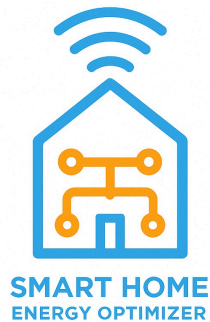


Smart Home Control Unit for Energy Optimization

SDDEC25-08



Our Team

Arie Kraayenbrink - Hardware/Software Developer

Alexander Ryan - UX/Software Developer

Erik Schmidt - Embedded Systems Developer

Gamage Binula Karunaratne - Hardware Developer

Mitchell Jensen - Hardware Developer

Ahamed, Md Maruf - Advisor + Client

Website:

<https://sddec25-08.sd.ece.iastate.edu/>

Executive Summary

The project develops a low-cost smart home control unit that optimizes residential energy use for heating, cooling, and lighting; there is particular emphasis on affordability for households that cannot justify expensive commercial systems such as Nest or Ecobee. This problem is important because heating, cooling, and lighting account for roughly half of residential energy consumption, with disproportionately high utility costs to low-income and energy-burdened households.

Key design requirements include rapid DIY-friendly installation, wireless communication via standards like Zigbee and Wi-Fi, accurate temperature and light sensing, secure data handling, multi-zone HVAC control for up to eight zones, and reliable offline operation through a local hub. Other requirements include safety (HVAC relay isolation and fail-safe behavior), compact and aesthetically unobtrusive hardware, affordable pricing, and usability for non-technical users through simple web and mobile interfaces.

The design centers around a Home Assistant hub running on an Intel NUC-with the future hope of downsizing to a Raspberry Pi-communicating via Zigbee with ESP32-based multi-sensor nodes and relay-based controllers. Each multisensor module combines temperature, ambient light, and presence sensing, feeding into the hub for adjusting HVAC setpoints, driving a custom light dimmer, and scheduling loads via rule-based automations in YAML/Node-RED. A React-based web and mobile UI allows users to override automations and review energy insights. An extended HVAC subsystem adds an ESP32-C6-based zone controller and 24 VAC relays, operating motorized dampers or hydronic valves to enable as many as eight independently controlled zones within the home.

The prototype implemented herein performs end-to-end functionality: the multi-sensors report real data over Zigbee; furnace and zone controllers actuate real HVAC equipment; Home Assistant rules respond within target latency. The testing of this system validates stable operations with quick recovery from network interruptions. User needs for cost, control, and simplicity have been addressed via open-source software, commodity hardware, and interfaces validated through usability and acceptance testing for criteria such as override behavior, ease of configuration, and offline capability. Remaining gaps include finalizing the custom light dimmer for flicker-free operation, deploying production PCBs, and performing long-term quantitative energy savings studies. Future steps include integrating more advanced energy-optimization algorithms, including machine learning, expanding energy monitoring and reporting, adding voice assistants and battery/solar backup, and deepening sustainability analysis around device lifecycle and environmental impact.

Learning Summary

Development Standards & Practices Used

Standard practices used in this project include modular hardware design, PCB design with isolation between low-voltage control and higher-voltage HVAC circuits, and relay-based actuation designed to fail safe for critical HVAC paths. Software practices include using an agile process with weekly sprints, Jira for task tracking, Git for version control, and layered testing - unit, interface, integration, system, regression, acceptance, security, and user testing - driven by explicit requirements. Interface and network practices follow standard IoT patterns: MQTT-style messaging, Zigbee mesh networking, latency budgets for control paths, and defensive handling of packet loss and disconnections.

Some of the considered and applied engineering standards are IEEE 802.15.4 (Zigbee) for low-power mesh networking, IEEE 802.11 (Wi-Fi) for higher-bandwidth communication, ISO/IEC 30162 for the testing of IoT devices, and ISO/IEC 27001 principles for information security management. Other domain-specific standards include things like UL 873 for temperature-indicating and regulating equipment and IEEE C37.90 for safe relay switching in HVAC control circuits.

Summary of Requirements

- Fast, DIY-friendly installation with clear documentation.
- Wireless communication support: Wi-Fi, Zigbee, or Z-Wave using standard IoT protocols.
- It serves to feed with ease low-consumption devices, battery or simple outlet.
- Reliable communication via good signal integrity, error detection, and consistent performance.
- Ability to control line-voltage loads with appropriate surge protection and automatic shutdown safety.
- Temperature control in up to eight multi-zone HVAC zones.
- Support for both motorized duct dampers and hydronic zone valves.
- Normally-open, relay-based actuation. Fail-safe on loss of power or control.
- Coordinating logic to prevent zone conflicts, equipment over-stress, and pressure-related issues.
- Compatibility with old, new, large, and small buildings.
- Security against hackers, including encryption, authentication, and protected local data.
- Correct energy monitoring with a graphical visualization and calibration mechanisms.
- Hub software that runs efficiently on a local server while synchronizing over cloud services.
- Low power sensor design for facilitating long runtimes and possible battery operation.
- Compact, unobtrusive, and visually appealing form factors for devices.

- Aesthetic design that fits varied interior styles.
- Economical pricing with dependable and quality service.
- Operate under project constraints: a limited two-semester timeline, budget limits, and reliance on ETG equipment.
- HVAC compatibility limited to standard 24 VAC systems, no proprietary buses in this prototype.
- Safety through low-voltage wiring practices and isolation for HVAC equipment.

Applicable Courses

In addition to the senior design sequence (CprE/EE 4910–4920), a number of other courses contains directly relevant material for this project.

CPR E 2810: Digital Logic - basic design of digital control logic and state machines; relevant to relay and controller behavior.

CPR E 2880: Embedded Systems I - microcontroller programming and interfacing concepts used for ESP32-based sensor and zone controller firmware.

CprE 3080: Operating Systems: Principles and Practice – useful for understanding Home Assistant’s host environment, resource management, and reliability.

CPR-E 3100: Theoretical Foundations of Computer Engineering – formal reasoning about system behavior, state and correctness for distributed control.

CPR E 3810: Computer Organization and Assembly Level Programming - low-level perspective supporting performance-aware firmware and I/O handling.

ComS 3090: Software Development Practices - structured processes, version control, code reviews, and testing methods reflected in this project's workflow.

ComS 3110: Algorithms – background for future optimization of scheduling, zone coordination, and energy-saving strategies.

EE 2010 / EE 2300: Electric and Electronic Circuits - circuit analysis and device behavior directly applicable to sensor front-ends, relay drivers, and power stages.

New Skills / Knowledge Acquired beyond ISU Curriculum

Practical deployment of Home Assistant, including add-ons, backups, and integration with custom Zigbee devices and HVAC equipment in a real residential environment.

Performs end-to-end Zigbee stack work: flashing coordinators, binding custom ESP32-based end devices, tuning mesh reliability, debugging RF issues with nodes of mixed distances.

Hands-on design of HVAC controls: interfacing to 24 VAC thermostats, furnace call signaling, zoning using motorized dampers/hydrionic valves, normally-open, fail-safe relay topologies.

Safety-oriented low voltage wiring for residential HVAC and dampers; topics include transformer sizing, fusing and isolation practices not typically covered in general laboratory courses.

PCB and enclosure design focused on installability and aesthetics in real homes: clearances in ducts, space in mechanical rooms, user-visible sensor housings. Usability for non-technical homeowners through UI/UX, iterative user testing of overrides, presets, and energy visualization within a smart home context. Practical security hardening of a self-hosted smart home: network segmentation, access control, and encryption aligned to ISO/IEC 27001-style thinking, rather than classroom-only examples. Agile project management at multi-subsystem scale: integrating hardware, firmware, backend, and UI sprints while managing procurement delays and shifting scope-including the addition of 8-zone HVAC.

Table of Contents

Smart Home Control Unit for Energy Optimization	0
1. Introduction	7
1.1. PROBLEM STATEMENT	7
1.2. INTENDED USERS	9
2. Requirements, Constraints, And Standards	11
2.1. REQUIREMENTS & CONSTRAINTS	11
2.2 ENGINEERING STANDARDS	15
3 Project Plan	18
3.1 PROJECT MANAGEMENT/TRACKING PROCEDURES	18
3.2 TASK DECOMPOSITION	19
3.3 PROPOSED PROJECT MILESTONES, METRICS, AND EVALUATION CRITERIA	20
3.4 PROJECT TIMELINE/SCHEDULE	21
3.5 RISKS AND RISK MANAGEMENT/MITIGATION	23
3.6 PERSONNEL EFFORT REQUIREMENTS	25
3.7 OTHER RESOURCE REQUIREMENTS	26
4 Design	27
4.1 DESIGN CONTEXT	27
4.1.1 Broader Context	27
4.1.2 Prior Work/Solutions	28
4.1.3 Technical Complexity	28
4.2 DESIGN EXPLORATION	29
4.2.1 Design Decisions	29
4.2.2 Ideation	30
4.2.3 Decision-Making and Trade-Off	30
4.3 FINAL DESIGN	31
4.3.1 Overview	31
4.3.2 Detailed Design and Visuals	31
4.3.3 Functionality	35
4.3.4 Areas of Challenge	35
4.4 TECHNOLOGY CONSIDERATIONS	36
5 Testing	38
5.1 UNIT TESTING	38
5.2 INTERFACE TESTING	39
5.3 INTEGRATION TESTING	40
5.4 SYSTEM TESTING	41
5.5 REGRESSION TESTING	43
5.6 ACCEPTANCE TESTING	43
5.7 SECURITY TESTING	44
5.8 USER TESTING	45

5.9 RESULTS	45
6 Implementation	46
6.1 Design Analysis	50
7 Ethics and Professional Responsibility	53
7.1 AREAS OF PROFESSIONAL RESPONSIBILITY/CODES OF ETHICS	53
7.2 FOUR PRINCIPLES	55
7.3 VIRTUES	56
8 Conclusions	60
8.1 CONCLUSION	60
8.2 VALUE PROVIDED	60
8.3 WHAT'S NEXT	61
9 References	62
10 Appendices	63
10.1 OPERATION MANUAL	63
10.2 ALTERNATIVE/ INITIAL VERSION OF DESIGN	67
10.3 CIRCUIT DIAGRAM	69
10.4 CODE	71
10.5 TEAM CONTRACT	72

1. Introduction

1.1. PROBLEM STATEMENT

Today, many home and apartment occupants face significant challenges in controlling energy use efficiently. While commercial home automation options exist, their prohibitive cost excludes a large segment of the population. As a result, many residents struggle with high energy bills and rely on inefficient manual control of temperature and lighting. Traditional heating, cooling, and lighting systems often lack intelligent automation, leading to unnecessary energy waste and increased costs.

This issue is particularly pressing for low-income households, who are disproportionately affected by high energy burdens. Research shows that one in four low-income households spends over 15% of their income on energy bills [1]. These households are more likely to reside in older, less energy-efficient homes with inadequate insulation and outdated systems [1]. Furthermore, the U.S. Department of Energy reports that heating and cooling account for nearly half of a typical U.S. home's energy consumption [2]. The impact extends beyond financial strain, as families often face difficult choices between paying energy bills and purchasing other essentials like food and medicine.

According to the U.S. Energy Information Administration's Residential Energy Consumption Survey, space heating, air conditioning, and lighting collectively represent approximately 51% of residential energy consumption [3]. Studies have shown that smart thermostats alone can reduce heating and cooling costs by an average of 10-12% for heating and 15% for cooling [4]. For the average American household spending \$2,000 annually on energy bills [5], these inefficiencies represent hundreds of dollars in unnecessary expenses each year.

Moreover, inefficient energy use in residential buildings contributes significantly to greenhouse gas emissions and climate change. By addressing this issue, we can not only alleviate financial burdens on households but also make a positive environmental impact.

Existing commercial solutions not only come with prohibitive upfront costs but often require professional installation and ongoing subscription fees. Market research indicates that price sensitivity remains the primary barrier to smart home technology adoption, with 42% of non-adopters citing cost as their primary concern [6].

Our project aims to tackle this energy issue by developing a low-cost home automation system that allows residents to control their home's temperature and lighting levels with minimal effort. Our solution will employ affordable smart sensors, automated scheduling, and remote access through a mobile application to optimize energy use, reduce costs, and improve comfort. This approach aligns with the growing trend of smart home technologies, which have shown the potential to reduce energy consumption by 10-30%[7].

By providing an accessible and user-friendly system, we aim to empower a broad range of users, including those in lower-income brackets, to make energy-efficient choices for their homes without the need for expensive installations. Users can visualize their consumption patterns, identify opportunities for

savings, and implement changes that reduce both their environmental footprint and monthly expenses. Through data collection and reports, users will gain insights into their energy consumption patterns, enabling them to make informed decisions and further reduce their energy footprint.

Ultimately, our project seeks to address not only the immediate need for cost-effective energy management but also to contribute to broader goals of energy conservation, reduced carbon emissions, and improved quality of life for residents across various socioeconomic backgrounds.

In addition to addressing energy inefficiency at a whole-home level, many traditional heating and cooling systems suffer from uneven temperature distribution due to single-zone HVAC control. This leads to unnecessary energy consumption, thermal discomfort, and over-conditioning of unoccupied areas. To address this problem, our system now supports multi-zone environmental control. Using a combination of temperature sensors, software-based control logic, and motorized duct dampers or hydronic valve actuators, the system enables independent temperature regulation in up to eight zones within a single building. This zoned HVAC capability improves comfort, reduces energy consumption, and provides users with precision control over heating and cooling distribution throughout their homes.

1.2. INTENDED USERS

Homeowners

Homeowners have the regular man's problem: the utility bill is too high. All they want is a good home control unit that is affordable, reduces utility costs, and increases the level of comfort in their home. The average homeowner does not need or want a complex, flashy, smart home system.

Example - Reginald Miller

- Reginald needs a way to reduce his energy consumption because his utility bill is too high.
- Reginald needs a way to automate the lights in his house because he hates it when the lights are left on; it might even lead to a divorce.

Property managers

Property managers have long-term financial goals in mind. They require a reliable way to track the energy consumption in their properties to make informed decisions. They need some that will help them save money in the long run.

Example - Joe Swanson

- Joe needs (a way) to track the consumption of energy over time in buildings because he wants to reduce input costs and make informed decisions for future energy upgrades.

Business owners

Just like property managers, business owners are profit-oriented. They are in need of an efficient way to increase profit by lowering energy costs and consumption. Business owners also have their employees in mind and want to make the employees' jobs more comfortable and efficient.

Example - Robert Robinson

- Robert needs a way to lower the energy costs of his business property because he wants to increase his profit margins.
- Robert needs a way to allow each of his employees to easily control the lighting in the main building.

Tech Enthusiast homeowners

Tech enthusiasts have different priorities than regular homeowners. They have an eye for tech and want to upgrade their homes with innovative technology. They need a smart home control unit that is efficient and up-to-date with the latest technology to satisfy their inner computer enthusiast.

Example - Stevie Wonder

- Stevie needs a way to integrate his love of technology into his home because he enjoys being on the cutting edge.

Contractors

Contractors are always trying to find ways to improve their projects. They want to find a way to make the new houses they build more attractive by installing smart control units. However, they need to find an affordable yet effective device that will not drastically increase their budget.

Example - Mike Michaels

- Mike needs a way to make new homes smarter without making significant increases to his budget.

Environmentalists

Environmentalists have sustainability in mind. They want a way to track their home's energy usage to reduce wastage and to live a more environmentally friendly lifestyle. They also benefit if they can find an affordable way to do this tracking.

Example - Alice Wonderwall

- Alice needs a way to minimize her energy usage because she is interested in conserving natural resources.

Homeowners with Multi-Room Temperature Needs

Some homeowners live in larger homes or multi-level houses where maintaining even and comfortable temperatures can be difficult with a single thermostat. These users require a system that allows temperature customization in different areas without wasting energy on unused rooms. Multi-zone HVAC control enables efficient heating and cooling by opening or closing duct dampers or hydronic valves based on room usage and user preference.

Example – Sarah Johnson

- Sarah lives in a two-story home where the upstairs bedrooms are often too warm while the basement family room stays cold.
- Sarah needs a way to independently control the temperature in different parts of her home without installing multiple HVAC systems.
- With multi-zone temperature control, Sarah can keep the bedrooms cooler at night, warm the basement only when it is in use, and reduce unnecessary heating in unoccupied areas—saving energy while improving comfort for her family.

2. Requirements, Constraints, And Standards

2.1. REQUIREMENTS & CONSTRAINTS

Functional Requirements

Installation

- Fast to Install
 - The system should be designed for quick and straightforward installation.
 - Easy-to-understand documentation will be provided to make the installation process as simple as possible.

Communication

- Wireless Communication
 - The system should support wireless communication such as Wi-Fi, Zigbee, or Z-Wave.
 - Ensure compatibility with standard IoT communication protocols.

Power Management

- Easy to Power
 - A battery or a simple outlet plug should power the system.
 - Energy-efficient operation to minimize power consumption.

Reliability

- Good Signal Integrity
 - The system should maintain strong and stable communication signals throughout the operation.
 - Error detection and correction mechanisms should be in place to prevent data corruption and data loss.

Consistency

- The system should provide stable and repeatable performance under varying environmental conditions.

