

# LABORATOR 5 - Afișor cu cristale lichide (LCD) și ADC

## Prezentarea afișorului cu cristale lichide.

Orice afișor LCD este alcătuit din două componente:

- Interfața propriu zisă.
- Matricea de puncte din cristale lichide (dot-matrix liquid crystal display)

Controlerul IO are codul HD44780. Circuitul integrat HD44780 este cel mai popular controler pentru afișoare cu cristale lichide, 99% din controlerile prezente astăzi pe piață fiind compatibile cu acesta.

Datele afișate pe LCD sunt memorate în memoria internă a controlerului. Această memorie se numește DDRAM - Display Data RAM și are capacitatea de 80 de octeți.

Controlerul poate comanda un LCD cu una sau două linii. Lungimea maximă a unei linii este de 40 de caractere. Afișorul LCD de care dispunem la acest laborator poate afișa **două linii**, fiecare linie având **lungimea de 16 caractere**. Când controlerul este configurat astfel încât să se utilizeze o singură linie, se va afișa numai o linie, cea de a doua nefiind utilizată.

Modul în care informația din DDRAM este afișată pe LCD în cazul configurării pe două linii este:

Tabel 1

	Coloana	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Linia 1	Adresa DDRAM	00h	01h	02h	03h	04h	05h	06h	07h	08h	09h	0Ah	0Bh	0Ch	0Dh	0Ef	0Fh
	Conținut DDRAM	48h	65h	6ch	6ch	6fh	20h	20h	20h	20h	20h	20h	20h	20h	20h	20h	20h
	Ce se vede	H	e	l	l	o											
Linia 2	Ce se vede						w	o	r	l	d	!	_				
	Conținut DDRAM	20h	20h	20h	20h	20h	20h	77h	6fh	72h	6ch	64h	21h	20h	20h	20h	20h
	Adresa DDRAM	40h	41h	42h	43h	44h	45h	46h	47h	48h	49h	4Ah	4Bh	4Ch	4Dh	4Ef	4Fh

Caracterele din tabelul anterior sunt **codificate ASCII**. De exemplu, caracterul ,H' are valoarea 48h (0x48).

**Cursorul** ( \_ ) indică poziția în care se va face următoarea scriere. Poziția în care se va face următoarea scriere se va numi în continuare poziția curentă. Cursorul este vizibil sau nu, în funcție de setare.

**Schema** de cuplare a afișorului LCD și **proiectul** ce conține funcțiile de pentru lucru cu LCD se află în platforma lucrării de laborator.

Sunt disponibile următoarele funcțiile de lucru cu LCD-ul:

<b>void initLCD()</b>	Configurare controler LCD. Această funcție setează numărul de linii, tipul fontului, forma cursorului. De asemenea configurează portul D pentru conectarea cu LCD-ul. Această funcție se apelează o singură dată în secțiunea de inițializare (înainte de <code>while(1)</code> )
<b>void putchLCD(char ch)</b>	Scrie caracterul <b>ch</b> la poziția cursorului și mută cursorul la dreapta.

	Exemplu: <code>putchLCD('A');</code>
<code>void putsLCD(char *ch)</code>	<p>Funcția scrie pe LCD caracterele începând de la adresa specificată (<b>ch</b>) până când ajunge la caracterul de terminare (<code>,\0'</code>). Acest caracter nu este copiat.</p> <p>Caracterele sunt scrise începând cu poziția cursorului. După încheierea apelului cursorul se poziționează după ultimul caracter scris.</p> <p>Exemplu: <code>putsLCD("Hello");</code></p>
<code>void clrLCD()</code>	Șterge conținutul LCD și mută cursorul pe prima linie, prima coloană.
<code>void gotoLC(unsigned char line, unsigned char col)</code>	<p>Mută cursorul în pe linia <b>line</b>, coloana <b>col</b>. Conținutul LCD nu se modifică. Valorile valide pentru parametrul <b>line</b> sunt 1 și 2. Valorile valide pentru parametrul <b>col</b> sunt în intervalul [1, 16].</p> <p>Dacă vreunul din parametrii <b>line</b> sau <b>col</b> este invalid, comanda va fi ignorată. De exemplu <code>gotoLC(3,1)</code> este invalidă și nu va produce efecte.</p>

Atenție: Este posibil ca scrierea unui caracter sau a unui șir de caractere să pară că nu se face corect. Să presupunem ca cursorul se află pe linia 1 coloana 15, ca în tabelul următor:

**Tabel 2**

	Coloana	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16					
Linia 1	Adresa DDRAM	00h	01h	02h	03h	04h	05h	06h	07h	08h	09h	0Ah	0Bh	0Ch	0Dh	0Eh	0Fh	10h	11h	12h	...	27h
	Ce se vede																					
Linia 2	Ce se vede																					
	Adresa DDRAM	40h	41h	42h	43h	44h	45h	46h	47h	48h	49h	4Ah	4Bh	4Ch	4Dh	4Eh	4Fh	50h			...	67h

