

The Latest Advances In ROS 2

Katherine Scott
Developer Advocate
Open Robotics
[@kscottz](https://twitter.com/kscottz)

Overview

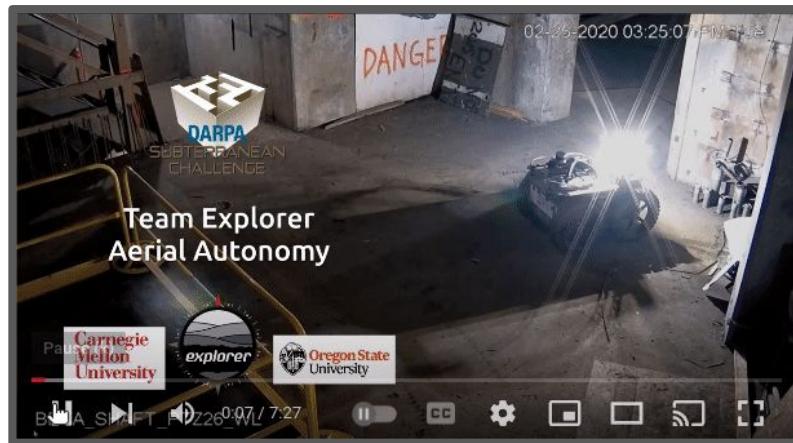
- What is ROS?
 - Why you should care
 - How it relates to PX4
 - Social Proof
 - General Features
 - Resources to Get Started
- Recent Advances
- Relevant Projects

Why should I care about this ROS stuff?

**ROS is FOSS used all over the world
to build the most advanced robots.**

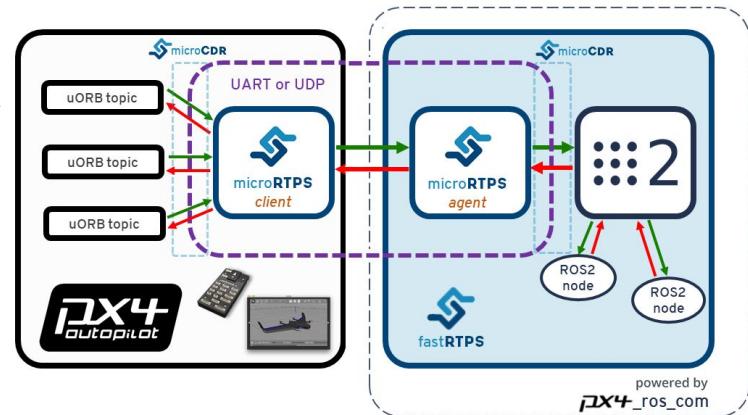
**ROS has lots of relevant features
that enable advanced drone
behaviors.**

Motivation



BUILD COOL STUFF

Motivation



2



Social Proof – Who uses ROS

ROS 2 TSC



ROS Industrial



78 Members
bit.ly/ROSIMembers

Government

- NASA
- NHSTA / USDOT
- DARPA
- Army / Navy / AF
- NIST
- Dozens of Universities
- Singapore Hospital System

Want to work at one of these places? They all use ROS

How big is ROS?

	Year Ending July 2020	Year ending July 2021	YoY Change
Total Packages	402,106	600,660	49.38%
Distinct Packages	16,083	19,884	23.63%
Deb Downloads	395,187,333	595,524,493	50.69%
Data Served (TB)	231	156	-32.26%

Direct downloads from packages.ros.org - August 2020 - July 2021

Why ROS 2?

Tooling

- Message Passing
- Sync / Async Commands
- Simulation
- Visualization
- Logging
- Deployment
- Package Mgmt
- ***Don't reinvent the wheel***

Track Record

- Decade of development
- Many dev resources
- Used in every robotics vertical
- No vendor lock-in
- Open Source

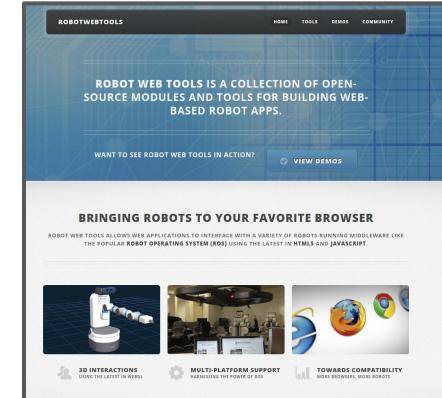
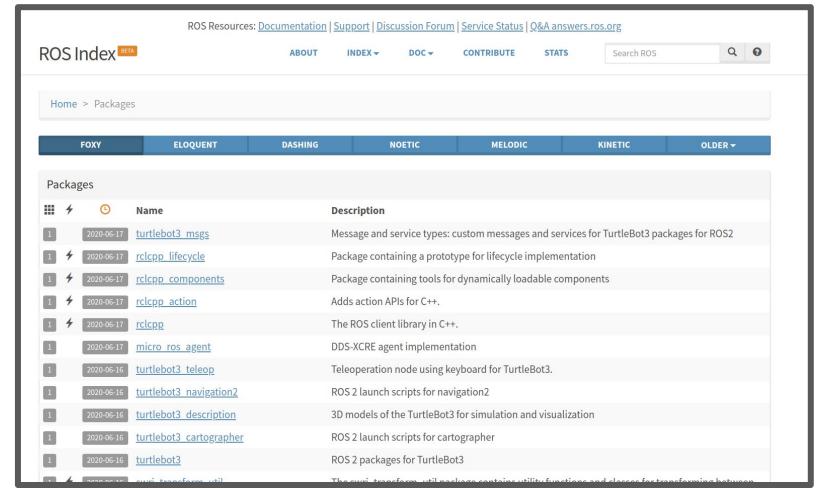
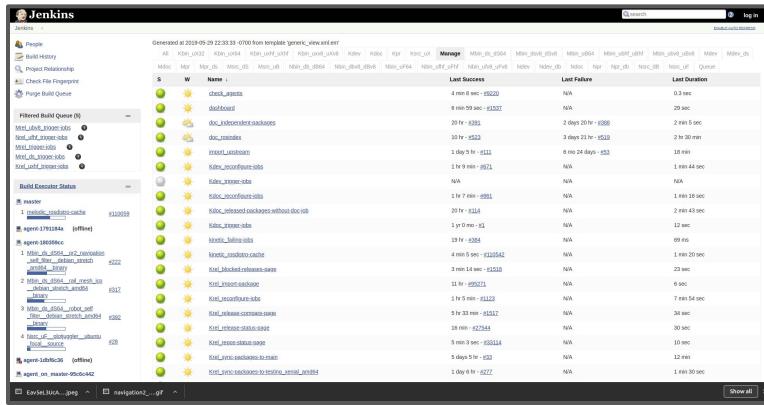
Time to Market

- ROS is Python for robots
- Focus on your core capabilities
- Hardware agnostic
- Make it work, then make it fast

We keep it real! We're not selling you a product. We're just a community of engineers

ROS Packages: Batteries Included

- ROS has a concept of packages.
 - Collections of code, that can be combined
 - C++, Python, and many others!
 - ROS federation means we need a multi-repo, multilingual meta build tool: Colcon
 - Analogous to PIP / Python Wheels



Don't Reinvent: SLAM / Navigation

Navigation 2

Docs » ROS 2 Navigation

Edit

ROS 2 Navigation



Navigation 2 project overview

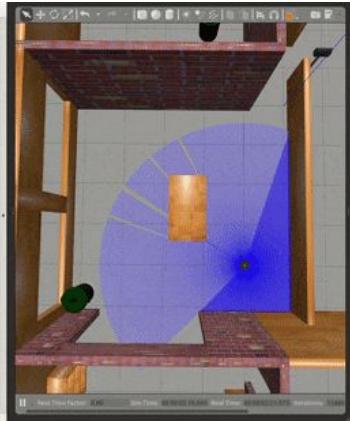
The Navigation 2 project is the spiritual successor of the ROS Navigation Stack. This project seeks to find a safe way to have a move from point A to point B. This will complete dynamic path planning, compute velocities for motors, avoid obstacles, and recovery behaviors. To learn more about this project see [About and Contact](#).

Navigation 2 uses behavior trees to call modular servers to complete an action. An action can be to compute a path, control e recovery, or any other navigation related action. These are each separate nodes that communicate with the behavior tree (BT) action server. The diagram below will give you a good first-look at the structure of Navigation 2. Note: It is possible to have multiple controllers, planners, and recoveries in each of their servers with matching BT plugins. This can be used to create context behaviors. If you would like to see a comparison between this project and ROS (1) Navigation, see [ROS to ROS2 Navigation](#).

The expected inputs to Navigation2 (Nav2) are TF transformations conforming to REP-105, a map source if utilizing the Static Layer, a BT XML file, and any relevant sensor data sources. It will then provide valid velocity commands for the motors of a holonomic robot to follow. We currently support holonomic and differential-drive base types but plan to support Ackermann (car-like) robots as well in the near future.

