

Welcome to **NEURA Robotics**, the innovator of the robotics world. Our goal is to equip collaborative robots with groundbreaking cognitive capabilities to enable safe and intuitive collaboration with humans. Under the leadership of founder David Reger, we have spent the first years of **NEURA Robotics** laying the foundations for humans and robots to work hand in hand.

"We serve humanity" is not just a motto, but our mission. Become part of our ambitious, international company and shape the future of robotics with us.

Welcome to **NEURA Robotics** - where innovation meets team spirit.

Your mission & challenges

Every NEURA robot runs on a foundation of systems integration work: the real-time OS that guarantees microsecond scheduling on bare metal, the hardware interface layer that abstracts every sensor and actuator, the motion planning pipeline that generates the trajectories robots execute, and the industrial device servers that connect robots to factory automation systems. As Staff Engineer for our Robot Systems & Integration cluster, you own the technical architecture of that foundation.

This is a pure individual contributor role. You carry no people management responsibility. Your authority is technical: you set architectural direction, hold design decisions within your scope, and resolve the cross-layer integration conflicts that arise when OS scheduling, middleware, hardware drivers, and motion planning pipelines all meet at the same seam.

You will provide technical leadership to engineers across platform guild and product-anchor roles. You write code, lead design reviews, author technical RFCs, and maintain current hands-on expertise. This

is not an architecture-only position.

- Own the RT scheduling policy that all software in this cluster must respect: priority assignment standards, memory locking requirements, and interrupt latency contracts for 1 kHz control loops and 500 Hz state estimation
- Own the co-review protocol at the boundary between RT OS scheduling and the EtherCAT master — any board support package change must pass a joint latency budget review before merge; you own this process alongside the Robot Communication cluster lead
- Ensure the unified compute platform engineers have clear RT OS architecture guidance and that every robot platform profile's timing budget is validated after each hardware revision
- Own the motion planning pipeline architectural direction: planner selection criteria, planning stack configuration standards, and atomic skill primitive API design and evolution across all robot platforms
- Enforce the shared skill primitive library: patterns surfaced from product-specific stream work must go through design review before being forked platform-specifically; you enforce this at code review
- Define the server/protocol boundary in industrial integration: the protocol stack belongs to the Robot Communication cluster; the server layer — command dispatch, state machine integration, hardware interfaces — belongs here; you arbitrate when server architecture and protocol constraints conflict
- Own the end-effector server architecture: gripper lifecycle management, tool change sequencing, and the hardware interface contracts that motion planning and the operational state machine depend on
- Lead quarterly cluster knowledge days: peer-to-peer problem exchange where every engineer brings one unsolved and one solved problem; output is a shared library ticket or design document, never slides
- Write design documents that reduce knowledge concentration; mentor Senior engineers toward Staff level; drive cluster hiring sourcing

What we can look forward to

- Robot systems integration depth (at least two of the following, in depth)
 - Embedded Linux systems integration at the RT level: Yocto BSP development, RT kernel tuning (Xenomai or PREEMPT_RT), RT scheduling design, and WCET analysis on production robot hardware — not simulation
 - ROS2 (or equivalent) hardware interface architecture: e.g. ros2_control hardware interface lifecycle, Nav2 and SLAM integration on physical mobile platforms, diagnostics framework design

- Motion planning pipeline for production manipulation: MoveIt2 planning stack, OMPL planner configuration and tuning, manipulation primitive library design — deployed on real robot hardware, not only in simulation
- Industrial device integration at the server layer: hardware interface design for fieldbus-connected devices, gripper lifecycle management server architecture, functional safety constraints in state machine dispatch
- Real-time skill execution or soft-RT scheduling: deterministic scheduling for action dispatch, worst-case latency analysis, RT-aware behaviour tree implementation in a production context
- Systems integration breadth
 - Sufficient understanding across the full stack to review MRs and arbitrate design decisions spanning OS, middleware, motion planning, and device integration domains simultaneously
 - Demonstrated experience debugging cross-layer failures at the OS–middleware–hardware boundary: RT scheduling overruns, ROS2 (or other middleware) executor latency issues under 1 kHz RT constraints, hardware interface failures during robot bring-up
 - C++17/20 systems programming: lock-free patterns, RAII, RT-safe memory management, 1 kHz control loop discipline
- Staff-level leadership (mandatory)
 - Demonstrated cross-team architectural impact: your design decisions changed how multiple teams work, not just your own domain
 - RFC or design document leadership with cross-team reach: you have resolved integration conflicts between teams and had your proposal adopted as the standard
 - Mentoring track record: at least one engineer you have materially accelerated toward a senior or staff-equivalent level
 - 8+ years of hands-on engineering experience with a strong robot systems integration focus
- Nice to have
 - RT OS kernel-level experience combined with ROS2 (other other middleware) depth — candidates who have debugged failures spanning both layers are rare and highly valued
 - Experience shipping a motion planning pipeline from development through to production on a manipulator or mobile manipulator
 - Industrial functional safety background at the software architecture level
 - Real-time execution engine implementation experience (not just usage)

- Open-source contributions to robot systems infrastructure: ros2_control, MoveIt2, Nav2, Xenomai, or equivalent

What you can look forward to

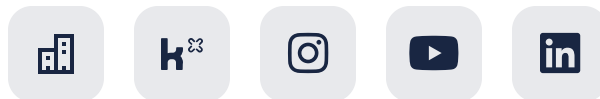
- Become part of an agile company, actively shape topics and benefit from flat hierarchies in a highly motivated team
- Enjoy an attractive salary, flexible working hours and 30 days of vacation
- The freedom to contribute your own ideas and drive them forward
- Celebrate successes together with company events
- Take advantage of our corporate benefits program
- And even more fun with great colleagues

[Apply](#)

We are looking forward to meeting you and shaping the future of robotics together. Are you in?

Couldn't find a suitable position? Please send us an unsolicited application.

We are always looking for passionate tech enthusiasts to help us revolutionize the world of robotics!



NEURA
ROBOTICS