Door Lock Security System Using Arduino

A report submitted to the

Department of Electrical and Information Engineering

Faculty of Engineering

University of Ruhuna Sri Lanka

On 19th March 2025

In completing a report for the module

EE6352 - Embedded Systems

By Group 33
Group members

Arachchi W.A.T.T.W.	EG/2021/4412
Arivanan V.	EG/2021/4414
Kahatagahawatta Y.P.U.	EG/2021/4592
ShageethPratheep V.	EG/2021/4809

Contents

List of Figures

List of Tables

1 Introduction

1.1 Introduction to the problem

In today's rapidly changing world, security and convenience are essential especially when it comes to controlling access to homes, offices, or restricted areas. Traditional key based door locks have been reliable for decades, but they present several challenges and inconveniences in real life situations.

Imagine a rainy day when the person in charge forgets the key, leaving staff waiting outside until the key is retrieved. Or at the end of a long, busy day, you stand in front of the door, struggling to find the correct key among a bunch of others. Worse yet, if the key is lost or accidentally left in the lock, it becomes an open invitation for intruders. These everyday scenarios highlight the inefficiency and vulnerability of traditional locks.

Modern door lock systems often incorporate biometric authentication, RFID cards, PIN codes, or smartphone apps to enhance security and improve convenience. Fingerprint based locks, in particular are widely adopted due to their balance of high security and user friendliness. They are commonly found in smart homes, offices, and secure labs.

Figure 1: Modern Fingerprint Lock Used in Sri Lanka
With Unlock Methods: Fingerprint, Password, Mifare Card, and Mechanical Key, Developed by Raniyo (Pvt) Ltd

In this project, we implement an existing solution to a common problem but with our own customized design, ensuring it remains simple, cost-effective, and adaptable for personal or small business use. With biometrics, each user's fingerprint acts as a unique "key," eliminating the need to carry or manage physical keys altogether.

The system is designed to store up to 127 unique fingerprints and provides three operational modes for flexibility and ease of management:

- Enrollment Mode ('e'): Allows new users to register a fingerprint with an assigned ID.
- Deletion Mode ('d'): Removes a registered fingerprint from the system based on the provided ID.

- Reset Mode ('r'): Returns the system to normal operation, ready to identify fingerprints.

If a valid fingerprint is detected, the system displays "System Ready" on an LCD screen, unlocks the door by triggering a relay, and provides positive feedback through a buzzer and LED. In case of an error, it displays "Sensor Error" and halts all operations to ensure security.

This solution improves security, saves time, and offers a modern alternative to physical key-based access. Whether in homes, offices, or restricted areas, this system ensures only authorized individuals gain entry without the problem of traditional locks.

1.2 Objective

The key objectives of this project are as follows.

- Design and build a fingerprint-based access control system using Arduino and a fingerprint sensor.
- Enhance security by replacing traditional key-based locks with fingerprint recognition.
- Improve user convenience by eliminating the need to carry, find, or manage physical keys.
- Implement a flexible management system with three modes:
    - Enroll Mode for adding new users.
    - Delete Mode for removing user fingerprints.
    - Reset Mode to return to standard operation.

- Provide clear user feedback through an LCD display, buzzer, and LED to indicate access granted, access denied, or errors.
- Ensure ease of use for non-technical users, allowing management of the fingerprint database through serial commands (e, d, r).
- Secure the door lock system using a relay module to control the locking mechanism, ensuring only authorized users can trigger the relay to unlock the door.
- Allow scalability by supporting up to 127 fingerprint IDs, making it suitable for both personal and small-scale commercial use.

1.3 Scope

While this project provides a smart and efficient solution, it has a few limitations to consider.

The system can store a maximum of 127 unique fingerprints, which may be insufficient for large organizations or multiuser environments.
Though fingerprints are unique and secure, the sensor's performance may be affected by wet, oily, or dirty fingers. There is a potential challenge in rough outdoor environments.
The system requires a continuous power supply. Power outages will disable the lock and making a backup power source necessary for real world use.

Switching modes relies on serial commands (`e`, `d`, `r`). While effective, a physical button interface might improve usability for the non-technical users.
The system controls one lock at a time. For multi lock setups, multiple units or a networked system would be required.