Control

- Able to Control Line Voltages

- The system should be able to control electrical loads at different voltages, ensuring compatibility with various appliances and industrial systems.
- Implement safety features such as surge protection and automatic shutdown to protect the components.
- Multi-Zone Temperature Control

The system shall support temperature control for up to **eight independent HVAC zones** to enhance comfort and improve energy efficiency.
- Motorized Damper and Valve Control

The system shall support both **duct dampers** (forced air systems) and **hydronic valves** (boiler or radiant systems), depending on the user's house configuration.
- Relay-Based Actuation

The zone controller shall use **normally open relays** to control damper or valve actuators, ensuring safe airflow in the event of power failure.
- Zone Coordination Logic

The system shall prevent zone conflicts by coordinating HVAC calls to avoid equipment strain, pressure buildup, or excessive cycling.
- Fail-Safe Operation

The system shall default to safe operating conditions (e.g., opening all dampers) in the event of control or communication failure.

Compatibility

- Works with All Types of Buildings
 - The system should be adaptable to different building structures, including:
 - Old Buildings – Ensure the design works well with old electrical components used in older buildings.
 - New Buildings – Designed for integration with modern construction.
 - Large Buildings – Capable of handling multiple floors and large areas.
 - Small Buildings – Work well in smaller buildings with many communications bouncing back and forth.

Security

- Secure from Hackers
 - Implement encryption and secure authentication methods to protect against unauthorized access.
 - Ensure locally stored data is protected from tampering or unauthorized retrieval.

Accuracy

- Energy Monitoring

- The system should provide precise energy consumption data and graphically display it.
- Calibration mechanisms should be in place to measure the accuracy of components.

Resource Requirements

Hub Software

- Local Server Compatibility
 - The central hub software should run efficiently on a local server while maintaining a connection to an external cloud server.
 - Ensure minimal resource consumption while maintaining high performance.

Sensors

- Low Power Consumption
 - Sensors should be designed for minimal energy usage to prolong battery life or reduce dependency on external power sources.

Physical Requirements

- Compact Design
 - The device should be as small as possible to allow easy installation and avoid obtrusiveness.

Aesthetic Requirements

- Blends into Different Styles
 - The design should allow seamless integration into various home decorating styles, from modern to traditional.
- Visually Pleasing
 - The device should maintain a sleek and elegant appearance.

User Experience Requirements

- Affordable Pricing
 - The system should be cost-effective and offer competitive pricing to ensure accessibility for a wide range of users.
 - Balance cost with high-quality components and reliable functionality.

Project Constraints

- Timeline
 - Our project is limited in the time we are given to complete it.

- Given only two semesters may lead to un-added features and capabilities.

- **Budget**
 - Components for designing and implementing smart home devices are not cheap, and the team will be relying heavily on ETG for most of our equipment.
 - ETG may be unable to supply the best and most efficient hardware for our project.

- **HVAC Compatibility Constraint**

The zoning subsystem must be compatible with **conventional 24VAC thermostat control systems** commonly used in residential furnaces and air handlers. Systems using **proprietary HVAC control buses** (e.g., Carrier Infinity, Trane ComfortLink) are not supported in the first prototype.

- **Safety Constraint**

The relay control circuitry must comply with **low-voltage wiring safety practices** and include isolation to prevent damage to HVAC equipment.

2.2 ENGINEERING STANDARDS

Importance of Engineering Standards

Engineering standards are standardized standards and specifications adopted to guarantee that products, services, and systems are safe, reliable, and perform as intended. They allow for interoperability between different devices and systems, improve safety, and assure quality across industries. Adherence to these standards by engineers and manufacturers can assure that products meet known standards, simplify design processes, reduce costs, and increase user confidence. For instance, for smart appliances in the home, standards like IEEE 802.11 (Wi-Fi) ensure that appliances of different vendors can communicate seamlessly among themselves and give an integrated user experience.

Relevant Standards for the Smart Home Control Unit

IEEE 802.15.4: Zigbee

This standard defines the physical and media access control layers for low-rate wireless personal area networks (LR-WPANs). IEEE 802.15.4 is employed by Zigbee to offer a low-power and secure mesh network that has extensive application in smart homes due to the low power consumption and high density, thus leading to an enhanced and efficient networking solution.

IEEE 802.11: Wi-Fi

IEEE 802.11 is a set of WLAN standards. Wi-Fi offers a method through which devices are able to get internet connectivity and share information through a wireless medium. For a smart home, Wi-Fi offers fast information transfer between the controller unit, sensors, and interfaces for the enablement of such functionality as real-time monitoring and remote access.

ISO/IEC 30162: Internet of Things (IoT) Device Testing

This standard defines the testing and assessment of IoT devices for compliance with performance, safety, and security specifications. Adherence to ISO/IEC 30162 ensures the smart home control unit's reliability in an IoT setting while being interoperable and industry best practice compliant.

Relevance of Selected Standards to the Project

IEEE 802.15.4: Zigbee

The low power usage and mesh networking of Zigbee make it highly relevant to network many sensors and devices in the smart home control system. Its ability to support many devices in a network offers scalability and flexibility in system design.

IEEE 802.11: Wi-Fi

The popularity and high rate of Wi-Fi data transfer are crucial in enabling seamless communication between the control unit, cloud services, and user interfaces. Its integration ensures real-time monitoring and control, enhancing user experience and system responsiveness.

ISO/IEC 30162: IoT Device Testing

Following this standard ensures that each IoT device within the smart home system is extensively tested for performance, security, and compatibility. This conformity minimizes potential issues, improves reliability, and wins users' trust in the system's functionality.

Other Standards

ISO/IEC 27001: Information Security Management

This standard provides requirements for implementing, maintaining, and improving an information security management system (ISMS). In a smart home system, being ISO/IEC 27001 compliant ensures that the user data and functions of the system are secure from threats.

IEEE 1901: Broadband over Power Line Networks

This standard is for high-speed communication over power lines, offering an alternative medium for home communication. It can be highly effective in applications where wireless signals are poor or power lines can be used for data communications.

Changes to Include Chosen Standards

Communication Protocol Integration

Add both Wi-Fi and Zigbee modules to enable flexible modes of communication, allowing devices to use the most appropriate protocol based on their data needs and power constraints.

Security Enhancements

To reduce potential risks, develop and implement a complete information security management system to ISO/IEC 27001 standards with data encryption, access control, and regular security audits.

Create a test framework according to ISO/IEC 30162 to rigorously test device performance, interoperability, and security before release, ensuring all components meet the established standards and coexist peacefully in the system.

By integrating these standards into project design, the Smart Home Control Unit for Energy Optimization will enjoy enhanced reliability, security, and user satisfaction, a stable product in the smart home market.

In addition to communication and IoT standards, the zoning system design must also conform to HVAC control safety standards and relay control best practices. Specifically, **UL 873 – Standard for Temperature-Indicating and Regulating Equipment** applies to safe temperature control in HVAC systems, and **IEEE C37.90** guides proper electrical relay switching practices. These standards inform relay selection, wiring safety, and control system behavior to ensure HVAC equipment protection and user safety.

3 Project Plan

3.1 PROJECT MANAGEMENT/TRACKING PROCEDURES

Project Management

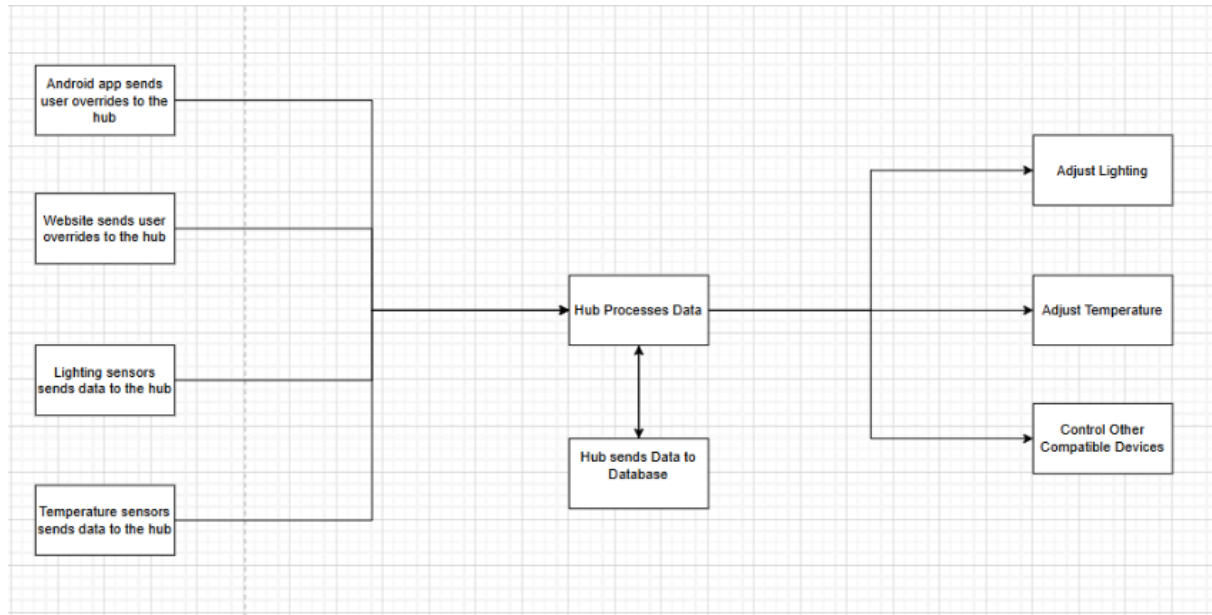
The group decided to follow an agile approach for this project. The agile approach was the most familiar to the group, which was one reason it was chosen. The project being developed used a variety of different hardware components, and breaking these areas down into smaller sprints helped the team manage the tasks. With the agile approach, the group met frequently and checked each other's work to ensure good practices were followed. This also allowed the team to make changes to its strategy and discuss topics regularly, enabling quick adjustments to the project.

As the project scope evolved, an additional subsystem was introduced: an 8-zone HVAC control module utilizing relay-driven motorized dampers or hydronic valves. The integration of this feature was incorporated into the agile project plan by treating it as a new feature epic within sprint cycles. This ensured zone control development progressed in parallel with sensor integration and software configuration while maintaining incremental subsystem deliverables and continuous testing.

Tracking Procedures

For tracking, the team used Jira for monitoring progress and assigning tasks. This allowed the group to see the entire process and understand what everyone was working on. Using Jira provided a progression track for the team and laid the groundwork for a successful project. The team also used Git for version control, which allowed collaboration without conflicts and supported efficient development while maintaining backups.

3.2 TASK DECOMPOSITION



The above illustration shows how our group has decided to split the project into subtasks. Each box represents a part of the project that must be completed in order to develop the main components. As you can see, everything is connected in some way or another, which shows how important each individual cell is for success. Now that we have split the project into parts, we can start to develop a schedule and plan to complete the work. This task decomposition also allows both the hardware and software teams to see how the different pieces come together.

As the design evolved, the team added **HVAC Zone Control** as an additional subsystem to support multi-room temperature regulation. This task includes integrating relay-based actuation for duct dampers or hydronic valves and coordinating zone behavior through Home Assistant. The new block is functionally related to both the hardware (sensor/relay control) and software (automation logic) tasks already depicted in the task decomposition structure.

3.3 PROPOSED PROJECT MILESTONES, METRICS, AND EVALUATION CRITERIA

The deliverable product that we need to have at the end of the project is a fully functional set of sensors, a web interface, and an app to provide input to a hub. In turn, the hub sends data to the database and makes the necessary adjustments to the temperature and lighting level. The hub is the lynchpin of the project. Thus, setting up a functional hub is the first milestone of the project. Without any sensors or other project deliverables to test it with, the criteria for a functional hub are one that we can access and that has Home Assistant configured on it. Once the hub is configured, the group splits into teams to work on the web/database and hardware sides of the project. Each team performs its own configuration of the hub as needed.

The next milestone is functional light and temperature sensors. The sensors must connect to the hub and provide accurate temperature and light level readings, respectively. Sensors must also maintain a reliable connection. The sensor's accuracy will be tested against standalone sensors and should, initially, be within a margin of +/- 1% of the standalone sensor's reading. We are still in the process of choosing a protocol for connecting the sensors and the hub, which necessitates waiting to establish reasonable evaluation criteria for the connection. Though both types of sensors will be evaluated using the same basic criteria, the temperature sensors will be completed first, followed by the light sensors.

Following sensor validation, the next milestone introduces **HVAC zone control capability** to the system. This milestone focuses on developing and testing a relay-driven zone controller that can independently regulate airflow or water flow to up to eight zones using either **motorized duct dampers** (forced air HVAC systems) or **hydronic valves** (boiler or radiant systems). The zone controller must interface with Home Assistant and accept commands based on temperature readings from individual zones. Successful completion of this milestone will be measured by the ability to actuate individual zones on command, maintain default **normally-open, fail-safe behavior**, and verify communication reliability using Zigbee.

Concurrently, another team will be working on the database. The database must have reliable internet communication with the hub and store the data such that it can be easily accessed and used. The database-to-hub connection must also gracefully handle disruptions in service from external factors, like power and wifi outages.

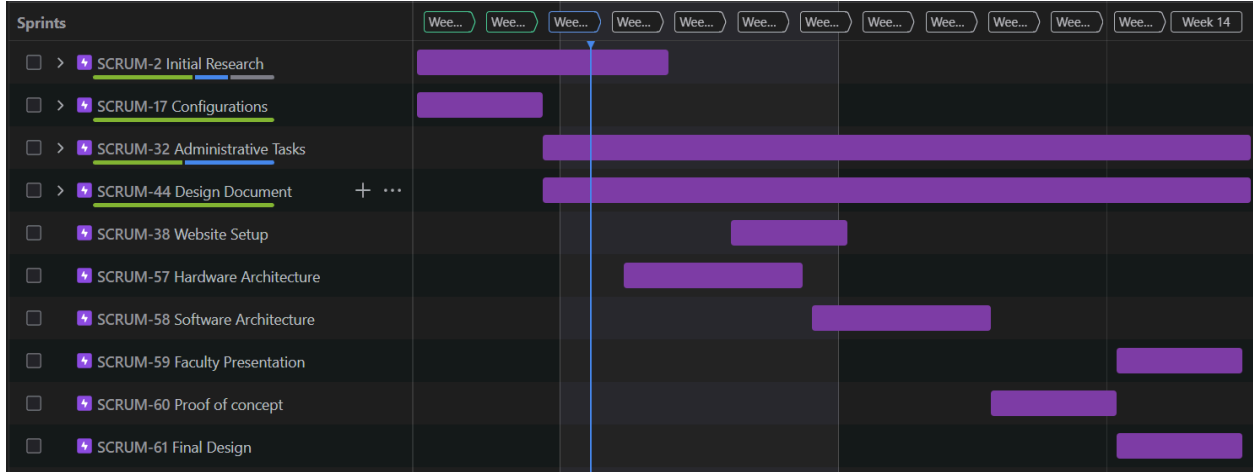
The next pair of milestones is the light/temperature adjustments and implementing the local and remote web interfaces. The light and temperature controls must connect to the hub, reliably affect the changes in light or temperature, and accurately communicate their status with the hub. The web interfaces must reliably interact with the hub and sensors to override sensor presets so the temperature and light level will be consistent with the user's preferences. Both web interfaces will be the same, except one will run on the remote server and one on the hub. The local web interface that runs on the hub must provide the same user experience with or without the internet and cause no errors that interrupt the function of the hub.

After ensuring the adjustments work correctly and implementing the web interfaces, the hardware team will ensure that other devices that can be managed with Home Assistant will play nicely with our

sensor and temperature management system. The other sensors must not cause any disruption or interference with the rest of the project and perform their own functions as designed. The web team, meanwhile, will implement the app interface. The app interface will be evaluated using the same criteria as the remote web interface since it performs the same functions.

The final milestone of the project will be the completion of the entire, top-to-bottom smart home system, the report, and the presentation. The report and presentation will be evaluated against the rubric for the class. The smart home system must accurately follow the user’s preferences to maximize efficiency. The hub, sensors, and remote server must maintain reliable communications, and if the internet connection to the hub goes down, the system must continue to function using only the local hub and web interface.

3.4 PROJECT TIMELINE/SCHEDULE



We have finalized our project timeline for this semester. The above Gantt chart breaks down how we will complete the design process and presentation. Per the plan, we expect to complete all of this semester’s work by week 16.

In the fall semester, our foremost objective is to complete the smart home system. As part of our proof of concept, we expect to completely set up the hub and at least one sensor connection this semester. Starting in the fall, we expect to spend the first one or two weeks reacquainting ourselves with the project and the work we have to do.

In Weeks 3 and 4, the hardware team will begin developing and configuring temperature sensors, validating their accuracy within a +/-1% margin compared to standalone sensors, and ensuring stable connectivity to the hub. Concurrently, the web/database team will set up the database, establish communication between the database and the hub, implement basic data storage and retrieval functions, and ensure resilience against external disruptions such as power and network outages. By the end of this phase, the system should have working temperature sensors and a functional, reliable database. During this phase, preliminary work will also begin on integrating **Zigbee-based HVAC zone control hardware**

to ensure compatibility with the existing sensor network prior to full zone logic implementation in the following milestone.

