

Univerzitet u Kragujevcu  
Fakultet inženjerskih nauka



Seminarski rad iz predmeta:  
**Arhitektura računarskih sistema**

Tema:  
mikroBoard for PIC 40-pin

Student:  
Jovan Anđelković 587/2015

Predmetni profesor:  
Aleksandar Peulić

Kragujevac 2016.

## SADRŽAJ

1. UVOD .....	2
2. ARHITEKTURA.....	4
2.1 PIC18F4520 .....	5
3. PROJEKTNII ZADATAK.....	11
4. REALIZACIJA PROJEKTOG ZADATAKA .....	12
5. ZAKLJUČAK .....	14
6. LITERATURA .....	14

## 1.UVOD

MikroBoard je pre svega napravljen radi postavljanja mikrokontrolera na razvojni sistem. Kada se kaže postavljanje, misli se na povezivanje mikroBoard-a sa razvojnom pločom odnosno razvojnim sistemom. Sad se postavlja pitanje: “Šta je mikroBoard?”.

Mikroboard je samo jedan od mnogih mikroprocesorskih razvojnih sistema, koji je napravljen na štampanoj ploči. Sadrži mikroprocesor, odnosno mikrokontroler kao i minimalnu pomoćnu logiku, koja je potrebna svakom inženjeru, kako bi manipulisao pločom. Sve je to potrebno kako bi se zadavale razne naredbe i postavljala podešavanja kao što su : ulazno-izlazna kola, takt, RAM memorija, čuvanje programske memorije, itd...

Svha je da ploča bude u startu korisna za rad, kako se ne bi gubilo vreme na programiranje osnovnih naredbi za kontrolu hardvera. MikroBoard služi korisniku za pravljenje prototipa aplikacija, koje bi se koristile na njemu za razne potrebe.

Za razliku od sistema opšte namene poput PC-a, mikroBoard sadrži malo (ili nema uopšte) korisničkog interfejsa. Može samo da prihvata i pokreće programe koji mu se šalju (na primer slanje podataka na ulazni port sa nekog senzora).

Razvojne ploče su napravljene najpre zbog potrebe učenja i sticanja praktičnog iskustva sa novim familijama mikrokontrolera, jer omogućavaju lako povezivanje i komunikaciju sa njima. Uklonjeno je sa njih sve što je bilo suvišno, kako bi se smanjili troškovi. Zbog toga cena hardvera stalno pada, pa mikrokontroleri postaju veoma pristupačni. Zahvaljujući svojim malim dimenzijama, lako su prenosivi.

Kao što je rečeno na početku, ključna stvar kod mikroBoard-ova je mikrokontroler. Najjednostavnije rečeno, mikrokontroler je mali računar u jednom integrisanom kolu. U modernoj terminologiji se kaže da je to sistem koji se nalazi u čipu (SoC). Mikrokontroler može da sadrži jedan ili više procesora (procesorskih jezgara) zajedno sa memorijom i ulaznom-izlaznim periferijama. Za razliku od mikroprocesora koji se koriste za PC i druge opšte namene, mikrokontroleri su dizajnirani za ugrađene aplikacije.

Mikrokontroleri se koriste za automatsko kontrolisanje raznih proizvoda i uređaja kao što su: sistemi za kontrolu motora u automobilskoj industriji, medicinski uređaji, daljinski upravljači, kancelarijske mašine i drugi sistemi.

Mogu da rade na veoma niskim frekvencijama (neki čak mogu da rade i na nižim od 4 kHz), uz malu potrošnju energije. Imaju sposobnost da zadrže svoju funkcionalnost dok čekaju da se neka operacija izvrši, kao što je pritisak tastera. U takvom “sleep” modu, mikrokontroler čeka sledeći prekid kada će obaviti zadate naredbe. Osim takta CPU-a, sve ostalo je ugašeno u mikrokontroleru, što dovodi do veoma male potrošnje koja se može izraziti u nanovatima (nW). Osim što mogu da štede bateriju, mogu se koristiti i modovi u kojima mikrokontroler koristi maksimalno svoje performanse. To su slučajevi gde je potrebno da deluje poput digitalnog procesora signala (DSP) sa većim brzinama takta i potrošnjom energije.

Što se tiče principa rada, imaju minimalne zahteve za memoriju i dužinu programa. Nemaju kompleksan softver i nemaju operativni sistem. Sastoje se od tipičnih ulaznih i izlaznih uređaja poput prekidača, releja, solenoida, LED-a, radio frekventnih uređaja... Neki imaju tačno-kristalne ekrane i senzore za podatke kao što su: temperatura, vlažnost vazduha, nivo svetlosti i mnoge druge. Uglavnom ovi “embedded” sistemi nemaju tastaturu, diskove, štampače i druge I/O uređaje koji služe za ljudsku interakciju (human interaction devices).

Prvobitno su mikrokontroleri bili programirani uz pomoć asemblerskog jezika, ali u međuvremenu su počeli da se koriste programski jezici visokog nivoa (kao što su C, Python, JavaScript,...) za programiranje mikrokontrolera. Neki mikrokontroleri imaju okruženje koje pomaže razvoju određene vrste aplikacije. Proizvođači često prave dostupne alate za lakše usvajanje hardvera.

