

SREDNJA TEHNIČKA ŠKOLA  
„NIKOLA TESLA“  
SREMSKA MITROVICA

# MATURSKI RAD

Tema: Pano za matorske slike

Mentor:  
Krsmanović Đoko

Učenici:  
Pejić Vladislav  
Spasojević Vojislav  
Džever Branislav  
Trnjanac Marko

# Pano za matuske slike

Upravljanje zapravo znači birati između različitih mogućnosti, na osnovu prikupljenih informacija. Osobina računara da prihvata, analizira i prenosi veliku količinu informacija velikom brzinom i tačnošću je od izuzetnog značaja pri upravljanju.

Sistem je skup jedinica (elemenata, delova, uređaja, organa, podsistema) funkcionalno povezanih u jednu celinu radi ostvarivanja određenog cilja korišćenjem, pretvaranjem i razmenom energije, materije i/ili informacija.

Sistem čije su sve izlazne veličine – upravljačke veličine je upravljački sistem.

Sistem sastavljen iz objekta i upravljačkog sistema, koje povezuje vektor upravljanja, je sistem upravljanja.

Proces koji se odvija u sistemu upravljanja, a čiji je zadatak da se ostvari željeno dinamičko ponašanje objekta, je proces upravljanja.

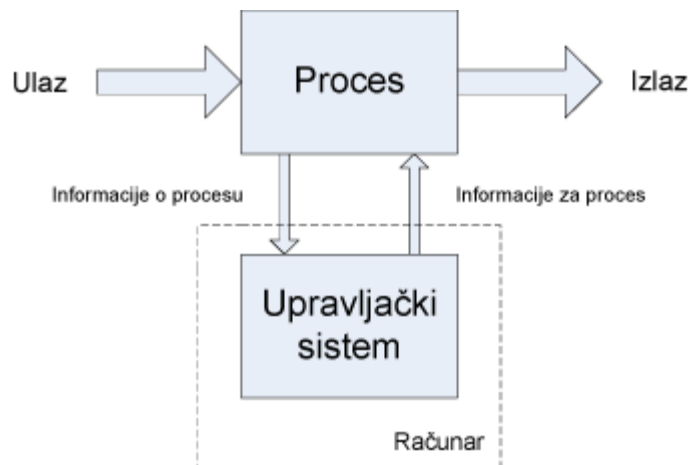
Sistem koji treba da ostvari određeno propisano (zadato, nominalno) kretanje (ponašanje, rad, proces) je objekat.

One ulazne veličine objekta koje izazivaju odstupanje njegovog stvarnog od željenog ponašanja, a nastale su bez korišćenja informacija o tom željenom ponašanju, su poremećajne veličine – poremećaji.

Primena računara u upravljanju procesima radi povećanja efikasnosti proizvodnje uključuje računar kao sastavni deo sistema koji upravlja procesom.

Svaki proces uključuje tri vrste protoka:

1. Protok materijala.
2. Protok energije.
3. Protok informacija.



Slika 1. Blok šema sistema.

Ulaz u proces: (informacija, materijal, signal) menja se u okviru procesa i napušta ga u izmenjenoj formi (izlaz procesa).

Upravljački sistem usmerava proces, mora da poznaje njegovo stanje, na osnovu koga odlučuje kakva uputstva treba dati za njegov dalji rad. Uloga upravljačkog sistema sastoji se u obradi informacija.

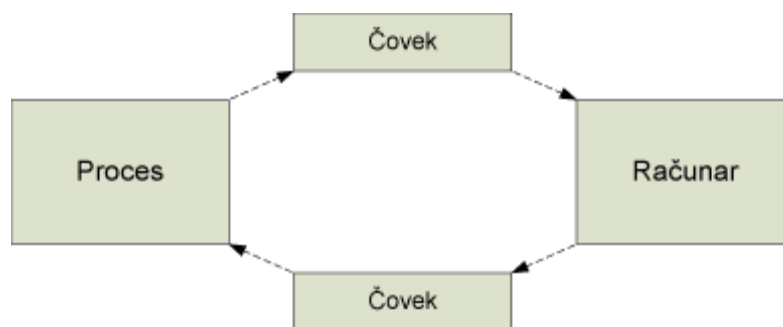
U okviru računarskog upravljanja sistemom izdvajaju se ulazni, izlazni, upravljački i komunikacioni programski zadaci. U odnosu na primenu računara u upravljanju procesima posebno su važni procesni U/I uređaji preko kojih se ostvaruje razmena podataka između procesa i mikroračunara. Procesni U/I obuhvataju: digitalne paralelne i serijske uređaje, A/D i D/A konvertore za prenos analognih signala, kao i brojače i generatore impulsa.

