MINISTRY OF SCIENCE AND HIGHER EDUCATIONOF THE RUSSIAN FEDERATION
Federal state autonomous educational institution of higher education“South Ural State University (National Research University)”
Institute of Engineering and Technology
Electric drive, mechatronics and electromechanics department
15.03.06 “Mechatronics and Robotics”

ALLOW TO DEFEND
Head of department Electric drive, mechatronics and electromechanics PhD, professor
________________M.A. Grigor’ev
«______»______________2025

Smart Home Automation System Using IoT

THESIS WORK OF THE BACHELOR'S PROGRAM
“MECHATRONICS AND ROBOTICS”
SUSU-15.03.06.2025.454 EN TW

Head of the work, PhD,
Associate professor
_______________D. Yu. Khriukin
«______»_______________2025

Author of work,
Student of Bachelor’s program
of P-454 group
__________________H. Hussein

«______»______________2025

Normocontroller
____________N.M. Maksimov
«_____»_______________2025

Chelyabinsk 2025

MINISTRY OF SCIENCE AND HIGHER EDUCATIONOF THE RUSSIAN FEDERATION
Federal state autonomous educational institution of higher education“South Ural State University (National Research University)”
Institute of Engineering and Technology
Electric drive, mechatronics and electromechanics department
15.03.06 “Mechatronics and Robotics”

ALLOW TO DEFEND
Head of department Electric drive, mechatronics and electromechanics PhD, professor
______________________M.A. Grigor’ev
«__________»___________________2025

TASK
of thesis work (TW) of the student of Bachelor's program

Hiba Hussein
P-454 group

1. The theme of TW of the undergraduate:
   Smart Home Automation System Using IoT

and it is approved as the order on university from 21.04.2025 №654-13/12 (Приложение №13)

2. The deadline of thesis work: 20.06.2025.
3. Basic data to work:
   Results and studies on the implementation of intelligent systems for home automation using IoT technologies, including sensor networks, remote control via cloud services, and AI-based optimization of energy usage.
4. Contents of the task for explanatory note of thesis:

ANNOTATION
ABSTRACT
INTRODUCTION: Background and significance of Smart Home Automation in modern society.
LITERATURE REVIEW: Major Observations from Previous Studies. Problem Identification. Standards. Constraints in the Project. Trade-Offs in the Project. Conventional Smart Home Automation System
PROJECT DESCRIPTION: System Architecture. Objectives. Methodology. Application. IoT-Based Smart Home Automation Framework. Hardware Elements. Future Prospects of Smart Home Automation.
HARDWARE OVERVIEW: System Specifications. Components Used.
SIMULATED CIRCUIT DESIGN AND OUTPUT ANALYSIS: Simulation-Based Evaluation. Proteus Design Suite. Fundamental System Structure. System Operation

Simualtion output. Embedded Program for Arduino IDE 60. Source Code Used For Hardware
CONCLUSIONS
REFERENCES
5. Slides:

1. Architecture of Smart Home and sensor-actuator modeling
2. Implementation of remote cloud control and automation
3. Optimization scenarios: lighting, security, fire response
4. Energy optimization model: variables and objective functions
5. Case study: lighting and HVAC results
6. Case study: security and emergency automation results
7. Validation of prototype using Arduino and Proteus
8. SWOT analysis of the smart home system
9. K. Levin’s model applied to adoption barriers and drivers
10. Gantt chart of development and implementation phases

6. The task is given September 02, 2025

Head of department________________________________M.A. Grigor’ev
Head of work_____________________________________ D. Yu. Khriukin
The task I’ve accepted to execution____________________ H. Hussein

ANNOTATION

H.Hussein. Research Smart Home Automation System Using IoT – Chelyabinsk: SUSU, PI, PEF; 2025,75p.,21figures, references – 12, 10 slides of presentation

This thesis investigates the conceptualization, development, and assessment of a Smart Home Automation System grounded in Internet of Things (IoT) technologies. The primary objective is to create an intelligent and user-centric system that optimizes residential efficiency, comfort, safety, and energy utilization by enabling automated control and remote access functionalities.

The project leverages a modular architecture consisting of multiple hardware components, including microcontrollers (e.g., ESP32/Arduino), sensors (temperature, motion, humidity, gas), relays, and actuators, interconnected via Wi-Fi and controlled using the MQTT protocol. A central control unit collects and processes real-time data, enabling the automation of household functions such as lighting, HVAC systems, door locks, and security alarms. Additionally, the system is integrated with a mobile application and a web dashboard, allowing users to monitor home status and issue control commands remotely and securely.

The results demonstrate that the proposed IoT-based smart home automation system is not only technically viable but also economically practical, scalable, and adaptable to a wide range of user needs. It contributes to the growing field of sustainable intelligent living and provides a foundation for further research and development in home automation and smart cities.

TABLE OF CONTENTS

ABSTRACT

This research introduces a non-intrusive home automation solution developed within the domain of assistive technology. The system architecture is centered around the Arduino microcontroller and incorporates Bluetooth communication, specifically tailored to support elderly individuals and persons with disabilities. A user-friendly interface, deployed on an Android-based smartphone, enables intuitive and accessible interaction with the system. Experimental demonstrations confirm that the proposed system effectively empowers the target user group to manage a range of household appliances, including lighting, climate control systems, and security infrastructure.

Achieving optimal living conditions in residential settings necessitates both environmental monitoring and automation. Human comfort can be categorized into multiple dimensions, with thermal comfort—governed by ambient temperature and humidity—being of primary importance. Additional aspects include visual comfort, determined by lighting intensity and color composition, and hygienic comfort, which pertains to indoor air quality. The transition from a conventional residence to a smart home requires the intelligent and automated execution of control commands derived from sensor data analysis—an objective facilitated by IoT-based systems.

The system design leverages the Emmons platform to collect, visualize, and manage monitored environmental data, while also enabling remote control of connected household devices. Emmons offers a highly adaptable and user-centric interface. Environmental sensing within the home is accomplished using the NodeMCU-ESP8266 microcontroller, which provides real-time data acquisition, processing, and two-way communication with the Emmons cloud infrastructure for continuous data transmission and command execution.

1 INTRODUCTION

Cloud-based home automation systems offer users the ability to control household devices and functions remotely using internet-connected devices such as smartphones, tablets, or computers. Often referred to as *smart homes*, these systems provide convenience, security, and energy efficiency by allowing users to manage appliances, lighting, and other systems from any location in the world. The primary advantage of such systems lies in their remote accessibility, which distinguishes them from conventional home control setups.

In this project, we designed and implemented a smart home automation system that integrates mobile devices, cloud networking, wireless communication, and power-line communication. As shown in Figure 1.1, the system enables remote control of various home appliances and lighting systems. It features three user interface options: a smartphone application, a PC-based control program, and a handheld wireless remote. This variety allows users to interact with the system based on their preferences and ensures flexibility in operation.

One notable feature of this system is its ability to function without an internet connection through the in-home wireless remote. This provides local control even in cases of internet outages, which is a limitation in many cloud-dependent systems.

The system was also designed with scalability and cost-effectiveness in mind. It supports a wide range of devices and can be expanded to accommodate future needs. This makes it a practical solution not only for residential applications but also for small businesses. Its user-friendly interface and flexible architecture make it a reliable and accessible option for modern home automation.

Figure 1.1 – Overview of a Smart Home System

Recent advancements in computer vision technology present significant opportunities to modernize waste collection processes within hospital environments. Effective and automated waste management is critical not only for maintaining hygiene and reducing exposure to airborne pathogens among medical staff but also for supporting the broader implementation of precision healthcare systems. This study introduces a comprehensive solution for automated waste collection by detailing the development of an intelligent "Zig Bin" system. The process spans from image acquisition and processing to real-time monitoring through a web-based interface.

The proposed system incorporates a smart bin integrated with a line-following robot, designed to navigate hospital corridors and autonomously collect waste without the need for human operation. By utilizing automation and networked communication, the system improves operational efficiency and safety in clinical settings.

Central to the design is the use of an embedded system, defined as a computer system built around a microprocessor or microcontroller that performs dedicated functions. These systems can vary in complexity—from simple setups with a single microcontroller to sophisticated configurations involving multiple processors, various peripherals, and communication networks. Interfaces may range from minimal to advanced graphical user interfaces (GUIs), depending on the intended functionality. Notably, embedded systems account for nearly 98% of all microprocessor applications, highlighting their importance in specialized and mission-critical environments such as healthcare.

A significant majority of microprocessors manufactured today are used within embedded systems, demonstrating their widespread role across various industrial and technological domains. These systems are tailored to perform dedicated tasks with high precision and are especially valuable in environments that require reliability and efficiency.

Cloud computing enables the delivery of a wide range of services—such as data storage, computing power, networking, and software—via the internet. Instead of relying on local storage devices, users can store and access their data through remote servers. This approach has become increasingly popular due to its flexibility, cost-effectiveness, scalability, enhanced performance, and robust security features.

Interconnection networks form the structural foundation that connects different system components, such as processors and memory modules. These networks consist of switching elements and follow specific topological designs that define how elements are linked. They are commonly divided into two categories:

- Direct networks establish static, point-to-point links between neighboring nodes. Examples include ring, mesh, and hypercube configurations.
- Indirect networks allow dynamic communication paths, which can change based on application requirements. These include bus architectures, multistage networks, and crossbar switch designs.

The integration of computer vision, embedded systems, cloud infrastructure, and advanced networking technologies creates a powerful foundation for smart waste management in healthcare facilities. This combination facilitates real-time automation, improves safety, reduces the risk of human exposure to hazardous materials, and supports efficient hospital operations.

2 LITERATURE REVIEW

2.1 Major Observations from Previous Studies

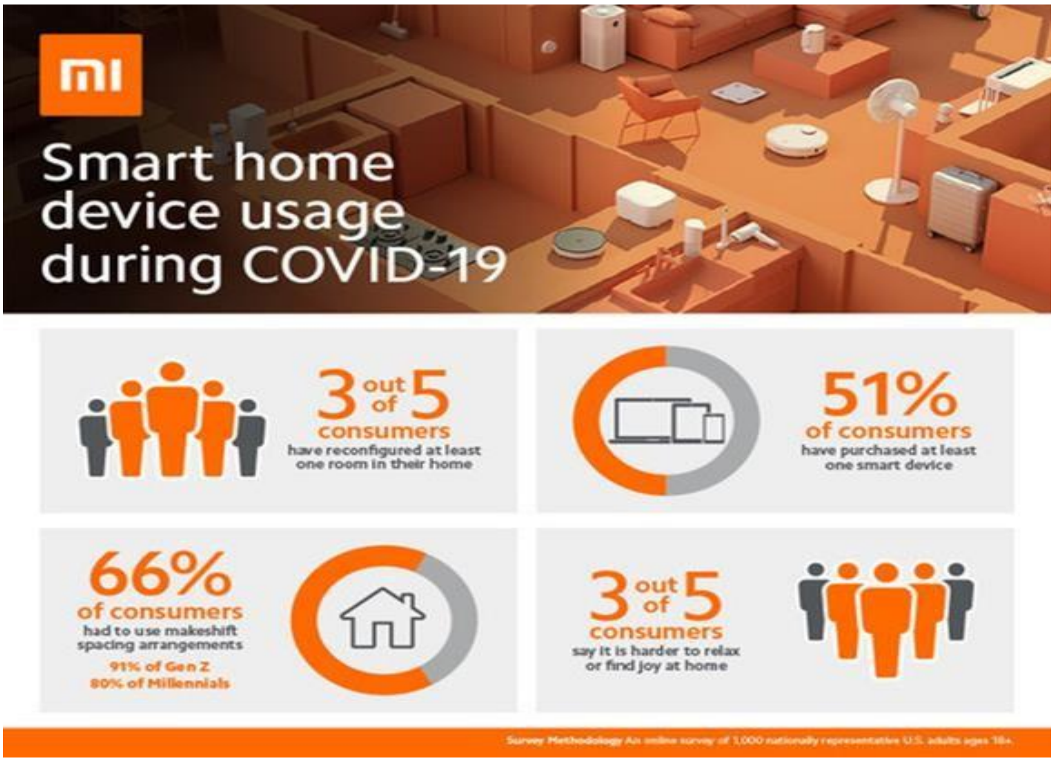


Figure 2.1 – Usage of Smart Applications During the COVID-19 Lockdown

As depicted in Figure 2.1, a recent study indicates that approximately 70% of consumers made enhancements to their homes during the COVID-19 pandemic. Notably, over half of these improvements involved the installation of smart technologies. The shift towards smart home adoption has been instrumental in helping individuals adjust to spending more time indoors during the global health crisis.

The concept of smart living revolves around the transformation and optimization of residential environments through technological innovation. This approach aims to solve everyday challenges and adapt to changing lifestyles. According to Daniel Desmarais, Global Product Marketing Manager at Xiaomi, the year 2020 marked a significant acceleration in this trend. He emphasized that the integration of connected devices, automated systems, and innovative technologies is enabling individuals to build comprehensive and responsive home ecosystems.

This technological transition has enabled households to address the distinct challenges brought about by prolonged periods of home confinement. Many individuals have adapted by reconfiguring their living spaces—transforming areas into home offices, virtual classrooms, or multifunctional zones. Others have focused on streamlining daily routines and improving convenience through enhanced automation and control. These innovations are redefining the concept of home living by integrating technology into everyday activities.

Such a shift highlights the increasing relevance of smart home systems in promoting both functional efficiency and environmental flexibility. As user needs continue to evolve, these technologies empower individuals to adapt their living environments accordingly—offering a tailored and responsive approach to modern domestic life.

