RESEARCH-ARTICLE

Secure-by-Design Hardware Architecture for Real-Time IoT with Cycle-Bounded Trust and Redundancy-Aware Sensor Fusion

MAITHAM AL-RUBAYE, University of Vienna, Vienna, Vienna, Austria

ATAKAN ARAL, University of Vienna, Vienna, Vienna, Austria

Open Access Support provided by:

University of Vienna

Published: 01 December 2025

Citation in BibTeX format

UCC '25: 2025 IEEE/ACM 18th International Conference on Utility and Cloud Computing
December 1 - 4, 2025
France, France

Conference Sponsors:
SIGARCH

Secure-by-Design Hardware Architecture for Real-Time IoT with Cycle-Bounded Trust and Redundancy-Aware Sensor Fusion

Maitham Al-Rubaye
University of Vienna
Vienna, Austria
maitham.al-rubaye@univie.ac.at

Atakan Aral
Faculty of Computer Science
University of Vienna
Vienna, Austria, Austria
atakan.aral@univie.ac.at

Abstract

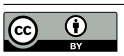
Real-time industrial IoT control systems that operate on physical variables depend on sensor readings that are assumed to be accurate, timely, and trustworthy. These assumptions break down under attack scenarios, sensor degradation, or physical tampering. We present a hardware-timed trust enforcement design that enforces trust decisions at the sensor fusion layer. The design targets post-ADC gating; all timing and fault results were obtained on a cycle-bounded, non-preemptive ESP32 firmware realization. Selected enforcement blocks were also built as discrete hardware but did not produce the reported metrics. The design combines redundancy-aware data integration with a continuous trust evaluation driven by hardware logic. Sensor nodes are linked to their identity through an on-chip eFuse device identifier read at boot and bound to the sensing channel. Data from redundant sources is filtered through a median-based fusion rule that considers delay patterns, value consistency, and trust history. When trust degrades, the ESP32 realization applies a cycle-synchronous digital mask to the post-ADC sample; GPIO lines are used for timing/observability only. We validate the architecture on a closed-loop testbed for IoT process control under spoofed and fault inputs. The results show bounded recovery latency and isolation of corrupted data within the same cycle in which degradation is first detected or, in the worst case, within four control cycles, depending on the pre-update trust level. In this prototype, gating occurs after ADC: corrupted digital samples are blocked from fusion within the same cycle, while analog front-end saturation cannot be prevented. These results were obtained with an ESP32 firmware realization that applies post-ADC gating within the control tick; discrete hardware builds were not used for the reported measurements.

CCS Concepts

• **Computer systems organization → Real-time systems**; **Embedded and cyber-physical systems**; • **Hardware → Safety critical systems**; • **Security and privacy → Embedded systems security**.

Keywords

real-time IoT control, hardware trust enforcement, redundant sensor fusion, cycle-synchronous gating.



UCC '25, Nantes, France

ACM ISBN 979-8-4007-2285-1/25/12
https://doi.org/10.1145/3773274.3774264

ACM Reference Format:
Maitham Al-Rubaye and Atakan Aral. 2025. Secure-by-Design Hardware Architecture for Real-Time IoT with Cycle-Bounded Trust and Redundancy-Aware Sensor Fusion. In *2025 IEEE/ACM 18th International Conference on Utility and Cloud Computing (UCC '25), December 01–04, 2025, Nantes, France.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3773274.3774264

1 Introduction

IoT-enabled industrial control systems rely on distributed sensing to monitor physical quantities such as level, flow, vibration or temperature [9, 15]. The sensors involved are exposed and installed in environments where tampering, signal degradation, and miscalibration occur without immediate detection [6, 9, 13]. These systems depend on synchronized and trustworthy input from their sensors and operate under strict real-time constraints that leave little opportunity for verification or recovery after data processing [3, 14]. Unintended or intended disruptions to sensor trustworthiness can propagate through the control path and modify the system behavior before any security mechanisms take effect [7, 19]. Software-level security measures are useful in many domains, but are often too delayed or inflexible to prevent physical-layer data faults from reaching actuators in real time [10, 12].

Hardware-based trust mechanisms can enforce sensor authenticity and measurement validity without relying on task schedulers or operating systems [8, 12]. In this work the enforcement logic is designed for post-ADC gating; parts were built as discrete hardware, but measurements use the ESP32 firmware realization. To evaluate timing behavior and collect controlled fault-injection data, we additionally provide a cycle-bounded, OS-independent firmware realization on ESP32, where gating is applied after ADC. In this work, we focus on embedding physical trust mechanisms in the data acquisition and fusion pipeline. Each sensor is linked to a unique on-chip eFuse device identifier read at boot and associated with its input channel and this identifier anchors the trust mapping before data is accepted [8, 10]. Trust is managed as a dynamic value, not a binary state [19]. Incoming data is verified for format consistency and evaluated against historical trust metrics [19]. Redundant data streams are checked in real time, and the fusion logic evaluates the agreement between sources based on both content and arrival patterns [4, 11]. In the event of trust degradation, dedicated hardware isolates the affected sensor to prevent altered data from affecting the control logic of the real-time execution path [12].

The architecture is designed to operate in real-time IoT control loops with bounded cycle deadlines [14, 15]. Each sensing node communicates with a local fusion controller that maintains low-latency trust evaluations and applies rejection or acceptance policies within microsecond limits [15]. The decision process runs independently

of the central processor and avoids latency caused by task switching or software scheduling [12]. This setup ensures that degraded or tampered signals are blocked in the hardware before reaching the actuator control path [15]. The trust evaluation process considers timing inconsistency, outlier distance, and vote consistency among redundant sources [4, 11]. Sensor removal is non-blocking and does not stall real-time operation while maintaining real-time behavior even during partial failure or ongoing attack [1]. The architecture was tested both in hardware-level circuits and in a dual-tank testbed with redundant sensor arrays. For systematic evaluation and easier trace collection, a pair of ESP32 microcontrollers was used to simulate the enforcement path in a cycle-bounded firmware environment.

The attack scenarios included delayed signals, spoofed values, and impersonation of the sensor node [9]. The system response was measured in terms of detection latency, isolation of invalid data, and uninterrupted operation of the control output. The hardware-based isolation path blocked corrupted data within the same execution cycle once trust dropped below threshold; in cases where the pre-update trust score started well above threshold, isolation followed within at most four cycles, consistent with the bound in Section 4.8. In contrast to methods dependent on firmware logic or cloud validation [18], this design avoids external dependencies and ensures that decisions are made within the latency limits of the physical process. This work introduces a practical architecture that enforces trust directly at the boundary between sensing and actuation in real-time IoT control systems. The design is evaluated as a hardware-timed, OS-independent firmware path with hardware gating (post-ADC) on ESP32 for timing/fault studies, while selected enforcement functions were also validated in direct hardware implementations.

This study asks whether sensor-level trust enforcement, implemented as a cycle-bounded, non-preemptive firmware path with hardware gating, can preserve control-loop integrity in IoT systems. The investigation is limited to three mechanisms: (i) on-chip verification of physical identification marks and timing regularity; (ii) cycle-synchronous isolation of faulty or spoofed measurements through combinational gating; and (iii) redundancy-aware fusion that admits only data streams whose local trust scores remain above threshold. No reliance is placed on higher-layer cryptography, operating-system services, runtime configuration, or cloud connectivity. Our main research question is *Can cycle-bounded trust enforcement without OS preemption sustain deterministic and safe real-time sensor fusion in IoT control loops?*

Validation takes place on a dual-node testbed outfitted with six redundant sensing channels per physical quantity, where latency, value offset and impersonation faults are injected under closed-loop conditions. Key metrics are detection latency, gating determinism, and safety of actuator commands when some or all sensors are excluded. Long-haul network security, power-analysis resistance of the MCU/SoC fabric, and economic cost modeling lie outside the present scope. The remainder of the paper addresses this question through circuit design, timing-constrained integration, and empirical evidence.

