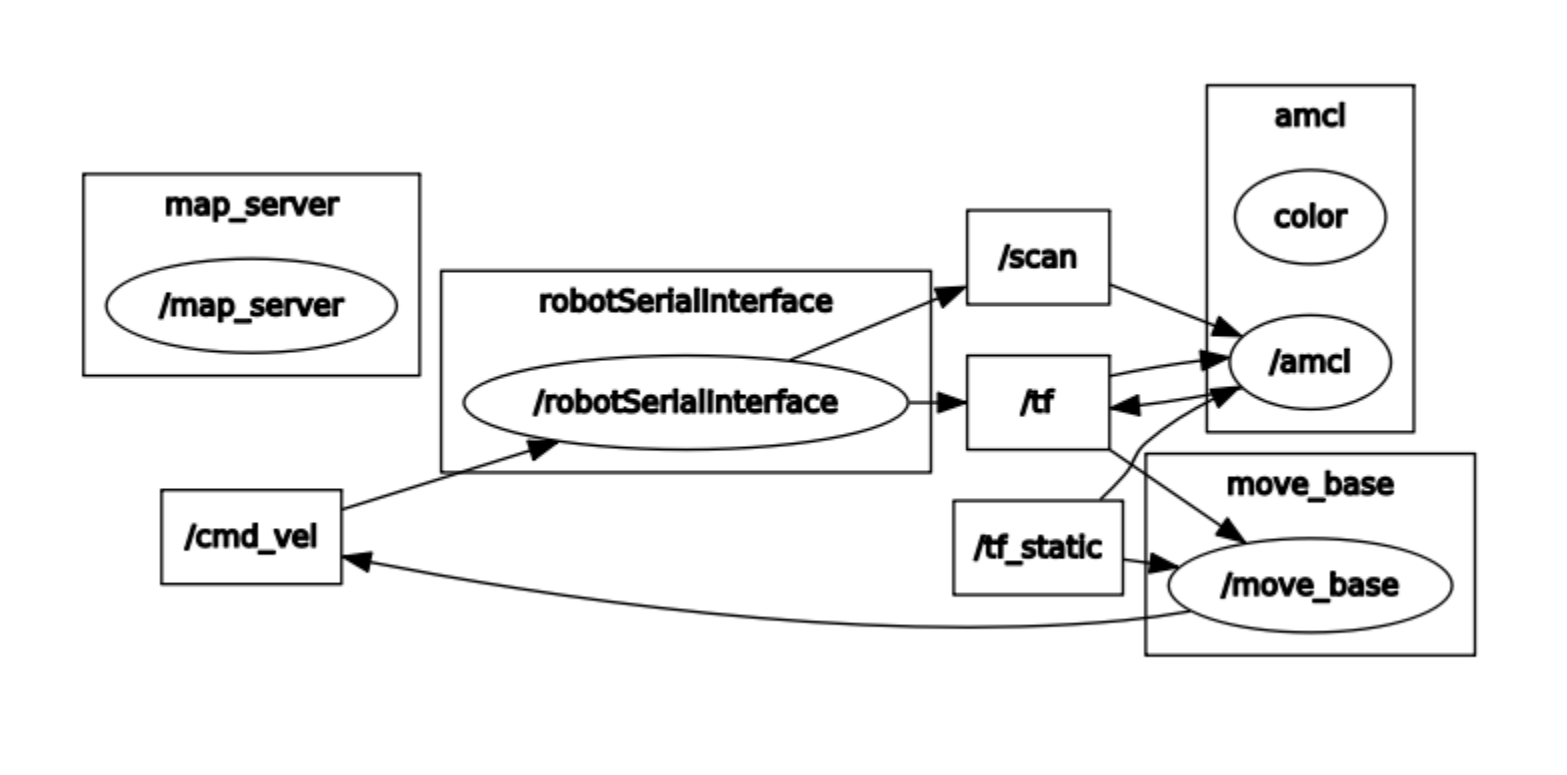
ROS – software framework

Today we learn to

1. Run
2. Inspect
3. Write

ROS components

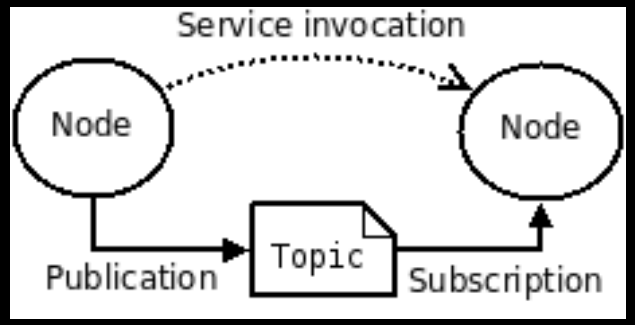


Nodes in circle – modular executables
Topics in rectangles – strongly typed message bus

SUTD ORGANIZATION OF AUTONOMOUS ROBOTICS

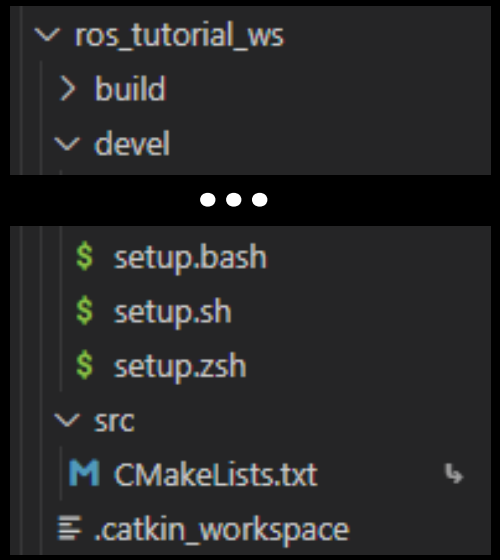
ROS Concepts

(1) ROS Computation Graph
- Run and Inspect



1. Nodes, Messages, Topics
2. rosmaster
3. Parameter Server
4. Services

(2) ROS Filesystem
- Yoink and Write



1. Packages
2. Message and Services
3. Manifests

SUTD ORGANIZATION OF AUTONOMOUS ROBOTICS

Setup your CLI environment

```
Command 'rosnode' not found,
bash: rostopic: command not found
```

1. Source ROS environment

```
source /opt/ros/${ROS_DISTRO}/setup.bash
```

```
# Sets various environment variables and sources additional environment hooks.
# It tries it's best to undo changes from a previously sourced setup file before.
```

2. Configure to autosource on start

218 `.bashrc` is a Bash shell script that Bash runs whenever it is started interactively. It initializes an interactive shell session. You can put any command in that file that you could type at the command prompt.

```
nano ~/.bashrc
```

```
root > $ .bashrc
#if [ -f /etc/bash_completion ] && ! shopt -oq posix; then
#    . /etc/bash_completion
#fi
source /opt/ros/noetic/setup.bash
```

SUTD ORGANIZATION OF AUTONOMOUS ROBOTICS

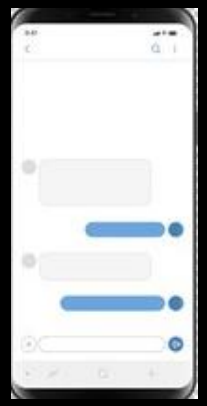
SUTD ORGANIZATION OF AUTONOMOUS ROBOTICS

#1 ROS Computation Graph

ROS Tutorial Ripoff

How Modules Communicate

1. Nodes and Topics (many-to-many, publish-subscribe)



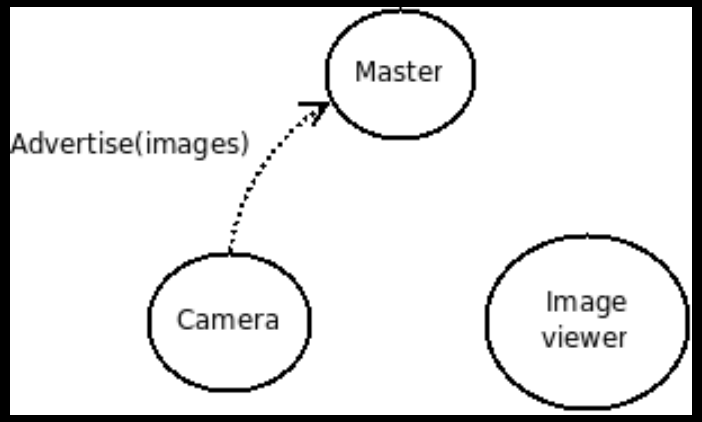
Telegram	ROS master
Telegram Groups	ROS topic
Users	ROS node
User get added to a group	ROS node subscribes to topic*
User can read the messages	ROS node reads the data*

