PRODUCT

# Precision Irrigation

## Agriculture Precision Irrigation Controller Raspberry Pi

SiliconCentric | Product overview

SiliconCentric

# I

# Introduction

AGRICULTURE PRECISION IRRIGATION CONTROLLER RASPBERRY PI

PRODUCT

# Introduction

Our client is an agricultural-technology company. Their product is a solar-powered precision irrigation controller designed to run unattended at the far end of a farm, where nobody can reboot it. The board pairs a Raspberry Pi CM4 on a custom carrier — scheduling, analytics, connectivity — with a dedicated Raspberry Pi Pico W companion that owns the valve drivers and the sensing loop. The client wanted the first-article boards independently proven before deploying a pilot fleet. We collaborated with them on this project by providing independent verification and validation of the board.

PRODUCT

# Introduction

- Compute topology: CM4 plus external Pico W companion; the CM4 plans irrigation, the Pico W executes it — valve actuation, flow counting, and moisture sampling continue even when the CM4 sleeps to save power.
- Under validation: sixteen high-side solenoid driver channels with flyback protection for latching DC valves, four pulse-counting flow-meter inputs, eight soil-moisture ADC channels behind a multiplexer, a LoRa radio module on SPI, a solar input with MPPT charge controller and 12 V battery, a 4G modem on USB, eMMC on the CM4, and an IP-rated enclosure whose thermal behavior was part of the envelope.
- Program state: fabricated first articles on the bench, a pilot deployment across three farms scheduled for the coming season.

# Introduction

- Independence: no technical, managerial, or financial stake in the design — a finding against the power architecture delays a pilot, not our invoice.
- Bounded scope: independent validation only — no scheduling-algorithm 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

AGRICULTURE PRECISION IRRIGATION CONTROLLER RASPBERRY PI

PRODUCT

# Hardware Topology

The topology is SoC-plus-companion-uC with a power motive: the Pico W runs continuously on milliwatts while the CM4 duty-cycles. Scope enclosed the carrier, the actuation and sensing chains, and the energy system; irrigation agronomy and the cloud backend stayed outside.

- Topology classification: CM4 plus external Pico W, interconnected by UART for command traffic and one GPIO wake line from the Pico W to the CM4's power controller.

PRODUCT

# Hardware Topology

- The Pico W's job: valve sequencing with per-channel current confirmation, flow-pulse counting, moisture-mux scanning, battery supervision, and waking the CM4 on schedule or alarm.
- In-scope interfaces: UART (inter-processor), SPI (LoRa), USB (4G modem), I2C (charge controller telemetry, temperature/humidity), sixteen driver channels, four pulse inputs, eight muxed ADC channels, eMMC.
- Envelope in scope: −10 °C to +60 °C inside the sealed IP-rated enclosure in direct sun, solar input from 0 to 150 W, battery autonomy through multi-day overcast, and season-long unattended duty.

# Hardware Topology

- Out of scope: irrigation scheduling logic, cloud integration, and any agricultural-equipment conformity sign-off — we produce the evidence; the client and their pilot partners judge it.
- The Raspberry Pi family is one of the six architectures Polaris already runs on, so the OS baseline was inherited and effort went to the carrier delta rather than silicon bring-up.

3

# Problem Statement

AGRICULTURE PRECISION IRRIGATION CONTROLLER RASPBERRY PI

PRODUCT

# Problem Statement

The boards existed, the pilot season had a hard planting-calendar deadline, and a controller that fails in a field fails invisibly — crops under-watered for weeks before anyone drives out. Every part had maker-community or vendor pedigree; the carrier tying them together had none.

- Proven in isolation, never as a board: the CM4 is proven silicon and the Pico W a known quantity, but this carrier's solenoid drivers, its muxed analog front end, its MPPT charge path, and its UART/wake topology had never been exercised together — and carriers, not modules, are where such designs fail.

# Problem Statement

- Firmware coverage is feature-shaped, not board-shaped: the pilot firmware used eight of sixteen valve channels and two of four flow inputs, treated the 4G modem as primary with LoRa as fallback, and never exercised the battery's deep-discharge lockout — the unused half of the actuation hardware and the worst-case energy paths would reach the field untested.
- No stress, no extremes, no duration: no 60 °C enclosure soak in simulated sun, no multi-week duty-cycle endurance, no brown-out campaign against cloudy-week battery profiles, no valve-load switching at temperature — the actual life of solar field hardware.
- Nothing runs alongside development: validation was an end-gate, so a power-path defect would surface mid-season as dead controllers on three farms, debugged by truck roll.

