OBJECTIVE:

Design and develop an integrated door access control system aimed at providing secure and convenient door management through the use of soft keys. The primary objectives include implementing a user-friendly interface for local access via soft keys (such as a keypad or touchscreen) and enabling remote door control through a secure wireless connection. This system aims to offer seamless and reliable door opening/closing functionalities both on-site and remotely while prioritizing robust security measures to prevent unauthorized access.

INTRODUCTION:

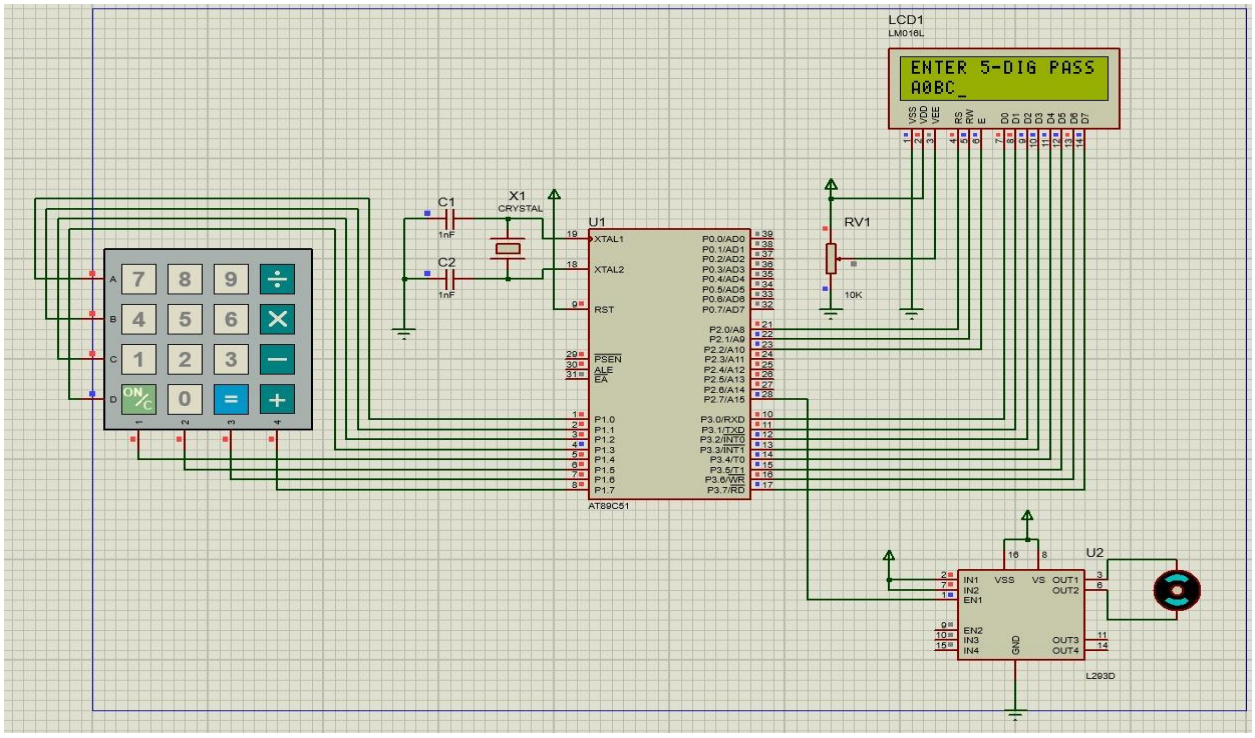
In an era where convenience intertwines with security, the need for efficient, adaptable door access systems has surged. Imagine a world where traditional keys are no longer the sole means of securing entry, where the simplicity of soft keys, accessible both locally and remotely, empowers seamless control over access points.

The advent of embedded systems has catalyzed the evolution of door control mechanisms, allowing for innovative solutions that transcend the limitations of physical keys. This solution presents a paradigm shift, offering a sophisticated yet user-friendly approach to managing entryways. This comprehensive exploration outlines the fundamental components, functionalities, security measures, and the development process involved in creating a door access system that transcends the limitations of conventional physical keys. It embodies the fusion of modern technology and security, heralding a new era in door access control.

