

LCD (*Liquid Crystal Display*)

Veliki broj sistema baziranih na mikrokontroleru koristi LCD za vizuelni prikaz izlaza. Standardni LCD-i mogu prikazati ASCII set karaktera, kao i japanske, grčke i matematičke simbole. Za rad u 8-bitnom modu displej zahtijeva pinove za masu i napajanje kao i 11 ulazno/izlaznih (I/O) pinova, dok je za rad u 4-bitnom modu potrebno 7 I/O pinova. Preko tri I/O pina LCD razmjenjuje sa mikrokontrolerom kontrolne signale:

- **E (Enable)** - Ukoliko je E='0', LCD ne prihvata ostale kontrolne signale. Ukoliko je E='1', LCD se ponaša u skladu sa stanjem ostalih kontrolnih linija.
- **R/W (Read/Write)** - Kontrolni signal koji određuje smjer komunikacije između LCD-a i mikrokontrolera. Ukoliko je R/W='0', podaci se upisuju u RAM LCD-a. Ukoliko je R/W='1', podaci se iščitavaju iz RAM-a LCD-a.
- **RS (Register Select)** - Kontrolni signal kojim se LCD-u ukazuje na tip podatka na liniji (instrukcija ili karakter). Ukoliko je RS='0', u pitanju je instrukcija, ukoliko je RS='1', u pitanju je karakter.

Primjer:

Uz pomoć razvojne mikrokontrolerske ploče LV24-33 realizovati aplikaciju koja na LCD-u ispisuje poruku "hello".

Da bi bilo moguće uputiti LCD-u instrukciju ili karakter za ispis, najprije je potrebno izvršiti inicijalizaciju displeja:

```
sbit LCD_RS at LATD2_bit;
sbit LCD_EN at LATD3_bit;
sbit LCD_D4 at LATD4_bit;
sbit LCD_D5 at LATD5_bit;
sbit LCD_D6 at LATD6_bit;
sbit LCD_D7 at LATD7_bit;

sbit LCD_RS_Direction at TRISD2_bit;
sbit LCD_EN_Direction at TRISD3_bit;
sbit LCD_D4_Direction at TRISD4_bit;
sbit LCD_D5_Direction at TRISD5_bit;
sbit LCD_D6_Direction at TRISD6_bit;
sbit LCD_D7_Direction at TRISD7_bit;

Lcd_Init();
```

Brisanje teksta ispisanog na LCD-u može se izvršiti upućivanjem odgovarajuće komande displeju:

```
Lcd_Cmd(_Lcd_Clear);
```

Sljedeći nekoliko korisnih komandi koje se mogu uputiti LCD-u.

<code>_LCD_FIRST_ROW</code>	Prelazak kursora u prvi red
<code>_LCD_SECOND_ROW</code>	Prelazak kursora u drugi red
<code>_LCD_CLEAR</code>	Brisanje sadržaja prikazanog na LCD-u
<code>_LCD_CURSOR_OFF</code>	Isključenje kursora
<code>_LCD_TURN_ON</code>	Uključenje LCD-a
<code>_LCD_TURN_OFF</code>	Isključenje LCD-a

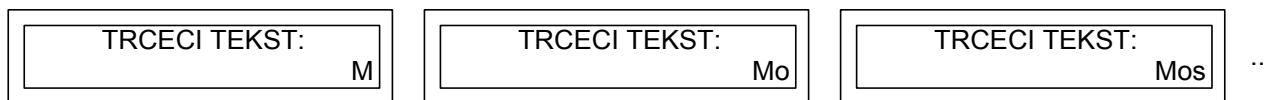
Ispis poruke "hello" može se ostaviti na sljedeći način:

```
Lcd_Out(1, 1, "hello");
```

Prvi argument funkcije `Lcd_Out(unsigned int, unsigned int, char *)` odnosi se na red displeja u kojem će biti ispisan tekst. Drugi argument odnosi se na poziciju u zadatom redu od koje će se ispisivati tekst, dok se treći argument odnosi na tekst koji se želi ispisati.

