International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 01 | Jan - 2025 SJIF Rating: 8.448 ISSN: 2582-3930

PASSWORD BASED CIRCUIT BREAKER

M. Charitha ECE IARE

Y.Chandu Priya ECE IARE

Abstract -
The project's goal is to operate a circuit breaker alone with the use of a password. To enter the password, a keypad is attached to the project. Because of poor communication and coordination between the maintenance crew and the electric substation employees, fatal electrical mishaps involving linemen are becoming more common during electric line repairs. A solution that can guarantee the safety of the maintenance personnel, such as linemen, is offered by the proposed system. The line man alone is in charge of turning the line ON or OFF. This system is set up so that a password is needed to turn the circuit breaker on or off. When a lineman returns to the substation after comfortably fixing a problem, the supply can be turned back on.
Keywords : Line man, Arduino, Keypad, circuit breaker, Relay, LCD.

1. INTRODUCTION

Nowadays, electrical accidents to the line man are increasing, while repairing the electrical lines due to the lack of communication between the electrical substation and maintenance staff. This project gives a solution to this problem to ensure line man safety. In this proposed system the control (ON/OFF) of the electrical lines lies with line man. This project is arranged in such a way that maintenance staff or line man has to enter the password to ON/OFF the electrical line. Now if there is any fault in electrical line then line man will switch off the power supply to the line by entering password and comfortably repair the electrical line, and after coming to the substation line man switch on the supply to the particular line by entering the password. The relay ON/OFF operation will be indicated by the LED's; also it sends a message to the receiver about the line disconnection. As soon as the maintenance work is finished then line man should enter the same password as used to disconnect the line earlier. Advantages:

• Save the life of line man.

• User friendly operation of main line.

• Easy to install and operate. Cost effective.

• Easy to maintain and repair

2. Body of Paper

With the increasing focus on security and convenience, password-based systems have gained significant attention. A password-based circuit breaker system provides a unique solution for securing electrical circuits by preventing unauthorized access and controlling the power flow based on a predefined password. This project aims to design a simple, efficient, and secure password-based circuit breaker system without the use of an LCD display.

The system uses a microcontroller (e.g., Arduino) to control the circuit breaker, and the password is entered via a keypad. The user is prompted through a series of LEDs or simple indicators, providing clear feedback about the status of the system.

2. Objective

The objective of this project is to design and implement a **password-based circuit breaker system** using an Arduino microcontroller, which aims to achieve the following:

- **Simplified Design**: Create a straightforward and cost-effective system that does not rely on additional components like LEDs, buzzers, or LCD displays. The design focuses on minimalism while still ensuring effective control of the circuit.
- **Secure Control**: Provide a secure means of controlling the electrical circuit by requiring the user to enter a password through a keypad before allowing or cutting off power flow
- **Password Verification**: Ensure that only users with the correct password can activate or deactivate the circuit breaker. Incorrect password entries will result in the relay remaining open, preventing unauthorized access.
- **Relay Control**: Use a relay to control the circuit breaker, enabling the system to either allow or

block the flow of electricity based on the correct or incorrect password input.

- **Basic Feedback**: While no visual or audio indicators are used, the system will be simple to operate and maintain, relying on the correct functioning of the keypad and the relay mechanism for user interaction.
- **Low-Cost Solution**: Provide an affordable and easily implementable security solution for electrical circuits, making it suitable for various applications, including home, industrial, or office environments..

3. System Components

The password-based circuit breaker system requires the following key components for its operation

1. **Arduino Microcontroller (e.g., Arduino Uno)**:

- A The central controller of the system that processes the password input and controls the relay to switch the circuit on or off.
- It reads inputs from the keypad and compares the entered password with the stored one.
- It activates or deactivates the relay based on the password verification.

2. **4x4 Matrix Keypad**:

- keypad with 16 buttons arranged in a 4x4 matrix configuration.
- Used for entering the password by the user.
- It sends the entered key signals to the Arduino for password verification.

3. **Relay Module**:

A relay acts as the circuit breaker and is used to control the power flow to the electrical load.