Postoje dve različite sprege računara i procesa:

a) Off-line „nije na liniji“

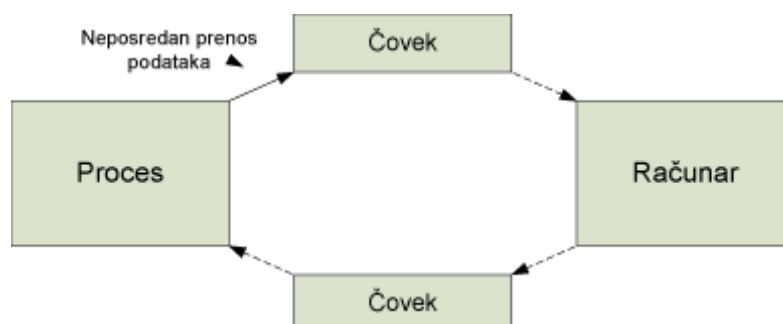
Osnovna odlika ove sprege je da računar prima informacije iz procesa posredstvom nekog medija (uz asistenciju operatera - čoveka) i da se rezultati izračunavanja prenose u proces na isti način.

Ne postoji direktna veza između procesa i računara.



Slika 2. Ručno prikupljanje podataka.

Ručno prikupljanje podataka sastoji se u očitavanju mernih instrumenata (njihovim zapisivanjem na odgovarajućim listama). Ovi podaci se zatim unose u računar preko tastature.



Slika 3. Automatsko prikupljanje podataka.

Automatsko prikupljanje i zapisivanje snimanjem na magnetni medijum (traka, disketa) u obliku u kom ih računar može direktno koristiti.

Off-line sprege primenjuje se tamo gde vreme potrebno za prenos informacija nije od bitnog značaja za rad procesa i gde ručno prikupljanje i prenos informacija nije preobiman posao.

Primeri primene off-line sprege su:

- Eksperimentalna analiza rada procesa.
- Ispitivanje mogućih upravljačkih algoritama.
- Periodično izveštavanje o stanju na procesu.

b) On-line „na liniji“

Postoji čitav niz procesa kod kojih su zahtevi u pogledu dinamike prenosa informacija mnogo strožiji i gde kašnjenje do koga dolazi pri posrednom prenosu dovode u pitanje mogućnost računarskog upravljanja.

On-line sprege računara sa procesom se odlikuje time što je računar fizički vezan za proces.

Računar ima sposobnost da prima signale direktno sa mernih instrumenata postavljenih na procesu i da ih transformiše u formu pogodnu za računarsku obradu. Ovakavi računari nazivaju se procesni ili upravljački računari – računari za rad u realnom vremenu.

Način sprežanja odražava se kako na računarsku opremu za prihvatanje i prenos informacija tako i na računarsko vreme u kome je potrebno da se te operacije obave.



Slika 4. Posredna povratna sprega.

Otvorena sprega (posredna povratna sprega) – uloga računara u ovim slučajevima je informativnog karaktera.



Slika 5. Automatska povratna sprega.

Zatvorena povratna sprega (automatska povratna sprega) – na osnovu obrade informacija primljenih iz procesa dobijaju se uputstva u pogledu daljnjeg rada procesa.

Računar koji je on-line spregnut sa procesom radi u *realnom vremenu* ako se operacije u njemu izvode u okviru iste vremenske skale u kojoj radi i spoljašnji proces.

Redosled obrade informacija u računaru koji radi u realnom vremenu određen je protokom vremena ili nastankom događaja izvan računara.

Rezultat obrade može zavistiti od trenutka vremena u kome se obrada odvija.

Izvođenje pojedinog programskog zadatka može biti uslovljeno:

- a) vremenski,
- b) događajem,
- c) interaktivno.

Vremenska uslovljenost – računarski program je vremenski uslovljen ukoliko njegovo izvršavanje zavisi od protoka vremena. Operacije se izvršavaju periodično i moraju se sve zahtevane funkcije izvršiti u okviru jedne periode, a nakon toga se ponavlja ceo postupak.

Primer je on-line sistem za upravljanje procesom. U takvom sistemu treba izvršiti merenje određenih veličina, na osnovu rezultata merenja izračunati odgovarajuće upravljačke signale i preneti ih na izvršne organe procesa.

Uslovljenost događajem – određene aktivnosti se ne izvršavaju u datim trenucima vremena, već su uzrokovani nekim događajem. Ovakvi sistemi zahtevaju da na procesu postoje senzori koji ukazuju na nastanak nekog događaja (uslova), a od računarskog programa se očekuje da na dati događaj odgovori u okviru maksimalnog dozvoljenog vremenskog intervala. Ovakvi sistemi se koriste da ukažu na nastanak neke alarmantne situacije na nekom procesu, nakon čega se pokreće odgovarajući postupak kojim se uklanja nastala opasnost.

Interaktivni sistemi – njihova osnovna karakteristika je da srednje vreme izvršavanja zahtevane operacije ne bude veće od nekog zadatog vremena. Obuhvataju sisteme za obavljanje novčanih transakcija u bankama, rezervacije hotela ili avionskih karata i slično.

