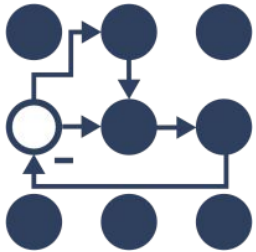
Zenbedded: Bringing ros2_control Closer to the Embedded World



Christoph Froehlich
Eneji Peacemaker Ohieku, Shahazad Abdullah, Kamalkant Thangaraju,
Krzysztof Pochwała, Bence Magyar

$ whoami

Christoph Fröhlich

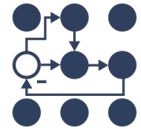
- PhD in control engineering
- Research Engineer at
  AIT Austrian Institute of Technology
- Large-Scale Robotics Lab
- Motion planning, deliberation and system integration
- ros2_control maintainer and PMC member



christophfroehlich

froehlichchristoph

Zenbedded





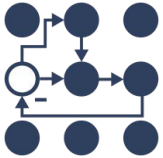
Zephyr

+



Zenoh

+



ros2_control



open source robotics alliance

osralliance.org



Goals of the OSRA

Provide governance for Open Robotics projects

Ensure the long-term sustainability of Open Robotics projects, both financially and technically

Be vendor-neutral and ensure the projects operate and grow for the good of all involved in robotics

ROS: Robot Operating System

https://ros.org/

ROS = Plumbing + Tools + Capabilities + Ecosystem

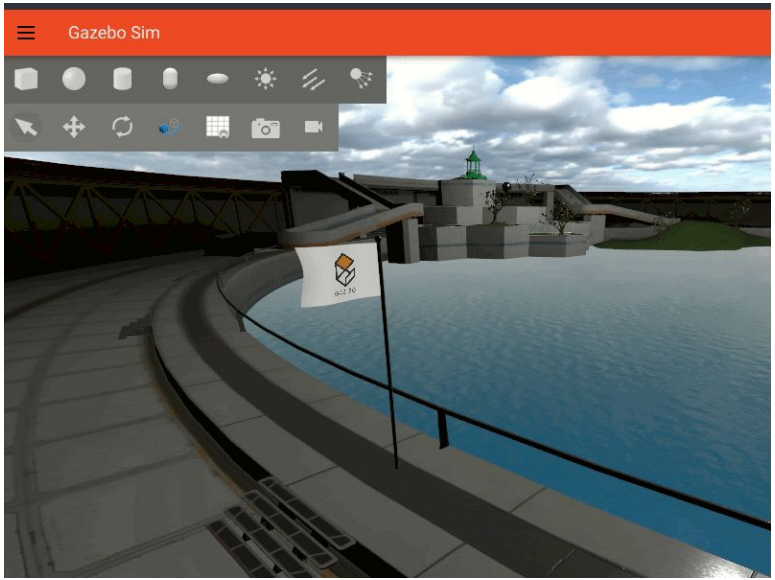


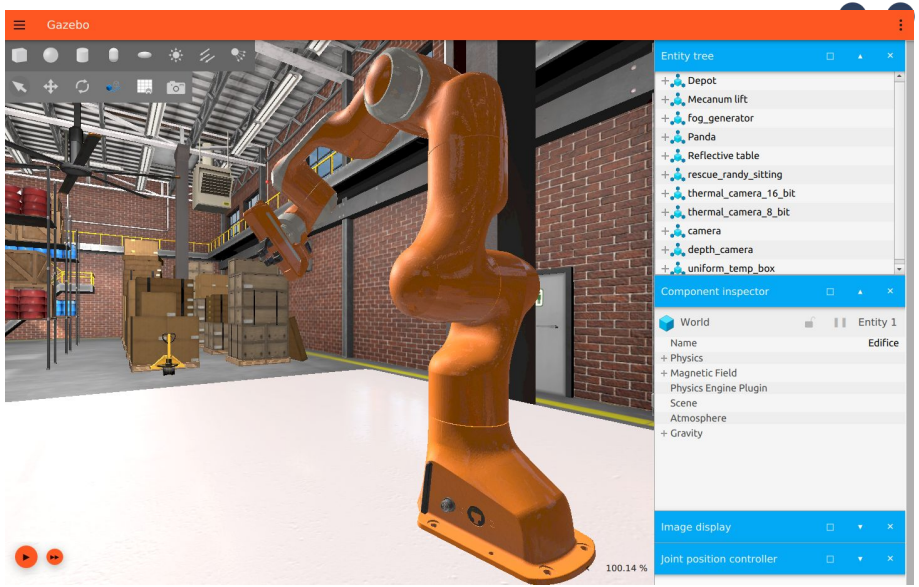






Gazebo, the robot simulator https://gazebosim.org/

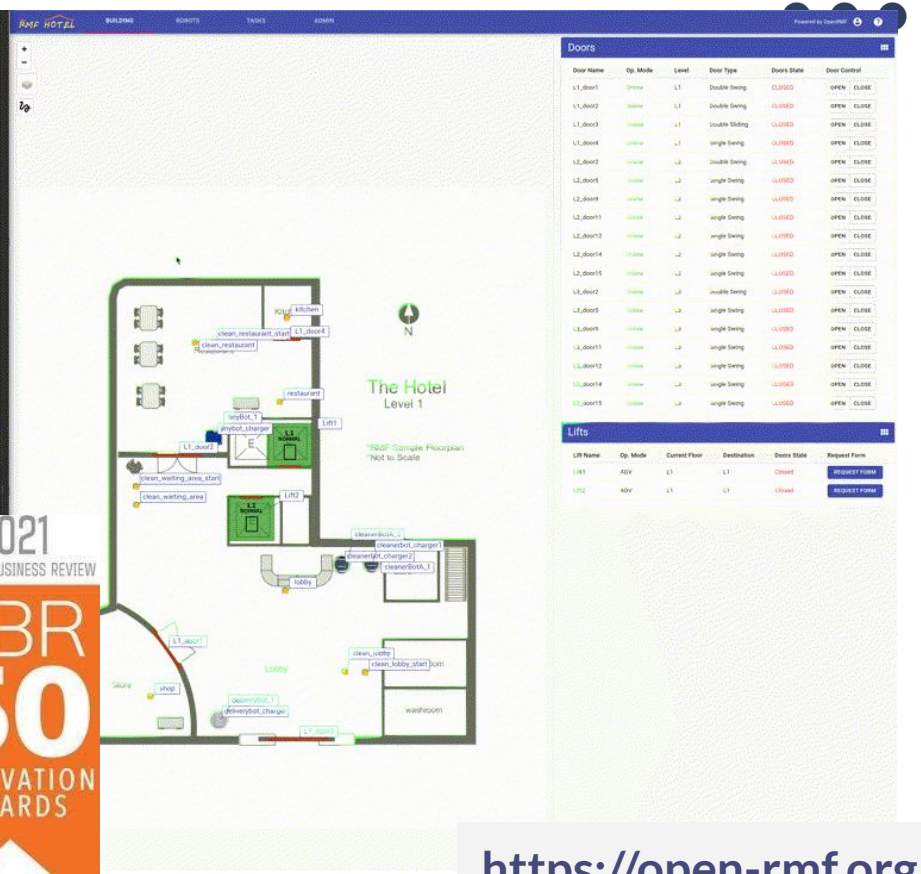






Open-RMF, the open fleet manager

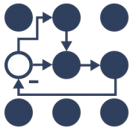




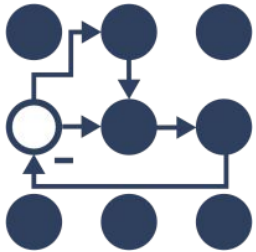
2021
ROBOTICS BUSINESS REVIEW
RBR 50
INNOVATION AWARDS

Open-RMF

https://open-rmf.org/



ros-controls



control.ros.org

ros-controls Organization

Charter



Project charter

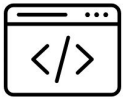
Meetings





CEST

Repositories



GitHub ros-controls

Docs



control.ros.org

ros-controls PMC members

Bence Magyar [Bent'seh]
- PhD in Robotics
- Principal Software Engineer at Locus Robotics
- ros2_control PMC Leader



Denis Štogl [Dr. Denis]
- PhD, Control Engineer and Roboticist
- Robotics Consultant at Stogl Robotics Consulting
- ros2_control PMC Co-Leader



Christoph Fröhlich
- Research Engineer at AIT Austrian Institute of Technology
- Motion planning and system integration
- ros2_control PMC Member



Sai Kishor Kothakota
- Experienced Roboticist
- Senior Robotics Engineer at PAL Robotics
- ros2_control PMC Member