Kako se tehnologija razvijala, tako su počeli da se pojavljuju i simulatori za neke mikrokontrolere. Oni omogućavaju programeru da simulira ponašanje mikrokontrolera, odnosno da prikaže šta bi se desilo kada bi programer koristio realan uređaj. Simulator prikazuje unutrašnje stanje procesora kao i izlaza, pri generisanju ulaznih signala, što pomaže korisniku u analizi njegovog programa. Dok je u nekim situacijama simulator ograničen u simulaciji realnog hardvera, u drugim slučajevima je dobar za reprodukciju neke implementacije koja bi u realnoj situaciji bila komplikovana. Zato je simulator nekada bolji i brži način za “debug” i analizu problema.

Postoje nekoliko desetina proizvođača mikrokontrolera, od kojih su ovde navedeni samo neki:

- ARM core processors
  - ARM Cortex-M
- Freescale ColdFire
- Freescale 68HC11
- Intel 8051
- MIPS
- Microchip Technology PIC
- Parallax Propeller
- PowerPC ISE
- STMicroelectronics
- Texas Instruments TI MSP430
- Toshiba TLCS-870

## 2. ARHITEKTURA

MikroBoard for PIC 40-pin je dizajniran za povezivanje mikrokontrolera sa razvojnim sistemom. PIC je familija mikrokontrolera napravljena od strane kompanije „Microchip Technology“, a ime PIC je nastalo od „Peripheral Interface Controller“. Kompanija je počela da proizvodi ovu familiju još 1976, a do 2013. godine su prodali preko 12 milijardi delova koji se koriste kod „embedded“ sistema. U početku su modeli imali samo ROM memoriju sa malim prostorom za čuvanje programa (takozvani „EPROM“). Danas svi modeli koriste fleš memoriju za programe, dok novije verzije omogućavaju PIC-u da se reprogramira.

Programska memorija i memorija podataka su razdvojene. Memorija podataka se kreće od 8 do 32-bitne širine kod najnovijih modela. Skup instrukcija takođe zavisi od modela i broja čipova koji obezbeđuju funkciju digitalne obrade signala. PIC uređaji sadrže razne ulazno-izlazne pinove, ADC i DAC module i komunikacione portove kao što su UART, I2C, CAN i USB. Sadrže više modova rada poput „low-power“ i „high-speed“ varijacija.

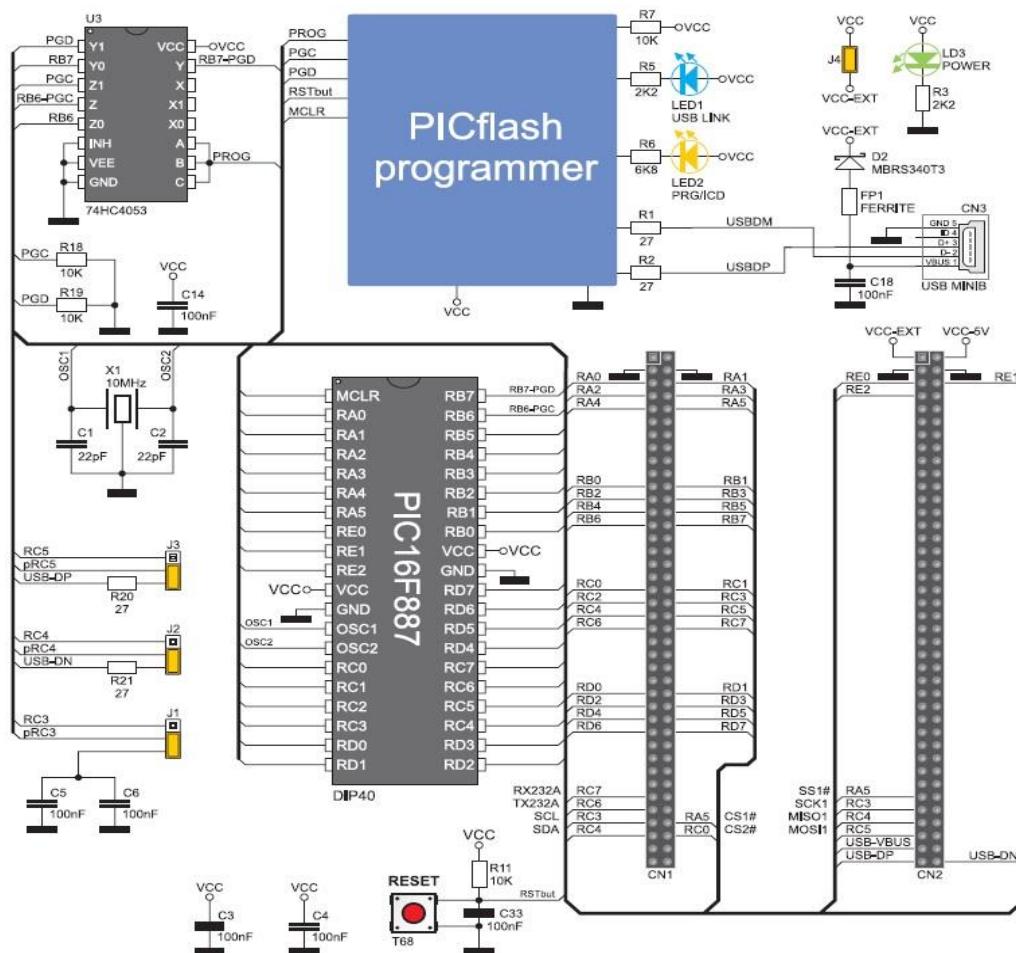
PIC mikročipovi su dizajnirani na principu Harvard arhitekture. U okviru ove familije mikrokontrolera, postoji veliki broj porodica, ali se svi svode na 2 oznake. To su oznake PICnCxxx (CMOS) i PICnFxxx (Flash). Uređaji „C“ se klasifikuju kao „End-Of-Life“ i oni se više ne proizvode. To su bili početni modeli koji se opisuju kao OTP, ROM ili EEPROM, dok „F“ proizvodi koriste Flash memoriju i dalje su aktuelni.

