ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25jun180

Microcontroller-Based Door Locking System

Kapil Kumar Bhati[1]; Mohit Kumar[2]; Pankaj Maurya[3]; Mohammad Shahzad[4]; Vishnu Kumar Singh[5]

[1, 2,3,4,5] Dept of Mechanical Engineering, IIMT College of Polytechnic, Greater Noida

Publication Date: 2025/07/14

Abstract: This paper presents the design and development of a low-cost, microcontroller-based door locking system utilizing an Arduino Uno, a keypad for password input, an LCD for user feedback, and a servo motor to simulate a locking mechanism. The project serves as a practical exploration of fundamental concepts in embedded systems and access control. The system's architecture, hardware integration, and software implementation for password entry, verification, and actuation are detailed. Functional testing demonstrates the successful operation of the core components, highlighting the potential for keyless entry and basic security enhancement. The paper also discusses the system's limitations, including security vulnerabilities and the non-robust locking mechanism, and proposes avenues for future work such as enhanced security protocols, multi-user support, and integration with smart home systems. This work contributes to the understanding of basic electronic access control principles and the capabilities of low-cost microcontroller platforms in security applications.

Keywords: *Microcontroller-Based System, Door Locking System, Access Control, Embedded Security, Arduino.*

How to Cite: Kapil Kumar Bhati; Mohit Kumar; Pankaj Maurya; Mohammad Shahzad; Vishnu Kumar Singh (2025) Microcontroller-Based Door Locking System. *International Journal of Innovative Science and Research Technology*, (RISEM–2025), 136-144. https://doi.org/10.38124/ijisrt/25jun180

I. INTRODUCTION

The increasing demand for secure and convenient access control solutions has driven significant advancements in electronic locking systems. Traditional mechanical locks, while ubiquitous, suffer from limitations such as the risk of key loss or duplication and a lack of flexibility in access management. Electronic door locking systems offer potential advantages in terms of security, convenience, and the ability to integrate with other smart technologies. Microcontroller-based platforms, like the Arduino, provide a cost-effective and versatile foundation for prototyping and developing such systems, making them accessible for educational purposes, DIY projects, and even basic commercial applications. This paper explores the design and implementation of a fundamental microcontroller-based door locking system, highlighting its architecture, functionality, and potential for future development.

This work details the development of a system employing a keypad for user input, an LCD for feedback, and a servo motor as a simulated lock, all controlled by an Arduino Uno. The project serves as a case study in applying embedded systems principles to access control, examining the interplay between hardware and software components. By outlining the design process, testing results, and inherent limitations, this paper aims to contribute to the understanding of the basic principles involved in electronic door locking mechanisms and to identify key areas for future research and improvement in low-cost access control solutions.

- *Background and Motivation*

The evolution of access control from traditional mechanical locks to sophisticated electronic systems has been driven by the need for enhanced security and convenience. Electronic door locking systems, leveraging technologies like PIN codes, RFID, and biometrics, offer advantages over mechanical counterparts, including flexible access management and integration potential. Microcontroller platforms, such as the Arduino, provide a low-cost and accessible means for developing these systems, making them relevant for educational purposes and basic practical applications. This paper explores the fundamental principles of such systems through the design and implementation of a basic Arduino-based door lock.

- *System Overview*

This paper details the development of a basic electronic door locking system centered around the Arduino Uno microcontroller. The system incorporates a 4x4 keypad for users to input a password, a 16x2 I2C LCD module to provide visual feedback during password entry and system status updates, and a small servo motor to simulate the mechanical locking and unlocking of a door. A passive buzzer is included to provide auditory feedback for successful or unsuccessful access attempts. The Arduino Uno acts as the central processing unit, reading input from the keypad, controlling the LCD display, actuating the servo motor, and generating sound through the buzzer based on the implemented software logic for password verification.

➢ *Scope and Objectives of the Research*

- To design and implement a basic microcontroller-based door locking system using an Arduino Uno.
- To develop a functional password-based access control mechanism utilizing a keypad and a simulated locking mechanism.
- To evaluate the system's core functionalities, including password entry, verification, and actuation.
- To identify the limitations of this basic implementation and propose potential avenues for future development in low-cost electronic access control.

II. LITERATURE REVIEW

➢ *Microcontroller-Based Access Control Systems*

The use of microcontrollers, particularly platforms like Arduino, has been widely investigated for developing access control systems due to their affordability and adaptability. Research has demonstrated the feasibility of employing Arduino for creating customizable security solutions suitable for various applications [1]. A significant area of exploration involves integrating Radio Frequency Identification (RFID) technology with Arduino to achieve contactless access control, enhancing both security and user convenience [2], [3].

These systems typically utilize RFID readers to identify users through unique tags, with the Arduino microcontroller verifying these credentials and subsequently controlling electronic locking mechanisms such as solenoid locks [3]. Furthermore, studies have shown the potential to extend the functionality of Arduino-based access control by incorporating web-based interfaces for user management and system monitoring, indicating a pathway towards more sophisticated and manageable security solutions [2].

