

Liquid-Crystal Display (LCD) i Real-Time Clock/Calendar (RTCC) modul

1 LCD (*Liquid-Crystal Display*)

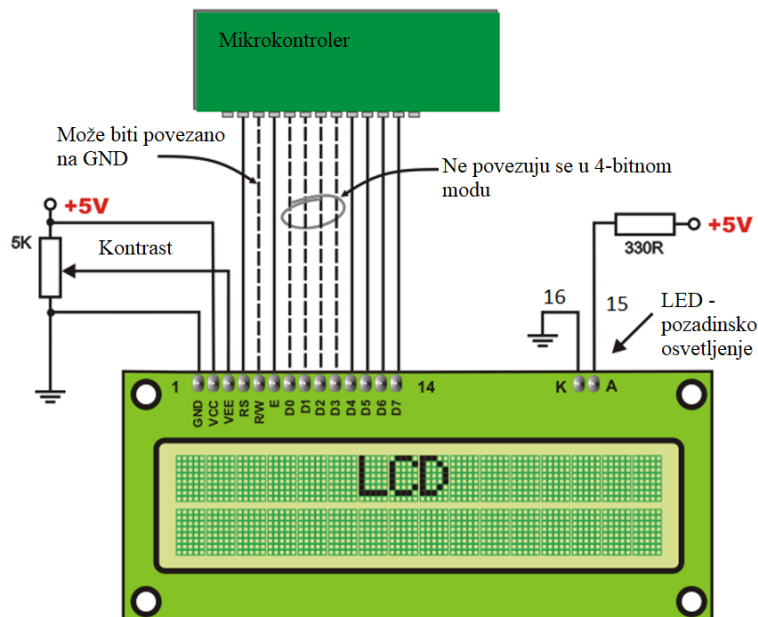


Slika 1: LCD

Na ovakvim LCD displejima moguće je prikazati simbole kao što su slova alfabeta, brojevi, znakovi interpunkcije, matematički simboli i sl. Osnovni element LCD displeja je matrica piksela veličine 5x8. Popunjavanjem piksela ove matrice ostvaruje se prikaz različitih simbola (slika 2).

	d4	d3	d2	d1	d0	BINARY	HEX
b0	□	□	□	□	□	xxx00000	0x00
b1	□	□	□	□	□	xxx00000	0x00
b2	□	■	□	■	□	xxx01010	0x0A
b3	□	□	□	□	□	xxx00000	0x00
b4	■	□	□	□	■	xxx10001	0x11
b5	□	■	■	■	□	xxx01110	0x0E
b6	□	□	□	□	□	xxx00000	0x00
b7	□	□	□	□	□	xxx00000	0x00

Slika 2: LCD element



Slika 3: Povezivanje mikrokontrolera i LCD displeja

Oznaka pina	Funkcija pina
GND	Ground
VCC	Napajanje displeja
VEE	Podešavanje kontrasta displeja
RS	Odabir slanja komandi (logička 0), odabir slanja podataka (logička 1)
EN	<i>Enable pin</i> – koristi se za slanje upisane instrukcije, kada se dogodi tranzicija sa logičke jedinice na logičku nulu
RW	Pin se koristi za komunikaciju sa LCD kontrolerom, kada je pin na logičkoj 1 vrši se čitanje, kada je pin na logičkoj 0 vrši se pisanje
D0-D7	Pinovi za prenos podataka/komandi (mogu se koristiti 4 pina ili 8 pinova)

Tabela 1: Pinovi LCD displeja

1.1 Organizacija memorije LCD-a

- **DDRAM (*Display Data RAM*)**

DDRAM memorija LCD displeja predstavlja memorijski blok u kome se skladište karakteri koji će biti prikazani na ekranu. DDRAM ima mogućnost skladištenja do 80 karaktera. Memorija je organizovana u dva reda. Prikaz karaktera počinje od nulte adrese i u jednom trenutku prikazuje se 16 karaktera, dok se naredni karakteri prikazuju šiftovanjem u desno.

- **CGRAM (*Character Generator RAM*)**

CGRAM memorija zadužena je za kreiranje korisničkih karaktera. Ukoliko predefinisani set karaktera koje je moguće prikazivati na ekranu nije dovoljan, korisnik ima mogućnost definisanja proizvoljnih karaktera upisom u CGRAM memoriju. Kapacitet CGRAM memorije je 64 bajta, gde je svaki registar dužine 8 bitova. Kako je veličina karaktera 5x8, koristi se samo donjih pet bitova. Dakle, svaki registar odgovara jednom redu matrice 5x8. Upisom logičke 1 ostvaruje se taman, odnosno, popunjen piksel na ekranu, dok logička 0 u registru CGRAM-a označava osvetljen tj. prazan piksel. Na ovaj način, upisom 0 i 1 na odgovarajuća mesta CGRAM-a moguće je definisati proizvoljan karakter.

