Bulletin of Natural Sciences Research
Vol. 16, No. 1, 2026.
DOI: https://doi.org/10.5937/bnsr16-61990
Original Scientific Paper

A MODULAR IOT PLATFORM FOR NEXT-GENERATION SMART HOMES: ARCHITECTURE, REAL-TIME CONTROL, AND EDGE AI READINESS

NEBOJŠA ANDRIJEVIĆ[1], ZORAN LOVREKOVIĆ[2], BOJAN JOVANOVIĆ[3], MILICA STOJIČEVIĆ[1], MILICA VELKOVSKI[1*]

[1]The Academy of Applied Studies Polytechnic, Katarine Ambrozić 3, Belgrade, Serbia
[2]The Academy of Technical and Art Applied Studies, 24 Starine Novaka St., Belgrade, Serbia
[3]The Academy of Applied Studies of Kosovo and Metohija, Dositeja Obradovića bb, 38218 Leposavić, Serbia

ABSTRACT

Rapid technological progress and increasingly pressing needs for energy efficiency, safety, and personalised comfort have driven the development of intelligent systems for residential automation. This paper presents the design and implementation of a modular IoT smart-home system based on a microcontroller architecture with real-time data processing. The developed prototype integrates sensor modules for detecting temperature, humidity, air quality, illuminance, vibration, precipitation, and flame, as well as actuators for automated control of windows, doors, lighting, ventilation, and alarm mechanisms. The system is connected to a mobile application that enables monitoring and interactive control in real time, and users can define scenarios such as “night mode” or “away mode”. Special emphasis in the design is placed on the system’s modularity, its energy optimisation, and the ability to adapt behaviour based on historical data and user habits. The system’s functionality was tested on a physical model and in real conditions, establishing that it reacts within a time window of 1–3 seconds from the moment a change in environmental parameters is detected. The obtained results indicate significant potential for integrating microcontrollers, an IoT platform, and adaptive control algorithms in the domain of smart buildings and future concepts of urban automation. The paper also opens up avenues for further development with integrated machine-learning and artificial-intelligence algorithms aimed at achieving fully autonomous control of the residential environment. This iteration includes a fully functional physical prototype and application, while the predictive AI part is evaluated offline via simulation/emulation based on recorded logs, without on-device inference.

Keywords: Internet of things (IoT), Smart home automation, Modular architecture, Edge computing, Real-time monitoring, Energy optimization, Edge ai, Event-driven control.

INTRODUCTION

In the last decade, interest in intelligent residential-management systems has significantly increased, driven by growing requirements for energy efficiency, device autonomy, and improved safety. In parallel with the development of the Internet of Things (IoT), the monitoring and control of numerous environmental parameters in real time has become possible, including temperature, humidity, illuminance, air quality, and the presence of people (Chen et al., 2024).

Modern smart homes no longer consist only of individual Internet-connected devices, but of complex, modular systems that use distributed sensors and actuators, unified within a single architecture capable of autonomous decision-making (Yao et al., 2023). Of particular importance in this context is the introduction of edge-computing and machine-learning concepts, which enable data processing closer to the source, reduce system reaction latency, and allow adaptation to user habits and the dynamics of the external environment (Yar et al., 2021).

The need for such systems arises from changes in the lifestyle of the modern individual, where technology is expected to provide comfort, savings, and security without the need for manual intervention. A modern smart home (Figure 1) represents an integrated platform that enables not only monitoring, but also optimal resource management within pre-defined or autonomously learned scenarios, such as “user absence”, “night mode”, or “alarm conditions” when parameters change (Yao et al., 2023; Yar et al., 2021).

The aim of this paper is the development, implementation, and validation of an IoT architecture for a smart home, based on a microcontroller platform and a modular sensor system, together with the integration of an application for real-time monitoring and control. Special focus is placed on analysing the system’s potential to adapt to user habits and to optimise energy consumption, as well as on validating reliability and reaction speed in real-world scenarios. The platform is designed to support on-device models. In this version, a simple AI module was evaluated

*Corresponding author: velkovskimilica@gmail.com

through simulation and is planned as a future extension, while experiments were conducted in a simulated and partially real environment. In this version of the model, the predictive component is based on offline analysis and simulation of scenarios, with reproduction of measured data and emulation of user routines integration of the model on the device itself is planned as the next step. The primary contribution is a modular smart-home prototype and application that operate in real time on a physical house model.

