

Slika 1 Krmiljenje DC motorjev z orodjem Flowcode (Vir: Lastni viri)

Krmiljenje enosmernih motorjev z mikrokontrolerji programiranimi v »Flowcode« orodju

PETER ZOROJA



SPLOŠNE INFORMACIJE O GRADIVU

Izobraževalni program:

Mehatronika

Ime modula:

Informacijski sistemi 2

Naslov učnih tem ali kompetenc, ki jih obravnava učno gradivo

Uporaba mikroprocesorskih vezij, programiranje v orodju »Flowcode«, izdelava in uporaba algoritmov za krmiljenje enosmernih motorjev, uporaba razvojnega okolja mikrokontrolerjev, programiranje mikrokontrolerjev.

Avtor: Peter Zoroja

Recenzent: Pinosa Matej

Lektor: Petra Gabriel

Datum: julij 2012



To delo je ponujeno pod Creative Commons Priznanje avtorstva-Nekomercialno-Deljenje pod enakimi pogoji 2.5 Slovenija licenco.

Učno gradivo je nastalo v okviru projekta Munus 2. Njegovo izdajo je omogočilo sofinanciranje Evropskega socialnega sklada Evropske unije in Ministrstva za šolstvo in šport.



POVZETEK

Gradivo je namenjeno dijakom 2. letnika smeri tehnik mehatronik. Zajema vsebinski del, naveden v katalogu znanja. Programiranje mikrokontrolerjev poteka v smislu izgradnje diagramov poteka po sistemu "drag and drop". Primeri programov, ki so izdelani v razvojnem okolju Flowcode, verzija 4, so namenjeni Mikrochipovim mikrokontrolerjem oz. razvojnim ploščam, modulom, ki vsebujejo PIC mikrokontrolerje. Gradivo poleg opisa razvojnega okolja »Flowcode« vsebuje opise programskih aplikacij za DC, koračne in servo motorje z ustreznimi vezalnimi shemami krmiljenih periferij. Na koncu posameznih poglavij so vaje, namenjene utrjevanju znanja in razvijanju algoritmičnega razmišljanja.

Ključne besede: mikrokontroler, mikrokrmilnik, algoritem, diagram poteka, programiranje mikrokontrolerja, mikroprocesorska vezja, vhodno-izhodne enote mikrokontrolerja, periferija, instrukcije, parametri, pulzno širinska modulacija (PWM), enosmerni motor, koračni motor, servo motor, konfiguracijska beseda, oscilator, simulacija.



KAZALO

UČNI CILJI	6
MIKROKONTROLER IN OSNOVNI POJMI PROGRAMSKIH STRUKTUR	6
UVOD	6
ALGORITEM	7
DIAGRAM POTEKA (angl. FLOW CHART)	7
OSNOVE UPORABE RAČUNALNIŠKEGA ORODJA FLOWCODE V4	10
NASTAVITVE TIPA MIKROKONTROLERJA IN FREKVENCE OSCILATORJA	10
NASTAVITEV KONFIGURACIJSKE BESEDE	13
SIMULACIJA PROGRAMSKIH APLIKACIJ V FLOWCODE ORODJU	13
VPIS, NALAGANJE PROGRAMSKE APIKACIJE V ECIO MODUL	16
KRMILJENJE ENOSMERNEGA MOTORJA	17
PRIKLOP PERIFERIJE NA MIKROKONTROLER	17
POSTOPEK PROGRAMIRANJA APLIKACIJE ZA KRMILJENJE ENOSMERNEGA MOTORJA (angl. BRUSHED DC MOTOR)	18
VAJE	22
KRMILJENJE SERVO MOTORJA	24
POSTOPEK PROGRAMIRANJA APLIKACIJE ZA KRMILJENJE SERVO MOTORJA	24
VAJE	29
KRMILJENJE KORAČNEGA MOTORJA	31
VAJE	36
VIRI	38



KAZALO SLIK:

<i>Slika 1 Krmiljenje DC motorjev z orodjem Flowcode (Vir: Lastni viri)</i>	<i>0</i>
<i>Slika 2 Začetni in končni blok (Vir: Lastni viri)</i>	<i>7</i>
<i>Slika 3 Prireditveni blok (Vir: Lastni viri).....</i>	<i>8</i>
<i>Slika 4 Odločitveni blok-IF stavek (Vir: Lastni viri).....</i>	<i>8</i>
<i>Slika 5 Vhodni in izhodni blok (Vir: Lastni viri).....</i>	<i>8</i>
<i>Slika 6 Določitev tipa mikrokontrolerja in frekvence oscilatorja (Vir: Lastni viri)</i>	<i>10</i>
<i>Slika 7 Testno vezje z mikrokontrolerjem PIC16F877a in oscilatorjem 4MHz (Vir: Lastni viri)</i>	<i>11</i>
<i>Slika 8 Mikrokontrolerski USB modul ECIO40 (Vir: Lastni viri)</i>	<i>11</i>
<i>Slika 9 Mikrokontrolerski USB modul ECIO28 (Vir: Lastni viri)</i>	<i>12</i>
<i>Slika 10 Nastavitev konfiguracijske besede testnega vezja s PIC16F877a (Vir: Lastni viri).....</i>	<i>13</i>
<i>Slika 11 Simulacija »po korakih« (Vir: Lastni viri).....</i>	<i>14</i>
<i>Slika 12 Funkcije na spremenljivke pri simulaciji »po korakih« (Vir: Lastni viri).....</i>	<i>14</i>
<i>Slika 13 »Tekoča« simulacija (Vir: Lastni viri)</i>	<i>15</i>
<i>Slika 14 Vpis programske aplikacije v ECIO modul (Vir: Lastni viri).....</i>	<i>16</i>
<i>Slika 15 Primer vezalne sheme mikrokontrolerskega sistema za krmiljenje DC motorja (Vir: Lastni viri).....</i>	<i>17</i>
<i>Slika 16 PWM element (Vir: Lastni viri).....</i>	<i>18</i>
<i>Slika 17 Nastavitev frekvence PWM signala (Vir: Lastni viri).....</i>	<i>19</i>
<i>Slika 18 PWM programska aplikacija za krmiljenje enosmernega motorja (Vir: Lastni viri).....</i>	<i>19</i>
<i>Slika 19 Programski blok 1 (Vir: Lastni viri)</i>	<i>20</i>
<i>Slika 20 Programski blok 2 (Vir: Lastni viri)</i>	<i>20</i>
<i>Slika 21 Programski blok 3 (Vir: Lastni viri)</i>	<i>21</i>
<i>Slika 22 Vezalna shema ECIO28 modula s periferijo (Vir: Lastni viri)</i>	<i>22</i>
<i>Slika 23 Diagram poteka aplikacije in prikaz nastavitve vhodne periferije (Vir: Lastni viri).....</i>	<i>23</i>
<i>Slika 24 Servo motor (Vir: Lastni viri)</i>	<i>24</i>



<i>Slika 25 Servo element (Vir: Lastni viri)</i>	<i>25</i>
<i>Slika 26 Primer programska aplikacija za krmiljenje servo motorja (Vir: Lastni viri)</i>	<i>26</i>
<i>Slika 27 Programski blok 1 (Vir: Lastni viri)</i>	<i>27</i>
<i>Slika 28 Programski blok 3 (Vir: Lastni viri)</i>	<i>27</i>
<i>Slika 29 Programski blok 4 (Vir: Lastni viri)</i>	<i>28</i>
<i>Slika 30 Izhodna periferija mikrokontrolerja (Vir: Lastni viri)</i>	<i>31</i>
<i>Slika 31 Koračni motor (Vir: Lastni viri)</i>	<i>32</i>
<i>Slika 32 Element koračnega motorja (Vir: Lastni viri)</i>	<i>33</i>
<i>Slika 33 Primer programske aplikacije za krmiljenje koračnega motorja (Vir: Lastni viri)</i>	<i>34</i>
<i>Slika 34 Programski blok 1 (Vir: Lastni viri)</i>	<i>35</i>
<i>Slika 35 Programski blok 3 (Vir: Lastni viri)</i>	<i>35</i>



UČNI CILJI

Dandanes si življenja brez programljivih elementov ne predstavljamo. Skoraj vsako napravo, motor, ki ga dnevno uporabljamo, krmili kateri od mikrokontrolerjev, ki skrbi za njegovo pravilno delovanje. V različnih procesih se stremi k zmanjševanju stroškov dela in k čim večji kakovosti izdelkov, fleksibilnosti, reprogramljivosti naprave ... Zaradi tega sta poznavanje osnov programiranja mikroprocesorskih naprav in znanje algoritmičnega razmišljanja postala osnovni sestavini funkcionalne pismenosti mehatronika, elektronika in računalničarja.

Cilji:

- Spoznavanje delovanja mikroprocesorskih vezij;
- poznavanje in uporaba razvojnega okolja FLOWCODE in izdelava programov;
- načrtovanje in izdelava mikroprocesorskih vezij;
- razvijanje algoritmičnega razmišljanja;
- načrtovanje in izdelava programov za mikroprocesorje;
- programiranje mikroprocesorskih vezij;
- izvajanje krmiljenja motorjev, krmiljenje motorjev z mikroprocesorskimi veziji;
- uporaba znanih rešitev na novih primerih programov za mikroprocesorje;
- razvijanje sposobnosti ustvarjanja;
- iskanje informacij za obstoječe in nove rešitve v programih.



MIKROKONTROLER IN OSNOVNI POJMI PROGRAMSKIH STRUKTUR

UVOD

Mikrokontroler ali mikrokontroler je integrirano vezje, čip, ki vsebuje skoraj vse sestavine mikroročunalnika (procesor, notranji pomnilnik, vmesnike ...). Da bi bil mikrokontroler popoln mikroročunalnik, mu manjkajo le vhodno-izhodne enote (tipke, senzorji, elektromotorji, žarnice ...), ki niso primerne za vgradnjo v čip. Mikrokontrolerje srečamo v večini modernih elektronskih naprav. Na primer v mobilnem telefonu, televiziji, DVD-predvajalniku, mikrovalovni pečici,



pralnem in pomivalnem stroju ... V sodobnem avtomobilu in osebni računalniku jih je cela množica. Razlog za to je v njihovi univerzalni uporabnosti in cenenosti.

Najpomembnejša sestavina mikrokrmilnika je procesor. Vsi mikrokrmilniki, ki so nadgradnja istega procesorja, tvorijo družino mikrokrmilnikov. Med seboj se predstavniki družine razlikujejo po vrsti in količini dodanih enot. Procesorji v mikrokrmilniku se razlikujejo v številu bitov, ki jih obdelujejo hkrati, in v hitrosti delovanja oziroma taktu. Največkrat so osembitni in »tečejo« s hitrostjo nekaj MHz. V primerjavi z glavnim procesorjem v osebni računalniku je to malo, vendar v večini primerov zadostuje.

ALGORITEM

Algoritem je navodilo, s katerim rešujemo določen problem. Običajno je zapisan kot seznam korakov, ki nas pripeljejo do rešitve problema. V osnovi mora algoritem izpolnjevati sledeče pogoje:

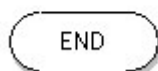
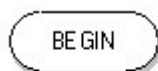
- Biti mora natančen.
- Biti mora nedvoumen.
- Enolično mora določati zaporedje korakov, s katerimi dosežemo zadani cilj.
- Algoritem se mora izvesti v končnem številu korakov – biti mora končen.

V računalništvu algoritem predstavlja načrt, po katerem pripravimo računalniški program. Algoritem lahko predstavimo na različne načine. Ena od pogostejših oblik je predstavitev v grafični obliki z diagramom poteka. Lahko pa algoritem predstavimo ustno ali pisno v psevdojeziku in grafično z različnimi diagrami.

DIAGRAM POTEKA (angl. FLOW CHART)

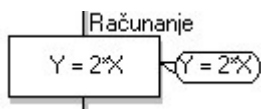
Gradniki diagrama poteka so grafični simboli. Imamo pet osnovnih simbolov:

- Začetni ali končni blok. S tem blokom označimo začetek ali konec postopka.



