

Vežba 3

Tajmeri/brojači i sistem prekida kod mikrokontrolera 8051

1. TAJMERI/BROJAČI

1.1 Tajmeri 0 i 1

Mikrokontroler 8051 sadrži dva šesnaestobitna tajmersko-brojačka registra. Označavaju se kao tajmer 0 i tajmer 1. Mogu služiti kao brojački registri, ili tajmeri koji mere zadati vremenski interval. Oba tajmera mogu raditi u četiri radna režima.

Kada radi kao brojač, sadržaj tajmerskog registra se uvećava za jedan na svaku silaznu ivicu odgovarajućeg ulaza mikrokontrolera (**T0** za tajmer 0 i **T1** za tajmer 1). Maksimalna frekvencija ulaznog signala koju brojač može da prati je 24 puta manja od radnog takta mikrokontrolera.

Kad radi kao tajmer, sadržaj tajmerskog registra se uvećava za jedan u svakom mašinskom ciklusu. Jedan mašinski ciklus traje 12 perioda radnog takta mikrokontrolera. (Ako je radni takt 12MHz, registar se uvećava svake mikrosekunde.)

(MSB)				(LSB)			
GATE	C/T	M1	M0	GATE	C/T	M1	M0
TAJMER 1				TAJMER 0			
				M1	M0		
GATE	kontrola gejt. Kada je ovaj bit na jedinici, odgovarajući tajmer može da broji kada je pripadajući TR bit jedinica i pripadajući INT ulaz mikrokontrolera visok. Ako je vrednost GATE bita 0, dovoljno je da samo odgovarajući bit TR bude visok			0	0	Mod 0. tajmer radi kao 13-bitni brojač	
				0	1	Mod 1. tajmer radi kao 16-bitni brojač	
				1	0	Mod 2. tajmer radi kao osmобitni brojač. Brojački registar je TL. nakon preticanja, TL se puni sadržajem TH registra (auto reload)	
C/T	ovaj bit određuje da li će odgovarajući tajmer da radi kao brojač opadajućih ivica na odgovarajućem T ulazu (C/T=1) ili kao tajmer (C/T=0)			1	1	Mod 3. Tajmer 1 je u ovom modu zaustavljen, a tajmer 0 radi kao dva odvojena osmобitna tajmera. Za tajmer 1 ovaj mod se ne koristi (zabranjeno je).	

Slika 1.1 Registar TMOD

Svakom tajmeru/brojaču pripadaju po dva osmобitna registra, **TH0** i **TL0** tajmeru 0, a **TH1** i **TL1** tajmeru 1.

Kontrola rada tajmera 0 i 1

Radom tajmera upravlja se pomoću registara **TMOD** (slika 1.1) i **TCON** (slika 1.2). Gornja četiri bita registra **TMOD** kontrolišu rad tajmera 1, a donja četiri rad tajmera 0. Bitovi **TR1** i **TR0** registra **TCON** određuju da li će odgovarajući tajmer/brojač biti aktivan ili ne. Npr. ako je brojač 0 neaktivan on se neće uvećavati kada na ulazu **T0** nastupi prelaz sa visokog na nizak nivo. Ovo je ilustrovano na slikama 1.3, 1.4 i 1.5. Način rada tajmera bira se kombinacijom bitova **M1** i **M0** registra **TMOD**.

(MSB)				(LSB)			
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
TF1	zastavica preteka tajmera 1. Automatski se briše kada se skoči na potprogram za opsluživanje prekida	TF0	zastavica preteka tajmera 0. Automatski se briše kada se skoči na potprogram za opsluživanje prekida	IE1	zastavica detektovanog spoljašnjeg zahteva za prekidom 1. Kada je zahtev detektovan, postavlja se na jedinicu, a briše se automatski kada se skoči na potprogram za opsluživanje prekida	IE0	zastavica detektovanog spoljašnjeg zahteva za prekidom 0. Kada je zahtev detektovan, postavlja se na jedinicu, a briše se automatski kada se skoči na potprogram za opsluživanje prekida
TR1	postavljanjem ovog bita tajmer 1 počinje da broji. U suprotnom ne broji	TR0	postavljanjem ovog bita tajmer 0 počinje da broji. U suprotnom ne broji	IT1	određuje da li je ulaz za spoljašnji prekid 1 osetljiv na opadajuću ivicu ili na nizak nivo	IT0	određuje da li je ulaz za spoljašnji prekid 0 osetljiv na opadajuću ivicu ili na nizak nivo

Slika 1.2 Registar TCON

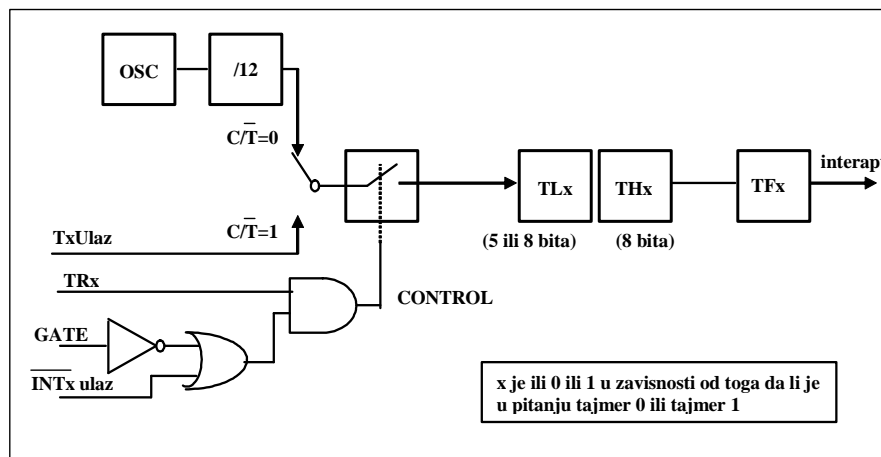
Biti **C/T** registra **TMOD** (u viša ili niža četiri bita u zavisnosti od tajmera) definišu da li odgovarajući tajmer radi kao tajmer (**C/T**=0) ili kao brojač (**C/T**=1). Biti **GATE** svojim stanjem 0 omogućuju odgovarajućem tajmeru da radi uvek kada je odgovarajući **TR** bit jednak jedinici. Ako je **GATE**=1, rad tajmera uslovljen je i stanjem na ulazu **INT1** odnosno **INT0** u zavisnosti od tajmera. Odgovarajući **INT** ulaz tada mora biti jedinica da bi tajmer radio.

Objašnjenja koja slede važe za oba tajmera osim tamo gde je drugačije napomenuto.