După apelul `putsLCD(„Hello”);` LCD-ul va arăta ca în tabelul următor:

**Tabel 3**

	Coloana	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16					
Linia 1	Adresa DDRAM	00h	01h	02h	03h	04h	05h	06h	07h	08h	09h	0Ah	0Bh	0Ch	0Dh	0Eh	0Fh	10h	11h	12h	...	27h
	Ce se vede																					
Linia 2	Ce se vede																					
	Adresa DDRAM	40h	41h	42h	43h	44h	45h	46h	47h	48h	49h	4Ah	4Bh	4Ch	4Dh	4Eh	4Fh	50h			...	67h

Nu trebuie uitat că afișorul LCD propriu zis are 16 coloane dar controlerul are 40 de coloane. Controlerul nu știe câte coloane există fizic și nici nu există vreo modalitate de a-i spune. Așa că după scrierea caracterului ,e' la adresa 0Fh următorul caracter, l, se va scrie la adresa 10h. Adresa 0Fh se afișează iar adresa 10h, nu. De ce? Pentru ca nu există afișaj fizic pentru adresa 10h!

În concluzie, dacă apar mesaje trunchiate înseamnă că ați depășit dimensiunea afișorului fizic.

Proiectul Microchip Studio conține trei fișiere:

**LCD.c** conține toate funcțiile prezentate anterior.

**LCD.h** conține prototipurile funcțiilor din **LCD.c** precum și definiții utile.

**test\_LCD.c** Va conține codul care testează funcțiile LCD. Inițial conține scheletul programului de test.

### Pasul 1: **putchLCD(char ch)**

Pentru a testa funcția `putchLCD(char ch)` fișierul **test\_LCD.c** va avea următoarea structură:

```
#include "LCD.h"
#include <avr/io.h>

int main() {

    initLCD();
    putchLCD('H');

    while(1) {
    }
}
```

Dacă adăugați mai multe apeluri `putchLCD` puteți scrie „Hello Micro”. Nu e obligatoriu.

### Pasul 2: **clrLCD()**

Pentru a testa funcția `clrLCD()` scriem un caracter, așteptăm o secundă cu funcția `_delay_ms(...)`, ștergem display-ul cu `clrLCD()` și în final așteptăm 0,5 s. Structura buclei principale din programului de test este:

```
...
while(1) {
    //scrie caracter
    _delay_ms(500);
    //șterge display
    _delay_ms(500);
}
...
```

### Pasul 3: **putsLCD(char string[ ])**

Folosind `putchLCD` scrieți funcția `putsLCD` – scrie șir de caractere. Primul caracter al mesajului se va scrie la poziția cursorului. Adăugați la programul de test de la pasul 2 o secțiune care să afișeze mesajul „Hello Micro”! Structura buclei principale din programului de test este:

```
...
while(1) {
    //scrie caracter
    _delay_ms(1000);
    //șterge display
    _delay_ms(500);
    //Afișează mesajul „Hello micro!”
    _delay_ms(1000);
}
...
```

## Pasul 4: gotoLC

Funcția `putsLCD` scrie mesajul începând de la poziția curentă a cursorului. Pentru a putea scrie mesaje pe orice linie, începând cu orice colană, este nevoie de o funcție care să mute cursorul. Această funcție se numește `gotoLC` și va avea ca parametrii pe `line` și pe `col`:

```
void gotoLC(unsigned char line, unsigned char col);
```

unde *line* este linia pe care se va poziționa cursorul iar *col* este coloana.

Modificați programul de la pasul 3 astfel încât mesajul să apară ca în Tabel 1. Păstrați afișarea caracterului.

**Verificare intermediară: Când funcționează chemați profesorul.**

## Pasul 5: Testare complexă

Se vor adăuga testele scrise cu roșu, după afișare caracter și string:

```
...
while(1){
    //scrie caracter
    _delay_ms(1000);
    //șterge display
    _delay_ms(500);
    //Afișează mesajul „Hello micro!” ca în Tabel 1
    _delay_ms(1000);

    // Din linia 1, coloana 1 scrie caracter cu caracter textul 0123456789abcdef.
    // După scrierea fiecărui caracter așteaptă 0.5 secunde.

    // Din linia 2, coloana 1 scrie mesajul „ghijklmnopqrstuv”

    // Mută cursorul pe linia 2, coloana 16.

    //Mută cursorul către stânga caracter cu caracter până pe linia 2, coloana 1.
    // După fiecare mutare așteaptă 0.5 secunde.

    // Mută cursorul pe linia 1, coloana 16.

    //Mută cursorul către stânga caracter cu caracter până pe linia 1, coloana 1.
    // După fiecare mutare așteaptă 0.5 secunde.

    // Așteaptă 0.5 secunde.

    // Șterge LCD
}
...
```

Pentru a înțelege mai bine modul de testare încărcați în microcontroler fișierul [exemplu.hex](#) din arhiva laboratorului. **NU** implementați ultimul mesaj, "Exemplu creat de profesor".

**Verificare finală: Când funcționează, apelați profesorul pentru validare.**