It has tools to:

- load, serve, and store maps (Map Server)
- localize the robot on the map (AMCL)
- plan a path from A to B around obstacles (Nav2 Planner)



ROS: Getting to Plug and Play Hardware

[UniversalRobots / Universal_Robots_ROS_Driver](#)

Driver enabling ROS operation of UR robots.

Branch: master · New pull request

Code Issues Pull requests Actions Projects Wiki Security Insights

Driver enabling ROS operation of UR robots.

312 commits 1 branch 12 packages 1 releases 30 contributors

Branch: master · New pull request Create new file Upload files Find file Close or download

Latest commit authored by May 13, 2019

- ghibli: GitHub actions (#10)
- gratitiae_templates: restore release ci and issue templates
- coordinator_dagger: Use system commands for each individual topics/parameters/parameter
- ur_identifiers: Update ur_identifiers to new template structure
- ur_uridates: Use URIXX license identifiers. (#14)
- ur_uridate_merge: Merge pull request #13 from UniversalRobotsofUkraine/dependencies
- ur_uridate_fixes: replace old references that shouldn't be there (#17)
- ur_uridate_fixes: build ur_uridate_uridate_fixes
- cloning_fixes: Updated cloning-format and added cloning-fixes instructions
- cloning_fixes: Updated cloning-format and added cloning-fixes instructions
- ghibli: Header files for messages added
- ghibli: build_uridate_uridate_fixes
- HEADLINE.indd: Use badge only from pushes

README.md

Universal_Robots_ROS_Driver

Universal Robots have become a de-facto standard of high-quality, robotic manipulation for industry, as well as for scientific research and education. The Robot Operating System (ROS) has developed from a community-centered movement to a mature framework and quasi standard, providing a rich set of powerful tools for robot engineers and researchers, working in many different domains.

With the release of Universal Robots ROS Driver, we developed a ROS driver that supports the new manipulators and the newest ROS features and packages for the ROS ecosystem that have been released. The goal of this driver is to provide a stable and sustainable interface between UR robots and ROS that strongly benefits all parties.

It is the core value of Universal Robots, to empower people to achieve any goal within automation. The success criteria of this driver release is to follow this vision, by providing the ROS community with an easy to use, stable and powerful driver, that improves the community to reach their goals in research and automation without struggling with unimportant technical challenges, instability or failing hardware.



[ouser-lidar / ouster_example](#)

Ouster sample code

Code Issues Pull requests Actions Projects Wiki Security Insights

31 commits 1 branch 12 packages 3 releases 12 contributors

Branch: master · New pull request Create new file Upload files Find file Close or download

Latest commit authored by May 13, 2019

- aghijklmn: update for v1.3. See changelog for details
- aghijklmn: Update for v1.3. See changelog for details
- aghijklmn: Update for v1.3. See changelog for details
- aghijklmn: Update for v1.3. See changelog for details
- aghijklmn: Performance improvements and bug fixes
- CHANGELOG.md: Update for v1.3. See changelog for details
- LICENSE: Ouster example client and ROS node
- README.md: Update for v1.3. See changelog for details

README.md

OSI Example Code

The OSi, reading and visualizing data, and interfacing with ROS.

status:

sent for the OSi sensor

SICK sensor

for publishing point

Added ROS code is as follows:

[SICKAG / sick_scan](#)

SICKAG sick_scan

Code Issues Pull requests Actions

513 commits 4 branches 14 packages 11 contributors Apache 2.0

Branch: master · New pull request Create new file Upload files Find file Close or download

michael1309: 1.6.0

- cfg: mrtoolbox launch test and fix for scandata config
- config: pointcloudimage filter added, timestamping optimized
- doc: Update faq.md
- doxygen: Doxygen switch to SICKAG-repo
- driver/src: NAV 210-NAV245 support added code reformat
- example_bags: support for rosbag prepared
- include: NAV 210-NAV245 support added code reformat
- launch: NAV 210-NAV245 support added code reformat
- meshes: Updated meshes
- msg: Encoder support added
- test: cleanup test program
- tools: added pointcloud chopping
- util: sick_scan.xacro: fix missing forwarding of parameters min_angle ...
- ignore: added Python script to detect scanners

README.md





[Intel\(R\) RealSense / realsense-ros](#)

Intel(R) RealSense(TM) ROS Wrapper for D400 series, SR300 Camera and T265 Tracking Module <http://wiki.ros.org/RealSense>

Code Issues Pull requests Actions Projects Wiki Security Insights

793 commits 4 branches 22 packages 45 releases 53 contributors Apache 2.0

Branch: development · New pull request Create new file Upload files Find file Close or download

donsell: Update README.indd

- realSense2_camera: add CHANGELOG.rst
- realSense2_description: add CHANGELOG.rst
- glog: use dynamic_reconfigure and support ROS2 (#526)
- axis.yml: fix point cloud test to work with outsiem_100v bag file
- LICENSE: Updated LICENSE and package files to reference correct Apache-2 license
- NOTICE: use dynamic_reconfigure and support ROS2 (#538)
- README.indd: Update README.indd

README.md

ROS Wrapper for Intel® RealSense™ Devices

These are packages for using Intel RealSense cameras (D400 series SR300 camera and T265 Tracking Module) with ROS.

RealSense supported version: v2.31 (see [realSense2_camera](#) release notes)

Installation Instructions

The following instructions are written for ROS Kinetic, on Ubuntu 16.04 but apply to ROS Melodic on Ubuntu 18.04 as well, by replacing kinetic with melodic wherever is needed.

ROS 2: Hardware Ready

Bridge between
ROS 2 DDS layer
and an RTOS

Currently on a
bunch of
different RTOSSs

Recent IDE
integration make
life easier

The screenshot shows the homepage of the micro-ROS website. At the top, there's a navigation bar with links to 'micro-ROS', 'Overview', 'Concepts', 'Tutorials', and 'Blog'. A search bar is located at the top right. The main content area features a large logo for 'μROS' where the 'μ' is a stylized microcontroller chip icon. Below the logo, there's a section titled 'micro-ROS' with the subtitle 'puts ROS 2 onto microcontrollers.' To the left, there's a 'Main Ideas' section with a pencil icon and a brief description of the major changes compared to regular ROS 2. To the right, there's a 'Getting Started' section with a gear icon and a 'Get Involved' section with a person icon. Both sections contain links to various parts of the website. At the bottom, there's a call-to-action button labeled 'GET MY INVITE' and a form field asking for an email address.

micro-ROS

Overview Concepts Tutorials Blog

Search via Lunr.js

micro-ROS puts ROS 2 onto microcontrollers.

Main Ideas

The major changes compared to "regular" ROS 2 is that micro-ROS uses a Real-Time Operating System (RTOS) instead of Linux, and DDS for extremely Resource Constrained Environments (DDS-XRCE) instead of classical DDS. Above that, we run the ROS 2 stack! Well, with a few cool improvements for taking advantage of microcontroller specific things, but largely the same.

Getting Started

Get Involved

Source code can be found at <https://github.com/micro-ROS/> and we answer both tickets and pull requests as usual. If you have questions on concepts and development, you're very welcome to post to the [embedded category on ROS Discourse](#). For usage questions, head over to [ROS Answers](#) and please tag your question with [embedded](#).

Join Micro-ROS on Slack

What is your email address?

GET MY INVITE

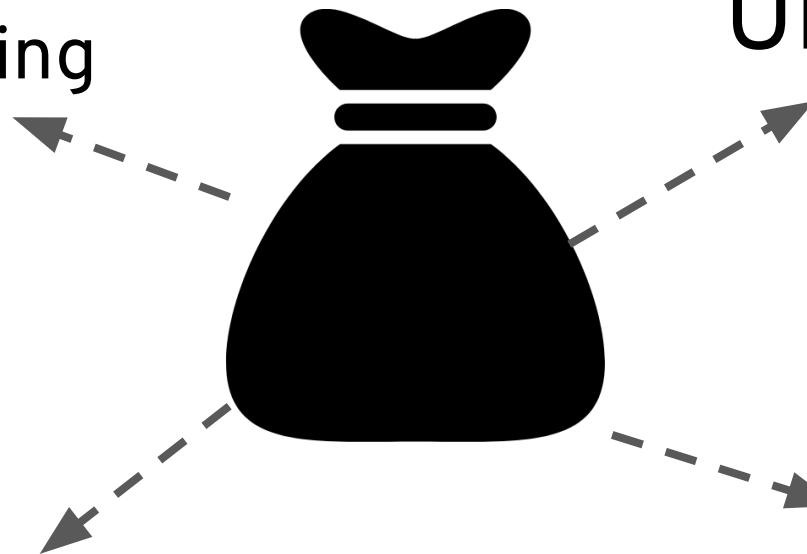
BAGS ARE AMAZING

ML/CV / DL
Training

Unit Tests!

