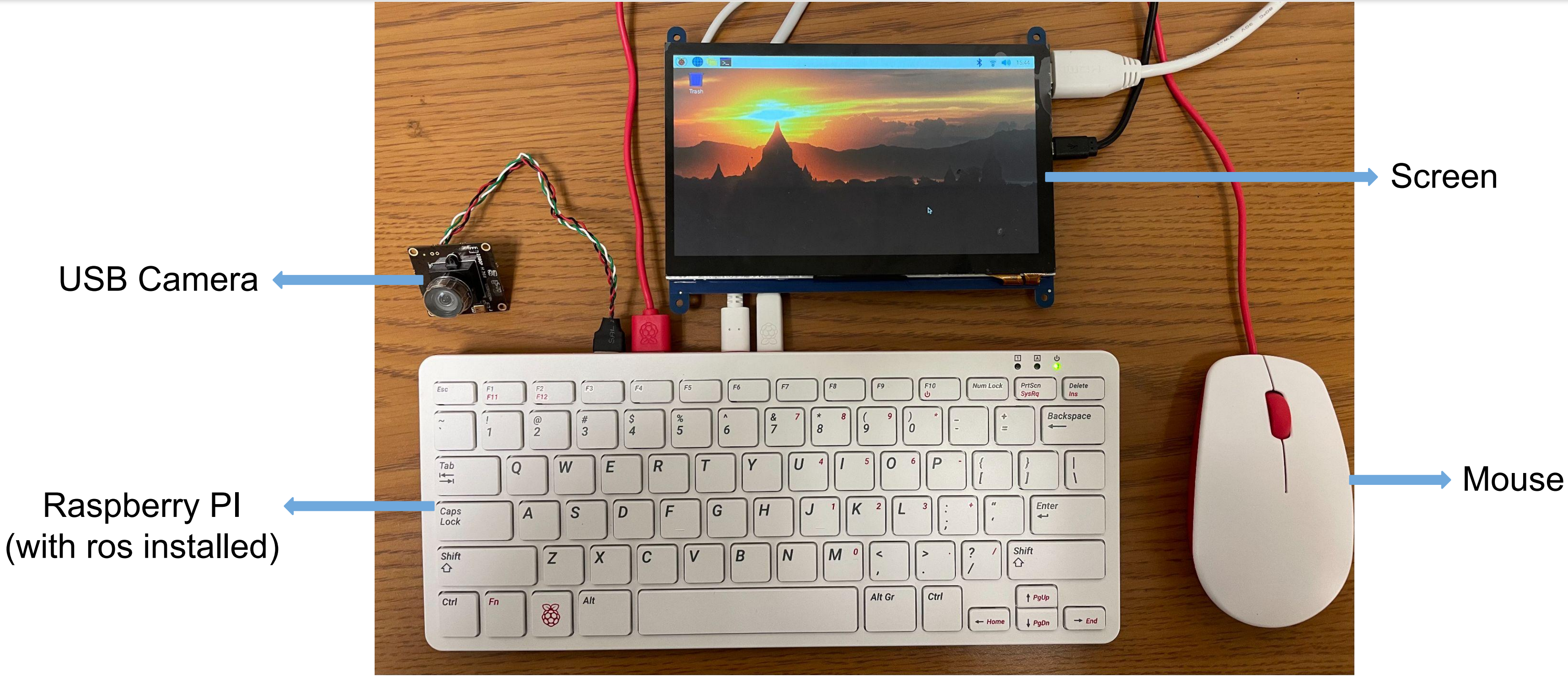
```
$ rosnode info/kill/list/machine/ping/cleanup
```

# rosnode command can display debug information about ROS Nodes, including publications, subscriptions and connections

```
$ rostopic info/list/echo/type/pub/bw/delay/find
```

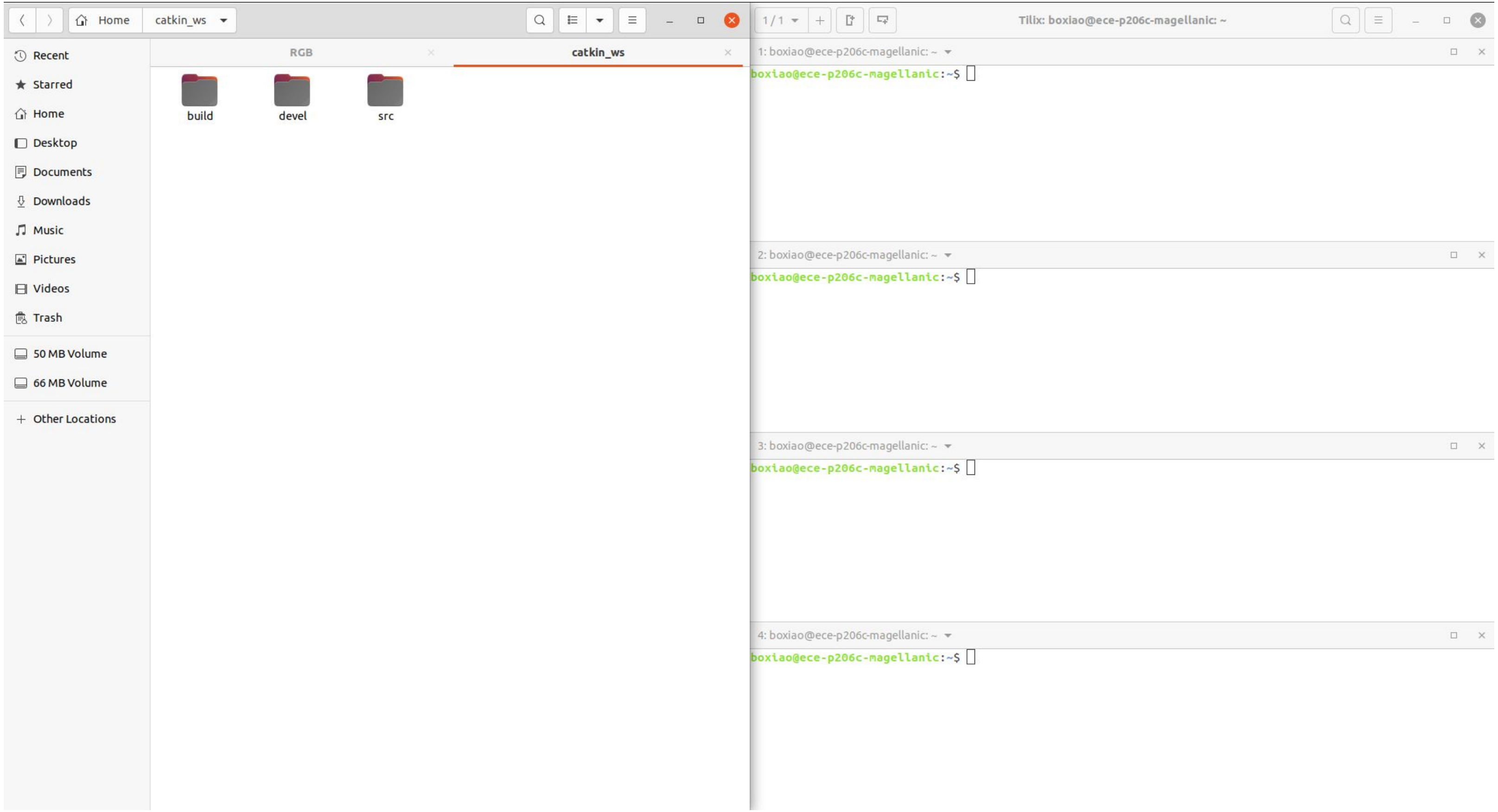
# rostopic command can display debug information about ROS Topics, including publishers, subscribers, publishing rate, and ROS Messages

Hands-on Case Study: Camera Interfacing



⇒ Use your own ROS system (PC, Jetson nano, PIs, etc.) with any USB camera.
⇒ You can also use your laptop's built-in webcam (device id: 0) for this!