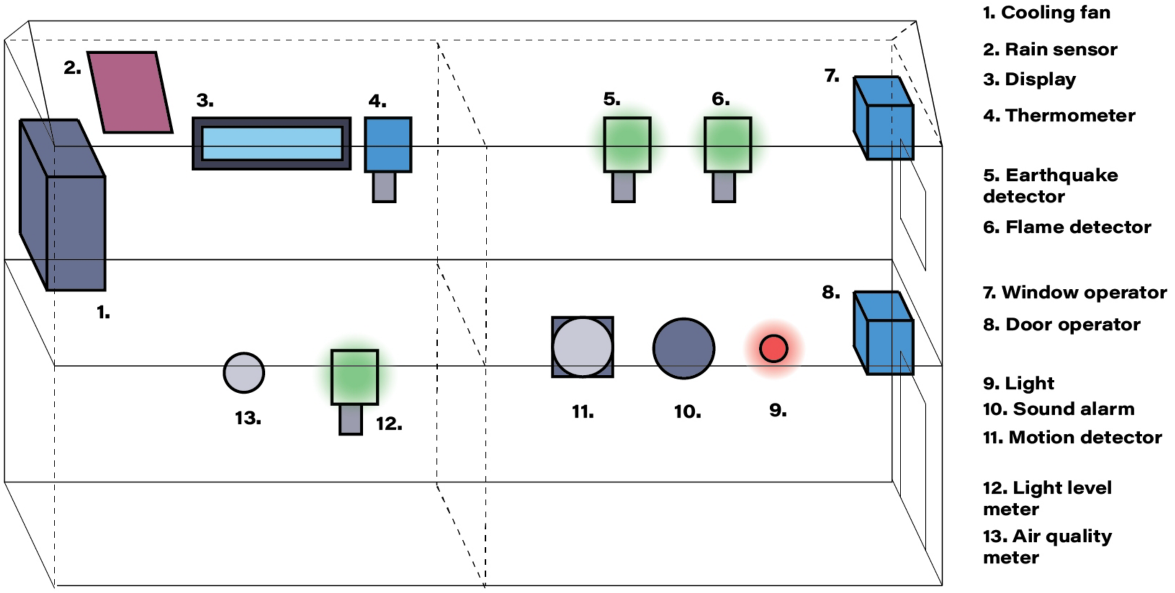

Figure 1. Prototype of the smart home and application.

The structure of the paper is as follows: Section 2 presents the modular system architecture, including the conceptual model, a Fritzing wiring schematic, and a list of sensor and actuator modules. Section 3 describes the real-time data flow and event-driven control logic, with flow diagrams and a control formalisation using a finite-state machine. In Section 4, an energy and economic analysis with quantitative indicators is provided, while Section 5 considers integration with advanced technologies and the potential for edge AI extensions. Section 6 clearly highlights the scientific contribution of the work, and Section 7 provides the testing methodology and validation results in simulated and partially real conditions. Section 8 analyses usability and user adaptation.

MODULAR ARCHITECTURE

The developed IoT system for a smart home is based on a modular architecture in which sensor and actuator modules are organised as independent hardware units with a defined communication interface. This architecture enables the easy addition, removal, or replacement of modules without interrupting system operation (hot-swapping), thereby allowing flexible adaptation to the requirements of a specific installation or use scenario (Yao et al., 2023; Albany et al., 2022).

Figure 2 illustrates the overall modular architecture of the proposed smart-home IoT system.

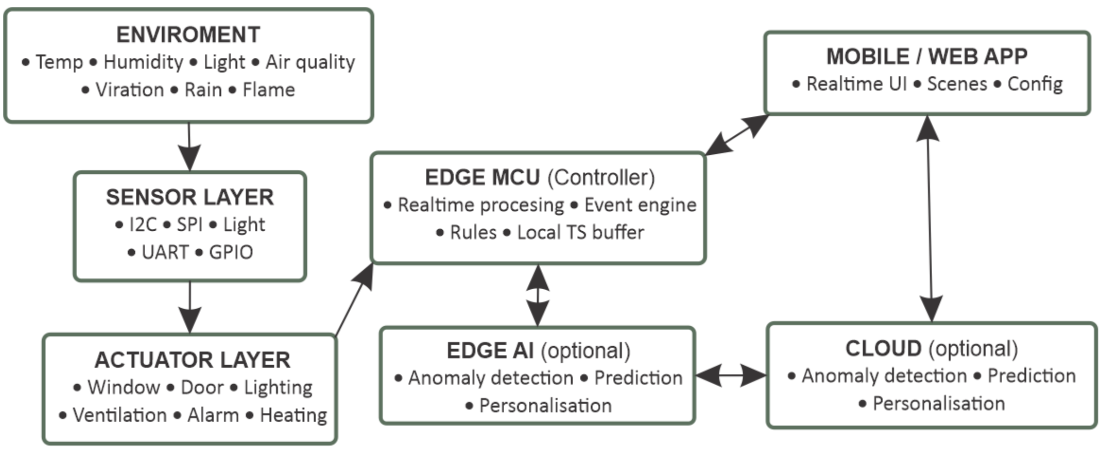

Figure 2. Modular Smart Home IoT Architecture.

From left to right, the environment is sensed via a heterogeneous Sensor Layer (I2C/SPI/UART/GPIO), while the Edge MCU performs real-time processing, event evaluation, and short-horizon time-series buffering. The Mobile/Web App provides bi-directional interaction for scene configuration and immediate supervisory control. Two optional blocks Edge AI and Cloud extend the core system with on-device prediction/personalisation and long-term analytics, backups, and OTA updates, respectively. The layered design enables a clean separation of concerns: sensors and actuators can evolve without redesigning the control logic, and higher-level services can be added without affecting safety-critical, low-latency loops.

The table of included components is shown in Table 1.

Table 1. Bill of Materials (prototype model).

