Agentic AI-Based HVAC and Lighting Control at the Edge Using Raspberry Pi

Abstract

Energy-efficient building operation requires intelligent control of Heating, Ventilation, and Air Conditioning (HVAC) systems and lighting, while simultaneously maintaining occupant comfort. This paper presents a real-time, zone-level cyber-physical prototype that implements Agentic AI principles for autonomous HVAC airflow and lighting control using low-cost edge hardware. The system integrates environmental sensing via a BME280 sensor, human presence detection using a PIR sensor, multi-agent decision coordination using LangGraph, and local large language model (LLM) reasoning via Phi-3 Mini deployed with Ollama. Unlike simulation-only studies, this work evaluates a physical deployment in an indoor environment, demonstrating continuous perception–decision–action loops operating entirely on-device. The prototype implements occupancy-based lighting control and temperature-threshold-based airflow control, while the Agentic AI layer provides structured reasoning, explainability, and extensibility toward predictive and learning-based control. Experimental observations validate reliable control behavior, low actuation latency, and stable operation, establishing a foundation for future intelligent Building Management Systems (BMS) using agentic architectures.

1. Introduction

Buildings represent one of the largest consumers of electrical energy worldwide, with heating, ventilation, and air-conditioning (HVAC) systems accounting for a substantial fraction of that demand. In both residential and commercial environments, HVAC operation is continuous, energy intensive, and highly sensitive to environmental conditions, occupancy patterns, and building structure. In parallel, lighting systems remain a persistent contributor to unnecessary energy usage when spaces are unoccupied or over-illuminated. Together, HVAC and lighting form the primary targets for energy optimization in smart building research and industrial Building Management Systems (BMS).

Traditional HVAC and lighting control mechanisms rely primarily on static schedules, thermostatic thresholds, and manual overrides. These approaches are effective only under predictable usage patterns and cannot respond efficiently to dynamic real-world behavior such as fluctuating occupancy, door openings, weather changes, or equipment heat loads. In large buildings, centralized supervisory systems coordinate multiple zone controllers, but the underlying zone-

level logic often remains rule-based and reactive. As a result, energy is frequently consumed even when thermal comfort requirements are already satisfied or when spaces are unoccupied.

Recent advancements in intelligent control systems seek to address these inefficiencies through data-driven and adaptive approaches. Research in Model Predictive Control (MPC) demonstrates that predictive models of building thermal dynamics can optimize HVAC operation by forecasting future conditions. However, MPC approaches depend heavily on accurate thermal models and require computationally intensive optimization, which becomes challenging in buildings with heterogeneous construction materials, variable occupancy, and unpredictable environmental disturbances. Maintaining and tuning these models for real-world deployment is costly and complex.

To overcome modeling limitations, learning-based approaches such as Deep Reinforcement Learning (DRL) have been proposed, in which HVAC control is formulated as a Markov Decision Process and policies are learned through repeated interaction with simulated building environments. DRL-based controllers can learn complex nonlinear relationships between environmental states and control actions, achieving significant energy savings in simulation studies. Nevertheless, such systems often require large training datasets, long convergence times, and reliable simulation environments such as EnergyPlus. Transferring learned policies from simulation to real buildings remains an open challenge due to modeling inaccuracies and sensor noise.

Alongside learning-based control, recent developments in artificial intelligence emphasize the concept of Agentic AI. Agentic AI systems are characterized by autonomous goal-driven behavior, modular reasoning, task decomposition, and continuous interaction with the environment. Rather than operating as monolithic decision functions, Agentic AI architectures are composed of multiple specialized agents responsible for perception, safety enforcement, optimization, planning, and explanation. These agents coordinate through structured execution frameworks that ensure predictable and interpretable behavior.

When Agentic AI is deployed at the edge, directly on local controllers close to physical systems, it enables low-latency response, resilience to network failures, and privacy-preserving data processing. Edge deployment is particularly important for safety-critical and time-sensitive applications such as HVAC control, where delays of even a few seconds can lead to discomfort, energy waste, or equipment stress. Industrial BMS architectures already rely on distributed edge controllers, such as Direct Digital Controllers (DDCs), that perform zone-level control independently while reporting to centralized supervisory platforms. Integrating Agentic AI into such edge controllers represents a natural evolution of building automation technology.

In parallel, the emergence of large language models (LLMs) and multimodal foundation models has introduced new capabilities for reasoning, explanation generation, and contextual interpretation of sensor data. While LLMs are not suitable for direct low-level control due to safety and determinism requirements, they offer strong potential for high-level reasoning, anomaly

detection, predictive analysis, and operator interaction. When combined with deterministic control agents, LLMs can enhance system transparency and support adaptive strategies without compromising safety.

Most existing research on generative or agentic AI for HVAC control has focused on either simulation environments or large-scale office deployments with extensive sensing infrastructure. Simulation-based studies allow controlled experimentation but do not capture real-world uncertainties such as sensor noise, hardware delays, and unpredictable human behavior. Large industrial deployments, while valuable, are costly and inaccessible for academic experimentation and student research.

There is therefore a strong need for experimental platforms that bridge the gap between theoretical intelligent control algorithms and real-world building automation. Such platforms must support real-time sensing, physical actuation, modular AI integration, and safe experimentation, while remaining affordable and easy to modify. Zone-level prototypes using low-cost hardware provide an ideal environment for exploring agent-based control strategies, validating cyber-physical integration, and developing scalable architectures for future deployment.