Catkin Workspace And Terminals


Home
catkin_ws
Recent
Starred
Home
Desktop
Documents
Downloads
Music
Pictures
Videos
Trash
50 MB Volume
66 MB Volume
Other Locations
RGB
catkin_ws
build
devel
src
1 / 1
Tilix: boxiao@ece-p206c-magellanic: ~
1: boxiao@ece-p206c-magellanic: ~
2: boxiao@ece-p206c-magellanic: ~
3: boxiao@ece-p206c-magellanic: ~
4: boxiao@ece-p206c-magellanic: ~

```
boxiao@ece-p206c-magellanic:~$
```

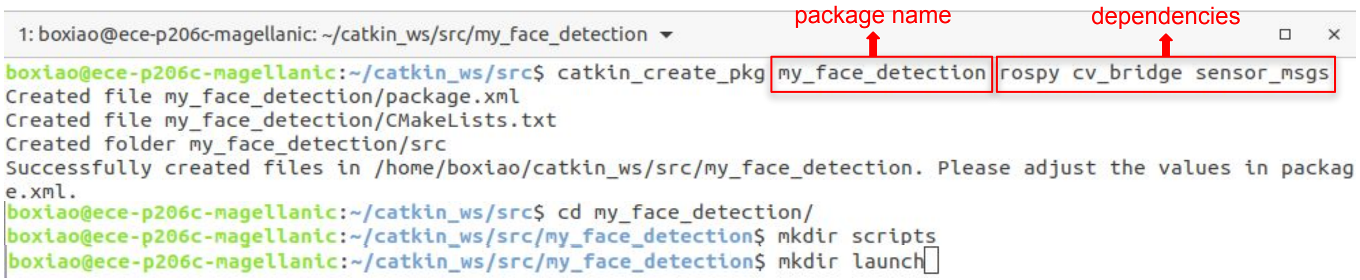
```
boxiao@ece-p206c-magellanic:~$
```

```
boxiao@ece-p206c-magellanic:~$
```

```
boxiao@ece-p206c-magellanic:~$
```

Create ROS Package In Catkin

Create a new package and corresponding scripts and launch folder

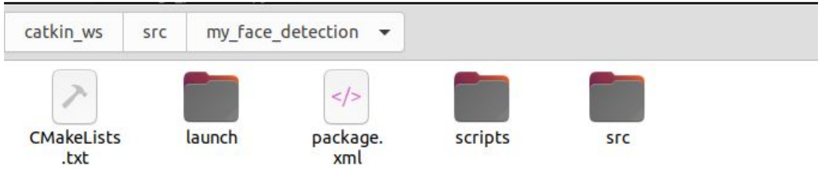

1: boxiao@ece-p206c-magellanic: ~/catkin_ws/src/my_face_detection

package name

dependencies

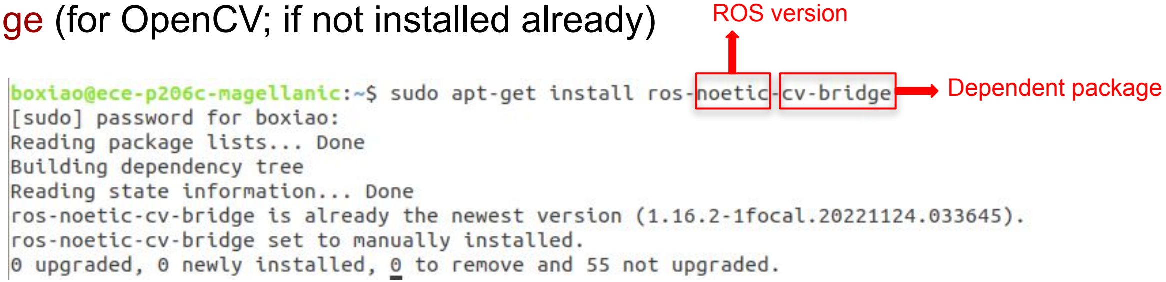
```
boxiao@ece-p206c-magellanic:~/catkin_ws/src$ catkin_create_pkg my_face_detection rospy cv_bridge sensor_msgs
Created file my_face_detection/package.xml
Created file my_face_detection/CMakeLists.txt
Created folder my_face_detection/src
Successfully created files in /home/boxiao/catkin_ws/src/my_face_detection. Please adjust the values in packag
e.xml.
boxiao@ece-p206c-magellanic:~/catkin_ws/src$ cd my_face_detection/
boxiao@ece-p206c-magellanic:~/catkin_ws/src/my_face_detection$ mkdir scripts
boxiao@ece-p206c-magellanic:~/catkin_ws/src/my_face_detection$ mkdir launch
```

Your package folder should look like this (for Python; use *roscpp* instead of *rospy* for C++)



Build Package

Install cv-bridge (for OpenCV; if not installed already)

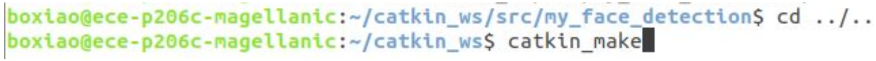


ROS version

Dependent package

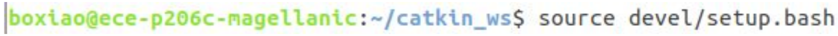
```
boxiao@ece-p206c-magellanic:~$ sudo apt-get install ros-noetic-cv-bridge
[sudo] password for boxiao:
Reading package lists... Done
Building dependency tree
Reading state information... Done
ros-noetic-cv-bridge is already the newest version (1.16.2-1focal.20221124.033645).
ros-noetic-cv-bridge set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 55 not upgraded.
```

Build the workspace with your new empty package



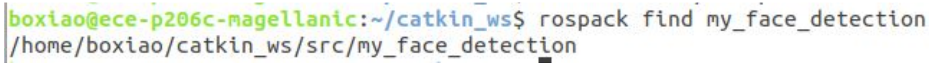
```
boxiao@ece-p206c-magellanic:~/catkin_ws/src/my_face_detection$ cd ../..
boxiao@ece-p206c-magellanic:~/catkin_ws$ catkin_make
```

Make the workspace visible to the file system (Linux way)



```
boxiao@ece-p206c-magellanic:~/catkin_ws$ source devel/setup.bash
```

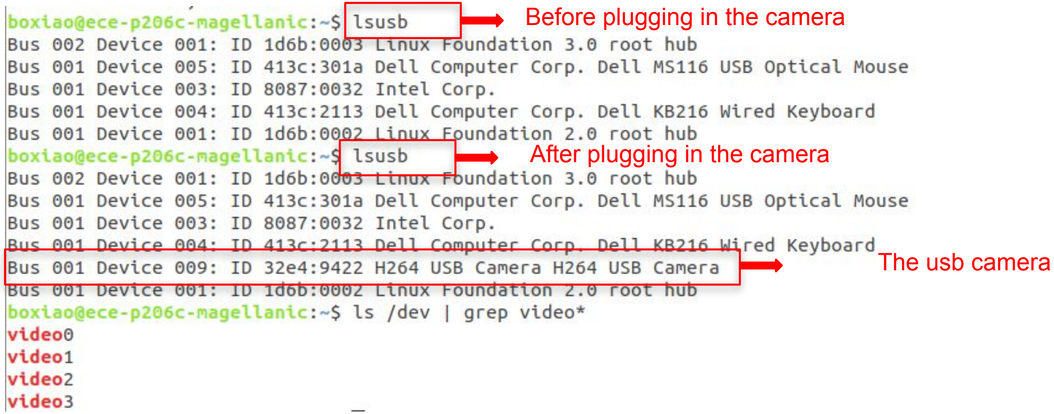
Try to find your package that you just created



```
boxiao@ece-p206c-magellanic:~/catkin_ws$ rospack find my_face_detection
/home/boxiao/catkin_ws/src/my_face_detection
```

Check The USB Camera

Plug and check if camera was recognized by system



```
boxiao@ece-p206c-magellanic:~$ lsusb
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 005: ID 413c:301a Dell Computer Corp. Dell MS116 USB Optical Mouse
Bus 001 Device 003: ID 8087:0032 Intel Corp.
Bus 001 Device 004: ID 413c:2113 Dell Computer Corp. Dell KB216 Wired Keyboard
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
boxiao@ece-p206c-magellanic:~$ lsusb
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 005: ID 413c:301a Dell Computer Corp. Dell MS116 USB Optical Mouse
Bus 001 Device 003: ID 8087:0032 Intel Corp.
Bus 001 Device 004: ID 413c:2113 Dell Computer Corp. Dell KB216 Wired Keyboard
Bus 001 Device 009: ID 32e4:9422 H264 USB Camera H264 USB Camera
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
boxiao@ece-p206c-magellanic:~$ ls /dev | grep video*
video0
video1
video2
video3
```