Modovi rada tajmera 0 i 1

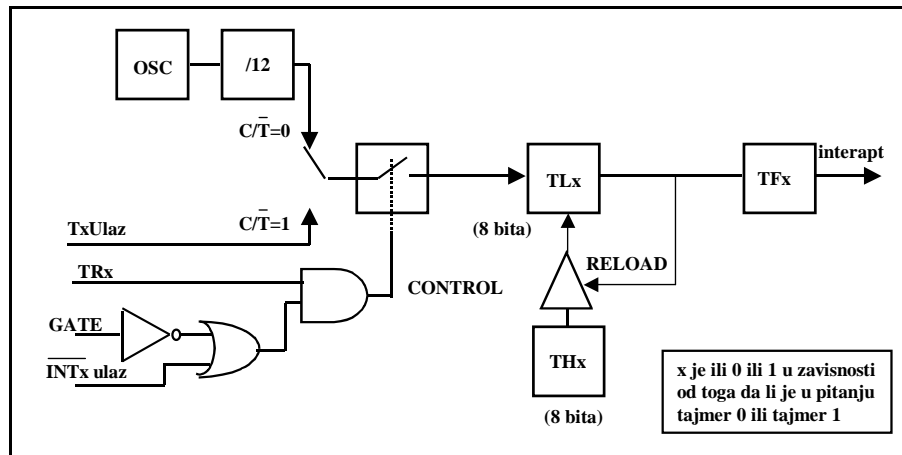
Mod 0

Ovaj mod je ilustrovan na slici 5.3. U ovom modu tajmer predstavlja 13-bitni tajmer/brojač. Kada svi bitovi registra postanu jedinice, sledećim uvećanjem sadržaja registra svi bitovi postaju nule. To se zove preticanje tajmera/brojača. Preticanje izaziva postavljanje odgovarajuće interapt zastavice (bit **TF1** registra **TCON** za tajmer 1 i bit **TF0** za tajmer 0). Postavljanje interapt zastavice izaziva prelazak na izvršavanje servisne rutine ako je odgovarajući interapt dozvoljen. (Biti **TF1** i **TF0** se automatski vraćaju na vrednost 0 kada se desi skok na interapt servisnu rutinu).



Slika 1.3 Mod 0 i 1 Tajmera 0 ili 1

Pomenuti 13-bitni registar sastoji se od svih osam bita **TH** i nižih pet bita **TL** odgovarajućeg tajmer registra. Viša tri bita **TL** registra u ovom modu su nedefinisana. Izbor moda 0 vrši se postavljanjem bita **M1** i **M0** u registru **TMOD** na 0.



Slika 1.4 Mod 2 Tajmera 0 ili 1

Mod 1

Ovaj mod je po svemu isti kao i mod 0 s tim što je brojački registar u ovom slučaju 16-bitni. Koristi se svih osam bitova i **TL** i **TH** registra. Bit **TF1** ili **TF0** se opet postavlja pri prelasku iz stanja svih šesnaest bita jednako 1 u stanje svih šesnaest bita jednako 0.

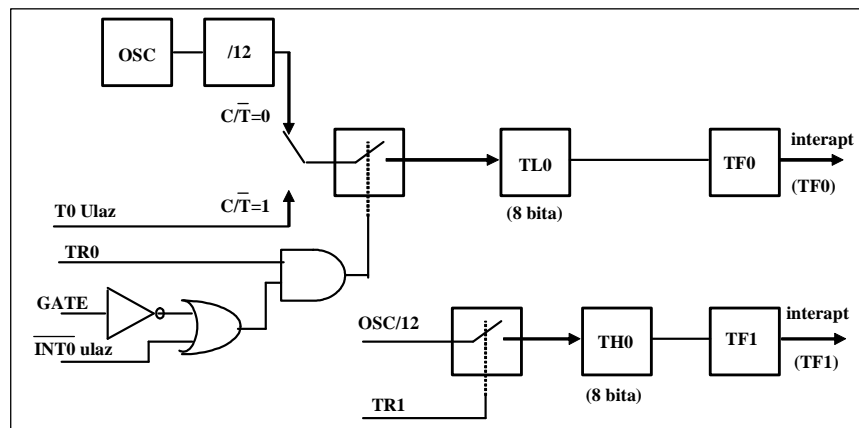
Mod 2

U ovom modu tajmer radi kao osmobitni tajmer sa automatskim punjenjem inicijalne vrednosti nakon preticanja. Rad u modu 2 ilustrovan je na slici 1.4. Kao tajmer/brojač uvećava se odgovarajući **TL** registar, a kada dođe do preticanja postavlja se bit **TF1** ili **TF0** u zavisnosti od tajmera, dok se vrednost iz odgovarajućeg **TH** registra upisuje u **TL**. Vrednost **TH** registra ostaje neizmenjena.

Mod 3

Ovaj mod različito prihvataju tajmer 0 i tajmer 1. Kada se tajmer 1 prebaci u mod 3 on jednostavno prestane da broji zadržavajući svoje prethodno stanje, kao da je bit **TR1**=0. Ako je tajmer 0 u modu 3, a tajmer 1 u bilo kom modu osim moda 3, tajmer 1 nastavlja svoj regularan rad u datom modu samo što ne može da generiše interapt jer je bit **TF1** upravljan tajmerom 0.

Tajmer 0 u modu 3 radi kao dva odvojena osmobitna tajmera. Bitovi **GATE**, **TR0** i **C/T** (i ulazi **T0** i **INT0**) kontrolišu brojanje registra **TL0** čijim preticanjem se postavlja bit zastavica **TF0**. Registar **TH0** broji mašinske cikluse, a njegovim preticanjem postavlja se bit zastavica **TF1**. Bit **TR1** kontrolišu brojanje **TH0**. Dakle, tajmerom 0 je kontrolisano izazivanje interapta i tajmera 0 i tajmera 1.



Slika 1.5 Mod 3 Tajmera 0

1.2 Tajmer 2

Tajmer 2 je 16 – bitni tajmer/brojač koji može da radi u tri moda: capture, auto-reload i kao baud rate generator. Tajmer 2 sadrži dva 8 – bitna registra **TH2** i **TL2**, koji uvek rade kao jedinstveni 16-bitni brojač. Osim toga postoje još dva dodatna registra, **RCAP2H** i **RCAP2L** koji se takođe ponašaju kao 16-bitni registar (**RCAP2**). Kada je u funkciji tajmera, **TH2:TL2** registar se uvećava svakog mašinskog ciklusa. Pošto se mašinski ciklus sastoji od 12 perioda oscilatora, brzina brojanja je 1/12 frekvencije oscilatora.

Kontrola rada tajmera 2

Upravljački registar **T2CON** prikazan je na slici 1.6:

7	6	5	4	3	2	1	0
TF2	EXF2	RCLK	TCLK	EXEN2	TR2	$C/\overline{T2}$	$CP/\overline{RL2}$

Slika 1.6 Registar **T2CON**

TF2: Setuje se kod prekoračenja brojača (promena vrednosti sa 0FFFFh na 0000h) i mora da se resetuje softverski. Ovaj bit neće biti setovan ako se tajmer 2 koristi u modu *baud rate* generatora, tj. ako je **RCLK**=1 ili **TCLK**=1.

EXF2: Setuje se kad negativna ivica na **T2EX** nožici izazove prepis stanja u **RCAP2** u tajmer 2, ili iz tajmera 2 u **RCAP2** (zavisno od moda u kom tajmer 2 radi, konkretno od bita $CP/\overline{RL2}$). Kad je prekid tajmera 2 omogućen (setovani biti **ET2** i **EA** u registru **IE**), **EXF2** će (ako je setovan) uzrokovati prekid. Ovaj bit se, kao i **TF2**, ne resetuje automatski već se mora resetovati softverski.

RCLK: Setovanje ovog bita prebacuje tajmer 2 u režim *baud rate* generatora za prijemnu sekciju serijskog porta.

TCLK: Kao i kod **RCLK** bita, ali za predajnu sekciju serijskog porta. To omogućava da se generišu različite učestanosti za prijem (**RXD**) i predaju (**TXD**), jer se za jedan od ova dva smera može generisati učestanost pomoću tajmera 1, a za drugi (kome je setovan **RXD** ili **TXD**), pomoću tajmera 2. Ako su učestanosti za prijem i predaju jednake mogu da se setuju i **RCLK** i **TCLK**. U režimu *baud rate* generatora ulazni takt tajmera 2 je 1/2 a ne 1/12 frekvencije oscilatora.

EXEN2: Ovaj bit, ako je setovan i ako se tajmer 2 ne koristi u *baud rate* generatorskom modu (**RCLK**=0 i **TCLK**=0), tada se na opadajuću ivicu signala na nožici **T2EX** izaziva prepis sadržaja iz tajmera 2 u **RCAP2** ako je $CP/\overline{RL2}$ setovan (*capture* mod), a prepis iz **RCAP2** u tajmer 2 ako je taj bit resetovan (*auto-reload* režim).

TR2: Ovo je start/stop bit za tajmer 2. Ako je **TR2**=1, tajmer je startovan.

$C/\overline{T2}$: Kada je setovan, tajmer 2 radi kao brojač opadajućih ivica na nožici **T2EX** (takt se dovodi spolja), a ako je resetovan, tajmer 2 se taktuje internim taktom, čija je frekvencija 1/12 učestanosti oscilatora.

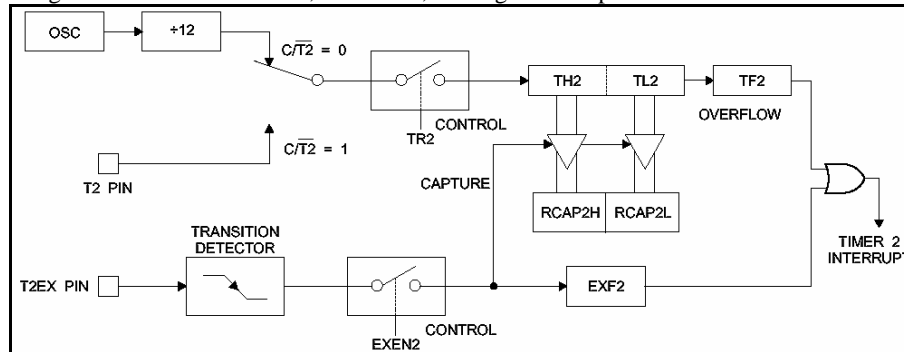
$CP/\overline{RL2}$: Ovo je *capture/auto-reload* selekcija. Kada je ovaj bit setovan, tajmer 2 je u *capture* modu (na silaznu ivicu signala na nožici **T2EX** stanje tajmera 2 se prebacuje u **RCAP2** registar). Kada je ovaj bit resetovan, onda je u *auto-reload* modu (tajmer 2 se puni iz registra **RCAP2** na prelazu iz stanja 0FFFFh u 0000h). Ako je setovan **RCLK** ili **TCLK**, važeći mod je *baud rate* generator i stanje ovog bita se ignoriše.

Modovi rada tajmera 2

U sledećoj tabeli prikazani su mogući modovi tajmera 2:

$RCLK+TCLK$	$CP/\overline{RL2}$	TR2	MOD
0	0	1	16-bit Auto-Reload
0	1	1	16-bit Capture
1	X	1	Baud Rate generator
X	X	0	Off

Slika 1.7 prikazuje tajmer 2 u *capture modu*. Ako je **EXEN2=0**, tajmer 2 je 16-bitni tajmer ili brojač, čije prekoračenje setuje bit **TF2** u registru **T2CON**. Ovaj bit se može koristiti za generisanje prekida. Ako je **EXEN2=1**, tajmer 2 i dalje radi na isti način, a na svaku opadajuću ivicu signala na ulazu **T2EX**, u registar **RCAP2** se upisuje trenutno stanje tajmera 2, pri čemu se istovremeno setuje bit **EXF2** u registru **T2CON**. Bit **EXF2**, kao i **TF2**, može generisati prekid.



Slika 1.7 Tajmer 2 u Capture modu

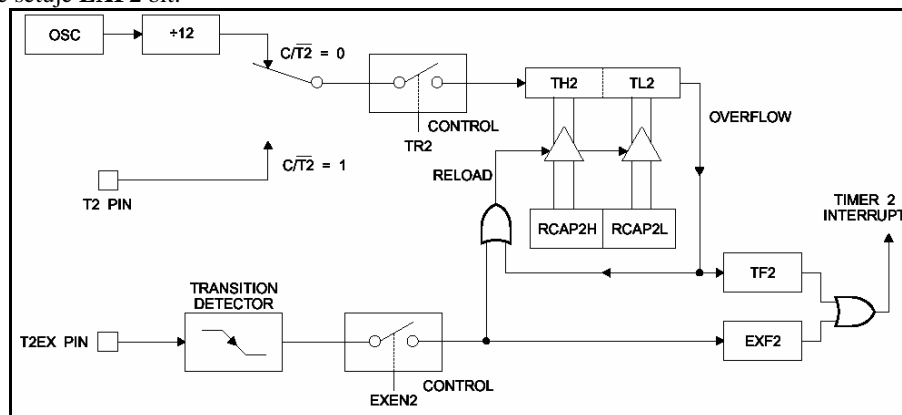
U *auto-reload modu* tajmer 2 može biti programiran da broji na gore/dole kada je konfigurisan u 16-bitnom auto-reload modu. Ta opcija se bira pomoću **DCEN** (Down Counter Enable) bita, koji se nalazi u registru **T2MOD** (ovaj registar ne postoji u standardnom 8052), koji je prikazan u sledećoj tabeli.

