



Faculty of Computer Science and Information Technology

INTELLIVENT: SMART WINDOW AUTOMATION FOR WEATHER AND TIME OPTIMIZATION

Nouf Hasya Ilma Binti Badaruddin (78334)

Bachelor's in Computer Science (Network Computing) with Honours

2024

INTELLIVENT: SMART WINDOW AUTOMATION FOR WEATHER AND TIME OPTIMIZATION

Intellivent: Smart Window Automation For Weather And Time Optimization

NOUF HASYA ILMA BINTI BADARUDDIN

This project is submitted in partial fulfilment of the requirements for the degree of
Bachelor of Network Computing with Honours

Faculty of Computer Science and Information Technology
Universiti Malaysia Sarawak

2025

INTELLIVENT: AUTOMASI TINGKAP PINTAR UNTUK PENGOPTIMUMAN CUACA DAN MASA

NOUF HASYA ILMA BINTI BADARUDDIN

Projek ini merupakan salah satu keperluan untuk
Ijazah Sarjana Muda Kejuruteraan Perisian dengan Kepujian

Fakulti Sains Komputer dan Teknologi Maklumat
Universiti Malaysia Sarawak
2025

UNIVERSITI MALAYSIA SARAWAK

THESIS STATUS ENDORSEMENT FORM

TITLE **IntelliVent: Smart Window Automation for Weather
and Time Optimization**

ACADEMIC SESSION: 2024/2025

NOUF HASYA ILMA BINTI BADARUDDIN
(CAPITAL LETTERS)

hereby agree that this Thesis* shall be kept at the Centre for Academic Information Services, Universiti Malaysia Sarawak, subject to the following terms and conditions:

1. The Thesis is solely owned by Universiti Malaysia Sarawak
2. The Centre for Academic Information Services is given full rights to produce copies for educational purposes only
3. The Centre for Academic Information Services is given full rights to do digitization in order to develop local content database
4. The Centre for Academic Information Services is given full rights to produce copies of this Thesis as part of its exchange item program between Higher Learning Institutions [or for the purpose of interlibrary loan between HLI]
5. ** Please tick (✓)

☒

CONFIDENTIAL (Contains classified information bounded by the OFFICIAL SECRETS ACT 1972)

☒

RESTRICTED (Contains restricted information as dictated by the body or organization where the research was conducted)

☒

UNRESTRICTED

Validated by



(AUTHOR'S SIGNATURE)

(SUPERVISOR'S SIGNATURE)

Permanent Address

RB2-1, Persiaran Bayan 4,
Taman Tunas Muda, Sg. Ara
Bayan Lepas, 11900,
Date: 23 June 2025

Date: _____

Note * Thesis refers to PhD, Master, and Bachelor Degree

** For Confidential or Restricted materials, please attach relevant documents from relevant organizations / authorities

DECLARATION

I hereby declare that the Intellivent UNIMAS project is my original work, developed exclusively for academic purposes. I affirm that no part of this work has been copied from another student's work or any other source without proper citation or acknowledgment. Furthermore, no part of this project has been written or completed on my behalf by any other individual.



Nouf Hasya Ilma Binti Badaruddin

23/06/2025

Date

ACKNOWLEDGEMENT

I would like to take this opportunity to express my deepest gratitude to everyone who helped in my completion of Final Year Project I and II. First and foremost, I am really grateful to my supervisor, Professor Madya Dr. Halikul bin Lenando, for his invaluable guidance, continuous support, and constructive feedback throughout the development of this project. His expertise and encouragement have been instrumental in shaping the project and helping me surmount the challenges that came along the way. I also would like to thank our coordinator for the Final Year Project, Professor Dr. Wang Yin Chai, for his guidelines towards the progress of the project. My appreciation goes out to my comrades and colleagues for offering their insights, suggestions, and moral support during this project. Last but not least, I am always indebted to my family and friends, Maisarah, Olie, Izzan and Irdina, who are very supportive and understanding, and when things got tough, urged me to hold on. Without the efforts of all these people put together, this project would not have seen the light of day, and I dedicate this achievement to them.

ABSTRACT

Growing demands for energy-efficient solutions in smart home automation have increase the demand for the development of innovative systems that minimize dependencies on HVAC to provide comfort for the occupants. Most of the existing systems lack the integration of real-time environmental data, which results in several inefficiencies in energy consumption. This project fills this gap by developing IntelliVent, a smart window automation system using environmental sensors and a dynamic control algorithm that will come up with an optimal natural ventilation strategy. It will integrate real-time data from on-site sensors, a robust automation logic, and a user-friendly mobile interface to monitor and further enhance system performance. The expected outcomes are significant energy savings, improved indoor air quality and increased adaptability to environmental changes thus providing a scalable solution for sustainable smart home technologies.

ABSTRAK

Permintaan yang semakin meningkat untuk penyelesaian cekap tenaga dalam automasi rumah pintar telah mendorong pembangunan sistem inovatif yang dapat mengurangkan kebergantungan terhadap sistem HVAC (Pemanasan, Pengudaraan, dan Penyaman Udara) bagi menyediakan keselesaan kepada penghuni. Kebanyakan sistem sedia ada tidak mempunyai integrasi data persekitaran masa nyata, yang mengakibatkan ketidakcekapan dalam penggunaan tenaga. Projek ini mengisi jurang tersebut dengan membangunkan IntelliVent, sebuah sistem automasi tingkap pintar yang menggunakan sensor persekitaran dan algoritma kawalan dinamik untuk menghasilkan strategi pengudaraan semula jadi yang optimum. Sistem ini akan mengintegrasikan data masa nyata daripada sensor di tapak, logik automasi yang mantap, serta antara muka mudah alih yang mesra pengguna untuk memantau dan meningkatkan lagi prestasi sistem. Hasil yang dijangkakan ialah penjimatan tenaga yang signifikan, kualiti udara dalaman yang lebih baik, dan kebolehsuaian yang dipertingkatkan terhadap perubahan persekitaran, sekali gus menyediakan penyelesaian berskala untuk teknologi rumah pintar yang mampan.

TABLE OF CONTENTS

DECLARATION.....	2
ACKNOWLEDGEMENT.....	3
ABSTRACT.....	4
ABSTRAK.....	5
TABLE OF CONTENTS.....	6
LIST OF FIGURES	10
LIST OF TABLES.....	12
CHAPTER 1: INTRODUCTION.....	13
1.1 Background of Study	13
1.2 Problem Statement	14
1.3 Objectives of Study.....	16
1.4 Scope and Limitations.....	16
1.5 Definition of Terms	18
1.6 Methodology	19
1.7 Significance of the Project	20
1.8 Project Schedule.....	21
1.9 Expected Outcome	23
1.10 Summary	24
CHAPTER 2: LITERATURE REVIEW.....	26
2.1 Background of Study	26
2.2 Review of Existing Smart Home Systems	26

2.3 Window Automation Technologies	27
2.3.1 Existing Solutions	27
2.3.2 Limitations and Opportunities	27
2.4 Sensor Technologies and Data Integration.....	28
2.5 Energy Efficiency and Security in Smart Systems	28
2.6 Comparative Analysis	29
2.8 Summary	31
CHAPTER 3: RESEARCH METHODOLOGY	32
3.1 Overview	32
3.2 Project Methodology.....	32
3.3 Iteration 1: Requirement Analysis and Planning	33
3.4 Iteration 2: Prototype Development.....	38
3.4.1 Hardware Assembly	38
3.4.2 Firmware Development	39
3.4.3 Mobile Interface Development	45
3.5 Iteration 3: Iterative Testing and Refinement	46
3.5.1 Functional Testing.....	46
3.5.2 Real-World Simulations.....	46
3.5.3 Security System Testing.....	47
3.6 Iteration 4: Advanced Features Integration.....	48
3.6.1 Security Enhancements and Mobile Alerts	48

3.6.2 Testing and Calibration	49
3.7 Iteration 5: Deployment and Final Evaluation.....	50
3.7.1 System Deployment.....	50
3.7.2 Performance Monitoring.....	51
3.7.3 User Familiarization and Documentation	52
3.8 Summary.....	53
CHAPTER 4: DEVELOPMENT AND IMPLEMENTATION	55
4.1 Introduction.....	55
4.2 Software Installation and Configuration.....	55
4.3 Initial Testing and Component Verification	56
4.4 Hardware Components and Integration	58
4.5 Circuit Design and Wiring	58
4.6 Mobile Interface Implementation	64
4.6 Programming and Algorithm Logic	70
4.7 Prototype Assembly and Smart Home Design.....	76
4.8 Testing, Calibration and Troubleshooting.....	83
4.9 Final Implementation.....	85
4.10 Summary.....	86
CHAPTER 5: TESTING AND EVALUATION	87
5.1 Introduction.....	87
5.2 Functional Testing.....	87

5.3 Non-functional Testing	89
5.4 Scenario-Based Validation	90
5.5 Analysis.....	91
5.6 Conclusion	92
CHAPTER 6: CONCLUSION AND FUTURE WORKS	93
6.1 Introduction.....	93
6.2 Objectives Achieved	93
6.3 System Limitations and Constraints	94
6.4 Future Work	95
6.5 Conclusion	96
REFERENCES.....	97
APPENDIX A: SOURCE CODE.....	101
APPENDIX B: PROJECT SCHEDULE	109
APPENDIX C: REPORT SIMILARITY PERCENTAGE	0

LIST OF FIGURES

Figure 1: Gantt Chart of Project Schedule (FYP 1)	23
Figure 2: Gantt Chart of Project Schedule (FYP 2)	23
Figure 3: The Iterative Development Model Used in This Project Based on Agile Principles	33
Figure 4: Data Flow and The IoT Architecture	38
Figure 5: Rain Sensor Working Principle (Kernbach et al., 2019)	40
Figure 6: LDR Working Principle (Sawkare, 2024)	41
Figure 7: Magnetic Reed Switch Logic	41
Figure 8: Firmware Flowchart	44
Figure 9: Intrusion Alert Push Notification via Blynk App	47
Figure 10: Flow of Security Features	49
Figure 11: Servo Motor Testing	56
Figure 12: Rain Sensor Testing	56
Figure 13: LDR Sensor Testing	57
Figure 14: DHT11 Sensor Testing	57
Figure 15: Whole Wiring Diagram	62
Figure 16: Close Up of Wiring Diagram of the Main System	62
Figure 17: Wiring Diagram of Wall Mounted Components	63
Figure 18: Placement of MC-38 in Prototype	63
Figure 19: Sensor Components from Outside	64
Figure 20: Successful Connection with Blynk via Wi-fi	65
Figure 21: Screenshot of Datastream Setting in Blynk Web Console	67
Figure 22: Blynk Dashboard at Daytime	68
Figure 23: Blynk Dashboard at Nighttime	69
Figure 24: Algorithm in LaTeX Format	75

Figure 25: Prototype Diagram – Inside from Side.....	77
Figure 26: Prototype Diagram – Inside from Center	78
Figure 27: Prototype Diagram – Front View	79
Figure 28: Prototype Diagram – Close Up of Inside from Center	79
Figure 29: Prototype Diagram – Close Up of Inside from Side	80
Figure 30: Prototype Diagram – Close Up of Main System.....	81
Figure 31: Prototype Diagram – Close Up of Window.....	82

LIST OF TABLES

Table 1: Project Schedule.....	21
Table 2: Comparative Analysis of Existing Window Automation Systems and IntelliVent	29
Table 3: Hardware Component Specifications	33
Table 4: Wiring Table.....	59
Table 5: Blynk Datastream Configuration for IntelliVent.....	65
Table 6: Summary of Hardware Testing and Calibration	83
Table 7: Functional Test Results	88
Table 8: Non-Functional Test Results	89

CHAPTER 1: INTRODUCTION

1.1 Background of Study

The rapid proliferation of the Internet of Things (IoT) has ushered in a new era of smart home automation, fundamentally transforming how individuals interact with their living spaces. Modern homeowners increasingly expect not only convenience but also enhanced security, privacy and energy efficiency. Among the various components of a smart home, windows play a pivotal yet often overlooked role. As primary interfaces between indoor and outdoor environments, they contribute significantly to natural ventilation, daylighting, and passive solar heating. These factors are that directly influence energy consumption and indoor comfort.

Despite their importance, window management in most homes remains largely manual and reactive. Homeowners frequently forget to close windows during sudden weather changes such as rainstorms, leading to interior water damage and increased energy loss. Conversely, closed windows during favourable conditions reduce opportunities for natural cooling and ventilation, thus increasing reliance on energy-intensive Heating, Ventilation, and Air Conditioning (HVAC) systems and artificial lighting. Additionally, windows left open at night pose a serious security risk, making homes vulnerable to intrusion.

While the smart home market has advanced rapidly in areas like lighting, thermostats, and surveillance systems, intelligent window automation remains underdeveloped. Existing solutions are often restricted to basic timers or rely on remote weather APIs that may not reflect real-time, localized environmental conditions. They also typically fail to adapt dynamically to factors such as time of day, rain, or sunlight intensity.

This project, IntelliVent, is designed to address the inefficiencies of traditional window management systems by introducing a localized, sensor-based automation solution. By

integrating advanced environmental sensors like rain, light, temperature and humidity sensors, a sliding window mechanism, and a mobile IoT interface, IntelliVent enables intelligent, real-time adjustment of windows based on actual environmental conditions. The system includes features such as automatic rain detection and closure, time-based automation such as closing at night for security, and an integrated intrusion alarm via magnetic reed switch for enhanced safety.

Furthermore, the mobile interface allows users to receive real-time notifications and remotely control their window settings, ensuring both online and offline operability. By reducing the need for manual intervention, IntelliVent promotes energy sustainability, occupant comfort, and overall home security. Ultimately, this project envisions a smarter, safer and more energy-efficient living environment by transforming how households interact with one of their most vital architectural elements which is the window.

1.2 Problem Statement

Despite advancements in smart home technologies, the vast majority of residential and small commercial buildings still rely on manual methods for window control. This outdated approach introduces multiple inefficiencies and risks that impact energy consumption, occupant convenience as well as home security. Open windows during adverse weather such as sudden rainstorms can cause interior water damage, increase HVAC loads and result in substantial energy loss (Feng et al., 2024) (Niu et al., 2022). Additionally, the need for constant manual operation in response to changing environmental conditions presents a persistent inconvenience for occupants.

From an energy standpoint, improperly managed windows are a major source of inefficiency. Open windows during hot days allow conditioned air to escape and allow natural

ventilation, while during cold nights, they permit warm air to leak, forcing HVAC systems to overcompensate. This results in higher energy bills and a larger carbon footprint. In terms of security, windows left open or unlocked, especially at night or when the home is unoccupied, which pose significant safety vulnerabilities and may serve as potential entry points for intruders when homeowner is not around.

While smart home systems have made strides in automating lighting and climate control, smart window automation remains underdeveloped. Existing solutions are typically limited to basic timers or manual app-based controls. Even those that incorporate weather awareness often depend on remote internet-based forecasts, which may not reflect hyper-local, real-time environmental changes like a passing shower. These limitations undermine the potential for seamless and adaptive automation expected in modern smart homes (Niu et al., 2022) (Feng et al., 2024).

There is a notable gap in the availability of affordable and integrated solutions that combine localized weather sensing like rainfall, ambient light detection like day and night, time-based automation and real-time security status. This project, IntelliVent, seeks to bridge that gap by developing a smart window automation system that uses on-site environmental sensors and intelligent control logic. The system aims to autonomously optimize window operation for energy efficiency, occupant comfort, and security. Through the integration of weather-responsive and time-based automation, IntelliVent provides a sustainable and user-friendly solution for modern window management.

1.3 Objectives of Study

The objectives of this project are as follow.

- To develop a smart window system for sliding windows that autonomously operates based on real-time weather and ambient conditions.
- To develop a mobile interface using the Blynk IoT platform that allows users to monitor real-time sensor data.
- To enhance home security by integrating an intrusion detection system using a magnetic reed switch, which triggers an audible alarm and sends real-time notifications to the user.

1.4 Scope and Limitations

The IntelliVent project is focused on the development of a smart window automation system tailored for residential and small commercial buildings with an emphasis on sliding windows. The primary goal is to create a proof-of-concept prototype that automates window operation using real-time environmental data and offers remote control capabilities via a mobile IoT interface.

The project scope is defined by the following functional and technological parameters:

- **Autonomous Window Control:** The system automatically opens the window during the day if no rain is detected and closes it at night for security and energy conservation, regardless of weather conditions.
- **Automatic Rain Response:** Upon detecting rainfall, the system immediately closes the window, overriding all other logic to prevent water ingress and damage.
- **Intrusion Detection:** A magnetic reed switch monitors the window's position. If the window is forcefully opened while marked as "closed," an audible alarm is triggered in

repeating cycles, and a push notification is sent to the user. The alarm persists until the window is closed properly or manually silenced via the mobile app.

- **Technological Stack:**
 - **Hardware:** ESP32 microcontroller, YL-83 rain sensor, Light Dependent Resistor (LDR), magnetic reed switch, SG90 360° continuous rotation servo motor, and a piezoelectric buzzer.
 - **Software:** Firmware written in C++ using the Arduino IDE, with mobile/cloud integration via the Blynk IoT platform.
- **Target Environment:** The system is designed for standard sliding windows in typical residential settings. Support for other linear window types is a secondary consideration.

While IntelliVent demonstrates significant progress toward intelligent window automation, several limitations exist due to practical, technical and economic constraints of the prototype stage.

- **Sensor Accuracy:** The prototype uses affordable sensors such as YL-83 rain sensor and LDR are suitable for early-stage development. These lack the precision needed to differentiate between light drizzle and heavy rainfall, potentially affecting decision accuracy.
- **No Obstacle Detection:** The system currently does not include safety mechanisms such as ultrasonic or infrared sensors to detect obstructions in the window's path during opening or closing.
- **Power and Network Dependency:**
 - A continuous 5V power supply is required for full functionality.
 - While basic automation logic can function offline, features such as remote control, real-time monitoring, and push notifications depend on a stable Wi-Fi connection.

- **Mechanical Constraints:** The actuation mechanism is optimized for a lightweight prototype. Without hardware upgrades, it may not generate sufficient torque to operate larger or heavier real-world windows.
- **Integration Challenges:** Retrofitting older buildings or homes with non-standard window designs may face compatibility issues which limits widespread adoption without customization.
- **Environmental Suitability:** The current design may not perform optimally in extreme climates where rapid or frequent environmental changes require more nuanced control or manual intervention.

These scope boundaries and limitations provide a solid foundation for future work, including improvements in sensor accuracy, mechanical strength, safety integration, and system resilience, toward developing a fully commercial and scalable smart window solution.

1.5 Definition of Terms

1. **Smart Home:** Integration of technologies to automate tasks, enhance convenience, safety, and energy efficiency, and support comfort and accessibility (Kadam et al., 2015).
2. **IoT (Internet of Things):** A network of physical objects ("things") embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the internet (Kowsalyadevi, 2023).
3. **Microcontroller:** These serve as the central processing units, managing inputs from sensors and controlling outputs (Eltayeb, 2024).
4. **Firmware:** A specific class of software providing low-level control for a device's hardware. The C++ code developed in the Arduino IDE and uploaded to the ESP32 is the system's firmware (Paladines Condoy et al., 2024).

5. **Sensors and Actuator:** Sensors like rain sensors detect environmental conditions, while actuators perform physical actions based on commands such as the servo motor moving the window. (Eltayeb, 2024) (Surana et al., 2023).
6. **HVAC Systems:** HVAC stands for Heating, Ventilation, and Air Conditioning, crucial for maintaining indoor air quality and comfort (Figueiredo et al., 2021).
7. **User Interface (UI):** Interfaces such as mobile apps and voice commands enable user interaction with smart home systems, facilitating remote control and automation (Bethel, 2023).
8. **IoT Platform (Blynk):** A suite of software components that enables the connection of hardware to the cloud, communication, and the creation of user-facing applications. In this project, Blynk is the IoT platform used, providing the mobile app, cloud connectivity, and hardware libraries for the ESP32. (Sarkar et al., 2024)

1.6 Methodology

The IntelliVent project adopts iterative methodology, emphasizing flexibility, continuous improvement and user-centred design. To implement this, the development is structured into a series of fixed-length iterations (or cycles), drawing on principles from Agile frameworks to ensure adaptability and thoroughness.

