

*Dr. sc. Danko Kezić*

**LABORATORIJSKE VJEŽBE IZ BRODSKIH PROCESNIH  
RAČUNALA**

*U Splitu, 21.01. 2004.*

## VJEŽBA –1 UVOD U LABORATORIJSKE VJEŽBE S MIKROKONTROLERIMA

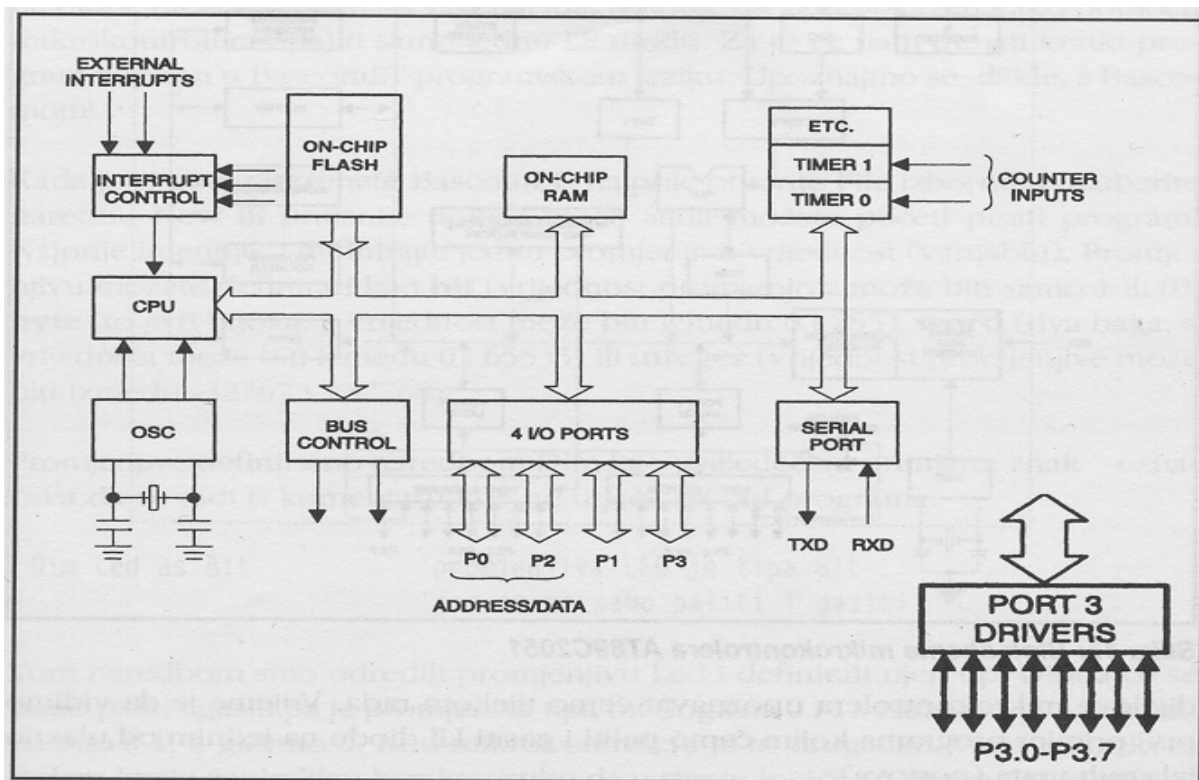
Nagli razvoj elektronike uvjetovao je razvoj mikrokontrolera, koji su u cijelosti integrirani u jednom čipu. Njihova cijena iz dana u dan pada i danas iznosi reda veličine nekoliko desetaka US\$. Više se i ne mogu zamisliti ozbiljnije elektroničke naprave bez mikrokontrolera.

Unatoč tome, većina električara koji se bave analognom i klasičnom digitalnom tehnikom zazire od upotrebe mikrokontrolera. Razlog tome je što je građa, programiranje i uporaba mikrokontrolera prilično zamršena. Potrebno je poznavati asemblerski ili C – jezik. Ipak, danas postoje programski alati koji omogućavaju bitno lakše programiranje i time omogućuju da se sve veći broj električara upozna sa programiranjem i radom sa mikrokontrolerima.

Svrha ovih laboratorijskih vježbi jeste upravo da studenti na što je moguće lakši način nauče osnove programiranja mikrokontrolera. Pri tome će koristiti tri alata. Prvi alat je programski jezik BASCOM – LT u kojemu će studenti pisati svoje programe. Drugi alat je Programator PG302 koji služi da se razvijeni i asemblirani programi prenesu na mikrokontroler. Treći alat je BASCOM TESTNA PLOČICA na kojoj će studenti provjeriti da li mikrokontroler pravilno izvodi program koji su u njega unijeli.

### OSNOVNA GRAĐA MIKROKONTROLERA

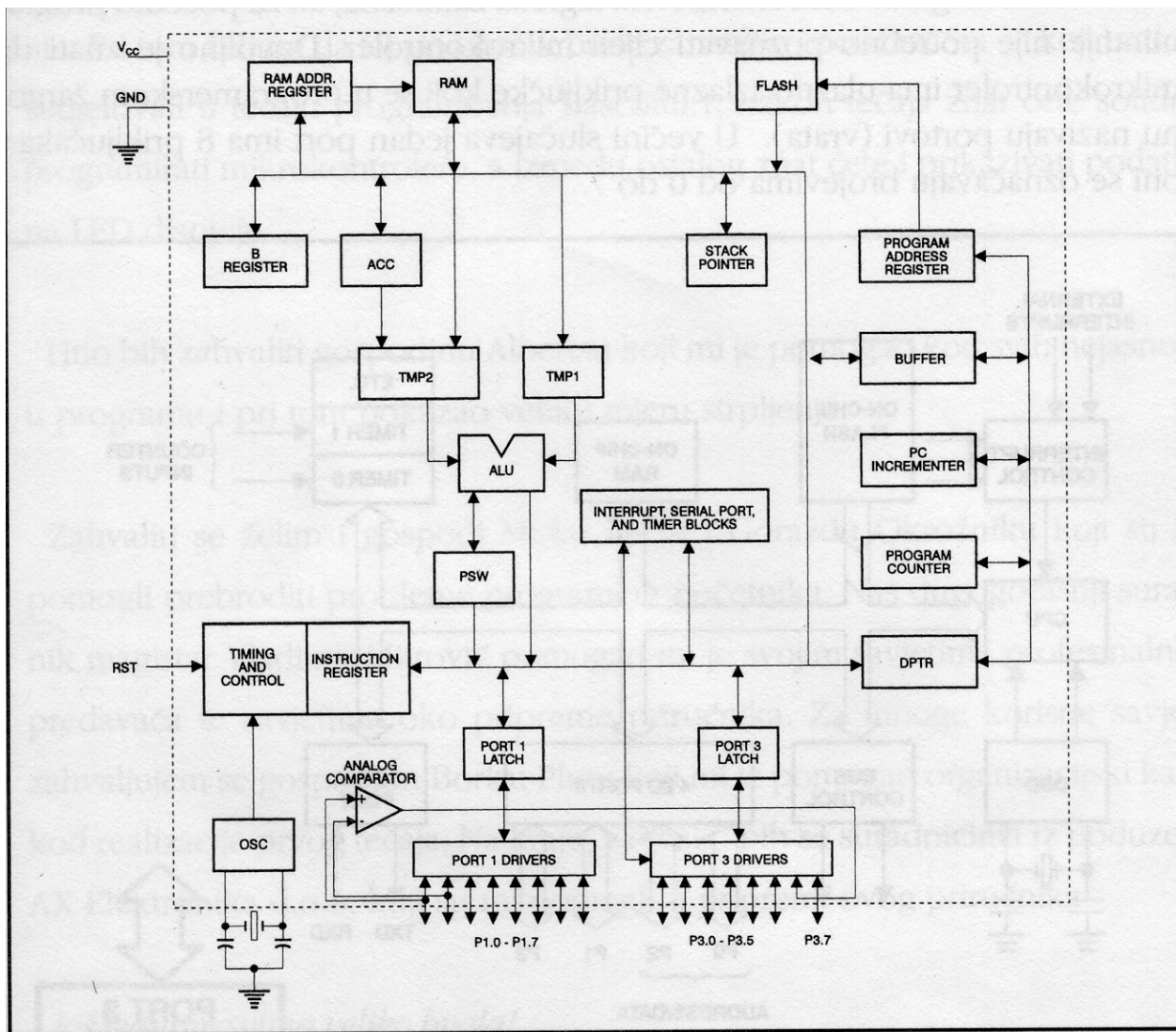
Osnovna građa mikrokontrolera porodice 8051 prikazana je na slici 1



Slika 1: Osnovna građa mikrokontrolera serije 8051

Mikrokontroler se sastoji od mikroprocesora ili CPU (centralne upravljačke jedinice) koja izvodi program koji smo napisali. CPU pogoni oscilator taktnog signala koji osigurava stabilan i točan takt od 12 MHz. Na CPU je priključena 8 bitna sabirnica podataka na kojoj se nalazi povezano niz jedinica (BUS control, 4 I/O ports,

serial port, timeri, interrupt controler, izbrisiva ROM memorija – on chip flash i mala RAM memorija ). Sve to zajedno čini mikrokontroler. Ono što je bitno znati bez ulaza u detalje funkcioniranja mikrokontrolera ja da se mikrokontroler povezuje sa vanjskim svijetom preko ulazno-izlaznih portova (vrata ). Mikrokontroler može imati 2 ili 4 porta. Mikrokontroler koji ćemo koristiti u laboratorijskim vježbama ima 2 porta sa 8 priključaka na svakom portu. Priključci na portu se označavaju kao P1.0, P1.1 ... do P1.7. Prvi broj označava broj porta, a drugi broj označava broj priključka na tom portu. Pojam ulazno-izlazni znači da se programski može postaviti da pojedini portovi ili priključci u tim portovima budu ulazni ili izlazno ovisno o aplikaciji. Na slici 2 prikazana je detaljnija blok shema mikrokontrolera AT89C2051 kojeg ćemo koristiti u našim vježbama.



Slika 2: Detaljna blok shema mikrokontrolera 8051

## BASCOM LT BASIC COMPILER ZA MIKROKONTROLE OBITELJI 8051

BASCOM-LT je novi viši programski jezik koji je nastao u USA, a koji je isključivo namijenjen programiranju mikrokontrolera. BASCOM-LT je jezik vrlo sličan BASIC-u i stoga je veoma lagan za učenje jer ima velike sličnosti sa BASICom, kojeg su studenti već naučili u kolegiju Osnove informatike. Naravno ovaj jezik ima i

svoje specifičnosti. Program se pokreće na PC-iju. Omogućuje pisanje programa za mikrokontroler, njegovo prevođenje u strojni jezik i prenošenje strojnog jezika u mikrokontroler koji je preko serijskog izlaza i programatora PG302 spojen sa PCijem.

U tablici 1 se vidi većina naredbi u BASCOMLT. Te naredbe omogućuju pregledno, strukturirano programiranje ( IF-THEN-ELSE-ENDIF, DO-LOOP ili WHILE-END ). Osim toga, posebnim naredbama možemo upravljati TIMERIMA, COUNTERIMA ili INTERRUPT-ima. Možemo posezati i izravno do sadržaja internih registara mikrokontrolera, a u BASCOM programu možemo uključiti djelove programa pisane u assembleru. Dakle vidimo da BASCOM ima osim značajki višeg programskog jezika i značajke nižeg programskog jezika. Sve naredbe su dobro dokumentirane u helpu programa BASCOM-LT. Nazivi varijabli i oznake programskih dijelova (labele) mogu biti dugački 32 znaka. Možemo koristiti BIT, BYTE, WORD ili INTEGER varijable.

<b>Naredba</b>	<b>Značenje</b>
<b>DEKLARACIJE</b>	
<b>DIM var AS type</b>	Definicija promjenjive
<b>DIM symbol AS CONST value</b>	Definicija konstante
<b>BIT, BYTE, WORD, INTEGER</b>	Tipovi podataka
<b>DEFBIT, DEFBYTE, DEFINT</b>	Default deklaracija promjenjivih
<b>LOGIČKE I ARITMETIČKE OPERACIJE</b>	
<b>AND, MOD, NOT, OR, XOR</b>	Logičke operacije
<b>DEC var, INC var</b>	Smanjenje, povećanje za 1
<b>SET bit, RESET bit</b>	Postavlja ili briše pojedini bit
<b>ROTATE var, LEFT / RIGHT</b>	Rotiranje bitova neke promjenjive u lijevo /u desno
<b>SWAP var1, var2</b>	Zamjena vrijednosti dviju promjenjivih
<b>GETDATA, SETDATA</b>	Međumemorija
<b>PRETVORBE</b>	
<b>CHR, BCD</b>	Pretvorba iz binarne vrijednosti u Character ili BCD
<b>MAKEBCD, MAKEDEC</b>	Pretvorba između BCD i decimalne vrijednosti

Tablica 1: Popis naredbi BASCOM-LT

Naredba	Značenje
<b>NAREDBE ZA TVORBU PROGRAMSKIH STRUKTURA</b>	
DO ... naredbe ... LOOP UNTIL izraz	DO petlja
FOR var=start TO/DOWNT0 end ... naredbe ... NEXT	FOR-NEXT petlja
WHILE izraz ... naredbe ... WEND	WHILE petlja
EXIT FOR/DO/WHILE	Neprosredni izlaz iz petlje
IF izraz THEN ... naredbe ... ENDIF	Uvjetni skokovi
CALL (do 9 parametara)	Pozivanje rutina s prijenosom parametara
GOSUB, RETURN	Poziv potprograma i povratak
GOTO	Skok na oznaku
ON var GOTO/GOSUB	Granjanje ovisno o vrijednosti promjenjive
STOP, END	Zaustavljanje, kraj programa
<b>ULAZNO IZLAZNE NAREDBE</b>	
INKEY	Prihvata ASCII vrijednost sa serijskog kanala
INPUT ("tekst"), var INPUTHEX("tekst"), var	Unos preko RS232 međusklopa
PRINT var; "tekst" PRINTHEX var	Izlaz preko RS232 međusklopa
<b>PREKIDI (INTERRUPT)</b>	
ON INTx, SERIAL, TIMERx	Definiranje prekidnih rutina za različite vrste prekida
PRIORITY, DISABLE, ENABLE	Kontrola prekida
<b>BROJAČ/TIMER</b>	
LOAD TIMERx, vrijednost	Inicijalizacija timera
START TIMERx, STOP TIMERx	Pokretanje, zaustavljanje timera
COUNTER x	Broji impulse na P3.4, P3.5
<b>OPĆENITO</b>	
REM	Komentar
SOUND pin, trajanje, frekvencija	Generiranje jednostavnih tonova
DELAY	Kratkotrajno kašnjenje
WAIT sekunde, WAIT ms	Kašnjenje proizvoljnog trajanja
BITWAIT x, SET/RESET	Čekanje dok neki bit nije postavljen/izbrisan
<b>MODE - NAČINI RADA CPU</b>	
IDLE, POWERDOWN	Postavljanje mikrokontrolera u stanje manje potrošnje
<b>I2C naredbe</b>	
I2CRECEIVE, I2CSEND	Prijam, pošiljanje podataka nekoj I2C jedinici
I2CSTART, I2CSTOP	Start, odn. Stop uvjet na I2C sabirnici
I2CRBYTE, I2CWBYTE	Prijam, pošiljanje jednog bajta
<b>NAREDBE ZA UPRAVLJANJE RADOM LCD</b>	
CLS	Brisanje displaya i postavljanje kurzora u početni položaj
CURSOR ON/OFF BLINK/NOBLINK HOME UPPER/LOWER DISPLAY ON/OFF	Upravljanje kursorom i displejom
LCD var, LCDHEX var	Ispis vrijednosti promjenjive na LCD
LOCATE y,x	Pozicioniranje kursora
LOWERLINE, UPPERLINE	
SHIFTLCD LEFT/RIGHT	Pomicanje sadržaja LCD-a lijevo/desno
SHIFTCURSOR LEFT/RIGHT	Pomicanje kursora lijevo/desno

Tablica 1: Popis naredbi BASCOM-LT

## BASCOM – TESTNA PLOČA ZA PORODICU MIKROKONTROLERA 8051

BASCOM testna ploča razvijena je za porodicu mikrokontrolera 8051. Testna ploča predstavlja univerzalno radno okruženje za testiranje programiranih mikrokontrolera. Na slici 2 i 3 prikazana je montažna i električna shema BASCOM testne ploče.