2 Specifications

2.1 Functional Specifications

2.1.1 Fingerprint Authentication System

The system must accurately capture, store, and recognize up to 127 unique fingerprints. It should authenticate users within 1 second to ensure quick access. If an unregistered fingerprint is detected, the system must deny access and trigger an error response. The fingerprint sensor must handle repeated scans without degradation in performance.

2.1.2 Multiple Operation Modes

The system must include three distinct operational modes.

In normal mode system continuously scans for fingerprints. Upon recognizing a valid fingerprint, it activates the door lock mechanism.
Enrollment Mode is initiated by a serial command ('e'), the system prompts the user to input a new ID (1-127). The user must scan their fingerprint twice for confirmation before saving the fingerprint data.
Deletion Mode is triggered by serial command ('d'), the system asks for an ID to delete from the fingerprint storage. The corresponding fingerprint entry is permanently removed.
In Reset Mode which is activated with command ('r'), resets the system to Normal Mode, displaying "System Ready."

2.1.3 Secure Access Control

On successful fingerprint recognition, the solenoid lock must unlock the door for 5 seconds, after which it automatically relocks. If the fingerprint is invalid, the system should display "Access Denied" and sound an error buzzer.

In case of sensor malfunction or wiring issues, the system must halt with "Sensor Error" displayed on the LCD.

2.2 Hardware Specifications

Table 1: Hardware Specifications of the Fingerprint Door Lock System

Component	Specification/Description
Arduino UNO	The central controller that manages fingerprint data, user input, and component behavior.
Fingerprint Sensor	Captures and verifies fingerprints, sending data to the Arduino for processing.
Solenoid Lock	Unlocks when access is granted, controlled by a relay.
Relay Module	Controls the solenoid lock using signals from the Arduino.
16x2 I2C LCD Display	Provides real-time feedback, displaying system status, enrollment, errors, and access results.
Buzzer	Emits distinct sounds for successful access, errors, or fingerprint failures.
LED Indicator	Lights up on successful access and blinks for error feedback.
12V Adapter	Powers the entire system, ensuring consistent and reliable performance.

3 Implemented Solution

3.1 Block Diagram

The block diagram shows the main components and their interactions within the system.