1. **Requirement Analysis and Planning:** This initial sprint focused on defining the core functionalities. Key requirements identified were automation based on local rain and light sensors, a user-friendly mobile interface and an intrusion detection system. This phased involved selecting the microcontroller, the necessary sensors and IoT platforms for its rapid development capabilities.
2. **Prototype Development:** A Minimum Viable Product (MVP) was constructed to validate the core automation logic. This involved integrating the primary sensors with

the servo motor. The initial firmware was written to control the window based on rain and day/night conditions and a basic mobile dashboard was created for monitoring.

3. **Iterative Testing and Refinement:** The MVP underwent rigorous testing to ensure its stability and reliability. This phase focused on calibrating sensor thresholds, optimizing the system's response time and refining the servo motor's movement to prevent mechanical issues. Both hardware and firmware were adjusted based on these initial tests.
4. **Advanced Features Integration:** With a stable core system, the focus shifted to enhancing functionality. The security subsystem was integrated. The firmware was updated to include the logic for the audible intrusion alarm and to send real-time push notifications to the user's mobile device.
5. **Final Deployment and Evaluation:** The complete system with all automation and security feature was deployed into the final physical prototype. Comprehensive, scenario-based testing was conducted to validate the seamless interaction between all components. Performance data was collected to evaluate the system's overall effectiveness and user-friendliness which results in a market-ready solution.

This iterative agile approach ensures the delivery of a robust and user-focused smart window automation system which allows for progressive development and continuous validation at each stage.

1.7 Significance of the Project

The IntelliVent smart window automation system enhances energy efficiency, comfort, security and privacy in modern homes and small businesses. Through its automated control of the window opening based to weather and time, it minimizes artificial heating, cooling, and

lighting, leading to lower energy costs and a smaller environmental footprint. It optimizes natural ventilation and daylight, promoting a healthier indoor environment.

It closes and opens the window automatically when it rains preventing water damage and maintaining privacy and security at night. With an integrated lock and an alarm system, safety is increased by preventing unauthorized entry. Overall, IntelliVent illustrates how IoT can be practically applied to make smart home living cost-effective, comfortable, and eco-friendly.

1.8 Project Schedule

This final year project I starts on 7th October 2024 and has completed on 19th January 2025 which takes more than 3 months, while final year project II starts on 11th March 2025 and is expected to be completed on 28th July 2025. The details of the project are described and presented below in Table 1.

Table 1: Project Schedule

Task	Start Date	End Date
Project Proposal	7 October 2024	14 November 2024
Brief Proposal	7 October 2024	18 October 2024
Submission of the Approved Brief Proposal	19 October 2024	20 October 2024
Feedback and Comment from Reviewer/Examiner	21 October 2024	28 October 2024
Submission of Full Proposal	29 October 2024	14 November 2024
FYP 1	15 November 2024	17 January 2025
Chapter 1: Introduction	15 November 2024	20 November 2024

Submission of Chapter 1	21 November 2024	21 November 2024
Chapter 2: Literature Review	22 November 2024	12 December 2024
Submission of Chapter 2	13 December 2024	13 December 2024
Chapter 3: Research Methodology	14 December 2024	4 January 2025
Submission of Chapter 3	5 January 2025	5 January 2025
Submission of FYP 1 Final Report, Paper Assessment and Presentation Video	6 January 2025	17 January 2025
FYP 2	11 March 2025	28 July 2025
Submission of the proposed/revised structure of FYP report, title and Gantt chart	11 March 2025	1 April 2025
Submission of First Draft for Chapter 4	1 April 2025	15 May 2025
Submission of First Draft for Chapter 5, 6, and Abstract for Paper	15 May 2025	31 May 2025
Submission of First Draft for FYP 2 Full Report and Paper	31 May 2025	10 June 2025
Submission of Final Report, source code, installation kits, user manual and paper for assessment	10 June 2025	23 June 2025
FYP Symposium	23 June 2025	1-2 July 2025
Amendment and Modification Period for FYP	1 July 2025	18 July 2025
Submission of Final Report	18 July 2025	28 July 2025

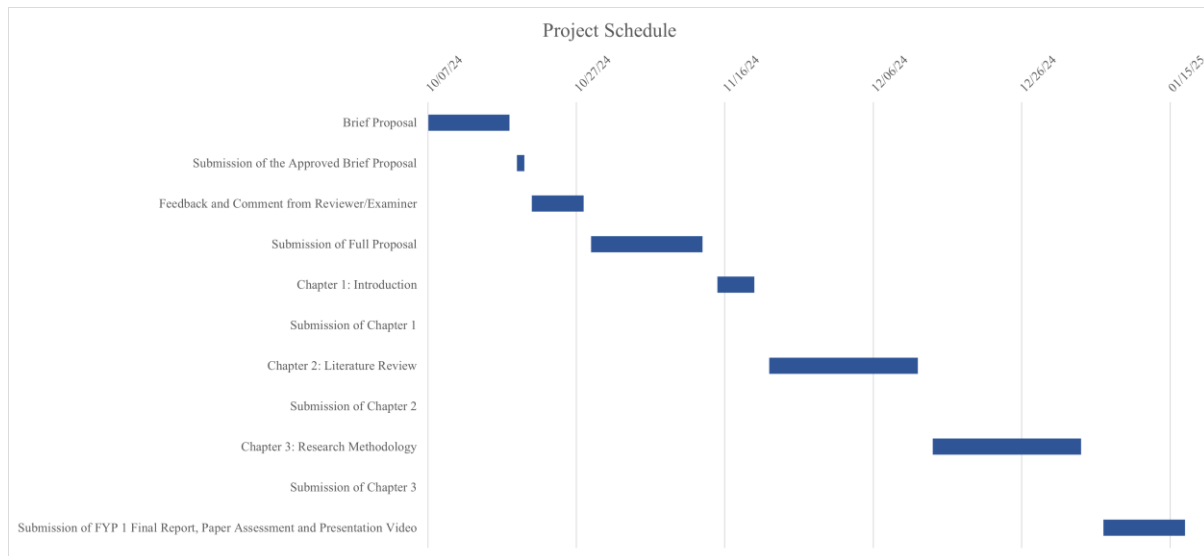


Figure 1: Gantt Chart of Project Schedule (FYP 1)

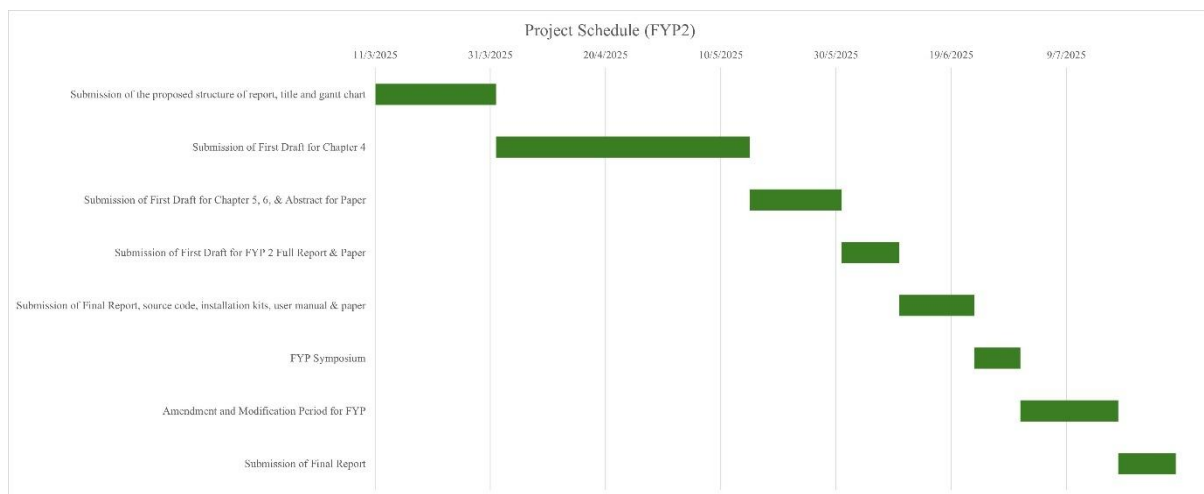


Figure 2: Gantt Chart of Project Schedule (FYP 2)

1.9 Expected Outcome

Upon completion, the IntelliVent project is expected to deliver a fully functional smart window automation prototype that integrates environmental responsiveness, user control and home security. The core objective is to create a robust and real-time system capable of intelligently managing a window's position based on weather conditions and light levels. Specifically, the system will autonomously close the window when rain is detected to prevent

water ingress and at night to improve security and privacy. Conversely, it will open the window during clear daytime conditions to promote natural ventilation and energy efficiency.

Another key outcome is the implementation of an integrated intrusion detection module. The system can identify unauthorized window openings when the window is supposed to be closed. This event will trigger a repeating audible alarm to deter intruders and send an instant push notification to the user. The alarm will continue until the window is securely closed again or the user manually silences it from the mobile app, ensuring both local deterrence and remote situational awareness.

The project also aims to provide a user-friendly and responsive mobile interface. This dashboard will allow users to monitor real-time sensor data as well as check the current state of the window. Users will have the ability to manually override the automation settings at any time for added control and flexibility.

Ultimately, IntelliVent will serve as a proof-of-concept that demonstrates how locally gathered sensor data when integrated with cloud-based IoT tools can be used to develop a smart home solution that enhances daily convenience, improves home security and supports sustainable living through energy-conscious automation.

1.10 Summary

This chapter has introduced the IntelliVent project, a smart window automation system designed to overcome the limitations of manual window management. The project leverages the Internet of Things (IoT) to create a system that intelligently responds to its immediate environment. It addresses the crucial role windows play in a home's energy efficiency, natural ventilation, and security.

The core objective is to develop a system that autonomously operates a window based on local sensor data, closing for rain and nighttime conditions while opening for clear daylight. The project also integrates a critical security feature, which detects unauthorized entry and triggers both a local alarm and a remote mobile notification.

A user-friendly mobile application provides capabilities for remote monitoring. The development follows an Agile methodology, ensuring an iterative and robust process focused on reliability and user satisfaction. Ultimately, IntelliVent aims to deliver a practical and scalable smart home solution that enhances convenience, security and energy efficiency through intelligent and real-time automation.

CHAPTER 2: LITERATURE REVIEW

2.1 Background of Study

The following project development is based on developments in the Internet of Things (IoT), energy-efficient technologies, and home automation. The chapter presents a literature review of existing technologies related to smart home automation with the focus on window management, sensor integration and energy optimization. By examining related works, this review identifies the gaps in current solutions that IntelliVent aims to address.

2.2 Review of Existing Smart Home Systems

IoT integration into smart home systems revolutionizes the traditional home environments through automation, energy efficiency, and security. While the development of home automation has seen huge leaps forward through systems such as Google Nest and Amazon Alexa, there is still great scope for the automation of windows. This gap presents an opportunity to improve energy efficiency and indoor comfort by incorporating window management into existing IoT frameworks (Oseiwe, 2024).

Current smart home systems predominantly emphasize HVAC optimization, often neglecting the significant benefits of natural ventilation through windows. This oversight presents an opportunity to enhance energy efficiency and indoor comfort by integrating window management into existing IoT frameworks. For example, natural ventilation can significantly reduce indoor temperatures, with studies showing potential reductions of up to 8 °C when properly scheduled (Yoon et al., 2023). Moreover, utilizing window management systems can optimize airflow and thermal mass, improving overall building resilience and energy savings (Yoon et al., 2023).

2.3 Window Automation Technologies

2.3.1 Existing Solutions

Automated window systems have been explored in various contexts, primarily for commercial buildings. Systems such as Velux Active integrate basic automation for opening and closing windows based on temperature and air quality (Velux, 2021). However, these solutions often lack advanced weather prediction capabilities and do not incorporate time-based automation.

Recent advancements include predictive technologies such as deep learning models, which predict window states based on environmental data, achieving high accuracy rates (Banihashemi et al., 2024). Additionally, weather-responsive systems like those using rain sensors have demonstrated practical applications, automatically closing windows during rain to prevent interior damage (A'roof & Iswanto, 2024). These innovations demonstrate the potential to address the limitations of current automatic window systems, including their reliance on predefined conditions and outdated sensor technologies.

2.3.2 Limitations and Opportunities

The limited functionality in current window automation systems underlines the need for more adaptable solutions. By integrating real-time weather data and predictive analytics, IntelliVent addresses these shortcomings, ensuring windows respond dynamically to environmental changes and user preferences.

2.4 Sensor Technologies and Data Integration

Sensors are fundamental to the functionality of automated systems. Advanced sensors, like rain and light detectors, enable real-time data collection which enhance the responsiveness of smart systems. Multi-sensor networks enable comprehensive environmental data collection, facilitating accurate monitoring of conditions such as temperature and humidity (Bhardwaj et al., 2024).

The integration of IoT platforms like Blynk into systems like IntelliVent enhances user control and system monitoring. These platforms facilitate real-time data visualization and complementing the on-site sensor automation.

2.5 Energy Efficiency and Security in Smart Systems

Smart systems have proven to be contributing to reduction of energy consumption and enhance security. Automated window systems contribute to energy efficiency by optimizing natural ventilation and reducing HVAC loads. Studies show that smart home technologies can lead to energy savings of up to 25% (Singh et al., 2024).

Security features, such as automated locking mechanisms and intrusion detection, further enhance the value of smart systems. IntelliVent incorporates these capabilities, ensuring homeowners benefit from both energy savings and improved safety (Sinha et al., 2024).

2.6 Comparative Analysis

Table 2 provides a comparative analysis of existing solutions and IntelliVent, highlighting key features and innovations.

Table 2: Comparative Analysis of Existing Window Automation Systems and IntelliVent

Feature	Existing Solutions	IntelliVent
Automation Logic		
Weather-Based Automation	Often relies on internet weather APIs or single-function sensors like temperature only	Real-time, hyper-local automation using an on-site rain sensor.
Time-Based Automation	Typically limited to user-set schedules such as close at 10 PM	Integrated day/night automation based on ambient light levels (LDR).
User Interaction		
Mobile Interface	Basic open/close commands. May lack real-time sensor feedback or customization.	Advanced dashboard (Blynk) with real-time sensor data, status logs, and manual off alarm toggle.
Security Integration		
Security Features	Minimal or non-existent. At most, a simple "closed" status indicator.	Integrated intrusion detection system with a magnetic reed switch.
Alert System	Generally no alerts for security breaches.	Dual-alert system: local audible buzzer and remote push notifications.

Technical Components		
Microcontroller Used	Varies; often proprietary controllers in commercial systems or basic Arduino in DIY.	ESP32 Microcontroller with built-in Wi-Fi and superior processing power.
Actuator Type	Commonly linear actuators or specialized motors, which can be expensive.	Cost-effective continuous rotation servo motor for a scalable prototype design.
Connectivity	Bluetooth for short-range control or Wi-Fi for API access.	Wi-Fi for robust cloud connectivity and remote management via Blynk.
Key Limitation	Lack of integrated intelligence. Systems are often single-purpose and do not combine weather, time and security into a cohesive, autonomous logic.	Prototype-level mechanics. The current design is not scaled for heavy, real-world window loads and lacks obstruction detection.

As demonstrated in Table 2, while existing solutions offer some degree of automation, they often operate in silos. Commercial systems may provide reliable scheduling but lack the responsiveness of local sensors, whereas many DIY projects fail to integrate robust security features. IntelliVent distinguishes itself by creating a holistic system where weather, time, and security are not separate functions but are integrated into a single, hierarchical decision-making logic managed by a capable ESP32 microcontroller. The key innovation lies in its ability to autonomously react to its immediate, local environment while simultaneously providing a comprehensive security and remote monitoring framework, addressing a significant gap in the current market.

2.8 Summary

This chapter reviewed the existing literature and technologies related to smart home automation, window management and energy optimization. It identified the limitations of current systems, such as their reliance on static conditions and lack of predictive capabilities. By integrating real-time weather data, advanced sensors and security features, IntelliVent offers a significant improvement over existing solutions. These findings provide a strong foundation for the development and implementation of IntelliVent, aligning with the project's objectives to enhance energy efficiency, security, privacy and user convenience.

CHAPTER 3: RESEARCH METHODOLOGY

3.1 Overview

This chapter presents the detailed description of the research methodology adopted in the development of the IntelliVent smart window automation system. It outlines the structured steps taken in designing, implementing, and validating the system to ensure that each phase of development aligns with the project's objectives and user requirements. The chapter delves into the iterative Agile methodology employed, emphasizing its suitability for IoT-based systems that demand real-time responses and user-centric functionalities. Overall, this chapter provides a detailed roadmap of the research methodology, ensuring clarity and reproducibility for future enhancements or related projects.

3.2 Project Methodology

As introduced in Chapter 1, an iterative and incremental methodology is implemented in the development of the IntelliVent smart window automation system. This approach draws heavily from Agile principles, which are ideal for IoT projects that require flexibility and continuous feedback. This methodology enables the development process to be divided into manageable, time-boxed iterations. Hence, the time management and process development of the system will be more manageable and minimise cost damage with a proper plan throughout the project. The project is structured into five distinct iterations, each with defined goals and deliverables. The following sections explain each iteration in detail.

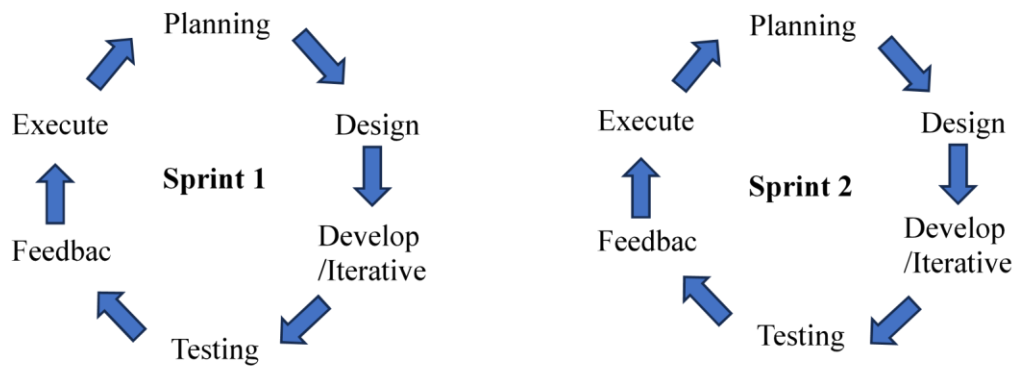


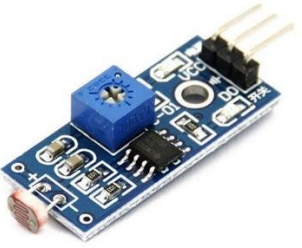
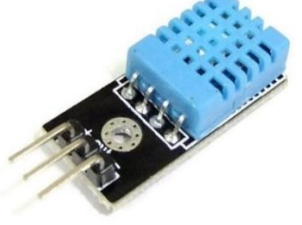
Figure 3: The Iterative Development Model Used in This Project Based on Agile Principles

3.3 Iteration 1: Requirement Analysis and Planning

The primary objective of the first sprint was to establish a solid foundation for the IntelliVent system by clearly defining its requirements, identifying the core functionalities and analyzing the available technologies and resources. The sprint began with a comprehensive requirements-gathering phase which focused on understanding the practical needs of the system and translating them into actionable development goals such as rain-sensing automation, daylight optimization, integrated intrusion alarm and remote monitoring and control. Table 3 shows components that were chosen to be used in this project.

Table 3: Hardware Component Specifications

Component	Function	Specifications
	Central control unit for sensors, actuators, and IoT	Offers dual-core CPU, Wi-Fi, Bluetooth and has GPIO of 36 Pins

Servo Motor SG90 (360) 	Controls sliding window movement	Continuous rotation servo (infinite rotation by speed control and not fixed position)
Rain Sensor (YL-83) 	Detects precipitation to trigger window closure	Analog rain detection sensor and has detection area of about 4cm²
Light Sensor (LDR) 	Monitors daylight intensity for daylight optimization	Passive light-sensitive resistor and response time of 20ms – 30ms
Temperature and Humidity Sensor (DHT11) 	Measures environmental temperature and humidity	Detects temperature range of 0–50°C and humidity range of 20% to 90% with accuracy of ±5% RH

Large Piezo Buzzer Module 	Provides audible alarm to alert surrounding before moving the window	Generates sound when powered and does not need PWM signal, sound output at 85dB
Magnetic Contact Switch Sensor (MC-38)(Normally Open) 	Detects the opening or closing of doors and windows and unauthorized window opening attempts	A normally open magnetic reed switch rated up to 100VDC and 500mA with a 15–25 mm activation distance.

