



Summative Assessment Final Examination

Name: German E Felisarta III Group Number: 1

Date Completed: 12/16/2020

NOTE:

PASSWORD IS 1234

Design the I/O and memory interfaces of the microprocessor system:

Fig 1. I/O design

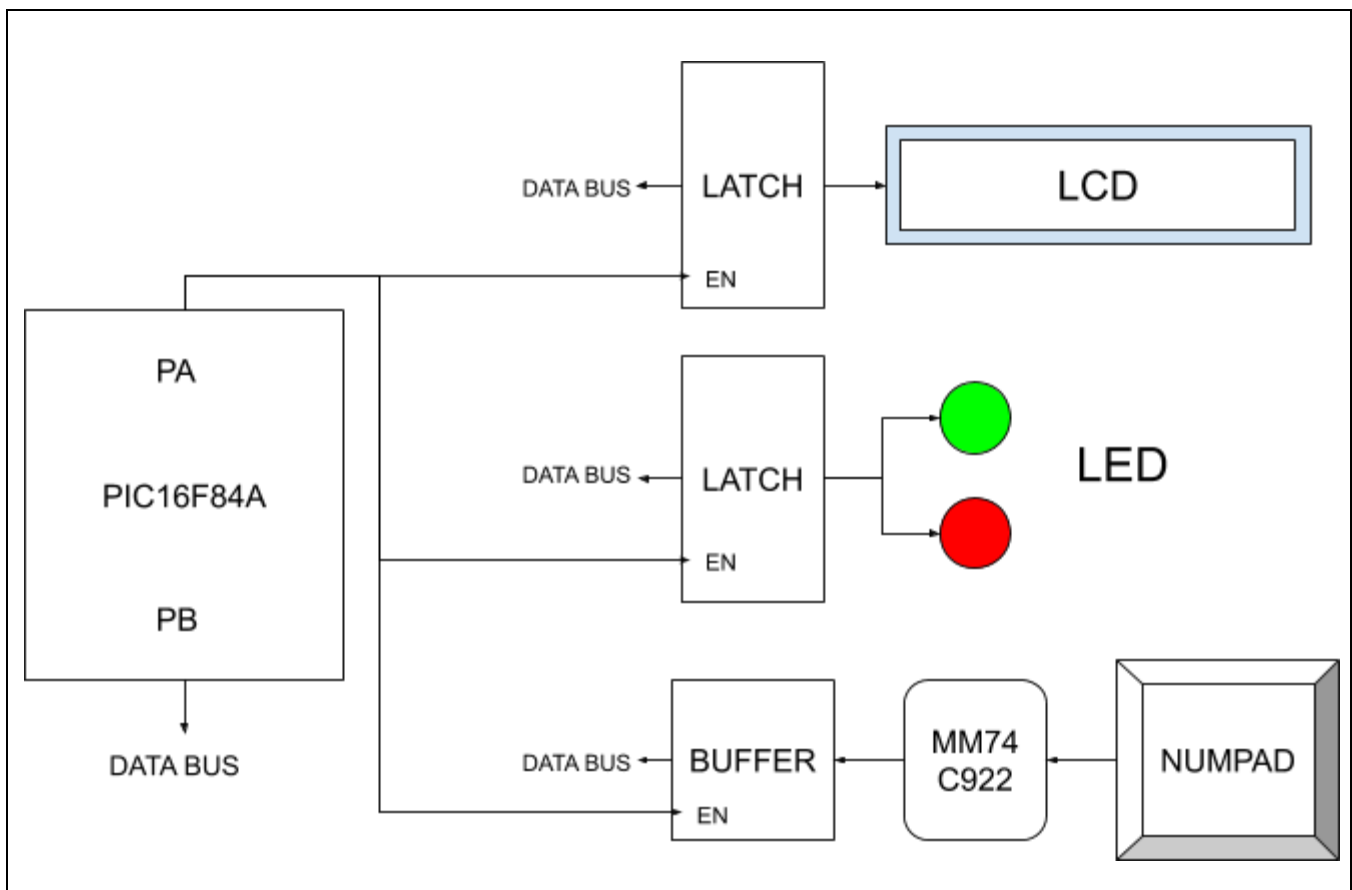




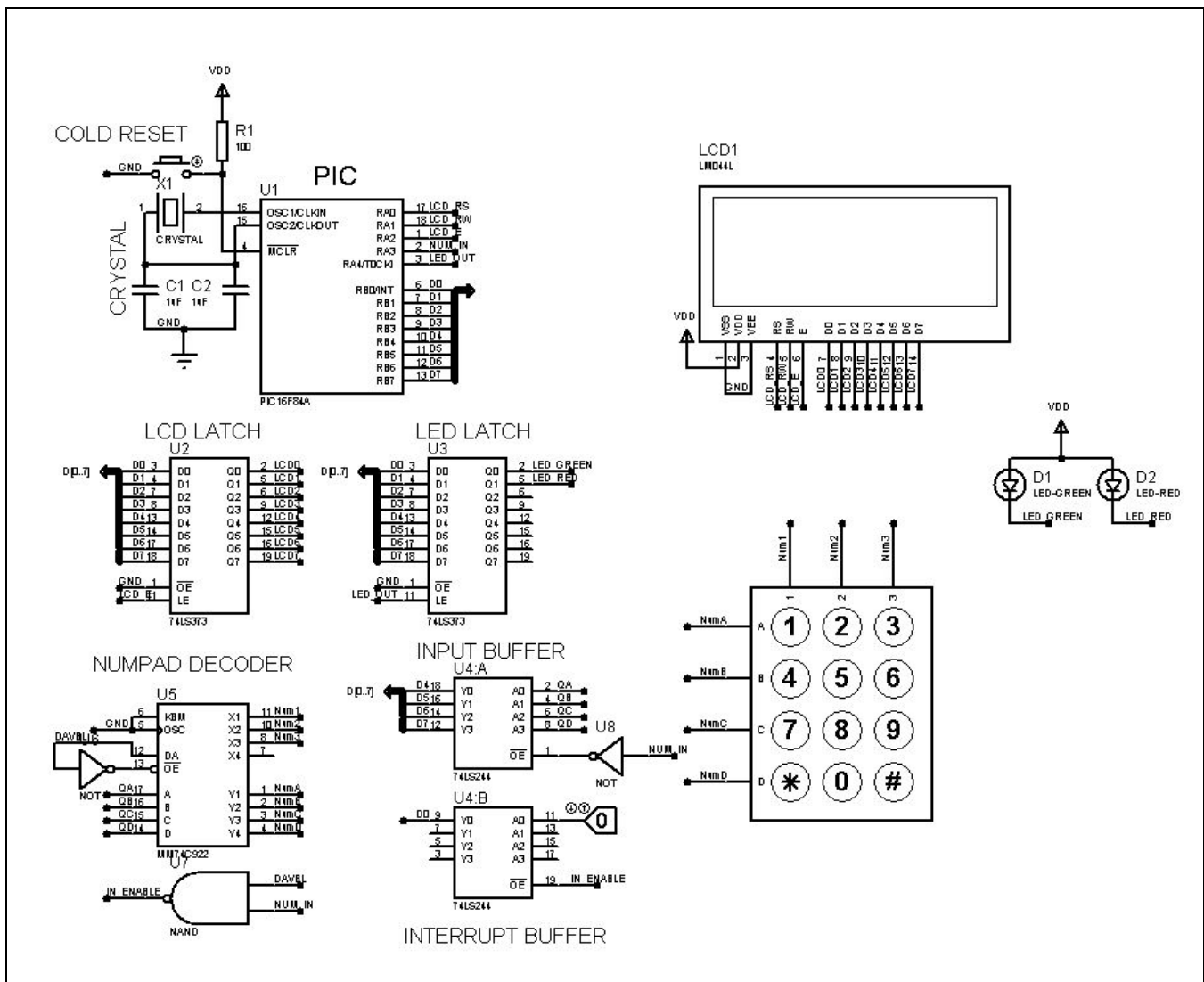
Fig 2. Memory

Memory allocated is only 1 byte

Count EQU at location 0CH

The Schematic diagram in Proteus:

Fig 3. Circuit Schematic





The programmed I/O assembly code for PIC16F84A:

Fig 4. Assembly code

```

LIST P=16F84A, F=INHX8M           ; MCU is PIC16F84A, output is Intel Hex
INCLUDE<P16F84A.INC>               ; include this file to use register names instead of addresses
__CONFIG _CP_OFF & _WDT_ON & _XT_OSC ; code protection off, Watchdog Timer on, XTAL osc used

COUNT EQU 0CH

ORG 000H
GOTO MAIN
ORG 004H                          ; interrupt vector
GOTO INT_RTN                      ; jump to label INT_RTN or the interrupt service routine

MAIN
; initialize PIC
BSF STATUS, RP0                  ; set register bank to 1
CLRF TRISA                       ; clear register TRISA (bank 1)
                                ; sets all bits in PORTA as "output"

CLRF TRISB
;MOVLW 0FFH                      ; load literal value 0FH to Wreg
;MOVWF TRISB                    ; move data in Wreg to register TRISB (bank 1)
                                ; set RB0 in PORTB as "input", the rest as "output"
MOVLW 02CH                      ; load literal value 0CH to Wreg
                                ; prescaler 1:32 assigned to Watchdog Timer (WDT)
MOVWF OPTION_REG                 ; move data in Wreg to register OPTION_REG (bank 0/1)
BCF INTCON, INTF                 ; clear the RB0/INT interrupt flag
BSF INTCON, INTE                 ; unmask (enable) RB0/INT interrupt source
BSF INTCON, GIE                 ; enable all unmasked interrupt
BCF STATUS, RP0                 ; set register bank to 0

MOVLW 00H
MOVWF COUNT

START
;THIS PART WILL CHECK IF MAX TRIES IS REACHED
INCF COUNT, w                   ; increment count
MOVWF COUNT

SUBLW 04H
BTFSS STATUS, Z                 ; checks if equal
GOTO CONT                      ; not skip if equal

CALL OUTPUT_PORTB               ; reset lcd and goto system lockup
CALL INIT_LCD
GOTO DISPLAY_FAIL

CONT
CALL OUTPUT_PORTB
BSF PORTA, 4                   ; INITIALIZE LED
MOVLW 02H                      ; Turn on only green led
MOVWF PORTB
BCF PORTA, 4