MikroBoard se koristi kao razvojni alat koji u sebi sadrži mikrokontroler, programer, mali proto board i dva „muška“ ulaza za povezivanje sa razvojnim sistemom UNI-DS6. Svaki mikroBoard poseduje integrisani programer koji se koristi za MCU programiranje (programiranje mikrokontrolera). Osim mogućnosti korišćenja mikroBoard-a sa razvojnim sistemom UNI-DS6, ovaj 40-pinski alat se može koristiti i kao samostalan uređaj.

Na 40-pinskom mikroBoard-u se nalazi mikrokontroler PIC18F4520, koji pripada arhitekturi PIC18. Brzina ovog mikrokontrolera je 20 MHz i ima 32 Kb Flash memorije kao i 1536 Kb RAM. Ima USB priključak preko koga dobija potreban napon od 5V. Microchip je predstavio ovu arhitekturu 2000. godine. Ona je razliku od prethodnih uređaja (koji su koristili assembler), koristila C kao dominantan programski jezik. Takođe se ovoj seriji PIC18 dodaju nove važne karakteristike:

- Call stack je 21-bit širok i dosta „dublji“
- Grananje instrukcija
- Indeksno adresiranje (PLUSW)
- Proširivanje FSR registra na 12 bita, što im je omogućilo da linearno adresiraju čitav adresni prostor

Ove i još mnoge druge promene su pre svega usmerene na unapređenje efikasnosti stack-a. FSR registar se koristi kao „stack pointer“ odnosno pokazivač, tako da su ove promene dovele do korišćenja ovog registra, radi lakšeg indeksiranja.



**Slika 1. Šema mikroBoard-a**

## 2.1 PIC18F4520

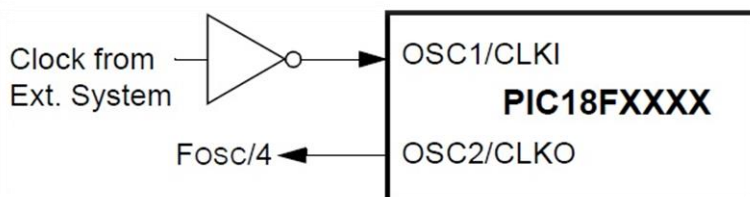
Ovaj mikrokontroler sadrži niz funkcija koji mogu značajno da smanje potrošnju energije tokom rada. Postoje više modova rada, poput *Alternativnog moda* kod koga se potrošnja energije u toku izvršavanja koda može smanjiti za čak 90%. U ovom modu, kontroler radi po taktu koji dolazi iz clock-a Timer1. U *Multiple Idle* modu se može još više smanjiti potrošnja (čak do 4% normalnih potreba za izvršavanje operacija), jer se kontroler oslanja na periferiju, a CPU je isključen.

*On-the-fly* mod se poziva od strane korisnika tokom obavljanja operacija. Korisnik sam podešava aplikaciju uštede energije. Poslednji režim štednje je *Low Consumption*. Kod ovog moda se koristi minimalno napajanje za Timer1 i Watchdog Timer, koji diktiraju takt kontrolera.

PIC mikrokontroler nudi 10 različitih opcija za oscilatore, što daje korisniku veliki izbor pri razvijanju aplikacija. To su:

- Četiri *Crystal* moda koji koriste kristale ili keramičke rezonatore
- Dva *External Clock-a* kod kojih se mogu koristiti dva pina (ulaz za oscilator i "divide-by-4 clock" izlaz) ili jedan pin (ulaz za oscilator uz još jedan I/O pin za opštu namenu)
- Dva *External RC Oscillator* režima sa istim pin opcijama kao kod *External Clock-a*
- *Internal oscillator block* koji daje 8 MHz clock i INTRC izvor (približno 31 kHz) kao i niz od 6 različitih frekvencija između 125 kHz i 4 MHz, koje mogu da se biraju
- *Phase Lock Loop* (PLL) je množak frekvencije. On je dostupan svim preostalim modovima i nudi frekvencije do 40 MHz. U kombinaciji sa unutrašnjim oscilatorom, PLL može da ponudi frekvencije od 31 kHz do 32 MHz i to bez dodatnih kristalnih oscilatora.
- *Fail-Safe Clock Monitor* se koristi za stalno praćenje glavnog izvora takta i takta koji stvara Internal oscilator. Ako dođe do greške kod glavnog clock-a, kontroler prelazi na Internal oscillator block, kako bi nastavio dalje sa operacijama ili kako bi se bezbedno ugasio.
- *Two-Speed Start-up* je opcija koja koristi Internal oscillator kao glavni izvor takta u situacijama kada kontroler prelazi iz Power-on u Reset i iz Sleep u Wake-up režim. Internal oscillator radi sve dok glavni clock ne postane dostupan i spreman za rad.

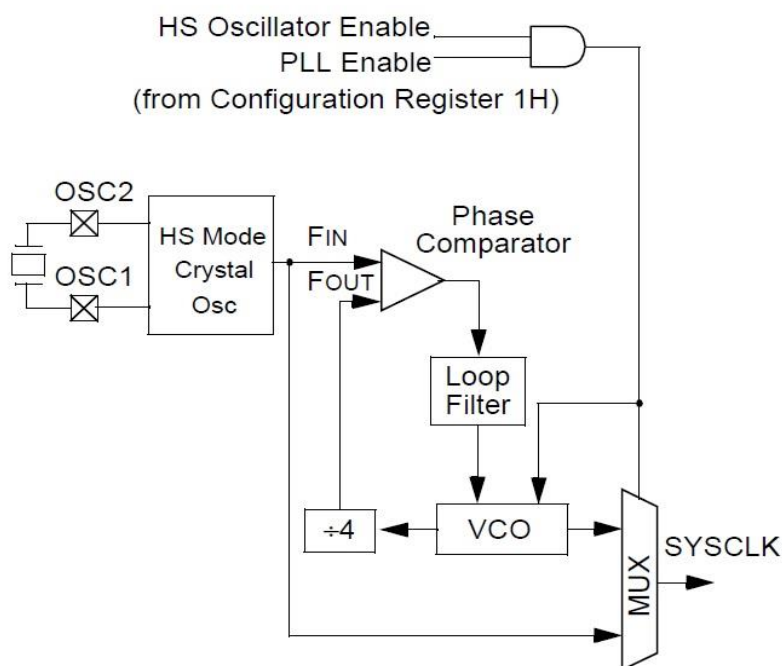
EC (External Clock sa FOSC/4 izlazom) i ECIO (External Clock sa I/O) oscilatori zahtevaju povezivanje dodatnog izvora takta na OSC1 pin. Frekvencija oscilatora podeljena sa 4 je kod EC-a dostupna na OSC2 pinu. Signal se može koristiti za neka testiranja ili sinhronizaciju drugih logičkih komponenti. ECIO Oscillator mod funkcioniše kao i EC. Jedina je razlika što OSC2 pin kod ECIO-a postaje I/O pin, odnosno PORTA (RA6).



Slika 2. Konfiguracija EC-a

Za “taktovanje” nekih neosetljivih aplikacija, RC Oscillator nudi dodatnu uštedu. Ovaj oscilator u sebi sadrži više faktora poput napona, vrednosti spoljnog otpornika (REXT) i kondenzatora (CEXT), radna temperatura,... U ovom modu je takođe dostupna “frekvencija/4” na OSC2 pinu.

Phase Locked Loop (PLL) kolo je opciono podesivo kolo, koje korisnik koristi za nisku frekvenciju oscilatora ili da podesi uređaj da radi na najvišoj mogućoj frekvenciji uz pomoć kristalnog oscilatora. Ovo može veoma biti korisno kada je potrebno ubrzavanje clock-a. PLL je dostupan i u kombinaciji sa Internal oscillator block-om. Generiše frekvencije do 32 MHz.



Slika 3. PLL šema

Postoje 3 tipa memorije kod PIC18F4520:

- Programska memorija
- RAM
- EEPROM

Pošto je ovaj mikrokontroler dizajniran po Harvard arhitekturi, podaci i programska memorija koriste posebne magistrale. PIC18 ima 21-bit programski brojač koji je sposoban za adresiranje 2 Mb memorijskog prostora. Pristup lokaciji koja se nalazi između fizički implementirane memorije i 2 Mb vraća “0”. PIC18F4520 ima 32 Kb Flash memorije i može da sačuva do 16 384 single-word instrukcija i ima dva interrupt vektora (reset vektor ima adresu 0000h i interrupt vektor na adresama 0008h i 0018h).



