PROGRAMMING FOR AUTONOMOUS SYSTEMS

Fred Livingston, PhD

Work Shop 001

© 2023 FRED LIVINGSTON. ALL RIGHTS RESERVED.

CONTACT INFO

Fred Livingston, PhD

Email: fjliving@ncsu.edu

Mobile: 919.795.4710

Web: https://livingston.wordpress.ncsu.edu/

Bitbucket: https://bitbucket.org/livingston_ai/

© 2023 FRED LIVINGSTON. ALL RIGHTS RESERVED.

SPRING 2023 WORKSHOP SERIES

- WS 001 – Introduction to Robot Programming using ROS2 [Feb 17th, 2023]
- WS 002 – Navigation [March 3rd, 2023]
- WS 003– Reactive Behaviors [TBD]

© 2023 FRED LIVINGSTON. ALL RIGHTS RESERVED.

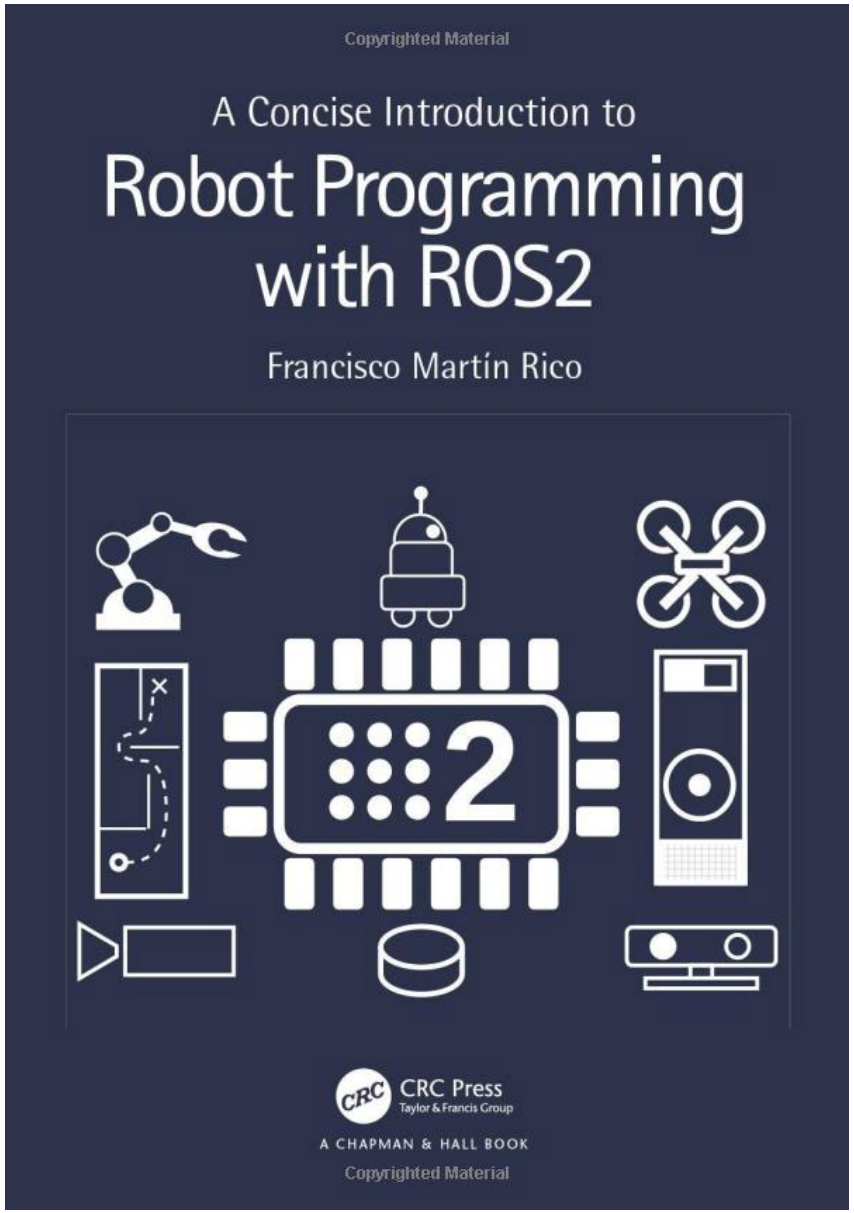
ROBOT PROGRAMMING USING ROS2

- Introduction to ROS2
- ROS Client Layer
- Simulation Tools
- robot program: architectures
- Task1: Basic control of ground vehicle

© 2023 FRED LIVINGSTON. ALL RIGHTS RESERVED.

RESOURCES

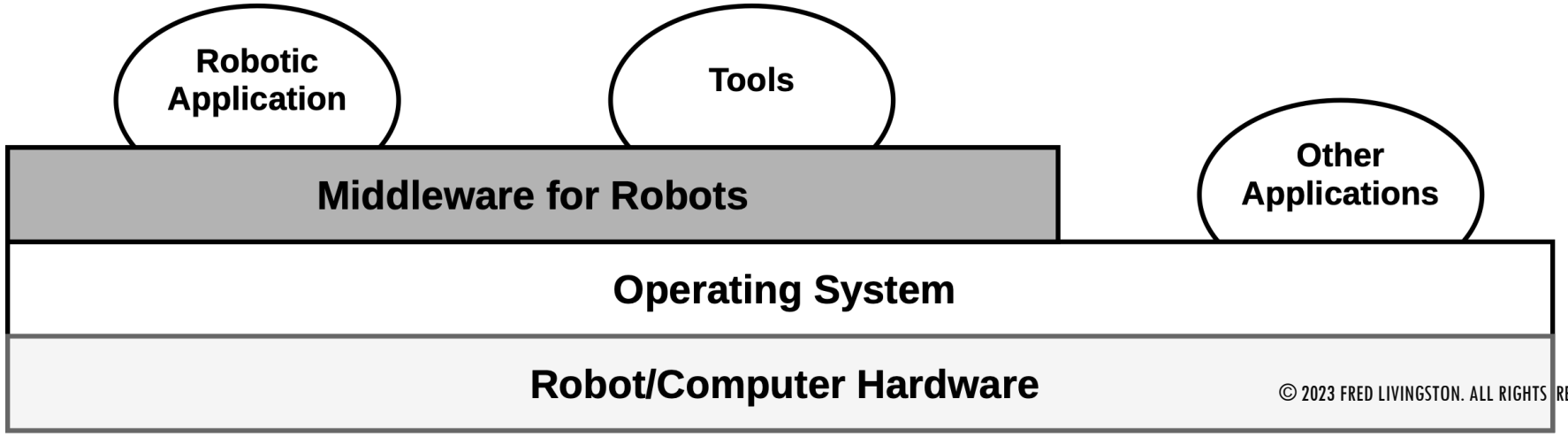
- ROS/2 Tutorials
  - http://docs.ros.org/en/rolling/Tutorials.html
  - https://roboticsbackend.com/category/ros2/
  - https://www.theconstructsim.com/robotigniteacademy_learnros/ros-courses-library/



© 2023 FRED LIVINGSTON. ALL RIGHTS RESERVED.

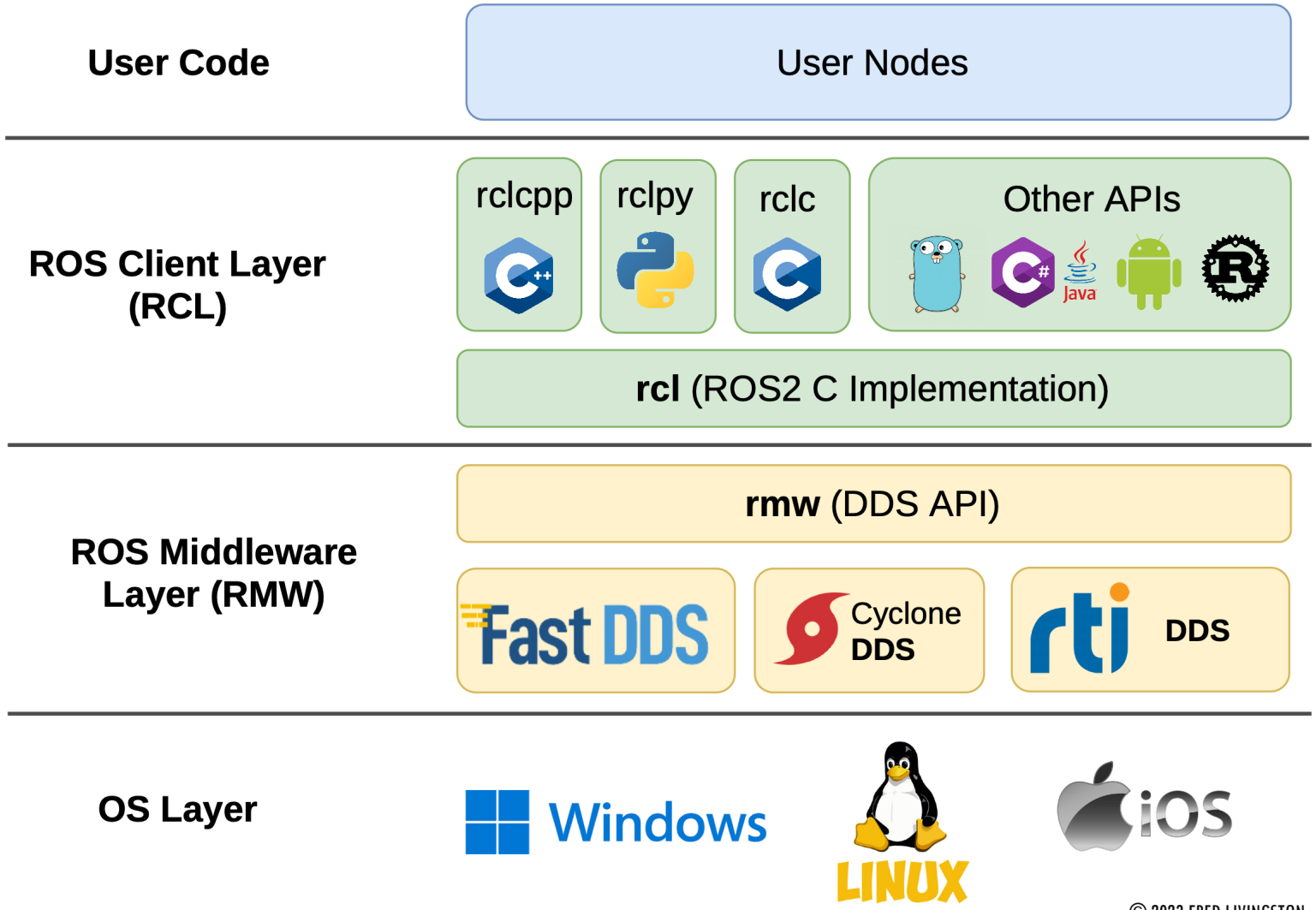
PROGRAMMING ROBOTS

- Robots must be programmed to be useful
- We need Middlewares
- Robot programming middlewares provide drivers, libraries, and methodologies
- Few of them have survived the robot for which they were designed or have expanded from the laboratories where they were implemented
- The big difference is the ROS developers community around the world.



© 2023 FRED LIVINGSTON. ALL RIGHTS RESERVED.

ROS2 DESIGN



© 2023 FRED LIVINGSTON. ALL RIGHTS RESERVED.

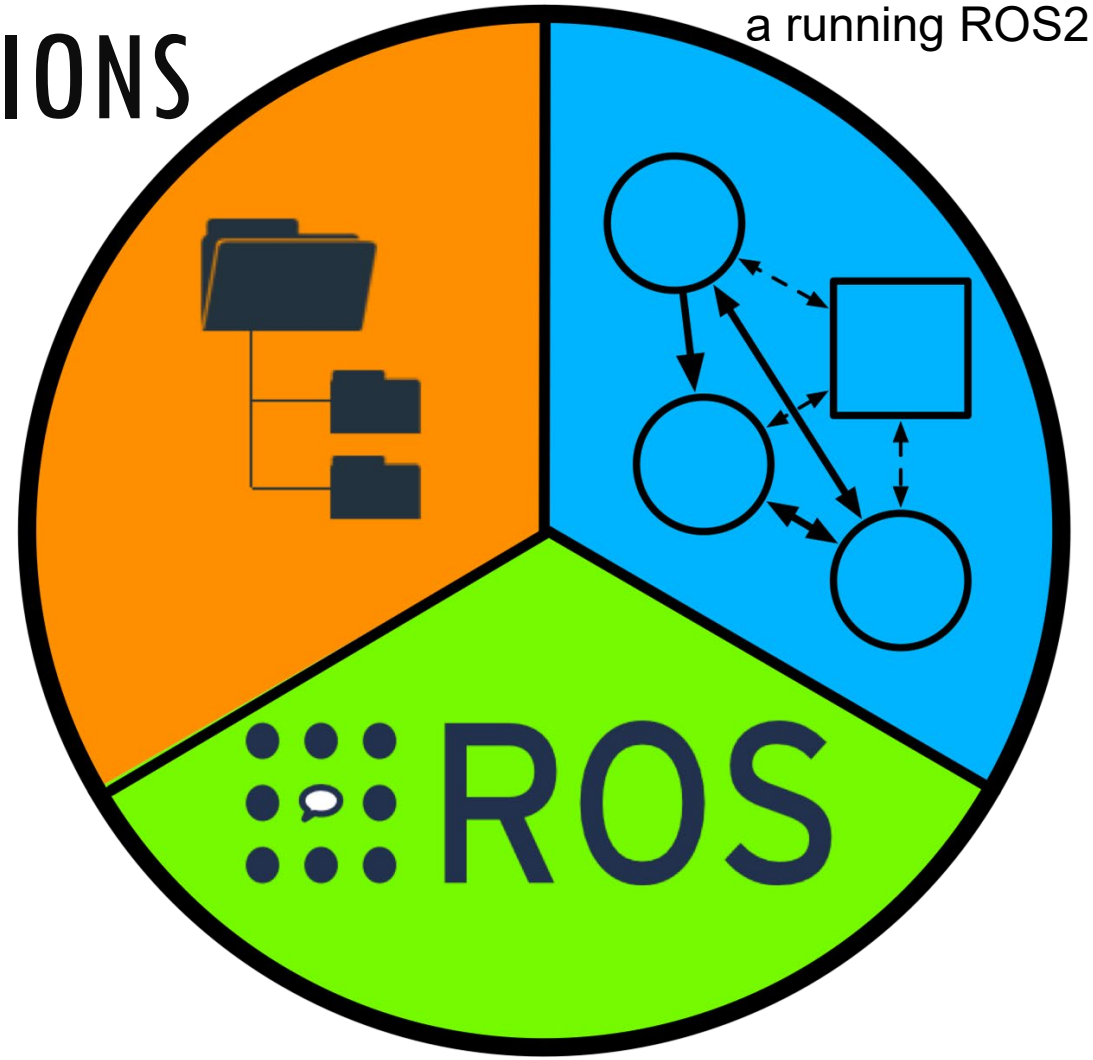
ROS DIMENSIONS

Computation Graph:

a running ROS2 application

Workspace:

the set of software installed on the robot or computer, the programs that the user develops, and tools to build

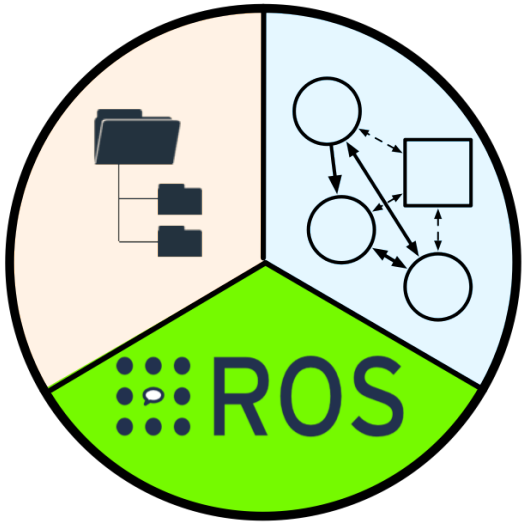


Community: vast community of developers who con- tribute with their own applications and utilities through public repositories, to which other developers can contribute

© 2023 FRED LIVINGSTON. ALL RIGHTS RESERVED.