This project addresses this need by implementing a complete edge-based cyber-physical HVAC and lighting control system using a Raspberry Pi, environmental sensors, relay-actuated devices, and a multi-agent decision framework coordinated using LangGraph. The prototype represents a single building zone and mirrors the operational principles of professional BMS zone controllers. All sensing, reasoning, and actuation occur locally at the edge, without cloud dependency, enabling fast response and safe offline operation.

Unlike optimization-focused approaches, the primary objective of this prototype is to demonstrate structured autonomous control using Agentic AI principles, with clear separation between perception, safety enforcement, comfort evaluation, energy optimization, actuation, and explainability. The integration of a local LLM further enables reasoning over historical trends and supports future predictive extensions. Through real-time deployment and experimental observation, this work demonstrates how agent-based architectures can be practically implemented on embedded edge devices for intelligent building automation.

2. Related Work

Intelligent control of building HVAC and lighting systems has been an active research area for several decades. Prior work can be broadly categorized into model-based control approaches, learning-based optimization methods, and recent agentic and generative AI-driven systems. This section reviews major directions relevant to the proposed prototype and positions this work within existing research trends.

2.1 Model Predictive Control (MPC) for HVAC Systems

Model Predictive Control has been widely studied for HVAC optimization because it can explicitly incorporate system dynamics, constraints, and future predictions. MPC frameworks rely on mathematical models of building thermal behavior, often represented using resistance–capacitance (RC) networks or physics-based thermal equations. These models predict future temperature evolution based on current state, external weather conditions, and control inputs such as airflow or valve positions.

MPC-based HVAC controllers can optimize energy consumption while maintaining temperature within comfort bounds by solving constrained optimization problems at each control step. Studies have demonstrated that MPC can significantly reduce energy usage when accurate models are available and weather forecasts are reliable. However, real-world deployment remains challenging due to several factors: thermal models vary across buildings, internal heat gains from occupants and equipment are difficult to predict, and computational complexity increases rapidly for multi-zone buildings. Model calibration and maintenance also require expert knowledge, making MPC expensive to deploy at scale.

2.2 Reinforcement Learning and Deep Reinforcement Learning Approaches

To avoid dependence on explicit thermal models, reinforcement learning (RL) methods formulate HVAC control as a Markov Decision Process (MDP), where the controller learns policies by interacting with the environment. Classical RL methods such as Q-learning have been applied to HVAC control in limited state spaces but struggle with high-dimensional continuous environments.

Deep Reinforcement Learning (DRL) extends RL by using neural networks to approximate value functions or policies, enabling operation in complex continuous state spaces. DRL-based HVAC controllers are typically trained using detailed simulation environments such as EnergyPlus. These systems can learn nonlinear control strategies that outperform rule-based controllers in simulation, achieving significant reductions in energy cost while maintaining thermal comfort.

Despite promising simulation results, DRL-based HVAC control faces several challenges in practice. Training requires extensive interaction data, which is infeasible to collect directly from real buildings due to safety and comfort risks. Simulation-to-real transfer is difficult because simulated models cannot fully capture real-world uncertainties such as sensor noise, occupant

behavior, and equipment degradation. Additionally, DRL policies are often opaque and difficult to interpret, which raises concerns for safety-critical building infrastructure.

2.3 Occupancy-Based and Sensor-Driven Rule Systems

Commercial BMS deployments commonly implement occupancy-based control using PIR sensors, CO_2 sensors, or badge-based presence detection. These systems adjust airflow and lighting based on detected usage patterns, improving energy efficiency compared to fixed schedules. Rule-based strategies remain attractive due to their simplicity, predictability, and ease of certification for safety compliance.

However, purely rule-based systems lack adaptability to changing usage patterns, seasonal variations, and evolving building behavior. While thresholds and schedules can be manually tuned, such tuning does not scale well across large facilities with diverse zone characteristics. As a result, rule-based systems often operate conservatively, leading to suboptimal energy performance.

2.4 Agentic AI and Multi-Agent Architectures for Control

Agentic AI introduces a paradigm in which autonomous agents cooperate to achieve high-level goals through perception, reasoning, planning, and action. Rather than embedding all logic in a single controller, responsibilities are distributed across specialized agents such as safety agents, optimization agents, monitoring agents, and planning agents. Multi-agent architectures improve modularity, interpretability, and robustness of complex control systems.

Recent research and industrial white papers propose agent-based HVAC control systems in which local edge controllers host multiple agents responsible for zone-level optimization while coordinating with building-level supervisory agents. This hierarchical agent structure aligns naturally with existing BMS architectures, where zone controllers operate independently but report to central systems. Agent-based control enables flexible integration of learning modules, expert rules, and optimization strategies within a unified decision framework.

2.5 Generative AI and LLM-Based Control Reasoning

Large Language Models (LLMs) have recently been explored for control reasoning, simulation, and explanation tasks. In HVAC research, generative AI has been used to infer optimal setpoints, simulate building response, and generate control code. Some studies demonstrate that LLMs can approximate control strategies in simulation environments without explicit training, relying on their general reasoning capabilities.

However, LLMs are inherently probabilistic and lack formal safety guarantees, making them unsuitable for direct low-level actuation in safety-critical systems. As a result, emerging architectures separate deterministic control logic from generative reasoning modules. LLMs are used for explanation, anomaly detection, predictive analysis, and supervisory recommendations, while hard safety constraints remain enforced by rule-based or model-based controllers.

