KWAME NKRUMAH UNIVERSITY OF SCIENCE AND TECHNOLOGY
COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER ENGINEERING

MICROPROCESSORS

Group 2

Amoako Ernestina Afriyie	1816322
Frimpong owusu Kelvin	1822122
Jesse Yaw Owusu	1826922
Abdallah Fatihu Mandeya	1812322
Wandaogo Adam Musah	1829522
Boadi Kingsley Naab	1819722
Blessing Kwaku Obinna Amegee	7090921
William Oseib Aborah	1826522
Jossph Lebuni Daveson	1821322
Nuamah Asamoaa Hafsa	1825522

ABSTRACT

Computing and embedded systems rely heavily on specialized components, particularly microprocessors and microcontrollers. Microprocessors operate as the primary computing unit in devices, adept at handling advanced calculations and complex processes, making them ideal for systems like desktop computers and data servers. On the other hand, microcontrollers, with their integrated structure combining processor, memory, and peripherals, are perfectly suited for tasks involving direct hardware control in embedded systems such as industrial machinery, household devices, and automobiles. While microprocessors thrive in sophisticated operations, microcontrollers excel in tasks requiring quick responses and minimal energy consumption.

Table of Contents

INTRODUCTION

Problem Statement

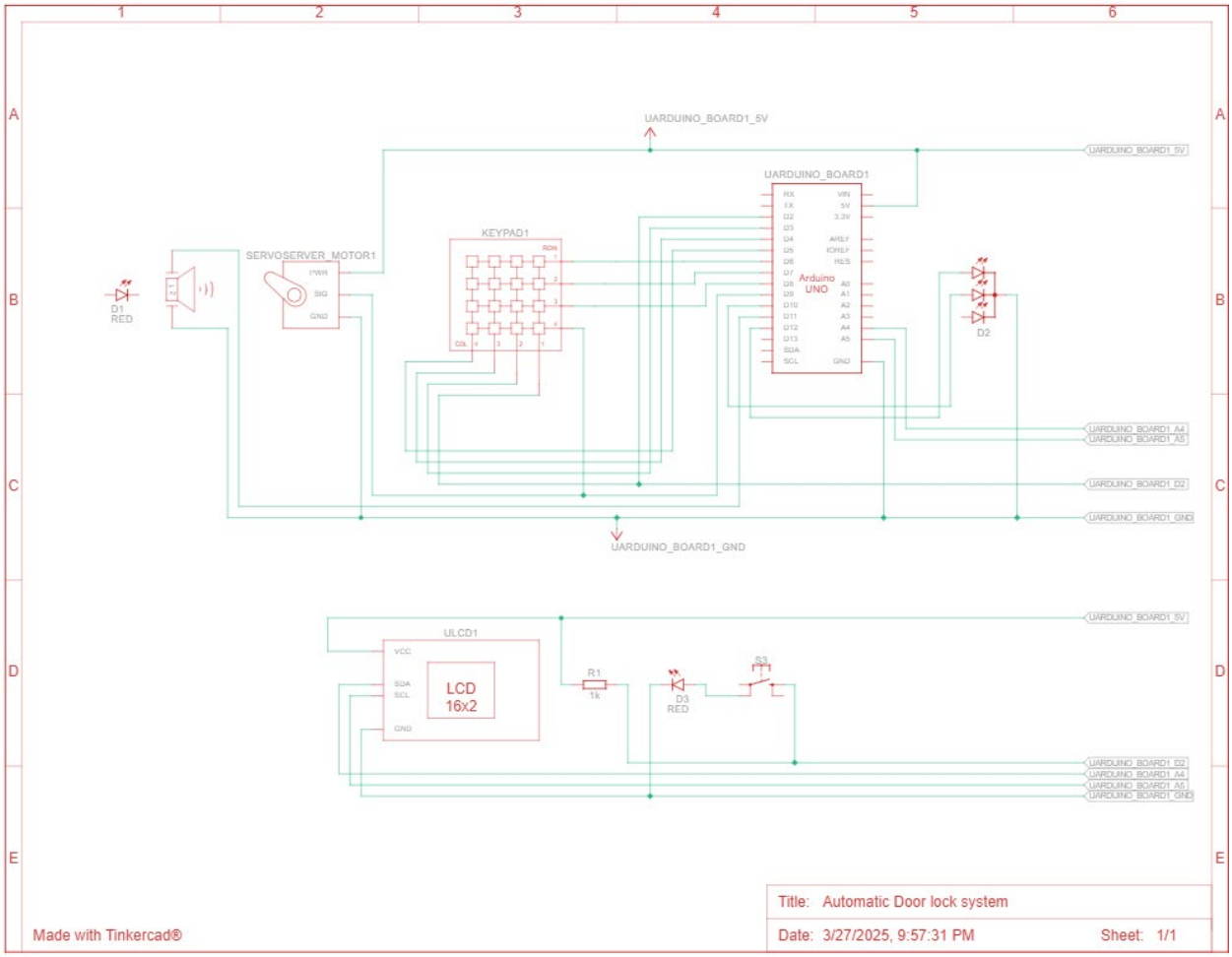
Traditional padlocks have several security flaws: they can be easily picked, keys can be duplicated, and they provide no real-time monitoring.