Marq Rasmussen
- Software Engineer at Kuka Robotics
- Mobile manipulation and Planning
- ros2_control PMC Member



History

? 10^4-10^5 ?
Robots worldwide

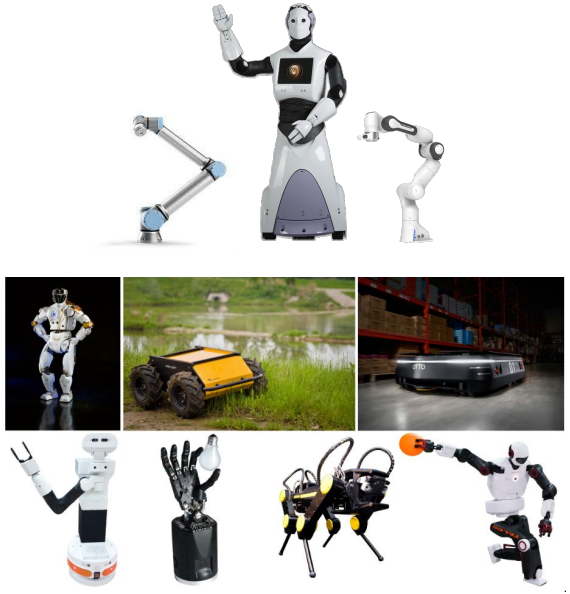
pr2_controller_manager

(pr2_mechanism)



ros_control

2012/2017

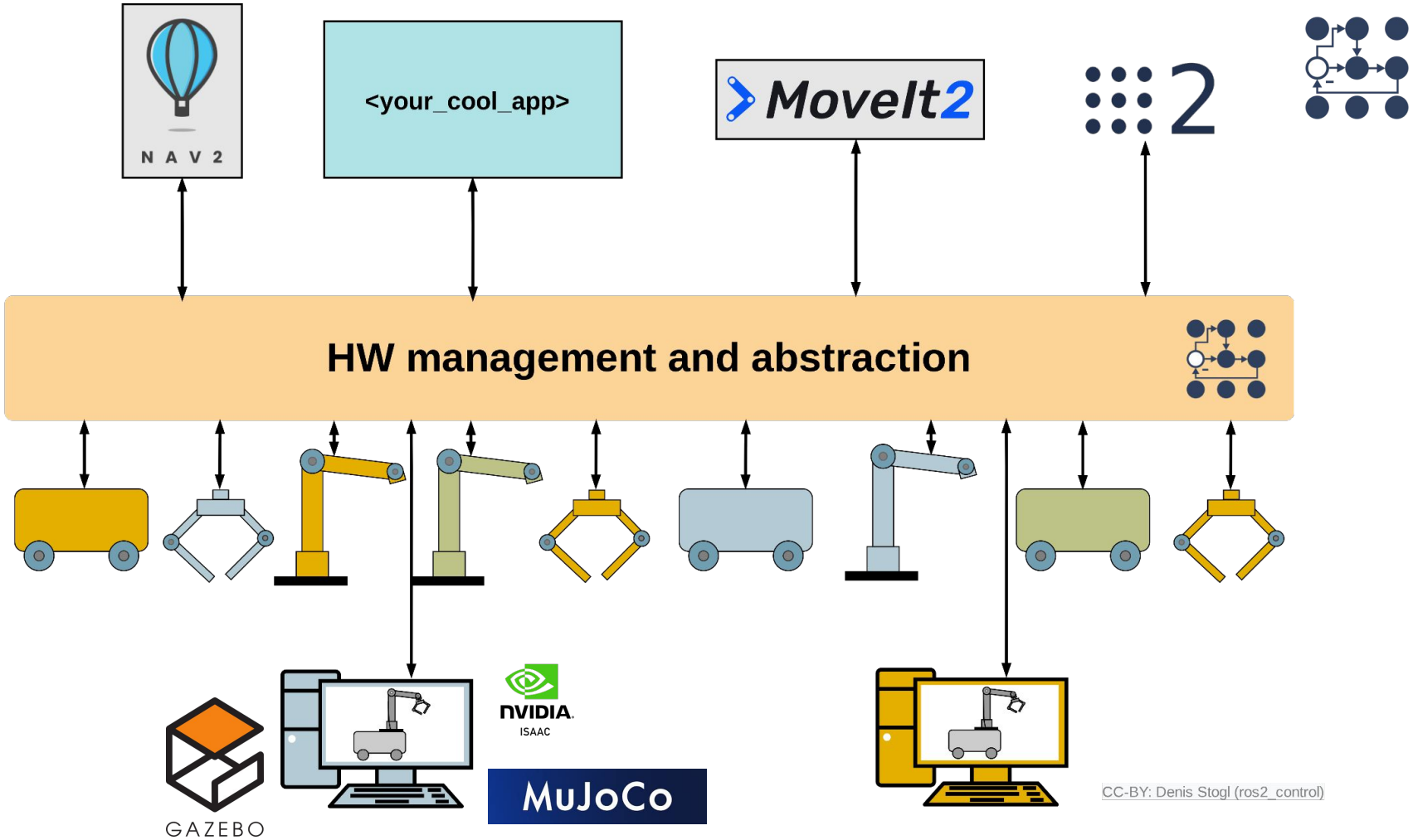


ros2_control

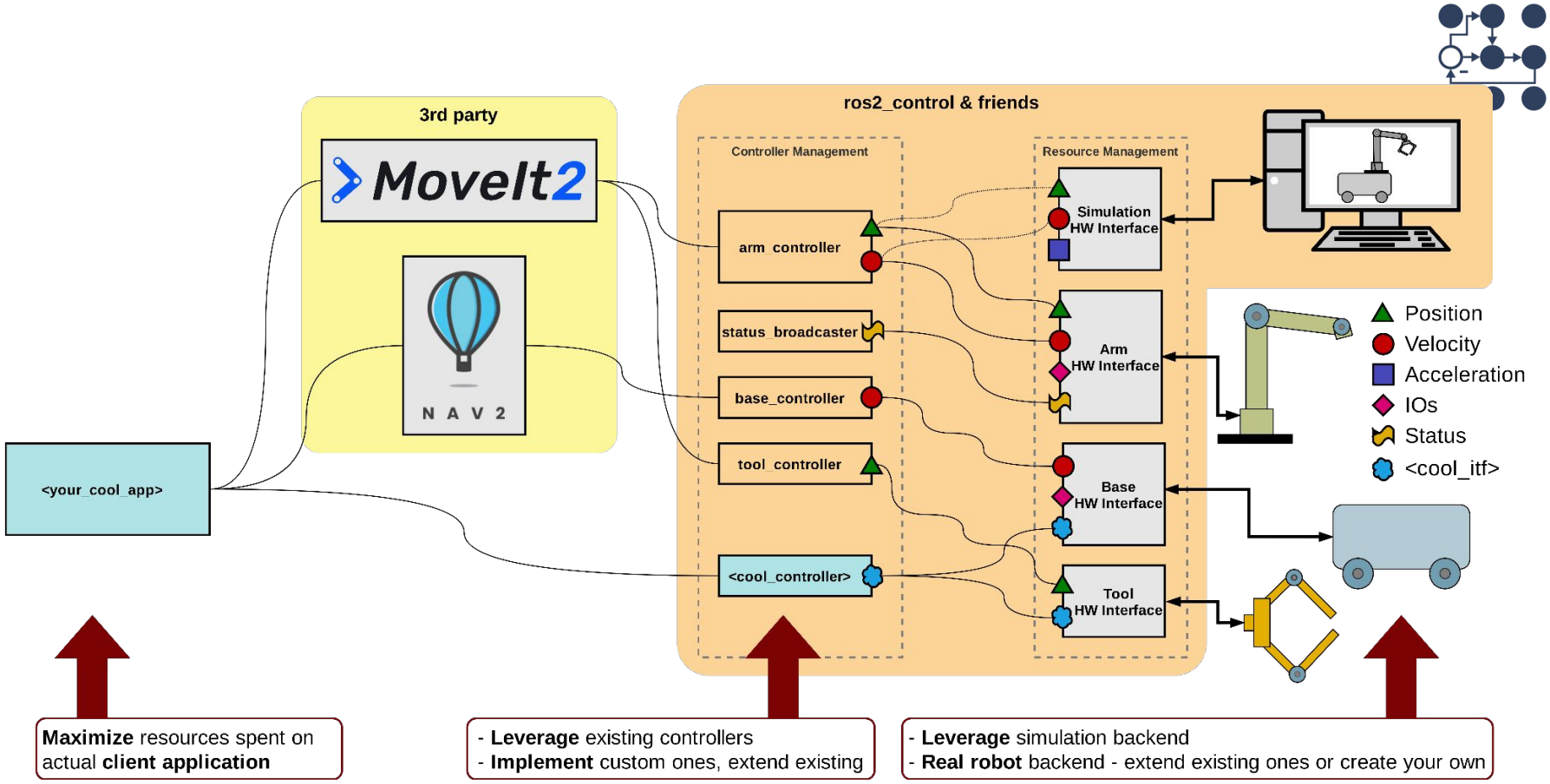
2017/today



https://control.ros.org/master/doc/supported_robots/supported_robots.html



CC-BY: Denis Stogl (ros2_control)


