An-Najah National University
Faculty of Engineering and IT

جامعة النجاح الوطنية
كلية الهندسة وتكنولوجيا المعلومات

Computer Engineering Department
Course Name: Microprocessor Lab Number: 10636392

Lab Report Grading Sheet

Instructor: Dr.Manar Qamhieh	Experiment #: 10
Academic Year: 2023/2024	Experiment Name: Graphical 128x64 Dot Matrix LCD conversion
Semester:2nd	

Students	
1-Salsabeel Dwaikat	2-Doaa Yasin Jararaa
3-Wala' Essam Ashqar	4-
Performed on: 5/3/024	Submitted on:12/3/2024

Report's Outcomes

ILO __ =( ) %	ILO __ =( ) %	ILO __ =( ) %	ILO __ =( ) %	ILO __ =( ) %

Evaluation Criterion	Grade	Points
Abstract answers to the questions: "What did you do? How did you do it? What did you find?"	0.5	
Introduction and Theory Sufficient, clear, and complete statement of objectives. In addition, it Presents sufficiently the theoretical basis.	1.5	
Apparatus/ Procedure Apparatus sufficiently described to enable another experimenter to identify the equipment needed to experiment. The procedure is sufficiently described.	2	
Experimental Results and Discussion (In-Lab Worksheet) Crisp explanation of experimental results. Comparison of theoretical predictions to experimental results, including discussion of accuracy and error analysis in some cases.	4	
Conclusions and Recommendations Conclusions summarize the major findings from the experimental results with adequate specificity. Recommendations are appropriate in light of the conclusions. Correct grammar.	1	
Appearance The title page is complete, page numbers are applied, content is well organized, correct spelling, fonts are consistent, good visual appeal.	1	
Total	10	

Objectives:

in this Experiment, we have a couple of goals to achieve

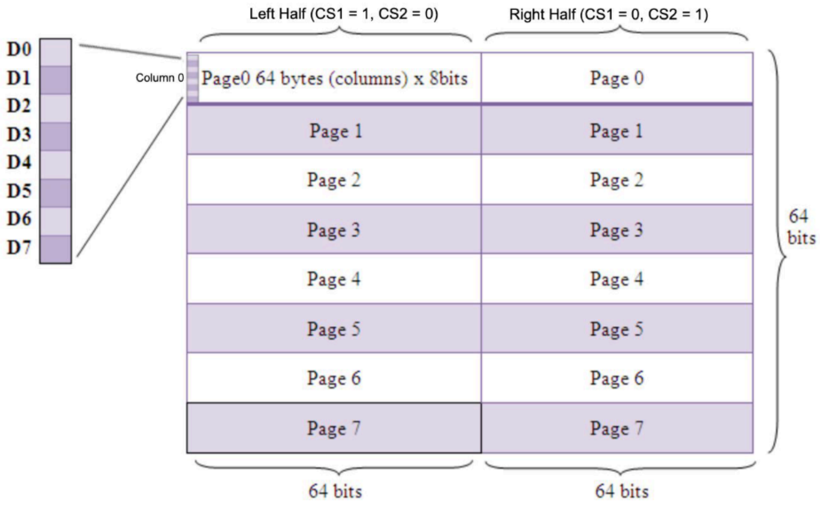
- Understand the working principle and display method of the dot-matrix liquid crystal display.
- Mastering the interface design and programming between the 8088 microcomputer system and liquid crystal display (LCD)

introduction:

The Graphical LCDs are thus used to display customized characters and images. The LCD used in this lab has 128×64 dots and has a yellow-green color backlight. The 128x64 LCD is divided into two equal halves with each half being controlled by a separate controller, and the whole LCD is divided equally into 8 pages Each page consists of 8 rows and 64 columns

So two horizontal pages make 128 (64×2) columns and 8 vertical pages make 64 rows (8×8).