- It is controlled by the Arduino to either close (turn on) or open (turn off) the circuit.
- The relay is used for switching high-voltage circuits with the low-voltage Arduino output.

4, **Power Supply**:

A DC power supply (e.g., 9V battery or adapter) is used to power the Arduino and the relay module.

- The power supply ensures that the Arduino and relay have sufficient voltage and current for operation.

5. **Electrical Load** (e.g., lamp, motor, fan):

- The device that is controlled by the circuit breaker. The relay will either allow or cut off power to this load based on the password entered.
- This is the actual electrical system that the circuit breaker protects or controls

4. Working Principle

The password-based circuit breaker system operates by using a **microcontroller (Arduino)** to verify a password inputted through a **4x4 matrix keypad** and then controlling a **relay** to either allow or block the flow of electricity to the connected electrical load. The system works as follows:

1. System Initialization:

- When the system is powered on, the Arduino initializes the components (keypad and relay) and waits for the user to input a password.
- The predefined correct password is stored in the Arduino's memory, and the system is ready to receive input from the user.

2. Password Input:

- The user enters the password by pressing keys on the **4x4 matrix keypad**. The keypad consists of 16 buttons arranged in a grid of 4 rows and 4 columns.
- As each key is pressed, the corresponding character (digit or symbol) is captured by the Arduino and stored in a variable.

3. Password Verification:

- Once the user has entered a full password (typically 4 digits), the Arduino compares the entered password with the predefined correct password stored in its memory.
- If the entered password matches the correct password, the Arduino verifies the input as valid. If the entered password is incorrect, the system ignores the password and leaves the relay in its default state (open or off).

4. Relay Control:

- If the password is correct, the Arduino sends a signal to the **relay** to close the circuit. Closing the relay allows power to flow to the connected **electrical load** (such as a lamp, fan, or other devices).
- If the password is incorrect, the relay remains open, preventing power from reaching the electrical load, and the system is locked out from further operation until a correct password is entered.

5. No Visual or Audio Feedback:

- In this system, there are no **LEDs** or **buzzers** for feedback. The system functions solely based on the password entered and the relay's state. The user can determine if the password was correct by the status of the connected load (whether it is powered on or off).

6. Security Considerations:

- If an incorrect password is entered, the system does not proceed with activating the relay, ensuring the electrical circuit remains secure.
- To improve security, the system could include additional features, such as limiting the number of password attempts or implementing a delay between attempts, but these are not part of the basic design.

7. Circuit Operation:

- **Correct Password**: When the user enters the correct password, the Arduino sends a signal to the relay to close it, thereby activating the connected load and allowing power to flow.
- **Incorrect Password**: If the password is incorrect, the Arduino does not activate the relay, and the circuit remains open, cutting off the power supply to the load.

5. Circuit Design

The circuit consists of:

The circuit design for the password-based circuit breaker system is straightforward, consisting of the following key components:

1. **Arduino Microcontroller (e.g., Arduino Uno)**
2. **4x4 Matrix Keypad**
3. **Relay Module**
4. **Transistor (Optional)**
5. **Power Supply**
6. **Electrical Load (e.g., lamp, fan)**
7. **Connecting Wires**
8. **Resistors (Optional)**

Below is the description of how each component connects to form the circuit:

1. Arduino Microcontroller (e.g., Arduino Uno)

- The Arduino Uno will serve as the central processing unit that reads input from the keypad and controls the relay.
- **Pins Used**:
  - **Digital Pins**: These are used for reading input from the keypad and controlling the relay.
  - **Power Pins**: These provide power to the Arduino and other components.

2. 4x4 Matrix Keypad

- The 4x4 matrix keypad consists of 16 buttons arranged in 4 rows and 4 columns.
- **Connections**:
  - The **rows** and **columns** of the keypad are connected to specific **digital pins** on the Arduino. For example, rows can be connected to Arduino pins 9, 8, 7, and 6, and columns to pins 5, 4, 3, and 2.

