Microcontroller Based System Design Lab

EEE 4706

Project Title: **Password Protected DC Motor**

Submitted By:

Group No. : 01

Section : B2

- Abdullah Taharat, **190021232**
- Md. Asib Rahman Jahin, **190021202**
- Rayan Hossain Khan, **190021214**
- Md. Sakib-Ul-Islam, **190021216**
- Nasirullah, **190021238**

Date of Submission: 10.01.2024

Table of Contents

Introduction

In today's modern digital world, security and access control have become a rising concern in safeguarding sensitive information and secure details. From personal devices to industrial machinery, we need reliable authentication and safe access control mechanisms more than ever. In the realm of microcontroller systems, the Password Protected DC Motor project lets us create a significant application that addresses this demand for secure control and operation of essential devices.

In this report, we will dive into the project's system design, development, and implementation, detailing the core components of our Password Protected DC Motor system. We will provide insights into the fundamental concepts of microcontroller programming, electronic circuitry, and security systems, which describe the functionality of our system. Additionally, we will describe the challenges encountered during the project, our problem-solving approach, and the lessons learned throughout the development process of this project.

Objectives

The objectives of our project are:

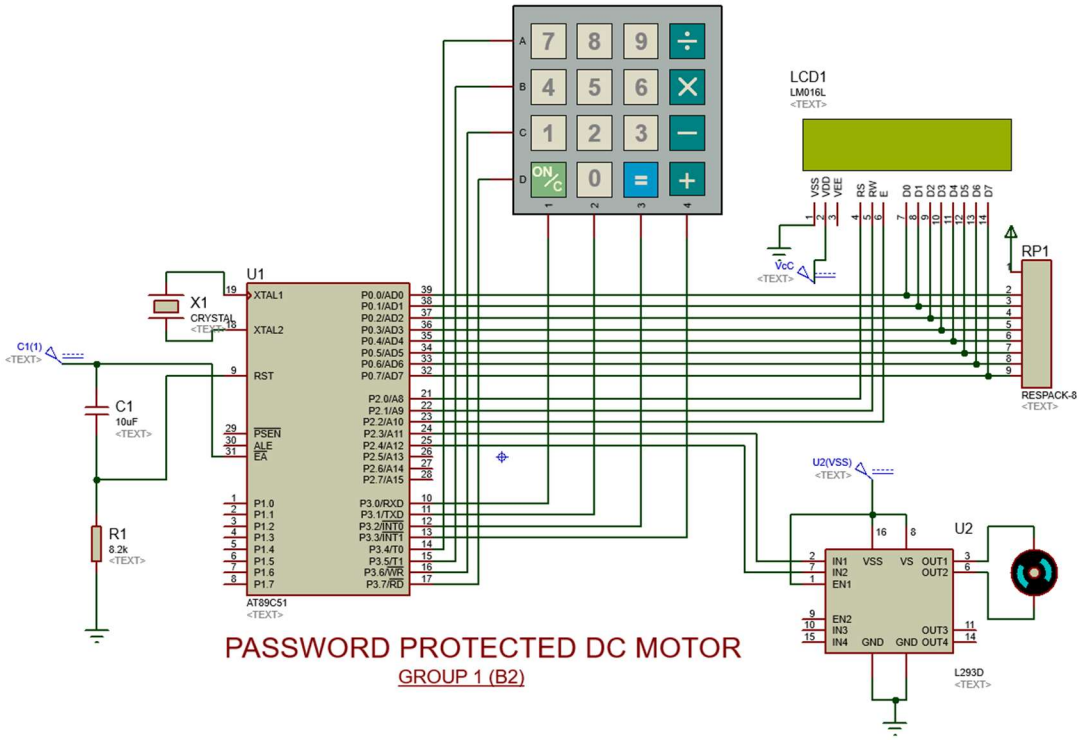
- To develop a password protected motor system
- To only let the dedicated user use the device
- To make it easier for the user to operate the motor after accessing the controls
- To learn the art of writing and debugging codes
- To understand the interfacing between different input and output devices within the microcontroller
- To be able to match theoretical knowledge in the world of practical implementations

Required Components

The following components were required to implement our project and their approximate market price are also listed:

Sl No.	Component Name	Value/Model	Quantity	Price (BDT)
1	8051 Microcontroller	AT89C51	1	3,500/-
2	Motor Driver	L293D	1	1,000/-
3	4x4 Matrix Keypad	KEYPAD-SMALLCALC	1	500/-
4	LCD Display	LM016L	1	1,000/-
5	Motor	5 V	1	50/-
6	Crystal Oscillator	11.0592MHz	1	300/-
7	Resistor	8.2 kΩ	1	50/-
8	Capacitor	33 pF, 10 uF	3	90/-
9	Push Button	-	1	50/-
10	DC source	5 V	3	300/-
11	Jumper Wires	-	21	250/-
Total				**7,090/-**

Circuit Diagram



Components used:

SL No.	Component Name	Value/Model	Quantity
1	8051 Microcontroller	AT89C51	1
2	Motor Driver	L293D	1
3	Keypad	KEYPAD-SMALLCALC	1
4	LCD Display	LM016L	1
5	Motor	5 V	1
6	Crystal Oscillator	11.0592MHz	1
7	Resistor	8.2 kΩ	1
8	Capacitor	33 pF, 10 uF	3
9	Push Button	-	1
10	DC source	5 V	3

Explanation:

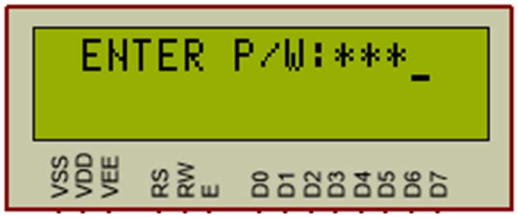
The usage and significance of the components used in this project are discussed below:

1. **8051 Microcontroller**: The 8051 microcontroller serves as the brain of our project, responsible for processing and executing the program logic to control the DC motor based on the user's input from the keypad. It provides the computational power and control capabilities required to manage the authentication process, interface with peripherals, and drive the motor.
2. **Motor Driver**: The motor driver acts as an interface between the microcontroller and the DC motor. It amplifies the control signals from the microcontroller to drive the motor. It ensures safe and efficient operation of the DC motor, translating the microcontroller's commands into actions, and protecting the microcontroller from potential voltage and current spikes.
3. **Keypad**: The keypad is an input device used for entering the password required to control the DC motor. Users enter the password via the keypad for authentication. It provides a user-friendly and secure means of input for authentication, making it a vital component of our access control system.
4. **LCD Display**: The LCD display serves as an output device, showing information related to the system status, such as prompting the user for the password and displaying messages. It enhances user interaction and provides feedback, making the system more intuitive and informative.
5. **Motor**: The DC motor is the output component of your project, and it performs the physical action, like opening a door or gate, upon successful password entry.
6. **Crystal Oscillator**: The crystal oscillator provides clock pulses to the microcontroller, ensuring precise and consistent timing for the execution of instructions in the program.
7. **Resistor and Capacitor**: These passive components are used in various parts of the circuit to set timing, voltage levels, and current flow, contributing to the overall stability and functionality of the system.
8. **Push Button**: Push buttons are used as control switches for system reset functions, providing manual input for specific actions.
9. **DC source**: The DC source provides the necessary power supply for all the electronic components, ensuring they operate within their specified voltage and current ranges.

Features

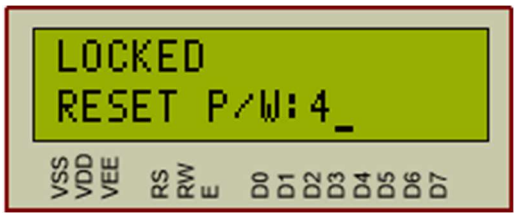
Mandatory Features: These are the mandatory features set by the course instructor. The device must perform all these features flawlessly:

1. Show * when a key is pressed during entering the password. This lets the user enter his password confidentially.



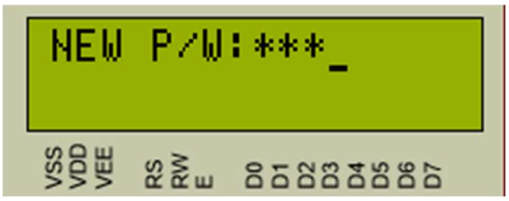
2. Implement an "ALMOST RIGHT" message when the input closely matches the password. We assumed this condition is true only when exactly one digit (any one digit among 4 digits) of the input password is entered wrong.

3. Locking the system for 3 unsuccessful tries. After the system is locked, the user has to press ‘4’ to reset the password and set a new password to operate the device again.

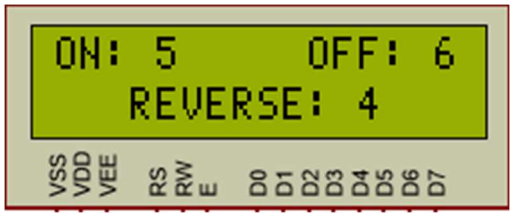


Additional Features:

1. Allowing the user to set a new password after 3 failed attempts, i.e. when the system is locked. After a new password is set, the user will be given the chance to try to enter again from the start. We will show * when a key is pressed during setting the new password.