➢ *Security Considerations in Embedded Locking Systems*

Ensuring the security of embedded door locking systems is a critical area of research. Password-based authentication, while common in low-cost systems, presents several security challenges. Studies have examined the vulnerability of these systems to brute-force attacks, where an attacker systematically tries all possible password combinations [4]. Furthermore, the risk of shoulder surfing, where unauthorized individuals observe the password being entered, is a significant concern [5]. To mitigate these risks, researchers have explored techniques such as implementing lockout periods after multiple failed attempts and using non-sequential or randomly generated default passwords [6].

Hardware security is another crucial aspect, with investigations into protecting the microcontroller and memory from physical tampering to prevent unauthorized access to stored passwords or system control [7]. Secure storage of passwords within the limited resources of embedded systems is also an active area of research, with hashing and salting algorithms being adapted for resource-constrained environments to enhance security against data breaches [8].

➢ *User Interface and Experience in Electronic Locks*

The design of the user interface in electronic door locks is paramount for both security and user acceptance. While keypad-based systems offer a straightforward input method, they can present usability challenges if not designed thoughtfully [9]. Effective Human-Computer Interaction (HCI) principles emphasize the need for clear feedback mechanisms, such as visual or auditory cues confirming key presses and the outcome of authentication attempts, to guide users and minimize errors [10].

Research suggests that providing temporary visual display of entered characters can improve accuracy compared to solely masked input [11]. Furthermore, a consistent and intuitive design language is crucial for enhancing memorability and reducing the cognitive load on users interacting with the system [12]. Ultimately, the challenge lies in striking a balance between implementing robust security measures and ensuring a user-friendly experience that doesn't lead to insecure workarounds [13].

III. COMPONENTS

➢ *Arduino Uno Microcontroller:*

At the heart of our electronic door locking system lies the Arduino Uno, a compact yet capable single-board computer. This board houses the ATmega328P microcontroller, which acts as the system's brain, executing the instructions we program into it. With a processing speed of 16 MHz, it's quick enough to handle the real-time tasks of reading user input, updating the display, and controlling the locking mechanism and buzzer. The Arduino Uno provides a set of digital input/output pins, some of which can generate Pulse Width Modulation signals essential for controlling the servo motor's precise movements. It also features analog input pins, though these aren't directly utilized in our current password-based lock.

The board is designed for ease of use, featuring a convenient USB connection that serves both for uploading our program code and for powering the device during development. It also has a separate power jack for situations where external power is needed. The Arduino's memory includes flash memory for storing the program, SRAM for temporary data during operation, and EEPROM for persistent storage if we needed to save information like the password even when the power is off (though our basic system might not use EEPROM). The combination of its processing power, versatile input/output capabilities, and the user-friendly Arduino development environment makes the Uno an excellent platform for creating and experimenting with embedded systems like our door lock.

