Kingdom Of Saudi Arabia
Ministry Of Education
Onaizah colleges
College of Engineering and IT
Computer Science Department
Cyber Security Program

المملكة العربية السعودية
وزارة التعليم
كليات عنيزة الأهلية
كلية الهندسة وتقنية المعلومات
قسم علوم الحاسب
برنامجد الأمن السيبراني

نظام أمن منزلي قائم على إنترنت الأشياء باستخدام أردوينو

الطلاب :

تركي حمود المنصور	451110822
عزام خالد الزعاقي	432110816
محمد أيمن النجران	441110405
محمد عارف الحربي	432110828
ياسر غزاي الحربي	441110373

المشرف :

د. أحمد علي الحجي

تقرير مشروع مُقدم كجزء من متطلبات درجة البكالوريوس في الأمن السيبراني.

كليات عنيزة، المملكة العربية السعودية

2024/2025- 1446

Kingdom Of Saudi Arabia
Ministry Of Education
Onaizah colleges
College of Engineering and IT
Computer Science Department
Cyber Security Program

المملكة العربية السعودية
وزارة التعليم
كليات عنيزة الأهلية
كلية الهندسة وتقنية المعلومات
قسم علوم الحاسب
برنامج الأمن السيبراني

IoT based home security system using Arduino

Students:

Turki Hamud Al-Manssor	**451110822**
Azzam Khaled Alzaaqi	**432110816**
Mohammed Aiman Alnjran	**441110405**
Mohammed Aref Alharbi	**432110828**
Yasser Ghazai Alharbi	**441110373**

Supervisor:

Dr. Ahmad Ali Al-Hajji

A project report submitted in partial fulfillment of the requirements for B.Sc. degree in Cyber Security.

Onaizah Colleges, Saudi Arabia
2024/2025- 1446

Table of Contents

CHAPTER ONE

CHAPTER TWO

CHAPTER THREE

CHAPTER FOUR:

IMPLEMENTATION AND TESTING

CHAPTER FIVE:

RESULTS AND DISCUSSION

CHAPTER SIX:

CONCLUSIONS AND FUTURE WORK

Appendix

List of Figures

List of Tables

Certificate

It is certified that project report has been prepared and written under my direct supervision and guidance. The project report is approved for submission for its evaluation.

Dr. Ahmad Ali Al-Hajji

Dedication

"To those who motivated me to grasp the knowledge and paved my way with a light of hope." – Yasser Ghazai Alharbi

"To the friends who have had great experiences with me, you are my support." – Mohammed Aref Alharbi

"My family, you are my strength and my inspiration every moment of my life." – Azzam Khaled Alzaaqi

"To my companions Mohammed Aiman Alnjran and Turki Hamud Al-Manssor, your presence has been a constant source of encouragement and brotherhood."

Acknowledgement

I would like to express my sincere gratitude to all those who contributed to the successful completion of this project, "IoT-Based Home Security System Using Arduino."

First and foremost, I thank my project supervisor, Dr. Ahmad Ali Al-Hajji, for their invaluable guidance, support, and encouragement throughout the project. Their expertise and insights were instrumental in shaping the direction of my work.

I also wish to acknowledge my peers and colleagues for their collaborative spirit and constructive feedback, which greatly enriched my learning experience. Special thanks to Yasser Ghazai Alharbi and Turki Hamud Al-Manssor and Azzam Khaled Alzaaqi and Mohammed Aiman Alnjran and Mohammed Aref Alharbi for their assistance in brainstorming ideas and troubleshooting technical challenges.

Finally, I would like to thank the creators of Arduino and the various open-source libraries that facilitated the implementation of this project. Their dedication to innovation and education inspired and empowered me to explore new technologies.

This project reflects the collective efforts and support of many, and I am truly thankful for everyone's contributions.

Turki Hamud Al-Manssor
Azzam Khaled Alzaaqi
Mohammed Aiman Alnjran
Mohammed Aref Alharbi
Yasser Ghazai Alharbi

Abstract

This project aims to provide an advanced and reliable security solution based on Internet of Things technologies, ensuring effective and high-efficiency protection for homes against security threats. The (IoT) has revolutionized every aspect of modern life, including home security. This project showcases the design and implementation of an IoT-based residential security system using Arduino to provide an efficient, flexible, and cost-effective solution for monitoring and security of residential properties. The system integrates various sensors and modules such as motion sensors, door and window sensors, gas or smoke detectors, and a Wi-Fi module (ESP8266 or similar) to provide real-time monitoring and alarm capabilities. The heart of the system is the Arduino microcontroller, which acts as a control unit. It collects data from connected sensors and processes it to detect potential security threats such as unauthorized access, motion detection, or dangerous gas levels. Once a threat is detected, the system triggers an alarm and sends an immediate notification to the homeowner's smartphone or email via the Internet. This project explores four different Arduino-based systems designed to enhance automation and security. Each system uses sensors and basic electronics to perform a specific function: a home security alarm, a password-based lock, a radar scanner, and a sound-activated LED system. The goal is to demonstrate how low-cost microcontroller solutions can address practical problems. Data was collected and analyzed for each system to assess performance and reliability. Results show that with simple components and code, effective and functional systems can be built for real-world applications.

CHAPTER ONE

INTRODUCTION

1.1 Introduction

The rise of the (IoT) has made smart home security systems more accessible than ever. By leveraging Arduino, a user-friendly microcontroller, along with IoT technology, you can create a smart home security system that monitors your home and alerts you in real-time through your smartphone. This system utilizes sensors such as motion detectors (PIR) and door/window sensors connected to the Arduino. If any unusual activity is detected, the system triggers an alarm and sends a notification to your phone via Wi-Fi (using ESP8266), allowing you to monitor your home from anywhere [1]. The term IoT refers to a network of physical devices equipped with electronic components that communicate with one another and with the cloud. This connectivity enables a smart ecosystem where users can easily manage their home devices. The affordability of microcontrollers like Arduino, coupled with high-bandwidth internet connections, has accelerated the adoption of IoT in daily life, providing advanced services across various sectors, including healthcare, agriculture, and smart cities [2]. As crime rates continue to rise, the need for effective home security solutions has become increasingly urgent. While advanced security systems are available, they are often not accessible to everyone. This paper proposes a solution by creating a cost-effective electronic device that can detect intruder movements, sound an alarm, and notify users. The system utilizes infrared sensors to detect movement within its range, triggering an alarm and sending signals to the Arduino, which processes the information and generates alerts [3]. Home security is a growing concern today, where the increasing prevalence of crime poses threats to personal safety and property. Traditional security measures, such as locks and alarms, often fall short of providing comprehensive protection. Smart home security systems offer a more effective solution, integrating technology that allows for real-time monitoring and instant notifications. This level of responsiveness not only enhances safety but also provides peace of mind for homeowners [4] Arduino is an open-source electronic development board designed to facilitate interactive electronics projects. It allows users to program microcontrollers easily, enabling the creation of diverse projects involving sensors for temperature, light, and motion. Arduino's flexibility and user-friendly programming environment make it a popular choice for hobbyists and professionals like [5] The IoT-Fog-Cloud hybrid solution offers a robust mechanism for storing, managing, and interpreting data in real-time. IoT sensors, wireless networks, and RFIDs can transmit pervasive information to remote devices, generating data with minimal delay. To enhance performance, sophisticated artificial intelligence technologies, such as machine learning, can be integrated into the system. Despite the advancements in IoT-based security systems, several research gaps remain[6]Current research has not adequately addressed the detection of identity-based parameters of intruders, highlighting the need for user-centered decision-making strategies There is minimal research on the regular monitoring of home security attributes, which can compromise safety Advanced detection techniques The integration of methods like ANFIS-PSO for interactive intruder detection decision-making has been underexplored Quantification of identity parameters Limited work has been done to quantify identity parameters for effective decision-making by security officials and users as shown in Figure.

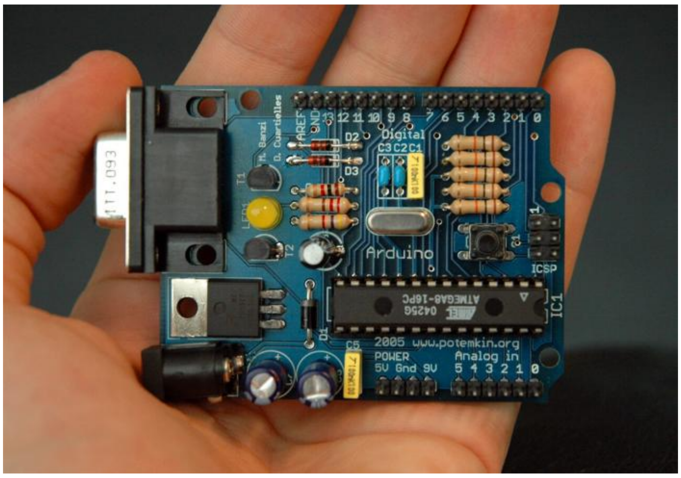


Figure 1-1 Arduino Serial board

1.2 Problem Specification and Motivation

Problem Specification: Traditional home security systems are often inefficient because they depend on physical measures such as locks and alarms, which are only passive and cannot provide real-time monitoring and alerts that can stop a threat at that moment before it happens. The cost of innovative security technology is so high that it becomes a contributing factor in decreasing access to these solutions, especially to those in low-income groups. Moreover, current systems are often characterized by a lack of customization and scalability, which compromises their security capability as security requirements evolve. There is an urgent need for a cheap flexible and user-friendly security system that promotes live monitoring of events, attributes, and data privacy, which is critical for the customer's peace of mind. Motivation: The (IoT) enabled home security system, a system developed through the (IoT) technology that has been the answer to the urgent need for a straightforward, affordable, and reliable alternative that directly addresses the shortcomings of traditional security systems. Moreover, the project by means of the integration of real-time monitoring and instant alerts, which is a smart way of increasing the aesthetics of a house and the ease with which an owner can react to threats leads to the boost of security and well-being in general. In addition, if Arduino and IoT technologies are used, it is possible to create a low-cost security system that more people can afford, which thus can lead to equity in the safety of houses. Besides, the design is aimed at flexibility and can be scaled to the desired security level, which allows users to choose the security settings that fit them best and introduce new features when needed. Moreover, the project aims to build user confidence through strict adherence to data security and a user-friendly interface, which is the first stage of the construction of a smart home technology revolution and allows homeowners to efficiently protect their homes.

1.3 Goals and Objectives

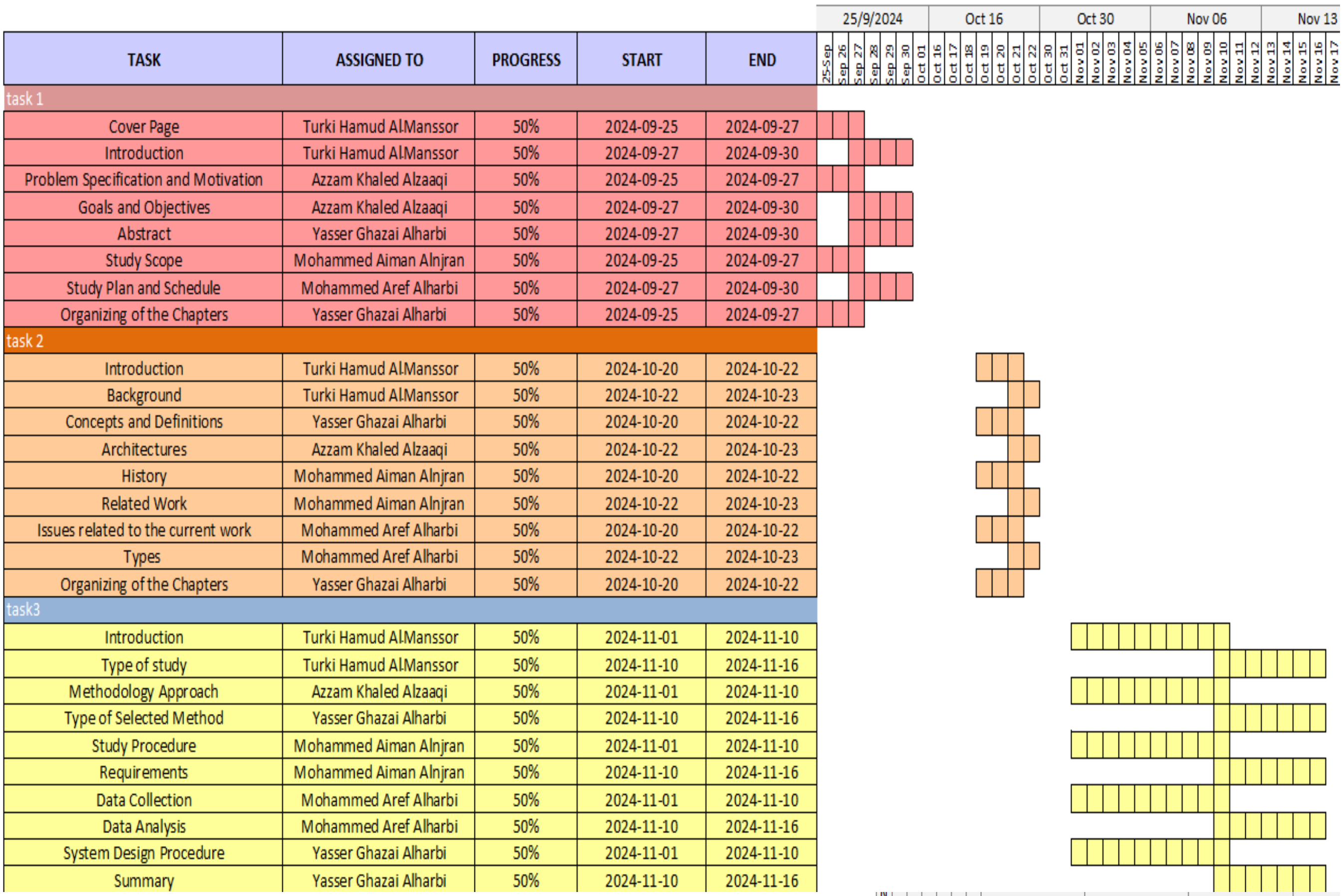
Goals: The main objective of this project is to create a thorough and groundbreaking home security system that fully incorporates the latest IoT technology, thereby safeguarding the future of residences with personal monitoring that is both efficient and adaptable. This system is designed to enable users to take a more active role in the security of their property and besides being a deterrent of intrusions and hazards, it could also mean a stronger feeling of safety and peace of mind. Objectives: Specific and measurable objectives to achieve these goals include Designing a Robust Security Framework: Assemble a home security mechanism which is made up of various IoT components (for instance, sensors, cameras, alarms) for efficient threat detection. Implementing Real-Time Monitoring and Alerts: Set up a cloud-based monitoring system that sends out real-time alerts to homeowners through mobile notifications or emails during security breaches, thus allowing consumers to respond in a timely manner. Creating an Intuitive User Interface: Design a user-friendly mobile and web application that allows homeowners to easily access and control their security settings, view live feeds, and customize alerts, enhancing user engagement. Ensuring Data Security and Privacy: Adopt strong data encryption and safe communication methods to secure user information and privacy, thereby building trust in the users. Facilitating System Scalability and Adaptability: Design the system to be scalable, enabling the easy addition of new components without causing interruption in the operation of the existing functionalities, thereby securing its continuity in the path of user needs.

1.4 Study Scope

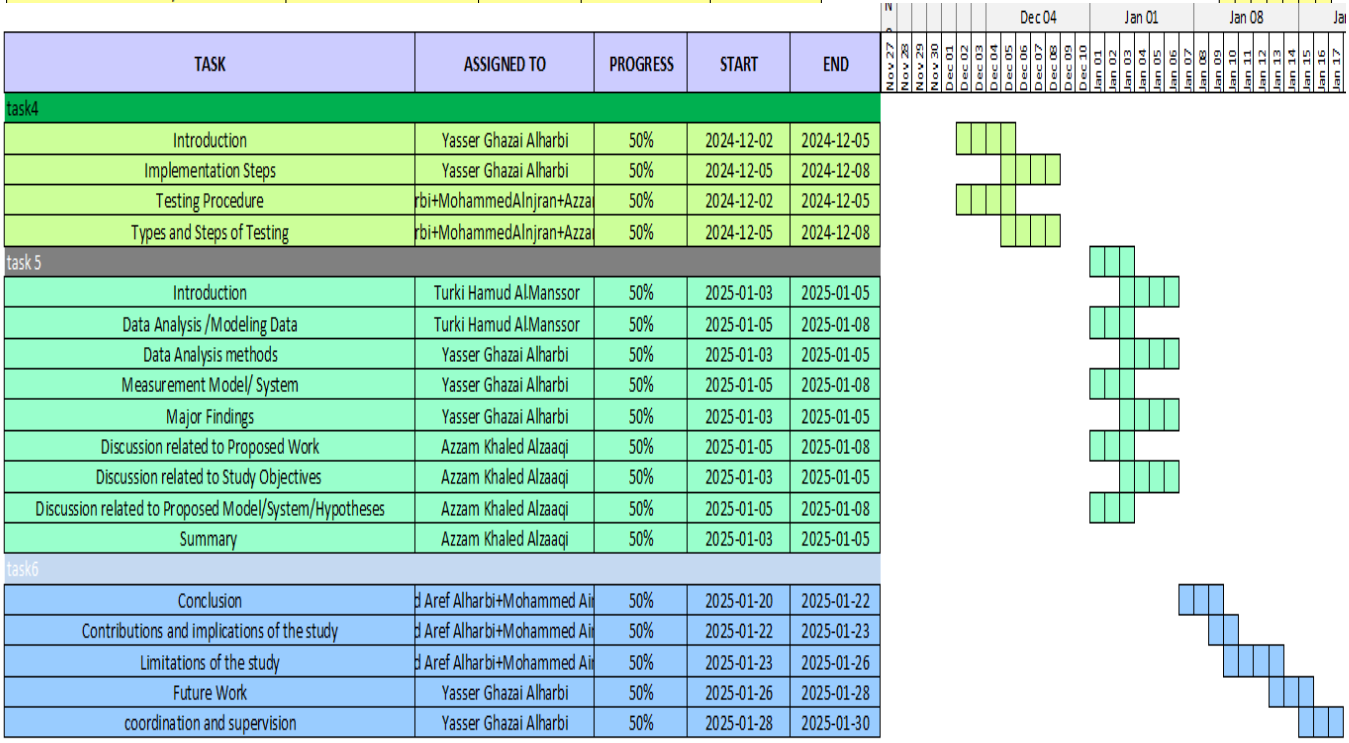
This study focuses on the design and implementation of an IoT-based home security system using Arduino. The project aims to create a comprehensive solution that enhances residential safety through advanced technology. The scope includes the following key elements: System Components: The project will integrate a variety of essential hardware components, including: Sensors: PIR motion sensors, door and window contact sensors, gas and smoke detectors. Alarm System: A Pizzo and an alarm mechanism for immediate alerts. WLAN Module: Modules for seamless connectivity. Power Source: Options for either battery or direct power supply. Functional Features: The system will offer robust capabilities such as: Real-time monitoring and instant alerts for potential security threats. Remote access for homeowners through a user-friendly mobile or web application, enabling them to manage their security system from anywhere.- Integration with IoT Technologies: The project will explore the integration of various IoT technologies to improve the efficiency and reliability of the home security system. This includes data flow management from sensors to Arduino, and from the Arduino to the cloud server and user interface. - User-Centric Design: Emphasis will be placed on developing an intuitive and accessible design that accommodates users with varying levels of technical expertise, ensuring ease of use and effective interaction with the system. - Performance Evaluation: A thorough evaluation of the system’s performance will be conducted under various scenarios to verify its effectiveness and reliability in real-world applications. This evaluation will involve stress testing and user feedback.