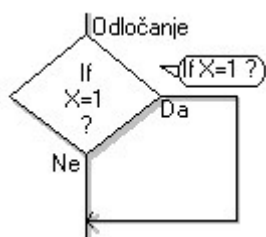
Slika 2 Začetni in končni blok (Vir: Lastni viri)

- Prireditveni blok ali procesni blok. V tem bloku se izvajajo različne prireditvene, matematične operacije.



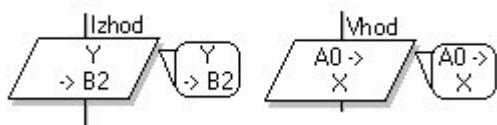
Slika 3 Prireditveni blok (Vir: Lastni viri)

1. Odločitveni blok. Ta blok omogoča vejitve v algoritmu, uporabljamo ga tudi pri zankah, ko se določen del algoritma lahko večkrat ponovi.



Slika 4 Odločitveni blok-IF stavek (Vir: Lastni viri)

- Vhodni ali izhodni blok. S tem blokom zagotavljamo interakcijo algoritma z okoljem, uporabnikom ali algoritmi.



Slika 5 Vhodni in izhodni blok (Vir: Lastni viri)



FLOWCODE

FLOWCODE je komercialna programska oprema za programiranje mikrokontrolerjev, ki ga je razvilo podjetje Matrix Multimedia.

Je eden izmed najbolj naprednih grafičnih jezikov za programiranje mikrokontrolerjev. Njegova največja prednost pred ostalimi je, da omogoča programiranje kompleksnih sistemov tudi tistim, ki se šele spoznavajo z mikrokontrolerji. Preveden je v več kot 20 jezikov, tudi v slovenščino. Na voljo je za PIC, AVR in ARM mikrokontrolerje.

Prednosti:

- Začetnikom omogoča hitro in enostavno programiranje kompleksnih sistemov.
- Dostopen je za različne mikrokontrolerje.
- Podpora za elemente E-Block, Formula Flowcode, ECIO.

Značilnosti:

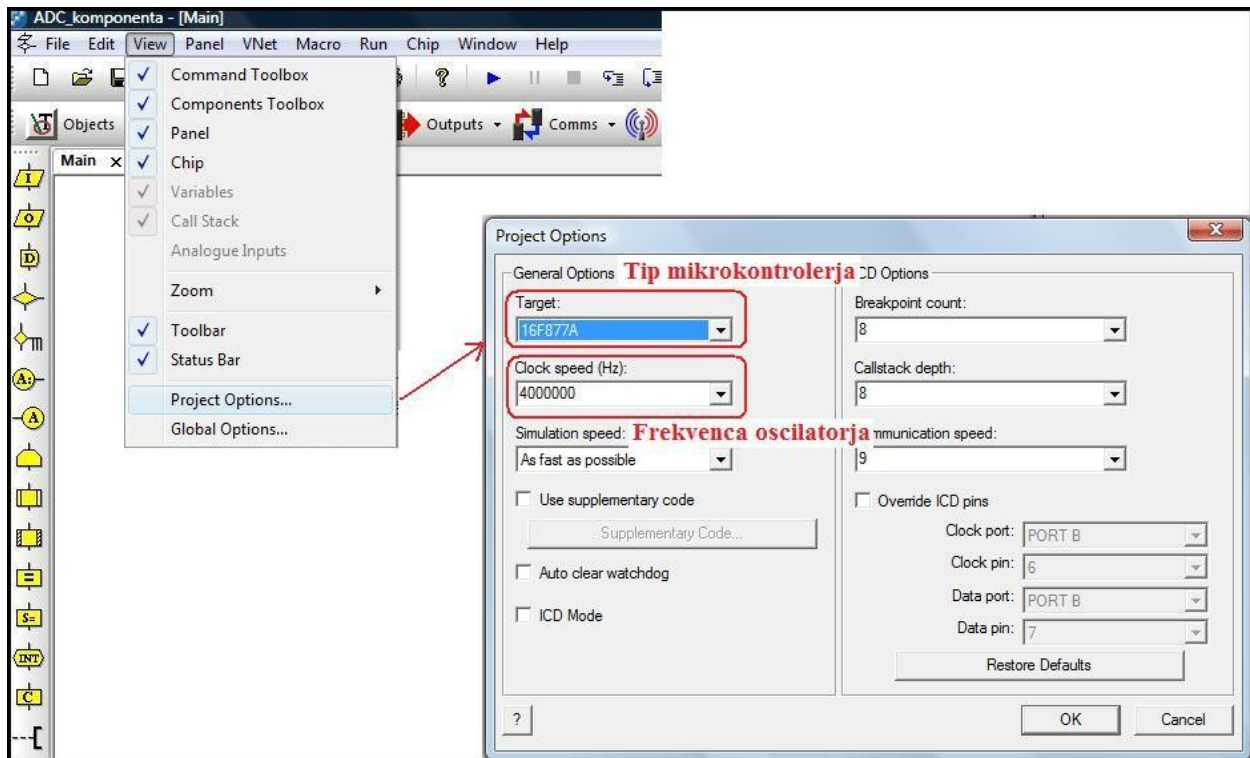
- enostaven uporabniški vmesnik, gradnja diagrama poteka po sistemu "drag and drop";
- vsebuje visokonivojske podrutine za strojne komponente;
- sistem za načrtovanje, simuliranje, "debugging" in nalaganje programa;
- odprta arhitektura omogoča vpogled v generirani C in ASM kodi;
- odlična tehnična podpora z dokumentacijo za učenje in razvoj "embedded" sistemov.



OSNOVE UPORABE RAČUNALNIŠKEGA ORODJA FLOWCODE V4

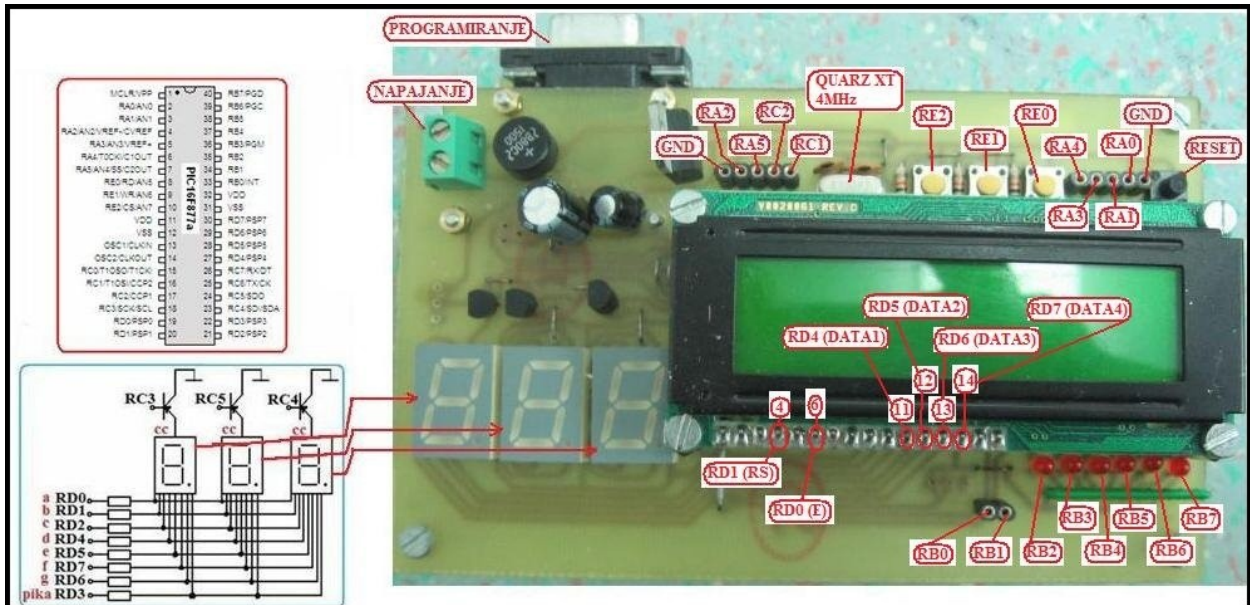
NASTAVITVE TIPA MIKROKONTROLERJA IN FREKVENCE OSCILATORJA

Ob odprtju novega okna, projekta v orodju Flowcode, moramo nastaviti tip mikrokontrolerja in frekvenco oscilatorja, pod katero deluje mikrokontroler.



Slika 6 Določitev tipa mikrokontrolerja in frekvence oscilatorja (Vir: Lastni viri)

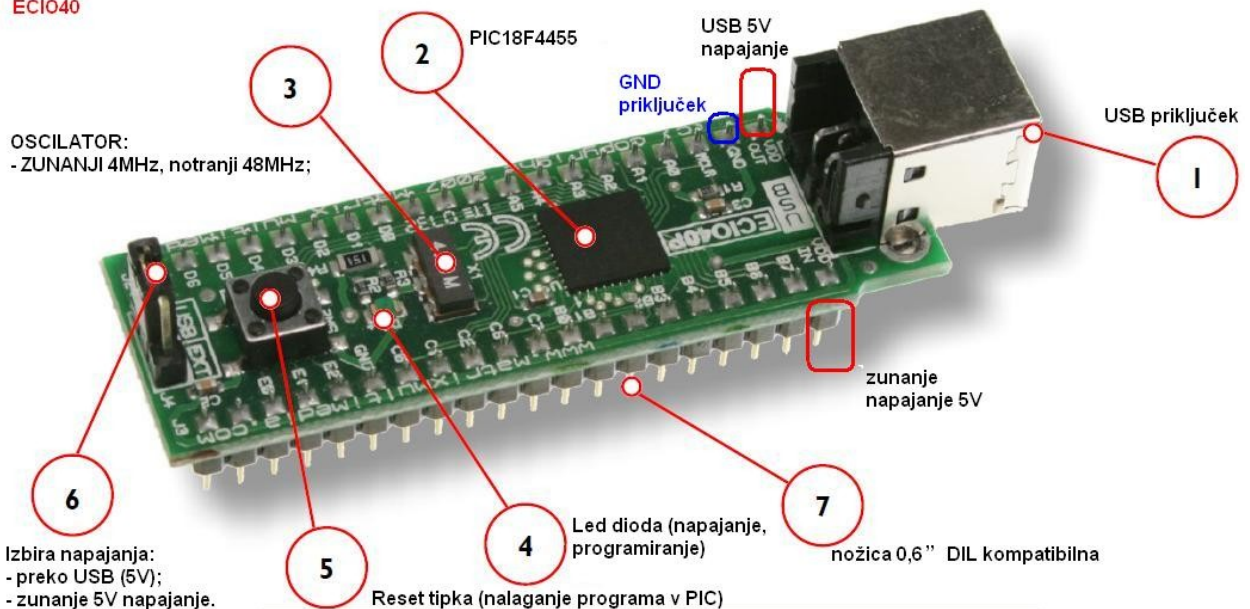
Testno vezje, ki smo ga projektirali na TŠC Nova Gorica, vsebuje mikrokontroler PIC16f877a in zunanji oscilator Quartz frekvence 4 MHz. Nastavitve s slike 6 se nanašajo na ta dejstva.