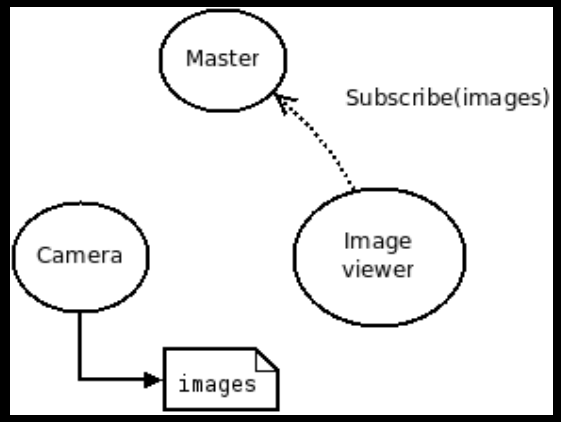
2. Services (one-way, request-response)

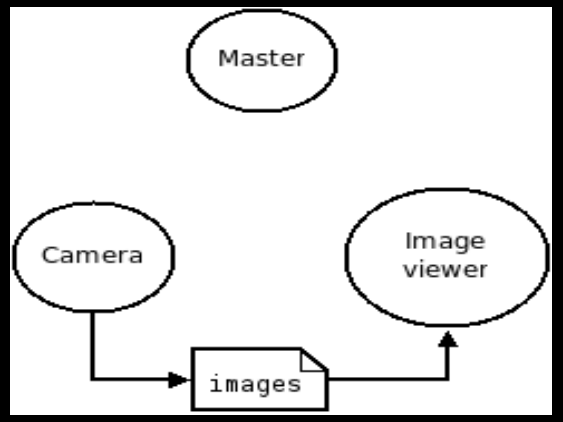
SUTD ORGANIZATION OF AUTONOMOUS ROBOTICS

Manager: rosmaster

- Name registration
- Allow nodes to exchange messages (publish, subscribe)
- Allow services to be invoked







SUTD ORGANIZATION OF AUTONOMOUS ROBOTICS

Publisher-Subscriber: Nodes, Topics, Messages

Many-to-many

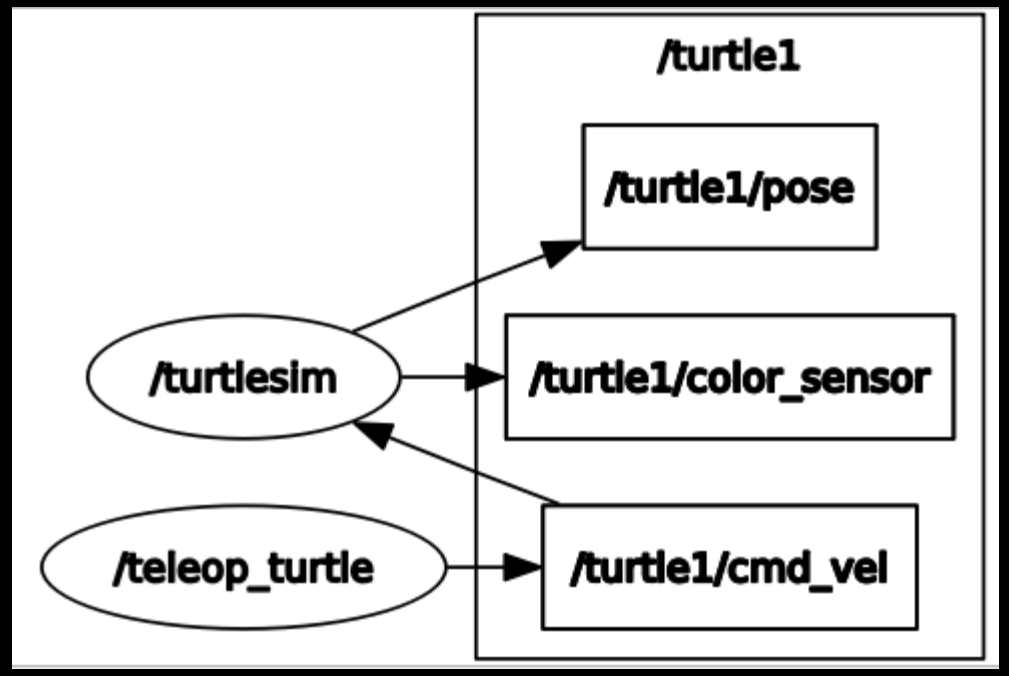
SUTD ORGANIZATION OF AUTONOMOUS ROBOTICS

Software Part: Nodes

- Modular process
- Part of full robot control system
- Sensor driver, hardware driver, controller, algorithm

Examples

- /robotSerialInterface - motor controller
- /rplidar_node – laser sensor
- /amcl – localization
- /local_planner – object avoidance



SUTD ORGANIZATION OF AUTONOMOUS ROBOTICS

Run some nodes!

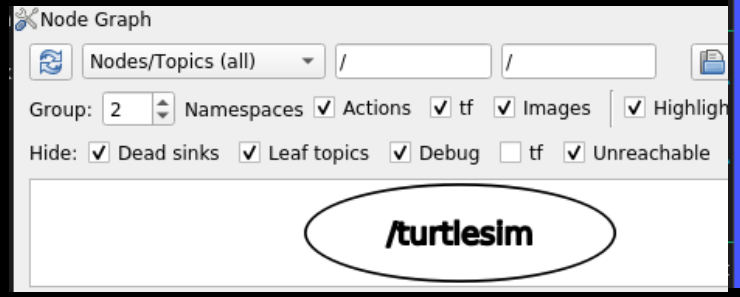
$ roscore

$ rqt_graph

$ rosrun [package] [node]

Package: turtlesim

Node: turtlesim_node




```
process[master]: started with pid [5573]
ROS_MASTER_URI=http://d7419e84fff0:11311/

setting /run_id to bbbdd934-7b69-11ec-ade0-0242ac120002
process[rosout-1]: started with pid [5583]
started core service [/rosout]

root@d7419e84fff0:/usr/src/ros-tutorial# rqt_graph
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/r
untime-root'
root@d7419e84fff0:/usr/src/ros-tutorial# rqt_graph
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/r
untime-root'

root@d7419e84fff0:/usr/src/ros-tutorial# rosrun turtlesim turt
lesim_node
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/r
untime-root'
[ INFO] [1642845488.216981999]: Starting turtlesim with node r
ame /turtlesim
[ INFO] [1642845488.227577677]: Spawning turtle [turtle1] at x
=[5.544445], y=[5.544445], theta=[0.000000]
```

Inspect some nodes!

$ rosnode list

$ rosnode info [node name]

Node name: turtlesim

```
process[master]: started with pid [5573]
ROS_MASTER_URI=http://d7419e84fff0:11311/

setting /run_id to bbbdd934-7b69-11ec-ade0-0242ac120002
process[rosout-1]: started with pid [5583]
started core service [/rosout]
```

```
root@d7419e84fff0:/usr/src/ros-tutorial# rqt_graph
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/r
untime-root'
root@d7419e84fff0:/usr/src/ros-tutorial# rqt_graph
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/r
untime-root'
```

```
root@d7419e84fff0:/usr/src/ros-tutorial# rosrun turtlesim turt
lesim_node
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/r
untime-root'
[ INFO] [1642845488.216981999]: Starting turtlesim with node n
ame /turtlesim
[ INFO] [1642845488.227577677]: Spawning turtle [turtle1] at x
=[5.544445], y=[5.544445], theta=[0.000000]
```

```
root@d7419e84fff0:/usr/src/ros-tutorial# rosnode list
/rosout
/rqt_gui_py_node_6238
/turtlesim
root@d7419e84fff0:/usr/src/ros-tutorial#
```

```
Node [/turtlesim]                                          [6/10]
Publications:
 * /rosout [rosgraph_msgs/Log]
 * /turtle1/color_sensor [turtlesim/Color]
 * /turtle1/pose [turtlesim/Pose]

Subscriptions:
 * /turtle1/cmd_vel [unknown type]

Services:
 * /clear
 * /kill
 * /reset
 * /spawn
 * /turtle1/set_pen
 * /turtle1/teleport_absolute
 * /turtle1/teleport_relative
 * /turtlesim/get_loggers
 * /turtlesim/set_logger_level

contacting node http://d7419e84fff0:40951/ ...
Pid: 5769
Connections:
```

???