2.6 Positioning of the Proposed Prototype

The proposed prototype integrates concepts from multiple research directions while maintaining practical deployability. Similar to occupancy-based rule systems, it employs deterministic control rules for lighting and airflow to ensure safety and predictability. Inspired by agentic AI architectures, it decomposes control logic into modular agents coordinated using a structured execution graph. Unlike centralized cloud-based systems, all intelligence is deployed at the edge, reflecting real BMS zone controller operation.

Unlike DRL-based optimization systems, this prototype does not rely on simulation-based training or learned policies. Instead, it prioritizes real-time cyber-physical integration, explainability, and extensibility. The inclusion of a local LLM enables higher-level reasoning and future predictive extensions while remaining isolated from direct actuation pathways.

This hybrid design allows safe experimentation with agent-based control strategies in real environments and provides a foundation for gradual integration of learning-based optimization techniques without compromising operational reliability.

3. Prototype Design Philosophy and Objectives

3.1 Zone-Level Control Concept

Commercial buildings are typically divided into multiple thermal zones, each regulated by a Direct Digital Controller (DDC) that manages airflow, temperature, and sometimes lighting. These zone controllers operate under supervisory commands from a central BMS but perform fast local control based on sensor feedback.

In this prototype, the Raspberry Pi functions as a software-defined DDC for a single zone. It interfaces directly with sensors and actuators while executing local decision logic and AI-based reasoning modules. This design mirrors industrial BMS hierarchy at a smaller scale.

3.2 Design Objectives

The prototype is designed to achieve the following goals:

- Continuous real-time environmental monitoring
- Autonomous closed-loop control of lighting and airflow
- Modular agent-based decision structure
- Safe actuation with deterministic constraints
- Explainable AI reasoning for system transparency
- Expandability toward predictive and learning-based control

The system is intentionally kept simple in hardware while rich in software structure to allow experimentation with advanced AI control strategies without industrial safety risks.

4. System Architecture

4.1 Overall Physical Architecture

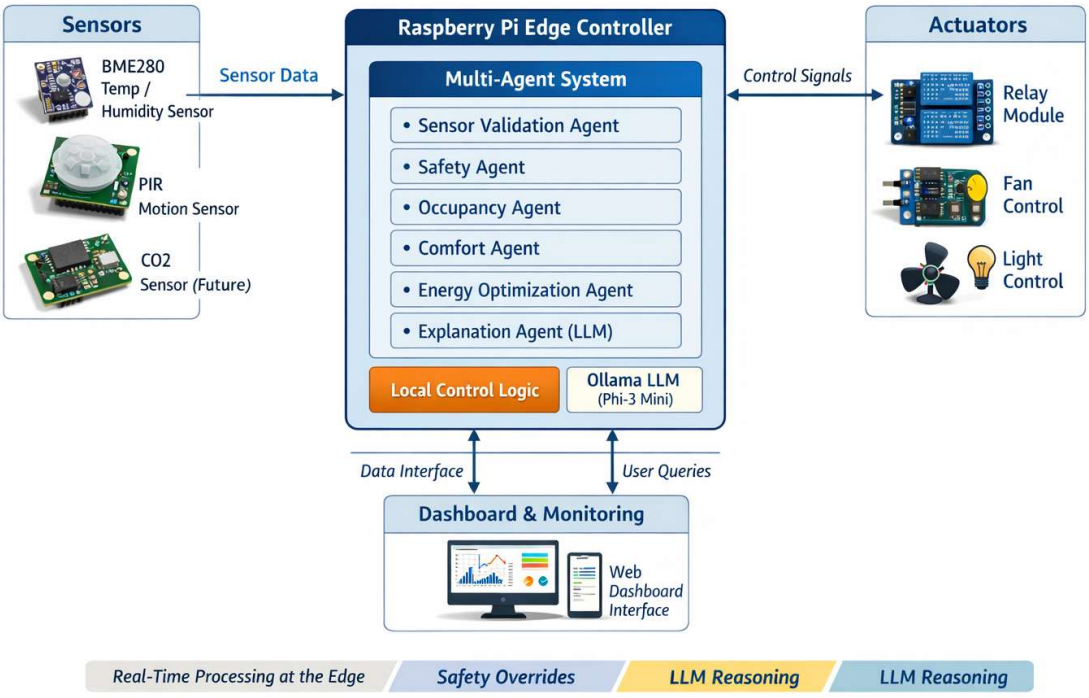


Figure 1: Overall System Architecture of *the Edge-Based Agentic HVAC and Lighting Control System*

Sensor Layer → Edge Controller (Raspberry Pi) → Multi-Agent Decision Layer → Relay Actuation → Physical Loads → Monitoring Dashboard → AI Reasoning Feedback

All sensing, decision-making, and actuation processes in the system are executed directly on the Raspberry Pi, which acts as the local edge controller. Sensor data from the BME280 and PIR sensors are processed immediately by the onboard software without being transmitted to any external server. The multi-agent decision logic, safety checks, and control algorithms also run entirely on the same device, and final control commands are sent directly to the relay modules through GPIO pins.