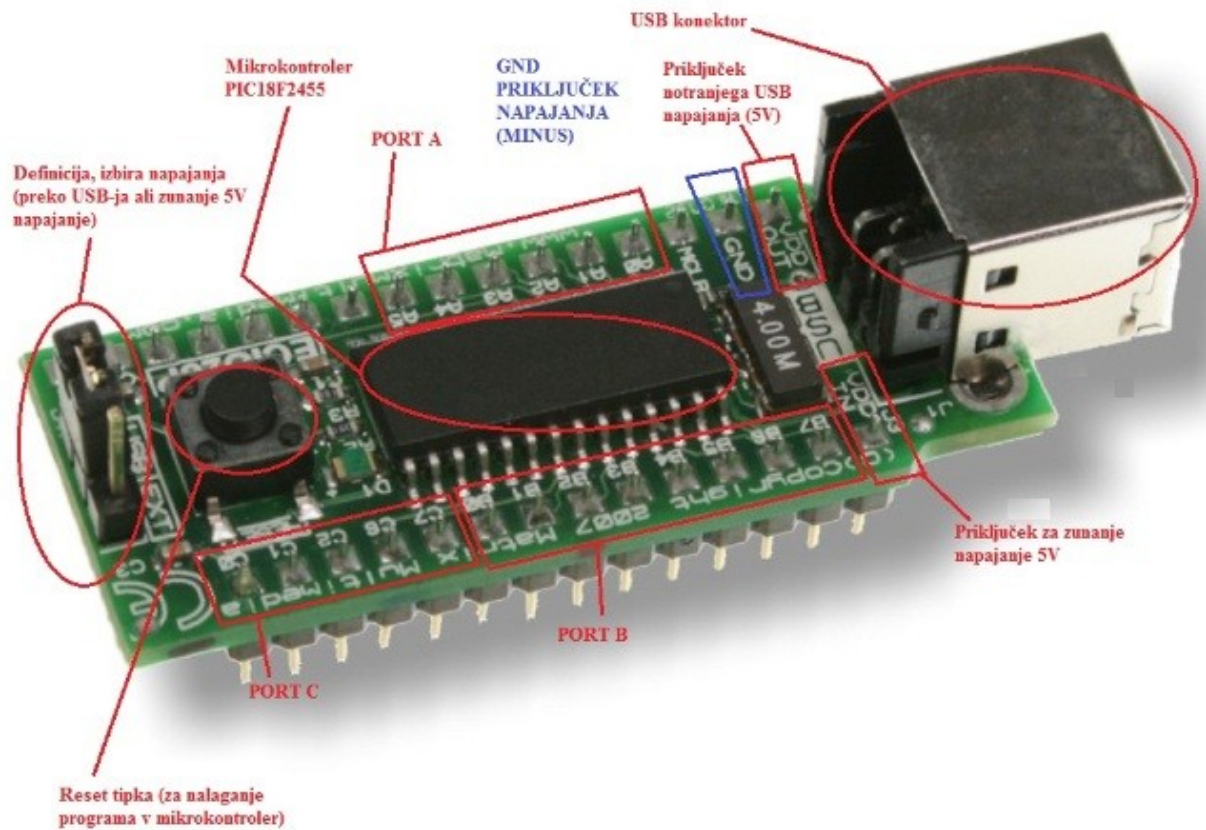
Slika 7 Testno vezje z mikrokontrolerjem PIC16F877a in oscilatorjem 4MHz (Vir: Lastni viri)

V primeru, da uporabljamo pri aplikacijah USB mikrokontrolerske module podjetja Matrix Multimedia (ECIO28, ECIO40), se v oknu nastavitve izbire samo ustrezen modul. Frekvenca oscilatorja za ECIO module je avtomatsko nastavljena na 48 MHz.

ECIO40



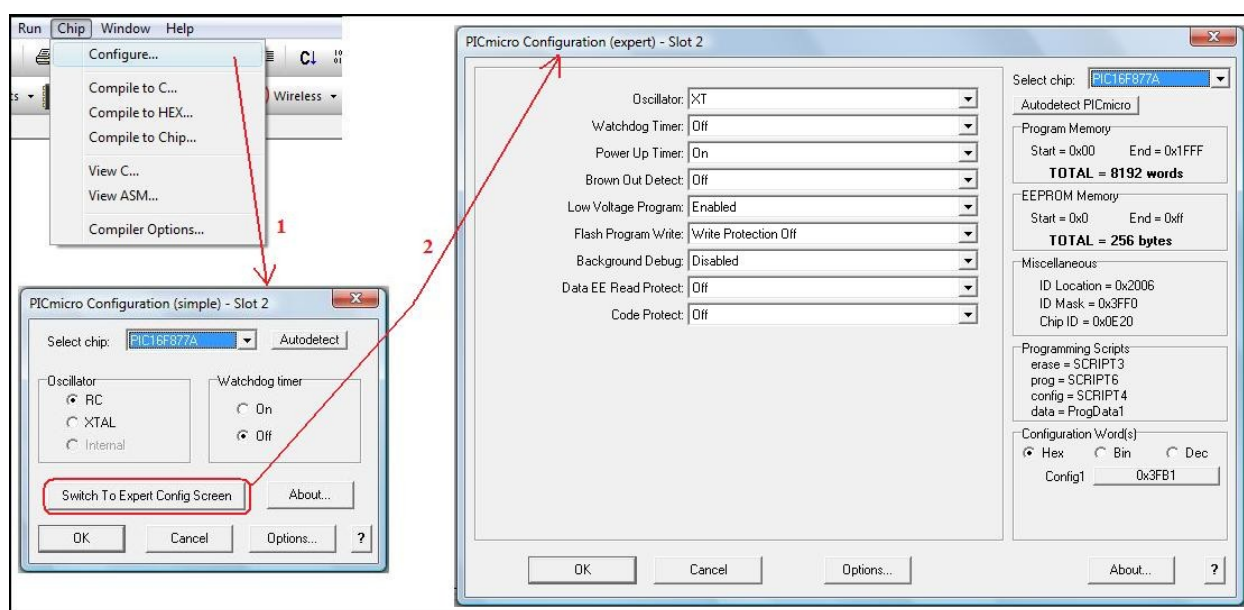
Slika 8 Mikrokontrolerski USB modul ECIO40 (Vir: Lastni viri)



Slika 9 Mikrokontrolerski USB modul ECIO28 (Vir: Lastni viri)

NASTAVITEV KONFIGURACIJSKE BESEDE

V oknu t. i. konfiguracijske besede nastavimo ostale lastnosti, pod katerimi deluje izbrani mikrokontroler (tip mikrokontrolerja, način programiranja, zaščita programske aplikacije ...). Podatke o tem, kaj pomeni posamezen parameter v konfiguracijski besedi, dobimo v katalogu mikrokontrolerja. Katalog izbranega mikrokontrolerja je najenostavneje pridobiti na spletu. V iskalniku Google napišemo besedo Datasheet in oznako mikrokontrolerja. Tako nam je katalog mikrokontrolerjana na voljo (primer: Datasheet PIC16F877a). Moduloma ECIO28 in ECIO40 pa konfiguracijske besede ni potrebno nastavljati oz. jo je definiral proizvajalec, podjetje Matrix Multimedia, ki je tako poenostavilo delo s tema dvema moduloma.



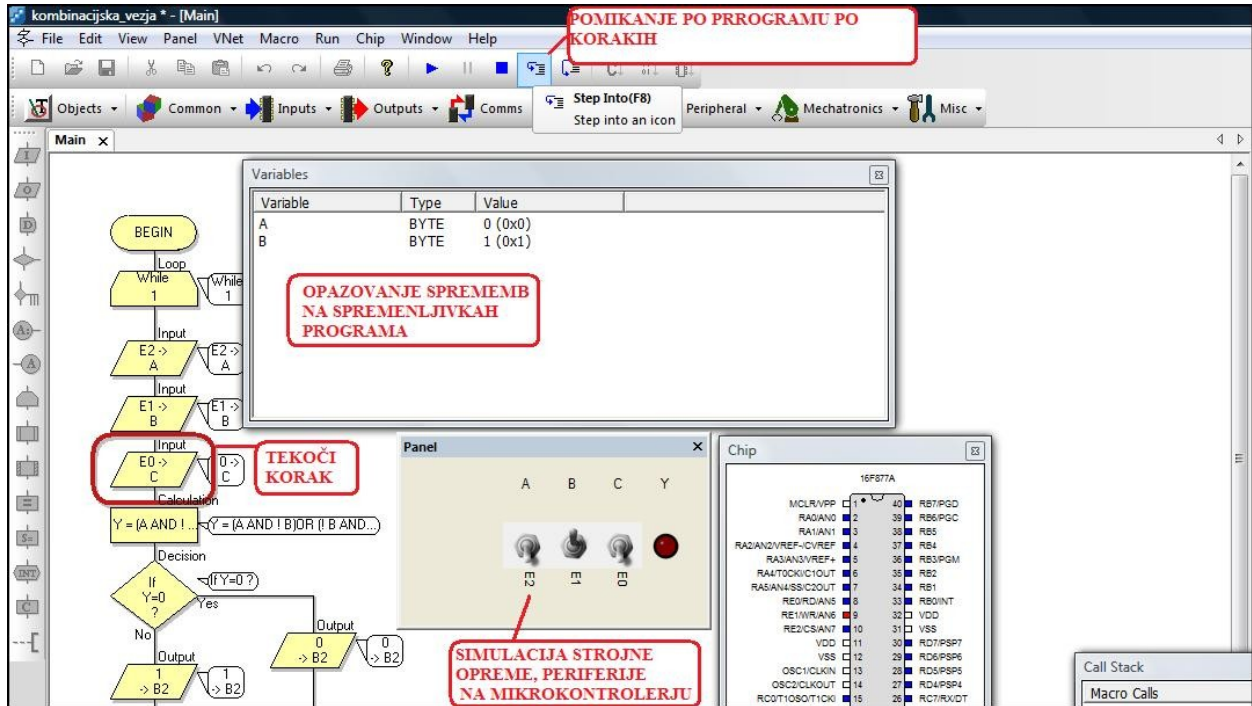
Slika 10 Nastavitev konfiguracijske besede testnega vezja s PIC16F877a (Vir: Lastni viri)

SIMULACIJA PROGRAMSKIH APLIKACIJ V FLOWCODE ORODJU

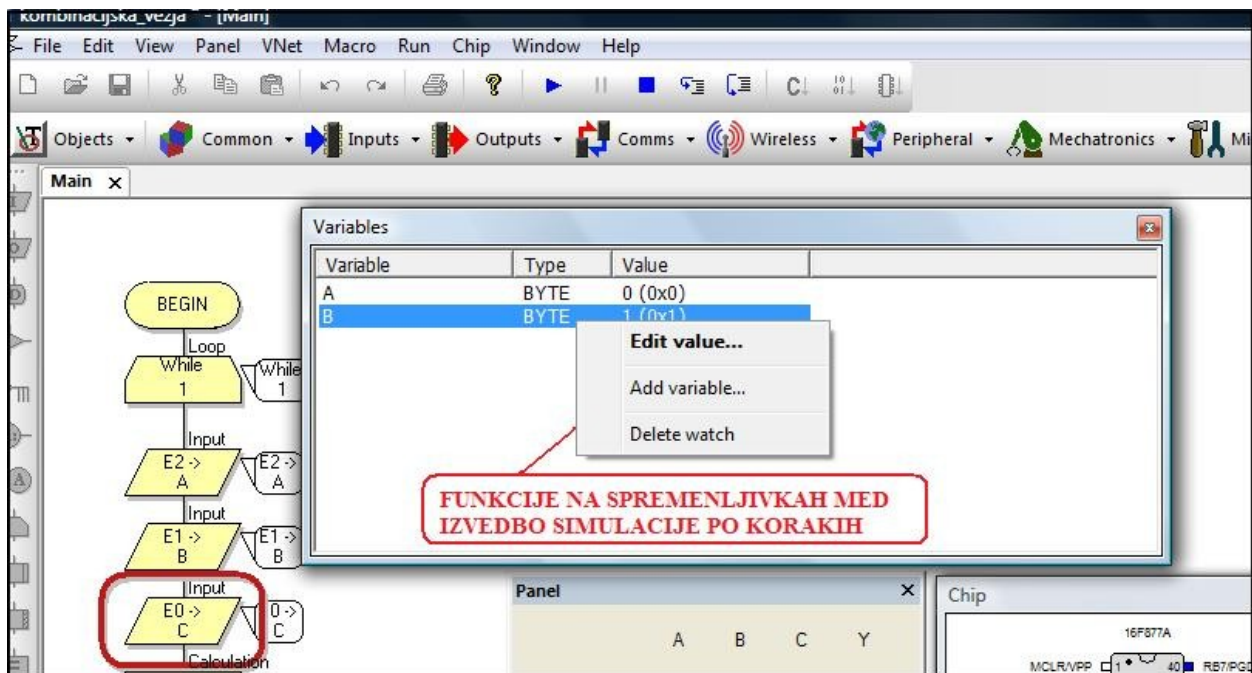
V orodju Flowcode lahko v osnovi izvajamo dva tipa simulacije delovanja programske aplikacije:

- Simulacija aplikacije »po korakih«;
- »tekoča« simulacija aplikacije.