3rd party
MoveIt2
N A V 2
ros2_control & friends
Controller Management
Resource Management
arm_controller
status_broadcaster
base_controller
tool_controller
<cool_controller>
Simulation HW Interface
Arm HW Interface
Base HW Interface
Tool HW Interface
<your_cool_app>
Position
Velocity
Acceleration
IOs
Status
<cool_itf>
Maximize resources spent on actual **client application**
- **Leverage** existing controllers
- **Implement** custom ones, extend existing
- **Leverage** simulation backend
- **Real robot** backend - extend existing ones or create your own

Overview of Hardware Components (Drivers)

Automation / Communication

CANopen

EtherCAT

ctrlX AUTOMATION

BECKHOFF TwinCAT ADS















Non-robot devices

End-effectors







DG 5F



DG 4F

https://control.ros.org/rolling/doc/supported_robots/supported_robots.html

Overview of Hardware Components (Drivers)

Official Robot Drivers
















From Community

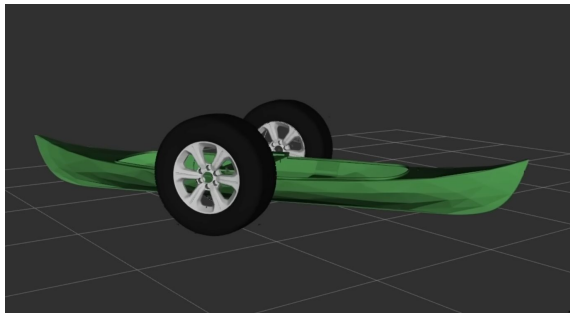


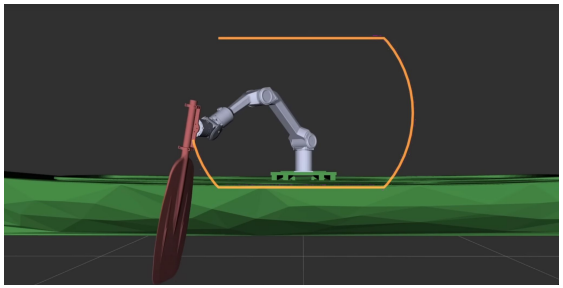


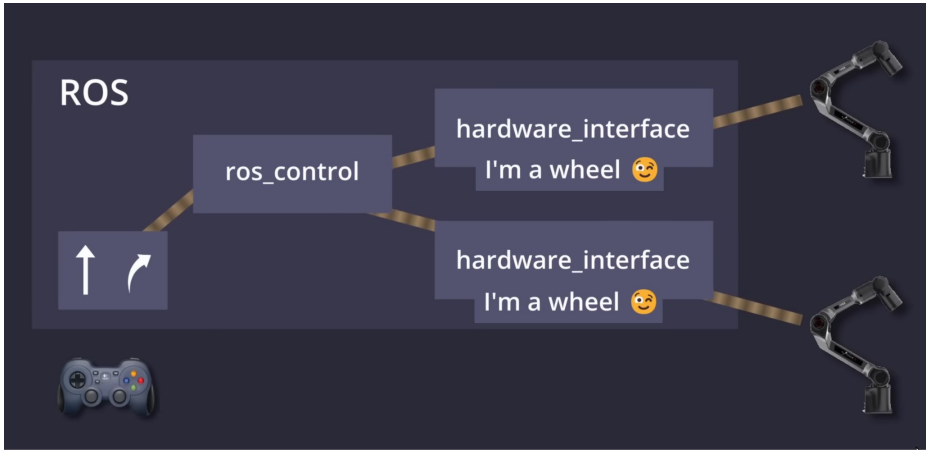

https://control.ros.org/rolling/doc/supported_robots/supported_robots.html

Rowing boat with 2 robot arms & ros2_control







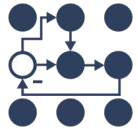


diff_drive_controller robot

By Dave's Armoury

https://youtu.be/XQX0SXHnbyk https://github.com/davesarmoury/rowboboat

Simulator Integrations



- maintained by ros-controls



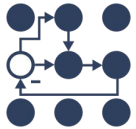


(Mock Hardware)

- 3rd party integrations

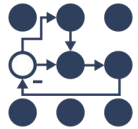






ros2_control and Embedded Hardware

What existed so far



rosserial / roslib
strictly running with an agent node. Messages go over serial or TCP to a bridge node on the host (rosserial_python), which republishes them into the ROS 1 network.

ROS 1 approach
End-of-life but conceptually relevant

micro-ros
lightweight middleware (XRCE-DDS).
An agent on a host PC bridges those devices into the normal ROS 2 network.

RTOS friendly, claims to support static memory but tricky.
It is lightly tied to FastDDS (may have quirks when used with other DDS providers)

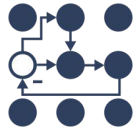
pico-ros
a minimal ROS client built on zenoh-pico and Micro-CDR, designed to work alongside rmw_zenoh.

still a little rough around the edges, ways to go for static memory,
Work In Progress but it’s Zenoh!


Zenoh



Zenoh



Developed by Zettascale as OSS, now ECLIPSE™ FOUNDATION

Pub/Sub/Query protocol that **Unifies data** in **motion**, data at **rest** and **computations** from embedded microcontrollers up the data centre

Provides **location-transparent** abstractions for **high performance pub/sub** and **distributed queries** across heterogeneous systems

Zenoh Implementations

Two major implementations:

Zenoh written in Rust and **Zenoh Pico** written in C and targeting micro-controllers
Written in Rust for security, safety and performance

Native libraries and **API bindings** for many **programming languages**, e.g., Rust, C/C++, Python, JS, REST, C# and Kotlin



Zenoh Runs everywhere

OS

Linux, MacOS, Windows, Android, iOS, QNX

Embedded Targets

Arduino, ESP32, mbed, Zephyr, FreeRTOS, ...

Automotive Targets

AUTOSAR Classic (microSAR)

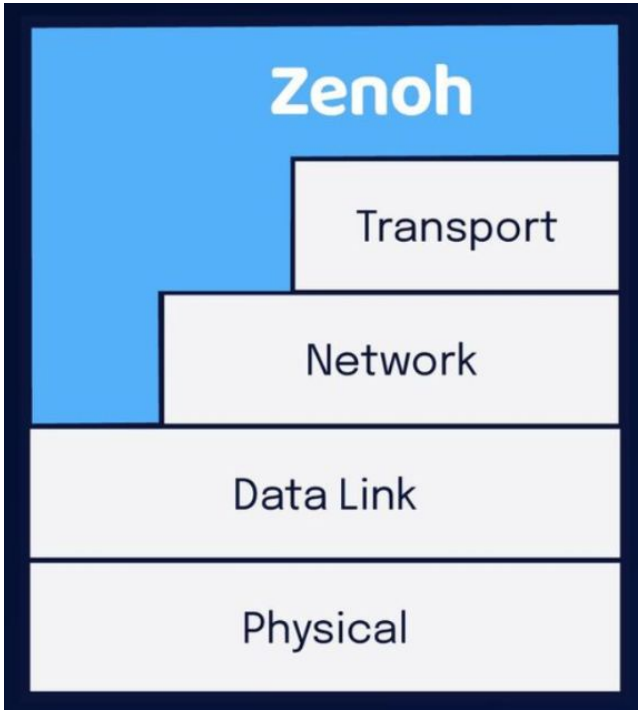


Zenoh Talks Everywhere

Built-in support Shared Memory and Zero Copy

Supports **network technologies** from **transport layer down-to** the **data link**. Currently runs on, TCP/IP, UDP/IP, QUIC, Serial, Bluetooth, OpenThreadX, Unix Sockets, Shared Memory

Available on **embedded** and **extremely constrained devices** and **networks** – 5 bytes minimal overhead