Opisani sistemi se dele na dve klase. Prvu klasu čine sistemi kod kojih je srednje vreme izvršavanja obrade manje od nekog zadatog maksimuma, dok drugu klasu čine sistemi kod kojih se obrada svaki put mora završiti u okviru datog maksimalnog vremenskog intervala.



Početna ideja kojom smo se vodili je bila da napravimo pano za maturalne slike koji će odražavati četiri godine provedene u školi i svo ono znanje koje smo stekli za te četiri godine, ali i takođe da napravimo nešto nsvakidašnje.

Pano za maturalne slike sastoji se iz tri glavne celine: propelera za ispisivanje slova pomoću LED dioda, trostrane prizme na kojoj se nalaze maturalne slike i natpisa „Automatičari“ čija su slova sastavljena od LED dioda.

Sam pano je izrađen od medijapana debelog 10mm koji je pričvršćen na ram od metalnih cevi. Deo rama čini i trostrana prizma koja spaja dva kraja rama koja čine postolje panoa. Preko medijapana sa prednje strane je nalepljen tapet koji služi kao pozadina. Na tapetu je prikazana munja koja slimboliše električnu energiju, a samim tim i našu struku.

Za kontrolu motora, LED dioda i ostalih komponenti koje čine pano koristili smo Microchip-ove PIC kontrolere. Programe za njih smo sami pisali u PIC Basicu. Kontrolere smo programirali pomoću ALLPIC programatora i programa WINPIC 800. Sve pločice smo prvo projektovili

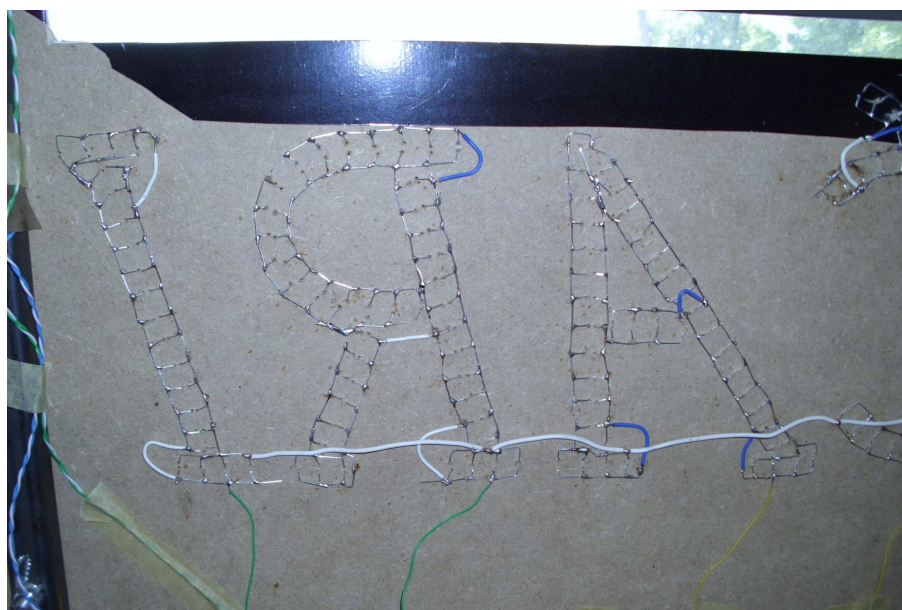
na računaru u programu PCB Designer. Nakon toga smo ih razvili koristeći foto postupak.

## Natpis „Automatičari“



Slika 1.

Ovaj natpis smo napravili pomoću LED dioda i prikazali smo ga na slici 1. Koristili smo oko 400 LED dioda crvene boje, jačine 320mcd. Pomoću njih smo formirali natpis. Sve diode su paralelno povezane i svako slovo predstavlja jednu celinu i jedan izlaz sa mikrokontrolera. Na anodu LED dioda smo direktno doveli napon napajanja od 5V preko otpornika kao na slici 2.