COMPONENTS USED:

1. 8051(AT89S52) Micro-Controller Development Board
2. Bread board
3. 4x4 Matrix Keypad
4. LCD Display(16x2)
5. Potentiometer(10kohm)
6. Connecting Wires
7. DC Motor
8. L293D MOTOR DRIVE

SCEMATIC CAPTURE:



CIRCUIT DESCRIPTION:

Designing a digital clock using the 8051 micro controller involves interfacing the micro controller with a 4x4 MATRIX Keypad which is connected to port 1 and a LCD Display(16x2) whose data pins are connected to port3 and control pins are connected to port p2. Below is a simplified circuit description for a door-lock system using the 8051 micro controller:

Component description

1. **8051 AT89S52 microcontroller:**

It is a versatile platform for 8051-based embedded systems. It features the AT89S52 microcontroller with 8KB of Flash memory, 32 I/O pins, and UART for serial communication. The board includes essential components like crystal oscillator, reset circuit, and power supply, making it suitable for diverse applications. It supports in-system programming, allowing code updates without removing the microcontroller. With its compact design and rich feature set, the board is ideal for learning and prototyping 8051 microcontroller projects

2. **LCD (Liquid Crystal Display):**

It is a flat-panel technology commonly used for visual output in electronic devices. It consists of a grid of pixels, each containing liquid crystals that modulate light to produce images. LCDs are widely used in devices like TVs, monitors, and digital screens due to their thin profile, low power consumption, and ability to display high-quality images. They come in various types, such as TFT (Thin Film Transistor) and IPS (In-Plane Switching), offering different levels of performance and viewing angles. LCDs provide a reliable and versatile solution for displaying information in electronic devices.

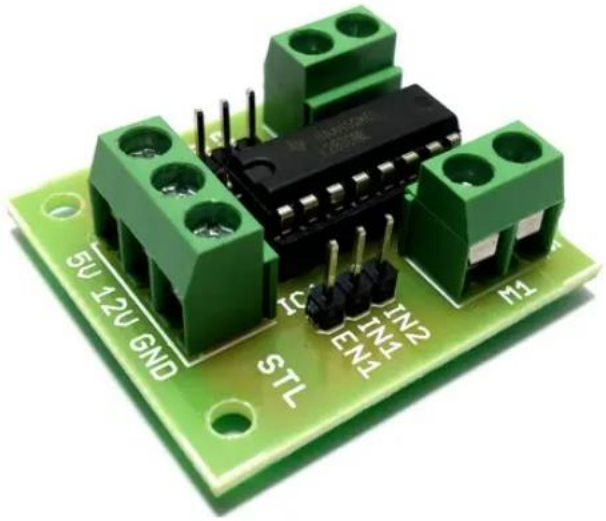
3. **4x4 keypad:**

A 4x4 keypad is a matrix keypad that consists of four rows and four columns, totaling 16 buttons in a grid layout. Each button represents a specific key or character, and the arrangement is similar to a standard telephone keypad. The keys on a 4x4 keypad are typically labeled with numbers 0 through 9 and the letters A to F, or they may have other symbols and functions depending on the specific application

4. **Potentiometer:**

A 10kohm potentiometer is a three-terminal variable resistor with 10,000 ohms of resistance. Its sliding wiper allows for adjustable resistance, commonly used in volume controls and voltage dividers. Widely employed in electronic circuits, it serves roles in audio equipment, tuning circuits, and as adjustable setpoints. The potentiometer's linear or logarithmic types cater to various applications, with the logarithmic version suitable for audio volume control.

5. DC MOTOR DRIVE



The L293D is a 16-pin Motor Driver IC which can control a set of two DC motors simultaneously in any direction. The L293D is designed to provide bidirectional drive currents of up to 600 mA (per channel) at voltages from 4.5 V to 36 V (at pin 8!). You can use it to control small dc motors

6. Dc motor

Working voltage: DC 3v 12v (suggest 3v 6v) maximum torque: 800gf cm min no load speed: 1: 48 load current: 70ma (250ma max) motor speed: 130 rpm length: approx. 65Mm/ 2. 56 Inch height: approx. 23Mm/0. 9 Inch thickness: approx. 18Mm/0. 71 Inch main color: yellow flat axis, has 1. 9Mm hole inner, can insert 2mm car axle closely package includes: 1 x DC geared motor.

Connection:

1. 8051 Micro controller Connections:

- Connect VCC and GND of the micro controller to the power supply.
- Connect the crystal oscillator across XTAL1 and XTAL2 pins of the micro controller.
- Connect the RESET pin to VCC through a pull-up resistor.