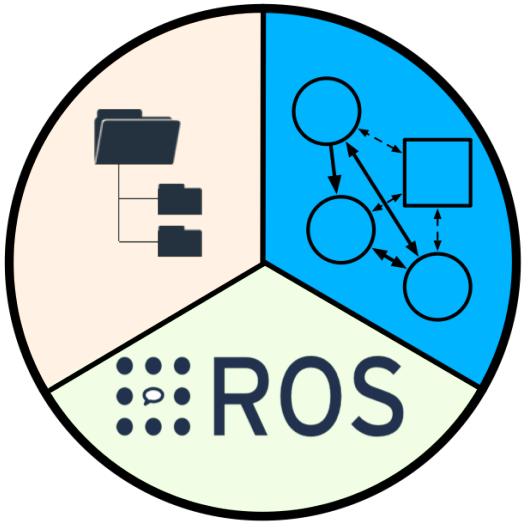
THE COMMUNITY

- Open Source and Licenses
- ROS2 organizes software development in federal model
- Packages and distributions
- Online resources



© 2023 FRED LIVINGSTON. ALL RIGHTS RESERVED.

THE COMPUTATION GRAPH

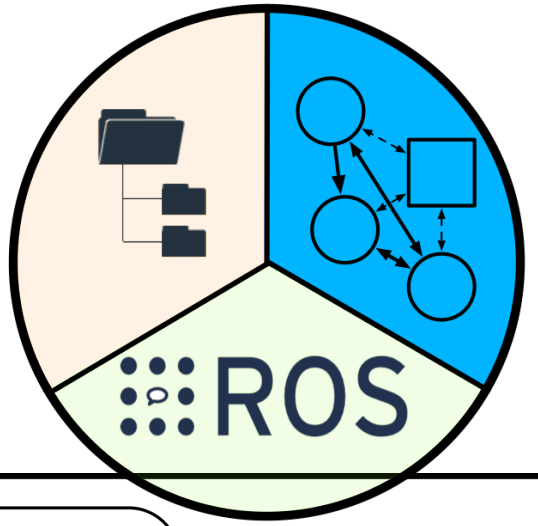


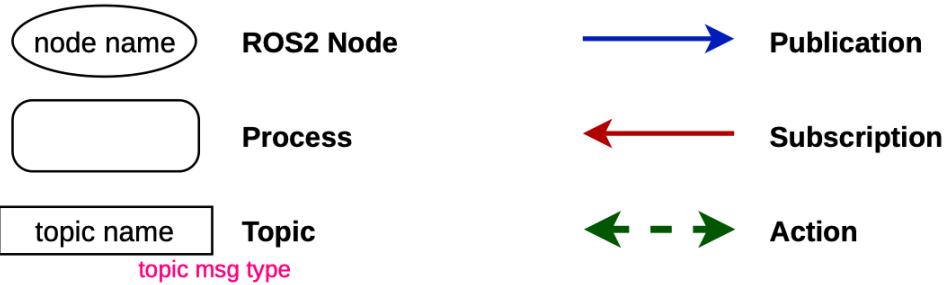
- A robot's software looks like during its execution.
- A Computation Graph contains ROS2 nodes that communicate with each other so that the robot can carry out some tasks.
- The logic of the application is in the nodes, as the primary elements of execution in ROS2.
- Communication mechanisms:
- **Publication/Subscription**: Asynchronous N:M
- **Services**: Synchronous 1:1
- **Actions**: Asynchronous 1:1

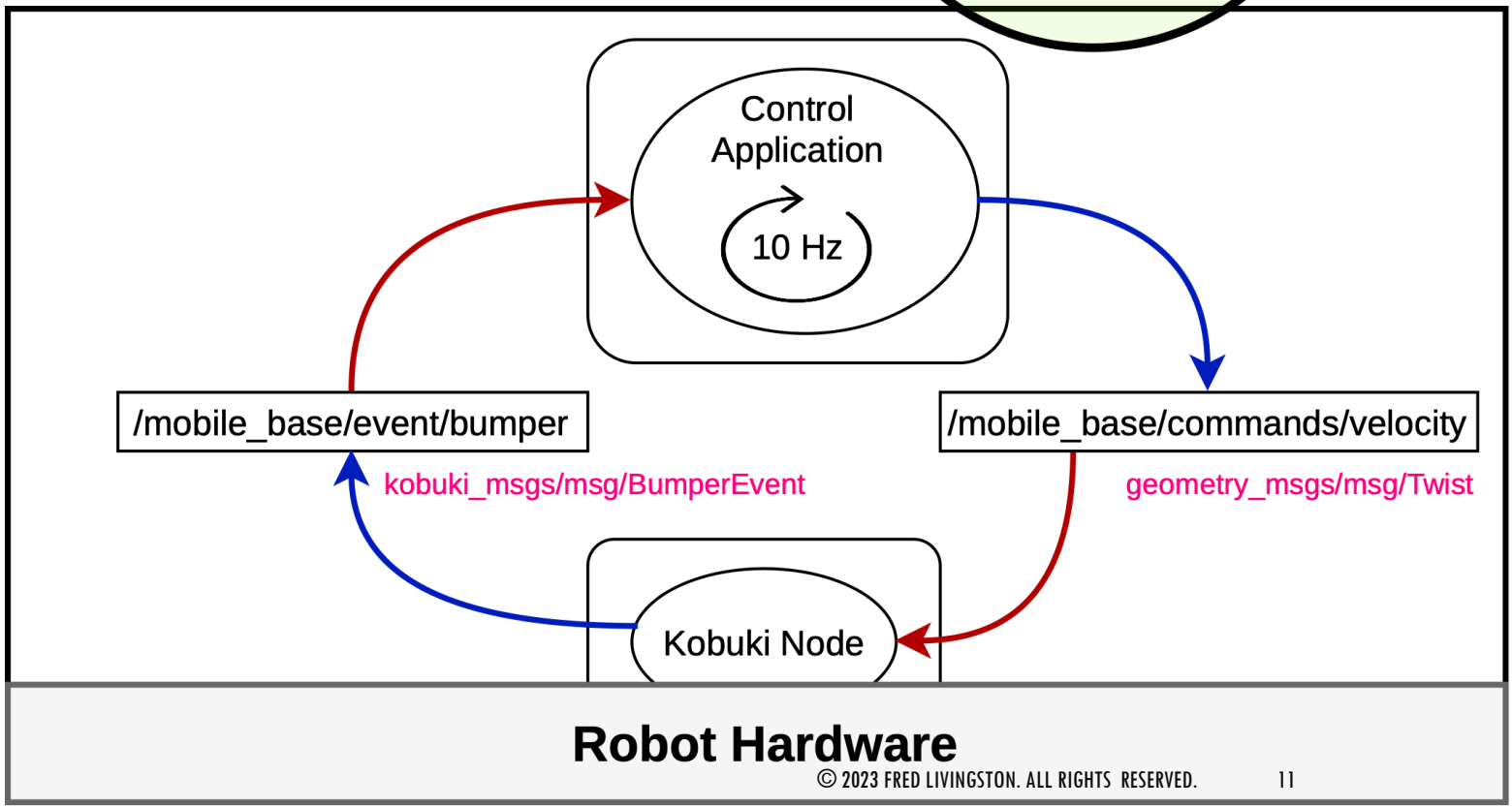
© 2023 FRED LIVINGSTON. ALL RIGHTS RESERVED.

THE COMPUTATION GRAPH

- Execution model
  - Iterative
  - Event-Oriented

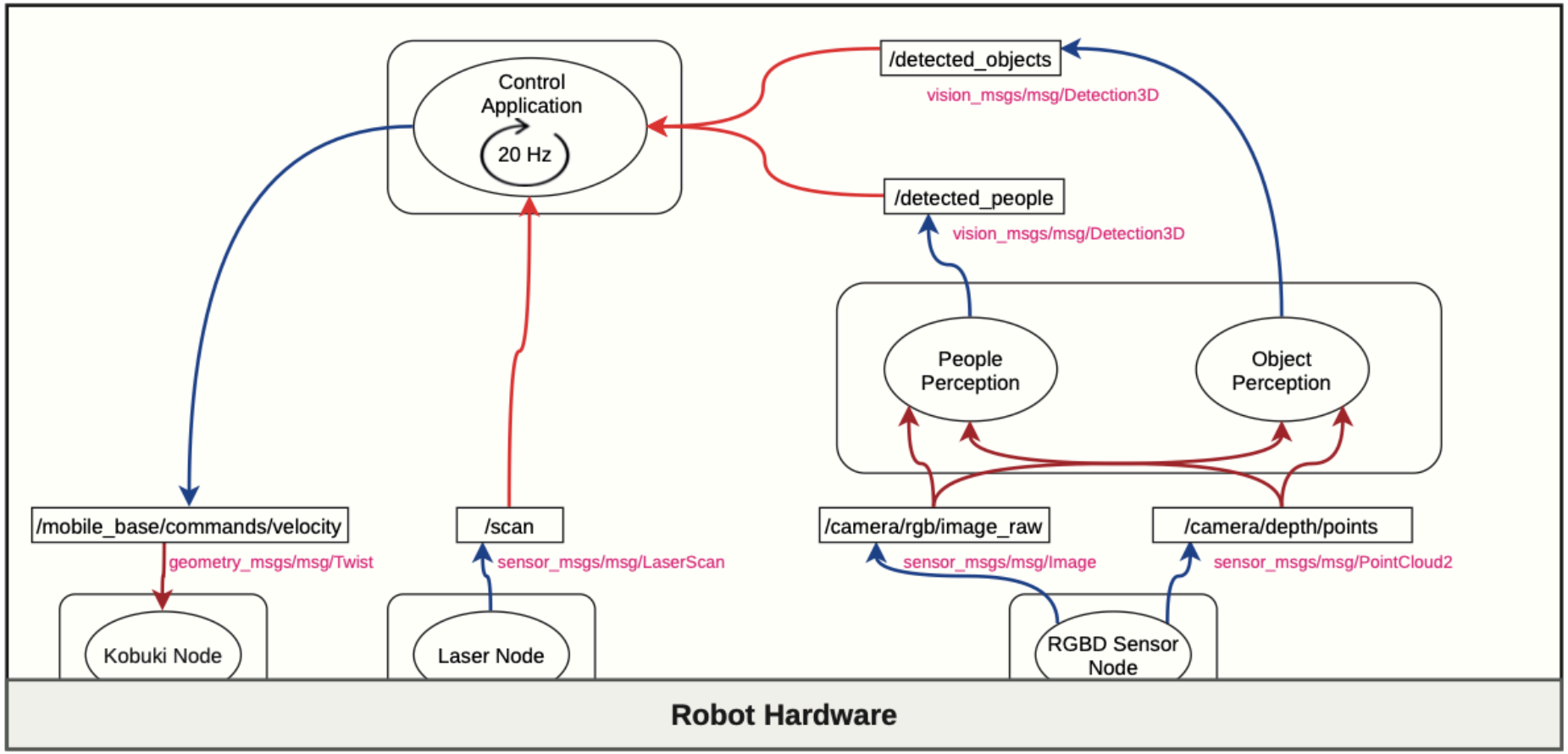






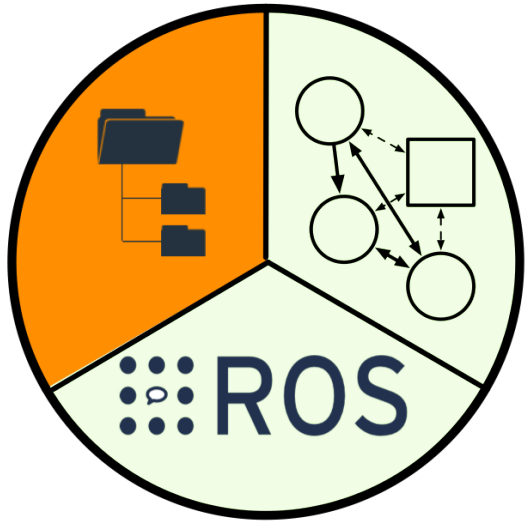
© 2023 FRED LIVINGSTON. ALL RIGHTS RESERVED.

THE COMPUTATION GRAPH - EXAMPLE



© 2023 FRED LIVINGSTON. ALL RIGHTS RESERVED.

THE WORKSPACE



- Approaches ROS2 software from a static point of view.
- Where the ROS2 software is installed, organized, and all the tools and processes that allow us to launch a computing graph.
- This includes the build system and node startup tools.
- Elements:
    - **Package**:
        - It is the minimum functional set of software.
        - Contains executables, libraries, or message definitions with a common purpose.
    - **Workspace**:
        - A directory that contains packages.
        - Activable to be available to use.
- ***Underlay*** y ***overlay***

© 2023 FRED LIVINGSTON. ALL RIGHTS RESERVED.

ROS2::FOXY INSTALATION

- VirtualBox (Option 1)
- https://www.virtualbox.org/wiki/Downloads

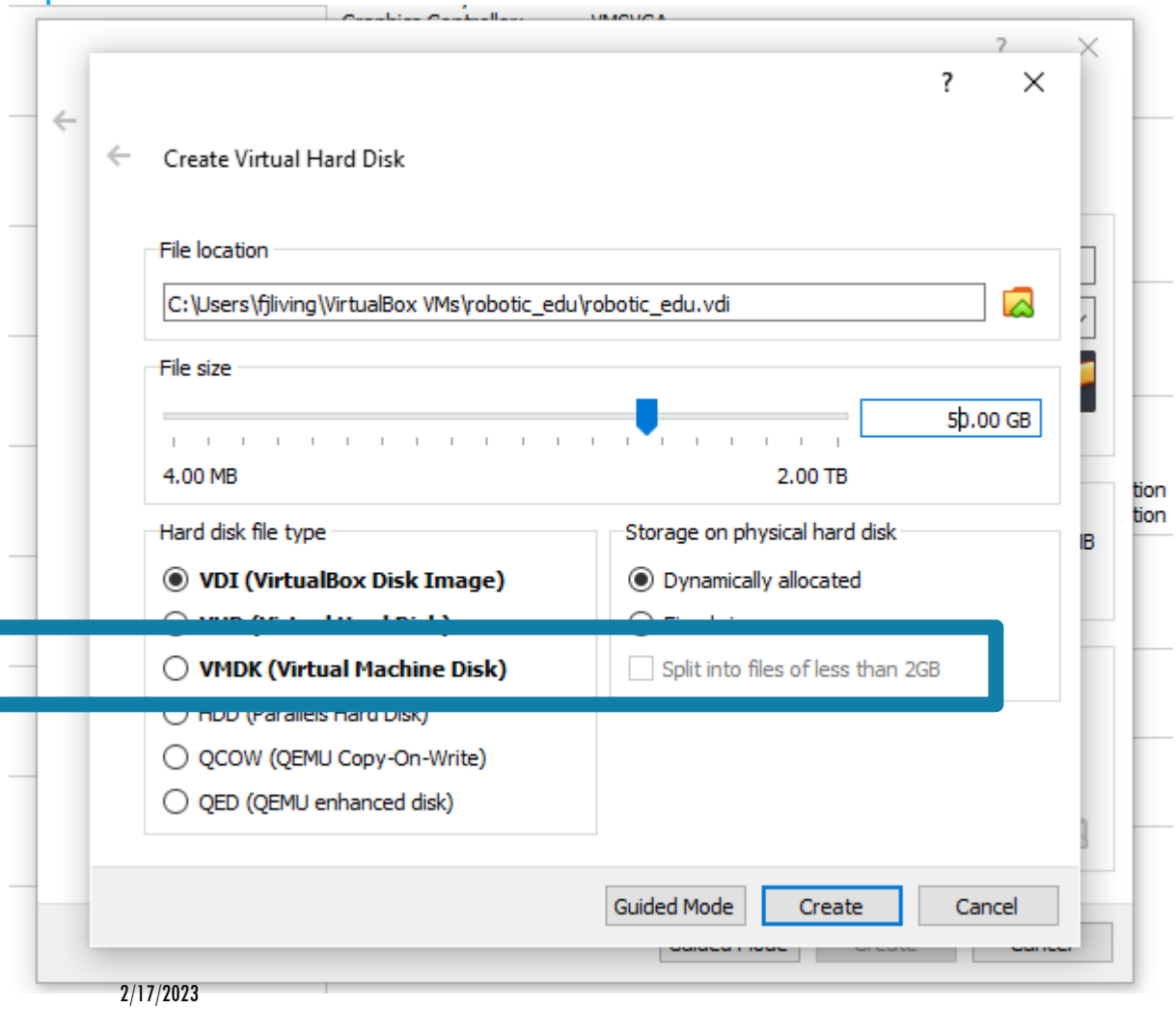
- **VMWare (Option 2 – Preferred)**
- https://www.vmware.com/products/workstation-player/workstation-player-evaluation.html

