The 39th Conference of the Mechanical Engineering Network of Thailand
15 - 18 July 2025, Khon Kaen

SBT-0002

A Cost-Effective Building Automation System: Enhancing Open-Source Microcontrollers with Industrial-Grade Capabilities

Siwat Sirichai[1], and Tawan Paphapote[2]

[1] Faculty of Engineering, Chulalongkorn University, Phathum Wan, Bangkok, 10330, Thailand
*Corresponding Author: siwat.s@satitm.chula.ac.th

Abstract

With increasing urbanization, new buildings often require more efficient energy consumption tracking and remote control of electrical appliances and lighting. Two main approaches have been used to address these needs: large-scale building automation systems utilizing industrial-grade logic controllers [1] and proprietary network protocols (such as Clipsal's C-BUS and Siemens's ProfiNET and ProfiBUS), and consumer-grade smart home equipment that leverages wireless technologies to simplify installation. Protocols commonly used in consumer IoT include, but are not limited to: ZigBee, Z-Wave, RF433, and WiFi [2]. While consumer-grade devices tend to be less expensive, they may present challenges for large-scale deployments—for example, RF433 often requires repeaters with complex logic to achieve wide coverage, and wireless systems can be vulnerable to jamming and interference, potentially increasing susceptibility to vandalism.

This research proposes a building automation system designed to balance the robustness of industrial-grade controllers with the cost efficiency of consumer solutions. Our approach combines the advantages of open-source microcontroller platforms, specifically the ESP32, renowned for its large open-source community and extensive peripherals, with industrial features such as wired Ethernet connectivity and modularity through differential I²C. We also explore the use of MQTT for automation signaling. Although preliminary evaluations and test runs indicate that the stability of our solution may not yet fully match that of established industrial systems, the approach shows promise as a cost-effective option for installations in industrial-grade environments.

Keywords*:* Internet of Things, Programmable Logic Controller, Building Automation System

1 Background

Building Automation Systems have become increasingly critical in the modern urban environment due to the demands for increased energy efficiency, and higher granularity of controls. Historically, this domain has been dominated by two primary approaches: large-scale building automation systems utilizing industrial-grade Programmable Logic Controllers (PLCs) and consumer-grade smart home solutions, many of them leveraging wireless technologies. [2]

Industrial-grade PLCs have long been the de facto standard for robust and reliable automation, especially in demanding industrial environments. [1] These systems, however, often come with significant costs and limited flexibility in integrating with contemporary Internet of Things (IoT) protocols. Conversely, the proliferation of consumer-grade smart home devices focuses on affordability and ease of installation but often neglects reliability and security. This is a concern, especially when used in larger building deployments. These wireless solutions, relying on protocols like Zigbee, Z-Wave, and Wi-Fi, often struggle with range limitations, interference, potential vulnerabilities, and scalability.

Recognizing the limitations of both extremes, this paper proposes a building automation system that strikes a balance between the robustness of industrial PLCs and the cost-effectiveness of consumer IoT devices. Our approach centers on the development of a custom-designed proof-of-concept hardware mainboard utilizing distributed microcontrollers to emulate the modularity and reliability of traditional PLCs. This hardware-centric approach allows for a system that leverages the flexibility and programmability of open-source platforms while addressing the critical reliability and security concerns associated with wireless solutions. By developing a custom hardware solution, we gain precise control over the system's architecture, enabling us to optimize it for the specific requirements of building automation. This system is designed to provide a cost-effective and scalable alternative, focusing on wired communication and modular design to ensure reliability and expandability. The following sections will delve into the design and implementation of this hardware, detailing our choices and the rationale behind them.

2 Methodology

This research aims to evaluate the viability of a distributed microcontroller-based building automation system as a replacement for existing, aging systems. Specifically, we will conduct a real-world test deployment at Chulalongkorn University Demonstration Secondary School, where an existing Clipsal C-BUS-based automation system is slated for replacement. Our project begins with a thorough survey of the existing C-BUS installation to document its functionality, I/O requirements, operational characteristics, and automation scripts. This survey will establish the baseline requirements for our replacement system. Based on these findings, we proceed with the design and development of our proposed system, detailed in the following section.