Zadatak:

Pomoću mikrokontrolerske razvojne ploče LV24-33 realizovati uređaj koji će na LCD-u prikazivati "trčeći" tekst *"Most microcontroller devices are using 'smart LCD' displays to output visual information."*, kako je prikazano na slici 1.



Slika 1

Prekidi (*interrupts*)

Prekidi (*interrupts*) su neočekivani događaji u toku sekvencijalnog izvršavanja instrukcija koji prekidaju regularan tok programa. Uzroci prekida su različiti. Kod PIC24FJ96GA010 postoji 43 moguća izvora prekida, čiji prioriteti su definisani tabelom - IVT (*Interrupt Vector Table*). Ova tabela smještena je u programskoj memoriji, sa startnom lokacijom 000004h. Osim ove, postoji i alternativna tabela prekida - AIVT (*Alternate Interrupt Vector Table*), koja pruža mogućnost implementacije alternativnih *interrupt* rutina. Za kontrolu i upravljanje prekidima zadužen je *Interrupt Controller* koji sadrži 28 registara:

- INTCON1
- INTCON2
- IFS0 - IFS4
- IEC0 - IEC4
- IPC0 - IPC14 i IPC16

Brojač (*Timer*)

Brojač je jedna od najvažnijih komponenti mikrokontrolera. Može obavljati funkciju običnog sata - da jednostavno prati tok vremena. Može poslužiti za brojanje određenih događaja: koliko je bita poslato ili primljeno, koliko je koraka obavljeno u određenoj operaciji, koliko je obrtaja napravio motor... Svi procesori sadrže programski brojač - PC (*Program Counter*): postoji lista instrukcija koje treba da se izvrše jedna za drugom, u svakom sljedećem taktu brojač se inkrementira i naredna instrukcija se može izvršiti. Tako je ostvarena sinhronizacija. Fizički, brojač je registar čija se vrijednost inkrementira ili dekrementira. Ovaj proces se obavlja u pozadini svega što mikrokontroler radi.

PIC24FJ96GA010 sadrži pet brojača:

- Timer1 - 16-bitni brojač
- Timer2/3 - 32-bitni brojač, može se koristiti kao dva 16-bitna
- Timer4/5 - 32-bitni brojač, može se koristiti kao dva 16-bitna

Svaki brojač sadrži sljedeće registre:

- TMRx - 16-bitni brojački registar koji sadrži trenutno stanje brojača
- PRx - 16-bitni registar koji sadrži vrijednost periode brojača
- TxCON - 16-bitni kontrolni registar brojača

Svakom brojaču takođe su dodijeljeni odgovarajući biti za kontrolu prekida:

- TxIE - bit za omogućavanje prekida
- TxIF - statusni bit *flag*-a prekida
- TxIP<2:0> - tri kontrolna bita za podešavanje prioriteta prekida (registar IPCx).

Timer1

Postupak konfiguracije:

- Setovati TON bit (=1)
- Odabrati preskaler brojača, TCKPS1:TCKPS0
- Podesiti TCS i TGATE bite
- Podesiti TSYNC bit - sinhroni ili asinhroni mod
- Učitati period brojača u registar PR1
- Ukoliko je potrebno omogućiti prekide - setovati T1IE bit. Prioritet prekida izabrati pomoću T1IP2:T1IP0 bita.

Kontrolni registar Timer1-a je prikazan na slici 2.

REGISTER 10-1: T1CON: TIMER1 CONTROL REGISTER

Upper Byte:							
R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
TON	—	TSIDL	—	—	—	—	—
bit 15							bit 8

Lower Byte:							
U-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	U-0
—	TGATE	TCKPS1	TCKPS0	—	TSYNC	TCS	—
bit 7							bit 0

Slika 2 - T1CON (kontrolni registar *Timer1*-a)