```



```
SLEEP
CALL OUTPUT_PORTB
CALL INIT_LCD
CALL DISPLAY_CHECKPASS
MOVLW 0DCH                                ; MOVE TO LAST ROW
CALL INST_CTRL
;ENABLE INPUTS PORTB
CALL INPUT_PORTB
BSF PORTA, 3
CLRF PORTB

CHECK_1                                    ; First Digit
NOP                                       ; loop while waiting for interrupt
CLRWDW                                     ; reset timer to prevent system reset
BTFSC PORTA, 3
GOTO CHECK_1

SUBLW 01H
BTFSS STATUS, Z                         ; checks if equal
GOTO START                             ; skip if equal

CALL OUTPUT_PORTB
CALL DISPLAY_AST
CALL INPUT_PORTB
BSF PORTA, 3

CHECK_2                                    ; Second Digit
NOP                                       ; loop while waiting for interrupt
CLRWDW                                     ; reset timer to prevent system reset
BTFSC PORTA, 3
GOTO CHECK_2

SUBLW 02H
BTFSS STATUS, Z                         ; checks if equal to 1
GOTO START

CALL OUTPUT_PORTB
CALL DISPLAY_AST
CALL INPUT_PORTB
BSF PORTA, 3

CHECK_3                                    ; Third Digit
NOP                                       ; loop while waiting for interrupt
CLRWDW                                     ; reset timer to prevent system reset
BTFSC PORTA, 3
GOTO CHECK_3

SUBLW 03H
BTFSS STATUS, Z                         ; checks if equal to 1
GOTO START

CALL OUTPUT_PORTB
CALL DISPLAY_AST
```



```
CALL INPUT_PORTB
BSF PORTA, 3

CHECK_4                                ; Fourth Digit
NOP                                    ; loop while waiting for interrupt
CLRWDT                                ; reset timer to prevent system reset
BTFSC PORTA, 3
GOTO CHECK_4

SUBLW 04H
BTFSS STATUS, Z                        ; checks if equal to 1
GOTO START

CALL OUTPUT_PORTB
CALL DISPLAY_AST
CALL INIT_LCD
CALL DISPLAY_PASS

CALL INPUT_PORTB

CHECK_LOCK
BSF PORTA, 3
CHECK_LOCK_LOOP                        ; LOOP UNLOCKED
NOP
CLRWDT
BTFSC PORTA, 3
GOTO CHECK_LOCK

SUBLW 0AH
BTFSS STATUS, Z                        ; checks if equal to 1
GOTO CHECK_LOCK

GOTO START

;=====
; EXTRA FUNCTIONS
;=====

DISPLAY_AST
CALL OUTPUT_PORTB
MOVLW 02AH                            ; * - 00101010
CALL DATA_CTRL
RETURN

OUTPUT_PORTB
CLRF PORTA
CLRF PORTB
BSF STATUS, RP0                        ; set register bank to 1
CLRF TRISB                             ; move data in Wreg to register TRISB (bank 1)
BCF STATUS, RP0                        ; set register bank to 0
RETURN

INPUT_PORTB
```



```
CLRF PORTA
CLRF PORTB
BSF STATUS, RP0           ; set register bank to 1
MOVLW 0FFH                ; load literal value 01H to Wreg
MOVWF TRISB               ; move data in Wreg to register TRISB (bank 1)
BCF STATUS, RP0           ; set register bank to 0
RETURN
```

```
DISPLAY_CHECKPASS
MOVLW 086H                ; First row, 1st column
CALL INST_CTRL
MOVLW 050H                ; P - 01010000
CALL DATA_CTRL
MOVLW 041H                ; A - 01000001
CALL DATA_CTRL
MOVLW 053H                ; S - 01010011
CALL DATA_CTRL
MOVLW 053H                ; S - 01010011
CALL DATA_CTRL
MOVLW 043H                ; C - 01000011
CALL DATA_CTRL
MOVLW 04FH                ; O - 01001111
CALL DATA_CTRL
MOVLW 044H                ; D - 01000100
CALL DATA_CTRL
MOVLW 045H                ; E - 01000101
CALL DATA_CTRL
MOVLW 03AH                ; : - 00111010
CALL DATA_CTRL
RETURN
```

```
DISPLAY_PASS
MOVLW 084H                ; First row, 4th column
CALL INST_CTRL
MOVLW 050H                ; P - 01010000