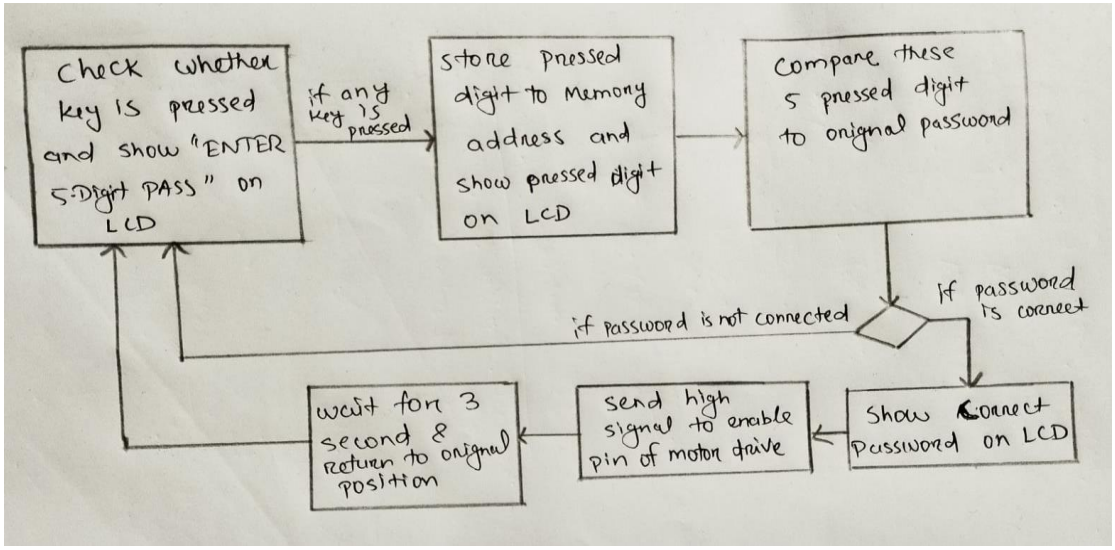
2. LCD Display(16x2) Connections:

- Connect Vss and Vdd of the LCD to ground and 5V respectively.
- Connect data pins to P3 port of 8051.
- Connect RS,R/W,E pins to p2.0,p2.1,p2.2 respectively.

3. 4x4 matrix keypad Connections:

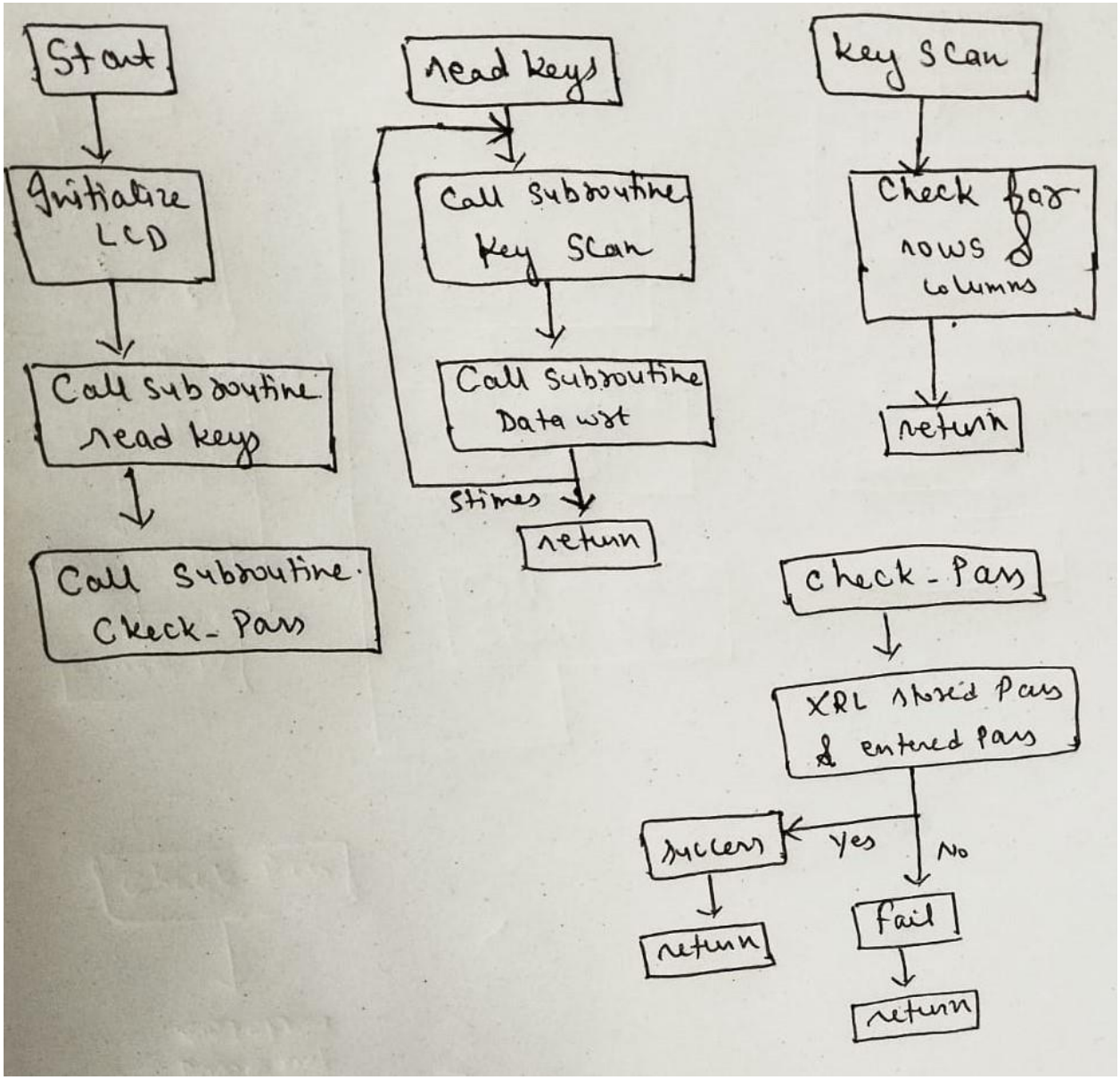
- Connect all row pins to p1.0 to p1.3
- Connect all column pins to p1.4 to p1.7