1.5 Study Plan and Schedule

Project timeline



TASK	ASSIGNED TO	PROGRESS	START	END
task 1				
Cover Page	Turki Hamud AlManssor	50%	2024-09-25	2024-09-27
Introduction	Turki Hamud AlManssor	50%	2024-09-27	2024-09-30
Problem Specification and Motivation	Azzam Khaled Alzaaqi	50%	2024-09-25	2024-09-27
Goals and Objectives	Azzam Khaled Alzaaqi	50%	2024-09-27	2024-09-30
Abstract	Yasser Ghazai Alharbi	50%	2024-09-27	2024-09-30
Study Scope	Mohammed Aiman Alnjran	50%	2024-09-25	2024-09-27
Study Plan and Schedule	Mohammed Aref Alharbi	50%	2024-09-27	2024-09-30
Organizing of the Chapters	Yasser Ghazai Alharbi	50%	2024-09-25	2024-09-27
task 2				
Introduction	Turki Hamud AlManssor	50%	2024-10-20	2024-10-22
Background	Turki Hamud AlManssor	50%	2024-10-22	2024-10-23
Concepts and Definitions	Yasser Ghazai Alharbi	50%	2024-10-20	2024-10-22
Architectures	Azzam Khaled Alzaaqi	50%	2024-10-22	2024-10-23
History	Mohammed Aiman Alnjran	50%	2024-10-20	2024-10-22
Related Work	Mohammed Aiman Alnjran	50%	2024-10-22	2024-10-23
Issues related to the current work	Mohammed Aref Alharbi	50%	2024-10-20	2024-10-22
Types	Mohammed Aref Alharbi	50%	2024-10-22	2024-10-23
Organizing of the Chapters	Yasser Ghazai Alharbi	50%	2024-10-20	2024-10-22
task3				
Introduction	Turki Hamud AlManssor	50%	2024-11-01	2024-11-10
Type of study	Turki Hamud AlManssor	50%	2024-11-10	2024-11-16
Methodology Approach	Azzam Khaled Alzaaqi	50%	2024-11-01	2024-11-10
Type of Selected Method	Yasser Ghazai Alharbi	50%	2024-11-10	2024-11-16
Study Procedure	Mohammed Aiman Alnjran	50%	2024-11-01	2024-11-10
Requirements	Mohammed Aiman Alnjran	50%	2024-11-10	2024-11-16
Data Collection	Mohammed Aref Alharbi	50%	2024-11-01	2024-11-10
Data Analysis	Mohammed Aref Alharbi	50%	2024-11-10	2024-11-16
System Design Procedure	Yasser Ghazai Alharbi	50%	2024-11-01	2024-11-10
Summary	Yasser Ghazai Alharbi	50%	2024-11-10	2024-11-16



TASK	ASSIGNED TO	PROGRESS	START	END
task4				
Introduction	Yasser Ghazai Alharbi	50%	2024-12-02	2024-12-05
Implementation Steps	Yasser Ghazai Alharbi	50%	2024-12-05	2024-12-08
Testing Procedure	rbi+MohammedAlnjran+Azza	50%	2024-12-02	2024-12-05
Types and Steps of Testing	rbi+MohammedAlnjran+Azza	50%	2024-12-05	2024-12-08
task 5				
Introduction	Turki Hamud AlManssor	50%	2025-01-03	2025-01-05
Data Analysis /Modeling Data	Turki Hamud AlManssor	50%	2025-01-05	2025-01-08
Data Analysis methods	Yasser Ghazai Alharbi	50%	2025-01-03	2025-01-05
Measurement Model/ System	Yasser Ghazai Alharbi	50%	2025-01-05	2025-01-08
Major Findings	Yasser Ghazai Alharbi	50%	2025-01-03	2025-01-05
Discussion related to Proposed Work	Azzam Khaled Alzaaqi	50%	2025-01-05	2025-01-08
Discussion related to Study Objectives	Azzam Khaled Alzaaqi	50%	2025-01-03	2025-01-05
Discussion related to Proposed Model/System/Hypotheses	Azzam Khaled Alzaaqi	50%	2025-01-05	2025-01-08
Summary	Azzam Khaled Alzaaqi	50%	2025-01-03	2025-01-05
task6				
Conclusion	d Aref Alharbi+Mohammed Ai	50%	2025-01-20	2025-01-22
Contributions and implications of the study	d Aref Alharbi+Mohammed Ai	50%	2025-01-22	2025-01-23
Limitations of the study	d Aref Alharbi+Mohammed Ai	50%	2025-01-23	2025-01-26
Future Work	Yasser Ghazai Alharbi	50%	2025-01-26	2025-01-28
coordination and supervision	Yasser Ghazai Alharbi	50%	2025-01-28	2025-01-30

1.6 Organizing of the Chapters

Chapter One: Introduction Provides an overview of the topic and its importance. It defines the problem, explains the motivation behind the study, states the main and specific objectives, outlines the scope and limitations, and presents a study plan with a timeline.

Chapter Two: Literature Review It starts with an introduction and background information. It defines key concepts, discusses relevant types and architectures, and traces the historical development. Reviews related work, previous studies, theories, and frameworks. Presents the proposed work, including models, anticipated challenges, and hypotheses, ending with a summary.

Chapter Three: Methodology Introduces the study type (theoretical, simulation-based, or prototype-based). Describes the research methods, procedures, requirements, data collection and analysis techniques, and the system design process. Concludes with a summary.

Chapter Four: Implementation and Testing

Begins with an introduction, followed by the steps and procedures of implementation. It explains the sampling and testing procedures, describes testing types and steps, and concludes with a summary.

Chapter Five: Results and Discussion Introduces the section and explains data analysis methods. Highlights major findings and discusses them in relation to the proposed work, study objectives, and hypotheses. Ends with a summary of the key findings and their implications.

Chapter Six: Conclusions and Future Work Summarizes the study’s main findings, contributions, and implications. Discusses the study’s limitations and suggests directions for future research.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

Internet of Things technology has been a major driver for the technological progress of Information and Communication Technology (ICT) [7]. Developments of small, internet-enabled, and wireless sensors have not only revolutionized the ubiquitous data perception methodology, but it has inserted a vision of smartness in the ambient environment everywhere [8]. According to the Statista survey, the global IoT market is expected to pass 1.6 trillion US Dollars by 2025 (Source: https://www.statista.com). This includes an estimation of 34.2 billion connected IoT devices around the world (Source: https://iot-analytics.com). Advancements of the IoT paradigm has realized numerous innovations that were nearly impossible in previous decades due to underdeveloped technology. Some of these include Smart Homes [6], Mobile Healthcare [7], Intelligent Transportation [8], Smart Food Hubs [9], and Smart Agriculture [10]. Additionally, IoT technology contributes significantly to provisioning security in the form of wireless cameras, motion sensors, and smart locking mechanisms [11]. However, continuous research is still going on in the development of smart security systems for homes and buildings by incorporating the novel vision of IoT technology and Fog Computing [12]. Fog computing, as coined by [13], is a virtual platform for provisioning real-time computation, the capacity to store, and services at the network between the users and data storage, which are present over the network [14]. The collaboration of IoT-Fog technology can analyze IoT data for security-based decision-making in time-sensitive manner [15] Research Field 1.0 Smart Homes and Intelligent Buildings are characterized by ubiquitous services and effective decision-making with enhanced accuracy. Designing smart security solutions has always been a challenging aspect for researchers around the world. Security in the form of biological-locks, automated doors, and smart alarms has been deployed to wireless protection for homes, buildings, and parking. However, recent studies have shown vulnerabilities in these security solutions. A survey report presented by Protect America depicts 75% of the smart devices are vulnerable to the security breach (Source: https://www.protectamerica.com/). Moreover, there are 32 risks identified by Jacobsson et al. [16] out of 25% were classified as highly vulnerable. Therefore, need-of-the-hour is to minimize current smart security risks which are derived from human factors either directly or indirectly. This demands novel security solutions that can be easily implanted in the smart home scenario as extra protection. With an advanced model of smart security, both privacy and safety can be enforced to further assess the capability of IoT-inspired secure frameworks. The IoT-Fog-Cloud hybrid solution offers an appropriate mechanism to store, manage, and interpret all-encompassing information in a time-sensitive manner. IoT sensors, including wireless networks, preceptors, and RFIDs can transmit pervasive information to remote devices [17]. The development of sensing technologies can also generate stochastic data with minimal delay [18]. For optimum performance, sophisticated artificial intelligence technologies such as machine/deep learning are incorporated for accomplishing the notion of the smart mat-based framework [19]. The core part of the presented system is to map and analyze the various identity-oriented parameters in real time. Additionally, the comprehensive literature review has identified multiple research gaps -Detection of real-time identity-based parameters of intruder personnel has not been addressed specifically by the researchers. It is essential to develop user-centered decision-making strategies.

-Minimal research has been presented for regularized monitoring of home security and related attributes by the monitoring officials, thereby compromising the home security.

Intrusion Detection Systems (IDS) are critical components in the field of cybersecurity, designed to monitor network traffic and detect suspicious activities or policy violations. IDS can be classified into two main types: Network-based IDS (NIDS), which analyzes traffic on the network level, and Host-based IDS (HIDS), which monitors individual devices for malicious activities. [21] Detection Identifying potential threats using various techniques, such as signature-based detection (matching known attack patterns) and anomaly-based detection (flagging unusual behavior). Alerting, when suspicious activities are detected, the IDS generates alerts for system administrators to investigate further. Loggin Recording all relevant data related to detected incidents, which is crucial for forensic analysis and compliance purposes. The importance of IDS has grown with the increasing complexity of cyber threats, including advanced persistent threats (APTs) and zero-day exploits. Effective IDS can enhance an organization's security posture by providing real-time insights into potential vulnerabilities and breaches. Research in IDS focuses on improving detection accuracy, reducing false positives, and integrating advanced technologies such as machine learning and artificial intelligence to enhance threat detection capabilities. As IoT devices become more prevalent, the need for robust IDS tailored to handle the unique challenges of these environments is becoming increasingly critical. Overall, IDS plays a vital role in safeguarding information systems, enabling proactive responses to potential intrusions and enhancing overall security resilience. -Another factor that has been minimally explored in state-of-the-art research is the incorporation of ANFIS-PSO for interactive intruder detection decision-making. -Finally, limited work has been done to quantify the identity parameters for effective decision-making by security officials and users State-of-the-Art Research Objectives. This section provides major contributions presented by the proposed home security framework. In the current research, several compact, internet-equipped IoT sensors are embedded in the smart mat, which consistently gathers data regarding the user's identity parameters for preventing intrusion. When an IoT embedded mat is stepped on by the user, the identity of the user is monitored based on certain parameters. Moreover, users get alerted by activation of pleasant chime or loud alarm as per acquaintance or intruders’ presence, respectively. Significantly, identity-related information is generated which includes parameters like weight, foot size, pressure, movement, which is further analyzed for person identification via the fog computing node. Figure 1 displays the conceptual illustration of the proposed framework. Specifically, the presented framework is aimed at realizing the following objectives to realize the overall secure mat-based system.

2.2 Background

Research Title: Developing a Home Security System Based on the Internet of Things Using Arduino Abstract: With the rapid development of (IoT) technologies, it has become possible to design high-income, low-cost smart home security systems. This research aims to provide a home security system based on the Arduino platform as a control and processor tool, so that it monitors the house and detects any unusual activity using a set of sensors such as the PIR sensor and door and window sensors. Arduino collects and processes data from these sensors, then sends instant notifications to the homeowner via Wi-Fi using the ESP8266 module, to ensure that there is no danger. This system is characterized by ease of installation, cost, and scalability of data for different users. The transformation and expansion of the use of Internet of Things technologies may contribute significantly to changing the lifestyle of individuals, especially home security. Internet of Things technologies allow the connection of data parts later to provide smart services, including security systems. This research aims to review how to build a home security system based on IoT technologies using the Arduino board, which provides an effective and low-cost alternative to traditional security systems [1]. Tools and components used: (Arduino UNO): used as the main health processor. Motion sensor (PIR): detects any unusual movement inside or outside the house. Door and window locators: to control the opening or closing of doors and windows. ESP8266 Wi-Fi module: sends data and sends notifications to the user. Camera (optional): also pictures when an alarm occurs. Audio alarm: to warn the residents of the house in the event of a Bluetooth error [2]. Working mechanism: (Data collection): collects data from sensors, doors and windows Data processing: handles the Arduino data and determines the extent of the situation. Sending alerts: In case of an error, the system notifies the homeowner via the Internet, using the ESP8266 module, which sends an instant alert or email. New extensions: Possibility of adding additional surveillance cameras and comprehensive protection. Available results: The proposal is proposed for home security, through the user's use of the apartment monitoring, with great speed for tourist accidents. The system is flexible to improve and expand as needed, making it a practical and highly efficient option for those who want to have a security system that is easy to install [3]. Conclusion: This system is a home model for modern security systems based on Internet of Things technologies. Thanks to the use of Arduino and ESP8266, a grain security system can be created, easy to use and cost-effective. This provides a practical framework that can be developed in the future to include additional features such as facial recognition and data analysis via artificial intelligence. Arduino is an electronic development board consisting of an open-source electronic circuit with a microcontroller that can be programmed via a computer and is designed to facilitate the use of interactive electronics in multidisciplinary projects. Arduino is mainly used in designing interactive electronic projects or projects that aim to build various environmental sensors such as temperature, wind, light, pressure, etc. Arduino can be connected to various programs on a personal computer, and its programming depends on the open-source programming language Processing, and the programming codes for the Arduino language are similar to the C language and are considered one of the easiest programming languages used to write microcontroller programs. Some studies have shown that Arduino chips are an important entry point through which it is easy to learn the principles of computer science, electrical and mechanical engineering, as well as crafts and arts, combined in one environment[4] The Arduino Integrated Development Environment (IDE) is a cross-platform application (for Windows, macOS, and Linux) written in C and

C++ functions. It is used to write and upload programs to Arduino-compatible boards, but also with the help of third-party kernels, other vendors' development boards [5] The source code for the IDE is released under the GNU General Public License, version 2. The Arduino IDE supports C and C++ languages using special rules for structuring the code. The Arduino IDE provides a software library from the Wiring project, [6] which provides many common input and output routines. The user-written code requires only two basic functions, to start the sketch and the main program loop, which are compiled and linked by the stub main () program into a cyclic executable program with the GNU toolchain, also included in the IDE distribution. The Arduino IDE uses the avrdude program to convert the executable code into a hexadecimal text file that is loaded onto the Arduino board by a program loaded into the board's firmware. By default, avrdude is used as the loader to flash user code on official Arduino boards [7] The Arduino IDE is a derivative of the Processing IDE, but as of version 2.0, the Processing IDE will be replaced by the Eclipse Theia IDE framework based on Visual Studio code. As Arduino has grown in popularity as a software platform, other vendors have begun implementing custom open-source compilers and tools (kernels) that can create and upload sketches to other microcontrollers not supported by the official Arduino line of microcontrollers [8].

2.2.1 Concepts and Definitions

The concept of the (IoT) refers to a network of physical objects equipped with sensors, software, and other technologies designed to connect and exchange data with other devices and systems over the Internet. Smart homes can be defined as residences that fully integrate home automation and security within the (IoT) paradigm. By connecting various household objects to the Internet, homeowners can remotely monitor and control their environment. This includes features such as lamps programmed to turn off at specific times and smart thermostats that regulate temperatures while providing detailed energy usage reports. Smart home security systems require automated and managed solutions, encompassing controlled networks, communication systems, emergency response mechanisms, and anti-theft monitoring. The rise of affordable smartphones, microcontrollers, and open-source hardware, along with the increasing adoption of cloud services, has facilitated the development of low-cost smart home security systems. Furthermore, these systems are particularly beneficial for families with busy lifestyles and can accommodate household members with limited mobility, such as the elderly and individuals with disabilities [1]. Arduino is a highly versatile microcontroller and development environment that facilitates control over devices and allows for data retrieval from various sensors. Its user-friendly design and adaptability, coupled with widespread popularity and user adoption, have resulted in a range of hardware extensions and software libraries that support both wired and wireless connectivity to the Internet. Thus, Arduino serves as an ideal open hardware platform for exploring the possibilities of the Internet of Things.

2.2.2 Types of smart systems

Anti-theft alarm system

Use motion sensor (PIR sensor) to detect unauthorized access and trigger alarm or notification. Components: Arduino, PIR motion sensor, buzzer, Wi-Fi module (such as ESP8266) and smartphone app for alarm as shown in Figure [1].

Figure 2.1 smartphone app for alarm

Security Camera System

Integrate a camera with Arduino to monitor live feeds and detect motion. Components: Arduino, camera module (e.g. ESP32-CAM), micro-SD card for storage, and web server for live streaming as shown in Figure [2].

Figure 2.2 web server for live streaming

Smart Door Lock System

Electronic lock controlled by smartphone app or RFID card.

Components: Arduino, servo motor, RFID reader, Wi-Fi module and mobile app for access control as shown in Figure [3].

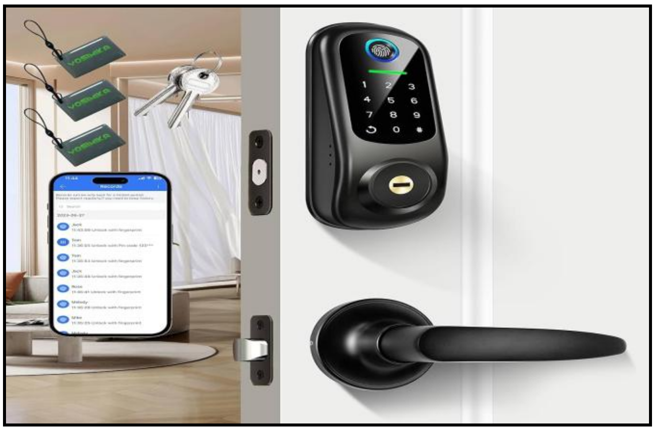


Figure 2.3 Smart Door Lock

Environmental Monitoring System

Monitor environmental conditions that may indicate a safety threat, such as smoke, gas leaks, or temperature changes. Components: Arduino, gas sensor, temperature/humidity sensor, buzzer, and Wi-Fi module for alarm as shown in Figure [4].

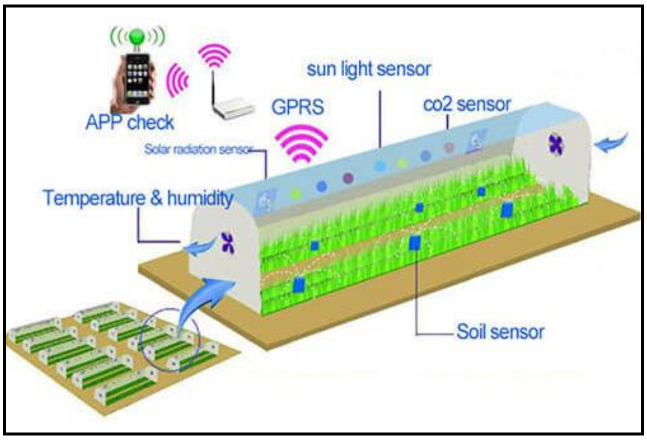


Figure 2.4 Agricultural Environment Monitoring System

Automated Lighting Control

Activates and deactivates lights based on occupancy or time, improving security by mimicking presence. Components: Arduino, relay module, light sensors, and PIR sensors as shown in Figure [5].

Figure 2.5 Automatic Room Lights using Arduino and PIR Sensor

Window and Door Sensor System

Employs magnetic sensors to monitor the opening of windows and doors. Components: Arduino, magnetic switches, buzzer, and Wi-Fi module for alerts as shown in Figure [6].

Figure 2.6 Wi-Fi Door Window Alarm Sensor with Magnetic Sensor

Face Recognition System

Utilizes a camera and machine learning to identify authorized individuals. Components: Arduino, camera module, face recognition module (e.g., OpenCV), and a notification system as shown in Figure [7].

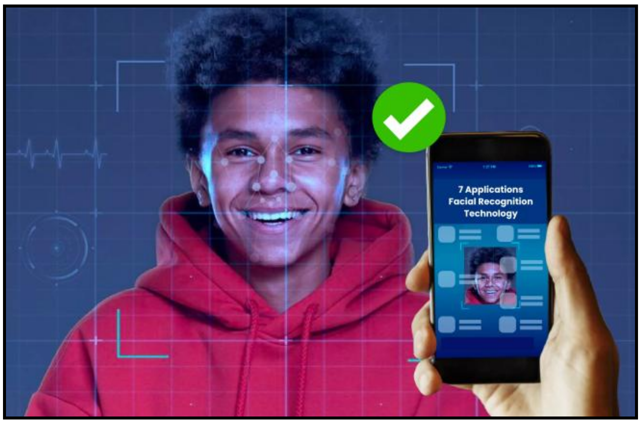


Figure 2.7 Facial Recognition Technology

Intelligent Alarm System

A customizable alarm system capable of being activated by a variety of sensors, including motion and door/window sensors. Components: Arduino, various sensors, buzzer, and mobile notification system as shown in Figure [8].

Figure 2.8 Interchangeable Wireless Home Security System with Alarm System

2.2.3 Architectures

In this section, details of the various architectural frameworks that are the most used in the IoT-based home security systems have been discussed. The architecture typically consists of Sensor Layer: This includes devices such as motion sensors, door/window contacts, smoke detectors, and cameras. The relevant ones are these sensors or things that change according to the environment or possible intrusions are being recorded by them. Communication Layer: This layer serves as the data transmission bridge between sensors and the central processing unit. Among the commonly used communication protocols are Wi-Fi, Zigbee, Z-Wave, and GSM. The protocol for each connection is chosen based on factors like distance, power consumption, and data transfer rate Processing Layer: The core of this system consists of microcontrollers, such as Arduino and Raspberry Pi, which process data coming from sensors. This layer is responsible for carrying out the algorithms that decide whether the raw data, which usually is a digital version of the analog sensors, contain any signs of anomalies and further steps such as sending an alert should be made or not. User Interface Layer: In this layer, through mobile apps or web interfaces, homeowners can log in and access their home system. Through the online interface, users can keep track of their homes in real-time, they can receive alerts and even control different security features remotely.

