INSTR F241 : Microprocessor Programming and Interfacing

DESIGN ASSIGNMENT:
PROJECT-14

DOOR SECURITY CONTROL SYSTEM

GROUP NO : 98

MEET PARIKH	2015A8PS375P
MOHD INSHA KHAN	2015A8PS378P
ADITYA WAGH	2015A8PS382P
NIKHIL GOEL	2015A8PS385P

Acknowledgement

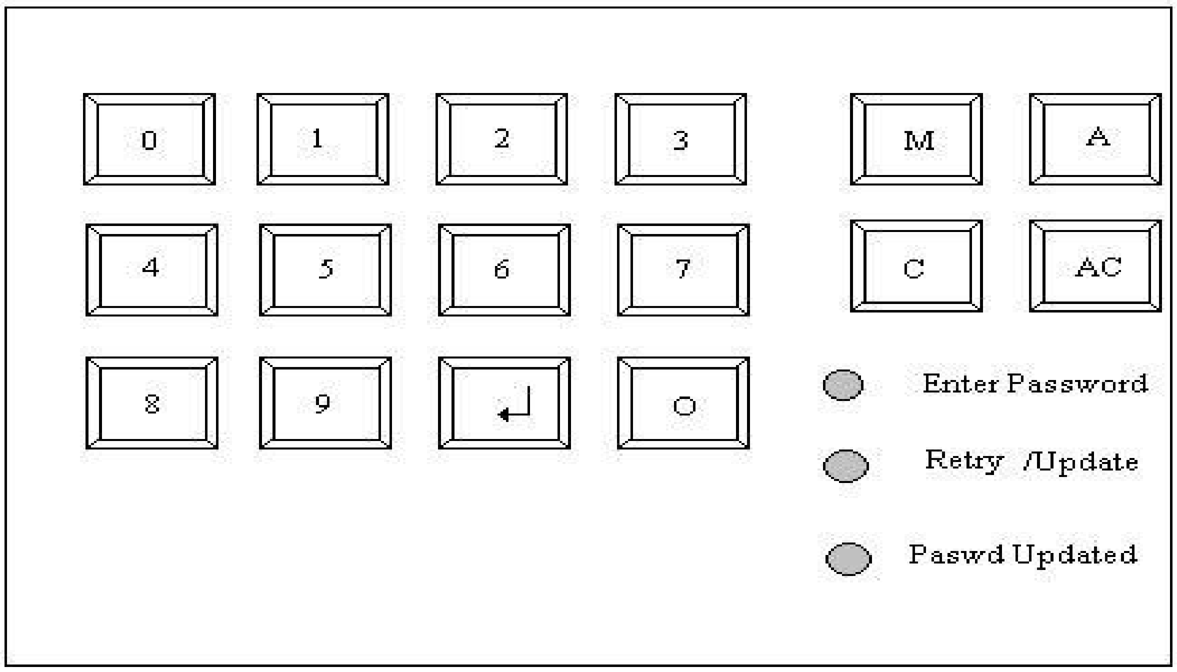
We, Meet Parikh, Mohd Insha Khan, Aditya Wagh and Nikhil Goel, thank Dr. Nitin Chaturvedi, Dr, Anupama and the entire MuP team for their impeccable documentation and thorough courseware, without which we could not have done this project. We would also like to thank our project mentors, and the instruction team for carefully and thoughtfully devising and executing the evaluation components, all of which helped strengthen our concepts and take this design to completion. This project has been an enjoying and enriching experience for our team.

Contents

PROBLEM STATEMENT

DESCRIPTION: This system controls the opening and closing of a door based on password entry. If the password is correct the person can enter. Each person is given two chances to enter the correct password. On failure an alarm is sounded. Inside the room a button is available when the button is pressed the door opens for 2 Min, so that the person can leave the room.

USER INTERFACE: There are three sets of passwords:
(1) User
(2) Master
(3) Alarm off



- The Master password is used by the security Personnel for updating Password of the day. Pressing the M button activates this mode. The system glows Enter Password LED asking the personnel to enter the password. The master password is a 16-digit value. The master is given only a single chance to enter the password. If authenticated, the retry/Update LED glows. If there is a failure in authentication the alarm is sounded. When the retry/ Update LED glows the user has to enter password of the day. This is 12-digit value. Once this value has been accepted by the system the Passwd Updated LED glows.
- User has to press the O key when he wants to enter the room. The Enter Password LED prompts the user to enter the password. The user is given C/AC option as well. If the first attempt fails, the RETRY LED glows. The user is allowed to re-enter password, on authentication door opens for a period of 1 Min. On Failure an ALARM is sounded.
- To Turn-off the Alarm the A button has to be pressed. Enter Password LED glows prompting user to enter the 14-digit password for turning of alarm, no retries are allowed. If authentication is successful then the alarm is turned off.
- To leave the room a button is available inside the room, when the button is pressed the door opens for 1 Minute so that the person can leave the room.
- LCD show the entry as asterisk when the password characters are entered.

ASSUMPTIONS

- All operations are sequential and so, no two operations can be carried out at the same time.

- User cannot access the door lock system at the 24 hour mark. First, a user password needs to be set using the master mode and only then can the door lock be accessed.

- Once you have entered a mode, it is necessary to complete the entire procedure. Pressing any other mode buttons like M, O or A will not override the mode.

- In case of mistake in typing the password to turn down the alarm, the system will get locked in it's last state and to use it again, it is necessary to shutdown and restart the system.

- The 24 hour clock starts running as soon as we turn on the system.

- The Master Password and the Alarm Password have been hardcoded in the system.

  Master Password: 9999999999999999
  Alarm Password: 99999999999999

<u>LIST OF COMPONENTS USED</u>

Sr. No.	Hardware Device	Description	Quantity
1.	8086	16 bit Microprocessor	1
2.	74LS373	Octal Latch	3
3.	74LS245	Transceiver	2
4.	7432	OR gate	6
5.	NOT	NOT gate	6
6.	2732	ROM	2
7.	6116	RAM	2
8.	74LS138	3-to-8 Decoder	1
9.	8255A	Programmable Peripheral Interface	2
10.	8253A	Programmable Interval Timer	2
11.	KEYPAD-PROJECT	Hex-keypad	1
12.	BUTTON	Button	1
13.	LM016L	16x2 LCD display	1
14.	BUZZER	Alarm	1
15.	LED-BLUE	Blue LED	1
16.	LED-RED	Red LED	3
17.	MOTOR-STEPPER	Stepper Motor	1

18.	OMIH-SH-105D	Relay	1

19.	SW-SPDT-MOM	SPDT Switch	1
20.	ULN2003A	Darlington Transistor	1
21.	74LS241	Tri-state Buffer	1

Memory Mapping

Size of 2732 (ROM) : 4k

ROM(even) : 00000h – 01FFEh (A0 = 0)
ROM (odd) : 00001h – 01FFFh (A0 = 1)

A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	0	0	X	X	X	X	X	X	X	X	X	X	X	X	X

Size of 6116 (RAM) : 2k

RAM(even) : FF000h – FFFFEh (A0 = 0)
RAM(odd) : FF001h – FFFFFh (A0 = 1)

A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
1	1	1	1	X	X	X	X	X	X	X	X	X	X	X	X