Blockdiagram

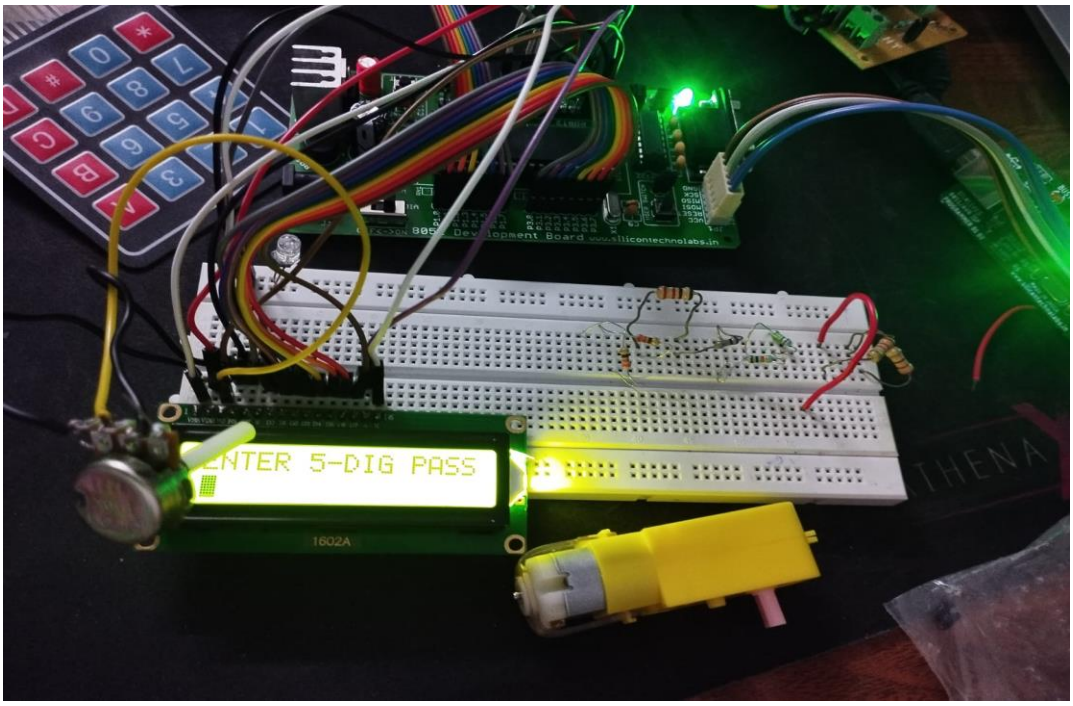


- **FLOW CHART:**

-


Start
Initialize LCD
Call Subroutine read keys
Call Subroutine Check_Pass
read keys
Call Subroutine Key Scan
Call Subroutine Data wrt
5times
return
Key Scan
Check for rows & columns
return
check_Pass
XRL stored Pass & entered Pass
Success
Yes
No
return
fail
return

HARDWARE DISPLAY:



ASSEMBLY CODE:

main.asm

```
org 0000H
sjmp main

MAIN: clr p2.7  ;CLEAR THE MOTOR TERMINAL
ACALL LCD_INIT   ;INITIALIZE LCD WITH INITIAL COMMANDS
ACALL DELAY    ;DELAY FOR 0.036 SEC
ACALL INIT_MSG  ;DISPLAYING INITIAL TEXT
ACALL DELAY
ACALL LINE2 ;MOVE THE CURSOR TO THWE BEGINNING OF 2ND LINE OF LCD
ACALL DELAY
;TAKING INPUT FROM 4X4 KEYPAD
ACALL READKEYS ;TAKING PASWORD FROM KEYPAD
ACALL DELAY
ACALL CLRSCR ; CLEAR THE SCREEN OF LCD
ACALL DELAY
ACALL PASS_CHECK  ;DISPLAYING TEXT
ACALL DELAY2  ;DELAY FOR 0.271 SECOND
ACALL CLRSCR   ;CLEAR THE LCD SCREEN
ACALL CHECK_PASS  ;CHECK FOR A PASSWORD WITH THE ENTERED PASSWORD AND STORED PASSWORD
SJMP MAIN


;LCD INITIALIZATION SUBROUTINE
;COMMAND: DB 38H,0FH,01H,06H,80H
LCD_INIT:MOV DPTR,#COMMAND
C1:CLR A
MOVC A,@A+DPTR
JZ DAT
ACALL COMWRT ;CALL SUBROUTINE TO SEND COMMAND TO LCD
ACALL DELAY
INC DPTR
SJMP C1
DAT:RET

```

main.asm

```
34
35  ;INI_MSG: DB "ENTER 5-DIG. PASS",0
36  INIT_MSG:MOV DPTR,#INI_MSG
37  C2:CLR A
38  MOVC A,@A+DPTR
39  JZ RT
40  ACALL DATAWRT  ; CALL SUBROUTINE TO SEND DATA TO LCD
41  ACALL DELAY
42  INC DPTR
43  SJMP C2
44  RT:RET
45
46
47  ;SEND COMMAND TO INITILIZE THE LCD
48  COMWRT:MOV P3,A  ; MOVE THE CONTENT FROM A TO P3
49  CLR P2.0     ;REGISTER SELECT FOR COMMAND
50  CLR P2.1     ; WRITE OPERATION
51  SETB P2.2     ;ENABLE HIGH
52  ACALL DELAY
53  CLR P2.2      ;ENABLE LOW
54  RET
55
56  ;SEND DATA TO LCD
57  DATAWRT:MOV P3,A    ; MOVE THE CONTENT FROM A TO P3
58  SETB P2.0      ;REGISTER SELECT FOR DATA
59  CLR P2.1     ; WRITE OPERATION
60  SETB P2.2  ;ENABLE HIGH
61  ACALL DELAY
62  CLR P2.2  ;ENABLE LOW
63  RET
64
65
```

main.asm

```
67
68  ;MOVE TO SECOND LINE OF LCD
69  LINE2:MOV A,#0C0H
70  ACALL COMWRT  ;CALL SUBROUTINE TO SEND COMMAND TO LCD
71  RET
72
73  CLRSCR:MOV A,#01H ;CLEAR THE LCD SCREEN
74  ACALL COMWRT  ;CALL SUBROUTINE TO SEND COMMAND TO LCD
75  RET
76
77  ; SUBROUTINE TO READ THE PASSWORD FROM 4X4 KEYPAD
78  READKEYS:MOV R0,#5 ;LENGTH OF PASSWORD
79  MOV R1,#2FH  ;MEMORY ADDRESS TO STORE THE PASSWORD
80  ACALL DELAY
81  ROTATE:ACALL KEY_SCAN  ;MOVE TO SUBROUTINE TO CHECK WHETHER A KEY IS PRESSED OR NOT AND STORE THE CORRESPONDING VALUE TO A
82  MOV @R1,A  ;TAKE THE VALUE OF A AND STORE IN MEMORY LOACTION POINTED BY R1
83  ACALL DELAY2
84  ACALL DATAWRT  ;SHOW DATA IN LCD
85  ACALL DELAY2
86  INC R1   ; STORE THE NEXT DATA IN NEXT MEMORY LOCATION
87  DJNZ R0,ROTATE ;SIMILAR FOR NEXT DIGIT
88  RET
89
90  ;SUBROUTINE TO CHECK WHETHER A KEY IS PRESSED OR NOT
91  KEY_SCAN:
92  ;SCANNING KEY PRESS FOR 1ST ROW
93  MOV P1,#0FFH ; MAKE PORT 1 AS INPUT PORT
94        CLR P1.0  ;CLEAR P1.0 TO CHECK WHETHER A KEY IS PRESSED FROM ROW 1 OR NOT
95        JB P1.4,NEXT1 ; CHECK FOR COLUMN , IF P1.4 IS 1 KEY IS PRESS FROM COLUMN 1
96        MOV A,#"1";MOVE 1 IN A
97        RET
98  NEXT1:JB P1.5,NEXT2 ; CHECK FOR COLUMN , IF P1.4 IS 1 KEY IS PRESS FROM COLUMN 2
99        MOV A,#"2"
100       RET
```