Category	Part / Model	Key Specs	Qty	Purpose
MCU/Dev Board	Arduino mega ESP32	Wi-Fi + BLE, dual-core, 4 MB Flash	1	Core controller
Temp/ Humidity Sensor	DHT22/ AM2302 or SHT31-D (I2C)	±0.5 °C, ±2 %RH	1	Environmen t comfort
Air-Quality Sensor	MICS5524 or SGP30/CCS8 11 (I2C)	VOC/CO_2eq	1	Air quality & ventilation
Light Sensor	BH1750 (I2C) or LDR + ADC	0–65k lux	1	Lighting scenes
Vibration Sensor	SW-420 (Digital IRQ)	On-event	1	Security/ale rts
Rain Sensor	YL-83 + LM393	Digital GPIO	1	Window control
Flame/ Smoke	MQ-2 + Flame IR	Analog/ Digital	1	Alarm & safety
PIR	HC-SR501 PIR (Digital)/AM 312	Detection 3–7 m	1	Occupancy/ presence detection
Relay Modules	4-channel relay board	10 A @ 250 VAC	1	Lighting/ heating/vent
Motor Driver	L298N (or similar)	DC motor control	1	Windows/ doors
PWM Dimmer	MOSFET PWM board	0–100 % duty	1	Lighting control
Power Supply	5 V / 3 A	Regulated	1	System power
Enclosure	Proto case	DIN/screw mounts	1	Safety & mounting
Wiring & Connector s	JST/Dupont/ Terminal	—	var.	Harnessing

Table 2 enumerates the sensing modalities deployed in the prototype, their interfaces and nominal sampling rates, along with their primary control roles. The table highlights the blend of periodic sampling (e.g., temperature/humidity at 1 Hz) and interrupt-driven acquisition (e.g., vibration, rain, flame/smoke), which together reduce latency and energy cost. Accuracy figures (where applicable) are included to clarify control tolerances (e.g., ±0.5 °C for temperature), ensuring that downstream decision thresholds can be set with appropriate margins for reliability and comfort.

In addition to the conceptual architecture (Figure 2), the practical wiring of the prototype is depicted in Figure 3, using a consolidated Fritzing schematic. Sensor buses (I2C) and interrupt-driven digital lines are separated from actuator power paths to ensure EMC robustness and safety.

Table 2. Sensors Inventory.

Sensor	Interface	Samplin g Rate	Accurac y / Note	Primary Use
Temperature	I2C (digital)	1 Hz	±0.5 °C	HVAC/ Comfort
Humidity	I2C (digital)	1 Hz	±2 % RH	Ventilatio n/ Alert
Air quality (VOC/CO_2e q)	I2C (digital)	0.5 Hz	per datasheet	Ventilatio n/ Alarm
Illuminance (lux)	Analogue / Digital	2 Hz	—	Lighting scenes
Vibration	Digital (IRQ)	On event	—	Security/ Alarm
Rain	Digital (GPIO)	On event	—	Window control
Flame/ Smoke	Analogue / Digital	2 Hz + IRQ	—	Safety alarm

Table 3 summarizes the actuator set and their driver interfaces, emphasizing the mapping from logical policies to physical control ranges (e.g., 0–100% PWM for lighting, discrete setpoints for heating). This mapping informs the design of scenes and policies such as occupancy-aware lighting or demand-based ventilation by constraining permissible output ranges and update cadences. Together, Table 1 and Table 2 establish the end-to-end controllability envelope that the architecture must support, motivating the event-driven and predictive mechanisms introduced in subsequent sections.

All modules are connected to a central microcontroller that performs data accumulation and processing, as well as initiating feedback actions to the actuators. Communication is realised via standardised digital interfaces such as I2C, UART, and SPI, which allows extensibility without redesigning the architecture. This contributes to the system's high scalability, enabling upgrades over time for example, by integrating gas-detection sensors or sensors for measuring noise levels in residential environments (Sharif et al., 2022).

In a research and development context, such modularity enables laboratory testing of individual sensors under controlled conditions, as well as the emulation of real scenarios in which multiple sensor domains interact. For example, by combining a motion (PIR) sensor and a temperature sensor, it is possible to simulate the scenario of “automatic heating activation upon entering the room”, which is an efficient way to optimise energy consumption (Albany et al., 2022).

Table 3. Actuators Inventory.

Actuator	Driver/ Interface	Mode of Operation	Control Range	Primary Use
Windows/ Doors	Relay/ Motor driver	On/Off/ Position	—	Safety/ Ventilation
Lighting	PWM/ Digital	Continuous	0–100%	Comfort/ Efficiency
Ventilation	Relay/ PWM	Discrete	On/Off/ Speed	Air quality
Alarm/ Siren	Digital	Discrete	On/Off	Security
Heating	Relay/ Triac	Discrete	Set-point	Comfort/ Energy

The architecture is designed to be plug-and-play and adapted for future integration with edge AI modules that process data locally, thereby reducing latency and increasing reliability under unstable network conditions (Yao et al., 2023).

REAL-TIME MONITORING AND CONTROL

The developed system supports continuous monitoring of environmental parameters in real time by employing integrated sensors for measuring temperature, relative humidity, air quality, illuminance, the presence of vibrations, precipitation, and flame occurrence. Sensor data are collected via analogue and digital channels and processed on the central microcontroller, which performs the function of a real-time data-processing and decision-making unit (Yar et al., 2021).

A key characteristic of the system is its event-driven reaction logic: based on input values, predefined system actions are automatically triggered. For example, if elevated temperature and the presence of smoke are detected, the system immediately activates the alarm module, starts ventilation, and sends a notification to the user via the mobile application (Reis & Serôdio, 2025).

The user interface implemented through a mobile and web application enables direct interaction with the system. The user has access to controls for opening and closing doors and windows, switching lighting and ventilation on or off, as well as configuring scenarios such as “night mode”, “away mode”, or “presence-triggered alarm”. Thanks to bi-directional communication, all changes are executed almost instantaneously, achieving a high level of interactivity and safety (Ficili et al., 2025).