3.2 Hardware Architecture

Component	Function
Raspberry Pi 5	Edge controller and AI execution platform
BME280 Sensor	Measures temperature and humidity (I2C)
PIR Motion Sensor	Detects human presence (GPIO)
2-Channel Relay Module	Electrical isolation and switching
Fan	Represents HVAC airflow supply
Light Bulb	Represents lighting load

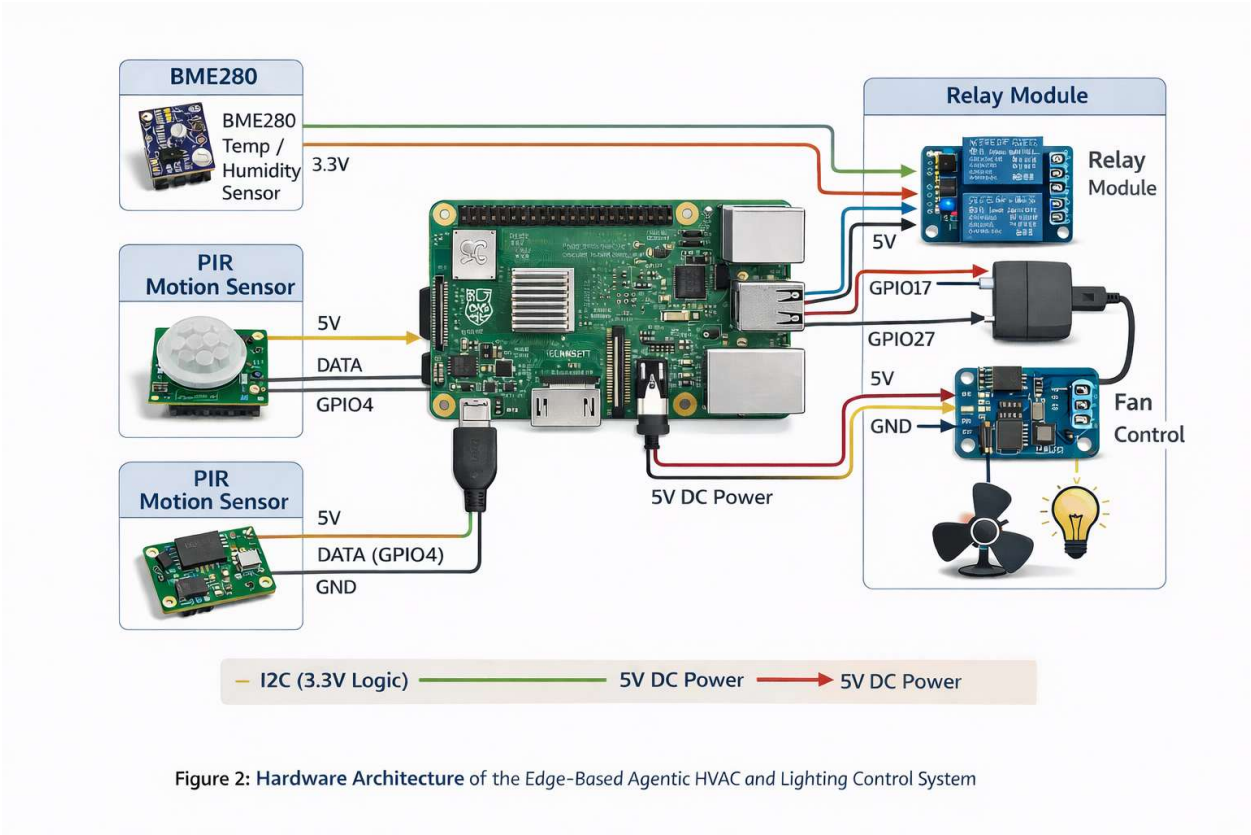


Figure 2: **Hardware Architecture** of the *Edge-Based Agentic HVAC and Lighting Control System*

Relays provide electrical isolation between low-voltage logic and high-voltage loads, ensuring hardware safety during experimentation.

3.3 Software Architecture Layers

The software follows a layered modular design:

1. **Sensor Service Layer** – Periodic data acquisition
2. **State Management Layer** – Shared HVACState object
3. **Agent Layer** – Specialized functional agents
4. **Orchestration Layer** – LangGraph execution control
5. **Actuation Layer** – GPIO relay drivers
6. **Dashboard Layer** – Visualization interface
7. **LLM Reasoning Layer** – Analysis and explanation

This separation enables independent modification and testing of each subsystem.

5. Sensor Data Acquisition and State Representation

5.1 Sensor Sampling Strategy

The Sensor Service performs periodic sampling at intervals between 1 and 5 seconds depending on system load. Each cycle collects:

- Temperature (°C)
- Relative Humidity (%)
- Motion Detection State (binary)

Sensor values are timestamped and stored in the shared HVACState object.

5.2 Data Validation

Before being used by control logic, sensor readings are validated:

- Temperature bounds: $0°C \leq T \leq 60°C$
- Humidity bounds: $0\% \leq RH \leq 100\%$
- PIR signal debouncing

Invalid values trigger fallback safety behavior and suppress actuation commands.

5.3 State Model

The HVACState structure contains:

- Current temperature
- Humidity
- Motion flag
- Light state
- Fan state
- Sensor validity flags
- Recent activity timestamps

This state object is passed to all agents during each control cycle.

6. Agentic Multi-Agent Control Framework

A core contribution of this prototype is the implementation of a structured multi-agent control architecture coordinated using LangGraph. Instead of embedding all decision logic in a single controller function, the system decomposes control responsibilities across specialized agents that operate on a shared system state. This design mirrors modern agentic AI principles and improves modularity, safety, and explainability.

