



Lucrarea de laborator 4.

Dispozitive
periferice de I/O
sub FreeBSD.
Interfața paralelă.
Aplicație practică.



1. Scopul lucrării

Scopul lucrării este de a prezenta un exemplu de aplicație practică pentru interfața paralelă, folosind ecrane LCD 2x16 bazate pe controlere Hitachi HD44780.

Pe parcursul acestei lucrări, se vor prezenta schemele generare ale unui LCD 2x16 (2 linii, 16 coloane) comandat de un controler HD44780, schemele de conectare a LCD-ului la interfața paralelă a calculatorului, setul de comenzi a controlerului Hitachi HD44780, precum și câteva exemple de programe scrise în NASM pentru a demonstra modul de funcționare a LCD-ului.

2. Considerații teoretice



Ce este HD44780?



HD44780 este un controler folosit pentru comandarea display-urilor LCD (Liquid Crystal Display), fabricat de firma Hitachi. Cu ajutorul lui, LCD-ul poate afișa caractere alfanumerice sau diferite simboluri atât de tipul 5x8 cât și 5x10. Un manual complet al HD44780 poate fi consultat la <http://www.electronic-engineering.ch/microchip/datasheets/lcd/hd44780.pdf>

2.1. LCD 2x16 bazat pe HD44780

În cadrul lucrării, se va folosi un model de LCD cu 2 linii și 16 coloane, prezentat în figurile 4.1 respectiv 4.2, bazat pe controlerul HD44780.



Figura 4.1. LCD 2x16 bazat pe HD44780 (față).

Se observă cei 16 pini prezenți în componența LCD-ului. Aceștia vor fi folosiți la interfațarea LCD-ului (prin controlerul HD44780) cu portul paralel al calculatorului. Mai multe informații despre modul de conectare se găsesc în secțiunea 2.2 a lucrării.

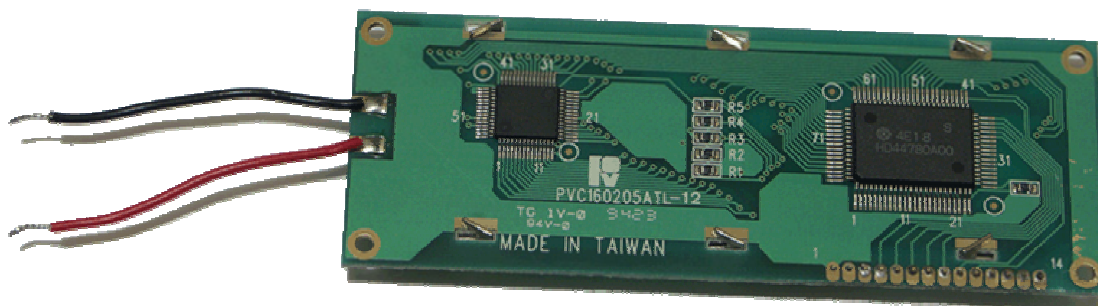


Figura 4.2. LCD 2x16 bazat pe HD44780 (spate).

În anexa lucrării sunt prezentate o serie de adrese web care conțin informații despre LCD-uri, posibilitatea conectării acestora la interfața paralelă a calculatorului, precum și exemple de programe care interacționează cu acestea.

2.2. Interfațarea cu portul paralel al calculatorului (LPT)

Pentru conectarea la interfața paralelă, se vor folosi cei 16 pini ai LCD-ului. Semnificația acestora este prezentată în tabelul 4.1.

Pin	Semnal	Funcție	Descriere
1	Vss	Putere	Legat la GND.
2	Vdd	Putere	Legat la +5V.
3	Vee	Ajustare contrast	0-5V pentru contrast variabil.
4	RS	Control	Selecție registru date sau comenzi. (en. Register Select)
5	R/W	Control	Selecție citire sau scriere (en. Read/Write). Legat la GND pentru scriere.
6	E	Control	Inițiază transferul efectiv al datelor. (en. Enable)
7	D0	I/O	Linia de date 0.
8	D1	I/O	Linia de date 1.
9	D2	I/O	Linia de date 2.
10	D3	I/O	Linia de date 3.
11	D4	I/O	Linia de date 4.
12	D5	I/O	Linia de date 5.
13	D6	I/O	Linia de date 6.
14	D7	I/O	Linia de date 7.
15	GND	Putere	Legat la GND.
16	BL	I/O-Control	Activare lumină fundal.