Weeks 5 and 6 will be dedicated to finalizing light sensors and further testing the database. The hardware team will develop and configure light sensors, ensuring their accuracy and reliable connection to the hub. Meanwhile, the web team will conduct stress tests to confirm the database's ability to handle connection disruptions and optimize data retrieval performance. The expected deliverables at this stage include fully functional light sensors and a robust database capable of handling real-world conditions.

Moving into Weeks 7 and 8, the hardware team will shift focus to implementing light and temperature control mechanisms, ensuring that adjustments reliably affect environmental conditions and communicate their status back to the hub. Simultaneously, the web team will develop early versions of both the local and remote web interfaces, which must allow users to override sensor presets and manually adjust temperature and lighting settings. By the end of this phase, the system should have working control mechanisms and prototype web interfaces integrated with the hub.

Weeks 9 and 10 will be dedicated to refining the web interfaces and ensuring compatibility with additional Home Assistant devices. The hardware team will verify that these additional devices do not interfere with the core functionality of the project. At the same time, the web team will finalize the local web interface and implement the remote web interface, ensuring both provide a seamless user experience. By the end of this stage, all web interfaces should be fully functional, and additional Home Assistant devices should be successfully integrated.

During Weeks 11 and 12, the web team will implement the mobile app interface, ensuring it mirrors the functionality of the remote web interface while optimizing for usability and reliability. Meanwhile, the hardware team will conduct system-wide stability and integration testing to confirm that all components work together as expected. The deliverables for this phase will include a fully functional app interface and a comprehensive system integration report.

Finally, in Weeks 13 and 14, the project will undergo final testing, including verifying that the smart home system remains operational during internet outages and conducting user experience evaluations. The team will also complete the final project report and presentation, ensuring that the deliverables meet the class rubric requirements. By the end of the project, the smart home system should reliably manage temperature and lighting based on user preferences, maintain communication between all components, and function seamlessly even in offline mode. After completing the hub and finishing all the testing, we will take the next week to finalize the presentation and then present it the week after.

3.5 RISKS AND RISK MANAGEMENT/MITIGATION

Hardware Compatibility Issues

Risk: The sensors and control components may not be fully compatible with the hub or require additional configuration.

Mitigation: We will conduct thorough research and compatibility testing before finalizing hardware choices. We will also use standardized communication protocols like Zigbee or Z-Wave to improve interoperability.

Probability: Medium ($\approx 40\%$) – You are using multiple microcontrollers, relays, dampers/valves, and Zigbee/Z-Wave devices, so at least some configuration or wiring mismatch is quite likely, but you are already planning standards-based parts (Zigbee, 24VAC, etc.).

Software Integration Challenges

Risk: Integrating Home Assistant, the database, and the web/app interfaces may introduce unforeseen compatibility issues.

Mitigation: We will implement a modular development approach, testing individual components in isolation before integrating them. Frequent debugging and iterative testing will help address any integration issues early.

Probability: High ($\approx 60\%$) – Home Assistant + custom ESP32 firmware + database + web/app + Zigbee HVAC zoning is a complex stack, and you explicitly note modular integration and lots of testing, which implies non-trivial integration risk.

Sensor Accuracy and Reliability

Risk: Temperature and light sensors may produce inaccurate readings, affecting system efficiency.

Mitigation: We will calibrate all sensors against industry-standard benchmarks and conduct repeated accuracy tests. Additionally, we will implement software-based correction factors to adjust for minor discrepancies.

Probability: Medium ($\approx 35\%$) – You specify tight accuracy goals and calibration plans, but low-cost sensors plus wireless links and battery power create real risk of drift or intermittent failures.

Network and Connectivity Failures

Risk: The hub relies on a stable internet connection for remote control and data synchronization, and service disruptions could affect performance.

Mitigation: The local hub will be designed to function independently of an internet connection, ensuring that core automation features continue working offline. A backup power solution will also be considered for critical operations.

Probability: Medium ($\approx 30\%$) – Home Assistant and remote access depend on Wi-Fi and internet, and you call out offline operation as a design goal, which suggests outages are expected but partially mitigated by a local-only mode

Security and Privacy Concerns

Risk: Unauthorized access to the system could compromise user data and allow malicious control over the smart home environment.

Mitigation: We will implement robust encryption for data transmission and user authentication measures. Regular security audits will be conducted to identify vulnerabilities and ensure compliance with best security practices.

Probability: Medium ($\approx 30\%$) – Any internet-connected smart home has real security exposure, but you explicitly plan encryption, authentication, and audits aligned with standards such as ISO/IEC 27001, which lowers the likelihood compared to an ad-hoc design

Time Constraints and Project Delays

Risk: Delays in component procurement, unforeseen technical challenges, or team scheduling conflicts may impact progress.

Mitigation: We will follow an agile development methodology, regularly updating our timeline and prioritizing critical tasks. Buffer periods are built into the schedule to accommodate unexpected delays.

Probability: High ($\approx 65\%$) – Two-semester schedule, dependence on ETG equipment, new HVAC zoning subsystem, and the number of milestones all point to a high chance of slipping individual tasks even if the overall project finishes.

Budget Overruns

Risk: Costs associated with hardware procurement, software licensing, or cloud services may exceed initial estimates.

Mitigation: We will track expenses closely and explore cost-effective alternatives where possible. We will communicate with stakeholders early to adjust plans accordingly if additional funding is required.

Probability: Medium ($\approx 40\%$) – You rely on ETG and low-cost hardware, but zone hardware (dampers/valves, transformers, relay boards) and multiple dev boards can easily exceed initial estimates if parts fail or need upgrading.

Summary

The risks identified above reflect both technical and logistical challenges inherent to developing a distributed smart home system. The addition of HVAC zone control introduces new hardware and safety considerations, which are mitigated through fail-safe relay design and progressive subsystem testing. The team will reevaluate risk levels at the end of each sprint to respond to evolving project conditions, in alignment with Agile practices.

3.6 PERSONNEL EFFORT REQUIREMENTS

Task	Description	Time dedicated
Research	Conduct research on hardware, software, connectivity, and other elements of the project.	50 Hours
Hardware Integration	Build devices such as the temperature, light, and presence sensors and the hub needed for the project using the hardware components selected through research.	80 Hours
Software Integration	Build the software that runs the functions of our Smart Home Control Unit using the open-source project, Home Assistant. Set up the database, establish communication between the database and the hub	80 Hours
Wireless Connectivity	Figure out the wireless connectivity between all the devices. The goal is to make this seamless and efficient.	80 Hours
Web Interface & Mobile App Development	Develop a web interface that allows users to override sensor presets and manually adjust temperature and lighting settings. Develop a mobile app for this web interface.	80 Hours
Testing & Validation	Run tests on all the components of the Smart Home Control Unit and validate their functionality.	50 Hours

3.7 OTHER RESOURCE REQUIREMENTS

The following are the other resources we require to complete our project. This list is subject to change and growth.

Hardware Resources for Sensor

Xiao ESP32C6
Xiao nRF52840
Beetle ESP32C6 Mini
ESP32 Relay Board
Seeed Studio Static Human Presence Sensor
TMP117
TEMT6000
DFRobot RD-03
HLK LD2420
Zigbee Coordinator

Zone Control Hardware

- 8-Channel 24VAC Relay Board (fail-safe, normally open configuration)
- Motorized HVAC Dampers (8-inch, normally open) *or* Hydronic Zone Valves (24VAC)
- 24VAC HVAC Transformer for zone power
- Inline fuse and isolation modules for relay board protection

Other Hardware Resources

Raspberry Pi 5
Intel NUC
PCB Boards
Wires/ Buttons
3D Printing Material

Software Resources

Home Assistant
Espressif SDK
Espressif Zigbee SDK

Testing Resources

Gitlab

4 Design

4.1 DESIGN CONTEXT

4.1.1 Broader Context

Area	Description	Examples
Public health, safety, and welfare	Our project enhances well-being by reducing the financial and physical strain of high energy costs, particularly among vulnerable groups. Automated lighting and heating make environments safer, reducing fire and electrical risks.	Reducing heating bills in winter for low-income families, minimizing the risk of unattended appliances.
Global, cultural, and social	By developing a solution that is affordable to various socioeconomic groups, we promote equitable access to energy-saving technology. The system also demonstrates respect for user preferences through override options and privacy protection.	Supporting sustainability goals in varied communities, encouraging tech adoption in culturally diverse homes.
Environmental	The system reduces unnecessary heating and cooling by enabling automation and zone-specific control, which prevents energy waste in unoccupied areas. By supporting up to eight independently controlled HVAC zones, the design decreases total system runtime and carbon footprint by aligning energy usage with real occupancy and environmental needs.	Lowering HVAC usage based on occupancy, shutting off airflow to unused zones, and dimming lights during daylight.
Economic	We aim to create an affordable, DIY-friendly smart home system that keeps long-term utility costs at a minimum. The low-cost design makes technology accessible to everyone, promoting economic inclusiveness.	Saving \$200–\$600 annually on energy bills per household; affordable components like ESP32 and Zigbee sensors.

4.1.2 Prior Work/Solutions

There are several commercial smart home systems available in the market, such as Google Nest, Ecobee, and Samsung SmartThings. The systems offer advanced features but are generally costly and need to be professionally installed.

- **Google Nest:** Offers smart thermostat control and energy reports. But it is highly costly to begin with and siphons an enormous amount of user data [8].
- **Ecobee:** Provides occupancy-based thermostat control and remote control. Requires technical installation and is potentially less accessible for budget consumers [9].
- **Samsung SmartThings:** Provides device management and automation functions but requires coupling with a supported platform and a stable internet connection [10].

In comparison to these solutions, our system has the following advantages and disadvantages:

- **Advantages:** Affordable cost, open-source implementation, DIY installation, multi-platform compatibility, and offline support.
- **Disadvantages:** Few sophisticated AI functions, additional user configuration, and manual initial calibration.

4.1.3 Technical Complexity

Our project integrates numerous engineering principles across hardware and software subsystems:

- **Sensor Subsystems:** Employ principles of thermodynamics, photometry, and RF communications (e.g., TMP117, TEMT6000, LD2420).
- **Microcontroller Communication:** Use protocols such as Zigbee and MQTT, requiring networking, low-power design, and real-time responsiveness.
- **Home Assistant Automation:** Involves logical rule creation (using YAML and Node-RED), device state monitoring, and embedded systems software.
- **App/Web UI Development:** Requires front-end development (React Native), secure backend integration, and latency optimization.
- **Energy Optimization Engine:** Leverages scheduling algorithms and will eventually support ML-based prediction for peak-time control.

4.2 DESIGN EXPLORATION

4.2.1 Design Decisions

Software Hub

Deciding on software is essential to our project because we need to ensure that the hardware can run Home Assistant and other programs required to complete the project. The software hub must also be as compact as possible while having the necessary power and memory constraints. As a group, we researched multiple different potential software hubs that work well with Home Assistant, but we decided to go with an Intel NUC because ETG could rent it to the group for free. The Intel NUC has much more power than we predicted we would need, so the group decided it would be helpful to implement the project and downsize to a smaller device like a Raspberry Pi.

Open Source Project

Another critical decision our group had to make was whether we wanted to expand on an open-source project or develop from scratch. Open source was the way to go, mainly to allow the group to focus on the essential features and not waste time creating something that already exists. Going to open source also helps us with external device compatibility, meaning other companies' smart devices can connect to the same software as ours. Our group focused on two leading open-source projects, Home Assistant and OpenHab. These are the two most significant and most supported projects, but as a group, we decided on Home Assistant due to its superior compatibility and ease of use.

Microcontroller Board

Our project also needs a microcontroller to bridge the communication gap between the various sensors and the software hub running Home Assistant. The board can send sensor information to the central hub to conduct logic and send out controller signals. After researching, our group has decided to go with the DFR1117 from DigiKey Electronics.

Zone Control

In addition to the primary system design decisions, the team expanded the HVAC control capability to include **multi-zone temperature management**. This decision was made to address uneven heating and cooling behavior common in multi-level homes and larger residential spaces. Rather than redesigning the HVAC control subsystem, this enhancement builds on the existing relay-based furnace and HVAC integration by adding **branch-level control using motorized dampers or hydronic valves**. This modular extension allows the system to support up to eight independent temperature zones while maintaining full

compatibility with the lighting automation and energy monitoring framework already established earlier in the design.

4.2.2 Ideation

Deciding on a Software Hub

Our group started researching a potential software hub after deciding to use Home Assistant as our Open-Source project. Our first step was looking at the installation page for Home Assistant and seeing what options are included. We created a list of possible software hubs, including Raspberry Pi, ODROID, Intel NUC, Home Assistant Green, and Home Assistant Yellow. Each option had an easy installation process, and the Home Assistant Yellow and Green had it pre-installed. The basic requirements were simple: we needed a device capable of running Home Assistant, something small and affordable, and a device with the correct hardware to ensure it would run smoothly with the rest of the system.

As the design evolved, an additional ideation effort focused on extending temperature control from a single-zone HVAC configuration to a distributed zoning architecture. The team considered three alternatives: upgrading to a commercial zoning controller (e.g., Honeywell TrueZONE), using smart vents for room-level airflow control, or implementing a custom zone controller using **motorized HVAC dampers or hydronic valves**. Commercial zoning controllers were dismissed due to cost and proprietary wiring limitations, while smart vents were eliminated due to airflow restriction and noise concerns. The team ultimately determined that a **custom zone control subsystem** using **ESP32-C6 Zigbee endpoints and 24VAC relay actuation** offered the best balance of scalability, compatibility, and flexibility. This approach integrates naturally with the existing lighting dimmer subsystem and temperature control modules while preserving the project's emphasis on affordability and do-it-yourself installation.

4.2.3 Decision-Making and Trade-Off

After making a list of potential software hubs, each group member was assigned a different option to research and compile all the information they could find on the system. This research included hardware specifications, potential upgrades, and a pros and cons list. When the study was concluded, the team was given time to review all the findings and different options and formulate their own decision on the best choice. Then, a meeting was held for a discussion to allow a team member to talk about their pick and the reasons behind it. Finally, there was a vote that was decided by the majority to determine the software hub we would use going forward.

Throughout the decision-making process, trade-offs were evaluated based on cost, system complexity, user installation feasibility, and long-term scalability. The custom lighting dimmer was selected over commercial smart switches because it provided finer real-time control, lower cost per unit, and did not rely on vendor-specific APIs. Similarly, the introduction of HVAC zoning was pursued only after determining it could be added without increasing system complexity beyond the capabilities of the existing control architecture. While commercial zoning solutions offered convenience, they imposed budget constraints and reduced flexibility. The chosen approach—an ESP32-C6-based zone controller using Zigbee communication—maximized interoperability with the main control hub while preserving the

modular nature of the system design. These decisions reflect a balanced engineering approach that prioritizes extensibility and practical deployment.

4.3 FINAL DESIGN

4.3.1 Overview

Our home automation system encourages household energy consumption optimization through intelligent automation. In contrast to most systems that are worried about user comfort, this system focuses on sustainability, efficiency, and reducing the wastage of energy.

The core Smart Home Hub is the system's brain that automatically uses information from smart devices like thermostats, lights, appliances, and presence sensors to automatically define optimum energy consumption levels. The system learns about household behaviors over time and pre-emptively makes decisions to conserve energy without requiring continuous input from the user.

For instance, lights automatically switch off or become dim in an unoccupied room, heating and cooling are governed by weather conditions and room usage patterns, and appliances work during off-peak hours. While users retain the ability to control equipment using voice or an app manually, the system is based on working proficiently with minimum intervention.

4.3.2 Detailed Design and Visuals

This system is composed of the following five interconnected subsystems:

1. **Smart Home Hub**

- A central controller (Raspberry Pi 4 or equivalent) running a lightweight Linux OS.
- Hosts automation logic, scheduling routines, and communication protocols.
- Interfaces with all connected devices and sensors via MQTT and REST APIs.

2. **Sensor Network**

- Includes occupancy, light, temperature, and door/window sensors.
- Feeds real-time data to the hub for environmental awareness.
- Enables predictive control of lighting, HVAC, and appliance usage.