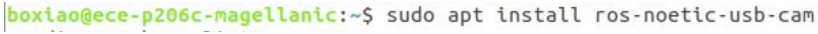
Before plugging in the camera

After plugging in the camera

The usb camera

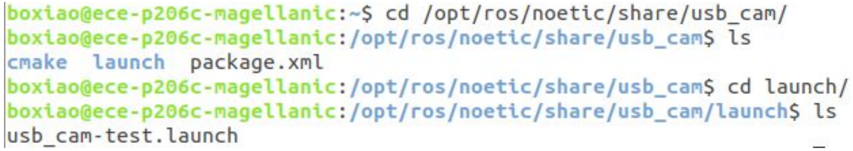
Install *usb_cam* Node

Install the usb_cam package (*ie*, camera driver)

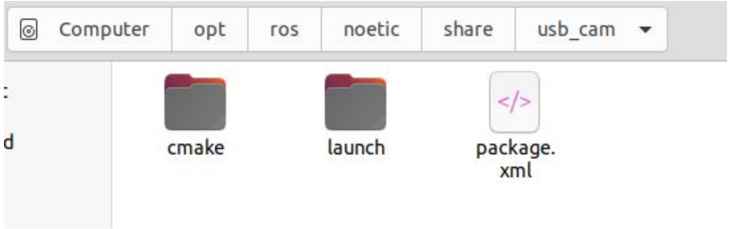


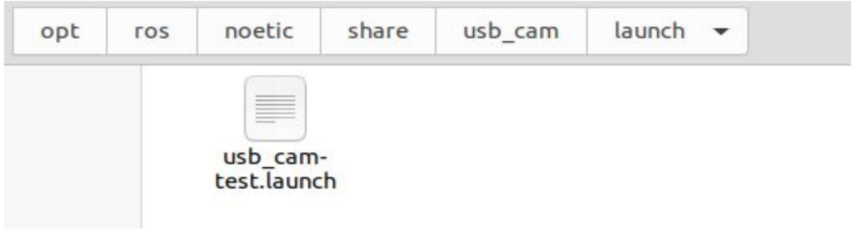
```
boxiao@ece-p206c-magellanic:~$ sudo apt install ros-noetic-usb-cam
```

Check where the packages get installed!



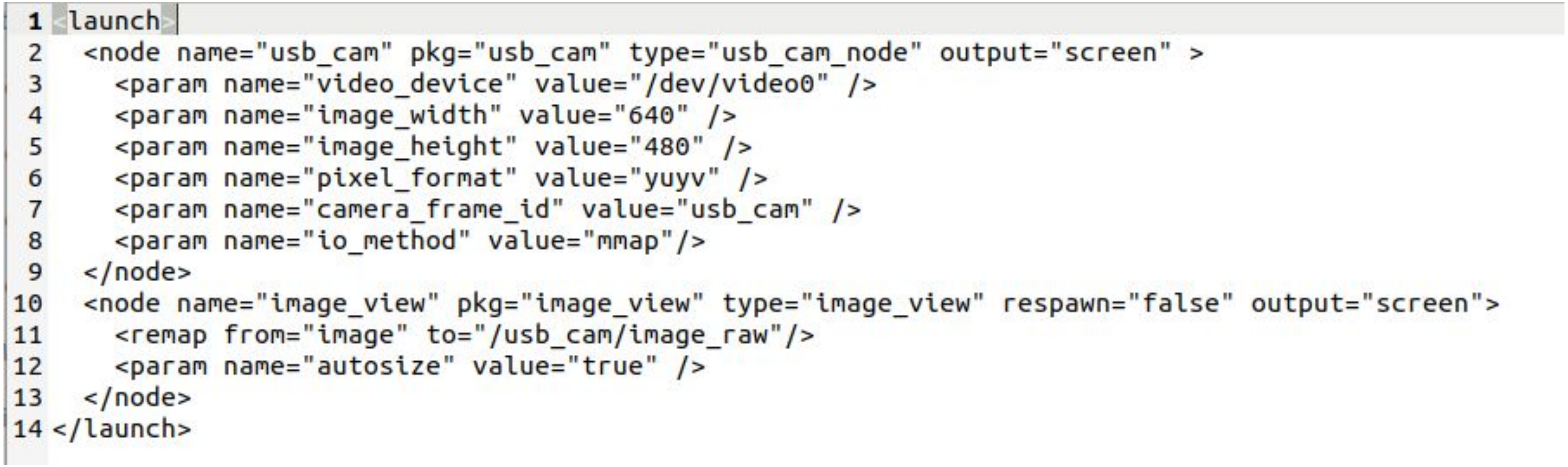
```
boxiao@ece-p206c-magellanic:~$ cd /opt/ros/noetic/share/usb_cam/
boxiao@ece-p206c-magellanic:/opt/ros/noetic/share/usb_cam$ ls
cmake  launch  package.xml
boxiao@ece-p206c-magellanic:/opt/ros/noetic/share/usb_cam$ cd launch/
boxiao@ece-p206c-magellanic:/opt/ros/noetic/share/usb_cam/launch$ ls
usb_cam-test.launch
```





Check The Launch File

usb_cam package comes with a sample test *launch file*

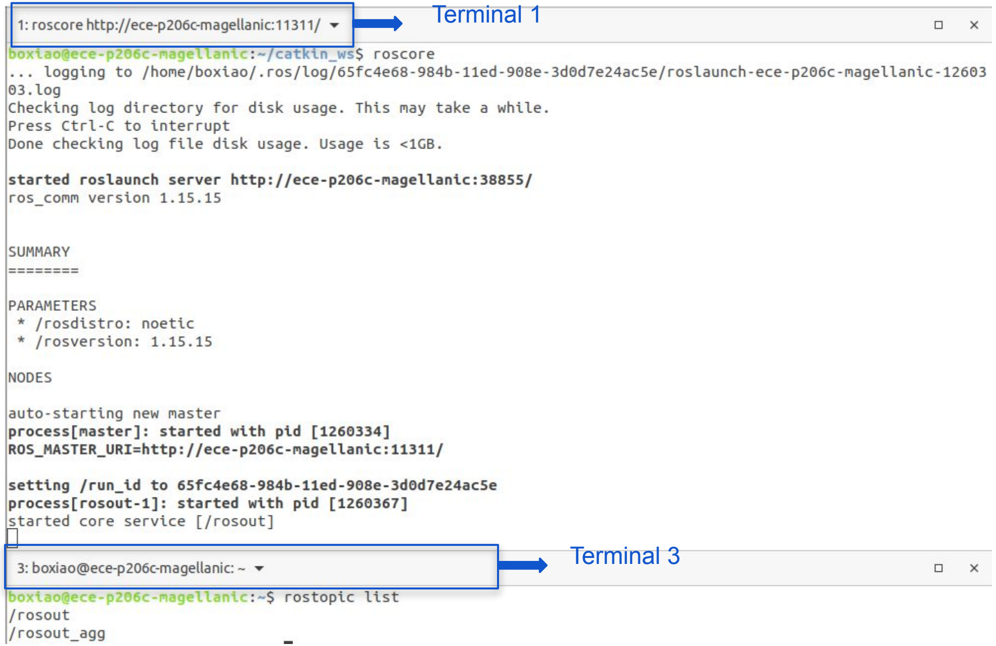


```
<launch>
  <node name="usb_cam" pkg="usb_cam" type="usb_cam_node" output="screen" >
    <param name="video_device" value="/dev/video0" />
    <param name="image_width" value="640" />
    <param name="image_height" value="480" />
    <param name="pixel_format" value="yuyv" />
    <param name="camera_frame_id" value="usb_cam" />
    <param name="io_method" value="mmap"/>
  </node>
  <node name="image_view" pkg="image_view" type="image_view" respawn="false" output="screen">
    <remap from="image" to="/usb_cam/image_raw"/>
    <param name="autosize" value="true" />
  </node>
</launch>
```

Start *roscore*

Before run the launch file, start roscore on one of the terminal

- Keep roscore running
- Check topics on another terminal before starting usb_cam