Main contributions.

(1) Cycle-bounded trust evaluation that gates faulty/spoofed values within the same control cycle, implemented as non-preemptive firmware (post-ADC).
(2) Redundancy-aware freeze rule that prevents mutual degradation under disagreement and stabilizes isolation decisions.
(3) Safe-state hold on total sensor loss with automatic resumption within three cycles of the first recovered channel.
(4) Empirical study on a dual-node testbed: 5× lower isolation latency and 10× lower false-isolate rate than firmware checks and 2-of-3 voting.

2 Threat and Failure Model

We structure the threat and failure model using a standard industrial control security taxonomy. The assets, adversary, fault classes, protection goals, and non-goals are consistent with established treatments of resilient control and ICS security [9, 15]. The focus reflects closed-loop timing and sensor integrity in attack-resilient control studies [14, 15]. Scope is limited to in-loop tampering and timing faults at the sensing boundary. Remote network intrusion paths and trusted execution for controller logic are excluded. This organization aligns with previous IIoT work on redundancy, diversity, and reinforcement at the sensing layer [6, 7].

Assets. The architecture safeguards (i) per-cycle integrity of each scalar measurement delivered to the fusion stage and (ii) deterministic timing of the closed loop.

Adversary. Adversary with physical access to the field wiring may

- inject/replay sensor values by analog spoofing/digital replay,
- defer genuine samples by inserting buffering hardware,
- substitute the transducer on the field wiring; controller–channel identity binds at boot and does not authenticate the transducer,
- perturb clock or power rails to provoke timing faults.

Identity binding applies to the controller input channel. The eFuse identifier is read at boot and never placed on the field bus, ensuring the provenance of the channel–controller pair but not authenticating the remote transducer. Substitution of the transducer element on the wiring, therefore, bypasses identity checks and is only constrained by timing, rate, or peer-consistency violations.

These assumptions reflect the deployment model: configuration registers are written only once at provisioning and protected by hardware lock bits, making runtime modification infeasible. Internal routing is not exposed to field wiring and cannot be probed via the I/O interface. Gating is implemented after ADC conversion, which prevents contaminated digital samples from propagating into fusion and actuator commands within the same cycle. However, analog injection at the sensor line may still saturate the ADC.

Fault classes. Accidental faults include drift, aging, electromagnetic interference, and transient disconnections. Malicious faults encompass value spoofing, timing deviations that exceed the cycle budget and channel-identity replay or misbinding (see Table 1).

Protection goal. For any single fault or attack persisting for at least one control cycle, the system must prevent the corrupted *digital* measurement from influencing the fused estimate during that cycle and must continue supplying bounded-latency estimates from the remaining trusted channels. This guarantee applies to post-ADC data: analog-front-end saturation from upstream injection cannot be prevented by digital gating, though post-ADC gating still blocks contaminated samples from entering fusion within the same cycle.

Table 1: Adversary capabilities, detection paths, and risks.

Capability	Detection path	Latency	Residual risk
Value spoofing (step)	Rate-of-change δ_{rate} check	1 cycle	Slow ramp avoids trigger
Timing manipulation	Sync bound δ_{sync}	1 cycle	Perfect alignment
Peer mismatch	Pairwise consensus	Multi-cycle (trust decay)	If collusion >= majority
Identity replay	eFuse ID binding at boot	N/A (boot only)	Physical substitution
analog injection (line/sensor excitation)	Not directly detected	Depends on manifestation	ADC saturation not prevented

Non-goals. The study excludes long-haul network attacks, side-channel extraction of configuration bits, board-level denial-of-service that forces a global reset, **and perfect bus-level man-in-the-middle injection that preserves timing regularity and value statistics while replacing the analog sensing element.**

Attacks that act upstream of the ADC, such as sensor excitation spoofing or analog line injection are not prevented from reaching the converter. Post-ADC gating still constrains their impact by excluding corrupted samples before they influence fusion or actuators within the same cycle. If the injected analog amplitude drives the ADC into saturation, this cannot be reversed by digital gating. Analog front-end gating or sample-and-hold interlocks would be required to stop such attacks, but these remain out of the scope.

3 Related Work and Gaps

IoT-based control systems have traditionally prioritized physical dependability and digital security treated as a secondary concern [15]. The adoption of wireless sensor networks introduced new benefits but negatively expanded the attack surface, making control loops exposed to real-time modification, signal injection, and unauthorized access [9]. Redundancy and fault tolerance have been recommended at the sensor level to reduce the risk of failures [6] [7]. These methods are efficient for architectural planning, but they ignore the validation of sensor readings and rely on software-level checks, which expose the physical sensing layer to in-loop contamination. Operational guidance for ICS/OT codifies these timing and safety constraints [2, 16]. Trust management models have introduced mechanisms to evaluate sensor behavior using direct and indirect trust, such as reward-penalty and feedback-based methodologies to tackle deceiving behavior in sensors and edge devices [19]. However, their reliance on historical data and delayed evaluations makes them misaligned with the time requirements of closed-loop industrial control. Trust decisions are managed by edge brokers or software agents, not within the sensing path, creating temporal gaps that reduce protections against transient attacks.

Hardware-based isolation is proposed as a strategy to reduce surface attacks in critical components of industrial systems. Dual microcontroller configurations separate the I/O control from the network to reduce the exposure to external attacks [12]. This protects the physical control cycle from incoming network traffic interference or delays by using real-time sensing and actuation tasks. The trust evaluation in this setup remains implicit; the sensor verification does not happen in the physical layer and there are no approaches to drop measurements based on timing anomalies, past data or expected behavior. In an architecture lacking this enforcement mechanism, the response remains reactive and isolation occurs only after the damage has impacted the output path.

The enforcement of trust in programmable logic controllers (PLCs) at the hardware level using trusted execution environments (such as ARM TrustZone) has improved the protection of control logic against runtime manipulation and memory-based attacks [10]. The method overlooks direct sensor input inspection, trust is limited by the scan cycle, and forged or replayed data can enter the control loop unchallenged, especially in the absence of authenticated input peripherals. These leave a gap between secure execution and secure sensing, which is critical in real-time systems.

Sensor fusion strategies (e.g. redundancy and voting mechanisms) are applied to improve fault tolerance and resilience by combining multiple data streams. Modified MAP fusion rules can smooth out noise and minimize the outlier effect, but treat all sensors as equally trusted once they are deployed [4]. However, the absence of dynamic degradation and behavior evaluations means that faulty or drifting sensors can persist in affecting estimates if their outputs remain within statistical boundaries, regardless of physical inconsistency. IoT fusion techniques for real-time monitoring presuppose that sensing streams are always accessible and accurate, even in suboptimal circumstances. Centralized IoT-based fusion platforms are designed for large-scale monitoring, long-term data analysis and to enable efficient visualization and storage [11]. These designs are well-suited for monitoring, but do not address trustworthiness under real-time constraints. Trust remains constant throughout the operation, and anomaly detection occurs afterwards, which can lead to delays and compromise actuation integrity in real-time systems. These represent after-the-fact anomaly detection; in contrast, our design enforces inclusion/exclusion inside the sensing path and within the control cycle.

Hardware trust enforcement generally prioritizes execution protection over sensing integrity. Even robust solutions like TrustZone-based isolation [10] focus on code-level assurance, while fusion models [4, 11] suppose sensor validity even under compromised conditions. Across these systems, trust is not continuously updated or enforced in real time. Lack of hardware-level enforcement during sensing means that control systems remain exposed to attacks.

Fault-tolerant hardware designs and partial reconfiguration techniques have been proposed to recover from hardware faults. Authentication schemes confirm node identity and ensure access control [8] [1]. At the sensing boundary, PUF-anchored authenticated sensor designs bind device identity and keys to device physics to establish a hardware root of trust in the sensor itself [5, 17]. However, they do not evaluate real-time sensor behavior or enforce trust aligned with actuation cycles. Trust-aware architectures at the application or network layers (such as those proposed for water or environmental systems [13] [19]) introduce additional latency, making them unsuitable for hard real-time applications. Across these systems, trust is static, delayed, or limited to node identity. There are no existing designs that currently enforce trust constraints directly

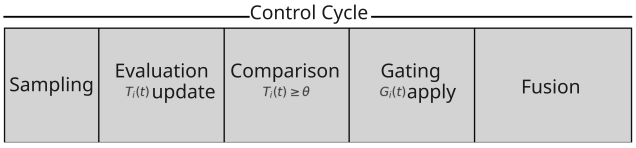


Figure 1: Cycle-timed trust enforcement.

in the sensing path, at the point of measurement, and under the timing constraints required by physical control systems.

Model-based automation strategies often apply predictive diagnostics to anticipate failures or adjust control actions based on internal estimators. For example, prognostics enabled control loops in building automation systems use modeled degradation behavior to drive corrective responses [3]. Such approaches require runtime estimation, learning, or model tuning, which are excluded from the present architecture to maintain cycle-level determinism and software independence. Some approaches attempt to co-design resilient control systems with platform-aware constraints on timing, deployment, and attack resilience [14]. These focus on synthesis and verification of system properties rather than enforcing real-time trust decisions within the sensing path. This architecture applies bounded enforcement in line with signal acquisition and without offloading verification to external analysis tools.

The existing literature separates sensor identity from sensor behavior, delays trust evaluation beyond the timing limits of the control loop, and applies recovery mechanisms only after system exposure. We are not aware of any prior work that enforces trust constraints directly in the measurement path (at the time of sensing) under hardware-timed conditions. The absence of real-time enforcement leaves a critical gap in control systems that depend on immediate and irreversible physical interaction. This work addresses this gap by integrating trust enforcement at the sensor interface itself, evaluated in real-time and synchronized with control-cycle deadlines, without relying on software or after-correction.

4 System Design

We distinguish the trust value at cycle start and end:

$$T_i^-(t) \text{ (pre-update at start of cycle } t\text{)},$$
$$T_i^+(t) \text{ (post-update after processing } x_i(t)\text{)}.$$

State rolls forward as $T_i^-(t{+}1) = T_i^+(t)$.

The architecture enforces trust decisions in the sensing layer under a cycle-bound model. Each sample $x_i(t)$ passes through a hardware evaluator that maintains a degradable trust score $T_i(t) \in [T_{\min}, T_{\max}]$. A comparator applies the gating condition

$$G_i(t) = \begin{cases} 1, & T_i^+(t) \geq \theta, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

to allow or block sensor data within the same control cycle. All updates (score adjustment, comparison, and gating) are completed before fusion begins. Figure 1 shows this alignment within the fixed-cycle timing window. No higher-layer software arbitration or task scheduling is involved in the enforcement path.

4.1 Architecture Overview

Each sensing node drives a trust evaluator that applies per-cycle score update and gating without OS involvement or preemption. That evaluator tracks temporal deviation, rate change, and peer consistency to update the trust register $T_i(t)$ in fixed-point logic. A sensor output is forwarded to fusion logic only if $G_i(t)$; see Eq. (1).

The score is stored in a saturating register and modified in each cycle as in Eq. (2).

$$T_i^+(t) = \min\Big(T_{\max},\ \max\big(T_{\min},\ T_i^-(t) + \delta^+ \mathbf{1}\{\neg v_i(t)\} - \delta^- \mathbf{1}\{v_i(t)\}\big)\Big) \tag{2}$$
$$T_i^-(t{+}1) = T_i^+(t).$$

where $v_i(t)$ denotes a detected violation in the current measurement and $\mathbf{1}\{\cdot\}$ is the indicator function (1 if the predicate holds, 0 otherwise). Here $\delta^+ > 0$ and $\delta^- > 0$ are fixed step sizes: compliance with all checks ($\neg v_i(t)$) increases trust by δ^+, while any violation ($v_i(t)$) decreases trust by δ^-. All arithmetic executes in a straight-line, branch-bounded firmware path aligned to the control tick with no OS scheduling or blocking calls. Trust evaluation, comparison, and gating are cycle-aligned and applied before fusion begins. Local agreement logic for redundant sensors runs in parallel and checks pairwise consistency. All paths are isolated, clock-synchronous, and enforced without OS arbitration, polling, or dynamic allocation.

4.2 Sensing Path and Trust Evaluation Logic

Each analog signal $x_i(t)$ is digitized and passed to a local evaluator that computes trust based on timing variation, rate change, and peer agreement.

Violation predicate. We use fixed thresholds $\delta_{\text{sync}} > 0$, $\delta_{\text{rate}} > 0$, and $\Delta_{\text{dev}} > 0$ and define

$$v_i^{\text{tim}}(t) := \big|\, t_i(t) - t_{\text{nom}} \,\big| > \delta_{\text{sync}},$$
$$v_i^{\text{rate}}(t) := \big|\, x_i(t) - x_i(t{-}1) \,\big| > \delta_{\text{rate}},$$
$$v_i^{\text{peer}}(t) := \Big(\forall j \neq i :\ \big|\, x_i(t) - x_j(t) \,\big| > \Delta_{\text{dev}}\Big),$$

and the composite predicate

$$v_i(t) \ = \ v_i^{\text{tim}}(t) \ \vee \ v_i^{\text{rate}}(t) \ \vee \ v_i^{\text{peer}}(t).$$

ADC saturation rule: if the ADC returns full-scale code for channel i at cycle t, set $G_i(t) \leftarrow 0$ immediately, independent of $T_i^+(t)$.
Here $t_i(t)$ is the ADC sample-ready timestamp for channel i at cycle t, and t_{nom} is the nominal sample time on the control tick.

The trust score evolves according to (2), where $T_i^-(t)$ is the pre-update value at cycle start and $T_i^+(t)$ is the post-update value after processing $x_i(t)$. The trust score register operates with saturating arithmetic, ensuring that

$$T_{\min} \leq T_i(t) \leq T_{\max}, \tag{3}$$

for both pre- and post-update states at all times.

At the end of each cycle, the updated trust score $T_i^+(t)$ is compared against the gating threshold θ; see Eq. (1) and the result $G_i(t) \in \{0, 1\}$ determines whether the measurement is gated to the fusion logic. All computations are completed within the same cycle without intermediate buffering, polling, or blocking calls.

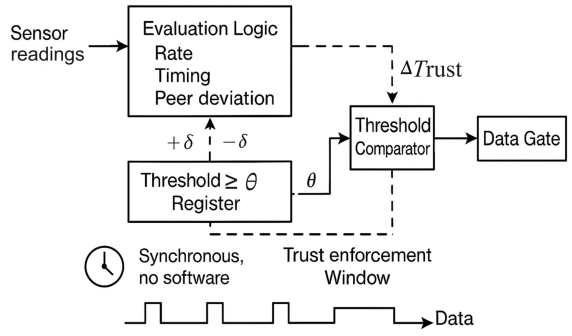


Figure 2: Trust score update and gating logic for real-time isolation within a clock cycle.