With burglary and unauthorized entry on the rise, relying solely on conventional locks is no longer sufficient to ensure safety.

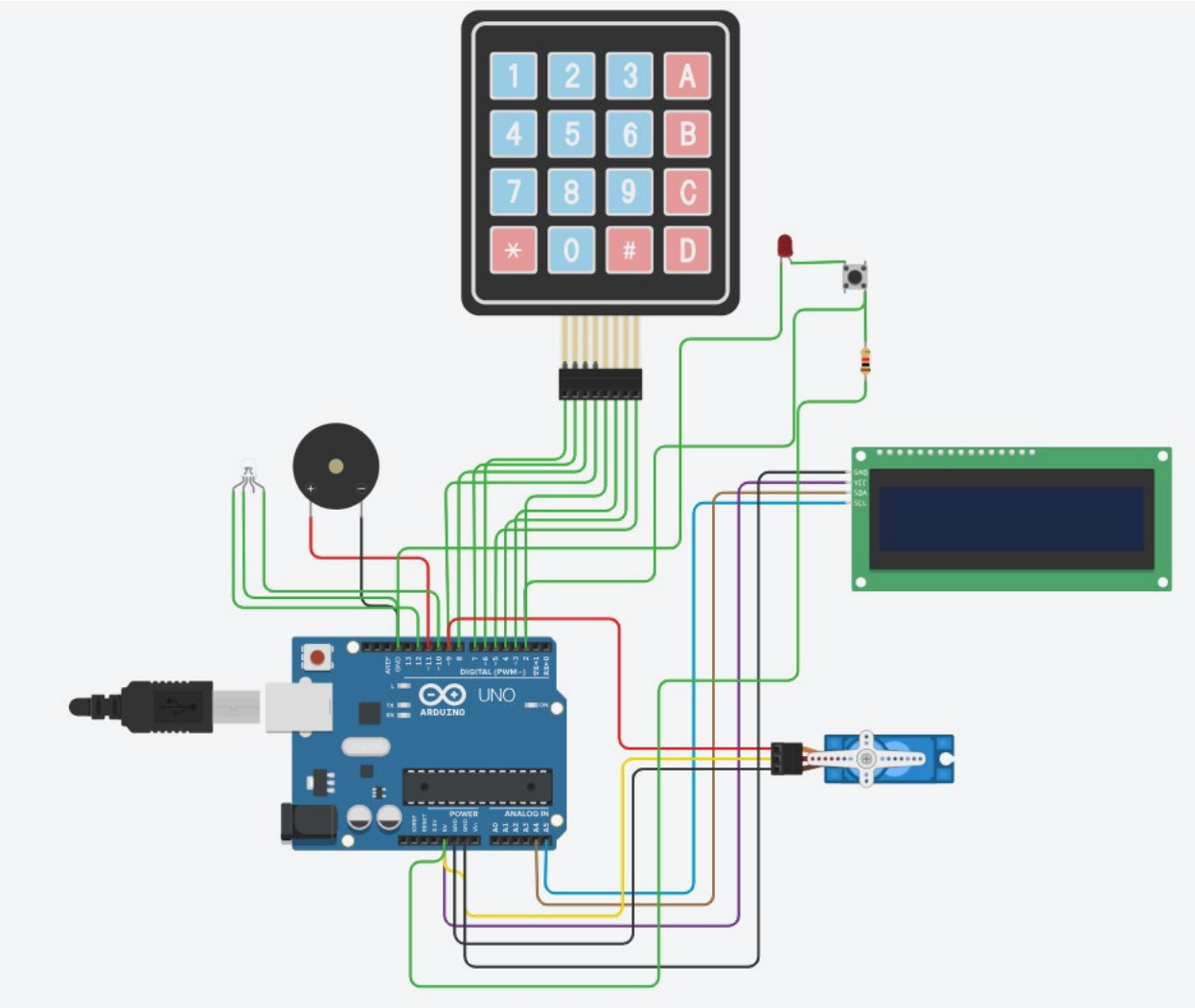
Many security breaches occur because lost or stolen keys grant immediate access to intruders.

