ENGINEERING CONSULTANCY · CAPABILITY STATEMENT

Motion, control, and precision-engineered systems — from concept to commissioned site.

SimpleMotion is a multidisciplinary engineering consultancy specialising in motion control, automation, and the electromechanical systems behind entertainment, architectural, and industrial installations.

SimpleMotion
Unit 2 / 53 Burrows Road, Alexandria NSW 2015
greg.gowans@simplemotion.com

We take projects the full distance: early feasibility and design, through fabrication and quality assurance, to on-site installation, commissioning, and handover. Mechanical, electrical, and software engineering are integrated under one roof — a single accountable partner rather than a chain of subcontractors.

What we do

- **Motion control & automation** — servo and stepper systems, multi-axis coordination, real-time control, and safety-rated motion.
- **Industrial networking & fieldbus** — EtherCAT and functional-safety communications (FSoE), with in-house protocol and firmware expertise.
- **Functional safety** — safe-motion systems designed to recognised SIL / functional-safety principles, including safe torque-off and diagnostic architectures.
- **Control software & HMI** — bespoke operator interfaces and embedded control software, built for reliability and long service life.
- **Custom hardware & mechatronics** — bespoke mechanical assemblies, control panels, and electromechanical hardware.
- **Entertainment & architectural motion** — rigging, traverse and kinetic systems, and large-format LED / video-wall installations, including mobile and touring configurations.

End-to-end delivery lifecycle

Our capability is organised around the full project lifecycle. A client can engage us for a single stage or for turnkey delivery.

Stage	What we provide
1. Client & scoping	Discovery, requirements capture, feasibility, proposals, and transparent quotation.
2. Project management	Scheduling, delivery oversight, and coordination from kick-off to close-out.
3. Engineering design	Mechanical, electrical, and software design — CAD, control architecture, and detailed engineering.
4. Supply & procurement	Component sourcing, vendor management, and supply-chain co-ordination.
5. Fabrication & manufacturing	Shop drawings, work orders, build records, and manufacturing execution.
6. Quality & safety	QA, safety reviews, and inspection workflows built into every stage.
7. Deployment & commissioning	On-site installation, commissioning, and structured handover.

Stage	What we provide
8. Knowledge & standards	A maintained corpus of technical references, regulatory standards, and vendor documentation underpinning every project.

This means quality assurance and safety are not a bolt-on at the end — they run alongside design, build, and deployment throughout.

Software engineering best practice, delivered into automation

Automation software controls physical machinery — a defect isn't a crashed app, it's a moving axis, a safety function, or a live site. SimpleMotion writes control and embedded software to the standard that reality demands, applying software-engineering rigour more often seen in aerospace and medical devices than on the plant floor.

- **Memory- and thread-safe by construction.** We build on Rust as our default control-software language. Whole classes of field failure that plague traditional C/C++ automation code — buffer overruns, use-after-free, data races between control tasks — are eliminated at compile time rather than found the hard way on site.
- **Functional safety and qualified toolchains.** For safety-related functions we develop against recognised functional-safety principles (SIL-rated design, FMEDA-based failure analysis, defined safe-state architectures) using compiler toolchains qualified for safety use — so the tools that build the software are themselves part of the safety case.
- **Deterministic, real-time by design.** Safety- and control-critical paths are written without dynamic memory allocation and with bounded, predictable execution, giving the timing determinism that motion control and safe-motion functions require.
- **Simulation- and test-first development.** We prove behaviour before it reaches hardware — using software emulation and hardware-in-the-loop testing with deliberate fault injection, backed by extensive automated test suites. Faults are exercised in the lab, not discovered during commissioning.
- **Supply-chain security and dependency hygiene.** We keep dependencies minimal (often zero) and continuously screen them against published security advisories and licence policy through automated tooling. Release artifacts are cryptographically signed with verifiable build provenance, so what runs on site is exactly what we built.
- **Automated CI/CD and reproducible builds.** Every change passes automated quality gates before it can be merged, and builds are reproducible and traceable. Software moves through controlled release channels (develop → testing → preview → release) on a build-once, promote-forward basis — the tested artifact is the deployed artifact.
- **Traceability and disciplined versioning.** Strict semantic versioning, annotated release tags, and a full change history give a clean audit trail from a deployed system back to the exact source that produced it — essential for long-lived automation assets and safety documentation.

- **Built to last, built to maintain.** Software is documented, standards-led, and portable across hardware targets, so systems remain serviceable and upgradable over the decades an automation asset stays in service.