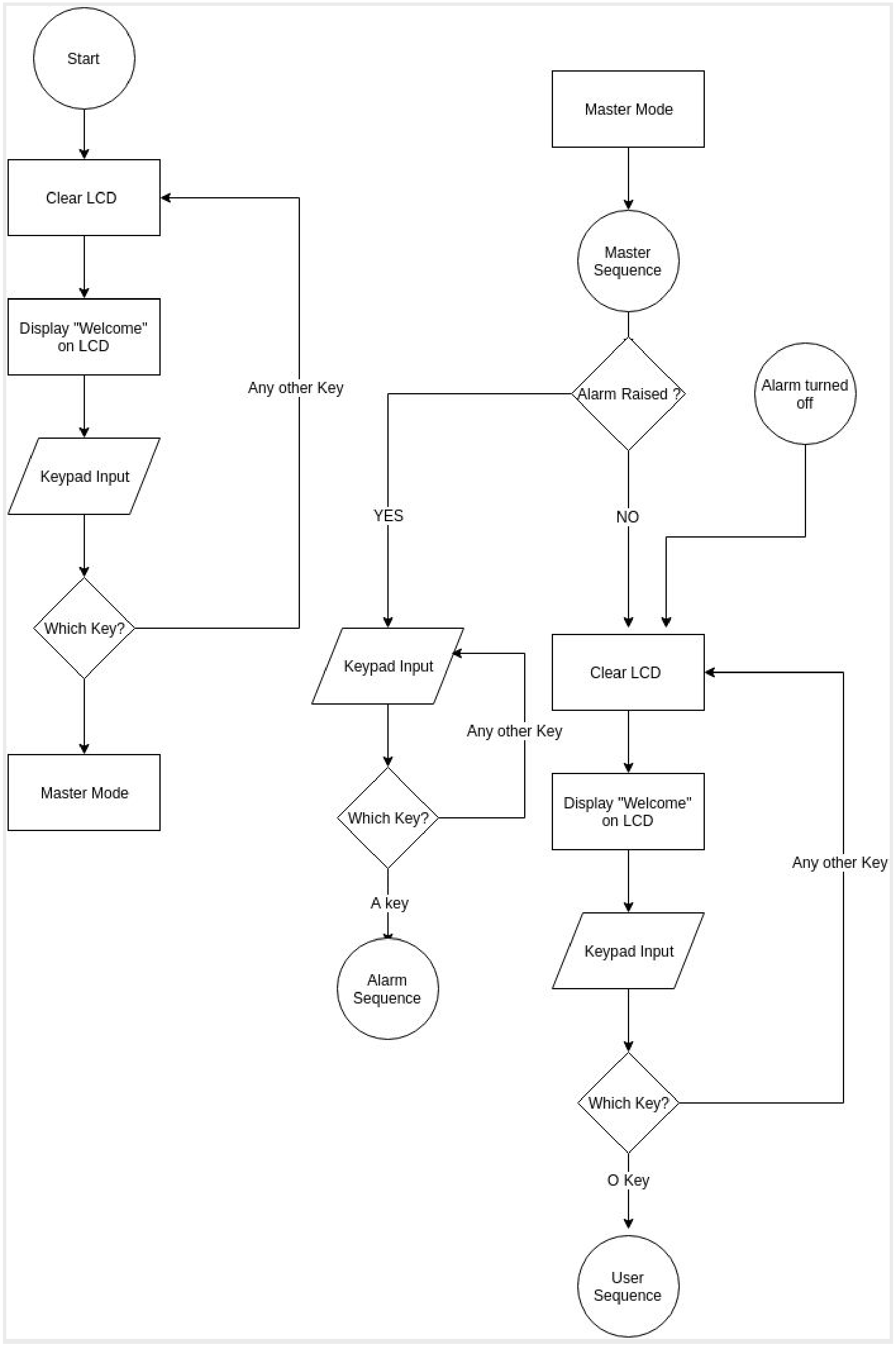
I/O Mapping

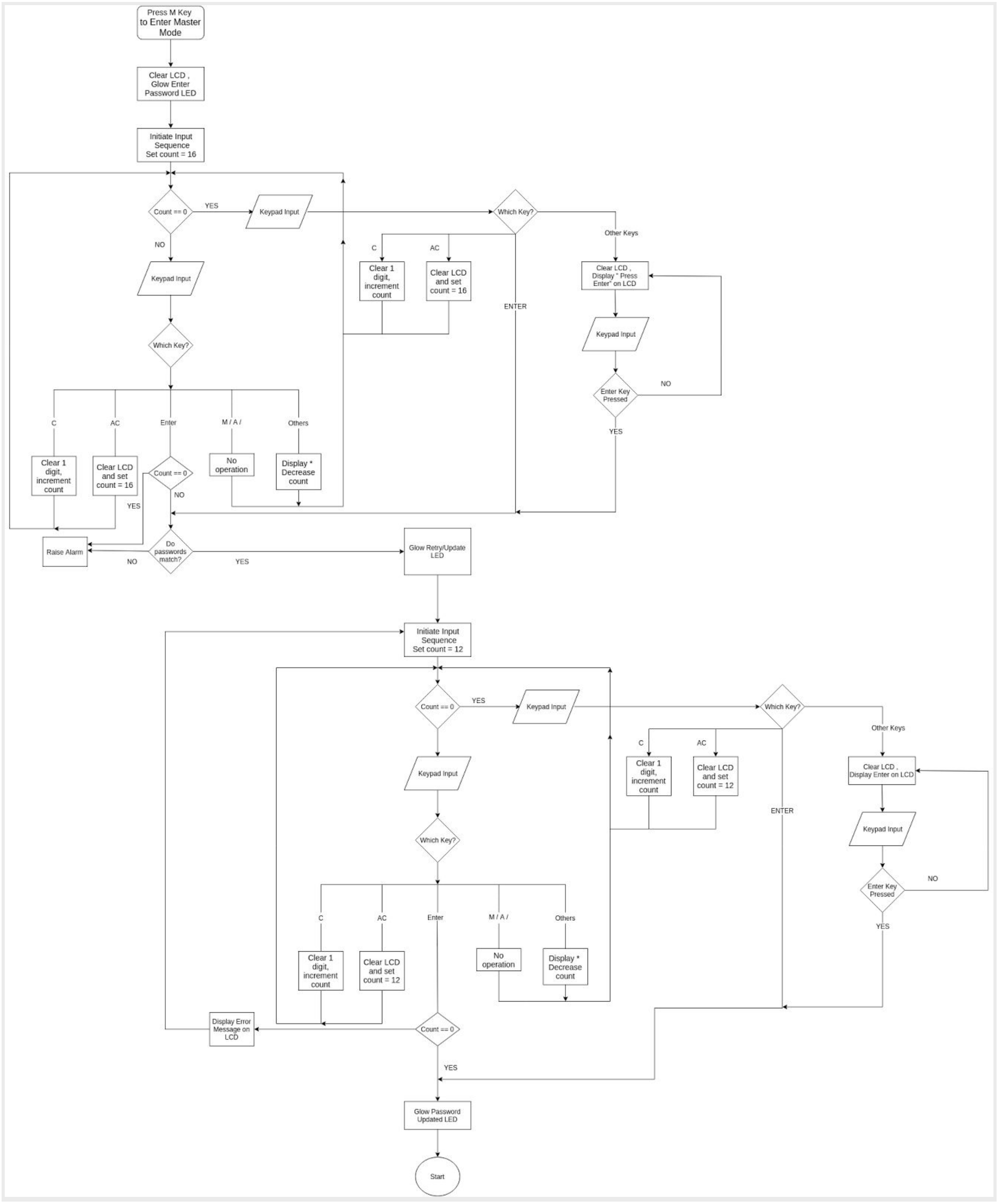
Address of 8255-1 port-A	: 00h
Address of 8255-1 port-B	: 02h
Address of 8255-1 port-C	: 04h
Address of 8255-1 control register	: 06h
Control word of 8255-1	: 88h
Address of 8255-2 port-A	: 08h
Address of 8255-2 port-B	: 0Ah
Address of 8255-2 port-C	: 0Ch
Address of 8255-2 control register	: 0Eh
Control word of 8255-2	: 89h
Address of 8253-1 count0	: 10h
Address of 8253-1 count1	: 12h
Address of 8253-1 count2	: 14h
Address of 8253-1 control register	: 16h
Control word of 8253-1 count0	: 36h
Control word of 8253-1 count1	: 56h
Control word of 8253-1 count2	: 92h
Address of 8253-2 count0	: 18h
Address of 8253-2 count1	: 1Ah
Address of 8253-2 count2	: 1Ch
Address of 8253-2 control register	: 1Eh
Control word of 8253-2 count0	: 34h
Control word of 8253-2 count1	: 5Ah
Control word of 8253-2 count2	: 94h

