

Hrvoje Duvančić

MIKRORAČUNALA
Laboratorijske vježbe i zbirka zadataka

Izdavač
KIGEN d.o.o.

Za izdavača
Erna Lojna Lihtar

Urednik
Nenad Lihtar

Recenzenti
doc.dr.sc. Vlado Struk
Ante Duvančić, dipl. ing.

Grafički urednik
Nedjeljko Zarić

Oblikovanje naslovnice
Biserka Pađan

Lektura
Vera Vujović

Tisak
GZH d.o.o.

ISBN 978-953-6970-44-5

Sva prava pridržana. Nijedan dio ove knjige ne smije se umnožavati ili prenositi elektronički ili mehanički u bilo kojem obliku ili značenju, uključujući fotokopiranje, presnimavanje te bilo koji drugi sustav prenošenja informacija, bez pismene dozvole Izdavača i Autora.

Hrvoje Duvančić

MIKRORAČUNALA

Laboratorijske vježbe i zbirka zadataka



Zagreb, ožujak 2008

PREDGOVOR

Ova skripta namijenjena je učenicima završnih razreda srednjih elektrotehničkih škola, kao pomoć u svladavanju laboratorijskih vježbi iz predmeta Mikroračunala.

U dvanaest vježbi obuhvaćeno je programiranje u programskom jeziku BasCom i sheme spajanja za izradu elektroničkih sklopova. Oni izvršavaju operacije određene napisanim programima.

Vježbama su obuhvaćene osnovne naredbe koje su potrebne prilikom pisanja svih programa, od školskih primjera do onih za profesionalnu primjenu. Obuhvaćeni su i praktični primjeri koji mogu prikazati hardversku primjenu napisanih programa i korištenih naredbi.

Osim shema korištenih u vježbama, na kraju skripte nalazi se još nekoliko primjera spajanja vanjskih uređaja na mikrokontroler, a neki od njih koriste se i u zadacima na kraju svake vježbe.

Zadaci su zamišljeni za rješavanje na satu; učenik treba napisati program u prostoru koji je predviđen za to i nacrtati odgovarajuću shemu. Taj program također treba napisati i na računalu te snimiti na disketu, kako bi nastavnik provjerio njegovu ispravnost. Na kraju sata listove na kojima se nalaze rješenja treba iskidati iz Zadataka za vježbu i predati nastavniku.

Sve čitateljske primjedbe i prijedloge razmotriti ću u nastojanju da zadaci budu još pristupačniji i bolji.

Autor

SADRŽAJ

UVOD U BasCom.....	1
--------------------	---

VJEŽBE

1. Upravljanje izlazima mikrokontrolera.....	9
2. Upravljanje radom μ C pomoću tipkala	15
3. Upravljanje 7-segmentnim LED displejem	21
4. Upravljanje alfanumeričkim LCD-om	27
5. For – Next petlja	33
6. Potprogrami	39
7. Prekidni način rada	45
8. Watchdog timer	51
9. Serijska komunikacija	57
10. 1-wire komunikacija	67
11. I ² C - komunikacija	79
12. Regulacija zakreta servo motora.....	89

PRIMJERI SPAJANJA	97
-------------------------	----

ZADACI ZA VJEŽBU

1. Upravljanje izlazima mikrokontrolera.....	102
2. Upravljanje radom μ C pomoću tipkala	108
3. Upravljanje 7-segmentnim LED displejem	115
4. Upravljanje s alfanumeričkim LCD-om	120
5. For – Next petlja	123
6. Potprogrami	129
7. Prekidni način rada	133
8. Watchdog timer	141
9. Serijska komunikacija	142
10. 1-wire komunikacija	152
11. I ² C - komunikacija	154
12. Regulacija zakreta servo motora.....	157

POPIS NAREDBI.....	159
--------------------	-----

LITERATURA	163
------------------	-----

UVOD U BASCOM

Bascom je relativno nov programski jezik. Njegov razvoj počeo je na kraju devedesetih godina prošlog stoljeća. Prvi izlazak na tržište bio je s verzijom BascomLT, zatim Bascom 8051, Bascom AVR, a kao posljednju inačicu imamo verziju za programiranje PIC-eva.

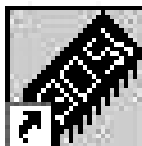
Bascom je kratica od BASic COMpiler, a to znači da se programi pišu u basicu, a zatim se prevode u oblik koji prepoznaje mikrokontroler. Tako Bascom možemo smatrati samo jednim dijalektom obitelji Basic-jezika (Basic, Q Basic, Visual Basic...).