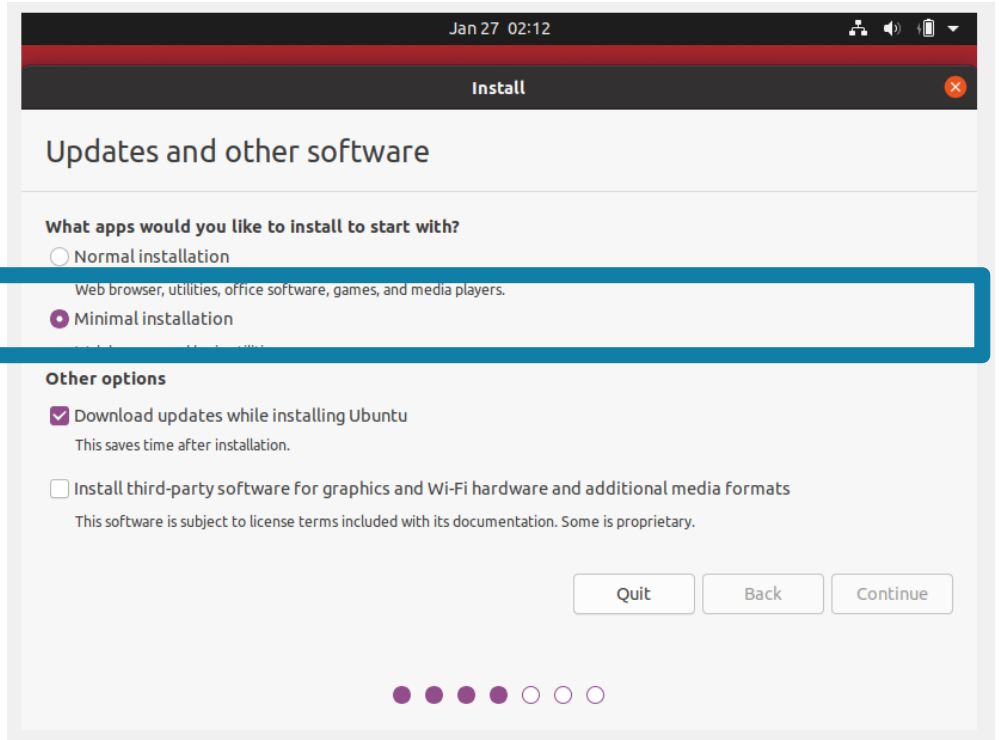
- Ubuntu **20.04**
- https://releases.ubuntu.com/22.04/ubuntu-22.04.1-desktop-amd64.iso

- ROS2::**Foxy** (LTS)
- https://docs.ros.org/en/foxy/Installation/Ubuntu-Install-Debians.html

© 2023 FRED LIVINGSTON. ALL RIGHTS RESERVED.

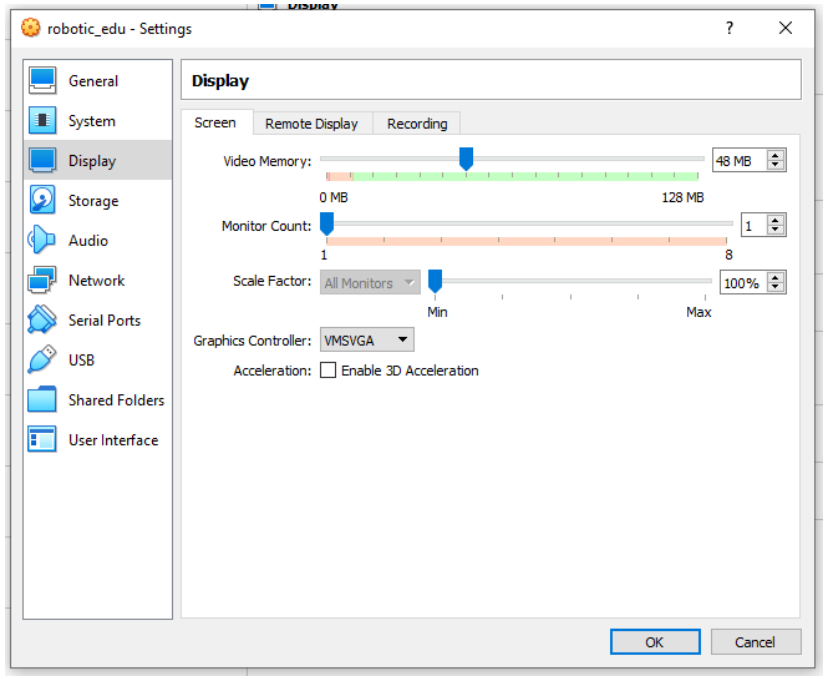
VM – RECOMMENDED SETTINGS

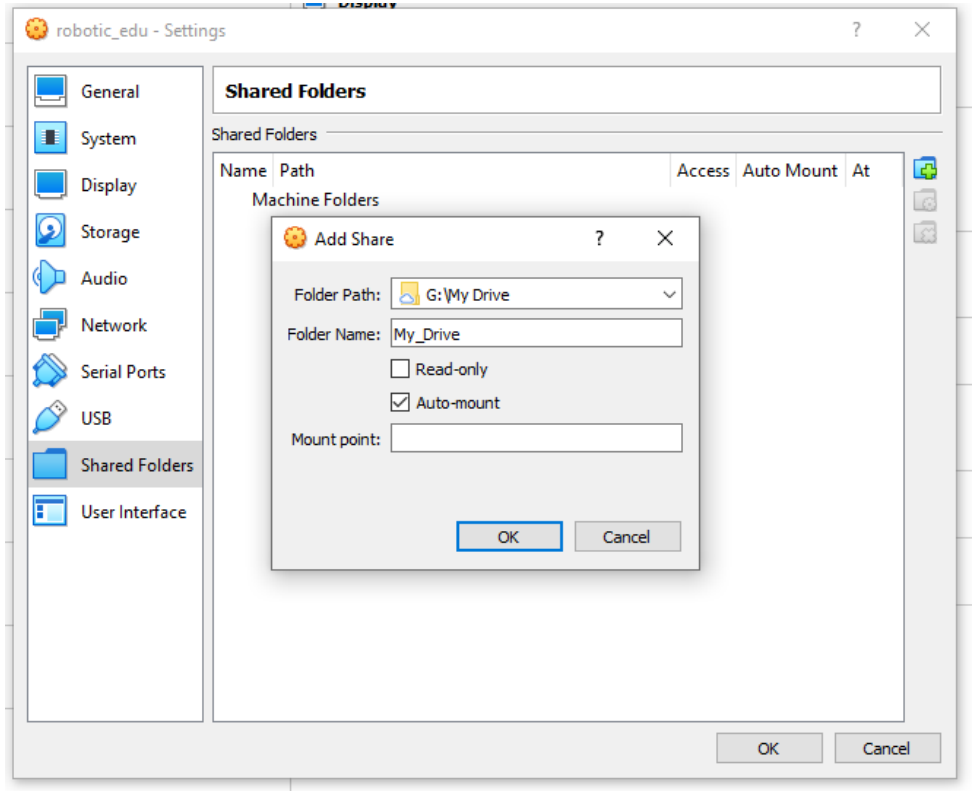




© 2023 FRED LIVINGSTON. ALL RIGHTS RESERVED.

VM – RECOMMENDED SETTINGS

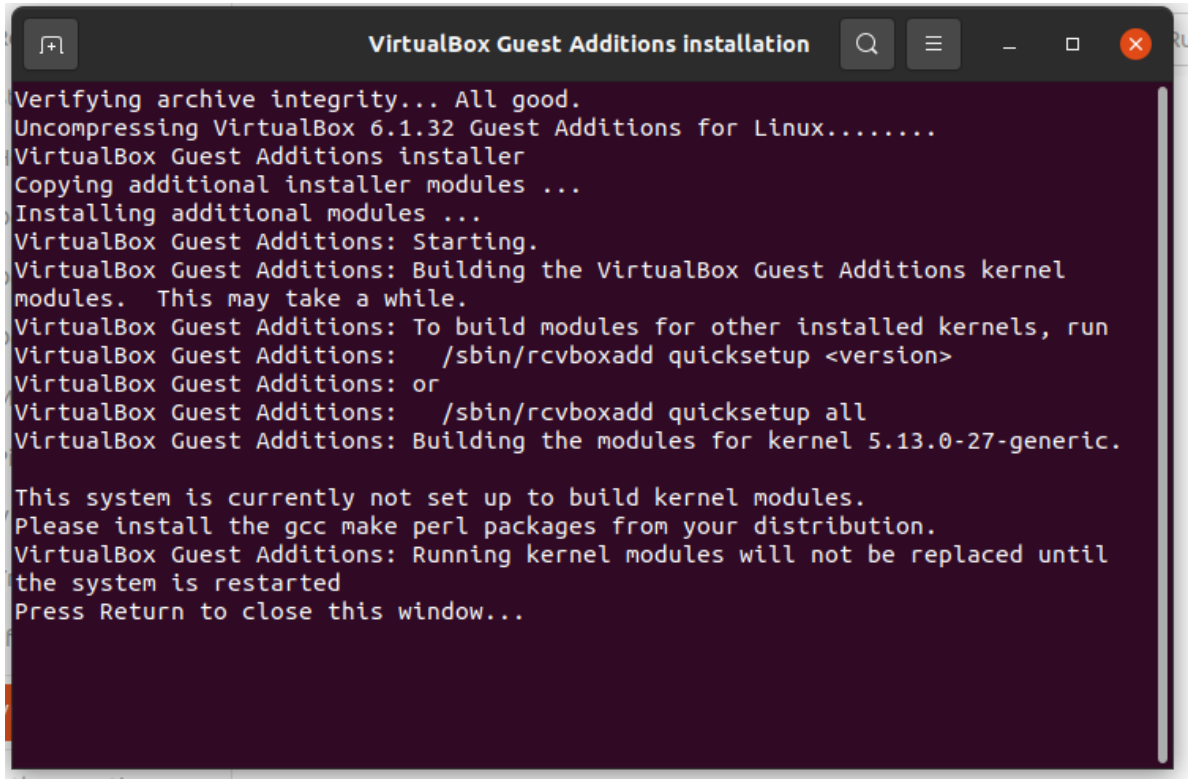


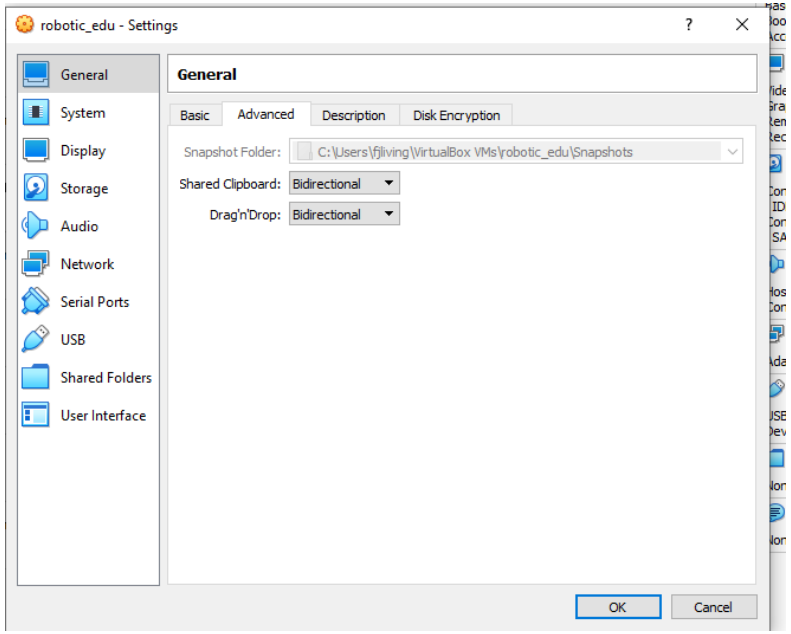


- Make sure 3D Acceleration is Disabled

© 2023 FRED LIVINGSTON. ALL RIGHTS RESERVED.

VM – RECOMMENDED SETTINGS

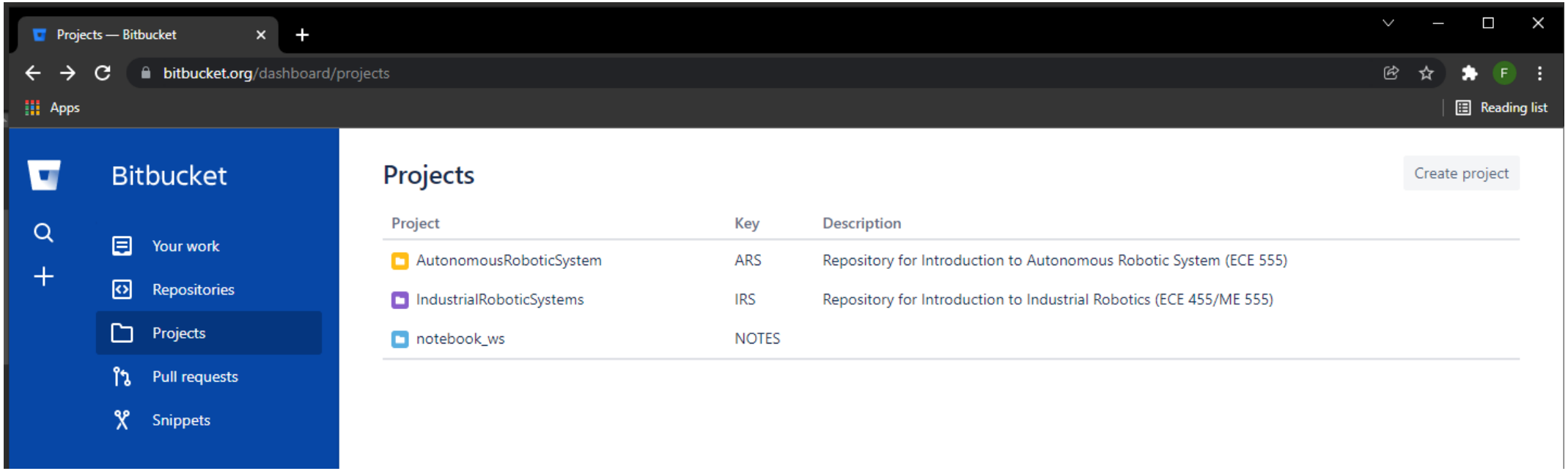




- Install Guest Additions

© 2023 FRED LIVINGSTON. ALL RIGHTS RESERVED.

BITBUCKET



- https://bitbucket.org/livingston_ai/

© 2023 FRED LIVINGSTON. ALL RIGHTS RESERVED.