The paging scheme of the graphical LCD can be easily understood from the following table:



Tools and equipment:

- ❖ MML8086K
- ❖ MML8086K Software: dice8088
- ❖ liquid crystal display (GLCD)
- ❖ 8255(port A and C)

Procedure:

In the first task, we were asked to print a single character (A) at the left-top side of the LCD (left half, first column, first page)

The CODE:

```
CODE SEGMENT
ASSUME CS:CODE
PA EQU 0FF28H         ; PA Data port
PCTL EQU 0FF2BH         ; 8255 Command port
RSN EQU 00H ; PC0   bit set/reset mode of 8255 (PC)
RS EQU 01H ;set
RWN EQU 02H ; PC1
RW EQU 03H ;set
EN EQU 04H ; PC2   Enable = 0
E EQU 05H    ;;set      Enable = 1
CS1N EQU 08H ; PC4
CS1 EQU 09H ;set
CS2N EQU 0Ch ;  PC6
CS2 EQU 0Dh ;set
ORG 22E0h

JMP START
 YR  db   ?              ; column address
 pag db  ?             ; page address
 ZR  db   0c0H          ; always first row of page (don't change)
 val db  ?             ; value of command or data

START:
; configure 8255
mov dx,0ff2bh
mov al,80h
out dx,al
; initialize LCD (Display ON command) for both halves (call sendComm procedure)
; left half
call selectleft
mov al,03fh
mov val,al
call SendComm
; delay
call DELAY2MS
; delay
call DELAY2MS
mov al,03fh
mov val,al
call sendcomm
call DELAY2MS

MAIN:
; select first column and first page
mov yr,40h
mov pag,0b8h

; select left half of LCD and set cursor
call selectleft
call setCursor

; get offset of character to be displayed and call sendData procedure
mov si,offset CHARA
call dispComm

; this can be repeated to print an many characters as require
jmp $                 ; stay at current location

; procecure to display a single character (8 columns)
; loop through the columns of the character and call sendData

dispComm:
mov cx,8
l:
mov al,[si]
mov val,al
call senddata
inc si
loop l
RET

; Procedure to send a command to LCD (command value is in variable called val)
SendComm:
mov al, val
mov dx,pa
out dx,al
mov al, rsn
mov dx,pctl
out dx,al
mov al, RWN
mov dx,pctl
out dx,al
mov al, EN
mov dx,pctl
out dx,al
call DELAY2MS
mov al, E
mov dx,pctl
out dx,al
call DELAY2MS
mov al, EN
mov dx,pctl
out dx,al
call DELAY2MS
RET


; Procedure to send a Data (single column) to LCD (data value is in variable called val)
SendData:
mov al, val
mov dx,pa
out dx,al
mov al, rs
mov dx,pctl
out dx,al
mov al, RWN
mov dx,pctl
out dx,al
mov al, EN
mov dx,pctl
out dx,al
call DELAY2MS
mov al, E
mov dx,pctl
out dx,al
call DELAY2MS
mov al, EN
mov dx,pctl
out dx,al
call DELAY2MS
RET

; set cursor of LCD to a certain page line and a certain column.
; LCD half should be already selected
setCursor:
mov al,yr
mov val,al
call sendcomm
mov al,pag
mov val,al
call sendcomm
mov al,zr
mov val,al
call sendcomm
; set column (send YR value as command
; set page (send PAG (x address) value as command)
; set row (send ZR value as command)
RET


; enable left half of the LCD (CS1 = 1, CS2 = 0)
SELECTLEFT:
mov al,cs1
mov dx,PCTL
out dx,al
mov al,cs2n
mov dx,PCTL
out dx,al
RET

; enable right half of the LCD (CS1 = 0, CS2 = 1)
SELECTRIGHT:
mov al,cs2
mov dx,PCTL
out dx,al
mov al,cs1n
mov dx,PCTL
out dx,al
 RET
DELAY2MS:
push cx
MOV CX,78H
LOOP $            ; current position
pop cx
RET

CharEmpty: DB  00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h        ; empty block of 8x8 pixels
CHARA:     DB  07eh,011h,011h,011h,07eh, 00h,00h,00h        ; character A on a block of 8x8
pixels
CHARFULL:  DB  0FFh, 0FFh, 0FFh, 0FFh, 0FFh, 0FFh, 0FFh, 0FFh  ; all black

CODE ENDS
END START
```