Pairwise agreement is enforced using fixed-point arithmetic across all active redundant sensors. A deviation exceeding the margin Δ_{dev} (from all peers) applies a penalty decrement δ^-. If multiple sources disagree without a clear majority, trust scores are frozen:

$$T_i^-(t{+}1) = T_i^-(t) \quad \forall i \in \text{group}, \quad \text{if no consensus},$$

This rule avoids mutual degradation and oscillation under transient divergence. The full evaluator logic (sampling, score update, comparison, and gating) is implemented as fixed-time, branch-bounded firmware executed each control tick. All paths are aligned to the main control clock to trace and repeat enforcement without task scheduling, blocking calls, or dynamic allocation. Figure 2 shows the enforced update and gating logic for a single sensor path.

4.3 Hardware-Gated Fusion Path

Each sensor channel is gated based on its trust score as a post-ADC digital mask in the ESP32 firmware realization used for measurement. GPIO is used for timing/observability; no line-side analog switch is exercised in the reported results. The mask is applied within the same cycle without buffering. The fusion stage receives the set $\{x_i(t) \mid G_i(t) = 1\}$ and computes an estimate:

$$\hat{x}(t) = \begin{cases} \text{median}\big(\{\, x_i(t) : G_i(t) = 1 \,\}\big), & \sum_i G_i(t) \geq 2, \\ x_{i^\star}(t) \text{ (low-confidence)}, & \sum_i G_i(t) = 1, \\ \varnothing, & \sum_i G_i(t) = 0. \end{cases} \tag{4}$$

where $i^\star$ denotes the unique index with $G_{i^\star}(t) = 1$ when exactly one channel is trusted.

No interpolation is applied, and missing values are not reconstructed. The system marks low-confidence outputs and switches mode transparently, which enforces strict separation between trust evaluation and fusion output. When $\sum_i G_i(t) = 0$ so that $\hat{x}(t) = \varnothing$, the actuator stage performs an implicit last-value hold until the explicit safe-state mask (Section 4.7) engages after n_{safe} consecutive such cycles. The gating layer transforms continuous trust evaluation into discrete per-cycle inclusion decisions, maintaining determinism and preventing contamination. Fusion remains real-time, stateless, and invariant under input cardinality changes.

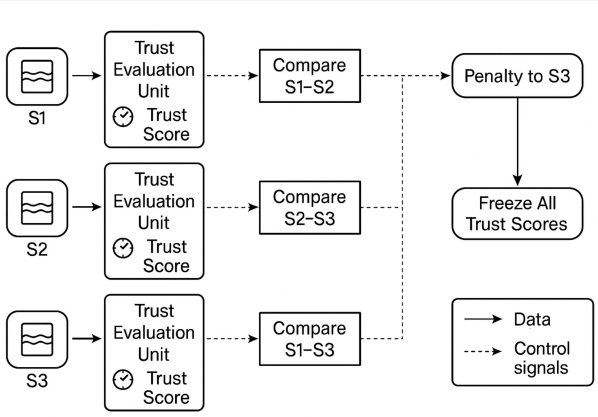


Figure 3: Cycle-synchronous pairwise comparison and trust response in redundant sensor groups.

4.4 Redundancy-Aware Conflict Logic

Redundant sensors $\{x_1(t), \ldots, x_n(t)\}$ measuring the same physical quantity are grouped. Each sensor i evaluates local pairwise disagreement with all other group members:

$$d_{ij}(t) = |x_i(t) - x_j(t)| > \Delta_{\text{dev}} \quad \text{for } j \neq i.$$

If multiple sources disagree without a clear majority, trust updates for the group are frozen and channels with insufficient agreement are *temporarily gated out*: during freeze, set $G_i(t) \leftarrow 0$ for any channel with $a_i(t) < m{-}1$ until a strict majority reappears. For $|\mathcal{G}|{=}2$, gate both channels during freeze. This prevents outliers from influencing fusion while trust updates are suspended.

Fusion behavior follows Eq. (4) based on the set of trusted channels at the current cycle: if at least two channels are trusted, the median is used; if exactly one is trusted, its value is forwarded with a low-confidence tag; if none are trusted, no fused output is produced. The group-level freeze condition suspends trust updates during disagreement but does not alter the fusion operator or its thresholds. When any sensor regains agreement and exceeds threshold $T_i(t) \geq \theta$ for two cycles, its data re-enters fusion without reset. Figure 3 illustrates local pairwise comparison and response logic under redundancy-aware trust enforcement.

4.5 Trust Enforcement and Isolation Signal Path

Each sensor channel maintains $T_i(t) \in [T_{\min}, T_{\max}]$ with the per-cycle update of Eq. (2); gating compares the post-update value $T_i^+(t)$ to θ via Eq. (1). Each cycle, the trust comparator evaluates the post-update value $T_i^+(t)$ against the gating condition in Eq. (1), which then applies a digital mask to the post-ADC sample before fusion. If $G_i(t) = 0$, the signal is suppressed by a zero mask, tri-state block or logic-level disconnect, depending on the target. No buffering or intermediate latches exist in the path and comparator output takes effect within the same cycle. All trust update, comparison, and gating occur synchronously within the control window, with bounded control flow and without polling, blocking calls, or dynamic allocation. Reenable is automatic when $T_i(t) \geq \theta$, requiring no reset or arbitration.

Threshold Selection.

A five-minute sweep ($\sim$ 1.5 M cycles at 200 μs per cycle) with clean operation, followed by offset perturbations ($\pm$ 0.5–2.0%), produced 24,000 labeled events for ROC analysis via stratified subsampling. The configuration $\delta^- = 4$, $\theta = 48$, $\delta^+ = 1$ minimized false isolates to $< 2 \times 10^{-5}$ per cycle while keeping missed detections under 0.3%. Only these three parameters are patched in the header at deployment; no rebuild of the control logic is required.

Table 2: Parameters for trust enforcement.

Symbol	Value	Role
δ^+	1	Trust increment per consistent sample
δ^-	4	Trust decrement per violation
θ	48	Gating threshold
$T_{\min}, T_{\max}$	0, 63	Score bounds (saturating register)
Δ_{dev}	1.0% FS	Peer-deviation tolerance
δ_{rate}	0.5% FS/-cycle	Rate-of-change bound
δ_{sync}	4 μs	Timing regularity bound
T_{cycle}	200 μs	Control cycle period
Δ_{trust}	$\leq$ 0.05 ms	Enforcement window
n_{safe}	6	Safe-hold cycles before fallback

4.6 Timing Boundaries and Constraints

Each control cycle operates under a fixed-phase timing model:

$$T_{\text{cycle}} = T_{\text{sample}} + T_{\text{eval}} + T_{\text{gate}} + T_{\text{fusion}} + T_{\text{actuate}},$$

where T_{sample} includes ADC latency and signal stabilization, T_{eval} computes trust score updates, T_{gate} applies gating logic based on $T_i(t)$, T_{fusion} processes available values, and T_{actuate} commits control output. T_{eval} includes the computation of $v_i(t)$ and the update $T_i^+(t)$ from the pre-update state $T_i^-(t)$. Trust enforcement is cycle-synchronous and satisfies:

$$T_{\text{eval}} + T_{\text{gate}} + T_{\text{sync}} \leq \Delta_{\text{trust}} < T_{\text{fusion}},$$

for isolation decisions precede data entry into the fusion stage. Under synthetic ISR load (1 kHz GPIO + ADC), $T_{\text{eval}}+T_{\text{gate}}+T_{\text{sync}} \leq 40\ \mu\text{s} < 0.25\, T_{\text{cycle}}$.

This bound guarantees that every trust evaluation and gating decision is computed within the same control cycle as the triggering sample. Multi-cycle latency in isolation arises only from the trust-score decrement path, not from timing delays in the evaluator. No task switching, polling, or asynchronous behavior is present, and all trust operations follow fixed time code paths aligned to the control clock. Cross-domain lines (ADC→CPU, CPU→GPIO) use two-flip-flop synchronization; the budgeted T_{sync} is included in Δ_{trust}. All stages of trust evaluation, threshold comparison and gating are complete within the interval Δ_{trust}, with output readiness synchronized to the fusion clock. No delay paths extend beyond the enforcement window. Figure 4 illustrates this alignment.

4.7 Safe-State and Degradation Handling

Hold condition. Let $n_{\text{safe}} = 6$ denote the maximum allowable consecutive cycles without trusted input. If $G_i(t) = 0\ \forall i \in \mathcal{G}$, for

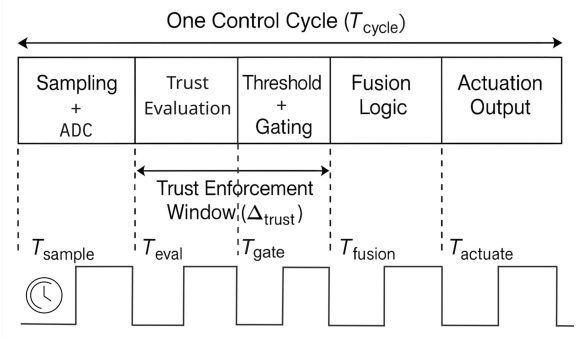


Figure 4: Timing boundaries for sensing, trust enforcement and fusion in one control cycle.

n_{safe} successive cycles, the system applies a zero-order hold:

$$u(t) = u(t_{\text{last trusted}}), \quad \forall t \geq t_{\text{loss}},$$

where $u(t)$ is the actuator setpoint and $\mathcal{G}$ is the sensor group monitoring a common quantity.

Recovery rule. Exit from safe-state occurs once at least one sensor in $\mathcal{G}$ remains trusted for two consecutive cycles. Here $G_i(t) = \mathbf{1}\{T_i^+(t) \geq \theta\}$ follows the post-update value:

$$\exists i \in \mathcal{G} \text{ s.t. } G_i(t) = 1 \ \wedge \ G_i(t+1) = 1.$$

Upon reentry, the first recovered value is smoothed with a lag:

$$x_{\text{fused}}(t+1) = \frac{1}{2}\left(x_{\text{fused}}(t) + x_i(t+1)\right).$$

We choose n_{safe} such that $n_{\text{safe}}\, T_{\text{cycle}} \lesssim 0.1\tau$, where τ is the dominant plant time constant; in our rig, $n_{\text{safe}} = 6$ satisfied this bound.

Timer implementation. A 6-bit down-counter (capacity 64 cycles) decrements each cycle without a trusted input and resets when $\bigvee_i G_i(t) = 1$. For this prototype, the safe-hold limit was set to $n_{\text{safe}} = 6$ cycles, well within the counter capacity. When the counter reaches the programmed limit, a mask bit inhibits the actuator-update register. Both counter and mask are clock-synchronous to ensure deterministic entry and exit from hold mode.

Experimental trace. Figure 5 shows a loss-recovery run where all three sensors were gated during cycles 400–430. The actuator maintained its last setpoint (55%) during loss and resumed correct fusion output within three cycles of recovery in this trace. The hold duration depends on when at least one sensor regains trust, while entry is fixed at $n_{\text{safe}} = 6$ cycles without input.

4.8 Formal Guarantees

Bounded-latency constraint. Let T_{eval} and T_{gate} denote the post-build measured code-path delays of the evaluation logic and gating driver, respectively. The timing model satisfies:

$$T_{\text{eval}} + T_{\text{gate}} \leq \varepsilon, \quad \varepsilon = 0.040\,\text{ms} < \tfrac{1}{4} T_{\text{cycle}}.$$

Thus, every gating decision completes within the control cycle of the triggering sample. This timing bound applies to the evaluator logic itself. The longer isolation horizon discussed below (up to

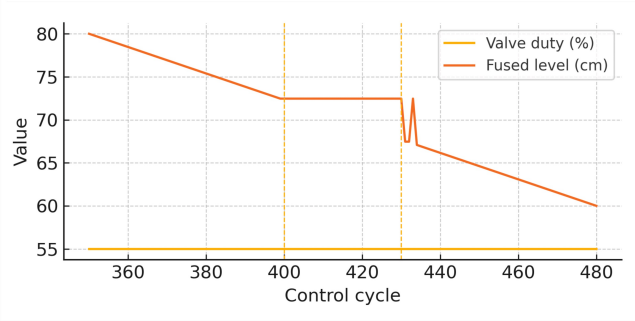


Figure 5: Valve duty (top) and fused level (bottom) during complete sensor loss and graceful recovery for Node A.

Table 3: Measured per-stage timing (ESP32 prototype).

Stage	n=1 channel	n=32 channels
Evaluation T_{eval}	0.8 μs	25.6 μs
Gating T_{gate}	0.3 μs	0.3 μs
Sync T_{sync}	0.8 μs	0.8 μs
Fusion T_{fusion}	1.0 μs	8.4 μs
Total enforcement	1.9 μs	26.7 μs

four cycles) results from gradual decay of the trust score relative to the threshold, not from evaluator delay.

Isolation. If $T_i^+(k) < \theta$ then

$$G_i(k) = 0 \quad \text{and} \quad x_i(k) \notin \hat{x}(k),$$

because the comparator resolves within $T_{\text{eval}}+T_{\text{gate}} \leq \varepsilon < T_{\text{fusion}}$.

Coverage bound. For fixed δ^-, the worst-case number of cycles until gating depends on the pre-update state. If a violation first occurs at cycle k with pre-update value $T_i^-(k)$, then

$$m^\star \;=\; 1 + \left\lfloor \frac{T_i^-(k) - \theta}{\delta^-} \right\rfloor$$

gives the number of cycles from that violating sample to its exclusion. With $T_{\max} = 63$, $\theta = 48$ and $\delta^- = 4$, the bound is $m^\star \leq 4$.

Saturation condition. All trust scores $T_i(t) \in [T_{\min}, T_{\max}]$ use saturating arithmetic:

$$T_i(t{+}1) = \min\left(\max\left(T_i(t) \pm \delta, T_{\min}\right), T_{\max}\right),$$

guaranteeing no overflow or wraparound. Static analysis and runtime assertions confirm this constraint.

4.9 Qualitative Resource Footprint

Hardware target. The prototype executes on ESP32-WROOM-32 (Xtensa LX6, 240 MHz). All evaluator logic is statically linked under ESP-IDF v5.0 and pinned to PRO_CPU. Second core remains idle.
Memory usage. Each channel reserves a register for $T_i(t)$, an accumulator for timing, and a register for $x_i(t{-}1)$, with total per-channel RAM < 1 kB. Total usage remains within 320 kB SRAM across full deployments.
Timing cost. Total cycle latency scales linearly with active channels:

$$T_{\text{evaluator}}(n) = n \cdot T_{\text{op}},$$

Table 4: Hardware testbed used for evaluation

Component	Specification
Sensor redundancy	6 channels per quantity (independent sensing paths)
Controller MCUs	2× ESP32-WROOM-32 (Xtensa LX6, 240 MHz, 320 kB SRAM)
Sampling/control timing	Fixed control period $T_{\text{cycle}} = 200\,\mu$s; on-chip ADC; cycle-synchronous gating
Trust gating I/O	GPIO-matrix dual-rail lines ENABLE_P/ENABLE_N; digital-domain tri-state/zero-mask disconnect after ADC
Clocking	Multiple domains (SAR ADC vs. APB/GPIO); two-flip-flop synchronizers; comparator and gate resolved within ≤ 0.05 ms
Power rail	3.3 V; monitored with INA219; additional evaluator power $P_{\max} = 46$ mW (32 channels)
Adversarial equipment	ChipWhisperer CW308 for clock/power glitching (±150 mV, 5–80 ns pulses)

where T_{op} is the fixed-time trust evaluation per channel. The measured latency under full load remained below 0.05 ms.
Portability. No specialized coprocessor instructions, cache controls or memory fences are required. Any microcontroller with a deterministic interrupt model and sufficient clock rate can host the evaluator. Lower-frequency cores will limit channel count to preserve timing bounds.
Multi-quantity trust domain. The present design evaluates trust independently per sensor group (e.g., level, pressure). Inter-domain correlation (e.g., flow vs. pump current) is not enforced in this layer. Extending cycle-bounded trust to multi-quantity logic requires cross-group agreement rules and shared gating dependencies, which are excluded to preserve modularity and timing traceability.

5 Evaluation

The experiments were conducted on a dual-node hardware testbed representative of IoT process control. Table 4 summarizes the physical components and operating conditions used for fault injection and closed-loop validation. Faults were injected as additive offsets, jitter spikes, and peer mismatches during fixed-cycle windows. Each sensor $x_i(t)$ was perturbed by offset: $(x_i(t) \leftarrow x_i(t) + \Delta_{\text{off}})$, jitter: $(|x_i(t) - x_i(t{-}1)| > \delta_{\text{rate}})$, and mismatch: $(|x_i(t) - x_j(t)| > \Delta_{\text{dev}}\ \forall j \neq i)$. Here Δ_{off} is the offset amplitude used for fault injection; Δ_{dev} remains the peer-deviation margin for $v_i^{\text{peer}}(t)$.

Trust scores $T_i^-(t)$ declined under violation and recovered under compliance:

$$T_i^+(t) = \min\Big(T_{\max},\ \max\big(T_{\min},\ T_i^-(t) + \delta^+ \mathbf{1}\{\neg v_i(t)\} - \delta^- \mathbf{1}\{v_i(t)\}\big)\Big),$$

$$T_i^-(t{+}1) = T_i^+(t),$$

$$G_i(t) = \mathbf{1}\{T_i^+(t) \geq \theta\}.$$

This notation explicitly distinguishes $T_i^-(t)$ (the trust score at the start of cycle t) from $T_i^+(t)$ (the updated value after processing $x_i(t)$), and gating always compares against the post-update state.

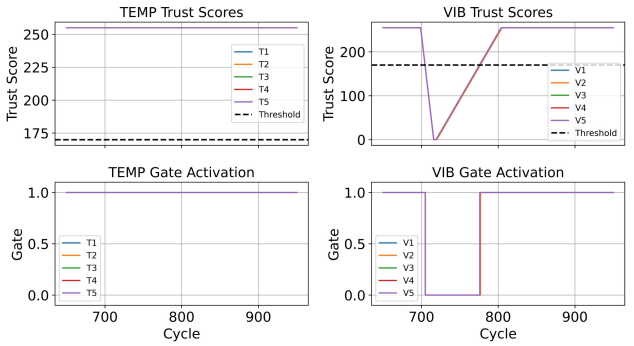


Figure 6: Trust scores and gate deactivation for TEMP and VIB during fault injection.

All updates completed within the control window. No gated channel contributed to fusion during degradation. Figure 6 shows score evolution and gate transitions for TEMP and VIB under fault injection. Fusion output followed the trusted input set in Eq. (4).

Fallback cleared automatically once any $T_i(t) \geq \theta$. Coverage exceeded 98% across all quantities except VIB, which dipped to 93.2% during cycles 700–720. Freeze response suspended trust updates across a sensor group when pairwise disagreement or rate violations prevented consensus.

$$a_i(t) = \sum_{\substack{j \in \mathcal{G} \\ j \neq i}} \mathbf{1}\left\{ |x_i(t) - x_j(t)| \leq \Delta_{\text{dev}} \right\},$$

Freeze condition:

$$C(t) = 1 + \max_{i \in \mathcal{G}} a_i(t), \qquad m = \left\lfloor \frac{|\mathcal{G}|}{2} \right\rfloor + 1, \tag{5}$$

$$\text{If } C(t) < m \implies T_i^-(t{+}1) = T_i^-(t), \quad \forall i \in \mathcal{G}.$$

where $\mathcal{G}$ is the sensor group. The hold lasted three cycles or until pairwise agreement resumed.

No sensor degraded due to a peer mismatch while frozen. Figure 7 shows freeze activation across quantities during fault intervals.

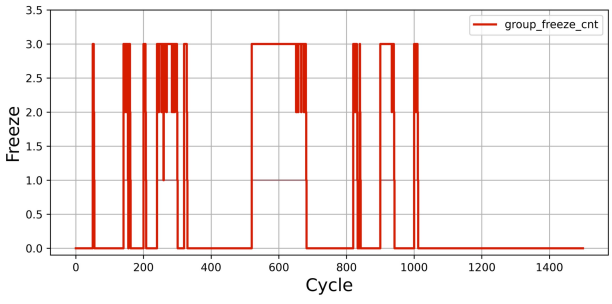


Figure 7: Freeze activation and release across sensor groups.

Baseline Comparison. The same test scenario was executed under two reference schemes:

- *Firmware checker:* sensor filtering by FreeRTOS task with $t_{\text{median}} = 190\,\mu\text{s}$, $t_{95\%} = 240\,\mu\text{s}$
- *Majority voter:* strict 2-out-of-3 scheme with no timing enforcement (we subsampled 3 of the 6 redundant channels to match the standard 2-of-3 voter).

Spoofed values of equal magnitude bypassed the voter in 47% of trials; note that this bypass rate (attack success under collusion) is a different metric from the false-isolate rate reported for SW/HYB/HW, which measures benign misclassification when no attack is present. In the step-to-offset trials we initialized T_i^- near θ, so a single δ^- decrement suppressed both channels; the measured evaluator+gate decision time was $38 \pm 4\,\mu$s. In general, isolation may require up to $m^\star \leq 4$ cycles depending on T_i^-, consistent with Section 4.8. Across 1000 injections:

$$\text{False isolate rate} < 5{\times}10^{-5} \text{ per cycle.}$$

This reduced false-isolate rate by ~14× vs. SW ($7.2 \times 10^{-4} \nrightarrow 5{\times}10^{-5}$) and ~2× vs. HYB ($1.1{\times}10^{-4} \nrightarrow 5{\times}10^{-5}$). The 2-of-3 voter was bypassed in 47% of equal-magnitude spoofing trials (colluding pair outvotes the honest channel). In step-to-offset injections, both spoofed channels violated δ_{rate} and were gated within $38 \pm 4\,\mu$s. Slow-ramp collusion (sub-δ_{rate}) remains a residual case without coalition detection or cross-quantity checks (out of scope).

Violation Analysis. Let $\mathcal{V}_i(t) \subset \{\text{tim}, \text{rate}, \text{peer}\}$ denote the active violation set for sensor i at cycle t. The per-type count was accumulated as:

$$N_{\text{type}}(t) = \sum_i \mathbf{1}_{\{\text{type} \in \mathcal{V}_i(t)\}}.$$

Peer mismatches dominated with over 2200 events. Rate-of-change violations registered moderate activity during fault windows. Timing-bound violations remained sparse and aligned with synthetic excursion intervals.

Score Dynamics. Scores followed bounded saturating updates:

$$T_i(t{+}1) = \min\Big(T_{\max},\ \max\big(T_{\min},\ T_i(t) + \delta^+ \mathbf{1}\{\neg v_i(t)\} - \delta^- \mathbf{1}\{v_i(t)\}\big)\Big).$$

Sensors without violation reached $T_{\max}$ and remained stable. Faulted sensors showed monotonic degradation and recovered automatically upon agreement. No overflow occurred. Figure 8 confirm regular, step-consistent score evolution across sensors and quantities.

Statistical Confidence. Latency and false-isolate metrics were drawn from ten independent runs of 20000 control cycles each, with 100 random fault injections per run (nonparametric bootstrap, 10k resamples). Let τ_{iso} denote isolation latency and ρ_{false} the false gating rate. The 95% confidence intervals across all runs satisfied:

$$\text{CI}_{95}(\text{median } \tau_{\text{iso}}) \leq \pm 3.4\,\mu\text{s}, \quad \text{CI}_{95}(\rho_{\text{false}}) \leq \pm 0.7{\times}10^{-5}.$$

These bounds confirm that per-run measurements shown in Figure 6 reflect steady-state system behavior.

Score Trace Validation. Figure 8 show per-sensor trust score evolution under spoofing and jitter faults. All transitions conformed to expected saturating dynamics with exact step sizes and bounded slope. The dip in trust score around cycle 700 in Figure 8 corresponds to the jitter fault injection window for the VIB channel. In this interval, the sequential measurements violated the rate-of-change bound δ_{rate} and trigger penalty decrements δ^- in Eq. (2). Once the injected jitter subsided and the sensor returned to a consistent timing and progression of the values, the score recovered

monotonically under the positive increment δ^+. The dashed threshold line in Figure 8 shows that gating was activated only during the violation window with reactivation occurring immediately once the trust score exceeded θ.

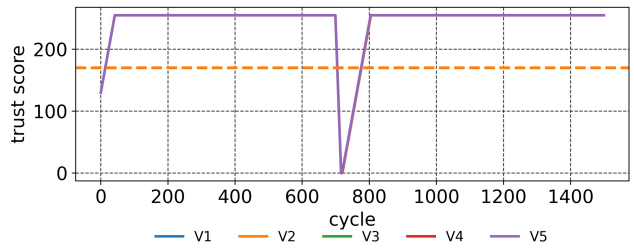


Figure 8: VIB sensor trust scores under induced jitter.

Gate Transition Determinism. Each gate transition $G_i(t) \rightarrow G_i(t{+}1)$ followed the comparator result without delay. Transitions matched trust state exactly:

$$G_i(t) = \mathbf{1}_{\{T_i^+(t) \geq \theta\}}, \quad \forall t.$$

No premature reactivation or missed deactivation was observed. All gates followed cycle-synchronous logic with no polling, memory buffering or runtime arbitration.

Unaddressed cases. Slow drift collusion, bursty dropout, shared mode failures were not tested. These require extended experimental runs and cross-quantity monitoring. In the present work they are treated as residual cases, noted in Section 2 and left as future work.

Power Profile. Power draw was measured with a calibrated INA219 monitor at 3.3 V:

$$I_{\text{idle}} = \text{baseline current (no evaluator)}$$
$$I_{\text{full}} = I_{\text{idle}} + 14\,\text{mA} \quad \text{(32 channels active)}$$
$$I_{\text{half}} = I_{\text{idle}} + 6\,\text{mA} \quad \text{(16 active)}$$
$$P_{\text{max}} = 46\,\text{mW} \ll 250\,\text{mW (RTU budget)}$$

The evaluator operates with the energy constraints of a solar powered system without modification to the power subsystem.

5.1 Baseline Comparison

Reference configurations. Two baselines were synthesized to assess latency and predictability:

- **SW:** A FreeRTOS task monitors sensor inputs and updates trust scores in software.
- **HYB:** Trust evaluation is performed in software, but gating is applied in hardware via GPIO pins at the end of each control cycle.

All test conditions (e.g., sensor types, sample rate, task mapping and gating topology) were held constant across variants.

Latency and false isolate rate. We compare three implementations—**SW** (software-only evaluator), **HYB** (software evaluator with hardware gating) and **HW** (fully hardware-timed evaluator and gating). Isolation latency (τ_{iso}) is measured from sample arrival to the gating decision; the false-isolate rate counts benign samples that were incorrectly gated. All results were collected under 1000 fault injections with the same cycle budget $T_{\text{cycle}} = 200\,\mu\text{s}$. As anticipated, HW sustains $\tau_{\text{iso}} < 0.25\,T_{\text{cycle}}$ with the lowest false-isolate rate, while HYB sits between HW and SW. Table 5 reports the latencies and the false-isolate rate for each variant.

Table 5: Latency and false isolate rate under 1000 faults.

Variant	Median latency	95th percentile	False isolate rate
SW	190 µs	240 µs	7.2×10^{-4}
HYB	86 µs	106 µs	1.1×10^{-4}
HW	38 µs	42 µs	$< 5 \times 10^{-5}$

The SW/HYB 95th-percentile tails were dominated by ISR bursts and cache effects; HW eliminates these by a fixed-time branch-bounded path.

Timing model. All variants share the same cycle duration $T_{\text{cycle}} = 200\,\mu\text{s}$. Let τ_{iso} denote the isolation latency from sample arrival to enforcement decision:

$$\tau_{\text{iso}}^{\text{SW}} > \tau_{\text{iso}}^{\text{HYB}} > \tau_{\text{iso}}^{\text{HW}} \quad \text{with} \quad \tau_{\text{iso}}^{\text{HW}} < 0.25 T_{\text{cycle}}.$$

Interpretation. The HW variant enforces trust decisions within the same control cycle and without jitter from the task. HYB reduces latency but inherits delays from the software loop. SW experiences variable enforcement delay and occasionally misses the deadline.

Relation to redundancy logic. The baseline schemes (SW, HYB, HW) implement per-sensor filtering but do not include the redundancy-aware freeze condition introduced in Section 4. In those designs, mutually inconsistent sensors continue to degrade each other, whereas our hardware-timed evaluator applies the consensus threshold rule (Figure 7) to suspend updates until agreement resumes. This difference explains part of the false-isolate reduction in Table 5.

6 Security Analysis

Bypass attempts. Clock-glitch and power-glitch attacks were applied using a ChipWhisperer CW308. Across 5000 pulses in the range $\{-150\,\text{mV}, +150\,\text{mV}\}$ and widths $\{5\,\text{ns}, 80\,\text{ns}\}$, no instance of false increment or premature $G_i(t) = 1$ transition was observed. It tolerates single-cycle supply perturbations up to these limits.

Replay and cloning. Each controller uses a 64-bit eFuse device identifier read once at boot; the value is never placed on the field bus. Channel–sensor binding is enforced at board and cabling level during provisioning and stored as a static map between input port and identifier. A bus-level adversary therefore cannot replay or broadcast the identifier itself, but could still inject measurement values if able to drive the line with valid timing and maintain consistency with peer sensors. Timing-regularity checks

$$|t_{\text{ADC}}^{\text{recv}} - t_{\text{ADC}}^{\text{nominal}}| < \delta_{\text{sync}} = 4\,\mu\text{s}$$

reduce this attack surface, though they do not prevent analog substitution that perfectly mimics both timing and value statistics. This residual case is excluded under the stated non-goals in Section 2.