1: roscore http://ece-p206c-magellanic:11311/
3: boxiao@ece-p206c-magellanic: ~

Terminal 1

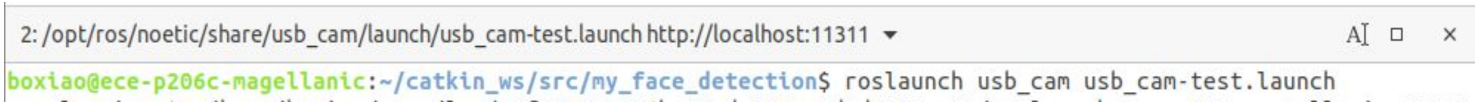
```
boxiao@ece-p206c-magellanic:~/catkin_ws$ roscore
... logging to /home/boxiao/.ros/log/65fc4e68-984b-11ed-908e-3d0d7e24ac5e/roslaunch-ece-p206c-magellanic-12603
03.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://ece-p206c-magellanic:38855/
ros_comm version 1.15.15


SUMMARY
========

PARAMETERS
 * /rosdistro: noetic
 * /rosversion: 1.15.15

NODES

auto-starting new master
process[master]: started with pid [1260334]
ROS_MASTER_URI=http://ece-p206c-magellanic:11311/

setting /run_id to 65fc4e68-984b-11ed-908e-3d0d7e24ac5e
process[rosout-1]: started with pid [1260367]
started core service [/rosout]
```

Terminal 3

```
boxiao@ece-p206c-magellanic:~$ rostopic list
/rosout
/rosout_agg
```

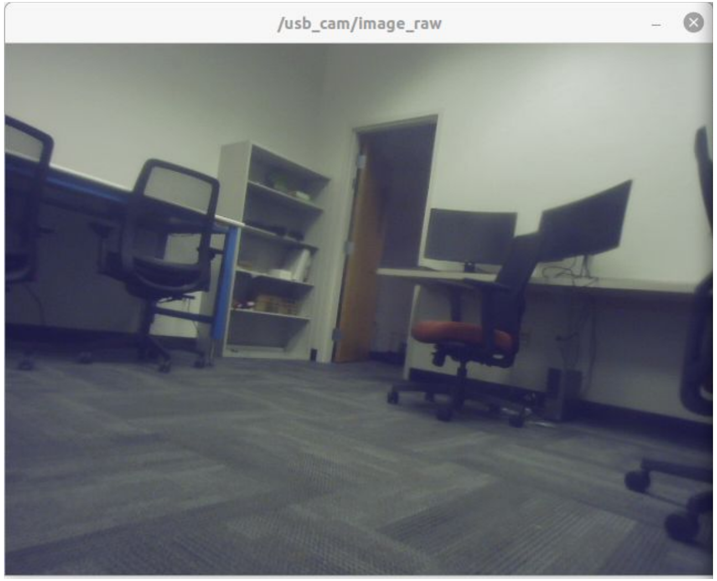
Initiate Camera

Now start usb_cam with roslaunch on a new terminal



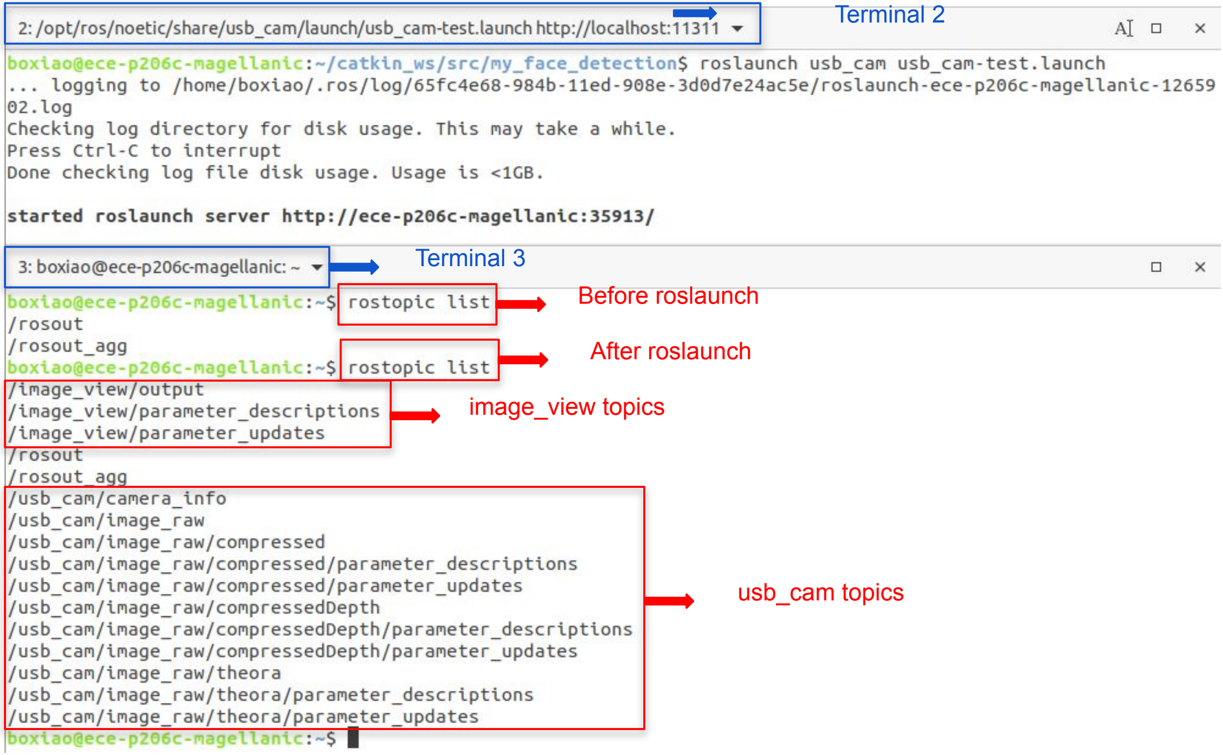
```
2: /opt/ros/noetic/share/usb_cam/launch/usb_cam-test.launch http://localhost:11311
boxiao@ece-p206c-magellanic:~/catkin_ws/src/my_face_detection$ roslaunch usb_cam usb_cam-test.launch
```

The image view window will be displayed



Get The Image Topics

Keep roslaunch running, check topics after starting usb_cam



Terminal 2

```
2: /opt/ros/noetic/share/usb_cam/launch/usb_cam-test.launch http://localhost:11311
boxiao@ece-p206c-magellanic:~/catkin_ws/src/my_face_detection$ roslaunch usb_cam usb_cam-test.launch
... logging to /home/boxiao/.ros/log/65fc4e68-984b-11ed-908e-3d0d7e24ac5e/roslaunch-ece-p206c-magellanic-12659
02.log
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://ece-p206c-magellanic:35913/
```

Terminal 3

```
3: boxiao@ece-p206c-magellanic: ~
boxiao@ece-p206c-magellanic:~$ rostopic list
/rosout
/rosout_agg
boxiao@ece-p206c-magellanic:~$ rostopic list
/image_view/output
/image_view/parameter_descriptions
/image_view/parameter_updates
/rosout
/rosout_agg
/usb_cam/camera_info
/usb_cam/image_raw
/usb_cam/image_raw/compressed
/usb_cam/image_raw/compressed/parameter_descriptions
/usb_cam/image_raw/compressed/parameter_updates
/usb_cam/image_raw/compressedDepth
/usb_cam/image_raw/compressedDepth/parameter_descriptions
/usb_cam/image_raw/compressedDepth/parameter_updates
/usb_cam/image_raw/theora
/usb_cam/image_raw/theora/parameter_descriptions
/usb_cam/image_raw/theora/parameter_updates
boxiao@ece-p206c-magellanic:~$
```

Before roslaunch

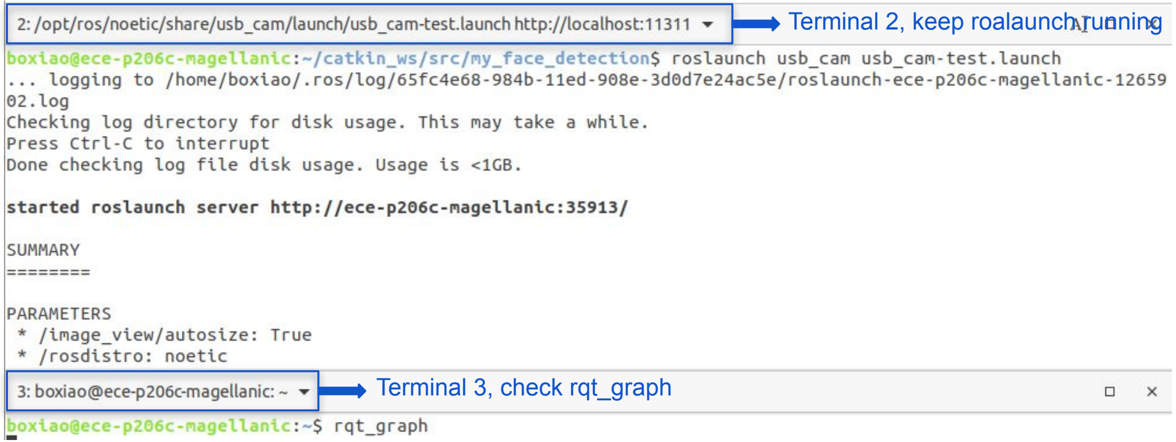
After roslaunch

image_view topics

usb_cam topics

Check The Graph!