2. Print (Hello) starting at the same previous position.

```
CODE SEGMENT

ASSUME CS:CODE

PA EQU 0FF28H        ; PA Data port
```

```
PCTL EQU 0FF2BH          ; 8255 Command port
RSN EQU 00H ; PC0   bit set/reset mode of 8255 (PC)
RS EQU 01H ;set
RWN EQU 02H ; PC1
RW EQU 03H ;set
EN EQU 04H ; PC2   Enable = 0
E EQU 05H    ;;set  Enable = 1
CS1N EQU 08H ; PC4
CS1 EQU 09H ;set
CS2N EQU 0Ch ;  PC6
CS2 EQU 0Dh ;set
ORG 22E0h

JMP START
 YR  db   ?              ; column address
 pag db  ?              ; page address
 ZR  db   0c0H          ; always first row of page (don't change)
 val db  ?              ; value of command or data

START:
; configure 8255
mov dx,0ff2bh
mov al,80h
out dx,al

; initialize LCD (Display ON command) for both halves (call sendComm procedure)
; left half
call selectleft
mov al,03fh
mov val,al

call SendComm


; delay

call DELAY2MS

mov al,03fh

mov val,al

call sendcomm

call DELAY2MS


MAIN:

; select first column and first page

Mov yr,40h

mov pag,0b8h

; select left half of LCD and set cursor

call selectleft

call setCursor

; get offset of character to be displayed and call sendData procedure

; this can be repeated to print an many characters as required

mov si,offset CHARH

call dispComm

mov si,offset CHARE

call dispComm

mov si,offset CHARL

call dispComm

mov si,offset CHARL

call dispComm

mov si,offset CHARO

call dispComm
```

```
jmp $                 ; stay at current location


; procecure to display a single character (8 columns)

; loop through the columns of the character and call sendData

dispComm:

mov cx,8

l:

mov al,[si]

mov val,al

call senddata

inc si

loop l

RET


; Procedure to send a command to LCD (command value is in variable called val)

SendComm:

mov al, val

mov dx,pa

out dx,al

mov al, rsn

mov dx,pctl

out dx,al

mov al, RWN

mov dx,pctl

out dx,al

mov al, EN

mov dx,pctl

out dx,al

call DELAY2MS
```

```
mov al, E
mov dx,pctl
out dx,al
call DELAY2MS
mov al, EN
mov dx,pctl
out dx,al
call DELAY2MS
RET

; Procedure to send a Data (single column) to LCD (data value is in variable called val)
SendData:
mov al, val
mov dx,pa
out dx,al
mov al, rs
mov dx,pctl
out dx,al
mov al, RWN
mov dx,pctl
out dx,al
mov al, EN
mov dx,pctl
out dx,al
call DELAY2MS
mov al, E
mov dx,pctl
out dx,al
call DELAY2MS
```

```
mov al, EN

mov dx,pctl

out dx,al

call DELAY2MS

RET


; set cursor of LCD to a certain page line and a certain column.

; LCD half should be already selected

setCursor:

mov al,yr

mov val,al

call sendcomm

mov al,pag

mov val,al

call sendcomm

mov al,zr

mov val,al

call sendcomm

; set column (send YR value as command

; set page (send PAG (x address) value as command)

; set row (send ZR value as command

RET


; enable left half of the LCD (CS1 = 1, CS2 = 0)

SELECTLEFT:

mov al,cs1

mov dx,PCTL

out dx,al

mov al,cs2n
mov dx,PCTL
out dx,al
RET

; enable right half of the LCD (CS1 = 0, CS2 = 1)
SELECTRIGHT:
mov al,cs2
mov dx,PCTL
out dx,al
mov al,cs1n
mov dx,PCTL
out dx,al
RET

DELAY2MS:
push cx
MOV CX,78H
LOOP $          ; current position
pop cx
RET

CharEmpty: DB  00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h       ; empty block of 8x8 pixels
CHARH:     DB  7Fh,8h,8h,8h,7Fh,00h,00h,00h
CHARE:     DB  7fh,49h,49h,49h,41h,00h,00h,00h
CHARL:     DB  7fh,40h,40h,40h,40h,00h,00h,00h
CHARO:     DB  3EH,41H,41H,41H,3EH,00H,00H,00H
CHARFULL:  DB  0FFh, 0FFh, 0FFh, 0FFh, 0FFh, 0FFh, 0FFh, 0FFh  ; all black
CODE ENDS
END START
```

This is the code for the clear part:

```
MOV pag ,0b8h
mov cx,8
lp4:
MOV yr,40h
push cx
mov cx,8
lp3:
mov si,offset charempty
call dispComm
loop lp3
inc pag
pop cx
loop lp4
```

Conclusions:

In this experiment, we learned how to deal with LCD and control it(display information on it or clear it)

OUTPUT: