PRODUCT

Automation Hub

Building Automation Security Automation Hub Intel X86

SiliconCentric | Product overview

SiliconCentric

Introduction

BUILDING AUTOMATION SECURITY AUTOMATION HUB INTEL X86

Introduction

Our client is a building-systems manufacturer. Their product is a building automation and security hub — when it loses power, the doors, alarms, and HVAC controllers downstream all inherit the consequences. First-article boards were back from fabrication: an Intel x86 SBC paired with a dedicated external STM32 companion owning the field buses and supervised I/O. We collaborated with them on this project by providing independent verification and validation of the board before the controls-firmware team took it over.

Introduction

- Compute topology: x86 SBC plus external STM32 companion; the x86 side runs the building-management application, database, and network services, the STM32 owns the RS-485 field buses, door I/O, and power supervision with real-time guarantees the x86 platform cannot make.
- Under validation: two RS-485 ports carrying BACnet MS/TP, one RS-485 port carrying Modbus RTU, eight supervised door-controller inputs with end-of-line resistor monitoring, four relay outputs for lock power and alarm circuits, a Wiegand reader interface pair, dual GbE, a PoE budget feeding downstream readers, an NVMe drive, a TPM, and a UPS path with battery telemetry over I2C.

Introduction

- Program state: fabricated first articles, powered for bring-up, controls firmware not yet started.
- Independence: no technical, managerial, or financial stake in the design; the manufacturer's enterprise customers audit their suppliers' evidence, and third-party findings carry further than internal ones.
- Bounded scope: independent validation only — no BMS-application work, no firmware authoring, no out-of-scope engineering.
- Working inputs: the hardware design package — schematics, ICDs, BOM — with no access to firmware source.

2

Hardware Topology

BUILDING AUTOMATION SECURITY AUTOMATION HUB INTEL X86

PRODUCT

Hardware Topology

The topology is SoC-plus-companion-uC in its x86 variant: a general-purpose SBC for the application plane, an STM32 for everything with a deadline or a safety consequence. Scope enclosed the board set, its field interfaces, and its power architecture; the BMS application, access-control policy logic, and any listing-body testing stayed outside.

- Topology classification: Intel x86 SBC plus external STM32, interconnected by USB (bulk command channel) and UART (fallback console), with a hardwired fault discrete from the STM32 that operates independently of the x86 state.

Hardware Topology

- The STM32's job: MS/TP token timing on both BACnet segments, Modbus polling, supervised-input scanning with tamper detection, relay sequencing with fail-secure/fail-safe policies, and UPS switchover supervision.
- In-scope interfaces: RS-485 x3, Wiegand x2, supervised inputs x8, relays x4, dual GbE, PoE output budget, NVMe, TPM, I2C (UPS battery, temperature), USB and UART inter-processor links.
- Envelope in scope: 0 °C to +50 °C in an equipment-closet enclosure, mains power with UPS backup, 24/7 duty measured in years, and the electrical environment of a building riser.

PRODUCT

Hardware Topology

- Out of scope: BMS software, access-control policy, UL-listing tests for alarm circuits — we produce integration evidence; the client's listing labs and enterprise auditors judge it.
- The Intel/AMD x86 SBC class is one of the six architectures Polaris already runs on, so the OS baseline was inherited and effort went to this board set's delta rather than platform bring-up.

3

Problem Statement

BUILDING AUTOMATION SECURITY AUTOMATION HUB INTEL X86

Problem Statement

The boards existed, enterprise pilot sites were being scheduled, and the controls-firmware team was about to inherit hardware whose field buses, power architecture, and supervised I/O had never been proven together. A hub that drops its MS/TP token quietly desynchronizes a floor's controllers; a hub that reboots on mains loss unlocks the wrong doors at the wrong moment.

Problem Statement

- Proven in isolation, never as a board: the SBC was a proven commercial platform and the transceivers carried vendor data, but this design's RS-485 termination and biasing across three segments, its Wiegand front ends, its supervised-input analog chains, and its UPS switchover path had never been exercised as a system.
- Firmware coverage is feature-shaped, not board-shaped: the controls team would exercise BACnet and the primary door paths daily, but the Modbus segment, the second Wiegand port, the end-of-line tamper distinctions, and the relay fail-safe policies under power fault fell outside planned features.

Problem Statement

- No stress, no extremes, no duration: no 50 °C closet soak, no multi-week 24/7 endurance with full segment traffic, no mains-loss and brown-out campaign against the UPS path, no MS/TP timing verification at maximum segment loading.
- Nothing runs alongside development: validation was an end-gate, so a switchover defect would surface during a site's first real power event — with security consequences, not just downtime.
- Enterprise security customers audit hardware evidence; alarm-circuit listing bodies expect documented supervised-input behavior; neither accepts assertion.
- MS/TP token timing under 50-node segment load is a board-and-firmware property no transceiver datasheet addresses.