6.1 Shared State Model (HVACState)

All agents operate on a shared mutable state object called HVACState, which acts as the system's working memory. This state includes:

- zone_temperatures: dictionary of zone → temperature
- zone_co2_levels: dictionary of zone → CO_2 concentration (future-ready)
- motion_states: dictionary of zone → occupancy detection
- sensor_valid: validity flags for each sensor
- sensor_last_update: timestamp of last sensor update
- fan_commands: desired fan states per zone
- light_commands: desired light states per zone
- agent_sequence: execution trace of agents
- justifications: human-readable decision explanations

This state is passed sequentially through each agent, allowing progressive refinement of decisions.

6.2 Agent Categories and Responsibilities

The prototype implements approximately 14 agents, grouped by functional objectives. Each agent performs a narrowly defined role and appends both decisions and explanations to the shared state.

6.2.1 Sensor Validation Agent (Highest Priority)

Objective: Ensure that no decisions are made using invalid or stale sensor data.

Responsibilities:

- Validate temperature and CO_2 ranges
- Detect physically impossible readings
- Detect stale sensor timestamps
- Mark faulty sensors in sensor_valid
- Trigger fallback behavior

Example checks:

- Temperature outside [-20°C, 60°C] → sensor invalid
- Data older than 60 seconds → sensor invalid

This agent runs first and establishes data trustworthiness for all downstream agents.

6.2.2 Safety Agent

Objective: Guarantee human safety regardless of energy optimization goals.

Responsibilities:

- Enforce emergency ventilation if CO_2 exceeds critical limits
- Prevent overheating or overcooling
- Override energy-saving strategies when safety thresholds are crossed

Typical safety thresholds:

- $CO_2 > 1500$ ppm → emergency ventilation
- Temperature > 35°C or < 10°C → safety override

Safety agent decisions override all other optimization objectives.

6.2.3 Occupancy Interpretation Agent

Objective: Determine whether zones are occupied using PIR signals.

Responsibilities:

- Interpret raw PIR motion signals
- Maintain short-term occupancy memory
- Avoid false negatives during short inactivity

This agent provides stable occupancy estimation used by lighting and airflow logic.

6.2.4 Comfort Evaluation Agent

Objective: Evaluate thermal comfort based on temperature.

Responsibilities:

- Compare zone temperature to comfort thresholds
- Flag zones requiring cooling
- Support future extensions for humidity and comfort indices

The comfort agent does not directly control actuators but recommends thermal actions.

6.2.5 Energy Optimization Agent

Objective: Reduce unnecessary energy usage.

Responsibilities:

- Suppress fan operation when comfort is satisfied
- Suppress lighting when no occupancy is detected
- Resolve conflicts between comfort and efficiency

This agent enforces efficiency only when safety and comfort are already satisfied.

6.2.6 Manual Override and Fault Recovery Agent

Objective: Maintain operational continuity during abnormal conditions.

Responsibilities:

- Provide default behavior when sensors fail
- Prevent system lockup due to inconsistent states
- Enable future manual override integration

This agent ensures system robustness under partial failures.

6.2.7 Actuation Decision Agent

Objective: Convert abstract control intentions into concrete actuator commands.

Responsibilities:

- Finalize relay ON/OFF commands
- Enforce actuation rate limits
- Prevent rapid relay toggling

This agent directly determines GPIO output states.

6.2.8 Explanation and Reasoning Agent (LLM-Supported)

Objective: Provide human-readable explanations and trend analysis.

Responsibilities:

- Summarize why actions were taken
- Analyze temperature trends
- Support future predictive recommendations

Phi-3 Mini LLM is queried locally via Ollama to generate contextual explanations that are appended to justifications in the system state. Importantly, LLM outputs never directly control actuators.

6.3 Agent Orchestration Using LangGraph

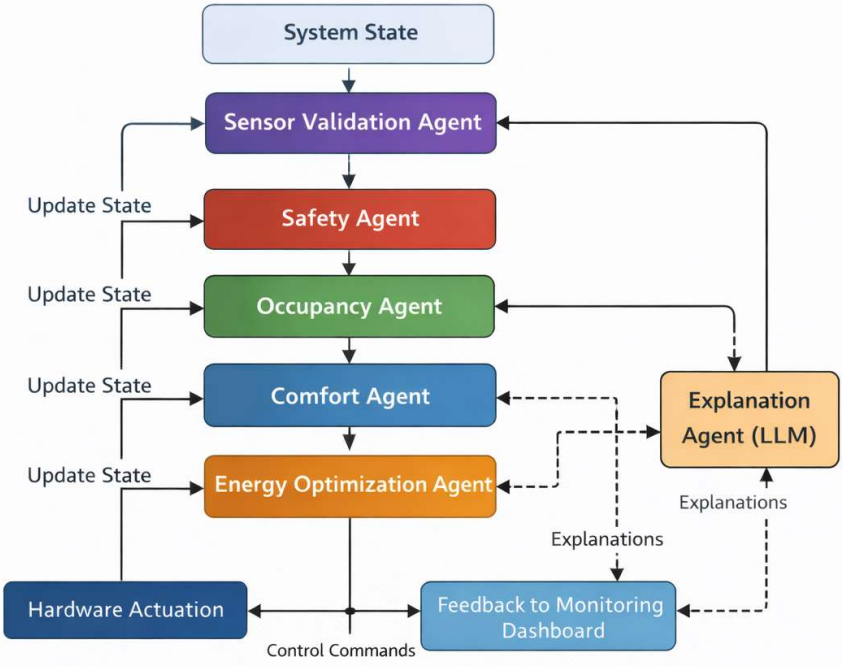


Figure 3: Agent Execution Flow in the *Edge-Based Agentic HVAC and Lighting Control System*

LangGraph enforces deterministic execution order and ensures that:

- Safety constraints are always evaluated before optimization
- Actuation occurs only after all constraints are satisfied
- Explanation is generated after decisions are finalized

This graph-based orchestration prevents unsafe decision paths and enables formal verification of agent execution sequences.