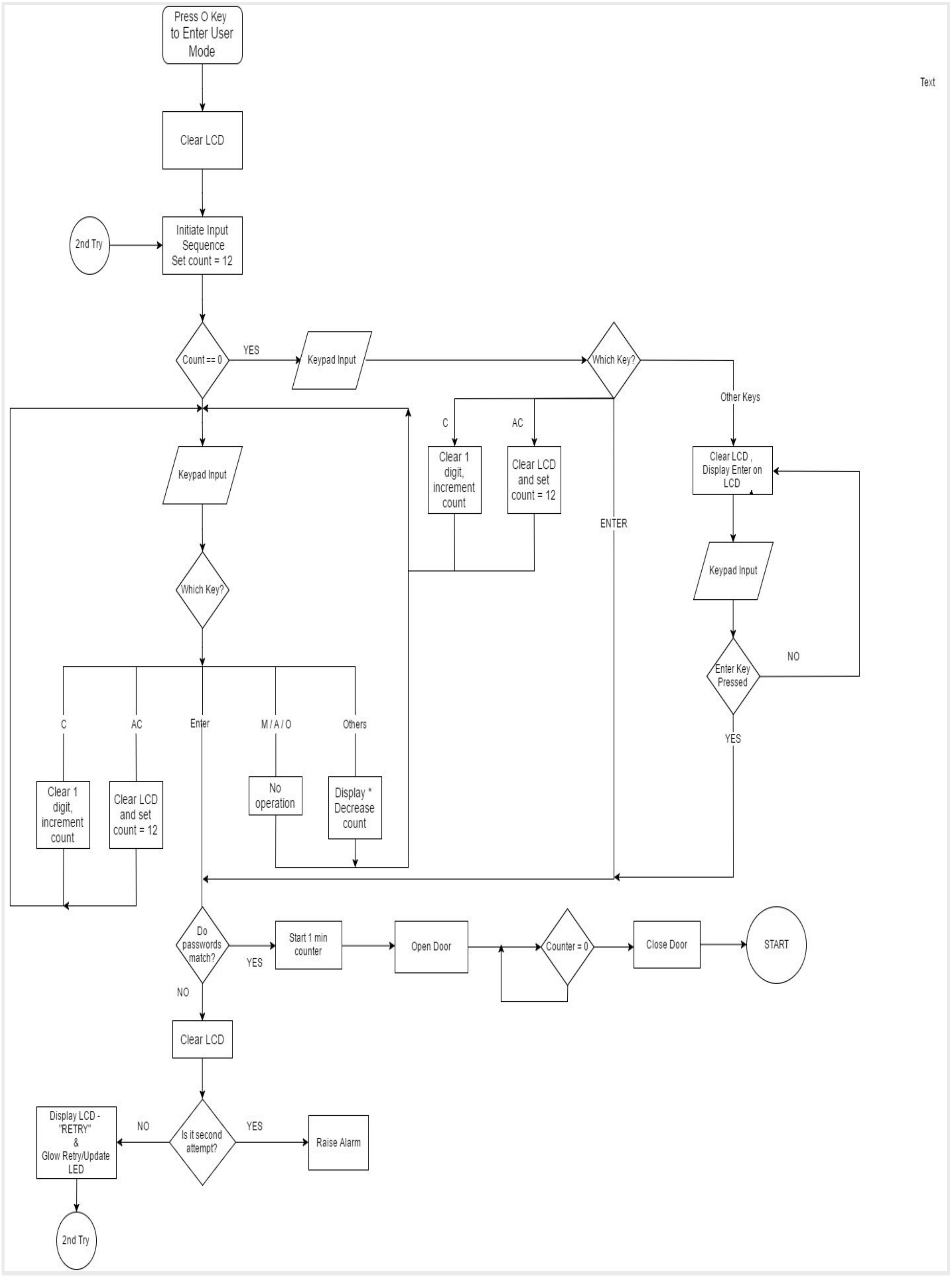
IVT

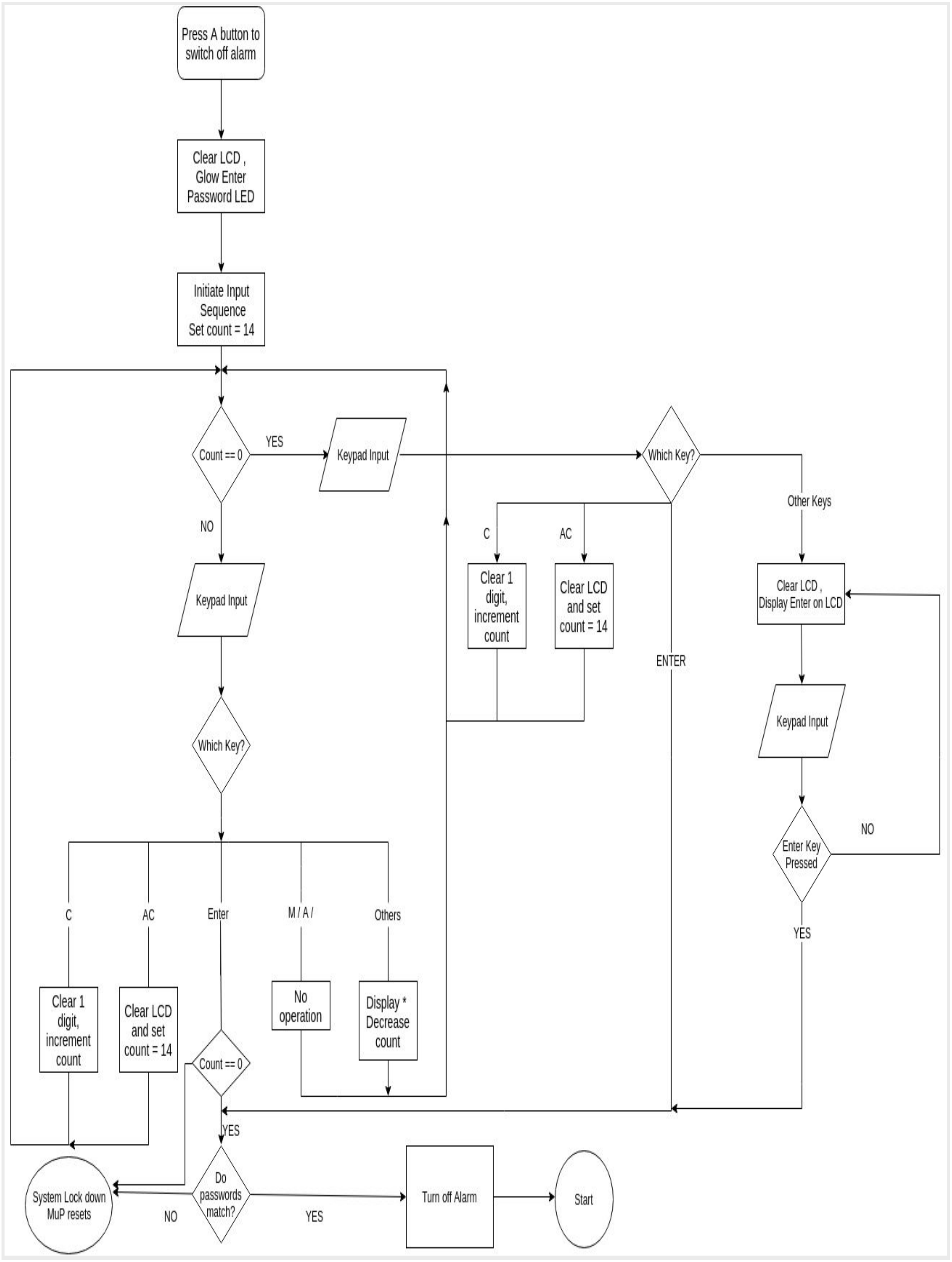
Interrupt Vector No.	Physical Address	Contains
INT 02h	00008h 0000Ah	IP2 CS2
INT 80h	00200h 00202h	IP128 CS128

Flow-Chart of the System




Press M Key to Enter Master Mode
Clear LCD , Glow Enter Password LED
Initiate Input Sequence Set count = 16
Count == 0
YES
NO
Keypad Input
Which Key?
C
AC
Enter
M / A /
Others
Clear 1 digit, increment count
Clear LCD and set count = 16
No operation
Display * Decrease count
ENTER
Other Keys
Clear LCD , Display " Press Enter" on LCD
Enter Key Pressed
Raise Alarm
Do passwords match?
Glow Retry/Update LED
Initiate Input Sequence Set count = 12
Clear LCD and set count = 12
Clear LCD , Display Enter on LCD
Display Error Message on LCD
Glow Password Updated LED
Start