7	6	5	4	3	2	1	0
-	-	-	-	-	-	T2OE	DCEN

Simbol	Funkcija
-	Bit nije implementiran, rezervisan je za buduću nadogradnju
T2OE	Tajmer 2 Output Enable bit
DCEN	Kada je DCEN=1 tajmer 2 je konfigurisan kao brojač na gore/dole

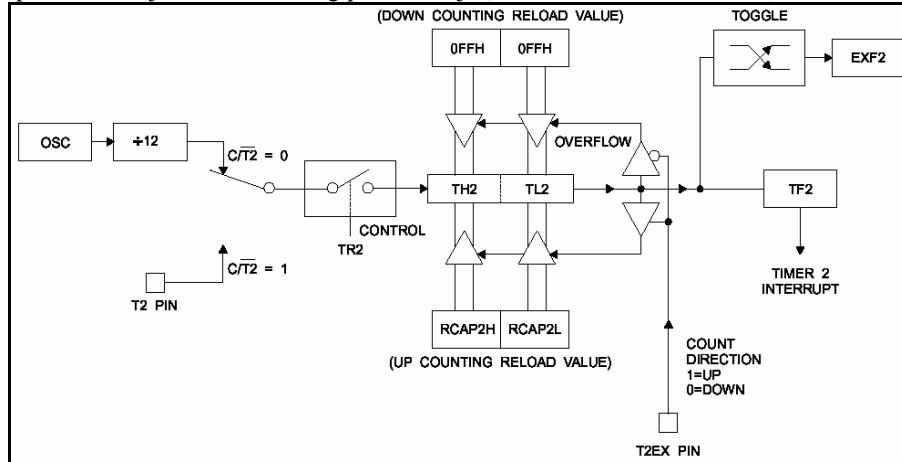
Nakon resetu, **DCEN** je postavljen na nulu pa tajmer 2 inicijalno broji na gore. Kada se bit **DCEN** setuje, tajmer 2 može da broji na gore ili na dole, u zavisnosti od vrednosti signala na nožici **T2EX**.

Slika 1.8 prikazuje tajmer 2 kada je **DCEN=0** (ovo je standardni način za 8052). U tom modu postoje dve opcije u zavisnosti od bita **EXEN2** u registru **T2CON**. Ako je **EXEN2=0**, tajmer 2 broji na gore do 0FFFFh, a na sledeći takt setuje bit prekoračenja **TF2**. Prekoračenje takođe prouzrokuje da registri tajmera 2 (**TH2** i **TL2**) dobiju vrednost registra **RCAP2** (16-bitni reload). Vrednost registra **RCAP2** (**RCAP2H** i **RCAP2L**) se postavljaju softverski. Ako je **EXEN2=1**, 16-bitni reload će se desiti ili prilikom prekoračenja, ili na silaznu ivicu signala na nožici **T2EX**. Ta silazna ivica signala takođe setuje **EXF2** bit.



Slika 1.8 Tajmer 2 u Auto-reload modu (DCEN=0)

Na slici 1.9 prikazan je slučaj kada je **DCEN=1** (ova konfiguracija postoji samo kod AT89C52, a ne i u standardnom 8052) tako da tajmer 2 može da broji i na gore i na dole, zavisno od stanja signala na nožici **T2EX** (1-broji na gore, 0-broji na gore). U slučaju brojanja na gore, tajmer 2 se ponaša kao u standardnom *auto-reload* režimu (tajmer se puni iz **RCAP2** na prelazu iz 0FFFh u 0). Ako tajmer broji na dole, tada se, u momentu izjednačavanja sadržaja tajmera 2 i registra **RCAP2**, tajmer 2 puni sa 0FFFFh. U oba slučaja punjenja tajmera 2 setuje se i bit **TF2**. Pri tome, bit **EXF2** ne izaziva prekid, a setuje se nakon svakog prekoračenja (može se koristiti kao 17. bit).

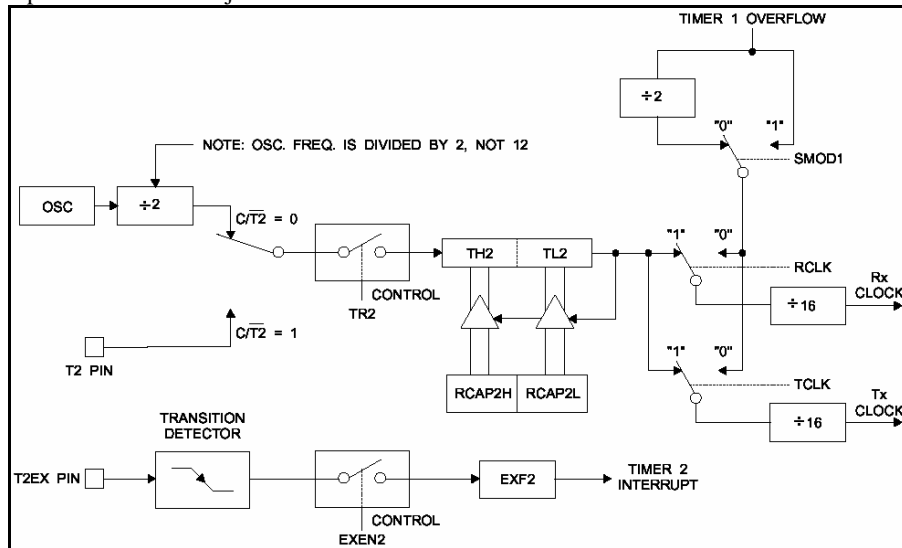


Slika 1.9 Tajmer 2 u Auto-reload modu (DCEN=1)

Slika 1.10 prikazuje tajmer 2 u *baud rate generator modu*. Da bi tajmer 2 radio u tom modu potrebno je setovati bit **TCLK** i/ili **RCLK** u registru **T2CON**. Ovaj mod je sličan *auto-reload* modu, po tome što prekoračenje tajmera 2 uzrokuje njegovo punjenje sadržajem registra **RCAP2**. U ovom režimu tajmer 2 se taktuje sa 1/2 frekvencije oscilatora, pa se brzina komunikacije u modovima 1 i 3 određuje sledećom jednačinom:

$$\text{BaudRate} = \frac{\text{Timer2_Overflow_Rate}}{16} = \frac{\text{Oscillator_Frekvency}}{32 * [65536 - (\text{RCAP2H}, \text{RCAP2L})]}$$

U ovom načinu rada tajmera 2 prekoračenje tajmera 2 ne setuje **TF2**, tako da ne prouzrokuje prekid. Ako je **EXEN2=1**, negativna ivica signala na nožici **T2EX** može da setuje **EXF2**, ali to ne utiče na rad tajmera 2. Takođe, u ovom režimu se ne preporučuje upis u **TH2**, **TL2**, **RCAP2H** i **RCAP2L** dok tajmer 2 radi (**TR2=1**). Pre eventualnog upisa tajmer 2 treba zaustaviti (**TR2=0**), a tek nakon upisa se može ponovo startovati tajmer.



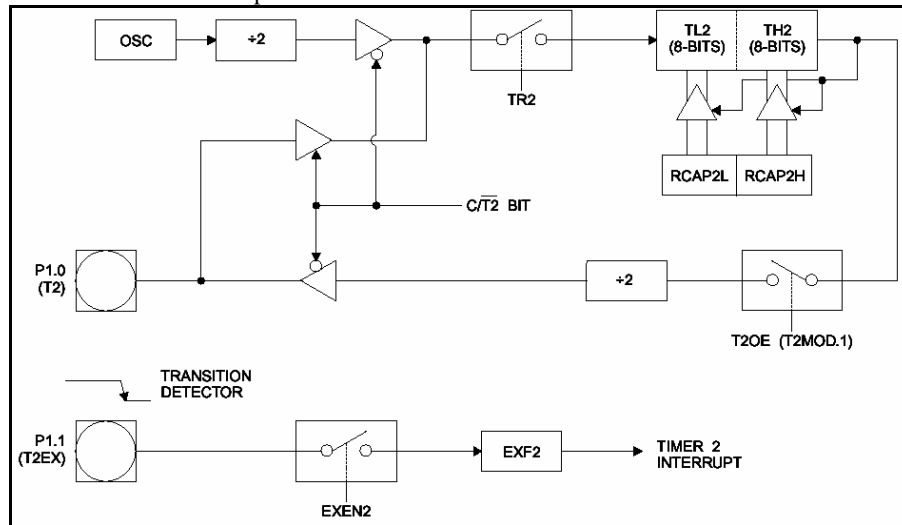
Slika 1.10 Tajmer 2 u Baud Rate Generator modu

Na slici 1.11 prikazano je dobijanje takta faktora ispune od 50% promenljive učestanosti na nožici P1.0 (važi samo za AT89C52). Sem već opisane funkcije, ova nožica se može koristiti kao izlaz takta,

faktora ispune 50% i frekvencije u opsegu $1/(4 \cdot 65536)$ do $1/4$ frekvencije oscilatora. Da bi se konfigurisao tajmer/brojač 2 kao **generator takta** (clock generator), potrebno je da $C/\overline{T2}=0$ i $T2OE=1$. Bit $TR2$ startuje i zaustavlja tajmer. Frekvencija takta zavisi od frekvencije oscilatora i vrednosti sadržaja registra $RCAP2$, koji se upisuje u registre tajmera 2:

$$\text{Clock_Out_Frequency} = \frac{\text{Oscillator_Frequency}}{4 \cdot [65536 - (RCAP2H, RCAP2L)]}$$

Kada se tajmer 2 koristi na ovaj način, on ne generiše prekid. Moguće je koristiti tajmer 2 kao *baud rate* generator i kao generator takta istovremeno. Međutim, brzina prenosa i frekvencija takta nisu međusobno nezavisne veličine pošto obe koriste **RCAP2H** i **RCAP2L**.



Slika 1.11 Tajmer 2 u Clock-out modu

2. SISTEM PREKIDA MIKROKONTROLERA 8051

U mikroprocesorskim sistemima često je potrebno istovremeno pratiti rad više perifernih jedinica. To se može postići neprestanim prozivanjem jedne po jedne jedinice, proveravajući njihova stanja. Ako se utvrdi da je došlo do neke promene na nekoj od tih perifera na koju treba reagovati, preduzimaju se odgovarajuće akcije kao odgovor na promene. Na primer, nakon startovanja konverzije A/D konvertora neprekidno se vrši čitanje stanja **BUSY** nožice konvertora, koje označava da li je konverzija u toku ili je završena. Ako neko očitavanje pokaže da je konverzija završena, tada se učitava odgovarajući podatak sa konvertora. U ovom slučaju mikrokontroler je neprestano zauzet proverom stanja na liniji **BUSY** A/D konvertora.

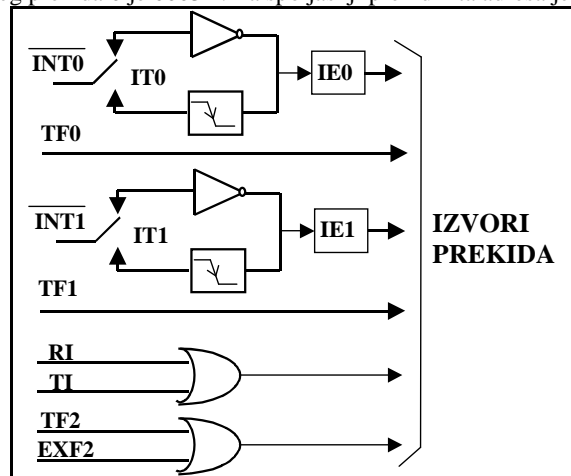
Drugi način praćenja rada više perifernih jedinica je da same jedinice jave kada je potrebno opsluživanje. Ta metoda se naziva metodom prekida, dok se prethodno opisana naziva metodom prozivanja (pooling). Dakle, kada se koristi metoda prekida, periferna jedinica posebnim signalom javlja kada je neophodna reakcija upravljačkog uređaja tj. mikrokontrolera. Taj signal se zove zahtev za prekidom odnosno interaptom (interrupt request). Slučaj iz prethodnog primera bi se mogao rešiti povezivanjem **BUSY** nožice konvertora na odgovarajuću **INT** (spoljašnji prekid) nožicu mikrokontrolera. Kada se **BUSY** deaktivira izaziva se prekid. Nakon što mikrokontroler uvaži zahtev, prelazi na podprogram za opsluživanje prekida (u ovom slučaju čitanje vrednosti konverzije), a nakon toga nastavlja sa izvršavanjem programa gde je prekinut u trenutku stizanja zahteva za prekidom. U ovom slučaju mikrokontroler se ne opterećuje proverom stanja na periferiji nego samo izvodi odgovarajuću akciju kada je to potrebno. Iz ovoga je jasno da u ovom slučaju mikrokontroler potroši mnogo manje vremena za opsluživanje perifera, zbog čega je i program može biti mnogo efikasniji.

2.1 Izvori prekida kod 8051

Kod ovog mikrokontrolera postoji pet (šest za 8052) izvora prekida:

Spoljašnji prekidi 0 i 1 (external interrupt)

Mikrokontroler poseduje dva ulaza – **INT0** i **INT1** (deo su porta 3). Na njih se može dovesti signal zahteva za prekidom sa bilo kog uređaja iz okruženja mikrokontrolera. Dva bita, **IT0** i **IT1** registra **TCON** (pominjan kod tajmera), određuju hoće li ulazi **INT0** i **INT1** biti osetljivi na nizak nivo ili opadajuću ivicu signala zahteva za prekidom. Na primer, ako je **IT0=0** pojava niskog nivoa na **INT0** će se smatrati zahtevom za prekidom, dok u slučaju **IT0=1** samo opadajuća ivica signala na **INT0** izaziva prekid, a sledeći zahtev može se desiti se tek kada se **INT0** vrati na visok nivo i ponovo padne na nizak (slika 2.1). U oba slučaja važeći zahtev za prekidom postavlja odgovarajući bit **IE** registra **TCON** na jedinicu (zahtev na **INT0** bit **IE0**, a na **INT1** bit **IE1**). Tek postavljanje ovih bita izaziva prelazak na odgovarajući potprogram za opsluživanje prekida (ako je prekid dozvoljen tj. nije maskiran u registru **IE**). Bitovi **IE0** i **IE1** se automatski vraćaju u stanje 0 kada mikrokontroler pređe na potprogram za opsluživanje odgovarajućeg prekida. Adresa od koje počinje potprogram za opsluživanje spoljašnjeg prekida 0 je 0003H. Za spoljašnji prekid 1 ta adresa je 0013H.



Slika 2.1 Izvori prekida kod 8051/8052

Prekidi tajmera 0 i 1 (timer interrupt)

Preticanje tajmera 0 ili 1, u bilo kom modu, može predstavljati zahtev za prekidom. Kao što je pomenuto ranije, preticanje tajmera 0 ili 1 postavlja bite **TF0** ili **TF1** registra **TCON**, označavajući da je došlo do preticanja. Ako je odgovarajući prekid dozvoljen, tj. nije maskiran, prelazi se na potprogram za opsluživanje prekida, a time se automatski bit **TF0** ili **TF1**, u zavisnosti od opsluženog prekida, vraća na vrednost 0. Servisni potprogram za prekid tajmera 0 počinje na adresi 000BH, a za prekid tajmera 1 na adresi 001BH.

Prekid serijskog interfejsa (serial port interrupt)

Mikrokontroler poseduje i dvosmerni serijski interfejs. Ukoliko je dozvoljen prekid serijskog interfejsa, zahtev za prekidom će se desiti svaki put kada je završeno slanje reči preko serijske veze, kao i svaki put kada je primljena cela reč. Servisni potprogram za prekid serijskog interfejsa počinje na adresi 0023H.

Prekid tajmera 2 (za 8052)

Prekid tajmera 2 generiše se kao logičkim sabiranjem (logičko ili) bita **TF2** i **EXF2** u registru **T2CON**. Nijedan od ova dva bita se ne briše hardverski kada se pređe na opsluživanje prekida koji izazivaju. U stvari, funkcija za opsluživanje prekida mora prethodno na osnovu stanja ovih bita da utvrdi koji od ta dva izvora je zapravo prekid izazvao i tek nakontoga da softverski obriše odgovarajući bit.

2.2 Maskiranje prekida

Svaki od pomenutih prekida se može dozvoliti ili zabraniti tj. maskirati. Kada je prekid maskiran, odgovarajući biti (**TF0**, **TF1**, **IE0** i **IE1**) se postavljaju na 1, ali to ne izaziva prelazak na odgovarajući servisni potprogram. Kod 8051 svaki od izvora prekida se može maskirati, što se kontroliše preko registra **IE** (*interrupt enable*, slika 2.2).

Kao što se vidi na slici 2.2, postavljanjem određenog bita na 1 odgovarajući prekid biva dozvoljen. U suprotnom je maskiran. Najviši bit **EA** (*enable all*) kontroliše maskiranje svih prekida. Ukoliko je on 0, svi prekidi su maskirani bez obzira na stanje bitova istog registra koji kontrolišu maskiranost pojedinačnih izvora prekida.

(MSB)				(LSB)			
EA	X	(ET2)	ES	ET1	EX1	ET0	EX0
EA	ako je ovaj bit 0, svi prekidi su maskirani. Ako je 1, svaki prekid se pojedinačno maskira pomoću ostalih bita ovog registra			ET1	ako je ovaj bit jednak jedinici, prekid zbog preticanja tajmera 1 je dozvoljen, u suprotnom je maskiran		
ET2	ako je ovaj bit jednak jedinici, prekid zbog preticanja tajmera 2 je dozvoljen, u suprotnom je maskiran. Bit postoji samo u 8052.			EX1	ako je ovaj bit jedinica, spoljašnji prekid 1 je dozvoljen, u suprotnom je maskiran		
ES	ako je ovaj bit jednak jedinici, prekid serijskog porta je dozvoljen, inače je maskiran			ET0	ako je ovaj bit jednak jedinici, prekid zbog preticanja tajmera 0 je dozvoljen, u suprotnom je maskiran		
				EX0	ako je ovaj bit jedinica, spoljašnji prekid 0 je dozvoljen, u suprotnom je maskiran		

Slika 2.2 Registar za maskiranje prekida (IE)

2.3 Struktura prioriteta prekida

Kod 8051 svaki prekid može da ima dva nivoa prioriteta. Ako se tokom opsluživanja nekog prekida desi novi prekid, opsluživanje prethodnog će se prekinuti samo ako je novi prekid višeg prioriteta. Prioritet prekida određuje stanje registra **IP** (slika 2.3). Ako je neki bit 0, njemu pripadajući prekid je nižeg prioriteta, a u suprotnom višeg. Na taj način se svi prekidi dele u dve grupe po prioritetu. U slučaju da dva (ili više) prekida nastupe istovremeno, prvo se opslužuje prekid sa višim, a zatim sa nižim prioritetom. Ako su oba prekida istog prioriteta, tada je redosled prioriteta fiksna i sledeći: spoljašnji prekid 0, tajmer 0, spoljašnji prekid 1, tajmer 1 i na kraju prekid serijskog interfejsa. Naravno, ovaj redosled važi samo ako zahtevi za prekidom stignu istovremeno.

(MSB)				(LSB)			
X	X	(PT2)	PS	PT1	PX1	PT0	PX0
PT2	postavljanjem ovog bita na jedinicu, prekid tajmera 2 se programira kao prekid višeg prioriteta (samo u 8052)			PX1	postavljanjem ovog bita na jedinicu, spoljašnji prekid 1 se programira kao prekid višeg prioriteta		
PS	postavljanjem ovog bita na jedinicu, prekid serijskog porta se programira kao prekid višeg prioriteta			PT0	postavljanjem ovog bita na jedinicu, prekid tajmera 0 se programira kao prekid višeg prioriteta		
PT1	postavljanjem ovog bita na jedinicu, prekid tajmera 1 se programira kao prekid višeg prioriteta			PX0	postavljanjem ovog bita na jedinicu, spoljašnji prekid 0 se programira kao prekid višeg prioriteta		

Slika 2.3 Registar za kontrolu prioriteta prekida (IP)

2.4 Procedura pozivanja potprograma za opsluživanje prekida

Potrebno je još razmotriti kako se tačno odvija priznavanje zahteva za prekidom i prelazak na potprogram za opsluživanje prekida. Nakon izvršenja svake pojedine instrukcije proverava se stanje svih bita koji označavaju validan zahtev za prekidom (**IE0**, **TF0**, **IE1**, **TF1**, **TI** i **RI**). Svaki od ovih bita može biti postavljen automatski, validnim zahtevom za prekidom (uobičajena situacija), ili softverski, upisivanjem u odgovarajući registar (softversko izazivanje prekida) – efekat je isti. Ako se utvrdi da postoji zahtev za prekidom, a pri tome:

- Taj prekid nije maskiran.
- Nije u toku opsluživanje prekida istog ili višeg prioriteta.
- Instrukcija koja je sledeća po redu nije **RETI** ili instrukcija upisa u registre **IE** ili **IP**.

tada se na stek stavlja sadržaj programskog brojača (**PC** – 16 bita), a sam programski brojač se puni adresom početka servisnog potprograma tog konkretnog prekida (registar **PSW** – registar stanja - se *NE* čuva na steku). Početne adrese servisnih potprograma su fiksne i unapred zadate za svaku vrstu prekida:

vrsta prekida	početna adresa servisnog potprograma
spoljašnji prekid 0	0003H
prekid tajmera 0	000BH
spoljašnji prekid 1	0013H
prekid tajmera 1	001BH
prekid serijskog porta	0023H
prekid tajmera 2	002BH

Ovo je, u principu, ekvivalentno izvršenju instrukcije **LCALL** sa početnom adresom servisnog potprograma kao argumentom.

Kada se servisni potprogram piše u mašinskom jeziku, potrebno je obratiti posebnu pažnju na sledeće. Servisni potprogram se obavezno mora završiti instrukcijom **RETI** koja sa steka skida poslednji gurnuti 16-bitni podatak i ubacuje ga u programski brojač, što je adresa instrukcije koja je trebalo da se izvrši kada je prekid nastupio. Dakle, nakon instrukcije **RETI** nastavlja se sa redovnim izvršavanjem programa. Važno je primetiti da isto to radi i instrukcija **RET** samo što ona ne obaveštava hardver mikrokontrolera da je prekid opslužen te je instrukcija **RETI** obavezna. Dodatno, za razliku od naredbe **RET**, naredba **RETI** briše interne indikatore tekućeg prekida, čime su novi prekidi istog prioriteta dozvoljeni. Ako se servisna funkcija piše pomoću C ili drugog prevodioca, za pomenuto će se prevodilac sam pobrinuti ako se servisna funkcija deklariše na odgovarajući način.

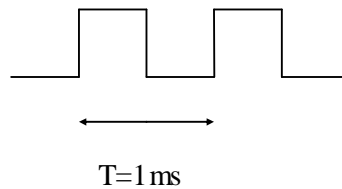
3. JEDNOSTAVNIJI PRIMERI

Osnovni primeri u assembleru

Sledi nekoliko primera sa rešenjem datim na mašinskom jeziku, odnosno sam kod programa je dat na simboličkom jeziku gde je svaka mašinska instrukcija data sa slovnom (mnemoničkom) oznakom –u assembleru. U svim zadacima se koriste prekidi.

Generisanje pravougaonih impulsa

Sa mikrokontrolerom 8051, čija je frekvencija spoljašnjeg oscilatora 12MHz realizovati povorku pravougaonih impulsa jednakog trajanja (četvrtki) frekvencije $f=1\text{kHz}$ (slika 3.1) na portu **P1**, bit 0 (**P1.0**). Mikrokontroler isprogramirati u assembleru.



Slika 3.1 Vremenski oblik izlaznog signala

Rešenje:

Kako je potrebna kontrola vremena trajanja stanja 0 i 1 impulsa, koristiće se tajmer 0. Sad je pitanje u kom režimu treba tajmer da radi. Da bi se to znalo, treba prvo videti koliko impulsa tajmer treba da izbroji. Pošto je potrebno 12 perioda ciklusa takta oscilatora za jedan taktni impuls tajmera, perioda ulazne frekvencije tajmera je:

$$T_{\text{timer}} = 12 \cdot T_{\text{osc}} = 1 \mu s$$

Kako se tajmer uvećava za 1 posle T_{timer} , a potrebno je da se stanje impulsa menja svakih $T/2$, onda je broj koji tajmer treba da odbroji pre nego što napravi prekid (interapt):

$$n = \frac{T/2}{T_{\text{timer}}} = 500$$

Pošto ovoliki broj ne može da izbroji osmobarbitni tajmer, uzima se 16-bitni. Sad treba odrediti od kog broja tajmer treba da počne da broji, imajući u vidu da tajmer broji na više i da se prekid događa kada tajmer prelazi iz stanja FFFFh u stanje 0:

0000 - 01F4 = FE0C jer je 500dec=01F4hex.

Program:

```
DSEG                                ; naredni segment se odnosi na
                                    ; interni RAM
    ORG    20h                      ; podaci pocinju na 20h
Var1: DS    1                      ; rezervisanje jednog bajta
                                    ; (bice na adresi 20h)
Stanje BIT  Var1.0                 ; dodeljivanje adrese bita
Izlaz  BIT  P1.0                   ; simbolima Stanje i Izlaz

CSEG                                ; naredni segment odnosi na
                                    ; programsku memoriju
    ORG    0000H                   ; kad se startuje, program
                                    ; kreće od adrese 0000h
    AJMP   INICIJALIZACIJA

    ORG    000BH                   ; kod za prekid - adresa 000Bh
    AJMP   PREKID                  ; skok na prekidnu rutinu
```

INICIJALIZACIJA:

```

MOV    IE,#82H      ; dozvola interapta
                      ; prekoračenja tajmera 0
                      ; (EA=1,ET0=1)
MOV    TMOD,#01H    ; tajmer 0 je 16-bitni
MOV    TH0,#0FEH    ; početna vrednost
MOV    TL0,#0CH     ; tajmera je FE0C
SETB   Stanje       ; jer je početno stanje porta 1
SETB   TR0          ; start tajmer 0 (TR0=1)

```

PETLJA:

```

NOP                ; program se vrti u ovoj
                  ; beskonačnoj petlji i čeka
SJMP   PETLJA      ; prekide tajmer 0

```

PREKID:

```

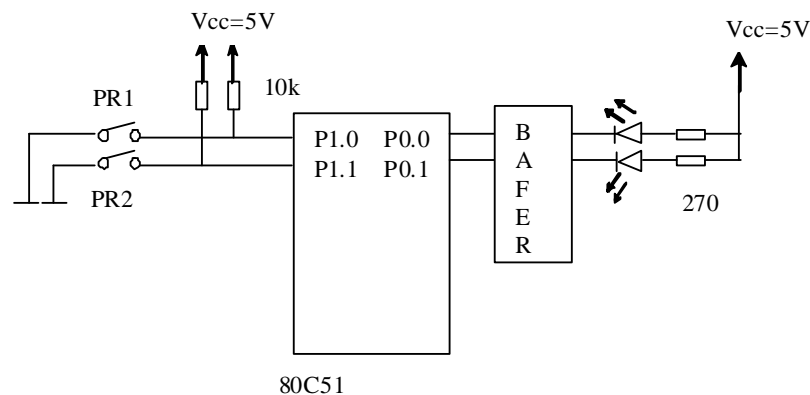
CLR    TR0          ; tajmer je potrebno zaustaviti
                      ; pre punjenja,
MOV    TH0,#0FEH    ; a zatim ponovo inicijalizovati
                      ; tajmer 0, jer on
MOV    TL0,#0CH     ; kad jednom odbroji,
                      ; nastavlja da broji od 0
SETB   TR0          ; start tajmera
CPL    Stanje       ; komplementira tekuće stanje
MOV    C,Stanje     ; prebacuje tekuće stanje na izlaz
MOV    Izlaz,C      ;
RETI                    ; povratak iz prekidne rutine

```

END

Treptanje LED diode

Sa mikrokontrolerom 8051, čija je frekvencija spoljašnjeg oscilatora 12MHz izvršiti kontrolu rada dve LED diode. Izbor LED diode se vrši prekidačem **PR1** (kada je **PR1** otvoren trepće crvena, a kada je zatvoren trepće zelena LED dioda), a brzina treptenja LED diode se određuje prekidačem **PR2** (kada je **PR2** otvoren dioda treba da trepće sa učestanošću od 10Hz, a kada je zatvoren, dioda treba da trepće sa učestanošću od 0.5Hz). Obe LED diode treba da se uključe na nizak logički nivo nožice **P0.0** (crvena) i **P0.1** (zelena). Mikrokontroler isprogramirati u assembleru.



Rešenje:

Broj tajmerskih ciklusa koji treba da se odbroji kada je PR2 otvoren je $N1 = 50\text{ms}/1\mu\text{s} = 50000$, a kada je zatvoren $N2 = 1\text{s}/1\mu\text{s} = 1000000$. Kako 16-bitni tajmer može maksimalno da odbroji 65536 puta potrebno je uvesti pomoćni brojač u prekidnoj rutini za detekciju broja prekida. Pošto je $N2 = 20 * 50000$ vidi se da je potrebno da se desi 20 prekida pa da se onda izvrši promena logičkog nivoa signala na

nožici **P0.0** ili **P0.1**, za frekvenciju od 0,5Hz. U Timer0 je potrebno uneti vrednost: 0-50000=0000h-0C350h=3CB0h

Program:

```

F10    EQU    1                ; dodela vrednosti 1 simbolu F10
                                ; (označava da je na frekv.
                                ; 10Hz potreban 1 prekid pre
                                ; promene stanja LED diode)
F05     EQU    20              ; dodela vrednosti 20 simbolu F05
                                ; (označava da je na frekv. 0.5Hz
                                ; potrebno 20 prekida pre promene
                                ; stanja LED diode)
CLED    BIT    P0.0           ; izlazno stanje crvene LED diode
ZLED    BIT    P0.1           ; izlazno stanje zelene LED diode
PR1     BIT    P1.0           ; ulazno stanje prekidača PR1
PR2     BIT    P1.1           ; ulazno stanje prekidača PR2

DSEG                                ; naredni segment se odnosi na
                                ; interni RAM
                                ORG    20h                ; naredni podaci se smestaju od
                                ; adrese 20h unutar internog RAM-a
Mask: DS    1                  ; rezervisanje jednog bajta
                                ; (bice na adresi 20h)
Count:     DS    1            ; rezervisanje jos jednog bajta
                                ; (bice na adresi 21h)

CSEG                                ; nastavak predstavlja programski kod
                                ; i smestace se u programsku memoriju
                                ORG    0000H              ; program ce se u programsku memoriju
                                ; pocev od adrese zadate sa ORG: 0h
                                JMP     INICIJALIZACIJA

                                ORG    000BH              ; kao odgovor na prekid tajmera 0,
                                ; skace se na odresu 000Bh. Zato se
                                ; naredna instrukcija mora naci na
                                ; adresi 000Bh
                                JMP     PREKID            ; skok na prekidnu rutinu

INICIJALIZACIJA:
    MOV     IE,#82H            ; dozvola interapta
                                ; prekoračenja Tajmera0
    MOV     TMOD,#01H          ; tajmer 0 je 16-bitni
    MOV     TH0,#3CH           ; početna vrednost
    MOV     TL0,#0B0H          ; tajmer 0 je 3CB0h
    MOV     R4,#1              ; R4 <- 1, R4 je pomoćni brojač
                                ; za brojanje N2
    SETB    TR0                ; start tajmera 0

MAIN:                                ; program se vrti u ovoj petlji
                                ; (proverava prekidače) i čeka
                                ; prekide
    MOV     A,#1               ; izbor aktivne LED diode
    JB      PR1,Crvena         ; skoči na labelu Crvena ako je PR1=1
    MOV     A,#2
    SETB    CLED               ; P0.0=1 (zabrana paljenja crvene
                                ; LED diode)
    SJMP    L1

Crvena:
    SETB    ZLED               ; P0.1=1 (zabrana paljenja
                                ; zelene LED diode)

```

```
L1:  MOV    MASK,A
      MOV    A,#F10      ; Izbor frekvencije
      JB     PR2,PostaviBrojac
      MOV    A,#F05
PostaviBrojac:
      MOV    Count,A
      SJMP   MAIN        ; nazad u petlju

PREKID:
      CLR    TR0          ; Tajmer0 je potrebno zaustaviti
      MOV    TH0,#3CH     ; početna vrednost
      MOV    TL0,#0B0H    ; Timera0 je 3CB0
      SETB   TR0
      DJNZ   R4,Int_Kraj
      MOV    R4,Count
      MOV    A,Mask
      XRL    P0,A
Int_Kraj:
      RETI
END
```