CALL DATA_CTRL
MOVLW 041H                ; A - 01001000
CALL DATA_CTRL
MOVLW 053H                ; S - 01010011
CALL DATA_CTRL
MOVLW 053H                ; S - 01010011
CALL DATA_CTRL
MOVLW 043H                ; C - 01000011
CALL DATA_CTRL
MOVLW 04FH                ; O - 01001111
CALL DATA_CTRL
MOVLW 044H                ; D - 01000100
CALL DATA_CTRL
MOVLW 045H                ; E - 01000101
CALL DATA_CTRL
MOVLW 0A0H                ; [space] - 10100000
CALL DATA_CTRL
MOVLW 04FH                ; O - 01001111
```



```
CALL DATA_CTRL
MOVLW 04BH                ; K - 01001011
CALL DATA_CTRL
MOVLW 021H                ; ! - 00100001
CALL DATA_CTRL

MOVLW 0D7H                ; Last row, 7th column
CALL INST_CTRL
MOVLW 050H                ; P - 01010000
CALL DATA_CTRL
MOVLW 052H                ; R - 01010010
CALL DATA_CTRL
MOVLW 045H                ; E - 01000101
CALL DATA_CTRL
MOVLW 053H                ; S - 01010011
CALL DATA_CTRL
MOVLW 053H                ; S - 01010011
CALL DATA_CTRL
MOVLW 0A0H                ; [space] - 10100000
CALL DATA_CTRL
MOVLW 02AH                ; * - 00101010
CALL DATA_CTRL
MOVLW 0A0H                ; [space] - 10100000
CALL DATA_CTRL
MOVLW 054H                ; T - 01010100
CALL DATA_CTRL
MOVLW 04FH                ; O - 01001111
CALL DATA_CTRL
MOVLW 0A0H                ; [space] - 10100000
CALL DATA_CTRL
MOVLW 04CH                ; L - 01001100
CALL DATA_CTRL
MOVLW 04FH                ; O - 01001111
CALL DATA_CTRL
MOVLW 043H                ; C - 01000011
CALL DATA_CTRL
MOVLW 04BH                ; K - 01001011
CALL DATA_CTRL
RETURN
```

```
DISPLAY_FAIL
MOVLW 0C3H                ; Last row, 7th column
CALL INST_CTRL
MOVLW 053H                ; S - 01010011
CALL DATA_CTRL
MOVLW 059H                ; Y - 01011001
CALL DATA_CTRL
MOVLW 053H                ; S - 01010011
CALL DATA_CTRL
MOVLW 054H                ; T - 01010100
CALL DATA_CTRL
MOVLW 045H                ; E - 01000101
CALL DATA_CTRL
```



```
MOVLW 04DH                ; M - 01001101
CALL DATA_CTRL
MOVLW 0A0H                ; [space] - 10100000
CALL DATA_CTRL
MOVLW 04CH                ; L - 01001100
CALL DATA_CTRL
MOVLW 04FH                ; O - 01001111
CALL DATA_CTRL
MOVLW 043H                ; C - 01000011
CALL DATA_CTRL
MOVLW 04BH                ; K - 01001011
CALL DATA_CTRL
MOVLW 055H                ; U - 01010101
CALL DATA_CTRL
MOVLW 050H                ; P - 01000011
CALL DATA_CTRL
```

```
BSF PORTA, 4              ; ITURN OFF GREEN LED
MOVLW 01H
MOVWF PORTB
MOVLW 02H
```

```
LOOPFAIL                  ; LOOP FLASHING RED LED
XORWF PORTB
SLEEP
GOTO LOOPFAIL

RETURN
```

```
INIT_LCD
MOVLW 038H                ;set function
CALL INST_CTRL
MOVLW 00EH                ;Display on with cursor
CALL INST_CTRL
MOVLW 001H                ;Display Clear
CALL INST_CTRL
MOVLW 006H                ;Entry Mode Set
CALL INST_CTRL
RETURN
```

```
INST_CTRL MOVWF PORTB      ; move data of Wreg to LCD data bus
MOVLW 004H                ; setup control (RS='0', RW='0', E='1')
MOVWF PORTA               ; send control signals to LCD
SLEEP                     ; delay
BCF PORTA, 2              ; clear enable bit (E='0')
RETURN                    ; return from function call
```

```
DATA_CTRL MOVWF PORTB      ; move data of Wreg to LCD data bus
MOVLW 005H                ; setup control (RS='1', RW='0', E='1')
MOVWF PORTA               ; send control signals to LCD
SLEEP                     ; delay
BCF PORTA, 2              ; clear enable bit (E='0')
```