Problem Statement

- A fleet of hubs in enterprise estates makes any hardware defect a contractual event — remediation terms were already written into the pilot agreements.

Customer Requirements

BUILDING AUTOMATION SECURITY AUTOMATION HUB INTEL X86

Customer Requirements

The client's requirements were direct: validate the whole board set, under stated limits, with evidence that survives enterprise security audits and listing-lab scrutiny.

- Operating temperature: the board set shall operate from 0 °C to +50 °C equipment-closet ambient, 24/7.
- Supply voltage: mains-fed 12 V DC system rail, operating window 10.8 V to 13.2 V. On mains loss the UPS path shall take over with zero gap on the 12 V rail — hold-up capacitance shall bridge the switchover with no reset of either processor.

Customer Requirements

- Brown-out: the board shall ride through 12 V rail sag to 10.8 V; below that the STM32 shall hold relay policies (fail-secure/fail-safe) and record the event with no record loss.
- Over-current: the PoE budget shall enforce per-port class limits with over-current shutdown; each relay circuit shall be protected against lock-power faults; a shorted reader shall not disturb any other port or the field buses.
- Over-voltage: the 12 V input shall survive transients to 16 V without damage; over-voltage protection shall clamp above 13.2 V continuous.
- Humidity: the board shall operate at up to 90% RH non-condensing and shall not be exposed beyond that.

Customer Requirements

- Full-board functional validation: the x86 platform's cores, memory, NVMe, TPM, and dual GbE; all three RS-485 segments; both Wiegand ports; all eight supervised inputs across their end-of-line states; all four relays; the PoE budget; the UPS path; and the STM32 loop — one integrated system.
- Coverage independent of firmware scope: every subsystem validated regardless of the controls feature set, through a HAL built from the design package.
- Stress matrix: 0–50 °C sweep, three-week 24/7 endurance at full segment load, mains-loss and brown-out campaigns across the UPS path, MS/TP timing at maximum node loading, PoE budget audit under reader worst case — vibration per equipment-closet profile.

Customer Requirements

- Continuous validation alongside development: CI integration revalidating every firmware candidate on the first articles.
- Evidence: timestamped, operator-attributed, version-pinned records with bidirectional traceability, packaged for enterprise audits and the listing-lab file.
- A web console the manufacturer's test staff run without embedded expertise.
- An immutable, append-only evidence store anchoring the pilot baseline and future revisions.
- Setup in minutes: documented bench — board set, segment simulators, programmable mains path, UPS, host — reproducible by any operator.

Architecture

BUILDING AUTOMATION SECURITY AUTOMATION HUB INTEL X86

Architecture

The x86 board runs the validation OS, HAL, and client; a separate x86 host runs the server, dashboard, and SDK — the target is never its own judge, even when both are x86. Execution stays isolated from data processing and the target keeps its resources for representative behavior, which mattered because MS/TP timing had to be measured with the application plane loaded like a real site head-end.

- Polaris OS (the validation OS) — a reproducible Linux image for the SBC booting to a known state, with the TPM in the measured-boot chain, one baseline across the first-article set.

Architecture

- HAL — the single board-specific layer, exercising all three RS-485 segments, both Wiegand ports, the supervised-input chains, relays, PoE controller, UPS telemetry, NVMe, and the STM32 link behind a uniform interface that keeps the building-systems suites portable across revisions.
- Client — the on-target agent driving tests through the HAL; gRPC on the x86 target, FlatBuffers to the STM32's Zephyr image; three timestamps per event (origin, relay, server); light enough that timing evidence reflects the site head-end, not the tooling.
- Server — orchestration, scheduling, and the append-only records store on the separate host, four board sets in parallel, dashboard and API hosted.

Architecture

- Dashboard — the console: provision a board set, configure and launch, live segment health, token-rotation timing, supervised-input state matrix, power-path telemetry, time-series review; every state-changing action operator-attributed.
- Python SDK — automation and CI; scripts everything the console does; C bindings available, unused here.
- The OS and HAL are the two board-specific layers and were heavily modified for this board set and this platform; 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

BUILDING AUTOMATION SECURITY AUTOMATION HUB INTEL X86

PRODUCT

Design

The binding constraints: MS/TP token discipline across loaded segments, supervised inputs whose four resistance states are a listing requirement, a power architecture where a gap of milliseconds is a security event, and audit customers who read records like contracts.