Programski brojač (PC) je smešten u tri odvojena 8-bitna registra. Niži bajt je poznat kao PCL registar u koji se može nešto upisati i iz njega čitati. Viši bajt je PCH registar sadrži 15:8 bitova PC-a i on ne dozvoljava direktno upisivanje ili čitanje. PCH ažuriranja se obavljaju preko PCLATH registra. Gornji bajt se naziva PCU. On u sebi sadrži 20:16 bitova programskog brojača i takođe nije podložan direktnom upisivanju ili čitanju. Svoja ažuriranja vrši preko PCLATU registra. Sadržaj iz PCLATH i PCLATU se prenose do programskog brojača za bilo koju operaciju koju upiše PCL. Analogno tome, gornja dva bajta programskog brojača se prenose u PCLATH i PCLATU za operaciju koju pročita PCL.

PIC18 u sebi sadrži takozvani „fast register stack“ koji sarađuje samo sa određenim brojem registara. Njegova uloga je da omogući „brz povratak“ informacija u registre, nakon prekida. Za svaki odabrani registar postoji jedan nivo brzog steka, u koji se ne može ništa upisati ili iz njega pročitati. Unutra se mogu nalaziti samo trenutne vrednosti odgovarajućeg registra. Od prioriteta prekida zavisi da li će se za čuvanje podataka koristiti brzi stek registar ili običan stek.

Memorija podataka se u PIC18F4520 realizuje kao statički RAM. Svaki registar u memoriji ima 12-bit adresu i može da sadrži do 4096 bajtova memorije. Memorija sadrži SFR (Special Function Register) i GPR (General Purpose Register) registre. SFR se koristi za kontrolu i status kontrolera kao i periferne funkcije, dok se GPR koristi za prenos i čuvanje podataka (kao i druge namene koje određuje korisnik).

Što se tiče Flash programske memorije, ona se može čitati, brisati i u nju se može upisivati. Čitanje iz memorije se izvršava bajt po bajt, dok se upisivanje/brisanje programske memorije vrši blok po blok (svaki blok sadrži 64 bajta). Pisanje ili brisanje programske memorije će prekinuti dalje prikupljanje instrukcija sve dok se operacija ne završi. Dok neka od ove dve operacije traje, ne može se pristupiti Flash memoriji.

EEPROM je trajna memorija koja se koristi za dugoročno čuvanje programskih podataka. Kao što je već rečeno, ona se razlikuje u odnosu na RAM i programsku memoriju, pa je samim tim EEPROM memorija odvojena od njih. Ova vrsta memorije nije direktno postavljena u određenim registrima ili programskoj memoriji, već je indirektno adresirana preko SFR registara. EEPROM je „čitljiva i upisiva“ tokom normalnog rada kontrolera i to se izvodi uz pomoć četiri SFR-a:

- EECON1
- EECON2
- EEDATA
- EEADR

Kada EEPROM pristupa memorijskom bloku podataka, EEDATA sadrži 8-bit podatak za čitanje/upisivanje, a EEADR registar sadrži adresu lokacije kojoj EEPROM pristupa. Pri pisanju u EEPROM memoriju na određenu lokaciju, automatski se brišu podaci sa te lokacije i unose se novi („erase-before-write“). Vreme upisivanja se kontroliše uz pomoć tajmera koji se nalazi na čipu, a pored čipa vreme zavisi i od napona i temperature. Smatra se da su EEPROM podaci veoma izdržljivi i da je njihovo adresiranje optimizovano za čuvanje informacija koje se često ažuriraju.

PIC18F4520 sadrži deset registra koji se koriste za kontrolu prekida:

- RCON
- INTCON
- INTCON2
- INTCON3
- PIR1, PIR2
- PIE1, PIE2
- IPR1, IPR2

U suštini, izvori prekida imaju tri bita kojima kontrolišu rad. „Flag bit“ ukazuje na trenutak kada je potreban prekid. Tu je i „Enable bit“ koji dopušta pokretanje programa koji poziva prekid. Enable se pokreće tek kada je Flag postavljen. Treći bit je „Priority bit“, koji određuje da li prekid ima viši ili niži prioritet u odnosu na ostale.

Funkcija prioriteta prekida se uključuje postavljanjem IPEN bita (sedmi bit RCON registra). Kada je postavljen IPEN bit, dalje se posmatraju dva bita koja globalno omogućavaju prekide. Svi prekidi višeg prioriteta (Priority bit = 1) su dozvoljeni ako je postavljen GIEH bit (sedmi bit INTCON registra), dok postavljanjem GIEL bita (šesti bit INTCON) se dozvoljavaju prekidi nižeg prioriteta (Priority bit = 0).