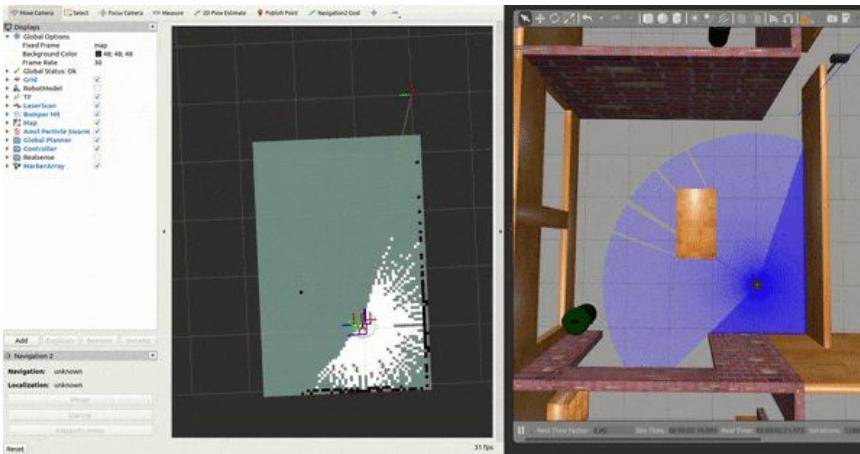
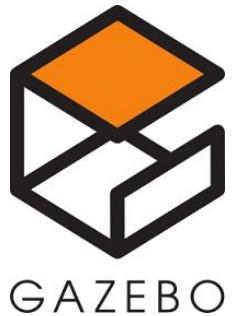
Collaborators

“Black Box”



Don't Reinvent: Simulation

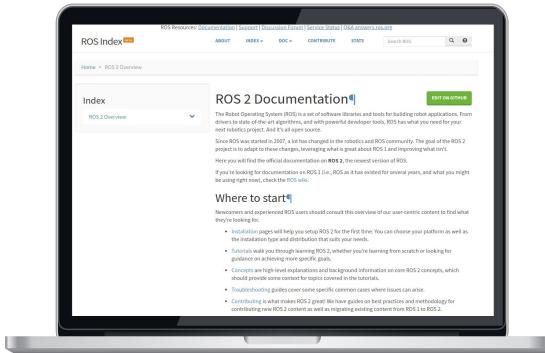
- Ignition Gazebo, build a virtual robot
- Simulate Physics, robots, and sensors
- Plays well with ROS 2
- Allows you build, test, and debug from the comfort of your desk.
- Gazebo Fortress is recommended for Humble



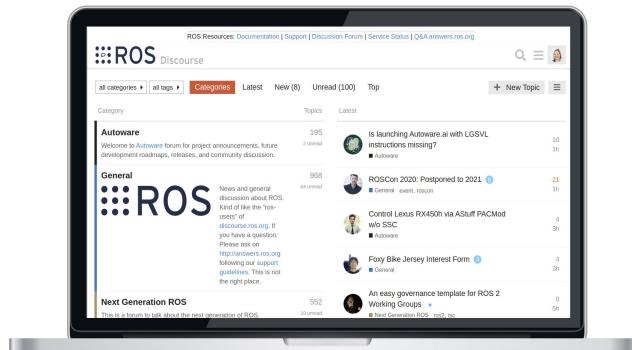
Where to get started?



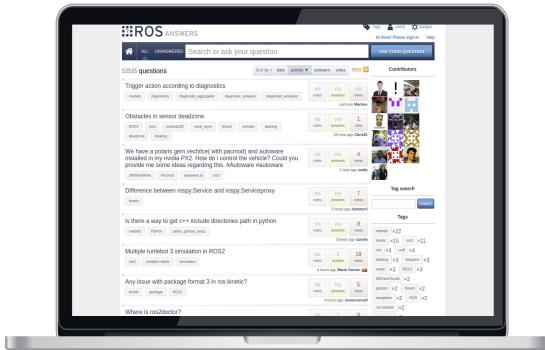
ros.org



docs.ros.org



discourse.ros.org



answers.ros.org

NEW ROS 2 Paper

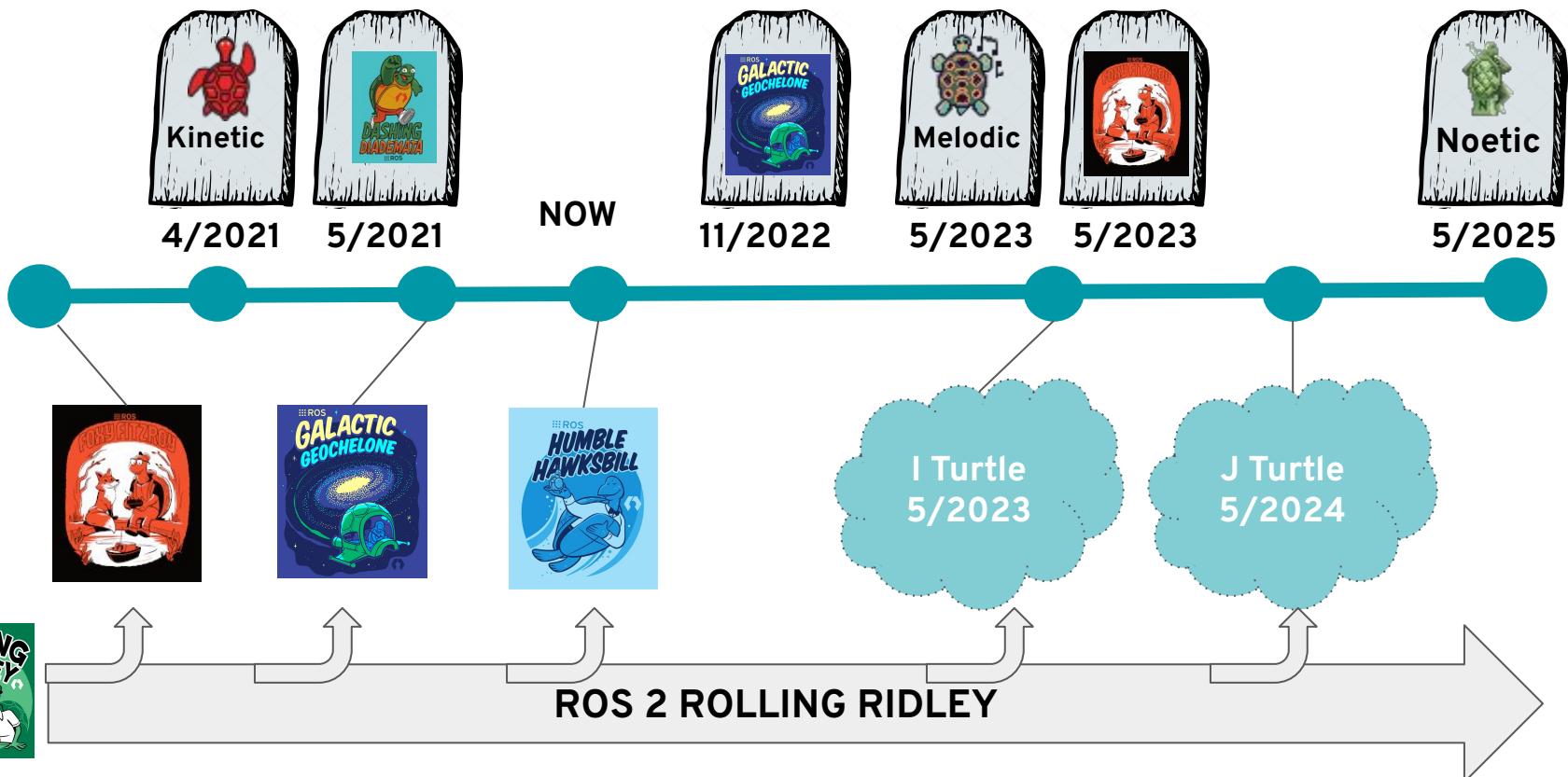
- Design considerations for ROS 2
- Five “real-world” applications
- Land, Sea, Air, Space, Industry
- Auterion’s ROS Applications
- ROS for NASA VIPER
- PLEASE USE THIS FOR CITING ROS 2



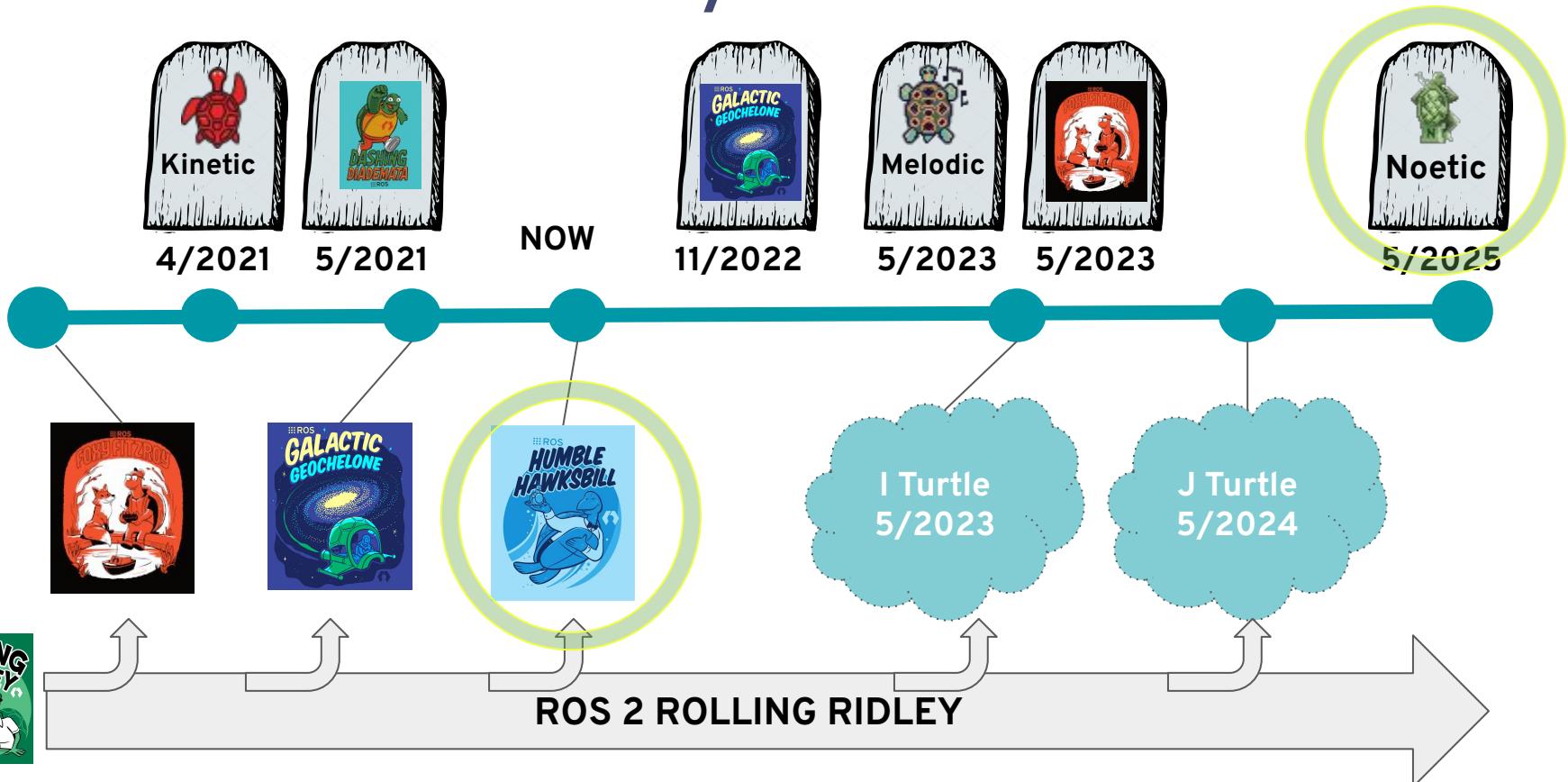
bit.ly/ROS2Science



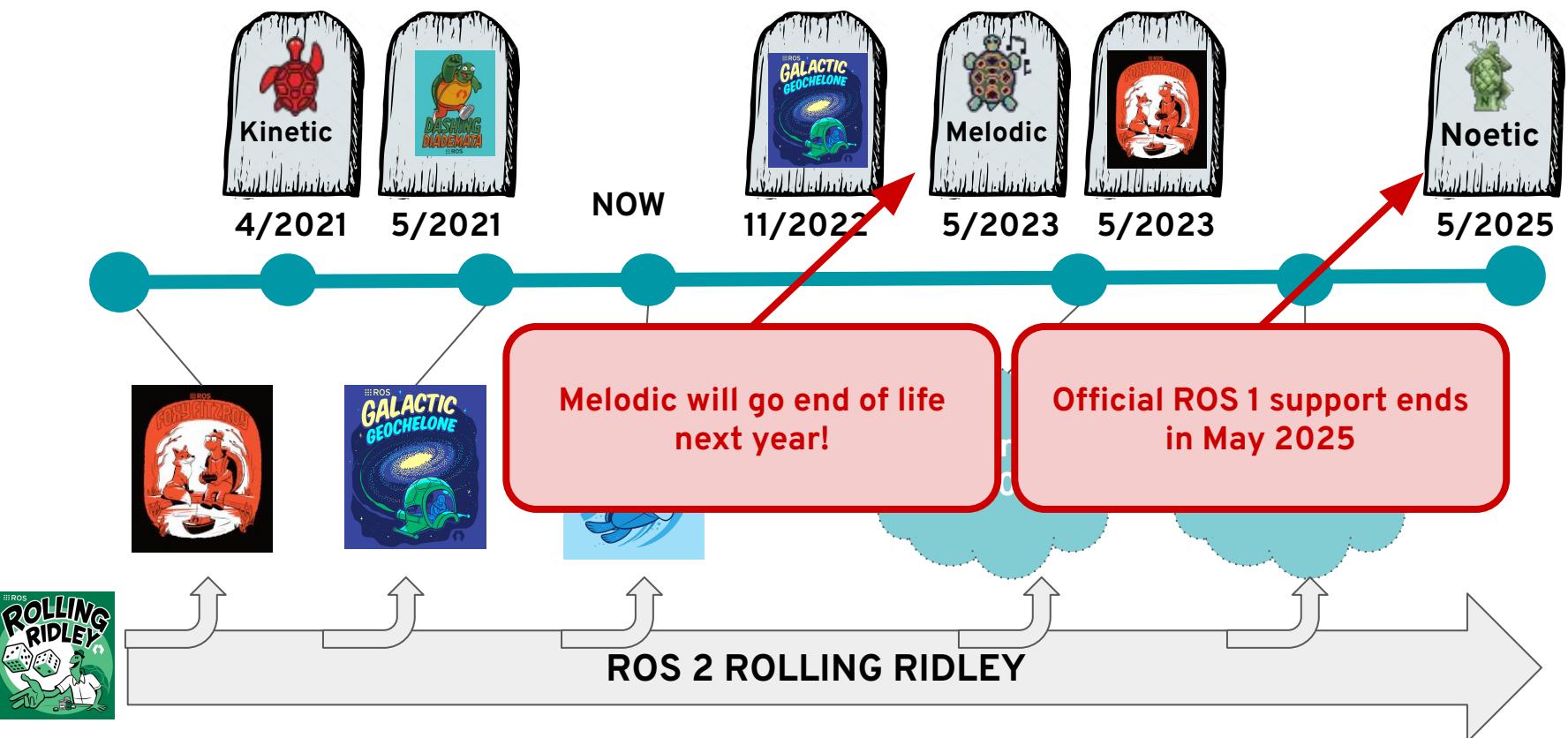
REP-2000: Distro Lifecycle



REP-2000: Distro Lifecycle



REP-2000: Distro Lifecycle





YOU SHOULD BE THINKING ABOUT MOVING TO ROS 2

ROS 2 Humble Hawksbill

- Long term support until 2027
- Binaries and docker containers available
- Lots of new docs and great features with TONS of features from the community
- If you are getting started, start here



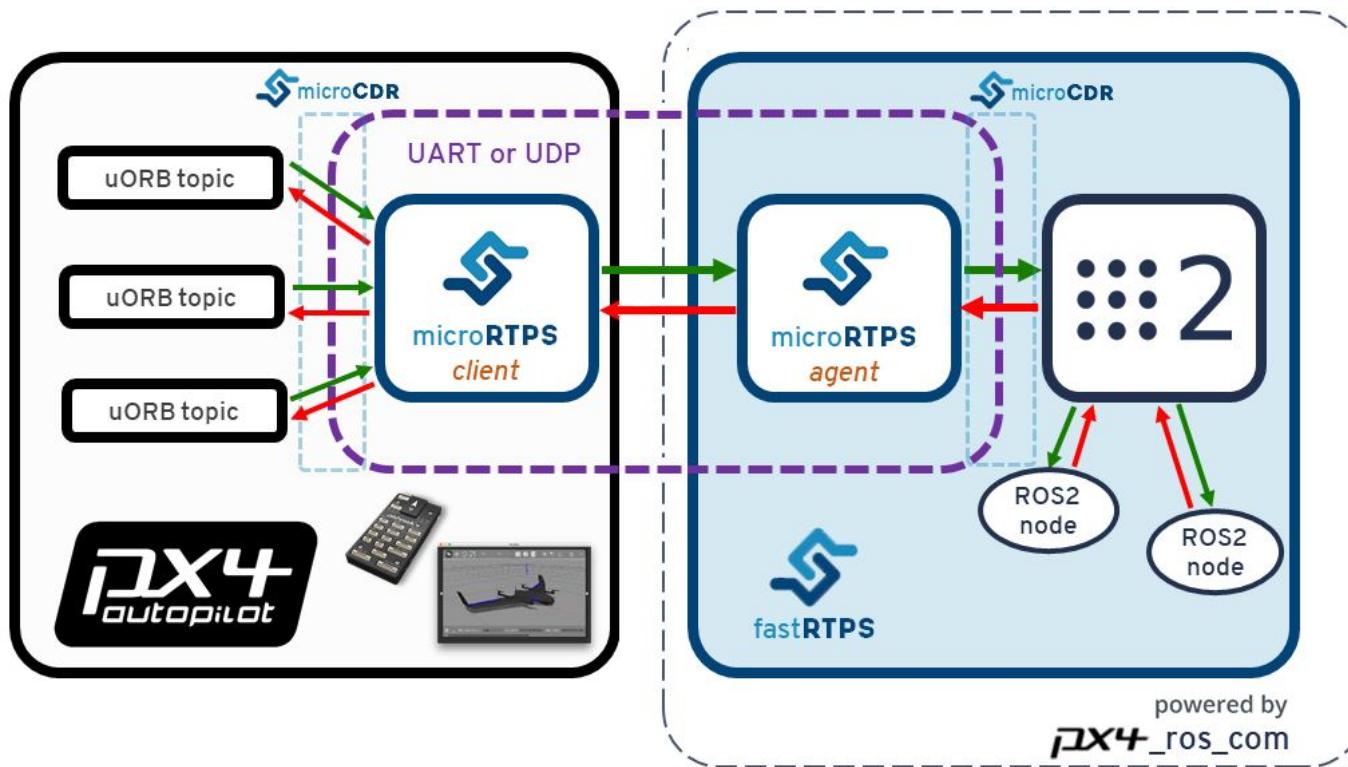
bit.ly/ROS2Humble

MAVROS Now Available in ROS 2 Humble

- Humble was released a month ago and MavROS packages are already available
- ROS 2 only works with MavROS 2.0
- Fairly robust set of tutorials and API documentation

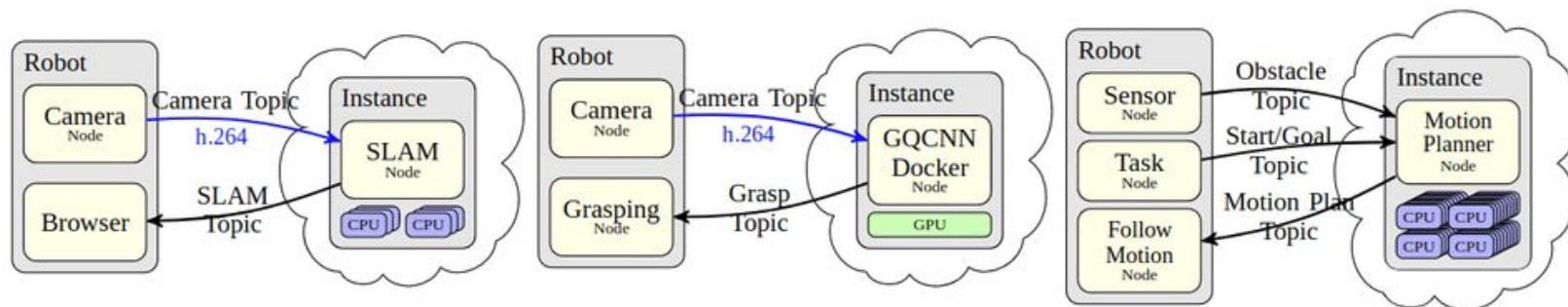
The screenshot shows the ROS Index website interface. At the top, there's a navigation bar with links to Documentation, Support, Discussion Forum, Service Status, and Q&A answers. Below the navigation is a search bar and a dropdown menu for ROS distributions: HUMBLE (selected), GALACTIC, FOXY, ROLLING, NOETIC, MELODIC, and OLDER. The main content area displays the 'mavros' package from the 'mavros' repository. It includes a 3D model icon, a brief description ('mavros package from mavros repo'), and tags: libmavcon, mavros, mavros_extras, and mavros_msgs. Below this is a section titled 'GITHUB-MAVLINK-MAVROS'. The page is divided into several tabs: Overview (selected), Assets (0), Dependencies (36), Tutorials (0), and Q & A (36). The 'Package Summary' tab shows details like Tags (jav, mav, mavlink, plugin, apm, px4), Version (2.1.1), License (GPLv3), Build type (AMENT_CMAKE), and Use (RECOMMENDED). The 'Repository Summary' tab provides information about the GitHub repository, including Checkout URI (<https://github.com/mavlink/mavros.git>), VCS Type (git), VCS Version (fo2), Last Updated (2022-06-07), and Dev Status (DEVELOPED). The 'Package Description' tab contains a brief description of MAVROS as a MAVLink extendable communication node for ROS with proxy for Ground Control Station. The 'Additional Links' tab lists Website, Repository, and Bugtracker. The 'Maintainers' tab lists Vladimir Ermakov. The 'Authors' tab also lists Vladimir Ermakov. On the right side of the page, there are links for API Docs, Browse Code, and Wiki.

Connecting PX4 to ROS 2 Humble



bit.ly/PX4ROS2

FogROS 2 (UC Berkeley / Ken Goldberg)



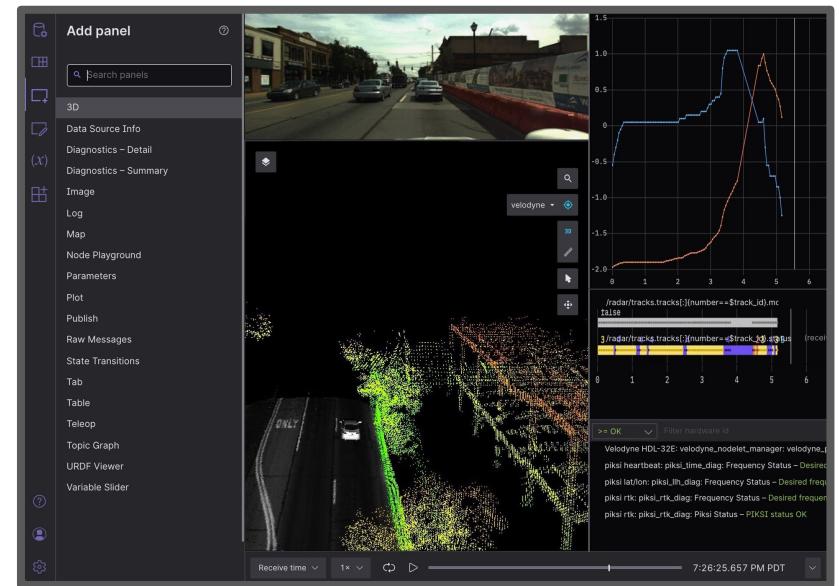
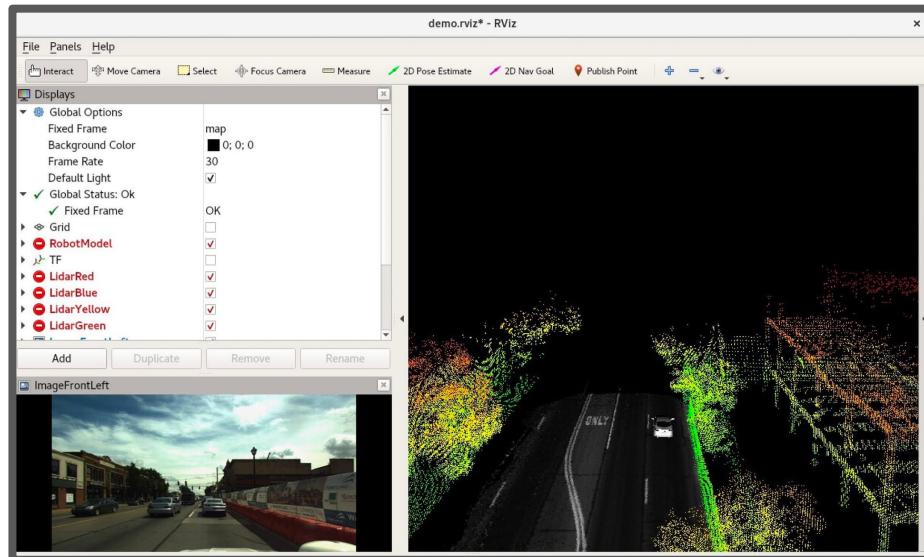
- Solves problem of off loading compute / storage to the “cloud”
- Useful when you have 5G / Good Wifi
- Solves: 3D reconstruction, data logging, deep learning, heavy planning tasks
- Currently AWS but promises for other clouds



bit.ly/FogROS2

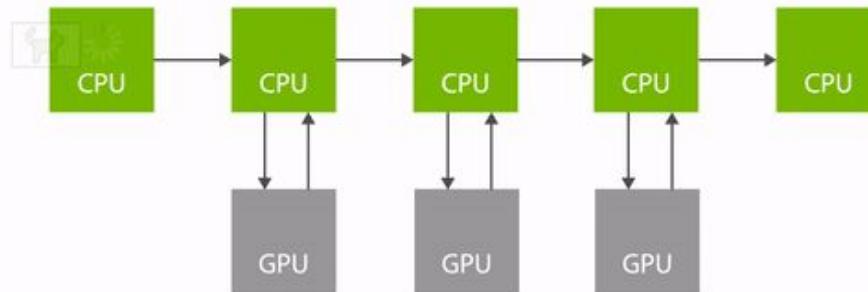
User Friendly

Community and companies are filling in the gaps

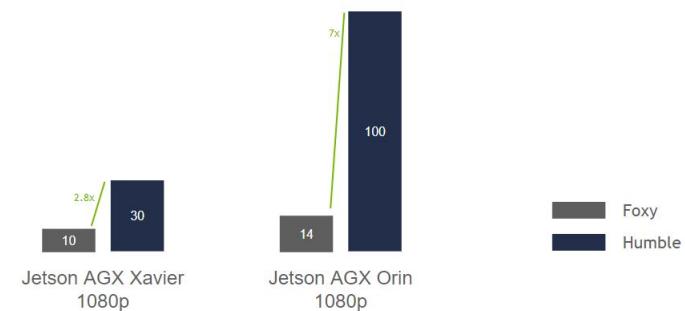
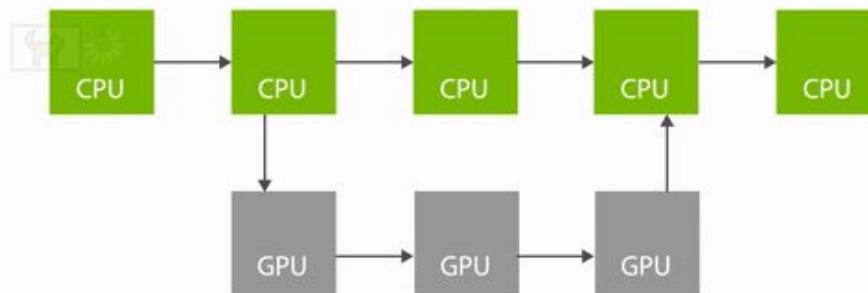


NVIDIA Hardware Support => FAST PERCEPTION

Inefficient Hardware Acceleration



Efficient Hardware Acceleration



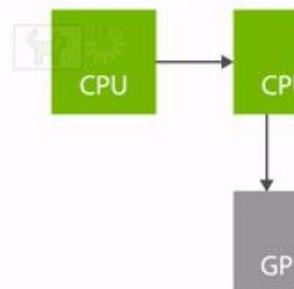
type adapt framework performance comparison



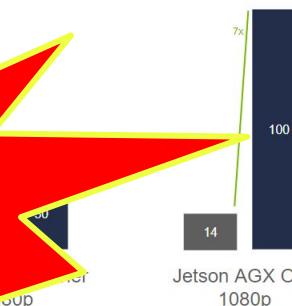
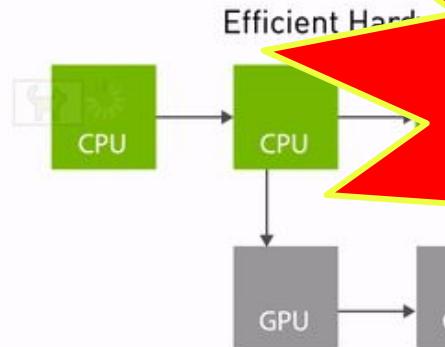
bit.ly/NVIDIAROS2

NVIDIA Hardware Support => FAST PERCEPTION

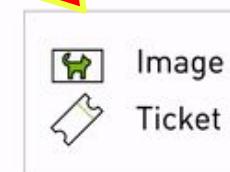
Inefficient Hardware Acceleration



**OTHER VENDORS ARE
WORKING ON SIMILAR
ENHANCEMENTS!**



type adapt framework performance comparison



bit.ly/NVIDIAROS2

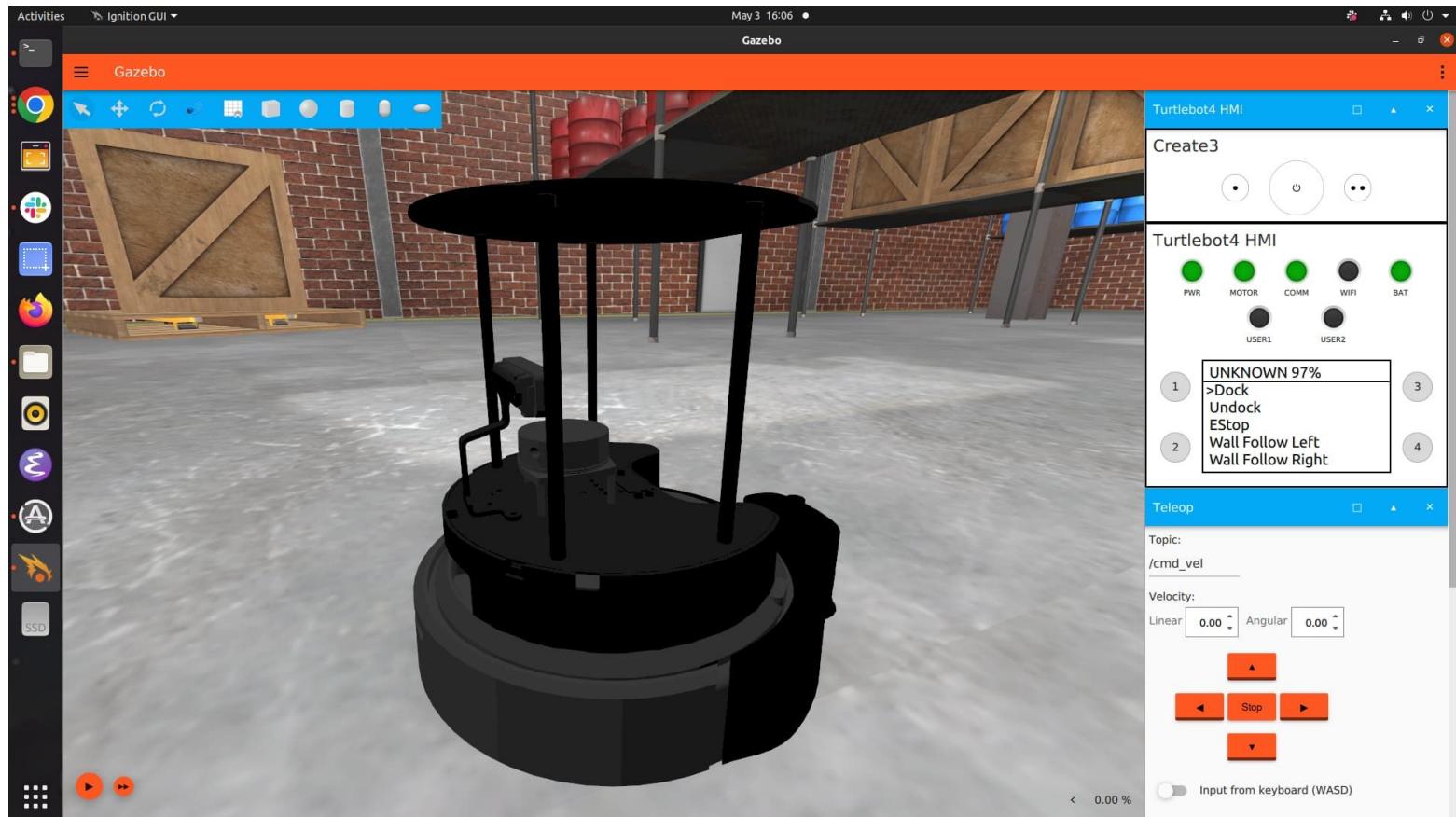
TurtleBot 4 Launch



TurtleBot 4		TurtleBot 4 Lite
SIZE AND WEIGHT		
EXTERNAL DIMENSIONS (LxWxH)	341 x 339 x 351 mm [13.4 x 13.3 x 13.8 in]	341 x 339 x 192 mm [13.4 x 13.3 x 7.5 in]
WEIGHT	3.9 kg (8.6 lbs)	3.3 kg (7.2 lbs)
WHEELS (Diameter)	72 mm (0.55 in)	
GROUND CLEARANCE	4.5 mm (0.17 in)	
SPEED AND PERFORMANCE		
MAX PAYLOAD	9 Kg - Default 15 kg - Custom Configuration	
MAX SPEED	0.31 m/s (safe mode), 0.46 m/s (cliff sensors disabled)	
MAX ROTATIONAL SPEED	1.90 rad/s	
BATTERY AND POWER SYSTEM		
CHEMISTRY	26 Wh Lithium Ion (14.4V nominal) Rechargeable	
CHARGE TIME	2.5 hrs	
OPERATING TIME	2.5 - 4.0 hrs (load dependent)	
USER POWER	VBAT @ 300mA, 12V @ TBD mA, 5V @ 500mA 3.3V @ 250mA	VBAT @ 1.9A (14.4V nominal) Low current 5V and 3.3V via Raspberry Pi GPIO