PYTHON 2 UNINSTALL (OPTIONAL)

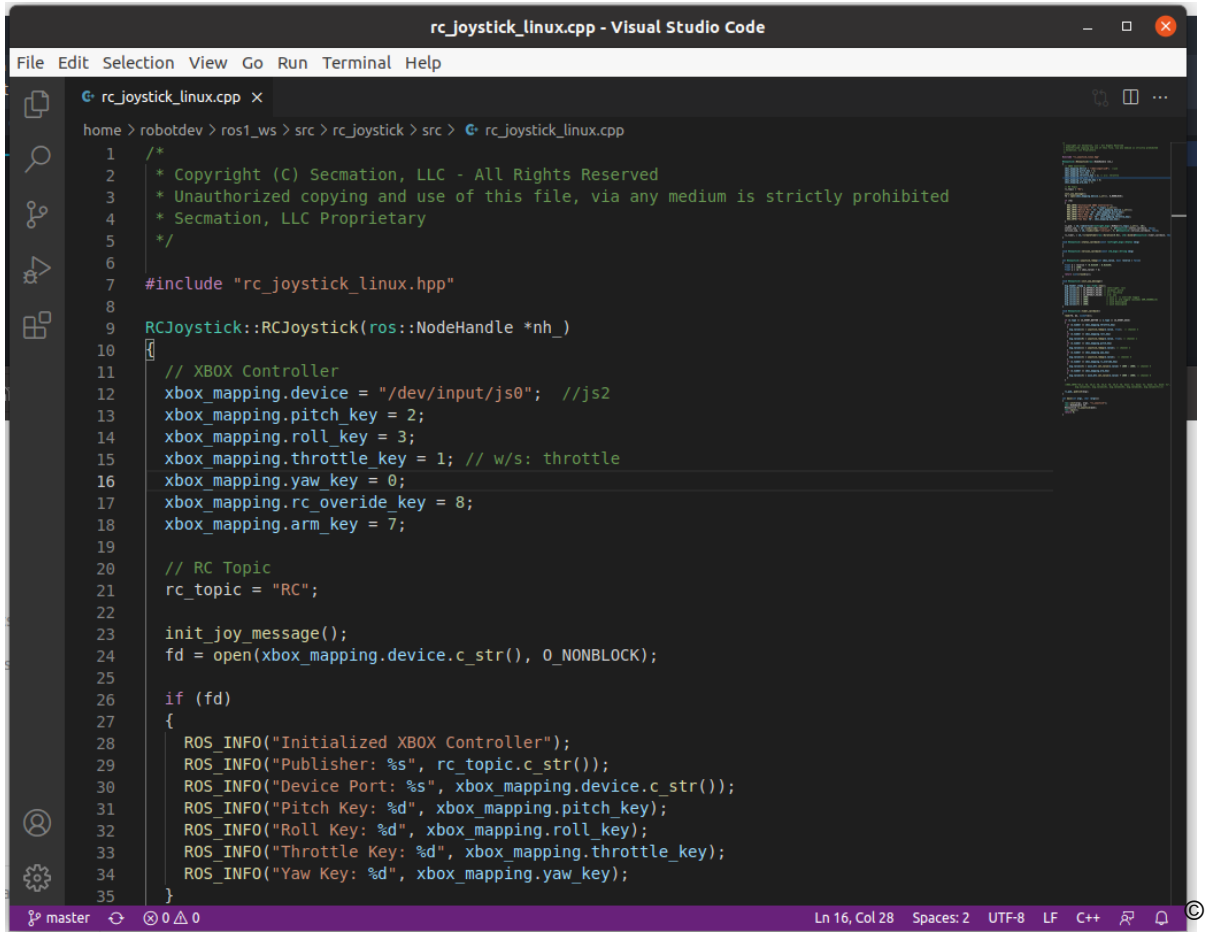
- https://askubuntu.com/questions/1242702/how-to-remove-python-2-from-ubuntu-20-04

- $sudo apt-get install python-is-python3
- $sudo apt-get autoremove --purge

© 2023 FRED LIVINGSTON. ALL RIGHTS RESERVED.

VSCODE (OPTIONAL)

- $sudo snap install code --classic



© 2023 FRED LIVINGSTON. ALL RIGHTS RESERVED.

ROS2::FOXY INSTALLATION

- https://bitbucket.org/livingston_ai/ros2_robot_programming/src/master/ros2_foxy_setup.ipynb?viewer=nbviewer

© 2023 FRED LIVINGSTON. ALL RIGHTS RESERVED.

ROS 2 Cheats Sheet

Command Line Interface

All ROS 2 CLI tools start with the prefix 'ros2' followed by a command, a verb and (possibly) positional/optional arguments.

For any tool, the documentation is accessible with,

```
$ ros2 command --help
```

and similarly for verb documentation,

```
$ ros2 command verb -h
```

Similarly, auto-completion is available for all commands/verbs and most positional/optional arguments. E.g.,

```
$ ros2 command [tab][tab]
```

Some of the examples below rely on:
ROS 2 demos package.

---

action Allows to manually send a goal and displays debugging information about actions.

Verbs:

info	Output information about an action.
list	Output a list of action names.
send_goal	Send an action goal.
show	Output the action definition.

Examples:

```
$ ros2 action info /fibonacci
$ ros2 action list
$ ros2 action send_goal /fibonacci \
 action_tutorials/action/Fibonacci "order: 5"
$ ros2 action show action_tutorials/action/Fibonacci
```

---

bag Allows to record/play topics to/from a rosbag.

Verbs:

info	Output information of a bag.
play	Play a bag.
record	Record a bag.

---

list	Output a list of running containers and components.
load	Load a component into a container node.
standalone	Run a component into its own standalone container node.
types	Output a list of components registered in the ament index.
unload	Unload a component from a container node.

Examples:

```
$ ros2 component list
$ ros2 component load /ComponentManager \
 composition composition::Talker
$ ros2 component types
$ ros2 component unload /ComponentManager 1
```

---

daemon Various daemon related verbs.

Verbs:

start	Start the daemon if it isn't running.
status	Output the status of the daemon.
stop	Stop the daemon if it is running

---

doctor A tool to check ROS setup and other potential issues such as network, package versions, rmw middleware etc.

Alias: **wtf** (where's the fire).

Arguments:

--report/-r	Output report of all checks.
--report-fail/-rf	Output report of failed checks only.
--include-warning/-iw	Include warnings as failed checks.

Examples:

```
$ ros2 doctor
$ ros2 doctor --report
$ ros2 doctor --report-fail
$ ros2 doctor --include-warning
```

---

interface Various
related verbs. Inte
the following optio
srvs'.

Verbs:

list	Lis
package	Ou wi
packages	Ou te
proto	Pr fa
show	Ou

Examples:

```
$ ros2 interface
$ ros2 interface
$ ros2 interface
$ ros2 interface
$ ros2 interface
```

---

launch Allows to
without to 'cd' the

Usage:

```
$ ros2 launch <
```

Example:

```
$ ros2 launch de
```

---

lifecycle Various

Verbs:

get	Get li
list	Outp
nodes	Outp
set	Trigg

---

msg (deprecated
messages.

Verbs:

© 2023 FRED LIVINGSTON. ALL RIGHTS RESERVED.

```
$ ros2 msg list
$ ros2 msg package std_msgs
$ ros2 msg packages
$ ros2 msg show geometry_msgs/msg/Pose
```

---

multicast Various multicast related verbs.

Verbs:

receive	Receive a single UDP multicast packet.
send	Send a single UDP multicast packet.

---

node Displays debugging information about nodes.

Verbs:

info	Output information about a node.
list	Output a list of available nodes.

Examples:

```
$ ros2 node info /talker
$ ros2 node list
```

---

param Allows to manipulate parameters.

Verbs:

delete	Delete parameter.
describe	Show descriptive information about declared parameters.
dump	Dump the parameters of a given node in yaml format, either in terminal or in a file.
get	Get parameter.
list	Output a list of available parameters.
set	Set parameter

Examples:

```
$ ros2 param delete /talker /use_sim_time
$ ros2 param get /talker /use_sim_time
$ ros2 param list
$ ros2 param set /talker /use_sim_time false
```

---

pkg Create a ros2 package or output package(s)-related information.

Verbs:

create	Create a new ROS2 package.

```
$ ros2 pkg executables demo_nodes_cpp
$ ros2 pkg list
$ ros2 pkg prefix std_msgs
$ ros2 pkg xml -t version
```

---

run Allows to run an executable in an arbitrary package without having to 'cd' there first.

Usage:

```
$ ros2 run <package> <executable>
```

Example:

```
$ ros2 run demo_node_cpp talker
```

---

security Various security related verbs.

Verbs:

create_key	Create key.
create_permission	Create keystore.
generate_artifacts	Create permission.
list_keys	Distribute key.
create_keystore	Generate keys and permission files from a list of identities and policy files.
distribute_key	Generate XML policy file from ROS graph data.
generate_policy	List keys.

Examples (see sros2 package):

```
$ ros2 security create_key demo_keys /talker
$ ros2 security create_permission demo_keys /talker  \
 policies/sample_policy.xml
$ ros2 security generate_artifacts
$ ros2 security create_keystore demo_keys
```

---

service Allows to manually call a service and displays debugging information about services.

Verbs:

call	Call a service.
find	Output a list of services of a given type.
list	Output a list of service names.

---

srv (deprecated)

Verbs:

list	Ou
package	Ou wi
packages	Ou se
show	Ou

---

test Run a ROS2

---

topic A tool for d topics, including i and messages.

Verbs:

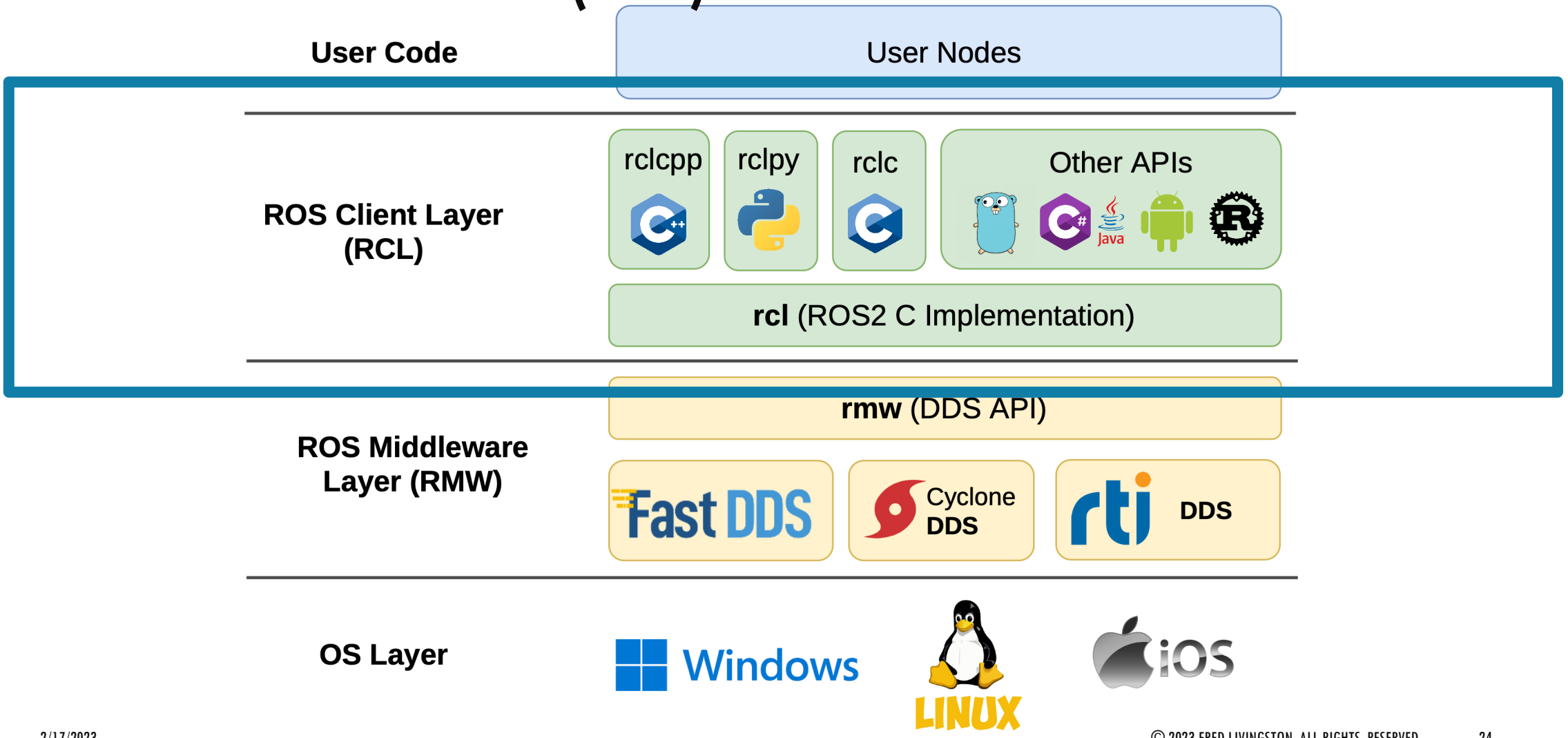
bw	Displa
delay	Displa header
echo	Outpu
find	Find t
hz	Displa
info	Outpu
list	Outpu
pub	Publis
type	Outpu

Examples:

```
$ ros2 topic bw
$ ros2 topic ech
$ ros2 topic finc
$ ros2 topic hz
$ ros2 topic infc
$ ros2 topic list
$ ros2 topic pub
 'data: Hello RC
$ ros2 topic typ
```


© 2023 FRED LIVINGSTON. ALL RIGHTS RESERVED.

ROS CLIENT LAYER (RCL)



© 2023 FRED LIVINGSTON. ALL RIGHTS RESERVED.

```
$ ros2

usage: ros2 [-h] Call ‘ros2 <command> -h‘ for more detailed usage. ...
ros2 is an extensible command-line tool for ROS 2.

...
```

```
ros2 <command> <verb> [<params>|<option>]*
```

```
action      extension_points  node        test
bag         extensions        param       topic
component   interface         pkg         wtf
launch      run               daemon      lifecycle
security    doctor            multicast   service
```

Further readings:

- https://github.com/ros2/ros2cli
- https://github.com/ubuntu-robotics/ros2_cheats_sheet/blob/master/cli/cli_cheats_sheet.pdf

© 2023 FRED LIVINGSTON. ALL RIGHTS RESERVED.

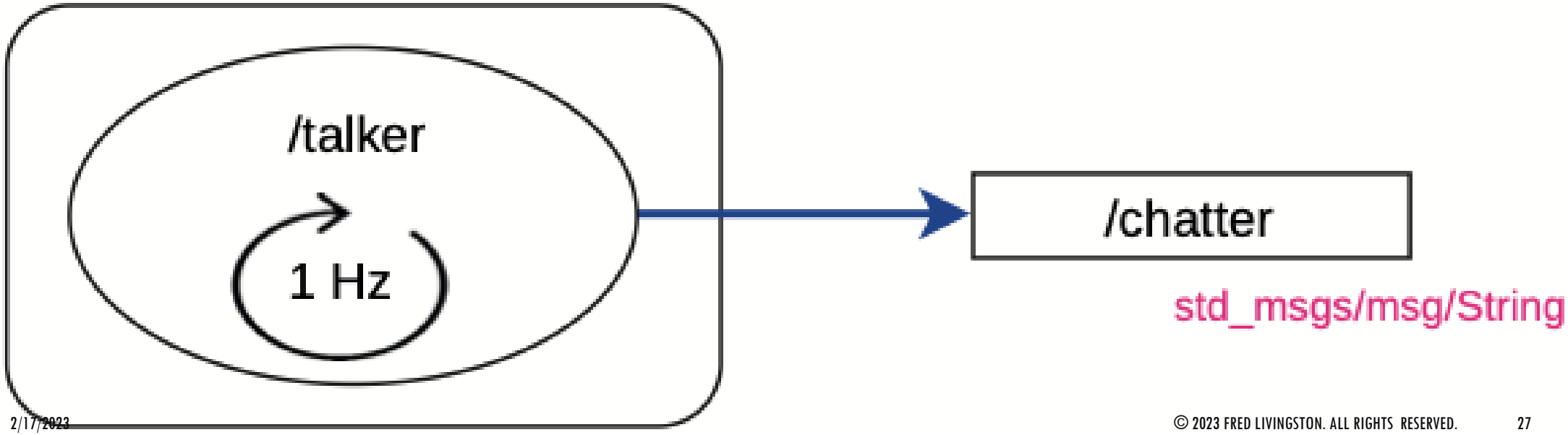
```
$ ros2 pkg list

ackermann_msgs
action_msgs
action_tutorials_cpp
...
```

```
$ ros2 pkg executables demo_nodes_cpp

demo_nodes_cpp add_two_ints_client
demo_nodes_cpp add_two_ints_client_async
demo_nodes_cpp add_two_ints_server
demo_nodes_cpp allocator_tutorial
...
demo_nodes_cpp talker
...
```


© 2023 FRED LIVINGSTON. ALL RIGHTS RESERVED.