3. **Device Communication Subsystem**

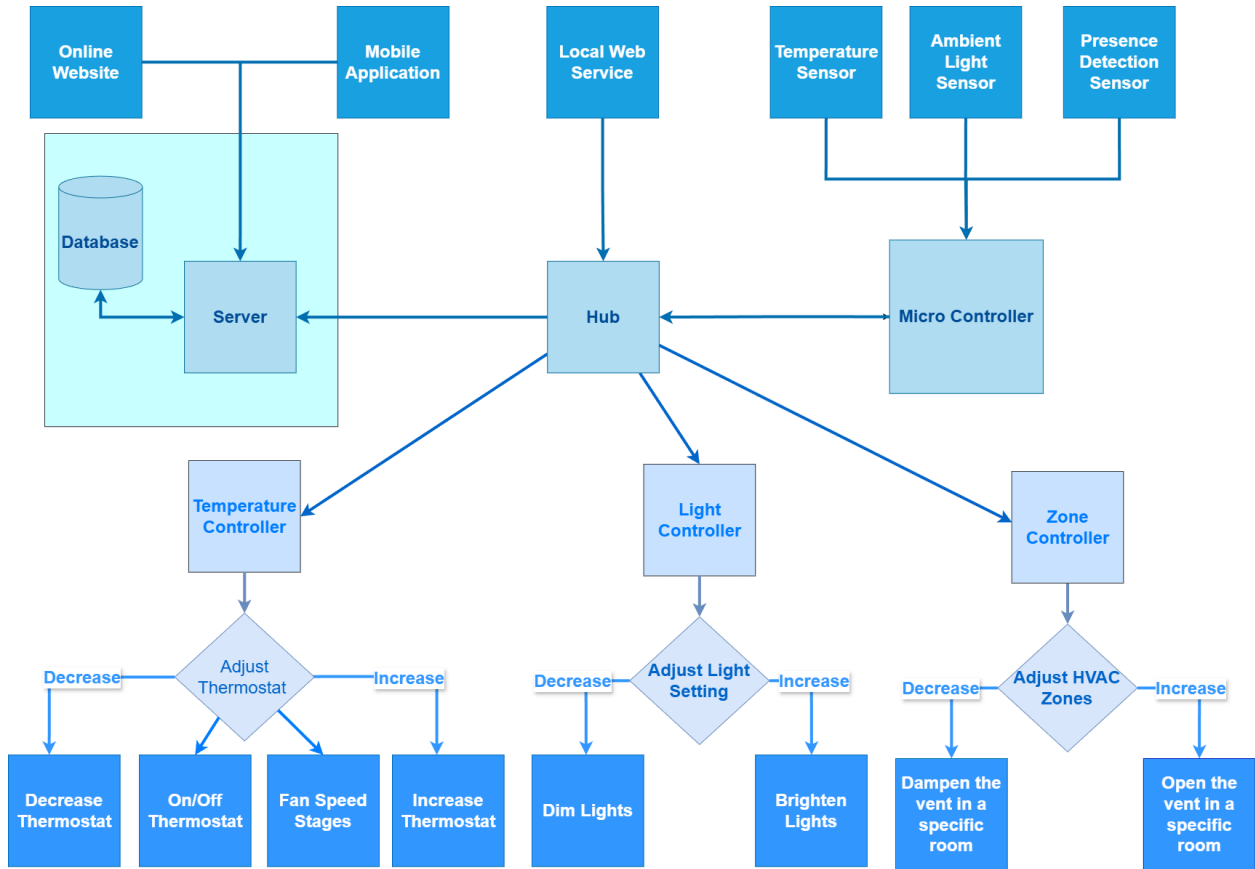
- Supports Zigbee, Z-Wave, and Wi-Fi protocols using USB dongles and open-source libraries.
- Manages bidirectional communication with smart lights, thermostats, smart plugs, etc.

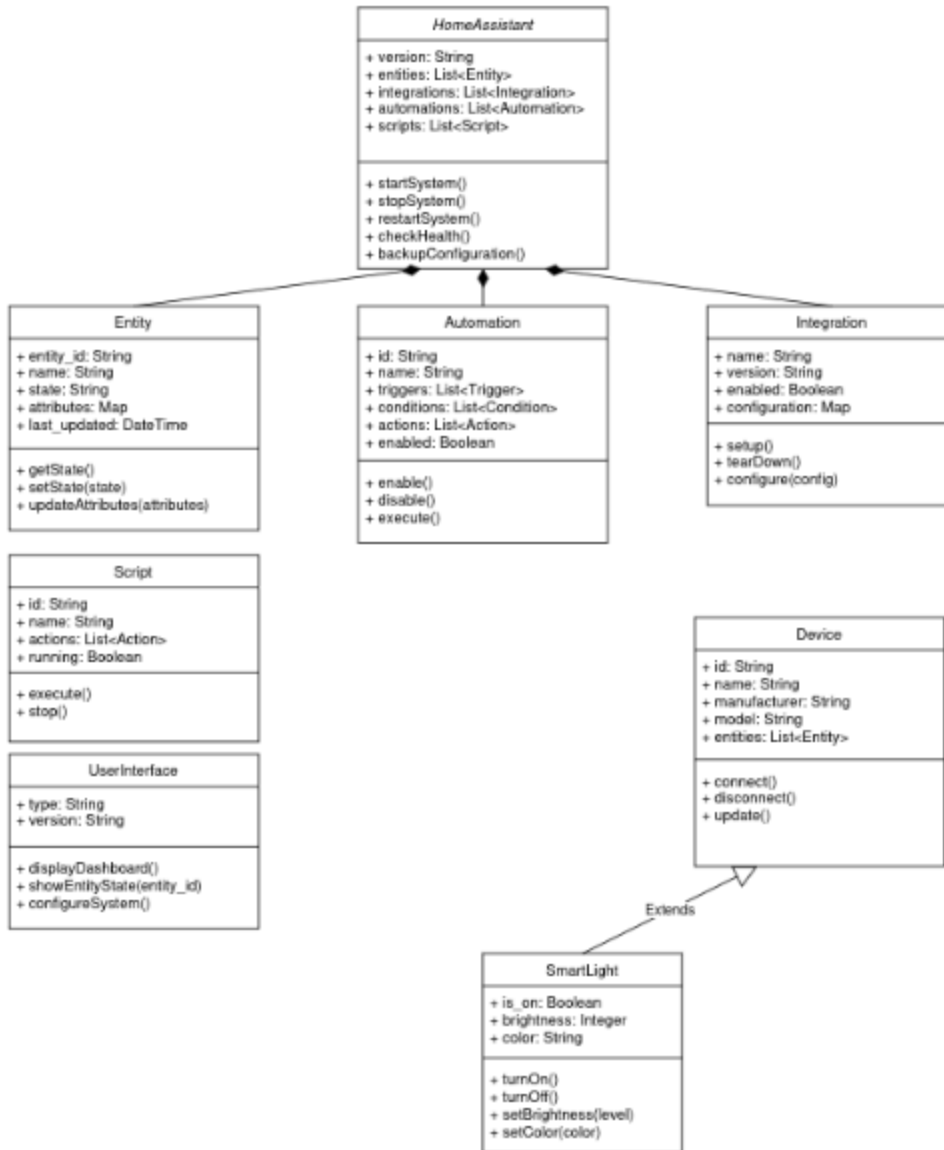
4. **Automation & Energy Optimization Engine**

- A rule-based system using YAML logic and Node-RED flows.
- Future integration of reinforcement learning models for predictive control.
- Capable of load balancing and scheduling based on energy rates and occupancy trends.

5. **User Interface (App/Web)**

- Users can configure automation rules, view energy reports, and manually override settings.
- Built using React Native for cross-platform compatibility.





Subsystem Details:

- **Sensor Fusion Logic:** Combines data from several sensors (e.g., occupancy + ambient light) to make wise decisions (e.g., turn off hallway lights if there is no activity for 5 minutes and daylight is sufficient).
- **Smart Thermostat Control:** Dynamically controls temperature settings according to occupancy and time-of-day patterns. It can be combined with weather APIs for forward-looking control.
- **Appliance Scheduling:** Uses plug-level monitoring and smart plugs to schedule high-power appliances (e.g., dishwashers or dryers) to off-peak times automatically.
- **Energy Monitoring:** Tracks device-level consumption, highlights inefficiencies, and recommends user actions through the app.

4.3.3 Functionality

This system operates with minimal user involvement. Here's a typical usage scenario:

- The user leaves home at 8 AM. Motion sensors detect no occupancy, triggering an automation routine that lowers the thermostat and shuts off unnecessary lights.
- At 2 PM, sunlight is sufficient to light the living room, so the system dims or turns off the lights there, even if someone is home.
- Based on utility rate data and user-defined limits, the dryer is scheduled to run at 3 AM when electricity is cheaper.
- The user receives a weekly report through the app showing estimated energy savings and suggestions for further optimization.

4.3.4 Areas of Challenge

Satisfaction of Requirements

- The current design prioritizes energy efficiency through automation and real-time monitoring. It meets core objectives like reducing energy waste and minimizing manual intervention while remaining extensible for future enhancements.

Primary Concerns

- **Sensor Calibration & Accuracy:** Precise environmental data is critical for effective automation.
- **Energy Optimization Trade-offs:** Balancing comfort (e.g., heating levels) with aggressive energy-saving rules.
- **Data Privacy:** Usage patterns and sensor data must be securely handled.
- **Integration Complexity:** Supporting a wide range of third-party smart devices consistently.

Development Plans

- Perform real-world testing to tune automation thresholds.
- Collect baseline energy usage data to validate efficiency improvements.

- Expand rule engine flexibility and add user-configurable constraints (e.g., temperature minimums).
- Explore integration with utility APIs for real-time pricing.

Questions for Clients/Advisers

- What level of trade-off between energy efficiency and comfort is acceptable?
- Are there specific energy reporting metrics users would find most useful?
- Would solar or battery integration be a relevant future feature?

4.4 TECHNOLOGY CONSIDERATIONS

Hardware

Xiao ESP32C6 is a development board from Seeed Studio, based on the ESP32-C6 chip. We will be using this chip as the microcontroller for the sensors. It is very low-cost and has low power consumption. It supports WiFi 6 and Bluetooth 5 connectivity and is very compact in size. Its downfalls are that it has very limited resources and community support available since it is using a newer version of ESP32.

Beetle ESP32C6 Mini is another development board based on the ESP32-C6 chip made by DFRobot. This also supports Wi-Fi 6 and Bluetooth 5 and has a compact size. It is also cost and power-efficient. It has similar drawbacks to the Xiao ESP32C6.

Xiao nRF52840 is a development board from Seeed Studio, based on the nRF52840 chip by Nordic Semiconductor. It has Bluetooth Low Energy, ideal for low-power, short-range communication. It also has a small compact size. However, Bluetooth Low Energy has a shorter range than Wi-Fi or Zigbee. The nRF52840 chip has less processing power than the ESP32 and contains fewer features.

ESP32 Relay Boards integrate the ESP32 microcontroller with one or more relays. They simplify the control of appliances directly using the microcontroller. They also allow Wi-Fi and Bluetooth connectivity. However, relays can take up more space and consume more power. Relays usually cause electrical noise as well.

Seeed Studio Static Human Presence Sensor is a sensor module that detects the presence of humans without the need for motion. It is battery-powered and very energy efficient. It is easy to integrate. Drawbacks are that these sensors have a very limited range and can be triggered by motion in the environment, not just human presence.

TMP117 is a digital temperature sensor made by Texas Instruments. This sensor has extremely precise measurements and consumes low power. This sensor can be easily integrated into other microcontrollers. This component has no major drawbacks to our project other than the fact that it can not measure humidity.

TEMT6000 is an ambient light sensor from TT Electronics. This sensor has low power consumption and a very simple interface to connect with a microcontroller. This component is also very inexpensive. One drawback of this sensor is that its sensitivity of this sensor is not as accurate in brighter environments.

DFRobot RD-03 is a presence and motion sensor using millimeter wave radar from DFRobot. This sensor has documentation available readily online for integration purposes. This sensor is inexpensive; however, it is relatively more expensive compared to the HLK LD2420.

HLK LD2420 is also a presence and motion sensor that uses millimeter wave radar. This sensor is made by HELINCO. This sensor is cheaper compared to the sensor from DFRobot, RD-03. However, documentation for integrating this sensor is not available in English, which is a major drawback.

Zigbee Coordinator is a key component of a Zigbee-based wireless network made by several companies. Zigbee consumes less energy compared to Wi-Fi. Zigbee works with many ecosystems. One drawback is that Zigbee networks are complex to set up and manage.

Raspberry Pi 5 is a single-board computer and the latest version of the Raspberry Pi series. It has great computational power and is very versatile. This component will be used for the hub of the control unit. This is inexpensive when compared to the NUC from Intel; however, it has less performance.

Intel NUC is a small form factor computer from Intel. This has very high performance and computational power. It is much more pricier than the Raspberry Pi 5 and requires more advanced knowledge to configure.

Software

Home Assistant is an open-source platform for home automation technology. Its major advantage is that it is free to use. Home Assistant supports devices across various ecosystems, including Zigbee, Z-Wave, Wi-Fi, and MQTT, making it very easy to integrate. The interface is very. The only major drawback is that updates and maintenance are required somewhat frequently.

React is a popular JavaScript library used for building fast, interactive user interfaces, especially single-page applications. It allows developers to create reusable UI components that efficiently update and render when data changes, resulting in a smooth and dynamic user experience.

5 Testing

5.1 UNIT TESTING

Temperature, Light, and Presence Sensors:

The team tested the sensors to ensure they read accurate, real-time information. A mock environment was used to simulate temperature and light changes, and the resulting data was compared with other sensors to verify that readings stayed within a reasonable margin of error. Presence detection was validated by simulating humans entering and exiting the mock environment. Long term reliability tests were also performed, where the sensor was run for a long period of time and the values were checked occasionally against known accurate sensors or data. Power Consumption tests were also performed to test the viability of a battery powered sensor. These were done by logging voltage and current draw on running prototype sensors with CPU deep sleep to simulate exact conditions in real world use.

Thermostat Controller:

Testing of the thermostat controller verified that the device correctly read all commands sent from the hub and executed them accordingly. Commands were sent from Home Assistant to the central hub, which then relayed commands to the thermostat controller to turn the system on or off, confirming correct operation.

Light Dimmer Controller:

Light dimming was tested to confirm support for an efficient smart home lighting system. The team verified that commands sent from the central hub were received correctly and that the lights dimmed to the appropriate levels with low response time from command to action. The dimming process was also observed to ensure that lights did not flicker as brightness was increased or decreased.

Home Assistant Logical Rules:

Logical rules configured in Home Assistant were tested to confirm that the system responded correctly as sensor data changed. Rules that activated the thermostat and lighting controllers were triggered using mock sensor data to simulate real-world conditions, and the time from data change to hub signal was measured to ensure acceptable responsiveness.

Web application:

The web application was tested to ensure a seamless and intuitive user experience. Usability tests were conducted, real-time alerts and latency were verified, and feedback was collected from potential users to confirm that the application was accurate and user-friendly.

Central Communication Hub:

The central hub will manage all Zigbee communication sent from the microcontrollers that receive the sensor data. It is vital that the central hub can communicate with all smart devices at all times. Our group plans to test packet loss, signal integrity, and device recognition. We will use Wireshark for packet testing and Home Assistant to check connectivity and device recognition.

5.2 INTERFACE TESTING

Critical Integration Paths:

1. Sensor data → Central Hub → Home Assistant → Controllers
2. User input via web app → Central hub → Controllers

Testing and Tools:

The team used a combination of simulation, real device testing, and monitoring tools to verify the interface behavior.

- **Home Assistant Developer Tools:** Tracing events, testing automation, and monitoring state changes in real-time across the entire interface path.
- **Zigbee2MQTT:** We can monitor and analyze Zigbee communication between devices and the central hub to ensure commands and data are correctly sent and received.
- **Wireshark:** The leading software our team will use to monitor network packets between the web app, central hub, and Home Assistant to verify there is little to no packet loss and delay.
- **Postman and Browser Dev Tools:** Test API calls and observe requests and responses between the web application and the backend.
- **Manual Testing Scripts:** Simulate sensor input and user commands being sent to the central hub.

5.3 INTEGRATION TESTING

Critical Integration Paths:

- **Sensor Data → Central Hub → Home Assistant → Controllers:**
This path is critical as it represents the core automation flow of the smart home. It ensures that sensor input leads to logical decisions by the Home Assistant, which then communicates control signals to devices such as thermostats and dimmers.
- **User Input via Web App → Central Hub → Controllers:**
This path is equally critical as it enables manual control and overrides from the user, maintaining usability and trust in the system. Ensuring low latency and command integrity is essential for a responsive user experience.

Justification for Criticality:

These paths support autonomous and manual control, aligning with key project requirements: real-time responsiveness, user override capability, and seamless integration across subsystems.

Integration Testing Strategy:

1. **Test Cases:**
 - Simulate temperature or light changes using mocked sensor inputs and verify the correct activation of HVAC and lighting systems.
 - Send manual commands from the web app and validate the real-time behavior of devices.
 - Inject errors (e.g., dropped packets, invalid data formats) to observe system recovery and logging.
2. **Tools:**
 - **Home Assistant Developer Tools:** Monitor automation triggers, execution traces, and state changes.
 - **Zigbee2MQTT:** Track Zigbee device communication with the central hub.
 - **Wireshark:** Analyze packet-level traffic between the hub, Home Assistant, and web interfaces.
 - **Postman & Browser Developer Tools:** Validate API response accuracy and timing.
 - **Manual Test Scripts:** Replicate realistic sequences of inputs for end-to-end testing.

3. **Success Criteria:**

- Sensor inputs consistently trigger appropriate controller actions.
- All commands sent from the user interface result in the expected device behavior within 2 seconds.
- The packet loss rate remains below 1%, and invalid commands are gracefully handled.

5.4 SYSTEM TESTING

System-Level Testing Strategy:

The system testing plan built on the completed unit, interface, and integration testing to validate that the smart home system behaved as expected under realistic operating conditions.