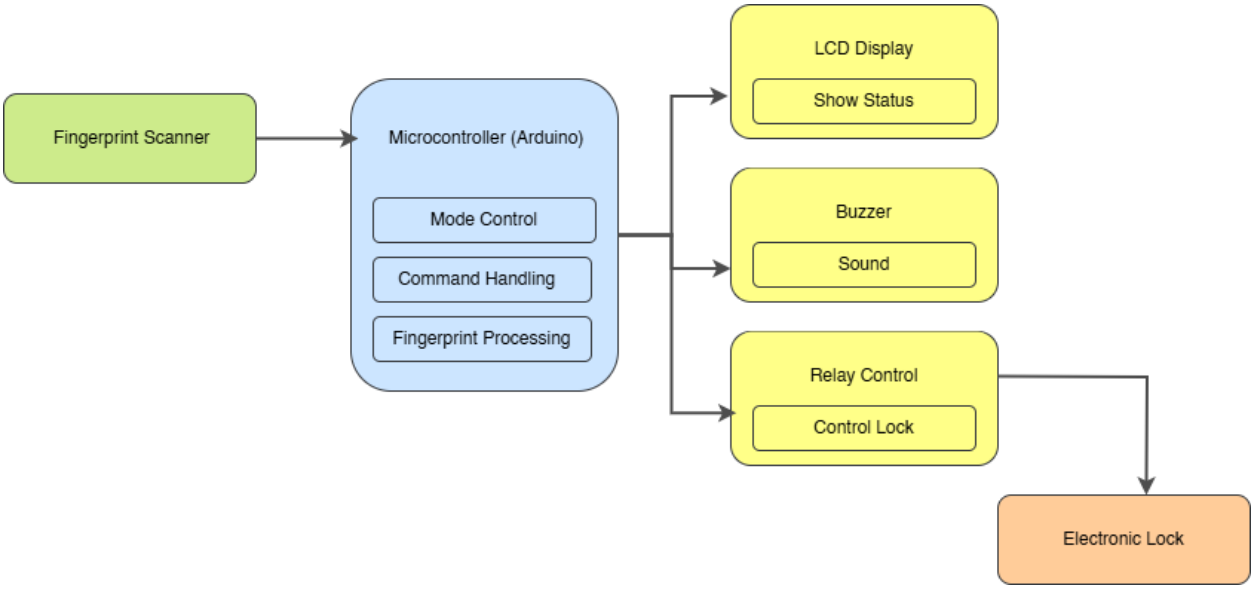

Figure 2: Block Diagram of the Fingerprint Door Lock System

The central processing unit is the Arduino UNO, which communicates with the fingerprint sensor to capture and process finger print data. The relay module is used to control the solenoid lock based on the authentication result. The LCD display provides a status message such as "Access Granted" or "Sensor Error," while the LED serves as a visual indicator for status of the system. The buzzer provides audible feedback for successful or unsuccessful operations. Each block represents a specific function that contributes to the overall system. It ensures seamless and secure access control.

3.2 Model Solution (Simulation)

The model solution for this project was simulated using Proteus software and Arduino IDE to validate the system's functionality before the implementation. The simulation focuses on the fingerprint recognition process. It ensures that valid fingerprints are correctly authenticated, and invalid ones are rejected. Additionally, the behavior of the solenoid lock, relay module, and feedback mechanisms (LCD, buzzer, LED) was tested under different conditions.

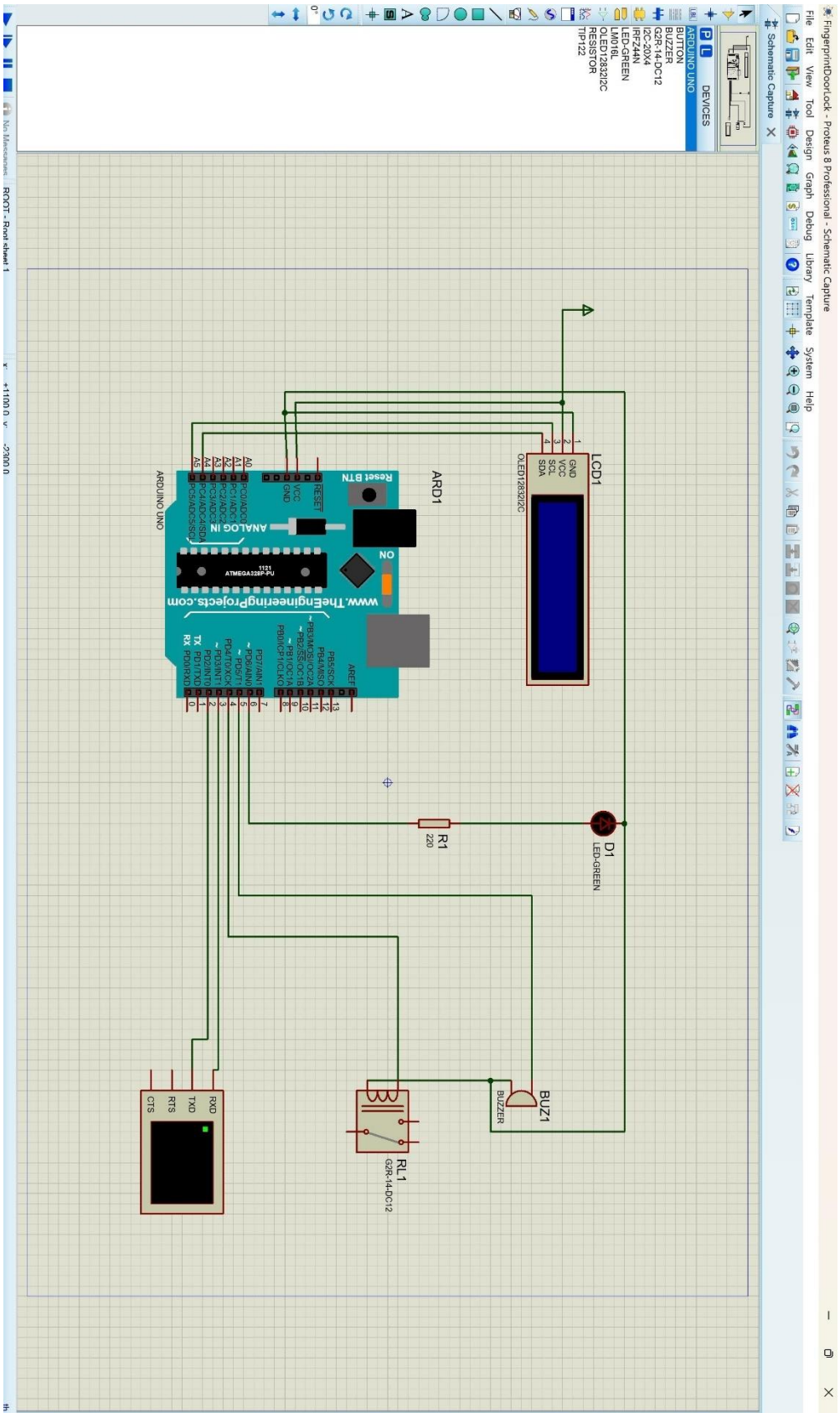

Figure 3: Model Simulation using Proteus

3.3 Flow Chart of the Firmware

The flowchart outlines the step-by-step process that the firmware follows to operate the fingerprint door lock system. It begins with initializing the system, including the fingerprint sensor and LCD display. In normal mode, the system waits for a fingerprint scan and compares it with stored fingerprint templates. If a match is found, the solenoid lock is activated, and access is granted. If no match is found, the system provides feedback throught the LCD and buzzer. In enrollment and deletion modes, users can add or remove fingerprints from the system, with clear instructions provided on the LCD.

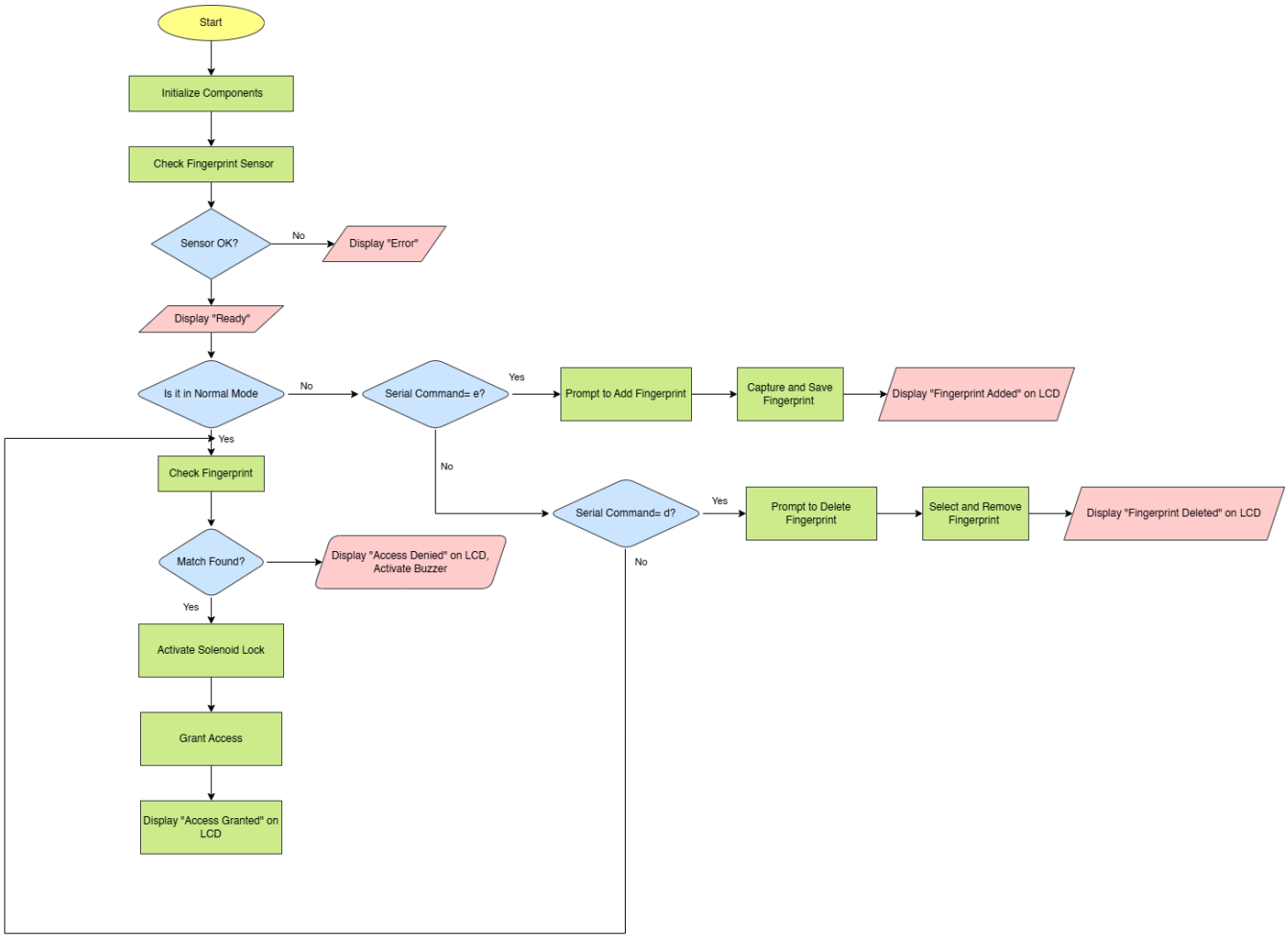

Figure 4: Flow Chart of the Fingerprint Door Lock System Firmware

3.4 Images of the final product

The final product consists of a compact and functional fingerprint door lock system. The hardware components are housed in a protective enclosure, with the fingerprint sensor placed on the front for easy user interaction. The LCD display shows real-time feedback, while the LED and buzzer provide additional status updates.

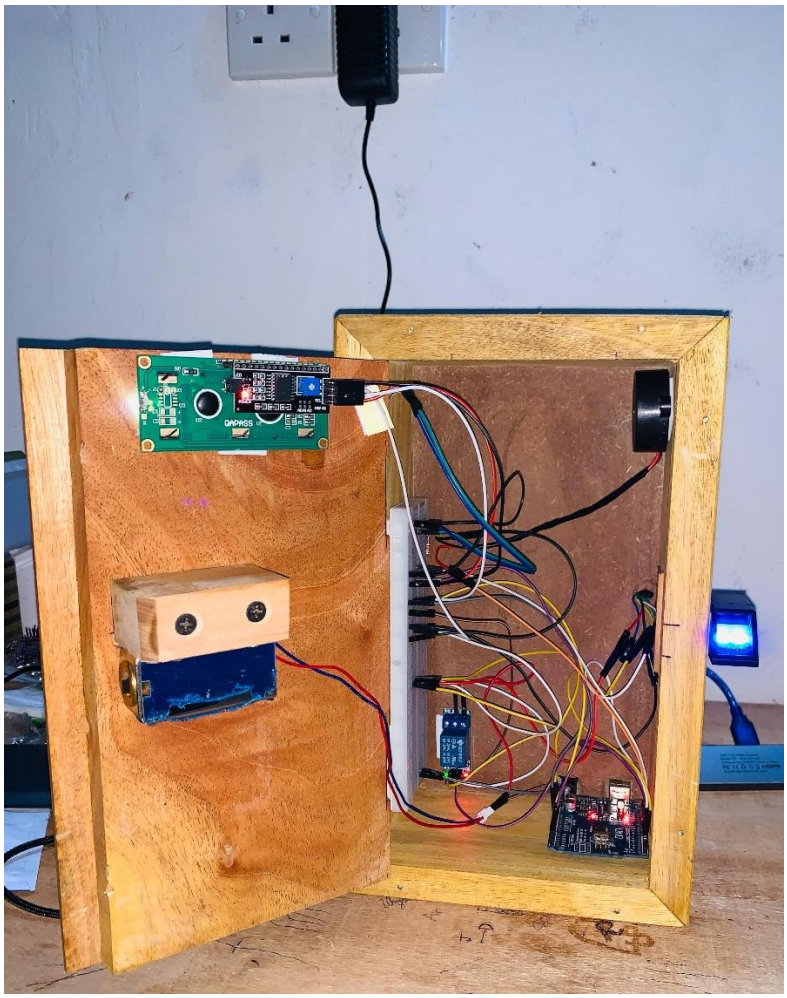

Figure 5: Complete Implementation of the Final Product

i) Normal Mode

Figure 6: Implementation of the system when it is in normal mode and starting the process.

Figure 7: Implementation of the system when Access Granted

Figure 8: Implementation of the system when Access Denied

ii) Enroll Mode

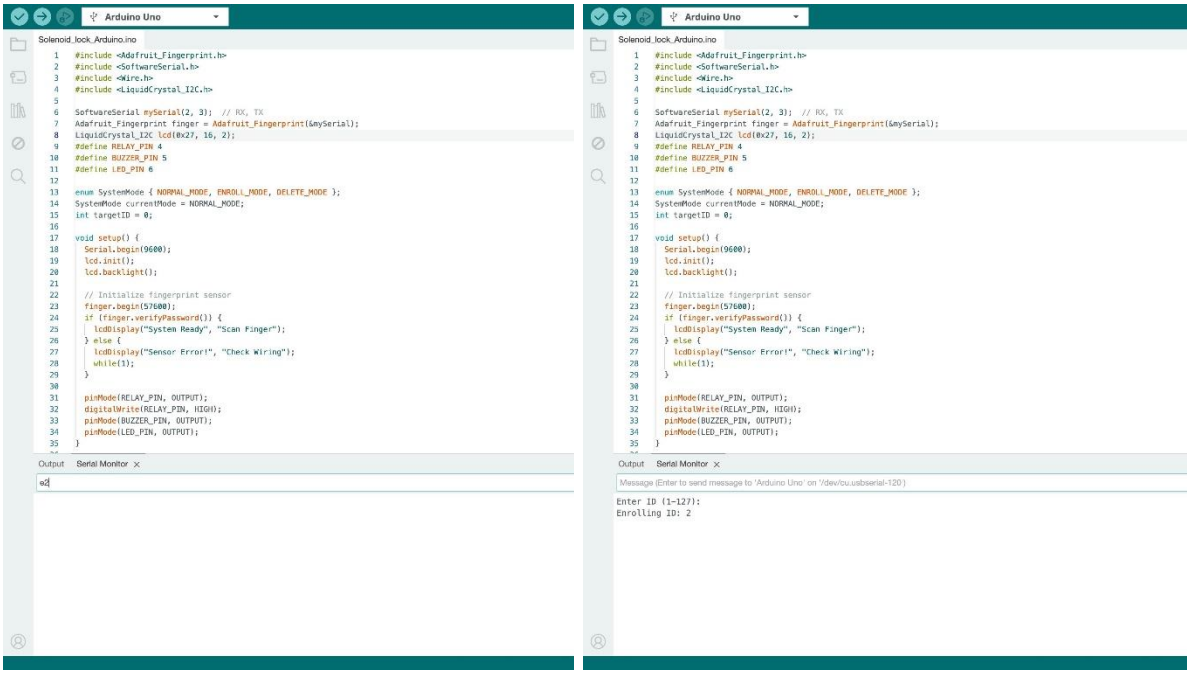

Figure 9: Enrolling a new ID using Arduino IDE

Figure 10: Implementation of the first step of the enrollment process

Figure 11: Displays 'Remove Finger' and waits for 2 seconds before the second scan.

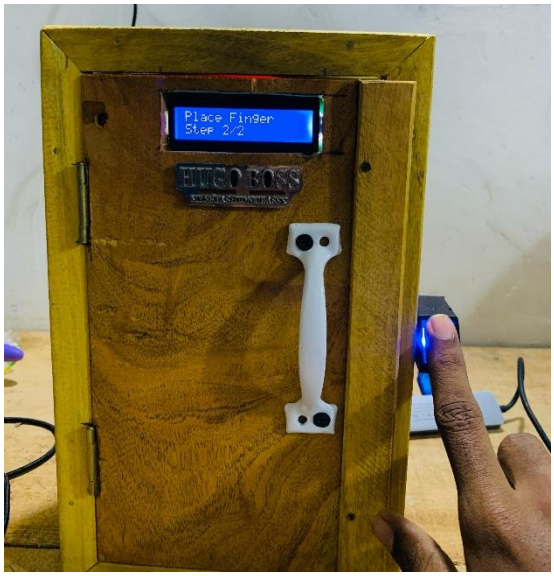

Figure 12: Displays the message "Place Finger" again, indicating that the user should place their finger for the second scan.

Figure 13: Implementation after successful enrollment.

Figure 14: Implementation after failed enrollment.

iii) Delete Mode

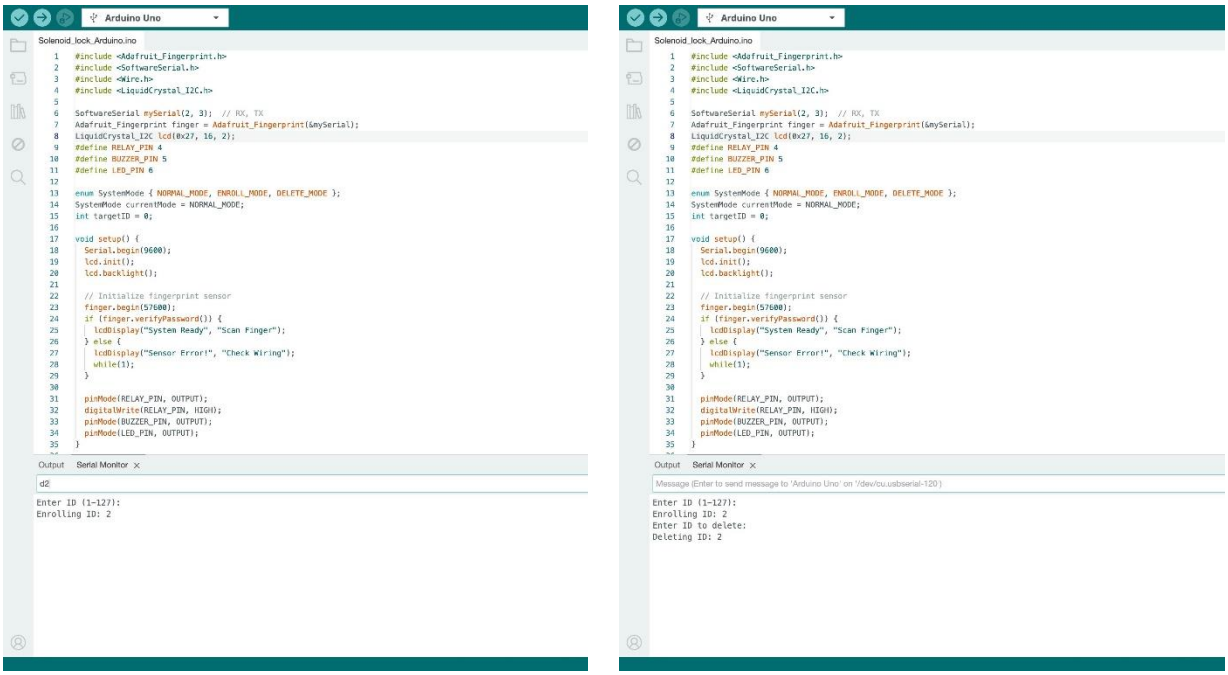

Figure 15: Deleting an ID using Arduino IDE

4 References

[1] (PDF) fingerprint door lock using Arduino, https://www.researchgate.net/publication/367160705_FINGERPRINT_DOOR_LOCK_USING_ARDUINO.

[2] “Arduino Based Automatic Fingerprint Door-Lock,” *projecthub.arduino.cc*. https://projecthub.arduino.cc/AnniBot/arduino-based-automatic-fingerprint-door-lock-c68204

[2] SEO2 Redcube, “Why Do We Need Biometric Door Lock?,” *Lockly*, Mar. 25, 2022. https://lockly.com/blogs/all/why-do-we-need-biometric-door-lock.