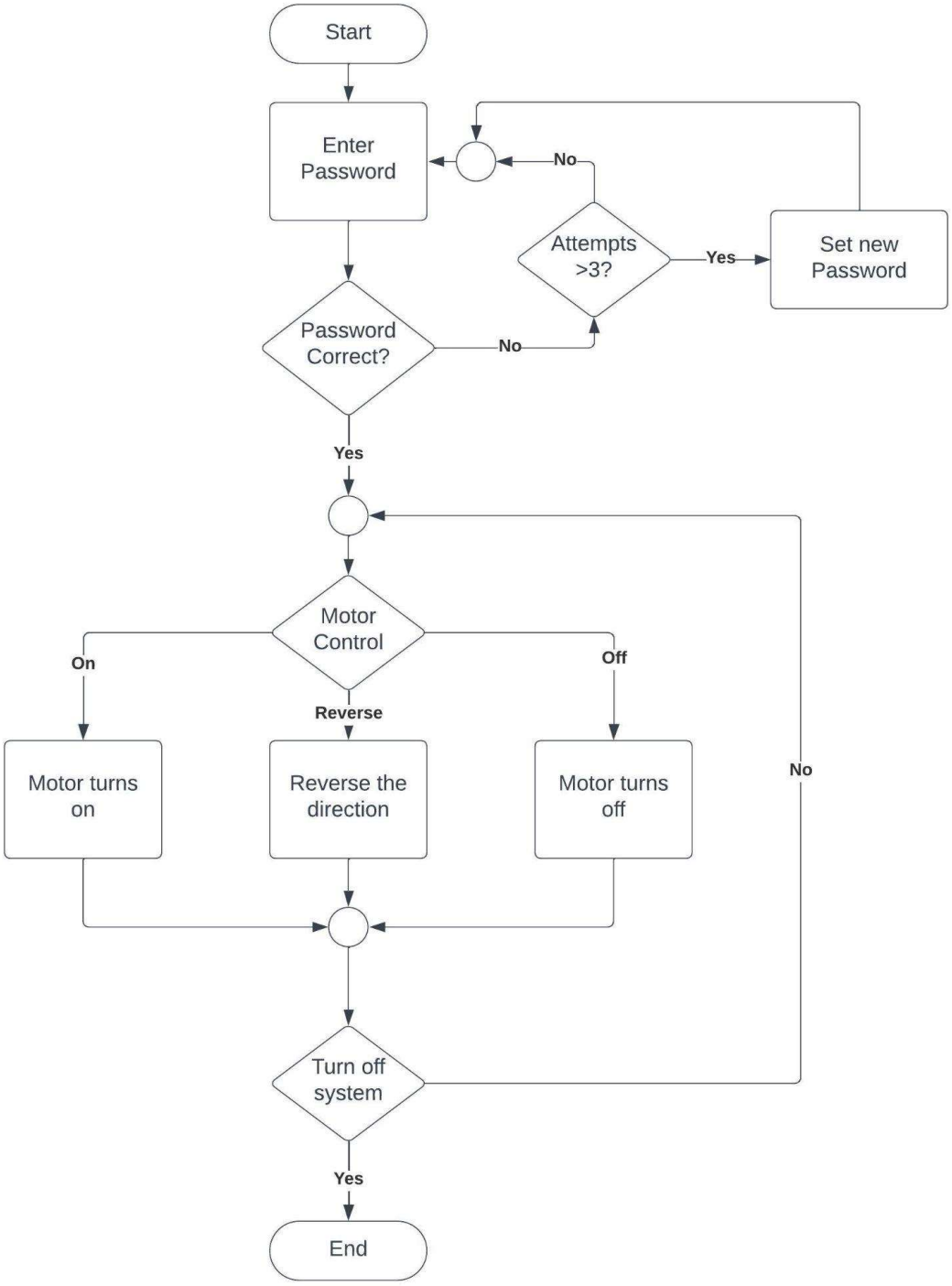


2. Option of turning on/off the motor as per the user's wish once the password is correctly entered. Now, the user will be allowed to turn on the motor without the need to enter the password again. Additionally, the will also be able to reverse the direction of rotation of the motor using the corresponding controlling key.



Working Principle

Flowchart:



Explanation:

Here's a summary of the steps involved in using the password-protected DC motor system:

Development Process:

1. **Code Writing:** Use MIDE-51 software to write assembly code for the system.
2. **Hex File Generation:** Generate a .hex file containing the machine code.
3. **Simulation and Testing:**
    - Burn the .hex file onto a microcontroller in Proteus.
    - Simulate and test the code thoroughly.
    - Debug and refine the code until it functions correctly.
4. **Hardware Setup:** Burn the working code onto the hardware setup using AVRDUDES.

User Interaction:

1. **Welcome Message:** Upon system startup, the LCD displays "WELCOME!"
2. **Password Prompt:** The LCD then displays "ENTER P/W: " to request password entry.
3. **Password Validation:** System checks the entered password:
    - If correct, display "ACCESS GRANTED" and proceed to motor control.
    - If incorrect:
        - Count the number of incorrect digits.
        - If more than 1 digit is wrong, display "ACCESS DENIED".
        - If only 1 digit is wrong, display "ALMOST RIGHT".
        - Allow up to 3 password attempts.

- **Lockout and Reset:**
    - If 3 incorrect attempts occur, the system locks.
    - Prompt the user to press '4' to unlock and set a new password.
    - The LCD then displays "NEW P/W: " to set a new password for the system.
    - Once the new password is set, the user is welcomed to the system again, and given the chance to enter the password and access the system through the use of the new password.

