

Elektronski Fakultet, Niš
Katedra za Elektroniku
Mikroprocesorski sistemi

Seminarski rad

Mikrokontroler AT89S8253 sa
Tajmerom Intel 82c54, kao periferijom

Niš, 2.9.2007.

Petrovi Radovan
10052

Sadržaj

1. Uvod.....	3
2. Struktura sistema.....	4
- Šta su mikrokontroleri.....	5
- Mikrokontroleri: tipovi arhitektura i blok šema.....	6
- Mikrokontroleri u odnosu na mikroprocesore.....	7
- Struktura mikroprocesora.....	9
- Struktura mikrokontrolera.....	9
- Razlike između mikroprocesora i mikrokontrolera.....	10
- Šta su tajmeri.....	11
- Opis sistema.....	14
- Periferija – Tajmer 82c54.....	17
3. Software.....	21
- Opis kôda.....	21
- Programiranje tajmera 82c54.....	23
- Programiranje mikrokontrolera.....	26
4. Primeri za Tajmer 82c54.....	29
5. Zadatak za studente.....	44
6. Zaključak.....	50
7. Literatura.....	50

1. Uvod

Nagli razvoj elektronike je omogućio njenu primenu u svakodnevnom životu. Nije moguće zamisliti ni jedan trenutak života bez "mnoštva" izuma. Raunari, kao primer pružaju pomoć u smislu informisanosti, mobilnosti, rešavanja matematičkih problema, projektovanja elektronskih kola, nadzora, kontrole procesa, obrade signala. Raunari na čipu – mikrokontroleri se odlikuju kompaktnošću. Pomoću mikrokontrolera, možemo da utičemo, po sopstvenoj zamisli, na ono na šta želimo. Precizno upravljanje elektro motorima, elektronskih kola, kontrola raznih displeja, rasvete. Mogućnosti su velike.

Ovaj projekat se bavi kontrolom tajmera 82c54, koji, tako, daje određena vremenska kašnjenja. Potrebno je realizovati mikroprocesorski sistem baziran na mikrokontroleru AT89S8253 i, kao periferiju, povezati tajmer 82c54 i programirati njegov tajmer 0 da podeli ulaznu frekvenciju od 500 KHz sa 500. Tako podeljena frekvencija se iskoristi za tajmer 1 i 2, u cilju analize svih režima rada.

2. Struktura sistema

Najvažniji faktor koji je imao dominantni uticaj na masovno korišćenje računara u industriji predstavlja razvoj mikroprocesora. Mikroprocesor je u suštini računar na čipu.

Da bi obavili koristan zadatak, sistemi zasnovani na mikroprocesorima prve generacije zahtevali su ugradnju velikog broja dodatnih komponenata (oscilatore, adresne registre, registre za podatke i druga LSI kola relativno male složenosti). Kasnije, napredak na polju VLSI tehnologije doveo je do proizvodnje mikro računarskih sistema kod kojih su sva neophodna kola bila ugrađena (*embedded*) u relativno mali broj pratećih integrisanih kola (tipično su to bili integrisane CPU jedinice relativno velike složenosti, kontroleri prekida, DMA kontroleri, tajmeri, UART-i, programibilni paralelni periferni interfejsi (PIA), i dr.). Kao plod dalje integracije nastaju jednočipni mikrokontroleri koji u sebi imaju ugrađeno analogni i digitalni ulazno-izlazni podsistem, tajmere i brojače, LCD drajvere i druge logike. Time su ova kola, takorekćeno i bez dodatnog hardvera, sada bila u stanju da obavljaju relativno složene upravljačke funkcije koje se odnose na rad sistema u realnom vremenu. Primeri ovakvih tipova mikrokontrolera su Intel 8051 i 8096, Motorola MCH 68HC11, Microchip PIC 18F1x20, i dr. Ovi čipovi su pre svega razvijeni za potrebe automobilske industrije, procesne industrije, elektromedicine, telekomunikacija, robe široke potrošnje, i td.

Mikrokontrolere delimo na 4-, 8-, 16-, i 32-bitne. Ilustracije radi, 4-bitni mikrokontroleri uglavnom se koriste u automatima za igru, 8-bitni kao programatori u većim mašinama, 16-bitne srećemo u sistemima za upravljanje klimom u kolima, a 32-bitni se primenjuju da obave složene telekomunikacione funkcije u mobilnim bežičnim uređajima.

Osnovne karakteristika mikrokontrolera su:

1. ugrađeni (*built-in*) ROM- naj veća kapaciteta 4 kB na samom čipu u kome se čuva upravljački program,
2. ugrađeni RAM- naj veća kapaciteta 128 B, koji se koristi za čuvanje privremeno promenljivih,
3. CPU koji je, zbog efikasnijeg korišćenja ograničene programske memorije, u stanju da izvršava instrukcije koje manipulišu sa jedinim bitovima,
4. veliki broj mikrokontrolera ima ugrađeni koprocesor *Boole*-ovog tipa. Ovaj koprocesor zajedno sa CPU-om ima mogućnost da pojednostavljeno implementira *Boole*-ove izraze koji se veoma često javljaju u upravljačkim aplikacijama,
5. mikrokontroleri imaju ugrađene ulazno-izlazne portove koji se koriste za efikasno upravljanje, kao i jednostavnu i laku interakciju sa spoljnim uređajima.

U daljem tekstu ukazujemo na osnovne karakteristike mikrokontrolera bez ulaženja u detalje koji se odnose na specifičnosti njihovih realizacija.

Šta su mikrokontroleri ?

Svi današnji računari su realizovani od istih gradivnih blokova. To su centralni procesor ili CPU, memorijski podsistem, ulazno-izlazni podsistem, generatora taktne pobude, i sistemska magistrala koja me usobno povezuje pomenute gradivne blokove. Sistemska magistrala se sastoji od adresne magistrale, magistrale podataka, i upravljačke magistrale.

Najveći i broj standardnih računara, kakav je na primer PC mašina, se smešta u jedinstveno kućište. Periferije kakvi su štampa, displej, skener, tastatura, miš, i druge se povezuju na računarski sistem preko konektora koji su montirani na kućištu. Druge sistemske komponente kakvi su diskovi, memorije proširenja, mrežne kartice i dr., su locirane u samom kućištu ili pak na osnovnoj ploči (*motherboard*) računara.

Mikroprocesor je integrisano kolo ili čip lociran na osnovnoj ploči koje u suštini predstavlja CPU računarskog sistema. Uglavnom svet PC mašina je taj koji je bio glavni pokretač razvoja sistema baziranih na mikroprocesorima. Imaju i u vidu ovakve zahteve za mikroprocesore kažemo da su u novije vreme postali relativno specijalizovane komponente, posebno pogodne (prilagođene) za veoma brzo manipulisanje podacima.

Ne kod određeni aplikacija veoma često je neophodno ugraditi određeni iznos lokalnog procesiranja na licu mesta procesa. Tako na primer, automat za izdavanje karata na autobuskim stanicama treba da ima određeni nivo autonomnosti u radu (pameti) tako da za unete podatke preko lokalne tastature sam izračuna cenu karte i izda je. Zbog ovakvog načina rada računarske mašine opšte namene ne predstavljaju dobro rešenje (izbor) za ovakav tip aplikacije, prvenstveno zbog velikog gabarita (dimenzije) mašine, a tako je i zbog visoke cene. Upravo su ovo idealne aplikacije za mikrokontrolere. Nasuprot mikroprocesorima, mikrokontroleri su više orijentisani ka postizanju superiornijih performansi kod aplikacija namenjanih upravljanju, bezbednosti i pouzdanosti u radu. Imaju i u vidu da je cena uređaja baziranih na mikrokontrolerima od izuzetne važnosti za krajnjeg korisnika, mikrokontroleri su integrisali na čipu veliki broj standardnih periferala, kontrolere raznih tipova uređaja, kao i memoriju. Na osnovu prethodnog mogli bi da kažemo sledeće: **Mikrokontroler je računar na čipu**. On sadrži skoro sve osnovne gradivne blokove računarskog sistema (CPU, ROM, RAM, U/I periferije tipa AD i DA konvertore, serijske komunikacione interfejsne (UART), paralelne interfejsne portove (PIA), tajmere/brojila, LCD drajvere, i dr.) koje se mogu integrisati na jedinstvenom čipu. Danas su mikrokontroleri najviše prodavan tip procesora. Ne bez razloga projektanti sistema relativno male složenosti kažu da su to čipovi upravo projektovani po njihovoj meri, tj. čipove o kojima sanjaju.

Nasuprot mikroprocesorima, cena mikrokontrolera je veoma niska (do nekoliko dolara u odnosu na cenu od stotinu dolara kakva je cena mikroprocesora). Za određene aplikacije niska cena mikrokontrolera je idealno rešenje.

Taktna frekvencija na kojoj rade, obim adresibilne memorije, i obim podataka sa kojima manipulišu varira od jednog mikrokontrolera do drugog, šta više i kod jednog istog proizvođača, tako da je posao projektanta da izabere najbolji mikrokontroler za datu aplikaciju.

Drugi važan razlog korišćenja mikrokontrolera, u odnosu na PC mašinu, predstavlja njegova kompaktnost koja se može iskazati konstatacijom: *Računar na jednom čipu*. Svi mikrokontroleri imaju usmenu memoriju na čipu kao i veći broj ulazno-

izlaznih interfejs linija. Veliki broj mikrokontrolera poseduje AD i DA konvertore, impulsno-širinski modulirane (PWM) generatore, sofisticirani sistem prekida, veći broj serijskih i paralelnih ulazno-izlaznih portova, fleksibilni sistem tajmer-broja događaja, LCD drajver, i dr.

Mikrokontroleri se danas koriste u *embedded* sistemima za upravljanje raznim funkcijama koje sistem treba da obavi. Na primer, u današnjim automobilima ugrađuje se veliki broj (reda 100) mikrokontrolera koji se koriste za upravljanje radom koionog sistema, ubrizgavanjem goriva, klima sistemom za grejanje-hlađenje, prikazom informacije na pokaznoj tabli, i td. Drugim rečima, jednoipni računari omogućavaju projektantima da ugrade računare u bilo kom uređaju koji postavlja makar i minimalne zahteve za određenim iznosom izdatka.

Mikrokontroleri se veoma često koriste i u situacijama kada treba da se veoma brzo reaguje na spoljne signale- posebno ako se takvi sistemi koriste za rad u realnom vremenu- pa zbog toga koriste procesiranje bazirano na obradu prekida. Mikrokontroleri mogu takođe da rade po nadzorom operativnih sistema za rad u realnom vremenu (RTOS) čiji je obim manji, pri čemu ovakvi sistemi bolje reaguju (za kraće vreme) na odzive sistema koji se odnose na rad u realnom vremenu, nego što je to slučaj sa operativnim sistemima opšte namene koji se instaliraju na većiim mašinama kakve su recimo PC mašine.

Mikrokontroleri: tipovi arhitektura i blok šema

Arhitekture mikrokontrolera se mogu podeliti na sledeća dva tipa:

1. *Harvard* tip- karakteriše se razdvojenim memorijama za program i podatke. Svaki tip memorije ima svoj sopstveni fizički adresni prostor i koristi sopstvenu internu adresnu magistralu. Prednost ove arhitekture je ta što se pristup programskoj memoriji i memoriji za podatke ostvaruje konkurentno. Običnim pristupom se skraćuje ukupno vreme izvršenja programa. Proizvođači mikrokontrolera Microchip, Zilog, National Semiconductor i Cypress zasnovavaju svoje proizvode na *Harvard* arhitekturi, mada postoje neke neznatne ali važne razlike u pristupima kako ove kompanije implementiraju arhitekturu. Arhitekture ovih mikrokontrolera su tipične za RISC pristup.
2. *Princeton* tip (tzv. von Neumann)- programska memorija i memorija za podatke dele isti adresni prostor. To znači da se instrukcioni kôd može izvršavati kako iz programske tako i iz memorije za podatke. Poznati proizvođači ovih tipova mikrokontrolera su Intel, Atmel, Motorola i dr. Ključna osobina ovih proizvoda je povećan broj instrukcija i adresnih linija rada koji u kombinaciji sa relokabilnim softverskim magacinom (*stack*) omogućavaju efikasno korišćenje programskog jezika visokog nivoa C. Arhitekture ovih mikrokontrolera su tipične za CISC pristup.

Mikrokontroleri i digitalni signal procesori (DSP) su poznati tipovi aplikaciono integrisanih procesora (ASIP). U suštini mikrokontroler je mikroprocesor koji je optimiziran za *embedded* upravljanje aplikacijama. Kod ovakvih aplikacija se obično nadgledaju (monitorišu) i postavljaju brojni jedno-bitni upravljački (*control*) signali, a pri

tome se ne obavlja neko intenzivno izra unavanje nad podacima. Zbog ovoga mikrokontroleri imaju jednostavne staze podataka (*datapaths*) koje su prilago ene za brzo izvršenje operacija na nivou-bita (*bit manipulation*) kao i operacija itanja i upis sadržaja bitova sa spoljnih pinova ipa.

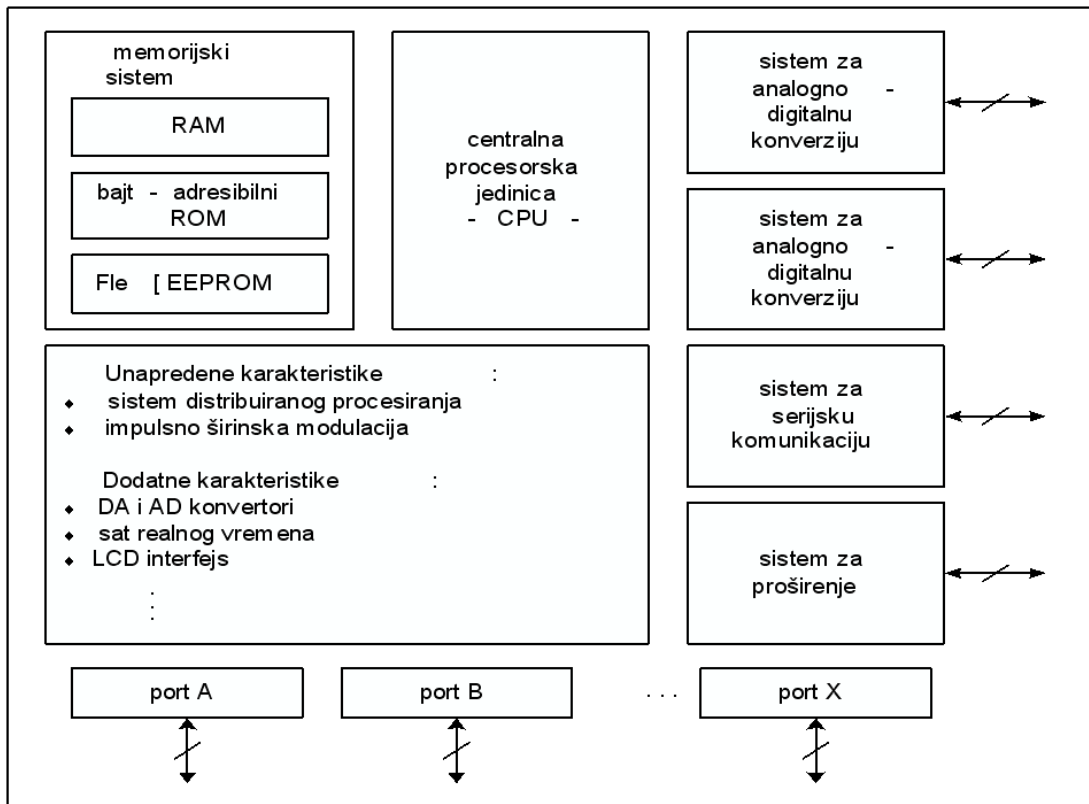
Pored jednostavne CPU, na istom ipu, mikrokontroleri imaju inkorporirano nekoliko periferalnih komponenata tipi ne za upravlja ke aplikacije, kakve su serijske komunikacione periferije, tajmere/broja e doga aja, PWM generatore, AD i DA konvertore, i dr. Pored ostalog programska memorija i memorija za podatke su tako e inkorporirane na samom ipu.