Keep roslaunch running, check ROS computational graph



2: /opt/ros/noetic/share/usb_cam/launch/usb_cam-test.launch http://localhost:11311

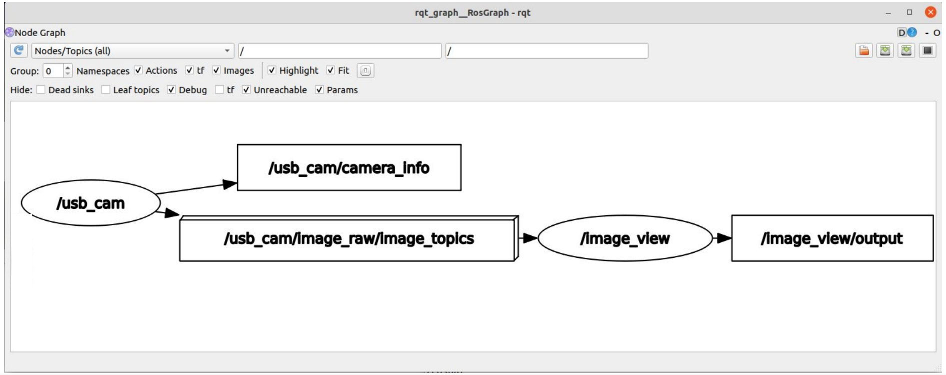
Terminal 2, keep roalaunch running

```
boxiao@ece-p206c-magellanic:~/catkin_ws/src/my_face_detection$ roslaunch usb_cam usb_cam-test.launch
... logging to /home/boxiao/.ros/log/65fc4e68-984b-11ed-908e-3d0d7e24ac5e/roslaunch-ece-p206c-magellanic-12659
02.log
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://ece-p206c-magellanic:35913/

SUMMARY
========

PARAMETERS
 * /image_view/autosize: True
 * /rosdistro: noetic
```

3: boxiao@ece-p206c-magellanic: ~

Terminal 3, check rqt_graph

```
boxiao@ece-p206c-magellanic:~$ rqt_graph
```



rqt_graph__RosGraph - rqt

Node Graph

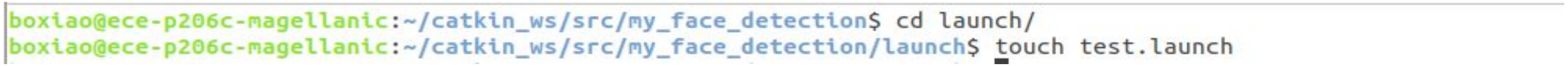
Nodes/Topics (all) | / | /

Group: 0 ☐ Namespaces ☑ Actions ☑ tf ☑ Images ☑ Highlight ☑ Fit

Hide: ☐ Dead sinks ☐ Leaf topics ☑ Debug ☐ tf ☑ Unreachable ☑ Params

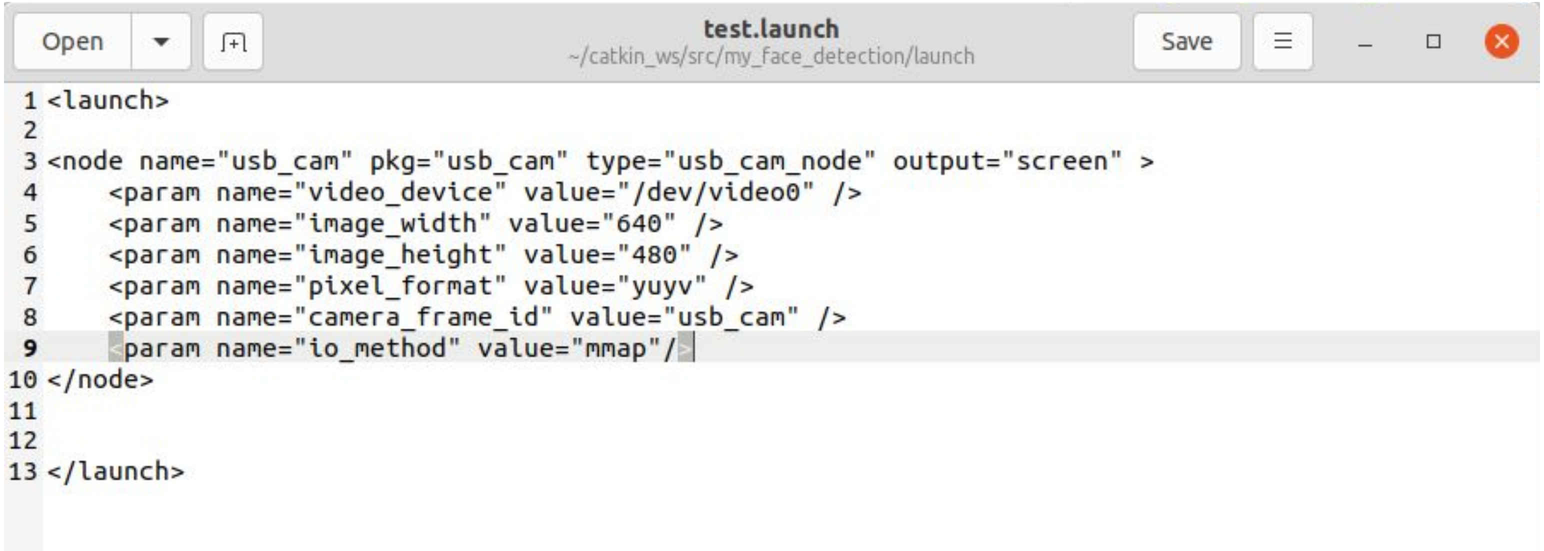
Copy-Paste Magics

Go to the launch folder of the new package, create a new launch file



```
boxiao@ece-p206c-magellanic:~/catkin_ws/src/my_face_detection$ cd launch/
boxiao@ece-p206c-magellanic:~/catkin_ws/src/my_face_detection/launch$ touch test.launch
```

Copy the content of the usb_cam node from the usb_cam launch file to the new launch file, and save

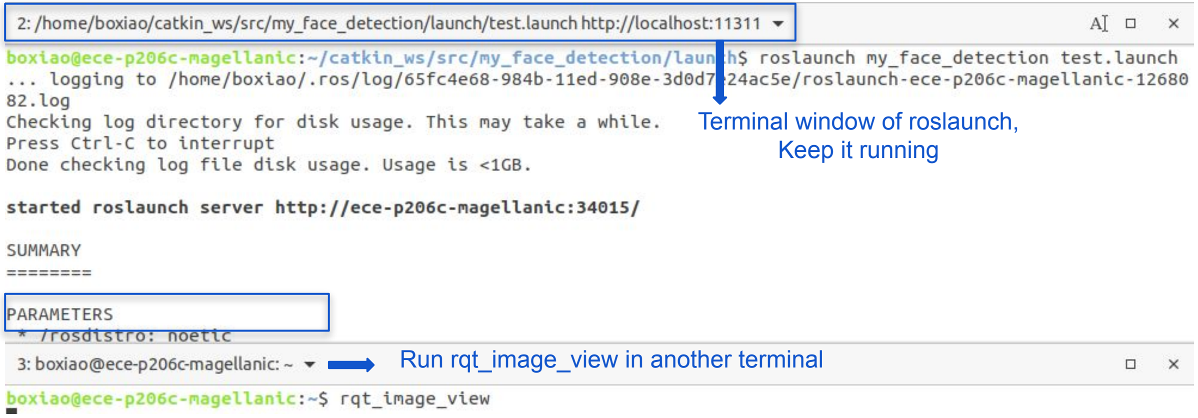


```
<launch>

<node name="usb_cam" pkg="usb_cam" type="usb_cam_node" output="screen" >
    <param name="video_device" value="/dev/video0" />
    <param name="image_width" value="640" />
    <param name="image_height" value="480" />
    <param name="pixel_format" value="yuyv" />
    <param name="camera_frame_id" value="usb_cam" />
    <param name="io_method" value="mmap"/>
</node>


</launch>
```