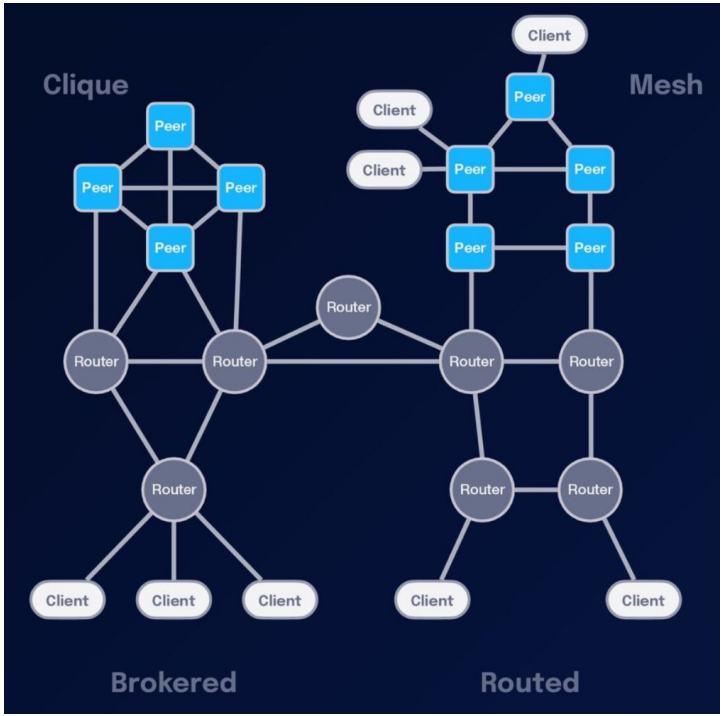


Zenoh Any Topology

Peer-to-peer
Clique and mesh topologies

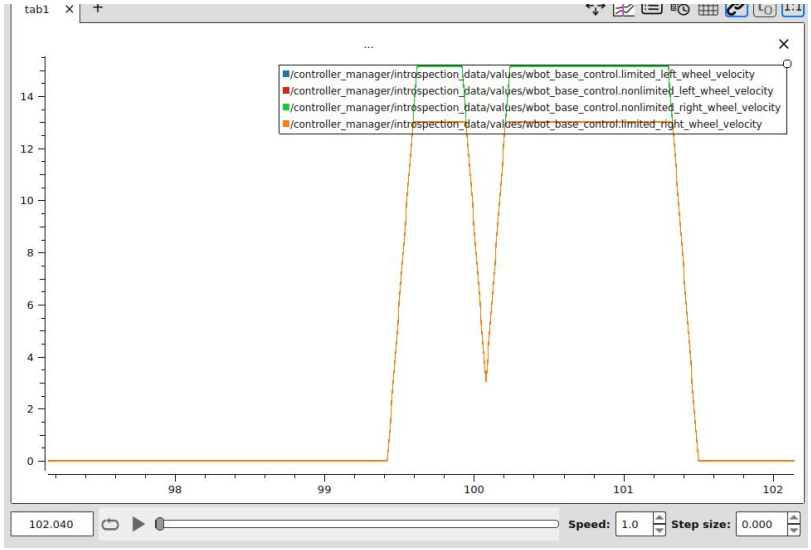
Brokered
Clients communicate through a router or a peer

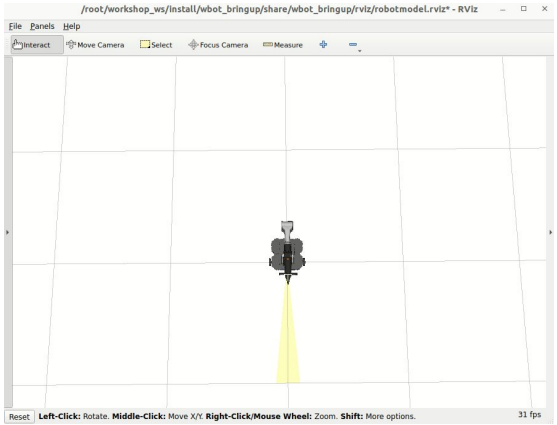
Routed
Routers forward data to and from peers and clients



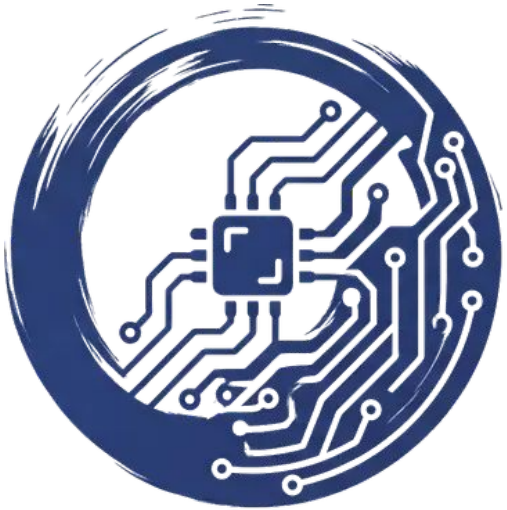
Workshops 2025

- Successful workshops with 100+ people in total at ROSCon and ROSCon UK 2
- ESP32 board speaking ROS natively over Zenoh Pico over serial or Wifi
- We () want to push for more embedded support









ZEN BEDDED

!! Only temporary, AI generated logo !!

Zenbedded Team

Eneji Peacemaker Ohieku
- Mechatronics Engineering Student



Shahazad Abdulla
- Robotics Engineer Student
- Team Captain, SAE-INDIA Autonomous BAJA
- Open Source Contributor



Kamalkant Thangaraju
- Mathematics and Computing Student



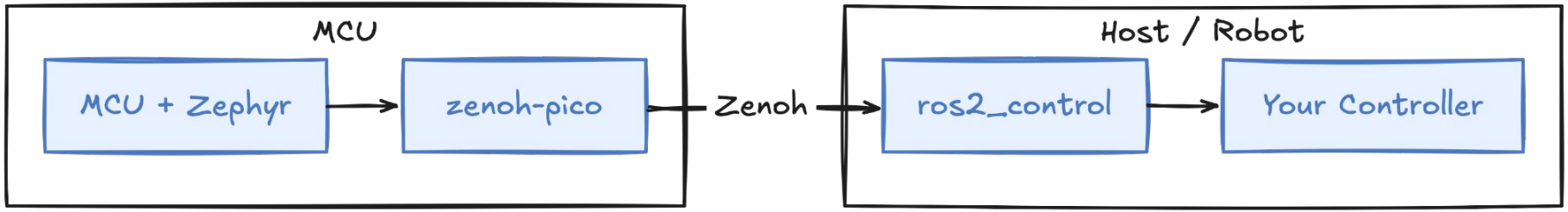
Krzysztof Pochwała
- Senior Software Engineer at Flytronic
- ROS2 enthusiast, creator of kabot.io



Bence Magyar
- PhD in Robotics
- Principal Software Engineer at Locus Robotics
- ros2_control PMC Leader

