PRODUCT

# Smart Home Controller

Consumer Electronics Smart Home Controller Nxp Imx

SiliconCentric | Independent Verification & Validation

SiliconCentric

I

PRODUCT

# Introduction

Our client is a consumer-electronics manufacturer. Their product is a wall-mounted smart home controller, mid-way through hardware development — prototype boards on the bench, a rev B layout still open. The design is a standalone NXP i.MX SoC — a heterogeneous part whose internal Cortex-M7 acts as the companion core to the Cortex-A application cluster, with no second package on the board. We collaborated with them on this project by providing independent verification and validation of the board before rev B froze.

- Compute topology: a single i.MX package; Cortex-A cores run the product UI and connectivity stack, the internal Cortex-M7 handles the always-on sensor loop and wake logic.
- Hardware under validation: a Zigbee/Thread radio module on SPI, a Wi-Fi/BLE combo module on SDIO, a 5-inch MIPI-DSI panel with I2C capacitive touch, a four-element PDM microphone array, an I2S audio codec, eMMC and LPDDR4, USB-C, and temperature, humidity, and ambient-light sensors.
- The client was mid-development: rev A prototypes in bring-up, known layout escapes, a rev B spin already budgeted.
- Independence was the point — we hold no technical, managerial, or financial stake in the design, so a finding against the antenna placement costs us nothing to report.
- Scope stayed bounded: independent validation only, no feature development, no product firmware 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

PRODUCT

# Hardware Topology

In engineering terms this is a standalone heterogeneous SoC: one i.MX device, with the A-cluster/M7 split providing the application/companion division that other products implement with two packages. The scope line ran around the board and its interfaces; the client's Matter application stack and any certification sign-off sat outside it.

- Topology classification: standalone SoC with an internal companion core; A-cores and M7 communicate over RPMsg through shared OCRAM, with a hardware mailbox for doorbell signaling.
- The M7's job: poll the environmental sensors, run wake-word pre-detection on the microphone stream, and wake the A-cluster from suspend — the loop that defines the product's idle power.
- In-scope buses: SPI (Zigbee), SDIO (Wi-Fi/BLE), MIPI-DSI, I2C x3 (touch, codec control, sensors), I2S, PDM, USB 2.0 over USB-C, eMMC at HS200.
- Operating envelope: 0 °C to +45 °C ambient in a sealed plastic enclosure, single 5 V input, always-on duty cycle with deep-sleep targets in the tens of milliwatts.
- Out of scope: the Matter/Thread application stack, UI firmware, FCC/CE filings — we generate pre-compliance and integration evidence; the labs and assessors judge it.
- The i.MX family is among the six architectures Polaris already runs on, so the OS baseline was inherited and the engagement's effort went to this board's delta, not silicon bring-up.

3

PRODUCT

# Problem Statement

Rev A boards were half-alive on the bench, the connectivity firmware team was staffed and waiting, and a retail launch window anchored the schedule. Nothing independent proved that the radios, display, audio path, and power tree worked as one board — and in the smart-home market, a controller that drops its Zigbee network is a one-star review at scale.

- Proven in isolation, never as a board: the radio modules carry their own module-level certifications and the panel worked on the vendor's demo kit, but this board's antenna placement, SDIO routing, DSI lane geometry, and shared 3.3 V rail had never been exercised together.
- Firmware coverage is feature-shaped, not board-shaped: launch firmware needed one microphone pair, not four, and never touched USB-C data or the third I2C bus — so half the microphone array, the USB path, and the sensor bus would go to production untested by the feature team.
- No stress, no extremes, no duration: no soak at 45 °C inside the sealed enclosure, no multi-day radio-traffic endurance, no fast power-cycle campaign against the flaky wall power this product lives on, no eMMC wear characterization for an always-logging device.
- Nothing runs alongside development: with validation planned as a pre-launch gate, a coexistence problem would surface as "Zigbee is unreliable" in the firmware backlog — a hardware fault filed as a software bug.
- FCC Part 15 and CE EMC testing loomed, and a failed chamber visit mid-schedule costs weeks per iteration.
- Both radios and the product's Thread network share 2.4 GHz on a board 90 mm wide — the coexistence risk was structural, not hypothetical.
- eMMC at HS200 over a routing that had already required one rev A bodge wire needed margin data, not optimism.