Check Image Topics In *rqt_image_view*

Run the new launch file (notice that we only copied one node, to initiate the camera)

- When you see the image topics (rostopic list), you can view those using rqt_image_view

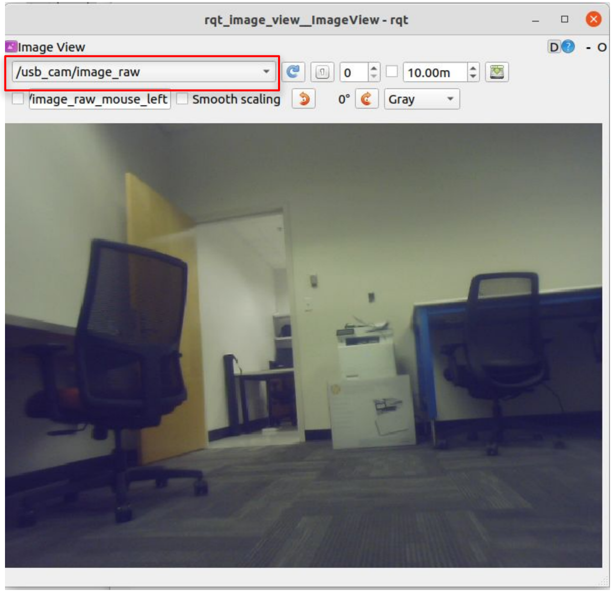


```
2: /home/boxiao/catkin_ws/src/my_face_detection/launch/test.launch http://localhost:11311
boxiao@ece-p206c-magellanic:~/catkin_ws/src/my_face_detection/launch$ roslaunch my_face_detection test.launch
... logging to /home/boxiao/.ros/log/65fc4e68-984b-11ed-908e-3d0d7e24ac5e/roslaunch-ece-p206c-magellanic-12680
82.log
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://ece-p206c-magellanic:34015/

SUMMARY
========

PARAMETERS
 * /rosdistro: noetic
```

Terminal window of roslaunch,
Keep it running

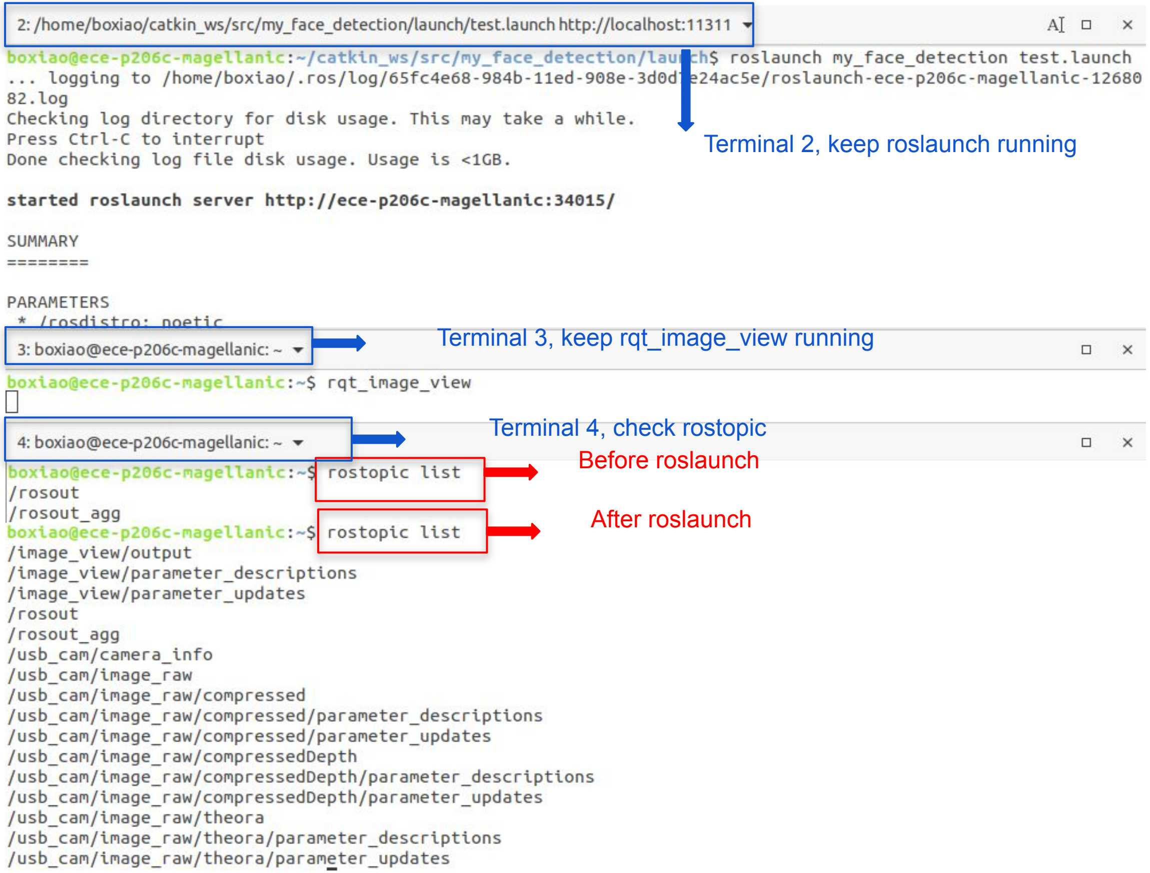
```
3: boxiao@ece-p206c-magellanic: ~
boxiao@ece-p206c-magellanic:~$ rqt_image_view
```

Run rqt_image_view in another terminal



Camera data provided by rqt_image_view

Checking Rostopics



2: /home/boxiao/catkin_ws/src/my_face_detection/launch/test.launch http://localhost:11311

```
boxiao@ece-p206c-magellanic:~/catkin_ws/src/my_face_detection/launch$ roslaunch my_face_detection test.launch
... logging to /home/boxiao/.ros/log/65fc4e68-984b-11ed-908e-3d0d7e24ac5e/roslaunch-ece-p206c-magellanic-12680
82.log
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://ece-p206c-magellanic:34015/

SUMMARY
========

PARAMETERS
 * /rosdistro: noetic
```

Terminal 2, keep roslaunch running

Terminal 3, keep rqt_image_view running

```
boxiao@ece-p206c-magellanic:~$ rqt_image_view
```