- Bit 15 TON - 1 - uključi, 0 - isključi brojač
- Bit 13 TSIDL - 1 - zaustavi rad u *idle* modu, 0 - nastavi da brojiš u *idle* modu
- Bit 6 TGATE - ukoliko je TCS=1, nema ulogu; ukoliko je TCS=0, 1 - omogući *gate* operaciju, 0 - zabrani *gate* operaciju
- Bit 5 i 4 TCKPS1:0 - odaberi preskaler: 1:1 (00), 1:8 (01), 1:64 (10), 1:256 (11)
- Bit 2 TSYNC - ukoliko je TCS=0, nema ulogu; ukoliko je TCS=1, 1 - sinhronizuj eksterni takt, 0 - ne sinhronizuj eksterni takt
- Bit 1 TCS - odabir izvora takta: 0 - interni oscilator ($f_{osc}/2$), 1 - eksterni takt sa pina T1CK

Primjer:

Uz pomoć razvojne mikrokontrolerske ploče LV24-33 realizovati aplikaciju koja na LCD-u prikazuje izmjereno vrijeme u sekundama sa korakom od 64 ms, koliko i iznosi perioda brojača. Koristiti *Timer1*.

Osim inicijalizacije LCD-a, potrebno je, prije svega, izvršiti konfiguraciju brojača:

```
PR1=5000; //period brojača 5000 taktova
```

```
T1CON = 0x8020; //0b1000000000100000 - T1ON=1, TCS=0, TCKPS1=1, TCKPS0=0
```

Dakle, brojač koristi interni oscilator frekvencije $f_{osc}/2=5$ MHz (TCS=0) i preskaler 1:64 (TCKPS1:0=10), što znači da je perioda brojača:

$$T_{T1} = PR1 \cdot \frac{2}{f_{osc}} \cdot TCKPS = 5000 \cdot \frac{2}{5 \cdot 10^6} \cdot 64 = 64 \text{ ms}$$

Po odbrojanoj periodi brojača generiše se prekid koji se može iskoristiti za osvježavanje sadržaja LCD-a. Naravno, prekid brojača se mora omogućiti:

```
IEC0 = IEC0 | 0x0008;
```

Na osnovu IV (*Interrupt Vector*) tabele poznato je da se prekid *Timer1*-a može omogućiti setovanjem trećeg bita IEC0 kontrolnog registra, slika 3.

Na kraju, potrebno je prikazati odbrojano vrijeme na LCD-u u trenutku generisanja prekida brojača, odnosno po isteku svake periode brojača:

```
void Timer1Int() org 0x001A
{
    IFS0 = IFS0 & 0xFFFF7;    // izbrisi flag
    f += 0.064;    // f je globalna promjenljiva tipa float
    sprintf(s, "%.3f", f);
    Lcd_Out(1,1,s);
}
```

Adresa 0x001A preuzeta je iz IV tabele i na tu adresu će se usmjeriti tok izvršavanja programa kada se prekid desi, a gdje je u fazi kompajliranja "smještena" prethodno navedena metoda. *Flag* bit prekida *Timer1*-a je treći bit IFS0 registra, IV tabela.

Ključna riječ `org` koristi se kada je potrebno da se određena metoda (funkcija, procedura) sačuva na određenoj poziciji u programskoj memoriji. Upotreba `org` direktive neophodna je pri radu sa prekidima. Metode određenog prekida moraju se nalaziti na odgovarajućim pozicijama programske memorije.