2.2.4 History

It was in 1853 that the first type of home security system was invented by Augustus Russell Pope in the US. This is one of the electro-magnetic alarm systems which was a technological breakthrough for house owners at that time whose security systems were alarm calls of animals. The system was working by triggering whenever the circuit got closed, each door and window was connected as a separate, parallel line. Hence, opening the door would result in the flow of the current which will cause the magnet to vibrate, and hammer would strike the bell. Holmes is often credited with being the first person to develop the electric burglar alarm system, but infect, it was Pope who invented the home security system. In the late 1900s and early 2000s, security systems that included major upgrades, became the leading home security equipment: in 1966, Marie Van Brittan Brown invented the first system to open the door with the remote control, having a camera. The 1970s saw the development of video cameras that were motorized and capable of transmitting grainy images to televisions. Besides cameras, intercoms, and alarm buttons have been added as well. The 1980s led to the introduction of infrared technology to systems that greatly reduced mishaps even with the advent of new opportunities like garage security. Home security systems became popular and cheap enough for the average family to afford by the 1990s [7].

2.3 Related Work

Over the last few years, home security systems development has gained significant attention, particularly with the combination of GSM and IoT. The GSM-based home security system with Arduino platform, discussed by the paper [1], aims to make the homeowners' life more secure and safe. It points out several types of systems that are already in place which use sensory instruments such as PIR for detecting movement, GSM modules for communication, and IoT technologies. The system is meant to find any unauthorized entry into a house and send notifications to homeowners by SMS or phone calls immediately which ensures quick reaction to every possible danger. In same context, the study [2] introduces a home security system based on the Internet of Things and powered by Arduino, suitable for domestic surroundings aiming at safety enhancement. It makes use of different sensors, such as infrared, fire, temperature, and gas sensors, to detect intrusion and hazardous situations. The system goes on and emits beeping when sudden movements or environment disturbances are detected, thus allowing for the real-time notification of the user via Wi-Fi module. The microcontroller works with a keyboard to get user input and an LCD for showing system status, thus, it is easy to operate. Another contribution is found in paper [3] proposes a smart home security system with the aid of Arduino boards and IoT technology to keep control over potential security threats in real-time and avoid them by sending alarm signals to authorized people in different events such as unauthorized entries, water leaks, and temperature fluctuations. The system is equipped with various sensors, such as PIR sensors, to detect motion, and it is also linked to the cloud. This way, the user can be notified instantly through SMS, no matter where he/she is. The survey of the respondents of the study showed that the mature population was keenly interested in smart home security topics, thus proving the fact that the interfaces of the corresponding arrangements must be friendly for the users and the alert mechanisms trustworthy. The paper [4] is a comprehensive design for a home security system that uses several different technologies, such as a GSM module, ultrasonic sensors, PIR, and gas detectors, and it increases the safety and monitoring functions of the house. To improve the system, one of the points to highlight is the GSM technology by which the system can transmit the warning signals to the mobile phones of the owners instantly in case it detects intrusions or gas leakage. Additionally, paper [5] proposes Smart Home Security System (SHSS) by means of an Internet of (IoT) based Intrusion Detection System (IDS) constructed on Arduino that provides the necessary safety at home through the detection and alerting of intruders. This approach incorporates an ultrasonic sensor that is triggered by motion when an Arduino Uno microcontroller detects motion, and a GSM module can send SMS alerts to homeowners when an intrusion is identified. The ultrasonic sensor is the one that determines the distance of an intruder from another by producing sound waves and capturing the reflection of the sound waves, while the GSM module offers mobile communications. It is a cheap option for computers that automate monitoring, send notifications in real-time and solve the growing need for security systems in smart homes. The system has been experimented with and proved to be effective and accurate, which makes it capable of informing the homeowner in case of intrusion and at the same time, it still is functional even when the conditions are difficult, and thus the field of smart home security technologies is enriched. The paper [6] is a comprehensive study on the design and implementation of a home security system which is prototype using Arduino and other sensors. In fact, the paper mainly demonstrates the concept of & IoT technology such that the house is under the direct supervision of different IoT components around or inside the house to provide security in real time along with alert capacity. The system will include sensors such as PIR (Passive Infrared) used for motion discernment and gas/smoke detectors, if triggered it will set off alarms and communicate with the user through an online connection.

2.3.1 Issues related to the current work

The cited systems have shown developments in home security tech although still several hurdles and issues that must be dealt with. For example, the usage of GSM technology may be a disadvantage in areas with poor cellular coverage, as reported in paper [4], and the system's efficacy could be impacted in such cases of emergencies when alarm signals cannot be transmitted. Besides, the synthesis of many sensor types, such as seen in papers [4] and [5], introduces the difficulties of system supervision as well as the possibilities of false alarms coming up from the environmental factors. Further, the interaction design and the user interface must be investigated thoroughly to ensure the systems are easy to use and intuitive, as pointed out in paper. User trust is a vital element of smart home technologies, so the mechanisms should be reliable, and the users should be assured of their stay in line with the trend of smart home technologies. In conclusion, although paper [6] refers to the potential of IoT parts for real-time security, this is just one side of the coin that user data security and privacy should also be thought of carefully. In this way, user data could still be guarded while efficient home security solutions would be through.

2.3.2 Previous studies

The (IoT-based Home Security System using Arduino) is a popular topic in the field of smart home technology. With the rapid advancement of the (IoT), traditional home security systems are being upgraded to provide enhanced protection and convenience. This system integrates IoT technology with Arduino microcontrollers to create a smart, efficient, and cost-effective security solution. In this article, I will review some of the previous studies on IoT-based home security systems using Arduino, examining their approaches, results, and potential areas for improvement [1]. Concept of IoT-Based Home Security System an IoT-based home security system allows users to monitor and control their homes remotely using mobile applications. The system typically consists of various sensors such as motion detectors, cameras, and door/window sensors, all connected to an Arduino controller. The Arduino board acts as the brain of the system, processing input from the sensors and sending alerts to the user's smartphone via the internet in case of any security breach or unauthorized access as shown in Figure [9].

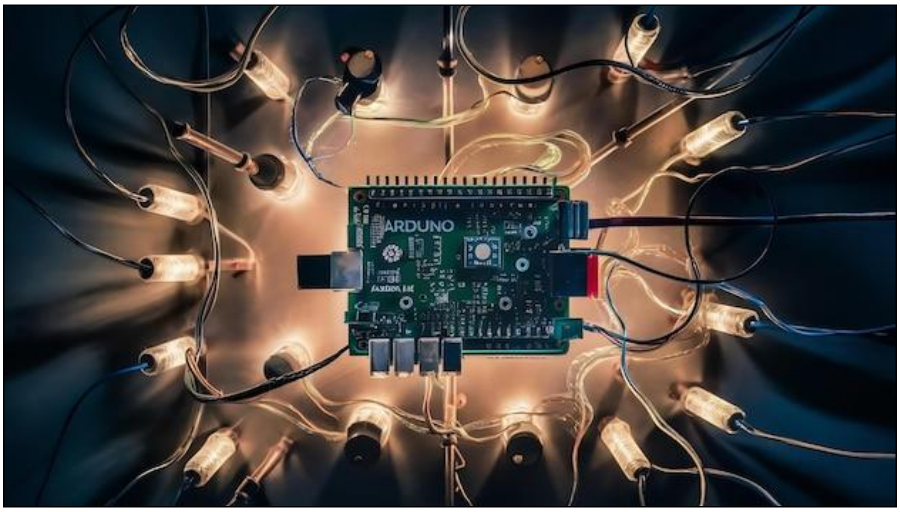


Figure 2.9 Arduino control broad

This study focused on creating a low-cost, efficient home security system using Arduino and various sensors, including motion detectors and a camera. The system was integrated with Google’s Firebase service to send real-time notifications to the user's mobile phone. Whenever the system detected unusual motion, it captured an image and sent an alert notification to the user. Results: The system successfully detected motion and sent real-time alerts. However, there were some delays in notifications due to unstable internet connections. The overall cost was reduced, making it a viable solution for budget conscious users. This research implemented voice control features using Google Assistant to enhance user convenience. The system included door sensors, smart cameras, and an Arduino microcontroller. Users could give voice commands such as "Open the door" or "Start recording video," and the system would execute these actions accordingly.[9] Results: Voice control significantly improved user experience by making the system more intuitive. However, recognizing voice commands in noisy environments was a challenge, requiring further enhancement in voice recognition accuracy. The integration of voice control added a layer of convenience, especially for users with accessibility needs. This study utilized the Blynk application to provide an easy-to-use interface for monitoring the home security system. The system was equipped with gas and smoke sensors connected to an Arduino board. Users could monitor the levels of gas and smoke in their homes through the Blynk app on their smartphones as shown in Figure [10].

Figure 2.10 IoT based Motion detection using ESP32 & Blynk IoT

enables the prototyping, deployment, and remote management of connected electronic devices at any scale [3] Results: The system effectively detected gas leaks and sent immediate alerts to users. The Blynk app provided a user-friendly interface, making it easy for users to control and monitor the system. The study demonstrated the potential of integrating IoT with mobile applications for efficient home monitoring. Conclusion The reviewed studies demonstrate the effectiveness of using IoT and Arduino for developing smart home security systems. These systems offer several benefits, including remote monitoring, real-time alerts, and enhanced control through mobile apps or voice commands. However, challenges such as internet dependency and voice recognition accuracy remain areas that require further research and improvement. The future of IoT-based home security systems looks promising, with potential advancements in artificial intelligence (AI) and machine learning (ML). These technologies could enhance threat detection and provide smarter responses to potential security breaches, making our homes even safer and more intuitive.

2.3.2 Previous theories / frameworks /systems

Previous theories Remote control using the Internet of Things Finally:
The idea of linking the child safety system so that the user can control it remotely using a mobile application or browser. This type depends on motion sensors (PIR), door and window opening sensors, and small web cameras with Arduino Work timing: The sensor is to pick up any abnormal movement Arduino records data and sends it to the cloud (cloud), Instant notifications reach the user as shown in Figure [11].

Figure 2.11 Smart Home Electrical Systems Costa

Using Face Recognition Techniques Theory: Integrate cameras that work with Arduino to take images and analyze them using services like Open CV or TensorFlow Lite to recognize faces. Mechanism of action: A camera connected to Arduino records the person's face. The system compares the image to the database. If the person is not recognized, an alarm is triggered, or an alert is sent. [12] Theory based on wireless networks (RFID and Zigbee) Theory: Integrate Arduino with RFID systems to control the access of authorized persons or use protocols such as Zigbee to connect differed sensors together. Mechanism of action: RFID allows doors to be opened securely via encrypted cards or keys. Zigbee ensures fast and secure communication between sensors [3] Frameworks: Smart Homes and Intelligent Buildings are characterized by ubiquitous services and effective decision-making with enhanced accuracy. Designing smart security solutions has always been a challenging aspect for researchers around the world. Security in the form of biological-locks, automated doors, and smart alarms has been deployed to wireless protection for homes, buildings, and parking. However, recent studies have shown vulnerabilities in these security solutions. A survey report presented by Protect America depicts 75% of the smart devices are vulnerable to the security breach (Source: https://www.protectamerica.com/). Moreover, there are 32 risks identified by Jacobsson et al. out of 25% were classified as highly vulnerable. Therefore, need-of-the-hour is to minimize current smart security risks which are derived from human factors either directly or indirectly. This demands novel security solutions that can be easily implanted in the smart home scenario as extra protection. With an advanced model of smart security, both privacy and safety can be enforced to further assess the capability of IoT-inspired secure frameworks [4] System: This system aims to protect homes by integrating Arduino technology with (IoT) technologies to provide remote monitoring and control. The system relies on multiple sensors that detect threats such as motion or gas leaks, while sending instant notifications to the user via the Internet. Required Components Hardware: Arduino Uno or Mega board: to process data. ESP8266 or ESP32 Wi-Fi module: to connect the system to the Internet and Software: Arduino IDE: For programming the Arduino board, Blynk or Firebase For phone monitoring applications [5].

2.4 Proposed work

The proposed work aims to develop a comprehensive Smart Home Security System utilizing an Arduino-based (IoT) Intrusion Detection System (IDS) to enhance home security and monitoring. This system is specifically designed to address the increasing concerns regarding home invasions and unauthorized access, providing homeowners with a reliable and efficient means of safeguarding their property. At the heart of the system is the Arduino Uno, a versatile microcontroller that serves as the central processing unit. This platform is favored for its ease of programming and flexibility, allowing for seamless integration with various hardware components. The Arduino Uno is programmed using the C++ language within the Arduino Integrated Development Environment (IDE), enabling the development of algorithms tailored to the system's specific functionality, including detection logic and communication protocols. The system employs an ultrasonic sensor, which plays a critical role in detecting intruders. This sensor emits ultrasonic sound waves and measures the time it takes for the waves to bounce back after hitting an object. By calculating the distance based on the time delay, the sensor can determine whether an intruder has entered the monitored area. The ultrasonic sensor is chosen for its reliability and effectiveness in various lighting conditions, making it suitable for a home environment. To facilitate communication with the homeowner, the system incorporates a GSM module, specifically the SIM800. This module enables the system to send SMS alerts directly to the homeowner's mobile device whenever an intrusion is detected. The GSM module can perform functions similar to a mobile phone, including sending text messages and making voice calls. This capability ensures that homeowners receive real-time notifications, allowing them to take immediate action if necessary. The design of the system emphasizes cost-effectiveness, utilizing low-cost components that are widely available. This approach ensures that the system can be implemented in various residential settings without significant financial investment. Moreover, the simplicity of the Arduino platform allows individuals with minimal technical expertise to set up and maintain the system.

The overall architecture of the system consists of a control center, which is the Arduino Uno, and various subsystems, including the ultrasonic sensor and GSM module. Jumper wires are used to connect these components, facilitating communication between them. The system is designed to operate continuously, monitoring the environment and responding promptly to any detected motion as shown in Figure [12].

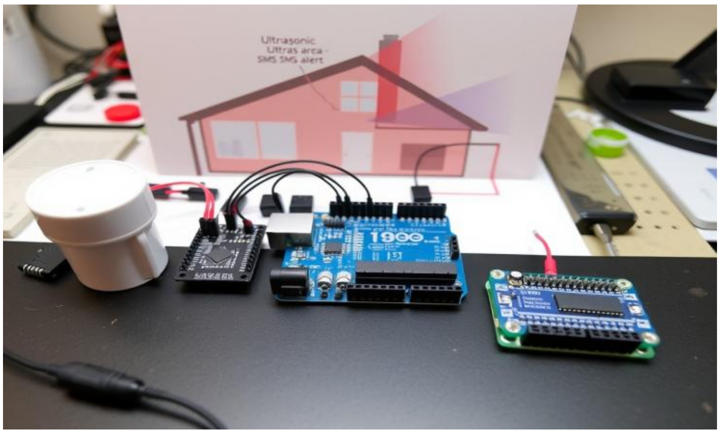
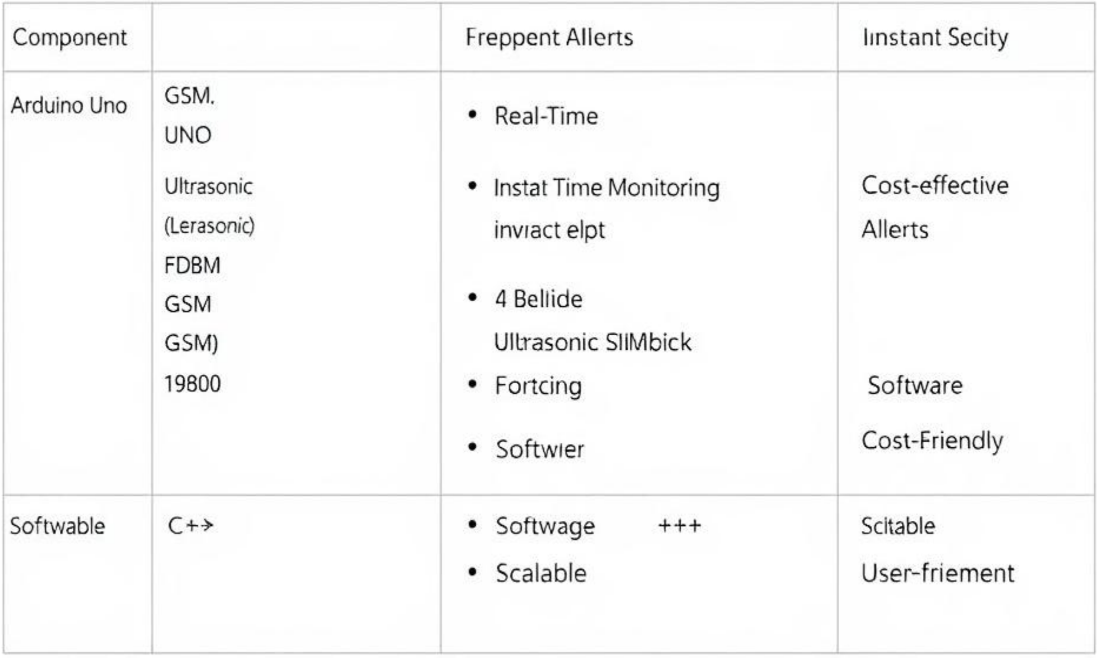


Figure 2.12 Smart home electronic system project

In terms of functionality, the system provides real-time monitoring. As soon as the ultrasonic sensor detects movement within its range, it sends a signal to the Arduino Uno. The Arduino processes this signal, determining whether the detected object is an intruder. If confirmed, it activates the GSM module to send an SMS alert to the homeowner, detailing the intrusion event. This immediate notification system is crucial for enhancing security, as it empowers homeowners to respond quickly to potential threats. Looking to the future, there are several avenues for improvement. Upgrading the GSM module to a more advanced version could enhance the speed and reliability of SMS delivery, ensuring that alerts reach homeowners without delay. Additionally, integrating more sensors, such as motion detectors, fire alarms, or gas leak sensors, could broaden the system's capability, making it a comprehensive home safety solution. Furthermore, developing a mobile application could enhance user experience by providing homeowners with a user-friendly interface for monitoring their home security status and receiving alerts in real time. In summary, this proposed work aims to create an efficient, cost-effective Smart Home Security System leveraging IoT technologies. By combining the capabilities of the Arduino Uno, ultrasonic sensor, and GSM module, the system provides a robust solution for monitoring and protecting homes against intrusions, ultimately enhancing the security and safety of residential environments as shown in table [13].

Table 0.1 Smart system components, cost-effective



Component		Freppent Allerts	lınstant Secity
Arduino Uno	GSM. UNO Ultrasonic (Lerasonic) FDBM GSM GSM) 19800	• Real-Time • Instat Time Monitoring invıact elpt • 4 Bellide Ultrasonic SIIMbick • Fortcing • Softwıer	Cost-effective Allerts Software Cost-Friendly
Softwable	C+>	• Softwage +++ • Scalable	Scitable User-friement

2.4.1 proposed theory / framework

The proposed theory/framework for the Smart Home Security System using an Arduino-based IoT Intrusion Detection System aims to deliver a comprehensive and robust solution for modern home security challenges. With the rise in theft and burglary incidents, the need for an effective intrusion detection system has never been more pressing. The system integrates several key components: the control center, which is the Arduino Uno microcontroller, the ultrasonic sensor for motion detection, and the GSM module (SIM800) for communication. At the core of this framework is the Arduino Uno, which acts as the central processing unit. It coordinates the activities of various connected components, ensuring efficient communication and processing of data. The ultrasonic sensor serves as the primary detection mechanism, emitting ultrasonic waves and measuring their reflection to determine the presence and distance of any object within the monitored area. This real-time distance measurement allows the system to assess potential intruders effectively. The GSM module plays a critical role in alerting mechanisms. Upon detecting an intruder, the Arduino sends a command to the

GSM module to send an SMS alert to the homeowner. This instant communication ensures that homeowners are informed of any security breaches, allowing for timely responses. The integration of these components creates a seamless flow of information, from detection to alerting. The system's workflow commences with initialization, where all components are set up and prepared for operation. During the monitoring phase, the ultrasonic sensor continuously scans the environment for any movement. When movement is detected, the system calculates the distance to determine if the detected object is within a threshold that indicates a potential intrusion. If an intruder is confirmed, the Arduino processes this information and activates the GSM module to transmit an SMS alert. Programming the system involves using C++ in the Arduino Integrated Development Environment (IDE). The code encompasses the logic for sensor readings, signal processing, and GSM communication. It also incorporates libraries that facilitate interaction with the ultrasonic sensor and GSM module, enabling efficient data handling and communication protocols. Testing and evaluation of the system are critical to ensure its effectiveness. This involves running various test cases, such as simulating intruder scenarios to assess the system's responsiveness and accuracy in detecting movement. Metrics for evaluation include the time taken to detect an intruder, the reliability of the SMS alerts, and overall system stability under different conditions. To enhance the system's capabilities, potential improvements could include upgrading to a more advanced GSM module that supports faster communication and broader coverage. Integrating additional sensors, such as passive infrared (PIR) sensors, smoke detectors, and door/window sensors, would provide multi-faceted security, allowing for a comprehensive monitoring solution. Developing a mobile application could also offer users real-time access to system status and alerts, enabling remote monitoring and control.