Each component is chosen based on its compatibility, cost-effectiveness, and suitability for this project's goals. First, the ESP32 microcontroller acts as the central control unit, which manages data from sensors and controlling actuators. The ESP32 is chosen for superior processing, having a dual-core 240 MHz processor with 520 KB SRAM, hence enabling it to handle multiple sensors and IoT platform integrations smoothly. Moreover, on-board Wi-Fi and dual-mode Bluetooth are available for appropriate connectivity in real time with the mobile app. The ESP32 enjoys greater performance and it is cheaper too, while Arduino Nano 33 IoT operates on one-core 48 MHz. Because the ESP32 has up to 36 GPIO pins, this means more flexibility compared to Nano 33 IoT with its 14 digital pins.

To automate window movement, a servo motor is selected as the actuator. Servo motors are affordable, widely available and compact, which makes them ideal for small-scale prototypes. Specifically, the SG90 360° continuous rotation servo motor was chosen due to its ability to rotate continuously in either direction, making it suitable for simulating the sliding motion required in this project. While linear actuators provide direct linear movement, servo

motors are generally easier to implement in constrained spaces and offer sufficient control for this application. Their ease of integration, compatibility with the ESP32 platform, and lower cost make them a practical and efficient choice for prototyping smart home automation systems.

Next, the rain sensor (YL-83) is an essential component for detecting precipitation and triggering window closures. Its low cost, reliability and simplicity make it an ideal choice for weather-responsive automation. While alternatives like capacitive rain sensors offer higher accuracy, their higher cost and complexity are unnecessary for the system's requirements, where functionality and affordability are prioritized.

For monitoring daylight intensity, the system uses a light-dependent resistor (LDR). The LDR is a cost-effective solution that provides sufficient accuracy for daylight optimization, ensuring windows open to maximize natural light during favourable conditions. While digital sensors like the TSL2561 offer greater precision, they are more complex to integrate and not essential for this system's objectives. The LDR strikes an appropriate balance between simplicity and performance.

The temperature and humidity sensor for monitoring the ambient environment will be DHT11. The DHT11 is suitable for Malaysia's climate, where a temperature range of 0–50°C that fits well with the country's average temperatures of 24–35°C. The DHT22 is available where higher precision applications are needed, or there is a need to measure a wider range of -40 to 80°C. Compared to other sensors like the BMP280, the DHT11 is a cheaper option and tailored to the system's needs.

The security system in the IntelliVent project incorporates two key components to detect and respond to unauthorized access. Central to this mechanism is the MC-38 magnetic contact switch sensor, a reliable and cost-effective normally open reed switch. When the

window is closed, the magnet and sensor remain in close proximity, completing the circuit. If the window is forcibly opened and the magnet is separated from the sensor, the circuit breaks and immediately triggering the intrusion detection response.

As an audible deterrent, a buzzer module is activated upon intrusion. It emits a repeating sequence of four short beeps followed by a brief pause, continuing in a loop until the window is securely closed again. Additionally, the system sends a push notification to the user's smartphone via the Blynk IoT platform, ensuring both local awareness and remote notification of any security breach. This simple yet effective setup enhances home security by providing immediate alerts and continuous feedback during intrusion events.

Finally, the mobile interface is developed using Blynk, a platform designed for quickly building IoT applications with minimal coding. Blynk provides a user-friendly environment for connecting the ESP32 microcontroller to a smartphone via Wi-Fi, enabling real-time monitoring and control. Users can view sensor data and receive status updates through a customizable interface, significantly simplifying app development while maintaining effective remote functionality.

Overall, the components selected for the IntelliVent system were chosen to ensure efficiency, scalability and ease of integration. The ESP32 microcontroller with its built-in Wi-Fi and ample processing capabilities is a strong choice for handling real-time sensor input, servo control and IoT connectivity. The DHT11 sensor, although basic, is sufficient for monitoring temperature and humidity levels in Malaysia's tropical climate. The YL-83 rain sensor and LDR offer cost-effective and responsive solutions for detecting weather and ambient light conditions, essential for triggering automated window actions. For intrusion detection, the MC-38 magnetic contact switch sensor provides reliable performance and greater tamper resistance compared to mechanical alternatives. The buzzer, with its 85dB output, delivers a

clear and audible local alarm to deter unauthorized access, without causing excessive noise pollution. Together, these components form a practical, robust, and budget-conscious smart window automation system that balances performance with user-friendly operation.

3.4 Iteration 2: Prototype Development

3.4.1 Hardware Assembly

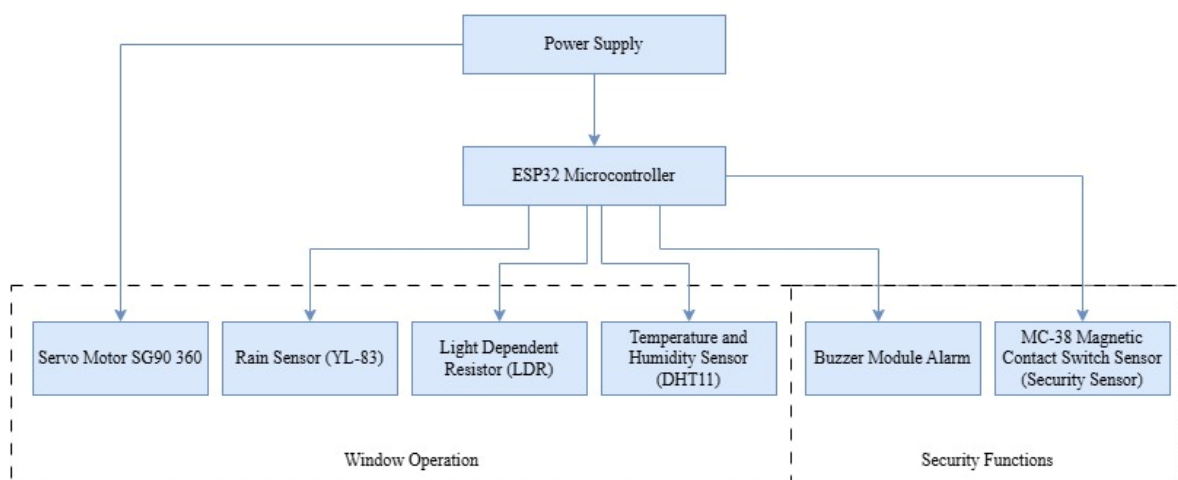


Figure 4: Data Flow and The IoT Architecture

The hardware assembly process emphasizes the seamless integration of sensors, actuators, and the central control unit to enable automated window operations. The sensors, including the rain sensor (YL-83), light-dependent resistor (LDR), and temperature and humidity sensor (DHT11), are strategically mounted on the sliding window prototype. These sensors are connected to the ESP32 microcontroller, which acted as the central hub for collecting environmental data and managing system responses.

The actuator system of the IntelliVent prototype utilizes an SG90 360° continuous rotation servo motor, which is directly interfaced with the ESP32 microcontroller. Based on sensor input and automation logic, the ESP32 sends control signals to the servo motor to either

open or close the sliding window. The motor's rotational motion is mechanically linked to the window panel to simulate realistic sliding behaviour suitable for home ventilation.

To provide accurate window position feedback and enhance the system's automation and security features, the MC-38 magnetic contact switch sensor is integrated with the ESP32. This reed switch detects whether the window is fully closed or open by monitoring the proximity between the magnet and the sensor body. Its input is used both to prevent redundant motor actions and to detect unauthorized opening attempts.

In the event of an intrusion, such as when the window is forced open, the system immediately triggers an audible alert via a buzzer module, which continuously emits bursts of sound until the window is properly closed again, or the user remotely silences it through the Blynk mobile interface.

Power for the IntelliVent system is distributed through a centralized regulated supply that supports the ESP32 microcontroller and all connected environmental sensors, including the DHT11, rain sensor, LDR, and reed switch. To prevent voltage drops and ensure reliable performance, the servo motor is powered separately using an independent 5V source with sufficient current capacity. This separation minimizes electrical noise and avoids power instability that could affect the microcontroller or sensor readings. Altogether, this power architecture ensures smooth operation of the automation system, maintaining responsiveness and reliability in both online and offline conditions.

3.4.2 Firmware Development

The firmware for the system will be developed utilizing the Arduino Integrated Development Environment (IDE), leveraging its C++ based programming language and targeting the ESP32 microcontroller as the central processing unit. The firmware design is

pivotal to achieving the project's core automation objectives specifically enabling the system to autonomously respond to environmental inputs such as rain and ambient light conditions. The planned firmware architecture will encompass several key functional components.

1. Sensor Data Acquisition and Processing

- Rain Detection Logic:** The firmware will be programmed to interface with the YL-83 rain sensor. It will interpret the sensor's digital output signal, where a LOW state signifies the presence of precipitation. Upon such detection, the firmware will initiate the window-closing sequence.

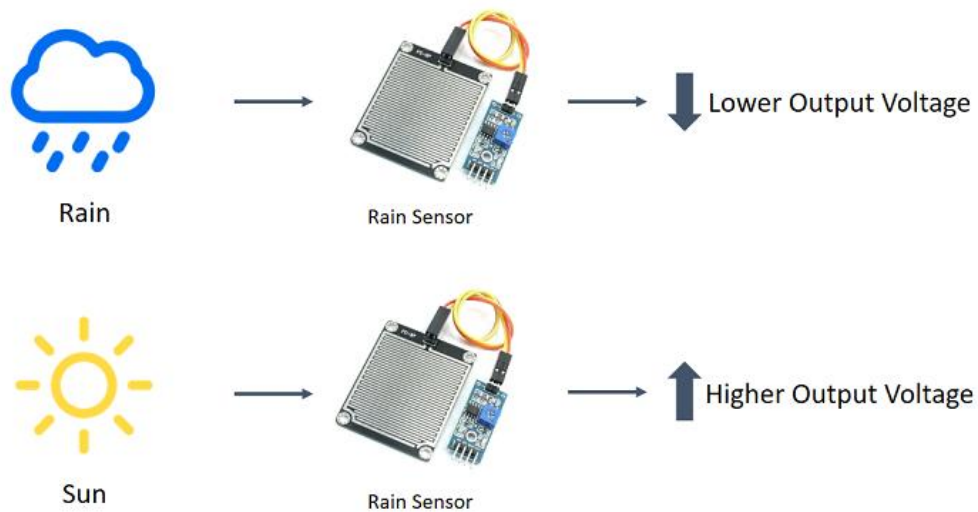


Figure 5: Rain Sensor Working Principle (Kernbach et al., 2019)

- Ambient Light Level Interpretation (Day/Night Determination):** Code will be developed to read and process the analog voltage output from the Light Dependent Resistor (LDR). A configurable threshold value within the firmware will be used to differentiate between daytime and nighttime ambient light levels, influencing window operation accordingly.

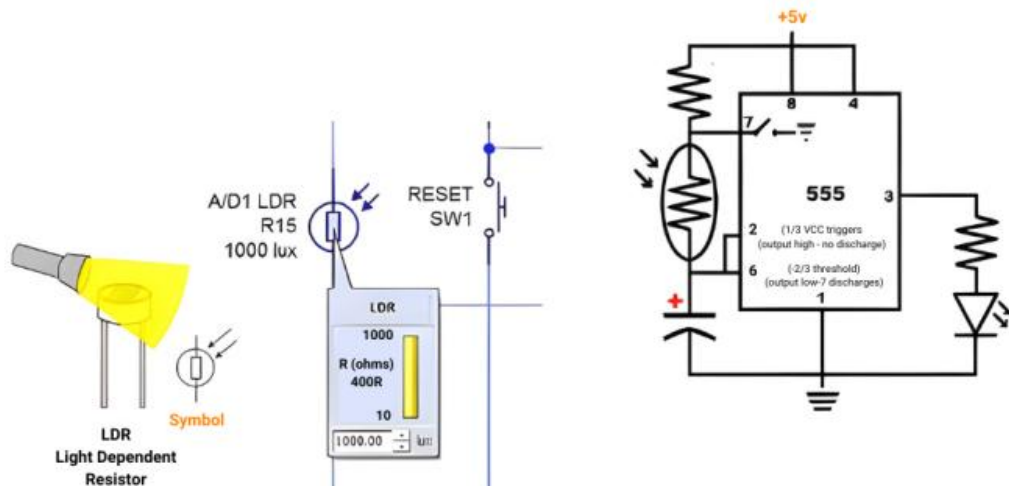


Figure 6: LDR Working Principle (Sawkare, 2024)

- Magnetic Contact Switch Sensor Logic:** When the magnet is near the reed switch (window closed), the circuit remains closed (LOW signal if using internal pull-up). When the magnet moves away (window opens), the circuit becomes open (HIGH signal if using internal pull-up), which can be used to trigger an alert or automation.

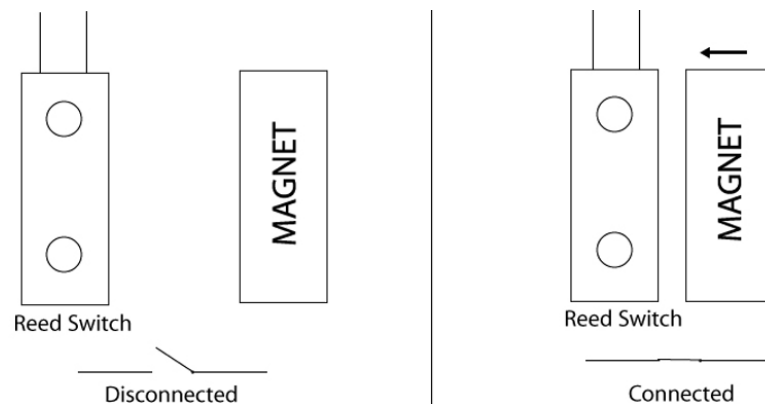


Figure 7: Magnetic Reed Switch Logic

- Humidity and Temperature Environmental Data:** Provision will be made in the firmware structure to integrate the DHT11 sensor for acquiring temperature and humidity data. While initial automation logic may not directly utilize this data, it

will be collected and made available for monitoring and potential future enhancements.

2. Actuator Control Subsystem

- **Servo Motor Management:** Dedicated functions will be developed to precisely control the servo motor, enabling the simulation of window opening and closing actions. This will involve commanding the servo to specific rotational positions or to operate for defined durations to achieve the desired physical window state.

3. Core Automation Algorithm

- The central control loop of the firmware will be designed for continuous environmental monitoring by polling the rain sensor and LDR.
- **Hierarchical Decision-Making:** The automation logic will implement a priority-based decision tree. Detection of rain will supersede all other conditions, compelling an immediate window closure.
- **Conditional Light-Based Operation:** In the absence of rain, the system's behaviour will be governed by the ambient light level: opening during detected daytime conditions and closing during detected nighttime conditions.
- **State Management and Debounce Implementation:** To ensure smooth and reliable operation, the firmware will incorporate state-tracking mechanisms. This will prevent redundant or erratic servo actuations that could arise from transient sensor signal fluctuations, ensuring that window movements are deliberate and correspond to confirmed environmental state changes.

4. Mobile Interface Integration (Blynk Protocol)

- The firmware will integrate the necessary Blynk libraries to establish and maintain a Wi-Fi connection with the Blynk IoT platform.
- **Telemetry Transmission:** The firmware will be responsible for periodically transmitting current sensor readings (rain status, LDR value, temperature and humidity) to the Blynk dashboard, enabling real-time remote monitoring by the user.

Figure 8 provides a visual representation of this intended operational logic. This structured approach to firmware development aims to produce a robust, reliable, and extensible control system for the IntelliVent prototype.

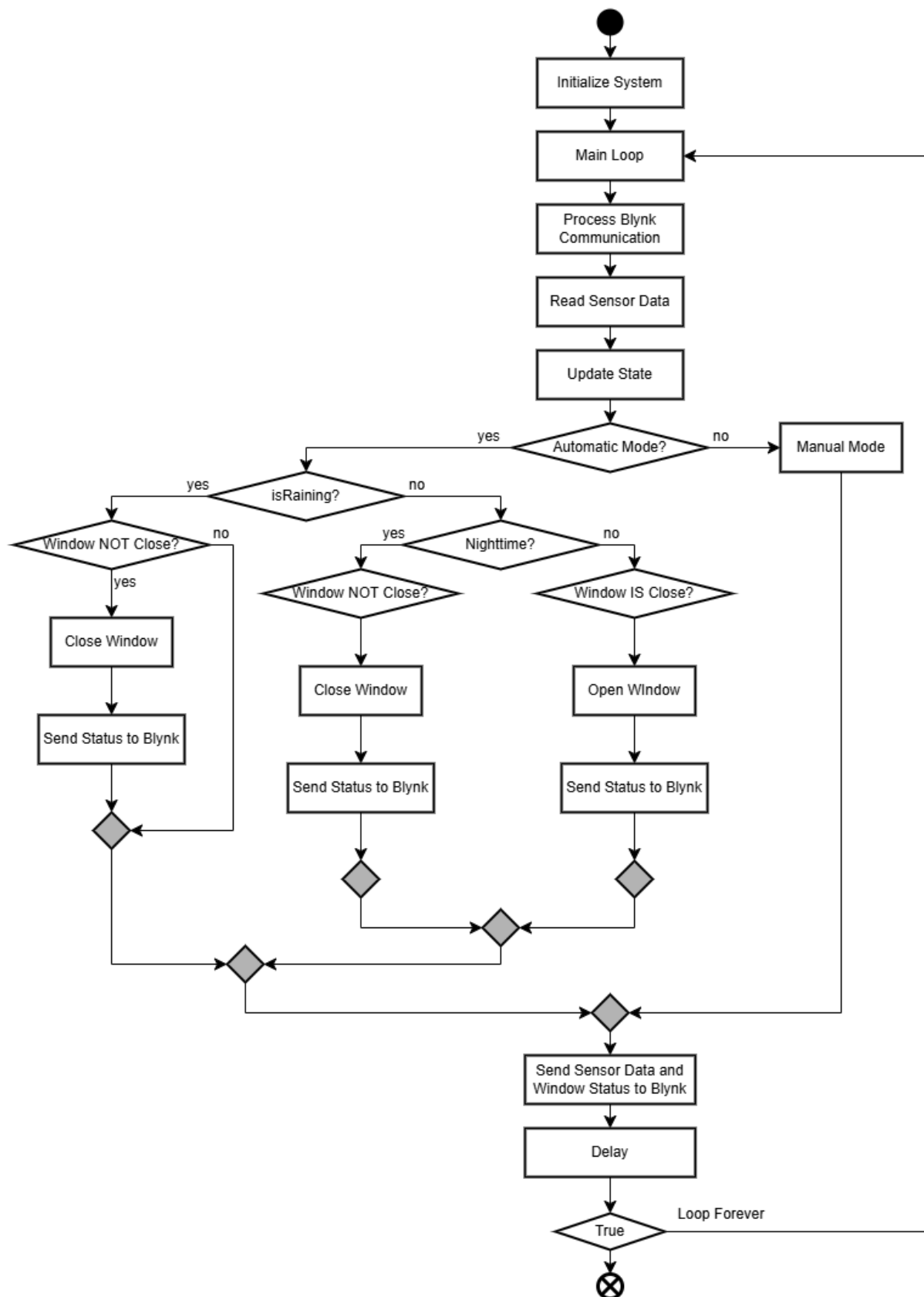


Figure 8: Firmware Flowchart

3.4.3 Mobile Interface Development

The mobile interface for IntelliVent is developed using the Blynk IoT platform. Blynk was selected for its rapid development capabilities that enables the creation of a functional mobile dashboard for monitoring and controlling the ESP32 system without requiring extensive custom mobile application development.

The Blynk application is configured to provide the following core functionalities:

1. Sensor Data Monitoring

- **Rain Status:** A labelled display widget shows the current status from the rain sensor such as raining or clear.
- **Light Level:** A gauge display shows the current reading from the LDR that indicates light ambience.
- **Temperature & Humidity:** Value display widgets show the current temperature and humidity readings.

2. Manually Off Alarm Control

- **Switch Off Buttons:** Allow the user to manually command the buzzer to stop alarm when ringing, temporarily overriding the automatic logic. This provides user flexibility.

3. Real-Time System Feedback

- **A Terminal:** To display live status messages from the ESP32, such as "Rain Detected. Closing Window" or "Intrusion Alert!". This provides a transparent view into the system's decision-making process.