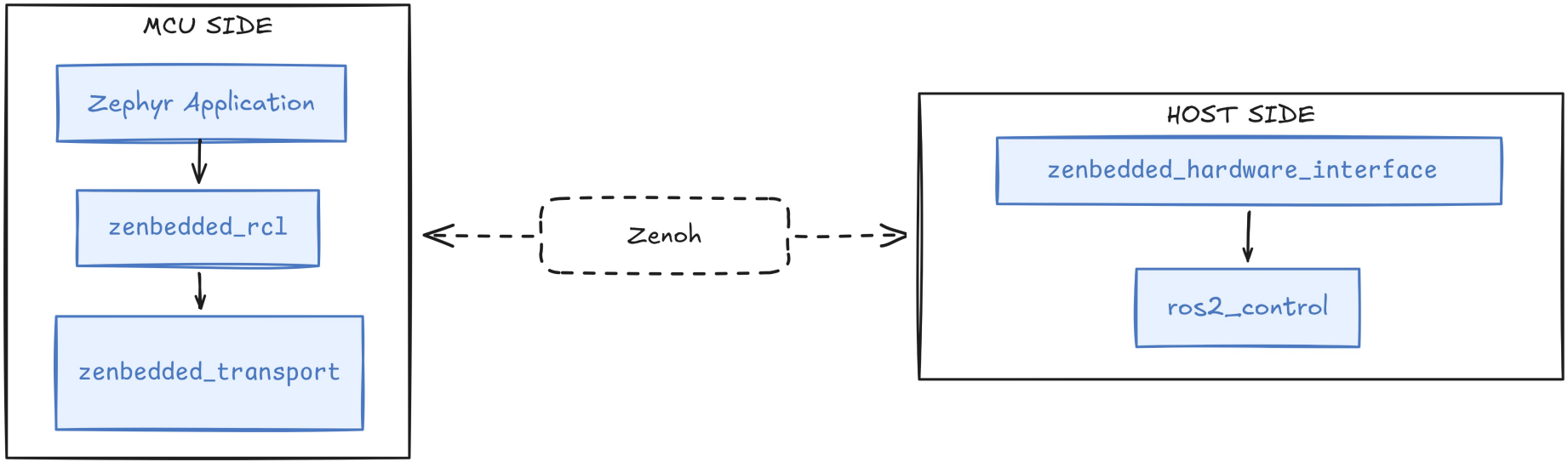


What Zenbedded Targets

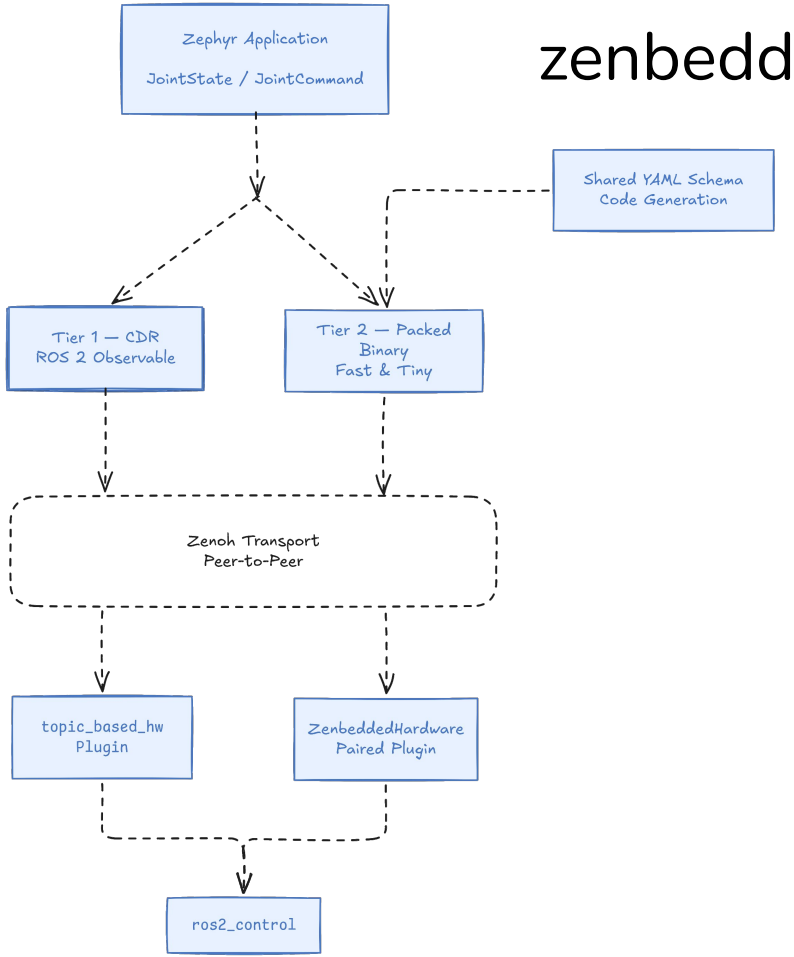


- **Lower Latency** → Direct MCU → zenoh → ROS 2 communication.
- **Less Jitter** → Choose the data format you need for each control loop, including **CDR or compact binary**.
- **Focus on control (for now)** → **Only a very small set of messages**, can take a few shortcuts and optimizations.

Zenbedded


MCU SIDE
Zephyr Application
zenbedded_rcl
zenbedded_transport
Zenoh
HOST SIDE
zenbedded_hardware_interface
ros2_control

zenbedded_transport



Core Idea

- **One Application | Two Serialization Choices**
- **Tier 1: CDR ROS 2**
  - Interoperability and observability
- **Tier 2: Packed Binary**
  - Compact data with zero runtime serialization
- **Common Transport**
  - Both tiers leverage Zenoh

Tier 1 CDR Architecture

- **Standard ROS 2 CDR**

Compatible JointState / JointCommand serialization.

- **Zero-Allocation Runtime**

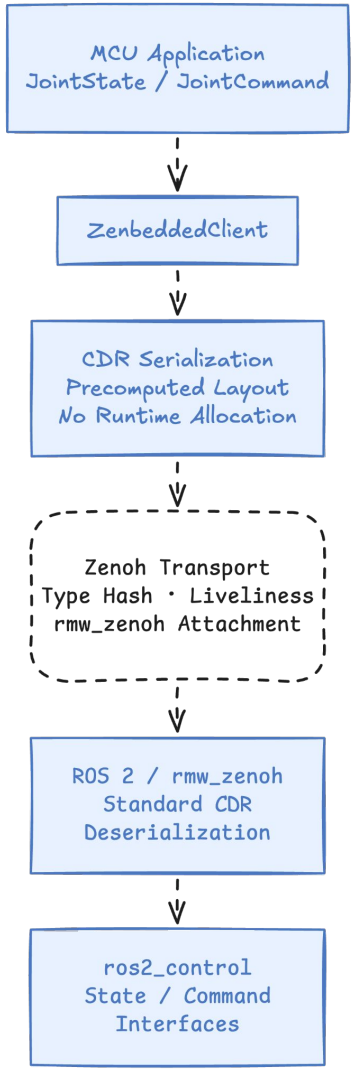
Precomputed buffer layouts and offsets

- **ROS 2 Interoperability**

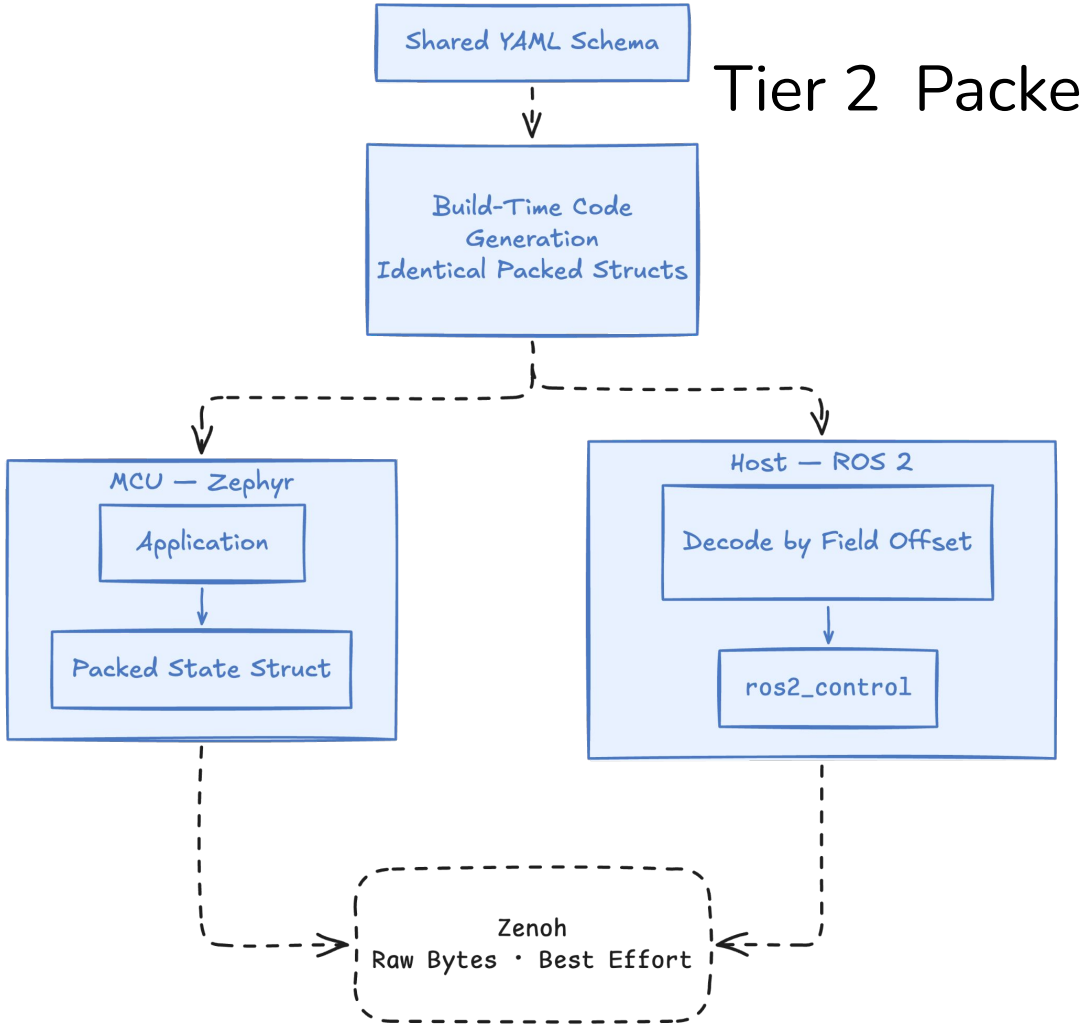
Type hash, liveliness, and rmw_zenoh attachment

- **Agent-less Communication**

MCU communicates directly through Zenoh



Tier 2 Packed Binary Architecture



- **Struct = Wire Format:** Raw packed C structs transferred over Zenoh.
- **Schema-Driven:** One YAML schema generates matching MCU and host layouts.
- **Efficient Transfer:** No CDR serialization; raw bytes wrapped in a Zenoh slice.
- **Trade-off:** Compact and paired-plugin dependent; not directly ROS 2 observable.
- **Agent in Communication:** Even though the decode section basically implements an agent, it is very light.

zenbedded_rcl: Embedded Client Library

- **Codec-Driven Client**

Connects the Zephyr application to the transport layer.