Tabelul 4.1. Semnificația pinilor LCD

Componentele necesare conectării precum și schema de conectare sunt prezentate în figurile 4.3, respectiv 4.4.

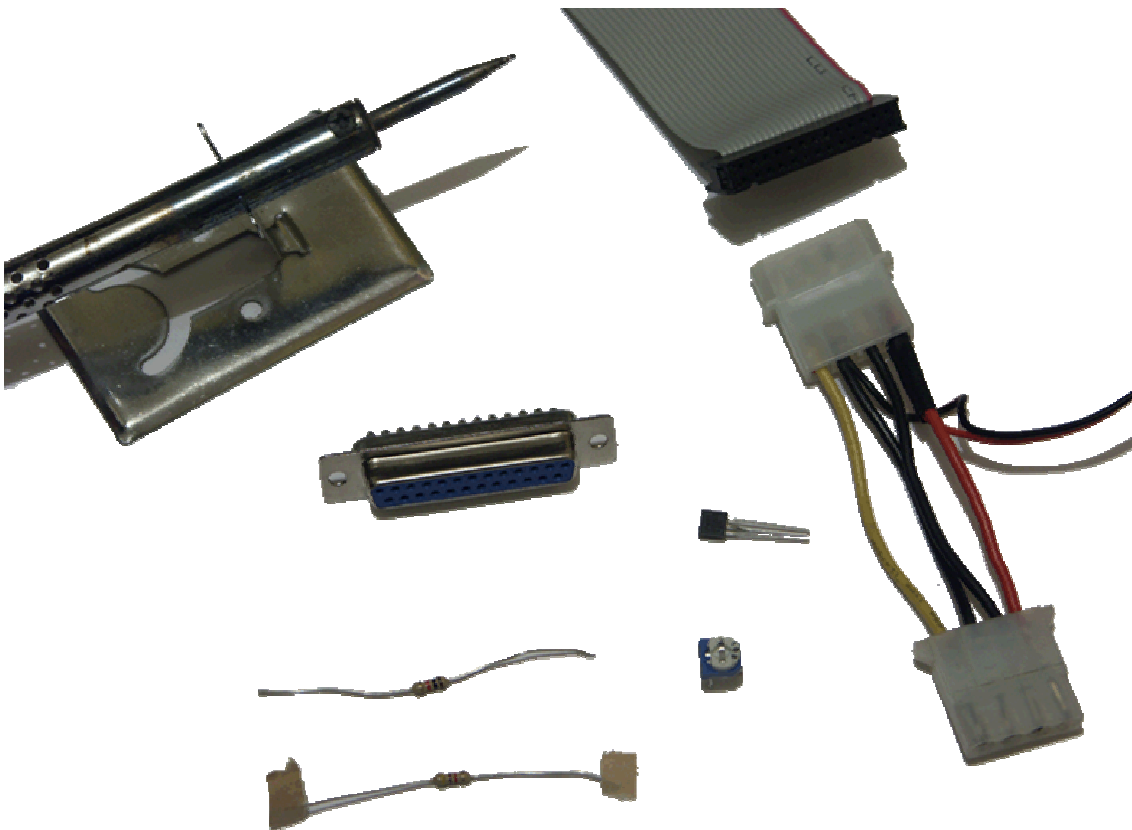


Figura 4.3. Unelte necesare pentru conectarea LCD-ului la interfața paralelă.

Uneltele necesare pentru conectarea fizică a LCD-ului la interfața paralelă sunt: pistol/ciocan de lipit, mufă paralelă, cablu multifilar (preferabil IDE) cu cel puțin 11 fire, rezistențe de 1K și 4.7K, potențiometru 10K și tranzistor BC327. În cazul în care mufa paralelă este mamă, va fi nevoie și de un cablu paralel tată \longleftrightarrow tată pentru a interfața montajul cu mufa mamă a portului paralel.