RUNNING A ROS2 PROGRAM (PUBLISHER)

```
$ ros2 run demo_nodes_cpp talker

[INFO] [1643218362.316869744] [talker]: Publishing: 'Hello World: 1'
[INFO] [1643218363.316915225] [talker]: Publishing: 'Hello World: 2'
[INFO] [1643218364.316907053] [talker]: Publishing: 'Hello World: 3'
...
```

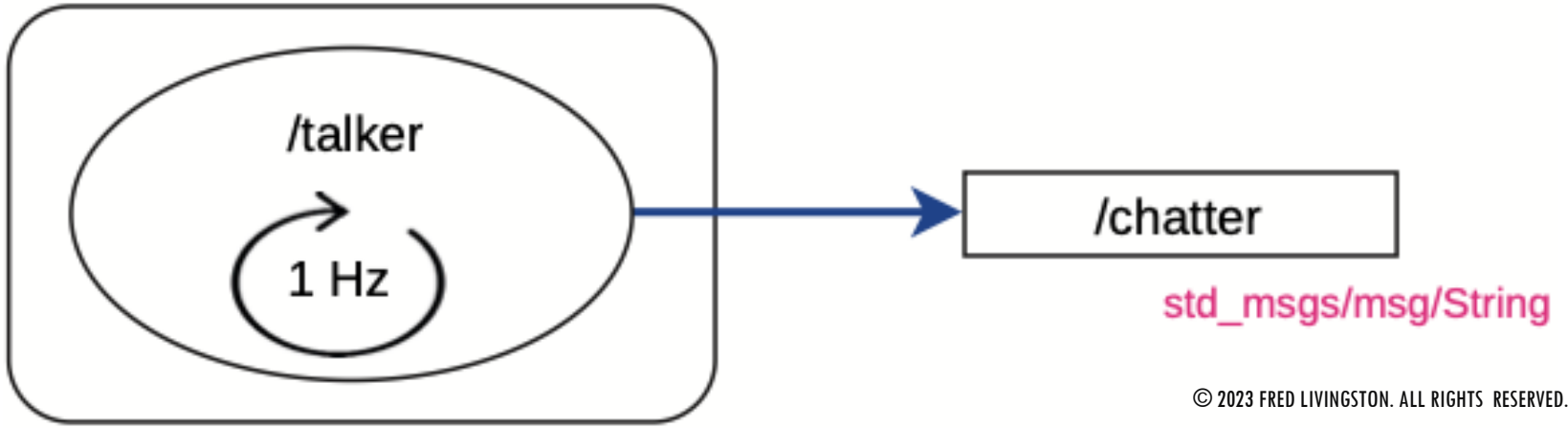


© 2023 FRED LIVINGSTON. ALL RIGHTS RESERVED.

RUNNING A ROS2 PROGRAM

```
$ ros2 node list

/talker
```

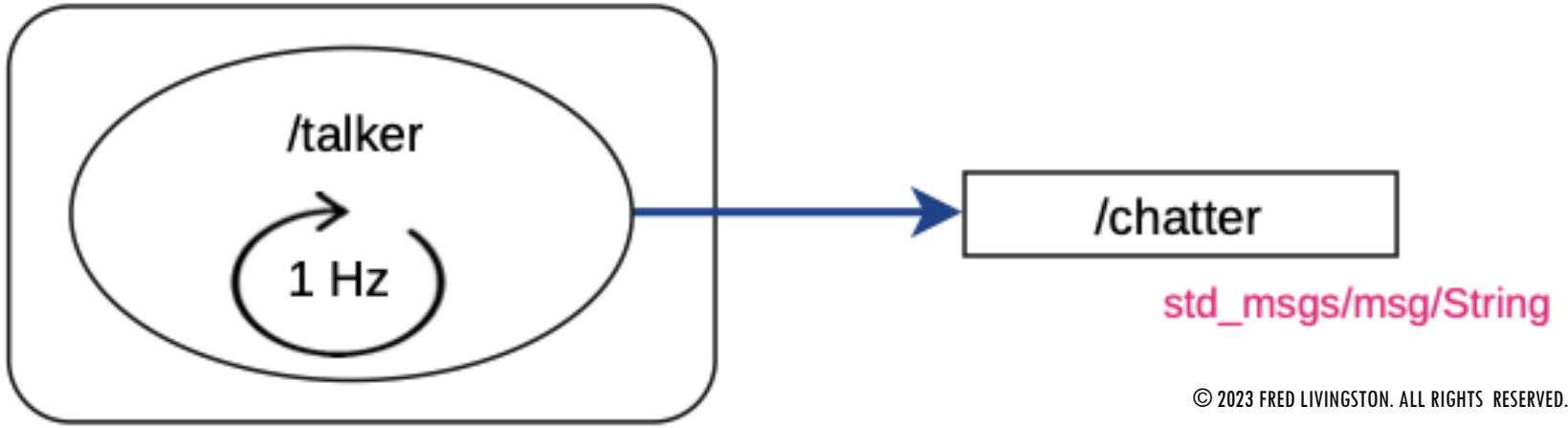
```
$ ros2 topic list

/chatter
/parameter_events
/rosout
```



© 2023 FRED LIVINGSTON. ALL RIGHTS RESERVED.

RUNNING A ROS2 PROGRAM

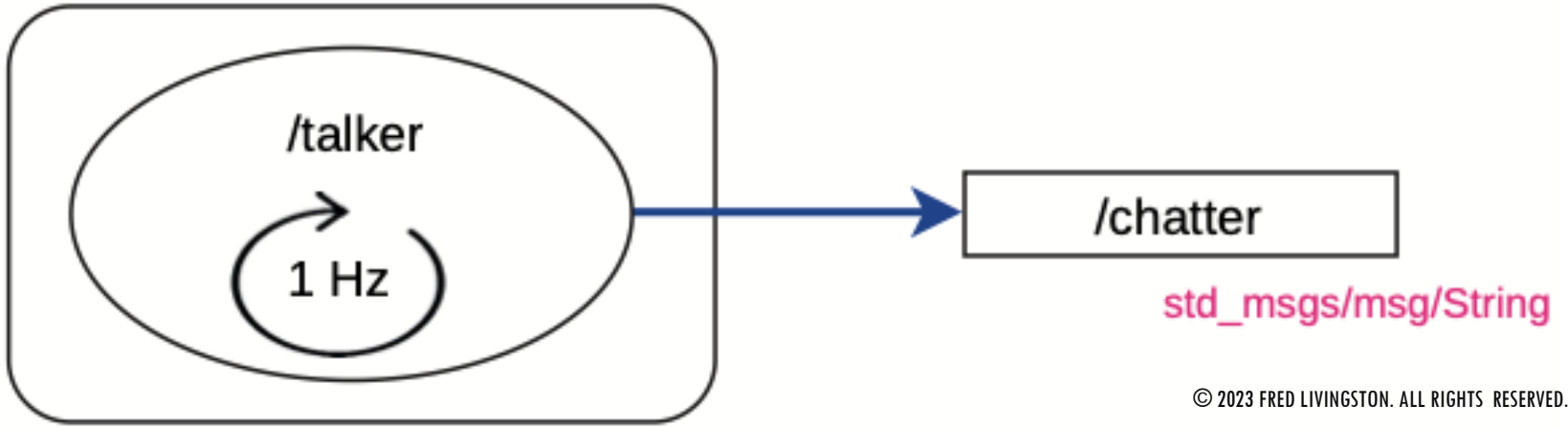
```
$ ros2 node info /talker

/talker
   Subscribers:
        /parameter_events: rcl_interfaces/msg/ParameterEvent
   Publishers:
        /chatter: std_msgs/msg/String
        /parameter_events: rcl_interfaces/msg/ParameterEvent
        /rosout: rcl_interfaces/msg/Log
   Service Servers:

...
```



© 2023 FRED LIVINGSTON. ALL RIGHTS RESERVED.

RUNNING A ROS2 PROGRAM

```
$ ros2 topic info /chatter

Type: std_msgs/msg/String
Publisher count: 1

Subscription count: 0
```



© 2023 FRED LIVINGSTON. ALL RIGHTS RESERVED.

INTERFACES

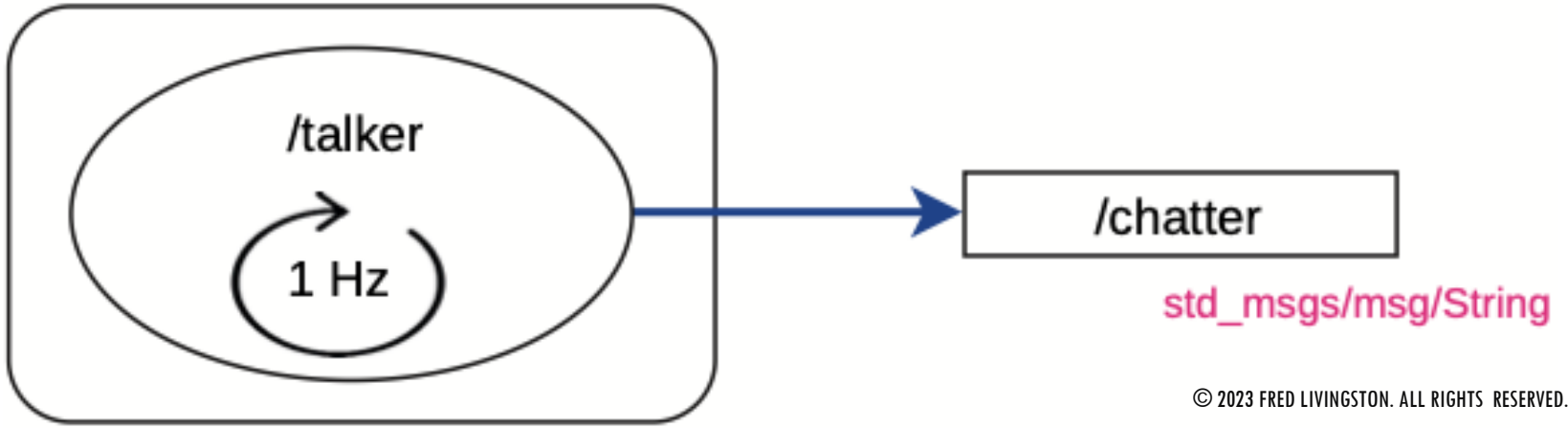
```
$ ros2 interface list

Messages:
    ackermann_msgs/msg/AckermannDrive
    ackermann_msgs/msg/AckermannDriveStamped
    ...
    visualization_msgs/msg/MenuEntry
Services:
    action_msgs/srv/CancelGoal
    ...
    visualization_msgs/srv/GetInteractiveMarkers
Actions:
    action_tutorials_interfaces/action/Fibonacci
    ...
```

```
$ ros2 interface show std_msgs/msg/String

... comments

string data
```


© 2023 FRED LIVINGSTON. ALL RIGHTS RESERVED.

TOPICS

```
$ ros2 topic echo /chatter

data: 'Hello World: 1578'
---
data: 'Hello World: 1579'

...
```



© 2023 FRED LIVINGSTON. ALL RIGHTS RESERVED.

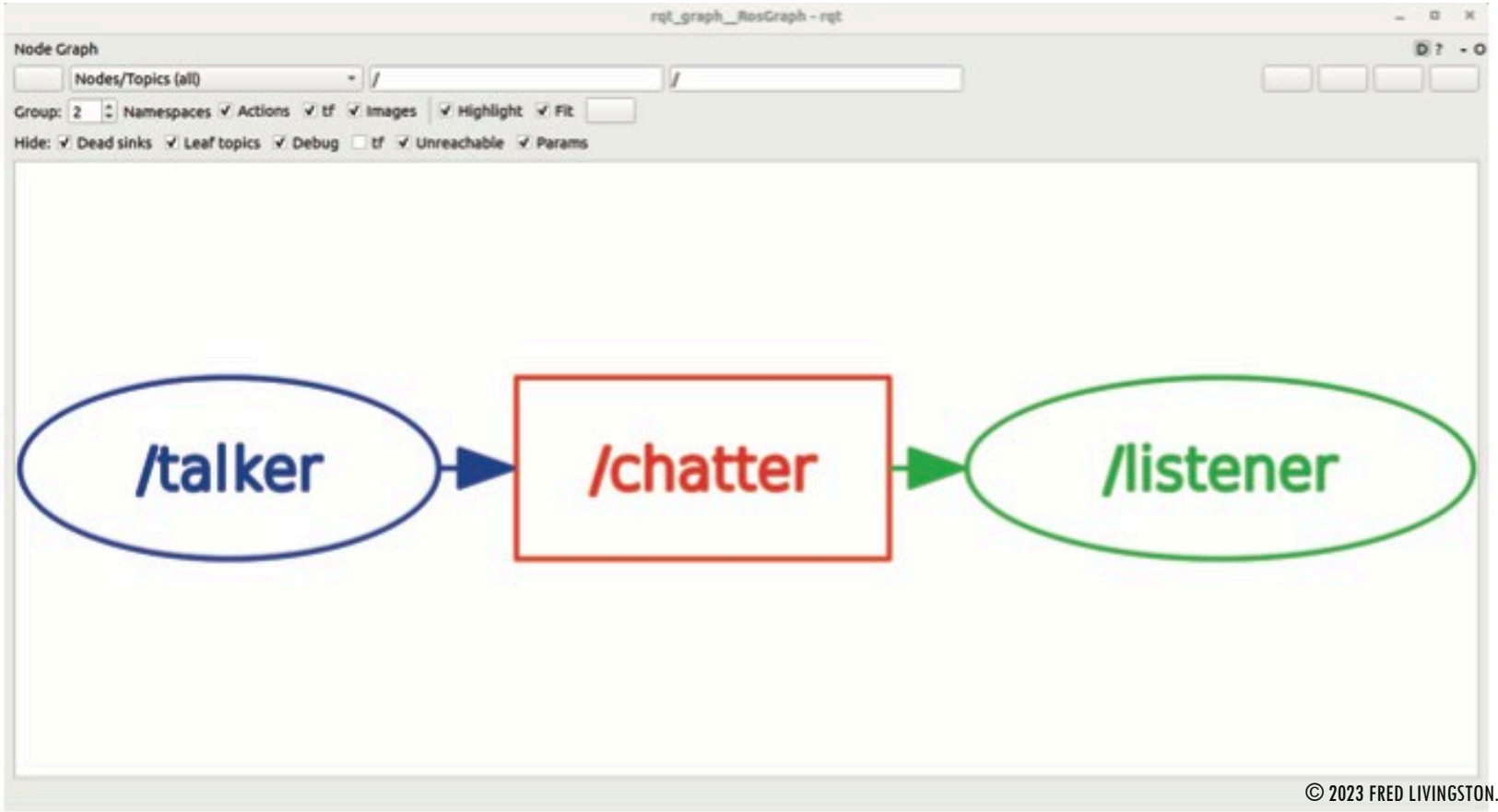
RUNNING A SUBSCRIBER

```
$ ros2 run demo_nodes_py listener

[INFO] [1643220136.232617223] [listener]: I heard: [Hello World: 1670]
[INFO] [1643220137.197551366] [listener]: I heard: [Hello World: 1671]
[INFO] [1643220138.198640098] [listener]: I heard: [Hello World: 1672]

...
```

© 2023 FRED LIVINGSTON. ALL RIGHTS RESERVED.

RQT TOOLS

```
$ ros2 run rqt_graph rqt_graph
```



© 2023 FRED LIVINGSTON. ALL RIGHTS RESERVED.