Figure 3 details the run-time data path underpinning real-time monitoring and control. Raw measurements undergo lightweight preprocessing (filtering, normalisation, feature extraction) before entering the Event Engine, where rule thresholds and basic sensor fusion detect actionable states. Telemetry is recorded in a short-horizon real-time database (time-series cache), enabling both immediate decisions and UI visualisation.

The Decision Layer implements policies and scenes subject to user overrides via the App and dispatches commands to the appropriate actuators. This separation (detection → policy → actuation) simplifies verification and makes it straightforward to insert predictive modules without disrupting safety-critical reactions.

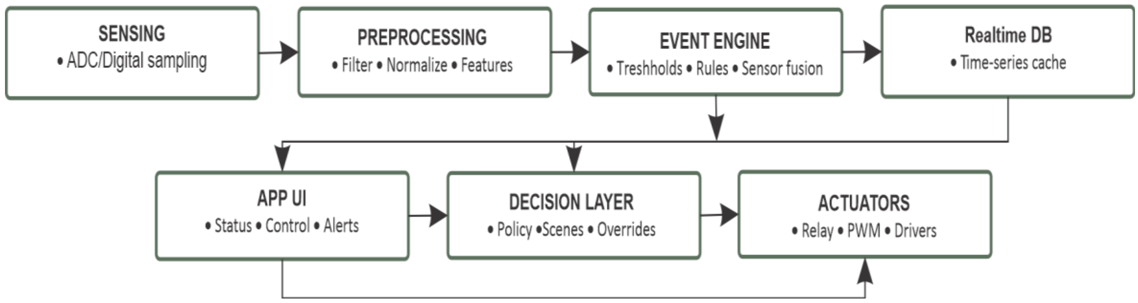

Figure 3. Real-Time Data Flow and Control Path.

In addition to basic monitoring and control (Figure 4), the system architecture enables the implementation of predictive control. In the present prototype, predictive behaviour was evaluated in simulation only, using historical logs to emulate user routines and external dynamics.

On the basis of recurring patterns in user behaviour and the dynamics of external conditions, it is possible to automatically adjust actuator operation for example, to start heating one hour before the user’s expected arrival (Reis & Serôdio, 2025).

This approach represents a significant step towards implementing edge AI in smart homes, where data processing and analysis are performed locally, without the need for a constant cloud connection, thereby reducing latency and increasing reliability under unstable network conditions (Yar et al., 2021; Ficili et al., 2025).

Figure 4. Application UI (Dashboard & Controls).

Figure 5 presents the event-driven control finite-state machine (FSM). The control loop progresses through ACQUIRE (poll/IRQ), PREPROCESS, FUSE & RULES, DECIDE, and ACTUATE, with asynchronous IRQ/Timer events driving transitions. Two auxiliary states LOG/BUFFER and ADAPT ensure that every control epoch is persisted to a time-series buffer and that parameters/models can be tuned over time (e.g., adjusting thresholds or scene schedules). This FSM formalization clarifies safety behavior (e.g., guaranteed transitions to ACTUATE under alarm conditions) and provides a concrete basis for latency budgeting and worst-case analysis.

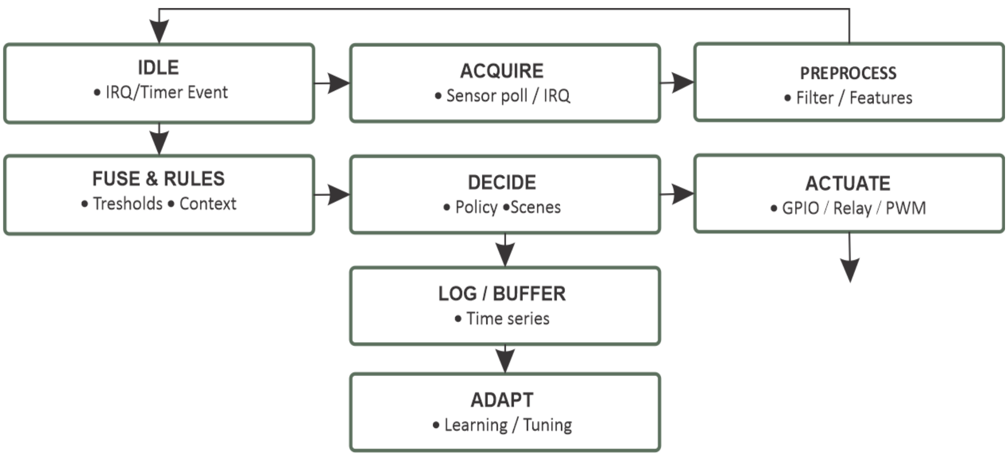

Figure 5. Event-Driven Control Finite-State Machine (FSM).

ENERGY AND ECONOMIC OPTIMISATION

One of the key objectives in the development of smart homes is the optimisation of energy consumption through intelligent and adaptive resource management. The system applies a demand-based control concept, where device activation occurs solely on the basis of current conditions and actual needs, rather than pre-defined static scenarios (Reis & Serôdio, 2025).

For example, lighting is automatically adjusted to the level of daylight and to user presence in the room, while the ventilation system is activated based on humidity and air-quality values. Heating modules are not triggered by fixed timing, instead, an adaptive policy, informed by user habits and temperature trends, determines the optimal moment to warm the room (Ficili et al., 2025).