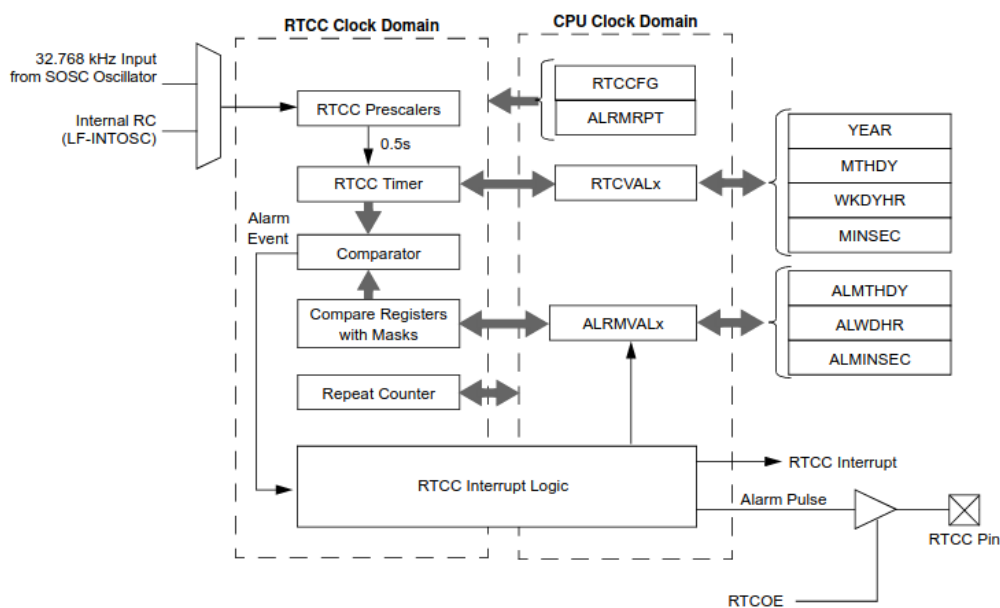
- **CGROM (*Character Generator ROM*)**

CGROM sadrži standardnu mapu karaktera koji se mogu prikazati na dispelju. Ova memorija je read only i pamti podatke i po isključenju napajanja. Svakom karakteru dodeljena je određena lokacija, gde raspored odgovara ASCII tabeli. Tako je karakter P na lokaciji 0b0101 0000, odnosno, 0x50, što je upravo heksadecimalna vrednost u ASCII tabeli koja odgovara slovu P.

2 RTCC (*Real-Time Clock/Calendar*)

2.1 Unutrašnja struktura RTCC modula

RTCC modul mikrokontrolera koristi se za praćenje datuma i vremena, gde je vremenska rezolucija 1 s. Unutar mikrokontrolera PIC18F87K22 nalazi se RTCC modul, čiji je blok dijagram prikazan na slici 4.



Slika 4: Blok dijagram RTCC modula

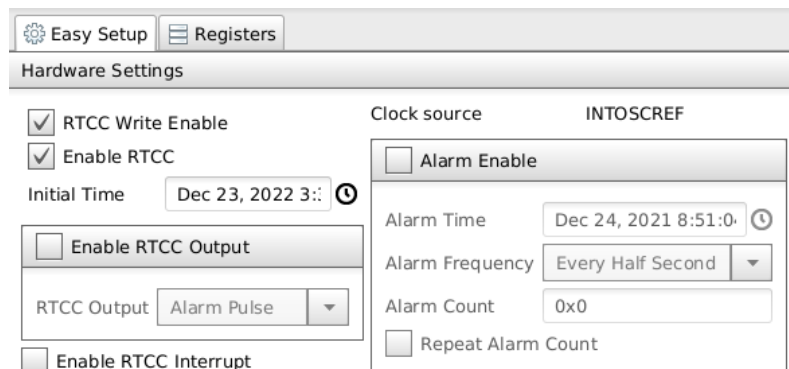
Kao izvor takta moguće je koristiti spoljašnji sekundarni oscilator (SOSC pin) gde se povezuje oscilator 32,768 kHz. Prednost ovog oscilatora je mogućnost korišćenja za modove uštede energije (zbog niže frekvencije). Takođe, deljenjem frekvencije 32,768 kHz sa 2^N može se dobiti vrednost frekvencije pogodna za takt RTCC modula. U ovom slučaju je ta vrednost 2 Hz tj. N je 14 (takt RTCC tajmera na blok dijagramu je 0.5 s). Kalibrisani sekundarni oscilator, spram specifikacije proizvođača ima, ima grešku manju od 3 s na mesec dana. Umesto sekundarnog moguće je koristiti interni RC oscilator, ali treba imati u vidu da je greška ovog oscilatora znatno veća.

