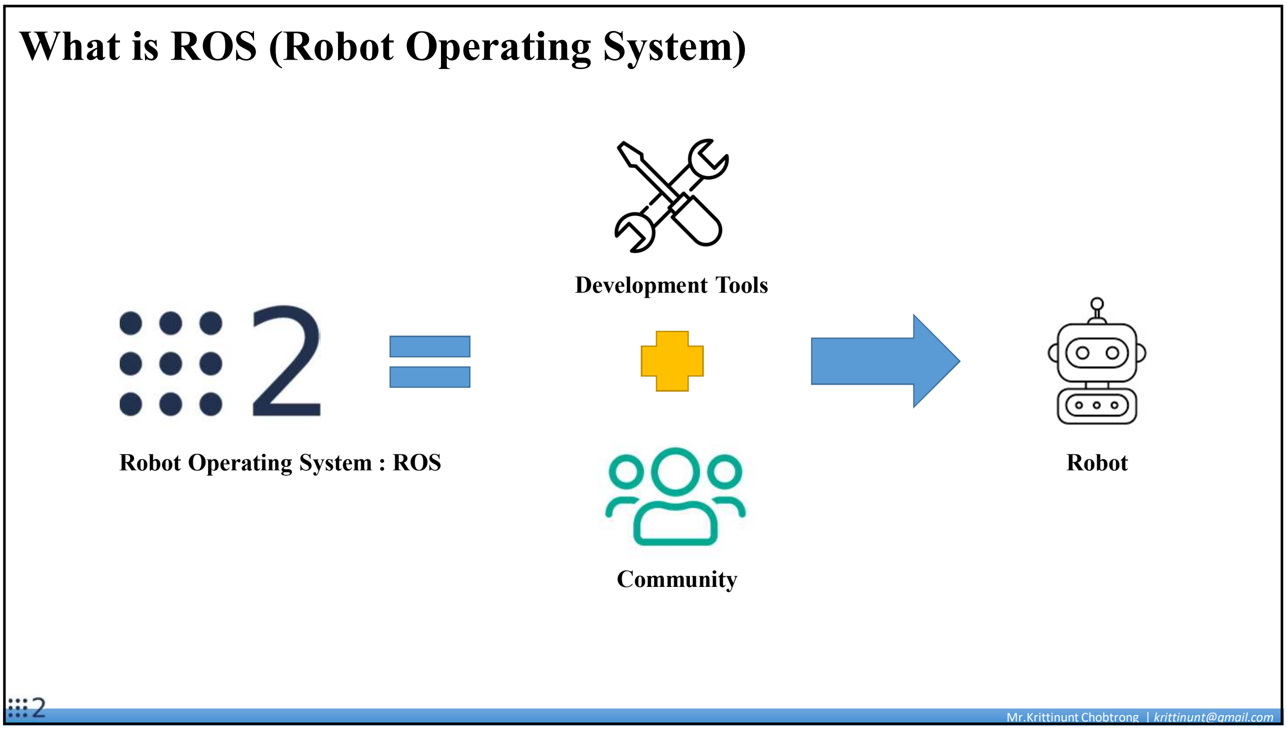
ROS 2™

Robot Operating System

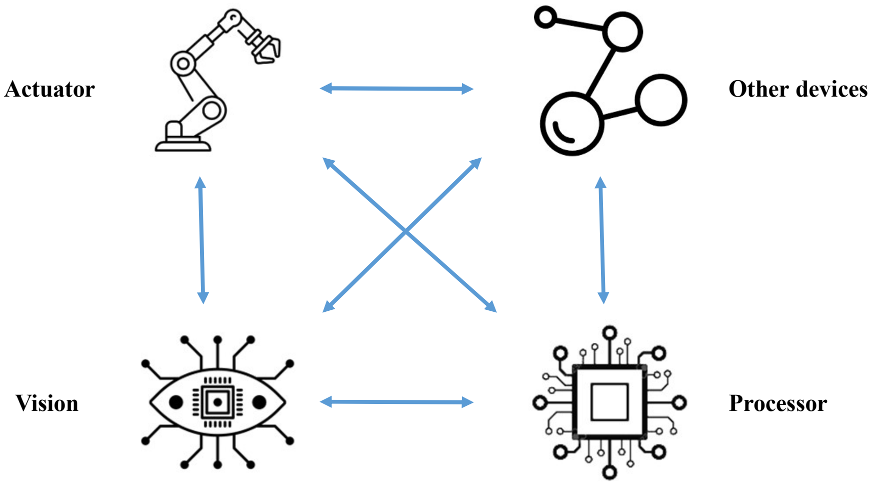
Mr.KRITTINUNT CHOBTRONG
krittinunt@gmail.com

https://github.com/krittinunt/ros/



What is ROS (Robot Operating System)

Robot Operating System



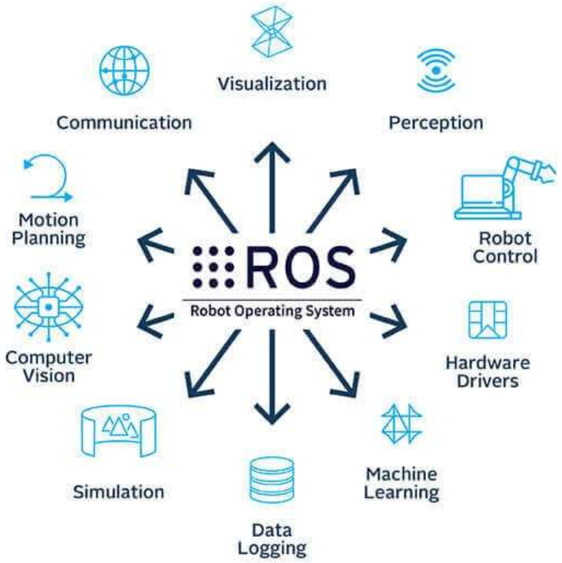
Actuator

Other devices

Vision

Processor

Robot Operating System



ROS

Robot Operating System

Visualization

Communication

Perception

Motion Planning

Robot Control

Computer Vision

Hardware Drivers

Simulation

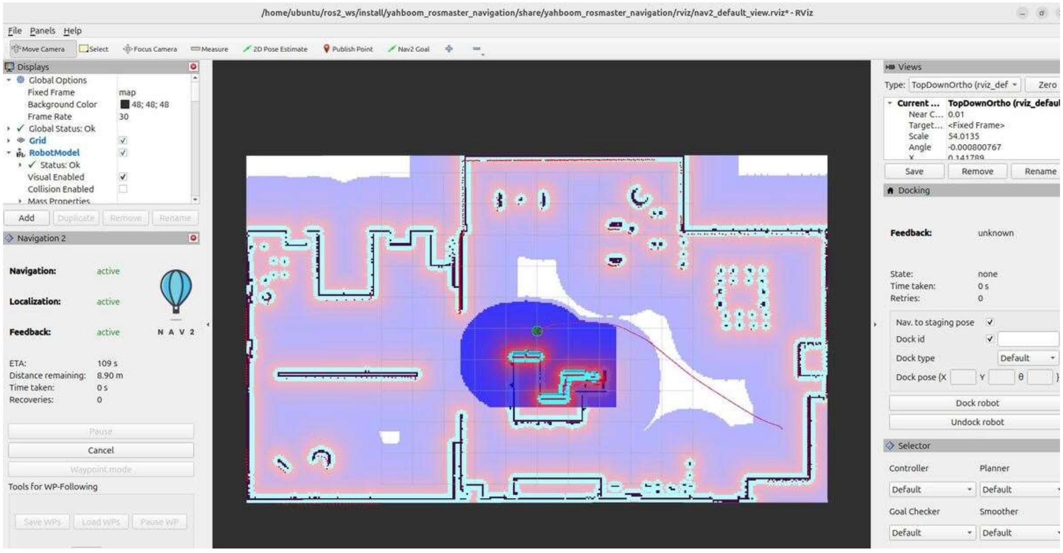
Machine Learning

Data Logging

Robot Operating System

NAV2

OPEN NAVIGATION



/home/ubuntu/ros2_ws/install/yahboom_rosmaster_navigation/share/yahboom_rosmaster_navigation/rviz/nav2_default_view.rviz* - RViz

Displays

- Global Options
  - Fixed Frame: map
  - Background Color: 48; 48; 48
  - Frame Rate: 30
- ✓ Global Status: Ok
- Grid ☑
- RobotModel ☑
  - ✓ Status: Ok
  - Visual Enabled ☑
  - Collision Enabled □
  - Mass Properties

Navigation 2

Navigation: active

Localization: active

Feedback: active

ETA: 109 s
Distance remaining: 8.90 m
Time taken: 0 s
Recoveries: 0

Tools for WP-Following

Views

Type: TopDownOrtho (rviz_def Zero

- Current ... TopDownOrtho (rviz_defaul
  - Near C... 0.01
  - Target... <Fixed Frame>
  - Scale 54.0135
  - Angle -0.000800767
  - X 0.141789

Docking

Feedback: unknown

State: none
Time taken: 0 s
Retries: 0

Nav. to staging pose ☑
Dock id ☑
Dock type: Default
Dock pose {X Y θ }

Selector

Controller: Default
Planner: Default
Goal Checker: Default
Smoother: Default

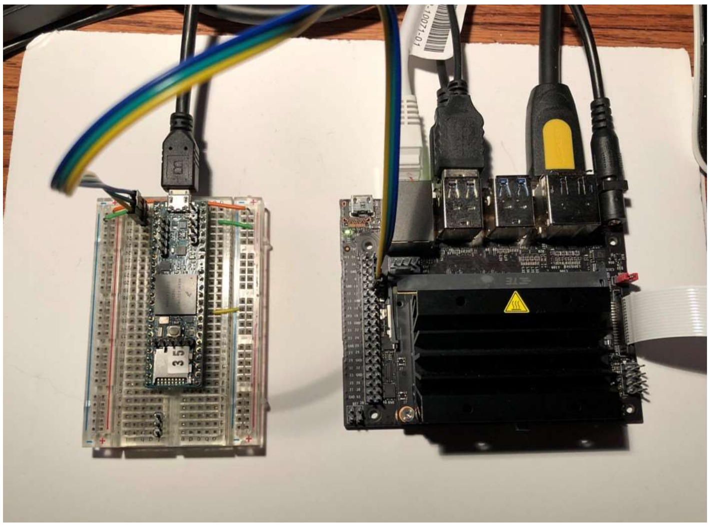
Robot Operating System

Activities
rviz2
Sep 25 18:51
/home/beto/ackbot_ws/src/ackbot_description/rviz/ackbot.rviz* - RViz
2D Pose Estimate
2D Goal Pose
Publish Point

Robot Operating System

Microcontroller

- ESP-32
- STM32
- Teensy 4.1



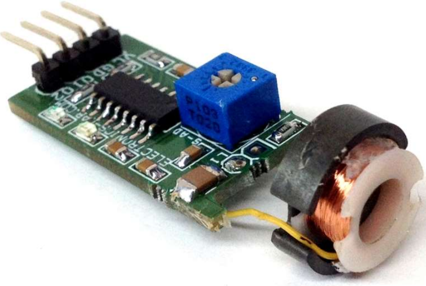
Single-Board Computer

- Raspberry Pi 4 / 5
- NVIDIA Jetson Nano / Orin / Xavier

Classification of Sensors

Tactile sensors

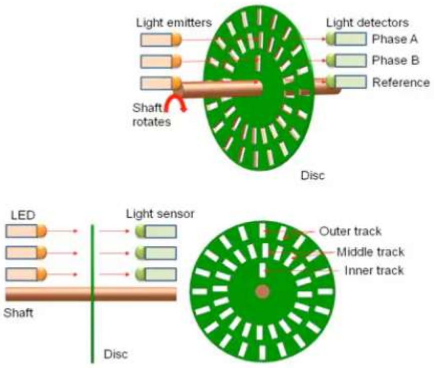
- Detection of physical contact
- Contact switches, bumpers optical barriers, Noncontact proximity sensors

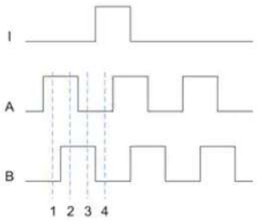


Classification of Sensors

Wheel/motor sensors

- Wheel/motor speed and position
- Optical encoders, Magnetic encoders

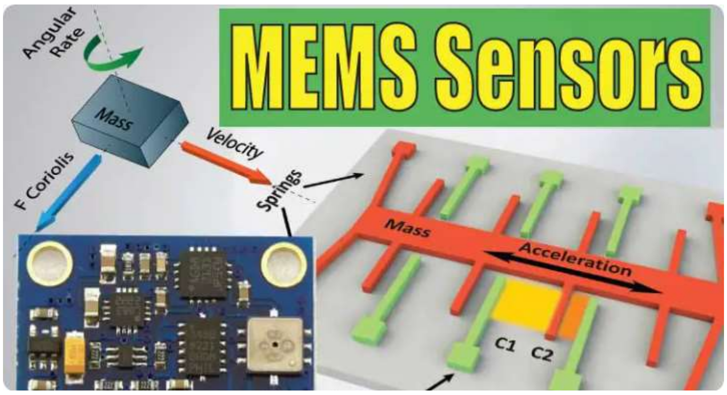


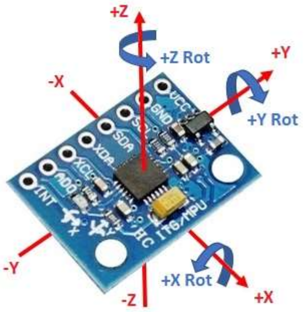


Classification of Sensors

Heading sensors

- Orientation of the robot in relation to a fixed reference frame
- Compass, Gyroscopes, Inclinometers, Accelerometer

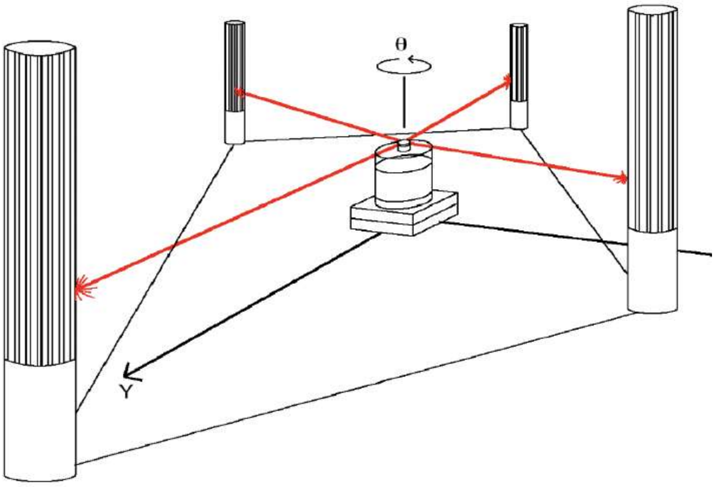




Classification of Sensors

Ground-based positioning

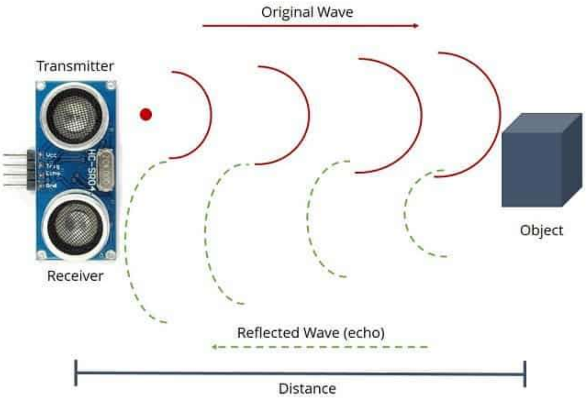
- Localization in a fixed reference frame
- GPS, Active optical or RF beacons Active ultrasonic beacons Reflective beacons