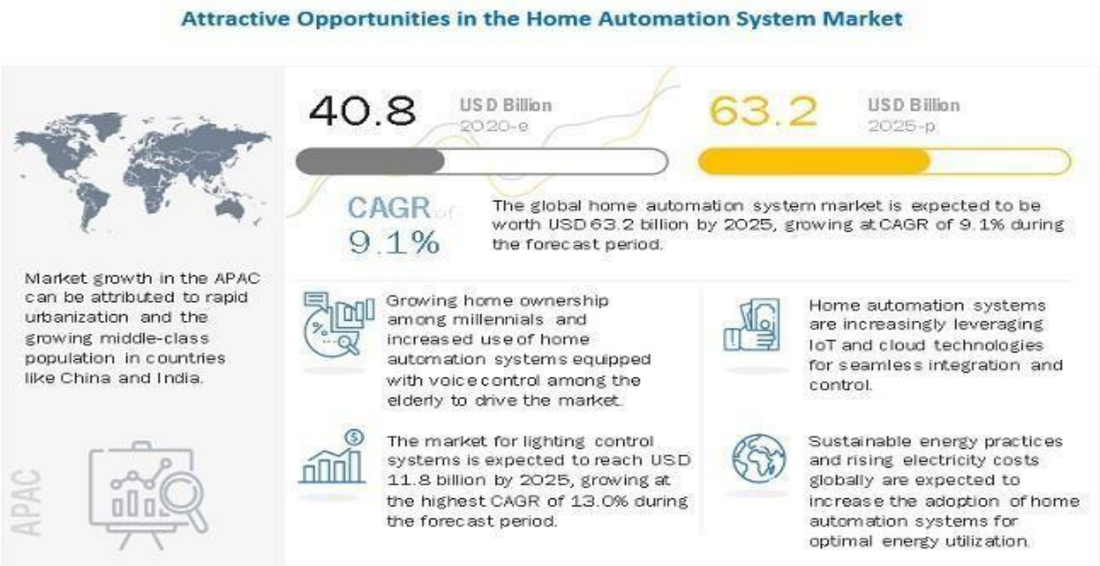


Figure 2.2: Market Growth of Home Automation Systems

Based on data from Markets and Markets, as shown in Figure 2.2, the global market for home automation systems is expected to experience substantial growth—rising from USD 40.8 billion in 2020 to USD 63.2 billion by 2025. This increase represents a Compound Annual Growth Rate (CAGR) of 9.1% over the forecast period (2020–2025).

The projected expansion is largely attributed to the growing implementation of diverse home automation technologies across the globe. These include both high-end, fully integrated smart home solutions and more accessible do-it-yourself (DIY) systems.

Such technologies not only improve user convenience and comfort but also contribute to increased energy efficiency, enhanced security, and smarter resource management within residential settings.

Enhanced home security and improved energy efficiency are among the primary benefits driving interest in smart home solutions. The aim of this report is to outline, explain, and project the development of the home automation market by examining key factors such as system management, product categories, software and algorithm integration, and geographic distribution. Important market growth drivers include reduced insurance premiums for homeowners who install smart security systems, along with the growing number of service offerings from telecommunications providers.

A relevant study by Indeervar Reddy et al. (2017), titled "Home Automation of Lights & Fans Using IoT," investigates the impact of technological advancement on electricity consumption patterns at both local and global levels. The authors highlight a steady rise in demand for electrical energy across residential and commercial sectors, which has led to several challenges, including power shortages, load shedding, and outages caused by adverse weather conditions such as heavy rain and storms. These disruptions can negatively affect household appliances and overall energy management efficiency.

To mitigate such issues, the researchers propose a comprehensive home automation solution aimed at optimizing energy usage. Their system introduces three modes of appliance automation: local control, web-based access, and mobile application-based management. The local automation is achieved through a microcontroller interface, while remote access is facilitated through a dedicated web portal or smartphone application. This structure allows users to efficiently manage power usage by switching appliances on or off either automatically or manually, based on their specific needs.

Experimental findings confirm the practicality of the proposed system, with recorded accuracy rates of 88.71% for local control, 88.55% for web-based control, and 88.56% for app-based automation. These results validate the system's reliability and potential for reducing energy consumption in smart home environments.

The system also addresses emergency scenarios by calculating the electrical load in real-time and performing intelligent switching to safeguard connected appliances. This feature ensures protection against damage from abnormal voltage conditions, such as power surges or drops, thereby extending the lifespan of household devices. Collectively, the proposed approach marks a meaningful advancement in efficient energy management and the broader development of smart home automation.

In a related study, Gomathi B. et al. (2015) introduced a system titled "Implementation of Fan ON/OFF Control Using the Internet of Things for Home Automation." Their research emphasizes the role of the Internet of Things (IoT) as a key enabler for the advancement of Smart Cities and intelligent energy systems. IoT is a rapidly evolving technology that facilitates the remote monitoring and control of devices through internet connectivity. Its relevance is particularly evident in the automation of industrial and residential systems using wireless communication, sensor integration, and embedded controllers.

Smart homes—driven by home automation technologies—are designed to enhance the quality of life through improved comfort, convenience, flexibility, security, and energy savings. The study demonstrates how IoT can be employed to automate household appliances, thereby creating an interconnected and responsive home environment. The proposed solution centers on the control of a fan's speed via an online interface built using HTML, while control commands are executed through a microcontroller programmed in Embedded C. This combination results in a responsive and reliable home automation system that can be accessed over the internet.

Such systems are especially advantageous for users who require remote access to their appliances, whether for convenience or necessity. Furthermore, this technology holds significant value for individuals with limited mobility, including the elderly and persons with physical or visual impairments, by offering greater autonomy and ease of use. The study concludes that the solution is both cost-effective and user-friendly, reinforcing its viability for widespread adoption in modern smart homes.

The core aim of these proposed systems is to design and implement home automation solutions that are not only effective but also cost-efficient, making smart living accessible to a wider audience.

Sudha Kousalya et al. (2014), in their study titled "IoT-Based Smart Security and Smart Home Automation," highlight the rapid emergence of the Internet of Things (IoT) as a transformative force in both technology and business. A key focus of their work is on the development of standardized wireless communication protocols, which are essential for ensuring compatibility and seamless connectivity among various IoT-enabled devices. Their findings emphasize the pivotal role of IoT in turning conventional homes into secure, energy-efficient smart environments.

In another study, Kusuma S. M2 discusses "Home Automation Using the Internet of Things," illustrating how advancements in automation have simplified tasks and improved quality of life across multiple sectors. The increasing reliance on automated systems, as opposed to manual operations, is attributed to greater efficiency and ease of use. The study notes that with the widespread growth of internet access over the past decade, IoT has emerged as a highly promising technological trend. IoT encompasses a broad ecosystem of interconnected devices—from industrial tools to everyday household items—capable of sharing data and performing tasks with minimal user intervention.

One application of this technology is the Wireless Home Automation System (WHAS), which enables users to control key household operations remotely through mobile devices or computers via internet access. These systems, often labeled as "smart homes," offer more than just convenience—they contribute to energy savings and reduce the need for physical effort. A defining feature of such systems is their remote accessibility, allowing users to manage their homes from virtually anywhere in the world with a network connection.

Further contributing to this growing field, Vaishnavi S. Gunge and Pratibha S. Yalagi presented a paper titled "Smart Home Automation: A Literature Review" at the National Seminar on Recent Trends in Data.

Presented at the National Seminar on Recent Trends in Data Mining (RTDM, 2016), Vaishnavi S. Gunge and Pratibha S. Yalagi emphasize the role of the internet in enabling both real-time data monitoring and manual control of smart home devices. They point out that most current versions of the Raspberry Pi come with built-in USB and Ethernet ports, making it easier to connect to the internet and upload data. This functionality greatly improves the integration of smart home systems, allowing users to monitor their environment, collect data, and control appliances more efficiently from remote locations.

2.2 Problem Identification

In today's fast-evolving technological landscape, home automation has emerged as a transformative tool aimed at enhancing quality of life. With continuous advancements in smart technology, the integration of automated systems into residential settings is increasingly recognized as a practical solution for achieving convenience, operational efficiency, and improved control over routine activities. A key benefit of home automation is its ability to simplify the management of household devices, thereby increasing comfort for users. Through the use of smart gadgets such as smartphones and tablets, individuals can effortlessly control various home functions—ranging from lighting and entertainment systems to remote door access—without needing to be physically present.

This centralized control framework not only enhances user experience but also supports more sustainable energy use. Automation systems can help reduce energy waste by enabling users to schedule or adjust energy consumption based on actual needs, leading to lower utility costs and contributing positively to environmental conservation. In this way, home automation systems are aligned with broader global efforts toward energy efficiency and responsible consumption.

Despite these advantages, several challenges have limited the widespread adoption of home automation technologies. The most prominent barrier is the high cost of installation and maintenance, which makes such systems inaccessible for many households—especially those in lower- or middle-income segments. The technical complexity involved in designing and configuring automation systems further compounds the issue. Many users are discouraged by the need for professional expertise or advanced technical knowledge, which adds both financial and logistical burdens.

As a result, there is a clear need to explore alternative automation solutions that are affordable, scalable, and easy to implement. Systems developed with simplicity and cost-efficiency at their core could greatly expand accessibility and encourage more widespread use. Ideally, such systems should be user-friendly enough to be configured and operated by individuals with minimal technical background, making them suitable for a broad range of environments—including homes, educational institutions, and workplace settings.

Moreover, home automation has the potential to significantly improve the daily lives of elderly individuals and those with physical disabilities. These groups often face difficulties in performing basic tasks within their living environments. Through automation, tasks such as adjusting temperature settings or operating kitchen appliances can be made more manageable, promoting independence and improving overall quality of life.

Tasks such as operating household appliances, managing home functions, or communicating with others can be easily carried out using a touchscreen interface or simple voice commands. This level of accessibility significantly enhances the independence and self-reliance of individuals with physical limitations or age-related difficulties. By minimizing the need for constant physical support or supervision, smart home technologies promote a greater sense of dignity and autonomy among these users.

In summary, while challenges such as high initial costs and system complexity continue to hinder the widespread implementation of home automation, the technology holds considerable promise. As innovation continues to drive down costs and improve usability, home automation systems have the potential to become a mainstream component of everyday life. Beyond offering convenience, these solutions support more inclusive living environments, contribute to energy conservation, and improve overall quality of life across diverse user groups.

2.3 Standards

In the rapidly advancing domains of smart technology and the Internet of Things (IoT), the role of standardization is fundamental to achieving consistent interoperability, system reliability, and secure operation across a diverse range of devices and platforms. Various IEEE standards provide essential guidelines that support developers, engineers, and researchers in building and maintaining cohesive, high-performance systems. These standards supply shared frameworks, terminologies, and methodological approaches that streamline design, enhance compatibility, and simplify system integration. The following section outlines key IEEE standards that are particularly relevant to sensor systems, IoT infrastructure, blockchain connectivity, and embedded system development.

IEEE 2700-2017: Standard for Sensor Performance Parameter Definitions

The IEEE 2700-2017 standard presents a standardized approach to defining the performance parameters of a wide array of sensors used in both industrial and consumer electronic systems. It encompasses numerous sensor types, including accelerometers, gyroscopes, magnetometers, pressure sensors, humidity and temperature sensors, barometers, ambient light sensors, and proximity detectors. This standard aims to unify terminology, performance criteria, testing conditions, and units of measurement across sensor technologies, fostering a shared specification language that benefits sensor manufacturers, developers, and application integrators.

By reducing inconsistencies in how sensor performance is described and evaluated, IEEE 2700-2017 facilitates smoother integration of sensor components into complex systems. It supports faster product development cycles by minimizing ambiguity during specification and testing phases. The standard is particularly relevant for sensors using digital interfaces and is highly applicable in multi-sensor environments found in applications such as smart home systems, wearable technology, and robotic platforms. It defines a baseline set of essential performance metrics necessary for effective cross-platform sensor communication and interoperability.

By establishing a required set of performance parameters, IEEE 2700-2017 allows system designers to assess and compare sensor modules with greater clarity and consistency. This standard plays a key role in supporting informed decision-making during the design and integration of sensor-based systems.

IEEE 2144.1-2020: Framework for Blockchain-Based IoT Data Management

The IEEE 2144.1-2020 standard provides a comprehensive framework for managing data generated by IoT devices through the use of blockchain technology. As IoT systems increasingly generate large volumes of sensitive and distributed data, the need for secure, transparent, and verifiable data management has become critical. This standard defines the architectural components and interface requirements needed to implement blockchain-based systems that handle data collection, processing, storage, and exchange.

The framework is adaptable to a wide range of business use cases, including internal analytics, external data marketplaces, and collaborative data-sharing environments. It also addresses essential factors such as data security, user privacy, and cross-system compatibility. IEEE 2144.1-2020 serves as an important guideline for building scalable and secure IoT ecosystems, especially in sectors where data authenticity and controlled access are vital—such as healthcare, industrial automation, and smart urban infrastructure.

IEEE 2413-2019: Architectural Framework for the Internet of Things (IoT**)**

IEEE 2413-2019 outlines a high-level architectural model for IoT systems that can be applied across various domains. It defines the core components, interface protocols, and interaction mechanisms that form the foundation of general IoT system architecture. The standard’s main objective is to establish a common language and structural approach for stakeholders involved in IoT development, including engineers, architects, and end users.

This framework promotes the creation of interoperable IoT systems that prioritize openness, scalability, and security. Additionally, IEEE 2413-2019 supports the convergence of diverse technologies and platforms, helping to reduce fragmentation within the IoT landscape. It provides essential guidance on addressing systemic challenges such as data transmission, device lifecycle management, network configuration, and privacy protection. As a result, the standard serves as a key reference for constructing reliable and flexible IoT infrastructures.

IEEE 1118.1-1990: Microcontroller System Serial Control Bus

Though currently classified as inactive and officially withdrawn, IEEE 1118.1-1990 played a foundational role in shaping serial communication standards for microcontroller-based systems. It introduced a bit-serial, multidrop bus system designed for connecting multiple devices within distributed environments, which proved especially useful in automation and embedded system applications.

Building upon the established BITBUS protocol, the standard enhanced communication functionality while retaining backward compatibility with earlier systems. It detailed an integrated architecture covering communication services, data link protocols, and options for physical media. Although no longer active, IEEE 1118.1-1990 continues to serve as a valuable resource for understanding the historical development of serial communication technologies and remains influential in the design of modern embedded system protocols.

IEEE 1621-2004: User Interface Elements for Power Control in Electronic Devices

IEEE 1621-2004 outlines design guidelines for user interface components used to control power in both office equipment and consumer electronics. The standard offers recommendations related to visual indicators, audio signals, tactile feedback, and control mechanisms aimed at improving user interaction with power management features.

While the standard was published in 2004, it still holds relevance today as a reference for designing accessible and intuitive power control interfaces. It contributes to energy efficiency initiatives by promoting designs that are easy to use and that encourage users to engage with energy-saving features. Particularly in consumer electronics, where usability and experience are crucial, IEEE 1621-2004 continues to influence best practices in user interface design related to power functionality.

2.4 Constraints in the Project

One of the primary limitations in implementing home automation systems is the financial investment required. The initial setup can be expensive, particularly when multiple smart devices or advanced system configurations are involved. This can restrict accessibility for households with limited budgets.

