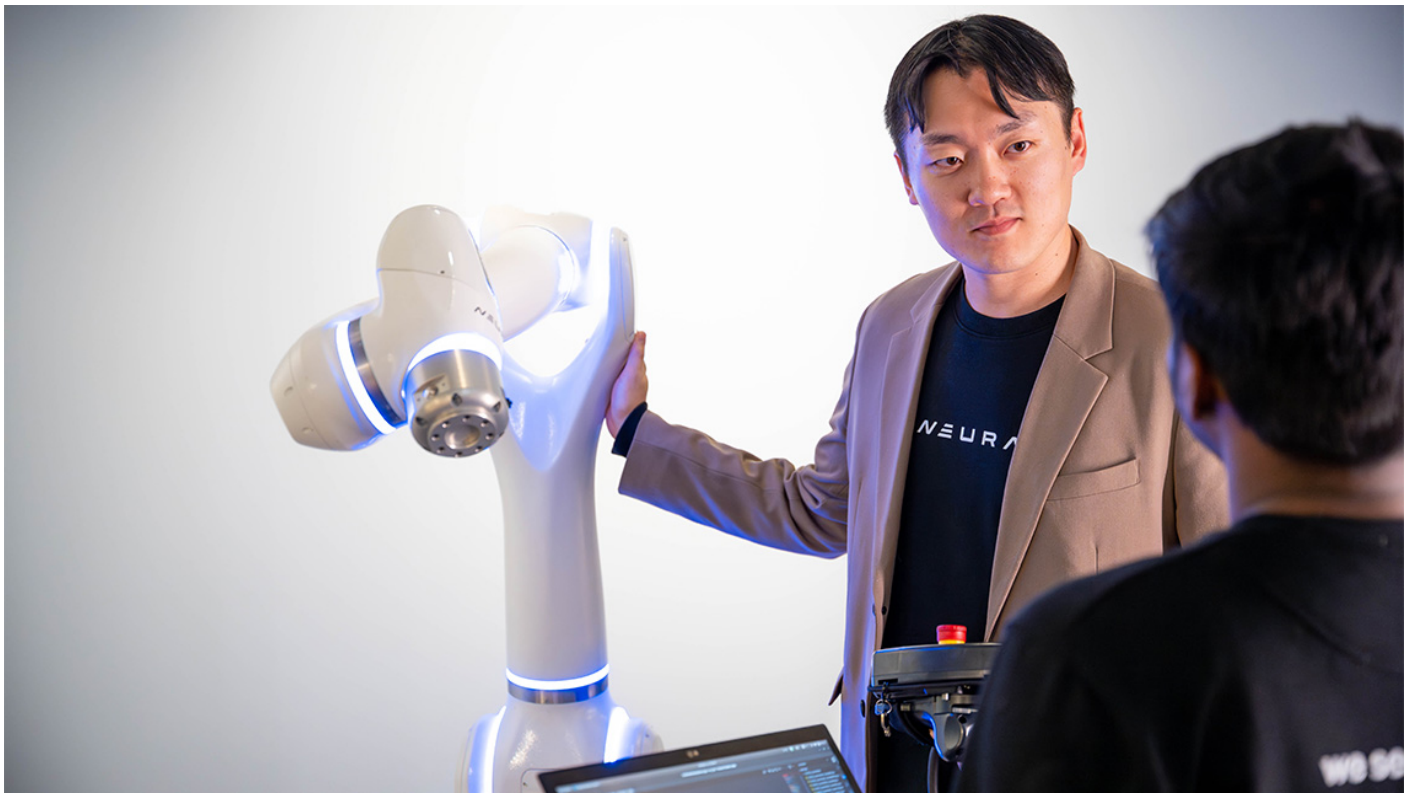

Welcome to **NEURA Robotics**, the innovator of the robotics world. Our goal is to equip collaborative robots with groundbreaking cognitive capabilities to enable safe and intuitive collaboration with humans. Under the leadership of founder David Reger, we have spent the first years of **NEURA Robotics**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

We are building dexterous humanoid hand platforms — multi-DOF tendon-driven systems with embedded tactile sensing, real-time motor control, and AI-driven perception running on edge compute. The stack spans custom PCBs, bare-metal firmware, ROS 2 middleware, and learned inference models, all converging in a physical system that must work reliably in the real world.
We are looking for a Test Engineer who can own validation across that full stack. Not a QA function that arrives at the end of a development cycle — someone who builds the test infrastructure from the ground up, defines what "working" means at every layer, and keeps the team honest throughout development.
The programme is early-stage. There is no existing test framework to inherit. You will design it.