IC1 i IC2 su 20 polna i 40 polna podnožja koja služe da bi se u njih uložili programirani mikrokontroleri. Ploča je predviđena za testiranje 20 pinskih i 40 pinskih mikrokontrolera. U našim vježbama koristit ćemo isključivo 20 pinske mikrokontrolere AT89C2051, pa ćemo stoga za testiranje koristiti samo podnožje IC1. Na ploči su ugrađeni 12 MHz kvarc i pripadajući elementi za generiranje takta mikrokontrolera. Na ploči se nalazi niz od 2 x 8 LED dioda i četiri 7 segmentna displeja. Na ploči se nalaze 3 tipkala ( TP1, TP2 i TP3 ) pomoću kojih se mogu simulirati digitalni ON/OFF signali sa vanjskih senzora. Tipkalo TP4 služi za resetiranje mikrokontrolera. Funkcija kratkospojnika i ostalih elemenata navedena je kako slijedi:

Kratkospojnici JP4/JP1/JP5 i JP3/JP2/JP6 – služe da bi se kod 40 pinskih mikrokontrolera koji imaju 4 porta po želji odabrala 2 porta koja se spojuju na LED diode ili na displeje. Dakle moguće je kod 40 pinskih mikrokontrolera odabrati izlaze P1 (JP1/JP4) ili P2 (JP1/JP5) na lijevom setu dioda ( D1 – D8 ), te izlaze P3 (JP2/JP6) ili P4 (JP2/JP3) na desnom setu dioda ( D9 – D16). S obzirom da ćemo mi u našim vježbama koristiti isključivo 20 pinske mikrokontrolere koji imaju samo 2 porta (P1 i P3), kratkospojnici trebaju biti u položaju JP4/JP1 i JP2/JP6.

JP10 i JP11 – uključuju napajanje za lijevi (D1-D8) i desni (D9-D16) set LED dioda

JP12, JP13, JP14, JP15 – uključuju se pojedini 7 segmentni displeji

Važno je napomenuti da naš mikrokontroler ne može istovremeno pogoniti i diode i displeje. Stoga je nužno da se kratkospojnici JP12 – JP15 isključe kada radimo sa diodama i kada moraju biti uključeni kratkospojnici JP10 i JP11. Kada radimo sa displejima tada moramo uključiti kratkospojnike JP12 –JP 15, a isključiti JP10 i JP11.

Tipkalo TP1 je spojeno na P3.3 (JP2/JP6) ili P0.3 (JP2/JP3)

Tipkalo TP2 je spojeno na P 3.2 (JP2/JP6) ili P0.2 (JP2/JP3)

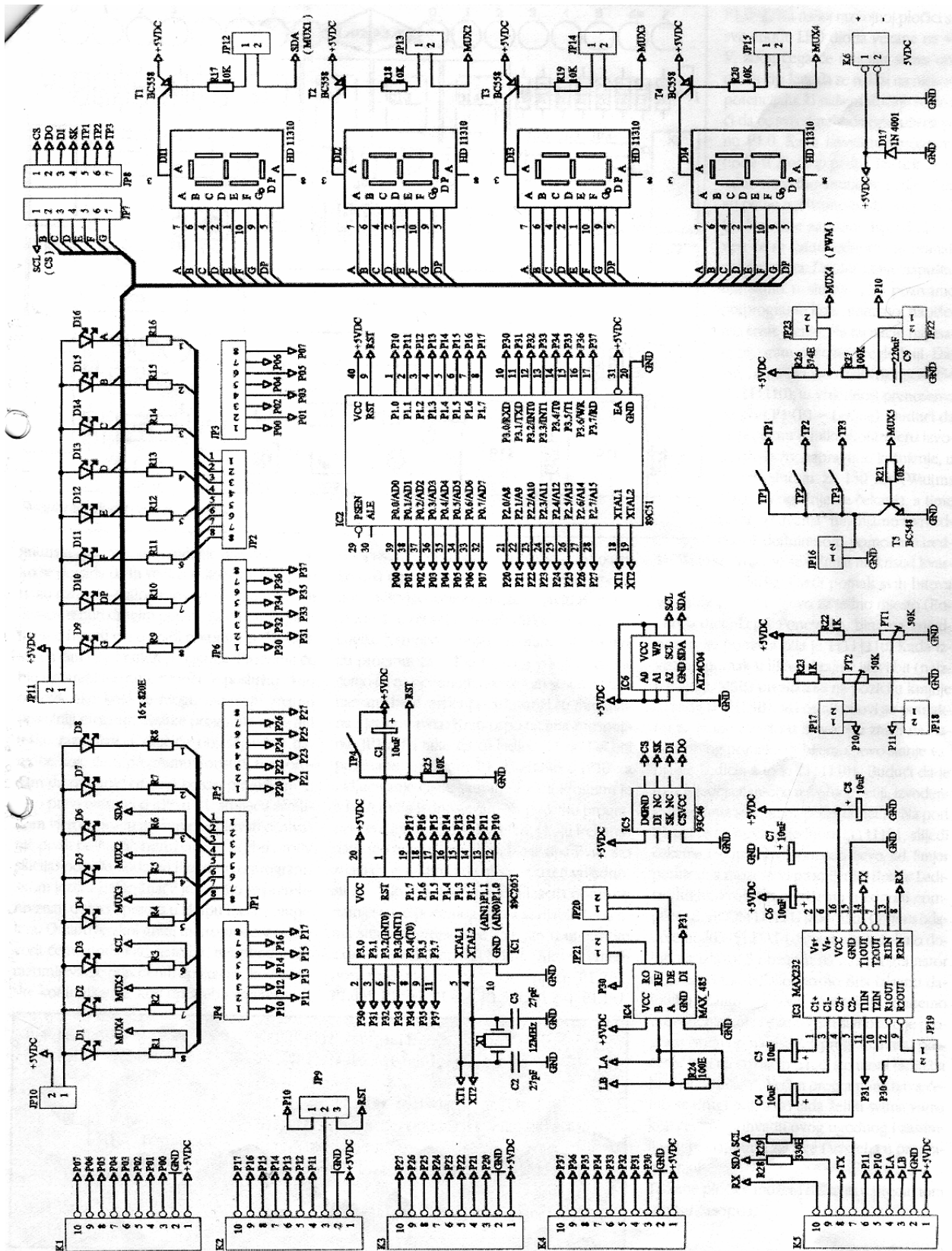
Tipkalo TP3 je spojeno na P 3.0 (JP2/JP6) ili P0.0 (JP2/JP3)

Tipkalo TP4 služi za RESET

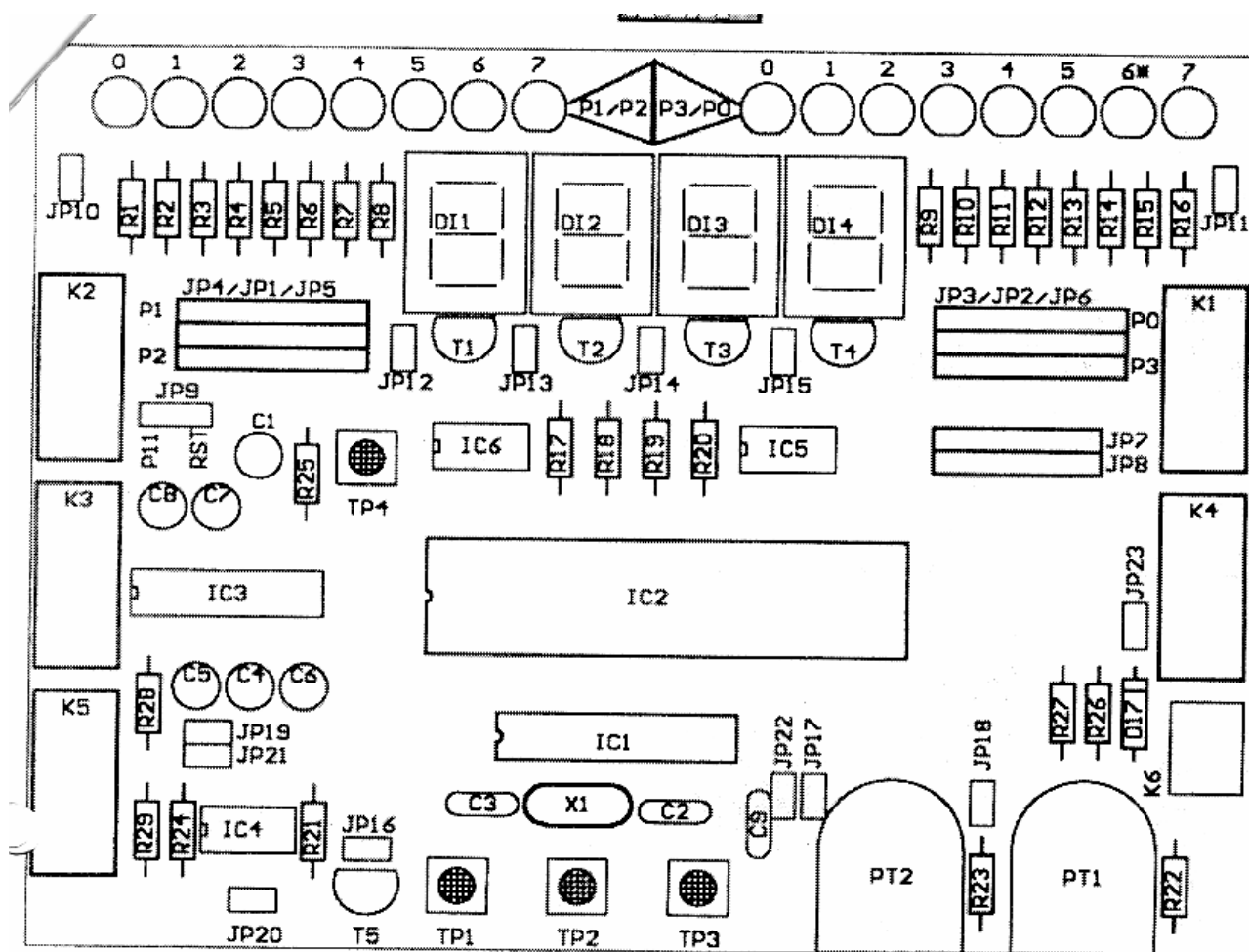
Tipkala TP1, TP2, TP3 možemo koristiti kao ulaze. Tipkala mogu biti multipleksirana (MUX 5) ili trajno spojena na masu što se odabire kratkospojnikom JP16.

Na konektorima K1, K2, K3, K4 su spojeni svi portovi sa mikrokontrolera tako da je omogućeno spajanje vanjskih jedinica (A/D pretvarača D/A pretvarača i slično) na mikrokontroler. ( Konektor K2 dodatno omogućuje programiranje procesora sa SPI sabirnicom izravno na ploči – u tom slučaju staviti JP9 u položaj RST, no to u našim laboratorijskim vježbama nećemo koristiti ).





Slika 3 – Električna shema Bascom-LT pločice



*Raspored elemenata na pločici*

Slika 4 – Montažna shema Bascom-LT pločice

Konektor K5 služi za spajanje mikrokontrolera sa testne pločice sa nekim drugim mikrokontrolerim preko RS232, RS485 ili I2C protokola. Za RS232 služe signali RX i TX, za RS485 služe signali LA i LB, a za I2C služe SDA i SCL. Želimo li koristiti koji od ovih protokola potrebno je samo utaknuti čip poput MAX232 (konvertira RS232 u TTL i obratno) ili MAX485 (konvertira RS485 u TTL i obratno), a sve ostalo se već nalazi na pločici. Osim toga na konektoru K5 su dostupni P1.0 i P1.1 za priključak vanjskih analognih ulaza.

Testna pločica podržava rad sa najraširenijim serijskim EEPROM-ima sa I2C sabirnicom (93C46 i 24C04) koji se mogu umetnuti u IC5 ili IC6. Ne mogu se koristiti oba EEPROM-a istovremeno.