Press O Key to Enter User Mode
Clear LCD
2nd Try
Initiate Input Sequence Set count = 12
Count == 0
YES
Keypad Input
Which Key?
C
AC
Clear 1 digit, increment count
Clear LCD and set count = 12
ENTER
Other Keys
Clear LCD , Display Enter on LCD
Keypad Input
Enter Key Pressed
NO
YES
Keypad Input
Which Key?
C
AC
Enter
M / A / O
Others
Clear 1 digit, increment count
Clear LCD and set count = 12
No operation
Display * Decrease count
Do passwords match?
YES
NO
Start 1 min counter
Open Door
Counter = 0
Close Door
START
Clear LCD
Is it second attempt?
NO
YES
Display LCD - "RETRY" & Glow Retry/Update LED
Raise Alarm
2nd Try
Text


Press A button to switch off alarm
Clear LCD , Glow Enter Password LED
Initiate Input Sequence Set count = 14
Count == 0
YES
NO
Keypad Input
Which Key?
C
AC
Enter
M / A /
Others
Clear 1 digit, increment count
Clear LCD and set count = 14
No operation
Display * Decrease count
Count == 0
YES
Do passwords match?
NO
YES
System Lock down MuP resets
Turn off Alarm
Start
Keypad Input
Which Key?
C
AC
ENTER
Other Keys
Clear 1 digit, increment count
Clear LCD and set count = 14
Clear LCD , Display Enter on LCD
Keypad Input
Enter Key Pressed
NO
YES