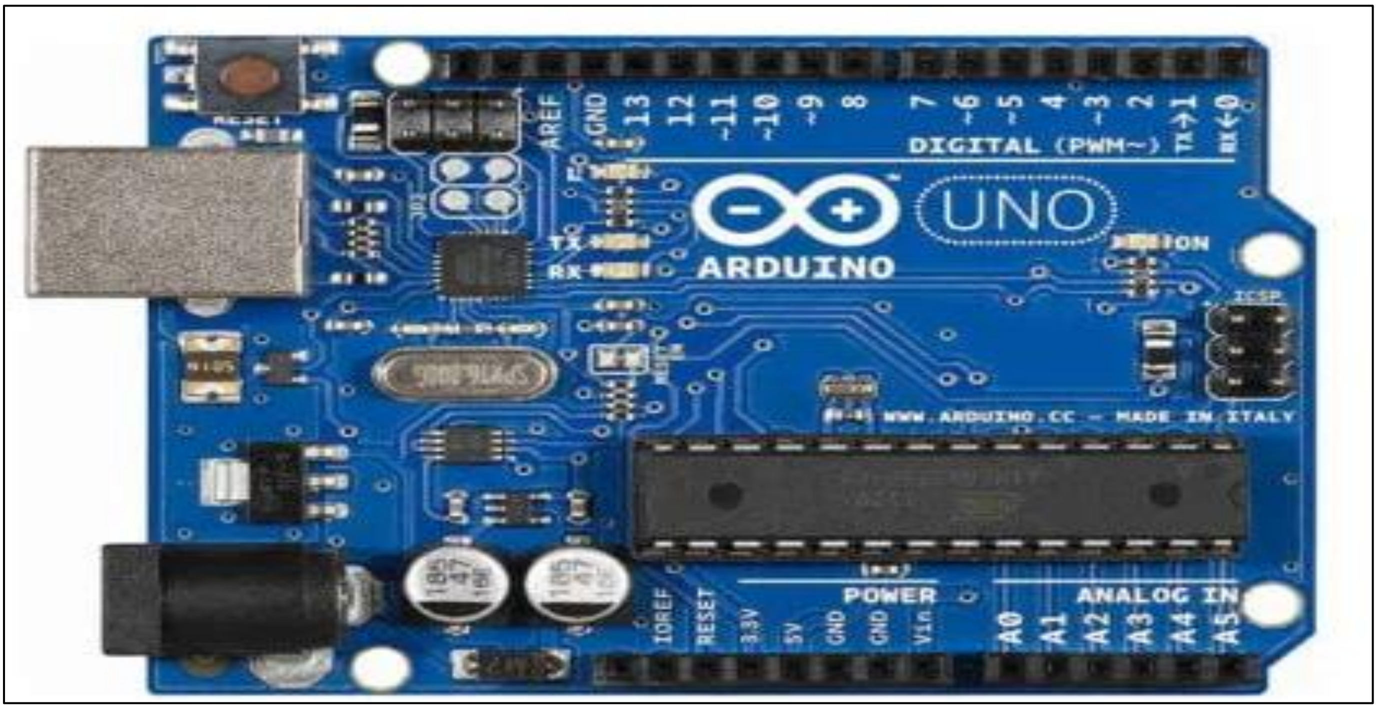

Fig 1 Arduino Uno Microcontroller

➢ *4x4 Membrane Keypad:*

The 4x4 keypad serves as the primary human interface for our door locking system, allowing users to input their unique access code. This input device is structured as a grid of sixteen individual push-button switches, arranged in four rows and four columns. Each button, when physically pressed, creates a temporary electrical connection between a specific row and a specific column within this matrix. The Arduino microcontroller continuously monitors the electrical states of these rows and columns. To determine which button has been pressed, the Arduino employs a technique called row-column scanning. This involves sequentially activating each row by setting it to a high or low logic level and then reading the status of the columns. If a button in that row is pressed, the signal will propagate to the corresponding column, allowing the Arduino to pinpoint the exact button that is being held down

.

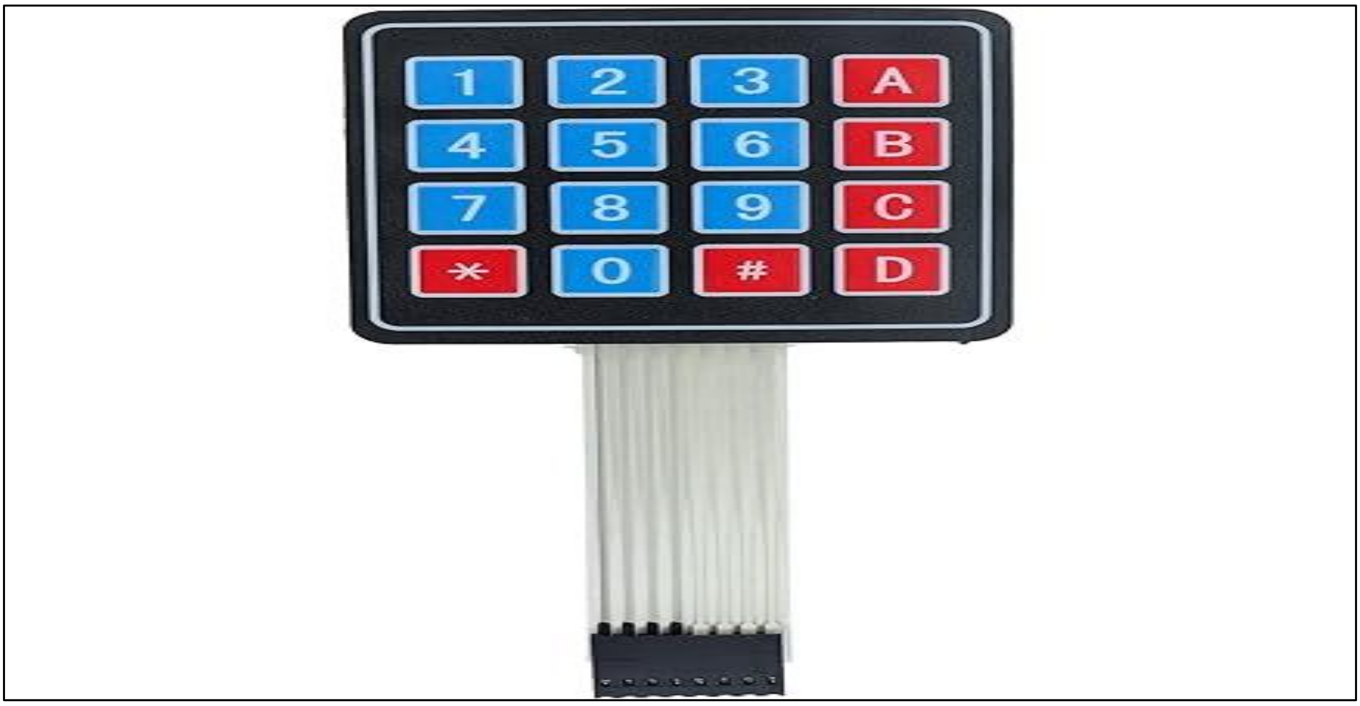

Fig 2 4*4 Membrane Keypad

This matrix arrangement is an efficient way to manage a relatively large number of buttons using a smaller number of microcontroller pins. Instead of needing a dedicated pin for each of the sixteen buttons, the 4x4 keypad typically requires only eight pins (four for the rows and four for the columns) to be connected to the Arduino. The Keypad.h library, commonly used in Arduino projects, provides pre-built functions that handle the complexities of scanning the keypad, debouncing the mechanical switches (to prevent multiple readings from a single press), and returning the character associated with the pressed key. In our door locking system, each button on the keypad will likely be assigned a specific digit (0-9) and potentially some control characters (like '#' for enter or '*' for clear), forming the vocabulary that users can use to enter their password. The accurate and reliable reading of these key presses by the Arduino is fundamental to the security and usability of our system, as it forms the basis for the password verification process.