main.asm

```
KEY_SCAN:
;SCANNING KEY PRESS FOR 1ST ROW
MOV P1,#0FFH ; MAKE PORT 1 AS INPUT PORT
      CLR P1.0  ;CLEAR P1.0 TO CHECK WHETHER A KEY IS PRESSED FROM ROW 1 OR NOT
      JB P1.4,NEXT1 ; CHECK FOR COLUMN , IF P1.4 IS 1 KEY IS PRESS FROM COLUMN 1
      MOV A,#"1";MOVE 1 IN A
      RET
NEXT1:JB P1.5,NEXT2 ; CHECK FOR COLUMN , IF P1.4 IS 1 KEY IS PRESS FROM COLUMN 2
      MOV A,#"2"
      RET
NEXT2:JB P1.6,NEXT3  ; CHECK FOR COLUMN , IF P1.4 IS 1 KEY IS PRESS FROM COLUMN 3
      MOV A,#"3"
      RET
NEXT3:JB P1.7,NEXT4  ; CHECK FOR COLUMN , IF P1.4 IS 1 KEY IS PRESS FROM COLUMN 4
      MOV A,#"A"
      RET
;SCANNING KEY PRESS FOR 2ND ROW
NEXT4:SETB P1.0
      CLR P1.1  ;CLEAR P1.0 TO CHECK WHETHER A KEY IS PRESSED FROM ROW 2 OR NOT
      JB P1.4,NEXT5 ; CHECK FOR COLUMN , IF P1.4 IS 1 KEY IS PRESS FROM COLUMN 1
      MOV A,#"4"
      RET
NEXT5:JB P1.5,NEXT6  ; CHECK FOR COLUMN , IF P1.4 IS 1 KEY IS PRESS FROM COLUMN 2
      MOV A,#"5"
      RET
NEXT6:JB P1.6,NEXT7  ; CHECK FOR COLUMN , IF P1.4 IS 1 KEY IS PRESS FROM COLUMN 3
      MOV A,#"6"
      RET
NEXT7:JB P1.7,NEXT8 ; CHECK FOR COLUMN , IF P1.4 IS 1 KEY IS PRESS FROM COLUMN 4
      MOV A,#"B"
      RET
```

main.asm

```
      RET
;SCANNING KEY PRESS FOR 3RD ROW
NEXT8:SETB P1.1
      CLR P1.2  ;CLEAR P1.0 TO CHECK WHETHER A KEY IS PRESSED FROM ROW 3 OR NOT
      JB P1.4,NEXT9  ; CHECK FOR COLUMN , IF P1.4 IS 1 KEY IS PRESS FROM COLUMN 1
      MOV A,#"7"
      RET
NEXT9:JB P1.5,NEXT10  ; CHECK FOR COLUMN , IF P1.4 IS 1 KEY IS PRESS FROM COLUMN 2
      MOV A,#"8"
      RET
NEXT10:JB P1.6,NEXT11  ; CHECK FOR COLUMN , IF P1.4 IS 1 KEY IS PRESS FROM COLUMN 3
       MOV A,#"9"
       RET
NEXT11:JB P1.7,NEXT12  ; CHECK FOR COLUMN , IF P1.4 IS 1 KEY IS PRESS FROM COLUMN 4
       MOV A,#"C"
       RET
;SCANNING KEY PRESS FOR 4TH ROW
NEXT12:SETB P1.2
       CLR P1.3  ;CLEAR P1.0 TO CHECK WHETHER A KEY IS PRESSED FROM ROW 4 OR NOT
       JB P1.4,NEXT13  ; CHECK FOR COLUMN , IF P1.4 IS 1 KEY IS PRESS FROM COLUMN 1
       MOV A,#"*"
       RET
NEXT13:JB P1.5,NEXT14  ; CHECK FOR COLUMN , IF P1.4 IS 1 KEY IS PRESS FROM COLUMN 2
       MOV A,#"0"
       RET
NEXT14:JB P1.6,NEXT15  ; CHECK FOR COLUMN , IF P1.4 IS 1 KEY IS PRESS FROM COLUMN 3
       MOV A,#"#"
       RET
NEXT15:JB P1.7,BACK  ; CHECK FOR COLUMN , IF P1.4 IS 1 KEY IS PRESS FROM COLUMN 4
       MOV A,#"="
       RET
       BACK:LJMP KEY_SCAN
```