4. Security Integration

- The mobile interface is directly linked to the security system. Upon an intrusion detection event, the Blynk platform sends an instant push notification to the user's

smartphone, ensuring they are alerted to a potential security breach regardless of their location.

The Blynk dashboard is designed with clarity in mind, using standard widgets like buttons, value displays, gauges, and LED indicators to present information and provide control.

3.5 Iteration 3: Iterative Testing and Refinement

3.5.1 Functional Testing

Functional testing forms the foundation of the evaluation process, examining the accuracy and reliability of each sensor and actuator component. The YL-83 rain sensor undergoes extensive testing to verify its precipitation detection capabilities under various conditions. Similarly, the LDR's performance is evaluated across different lighting conditions throughout the day to ensure consistent response. The DHT11 sensor's temperature and humidity reporting accuracy is verified against calibrated reference instruments. Linear motor testing focused on movement precision and reliability, with particular attention to position control and response time.

3.5.2 Real-World Simulations

Real-world simulation testing provides crucial insights into system performance under various environmental conditions. These simulations include controlled water spray tests to verify rain-sensing automation, artificial lighting scenarios to validate daylight-based adjustments, and temperature variation tests using heating and cooling equipment.

3.5.3 Security System Testing

Before performing actual testing, several planned test procedures are prepared to verify the security features of the IntelliVent system. These tests focus on how well the system detects an intrusion and responds with a sound alarm and a mobile notification through the Blynk platform.

- **Intrusion Detection Check:** The MC-38 magnetic reed switch will be tested by moving the magnet away from the sensor to simulate someone opening the window. This helps to confirm that the ESP32 can detect the change and recognize it as a forced entry.
- **Buzzer Alarm Test:** After the intrusion is detected, the buzzer is expected to beep in a repeating pattern to act as a warning. The test will check whether the buzzer turns on correctly and whether the beep pattern matches the one programmed into the system.
- **Blynk Notification Test:** When the system detects an intrusion, a notification should be sent to the user's phone through the Blynk app. This test will check if the alert is sent properly and how quickly it arrives after the intrusion is triggered.

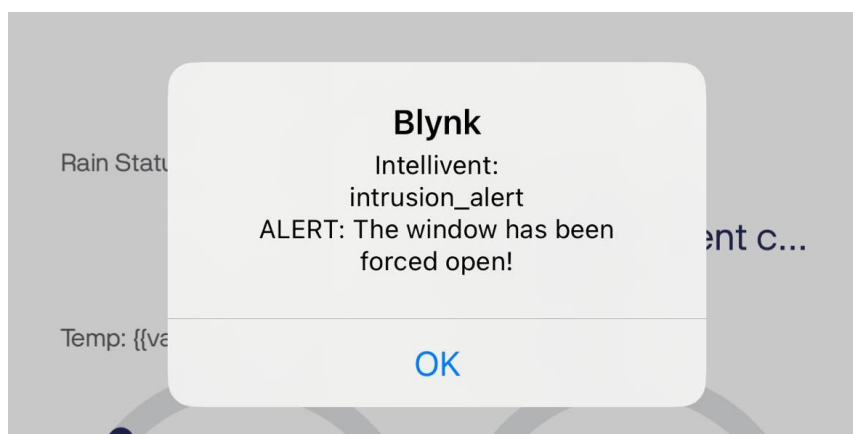


Figure 9: Intrusion Alert Push Notification via Blynk App

- **System Behaviour with Automation:** The test will also include checking how the intrusion alert works alongside other automation actions like opening or closing the window. This is to ensure both functions can run together without causing any problems or delays.

These tests are important to confirm that the IntelliVent security system works as intended like detecting intrusions, alerting users immediately and running smoothly with the rest of the automation logic.

3.6 Iteration 4: Advanced Features Integration

3.6.1 Security Enhancements and Mobile Alerts

Security features are improved with the full integration of the MC-38 magnetic reed switch. The enhanced security system can now send immediate push notifications to the user's mobile device via the Blynk platform when an unauthorized window opening is detected, providing real-time awareness of potential security breaches.

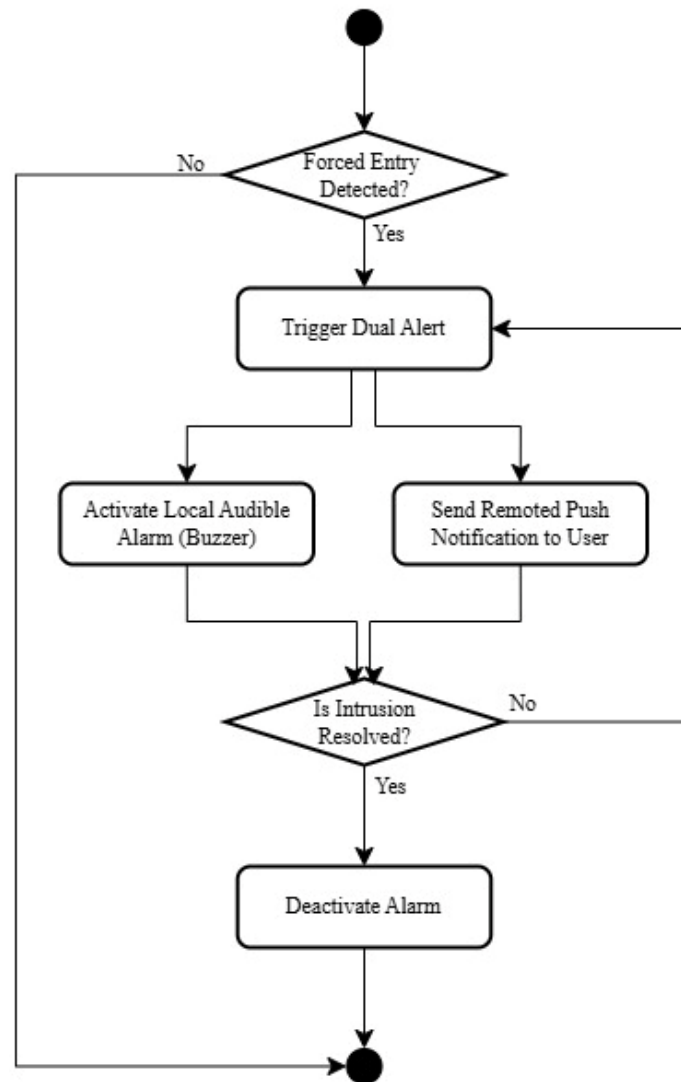


Figure 10: Flow of Security Features

3.6.2 Testing and Calibration

The testing and calibration phase focused on ensuring the long-term durability and accuracy of the IntelliVent system. Long-term simulations are conducted under varying environmental conditions, including simulated rain, fluctuating light intensities, and temperature changes, to assess the hardware's robustness and performance over extended periods. Calibration processes are performed on all sensors and actuators to optimize their precision, such as fine-tuning the rain sensor to avoid false triggers caused by high humidity and adjusting the light sensor for appropriate responses to ambient light changes. Actuators are

calibrated to ensure smooth and precise window movements, minimizing the risk of mechanical failure. Additionally, stress tests are conducted to evaluate the durability of moving parts, including actuators and locking mechanisms, under continuous operation and challenging conditions like power outages and extreme weather. The outcomes of this phase demonstrated the system's reliability and efficiency, with refinements integrated into the final design to ensure optimal real-world performance.

3.7 Iteration 5: Deployment and Final Evaluation

3.7.1 System Deployment

The IntelliVent system is installed in a test smart home environment, incorporating all hardware components such as sensors, actuators, and the ESP32 microcontroller, as well as software elements like the mobile app integration. The deployment process began with configuring the sensor thresholds, which included adjusting rain sensor sensitivity and defining light intensity levels for daylight optimization. The actuation system, including linear motors for window movement, is calibrated to ensure smooth operation. The security features, such as the locking mechanism and alarm system, are tested to verify responsiveness.

During this phase, the system's software is customized to meet specific user preferences. For instance, thresholds for light intensity and rain detection are adjusted based on user feedback and environmental conditions. Additionally, integration with the mobile interface allowed users to monitor and control the system remotely that provides real-time status updates and notifications.

3.7.2 Performance Monitoring

The system undergoes continuous monitoring over several weeks to evaluate its reliability, efficiency, and user experience in real-world conditions. The focus is on ensuring the system performs as expected and identifying any potential areas for improvement. Key performance indicators include:

- **Sensor Accuracy:** The rain sensor, LDR (light sensor), and DHT11 (temperature and humidity) maintain over 95% accuracy during observation. The system consistently responds to environmental changes such as rainfall and shifting light levels, with minimal false readings.
- **Automation Responsiveness:** The window responds within seconds to changes in the environment. It reliably closes during rainfall and opens during daylight, with optimized logic preventing redundant or repeated movements.
- **Energy Efficiency:** Users report a noticeable reduction in HVAC usage due to improved airflow and natural ventilation. The system helps regulate indoor temperature more effectively, particularly during the day.
- **Intrusion Detection Reliability:** The security feature, using the MC-38 magnetic contact switch, detects all unauthorized window openings accurately. The buzzer activates immediately, and push notifications are delivered to the user's mobile device without delay.
- **Offline Functionality:** Even during Wi-Fi interruptions, the core automation logic continues to operate. The system remains capable of controlling the window and detecting intrusions in offline mode, ensuring basic functionality is always maintained.
- **Connectivity Observations:** Occasional issues arise when the Wi-Fi connection is unstable, causing brief delays in push notifications or disconnections in the Blynk

interface. These issues are documented for future improvement, potentially by introducing offline data handling or backup network options.

Overall, the system demonstrates strong and consistent performance in both automation and security operations. The system's responsiveness, accurate sensor data and intuitive mobile interface make it a practical and user-friendly smart home solution.

3.7.3 User Familiarization and Documentation

In alignment with the IntelliVent system's design philosophy of autonomous operation, the user onboarding process eschews traditional, intensive training in favour of a streamlined, documentation-centric approach. The objective is to facilitate effective knowledge transfer, ensuring users can understand the system's automated behaviours and utilize its manual override features with confidence and a minimal learning curve. The methodology is structured around the provision of a comprehensive documentation suite.

The documentation is designed to address three critical areas of user interaction:

1. **Delineation of Core Automation Logic:** The primary user manual clearly delineates the system's operational rules to manage user expectations regarding its autonomous behaviour. It specifies the hierarchy of control logic:
 - **Rain Detection Priority:** The system will automatically close the window upon detecting precipitation, a rule that takes precedence over all other conditions.
 - **Nighttime Security Protocol:** In the absence of rain, the system will close the window upon detecting low ambient light levels to ensure nighttime security, privacy, and energy conservation.

- **Daylight Ventilation Mode:** The system is programmed to open the window during daylight hours, provided it is not raining, to facilitate natural ventilation and lighting.
2. **Mobile Interface Guidance:** A dedicated section of the user manual guides the user through the functionalities of the Blynk mobile graphical user interface (GUI). This guide details the following capabilities:
- **System Monitoring:** Instructions on how to interpret real-time sensor telemetry, including temperature, humidity, light levels, and rain status.
 - **Status Indicators:** A legend for all visual indicators within the application, allowing the user to ascertain the current state of the system at a glance.
3. **System Setup and Troubleshooting:** A concise quick-start guide is provided to facilitate initial system deployment. It outlines the power-up sequence and basic configuration. Additionally, a troubleshooting section in the main manual addresses common issues, such as connectivity problems or sensor recalibration, empowering users to resolve minor issues independently.

This documentation-focused strategy ensures that the system is accessible and transparent to the end-user, reinforcing the project's goal of creating a sophisticated yet user-friendly smart home solution.

3.8 Summary

The chapter concludes by detailing the systematic development approach taken for the IntelliVent smart window automation system. Through careful implementation of Agile methodology across five structured sprints, the project evolved from initial concept to fully functional prototype. The iterative development process, incorporating continuous testing, real-

world simulations, and user feedback, resulted in a refined system equipped with predictive automation, smart film technology, and advanced security features. The methodical approach to planning, component selection, and testing ensured successful realization of both core functionalities and innovative features, creating a robust and user-friendly smart window automation solution.

CHAPTER 4: DEVELOPMENT AND IMPLEMENTATION

4.1 Introduction

This chapter outlines the step-by-step development and implementation of the Intellivent Smart Window Automation System using a mini smart home prototype. The system integrates both electronic and mechanical components to create a working model of a responsive window setup. From initial component testing to software configuration and final assembly. Each phase of development is documented to demonstrate the project's functionality and feasibility.

4.2 Software Installation and Configuration

The project development began with the setup of the software environment which included:

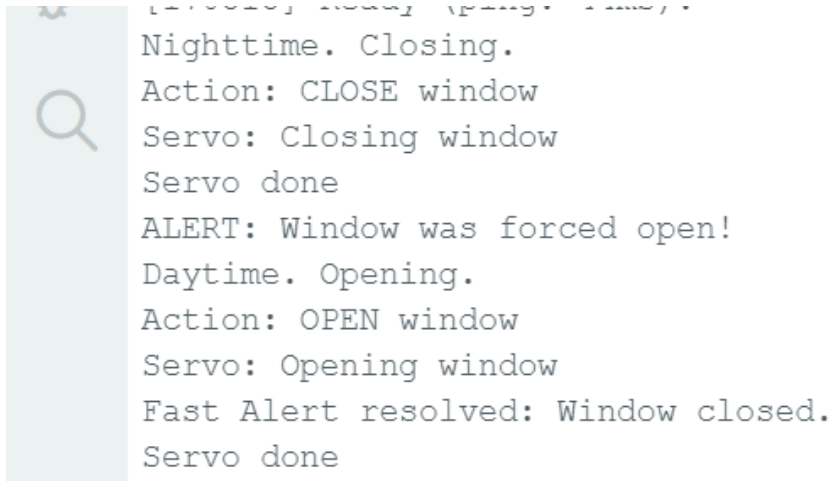
- **Arduino IDE:** Used as the main platform for writing, compiling, and uploading code to the microcontroller.
- **CP210x USB to UART Bridge VCP Driver:** Installed to ensure communication between the ESP32/Arduino and the PC.
- **Essential Libraries:** Included via the Library Manager, particularly Servo.h, for motor control, and libraries for other sensors used in later stages.

The ESP32 microcontroller board was successfully tested for connectivity, code uploading, and serial communication.

4.3 Initial Testing and Component Verification

Before proceeding with full system integration, individual components were meticulously tested to validate their core functionality and ensure basic operational integrity with the ESP32 microcontroller. This preliminary phase is crucial for identifying any defective components or fundamental interfacing issues early in the development cycle.

- **Servo Motor Testing:** A continuous rotation servo motor was first tested independently using a basic sketch. This test aimed to confirm that the motor responded to Pulse Width Modulation (PWM) signals to rotate in both clockwise and counterclockwise directions, simulating the desired range of motion for opening and closing a window.



```

[NIGHTTIME], Raining: No
Nighttime. Closing.
Action: CLOSE window
Servo: Closing window
Servo done
ALERT: Window was forced open!
Daytime. Opening.
Action: OPEN window
Servo: Opening window
Fast Alert resolved: Window closed.
Servo done

```

Figure 11: Servo Motor Testing

- **Rain Sensor (YL-83) Testing:** The YL-83 rain sensor was connected and its digital output was monitored. Water was applied to the sensor's surface to verify that its output transitioned to a LOW state which indicates rain and reverted to a HIGH state when dry. This confirmed its basic water detection capability.

```

(NIGHTTIME), Raining: No
(NIGHTTIME), Raining: No

```

Figure 12: Rain Sensor Testing

- **LDR (Light Dependent Resistor) Sensor Testing:** The LDR was connected to an analog input pin, typically as part of a voltage divider circuit. Its analog readings were observed via the Serial Monitor while varying the ambient light conditions by covering the sensor then exposing it to the bright light. The test confirmed that the sensor provided a distinguishable range of values corresponding to dark and light environments.

```

-----
1%, Light: 4095 (NIGHTTIME), Ra
1%, Light: 4095 (NIGHTTIME), Ra

```

Figure 13: LDR Sensor Testing

- **DHT11 (Temperature and Humidity Sensor) Testing:** The DHT11 sensor was interfaced with the microcontroller and a library-specific sketch was used to read temperature and humidity values. These values were displayed on the Serial Monitor to ensure the sensor was providing plausible environmental data.

```

-----
Humidity: 60.00% | Temperature: 32.60 °C
Humidity: 60.00% | Temperature: 32.20 °C
Humidity: 60.00% | Temperature: 32.00 °C

```

Figure 14: DHT11 Sensor Testing

- **Buzzer Testing:** The piezo buzzer was connected to a digital output pin. A simple sketch was employed to toggle the pin HIGH and LOW which verifies that the buzzer produced a clear audible tone when activated.

These preliminary tests confirmed the operational status of each key component and helped identify any immediate limitations. This allowed for early refinements in the control logic and hardware interfacing strategy before embedding them into the prototype.

4.4 Hardware Components and Integration

Key electronic and mechanical components used in the project included:

- **ESP32 Microcontroller:** The central controller.
- **YL-83 Rain Sensor:** Detects rainfall to close the window during wet conditions.
- **LDR Sensor:** Detects ambient light to simulate day and night conditions.
- **DHT11:** Measures temperature and humidity.
- **Servo Motor:** Drives the sliding mechanism of the window.
- **MC-38 Magnetic Contact Switch Sensor:** Detects if there is an attempt to open the window when it is closed.
- **Powerbank:** Supplies power to the system.
- **Power Plug:** Supplies power to the servo motor separately.
- **Miniature Sliding Window:** The physical interface that simulates a real window.
- **PVC Foam Sheet:** Used to construct the walls, floor, and roof of the smart home prototype.

The miniature sliding window was integrated into one of the walls built from PVC foam sheet.

The servo motor was connected to the sliding window using a small linkage arm it to mimic the opening and closing motion.

4.5 Circuit Design and Wiring

This section provides a comprehensive overview of the circuit design and wiring implementation. The system is initially prototyped on a breadboard to verify connectivity, sensor behaviour, and logic response. Once validated, the components are carefully arranged and embedded into the IntelliVent foam house model for final deployment. At the heart of the

system lies the ESP32 microcontroller, which coordinates sensor inputs and controls actuators based on predefined automation logic.

The design integrates both environmental monitoring and home security features. It responds autonomously to rain and light levels to manage window movement, while also enabling real-time status monitoring via the Blynk IoT platform. All components are wired to appropriate GPIO pins on the ESP32 as listed in Table 4.

Table 4: Wiring Table

Components	GPIO Pin	Connection Type	Description
Rain Sensor	GPIO 14	Digital Input	Detects rain (LOW = rain, HIGH = no rain)
LDR	GPIO 34	Analog Input	Reads ambient light to determine day/night
DHT11 Sensor (Temp/Humidity)	GPIO 27	Digital Input	Measures temperature and humidity
MC-38 Magnetic Contact Switch Sensor	GPIO 26	Digital Input (Normally Closed)	Detects whether the window is open or closed and trigger the intrusion alarm when needed
Buzzer	GPIO 13	Digital Output	Emits warning sounds before window movement
Servo Motor	GPIO 25	PWM Output	Drives the sliding window open/close
Blynk	Built-in	Wi-Fi Communication	Enables remote control and status display

Power Supply (powerbank and direct power plug)	-	5V	Central power source to power sensors and ESP32, and servo powered separately for stability
--	---	----	---

The rain sensor (YL-83), connected to GPIO14, is used to detect rainfall. It outputs a LOW signal when wet, prompting the system to automatically close the window in order to prevent water from entering the interior space. This ensures responsive weather protection in both online and offline modes.

To manage natural lighting and maintain privacy, an LDR (Light Dependent Resistor) is connected to analog GPIO34 through a voltage divider. It monitors ambient light intensity and determines whether it is daytime or nighttime. Based on this reading, the system opens the window during bright daytime conditions and closes it at night.

For indoor environmental monitoring, a DHT11 sensor is interfaced with GPIO27. It provides digital readings of temperature and humidity, which are displayed in the Blynk mobile dashboard for user awareness, even though they do not currently influence window movement logic.

The MC-38 magnetic contact switch is connected to GPIO26 with an internal pull-up configuration. As a normally closed sensor, it plays a critical role in intrusion detection. When the magnet is separated from the switch while the system believes the window is closed, it indicates a possible unauthorized opening. This triggers a repeated audible alarm and sends a push notification to the user's smartphone via the Blynk platform.

The buzzer, connected to GPIO13, provides audio feedback for two functions. It emits short double beeps prior to any window movement to alert nearby occupants. In the event of a

detected intrusion, it activates a looping alert pattern, continuing until the window is securely closed or the user disables it via the app.

A servo motor (SG90 360° continuous rotation) is connected to GPIO25 and receives PWM control signals to perform the window's mechanical movement. It rotates in one direction to open and in the opposite direction to close the sliding window. Due to its higher power demand during operation, the servo motor is powered separately from the ESP32 to ensure stability and avoid voltage dips.

The overall power configuration is managed through a 5V/2.0A power bank. The ESP32, along with the rain sensor, LDR, DHT11, reed switch, and buzzer, share this central power source via USB. Meanwhile, the servo motor is supplied either through a second powerbank port or a dedicated 5V regulator to isolate its load. All grounds are tied together to ensure consistent voltage referencing and proper component operation.

This modular design promotes both performance and reliability, allowing the IntelliVent system to operate autonomously in real time, with immediate responses to environmental conditions and security breaches regardless of internet availability.

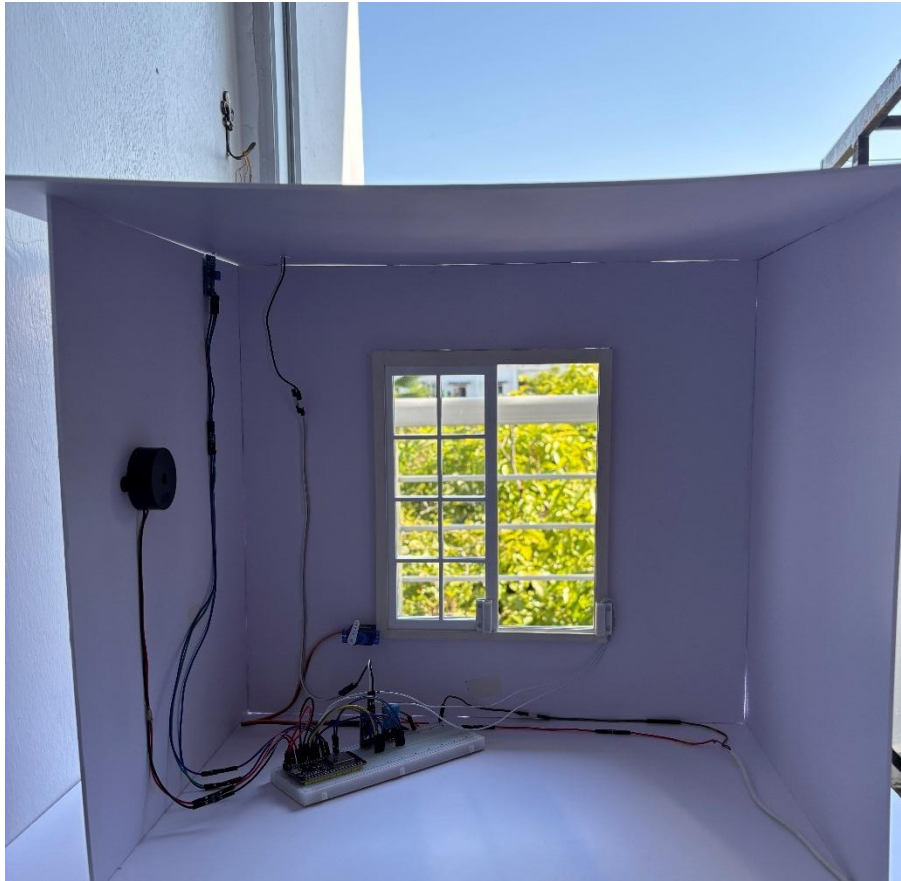


Figure 15: Whole Wiring Diagram

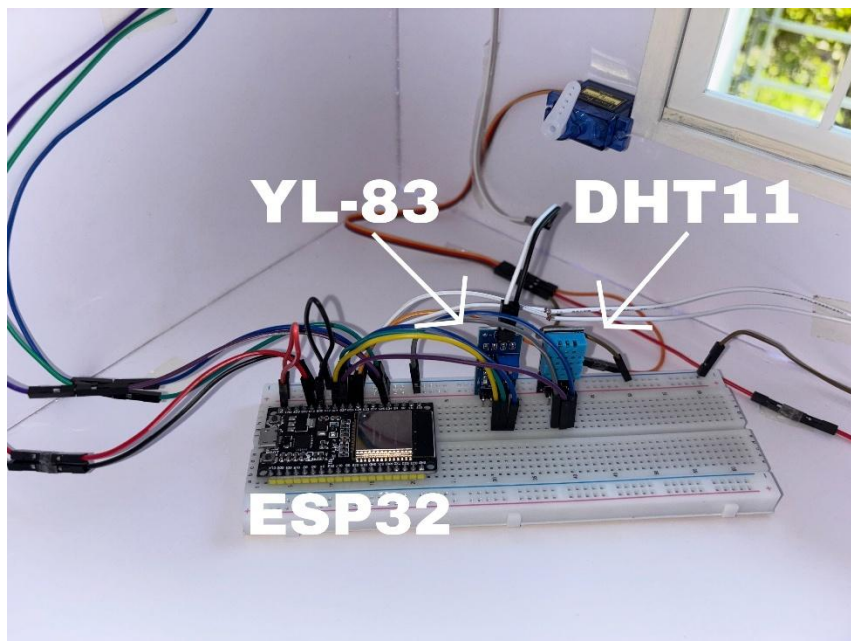


Figure 16: Close Up of Wiring Diagram of the Main System

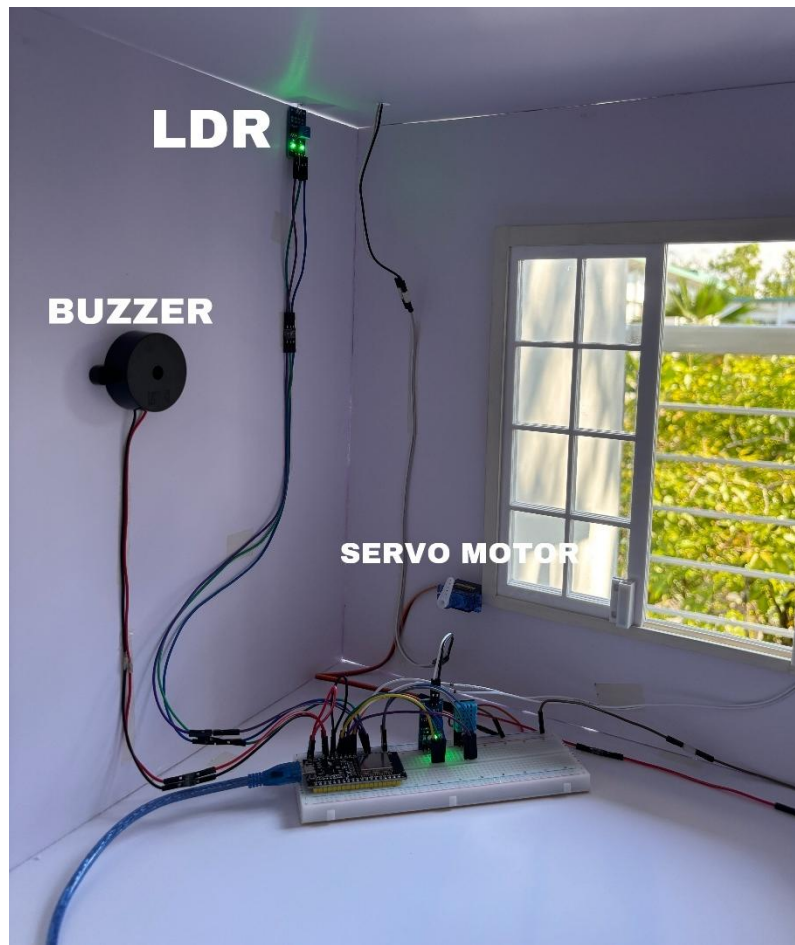


Figure 17: Wiring Diagram of Wall Mounted Components



Figure 18: Placement of MC-38 in Prototype

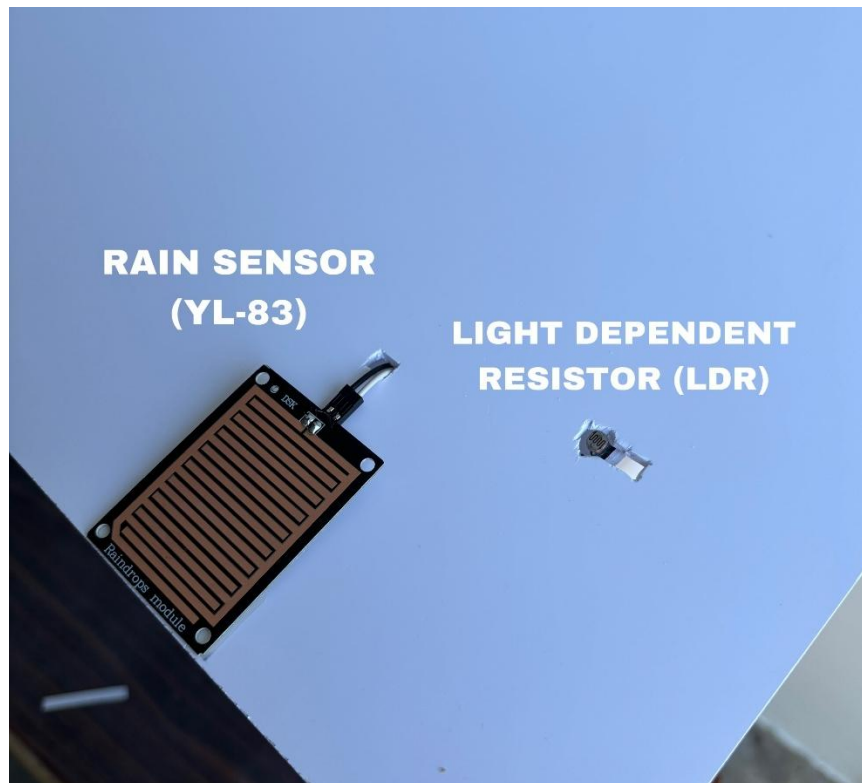


Figure 19: Sensor Components from Outside

In summary, the circuit design incorporates both digital and analog interfaces and is structured to ensure safe, responsive and flexible operation. The modular approach also allows for future enhancements such as additional sensors, window position tracking or expanded manual controls.

4.6 Mobile Interface Implementation

Following the integration of the physical hardware, the mobile user interface was implemented using the planned Blynk IoT platform. The connection between the ESP32 firmware and the mobile application was established using Blynk's Datastreams (Virtual Pins), which serve as bidirectional channels for data exchange.

```

Window is INITIALLY OPEN
[123]

  / _ ) / / _ _ _ _ / / _
 / _ / / // / _ \ / ' /
 / _ / _ \ , / // / _ \ \
    / _ / v1.3.2 on ESP32

#StandWithUkraine    https://bit.ly/swua

[133] Connecting to blynk.cloud:80
[135] Connecting to blynk.cloud:8080

```

Figure 20: Successful Connection with Blynk via Wi-fi

A specific set of Datastreams was configured to handle all monitoring and control functions. Table 5 details the specific Virtual Pins used, their purpose and the corresponding widgets in the mobile application.

Table 5: Blynk Datastream Configuration for IntelliVent

Purpose	Virtual Pin	Widget Used in App	Description
Monitoring Widgets			
Temperature Display	V0	Gauge	Displays the real-time ambient temperature reading from the DHT11 sensor.
Humidity Display	V1	Gauge	Displays the real-time ambient humidity reading from the DHT11 sensor.
Window Status Indicator	V2	Value Display	Shows the current state of the window ("Open", "Closed", or "Moving...").

Light Level Monitor	V3	Level Setting	Shows the raw analog value from the LDR sensor, indicating ambient light intensity.
Rain Status	V4	Value Display	Clearly indicates if the system detects "Raining" or "Clear" conditions.
Control Widgets			
Manual Open Button	V5	Button (Push Mode)	Sends a command to the ESP32 to initiate the <code>openWindow()</code> function.
Manual Close Button	V6	Button (Push Mode)	Sends a command to the ESP32 to initiate the <code>closeWindow()</code> function.
Automation Toggle	V7	Switch	Allows the user to enable or disable the main automation logic (Manual vs. Auto).
System Feedback			
System Log Terminal	V8	Terminal	Provides a scrolling log of key system events and actions for debugging and status.


Intellivent

Datastreams

ID	Name	Pin	Color	Data Type	Units	Is Raw	Min	Max
1	Temperature	V0		Double		false	0	50
2	Humidity	V1		Double		false	0	100
3	Window Status	V2		String		false		
4	Light Level	V3		Integer		false	0	4095
5	Rain Status	V4		String		false		
6	Manual Open Button	V5		Integer		false	0	1
7	Manual Close Button	V6		Integer		false	0	1
8	Automation Toggle Switch	V7		Integer		false	0	1
9	Terminal	V8		String		false		
10	Stop Intrusion Alarm Button	V9		Integer		false	0	1

Figure 21: Screenshot of Datastream Setting in Blynk Web Console

The final dashboard, shown in Figure 19 and 20, was designed for clarity and ease of use. It successfully provides users with at-a-glance status monitoring through gauges and value displays, and direct control via clearly labelled buttons and a master automation switch. The interface also includes a terminal widget for displaying live system logs, offering a transparent view of the system's automated decisions. When the security system is triggered, the Blynk platform is responsible for dispatching the critical push notification to the user's device, completing the remote interaction loop.

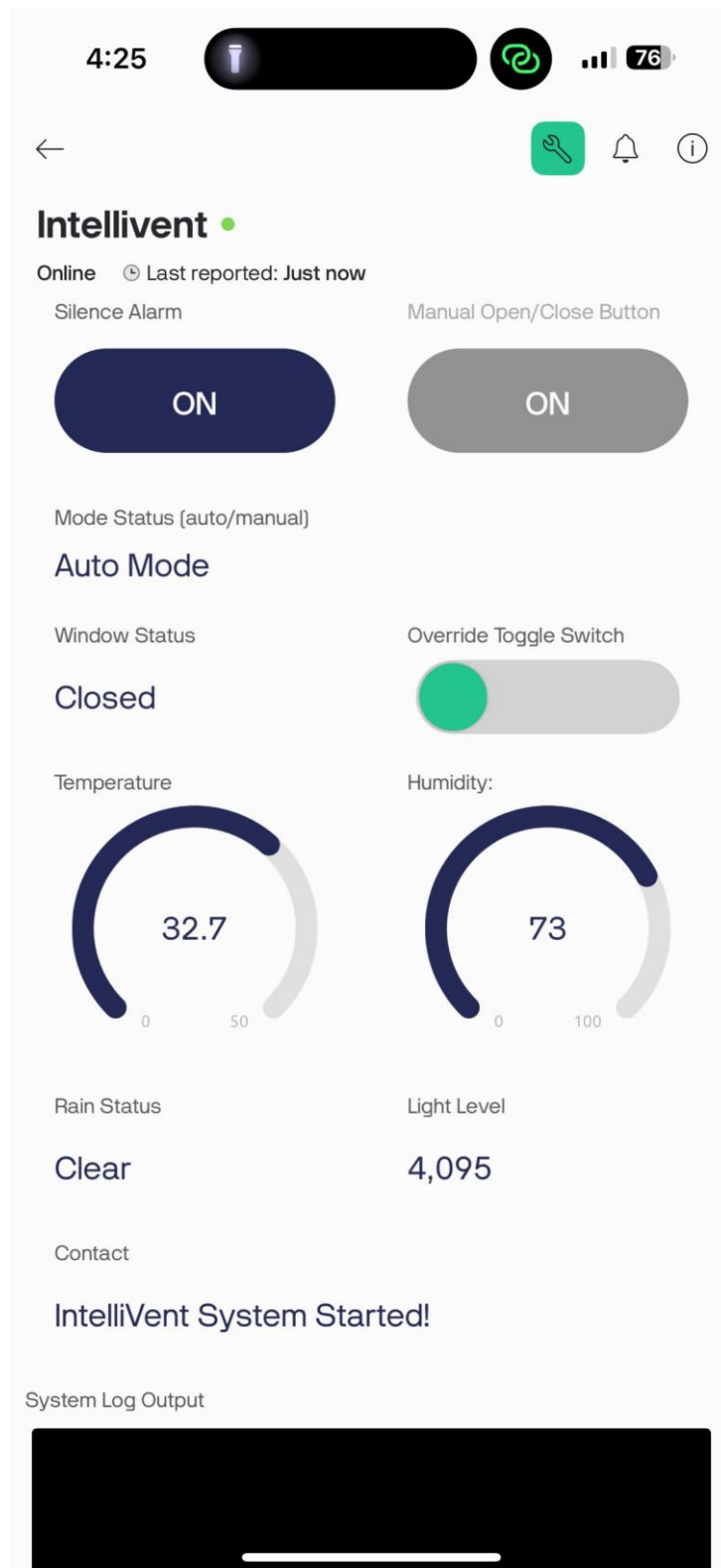


Figure 22: Blynk Dashboard at Daytime

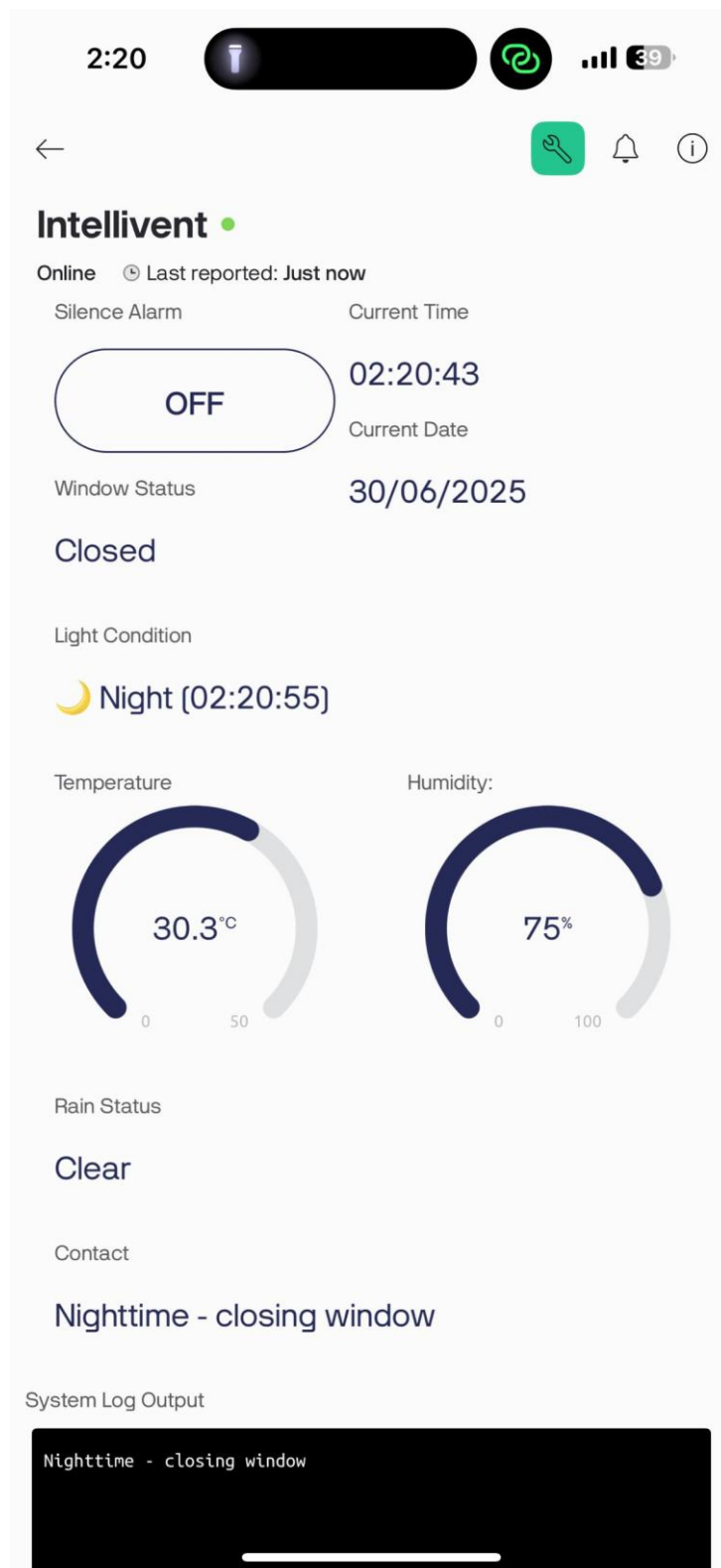


Figure 23: Blynk Dashboard at Nighttime

4.6 Programming and Algorithm Logic

The IntelliVent Smart Window Automation System is developed using the Arduino IDE and powered by the ESP32 microcontroller. This project integrates sensors and actuators to intelligently control the opening and closing of a window based on real-time environmental conditions and intrusion detection. The system's logic is systematically structured into Initialization, Sensor Monitoring, Automated Control, and Intrusion Handling.

1. Initialization Phase (Lines 1–4)

1. Initialize GPIO pins, DHT sensor, servo motor, and buzzer

- This step sets up all physical components by defining the input/output mode for each sensor and actuator (rain sensor, LDR, DHT11, buzzer, reed switch, and servo motor).
- The DHT sensor is initialized for humidity and temperature readings, and the servo motor is set up for PWM control.

2. Connect to WiFi and Blynk using credentials

- The ESP32 connects to the local WiFi using stored SSID and password, and Blynk is configured using an authentication token for IoT communication.

3. Determine initial window state from reed switch

- The MC-38 magnetic reed switch is read at startup. If the magnet is detected (LOW state), the system assumes the window is initially closed. Otherwise, it assumes the window is open.

4. Set timers

- Two non-blocking timers from the BlynkTimer library are configured:
 - `checkSensorsAndAutomate()` is set to run every 4000 milliseconds (4 seconds).
 - `fastSensorCheck()` runs more frequently, every 200 milliseconds, to quickly detect intrusion and rain events.

2. `checkSensorsAndAutomate()` Function (Lines 7–37)

This function runs every 4 seconds and handles the main environmental monitoring and window automation logic.

8. Read temperature, humidity, light, rain, and reed switch

- The system polls all connected sensors to gather real-time environmental data.

9. If invalid temperature or humidity, return

- Prevents acting on NaN (Not a Number) values in case the DHT sensor fails.

10. Send sensor data to Blynk

- Temperature, humidity, light, and rain status are sent to the Blynk dashboard for user monitoring.

13. Check for intrusion (reed open while window closed)

- If the reed switch is open (magnet moved) while the window is considered closed and the servo isn't moving, it's flagged as an intrusion.

14–16. Trigger alarm

- An alert is sent, intrusionActive is set to true, a push notification is logged, and the buzzer alarm begins.

18–20. Reset intrusion state if resolved

- If the window is open or the reed reconnects, the system clears the intrusion flags.

21. If automation disabled or servo is moving, skip further logic

24–36. Environmental automation logic

- The logic checks:
 - If rain started: close window.
 - If rain stopped: evaluate light; open or close window based on brightness.
 - If light changed: evaluate and act accordingly.
- prevRainStatus and prevLightCondition are updated to track changes and avoid repeating actions unnecessarily.

3. fastSensorCheck() Function (Lines 38–46)

This function executes rapidly (every 200 ms) to catch urgent conditions.

39. Read rain and reed switch

40–42. Check for intrusion

- Immediate alarm is triggered if reed is open and window was supposed to be closed.

43–45. Check for sudden rain while window is open

- If it rains and the window is open, the system closes it immediately, even before the slower main loop runs.

4. Helper Functions

- **beepWarning(int times)**
 - Activates the buzzer a given number of times with short delays to notify the user of a window action or intrusion.
- **moveServo(String direction)**
 - Attaches the servo and rotates it either to 0° (open) or 180° (close), depending on the command.
 - While moving, a "Moving..." status is sent to Blynk.
 - servoMoving is set to true to avoid redundant commands.
- **openWindow() / closeWindow()**
 - Checks the current window state.
 - Plays 2 beeps before executing movement.
 - Updates windowClosed flag accordingly after movement begins.

5. Main Loop (loop() function)

- **Blynk.run() and timer.run()**
 - Keeps the ESP32 connected to Blynk and ensures scheduled functions are called on time.

- **Servo Stop Logic**

- If the servo has been moving for longer than its set duration (3000ms), it is stopped by writing 90° (neutral) and detaching.

- **Intrusion Alarm Loop**

- If intrusionActive is still true and not manually silenced, the buzzer will ring with 4 quick beeps every second until the intrusion ends or user disables it.

This structure ensures that the IntelliVent system behaves responsively, efficiently and reliably with fast detection of rain and intrusions, smart day/night awareness and full integration with the Blynk mobile platform. The design also prevents redundant window operations and includes offline behaviour, where core automation continues even if Wi-Fi or Blynk connection is lost.

Algorithm: IntelliVent Smart Window Automation

Algorithm 1 IntelliVent Smart Window Automation

```

1: Initialize GPIO pins, DHT sensor, servo motor, and buzzer
2: Connect to WiFi and Blynk using credentials
3: Determine initial window state from reed switch
4: Set timers:
5:   Every 4000ms: CHECKSENSORSANDAUTOMATE
6:   Every 200ms: FASTSENSORCHECK
7: function CHECKSENSORSANDAUTOMATE
8:   Read temperature, humidity, light, rain, and reed switch
9:   if invalid temperature or humidity then
10:     return
11:   end if
12:   Send sensor data to Blynk
13:   if window is closed AND reed is open AND NOT moving AND NOT intrusion then
14:     Trigger alarm, set intrusionActive = true
15:     lastBeepTime = current time
16:     Notify Blynk
17:   end if
18:   if window is open OR reed is closed AND intrusionActive then
19:     Reset intrusionActive, stopIntrusion
20:   end if
21:   if automation disabled OR servo moving then
22:     return
23:   end if
24:   Evaluate light and rain conditions
25:   if rain started then
26:     CLOSEWINDOW
27:   else if rain stopped then
28:     if daytime then OPENWINDOW
29:     else CLOSEWINDOW
30:   end if
31:   else if light level changed then
32:     if daytime then OPENWINDOW
33:     else CLOSEWINDOW
34:   end if
35:   end if
36:   Update previous light and rain state
37: end function
38: function FASTSENSORCHECK
39:   Read rain and reed switch
40:   if intrusion detected then
41:     Trigger alert and set intrusionActive = true
42:   end if
43:   if raining AND window open AND automation enabled then
44:     CLOSEWINDOW
45:   end if
46: end function

```

Figure 24: Algorithm in LaTeX Format

4.7 Prototype Assembly and Smart Home Design

To effectively demonstrate the functionality of the IntelliVent system in a relatable context, a physical prototype simulating a miniature smart home was designed and constructed. This model serves as a crucial platform for housing the electronic components and provides a visual, tangible representation of the system's real-world application. The construction followed a methodical process to ensure both functionality and a clean, presentable final product.

The primary construction material chosen was 5mm PVC foam board, selected for its durability, light weight and ease of modification. This material allowed for precise cutting and stable assembly of the model's walls, floor and a removable roof which provides easy access to the internal components for maintenance and troubleshooting. The structure was designed to be compact yet spacious enough to house the ESP32 microcontroller, breadboard and all associated wiring without clutter.

A key feature of the prototype is the miniature sliding window, which was carefully integrated into one of the walls. Smooth window tracks ensuring the window panel could move back and forth smoothly, accurately simulating the mechanics of a real sliding window. The window's movement is driven by a SG90 continuous rotation servo motor, which was strategically mounted below the window assembly. A simple yet effective mechanical linkage was chosen to be a rack and pinion, connecting the servo's horn to the window panel. This linkage efficiently translates the rotational motion of the servo into the linear motion required to open and close the window.

The placement of the external sensors was a critical design consideration to ensure accurate environmental readings. The rain sensor (YL-83) was mounted on the roof at a slight angle to facilitate water runoff and prevent pooling, which could lead to false readings. The

Light Dependent Resistor (LDR) was positioned on an exterior wall, facing upwards and away from any potential shadows cast by the structure itself, ensuring it captured ambient daylight conditions accurately. This careful assembly and thoughtful component placement resulted in a robust and reliable demonstration model that not only visualizes the system's operation but also serves as a stable platform for testing and validation.