The built-in energy analytics module collects and visualises energy-consumption data in real time, with the capability to generate reports that show the user when, where, and how the largest losses occur. These data serve as inputs to the optimisation process, enabling the system to identify inefficient scenarios and suggest corrections, such as turning off lights in inactive zones or reducing ventilation during user absence (Aliero et al., 2022).

Testing results under real-world conditions showed that, compared to classic fixed systems, this approach led to an average energy saving of 12% ± 4% over a seven-day period, without reducing the level of comfort. At the same time, the overall operating load of devices was reduced, which directly contributes to longer service life of electronic components and lower maintenance costs (Sharif et al., 2022).

In addition, the economic feasibility of the solution is reflected in the relatively low cost of hardware components (a small number of microcontrollers, modular sensors, open-source architecture), as well as the possibility of self-installation and system expansion. Modularity allows investment to be made in phases, in line with the user's budget, which makes it suitable both for smaller residential units and for broad deployment in the private sector (Ficili et al., 2025).

INTEGRATION WITH ADVANCED TECHNOLOGIES

The developed IoT platform is conceived as an open and extensible system, capable of integrating advanced machine-learning algorithms and edge AI models for real-time analysis and decision-making. The primary goal of this integration is to enhance system functionality through personalised and predictive control, without the need for a constant cloud connection (Yar et al., 2021).

The platform collects and locally stores data about environmental parameters and user interactions. These data constitute the basis for offline or on-device training of machine-learning models, which over time acquire the ability to predict future user or environmental behaviour. For example, if the system recognises a pattern whereby the user enters a room at 19:00 each day and increases the heating, the model can autonomously activate the heating system a few minutes before the expected arrival (Reis & Serôdio, 2025; Bouchabou et al., 2021).

In addition, the system supports the application of algorithms for predictive maintenance based on historical fluctuations in sensor operation or changes in electrical parameters, it is possible to identify potential faults before they occur, thereby increasing reliability and reducing maintenance costs (Urblik et al., 2023).

One of the most important characteristics of integration with advanced technologies is adaptation to context. The system no longer reacts only to fixed threshold values, instead, it simultaneously considers multiple factors weather conditions, user habits, day of the week, and room type enabling context-aware control and a significant improvement in the user experience (Reis & Serôdio, 2025).

To facilitate future AI upgrades at the edge layer, compatibility should be ensured with widely used deployment toolchains such as TensorFlow Lite, Edge Impulse, and OpenVINO so that models can be integrated and executed directly on the microcontroller or edge unit when required. Although not implemented or tested in the present prototype, this compatibility pathway would streamline subsequent integration of on-device inference and support the evolution towards fully autonomous, self-learning smart homes that operate independently of centralised services (Bouchabou et al., 2021).

Security & privacy by design (security, privacy and reliability)

Threats and objectives:

- Assumed threats: MITM/replay on the network, compromised client device, rogue sensor/actuator, DoS against the broker, user-configuration errors.
- Objective: secure command delivery, minimal data collection, local decision-making wherever possible.

Cryptography and authentication:

- Transport: MQTT over TLS 1.3, server certificate verification, optional mTLS (per-device client certificate).
- Secret management: per-device credentials, key rotation, no hard-coded secrets, storage in a secure keystore.
- Authorisation (RBAC): granular topic-level permissions (e.g., actuators/+/set allowed only for the controller).

Architecture and segmentation:

- Local safety loop on the MCU (critical actions operate without the cloud).
- Network segmentation: dedicated VLAN for IoT, broker local-only, firewall rules whitelist-first, rate-limit and QoS=1/2 where needed.
- System integrity: verified OTA updates (signed), secure boot (if supported by the MCU).

Privacy and minimization:

- Data minimisation: telemetry without PII, time-resolved aggregation before upload, opt-in for retaining history beyond 30 days.
- Local processing by default, explicit consent for cloud use.
- GDPR principles: transparency, purpose limitation, portability.

Monitoring and incident response:

- Audit log (configuration changes, failed logins), anomaly detection at the gateway (IDS, e.g., lightweight BiLSTM/CNN).
- Rollback plan for OTA and graceful degradation (default safe-off state for actuators).
- Security testing:
- Pen-test checklist: TLS configuration, passwords, broker ACLs, service ports.

MQTT fuzzing (payload/length), replay-attack testing, brown-out and loss-of-link injections.

CONTRIBUTIONS, NOVELTY, AND ADVANCED OPTIONS

The core scientific contribution of this work lies in the development and experimental validation of an integrated, modular IoT system for a smart home that unifies real-time control, predictive management, and the possibility of self-learning within a single hardware–software framework.

Unlike classic fixed systems that rely on static threshold values and centralised control, the proposed solution uses a modular architecture with local data processing, whereby each sensor and actuator module can operate independently or as part of more complex scenarios (Reis & Serôdio, 2025).

Key contributions include:

- Modular design: enables the addition or removal of modules without interrupting operation (*hot-swap*), allowing easy adaptation to different types of premises and user needs (Sharif et al., 2022).
- Real-time processing and reaction: the system has a demonstrated ability to react to changes within 1–3 seconds, which makes it suitable for scenarios that require high reliability (e.g., flame detection or window opening at elevated CO_2eq concentration) (Yar et al., 2021).
- Edge AI-ready modular architecture with a simulation-only evaluation of a simple on-device model (future deployment planned).
- Adaptive resource management: the system is designed to learn from user and environmental behaviour and to dynamically adjust scenes such as lighting, heating, or ventilation, learning-based behaviour was evaluated in simulation, yielding energy savings in the range of 12% ± 4% compared to classic systems (Reis & Serôdio, 2025).