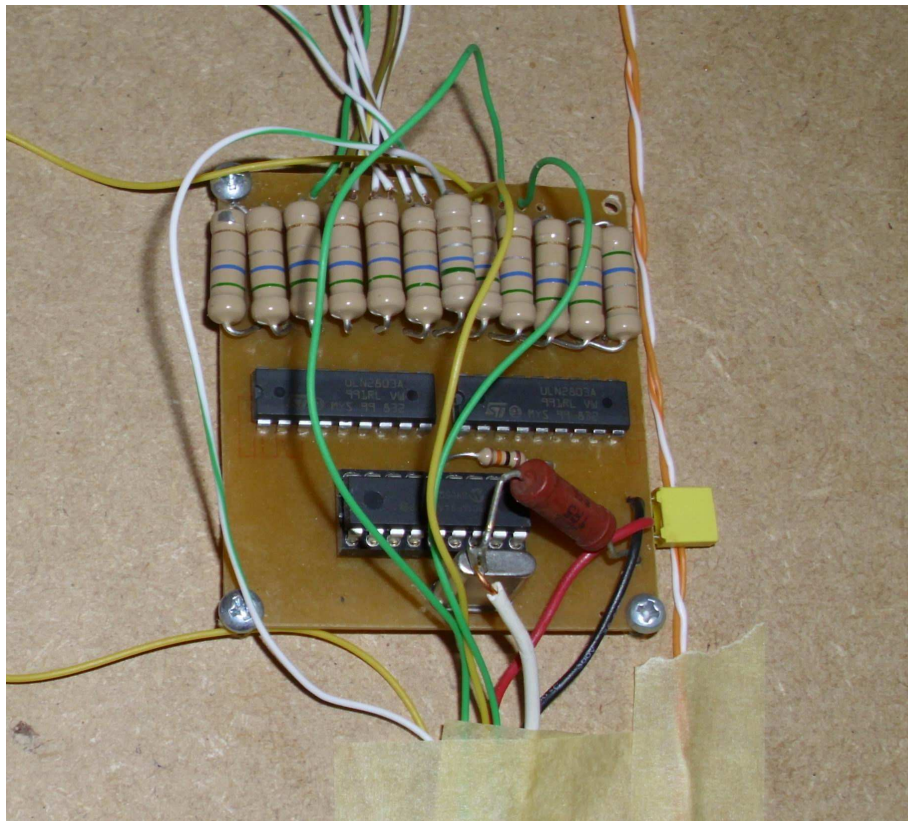


Slika 2.

Pomoću kontrolera PIC 16F84 smo kontrolisali redosled uključivanja i isključivanja LED dioda s ciljem dobijanja određenih efekata. Sve raspoložive portove smo definisali kao izlazne. Njih smo vodili na dva ULN2803. ULN2803 se sastoji iz 8 Darlingtonovih spojeva i služi za povezivanje logičkih kola kao što su TTL, CMOS, itd. i potrošača sa visokom potražnjom napona ili struje. Sa njih smo preko otpornika od  $1\Omega$  snage 2W povezali portove kontrolera sa LED diodama koje čine slova.



Ove otpornike smo stavili zbog velikog broja dioda po izlazu ULN-a tj da bi ograničili struju koja protiče kroz njih, a samim tim smanjili grejanje ULN-ova. Na pločici smo takođe dodali na ulaz napajanja i blok kondenzator od 1mF. Njegova uloga je da eliminiše „pikove“ u naponu napajanja nastalih usled drugih potrošača. Za potrebe ovog natpisa koristili smo oscilator frekvencije od 4MHz. Pošto PIC 16F84 ima invertovan Master Clear morali smo da preko otpornika od 10KΩ dovedemo na njega napon od +5V tj logučku 1. Pločica je prikazana na slici 3., a šemu pločice smo dali u prilogu.



Slika 3.

```
trisa=0
trisb=0
t var byte
p var byte
porta=0
portb=0
t=100
pause 500
pocetak:
portb.7=1: pause t: portb.6=1: pause t: portb.5=1: pause t: portb.4=1
```

```

pause t :portb.3=1 :pause t: portb.2=1 :pause t: portb.1=1: pause t
portb.0=1 pause t: porta.4=1: pause t: porta.3=1: pause t: porta.2=1
pause t: porta.1=1
pause 5000
porta.1=0: pause t: porta.2=0: pause t: porta.3=0: pause t: porta.4=0
pause t: portb.0=0: pause t: portb.1=0: pause t: portb.2=0: pause t
portb.3=0: pause t: portb.4=0: pause t: portb.5=0: pause t: portb.6=0
pause t: portb.7=0: pause t
pause 2000

portb.7=1: porta.1=1: pause t: portb.6=1: porta.2=1: pause t: portb.5=1
porta.3=1: pause t: portb.4=1: porta.4=1: pause t: portb.3=1: portb.0=1
pause t: portb.2=1: portb.1=1
pause 5000
portb.7=0: porta.1=0: pause t: portb.6=0: porta.2=0: pause t: portb.5=0
porta.3=0: pause t: portb.4=0: porta.4=0: pause t: portb.3=0: portb.0=0
pause t: portb.2=0:portb.1=0
pause 2000

for p=1 to 3
porta=255
portb=255
pause 500
porta=0
portb=0
pause 500
next p

goto pocetak
end