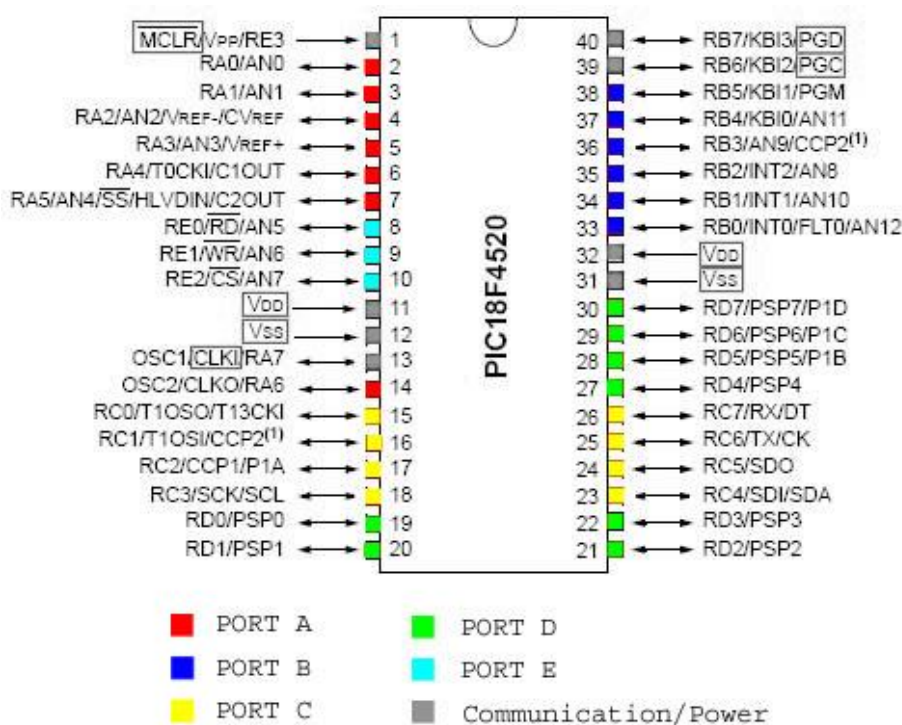
Kada se obavi ova procedura određivanja bitova, interrupt se odmah adresira na lokaciju 0008h ili 0018h, u zavisnosti od postavke prioriteta. Kada je IPEN bit u početnom stanju, funkcija postavljanja prioriteta je isključena i svi prekidi se klasifikuju po definiciji PIC18 (Compatibility mode).

Neki pinovi kod mikrokontrolera se multipleksiraju sa nekom alternativnom funkcijom sa periferije, ali to zavisi od datuma proizvodnje PIC18. Svaki port se sastoji od tri registra:

- TRIS registar (određuje smer kretanja podataka)
- PORT registar (proverava broj pinova priključenog uređaja)
- LAT registar (izlazni latch)

Data Latch (LAT) je koristan za „read-modifywrite“ operacije nad podacima koji se primaju/šalju na I/O pinove. Za PIC18F4520 postoje portovi A, B, C, D i E, tako da svaki port ima svoja tri registra. PORTA je širine 8 bita i obavlja komunikaciju u oba smera. Smer komunikacije određuje TRISA bit. Kada se postavi TRISA bit = 1, onda je PORTA ulazni pin. U suprotnom, PORTA je izlazni pin. LATA registar primenjuje svoje funkcije (kao i svaki LAT registar) na informacije koje prolaze kroz PORTA. Čak i slučajevima kada se PORTA koristi kao analogni ulaz, TRISA kontroliše smer PORTA pina. U tom slučaju, korisnik mora da proveri kako su postavljeni bitovi u TRISA registru.

Ostali portovi (PORTB, PORTC, PORTD i PORTE) su iste konfiguracije kao i PORTA. Svi su bidirekionalni, širine 8 bita, imaju gotovo iste funkcije,... Razlikuje se PORTE po širini, jer kod 40/44-pin uređaja, mikrokontroler PIC18 ima PORTE širine 4 bita.