Inkorporiranjem periferala i memorije na istom integrisanom kolu smanjuje se broj dodatnih integrisanih kola koje treba ugraditi u sistem, što rezultira kompaktnijoj implementaciji i globalno posmatrano smanjenoj potrošnji, tj rešenje postaje tipa *low-power*.

injenica da se programskim putem može direktno pristupati spoljnim pinovima ipa omogu ava da program može lako da monitoriše stanje senzora, postavlja u definisano stanje aktuatora, i vrši prenos podataka ka/iz drugih ure aja.

Veliki broj proizvo a a naziva mikrokontrolere *embedded* procesore. Razlika izme u *embedded* procesora i mikrokontrolera esto nije jasna, mada se u stru noj literaturi termin *embedded* procesor naj eš e koristi za ve e (32-bitne) procesore, a termin mikrokontroler za manje (4-, 8-, i 16-bitne) procesore.

Opšti blok dijagram mikrokontrolera je prikazan na slici 1.

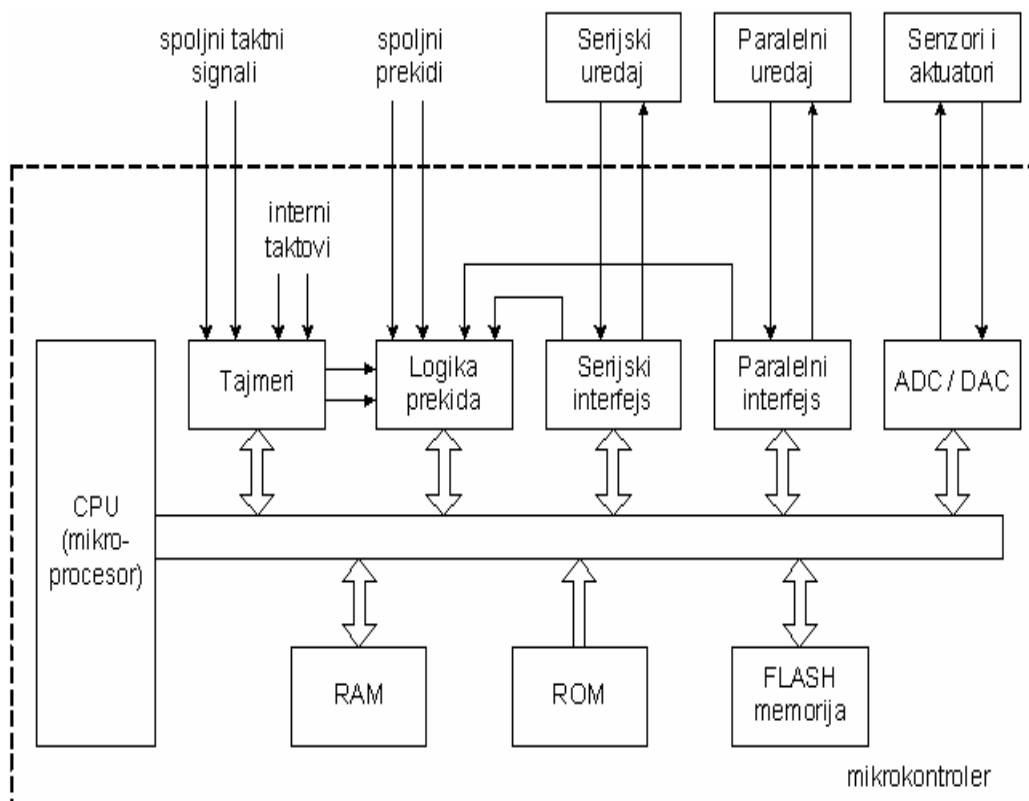


Slika 1. Blok dijagram mikrokontrolera

Mikrokontroleri u odnosu na mikroprocesore

Da bi ukazali na to kakva razlika postoji između mikroprocesora i mikrokontrolera analiziramo sliku 2 koja predstavlja jedan detaljan blok dijagram mikroracunarskog sistema.

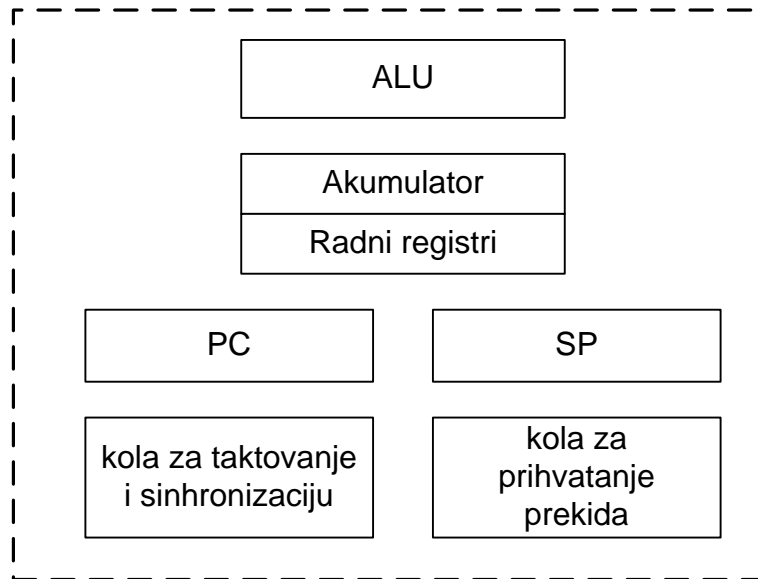
Dok je mikroprocesor (CPU) integrirana komponenta na jedinstvenom čipu, sam mikrokontroler, na jedinstvenom čipu ima integrirano CPU, RAM i ROM memoriju i ostale ulazno-izlazno orijentisane gradivne blokove (paralelni i serijski interfejsi, tajmeri, logika za prihvatanje prekida, A/D i D/A konvertore i dr.).



Slika 2. Detaljni blok dijagram mikroracunarskog sistema

Struktura mikroprocesora

Na slici 3 prikazan je blok dijagram mikroprocesora. CPU ine slede i blokovi: ALU, PC, SP, odre eni broj radnih registara, kola za taktovanje i sinhronizaciju i kola koja se koriste za prihvatanje zahteva za prekid.



Slika 3. Blok dijagram mikroprocesora(CPU-a)

Da bi se kompletirao mikrorra unarski sistem pored mikroprocesora potrebno je dodati ROM, RAM memorijske dekodere, oscilator, odre eni broj ulazno-izlaznih ure aja kakvi su paralelni i serijski portovi za podatke, A/D i D/A konvertore i drugo. Pored ulazno-izlaznih ure aja specijalne namene, esto se javlja i potreba da se ugrade i kontroleri prekida, DMA kontroleri, kao i brojac/tajmeri iji je zadatak da oslobode CPU-a od obavljanja U/I aktivnosti. Kada se u sistem instaliraju i ure aji za masovno memorisanje (hard disk, CD drajver), kao i tastatura, mis i CRT displej, tada se taj "mali racunar" moze koristiti za razlicite aplikacije opste namene.

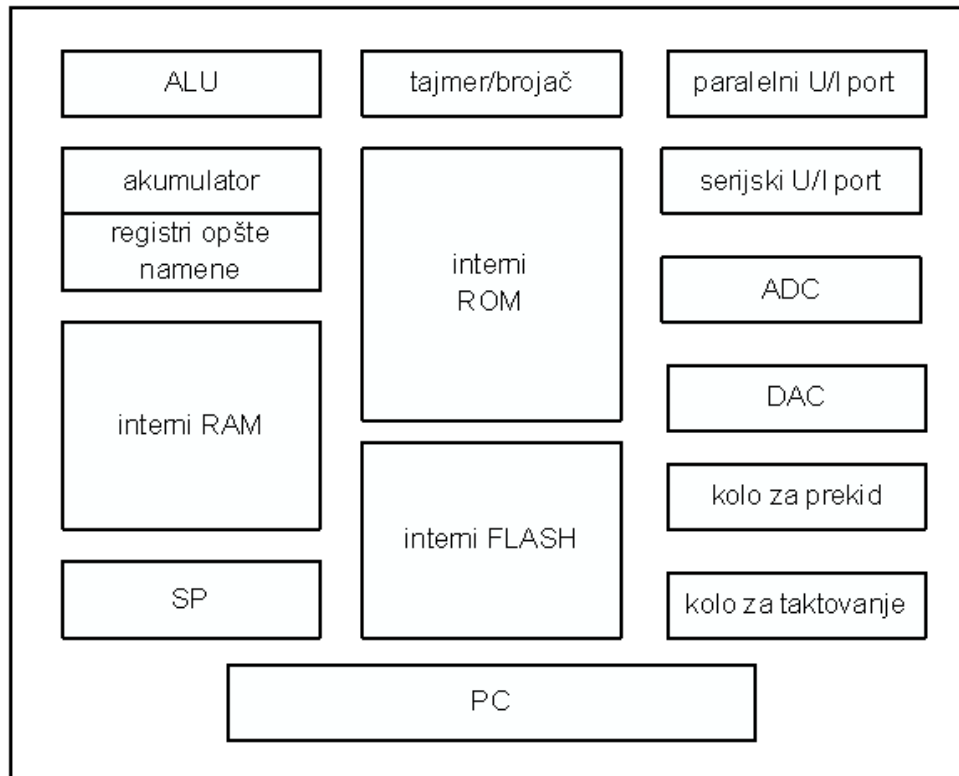
Osnovna namena CPU-a je da pribavlja podatke, obavlja izra unavanja nad podacima i memorise rezultate izra unavanja na disku, kao i da za potrebe korisnika prikaže te rezultate na displeju (CRT, TFT, LED i dr.). Programi koje koristi mikroprocesor memorisani su na disku odakle se itaju i smeštaju u RAM. Deo programa, naj eš e malog obima, se obi no smešta i u ROM-u.

Struktura mikrokontrolera

Blok dijagram mikrokontrolera prikazan je na slici 4. Mikrokontroler je u suštini pravi "mali ra unar" na ipu, koji sadrzi sve gradivne blokove CPU-a (ALU, PC, SP, registre i dr.), ali tako e i RAM, ROM, paralelne i serijske U/I portove, generatore takta i dr.

Kao i mikroprocesor, i mikrokontroler je ure aj opšte namene, koji pribavlja podatke, obavlja ograni enu obradu nad tim podacima, i upravlja svojim okruženjem na

osnovu rezultata izra unavanja. Mikrokontroler u toku svog rada koristi fiksni program koji je smešten u ROM-u i koji se ne menja u toku životnog veka sistema.



Slika 4. Blok dijagram mikrokontrolera

Mikrokontroler koristi ograničen skup jedno- ili dvo-bajtnih instrukcija koje se koriste za pribavljanje programa i podataka iz interne memorije. Veliki broj ulazno-izlaznih pinova mikrokontrolera se može koristiti za više namena, što se softverski definiše.

Mikrokontroler komunicira sa spoljnim svetom (pribavlja i predaje podatke) preko svojih pinova, pri čemu je arhitektura i skup instrukcija projektovan za manipulisanje sa podacima obima bajt ili bit.

Razlike između mikroprocesora i mikrokontrolera

Razlike su brojne ali one koje su najvažnije su sledeće:

1. Mikroprocesori najčešće su CISC tipa za kopiranje podataka iz spoljne memorije u CPU koriste veći broj op-kôdova, dok mikrokontroleri jedan ili dva.
2. Za manipulisanje sa podacima tipa bit mikroprocesori koriste jedan ili dva tipa instrukcija, dok kod mikrokontrolera taj broj je veći.
3. Mikroprocesori su projektovani za brzi prenos podataka iz programa sa spoljno adresiranih lokacija u 32-bit, dok se kod mikrokontrolera brzi prenos bitova obavlja u okviru 8-bit.

4. Mikrokontroler može da funkcioniše kao računalo bez dodatka spoljnih gradivnih blokova (memorije i U/I uređaja), dok operativnost mikroprocesora bez spoljne memorije i U/I podсистema nije moguća.

Šta su tajmeri ?