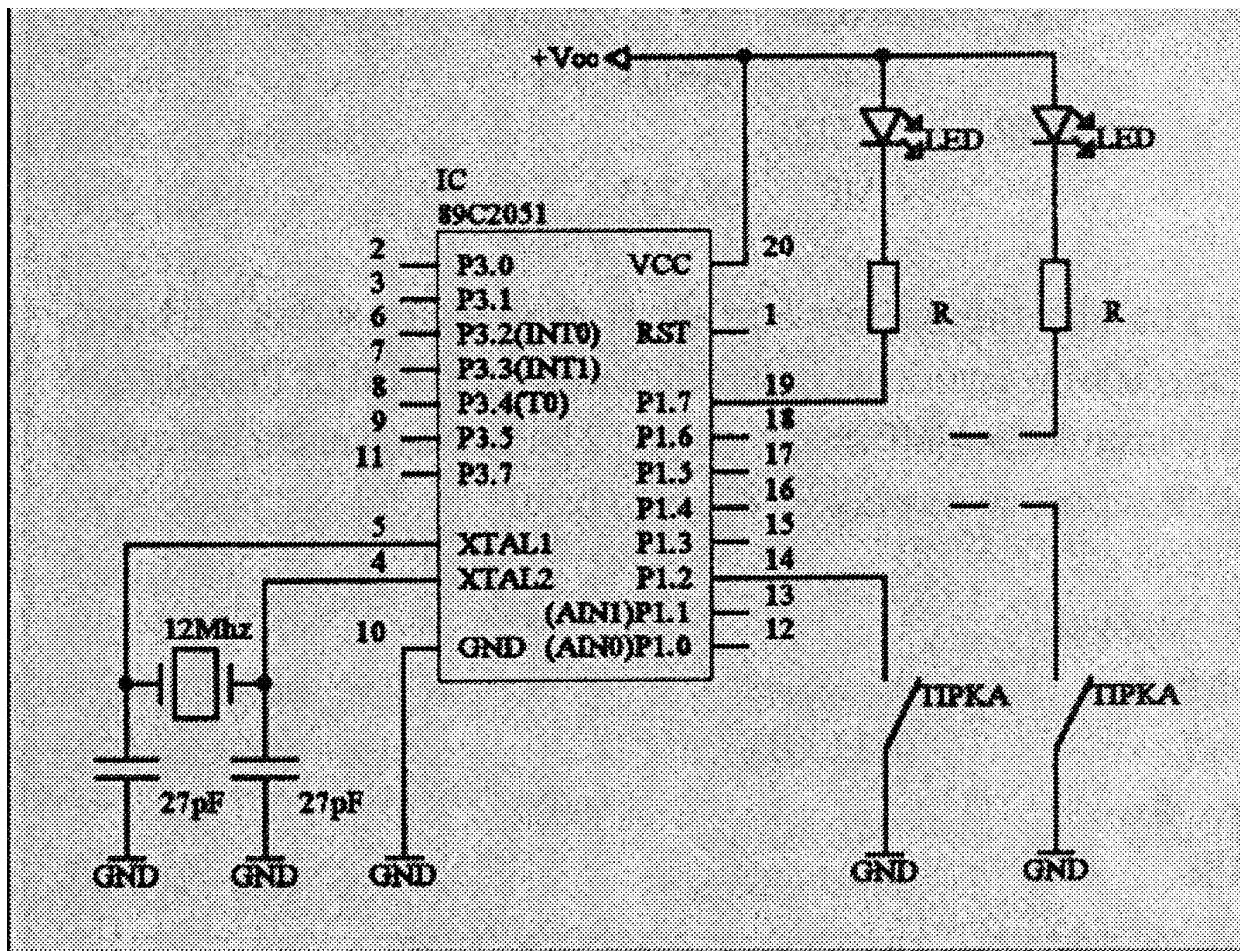
Kratkospojnici JP17 i JP18 omogućavaju spajanje dva trimera pomoću kojih se može simulirati analogni signal koji se dovodi na ulaz na P1.0 i P1.1 – mikrokontroler ima ugrađeni analogni komparator koji se takođe može koristiti u određenim aplikacijama.

R26, R27 i C9 su potrebni za testove sa A/D konvertorom.

## PRAVILA ZA PRIKLJUČENJE PINOVA MIKROKONTROLERA 8051



Pravilan priključak pinova mikrokontrolera prikazan je na slici 5.



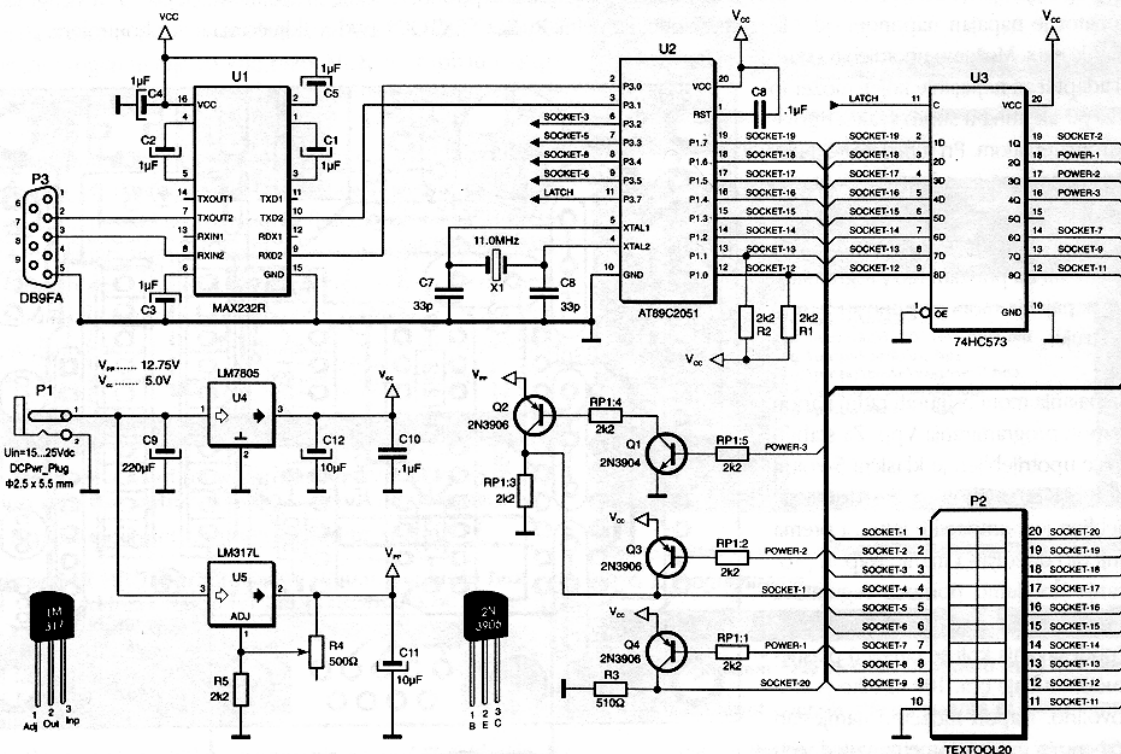
Slika 5: Priključenje mikrokontrolera

Kao što je iz slike vidljivo treba biti oprezan kada neko trošilo ( npr. LED diodu ) spajamo između napajanja i pinova mikrokontrolera. LED diode moramo spajati preko otpornika R, kako bi ograničili struju kroz mikrokontroler na maksimalno 20 mA, a koliko je dozvoljeno za porodicu mikrokontrolera 8051. Ukoliko trošilo spojimo bez otpornika, može poteći velika struja koja će trajno uništiti mikrokontroler. Kada je pin mikrokontrolera u stanju 1 dioda ne svijetli – kroz pin mikrokontrolera ne teče struja, a kad je pin u stanju 0, poteče struja kroz LED diodu koja je ograničena otporom R i ona počinje svijetliti. **Dakle portove mikrokontrolera ne smijemo direktno spajati na napajanje – samo uz predotporu.**

Pinove mikrokontrolera možemo bez posljedica kratko spojiti na masu. Razlog tome je što je maksimalna izlazna struja iz mikrokontrolera od nekoliko  $\mu\text{A}$  kada je pin u logičkoj jedinici. Dakle ako kratko spojimo sa masom pin koji je u logičkoj jedinici, napon na njemu pada i izlazni pin prelazi u logičko stanje 0. Kada spajamo tipkala sa pinovima mikrokontrolera, jedan kraj tipkala spajamo sa masom a drugi direktno sa pinom mikrokontrolera. Provjeravanje stanja uključenosti tipkala vrši se na način da se pin na kojem se nalazi tipkalo programski pokušava staviti u logičko stanje 1. Nakon toga se programski 'pročita' stanje na pinu. Ukoliko je tipkalo nije uključeno, pin će ostati u logičkoj jedinici, a ako je uključeno pin prelazi u logičko 0. **Portovi mikrokontrolera se mogu slobodno bez otpora direktno spajati na masu.**

## PROGRAMATOR PG-302

Programator PG-302 je namijenjen programiranju mikrokontrolera porodice 8051. Ovaj programator spada u klasu amaterskih i poluprofesionalnih uređaja. Električna shema je prikazana na slici 6.



Slika 6. Električna shema programatora PG-302

Programator služi da bi se datoteka sa programskim kodom iz PC-ija upisala u Flash memoriju (izbrisivi ROM) mikrokontrolera. Upisivanje podataka u FLASH memoriju se obavlja preko portova P1 i P3. Pri tome je potrebno po određenom algoritmu narinuti naponske valne oblike na RST pin mikrokontrolera koji se programira. Mikrokontroler se programira sa dva različita napona: radni napon od 5V i napon programiranja od 12,5 V. Programatorom se tako može programirati mikrokontroler 10000 puta. Svako upisivanje novog programa briše stari program.

Srce programatora je mikrokontroler 89C2051 koji radi taktom od 11 MHz, a čiji je zadatak da preuzme programski kod iz PC-ija te da ga po određenom algoritmu pohrani u memoriju mikrokontrolera kojeg programiramo. Datoteka sa programskim kodom stiže preko serijskog RS232C porta (9 pinskog) koji spaja PC i mikrokontroler za programiranje. IC MAX232C služi da bi pretvorila RS232C naponske nivoe od 9 V u TTL naponske nivoe.

Budući da AT89C2051 ima premalo portova za programiranje mikrokontrolera (ima cijeli samo port 1 dok mu je port 3 djelomično na raspolaganju) koristi se i IC74HC573 koji demultipleksira port 1 i upsuje u port 1 i port 3 mikrokontrolera kojeg programira te takođe preklapa između programskog i normalnog načina rada. Sa demultipleksorom upravlja mikrokontroler preko P3.7.

Programator se napaja naponom od 15 do 25 VDC (min 200 mA). Izvor napajanja mora osigurati stabilan radni napon  $U_{cc}$  od 5 V (LM7805 5V/1A regulator) i napon programiranja  $U_{pp}$  od 12,75 V (regulator LM317LZA cca. 100 mA). Izlazni napon programiranja moguće je podesiti pomoću trimer potencijometra R4.

Procesor IC1 sa signalima Power\_2 i Power\_3 preklapa između napona programiranja i RESET napona uključivanjem tranzistora Q1 ili Q3. Niski signal Power\_2 predstavlja RESET napon, a visoki signal Power\_3 predstavlja napon programiranja. Sa signalom Power\_1 uključujemo ili isključujemo radni napon mikrokontrolera kojeg programiramo.

## VJEŽBA 2 – UKLJUČIVANJE TROŠILA MIKROKONTROLEROM

U ovoj vježbi bit će prikazan način na koji se mogu uključivati i isključivati potrošači uz pomoć mikrokontrolera. Ukoliko se postave kratkospojnici kao što je na slici naznačeno, električna shema spoja izgleda kao na slici 7. LED dioda D15 ( šesta desno ) nije priključena na mikrokontroler, jer mikrokontroler nema izvod P3.6 ( taj izvod ima drugu svrhu ).

Zadatak vježbe je napisati više programa za paljenje i gašenje LED dioda. Na isti način kao što se upravlja LED diodama moguće je vršiti uključivanje i isključivanje većih potrošača kao što su releji, sklopnici i elektromotori. Naravno, za uključivanje većih potrošača neophodno je potrebno konstruirati odgovarajuće prilagođavajuće sklopove između kontrolera i potrošača.

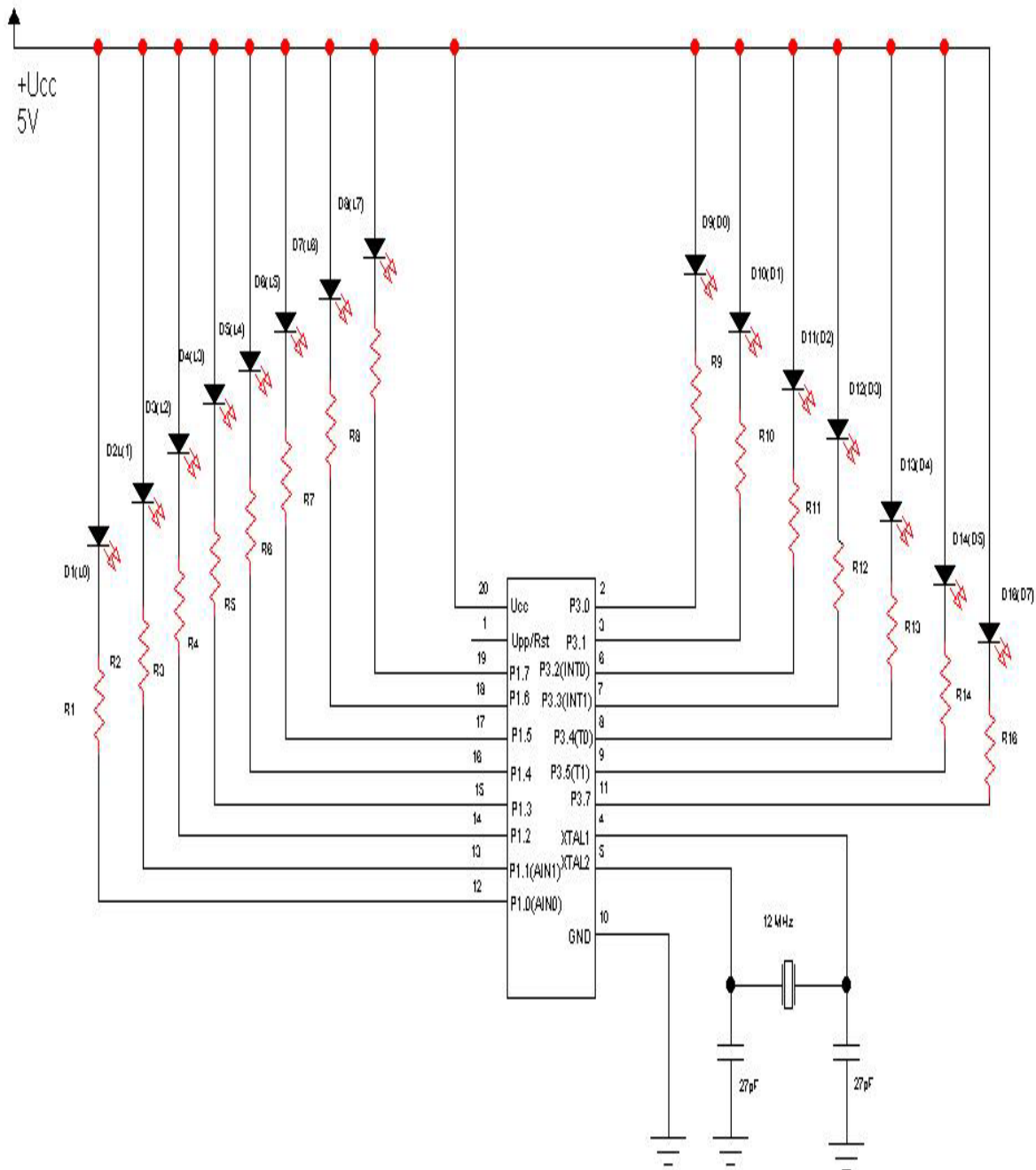
Samo programiranje u jeziku BASCOM-LT trebalo bi početi tek nakon što studenti izuče sve naredbe tog programskog jezika. No, s obzirom da je jezik dosta sličan BASIC-u kojeg studenti već znaju, naredbe ćemo upoznati kroz pisanje jednostavnijih programa. Primjer najjednostavnijeg mogućeg programa slijedi.

