

# LABORATOR 6 - Interfață pentru afișor cu cristale lichide (LCD), partea II-a

## Scopul lucrării

În laboratorul anterior s-au implementat funcțiile `wr_LCDreg(char vRS, char data)`, `void initLCD()` și `sysinit()`.

În continuare se vor implementa și testa următoarele funcții:

<code>void putchLCD(char ch)</code>	scrie caracter
<code>void putsLCD(char *ch)</code>	scrie șir de caractere
<code>void clrLCD()</code>	șterge LCD și mută cursorul în prima poziție
<code>void gotoLC(unsigned char line, unsigned char col)</code>	mută cursorul

Corpul acestor funcții se va scrie în fișierul `IOfn.c`, prototipul lor se va adăuga în fișierul `defs.h` și se vor apela din `LCDtest.c`

În secțiunea opțională se vor implementa și alte funcții.

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

Scrieți funcția `putchLCD` în `IOfn.c`. Pentru ca un caracter să apară pe LCD la poziția cursorului scrieți respectivul caracter în registrul **DRegWr**. Nu uitați ca după scriere să așteptați 50 uS.

Apoi modificați `LCDtest.c`. Pentru a scrie un caracter `LCDtest.c` va avea următoarea structură:

```
int main() {  
  
    sysinit();  
    putchLCD('H');  
  
    while(1) {  
    }  
}
```

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

## Pasul 2: `putsLCD(...)`

Folosind `putchLCD` scrieți funcția `putsLCD` – scrie șir de caractere. Primul caracter al mesajului se va scrie la poziția cursorului. Cu un singur apel `putsLCD`, scrieți mesajul „Hello Micro”!

Informație utilă: în C un șir de caractere se termină `0x00 = '\0'`.

## Pasul 3: `clrLCD()`

Scrieți funcția `clrLCD`. Folosiți instrucțiunea „Clear Display”. Pentru a testa funcționarea acestei comenzi, scriem un mesaj, așteptăm 1-2 secunde cu funcția `wait(...)` și apoi ștergem display-ul cu `clrLCD`. Structura programului de test este:

```

...
int main() {
    sysinit();

    while(1) {
        putsLCD("Hello Micro!");
        wait(500*MS);
        clrLCD();
        wait(500*MS);
    }
}

```

#### Pasul 4: gotoLC

Funcția *putsLCD* pe care ați scris-o anterior 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 va numi *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.

**Implementarea funcției se va baza pe comanda „Set DDRAM Address”.** Analizați instrucțiunile LCD din tabelul 3, laboratorul 5. Se observă că pentru fiecare instrucțiune o parte din biții cei mai semnificativi au valoare constantă. De exemplu, pentru instrucțiunea LCD „Function set” biții 7:5 au întotdeauna valoarea „001”. Putem considera că biții constanți ai instrucțiunii LCD sunt opcode-ul instrucțiunii.

Identificați opcode-ul instrucțiunii „Set DDRAM Address”. **Toate instrucțiunile „Set DDRAM Address” trebuie să conțină acest opcode!**

După opcode în instrucțiunea LCD „Set DDRAM Address” urmează biții adresei DDRAM. Pentru a afla adresa DDRAM corespunzătoare unei locații LCD **analizați** următoarele informații din figura următoare: **Linia 1, Linia 2, Coloana** și **Adresa DDRAM**.

**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	0Eh	0Fh	10h	11h	...	27h
	Conținut DDRAM	20h	20h	20h	20h	20h	20h	20h	20h	20h	20h	20h	20h	20h	20h	20h	35h	36h	20h		20h
	Ce se vede																5	6	–		
Linia 2	Ce se vede																				
	Conținut DDRAM	20h	20h	20h	20h	20h	20h	20h	20h	20h	20h	20h	20h	20h	20h	20h	20h	20h	20h		20h
	Adresa DDRAM	40h	41h	42h	43h	44h	45h	46h	47h	48h	49h	4Ah	4Bh	4Ch	4Dh	4Eh	4Fh	50h	51h	...	67h

Parametrul *line* poate lua valorile 1 sau 2. Parametrul *col* ia valori în intervalul [1, 16]. Dacă vreunul din parametrii *line* și *col* sunt invalizi, comanda va fi ignorată. De exemplu *gotoLC*(3,1) este invalidă și nu va produce efecte.

În esență trebuie **să scrieți codul care calculează adresa DDRAM** corespunzătoare unei poziții LCD identificată prin *line* și *col*. De exemplu, dacă *line* = 1, *col* = 10 adresa este 09. Dacă *line* = 2, *col* = 5 adresa este 0x44. Analizați Tabel 1!

Testați funcția cu secvența:

```

...

int main() {
    unsigned char k;

    sysinit();

```

```

while(1) {
    putsLCD("0123456789abcdef");
    gotoLC(2,1);
    putsLCD("ghijklmnopqrstuv");

    for(k=16;k>0;k--) {
        gotoLC(2,k);
        wait(500*MS);
    }

    for(k=16;k>0;k--) {
        gotoLC(1,k);
        wait(500*MS);
    }

    clrLCD();
    wait(500*MS);
}
}

```

**Când funcționează, apăsați profesorul pentru validare.**