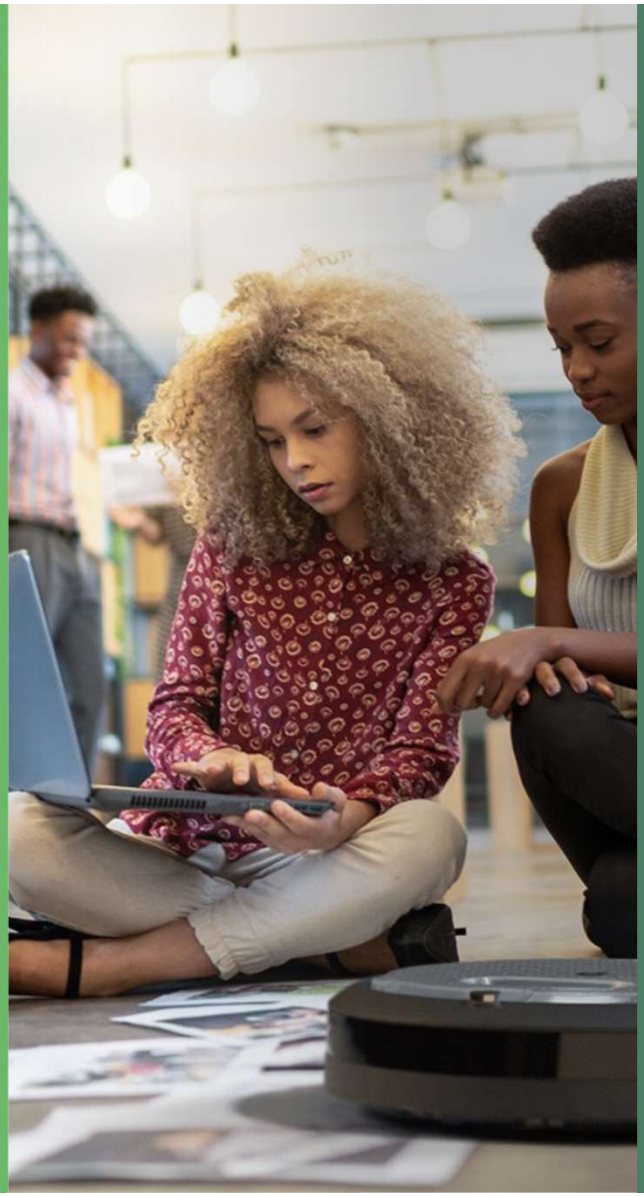
OCTOBER 15, 2023

Building the Create® 3 Robot:

Challenges and Solutions for ROS 2 Consumer Robotics

Presented by:
Alberto Soragna and Steven Shamlian



1. ***About:*** The History Behind iRobot's Create® Robots
2. ***Architecture:*** Create® 3 Hardware & Software
3. ***Challenges & Solutions:*** Create® 3 x ROS 2
4. ***Usage:*** Create® 3 Internal and Community Use

“ Innovators dream of creating useful robots, but they often get bogged down with designing a mobile platform that works. iRobot® Create® fills a need in the robot industry for a standard, durable hardware platform on which to rapidly develop new, innovative mobile robots. ”

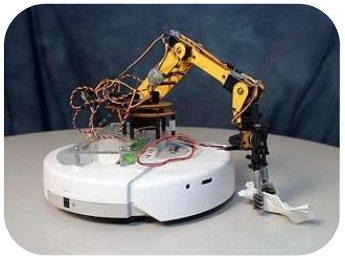
- HELEN GREINER, CO-FOUNDER AND CHAIRMAN OF IROBOT

JANUARY 8, 2007

About | The Original Create® Robot 2007

Roomba® Discovery Series

Create® 1 Series



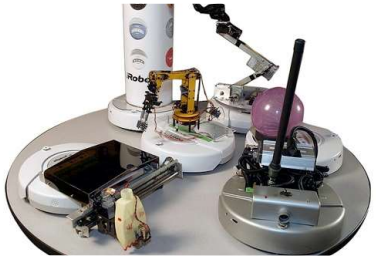


Image of TurtleBot from REP-0119 by Willow Garage, in the public domain.
TurtleBot is a trademark of the Open Source Robotics Foundation.

About | The Create® 2 Robot 2014

Roomba® 600 Series

Create® 2 Series

The photo of TurtleBot Euclid is used by permission from Clearpath Robotics.
TurtleBot is a trademark of the Open Source Robotics Foundation.

