Password-Based-Circuit-Breaker

INTRODUCTION

A password-based circuit breaker solution aimed at addressing the rising number of fatal accidents involving lineman personnel due to electric shocks. These accidents often result from a need for coordination between maintenance staff and electric substation personnel.

It offers a robust solution to this critical problem, ensuring the safety of lineman personnel during maintenance tasks. The key feature is centralized control over the circuit, which is located at the substation. We have implemented a circuit breaker system that incorporates a secure keypad for password entry.

Here's how the system operates:

Password-Controlled Circuit: The lineman gains control over the circuit through a password-protected mechanism. This control is established at the substation.

Safety during Maintenance: When the lineman needs to perform repairs or maintenance work, they can enter the correct password on the keypad to switch OFF the circuit. This action ensures their safety by disconnecting the electrical supply.

Secure Work Environment: The lineman can confidently perform their tasks, knowing they have control over the circuit's status.

Reactivation after Maintenance: The lineman returns to the substation after completing the necessary repairs. To reactivate the circuit, they must once again enter the password on the keypad.

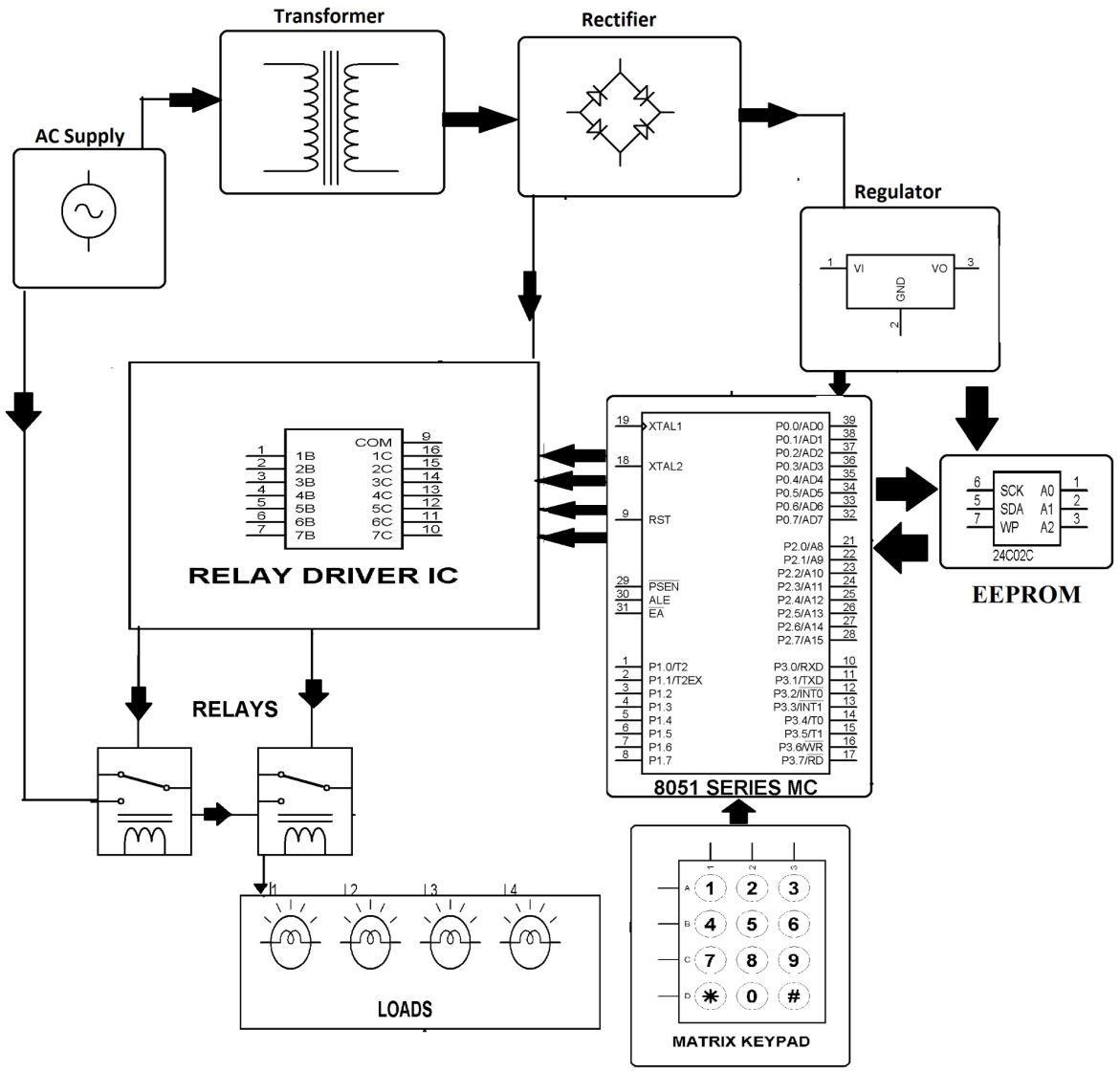
Accident Prevention: Because the control to switch ON/OFF the circuit is in the hands of the lineman, there is minimal risk of accidents caused by miscommunication or lack of coordination between different personnel.

To implement this system, we employ an 8051 microcontroller to manage the password verification process, and a matrix keypad is used to input the password. Furthermore, we plan to enhance the system by integrating an EEPROM, which will allow for password changes as needed, ensuring continued security and flexibility.

In summary, our password-based circuit breaker system offers a proactive solution to safeguard lineman personnel, reduce accidents, and improve coordination within electrical maintenance teams.

