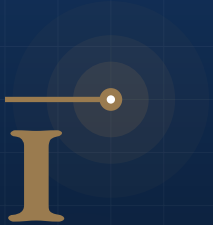
PRODUCT

Industrial Automation – R

Industrial Automation Robotics Nvidia Jetson

SiliconCentric | Product overview

SiliconCentric



Introduction

INDUSTRIAL AUTOMATION ROBOTICS NVIDIA JETSON

PRODUCT

Introduction

Our client is an industrial-robotics manufacturer. Their product is a mobile-manipulator robot whose control board pairs an NVIDIA Jetson SoM — perception and motion planning — with a dedicated external STM32 that owns the safety I/O, the safe-torque-off (STO) chain, and the motion watchdog. Six first-article boards arrived from fabrication two months before the motion-firmware team was due to start, and the client wanted independent proof the hardware was worth that team's time. We collaborated with them on this project by providing independent verification and validation of the board.

Introduction

- Compute topology: Jetson SoM plus external STM32 companion; the Jetson plans, the STM32 enforces — safety I/O, STO, watchdog, and drive-enable interlocks live on the microcontroller.
- Under validation: an EtherCAT master port serving six servo drives, dual BiSS-C absolute encoder inputs, a six-axis force-torque sensor on EtherCAT, dual-channel safety inputs and STO outputs, a 48 V-to-12 V power stage, CAN for the gripper, GigE for a wrist camera, and NVMe storage.
- Program state: fabricated first articles on the bench, motion firmware not yet started, commissioning date contracted with a launch customer.

Introduction

- Independence was the point: no stake in the design technically, managerially, or financially — findings against the safety chain cost us nothing to report.
- Scope was bounded to independent validation only: no feature development, no motion-control authoring, no out-of-scope engineering.
- Inputs were the hardware design package — schematics, ICDs, BOM — with no access to the client's firmware source.

2

Hardware Topology

INDUSTRIAL AUTOMATION ROBOTICS NVIDIA JETSON

PRODUCT

Hardware Topology

The topology is SoC-plus-companion-uC with a hard division of authority: the Jetson may request motion, only the STM32 may permit it. Scope enclosed both processors, their interconnect, the fieldbus, and the safety chain; motion firmware and any SIL certification sign-off stayed out.

- Topology classification: Jetson SoM plus external STM32, interconnected by SPI for cyclic state exchange and two hardwired discrete lines — drive-enable and fault — that bypass software entirely.

Hardware Topology

- The STM32's job: dual-channel safety-input evaluation, STO output control, the 20 ms motion watchdog on the Jetson, and 48 V rail supervision.
- In-scope buses: EtherCAT (drives and force-torque sensor), BiSS-C x2, CAN, GigE Vision, SPI (inter-processor), I2C (temperature, power telemetry), PCIe to NVMe.
- Envelope in scope: 0 °C to +50 °C cabinet ambient, 48 V nominal input with contactor-switching transients, and continuous-operation duty typical of a two-shift production cell.
- Out of scope: motion firmware, application code, and IEC 61508 certification sign-off — we produce the evidence; the client's functional-safety assessor judges it.

PRODUCT

Hardware Topology

- Jetson is one of the six architectures Polaris already runs on, so the OS baseline was inherited and effort went to the carrier delta, not silicon bring-up.

3

Problem Statement

INDUSTRIAL AUTOMATION ROBOTICS NVIDIA JETSON

PRODUCT

Problem Statement

The boards were real, the launch-customer commissioning date was contractual, and the motion team was about to build a SIL-2-adjacent product on hardware nobody had independently exercised. Vendor data covered every chip; nothing covered the board.

- Proven in isolation, never as a board: the EtherCAT PHY, the SoM, the safety relays, and the 48 V converter all had vendor test reports, but this carrier's EtherCAT routing, encoder front ends, STO chain, and inter-processor discretes had never run as one system.

Problem Statement

- Firmware coverage is feature-shaped, not board-shaped: motion firmware would exercise the drives and encoders daily, while the second safety-input channel, the CAN gripper port, the STO fault-injection paths, and power-telemetry I2C would go to commissioning untested by that team.
- No stress, no extremes, no duration: no 50 °C cabinet soak under six-axis load, no multi-day continuous-motion endurance, no contactor-transient and brown-out campaign on the 48 V input, no EtherCAT bus-saturation runs — the daily reality of a production cell.

Problem Statement

- Nothing runs alongside development: validation was planned as a pre-commissioning gate, so a marginal encoder input would surface as "the arm drifts" in the motion team's backlog — hardware faults filed as control-tuning problems.
- IEC 61508 SIL-2 claims on the safety chain require measured, documented STO timing and fault-response evidence, not design intent.
- EtherCAT cycle jitter budgets were tight — 250 µs cyclic with six drives — and jitter is a board-level property no vendor datasheet can promise.
- A safety-chain failure in a production cell is an injury investigation; a missed commissioning date is penalty clauses. The client needed both risks retired early.