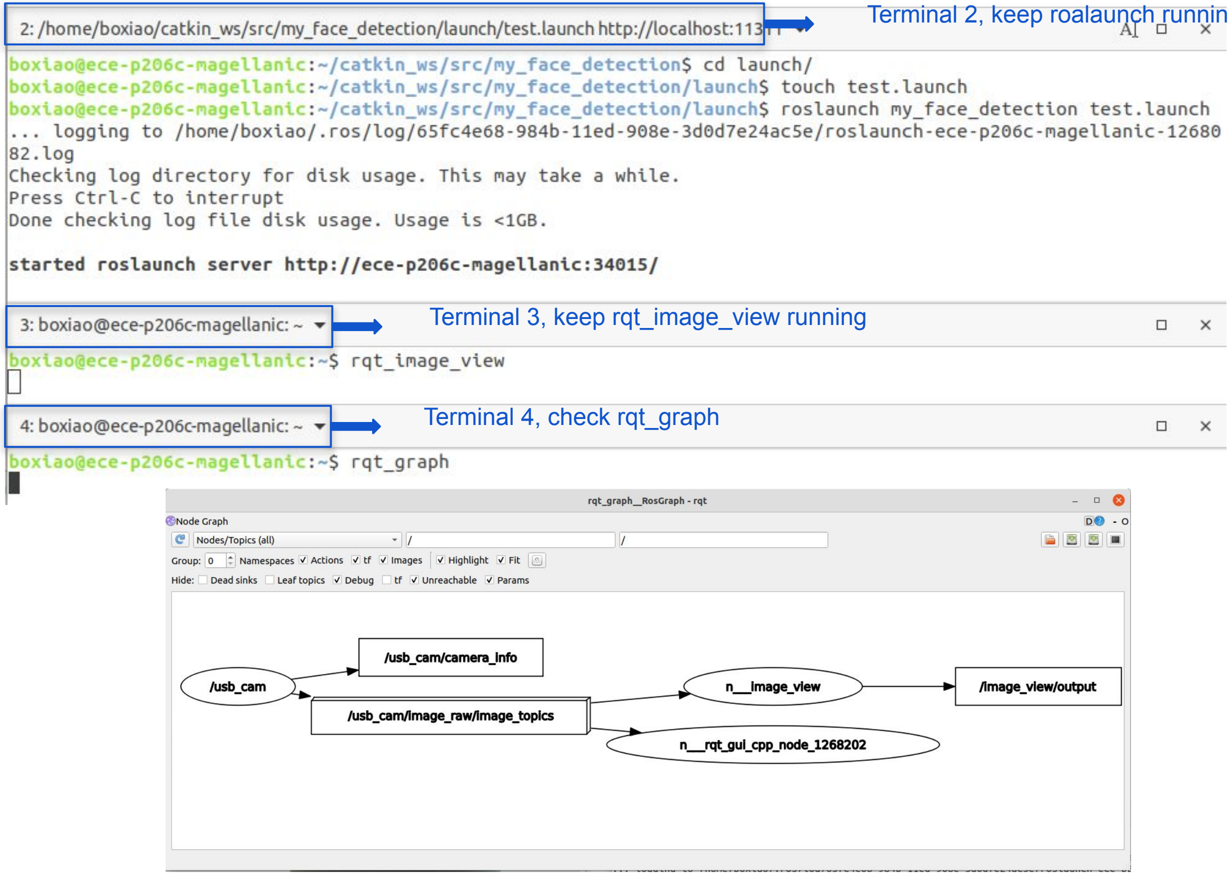
Terminal 4, check rostopic

```
boxiao@ece-p206c-magellanic:~$ rostopic list
/rosout
/rosout_agg
boxiao@ece-p206c-magellanic:~$ rostopic list
/image_view/output
/image_view/parameter_descriptions
/image_view/parameter_updates
/rosout
/rosout_agg
/usb_cam/camera_info
/usb_cam/image_raw
/usb_cam/image_raw/compressed
/usb_cam/image_raw/compressed/parameter_descriptions
/usb_cam/image_raw/compressed/parameter_updates
/usb_cam/image_raw/compressedDepth
/usb_cam/image_raw/compressedDepth/parameter_descriptions
/usb_cam/image_raw/compressedDepth/parameter_updates
/usb_cam/image_raw/theora
/usb_cam/image_raw/theora/parameter_descriptions
/usb_cam/image_raw/theora/parameter_updates
```

Before roslaunch

After roslaunch

Check *rqt_graph*



Terminal 2, keep roalaunch running

2: /home/boxiao/catkin_ws/src/my_face_detection/launch/test.launch http://localhost:113

```
boxiao@ece-p206c-magellanic:~/catkin_ws/src/my_face_detection$ cd launch/
boxiao@ece-p206c-magellanic:~/catkin_ws/src/my_face_detection/launch$ touch test.launch
boxiao@ece-p206c-magellanic:~/catkin_ws/src/my_face_detection/launch$ roslaunch my_face_detection test.launch
... logging to /home/boxiao/.ros/log/65fc4e68-984b-11ed-908e-3d0d7e24ac5e/roslaunch-ece-p206c-magellanic-12680
82.log
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://ece-p206c-magellanic:34015/
```

Terminal 3, keep rqt_image_view running

3: boxiao@ece-p206c-magellanic: ~

```
boxiao@ece-p206c-magellanic:~$ rqt_image_view
```

Terminal 4, check rqt_graph

4: boxiao@ece-p206c-magellanic: ~

```
boxiao@ece-p206c-magellanic:~$ rqt_graph
```

rqt_graph__RosGraph - rqt

Node Graph

Nodes/Topics (all) / /

Group: 0 Namespaces ☑ Actions ☑ tf ☑ Images ☑ Highlight ☑ Fit

Hide: □ Dead sinks □ Leaf topics ☑ Debug □ tf ☑ Unreachable ☑ Params

HH1: Hands-on Homework #1

Tasks Overview:

A. Prepare Workspace: ROS, Catkin, and Python-OpenCV Packages
B. Interface webcam / usb camera in ROS
   i. Initiate camera and visualize image topics
   ii. Subscribe to image topic and extract data: OpenCV-Bridge
   iii. Perform image processing: detect face draw bounding boxes (in OpenCV)
C. Publish the output image (with face boxes) as a topic: visualize topics in `rqt_image_view`
D. Write a single launch file for the whole project, *ie, that does the following*
   i. Starts the `usb_cam` node (for step B.i)
   ii. Start the `face_detector` node (for step B.ii, B.iii, and C)
   iii. Start the `rqt_image_view` node for visualization

Grading Breakdown

EEL 4930	**EEL 5934**
• Part A: 25%	• Part A: 20%
• Part B: 50% (20% + 20% + 10%)	• Part B: 45% (15% + 20% + 10%)
• Part C: 25%	• Part C: 20%
• Part D: extra! (not required, may get bonus points)	• Part D: 15%

HH1 Logistics

References:

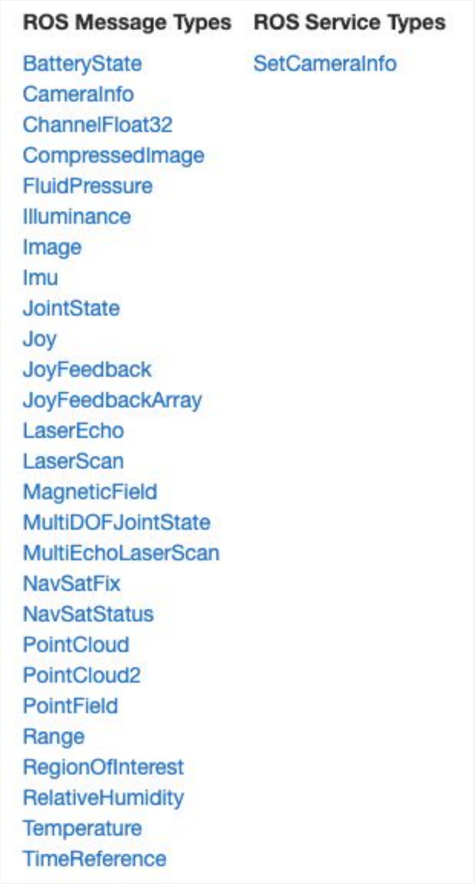
- Lecture 1-2 contents and ROS wiki
- Recommendations:
    - Use a linux laptop (virtual OS is fine) and its built-in camera
    - Alternatingly use a PC or Raspberry PI (3 or 4) or Jetson nano (use any USB camera)

Submission: [Through Canvas only; Due: Feb 7, 2023 by 11.55pm]

- A single zip file with no more than 10MB size
    - A readme.txt with your name, GatorID, ROS version, OS version, etc.
    - Your ROS package (only your new Catkin package, do not include anything else)
    - A PDF of step-by-step demo with screen-shots of terminal outputs
- Assignment more than 10 MB file size will get negative penalty (-10% to -50%)

Check the HH1 assignment: HH1_AuRo.pdf in Canvas

ROS Message Types



ROS Message Types

- BatteryState
- CameraInfo
- ChannelFloat32
- CompressedImage
- FluidPressure
- Illuminance
- Image
- Imu
- JointState
- Joy
- JoyFeedback
- JoyFeedbackArray
- LaserEcho
- LaserScan
- MagneticField
- MultiDOFJointState
- MultiEchoLaserScan
- NavSatFix
- NavSatStatus
- PointCloud
- PointCloud2
- PointField
- Range
- RegionOfInterest
- RelativeHumidity
- Temperature
- TimeReference

ROS Service Types

- SetCameraInfo

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

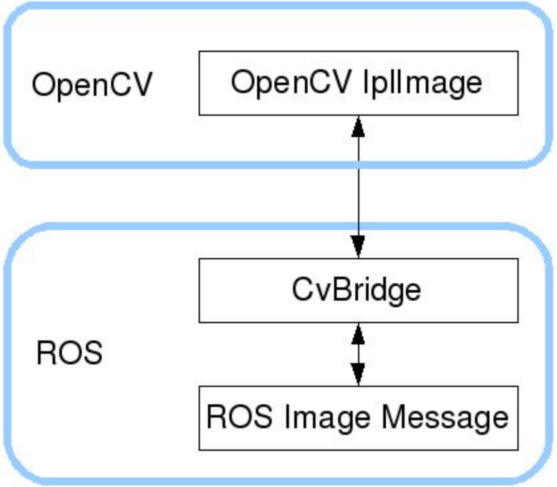
- Most commonly used ones
  - Image, CameraInfo, LaserScan, Range
  - Joy, Imu, PointCloud, PointCloud2

Interfacing sensor messages

- Check the data structure syntaxes from ROS wiki
- Conform / adjust (*ie*, wrap) data for later use
- See example codes!

Use case: how to get image from camera sensor topic to OpenCV (as Numpy array)?

ROS CVBridge



CvBridge is a ROS library

- Provides an interface between ROS and OpenCV
- Converts ROS image messages to OpenCV images
  - `CvBridge().imgmsg_to_cv2`
- Also converts ROS image messages to OpenCV images
  - `CvBridge().cv2_to_imgmsg`
- Various encoding is available
  - read more on the wiki

Subscribe:

```
    imCV = CvBridge().imgmsg_to_cv2(ros_msg, "bgr8")
```

Publish:

```
    ros_msg = CvBridge().cv2_to_imgmsg( imCV, encoding="bgr8")
```

Sample Code!

```
import cv2
import rospy
from sensor_msgs.msg import Image
from threading import Lock
from cv_bridge import CvBridge, CvBridgeError

class ImagePipeline:
    def __init__(self):
        self.mutex = Lock()
        rospy.init_node('my_node', anonymous=True)
        self.bridge = CvBridge()
        topic = '/usb_cam/image_raw'
        imRos = rospy.Subscriber(topic, Image, self.imaCallBack, queue_size=3)

        self.ImOut = rospy.Publisher('/out/image', Image, queue_size=3)

        try:
            rospy.spin()
        except KeyboardInterrupt:
            print("Rospy Spin Shut down")

    def imageCallBack(self, inp_im):
        try:
            imCV = self.bridge.imgmsg_to_cv2(inp_im, "bgr8")
        except CvBridgeError as e:
            print(e)
        if imCV is None:
            print ('frame dropped, skipping tracking')
        else:
            self.ImageProcessor(imCV)
```

How TO Detect Face In OpenCV?

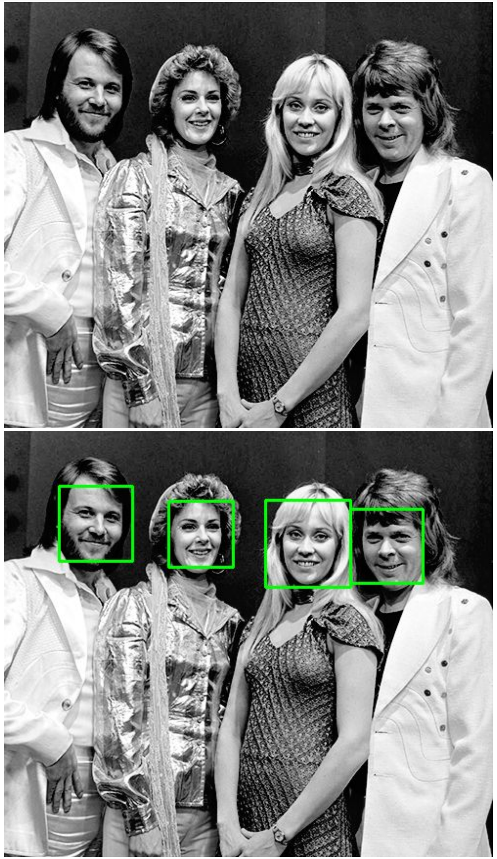
- Read an image (grayscale mode) file given the path

  -
    ```
    image = cv2.imread(img_path, 0) #grayscale-mode
    ```
- Load the cascade classifier model

  -
    ```
    faceCascade = cv2.CascadeClassifier(cascade_path)
    ```
- Detect faces

  -
    ```
    faces = faceCascade.detectMultiScale(image,
                                   scaleFactor=1.1,
                                   minNeighbors=5,
                                   minSize=(30, 30)
                                   )
    ```
- Detect bounding boxes on the image

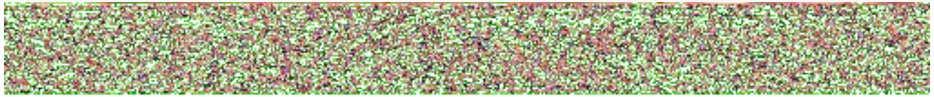
  -
    ```
    for (x, y, w, h) in faces:
        cv2.rectangle(image, (x, y), (x+w, y+h),
                             color = (0, 255, 0),
                             thickness = 2
                             )
    ```



Viola-Jones Concept

The famous Viola-Jones Algorithm

- Works with frontal face images with visible
  - Eyes and eyebrows, nose, and lips.
  - Symmetry and positioning of facial features
- Uses Haar features (see this)
- Calculates pixel features with different window sizes
- Then it finds the best features using Adaptive Boosting (Adaboost) an ML algorithm. See this for more information.
- Then uses a cascade of classifiers to identify the presence of each features.
- The accumulated scores gives the final result.

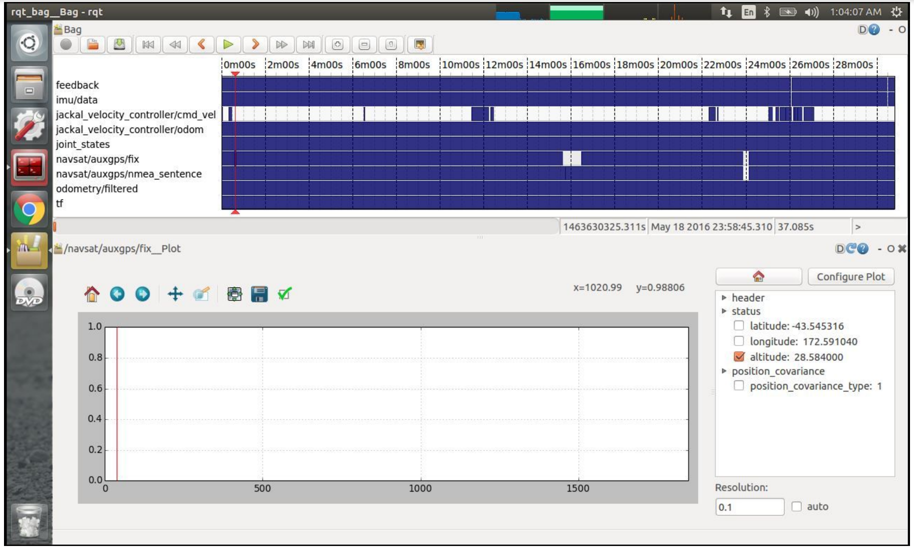


https://medium.datadriveninvestor.com/how-the-facial-detection-algorithms-work-viola-jones-algorithm-and-opencv-bd694936512f

ROS Bagging:

Useful Bag Tools

- rosbag: unified console tool for recording, playback, and other operations.
- rqt_bag: graphical tool for visualizing bag file data.
- rostopic: the echo and list commands are compatible with bag files.



feedback
imu/data
jackal_velocity_controller/cmd_vel
jackal_velocity_controller/odom
joint_states
navsat/auxgps/fix
navsat/auxgps/nmea_sentence
odometry/filtered
tf

1463630325.311s May 18 2016 23:58:45.310 37.085s

x=1020.99 y=0.98806

- header
- status
  - □ latitude: -43.545316
  - □ longitude: 172.591040
  - ☑ altitude: 28.584000
- position_covariance
  - □ position_covariance_type: 1

Resolution: 0.1 □ auto

Example commands

```
rosbag record rosout tf cmd_vel
rosbag play recorded.bag
```

See more at

- http://wiki.ros.org/Bags
- http://wiki.ros.org/rosbag/Commandline