Strategy Breakdown:

- **Functional Testing (Driven by Requirements):**

- Verified that all sensors reliably sent data at the specified intervals.
- Confirmed that the thermostat and dimmer controllers responded with latency under 2 seconds.
- Ensured that rules in Home Assistant were triggered correctly and consistently for all defined scenarios.

- **Non-Functional Testing:**

- Verified that the system recovered from network interruptions within 5 seconds
- Confirmed that the web interface maintained usability across different screen sizes and browsers.
- For battery-operated devices, verified that they reported battery status and entered low-power mode when idle

- **System Test Scenarios:**

- Executed complete automation run-throughs during simulated day-night cycles.

- Simulated multiple users issuing commands via the web app simultaneously.
- Simulated device failure (for example, unplugging a Zigbee sensor) and monitored fallback and recovery behavior.
- Performed simulated power cycles and restarts of the central hub and Home Assistant instance and observed system restoration.

Tools:

- Home Assistant logging and dashboards
- Zigbee2MQTT logging
- Selenium or similar UI automation tools (for web app testing)
- Wireshark (for system-wide network analysis)
- Battery simulators or logging for energy-efficient devices

Outcome Expectations:

- The system was expected to perform all intended behaviors without requiring manual intervention during normal operation.
- For each event and command, traceable logs were generated
- No system crashes or undefined states should occur under normal or stressed conditions.

5.5 REGRESSION TESTING

At this point this is a manual process of checking the functionality after a change has been made. Since at this stage in our development process, most of the work is on separate hardware, the need to test the entire system has not yet arisen. Furthermore, the current development has been more expository in nature rather than finished products. This means that our current standard is more along the lines of does it compile, then does this break feature xyz.

However, going forward, we plan to implement an automated testing suite that will ensure new features are tested against previous implementations. The specific tools have yet to be decided, but we will be integrating automated testing with GitLab. We will base our tests on the requirements and use these to ensure that, in the end, we have a well-tested and functional product.

Specific tests will focus on some of the following: Does the sensor still read values correctly? Does the thermostat controller still operate the furnace correctly? Are lights still dimming properly? Does the user interface still display values and permit changes to outputs?

5.6 ACCEPTANCE TESTING

Acceptance testing is an end-to-end evaluation of our project to make sure that it creates the right results given a set of inputs. Additionally, acceptance tests need to ensure that the actuality of the final product matches what our client needs. The means of testing will be different for the functional and non-functional requirements. Where our client's acceptance criteria directly apply, our tests will directly assess how well our final product meets the client's functional requirements.

- Sensors will be tested against a benchmark thermometer/light sensor to provide a reading within a margin of error.
- Temperature controllers must adjust the temperature of a room within a margin of error to the specified temperature.
- The system must keep the light and temperature levels in the home within a certain percentage of the specified temperature and light levels.
- The system needs to apply overrides to the standard preferences.
- The hub must be able to perform its basic function, keeping light and temperature levels in a desired range while maximizing efficiency, without the internet, as well as with an internet connection.
- The database and operational software must preserve their state and recover from a system failure or power loss to function properly afterward.
- The auxiliary devices and hub must maintain a reliable connection under normal operations.

We are also going to test the non-functional requirements, which include the following.

- The GUI needs to be easy to use for the common, non-technical consumer.
- The system must be secured against information leakage and hacking.

- The system needs to recover from issues gracefully, either automatically or by simple user intervention.
- A non-technical user must be able to fully configure and operate the system to meet their needs.

5.7 SECURITY TESTING

Security testing is very applicable to our project. Any device that is connected to a network can present a security risk. It is important to test a system to ensure it is as secure as possible without compromising functionality. Fortunately, the backbone of our project is Home Assistant, which is a well-established and tested system. We are currently planning to use TailScale to facilitate our remote interfaces. Additionally, testing all the necessary layers in a home cybersecurity model (such as the effectiveness of the network firewall) is beyond the scope of this video. Testing the inherent security of Home Assistant and TailScale, existing and vetted systems, is also beyond the necessary scope of our project.

Instead, our security testing will analyze the parts that we are developing ourselves: the sensors, temperature controllers, and the security of how those components interact with the hub on the internal network. We will have to test if the sensors or control components can be used maliciously to take actions on the hub.

Security Testing Plan

- Run security scans, like Nmap and Metasploit, against our hub to find security vulnerabilities.
- Sniff the data transmitted between the hub and sensors to make sure it's not leaking unnecessary information.
- Attempt to connect to the hub as an unauthorized user with default credentials.
- Test the app and web interface for OWASP's top 10 vulnerabilities, including cross-site scripting and injection vulnerabilities.
- Attempt to connect a sensor to the hub without admin authentication.

As the implementation of the project becomes even more complex, we plan to formulate additional tests on the security of our smart home system.

5.8 USER TESTING

User testing confirmed that the system was intuitive, responsive, and easy to operate. Test users successfully viewed data, controlled devices, and interacted with the dashboard without assistance. Feedback verified that the system met expectations for usability and comfort.

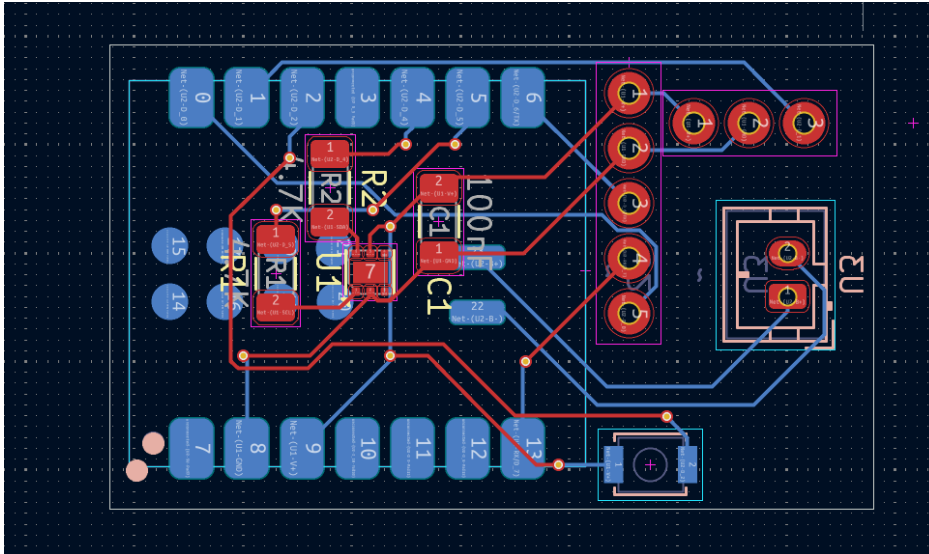
5.9 RESULTS

The team completed all tests outlined in the testing plan and used the resulting data to evaluate the performance, reliability, and security of each subsystem. The test outcomes were used to verify that project requirements were satisfied and to guide targeted improvements to both hardware and software components. Overall, the completed testing effort provided evidence that the product met its defined expectations and was ready for deployment and continued monitoring.

6 Implementation

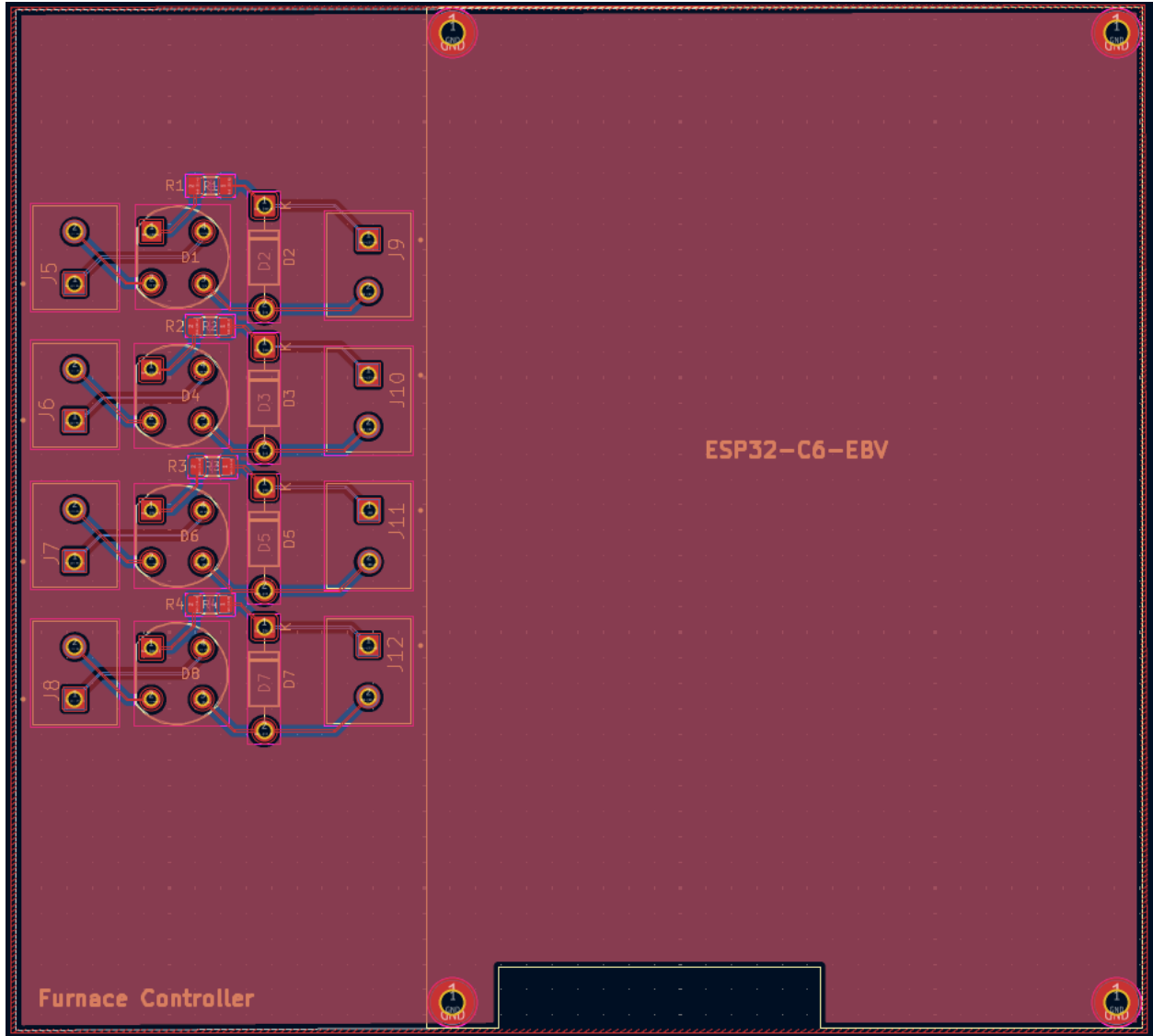
At this point, the group has put in place the basic components of the smart home energy optimization system and constructed a functional end-to-end system. The Home Assistant software runs on an Intel NUC and is the central hub of the system. The hub handles communications, runs the automation logic, and provides both the default Home Assistant interface and a custom React-based interface designed for non-technical users. All the devices in the house (sensors and actuators) talk to the central hub through Zigbee, making the network simpler and eliminating the need for the WiFi network in the control flow. This proves the architecture decision to utilize a central automation server to communicate with a network of distributed Zigbee devices.

For the sensor side, a multi-sensor module has been implemented to offer the environment-sensing input necessary for energy-conscious control decision-making. The multi-sensor module combines presence detection, temperature sensing, and ambient light sensing in a board based on the ESP32 microcontroller, wirelessly communicating with the hub through a Zigbee connection. The existing prototype has demonstrated the capability to provide Home Assistant with information regarding room occupancy, temperature, and light levels, allowing the control algorithm to make decisions based upon the presence or absence of people and light changes in the environment. The report interval for each sensor is adjustable, with a default time of 5 seconds per report. The report will also only be sent to the hub if the value has changed, which saves power. This would allow for a potential battery powered option in the future, if more development was allowed. Both the temperature and light sensor communicate with the micro controller via I2C. While the Presence Detection Currently, the sensor has a USB C input, and can be driven off of any source that can provide 5 Volts 0.1 Amp. Printer circuit board (PCB) designs for the multi-sensor module have been submitted for production and, although the PCBs will be received in the following weeks, they will be too late to implement and test in the current document preparation period and will instead be implemented in a prototype form to illustrate functionality and interaction. The target size for the final sensor package is 1.75" x 1.5" x 0.5" (LxWxH).



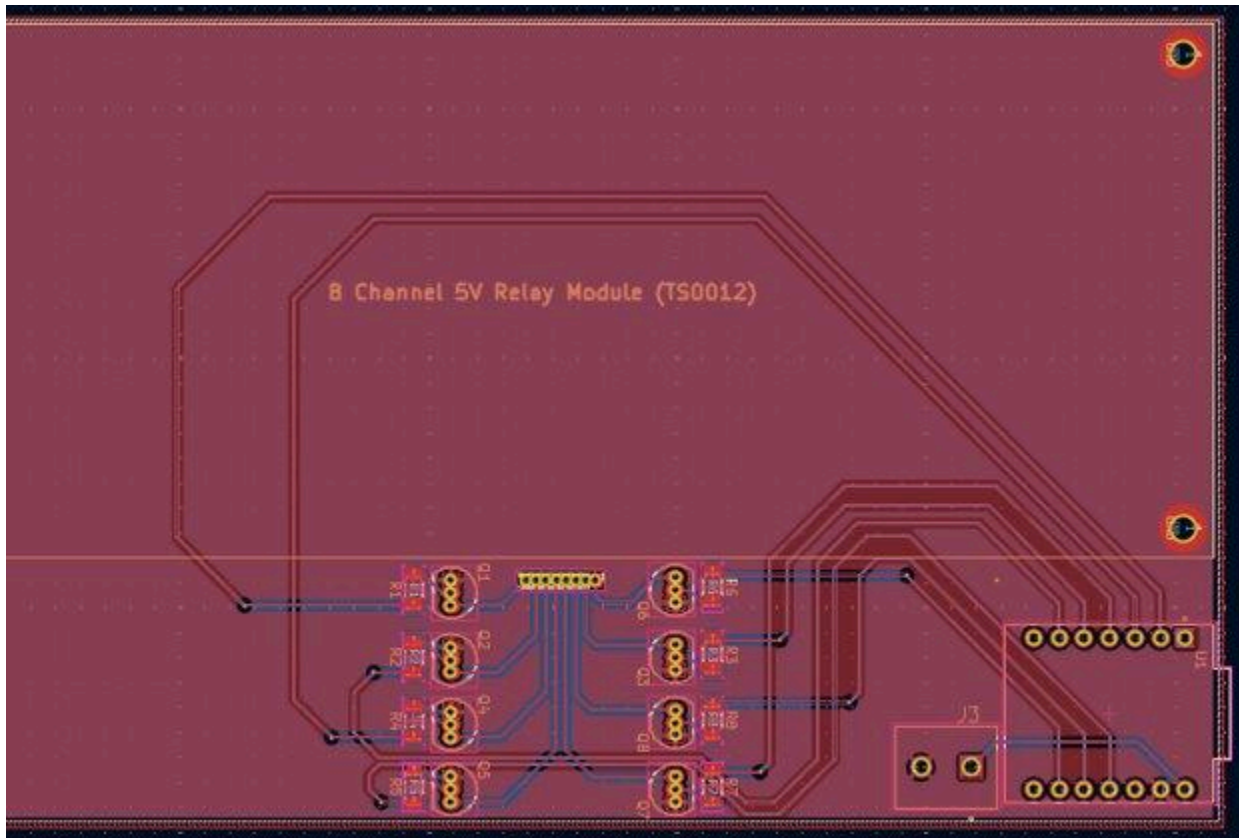
Sensor PCB Design.

For HVAC control, the members have designed and implemented a furnace control module to enable the Home Assistant system to serve as a smart thermostat solution for the house. The control module employs the use of relay components to interact with the HVAC equipment in the house through an ESP32 module equipped with the required entities to be controlled and monitored through the automated hub, communicating through the Zigbee protocol. Through the implemented design, the system allows adjustment and control of HVAC functions according to the predefined rules and sensor data input, for instance, relaxing the temperature range for an unoccupied room and the opposite upon detection of room occupancy. The designed relay circuitry and implementation consider safety standards in HVAC control and relay functionality, such as UL873 for temperature-regulating equipment and IEEE C37.90 for safe relay operation.