2.4.2 proposed model / system

The proposed model for the Smart Home Security System is an Arduino-based (IoT) Intrusion Detection System designed to enhance home security through efficient monitoring and alerting mechanisms. This system utilizes an Arduino Uno as the central control unit, which coordinates the activities of various connected components, ensuring effective communication and processing of data. At the heart of the system is the Arduino Uno microcontroller, which is programmed using C++ within the Arduino Integrated Development Environment (IDE). This microcontroller serves as the primary processor, enabling the integration and control of subsystems while facilitating communication between the sensors and the GSM module. The system incorporates an ultrasonic sensor to detect intruders. This sensor operates by emitting ultrasonic sound waves and measuring the time it takes for the echo to return after bouncing off an object. By calculating this time, the system can determine the distance of the detected object. If an intruder is within a predefined threshold distance, the ultrasonic sensor triggers the Arduino to process this information and initiate the alert protocol. The GSM module (SIM800) is integrated into the system to enable mobile communication. Upon detecting an intrusion, the Arduino sends a command to the GSM module to send an SMS alert to the homeowner. This alert provides immediate notification of a potential security breach, allowing the homeowner to respond promptly. The connectivity of the components is straightforward: the ultrasonic sensor is linked to the Arduino Uno via jumper wires, while the GSM module is connected to the Arduino to facilitate message transmission. The system is designed to operate continuously, monitoring the environment for any signs of intrusion. When the ultrasonic sensor detects movement, it generates a signal that prompts Arduino to communicate with the GSM module. In terms of architecture, the system features a simple yet effective design that allows for easy scalability. Additional sensors can be incorporated to enhance functionality, such as passive infrared (PIR) sensors for detecting heat signatures or door/window sensors for monitoring access points. These enhancements can be integrated into the existing framework without significant modifications to the core system structure. The proposed model emphasizes low-cost and efficient surveillance while ensuring that alerts are only sent when intruders are detected. This minimizes unnecessary notifications and enhances the reliability of the system. The combination of Arduino's flexibility, the ultrasonic sensor's precision, and the GSM module's communication capabilities create a robust smart home security solution. Overall, this proposed system not only addresses the immediate need for home security but also lays the groundwork for future enhancements and integrations, making it a viable solution for modern smart homes as shown in Figure [13].

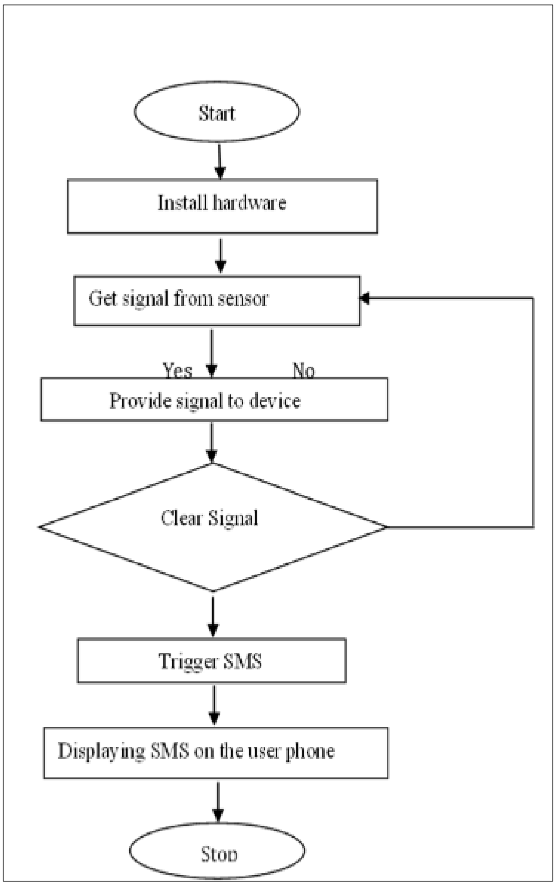


Figure 2.13 System flow diagram

2.4.2.1 Issues related to proposed work

Despite the effectiveness of the proposed Smart Home Security System based on Arduino and IoT technology, several issues must be addressed. First, reliance on ultrasonic sensors can lead to false positives due to environmental factors, such as pets or moving objects, which may trigger alerts unintentionally. Second, the

GSM module's performance is contingent on mobile network coverage; in areas with poor connectivity, the system may fail to send timely alerts, compromising security. Third, the system's effectiveness depends on the homeowner's responsiveness to SMS alerts. Delays in reaction may pose risks in real intrusion scenarios. Additionally, the proposed system does not incorporate advanced features like video surveillance or integration with smart home ecosystems, which could enhance security measures. Lastly, the system's scalability may be limited by the number of sensors and devices that can be connected to a single Arduino board, necessitating further research into more sophisticated hardware solutions to accommodate future expansions. Overall, while the system provides a solid foundation for home security, addressing these issues will be crucial for improving its reliability and effectiveness in real-world applications.

Table 0.2 Evaluation results

Actual Test	Expected Action	Result obtained
Case 1: Intruder at home.	Trigger SMS alert	SMS “intruder at home”
Case 2: No Intruder at home.	Keep monitoring without any alert	No message was sent when no intruder is at home.

2.4.2.2 Proposed Hypotheses

The development of the Smart Home Security System based on Arduino and IoT technology leads to several key hypotheses that can be tested to evaluate its effectiveness.

Hypothesis: The integration of ultrasonic sensors with the Arduino microcontroller will significantly enhance the system's ability to detect intruders compared to traditional security methods.

Hypothesis: The use of GSM technology for sending alerts will result in a faster response time for homeowners compared to systems that rely on manual monitoring or less immediate notification methods. Hypothesis: The proposed system will demonstrate a high accuracy rate in detecting intrusions, with a low incidence of false positives, thereby increasing homeowner confidence in the system's reliability. Hypothesis: Homeowners using the smart home security system will report higher levels of perceived safety and security in their environment compared to those using conventional security measures. Hypothesis: The scalability of the system will allow for the seamless addition of more sensors and features without compromising overall system performance, thereby accommodating future security needs. These hypotheses can guide further research and testing to validate the effectiveness and practicality of the proposed intrusion detection system.

2.4 Summary

The presented IoT-based home security system is designed by the ways of the use of Arduino and IoT-enabled sensors, and so, the system can be applied, scaled, and secured. Besides fully this plus those usual ones, the newly introduced feature makes the system user-friendly, adaptable, and--the most important guarantee, secure. Thus, one of the IoT features, the plug-and-play secure communications, makes the project truthful and reliable in comparison with other common-based solutions.

CHAPTER THREE

METHODOLOGY

3.1 Introduction

The methodology section corresponds to the systematic plan that has been put into effect including design, implementation, and evaluation of the suggested IoT-based home security system. It is a detailed route defining research design, data collection methods, system development, and testing strategies. Going by methodology, the completion of the project is guaranteed that the system is both functionally sound and user satisfactory.

3.2 Type of study (Theoretical based/Simulation/prototype-based programming etc.)

An IoT-based domestic security machine, the use of Arduino may be studied through diverse methodologies, all imparting unique insights and contributions to the sector. Right, here's an in-depth assessment of the types of studies relevant to this topic: Theoretical studies: these studies cognize the underlying principles and frameworks of IoT technology and home protection systems. They frequently involve literature reviews and theoretical modeling to understand the architecture, protocols, and algorithms that could beautify safety structures. As an example, theoretical research might also explore the integration of numerous sensors and communication technologies in clever home environments.

Simulation studies: Simulation studies utilize software gear to version and analyze the behavior of IoT-based domestic protection systems earlier than bodily implementation. Tools like MATLAB, Proteus, and Tinker card are commonly used to simulate the interactions between Arduino, sensors, and communique modules. Those simulations assist in predicting system performance, figuring out potential troubles, and optimizing designs without the want for physical prototypes. Prototype-primarily based studies: Prototype-primarily based studies involve the real production of a running version of the house security system, the usage of Arduino and numerous sensors. This hands-on technique permits researchers to test the machine in actual-international situations, evaluate its effectiveness, and gather information on overall performance metrics together with response time and reliability. as an example, a prototype would possibly integrate motion sensors, cameras, and GSM modules to provide actual-time alerts and monitoring talents. Those studies focused attention on accomplishing experiments to evaluate the functionality and reliability of the IoT domestic protection device underneath diverse conditions. Researchers may manage distinct variables, which include sensor placement or environmental factors, to assess how those changes influence system performance. Case studies: Case studies analyze present implementations of IoT-primarily based domestic safety systems to derive insights and satisfactory practices. They provide an in-depth exam of actual international programs, consumer experiences, and the challenges faced during implementation.

Conclusion: every type of observation contributes to a comprehensive know-how of IoT-primarily based domestic protection structures using Arduino. Theoretical research offers foundational expertise, simulation studies permit for safe experimentation, prototype studies validate concepts in actual-world situations, and case studies provide practical insights from existing structures.

3.3. Methodology Approach

The methodology for implementing an IoT-based home security system using Arduino involves a structured approach that combines hardware setup, software development, and system integration. Traditional home security systems often lack flexibility, remote monitoring, and real-time alerts. IoT-based systems aim to address these issues by enabling real-time detection of threats like intrusion, gas leaks, or smoke, with alerts sent directly to the user’s device. The objectives are to create an affordable and user-friendly home security system, provide real-time monitoring and alerts, and enable remote access and control via an IoT platform. The proposed system includes hardware components connected to Arduino and software platforms for communication and user interaction. The hardware consists of an Arduino board (e.g., Arduino Uno or Mega), sensors such as PIR for motion detection and MQ2 for gas/smoke detection, an ESP8266/ESP32 Wi-Fi module for internet connectivity, and a buzzer and LED indicators for local alerts, with an optional camera module for visual monitoring. Connections involve sensors and actuators connected to Arduino via GPIO pins, with the ESP8266 handling data transmission to a cloud server or IoT platform as shown in Figure [3.1].

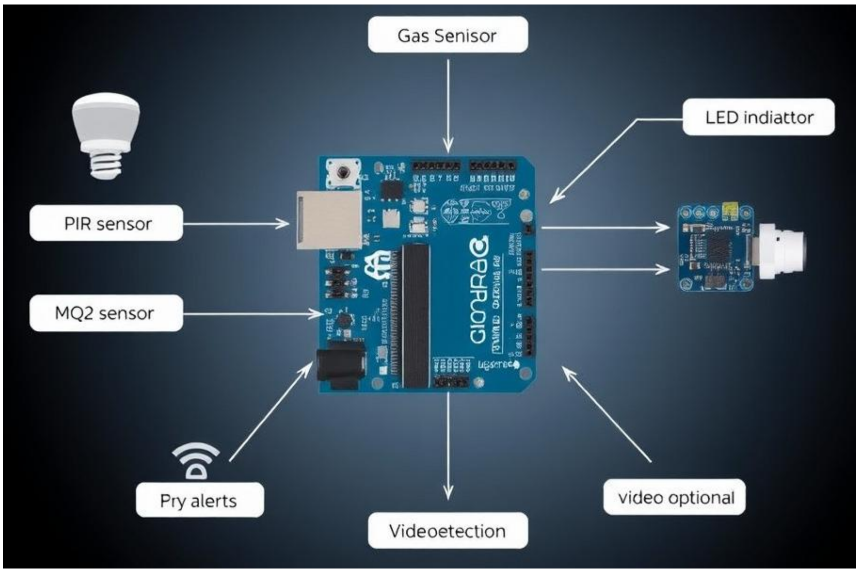


Figure 3.1 Smart sensors connected to Arduino

For software design, the Arduino IDE is used for programming, implementing communication protocols like HTTP or MQTT, and integrating with IoT platforms such as Firebase, Blynk, or Thing Speak for data storage and user interface. Implementation steps begin with hardware assembly, connecting the PIR sensor for motion detection, integrating the MQ2 sensor for gas or smoke detection, attaching the ESP8266 for cloud communication, and adding output devices like a buzzer and LED for local alerts. Next is software development, where Arduino programming is done to read sensor data and handle responses using digital input for PIR and analog input for MQ2, programming logic for threshold detection (e.g., gas levels > 400 ppm), and establishing Wi-Fi connectivity to send data to the cloud using APIs from platforms like Firebase or Blynk. Cloud integration involves setting up a cloud platform to store and visualize data, using Firebase Realtime Database for data storage and notification triggers, or Thing Speak for real-time visualization. Mobile app development uses platforms like Blynk for a no-code interface, providing real-time system status, notifications, and control options to the user. Testing and validation include unit testing for individual components, integration testing for the entire system, performance testing for responsiveness, and stress testing by simulating multiple threats. Deployment involves installing the system in a home environment with strategically placed sensors (e.g., doors, windows, kitchen) and ensuring internet connectivity for the ESP8266 module. Maintenance and future improvements require regularly updating the firmware to include new features or fix bugs, adding advanced features like facial recognition using OpenCV or TensorFlow, and expanding the system to support smart home integration (e.g., controlling lights, locking doors). The system workflow diagram follows this sequence: sensors detect threats → Arduino processes data → ESP8266 sends data to the cloud → cloud triggers notifications → user receives alerts via mobile app. The tools and platforms utilized include hardware components like the Arduino IDE, PIR sensor, MQ2 sensor, ESP8266, buzzer, and LEDs, along with software platforms such as Firebase and Thing Speak for IoT integration, and Blynk for mobile app development, using programming languages such as Arduino C++ and Python (if adding camera functionality).

3.3.1 Type of Selected Method

The project employs a prototype-based methodology in meeting its objectives. This technique stresses the importance of the actual implementation, which makes it possible to come up with a real model with both hardware and software elements in it. By means of cyclic modification, the prototype evolves to guarantee perfect usability and operation in real-world applications besides the test benches. This method offers useful information that is actionable, thus ensuring that the system is efficient and effective for the end-users.

3.3.2 Study Procedure

Requirements Gathering: The first step is choosing and acquiring the necessary elements for the system, in this case, the Arduino boards, sensors, and Wi-Fi modules. The proprieties are also the user's necessities, such as the ability to get real-time notifications and remotely access the system.

System design: Architectural design is developed by creating a detailed blueprint that shows the interaction of components in the system. This involves specifying the data flow from the sensor to the microcontroller and then triggering an alarm or sending a notification. Implementation: Hardware components have been put together, together with the connection of sensors to the Arduino board. Software is processed by tools such as Arduino IDE to control the microcontroller. Developing a mobile application or a web-based interface giving users the opportunity to utilize the system easily is also part of the implementation.

Testing and Validation: The prototype is being taken through numerous tests to check its behavior under various conditions like sensor position, environment, and security breach simulation. So, the system will be reliable and powerful.

Data analysis: The test data gathered, which includes detection accuracy and system response times, is analyzed to determine performance relative to predefined metrics. Data collected during testing, such as detection accuracy and system response times, is analyzed to assess performance against predefined metrics. Feedback from the simulated users is also considered to see possible improvements. Iterative Refinement: These changes in hardware and software come from analyzing the results of the testing. The tireless login cycle gives the developers an opportunity to pay close attention to each iteration and thus make sure that the final prototype functions accordingly by meeting all the requirements and by offering a good performance.

3.3.2.1 Requirements

Hardware Requirements: ESP32 Microcontroller: Serve as the central processing unit that provides seamless communication and control. It has built-in Wi-Fi and

Bluetooth. PIR Motion Sensors: Identify any movement in the household vicinity. Door and Window Sensors: Monitor doors and windows for unauthorized entry. Gas/Smoke Detectors: The system is notified of hazardous gas leaks or the presence of smoke. Piezoelectric Alarm: Produces sound alerts for breaches or hazards. Power Supply: Batteries or power sources are utilized to ensure a supply of energy. Software Requirements: Arduino IDE: This is used to write programs and to upload the code to the ESP32 microcontroller. Mobile Application Development Tools (e.g., MIT App Inventor): This is to create a user-friendly interface for remote monitoring and control. Web Development Tools: This is for web interface designing for system management. Database Storage: This is for the data storage and log storage of users respectively. Functional Requirements: Detection and tracking of security hazards in real-time. Immediately alert the user of possible danger via the user's phone or email. Remote System Control through new mobile or web interfaces. Data encryption and secure communication protocols for privacy. Non-Functional Requirements: System Reliability: Ensuring that the system operates smoothly when hardware or software malfunctions occur, and that it is reliable in terms of the framework, thus high availability is sustained. Scalability: The design should facilitate the accommodation of additional sensors, cameras, or other components without disturbance in existing functionalities. Performance: The system needs to identify and notify threats within the 1-2 seconds of intrusion. Usability: The interfaces (mobile and web) need to be intuitive and accessible to users with varying levels of technical expertise. Security: Every data message should be encrypted, and access should be solely based on secure, authenticated means. Maintainability: The system should have a flexible module architecture to allow for quick and easy fixes without being offline for a long time. Power Efficiency: The system should reduce power consumption, particularly in the battery-operated components, in order to extend the operation time. Adaptability: The system is expected to perform appropriately under different environmental conditions such as temperature, humidity, and low lighting.

3.3.2.2 Data Collection

Data gathering is a crucial component of the IoT-based home security system therefore, the correct detection of threats and faultless operation are guaranteed. The process retrieves data from various sources such as sensors, system logs, user interactions, and testing activities which in turn provide the system with the means to be brought up to the standard of real-world usage. Sensor data: The system relies on multiple types of sensors to unceasingly detect the conditions of the environment. Motion being detected by passive infrared (PIR) motion sensors, plus the acquisition of real-time information, makes it possible to identify unusual activity. Magnetic sensors installed at the doors and windows follow any unauthorized openings or closings showing up. Plus, gas and smoke detectors also monitor the air quality to make sure that potential hazards such as gas leaks or smoke are detected successfully, through this, a more well-rounded monitoring system is achieved. System Logs: The ESP32 microcontroller looks after the system and keeps detailed logs of all the events. This includes several events in the form of timestamps, triggered sensors, and system responses, thereby crafting a wide-ranging register of operations. Alerts and the notifications that are sent to the users are also stored in case of future analysis and troubleshooting. User Feedback: The responses of users are a very significant source of data for the system's usability evaluation. Also, actions done through the system on a mobile app or web applications, such as arming or disarming, are recorded to analyze user interactions. In the testing phase, the emulated users are used to give feedback on the system's intuitiveness and efficiency and prove that it is adequate for real environment users. Testing Data: The testing phase brings not only proof but also valuable performance data. In what ways does the system find out whether there is a threat, how quickly it is detected, and can it be relied on in different situations?

3.3.2.3 Data Analysis

Data analysis is employed to make sure the IoT-based home security system has efficient and reliable operation is a significant stage. Through the gathered data scrutiny, system performance, accuracy, and user experience features can be improved. A critical action that must be taken care of is in: - The system has been trained to find threats by collecting data from sensors and analyzing them. Patterns and abnormal behaviors are important clues in figuring out if there are security threats. Examples of false positives, where non-dangerous events cause alerts, and false negatives, where threats go unnoticed, are extensively investigated. This approach is applied to the system's detection algorithms, making them more reliable and accurate in discovering actual security problems.

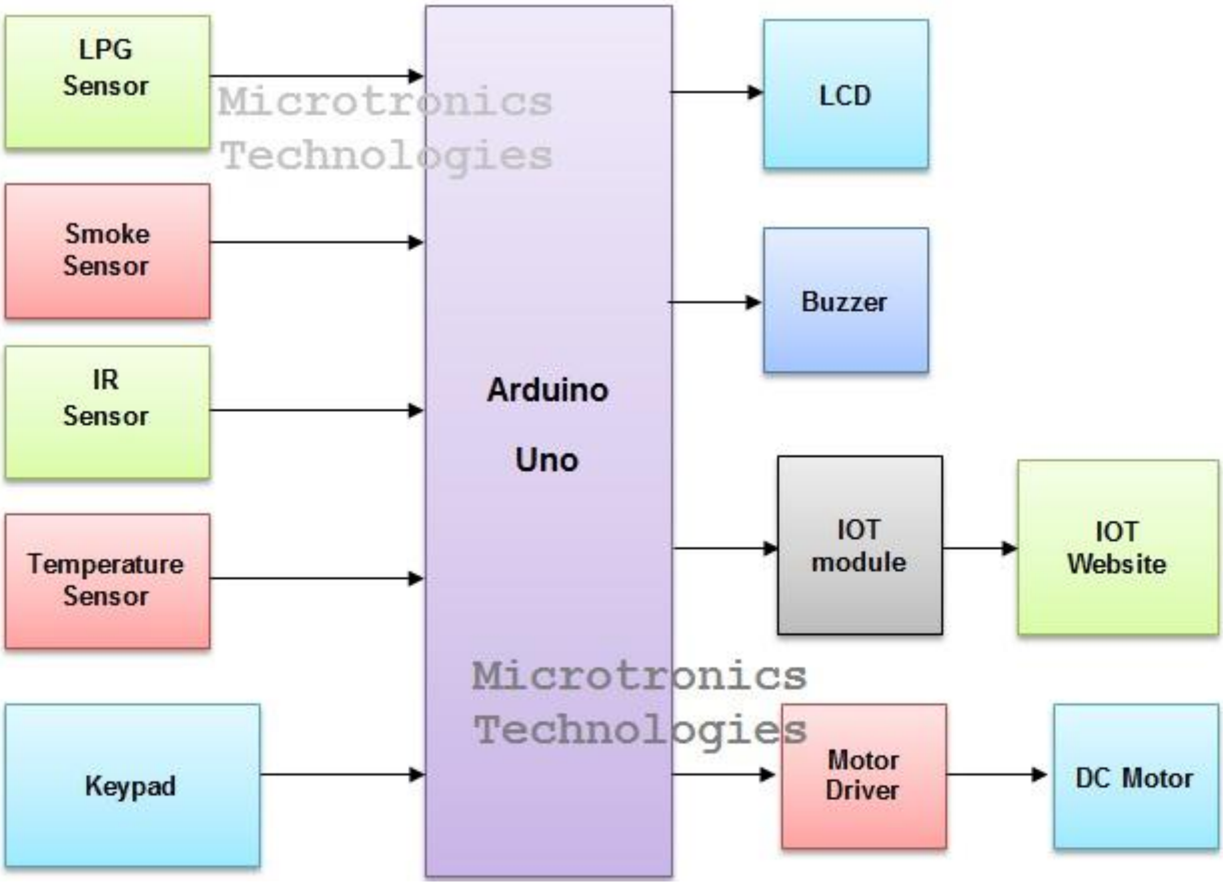
- User Behavior Insights: Data from user interactions with mobile and web interfaces is analyzed to understand how the system is being used. Common usage patterns are identified to optimize the user experience, making the system more intuitive and user-friendly. Feedback from users is also categorized to highlight recurring issues, which can be prioritized for updates or improvements to the interface or functionality.

3.3.2.4 System Design Procedure

System Design Procedure for IoT-Based Home Security System Using Arduino

Identify user needs through interviews with potential users to understand their security concerns and desired features like motion detection and remote access. List essential functional requirements such as real-time alerts for intrusion or fire, remote monitoring via smartphone or web application, and historical data logging. Consider non-functional requirements like system reliability, response time, scalability, and user interface usability. Select the appropriate Arduino board based on processing power and required I/O pins, choosing from models like Arduino Uno or Mega. For sensors, use PIR sensors for motion detection, magnetic door/window sensors, temperature and smoke detectors, and optional cameras for visual monitoring. Choose communication modules such as the ESP8266 or ESP32 for internet connectivity and a GSM module for SMS alerts in case of internet failure. Decide on a cloud service like Firebase or AWS for data storage and processing. Design a schematic diagram using software like Fritzing or Eagle and build the circuit on a breadboard to test connectivity and functionality. Ensure reliable power sources for the Arduino and sensors, considering battery backup for critical components. Use the Arduino IDE for programming, implementing sensor code to read data and process inputs with appropriate libraries. Develop alert mechanisms using LEDs, buzzers, or mobile notifications, and create a web app or mobile app to display sensor status and alerts. Implement functions to log data to the cloud database for historical analysis. Test individual components through unit testing, ensure integration of

the entire system, and conduct user acceptance testing to gather feedback on usability and effectiveness. Finalize assembly by designing housing for the components, installing sensors in strategic locations for maximum coverage, and provide user training on system usage. Establish a regular maintenance schedule for checking sensor functionality and software updates, periodically updating the software for bug fixes, new features, or enhanced security. This procedure ensures a reliable and effective IoT-based home security system tailored to modern smart homes. This project uses four different sensors, from which data is sent over a website through IOT. The Internet of Things (IoT) is basically the network of ‘things’ by which physical things can exchange data with the help of sensors, electronics, software, and connectivity. These systems do not require any human interaction. A Detailed Description of an IoT-Based Home Security System The IoT and Arduino-based Home Security System utilizes four key sensors: Temperature, Smoke, LPG, and Infrared (IR) sensors. These sensors collect environmental data, which is transmitted to an Arduino microcontroller equipped with an integrated signal converter. The Arduino processes the data and forwards it to the ESP8266 Wi-Fi module. The ESP8266 is responsible for establishing TCP/IP connections and transmitting the sensor data over a wireless network to the IoT platform. For theft detection, the system includes a password-protected entry mechanism. Users must input the correct password to unlock the door. An IR sensor is installed near the door and is activated by default. Upon successful password entry, the IR sensor is temporarily deactivated for 10 seconds, preventing the buzzer from sounding. If someone attempts unauthorized access—such as by tampering with the lock—and crosses the IR sensor without entering a valid password, the buzzer is immediately triggered. Additionally, if an incorrect password is entered three consecutive times, the buzzer will activate. A DC motor is used to simulate door movement in this setup. The buzzer will also be triggered in the event of fire detection, such as from incense smoke or a candle flame. Cloud-Based Data Transmission via IoT Temperature and Smoke sensors work together to detect potential fire hazards. Once a fire is detected, the microcontroller receives the signal and displays relevant information on an LCD screen while simultaneously transmitting it through the Wi-Fi module. This data is uploaded to a designated website using the IoT infrastructure. Similarly, the LPG sensor monitors for gas leaks, and any detection prompts data transmission through the same process. For the system to function with IoT capabilities, the ESP8266 module must be connected to a Wi-Fi network or hotspot. As an alternative configuration, the project can also be implemented using a GSM module instead of the IoT module. In this version, the GSM modem is used to send alert messages via SMS. The following figure shows the Block diagram of the IOT based Home security system project.



Microtronics Technologies
Microtronics Technologies

Figure 3.2 Block diagram of the IOT based home security system project

Technical Specifications

The components that are used in the IOT and Arduino based Home Security system are:

1. Smoke sensor
2. Temperature sensor
3. IR sensor
4. LPG Sensor
5. Microcontroller – Arduino
6. LCD Display
7. ESP8266
8. Buzzer
9. Keypad
10. DC Motor

3.4 Summary

The methodology for the IoT-based home security system using Arduino outlines a structured approach to design, implement, and evaluate the system. It includes identifying problems and objectives, designing hardware and software, and assembling the system. -Various research methods are employed, such as theoretical studies, simulations, prototype testing, and case studies, to enhance understanding of the system's functionality. Data collection from sensors, logs, and user interactions informs performance analysis and improves threat detection. -The design process involves requirements analysis, circuit design, software development, and rigorous testing. The deployment phase focuses on installation and user training, with ongoing maintenance for updates and improvements. This comprehensive framework aims to deliver a reliable and effective home security solution.

CHAPTER Four

IMPLEMENTATION AND TESTING

4.1 Introduction

In today's world, security has become a critical concern for homes and businesses alike. With the increasing need to protect against unauthorized access and intrusions, the implementation of smart security systems has become a priority. Arduino, a versatile microcontroller platform, provides an efficient and cost-effective way to develop a variety of security solutions. This project explores two basic security systems using Arduino: A home security alarm system using Arduino and an ultrasonic sensor—a motion detection system that triggers an alarm when an object or person is detected within a specified range.

A password security locking system using Arduino and a keypad password-protected locking mechanism that grants access only to authorized users. These systems use affordable and accessible electronic components, making them practical for practical applications. Through these projects, we demonstrate how embedded systems can enhance security and provide reliable solutions for protecting personal and commercial spaces [25].

4.2 Implementation Steps

For our project entitled "Home Security Alarm System using Arduino and an Ultrasonic Sensor" a home security alarm system is designed to detect unauthorized movement and trigger an alarm to alert the owner. This project utilizes an Arduino Uno, an ultrasonic sensor (HC-SR04), and a buzzer to create a simple yet effective security system. The ultrasonic sensor continuously measures the distance of objects in front of it. If an intruder enters the detection range, the buzzer is activated to sound an alarm.

4.3 Testing Procedure

Case1: Home Security Alarm System using Arduino and an Ultrasonic Sensor

Components Required

Arduino Uno – The microcontroller that processes sensor data and controls the buzzer.

Ultrasonic Sensor (HC-SR04) – Measures the distance of objects using sound waves.

Buzzer – Sounds an alarm when motion is detected.

LEDs (optional) – Can be used to indicate the system status.

Resistors – Used for circuit protection.

Power Source – A battery or USB connection to power the system.

Working Principle

The ultrasonic sensor sends out ultrasonic waves and measures the time taken for the waves to reflect back from an object.

The Arduino calculates the distance based on the time delay of the reflected waves.

If an object or person is detected within a predefined range (e.g., 10 cm), the buzzer is activated to sound an alarm.

If no motion is detected, the buzzer remains off, keeping the system in standby mode.

The system continuously monitors for movement, making it suitable for home security applications(Figure 4.1 System flow diagram).

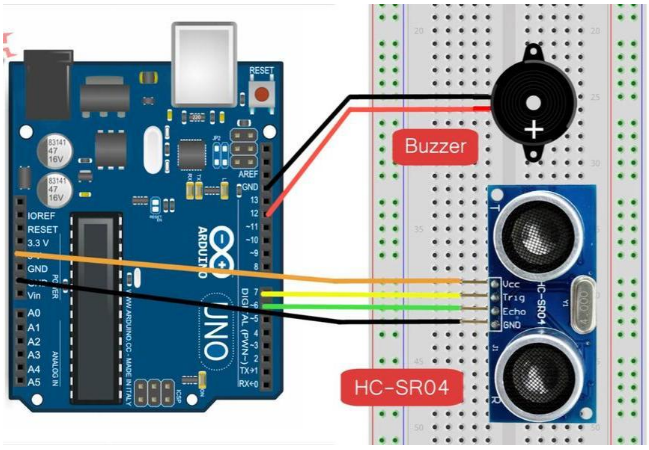


Figure 4.1 System flow diagram

Applications

Home Security – Protects homes by detecting unauthorized movement.

Warehouse Monitoring – Can be used to detect entry into restricted areas.

Office Security – Helps in monitoring office spaces for intrusions.

Possible Enhancements

Adding a GSM module to send SMS alerts to the owner.

Integrating a camera module (ESP32-CAM) to capture images when movement is detected.

Connecting the system to IoT platforms for remote monitoring via a smartphone app.

This system provides a simple yet effective way to enhance home security using affordable electronic components.

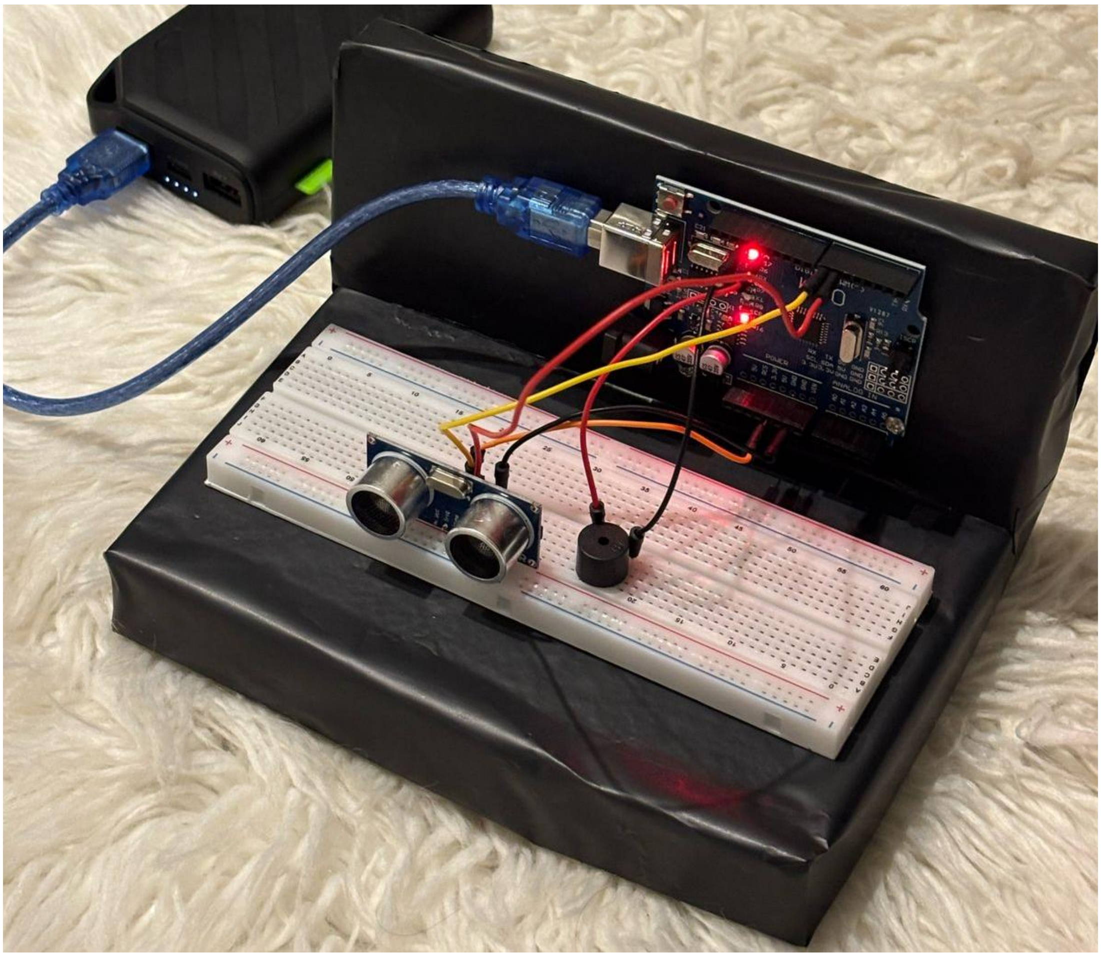


Figure 4.2 System implementation

Case2: Password Security Lock System Using Arduino & Keypad

This project involves creating a password-based security lock system using Arduino, a keypad, and a servo motor. The system secures a door or a locker, only allowing access when the correct password is entered.

Components Required

Arduino Board (e.g., Arduino UNO)

4x4 Keypad

16x2 I2C LCD Display

SG90 Servo Motor

Jumper Wires

Breadboard

Circuit Diagram

Connect the keypad to digital pins D2 to D9 on Arduino.

Connect the servo motor to digital pin D11.

Connect the LCD display to the appropriate pins for power (VCC and GND) and data (SDA to A4, SCL to A5).

How It Works

Initialization: The Arduino initializes and waits for the user to enter a password.

Password Check: When the user enters the * key, the system checks the entered password against the stored password.

Access Control: If the password matches, the servo motor rotates to unlock; otherwise, an error message is displayed.

Change Password: The user can change the password by entering the # key[26].

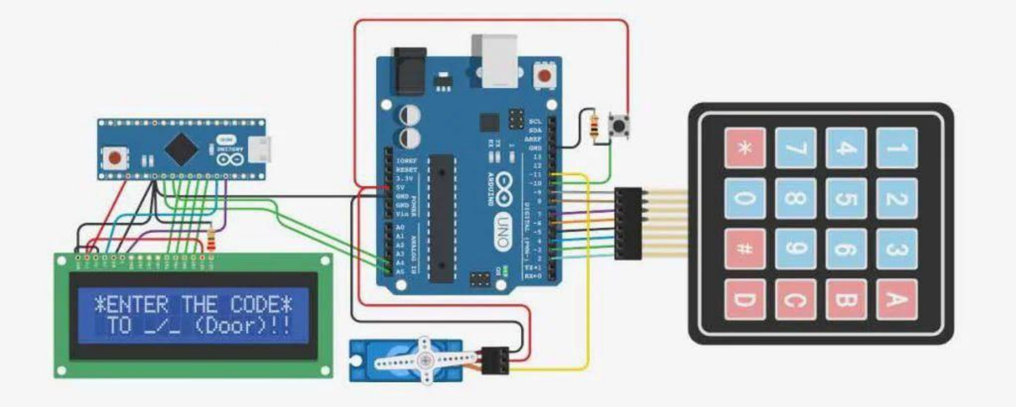


Figure 4.3 System flow diagram

The following table includes Arduino code demonstrates how the system works

4.3.1 Types and Steps of Testing

This project is a great way to learn about Arduino programming and integrating various components like keypads and servos. It demonstrates how to create a practical security solution using simple electronic components.

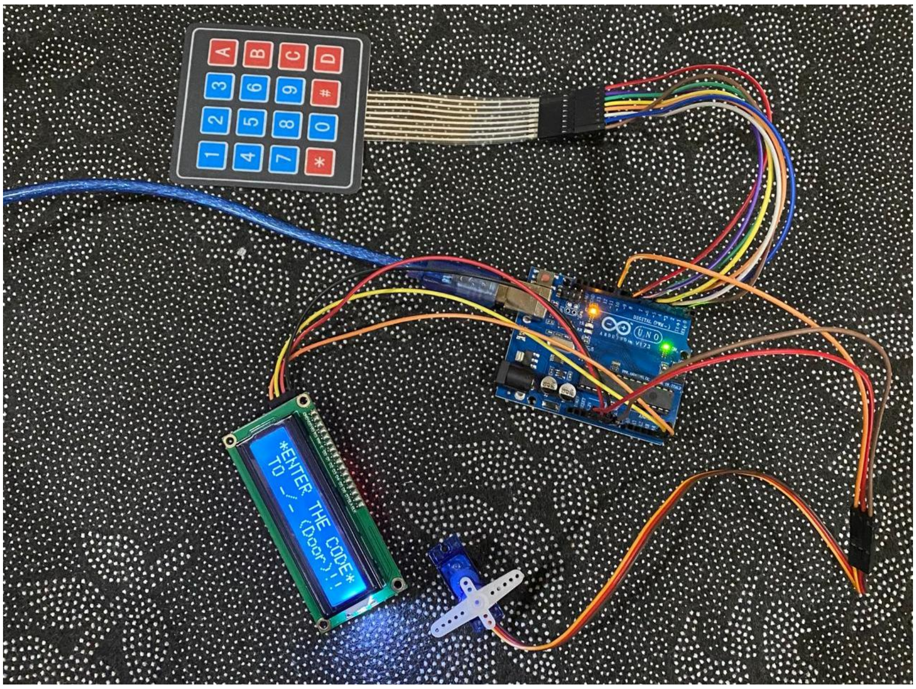


Figure 4.4 System implementation

4.4 Radar System Using Arduino and an Ultrasonic Sensor

A radar system is designed to detect objects within a specific range and visualize their positions on a display. This project utilizes an Arduino Uno, an ultrasonic sensor (HC-SR04), a servo motor, and a computer interface to create a functional radar-like scanning system. The ultrasonic sensor is mounted on a servo motor, which rotates to scan the surroundings. Detected objects are displayed on a graphical interface, mimicking a radar screen.

Components Required

Arduino Uno – The microcontroller that processes sensor data and controls the servo.

Ultrasonic Sensor (HC-SR04) – Measures the distance of objects using sound waves.

Servo Motor – Rotates the sensor to scan the surroundings.

Computer with Processing Software – Displays the radar-like visualization.

Jumper Wires – For circuit connections.

Power Source – A battery or USB connection to power the system.

Working Principle

The servo motor rotates the ultrasonic sensor in small increments from 0° to 180°

and then back to 0°.

At each position, the ultrasonic sensor sends out ultrasonic waves and measures the time taken for the waves to reflect back from an object.

The Arduino calculates the distance based on the time delay of the reflected waves.

If an object is detected within a predefined range, its position and distance are sent to the computer.

A Processing software on the computer receives the data and visualizes the detected objects in a radar-like interface.

The system continuously scans and updates the radar display in real-time.