```

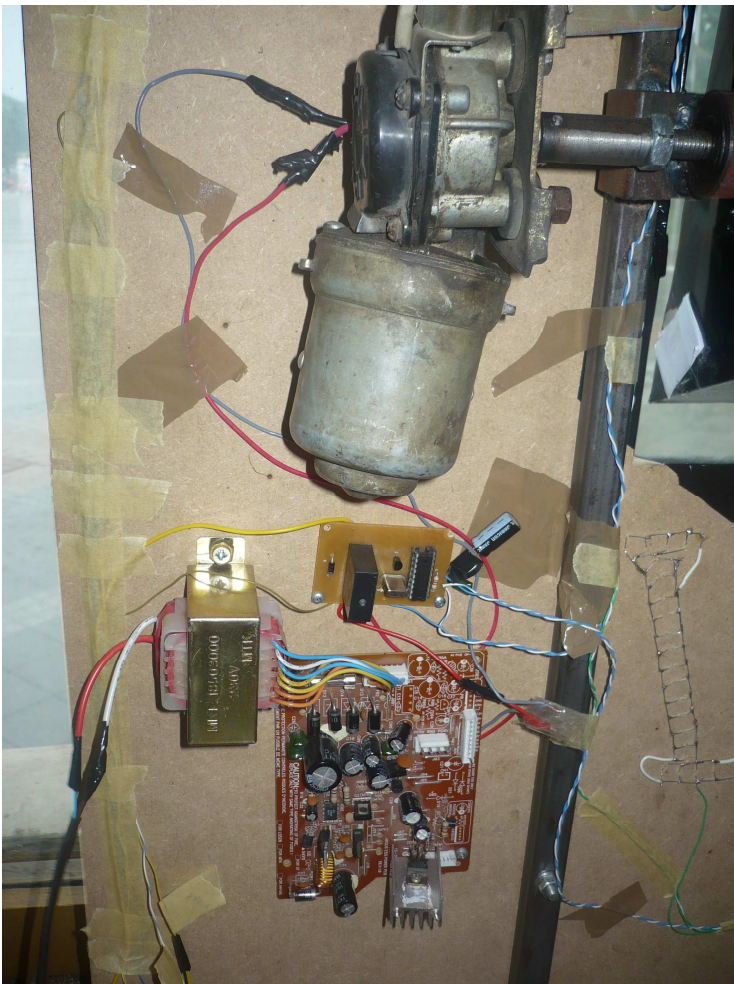
Na datom primeru se vidi deo programa koji smo napisali za PIC s ciljem dobijanja efekata. Program je pisan u PIC Basic programu. U uvodnom delu programa smo kao prvi korak definisali smo portove a i b kao izlazne pomocu komande „tris“ (0 definiše port kao izlazni, a 1 kao ulazni). Takođe smo definisali i neke promenljive koje će nam koristiti kasnije. Drugi korak je bio da zadamo vrednosti portova a i b kao i promenljive t koju koristimo kao pauzu izmedju komandi. Posle ovog uvodnog dela smo pristupili pisanju programa za efekte. Kada želimo da upalimo određeno slovo, za vrednost porta koji kontroliše to slovo (na

primer portb.2) zadajemo 1, a kada želimo da ga ugasimo zadajemo vrednost 0. Sve efekte smo dobili kombinacijom paljenja i gašenja slova.

## Prizma sa maturalim slikama



Slika 4.

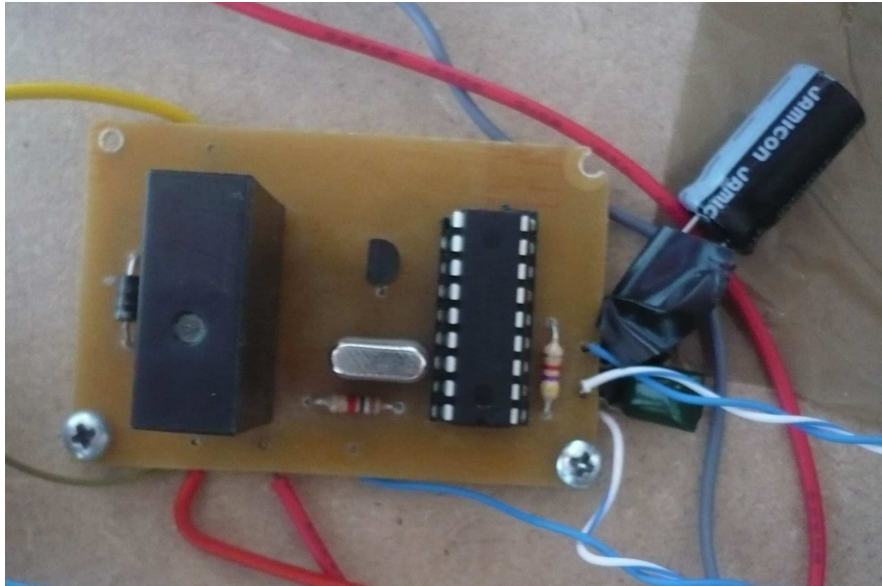


Na slici 4. se vidi jedna stranica prizme na kojoj se nalaze maturalne slike. Prizma je izređena od medijapana debljine 20mm. Na krajevima prizme pričvršćene su dve metalne šipke koje služe kao osovinna na kojoj se prizma obrće. Na jednu od šipki je narezan navoj radi lakšeg spajanja sa osovinom motora. Motor koji smo koristili je bio od brisača. On je bio pogodan zbog toga što ima pužni prenos koji i pored velike mase prizme nema inerciju. Da motor zbog svog rada ne bi unosio smetnje pri napajanju kontrolera i drugih komponenti, koristili smo posebno napajanje od 12V. Na slici pored vidi se ceo sklop:

motor, napajanje i pločica za kontrolu motora sa kontrolerom.