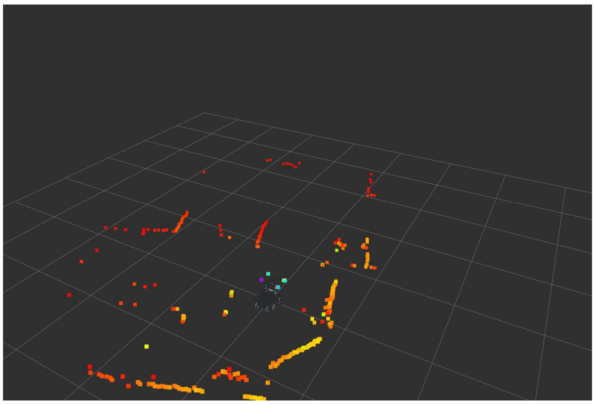


Classification of Sensors

Active ranging

- Reflectivity, time-of-flight, and geo-metric triangulation
- Reflectivity sensors, Ultrasonic sensor, Laser rangefinder

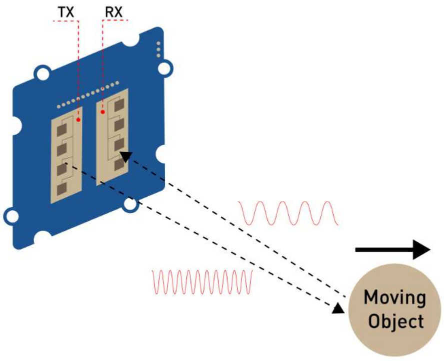


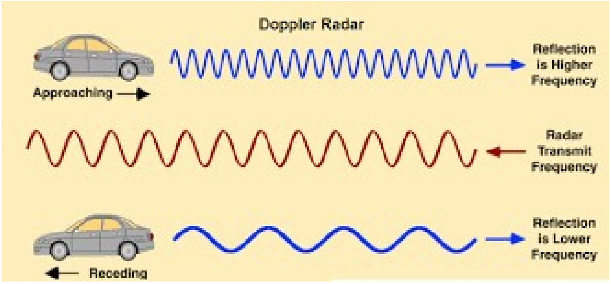


Classification of Sensors

Motion/speed sensors

- Speed relative to fixed or moving object
- Doppler radar, Doppler sound

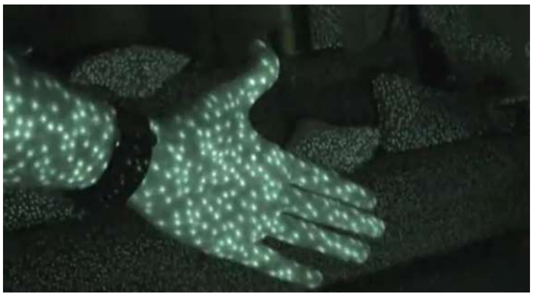




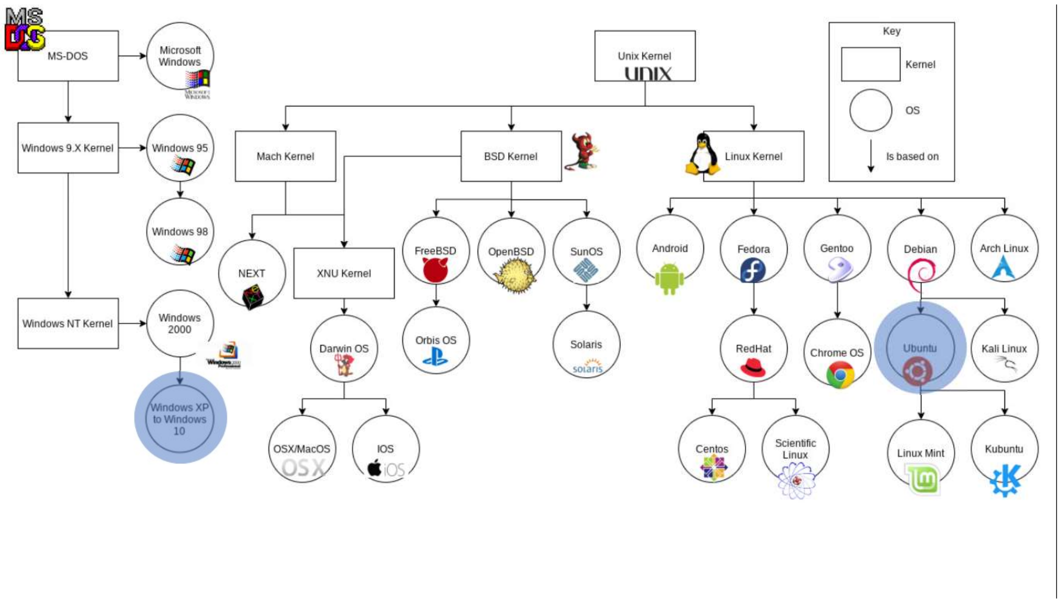
Classification of Sensors

Vision-based sensors

- Visual ranging, whole-image analysis, segmentation, object recognition
- CCD/CMOS camera Visual ranging packages Object tracking packages



ROS2 Operating System Support



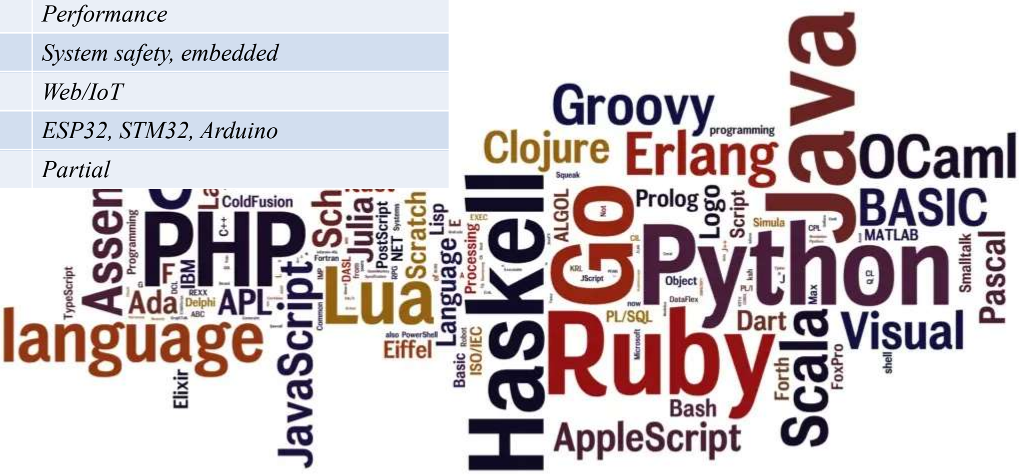
MS-DOS
Microsoft Windows
Windows 9.X Kernel
Windows 95
Windows 98
Windows NT Kernel
Windows 2000
Windows XP to Windows 10
Unix Kernel
UNIX
Mach Kernel
BSD Kernel
Linux Kernel
NEXT
XNU Kernel
FreeBSD
OpenBSD
SunOS
Darwin OS
Orbis OS
Solaris
OSX/MacOS
IOS
Android
Fedora
Gentoo
Debian
Arch Linux
RedHat
Chrome OS
Ubuntu
Kali Linux
Centos
Scientific Linux
Linux Mint
Kubuntu
Key
Kernel
OS
Is based on

Ubuntu (Open-source Operating System)

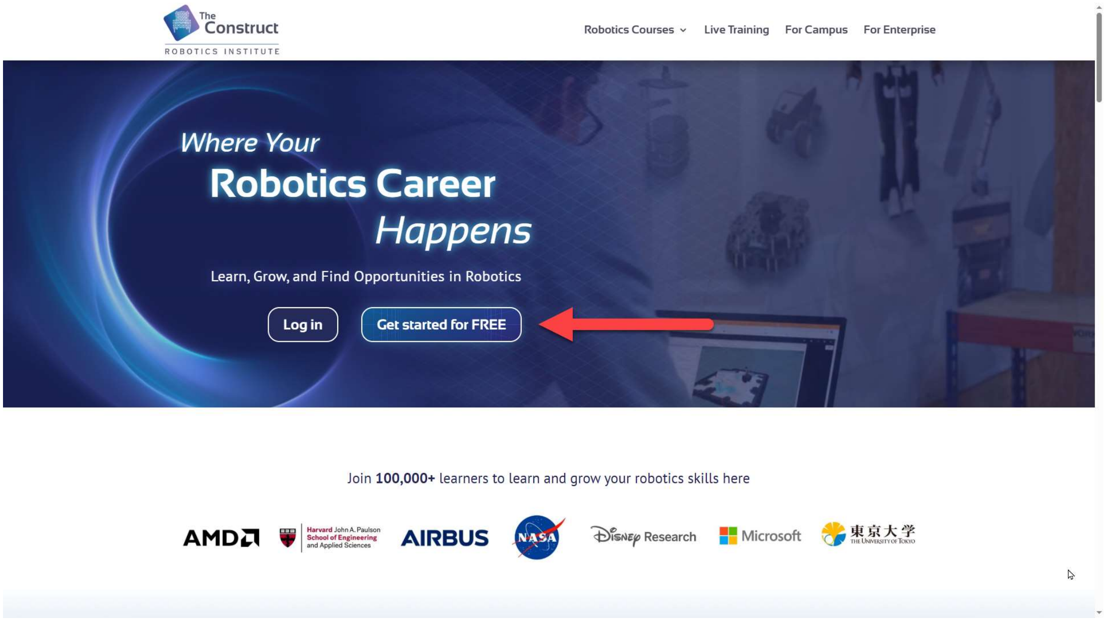
- Free to Use
- Better Community Support
- Secure
- Can revive older computers
- Perfect For Programmers
- Software Updates
- Customization

ROS2 Programming language support

Language	support
C++	*Performance*
Python	*Performance*
Rust	*System safety, embedded*
Node.js	*Web/IoT*
C (RTOS / MCU)	*ESP32, STM32, Arduino*
Java, Go, Lua	*Partial*



https://www.theconstruct.ai

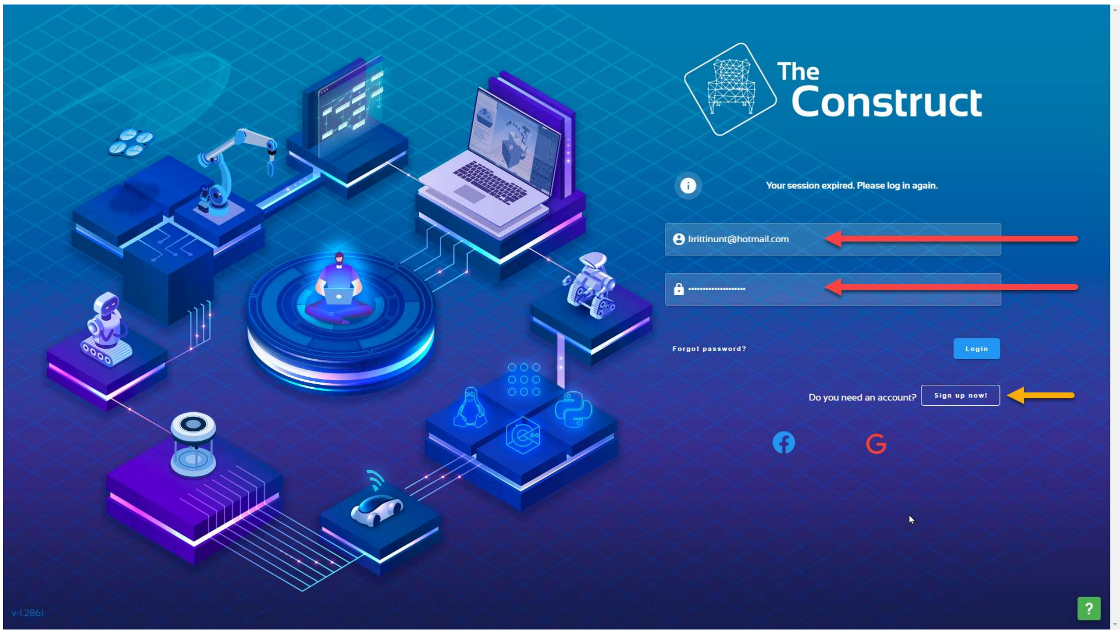


Where Your Robotics Career Happens

Learn, Grow, and Find Opportunities in Robotics

Join **100,000+** learners to learn and grow your robotics skills here

https://www.theconstruct.ai



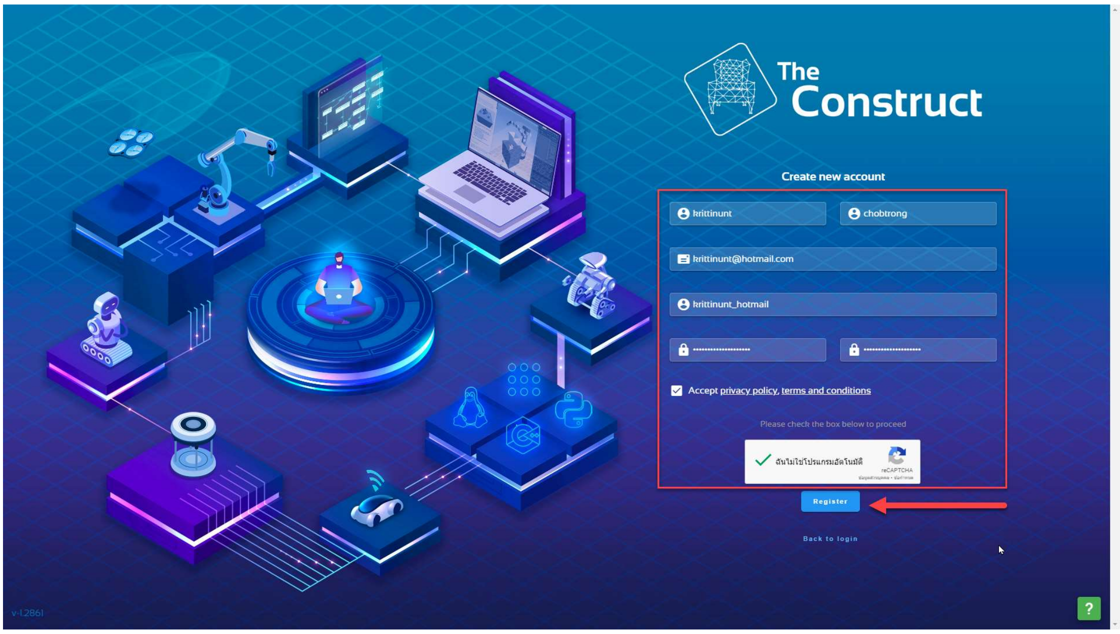
The Construct

Your session expired. Please log in again.