Design

- Validation OS design — measured boot through the TPM with a read-only rootfs and A/B rollback, because an enterprise security product's validation image must itself be attestable; the trade-off is bootloader complexity on commodity x86 firmware, worked through UEFI rather than around it. The STM32 runs Zephyr — its native UART and timer driver model fit the token-timing work.
- HAL design — partitioned by schematic sheet against ICD clauses; RS-485 events, Wiegand edges, and supervised-input transitions are interrupt-driven on the STM32 since token timing and tamper edges are the evidence, while UPS and PoE telemetry poll at 1 Hz. USB latency between x86 and STM32 makes the x86 side unfit to timestamp bus events at all — which is exactly why origin timestamps live on the microcontroller.

Design

- Client design — origin timestamps taken on the STM32 for field-bus events and correlated across the USB link; on mains events the client flushes records through the switchover window — the UPS path was itself under test — so every power transition leaves a complete record.
- Server design — three-timestamp causality orders a mains sag against the token loss it caused; retention sized for three-week 24/7 streams from four board sets, with per-token MS/TP traces retained raw during timing campaigns.

Design

- Dashboard design — segment token-rotation times, supervised-input state matrices with end-of-line classifications, relay states, and power-path telemetry are first-class views — the triage order of a building-controls test engineer; run health and drop counts on every screen.
- SDK design — the client scripted the acceptance gate, segment-load matrices, and the mains-event campaign; the direct API path served single-segment bench checks, at the stated cost that those runs produce no records.
- OS and HAL design was board-specific and heavily reworked; the four upper components were configured to this client's audit-driven workflow while their structure and working style stayed constant.

7

Implementation

BUILDING AUTOMATION SECURITY AUTOMATION HUB INTEL X86

Implementation

Work went to the OS delta, the HAL, and the building-systems campaigns; the four upper components were configured, not constructed.

- Validation OS — Yocto image for the x86 SBC: platform BSP inherited, board-set device configuration for the RS-485 complex, PoE controller, and UPS telemetry, minimal kernel with measured boot; OTA with A/B rollback; boot under 30 seconds; per-build SBOM with CVE tracking; the STM32 Zephyr image built in the same tree.

Implementation

- HAL — implemented sheet by sheet and bench-verified as written: STM32-side MS/TP token engines with hardware timestamping, Wiegand decoders, supervised-input ADC classification across the four end-of-line states, relay drivers with policy enforcement, PoE and UPS telemetry drivers; the design-package pass surfaced a schematic-to-BOM mismatch — the failsafe bias resistors on the Modbus segment appeared on the schematic but not in the BOM, leaving the segment undefined when idle, an intermittent-corruption source caught before it could pass as a protocol bug.
- Client — C++ daemon on the x86 target, event and polling paths, cross-domain timestamp correlation, power-event-hardened flushing, gRPC upstream, FlatBuffers to the STM32.

Implementation

- Server — telemetry receiver, versioned schema, scheduler, REST API, dashboard hosting, four-set parallelism.
- Dashboard — board-set provisioning, campaign configuration, live segment and power streaming, time-series review, export for audits, operator attribution.
- SDK — Python bindings, example scripts per subsystem, CI wiring firing the acceptance suite on every firmware candidate.
- Board-specific: OS delta and HAL. Configuration: client correlation, server retention, dashboard views, SDK examples.

Test – IV&V

BUILDING AUTOMATION SECURITY AUTOMATION HUB INTEL X86

PRODUCT

Test - IV&V

Testing ran stack-first, then OS/HAL, then the board set across one-shot, long-running, event-based, and monitoring campaigns — 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 it produced evidence about this board set.
- OS/HAL verification: A/B boot and rollback exercised, OTA verified end to end, measured boot verified on accept and reject paths through the TPM.

Test - IV&V

- Bench setup: board set cabled to MS/TP and Modbus segment simulators with configurable node counts, reader emulators on both Wiegand ports, programmable mains path with UPS, provisioning from the dashboard — documented, under thirty minutes.
- Functional campaign: MS/TP token and data suites on both segments, Modbus exchange suites, Wiegand decode-integrity runs, supervised-input classification across normal/alarm/tamper-short/tamper-cut states, relay policy suites, PoE class and budget runs, NVMe and TPM checks.

Test - IV&V

- Compute and memory checks: DRAM pattern stress and NVMe endurance profiling under the 24/7 duty model, CPU thermal-throttling behavior characterized at 50 °C closet ambient so head-end response stays bounded, and repeated-boot verification through the measured-boot chain.
- Watchdog and reset campaign: induced x86 hangs confirming the STM32's supervision and staged recovery while relay fail-secure/fail-safe policies held state, the STM32's own independent watchdog exercised, and warm/cold reset combinations across both processors verified to rejoin the field buses without token disruption.