Denial-of-service on gating lines. A stuck-at fault on the gate-enable pin would bypass enforcement. To mitigate, the design supports dual-rail control:

$$x_i(t) \in \hat{x}(t) \iff \text{ENABLE\_P} = 1 \ \wedge \ \text{ENABLE\_N} = 0.$$

Both lines are driven by complementary logic within the same GPIO matrix, removing single-point failure paths.

Residual risk. The architecture does not resist physical compromise involving reflashing, depackaging or probe-level extraction, which are out of scope per the board-level denial model in Section 2.

Credential life-cycle. Sensor replacement invokes a one-time key derivation routine triggered by a recessed button press. The new 64-bit digest is written to an unused eFuse word; the previous entry is invalidated in hardware. No flash sectors store identity material, eliminating the need for runtime erase procedures.

7 Conclusion

This work presented a secure-by-design architecture for real-time IoT control systems with cycle-bound trust enforcement at the physical sensing layer. The enforcement logic runs as a hardware-timed, non-preemptive ESP32 firmware path with post-ADC gating; selected blocks were built in hardware but did not produce the reported metrics. Timing behavior, trust degradation and recovery under controlled fault injection were evaluated on an ESP32 prototype with post-ADC gating and selected enforcement functions were realized directly in hardware. In both cases, faults triggered bounded trust degradation, deterministic gating and automatic reintegration without rollback. Fusion behavior remained consistent under isolation and trust alignment enforced in real time. All enforcement was completed within the same control cycle and remained stable under fault and attack conditions.

The evaluation was conducted on a real-time testbed using cycle-synchronous fault injection under predefined intervals. The testbed enforces timing constraints and replicates hardware execution, but does not capture electrical noise, signal integrity degradation or cross-domain interference that may arise in larger deployments. The implementation employs two flip-flop synchronizers for ADC→PRO_CPU and PRO_CPU→GPIO crossings and budgets T_{sync} in Δ_{trust}, longer-term clock drift was not evaluated. Recovery logic operated on fixed hold periods without adaptation to extended disagreement or delayed convergence. All trust scoring parameters were static and selected empirically. Future work may investigate coordinated trust degradation across multiple quantities and evaluate the system ability to detect and respond to multi-vector attacks in redundant sensing environments.

Acknowledgments

This research was funded in part by CHIST-ERA-22-SPiDDS-07 (TROCI Project) and Austrian Science Fund (FWF) [10.55776/I6647].

References

[1] Gehad I. Alkady, Ramez M. Daoud, Hassanein H. Amer, Malak Y. ElSalamouny, and Ihab Adly. 2017. Fault-Tolerant FPGA-based Controllers in Factory Automation. In *2017 6th Mediterranean Conference on Embedded Computing (MECO)*. 1–6.

[2] Industrial automation and control systems security. 2013. IEC 62443-3-3: System security requirements and security levels.

[3] Kun Ji, Yan Lu, Linxia Liao, Zhen Song, and Dong Wei. 2011. Prognostics Enabled Resilient Control for Model-based Building Automation Systems. In *Proceedings of Building Simulation 2011*. 286–288.

[4] Ebrahim Karami, Francis M. Bui, and Ha H. Nguyen. 2012. Multisensor Data Fusion for Water Quality Monitoring Using Wireless Sensor Networks. In *2012 IEEE International Conference on Communications (ICC)*. IEEE, 1–5.

[5] Takeshi Fujino Kota Yoshida, Kuniyasu Suzaki. 2021. Towards Trusted IoT Sensing Systems: Implementing PUF as Secure Key Generator for Root of Trust and Message Authentication Code. In *Proceedings of the 10th International Workshop on Hardware and Architectural Support for Security and Privacy (HASP'21)*. ACM. doi:10.1145/3505253.3505258

[6] Aron Laszka, Waseem Abbas, Yevgeniy Vorobeychik, and Xenofon Koutsoukos. 2018. Synergistic Security for the Industrial Internet of Things: Integrating Redundancy, Diversity, and Hardening. In *2018 IEEE International Conference on Industrial Internet (ICII)*. 218–227.

[7] Aron Laszka, Waseem Toori Abbas, Xenofon Koutsoukos, and Yevgeniy Vorobeychik. 2019. Integrating redundancy, diversity, and hardening to improve security of industrial internet of things. In *Cyber-Physical Systems*. 1–27.

[8] Shengyu Li, Qingqing Liu, and Dashun Zhang. 2020. Authentication in Distributed Systems: Theory and Practice. In *International Conference on Information Technology and Systems*. 1–12.

[9] Shengyu Li, Fanjun Meng, Dashun Zhang, Qingqing Liu, Li Lu, and Yalan Ye. 2021. Research on Security Defense System of Industrial Control Network. In *2021 IEEE 2nd International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBA)*. 631–637.

[10] Zhiang Li, Daisuke Mashima, Wen Shei Ong, Ertem Esiner, Zbigniew Kalbarczyk, and Ee-Chien Chang. 2024. On Practicality of Using ARM TrustZone Trusted Execution Environment for Securing Programmable Logic Controllers. In *arXiv preprint arXiv:2403.05448*.

[11] B. Nandhini, S. Karthika, K. Krishnaveni, and S. Karthiyayini. 2023. Sensor Fusion Techniques for Enhanced Environmental Monitoring in IoT. In *2023 International Conference on Sustainable Communication Networks and Applications (ICSCNA)*. 445–450.

[12] Matthias Niedermaier, Dominik Merli, and Georg Sigl. 2019. A Secure Dual-MCU Architecture for Robust Communication of IIoT Devices. In *2019 8th Mediterranean Conference on Embedded Computing (MECO)*. 1–4.

[13] Olugbenga Kayode Ogidan, Moses Olla, and Isaac Odey. 2022. Water Distribution Control with Real-Time Monitoring. In *2022 IEEE Nigeria 4th International Conference on Disruptive Technologies for Sustainable Development (NIGERCON)*. 1–6.

[14] Miroslav Pajic, Nicola Bezzo, Rajeev Alur James Weimer, Rahul Mangharam, Nathan Michael, George J. Pappas, Oleg Sokolsky, Paulo Tabuada, Stephanie Weirich, and Insup Lee. 2013. Towards Synthesis of Platform-Aware Attack-Resilient Control Systems. In *Conference Paper, HiCoNS13*. 75–76.

[15] Craig G. Rieger, David I. Gertman, and Miles A. McQueen. 2009. Resilient Control Systems: Next Generation Design Research. In *2009 2nd Conference on Human System Interactions*. 632–639.

[16] Keith Stouffer, Michael Pease, CheeYee Tang, Victoria Pillitteri Timothy Zimmerman, Suzanne Lightman, Adam Hahn, Stephanie Saravia, Aslam Sherule, and Michael Thompson. 2023. *Guide to Operational Technology (OT) Security*. Technical Report NIST SP 800-82 Rev. 3. National Institute of Standards and Technology.

[17] Naing Win Tun and Asahiro Mambo. 2024. Secure PUF-Based Authentication Systems. *Sensors* 24, 16 (2024), 5295. doi:10.3390/s24165295

[18] Saira Varghese, Akshara Vinod, Sreela Sreedhar, T. N. Nafla, Janaki S. Nair, and Sesha Sofia Thomas. 2023. Portable Framework for Real-Time Water Quality Assessment and Analytics. In *2023 4th International Conference for Emerging Technology (INCET)*. 1–5.

[19] Ya Yu, Qiucheng Lu, and Yusun Fu. 2024. Dynamic Trust Management for the Edge Devices in Industrial Internet. In *IEEE Internet of Things Journal*, Vol. 11. 18410–18427.