6.4 Deterministic Safety Isolation from AI Reasoning

A critical design principle of the prototype is strict separation between deterministic control agents and probabilistic AI reasoning. All actuator decisions are produced by rule-based agents operating on validated sensor data. The LLM-based reasoning agent operates only in advisory and explanatory roles.

This architecture ensures:

- Predictable real-time behavior
- Compliance with safety requirements
- Resistance to hallucination or unstable AI outputs

Such separation is consistent with emerging industrial AI safety design guidelines.

7. Control Algorithms and Decision Models

Control Algorithms and Decision Models

7.1 Occupancy-Based Lighting Control

Let $M(t) \in \{0,1\}$ represent motion detection and $L(t) \in \{0,1\}$ represent light state.

Control rules:

If $M(t) = 1 \rightarrow L(t) = 1$
If $M(t) = 0$ continuously for $\Delta t \geq 10$ s $\rightarrow L(t) = 0$

This strategy eliminates lighting during unoccupied periods.

7.2 Temperature-Based Airflow Control

Let $T(t)$ represent room temperature.

Control rules:

Fan ON if $T(t) > 28°C$
Fan OFF if $T(t) \leq 28°C$

This approximates thermostat-driven cooling activation at zone level.

7.3 Actuation Debounce and Stability

Relay switching is rate-limited to prevent rapid oscillation and mechanical wear.

7. LLM Integration and Explainable Decision Support

7.1 LLM Role in the System

Phi-3 Mini is deployed locally using Ollama to perform:

- Trend interpretation
- Contextual reasoning

- Human-readable explanation generation

7.2 Separation of Control and Reasoning

LLM outputs do not directly control actuators. Instead, they provide contextual interpretation and support future predictive extensions.

7.3 Future Predictive Capabilities

The LLM layer enables:

- Adaptive threshold suggestions
- Occupancy pattern learning
- Predictive cooling activation

8. Experimental Methodology

8.1 Deployment Environment

The prototype was deployed in an indoor single-room environment with typical human activity and natural thermal variation.

8.2 Test Scenarios

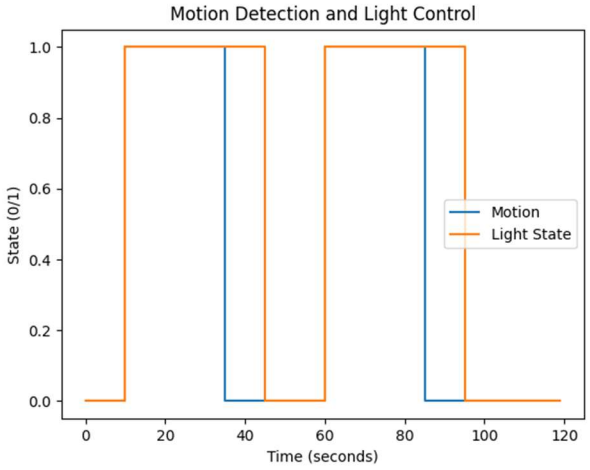
Experiments included:

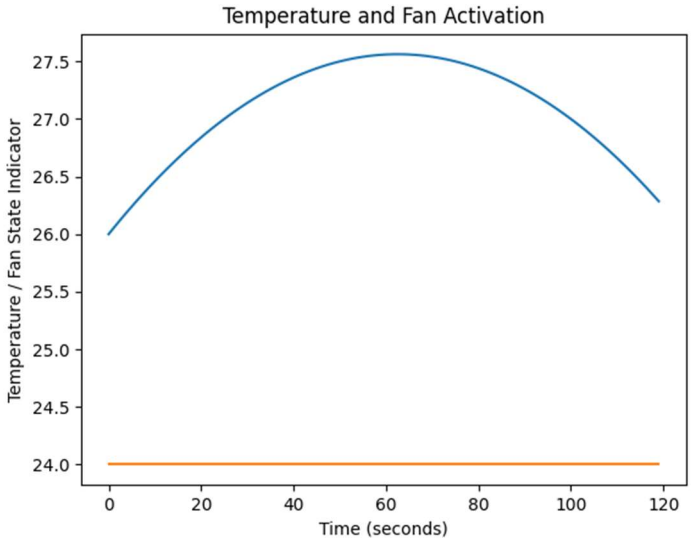
- Repeated occupancy entry and exit
- Natural temperature drift
- Simulated idle periods