### Uključivanje i isključivanje jedne LED diode

Ovaj program pali i gasi LED diodu koja se nalazi priključena na P1.0 , a to je LED dioda D1 ( na pločici 0 lijevo ). Program će prvo upaliti LED diodu jednu sekundu, a zatim je ugasiti jednu sekundu. Beskonačna Loop petlja će se stalno ponavljati. Dobili smo žmigavac. Treba napomenuti da se LED dioda pali onda kad je P1.0 u logičkoj 0, a gasi se kad je P1.0 u logičkoj 1 ( shvatljivo iz sheme ). Mikrokontroler u stanju 1 ne može dati veću stuju od nekoliko  $\mu\text{A}$ , dok u stanju 0 može povući struju do 25 mA što je dovoljno za uključivanje trošila kao što je LED dioda.

<i>Dim LED as Bit</i>	<i>'definiranje varijable tipa Bit jer će se samo jedna LED 'dioda paliti i gasiti</i>
<i>Led=0</i>	<i>'definiranje varijable Led kojoj je početna vrijednost 0</i>
<i>Do</i>	<i>'početak Do-Loop petlje</i>
<i>P1.0 = Led</i>	<i>'Izlaz P1.0 poprima iznos varijable Led</i>
<i>Wait 1</i>	<i>'Pričekaj 1 sekundu</i>
<i>Led = Not Led</i>	<i>'invertirajmo vrijednost varijable</i>
<i>Loop</i>	<i>'kraj petlje</i>
<i>End</i>	

Nakon što ste napisali program u BASCOMLT, program valja iskompajlirati, a zatim strojni kod programa prebaciti pomoću programatora PG302 u mikrokontrolerski čip. Nakon što je programiranje uspješno završeno, premjestiti čip sa programatora na testnu pločicu, uključiti napajanje i istestirati rad mikrokontrolera.



Slika 7: Spoj mikrokontrolera 8051 sa LED diodama

**PAŽNJA: MIKROKONTROLER AT89C2051 JE IZRAĐEN U CMOS TEHNOLOGIJI ŠTO ZNAČI DA JE OSJETLJIV NA STATIČKI ELEKTRICITET. PRILIKOM PREMJEŠTANJA ČIPA SA TESTNE PLOČICE U PROGRAMATOR I OBRATNO NIKAKO SE NE SMIJU PRŠTIMA DODIRIVATI NOŽICE ČIPA, JER SE MIKROKONTROLER MOŽE TRAJNO UNIŠTITI.**

## PREPUSTITE ZA POČETAK PREMJEŠTANJE ČIPA DEMONSTRATORIMA DOK VI NE STEKNETE ODREĐENU RUTINU.

**Samostalna vježba 1: Pokušajte napisati program koji pali i gasi LED diodu D3 (na pločici 2 lijevo)**

### Paljenje više LED dioda odjednom

Ovaj program je vrlo sličan prethodnom programu, međutim razlika je u tome što ovaj program omogućava istodobno paljenje i gašenje cijele skupine LED dioda. U ovom programu varijabla Led je definirana kao 'Byte' a ne više kao 'Bit' što znači da ima 8 binarnih mjesta i može poprimiti dekadsku vrijednost između 0 i 255. Binarnom broju 00000000 odgovara dekadski broj 0, binarnom broju 11111111 odgovara dekadski broj 255.

```
Dim LED as Byte      'definiranje varijable tipa Byte jer će se samo paliti i gasiti 'grupa od 8 dioda
                    'koji pripadaju portu 1
Led=0                'definiranje varijable Led kojoj je početna vrijednost 0
Do
    P1 = Led          'početak Do-Loop petlje
                    'Izlazni port P1 poprima iznos varijable Led ( 00000000 )
    Wait 1            'Pričekaj 1 sekundu
    Led = Not Led     'invertirajmo vrijednost varijable ( 11111111 )
Loop
End                  'kraj petlje
```

### Pomične diode

Pomoću ovog programa moguće je dobiti efekt trčećeg svjetla. Potrebno je odrediti koji redosljed paljenja i gašenja LED dioda želimo i prema tome odrediti koji 8 bitni binarni broj poslati na port1 na kojemu su spojene diode.

```
Dim LED as Byte      'definiranje varijable tipa Byte jer će se samo paliti i gasiti 'grupa od 8 dioda
                    'koji pripadaju portu 1
Led=146              'definiranje varijable Led kojoj je početna vrijednost 146
                    '( 10010010 ) – gore LED 0,2,3,5,6 sa lijeva
Do
    P1 = Led          'početak Do-Loop petlje
                    'Izlazni port P1 poprima iznos varijable Led
    Wait 1            'Pričekaj 1 sekundu
    Led = 36          'vrijednost varijable LED je 00100100
    P1 = Led          'Izlazni port P1 poprima iznos varijable Led gore LED 0,1,3,4,6,7 sa lijeva
    Wait 1            'Pričekaj 1 sekundu
    Led = 73          'vrijednost varijable LED je 01001001
    P1 = Led          'Izlazni port P1 poprima iznos varijable Led
    Wait 1            'Pričekaj 1 sekundu
    Led = 146         'vrijednost varijable LED je 10010010
Loop
End                  'kraj petlje
```



***Samostalna vježba 2: Napisati program koji pali diode izvana prema unutra po slijedećoj tablici:***

***01111110  
10111101  
11011011  
11100111  
11011011  
10111101  
01111110***

***Samostalna vježba 3: Napisati program koji pali diode izvana prema unutra po slijedećoj tablici:***

***10000000  
01000000  
00100000  
00010000  
00001000  
00000100  
00000010  
00000001  
00000010  
00000100  
00001000  
00010000  
00100000  
01000000  
10000000***

## VJEŽBA 3 – ULAZI I IZLAZI IZ MIKROKONTROLERA

Svrha ove vježbe je upoznavanje načina priključenja digitalnih ulaza na kontroler. Prema stanju digitalnih ulaza, kontroler izvršava zadani program. U našem primjeru digitalne ulaze ćemo simulirati tipkama TP1, TP1, TP3, a izlaze porta 1 mikrokontrolera ćemo spojiti na LED diode od D1 do D8.

Načinit ćemo program koji će neprestano motriti stanje uključenosti tipkala TP1 i TP2. Ukoliko je tipkalo TP1 uključeno onda će se LED diode 'kretati' u jednom smjeru, a ukoliko je pritisnuto tipkalo TP2 tada će se promijeniti smjer 'kretanja' LED dioda. Dakle vidljivo je da program u mikrokontroleru reagira na komande iz vanjskog svijeta.

### A. Program

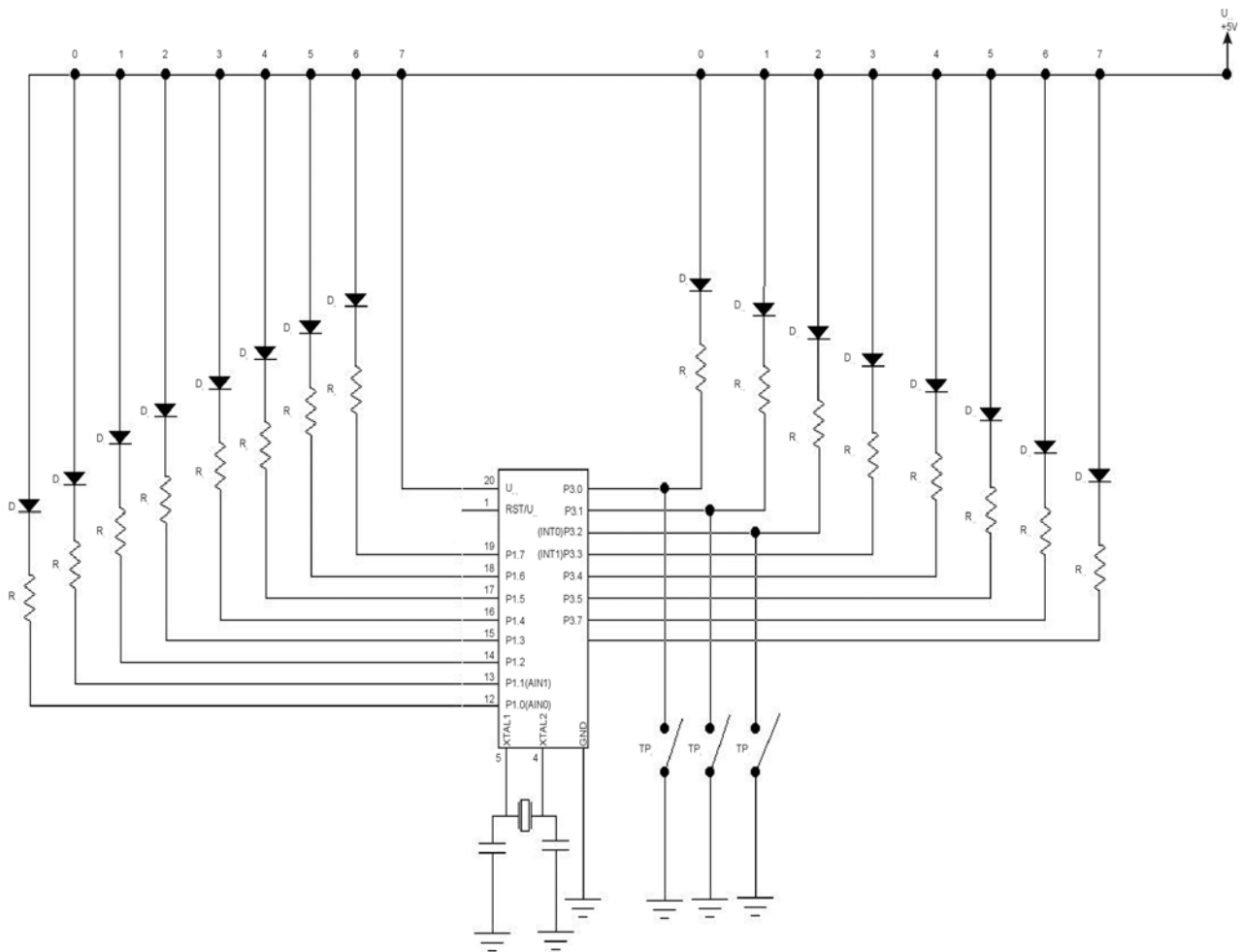
Na slici 8 je prilazana električna shema sklopa kojeg smo pripremili na testnoj pločici spajajući odgovarajuće kratkospojnike.

U programu se upravlja paljenjem i gašenjem LED dioda na portu 1. Uključenjem programa sve LED diode na portu 1 su upaljene osim LED diode D1 ( lijevo 0 ). Ukoliko pritisnemo tipku T2 i držimo je ( za vrijeme pritisnutosti tipke aktivira se dioda D11 - desno 2 ) program pomiče ugašenu LED diodu svaku sekundu za jedno mjesto udesno. Ukoliko pustimo tipku T2, program staje sa izvršenjem. Ako sada stisnemo tipku T1 program pomiče ugašenu LED diodu ulijevo.

```
'primjer trčućeg svjetla ulijevo i udesno
Dim Led As Byte , Gore As Bit , Dolje As Bit
Led = 1          ' Led = 00000001 binarno
P1 = Led        ' P1 = 00000001binarno-gore sve diode na P1 osim D1
Do
  Desno = P3.2  'tipka t2 je komanda za desno
  Lijevo = P3.3 'tipka t1 je komanda za lijevo
  If Desno = 0 Then
    Rotate Led , Left 'ako stisnes tipku t2
    'tada zarotiraj bitove varijable Led ulijevo
    'dakle Led=00000010 – sve su upaljene 'osim D2
    ' kao da se ugašena dioda makla udesno
  P1 = Led 'to prikazi
  Wait 'čekaj 1 sekundu
  If Led = 0 Then 'došao si do kraja Led=00000000 ?? sve
    ' diode na P1 svijetle
    Wait 1
    Led = 1 'Postavi Led na 1 (00000001 – opet upali D1)
    P1 = Led 'to prikazi
    Wait 1
  End If
End If
If Lijevo = 0 Then
  Rotate Led , Right 'ako stisnes tipku t1
  'tada zarotiraj bitove varijable Led udesno
  ' Led = 00000000
  P1 = Led 'to prikazi
  Wait 1
  If Led = 0 Then 'dosao si do kraja Led = 00000000
    Wait 1
    Led = 128 'Postavi Led na 128 (10000000) – sve
```

*P1 = Led*  
*Wait 1*  
*End If*  
*End If*  
*Loop*

*' upali osim D8 (lijevo 7)*  
*'to prikazi*

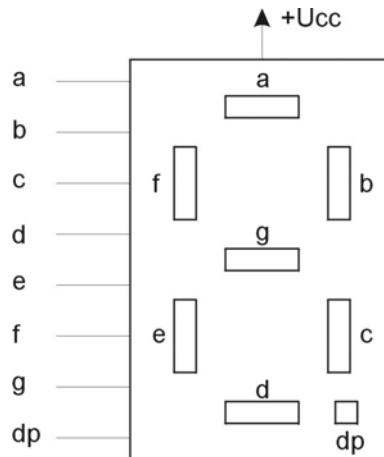


Slika 8: Kontroler sa LED diodama i tipkalima

**Samostalna vježba 4:** *Napravite program koji će koristiti dvije tipke TP2 i TP3 i LED diode porta 1. Neka sve LED diode na P1 budu ugašene kod inicijalizacije programa. Kod pritiska tipke TP3 neka se sukcesivno pale LED diode od D1 do D8 svakih jednu sekundu dok se sve ne upale. Pritiskom tipke TP3 LED diode se sukcesivno gase obrnutim redosljedom.*