Tajmer-kolo je periferno kolo koje je u stanju da meri vremenske intervale. Ovakvo kolo se može koristiti da:

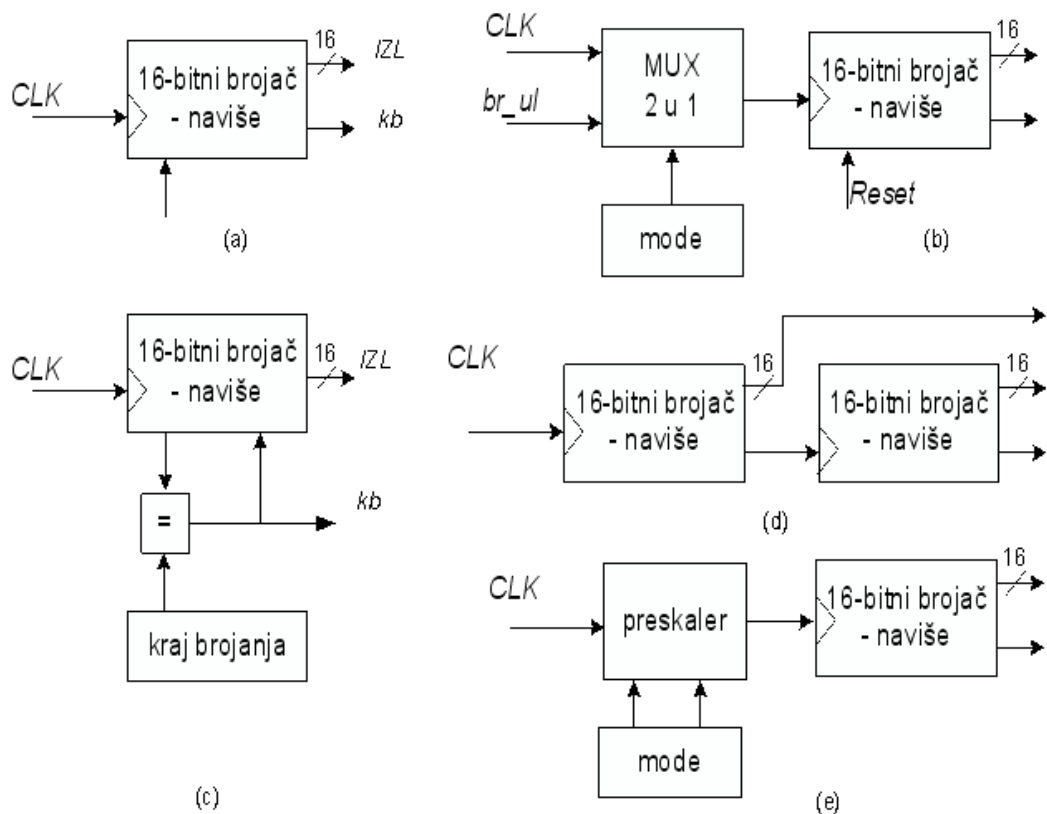
- a) generiše događaje u specifičnim vremenskim trenucima. Tipične aplikacije su one koje se odnose na uključivanje crvenog/žutog/zelenog svetla na semaforu za specifični vremenski period, recimo crveno svetlo traje 20 s, žuto 3 s, a crveno 15 s.
- b) odredi vreme trajanja između dva spoljašnja događaja. Obične aplikacije ovakvog tipa srećemo kod merenja brzine kretanja vozila, tj., meri se vremenski interval koji je potreban da bi limuzina prešla između dva senzora na putu, pri čemu su senzori razmaknuti za poznato rastojanje.

U principu treba praviti razliku između pojmova tajmer i brojača, a ona se sastoji u sledećem:

1. Tajmer je kolo koje meri vreme putem odbrojanja taktih impulsa poznate periode koji se dovode na ulazu ovog kola. Tako na primer, ako je perioda pobudnog taktog impulsa $1 \mu\text{s}$, a izbrojano je 5000 pobudnih taktih impulsa, to znači da je prošlo vreme od 5 ms.
2. Brojač predstavlja generalniju verziju tajmera. Naime, umesto da odbrojava taktne impulse generisani od strane internog oscilatora CPU-a, brojač broji impulse koji su generisani od strane nekog drugog, obično spoljnog, generatora signala. Tako na primer, brojač se može koristiti za brojanje broja vozila koja prolaze na nekoj deonici autoputa, pri čemu je svaki prolaz detektovan od strane odgovarajućeg senzora. Brojač se takođe može koristiti i za brojanje ljudi koji ulaze ili izlaze u/iz neke prostorije.

Veoma često u cilju merenja brzine okretanja nekog tela moguće je kombinovati korišćenje brojača i tajmera. Tako na primer, kod merenja brzine kojom se okreće rotor neke mašine tajmer se može koristiti za generisanje referentnog vremenskog intervala merenja, recimo 1 s, a brojač za merenje broja impulsa koji odgovaraju ukupnom broju okretanja rotora u trajanju od 1 s.

Kada projektant hoće da koristi tajmer-kolo neophodno je prvo da programer konfigurise njegove ulaze, a zatim da nadgleda njegove izlaze, tj. da ga konfigurise kao tajmer ili brojač. Da bi se obavio ovaj zadatak potrebno je poznavati internu strukturu tajmer-kola, koja može u značajnoj meri da varira od jednog do drugog proizvođača. Standardne strukture koje se odnose na izvođenje tajmer-kola prikazane su na slici 5.



Slika 5. Strukture tajmer-kola: (a) osnovni tajmer; (b) tajmer/brojač događaja; (c) tajmer sa krajem brojanja; (d) 16/32-bitni tajmer; (e) tajmer sa preskalerom

Napomena: CLK- ulaz taktnog signala (impulsi generisani od strane internog oscilatora CPU-a); kb- izlaz kraj brojanja; br_ul- brojački impulsi generisani od strane spoljnog izvora; mode- režim rada; "="- komparator; MUX- multiplexer

Struktura jednostavnog tajmera je prikazana na slici 6 a). Tajmer se sastoji od jedan 16-bitni brojač koji inkrementira svoju vrednost nakon nailaska svakog taktnog impulsa. Izlazna vrednost brojača *izl* odgovara broju pobudnih impulsa CLK nakon što je brojač zadnjeg puta bio resetovan na nulu. Da bi interpretirali ovaj broj kao vremenski interval neophodno je da znamo frekvenciju ili periodu taktnog pobudnog signala CLK. Tajmer sa slike 6 a) ima jedan dodatni ulaz *kb*. Trenutak generisanja impulsa *kb* na ovom izlazu ukazuje da je brojač dostigao svoju vršnu vrednost brojanja, alternativno nazvana premašaj. Nakon ovog trenutka brojač prelazi u stanje kada je *izl* postavljen na nulu, a nailaskom narednih pobudnih impulsa CLK produžava sa brojanjem. Obično se izlaz *kb* dovodi na ulazni pin INTR CPU-a. U tom slučaju uslužna rutina za obradu prekida odbrojava broj puta za koji je ta rutina pozvana, ime se na jedan indirektna na in procenjuje protekli vremenski period. Veliki broj mikrokontrolera ima usloveno tajmere u svoju strukturu, a takođe i specijalne INTR linije za tajmere koje se razlikuju od spoljnih INTR linija. Sa aspekta korišćenja tajmera neophodno je da projektant razume korišćenje sledeća dva parametra. Prvi se odnosi na opseg (range), koji predstavlja maksimalni vremenski interval koga tajmer može da odbroji, a drugi rezolucija, to je minimalni interval koga tajmer može da odbroji. Ilustracije radi, neka je frekvencija signala CLK 10

MHz, to znači da je perioda ovog signala 100 ns, i ako izlaz tajmera u trenutku kada smo njegovu vrednost pročitali iznosi $izl = 40000$, to znači da je od početka resetiranja do tog trenutka prošlo $40 \cdot 10^3 \cdot 100 \cdot 10^{-9} = 4$ ms. S obzirom da je broj $broj$ sa slike 1 a) 16-bitni, on može da broji od vrednosti 0 do 65535. Za konkretan slučaj mereni opseg se kreće od 0 do $65535 \cdot 100$ ns = 6,5535 ms sa rezolucijom od 100 ns.

Struktura sa slike 6 b) predstavlja nešto unapred napravljeniju verziju tajmera koji se takođe može konfigurisati kao broj $broj$. Izlaz jedno-bitnog registra mode (postavlja se softverski) se dovodi na ulaz multipleksera MUX. Multiplekser bira trenutnu pobudu za 16-bitni broj $broj$. Kada je selektovan interni ulazni signal CLK struktura sa slike 6 b) se ponaša kao tajmer, a za slučaj da je selektovan spoljni signal br_ul struktura se ponaša kao broj $broj$ odbrojavajući i pri tome pojavu impulsa na ulazu br_ul . Signal na ulazu br_ul se obično dovodi od nekog spoljnog izvora, pri čemu nailazak impulsa na ovom ulazu se obično dešava u vremenskim neodređenim (nedefinisanim) intervalima (kakav je recimo izlaz detektora prolaska vozila na autoputu). To znači da u konkretnom slučaju nije moguće meriti vremenski interval kada se broje impulsi na ulazu br_ul ako njihova perioda nije poznata.

Struktura tajmera sa slike 6 c) se koristi da informiše sistem kada je istekao pojedini vremenski interval. Registar kraj-brojanja (postavlja se softverski) čuva vrednost broja koja odgovara željenom vremenskom intervalu. Na primer, ako je perioda podudnog signala jednaka 100 ns, a želimo da generišemo trajanje od 4 ms, tada treba da odbrojimo 40000 impulsa. Sastavni deo struktura sa slike 6 c) je kolo komparatora koje generiše impuls kb . Impuls kb se koristi da resetuje broj $broj$ na nuli i informiše korisnika tajmera da je istekao željeni vremenski interval. Izlaz kb se najčešće vezuje na ulaz INTR CPU-a. Prekidna rutina u tom slučaju preuzima odgovarajuće akcije koje se odnose na istek specificiranog vremenskog intervala.

Sa ciljem da se poveća efikasnost strukture sa slike 6 c), umesto da tajmer broji naviše od vrednosti 0 pa do specificirane (softverski definisane) vrednosti u registar kraj-brojanja se upisuje specificirana vrednost pa se sada umesto tehnike brojanja unapred koristi tehnika brojanja unazad. To znači da se nakon resetiranja broj $broj$ postavlja na specificiranu vrednost odbrojavanja. U konkretnom slučaju broj odbrojavanja se detektuje kada vrednost broja $broj$ dostigne vrednost nula. Hardver za detekciju nule se realizuje kao 16-bitno (ulazno) NOR kolo (ili kombinaciona mreža koja obavlja istu funkciju) koje je sa aspekta površine na Silicijumu i potrošnje-energije znatno efikasnije rešenje u odnosu na 16-bitni komparator.

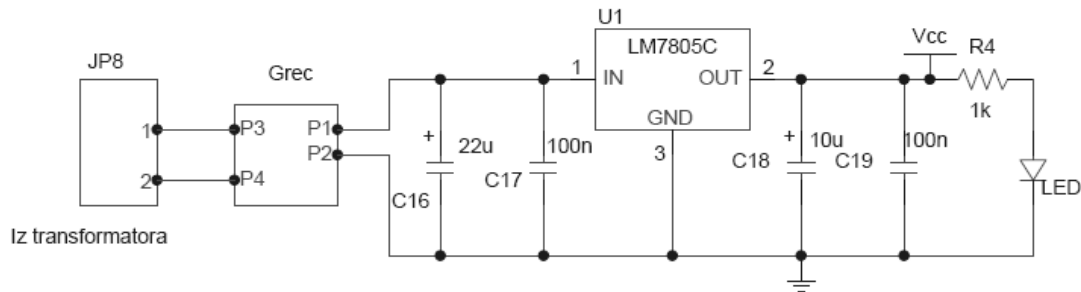
Struktura sa slike 6 d) obezbeđuje da se tajmer može konfigurisati kao 16- ili 32-bitni tajmer. Tajmer koristi izlaz $kb1$ sa svog prvog 16-bitnog broja $broj$ naviše kao taktni ulaz za drugi 16-bitni broj $broj$. Ova veza se naziva kaskadna veza broja $broj$.

Na slici 6 e) prikazan je tajmer koji u svojoj strukturi ima ugrađeno i preskaler. Preskaler je u suštini konfigurabilno kolo koje deli taktni pobudni signal CLK. U zavisnosti od softverski postavljene vrednosti u registru mode moduo deljenja preskalera može biti 1, 2, 4, 8, itd. Na ovaj način preskaler se koristi da proširi opseg tajmera po ceni redukcije rezolucije. Tako na primer, ako je frekvencija impulsa CLK jednaka 10 MHz, što odgovara rezoluciji od 100 ns, a moduo deljenja preskalera je 8, to znači da tajmer broji u stanju da meri vremenski interval u opsegu od 0 do 52,427 ms sa rezolucijom od 800 ns.

Strukture tajmera kod različitih mikrokontrolera mogu da variraju. Uglavnom se modifikacije odnose na ugradnju dodatnih konfigurabilnih osobina (karakteristika). Jedna od tih osobina se odnosi na ugradnju dodatnog mode-bit flip-flopa koji dozvoljava ili zabranjuje brojanje. Druga karakteristika se tiče ugradnje flip-flopa koji dozvoljava ili brani generisanje prekida kada brojka dostigne kraj brojanja.

Opis sistema

Napajanje koje se koristi je sa transformatora 220/9 V i taj napon se ispravlja preko Greca ispravljača i stabilizuje sa LM7805 na 5V, odakle se vodi na mikrokontroler i spoljašnji tajmer 82c54.



Slika 8. Električna šema napajanja

Koristi se mikrokontroler AT89S8253 sa kristalom od 12 MHz. Izbor ovog kristala je zbog celobrojne vrednosti frekvencije, koja je potrebna radi preglednosti talasnih oblika signala na izlazu sistema.

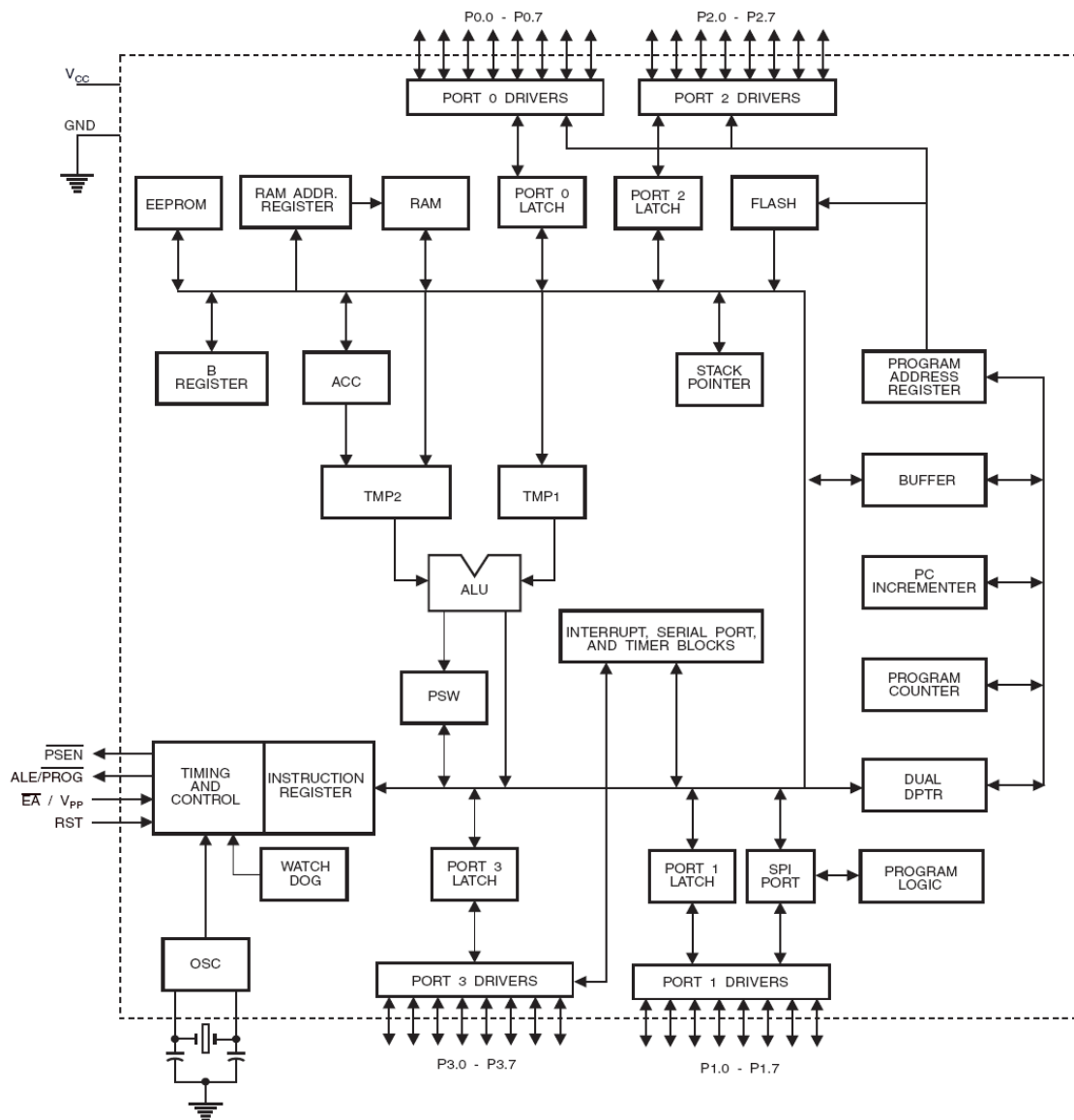
AT89S8253 odlikuju:

- 12 kB ISP fleš program memorija - 10000 read/write ciklusa
- 2kB EEPROM - 100000 write/read ciklusa

- SPI serijski interfejs za učitavanje programa
- 64B korisni ka matrica
- Napon napajanja 2.7 do 5.5 V
- Frekvencija od 0 do 24 MHz
- Memorijsko zaključavanje u 3 nivoa
- 256 x 8-bit-ni unutrašnji ram
- 32 programabilne I/O linije
- Tri 16-bitna brojača/tajmera
- Devet izvora prekida
- Poboľjšani UART serijski port
- Poboľjšani SPI interfejs
- Male snage napajanja
- Uspostavljanje prekida u niskom režimu
- Programabilni 'Watchdog' tajmer
- Dvostruki pokazivač podataka
- Isključivanje pomoću znaka (POWER OFF)
- Fleksibilni ISP programator (byte i page mode) - Page mode: 64 byte/page za CODE memoriju, 32 byte/page za DATA memoriju
- Poboľjšani kontroler prekida u 4 nivoa
- Programabilna x2 clock opcija
- Unutrašnji POWER-ON reset
- 42-pin PDIP model kućišta za redukovanje EMS emisije
- Zelena (Pb/Holide-free) opcija kućišta

AT89S8253 je procesor visokih performansi i male potrošnje snage. Radi se o CMOS 8-bitnom mikrokontroleru sa 12 kB flash memorije i 2kB EEPROM-a. Napravljen je u Atmel korporaciji i kompatibilan je sa poznatim i popularnim procesorom 80C51. Reprogramiranje memorije je moguće pomoću SPI serijskog ulaza ili pomoću programatora memorija. AT89S8253 se koristi za operacije niskih frekvencija, i ima 2 softverska moda. IDLE mod zaustavlja CPU dok dozvoljava rad RAM-a, brojača/tajmera, serijskog porta i sistema prekida. Niski mod ušteda RAM sadržaj ali i zaustavlja oscilator isključujući sve ostale funkcije u kolu sve dok ne naiđe sledeći spoljašnji prekid ili hardverski reset. FLASH memoriji se može pristupiti preko SPI interfejsa.

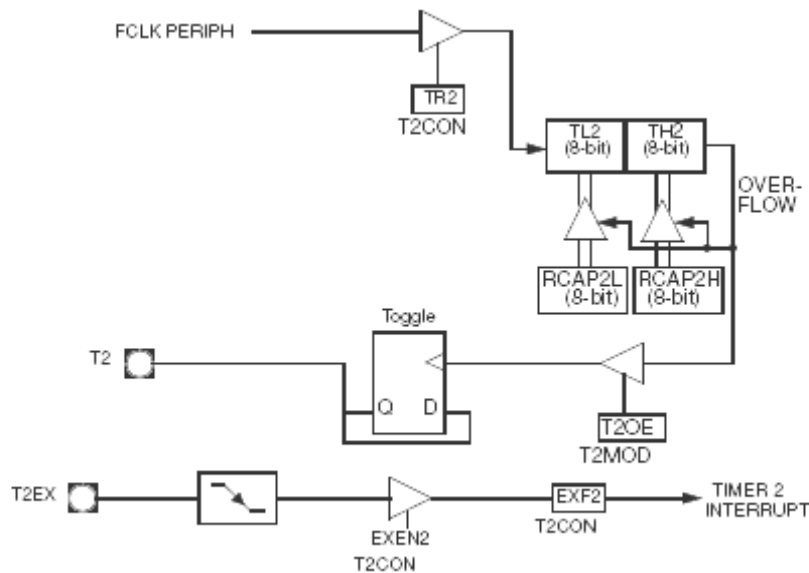
Blok šema mikrokontrolera AT89S8253 data je na slici.



Slika 9. Blok šema mikrokontrolera AT89S8253

Dalji opis mikrokontrolera i njegovih portova može se na i na www.atmel.com u AT89S8253 datasheet-u.

Zadatak ovog projekta je programiranje eksternog tajmera, za iji rad je potreban spoljašnji klock. Ovaj klock se dobija sa pina P1.0 mikrokontrolera. Koristi se Tajmer 2 mikrokontrolera u CLOCK-OUT režimu:



Slika 10. Clock-Out režim Tajmera 2

Princip rada tajmera je takav da se pri dostizanju prekora enja u brojanju, od zadate vrednosti brojanja, sadržaj registara RCAP2H i RCAP2L u itava u TH2 i TL2, pri emu nema generisanja prekida.

Izlazna frekvencija ra una se po formuli

$$F_{out} = \frac{F_{osc}}{4 \times [65536 - (RCAP2H/RCAP2L)]}$$

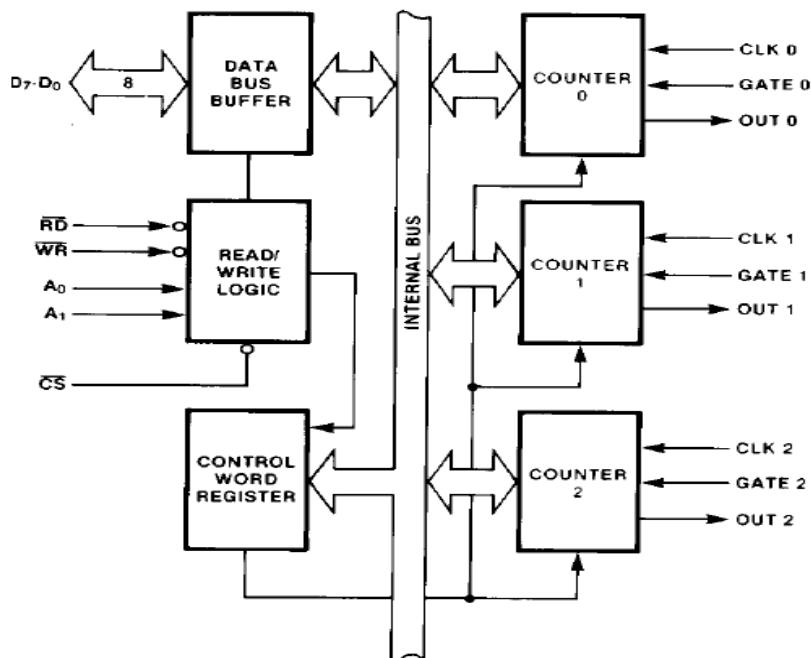
Periferija – Tajmer 82c54

Kao periferija koristi se CHMOS programabilni interval tajmer Intel 82c54 koji se karakteriše sa ulaznom frekvencijom do 10MHz, tri 16-bitna broja a/tajmera, malom potrošnjom, oko 10mA pri taktu od 8MHz. Ovaj tajmer je TTL kompatibilan, ima 6 režima rada, može da broji binarno ili BCD.

Tajmerom 82c54 mogu da se generišu potrebna kašnjenja, i neke od primena su:

- sat realnog vremena
- broja događaja
- generator pravougaonih impulsa
- kontroler motora
- binarni množa frekvencije

Blok šema tajmera je na sledejoj slici



Slika 11. Blok šema tajmera 82c54

Bafer podataka

Trostrani, 8-bitni, bi-direkcionni bafer i koristi se kao interfejs za povezivanje 82c54 sa mikrokontrolerom.

R/W Logika

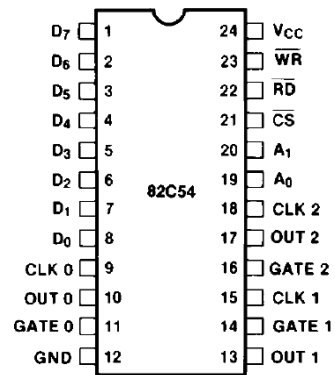
Prihvata upravljačke signale, i generiše signale za rad celokupnog kola. Rad bloka se kontroliše pomoću signala **CS**.

CS - Ulazni signal kojim se dozvoljava rad kola

RD - Ulazni signal koji treba da bude u stanju logičke nule, pri čitanju.

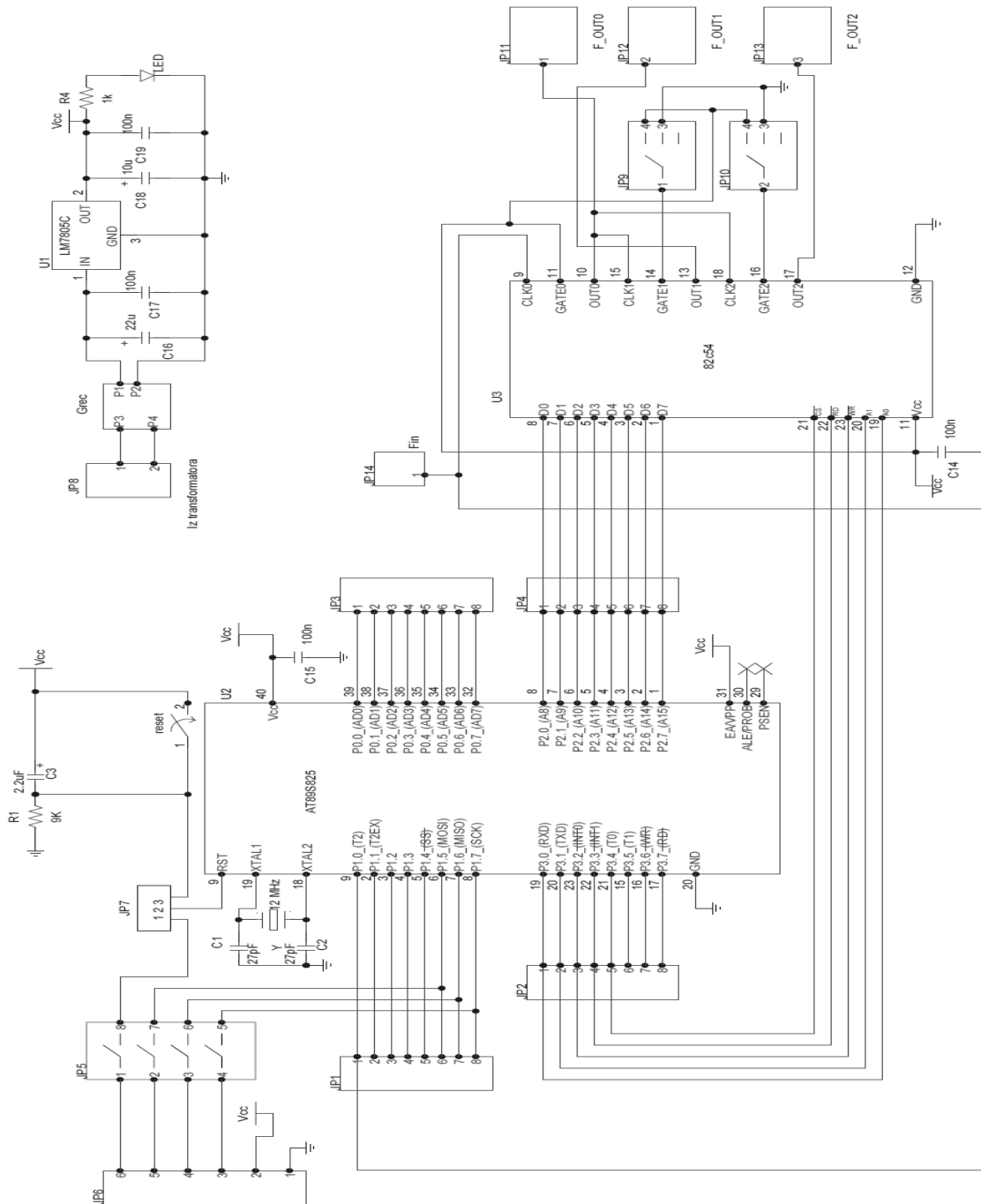
WR - Ulazni signal koji treba da bude u stanju logi ke nule, pri upisu.

A1, A0 - Ulazi kojim se vrši selekcija jednog od tri broja a za upis inicijalne vrednosti za brojanje, ili *Kontrolne reci*.



Slika 12. Raspored pinova tajmera 82c54

Elektri na šema ure aja sa svim komponentama data je na slici 13.



Slika 13. Elektri na šema sistema

Radi se o ISP varijanti za AT89S8253 – In System Programming. Programiranje se vrši povezivanjem ISP kabla sa ra unara na ISP konektor na plo ici.

Data Bus tajmera 82c54 povezan je sa Portom2 mikrokontrolera, dok je Read/Write logika povezana sa Portom P3, u redosledu:

$\overline{\text{CS}}$, $\overline{\text{RD}}$, $\overline{\text{WR}}$, A1, A0 (kao u tabeli 6),

(na P3.4 ide CS, itd). *Izlaz tajmera 0 povezan je na ulaz tajmera 1 i 2.* Trigerovanje gejta 1 i 2 se ostvaruje džamperom Gate 1 i 2. Gejt tajmera 0 je uvek na visokom nivou.

Izlazi tajmera 0 i 1 i 2 su F_OUT 0, F_OUT 1 i F_OUT2.

3. Software

Kao software koristi se *Keil μ Vision3 V3.5* za pisanje kôda i simulaciju. Kôd je napisan u C-u.

Opis kôda

Najpre se ukine prekid, izvrši programiranje Tajmera 2 mikrokontrolera AT89S8253, tako što se odredi vrednost brojanja (za 500KHz na P1.0, je potrebno da vrednost brojanja bude FFFA ili 65530, što se dobija po formuli za Fout). U registar T2CON se upiše 253 (C/T2 se postavi na 0), dok se u T2MOD registru T2OE postavi na 1. Sa TR2=1 startuje tajmer sa generisanjem frekvencije. Programira se, zatim, tajmer 0 eksternog tajmera 82c54, da generiše 500 puta manju frekvencu, i njegov izlaz se vodi na ulaz tajmera 1 ili 2, koji dalje deli, sa odre enom vrednoš u, i radi u potrebnom režimu. (U ovom slu aju (MODE3 – SQUARE WAVE MODE) tajmer 0 deli sa 500d (1F4 H) – LSB i MSB, dok tajmer 1(ili 2) deli sa 1000d (3E8 H) – LSB i MSB). Oba tajmera eksternog tajmera broje binarno.

