Building a Smart Home System using Embedded Systems and IoT

Supervisor: [Insert your supervisor's full name]
Students: Nguyễn Quốc Đại, Đặng Công Minh, Lê Quốc Đạt
Class: [Insert class]
Major: Applied Information Technology
Date: [Insert submission date]

Acknowledgements

We would like to express our sincere gratitude to our supervisor [Supervisor’s Name] for their invaluable support and guidance throughout this project. Our thanks also go to the faculty members of the International School, Vietnam National University, for equipping us with foundational knowledge. Lastly, we thank our families and friends for their encouragement.

Team Members and Contributions

No	Name	Task	Contribution
1	Nguyễn Quốc Đại	Hardware design, circuit assembly based on verified schematics	33.3%
2	Đặng Công Minh	Core implementation, ESP32 programming, simulation, hardware testing	33.3%
3	Lê Quốc Đạt	UI/web system implementation, backend communication (MQTT + NodeJS)	33.3%

Abstract

This project presents a smart home prototype using embedded systems and IoT technologies. The system enables real-time monitoring and remote control of home appliances such as lights, fans, AC, and curtains. Using ESP32 microcontrollers, sensors (DHT11 and LDR), and relays, the system gathers and transmits sensor data to a NodeJS backend via MQTT protocol. A responsive web interface allows users to visualize and control connected devices. The system was tested over seven days, showing reliable performance with an average latency of 1.2 seconds and uptime of 98.5%. Although some AI features are under development, the system demonstrates significant potential for practical smart home deployment.

1. Introduction

1.1 Problem Statement / Motivation

Modern households require smarter automation solutions to enhance energy efficiency, convenience, and security. Our project addresses this by proposing an affordable, modular, and scalable smart home system using embedded devices and IoT technologies.

1.2 Project Objectives

- Design an IoT-enabled smart home system.
- Implement remote control for home appliances.
- Develop a web interface for monitoring and control.
- Ensure data security and optimize system performance.

1.3 Scope and Limitations

Scope:

- Control of devices via dashboard: lights, fans, AC, curtains.
- Real-time sensor data visualization (temperature, humidity, light).
- Remote access through MQTT.

Limitations:

- Prototype is limited in scale.
- AI-based optimization not yet integrated.
- Requires stable internet connectivity.

1.4 Methodology Overview

We adopted an iterative development approach combining hardware assembly and embedded programming using Arduino IDE. The web dashboard was built with NodeJS, while MQTT handled device-server communication.

1.5 Significance of the Project

Scientific Significance:
The project contributes to embedded systems research by integrating real-time IoT communication and control using ESP32 and MQTT.

Practical Significance:
The prototype has real-world applications in residential automation, especially in low-cost, easily extensible systems.

2. Literature Review

2.1 Theoretical Background

Key concepts explored include MQTT protocol, sensor integration with microcontrollers, real-time communication models, and UI/UX principles in dashboard design.

2.2 Related Works

Past works have proposed similar smart control systems, but many lack integration of sensor feedback with modular dashboards or secure communication layers. Our system enhances these aspects.

2.3 Analysis and Choice of Technology

We chose:

- **ESP32** for its built-in Wi-Fi and GPIO capabilities.
- **MQTT (Mosquitto)** for lightweight, real-time messaging.
- **NodeJS** for efficient backend handling.
- **Web dashboard** for intuitive control.

3. Methodology and System Design

3.1 Development Process

Hardware:

- ESP32 microcontroller, OLED 0.96", DHT11 (temperature/humidity), LDR, relays.

Software:

- Arduino IDE (sensor logic, relay control, display output).

- NodeJS + MQTT broker (backend).
- Frontend web dashboard (real-time UI using MQTT).

3.2 Requirements

Functional:

- FR-001: Remotely control devices.
- FR-002: Display real-time sensor data.
- FR-003: Maintain bidirectional data communication.

Non-Functional:

- NFR-001: Update rate: every 5 seconds.
- NFR-002: Secure communication via TLS.
- NFR-003: Responsive dashboard UI.

3.3 System Architecture

As illustrated in *slide 11*, the architecture includes:

1. Sensors →
2. ESP32 →
3. MQTT Broker →
4. NodeJS server →
5. Web dashboard.

Modules:

- **Sensor Module**: Collects DHT11 and LDR data.
- **Controller Module**: Issues relay commands.
- **Display Module**: OLED shows local device status.

- **Web Module**: User interaction interface.

4. Results and / Discussion

4.1 Results

- Controlled 4 devices remotely.
- Latency: ~1.2 seconds.
- 98.5% uptime over 7 days.

4.2 Analysis

The system showed consistent performance, effectively demonstrating the integration of software-hardware layers in IoT automation. Data visualizations were clear, and control signals were processed with minimal delay.

4.3 Evaluation

Strengths:

- Modular design.
- Real-time response.
- Easy scalability.

Weaknesses:

- AI module not yet implemented.
- Requires internet for MQTT operation.

Objective Achievement:
3 out of 4 objectives fully achieved. AI remains for future development.

5. Conclusion and / Future Work

5.1 Conclusion

This project successfully developed and validated a working smart home prototype with ESP32, MQTT, and web interface. The system demonstrates reliable automation and monitoring capabilities.

5.2 Recommendations

- Enhance system reliability with offline fallback mode.
- Improve mobile UI responsiveness.

5.3 Future Work

- Integrate AI for predictive automation.
- Support camera and smoke sensor inputs.
- Improve UX and mobile control latency.

6. Bibliography

7. Appendix (optional)