Princip rada prizme je sledeći: motor obće prizmu sve dok jedana od stranica ne pritiske mikroprekidač. On daje impuls mikrokontroleru koji

gasi relej, a samim tim i motor. Motor ostaje ugašen 20 sekundi. Nakon toga se pali i ponovo obrće prizmu i ceo proces se ponavlja. Na slici 5. možete videti kako izgleda pločica na kojoj se nalazi kontroler koji reguliše rad motora, a ujedno i motora propelera.



Slika 5.

Pločica se sastoji od kontrolera PIC 16F84, releja koji se napaja naponom od 12V, tranzistora BC639, diode IN4007, oscilatora frekvencije 4MHz, kondenzatora reda veličine 1000µF i otpornika veličina 1KΩ i 10KΩ. PIC čeka impuls od mikroprekidača. Kada impuls naiđe, kontroler na port a.1 šalje napon. Ovaj napon se preko otpornika od 1KΩ prenosi na bazu tranzistora koji provodi. Pošto je relej u kolektorkom kolu tranzistora i pošto imamo kolektorsku struju relej provodi i spaja kontakte. Kada imamo 0 na izlazu kontrolera tranzistor ne vodi, nema kolektorske struje i relej ne provodi i kontakti su odspojeni.

```
trisa.0=1
trisa.1=0
trisa.2=0
x var byte
pause 2000
x=0
porta.2=1
pocetak:
porta.1=1
```

```
if porta.0=0 then pocetak
porta.1=0
pause 20000
porta.1=1
pause 400
x=x+1
if x>6 then
TOGGLE porta.2:x=0
endif
goto pocetak
end
```

Ovo je program kojim smo programirali PIC. Na početku smo definisali portove, neke kao ulazne neke kao izlazne preko komande TRIS. Port a.0 je izlazni, a portovi a.1 i a.2 su ulazni. Takođe smo definisali i promenljive koje smo koristili u programu. Portove a.0 i a.1 smo koristili za kontrolu motora prizme, a port a.2 za kontrolu motora propelera. Vrednost porta a.2 smo odredili da je 1 i uključili smo motor propelera. Posle labele POČETAK smo uključili motor prizme i zadali petlju kojom program očekuje impuls na portu a.0. Kada ga registruje gasi motor, čeka 20 sekundi i ponovo ga pali. Takođe povećava vrednost promenljive x za 1. Ona služi kao projač za kontrolu motora propelera što je u nastavku objašnjeno.

## Propeler



Propeler za ispisivanje slova pomoću LED dioda pokreće jednosmerni motor koga napajamo sa +12V. Moguće je i povećati napon tj. brzinu obrtanja motora, ali se time povećava i grejanje motora. Kontrolu rada motora smo vršili preko istog kontrolera koji kontroliše rad motora prizme. Propeler radi 2 minuta, a zatim se gasi na 2 minuta. Ovo smo morali da uradimo zbog grejanja motora, a postigli smo tako što smo na svakih 6 obrta prizme gasili i palili motor propelera. To se može videti i iz programa kojim smo programirali PIC:

```
trisa.0=1
trisa.1=0
trisa.2=0
x var byte
pause 2000
x=0
porta.2=1
pocetak:
porta.1=1
if porta.0=0 then pocetak
porta.1=0
pause 20000
porta.1=1
pause 400
x=x+1
if x>6 then
TOGGLE porta.2:x=0
endif
goto pocetak
end
```

Prvi deo programa je vezan za kontrolu motora prizme. Drugi se odnosi na motor propelera. Pri svakom detektovanju impulsa sa mokropekidača program povećava promenljivu x za 1. Kada ona dostigne vrednost 6 program menja vrednost na portu a.2 tj. ako je bila 1 onda je menja na 0 i obrnuto. Ovo gasi ili pali motor. Stanje na portu a.2 se prenosi preko otpornika od  $1K\Omega$  na bazu tranistora BC639 Ako je izlaz na portu a.2 jednako 1 tranzistor ima struju na bazi i provodi. Pošto je rejel u kolektorskom kolu, on provodi i spaja kontakte i motor radi. Kada je izlaz porta a.2 jednak 0 onda ne vodi tranzistor, a ni relej i motor je isključen. Zbog dužine kablova na ulaz između napajanja stavili smo kontenzator od

10 $\mu$ F. Šemu pločice propelera je data u prilogu dok se na šemi koja je data uz kontrolu motora prizme vidi kako je rešena kontrola motora propelera.