1000d = 3E8h; LSB=E8, MSB=3; 500d = 1F4h; LSB=F4, MSB=1

Primer kôda:

```
#include <AT898252.H>
```

```
sbit Izlaz = P1^0;  
sbit Pin = P1^2;  
sbit GATE1_Pin = P1^4;  
sbit GATE2_Pin = P1^5;
```

```
unsigned int k;  
unsigned int l;
```

```
void main(void)  
{
```

```

P1=1;
GATE1_Pin = 1;
GATE2_Pin = 1;

// Inicijalizacija sistema - programiranje tajmera 2 kod AT89S8253 u Clock-Out
rezimu
EA = 0; // GIE = 0
RCAP2H = 0xFF; // 0xFFFFA za 12Mhz XTAL; na PIN-u 1.0 500KHz
RCAP2L = 0xFA;

T2CON &= 253; // C/T2 treba da bude Clear

T2MOD |= 2; // T2OE = 1;

TR2 = 1; // Start Tajmera

// Programiranje eksternog TAJMERA - 82c54, T1 u modu 3

for(k=0; k<=7000; k++); //kasnjenje od 20ms pre pocetka programiranja tajmera 0

// Mode 3 - generisanje pravougaonih impulsa

// programiranje tajmera 0 - deljenje 500KHz sa 500 ( sa 1F4 )

P3 = 0x0F; // nedozvoljeno stanje za tajmer 82c54 u skladu sa tabelom 6
P2 = 0x36; // Kontrolna rec 110110b – tabela 1,2,3,4,5
P3 = 0x0B; // upis Kontrolne reci; iz tabele 6 je 01011b=Bh
P3 = 0x0F;
P2 = 0xF4; // LSB vrednosti brojanja
P3 = 0x08; // upis LSB-a u tajmer 0; iz tabele 6 je 01000b=8h
P3 = 0x0F;
P2 = 0x01; //MSB vrednosti brojanja
P3 = 0x08; //upis MSB-a u tajmer 0
P3 = 0x0F; // nedozvoljeno stanje za tajmer 82c54

for(l=0; l<=7000; l++); //kasnjenje od 20ms posle pocetka rada tajmera 0

// programiranje tajmera 1 sa inicijalnom vrednoscu od 1000 (3E8)
// Mode 3 - generisanje pravougaonih impulsa
P3 = 0x0F;
P2 = 0x76; //kontrolna rec 1110110 – tabela 1,2,3,4,5
P3 = 0x0B; // iz tabele 6 je 01011b=Bh
P3 = 0x0F;
P2 = 0xE8; //LSB vrednosti brojanja

```

```

P3 = 0x09;    //upis LSB-a u tajmer 1
P3 = 0x0F;
P2 = 0x03;    //msb vrednosti brojanja
P3 = 0x09;    //upis MSB-a u tajmer 0; iz tabele 6 je 01001b=9h
P3 = 0x0F;    // nedozvoljeno stanje za tajmer 82c54

    while(1)
    {
        }
}

```

Iz datog kôda se vidi da je za programiranje Tajmera 2 eksternog tajmera 82c54 potrebno samo promeniti vrednost Kontrolne reči, u skladu sa tabelama 1,2,3,4,5, i vrednost brojanja, dok Tabela 6 omogućava kontrolu upisa, i vrednosti treba menjati u skladu sa istom, u zavisnosti od toga da li se programira tajmer ili upisuje Kontrolna reč.

Napomena: Tajmer 0 se programira samo u Modu3 sa vrednošću brojanja 1000d i uvek je aktivan, Gate (dozvola rada) mu je na logičkoj jedinici. Tajmeri 1 i 2 se programiraju za sve režime rada upisom odgovarajućeg kôda u mikrokontroler i trigeruju džamperom, u Modu 1 i 5. Kodovi su dati u poglavlju 4.

Programiranje tajmera 82c54

Programiranje se vrši upisom *kontrolne reči* i posle toga upisom inicijalne vrednosti za brojanje, koja se ne upisuje u Control Word registar, već u sam tajmer.

Format upisa:

1. Kontrolna reč
2. LSB vrednosti brojanja
3. MSB vrednosti upisa

Ili:

1. Kontrolna reč
2. LSB vrednosti brojanja

Ili:

1. Kontrolna reč
2. MSB vrednosti brojanja

Ovaj redosled treba poštovati u skladu sa Kontrolnom rečom, tj. ako je broj , Kontrolnom rečom, predviđen za upis LSB, pa MSB tako i treba upisati pri upisu vrednosti brojanja. Ako se navede *samo LSB* ili *MSB* u Kontrolnoj reči, onda treba upisati samo tu jednu vrednost – najviše 8-bitnu.

LSB – least significant byte – niži bajt

MSB – most significant byte – viši bajt

Kontrolna re se upisuje u Control Word registar i time se određuje broja , mod rada, i na in brojanja, pri emu je:

$A1, A0 = 11$ $\overline{CS} = 0$ $\overline{RD} = 1$ $\overline{WR} = 0$; videti tabelu 6

Tabela 1. Format Kontrolne re i

D7	D6	D5	D4	D3	D2	D1	D0
SC1	SC0	RW1	RW0	M2	M1	M0	BCD

Tabela 2. SC - Selektovanje tajmera

SC1	SC0	
0	0	Tajmer 0
0	1	Tajmer 1
1	0	Tajmer 2

Tabela 3. M - Režim rada

M2	M1	M0	
0	0	0	Mode 0
0	0	1	Mode 1
0	1	0	Mode 2
0	1	1	Mode 3
1	0	0	Mode 4
1	0	1	Mode 5

Tabela 4. RW – Citanje/Upis

RW1	RW0	
0	1	R/W samo LSB
1	0	R/W samo MSB
1	1	R/W LSB najpre, zatim MSB

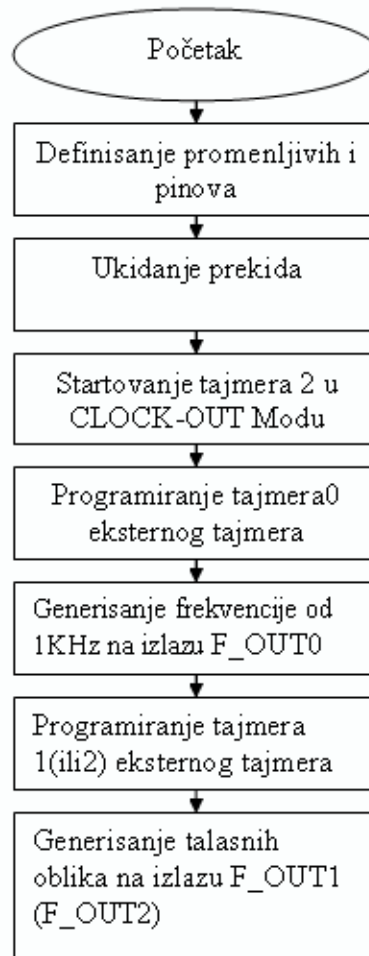
Tabela 5. BCD – Na in brojanja

0	Binarno brojanje 16-bit
1	BCD brojanje

Tabela 6. Write operacije

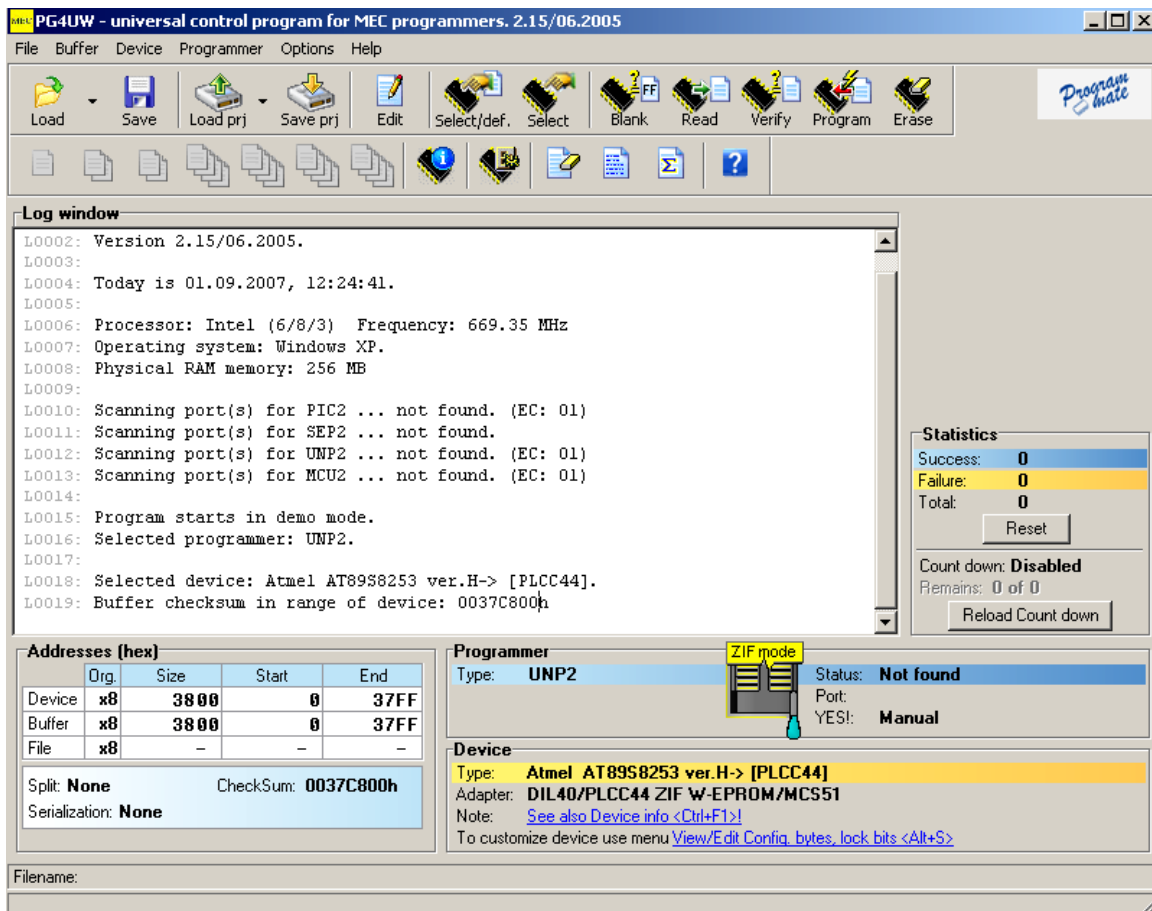
\overline{CS}	\overline{RD}	\overline{WR}	A1	A0	
0	1	0	0	0	u itavanje tajmera 0
0	1	0	0	1	u itavanje tajmera 1
0	1	0	1	0	u itavanje tajmera 2
0	1	0	1	1	upisivanje Kontrolne reci
0	0	1	1	1	neaktivno stanje
1	1	1	1	1	neaktivno stanje
0	1	1	1	1	neaktivno stanje

Algoritam rada



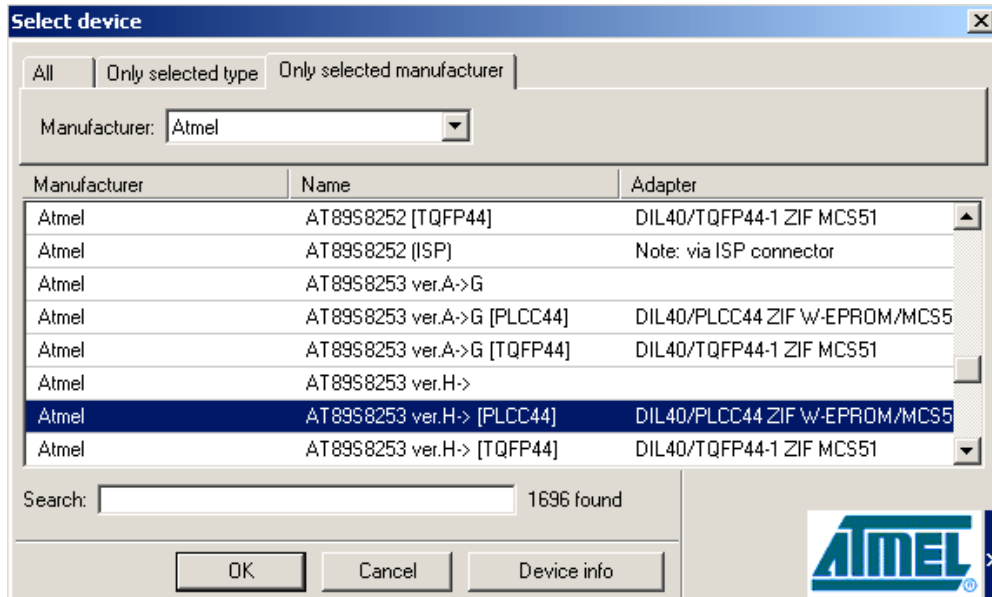
Programiranje mikrokontrolera

Upis koda u mikrokontroler može da se vrši preko ISP kabla i ISP konektora na plo ici ili koriš enjem programatora. U ovom slu aju koriš en je programator *MEC ProgramMate UNP2 Universal Programmer* i software *MEC PG4UW*. Postupak programiranja je objašnjen na slede im slikama.



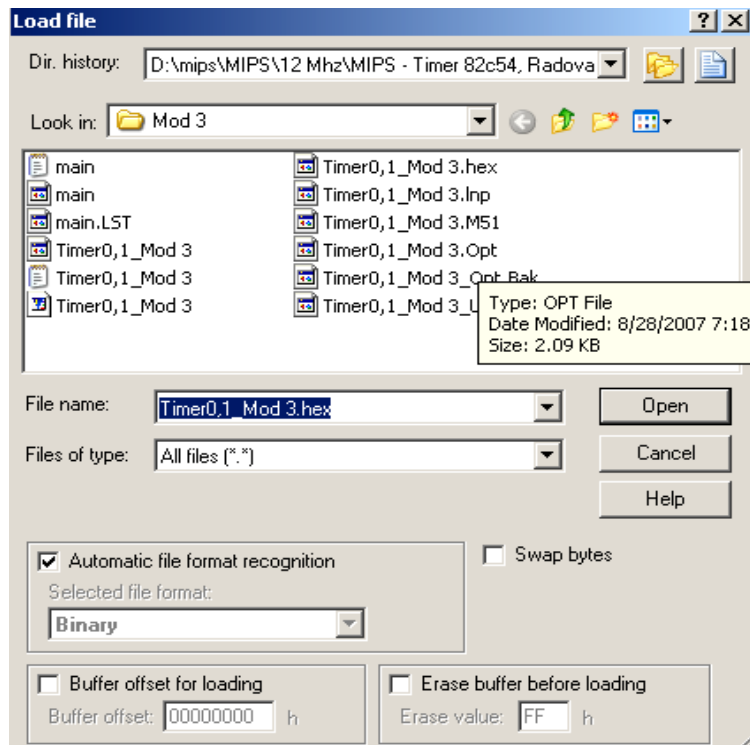
Slika 14. Osnovni prozor programa MEC PG4UW

Posle startovanja programa izabere se opcija **Device/Select Device**, gde se izabere Atmel AT89S8253.