- *16x2 I2c Lcd Module:*

The 16x2 Liquid Crystal Display (LCD) module serves as the primary visual output interface for our door locking system, providing essential real-time feedback to the user. This display has the capability to present sixteen characters across each of its two lines, allowing for clear and concise textual communication regarding the system's operation. In our project, the LCD is used to show the characters entered by the user via the keypad, often masked with asterisks to maintain basic password security during input. Furthermore, it plays a vital role in conveying the system's status, displaying messages such as a welcome screen upon startup, confirmation of a successful password entry, or an indication of an incorrect attempt, thus guiding the user through the interaction.

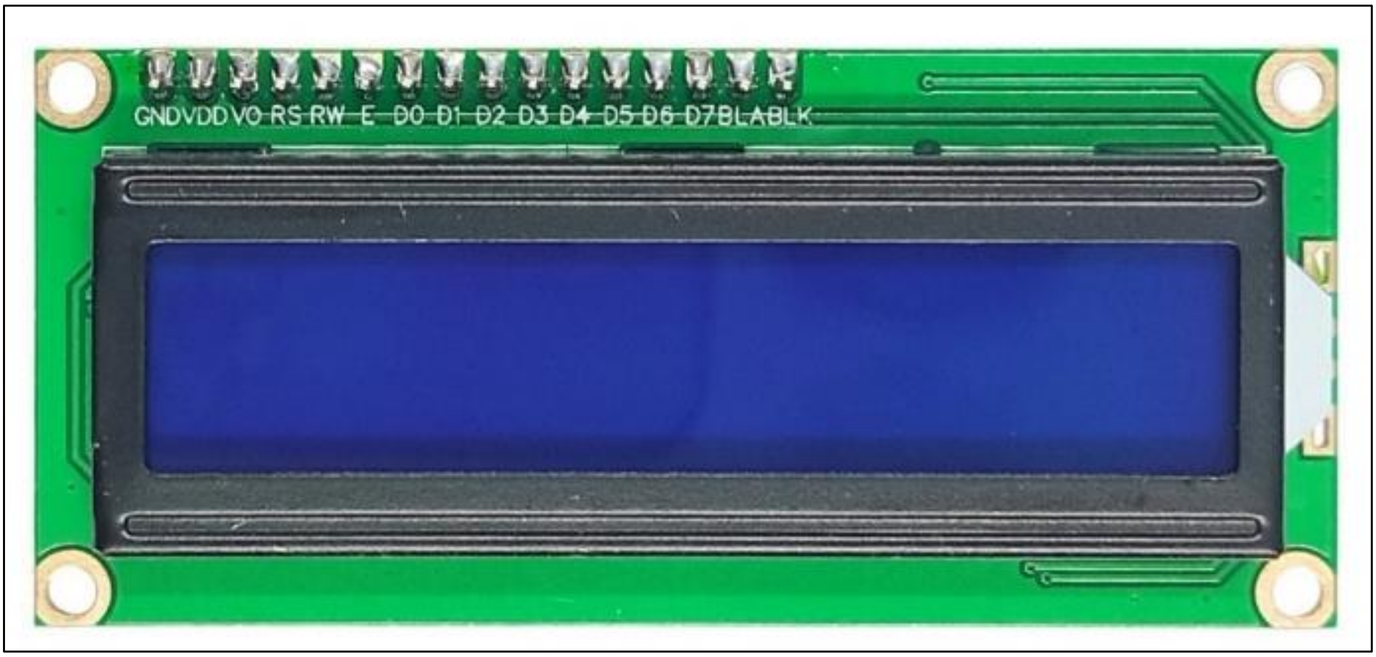

Fig 3 The 16x2 Liquid Crystal Display (LCD) Module Providing A Visual Interface

The integration of the I2C (Inter-Integrated Circuit) interface on this LCD module offers a significant advantage in terms of wiring simplicity. Unlike traditional LCDs that require numerous data and control connections to the microcontroller, the I2C interface communicates using only two data lines (SDA and SCL), along with power and ground. This streamlined connection not only reduces the complexity of the physical wiring but also conserves valuable digital input/output pins on the Arduino Uno, making it easier to incorporate other sensors or actuators in more complex versions of the system. Utilizing a library like LiquidCrystal_I2C.h within the Arduino programming environment simplifies the process of controlling the LCD, allowing us to easily display text and manage the cursor position on the screen, which is crucial for providing a user-friendly experience with our password-based door lock.