Osnovna mu je namjena stvaranje aplikacija koje će izvršavati mikrokontroleri iz obitelji 8051. Predstavnik te obitelji, a proizvedene u pogonima tvrtke Atmel, s osnovnim podacima imamo u sljedećoj tablici:

	FLASH-MEMORIJA	SDRAM	I/O LINIJE	BROJ TIMERERA	PINOVA	V _{CC}
89C1051	1k	64	15	1	20	2,7 – 6 V
89C2051	2k	128	15	2	20	2,7 – 6 V
89C4051	4k	128	15	2	20	3 – 6 V
89C51	4k	128	32	2	40	4 – 6 V
89C52	8k	256	32	3	40	4 – 6 V
89LS8252	8k+2k EEPROM	256	32	3	40	2,7 – 6 V

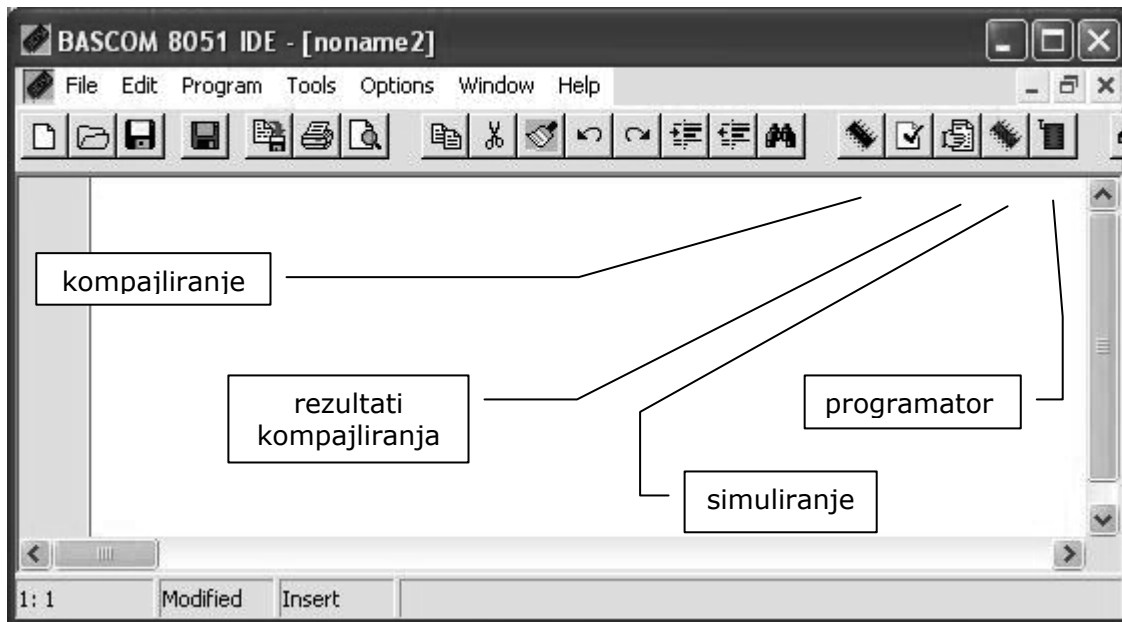
Osim ovih mikrokontrolera postoje još mnoge vrste drugih proizvođača. Uglavnom, najvažnije razlike između svih njih su u veličini memorije u koju spremamo njihove programe te u broju ulazno-izlaznih pinova.

Bascom se pokreće preko ikone koja izgleda ovako:



Naravno, osim pomoću ikone na desktopu, Bascom se može pokrenuti i pronalaskom u «Start» izborniku, pod «Programs», zatim «MCS Electronics» i «Bascom 8051 DEMO» (ako se koristi demo inačica).

Posljedica je otvaranje glavnog radnog prozora koji izgleda kao na sljedećoj slici:

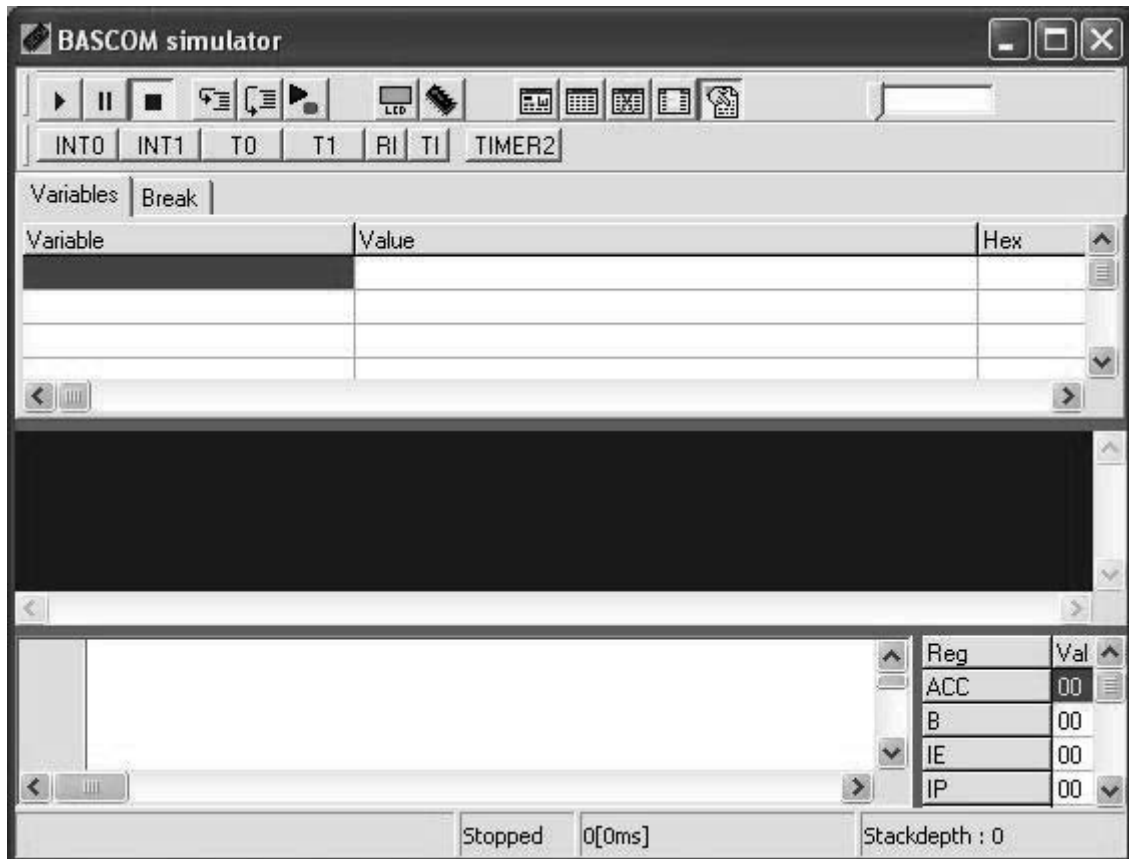


U alatnoj traci vidimo uobičajene naredbe i funkcije koje obavlja svaki programski paket napisan za Windows-okruženje (otvaranje novog dokumenta, otvaranje postojećeg, snimanje...). Osim tih postoje i nove naredbe koje su karakteristične samo za ovaj programski paket i vidimo ih označene s oblačićima. Uz njih postoje još dvije funkcije koje se ne vide na gornjoj slici. Njih nećemo obrađivati u sklopu ovih vježbi.

Prva istaknuta posebna funkcija je «kompajliranje» (compile current file), a namjena joj je prevođenje napisanog programa u oblik koji prepoznaje mikrokontroler. Do nje se nalazi «rezultati kompajliranja» (show result). Kad kliknemo na ovu ikonice otvori se novi prozor u kojem budu zapisani rezultati kompajliranja, kao što su veličina napisanog programa, početna adresa u memoriji mikrokontrolera na kojoj se nalazi naš program, broj korištenih varijabla, tip varijabla itd.

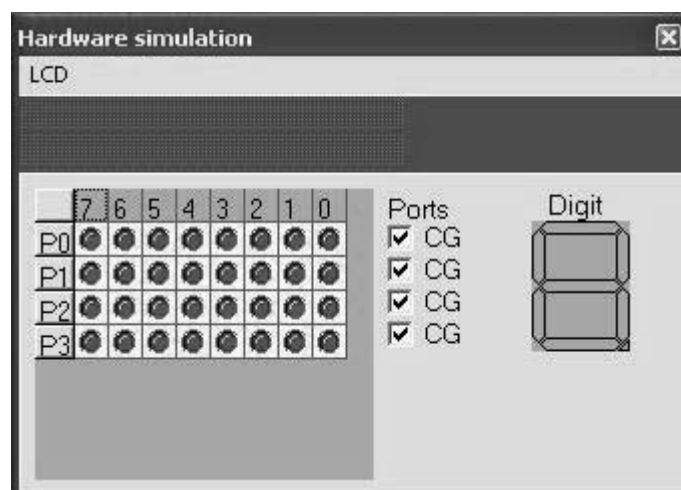
Između ove dvije funkcije nalazi se jedna «pomoćna». Njen zadatak je da provjeri je li «tekst» programa napisan pravilno, tj. jesu li sve naredbe istipkane pravilno, jesu li stavljeni razmaci na odgovarajućim mjestima i silčne stvari (syntax check).