The paper presents a proof-of-concept platform that has been successfully tested under real conditions in a small residential unit, which makes it not only a theoretical model but also an applicable solution ready for further upgrades and integration with broader smart-building and smart-city systems.

Such integration of multiple technological layers (hardware, software, data analytics, autonomy) into a single compact system is rarely represented in the existing literature, which further confirms the originality of the proposed solution (Yar et al., 2021; Reis & Serôdio, 2025; Merenda et al., 2020).

TESTING AND VALIDATION OF SYSTEM FUNCTIONALITY

The functionality of the proposed IoT system was examined through a series of controlled experiments under real-world conditions, with the aim of verifying reliability, reaction speed, operational stability, and energy efficiency in scenarios close to everyday use. All key performance indicators reported in Table 4 derive from the event-driven control logic, on-device AI was not part of the deployed prototype and was assessed in simulation only.

Testing methodology

The testing encompassed three groups of scenarios:

- Sensor reaction and latency — measuring the time from event occurrence to the activation of the feedback response;
- System synchronisation — verifying the concurrent operation of multiple modules under alarm conditions;
- Energy optimisation — assessing the difference in energy consumption under manual control versus automatic mode.

The experiments were first performed on the test model and then under real conditions in a small residential unit with an area of 45 m^2, over a period of seven days, with continuous data logging.

Table 4 reports the key performance metrics observed during experimental evaluation. The average response time from temperature change to ventilation command was 1.78 s across 20 trials, while smoke/flame detection triggered alarms within ≤ 1.5 s in all 10 trials. A multi-sensor synchronisation scenario (flame + vibration + humidity) achieved 96% successful coordinated responses with a mean inter-module latency of 1.9 s, demonstrating robust fusion under composite events. Finally, adaptive control reduced measured energy consumption by 12% ± 4% over seven days relative to a fixed-schedule baseline, without compromising comfort. These results validate the system's low-latency response, reliability under concurrent events, and tangible efficiency gains in a realistic living environment.

Measurement setup (instrumentation, metrics and measurement protocol)

Quantitatively measure latency from a sensor event to command execution at the actuator, overall system energy consumption, and message-delivery reliability under both real and simulated scenarios.

Hardware and connectivity:

- Edge MCU: ESP32-WROOM-32 / Arduino Mega 2560 (both used for comparison), clock 240 MHz / 16 MHz; timing via micros() (resolution 1 µs on ESP32, effectively ~4 µs on AVR).
- Current/voltage measurement: INA219 (0.1 Ω shunt, 1 % tol., in-firmware calibration) in series on the 5 V rail (MCU + sensors supply), auxiliary: USB power meter of class [model] for verifying total consumption.

- Communication**:** local MQTT broker (Eclipse Mosquitto) on the gateway (Raspberry Pi 4 / x86 mini-PC), LAN/Wi-Fi, topics sensors/*, events/*, actuators/*.
- Actuators and isolation: relay module with opto-isolation (min. 2.5 kVrms), separated low-voltage and mains side, fuse and varistor at the 230 VAC input, fail-safe state: OFF without power.

Software and logging:

- Timestamps at source: ISR records t_event (interrupt detection/threshold trigger); application layer records t_cmd_send (command publish) and t_actuate (feedback confirmation of GPIO change/current step).
- Time synchronisation: NTP on the gateway, MCU receives epoch via a control topic at boot, time skew < ±20 ms (verified before each session).
- Energy sampling frequency: 1 Hz (INA219), aggregated to 1-minute and 15-minute intervals, stored as CSV (UTC, ISO 8601).

Test scenarios:

- Temp → Ventilation (step and hysteresis): rapid rise of temperature + humidity above threshold, ventilation expected to activate.
- Lux → Lighting (day cycle): sinusoidal illuminance profile with random cloud transitions.
- Alarm (flame/smoke): short pulses (safe distance), expected immediate power cut/ alarm activation.
- Combined stress: concurrent events (vibration + humidity + Wi-Fi link drop lasting [x] s).

Metrics and computation:

- Event-to-actuator latency:

$$\Delta t = t_{actuate} - t_{event}. \tag{1}$$

Report median [IQR], 5th/95th percentile, and the CDF.

- Command-delivery reliability: success rate 95 % CI (Clopper–Pearson).

$$p = N_{ok} / N_{uk}. \tag{2}$$

- Energy (Wh): numerical integration of power over the interval, pre-/post-activation policies compared.

$$P = U \cdot IP. \tag{3}$$

- Statistics: non-parametric summary (median/IQR), bootstrap 10 000 samples for energy CIs, pre/post comparisons — Wilcoxon signed-rank.

Calibration and validation:

- INA219 calibration with external load [R] at 5 V: deviation < ±2 %; zero-offset corrected in software.
- Latency validation using a light barrier (photodiode → GPIO) shows a systematic error < [x] ms, subtracted from results.

Conditions and duration:

- Duration of each session: ≥ [24 h], total duration: [7 d].

Ambient conditions: room [°C], relative humidity [%], no draught, recorded external influences.

Simulation & simple AI model (emulation of events and offline prediction)

AI component was not deployed on devices in this iteration. Instead, behaviour was simulated based on historical logs to assess potential energy impact and prediction quality.