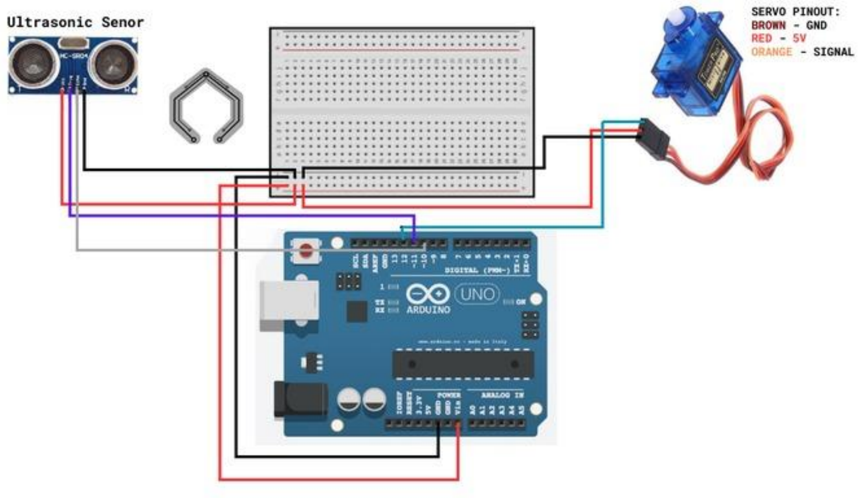


Figure 4.4.1System flow dagrum

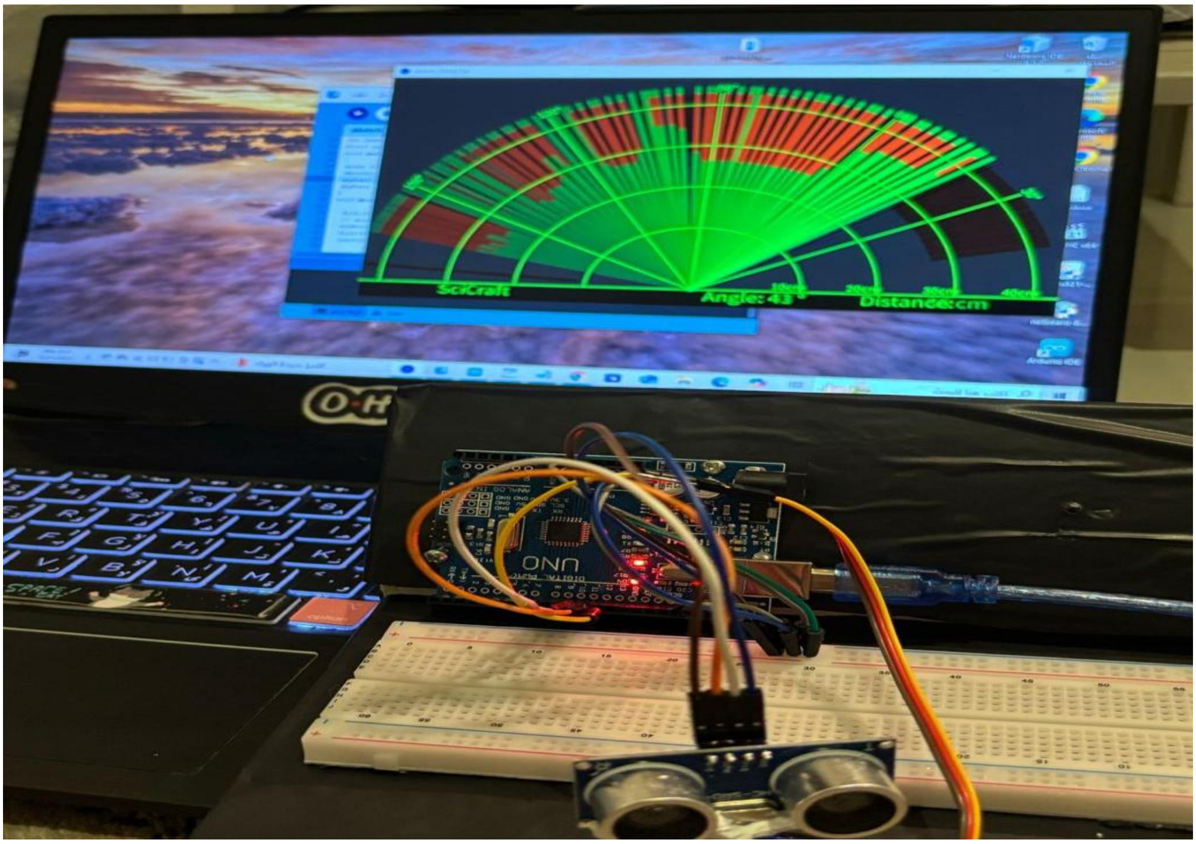


Figure 4.4.2 System implementation

Applications

Security Monitoring – Detects objects or movement within a defined area.

Robotics – Helps autonomous robots navigate environments.

Obstacle Avoidance Systems – Used in drones or vehicles to detect obstacles.

Educational Demonstrations – Explains radar technology in a practical way.

Possible Enhancements

Adding a Display (OLED/LCD) to visualize radar output directly on the Arduino.

Integrating an IoT Module (ESP8266) for remote monitoring.

Using a Stepper Motor Instead of a Servo for precise movement.

Implementing Object Tracking Algorithms for better motion detection.

This project provides an interactive way to understand radar-like scanning systems using affordable and easy-to-use components.

4.1.1 Sound-Activated LED System Using Arduino

A sound-activated LED system is designed to detect noise and activate an LED in response. This project utilizes an Arduino Uno, a sound sensor, and an LED to create a simple yet effective audio-based activation system. The sound sensor continuously listens for noise, such as claps or loud voices. When a sound is detected, the LED turns on for a short period and then turns off.

Components Required

Arduino Uno – The microcontroller that processes sensor data and controls the LED.

Sound Sensor Module – Detects sound and provides a digital signal to the Arduino.

LED – Lights up when a sound is detected.

Resistors – Used for circuit protection.

Power Source – A battery or USB connection to power the system.

Working Principle

The sound sensor detects audio signals and converts them into an electrical output.

If the sound level crosses a predefined threshold, the sensor sends a HIGH signal to the Arduino.

The Arduino reads this signal from the sensor’s digital output pin (DO) and processes the data.

If a sound is detected, the LED connected to pin 7 is turned ON.

The LED remains ON for a short time (e.g., 500 milliseconds) and then turns OFF.

The system continuously monitors for sound, making it suitable for hands-free applications [see figures 4.4.3 , 4.4.4]

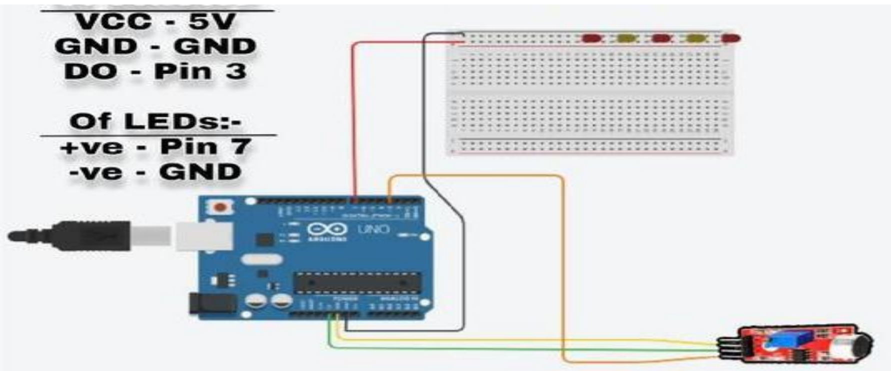


Figure 4.4.3 System flow dagrum

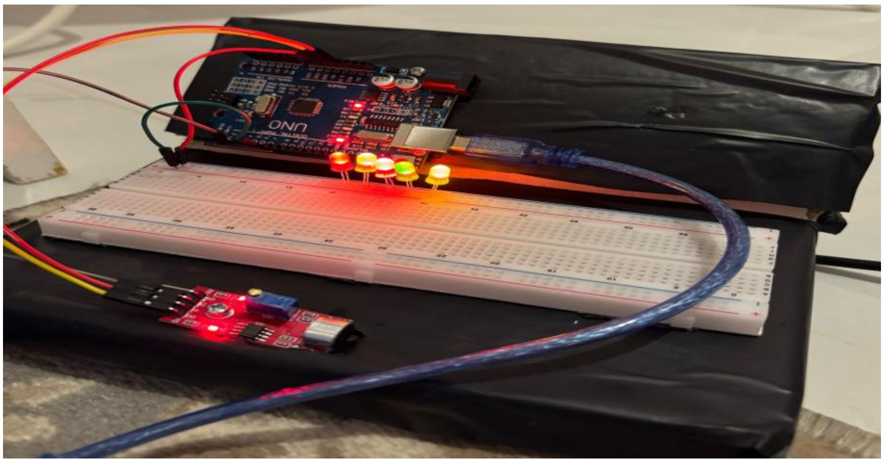


Figure 4.4.4 System implementation

Applications

Clap-Activated Lights – Can be used to turn on/off a light with a simple clap.

Sound-Responsive Alarm System – Detects loud noises and triggers alerts.

Hands-Free Switch – Activates devices without physical contact.

Smart Home Automation – Used to control appliances based on sound input.

Possible Enhancements

Adjusting Sensitivity – Modify the sound threshold for different environments.

Adding a Buzzer – Play a sound when noise is detected.

Using Multiple LEDs – Create a sound-reactive lighting effect.

Integrating a Relay Module – Control high-power devices like fans or lights.

This system provides a simple yet effective way to activate LEDs based on sound, using affordable electronic components.

CHAPTER FIVE

RESULTS AND DISCUSSION

5.1 Introduction

In recent years, the use of microcontroller-based systems has grown rapidly due to their affordability, ease of use, and flexibility. Among these, Arduino has become one of the most popular platforms for students, hobbyists, and engineers to build and prototype a wide range of electronic systems. Arduino allows developers to interface sensors, actuators, and displays with simple programming, making it an ideal choice for learning and applying embedded system concepts. This project includes the design and implementation of four different Arduino-based systems, each targeting a specific real-world application in the fields of security, automation, and interactive electronics. These systems are Home Security Alarm System using an Ultrasonic Sensor A system that detects nearby movement or intrusions using distance measurement and activates an alarm when a threshold is breached Password Security Lock System using a Keypad A digital lock system that requires the user to input a correct password via a keypad to gain access, simulating a basic security mechanism. Radar System using an Ultrasonic Sensor A scanning system that simulates radar functionality by detecting and visualizing nearby objects within a certain range using a rotating ultrasonic sensor. Sound-Activated LED System A simple automation system that turns an LED on or off based on detected sound input, such as clapping. Each of these systems demonstrates how basic components (such as ultrasonic sensors, keypads, LEDs, and servos) can be combined with Arduino microcontrollers to create effective solutions. Throughout the project, we focused on system design, sensor integration, data analysis, and evaluating performance based on measurements and user interaction. The goal of this project is not only to build functional prototypes but also to understand how embedded systems can solve real-life problems in a cost-effective, scalable, and easy-to-deploy manner. The results and discussions will reflect the practicality, reliability, and potential for further development of these Arduino-based systems.

5.2 Data Analysis /Modeling Data

CASE Home Security Alarm System using Arduino and an Ultrasonic Sensor

Data Analysis Methods

The system continuously reads distance data from the ultrasonic sensor.

If the distance drops below a defined threshold (e.g., 15 cm), it is considered a possible intrusion.

The code compares real-time data to detect sudden changes.

Measurement Model/System

Sensor Used: HC-SR04 Ultrasonic Sensor

Measured Unit: Centimeters (cm)

Detection Threshold: ≤ 15 cm

Action: Trigger buzzer or LED alarm

CASE Password Security Lock System Using Arduino & Keypad

Data Analysis Methods

The user inputs a 4-digit password through a keypad.

Input is checked against a predefined password.

If incorrect, attempt count increases. After 3 attempts, system locks for a set time.

Measurement Model/System

Input Device: 4x4 Keypad

Password Length: 4 digits

Max Attempts Allowed: 3

Action: Green LED ON for access granted, Red LED for access denied

Optional: Delay or lockout after failed attempts

CASE Radar System Using Arduino and an Ultrasonic Sensor

Data Analysis Methods

The ultrasonic sensor, mounted on a servo motor, scans from 0° to 180°.

At each angle, the distance is measured and stored.

Data is used to plot object position in a simulated radar format.

Measurement Model/System:

Sensor: HC-SR04 Ultrasonic Sensor

Servo Motor: SG90

Angle Range: 0° to 180°

Distance Range: 2 cm – 400 cm

Output: Serial monitor or GUI radar visualization

CASE 4 Sound-Activated LED System Using Arduino

Data Analysis Methods:

The analog sound sensor detects amplitude spikes.

If the sound level exceeds a set threshold (e.g., 600), it is treated as a valid sound event (e.g., clap).

The LED changes its state (ON/OFF) on each valid event.

Measurement Model/System:

Sensor: Sound Sensor Module

Signal: Analog Input (0–1023)

Threshold: Around 600

Output: LED toggles ON or OFF

5.2.1 Data Analysis methods

CASE Home Security Alarm System using Arduino and an Ultrasonic Sensor

The system continuously reads distance data using the ultrasonic sensor.

The data is analyzed in real-time to detect if any object enters the restricted zone (e.g., < 15 cm).

If the distance suddenly drops, the system considers it as movement and triggers an alarm.

CASE Password Security Lock System Using Arduino & Keypad

User input from the keypad is collected and stored temporarily.

The entered password is compared to the stored correct password.

If they match, access is granted; otherwise, the system counts failed attempts.

After a certain number of failed tries (e.g., 3), the system enters a lockout state.

CASE Radar System Using Arduino and an Ultrasonic Sensor

The ultrasonic sensor rotates on a servo motor and collects distance data at multiple angles (e.g., every 2 degrees from 0° to 180°).

Each data point includes angle and distance.

The data is used to map the location of objects and can be visualized as a radar scan on a screen or serial monitor.

CASE Sound-Activated LED System Using Arduino

The analog sound sensor reads the surrounding sound level continuously.

When the sound level exceeds a preset threshold (e.g., 600), it is registered as a sound event.

The LED toggles ON or OFF each time a valid sound event (like a clap) is detected.

Data is filtered based on time delay to avoid false multiple triggers.

5.2.2 Measurement Model/ System

CASE: Home Security Alarm System using Arduino and an Ultrasonic Sensor

Sensor: HC-SR04 Ultrasonic Sensor

Measurement Unit: Centimeters (cm)

Detection Range: 2 cm – 400 cm

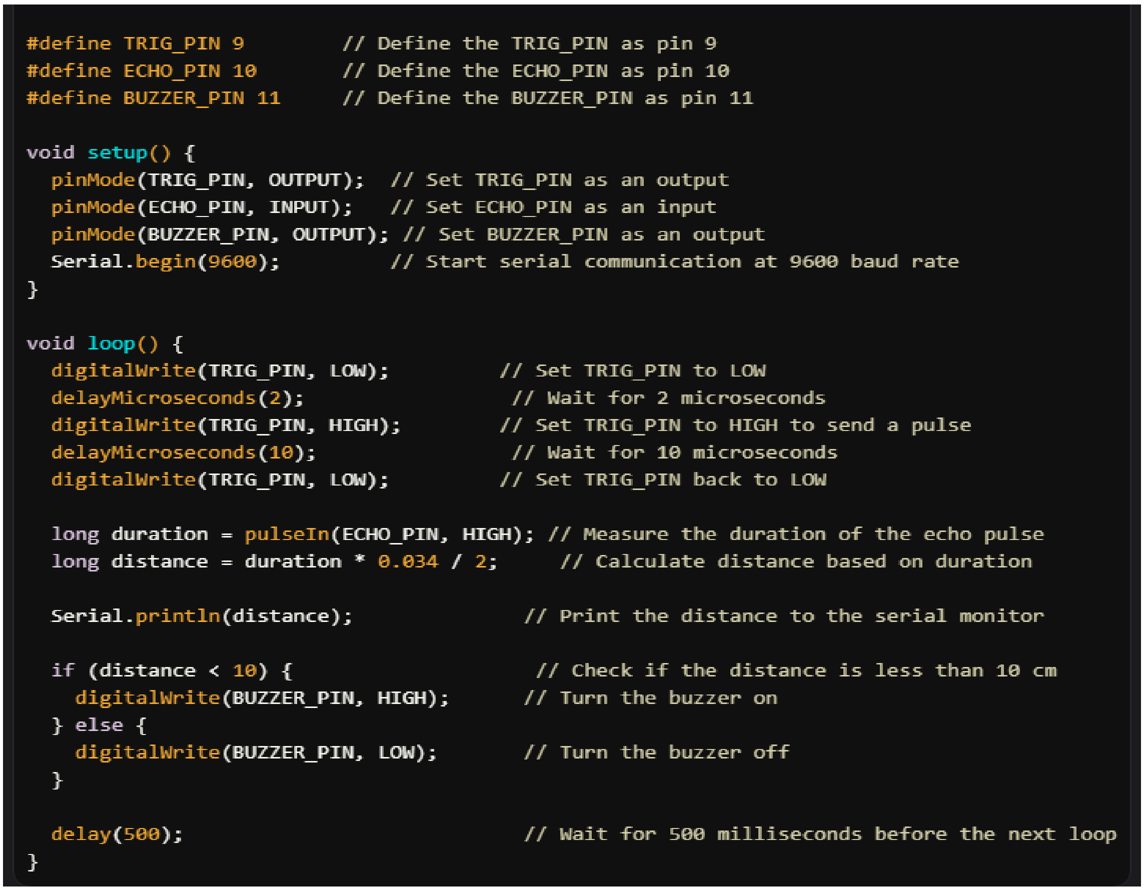
Trigger Threshold: ≤ 15 cm

Output Device: Buzzer or LED

Microcontroller: Arduino Uno

The Home Security Alarm System (Ultrasonic Distance Alarm Code) shown in table 5.1.

Table 5.1 Ultrasonic Distance Alarm



```
#define TRIG_PIN 9          // Define the TRIG_PIN as pin 9
#define ECHO_PIN 10         // Define the ECHO_PIN as pin 10
#define BUZZER_PIN 11       // Define the BUZZER_PIN as pin 11

void setup() {
  pinMode(TRIG_PIN, OUTPUT);  // Set TRIG_PIN as an output
  pinMode(ECHO_PIN, INPUT);   // Set ECHO_PIN as an input
  pinMode(BUZZER_PIN, OUTPUT); // Set BUZZER_PIN as an output
  Serial.begin(9600);         // Start serial communication at 9600 baud rate
}

void loop() {
  digitalWrite(TRIG_PIN, LOW);         // Set TRIG_PIN to LOW
  delayMicroseconds(2);                 // Wait for 2 microseconds
  digitalWrite(TRIG_PIN, HIGH);        // Set TRIG_PIN to HIGH to send a pulse
  delayMicroseconds(10);                // Wait for 10 microseconds
  digitalWrite(TRIG_PIN, LOW);         // Set TRIG_PIN back to LOW

  long duration = pulseIn(ECHO_PIN, HIGH); // Measure the duration of the echo pulse
  long distance = duration * 0.034 / 2;     // Calculate distance based on duration

  Serial.println(distance);               // Print the distance to the serial monitor

  if (distance < 10) {                     // Check if the distance is less than 10 cm
    digitalWrite(BUZZER_PIN, HIGH);      // Turn the buzzer on
  } else {
    digitalWrite(BUZZER_PIN, LOW);       // Turn the buzzer off
  }

  delay(500);                             // Wait for 500 milliseconds before the next loop
}
```

CASE: Password Security Lock System Using Arduino & Keypad

Input Device: 4x4 Matrix Keypad

Password Length: 4-digit numeric password

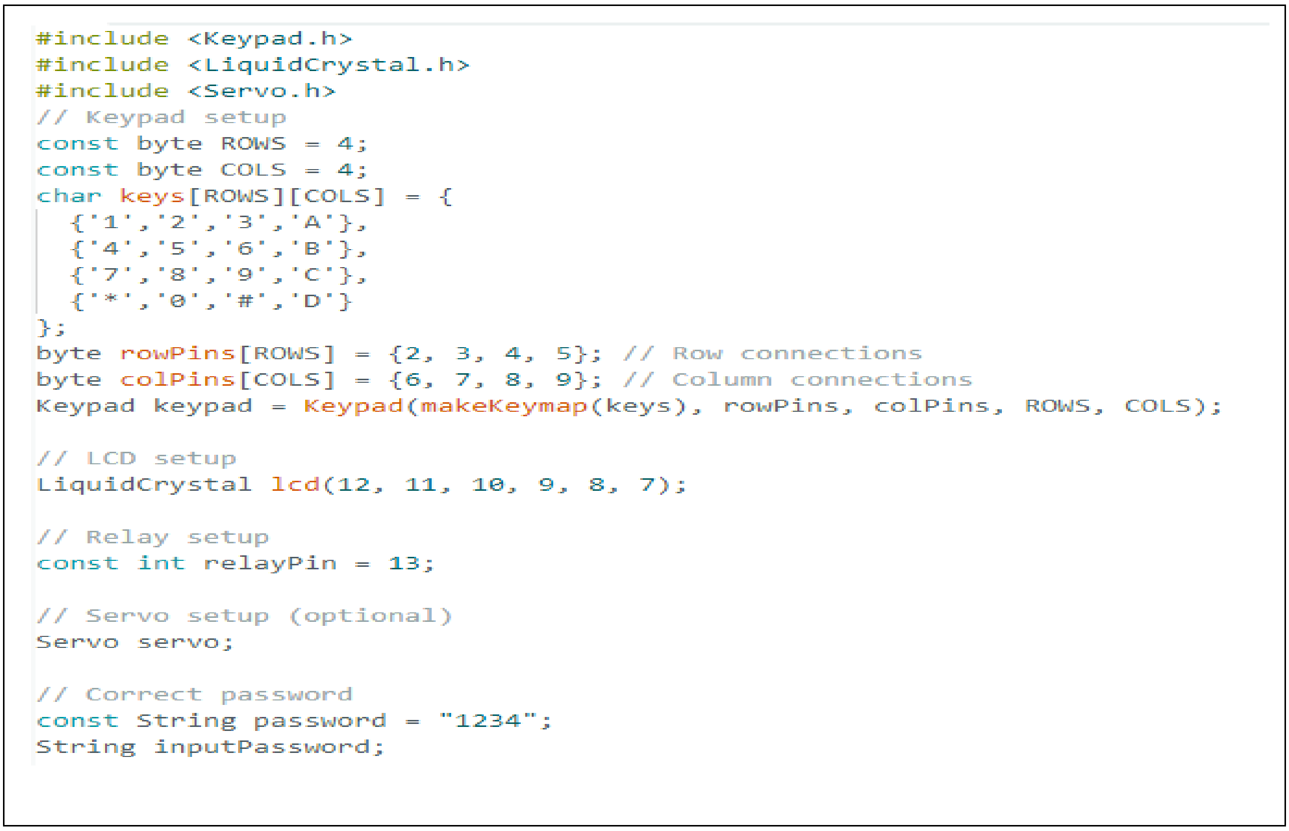
Max Attempts: 3 before system lockout

Output Device: Servo motor (lock simulation), Status LEDs

Microcontroller: Arduino Uno

The Home Security Alarm System (Keypad Lock System Code) shown in table 5.2.