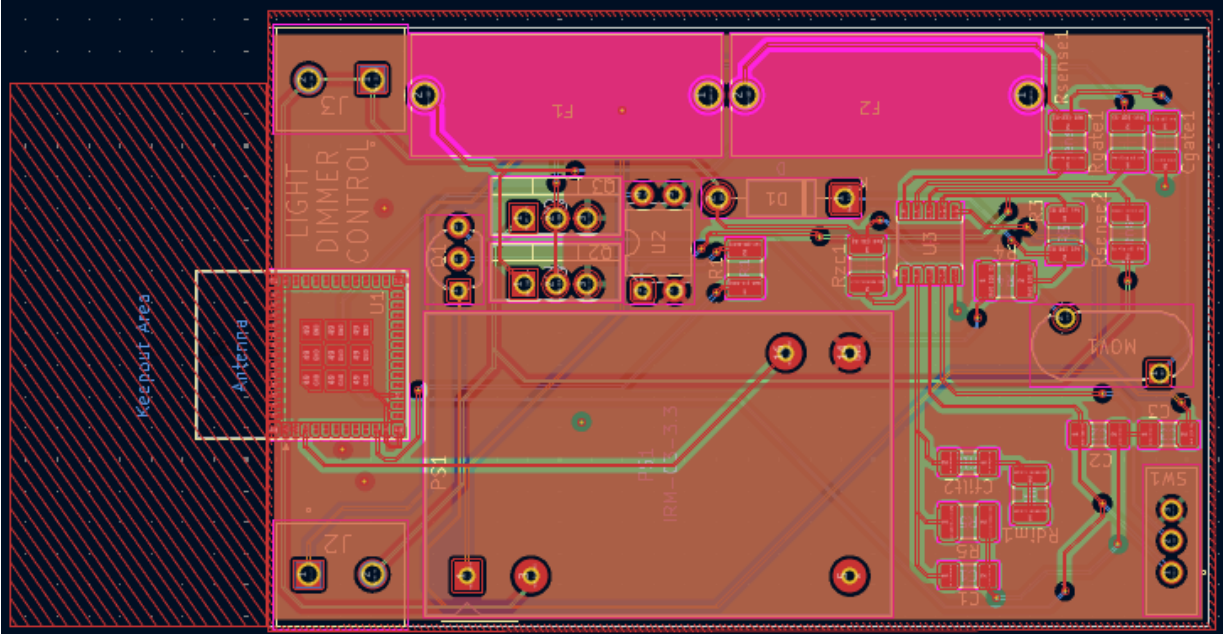
Furnace control PCB design

The HVAC zone controller has evolved from a conceptual design to an operational system module. The functional zone controller constructed and implemented utilizes a Zigbee node made of an ESP32 microcontroller to control the relays connected to numerous duct dampers/valves to adequately control the independent conditioning of the various room spaces instead of a generalized space. The functionality testing of the zone control module has demonstrated the capability to open and close dampers according to command signals triggered by Home Assistant and to control the zone and the furnace module to supply conditioned air to spaces to be conditioned and ignore the rest. The fail-safe functionality has also been evaluated on the test bench and validated. This system proves the design of the multi-zone system conceptualized in the design to be functional and part of the complete control system solution.



Zone controller PCB design

The lighting system is partially functional. The team has developed a custom-built light dimmer controlled by the Zigbee protocol and an ESP32 microcontroller and supporting electronics for the purpose of controlling light intensity. Initial testing reveals that the light dimmer is indeed able to control the intensity of a compatible light source, but more development is needed to mitigate problems with flicker, noise, and the sort before the design will be deemed a complete robust solution for unattended operations in a long-term sense. PCBs for the light dimmer have been designed and ordered for production, but, again similar to the multi-sensor PCBs, they will be too late to be tested and incorporated into the project for this phase of development. The current implementation of the lighting system in the current prototype will utilize the partially functioning light dimmer solution in a lab environment along with Home Assistant automations and existing smart devices if needed.



Light dimmer PCB design

Apart from the hardware components, the implemented system has a number of energy-monitoring automation rules in the Home Assistant platform to bridge the sense and actuate components based on the "sense → decide → actuate" process according to the required functionality. The rules make use of the occupancy and light level information from the multi-sensor to control the furnace controller, zone controller, and the dimmer to turn off the conditioning and the lights in unoccupied space while keeping the occupied space comfortable and well lit. Other more complex functionalities required in the final design phase, such as the energy monitoring interface, the scheduling functionality based on the application level, the mobile app, and better optimization functionality, are only partially implemented and unimplemented due to the project time and scope constraints, respectively. The current implemented system supports the essential standards and best practices outlined earlier in the project lifecycle, mainly the Zigbee based on the IEEE 802.15.4 low-power sensor and actor mesh networking standard and respective HVAC safety standards and the security and testing recommendations based on ISO/IEC 27001 and ISO/IEC 30162 best practices.

6.1 Design Analysis

The developed prototype proves the feasibility and integrity of the design architecture for a Zigbee-based smart home energy optimization system. The data transmission from the multi-sensor to the Home Assistant hub works properly, and the hub is able to make decisions based on this input. The furnace and zone controllers complete the closed loop because the software is able to control the real HVAC equipment through the relay output, meaning the system runs on real hardware and is not simply a simulation. At the systems level, the choice to utilize the Home Assistant software platform as a central

integrator, combined with the ESP32-based Zigbee nodes acting as the intelligent edges, proves to be a valid architecture choice.

Practically speaking, the areas of the design that are the most developed—the hub, multi-sensor, furnace controller, and zone controller—are quite functional and offer a good platform for further optimization in the future. The zone controller has demonstrated in early testing the capability to focus its conditioning efforts on a particular room or zone while letting other areas be a little warmer, in keeping with the aim of the project to minimize energy costs while keeping comfort levels high in the areas that need it most. The presence of a Zigbee network has been stable enough to be effective in testing and development, and the centralization of the automated systems into the hub has made programming and troubleshooting much easier. The implementation of the project, therefore, appears to be well-suited to the protocol used (Zigbee), the platform (ESP32 nodes and a central server), and the software platform architecture.

The current version of the sensor provides accurate data, with high reliability and low power consumption. The prototype sensor has demonstrated the ability to be used for extended periods of time with zero downtime caused by the sensor itself. There are four main categories that can be assessed with the functionality of the sensor. The temperature sensor, the light sensor, the presence sensor and the microcontroller. The temperature sensor currently works as intended, and reports data at the frequency and with accuracy that was desired. The same can be said for the light sensor. It can report accurate data with both implementations of a calibrated sensor with an ADC pin on the microcontroller, as well as a I2C based sensor version that increases production cost, but is more accurate without calibration. For the presence detection implementation, the current mmWave based presence sensor works extremely at eliminating false positives and negatives. While there are several main downsides. The first being that it is impossible to have a deep sleep cycle with this current sensor. The sensor needs 30 seconds of initial power on time in order to adjust to the current room. If the calibration time is lowered below this, the sensor begins to produce false negatives and positives. This means that the sensor cannot be turned on, send a sensor report, and then power off. It needs to be run for 30 seconds at least to gather enough data for the presence detection to work. This severely increases the power draw of the device. This doesn't increase power draw enough to cause issues when the device is wall powered, but eliminates the possibility of long term usability while being battery powered, and having similar battery life to already existing devices. The current devices on the market use a Passive Infrared sensor, or PIR for presence detection, but this can often produce false negatives for rooms with people, who happen to be stationary, (i.e sitting at a desk). In the end, it was decided that having to be wall powered with a mmWave sensor was a better option.

Nonetheless, a number of areas in the design are currently in a state where their functionality is incomplete, making it difficult to assess their performance and the amount of energy saved. The custom dimmer circuitry has been designed and implemented to a certain extent, and while functional testing has been conducted to a certain extent, additional work is required to handle corner cases while providing a safe and flicker-free output for a variety of lighting fixtures and loads. The multi-sensor and dimmer PCBs are designed and submitted for manufacturing but are in a state where the current implementation utilizes the prototypes instead of the production boards. The energy-saving algorithm has a simple design and rules-based functionality and has thus not involved a comprehensive assessment of the energy

consumed before and after implementation. If given more time, a number of changes to the implementation process would be made to improve the design and its assessment as well. The incorporation of the PCB versions of the multi-sensor and the dimmer will make the design more reliable and easier to evaluate in terms of manufacturability and installation difficulty. The increase in the automated logging process for the values of occupancy, temperature, light, and HVAC runtimes will provide a means to quantitatively assess the design in terms of its energy savings effect. Lastly, the systematic design process of the dimmer, first through isolated test beds, followed by the extension of the test bed to be compatible with a variety of light sources, will allow the lighting system to advance from a lab-based solution to a functioning one and ensure the entire design attains its purpose of performing intelligent and occupancy-sensitive energy savings.

7 Ethics and Professional Responsibility

As a team, we define engineering ethics and professional responsibility as the obligation to make technically sound and environmentally conscious decisions in the designing and developing process. For our smart home automation system, this means prioritizing user privacy, safety, accessibility, and sustainability.

7.1 AREAS OF PROFESSIONAL RESPONSIBILITY/CODES OF ETHICS

Area of Responsibility	Definition	Relevant Item from IEEE Code of Ethics	Team Application
Work Competence	Performing high-quality, timely, and skillful work.	Item 6: “maintain and improve our technical competence”	Our team has conducted research on all aspects of the project to ensure we develop quality work without wasting time developing suboptimal prototypes.
Communication Honesty	Being honest and transparent when communicating project capabilities and limitations.	Item 5: “Be honest and realistic in stating claims or estimates based on available data.”	Our team communicates that all aspects of the project at this point are prototypes and are not meant for consumer use.
Health, Safety, and Well-Being	Design that will protect the physical safety and well-being of all users.	Item 1: “Hold paramount the safety, health, and welfare of the public.”	The group is creating 3D-printed cases to encapsulate the electrical components of the project to ensure safety.

Property Ownership	Respecting intellectual and physical property belonging to others.	Item 5: “Credit properly the contributions of others.”	We gave credit to external resources, such as libraries and frameworks used in Home Assistant and open-source ESP firmware tools.
Sustainability	Protecting the environment through efficient engineering.	Item 1: “sustainable development practices”	Our automation aims to reduce household energy waste by automating lighting and HVAC only when necessary, based on sensor input.
Social Responsibility	Creating solutions that benefit society while upholding ethical and lawful behavior.	Item 1: “enhance the honor, reputation, and usefulness of the profession.”	We aim to improve energy efficiency and home comfort, which has a positive environmental impact and quality of life.

Strong Performance Area: Work Competence

Our team has performed particularly well in the area of Work Competence. Despite being a student project, we prioritized building a reliable and well-thought-out system by researching and using industry-relevant tools like Home Assistant and Zigbee. Each team member focused on gaining technical knowledge in their assigned area. This not only ensured the quality of individual modules but also supported our goal of building a scalable smart home ecosystem. Our early implementation of the sensor node and temperature controller shows the effectiveness of our competent and timely work.

Improvement Area: Sustainability

While we’ve made efforts toward energy efficiency through automation, Sustainability remains an area with room for improvement. Our system still uses consumer-grade microcontrollers and components that may not be optimized for long-term environmental impact. So far, we have not conducted a full lifecycle or energy consumption analysis for the devices used.

To improve, our team should evaluate the energy usage of deployed devices, explore more power-efficient alternatives. This will help us ensure that our product is energy-saving and prove the claims that this product will save consumers money over time.

7.2 FOUR PRINCIPLES

Broader Context Area	Beneficence	Nonmaleficence	Respect for Autonomy	Justice
Public Health, Safety, Welfare	Automates lighting and heating to improve comfort and reduce strain.	Reduces risk of fire/electrical hazards by turning off devices when not in use.	Allows manual override of automation to ensure users are in control of their environment.	Prioritizes affordable energy-saving tools for all income types.
Global, Cultural, Social	Offers technology that adapts to user needs across cultures and lifestyles.	Protects user data locally to prevent misuse or cultural discomfort.	Users can configure system settings to fit their personal preferences.	Provides equitable access to smart home features.
Environmental	Reduces unnecessary energy use, supporting sustainability and lower emissions.	The system does not yet optimize for hardware lifecycle or electronic waste.	Allows users to opt into energy-saving modes or disable them based on their priorities.	Contributes to climate justice by enabling energy savings.
Economic	Saves money through energy reduction; components are cost-effective and open-source.	May unintentionally increase electricity use if misconfigured or poorly maintained.	Users control which devices are automated and when, allowing economic flexibility.	Promotes economic inclusion through DIY design and affordability.

Strong Broader Context–Principle Pair: Economic – Beneficence

Our project strongly aligns with Beneficence in the Economic context. By designing an affordable system using open-source software and inexpensive components, we make smart home energy efficiency solutions accessible to households that otherwise may not afford name-brand alternatives. This provides tangible financial benefits, such as significant reductions in energy bills, which improve quality of life.

Weak Broader Context–Principle Pair: Environmental – Nonmaleficence

One area where our project is currently lacking is Environmental – Nonmaleficence. While our design promotes reduced energy use, we have not thoroughly considered the environmental impact of the hardware itself, such as the long-term sustainability of ESP32 devices, PCB manufacturing waste, or end-of-life disposal. To improve, our team should explore more eco-conscious sourcing, energy-efficient firmware strategies, and user education on responsible device disposal. These steps would help mitigate our environmental footprint.

7.3 VIRTUES

1. Integrity

Definition: Acting honestly and transparently in all aspects of the project.

Team Action: We prioritize open communication about project progress, challenges, and limitations. Team members document their work accurately, and we ensure credit is given where due.

2. Collaboration

Definition: Working effectively with others by sharing knowledge, listening, and supporting team goals.

Team Action: We use weekly meetings and shared documentation tools to make sure everyone is aligned. Tasks are divided based on individual strengths, and we help each other when issues arise.

3. Responsibility

Definition: Being dependable and accountable for your work and its impact.

Team Action: Each member commits to deadlines and tasks. When someone is behind or uncertain, the team reassigns or redistributes work as needed to keep the project on track and maintain quality standards.

INDIVIDUAL REFLECTIONS

1. Arie Kraayenbrink

Virtue Demonstrated: Integrity

- **Why it's important:** Integrity is the foundation on which many other things are built. It allows the team to function efficiently without guessing about the details provided by others.
- **How I've demonstrated it:** One area where this has been demonstrated is in the way parts are ordered. By using a spreadsheet to track ordered parts: when the parts were ordered and by whom, when the parts were picked up and by whom, and what the cost of those parts was.

Virtue Not Yet Demonstrated: Consistent collaboration

- **Why it's important:** Teams are able to work together much more efficiently when there is consistency in the way things are done. This means that when done well, there are better outcomes and higher quality work.
- **How I might demonstrate it:** Work more with others outside of just the team meetings, and focus on doing tasks together as a team where possible.

2. Alexander Ryan

Virtue Demonstrated: Diligence

- **Why it's important:** Diligence is essential to me because it ensures that work is completed thoroughly and on time, especially when dealing with complex, interconnected systems like those in a smart home environment. In engineering, missing a detail can cause an entire system to malfunction, which is unacceptable in real-world applications.
- **How I've demonstrated it:** I consistently documented our system architecture and thoroughly tested algorithms to ensure reliable automation behavior. I also took initiative to fix integration issues within our implementation and spent extra hours debugging automations and our custom React app so that our system could run smoothly and safely.

Virtue Not Yet Demonstrated: Patience

- **Why it's important:** Patience is important because real-world engineering often involves trial and error, setbacks, and working with others who may have different approaches or paces. Rushing or getting frustrated can lead to poor decisions and tension within a team.
- **How I might demonstrate it:** I can demonstrate patience by being more supportive when teammates face challenges with unfamiliar tools or debugging difficult errors. Instead of trying to take over or quickly “solve it,” I will collaborate calmly, help troubleshoot step-by-step, and allow time for learning.

3. Erik Schmidt

Virtue Demonstrated: Collaboration

- **Why it's important:** Collaboration in a team setting means working together by sharing ideas, supporting each other, and combining strengths to achieve a common goal more effectively.
- **How I've demonstrated it:** Working with other members of the hardware team, helping them solve their issues, and they have helped me with mine.

Virtue Not Yet Demonstrated: Perseverance

- **Why it's important:** Perseverance is important because it allows you to continue and stay committed, even if things become challenging.
- **How I might demonstrate it:** I will continue working through technical issues without giving up or saying that the features aren't needed. When I face a setback, I will keep troubleshooting until I find a solution.

4. Gamage Binula Karunaratne

Virtue Demonstrated: Collaboration

- **Why it's important:** Collaboration is important because most of the engineering projects in the industry are carried out in groups. It is important to our project to get the most out of a group and create something better than the sum of its parts.
- **How I've demonstrated it:** By constantly joining group meetings and sharing my opinion on matters. Also, by being responsive to any queries fellow group members have on the project.

Virtue Not Yet Demonstrated: Time Management

- **Why it's important:** Time management is important to hit deadlines in projects. Also, time management is important to balance multiple projects, prioritize tasks, and efficiently utilize time to enhance productivity and well-being.
- **How I might demonstrate it:** I will prioritize the work of this project and manage my time better to give my best to this project. I will be better at balancing my other coursework and this project.

5. Mitchell Jensen

Virtue Demonstrated: Cooperativeness

- **Why it's important:** Cooperativeness is extremely important to me as working together is the best way to complete a project.
- **How I've demonstrated it:** By attending meetings that are not necessarily associated with my expertise. I have also worked to bring other members of the team into my area of expertise to increase their experience in the subjects that I am familiar with.

Virtue Not Yet Demonstrated: Creativity

- **Why it's important:** Creativity is important to be able to solve problems in new and better ways than previously done.
- **How I might demonstrate it:** I will work on expanding my thought processes to search out more creative ways of meeting the project requirements

8 Conclusions

8.1 CONCLUSION

The team had the design for a low-cost smart home control unit that enabled users to monitor and control lighting and temperature completed. At this point, the sensor component, furnace control unit, and the hub had been built and were being tested. The primary goal of the smart-home control unit was to reduce energy consumption and increase affordability for a variety of different user types. The system utilized the open-source platform Home Assistant with cost-effective microcontrollers to deliver smart home control automation. The team still needed to finish the light dimmer unit and connect everything together using the software. Finally, a mobile application had to be created to improve the user experience.