Motor Control (if access granted):

- **Instructions:** LCD displays instructions for motor control.
- **Keypad Controls:**
    - ➔ '5': Switch motor ON
    - ➔ '6': Switch motor OFF
    - ➔ '4': Reverse motor direction

Code

```
        ;DEFINING PINS AND THEIR CORRESPONDING FUNCTIONS FIRST
        RS EQU P2.0                     ;RS PIN OF LCD
        RW EQU P2.1                     ;RW PIN OF LCD
        E EQU P2.2                      ;ENBL PIN OF LCD
        MOTOR EQU P2.3                  ;MOTOR DRIVER PIN 1
        REV_MOTOR EQU P2.4              ;MOTOR DRIVER PIN 2
        PASSWORD_NO_DIGITS EQU 4        ;NUMBER OF DIGITS IN THE PASSWORD
        PIN_RESET_KEY EQU '4'
        ON_KEY EQU '5'
        OFF_KEY EQU '6'
        REV_KEY EQU '4'

        ;ROWS OF KEYPAD:
        ROW_A EQU P3.4
        ROW_B EQU P3.5
        ROW_C EQU P3.6
        ROW_D EQU P3.7

        ;COLUMNS OF KEYPAD:
        COL_1 EQU P3.0
        COL_2 EQU P3.1
        COL_3 EQU P3.2
        COL_4 EQU P3.3

        ORG 00H                 ;CODE STARTS FROM HERE ON
        MOV SP, #70H
        MOV PSW, #00H

        ;INITIALIZING INPUT & OUTPUT PORTS
        MOV P0,#00H
        MOV P3,#0FH
        MOV P2,#00H

        MOV R1, #50H                        ;POINTER 2 FOR STORING CORRECT PASSWORD
        MOV R3,#PASSWORD_NO_DIGITS          ;LOOP COUNTER FOR COPYING PASSWORD
        MOV DPTR,#REAL_PASSWORD

COPY_PASSWORD:      ;COPYING THE REAL PASSWORD TO RAM LOCATIONS STARTING FROM 50H
        CLR A
        MOVC A,@A+DPTR
        MOV @R1,A
        INC R1
        INC DPTR
        DJNZ R3,COPY_PASSWORD

START:  MOV DPTR, #MYLCD
LCD_IN: CLR A
        MOVC A,@A+DPTR
        ACALL COMNWRT
        ACALL DELAY
```

```
        JZ LOAD_WELCOME
        INC DPTR
        SJMP LCD_IN

LOAD_WELCOME:                       ;LOAD THE WELCOME MESSAGE
        MOV DPTR,#WELCOME
        ACALL TXT_LED
        LCALL DELAY_H

        MOV R7,#0       ;COUNTING THE NUMBER OF TIMES INCORRECT PASSWORD IS ENTERED

INITIALIZATION:
        CLR A
        MOV R3,A        ;INITIALLY SET R3 AND R4 AS ZERO
        MOV R4,A

        MOV R5, #PASSWORD_NO_DIGITS         ;NUMBER OF DIGITS IN THE PASSWORD
        MOV R0, #40H    ;POINTER 1 FOR CHECKING PASSWORD IS CORRECT OR NOT
        MOV R1, #50H    ;POINTER 2 FOR CHECKING PASSWORD IS CORRECT OR NOT

CONTINUE:
        LCALL CLEAR_DISPLAY_FINAL           ;CLEAR LCD DISPLAY

        MOV DPTR,#PASSWORD                  ;PROMPT USER FOR ENTERING THE PASSWORD
        ACALL TXT_LED

GO_:    MOV A,#8BH                          ;MOVE CURSOR TO CORRECT POSITION
        ACALL COMNWRT
        ACALL DELAY

KEYPAD: ACALL KEYBOARD                      ;KEYBOARD SUBROUTINE ACTIVATED AND CALLED

PASSWORD_STORE:
        CLR A
        MOVC A,@A+DPTR
        MOV @R0,A               ;STORE ENTERED PASSWORD IN RAM LOCATIONS
        MOV A,#'*'              ;SHOW ENTERED PASSWORD IN WITH '*' SIGN
        ACALL DATAWRT           ;CALL DISPLAY SUBROUTINE
        ACALL DELAY
        INC R0                  ;INCREMENT THE POINTER
        DJNZ R5,NEXT            ;UNLESS R5 BECOMES ZERO, TAKE THE NEXT PASSWORD DIGIT INPUT
        SJMP DO                 ;IF ENTIRE PASSWORD ENTERED THEN JUMP TO DO
        NEXT: SJMP KEYPAD

DO:     MOV R5,#0                                   ;COUNT THE NUMBER OF ERROR IN DIGITS
        MOV R2,#PASSWORD_NO_DIGITS                  ;LOOP COUNTER FOR PASSWORD_CHECK
        MOV R0,#40H
        MOV R1,#50H

REAL_PASSWORD_CHECK:        ;CHECKING IF THE ENTERED PASSWORD IS CORRECT OR NOT
        MOV A,@R0
        MOV B,A
        CLR A
        MOV A,@R1
```

```
        CJNE A,B,COUNT
        SJMP NOT_COUNT
COUNT: INC R5 ;IF THE ENTERED PASSWORD DOESN'T MATCH COUNT THE NUMBER OF DIGITS IN ERROR
NOT_COUNT:
        INC R0
        INC R1
        DJNZ R2,REAL_PASSWORD_CHECK

        CJNE R5,#0,NEXT2        ;IF R5 = 0, NO ERROR IN ENTERED PASSWORD
        SJMP GRANT              ;THEN GRANT PERMISSION TO USER FOR CONTROLLING THE MOTOR

NEXT2:  CJNE R5,#1,DENY         ;IF NUMBER OF DIGITS IN ERROR IS 1, WE ARE CLOSE
                                ;OTHERWISE WE SHOW ACCESS DENIED

VERY_CLOSE:
        ACALL NEW_LINE
        MOV DPTR,#CLOSE                 ;SHOWING THAT THEY ARE CLOSE TO THE REAL PASSWORD
        ACALL TXT_LED
        SJMP CHECK_ATTEMPTS             ;CHECKING IN THE 3RD ATTEMPT IS REACHED OR NOT

GRANT: ACALL NEW_LINE
        MOV DPTR,#GRANTED
;WE SHOW THAT THE USER HAS BEEN GRANTED THE PERMISSION WHEN PASSWORD IS CORRECT
        ACALL TXT_LED
        SJMP SUCCESS

DENY:   ACALL NEW_LINE
        MOV DPTR,#DENIED