About | A New Horizon 2019-2021

Elementary | Middle | High School | College | Workforce

Lite

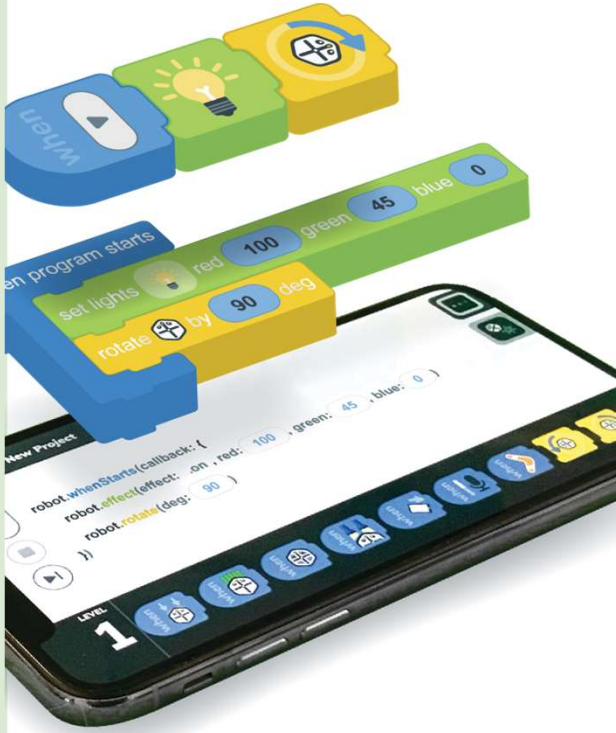
Pro

Designed and piloted with educators to reduce common barriers to STEM education, iRobot's Root® coding robots advance alongside learners to deconstruct abstract STEM subjects into uniquely engaging and interactive experiences.

By promoting visual, auditory, and kinesthetic learning, Root® coding robots bring classroom learning to life and help teachers form cross-curricular connections.

Python 3 is governed by the Python Software Foundation.

Compatible with the iRobot Coding App and the iRobot Python Web Playground. Lessons available in our Learning Library.



About | The Create® 3 Robot 2022

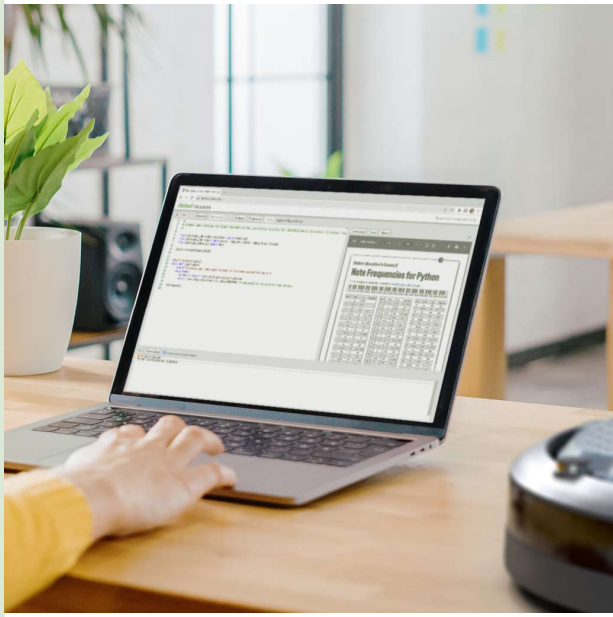
Elementary | Middle | High School | College | Workforce

Built from the blueprint of the Roomba® i3 robot, the Create® 3 educational robot provides an out-of-the-box buildable mobile robot development platform.

Designed for advanced learners, the Create® 3 robot offers a suite of smart sensors and actuators for building robotics applications using Python and ROS 2—from multi-robot exploration to navigation and mapping technology, and more.

ROS 2 is governed by Open Robotics and Python 3 is governed by the Python Software Foundation.

Compatible with the iRobot Python Web Playground. Lessons available in our Learning Library.

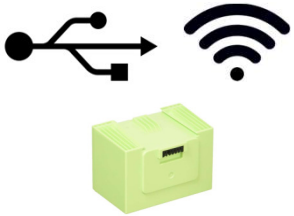


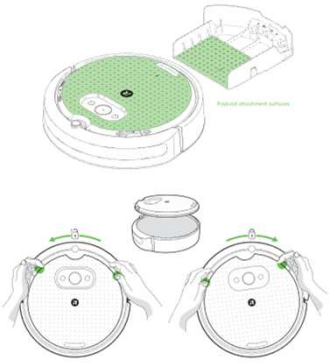
ROS 2™

About | The Create® 3 Robot 2022

1. Leverage well-tested platform for sensing, mobility, computation, connectivity, and power management
2. Implement software architecture using standard software platform
3. Easy access to wired power and wired/wireless data interfaces
4. Add easy (and easily serviced) physical interfaces for mounting sensors, computers, actuators
5. Onboard inexperienced learners with the iRobot® (née-Root®) Coding open protocol / native apps / Python® web playground

ROS 2™





HARDWARE + SOFTWARE

ARCHITECTURE

Architecture | Create® 3 Mechanical Hardware

Roomba® i3 Series

Create® 3 Series

As easy as replacing some parts…

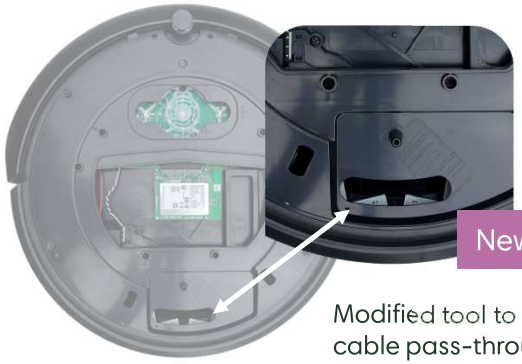
right?