The best approach for achieving these goals was to use cost-effective and efficient hardware parts when designing the circuits for each component of the home control unit. The team also utilized the open-source platform Home Assistant as the software component. This platform was free to use and relatively easy to customize to the project's needs. The constraints at that time were component availability, documentation gaps in emerging hardware, and time limitations.

In future iterations, the team plans to enhance the solution by expanding to include machine-learning-based prediction models for even better optimization, improving UI accessibility, and offering voice assistant integration. Additionally, battery or solar backup support will be added for sustainability and resiliency, and broader field testing will be conducted to ensure robustness in various home layouts and climates.

8.2 VALUE PROVIDED

It directly addresses core user needs around the reduction of utility bills, increases comfort, and provides ease of control for non-technical users by marrying low-cost hardware with an intuitive web and mobile interface. Homeowners, property managers, and environmentalists alike can benefit from automated temperature, lighting, and multi-zone HVAC controls that reduce conditioning wasted in unoccupied rooms while still allowing for comfort or special situations manual overrides. Testing conducted on the early development stage of zone controllers showed that the system is capable of focusing conditioning on occupied rooms while relaxing setpoints elsewhere—a function fully aligned with the original goal of reducing energy consumption without sacrificing comfort where it matters most.

This would nicely fit into the larger context of smart homes and energy efficiency by offering an open, standards-based alternative to proprietary, expensive platforms such as Nest or Ecobee. Using Home Assistant, ESP32-based Zigbee nodes, and standards like IEEE 802.15.4, UL 873, and ISO/IEC 27001, this system proves that a low-cost DIY solution is still interoperable, safe, and secure. Unit, integration, system, and acceptance level testing gives evidence that such a prototype will meet requirements of

responsiveness, ability to operate offline, and, above all, usability. This means that architecture robust enough to deploy in real homes for continued evaluation and refinement.

Beyond individual users, the design has potential societal value for reducing residential energy waste and associated emissions, especially when deployed in older, less efficient housing or multi-level homes that have large uneven temperature variations. The architecture creates a reusable platform of hub, sensor network, and control logic that future teams can extend with richer analytics or new devices, amplifying long-term impact.

8.3 WHAT'S NEXT

Immediate next steps include integrating and validating the production PCB versions of the multi-sensor, furnace controller, zone controller, and light dimmer such that the system may transition from breadboard prototypes to installation-ready hardware. Completion and hardening of the custom dimmer, which will eliminate flicker and noise across diverse fixtures, and completion of the mobile application will also be critical in offering a polished, end-to-end user experience.

From a research and product-evolution perspective, an important follow-up project would entail implementing solid data logging and analytics to quantify actual energy savings compared to baseline usage over multiple seasons and house types. It would offer grounding for the development of more advanced optimization strategies, including schedule-aware or machine learning-based control that predicts occupancy and weather effects and utility rate patterns. Additional work could explore deeper sustainability features, including battery or solar integration for resilience, life-cycle analysis of hardware, and guidance for responsible device disposal to improve the system's environmental nonmaleficence. Finally, future teams may pursue user-centered enhancements such as accessibility improvements in the UI, support for multiple languages, or embedded onboarding wizards that guide nontechnical installers through sensor placement, zoning, and security setup. Parallel projects may expand compatibility to more types of HVAC buses, integrate with utility demand-response programs, or package the system as a contractor-friendly kit for new construction—all logical extensions given the current architecture and partially completed features.

9 References

- [1] American Council for an Energy-Efficient Economy, "Study: One in four low-income households spend over 15% of income on energy bills," Sep. 15, 2024. <https://www.aceee.org/press-release/2024/09/study-one-four-low-income-households-spend-over-15-income-energy-bills> (accessed Feb. 28, 2025).
- [2] U.S. Department of Energy, "Heating and Cooling," Energy.gov, 2022. [Online]. Available: <https://www.energy.gov/energysaver/heating-and-cooling>
- [3] U.S. Energy Information Administration, "Residential Energy Consumption Survey (RECS)," 2020. [Online]. Available: <https://www.eia.gov/consumption/residential/>
- [4] D. Nadel, "Energy Savings from Smart Thermostats," American Council for an Energy-Efficient Economy (ACEEE), Technical Report, 2020.
- [5] U.S. Energy Information Administration, "How much energy is consumed in U.S. residential and commercial buildings?," 2021. [Online]. Available: <https://www.eia.gov/tools/faqs/faq.php?id=86&t=1>
- [6] Parks Associates, "Smart Home and Security Tracker," Market Research Report, 2023.
- [7] Aprasi, "Smart home automation: Boost energy efficiency in 2024," 2024. <https://www.aprasi.com/blogs/trending-topics/smart-home-automation-boost-energy-efficiency-in-2024> (accessed Feb. 28, 2025).
- [8] Nest Labs, "Nest Thermostat Specifications," 2024. [Online]. Available: <https://nest.com>
- [9] Ecobee, "How Ecobee Smart Thermostats Save You Money," 2023. [Online]. Available: <https://www.ecobee.com>
- [10] Samsung SmartThings, "SmartThings Hub Overview," 2024. [Online]. Available: <https://www.smarthings.com>

10 Appendices

10.1 OPERATION MANUAL

SMART HOME ENERGY OPTIMIZER USER MANUAL

For Hub, Sensors, Furnace Controller, Light Dimmer Controller, and HVAC Zone Controller

1. System Overview

The Smart Home Energy Optimizer automates **temperature, lighting, and energy management** through a central hub.

The system includes:

- **Central Hub** – Communicates with all devices (Zigbee / Wi-Fi).
 - **Multi-Sensors** – Monitor room conditions.
 - **Furnace / Thermostat Controller** – Controls heating & HVAC calls.
 - **Light Dimmer Controller** – Adjusts room brightness.
 - **HVAC Zone Controller** – Controls motorized dampers for multi-zone temperature regulation.
 - **Web Dashboard** – View energy data and control all devices from any device.
-

2. Getting Started

2.1 Unboxing Checklist

The Smart Home Energy Optimizer should contain:

- Central Hub
 - Power supply
 - Sensors (temperature, light, presence)
 - Furnace/Thermostat Controller
 - Light Dimmer Module
 - HVAC Zone Controller
 - QR code for dashboard setup
 - Mounting accessories (tape, screws, brackets)
-

3. Installation

3.1 Install the Central Hub

1. Place the hub in a **central location** in your home.
 2. Connect the hub to power.
 3. Connect the hub to your **home network** (Wi-Fi or Ethernet).
 4. Wait for the solid connection LED (indicates it's ready).
-

3.2 Install the Multi - Sensors

1. Mount **5 feet above the floor**, away from vents or sunlight.
2. Peel adhesive backing or use screws.
3. Plug into the wall.

Once powered, sensors automatically pair with the hub.

3.3 Install the Furnace / Thermostat Controller

1. Turn off your furnace power switch.
2. Locate the 24V thermostat terminals.
3. Connect the controller wires to:
4. Restore power and wait for the pairing confirmation on the hub.

The hub will automatically detect the controller as "**Furnace Controller**".

3.4 Install the Light Dimmer Controller

1. Turn off the power at the breaker.
 2. Remove the switch plate.
 3. Install the dimmer in series with the load following labeled terminals.
 4. Restore power and check that the dimmer appears in the hub.
-

3.5 Install the HVAC Zone Controller

(For multi-zone homes)

1. Connect damper actuators or hydronic valves to the zone controller's **normally-open relays**.
 2. Assign each relay to a zone (Zone 1–8).
 3. Pair the zone controller with the hub.
 4. Name each zone in the dashboard (e.g., **Upstairs Bedrooms, Basement, Main Floor**).
-

4. Using the Web Dashboard

You can access the dashboard via the QR code

4.1 Dashboard Tabs

Home

- Shows current temperatures, light levels, motion, and system status.

Controls

- Adjust lighting levels (0–100%).
- Change temperature setpoints.
- Override HVAC modes (Heat / Cool / Auto).
- Manage HVAC zones individually.

Insights (Energy + Cost Savings)

- View real-time energy use.
- See cost savings from automation.
- Identify high-usage rooms or systems.

Automations

- Schedule lights
- Set temperature rules
- Add presence-based actions (e.g., “Turn lights off when no one is home”).

Settings

- Rename rooms

- Pair new devices
 - Update firmware
 - Manage notifications
-

5. Daily Use Guide

Adjusting Temperature

1. Open **Controls** → **Temperature**.
2. Select a zone.
3. Set the desired temperature.
4. The zone controller will open/close dampers automatically.

Controlling Lights

1. Go to **Controls** → **Lighting**.
2. Use slider to dim lights.
3. Use On/Off toggle for quick control.

Viewing Energy Data

1. Go to **Insights**.
 2. View graphs of daily/weekly energy usage.
 3. Compare estimated cost savings with automation enabled.
-

6. Troubleshooting

Sensor Not Updating

- Ensure the power outlet works.
- Move sensor closer to hub
- Re-pair using dashboard → Settings → Devices

No Response from a Controller

- Ensure it has power
- Check wiring (for furnace controller / dimmer)
- Reboot hub

Dashboard Slow or Unresponsive

- Refresh browser
- Restart the hub
- Ensure your home network is stable

Zone Not Heating/Cooling

- Confirm damper actuator is wired to the correct relay
- Check zone temperature settings
- Look for obstructions in ductwork

10.2 ALTERNATIVE/ INITIAL VERSION OF DESIGN

Alternative / Initial Versions of Design

This appendix outlines major design versions considered in the entire project, including very early concepts explored prior to finally understanding the needs of the client, alternatives considered in design exploration, and approaches that did not meet specifications and were revised.

1. Hub Platform Alternatives

Initial Version: Raspberry Pi / Home Assistant Green as Primary Hub

The team initially intended to deploy Home Assistant on a Raspberry Pi or turnkey Home Assistant Green appliance in order to keep the cost and size as low as possible. After being told that development tools, additional services, and future expansion-e.g., more Zigbee devices along with logging-would see major benefits from additional compute and storage capacity, and after the client emphasized stability over minimum hardware cost, this was revised. The design is updated using an Intel NUC as the main hub for development, with a pathway to downsize to a Raspberry Pi 5 once resource needs are better quantified.

2. Automation Platform Alternatives

Initial Version: Custom-Built Automation Stack

An early idea was to build a bespoke automation backend and device integration layer instead of using an existing platform. This track had been abandoned because, aside from the already mentioned time scale of two semesters being too short, an implementation of protocols, integrations, and a full rules engine from scratch also replicate very mature open-source work. The team instead used Home Assistant after evaluating it with OpenHAB, choosing Home Assistant based on community support, breadth of integration, and client familiarity.

Alternative: OpenHAB-Based System

OpenHAB was seriously considered as the core automation engine. It was not chosen after evaluation because, compared to Home Assistant, it had better documentation, smoother onboarding, and easier UI customization for non-technical users, which aligns more directly with project requirements and client expectations.

3. Sensor and Presence Detection Variants

Battery-Powered Multi-Sensor with mmWave Presence

One key concept in the early stages was a fully battery-powered multisensor node using an mmWave radar presence sensor to avoid the false negatives common with PIR sensors. The mmWave module in practice required a long warm-up period, and continuous operation caused significant increases in power draw that made long-lived battery operation impractical. This proved not to meet the requirements of low power and thus was revised to prioritize wall-powered sensors with optional future optimization for battery-backed deployment.

PIR-Only Presence Detection

The team initially considered using only PIR sensors to minimize cost and power consumption. User scenarios with occupants sitting still-desks, couches-revealed that PIR alone would not reliably detect presence, risking comfort and trust in automations. The final design blends mmWave-based presence detection-for reliability-with the assumption of wired power in most installations, deferring a full battery-optimized revision to future work.

4. HVAC Control and Zoning Alternatives

Single-Zone Smart Thermostat Only

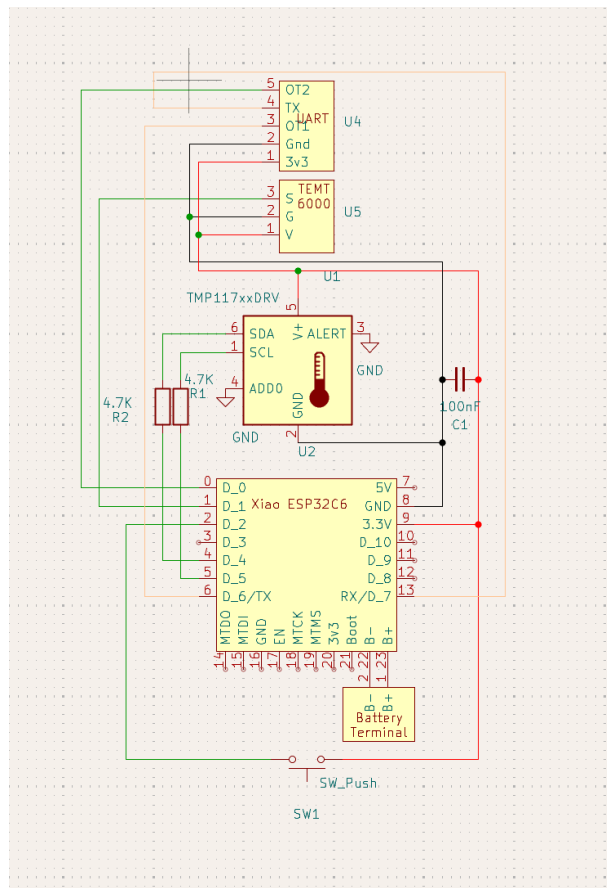
The first concept for HVAC was no more than a replacement of the standard thermostat with a single controller driven by rules in Home Assistant. As the team learned about user stories, such as multi-level homes with hot upstairs and cold basements, it became clear that a single-zone approach was far from solving the pain of uneven comfort and energy waste in unused rooms. This led to the development of an 8-zone HVAC controller using ESP32-based Zigbee nodes and 24 VAC relays controlling dampers or hydronic valves. Commercial Zone Controller Integration: Honeywell TrueZONE The use of a commercial zoning panel was considered to eliminate the need for custom relay logic. Abandoned due to high cost, proprietary constraints, and inability to easily experiment with custom control algorithms and integrations within Home Assistant, this route was taken. A custom zone controller was adopted to maintain affordability, extensibility, and full observability of HVAC behavior. Smart Vent-Based Zoning Smart vents were considered by the team to be a simpler retrofit option for achieving room-level control. This concept was discarded after concerns about airflow restriction, noise, and cost per vent, and recognizing that it may introduce safety issues for some HVAC systems. The project focused on duct

dampers and hydronic valves under central control, an approach better aligned with HVAC safety standards (UL 873, IEEE C37.90) and long-term maintainability.

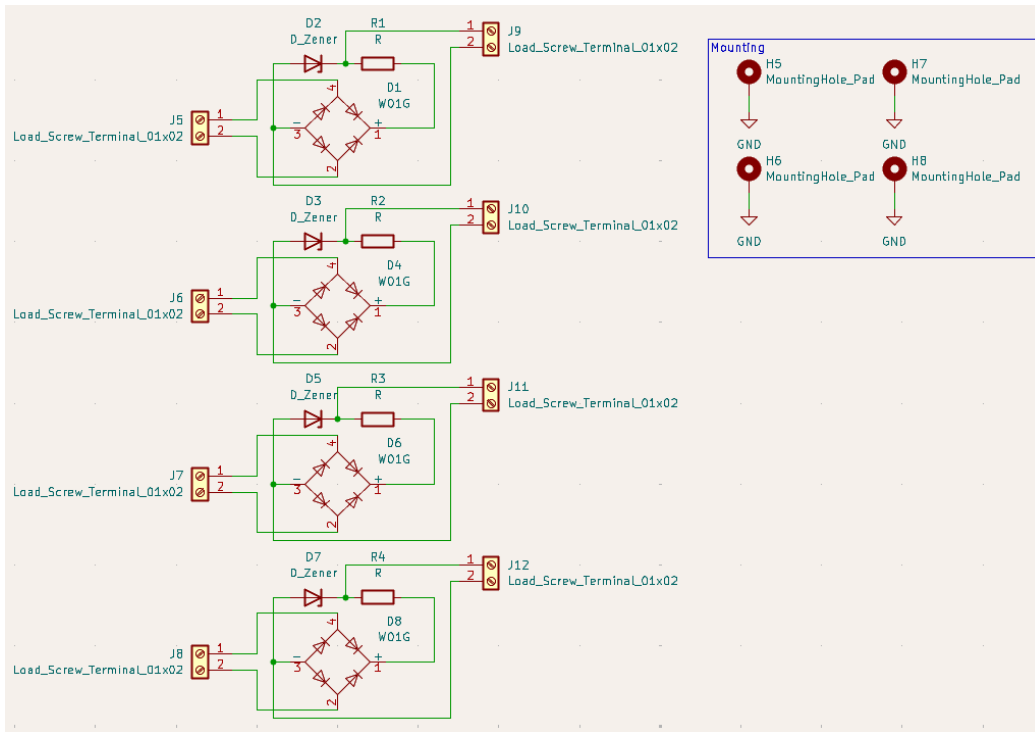
5. Lighting Control Approaches Only Commercial Smart Switches and Bulbs A variant of the design relied entirely on off-the-shelf smart switches and bulbs integrated into Home Assistant. While this greatly simplified hardware development, it limited fine-grained dimming control, increased per-room cost, and explicitly tethered the system to vendor APIs that might change at any time. The team revised the plan to include a custom ESP32-based dimmer in search of more precise control and lower cost, although the current dimmer still requires additional work to handle flicker and corner cases.