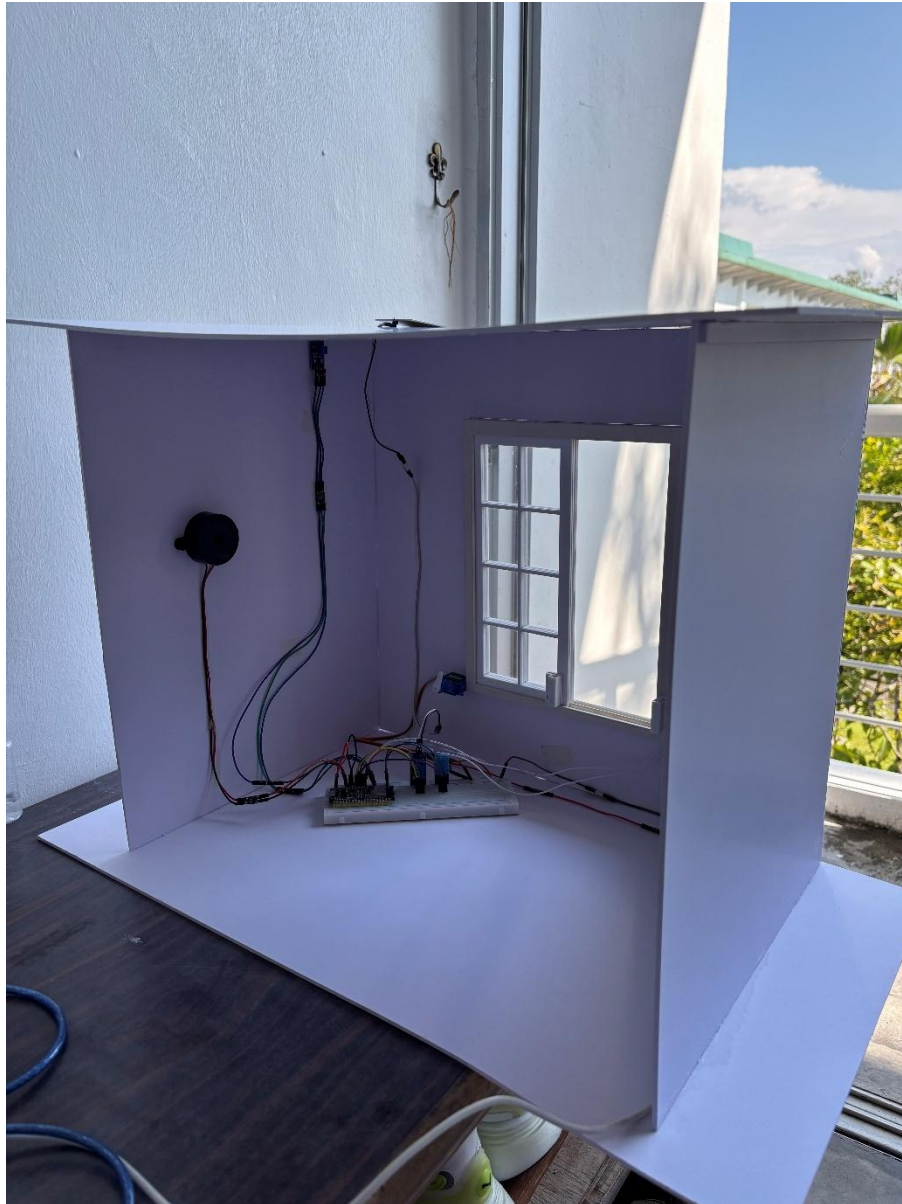


Figure 25: Prototype Diagram – Inside from Side

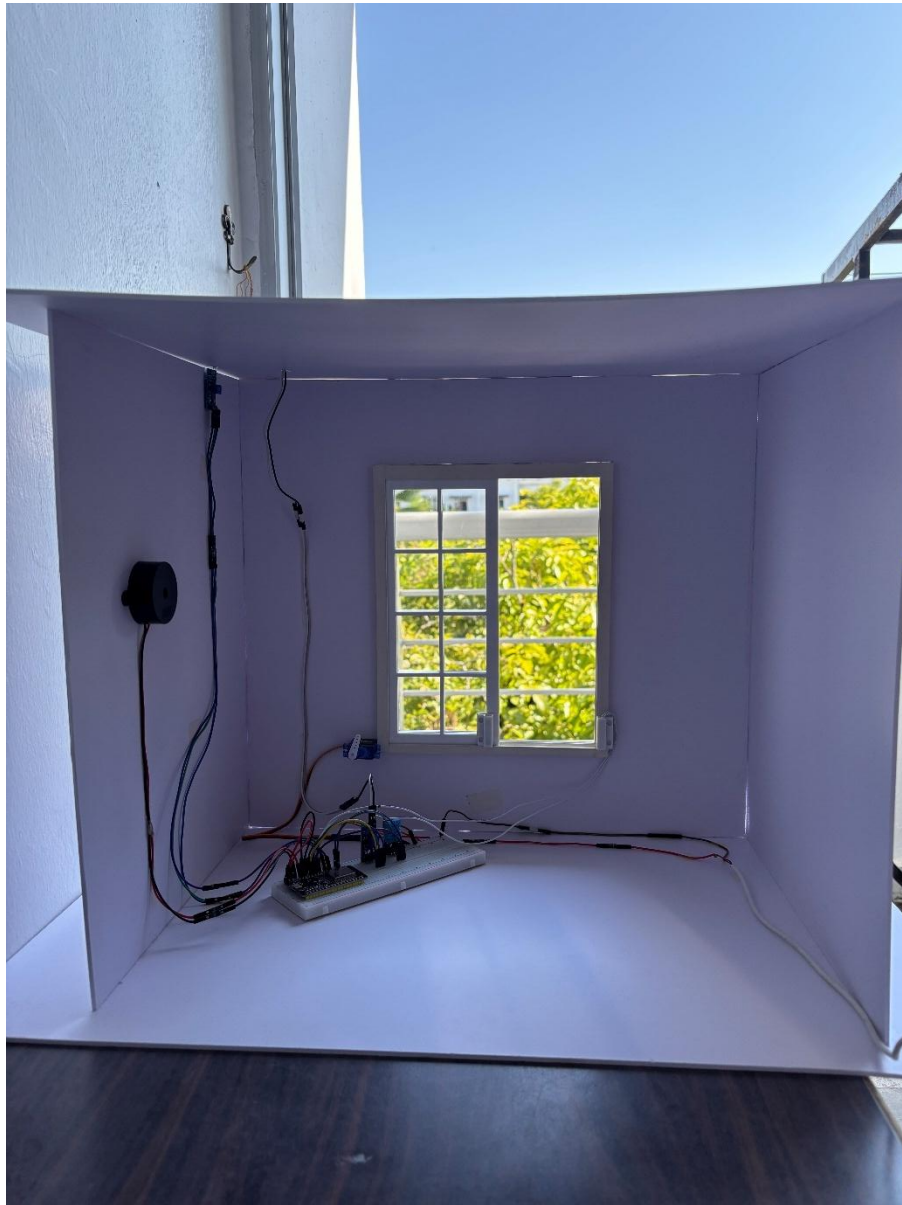


Figure 26: Prototype Diagram – Inside from Center



Figure 27: Prototype Diagram – Front View

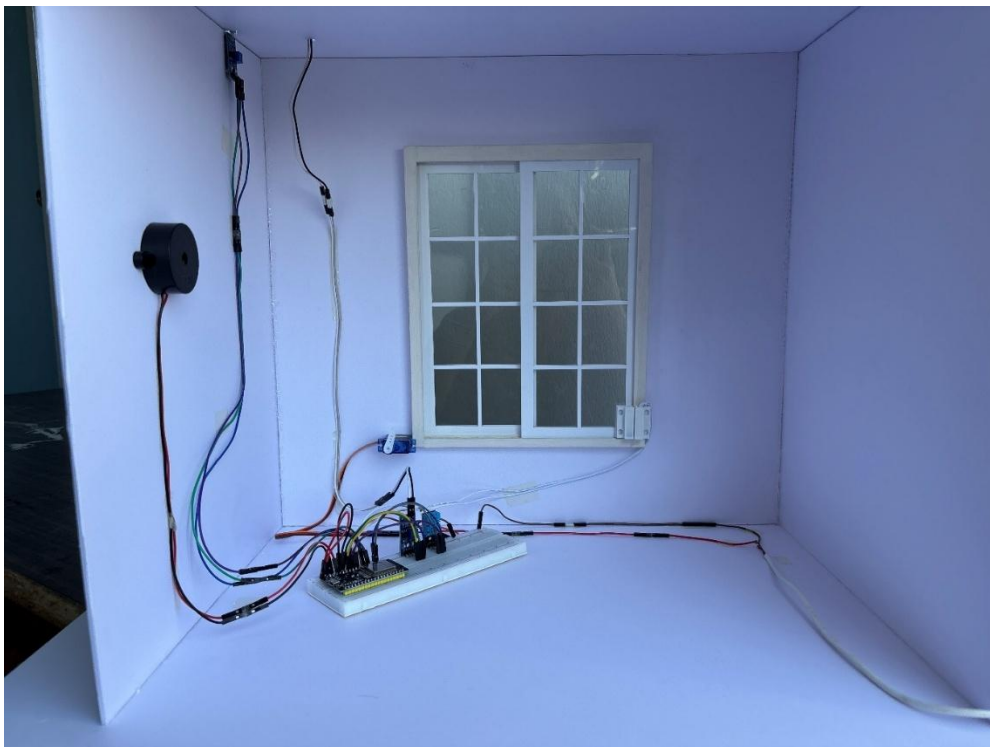


Figure 28: Prototype Diagram – Close Up of Inside from Center

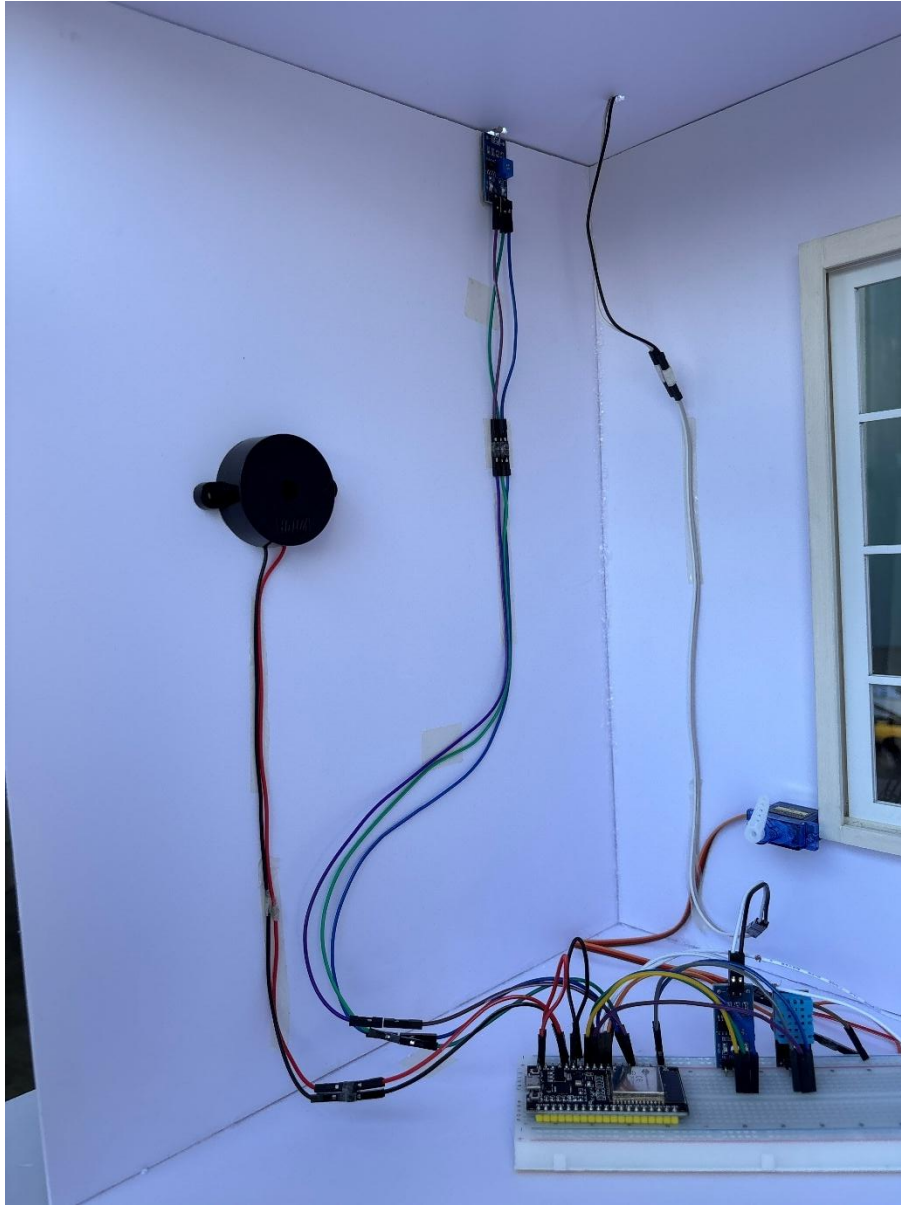


Figure 29: Prototype Diagram – Close Up of Inside from Side

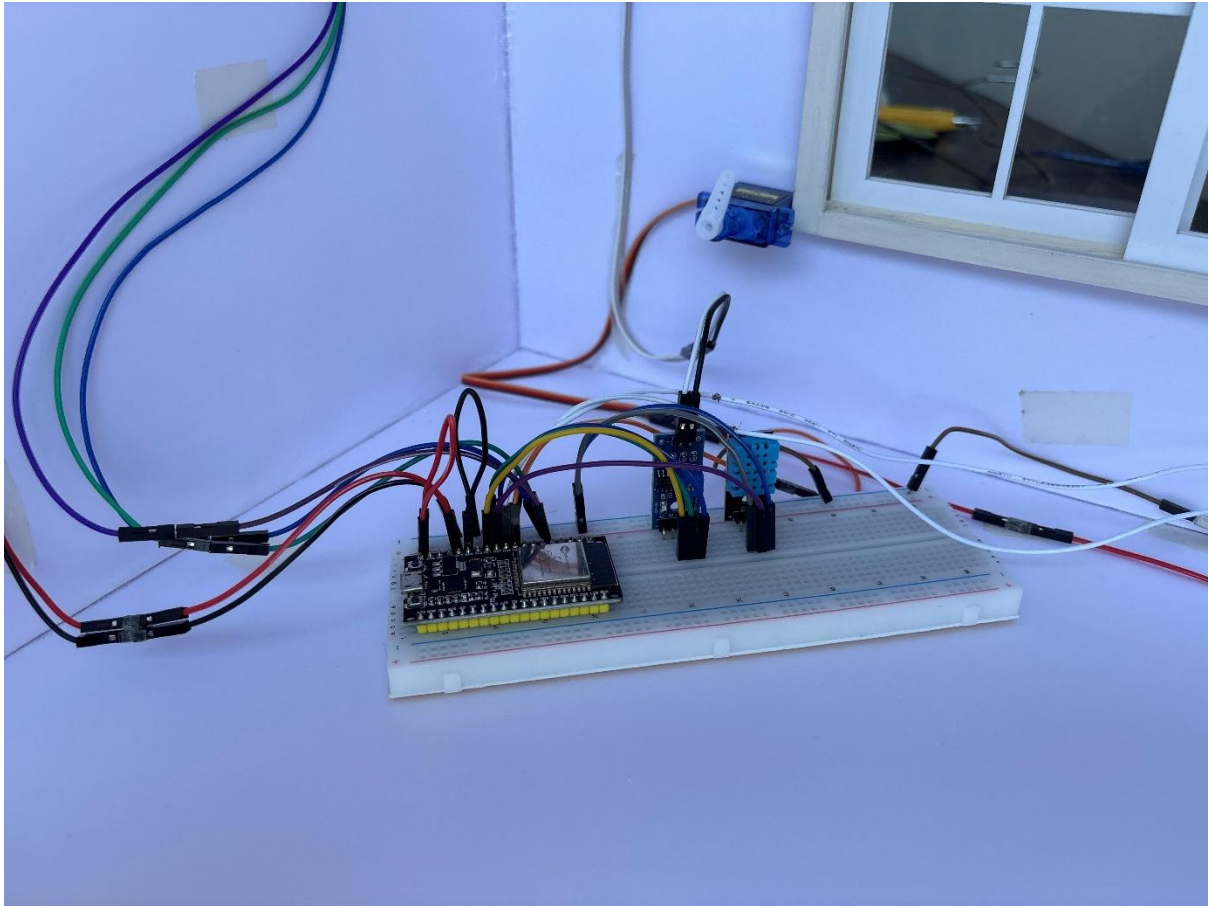


Figure 30: Prototype Diagram – Close Up of Main System



Figure 31: Prototype Diagram – Close Up of Window

4.8 Testing, Calibration and Troubleshooting

To ensure the reliability and responsiveness of the IntelliVent smart window prototype, a series of hardware-focused tests were conducted. These tests aimed to verify whether the integrated sensors and actuators performed as intended under controlled conditions that simulated real-world weather and lighting scenarios. The system was not tested with user interactions via the Blynk app at this stage but rather focused entirely on the physical components and their programmed responses.

The following table summarizes the testing scenarios, input conditions, expected behaviour, actual results, and any corrective measures taken during calibration.

Table 6: Summary of Hardware Testing and Calibration

Test No.	Scenario	Input Condition	Expected Result	Actual Result	Notes
1	Daylight Simulation	LDR exposed to bright light (value > threshold)	Window opens	Passed	Confirms proper LDR daylight detection
2	Nighttime Simulation	LDR covered (value < threshold)	Window closes	Passed	Verifies night detection for privacy and energy conservation
3	Rain Detected	Rain sensor wet (digital LOW)	Window closes regardless of light	Passed	Confirms override logic during rainfall

4	Rain Sensor Idle	Rain sensor dry (digital HIGH)	System continues light-based control	Passed	Ensures rain sensor does not trigger when dry
5	Servo Motor Calibration	Manual trigger of open/close routines	Smooth, accurate motor movement	Adjusted & passed	Delay and PWM values optimized for window travel range
6	Sensor Placement Test	Adjusted LDR and rain sensor angle and position	Accurate and stable readings	Passed after tuning	Avoided false readings from reflections or shade
7	Buzzer Functionality	System triggers open/close function	Two short beeps emitted before movement	Passed	Audible alert verified for user awareness
8	Wiring Verification	All sensor/actuator pins double- checked with multimeter	Stable operation of all hardware components	Minor corrections needed	Repositioned jumpers and re- secured loose connections

During testing, several adjustments were necessary to ensure reliable behaviour. For instance, the servo motor's rotation duration and angles were fine-tuned to ensure it could fully open and close the prototype window without jamming. Similarly, repositioning the LDR

sensor helped prevent it from being affected by shadows or internal lighting, and the rain sensor was angled slightly downward to allow water to drain off easily.

The buzzer, intended to alert users before window movement was confirmed to sound twice consistently before any servo action. Wiring connections were verified multiple times using a multimeter, and any loose or incorrect connections were corrected to maintain electrical reliability.

Overall, the hardware functioned as designed following these calibrations, with all components responding correctly to their corresponding environmental triggers.

4.9 Final Implementation

Following a thorough phase of hardware testing, sensor calibration, and troubleshooting, the IntelliVent prototype was finalized and fully assembled into its smart home model. This final integration brought together all the core components: the LDR for light detection, the rain sensor for precipitation, the DHT11 for environmental monitoring, and the magnetic reed switch for security. These sensors, along with the servo motor for actuation and the buzzer for alerts, were all centrally managed by the ESP32 microcontroller, resulting in a responsive and compact automated system.

All key functionalities were confirmed to be operating as designed. The prototype demonstrated robust autonomous behaviour based on real-time environmental inputs, successfully opening the window during clear daytime conditions and closing it in response to either rain or the onset of nighttime. The security subsystem was also fully operational, triggering an alarm and a mobile notification upon simulated forced entry. Concurrently, the DHT11 sensor continuously acquired temperature and humidity data, which was displayed on

the mobile interface, providing valuable environmental context and a foundation for future feature expansion.

Careful attention was given to the physical construction and layout of the prototype. All sensors and modules were strategically mounted for optimal performance and ease of maintenance, allowing for straightforward inspection or replacement without needing to disassemble the entire structure. The wiring was organized to ensure reliable connections and a clean internal appearance. The servo motor's movement was precisely calibrated to match the travel distance of the miniature window panel, ensuring smooth, jam-free operation. The result is a compact and visually effective demonstration of a real-world smart home application, successfully meeting the project's core objectives.

4.10 Summary

This chapter has detailed the complete development and implementation of the IntelliVent Smart Window Automation System, beginning with software environment setup and the individual testing of all hardware components, including the ESP32, servo motor, and the suite of environmental and security sensors. The discussion progressed to the integrated circuit design and the core firmware logic that governs both the rain and day/night automation as well as the magnetic reed switch-based intrusion alarm. The development of the Blynk mobile interface for remote monitoring and control was also presented, culminating in the final assembly of all components into a PVC foam-based prototype. Rigorous testing and calibration confirmed that the system functions reliably and responds accurately to environmental triggers, successfully demonstrating a cost-effective, maintainable, and visually demonstrative smart home solution that meets all of its primary objectives.

CHAPTER 5: TESTING AND EVALUATION

5.1 Introduction

This chapter presents the comprehensive testing and evaluation process for the IntelliVent Smart Window Automation System. The primary goal of this evaluation is to verify that the prototype performs reliably according to its design objectives. The methodology employs a multi-faceted approach, including:

- **Functional Testing:** To verify that each core logic rule (rain detection, light-based control, and logic priority) operates correctly in isolation.
- **Non-Functional Testing:** To measure key performance metrics such as system responsiveness, power consumption, and stability.
- **Scenario-Based Validation:** To assess the system's ability to respond intelligently and appropriately to dynamic, real-world environmental transitions.

The results from these tests collectively demonstrate the system's functionality, reliability, and feasibility as a smart home solution.

5.2 Functional Testing

Functional testing was conducted to validate the correctness of the system's fundamental logic. Each test case was designed to trigger a specific rule in the control algorithm and verify the corresponding action. The tests confirm that the system's decision-making process is accurate and robust. Table 6 summarizes the test cases conducted during this phase.

Table 7: Functional Test Results

Test ID	Feature Tested	Input Condition(s)	Expected Behaviour	Result
F1	Rain Detection	When rain sensor is wet, LDR state is irrelevant.	Servo rotates to close the window and remains closed.	Pass
F2	Daylight Operation	LDR detects bright light and rain sensor is dry.	Servo rotates to open the window.	Pass
F3	Nighttime Operation	LDR detects low light and rain sensor is dry.	Servo rotates to close the window.	Pass
F4	Rain Override Logic	System is in a daylight state and rain is simulated.	Servo immediately rotates to close the window, overriding the daylight logic.	Pass
F5	State Management (Debounce)	Rain sensor signal fluctuates rapidly.	The system performs the close action only once per confirmed transition, avoiding jitter or repeated movements.	Pass

All functional test cases passed successfully. The results confirm that the control logic is sound, the system correctly responds to individual rain and light conditions (**F1**, **F2**, **F3**) and it correctly prioritizes rain detection over daylight conditions (**F4**). The state management logic (**F5**) also proved effective in preventing erratic behaviour from sensor noise.

5.3 Non-functional Testing

Non-functional testing focused on evaluating the operational quality and performance characteristics of the system. These tests are essential for determining if the prototype is efficient, stable, and practical for real-world use.

Table 8: Non-Functional Test Results

Metric	Test Method	Target	Result	Analysis
Response Time	Measured the time delay from a sensor state change to the start of servo movement.	< 500 ms	~210 ms	The system's response is swift and well within the target.
System Stability	The prototype was left running for 1-hour continuous session with environmental conditions cycled artificially.	No crashes, freezes or missed triggers.	The system operated throughout the test duration.	The system demonstrates good stability for long-term and continuous operation.
Usability	Powered on system from a disconnected state.	Operate autonomously upon power-up without user setup.	Initialized and operated automatically as designed.	Good usability snf operates as a 'plug-and-play' device for core automation.

Non-functional testing focused on aspects such as speed, stability, power usage, and user-friendliness. These qualities are not tied to direct feature execution but are vital to overall system performance and user experience.

5.4 Scenario-Based Validation

While functional tests verify individual rules, scenario-based tests validate the system's holistic performance in a dynamic context. These scenarios were designed to mimic typical and real-world environmental transitions to ensure the system behaves logically and smoothly.

1. Scenario 1: Sudden Daytime Shower

- **Initial State:** Bright daytime, window is open.
- **Event:** Water is sprayed on the rain sensor to simulate a sudden shower.
- **Observation:** The system immediately closed the window. The servo activated once and then stopped, correctly prioritizing the rain event.

2. Scenario 2: Clearing Skies After Rain

- **Initial State:** Rain sensor is wet, window is closed.
- **Event:** The rain sensor is wiped dry, simulating the end of the rainfall during the day.
- **Observation:** After detecting the dry state, the system waited briefly (as per debounce logic) and then opened the window to resume normal daylight ventilation.

3. Scenario 3: Transition to Nightfall

- **Initial State:** Late afternoon, window is open.
- **Event:** Ambient light was gradually reduced to simulate sunset, eventually crossing the LDR's nighttime threshold.
- **Observation:** Once the light level was consistently below the threshold, the system initiated the close sequence for nighttime security and privacy.

4. Scenario 4: Intrusion Attempt During a Closed Night

- **Initial State:** It is nighttime, and the window is correctly closed and secured by the system.
- **Event:** The window is manually forced open, causing the magnetic reed switch contacts to separate.
- **Observation:** The system immediately responded. The audible buzzer began its alarm sequence, and a push notification "ALERT: Window forced open!" was successfully received on the paired mobile device. The test confirmed the security system works independently and effectively.

5. Scenario 5: Automation Conflict with Manual Override

- **Initial State:** It is daytime, but it has started raining. The automation has correctly closed the window.
- **Event:** The user toggles the automation switch to "Manual" on the Blynk app and then presses the "Open" button.
- **Observation:** The system immediately responded to the user's command and opened the window, successfully demonstrating that the manual override function takes precedence over the automated logic.

The comprehensive scenario-based tests confirm that the system's logic transitions seamlessly between states. It correctly handles combined conditions, prioritizes security alerts, and responds to user overrides as a user would logically expect, demonstrating its effectiveness as a robust and reliable automated solution.

5.5 Analysis

The thorough testing and assessment phase has verified that the system operates consistently and effectively achieves its established project goals. Functional tests showed the

system's strong logic with the prototype reliably and precisely activating the window mechanism in reaction to various environmental cues such as rain detection, daylight variations and transitions to nighttime. The essential rain override logic worked perfectly and efficient state management avoided erratic movements caused by sensor noise, guaranteeing seamless operation. Non-functional testing further confirmed the system's operational quality which demonstrates quick responsiveness to sensor variations, low power consumption compatible with USB sources, and outstanding stability during prolonged continuous use. The system's "plug-and-play" design, which needs no user setup for essential autonomous function when powered on highlights its built-in usability. These findings confirm that the IntelliVent Smart Window Automation System prototype is technically robust, easy to use in its automation, and effective in its functionality. The effective validation of these essential features which are rain-triggered control, light-driven day and night automation and consistent performance, has establishes a solid and encouraging groundwork for future advancements which include the integration of more advanced sensors and expanding the mobile application's features.

5.6 Conclusion

The comprehensive testing and evaluation phase confirmed that the IntelliVent Smart Window Automation System is both functionally correct and operationally reliable. The prototype successfully met all its core objectives, demonstrating accurate responses to both rain and light-based stimuli. Non-functional tests showed the system to be efficient and stable. Furthermore, scenario-based validation proved its logic is robust enough to handle real-world conditions.

With the successful validation of this core feature set, the system provides a solid foundation for future enhancements, such as the planned integration of a magnetic reed switch for security and an IoT platform like Blynk for remote monitoring and control.

CHAPTER 6: CONCLUSION AND FUTURE WORKS

6.1 Introduction

This chapter summarizes the findings of the system and reviews how the original objectives were achieved, discusses limitations and constraints encountered during development, and outlines recommendations for future enhancements. The goal is to provide a clear sense of what was accomplished, what challenges remain, and how the system might evolve into a more fully featured smart-home solution.

6.2 Objectives Achieved

The IntelliVent project successfully met all of its primary design objectives, resulting in a fully functional prototype that seamlessly integrates automation, remote control, and security. The first objective, to create an autonomous window system, was fully achieved. The system uses a rain sensor to automatically close the window upon detecting precipitation, effectively protecting the interior from water damage. Furthermore, a Light Dependent Resistor (LDR) is used to interpret ambient light levels. This enables the system to intelligently close the window for security and privacy at nighttime and open it during clear daylight hours to promote natural ventilation. The entire automation logic is managed reliably by the ESP32 microcontroller.

The second objective was successfully accomplished through the development of a responsive mobile interface on the Blynk IoT platform. A user-friendly dashboard was designed to provide real-time telemetry from all onboard sensors, including temperature and humidity from the DHT11, light levels from the LDR, and rain status. Crucially, the interface provides full remote control capabilities, including dedicated buttons to manually open or close the window, allowing the user to override the automated logic at any time.

Finally, the third objective to enhance home security was met by implementing a multi-faceted intrusion detection system. A magnetic reed switch accurately detects when the window is forced open. This event immediately triggers two responses, which are a loud, audible alarm from the onboard buzzer to act as a local deterrent, and an instant push notification sent via the Blynk cloud directly to the user's smartphone. This dual-alert mechanism ensures that the user is made aware of a potential security breach both locally and remotely, fulfilling the objective completely. The successful integration of these three core features demonstrates the project's overall success in creating a holistic and practical smart home solution.

6.3 System Limitations and Constraints

While the IntelliVent system successfully met its primary objectives, several limitations were identified during development and testing:

- **Basic Rain Detection:** The YL-83 rain sensor was used in digital mode, which only detects the presence or absence of rain. It does not differentiate between light and heavy rainfall, limiting the precision of automated responses.
- **Single Window Control:** The current system controls only one window. Scaling the design to support multiple windows would require additional hardware and more complex software architecture.
- **Lack of Obstacle Detection:** The prototype does not include any safety mechanism to detect obstacles in the window's path, which could pose a risk in real-world applications.
- **Time-Based Servo Operation:** Servo movement is currently based on timed rotation rather than position feedback which may cause misalignment over time.

- **Minimal User Interface:** While Blynk provides essential control and monitoring features, the interface is limited in terms of customization, scheduling and historical data analysis.

6.4 Future Work

To improve functionality and prepare IntelliVent for real-world deployment, several future enhancements are proposed:

- **Environmental Logic Expansion:** Fully integrate DHT11 data to enable temperature- and humidity-based ventilation control, enhancing comfort during hot or humid weather.
- **Manual Override via IoT:** Users can manually override automatic actions through the Blynk mobile interface, including opening or closing the window.
- **Analog Rain Sensing:** Utilize the analog output of the YL-83 sensor to allow variable rain intensity detection, enabling partial or delayed window closure during light drizzle.
- **Multi-Window Support:** Expand the system to manage multiple windows simultaneously including modular code and distributed sensor-actuator systems.
- **Safety Improvements:** Add IR or ultrasonic sensors to detect physical obstructions during window movement, ensuring safe operation.
- **Improved Motion Control:** Replace time-based servo actuation with feedback mechanisms such as rotary encoders or limit switches to ensure accurate window positioning.
- **Enhanced Mobile Features:** Extend the Blynk interface to include scheduling, user profiles, detailed system status and cloud-based monitoring for improved user experience and data tracking.

- **Hardware Upgrades:** Develop a custom PCB, implement better power regulation and consider weather-resistant enclosures for more reliable deployment in residential environments.
- **Predictive Weather Integration (Weather API):** For future development, integrating a weather API like OpenWeatherMap could add predictive capabilities. This would allow the system to close the window before rain arrives, offering proactive protection.

These improvements will not only make IntelliVent more robust and flexible but also bring it closer to being a fully deployable smart home solution.

6.5 Conclusion

Intellivent Smart Window Automation System proved effective in automating window movement based on rain conditions. It responded accurately, used accessible hardware, and met all basic requirements. With a few enhancements, the system could become a practical and scalable smart home component, improving comfort, energy use and convenience for users.