ALP Code

```
#make_bin#

#LOAD_SEGMENT=FFFFh#
#LOAD_OFFSET=0000h#

#CS=0000h#
#IP=0000h#

#DS=0000h#
#ES=0000h#

#SS=0000h#
#SP=FFFEh#

#AX=0000h#
#BX=0000h#
#CX=0000h#
#DX=0000h#
#SI=0000h#
#DI=0000h#
#BP=0000h#



        jmp     st1
        db     5 dup(0)

        ;IVT entry for NMI (INT 02h)
        dw     Nmi_24hrtimer
        dw     0000
        db     500 dup(0)

       ;IVT entry for 80H
        dw     Switch_intR
        dw     0000
        db     508 dup(0)
       st1:      cli

       ;intialize ds, es,ss to start of RAM
        mov       ax,0200h
        mov       ds,ax
        mov       es,ax
        mov       ss,ax
        mov       sp,0FFFEH
       ; INITIALIZATION OF 8255,8253 BEGINS HERE

       sti
       mov al,89h       ; control word for 8255-2
       out 0Eh,al

       mov al,88h       ; control word for 8255-1
       out 06h,al

       mov al,36h       ;control word for 8253-1 counter 0
       out 16h,al

       mov al,56h       ;control word for 8253-1 counter 1
       out 16h,al

       mov al,92h       ;control word for 8253-1 counter 2
       out 16h,al

       mov al,34h       ;control word for 8253-2 counter 0
```

```
        out 1eh,al

        mov al,5ah      ;control word for 8253-2 counter 1
        out 1eh,al

        mov al,94h      ;control word for 8253-2 counter 2
        out 1eh,al

        mov al,50h      ;load count lsb for 8253-1 counter 0
        out 10h,al

        mov al,0C3h     ;load count msb for 8253-1 counter 0
        out 10h,al

        mov al,64h      ;load count for 8253-1 counter 1
        out 12h,al

        mov al,5h ;load count lsb for 8253-1 counter 2 (1 minute
Timer)
        out 14h,al


        mov al,40 ;load count for 8253-2  LSB counter 0 (24 hour
counter)
        out 18h,al

        mov al,0  ;load count for 8253-2  MSB counter 0 (24 hour
counter)
        out 18h,al

        mov al,3  ;load count for 8253-2 counter 1 (Switch trigger
counter)
        out 1ah,al
```

```
mov al,2   ;load count for 8253-2 counter 2
out 1ch,al
;INITIALIZATION OF 8255,8253 ENDS HERE


mov al,00h       ;default low output from 8255-2 upper port C
out 0ch,al

call DELAY_20ms ;LCD INITIALIZATION BEGINS
mov al,04h
out 02h,al
call DELAY_20ms
mov al,00h
out 02h,al

mov al,38h
out 00h,al

mov al,04h
out 02h,al
call DELAY_20ms
mov al,00h
out 02h,al
call DELAY_20ms
mov al,0Ch
out 00h,al
mov al,04h
out 02h,al
call DELAY_20ms
mov al,00h
out 02h,al
```

```
        mov al,06h
        out 00h,al
        call DELAY_20ms
        mov al,04h
        out 02h,al
        call DELAY_20ms
        mov al,00h
        out 02h,al
        mov al,4ch
        out 00h,al
        call DELAY_20ms  ;LCD INITIALIZATION ENDS

        mov ax,0200h
        mov ds,ax

        mov si,0000h
        mov al,0bdh      ;hard coding pass-word
;9999999999999999
        mov [si],al

        mov al,0bdh
        mov [si+1],al

        mov al,0bdh
        mov [si+2],al

        mov al,0bdh
        mov [si+3],al

        mov al,0bdh
        mov [si+4],al

        mov al,0bdh
mov [si+5],al

mov al,0bdh
mov [si+6],al

mov al,0bdh
mov [si+7],al

mov al,0bdh
mov [si+8],al

mov al,0bdh
mov [si+9],al

mov al,0bdh
mov [si+0ah],al

mov al,0bdh
mov [si+0bh],al

mov al,0bdh
mov [si+0ch],al

mov al,0bdh
mov [si+0dh],al

mov al,0bdh
mov [si+0eh],al

mov al,0bdh
mov [si+0fh],al

add si,000fh
```

```
        inc si


        mov al,0bdh      ;hard coding alarm pass-word ;
99999999999999
        mov [si],al

        mov al,0bdh
        mov [si+1],al

        mov al,0bdh
        mov [si+2],al

        mov al,0bdh
        mov [si+3],al

        mov al,0bdh
        mov [si+4],al

        mov al,0bdh
        mov [si+5],al

        mov al,0bdh
        mov [si+6],al

        mov al,0bdh
        mov [si+7],al

        mov al,0bdh
        mov [si+8],al

        mov al,0bdh
        mov [si+9],al
        mov al,0bdh
        mov [si+0ah],al

        mov al,0bdh
        mov [si+0bh],al

        mov al,0bdh
        mov [si+0ch],al

        mov al,0bdh
        mov [si+0dh],al

        add si,000dh
        inc si

        mov al,0ffh
        out 08h,al

        start:      call clear_LCD
            call welcome_msg

            mov bp,00h
            call keypad_input
            cmp al,0bbh
            jz master_mode
            jmp start

x6:     call clear_LCD
       call welcome_msg
       call keypad_input
        cmp al,0b7h
        jz User_mode
        jmp x6 ;press valid key

master_mode:
  call intm
  mov bp,0abcdh
  cmp ax,0abcdh
  jnz x6
x8:     call keypad_input
        cmp al,7Dh
        jz Alarm_mode
        jnz x8

Alarm_mode:
  call inta
  cmp dh,6h
  jz start
  cmp dh,1h
  jz x6
  jmp x70

User_mode:
  call intu
  cmp ax,0abcdh
  jz x8
  jnz x6

x70:
stop: jmp stop

DELAY_20ms proc

        MOV        CH,5
        X4:   NOP
```

```
                NOP
                DEC  CH
                JNZ  X4
        RET
DELAY_20ms endp
DELAY_0.04s proc

        MOV cx,4fffh
        X17:  NOP
                NOP
                DEC  cx
                JNZ  X17
        RET
DELAY_0.04s endp
DELAY_max proc

        MOV cx,0ffffh
        X16:  NOP
                NOP
                DEC  cx
                JNZ  X16
        RET
DELAY_max endp

enter_LCD proc
        mov al,0A0h
        out 00h,al
        call DELAY_20ms
        mov al,05h
        out 02h,al
        call DELAY_20ms
        mov al,01h
        out 02h,al  ;prints Space
```

```
mov al,0A0h
out 00h,al
call DELAY_20ms
mov al,05h
out 02h,al
call DELAY_20ms
mov al,01h
out 02h,al  ;prints Space

mov al,50h
out 00h,al
call DELAY_20ms
mov al,05h
out 02h,al
call DELAY_20ms
mov al,01h
out 02h,al  ;prints P

mov al,52h
out 00h,al
call DELAY_20ms
mov al,05h
out 02h,al
call DELAY_20ms
mov al,01h
out 02h,al  ;prints R

mov al,45h
out 00h,al
call DELAY_20ms
mov al,05h
out 02h,al
```

```
call DELAY_20ms
mov al,01h
out 02h,al  ;prints E

mov al,53h
out 00h,al
call DELAY_20ms
mov al,05h
out 02h,al
call DELAY_20ms
mov al,01h
out 02h,al  ;prints S

mov al,53h
out 00h,al
call DELAY_20ms
mov al,05h
out 02h,al
call DELAY_20ms
mov al,01h
out 02h,al  ;prints S

mov al,0A0h
out 00h,al
call DELAY_20ms
mov al,05h
out 02h,al
call DELAY_20ms
mov al,01h
out 02h,al  ;prints Space

mov al,45h
out 00h,al
```

```
call DELAY_20ms
mov al,05h
out 02h,al
call DELAY_20ms
mov al,01h
out 02h,al  ;prints E

mov al,4Eh
out 00h,al
call DELAY_20ms
mov al,05h
out 02h,al
call DELAY_20ms
mov al,01h
out 02h,al  ;prints N

mov al,54h
out 00h,al
call DELAY_20ms
mov al,05h
out 02h,al
call DELAY_20ms
mov al,01h
out 02h,al  ;prints T

mov al,45h
out 00h,al
call DELAY_20ms
mov al,05h
out 02h,al
call DELAY_20ms
mov al,01h
out 02h,al  ;prints E
```

```
        mov al,52h
        out 00h,al
        call DELAY_20ms
        mov al,05h
        out 02h,al
        call DELAY_20ms
        mov al,01h
        out 02h,al  ;prints R
RET
enter_LCD endp


intm proc

                call clear_LCD
                mov al,0feh
                out 08h,al        ;turns on enter password LED
                mov cx,16

enter_16bit:
                call keypad_input
                cmp al,7eh
                jz pressc
                cmp al,7bh
                jz pressac
                cmp al,77h
                jz press_enter
                cmp al,0bbh
                jz nop_master
                cmp al,0b7h
                jz nop_master
                cmp al,7dh
                jz nop_master
                mov [si],al
                CALL Print_*
                inc si
                dec cx
                jnz enter_16bit
disp_entermaster:
                call keypad_input
                cmp al,7eh
                jz pressc
                cmp al,7bh
                jz pressac
                cmp al,77h
                jz press_enter
asd:            CALL clear_LCD
                CALL enter_LCD
                call keypad_input
                cmp al,77h
                jz press_enter
                jnz asd
nop_master:     nop
                jmp enter_16bit
pressc:         call clear_1digit_LCD
                dec si
                inc cx
                jmp enter_16bit
pressac:
                CALL clear_LCD
                mov cx,16
                mov si,1eh              ;start of pass segment
                jmp enter_16bit
press_enter:
                CALL clear_LCD
```

```
          mov al,0ffh
          out 08h,al
          cmp cx,0
          jz cmp_pass
          jmp raise_alarm


day_pass:
          mov si,002Eh
          mov al,0fdh
          out 08h,al
          call DELAY_max
          call DELAY_max
          call DELAY_max
          call clear_LCD
          mov cx,12
enter_12bit:
          call keypad_input
          cmp al,7eh
          jz presscday
          cmp al,0bbh
          jz nop_day
          cmp al,0b7h
          jz nop_day
          cmp al,7dh
          jz nop_day
          cmp al,7bh
          jz pressacday
          cmp al,77h
          jz press_enterday
          mov [si],al
          CALL Print_*
          inc si
```

```
            dec cx
            jnz enter_12bit
disp_enter:
            call keypad_input
            cmp al,7eh
            jz presscday
            cmp al,7bh
            jz pressacday
            cmp al,77h
            jz press_enterday
asd1:            CALL clear_LCD
                 CALL enter_LCD
                 call keypad_input
                 cmp al,77h
                 jz press_enterday
                 jnz asd1
nop_day: nop
            jmp enter_12bit

presscday:
            call clear_1digit_LCD
            dec si
            inc cx
            jmp enter_12bit
pressacday:
            CALL clear_LCD
            jmp day_pass
press_enterday:
            CALL clear_LCD
            mov al,0ffh
            out 08h,al
            cmp cx,0
            jnz err_msg
            mov al,0fbh
            out 08h,al

            call DELAY_max
            call DELAY_max


            mov al,0ffh
            out 08h,al
            jz end_69h
err_msg:
            call error_msg
            jmp day_pass
cmp_pass:
            cld
            mov si,0000h
            mov di,001Eh
            mov cx,17
x5:         mov al,[si]
            mov bl,[di]
            dec cx
            jz day_pass
            cmp al,bl
            jnz raise_alarm
            inc si
            inc di
            jmp x5


raise_alarm:
            mov dh,5h
            mov al,0fh
            out 08h,al
```

```
                mov ax,0abcdh
end_69h:
ret
intm endp

Print_* proc
                mov al,2Ah
                out 00h,al
                call DELAY_20ms
                mov al,05h
                out 02h,al
                call DELAY_20ms
                mov al,01h
                out 02h,al  ;prints *
ret
Print_* endp

clear_LCD proc
        mov al,00h
        out 02h,al
        call DELAY_20ms
        mov al,01h                      ;Clear Display
        out 00h,al
        call DELAY_20ms
        mov al,04h
        out 02h,al
        call DELAY_20ms
        mov al,00h
        out 02h,al
RET
clear_LCD endp
```

```
keypad_input proc                ;SubR for keypad entry,al has
unique key input value.
x0:         mov al,00h
            out 04h,al
x1:         in al,04h
            and al,0f0h
            cmp al,0f0h
            jnz x1
            CALL DELAY_20ms

            mov al,00h                        ; Check for key press
            out 04,al
x2:
            in al,04h
            and al,0F0h
            cmp al,0F0h
            jz x2
            CALL DELAY_20ms

            mov al,00h                        ; Check for key press
            out 04,al
            in al,04h
            and al,0F0h
            cmp al,0F0h
            jz x2

            mov al,0Eh                        ;Check for key press
column 1
            mov bl,al
            out 04h,al
            in al,04h
            and al,0f0h
            cmp al,0f0h
```

```
          jnz x3

          mov al,0Dh                    ;Check for key press
column 2
          mov bl,al
          out 04h,al
          in al,04h
          and al,0f0h
          cmp al,0f0h
          jnz x3

          mov al,0Bh                    ;Check for key press
column 3
          mov bl,al
          out 04h,al
          in al,04h
          and al,0f0h
          cmp al,0f0h
          jnz x3

          mov al,07h                    ;Check for key press
column 4
          mov bl,al
          out 04h,al
          in al,04h
          and al,0f0h
          cmp al,0f0h
          jz x2

x3:       or al,bl
ret
keypad_input endp
```

```
inta proc
        mov al,00eh
        out 08h,al




        mov cx,14
        mov si,3ah                                      ;store the 16-bit entered
pass after the hard coded pass word
enter_14bit:
                call keypad_input
                cmp al,7eh
                jz pressc_alarm
                cmp al,0bbh
                jz nop_alarm
                cmp al,0b7h
                jz nop_alarm
                cmp al,7dh
                jz nop_alarm
                cmp al,7bh
                jz pressac_alarm
                cmp al,77h
                jz press_enter_alarm