A major technical challenge lies in ensuring interoperability among devices from different manufacturers. In many cases, smart devices operate on different communication protocols or platforms, which may hinder seamless integration and reduce overall system efficiency.

Home automation systems rely heavily on uninterrupted internet connectivity and stable power supply. Any disruptions, such as power outages or network failures, can impair system performance or cause critical features to become inoperable.

The use of internet-connected devices and sensors introduces potential risks related to data security and privacy. Without adequate cybersecurity measures, these systems may be vulnerable to hacking, unauthorized access, or data breaches, putting user information at risk.

For users without technical expertise, configuring and managing a home automation system can be challenging. The need to understand device settings, network configurations, and software platforms may act as a barrier to adoption.

Retrofitting smart systems into older buildings can present difficulties, especially if the existing electrical or networking infrastructure is outdated or incompatible with modern automation technologies. This may require additional modifications, increasing the overall cost and complexity of the project.

2.5 Trade-Offs in the Project

Home automation involves the application of modern technologies to control and manage various residential functions—such as lighting, climate control, security systems, and entertainment devices. These systems are typically operated remotely or automatically through smartphones, voice-activated assistants, or centralized control units. Although this integration offers substantial advantages in terms of convenience, security, and energy management, it also introduces important trade-offs that must be taken into account during the planning and implementation stages.

A primary trade-off is the substantial upfront investment required. A fully functional home automation system demands a range of components, including smart sensors, actuators, control hubs, and compatible appliances. The overall cost can be significantly higher for larger homes or for systems involving numerous interconnected devices. Additionally, when professional installation is needed—either due to user inexperience or the challenges of upgrading older electrical systems—the cost and complexity of deployment can increase further. This financial burden may limit the accessibility of such systems, especially for budget-conscious users.

Another significant trade-off associated with home automation is the risk of system malfunction or operational failure. Because these systems are dependent on electronic components and consistent internet connectivity, disruptions such as power outages, software errors, or network failures can render the system temporarily inoperative. This can cause major inconvenience, particularly when essential functions—like security alarms or smart door locks—are affected. In more severe instances, system vulnerabilities may be exploited by cyber attackers, potentially leading to breaches of privacy or unauthorized access to the home.

Privacy is also a critical concern in the context of smart home technologies. To operate effectively, these systems often gather and store sensitive personal information, such as user preferences, behavioral patterns, and activity logs. While this data supports automation and customization, it also raises concerns about surveillance and the potential misuse of user data. Without strong cybersecurity measures, there is a risk of unauthorized access, data leaks, or misuse by third-party services, which may deter privacy-conscious users from adopting the technology.

Furthermore, although one of the primary goals of home automation is to improve energy efficiency—through automated lighting, temperature control, and appliance management—this benefit may not always be realized. Poor configuration, lack of user oversight, or continuous background activity of connected devices can lead to unnecessary energy consumption. In such cases, users may experience increased electricity costs and a larger environmental impact, counteracting the intended sustainability benefits.

In summary, while home automation offers transformative advantages, it comes with trade-offs related to cost, reliability, security, privacy, and energy usage. These challenges must be carefully evaluated and addressed to ensure the long-term effectiveness and acceptance of smart home technologies.

2.6 Conventional Smart Home Automation System

The implementation of a conventional smart home automation system often involves the use of embedded platforms and microcontrollers to manage and control devices within the household. One such approach includes the construction of a micro web-server using the Arduino Mega 2560 in combination with the Arduino Ethernet Shield. Arduino, a widely used open-source prototyping platform, is favored for its simplicity, adaptability, and support for both hardware and software development—making it highly suitable for embedded control applications.

The Arduino Mega 2560, which operates on the ATmega2560 microcontroller, offers extensive I/O capabilities, featuring 54 digital input/output pins. This allows for seamless integration with a wide variety of sensors and actuators, making it possible to control multiple home appliances and automation components simultaneously. This setup enables users to create a centralized and responsive automation system, capable of performing remote operations and real-time monitoring via web-based interfaces.

In the development of a typical smart home automation system, internet-based communication is achieved by connecting the Arduino Mega 2560 to an Ethernet Shield through Serial Peripheral Interface (SPI) pins. This setup enables the microcontroller to interface with the internet for remote control functionality. To manage household appliances, low-voltage switching relays are used. These relays serve as a bridge between the Arduino's low-power output signals and the higher voltage AC-powered home devices, allowing effective control of various appliances such as lights and fans.

To enhance user interaction, a custom mobile application was created as part of the system. This application supports both manual controls and voice commands, enabling users to operate appliances such as air conditioners, fans, and washing machines with ease. It communicates with the Arduino-based system via both Wi-Fi and Bluetooth, offering flexibility in both local and remote usage. The app includes a user authentication process based on IP address and password verification to ensure secure access. Once logged in, users can control devices directly through on-screen buttons or using voice input for hands-free operation.

For environments with limited or unstable internet connectivity, the system incorporates a Bluetooth-controlled relay bot. This feature allows local wireless control, maintaining functionality even in the absence of a stable internet connection.

The Internet of Things (IoT) has become a fundamental component in modern home automation systems. With IoT technology, devices such as smart thermostats, lighting systems, and various sensors can be monitored and managed remotely through mobile applications. The increasing availability and affordability of smartphones have further accelerated the adoption of such technologies, making smart homes more accessible to the general public.

One of the primary advantages of incorporating IoT into home automation is its contribution to energy efficiency. These systems can automate energy-saving actions or provide users with real-time data on energy usage. As a result, homeowners are empowered to make informed decisions, reduce consumption, and lower their utility bills, aligning with sustainable energy practices.

However, several limitations still affect the widespread implementation of IoT-based automation. Replacing conventional household devices with smart alternatives can be expensive, creating a financial barrier for many users. Moreover, integrating older, non-IoT appliances into a modern automated setup often presents technical challenges. These devices may require additional hardware or middleware solutions to enable communication and compatibility with the broader smart system.

As emphasized by researchers like Gill et al., one of the primary obstacles to the widespread adoption of home automation is the invasive and complex nature of system installation. Retrofitting existing homes to support smart technologies often involves considerable time, financial investment, and structural modification. These factors can discourage homeowners from embracing fully automated environments, despite the long-term benefits.

In summary, the use of platforms such as Arduino and IoT devices illustrates the technical feasibility and potential advantages of home automation. However, for these systems to gain broader acceptance in everyday life, challenges related to affordability, ease of use, and cross-device compatibility must be effectively addressed.

A major technical challenge in advancing smart home adoption lies in integrating conventional, non-IoT appliances into modern automation systems. This process must be accomplished with minimal disruption to the existing home infrastructure. To promote mass adoption, retrofit solutions should be modular and non-invasive, enabling users—even those without technical expertise—to install and configure smart features without requiring complex modifications such as rewiring or replacing legacy equipment.

Another critical issue is the lack of standardized communication protocols among smart devices. As noted by Gill et al., the coexistence of various wireless standards (e.g., Zigbee, Z-Wave, Wi-Fi, Bluetooth, and proprietary technologies) leads to limited interoperability between devices from different vendors. This fragmentation reduces user convenience and impedes the creation of cohesive systems. Therefore, an ideal home automation architecture should adopt open and widely supported communication protocols, ensuring compatibility and seamless operation among devices from diverse manufacturers.

To address protocol limitations and extend the reach of automation solutions, researchers have explored alternative connectivity frameworks for remote monitoring and control. One promising direction is the use of General Packet Radio Service (GPRS) technology, which leverages mobile networks to enable remote access to home systems. GPRS-based solutions offer a cost-effective means of automation, especially in regions where fixed broadband access is limited or unavailable. By using mobile data networks, these systems allow users to monitor and manage their smart home devices from virtually any location, enhancing flexibility and system accessibility.

A significant application of GPRS in smart home development involves a system that integrates webcam surveillance and motion detection. This system captures images of the home environment and transmits them to the user's mobile device using GPRS. Motion detection is implemented by analyzing consecutive video frames for variations in light intensity. Notably, the live video streaming function utilizes the home's internet connection rather than relying solely on the GSM modem. This approach ensures better video quality while minimizing the consumption of mobile data.

In another study, U. Ali introduced a client-server-based automation model designed for both home and office environments. In this setup, a mobile device equipped with a micro-Java application functions as the user interface, connecting to a PC within the home. The PC, in turn, communicates with household devices via the parallel port. This configuration not only allows remote control of appliances but also enables real-time status monitoring, offering comprehensive management capabilities.

An alternative design leverages Wireless Sensor Networks (WSNs) in combination with GPRS to create a centralized automation system. Through this approach, users can control household appliances, monitor environmental parameters such as temperature and humidity, and receive status updates on their mobile devices. Unlike decentralized frameworks, this model employs a centralized controller to improve coordination and operational efficiency across the system.

In a separate implementation, S. R. Das and collaborators developed a security-oriented smart home solution for iOS devices. This system uses GPRS for communication between the user's mobile device and a cloud-based server. Equipped with motion sensors and surveillance cameras, the system activates video recording when motion is detected and sends real-time alerts to the homeowner. Users can access live camera feeds through GPS-enabled functions on their smartphones, enhancing both situational awareness and the ability to respond promptly to security events.

3 PROJECT DESCRIPTION

3.1 System Architecture

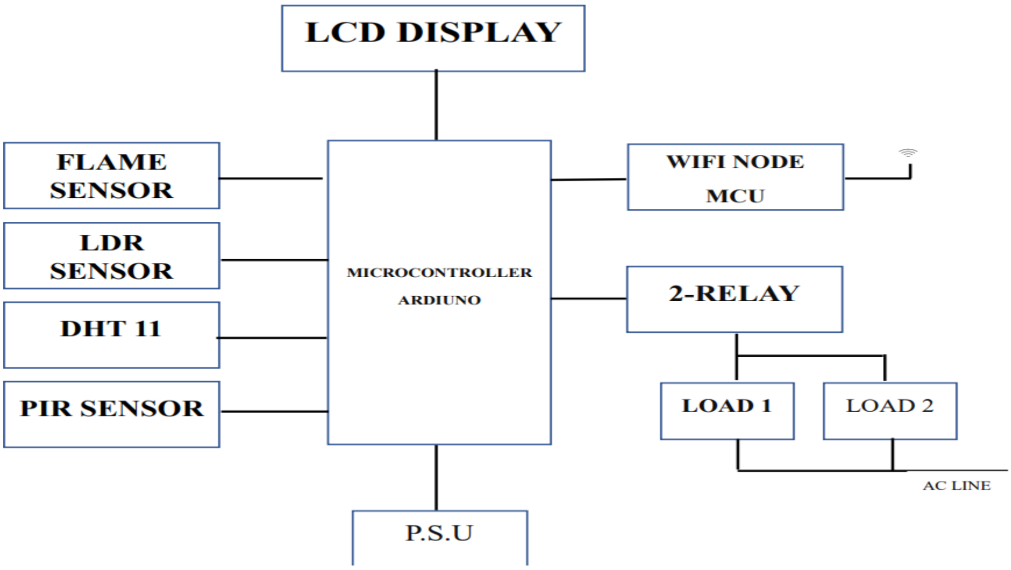


Figure 3.1: System Architecture of the Proposed Project

During the development of the smart home automation system, multiple sensors adhering to the IEEE 2700-2017 standard were utilized to track environmental parameters and improve system responsiveness. These sensors function as input components, capturing real-world data and transmitting it to the main controller to support intelligent system behavior and automation decisions

The smart home system employs several essential sensors to monitor environmental and security-related parameters:

- Passive Infrared (PIR) Sensor: This sensor detects motion within a specific range, making it valuable for enhancing home security and automating tasks like activating lights or sending alerts upon detecting unexpected movement.
- DHT11 Sensor: Used to measure ambient temperature and humidity, the DHT11 allows the system to monitor environmental conditions and manage heating, ventilation, and air conditioning (HVAC) systems accordingly.

- Light Dependent Resistor (LDR): This sensor measures the intensity of ambient light. It is primarily used to automate lighting, enabling lights to turn on or off depending on the level of natural daylight.
- Flame Sensor: Designed for fire detection, the flame sensor contributes to safety by providing early warnings in the event of a fire or flame-related hazard within the household.

All these sensors are physically connected to an Arduino UNO microcontroller, which serves as the main processing unit. As illustrated in Figure 3.1, the microcontroller receives input signals from the sensors, processes them, and actuates various loads (e.g., lights, fans, alarms) through a 2-channel relay module.

For remote monitoring and control, the system employs the NodeMCU ESP8266 module as the IoT communication interface. This module transmits sensor data via Wi-Fi to a custom-developed web dashboard, following the guidelines outlined in the IEEE 2144.1-2020 standard for secure IoT data handling. Through this web interface, users are able to monitor real-time environmental conditions and manage home appliances remotely from any device with internet access.

3.2 Objectives

The main goals of this project are structured as follows:

- Development of a Remote-Controlled Home Automation System:
  The project aims to create a home automation solution that utilizes Internet of Things (IoT) technology to provide real-time monitoring and control of household appliances. The system is designed to be accessible through any internet-enabled device, thereby offering users increased convenience, mobility, and flexibility in managing their home environments.
- Seamless Integration and Scalability:
  A key objective is to ensure smooth interoperability between smart devices and the central control unit. Using the Arduino platform as the core controller, the system is developed to be both cost-effective and easy to install. The

architecture must support user-friendly interfaces and allow for straightforward expansion to accommodate additional devices as needed.

- Intelligent Device Management through Sensor Analytics:
  The system will leverage real-time sensor data to enable automated responses to environmental conditions. This capability allows for dynamic adjustment of appliances, resulting in improved energy efficiency and enhanced safety within the home.
- System Testing and Performance Evaluation:
  After the hardware and software components—including sensors, microcontrollers, communication modules, and the user interface—have been fully integrated, the system will undergo extensive testing under simulated real-life conditions. Using mobile devices to send commands and interact with the system, various scenarios will be tested to validate functionality, stability, and response accuracy.

  During the testing phase, continuous data logging from sensors will support performance analysis. Key aspects such as device communication reliability, real-time responsiveness, and power consumption efficiency will be evaluated. Any irregularities or malfunctions identified during this phase will inform system refinement and optimization.

- Designing a Cost-Effective and Functional Smart Home System:
  An overarching objective of this project is to deliver a budget-friendly yet fully functional home automation solution. The system should maintain high standards of performance while remaining affordable for typical households.