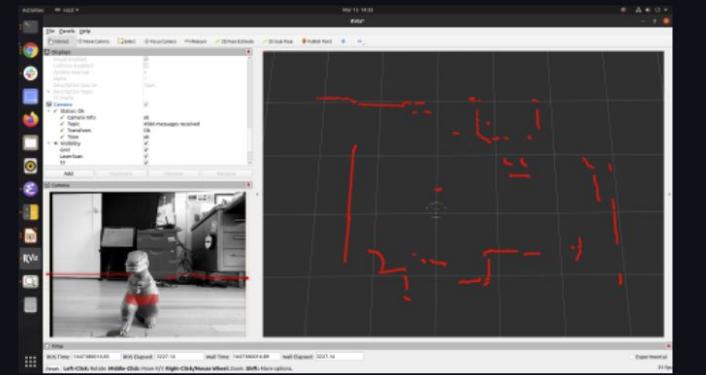
bit.ly/AboutTB4



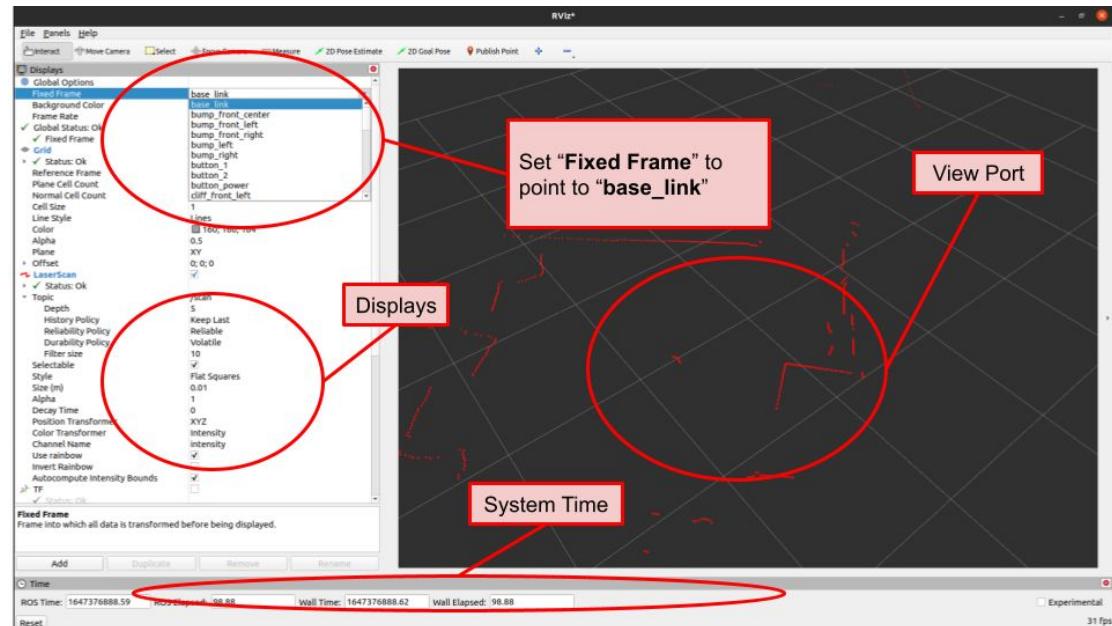
bit.ly/TB4Sim

Lessons – coming soon

Setup Camera and Laser

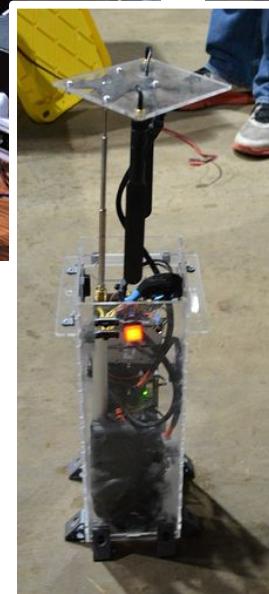
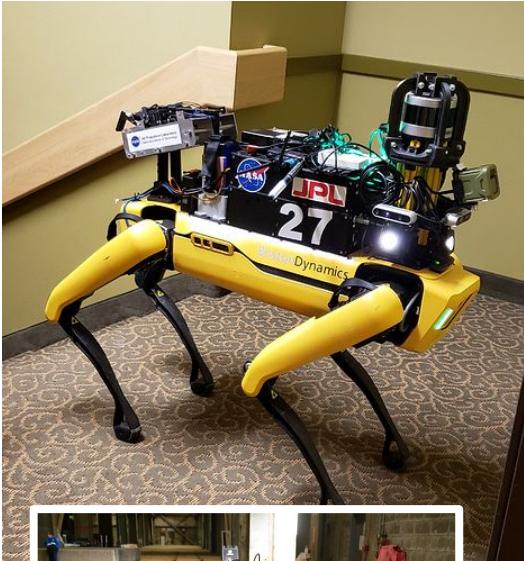


- Next we'll setup the TB4 camera and lidar.
- To view the LIDAR click **Add** in the display pane.
 - In the dialog that pops up select **LaserScan**.
 - Under **LaserScan** in the Displays pane click the checkbox to enable the display.
 - Set the **Topic** to **/scan**.
 - The laser scan should now be visible.
- To view the camera click the **Add** button under displays.
 - In the dialog that pops up select **Camera**.
 - Now enable the camera by clicking the checkbox in the Displays pane.
 - Set **Topic** to either **/right/image_rect** or **/left/image_rect**.