4

Customer Requirements

INDUSTRIAL AUTOMATION ROBOTICS NVIDIA JETSON

Customer Requirements

The client's requirements were direct: validate the whole board, under stated limits, with evidence a functional-safety assessor would accept for the SIL-2 case.

- Operating temperature: the board shall operate from 0 °C to +50 °C cabinet ambient.
- Supply voltage: 48 V nominal, operating window 43 V to 54 V. The board shall survive contactor-switching transients up to 60 V without damage; over-voltage protection shall trip above 60 V.

Customer Requirements

- Brown-out: the board shall ride through input sag to 36 V and recover without operator intervention; below 36 V it shall shut down cleanly with no record loss.
- Over-current: the 48 V-to-12 V stage and each downstream rail shall have over-current protection tripping within 10 ms; on any rail fault the STM32 shall drop drive-enable.
- STO timing: the safe-torque-off chain shall react within 10 ms on both channels, across the full temperature range.
- Humidity: the board shall operate at up to 85% RH non-condensing and shall not be exposed to condensing conditions.

Customer Requirements

- Full-board functional validation: all Jetson cores and the GPU, every DRAM region, EtherCAT with six drives and the force-torque sensor, both encoder inputs, both safety channels, STO outputs, CAN, GigE, NVMe, and the 48 V stage — as one integrated system.
- Coverage independent of firmware scope: every subsystem validated regardless of the motion feature set, through a HAL built from the design package.
- Stress matrix: 0–50 °C sweep, 120-hour continuous-motion-equivalent endurance, 48 V contactor-transient and brown-out campaigns, EtherCAT saturation with jitter measurement, and STO fault injection across temperature; vibration limited to cabinet levels.

Customer Requirements

- Continuous validation alongside development: CI integration revalidating every firmware candidate on real carriers throughout the program.
- Regulatory evidence: timestamped, operator-attributed, version-pinned records with bidirectional traceability, structured for the IEC 61508 assessment.
- A web console any cell technician can run — configure, launch, read results — without knowing the stack.
- An immutable, append-only evidence store, reusable as the baseline when the carrier revs for the next robot variant.

Customer Requirements

- Setup in minutes: standardized bench procedure, no custom rig, identical across all six first articles.

Architecture

INDUSTRIAL AUTOMATION ROBOTICS NVIDIA JETSON

Architecture

The carrier runs the validation OS, HAL, and client; an external x86 host runs the server, dashboard, and SDK. The split isolates execution from data processing and preserves the Jetson's compute for realistic load — which mattered because EtherCAT jitter had to be measured with the GPU busy, the way production will run it.

- Polaris OS (the validation OS) — a reproducible Linux image for the SoM booting to a known validation state, so six first articles start from one baseline instead of six bench configurations.

Architecture

- HAL — the single board-specific layer, exercising EtherCAT, BiSS-C, CAN, GigE, the STO chain, the SPI inter-processor link, and power telemetry behind a uniform interface that keeps the robotics suites portable across carrier revisions.
- Client — the on-target agent driving every test through the HAL; gRPC on the Jetson, FlatBuffers to the STM32's Zephyr image; three timestamps per event (origin, relay, server); low enough load that jitter measurements reflect field behavior.
- Server — orchestration, scheduling, and the append-only records store on the host, running all six carriers in parallel and hosting the dashboard and API.

Architecture

- Dashboard — the operator console: provision a carrier, configure and launch, live cycle-time, rail, and safety-state telemetry, time-series review after; every state-changing action operator-attributed for the safety file.
- Python SDK — automation and CI; scripts everything the console does; C++ bindings served the timing-injection harness.
- The OS and HAL are the two board-specific layers and were heavily modified for this carrier and this SoC; the client, server, dashboard, and Python SDK were configured to the client's requirements, but their core structure and working style remain the same across every board and every industry.

6

Design

INDUSTRIAL AUTOMATION ROBOTICS NVIDIA JETSON

Design

Four constraints bound the design: the 250 µs EtherCAT cycle, the 10 ms STO reaction budget, a 50 °C sealed cabinet, and a functional-safety evidence chain that must attribute every record to an operator.

- Validation OS design — PREEMPT_RT-patched kernel on the Jetson because cycle-jitter measurement demands it, read-only rootfs with A/B rollback; the trade-off is GPU-driver friction under RT patching, worked through rather than hidden. The STM32 runs Zephyr — chosen over FreeRTOS for its native dual-channel GPIO and watchdog driver model.

