Smart Door Lock System Using Arduino Uno

Ruth Arlene T. Necio., Chenaniah Keziah M. Pastrana., Renz Jio Lazaro., Dr. Ma. Magdalena V. Gatdula

Bulacan State University – Graduate School City of Malolos, Bulacan

DOI: https://dx.doi.org/10.47772/IJRISS.2025.91100467

Received: 28 November 2025; Accepted: 05 December 2025; Published: 18 December 2025

ABSTRACT

The demand for affordable and customizable security systems continues to grow as embedded microcontrollers become more accessible. This study explores the development of a Smart Door Lock System using Arduino Uno, implemented entirely in Tinkercad's block-based programming environment. The system integrates a potentiometer acting as an analog password input, a 16×2 LCD for real-time feedback, an LED indicator for status signaling, and a servo motor as the primary locking mechanism. Results from three test scenarios—valid input, invalid input, and user inactivity—demonstrated consistent and accurate system performance. Findings highlight the educational value of analog-to-digital processing and conditional logic for early embedded systems learning, while identifying limitations and opportunities for future enhancement.

Keywords: Arduino Uno; Smart Door Lock System; Tinkercad Simulation; Analog Password Input; Embedded Systems; Conditional Logic002E

INTRODUCTION

The integration of automation and security in households and small businesses has increased the relevance of smart lock systems. Microcontroller-based solutions, particularly with Arduino Uno, offer low-cost and flexible platforms for developing personalized implementations. Simulation tools such as Autodesk Tinkercad enable students to experiment with circuits and logic flows without requiring physical hardware. This study focuses on the design and simulation of a Smart Door Lock System utilizing an unconventional input device—a potentiometer serving as an analog password source. Despite its impracticality for real-world systems, the potentiometer provides an excellent foundation for demonstrating analog signal processing, threshold-based authentication, and actuator control in educational contexts. The purpose of this work is to document the system's structure, behavior, and performance, and to evaluate its suitability as a teaching tool for embedded systems principles.

METHODS

The system was composed of an Arduino Uno microcontroller, a 10kΩ potentiometer, a 16×2 LCD display, a standard LED, and an SG90 servo motor. Using Tinkercad's block-based coding interface, the potentiometer's analog values (0–1023) were scaled to a simplified range of 0–255. A password interval (180–190) was defined as the "correct" range. When the user adjusted the potentiometer to produce a value within this range, the LCD displayed "Access Granted," the LED illuminated, and the servo rotated to 0°, unlocking the simulated door. Incorrect values triggered "Access Denied," with the servo remaining locked at 180°. An inactivity timer displayed an idle message after 10 seconds without adjustment. Three test scenarios—correct input, incorrect input, and inactivity—were conducted to evaluate reliability, stability, and logical accuracy.

The Smart Door Lock System was created using Tinkercad's virtual electronics environment. The following components were used:

- **Arduino Uno** – Main controller that processes all inputs and outputs.
- **Potentiometer** – Used as the password input by turning the knob to change the value.
- **16×2 LCD Display** – Shows system messages such as "Access Granted" or "Access Denied."

- **LED Indicator** – Lights up when the correct password value is entered.
- **Servo Motor** – Acts as the lock, rotating to locked or unlocked positions.

All parts were connected according to standard Arduino wiring: the potentiometer to A0, the LCD through I2C, the LED to a digital pin with a resistor, and the servo to digital pin 9.

Figure 1. Tinkercad simulation circuit of the Smart Door Lock System.

System Logic and Programming

The program was created using Tinkercad's block-based coding. The potentiometer provides a value between 0–1023, which was scaled to 0–255. A password range of **180–190** was set as the correct input.

The system follows this simple logic:

- **Read the potentiometer value**
- **Check if the value is within the correct range**

If correct:

Display "Access Granted"

Turn on the LED

Rotate the servo to 0° (unlock)

If incorrect:

Display "Access Denied"

Keep LED off

Keep servo at 180° (locked)

If no change for 10 seconds:

Show an idle message like "Adjust Range"

Testing Process

Three main test scenarios were done:

Test 1: Correct Input

The potentiometer was adjusted to produce values between 180–190.