;WE SHOW THAT THE ACCESS IS DENIED WHEN THE ENTERED PASSWORD IS NOT CORRECT AND NOT
CLOSE AS WELL
        ACALL TXT_LED
        SJMP CHECK_ATTEMPTS             ;CHECKING IN THE 3RD ATTEMPT IS REACHED OR NOT

CHECK_ATTEMPTS:                         ;CHECKING IF MAX NUMBER OF ATTEMPTS IS REACHED OR NOT
        INC R7                          ;INCREMENTING R7, FOR EACH ATTEMPT
        CJNE R7,#3,NEXTRY               ;IF R7 IS EQUAL TO 3, JUMP TO NEXTRY
        SJMP LOCK                       ;IF R7=3, LOCK THE SYSTEM
NEXTRY:
        LJMP INITIALIZATION
;START FROM BEGINNING, ASKING USER TO ENTER PASSWORD IN NEXT ATTEMPT

LOCK:   LCALL CLEAR_DISPLAY_FINAL       ;DISPLAYING THAT THE SYSTEM IS LOCKED
        MOV DPTR,#LOCKED
        ACALL TXT_LED

        ACALL NEW_LINE
        MOV DPTR,#RESET_PIN             ;SHOWING THE INSTRUCTION TO  RESET THE PIN
        ACALL TXT_LED
        ACALL KEYBOARD

;NEW_PASSWORD

CHECK_IF_4_PRESSED:             ;CHECKING IF 4 IS PRESSED, IF NOT THEN SYSTEM REMAINS LOCKED
        CLR A
```

```
        MOVC A,@A+DPTR
        MOV R4,A
        CJNE R4,#PIN_RESET_KEY,LOCK
;IF RESET KEY IS NOT PRESSED THEN THE SYSTEM WILL REMAIN IN LOCKED STATE

YES_4_PRESSED:
        MOV R1,#50H                     ;USE R1 POINTER TO STORE THE NEW PASSWORD
        MOV R5,#PASSWORD_NO_DIGITS      ;LOOP COUNTER FOR NEW PASSWORD ENTER

PROMPT_NEW_PASSWORD:
        LCALL CLEAR_DISPLAY_FINAL
        MOV DPTR,#NEW_PASSWORD          ;PROMPT NEW PASSWORD FOR THE USER TO ENTER
        ACALL TXT_LED

KEYPAD1:ACALL KEYBOARD                  ;TAKE INPUT

STORE_NEW_PASSWORD:
        CLR A
        MOVC A,@A+DPTR
        MOV @R1,A
        MOV A,#'*'
        ACALL DATAWRT                   ;call display subroutine
        ACALL DELAY
        INC R1
        DJNZ R5,NEXT_    ;UNLESS R5 BECOMES ZERO, TAKE THE NEXT PASSWORD DIGIT INPUT
        SJMP DONE_NEW_PIN_INPUT
;IF ENTIRE PASSWORD ENTERED THEN JUMP TO ONE_NEW_PIN_INPUT

NEXT_:  LJMP KEYPAD1  ;OTHERWISE CONTINUE TAKING THE PASSWORD INPUT

DONE_NEW_PIN_INPUT:
        LJMP START
;ONCE NEW PASSWORD IS ENTERED, WE GO TO THE BEGINNING, ASKING THE USER TO ENTER THE
PASSWORD AND ENTER THE SYSTEM

SUCCESS:                ;WE REACH HERE IF THE ENTERED PASSWORD BY USER IS CORRECT
        LCALL CLEAR_DISPLAY_FINAL
        MOV DPTR,#INSTRUCT_CONTROL      ;SHOWING THE CONTROL INSTRUCTIONS FOR THE MOTOR
        ACALL TXT_LED

        ACALL NEW_LINE
        MOV DPTR,#MOTOR_CONTROL
;SHOWING THE NEXT SET OF CONTROL INSTRUCTIONS FOR THE MOTOR
        ACALL TXT_LED

KEYPAD2:ACALL KEYBOARD
;TAKING INPUT FROM THE USER THTORUHG KEYPAD ALLOWING THEM TO CONTROL THE MOTOR

CHECK_IF_ON_PRESSED:        ;CHECKING IF THE ON BUTTON IS PRESSED OR NOT I.E. '5'
        CLR A
        MOVC A,@A+DPTR
        MOV R4,A
        CJNE R4,#ON_KEY,CHECK_IF_OFF_PRESSED
;IF ON KEY NOT PRESSED, THEN GO TO CHECK IF OFF PRESSED
```

```
YES_ON_PRESSED:                 ;COMES HERE ONLY IF '5' IS PRESSED
        CLR REV_MOTOR
        SETB MOTOR
        LJMP KEYPAD2   ;ALLOWING THE USER TO ENTER THE NEXT CONTROL INPUT THROUGH KEYPAD

CHECK_IF_OFF_PRESSED:       ;CHECKING IF THE OFF BUTTON IS PRESSED OR NOT I.E. '6'
        CLR A
        MOVC A,@A+DPTR
        MOV R4,A
        CJNE R4,#OFF_KEY,CHECK_IF_REV_PRESSED

YES_OFF_PRESSED:                ;COMES HERE ONLY IF '6' IS PRESSED
        CLR MOTOR
        CLR REV_MOTOR
        LJMP KEYPAD2   ;ALLOWING THE USER TO ENTER THE NEXT CONTROL INPUT THROUGH KEYPAD

CHECK_IF_REV_PRESSED:       ;CHECKING IF THE REVERSE BUTTON IS PRESSED OR NOT I.E. '4'
        CLR A
        MOVC A,@A+DPTR
        MOV R4,A
        CJNE R4,#REV_KEY,WRONG_PRESS

YES_REV_PRESSED:     ;COMES HERE ONLY IF '4' IS PRESSED
        CPL MOTOR
        CPL REV_MOTOR
        LJMP KEYPAD2  ;ALLOWING THE USER TO ENTER THE NEXT CONTROL INPUT THROUGH KEYPAD

WRONG_PRESS:          ;COMES HERE ONLY WRONG KEY IS PRESSED, SO NOTHING HAPPENS TO MOTOR
        LJMP KEYPAD2  ;ALLOWING THE USER TO ENTER THE CONTROL INPUT THROUGH KEYPAD
AGAIN

FINISH: SJMP FINISH

;SUBROUTINES ARE DEFINED FORM HERE ON
        ORG 400H
;FIRST THE KEBOARD SUBROUTINE
KEYBOARD:

K1:     CLR ROW_A      ;FIRST CHECK IF NO KEY IS PRESSED IN THE KEYBOARD
        CLR ROW_B
        CLR ROW_C
        CLR ROW_D
        MOV A,P3
        ANL A,#00001111B
        CJNE A,#00001111B,K1

K2:     ACALL DELAY    ;NOW CHECK IF KEY IS PRESSED IN THE KEYBOARD
        MOV A,P3
        ANL A,#00001111B
        CJNE A,#00001111B,K3
        SJMP K2

K3:     ACALL DELAY    ;AGAIN CHECK IF KEY IS PRESSED IN THE KEYBOARD
        MOV A,P3
```