Nakon što je program napisan, snimljen i kompajliran, možemo provjeriti njegovu funkcionalnost. To se može provjeriti pomoću funkcije «simulator» (simulate program). Simulator koji je integriran u Bacom programski paket, tj. prozor koji se otvori nakon pokretanja te funkcije, izgleda ovako:



Ovdje vidimo gumbе pomoću kojih možemo upravljati s radom simulacije, a to su oni koji imaju klasične simbole za «play», «stop» i «pauzu». Samom brzinom izvođenja simulacije upravljamo s klizačem koji se nalazi u desnom gornjem kutu. Položaj u kojem je na slici osigurava najveću brzinu izvođenja simulacije.

Osim njih još nam je zanimljiv gumb na kojem je napisano LCD, jer se s njim pokreće tzv. hardverska simulacija. S njom se može simulirati rad LCD-displeja, 7 segmentnog LED-displeja ili s ulazom i izlazom pojedinog pina na I/O linijama. Nakon pokretanja otvara se sljedeći prozor:



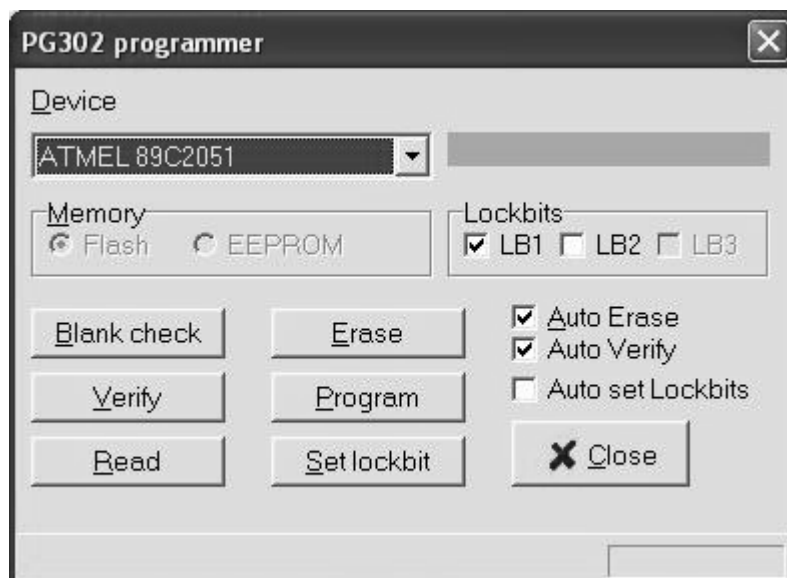
U simulatorskom prozoru valja obratiti pozornost na još neke njegove mogućnosti. Naime, ako se simulira rad programa koji ima ugrađene varijable, tada možemo pratiti stanje pojedinih varijabla u tablici «Variables». U stupac «Variable» utipkamo koja nas varijabla zanima, a u stupcu «Value» pratimo promjene njene vrijednosti tijekom izvođenja programa.

Još jedna stvar kod simulatora je vrlo zanimljiva, a nalazi se na dnu tog prozora u statusnoj traci (Status bar). U ovdje prikazanom primjeru vidimo da je u drugoj «kućici» upisano «Stopped». Tu se ispisiuje stanje u kojem se nalazi simulator općenito, tj. je li program pokrenut, zaustavljen, je li pritisnuta pauza i sl. U trećoj kućici su zapisani nekakvi brojevi. Broj koji se nalazi izvan uglatih zagrada je brojač strojnih taktova koji su potrebni da se izvrši program, dio program ili samo jedna naredba (ovisno o korisnikovim željama). Drugi broj (u uglatim zgradama) je također brojač. Međutim, on ne broji potrebne taktove, nego vrijeme u milisekundama. Oba brojača rade paralelno, tj. kad se vrši resetiranje resetiraju se oba zajedno. Resetiranje se provodi jednostavnim klikom mišem na njihovu kućicu. Ovo resetiranje može se izvesti u bilo kojem trenutku, tako da nije potrebno čekati da se program provede do kraja da bi se to obavilo.