3. Relay Module

- The relay is used to control the flow of power to the electrical load.
- **Connections**:
  - One end of the relay's **switching contacts** is connected to the **AC or DC load** (e.g., a lamp or fan).
  - The other end is connected to the **power supply** (either AC or DC).
  - The **control pin** of the relay is connected to one of the **digital pins** on the Arduino (e.g., Pin 12).
  - The **VCC** and **GND** pins of the relay module are connected to the **5V and GND** pins on the Arduino, respectively.

4. Power Supply

- A suitable **DC power supply** is used to power the Arduino and other components.
  - The **Arduino** can be powered with a 9V DC adapter or a USB connection.
  - The **relay module** may be powered through the Arduino's 5V pin or an external 5V power source, depending on the relay specifications.

5. Electrical Load

- The electrical load, such as a **lamp, fan, or motor**, is the device being controlled by the circuit breaker.
  - The load is connected to the **switching side** of the relay (i.e., between the relay's NO (Normally Open) and COM (Common) terminals).
  - The **power supply** to the load is connected to the relay's **COM terminal**.

6. Software Design

The software for the system needs to:

The **software design** for the password-based circuit breaker system using Arduino involves the development of a program that reads user input, compares it with the predefined password, controls hardware components (relay, LEDs), and provides feedback via the serial monitor. The software can be divided into key functional modules:

1. Initialization Module:

- **Function**: Initializes the necessary hardware components.
- **Tasks**:
  - Set up the **keypad** library for reading user input.
  - Configure **LED pins** for indicating password validation (Green for correct, Red for incorrect).
  - Configure **relay pin** for controlling the appliance.
  - Initialize **serial communication** for logging and debugging.

7. System Testing

Once the circuit is assembled and the code is uploaded to the microcontroller:

- Test the system by entering correct and incorrect passwords.

Keypad Sensitivity Test

- **Objective:** Verify that the keypad inputs are accurately read.

8.Conclusion

The **password-based circuit breaker system** designed using an **Arduino** is an effective way to control power to an appliance based on a password entered by the user. The key components—such as the keypad for password input, LEDs for feedback, the relay for controlling power, and the Serial Monitor for logging system status—work together to provide a functional and secure solution.

By **testing without an LCD display**, we used **LEDs** and the **Serial Monitor** for feedback, ensuring that the system responds correctly to user inputs. The testing process covered a range of scenarios, including:

- **Correct and incorrect password entries**.
- **Relay control** based on password validation.
- **Multiple failed attempts** and the system's response to incorrect inputs.
- **Password length validation** and proper handling of edge cases.

2.2.Software Components

- **Embedded C:** Used to program the Arduino UNO for system operations.
- **Arduino IDE:** Development environment for writing and uploading the code

2.3. Working

Working Principle The system's operation is based on the following steps:

The **password-based circuit breaker system** using **Arduino** operates by allowing users to control the power supply to an appliance via a relay, which is triggered by a correct password input through a keypad. The system works by validating the

password entered by the user and, depending on whether it is correct or incorrect, either allows or denies access to the appliance. Here's a breakdown of the working principle:

Key Components:

1. **Arduino Board** (e.g., Arduino Uno)
2. **4x4 Keypad** for user input
3. **Relay Module** for controlling the appliance
4. **LED Indicators** for feedback (Green for correct password, Red for incorrect)
5. **Serial Monitor** (for debugging and logging system messages)

Step-by-Step Working Principle:

1. Initial System Setup:

- The Arduino is powered on and the system begins in an idle state, waiting for the user to input a password.
- The **Keypad** and **LEDs** are initialized and set up to read user input and provide feedback.
- The **Relay** is connected to the appliance and remains in the **off** state by default, preventing any power from reaching the appliance until a valid password is entered.

2. User Input (Keypad Interaction):

- The user presses keys on the **4x4 Keypad** to enter the password.
    - Each key press is captured by the Arduino and processed to form an **entered password**.
- The entered password is stored in a string variable on the Arduino (enteredPassword).
- The system displays the entered password (via the **Serial Monitor**) for debugging purposes.