Program koji smo napisali za ispisivanje slova izgleda ovako:

```
DEFINE OSC 8
trisa=0
trisc.7=1
t var word
i var byte
j var word
p var word
x var word
p=60
x=50
t=3
portb=0
pause 3000
glavni:
j=0
start:
if portc.7=0 then start
pause 80
portb=64:pause t:portb=127:pause t:portb=66:pause t:portb=0:pause t:portb=24
pause t:portb=84:pause t:portb=84:pause t:portb=84:pause t portb=56:pause t
portb=0:pause t:portb=31:pause t:portb=32:pause t portb=64:pause t
portb=32:pause t:portb=31:pause t:portb=0:pause t portb=127:pause t
portb=0:pause t
pause 10
portb=64:pause t:portb=125:pause t:portb=68:pause t:portb=0:pause t:portb=0
pause t:portb=68:pause t:portb=127:pause t:portb=4:pause t:portb=0
portb=0:pause t:portb=120:pause t:portb=4:pause t:portb=4:pause t
portb=8:pause t:portb=124:pause t:portb=0:pause t:portb=0:pause t
portb=120:pause t:portb=84:pause t :portb=84:pause t portb=84:pause t
portb=32:pause t:portb=0:pause t:portb=0:pause t:portb=8:pause t
portb=4:pause t:portb=4:pause t:portb=8:pause t:portb=124:pause t:portb=0
portb=0:pause t:pause t:portb=124:pause t:portb=32:pause t:portb=64:pause t
portb=64:pause t:portb=60:pause t:portb=0:pause t:portb=0:pause t
portb=0:pause t:portb=68:pause t:portb=127:pause t:portb=4:pause t
portb=0:pause t::portb=0:pause t:portb=120:pause t:portb=84:pause t:portb=84
```

```
pause t:portb=84:pause t:portb=32:pause t:portb=0:pause t:portb=0:pause t
portb=127:pause t:portb=2:pause t:portb=12:pause t:portb=2:pause t
portb=127:pause t:portb=0
```

```
j=j+1
```

```
if j<300 then goto start
```

```
j=0
```

```
pocetak:
```

```
if portc.7=0 then pocetak
```

```
pause p
```

```
portb=120: pause t:portb=84:pause t:portb=84:pause t:portb=84:pause t
portb=32:pause t:portb=0:pause t:portb=64:pause t:portb=127:pause t
portb=65:pause t:portb=0:pause t:portb=32:pause t:portb=84:pause t:portb=84
pause t:portb=84:pause t:portb=72:pause t:portb=0:pause t:portb=24:pause t
portb=84:pause t:portb=84:pause t:portb=84:pause t:portb=84:pause t
portb=56:pause t:portb=1:pause t:portb=1:pause t:portb=127:pause t
portb=1:pause t:portb=1:pause t:portb=0:pause t:portb=0:pause t portb=0
```

```
pause t :portb=0:pause t:portb=0:pause t:portb=120:pause t:portb=84:pause t
portb=84:pause t:portb=84:pause t:portb=32:pause t:portb=0:pause t
portb=64:pause t:portb=127:pause t:portb=65:pause t:portb=0:pause t
portb=56:pause t:portb=68:pause t:portb=68:pause t:portb=68:pause t
portb=56:pause t:portb=0:pause t:portb=68:pause t:portb=40:pause t
portb=16:pause t:portb=127:pause t:portb=0:pause t:portb=0:pause t
portb=64:pause t:portb=125:pause t:portb=68:pause t:portb=0:pause t:portb=0
pause t:portb=127:pause t:portb=16:pause t:portb=8:pause t:portb=4:pause t
portb=127:pause t:portb=0
```

```
p=p+1
```

```
if p>150 then nazad
```

```
goto pocetak
```

```
nazad:
```

```
if portc.7=0 then nazad
```

```
pause p
```

```
portb=120:pause t:portb=84:pause t:portb=84:pause t:portb=84:pause t
portb=32:pause t:portb=0:pause t:portb=64:pause t:portb=127:pause t
portb=65:pause t:portb=0:pause t:portb=32:pause t:portb=84:pause t
portb=84:pause t:portb=84:pause t:portb=72:pause t:portb=0:pause t
portb=24:pause t:portb=84:pause t:portb=84:pause t:portb=84:pause t
portb=84:pause t:portb=56:pause t:portb=1:pause t:portb=1:pause t
portb=127:pause t:portb=1:pause t:portb=1:pause t:portb=0:pause t
portb=0:pause t:portb=0:pause t:portb=0:pause t:portb=0:pause t
```



```
portb=120:pause t:portb=84:pause t:portb=84:pause t:portb=84:pause t
portb=32:pause t:portb=0:pause t:portb=64:pause t:portb=127:pause t:portb=65
pause t:portb=0:pause t:portb=56:pause t:portb=68:pause t:portb=68:pause t
portb=68:pause t:portb=56:pause t:portb=0:pause t:portb=68:pause t
portb=40:pause t:portb=16:pause t:portb=127:pause t:portb=0:pause t:portb=0
pause t:portb=64:pause t:portb=125:pause t:portb=68:pause t:portb=0:pause t
portb=0:pause t:portb=127:pause t:portb=16:pause t:portb=8:pause t
portb=4:pause t:portb=127:pause t:portb=0
```

```
p=p-1
```

```
if p>60 then nazad
```

```
j=j+1
```

```
if j<3 then goto pocetak
```

```
for i=1 to 300
```

```
elektrotehnicar:
```

```
if portc.7=0 then elektrotehnicar
```

```
pause 60
```

```
portb=64:pause t:portb=125:pause t:portb=68:pause t:portb=0:pause t
portb=8:pause t:portb=4:pause t:portb=4:pause t:portb=8:pause t
portb=124:pause t:portb=0:pause t:portb=120:pause t:portb=84:pause t
portb=84:pause t:portb=84:pause t:portb=32:pause t:portb=0:pause t
portb=32:pause t:portb=69:pause t:portb=70:pause t:portb=69:pause t
portb=56:pause t:portb=0:pause t:portb=64:pause t:portb=125:pause t
portb=68:pause t:portb=0:pause t:portb=120:pause t:portb=4:pause t
portb=4:pause t:portb=8:pause t:portb=124:pause t:portb=0:pause t
portb=120:pause t:portb=4:pause t:portb=4:pause t:portb=8:pause t
portb=127:pause t:portb=0:pause t:portb=24:pause t:portb=84:pause t
portb=84:pause t:portb=84:pause t:portb=56:pause t:portb=0:pause t
portb=32:pause t:portb=64:pause t:portb=68:pause t:portb=63:pause t
portb=4:pause t:portb=0:pause t:portb=56:pause t:portb=68:pause t
portb=68:pause t:portb=68:pause t:portb=56:pause t:portb=0:pause t
portb=8:pause t:portb=4:pause t:portb=4:pause t:portb=8:pause t
portb=124:pause t:portb=0:pause t:portb=32:pause t:portb=64:pause t
portb=68:pause t:portb=63:pause t:portb=4:pause t:portb=0:pause t
portb=68:pause t:portb=40:pause t:portb=16:pause t:portb=127:pause t
portb=0:pause t:portb=24:pause t:portb=84:pause t:portb=84:pause t
portb=84:pause t:portb=56:pause t:portb=0:pause t:portb=64:pause t
portb=127:pause t:portb=65:pause t:portb=0:pause t:portb=65:pause t
portb=73:pause t:portb=73:pause t:portb=73:pause t:portb=127:pause t:portb=0
```

```
next i
```

```
goto glavni
```

```
end
```