                mov [si],al
                CALL Print_*
                inc si
                dec cx
                jnz enter_14bit
disp_enteralarm:
                call keypad_input
                cmp al,7eh
                jz pressc_alarm
                cmp al,7bh
```

```
            jz pressac_alarm
            cmp al,77h
            jz press_enter_alarm
asd2:            CALL clear_LCD
            CALL enter_LCD
            call keypad_input
            cmp al,77h
            jz press_enter_alarm
            jnz asd2
nop_alarm: nop
       jmp enter_14bit
pressc_alarm:
            call clear_1digit_LCD
            dec si
            inc cx
            jmp enter_14bit
pressac_alarm:
            call clear_LCD
            mov cx,14
            mov si,3ah                    ;start of pass segment
            jmp enter_14bit
press_enter_alarm:
            CALL clear_LCD
            mov al,0fh
            out 08h,al
            cmp cx,0
            jz cmp_pass_alarm
            jnz x56
cmp_pass_alarm:
            cld
            mov si,10h
            mov di,3ah
            mov cx,14
            repe cmpsb
            cmp cx,00h
            jnz x56
            mov al,0ffh
            out 08h,al
            add dh,1h
x56:
ret
inta endp



intu proc
            call clear_LCD
            mov dl,1                    ;flag for checking two inputs

            mov al,0feh
            out 08h,al
            mov cx,12
            mov si,48h                        ;store the 12-bit entered
pass after the hard coded pass word
enter_12bitu:
            call keypad_input
            cmp al,7eh
            jz pressc_user
            cmp al,7bh
            jz pressac_user
            cmp al,0bbh
            jz nop_user
            cmp al,0b7h
            jz nop_user
            cmp al,7dh
            jz nop_user
```

```
          cmp al,77h
          jz press_enter_user
          mov [si],al
          CALL Print_*
          inc si
          dec cx
          jnz enter_12bitu
disp_enter_user:
          call keypad_input
          cmp al,7eh
          jz pressc_user
          cmp al,7bh
          jz pressac_user
          cmp al,77h
          jz press_enter_user
asd3:          CALL clear_LCD
          CALL enter_LCD
          call keypad_input
          cmp al,77h
          jz press_enter_user
          jnz asd3
nop_user:
               nop
               jmp enter_12bitu
pressc_user:
          call clear_1digit_LCD
          dec si
          inc cx
          jmp enter_12bitu
pressac_user:
          call clear_LCD
          mov cx,12
          mov si,48h                    ;start of pass segment
```

```
                jmp enter_12bitu
press_enter_user:
                mov al,0ffh
                out 08h,al
                cmp cx,0
                jz cmp_pass_user
                jnz wrong_pass

cmp_pass_user:
                cld
                mov si,2eh
                mov di,48h
                mov cx,12
                repe cmpsb
                cmp cx,00h
                jnz wrong_pass
                jz open_door_user

wrong_pass :
                call clear_LCD
                mov si,48h
                mov cx,12
                cmp dl,0
                jz raise_alarm_user
                mov al,0fdh
                out 08h,al
                call retry_msg
                call DELAY_max
                call DELAY_max
                call clear_LCD
                mov cx,12
                dec dl
                jmp enter_12bitu
```

```
raise_alarm_user:
                mov dh,0
                mov al,0fh
                out 08h,al
                mov ax,0abcdh
                jmp end_70h
open_door_user:
                call open_door
end_70h:
ret
intu endp

ints proc

                call open_door

                ; CALL DELAY_0.04s
                ; mov al,00h
                ; out 0ch,al
ret
ints endp

open_door proc
        call clear_LCD
        mov al,8ah
        out 0Ah,al

        call DELAY_20ms

        mov al,0ah
        out 0Ah,al

x31:            in al,0ch
```

```
                    cmp al,0ffh
                    jnz x31
                    call DELAY_20ms
                    call close_door
ret
open_door endp

close_door proc
          mov al,03h
          out 0Ah,al
          call DELAY_max
          call DELAY_max
          call DELAY_max
          call DELAY_max
          call DELAY_max
          call DELAY_max
          call DELAY_max
          call DELAY_max
          call DELAY_max
          call DELAY_max
          call DELAY_max
          call DELAY_max
          call DELAY_max
          call DELAY_max
          call DELAY_max
          call DELAY_max
          call DELAY_max
          call DELAY_max
          call DELAY_max
          call DELAY_max


ret
```

```
close_door endp


welcome_msg proc
        mov al,0A0h
        out 00h,al
        call DELAY_20ms
        mov al,05h
        out 02h,al
        call DELAY_20ms
        mov al,01h
        out 02h,al  ;prints Space

        mov al,0A0h
        out 00h,al
        call DELAY_20ms
        mov al,05h
        out 02h,al
        call DELAY_20ms
        mov al,01h
        out 02h,al  ;prints Space

        mov al,0A0h
        out 00h,al
        call DELAY_20ms
        mov al,05h
        out 02h,al
        call DELAY_20ms
        mov al,01h
        out 02h,al  ;prints Space

        mov al,0A0h
        out 00h,al
```

```
call DELAY_20ms
mov al,05h
out 02h,al
call DELAY_20ms
mov al,01h
out 02h,al  ;prints Space

mov al,0A0h
out 00h,al
call DELAY_20ms
mov al,05h
out 02h,al
call DELAY_20ms
mov al,01h
out 02h,al  ;prints Space

mov al,57h
out 00h,al
call DELAY_20ms
mov al,05h
out 02h,al
call DELAY_20ms
mov al,01h
out 02h,al  ;prints W

mov al,45h
out 00h,al
call DELAY_20ms
mov al,05h
out 02h,al
call DELAY_20ms
mov al,01h
out 02h,al  ;prints E
```

```
mov al,4Ch
out 00h,al
call DELAY_20ms
mov al,05h
out 02h,al
call DELAY_20ms
mov al,01h
out 02h,al  ;prints L

mov al,43h
out 00h,al
call DELAY_20ms
mov al,05h
out 02h,al
call DELAY_20ms
mov al,01h
out 02h,al  ;prints C

mov al,4Fh
out 00h,al
call DELAY_20ms
mov al,05h
out 02h,al
call DELAY_20ms
mov al,01h
out 02h,al  ;prints O

mov al,4dh
out 00h,al
call DELAY_20ms
mov al,05h
out 02h,al
```

```
        call DELAY_20ms
        mov al,01h
        out 02h,al  ;prints M

        mov al,45h
        out 00h,al
        call DELAY_20ms
        mov al,05h
        out 02h,al
        call DELAY_20ms
        mov al,01h
        out 02h,al  ;prints E

ret
welcome_msg endp

update_msg proc
        mov al,55h
        out 00h,al
        call DELAY_20ms
        mov al,05h
        out 02h,al
        call DELAY_20ms
        mov al,01h
        out 02h,al  ;prints U

        mov al,50h
        out 00h,al
        call DELAY_20ms
        mov al,05h
        out 02h,al
        call DELAY_20ms
        mov al,01h
```

```
out 02h,al  ;prints P

mov al,44h
out 00h,al
call DELAY_20ms
mov al,05h
out 02h,al
call DELAY_20ms
mov al,01h
out 02h,al  ;prints D

mov al,41h
out 00h,al
call DELAY_20ms
mov al,05h
out 02h,al
call DELAY_20ms
mov al,01h
out 02h,al  ;prints A

mov al,54h
out 00h,al
call DELAY_20ms
mov al,05h
out 02h,al
call DELAY_20ms
mov al,01h
out 02h,al  ;prints T

mov al,45h
out 00h,al
call DELAY_20ms
mov al,05h
```

```
out 02h,al
call DELAY_20ms
mov al,01h
out 02h,al  ;prints E

mov al,0A0h
out 00h,al
call DELAY_20ms
mov al,05h
out 02h,al
call DELAY_20ms
mov al,01h
out 02h,al  ;prints Space

mov al,50h
out 00h,al
call DELAY_20ms
mov al,05h
out 02h,al
call DELAY_20ms
mov al,01h
out 02h,al  ;prints P

mov al,41h
out 00h,al
call DELAY_20ms
mov al,05h
out 02h,al
call DELAY_20ms
mov al,01h
out 02h,al  ;prints A

mov al,53h
```

```
out 00h,al
call DELAY_20ms
mov al,05h
out 02h,al
call DELAY_20ms
mov al,01h
out 02h,al  ;prints S

mov al,53h
out 00h,al
call DELAY_20ms
mov al,05h
out 02h,al
call DELAY_20ms
mov al,01h
out 02h,al  ;prints S

mov al,57h
out 00h,al
call DELAY_20ms
mov al,05h
out 02h,al
call DELAY_20ms
mov al,01h
out 02h,al  ;prints W

mov al,4Fh
out 00h,al
call DELAY_20ms
mov al,05h
out 02h,al
call DELAY_20ms
mov al,01h
```

```
        out 02h,al  ;prints O

        mov al,52h
        out 00h,al
        call DELAY_20ms
        mov al,05h
        out 02h,al
        call DELAY_20ms
        mov al,01h
        out 02h,al  ;prints R

        mov al,44h
        out 00h,al
        call DELAY_20ms
        mov al,05h
        out 02h,al
        call DELAY_20ms
        mov al,01h
        out 02h,al  ;prints D

ret
update_msg endp

clear_1digit_LCD proc
        mov al,00h
        out 02h,al
        call DELAY_20ms
        mov al,10h                  ;shift left by 1
        out 00h,al
        call DELAY_20ms
        mov al,04h
        out 02h,al
        call DELAY_20ms
        mov al,00h
        out 02h,al

        mov al,0A0h
        out 00h,al
        call DELAY_20ms
        mov al,05h
        out 02h,al
        call DELAY_20ms
        mov al,01h
        out 02h,al                  ;prints Space

        call DELAY_20ms
        mov al,10h                  ;shift left by 1
        out 00h,al
        call DELAY_20ms
        mov al,04h
        out 02h,al
        call DELAY_20ms
        mov al,00h
        out 02h,al

RET
clear_1digit_LCD endp

error_msg proc
        mov al,0A0h
        out 00h,al
        call DELAY_20ms
        mov al,05h
        out 02h,al
        call DELAY_20ms
        mov al,01h
```

```
out 02h,al  ;prints Space

mov al,45h
out 00h,al
call DELAY_20ms
mov al,05h
out 02h,al
call DELAY_20ms
mov al,01h
out 02h,al  ;prints E

mov al,4Eh
out 00h,al
call DELAY_20ms
mov al,05h
out 02h,al
call DELAY_20ms
mov al,01h
out 02h,al  ;prints N

mov al,54h
out 00h,al
call DELAY_20ms
mov al,05h
out 02h,al
call DELAY_20ms
mov al,01h
out 02h,al  ;prints T

mov al,45h
out 00h,al
call DELAY_20ms
mov al,05h
```

```
out 02h,al
call DELAY_20ms
mov al,01h
out 02h,al  ;prints E

mov al,52h
out 00h,al
call DELAY_20ms
mov al,05h
out 02h,al
call DELAY_20ms
mov al,01h
out 02h,al  ;prints R

mov al,0A0h
out 00h,al
call DELAY_20ms
mov al,05h
out 02h,al
call DELAY_20ms
mov al,01h
out 02h,al  ;prints Space

mov al,31h
out 00h,al
call DELAY_20ms
mov al,05h
out 02h,al
call DELAY_20ms
mov al,01h
out 02h,al  ;prints 1

mov al,32h
```

```
out 00h,al
call DELAY_20ms
mov al,05h
out 02h,al
call DELAY_20ms
mov al,01h
out 02h,al  ;prints 2

mov al,0A0h
out 00h,al
call DELAY_20ms
mov al,05h
out 02h,al
call DELAY_20ms
mov al,01h
out 02h,al  ;prints Space

mov al,44h
out 00h,al
call DELAY_20ms
mov al,05h
out 02h,al
call DELAY_20ms
mov al,01h
out 02h,al  ;prints D

mov al,49h
out 00h,al
call DELAY_20ms
mov al,05h
out 02h,al
call DELAY_20ms
mov al,01h
```

```
out 02h,al  ;prints I

mov al,47h
out 00h,al
call DELAY_20ms
mov al,05h
out 02h,al
call DELAY_20ms
mov al,01h
out 02h,al  ;prints G

mov al,49h
out 00h,al
call DELAY_20ms
mov al,05h
out 02h,al
call DELAY_20ms
mov al,01h
out 02h,al  ;prints I

mov al,54h
out 00h,al
call DELAY_20ms
mov al,05h
out 02h,al
call DELAY_20ms
mov al,01h
out 02h,al  ;prints T

mov al,53h
out 00h,al
call DELAY_20ms
mov al,05h
```

```
        out 02h,al
        call DELAY_20ms
        mov al,01h
        out 02h,al  ;prints S
RET
error_msg endp

retry_msg proc
        mov al,0A0h
        out 00h,al
        call DELAY_20ms
        mov al,05h
        out 02h,al
        call DELAY_20ms
        mov al,01h
        out 02h,al  ;prints Space

        mov al,0A0h
        out 00h,al
        call DELAY_20ms
        mov al,05h
        out 02h,al
        call DELAY_20ms
        mov al,01h
        out 02h,al  ;prints Space

        mov al,0A0h
        out 00h,al
        call DELAY_20ms
        mov al,05h
        out 02h,al
        call DELAY_20ms
        mov al,01h
```

```
out 02h,al  ;prints Space

mov al,0A0h
out 00h,al
call DELAY_20ms
mov al,05h
out 02h,al
call DELAY_20ms
mov al,01h
out 02h,al  ;prints Space

mov al,52h
out 00h,al
call DELAY_20ms
mov al,05h
out 02h,al
call DELAY_20ms
mov al,01h
out 02h,al  ;prints R

mov al,45h
out 00h,al
call DELAY_20ms
mov al,05h
out 02h,al
call DELAY_20ms
mov al,01h
out 02h,al  ;prints E

mov al,54h
out 00h,al
call DELAY_20ms
mov al,05h
```

```
        out 02h,al
        call DELAY_20ms
        mov al,01h
        out 02h,al  ;prints T

        mov al,52h
        out 00h,al
        call DELAY_20ms
        mov al,05h
        out 02h,al
        call DELAY_20ms
        mov al,01h
        out 02h,al  ;prints R

        mov al,59h
        out 00h,al
        call DELAY_20ms
        mov al,05h
        out 02h,al
        call DELAY_20ms
        mov al,01h
        out 02h,al  ;prints Y

ret
retry_msg endp


updateday_msg proc
        mov al,55h
        out 00h,al
        call DELAY_20ms
        mov al,05h
        out 02h,al
```

```
call DELAY_20ms
mov al,01h
out 02h,al  ;prints U

mov al,50h
out 00h,al
call DELAY_20ms
mov al,05h
out 02h,al
call DELAY_20ms
mov al,01h
out 02h,al  ;prints P

mov al,44h
out 00h,al
call DELAY_20ms
mov al,05h
out 02h,al
call DELAY_20ms
mov al,01h
out 02h,al  ;prints D

mov al,41h
out 00h,al
call DELAY_20ms
mov al,05h
out 02h,al
call DELAY_20ms
mov al,01h
out 02h,al  ;prints A

mov al,54h
out 00h,al
```