- *Servo Motor:*

In our door locking system, a small hobby servo motor acts as a simplified representation of a real door lock mechanism. Servo motors are a specific type of rotary actuator that allows for precise control over the angular position of their output shaft. Unlike standard DC motors that rotate continuously, a servo motor can be commanded to move to and hold a specific angle within its range of operation, typically between 0 and 180 degrees for standard models. This precise positioning is achieved through an internal feedback system that constantly monitors the motor's current angle and makes adjustments to match the desired commanded angle.

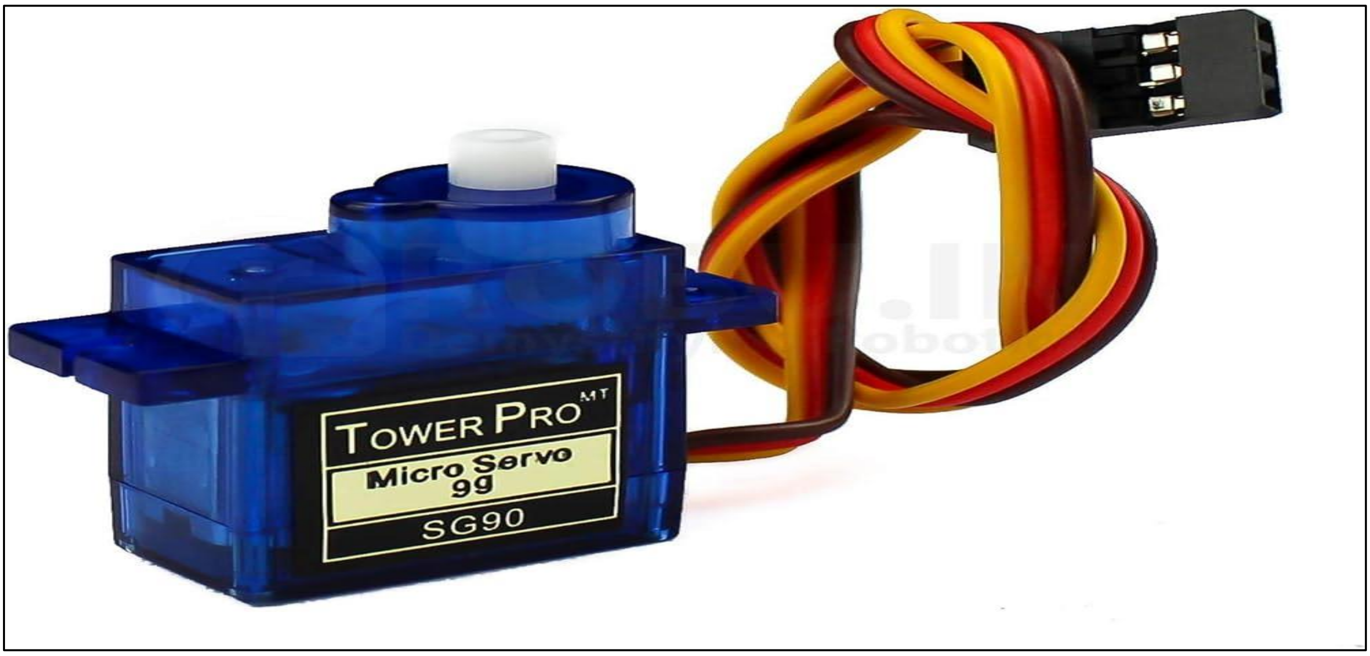

Fig 4 Servo Motor to Control the Angle of Door Locking System

The control of a servo motor is typically achieved using Pulse Width Modulation (PWM) signals sent from the microcontroller, in our case, the Arduino Uno. A PWM signal is a series of electrical pulses where the width of each pulse determines the commanded angle of the servo's shaft. A standard servo motor usually has three connection wires: one for power (typically +5V), one for ground, and one for the control signal that carries the PWM pulses. By varying the duration of the high pulse within a specific period (usually around 20 milliseconds), the Arduino can instruct the servo to rotate to a precise angular position. In our door lock simulation, we utilize this capability to define specific angles that correspond to the "locked" and "unlocked" states of the door. When the Arduino verifies a correct password, it sends a PWM signal that directs the servo to move to the "unlocked" angle. Conversely, for a locked state or an incorrect password, the signal commands the servo to remain or move to the "locked" angle. While providing a clear visual and mechanical analogy for a door lock's function, it's important to remember that a small hobby servo lacks the physical strength and security of a real-world locking mechanism.

- *Piezo Buzzer*

The buzzer in our door locking system serves as a crucial element for providing auditory feedback to the user, complementing the visual cues from the LCD. This small electromechanical component is capable of producing sound when an electrical signal is applied to it. In our specific implementation, we are using a passive buzzer. Unlike active buzzers, which contain their own internal oscillating circuit and generate a tone with a simple DC voltage, a passive buzzer requires an external alternating current (AC) signal to produce sound waves. This means that the Arduino microcontroller must actively generate the oscillating signal to make the buzzer emit a tone.