Nakon što smo provjerili rad napisanog programa možemo pristupiti programiranju mikrokontrolera, tj. prebacivanju programa napisanog u Bascomu u mikrokontroler. Taj dio posla možemo obaviti pomoću programatora koji se nalazi u sklopu Bascoma ili aktiviranjem nekog od programatora koji dolaze kao samostalni programi.

U Bascomu postoji nekoliko različitih programatora koji se mogu namjestiti po korisnikovim željama. Do tih programatora se dolazi pomoću «Options» na alatnoj traci u glavnom prozoru Bascoma. U otvorenom padajućem izborniku izabere se opcija «Programmer». Zatim se u novootvorenom prozoru ponovno potraži opcija «Programmer» i tu se izabere koji programator se želi koristiti. U ovom prozoru još je potrebno namjestiti i serijski port preko kojeg će komunicirati softverski i hardverski dio programatora.

Na slici je prikazan programator PG302:



Ovako pokrenut programator koristi za programiranje mikrokontrolera program koji je otvoren na glavnom prozoru Bascoma, a ne može prebacivati nijedan drugi program osim onog koji je trenutno otvoren. Programatori koji dolaze kao samostalni programi nisu ograničeni u tom pogledu i s njima možemo prebacivati u memoriju mikrokontrolera bilo koji program u bilo koje vrijeme. Naravno, potrebno je znati gdje se u memoriji PC-a nalazi hex-datoteka programa koji želimo prebaciti.

Prije početka prebacivanja programa u mikrokontroler potrebno je namjestiti i odgovarajuće parametre programiranja. Najvažnije je da u izborniku «Device» izaberemo vrstu mikrokontrolera koji se nalazi u programatoru. Nakon toga možemo pristupiti samom programiranju!

Prvo moramo izbrisati ono što se već nalazi u memoriji mikrokontrolera, a to obavljam s gumbom «Erase Device». Sljedeći korak je da se klikne na gumb «Program Device». Zatim se provjeri je li program pravilno prebačen, a za to služi gumb «Verify».

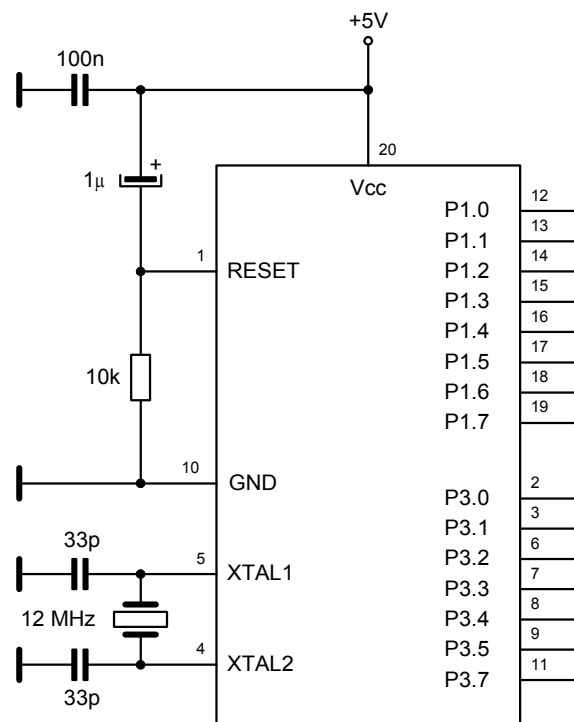
Između preostalih gumba u ovom prozoru istaknut ćemo «Blank check» koji se može iskoristiti za provjeravanje ispravnosti veze između PC-a i hardverskog dijela programatora. Naime, kabel za povezivanje PC-a i programatora može biti u prekidu, a to može uzrokovati problem koji se može protumačiti kao da je programator neispravan ili da je mikrokontroler pregorio!

O ostalim gumbima u ovom prozoru drugom prilikom.

MIKROKONTROLER AT89C2051

Mi ćemo u svom radu koristiti mikrokontroler 89C2051, a u slučaju da je program koji smo napisali veći od 2 KB, koji su nam na raspolaganju kod njega, možemo se poslužiti s 89C4051, koji ima potpuno isti raspored pinova, ali ima dvostruko veći kapacitet memorije. Isti raspored pinova ima i 89C1051.