In parallel, a comparative design will be created based on Schneider Electric's current building automation offerings to serve as a direct comparison point. The custom-designed system will be implemented and deployed at the Chulalongkorn University Demonstration Secondary School. A comprehensive comparative analysis will be conducted, evaluating both systems in terms of price and cost-effectiveness, functionality and feature set, and benefits and drawbacks. Note that Schneider Electric’s system already has a proven technical specification and datasheets. Furthermore, a long-term performance evaluation will be carried out over a period of one year. During this time, the following metrics will be collected and analyzed for our custom solution: Mean Time Between Failures (MTBF), breakdown frequency, maintenance requirements, and operational stability. This quantitative data will be compared with industry standards and data from other building automation systems. This methodology will provide a general assessment of the proposed microcontroller-based building automation system based on open-sourced technologies, establishing its feasibility as a sufficiently reliable and cost-effective alternative to existing solutions.

3 System Design and Implementation

Following the methodology outlined above, the design and implementation of our distributed microcontroller-based building automation system began with a detailed analysis of the requirements gathered from the C-BUS system survey. This analysis informed the architecture of our custom-designed mainboard, which is intended to emulate the modularity and robustness of industrial PLCs while leveraging the cost-effectiveness and flexibility of open-source microcontrollers.

Components were selected based on general availability within a reasonable price range. Most core components were specifically chosen to support I²C, a bus protocol that allows us to use only 2 I/O pins to support multiple bus devices, maximizing efficiency while minimizing cost.

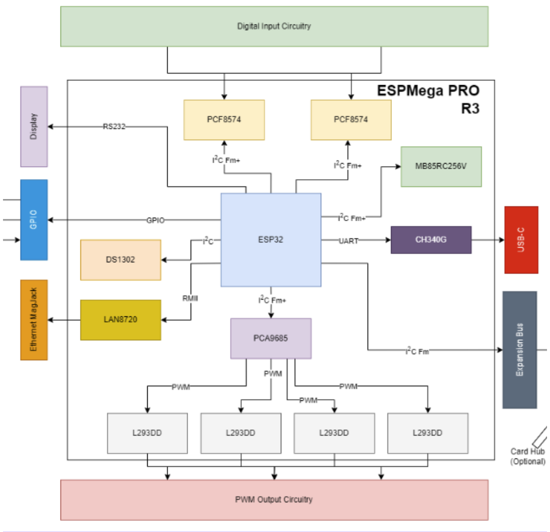

Fig 1: System Communication Architecture

Our design utilizes a distributed microcontroller architecture, employing a multi-core processor as the central processing unit, such as the ESP32 with its integrated Wi-Fi and Ethernet capabilities and active open-source community support as shown in *Figure 1*. [3] This approach enables a modular system, like PLC expansion cards, with communication between controllers facilitated by a robust communication bus like RS-485. This bus allows for multiple controllers to communicate with a master, facilitating scalable and centralized control.

For inter-device communication on the mainboard itself, I²C is the primary protocol, chosen for its efficiency in connecting numerous I/O expanders and peripheral ICs while minimizing pin usage. [4] A range of connectivity options are considered, including Ethernet for reliable network connection, essential for critical building management functions, RS-232 for devices like touch screen displays and generic serial devices, and infrared for controlling legacy equipment. For input modularity, ICs like the PCF8574 can be utilized, with appropriate overvoltage protection implemented using voltage dividers and Zener diodes. [5] Output modularity can be achieved using ICs with PWM capabilities, such as the PCA9685 [6], coupled with signal amplification using drivers like the L293DD [7] for smaller low current applications or MOSFET-based circuit for higher current rating.