krittinunt@hotmail.com

••••••••••••••••••••

https://www.theconstruct.ai



The Construct

Create new account

krittinunt

chobtrong

krittinunt@hotmail.com

krittinunt_hotmail

••••••••••••••••••••

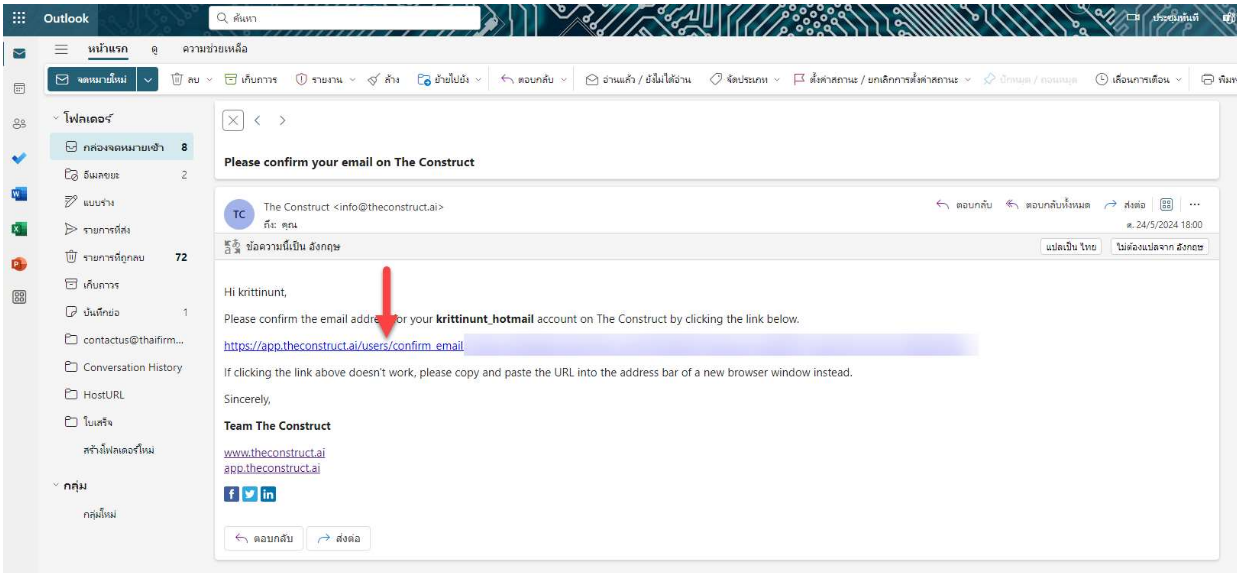
••••••••••••••••••••

☑ Accept privacy policy, terms and conditions

Please check the box below to proceed

✓ ฉันไม่ใช่โปรแกรมอัตโนมัติ

https://www.theconstruct.ai



Please confirm your email on The Construct

The Construct <info@theconstruct.ai>
ถึง: คุณ
ศ. 24/5/2024 18:00

Hi krittinunt,

Please confirm the email addre for your **krittinunt_hotmail** account on The Construct by clicking the link below.

https://app.theconstruct.ai/users/confirm_email

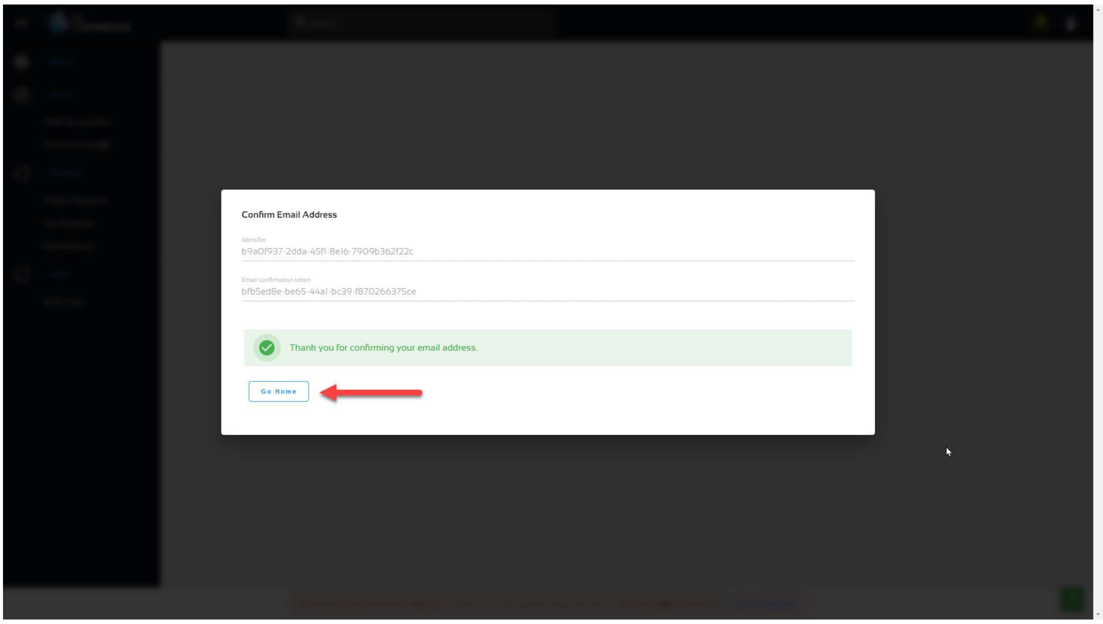
If clicking the link above doesn't work, please copy and paste the URL into the address bar of a new browser window instead.

Sincerely,

Team The Construct

www.theconstruct.ai
app.theconstruct.ai

https://www.theconstruct.ai



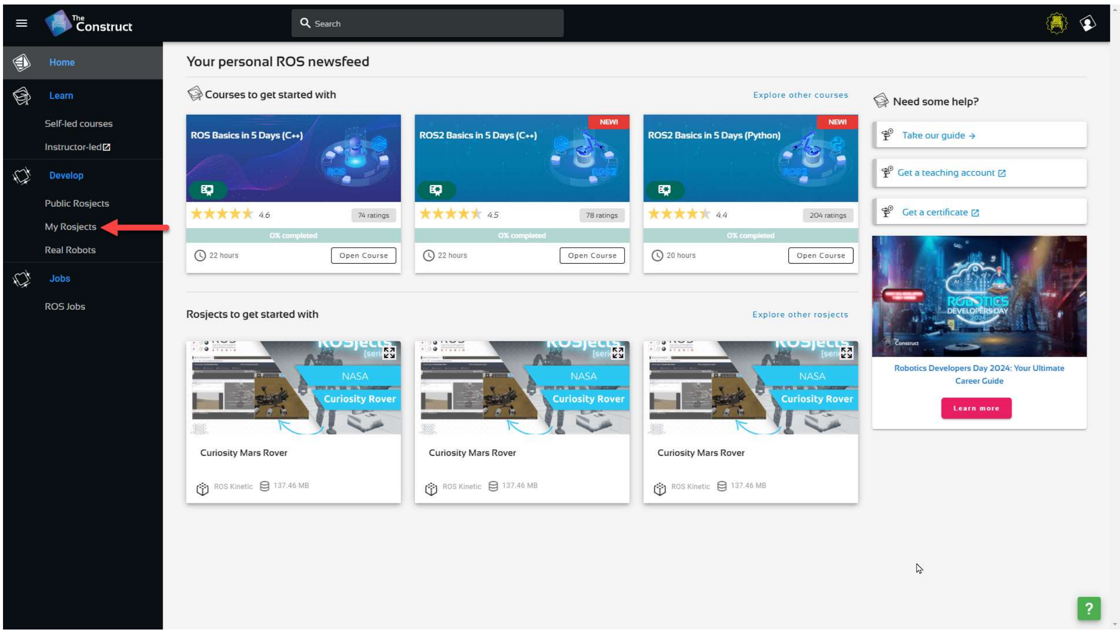
Confirm Email Address

Identifier
b9a0f937-2dda-45f1-8e16-7909b362f22c

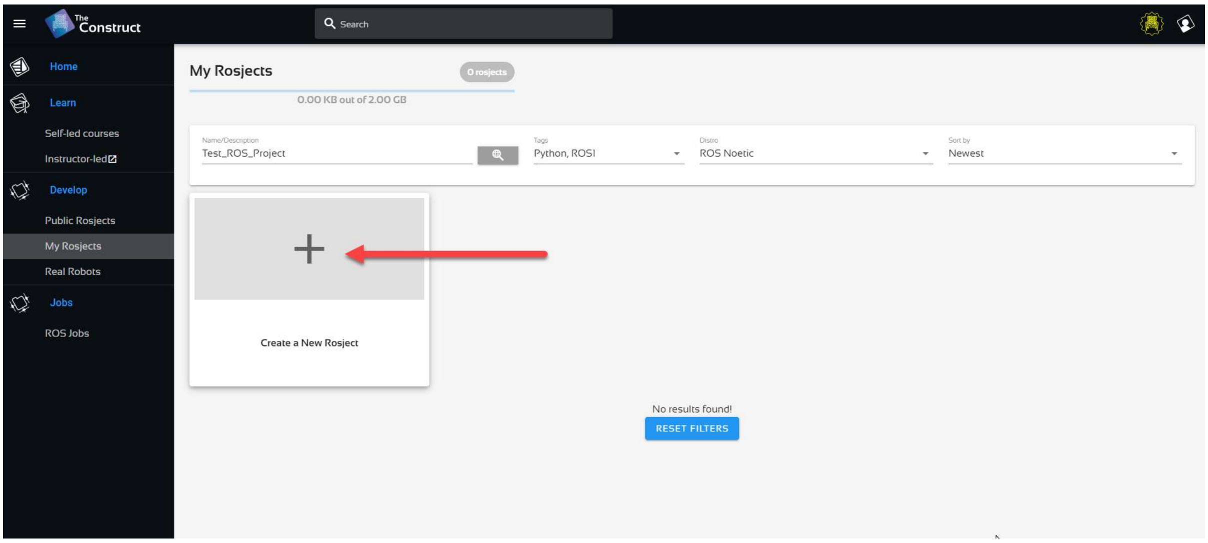
Email confirmation token
bfb5ed8e-be65-44a1-bc39-f870266375ce

Thank you for confirming your email address.

https://www.theconstruct.ai



https://www.theconstruct.ai



Name/Description: Test_ROS_Project

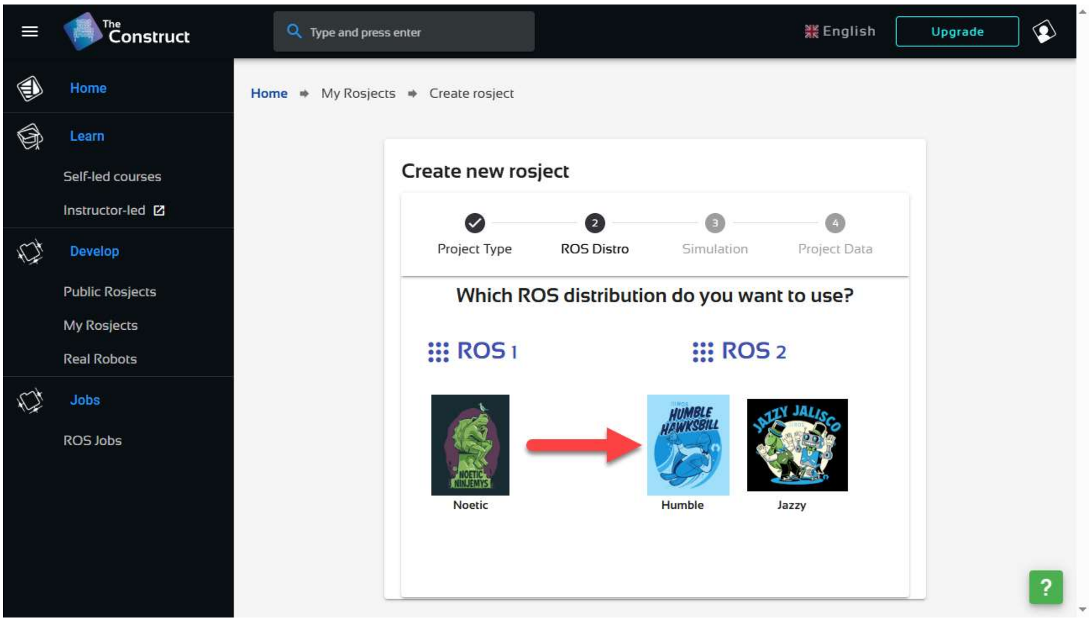
Tags: Python, ROS1

Distro: ROS Noetic

Sort by: Newest

No results found!

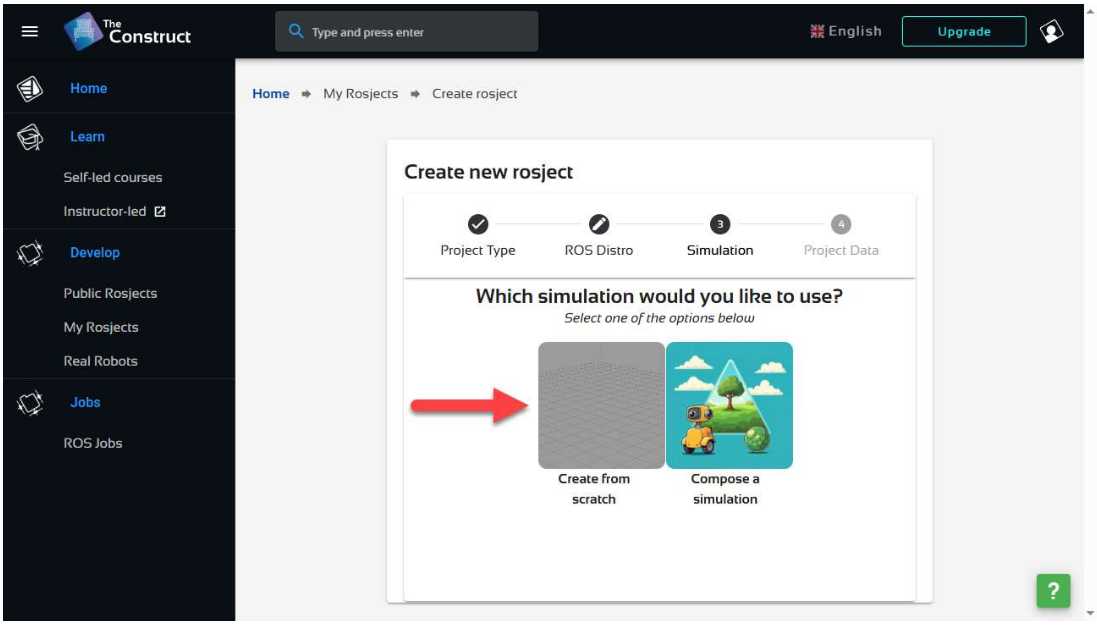
https://www.theconstruct.ai



Create new rosject

Which ROS distribution do you want to use?

https://www.theconstruct.ai

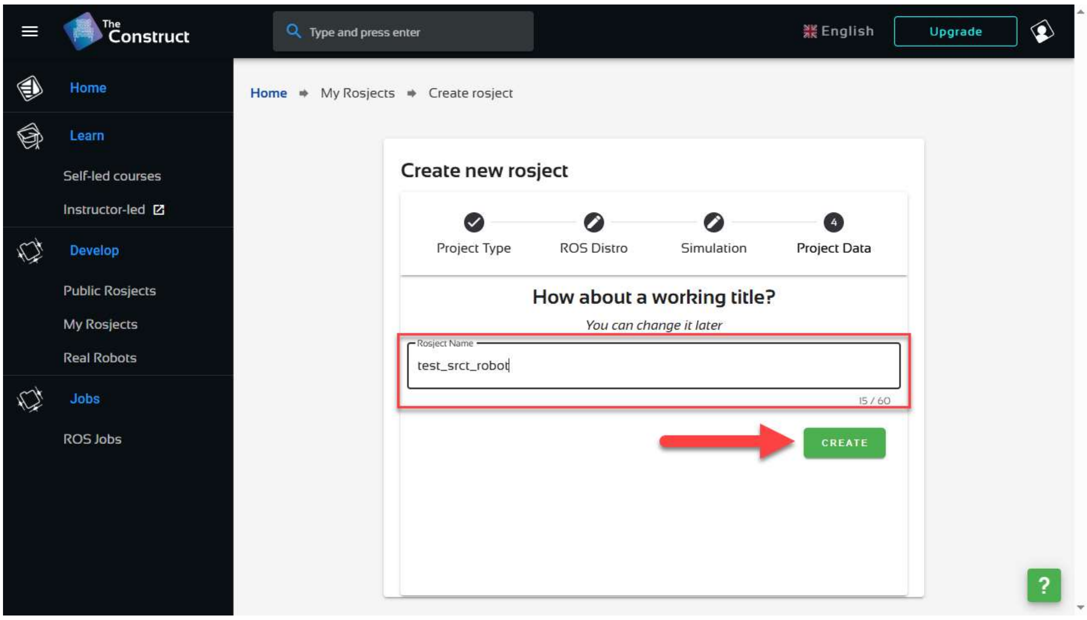


Create new rosject

Which simulation would you like to use?