Osnovna shema po kojoj spajamo ove mikrokontrolere izgleda ovako:



Na ovoj shemi prikazani su elementi neophodni za rad bilo kojeg mikrokontrolera. Tu vidimo da su na pinove 4 i 5 spojeni kondenzatori od 33 pF i kristal od 12 MHz. Oni zajedno određuju radni takt ovog mikrokontrolera, a u slučaju da želimo raditi s drugim radnim taktom, moramo staviti kristal s drugom frekvencijom osciliranja.

Osim ovih elemenata još je spojeno napajanje od +5 V, masa (GND) i RESET na prvom pinu. U slučaju da želimo resetirati mikrokontroler, tada treba kratko spojiti prvi pin na masu, tj. tu se može staviti tipkalo koje bi služilo za resetiranje mikrokontrolera. Drugim riječima, mikrokontroler normalno radi kad mu je na pinu RESET +5 V, a kad je tu 0 V tada dolazi do njegova resetiranja.

Vrijeme trajanja reset-stanja ovisi o izboru otpornika i kondenzatora. Na shemi su upotrijebljeni 1 μ F i 10 k Ω , a ako se to vrijeme želi produžiti potrebno je povećati iznos neke od tih komponenata.

Na shemi se vidi još i kondenzator od 100 nF. On se na pločici treba fizički nalaziti što bliže pinu 20, jer služi za fino filtriranje napona s kojim se napajao mikrokontroler.

Preostali pinovi, na desnoj strani, predstavljaju ulazno-izlazne pinove. Ovdje imamo samo port 1 i port 3, a to je karakteristika 89C1051, 89C2051, 89C4051 (s tim da **P3.6 nema vanjski priključak**), dok 89C51 i 89C52 imaju još port 0 i port 2.

Prilikom spajanja vanjskih uređaja na mikrokontrolere AT89C2051 treba voditi računa da maksimalna dozvoljena struja po pojedinom pinu u stanju «0» iznosi 25 mA.

1. UPRAVLJANJE IZLAZIMA MIKROKONTROLERA

Kao najjednostavniji primjer upravljanja uređajima spojenim na izlaz mikrokontrolera, uzet ćemo paljenje i gašenje LE dioda.

Rečeno je da se oba porta kod AT89C2051 mogu koristiti kao izlazni. Upravljanje se može izvesti s pojedinim pinom na svakom portu ili sa cjelokupnim portom. Upravljanje se svodi na to ima li ili nema signala na određenom pinu, tj. je li on u stanju logičke 0 ili logičke 1.

Za upravljanje s pojedinim pinom koriste se naredbe:

$$\mathbf{P1.x = y} \text{ ili } \mathbf{P3.x = y}$$

gdje je «x» broj pina koji želimo programirati, a «y» je logička vrijednost na tom pinu, **0** ili **1**. Kad želimo upravljati cijelim portom samo s «jednim potezom» tada koristimo naredbu:

$$\mathbf{P1 = y} \text{ ili } \mathbf{P3=y}$$

gdje je «y» vrijednost koja će biti na cijelom portu. Ona se može zapisati u decimalnom ili binarnom obliku, a mora imati vrijednost između **0** i **255**. Imamo li 8 pinova na raspolaganju broj svih mogućih kombinacija je 256. Kad broj pišemo u binarnom zapisu na početak moramo staviti **&B**. Kod porta 3 ne prakticiramo upravljanje s cijelim portom, jer se na shemi vidi da on ima samo 7 vanjskih kontakta, a ne svih 8.

U sljedećem primjeru primijenit ćemo sve ranije spomenuto. Prvo ćemo ugasiti sve LED na portu 1, a zatim ćemo upaliti diode na pinovima 3 i 5. Onda ćemo upaliti diode na svim pinovima prvog porta. To se obavlja naredbom **P1 = 255**, jer je ona u binarnom zapisu **11111111** (logičke jedinice na svim pinovima). Nakon toga ponovo upravljamo s cijelim prvim portom, ali s binarnim zapisom **&B11010111** postavljamo željene logičke vrijednosti na pojedine pinove. Znači, na kraju smo ugasili LED na pinovima 3 i 5.

```
$sim
```

```
P1 = 0  
P1.3 =1  
P1.5 = 1
```

```
P1 = 255  
P1 = &B11010111
```

```
End
```

Ovdje možemo vidjeti na početku programa «\$sim». Ova naredba se koristi kad želimo simulirati program, a prilikom pripremanja programa za prebacivanje u mikrokontroler moramo je izbrisati. Naime, u slučaju da smo imali u programu naredbu «Wait», onda bi simulator zapeo na prvoj takvoj naredbi. Kad se «\$sim» nalazi na početku tad simulator preskače te dijelove izvršavanja programa.