- **Static Double Buffers**

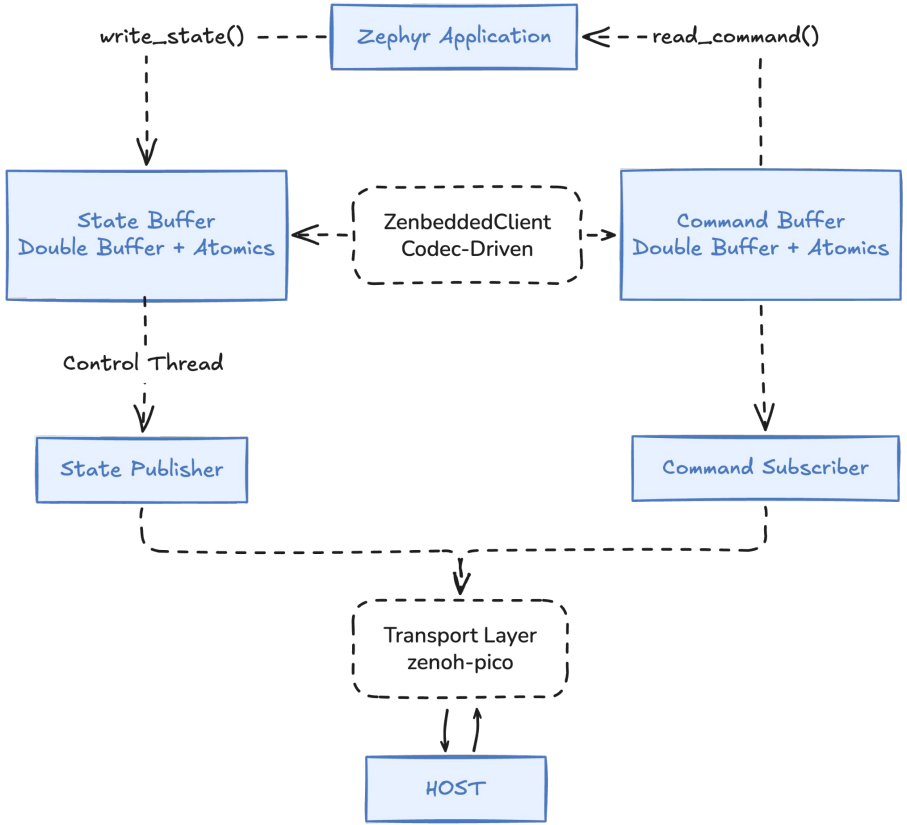
Utilizes versioned slots with acquire/release fences to enable lock-free data exchange.

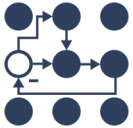
- **Background Control Thread**

Publishes state at a configurable, fixed frequency.

- **Tier-Agnostic Architecture**

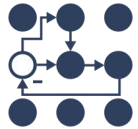
Seamlessly supports RawCodec (Tier 2) and CDR codecs (Tier 1)..





Benchmarks and Demos

Benchmarking



Roundtrip of 256 Bytes

native_sim (Theoretical Baseline on Intel I7 14th gen Host)

- Represents the absolute maximum transport ceiling.
- Raw throughput capability, **450 Hz** for **Tier 1** and **1250** Hz for **Tier 2**.

Theoretical Architecture Limits (Native Simulation)

Tier 1	*Tier 2*
450 Hz	*1250 Hz*

Real-World Hardware (ESP32 Classic)

- **Core & Network: Xtensa LX6 @ 240 MHz** operating over **2.4 GHz** Wi-Fi.
- **Tier 1 Performance: ~140** Hz (median RTT **~7.1 ms**, jitter **<1.8 ms**).
- **Tier 2 Performance: ~400** Hz (median RTT **~2.5 ms**, jitter **<0.5 ms**).
- **Resource Footprint:** Consumes **~16%** of a standard **4MB** Flash module and **<220 KB** RAM, preserving **>70%** of hardware capacity for user applications.

Real-World Hardware Limits (ESP32 Classic Wi-Fi)

Tier 1	*Tier 2*
~140 Hz	*~400 Hz*

Demo Project: Furuta Pendulum Robot

Overview & Hardware

What is it? Furuta pendulum

Hardware Specifications:

- **MCU Board:** ESP32-S3
- **RTOS:** Zephyr 4.4.2
- **Actuator:** Nema17 stepper motor
- **Sensor:** AS5600 Magnetic Encoder

Why this platform?

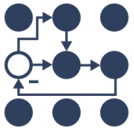
Requires high-frequency, low-latency control loops - a perfect stress test for Zenbedded's real-time capabilities.



Project video by
Henrique Ferrolho @ YouTube

Using Tier 2 transport, the state msg is 32 bytes and cmd 8 bytes!
+ 5 bytes Zenoh header

What's Happening Under the Hood



Control Loop

Controller Algorithm

LQR + Energy Controller

4 Input States

- Stepper angular position & velocity
- Pendulum angular position & velocity

Output Command

Stepper angular acceleration command

ROS 2 & Transport

ROS 2 Interface

ZenbeddedHardware (SystemInterface)

Transport Layer

Zenoh TCP connection over USB CDC using a Zenoh router on host

Target Firmware

Communicates directly with embedded zenbedded_rcl

Firmware & Safety

Multithreaded Architecture

400 Hz Thread: Sensor reading with Low-Pass Filter for angular velocity reading

100 Hz Thread: zenbedded publish thread

100 Hz Main Thread: Stepper motor control & RCL read/write

Safety Mechanisms

Software safety limits (e.g., hardstops at ±135°, 7rad/s velocity cap)

On disconnection stop motor

Demo Project: Furuta Pendulum Robot

Overview & Hardware

What is it? Furuta pendulum

Hardware Specifications:

- **MCU Board:** ESP32-S3
- **RTOS:** Zephyr 4.4.2
- **Actuator:** Nema17 stepper motor
- **Sensor:** AS5600 Magnetic Encoder

Why this platform?

Requires high-frequency, low-latency control loops - a perfect stress test for Zenbedded's real-time capabilities.



Demo Project: Mock Pendulum Hardware

Lagrangian Dynamics
MockPendulumHardware

- **Coupled 2-DOF Model**

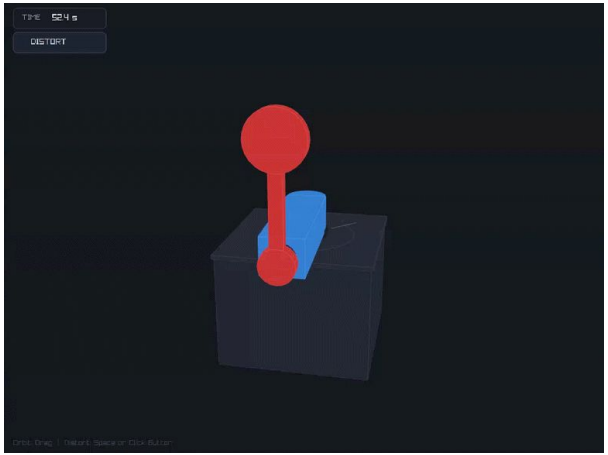
Parameterized with exact CAD values matching physical rig.

- **Actuator Feasibility**

Stepper torque saturation, step-loss limits & hard-stops.

- **Sensor Modeling**

Magnetic encoder, micro-steps, low-pass filter & noise.



3D Raylib Visualizer
PendulumVisualizer

- **Lock-Free Threading**

Runs independently using read() atomics without blocking.

- **Disturbance Injection**

UI [DISTORT] button triggers real-time balance testing.

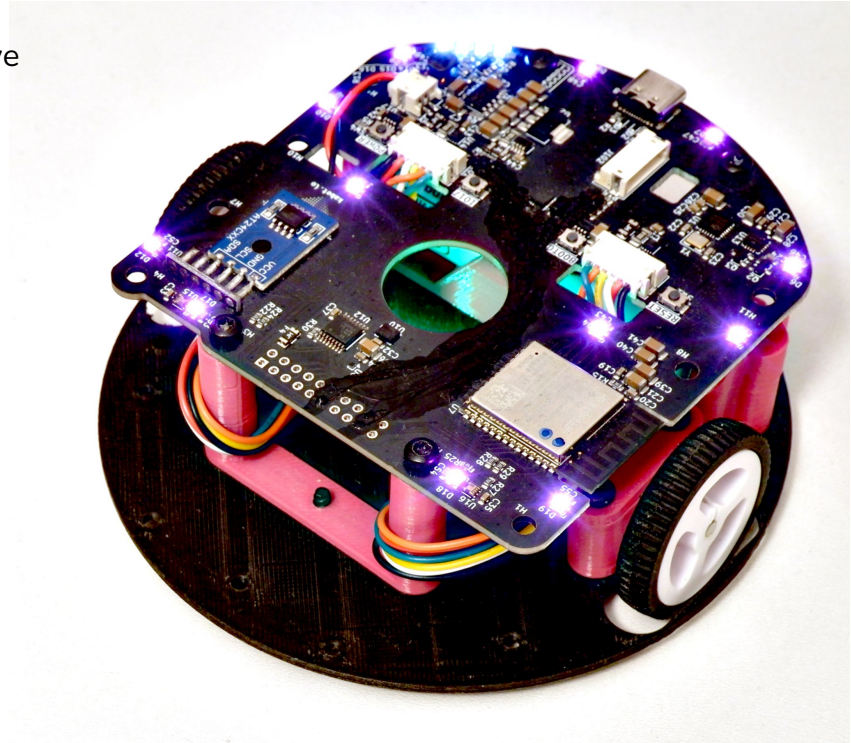
Demo Project: kabot.io diff drive platform

What is it? Open source (software and hardware) cost-effective differential drive robot for ROS2 learning

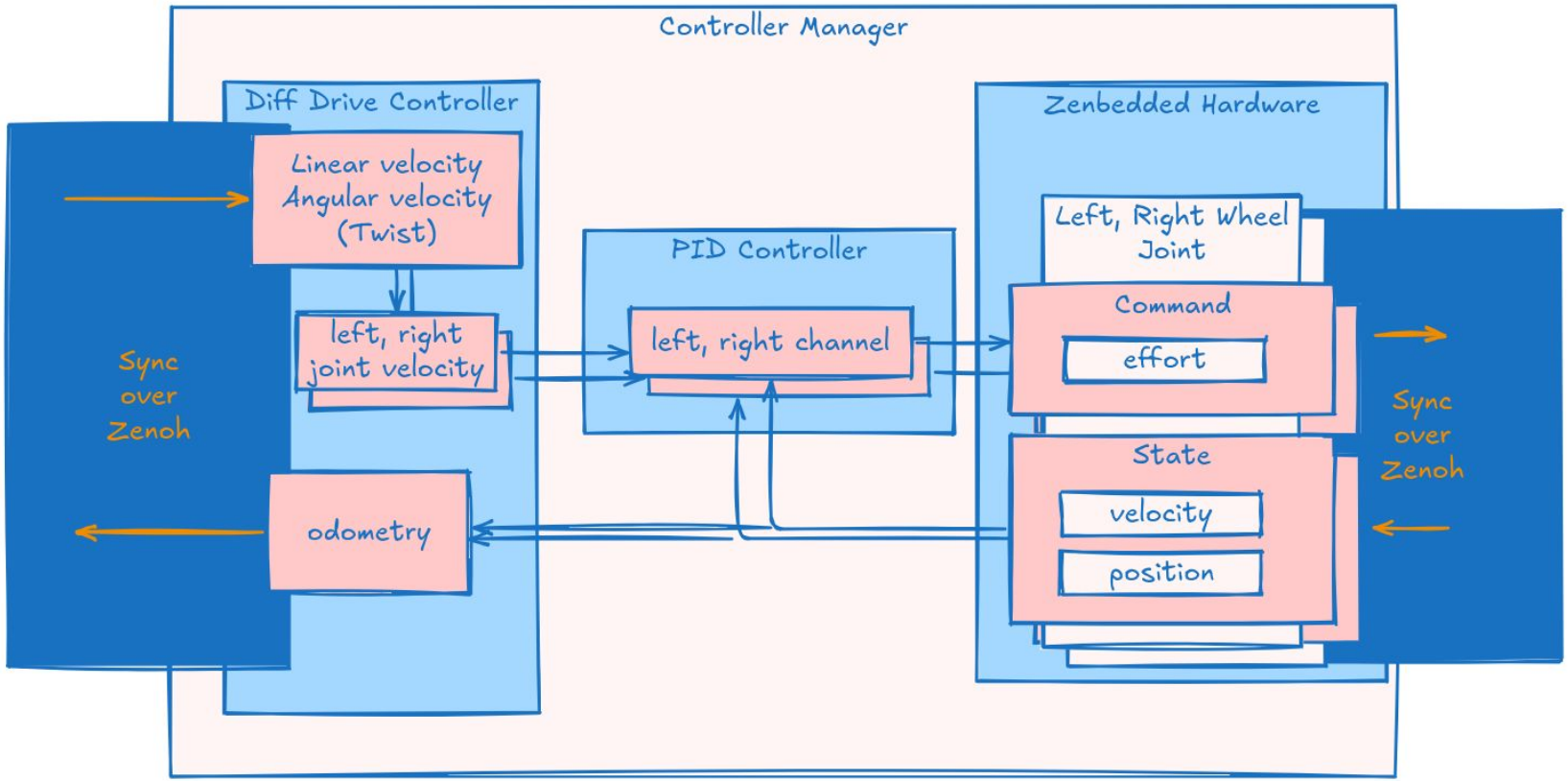
Hardware

- MCU: ESP32-S3 N16R8
- RTOS: Zephyr 4.4.1
- Actuators: 2x N20 298:1 DC Encoder Gearmotor
- Sensors:
  - ToF distance sensor
  - IMU
  - magnetometer
  - dual light sensors
  - motor current

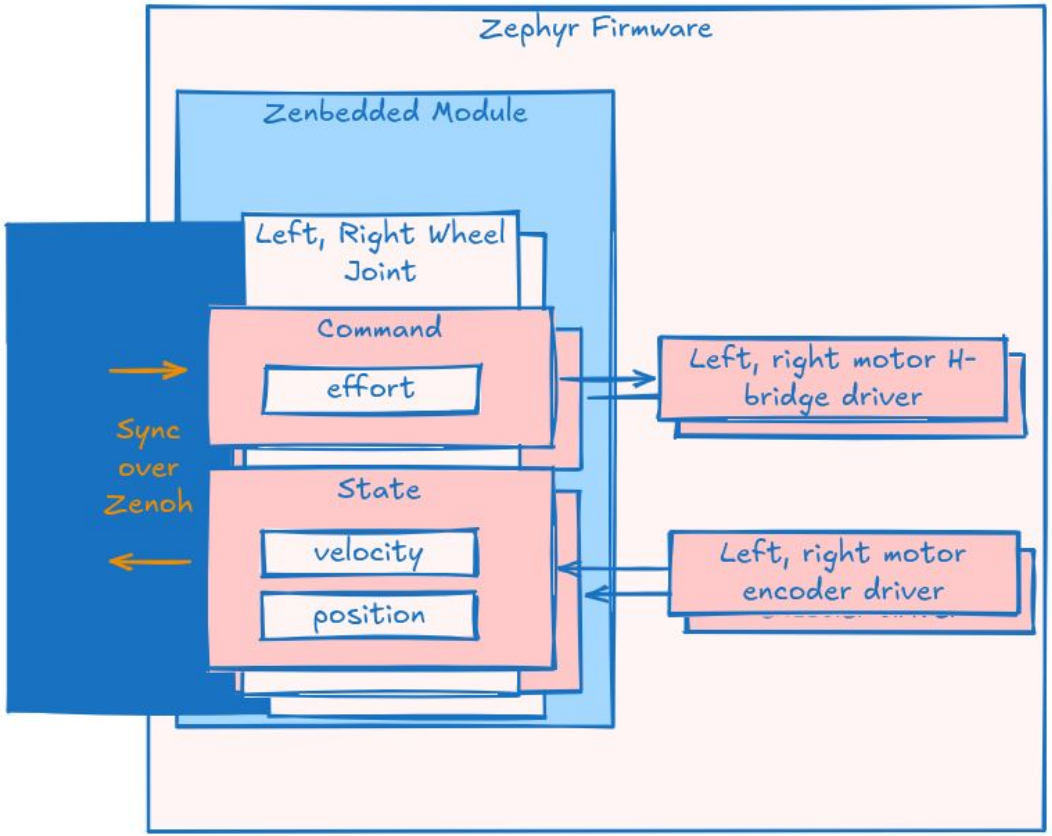
Why this platform? Multiple motors and sensors, Zenbedded generalization proof of concept.