This project aims to design an automated door lock system that enhances security by integrating password protection, eliminating key-related risks, and preventing unauthorized access. An automated door lock system offers enhanced security and convenience by combining hardware and embedded system technology. This project integrates components such as a servo motor, keypad, buzzer, and LCD display to create a smart locking mechanism. The system operates by requiring users to input a predefined password through the keypad, which is then validated. Upon successful authentication, the door unlocks temporarily, while incorrect entries trigger an error response. The apparatus uses real-time processing to manage user input, provide feedback, and control the locking mechanism, all displayed interactively on the LCD screen. This efficient design has potential applications in residential security systems, office environments, and industrial access control solutions.

SCHEMATIC


UARDUINO_BOARD1_5V
UARDUINO_BOARD1
KEYPAD1
SERVOSERVER_MOTOR1
D1
RED
D2
Arduino
UNO
UARDUINO_BOARD1_A4
UARDUINO_BOARD1_A5
UARDUINO_BOARD1_D2
UARDUINO_BOARD1_GND
UARDUINO_BOARD1_GND
ULCD1
LCD
16x2
R1
1k
D3
RED
S3
Made with Tinkercad®
Title: Automatic Door lock system
Date: 3/27/2025, 9:57:31 PM
Sheet: 1/1

CIRCUIT CONFIGURATION



COMPONENTS

Name	Quantity	Component
UArduino board1	1	Arduino Uno R3
KEYPAD1	1	Keypad 4x4
PIEZO2	1	Piezo
SERVOServer motor1	1	Positional Micro Servo
Ulcd1	1	PCF8574-based, 39 (0x27) LCD 16 x 2 (I2C)
S3	1	Pushbutton
D2	1	RCBG LED RGB
D1 D3	2	Red LED
R1	1	1 kΩ Resistor

METHODOLOGY

Pin Configuration

- Micro Servo
  - Signal = Pin9
  - Power = 5v
  - Ground = GND
- LCD 16*2 I2C
  - SCL = A5
  - SDA = A4
  - VCC = 5V
  - GND = GND
- Push Button
  - Terminal 2a = 5v
  - Terminal 1b = GND
  - Terminal 2b = RX->0
- Keypad 4*4
  - Column 1 = pin 2
  - Column 2 = pin 3
  - Column 3 = pin 4
  - Column 4 = pin 5
  - Row 1 = pin 6
  - Row 2 = pin 7
  - Row 3 = pin 8
  - Row 4 = pin 9
- Piezo

Positive = pin 11
Negative = GND

Functions of Each Component

1. **Arduino Uno (Microcontroller)**:
   - Acts as the brain of the project, controlling all connected components based on the program logic.
2. **Keypad (4x4)**:
   - Allows users to input the password to unlock the door.
   - Sends the entered password to the Arduino for processing.
3. **LCD (16x2) with I2C module**:
   - Displays messages such as "Enter Password" to guide the user.
   - Provides feedback on whether access is granted or denied.
4. **Pushbutton**:
   - Acts as a trigger to start the password input process.
   - Ensures the LCD only displays "Enter Password" after the button is pressed.
5. **Servo Motor**:
   - Controls the physical locking and unlocking mechanism of the door.
   - Rotates to a specified angle to unlock or lock the door.
6. **Piezo Buzzer**:
   - Provides audio feedback to indicate incorrect passwords or alarms.
   - Enhances security by signaling attempts to breach the system.
7. **Red and Green LEDs**:
   - Indicate the lock's status:
     - Red LED: Lock is engaged (access denied).
     - Green LED: Lock is disengaged (access granted).
8. **Power Supply**:
   - Provides the necessary voltage and current to operate all components.

How the System Works (Step-by-Step)

1. **Initialization**:
   - When powered on, the Arduino initializes the components (LCD, servo motor, etc.).
   - The system enters a standby state, with the door locked.
2. **Pushbutton Trigger**:
   - The user presses the pushbutton to activate the password input process.
   - The LCD displays "Enter Password" to prompt the user.
3. **Password Entry**:
   - The user enters a password via the 4x4 keypad.
   - Each keystroke is captured and processed by the Arduino.
   - The entered password is displayed partially or as asterisks on the LCD for user confirmation.
4. **Password Verification:**

   The Arduino compares the entered password with the pre-set password stored in its memory.

   If the password matches:

   The green LED lights up.

   The servo motor rotates to unlock the door.

   The LCD displays a success message ("Access Granted!").

   If the password does not match:

   The red LED lights up.

   The piezo buzzer sounds an alarm.

   The LCD displays an error message ("Access Denied!").
5. **Re-locking the Door**:
   - After a successful or failed attempt, the system resets.
   - The servo motor locks the door again by returning to the locked position.
   - The user can make another attempt by pressing the pushbutton.
6. **Security Features**:

- Incorrect passwords trigger the buzzer for a predefined duration, alerting nearby individuals.
- The system can be expanded to include additional security measures, such as an alarm or SMS alert on repeated failed attempts.

Key Advantages of the System

- Enhances home or office security by providing controlled access.
- User-friendly interface with LCD and keypad.
- Compact and efficient design powered by an Arduino.
- Scalable to include additional features like biometric authentication or smartphone integration.

Final Findings

1. **Enhanced Security**: The automated door lock system provides a reliable and efficient means of securing entry points in homes, offices, and industrial settings. The combination of hardware (servo motor, keypad) and software (embedded programming) ensures controlled access.

2. **User-Friendly Design**: The system's LCD interface and keypad make it easy for users to understand and operate. Clear visual and audio indicators ensure proper guidance during use.

3. **Responsive Mechanism**: The use of a microcontroller (Arduino Uno) ensures that all components work seamlessly in real-time, delivering quick responses to user inputs.

2. **Customization Potential**: The system can be scaled and upgraded with additional features like biometric authentication, smartphone integration, or remote monitoring for broader application.

Challenges Encountered

1. **Power Management**: During testing, intermittent power supply issues were observed, leading to unstable component performance.

2. **Button Debouncing**: The pushbutton occasionally registered false signals due to noise in the circuit, affecting the system's responsiveness.

3. **Password Memory**: Implementing secure storage and retrieval of passwords on the microcontroller posed challenges in ensuring data integrity.

4. **Component Synchronization**: Ensuring the servo motor, buzzer, LEDs, and LCD worked harmoniously required fine-tuning the code and circuit configuration.

5. **User Feedback Delay**: There were moments where the LCD response lagged slightly after user input, which could confuse the user.

Possible Improvements

1.**Battery Backup**: Integrate a rechargeable battery to address power outages and ensure system reliability in case of external power loss.

2. **Button Debouncing Circuit**: Add a capacitor or use software-based debouncing techniques to minimize false readings from the pushbutton.

3. **Password Encryption**: Enhance security by implementing encryption for storing and verifying passwords, reducing the risk of tampering.

4. **Wireless Connectivity**: Equip the system with Wi-Fi or Bluetooth modules for remote access, monitoring, and control via smartphones.

5.**Expand Security Measures**: Include additional features such as:

- Motion sensors to detect unauthorized attempts near the door.
- Alerts via SMS or email for repeated failed password entries.

6. **Performance Optimization**: Refine the programming to improve the system's speed and responsiveness, especially for LCD feedback.

7. **Scalability**: Adapt the system for multi-door setups, allowing centralized control for buildings with numerous access points.

References

1. **Arduino Documentation**: Arduino.cc. . (n.d.). *Arduino Reference*. Retrieved March 2025, from https://www.arduino.cc/reference/en/. This website offers detailed guidance on programming Arduino boards and integrating components like servos, LCDs, and keypads.

3. **Embedded Systems Design**: Yiu, J. (2015). *Embedded Systems Design with the ARM Cortex-M Microcontrollers*. This book is a valuable resource on embedded systems and their application in security solutions.

3. **Datasheets for Components**: Datasheets for individual components provide technical specifications and usage instructions:

- Servo Motor Datasheet: Retrieved from https://www.servodatasheet.com/.
- LCD I2C Datasheet: Retrieved from https://www.lcdmoduledatasheet.com/.
- Piezo Buzzer Datasheet: Retrieved from https://www.buzzerdatasheet.com/.
- Arduino Uno Datasheet: Retrieved from https://store.arduino.cc/arduino-uno-rev3.