Fig 5 Piezo Buzzer Providing Auditory Feedback

The Arduino can easily achieve this by rapidly toggling a digital output pin connected to the buzzer between a high and a low logic level. The speed at which this toggling occurs determines the frequency of the sound produced, and thus the pitch we hear. By carefully controlling the frequency and the duration of these signals in our program, we can create distinct auditory patterns to signify different events within the door locking system.

For example, a short, high-pitched beep might indicate that a button on the keypad has been successfully pressed, providing immediate confirmation to the user. A longer, perhaps lower-pitched tone could signal that the entered password has been verified as correct and the simulated lock has been disengaged. Conversely, a series of rapid beeps or a distinct, unpleasant tone might be used to alert the user to an incorrect password entry. This use of varied auditory feedback enhances the user experience by providing an additional layer of information about the system's state and the outcome of their interactions, making the system more intuitive to use.

IV. WORKING PRINCIPLE

➢ *System Initialization and user Input*

Upon power-up, the Arduino Uno initializes the connected components, primarily the LCD and the keypad. The LCD typically displays an initial message, prompting the user for input. The system then enters a state where it continuously monitors the 4x4 keypad for any button presses. When a user presses a key, the Arduino detects this using row-column scanning and registers the corresponding character.

For each valid input, the Arduino updates the LCD to show a masked character (like an asterisk), providing visual confirmation of input without revealing the actual password. Optionally, a short auditory beep from the buzzer may accompany each key press. The entered characters are stored sequentially in a temporary memory buffer within the Arduino.

➢ *Password Verification*

Once the user signals the completion of their password entry (either by pressing a designated "enter" key or after a predefined number of characters), the Arduino initiates the password verification process.

The sequence of characters stored in the input buffer is compared, character by character, against a pre-stored correct password that resides within the Arduino's program memory. This comparison is case-sensitive if implemented as such in the software. The outcome of this comparison determines the subsequent actions of the system.

➢ *Output and Feedback Mechanisms*

Based on the result of the password verification, the system provides feedback to the user through the output components:

- *Successful Authentication (Password Match):*

If the entered password matches the stored password, the Arduino sends a control signal (PWM) to the servo motor. This signal instructs the servo to rotate to a specific angle representing the "unlocked" state. Simultaneously, the buzzer emits a distinct auditory signal (e.g., a long beep), and the LCD displays a positive confirmation message such as "Password Accepted" or "Unlocked."

- *Failed Authentication (Password Mismatch):*

If the entered password does not match the stored password, the Arduino does not actuate the servo to the "unlocked" position. Instead, the buzzer produces a different auditory signal (e.g., a series of short beeps or a low-pitched tone), and the LCD displays an error message like "Wrong Password" or "Access Denied." Following a failed attempt, the system may reset, prompting for a new password entry, or it might implement a security lockout for a certain period after multiple incorrect attempts.

➢ *Circuit Diagram and Component Interconnection*

The functional operation of the microcontroller-based door locking system is directly enabled by the specific electrical connections between its constituent components, as illustrated in Figure 1. The 4x4 keypad's row and column pins are wired to designated digital input pins on the Arduino Uno, allowing the microcontroller to scan for button presses and identify user input. The 16x2 I2C LCD module is connected to the Arduino via the I2C interface, utilizing the Arduino's analog pins A4 (SDA) and A5 (SCL) for serial data communication, which simplifies the wiring required for displaying system feedback.