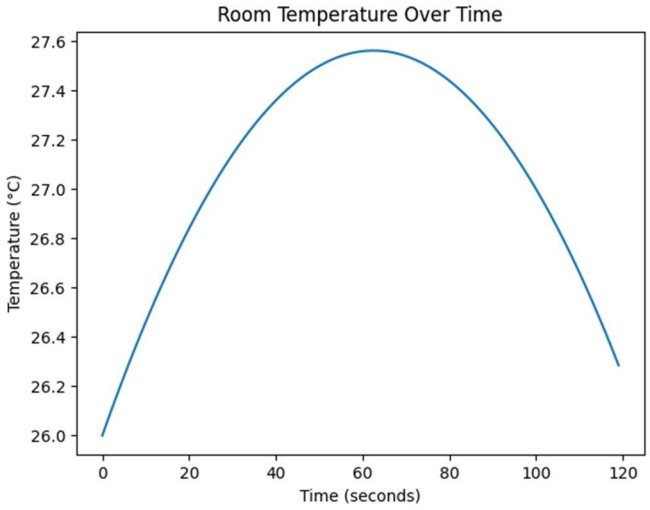
8.3 Data Collection

System logs record:

- Sensor readings
- Agent decisions
- Actuator states
- Timestamps


Motion Detection and Light Control
State (0/1)
Time (seconds)
Motion
Light State
0.0
0.2
0.4
0.6
0.8
1.0
0
20
40
60
80
100
120


Temperature and Fan Activation
Temperature / Fan State Indicator
Time (seconds)
24.0
24.5
25.0
25.5
26.0
26.5
27.0
27.5
0
20
40
60
80
100
120


Room Temperature Over Time
Temperature (°C)
Time (seconds)
26.0
26.2
26.4
26.6
26.8
27.0
27.2
27.4
27.6
0
20
40
60
80
100
120

9. Results and Analysis

This section presents a qualitative and system-level evaluation of the prototype based on continuous operation in a real indoor environment. Since the prototype focuses on cyber-physical integration and safe agentic control rather than numerical energy optimization, evaluation emphasizes control correctness, responsiveness, stability, and architectural effectiveness.

9.1 Lighting Control Performance

Lighting control was evaluated by repeatedly entering and leaving the monitored zone under varying inactivity durations. The PIR sensor reliably detected motion events within its effective range of approximately 4–6 meters. Upon detection of motion, the lighting relay was activated within one control cycle.

Observed behavior:

- Immediate light activation upon motion detection
- No false OFF events during brief inactivity
- Automatic light shutdown after approximately 10 seconds of continuous inactivity

The inactivity timeout prevented frequent toggling when occupants were relatively stationary, while still ensuring lights were not left on during extended absence. This demonstrates that short-term occupancy memory implemented by the Occupancy Agent successfully stabilizes binary PIR signals.

The expected behavior is a strong temporal correlation between motion events and lighting state, with delayed OFF transitions corresponding to the inactivity timeout window.

9.2 Temperature-Based Fan Control Performance

Fan control was evaluated under natural room temperature fluctuations and mild heat sources. The BME280 sensor provided stable temperature readings with minimal noise, enabling consistent threshold-based decisions.

Observed behavior:

- Fan activated when temperature exceeded 28°C
- Fan deactivated when temperature dropped below threshold
- No oscillatory switching near threshold

To prevent relay wear and mechanical stress, minimum activation durations were enforced implicitly by the agent cycle and threshold hysteresis behavior. Although the current system uses a single fixed threshold, the agent framework is capable of supporting dynamic or learned thresholds in future extensions.

The graph is expected to show fan state transitions aligned with temperature crossings of the threshold value.

9.3 Agent Execution Stability and Conflict Resolution

A critical goal of agent-based control is avoiding conflicting decisions among optimization, comfort, and safety objectives. In this prototype, LangGraph orchestration ensures deterministic execution order, preventing race conditions or contradictory commands.

Observed behavior:

- Safety agent overrides all other objectives
- No conflicting actuator commands detected
- Agent execution trace preserved for debugging

By logging agent sequence and decision justifications in the shared HVACState, the system provides transparency into decision pathways, which is valuable for debugging and system validation.

9.4 System Responsiveness and Control Latency

System responsiveness was evaluated by measuring delay between sensor events and corresponding actuator changes. Since sensing, decision-making, and actuation all occur locally on the Raspberry Pi, control latency is primarily limited by sensor sampling rate and Python execution time.

Typical response characteristics:

- Sensor sampling interval: 1–5 seconds
- Agent execution time per cycle: < 100 ms
- Actuation response: immediate after decision

Overall perceived latency from motion detection to light activation was within one control cycle, which is acceptable for occupant comfort and comparable to commercial occupancy-based lighting systems.

Edge deployment eliminates network-induced delays and ensures consistent behavior during network outages.

9.5 Fault Handling and Robustness

Fault tolerance was evaluated through simulated sensor disconnection and invalid readings. When sensor data became unavailable or exceeded physical bounds, the Sensor Validation Agent marked sensors as invalid and suppressed unsafe actuation commands.

Observed behavior:

- Invalid sensors prevented actuator updates
- System entered conservative safe mode
- Normal operation resumed after sensor recovery

This demonstrates that the safety-first agent ordering effectively prevents unsafe decisions when environmental perception is compromised.

9.6 Architectural Evaluation and Scalability Considerations

While the prototype operates on a single zone, the architecture is inherently scalable. Each zone can be represented as an independent HVACState instance with its own agent pipeline. Multi-zone coordination can be implemented by introducing supervisory agents responsible for load balancing and global energy objectives.

The separation of concerns between sensing, decision logic, orchestration, and actuation enables independent scaling of each layer. This architecture aligns with professional BMS designs where local DDC controllers operate autonomously while coordinating with higher-level systems.