Persistent storage for configuration and state data is crucial, and FRAM, like the Fujitsu MB85RC256V, offers significant advantages over EEPROM in terms of write endurance and speed, which is crucial for real-time state storage. [8] A real-time clock IC, such as the DS1307, backed by a coin cell battery, ensures accurate timekeeping, especially during power outages. [9] For PC communication, a USB connection can be established using a converter IC like the CH340G. Finally, a touchscreen display, potentially with an integrated processor to offload display processing, can enhance system responsiveness.

The wired connection architecture inherently rules out most accessible wireless attacks such as Radio Frequency (RF) jamming and signal interception, as physical intrusion is required to interact with or interfere with the signal. Additionally, during our testing phase, we implemented VLAN segregation for all PLC device traffic, managed by a FortiGate device to prevent unauthorized access from the local network. The system primarily uses MQTT over Ethernet for inter-device communication. At the MQTT layer, packets are published with QoS level 0. However, for coil operations (relay and digital output controls), when a set command is sent, a confirmation of modification payload is returned to ensure reliable state changes. This system embraces the concept of edge computing and edge independency. The PLC units can connect to each other and function as Remote Terminal Units (RTUs). This architecture ensures that latency-sensitive calculations are performed at the edge, with results reported back to the central controller. Since the central controller is implemented in JavaScript and structured as a microservice, it can scale horizontally to support hundreds of devices per network segment. During testing, devices were connected to a programmable DC power supply, showing current consumption of approximately 5W (excluding relays and contactors). This compares favorably to industrial solutions such as the Siemens S7-1200 PLC, which has a power rating of 9.6W (24V × 400mA) [10].

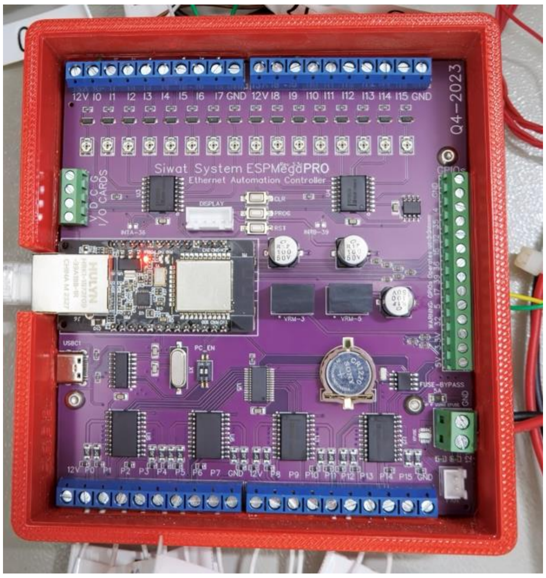

Fig 2: Assembled Controller PCB

The mainboard shown in *Figure 2* is designed using a two-layer FR4 PCB to minimize cost, incorporating dedicated input, output, and power stages with protection circuitry. The overall layout is optimized for compactness and efficiency. The prototype undergoes testing to validate its functionality and performance. Specifically, the test run at Chulalongkorn University Demonstration Secondary School focused on controlling the walkway lighting, classroom lighting fixtures, classroom fans, air purification system, air conditioning system, and UV pest control system. This comprehensive control of diverse building systems allowed for a thorough evaluation of the system's capabilities and reliability in a real-world setting. The software are available as open source and available on GitHub at https://github.com/SiwatINC/ESPMegaPRO3-library.

Hardware wise, as it follows the same conceptual design with I/O cards and modular components, the system can be serviced like any other PLC. C++ programming skills are beneficial when customizing the system.