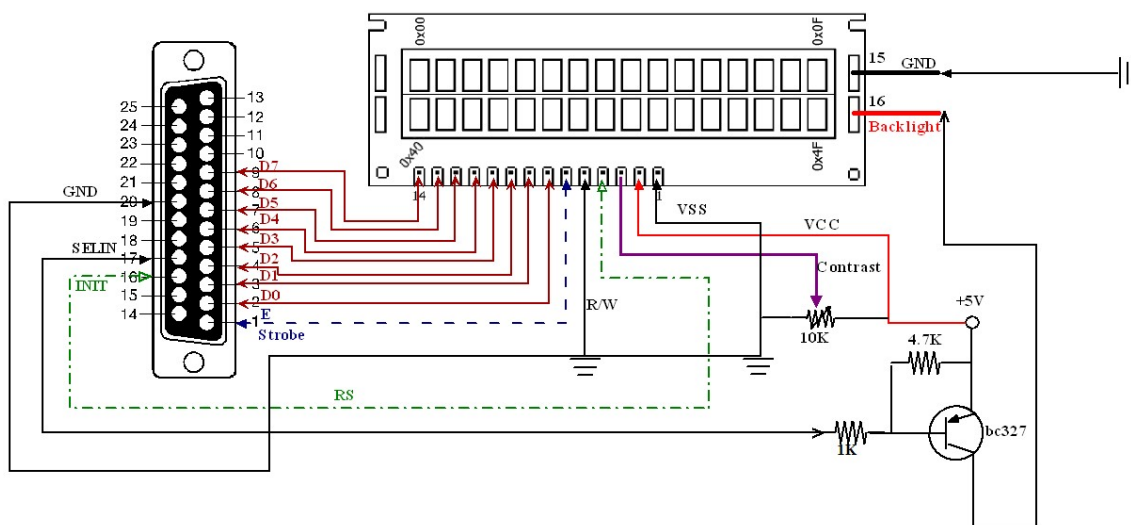


Figura 4.4. Schema de conectare a LCD-ului la interfața paralelă a calculatorului

Potențiometrul de 10Kohm se leagă la pinul 3 al LCD-ului pentru a ajusta contrastul acestuia. Activarea luminii de fundal (en. Backlight) se face printr-un montaj alcătuit din cele două rezistențe (1K și 4.7K) și tranzistorul BC327. În cazul în care se dorește activarea luminii se va pune pinul 17 al portului paralel pe 1 logic, iar în cazul în care se dorește dezactivarea pe 0 logic.

2.3. Setul de comenzi al HD44780

Setul de comenzi care va fi folosit în cadrul lucrării de față este prezentat în tabelul 4.2. Datorită faptului că pinul R/W a fost legat la GND, nu va fi posibilă folosirea comenzilor care citesc starea LCD-ului (prin setarea bitului R/W pe 1). Din acest motiv, tabelul 4.2 nu conține setul complet de comenzi al controlerului HD44780.

Un set complet de comenzi a HD44780, este prezent la: http://www.electronic-engineering.ch/microchip/datasheets/lcd/the_lcd_data_sheet.pdf.

Instrucțiune	RS	RW	D7	D6	D5	D4	D3	D2	D1	D0	Descriere
Șterge ecranul	0	0	0	0	0	0	0	0	0	1	Șterge ecranul și resetează contorul de adresă la zero.
Pune cursorul pe poziția [1,1]	0	0	0	0	0	0	0	0	1	x	Resetează contorul de adresă, poziționează cursorul pe poziția 1. Conținutul memoriei DDRAM rămâne neschimbat.
Setări deplasare cursor	0	0	0	0	0	0	0	1	I/D	S	Setează modul de deplasare al cursorului (I/D) precum și tipul de deplasare al ecranului (S).
Setări cursor/ecran	0	0	0	0	0	0	1	D	C	B	Setează ecranul on/off (D), cursorul on/off (C), și modul de clipire al cursorului (B).
Deplasare cursor	0	0	0	0	0	1	S/C	R/L	x	x	Mută cursorul (S/C) și specifică direcția (R/L).
Setări tip LCD	0	0	0	0	1	DL	N	F	x	x	Setează tipul LCD: nr biți (DL), nr linii (N) și tipul de font al caracterelor (F).
Setare adresă CGRAM	0	0	0	1	Adresa CGRAM					Setează adresa CGRAM. În continuare se vor trimite date în CGRAM.	
Setare adresă DDRAM	0	0	1	Adresa DDRAM					Setează adresa DDRAM. În continuare se vor trimite date DDRAM.		
Scrie date	1	0	Date					Scrie date în DDRAM sau CGRAM.			

Tabelul 4.2. Comenzile HD44780 folosite în cadrul lucrării

Semnificația biților de mai sus este prezentată în tabelul 4.3.

x : ne semnificativ	I/D	1	Incrementează	R/L	1	Deplasare la dreapta
		0	Decrementează		0	Deplasarea la stânga
	S	1	Deplasare automată a ecranului	DL	1	Interfață pe 8 bit
		0			Interfață pe 4 bit	
	D	1	Activare ecran (ON)	N	1	2 linii
		0	Dezactivare ecran (OFF)		0	1 linie
	C	1	Activare cursor(ON)	F	1	5x10 puncte
		0	Dezactivare cursor(OFF)		0	5x7 puncte
	B	1	Cursor în mod clipire	DDRAM : Display Data RAM		
		0				
	S/ C	1	Deplasare ecran	CGRAM : Character Generator RAM		
		0	Mutare cursor			

Tabelul 4.3. Semnificația biților din tabelul 4.2.

Correspondența între codurile numerice și caracterele ASCII ce pot fi afișate pe ecranul LCD este prezentată în figura 4.5.

	0 0 0 0 0 0 0 1 1 1 1 1 1 1
	0 0 0 1 1 1 1 0 0 1 1 1 1 1
	0 1 1 0 0 1 1 1 1 0 0 1 1
	0 0 1 0 1 0 1 0 1 0 1 0 1 0 1
xxxx0000	0 0 P \ P - 9 3 0 0
xxxx0001	! 1 A Q a q . 7 f 4 ä 9
xxxx0010	" 2 B R b r r i t z / ß 0
xxxx0011	# 3 C S c s j u t e s w
xxxx0100	\$ 4 O T d t \ i t t u 0
xxxx0101	% 5 E U e u . 7 1 0 ü
xxxx0110	& 6 F U f v 7 k c 3 p 2
xxxx0111	' 7 G W 9 w 7 7 7 7 q 0
xxxx1000	(8 H X h x i 7 7 7 j 7
xxxx1001) 9 I Y i y 7 7 7 j 7 u 7
xxxx1010	* : J Z j z e j 7 7 7 i 7
xxxx1011	+ ; K [k (7 7 7 0 * 7
xxxx1100	, < L 7 1 7 7 7 7 7 7
xxxx1101	- = M] m } 7 7 7 7 7 7
xxxx1110	. > N ^ n 7 7 7 7 7 7
xxxx1111	/ ? 0 _ o 7 7 7 7 7 7

Figura 4.5. Tabela ASCII a LCD-ului

Din figura 4.5 se observă că, codul simbolurilor caracter ce poate fi afișat pe LCD este similar cu codul ASCII al calculatoarelor personale. De aceea, codul caracterului de tipărit nu trebuie transmis sub forma binară, decât în cazul în care el nu se poate scrie altfel.



În mod normal, inițializarea LCD-ului se face automat la punerea sub tensiune. Pentru siguranță, se poate realiza inițializarea LCD-ului din program prin instrucțiuni. Exemple de inițializare a LCD-ului sunt date în secțiunea 2.4. Pentru mai multe informații a se consulta www.electronic-engineering.ch/microchip/datasheets/lcd/lcd_data_sheets.html.

2.4. Exemple de programare

În cele ce urmează, vor fi prezentate codurile sursă a câtorva exemple de programare a LCD-ului. Codul sursă conține comentarii în limba engleză.

2.4.1. Exemplu program LCD #1 – “Hello world”

```

:
:
; EPIP Labs copyleft (c) 2004 UTCluj
:-----
; Lab4P1 - LCD programming example 1 - "Hello world"
:
:
%include "system.inc"           ; include Asmutils

CODESEG

%define LPT1_DATA      0x378
%define LPT1_CONTROL  LPT1_DATA+2
%define DISPOFF       00001000b
%define DISPON        00001100b
%define CLEARDISPLAY  00000001b
%define LCD_5x7x2x8b  00111000b

%macro outputport 2
    mov dx, %1
    mov al, %2
    out dx, al
%endmacro

%macro lcdwrite 1
    ; enable SelIn (17) and Init (16)
    outputport LPT1_CONTROL, 00001100b    ; Backlight + RS enabled
    sys_nanosleep rqtp,NULL              ; sleep
    outputport LPT1_DATA, %1
    ; enable SelIn (17), Init (16) and Strobe (1)
    outputport LPT1_CONTROL, 00001101b
    sys_nanosleep rqtp,NULL              ; sleep
%endmacro

START:
; Open /dev/io for I/O port access
    sys_open path, 0_RDONLY
    jnc @_noerror
    jmp @_error
@_noerror:
; Initialising LCD (backlight off)

```

```

output LPT1_CONTROL, 00000000b
sys_nanosleep rqtp,NULL           ; sleep
output LPT1_DATA, LCD_5x7x2x8b    ; set 8bit, 2 lines, 5x7 dots
output LPT1_CONTROL, 00000001b
sys_nanosleep rqtp,NULL           ; sleep

output LPT1_CONTROL, 00000000b
sys_nanosleep rqtp,NULL           ; sleep
output LPT1_DATA, DISPOFF         ; turn off the display
output LPT1_CONTROL, 00000001b
sys_nanosleep rqtp,NULL           ; sleep

output LPT1_CONTROL, 00000000b
sys_nanosleep rqtp,NULL           ; sleep
output LPT1_DATA, CLEARDISPLAY    ; clear the LCD display
output LPT1_CONTROL, 00000001b
sys_nanosleep rqtp,NULL           ; sleep

output LPT1_CONTROL, 00000000b
sys_nanosleep rqtp,NULL           ; sleep
output LPT1_DATA, DISPON         ; turn on the display
output LPT1_CONTROL, 00000001b
sys_nanosleep rqtp,NULL           ; sleep

```

; Writing to LCD

```

lcdwrite 10100101b                ; see the ascii code table
lcdwrite "E"
lcdwrite "P"
lcdwrite "I"
lcdwrite "P"
lcdwrite "-"
lcdwrite "L"
lcdwrite "a"
lcdwrite "b"
lcdwrite "4"
output LPT1_CONTROL, 00001000b
sys_nanosleep rqtp,NULL           ; sleep
output LPT1_DATA, 11000000b      ; set DDRAM addr to 2nd line
output LPT1_CONTROL, 00001001b
sys_nanosleep rqtp,NULL           ; sleep
lcdwrite "H"
lcdwrite "e"
lcdwrite "l"
lcdwrite "l"
lcdwrite "o"

```

```

    lcdwrite " "
    lcdwrite "w"
    lcdwrite "o"
    lcdwrite "r"
    lcdwrite "l"
    lcdwrite "d"
    sys_exit 0
@_error:
    sys_write STDOUT, errmsg, errlen
    sys_exit 1

rqtp I_STRUC timespec
.tv_sec      _LONG 0
.tv_nsec    _LONG 100
I_END

DATASEG
errmsg      db 'Error setting I/O permissions!','_n
errlen     equ $-errmsg
path       db '/dev/io',EOL
END

```

Programul prezentat mai sus va inițializa LCD-ul conectat pe interfața paralelă a calculatorului (dacă adresa LPT diferă de 0x378, se va modifica și recompila programul), urmând a afișa apoi textul “•EPIP-Lab4” pe prima linie a LCD-ului respectiv “Hello world” pe cea de-a doua linie.

Compilarea și linkeditarea programului se realizează prin folosirea comenzilor:

```

# nasm -f bin -D__FREEBSD__ -D__OPTIMIZE__=__O_SIZE__ -D__ELF_MACROS__ \
    -D__ELF__ -o epip_l4p1 epip_l4p1.asm
# chmod +x epip_l4p1

```

2.4.2. Exemplu program LCD #2 – Afișarea unui text pe ecran pe ambele linii

```

;
; EPIP Labs copyleft (c) 2004 UTCluj
;-----
; Lab4P2 - LCD programming example 2 - split/write a string on both lines
;
;
%include "system.inc"           ; include Asmutils

CODESEG

%define LPT1_DATA      0x378

```

```

#define LPT1_CONTROL    LPT1_DATA+2
#define DISPOFF        00001000b
#define DISPON         00001100b
#define CLEARDISPLAY   00000001b
#define LCD_5x7x2x8b   00111000b

%macro output 2
    mov dx, %1
    mov al, %2
    out dx, al
%endmacro

%macro writestring 2
    mov ecx, %2
    mov esi, %1
    call lcdwrite
%endmacro

%macro LCDenable 0
; backlight on
    output LPT1_CONTROL, 00001000b
    sys_nanosleep rntp,NULL    ;sleep
    output LPT1_CONTROL, 00001001b
    sys_nanosleep rntp,NULL    ;sleep
%endmacro

START:
    sys_open path, 0_RDONLY
    jnc @_noerror
    jmp @_error
@_noerror:
; Initialising LCD
    output LPT1_DATA, LCD_5x7x2x8b
    LCDenable
    output LPT1_DATA, DISPOFF
    LCDenable
    output LPT1_DATA, CLEARDISPLAY
    LCDenable
    output LPT1_DATA, DISPON
    LCDenable
; Writing to LCD
    writestring message1, len1
    writestring message2, len2
    sys_exit 0
@_error:

```

```

sys_write STDOUT, errmsg, errlen
sys_exit 1

lcdwrite:
; Expects: ECX= number of bytes to write
;          ESI = string to write
    mov dx, cx          ; save length of string
    cmp cx, 16         ; more than one line ?
    jbe oneline
    sub dx, 16         ; nr of chars on the second line
    push dx            ; save it on the stack
    mov cx, 16
    call writeline     ; write first line on the LCD (16 chars)
    outport LPT1_DATA, 11000000b
    LCDenable
    pop cx             ; load from the stack
oneline:
    call writeline
    ret

writeline:
; Expects: ECX = number of bytes to write
;          ESI = string to write
    push cx
    lodsb              ; DS:SI -> AL
    xchg bl, al        ; put it in BL and call writechar
    call writechar
    pop cx
    loop writeline
    ret

writechar:
; enable SelIn (17) and Init (16)
    outport LPT1_CONTROL, 00001100b ; Backlight + RS enabled
    sys_nanosleep rntp, NULL ; sleep
    outport LPT1_DATA, bl ; BL holds the data
; enable SelIn (17), Init (16) and Strobe (1)
    outport LPT1_CONTROL, 00001101b
    sys_nanosleep rntp, NULL ; sleep
    ret

rntp I_STRUC timespec
.tv_sec    _LONG 0
.tv_nsec   _LONG 100
I_END

```

```

DATASEG
message1      db "0123456789abcdefghijklmnop"
len1          equ $-message1
message2      db " Hello "
len2          equ $-message2
errmsg       db 'Error setting I/O permissions!','\n'
errlen       equ $-errmsg
path         db '/dev/io',EOL
END

```

Ca și exemplul precedent, programul inițializează întâi LCD-ul, iar apoi va scrie un text pe ecran prin folosirea macro-ului *writestring*. Dacă textul de afișat depășește dimensiunea unei linii (16 caractere), se vor scrie primele 16 caractere din text pe prima linie, continuarea acestuia urmând a fi scrisă pe cea de-a doua linie a LCD-ului.

Compilarea și linkeditarea programului se realizează prin folosirea comenzilor:

```

# nasm -f bin -D__FREEBSD__ -D__OPTIMIZE__=__O_SIZE__ -D__ELF_MACROS__ \
        -D__ELF__ -o epip_l4p2 epip_l4p2.asm
# chmod +o epip_l4p2

```

2.4.3. Exemplu program LCD #3 – “Centrarea” unui text

```

;
; EPIP Labs copyleft (c) 2004 UTCluj
;-----
; Lab4P3 - LCD programming example 3 - write a string centered
;
;
%include "system.inc"           ; include Asmutils

CODESEG

%define LPT1_DATA      0x378
%define LPT1_CONTROL  LPT1_DATA+2

%define DISPOFF       00001000b
%define DISPON        00001100b
%define CLEARDISPLAY  00000001b
%define LCD_5x7x2x8b  00111000b

%macro outport 2
    mov dx, %1
    mov al, %2
    out dx, al
%endmacro

```

```

%macro LCDenable 0
; backlight on
    output LPT1_CONTROL, 00001000b
    sys_nanosleep rntp,NULL          ;sleep
    output LPT1_CONTROL, 00001001b
    sys_nanosleep rntp,NULL          ;sleep
%endmacro

%macro setcolumn 1
    push cx
    mov al, 10000000b                ; set DDRAM to point to the first row
    mov bl, %1                        ; set column
    add bl, al
    dec bl
    output LPT1_DATA, bl
    LCDenable
    pop cx
%endmacro

START:
    pop eax                          ; get argc
    cmp eax, 1                        ; more than one parameter?
    ja gotparam
    sys_write STDOUT, oneparam, lenparam
    sys_exit 1                        ; exit with error

gotparam:
    pop ecx                          ; get argv[0]
    sys_open path, O_RDONLY           ; open /dev/io for I/O access
    jnc @_noerror
    jmp @_error

@_noerror:
; Initialising LCD
    output LPT1_DATA, LCD_5x7x2x8b
    LCDenable
    output LPT1_DATA, DISPOFF
    LCDenable
    output LPT1_DATA, CLEARDISPLAY
    LCDenable
    output LPT1_DATA, DISPON
    LCDenable
    output LPT1_DATA, 10000000b      ; DDRAM address -> 1,1
    LCDenable
; Writing to LCD