Select one of the options below

https://www.theconstruct.ai



Create new rosject

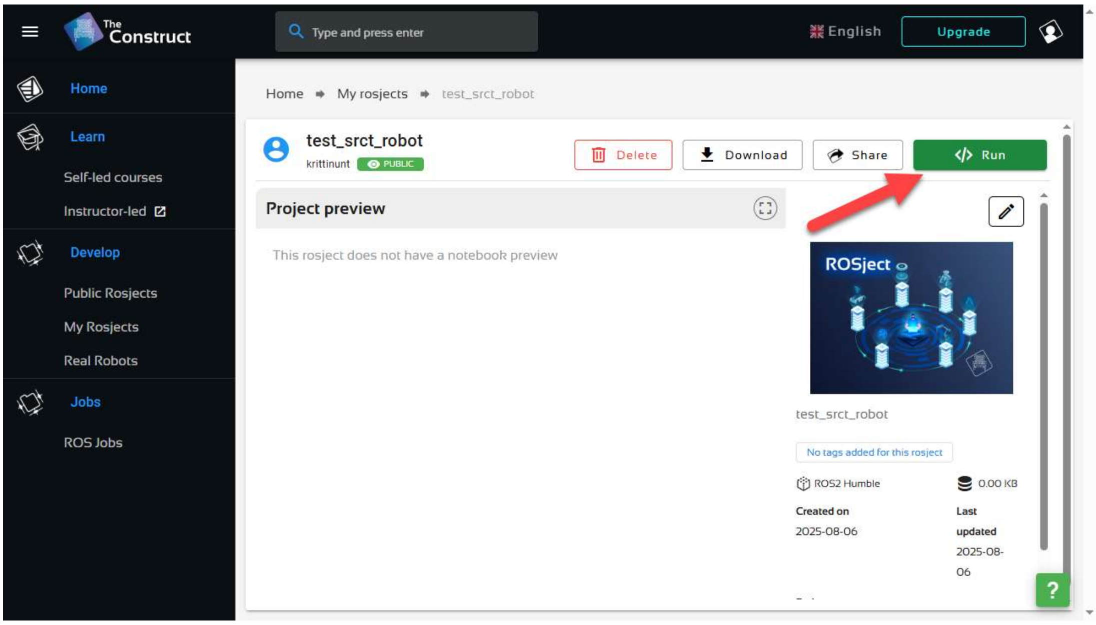
How about a working title?

You can change it later

Rosject Name: test_srct_robot

15 / 60

https://www.theconstruct.ai



test_srct_robot

krittinunt PUBLIC

Project preview

This rosject does not have a notebook preview

test_srct_robot

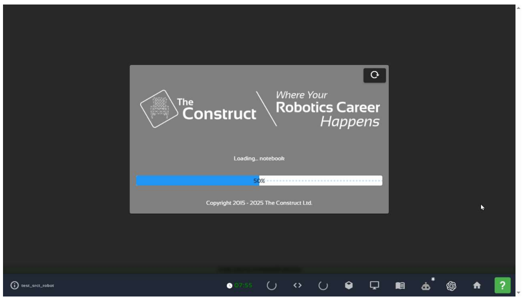
No tags added for this rosject

ROS2 Humble 0.00 KB

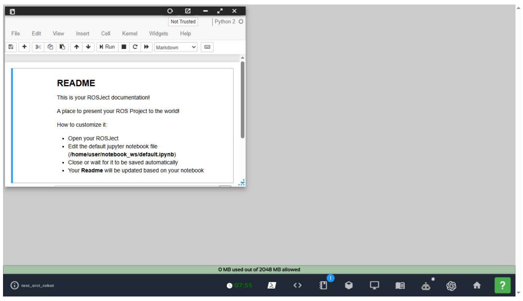
Created on 2025-08-06

Last updated 2025-08-06

https://www.theconstruct.ai



The Construct
Where Your Robotics Career Happens
Loading... notebook
50%
Copyright 2015 - 2025 The Construct Ltd.
07:55



Not Trusted
Python 2
File Edit View Insert Cell Kernel Widgets Help
Run
Markdown
0 MB used out of 2048 MB allowed
test_srct_robot
07:55

README

This is your ROSJect documentation!

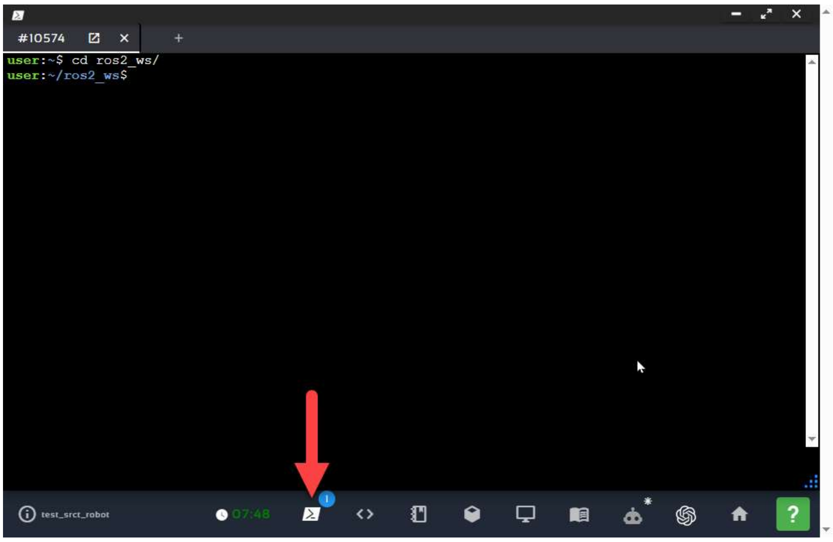
A place to present your ROS Project to the world!

How to customize it:

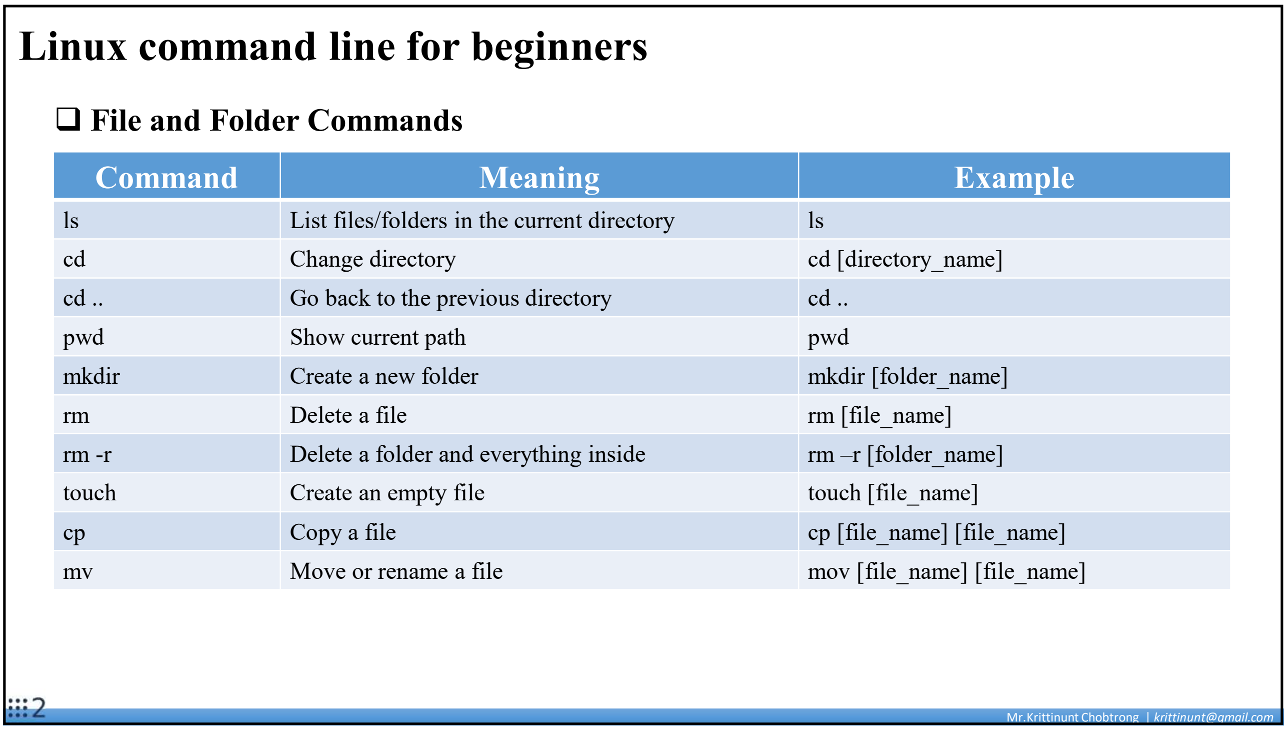
- Open your ROSJect
- Edit the default jupyter notebook file (**/home/user/notebook_ws/default.ipynb**)
- Close or wait for it to be saved automatically
- Your **Readme** will be updated based on your notebook

Run command line

❑ Open Web shell



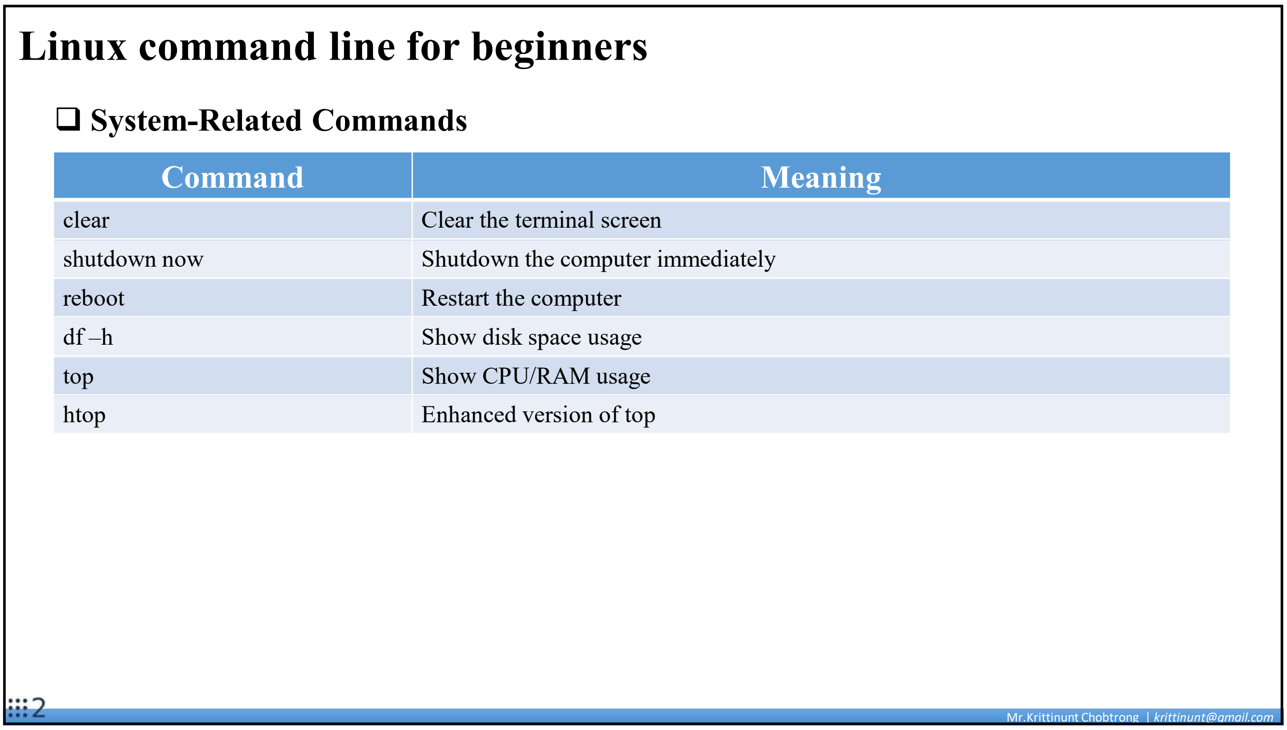
```
user:~$ cd ros2_ws/
user:~/ros2_ws$
```



Linux command line for beginners

❑ File and Folder Commands

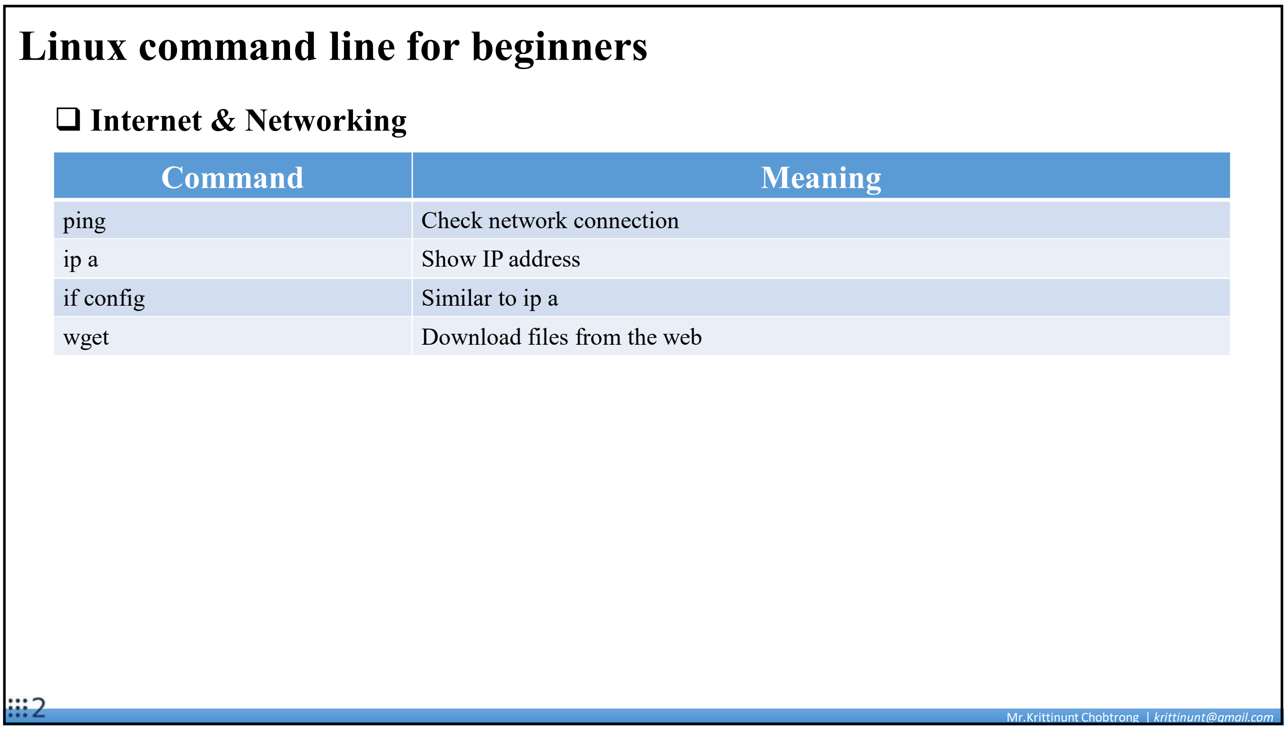
Command	Meaning	Example
ls	List files/folders in the current directory	ls
cd	Change directory	cd [directory_name]
cd ..	Go back to the previous directory	cd ..
pwd	Show current path	pwd
mkdir	Create a new folder	mkdir [folder_name]
rm	Delete a file	rm [file_name]
rm -r	Delete a folder and everything inside	rm –r [folder_name]
touch	Create an empty file	touch [file_name]
cp	Copy a file	cp [file_name] [file_name]
mv	Move or rename a file	mov [file_name] [file_name]