> **The payoff for the client:** fewer field failures, safer machinery, faster and lower-risk commissioning, and control software that is auditable, secure, and maintainable for the full life of the installation.

Why SimpleMotion

- **Single accountable partner** — mechanical, electrical, and software disciplines integrated, removing interface risk between vendors.
- **Deep specialism in motion and safety** — genuine in-house expertise in real-time control, fieldbus, and functional safety, not outsourced.
- **Concept-to-commissioning** — we don't hand off at the drawing; we see systems working on site.
- **Standards-led delivery** — engineering grounded in recognised regulatory and functional-safety standards.
- **Bespoke by default** — we design to the problem, whether that's a permanent installation or a mobile, touring-grade system.

Representative work

SimpleMotion has delivered engineered motion and display systems across entertainment, retail, and public-venue environments — including large-venue traverse and rigging systems, mobile large-format LED video-wall installations, and integrated motion features for retail and architectural spaces. Detailed project references and case studies are available on request.

Engage us

We work with clients from a single design study through to complete turnkey delivery. To discuss a project or request a capability statement tailored to your sector, contact:

SimpleMotion
Unit 2 / 53 Burrows Road, Alexandria NSW 2015
greg.gowans@simplemotion.com

Appendix A — Technical stack (tender / prequalification)

The following details the engineering stack and methods underpinning our software delivery. It is provided to support technical prequalification; conformance evidence for a specific project scope is available on request.

LANGUAGES & TOOLCHAINS

Element	Detail
Primary control-software language	Rust (edition 2024), pinned toolchain.
Safety-critical toolchain	Ferrocene — a Rust toolchain qualified for safety-related development; used for safety-function code targeting qualified compilation targets.
Embedded model	`no_std` / no-heap on control- and safety-critical paths for deterministic, allocation-free execution.
HMI / application UI	Rust with Slint; WebAssembly (WASM) targets for portable interfaces.

MOTION, CONTROL & FIELDBUS

Element	Detail
Industrial fieldbus	EtherCAT.
Functional-safety communications	FSoE (Safety-over-EtherCAT) — in-house protocol and safe-slave implementation.
Safe-motion functions	Safe Torque-Off (STO) and related safe functions, dual-channel with per-cycle readback.
Real-time control	Deterministic, bounded-execution control loops with independent watchdog supervision.

FUNCTIONAL SAFETY

Element	Detail
Framework	Developed to the principles of IEC 61508 (functional safety), targeting SIL-rated integrity for safety functions.
Failure analysis	FMEDA — quantified failure-mode, effects & diagnostic analysis (SFF / PFH), diagnostic inventory.
Safety architecture	1oo1D on lockstep processing (dual-lane compare → latched safe state); hardware fault tolerance HFT-1.

Element	Detail
Memory integrity	SECDED ECC monitoring with fault-reaction to safe state.
Fault handling	Fail-safe by default; bounded fault-reaction time independent of communications watchdog.

Note: functional-safety artifacts are developed to the above standards' principles. Formal certification status for a given deliverable is stated explicitly per project — no blanket certification is claimed.

QUALITY, TEST & VERIFICATION

Element	Detail
Test strategy	Simulation-/emulator-first with deliberate fault injection; hardware-in-the-loop (HIL) verification.
Automated testing	Extensive unit/integration suites (`cargo test`), run in CI on every change.
Verification approach	Behaviour proven over the bus in emulation before hardware bring-up.

SUPPLY-CHAIN SECURITY & BUILD INTEGRITY

Element	Detail
Dependency policy	Minimal-to-zero third-party dependencies on critical components.
Vulnerability & licence screening	Automated advisory (RUSTSEC) and licence-policy enforcement via `cargo-deny` in CI.
Artifact signing	Cryptographic signing (Sigstore / cosign) with verifiable build provenance (SLSA-aligned).
Reproducibility	Reproducible, traceable builds — deployed artifact is the tested artifact.

CI/CD, VERSIONING & TRACEABILITY

Element	Detail
CI/CD	GitHub Actions — canonical reusable pipelines with mandatory PR quality gates.
Release management	Controlled channels (develop → testing → preview → release), build-once / promote-forward.
Versioning	Semantic Versioning 2.0.0; annotated release tags; full change-log history.

Element	Detail
Target platforms	Embedded ARM (e.g. Cortex-R5F lockstep for safety, application-class hosts for non-safety); desktop and web front-ends.

This is a general capability overview. A sector- or project-specific statement, including detailed references and standards conformance, is available on request.