Expected behavior: unlock, LED ON, correct LCD message.

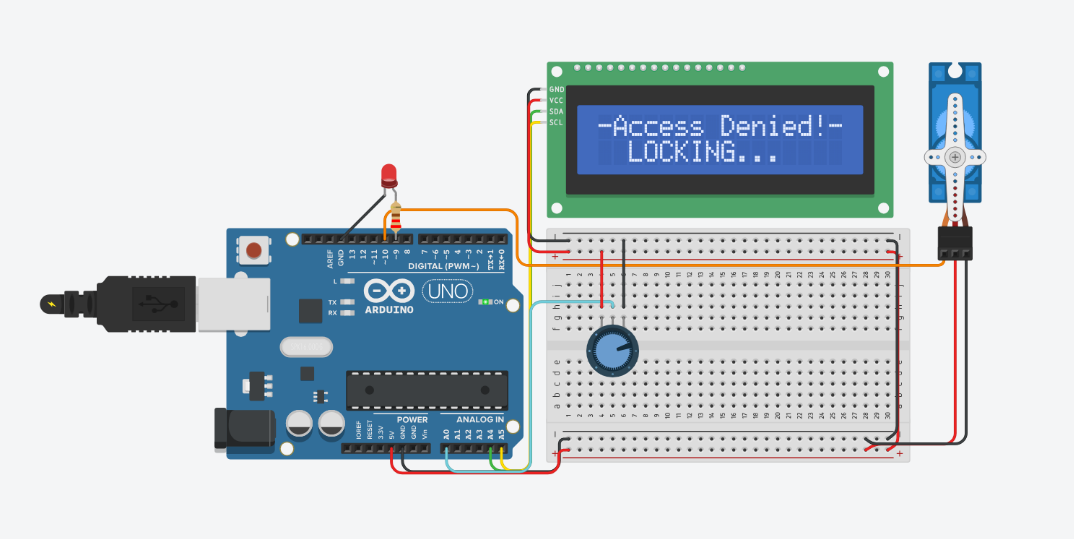

Figure 3. Testing 1

Test 2: Incorrect Input

Values outside the correct range were tested.

Expected behavior: access denied, servo stays locked.

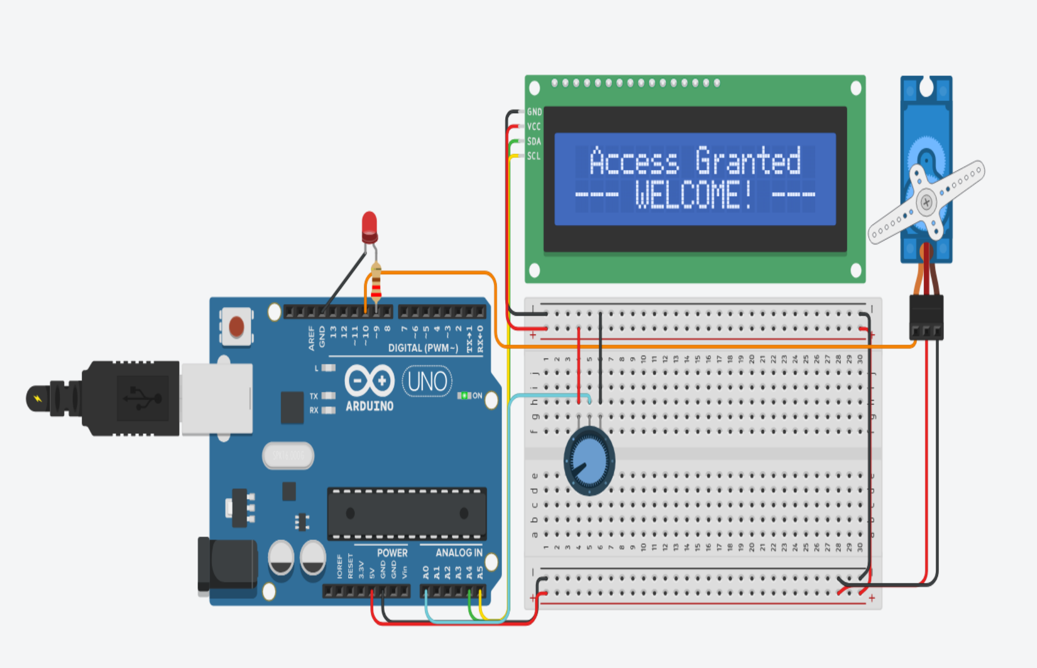

Figure 4. Testing 2

Test 3: No Input / Inactivity

The system was left untouched for 10 seconds.

Expected behavior: idle message and no movement.

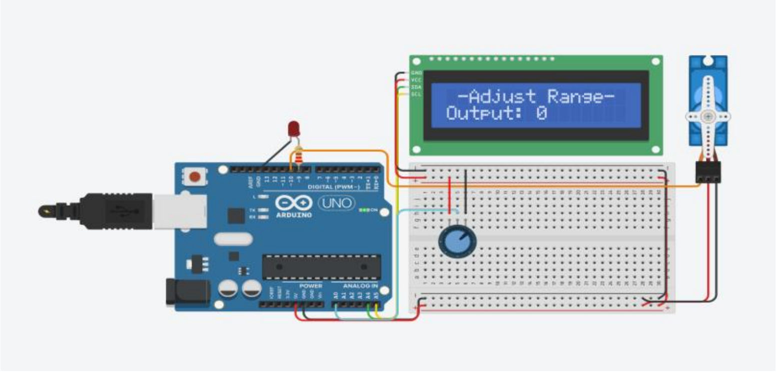

Figure 5. Testing 3

RESULTS

The system's performance across the three test scenarios showed consistent and accurate responses. During valid input tests, the LCD displayed "Access Granted," the LED activated immediately, and the servo motor reliably rotated to 0°, simulating an unlocked position. For invalid input ranges, the LCD correctly displayed "Access Denied," with no LED activation and the servo remaining locked at 180°. Inactivity tests showed stable idle behavior, with the LCD displaying a prompt for user interaction while maintaining hardware states unchanged. Across multiple trials, the system demonstrated stable analog reading behavior, minimal latency, and accurate conditional execution. The servo motor responded smoothly without jitter, and the LCD provided readable, real-time feedback.

DISCUSSION

This study demonstrates that a potentiometer, despite its unconventional role as an authentication device, can effectively illustrate password verification through analog value ranges. The system's robust performance in Tinkercad shows its potential as a teaching tool for sensor integration, actuator control, and decision-making logic in microcontroller environments. The combination of visual feedback (LCD), mechanical action (servo), and input variability (potentiometer) created an intuitive simulation of access control behavior. However, analog input systems are prone to noise and lack the precision needed for secure authentication. As such, future improvements could incorporate digital keypads, RFID systems, biometric sensors, Bluetooth authentication, or IoT connectivity for remote monitoring. Transitioning from block-based to text-based Arduino code would provide more flexibility and professional relevance, allowing for more complex logic structures and better scalability.

CONCLUSION

The Smart Door Lock System simulation successfully demonstrated fundamental principles of embedded systems design, including analog input processing, conditional logic, and actuator manipulation. The system reliably distinguished valid input ranges from invalid ones and responded appropriately across all test scenarios. Its stability during periods of inactivity further demonstrated the practicality of its logic structure. This project highlights how accessible simulation platforms like Tinkercad can support meaningful learning experiences in microcontroller programming and prototyping. With further refinement, the system can serve as a foundation for more advanced security-oriented applications in real hardware environment.

REFERENCES

1. Arduino. (2023). Arduino Uno technical specifications.
2. Autodesk. (2023). Introduction to Tinkercad circuits. Autodesk Education.
3. Mukherjee, S., & Roy, A. (2021). Microcontroller-based access control systems. International Journal of Embedded Systems, 14(3), 112–120.
4. Perez, L. (2020). Simulation-based learning in embedded system design. Journal of Engineering Education, 45(2),