Linux command line for beginners

❑ System-Related Commands

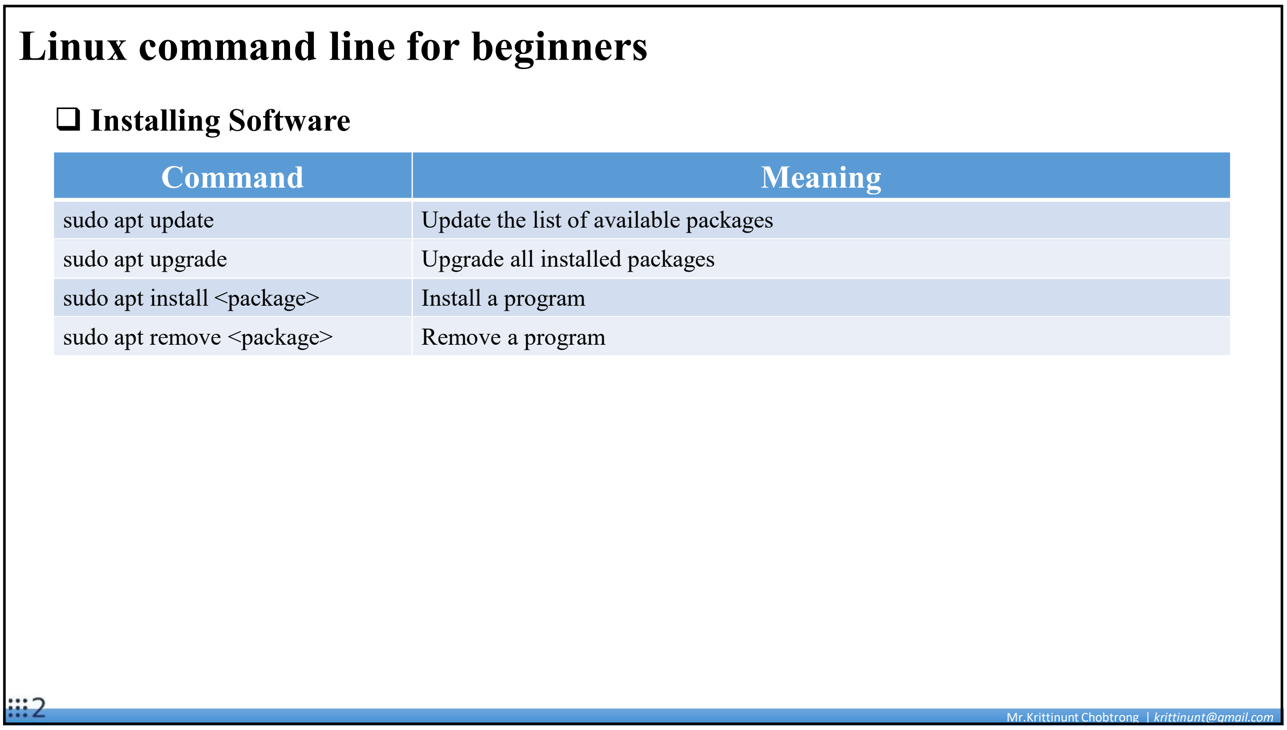
Command	Meaning
clear	Clear the terminal screen
shutdown now	Shutdown the computer immediately
reboot	Restart the computer
df –h	Show disk space usage
top	Show CPU/RAM usage
htop	Enhanced version of top



Linux command line for beginners

❑ Internet & Networking

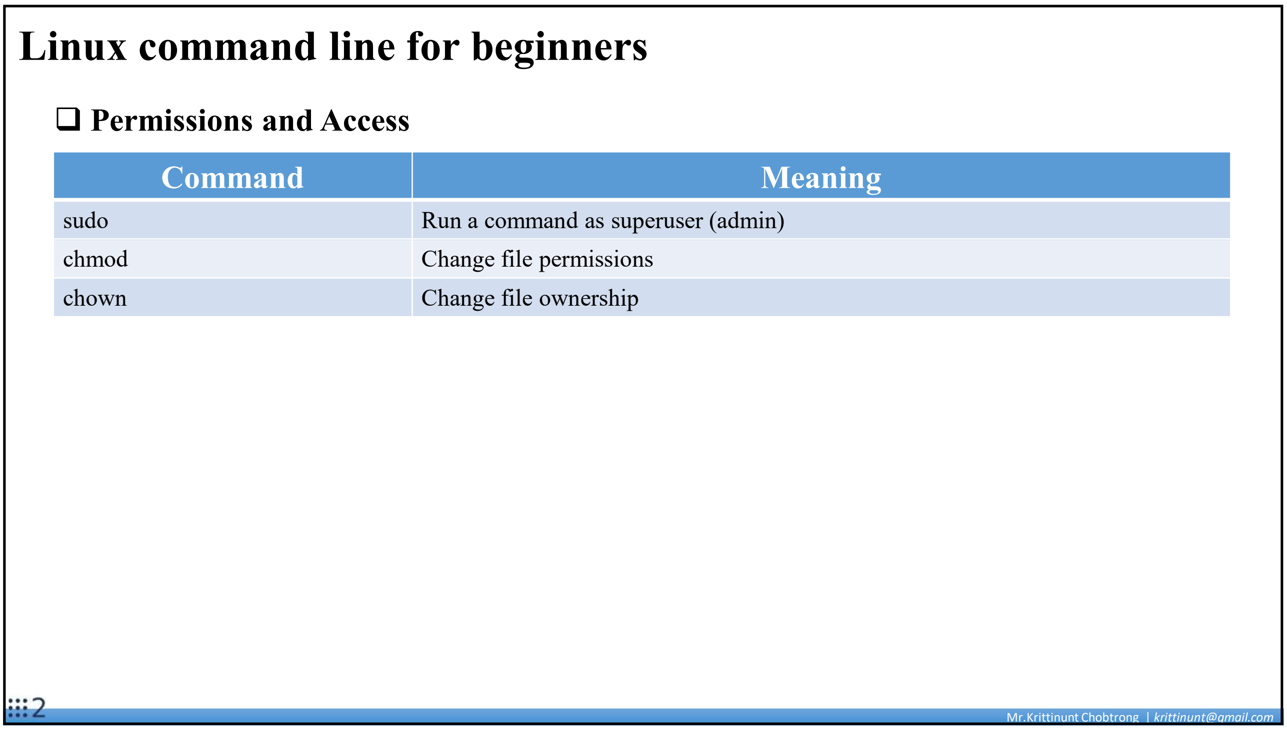
Command	Meaning
ping	Check network connection
ip a	Show IP address
if config	Similar to ip a
wget	Download files from the web



Linux command line for beginners

❑ Installing Software

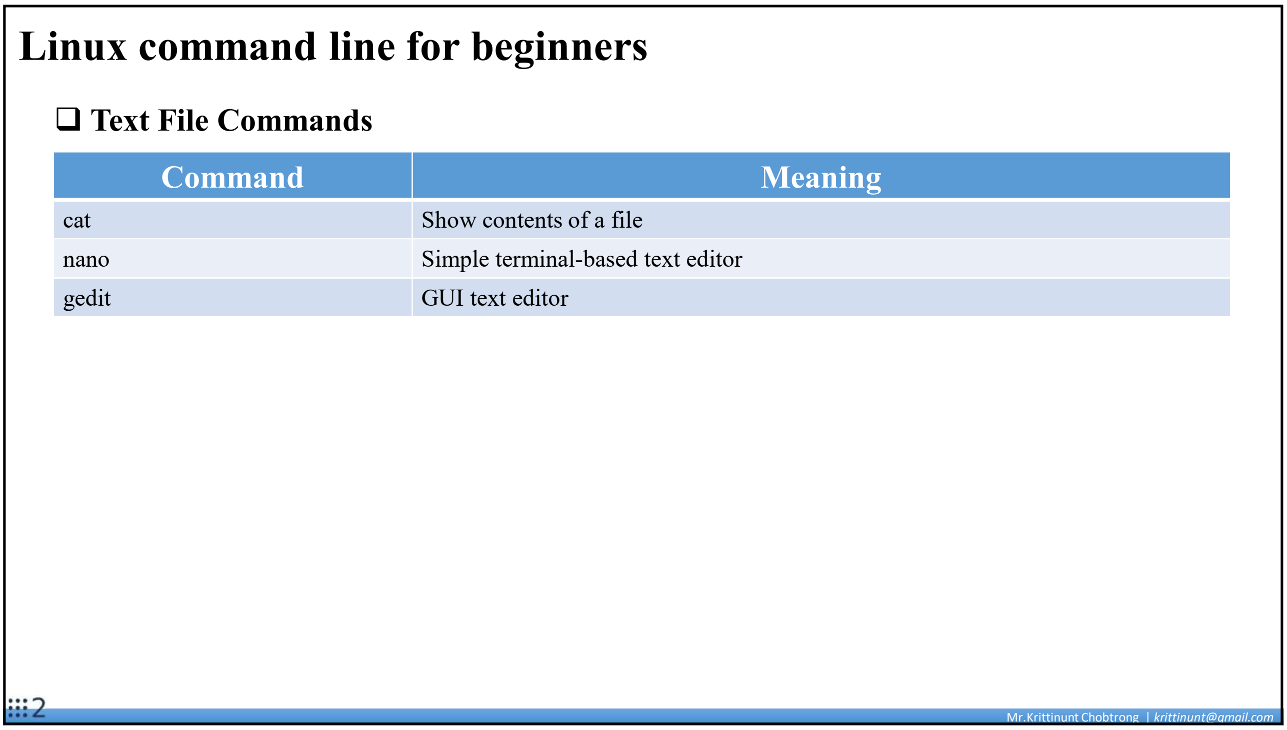
Command	Meaning
sudo apt update	Update the list of available packages
sudo apt upgrade	Upgrade all installed packages
sudo apt install <package>	Install a program
sudo apt remove <package>	Remove a program



Linux command line for beginners

❑ Permissions and Access

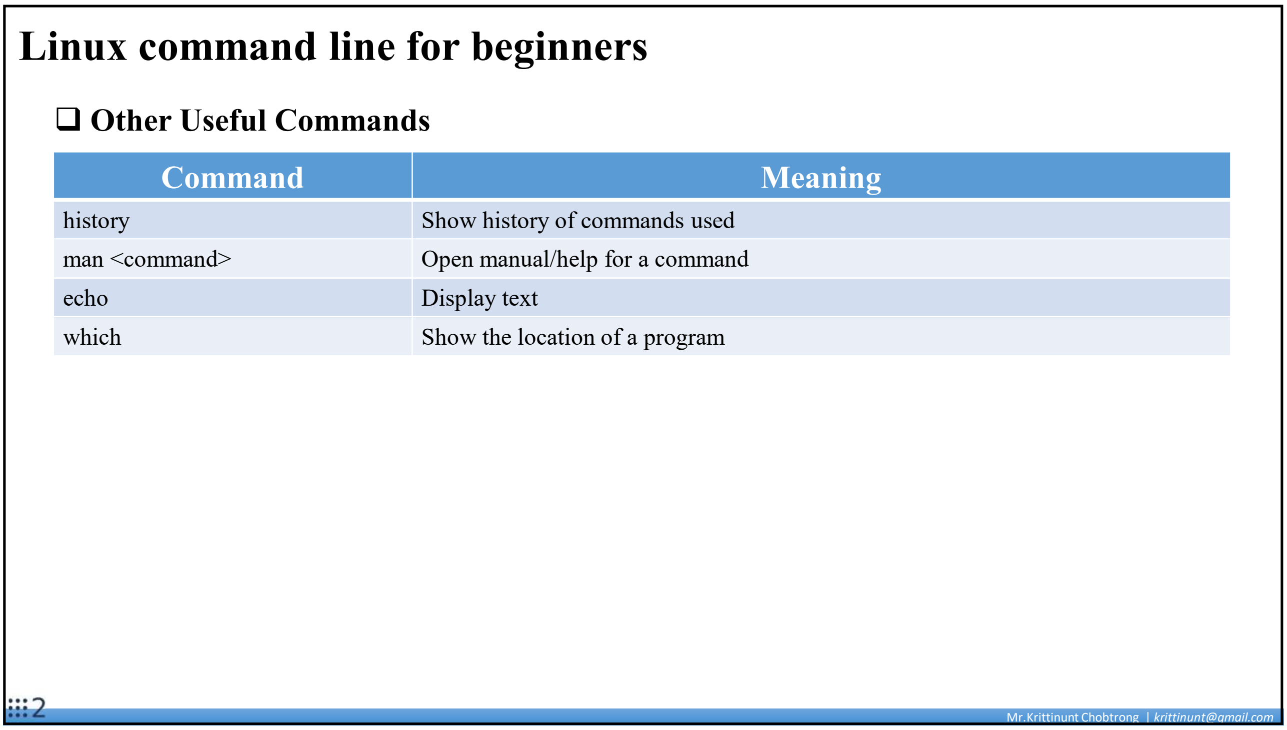
Command	Meaning
sudo	Run a command as superuser (admin)
chmod	Change file permissions
chown	Change file ownership



Linux command line for beginners

❑ Text File Commands

Command	Meaning
cat	Show contents of a file
nano	Simple terminal-based text editor
gedit	GUI text editor