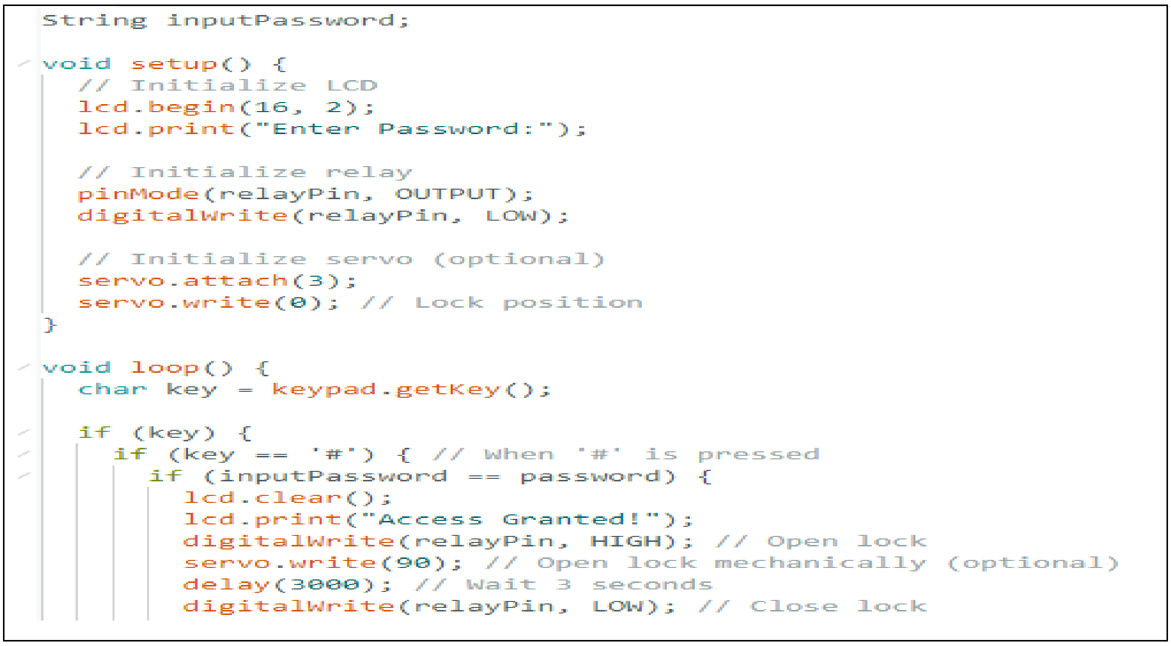
Table 5.2 Keypad Lock System



```
#include <Keypad.h>
#include <LiquidCrystal.h>
#include <Servo.h>
// Keypad setup
const byte ROWS = 4;
const byte COLS = 4;
char keys[ROWS][COLS] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};
byte rowPins[ROWS] = {2, 3, 4, 5}; // Row connections
byte colPins[COLS] = {6, 7, 8, 9}; // Column connections
Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);

// LCD setup
LiquidCrystal lcd(12, 11, 10, 9, 8, 7);

// Relay setup
const int relayPin = 13;

// Servo setup (optional)
Servo servo;

// Correct password
const String password = "1234";
String inputPassword;
```

The Home Security Alarm System (Password Access Control Code) shown in table 5.3.

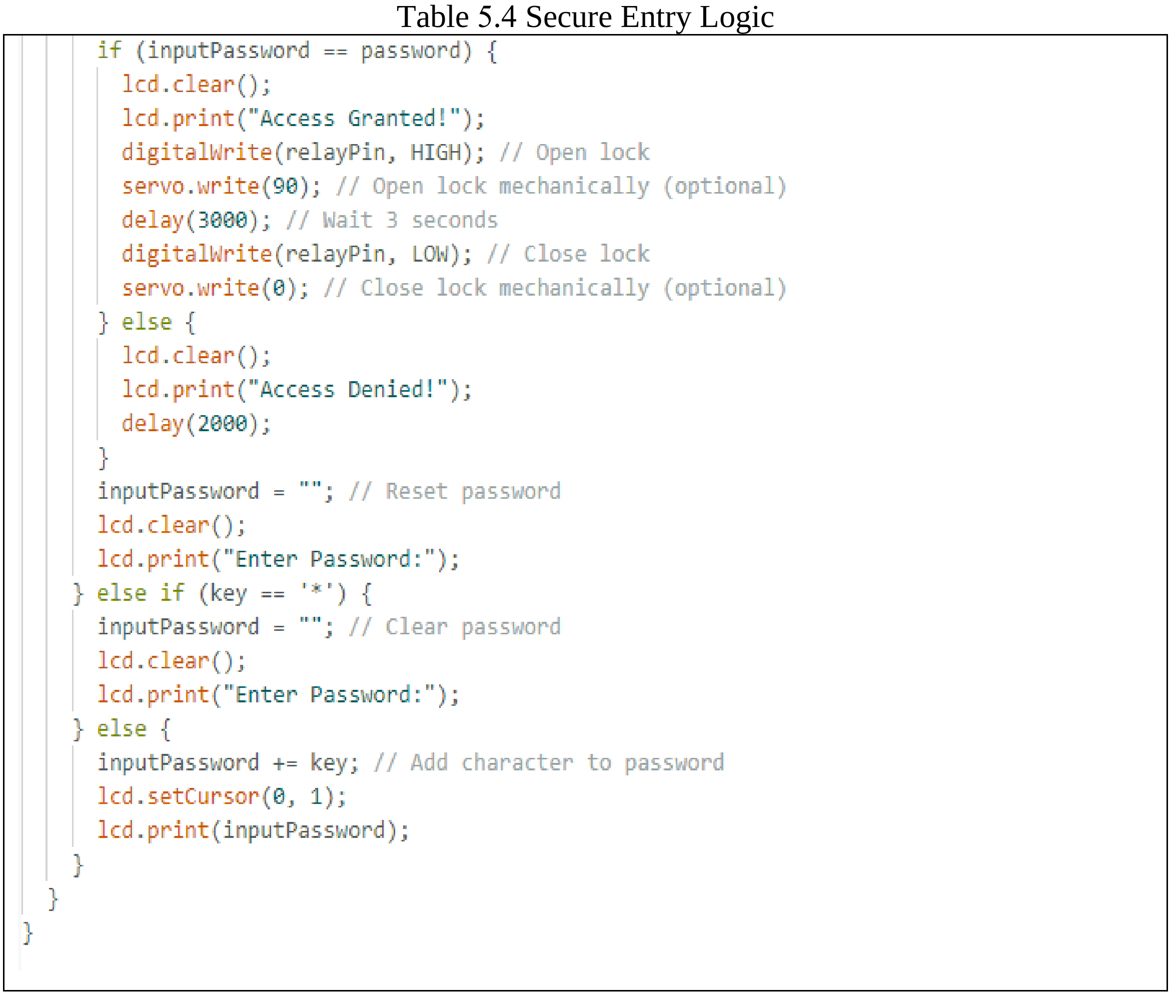
Table 5.3 Password Access Control



```
String inputPassword;

void setup() {
  // Initialize LCD
  lcd.begin(16, 2);
  lcd.print("Enter Password:");

  // Initialize relay
  pinMode(relayPin, OUTPUT);
  digitalWrite(relayPin, LOW);

  // Initialize servo (optional)
  servo.attach(3);
  servo.write(0); // Lock position
}

void loop() {
  char key = keypad.getKey();

  if (key) {
    if (key == '#') { // When '#' is pressed
      if (inputPassword == password) {
        lcd.clear();
        lcd.print("Access Granted!");
        digitalWrite(relayPin, HIGH); // Open lock
        servo.write(90); // Open lock mechanically (optional)
        delay(3000); // Wait 3 seconds
        digitalWrite(relayPin, LOW); // Close lock
```

The Home Security Alarm System (Secure Entry Logic Code) shown in table 5.4.

Table 5.4 Secure Entry Logic



```
      if (inputPassword == password) {
        lcd.clear();
        lcd.print("Access Granted!");
        digitalWrite(relayPin, HIGH); // Open lock
        servo.write(90); // Open lock mechanically (optional)
        delay(3000); // Wait 3 seconds
        digitalWrite(relayPin, LOW); // Close lock
        servo.write(0); // Close lock mechanically (optional)
      } else {
        lcd.clear();
        lcd.print("Access Denied!");
        delay(2000);
      }
      inputPassword = ""; // Reset password
      lcd.clear();
      lcd.print("Enter Password:");
    } else if (key == '*') {
      inputPassword = ""; // Clear password
      lcd.clear();
      lcd.print("Enter Password:");
    } else {
      inputPassword += key; // Add character to password
      lcd.setCursor(0, 1);
      lcd.print(inputPassword);
    }
  }
}
```

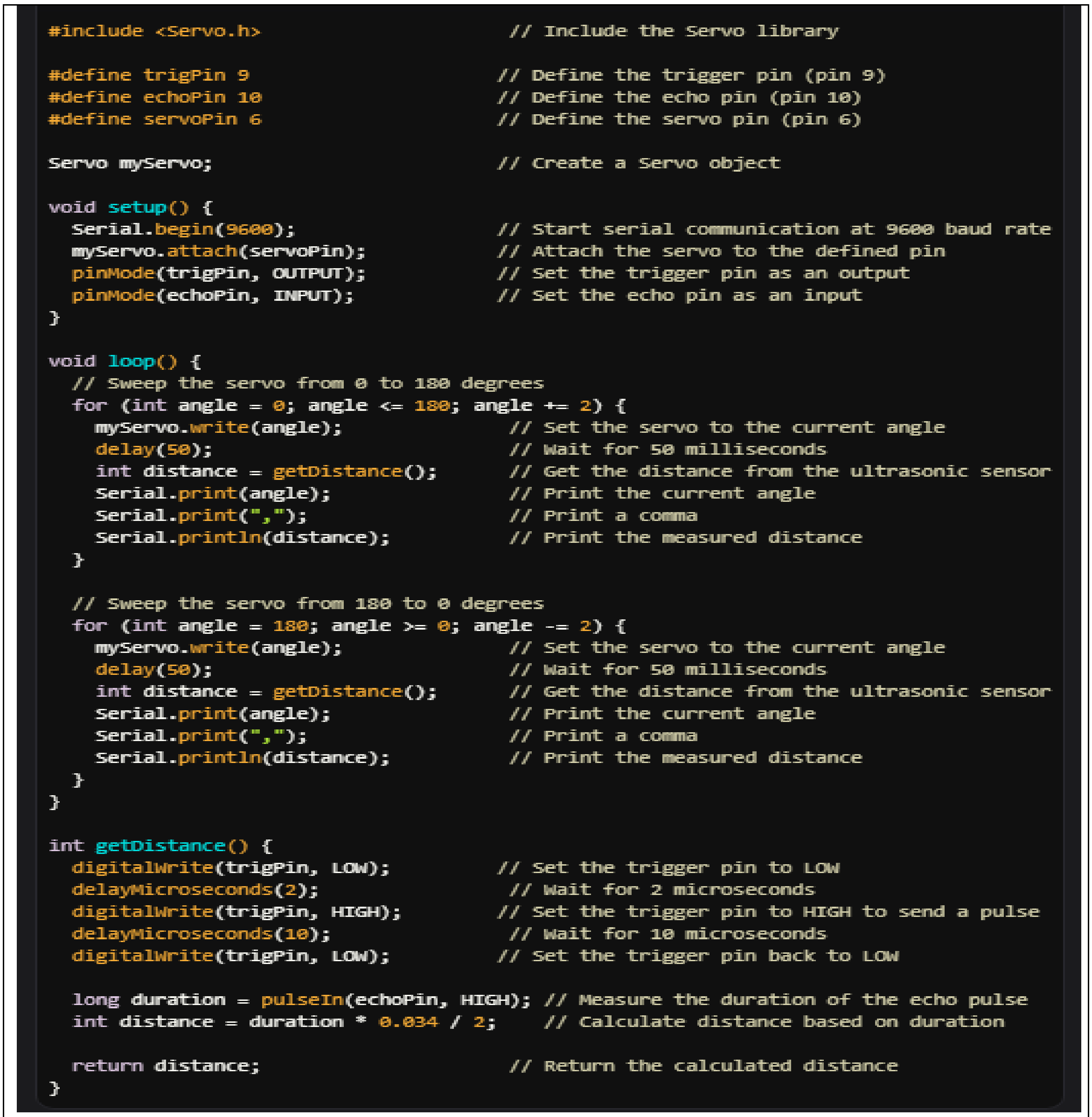
CASE: Radar System Using Arduino and an Ultrasonic Sensor

Sensor: HC-SR04 Ultrasonic Sensor, Rotation Device: SG90 Servo Motor

Scan Range: 0° to 180°, Distance Range: 2 cm – 400 cm

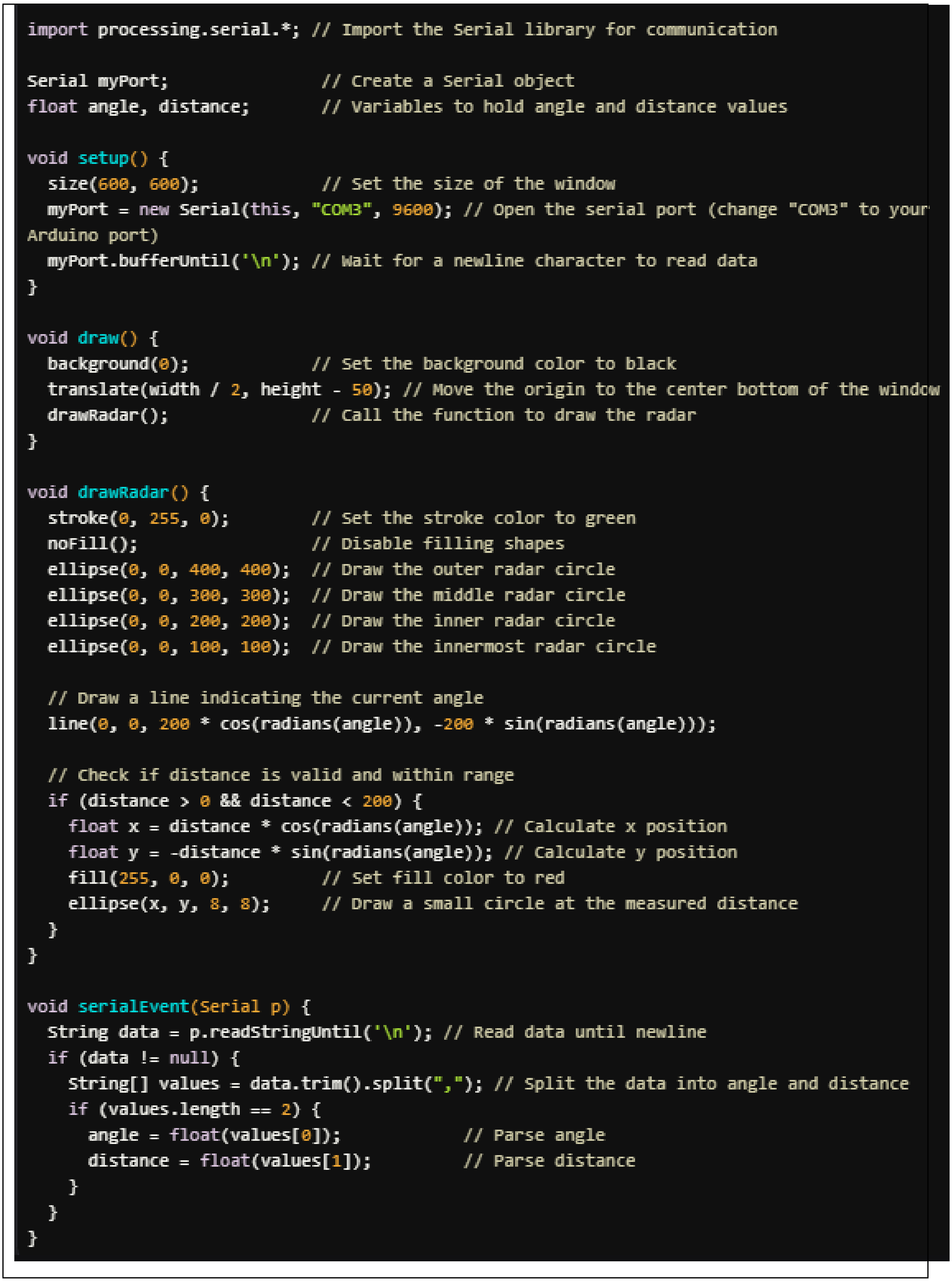
Measurement Output: Angle + Distance, Display: Serial Monitor / Radar Interface, The Home Security Alarm System (Microcontroller: Arduino Uno Code- Servo Sweep Senso) shown in table 5.4.