RETURN ; return from function call

KEY_VAL ADDWF PCL, F

RETLW 01H ; returns 01H for key 1 (addr 00H) DEFAULT
RETLW 02H ; returns 02H for key 2 (addr 01H) MOVWF 01H
RETLW 03H ; returns 03H for key 2 (addr 02H)
RETLW 00H ; returns 00H for key A (addr 03H).
RETLW 04H ; returns 04H for key 4 (addr 04H).
RETLW 05H ; returns 05H for key 5 (addr 05H).
RETLW 06H ; returns 06H for key 6 (addr 06H).
RETLW 00H ; returns 00H for key B (addr 07H).
RETLW 07H ; returns 07H for key 7 (addr 08H).
RETLW 08H ; returns 08H for key 8 (addr 09H).
RETLW 09H ; returns 09H for key 9 (addr 0AH).
RETLW 00H ; returns 00H for key C (addr 0BH).
RETLW 0AH ; returns 00H for key * (addr 0CH).
RETLW 00H ; returns 00H for key 0 (addr 0DH).
RETLW 00H ; returns 00H for key # (addr 0EH).
RETLW 00H ; returns 00H for key D (addr 0FH) - unreachable

=====

; INTERRUPT FUNCTIONS

=====

INT_RTN

BCF INTCON, GIE ; disable all unmasked interrupt to prevent interrupt overriding
BTFSS INTCON, INTF ; check the RB0/INT interrupt flag is '1' (interrupt source from RB0/INT)
GOTO EXIT ; exit ISR if not RB0/INT interrupt

CLRWDT ;reset timer to prevent system reset
SWAPF PORTB, 0
BSF STATUS, RP0
ANDLW 0FH
BCF STATUS, RP0
CALL KEY_VAL ; get the value from keypad

BCF PORTA, 3

EXIT BSF INTCON, GIE ; enable all unmasked interrupt
BCF INTCON, INTF
RETFIE ; return from interrupt routine

=====

END



Simulation demonstrating the required functions:

Fig 5. System Turns on and displays "PASSCODE"