```

```

    pop esi                ; get argv[1]
    mov ecx, esi          ; mark the beginning
loopfind:
    inc esi                ; check each char
    cmp byte [esi], 0     ; end of string?
    jne loopfind
    xchg ecx, esi         ; argv[1] back in esi
    sub ecx, esi          ; length in ecx
writeline:
; ESI = string to write
; ECX = length of string
    xor ax, ax            ; clear AX
    mov al, 16            ; size of one line
    sub al, cl            ; 16-(size of string)
    mov bl, 2
    div bl                ; al = (16-(sizeofstring))/2
    xchg al, dl           ; dl = (16-size)/2+1
    inc dl                ; dl = (16-size)/2+1
    setcolumn dl         ; set the cursor to the specified column
.writelineloop:
    push cx
    lodsb                 ; DS:SI -> AL
    xchg bl, al           ; put it in BL and call writechar
    call writechar
    pop cx
    loop .writelineloop
    setcolumn 1          ; set the cursor to 1,1
    sys_exit 0
@_error:
    sys_write STDOUT, errmsg, errlen
    sys_exit 1

writechar:
; enable SelIn (17) and Init (16)
    outport LPT1_CONTROL, 00001100b ; Backlight + RS enabled
    sys_nanosleep rqtp, NULL         ; sleep
    outport LPT1_DATA, bl           ; BL holds the data
; enable SelIn (17), Init (16) and Strobe (1)
    outport LPT1_CONTROL, 00001101b
    sys_nanosleep rqtp, NULL         ; sleep
    ret
rqtp I_STRUC timespec
.tv_sec      _LONG 0
.tv_nsec     _LONG 2000000
I_END

```

```

DATASEG
oneparam      db 'Syntax is: ./l4p3 <parameter>',__n
lenparam      equ $-oneparam
errmsg        db 'Error setting I/O permissions!,'__n
errlen        equ $-errmsg
path          db '/dev/io',EOL
END

```

Programul va inițializa LCD-ul, iar apoi va afișa centrat pe prima linie a ecranului primul parametru transmis programului. În cazul în care nu se transmite nici un parametru se va afișa un mesaj de eroare. În cazul în care lungimea parametrului depășește dimensiunea maximă a unei linii ecran LCD (16 caractere), se vor afișa doar primele 16 caractere din text.

Compilarea și linkeditarea programului se realizează prin folosirea comenzilor:

```

# nasm -f bin -D__FREEBSD__ -D__OPTIMIZE__=__O_SIZE__ -D__ELF_MACROS__ \
        -D__ELF__ -o epip_l4p3 epip_l4p3.asm
# chmod +o epip_l4p3

```

2.4.4. Exemplu program LCD #4 – “Mașina de scris”

```

;
; EPIP Labs copyleft (c) 2004 UTCluj
;-----
; Lab4P4 - LCD programming example 4 - STDIN read, LCD write
;
;
%include "system.inc"          ; include Asmutils

CODESEG

%define LPT1_DATA      0x378
%define LPT1_CONTROL  LPT1_DATA+2

%define DISPOFF        00001000b
%define DISPCURBLINKON 00001111b
%define CLEARDISPLAY  00000001b
%define LCD_5x7x2x8b  00111000b

%macro outport 2
    mov dx, %1
    mov al, %2
    out dx, al
%endmacro

```

```

%macro LCDenable 0
; backlight on
    output LPT1_CONTROL, 00001000b
    sys_nanosleep rqtp,NULL           ;sleep
    output LPT1_CONTROL, 00001001b
    sys_nanosleep rqtp,NULL           ;sleep
%endmacro

%macro writechar 1
; enable SelIn (17) and Init (16)
    output LPT1_CONTROL, 00001100b   ; Backlight + RS enabled
    sys_nanosleep rqtp,NULL           ; sleep
    output LPT1_DATA, byte [%1]
; enable SelIn (17), Init (16) and Strobe (1)
    output LPT1_CONTROL, 00001101b
    sys_nanosleep rqtp,NULL           ; sleep
%endmacro

START:
    sys_open path, O_RDONLY           ; open /dev/io for I/O access
    jnc @_noerror
    jmp @_error
@_noerror:
; Initialising LCD
    output LPT1_DATA, LCD_5x7x2x8b
    LCDenable
    output LPT1_DATA, DISPOFF
    LCDenable
    output LPT1_DATA, CLEARDISPLAY
    LCDenable
    output LPT1_DATA, DISPCURBLINKON
    LCDenable
    output LPT1_DATA, 10000000b
    LCDenable
; Writing to LCD
    mov byte [row], 10000000b         ; enable first line
    mov byte [col], 0
@loop:
    cmp byte [col], 16                 ; end of line?
    jb write
    xor byte [row], 01000000b         ; switch line
    output LPT1_DATA, [row]
    LCDenable
    mov byte [col], 0                 ; set to first column