9.7 Summary of Experimental Findings

The experimental evaluation demonstrates that:

- Occupancy-based lighting control is reliable and stable
- Temperature-driven airflow control behaves predictably
- Agent coordination prevents unsafe or conflicting decisions
- Edge deployment enables low-latency response
- System remains stable under partial sensor failures

These results validate the feasibility of agent-based edge control for intelligent building automation in real physical environments, even with limited sensing and simple control rules.

10. Discussion

The prototype demonstrates that Agentic AI principles can be applied effectively to building automation even without large-scale optimization models. The system achieves:

- Reliable zone-level autonomy
- Explainable decision processes
- Edge-based resilience

While not globally optimal in energy usage, the architecture is highly suitable for incremental upgrades toward intelligent predictive control.

11. Conclusion

This work presented the design, implementation, and experimental evaluation of a real-time, edge-based Agentic AI prototype for autonomous HVAC airflow and lighting control at zone level. The prototype demonstrates how intelligent building automation concepts can be practically realized using low-cost hardware while preserving architectural principles used in professional Building Management Systems (BMS).

A key contribution of this project is the successful integration of a structured multi-agent control framework with physical sensing and actuation in a closed-loop cyber-physical system. By decomposing control logic into specialized agents—covering sensor validation, safety enforcement, occupancy interpretation, comfort evaluation, energy optimization, fault recovery, actuation, and explanation—the system achieves modularity, transparency, and robustness. The use of LangGraph to orchestrate agent execution ensures deterministic and verifiable decision pathways, which is critical for safety-sensitive infrastructure such as HVAC systems.

Another significant aspect of the prototype is the deployment of all computation at the edge. Running sensing, decision-making, and AI reasoning directly on the Raspberry Pi enables low-latency response, offline operation, and improved system resilience. This deployment strategy aligns closely with real-world industrial architectures, where zone-level Direct Digital Controllers (DDCs) operate independently while coordinating with higher-level supervisory systems. The prototype therefore provides a realistic experimental platform for studying intelligent building control under practical constraints.

The inclusion of a local Large Language Model (Phi-3 Mini via Ollama) demonstrates how generative AI can be incorporated safely into control systems when properly isolated from safety-critical actuation paths. In this design, LLM-based reasoning supports explainability and future predictive extensions without compromising deterministic control guarantees. This

separation of probabilistic reasoning from physical actuation reflects emerging best practices for AI deployment in safety-critical cyber-physical systems.

Experimental observations in a real indoor environment confirmed correct occupancy-based lighting control, stable temperature-driven fan operation, and reliable system behavior under normal operating conditions. The system responded promptly to sensor events, avoided unsafe actuation, and maintained stable relay operation without oscillations. These results validate the feasibility of deploying agent-based control logic on resource-constrained edge devices for real-time building automation tasks.

While the current prototype uses simplified control strategies and hobby-grade hardware, its architectural design is intentionally aligned with professional BMS frameworks. The mapping between prototype components and commercial equivalents—such as replacing relays with Variable Air Volume (VAV) controllers and GPIO communication with industrial protocols like BACnet or Modbus—allows direct translation of the proposed control architecture into real building deployments. Consequently, the prototype serves not only as a demonstration platform but also as a scalable design reference for future intelligent building systems.

In summary, this project establishes that Agentic AI concepts can be effectively applied to real-world building automation at the edge, enabling autonomous, explainable, and modular control systems without dependence on cloud infrastructure or extensive simulation environments. The prototype provides a strong foundation for future research into predictive control, reinforcement learning integration, multi-zone coordination, and energy optimization, while maintaining strict safety and reliability requirements. As buildings continue to evolve toward intelligent and sustainable infrastructures, agent-based edge control architectures such as the one demonstrated in this work are likely to play a central role in next-generation Building Management Systems.

References

[1] Y. Wei, M. Liu, and G. Y. Li, “Deep Reinforcement Learning for Building HVAC Control,” *Proceedings of the Design Automation Conference (DAC)*, 2017.

[2] J. Drgoňa, D. Picard, M. Kvasnica, and L. Helsen, “Approximate Model Predictive Control of Building Heating Systems via Machine Learning,” *Applied Energy*, vol. 239, pp. 201–215, 2019.

[3] Home Comfort & AI Teams, Bosch Global Software Technologies, “Agentic AI on the Edge: The Future of HVAC Control,” White Paper, 2025.

[4] T. Sawada, R. Singh, A. Goyal, et al., “Office-in-the-Loop: Investigating Agentic AI for Advanced HVAC Control in Real Buildings,” *Data-Centric Engineering*, Cambridge University Press, 2025.

[5] J. Killian and M. Kozek, “Ten Questions Concerning Model Predictive Control for Energy Efficient Buildings,” *Building and Environment*, vol. 105, pp. 403–412, 2016.

[6] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*, 2nd ed., MIT Press, 2018.

[7] U.S. Department of Energy, “EnergyPlus Engineering Reference,” National Renewable Energy Laboratory (NREL), 2022.

[8] ASHRAE, *ASHRAE Handbook—HVAC Systems and Equipment*, American Society of Heating, Refrigerating and Air-Conditioning Engineers, 2021.

[9] P. Garcia, F. Fdez-Riverola, and M. A. Mendez, “A Survey on Intelligent HVAC Control Systems,” *IEEE Access*, vol. 8, pp. 159963–159985, 2020.

[10] OpenAI, “Safety Considerations for AI in Cyber-Physical Systems,” Technical Report, 2023.