F1TENTH GYM

https://github.com/f1tenth/f1tenth_gym_ros

Native on Ubuntu 20.04

Install the following dependencies:

- **ROS 2** Follow the instructions here to install ROS 2 Foxy.
- **F1TENTH Gym**

```
git clone https://github.com/f1tenth/f1tenth_gym
cd f1tenth_gym && pip3 install -e .
```

Installing the simulation:

- Create a workspace: `cd $HOME && mkdir -p sim_ws/src`
- Clone the repo into the workspace:

```
cd $HOME/sim_ws/src
git clone https://github.com/f1tenth/f1tenth_gym_ros
```

- Update correct parameter for path to map file: Go to `sim.yaml` https://github.com/f1tenth/f1tenth_gym_ros/blob/main/config/sim.yaml in your cloned repo, change the `map_path` parameter to point to the correct location. It should be `'<your_home_dir>/sim_ws/src/f1tenth_gym_ros/maps/levine'`
- Install dependencies with rosdep:

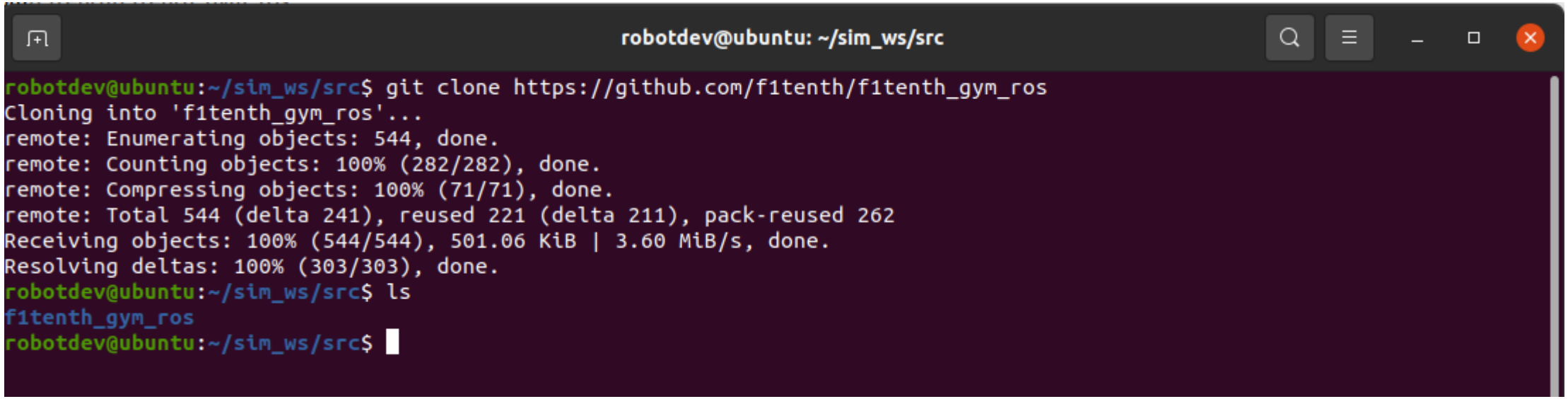
```
source /opt/ros/foxy/setup.bash
cd ..
rosdep install -i --from-path src --rosdistro foxy -y
```

- Build the workspace: `colcon build`

© 2023 FRED LIVINGSTON. ALL RIGHTS RESERVED.

F1TENTH GYM (CLONE REPO)

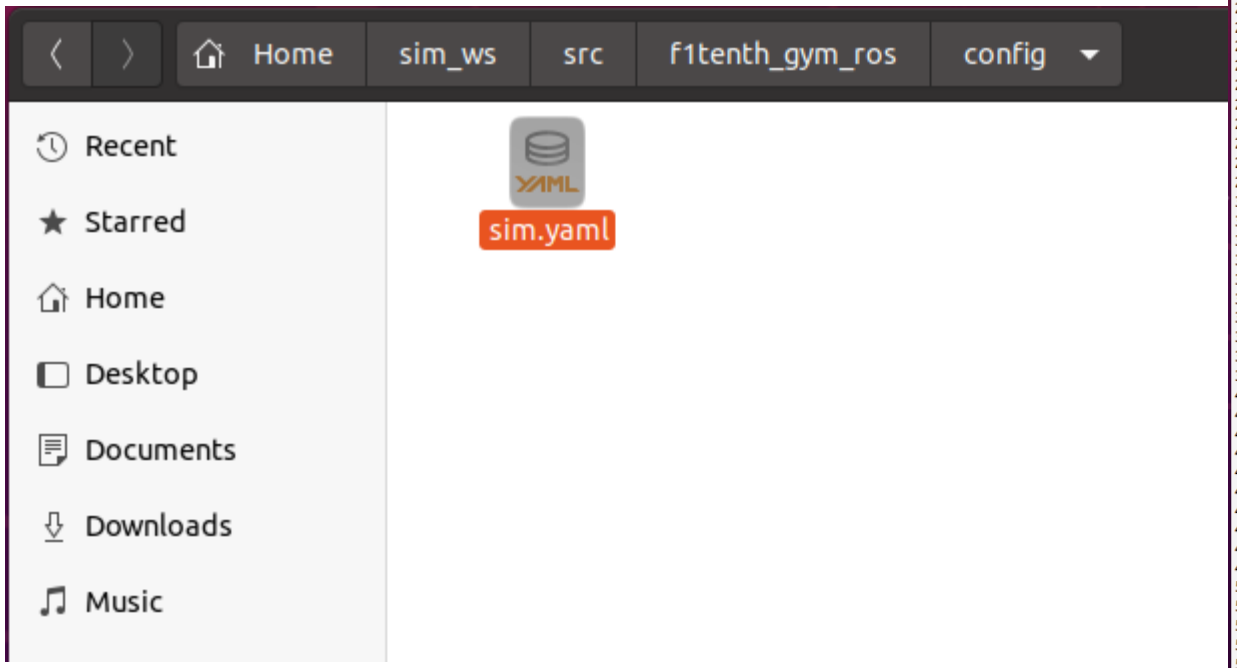
$ git clone https://github.com/f1tenth/f1tenth_gym_ros



© 2023 FRED LIVINGSTON. ALL RIGHTS RESERVED.

F1TENTH GYM (CONFIGURATION)

Map_path: must contain full path name
Num_agent: 1 or 2



sim.yaml
~/sim_ws/src/f1tenth_gym_ros/config

```yaml
# copies or substantial portions of the Software.

# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
# IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
# FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
# AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
# LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
# OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
# SOFTWARE.

bridge:
  ros__parameters:
    # topics and namespaces
    ego_namespace: 'ego_racecar'
    ego_scan_topic: 'scan'
    ego_odom_topic: 'odom'
    ego_opp_odom_topic: 'opp_odom'
    ego_drive_topic: 'drive'
    opp_namespace: 'opp_racecar'
    opp_scan_topic: 'opp_scan'
    opp_odom_topic: 'odom'
    opp_ego_odom_topic: 'opp_odom'
    opp_drive_topic: 'opp_drive'

    # transform related
    scan_distance_to_base_link: 0.0

    # laserscan parameters
    scan_fov: 4.7
    scan_beams: 1080

    # map parameters
    map_path: '/home/robotdev/sim_ws/src/f1tenth_gym_ros/maps/levine'
    map_img_ext: '.png'

    # opponent parameters
    num_agent: 1

    # ego starting pose on map
    sx: 0.0
    sy: 0.0
    stheta: 0.0

    # opp starting pose on map
    sx1: 2.0
    sy1: 0.5
    stheta1: 0.0

    # teleop
    kb_teleop: True
```

YAML Tab Width: 8 Ln 17, Col 24 INS

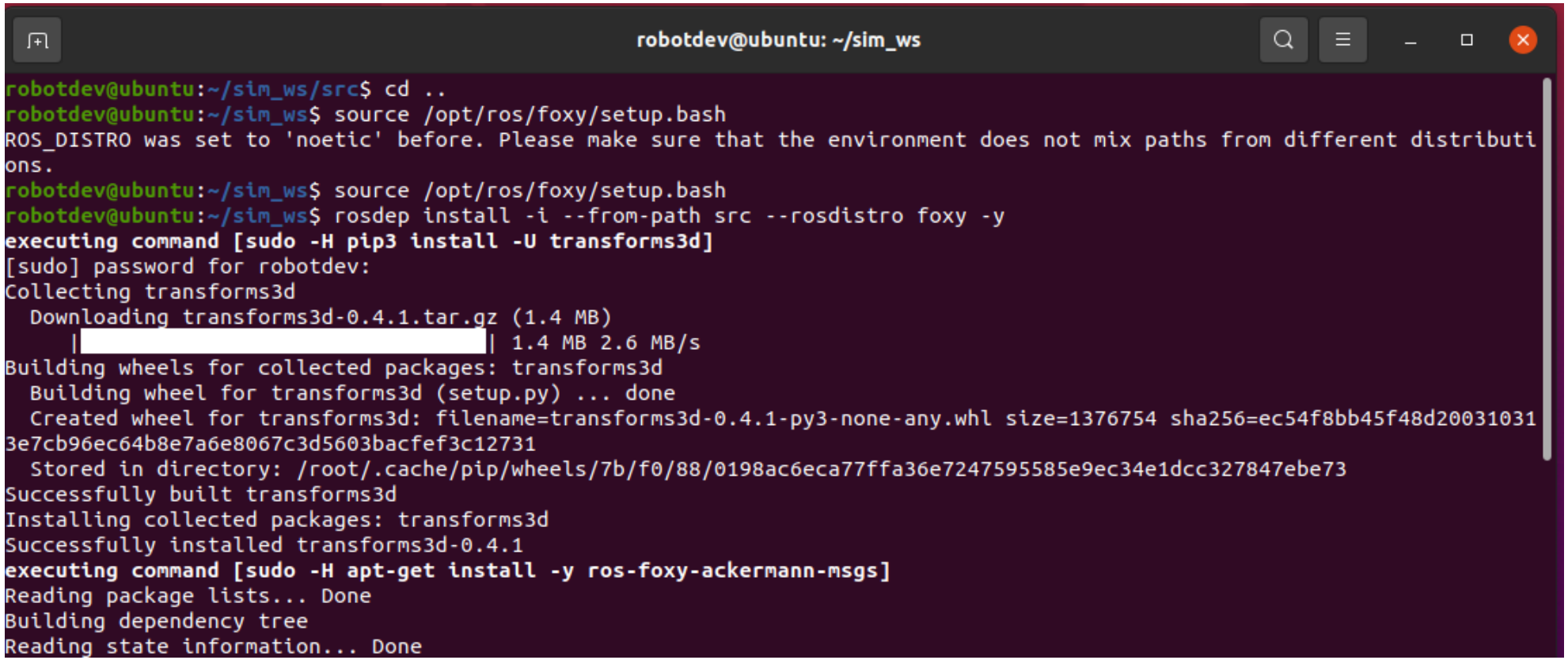
© 2023 FRED LIVINGSTON. ALL RIGHTS RESERVED.

F1TENTH GYM (INSTALL DEPENDENCIES)

$ cd ..

$ source /opt/ros/foxy/setup.bash

$ rosdep install -i –from-path src –rosdistro foxy -y



© 2023 FRED LIVINGSTON. ALL RIGHTS RESERVED.

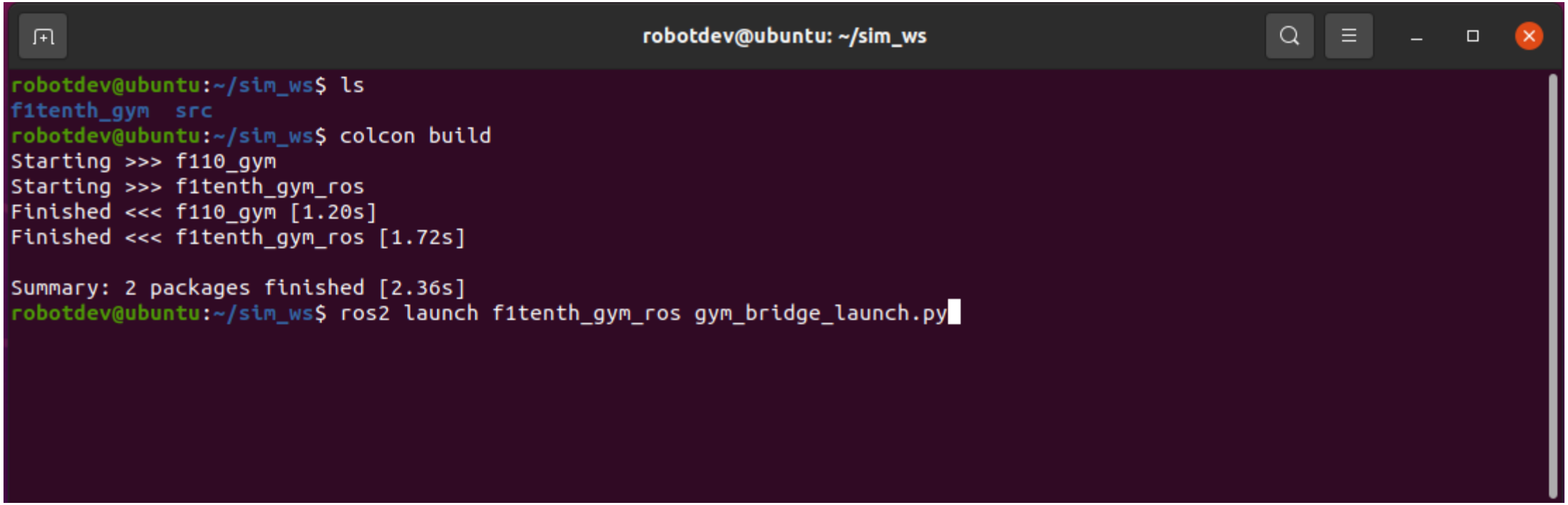
F1TENTH GYM (COMPILE SRC)

$ colcon build

© 2023 FRED LIVINGSTON. ALL RIGHTS RESERVED.

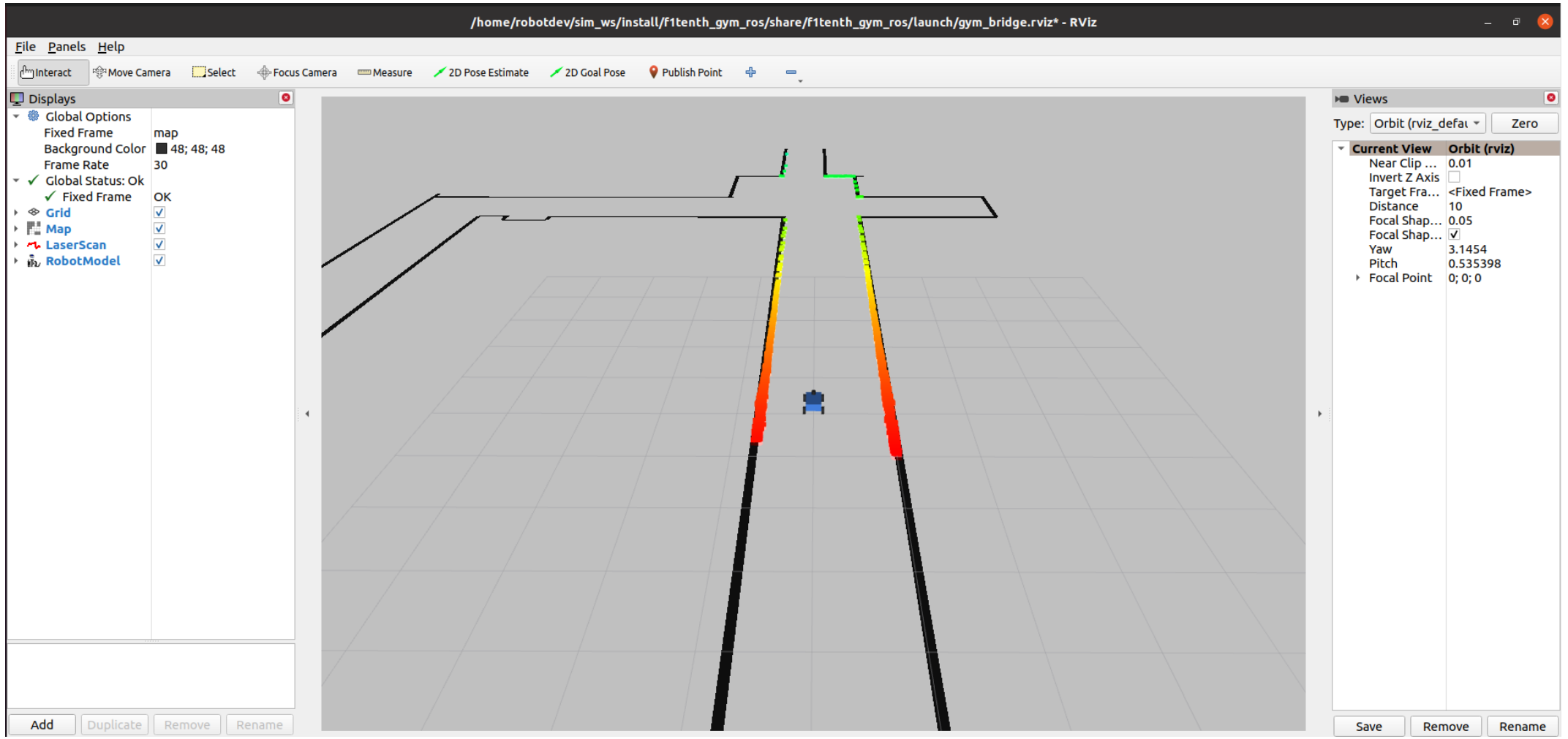
F1TENTH GYM (RUN SIMULATOR)

$ ros2 launch f1tenth_gym_ros gym_bridge_launch.py



© 2023 FRED LIVINGSTON. ALL RIGHTS RESERVED.