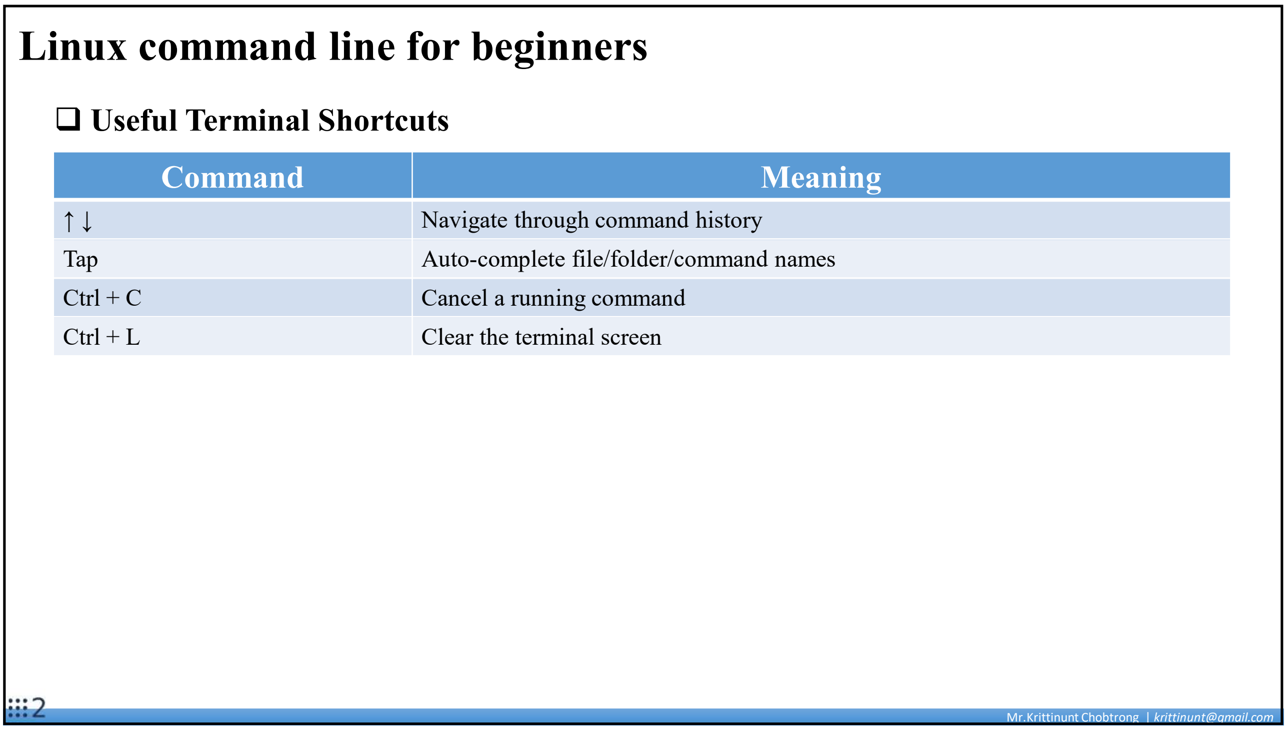


Linux command line for beginners

❑ Other Useful Commands

Command	Meaning
history	Show history of commands used
man <command>	Open manual/help for a command
echo	Display text
which	Show the location of a program

Mr.Krittinunt Chobtrong | *krittinunt@gmail.com*



Linux command line for beginners

❑ Useful Terminal Shortcuts

Command	Meaning
↑ ↓	Navigate through command history
Tap	Auto-complete file/folder/command names
Ctrl + C	Cancel a running command
Ctrl + L	Clear the terminal screen

Mr.Krittinunt Chobtrong | *krittinunt@gmail.com*

Basic Linux Commands for ROS2

❑ Check ROS2 Environment

Command	Meaning
printenv | grep ROS	Check if the ROS2 environment is sourced properly
ros2 --version	Display the installed ROS2 version

Basic Linux Commands for ROS2

❑ Start Using ROS2

Command	Meaning
source /opt/ros/humble/setup.bash	Source ROS2 environment
ros2 run <package_name> <executable_name>	Run a specific ROS2 node
ros2 pkg list	List all available packages
ros2 pkg executables <package_name>	Show all executables in a package

Basic Linux Commands for ROS2

❑ Workspace and Build

Command	Meaning
mkdir -p ~/ros2_ws/src cd ~/ros2_ws colcon build	Create a ROS 2 workspace and build it
source install/setup.bash	Source the local workspace after building

Basic Linux Commands for ROS2

❑ Node Commands

Command	Meaning
ros2 node list	List currently running nodes
ros2 node info <node_name>	Show info about a node

Basic Linux Commands for ROS2

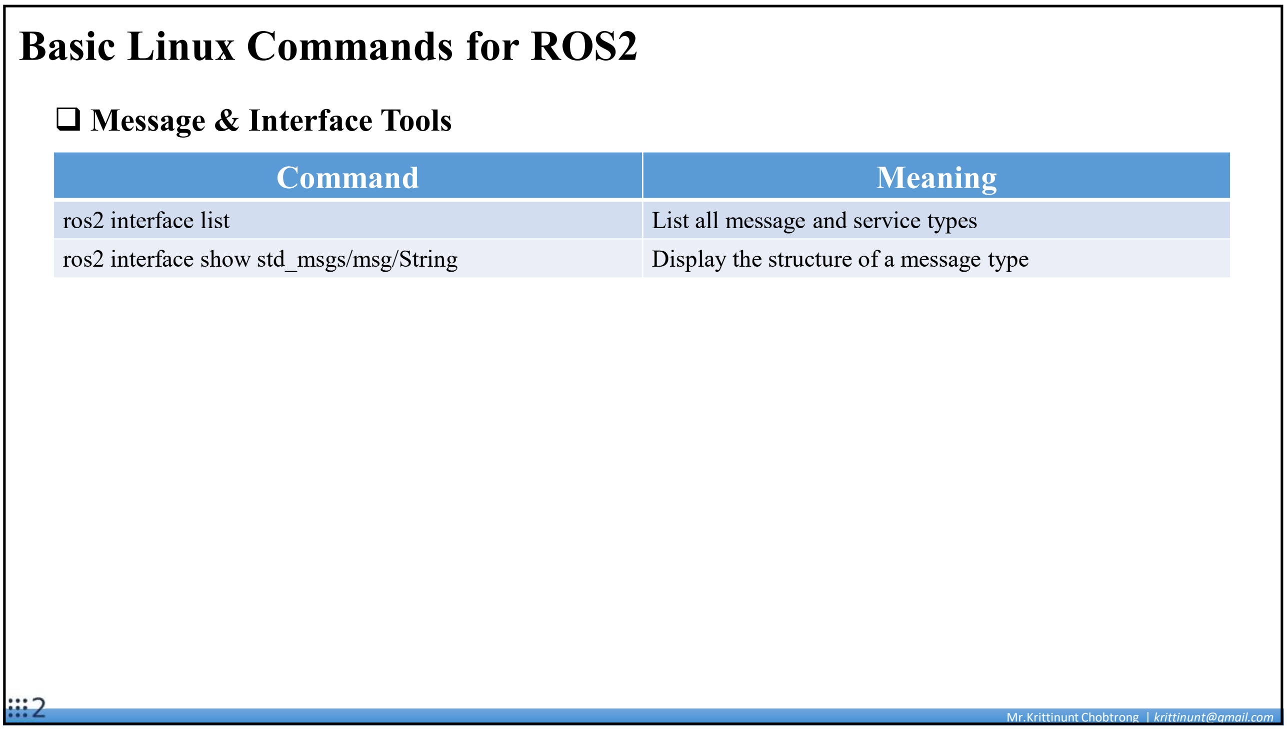
❑ Topic Commands

Command	Meaning
ros2 topic list	List all available topics
ros2 topic echo <topic_name>	View live data being published on a topic
ros2 topic pub <topic_name> <msg_type> '{data}'	Publish data to a topic

Basic Linux Commands for ROS2

❑ Service Commands

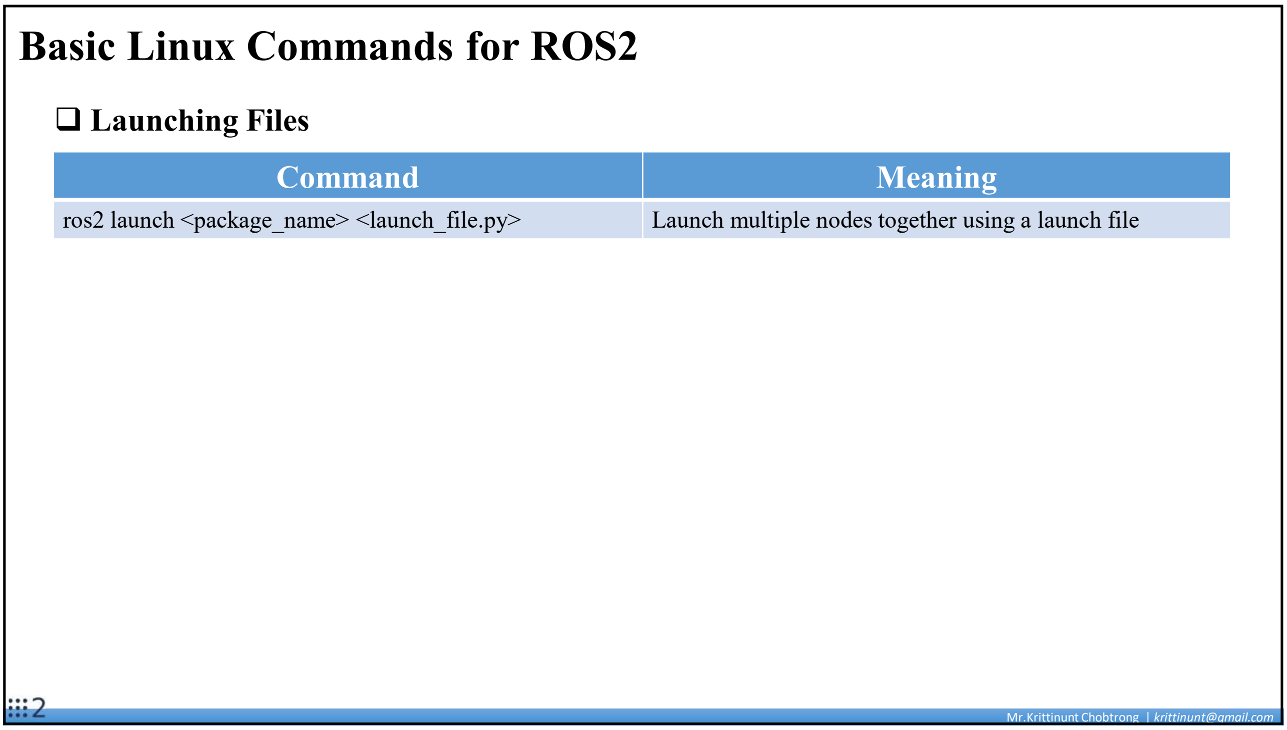
Command	Meaning
ros2 service list	List all available topics
ros2 service type <service_name>	Show the service type
ros2 service call <service_name> <srv_type> '{data}'	Call a service



Basic Linux Commands for ROS2

❑ Message & Interface Tools

Command	Meaning
ros2 interface list	List all message and service types
ros2 interface show std_msgs/msg/String	Display the structure of a message type



Basic Linux Commands for ROS2

❑ Launching Files

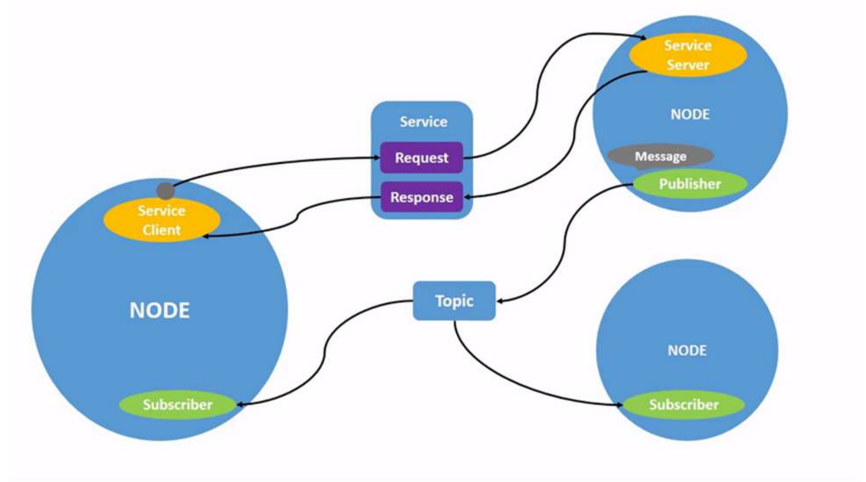
Command	Meaning
ros2 launch <package_name> <launch_file.py>	Launch multiple nodes together using a launch file

Basic Linux Commands for ROS2

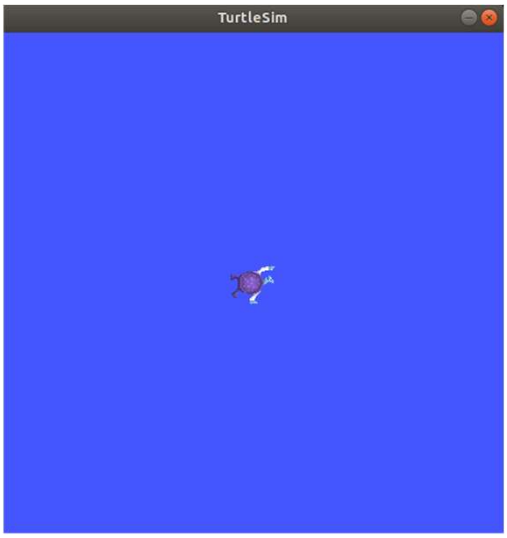
❑ Clean and Rebuild Workspace

Command	Meaning
colcon build --clean	Clean the workspace and rebuild from scratch

ROS2 basic graph structure

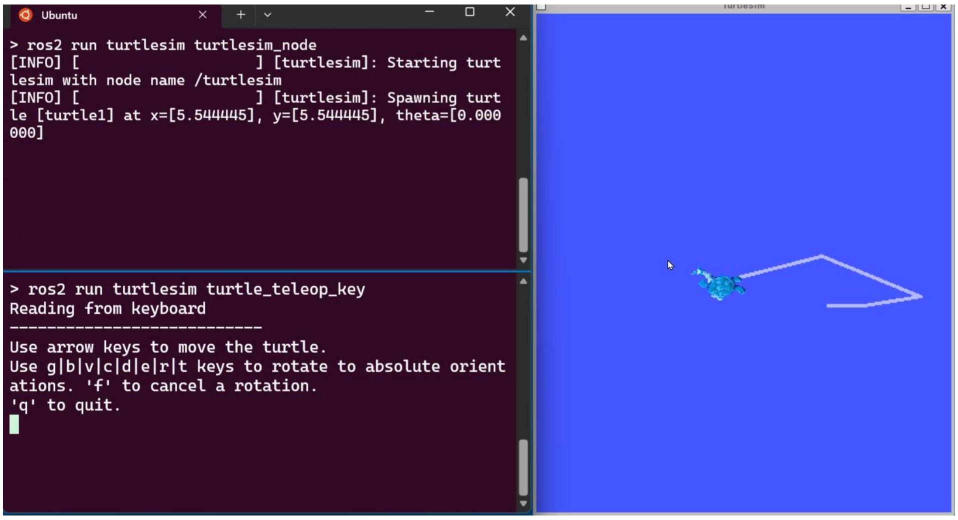


ROS2 Turtle sim



```
$ cd ros2_ws
$ ros2 run turtlesim turtlesim_node
```

ROS2 Teleop key

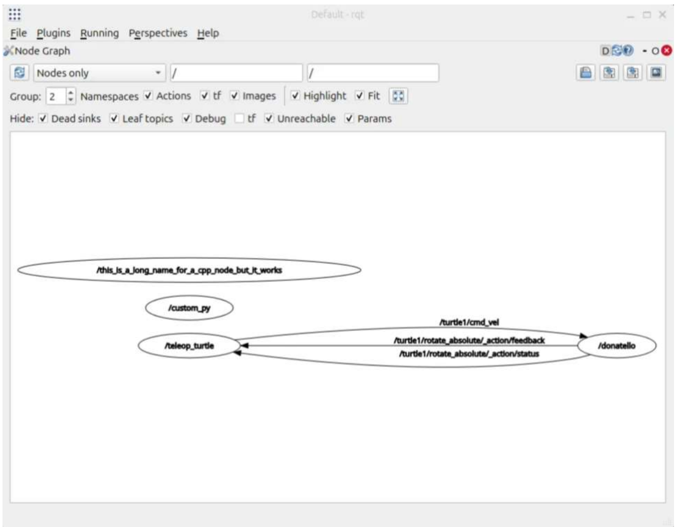


```
> ros2 run turtlesim turtlesim_node
[INFO] [                           ] [turtlesim]: Starting turt
lesim with node name /turtlesim
[INFO] [                           ] [turtlesim]: Spawning turt
le [turtle1] at x=[5.544445], y=[5.544445], theta=[0.000
000]
```

```
> ros2 run turtlesim turtle_teleop_key
Reading from keyboard
---------------------------
Use arrow keys to move the turtle.
Use g|b|v|c|d|e|r|t keys to rotate to absolute orient
ations. 'f' to cancel a rotation.
'q' to quit.
```

```
$ ros2 run turtlesim turtle_teleop_key
```