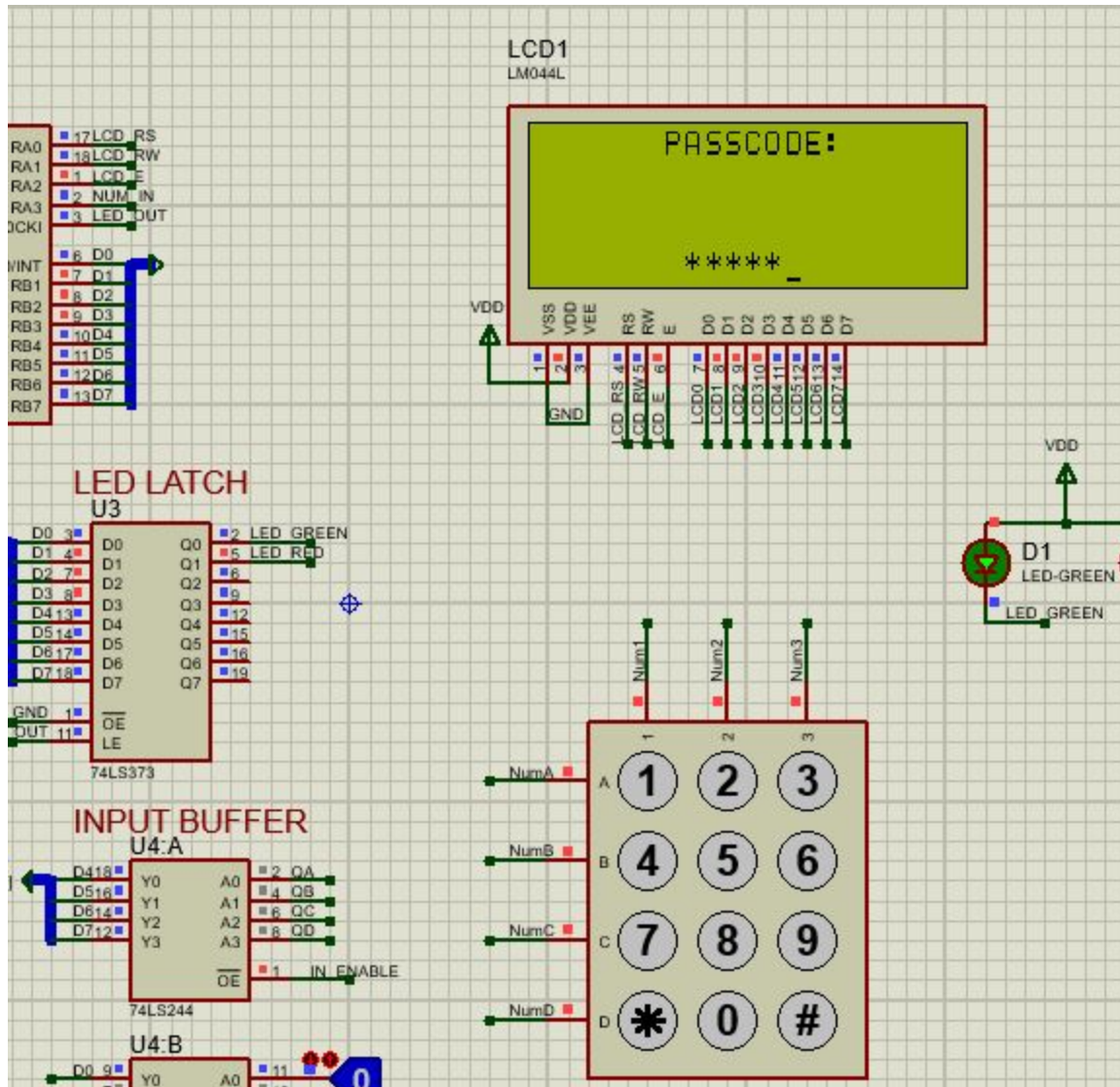


Fig 6. System displays "PASSCODE OK!"