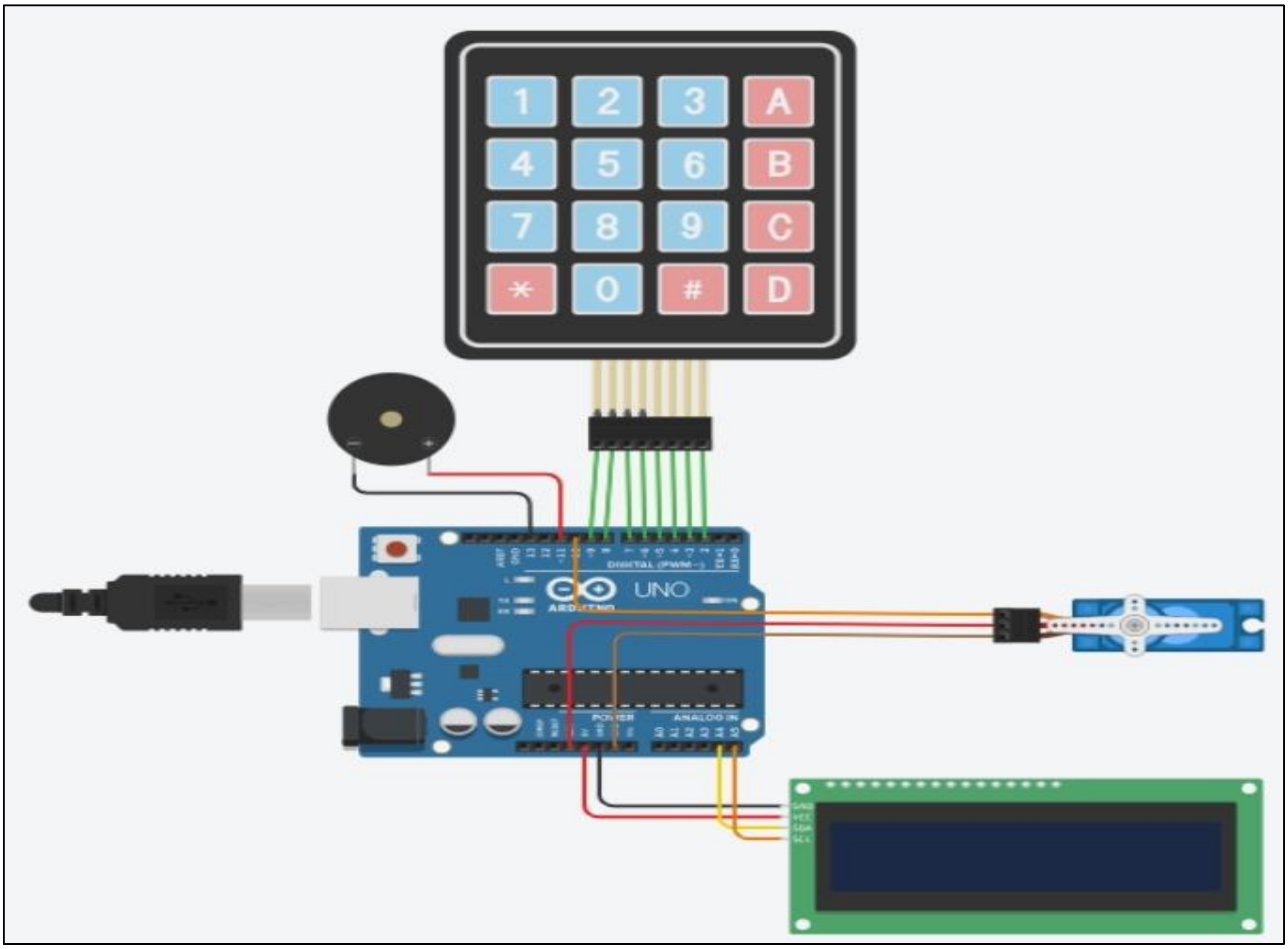

Fig 6 Circuit Diagram of System

The servo motor, responsible for simulating the door lock, is connected with its control signal wire to a specific digital pin on the Arduino that supports Pulse Width Modulation (PWM) output, enabling precise angular control. Finally, the passive buzzer is connected to another digital output pin on the Arduino, which the microcontroller can rapidly toggle to generate the necessary oscillating signal for producing sound. Power and ground connections are established from the Arduino's 5V and GND pins to each of the peripheral components, ensuring they receive the necessary electrical supply to operate. These interconnections form the physical pathways through which signals and power are exchanged, allowing the Arduino to sense user input and control the output devices according to the programmed logic that defines the system's working principle.

- ➢ *System Control Flow*

The entire operation of the door locking system is governed by the program logic implemented on the Arduino Uno. The microcontroller continuously cycles through the process of reading keypad input, updating the LCD, performing password verification when a complete entry is detected, and controlling the servo motor and buzzer based on the verification outcome. The software defines the correct password, the number of allowed attempts (if any), the duration of lockout periods (if implemented), and the specific visual and auditory feedback provided to the user at each stage of interaction.

V. MERITS

The "Microcontroller-Based Door Locking System" offers several key benefits, particularly in the context of learning, basic security awareness, and cost-effectiveness:

- ➢ *Demonstration of Fundamental Concepts:*

The project serves as a clear and tangible illustration of core principles in embedded systems, digital logic, and basic security implementation. It allows learners to understand the interaction between hardware components (microcontroller, input/output devices) and software programming for access control.

- ➢ *Cost-Effectiveness:*

Utilizing a low-cost microcontroller platform like the Arduino Uno and readily available peripherals (keypad, LCD, servo, buzzer) makes the system relatively inexpensive to build. This affordability makes it accessible for educational purposes, hobbyist projects, and basic DIY security applications.

- *Customizability and Expandability:*

The open-source nature of the Arduino platform allows for easy modification and expansion of the system's functionality. The password can be readily changed in the code, and additional features like logging attempts or different unlocking mechanisms could be explored.

- *Enhanced Convenience (Basic):*

Even in its basic form, the system offers a level of convenience over traditional key-based locks by providing keyless entry. Users do not need to carry physical keys that can be lost or duplicated.

- *Foundation for Security Awareness*

Building and understanding this system can raise awareness about the fundamental principles and potential vulnerabilities of electronic access control systems. It provides a hands-on introduction to security considerations that are crucial in more complex systems.

VI. DEMERITS

The "Microcontroller-Based Door Locking System," in its current basic implementation, also presents several limitations and potential drawbacks:

- *Limited Security:*

The security of the system is rudimentary. The password is typically stored in the microcontroller's code, making it potentially vulnerable to reverse engineering or unauthorized access if the hardware is compromised. It lacks advanced security features like encryption or protection against sophisticated attacks.

- *Single User Limitation:*

The current design typically supports only a single, pre-programmed password. It does not inherently handle multiple users with different access privileges or the ability to easily change or manage user access.