PRODUCT

# Customer Requirements

4

The client's requirements were direct: validate the whole board, under stated limits, with evidence usable both for the rev B decision and for the EMC lab visits.

- Operating temperature: the board shall operate from 0 °C to +45 °C ambient inside the sealed enclosure, always-on.
- Supply voltage: single 5 V input, operating window 4.75 V to 5.25 V. The board shall survive input transients to 6 V without damage; over-voltage protection shall clamp above 5.5 V.
- Brown-out: the board shall ride through input sag to 3.8 V and recover without operator intervention, including power restoration within 80 ms of loss; no eMMC corruption on any power event.
- Over-current: the USB-C port and the 3.3 V rail shall have over-current protection tripping within 10 ms; a shorted accessory shall not disturb the radios or the display.
- Sleep power: deep-sleep draw shall stay within the tens-of-milliwatts design target with the M7 loop running.
- Humidity: the board shall operate at up to 85% RH non-condensing and shall not be exposed beyond that.
- Full-board functional validation: all four A-cores, the M7, LPDDR4 regions, eMMC, both radios, DSI and touch, all four PDM microphones, the codec, USB-C, and every sensor, exercised as an integrated system.
- Coverage independent of firmware scope: every subsystem validated regardless of the launch feature set, through a HAL built from the design package rather than from the product firmware.
- Stress matrix: 0–45 °C thermal sweep in the enclosure, 96-hour endurance under simultaneous Wi-Fi and Zigbee traffic, 10,000 fast power cycles, brown-out to 3.8 V on the 5 V input, and radio coexistence characterization under load — vibration waived by product profile.
- Continuous validation alongside development: CI hooks so every firmware candidate re-runs the acceptance suite on real rev A and later rev B hardware.
- Evidence: timestamped, operator-attributed, version-pinned records with requirement-to-result traceability, packaged to support the EMC pre-scan narrative and the client's retail partners' quality audits.
- A web console the client's junior test technician can run without knowing the validation stack, learned once.
- An immutable, append-only evidence store carrying rev A findings forward as the baseline for rev B.
- Bench setup in minutes: boxed board to running campaign without a custom rig.

5

PRODUCT

# Architecture

The split places the validation OS, HAL, and client on the i.MX target while an external x86 host carries the server, dashboard, and SDK. Execution stays isolated from data processing, and the target keeps its cycles for representative behavior — which mattered here because idle-power measurements are exactly the thing a heavy on-target toolchain would destroy.

- Polaris OS (the validation OS) — a minimal, reproducible Linux image for this i.MX board, giving the client a known-good boot state independent of their in-flux product image.
- HAL — the single board-specific layer; on this board it drives SPI, SDIO, DSI, the three I2C buses, I2S, PDM, USB, and the RPMsg link to the M7, and its uniform interface means the consumer-device suites transfer intact to rev B.
- Client — the on-target agent driving tests through the HAL; gRPC on Linux, FlatBuffers to the FreeRTOS image on the M7; three timestamps per event (origin, relay, server); its footprint is small enough that sleep-current measurements remained valid.
- Server — orchestration, scheduling, and the append-only records store on the host; six rev A prototypes ran in parallel from one server, which also hosts the dashboard and API.
- Dashboard — the operator console: provision a board, configure, launch, watch live RSSI, PER, rail, and temperature telemetry, review time series afterward; every state-changing action is operator-attributed.
- Python SDK — automation and CI; everything the console does is scriptable; C/C++ bindings exist but went unused on this program.
- The OS and HAL are the two board-specific layers, heavily modified for this board and this SoC; the client, server, dashboard, and Python SDK were configured to this client's requirements, but their core structure and working style remain the same across every board and every industry.

6

PRODUCT

# Design

Four constraints bound the design: a sealed enclosure with no fan, an always-on duty cycle with milliwatt sleep targets, two radios sharing one band on a small board, and a consumer BOM that punishes every added component.

- Validation OS design — A/B partitions with rollback and a read-only rootfs, since a corrupted image on a board sealed in its thermal enclosure wastes a soak cycle; the cost is heavier update mechanics, accepted. The M7 runs FreeRTOS rather than Zephyr because the wake-loop timing model was simpler to verify there.
- HAL design — partitioned per schematic sheet with each function pinned to an ICD clause; the radios use event-driven capture while the slow sensors are polled — an explicit split, because interrupt-driving a 1 Hz humidity sensor adds complexity for nothing, and polling a radio hides the exact PER transients under test.
- Client design — origin timestamps taken at the driver boundary so coexistence events order correctly against traffic bursts; on power-fail the client abandons in-flight aggregation and saves raw records, trading post-processing effort for zero loss on a product that will see dirty wall power.
- Server design — three-timestamp ordering separates a Wi-Fi burst from the Zigbee PER spike it caused; retention was sized for 96-hour dual-radio runs across six boards, and coexistence records were kept raw rather than pre-aggregated — bulkier, but rev B comparisons needed the raw data.
- Dashboard design — PER, RSSI, rail voltage, and enclosure temperature are the first-class views, because a consumer-device operator triages radio health before anything else; drop counts and run health header every view.
- SDK design — the client scripted the nightly acceptance gate and the coexistence matrix; the direct API path handled quick single-radio checks at the bench, with the known trade-off that those checks produce no evidence records.
- OS and HAL design was board-specific and heavily reworked; the upper four components were configured to this client's workflow while their structure and working style stayed constant.

7

PRODUCT

# Implementation

Engineering hours went into the two board-specific layers and the consumer-electronics campaigns; the client, server, dashboard, and SDK were configuration exercises on day one.

- Validation OS — Yocto image for this i.MX board: inherited BSP and U-Boot, board-specific device tree covering the DSI panel, both radio modules, and the PDM interface; kernel trimmed to validated drivers; OTA with A/B rollback; sub-25-second boot; per-build SBOM with CVE tracking; the M7 FreeRTOS image built under the same tree.
- HAL — written subsystem by subsystem from the design package, each function bench-verified as it landed: SDIO bus-timing work for the Wi-Fi module, DMA capture paths for PDM audio, register-level control of the codec, RPMsg transport to the M7; this pass exposed a net-name drift finding — the schematic's 'SENS_I2C' bus mapped to a different SoC pad than the ICD claimed, a documentation error that would have read as a dead sensor bus during firmware bring-up.
- Client — C++ daemon, event and polling paths, hardware timestamping, graceful shutdown preserving records, gRPC upstream, FlatBuffers to the M7.
- Server — telemetry receiver, versioned schema, scheduler, REST API, dashboard hosting, six-board parallelism.
- Dashboard — board provisioning, campaign configuration, live radio and thermal streaming, time-series review, CSV export for the EMC lab pre-brief, operator attribution throughout.
- SDK — Python bindings, per-peripheral example scripts, and CI wiring that runs the acceptance suite against every firmware candidate.
- Board-specific: OS delta and HAL. Configuration: client transports, server retention, dashboard views, SDK examples and CI glue.

PRODUCT

# Test - IV&V

Verification ran inside-out: prove the stack, prove the OS/HAL on this board, then run the board itself through one-shot, long-running, event-based, and monitoring campaigns, all dashboard-launched work landing in the append-only store.