Design

- HAL design — partitioned by subsystem against schematic sheets and ICD clauses; EtherCAT and encoder capture are interrupt-driven since polling would corrupt the jitter statistics being measured. The risk is real: the force-torque sensor's fault mode flooded the interrupt line, and storm throttling had to be added after it locked a carrier mid-campaign.
- Client design — origin timestamps latched at the driver boundary; on STO trip the client records the full pre-trip window before yielding, trading 30 ms of shutdown latency for a complete record of every safety event.

PRODUCT

Design

- Server design — three-timestamp causality orders a 48 V sag against the drive fault it triggered across six carriers at once; retention sized for 120-hour runs times six boards, with cyclic telemetry downsampled after 30 days — a storage trade the client accepted.
- Dashboard design — cycle-time histograms, per-drive EtherCAT working-counter errors, STO channel states, and rail telemetry are first-class views, because that is what a robotics commissioning engineer triages; run health and drop counts on every view.
- SDK design — the client scripted the acceptance gate, the STO fault-injection matrix, and overnight endurance; the direct API path handled single-drive bench checks, with the explicit cost that those runs leave no records.

Design

- OS and HAL design was board-specific and heavily reworked; the four upper components were configured to this client's workflow while their structure and working style stayed constant.

7

Implementation

INDUSTRIAL AUTOMATION ROBOTICS NVIDIA JETSON

Implementation

Effort concentrated in the OS delta, the HAL, and the robotics campaigns; the client, server, dashboard, and SDK were configuration work on a stack that already existed.

- Validation OS — Yocto image for the Jetson SoM: inherited BSP and bootloader, carrier device tree (EtherCAT MAC, encoder front ends, STM32 SPI link), PREEMPT_RT kernel config; OTA with A/B rollback; boot to validation-ready under 30 seconds; per-build SBOM with CVEs tracked; STM32 Zephyr image built under the same stack.

Implementation

- HAL — built subsystem by subsystem from the design package and bench-verified as written: DMA paths for EtherCAT cyclic data, interrupt handlers for encoder and safety inputs, register-level control of the STO drivers, bus-timing work on the SPI link; the package review flagged a BOM error — the dual-channel safety inputs specified 24 V-rated optocouplers on the schematic while the BOM carried a 12 V part, caught before it could pass bench tests and fail in a 24 V cell.
- Client — C++ daemon, event and polling paths, hardware timestamping, graceful shutdown without record loss, gRPC upstream, FlatBuffers to the STM32.
- Server — telemetry receiver, versioned schema, scheduler, REST API, dashboard hosting, six-carrier parallelism.

PRODUCT

Implementation

- Dashboard — carrier provisioning, campaign configuration, live cycle and safety streaming, time-series review, export for the safety file, operator attribution throughout.
- SDK — Python bindings, C++ harness bindings, example scripts per subsystem, and CI wiring that fires the acceptance suite on every firmware candidate.
- Board-specific: OS delta and HAL. Configuration: client transport, server retention policy, dashboard views, SDK examples and CI glue.

Test - IV&V

INDUSTRIAL AUTOMATION ROBOTICS NVIDIA JETSON

PRODUCT

Test - IV&V

Testing ran stack-first, then OS/HAL, then the board across the four run shapes — one-shot, long-running, event-based, monitoring — with every dashboard-launched run committed to the append-only store.

- Stack self-verification: 100% test coverage of the client, server, dashboard, and Python SDK before any of it generated evidence about the carrier.
- OS/HAL verification: A/B boot and rollback cycled, OTA verified end to end, secure boot checked on accept and reject paths.

Test - IV&V

- Bench setup: carrier cabled to the host with a drive emulator rack and programmable 48 V supply, provisioning from the dashboard — repeatable in twenty minutes per carrier.
- Functional campaign: EtherCAT cyclic and mailbox suites across six drives, working-counter and jitter runs, BiSS-C position-integrity suites, force-torque data-path checks, CAN gripper suites, GigE throughput, dual-channel safety-input and STO verification.
- Compute and memory campaign: DRAM pattern stress and cache/DMA integrity under EtherCAT saturation, GPU load with thermal-throttling characterization at 50 °C cabinet ambient, and interrupt-latency distribution measured under the PREEMPT_RT kernel with perception load running.

Test - IV&V

- Watchdog and fault-injection campaign: the 20 ms motion watchdog exercised with induced Jetson hangs confirming drive-enable dropped within budget, dual-channel discordance injected on the safety inputs verifying single-channel faults are detected and latched, and the STM32's own independent watchdog proven to fail toward STO.
- Debug posture: JTAG/SWD on the STM32 verified locked in the deployment configuration, retained on first articles where trace anchored the STO timing measurements.
- One-shot tests as the acceptance gate: one pass/fail per subsystem per firmware candidate, per carrier.