3. Password Validation:

- Once the user finishes entering the password, they press the # key to submit the password for validation.
- The system then compares the entered password (enteredPassword) with the predefined correct password (e.g., "1234").
    - **If the password is correct:**
        - The system activates the **Green LED**, providing visual feedback to the user that the password is correct.
        - The **Relay** is activated, allowing current to pass through to the appliance and turning it **on**.
        - The **Serial Monitor** prints "Password correct!" to inform the user.
    - **If the password is incorrect:**
        - The system activates the **Red LED**, providing visual feedback to the user that the password is incorrect.
        - The **Relay** remains deactivated, keeping the appliance **off**.
        - The **Serial Monitor** prints "Incorrect password, try again" to inform the user.

4. Password Reset (Clearing Input):

- If the user presses the * key on the keypad, the **entered password** is cleared.
- This allows the user to re-enter a new password and start the process over.

5. Feedback Indicators:

- **Green LED**: Lights up when the correct password is entered and the relay is activated.
- **Red LED**: Lights up when the wrong password is entered, and the relay remains off.
- The LEDs give visual feedback to the user about the system's status.

6. Relay Control (Powering the Appliance):

- The **Relay** is the key component that controls whether the appliance is powered on or off.
- When the correct password is entered, the **Relay** is activated, closing the switch and allowing power to flow to the appliance.
- When the incorrect password is entered, the **Relay** stays deactivated, preventing power from reaching the appliance.

7. Serial Monitor (Logging and Debugging):

- The **Serial Monitor** is used to provide debugging information and display system status:
    - It shows prompts like "Enter password".
    - It logs the result of password validation ("Password correct!" or "Incorrect password").

- It helps developers and testers verify the behavior of the system during development and troubleshooting.

8. Timeout or Lockout Mechanism (Optional Security Feature):

- For added security, the system can be modified to include a **lockout mechanism**. After a certain number of failed password attempts (e.g., 3 incorrect entries), the system can lock the user out for a certain period, preventing further attempts.
- This would be implemented using a counter to track failed attempts and a delay function to introduce a waiting period after exceeding the allowed number of attempts.

2.4.Circuit Implementation
The hardware setup includes:

- **Keypad**: Connect the keypad's rows and columns to Arduino digital pins (e.g., pins 2 to 9).
- **Relay**: Connect the relay module to Arduino digital pin (e.g., pin 10).
- **Appliance**: Connect the appliance to the relay's output, with the relay controlling the power flow.

2.5.Software Development The software for the gas leakage detection system is written in C++ using the Arduino IDE. Key features of the code include:

1. **Input Handling**: Capture user password via the keypad.
2. **Password Validation**: Compare entered password with predefined correct password.
3. **Relay Control**: Switch appliance power on/off based on password validity.
4. **Feedback Mechanism**: Use LEDs to show correct/incorrect password status.
5. **Debugging & Logging**: Output messages via Serial Monitor for user feedback and error checking.

2.6.Safety Considerations To ensure safe operation, the system incorporates:

1. **Proper Wiring**: Ensure all electrical connections, especially for the relay and appliance, are securely insulated and properly rated for the voltage and current.
2. **Relay Rating**: Use a relay with sufficient current and voltage ratings to safely handle the appliance's power requirements.
3. **Short Circuit Protection**: Implement circuit protection (e.g., fuses or circuit breakers) to prevent electrical hazards in case of faults.
4. **Overcurrent Protection**: Ensure the power supply can handle the load, and include protection to avoid overheating or electrical fires.
5. **Arduino Grounding**: Ensure the Arduino and the appliance share a common ground to avoid potential short circuits.
6. **Isolation**: Use a relay with electrical isolation to separate the high-voltage appliance circuit from the low-voltage Arduino components.