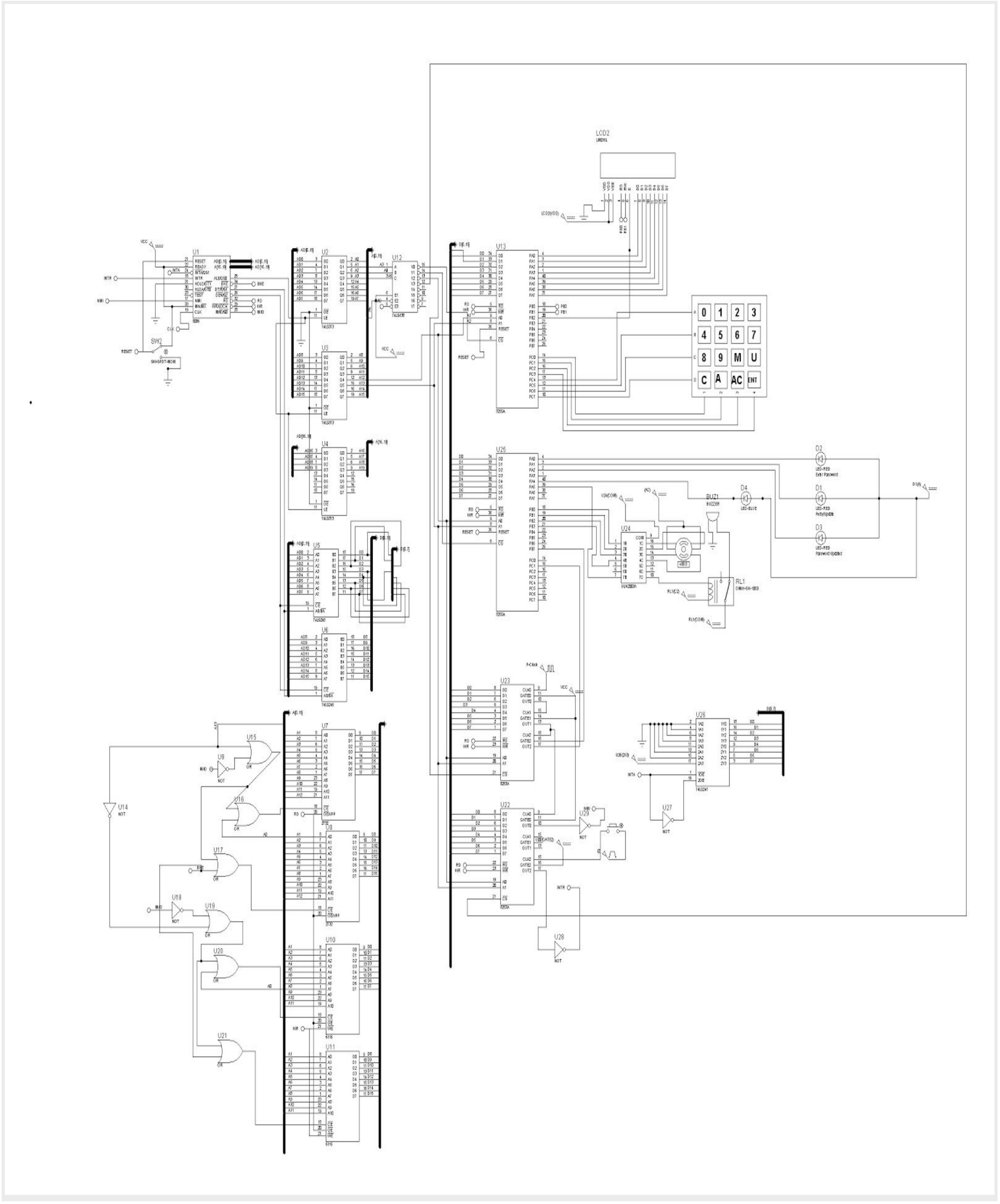
```
call DELAY_20ms
mov al,05h
out 02h,al
call DELAY_20ms
mov al,01h
out 02h,al  ;prints T

mov al,45h
out 00h,al
call DELAY_20ms
mov al,05h
out 02h,al
call DELAY_20ms
mov al,01h
out 02h,al  ;prints E

mov al,0a0h
out 00h,al
call DELAY_20ms
mov al,05h
out 02h,al
call DELAY_20ms
mov al,01h
out 02h,al  ;prints Space

mov al,44h
out 00h,al
call DELAY_20ms
mov al,05h
out 02h,al
call DELAY_20ms
mov al,01h
out 02h,al  ;prints D
```