Program je sastavljen od uvodnog dela, prvog, drugog i trećeg ispisa. U uvodnom delu kao i uvek smo definisali promenljive, odredili brzinu oscilatora, zadali početne vrednosti. Slova smo ispisivali vertikalni red po red. Promenljiva t nam predstavlja pauzu između redova u slovu, dok ostale se koriste za razne brojače. Prvi ispis je bio „Maturanti IVE1“. Pošto ispis počinje od nazad tako smo morali i da unosimo vrednosto koje će kontroler da šalje na port b. Imamo 7 dioda i svaka imam svoju vrednost od 1 do 64. Zbog pravca ispisa prvo idemo od „1“, pa „e“, pa „V“ i tako sve do kraja ispisa. Nakon toga uvodimo brojač i petlju koji rade na sledeći način: nakon svakog ispisa „Maturanti Ive1“ brojač se povećava za 1 i dok je on manji od 300 petlja se vraća na labelu START koja se nalazi na početku ispisa.

Kada vrednost pređe 300 program prelazi na sledeći ispis, a to je „Nikola Tesla“ koji se skroluje od jednog kraja do drugog i počinje labelom POČETAK. Isto pravilo važi i za ovaj ispis. Jedina razlika su petlje koje smo uveli. Prvo smo uveli petlju i brojač koji skroluju ispis napred. Oni to rade tako što smo pre početka ispisa uveli pauzu od koje zavisi gde će početi ispis, koja zavisi od promenljive p. Nakon svakog ispisa promenljiva se povećava za 1 i pomera ispis prema napred i to sve dok ne dostigne vrednost 150. Kada dostigne 150 program nastavlja dalje na labelu NAZAD koja ispisuje tekst i skroluje ga unazad. Ovo smo takođe izveli preko brojača i petlje samo ovaj put se promenljiva p smanjuje za 1 do vrednosti od 60 koju smo zadali na početku. Kada dostigne 60 vraća se na labelu POČETAK. Na kraju petlje koja skroluje tekst unazad smo stavili brojač koji povećava promenljivu j za 1 kada program odradi skrolovanje slova napred i nazad. Ovo će se ponavljati sve dok promenljiva j ne dostigne vrednost 3 kada program nastavlja dalje na labelu ELEKTROTEHNIČAR.

Ovo je zadnji ispis. Pre labelu ELEKTROTEHNIČAR smo uveli FOR petlju koja za vrednosti promenljive i od 1 do 300 ispisuje „Elektroteničari“. Kada se program ispiše ispis vrednos promenljive i se povećava za 1 i to se ponavlja sve dok ne dostigne vrednost 300 kada se program vraća na početak i na prvi ispis. Kod svih ispisa smo posle svakog slova smo ostavljali jedan vertikalni red razmaka, pauze između redova smo odredili da budu 3.