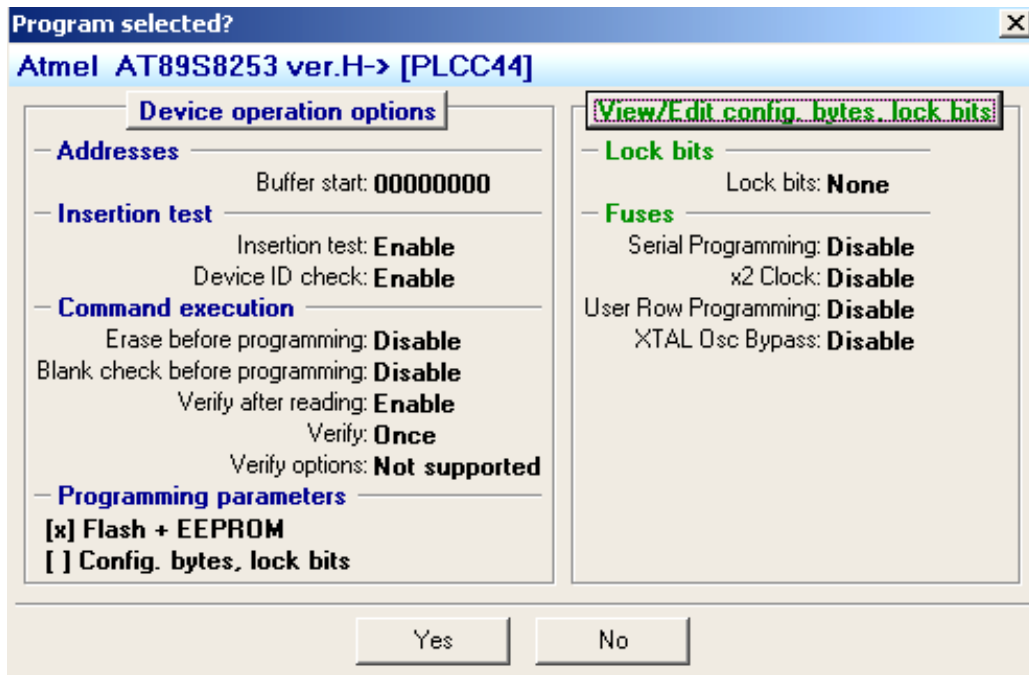
Slika 15. Selekcija AT89S8253

Sledeći korak je učitavanje fajla u bafer, što se postiže sa **Load** (Load File To Buffer) i navede putanja do fajla koji je potreban.



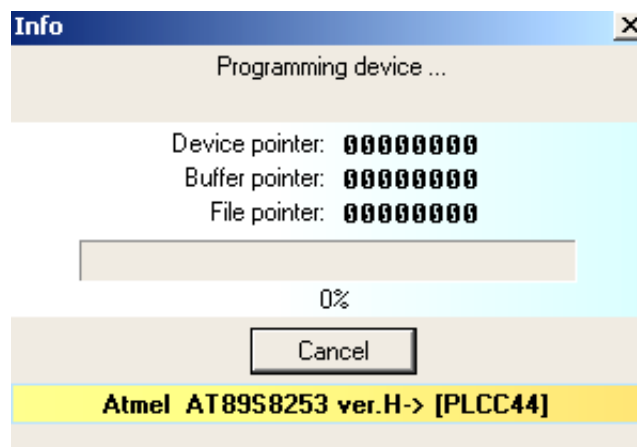
Slika 16. Odabir fajla

Sledeći korak je programiranje i postiže se sa **Device/Program**, gde se bira **Flash + EEPROM**



Slika 17. Poslednji korak programiranja

Klikom na **Yes** otvara se novi prozor koji prati postupak upisa kôda u mikrokontroler



Slika 18. Upis kôda u mikrokontroler

Nakon ovoga javlja se poruka o uspešnom programiranju, nakon čega se mikrokontroler vraća na ploču i spreman je za rad, a za ostale režime rada se isti postupak ponavlja.

4. Primeri za Tajmer 82c54

Ovde su dati primeri rada Tajmera u svih 6 režima (modova), zajedno sa kôdom i talasnim oblicima izlaznog signala. Dati su kôdovi za Tajmer 1; za Tajmer 2 je sve isto samo što treba promeniti vrednosti na Portu P3 i Portu P2, u kôdu, u skladu sa tabelama 1,2,3,4,5,6.

MOD0: Generisanje Prekida po završetku brojanja

Nakon upisa Kontrolne reči i Izlaz (OUT) je, inicijalno, u stanju logičke nule, i tu ostaje dok broj ne dostigne nulu. Izlaz onda ide u stanje logičke jedinice i tu ostaje do novog upisa. Gate=1 dozvoljava rad, Gate=0 zabranjuje, ali Gate nema uticaja na Izlaz.

Nakon upisa Kontrolne reči i vrednosti brojanja, vrednost brojanja se učitava na sledeći Klok impuls, koji ne smanjuje tu vrednost; pa za vrednost brojanja N, Izlaz ne izađe u stanje logičke jedinice pre N+1 Klok impulsa posle upisa inicijalne vrednosti brojanja.

Kôd:

```
#include <AT898252.H>
```

```
sbit Izlaz = P1^0;  
sbit Pin = P1^2;  
sbit GATE1_Pin = P1^4;  
sbit GATE2_Pin = P1^5;
```

```
unsigned int k;  
unsigned int l;
```

```
void main(void)  
{
```

```
    P1=1;  
    GATE1_Pin = 1;  
    GATE2_Pin = 1;
```

```

// Inicijalizacija sistema
EA = 0; // GIE = 0
RCAP2H = 0xFF; // 0xFFFFA za 12Mhz XTAL; na PIN-u 1.0 500Khz
RCAP2L = 0xFA;

T2CON &= 253; // C/T2 treba da bude Clear

T2MOD |= 2; // T2OE = 1;

//T2CON |= 4; // TR2 = Start Timer
TR2 = 1;

// Programiranje eksternog TAJMERA - 82c54, T1 u modu 0

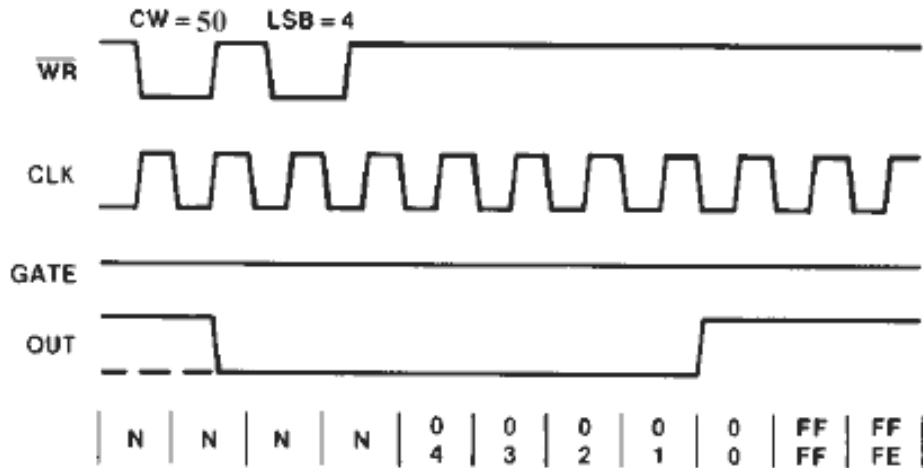
for(k=0; k<=7000; k++);
// Timer 0 - deljenje 500KHz sa 500 ( 1F4 )
P3 = 0x0F;
P2 = 0x36; // kontrolna rec
P3 = 0x0B;
P3 = 0x0F;
P2 = 0xF4; // lsb vrednosti brojanja
P3 = 0x08;
P3 = 0x0F;
P2 = 0x01; //msb vrednosti brojanja
P3 = 0x08;
P3 = 0x0F;

for(l=0; l<=7000; l++);
// Timer 1
P3 = 0x0F;
P2 = 0x50; //kontrolna rec
P3 = 0x0B;
P3 = 0x0F;
P2 = 0x04; //lsb vrednosti brojanja
P3 = 0x09;
P3 = 0x0F;

while(1)
{
}
}

```

Talasni oblici



CW = Kontrolna reč

MOD1: Monostabilni multivibrator koji se hardverski ponovo okida

Izlaz je, inicijalno, biti u stanju logi ke jedinice. Izlaz je pre i u stanje logi ke nule pod dejstvom Klok impulsa i uz trigerovanje gejta i po e e brojanje, pri emu izlaz ostaje u stanju logi ke nule dok tajmer ne dostigne nulu. Izlaz je onda oti i u stanje logi ke jedinice i tamo ostati do slede eg Klok impulsa i trigera. Vrednost brojanja od N rezultuje u trajanju od N Klok impulsa. Gate ne uti e na izlaz. Proces se može ponoviti, bez ponovnog upisa vrednosti brojanja u broja , novim trigerovanjem.

Napomena: Trigerovanje se ostvaruje prebacivanjem džampera Gate 1, na pločici sa tajmerom 82c54, u srednji položaj(GND) i vraćanjem na 5V.

Kôd:

```
#include <AT898252.H>
```

```
sbit Izlaz = P1^0;  
sbit Pin = P1^2;  
sbit GATE1_Pin = P1^4;  
sbit GATE2_Pin = P1^5;
```

```
unsigned int k;  
unsigned int l;
```

```

void main(void)
{

    P1=1;

    // Inicijalizacija sistema
    EA = 0;      // GIE = 0
    RCAP2H = 0xFF;    // 0xFFFFA za 12Mhz XTAL; na PIN-u 1.0 500Khz
    RCAP2L = 0xFA;

    T2CON &= 253;    // C/T2 treba da bude Clear

    T2MOD |= 2;      // T2OE = 1;

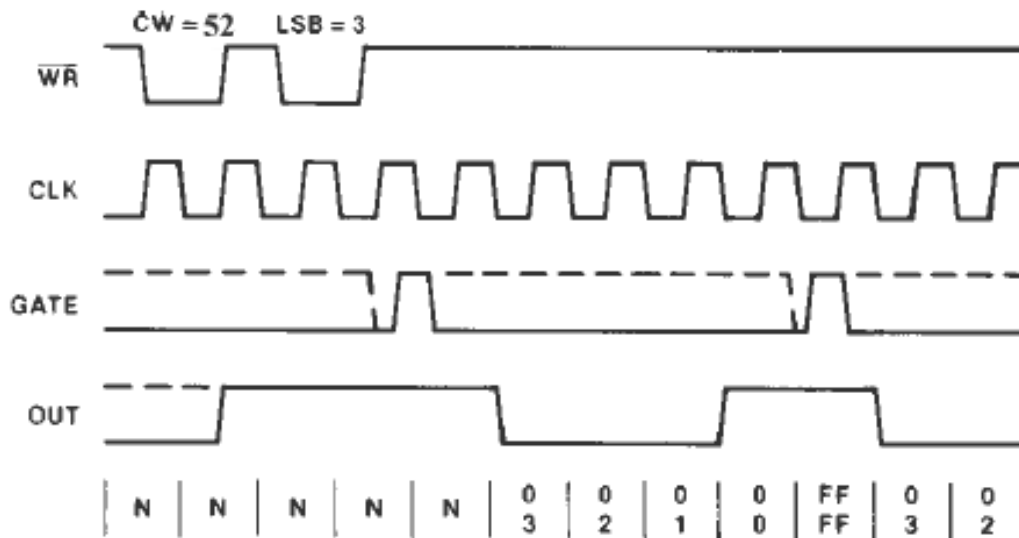
    //T2CON |= 4;    // TR2 = Start Timer
    TR2 = 1;

    // Programiranje eksternog TAJMERA - 82c54, T1 u modu 1

    for(k=0; k<=7000; k++);
    //Timer 0 - deljenje sa 500
    P3 = 0x0F;
    P2 = 0x36; //kontrolna rec
    P3 = 0x0B;
    P3 = 0x0F;
    P2 = 0xF4; //lsb
    P3 = 0x08;
    P3 = 0x0F;
    P2 = 0x01; //msb
    P3 = 0x08;
    P3 = 0x0F;
    for(l=0; l<=7000; l++);
    //Timer 1
    P3 = 0x0F;
    P2 = 0x52; //kontrolna rec
    P3 = 0x0B;
    P3 = 0x0F;
    P2 = 0x03; //lsb
    P3 = 0x09;
    P3 = 0x0F;
    while(1)
    {
        }
}

```


Talasnici



MOD2: Delitelj sa N

U ovom režimu broja radi kao delitelj sa N. Izlaz je, inicijalno, u stanju logi ke jedinice. Kada se vrednost brojanja smanji na 1, izlaz prelazi u stanje logi ke nule i tu ostaje za vreme jednog impulsa Kloka. Izlaz ponovo prelazi u stanje logi ke jedinice, broja ponovo u ita vrednost brojanja i proces se ponavlja periodi no. Za vrednost brojanja N, sekvenca se ponavlja na svakih N Klok impulsa.

Gate=1 dozvoljava brojanje, Gate=0 zabranjuje.

Ako se Gate prebaci na logi ku nulu u toku izlaznog impulsa, Izlaz odmah prelazi u logi ku nulu. Trigger ponovo u ita broja sa inicijalnom vrednoš u na slede i Klok impuls. Izlaz prelazi u logi ku nulu nakon N Klok impulsa posle triggerovanja. *

Nakon upisa Kontrolne re i i inicijalne vrednosti brojanja, broja e biti u itan na slede i Klok impuls. Izlaz ide u stanje logi ke nule i tamo je N Klok impulsa nakon upisa vrednosti brojanja.

* Napomena: Triggerovanje se ostvaruje prebacivanjem džampera u srednji položaj(GND) i vraćanjem na 5V.