RTCC modul ima mogućnost dodavanja alarm signala na pinu RTCC ili generisanjem alarma putem RTCC prekida. Mehanizam alarmiranja se najčešće koristi prilikom korišćenja u stanju spavanja (mod male potrošnje) mikrokontrolera. Kada se generiše alarm, mikrokontroler se budi i izvršava odgovarajuće akcije.

Važno je istaći da se u registrima RTCC modula podaci čuvaju u BCD (*Binary-Coded Decimal*) formatu. Ovde se svaka cifra broja zasebno predstavlja sa četiri bita tj. svaki nibl predstavlja zasebnu cifru. Tako bi se broj 32 kodovao u BCD kao 0x32 tj. 0b00110010. Prva četiri bita koduju cifru desetica, dok naredna četiri bita koduju cifru jedinica.

2.2 Rad sa RTCC modulom u MPLAB-u

Kako bi se koristio RTCC modul mikrokontrolera u *MPLAB Code Configurator (MCC)* alatu neophodno je dodati iz *Device Resources* padajućeg menija pod poljem *Peripherals* → *RTCC* - RTCC modul. Tom prilikom se u *MCC* dodaje nova kartica pod nazivom RTCC prikazana na slici 5.



Slika 5: Dodavanje RTCC modula u MCC

Mogućnost modifikacije vremena omogućena je *RTCC Write Enable* poljem, dok je RTCC modul uključen pomoću *Enable RTCC* polja. *Initial Time* predstavlja polje za unos vremena koje se želi inicijalno postaviti u RTCC modulu, a umesto unosa stringa moguće je odabrati vreme iz kalendara klikom na ikonicu sata. *Enable RTCC* polje podešava mogućnost generisanja signala na izlaznom pinu RTCC i u ovom slučaju je on isključen. Takođe, isključeno je i generisanje prekida jer *Enable RTCC Interrupt* polje nije označeno. Polje *Clock source* podešava izvor takta, gde

je u ovom primeru korišćen unutrašnji RC oscilator. Podešavanje alarma se vrši u podmeniju *Alarm Enable*, ali se on neće koristiti u ovom primeru.

Nakon podešavanja generisan je kod, čime su generisane funkcije:

- *RTCC_Initialize* - Inicijalizuje RTCC modul na vreme zadato u konfiguratoru, podešava oscilator i uključuje RTCC modul,
- *RTCC_TimeReset* - Deinicijalizuje RTCC modul, pa je moguće ponovo izvršiti inicijalizaciju vremena,
- *RTCCTimeInitialized* - Vraća *bool* vrednost (*true* RTCC modul inicijalizovan, u suprotnom je *false*),
- *RTCC_TimeGet* - Preuzima vrednost vremena i smešta ga u strukturu *tm* iz standardne GCC biblioteke *time.h*. Važno je istaći da se vrednosti moraju konvertovati iz BCD koda u standardan način kodovanja, što je realizovano pozivom funkcije *ConvertBCDToHex*,
- *RTCC_TimeSet* - Podešava vreme RTCC modula za prosleđenu strukturu *tm*. Pošto se vrednosti podešavaju u BCD kodu neophodno ih je konvertovati pozivom funkcije *ConvertHexToBCD*,
- *RTCC_BCDTimeGet* - Preuzima trenutno vreme RTCC modula u BCD kodu,
- *RTCC_BCDTimeSet* - Podešava vreme RTCC modula, ali prosleđena struktura mora sadržati polja u BCD kodu.
- *ConvertDateTimeToUnixTime* - Konvertuje vreme iz strukture *tm* u unix format vremena (broj sekundi proteklih počevši od 01.01.1970.),
- *ConvertUnixTimeToDateTime* - Konvertuje prosleđeni unix format vremena u format definisan strukturom *tm*.