2.7 Benefits of the System

1. **Energy Efficiency**: Helps manage appliance usage by providing control over when devices are powered on or off, potentially reducing unnecessary energy consumption.
2. **Cost-Effective**: Implements a low-cost solution using affordable hardware components like Arduino, keypad, and relay.
3. **User Feedback**: Provides visual and serial feedback through LEDs and the Serial Monitor, helping users understand system status and errors.
4. **Safety**: Protects appliances from unauthorized use and potential hazards by using controlled relay activation.

2.8 .Steps Involved

The steps involved in implementing the **password-based circuit breaker system** are:

Hardware Setup:

1. Connect the **keypad**, **relay**, **LEDs**, and **Arduino**.
2. Ensure correct wiring for powering the appliance and controlling the relay.
2.

Arduino Programming:

1. Write code to read user input from the **keypad**.
2. Implement logic to compare the entered password with the predefined password.

3. Program relay control to turn the appliance on/off based on password validation.

Input Handling:

1. Capture the password input via the **keypad** and store it.
2. Implement clear/reset functionality (e.g., using * key).

Password Validation:

1. Compare the entered password with the correct password.
2. Provide feedback through **LEDs** (Green for correct, Red for incorrect).

Relay Control:

1. Control the **relay** based on password validation to power the appliance on or off.

Serial Monitoring:

1. Output logs to the **Serial Monitor** for debugging and user feedback.

Testing and Debugging:

1. Test the system to ensure proper functioning and address any errors in password entry or relay control.

2.9 Challenges Faced

Some challenges faced in implementing the **password-based circuit breaker system** include:

1. **Keypad Debouncing**: Ensuring accurate reading of keys without false triggers due to key bounce.
2. **Password Security**: Ensuring the password is securely validated, avoiding issues like accidental password resets or security breaches.
3. **Relay Control**: Properly handling high-voltage components while ensuring safe relay activation and deactivation.
4. **Wiring and Connections**: Managing multiple components (keypad, relay, LEDs) and ensuring correct wiring to prevent short circuits or malfunctions.
5. **Power Handling**: Ensuring the relay and Arduino components can safely handle the power requirements of the appliance.
6.

2.10.Applications

The **password-based circuit breaker system** has several practical applications:

1. **Home Automation**: Allows homeowners to control appliances (e.g., lights, fans, air conditioners) securely via password, adding convenience and energy management.
2. **Security Systems**: Used in restricted areas where access control is necessary, ensuring only authorized individuals can control electrical devices or systems.
3. **Industrial Equipment Control**: Enables secure operation of machinery or equipment in factories, reducing the risk of unauthorized operation.
4. **Power Management**: In environments like server rooms or labs, it provides secure power management for sensitive equipment, ensuring only authorized personnel can turn devices on/off.
5. **Office Systems**: Can be used for controlling office equipment (e.g., computers, printers) remotely or based on user authentication.
6. **Energy Conservation**: Helps manage energy consumption by controlling when devices are powered on/off through a secure, user-friendly interface.

2.11 Advantages

The **password-based circuit breaker system** offers several advantages:

1. **Enhanced Security**: Only authorized users with the correct password can control the appliance, preventing unauthorized access or tampering.
2. **Convenience**: Easy and quick control over appliances using a keypad, reducing the need for manual switching and providing remote access.
3. **Energy Efficiency**: Allows users to manage appliance usage, ensuring devices are powered off when not needed, leading to energy savings.
4. **Cost-Effective**: Uses affordable components like Arduino and keypads, making the system accessible and cost-efficient for both home and industrial use.
5. **User-Friendly**: Simple password entry interface, with visual feedback from LEDs and status updates on the serial monitor, making it easy to use.

6. **Safety**: Protects appliances from unauthorized use and ensures secure electrical control, reducing the risk of electrical hazards.
7. **Flexibility**: Can be customized to control multiple devices or add features like time-based access or failed attempt lockouts.

2.12 .Changes and Improvements for Future Research

For future research and improvements of the **password-based circuit breaker system**, the following changes and enhancements can be considered

Multi-User Authentication:

1. Implement support for multiple user passwords or biometric authentication (e.g., fingerprint or face recognition) for higher security.