- Hardware-Software Integration Testing

    - Design and execute integration test protocols for the hand control stack: motor driver boards, embedded compute modules, sensor interfaces, and the communication backbone (DDS over TSN Ethernet and SPI)
    - Develop hardware-in-the-loop (HIL) test rigs for validating firmware behaviour against real actuator and sensor hardware — brushed DC and BLDC motor channels, absolute encoders, tactile sensor arrays, IMUs
    - Define and automate bring-up test sequences for new PCB revisions: power-on checks, bus enumeration, driver smoke tests, and channel-by-channel functional validation
    - Own the integration test protocol for the forearm-to-body elbow interface: DDS topic correctness, latency measurement, link-loss behaviour, and safe-state transitions under fault injection
    - Test the full closed-loop control pipeline end-to-end: sensor input → embedded inference → motor command → physical response, with instrumented ground truth at each stage
    - Instrument and measure system timing: control loop jitter, DDS publish latency per topic, NPU inference latency, and end-to-end perception-to-action latency against defined SLAs
    - Validate mechanical-electrical interfaces: connector continuity through range of motion, cable harness stress testing, signal integrity under flexion cycles
- Software Testing
    - Build and maintain the SW test suite covering: ROS 2 nodes and DDS topic pipelines, motion primitive state machines, grasp sequencer logic, and safety watchdog behaviour
    - Design unit and integration tests for the embedded inference pipeline: ONNX model output correctness versus CPU reference, ring buffer behaviour, multi-task DDS publishing under sustained load
    - Implement regression test coverage for the control stack: position control loop stability, force ceiling enforcement, corrective tighten response timing, and arbitration logic between concurrent control modes
    - Define and run fault injection tests in software: simulate link loss, sensor dropout, classifier confidence below threshold, consecutive high-severity slip events — confirm the system transitions to the correct safe state in every case
    - Build simulation-based tests where physical rigs are unavailable: URDF-based motion validation, kinematic limit checking, and trajectory feasibility before hardware deployment
    - Maintain CI pipeline integration: automated test runs on every firmware and software commit, with clear pass/fail gates and failure triage
    - Own the benchmark test protocol for external hand platforms: define repeatable, instrumented test procedures
- Test Infrastructure
    - Select and deploy appropriate test tooling across the stack: logic analysers, oscilloscopes, force/torque sensors, motion capture or camera-based ground truth, data loggers
    - Build a structured test results database: every test run logged with software version, hardware revision, configuration, and outcome — traceable and queryable

- Write test specifications that other engineers can execute independently and reproduce your results
- Define acceptance criteria for each subsystem before integration begins — not after

What we can look forward to

- Essential
  - Degree in electrical engineering, mechatronics, computer science, or a related field Hands-on experience testing embedded systems: bring-up, bus protocols (SPI, I2C, UART, CAN), signal integrity measurement, and firmware debugging with real hardware
  - Experience writing automated software tests in Python or C++ — unit tests, integration tests, regression suites
  - Ability to read and understand firmware and software well enough to identify what needs testing and where the edge cases are, without needing to write all the production code yourself
  - Experience with instrumentation and measurement: oscilloscopes, logic analysers, current probes — comfortable setting up a bench and capturing what is actually happening
  - Structured thinking about failure modes: given a system description, you should be able to enumerate the ways it can fail and design tests that would catch them
- Strongly Preferred
  - Experience with ROS 2, DDS middleware, or real-time communication systems
  - Experience with hardware-in-the-loop testing or physical test rig design
  - Familiarity with motor control systems — brushed DC or BLDC — and the characteristic failure modes of position control, current limiting, and encoder feedback loops
  - Experience building or contributing to CI/CD pipelines for embedded or robotics software
  - Familiarity with force/torque measurement and tactile sensing systems
  - Experience testing learned or probabilistic systems: validating model output distributions, testing confidence thresholds, and defining acceptable behaviour under out-of-distribution inputs
  - Background in or exposure to functional safety testing — understanding the difference between testing for correctness and testing for safety is a significant advantage

- What Makes This Role Unusual
  - Most test roles in robotics are either purely software or purely hardware. This one spans both and demands genuine fluency at the boundary — the interesting failures in a system like this are almost always at the interface between a PCB revision, a firmware change, and a DDS schema update happening simultaneously.

- You will also be testing a system where the perception layer is a learned model running on an NPU. Defining what "correct" means for a tactile slip classifier feeding a safety-relevant motor command is not a solved problem. You will need to think carefully about how to validate probabilistic outputs in a deterministic control context, and how to define test coverage for a system that can fail gracefully rather than fail cleanly.
- The benchmark framework for external hand platforms is also yours to own. That work has direct strategic value — the results inform which external platforms we integrate and how our own platforms compare. It is high-visibility work with real programme impact, not a back-office function.

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