Kôd:

```
#include <AT898252.H>

sbit Izlaz = P1^0;
sbit Pin = P1^2;
sbit GATE1_Pin = P1^4;
sbit GATE2_Pin = P1^5;

unsigned int k;
unsigned int l;

void main(void)
{
    P1=1;
    GATE1_Pin = 1;
    GATE2_Pin = 1;

    // Inicijalizacija sistema
    EA = 0; // GIE = 0
    RCAP2H = 0xFF; // 0xFFFFA za 12Mhz XTAL; na PIN-u 1.0 500Khz
    RCAP2L = 0xFA;
    T2CON &= 253; // C/T2 treba da bude Clear
    T2MOD |= 2; // T2OE = 1;
    //T2CON |= 4; // TR2 = Start Timer
    TR2 = 1;

    // Programiranje eksternog TAJMERA - 82c54, T1 u modu 2
    for(k=0; k<=7000; k++);
    //Timer 0 - deljenje sa 500
    P3 = 0x0F;
    P2 = 0x36; //kontrolna rec
    P3 = 0x0B;
    P3 = 0x0F;
    P2 = 0xF4; //lsb
    P3 = 0x08;
    P3 = 0x0F;
    P2 = 0x01; //msb
    P3 = 0x08;
    P3 = 0x0F;
    for(l=0; l<=7000; l++);
    //Timer 1
    P3 = 0x0F;
    P2 = 0x54; //kontrolna rec
```


MOD3: Generator pravougaonih impulsa

Sličan Modu 2, ali je izlaz povorka pravougaonih impulsa iste širine, i to samo u slučaju parne inicijalne vrednosti brojanja N. Ovo je periodičan režim rada i imamo pravougaone impulse sa periodom od N Klok ciklusa.

Gate=1 dozvoljava brojanje, Gate=0 zabranjuje.

Ako se Gate prebaci na logi ku nulu u toku izlaznog impulsa, izlaz odmah prelazi u logi ku nulu. Trigger ponovo uita broja sa inicijalnom vrednoš u na slede i Klok impuls. **

Nakon upisa Kontrolne re i i inicijalne vrednosti brojanja, broja e se u itati na slede i Klok impuls.

Imamo dva slu aja: parna i neparna vrednost brojanja

Parno N:

Izlaz je inicijalno u stanju logi ke jedinice. Vrednost brojanja se u ita na jedan impuls Kloka i smanji za dva na slede i Klok impuls. Kada se odbroji Izlaz menja vrednost i broja se ponovo u ita sa inicijalnom vrednoš u brojanja. Proces se ponavlja u beskona nosti.

Neparno N:

Izlaz je, inicijalno, u stanju logi ke jedinice. Inicijalna vrednost minus jedan (paran broj) se u ita na Klok impuls i smanji za dva na slede i Klok impuls. Posle vremena od jednog Klok impulsa nakon odbrojavanja tajmera, izlaz prelazi u stanje logi ke nule i broja se ponovo u ita sa inicijalnom vrednoš u minus jedan. Zatim se novim Klok impulsom vrednost brojanja umanju za dva. Kada se izvrši brojanje, Izlaz ponovo prelazi u logi ku jedinicu i broja je ponovo u itan sa vrednoš u brojanja minus jedan. Proces se ponavlja u beskona nosti. Izlaz e biti u stanju logi ke jedinice $(N+1)/2$, i u stanju logi ke nule $(N-1)/2$, od zadate vrednosti brojanja.

*** Napomena: Ovaj slucaj se ostvaruje prebacivanjem džampera u srednji položaj(GND) i vraćanjem na 5V.*

Kôd:

```
#include <AT898252.H>
```

```
sbit Izlaz = P1^0;  
sbit Pin = P1^2;  
sbit GATE1_Pin = P1^4;  
sbit GATE2_Pin = P1^5;
```

```
unsigned int k;  
unsigned int l;
```

```

void main(void)
{

    P1=1;
    GATE1_Pin = 1;
    GATE2_Pin = 1;

    // Inicijalizacija sistema
    EA = 0; // GIE = 0
    RCAP2H = 0xFF; // 0xFFFFA za 12Mhz XTAL; na PIN-u 1.0 500Khz
    RCAP2L = 0xFA;

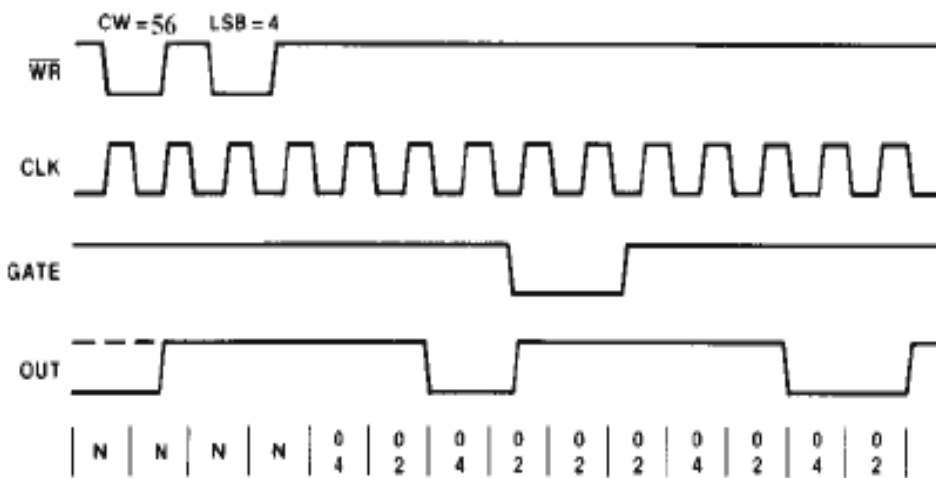
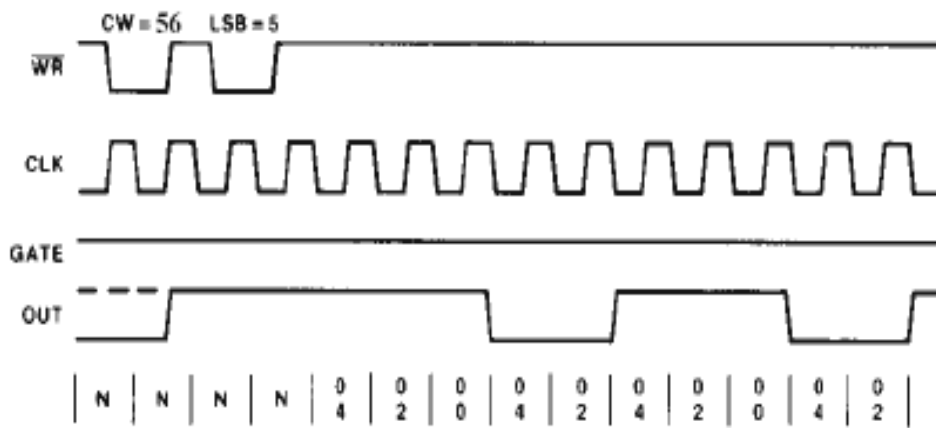
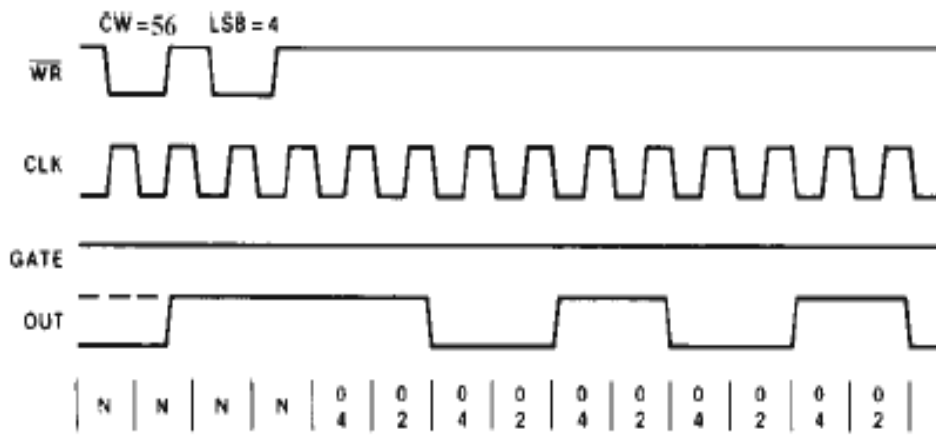
    T2CON &= 253; // C/T2 treba da bude Clear
    T2MOD |= 2; // T2OE = 1;
    //T2CON |= 4; // TR2 = Start Timer
    TR2 = 1;

    // Programiranje eksternog TAJMERA - 82c54, T0 i T1 u modu 3

    for(k=0; k<=7000; k++);
    //Timer 0 - deljenje sa 500
    P3 = 0x0F;
    P2 = 0x36; //kontrolna rec
    P3 = 0x0B;
    P3 = 0x0F;
    P2 = 0xF4; //lsb
    P3 = 0x08;
    P3 = 0x0F;
    P2 = 0x01; //msb
    P3 = 0x08;
    P3 = 0x0F;
    for(l=0; l<=7000; l++);
    //Timer 1 - deljenje frekvencije od 1KHz sa 4
    P3 = 0x0F;
    P2 = 0x56; //kontrolna rec
    P3 = 0x0B;
    P3 = 0x0F;
    P2 = 0x04; //lsb
    P3 = 0x09;
    P3 = 0x0F;
    while(1)
    {
        }
}

```

Talasní oblici:



MOD4: Softversko-trigerovani režim rada

Izlaz je, inicijalno, biti u stanju logičke jedinice. Kada broja odbroji, Izlaz ide u stanje logičke nule za vreme od jednog Klok impulsa i zatim ide ponovo u stanje logičke jedinice. Brojanje počinje upisom vrednosti brojanja.

Gate=1 dozvoljava brojanje, Gate=0 zabranjuje i ne utiče na Izlaz.

Nakon upisa Kontrolne vrednosti za brojanje, broja se učitava na sledeći Klok impuls, koji smanjuje vrednost brojanja, i za vrednost brojanja N Izlaz ne ide u stanje logičke jedinice sve dok ne prođe N+1 impulsa Kloka, od trenutka upisa vrednosti brojanja.

Kôd:

```
#include <AT898252.H>
```

```
sbit Izlaz    = P1^0;  
sbit Pin      = P1^2;  
sbit GATE1_Pin = P1^4;  
sbit GATE2_Pin = P1^5;
```

```
unsigned int k;  
unsigned int l;
```

```
void main(void)  
{
```

```
    P1=1;  
    GATE1_Pin = 1;  
    GATE2_Pin = 1;
```

```
    // Inicijalizacija sistema
```

```
    EA = 0;        // GIE = 0
```

```
    RCAP2H = 0xFF;    // 0xFFFFA za 12Mhz XTAL; na PIN-u 1.0 500KHz
```

```
    RCAP2L = 0xFA;
```

```
    T2CON &= 253;    // C/T2 treba da bude Clear
```

```
    T2MOD |= 2;      // T2OE = 1;
```

```
    //T2CON |= 4;    // TR2 = Start Timer
```

```
    TR2 = 1;
```

```

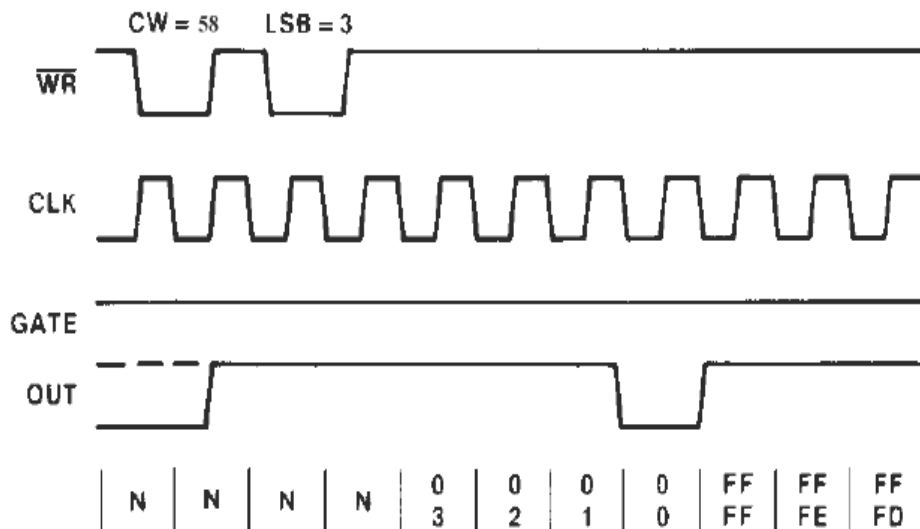
// Programiranje eksternog TAJMERA - 82c54, T1 u modu 4

for(k=0; k<=7000; k++);
//Timer 0 - deljenje sa 500
P3 = 0x0F;
P2 = 0x36; //kontrolna rec
P3 = 0x0B;
P3 = 0x0F;
P2 = 0xF4; //lsb
P3 = 0x08;
P3 = 0x0F;
P2 = 0x01; //msb
P3 = 0x08;
P3 = 0x0F;

for(l=0; l<=7000; l++);
//Timer 1
P3 = 0x0F;
P2 = 0x58; //kontrolna rec
P3 = 0x0B;
P3 = 0x0F;
P2 = 0x03; //lsb
P3 = 0x09;
P3 = 0x0F;
while(1)
{
    }
}

```


Talasnici



MOD5: Hardversko-trigerovani režim rada

Izlaz je bitni, inicijalno, u stanju logičke jedinice. Brojanje počinje rasti od ivice gejt. Kada se izvrši brojanje, Izlaz je prei u stanje logičke nule i ostati u tom stanju jedan klok impuls, nakon čega se vraća u stanje logičke jedinice.

Nakon upisa Kontrolne reži i inicijalne vrednosti brojanja, brojanje bitni u itan nakon Klok impulsa, kome prethodi trigerovanje gejt. Ovaj Klok impuls ne smanjuje vrednost brojanja, pa za inicijalnu vrednost brojanja N, Izlaz ne ide u stanje logičke nule sve dok ne prođe N+1 Klok impulsa posle trigerovanja. Sekvenca brojanja je retrigerabilna. Gejt ne utiče na izlaz.