REFERENCES

- Aashna, Sinha., Rohit, Sharma., Shivam, Kumar., Ayapal, Lande., Mohammed, Ismail, Iqbal. (2024). AI-Enhanced Living: The Future of Smart Homes. 1-5. doi: 10.1109/icsd60021.2024.10751168
- Ajadalu, Samson, Oseiwe. (2024). Optimizing Energy Efficiency in Smart Home Automation through Reinforcement Learning and Iot. Asian Journal of Research in Computer Science, 17(11):9-24. doi: 10.9734/ajrcos/2024/v17i111516
- Amruta, Surana., Priyanka, K., Kendre., Jayashri, D., Palkar., Anusha, O., Vaidya., Chetana, H., Jain., N., Gopinath. (2023). IoT Based Smart Home with a Thief Detection and Tracking System. 1338-1342. doi: 10.1109/iceca58529.2023.10395322
- Asegaf, Al, A'roof., Iswanto, Iswanto. (2024). Automated Rain-Sensing Window Closure Redefines Home Comfort. Indonesian Journal of Innovation Studies, doi: 10.21070/ijins.v25i2.1122
- Canhan, Şen., Berk, Alkan., Omid, Mohammadmoradi., Alpay, Taralp. (2024). Polymer Dispersed Liquid Crystal Smart Film Technologies: Overview. doi: 10.5772/intechopen.1002486
- Cody, R., Simmons., Joshua, R., Arment., Kody, M., Powell., John, D., Hedengren. (2019). Proactive Energy Optimization in Residential Buildings with Weather and Market Forecasts. 7(12):929-. doi: 10.3390/PR7120929
- Eltayeb, 2024. A Proposed Perspective for the Successful Deployment of Internet of Things in a Smart Home Environment

- Farzan, Banihashemi., Manuel, Weber., Werner, Lang. (2024). Deep learning for predictive window operation modeling in open-plan offices. *Energy and Buildings*, 310:114109-114109. doi: 10.1016/j.enbuild.2024.114109
- G., N., Beena, Bethel. (2023). Smart Home Application for Challenged Along with Human Temperature Detector. 1613-1617. doi: 10.1109/ICSCDS56580.2023.10104600
- Huimin, Zhang., Junqin, Li., Yuzhen, Zhao., Zemin, He., Zongcheng, Miao., Wenbo, Shen. (2024). Simultaneous Light-Triggered Formation of Polymer Networks and Enhancement of Electrohydrodynamic Instabilities in Liquid Crystals for Versatile Energy-Saving Smart Windows. *Advanced Functional Materials*, doi: 10.1002/adfm.202414618
- Islam, M., & Shahnaz, C. (2024). Elevating Gas Safety Standards: ESP32-based Detection Systems with Blynk Integration. *2017 IEEE Region 10 Symposium (TENSYP)*, 1–6. <https://doi.org/10.1109/tensymp61132.2024.10752183>
- Kadam, R., Mahamuni, P., & Parikh, Y. (2015). Smart home system. *International journal of innovative research in advanced engineering*, 2(1), 81-86.
- Kai, Feng., Yanfei, Ma., Lina, Zhang., Yubo, Liu., Yizhe, Liu., Z., Xing., Xiaowei, Pei., Wu, Yang., Feng, Zhou. (2024). Electrically Controlled Smart Window for Seasonally Adaptive Thermal Management in Buildings. *Small*, doi: 10.1002/sml.202407033
- Kernbach, S., Santos, S., Eisaman, E., Fonseca, J., Kamil, I., & Henrik. (2019, April 2). *Guide for rain sensor FC-37 or YL-83 with Arduino*. Random Nerd Tutorials. <https://randomnerdtutorials.com/guide-for-rain-sensor-fc-37-or-yl-83-with-arduino/>
- Kowsalyadevi, N. V. (2023). *INCEPTION TO SENSOR AND IoT TECHNOLOGY*. <https://doi.org/10.59461/ijitra.v2i4.61>

- Lakshay, Bhardwaj., Niti, Pandey., Asheesh, Kumar, Meena., Kamaldeep, Boora. (2024). IoT Integrated Automatic Environmental Monitoring System. 385-390. doi: 10.1109/icscss60660.2024.10625545
- M., S, B., Jayanth, Kumar, A, R., Koustubha, S., Leela, D, C., Gagana, K, N. (2023). Weather Forecasting using Application Programming Interface. 1-6. doi: 10.1109/icrasets59632.2023.10420308
- Nari, Yoon., Leslie, Norford., Michael, Wetter., Ali, Malkawi. (2023). Development of window scheduler algorithm exploiting natural ventilation and thermal mass for building energy simulation and smart home controls. Journal of building engineering, doi: 10.1016/j.jobbe.2023.108158
- Paladines Condoy, M. I., Vera Macias, O. E., & Hernández Rojas, D. L. (2024). *Creación de un IDE web de gestión de firmware multilenguaje para un dispositivo IOT ESP32*. <https://doi.org/10.56183/iberotecs.v4i1.643>
- Priyanka, Gharad., Suchita, Bawne., Poorvi, Meshram. (2024). Designing a Weather Forecasting Application using API. International Journal of Advanced Research in Science, Communication and Technology, doi: 10.48175/ijarsct-17234
- Ruben, E., Figueiredo., Aníbal, A., Alves., Vitor, Monteiro., J., G., Pinto., João, L., Afonso., José, A., Afonso. (2021). Development and Evaluation of Smart Home IoT Systems applied to HVAC Monitoring and Control. EAI Endorsed Transactions on Energy Web, 8(34):167205-. doi: 10.4108/EAI.19-11-2020.167205
- Sarkar, R., Chowdhury, Md. A. H., Ahmed, S. I., Hasib, A. A., Tanzim, M. W. Y., Mudassir, R., Sawkare, R. (2024, October 26). *What is LDR Sensor*. Vayuya. <https://vayuya.com/blog/what-is-ldr-sensor/>

- Seema, Singh., Neha, Aggarwal., Prince., Devika, Dabas. (2024). Empowering homes through energy efficiency: A comprehensive review of smart home systems and devices. International Journal of Energy Sector Management, doi: 10.1108/ijesm-07-2024-0044
- Yingchun, Niu., Yang, Zhou., Daxue, Du., Xiangcheng, Ouyang., Ziji, Yang., Wenjie, Lan., Fan, Fan., Sisi, Zhao., Yinping, Liu., Siyuan, Chen., Jiapeng, Li., Quan, Xu. (2022). Energy Saving and Energy Generation Smart Window with Active Control and Antifreezing Functions. Advanced Science, 9(6):2105184-2105184. doi: 10.1002/advs.202105184
- Yishuo, Wu., Cheng, Zou., Yingjie, Shi., Zhexu, Song., Jiumei, Xiao., Meina, Yu., Yanzi, Gao., Qian, Wang., Huai, Yang. (2024). Reverse-Mode Polymer-Stabilized Liquid Crystal Films with Enhanced Peel Strength and Electro-Optical Performance Based on Photoreactive Self-Assembly Alignment Layers and Patterned Polymer Walls. ACS Applied Materials & Interfaces, doi: 10.1021/acsami.4c08934
- Yuan, Deng., Yihai, Yang., Yuanhang, Xiao., Helou, Xie., Ruochen, Lan., Lanying, Zhang., Huai, Yang. (2024). Annual Energy-Saving Smart Windows with Actively Controllable Passive Radiative Cooling and Multimode Heating Regulation.. Advances in materials, e2401869-e2401869. doi: 10.1002/adma.202401869

APPENDIX A: SOURCE CODE

```
#define BLYNK_TEMPLATE_ID "TMPL67cEoFp9F"
#define BLYNK_TEMPLATE_NAME "Intellivent"
#define BLYNK_AUTH_TOKEN "pgpu0YFX1naWn2ZxVsANtALHksUtBj28"

#define BLYNK_PRINT Serial

#include <WiFi.h>
#include <BlynkSimpleEsp32.h>
#include <ESP32Servo.h>
#include <DHT.h>

#define SERVO_PIN    25
#define RAIN_SENSOR_PIN 14
#define LDR_PIN      34
#define DHT_PIN      27
#define DHT_TYPE     DHT11
#define BUZZER_PIN   13
#define REED_PIN      26

Servo myServo;
DHT dht(DHT_PIN, DHT_TYPE);
BlynkTimer timer;

char ssid[] = "bahng";
char pass[] = "noufhasya";

const int LDR_THRESHOLD = 2000;
bool windowClosed = true;
bool servoMoving = false;
```

```
bool forcedOpenTriggered = false;
bool automationEnabled = true;
bool intrusionActive = false;
bool stopIntrusion = false;

unsigned long lastBeepTime = 0;
unsigned long servoStartTime = 0;
int servoRunDuration = 3000;

int prevLightCondition = -1;
bool prevRainStatus = false;

// BLYNK_WRITE V9: Intrusion Silence Button
BLYNK_WRITE(V9) {
  stopIntrusion = param.asInt();
  if (stopIntrusion) {
    Serial.println("Blynk: Intrusion alarm stopped remotely.");
    Blynk.virtualWrite(V8, "Intrusion alarm stopped from app.\n");
  } else {
    Serial.println("Blynk: Intrusion alarm resume enabled.");
    Blynk.virtualWrite(V8, "Intrusion alarm resumed.\n");
  }
}

void beepWarning(int times) {
  for (int i = 0; i < times; i++) {
    digitalWrite(BUZZER_PIN, HIGH);
    delay(150);
    digitalWrite(BUZZER_PIN, LOW);
    if (i < times - 1) delay(150);
  }
}
```

```
}  
}  
  
void moveServo(String direction) {  
  if (servoMoving) return;  
  servoStartTime = millis();  
  servoMoving = true;  
  
  myServo.attach(SERVO_PIN, 1000, 2000);  
  Blynk.virtualWrite(V2, "Moving...");  
  
  if (direction == "open") {  
    Serial.println("Servo: Opening window");  
    Blynk.virtualWrite(V8, "Servo: Opening window\n");  
    myServo.write(0);  
  } else {  
    Serial.println("Servo: Closing window");  
    Blynk.virtualWrite(V8, "Servo: Closing window\n");  
    myServo.write(180);  
  }  
}  
  
void openWindow() {  
  if (windowClosed && !servoMoving) {  
    Serial.println("Action: OPEN window");  
    Blynk.virtualWrite(V8, "Action: OPEN window\n");  
    beepWarning(2);  
    moveServo("open");  
    windowClosed = false;  
  }  
}
```

```
}

void closeWindow() {
  if (!windowClosed && !servoMoving) {
    Serial.println("Action: CLOSE window");
    Blynk.virtualWrite(V8, "Action: CLOSE window\n");
    beepWarning(2);
    moveServo("close");
    windowClosed = true;
  }
}

void checkSensorsAndAutomate() {
  float temp = dht.readTemperature();
  float humidity = dht.readHumidity();
  int lightValue = analogRead(LDR_PIN);
  bool isRainingNow = (digitalRead(RAIN_SENSOR_PIN) == LOW);
  bool isReedSwitchClosed = (digitalRead(REED_PIN) == LOW);

  if (isnan(temp) || isnan(humidity)) return;

  Blynk.virtualWrite(V0, temp);
  Blynk.virtualWrite(V1, humidity);
  Blynk.virtualWrite(V3, lightValue);
  Blynk.virtualWrite(V4, isRainingNow ? "Raining" : "Clear");

  if (windowClosed && !isReedSwitchClosed && !servoMoving && !intrusionActive) {
    Serial.println("ALERT: Window was forced open!");
    Blynk.virtualWrite(V8, "\n!!! INTRUSION ALERT !!!\nWindow forced open!\n\n");
    Blynk.logEvent("intrusion_alert", "ALERT: The window has been forced open!");
  }
}
```

```
    intrusionActive = true;
    lastBeepTime = millis();
}

if ((!windowClosed || isReedSwitchClosed) && intrusionActive) {
    intrusionActive = false;
    stopIntrusion = false;
    Serial.println("Intrusion resolved: Window closed.");
    Blynk.virtualWrite(V8, "Intrusion cleared. Window is closed.\n");
}

if (!automationEnabled || servoMoving) return;

int currentLightCondition = lightValue <= LDR_THRESHOLD ? 1 : 0;
bool rainChanged = isRainingNow != prevRainStatus;
bool lightChanged = currentLightCondition != prevLightCondition;

if (isRainingNow && rainChanged) {
    Serial.println("Condition: Rain. Closing.");
    closeWindow();
} else if (!isRainingNow && rainChanged) {
    Serial.println("Rain stopped.");
    if (currentLightCondition == 1) openWindow();
    else closeWindow();
} else if (!isRainingNow && lightChanged) {
    if (currentLightCondition == 1) {
        Serial.println("Daytime. Opening.");
        openWindow();
    } else {
        Serial.println("Nighttime. Closing.");
    }
}
```

```
        closeWindow();
    }
}

prevRainStatus = isRainingNow;
prevLightCondition = currentLightCondition;
}

void fastSensorCheck() {
    bool isRainingNow = (digitalRead(RAIN_SENSOR_PIN) == LOW);
    bool isReedSwitchClosed = (digitalRead(REED_PIN) == LOW);

    if (windowClosed && !isReedSwitchClosed && !servoMoving && !intrusionActive) {
        Serial.println("FAST ALERT: Window was forced open!");
        Blynk.virtualWrite(V8, "\n!!! INTRUSION ALERT !!!\nWindow forced open!\n\n");
        Blynk.logEvent("intrusion_alert", "ALERT: The window has been forced open!");
        intrusionActive = true;
        lastBeepTime = millis();
    }

    if ((!windowClosed || isReedSwitchClosed) && intrusionActive) {
        intrusionActive = false;
        stopIntrusion = false;
        Serial.println("Fast Alert resolved: Window closed.");
        Blynk.virtualWrite(V8, "Fast alert cleared.\n");
    }

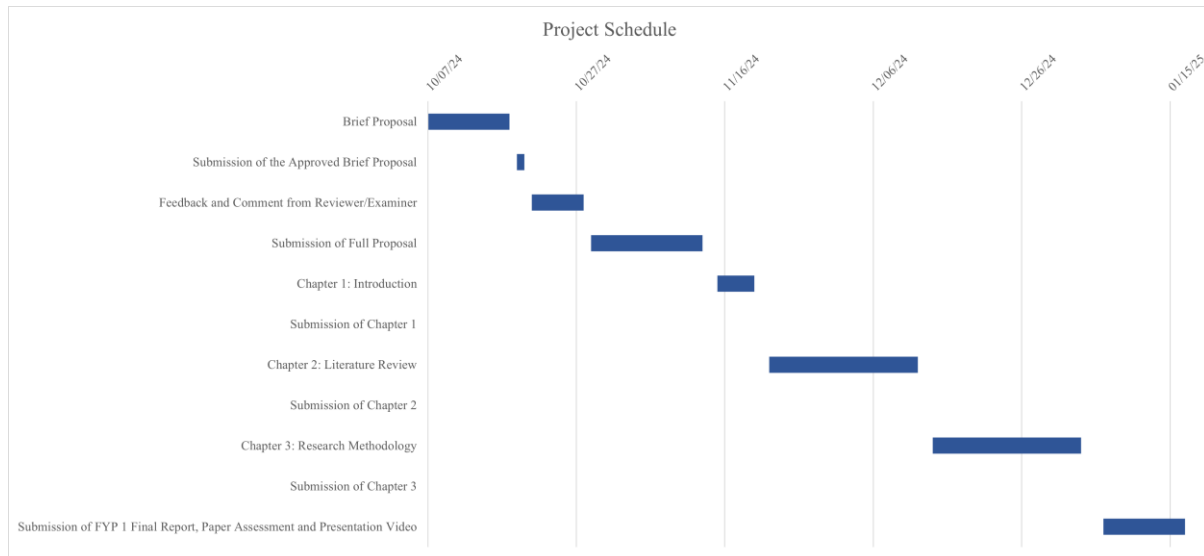
    if (automationEnabled && isRainingNow && !servoMoving && !windowClosed) {
        Serial.println("FAST: Rain Detected. Closing window.");
        closeWindow();
    }
}
```

```
}  
}  
  
void setup() {  
  Serial.begin(115200);  
  pinMode(RAIN_SENSOR_PIN, INPUT);  
  pinMode(LDR_PIN, INPUT);  
  pinMode(REED_PIN, INPUT_PULLUP);  
  pinMode(BUZZER_PIN, OUTPUT);  
  digitalWrite(BUZZER_PIN, LOW);  
  
  dht.begin();  
  
  windowClosed = (digitalRead(REED_PIN) == LOW);  
  
  Serial.println(windowClosed ? "Window is INITIALLY CLOSED" : "Window is  
INITIALLY OPEN");  
  
  WiFi.begin(ssid, pass);  
  Blynk.config(BLYNK_AUTH_TOKEN);  
  Blynk.connect(3000);  
  
  timer.setInterval(4000L, checkSensorsAndAutomate);  
  timer.setInterval(200L, fastSensorCheck); // Fast loop for intrusion and rain  
}  
  
void loop() {  
  Blynk.run();  
  timer.run();  
  
  if (servoMoving && (millis() - servoStartTime >= servoRunDuration)) {  
    myServo.write(90);  
    delay(150);  
  }  
}
```

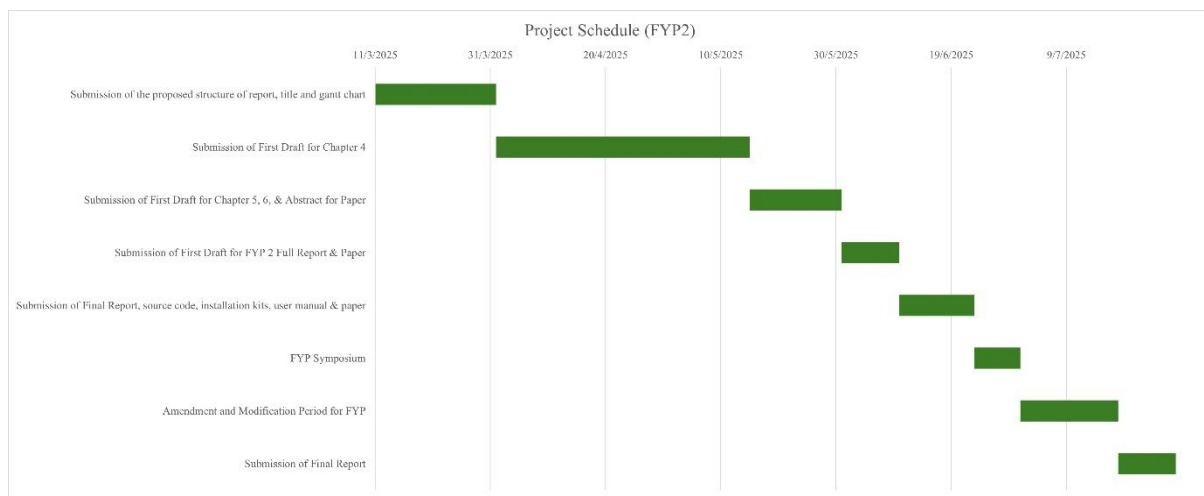
```
myServo.detach();
servoMoving = false;
Serial.println("Servo done");
Blynk.virtualWrite(V8, "Servo done\n");
Blynk.virtualWrite(V2, windowClosed ? "Closed" : "Open");
}

if (intrusionActive && !stopIntrusion) {
  if (millis() - lastBeepTime > 1000) {
    beepWarning(4);
    lastBeepTime = millis();
  }
}
}
```

APPENDIX B: PROJECT SCHEDULE



Gantt Chart of Project Schedule (FYP 1)



Gantt Chart of Project Schedule (FYP 2)

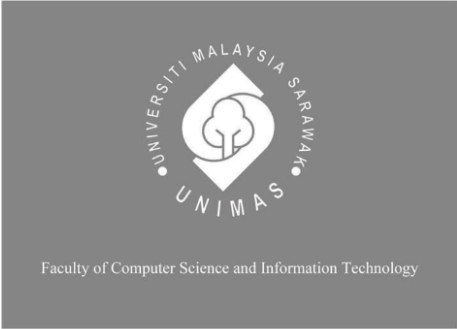
APPENDIX C: REPORT SIMILARITY PERCENTAGE



NOUF HASYA ILMA BINTI BADARUDDIN . FYP DRAFT

?





UNIVERSITI MALAYSIA SARAWAK
UNIMAS

Faculty of Computer Science and Information Technology

IntelliVent: Smart Window Automation for Weather and Time Optimization

Noof Hasya Ilma Binti Badaruddin (78334)



Match Overview

3%

< >

1

Submitted to Heriot-Wa...
Student Paper

2% >

2

ir.unimas.my
Internet Source

1% >

Page: 1 of 96

Word Count: 16894

Text-Only Report | High Resolution

On