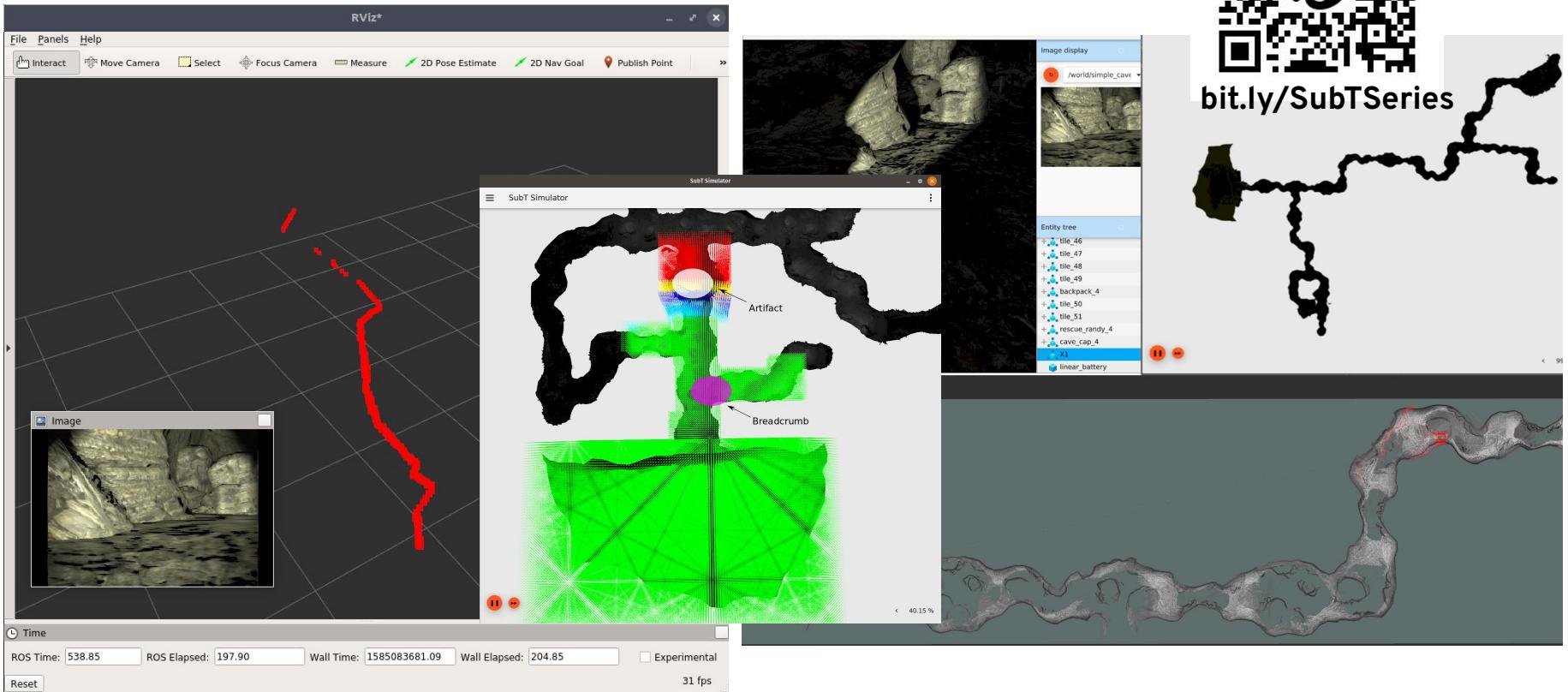


bit.ly/TB4Lesson

Recent Advances: DARPA SubT Challenge



Give DARPA SubT A Try



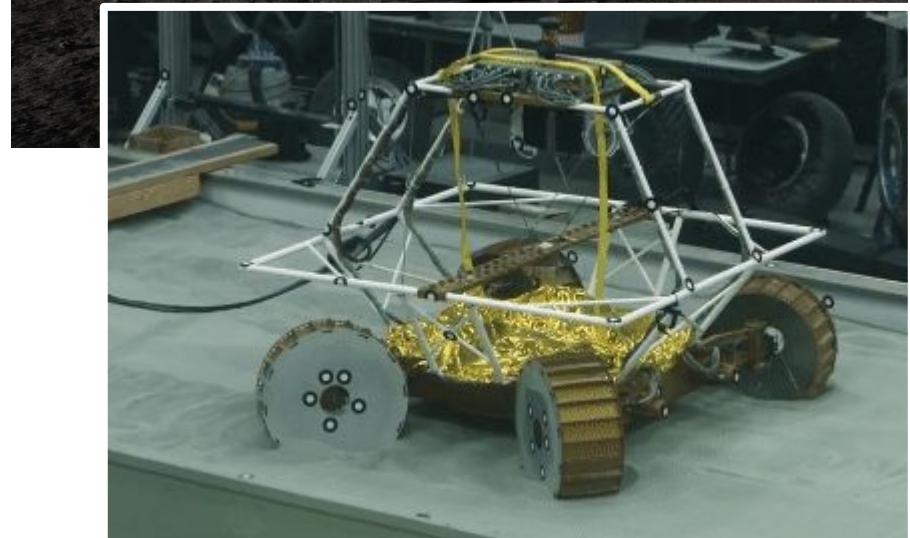
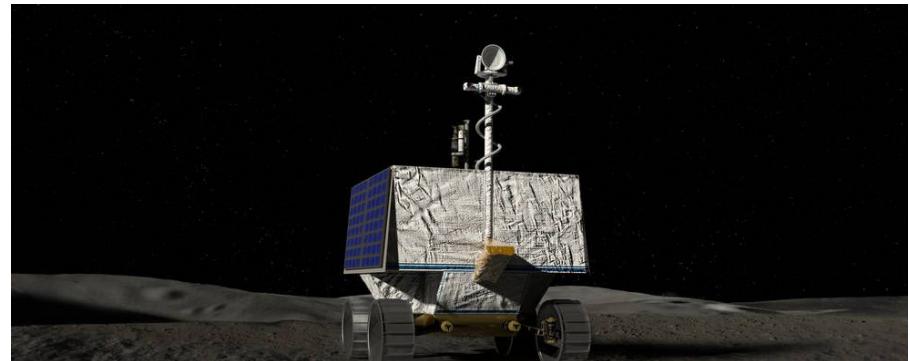
Lookin Ahead: Space

Volatiles Investigating Polar Exploration Rover, or VIPER

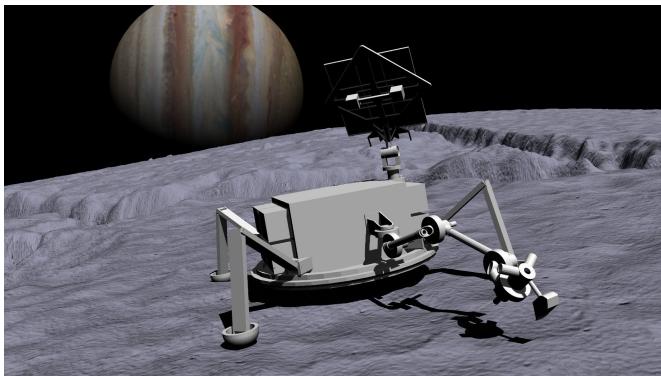
Moon mission late 2023

“With all the incoming data from the rover, the MOC team makes use of a web-based data visualization platform developed by NASA called Open Mission Control Technologies. The Open MCT software is open source, making it freely available not only for space missions like VIPER, but other applications in the public and private sectors.”

*“Collaborations with industry are also feeding into key aspects of VIPER’s software operating system. Rather than creating only custom code, the rover’s flight and ground-based software will make extensive use of open source software, including key components adapted from the **Robot Operating System 2 (ROS 2)** which is widely used in everything from robot vacuum cleaners to autonomous drones and self-driving cars. “*



Looking Ahead: Space ROS



Europa Lander Simulation

"The NASA Ames Intelligent Systems Division's Ocean Worlds Autonomy Testbed for Exploration Research and Simulation (OceanWATERS) software team announced their 9th public release of the OceanWATERS open source software on December 13, 2021"

bit.ly/OceanWATERS



Astrobee

"Three NASA Astrobee free-flying robots have been operating inside the International Space Station (ISS) since 2019. T Astrobee's flight software performs vision-based localization, provides autonomous navigation, docking and perching, manages various sensors and actuators, and supports human-robot interaction with co-located astronauts via screen-based displays, light signaling, and sound."

bit.ly/astrobee

The screenshot shows a NASA contract opportunity page for the "Space Robot Operating System". The page includes the NASA logo, a sidebar with links like "Contract Opportunity", "General Information", "Classification", "Description", "Attachments/Links", "Contact Information", "History", and "Award Notices". The main content area shows the following details:

ACTIVE	Contract Opportunity
Notice ID: RFI-GSFC-NExIS-2021	Related Notice
Department/Ind. Agency: NATIONAL AERONAUTICS AND SPACE ADMINISTRATION	Department/Ind. Agency
Sub-Tier: NATIONAL AERONAUTICS AND SPACE ADMINISTRATION	Sub-Tier
Office: NASA GODDARD SPACE FLIGHT CENTER	Office

General Information

Contract Opportunity Type: Sources Sought (Original)
All Dates/Times are: (UTC-05:00) EASTERN STANDARD TIME, NEW YORK, USA
Original Published Date: Nov 18, 2021 04:02 pm EST
Original Response Date: Jan 31, 2022 05:00 pm EST
Inactive Policy: Manual
Original Inactive Date: Mar 31, 2022
Initiative:

- o None

Space ROS

NASA, Blue Origin and industry partners are collaborating to develop the Space Robot Operating System (Space ROS) which is an open-source software framework the space robotics community can use to develop flight-quality robotic and autonomous space systems.

bit.ly/SpaceROS

New and Noteworthy!



f1tenth.org



New and Noteworthy: Indy Autonomous Challenge



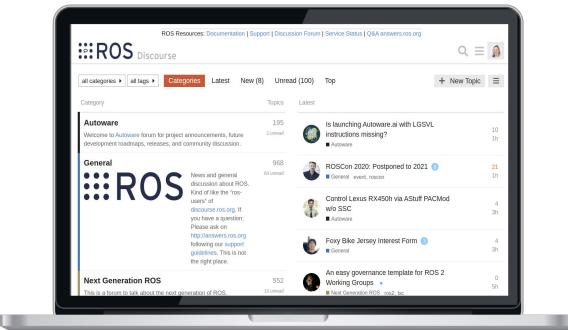
Stepping stone to



IndyAutonomousChallenge.com



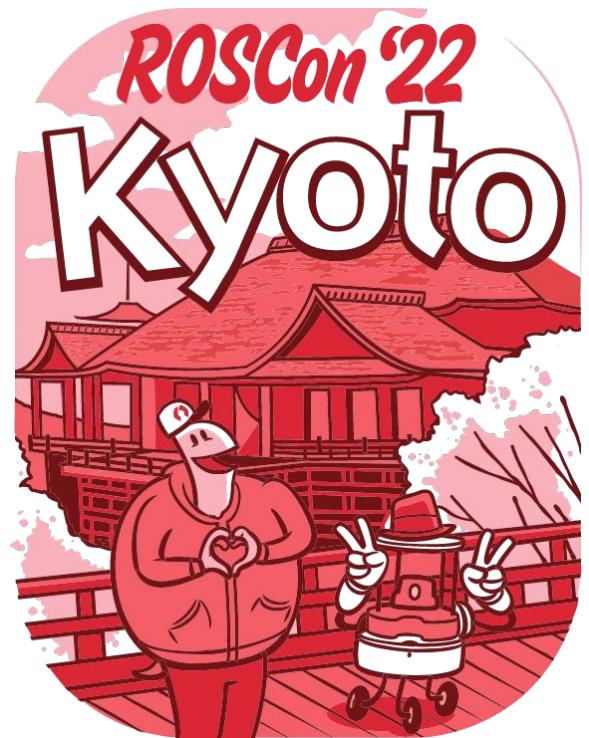
Shameless Self Promotion



discourse.ros.org



answers.ros.org



THANKS!
@openroboticsorg
@rosorg
@kscottz
ros.org