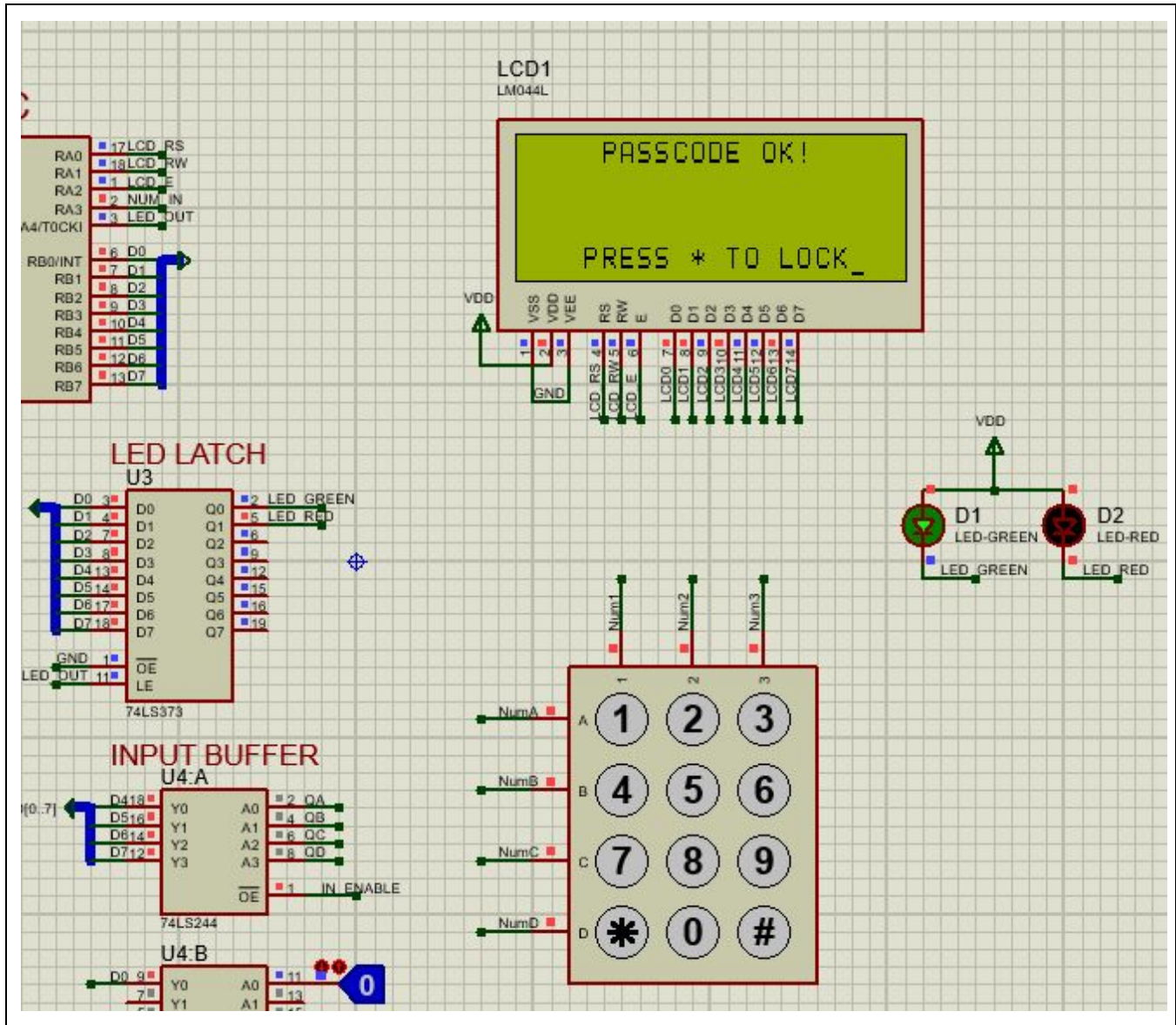
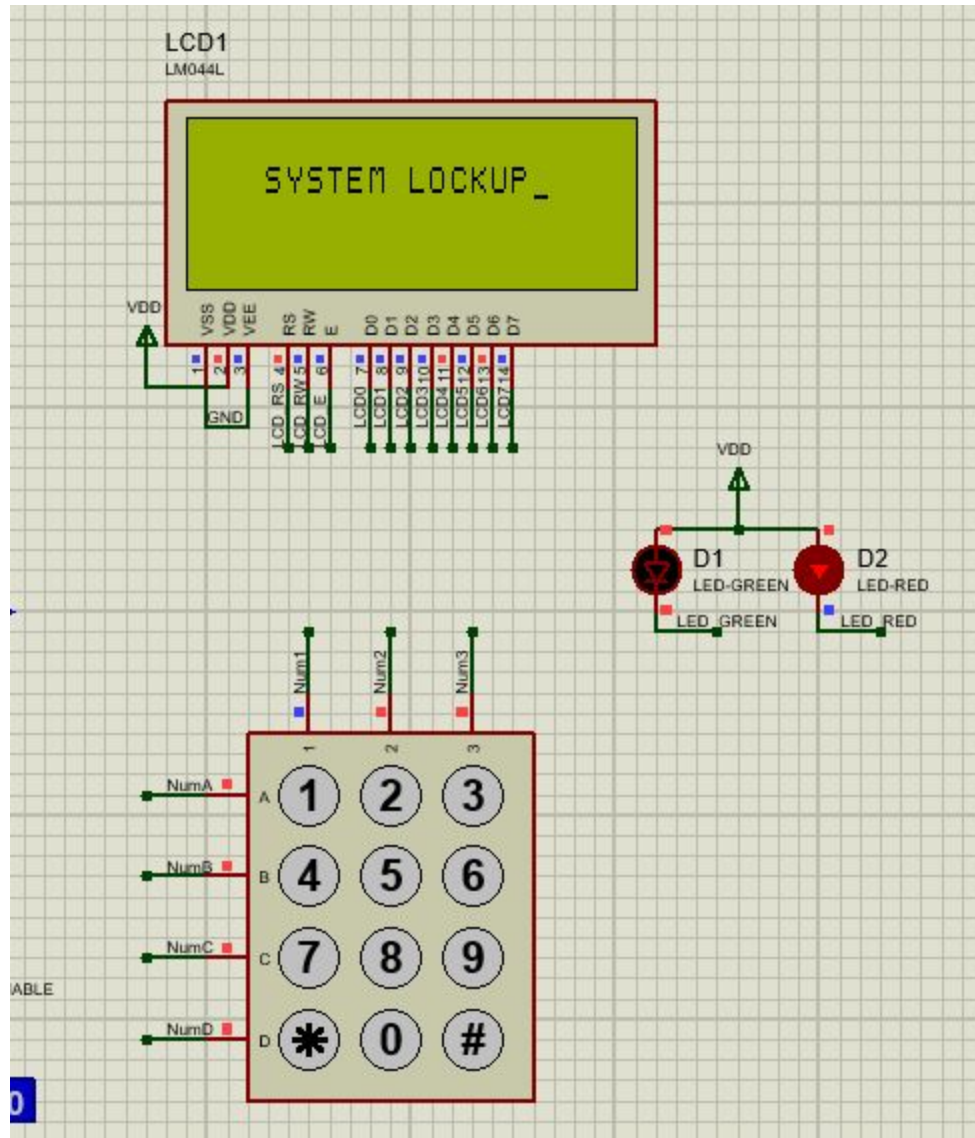


Fig 7. The system displays "SYSTEM LOCKUP_" and red LED



Video Demonstration:

<https://youtu.be/TAp916WYnNU>