While commercially available home automation systems are often associated with high installation and maintenance costs, this project investigates more affordable solutions by leveraging open-source hardware such as the Arduino UNO and NodeMCU, along with freely accessible software tools. These alternatives aim to deliver dependable performance and core functionality without incurring the significant expenses typical of proprietary systems.

By relying on commonly available components and a simplified system architecture, the financial burden is significantly reduced. This approach makes smart home technology more feasible and accessible, particularly for individuals in low- and middle-income regions, where affordability is a key consideration for adoption.

Designing a Simple, Safe, and Inclusive System for Elderly and Differently-Abled Users

Another important goal of this project is to ensure the smart home system is easy to use, secure, and designed with inclusivity in mind. Special consideration is given to older adults and individuals with physical disabilities. The system's interface—whether through a mobile app or voice-command functionality—is built to be clear and easy to operate, even for users with limited technical background.

In addition, strong security measures are integrated into the system to safeguard user data and prevent unauthorized access. These protections help preserve privacy and ensure safe control over smart devices. By combining accessibility with robust security, the project aims to offer a solution that is both technologically effective and socially inclusive.

3.3 Methodology

As we progress into the 21st century, the interaction between humans and computers has evolved significantly, breaking traditional boundaries and entering new dimensions. In today's technology-driven world, smartphones have become an integral part of everyday life, functioning far beyond simple communication tools.

This project aims to enhance the control of home appliances using smartphones. In traditional household setups, most devices are operated manually using physical switches. Such systems not only pose electrical safety risks but are also prone to energy wastage due to human error or inefficiencies.

The proposed solution introduces a smart control system that connects home appliances to a centralized intelligent circuit. This circuit communicates with the user's smartphone, allowing remote control and monitoring of home devices. By using a smartphone, users can turn devices on or off from any location, ensuring both convenience and energy efficiency.

Moreover, the system is designed to be more secure and reliable compared to traditional methods. It reduces the risks associated with manual operation and enables smart energy management, making it suitable for modern living.

3.4 Applications

Support for elderly and individuals with physical limitations: The system is particularly advantageous for users with mobility challenges, as it enables them to manage household appliances with minimal physical effort.

Remote monitoring and operation: The system allows users to access and control home devices from distant locations, enhancing both convenience and home security.

3.5 IoT-Based Smart Home Automation Framework

Creating a smart home automation system with Internet of Things (IoT) technology involves a structured and thoughtful design process. This process must take into account key aspects such as selecting appropriate hardware, developing suitable software, ensuring reliable connectivity, achieving seamless system integration, and implementing robust security measures.

3.6 Hardware Elements

The setup begins with the deployment of fundamental hardware components. Various types of sensors—such as those for detecting motion, measuring temperature, monitoring light levels, and identifying gas leaks—are installed to track real-time environmental conditions inside the home. Microcontrollers like the Arduino, ESP32, or Raspberry Pi function as the processing units, interpreting the data received from the sensors and executing commands accordingly.

Actuation devices such as relays, motors, and solenoids are included to physically operate appliances and systems within the household, including lighting fixtures, fans, and smart door locks. These hardware elements form the basis of the automation process, allowing the system to perform daily functions automatically, thus improving user comfort, optimizing energy use, and enhancing overall home security.

Reliable device-to-device communication is essential for effective IoT-based automation. In accordance with the IEEE 2413-2019 standard, achieving interoperability across different domains requires a well-defined network architecture. Connectivity options such as Wi-Fi, Bluetooth, Zigbee, and Z-Wave are chosen based on specific application needs, including coverage range, energy consumption, and data transmission speed.

The core intelligence of a smart home system is embedded in its software framework, which includes:

- Firmware installed on microcontrollers that interprets sensor inputs and executes corresponding control actions.
- A user interface, accessible via a mobile application or web dashboard, that allows users to manage and monitor household devices in real time through an intuitive and user-friendly platform.

Collecting sensor data alone does not maximize system efficiency. Data analytics is applied to interpret the collected information, revealing usage trends and supporting advanced functions like energy optimization, predictive system maintenance, and improved security measures.

A robust smart home system must be compatible with a variety of external platforms and services. Integration with virtual assistants such as Google Assistant and Amazon Alexa, as well as automation services like IFTTT, allows for rule-based automation and unified control across a wide range of smart devices.

Ensuring system security is a top priority in IoT-enabled smart homes. Protection against cyber threats—including unauthorized access and data breaches—requires secure communication protocols, encrypted data transmission, multi-layer authentication, and regular software updates. These measures safeguard user privacy and maintain the integrity of system operations.

Post-development, the system must undergo comprehensive testing to validate functionality, reliability, and security. Testing types include:

- Functional testing (does each feature work?)
- Performance testing (how well does the system handle loads?)
- Security testing (is the system protected from intrusion?)

Furthermore, ongoing maintenance, including software and firmware updates, is necessary to ensure the system remains secure, efficient, and compatible with emerging technologies.

Designing and implementing a smart home automation system using IoT is a multifaceted process requiring detailed attention to hardware selection, network design, software development, analytics, and security. By thoughtfully integrating each of these components, it is possible to develop a robust, intelligent home environment that offers convenience, energy savings, and enhanced quality of life.

3.7 Future Prospects of Smart Home Automation

Advancing Smart Homes: A Vision for the Next Generation of Intelligent Living

The evolution of smart homes is being driven by a digitally connected generation, fueled by continuous advancements in automation, IoT, and artificial intelligence (AI). What was once considered futuristic is now becoming part of everyday life—where voice commands and intelligent automation are reshaping how people interact with their living spaces. These technologies are at the heart of next-generation smart home systems, offering seamless control over lighting, climate, appliances, and security.

Enhanced Efficiency, Control, and Personalization through AI Artificial intelligence is expected to play a major role in elevating user comfort and system automation. It will allow centralized control of key aspects of the home environment—such as lighting, sound, climate, and security—through intelligent learning algorithms. By understanding user habits and preferences, AI-driven systems will be capable of delivering customized responses automatically, reducing the need for manual interaction. Furthermore, AI will strengthen home security by enabling real-time threat detection and instant alert notifications.

The increasing popularity of voice-controlled and app-based home automation has made integration with smart assistants like Amazon Alexa, Google Assistant, and Apple Siri a standard feature. Major technology providers are investing heavily in expanding the IoT landscape, leading to faster, more reliable, and highly interactive smart ecosystems. For instance, platforms such as Google Home now support features like proactive reminders, hands-free calls, and integration with popular music services including Spotify, SoundCloud, and Deezer. Devices like Google Home Mini and Home Max further enhance the ecosystem, enabling users to create a more connected and responsive living environment.

4 HARDWARE OVERVIEW

4.1 System Specifications

Required Hardware Components

Arduino UNO – Serves as the main microcontroller for managing sensor input and controlling output devices.

NodeMCU (ESP8266) – Facilitates Wi-Fi communication and acts as the IoT interface for remote monitoring and control.

DHT11 Sensor – Used to measure ambient temperature and humidity levels for environmental monitoring.

LCD Display – Displays real-time data and system status, enhancing user interaction.

PIR (Passive Infrared) Sensor – Detects motion, contributing to security and automation functions.

Flame Sensor – Identifies the presence of fire or flame for early hazard detection.

LDR (Light Dependent Resistor) Sensor – Measures light intensity, enabling automatic lighting control based on ambient conditions.

2-Channel Relay Module – Allows switching of high-voltage appliances using low-voltage control signals from the microcontroller. Connecting Wires – Provide electrical connections between components.

Power Supply Unit – Supplies regulated power to all modules and sensors in the system.

Required Software Tools

The development and implementation of the smart home automation system require the following software components:

- Embedded C – Used for programming microcontrollers, enabling interaction with sensors and actuators.
- Arduino IDE (Version 1.8) – The primary development environment used to write, compile, and upload code to the Arduino and NodeMCU boards.

- PHP and MySQL – Utilized for developing the web-based dashboard and backend database, allowing for data storage, user interaction, and remote device control through a browser interface.

4.2 Components Used

Arduino Uno

The Arduino Uno is a small, single-board microcontroller widely used for developing interactive electronic projects. It is based on the Atmega328 microcontroller and incorporates key features such as a USB port for programming, a power supply connector, six analog input channels, a 16 MHz ceramic resonator, an ICSP (In-Circuit Serial Programming) header, and a manual reset button. The board is programmed using the Arduino IDE, which is compatible with most standard desktop and laptop computers. The IDE supports programming in both C and C++, making it accessible for developers at various skill levels.

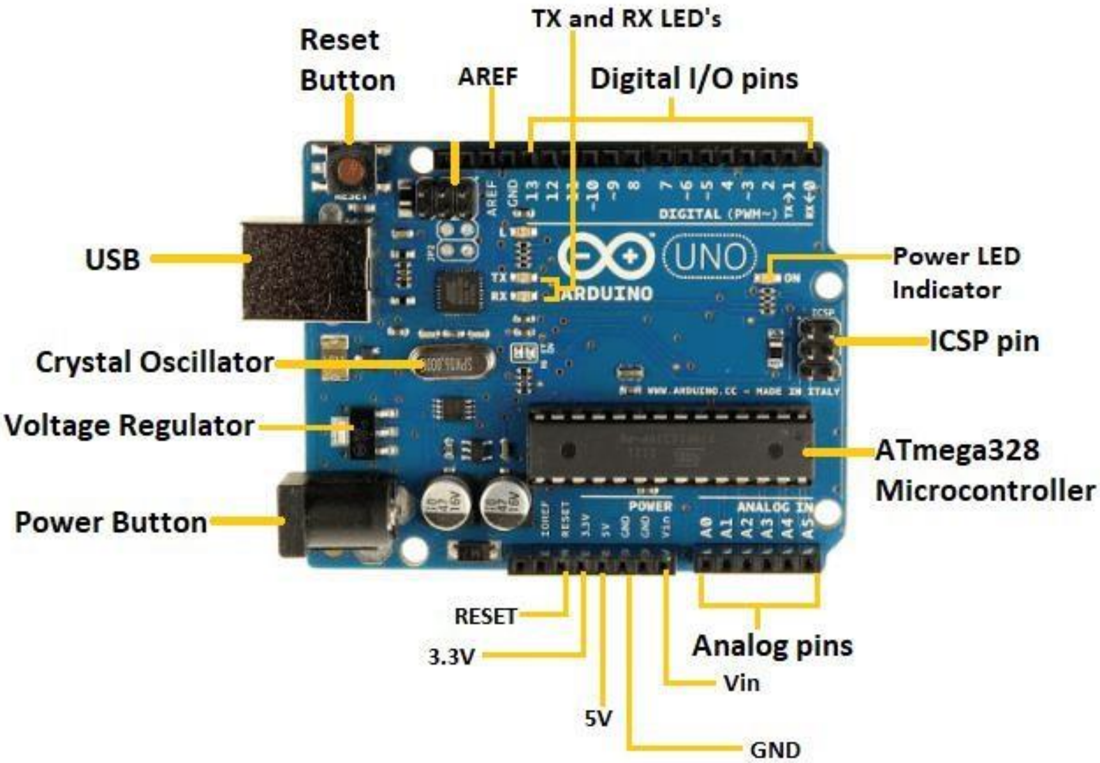


Figure 4.1: Overview of the Arduino Uno Microcontroller Board

Microcontroller: ATmega328-based Arduino Uno

Operating Voltage: 5V

Recommended Input Voltage: 7–12V

Input Voltage Range (Absolute Limits): 6–20V

Digital I/O Pins: 14 pins (including 6 with PWM functionality)

Analog Input Pins: 6

Maximum Current per I/O Pin: 40 mA

Maximum Current for 3.3V Pin: 50 mA

Flash Memory: Includes bootloader; total 32 KB

SRAM: 2 KB

EEPROM: 1 KB

Clock Frequency: 16 MHz

NodeMCU Development Board

NodeMCU originally refers to the firmware developed for the ESP8266 Wi-Fi System-on-Chip (SoC), rather than the hardware itself. This firmware is written in the Lua scripting language and is derived from the eLua project, utilizing the Espressif Non-OS SDK. It includes several open-source libraries, such as lua-cjson for JSON handling and SPIFFS for embedded file system support.

Designed specifically for the ESP8266 platform, NodeMCU provides a high-level programming environment that simplifies the development and prototyping of Internet of Things (IoT) applications. Alongside the firmware, the term "NodeMCU" also refers to the popular open-source development board that incorporates the ESP8266 module. This board features a built-in CP2102 USB-to-TTL serial converter, enabling seamless programming and serial communication through a standard USB interface. Its affordability and ease of use make it an ideal choice for IoT-based smart home automation projects.

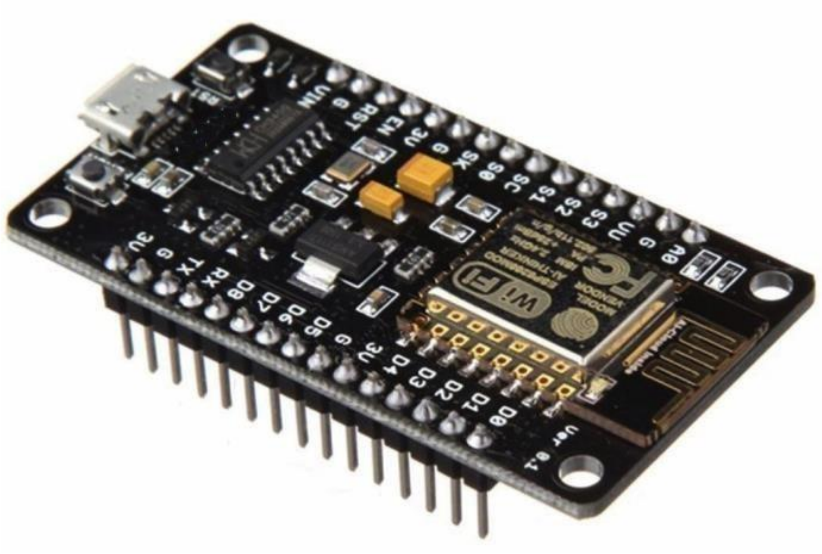


Figure 4.2: NodeMCU Wi-Fi Development Board

The NodeMCU board features a compact, breadboard-compatible design, making it ideal for prototyping and educational applications. Its small footprint and USB-powered interface enable convenient programming and power supply through a standard micro USB cable. This enhances portability and ease of use, making the board well-suited for developers and hobbyists engaged in wireless IoT-based projects.