- Stack self-verification: 100% test coverage of the client, server, dashboard, and Python SDK before any of it was used as evidence about the client's board.
- OS/HAL verification: A/B boot and rollback cycled, OTA verified end to end, secure boot exercised on accept and reject paths.
- Bench setup: board cabled to the host, programmable 5 V supply and enclosure thermocouple configured, provisioning from the dashboard — repeatable in under twenty minutes.
- Functional campaign: Zigbee PER sweeps, Wi-Fi throughput under interference, DSI pattern and touch-report suites, four-microphone beam checks, codec loopback, USB-C enumeration, eMMC HS200 margin runs, sensor-bus verification.
- A-cluster and memory campaign: LPDDR4 pattern stress across the thermal range, cache and DMA integrity under simultaneous radio and display load, and DVFS transitions with thermal-throttling behavior characterized in the sealed enclosure — a fanless always-on product lives or dies on its throttle curve.
- Low-power state campaign: every deep-sleep entry and exit path exercised against its wake sources — touch, wake-word pre-detection, sensor thresholds, USB attach — with M7 loop timing verified across A-cluster suspend/resume and residual sleep current attributed rail by rail.
- Watchdog and reset campaign: induced hangs on the A-cluster and the M7 confirming watchdog recovery to a fully restored state — radios reassociated, wake loop running — because a wall-mounted controller nobody reboots must always recover itself; the supervisor-reset fix for the PMIC brown-out hang was regression-verified through the same fast power-cycle matrix that found it.
- Debug posture: JTAG access verified fused off in the production-intent configuration, retained on rev A where trace anchored the wake-timing measurements.
- One-shot tests formed the acceptance gate — one pass/fail per subsystem per firmware candidate.
- Long-running: the 96-hour dual-radio soak surfaced the main defect — Zigbee PER climbing from 0.7% to 9% whenever Wi-Fi sustained high throughput, traced to antenna separation and a missing shield-can contact, fixed in rev B.
- Event-based: silent runs emitting records on PER thresholds, rail excursions, thermal crossings, and eMMC CRC errors; this mode caught HS200 CRC retries above 42 °C, and the link was derated with adjusted drive strength after margin analysis.
- Monitoring: passive telemetry — rails, SoC and enclosure temperature, CPU load, sleep current — alongside a representative always-on workload.
- Instrumentation correlation: spectrum-analyzer and oscilloscope captures anchored to records for the coexistence and eMMC findings; a fast power-cycle campaign also showed the PMIC occasionally hanging boot when power returned within 80 ms of loss — a brown-out recovery hole now handled in rev B with a supervisor reset.
- 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, at 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.

9

PRODUCT

# Deliverables

Delivery was the full validation platform fitted to this board — source included, records owned by the client, nothing locked in.

- Validation OS image with the complete Yocto build system, recipes, and per-build SBOM.
- Board-specific BSP and HAL source for every peripheral, maintained against the design package through the rev B spin.
- Compiled client for the A-cores plus the M7 FreeRTOS build.
- Server and dashboard as deployment-ready containers on the client's bench host.
- The full suites — functional, coexistence, thermal, endurance, power-cycle, eMMC margin — with recorded rev A baselines for rev B comparison.
- Python SDK with documentation, example scripts, and the CI integration gating every firmware candidate.
- The append-only records store as client property — their evidence for retail audits and EMC pre-briefs.
- The independent validation report: coexistence defect, eMMC margin finding, PMIC brown-out hole, net-name drift, metrics, pass/fail, coverage.
- Bench acceptance procedures, full documentation, no lock-in.

10

PRODUCT

# Conclusion

Rev B froze with the coexistence fix, the eMMC derating, and the PMIC supervisor already designed in — each defect caught on the bench while a layout change still cost days instead of a post-launch field program.

- Four hardware findings — radio desense, HS200 margin, the brown-out hang, and the sensor-bus documentation error — were closed before the connectivity team wrote against the board, converting months of misattributed firmware debugging into one planned spin.
- The suites and records now run as the CI acceptance gate and carry directly into rev B with only the HAL delta re-verified. The caveat stands — we produce the evidence; the client's labs and assessors judge it.

II

PRODUCT

# Lessons Learned

The engagement moved radio coexistence and power-quality testing earlier in our consumer-device pipeline.

- Coexistence and dirty-power campaigns now run in week one on any two-radio consumer board — both main defects here were findable before functional suites, and waiting for the soak test to expose them cost the schedule two weeks.
- On heterogeneous i.MX parts the A-core/M7 boot-order contract needs its own suite: RPMsg came up inconsistently when the M7 booted from OCRAM before the A-cluster clocked it, a shared-resource dependency the reference manual describes only in passing.

CONTACT

# SiliconCentric

| | |
|---|---|
| EMAIL | sandesh@siliconcentric.com |
| PHONE | (669) 356-1998 |
| WEB | https://www.siliconcentric.com |
| ADDRESS | San Jose, CA, USA |