# Problem Statement

- The client's pilot agreements specified uptime commitments to the partner farms; missed irrigation windows carry crop-loss liability.
- Sixteen inductive loads switched by a carrier sharing ground with precision ADC channels is a coupling risk owned entirely by this layout.
- eMMC on a solar power budget meets brown-out more often than most embedded storage — unclean power-loss behavior needed proof, not assumption.

4

# Customer Requirements

AGRICULTURE PRECISION IRRIGATION CONTROLLER RASPBERRY PI

# Customer Requirements

The client's requirements were direct: validate the whole board, under stated limits, with evidence supporting the pilot go/no-go decision and the partner-farm uptime commitments.

- Operating temperature: the board shall operate from −10 °C to +60 °C enclosure-internal, including direct-sun soak.

4

# Customer Requirements

- Supply voltage: 12 V battery system, operating window 10.5 V to 14.6 V; solar input 0 to 150 W. The board shall survive solar open-circuit voltage up to 24 V without damage; over-voltage protection on the charge path shall clamp above 15 V on the battery bus.
- Brown-out and deep discharge: below 10.5 V the board shall shut down cleanly with no record loss and no filesystem corruption; the deep-discharge lockout shall disconnect the load at 10.0 V and re-enable only above 12.0 V.
- Over-current: each of the sixteen valve driver channels shall have per-channel over-current protection tripping at 2 A within 10 ms; a single shorted valve shall not disturb any other channel or reset the CM4.

# Customer Requirements

- Humidity: the enclosure interior shall not exceed 90% RH non-condensing; the sealed IP-rated enclosure carries the external exposure.
- Full-board functional validation: the CM4's cores and memory, eMMC, all sixteen driver channels with current confirmation, all four flow inputs, all eight moisture channels, LoRa, the 4G modem path, the charge controller, and the Pico W loop — one integrated system.
- Coverage independent of firmware scope: every subsystem validated regardless of the pilot feature set, through a HAL built from the design package.

# Customer Requirements

- Stress matrix: −10 °C to +60 °C enclosure sweep, three-week duty-cycle endurance, solar-profile replay including five-day overcast, brown-out and deep-discharge campaigns, valve switching at thermal extremes, and unclean-power-loss storage testing — vibration waived, ingress verified by enclosure rating.
- Continuous validation alongside development: CI integration revalidating every firmware candidate on the first articles through the pilot.
- Evidence: timestamped, operator-attributed, version-pinned records with bidirectional traceability, packaged for the pilot review and partner reporting.

# Customer Requirements

- A web console the client's two-person hardware group can run without validation-stack expertise.
- An immutable, append-only evidence store anchoring the pilot baseline and the next carrier revision.
- Setup in minutes: a documented bench — carrier, valve load bank, solar simulator, host — reproducible by either engineer.

5

# Architecture

AGRICULTURE PRECISION IRRIGATION CONTROLLER RASPBERRY PI

# Architecture

The carrier runs the validation OS, HAL, and client; an external x86 host runs the server, dashboard, and SDK. Execution stays isolated from data processing and the carrier keeps its resources for representative behavior — which here meant energy measurements reflected the real duty cycle, since the tooling's own draw would otherwise dominate a milliwatt budget.

- Polaris OS (the validation OS) — a reproducible Linux image for the CM4 booting to a known state, one baseline across the first-article set.

# Architecture

- HAL — the single board-specific layer, exercising the driver bank, flow inputs, moisture mux, LoRa, modem path, charge-controller telemetry, and the Pico W link behind a uniform interface that keeps the agricultural suites portable to the next carrier spin.
- Client — the on-target agent driving tests through the HAL; gRPC on the CM4, FlatBuffers to the Pico W's FreeRTOS image; three timestamps per event (origin, relay, server); sleep-aware so duty-cycle energy evidence stays accurate.
- Server — orchestration, scheduling, and the append-only records store on the host, four carriers in parallel, dashboard and API hosted.

# Architecture

- Dashboard — the console: provision a carrier, configure and launch, live valve-current, flow, moisture, battery, and solar telemetry, time-series review; every state-changing action operator-attributed.
- Python SDK — automation and CI; scripts everything the console does; C bindings available, unused by this team.
- The OS and HAL are the two board-specific layers and were heavily modified for this carrier and this silicon; 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

AGRICULTURE PRECISION IRRIGATION CONTROLLER RASPBERRY PI

# Design

The binding constraints: a solar energy budget measured in watt-hours per day, sixteen inductive loads beside precision analog, a sealed enclosure in direct sun, and a season of unattended operation compressed into three weeks of evidence.

- Validation OS design — read-only rootfs with A/B rollback because eMMC on brown-out-prone power must never depend on a writable root; the trade-off is overlay complexity for test artifacts, accepted. The Pico W runs FreeRTOS — its dual-core M0+ split cleanly into a valve/flow core and a communications core.

# Design

- HAL design — partitioned by schematic sheet against ICD clauses; flow pulses and valve-current confirmations are interrupt-driven since polling would miss pulses at high flow rates, while the moisture mux is polled in a fixed scan. Mux settling time is the known risk — scanning too fast reads the previous channel's residue — so settling was characterized and enforced in the driver.
- Client design — origin timestamps at the driver boundary; across CM4 sleep windows the Pico W buffers events and back-fills on wake with relay timestamps marking the gap — the trade-off is coarser ordering during sleep, recorded explicitly rather than hidden.

PRODUCT

# Design

- Server design — three-timestamp causality orders a cloud-bank battery sag against the modem dropout it caused; retention sized for three-week duty-cycle streams from four carriers; solar and battery telemetry kept raw because the energy model was itself under validation.
- Dashboard design — battery state, solar harvest, per-valve current signatures, and moisture-channel health are first-class views — what a two-person hardware team actually triages; run health and drop counts on every screen.
- SDK design — the client scripted the acceptance gate, solar-profile replays, and the valve endurance matrix; the direct API path served single-valve bench checks, at the stated cost that those runs produce 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

AGRICULTURE PRECISION IRRIGATION CONTROLLER RASPBERRY PI

PRODUCT

# Implementation

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

- Validation OS — Yocto image for the CM4: inherited BSP and bootloader, carrier device tree covering the driver bank, mux, LoRa, and charge controller, minimal kernel with power-management states validated; OTA with A/B rollback; boot under 25 seconds; per-build SBOM with CVE tracking; the Pico W FreeRTOS image built in the same tree.

# Implementation

- HAL — implemented sheet by sheet and bench-verified as written: GPIO sequencing with current-confirmation ADC reads for the driver bank, edge-counting with debounce for flow inputs, mux scan control with enforced settling, SPI timing for LoRa, charge-controller register drivers; the design-package pass surfaced a net-name drift finding — the schematic's 'VALVE_FB_7' net landed on the ADC input labeled channel 8 in the ICD, an off-by-one that would have confirmed the wrong valve for two channels.
- Client — C++ daemon, event and polling paths, sleep-aware buffering with the Pico W, graceful shutdown without record loss, gRPC upstream, FlatBuffers to the Pico W.
- Server — telemetry receiver, versioned schema, scheduler, REST API, dashboard hosting, four-carrier parallelism.

# Implementation

- Dashboard — carrier provisioning, campaign configuration, live energy and actuation streaming, time-series review, export for pilot reporting, 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 sleep policy, server retention, dashboard views, SDK examples.

8

# Test - IV&V

AGRICULTURE PRECISION IRRIGATION CONTROLLER RASPBERRY PI

# Test - IV&V

Testing ran stack-first, then OS/HAL, then the carrier 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 carrier.
- OS/HAL verification: A/B boot and rollback exercised, OTA verified end to end, boot integrity checked on accept and reject paths.

# Test - IV&V

- Bench setup: carrier wired to a sixteen-valve load bank, flow simulators, a programmable solar source, and the host — documented, repeatable, under thirty minutes.
- Functional campaign: per-channel valve actuation with current-signature verification, flow-count accuracy runs, moisture-channel accuracy against reference resistive loads, LoRa link and range-proxy suites, modem attach and failover, charge-controller telemetry, eMMC integrity.

# Test - IV&V

- CM4 compute and memory checks: DRAM pattern stress and DMA transfer integrity under enclosure soak, CPU thermal-throttling behavior characterized at 60 °C so workloads degrade predictably rather than stall, and boot-path verification from eMMC across repeated power cycles; SWD access on the Pico W was retained for bench debug and confirmed absent from the field-facing connector set.
- Watchdog and reset campaign: the Pico W's independent watchdog and the CM4 supervisor exercised with induced firmware hangs — valves must fall to a de-energized safe state and the system must self-recover, since nobody reboots a controller at the far end of a farm; brown-out resets replayed across both processors to confirm clean rejoin of the inter-processor link.