???

???

SUTD ORGANIZATION OF AUTONOMOUS ROBOTICS

Comms: Messages

- Data structure
- Typed fields

Example

- geometry_msgs/Twist
- geometry_msgs/Vector3
- sensor_msgs/Imu

```
geometry_msgs/Vector3 linear
geometry_msgs/Vector3 angular
```

```
float64 x
float64 y
float64 z
```

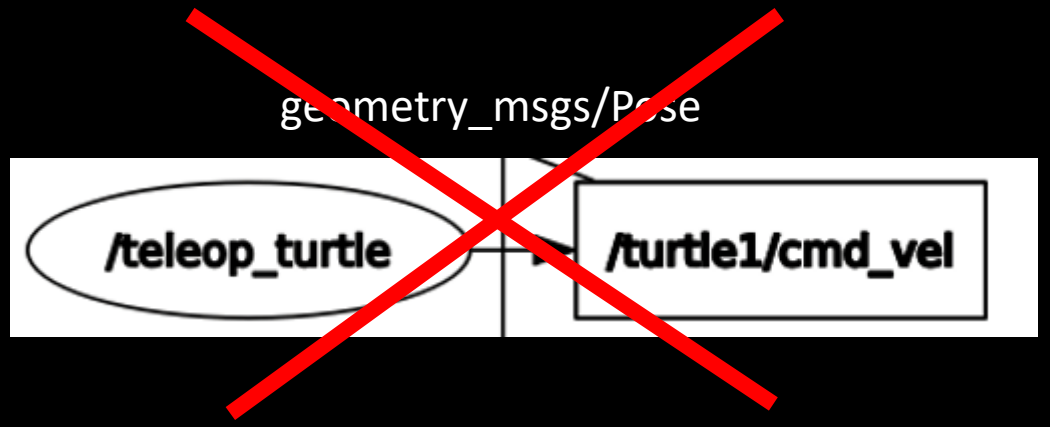
```
std_msgs/Header header
geometry_msgs/Quaternion orientation
float64[9] orientation_covariance
geometry_msgs/Vector3 angular_velocity
float64[9] angular_velocity_covariance
geometry_msgs/Vector3 linear_acceleration
float64[9] linear_acceleration_covariance
```

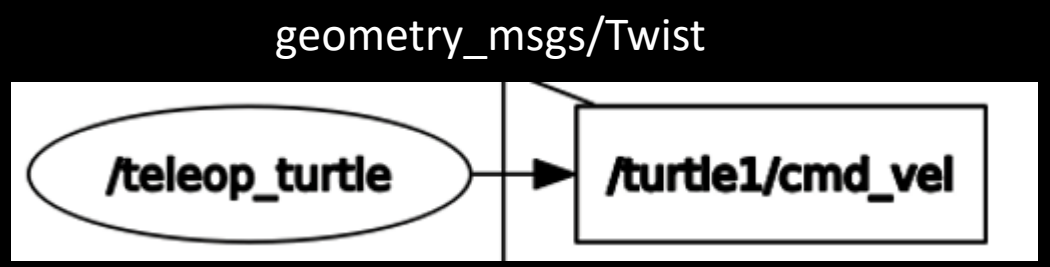
Comms: Topics

- Any node can send or receive from it
- **ONLY accept a message type**
- Publish and Subscribe

Examples

- /scan: sensor_msgs/LaserScan
- /cmd_vel: geometry_msgs/Twist






```
geometry_msgs/Vector3 linear
geometry_msgs/Vector3 angular
```

SUTD ORGANIZATION OF AUTONOMOUS ROBOTICS

Inspect some nodes!

```
process[master]: started with pid [5573]
ROS_MASTER_URI=http://d7419e84fff0:11311/

setting /run_id to bbbdd934-7b69-11ec-ade0-0242ac120002
process[rosout-1]: started with pid [5583]
started core service [/rosout]
```

```
root@d7419e84fff0:/usr/src/ros-tutorial# rqt_graph
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/r
untime-root'
root@d7419e84fff0:/usr/src/ros-tutorial# rqt_
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/r
untime-root'
```

```
root@d7419e84fff0:/usr/src/ros-tutorial# rosrun turtlesim turt
lesim_node
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/r
untime-root'
[ INFO] [1642845488.216981999]: Starting turtlesim with node n
ame /turtlesim
[ INFO] [1642845488.227577677]: Spawning turtle [turtle1] at x
=[5.544445], y=[5.544445], theta=[0.000000]
```

```
root@d7419e84fff0:/usr/src/ros-tutorial# rosnode list
/rosout
/rqt_gui_py_node_6238
/turtlesim
root@d7419e84fff0:/usr/src/ros-tutorial#
```

```
Node [/turtlesim]                                          [6/10]
Publications:
 * /rosout [rosgraph_msgs/Log]
 * /turtle1/color_sensor [turtlesim/Color]
 * /turtle1/pose [turtlesim/Pose]

Subscriptions:
 * /turtle1/cmd_vel [unknown type]

Services:
 * /clear
 * /kill
 * /reset
 * /spawn
 * /turtle1/set_pen
 * /turtle1/teleport_absolute
 * /turtle1/teleport_relative
 * /turtlesim/get_loggers
 * /turtlesim/set_logger_level

contacting node http://d7419e84fff0:40951/ ...
Pid: 5769
Connections:
```

Topics

Publishing to

Subscribed to

???

SUTD ORGANIZATION OF AUTONOMOUS ROBOTICS

Let them communicate!

$ rosrun [package] [node]

Package: turtlesim

Node: turtle_teleop_key

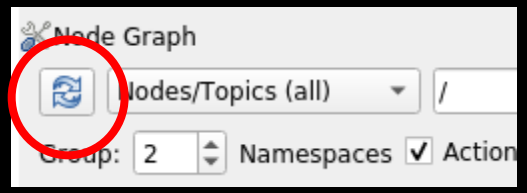
$ rostopic list -v

$ rostopic echo [topic name]

Topic name: /turtle1/cmd_vel

Control the turtle!

Refresh rqt, see anything different?

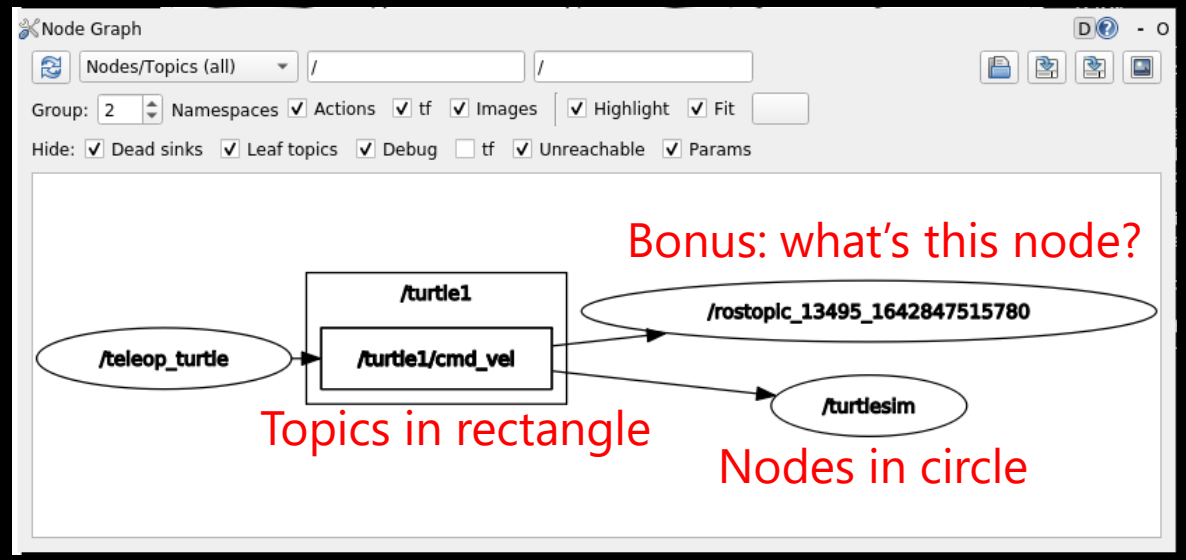



```
root@d7419e84fff0:/usr/src/ros-tutorial# rosrun turtlesim turt
le_teleop_key
Reading from keyboard
---------------------------
Use arrow keys to move the turtle. 'q' to quit.


Published topics:
 * /rosout_agg [rosgraph_msgs/Log] 1 publisher
 * /rosout [rosgraph_msgs/Log] 4 publishers
 * /turtle1/pose [turtlesim/Pose] 1 publisher
 * /turtle1/color_sensor [turtlesim/Color] 1 publisher
 * /turtle1/cmd_vel [geometry_msgs/Twist] 1 publisher

Subscribed topics:
 * /rosout [rosgraph_msgs/Log] 1 subscriber
 * /statistics [rosgraph_msgs/TopicStatistics] 1 subscriber
 * /turtle1/cmd_vel [geometry_msgs/Twist] 2 subscribers

root@d7419e84fff0:/usr/src/ros-tutorial#
```

```
linear:
  x: 2.0
  y: 0.0
  z: 0.0
angular:
  x: 0.0
  y: 0.0
  z: 0.0
---
```