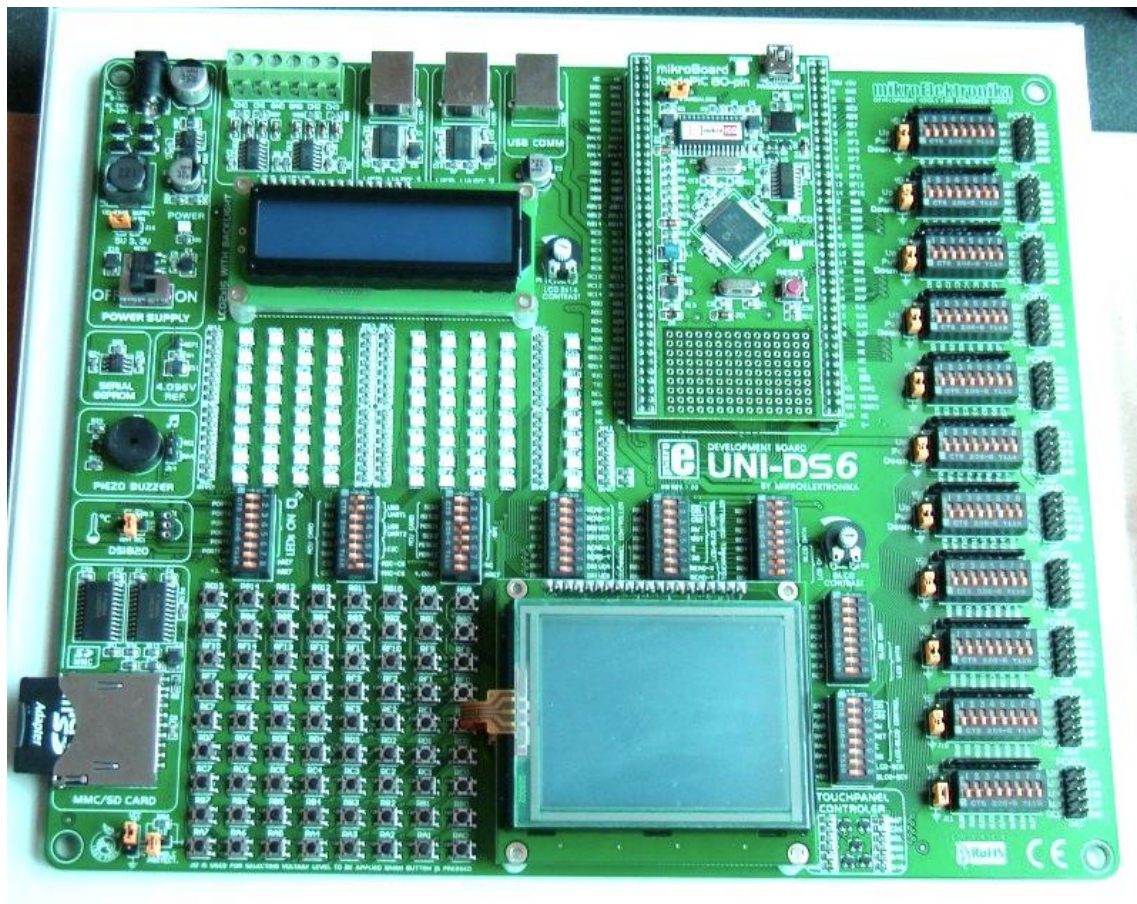
Slika 4. Raspored portova

PORTD pored svoje opšte funkcije, može da radi kao 8-bit PSP (Parallel Slave Port) ili microprocessor port. Operacije PSP-a se kontrolišu uz pomoć 4 viša bita registra TRISE. Sve dok je kontrolni bit (četvrti bit TRISE registra) jednak jedinici, PSPMODE je uključen i obavljaju se PSP operacije. U “Slave” modu port može asihrono da čita i upisuje podatke. PSP može direktno da komunicira sa 8-bit mikroprocesorom, tako da taj mikroprocesor može da čita iz PORTD latch-a i u njega upisuje podatke.

Kada je podešen kontrolni bit i uključen PSP mod, podešavaju se i I/O pinovi PORTE. Pin RE0 postaje ulazni RD pin sa kojeg se čitaju podaci koji pristižu. RE1 postaje WR ulazni pin, odakle se dalje upisuju podaci, a RE2 postaje CS (Chip Select) ulazni pin. Upisivanje počinje kada se detektuje nizak napon na CS i WR, a završava se kada se na nekoj od ove dve linije pojavi visok napon. Analogno važi i za čitanje podataka, samo se umesto WR-a prati napon na RD liniji.

### 3. PROJEKTNI ZADATAK

Zadatak ovog projekta je manipulacija LED dioda kod UNI-DS6 razvojnog sistema uz pomoć 40-pinskog mikroBoard-a (PIC18F4520 mikrokontrolerom). UNI-DS6 je razvojni sistem za programiranje i eksperimentisanje mikrokontrolera raznih proizvođača. Sadrži razne module poput 128x64 grafičkog LCD displej, 2x16 alfanumerički LCD displej, piezo buzzer, USB-UART itd. Da bi normalno funkcionisao UNI-DS6, potrebno je da se obezbedi napajanje preko AC/DC konektora CN19.



Slika 5. UNI-DS6

MikroBoard ima igrađen programator za MCU programiranje razvojnih sistema. Potrebno je samo povezati mikroBoard sa kompjuterom preko USB kabela i instalirati odgovarajući softver. Za programiranje i implementaciju se koristi [mikroC PRO for PIC](#) i [mikroProg Suite For PIC](#).

## 4. REALIZACIJA PROJEKTOG ZADATAKA

UNI-DS6 sadrži i 72 (8x9) LED diode koje se koriste za vizuelno prikazivanje stanja mikrokontrolera. Da bi neki LED zasvetleo, potrebno je izabrati odgovarajući port (PORTA, PORTB, PORTC, PORTD, PORTE ili PORTF/G) pomoću DIP prekidača SW12.