Architecture | Create® 3 Mechanical Hardware

Roomba® i3 Series

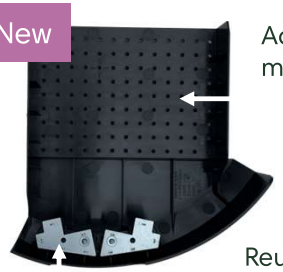
Create® 3 Series

No changes



New

Modified tool to add cable pass-through

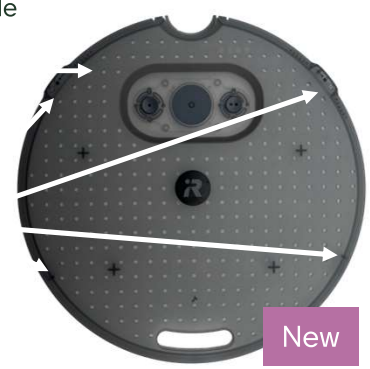


New

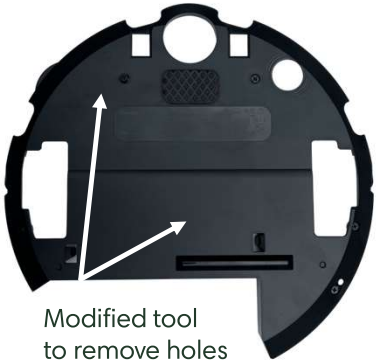
Added standard hole mounting pattern

Reused screw geometry to add 5° faceplate removal

Reused ballast blocks



New

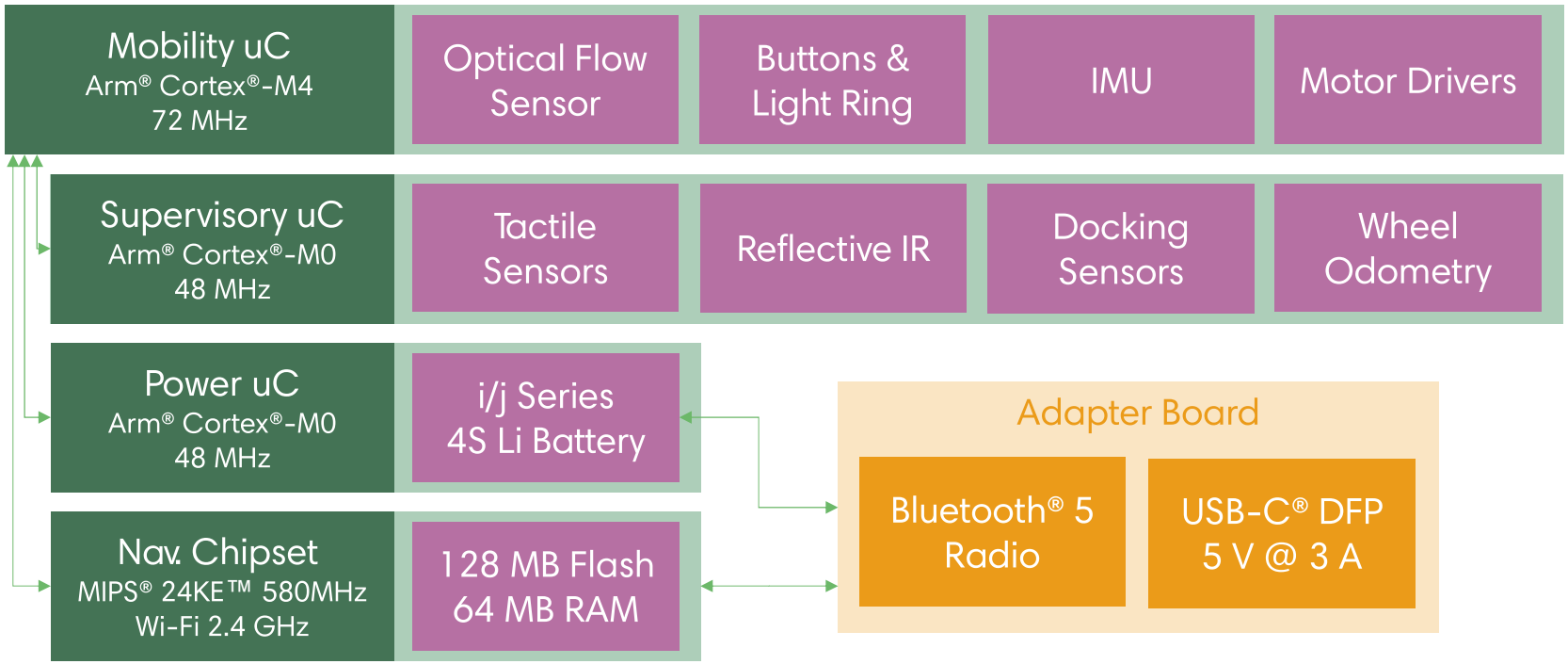


Modified tool to remove holes

New

Customized adapter board

Architecture | Create® 3 Electrical Hardware



Architecture | create-platform Navigation App

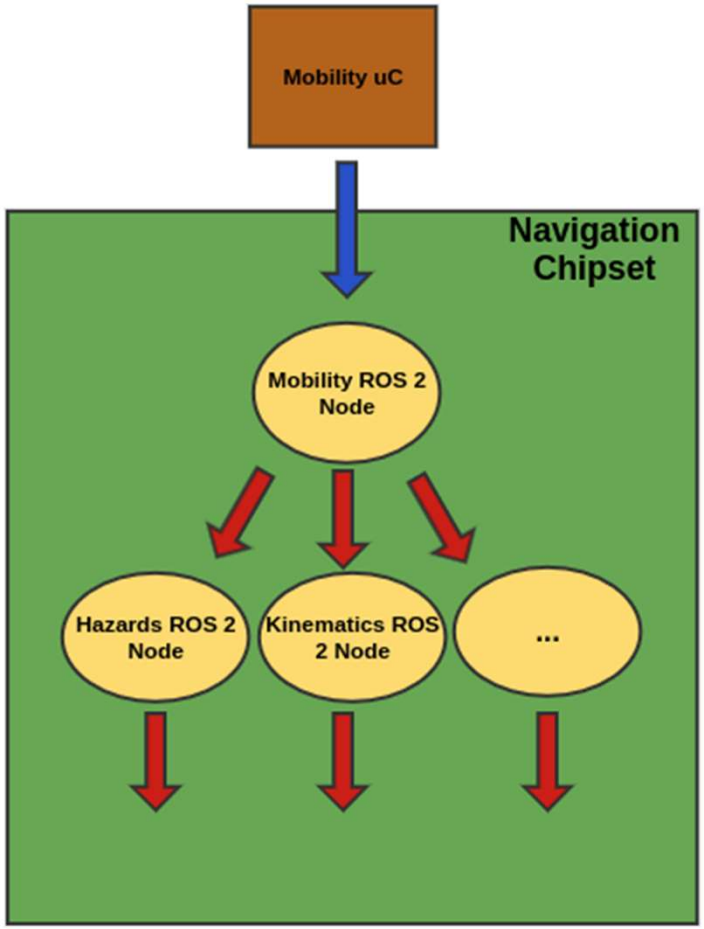
The create-platform is the application that implements the iRobot® Create® 3 robotics logic.

- Single-process ROS 2 application.
- Each component (12) is implemented as a ROS 2 lifecycle node.
- Multiple nodes share the same ROS 2 executors (4).
- Controlled startup sequence: lifecycle nodes are created, configured and activated.
- Executors spin until the app is instructed to terminate.

Architecture | Create® 3 Sensing Pipeline

Expose both raw sensor data as well as sensor-agnostic information to allow users to build upon our platform.

- Publish raw sensor data received from the mobility micro-controller as ROS 2 messages.
- Run sensor-fusion algorithms for inertial estimation: synchronize raw messages to compute dead-reckoning, slip and stop.
- Hazards detection from raw sensor signals and hazards representation using tf2 transformations.

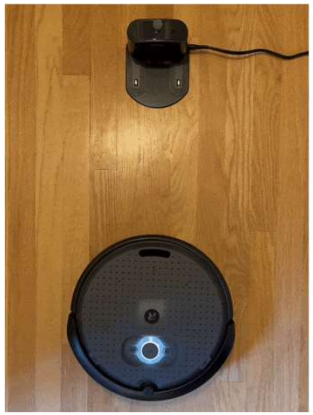


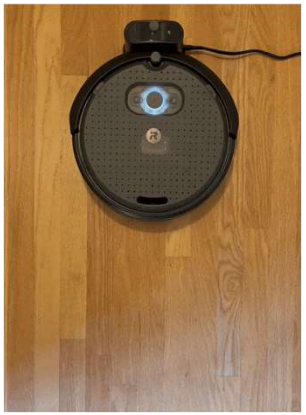
Architecture | Create® 3 Autonomous Behaviors

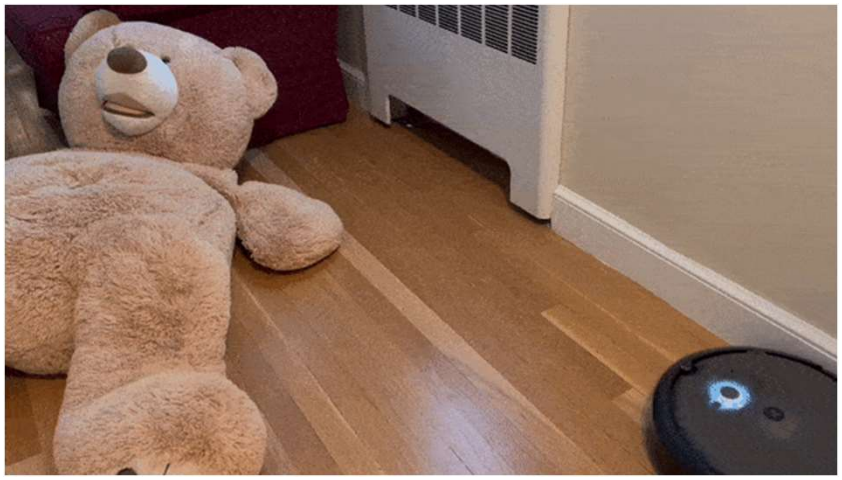
Expose ROS 2 actions to trigger autonomous behaviors. These are some of the same behaviors used also on Roomba robots!

- Proximity dock-search and IR-based docking.
- Undocking.
- Obstacles perimeter following.

All the raw data used internally by the behaviors is available through ROS topics. Users can reproduce (and improve!!) the existing behaviors.







Architecture | Create® 3 Motion Control Loop

Periodic task to manage behaviors and produce wheel velocities.

Multi-threaded control loop using ROS 2:

- Behavior plugins thread: execute ROS 2 action server callbacks to activate or terminate a behavior.
- ROS 2 timer thread: iterate over the active behavior goal (if any) and publish wheel velocities at 50Hz.
- A ROS 2 subscription to cartesian velocities on `/cmd_vel` topic allows users and third-party applications to take control of the robot.
- PID controllers in the mobility board apply a smooth acceleration profile to actuate the wheels at 200Hz.

Architecture | Create® 3 Stable ROS 2 Interfaces

Create® 3 shares some ROS 2 nodes with the Roomba robots. These nodes are not meant for public usage and do not have stable interfaces.

- Heavy usage of non-standard message types.
- Focused on performance rather than user experience.
- No stability or backward-compatibility guarantees.

We leverage ROS 2 hidden namespaces and a dedicated ROS 2 node that converts proprietary data-types into the stable Create® 3 ROS 2 interfaces[1].

[1] https://github.com/iRobotEducation/irobot_create_msgs

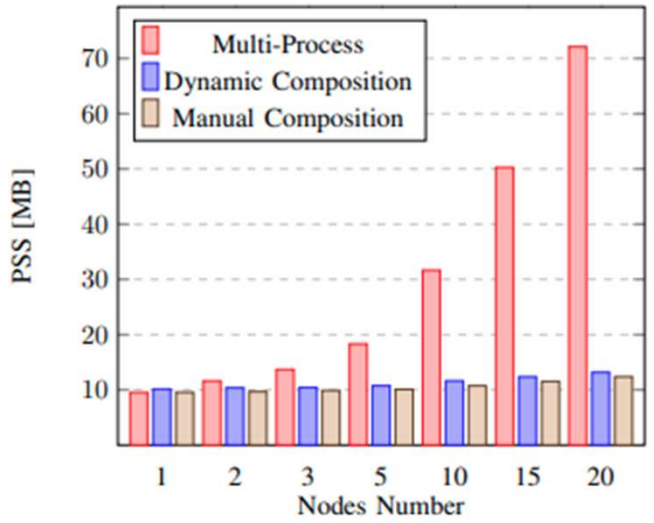
CHALLENGES + SOLUTIONS

ROS 2

Challenges + Solutions | RAM Usage

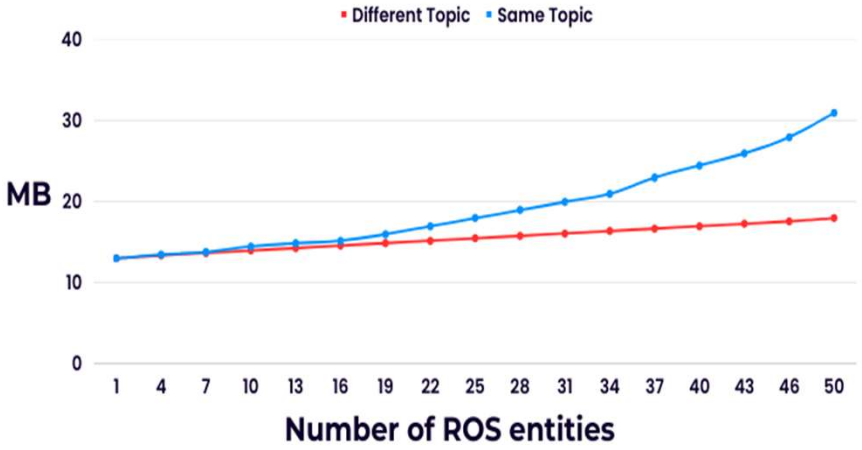
Reducing the number of ROS 2 processes and nodes is the most effective way to minimize RAM usage.

- Take advantage of ROS 2 composition!



The topology of the ROS 2 graph heavily influences the memory impact.

- Keep your graph simple! Use a pipelined approach rather than a fully-connected topology.
- Consider the impact of ROS 2 parameters.



S. Macenski, A. Soragna, M. Carroll and Z. Ge, "Impact of ROS 2 Node Composition in Robotic Systems," in *IEEE Robotics and Automation Letters*, vol. 8, no. 7, pp. 3996-4003, July 2023, doi: 10.1109/LRA.2023.3279614.

Challenges + Solutions | RAM Usage

A large part of the RAM usage can be attributed to the RMW layer.

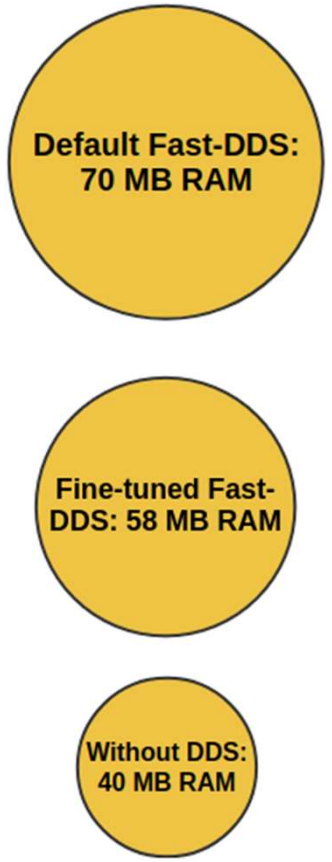
A single-process ROS 2 applications made of ~10 nodes, ~60 topics and ~30 services:

- 70 MB RAM usage with default Fast-DDS 2.6.4
- 40 MB RAM usage with iRobot® StubDDS[1]

DDS-based RMWs are highly customizable through XML profiles.

Fine-tuning Fast-DDS[2] we can reduce RAM usage from 70 MB to 58 MB:

- Disable unused built-in communication transports.
- Reduce maximum message size and maximum socket size.
- Do not pre-allocate message samples.
- Reuse the same locators for user-traffic and meta-traffic.

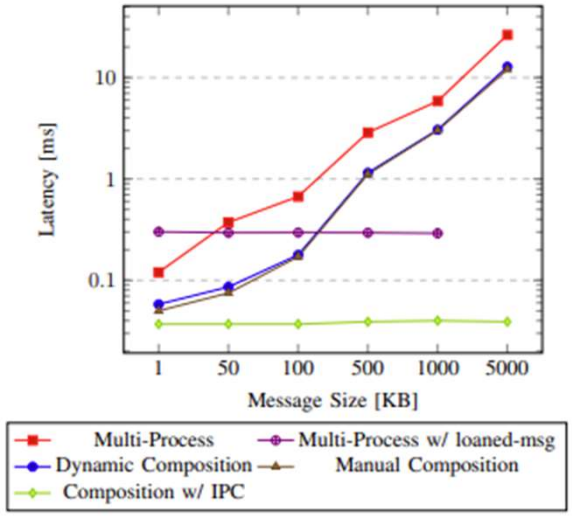


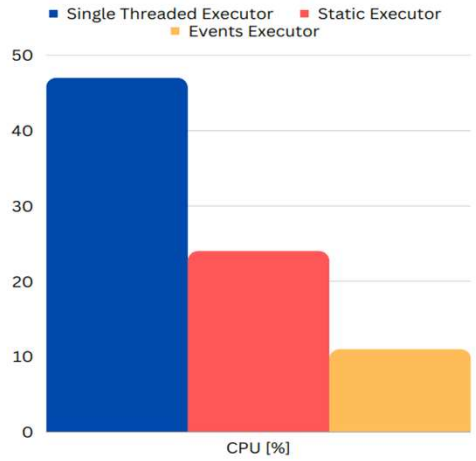
[1] StubDDS is not a real DDS: https://github.com/mauropasse/rmw_stubdds
[2] https://github.com/irobot-ros/roscon-2023-keynote/blob/main/fast-dds-profile.xml

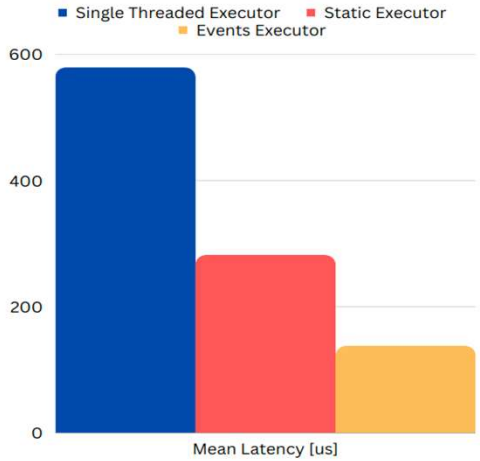
Challenges + Solutions | Latency and CPU Usage

Optimizing the "fast-path" is key to improve runtime performance. The ROS 2 abstraction layers have a non-negligible impact on performance.

- (Again) Take advantage of ROS 2 composition: this allows to enable intra-process communication.
- Use the Events-Executor.







S. Macenski, A. Soragna, M. Carroll and Z. Ge, "Impact of ROS 2 Node Composition in Robotic Systems," in *IEEE Robotics and Automation Letters*, vol. 8, no. 7, pp. 3996-4003, July 2023, doi: 10.1109/LRA.2023.3279614.

Challenges + Solutions | Improving Remote Communication

The CPU impact of ROS 2 remote communication is still an open problem.

- Create® 3 CPU usage is 55%-60% when no remote subscribers are present.
- Subscribing remotely to all its topics (~100KBps) raises the robot's CPU usage to ~85%.
- Live debugging (`rviz` or `rosbag`) perturbs the system in a non-acceptable way!
- Discovery traffic shouldn't be neglected.

Small high-frequency topics are problematic for all current Tier-1 RMWs:

- On Raspberry Pi 4, an application with ROS 2 Humble uses 12% CPU to produce 1 MBps across 10 topics.
- This increases to 45% CPU to produce 1 MBps across 100 topics.