4 Comparison to Industrial Systems

The system enclosures are 3D printed with polycarbonate (PC) plastic. In contrast to industrial PLC systems, which are rigorously designed for the harshest factory floor conditions, this system trades environmental resiliency for affordability to better align with typical building automation requirements. Our microcontroller-based building automation system offers a compelling cost advantage when compared with industrial-grade solutions like Schneider Electric's SpaceLogic Controller (which is Schneider's replacement for the outdated Clipsal solutions after they acquired Clipsal). The price differential is substantial: a single dimming controller from Schneider costs upward of 7,600 THB for a 2-channel dimmer, while our solution costs only 1,500 THB for the base controller and approximately 500 THB per channel when paired with industrial-grade 12V dimmers from local companies. Similarly, the display component for our system costs 3,500 THB compared to the SpaceLogic counterpart at 14,000 THB. The cost-benefit analysis suggests that our solution represents significant value in situations where absolute reliability isn't critical and occasional failures can be tolerated. Even accounting for potential failures, the money saved could be used to purchase multiple backup controllers for the price of a single SpaceLogic controller, providing redundancy while remaining cost-effective.

This comparison illustrates the practical economic advantage of our approach for budget-conscious implementations, particularly in settings where the marginal reliability benefits of industrial-grade systems do not justify their substantially higher cost in some situations.

5 Results and Conclusion

Implementing our distributed microcontroller-based building automation system at Chulalongkorn University Demonstration Secondary School yielded valuable insights into its performance and viability. As seen in *Figure 3* and *Figure 4*; while the system experienced some failures during the initial deployment phase, these were primarily attributed to installation faults, such as loose screws, sensor misalignment, and some initial configuration errors related to the complex scheduling automation systems. The system incorporates built-in fault detection capabilities. Since ICs and expansion cards operate on I^2C and differential I^2C protocols, I^2C acknowledgment (ACK) signals are monitored. When devices miss multiple ACK responses from known addresses, system alarms are triggered. Additionally, the ESP32 includes an integrated temperature sensor for thermal monitoring. Detailed analysis of the 9 hardware faults observed during evaluation revealed the following breakdown: stuck relay mechanisms, contact chattering due to loose installation screws, and ESP32 modules randomly disconnecting from the network. Most issues stemmed from installation errors, though some failures, such as ESP32 disconnections, were attributed to quality control variations in the ESP32 modules used. We are currently exploring alternative modules to address these reliability concerns. After addressing these initial issues, the system's operational stability was deemed acceptable by the building maintenance personnel, indicating that the system met the practical requirements of the site. The system successfully managed the diverse range of building systems outlined above, demonstrating its ability to handle complex control logic and scheduling. The automated walkway lighting, classroom lighting controls, and integrated management of air purification and air conditioning systems all performed according to specifications, providing the controllability necessary for future integration with larger automation systems.

A reliability analysis of the system showed promising results. Over the evaluation period from full-scale deployment on April 17, 2024, to February 28, 2025 outlined in *Figure 3*, we observed a total of 9 hardware faults across the 47 deployed devices. This translates to a total operational time of 357,576 device-hours and yields a Mean Time Between Failures (MTBF) of 39,730.7 hours, or approximately 1,655.4 days (4.53 years) per failure. While these results are encouraging, it's important to acknowledge that this represents an early evaluation period of approximately 10 months. The MTBF may decrease as the devices age and components begin to experience wear over their full life cycles. Nevertheless, these preliminary reliability metrics suggest that the system can provide a dependable solution for building automation applications within the intended operational context.

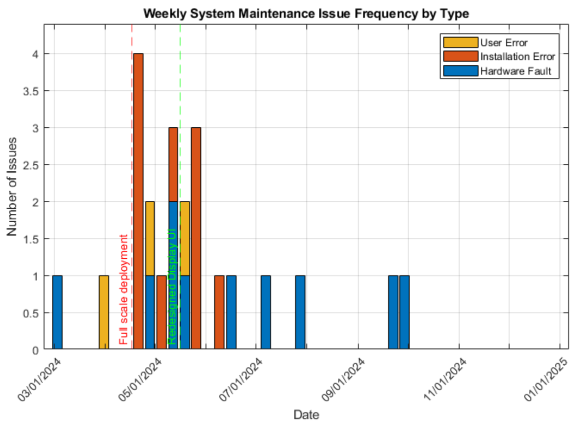

Fig 3: Maintenance Issues Over Time

In comparison to the industrial-grade solution offered by Schneider Electric, our system exhibited a lower level of initial reliability. However, this difference was largely due to the meticulous engineering and rigorous testing inherent in industrial-grade products, which come at a significantly higher cost.