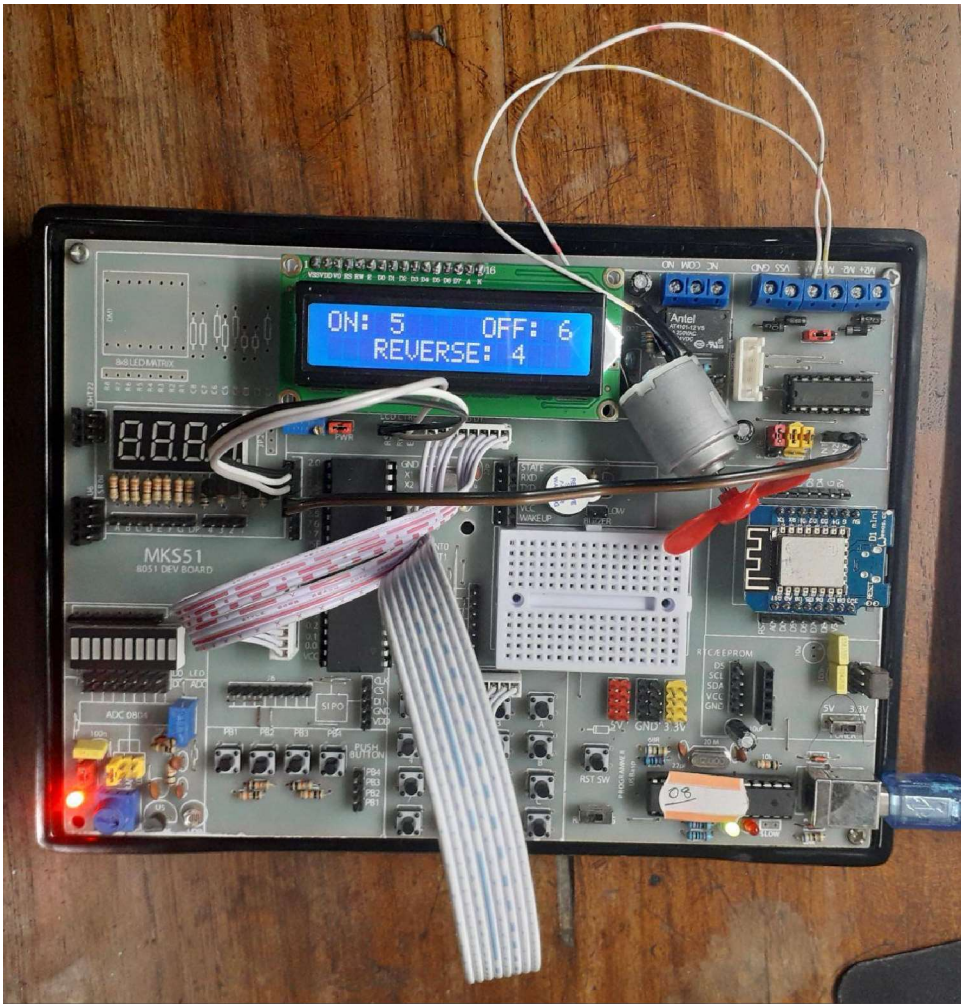
```
        ANL A,#00001111B
        CJNE A,#00001111B,K4
        SJMP K2

K4:     CLR  ROW_A      ;CHECKING IF KEY IS PRESSED FROM ROW_A
        SETB ROW_B
        SETB ROW_C
        SETB ROW_D
        MOV A,P3
        ANL A,#00001111B
        CJNE A,#00001111B,ROW_0

        SETB ROW_A      ;CHECKING IF KEY IS PRESSED FROM ROW_B
        CLR  ROW_B
        SETB ROW_C
        SETB ROW_D
        MOV A,P3
        ANL A,#00001111B
        CJNE A,#00001111B,ROW_1

        SETB ROW_A      ;CHECKING IF KEY IS PRESSED FROM ROW_C
        SETB ROW_B
        CLR  ROW_C
        SETB ROW_D
        MOV A,P3
        ANL A,#00001111B
        CJNE A,#00001111B,ROW_2

        SETB ROW_A      ;CHECKING IF KEY IS PRESSED FROM ROW_D
        SETB ROW_B
        SETB ROW_C
        CLR ROW_D
        MOV A,P3
        ANL A,#00001111B
        CJNE A,#00001111B,ROW_3
        LJMP K2

ROW_0: MOV DPTR,#KCODE0
        SJMP FIND
ROW_1: MOV DPTR,#KCODE1
        SJMP FIND
ROW_2: MOV DPTR,#KCODE2
        SJMP FIND
ROW_3: MOV DPTR,#KCODE3
        SJMP FIND

FIND:   RRC     A               ;SEE IF CY BIT IS LOW
        JNC     RETURN          ;IF ZERO, GET THE ASCII CODE
        INC     DPTR            ;IF CY IS NOT ZERO, POINT TO THE NEXT COLUMN ADDRESS
        SJMP    FIND            ;KEEP SEARCHING
RETURN: RET             ;UPON RETURN, THE DPTR HOLDS THE ASCII VALUE OF THE ENTERED KEY


;SUBROUTINE TO SHOW TEXT IN LCD
```

```
TXT_LED:CLR A
        MOVC A,@A+DPTR      ;COPYING THE DPTR ITEMS TO ACCUMULATOR ONE BY ONE
        JZ FUNC_END         ;IF ZERO, JUMP TO END OF FUNCTION
        ACALL DATAWRT
        ACALL DELAY
        INC DPTR
        SJMP TXT_LED
FUNC_END:
        RET

;SUBROUTINE TO CREATE NEW LINE IN THE LCD
NEW_LINE:
        MOV A,#0C0H
        ACALL COMNWRT
        ACALL DELAY
        RET

;SUBROUTINE TO CLEAR THE LCD DISPLAY
CLEAR_DISPLAY_FINAL:
        MOV DPTR, #CLEAR_DISPLAY  ;CLEARING DISPLAY INSTRUCTIONS ARE COPIED TO DPTR

START2:CLR A
        MOVC A,@A+DPTR
        ACALL COMNWRT
        ACALL DELAY
        JZ RETURNTO
        INC DPTR
        SJMP START2
        LCALL DELAY
RETURNTO: RET

;COMMAND WRITE SUBROUTINE FOR GIVING COMMAND TO LCD

COMNWRT:LCALL    READY              ;send command to LCD
        MOV      P1, A              ;copy reg A to port 1
        CLR      RS                 ;RS=0 for command
        CLR      RW                 ;R/W=0 for write
        SETB     E                  ;E-1 for high pulse
        ACALL DELAY                 ;give LCD some time
        CLR      E                  ;E=0 for H-to-L pulse
        RET

;DATA WRITE SUBROUTINE FOR WRITING DATA TO LCD