main.asm

```
154
155 ;PASSWORD: DB "A","0","B","C","*"
156 CHECK_PASS:MOV R0,#5  ;LENGTH OF PASSWORD
157 MOV R1,#2FH    ;MEMORY ADDRESS
158 MOV DPTR,#PASSWORD
159 RPT:CLR A
160 MOVC A,@A+DPTR    ;MOVE THE CONTENT OF A+DPTR TO A AND CHECK FOR CORRECT PASSWORD
161 XRL A,@R1  ;CHECKING FOR PASSWORD, IF 'A' AND CONTENT OF R1 IS SAME THE XOR OPERATION RESULT IN 0 ELSE NOT ZERO
162 JNZ FAIL  ;JUMP TO FAIL IF A IS NOT ZERO
163 INC R1
164 INC DPTR
165 DJNZ R0,RPT  ;SUCCESSIVELY SEND NEXT DIGIT AND REPEAT THIS
166 ACALL SUCCESS
167 RET
168
169 ;TEXT_FAIL: DB "WRONG PASSWORD",0
170 FAIL:ACALL DELAY
171 MOV DPTR,#TEXT_FAIL
172 ACALL SEND_DATA  ;SEND TEXT TO DISPLAY ON LCD
173 ACALL DELAY2
174 LJMP MAIN   ; IF PASSWORD IS WRONG START THE WHOLE PROESS AGAIN
175
176 ;TEXT3: DB "CHECKING PASS...",0
177 PASS_CHECK:MOV DPTR,#TEXT3 ;SUBROUTINE TO DISPLAY TEXT CHECKING PASS....
178 T9:CLR A
179 MOVC A,@A+DPTR
180 JZ DI
181 ACALL DATAWRT  ;SEND DATA TO LCD
182 ACALL DELAY
183 INC DPTR
184 SJMP T9  ;SUCCESSIVELY SEND NEXT CHARACTER TO LCD AND REPEAT THIS
185 DI:RET
```

main.asm

```
220
221 ; SUBROUTINE FOR DELAY OF 3 SECOND
222 DELAY3:MOV TMOD,#10H  ;INITIALIZE TIMER 1 IN MODE 1
223 MOV R3,#42   ;THE 71.1 ms *42 = 3sec delay
224 AGAIN1: MOV TL1,#00H   ; LOAD TL1 WITH 00H
225 MOV TH1,#00H  ; LOAD TH1 WITH 00H
226 SETB TR1  ;START TIMER 1
227 BACK1: JNB TF1,BACK1   ;WAIT UNTIL OVERFLOW FLAG OF TIMER 1 BECOME 1
228 CLR TR1   ;STOP THE TIMER 1
229 CLR TF1   ;CLEAR THE OVERFLOW FLAG
230 DJNZ R3,AGAIN1 ;LOOP AGAIN
231 RET
232
233 ;DELAY FOR 0.27 SECOND
234 DELAY2: MOV R3,#250 ; R3  = 250
235         MOV TMOD,#01H ;INITIALIZE TIMER 0 IN MODE 1
236 BACK2:  MOV TH0,#0FCH
237         MOV TL0,#018H  ;initial count value = FC18 is loaded into timer
238         SETB TR0 ;starting timer
239 HERE5:  JNB TF0,HERE5 ;monitor Timer flag if it is 1
240         CLR TR0 ; stop the timer
241         CLR TF0 ; reset the timer flag
242         DJNZ R3,BACK2 ; repeat this process 250 times
243         RET
244
245 ;SUBROUTINE TO PRODUCE 0.036SEC DELAY
246 DELAY:MOV R3,#65
247 HERE:MOV R4,#255
248 HERE2:DJNZ R4,HERE2
249 DJNZ R3,HERE
250 RET
```

main.asm

```
CLR TF1   ;CLEAR THE OVERFLOW FLAG
DJNZ R3,AGAIN1 ;LOOP AGAIN
RET

;DELAY FOR 0.27 SECOND
DELAY2: MOV R3,#250 ; R3  = 250
        MOV TMOD,#01H ;INITIALIZE TIMER 0 IN MODE 1
BACK2:  MOV TH0,#0FCH
        MOV TL0,#018H  ;initial count value = FC18 is loaded into timer
        SETB TR0 ;starting timer
HERE5:  JNB TF0,HERE5 ;monitor Timer flag if it is 1
        CLR TR0 ; stop the timer
        CLR TF0 ; reset the timer flag
        DJNZ R3,BACK2 ; repeat this process 250 times
        RET

;SUBROUTINE TO PRODUCE 0.036SEC DELAY
DELAY:MOV R3,#65
HERE:MOV R4,#255
HERE2:DJNZ R4,HERE2
DJNZ R3,HERE
RET

;TEXT TO BE DISPLAYED ON LCD
TEXT_S1: DB "ACCESS GRANTED",0
TEXT_S2: DB "OPENING DOOR",0
TEXT_FAIL: DB "WRONG PASSWORD",0
TEXT_S3: DB "CLOSING DOOR",0
TEXT3: DB "CHECKING PASS...",0
COMMAND: DB 38H,0FH,01H,06H,80H,0;INITIALIZE COMMAND OF LCD
INI_MSG: DB "ENTER 5-DIG PASS",0
PASSWORD: DB "A","0","B","C","*" ;CORRECT PASSWORD

END
```