Test - IV&V

- Debug posture: JTAG/SWD on the STM32 and the x86 debug console verified disabled in the deployment configuration — an open debug path on an access-control product is an audit finding in its own right.
- One-shot tests as the acceptance gate: one pass/fail per subsystem per firmware candidate.
- Long-running: the three-week 24/7 endurance at 50-node segment load caught token-rotation time degrading on segment A above 45 °C — a transceiver bias-network temperature sensitivity corrected with component values in the revision.

Test - IV&V

- Event-based: silent runs recording on token losses, input-state transitions, relay changes, and power events — this mode caught one supervised-input channel classifying a tamper-short as normal within a 200 Ω window, a divider tolerance stack that a listing lab would have failed, corrected in the revision.
- Monitoring: passive telemetry — rails, closet-simulated temperature, PoE draw, UPS battery state — through mains-event campaigns; the main finding surfaced here: mains loss produced a 12 ms gap on the 12 V rail before the UPS path took over, hard-rebooting the x86 side on every power event — the hold-up capacitance was calculated for the previous board revision's lower draw and never rechecked, fixed with added bulk capacitance and a switchover-controller change.

Test - IV&V

- Instrumentation correlation: oscilloscope captures anchored the switchover gap, the bias-network drift, and the supervised-input windows; a relay fail-secure policy was verified to hold state through the (now-fixed) switchover by direct contact monitoring.
- 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.
- Traceability: requirements traced bidirectionally to tests and records, schemas version-pinned, runs reproducible for auditors and the listing lab.

9

Deliverables

BUILDING AUTOMATION SECURITY AUTOMATION HUB INTEL X86

PRODUCT

Deliverables

The manufacturer received the complete platform fitted to this board set — source, records, procedures, no lock-in.

- Validation OS image with the full Yocto build system, measured-boot recipes, and per-build SBOM.
- Board-set-specific BSP and HAL source for every subsystem, maintained against the design package into the revision.
- Compiled client for the x86 target plus the STM32 Zephyr build.

Deliverables

- Server and dashboard as deployment-ready containers on the manufacturer's bench host.
- Complete suites — MS/TP timing, Modbus, Wiegand, supervised-input classification, relay policy, PoE audit, mains-event, endurance — with recorded first-article baselines.
- Python SDK with documentation, example scripts, and CI integration gating every firmware candidate.
- The append-only records store as client property — the evidence base for enterprise audits and the listing file.
- The independent validation report: the UPS switchover gap, the tamper-classification window, the Modbus bias BOM gap, the token-timing drift, metrics, pass/fail, coverage.

PRODUCT

Deliverables

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

Conclusion

BUILDING AUTOMATION SECURITY AUTOMATION HUB INTEL X86

Conclusion

A security hub that rebooted on every mains loss never reached a single site: the switchover gap, the tamper window that read as normal, and the undefined Modbus segment were all corrected in one revision, before the controls team started and before any pilot agreement's remediation clause could be invoked.

- The 12 ms switchover gap was the defect with security consequences — every building power event would have rebooted access control estate-wide; it cost capacitors and a controller change instead of pilot-site incidents, with scope-anchored records enterprise auditors accepted directly.

Conclusion

- The suites gate every firmware drop in CI and carry to the revision with only the HAL delta re-verified. We produce the evidence — the client's listing labs and enterprise auditors judge it.

Lessons Learned

BUILDING AUTOMATION SECURITY AUTOMATION HUB INTEL X86

Lessons Learned

The engagement added power-transition and inherited-assumption checks to the front of our building-systems pipeline.

- Hold-up and switchover arithmetic gets re-derived from the current BOM on every engagement, never inherited — this board's 12 ms gap existed because a previous revision's calculation survived a redesign that doubled the platform draw; supervised-input classification likewise now gets swept across its full resistance range, because the tamper window lived between the test points a nominal check would use.

Lessons Learned

- On x86-plus-microcontroller topologies the timestamp domain must be settled before any timing claim: USB latency between the SBC and STM32 varied from 1 to 9 ms under x86 load, making microcontroller-side origin timestamps the only defensible source for field-bus evidence — a constraint we now design into every x86-companion engagement from day one.

PRODUCT

LET'S TALK

SiliconCentric

EMAIL

sandesh@siliconcentric.com

PHONE

(669) 356-1998

WEB

https://www.siliconcentric.com

ADDRESS

San Jose, CA, USA

Automation Hub

Building Automation Security Automation Hub Intel X86

Built by SiliconCentric

SiliconCentric