Kôd:

```
#include <AT898252.H>
```

```
sbit Izlaz = P1^0;  
sbit Pin = P1^2;  
sbit GATE1_Pin = P1^4;  
sbit GATE2_Pin = P1^5;
```

```
unsigned int k;  
unsigned int l;
```

```

void main(void)
{
    P1=1;
    // Inicijalizacija sistema
    EA = 0;      // GIE = 0
    RCAP2H = 0xFF;    // 0xFFFFA za 12Mhz XTAL; na PIN-u 1.0 500Khz
    RCAP2L = 0xFA;
    T2CON &= 253;    // C/T2 treba da bude Clear
    T2MOD |= 2;      // T2OE = 1;
    //T2CON |= 4;    // TR2 = Start Timer
    TR2 = 1;
    // Programiranje eksternog TAJMERA - 82c54, T1 u modu 5

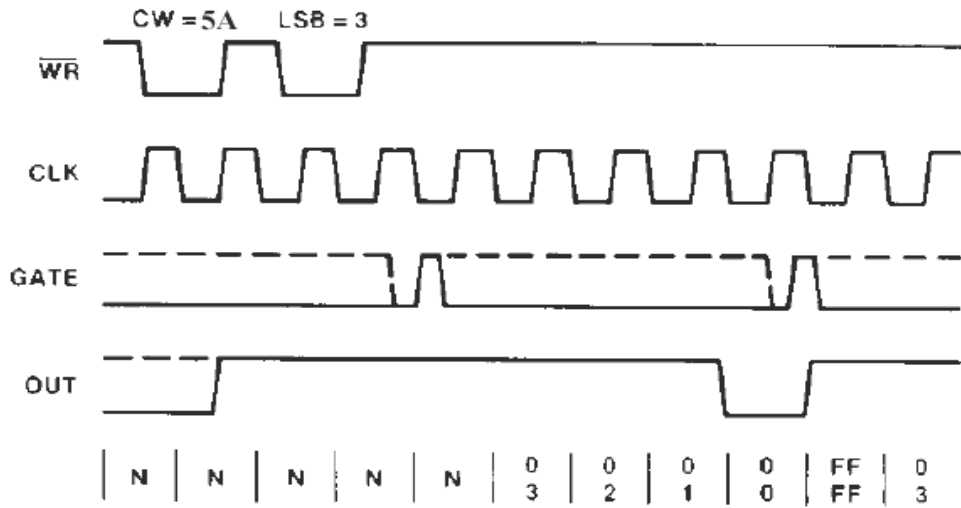
    for(k=0; k<=7000; k++);
    //Timer 0 - deljenje sa 500
    P3 = 0x0F;
    P2 = 0x36;    //kontrolna rec
    P3 = 0x0B;
    P3 = 0x0F;
    P2 = 0xF4; //lsb
    P3 = 0x08;
    P3 = 0x0F;
    P2 = 0x01; //msb
    P3 = 0x08;
    P3 = 0x0F;

    for(l=0; l<=7000; l++);
    //Timer 1
    P3 = 0x0F;
    P2 = 0x5A;    //kontrolna rec
    P3 = 0x0B;
    P3 = 0x0F;
    P2 = 0x03;    //lsb
    P3 = 0x09;
    P3 = 0x0F;

    while(1)
    {
        }
}

```

Talasni oblici



5. Zadatak za studente

Treba posmatrati ulaznu i izlaznu frekvenciju i uoiti njihov odnos.

Napomena: Tajmer 0 se uvek programira u Modu 3, kao generator pravougaonih impulsa frekvencije 1KHz. Ta frekvencija se koristi za tajmer 1 i (ili) 2.

Zadatak je menjati samo vrednost brojanja tajmera 1.

Timer 1 - MOD0:

Pokrenuti sa CD-a iz direktorijuma, Kod/Timer-82c54/Mod 0/Timer 1 fajl *Timer1_Mod 0.Uv2*, (*µVision Project File*), pri čemu se otvara prozor za slike 19.

Upisati, umesto postojeće, novu vrednost za brojanje, koja je zadata u vežbi.

Potrebno je, u redu 53, umesto vrednosti 4 na portu P2 ($P2 = 0x04$; //LSB vrednosti brojanja) upisati novu vrednost, i sa *Project/Rebuild all target files* izvršiti kompajliranje, nakon čega se kreira HEX fajl *Timer1_Mod 0.hex*, kojim će se mikrokontroler isprogramirati.

Posmatrati pomoću osciloskopa F_OUT0 i F_OUT1, na pločici sa tajmerom 82c54, i skicirati talasne oblike.

Slika 19. Osnovni prozor Keil-a gde će se vršiti promene

Zadaje se:

Grupa 1:

LSB podatak _____
Binarni oblik

_____ *Dekadni oblik*

Grupa 2:

LSB podatak _____
Binarni oblik

_____ *Dekadni oblik*

Grupa 3:

LSB podatak _____
Binarni oblik

_____ *Dekadni oblik*

Grupa 4:

LSB podatak _____
Binarni oblik

_____ *Dekadni oblik*

Grupa 5:

LSB podatak _____
Binarni oblik

_____ *Dekadni oblik*

Grupa 6:

LSB podatak _____
Binarni oblik

_____ *Dekadni oblik*

Grupa 7:

LSB podatak _____
Binarni oblik

_____ *Dekadni oblik*

Grupa 8:

LSB podatak _____
Binarni oblik

_____ *Dekadni oblik*

Grupa 9:

LSB podatak _____
Binarni oblik

_____ *Dekadni oblik*

Grupa 10:

LSB podatak _____
Binarni oblik

_____ *Dekadni oblik*

Grupa 11:

LSB podatak _____
Binarni oblik

_____ *Dekadni oblik*

Grupa 12:

LSB podatak _____
Binarni oblik

_____ *Dekadni oblik*

Grupa 13:

LSB podatak _____
Binarni oblik

_____ *Dekadni oblik*

Grupa 14:

LSB podatak _____
Binarni oblik

_____ *Dekadni oblik*

Grupa 15:

LSB podatak _____	_____
<i>Binarni oblik</i>	<i>Dekadni oblik</i>

Grupa 16:

LSB podatak _____	_____
<i>Binarni oblik</i>	<i>Dekadni oblik</i>

Timer 1 – MOD1

Pokrenuti sa CD-a iz direktorijuma, Kod/Timer-82c54/Mod 1/Timer 1 fajl *Timer1_Mod 1.Uv2*, (*µVision Project File*), pri emu se otvara prozor za slike 19.

Postupak je isti, upisati novu zadatu vrednost LSB-a vrednosti brojanja, u delu kôda, koji se odnosi na tajmer 1.

Nakon programiranja mikrokontrolera, trigerovati gejt 1 džamperom Gate 1, (prebacivanjem džampera na GND i vra anjem na +5V).

Posmatrati pomo u osciloskopa F_OUT0 i F_OUT1, na plo ici sa tajmerom 82c54 i skicirati talasne oblike.

Grupa 1:

LSB podatak _____	_____
<i>Binarni oblik</i>	<i>Dekadni oblik</i>

Grupa 2:

LSB podatak _____	_____
<i>Binarni oblik</i>	<i>Dekadni oblik</i>

Grupa 3:

LSB podatak _____	_____
<i>Binarni oblik</i>	<i>Dekadni oblik</i>

Grupa 4:

LSB podatak _____	_____
<i>Binarni oblik</i>	<i>Dekadni oblik</i>

Grupa 5:

LSB podatak _____	_____
<i>Binarni oblik</i>	<i>Dekadni oblik</i>

Grupa 6:

LSB podatak _____	_____
<i>Binarni oblik</i>	<i>Dekadni oblik</i>

Grupa 7:

LSB podatak _____	_____
<i>Binarni oblik</i>	<i>Dekadni oblik</i>

Grupa 8:

LSB podatak _____

*Binarni oblik**Dekadni oblik***Grupa 9:**

LSB podatak _____

*Binarni oblik**Dekadni oblik***Grupa 10:**

LSB podatak _____

*Binarni oblik**Dekadni oblik***Grupa 11:**

LSB podatak _____

*Binarni oblik**Dekadni oblik***Grupa 12:**

LSB podatak _____

*Binarni oblik**Dekadni oblik***Grupa 13:**

LSB podatak _____

*Binarni oblik**Dekadni oblik***Grupa 14:**

LSB podatak _____

*Binarni oblik**Dekadni oblik***Grupa 15:**

LSB podatak _____

*Binarni oblik**Dekadni oblik***Grupa 16:**

LSB podatak _____

*Binarni oblik**Dekadni oblik***Timer 1 - MOD2**

Pokrenuti sa CD-a iz direktorijuma, Kod/Timer-82c54/Mod 2/Timer 1 fajl *Timer1_Mod 2.Uv2*, (*µVision Project File*), pri emu se otvara prozor za slike 19.

Postupak je isti, upisati novu zadatu vrednost LSB-a vrednosti brojanja, u delu kôda, koji se odnosi na tajmer 1.

Posmatrati pomo u osciloskopa F_OUT0 i F_OUT1, na plo ici sa tajmerom 82c54 i skicirati talasne oblike.

Grupa 1:

LSB podatak _____

*Binarni oblik**Dekadni oblik***Grupa 2:**

LSB podatak _____

*Binarni oblik**Dekadni oblik*

Grupa 3: LSB podatak	_____	_____
	<i>Binarni oblik</i>	<i>Dekadni oblik</i>
Grupa 4: LSB podatak	_____	_____
	<i>Binarni oblik</i>	<i>Dekadni oblik</i>
Grupa 5: LSB podatak	_____	_____
	<i>Binarni oblik</i>	<i>Dekadni oblik</i>
Grupa 6: LSB podatak	_____	_____
	<i>Binarni oblik</i>	<i>Dekadni oblik</i>
Grupa 7: LSB podatak	_____	_____
	<i>Binarni oblik</i>	<i>Dekadni oblik</i>
Grupa 8: LSB podatak	_____	_____
	<i>Binarni oblik</i>	<i>Dekadni oblik</i>
Grupa 9: LSB podatak	_____	_____
	<i>Binarni oblik</i>	<i>Dekadni oblik</i>
Grupa 10: LSB podatak	_____	_____
	<i>Binarni oblik</i>	<i>Dekadni oblik</i>
Grupa 11: LSB podatak	_____	_____
	<i>Binarni oblik</i>	<i>Dekadni oblik</i>
Grupa 12: LSB podatak	_____	_____
	<i>Binarni oblik</i>	<i>Dekadni oblik</i>
Grupa 13: LSB podatak	_____	_____
	<i>Binarni oblik</i>	<i>Dekadni oblik</i>
Grupa 14: LSB podatak	_____	_____
	<i>Binarni oblik</i>	<i>Dekadni oblik</i>
Grupa 15: LSB podatak	_____	_____
	<i>Binarni oblik</i>	<i>Dekadni oblik</i>
Grupa 16: LSB podatak	_____	_____
	<i>Binarni oblik</i>	<i>Dekadni oblik</i>

Grupa 12:

LSB podatak _____

*Binarni oblik**Dekadni oblik***Grupa 13:**

LSB podatak _____

*Binarni oblik**Dekadni oblik***Grupa 14:**

LSB podatak _____

*Binarni oblik**Dekadni oblik***Grupa 15:**

LSB podatak _____

*Binarni oblik**Dekadni oblik***Grupa 16:**

LSB podatak _____

*Binarni oblik**Dekadni oblik***Timer 4 – MOD4**

Pokrenuti sa CD-a iz direktorijuma, Kod/Timer-82c54/Mod 4/Timer 1 fajl *Timer1_Mod 4.Uv2*, (*µVision Project File*), pri emu se otvara prozor za slike 19.

Postupak je isti, upisati novu zadatu vrednost LSB-a vrednosti brojanja, u delu kôda, koji se odnosi na tajmer 1.

Posmatrati pomo u osciloskopa F_OUT0 i F_OUT1, na plo ici sa tajmerom 82c54 i skicirati talasne oblike.

Grupa 1:

LSB podatak _____

*Binarni oblik**Dekadni oblik***Grupa 2:**

LSB podatak _____

*Binarni oblik**Dekadni oblik***Grupa 3:**

LSB podatak _____

*Binarni oblik**Dekadni oblik*

Grupa 4: LSB podatak	_____	_____
	<i>Binarni oblik</i>	<i>Dekadni oblik</i>
Grupa 5: LSB podatak	_____	_____
	<i>Binarni oblik</i>	<i>Dekadni oblik</i>
Grupa 6: LSB podatak	_____	_____
	<i>Binarni oblik</i>	<i>Dekadni oblik</i>
Grupa 7: LSB podatak	_____	_____
	<i>Binarni oblik</i>	<i>Dekadni oblik</i>
Grupa 8: LSB podatak	_____	_____
	<i>Binarni oblik</i>	<i>Dekadni oblik</i>
Grupa 9: LSB podatak	_____	_____
	<i>Binarni oblik</i>	<i>Dekadni oblik</i>
Grupa 10: LSB podatak	_____	_____
	<i>Binarni oblik</i>	<i>Dekadni oblik</i>
Grupa 11: LSB podatak	_____	_____
	<i>Binarni oblik</i>	<i>Dekadni oblik</i>
Grupa 12: LSB podatak	_____	_____
	<i>Binarni oblik</i>	<i>Dekadni oblik</i>
Grupa 13: LSB podatak	_____	_____
	<i>Binarni oblik</i>	<i>Dekadni oblik</i>
Grupa 14: LSB podatak	_____	_____
	<i>Binarni oblik</i>	<i>Dekadni oblik</i>
Grupa 15: LSB podatak	_____	_____
	<i>Binarni oblik</i>	<i>Dekadni oblik</i>
Grupa 16: LSB podatak	_____	_____
	<i>Binarni oblik</i>	<i>Dekadni oblik</i>

Timer 1- MOD5

Pokrenuti sa CD-a iz direktorijuma, Kod/Timer-82c54/Mod 5/Timer 1 fajl *Timer1_Mod 5.Uv2*, (*µVision Project File*), pri emu se otvara prozor za slike 19.

Postupak je isti, upisati novu zadatu vrednost LSB-a vrednosti brojanja, u delu kôda, koji se odnosi na tajmer 1.

Posmatrati pomo u osciloskopa F_OUT0 i F_OUT1, na plo ici sa tajmerom 82c54 i skicirati talasne oblike.

Grupa 1:

LSB podatak _____
Binarni oblik *Dekadni oblik*

Grupa 2:

LSB podatak _____
Binarni oblik *Dekadni oblik*

Grupa 3:

LSB podatak _____
Binarni oblik *Dekadni oblik*

Grupa 4:

LSB podatak _____
Binarni oblik *Dekadni oblik*

Grupa 5:

LSB podatak _____
Binarni oblik *Dekadni oblik*

Grupa 6:

LSB podatak _____
Binarni oblik *Dekadni oblik*

Grupa 7:

LSB podatak _____
Binarni oblik *Dekadni oblik*

Grupa 8:

LSB podatak _____
Binarni oblik *Dekadni oblik*

Grupa 9:

LSB podatak _____
Binarni oblik *Dekadni oblik*

Grupa 10:

LSB podatak _____
Binarni oblik *Dekadni oblik*

Grupa 11:

LSB podatak _____
Binarni oblik *Dekadni oblik*

Grupa 12:

LSB podatak _____

*Binarni oblik**Dekadni oblik***Grupa 13:**

LSB podatak _____

*Binarni oblik**Dekadni oblik***Grupa 14:**

LSB podatak _____

*Binarni oblik**Dekadni oblik***Grupa 15:**

LSB podatak _____

*Binarni oblik**Dekadni oblik***Grupa 16:**

LSB podatak _____

*Binarni oblik**Dekadni oblik*

Tajmer 2 se proverava na isti na in.

Treba otvoriti potreban direktorijum i u itati fajl ije ime odgovara tajmeru 2 i MOD-u u kome se tajmer 2 testira.

6. Zaključak

Na efikasan način obrađeni su svi režimi rada tajmera 82c54. Primenljivost mikrokontrolera pokazala se i slučajevima ovog tajmera.

7. Literatura

1. AT89S8253 dokumentacija – www.atmel.com
2. Intel 82c54 CMOS Programmable Interval Timer datasheet, October 1994.
3. Procesori, programski jezici i hardversko-softverske karakteristike *embedded* sistema, Mile Stojčević