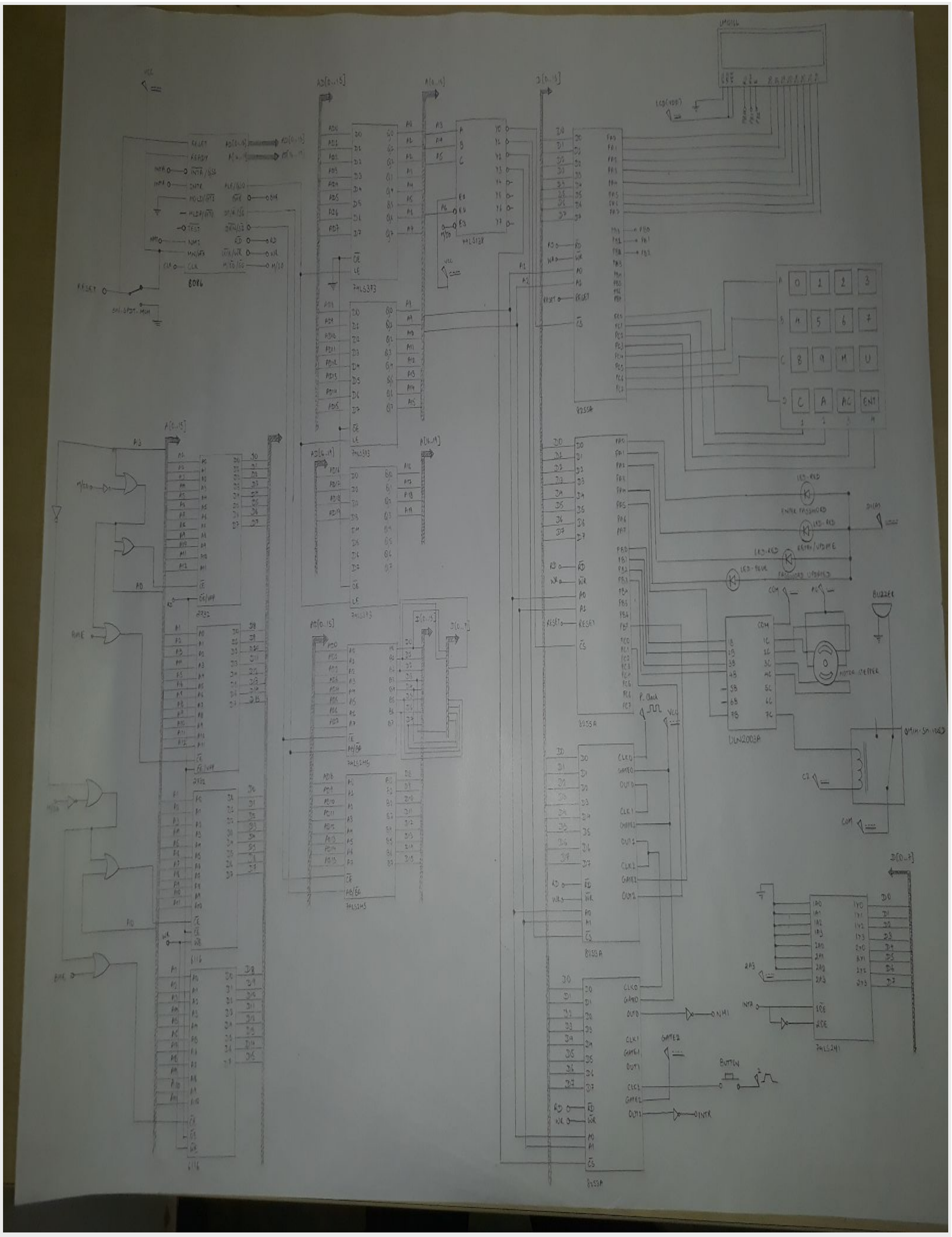
```
mov al,41h
out 00h,al
call DELAY_20ms
mov al,05h
out 02h,al
call DELAY_20ms
mov al,01h
out 02h,al  ;prints A

mov al,59h
out 00h,al
call DELAY_20ms
mov al,05h
out 02h,al
call DELAY_20ms
mov al,01h
out 02h,al  ;prints Y

mov al,0a0h
out 00h,al
call DELAY_20ms
mov al,05h
out 02h,al
call DELAY_20ms
mov al,01h
out 02h,al  ;prints Space

mov al,50h
out 00h,al
call DELAY_20ms
mov al,05h
out 02h,al
```

```
        call DELAY_20ms
        mov al,01h
        out 02h,al ;prints P

        mov al,41h
        out 00h,al
        call DELAY_20ms
        mov al,05h
        out 02h,al
        call DELAY_20ms
        mov al,01h
        out 02h,al  ;prints A

        mov al,53h
        out 00h,al
        call DELAY_20ms
        mov al,05h
        out 02h,al
        call DELAY_20ms
        mov al,01h
        out 02h,al  ;prints S

        mov al,53h
        out 00h,al
        call DELAY_20ms
        mov al,05h
        out 02h,al
        call DELAY_20ms
        mov al,01h
        out 02h,al  ;prints S
ret
updateday_msg endp
```

```
Nmi_24hrtimer:
                            call clear_LCD
                            call clear_1digit_LCD
                            call updateday_msg
startnmi:

              call keypad_input
              cmp al,0bbh
              jz master_mode
              jmp startnmi

 iret
 Switch_intR:
       call open_door
       sti
       cmp bp,0abcdh
       jz x6
       jnz start
```

Complete Circuit Diagram




8086
RESET
74LS373
74LS138
8255A
LM016L
0 1 2 3
4 5 6 7
8 9 M U
C A AC ENT
LED-RED
ENTER PASSWORD
LED-BLUE
PASSWORD UPDATED
BUZZER
MOTOR STEPPER
ULN2003A
2732
6116
74LS245
8253A
P. Clock
NMI
INTR
BUTTON
74LS241

References

- A 16x2 LCD display has been used. Following links were refered to understand its working:

  http://www.alldatasheet.com/view.jsp?Searchword=LMO16L
  http://www.sakshieducation.com/Engineering/Story.aspx?cid=12&nid=96054

- A 4x4 hex keypad has also been used.

  http://www.futurlec.com/Keypad4x4.shtml

- A stepper motor and a Darlington transistor has been used.

  https://www.youtube.com/watch?v=8aLkXsh1O44
  https://www.engineersgarage.com/electronic-components/uln2003-datasheet