- *Simulated Locking Mechanism:*

The use of a hobby servo motor to simulate a door lock provides a good demonstration of the concept but lacks the physical strength and security of a real-world locking mechanism like a solenoid bolt or a motorized deadbolt. It would be easily bypassed in a real security scenario.

- *Vulnerability to Brute-Force Attacks:*

Without implemented countermeasures, the system could be susceptible to brute-force attacks, where an attacker tries numerous password combinations in an attempt to gain unauthorized access.

- *Lack of Audit Trail:*

The basic system does not typically include any mechanism for logging access attempts, successful or failed. This lack of an audit trail can be a significant limitation in security-sensitive applications.

- *Potential for Power Dependency:*

As an electronic system, it is dependent on a reliable power source. In the event of a power outage, the locking mechanism might become inoperable unless a backup power supply is implemented.

VII. CONCLUSION

This paper presented the design and implementation of a basic microcontroller-based door locking system utilizing an Arduino Uno, a keypad for input, an LCD for feedback, and a servo motor to simulate a locking mechanism. The project served as a practical exploration of fundamental principles in embedded systems and access control. The system successfully demonstrated the core functionalities of password-based entry, verification, and simulated actuation, highlighting the potential of low-cost microcontroller platforms for creating functional prototypes of electronic security systems.

The merits of this approach include its value as an educational tool for understanding embedded security concepts, its cost-effectiveness due to the use of readily available components, and its potential for customization and expansion. However, the current implementation also exhibits limitations, notably in its basic security features, single-user support, and the use of a simulated locking mechanism. These demerits underscore the need for further development to enhance the system's robustness and suitability for real-world security applications.

Future work could focus on addressing the identified vulnerabilities through more sophisticated security protocols, implementing multi-user management, integrating more secure locking hardware, and exploring features such as access logging and tamper detection. This foundational project provides a valuable stepping stone for further investigation into the design and implementation of more advanced and secure microcontroller-based access control solutions.

REFERENCES

[1]. S. Ahmed and M. A. Khan, "Design and Implementation of a Low-Cost Password-Based Door Locking System Using Arduino," *Int. J. of Electrical and Computer Engineering (IJECE)*, vol. 8, no. 3, pp. 1546-1553, 2018.

[2]. R. Sureshkumar, S. Gokulakrishnan, and R. Senthilkumar, "Arduino Based Smart Door Locking System," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 5, no. 3, pp. 488-493, 2017.

[3]. N. Kumar, A. Goyal, and S. Sharma, "Microcontroller Based Door Locking System with Password Protection," *International Journal of Engineering Research and Technology (IJERT)*, vol. 4, no. 03, 2015.

[4]. S. A. Abbasi, M. A. Shaikh, and F. A. Memon, "Development of a Secure Door Locking System Using Microcontroller," *Engineering, Technology & Applied Science Research*, vol. 8, no. 1, pp. 2645-2649, 2018.

[5]. M. S. Islam, M. R. Amin, and M. U. Ahmed, "Design and Implementation of a Secure Password Based Door Lock System Using PIC Microcontroller," *International Journal of Scientific & Engineering Research*, vol. 4, no. 6, 2013.

[6]. A. Yesodha and S. V. Anu Bharathy, "Arduino Based Door Locking System with Mobile Notification," *International Journal of Engineering and Advanced Technology (IJEAT)*, vol. 8, no. 6S, pp. 31-35, 2019.

[7]. M. Sridevi, B. Janani, and R. Priya, "Smart Door Locking System Using Arduino and Bluetooth," *International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET)*, vol. 7, no. 3, 2018.

[8]. V. B. Bhandari and R. B. Keskar, "Design and Implementation of RFID Based Door Locking System using Arduino," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 5, no. 3, pp. 21-26, 2016.

[9]. T. O. Olugboji, O. S. Abolarin, and O. A. Awodele, "Development of a Biometric Door Locking System Based on Fingerprint Recognition and Microcontroller," *International Journal of Computer Applications*, vol. 179, no. 43, pp. 25-30, 2018.

[10]. R. S. Raut and S. S. Mali, "GSM Based Door Locking System," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 4, no. 5, pp. 28-32, 2014.

[11]. M. S. Kurniawan, A. Harjono, and A. Kurniawan, "Security System of Door Lock Using Face Recognition Based on Raspberry Pi," *Journal of Physics: Conference Series*, vol. 1198, no. 6, 2019.

[12]. S. A. Khan, A. A. Laghari, and N. U. Rehman, "Secure Door Locking System Using IoT and Raspberry Pi," *Mehran University Research Journal of Engineering & Technology*, vol. 38, no. 4, pp. 987-994, 2019.

[13]. A. Al-Ali, R. Zualkernan, and S. Gupta, "Smart Home Automation System Based on Bluetooth Low Energy," *2017 IEEE International Conference on Smart Computing and Communications (SmartCom)*, pp. 429-432, 2017.

[14]. D. P. Agrawal and Q.-A. Zeng, *Introduction to Wireless and Mobile Systems*, 3rd ed. Cengage Learning, 2010.