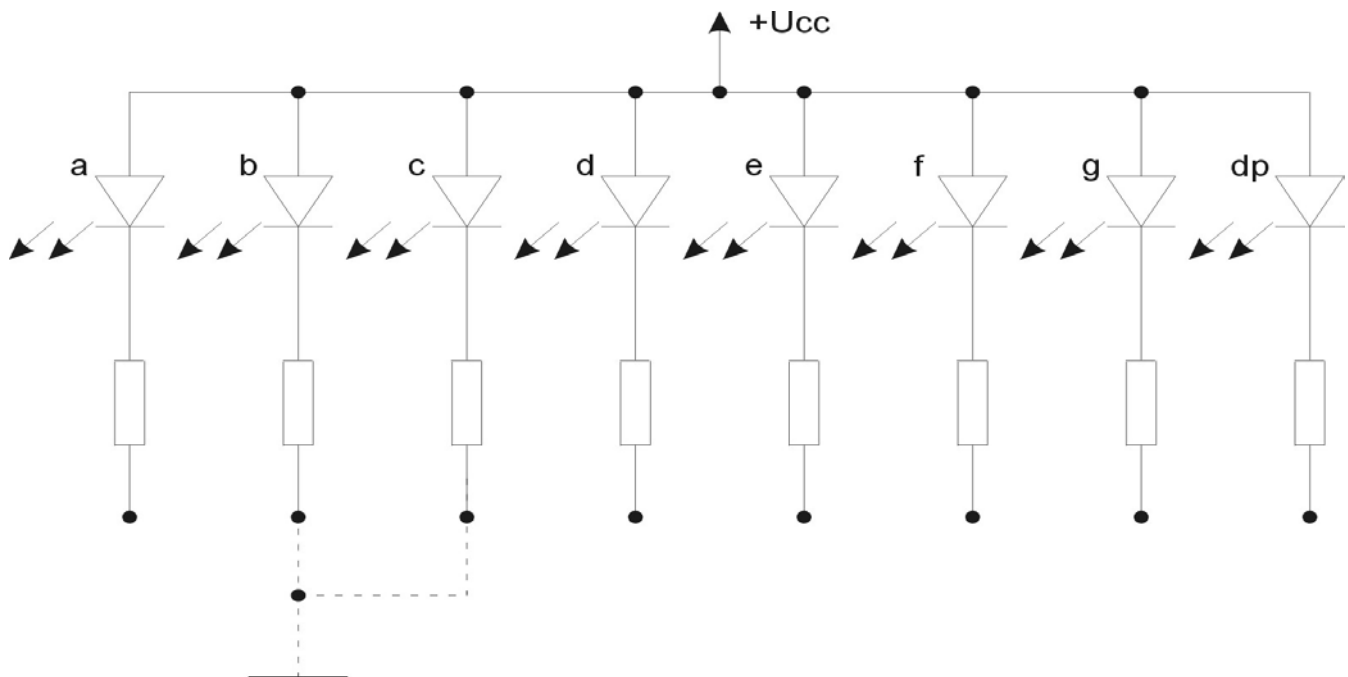
## VJEŽBA 4 – PRIKAZ PODATAKA NA LED DISPLEJU

Zadatak ove vježbe je naučiti na koji način je moguće prikazati podatke na LED displejima koji su upravljani mikrokontrolerom. Koristit ćemo sedam segmentne LED displeje prikazane na slici 9.



Slika 9: 7 segmentni displej

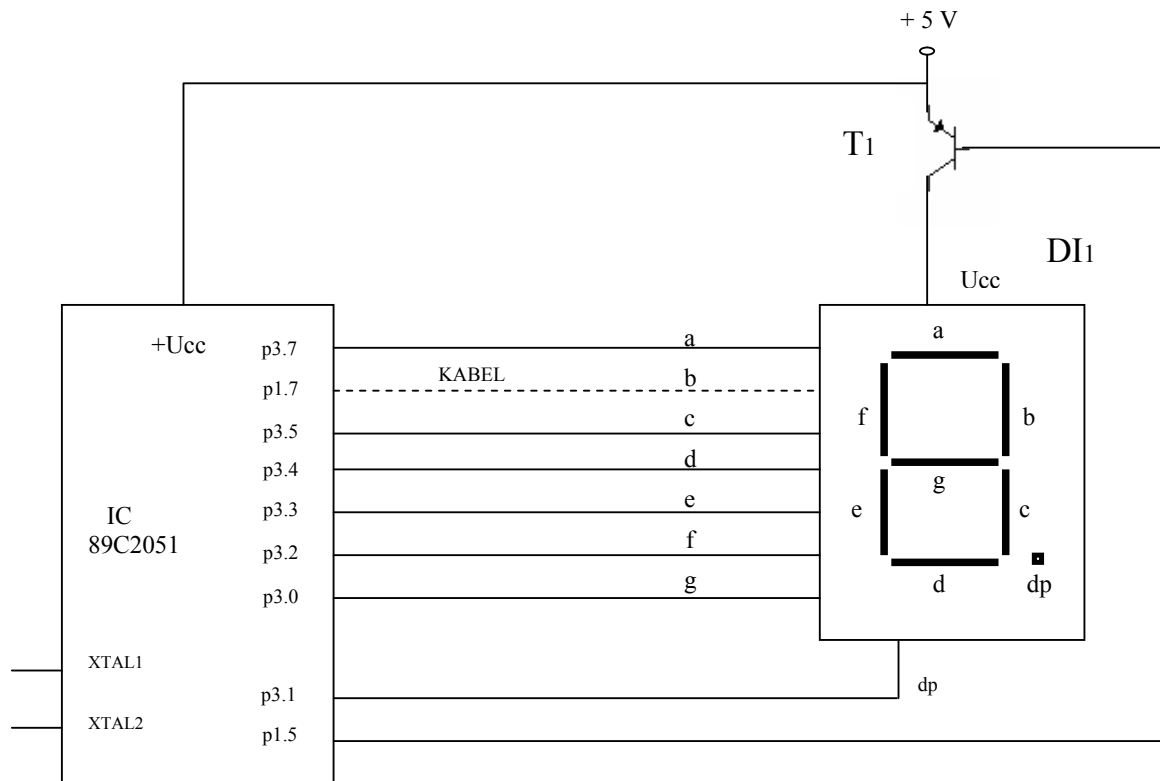
Sedam segmentni displej možemo zamjeniti skupom od osam led dioda kao što je prikazano na slici 10. Mikrokontroler upravlja svakim segmentom LED displeja posebno. Kad su svi priključci LED displeja ( a, b, c, d, e, f, g, dp ) na potencijalu  $U_{cc}$ , niti jedan segment LED displeja ne svijetli. Kada pojedini priključak spojimo sa masom, tada LED dioda segmenta počme svijetliti. Na slici 10 LED displej će pokazati broj 1 jer su priključci segmenta b i c spojeni sa masom. Na takav način se može prikazati bilo koja znamenka od 0 do 9.



Slika 10: Spajanje 7 segmentnog displeja za prikaz broja 1

## Program za uključivanje segmenata displeja DI1

Na slici 11 je prikazan spoj kontrolera sa sedam segmentnim displejem. Program sukcesivno svake sekunde uključuje segmente displeja DI1 (prvi s lijeva). Redosljed uključivanja segmenata je a, b, c, d, e, f, dp, g. Pin P1.5 cijelo vrijeme mora biti u logičkoj 0 (to se definira na početku programa) da bi LED displej preko tranzistora T1 dobio napajanje Ucc.



Slika 11: Spoj mikrokontrolera sa 7 segmentnim LED displejom

' Program za uključivanje segmenata displeja DI1, svaki segment je uključen 1 sekundu

Do

P1.5 = 0                   ' ukljuci napajanje DI1

P3.7 = 0                   ' ukljuci segment a

Wait 1                    ' cekaj 1 sekund1

P3.7 = 1                   ' iskljuci segment a

P1.7 = 0                   uključujuci segment b

Wait 1

P1.7 = 1                   ' iskljuci segment b

P3.5 = 0                   ' ukljuci segment c (P3.5=0)

Wait 1

P3.5 = 1                   ' iskljuci segment c

P3.4 = 0                   ' ukljuci segment d (P3.4=0)

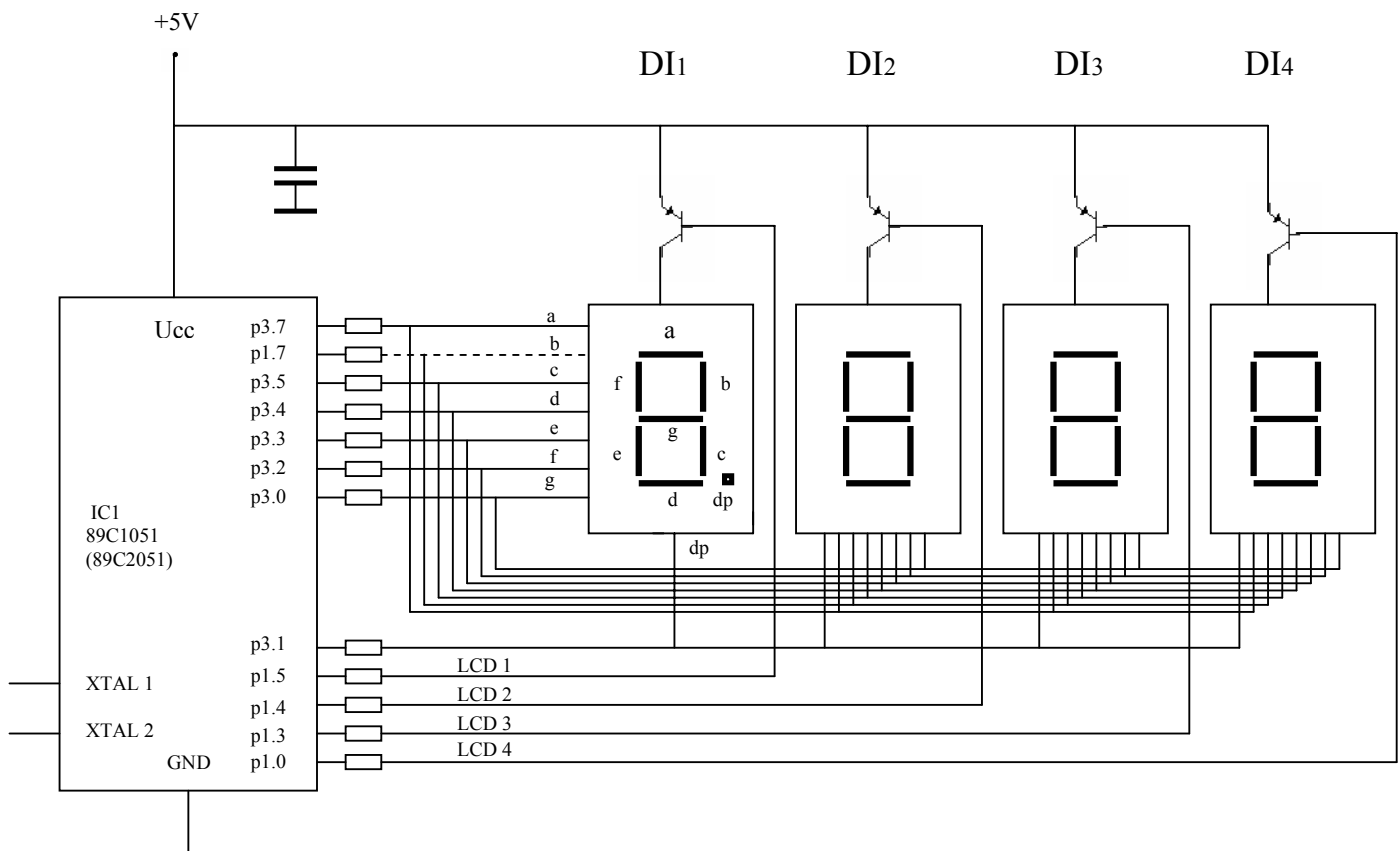
```

Wait 1
P3.4 = 1      'isključi segment d
P3 = &B11110111 'uključi segment e (P3.3=0)
Wait 1
P3 = &B11111011 'uključi segment f i isključi sve ostalo
Wait 1
P3 = &B11111101 'uključi segment dp i isključi sve ostalo
Wait 1
P3 = &B11111110 'uključi segment g i isključi sve ostalo
Wait 1
P3 = &B11111111 'ugasi sve segmente
Loop
End

```

## Program za multipleksirani LCD

Na slici 12 je prikazan spoj mikrokontrolera sa 4 LED displeja. Program ispisuje na LED displejima 'PULA' na način da mikrokontroler prvo upiše slovo 'P' na displeju DI1, pa nakon 5  $\mu$ sec slovo 'U' na displeju DI2, pa onda nakon 5  $\mu$ sec slovo 'L' na displeju DI3 i konačno nakon 5  $\mu$ sec slovo 'A' na displeju DI4. Program se neprekidno vrti u petlji i zbog tromosti oka čini nam se kao da su sva četiri slova istovremeno prikazana na displejima. Cijeli port 3 i P1.7 su zaduženi za paljenje segmenata na displejima, dok P1.0, P1.3, P1.4 i P1.5 su zaduženi za uključivanje napajanja za LED displeje.





Slika 12: Spoj mikrokontrolera sa LED displejima

Programski kod slijedi:

```

Dim P As Const &B00110010          ' slovo P
Dim U As Const &B10000011        ' slovo U
Dim L As Const &B11100011        ' slovo L
Dim A As Const &B00010010        ' slovo A

Do
  P1 = &B01011111                ' aktiviraj P1.5(DI1) i P1.7 (segment b)
  P3 = P                          ' ukljuci sve ostale segmente
  Waitms 5                        ' cekaj 5 mikrosekundi

  P1 = &B01101111                ' aktiviraj P1.4(DI2) i P1.7 (segment b)
  P3 = U                          ' ukljuci ostale segmente
  Waitms 5

  P1 = &B11110111                ' aktiviraj P1.3 (DI3) bez P1.7 (sada ne treba segment b)
  P3 = L                          ' ukljuci ostale segmente
  Waitms 5

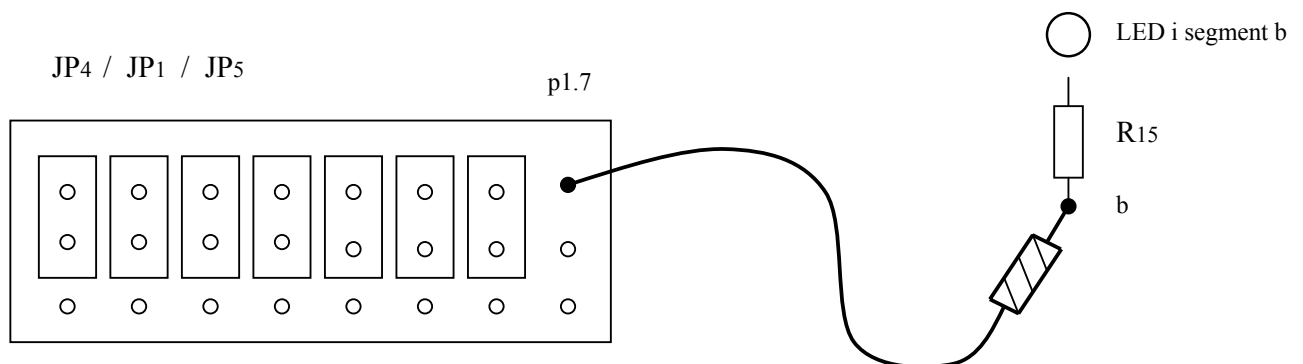
  P1 = &B01111110                ' aktiviraj P1.0 (DI4) i P1.7 (segment b)
  P3 = A                          ' ukljuci ostale segmente
  Waitms 5

Loop

```

**Samostalna vježba 5:** Potrebno je načiniti program koji pomiče natpis PULA s desna na lijevo. Inicijalno displeji ne prikazuju ništa da bi natpis PULA slovo po slovo ulazio na desni displej, potpuno se prikazao i nestao na lijevom displeju.

*Napomena:* S obzirom da je testna BASCOM-LT pločica prvenstveno građena za testiranje 40 pinskih mikrokontrolera, dok kod testiranja 20 pinskih mikrokontrolera imat ćemo neke dodatne probleme. Jedan od tih problema je nedostatak pina P3.6. Zbog toga za pogon segment 'b' moramo koristiti pin P1.7, te taj pin spojiti kabelom sa segmentom 'b' (vidi sliku 13). Obratite se za pomoć demonstratoru.



Slika 13: Prikaz spajanja kratkospojnika na testnoj pločici

## VJEŽBA 5 – timeri i interrupti

U ovoj vježbi ćemo naučiti koristiti timer i interrupte mikrokontrolera. Timeri su u stvari 16 bitni brojači koji broje impulse koji dolaze na njihov ulaz. Naš mikrokontroler ima dva timera: Timer1 i Timer2. sa pripadajućim ulazima ( vidi sliku 1 ). Postoje dva načina rada tajmera:

Prvi način rada je spoj tajmera na način da ulazni impulsi u tajmer čini takt mikrokontrolera podijeljen sa 12 (1 MHz). Brojač u tajmeru se može podesiti na određenu vrijednost. Kada tajmer izbroji broj impulsa koji je jednak toj namještenoj vrijednosti, tajmer se resetira na nulu i pošalje prekid (interrupt) jedinici koja se zove interrupt kontroler, a koja CPU šalje zahtjev za prekid izvršenja glavnog programa. CPU pospremi sve varijable glavnog programa, obavi potprogram (servisnu rutinu ) koja je pridružena tom tajmeru i ponovo se vraća na izvršenje glavnog programa.

Drugi način rada timera je da timer jednostavno broji impulse koji se dovode na vanjski priključak mikrokontrolera. Kad dosegne najveću moguću vrijednost  $2^{16}$  stanje se signalizira na jednom bitu, što je signal za poduzimanje daljnjih akcija .

U našim ćemo vježbama koristiti prvi način rada timera. Kratkospojnici na vježbi su podešeni kao kod vježbe 4.

Program počinje definiranjem varijabli koje ćemo koristiti. Naredba Config definira način rada Timera. Ona govori da radimo sa Timerom0, da koristimo interni interrupt (Gate) i da 16 bitni timer koristimo kao 8 bitni – ( Mode=2 ). Slijede naredbe kojim se definira ime potprograma koji starta u trenutku kada Timer0 da interrupt. Nadalje LOAD naredbom definiramo u kojim intervalima će se dešavati interrupti ( svaki 250  $\mu$ sec ). Nakon inicijalizacijskog dijela programa dolazi glavni program, prekidna rutina koja broji interrupte i potprogram za prikaz podataka.

Glavni program se stalno vrti u beskonačnoj petlji. Njegova zadaća je da neprestano ispituje stanje varijable Prikaz. Ukoliko je Prikaz = 0 ništa se ne dešava, a ukoliko je Prikaz=1 (svakih 5 msec) glavni ga program postavlja u 0 te skače u potprogram 'Prikaz' za prikaz znamenki.

Prekidna rutina Timer\_0\_int (koja se aktivira svakih 250  $\mu$ sec) ima zadatak da generira impuls svake sekunde. Točnost 1 sekundnog impulsa ovisi o točnosti kvarca kojeg koristimo. Naime kvarc oscilatora je 12 MHz. Ulazna frekvencija impulsa u tajmer iznosi dvanaestinu frekvencije (1 MHz ). Budući da smo naredbom Load Timer0 postavili na iznos 250  $\mu$ sec, prekid će se desiti svakih 250  $\mu$ sec, a to će pokrenuti prekidnu rutinu. Moramo dakle izbrojiti 4000 prekida da bi smo dobili 1 sec. ( 4000 x 250  $\mu$ sec ). U prekidnoj rutini postoje varijable Clock i Clock1. Obje su varijable tipa Byte pa najveća vrijednost u svakoj od njih može biti 255. Prva neka broji do 20, a druga do 200 (20 x 200 = 4000). Kad varijabla Clock dobroji do 20 setira se Prikaz=1 i inkrementira se varijabla Clock1 (svakih 5 msec). Kad varijabla Clock1 odbroji do 200, prošla je točno jedna sekunda, inkrementira se varijabla sekunde a varijabla Clock1 se resetira na nulu

Rutina za prikaz podataka na LED displeju omogućava da se znamenka od 0 do 9 koja je smještena u varijabli Pomocna\_v prikaže na 7-segmentnom displeju.

### A. Program za prikaz sekunda na displeju DI4

*' Prikaz s LED displejom prikazuje na led 4 sekunde i broji od 0 do 9*

*Dim Clock As Byte  
Dim Clock1 As Byte  
Dim Sekunde As Byte*

*Dim Pomocna\_v As Byte ,  
Dim Prikaz As Bit*

*Config Timer0 = Timer , Gate = Internal , Mode = 2  
' konfiguracija Timera2  
'nema vanjskog prekida  
'8 biotova auto reload*

*On Timer0 Timer\_0\_int ' Kad Timer0 ima intreupt skoci na rutinu Timer\_o\_int*

*Load Timer0 , 250 ' ovim definiras 250 us izmedju interapta  
Priority Set Timer0 'digni prioritet koji smo odredili timeru 0  
Enable Interrupts  
Enable Timer0 'Omoguæi rad Timera0  
Start Timer0 'Pokreni timer0*

*' \_\_\_\_\_ Glavni program \_\_\_\_\_*

*Clock = 0  
Clock1 = 0  
Sekunde = 0  
Do  
If Prikaz = 1 Then 'svakih 20 prekida 5 ms Prikaz = 1 inace 0  
Prikaz = 0  
P1.0 = 1 ' ukljuci DI4  
Pomocna\_v = Sekunde  
Gosub Prikaz ' osvijetli odgovarajuæe segmente  
P1.0 = 0 'iskljuci DI4  
End If  
Loop  
End*

*' \_\_\_\_\_ Potprogrami \_\_\_\_\_*

*'Prekidna rutina koji broji interne interupte  
Timer\_0\_int:  
Incr Clock 'Povecaj variablu Clock za jedan  
If Clock > 19 Then 'kad je Clock veci od 19  
Clock = 0  
Prikaz = 1 ' svakih 19 taktova prikaz=1  
Incr Clock1 'Povecaj Clock1  
If Clock1 > 199 Then 'ako je Clock1 veci od 199  
Clock1 = 0 ' resetiraj ga na nulu  
Incr Sekunde 'uvecaj varijablu sekunde  
If Sekunde > 9 Then 'ako sekunde predju 9 resetiraj ih  
Sekunde = 0  
End If  
End If  
End If  
Return*

*' \_\_\_\_\_  
'rutina za prikaz na LED displeju*

*Prikaz:*

```
If Pomocna_v = 0 Then
  P1 = &B01111111
  P3 = 3
End If
If Pomocna_v = 1 Then
  P1 = &B01111111
  P3 = 159
End If
If Pomocna_v = 2 Then
  P1 = &B01111111
  P3 = 38
End If
If Pomocna_v = 3 Then
  P1 = &B01111111
  P3 = 14
End If
If Pomocna_v = 4 Then
  P1 = &B01111111
  P3 = 154
End If
If Pomocna_v = 5 Then
  P1 = &B11111111
  P3 = 74
End If
If Pomocna_v = 6 Then
  P1 = &B11111111
  P3 = 66
End If
If Pomocna_v = 7 Then
  P1 = &B01111111
  P3 = 31
End If
If Pomocna_v = 8 Then
  P1 = &B01111111
  P3 = 2
End If
If Pomocna_v = 9 Then
  P1 = &B01111111
  P3 = 10
End If
Return
```

Potprogram Prikaz ( aktivni su segmenti gdje je logička 0 )

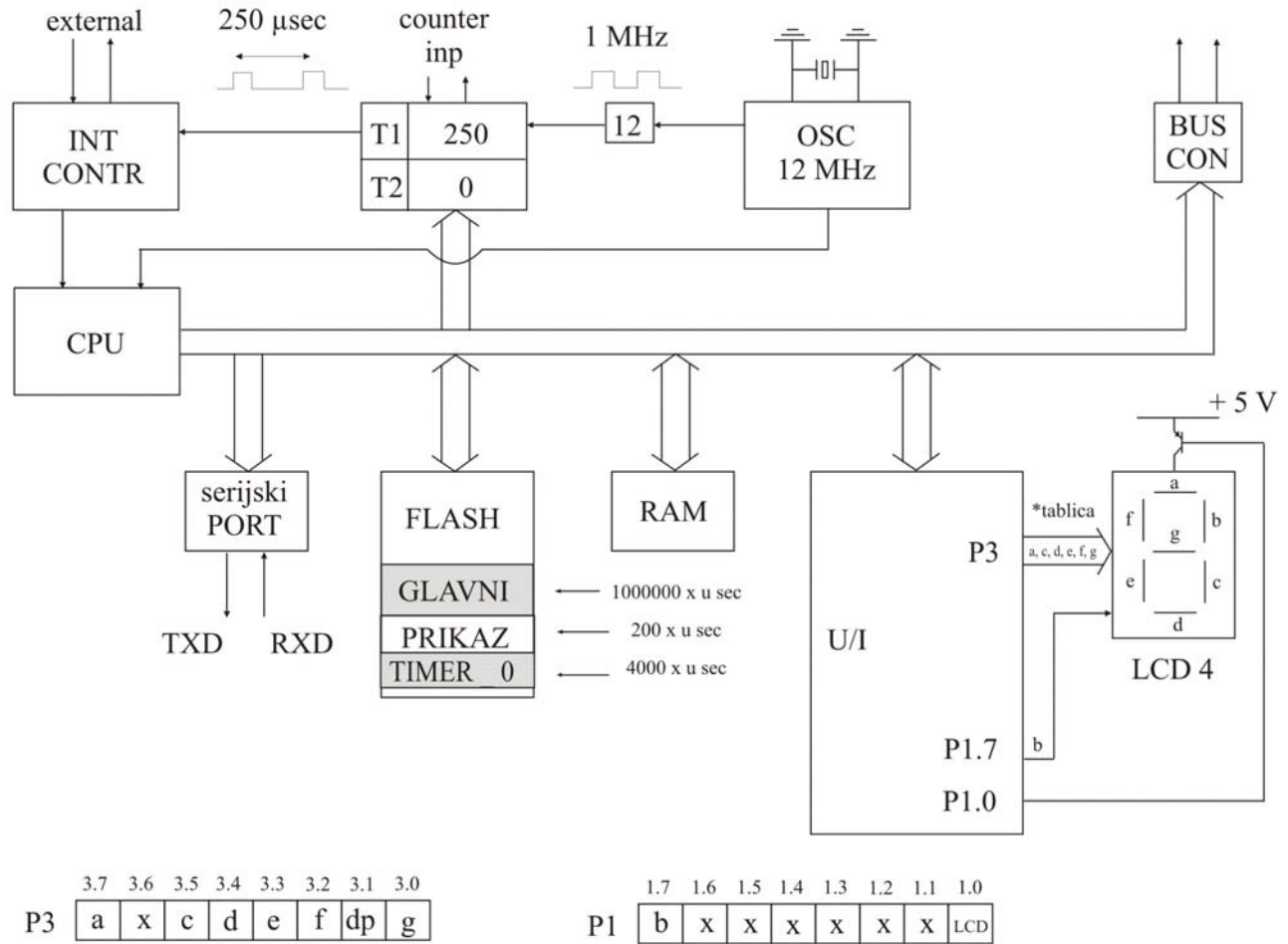
Znam	3.7-a	3.6b*?	3.5-c	3.4-d	3.3-e	3.2-f	3.1-dp	3.0-g	Dec
0	0	0 neakt	0	0	0	0	1	1	3
1	1	0 neakt	0	1	1	1	1	1	159
2	0	0 neakt	1	0	0	1	1	0	38
3	0	0 neakt	0	0	1	1	1	0	14
4	1	0 neakt	0	1	1	0	1	0	154
5	0	1 neakt	0	0	1	0	1	0	74
6	0	1 neakt	0	0	0	0	1	0	66
7	0	0 neakt	0	1	1	1	1	1	31
8	0	0 neakt	0	0	0	0	1	0	2
9	0	0 neakt	0	0	1	0	1	0	10

Znam	1.7-b	1.6	1.5	1.4	1.3	1.2	1.1	1.0
0	0	1	1	1	1	1	1	1
1	0	1	1	1	1	1	1	1
2	0	1	1	1	1	1	1	1
3	0	1	1	1	1	1	1	1
4	0	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1	1
6	1	1	1	1	1	1	1	1
7	0	1	1	1	1	1	1	1
8	0	1	1	1	1	1	1	1
9	0	1	1	1	1	1	1	1

Tablica 2: Stanje izlaznih pinova mikrokontrolera za prikaz pojedinih znamenki

**Samostalna vježba 6: Napraviti program koji prikazuje desetinke sekunda na displeju D4. Nakon jedne sekunde program se zaustavlja ( primjeniti naredbu STOP ).**

## VJEŽBA 5



\* 0 je aktivna i pali segment

GLAVNI PROGRAM + prikaz (potprogram) - svakih 5 msec ponovno upisuje znamenku na LCD (može biti stara ili nova)

PREKIDNA RUTINA TIMER\_0\_INT (provjerava da li su se sekunde promijenile)

- Mjeri vrijeme (broji INTERRUPT)
- Svakih 5 msec (20 interrupta) omogućuje glav. programu provjeru "SEK"
- Izračunava "sekunde" (4000 int)
- Kad sekunde > 9 vraća na 0

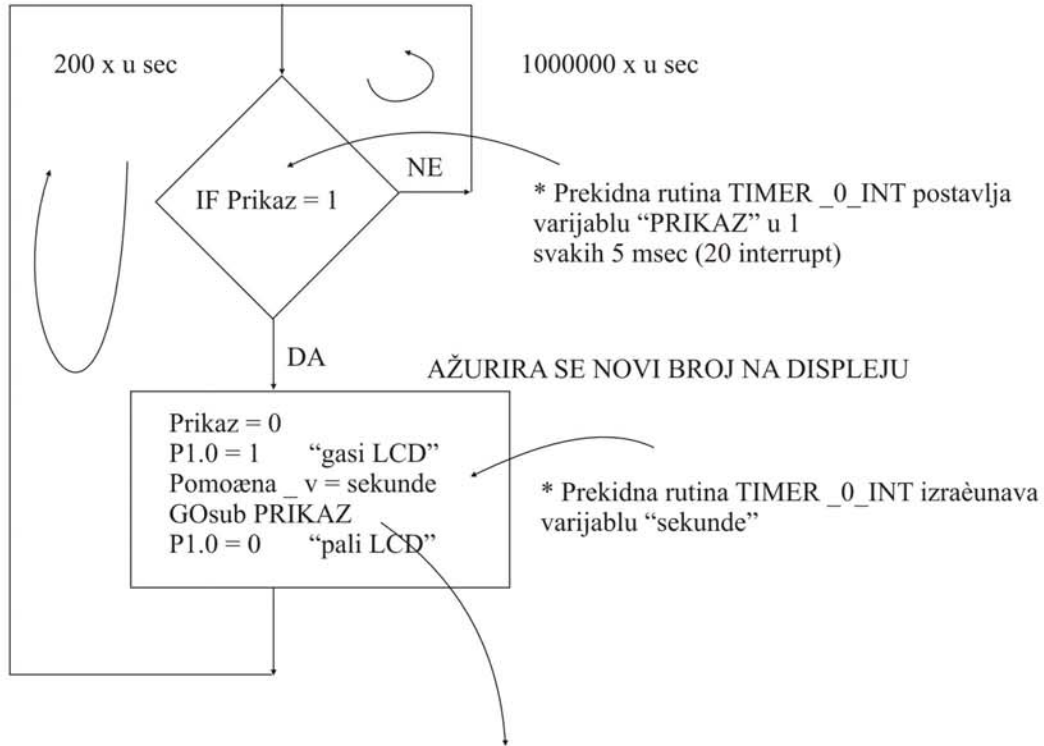


## GLAVNI PROGRAM

Varijabla prikaz = 1 200 x u sec

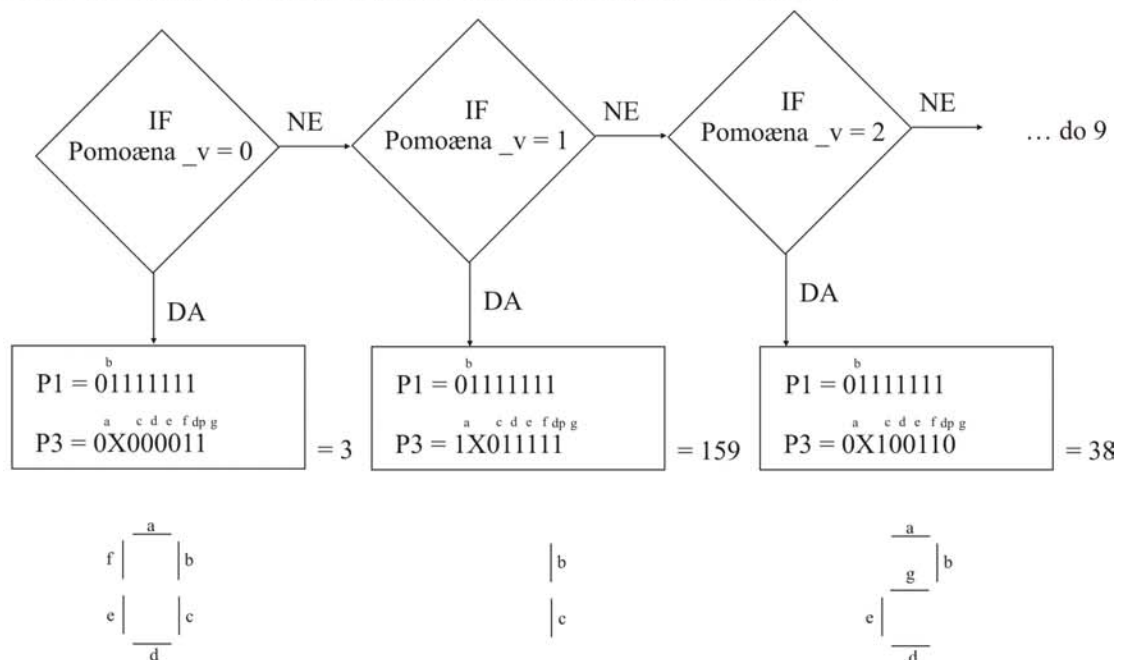
- Glavni program svakih 20 prekida ( $20 \times 250 \mu \text{ sec} = 5 \text{ msec}$ ) prikaže na LCD-u broj koji se nalazi u varijabli SEKUNDE

Dim Prikaz As Bit.



## RUTINA ZA PRIKAZ \*tablica

Pali segmente LED displeja i prikazuje znamenku koja piše u varijabli "Pomoæna - v"



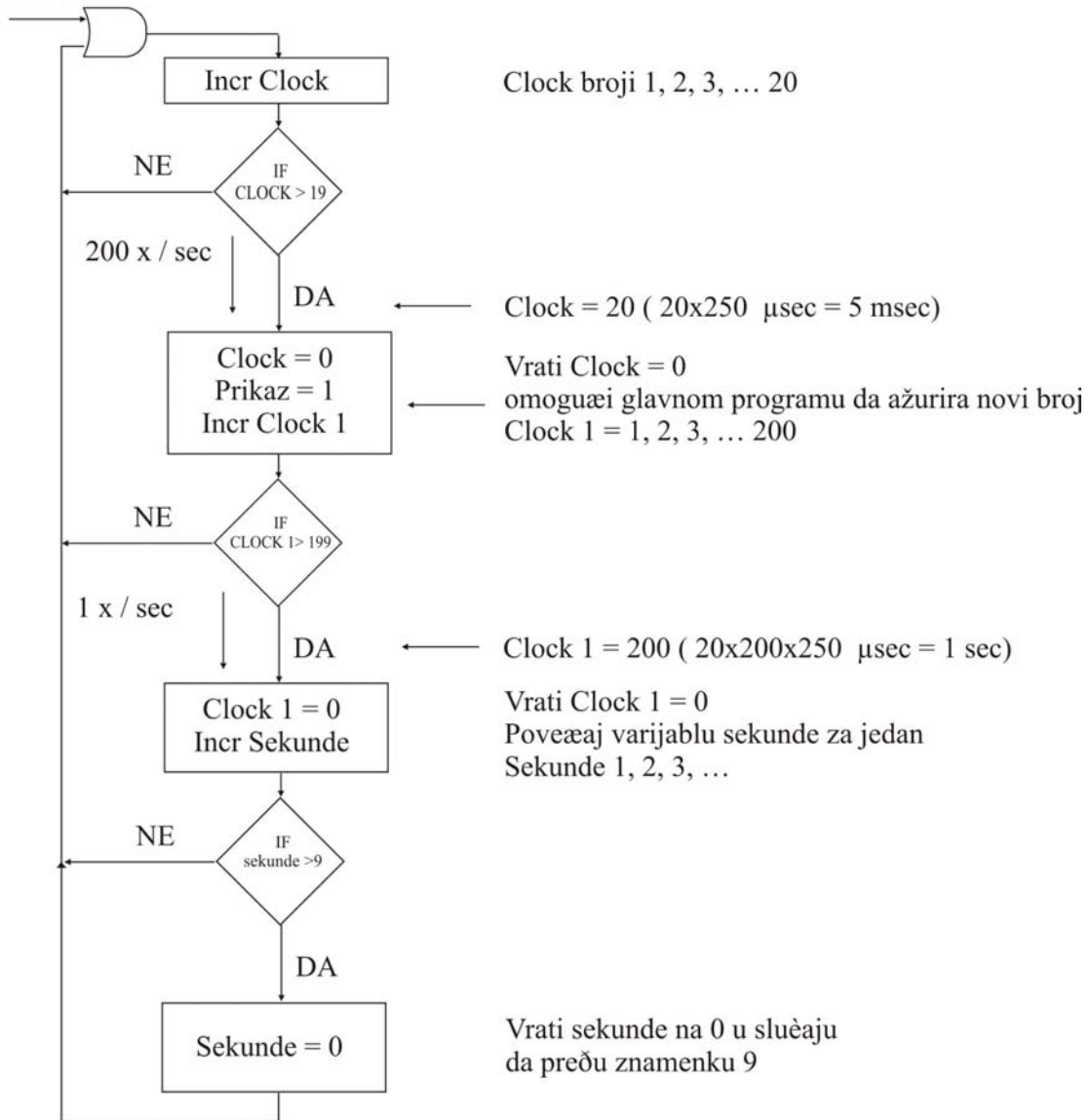
## PREKIDNA RUTINA TIMER \_0\_INT (4000 PUTA U SEC)

Aktivira se svakih 250  $\mu$ sec (na svaki int)

Clock, Clock 1, Sekunde As Byte

250  $\mu$ sec

(4000 x u sec)



## VJEZBA 6 – MULTIPLEKSIRANI PRIKAZ VREMENA

Ova vježba je vrlo slična prethodnoj, razlika je u tome što se vrijeme prikazuje multipleksirano na sva četiri displeja. Na displeju DI1 prikazane su deseci sekunda, na displeju DI2 sekunde, na displeju DI3 desetinke, a na displeju DI4 stotine sekunda. Kad se mikrokontroler uključi displeji počinju prikazivati vrijeme. U programu nije predviđeno zaustavljanje vremena.

Kratkospojnici na ploči isto kao u prethodnoj vježbi.

### A. Program za multipleksirani prikaz vremena

```
' Primjer multiplexsiranog prikaza na LCD
' Ovo je program koji prikazuje na LED1 - stotine LED2 - desetinke LED3 - sekunde LED4 desetke sekunda
' Kratkospojnike JP10 i JP11 isključiti, uključiti JP12 JP13 JP14 JP15 JP23
' Kratko spojiti donje krajeve otpornika R8 i R15
Dim Clock As Byte
Dim Clock1 As Byte
Dim Mux As Byte , Sekunde As Byte
Dim Stotine As Byte
Dim Pomoc As Byte
Dim Pomocna_v As Byte ,
Dim Prikaz As Bit , Izracun As Bit
Dim Lcd1 As Byte
Dim Lcd2 As Byte
Dim Lcd3 As Byte
Dim Lcd4 As Byte

Config Timer0 = Timer , Gate = Internal , Mode = 2
                                ' konfiguracija Timera2
                                'nema vanjskog prekida
                                '8 biotova auto reload

On Timer0 Timer_0_int           ' Kad Timer0 ima intreupt skoci na rutinu 'Timer_o_int

Load Timer0 , 250               ' ovim definiras 250 us izmedju interapta
Priority Set                     'podigni prioritet koji smo odredili timeru 0
Enable Interrupts
Enable Timer0                   'Omoguæi rad Timera0
Start Timer0                    'Pokreni timer0

' _____ Glavni program _____
Clock = 0
Clock1 = 0
Sekunde = 0

' Rutina za izracun

Do
  If Izracun = 1 Then           ' svake stotine
    Izracun = 0               'Rutina za izracun stotinki, desetinki, sekunda jedinica
```

*'Prekidna rutina broji stotinke i za svakih 100 stotinki broji sekunde*

```
Lcd3 = Stotinke / 10           '59 stotinki/10=5 => Lcd3 (nema ostatka)
Pomoc = Lcd3 * 10             '5*10=50 => Pomoc
Lcd4 = Stotinke - Pomoc       '59-50=9 => Lcd4
Lcd1 = Sekunde / 10           '74 sekund/10=7 => Lcd1
Pomoc = Lcd1 * 10             '7*10=70 => Pomoc
Lcd2 = Sekunde - Pomoc       '74-70=4 => Lcd2
```

End If

If Prikaz = 1 Then

```
Prikaz = 0
P1.3 = 1           'Iskljuci displej 3
P1.0 = 1           'Iskljuci displej 4
P1.4 = 1           'Iskljuci displej 2
P1.5 = 1           'Iskljuci displej 1
```

*'Rutina za prikaz stotinki*

```
If Mux = 0 Then
  Pomocna_v = Lcd4
  Gosub Prikaz           'skoci na potprogram prikaz
  P1.0 = 0               'pali displej 4
End If
```

*' Rutina za prikaz desetinki*

```
If Mux = 1 Then
  Pomocna_v = Lcd3
  Gosub Prikaz           'skoci na potprogram prikaz
  P1.3 = 0               'pali displej 3
End If
```

*'Rutina za prikaz jedinica sekundi*

```
If Mux = 2 Then
  Pomocna_v = Lcd2
  Gosub Prikaz           'skoci na potprogram prikaz
  P1.4 = 0               'pali displej 2
End If
```

*'Rutina za prikaz desetica sekundi*

```
If Mux = 3 Then
  Pomocna_v = Lcd1
  Gosub Prikaz           'skoci na potprogram prikaz
  P1.5 = 0               'pali displej 1
End If
```

End If

Loop

End

*' \_\_\_\_\_ Potprogram i tabele \_\_\_\_\_*

*'Prekidna rutina koji broji interne interupte*

Timer\_0\_int:

```
Incr Clock           'Povecaj variablu Clock za jedan
If Clock > 19 Then   'kad je Clock veci od 19
  Clock = 0
  Incr Mux           'varijabla Mux definira na kojem se displeju vrsi prikaz
  If Mux > 3 Then
```

```

    Mux = 0
    End If
    Prikaz = 1
    Incr Clock1
    If Clock1 > 1 Then
        Clock1 = 0
        Izracun = 1
        Incr Stotinke

        If Stotinke > 99 Then
            Incr Sekunde
            Stotinke = 0
        End If

        If Sekunde > 99 Then
            Sekunde = 0
        End If

    End If
End If
Return

```

```

'svakih 5 ms ( 20 interapta se mjenja displej za prikaz )
' svakih 19 taktova omoguci prikaz
'Povecaj Clock1
'ako je Clock1 veci od 1 znaci da su prosla
' 40 interrupta ( 10 ms ili jedna stotinka sekunde
' resetiraj ga na nulu
'omogucuje se Izracun u glavnom programu svake stotinke
'poveca varijablu stotinke

```

---

'rutina za prikaz na LED displeju

Prikaz:

```

    If Pomocna_v = 0 Then
        P1 = &B01111111
        P3 = 3
    End If
    If Pomocna_v = 1 Then
        P1 = &B01111111
        P3 = 159
    End If
    If Pomocna_v = 2 Then
        P1 = &B01111111
        P3 = 38
    End If
    If Pomocna_v = 3 Then
        P1 = &B01111111
        P3 = 14
    End If
    If Pomocna_v = 4 Then
        P1 = &B01111111
        P3 = 154
    End If
    If Pomocna_v = 5 Then
        P1 = &B11111111
        P3 = 74
    End If
    If Pomocna_v = 6 Then

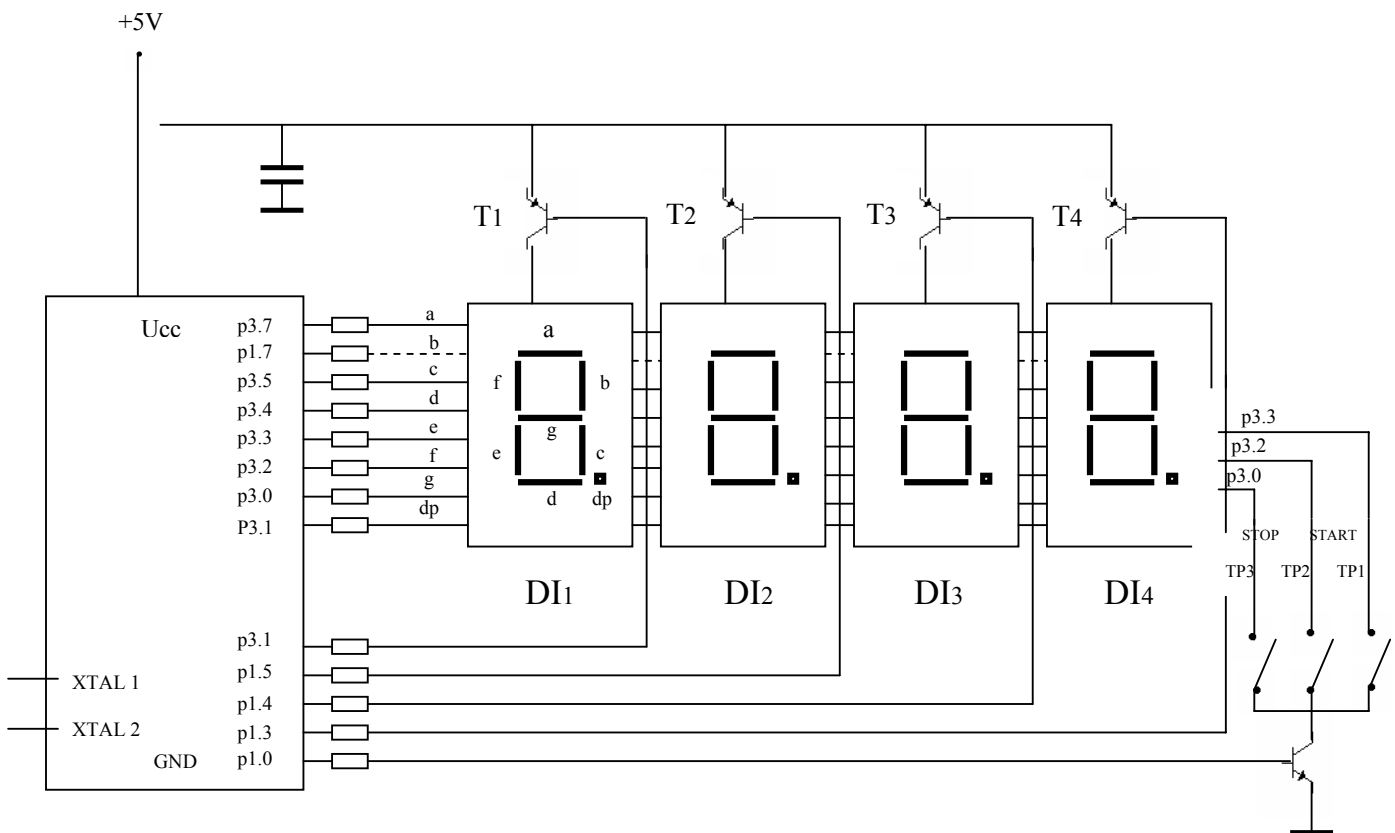
```

```

P1 = &B11111111
P3 = 66
End If
If Pomocna_v = 7 Then
  P1 = &B01111111
  P3 = 31
End If
If Pomocna_v = 8 Then
  P1 = &B01111111
  P3 = 2
End If
If Pomocna_v = 9 Then
  P1 = &B01111111
  P3 = 10
End If
Return

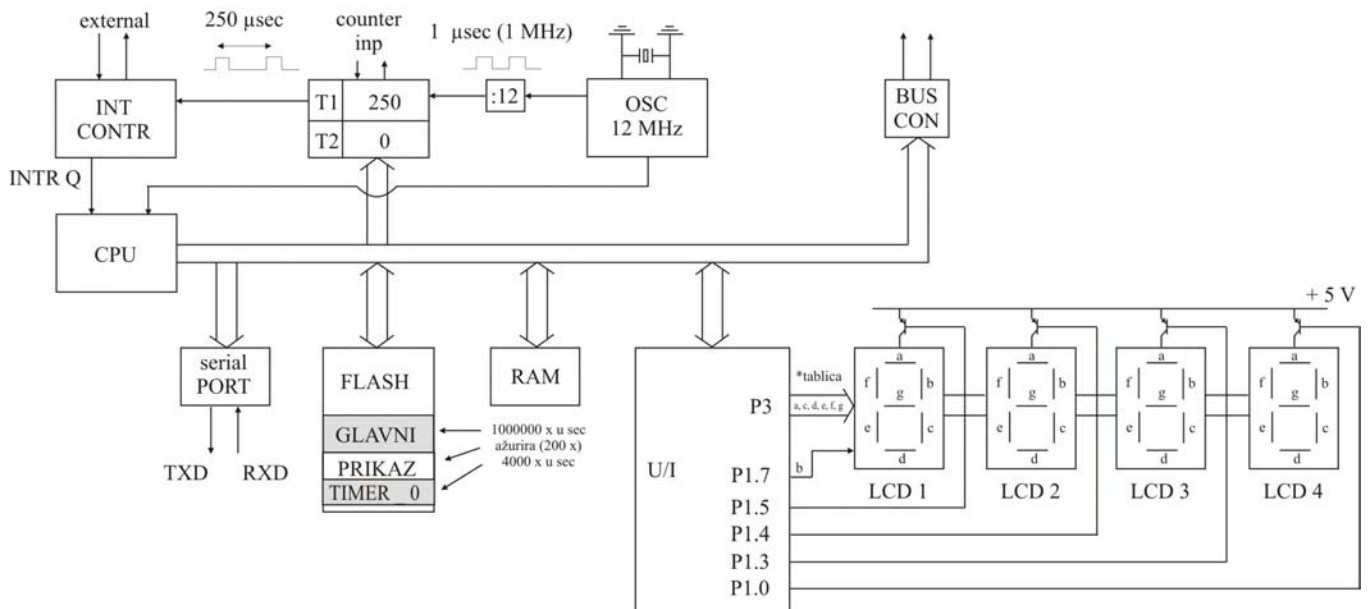
```

**Samostalna vježba 7: Namjestiti testnu pločicu po shemi na slici 14 i modificirati dati program na način da tipka TP1 starta brojač, a tipka TP2 zaustavi brojač.**



Slika 14: Shema vježbe za samostalnu vježbu br. 7

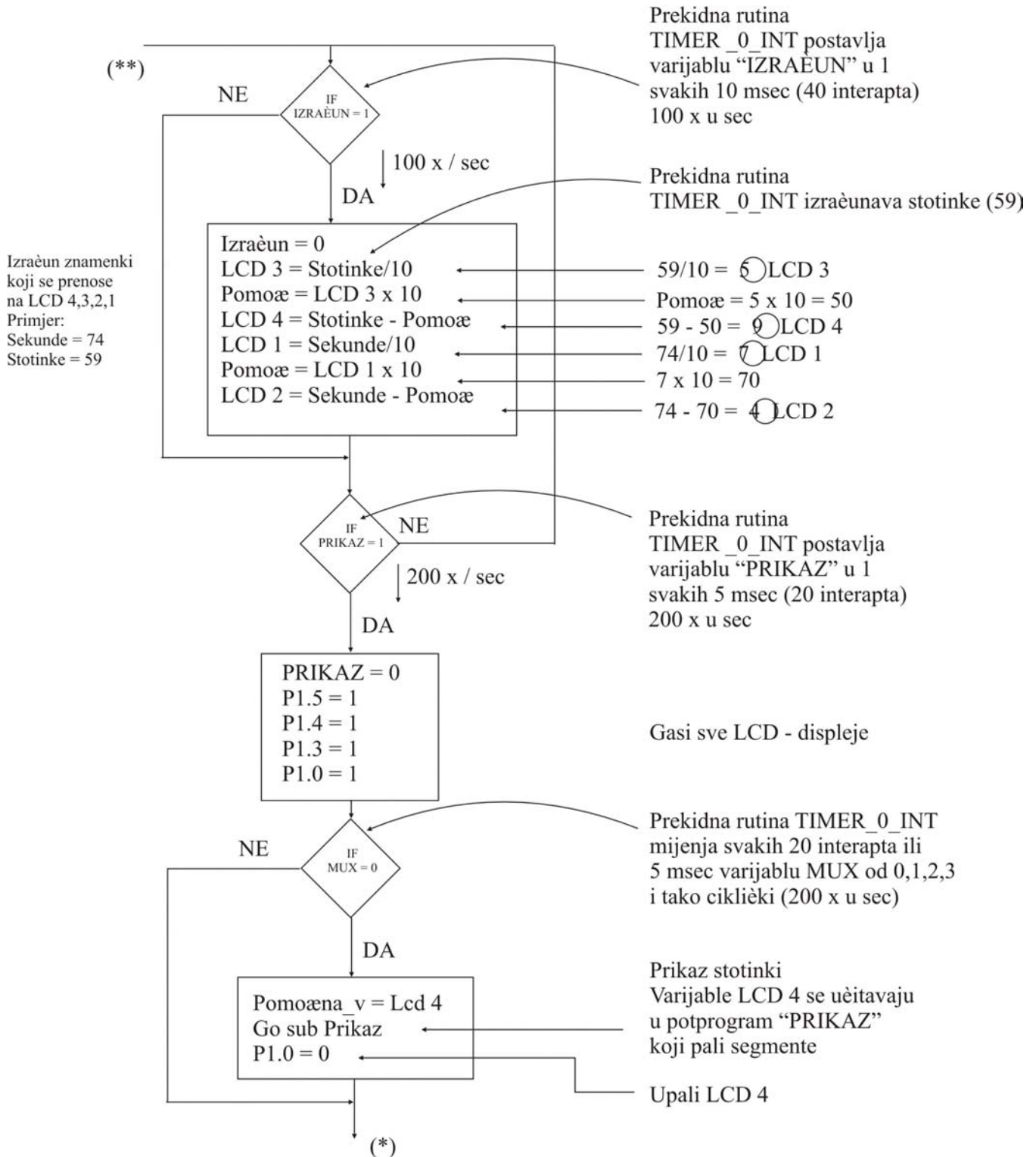
## VJEŽBA 6



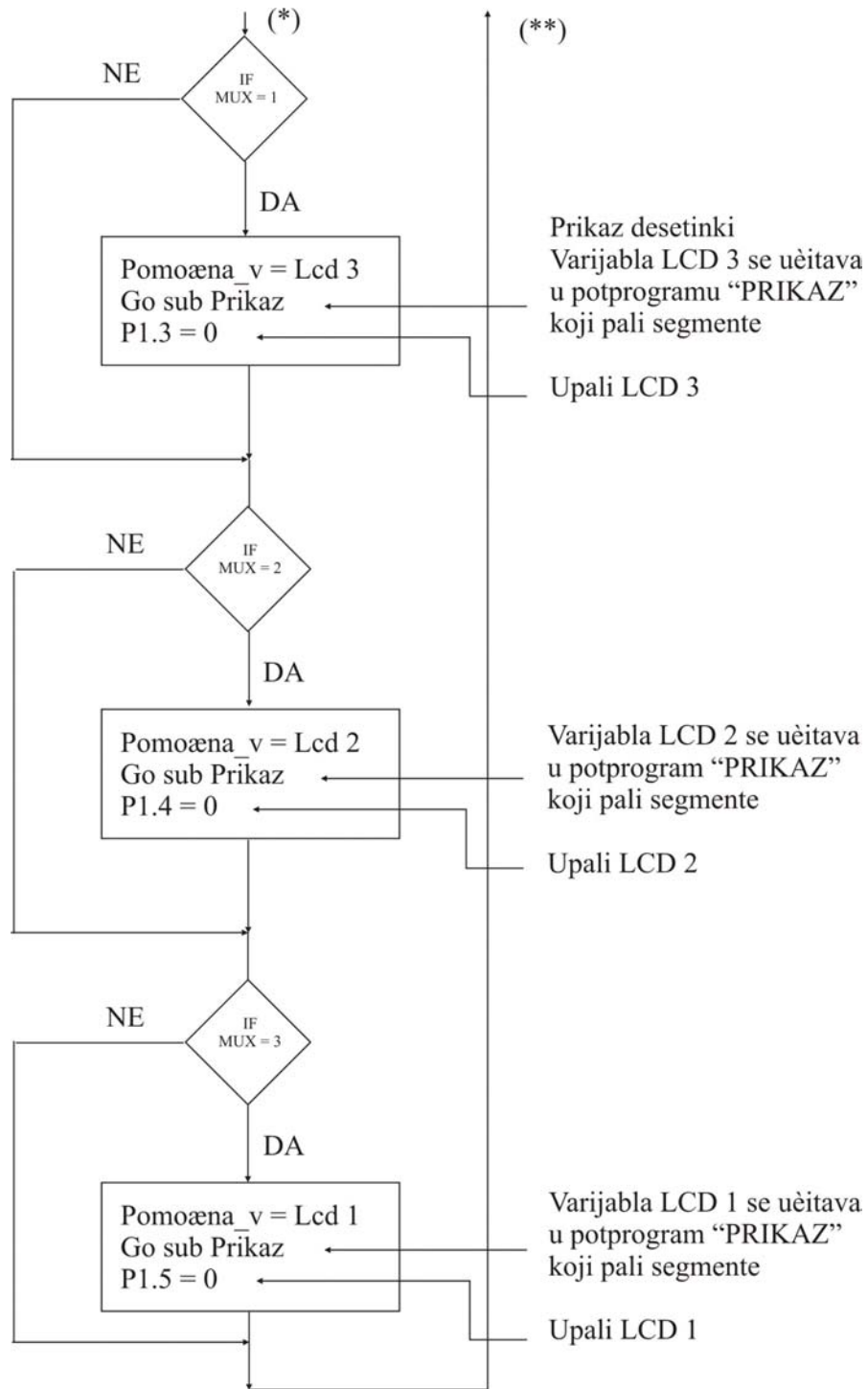
GLAVNI PROGRAM (+ prikaz (potprogram) - svakih 5 msec (20 interrupt) provjerava varijablu stotinke koja se izračunava u `TIMER_0_INT` te računa desetinke, sekunde, desetine sekunde i to prikazuje na jednom od displeja LCD 1 - LCD 4  
Svakih 5 msec mijenja se displej na kojemu se prikazuje znamenka.

PREKIDNA RUTINA `TIMER_0_INT` - Mjeri vrijeme (broji interrupt-e)  
 - Izračunava stotinke i sekunde  
 - Svakih 5 msec omogućuje da glavni program pročitava sekunde i na osnovu njih izračuna desetinke, sekunde, i desetine, te da to prikaže na odgovarajućim LCD displejima.

# GLAVNI PROGRAM DO LOOP







## PREKIDNA RUTINA

Aktivira se svakih 250  $\mu$ sec (na svaki int) Clock, Clock 1, Sekunde As Byte i = 0