PIN NAME ON NODE MCU DEVELOPMENT KIT	ESP8266 INTERNAL GPIO PIN NUMBER	PIN NAME ON NODE MCU DEVELOPMENT KIT	ESP8266 INTERNAL GPIO PIN NUMBER
0 [*]	GPIO16	**7**	GPIO13
1	GPIO5	**8**	GPIO15
2	GPIO4	**9**	GPIO3
3	GPIO0	**10**	GPIO1
4	GPIO2	**11**	GPIO9
5	GPIO14	**12**	GPIO10
6	GPIO12		

Figure 4.3: NodeMCU Pin Index to GPIO Mapping Diagram

LCD Display

Liquid Crystal Display (LCD) technology is widely used in laptops, handheld electronics, and various compact devices due to its lightweight design and ability to support slim screen profiles—making it a preferred alternative to traditional cathode ray tube (CRT) displays.

Unlike plasma displays, which rely on individual fluorescent elements to illuminate each pixel, LCDs function by manipulating light using liquid crystals. When an electric current is applied, these crystals rotate polarized light, thereby controlling the visibility of each pixel. This mechanism allows precise control over the display, enabling the generation of images and text by selectively activating or deactivating pixels as required.

Figure 4.4: 16x2 Alphanumeric LCD Display Module

LCD, short for Liquid Crystal Display, is a widely adopted technology used to present text, numerical data, and symbols in a variety of electronic devices. These displays are available in standard formats such as 8x1, 8x2, 10x2, 16x1, 16x2, 16x4, 20x2, 20x4, 24x2, 30x2, 32x2, and 40x2—where the first digit represents the number of characters per line and the second refers to the total number of display lines.

Several leading electronics companies, including Philips, Hitachi, and Panasonic, manufacture LCD modules customized for various applications. Despite differences in size and design, most alphanumeric LCDs operate using the same basic principles. They can display a range of characters, including letters, digits, standard ASCII symbols, and special characters.

These displays typically feature a standardized pin configuration—either 14 pins (numbered 0 to 13) or 16 pins (numbered 0 to 15)—making them broadly compatible with microcontrollers and simplifying the programming and interfacing process across different models.

Alphanumeric LCD modules are widely used across various electronic applications, including palmtop computers, word processing systems, photocopiers, point-of-sale (POS) terminals, medical diagnostic devices, mobile phones, and other digital platforms that require textual or symbolic display outputs.

One common example is a 16x2 LCD module designed specifically for use with E-blocks educational and development systems. This module features two rows, each capable of displaying 16 characters, and connects via a 9-pin D-type connector. This configuration ensures compatibility with most E-blocks I/O ports. The display communicates using a serial data protocol, with communication specifications and programming instructions provided in the associated user guide.

This module operates on a 24V DC power supply. However, it is crucial to ensure that certain input lines do not exceed 5V, as this could lead to irreversible damage. The required 24V is typically delivered either by an E-blocks multi-programmer or a regulated external power source designed for 24V output.

More advanced alphanumeric LCDs—such as 24 x 8 dot-matrix intelligent displays—are capable of rendering up to 224 distinct characters and symbols. The exact number and types of characters supported depend on the model and manufacturer. Detailed information regarding supported characters, connection requirements, and electrical specifications can usually be found in the product’s technical datasheet or user manual, often in dedicated sections (e.g., pages 7 and 8).

Applications

Sensor Data Display: Shows real-time readings from temperature, humidity, motion, and other environmental sensors in smart home systems.

System Status Indication: Used in embedded systems to indicate device status, such as power on/off, error messages, or operational states.

Smart Appliances: Displays settings, timers, or modes in home appliances like washing machines, microwaves, and air conditioners.

Security Systems: Shows alerts or messages related to motion detection, fire, or unauthorized access in alarm systems.

Educational Tools: Commonly integrated into microcontroller-based student projects for learning programming and interfacing.

Medical Devices: Used to show vital information on devices like digital thermometers, heart rate monitors, and portable diagnostic tools.

Point-of-Sale (POS) Systems: Displays transaction details in retail billing machines and other commercial equipment.

Industrial Panels: Serves as a visual interface for control units and machinery in factories.

DHT11 Sensor

The DHT11 is a widely used digital sensor designed to measure both temperature and relative humidity. It integrates a capacitive humidity sensor and a thermistor to deliver accurate readings of environmental conditions.

Relative humidity represents the ratio of current atmospheric moisture to the maximum moisture the air can retain at a given temperature, typically expressed as a percentage. The DHT11 measures humidity by detecting changes in electrical resistance caused by moisture levels. Simultaneously, it gauges ambient temperature with its internal thermistor.

This sensor offers a digital output, simplifying its integration with microcontrollers such as Arduino and NodeMCU. It operates within a temperature range of 0°C to 50°C and a humidity range of 20% to 90% with adequate precision for many home automation and weather-related applications.

Thanks to its compact design, affordability, and ease of use, the DHT11 is commonly implemented in projects involving environmental monitoring, smart HVAC systems, indoor air quality regulation, and other IoT-based smart home solutions.

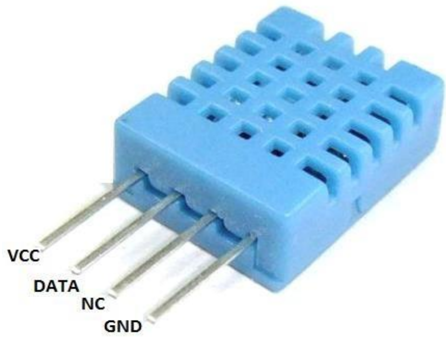


Figure 4.5: DHT11 Temperature and Humidity Sensor Module

The DHT11 is a digital sensor specifically developed to measure ambient temperature and relative humidity. It detects environmental changes and converts them into corresponding digital signals for further processing by a microcontroller. The sensor's internal structure includes two conductive plates separated by a polymer film that is sensitive to moisture. As this film absorbs water vapor from the surrounding air, the resulting change in moisture levels alters the electrical potential between the plates. These changes are then processed to produce accurate digital readings.

The DHT11 operates using hygrometric sensing principles, which are typically categorized as either capacitive or resistive:

- Capacitive humidity sensing involves detecting variations in the dielectric properties of the polymer layer, which change in response to absorbed moisture, thereby altering the sensor's capacitance.
- Resistive humidity sensing, alternatively, measures changes in electrical resistance. In this method, a comb-patterned electrode, often made from precious metals like gold or ruthenium oxide, is coated with a humidity-sensitive polymer. The film contains mobile ions, and as humidity levels change, so does the concentration of free ions, resulting in variations in electrical impedance.

These sensing techniques allow the DHT11 to provide dependable and reasonably accurate environmental data. Because of its affordability, simplicity, and ease of integration, it is widely used in applications such as weather monitoring systems, HVAC control, smart homes, and environmental data logging.

Key Features:

- Operating Voltage: Operates effectively at a 5V DC supply.
- Signal Output: Provides analog output in the range of 0 to 5 volts.
- High Accuracy and Efficiency: Engineered for reliable and precise performance in environmental monitoring applications.
- Long-Term Operational Stability: Designed to maintain consistent functionality and accuracy over extended periods of use.
- Narrow Tolerance Range: Offers minimal variation between units, ensuring consistent measurements.
- Cost-Effective Solution: Provides a balance between performance and affordability, making it suitable for budget-conscious projects.

Flame Detection Module

A flame sensor is a dedicated device developed to detect the presence of fire or flame and to initiate an appropriate system-level response. The reaction triggered by the sensor depends on the operational context and the predefined safety protocols. Typical responses may involve sounding visual or audible alarms, automatically shutting off fuel lines—such as those supplying natural gas or propane—and activating fire suppression mechanisms to contain or extinguish the fire.

These sensors are essential components in high-risk environments, including industrial plants, power stations, and chemical facilities, where rapid fire detection is crucial for minimizing hazards and protecting human life. Flame sensors function by identifying specific wavelengths of light associated with combustion, typically through ultraviolet (UV), infrared (IR), or combined UV/IR detection technologies.

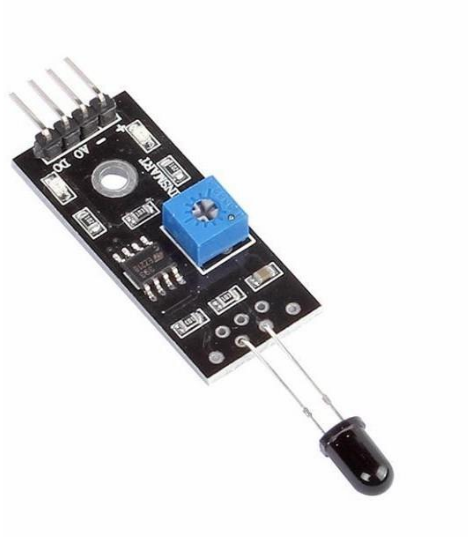


Fig. 4.6 Flame Detection Sensor

A flame sensor is a specialized device designed to detect the presence of fire or flame and initiate a corresponding response. In addition to identifying actual flames, the sensor is sensitive to light sources within the near-infrared spectrum, typically ranging from 760 nm to 1100 nm. Its detection range can extend up to 100 cm, depending on the flame's intensity and size.

The sensor supports both digital and analog outputs, offering flexibility for integration into various fire detection systems, including flame alarm circuits and autonomous fire-fighting robots. Due to its heightened responsiveness to visible and near-infrared light, it is particularly effective in indoor and controlled environments where reliable flame detection is essential.

Structurally, the module features a compact output interface panel that can be easily connected to the input/output (I/O) pins of a microcontroller. To prevent thermal damage and ensure stable performance, it is advised to install the sensor at an approximate distance of 80 cm from the flame. Higher flame intensities can allow for extended detection distances.

Additionally, the sensor is equipped with a built-in black LED indicator that displays both power status and output signal activity, enabling real-time monitoring. The analog signal can also be used for analog-to-digital (A/D) conversion, enhancing the accuracy and resolution of flame detection in precision-based applications.

Key Features:

- Operating Voltage: Compatible with a supply range of 3.3V to 5V DC
- Current Consumption: Operates with a typical current of approximately 15 mA
- Digital Output: Provides a logic-level signal ranging from 0V to 5V, with an adjustable threshold via an onboard preset
- Analog Output: Produces an analog voltage between 0V and 5V, proportional to the intensity of infrared radiation emitted by a flame
- Detection Range: The sensor detects flame within an angular field of approximately 60°, demonstrating high sensitivity to the flame's spectral signature
- **Stability:** Integrated with the LM393 comparator chip, which ensures consistent and reliable output readings

Passive Infrared (PIR) Sensor

A Passive Infrared (PIR) sensor is an electronic component designed to detect infrared (IR) radiation naturally emitted by objects within its sensing range. These sensors are primarily utilized in motion detection systems, where they identify variations in IR radiation caused by the movement of people, animals, or other heat-emitting bodies.

Commonly referred to as motion detectors, PIR sensors are widely implemented in both consumer and industrial environments. Due to their simplicity and effectiveness, they are frequently integrated into DIY electronics and prototyping platforms such as Arduino, enabling the development of custom motion-activated solutions like automatic lighting, intrusion alarms, and security systems.

From a technical perspective, PIR sensors operate as pyroelectric devices, meaning they are capable of detecting fluctuations in infrared energy. The term "passive" signifies that the sensor does not emit infrared radiation itself but instead senses thermal energy changes emitted by objects in its vicinity. The sensor relies on a pyroelectric material that reacts to infrared level changes, typically triggered by motion across the sensor's field of detection.

PIR sensors are highly regarded for their excellent sensitivity, dependable performance, and compatibility with ultra-low voltage operation, which makes them particularly suitable for battery-operated and energy-efficient automated control systems. These sensors are frequently employed in a variety of applications, including motion-activated lighting, intrusion detection alarms, and smart energy-saving devices. Their ability to detect motion and trigger responses automatically eliminates the need for manual input, enhancing both convenience and system efficiency.

Fig. 4.7: Passive Infrared (PIR) Motion Detection Sensor

From a practical perspective, all objects with a temperature above absolute zero (−273.15°C or 0 Kelvin) emit thermal energy in the form of infrared (IR) radiation—a core concept in thermodynamics. This radiation lies within the infrared region of the electromagnetic spectrum, which is invisible to the human eye but can be sensed using appropriate electronic devices.

This fundamental principle underpins the operation of Passive Infrared (PIR) sensors. Unlike active sensors, PIR sensors do not emit radiation; instead, they function by passively detecting infrared energy emitted by surrounding objects, particularly warm bodies such as humans and animals. When such an object enters the sensor's detection zone, it causes a variation in infrared radiation levels, which the sensor interprets as motion.

Because they utilize naturally emitted IR radiation, PIR sensors offer a non-intrusive, low-power solution for motion detection. As a result, they are widely used in applications such as security systems, automated lighting controls, and other intelligent environmental management systems.

Key Features:

- Detection Range: Covers an angle of less than 120° with an effective sensing distance of up to 7 meters
- Operating Temperature: Functions reliably within a temperature range of −15°C to +70°C
- Response Time: Fast lock time of approximately 0.2 seconds
- Power Consumption: Operates at a low current draw of around 65 mA, suitable for energy-efficient applications

Light-Dependent Resistor (LDR) Sensor

A Light-Dependent Resistor (LDR), also referred to as a photoresistor or photoconductive cell, is a type of variable resistor whose resistance is influenced by the intensity of light falling on its surface. As the illumination increases, the resistance of the LDR decreases—an effect known as photoconductivity. This property enables LDRs to function effectively in systems that require light detection or level monitoring. Common applications include automatic lighting control, ambient light sensors, and photographic exposure meters.

4.8: Light-Dependent Resistor (LDR) Sensor Module

A Light Dependent Resistor (LDR) functions based on the principle of photoconductivity—a phenomenon in which a material's electrical conductivity increases in response to light exposure. When light photons strike the surface of the LDR’s semiconductor material, their energy can excite electrons from the valence band into the conduction band, provided that the energy of the incident photons exceeds the semiconductor’s band gap.

As the number of incident photons with sufficient energy increases, more electrons are promoted to the conduction band, resulting in a higher concentration of charge carriers. This increase in free carriers leads to enhanced current flow through the material when part of a closed circuit, thereby reducing the overall resistance of the LDR as light intensity rises.

In essence, an LDR is a light-sensitive semiconductor device whose resistance decreases with increasing light levels. Commonly referred to as photoconductors, photocells, or photoconductive cells, LDRs are typically composed of high-resistivity semiconductor materials that exhibit a significant drop in resistance upon illumination. These properties make them particularly suitable for light-detection applications, including automatic lighting systems, brightness sensors, and electronic control circuits.