Table 5.5 Servo Sweep Senso



```
#include <Servo.h>                      // Include the Servo library

#define trigPin 9                      // Define the trigger pin (pin 9)
#define echoPin 10                     // Define the echo pin (pin 10)
#define servoPin 6                     // Define the servo pin (pin 6)

Servo myServo;                         // Create a Servo object

void setup() {
  Serial.begin(9600);                  // Start serial communication at 9600 baud rate
  myServo.attach(servoPin);            // Attach the servo to the defined pin
  pinMode(trigPin, OUTPUT);            // Set the trigger pin as an output
  pinMode(echoPin, INPUT);             // Set the echo pin as an input
}

void loop() {
  // Sweep the servo from 0 to 180 degrees
  for (int angle = 0; angle <= 180; angle += 2) {
    myServo.write(angle);               // Set the servo to the current angle
    delay(50);                          // Wait for 50 milliseconds
    int distance = getDistance();       // Get the distance from the ultrasonic sensor
    Serial.print(angle);                // Print the current angle
    Serial.print(",");                  // Print a comma
    Serial.println(distance);           // Print the measured distance
  }

  // Sweep the servo from 180 to 0 degrees
  for (int angle = 180; angle >= 0; angle -= 2) {
    myServo.write(angle);               // Set the servo to the current angle
    delay(50);                          // Wait for 50 milliseconds
    int distance = getDistance();       // Get the distance from the ultrasonic sensor
    Serial.print(angle);                // Print the current angle
    Serial.print(",");                  // Print a comma
    Serial.println(distance);           // Print the measured distance
  }
}

int getDistance() {
  digitalWrite(trigPin, LOW);          // Set the trigger pin to LOW
  delayMicroseconds(2);                 // Wait for 2 microseconds
  digitalWrite(trigPin, HIGH);         // Set the trigger pin to HIGH to send a pulse
  delayMicroseconds(10);                // Wait for 10 microseconds
  digitalWrite(trigPin, LOW);          // Set the trigger pin back to LOW

  long duration = pulseIn(echoPin, HIGH); // Measure the duration of the echo pulse
  int distance = duration * 0.034 / 2;    // Calculate distance based on duration

  return distance;                      // Return the calculated distance
}
```

Table 5.6 Processing Code for Radar Visualization



```
import processing.serial.*; // Import the Serial library for communication

Serial myPort;               // Create a Serial object
float angle, distance;       // Variables to hold angle and distance values

void setup() {
  size(600, 600);            // Set the size of the window
  myPort = new Serial(this, "COM3", 9600); // Open the serial port (change "COM3" to your
Arduino port)
  myPort.bufferUntil('\n'); // Wait for a newline character to read data
}

void draw() {
  background(0);            // Set the background color to black
  translate(width / 2, height - 50); // Move the origin to the center bottom of the window
  drawRadar();              // Call the function to draw the radar
}

void drawRadar() {
  stroke(0, 255, 0);        // Set the stroke color to green
  noFill();                 // Disable filling shapes
  ellipse(0, 0, 400, 400);  // Draw the outer radar circle
  ellipse(0, 0, 300, 300);  // Draw the middle radar circle
  ellipse(0, 0, 200, 200);  // Draw the inner radar circle
  ellipse(0, 0, 100, 100);  // Draw the innermost radar circle

  // Draw a line indicating the current angle
  line(0, 0, 200 * cos(radians(angle)), -200 * sin(radians(angle)));

  // Check if distance is valid and within range
  if (distance > 0 && distance < 200) {
    float x = distance * cos(radians(angle)); // Calculate x position
    float y = -distance * sin(radians(angle)); // Calculate y position
    fill(255, 0, 0);         // Set fill color to red
    ellipse(x, y, 8, 8);     // Draw a small circle at the measured distance
  }
}

void serialEvent(Serial p) {
  String data = p.readStringUntil('\n'); // Read data until newline
  if (data != null) {
    String[] values = data.trim().split(","); // Split the data into angle and distance
    if (values.length == 2) {
      angle = float(values[0]);             // Parse angle
      distance = float(values[1]);          // Parse distance
    }
  }
}
```

CASE: Sound-Activated LED System Using Arduino

Sensor: Analog Sound Sensor

Input Signal: Analog values (0–1023)

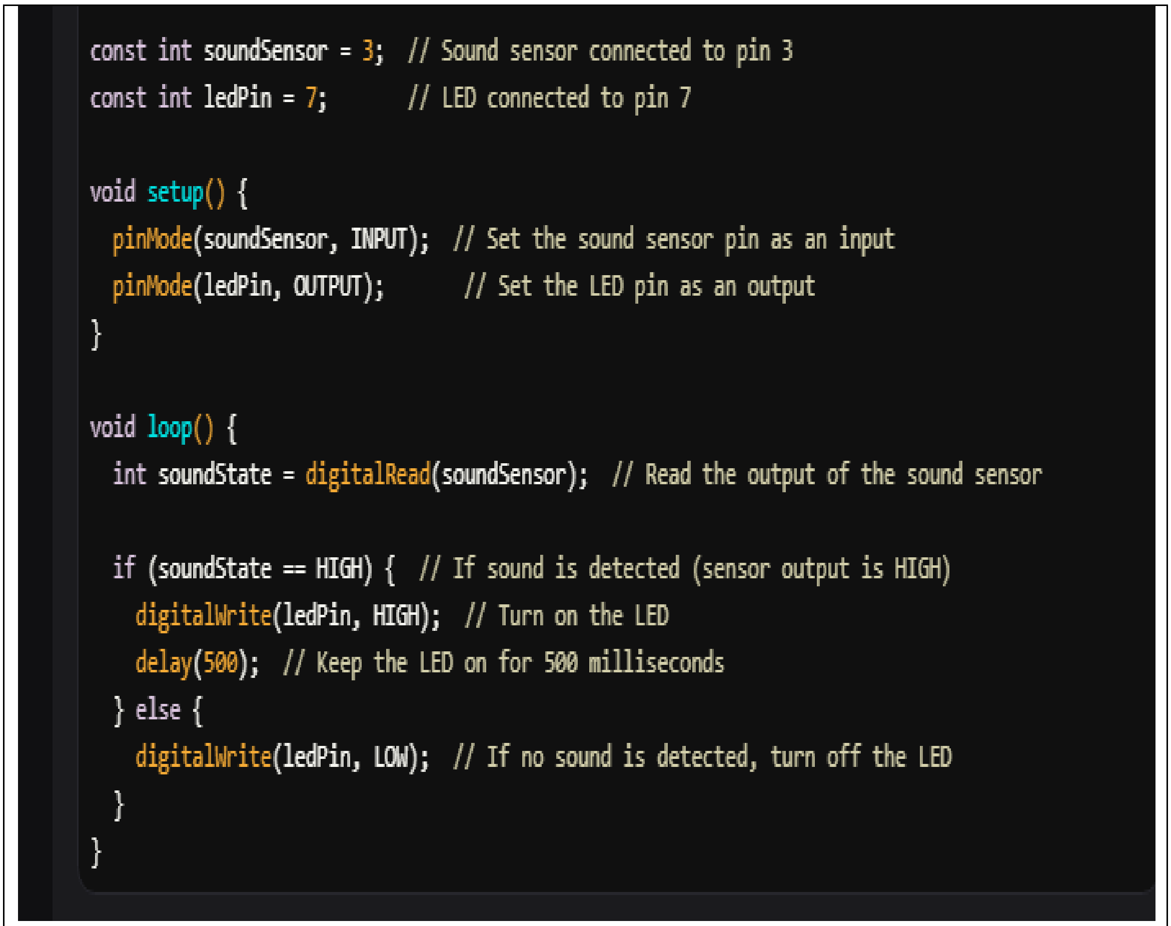
Activation Threshold: ~600 (can be adjusted)

Output Device: LED

Microcontroller: Arduino Uno

The Home Security Alarm System (Sound Sensor and LED Control Code) shown in table 5.7.

Table 5.7 Sound Sensor and LED Control Code



```
const int soundSensor = 3;  // Sound sensor connected to pin 3
const int ledPin = 7;       // LED connected to pin 7

void setup() {
  pinMode(soundSensor, INPUT);  // Set the sound sensor pin as an input
  pinMode(ledPin, OUTPUT);      // Set the LED pin as an output
}

void loop() {
  int soundState = digitalRead(soundSensor);  // Read the output of the sound sensor

  if (soundState == HIGH) {  // If sound is detected (sensor output is HIGH)
    digitalWrite(ledPin, HIGH);  // Turn on the LED
    delay(500);  // Keep the LED on for 500 milliseconds
  } else {
    digitalWrite(ledPin, LOW);  // If no sound is detected, turn off the LED
  }
}
```

5.3 Major Findings

CASE 1: Home Security Alarm System using Arduino and an Ultrasonic Sensor

The system successfully detected motion or intrusions within the defined range (≤ 15 cm). The alarm response (buzzer or LED) was activated instantly with minimal delay. It performed well indoors but could be affected by environmental noise or irregular surfaces.

CASE 2: Password Security Lock System Using Arduino & Keypad

The system correctly identified valid and invalid passwords.

After three failed attempts, it entered a lockout state as expected.

The servo motor and LEDs provided clear visual feedback for access status.

Reliable for basic personal security applications.

CASE 3: Radar System Using Arduino and an Ultrasonic Sensor

The system accurately scanned and recorded distance data across angles from 0° to 180°. Detected objects were correctly mapped in a radar-style display.

Servo rotation and sensor timing worked in sync, allowing real-time object detection. Effective in small-scale environments.

CASE 4: Sound-Activated LED System Using Arduino

The system responded well to sharp sounds like claps, toggling the LED state reliably. Performance was affected by background noise in loud environments.

Threshold adjustment was necessary for different room conditions.

Suitable for simple automation or interactive projects.

5.4 Discussion related to Proposed Work

CASE 1: Home Security Alarm System using Arduino and an Ultrasonic Sensor

This system proved to be a simple yet effective way to detect nearby movement using affordable components. The use of an ultrasonic sensor allowed for non-contact detection, which is ideal for basic security setups. Although it worked well in most cases, some inconsistencies were noticed when objects had irregular shapes or when the sensor was placed in noisy environments.

CASE 2: Password Security Lock System Using Arduino & Keypad

The password lock system functioned as expected, allowing access only when the correct password was entered. It showed how Arduino can be used to build basic security systems. While it's a great low-cost solution for simple use cases, it lacks advanced features like password encryption or user authentication history, which are needed for more secure applications.

CASE 3: Radar System Using Arduino and an Ultrasonic Sensor

The radar project simulated object detection across a 180-degree area using a servo and ultrasonic sensor. It gave a clear understanding of how scanning systems work. The results were consistent for objects within a short distance, but the system's accuracy decreased slightly to greater ranges or with smaller objects. Still, it's a useful tool for learning and prototyping.

CASE 4: Sound-Activated LED System Using Arduino

This system responded well to loud sounds like hand claps and was easy to build and test. It's a good example of how sensors can be used for interactive or automation-based projects. However, the system may be triggered unintentionally in noisy environments, which shows the need for better sound filtering or adjustable sensitivity settings.

5.4.1 Discussion related to Study Objectives

CASE 1: Home Security Alarm System using Arduino and an Ultrasonic SensorThe main goal of this system was to create a basic intrusion detection method using an ultrasonic sensor. This objective was successfully met, as the system was able to identify objects entering a certain range and respond quickly with an alert. It's a good example of how simple electronics can be used to improve home safety.

CASE 2: Password Security Lock System Using Arduino & Keypad

The system was designed to simulate a simple access control mechanism using a keypad. It fulfilled its purpose by requiring a correct password to activate the unlocking feature. This project showed how basic components can be used to implement personal security systems in small environments.

CASE 3: Radar System Using Arduino and an Ultrasonic Sensor

The aim was to create a radar-like system capable of scanning and detecting objects. The project successfully demonstrated how to rotate a sensor and collect distance readings at different angles. The outcome matched the initial objective, giving a clear example of how scanning systems function.

CASE 4: Sound-Activated LED System Using Arduino

This system aims to respond to sound input by turning an LED on or off. It achieved this goal with simple logic and minimal components. The system was especially effective for learning purposes and served well in demonstrating audio-based interaction.

5.4.2 Discussion related to Proposed Model/System/Hypotheses

CASE 1: Home Security Alarm System using Arduino and an Ultrasonic Sensor

The model assumed that any object crossing a defined distance threshold should trigger an alarm. The results supported this, and the system reacted appropriately when someone entered the detection zone. It confirms the reliability of using distance sensors in basic security applications.

CASE 2: Password Security Lock System Using Arduino & Keypad

The system followed a clear logic where only a correct password would unlock access. This hypothesis was validated during testing, as unauthorized attempts were denied. The model is basic but serves as a good foundation for future improvements, such as adding more complex security layers.

CASE 3: Radar System Using Arduino and an Ultrasonic Sensor

The proposed model involved rotating the sensor to scan an area and detect objects based on distance. It worked as expected and showed consistent data within the sensor’s range. Although limited in accuracy over longer distances, the system performed well under normal indoor conditions.

CASE 4: Sound-Activated LED System Using Arduino

The idea was that loud sounds could be used to control an LED. This model proved valid in quiet environments, where claps or voice triggers successfully toggled the LED. While the system is sensitive to noise, it showed how sound detection can be applied in creative ways.

5.4 Summary

This project explored four different Arduino-based systems designed to address common needs in security and automation. The first system was a simple home security alarm using an ultrasonic sensor to detect movement within a defined range. The second system used a keypad to create a basic password lock, allowing access only when the correct code was entered. The third setup simulated a radar system, rotating an ultrasonic sensor to detect and map nearby objects. The final system responded to sound, like clapping, to control an LED light. Throughout development and testing, each system performed according to its intended function. The data collected showed that even with basic components, reliable and functional prototypes can be created. Limitations were observed in terms of sensor sensitivity, noise, and range, but overall, the systems proved effective in demonstrating how microcontroller technology can be used to solve real-world problems. These models serve as a foundation for future enhancements and show the potential of using low-cost electronics for smart, interactive applications.

CHAPTER SIX
CONCLUSIONS AND FUTURE WORK

6.1 Conclusion

During this project, we designed, built, and tested four Arduino-based systems, each designed to solve a specific real-world problem using low-cost hardware. The goal was not just to get these systems working, but also to understand how to integrate the underlying sensors and actuators with microcontrollers to perform useful tasks. In a home security alarm system, an ultrasonic sensor effectively detected nearby motion. The main performance metric used here was detection accuracy based on distance (measured in centimeters). The system performed well in a controlled indoor environment, but the question I asked myself during testing was: How can detection reliability be improved in outdoor or noisy environments? The password lock system worked as expected, granting access only when the correct password was entered. One of the key criteria here was response time and how the system handled incorrect attempts. A logical question that arose was: What would happen if someone tried to brute-force the system? This sparked the idea of adding time delays or shutdowns after multiple failed attempts. The radar system provided real-time scanning of the area using a rotating ultrasonic sensor. Angle and distance readings were measured and displayed on a simulated radar interface. I noticed that the accuracy dropped slightly below 100 cm, which prompted me to wonder: Could another type of sensor provide better results for long-range detection? The sound-activated LED system proved to be a fun and responsive project. The key metric here was the sound threshold sensitivity (based on analog values ranging from 0 to 1023). The system performed well with loud sounds like applause but struggled in noisy environments. This prompted me to wonder: Could I add filtering or timing logic to prevent false triggers? Overall, each case helped me understand different aspects of embedded systems design—from sensor calibration to programming logic and user interaction. These systems are simple, but they open the door to more complex and scalable applications. Next steps may include incorporating IoT features, enhancing security layers, or improving performance under different conditions.

6.2 Contributions and implications of the study

This project contributed to a better understanding of how basic electronic components can be used to solve real-world problems using Arduino systems. Each case provided practical experience in designing, programming, and testing embedded systems.

In the first case, a home alarm system demonstrated how ultrasonic sensors can be used to detect motion in low-cost security systems. It also highlighted the importance of accurate distance and environmental conditions.

In the second case, a password lock system provided basic access control, which is useful for learning how digital security works at a simple level. It can be further developed into more secure systems.

The third case demonstrated how sensors and servo motors work together to create a basic scanning mechanism. This can be applied in robotics or obstacle detection projects.

The fourth case presented a voice control system that can be used in interactive applications or smart home automation.

Overall, the study demonstrates that even small, low-cost systems can provide useful functionality. These systems can be expanded and adapted for educational, personal, or even professional uses. It also encourages thinking about how simple technologies can be improved and applied in more advanced ways.

6.3 Limitations of the study

While the project achieved its main goals, there were some limitations that affected the performance and scalability of the systems.

In CASE 1, the home security system relied on a fixed detection range. It sometimes gave false readings due to surface reflections or environmental noise, especially in open spaces.

For CASE 2, the password system lacked encryption or secure storage. The password was stored directly in the code, which is not suitable for real-world applications that require higher security standards.

CASE 3 had limitations in accuracy and scanning speed. The radar system worked well for short distances, but detection became unreliable beyond 100–150 cm, especially with small or soft objects.

In CASE 4, the sound-activated LED was too sensitive in noisy environments. Background sounds could easily trigger the system unintentionally, and the sensor did not have filtering or advanced sound recognition.

Another general limitation across all cases was the use of basic hardware. The systems were built using low-cost components, which limited precision, performance, and advanced functionality.

6.2 Future Work

There are many ways these systems can be improved and developed further in the future

CASE 1 Home Security Alarm System

Add wireless communication (like GSM or Wi-Fi) to send alerts to a mobile phone. Use multiple sensors to cover larger areas and reduce blind spots.

Include a camera module for real-time monitoring.

CASE 2 Password Security Lock System

Store passwords securely using EEPROM or add encryption.

Introduce a display to show messages or feedback to the user.

Add biometric authentication (like fingerprint or RFID) for stronger security.

CASE 3 Radar System Using Ultrasonic Sensor

Replace the ultrasonic sensor with LiDAR or infrared for more precise readings.

Display the radar output on a graphical interface, like an OLED screen or computer dashboard. Improve scanning speed and smoothness with better servo control. CASE 4 Sound-Activated LED System

Add a digital sound sensor or microphone with filtering features.

Set up voice recognition for more advanced control.

Allow sensitivity adjustments through a user interface or mobile app.

By working on these improvements, each system could be made more reliable, user-friendly, and closer to real-world applications.

References

[1] International Journal of Advance Research, Ideas and Innovations in Technology

IoT based home security system using Arduino

2023

[2] Golder, A., Gupta, D., Roy, S., Al Ahasan, M. A., & Haque, M. A. GSM Based Home Security Alarm System Using Arduino Using Mobile Call. In 2023 IEEE 14th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON) (pp. 0268-0274). IEEE. (2023, October)

[3] Anand, Anisha & Sharma, Vatsala. IOT Based Home Security Smart System Using Arduino. (2024).

[4] Bhasker, M. SMART HOME SECURITY SYSTEM USING ARDUINO AND IoT. International Journal of Creative Research Thoughts, 9(8), c39–c42. (2021)

[5] Oyekola, P., Oyewo, T., Oyekola, A., & Mohamed, A. Arduino based smart home security system. Int. J. Innov. Technol. Explor. Eng, 8(12), 2880-2884. . (2019).

[6] Abiodun, N. O. J., & Okpe, N. O. A. Smart Home Security using Arduino-based (IoT) Intrusion Detection System. World Journal of Advanced Research and Reviews, 22(3), 857–864. (2024).

[7] Anand, A., Sharma, V., Government Engineering College Buxar, Singh, S., Bharti, R., Mishra, A., & Kumar, A. IOT based Home Security smart system using Arduino. In International Journal of Advance Research, Ideas and Innovations in Technology [Journal-article] , system. In the late 1900s and early 2000. (2023).

[8] A. Golder, D. Gupta, S. Roy, Md. A. A. Ahasan, and M. A. Haque, GSM

based home security alarm system using Arduino using mobile call. In 2023 IEEE 14th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), A customizable alarm system (2023).

[9] A. Anand, A. Kumar, and V. Sharma, "IOT Based Home Security Smart System Using Arduino," International Journal of Advance Research, Ideas and Innovations in Technology, Feb. (2024).

[10] M. Bhasker, "SMART HOME SECURITY SYSTEM USING ARDUINO AND IoT," International Journal of Creative Research Thoughts (IJCRT), vol. 9, no. 8, Art. no. IJCRT2108220, Aug. (2021).

[11] P. Oyekola, T. Oyewo, A. Oyekola, and A. Mohamed, "Arduino based smart home security system," International Journal of Innovative Technology and Exploring Engineering, vol. 8, no. 12, pp. 2880–2884, Oct. (2019).

[12] N. O. J. Abiodun and N. O. A. Okpe, "Smart Home Security using Arduino-based (IoTs) Intrusion Detection System," World Journal of Advanced Research and Reviews, vol. 22, no. 3, pp. 857–864, Jun. (2024).

[13] H. Kumar, R. Pratap Singh, N. Sharma, and P. Singh, "IOT BASED SMART SECURITY SYSTEM USING ARDUINO," JETIR, vol. 8, no. 8, Art. no. JETIR2108417,(2020).

[14] Sadie, "The history of home security," The NGCL, Nov. 23, (2023).

[15] T. S. Anjan Kumar, S. V. Kumar, "Simulation of IoT based totally clever domestic safety machine," international magazine of superior studies in computer technological know-how, vol. nine, pp. 43-forty eight, (2018).

[16] R. Sharma, P. ok. Gupta, "smart domestic security gadget the usage of IoT

and Arduino," worldwide magazine of computer applications, vol. 180, pp. 1-5, (2018).

[17] A. k. M. S. Hossain, "IoT based home safety machine using Arduino," https://www.researchgate.net/e-book/348937478_IoT_Based_Home_Security_System_Using_Arduino. ," (2023).

[18] J. A. k. A. M. A. Alomari, "Designing a clever home safety machine using IoT," (2021).

[19] J. A. Stankovic, "IoT: From Research and Innovation to Market Deployment," 1st edition, Springer, (2018).

[20] A. Bahga, V. Madisetti, "Internet of Things: A Developer's Guide," 1st edition, VPT, (2014).

[21] T. S. Anjan Kumar, S. V. Kumar, "Simulation of IoT Based Smart Home Security System," International Journal of Advanced Research in Computer Science, vol. 9, pp. 43-48,(2018).

[22] A. S. B. R. A. A. S. Al-Hallaj, "Smart Home Security System Using IoT and Arduino," International Journal of Engineering Research and Tec

[23]Arduino. (n.d.). Arduino Official Documentation. Retrieved from arduino.cc
Hossain, M. S., & Ahsan, M. A. "User-Centric IoT Security Systems: A Survey." Journal of Internet of Things Firebase. (n.d.). Firebase Documentation. Retrieved from firebase.google.com
AWS. (n.d.). Amazon Web Services Documentation. Retrieved from (2020).

[24] Blynk. (n.d.). Blynk Documentation. Retrieved from blynk.io
Khan, S., & Khan, F. (2021). "A Comprehensive Review on IoT-Based Home Security Systems." International Journal of Computer Applications, 2021.

[25] Arduino Official Documentation. (n.d.). *Arduino Reference*. Retrieved from

2023.

[26] Electronics Hub *Sound Activated LED Using Arduino*. Retrieved from: (2022).