DATAWRT:LCALL READY                         ;write data to LCD
        MOV      P1, A              ;copy reg A to port1
        SETB     RS                 ;RS=1 for data
        CLR      RW                 ;R/W=0 for write
        SETB     E                  ;E=1 for high pulse
        ACALL DELAY                 ;give LCD some time
        CLR      E                  ;E=0 for H-to-L pulse
        RET

;READY SUBROUTINE FOR THE LCD
READY: SETB     P0.7                ;DEFINING P0.7 AS INPUT PIN FOR NOW
```

```
        CLR     RS                          ;RS=0 FOR COMMAND REGISTER
        SETB    RW                          ;R/W=1 FOR READ
WAIT:   CLR     E                           ;E=0 OF L-TO-H PULSE
        LCALL DELAY                         ;GIVE LCD SOME TIME
        SETB    E                           ;E=1 OF L-TO-H PULSE
        JB      P0.7, WAIT                  ;AS LONG THE P0.7 BIT IS HIGH, WE WILL WAIT
        RET

;DELAY SUBROUTINE
DELAY:  PUSH 3                              ;PUSH R3 ONTO STACK
        PUSH 4                              ;PUSH R4 ONTO STACK
        MOV     R3, #50                     ;DELAY OF APPROXIMATELY 27.83 ms
HERE2:  MOV R4, #255
HERE:   DJNZ    R4, HERE__                  ;STAY UNTIL R4 BECOMES 0
        DJNZ    R3, HERE2_                  ;LOAD
        POP 4                               ;POP R4 FROM THE STACK
        POP 3                               ;POP R3 FROM THE STACK
        RET

MYLCD : DB 38H,0EH,01,06,80H,0              ;INITIALIZING LCD INSTRUCTIONS STORED HERE
CLEAR_DISPLAY : DB 01,06,80H,0                      ;CLEARING LCD INSTRUCTIONS STORED HERE
WELCOME: DB '           WELCOME!        ',0     ;SHOWING WELCOME
PASSWORD: DB ENTER P/W::',0                 ;PROMPTING TO ENTER PASSWORD

REAL_PASSWORD: DB '5','4','5','4'           ;DEFAULT PASSWORD IS THIS

DENIED: DB 'ACCESS DENIED',0                ;SHOWING DENIED
GRANTED: DB 'ACCESS GRANTED',0              ;SHOWING GRANTED PERMIT
CLOSE: DB 'ALMOST RIGHT',0                  ;SHOWING ALMOST RIGHT MESSAGE
LOCKED: DB 'LOCKED',0                       ;SHOWING SYSTEM IS LOCKED
RESET_PIN: DB 'RESET P/W:4',0               ;SHOWING RESET PASSWORD OPTION
NEW_PASSWORD: DB 'NEW P/W::',0              ;PROMPTING NEW PASSWORD

INSTRUCT_CONTROL: DB ' ON: 5    OFF: 6 ',0          ;SHOWING MOTOR CONTROL OPTIONS
MOTOR_CONTROL: DB '         REVERSE: 4      ',0     ;SHOWING REVERSE MOTOR OPTION


;ASCII LOOK-UP TABLE FOR EACH ROW
KCODE0:         DB      '1','2','3','A'                 ;ROW 0
KCODE1:         DB      '4','5','6','B'                 ;ROW 1
KCODE2:         DB      '7','8','9','C'                 ;ROW 2
KCODE3:         DB      '*','0','#','D'                 ;ROW 3

END
```