Mobile App Integration:

1. Allow remote control via a mobile application using Bluetooth, Wi-Fi, or IoT platforms (e.g., MQTT, Blynk) for easy access and monitoring.

Time-based Access Control:

1. Introduce time-based locking mechanisms, where users can only control the appliance within specified time windows, adding an extra layer of security and control.

Failed Attempt Lockout:

1. Add an automatic lockout mechanism after a certain number of incorrect password attempts, preventing brute-force attacks and improving security.

Cloud Connectivity:

1. Integrate cloud services to allow remote monitoring, logging, and control of appliances through a web-based interface or cloud-based IoT platforms.

User Interface Enhancement:

1. Implement LCD or touch screen displays for a more interactive and user-friendly interface, displaying system status, current time, or appliance usage data.

Energy Monitoring and Reporting:

1. Incorporate energy monitoring features to track power consumption of controlled appliances and provide reports or alerts to users for energy efficiency.

Automatic Appliance Control:

1. Integrate sensors (e.g., motion sensors, temperature sensors) to automate appliance control based on environmental conditions (e.g., turn on fan when room temperature exceeds a threshold).

Integration with Smart Home Systems:

1. Allow integration with existing smart home ecosystems (e.g., Amazon Alexa, Google Assistant) to control appliances via voice commands for greater convenience.

3. CONCLUSIONS

The **password-based circuit breaker system** using Arduino provides a reliable, secure, and user-friendly method to control appliances based on password authentication. This system ensures that only authorized users can access and manage electrical devices, making it an ideal solution for both home and industrial settings where security and control are important.

Key benefits include enhanced security, energy efficiency, cost-effectiveness, and safety, as it prevents unauthorized access to appliances and minimizes the risk of electrical hazards. Additionally, it is easy to implement using widely available components like Arduino, keypads, and relays, making it an accessible solution for various applications.

With potential improvements such as multi-user authentication, mobile app integration, and energy monitoring, the system can be further expanded to provide advanced features, offering greater

flexibility and adaptability to modern smart homes and industrial environments.

Overall, the system serves as a practical and scalable solution for managing and securing electrical devices, offering a foundation for further research and development in the field of secure appliance control and automation.

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to all those who contributed to the successful completion of this project.

First and foremost, I would like to thank my **supervisor** for their guidance, expertise, and continuous support throughout the development of this project. Their valuable insights and encouragement helped shape the direction of the project.

I also want to acknowledge the **Arduino community** for their open-source resources and tutorials, which were invaluable in the design and implementation of the system.

A special thanks to my **family and friends** for their encouragement and understanding throughout this process, and for being a constant source of motivation.

Finally, I would like to thank everyone who contributed in any way, directly or indirectly, to the completion of this project. Their help, advice, and support have been truly appreciated.

REFERENCES

Arduino Documentation and Guides
Arduino. (2021). *Arduino Documentation*. Retrieved from https://www.arduino.cc

Keypad Library for Arduino
M. M. (2012). *Keypad Library Documentation*. Retrieved from https://www.arduino.cc/en/Reference/Keypad

Relay Module Control Using Arduino
HowToMechatronics. (2017). *How to Control a Relay with Arduino*. Retrieved from https://howtomechatronics.com/tutorials/arduino/arduino-relay-module/

Arduino Project Hub: Password System
Arduino Project Hub. (2019). *Password Authentication System Using Arduino*. Retrieved from https://create.arduino.cc/projecthub

Arduino Tutorials on LED Control
CircuitDigest. (2018). *How to Control LED with Arduino*. Retrieved from https://circuitdigest.com

Relay and Safety Considerations
Electronics Hub. (2020). *Relay Safety and Handling for Arduino Projects*. Retrieved from https://www.electronicshub.org

"The Arduino Cookbook" by Michael Margolis
Margolis, M. (2011). *Arduino Cookbook: Recipes to Begin, Expand, and Enhance Your Projects*. O'Reilly Media.

Keypad Authentication Systems
S. B. (2015). *Password-Based Access Control Systems for Home Automation*. International Journal of Computer Applications, 121(15), 1-6.