SUTD ORGANIZATION OF AUTONOMOUS ROBOTICS

Inspect some topics!




```
root@d7419e84fff0:/usr/src/ros-tutorial# rosrun turtlesim turt
le_teleop_key
Reading from keyboard
---------------------------
Use arrow keys to move the turtle. 'q' to quit.

Published topics:
 * /rosout_agg [rosgraph_msgs/Log] 1 publisher
 * /rosout [rosgraph_msgs/Log] 4 publishers
 * /turtle1/pose [turtlesim/Pose] 1 publisher
 * /turtle1/color_sensor [turtlesim/Color] 1 publisher
 * /turtle1/cmd_vel [geometry_msgs/Twist] 1 publisher

Subscribed topics:
 * /rosout [rosgraph_msgs/Log] 1 subscriber
 * /statistics [rosgraph_msgs/TopicStatistics] 1 subscriber
 * /turtle1/cmd_vel [geometry_msgs/Twist] 2 subscribers

root@d7419e84fff0:/usr/src/ros-tutorial#
```

published

subscribed

```
linear:
  x: 2.0
  y: 0.0
  z: 0.0
angular:
  x: 0.0
  y: 0.0
  z: 0.0
---
```

data sent in topic

SUTD ORGANIZATION OF AUTONOMOUS ROBOTICS

Inspect harder…

$ rostopic type [topic name]

Topic name: /turtlesim1/cmd_vel

$ rosmsg show [message type]

Message type: geometry_msgs

$ rostopic pub [topic name] [message type] [message data] [extra args]

Topic name: /turtlesim1/cmd_vel

Message type: geometry_msgs/Twist

Message data: “linear: x:2.0 y:0.0 z:0.0 angular: x:0.0 y:0.0 z:1.8”

Extra args: -r 1

```
root@d7419e84fff0:/usr/src/ros-tutorial# rostopic info /[3/26]
1/cmd_vel
Type: geometry_msgs/Twist

Publishers: None

Subscribers:
 * /turtlesim (http://d7419e84fff0:43997/)
```

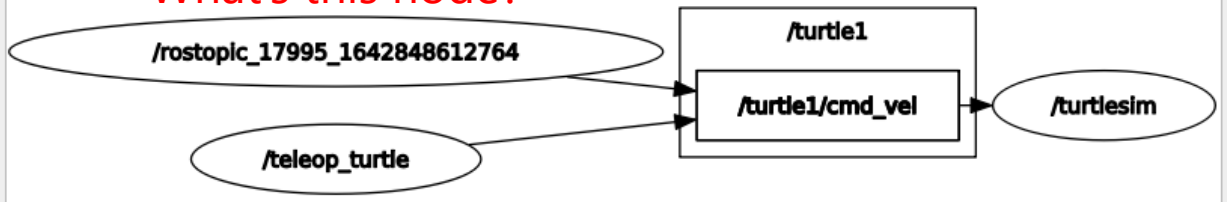
```
root@d7419e84fff0:/usr/src/ros-tutorial# rosmsg show geo[2/78]
msgs/Twist
geometry_msgs/Vector3 linear
  float64 x
  float64 y
  float64 z
geometry_msgs/Vector3 angular
  float64 x
  float64 y
  float64 z
```

Message structure

```
root@d7419e84fff0:/usr/src/ros-tutorial# rostopic pub /turtle1
/cmd_vel geometry_msgs/Twist "linear:
  x: 2.0
  y: 0.0
  z: 0.0
angular:
  x: 0.0
  y: 0.0
  z: 1.8" -r 1
```

Adhere to message structure

What’s this node?



ROS commands so far

Executables (rosrun, roslaunch)

```
roslaunch modular_pub_sub launch_Node.launch
```

```
rosrun turtlesim turtlesim_node
```

```
rosrun map_server map_saver -f ~/my_map
```

Command Package Node Extra arguments

Inspection (rosnode, rostopic)

```
rosnode info /turtlesim
```

```
rostopic echo /turtle1/command_velocity
```

Command Subcommand Topic/Node Name

Request-Response: Services, Parameters

One-way

SUTD ORGANIZATION OF AUTONOMOUS ROBOTICS

Comms: Services

- Request and response message structure
- Node offers service under a name
- Client sends request, awaits response

Examples

- make_plan: nav_msgs/GetPlan
- set_map: nav_msgs/SetMap

```
geometry_msgs/PoseStamped start
geometry_msgs/PoseStamped goal
float32 tolerance
---
nav_msgs/Path plan
```

```
nav_msgs/OccupancyGrid map
geometry_msgs/PoseWithCovarianceStamped initial_pose
---
bool success
```

SUTD ORGANIZATION OF AUTONOMOUS ROBOTICS

Inspect some services!

$ rosservice list

$ rosservice type [service]

Service: /spawn

$ rossrv show [service type]

Service type: turtlesim/Spawn

```
root@d7419e84fff0:/usr/src/ros-tutorial# rosservice list
/clear
/kill
/reset
/rosout/get_loggers
/rosout/set_logger_level
/rqt_gui_py_node_13234/get_loggers
/rqt_gui_py_node_13234/set_logger_level
/rqt_gui_py_node_17807/get_loggers
/rqt_gui_py_node_17807/set_logger_level
/spawn
/teleop_turtle/get_loggers
/teleop_turtle/set_logger_level
/turtle1/set_pen
/turtle1/teleport_absolute
/turtle1/teleport_relative
/turtlesim/get_loggers
/turtlesim/set_logger_level
```

```
root@d7419e84fff0:/usr/src/ros-tutorial# rosservice type /spawn[1/89]
turtlesim/Spawn
```

```
root@d7419e84fff0:/usr/src/ros-tutorial# rossrv show turtlesim[0/317]
float32 x
float32 y
float32 theta
string name
---
string name
```

Request structure

Response structure

SUTD ORGANIZATION OF AUTONOMOUS ROBOTICS

Call some services!

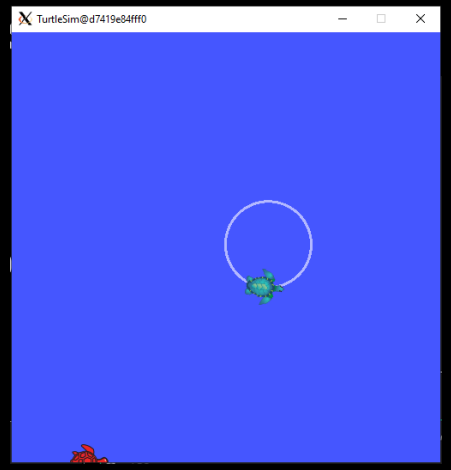
$ rossrv show [service type]

Service type: turtlesim/Spawn

$ rosservice call [service] [service request]

Service: /spawn

Service request: “x:2.0 y:0.0 theta:0.0 name:’turtle2’”




```
root@d7419e84fff0:/usr/src/ros-tutorial# rosservice call /spawn
"x: 2.0
y: 0.0
theta: 0.0
name: 'turtle2'"
name: "turtle2"
```

Adhere to request structure

Response

```
root@d7419e84fff0:/usr/src/ros-tutorial# rossrv show turtlesim/S
pawn
float32 x
float32 y
float32 theta
string name
---
string name
root@d7419e84fff0:/usr/src/ros-tutorial#
```

Request structure

Response structure

SUTD ORGANIZATION OF AUTONOMOUS ROBOTICS

Parameter Server

- Nodes store and retrieve from server
- For configuration parameters
- Globally viewable

Examples

~/camera/left/exposure: int

~/base_global_planner: string

~/min_particles: int

Get/Set some params!