Hardware Implementation

The figure below shows the finished HW part



Problems Faced

1. **Software Issue:**

- **Problem:** "Address out of range" error when using SJMP for subroutine calls.
- **Solution:** Use LJMP instead, as it can access a wider address range of address (64 kbytes).

2. **Hardware Issue:**

- **Problem:** Malfunctioning port 1 on the microcontroller board, possibly due to a manufacturing defect.
- **Solution:** Relocate connections from port 1 to port 0, leaving port 1 unused.

3. **Simulation Issue:**

- **Problem:** Simulation did not run properly after shifting connections from port 1 to port 0.
- **Solution:** Discovered that port 0 is designed for high-current devices, and an 8-way resistor pack must be connected to pins of port 0 in Proteus to accommodate its high-current design and ensure successful simulation in Proteus.

4. **LCD Screen Issue:**

- **Problem:** LCD screen didn't display matrix boxes when the new microcontroller board was connected.
- **Solution:** Adjusted the LCD contrast by rotating the potentiometer using a screwdriver and the boxes showed up.

Conclusion

In conclusion, the password-protected DC motor system successfully integrates security measures with user-friendly control. Developed using efficient assembly code and rigorously tested on both simulation and hardware platforms, this system grants access only to authorized users while offering clear feedback on password attempts. The keypad controls and informative LCD display empower users to operate the DC motor seamlessly, making this project a valuable example of secure and accessible design. Future advancements could involve incorporating additional security features like biometric authentication or time-based password expiration for even greater control and protection.