Programiranje se obavlja u programu mikroC PRO, koji ima sintaksu programskog jezika C. Na početku se usmeravaju portovi A, B, C i D da budu izlazni tako što im se zadaje vrednost 0 (za PORTA se unosi „TRISA = 0x00“). Potom se LED diode na svakom pinu postavljaju da budu ugašene komandom „LATX = 0x00“. Kada su određene početne vrednosti, može se preći na programiranje.

```

mikroC PRO for PIC v.5.6.1 - D:\mikroC PRO\LED_projekat\led_projekat.mcppi - NOT REGISTERED
File Edit View Project Build Run Tools Help
1024x768
led_projekat.c
Project Settings
Code Explorer

void main() {
    TRISA = 0x00;
    10 TRISB = 0x00; // pinovi se postavljaju kao izlazni
    TRISC = 0x00;
    TRISD = 0x00;

    LATA = 0x00;
    LATB = 0x00; // LED na svakom pinu se postavlja na pocetnu vrednost
    LATC = 0x00; // odnosno na nulu
    LATD = 0x00;

    20 while (1) {
        for (brojac = 0; brojac < 8; brojac++){
            LATA = 0x00;
            kasnjenje();
            LATA = 0x02; // prolazi se kroz for petlju 8 puta
            LATB |= 1 << brojac; // i pri svakoj inicijalizaciji se pali sledeci red
            LATC = 0x00; // dioda u kolonama B i D koje prati
            kasnjenje(); // paljenje i gasenje dioda u drugom redu kolona A i C
            LATC = 0x02;
            30 LATD |= 1 << brojac;

            kasnjenje();
        }

        brojac = 0; // Kada se upale sve diode, brojac se vraca na nulu
        while (brojac < 8) {
            LATB &= ~(1 << brojac); // Sledi gasenje dioda u kolonama B i D, dok su
            40 LATD &= ~(1 << brojac); // diode A i C u drugom redu upaljene

            kasnjenje();
            brojac++;
        }
    }
}

```

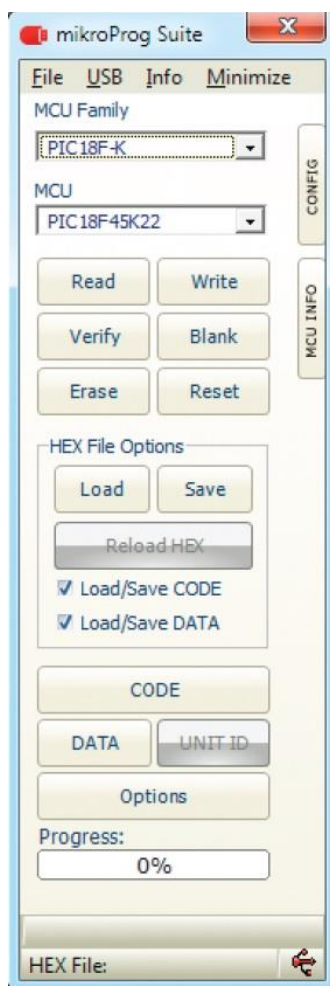
Slika 6. MikroC PRO kod

Projekat sadrži jednu beskonačnu *while* petlju (*while (1)*) koja se sastoji od *for* i *while* petlji. Prvo ide *for* petlja u kojoj se pri svakoj inicijalizaciji uključuju diode u narednom redu (od 1 do 8), samo u kolonama B i D (druga i četvrta kolona). Uključivanje dioda u koloni B, prati dioda RA1 svojim naizmeničnim uključivanjem i isključivanjem. Po istom principu dioda RC1 prati uključivanje kolone D. Između svakog uključivanja je uvedeno kašnjenje od 250ms.



Nakon 8 prolaza kroz for petlju, sve diode u kolonama B i D su uključene i for petlja je završena. Potom se prelazi na while petlju koja služi za isključivanje dioda u drugoj i trećoj koloni. Ova petlja takođe ima 8 inicijalizacija, jer se pri svakom prolazu isključuje jedan red.

Potrebno je isprogramirati mikroBoard uz pomoć programa mikroProg Suite. Kada se uključi mikroProg, potrebno je izabrati tačnu MCU familiju i model mikrokontrolera koji se koristi. Takođe je bitno da se obrati pažnja na oznaku za USB u donjem desnom uglu prozora. Ona pokazuje da li je mikrokontroler priključen na PC preko USB-a. Ako je oznaka crvene boje, to znači da je prepoznat mikrokontroler.



Slika 7. MikroProg Suite

Klikom na Load, otvara se novi prozor u kome treba izabrati fajl koji želimo da uploadujemo u MCU memoriju. Ovaj fajl treba da ima .hex ekstenziju. Nakon odabira fajla, potrebno je još samo kliknuti na Write i sačekati dok se završi proces programiranja mikrokontrolera.

## **5. ZAKLJUČAK**

Ovaj projekat je primer kako se može manipulirati razvojnim sistemom UNI-DS6 uz pomoć mikrokontrolera PIC18F4520. Prednost ovih mikrokontrolera je što oni ekonomično kontrolišu digitalne procese za više uređaja. Princip manipulacije LED dioda u ovom projektu je sličan principu putujućeg teksta. Putujući ili pokretni tekst ima široku primenu jer se preko njega mogu poslati bitne informacije koje su dosta uočljive. Pored LED dioda, ovaj tekst se može koristiti i na LCD ekranima. Najčešće se koristi kao reklama, na aerodromima, autobuskim i drugim prevoznim stanicama...

## 6. LITERATURA

- mikroElektronika. (2009. god). “*mikroC PRO for PIC*”
- mikroElektronika. (2009. god). “*mikroProg for PIC Manual ver. 1.00*”
- Microchip Technology Inc. (2007. god). “*PIC18F4520 Data Sheet*”
- <http://www.futurlec.com/Microchip/PIC18F4520.shtml>
- mikroElektronika. (2009. god). “*QUICK REFERENCE GUIDE FOR mikroC*”