CIRCUIT DIAGRAM:

Password-Based-Circuit-Breaker



ASSEMBLY LANGUAGE PROGRAM:

```
; Define memory locations for variables
ORG 0x00
PASSWORD_LENGTH EQU 4
PASSWORD_ADDR EQU 0x10    ; Address in EEPROM to store the password

; Define ports and pins
PORT_A equ P1            ; Connect keypad to Port 1
EEPROM_DATA equ P2        ; Connect EEPROM data pins to Port 2
EEPROM_CTRL equ P3        ; Connect EEPROM control pins to Port 3

; Define constants
READ_EEPROM equ 0xA0      ; Read command for AT24C02 EEPROM
WRITE_EEPROM equ 0xA2     ; Write command for AT24C02 EEPROM

; Initialize variables
COUNT DB 0                ; Counter for password entry
```

Password-Based-Circuit-Breaker

```
PASSWORD DB PASSWORD_LENGTH DUP(0) ; Array to store entered password
CORRECT_PASSWORD DB PASSWORD_LENGTH DUP(0) ; Array to store correct
password

; Define delay subroutine
DELAY:
    MOV R2, #30
L1: DJNZ R2, L1
    RET

; Subroutine to read a character from keypad
READ_KEYPAD:
    MOV P0, #0xFF        ; Set Port 0 as input
    CLR P1.0             ; Enable the keypad
    MOV P1.1, #0         ; Set P1.1 low

    MOV A, #0xFF
    MOV B, #8

LOOP_ROW:
    MOV P1.2, A
    MOV A, P0
    JNB ACC.0, KEY_PRESSED
    RRC A
    DJNZ B, LOOP_ROW

KEY_PRESSED:
    MOV R5, #0
    MOV R4, #10          ; Delay for key debounce
    CALL DELAY
    MOV A, P0
    CJNE A, #0xFF, KEY_PRESSED

    MOV A, #0xFF
    MOV B, #8

LOOP_COL:
    MOV P1.1, A
    MOV A, P0
    JNB ACC.0, READ_DONE
    RRC A
    DJNZ B, LOOP_COL

READ_DONE:
    MOV R6, A            ; Store the key code in R6
    CLR P1.1             ; Disable the keypad
    RET
```

```
; Subroutine to write a character to EEPROM
WRITE_CHAR_TO_EEPROM:
    MOV DPTR, #PASSWORD_ADDR
    MOVX @DPTR, R6       ; Write the character to EEPROM
    INC DPTR
    INC DPTR
    MOVX @DPTR, A
    INC DPTR
    INC DPTR
    INC DPTR
    INC DPTR
    INC DPTR
    INC DPTR
    INC DPTR
    INC DPTR
    INC DPTR
    INC DPTR
    INC DPTR
    INC DPTR
    INC DPTR
    INC DPTR
    INC DPTR
    INC DPTR
    INC DPTR

    RET

; Subroutine to read a character from EEPROM
READ_CHAR_FROM_EEPROM:
    MOV DPTR, #PASSWORD_ADDR
    MOVX A, @DPTR
    INC DPTR
    INC DPTR
    MOVX A, @DPTR
    INC DPTR
    INC DPTR
    INC DPTR
    INC DPTR
    INC DPTR
    INC DPTR
    INC DPTR
    INC DPTR
    INC DPTR
    INC DPTR
    INC DPTR
    INC DPTR
    INC DPTR
    INC DPTR
    INC DPTR
    INC DPTR
```

```
    INC DPTR
    INC DPTR

    RET

MAIN:
    MOV DPTR, #PASSWORD_ADDR
    MOVX A, @DPTR          ; Read the correct password from EEPROM
    INC DPTR
    INC DPTR
    MOVX CORRECT_PASSWORD, A
    INC DPTR
    INC DPTR
    INC DPTR
    INC DPTR
    INC DPTR
    INC DPTR
    INC DPTR
    INC DPTR
    INC DPTR
    INC DPTR
    INC DPTR
    INC DPTR
    INC DPTR
    INC DPTR
    INC DPTR
    INC DPTR
    INC DPTR
    INC DPTR

    MOV P1, #0xFF          ; Set Port 1 as output
    MOV P2, #0xFF          ; Set Port 2 as output
    MOV P3, #0xFF          ; Set Port 3 as output

    MOV R7, #PASSWORD_LENGTH  ; Initialize password entry counter

    PASSWORD_LOOP:
        CALL READ_KEYPAD   ; Read a key from the keypad
        MOV PASSWORD[R7], R6 ; Store the entered key in the password
array
        MOV A, R6
        CALL WRITE_CHAR_TO_EEPROM ; Write the entered key to EEPROM

        MOV A, PASSWORD[R7]
        CJNE A, #0xFF, CHECK_PASSWORD
        SJMP PASSWORD_LOOP

    CHECK_PASSWORD:
        MOV A, PASSWORD[R7]
```

```
        MOV B, CORRECT_PASSWORD[R7]
        CJNE A, B, PASSWORD_INCORRECT

        DJNZ R7, PASSWORD_LOOP ; Continue checking the next
character
        ; If all characters match, the password is correct

    PASSWORD_CORRECT:
        ; Implement circuit control logic to turn OFF the circuit
        ; Here you can add code to deactivate the circuit

        ; Display a message or take further action as needed

        SJMP PASSWORD_LOOP

    PASSWORD_INCORRECT:
        ; Implement error handling or display logic to indicate
incorrect password
        ; Here you can add code to handle incorrect password entry

        SJMP PASSWORD_LOOP

    END
```

CONCLUSION:

In conclusion, the provided 8051 assembly language code represents a basic framework for a password-based circuit breaker system using a matrix keypad and EEPROM for password storage. While this code serves as a starting point for building such a system, it should be noted that developing a fully functional and secure system requires careful consideration of hardware specifics, robust error handling, and enhanced security features.