Challenges + Solutions | Building & Packaging ROS 2 Apps

Most users install ROS 2 from Debian packages.
Then, they can `colcon build` their proprietary application on top of it.

This workflow is often not possible in consumer robotics:

- Custom Linux-based OS (e.g., built on Yocto).
- Use of proprietary platforms without officially released ROS 2 packages.
- Not enough resources to compile source-code on the platform.

Using the official Debian packages also adds an implicit dependency between your OS distribution and version (e.g. Ubuntu 22.04) and your ROS 2 distribution (e.g. Humble).

Challenges + Solutions | Our Cross-Compilation Workflow

Cross-Compilation consists of compiling source-code on a platform different from the one where it will be executed.

- Use the Conan package manager to define how to build all your dependencies from sources.[1]
- Cross-Compile dependencies from x86_64 to your target platform.[2]
- Upload the compiled artifacts into a self-hosted server.
- Use Conan to pull pre-compiled artifacts and make them available to standard build tools (such as CMake).

This gives us full control over the ROS 2 libraries. We maintain an Open-Source fork of ROS 2 where we cherry-pick relevant PRs that couldn't be back-ported due to API/ABI compatibility issues.[3]

[1] https://github.com/irobot-ros/conan-ros2
[2] https://docs.conan.io/1/systems_cross_building/cross_building.html
[3] https://github.com/irobot-ros/rclcpp/tree/galactic-events-executor

Challenges + Solutions | Our Packaging Workflow

Installation and packaging are not frequently discussed in the ROS community.

On platforms with limited flash-space (128MB) it's required to install only what you need.

x86_64 applications can use tools such as `ldd` to inspect the ELF file and find their *NEEDED* runtime dependencies.
The Open-Source project `cross-tool-ng` contains a script to emulate `ldd` with cross-compilation toolchains.[1]

This isn't sufficient with ROS 2!

ROS 2 dependencies can be dynamically loaded at runtime (e.g., RMW and plugins) so they do not appear in the ELF file and can't be known a-priori by the linker.
We developed a CMake script to find and install them by using prior information.[2]

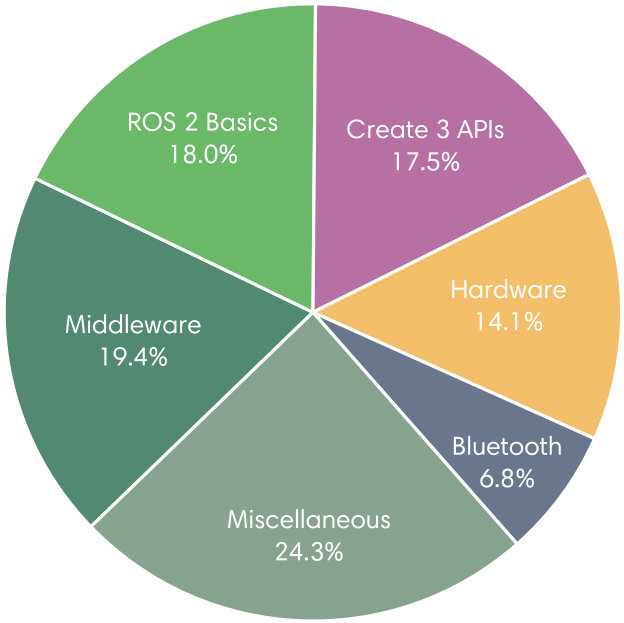
[1] https://github.com/crosstool-ng/crosstool-ng/blob/master/scripts/xldd.in
[2] https://github.com/irobot-ros/roscon-2023-keynote/blob/main/install-ros-script.cmake

Challenges + Solutions | Create® 3 Forum: The Data

Users come to our GitHub forum to ask questions about their robots and projects.

The RMW is a critical topic: users have a hard-time configuring and debugging it.

Many users also struggle with ROS 2 basic concepts such as how to build packages from sources.



[1] https://github.com/iRobotEducation/create3_docs/discussions

Challenges + Solutions | Create® 3 Forum: FAQs

"Why can't I communicate with my robot??"

This is the single mostly asked question on the forum (more than 10%).
A complex problem for both beginner and advanced users, without clear debugging information.

The troubleshooting process:

1. Are your robot and laptop on the same network?
2. Can you ping the robot IP address from the laptop?
3. Are the robot and the laptop using the same version of ROS 2?
4. Are the robot and the laptop using the same RMW implementation?
5. Are you using a particular network (e.g. corporate or school)?
6. Can you try switching RMW implementation?
7. Can you try unicast communication via XML profile?
8. The last resort: run network packet analyzers such as Wireshark or tcpdump

[1] https://github.com/iRobotEducation/create3_docs/discussions
Wireshark is a trademark of WIRESHARK FOUNDATION, INC.

Challenges + Solutions | ROS 2 Documentation

The amount of documentation for ROS 2 is large, but not sufficient to cover all the needs of its diverse userbase.