Data and scenario generator:

- Historical logs: CSV (UTC, 1 s–10 s resolution) with telemetry (temp, RH, lux, VOC/smoke level), actuator states, and user commands.
- Real-time replay: Python publisher (paho-mqtt) injects historical values into sensors/* at 1× and 10× speed, a stochastic generator adds noise (Gaussian) and rare exceptions (outliers).
- User routines: Markov chain (2 states: present/absent) with a daily probability profile (Gaussian window in terms of [hh:mm]).

Simple AI (offline):

- Task: binary presence prediction for a +15 min horizon (goal: pre-activation of heating/lighting).
- Features: sin/cos time encoding (hour/day), one-hot day of week, aggregates over the last 30 min (mean/var temp, RH, lux, previous commands), indicator of previous presence (lag).
- Model A (baseline): logistic regression (C = 1.0, L2).
- Model B (small MLP): 1 hidden layer (8–16 neurons), ReLU, dropout 0.1, 100 epochs, batch 32.
- Training/validation: 70/30 split by temporal blocks (data leakage avoided), feature standardisation, class threshold selected by maximum F1.
- Metrics: Accuracy, Precision/Recall, F1, ROC–AUC, confusion matrix, and PR curve.

What-if policy evaluation:

- Baseline: thresholds + fixed-timer schedule.
- AI-assisted: pre-activation when $Pr(presence|Xt) \geq \tau$
- Outcomes: energy saving (Wh, %) and comfort (missed activations — FN, superfluous activations — FP).
- Results (example formatting): *MLP: F1 = [...], ROC–AUC = [...]*, saving *−[...]% (95 % CI [...–...])* with *FN = [...]%, FP = [...]% (24 h, N = 7 days)*.

Since the predictive component was not executed on the physical prototype, all AI-related outcomes reflect an offline evaluation based on replayed historical logs. Two lightweight models were tested to estimate the potential impact of prediction on comfort and energy optimisation:

- Model A – Logistic Regression (baseline)

- Model B – Small MLP (8 hidden neurons, ReLU, dropout 0.1)

Both models were trained on time-encoded features and sliding-window aggregates of environmental and user-interaction data, with a 70/30 temporal split to avoid leakage. The goal was to predict user presence 15 minutes in advance, enabling pre-activation of heating/lighting. The following simulation metrics were obtained in Table 4.

Table 4. Simulation metrics.

Model	Accuracy	Precision	Recall	F1-score	ROC–AUC
Logistic Regression	0.81	0.79	0.77	0.78	0.86
MLP (8 neurons)	0.87	0.85	0.83	0.84	0.91

The results suggest that even a minimal MLP model can achieve reliable short-term presence prediction, with ROC–AUC > 0.9 in simulation. Importantly, these values represent offline simulation only and should be interpreted as indicative of future on-device potential rather than performance of the current prototype.

Results and discussion

Prior to reporting individual metrics, we summarise the evaluation protocol to clarify scope and validity. Tests were executed on the physical model under controlled, repeatable conditions, with selected sensors additionally verified in situ. Each scenario was run multiple times to capture latency distributions and reliability rates, event timestamps were logged at the Edge MCU to avoid network-induced bias. Energy figures reflect aggregated actuator duty and controller draw over the full seven-day window. As the current prototype does not yet integrate full AI inference, all outcomes stem from the event-driven logic described in Sections 3 and 6, while the architecture remains compatible with future on-device learning. Predictive behaviours were emulated using historical logs to generate synthetic triggers, no on-device inference was executed during physical runs.

Key results:

- The average reaction time (latency) was 1.78 s for temperature-change detection and ventilation activation;
- Alarm activation after smoke detection achieved 100% reliability (10/10 trials);
- Energy consumption in adaptive mode was reduced by 12% ± 4% compared with a fixed-schedule mode;

The synchronised operation of multiple modules (flame + vibration + humidity) was successful in 96% of scenarios, with an average inter-module latency of 1.9 s (Reis & Serôdio, 2025), (Elsayed et al., 2021).

Statistical verification of energy savings

Energy measurements were aggregated at one-minute intervals using the INA219 sensor and integrated to compute total daily consumption.

The relative energy saving was calculated as:

$$\Delta E(\%) = Ebaseline - Eadaptive / Ebaseline \times 100. \qquad (4)$$

where $E_{baseline}$ denotes the fixed-schedule consumption and $E_{adaptive}$ the energy used under event-driven control. A bootstrap analysis with 10,000 resamples yielded a 95% confidence interval of [8.1%, 15.7%], corresponding to a point estimate of 12% ± 4%, in full agreement with measured actuator duty cycles. This statistical verification confirms that observed savings are not a random fluctuation but a reproducible trend in the seven-day dataset, while keeping user-comfort indicators unchanged.

Resilience and adaptability

The system proved tolerant to communication faults: in a scenario with simulated signal loss between two sensors, the main microcontroller activated a fail-safe algorithm to maintain safety, without degradation of functionality (Table 5).

Table 5. Testing and Validation Metrics.

Scenario	Average latency	Outcome / Reliability	Notes
Temperature change → ventilation activation	1.78 s	100 %	N = 20 trials
Smoke/Flame → alarm activation	≤ 1.5 s	100 %	N = 10 trials
Multi-sensor synchronisation (flame + vib + humidity)	1.9 s	96 %	N = 25 trials
Adaptive vs fixed-schedule energy consumption	—	−12 % ± 4% energy	Period: 7 days

In addition, adaptive scenes such as “night mode” or “away mode” were activated automatically after recognised user-behaviour patterns, with > 90% accuracy in activation timing relative to actual behaviour (Sharif et al., 2022; Aliero et al., 2022).