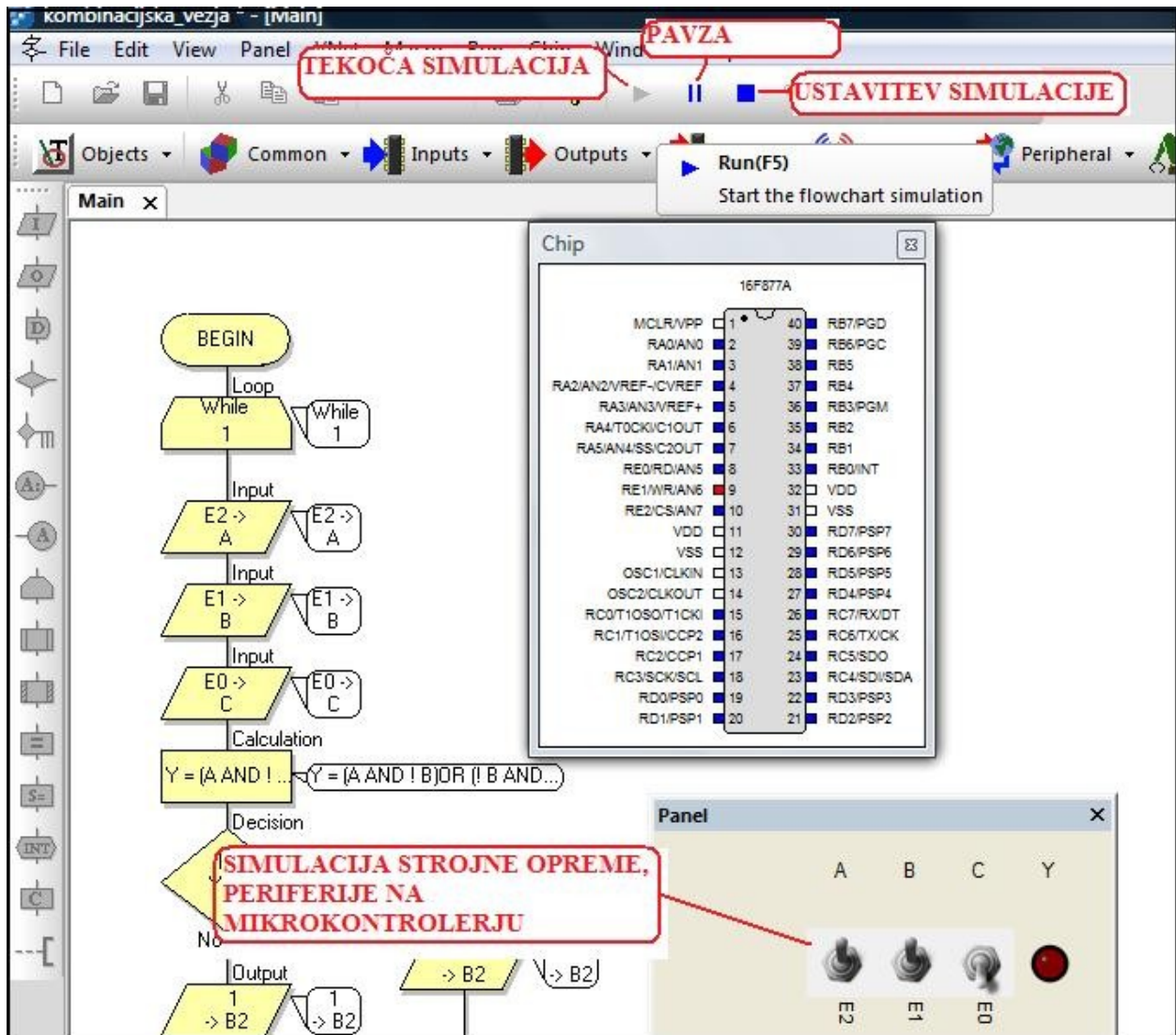
V simulaciji aplikacije »po korakih« lahko opazujemo dogajanje, reakcijo, ki jo sproži posamezen blok oz. instrukcija v aplikaciji. »Tekoča« simulacija pa nam prikaže dogajanje med periferijo in mikrokontrolerjem, kot jo zazna človek, okolica ob delovanju mikrokontrolerske naprave.



Slika 11 Simulacija »po korakih« (Vir: Lastni viri)



Slika 12 Funkcije na spremenljivke pri simulaciji »po korakih« (Vir: Lastni viri)



Slika 13 »Tekoča« simulacija (Vir: Lastni viri)



VPIS, NALAGANJE PROGRAMSKE APIKACIJE V ECIO MODUL

Za naložitev programske aplikacije v modul ECIO je predhodno potrebno namestiti gonilnike za module ECIO. Gonilniki se nahajajo na spletni strani podjetja Matrix Multimedia.

Postopek vpisa programske aplikacije v mikrokontroler:

- modul ECIO povežemo na USB-vrata PC-ja;
- s klikom na ukaz »Prevedi v mikrokontroler« sprožimo proces vpisa;
- po pojavu sporočila na ekranu o »resetiranju« mikrokontrolerja

aktiviramo prenos kode v mikrokontroler s tipko na modulu ECIO.



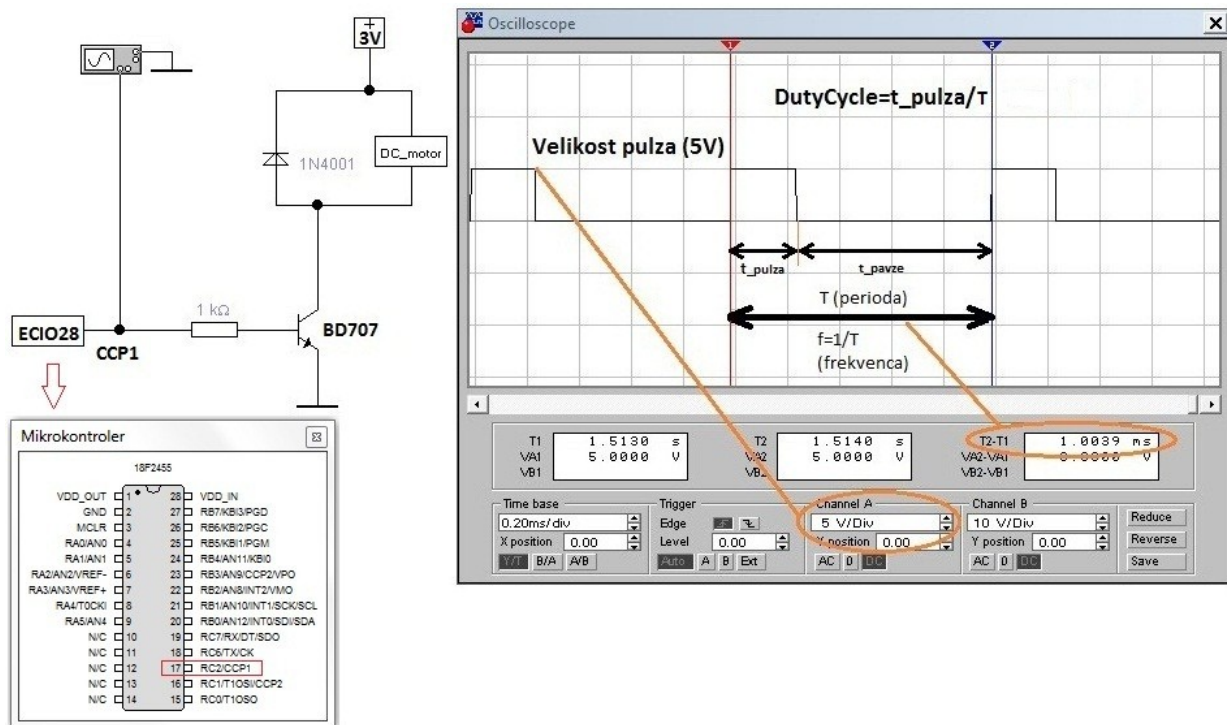
Slika 14 Vpis programske aplikacije v ECIO modul (Vir: Lastni viri)



KRMILJENJE ENOSMERNEGA MOTORJA

PRIKLOP PERIFERIJE NA MIKROKONTROLER

V vaji je uporabljen mikrokontrolerski modul ECIO28, na katerega je preko nožice CCP1 priključena periferija (tranzistor kot stikalo v vezavi CE-skupni emitor), ki posredno upravlja enosmerni motor. V vaji bi lahko namesto nožice CCP1 modula ECIO uporabili nožico CCP2.



Slika 15 Primer vezalne sheme mikrokontrolerskega sistema za krmiljenje DC motorja (Vir: Lastni viri)

POSTOPEK PROGRAMIRANJA APLIKACIJE ZA KRMILJENJE ENOSMERNEGA MOTORJA (angl. BRUSHED DC MOTOR)

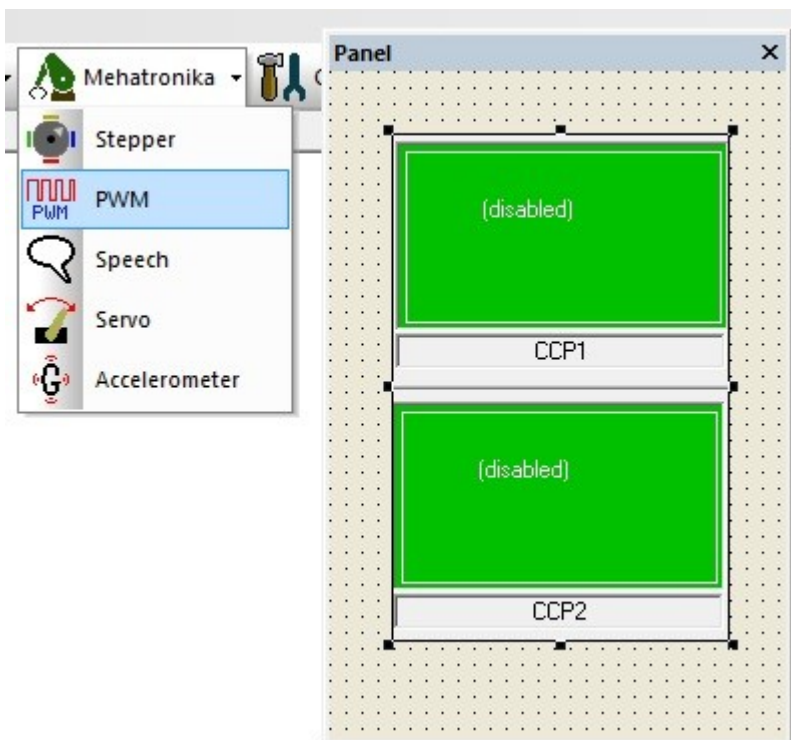
S programsko aplikacijo generiramo na priključku CCP t. i. signal PWM. Večji, kot je v signalu parameter »DutyCycle«, več energije pridobi motor in s tem se motor posledično hitreje vrti. Enosmerni motor ima dva priključka, ki sta vezana na navitje v rotorju motorja. Rotor oz. navitje na njem se nahaja v permanentnem magnetnem polju, ki ga ustvarja magnetni oklep oz. ogrodje (stator). Ob priključitvi enosmerne napetosti steče tok skozi navitje motorja. Posledično deluje sila (sila Biosavartovega zakona v magnetnem polju) na rotor in ta se začne obračati.

Prikaz realizacije programske aplikacije, ki krmili enosmerni motor z naslednjimi zahtevami:

- Frekvenca signala PWM na CCP1 je 10 kHz.
- Parameter DutyCycle je 75 %.

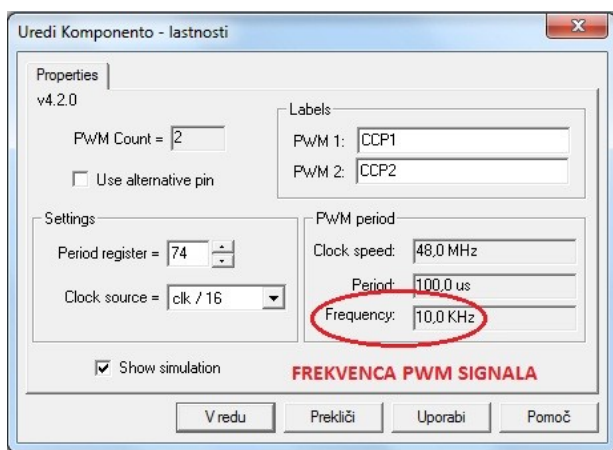
Razlaga rešitve:

- V orodni vrstici Mehatronika se pokaže več komponent. Izberemo modul PWM, kliknemo nanj in na »Panelu« se pojavi element PWM.



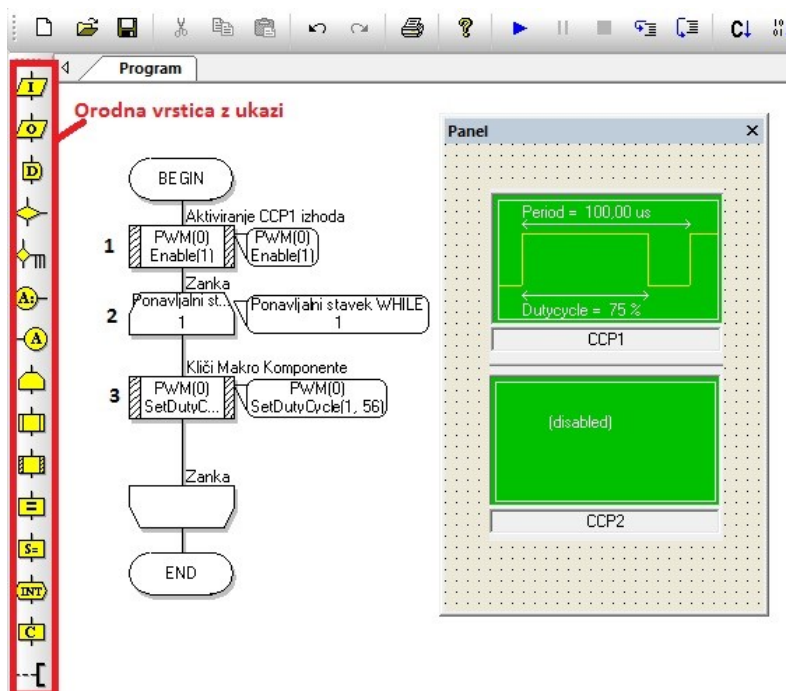
Slika 16 PWM element (Vir: Lastni viri)

- Z miško se pomaknemo na element PWM in z desnim klikom aktiviramo meni tega elementa. Kliknemo na lastnosti in odpre se nam okno »Uredi komponento – Lastnosti«. V tem oknu nastavimo parametra Clock source in Period register tako, da se parameter Frekvency v oknu spremeni na vrednost 10 kHz. To je način, kako nastavljamo frekvenco signala na nožicah CCP.



Slika 17 Nastavitev frekvence PWM signala (Vir: Lastni viri)

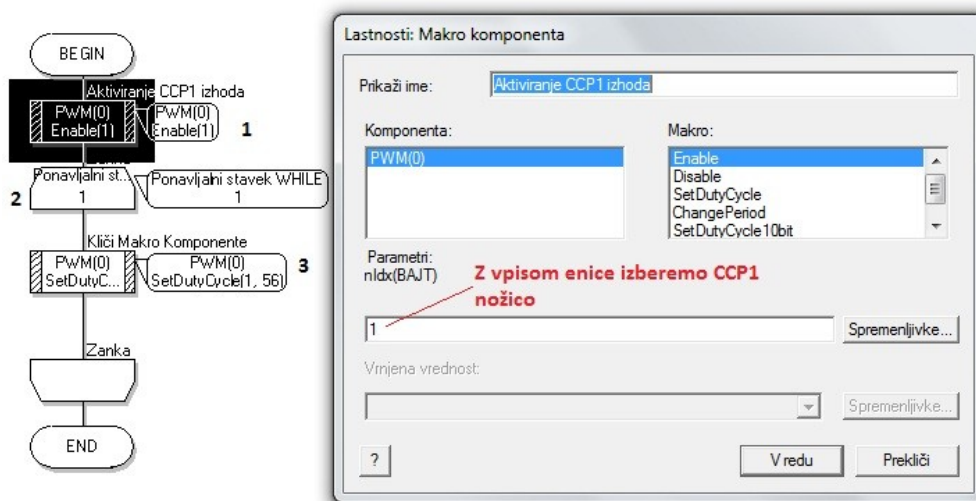
- Sledi pisanje programa po sistemu "drag and drop". Iz orodne vrstice z ukazi povlečemo ustrezne bloke in sestavimo diagram poteka.



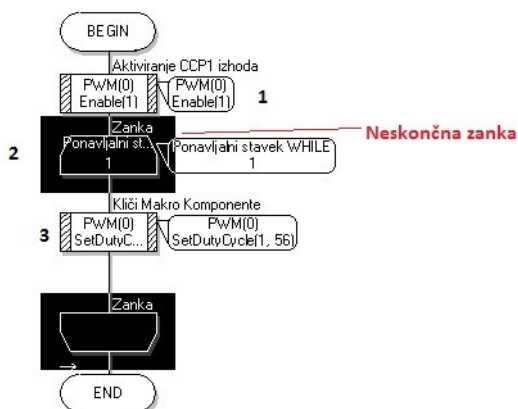
Slika 18 PWM programska aplikacija za krmiljenje enosmernega motorja (Vir: Lastni viri)

RAZLAGA DIAGRAMA POTEKA V APLIKACIJI KRMILJENJE ENOSMERNEGA MOTORJA

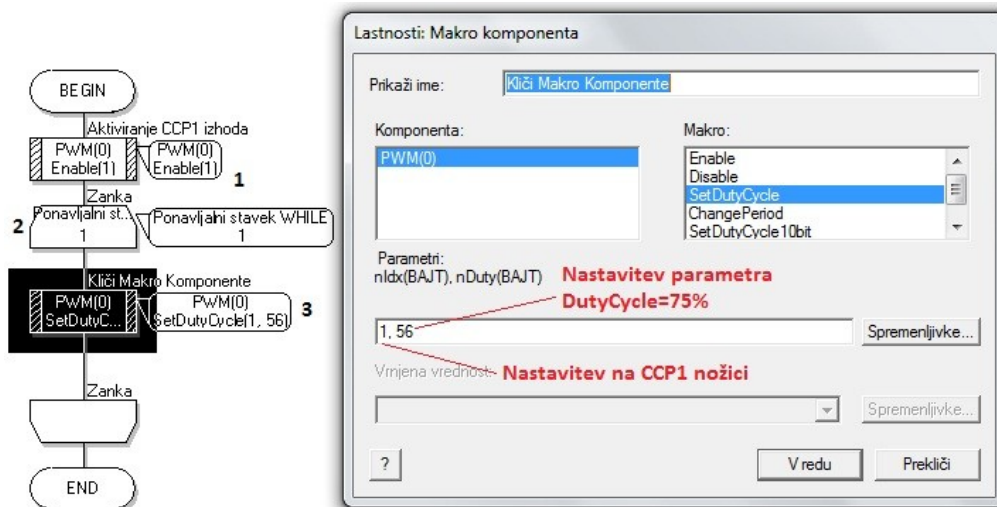
Aplikacija se začne z aktivacijo signala PWM na nožici CCP1 (programski blok 1). V naslednjem bloku program vstopi v neskončno zanko (programski blok 2). Programski blok 3 na nožici CCP1 nastavi »Duty Cycle« na 75 %. 75 % nastavitve parametra »Duty Cycle« nastavimo s številko 56.



Slika 19 Programski blok 1 (Vir: Lastni viri)



Slika 20 Programski blok 2 (Vir: Lastni viri)



Slika 21 Programski blok 3 (Vir: Lastni viri)

Število 56 izračunamo s »križnim« procentualnim izračunom. Na sliki 16 je nastavljen parameter »Period register« na 74 oz. to pomeni, da je parameter »Duty Cycle« 100 %. Procentualna pretvorba nam pokaže, da številu 56 ustreza parameter »Duty Cycle« vrednosti 75 %. Po zagonu »tekoče« simulacije se v oknu elementa PWM pod CCP1 pojavi zelen oscilogram.

Element PWM signala vsebuje pet macro komponent:

- Enable,
- Disable;
- SetDutyCycle;
- ChangePeriod;
- SetDutyCycle 10bit.

Enable »macro« komponenta aktivira PWM na CCP1, če je vpisana enica v prazno okence oz. aktivira PWM na CCP2, če je vpisana dvojka v prazno okence. **Disable** »macro« komponenta deaktivira PWM na CCP1 oz. na CCP2. **SetDutyCycle** »macro« komponenta določa Duty Cycle PWM signala. Spremenljivka, ki določa Duty Cycle pri tej komponenti, je tipa byte (256 vrednosti). **ChangePeriod** »macro« komponenta določa periodo oz. frekvenco PWM signala. Spremenljivki, ki določata periodo pri tej komponenti sta tipa byte. Prva spemenljivka (Period Register) lahko zavzame vrednosti med 0 in 255. Druga spremenljivka (prescalar value) lahko zavzame vrednosti ena, štiri ali šestnajst. **SetDutyCycle 10bit** »macro« komponenta določa Duty Cycle PWM signala. Spremenljivka, ki določa Duty Cycle pri tej komponenti, je tipa integer (1024 vrednosti).



VAJE

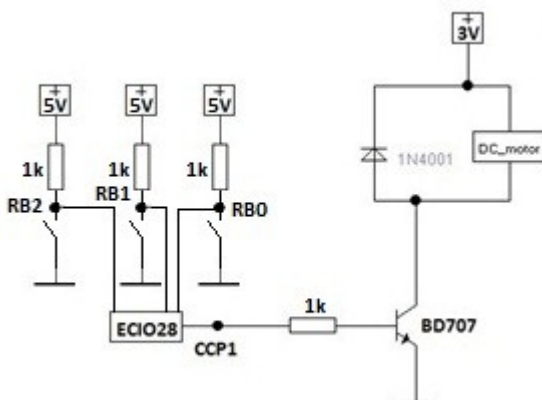
1. Izdelaj programsko aplikacijo za krmiljenje enosmernega motorja za modul ECIO28 s sledečimi zahtevami:
 - Uporabi CCP2 nožico;
 - parameter DutyCycle=61%.
2. Kateri motorji so »Brushed DC« motorji in katere so poglavitne značilnosti teh motorjev? Poišči ustrezen slovenski prevod za to vrsto motorjev!

Opomba: Vprašanje razišči sam ali s pomočjo učitelja angleščine in učitelja, pri katerem spoznaš električne elemente.

3. Kateri motorji so »BCDL Brushless« motorji in katere so poglavitne značilnosti teh motorjev? Poišči ustrezen slovenski prevod za to vrsto motorjev!

Opomba: Vprašanje razišči sam ali s pomočjo učitelja angleščine in učitelja, pri katerem spoznaš električne elemente.

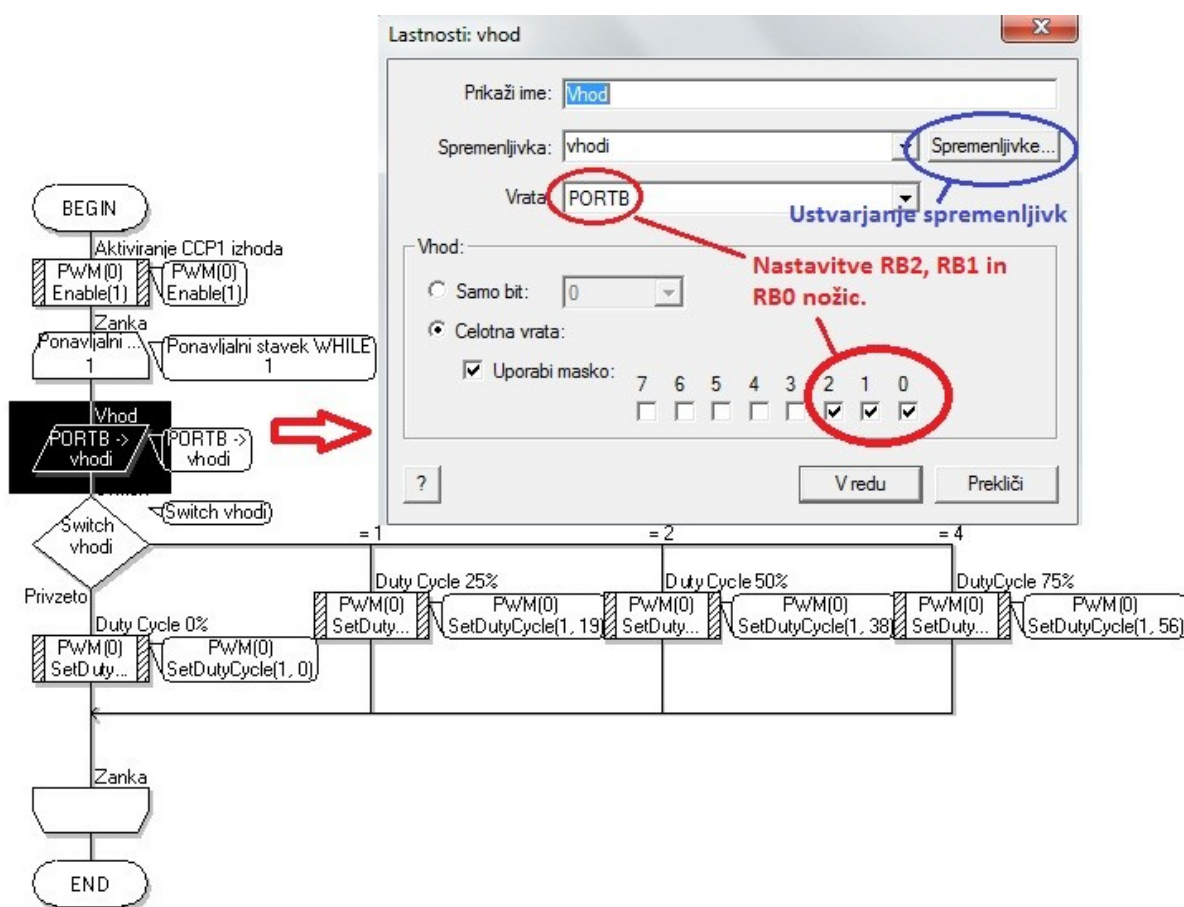
4. Izdelaj programsko aplikacijo za krmiljenje enosmernega motorja z modulom ECIO28 po naslednjem diagramu poteka oz. z upoštevanjem naslednjih zahtev in vezalne sheme vezja z mikrokontrolerjem:



Slika 22 Vezalna shema ECIO28 modula s periferijo (Vir: Lastni viri)

Zahteve:

- Frekvenca signala PWM na CCP1 je 10 kHz.
- V programski aplikaciji ustvari spremenljivko »vhodi« (byte spremenljivka). V spremenljivko »vhodi« naj se shranjujejo logična stanja tipk na nožicah RB0, RB1 in RB2. To realiziraj z vhodnim programskim blokom (glej sliko 23). V odvisnosti od spremenljivke »vhodi« aktiviraj veje v »Switch stavku« tako, da je parameter »Duty Cycle« v vsaki veji definiran drugače (0 %, 25 %, 50 %, 75 %).

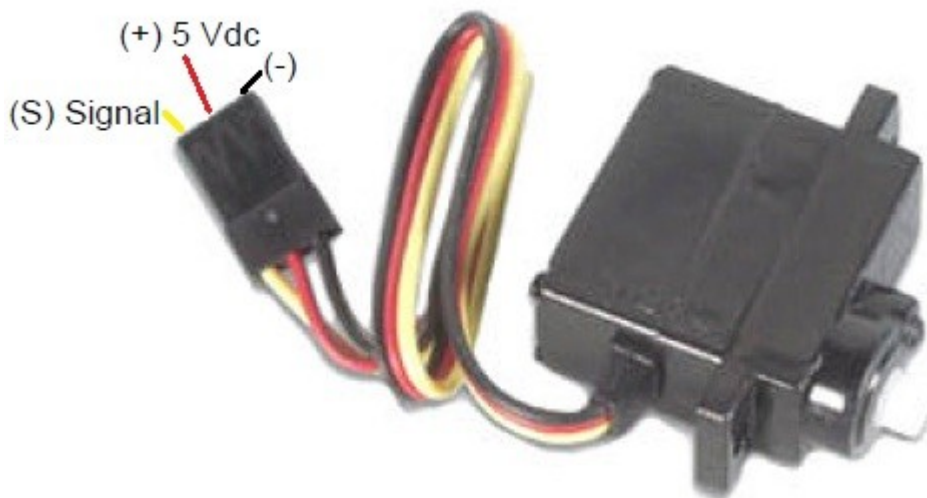


Slika 23 Diagram poteka aplikacije in prikaz nastavitve vhodne periferije (Vir: Lastni viri)



KRMILJENJE SERVO MOTORJA

V vaji je uporabljen mikrokontrolerski modul ECIO28 in servo motor. Mikrokontroler krmili servo motor s signalom preko nožice RB0. Lahko bi bila izbrana katerakoli nožica na vratih mikrokontrolerja (vrata A, vrata B, vrata C).



Slika 24 Servo motor (Vir: Lastni viri)

POSTOPEK PROGRAMIRANJA APLIKACIJE ZA KRMILJENJE SERVO MOTORJA

Servo motor ima tri priključke. Dva sta namenjena napajanju servo motorja (rdeča in črna žica), tretji priključek (rumena ali bela žica) pa je namenjen za krmilni signal PWM. Časovni interval krmilnega pulza v tem signalu določa, za kolikšen kot se bo os motorja obrnila.

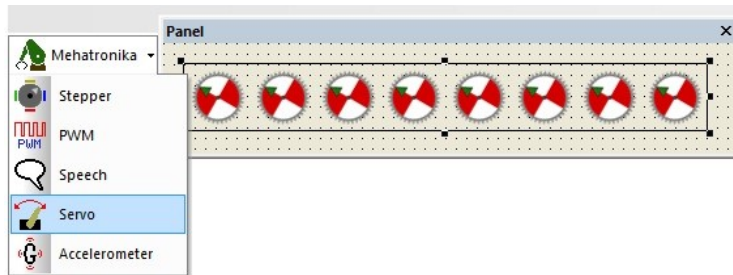
Prikaz realizacije programske aplikacije, ki krmili servo motor z naslednjimi zahtevami:

1. Os servo motorja naj opravi obrat za kot 45° , v naslednji sekundi naj se os povrne v začetni položaj.
2. Zgoraj opisano dogajanje naj se sekvenčno ponavlja v neskončnost v časovnem intervalu dveh sekund.



Razlaga rešitve:

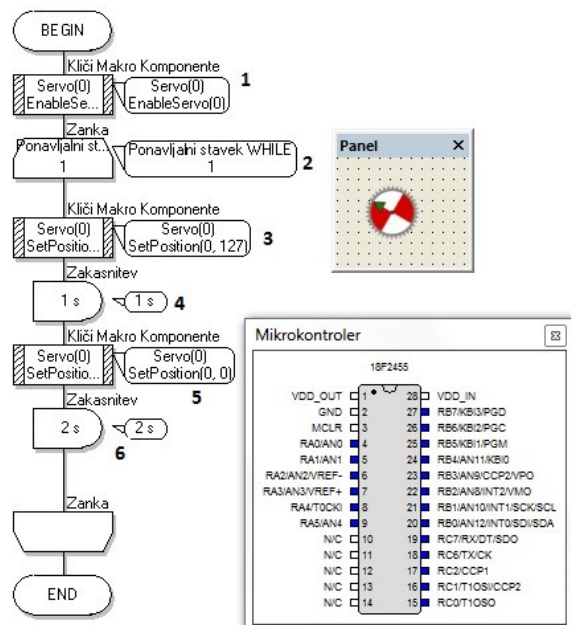
- V orodni vrstici pod Mehatronika izberemo med različnimi elementi servo modul. Kliknemo nanj in na »Panelu« se pojavi servo element.



Slika 25 Servo element (Vir: Lastni viri)

Z miško se pomaknemo na Servo element in z desnim klikom aktiviramo seznam tega elementa. Kliknemo na zunanje lastnosti in odpre se nam okno »Uredi komponento - Lastnosti«. V tem oknu nastavimo parameter »Number of channels« na 1. S tem definiramo, da bomo programirali en servo motor. Z aktiviranjem te spremembe se nam zaslonu pojavi en simbol rotirajočega dela servo motorja. S ponovnim desnim klikom na Servo elementu ponovno aktiviramo seznam tega elementa in izberemo aktivnost povezave. S klikom na povezave definiramo signalno informacijo za servo motor na nožici RB0.

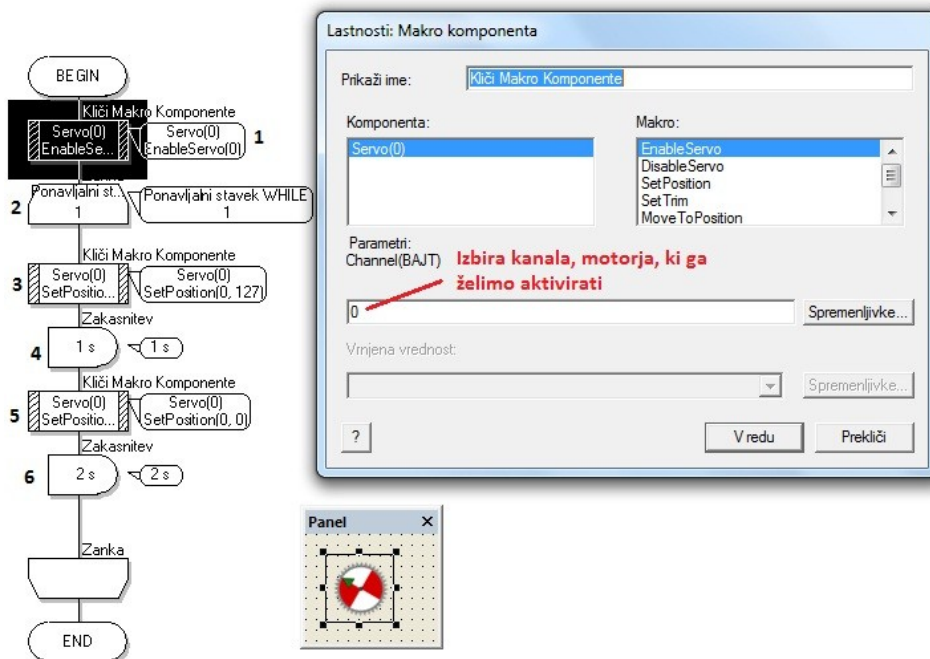
- Sledi pisanje programa po sistemu "drag and drop". Iz orodne vrstice z ukazi povlečemo ustrezne bloke in sestavimo diagram poteka.



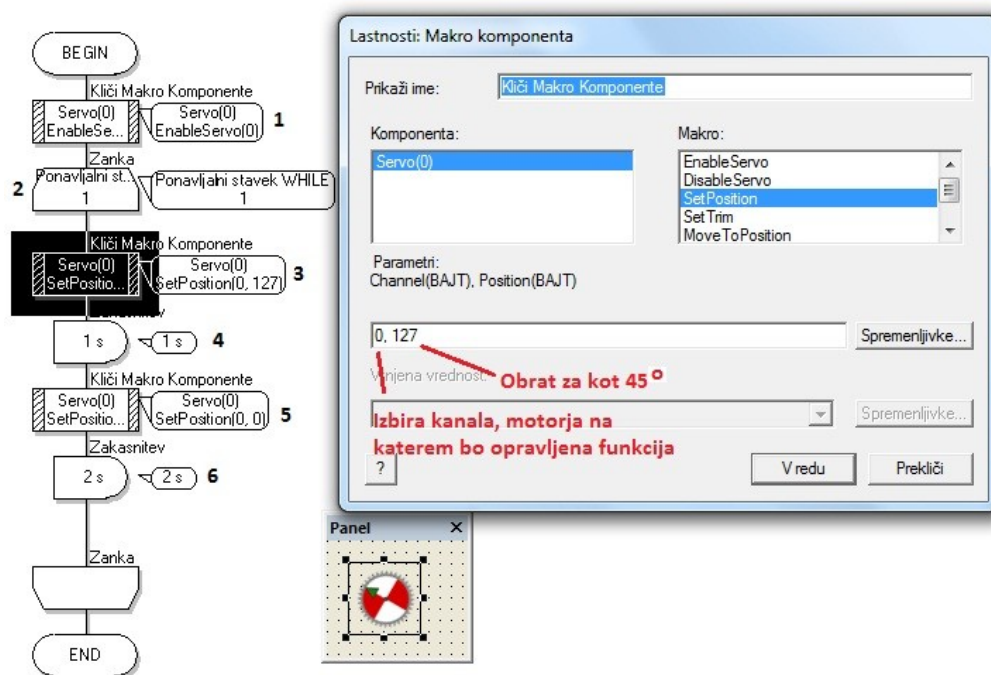
Slika 26 Primer programska aplikacija za krmiljenje servo motorja (Vir: Lastni viri)

RAZLAGA DIAGRAMA POTEKA V APLIKACIJI KRMILJENJE SERVO MOTORJA

Aplikacija se začne z aktivacijo servo signala na nožici RB0 (programski blok 1). V naslednjem bloku program vstopi v neskončno zanko (programski blok 2). Programski blok 3 na nožici RB0 pošlje signal, ki obrne rotirajoči del servo motorja za kot 45° (programski blok 3). Sledi pavza programa oz. zakasnitev za eno sekundo (programski blok 4). Programski blok 5 na nožici RB0 pošlje signal, ki povrne rotirajoči del servo motorja na začetni položaj. Potek programa ponovno izvede zakasnitev, tokrat za dve sekundi (programski blok 6).



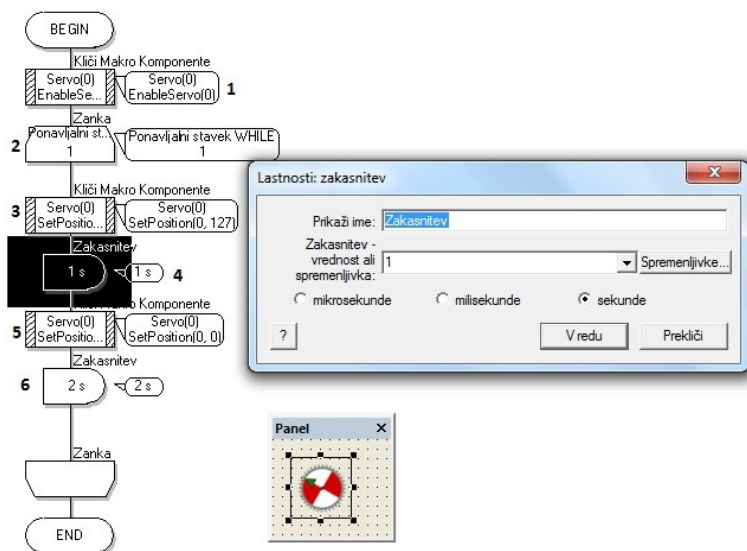
Slika 27 Programski blok 1 (Vir: Lastni viri)



Slika 28 Programski blok 3 (Vir: Lastni viri)

Macro komponenta »SetPosition« (glej sliko 25) se nahaja v programski aplikaciji v tretjem in četrtem bloku. Parameter »Position« definira obrat servo motorja. Največji obrat servo motorja (kot 90°) dosežemo, če vpišemo največjo število, ki jo lahko doseže spremenljivka vrste byte

(255). Z manjšimi vrednostmi te spremenljivke definiramo manjše obrate oz. kote servo motorja.



Slika 29 Programski blok 4 (Vir: Lastni viri)

Blok zakasnitev se nahaja v programski aplikaciji v četrtem in šestem bloku. Zakasnitev oz. časovni interval za ta blok lahko definiramo v mikrosekundah, milisekundah ali sekundah. Največja vrednost, ki jo lahko vpišemo v okence za zakasnitev, je 255. Vrednost oz. spremenljivka v okencu je definirana kot byte spremenljivka.

Element servo motorja vsebuje šest macro komponent:

- EnableServo,
- DisableServo,
- SetPosition,
- SetTrim,
- MoveToPosition,
- AvtoMoveToPosition.

EnableServo »macro« komponenta omogoči izbrano izhodno nožico, aktivira motor in mu dovoli, da se premakne na začetni položaj motorja. **DisableServo** »macro« komponenta onemogoči izbrano izhodno nožico in deaktivira motor. **SetPosition** »macro« komponenta



nastavi položaj izbranega motorja. **SetTrim** »macro« komponenta nastavi začetni položaj motorja. **MoveToPosition** »macro« komponenta premakne servo motor za korak. Funkcija mora biti ciklično klicana, da doseže želeno pozicijo. Ob dosegu zelene pozicije se funkcija vrne na logično enico. **AutoMoveToPosition** »macro« komponenta premakne servo motor po korakih do zelene pozicije. Frekvenca koraka je nastavljiva.

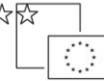


VAJE

1. Izdelaj programsko aplikacijo za krmiljenje servo motorja za modul ECIO28 s sledečimi zahtevami:
 - Os servo motorja naj opravi obrat za kot 80° , v naslednji sekundi naj se os povrne v začetni položaj. Krmilni signal naj se nahaja na nožici RAO mikrokontrolerja.
 - Zgoraj opisano dogajanje naj se sekvenčno ponavlja v neskončnost v časovnem intervalu 0,5 s.
2. Kateri motorji so »Servo« motorji in katere so pglavitne značilnosti teh motorjev? Poišči ustrezen slovenski prevod za to vrsto motorjev!

Opomba: Vprašanje razišči sam ali s pomočjo učitelja angleščine in učitelja, pri katerem spoznavaš električne elemente.

3. Izdelaj programsko aplikacijo za krmiljenje »Servo« motorja z modulom ECIO28 z upoštevanjem naslednjih zahtev:
 - V programski aplikaciji ustvari spremenljivko »vhod« (byte spremenljivka). V spremenljivko »vhod« nas se shranjuje logično stanje vhodne periferije s stikalom vezanim na RBO nožici. To realiziraj z vhodnim programskim blokom.
 - V odvisnosti od vrednosti spremenljivke »vhod« se aktivirata veji v »If stavku«:
 - Če je »vhod« = 1, deaktiviraj servo motor (DisableServo).
 - Če je »vhod« = 0, aktiviraj servo motor (EnableServo) in definiraj ciklično obnašanje motorja.



Opis cikličnega obnašanje motorja:

- Os servo motorja naj opravi obrat za kot 30° , v naslednji sekundi naj se os povrne v začetni položaj. Krmilni signal naj se nahaja na nožici RA0 mikrokontrolerja.
- Zgoraj opisano dogajanje naj se sekvenčno ponavlja v neskončnost v časovnem intervalu 1,5 s.



KRMILJENJE KORAČNEGA MOTORJA

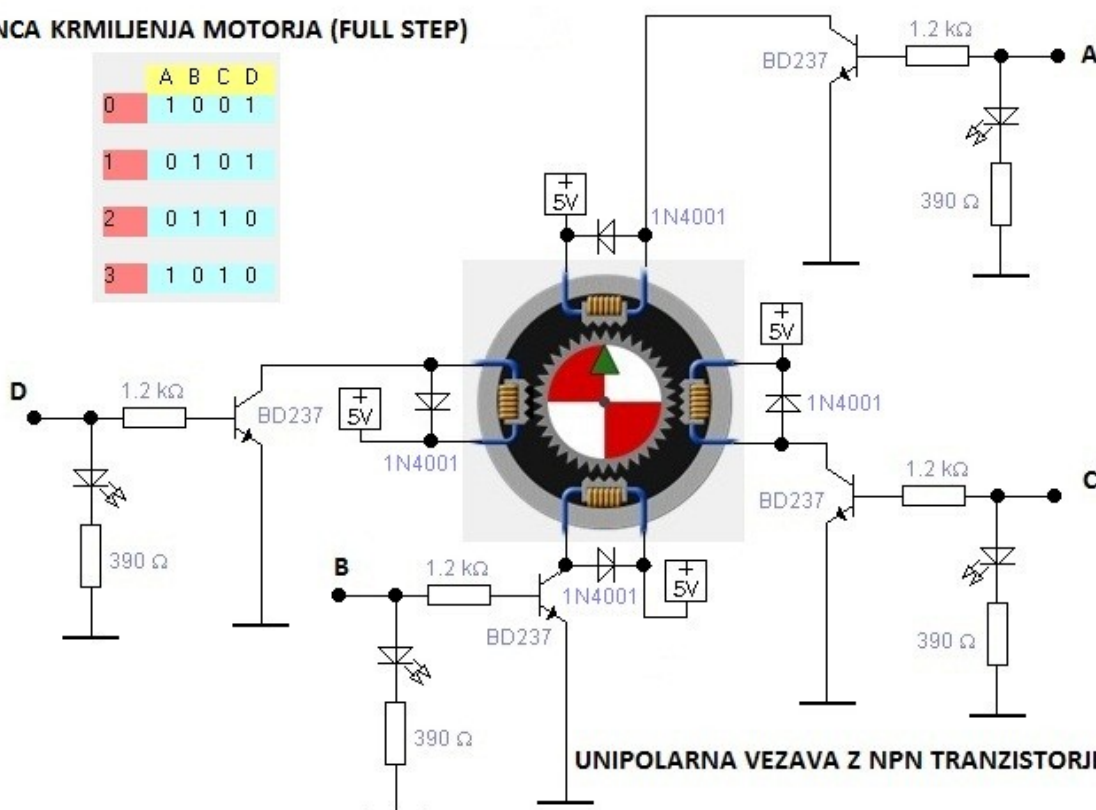
V vaji je uporabljen mikrokontrolerski modul ECIO28 in koračni motor. Mikrokontroler krmili koračni motor v unipolarni vezavi z bipolarnimi tranzistorji NPN s signali preko nožic RB0, RB1, RB2 IN RB3. Lahko bi bila izbrane katerekoli nožice na vratih mikrokontrolerja (vrata A, vrata B, vrata C).

Osnovni elektronski člen, električni sklop v unipolarni vezavi s tranzistorji NPN je tranzistor NPN v CE-vezavi (Skupni emitor, angl. Common emitter). Z digitalno sekvenco (FULL STEP), ki jo generiramo programsko na izhodih mikrokontrolerja (nožice RB0, RB1, RB2, RB3), krmilimo motor v unipolarni vezavi. Izhod mikrokontrolerja RB0 je priklopljen na vhod D, izhod mikrokontrolerja RB1 je priklopljen na vhod C, izhod mikrokontrolerja RB2 je priklopljen na vhod B in izhod mikrokontrolerja RB3 je priklopljen na vhod A unipolarne vezave servo motorja.

LED diode v vezavi imajo funkcijo prikaza logičnih stanj na izhodih mikrokontrolerja med delovanjem motorja.

SEKVENCA KRMILJENJA MOTORJA (FULL STEP)

	A	B	C	D
0	1	0	0	1
1	0	1	0	1
2	0	1	1	0
3	1	0	1	0



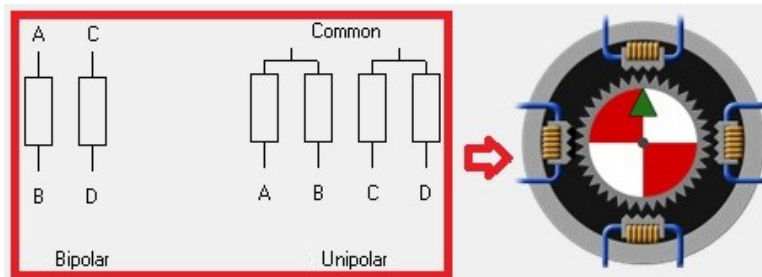
Slika 30 Izhodna periferija mikrokontrolerja (Vir: Lastni viri)

POSTOPEK PROGRAMIRANJA APLIKACIJE ZA KRMILJENJE KORAČNEGA MOTORJA

Koračni motor je zgrajen iz štirih tuljav in mehanizma (stator, rotor). Dobre lastnosti koračnega motorja se kažejo v resoluciji obrata motorja, navoru in moči koračnega motorja. Na trgu ga najdemo v dveh izvedbah:

- bipolarni koračni motor;
- unipolarni koračni motor.

Predstavljena aplikacija je napisana za krmiljenje unipolarnega koračnega motorja.



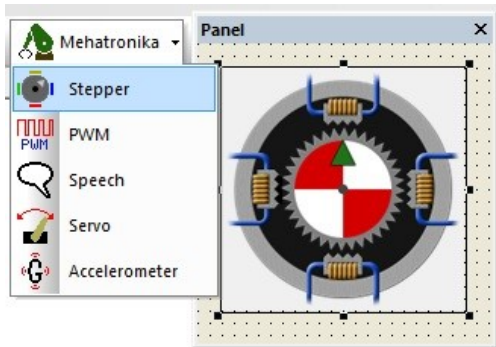
Slika 31 Koračni motor (Vir: Lastni viri)

Prikaz realizacije programske aplikacije, ki krmili unipolarni koračni motor z naslednjimi zahtevami:

3. unipolarni koračni motor krmili s polnim korakom (FULL STEP);
4. koraki si sledijo v časovnih intervalih ene sekunde.

Razlaga rešitve:

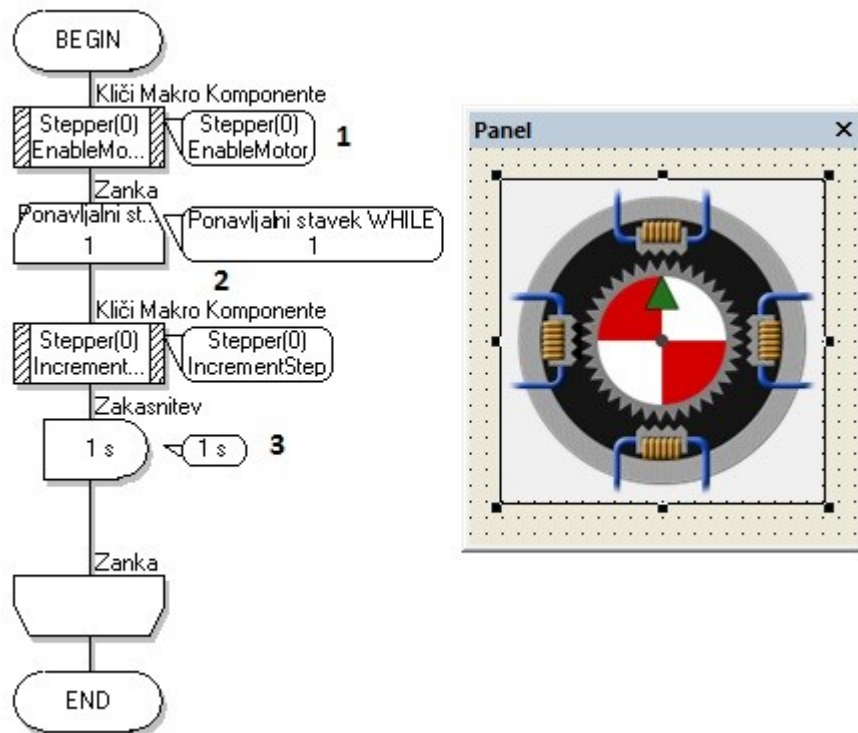
- V orodni vrstici pod Mehatronika izberemo med različnimi elementi modul Stepper. Kliknemo nanj in na »Panelu« se pojavi element koračnega motorja.



Slika 32 Element koračnega motorja (Vir: Lastni viri)

Z miško se pomaknemo na element koračnega motorja in z desnim klikom aktiviramo seznam tega elementa. Kliknemo na zunanje lastnosti in odpre se nam okno »Uredi komponento - Lastnosti«. V tem oknu nastavimo krmilno sekvenco krmilnih signalov na poln korak (FULL STEP) in definiramo vrsto in vezavo koračnega motorja («Unipolar-ve»). Z izbiro oznake Unipolar-ve se unipolarni koračni motor krmili s tranzistorji NPN. Skupni priključki koračnega motorja so vezani na pozitiven pol napajalne napetosti. S ponovnim desnim klikom na element koračnega motorja ponovno aktiviramo seznam tega elementa in izberemo aktivnost povezave. S klikom na povezave definiramo, iz katerih nožic mikrokontrolerja bo koračni motor pridobival krmilne signale. V predstavljeni aplikaciji so to nožice RB0, RB1, RB2 in RB3.

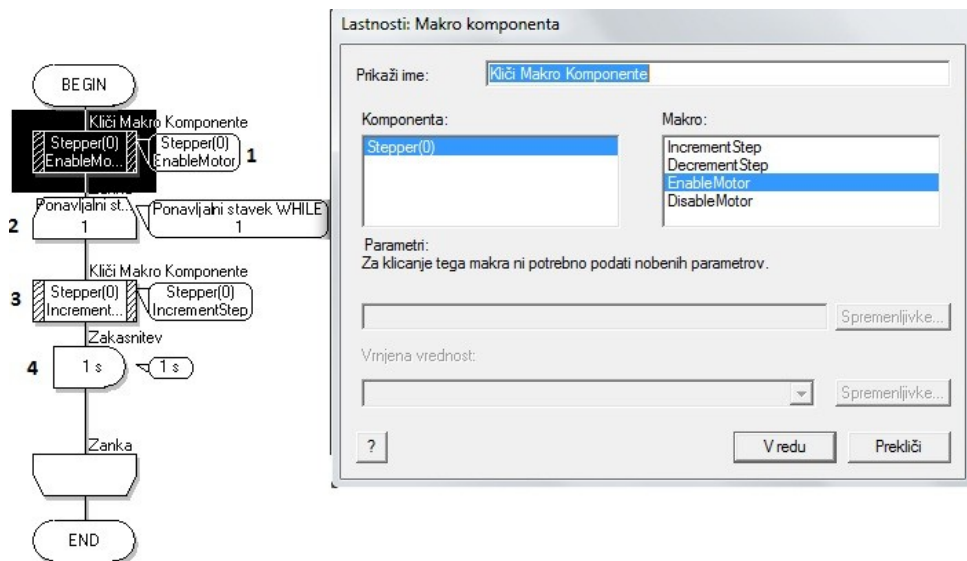
- Sledi pisanje programa po sistemu "drag and drop". Iz orodne vrstice z ukazi povlečemo ustrezne bloke in sestavimo diagram poteka.



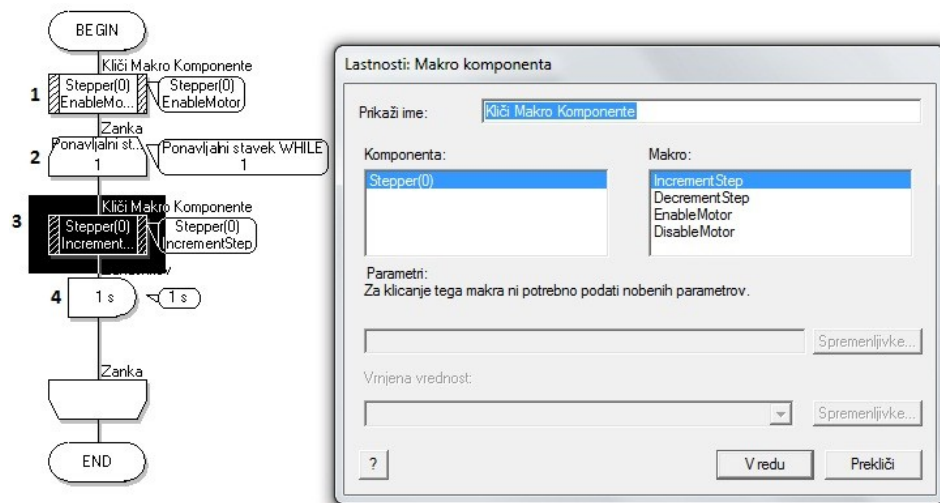
Slika 33 Primer programske aplikacije za krmiljenje koračnega motorja (Vir: Lastni viri)

RAZLAGA DIAGRAMA POTEKA V APLIKACIJI KRMILJENJE KORAČNEGA MOTORJA

Aplikacija se začne z aktivacijo koračnega motorja (programski blok 1). V naslednjem bloku program vstopi v neskončno zanko (programski blok 2). Programski blok 3 na nožice RB0, RB1, RB2 in RB3 pošlje signale, ki aktivirajo ustrezne tuljave v koračnem motorju in povzročijo obrat rotorja za en korak (programski blok 3). Sledi pavza programa oz. zakasnitev za eno sekundo (programski blok 4).



Slika 34 Programski blok 1 (Vir: Lastni viri)



Slika 35 Programski blok 3 (Vir: Lastni viri)

Element koračnega motorja vsebuje štiri macro komponente:

- IncrementStep,
- DecrementStep,
- EnableMotor,
- DisableMotor.



IncrementStep »macro« komponenta poveča korak motorja za ena. Ta komponenta je nasprotje **DecrementStep** »macro« komponente. Razlika v teh dveh komponentah je ta, da se enkrat rotor motorja vrti v eno smer, drugič pa v drugo.

EnableMotor »macro« komponenta aktivira koračni motor, **DisableMotor** »macro« komponenta pa deaktivira oz. ustavi koračni motor.



VAJE

1. Izdelaj programsko aplikacijo za krmiljenje koračnega unipolarnega motorja s skupnimi priključki koračnega motorja, vezanimi na negativni pol napajalne napetosti. Programska aplikacija naj bo izvedena za mikrokontrolerski modul ECIO40. Upoštevaj, da:
 - unipolarni koračni motor mikrokontroler krmili s polovičnim korakom (HALF STEP);
 - si koraki sledijo v časovnih intervalih 0,3 sekunde.

2. Kateri motorji so »Stepper« motorji in katere so poglavitne značilnosti teh motorjev? Poišči ustrezen slovenski prevod za to vrsto motorjev!

Opomba: Razišči sam ali s pomočjo učitelja angleščine in učitelja, pri katerem spoznaš električne elemente.

3. Izdelaj programsko aplikacijo za krmiljenje koračnega unipolarnega motorja s skupnimi priključki koračnega motorja, vezanimi na negativni pol napajalne napetosti motorja z modulom ECIO28 z upoštevanjem naslednjih zahtev:
 - V programski aplikaciji ustvari spremenljivko »vhod« (byte spremenljivka). V spremenljivko »vhod« naj se shranjujejo logično stanje vhodne periferije s stikalom vezanim na nožici RBO. To realiziraj z vhodnim programskim blokom.
 - V odvisnosti od vrednosti spremenljivke »vhod« se aktivirata veji v »Switch stavku«:
 - Če je »vhod« = 1:



- unipolarni koračni motor mikrokontroler krmili s polovičnim korakom (FULL STEP);
 - si koraki sledijo v časovnih intervalih 0,5 sekunde.
-
- Če je »vhod«=0:
 - unipolarni koračni motor mikrokontroler krmili s polovičnim korakom (FULL STEP);
 - si koraki sledijo v časovnih intervalih 1,5 sekunde. Motor se vrti v nasprotno smer.



VIRI

- Splet: <http://www.matrixmultimedia.com/>
- Splet: <http://www.cvel.clemson.edu/auto/actuators/motors-dc-brushed.html>
- Splet: <http://www.seattlerobotics.org/guide/servos.html>
- Splet: http://www.pyroelectro.com/tutorials/servo_motor/servomotor.html
- Splet: http://en.wikipedia.org/wiki/Stepper_motor
- Splet: <http://sl.wikipedia.org/wiki/Mikrokrmilnik>
- Ivič, Milan: Uporaba mikroprocesorskih naprav (2011), elektronsko gradivo.