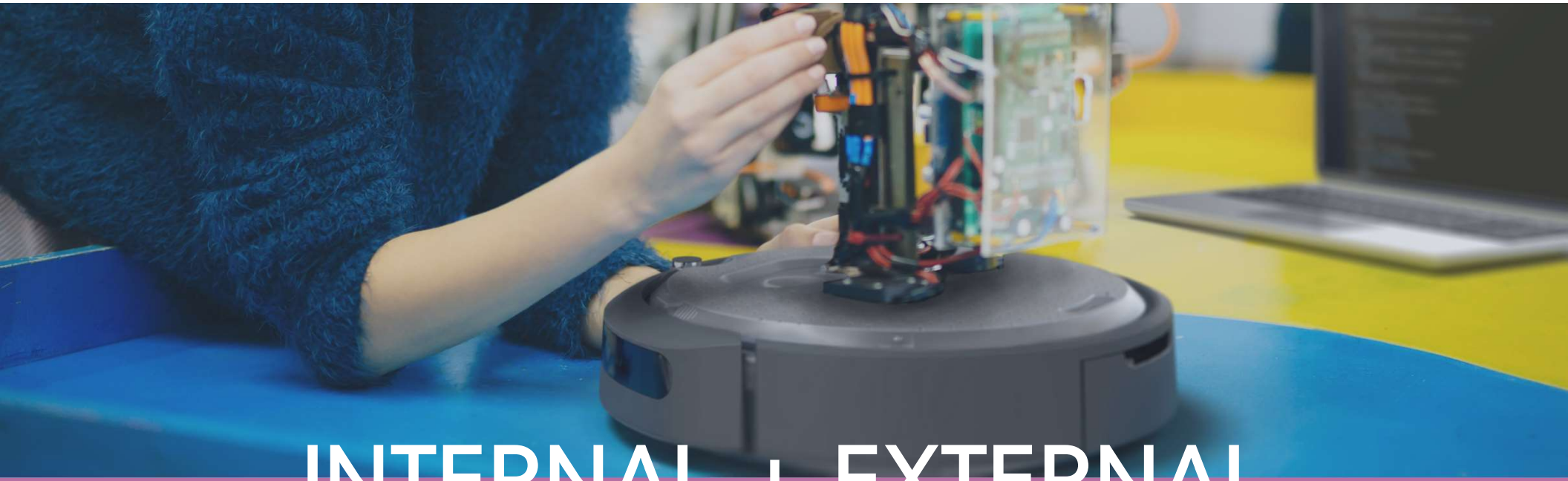
Beginners struggle to follow the official ROS 2 basic tutorials.

- Assumption of a non-negligible Linux expertise.
- We wrote tutorials to setup ROS 2 aiming for very beginners.[1]

Advanced users struggle to apply concepts to their applications.

- Tools, GUIs and ROS 2 nodes are well-documented to be used out-of-the-box, but there's a lack of API documentation for the underlying C++ or Python libraries.
- Very often they need to search their answers in the source-code.

[1] https://edu.irobot.com/units/getting-started-with-ros-2



INTERNAL + EXTERNAL USAGE

Usage | Create® 3: Internal Use

- ***Hackathons***
  - T.R.E.A.T. – An experimental prototype engineered to dispense dog treats on demand
  - Robot soccer competitions
  - Explorations with mobility and manipulation
- ***Product Testing***
  - Sensors
  - Prototyping algorithms
- ***Professional Development***
  - Learning ROS 2

Tasty Reward Emission Apparatus Thingy (T.R.E.A.T.)[1]

[1] https://www.youtube.com/watch?v=GBR-NpoLxwE

Usage | Create® 3: External Use

Job Re-Training

- It's never too late!

School/University Teaching

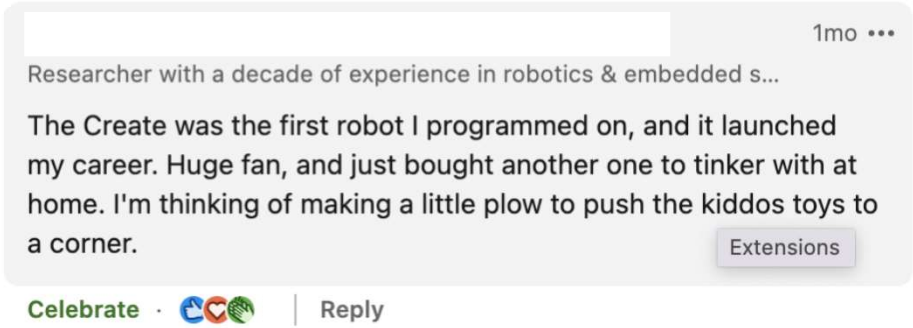
- Be here tomorrow at 11 AM!

Derivative Products & Prototypes

- Clearpath Robotics: TurtleBot 4
- Trossen Robotics: LoCoBot

Individual Projects/Experiments

- Create 3 Discussion Forum



TurtleBot is a trademark of the Open Source Robotics Foundation.
The photo of TurtleBot 4 models used with permission from Clearpath Robotics.
The photo of LoCoBot used with permission from Trossen Robotics.

Thank You! | What's Next?

Get Involved!

- *Ask questions when things are unclear*
- *Share your progress and projects*
- *Submit pull requests to improve Projects, Tutorials, & Simulator at https://github.com/iRobotEducation/*
- *Work with other learners and educators to improve ROS 2 training material with the student in mind*

Visit http://edu.irobot.com/ROSCON2023