6. Networking and Protocol Choices Wi-Fi–Only Sensor Network Early drafts assumed Wi-Fi for all sensors and controllers, because of simplicity. It became clear during small-scale testing and requirement analysis that a Wi-Fi-only design would put stress on typical home routers, increase power consumption, and make pairing a non-trivial task for non-technical users. The team moved to a Zigbee-centered mesh based on IEEE 802.15.4, using Wi-Fi mostly for the hub and user interfaces for improved reliability and power efficiency.

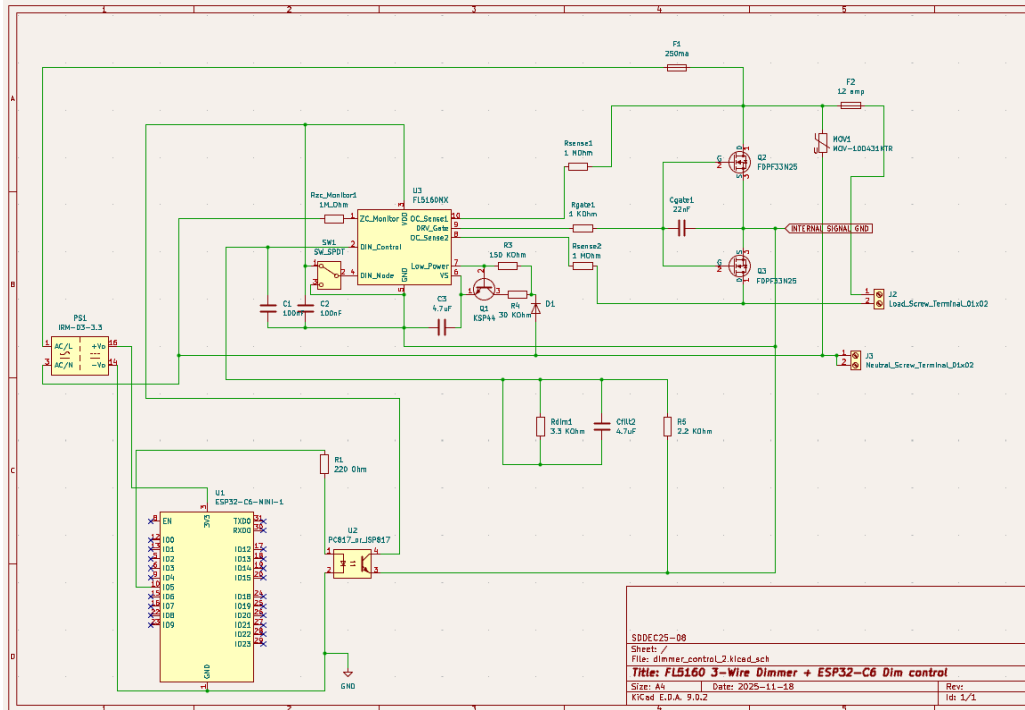
10.3 CIRCUIT DIAGRAM



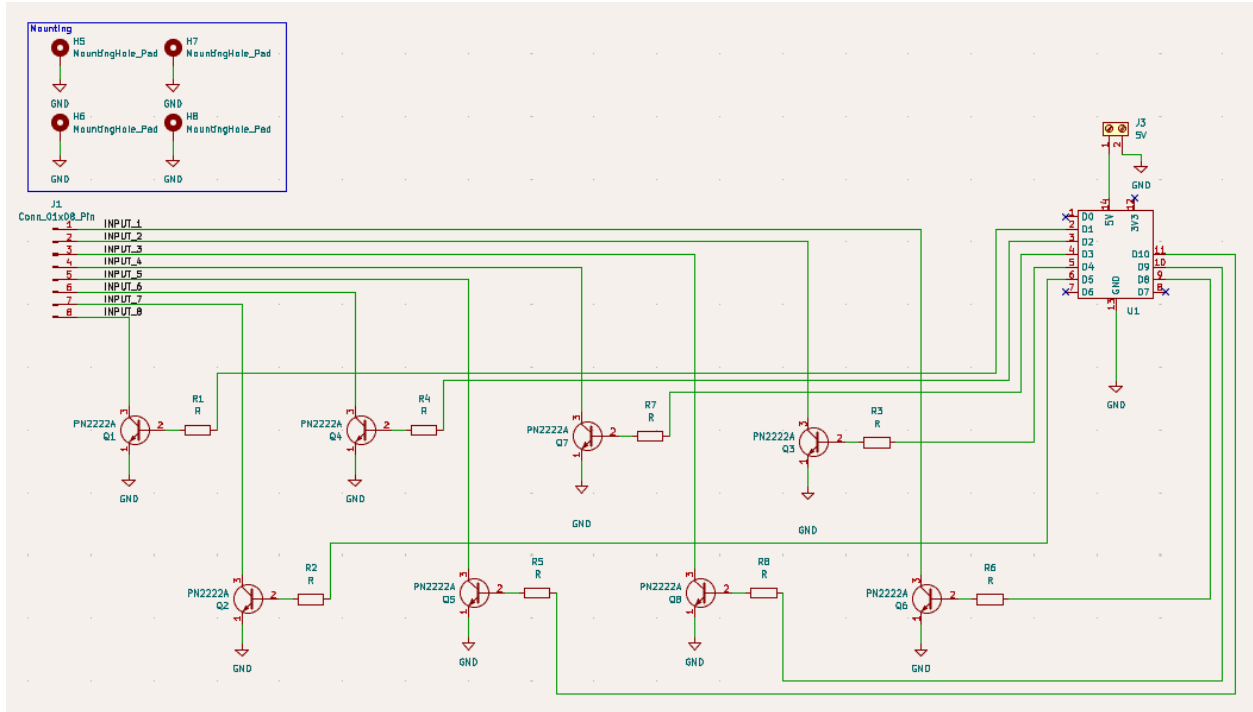
Multi-sensor schematic



Furnace Controller schematic



Light dimmer schematic



Zone controller schematic

10.4 CODE

<https://github.com/AlexanderTRyan/ha-custom-panel>

10.5 TEAM CONTRACT

TEAM MEMBERS

- 1) Arie Kraayenbrink
- 2) Alexander Ryan
- 3) Gamage Binula Karunaratne
- 4) Erik Schmidt
- 5) Mitchell Jensen

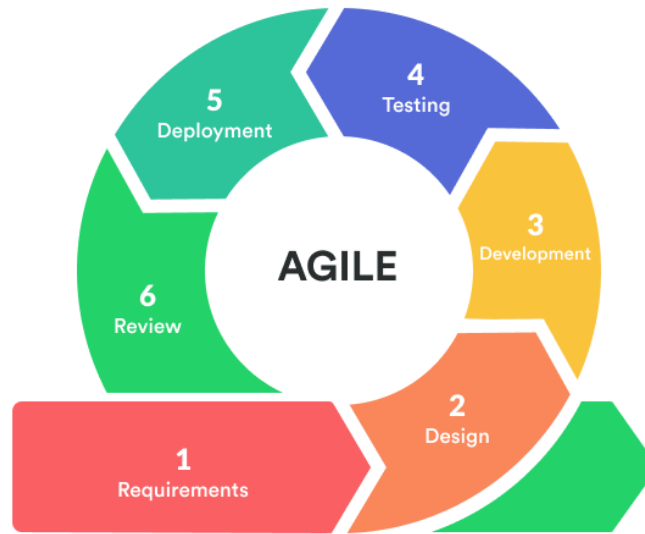
REQUIRED SKILL SETS FOR YOUR PROJECT

Skill Set	Requirement / Use Case
Embedded Systems	Sensor integration and microcontroller programming
Web Development	Front-end design for user interface
Backend Development	Communication with Home Assistant, database setup, and management
Wireless Networking	Implementing and debugging Zigbee/MQTT communication between sensors and the hub
Circuit Design and Prototyping	Hardware interfacing and PCB layout
System Integration and Automation Logic	Creating the automation logic and implementing it into Home Assistant
Testing and Validation	Functional, integration, and security testing
UI/UX Design	Making the app and web interface user-friendly

SKILL SETS COVERED BY THE TEAM

Skill Set	Team Member
Embedded Systems	Arie Kraayenbrink, Erik Schmidt, Seth Proctor, Alexander Ryan, Gamage Binula Karunaratne, Mitchell Jensen
Web Development	Arie Kraayenbrink, Seth Proctor, Alexander Ryan
Backend Development	Arie Kraayenbrink, Jacob Carnesi, Alexander Ryan
Wireless Networking	Arie Kraayenbrink, Erik Schmidt, Seth Proctor
Circuit Design and Prototyping	Erik Schmidt, Gamage Binula Karunaratne, Mitchell Jensen
System Integration and Automation Logic	Arie Kraayenbrink, Erik Schmidt, Jacob Carnesi
Testing and Validation	Arie Kraayenbrink, Erik Schmidt, Seth Proctor, Jacob, Alexander Ryan, Gamage Binula Karunaratne, Mitchell Jensen
UI/UX Design	Seth Proctor, Alexander Ryan

PROJECT MANAGEMENT STYLE ADOPTED BY THE TEAM



We chose agile because of its flexibility and iterative nature. Our team held weekly sprints and used Jira to manage and assign tasks. Regular meetings were held to ensure alignment across hardware and software tracks, and our approach allowed for quick adaptation when technical issues arose.

INITIAL PROJECT MANAGEMENT ROLES

- 1) Seth Proctor - **Hardware/Software Developer**
- 2) Arie Kraayenbrink - **Hardware/Software Developer**
- 3) Jacob Carnesi - **Project Manager, Security/Internet Developer**
- 4) Alexander Ryan - **UX/Software Developer**
- 5) Gamage Binula Karunaratne - **Hardware Developer**
- 6) Erik Schmidt - **Embedded Systems Developer**
- 7) Mitchell Jensen - **Hardware Developer**

TEAM CONTRACT

Team Name: SDDEC25-08

Team Members:

- 1) Arie Kraayenbrink
- 2) Alexander Ryan
- 3) Gamage Binula Karunaratne
- 4) Erik Schmidt
- 5) Mitchell Jensen

Team Procedures

1. Day, time, and location (face-to-face or virtual) for regular team meetings:
 - a. Monday, 1-1:30 pm, virtual. Friday 1-1:30 pm
2. Preferred method of communication updates, reminders, issues, and scheduling (e.g., e-mail, phone, app, face-to-face):
 - a. Text Microsoft Teams for the meeting.
3. Decision-making policy (e.g., consensus, majority vote):
 - a. Majority vote
4. Procedures for record keeping (i.e., who will keep meeting minutes, how will minutes be shared/archived):
 - a. Record Teams Meetings

Participation Expectations

1. Expected individual attendance, punctuality, and participation at all team meetings:
 - a. All members will attend meetings unless prior notice is given. Any absences must be communicated in advance.
 - b. If there are unforeseen circumstances that prevent a team member from attending a meeting, they shall notify the other team members beforehand.
2. Expected level of responsibility for fulfilling team assignments, timelines, and deadlines:
 - a. Each team member is expected to complete their assigned tasks on time and to the best of their ability.
 - b. If a team member is unable to complete their assigned task, they must notify the team and discuss a plan to move forward.
3. Expected level of communication with other team members:
 - a. Regular updates will be shared via Microsoft Teams. Any issues or delays must be communicated promptly.
 - b. Each team member is expected to respond to messages in a timely manner.
4. Expected level of commitment to team decisions and tasks:
 - a. All members must respect and adhere to team decisions. If disagreements arise, they will be discussed in meetings.

Leadership

1. Leadership roles for each team member (e.g., team organization, client interaction, individual component design, testing, etc.):
 - a. Project manager - Jacob Carnesi
 - b. Designer - Seth Proctor
 - c. Hardware - Gamage Binula Karunaratne
 - d. Software - Alexander Ryan
 - e. Client relations - Arie Kraayenbrink
2. Strategies for supporting and guiding the work of all team members:
 - a. Regular check-ins during meetings to assess progress.
 - b. Peer reviews of work before submission.
 - i. GitHub code reviews
 - c. Open communication and collaboration.
3. Strategies for recognizing the contributions of all team members:
 - a. Acknowledging contributions during meetings.
 - b. Giving constructive feedback and encouragement.
 - c. Celebrating milestones and completed tasks.

Collaboration and Inclusion

1. Describe the skills, expertise, and unique perspectives each team member brings to the team.
 - a. **Arie Kraayenbrink**
 - i. **Programming Languages:** C, C++, C#, Python, Java, JavaScript, Swift, HTML, CSS, PHP, VBA, Assembly, Go
 - ii. **Platforms & Frameworks:** Windows, Linux, Android, Azure, AWS, Google, .NET, React, Mongo, TSQL, Postgres, Firebase
 - iii. **Experience:** Electrician with experience installing some home automation systems. Personal use of home automation in my own house. Some DevOps. Some familiarity with Z-Wave, Zigbee, and other such protocols for home automation.
 - b. **Alexander Ryan**
 - i. **Programming Languages:** HTML, CSS, C, C++, C#, JS, JAVA, SQL, TS, TSX, JSX, Python, VB
 - ii. **Platforms:** Windows, Linux, Android, .NET, React, Mongo, Firebase
 - iii. **Dev Skills:** App + Web Development, Unit testing, Service Testing, E2E Testing
 - iv. **WorkFlows:** Git, CMD, Jira, Agile
 - c. **Gamage Binula Karunaratne**
 - i. **Programming Languages:** Verilog HDL, C
 - ii. **Circuit Simulation Software:** LTspice, Cadence Virtuoso, QuestaSim, ModelSim
 - iii. **Experiences:** VLSI Design, Circuit Design

- iv. **Other Skills:** Web design, Graphic design (Photoshop), 3D Modelling
- d. **Erik Schmidt**
 - i. **Programming Languages:** C, C++, C#, JAVA, Kotlin, SQL, VHDL, Python, Assembly
 - ii. **Platforms & Frameworks:** Windows, Linux, Android
 - iii. **Skills:** App Development, Unit Testing, Service Testing, Circuit Design, CAD, Machining, Fabrication
 - iv. **Tools & Workflow:** Teams, SharePoint, Git
 - v. **Experience:** Worked with open source smart home systems before, designed and printed custom circuit boards for projects similar in scope to smart home sensors.
- 2. Strategies for encouraging and supporting contributions and ideas from all team members:
 - a. Active listening and open discussions during meetings.
 - b. Encouraging feedback and brainstorming sessions.
 - c. Providing a supportive and inclusive environment.
- 3. Procedures for identifying and resolving collaboration or inclusion issues (e.g., how will a team member inform the team that the team environment is obstructing their opportunity or ability to contribute?)
 - a. Addressing concerns in meetings or private discussions.
 - b. Encouraging members to speak up if they feel unheard.
 - c. Seeking mediation if necessary (getting the instructor involved).

Goal-Setting, Planning, and Execution

- 1. Team goals for this semester:
 - a. Plan and set reasonable project milestones.
 - b. Complete all assigned project milestones on time.
 - c. Ensure effective communication and collaboration.
 - d. Be well on the way to developing a high-quality product next semester.
- 2. Strategies for planning and assigning individual and team work:
 - a. Assign tasks based on strengths and expertise.
 - b. Use a shared task board for tracking progress.
 - c. Regularly review progress and adjust plans as needed.
- 3. Strategies for keeping on task:
 - a. Setting clear deadlines.
 - b. Holding each other accountable.
 - c. Checking progress during meetings.

Consequences for Not Adhering to the Team Contract

- 1. How will you handle infractions of any of the obligations of this team contract?
 - a. First offense: Verbal warning and discussion in a meeting.
 - b. Second offense: Team discussion and redistribution of responsibilities.
 - c. Third offense: Discussion with the instructor.

2. What will your team do if the infractions continue?
 - a. If a team member repeatedly fails to meet expectations, they may be reported to the instructor for further action.

- a) *I participated in formulating the standards, roles, and procedures as stated in this contract.*
- b) *I understand that I am obligated to abide by these terms and conditions.*
- c) *I understand that if I do not abide by these terms and conditions, I will suffer the consequences as stated in this contract.*

- 1) Arie Kraayenbrink _____ DATE: 6 Feb 2025
- 2) Alexander Ryan _____ DATE: 6 Feb 2025
- 3) Gamage Binula Karunaratne _____ DATE: 6 Feb 2025
- 4) Erik Schmidt _____ DATE: 13 Feb 2025
- 5) Mitchell Jensen _____ DATE: 22 Aug 2025