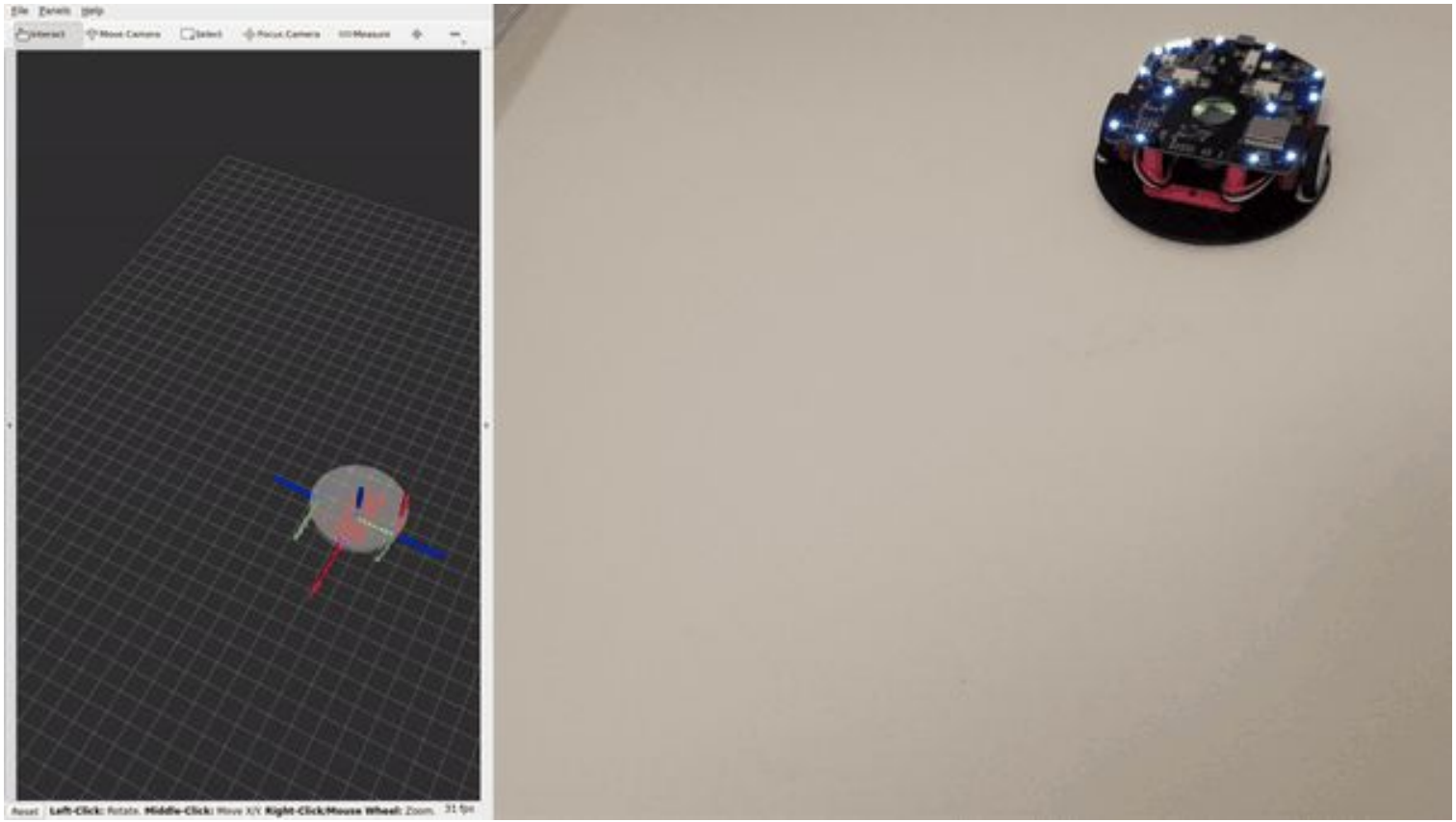
Demo Project: kabot.io diff drive controller - host



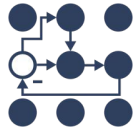
Demo Project: kabot.io diff drive controller - MCU



Demo Project: kabot.io diff drive rviz vs reality



What’s next



- **Make it Official:** Turn the project into an official ros2_control Working Group.
- **Hardware Expansion:** Extend hardware support beyond the ESP32 by leveraging Zephyr's broad board ecosystem.
- **Pipeline Optimization:** Transition schema generation to rosidl for native ROS 2 interfaces.
- **Transport Tiering:** Introduce a middle tier utilizing raw Zenoh values to optimize bare-metal data transport.
- **Generalize Tier 1:** Evolve Tier 1 into a general-purpose embedded-ros foundation.
- **Pure C/C++ Libraries:** Build out the hardware and controller interfaces as pure Zephyr libraries.
- **Simulation Integration:** Establish Zenbedded integration with Gazebo for robust virtual HIL testing with native_sim.

Shout out to friends @ Zephyr

Zephyr Robotics Working Group

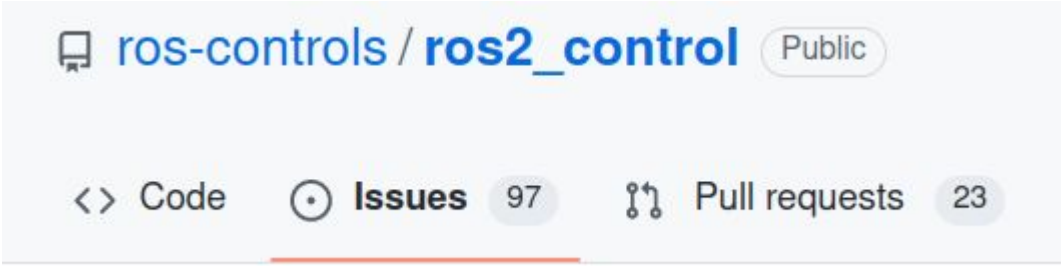


Join us!

Working Group Meetings
every second Wednesday!

Next one is 21st October!

- ros2_control presentations
  - https://control.ros.org/master/doc/resources/resources.html



- Github project to guide contributors to where they are most needed
  - https://github.com/orgs/ros-controls/projects/11



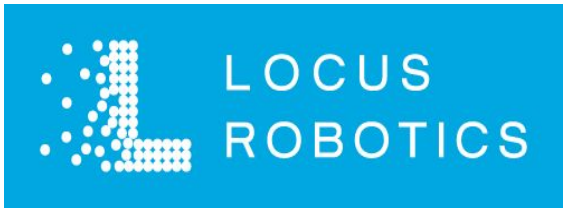
ros2_control reviewers



27 members



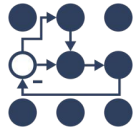












Christoph Fröhlich, Sai Kishor Kothakota, Marq Rasmussen, Bence Magyar, Denis Štogl, Erik Holum, Alejandro Hernández Cordero, Eneji Peacemaker Ohieku, Shahazad Abdullah, Kamalkant Thangaraju, Vedhas Talnikar, Ishan Pathak, Nikola Banović, Sahil Lakhmani, Christian Rauch, M. Ege Kural, Junius Santoso, Nathan Dunkelberger, Julia Jia, Soham Patil, Dennis Lanov, Sebastian Castro, Suryansh Singh, Tobias Fischer, Shlok Mehndiratta, José Luis Pérez Martín, Sergi de las Muelas, Krzysztof Pochwała, Shivam Maurya, Naitik Pahwa, Michele Cereda, Abdullah Theonewhomadethings, Felix Exner, Souri Rishik, Noel Jiménez García, Dhruv Patel ..... and many more!