U našem slučaju mogli smo program simulirati i bez nje, ali je dobro steći naviku da je napišemo uvijek prije simuliranja programa (prije simuliranja program se mora kompajlirati).

Kad se napiše program koji ide u mikrokontroler, tada se «\$sim» mijenja naredbom:

\$crystal = x

gdje x predstavlja frekvenciju kristala kvarca koji daje radni takt mikrokontroleru.

U našem slučaju će biti **\$crystal = 12000000**, jer imamo kristal od 12 MHz.

U ovaj primjer, ako ga želimo realizirati s mikrokontrolerom, moramo ubaciti i naredbu «Wait», jer bi se program izvršio prebrzo da bismo uočili sva paljenja i gašenja LE-dioda. Inače, naredba «Wait» može poprimiti vrijednost samo između 0 i 255, pa ako je potrebno duže čekanje tada moramo napisati više tih naredba.

Osim naredbe Wait, za čekanje se može upotrijebiti i jedna «preciznija» naredba, a to je «Waitms». Ona osigurava čekanje zadano u milisekundama. Jednako kao i naredba Wait, može poprimiti vrijednosti samo do 255, pa ako trebamo neki duži početak potrebno je staviti nekoliko ovih naredaba jednu iza druge.

Jedna od mogućih izvedbi je sljedeća:

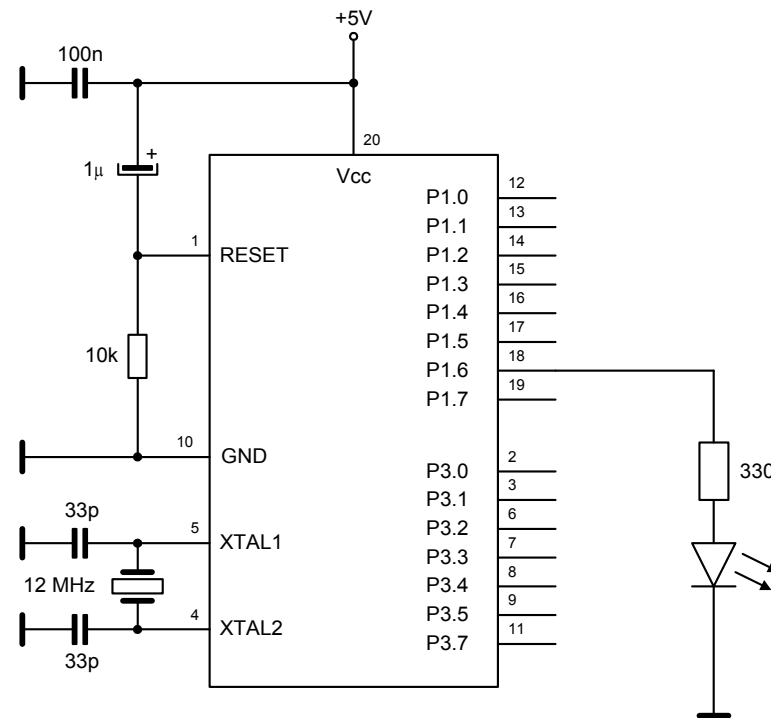
```
$crystal = 12000000

P1 = 0
  Wait 1
P1.3 = 1
  Wait 1
P1.5=1
  Wait 1

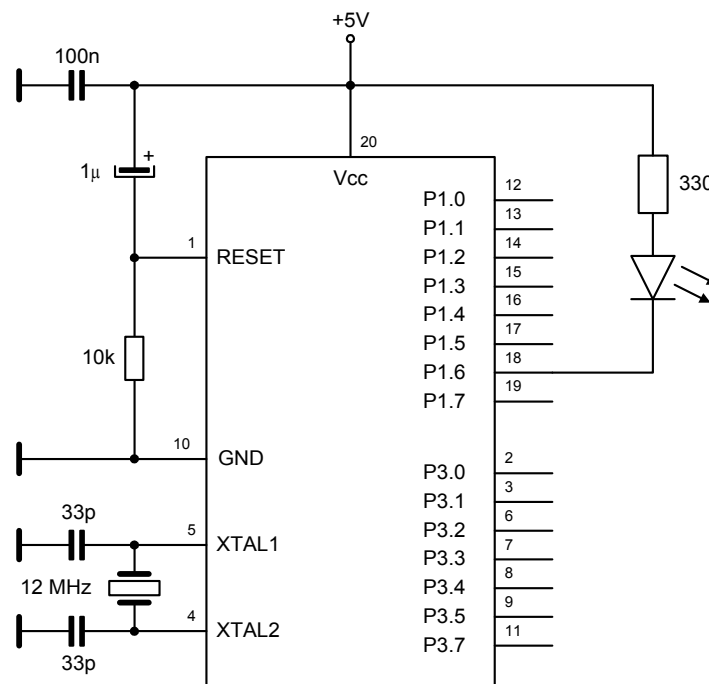
P1 = 255
  Wait 1
P1 = &B11010111

End
```