```

```

write:
    inc byte [col]
    sys_read STDIN, buf, nbyte      ; read from STDIN
    cmp byte [buf], 0xa            ; check for ENTER
    je exit                        ; exit if found
    writechar buf                  ; writechar if not
    jmp @loop                       ; loop
exit:
    sys_exit 0

@_error:
    sys_write STDOUT, errmsg, errlen
    sys_exit 1

nqtp I_STRUC timespec
.tv_sec      _LONG 0
.tv_nsec     _LONG 2000000
I_END

DATASEG
errmsg       db 'Error setting I/O permissions!','_n
errlen       equ $-errmsg
path         db '/dev/io',EOL

UDATASEG
col          resb 1
row          resb 1
buf          resb 1
nbyte       equ $-1
END

```

Programul de mai sus va funcționa aproape ca o mașină de scris, în sensul că se vor citi caractere introduse de la tastatură (și validate de tasta Enter) și se vor afișa pe ecranul LCD. În momentul în care se ajunge în capătul unei linii, se va trece automat pe linia următoare. Execuția programului se termină la apăsarea tastei Enter pe un rând gol.

Compilarea și linkeditarea programului se realizează prin folosirea comenzilor:

```

# nasm -f bin -D__FREEBSD__ -D__OPTIMIZE__=__O_SIZE__ -D__ELF_MACROS__ \
        -D__ELF__ -o epip_l4p4 epip_l4p4.asm
# chmod +o epip_l4p4

```



3. Desfășurarea lucrării

Se vor introduce, asambla, linkedita, și rula toate exemplele prezentate în secțiunea 2.4.

Se va modifica exemplul 2.4.4 astfel încât caracterele să fie scrise pe ecran în ordine inversă. Se va începe din **[16,2]** urmând ca în momentul în care se ajunge în **[1,2]** cursorul să treacă pe poziția **[16,1]**. Similar, în momentul în care cursorul se află în poziția **[1,1]**, următorul caracter tipărit va fi afișat pe **[16,2]**.

Se va modifica exemplul 2.4.3 prin adăugarea unui efect de *scroll* la stânga și la dreapta a textului afișat pe ecran.



4. Concluzii

Pe parcursul lucrării, s-au punctat următoarele aspecte:

- ❶ Display-urile LCD bazate pe HD44780 se conectează pe portul paralel folosind semnalele de date D0-D7, respectiv semnalele de control Init și SelIn.
- ❷ Pentru ajustarea contrastului LCD se poate folosi un potențiomtru de 10K, iar pentru activarea manuală a luminii de fundal (en. Backlight) este necesară o schemă cu un tranzistor BC327 și două rezistențe de 1K și 4.7K.
- ❸ LCD-ul se inițializează automat la punerea sub tensiune, dar se poate inițializa și manual (vezi exemplele din secțiunea 2.4).
- ❹ Atât din programele exemplu, cât și din documentația controlerului HD44780 reiese faptul că sunt necesare mici pauze după transmiterea fiecărei comenzi din calculator către controler, pentru ca acesta din urmă să le poată procesa și executa.



5. Anexe

- ▶ LCD Info page - <http://www.geocities.com/dinceraydin/lcd/index.html>
- ▶ LCDProc - <http://lcdproc.org/?continue=yes>
- ▶ LCD Information&Technical forum - www.eio.com/lcdintro.htm
- ▶ Hantronix Datasheets LCDs - http://www.hantronix.com/2_2.html
- ▶ HD44780-Based LCD Modules - <http://www.doc.ic.ac.uk/~ih/doc/lcd/>
- ▶ LCD HD44780 HOWTO - www.myrolypoly.com/lcd_project/lcd_project.html