

Izdelava mobilnega robota z razvojnim sistemom roboPIC (5)

Avtor: Silvan Bucik
E-pošta: silvan.bucik@tscng.net
www.svet-el.si/phpBB2/index.php

V pretekli številki smo se posvetili programskemu orodju, danes pa bi se zopet vrnil v hardverske vode. Pogledali si bomo princip krmiljenja enosmernih motorjev: spoznali bomo način dvosmernega krmiljenja ter regulacije hitrosti motorjev. Upamo, da boste ob branju uživali.

Nagradne naloge iz 123. številke:

Na zastavljeni nagradni nalogi iz predpretekle številke smo dobili veliko število odgovorov. Veseli nas, da je serija roboPIC sprožila tolikšno zanimanje med vami. Vsi, ki ste poslali pravilne rešitve, boste kot nagrado prejeli mikrokontroler PIC16F877. Nagrado je prispevalo podjetje ELBACOMP d.o.o., podjetje za trgovino, zastopanje in distribucijo elektronskih komponent. Za izkazano podporo se sponzorju iskreno zahvaljujemo. Tokratni nagrajenci pa so:

- Matevž Bošnjak,
- Andrej Medved,
- Miran Kokol,
- Primož Cvelbar in
- Darko Tibaut.

Čestitamo vsem nagrajencem, ostalim pa želimo več uspeha pri prihajajočih nalogah. Sedaj pa predlagam, da si skupaj ogledamo pravilne rešitve nalog.

Naloga 1: Napišite program, ki bo omogočil prižig led diod na priključkih RB2 in RB7 le ob sočasnem pritisku tipk RA2 in RE3. V kateremkoli ostalem primeru diodi ne smeta svetiti.

Pri tej nalogi bi se bralcem opravičil za tiskarsko napako: priključek RE3 vrat PORTE namreč ne obstaja. Vrata PORTE imajo le tri priključke, in sicer: RE0, RE1 ter RE2. Večina vas je napako opazila in neobstoječi priključek preimenovala v RE2. V želji, da bi napako popravili, smo kot pravilne rešitve upoštevali tudi programska besedila, v katerih ste se sklicevali na priključek RE3. Hvala za razumevanje.

Kot primer dobre rešitve si pogledjmo Miranovo nalogo:

```
void main (void)
{
  TRISA = 0B11111111; // vhodi
  ADCON1 = 0x07;
  TRISB = 0B01111011; // 2 in 7 sta izhoda za LED!
  TRISC = 0B11111111; // vhodi
  TRISD = 0B11111111; // vhodi
  TRISE = 0B00000111; // vhodi
  RB2 = 1; // diodi na RB2 in
  RB7 = 1; // RB7 sta ugasnjeni
  // GLAVNI PROGRAM
  do
  {
    if (RA2==0) // stikalo na RA2 vklopljeno
```

```
{
  if (RE2==0) // tudi stikalo na RE2 vklopljeno
  {
    RB2 = 0; // obe diodi
    RB7 = 0; // se prižgeta - GORITA!
  }
  else
  {
    RB2 = 1; // diodi se
    RB7 = 1; // ugasneta - NE gorita!
  }
}
else
{
  RB2 = 1; // diodi se
  RB7 = 1; // ugasneta - NE gorita!
}
}
while (1>0);
}
```

Naloga 2: Napišite program, ki bo povzročil utripanje led diode na priključku RB1 le ob pritisku tipke RA5.

Poglejmo si Primožovo rešitev:

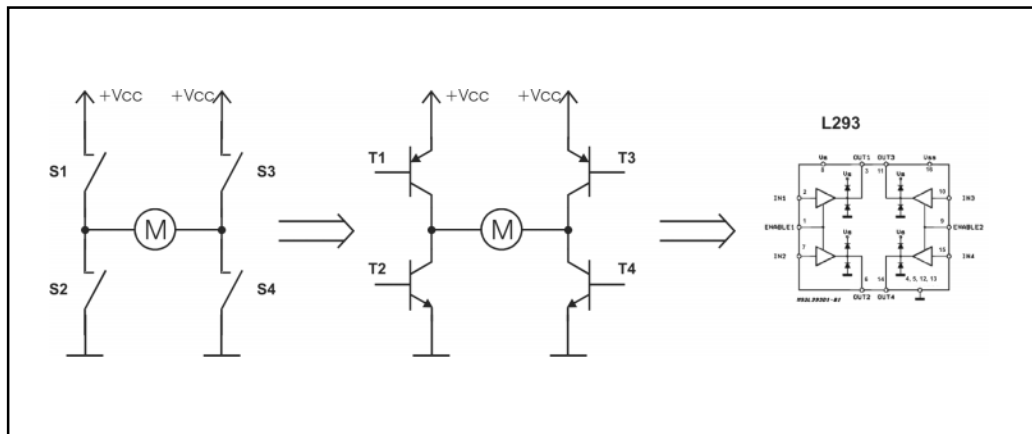
```
void main(void)
{
  TRISA = 0B11111111; // RA5 vhod
  ADCON1 = 0x07; // PORT A DIGITALNI
  TRISB = 0B11111101; // RB1 IZHOD
  TRISC = 0B11111111; // PORTC NE RABIMO
  TRISD = 0B11111111; // PORTD NE RABIMO
  TRISE = 0B00000111; // PORTE NE RABIMO
  RB1=1;
  // GLAVNI PROGRAM
  while (1>0)
  {
    if (RA5==0)
    {
      RB1=0;
      wait_mili (100) ;
```

```
RB1=1;
wait_mili (100) ;
}
else
RB1=1;
}
}
```

Krmiljenje enosmernega motorja

Enosmerne motorje ni pa ni nič kaj težko krmiliti, boste rekli. Priključimo ga na enosmerno napetost, pa je. To je res, toda pri mobilnih robotih so zahteve glede krmiljenja pogonskih motorjev malce strožje. Motorji se morajo vrteti v obeh smereh, z možnostjo spreminjanja obodne hitrosti vrtenja. To pa ni več tako enostavno. Tu nam bo v pomoč integrirano vezje L293 proizvajalca SGS Thomson.

Poenostavljen princip delovanja integriranega vezja je predstavljen na spodnji sliki:



Slika 1: Princip krmiljenja enosmernih motorjev

Če želimo enosmerni motor voditi dvosmerno, potrebujemo vezje, ki bo menjavalo polariteto napetosti na priključkih motorja. Delovanje takšnega vezja bomo najlažje razumeli, če si zamislimo mostično vezavo štirih stikal. Iz vezave je razvidno, da se bo motor vrtel v izbrano smer v primeru sklenitve stikal S1 in S4; ob pogoju, da sta stikali S2 in S3 razklenjeni. Smer vrtenja spremenimo, če sklenemo stikali S2 in S3, pri čemer morata biti stikali S1 in S4 razklenjeni. Vezava nam omogoča tudi ustavitve motorja, pri čemer nastopa več možnih kombinacij. Ena od možnosti je, in sicer najbolj zanesljiva, da so vsa stikala razklenjena; lahko pa dopuščamo kombinacije sklenjenih stikal S1-S3 ali S2-S4 oziroma je sklenjeno le eno (katerokoli) stikalo. Mostična vezava skriva tudi past, ki se ji je dobro izogniti, če ne želimo prehitro imeti praznih baterij. Nevarnost predstavljata kombinaciji sklenjenih stikal S1-S2 ali S3-S4. V tem primeru se pojavi kratek stik med napajalnima linijama, kar povzroči hitro praznjenje baterij, da o uničenih stikalnih elementih ne govorimo.

Boljšo rešitev od uporabe stikal (mogoče se je komu ponudila rešitev z uporabo relejev) predstavlja veza-

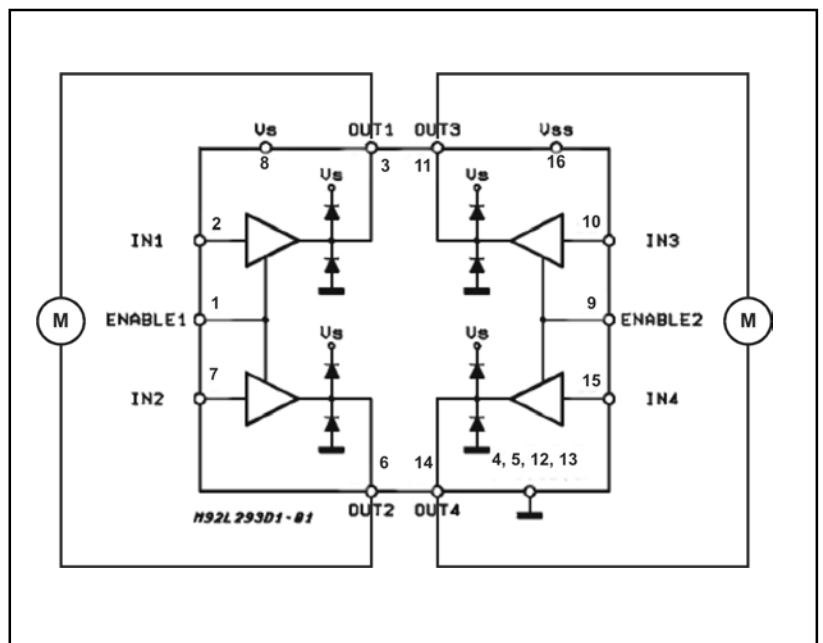
va s tranzistorji, ki delujejo kot stikalo. Tranzistorje krmilimo z napetostjo na bazi. Če na primer na bazo tranzistorja pripeljemo dovolj visoko napetost (nekje med 0,7 V in 0,8 V), se tranzistor odpre v nasičenje, kar predstavlja ekvivalent sklenjenemu stikalu. Če pa na bazo tranzistorja ni napetosti, ta ostaja zaprt in ne prevaja električnega toka (kar predstavlja ekvivalent razklenjenemu stikalu).

Kaj pa integrirano vezje L293? L293 je izboljšana različica tranzistorne mostične vezave. Pa si jo oglejmo поближе.

Gonilnik L293

L293 je krmilni modul (gonilnik) za upravljanje dveh enosmernih motorjev. Gonilnik omogoča popolni nadzor nad delovanjem enosmernega motorja: motorju lahko na enostaven način spreminjamo smer vrtenja, z dodatno možnostjo krmiljenja hitrosti vrtenja. Poglejmo si nekaj osnovnih značilnosti gonilnika L293. Vezje se napaja v območju od 4,5 V do 36 V, in sicer preko napajalnih sponk Vss in Vs. Preko priključka Vss je izvedeno napajanje krmilne logike, priključek Vs pa je namenjen napajanju motorjev. Ti

dve napajalni napetosti sta lahko različni, vendar mora biti napajanje motorjev nekoliko višje ali kvečjemu enako napajalni napetosti krmilne logike. Integrirano vezje je razdeljeno na dva neodvisno delujoča modula. Delovanje posameznega modula omogočimo z aktivno 1 na priključku ENABLE1 oziroma priključku ENABLE2. Ob logični 0 bo delovanje posameznega modula onemogočeno, ne glede na stanje ostalih krmilnih signalov. V tem primeru se motor ne more vrteti, ne glede na stanji na krmilnih vhodih IN1, IN2 oziroma IN3, IN4.



Slika 2: Notranja zgradba integriranega vezja L293

IN1 (IN3)	IN2 (IN4)	OUT1 (OUT3)	OUT2 (OUT4)	obnašanje motorja
0	0	0	0	motor stoji
0	1	0	1	vrtenje v levo
1	0	1	0	vrtenje v desno
1	1	0	0	motor stoji

Tabela 1: Delovanje integriranega vezja L293

Smer vrtenja motorjev nadziramo s krmiljenjem priključkov IN1, IN2 ter IN3, IN4. Vedenje motorja v odvisnosti od vhodnih podatkov opisuje tabela 1.

Sedaj pa preizkusimo delovanje motorja. Napišimo program, ki bo povzročil vrtenje motorja M2 (smer vrtenja poljubno izberemo) za čas petih sekund. Nato naj se motor ustavi. Izhodni priključek motorja M2 je na sponki X3, priključek motorja M1 pa na sponki X4.

Postopek reševanja naloge:

- Priključke mikrokontrolerja RC1, RC2, RC4 in RC5 definiramo kot izhode, vsi ostali priključki so vhodi.
- Omogočimo delovanje druge polovice gonilnika L293 s postavitvijo ENABLE2 na vrednost 1 (priključek RC1); ENABLE1 pa postavimo na vrednost 0 (priključek RC2).
- Poženemo motor s postavitvijo bitov RC4 in RC5 na ustrezni vrednosti (če je RC4 enak 1, potem je RC5 enak 0 ali obratno).
- Počakamo 5 sekund (uporabimo knjižnično funkcijo `wait_sec(zakasnitev_v_s)`).
- Ustavimo motor z brisanjem bitov RC4 in RC5.

Rešitev problema v programskem jeziku C

```
// INICIALIZACIJA
void main (void)      // IME GLAVNE FUNKCIJE
{
    TRISA = 0B11111111; // PORTA = vhod, ga ne uporabljamo
    ADCON1 = 0x07;     // PORTA definiramo kot digitalna
                       // vhodno-izhodna vrata
```

```
TRISB = 0B11111111; // PORTB = vhod, ga ne uporabljamo
TRISC = 0B11001001; // RC4, RC5, RC1, RC0 so izhodi
TRISD = 0B11111111; // PORTD = vhod, ga ne uporabljamo
TRISE = 0B00000111; // PORTE = vhod, ga ne uporabljamo

RC2 = 0; // brisanje bita ENABLE1; // onemogočeno delovanje sklopa MOT1 v
int. vezju L293
RC1 = 1; // postavitve bita ENABLE2; // omogočeno delovanje sklopa MOT2 v
int. vezju L293

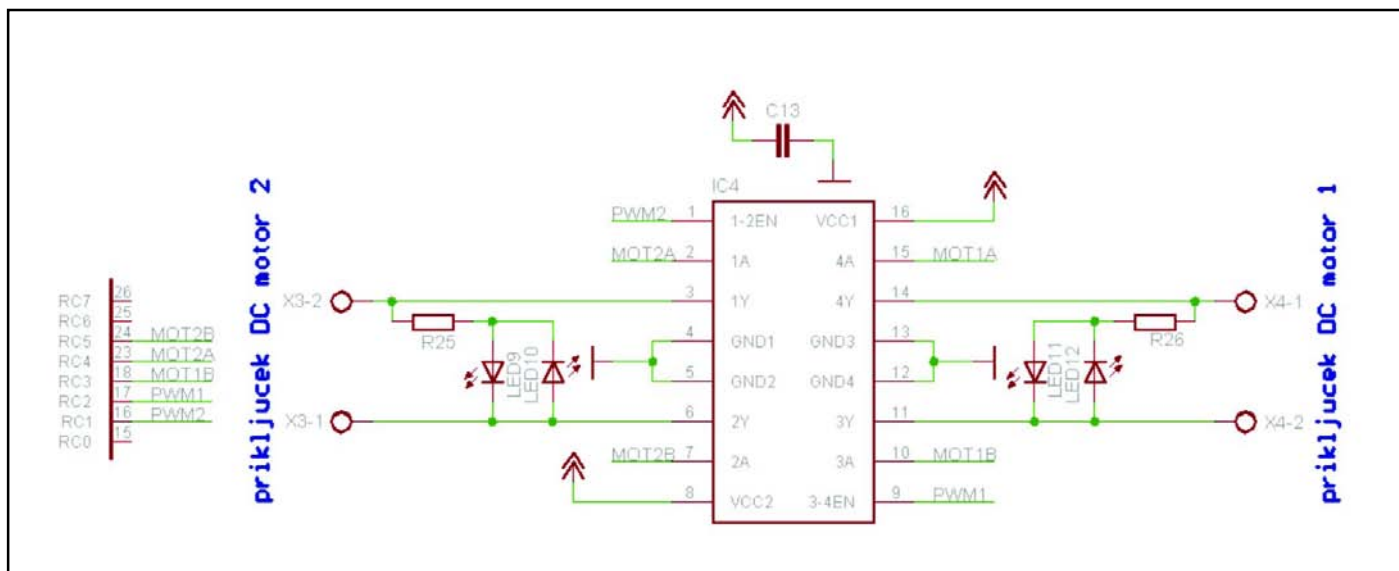
// GLAVNI PROGRAM
RC4 = 1; // vrtenje motorja
RC5 = 0; // v izbrano smer
wait_sec(5); // počakaj 5 sekund
RC4 = 0; // ustavi
RC5 = 0; // motor
stop;
goto stop;
}
```

Samostojno rešite sledeče naloge

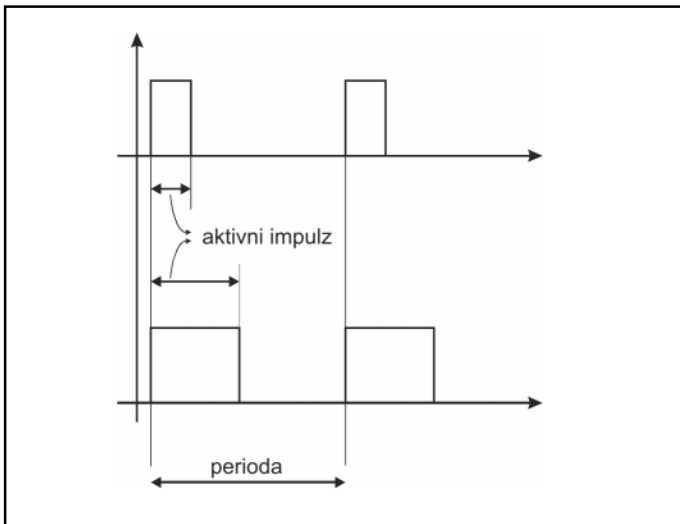
1. Spremenite program tako, da se bo namesto motorja M2 vrtel motor M1, ki je vezan na priključek X4.
2. Spremenite program, da bo motor tekel le primeru, ko bo pritisnjena tipka RA2.
3. Napišite program, ki bo povzročil zamenjavo smeri vrtenja motorja vsake 3 sekunde.

Regulacija vrtljajev enosmernega motorja

Sedaj nam ostane še regulacija vrtljajev. Vrtljaje nastavljam po principu delovanja ON/OFF regulacije: motor bomo za nekaj časa vzbudili, čez čas pa mu bomo napetost odvzeli. Delovanje tovrstne regulacije je moč lepo opazovati pri nizkih frekvencah preklapljanja, ki se kaže kot počasno preskakovanje motorja. Če bi hitrost preklapljanja povečali (dvignili frekvenco preklopov), bi omenjeno preskakovanje počasi izginilo in motor bi se pričel vr-



Slika 3: Povezava integriranega vezja L293 z mikrokontrolerjem



Slika 4: PWM signal

teti tekoče. Opazili bi, da se v tem primeru motor vrti z manjšo kotno hitrostjo, kot se je pri polni napetosti brez regulacije. Frekvenco preklopov določimo s poskusi, a za začetek naj bo ta okoli 200 Hz.

Hitrost vrtenja motorja nastavljamo s spreminjanjem ON/OFF prevajalnega razmerja krmilne napetosti. Prevajalno razmerje, podajamo ga v odstotkih, je razmerje med časom vklopa in celotno periodo cikla. Čim višje je to razmerje, tem hitreje se motor vrti. Čim nižje bo prevajalno razmerje, nižji bodo vrtljaji. Vendar tu ne gre pretiravati. Z nižanjem prične navor motorja padati ter pri določenih vrednostih prevajalnega razmerja pade pod kritično mejo in motor se zaustavi.

Pulzno-širinski modulator (PWM)

Za izvedbo ON/OFF krmiljenja bomo uporabili PWM modul (pulse width modulation), ki je že vgrajen v mikrokontrolerju. PWM modul je generator signala konstantne frekvence, z nastavljivim prevajalnim razmerjem. Pulzno-širinski modulator (PWM) je generator signala pravokotne oblike. Njegova posebnost je, da mu lahko poljubno nastavljamo širino aktivnega impulza, pri čemer se frekvenca signala ne spremeni. Razmerje nastavitve časov visokega (logična 1) in nizkega (logična 0) stanja izhodov imenujemo prevajalno razmerje (duty cycle). To lastnost PWM generatorja spretno izkoriščamo pri regulaciji moči enosmernega motorja ali pri krmiljenju servomotorjev. Poglejmo malo v »notranje organe« mikrokontrolerja PIC16F877. PWM generator je podsklop modula CCP, katerega takt določa časovnik Timer2. Mikrokontroler vsebuje dva takšna sklopa: CCP1 in CCP2, torej lahko uporabimo dva PWM generatorja. Generatorja med seboj nista neodvisna. Njuni frekvenci sta vedno enaki, le prevajalno ra-

zmerje lahko nastavljamo ločeno. Izhoda obeh sklopov sta na priključkih vrat PORTC, in sicer na priključku RC2 kot izhod modula PWM1 ter RC1 kot izhod modula PWM2.

Inicializacijo PWM modula izvedemo z nastavitvijo registrov TRISC, T2CON in CCP1CON (ali CCP2CON). Glede na izbiro uporabljenega sklopa določimo v registru TRISC ustrezni izhodni priključek (RC1 ali RC2) kot izhod.

V neposredni povezavi s sklopom PWM je tudi časovnik Timer2, ki določa frekvenco delovanja PWM sklopa. Njegovo delovanje aktiviramo v registru T2CON, obenem pa tu nastavimo tudi želeno preddelilno razmerje časovnika Timer2 (tabela a).

Postavljen bit TMR2ON na 1 omogoči delovanje časovnika, bita T2CKPS1 in T2CKPS0 pa določata njegovo preddelilno

T2CKPS1	T2CKPS0	delilno razmerje
0	0	1:1 (višja frekvenca)
0	1	1:4
1	x	1:16 (nižja frekvenca)

Tabela 2: Delilna razmerja časovnika Timer2

razmerje.

Z nastavitvijo bitov registra CCP1CON (oziroma CCP2CON) izberemo način delovanja sklopa CCP, in sicer kot PWM generator. To storimo s postavitvijo bitov CCP1M3 in CCP1M2 na logično vrednost 1 (tabela b).

Periodo PWM generatorja poleg delilnega razmerja določa tudi vrednost registra PR2. Še enkrat bi želeli opozoriti, da sta periodi obeh sklopov, tako PWM1 kot PWM2, enaki in ju ni možno nastavljati ločeno.

$$perioda_PWM = (PR2 + 1) \cdot \left(\frac{4}{frekvenc_oscilatorja}\right) \cdot predelilno_razmerje_Timer2$$

Širino impulza izberemo z nastavitvijo 10-bitne vrednosti, katere 8 višjih bitov hrani register CCP1L (oziroma CCP2L), spodnja dva bita pa se nahajata v registru CCP1CON (CCP2CON) na mestih bitov 5 in 4.

$$širina_impulza = \frac{4 \cdot CCP1L + bita_5_4_CCP1CON}{frekvenc_oscilatorja \cdot predelilno_razmerje_Timer2}$$

T2CON							
-	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)
-	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0
-	-	-	-	-	1		

R - omogočeno branje bita, W - omogočeno pisanje v bit, - - bit je neuporabljen ali ni dostopen, () - vrednost bita po resetu, (x) - vrednost bita po resetu ni definirana

Tabela a

CCP1CON							
-	-	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)
-	-	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0
-	-			1	1	x	x

R - omogočeno branje bita, W - omogočeno pisanje v bit, - - bit je neuporabljen ali ni dostopen, () - vrednost bita po resetu, (x) - vrednost bita po resetu ni definirana

Tabela b

Pri nastavitvi parametrov PWM modula bodimo pozorni na to, da je nastavljena širina impulza vedno manjša od periode signala. V nasprotnem primeru bo na izhodnih priključkih vedno nespremenjeno stanje - logična 1.

Pulzno-širinski modulator za telebane

Za lažje prebijanje ledu smo vam pripravili knjižnični funkciji `pwm1_out(,)` in `pwm2_out(,)`. Nastavitvi prevajalnih razmerij obeh funkcij sta med seboj neodvisni in ju nastavljamo v območju od 0 do 99%, perioda pa je pri obeh funkcijah enaka ter jo lahko spreminjamo v območju od 0 do 65535 ms.

Sintaksi:

```
pwm1_out(perioda_v_ms,prevajalno_razmerje_1_v_%); // primer
uporabe modula PWM1
pwm2_out(perioda_v_ms,prevajalno_razmerje_2_v_%); // primer
uporabe modula PWM2
```

Končno konec teorije

No, sedaj pa napišimo program, ki bo omogočil vrtenje motorja M1 s polovičnimi obrati. Frekvenca preklapljanja naj bo 200 Hz. Za osvežitev spomina naj povemo, da je izhodni priključek motorja M2 na sponki X3, priključek motorja M1 pa na sponki X4. Najprej bo potrebno določiti periodo in prevajalno razmerje PWM signala. Pri preklapljanju s frekvenco 200 Hz znaša perioda:

$$perioda = \frac{1}{frekvenca} = 5ms$$

Če želimo gnati motor s polovičnimi vrtljaji, nastavimo prevajalno razmerje PWM modula na vrednost 50%.

Kaj vse je potrebno pripraviti?

- Priključke mikrokontrolerja RC0, RC2 in RC3 določimo kot izhode, vsi ostali priključki so vhodi.
- Nastavimo periodo in prevajalno razmerje modula PWM1 (uporabimo knjižnično funkcijo `pwm1_out` (perioda, prevajalno razmerje)).
- Poženemo motor s postavitvijo bitov RC0 in RC3 na ustrezni vrednosti (če je RC0 enak 1, potem je RC3 enak 0 ali obratno).

Rešitev problema v programskem jeziku C:

```
// INICIALIZACIJA
void main (void) // IME GLAVNE FUNKCIJE
{
  TRISA = 0B11111111; // PORTA definiramo kot vhod, ga ne
  uporabljamo
  ADCON1 = 0x07; // PORTA definiramo kot digitalna
  vhodno-izhodna vrata
  TRISB = 0B11111111; // PORTB definiramo kot vhod, ga ne
  uporabljamo
  TRISC = 0B11110010; // RC0, RC3, RC2 so izhodi
  TRISD = 0B11111111; // PORTD definiramo kot vhod, ga ne
  uporabljamo
  TRISE = 0B00000111; // PORTE definiramo kot vhod, ga ne
  uporabljamo
  //GLAVNI PROGRAM
  pwm1_out(5000,50); // število vrtljajev je odvisno
  // od prevajalnega razmerja PWM1
```

```
// signala. Izbrana frekvenca znaša
// 200Hz (perioda = 5ms = 5000us);
// prevajalno razmerje znaša 50%
RC0 = 1; // zavrtimo motor
RC3 = 0; // v izbrano smer
stop:
goto stop;
}
```

Samostojno rešite sledeče naloge:

1. Spremenite program tako, da se bo motor vrtel le s četrtno moči.
2. Napišite program, ki bo motor pospeševal iz mirovanja do maksimalne hitrosti v treh stopnjah (mehki zagon).

Pa še nagradna naloga:

Napišite program, ki bo motor počasi zaviral od maksimalne hitrosti vrtljajev do popolne zaustavitve. Nato naj motor pospešuje do maksimalnih vrtljajev, pri tem naj se mu zamenja smer vrtenja. Rešitve pošljite na uredništvo revije Svet elektronike (stik@svet-el.si). Izmed pravih rešitev bomo izžrebali tri nagrajence, ki bodo dobili praktično darilo podjetja ELBACOMP d.o.o. (<http://www.elbacompany.si/>).

Do prihodnje številke

Bodi dovolj za ta mesec. Naslednjič bomo preizkušali delovanje servomotorjev. Dobro bi bilo, da si že ta mesec priskrbite svoj servomotorček, s katerim boste preizkusili delovanje programov. Pobrskajte malo med staro šaro po delavnici, mogoče se bo kakšen pokazal. V nasprotnem primeru pa se obrnite na katerega izmed modelarjev, mogoče boste pri njem našli kakšno idejo. Pa nasvidenje ... ●

Literatura:

- [1] Microchip, PIC16F87x data sheet, Microchip Technology Incorporated, 2001, <http://ww1.microchip.com/downloads/en/DeviceDoc/30292c.pdf>;
- [2] B. Peršič, Gradnja mikroprocesorskih sistemov, Ljubljana: Fakulteta za elektrotehniko, 1998,
- [3] HI-TECH, PICC Lite C Manual, HI-TECH Software, 2002, <http://www.htsoft.com/downloads/manuals.php>;
- [4] B.W. Kernighan, D.M. Ritchie, Programski jezik »C«, Ljubljana: Fakulteta za elektrotehniko, 1991,
- [5] STMicroelectronics, L293 datasheet, push-pull four channel drivers, <http://www.st.com/stonline/books/pdf/docs/1328.pdf>;

Programska oprema:

- [1] Microchip, MPLAB IDE v6.61, <http://ww1.microchip.com/downloads/en/DeviceDoc/mp661.zip>;
- [2] HI-TECH software, PICC lite COMPILER v8.05PL2, <http://www.htsoft.com/products/PICClite.php>;
- [3] Microchip, PIC18F/PIC16F Quick Programmer, http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1824&appnote=en012031;
- [4] Petr Kolomaznik, EHL elektronika, PIC downloader v1.08, <http://www.ehl.cz>;
- [5] Herman Aartsen, TNO - The Netherlands, PIC Bootloader +, <http://www.microchip.com/>

IR Electronic d.o.o.
Ziherlova ulica 2
1000 Ljubljana
Slovenija

Tel.: 01 2835604
Fax: 01 2835605
E-mail: info@irelectronic.si

Več informacij:
e-mail:

A
C
a
t
a
k
@
a
r
r
o
w
s
i
o
v
e
n
i
a
.
c
o
m

dsPIC® DSC PRODUCTS

Product	Program (FLASH) KBytes	Memory (FLASH) KWords	EE Bytes	SRAM Bytes	Packages	A/D 10C
dsPIC30F Motor Control and Power Conversion Controller Family						
dsPIC30F2010	12	4	1024	512	28SOG, 28SPG, 28MMG	
dsPIC30F4011	48	16	1024	2048	40PG, 44PTG, 44MMG	
dsPIC30F4012	48	16	1024	2048	28SOG, 28SPG	
dsPIC30F6010	144	48	4096	8192	80PF	
dsPIC30F General Purpose Controller Family						
dsPIC30F3014	24	8	1024	2048	40PG, 44PT	
dsPIC30F4013	48	16	1024	2048	40PG, 44F	
dsPIC30F5011	66	22	1024	4096	64PT	
dsPIC30F6013	132	44	2048	6144	80F	
dsPIC30F6014	144	48	4096	8192	8	
dsPIC30F Sensor Family						
dsPIC30F2011	12	4	0	1024	18	
dsPIC30F2012	12	4	0	1024	2F	
dsPIC30F3012	24	8	1024	2048		
dsPIC30F3013	24	8	1024	2048		

DsPIC z modificirano Harvard arhitekturo združuje fleksibilnost visoko zmogljivih 16-bitnih krmilnikov s popolno integracijo lastnosti DSP procesorjev, kar nam mogoča visoke hitrosti preračunavanj ob uporabi preprostih lastnosti krmilnika. Z notranjim Flash programskim in podatkovnim spominom omogoča uporabo v širokem spektru zahtevnih aplikacij.

RAZVOJNA ORODJA

Vsem interesentom omogočamo **brezplačno** izposajo razvojnih plošč: DM300016 (2 kosa), DM300014 (2 kosa) in DM300015 (2 kosa)

NOVO !!!

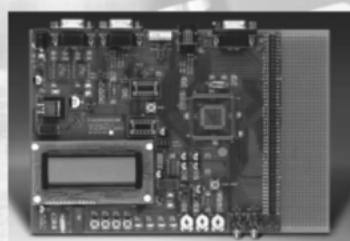
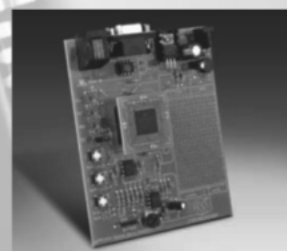


DV164005 ICD2

- razhroščevalnik in programator v enem
- podpira ICSP programiranje
- USB ali serijska komunikacija z osebnim računalnikom
- podpira dsPIC-e in vse Microchip Flash krmilnike

DM300016: dsPICDEM 28-pin Starter demo board

- vsebuje 64-pinski dsPIC30F6012 plug-in modul
- 9V zunanje napajanje
- ICD2 in ICE4000 podpora
- LED, tipke, potenciometer



DM300014:

dsPICDEM 1.1 General Purpose Development Board

- vsebuje dsPIC6014 plug in modul
- 2x UART, CAN, SPI, RS485
- SI3000 kodek
- 122x32 LCD
- ICD2 in ICE4000 podpora
- LED, potenciometer
- temperaturni sensor