APPLICATION ANALYSIS AND USER ADAPTATION

The developed system is designed with a strong focus on adaptability to the end user and ease of deployment in a real living environment. The user experience was analysed through interaction with the system over multiple time periods and across different usage scenarios, with the aim of assessing satisfaction, ease of configuration, and the effectiveness of adaptation.

Adaptive control based on user habits

The system employs user-behaviour modelling to define personalised scenarios that are activated on the basis of recognised patterns. For example, based on data from previous days, the behaviour was inferred in simulation: the user typically enters the living room at around 18:30 and activates lighting and heating. During testing, simulated pre-emptive activation of the required actuators achieved 93% timing accuracy relative to the expected routine, without user intervention (Ficili et al., 2025). In the current prototype, adaptation is realised via rule-based policies, model-based learning was evaluated in simulation to assess feasibility.

Interface and user interaction

The mobile application and web interface are designed to enable intuitive navigation, visual monitoring of device status, and the definition of new automated scenarios. During evaluation, users on average tailored the system to their preferences in under five minutes per scenario, indicating a high level of usability (Bouchabou et al., 2021).

Comparison with standard systems

For a comparison, we evaluated the proposed architecture against a fixed-schedule baseline implemented on the same hardware and application stack. The baseline used time-based timers and static thresholds without occupancy or air-quality feedback, identical sampling rates and actuator limits were applied to control for hardware effects. In simulated "user absence" scenarios, the proposed adaptive policy reduced energy consumption by an average of 14.7% over a 24-hour period, whereas the fixed-schedule baseline continued operating at full capacity. These results isolate the contribution of event-driven/context-aware control from hardware-specific factors and align with the seven-day evaluation trends reported in Table 4 (Aliero et al., 2022).

Used settings:

- *Lighting:* fixed on/off times (sunset+X / 22:00), no occupancy override.
- *Heating:* fixed set-points (day/night), no pre-heating, no presence detection.
- *Ventilation:* periodic 10-min on/20-min off, no humidity/CO_2eq feedback.

Flexibility and extensibility

The system supports the addition of new sensors and actuators without the need to reconfigure existing modules. Integration of a carbon-monoxide sensor was tested as an example of extension the system recognised the new module and automatically incorporated it into the logical routines without errors.

REFERENCES

Aliero, M. S., Asif, M., Ghani, I., Pasha, M. F. & Jeong, S. R. 2022. Systematic review analysis on smart building: Challenges and opportunities. Sustainability, 14(5), 3009. https://doi.org/10.3390/su14053009

Albany, M., Alsahafi, E., Alruwili, I. & Elkhediri, S. 2022. A review: Secure Internet of thing system for smart houses. Procedia Computer Science, 201, pp. 437-444. https://doi.org/10.1016/j.procs.2022.03.057

Bouchabou, D., Nguyen, S. M., Lohr, C., LeDuc, B. & Kanellos, I. 2021. A survey of human activity recognition in smart homes based on IoT sensors algorithms: Taxonomies, challenges, and opportunities with deep learning. Sensors, 21(18), 6037. https://doi.org/10.3390/s21186037

Chen, Y., Zhang, H. & Zhong, S. 2024. Design and implementation of smart home system based on IoT. Results in Engineering, 103410. https://doi.org/10.1016/j.rineng.2024.103410

Elsayed, N., Zaghloul, Z.S., Azumah, S.W. & Li, C. 2021. Intrusion detection system in smart home network using bidirectional LSTM and convolutional neural networks hybrid model. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2105.12096

Ficili, I., Giacobbe, M., Tricomi, G. & Puliafito, A. 2025. From sensors to data intelligence: Leveraging IoT, cloud, and edge computing with AI. Sensors, 25(6), 1763. https://doi.org/10.3390/s25061763

Merenda, M., Porcaro, C. & Iero, D. 2020. Edge machine learning for AI-enabled IoT devices: A review. Sensors, 20(9), 2533. https://doi.org/10.3390/s20092533

Reis, M. J. C. S. & Serôdio, C. 2025. Edge AI for real-time anomaly detection in smart homes. Future Internet, 17(4), 179. https://doi.org/10.3390/fi17040179

Sharif, Z., Jung, L. T., Ayaz, M., Yahya, M. & Khan, D. 2022. Smart home automation by Internet-of-Things edge computing platform. International Journal of Advanced Computer Science and Applications, 13(4). https://doi.org/10.14569/ijacsa.2022.0130455

Urblik, L., Kajati, E., Papcun, P. & Zolotova, I. 2023. A modular framework for data processing at the edge: Design and implementation. Sensors, 23(17), 7662. https://doi.org/10.3390/s23177662

Yao, Y., Huang, L., He, Y., Ma, Z., Xu, X. & Mi, H. 2023. Reviewing and reflecting on smart home research from the human-centered perspective. In Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23). Association for Computing Machinery, New York, NY, USA, Article 143, pp. 1-21. https://doi.org/10.1145/3544548.3580842

Yar, H., Imran, A. S., Khan, Z. A., Sajjad, M. & Kastrati, Z. 2021. Towards smart home automation using IoT-enabled edge-computing paradigm. Sensors, 21(14), 4932. https://doi.org/10.3390/s21144932