Test - IV&V

- Long-running: 120-hour continuous cyclic traffic with snapshots caught NVMe throttling above 47 °C cabinet ambient, and a slow EtherCAT jitter drift on one carrier traced to a marginal 25 MHz PHY crystal.
- Event-based: silent runs recording on working-counter errors, cycle overruns, rail excursions, and safety-state changes — this mode documented the STO chain exceeding its 10 ms reaction budget at 50 °C, measuring 13.7 ms on channel B, traced to an underdamped optocoupler stage and corrected in a component change.
- Monitoring: passive telemetry — rails, SoM and board temperatures, CPU/GPU load, cycle statistics — under a perception-plus-motion representative workload.

Test - IV&V

- Instrumentation correlation: oscilloscope captures anchored the STO timing and the 48 V contactor-transient behavior to the records; one brown-out profile forced a hard reset that the watchdog caught but that corrupted the NVMe filesystem — recovery was then verified deliberately.
- Recording rule: every run launched from the dashboard is recorded, always, into the append-only store; the Python SDK can run 100% of what the dashboard can run — module, peripheral, or system level — for quick independent checks, and SDK-launched runs of this kind are not written to the records store. The store is the evidence; the SDK is the bench tool.

Test - IV&V

- Traceability: requirements traced bidirectionally to tests and records, schemas version-pinned, every run reproducible for the assessor.

9

Deliverables

INDUSTRIAL AUTOMATION ROBOTICS NVIDIA JETSON

PRODUCT

Deliverables

The client took delivery of the full platform fitted to their carrier — source, records, procedures, no lock-in.

- Validation OS image with the complete Yocto build system, recipes, and per-build SBOM.
- Carrier-specific BSP and HAL source for every subsystem, maintained against the design package across revisions.
- Compiled client for the Jetson plus the STM32 Zephyr build.

Deliverables

- Server and dashboard as deployment-ready containers on the client's bench host.
- Complete suites — functional, jitter, STO fault-injection, thermal, endurance, 48 V transient — with recorded baselines for all six first articles.
- Python SDK with documentation, example scripts, C++ harness bindings, and CI integration gating every firmware candidate.
- The append-only records store as client property — the evidence base for the SIL-2 assessment.
- The independent validation report: the STO timing exceedance, the optocoupler BOM error, the interrupt-storm finding, the PHY crystal margin, metrics, pass/fail, coverage.

PRODUCT

Deliverables

- Bench procedures for carrier acceptance and re-runs; full documentation; no lock-in.

Conclusion

INDUSTRIAL AUTOMATION ROBOTICS NVIDIA JETSON

Conclusion

Six carriers were proven — and two safety-relevant defects corrected — before the motion team wrote its first trajectory. The STO chain now meets its budget at temperature with measured evidence behind it, and commissioning proceeded against hardware whose failure modes were already catalogued.

- The safety-input BOM error and the 13.7 ms STO exceedance were the kind of faults that pass a warm bench and fail a hot cell; both were caught for the price of a component change instead of an incident investigation, with assessor-ready records rather than internal assertion.

Conclusion

- The suites run on as the CI acceptance gate for every firmware drop and carry to the next carrier revision with only the HAL delta re-verified. We produce the evidence — the client's functional-safety assessor judges it.

Lessons Learned

INDUSTRIAL AUTOMATION ROBOTICS NVIDIA JETSON

Lessons Learned

The engagement pushed safety-chain timing and fault-mode testing earlier and taught us to treat sensor fault modes with the same suspicion as sensor data.

- Safety-chain reaction timing is now measured across the full temperature range in week one on any SIL-rated board — the STO exceedance existed only above 45 °C, and a room-temperature-only campaign would have certified a defect; peripheral fault modes get dedicated storm testing after the force-torque sensor's failure flooding locked a carrier.

Lessons Learned

- On Jetson-plus-STM32 topologies the discrete-line contract deserves its own suite: the drive-enable line's defined state during SoM reboot was ambiguous in the ICD, and we measured a 300 ms window where it floated — the pull-down that fixed it is now a standing design-review question, and ICD ambiguity checks moved ahead of functional testing in our pipeline.

PRODUCT

LET'S TALK

SiliconCentric

EMAIL

sandesh@siliconcentric.com

PHONE

(669) 356-1998

WEB

https://www.siliconcentric.com

ADDRESS

San Jose, CA, USA

Industrial Automation - Robotic

Industrial Automation Robotics Nvidia Jetson

Built by SiliconCentric

SiliconCentric