Key Features:

- Operating Voltage: 5V DC input
- Output Type: Analog signal output
- Light Sensitivity: High responsiveness to variations in ambient light
- Output Voltage Range: Up to 5V depending on light intensity

Two-Channel Relay Module

A relay is an electrically controlled switching device that enables one electrical circuit to regulate another—often involving differing voltage or current levels. The most commonly used type is the electromagnetic relay, which utilizes an internal electromagnet to mechanically open or close a set of contacts. In contrast, solid-state relays perform the same function electronically, without any moving components.

Relays are particularly valuable in applications where a low-power control signal must manage a higher-power load or where a single controller needs to control multiple independent circuits. For example, in embedded systems and automation, a microcontroller can use a single I/O pin to control high-voltage devices such as motors, lights, or pumps through a relay.

Historically, relays were first introduced in long-range telegraph networks as signal repeaters, enabling the transmission of messages over greater distances. They were later fundamental to early telephone switching systems and computing devices, where they executed basic logic functions.

A standard relay consists of an electromagnet and a set of contacts. When a low-power current energizes the electromagnet, it generates a magnetic field that actuates the contact mechanism—either opening or closing the circuit depending on the relay’s configuration (normally open or normally closed).

Relays enable both electrical and mechanical switching across a wide range of applications, including industrial control systems, automotive electronics, and smart home automation setups.

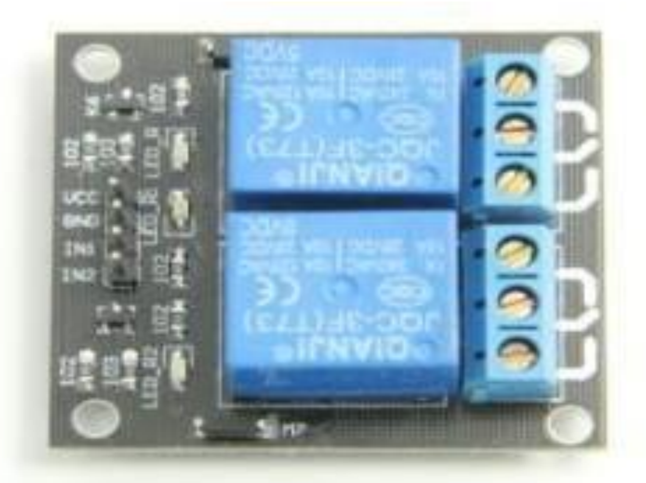


Figure 4.9: 2-Channel Electromechanical Relay Interface Board

Relays are commonly used to redirect electrical signals from a single source to one or more destinations, allowing centralized control over multiple devices. In high-demand scenarios, such as those involving electric motors or other high-power equipment, contactors are used. These are a specialized type of relay designed to manage significantly higher voltages and currents than standard relays.

At their core, relays function as electromechanical switches that are activated by an electric current. In a typical two-channel relay module, several essential sub-circuits work together to ensure stable and reliable operation:

Driver circuit: This section usually includes transistors, which act as electronic switches. When a control signal is applied, the transistor conducts, completing the circuit and energizing the relay coil to trigger the switching mechanism.

Power supply circuit: This provides the necessary voltage and current to operate the relay coils and associated components.

The isolation circuit is a crucial component in relay modules, typically implemented using diodes or optocouplers. Its primary function is to shield the microcontroller and driver transistors from voltage transients caused by the sudden collapse of the magnetic field when the relay coil is de-energized—an effect known as back electromotive force (back EMF).

The control signal required to activate the transistor is usually provided by a microcontroller unit (MCU), allowing precise and programmable control of the relay switching process. In this setup, each relay can independently control one or more external devices, making the system highly suitable for a wide range of applications, including home automation systems, industrial control panels, and robotic platforms.

Key Features

- Input Voltage: Operates at 12V DC
- Driver Unit: Equipped with a ULN2003A transistor array for efficient relay control
- Isolation Unit: Uses IN4007 diode for protection against voltage spikes and back EMF
- Switching Speed: Supports rapid switching operations
- Motor Control Capability: Enables both forward and reverse motor direction control

PSU: Conversion and Regulation of Electrical Power

A power supply unit (PSU) is an essential component in any electronic system, providing the necessary electrical energy to ensure proper functionality. In household devices such as televisions, printers, and audio systems, the PSU is responsible for converting high-voltage alternating current (AC) from the mains supply into low-voltage regulated direct current (DC), which is compatible with internal circuitry.

In computing systems, the PSU plays a critical role by converting AC power from a wall outlet into stable DC voltages required by various components, including the motherboard, storage devices, and peripheral hardware. Modern computers predominantly use switched-mode power supplies (SMPS) due to their higher efficiency, smaller form factor, and lower heat output compared to traditional linear power supplies. Depending on the model, PSUs may feature a manual voltage selector (e.g., 110V/220V) or an automatic input voltage adjustment system for compatibility with varying mains standards.

Power supplies are generally classified into two main types:

Linear Power Supplies reduce voltage through a step-down transformer and then convert it into DC using rectification and voltage regulation. While they offer low electrical noise and straightforward design, they are often larger and less efficient.

Switched-Mode Power Supplies (SMPS) utilize high-frequency switching to convert power, resulting in compact, lightweight, and energy-efficient solutions widely used in modern electronics.

An AC adapter—also referred to as an AC/DC adapter or converter—is a type of external PSU commonly supplied with portable or battery-operated devices. These adapters provide the required DC voltage externally, especially for devices lacking built-in conversion hardware. When used for recharging, they are often labeled as battery chargers.

The internal circuitry of external adapters is comparable to that of built-in power supplies, but the components are enclosed in an external housing. For instance, a standard 12V 1A AC adapter is frequently used to power devices such as Wi-Fi routers, LED lighting systems, and microcontroller development boards. Technical documentation often includes a pictorial representation of such adapters to guide users in proper and safe connections.

Figure 4.10: Standard 12V 1AMP Adapter for Electronic Devices

An adapter is an electronic component designed to facilitate compatibility between systems or devices that would otherwise be electrically or physically incompatible. Its primary function is to modify specific electrical characteristics—such as voltage levels, signal formats, or connector types—to enable proper communication or power transfer between different components.

Adapters serve diverse roles depending on the application. Some are used to convert electrical parameters, allowing devices with differing voltage requirements or signaling standards to interact. Others serve as physical interface converters, enabling connections between mismatched ports—for instance, adapting a USB Type-C connector to a USB Type-A interface.

In the realm of computing, the term "adapter" often refers to expansion hardware, such as graphics cards, sound cards, or network adapters, which are inserted into a motherboard's expansion slots. These components extend the functionality of a computer by introducing features not inherently available on the base system.

In summary, adapters are essential elements in modern electronics, ensuring interoperability and seamless integration across a wide range of consumer, industrial, and computing devices.

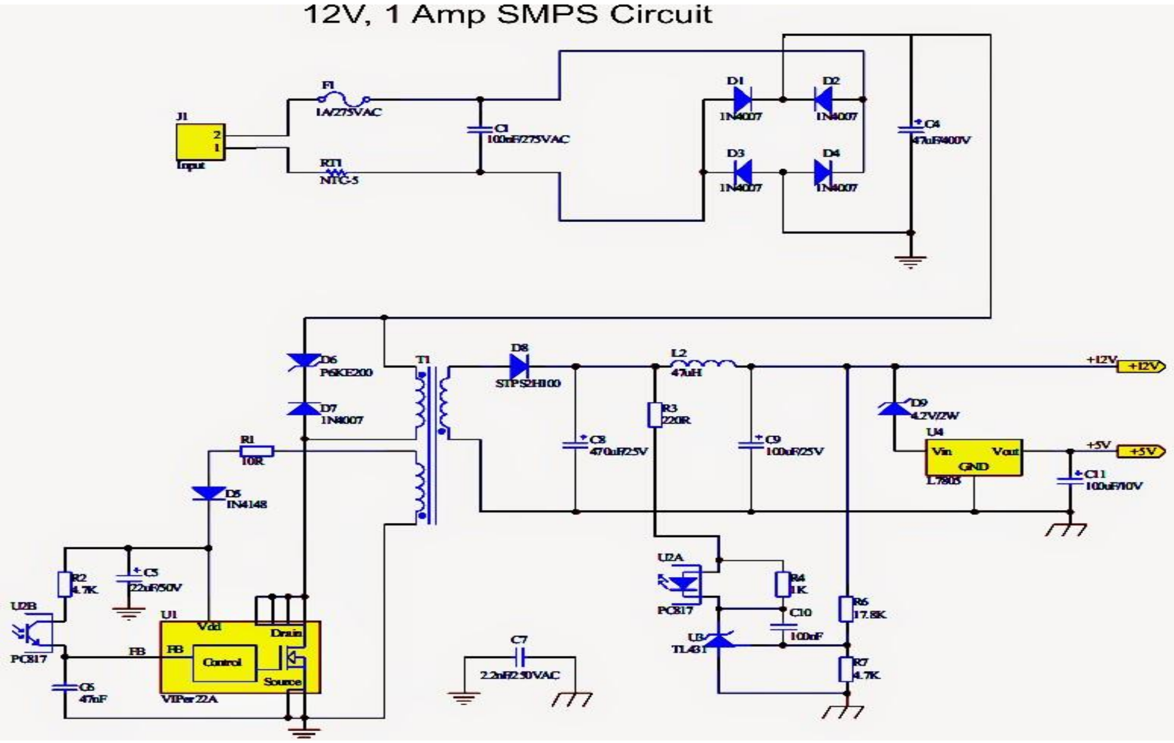


Fig. 4.11: Schematic Representation of Current Flow in a 12V DC Adapter

Figure 4.11 illustrates the internal current flow within a 12V power adapter, emphasizing its function in providing regulated DC power to electronic systems. Power adapters come in various forms, each tailored to meet specific functional requirements. A commonly encountered type is the plug adapter, which allows devices with power plugs designed for one region to connect to sockets with different physical formats—enabling mechanical compatibility without modifying voltage or frequency.

A more sophisticated form, the AC adapter (often referred to as a power adapter or recharger), acts as a compact external power supply. It converts standard AC mains voltage (typically 220–230V AC) into a stable DC output, such as 12V, suitable for powering electronics like routers, Arduino boards, LED lighting, and similar low-voltage devices. These adapters generally incorporate key components such as rectifiers, filters, and voltage regulators to maintain output stability and protect connected devices.

In addition to physical conversion, some adapters handle electrical or signal translation. For example, in computing environments, serial port adapters convert between different connector formats—such as 25-pin to 9-pin RS-232—while preserving signal integrity and voltage compatibility.

Overall, adapters are crucial enablers of interoperability, allowing electronic systems to function reliably across a range of power standards, interface types, and application domains.

Key Features**:**

- Output Current: 1 Ampere
- Input Voltage: 220–230V AC
- Output Voltage: 12V DC
- Cost-Efficiency: Offers reduced operating costs and enhanced system value
- Operational Value: Supports both front-end and back-end integration
- Data Accuracy: Ensures access to current, consistent, and reliable data
- Integration Support: Capable of generating adapter metadata in WSDL format using J2CA extensions

5 SIMULATED CIRCUIT DESIGN AND OUTPUT ANALYSIS

5.1 Simulation-Based Evaluation

The Arduino Integrated Development Environment (IDE) is a cross-platform, open-source software tool specifically designed for programming microcontroller-based systems, particularly those used in Arduino hardware. Serving as the primary interface between the user and the microcontroller, it enables efficient writing, compiling, and uploading of code to embedded devices. The platform adheres to established standards, including IEEE 1118.1-1990, ensuring structured and reliable communication between software and hardware components.

Compatible with Windows, macOS, and Linux, the Arduino IDE supports programming in C and C++, and provides a comprehensive collection of built-in libraries that facilitate interaction with external devices such as sensors, actuators, and communication modules. These libraries significantly reduce development time and complexity, promoting ease of use even in advanced embedded applications.

The IDE features a minimalist and intuitive user interface that includes:

A code editor for developing Arduino sketches,

A message area displaying compiler and upload notifications,

A project navigation panel,

And a serial monitor for real-time debugging and communication.

Overall, the Arduino IDE strikes a balance between simplicity and functionality, making it ideal for both beginners and experienced developers. Its open-source foundation and strong community support further enhance its value in educational, prototyping, and professional development contexts.

Arduino IDE Features:

- Capable of reading both analog and digital input signals directly through the software interface.
- Communicates with the microcontroller via a standard USB connection for code uploading and debugging.

- Supports multiple programming languages, including C, C++, and Java, providing flexibility in code development and compatibility across various project types.

5.2 Proteus Design Suite

Proteus Professional is a comprehensive Electronic Design Automation (EDA) software package designed to support the complete process of electronic circuit design, simulation, and printed circuit board (PCB) layout. Widely used in academic, research, and industrial environments, Proteus facilitates both the theoretical and practical aspects of electronic system development.

The suite consists of three primary modules:

Schematic Capture Module – Enables users to create and edit electronic circuit schematics through an intuitive graphical interface. It includes an extensive component library, allowing for the modeling of both basic and advanced electronic circuits.

PCB Layout Module – Translates schematics into PCB designs with tools such as manual routing, a powerful auto-router, and a design rule checker (DRC) to ensure compliance with manufacturing and electrical standards.

Simulation Module – Offers a high-performance simulation engine that allows virtual testing of circuits before physical implementation. This feature helps detect design errors early, significantly reducing the time and cost associated with hardware prototyping.

A key advantage of Proteus is its microcontroller simulation capability. The platform supports a wide range of programmable devices—including PIC, AVR, ARM, and Arduino—allowing users to simulate code execution in real-time alongside peripheral hardware behavior within the schematic.

Additionally, Proteus PCB Design is a streamlined version of the suite tailored specifically for PCB creation. While it includes essential features like auto-routing, multi-layer management, and rule checking, it does not support circuit simulation, making it suitable for projects focused solely on board design.

In summary, Proteus is a versatile and robust EDA solution that spans the full product development cycle. Its integration of schematic design, PCB layout, and embedded system simulation makes it especially valuable for students, educators, engineers, and hobbyists engaged in analog and digital circuit development.

Key Features of Proteus Design Suite

- Extensive Component Library: Offers access to over 15 million built-in and on-demand components, facilitating rapid design and simulation across a wide range of electronic systems.
- Integrated Reporting Tools: Includes a dedicated reporting module that supports comprehensive project documentation, aiding in design verification, collaboration, and record-keeping.
- Hybrid Circuit Simulation: Supports mixed-mode (analog and digital) circuit simulation with high precision, enabling accurate performance analysis and validation of complex electronic designs.

5.3 Fundamental System Structure

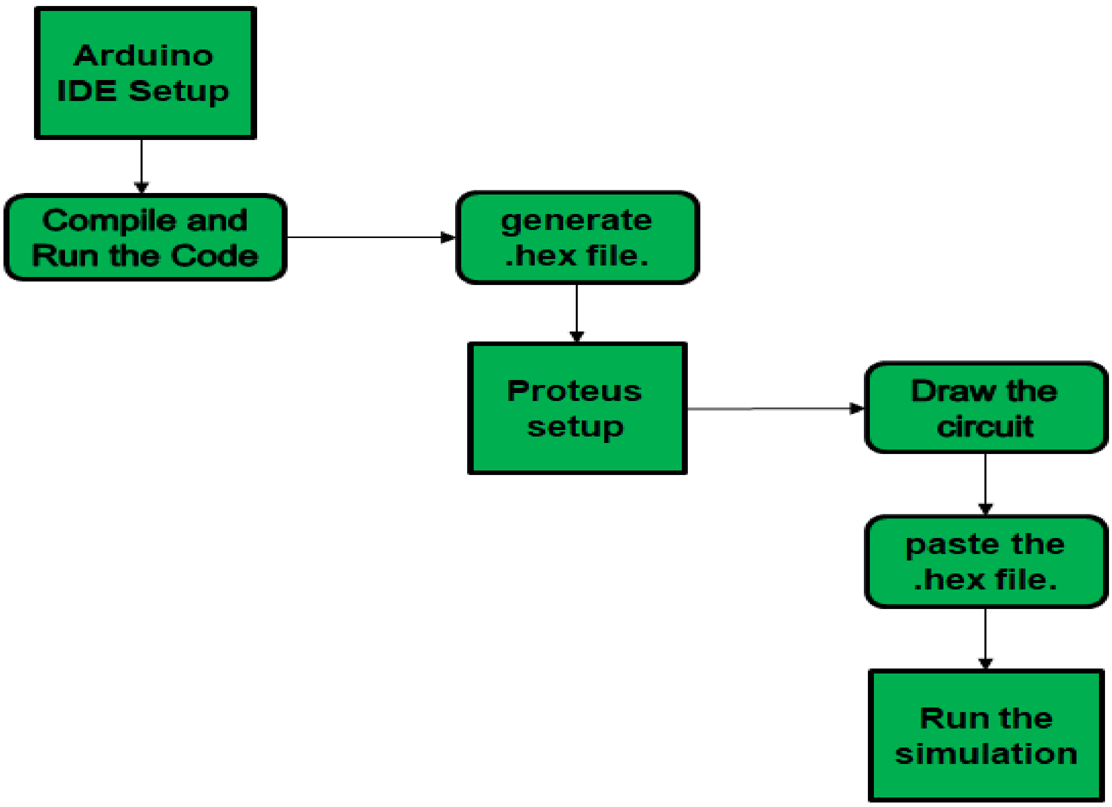


Fig. 5.1: Workflow of the System Simulation Process

5.4 System Operation

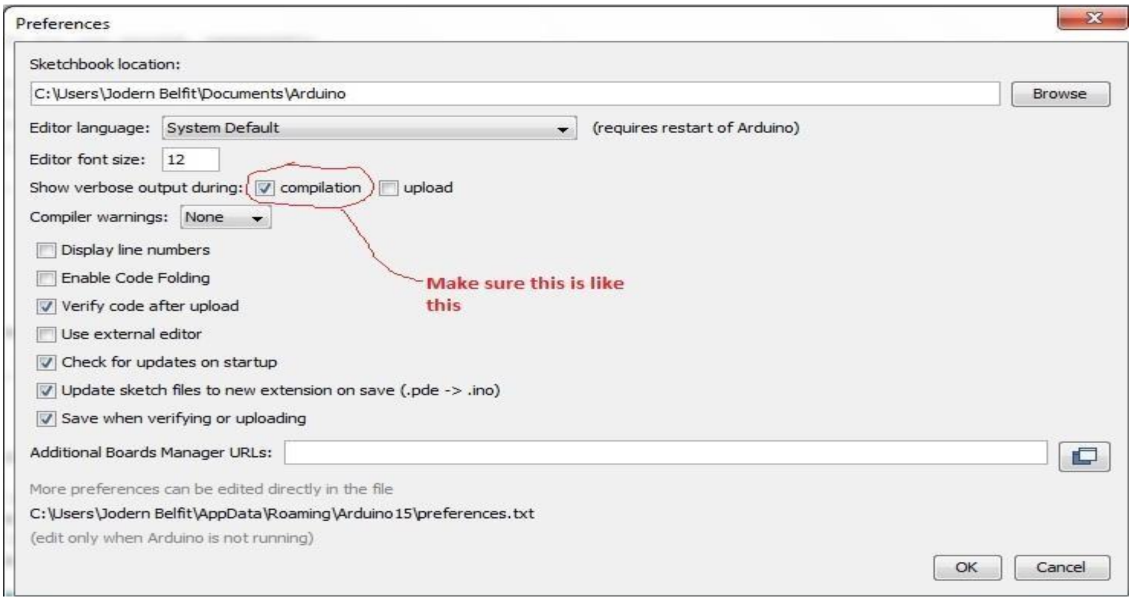


Fig. 5.2: Development Environment Setup in Arduino IDE

Before initiating the simulation in Proteus, it is essential to enable the “Compilation” option in the Arduino IDE. This can be done by navigating to File → Preferences and checking the “Show verbose output during: compilation” option. Enabling this ensures that a .hex file is generated after code compilation, which is required to load the program into the microcontroller within the Proteus environment. Without the .hex file, the simulated microcontroller will not execute the intended code.

Step 2: Configuring the Proteus Simulation Environment

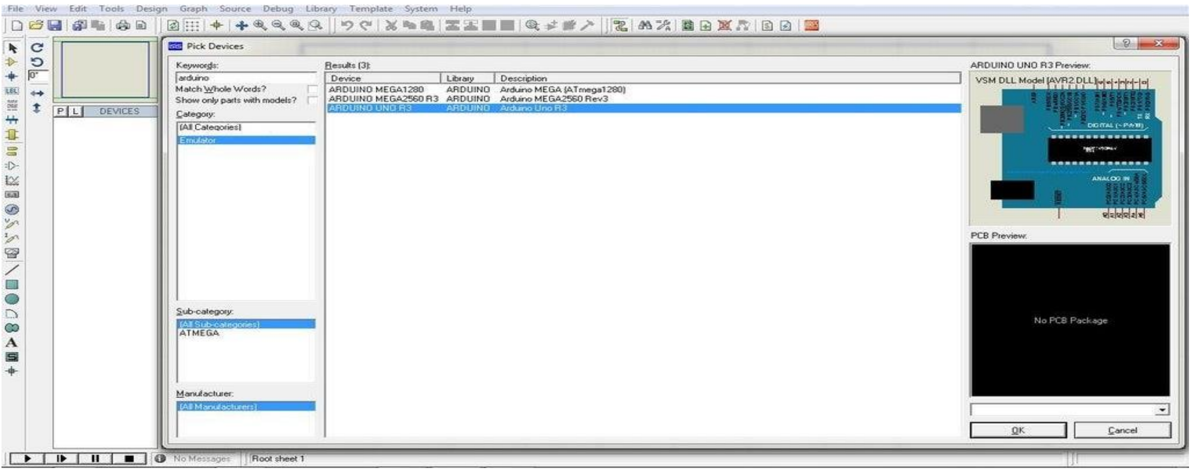


Figure 5.3: Initial Configuration of the Circuit in Proteus

Proteus is the software platform used for conducting electronic circuit simulations.

Before proceeding, ensure that Proteus is properly installed on your system. If the Arduino components are not available in the default library, you will need to manually add them. This involves downloading the necessary .rar archive, extracting its contents, and placing the extracted files into the **Proteus Library** directory.

For systems running Windows 10, the default library path is typically: C:\Program Files\Labcenter Electronics\Proteus 10 Professional\LIBRARY

Once the files are correctly placed, Arduino components will become accessible within the Proteus library for use in simulation projects.

Step 3: Arduino UNO programming

Figure 5.4 Step 3: Uploading Code to the Arduino UNO

To begin, launch the Arduino IDE and navigate to File > Examples > 01.Basics > Blink to open the built-in LED blinking example. Click the Verify button to compile the code. Once the compilation is complete, locate and copy the generated .hex file using Ctrl + C—this file will later be used in the Proteus simulation.

Step 4: Executing Arduino Code in Proteus

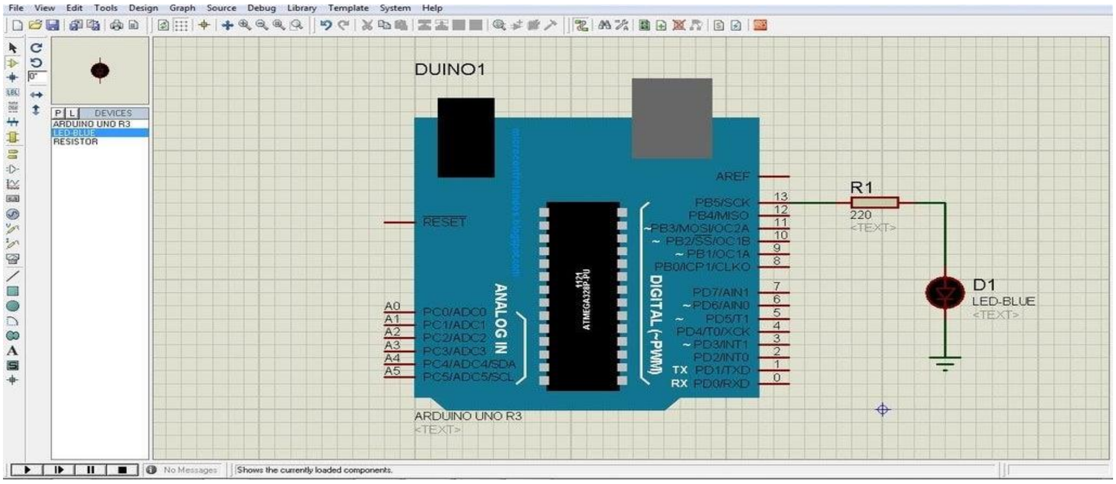
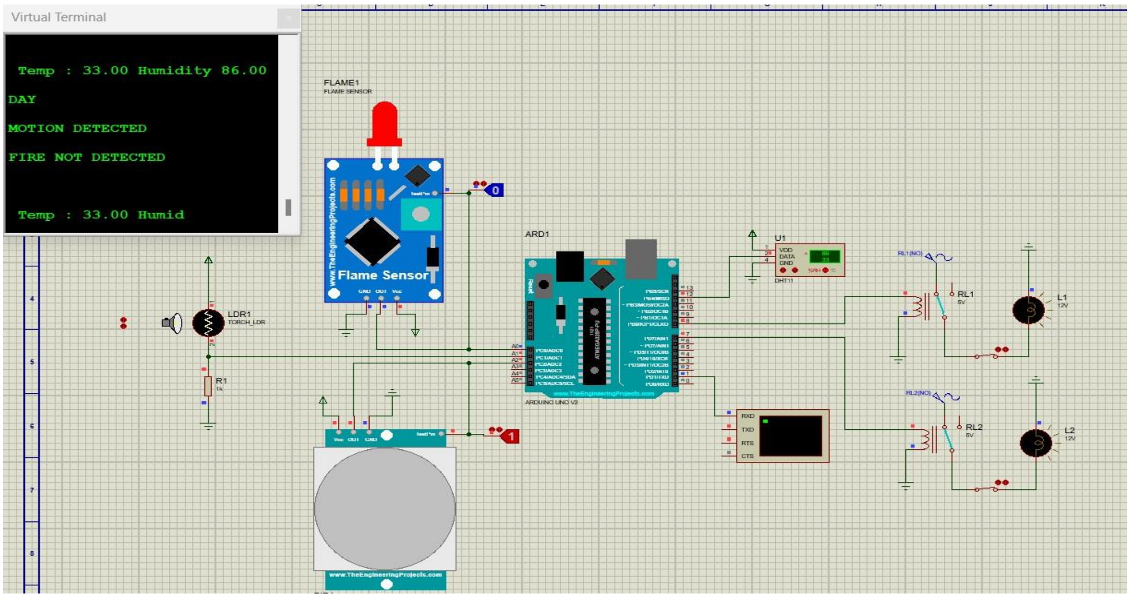


Fig. 5.5: Complete Circuit Configuration in Proteus

Launch the Proteus application and construct the circuit according to the provided schematic. To load the program, double-click the Arduino component within the workspace, then paste the previously copied .hex file into the Program File field. Once completed, initiate the simulation by clicking the Run button on the toolbar.

5.5 Simualtion output



5.1.4 Output from Simulated Circuit

Figure 5.6 illustrates the output generated from the simulation performed in the Proteus environment.

5.6 Embedded Program for Arduino IDE

#include <DHT.h> // Include the DHT sensor library to read temperature and humidity data

#define SENSOR_PIN 12   // Define the digital pin (D12) connected to the DHT11 sensor

#define SENSOR_TYPE DHT11 // Specify the type of the DHT sensor used

DHT climate(SENSOR_PIN, SENSOR_TYPE); // Create an instance of the DHT sensor class

void setup() {

Serial.begin(9600); // Begin serial communication at 9600 baud rate for monitoring

climate.begin();   // Initialize the DHT sensor

pinMode(7, OUTPUT); // Configure digital pin 7 as an output (e.g., for light or fan)

pinMode(8, OUTPUT); // Configure digital pin 8 as an output (e.g., for buzzer or second load)

}

void loop() {

// Read data from sensors

float temp = climate.readTemperature();   // Read temperature in °C

float hum = climate.readHumidity();   // Read humidity percentage

int lightLevel = analogRead(A1);   // Read light intensity from LDR sensor (A1)

int fireStatus = digitalRead(A0);   // Read flame sensor signal (A0), 1 = flame detected

int motionStatus = digitalRead(A2);   // Read PIR sensor signal (A2), 1 = motion detected

```
// Print temperature and humidity values to the Serial Monitor
Serial.print("Temperature: ");
Serial.print(temp);
Serial.print(" °C  |  Humidity: ");
Serial.print(hum);
Serial.println(" %");
Serial.println(""); // Spacer for readability

// Print day or night status based on LDR reading
if (lightLevel > 300) {
  Serial.println("Environment: DAY"); // Light level high → it's day
} else {
  Serial.println("Environment: NIGHT"); // Light level low → it's night
}

// Print motion detection status
if (motionStatus == 1) {
  Serial.println("Motion: DETECTED");
} else {
  Serial.println("Motion: NOT DETECTED");
}

// Print flame detection status
if (fireStatus == 1) {
  Serial.println("Flame: DETECTED");
} else {
  Serial.println("Flame: NOT DETECTED");
}
```

```
  Serial.println(""); // Blank line to separate output sections

  // Logic to control connected output devices based on sensor input
  if (fireStatus == 0) { // If no fire is detected
    if (lightLevel < 300 || motionStatus == 1) {
      digitalWrite(7, HIGH); // Turn ON output devices
      digitalWrite(8, HIGH);
    } else {
      digitalWrite(7, LOW);  // Turn OFF output devices
      digitalWrite(8, LOW);
    }
  } else {
    // If fire is detected, all devices are turned OFF as a safety precaution
    digitalWrite(7, LOW);
    digitalWrite(8, LOW);
  }

  // Final spacing and delay for loop refresh
  Serial.println("");
  Serial.println("");
  delay(1000); // Delay for 1 second before next loop cycle
}
```

5.7 Source code used for hardware:

```
#include <ESP8266WiFi.h>         // Library for Wi-Fi connection
#include <ESP8266HTTPClient.h>   // Library to handle HTTP communication
#include <WiFiClient.h>          // Basic Wi-Fi client functions
#include <ArduinoJson.h>         // JSON parsing library
#include <LiquidCrystal_I2C.h>   // Library for I2C LCD screen
```

```
LiquidCrystal_I2C lcd(0x27, 16, 2); // Create LCD object at I2C address 0x27 with 16x2 display

// Wi-Fi credentials and server URL
const char* ssid = "iot";
const char* password = "12345678";
const char* serverURL = "http://iotcloud22.in/2045_smarthome/post_value.php";

// Global instances for HTTP and client handling
WiFiClient client;
HTTPClient http;

// Sensor variables
int humidityValue, fireValue, temperatureValue, lightLevel, motionValue;
int inputLength, readCounter = 0;
int connectionFlag;
String inputBuffer = "";

void setup() {
  pinMode(D6, OUTPUT); // Output device 1
  pinMode(D5, OUTPUT); // Output device 2
  digitalWrite(D6, LOW);
  digitalWrite(D5, LOW);

  Serial.begin(9600);
  lcd.init();
  lcd.backlight();
```

```
  lcd.setCursor(0, 0);
  lcd.print(" IoT BASED SMART ");
  lcd.setCursor(0, 1);
  lcd.print(" HOME AUTOMATION ");

  // Connect to Wi-Fi
  WiFi.begin(ssid, password);
  Serial.println("Connecting to WiFi...");
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  Serial.println();
  Serial.print("Connected. IP Address: ");
  Serial.println(WiFi.localIP());

  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(" WiFi Connected ");
  lcd.setCursor(0, 1);
  lcd.print(WiFi.localIP());
  delay(3000);
  lcd.clear();
}

void loop() {
  // Read incoming serial data
  while (Serial.available() > 0) {
```

```
    char incomingChar = Serial.read();
    Serial.print(incomingChar);

    if (incomingChar == '\n' || incomingChar == '\r') {
      if (readCounter >= 1) {
        parseSensorData(inputBuffer);
        inputBuffer = "";
        readCounter = 0;
      } else {
        inputBuffer = "";
        readCounter = 0;
      }
    } else {
      inputBuffer += incomingChar;
      readCounter++;
    }
  }
}

// Parses incoming sensor string data
void parseSensorData(String data) {
  inputLength = data.length();

  // Expected format: B17.00H00.00F0L0P0#
  if (data[0] == 'B' && data[6] == 'H' && data[12] == 'F' &&
      data[14] == 'L' && data[16] == 'P' && data[18] == '#') {

    String tempStr = data.substring(1, 6);
    String humStr = data.substring(7, 12);
```

```
String fireStr = data.substring(13, 14);
String ldrStr = data.substring(15, 16);
String pirStr = data.substring(17, 18);

temperatureValue = tempStr.toInt();
humidityValue = humStr.toInt();
fireValue = fireStr.toInt();
lightLevel = ldrStr.toInt();
motionValue = pirStr.toInt();

// Display values on Serial Monitor
Serial.println("Temperature = " + String(temperatureValue));
Serial.println("Humidity = " + String(humidityValue));
Serial.println("Fire = " + String(fireValue));
Serial.println("LDR = " + String(lightLevel));
Serial.println("PIR = " + String(motionValue));

// Display values on LCD
lcd.setCursor(0, 0);
lcd.print("T: ");
lcd.print(temperatureValue);
lcd.setCursor(7, 0);
lcd.print(" H: ");
lcd.print(humidityValue);
lcd.setCursor(0, 1);
lcd.print("F: ");
lcd.print(fireValue);
lcd.print(" L: ");
lcd.print(lightLevel);
```

```
    lcd.print(" P: ");
    lcd.print(motionValue);

    // Send to cloud database
    sendToServer();
  }
}

// Sends sensor values to a cloud server
void sendToServer() {
  if (WiFi.status() == WL_CONNECTED) {
    http.begin(client, serverURL);
    http.addHeader("Content-Type", "application/x-www-form-urlencoded");

    String requestData = "&value1=" + String(temperatureValue) +
                         "&value2=" + String(humidityValue) +
                         "&value3=" + String(fireValue) +
                         "&value4=" + String(lightLevel) +
                         "&value5=" + String(motionValue);

    int responseCode = http.POST(requestData);

    if (responseCode > 0) {
      Serial.print("HTTP Response: ");
      Serial.println(responseCode);
    } else {
      Serial.print("HTTP Error: ");
      Serial.println(responseCode);
    }
```

```
    http.end();
  } else {
    Serial.println("WiFi Disconnected");
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("WiFi Disconnected");
    delay(1000);
  }

  fetchRemoteCommand(); // Pull control instructions from the server
  delay(1500);
}

// Fetches control commands (e.g., light ON/OFF) from cloud
void fetchRemoteCommand() {
  if (WiFi.status() == WL_CONNECTED) {
    http.begin(client, "http://iotcloud22.in/2045_smarthome/light.json");
    int httpCode = http.GET();

    StaticJsonDocument<256> doc;
    DeserializationError error = deserializeJson(doc, http.getString());

    Serial.println(http.getString());

    if (error) {
      Serial.print(F("JSON Parse Failed: "));
      Serial.println(error.f_str());
      return;
    }
```

```
    // Parse command for light 1
    String light1 = doc["light1"];
    if (light1 == "on1") digitalWrite(D6, LOW);
    if (light1 == "off1") digitalWrite(D6, HIGH);

    // Parse command for light 2
    String light2 = doc["light2"];
    if (light2 == "on2") digitalWrite(D5, LOW);
    if (light2 == "off2") digitalWrite(D5, HIGH);

    // Auto/manual control mode
    String mode = doc["light"];
    if (mode == "on") {
      runAutoMode(); // Execute automatic control
    } else if (mode == "off3") {
      Serial.println("MANUAL MODE");
    }

    http.end(); // Close HTTP connection
    delay(1000);
  }
}

// Handles automatic control logic based on sensor data
void runAutoMode() {
  if (motionValue == 0 && lightLevel < 4 && fireValue < 8) {
    lcd.setCursor(0, 1);
    lcd.print(" NIGHT MOTION ");
```

```
    digitalWrite(D5, LOW);
    digitalWrite(D6, LOW);
  } else if (motionValue == 0 && lightLevel > 4 && fireValue < 8) {
    digitalWrite(D5, HIGH);
    digitalWrite(D6, LOW);
    lcd.print(" DAY MOTION ");
  } else if (motionValue == 1 && lightLevel < 4 && fireValue < 8) {
    digitalWrite(D5, LOW);
    digitalWrite(D6, HIGH);
    lcd.print(" NIGHT CLEAR ");
  } else if (motionValue == 1 && lightLevel > 4 && fireValue < 8) {
    digitalWrite(D5, LOW);
    digitalWrite(D6, HIGH);
    lcd.print(" DAY CLEAR ");
  } else {
    lcd.print(" FIRE DETECTED ");
    digitalWrite(D5, HIGH);
    digitalWrite(D6, HIGH);
  }
}
```

CONCLUSION

This project has demonstrated the successful design and implementation of a smart home automation system based on the NodeMCU microcontroller and Wi-Fi technology. By enabling wireless communication and control, the system allows users to monitor and manage household appliances remotely through the internet, enhancing convenience, flexibility, and real-time access from virtually any location.

Designed in alignment with IEEE 1621-2004 standards, the system serves not only residential but also commercial and consumer applications. Beyond basic remote control, the system incorporates intelligent features capable of analyzing and adapting to user behavior, thereby automating routine operations. This intelligent control framework contributes to improved energy efficiency by optimizing appliance usage patterns and minimizing unnecessary power consumption.

In addition to its technical capabilities, the system also provides important social benefits. It is particularly advantageous for elderly individuals and persons with disabilities, offering them greater independence in handling everyday tasks such as controlling lighting, fans, or other electrical devices without external assistance.

Overall, the project delivers a practical, scalable, and inclusive smart home solution that integrates ease of use, energy efficiency, and accessibility—making it a valuable contribution to the advancement of home automation technologies.

The evolution of home automation is closely linked to the integration of advanced technologies, particularly Artificial Intelligence (AI), which is expected to significantly enhance the intelligence, adaptability, and autonomy of smart home systems. Future developments in this area will focus on enabling systems to learn user behavior, predict needs, automate repetitive tasks more efficiently, and make real-time decisions to improve energy efficiency and security.

A notable direction for advancement lies in the continued development of voice-controlled virtual assistants, such as Amazon Alexa, Google Assistant, and Apple Siri. As natural language processing capabilities progress, these systems will become more adept at understanding complex instructions, handling multitasking scenarios, and offering highly personalized user interactions based on individual routines and preferences.

Health monitoring integration represents another promising area. Future smart home systems may incorporate biomedical sensors to continuously monitor vital parameters such as heart rate, body temperature, and respiratory function. This application is particularly valuable for elderly individuals and patients requiring continuous supervision. Research in this domain will focus on enhancing sensor accuracy and developing intelligent algorithms that can detect anomalies and automatically notify caregivers or healthcare services.

In addition, energy optimization will remain a key focus. AI-based home automation systems are expected to incorporate intelligent load balancing, predictive energy management, and integration with renewable energy sources such as solar panels and smart grids. These enhancements will support sustainability goals by minimizing energy waste and empowering users to actively manage their energy consumption and production.

Overall, future work will emphasize the development of smarter, more sustainable, and health-aware home environments, extending the impact of home automation from convenience to critical applications in daily life and well-being.

REFERENCES

1. Aldrich D’mello, GauravDeshmukh, ManaliMurudkar, and GarimaTripathi, “Home automation using RaspberryPi 2,” in International Journal of Current Engineering and Technology, Vol.6, No.3 (June 2016)

2. Ana Marie. D Celebre, Ian Benedict A. Medina, Alec Zandrae D. Dubouzet, Adrian Neil M. Surposa, and Engr. Reggie C. Gustilo, “Home automation using RaspberryPi through Siri enabled mobile devices,” in Proc. 8th IEEE International Conference Humanoid, Nanotechnology, Information Technology Communication and Control, Environment and Management (HNICEM) The Institute of Electrical and Electronics Engineers Inc. (IEEE) – Philippine Section 9-12 December 2015, Waterfront Hotel, Cebu City, Philippines

3. MamataKhatu, NeethuKaimal, Pratik Jadhav, and Syedali Adnan Rizvi, “Implementation of Internet of Things for home automation,” in International Journal of Emerging Engineering Research and Technology Volume 3, Issue 2, February 2015, PP 7-11 p-ISSN 2349-4395 & e-ISSN 2349-4409

4. Narender M and Vijayalakshmi M, “RaspberryPi based advanced scheduled home automation system through E-mail,” in 2014 IEEE International Conference on Computational Intelligence and Computing Research

5. Pavithra.D and RanjithBalakrishnan, “IoT based monitoring and control system for home automation,” in Proceedings of 2015 Global Conference on Communication Technologies (GCCT 2015)

6. Pooja Patel, Mitesh Patel, VishwaPanchal, and VinitNirmal, “Home automation using Internet of Things,” in Imperial Journal of Interdisciplinary Research (IJIR) Vol-2, Issue-5, 2016 ISSN: 2454-1362 DhakadKunal, DhakeTushar, UndegaonkarPooja, ZopeVaibhav, and VinayLodha, “Smart home automation using IOT,” in International Journal of Advanced Research in Computer and Communication Engineering Vol. 5, Issue 2, February 2016

7. RozitaTeymourzadeh, Ceng, Salah Addin Ahmed, KokWai Chan, and MokVeeHoong, “Smart GSM based home automation system,” in IEEE Conference on

Systems, Process & Control (ICSPC 2013), 13 - 15 December 2013, Kuala Lumpur, Malaysia

8. Sarthak Jain, AnantVaibhav, and Lovely Goyal, “RaspberryPi based interactive home automation system through e-mail,” in 2014 International Conference on Reliability, Optimization and Information Technology ICROIT 2014, India, Feb 6- 8 2014

9. ShruthiRaghavan and Girma S. Tewolde, “Cloud based low-cost home monitoring and automation system,” in Proceedings of the 2015 ASEE North Central Section Conference 2015, American Society for Engineering Education

10.SilviuFolea, Daniela Bordencea, CasianaHotea, and HonoriuValean, “Smart home automation system using Wi-Fi low power devices,”

11.SirisillaManohar and D. Mahesh Kumar, “E-mail interactive home automation system,” in SirisillaManohar et al,International Journal of Computer Science and Mobile Computing, Vol.4 Issue.7, July-2015, pg. 78-87

12.Vinaysagar K N and Kusuma S M, “Home automation using Internet of Things” in International Research Journal of Engineering and Technology (IRJET), Volume: 02 Issue: 03 | June-2015, e-ISSN: 2395-0056, p-ISSN: 2395-0072