# Test – IV&V

- Inter-processor link robustness: the UART protocol's framing, CRC, sequence numbering, and ACK/retry behavior fuzzed with injected corruption and truncated frames, heartbeat-timeout and link-loss recovery measured, and the Pico W firmware-update-over-link path verified against mid-transfer power loss.
- One-shot tests as the acceptance gate: one pass/fail per subsystem per firmware candidate.
- Long-running: the three-week duty-cycle endurance replayed real irrigation schedules, catching moisture channel 5's readings drifting when the enclosure exceeded 50 °C — a thermally sensitive divider corrected with a component change.

# Test - IV&V

- Event-based: silent runs recording on valve-confirmation mismatches, flow anomalies, battery thresholds, and resets — this mode caught the main defect on day two: the CM4 rebooted every time a valve closed, traced to an underrated flyback clamp letting a 60 V inductive spike onto the 12 V bus, corrected with proper clamp diodes and a bulk-capacitance note.
- Monitoring: passive telemetry — battery voltage and current, solar harvest, enclosure and board temperature, sleep currents — under solar-profile replays; the five-day overcast profile exposed a second energy finding: the carrier back-fed roughly 4 mA into the dark solar panel through an incomplete blocking path, a slow leak the energy model had not budgeted, cutting overcast autonomy by more than a day.

# Test - IV&V

- Instrumentation correlation: oscilloscope captures anchored the flyback spike and back-feed findings, and unclean-power-loss testing corrupted one eMMC filesystem in 200 pulls until the journal configuration was corrected — verified with 500 clean recoveries after.
- 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.

PRODUCT

# Test - IV&V

- Traceability: requirements traced bidirectionally to tests and records, schemas version-pinned, runs reproducible.

9

# Deliverables

AGRICULTURE PRECISION IRRIGATION CONTROLLER RASPBERRY PI

# Deliverables

The client received the complete platform fitted to this carrier — source, records, procedures, no lock-in.

- Validation OS image with the full Yocto build system, recipes, and per-build SBOM.
- Carrier-specific BSP and HAL source for every subsystem, maintained against the design package into the next spin.
- Compiled client for the CM4 plus the Pico W FreeRTOS build.
- Server and dashboard as deployment-ready containers on the client's bench host.

# Deliverables

- Complete suites — actuation, flow, moisture accuracy, energy-profile replay, endurance, power-loss — 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 pilot go/no-go evidence base.
- The independent validation report: the flyback defect, the solar back-feed, the thermal divider drift, the net-name off-by-one, the eMMC journal finding, metrics, pass/fail, coverage.
- Bench procedures for acceptance and re-runs; full documentation; no lock-in.

10

# Conclusion

AGRICULTURE PRECISION IRRIGATION CONTROLLER RASPBERRY PI

# Conclusion

A controller meant to survive a season alone in a field went to its pilot with its reset-on-valve-close defect, its silent energy leak, and its wrong-channel feedback wiring already found and fixed — on a bench, before three farms depended on it.

- The flyback clamp alone would have ended the pilot in week one; caught on day two of event-based testing, it cost diodes and a layout note instead of dead controllers, truck rolls, and crop-loss claims — with records behind every finding.

# Conclusion

- The suites gate every firmware drop in CI and carry to the next carrier revision with only the HAL delta re-verified. We produce the evidence — the client and their pilot partners judge it.

# II

# Lessons Learned

AGRICULTURE PRECISION IRRIGATION CONTROLLER RASPBERRY PI

# Lessons Learned

The engagement changed our defaults for solar-powered actuation boards, where the energy model is as much the product as the electronics.

- Inductive-load abuse testing now precedes everything on any solenoid-driving carrier — the flyback defect was findable in the first hour with a scope and one valve, and every energy-path element (blocking diodes, ideal-diode controllers, charge-path leakage) gets dark-panel and deep-discharge verification early, because the back-feed leak was invisible in every sunny-day test.

# Lessons Learned

- The CM4/Pico W split needs its sleep-wake contract validated adversarially: the Pico W's wake line worked, but a UART character arriving mid-wake-transition was silently dropped, losing one command in roughly 300 wake cycles — a race we now test for explicitly on every duty-cycled two-processor design.

PRODUCT

LET’S TALK

# SiliconCentric

EMAIL

sandesh@siliconcentric.com

PHONE

(669) 356-1998

WEB

https://www.siliconcentric.com

ADDRESS

San Jose, CA, USA

# Precision Irrigation

## Agriculture Precision Irrigation Controller Raspberry Pi

Built by SiliconCentric

SiliconCentric