ROS2 Rqt Graphs


Default - rqt
File Plugins Running Perspectives Help
Node Graph
D
/this_is_a_long_name_for_a_cpp_node_but_it_works
/custom_py
/teleop_turtle
/turtle1/cmd_vel
/turtle1/rotate_absolute/_action/feedback
/turtle1/rotate_absolute/_action/status
/donatello

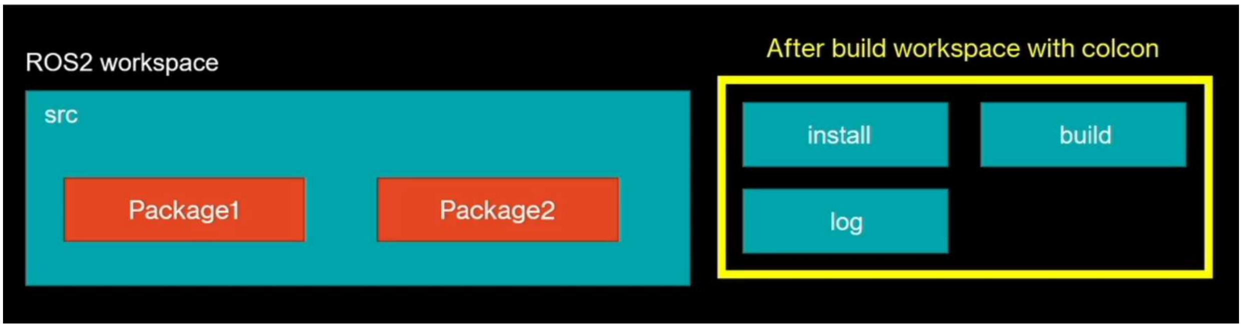
Nodes only / /

Group: 2 Namespaces ☑ Actions ☑ tf ☑ Images ☑ Highlight ☑ Fit

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

```
$ rqt_graph
```

ROS2 Workspace



ROS2 workspace

src

Package1

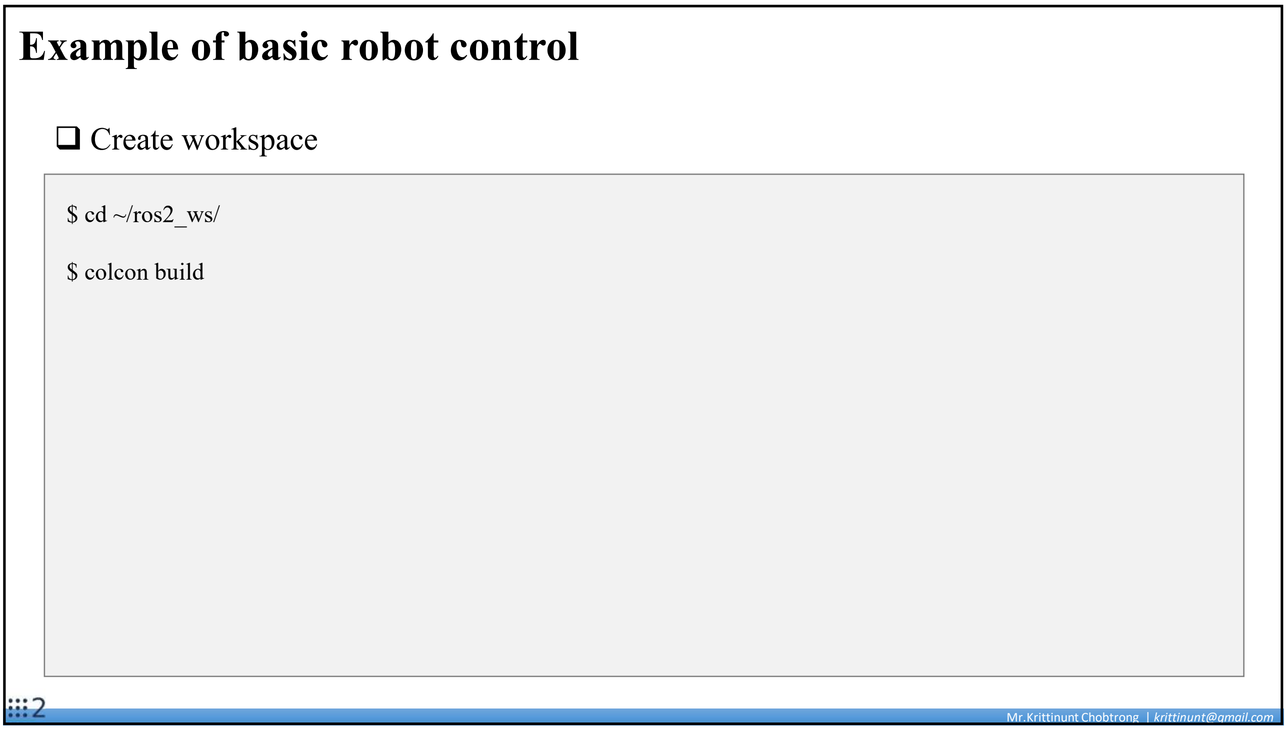
Package2

After build workspace with colcon

install

build

log

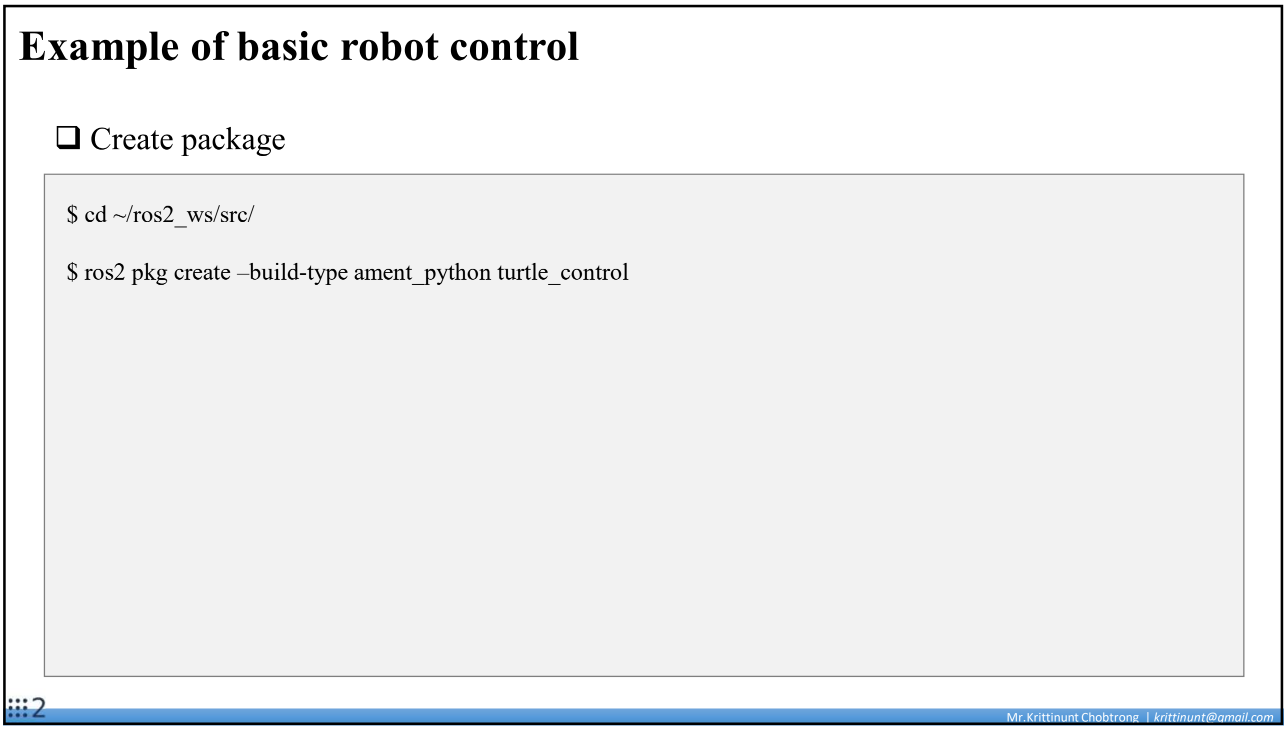


Example of basic robot control

❑ Create workspace

```
$ cd ~/ros2_ws/

$ colcon build
```



Example of basic robot control

❑ Create package

```
$ cd ~/ros2_ws/src/

$ ros2 pkg create –build-type ament_python turtle_control
```

Example of basic robot control

❑ Create file control_turtle.py

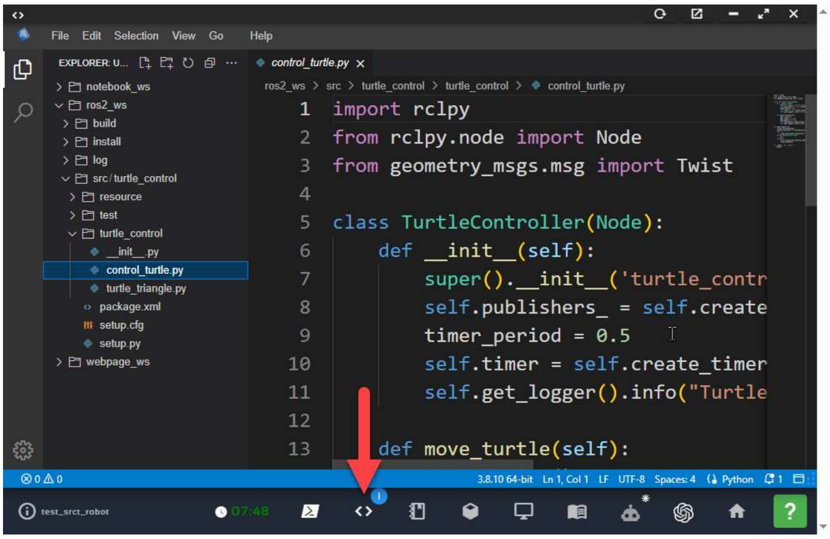
```
$ cd ~/ros2_ws/src/turtle_control/turtle_control/

$ touch control_turtle.py

$ chmod +x control_turtle.py
```

Example of basic robot control

❑ Open Code editor



```
import rclpy
from rclpy.node import Node
from geometry_msgs.msg import Twist

class TurtleController(Node):
    def __init__(self):
        super().__init__('turtle_contr
        self.publishers_ = self.create
        timer_period = 0.5
        self.timer = self.create_timer
        self.get_logger().info("Turtle

    def move_turtle(self):
```

Example of basic robot control

❑ Write code in file control_turtle.py

```
import rclpy
from rclpy.node import Node
from geometry_msgs.msg import Twist

class TurtleController(Node):
    def __init__(self):
        super().__init__('turtle_controller')
        self.publisher_ = self.create_publisher(Twist, '/turtle1/cmd_vel', 10)
        timer_period = 0.5
      self.timer = self.create_timer(timer_period, self.move_turtle)
        self.get_logger().info("Turtle Controller Started.")

    def move_turtle(self):
        msg = Twist()
        msg.linear.x = 1.0
      msg.angular.z = 0.5
      self.publisher_.publish(msg)
        self.get_logger().info('Publishing: Linear=%.2f Angular=%.2f' % (msg.linear.x, msg.angular.z))
```

```
def main(args=None):
    rclpy.init(args=args)
    turtle_controller = TurtleController()
    rclpy.spin(turtle_controller)
    turtle_controller.destroy_node()
    rclpy.shutdown()

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

Example of basic robot control

❑ Edit file setup.py by add entry point

```
entry_points={
        'console_scripts': [
        'control_turtle = turtle_control.control_turtle:main',
        ],
    },
```

Example of basic robot control

❑ Install dependencies

```
$ cd ~/ros2_ws

$ colcon build

$ source install/setup.bash
```

Example of basic robot control

❑ New web shell and run turtlesim

```
$ cd ~/ros2_ws

$ ros2 run turtlesim turtlesim_node
```

Example of basic robot control

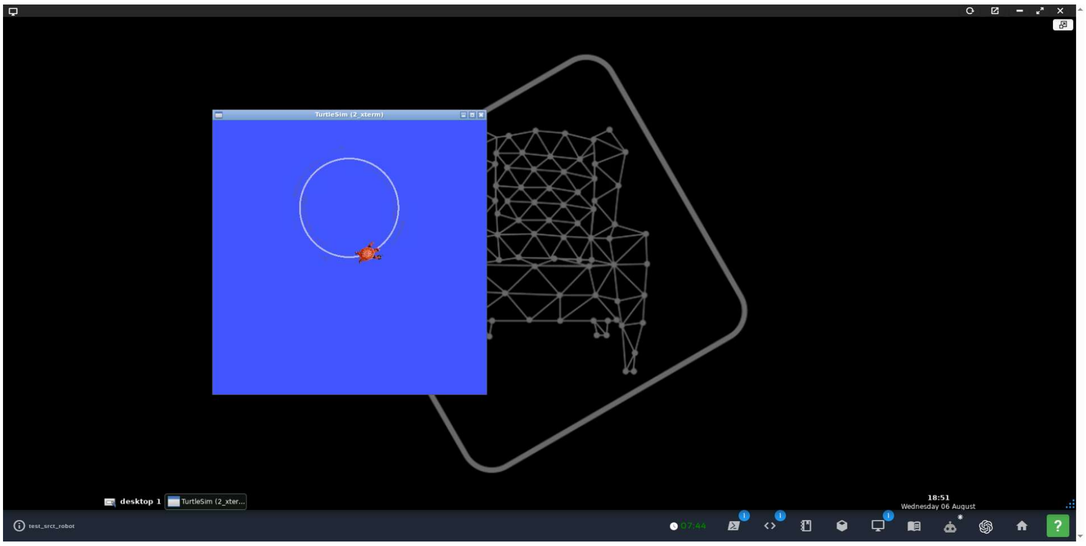
❑ New web shell and run

```
$ cd ~/ros2_ws

$ ros2 run turtle_control control_turtle
```

Example of basic robot control

❑ Output



Example of basic robot control

❑ Clear screen