ALGORITHM USED

We have stored the original password in ‘PASSWORD’ subroutine. We save the pressed key value to a memory address and whether entered key is equal to original password is checked by XOR gate i.e. XRL instruction and we set a condition if it is equal it will jump to other subroutine where it will show correct password on LCD and send high signal to motor drive for enabling the motor drive. If entered password is wrong it will jump to another sub-routine where it send commands to LCD to show ‘WRONG PASSWORD’.

ASSEMBLY CODE EXPLANATION:

The provided code is an assembly language program for a door lock using an 8051 micro-controller. Let's break down the explanation of assembly code:

1.Initialization:

Initialize the lcd by sending commands 38h,01h,06h,80h.. to lcd and displaying the initial text message on LCD i.e. ‘ENTER 5-DIGIT PASS’.

2. 4x4 keypad interface:

First make all pins of port 1 to high.

Send 0 to first row and check which key is pressed by checking the column pins. (if column 1 is pressed i.e. column pin will get 0)

Similarly we can checked for all pins to checked which pin is pressed.

The pressed key is then store to consecutively location starting from 2FH in RAM memory address.

4. password checking

- The store password is checked with the password entered by 4x4 keypad.
- This can be done by using XOR logic operation i.e. if the entered password is same as store password then XOR logic is 0 and password is correct else wrong.

4.Motor pins :

- If the password is correct micro controller will send high signal to motor drive enable pin to rotate and door will open.
- If password is wrong then motor will not rotate and door will not open.
- Jumps to the `DISPLAY_HOUR` routine.

5. Delay Function (`DELAY` routine):

- Initializes Timer 1 in Mode 1 for a delay.

- SET count to 0000h and start the timer. It will give us 71.1 ms delay. We make a loop for 42 to make it almost 3 second.

The program structure is a continuous checked which button is pressed in keypad. If any key is pressed we store this value to RAM memory address 2FH.

And send command to LCD to display ‘ENTER 5 DIGIT PASS’.

We set the password to be 5 digit so it will store the data to next memory address continuously. We have stored the original password to ‘password’ subroutine.

We used XOR gate to compare pressed button is equal to saved password . If pressed button are equal to saved password we send command to display correct password to LCD and send enable signal for 3 second to motor drive so motor rotate for 3 second.

PRACTICAL APPLICATION OF DOOR LOCK SYSTEM:

A password-based door lock system finds applications in various scenarios, including:

1. **Residential Security:** Protecting homes and apartments, allowing access only to authorized individuals with the correct password.
2. **Commercial Spaces:** Securing offices, meeting rooms, and storage areas, providing controlled access to employees and authorized personnel.

3. **Institutions:** Implementing secure access control in schools, universities, and hospitals to restrict entry to specific areas for safety and confidentiality purposes.
4. **Hospitality Industry:** Managing access to hotel rooms or rental properties, ensuring only guests with the correct password can enter.
5. **Industrial Facilities:** Enhancing security in factories, warehouses, and laboratories by limiting entry to authorized personnel.
6. **Shared Spaces:** Controlling access to shared spaces like gyms, community centers, or co-working spaces to ensure only paying members or authorized users can enter.
7. **Remote Access Control:** Allowing remote unlocking of doors via the internet or mobile apps, facilitating management and monitoring of access from a distance.

CONCLUSION:

Creating a door access system that utilizes password based system for both local and remote control offers convenience, flexibility, and enhanced security. Through a blend of hardware and software components, this system allows users to unlock or lock doors using passcodes entered via keypads, touchscreens, or remotely through a secure interface.

The local access functionality provides immediate control, allowing authorized individuals to input codes directly at the door for quick entry or secure closure. Meanwhile, the remote access feature extends this control beyond physical proximity, enabling authorized users to manage door access from anywhere using a mobile app or web interface.

THANK YOU