Interrupt Source	Vector Number	IVT Address	AIVT Address	Interrupt Bit Locations		
				Flag	Enable	Priority
ADC1 Conversion Done	13	00002Eh	00012Eh	IFS0<13>	IEC0<13>	IPC3<6:4>
Comparator Event	18	000038h	000138h	IFS1<2>	IEC1<2>	IPC4<10:8>
CRC Generator	67	00009Ah	00019Ah	IFS4<3>	IEC4<3>	IPC16<14:12>
External Interrupt 0	0	000014h	000114h	IFS0<0>	IEC0<0>	IPC0<2:0>
External Interrupt 1	20	00003Ch	00013Ch	IFS1<4>	IEC1<4>	IPC5<2:0>
External Interrupt 2	29	00004Eh	00014Eh	IFS1<13>	IEC1<13>	IPC7<6:4>
External Interrupt 3	53	00007Eh	00017Eh	IFS3<5>	IEC3<5>	IPC13<6:4>
External Interrupt 4	54	000080h	000180h	IFS3<6>	IEC3<6>	IPC13<10:8>
I2C1 Master Event	17	000036h	000136h	IFS1<1>	IEC1<1>	IPC4<6:4>
I2C1 Slave Event	16	000034h	000034h	IFS1<0>	IEC1<0>	IPC4<2:0>
I2C2 Master Event	50	000078h	000178h	IFS3<2>	IEC3<2>	IPC12<10:8>
I2C2 Slave Event	49	000076h	000176h	IFS3<1>	IEC3<1>	IPC12<6:4>
Input Capture 1	1	000016h	000116h	IFS0<1>	IEC0<1>	IPC0<6:4>
Input Capture 2	5	00001Eh	00011Eh	IFS0<5>	IEC0<5>	IPC1<6:4>
Input Capture 3	37	00005Eh	00015Eh	IFS2<5>	IEC2<5>	IPC9<6:4>
Input Capture 4	38	000060h	000160h	IFS2<6>	IEC2<6>	IPC9<10:8>
Input Capture 5	39	000062h	000162h	IFS2<7>	IEC2<7>	IPC9<14:12>
Input Change Notification	19	00003Ah	00013Ah	IFS1<3>	IEC1<3>	IPC4<14:12>
Output Compare 1	2	000018h	000118h	IFS0<2>	IEC0<2>	IPC0<10:8>
Output Compare 2	6	000020h	000120h	IFS0<6>	IEC0<6>	IPC1<10:8>
Output Compare 3	25	000046h	000146h	IFS1<9>	IEC1<9>	IPC6<6:4>
Output Compare 4	26	000048h	000148h	IFS1<10>	IEC1<10>	IPC6<10:8>
Output Compare 5	41	000066h	000166h	IFS2<9>	IEC2<9>	IPC10<6:4>
Parallel Master Port	45	00006Eh	00016Eh	IFS2<13>	IEC2<13>	IPC11<6:4>
Real-Time Clock/Calendar	62	000090h	000190h	IFS3<14>	IEC3<13>	IPC15<10:8>
SPI1 Error	9	000026h	000126h	IFS0<9>	IEC0<9>	IPC2<6:4>
SPI1 Event	10	000028h	000128h	IFS0<10>	IEC0<10>	IPC2<10:8>
SPI2 Error	32	000054h	000154h	IFS2<0>	IEC0<0>	IPC8<2:0>
SPI2 Event	33	000056h	000156h	IFS2<1>	IEC2<1>	IPC8<6:4>
Timer1	3	00001Ah	00011Ah	IFS0<3>	IEC0<3>	IPC0<14:12>
Timer2	7	000022h	000122h	IFS0<7>	IEC0<7>	IPC1<14:12>
Timer3	8	000024h	000124h	IFS0<8>	IEC0<8>	IPC2<2:0>
Timer4	27	00004Ah	00014Ah	IFS1<11>	IEC1<11>	IPC6<14:12>
Timer5	28	00004Ch	00014Ch	IFS1<12>	IEC1<12>	IPC7<2:0>
UART1 Error	65	000096h	000196h	IFS4<1>	IEC4<1>	IPC16<6:4>
UART1 Receiver	11	00002Ah	00012Ah	IFS0<11>	IEC0<11>	IPC2<14:12>
UART1 Transmitter	12	00002Ch	00012Ch	IFS0<12>	IEC0<12>	IPC3<2:0>
UART2 Error	66	000098h	000198h	IFS4<2>	IEC4<2>	IPC16<10:8>
UART2 Receiver	30	000050h	000150h	IFS1<14>	IEC1<14>	IPC7<10:8>
UART2 Transmitter	31	000052h	000152h	IFS1<15>	IEC1<15>	IPC7<14:12>

Slika 1 - Interrupt Vector Table (IVT)