$ rosparam list

$ rosparam get [param namespace]

Param namespace: / or /turtlesim

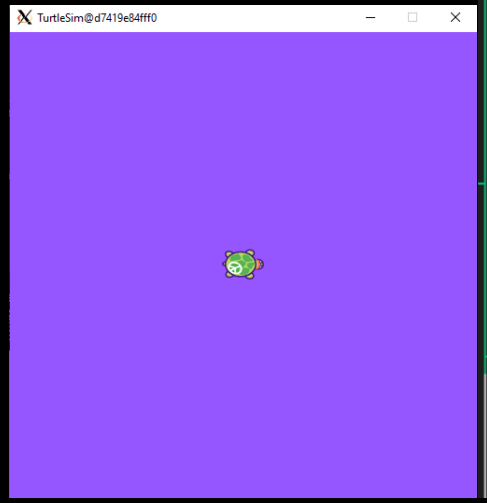
$ rosparam set [param namespace] [value]

Param namespace: /turtlesim/background_r

Value: 150

$ rosservice call [service]

Service: /clear




```
root@d7419e84fff0:/usr/src/ros-tutorial# rosparam list      [1/1]
/rosdistro
/roslaunch/uris/host_d7419e84fff0__34559
/rosversion
/run_id
/turtlesim/background_b
/turtlesim/background_g
/turtlesim/background_r
```

```
root@d7419e84fff0:/usr/src/ros-tutorial# rosparam get /     [2/2]
rosdistro: 'noetic

  .
roslaunch:
  uris:
    host_d7419e84fff0__34559: http://d7419e84fff0:34559/
rosversion: '1.15.11

  .
run_id: 3d6d5a34-7ba8-11ec-9a6d-0242ac120002
turtlesim:
  background_b: 255
  background_g: 86
  background_r: 69
```

```
root@d7419e84fff0:/usr/src/ros-tutorial# rosparam set /turtlesi
m/background_r 150
root@d7419e84fff0:/usr/src/ros-tutorial# rosservice call /clear
```

Save/Load some params!

$ rosparam dump [file path] [param namespace]

File path: turtle_params.yaml

Param namespace: /turtlesim

$ rosparam set [param namespace] [value]

Param namespace: /turtlesim/background_b

Value: 0

$ rosservice call [service]

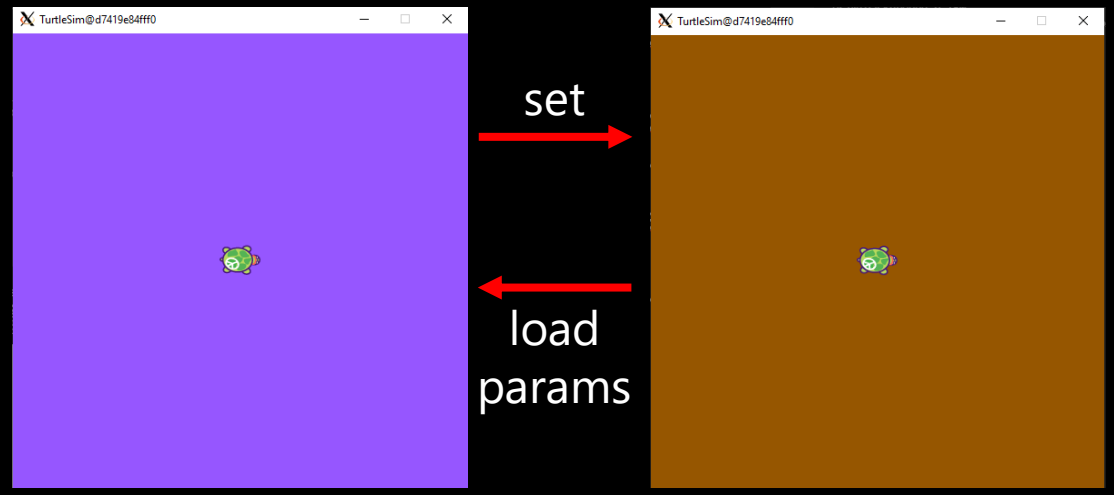
Service: /clear

$ rosparam load [file path] [param namespace]

File path: turtle_params.yaml

Param namespace: /turtlesim

```
root@d7419e84fff0:/usr/src/ros-tutorial/ros_tutorial_ws# rosparam dump params.yaml /turtlesim
root@d7419e84fff0:/usr/src/ros-tutorial/ros_tutorial_ws# ls
build  devel  params.yaml  src
root@d7419e84fff0:/usr/src/ros-tutorial/ros_tutorial_ws# rosparam set /turtlesim/background_b 0
root@d7419e84fff0:/usr/src/ros-tutorial/ros_tutorial_ws# rosservice call /clear

root@d7419e84fff0:/usr/src/ros-tutorial/ros_tutorial_ws# rosparam load params.yaml /turtlesim
root@d7419e84fff0:/usr/src/ros-tutorial/ros_tutorial_ws# rosservice call /clear
```



SUTD ORGANIZATION OF AUTONOMOUS ROBOTICS

Recap

Modules

- Nodes: Software Part

Comms

- Msgs: Typed data structure
- Topics: For nodes to read / write to, defined by Msgs
- Services: For nodes to request / respond to one another
- Params: Get/Set/Save/Load variables in nodes

SUTD ORGANIZATION OF AUTONOMOUS ROBOTICS

SUTD ORGANIZATION OF AUTONOMOUS ROBOTICS

#2 Workspaces, Packages

ROS Tutorial Ripoff

What's a workspace?

1. Catkin Workspaces

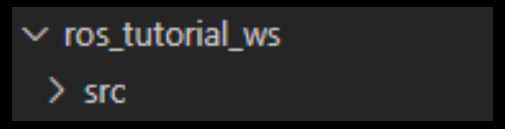
A catkin workspace is a folder where you modify, build, and install catkin packages. It is specified in REP 128. The following is the recommended and typical catkin workspace layout:

- ros_tutorial_ws ← Workspace
  - build ← Build space
  - devel ← Dev space
  - ...
  - setup.bash ← **source this to run anything from your workspace**
  - setup.sh
  - setup.zsh
  - src ← Source space
  - CMakeLists.txt
  - .catkin_workspace

Setup a workspace

1. Make a folder for the workspace

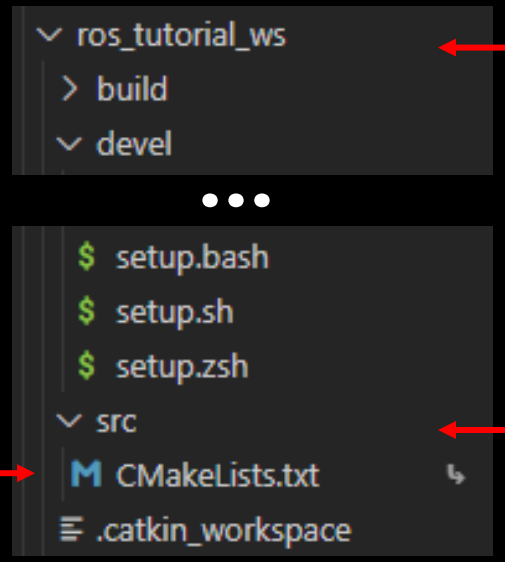
   $ mkdir –p [workspace name]/src



2. Build the workspace

   $ cd [workspace name]

   $ catkin_make

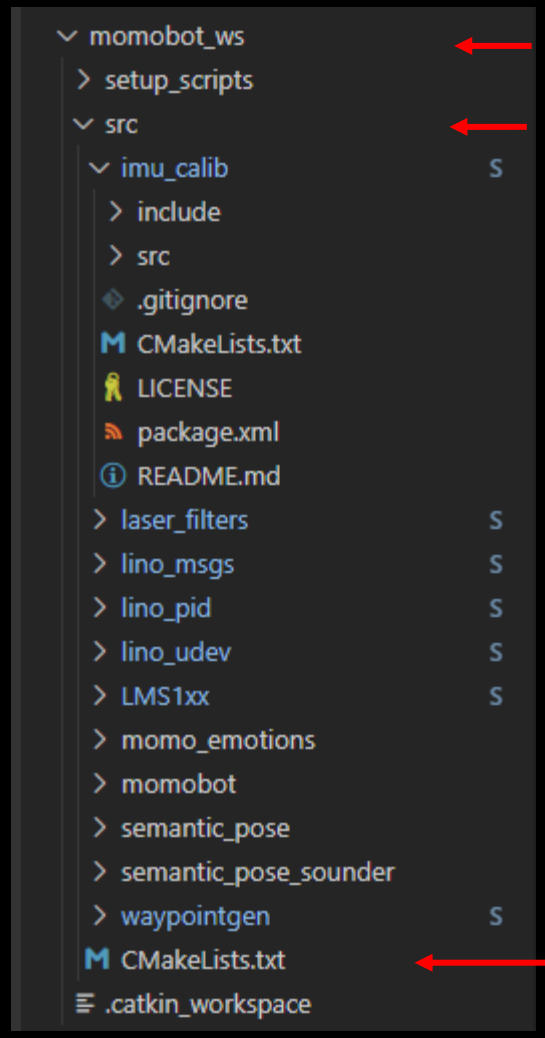


Workspace

Source space

Toplevel CMake

Get familiar with it



Workspace

Source space

Toplevel CMake

SUTD ORGANIZATION OF AUTONOMOUS ROBOTICS

Environment

Package - Module

- Software in ROS is organized in packages
- Logical, useful
- Reusable, lightweight

Example

- sensor/hardware driver
- local/global path planner

1. tf2-ros (7,519,789)
2. tf2-msgs (7,151,771)
3. tf2-py (7,151,201)
4. tf2 (7,131,219)
5. sensor-msgs (7,055,333)
6. message-filters (6,880,700)
7. cv-bridge (6,879,990)
8. pluginlib (6,877,906)
9. std-msgs (6,867,367)
10. geometry-msgs (6,838,309)
11. rosgraph-msgs (6,821,155)
12. image-transport (6,794,278)
13. tf (6,707,692)
14. actionlib-msgs (6,674,208)
15. urdf (6,580,563)

manages relationships between coordinate frames through time

ROS image <-> OpenCV image

custom message types

markdown to represent robot model

Yoink a package: from apt

$ apt search ros-noetic*

$ apt install ros-noetic-slam-gmapping

Black magic?

- You added the package repo during installation

1.2 Setup your sources.list

Setup your computer to accept software from packages.ros.org.

```
sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main" > /etc/apt/sources.list.
d/ros-latest.list'
```

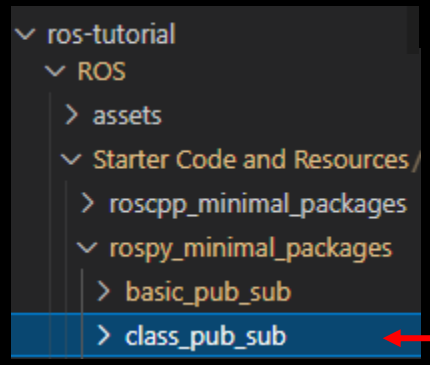
- apt checks requirements, downloads and installs them

SUTD ORGANIZATION OF AUTONOMOUS ROBOTICS

Yoink a package: from source

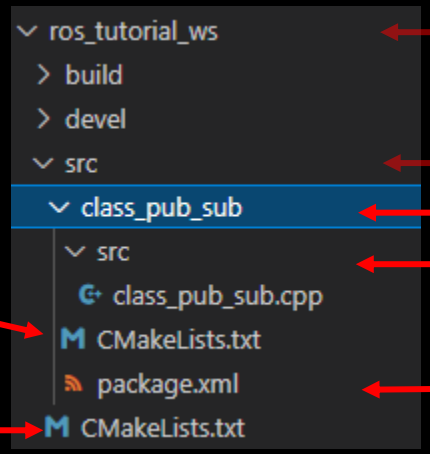
1. Find a repo online
2. Clone it locally
   $ git clone [github link]
3. Copy package into source space
   $ cp -r [path to package] [source path]

For the more adventurous:
https://github.com/ros-drivers/video_stream_opencv.git



This Package

https://github.com/methylDragon/ros-tutorials.git



Workspace

Source space

Package

Source code folder

Manifest (properties)

Package Make file

Toplevel Make file

SUTD ORGANIZATION OF AUTONOMOUS ROBOTICS

Get familiar with it

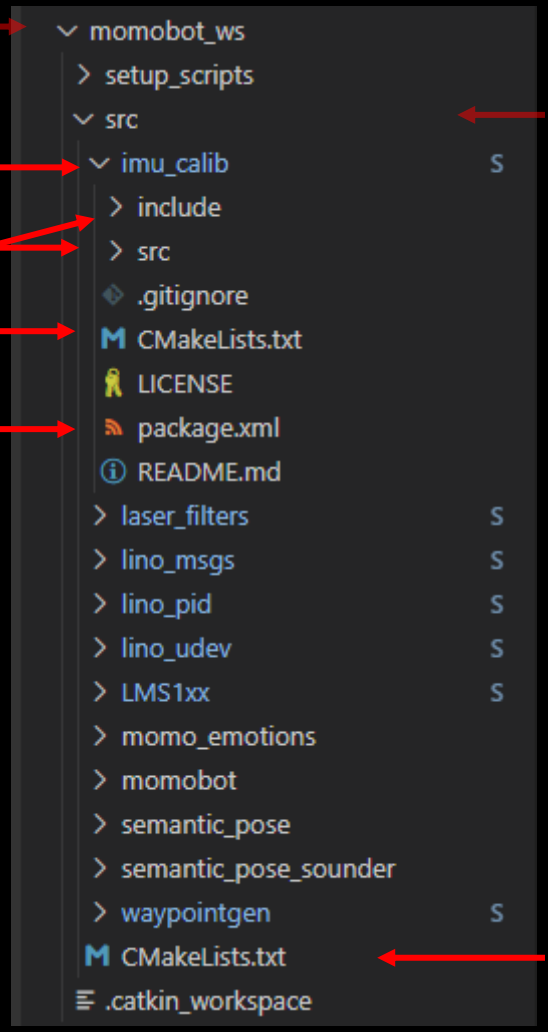
Workspace

Package

Source code

Make file

Manifest
(properties)



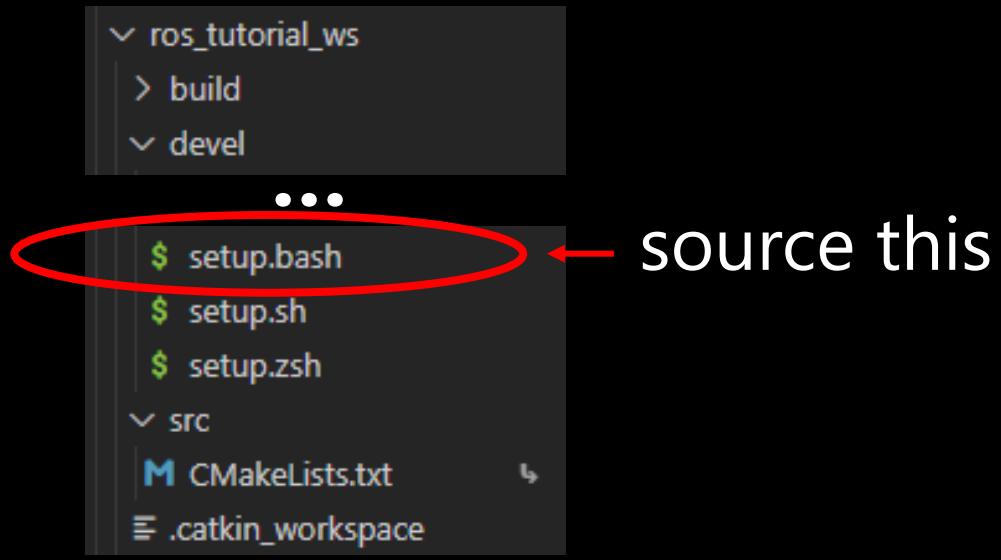
Source
space

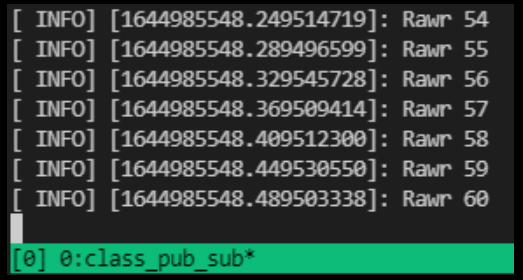
Toplevel
CMake

SUTD ORGANIZATION OF
AUTONOMOUS ROBOTICS

Build the workspace

1. Install dependencies
   $ sudo rosdep install --from-paths src
2. Build
   $ catkin_make # at workspace directory
3. Source
   $ source devel/setup.bash
4. chmod, edit shebang if needed
5. Run it!
   $ roscore
   $ rosrun [package] [node name]

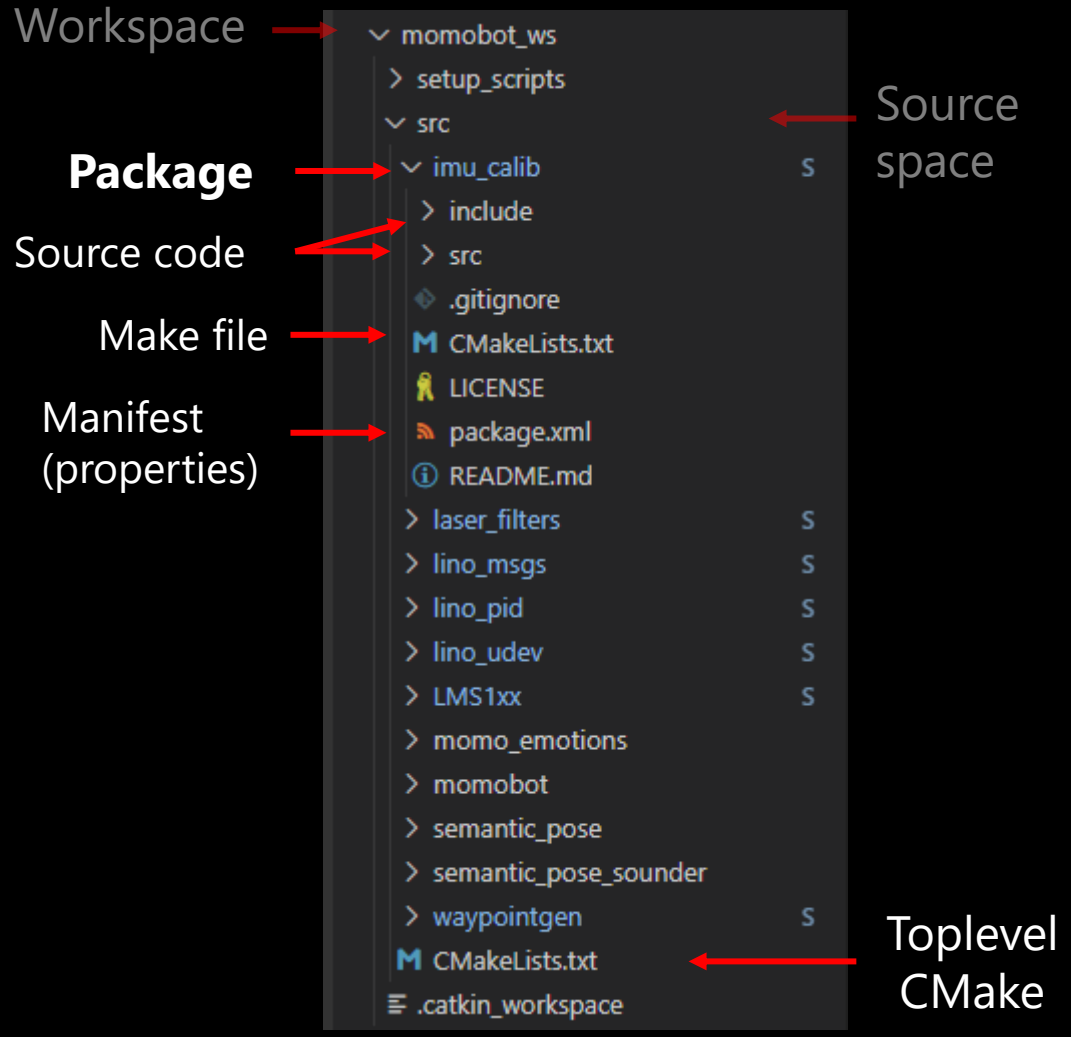




Package: video_stream_opencv
Node name: class_pub_sub

Package: video_stream_opencv
Script name: test_video_resource.py
Args: http://10.21.138.115:8080/video

Recap



Setup workspace
- Make a folder
- catkin_make

Yoink a package
- Clone locally
- Copy package to source space

Use the package
- catkin_make
- Source it

SUTD ORGANIZATION OF AUTONOMOUS ROBOTICS

#3 Write something

rospy

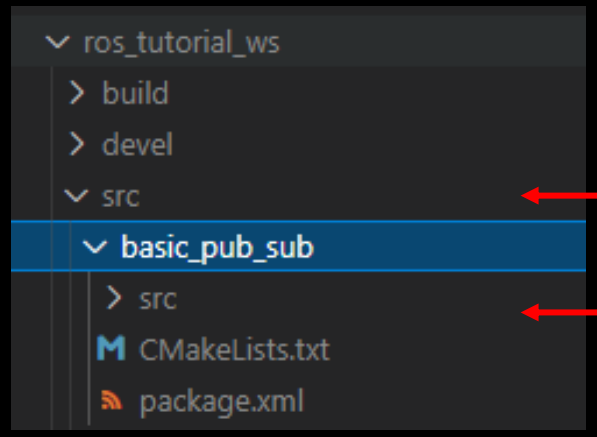
Create your package

1. Create the package

   $ catkin_create_pkg [package name] [dependencies]

   Package name: basic_pub_sub

   Dependencies: rospy



Workspace level source space

Package level source space

SUTD ORGANIZATION OF AUTONOMOUS ROBOTICS

Recap

Modules

- Nodes: Software Part

Comms

- Msgs: Typed data structure
- Topics: For nodes to read / write to, defined by Msgs
- Services: For nodes to request / respond to one another
- Params: Get/Set/Save/Load variables in nodes

SUTD ORGANIZATION OF AUTONOMOUS ROBOTICS

Mapping from graph to code

- Nodes
- Messages
- Topics
- Services
- Params

- Node
  - Publisher (specify topic)
  - Subscriber (specify topic)
  - Service (specify service)

- Interfaces
  - .msg
  - .srv

Your first publisher!

- Refer to given script
- Remember to build the packages when you are done!

```
[INFO] [1645039572.670441]: hello world 1645039572.6702573
[INFO] [1645039572.770479]: hello world 1645039572.7702954
[INFO] [1645039572.870412]: hello world 1645039572.870225
[INFO] [1645039572.970479]: hello world 1645039572.9702952
[INFO] [1645039573.070427]: hello world 1645039573.0702453
[INFO] [1645039573.170760]: hello world 1645039573.1704578
[INFO] [1645039573.270463]: hello world 1645039573.2702737
[INFO] [1645039573.370494]: hello world 1645039573.3703032
[INFO] [1645039573.470384]: hello world 1645039573.4702172
[INFO] [1645039573.570411]: hello world 1645039573.5702302
[INFO] [1645039573.670440]: hello world 1645039573.6702557
[INFO] [1645039573.770511]: hello world 1645039573.770321
[INFO] [1645039573.870455]: hello world 1645039573.8702815
[INFO] [1645039573.970562]: hello world 1645039573.9703417
[INFO] [1645039574.070623]: hello world 1645039574.0703835
```

Your first subscriber!

- Refer to given script
- Remember to build the packages when you are done!

```
[INFO] [1645039572.670441]: hello world 1645039572.6702573
[INFO] [1645039572.770479]: hello world 1645039572.7702954
[INFO] [1645039572.870412]: hello world 1645039572.870225
[INFO] [1645039572.970479]: hello world 1645039572.9702952
[INFO] [1645039573.070427]: hello world 1645039573.0702453
[INFO] [1645039573.170760]: hello world 1645039573.1704578
[INFO] [1645039573.270463]: hello world 1645039573.2702737
[INFO] [1645039573.370494]: hello world 1645039573.3703032
[INFO] [1645039573.470384]: hello world 1645039573.4702172
[INFO] [1645039573.570411]: hello world 1645039573.5702302
[INFO] [1645039573.670440]: hello world 1645039573.6702557
[INFO] [1645039573.770511]: hello world 1645039573.770321
[INFO] [1645039573.870455]: hello world 1645039573.8702815
[INFO] [1645039573.970562]: hello world 1645039573.9703417
[INFO] [1645039574.070623]: hello world 1645039574.0703835
```

```
039573.3703032
[INFO] [1645039573.473065]: /listener_9498_1645039571916 I heard hello world 1645
039573.4702172
[INFO] [1645039573.572800]: /listener_9498_1645039571916 I heard hello world 1645
039573.5702302
[INFO] [1645039573.672837]: /listener_9498_1645039571916 I heard hello world 1645
039573.6702557
[INFO] [1645039573.773074]: /listener_9498_1645039571916 I heard hello world 1645
039573.770321
[INFO] [1645039573.873298]: /listener_9498_1645039571916 I heard hello world 1645
039573.8702815
[INFO] [1645039573.973317]: /listener_9498_1645039571916 I heard hello world 1645
039573.9703417
[INFO] [1645039574.073954]: /listener_9498_1645039571916 I heard hello world 1645
039574.0703835
```

SUTD ORGANIZATION OF AUTONOMOUS ROBOTICS

Add a param!

```
# Set a parameter value!
rospy.set_param(‘/param_namespace’, ‘param_name’)
# Get a parameter value!
rospy.get_param(‘/param_namespace')
```

```
[INFO] [1645039737.885032]: hello world omaemo shinderu
[INFO] [1645039737.985012]: hello world omaemo shinderu
[INFO] [1645039738.085401]: hello world omaemo shinderu
[INFO] [1645039738.184440]: hello world omaemo shinderu
[INFO] [1645039738.284631]: hello world omaemo shinderu
[INFO] [1645039738.384541]: hello world nani?
[INFO] [1645039738.484960]: hello world nani?
[INFO] [1645039738.585654]: hello world nani?
[INFO] [1645039738.684923]: hello world nani?
[INFO] [1645039738.784446]: hello world nani?
[INFO] [1645039738.884696]: hello world nani?
[INFO] [1645039738.984518]: hello world nani?
[INFO] [1645039739.084639]: hello world nani?
[INFO] [1645039739.185024]: hello world nani?
[INFO] [1645039739.284822]: hello world nani?
```

```
eard hello world nani?
[INFO] [1645039738.687348]: /listener_9498_1645039571916 I h
eard hello world nani?
[INFO] [1645039738.787046]: /listener_9498_1645039571916 I h
eard hello world nani?
[INFO] [1645039738.887538]: /listener_9498_1645039571916 I h
eard hello world nani?
[INFO] [1645039738.986836]: /listener_9498_1645039571916 I h
eard hello world nani?
[INFO] [1645039739.087126]: /listener_9498_1645039571916 I h
eard hello world nani?
[INFO] [1645039739.188136]: /listener_9498_1645039571916 I h
eard hello world nani?
[INFO] [1645039739.288366]: /listener_9498_1645039571916 I h
eard hello world nani?
```

SUTD ORGANIZATION OF AUTONOMOUS ROBOTICS

Create a message package

1. Create the package

   $ catkin_create_pkg [package name] [dependencies]

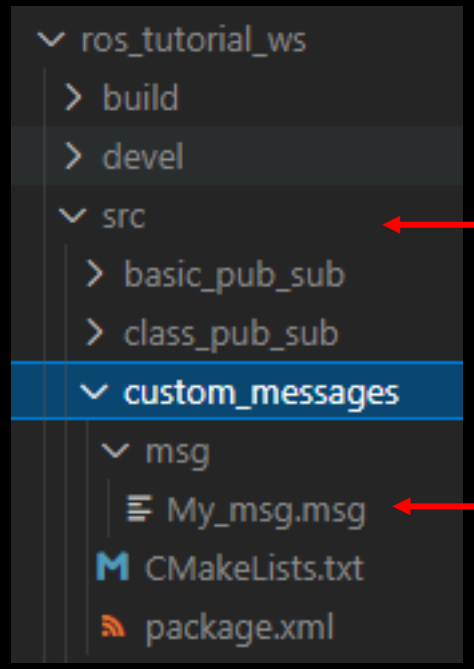
   Package name: custom_messages

   Dependencies: rospy

2. Create msg folder in src

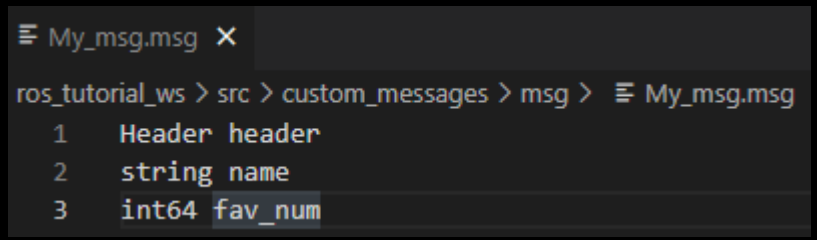
3. Add a custom message definition in a .msg file

```
Header header
string name
uint8 fav_num
```



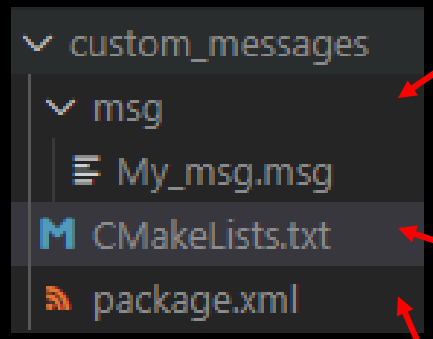
Workspace level source space

Your first msg definition



SUTD ORGANIZATION OF AUTONOMOUS ROBOTICS

CMakeLists and package.xml




```
Header header
string name
uint8 fav_num
```

```
find_package(catkin REQUIRED COMPONENTS
  rospy std_msgs
  message_generation  # <-- Add this!
)
add_message_files(My_msg.msg) # <-- Add this!
catkin_package(CATKIN_DEPENDS message_runtime) # <-- Add this!
```

```
<build_depend>message_generation</build_depend>
<exec_depend>message_runtime</exec_depend>
```

```
root@d7419e84fff0:/usr/src/ros-tu
std_msgs/Header header
  uint32 seq
  time stamp
  string frame_id
string name
uint8 fav_num
```

```
$ rosmsg show basic_pub_sub/My_msg
```

SUTD ORGANIZATION OF AUTONOMOUS ROBOTICS

Your first custom pubsub!

1. Create the package

   $ catkin_create_pkg [package name] [dependencies]

   Package name: custom_messages_pub_sub

   Dependencies: rospy

2. Attach dependencies in CMakeLists and package.xml

   CMakeLists: Add custom_messages in `find_package` and `catkin_package`

   package.xml: Add custom_messages in `build_depend`, `build_export_depend`, `exec_depend`

```
find_package(catkin REQUIRED COMPONENTS
  rospy
  std_msgs
  custom_messages # here!
)

catkin_package(CATKIN_DEPENDS
    custom_messages # here!
)
```

```
<build_depend>rospy</build_depend>
<build_depend>std_msgs</build_depend>
<build_depend>custom_messages</build_depend>

<build_export_depend>rospy</build_export_depend>
<build_export_depend>std_msgs</build_export_depend>
<build_export_depend>custom_messages</build_export_depend>

<exec_depend>rospy</exec_depend>
<exec_depend>std_msgs</exec_depend>
<exec_depend>custom_messages</exec_depend>
```

SUTD ORGANIZATION OF AUTONOMOUS ROBOTICS

Your first custom pubsub!

1. Copy and edit basic_pub and basic_sub from previous package!
   Change import to `from custom_messages.msg import My_msg`
   Change usage of String msg type to My_msg msg type
   Attach / read relevant data types to corresponding data fields
2. Refer to code base if you are really stuck
3. Test and troubleshoot
   Unable to import custom message: fix the dependencies as per previous slide
   Something about message typing: make sure you adhere to your message definition
   Unable to compile: check package name corresponds in CMakeLists and package.xml

SUTD ORGANIZATION OF AUTONOMOUS ROBOTICS

More resources!

- ROS tutorials
- methyldragon ROS tutorial