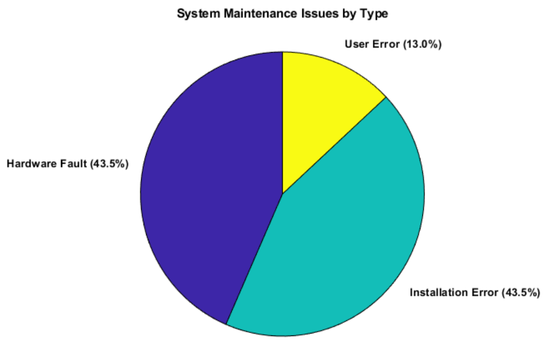

Fig 4: Defect Type Distribution

Our research demonstrates that utilizing affordable microcontrollers and integrated circuits, coupled with the support of a large community-driven development ecosystem, can effectively bridge the gap for cost-optimized building management solutions. The comprehensive evaluation, including security implementation, scalability analysis, and detailed fault characterization, proves the viability and cost-effectiveness of our proposed approach. The results show that a well-designed system using readily available components can provide a functional and acceptable alternative to expensive, proprietary solutions, while maintaining essential building automation functionalities and offering significant cost savings. This study highlights the potential for wider adoption of such systems in environments requiring controllable infrastructure for future integration with larger automation systems.

6 References

[1] G. Dunning, Introduction to Programable Logic Controllers (2nd Edition), Delmar Thomson Learning, 2002.

[2] Internet Society, "The Internet of Things: An Overview," October 2025. [Online]. Available: https://www.academia.edu/download/48790442/ISOC-IoT-Overview-20151014_0.pdf. [Accessed 20 Feburary 2025].

[3] ESPRESSIF, "ESP32 Series Datasheet (Ver. 4.8)," Janurary 2025. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf. [Accessed 27 Feburary 2025].

[4] NXP Semiconductors, "UM10204: I2C-bus specification and user manual Rev. 7.0," 1 October 2021. [Online]. Available: https://www.nxp.com/docs/en/user-guide/UM10204.pdf. [Accessed 24 Feburary 2025].

[5] Texas Instrument, "PCF8674 Remote 8-Bit I/O Expander for I2C Bus," September 2024. [Online]. Available: https://www.ti.com/lit/ds/symlink/pcf8574.pdf. [Accessed 20 Feburary 2025].

[6] NXP Semiconductors, "PCA9685 Product Datasheet (Rev.4)," 16 April 2015. [Online]. Available: https://www.nxp.com/docs/en/data-sheet/PCA9685.pdf. [Accessed 21 Feburary 2025].

[7] Texas Instrument, "L293x Quadruple Half-H Drivers," Janurary 2016. [Online]. Available: https://www.ti.com/lit/ds/symlink/l293.pdf. [Accessed 20 Feburary 2025].

[8] Fujitsu Semiconductor, "MB85RC256V Data Sheet," October 2015. [Online]. Available: https://www.fujitsu.com/uk/Images/MB85RC256V-20171207.pdf. [Accessed 21 Feburary 2025].

[9] Maxim Integrated, "DS1307 64 x 8, Serial, I2C Real-Time Clock," 10 Feburary 2008. [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/ds1307.pdf. [Accessed 21 Feburary 2025].

[10] SIEMENS, "SIMATIC S7-1200 Data Sheet (6ES7212-1AE40-0XB0)," 28 May 2025. [Online]. Available: https://support.industry.siemens.com/teddatasheet/?caller=SIOS&format=pdf&language=en&mlfbs=6ES7212-1AE40-0XB0.