F1TENTH GYM (RUN SIMULATOR)



© 2023 FRED LIVINGSTON. ALL RIGHTS RESERVED.

TOPICS PUBLISHED BY THE SIMULATION

In single agent:

/scan: The ego agent's laser scan

/ego_racecar/odom: The ego agent's odometry

/map: The map of the environment

In two agents:

In addition to the topics available in the single agent scenario, these topics are also available:

/opp_scan: The opponent agent's laser scan

/ego_racecar/opp_odom: The opponent agent's odometry for the ego agent's planner

/opp_racecar/odom: The opponent agents' odometry

/opp_racecar/opp_odom: The ego agent's odometry for the opponent agent's planner

© 2023 FRED LIVINGSTON. ALL RIGHTS RESERVED.

TOPICS SUBSCRIBED BY THE SIMULATION

In single agent:

/**drive**: The ego agent's drive command via **AckermannDriveStamped** messages

/**initalpose**: This is the topic for resetting the ego's pose via RViz's 2D Pose Estimate tool. Do NOT publish directly to this topic unless you know what you're doing.

In two agents:

In addition to all topics in the single agent scenario, these topics are also available:

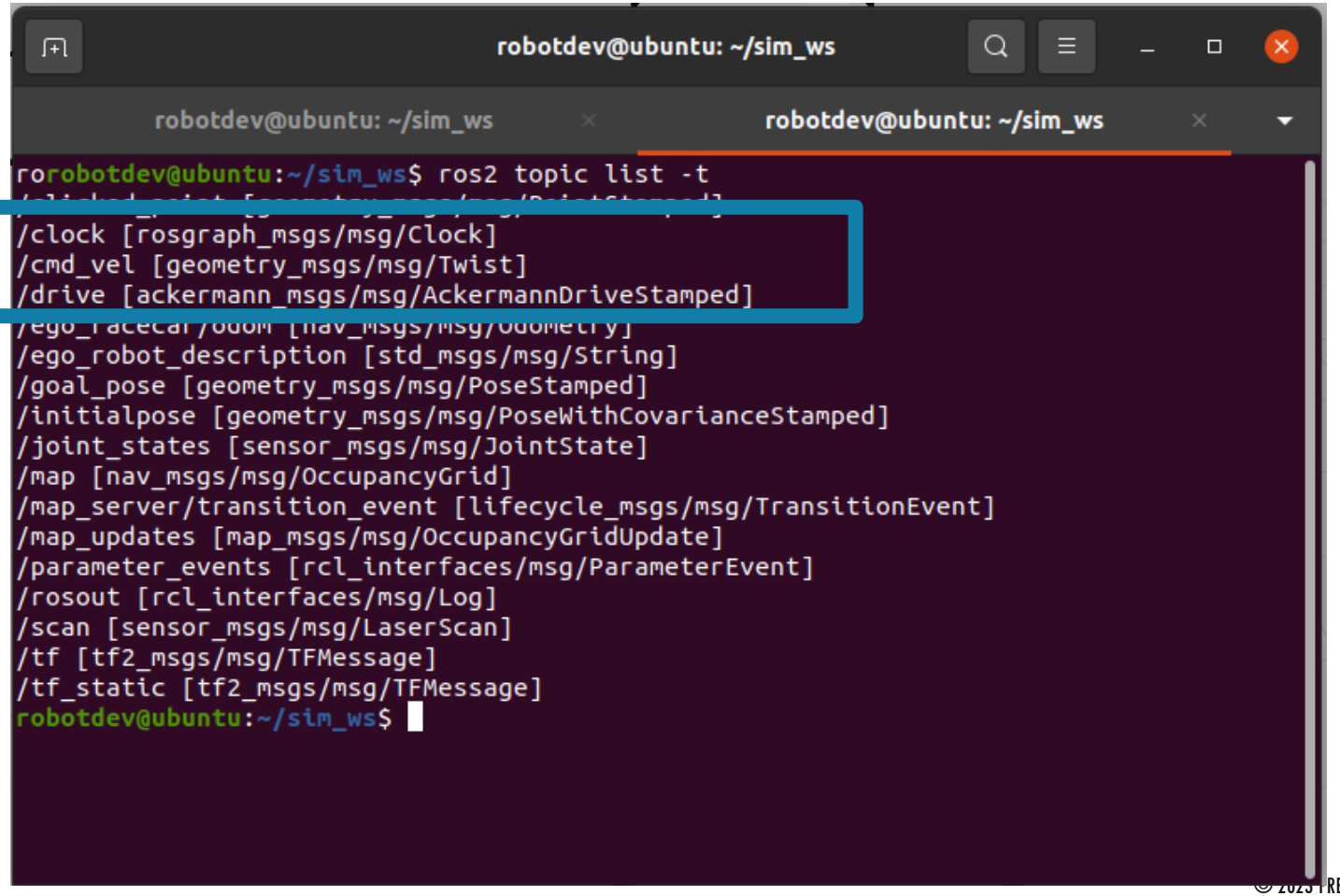
/**opp_drive**: The opponent agent's drive command via AckermannDriveStamped messages

/**goal_pose**: This is the topic for resetting the opponent agent's pose via RViz's 2D Goal Pose tool. Do NOT publish directly to this topic unless you know what you're doing

© 2023 FRED LIVINGSTON. ALL RIGHTS RESERVED.

F1TENTH GYM (TOPICS)

$ ros2 topic list -t



© 2023 FRED LIVINGSTON. ALL RIGHTS RESERVED.

GEOMETRY_MSGS/MSG/TWIST

http://docs.ros.org/en/noetic/api/geometry_msgs/html/msg/Twist.html

geometry_msgs/Twist Message

File: geometry_msgs/Twist.msg

Raw Message Definition

```
# This expresses velocity in free space broken into its linear and angular parts.
Vector3  linear
Vector3  angular
```

Compact Message Definition

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

autogenerated on Wed, 02 Mar 2022 00:06:53

© 2023 FRED LIVINGSTON. ALL RIGHTS RESERVED.

ACKERMANN_MSGS/MAG/ACKERMANNDRIVESTAMPED

http://docs.ros.org/en/melodic/api/ackermann_msgs/html/msg/AckermannDriveStamped.html

ackermann_msgs/AckermannDriveStamped Message

File: ackermann_msgs/AckermannDriveStamped.msg

Raw Message Definition

```
## Time stamped drive command for robots with Ackermann steering.
#  $Id$

Header          header
AckermannDrive  drive
```

Compact Message Definition

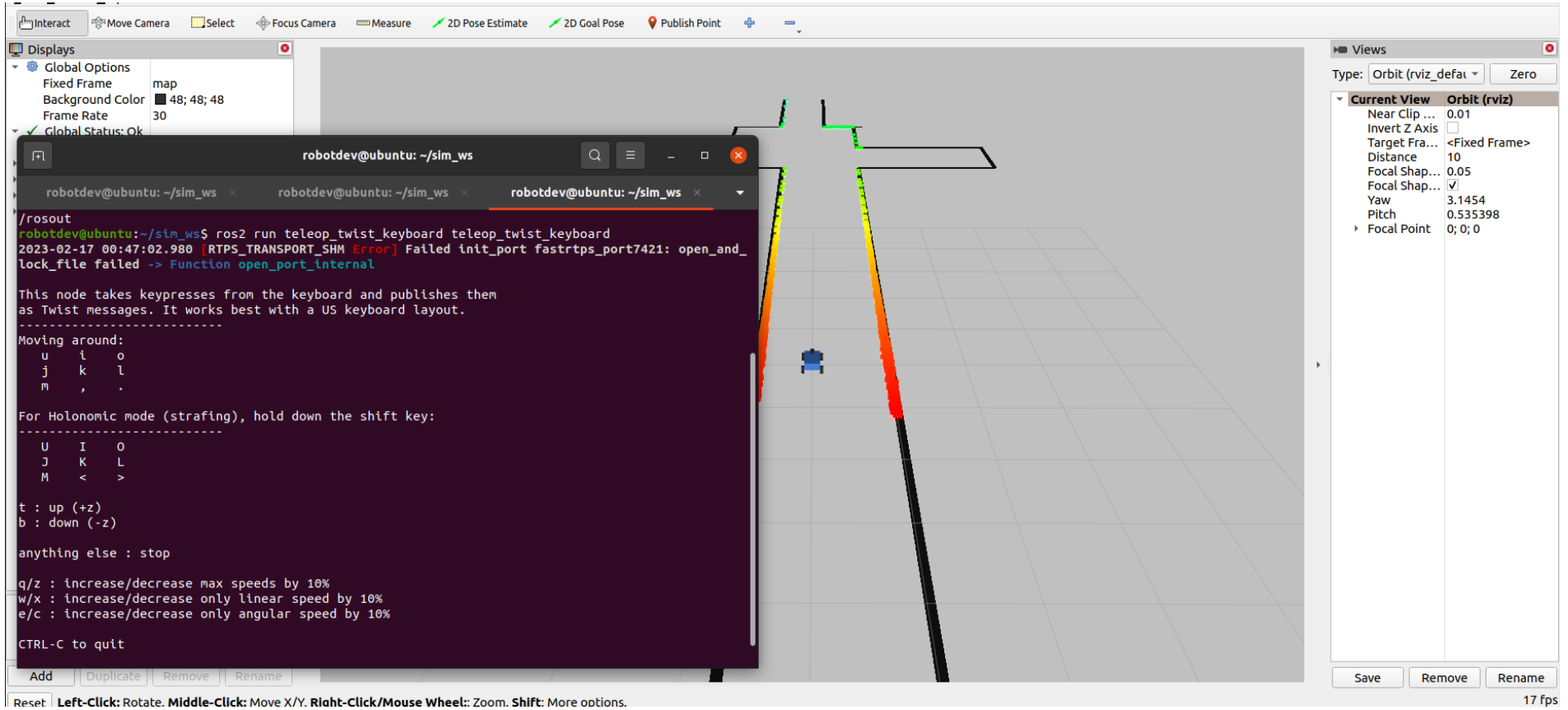
```
std_msgs/Header header
ackermann_msgs/AckermannDrive drive
```

autogenerated on Mon, 28 Feb 2022 21:32:24

© 2023 FRED LIVINGSTON. ALL RIGHTS RESERVED.

F1TENTH GYM (TELEOP)

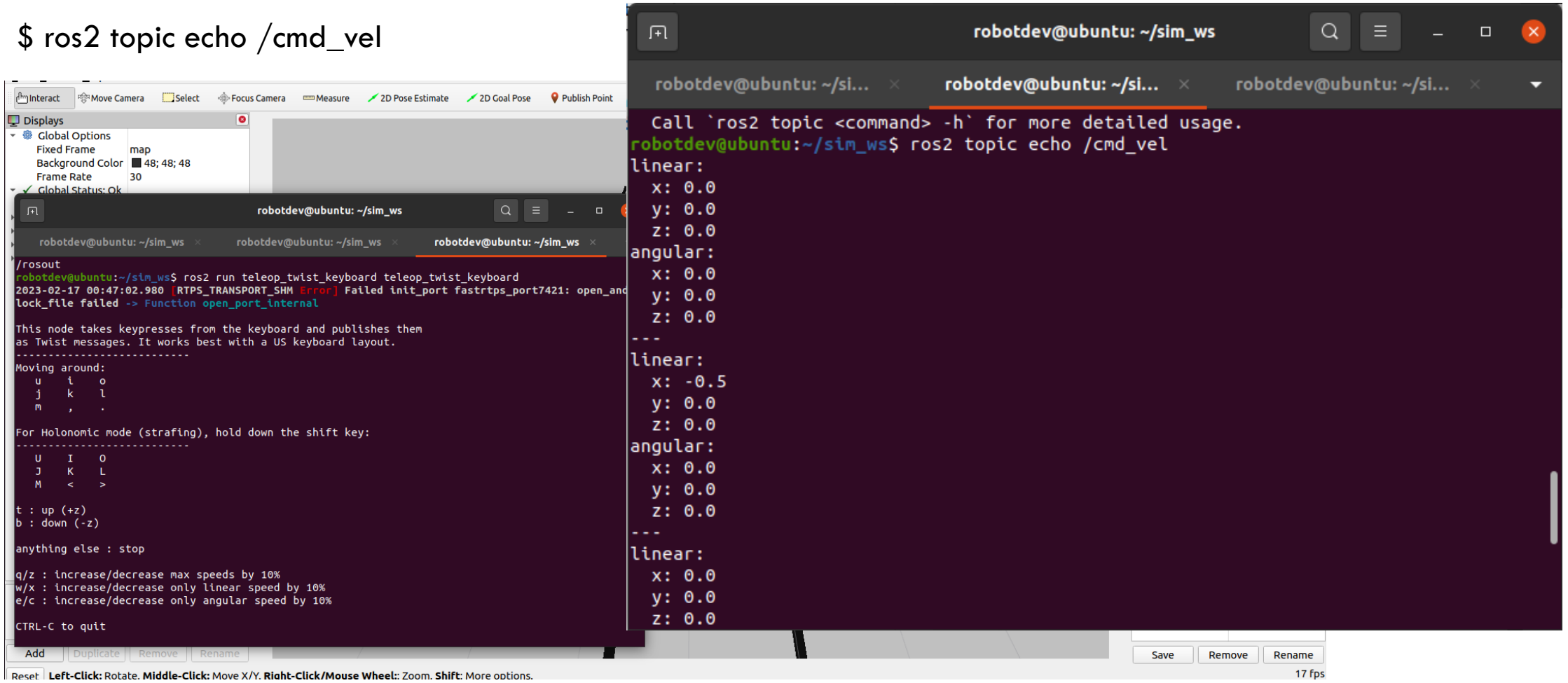
$ ros2 run teleop_twist_keyboard teleop_twist_keyboard



© 2023 FRED LIVINGSTON. ALL RIGHTS RESERVED.

F1TENTH GYM (TELEOP)

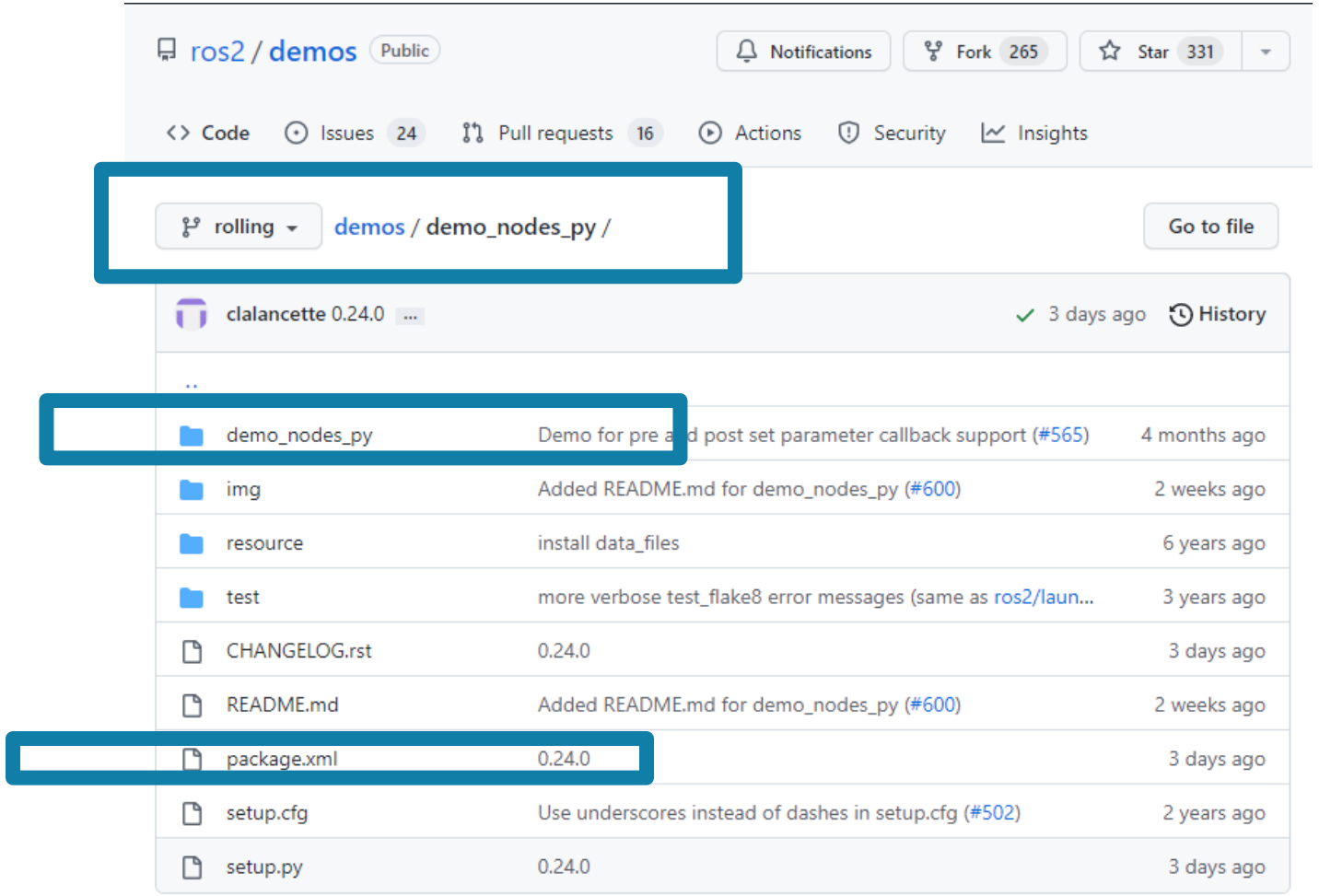
$ ros2 topic echo /cmd_vel



© 2023 FRED LIVINGSTON. ALL RIGHTS RESERVED.

DEVELOPING PROGRAMS

ROS2 Demos
https://github.com/ros2/demos



© 2023 FRED LIVINGSTON. ALL RIGHTS RESERVED.

PACKAGE.XML

rolling ▾ demos / demo_nodes_py / package.xml Go to file ···

clalancette 0.24.0 ✓ Latest commit 2829b31 3 days ago History

12 contributors

33 lines (27 sloc) | 1.22 KB Raw Blame

```
1  <?xml version="1.0"?>
2  <package format="2">
3    <name>demo_nodes_py</name>
4    <version>0.24.0</version>
5    <description>
6      Python nodes which were previously in the ros2/examples repository but are now just used for demo purposes.
7    </description>
8
9    <maintainer email="aditya.pande@openrobotics.org">Aditya Pande</maintainer>
10   <maintainer email="audrow@openrobotics.org">Audrow Nash</maintainer>
11   <maintainer email="michael.jeronimo@openrobotics.org">Michael Jeronimo</maintainer>
12
13   <license>Apache License 2.0</license>
14
15   <author email="esteve@osrfoundation.org">Esteve Fernandez</author>
16   <author email="mabel@openrobotics.org">Mabel Zhang</author>
17   <author email="michael@openrobotics.org">Michael Carroll</author>
18   <author>Mikael Arguedas</author>
19
20   <exec_depend>example_interfaces</exec_depend>
21   <exec_depend>rclpy</exec_depend>
22   <exec_depend>std_msgs</exec_depend>
23
24   <test_depend>ament_copyright</test_depend>
25   <test_depend>ament_flake8</test_depend>
26   <test_depend>ament_pep257</test_depend>
27   <test_depend>python3-pytest</test_depend>
28
29   <export>
30     <build_type>ament_python</build_type>
31   </export>
32 </package>
33
```


© 2023 FRED LIVINGSTON. ALL RIGHTS RESERVED.

CREATE A NEW PACKAGE

$ cd ~/sim_ws/src

$ ros2 pkg create my_robot_controller --build-type ament_python

```
robotdev@ubuntu:~/sim_ws/src$ ls
f1tenth_gym_ros
robotdev@ubuntu:~/sim_ws/src$ ros2 pkg create my_robot_controller --build-type ament_python
going to create a new package
package name: my_robot_controller
destination directory: /home/robotdev/sim_ws/src
package format: 3
version: 0.0.0
description: TODO: Package description
maintainer: ['robotdev <Fred.Livingston@gmail.com>']
licenses: ['TODO: License declaration']
build type: ament_python
dependencies: []
creating folder ./my_robot_controller
creating ./my_robot_controller/package.xml
creating source folder
creating folder ./my_robot_controller/my_robot_controller
creating ./my_robot_controller/setup.py
creating ./my_robot_controller/setup.cfg
creating folder ./my_robot_controller/resource
creating ./my_robot_controller/resource/my_robot_controller
creating ./my_robot_controller/my_robot_controller/__init__.py
creating folder ./my_robot_controller/test
creating ./my_robot_controller/test/test_copyright.py
creating ./my_robot_controller/test/test_flake8.py
creating ./my_robot_controller/test/test_pep257.py
robotdev@ubuntu:~/sim_ws/src$
```


LL RIGHTS RESERVED.

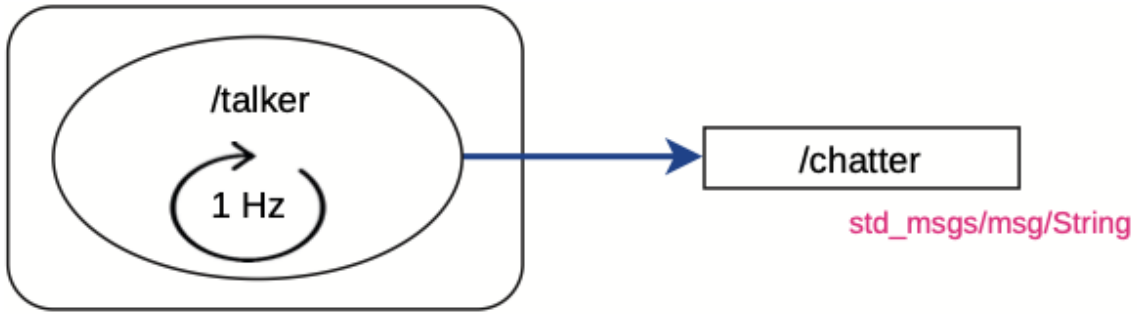
SIMPLE PUBLISHER (TALKER.PY)

https://github.com/ros2/demos/blob/rolling/demo_nodes_py/demo_nodes_py/topics/talker.py

Line 25:
super().__init__('NAME_OF_PROCESS')

Line 27:
self.create_publisher(String, 'chatter', 10)

Line 29:
self.create_timer(1, self.time_call_back)




```
14
15  import rclpy
16  from rclpy.executors import ExternalShutdownException
17  from rclpy.node import Node
18
19  from std_msgs.msg import String
20
21
22  class Talker(Node):
23
24      def __init__(self):
25          super().__init__('talker')
26          self.i = 0
27          self.pub = self.create_publisher(String, 'chatter', 10)
28          timer_period = 1.0
29          self.tmr = self.create_timer(timer_period, self.timer_callback)
30
31      def timer_callback(self):
32          msg = String()
33          msg.data = 'Hello World: {0}'.format(self.i)
34          self.i += 1
35          self.get_logger().info('Publishing: "{0}"'.format(msg.data))
36          self.pub.publish(msg)
37
```


© 2023 FRED LIVINGSTON. ALL RIGHTS RESERVED.

CREATE A PYTHON PROGRAM

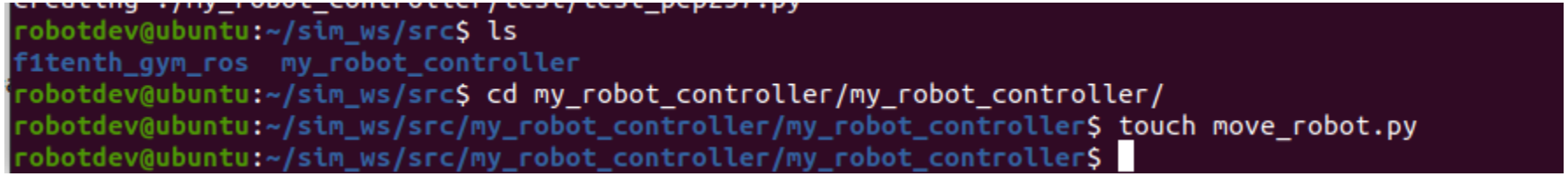
$ cd my_robot_controller/my_robot_controller/
$ touch move_robot.py
$ gedit move_robot.py

```
robotdev@ubuntu:~/sim_ws/src$ ls
f1tenth_gym_ros  my_robot_controller
robotdev@ubuntu:~/sim_ws/src$ cd my_robot_controller/my_robot_controller/
robotdev@ubuntu:~/sim_ws/src/my_robot_controller/my_robot_controller$ touch move_robot.py
robotdev@ubuntu:~/sim_ws/src/my_robot_controller/my_robot_controller$
```


© 2023 FRED LIVINGSTON. ALL RIGHTS RESERVED.

MOVE_ROBOT.PY

$ cd my_robot_controller/my_robot_controller/
$ touch move_robot.py
$ gedit move_robot.py



© 2023 FRED LIVINGSTON. ALL RIGHTS RESERVED.

MOVE_ROBOT.PY

move_robot.py

home > robotdev > sim_ws > src > my_robot_controller > my_robot_controller > move_robot.py > ...

```python
# move_robot.py
# Fred Livingston (fjliving@ncsu.edu) 2-17-2023

import rclpy
from rclpy.executors import ExternalShutdownException
from rclpy.node import Node

from geometry_msgs.msg import Twist


class Controller(Node):

    def __init__(self):
        super().__init__('move_robot')
        self.pub = self.create_publisher(Twist, 'cmd_vel', 10)

        # move robot fwd
        msg = Twist()
        msg.linear.x = 0.5
        msg.linear.y = 0.0
        msg.linear.z = 0.0
        msg.angular.x = 0.0
        msg.angular.y = 0.0
        msg.angular.z = 0.0
        self.pub.publish(msg)

        timer_period = 10.0
        self.tmr = self.create_timer(timer_period, self.timer_callback)

    def timer_callback(self):
        # stop robot
        msg = Twist()
        msg.linear.x = 0.0
        msg.linear.y = 0.0
        msg.linear.z = 0.0
        msg.angular.x = 0.0
        msg.angular.y = 0.0
        msg.angular.z = 0.0
        self.pub.publish(msg)
```

move_robot.py

home > robotdev > sim_ws > src > my_robot_controller > my_robot_controller > move_robot.py > ...

```python
        timer_period = 10.0
        self.tmr = self.create_timer(timer_period, self.timer_callback)

    def timer_callback(self):
        # stop robot
        msg = Twist()
        msg.linear.x = 0.0
        msg.linear.y = 0.0
        msg.linear.z = 0.0
        msg.angular.x = 0.0
        msg.angular.y = 0.0
        msg.angular.z = 0.0
        self.pub.publish(msg)


def main(args=None):
    rclpy.init(args=args)

    node = Controller()

    try:
        rclpy.spin(node)
    except (KeyboardInterrupt, ExternalShutdownException):
        pass
    finally:
        node.destroy_node()
        rclpy.try_shutdown()


if __name__ == '__main__':
    main()
```


© 2023 FRED LIVINGSTON. ALL RIGHTS RESERVED.

SETUP.PY

Open | setup.py ~/sim_ws/src/my_robot_controller | Save

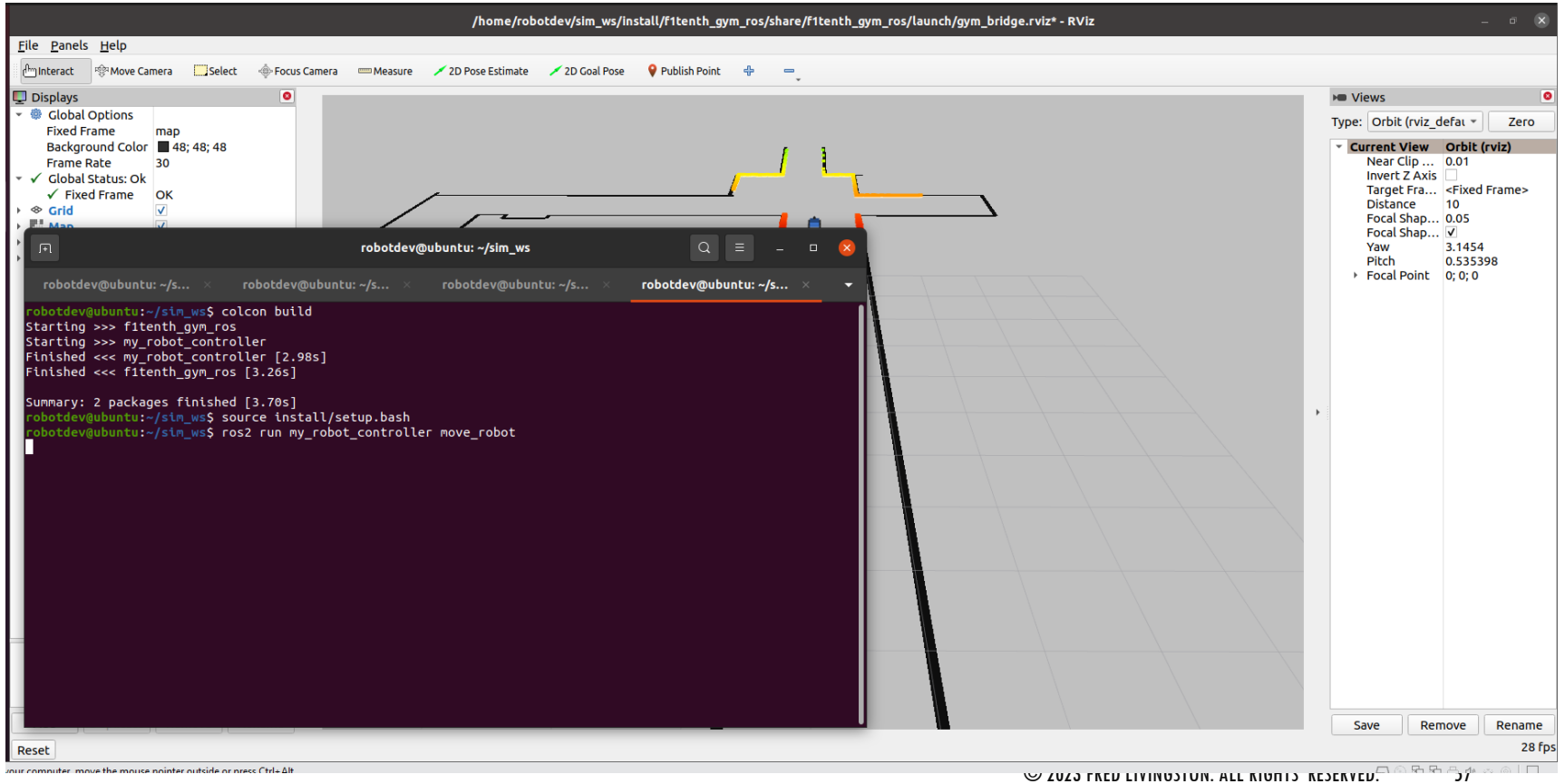
```
1 from setuptools import setup
2
3 package_name = 'my_robot_controller'
4
5 setup(
6     name=package_name,
7     version='0.0.0',
8     packages=[package_name],
9     data_files=[
10         ('share/ament_index/resource_index/packages',
11             ['resource/' + package_name]),
12         ('share/' + package_name, ['package.xml']),
13     ],
14     install_requires=['setuptools'],
15     zip_safe=True,
16     maintainer='robotdev',
17     maintainer_email='Fred.Livingston@gmail.com',
18     description='TODO: Package description',
19     license='TODO: License declaration',
20     tests_require=['pytest'],
21     entry_points={
22         'console_scripts': [
23         'move_robot = my_robot_controller.move_robot:main'
24         ],
25     },
26 )
```

Python | Tab Width: 8 | Ln 23, Col 20 | INS

© 2023 FRED LIVINGSTON. ALL RIGHTS RESERVED.

BUILD AND EXCUTE ROBOT CONTROLLER

$ colcon build
$ source install/setup.bash
$ ros2 run my_robot_controller move_robot



© 2023 FRED LIVINGSTON. ALL RIGHTS RESERVED.

END OF WORKSHOP

Fred Livingston (fjliving@ncsu.edu)

© 2023 FRED LIVINGSTON. ALL RIGHTS RESERVED.