```
$ cd ~/ros2_ws

$ ros2 service call /clear std_srvs/srv/Empty
```

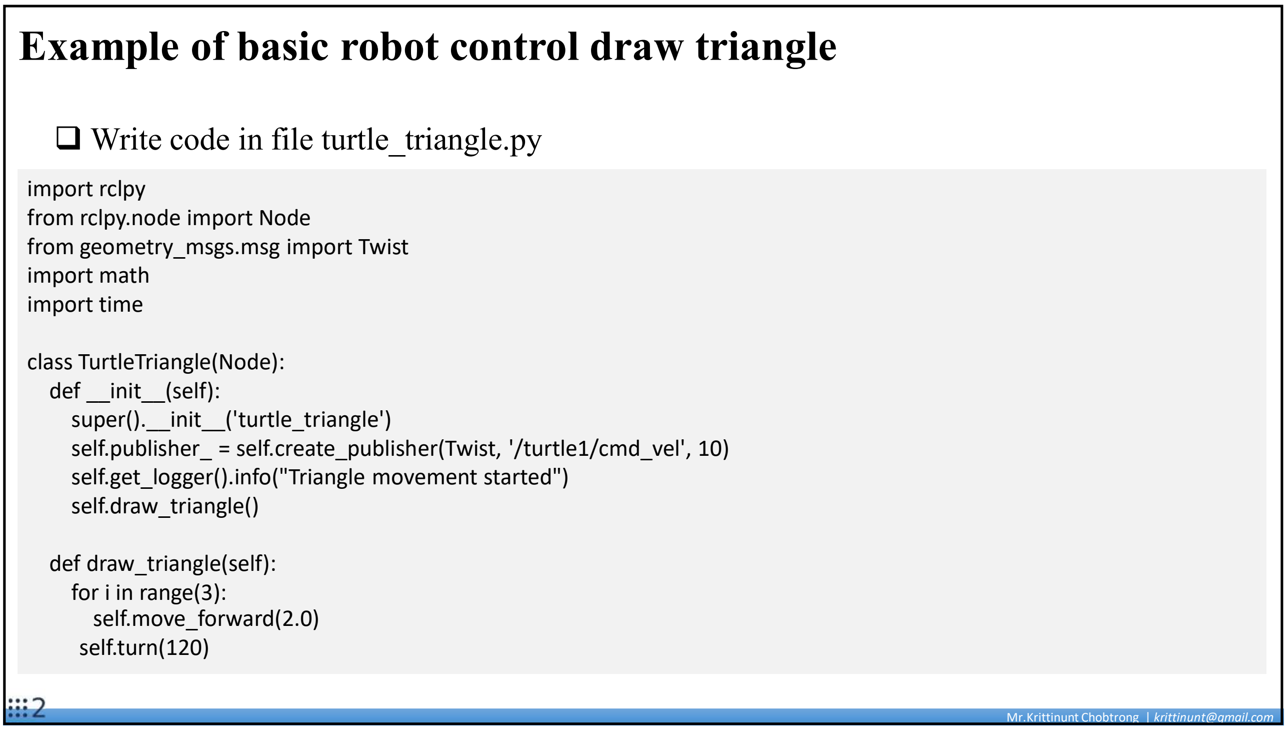
Example of basic robot control draw triangle

❑ Create file turtle_triangle.py

```
$ cd ~/ros2_ws/src/turtle_control/turtle_control/

$ touch turtle_triangle.py

$ chmod +x turtle_triangle.py
```



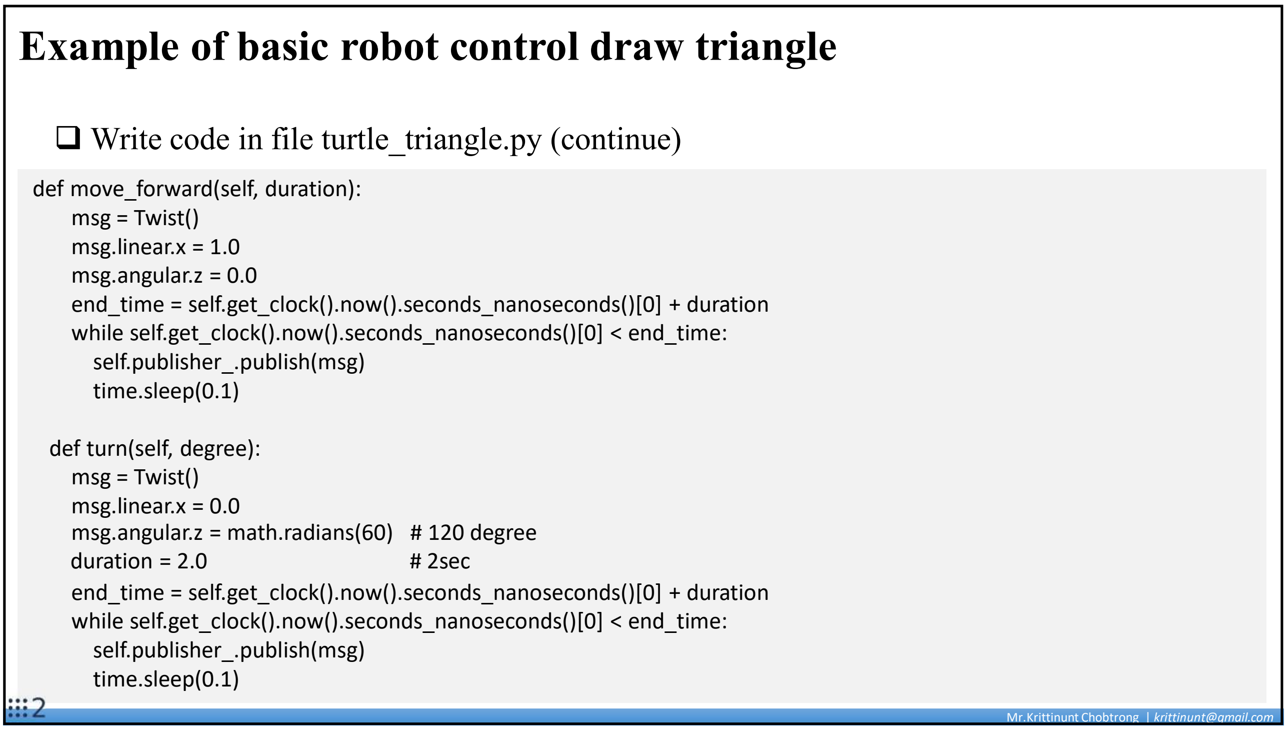
Example of basic robot control draw triangle

- ❑ Write code in file turtle_triangle.py

```
import rclpy
from rclpy.node import Node
from geometry_msgs.msg import Twist
import math
import time

class TurtleTriangle(Node):
  def __init__(self):
    super().__init__('turtle_triangle')
    self.publisher_ = self.create_publisher(Twist, '/turtle1/cmd_vel', 10)
    self.get_logger().info("Triangle movement started")
    self.draw_triangle()

  def draw_triangle(self):
    for i in range(3):
      self.move_forward(2.0)
     self.turn(120)
```

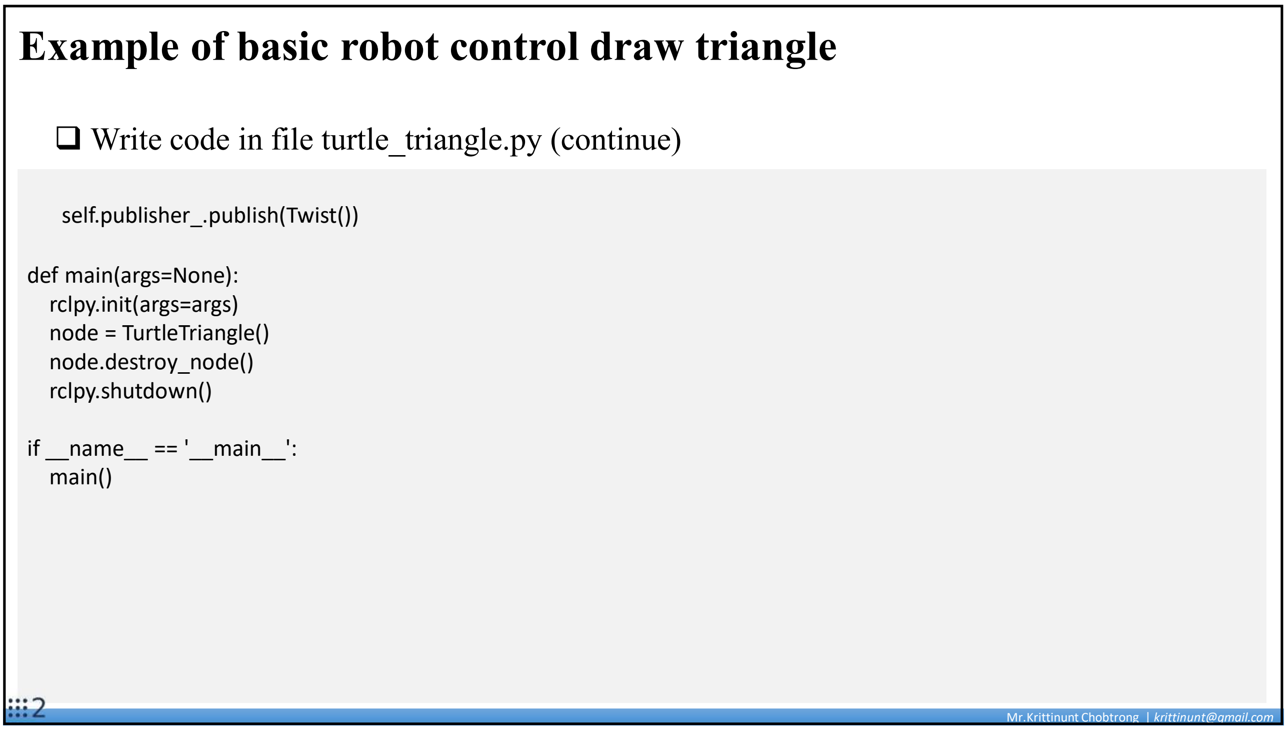


Example of basic robot control draw triangle

- ❑ Write code in file turtle_triangle.py (continue)

```
 def move_forward(self, duration):
    msg = Twist()
    msg.linear.x = 1.0
    msg.angular.z = 0.0
    end_time = self.get_clock().now().seconds_nanoseconds()[0] + duration
    while self.get_clock().now().seconds_nanoseconds()[0] < end_time:
      self.publisher_.publish(msg)
      time.sleep(0.1)

  def turn(self, degree):
    msg = Twist()
    msg.linear.x = 0.0
    msg.angular.z = math.radians(60)  # 120 degree
    duration = 2.0                    # 2sec
    end_time = self.get_clock().now().seconds_nanoseconds()[0] + duration
    while self.get_clock().now().seconds_nanoseconds()[0] < end_time:
      self.publisher_.publish(msg)
      time.sleep(0.1)
```



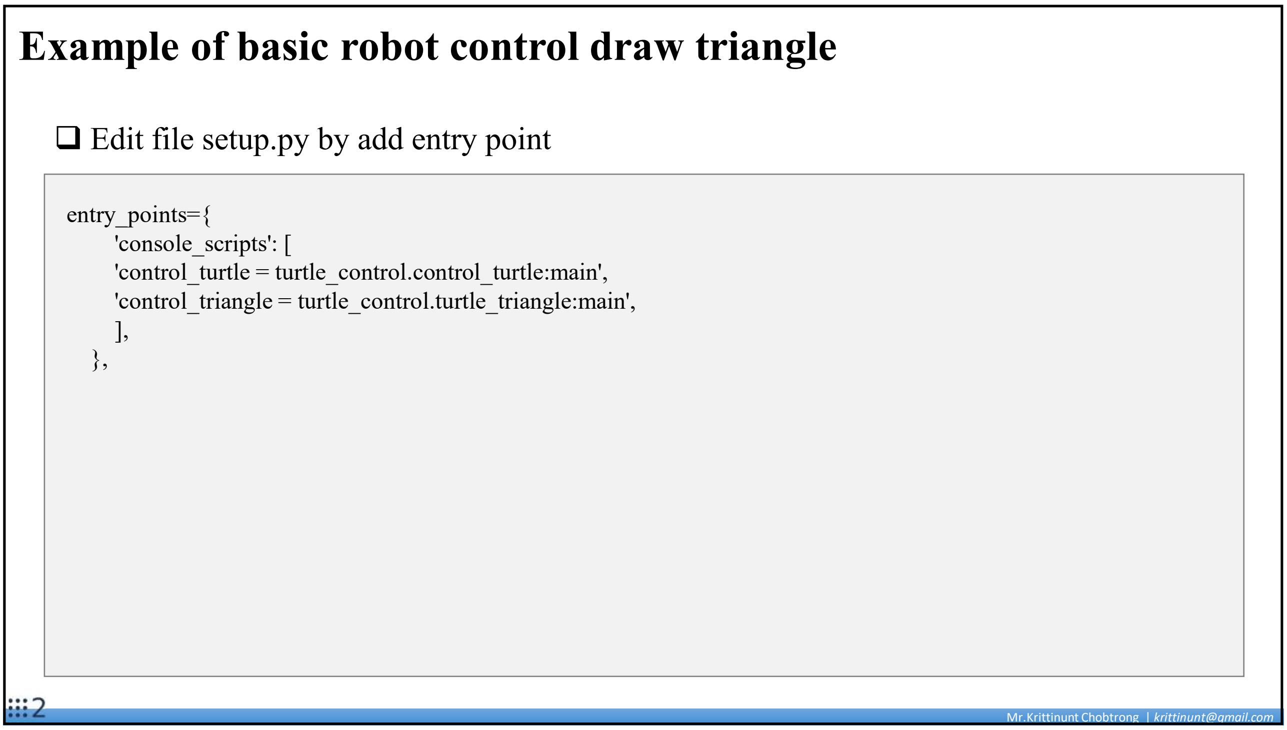
Example of basic robot control draw triangle

❑ Write code in file turtle_triangle.py (continue)

```
        self.publisher_.publish(Twist())

def main(args=None):
    rclpy.init(args=args)
    node = TurtleTriangle()
    node.destroy_node()
    rclpy.shutdown()

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



Example of basic robot control draw triangle

❑ Edit file setup.py by add entry point

```
entry_points={
        'console_scripts': [
        'control_turtle = turtle_control.control_turtle:main',
        'control_triangle = turtle_control.turtle_triangle:main',
        ],
    },
```

Example of basic robot control draw triangle

❑ Install dependencies

```
$ cd ~/ros2_ws

$ colcon build

$ source install/setup.bash
```

Example of basic robot control draw triangle

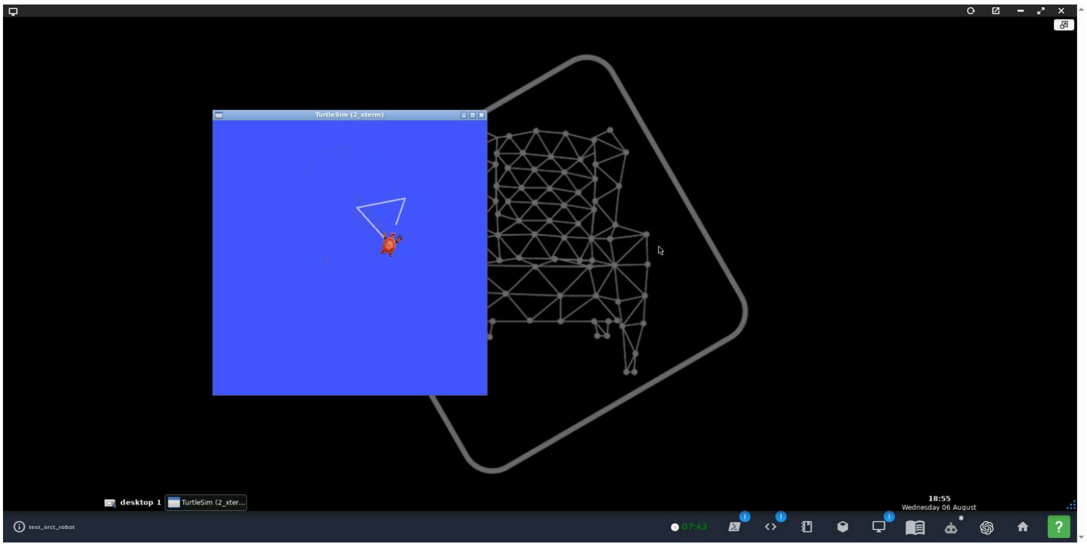
❑ New web shell and run

```
$ cd ~/ros2_ws

$ ros2 run turtle_control control_triangle
```

Example of basic robot control draw triangle

- ❑ Output


TurtleSim (2_xterm)
desktop 1
TurtleSim (2_xter...
18:55
Wednesday 06 August
test_srct_robot
07:43
?