Sklop koji bi izvršavao gornji program izgledao bi kao na sljedećoj slici (s tim da bi imao spojene LED na svim pinovima P1):



Sklop koji bi radio suprotno od ovog prikazanog na ovoj shemi, bio bi ako bi se LED spojile na način prikazan na sljedećoj shemi:



Suprotnost u njihovu radu nastaje zbog toga što logička jedinica na izlazu mikrokontrolera u naponskom smislu predstavlja napon od 5 V. Tako bi u prvoj shemi logička jedinica djelovala kao izvor napajanja za LED. U drugom slučaju bismo s tom logičkom jedinicom ugasili LED, jer tada na njenim krajevima ne bi postojala razlika potencija, tj. struja ne bi prolazila kroz LED. Inače, ovakav način rada i programiranje ovako spojenih LED možemo nazvati **obrnutom logikom**.

Iz ovog možemo izvući zaključak da je vrlo važan način na koji smo neki vanjski uređaj spojili na mikrokontroler!

ZADATAK:

1. Napišite program u Bascomu koji će upaliti sve LED, zatim ih gasiti _____ po _____. Vrijeme između pojedinih gašenja neka bude _____ s.
Za definiranje vrijednosti na izlazima treba koristiti binarne vrijednosti, a zatim napisati program koji će obavljati istu funkciju pomoću dekadskih vrijednosti.
2. Napišite program u Bascomu koji će prvo postaviti da su sve LED ugašene, a zatim ih paliti _____ po _____ od sredine prema krajevima. Kad dođe do kraja neka ih ugasi na isti način, od sredine prema krajevima. Vrijeme između pojedinih paljenja neka bude _____ s, a između gašenja _____ s.
Za definiranje vrijednosti na izlazima koristiti prvo binarne a potom dekadске vrijednosti.
3. Nacrtajte električnu shemu koja će moći ispuniti gore postavljene zahtjeve.

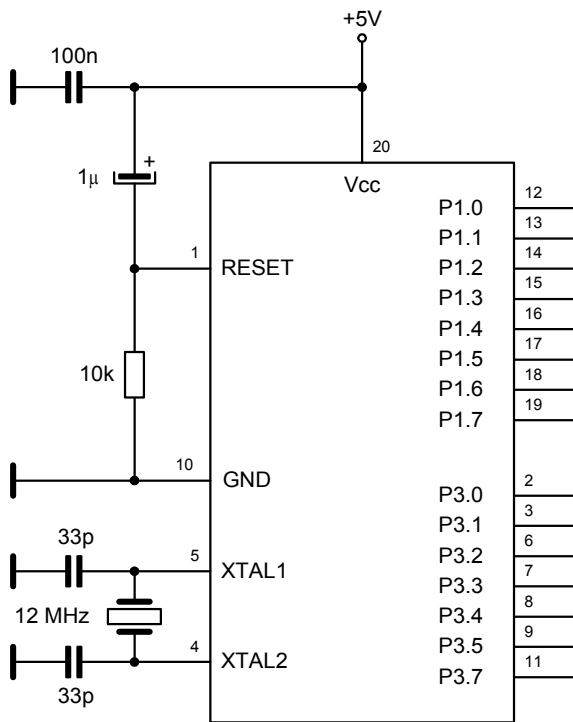
Rješenje 1:

PROGRAM S BINARNIM VRIJEDNOSTIMA	PROGRAM S DEKADSKIM VRIJEDNOSTIMA
----------------------------------	-----------------------------------

Rješenje 2:

PROGRAM S BINARNIM VRIJEDNOSTIMA	PROGRAM S DEKADSKIM VRIJEDNOSTIMA
----------------------------------	-----------------------------------

Rješenje 3:



Ime i prezime	Razred	Bodovi	Ocjena