

Univerza v Mariboru,  
Fakulteta za elektrotehniko, računalništvo in informatiko

MIKROPROCESORSKI SISTEMI  
2. letnik VS

# **POROČILO**

Maribor, 2013

Univerza v Mariboru,  
Fakulteta za elektrotehniko, računalništvo in informatiko

MIKROPROCESORSKI SISTEMI  
2. letnik VS

# **Priprave**

## **POROČILO 2. VAJE**

Avtor:  
Gregor Nikolić  
E1054204

Maribor, 2013

## KAZALO

1. UVOD .....	4
2. VAJA 1 .....	5
2.1 Besedilo naloge .....	5
2.2 Besedni opis delovanja programa.....	6
3. DIAGRAM POTEKA.....	7
4. PROGRAM.....	8

## 1. UVOD

Poročilo obsega programsko kodo ter diagram poteka za prvo vajo pri predmetu Mikroprocesorski sistemi, smer študija Elektrotehnika VS na Univerza v Mariboru, Fakulteta za Elektrotehniko Računalništvo in informatiko.

Gradivo se sme uporabljati v namene izobraževanja in se ga nikakor ne sme reproducirati ali spreminjati v komercialne namene brez soglasja avtorja.

Copyright © 2013, Gregor Nikolić, Maribor

## 2. VAJA 1

### 2.1 Besedilo naloge

Na mikroračunalnik z uporabo paralelnega vmesnika priključite koračni motorček, ki se je pred leti uporabljal v velikih disketnikih za natančno premikanje glave. Napišite preprost program za sistem KAT-Ce s katerim boste krmilili premikanje motorčka. Motorček naj se odziva na ukaze s tipkovnice na razvojnem računalniku na naslednji način:

- tipka '0' izključi vrtenje motorja
- tipka '1' vključi vrtenje motorja
- tipka 'L' vrtenje v nasprotni smeri urinega kazalca
- tipka 'D' vrtenje v smeri urinega kazalca
- tipka '+' hitreje
- tipka '-' počasneje.

#### Priporočilo:

Uporabite vrtenje po korakih, tako da motorčku preko vrat A zaporedoma pošiljate kode

##%1010,##%0110, ##%0101 in ##%1001.

Za inicializacijo vrat A in B uporabite naslednjo kodo:

```
PGCR      EQU $400001 ;Port General Control Register
PADDR     EQU $400005 ;Port A Data Direction Register
PBDDR     EQU $400007 ;Port B Data Direction Register
PACR      EQU $40000D ;Port A Control Register
PBCR      EQU $40000F ;Port B Control Register
PADR      EQU $400011 ;Port A Data Register
PBDR      EQU $400013 ;Port B Data Register

INIT:     MOVE. B #$00, PGCR ; Izberemo MODE 0
          MOVE. B #$80, PACR ; Izberemo SUBMODE 1X
          MOVE. B #$80, PBCR ; Izberemo SUBMODE 1X
          MOVE. B #%11111111, PADDR ; vsi pini na vratih A so izhodi
          MOVE. B #%00000000, PBDDR ; vsi pini na vratih B so vhodi
```

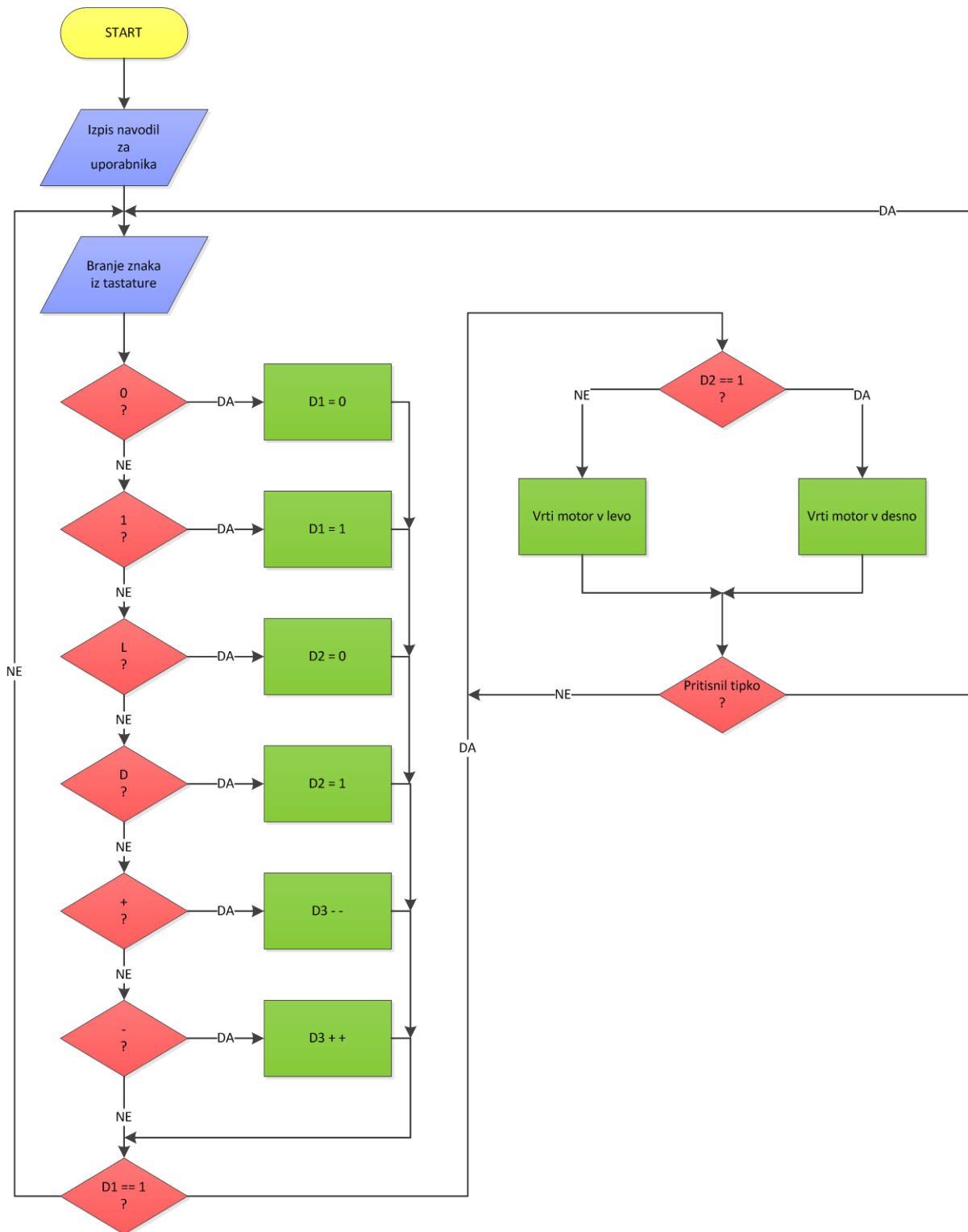
Primer pisanja in branja iz vrat A in B:

```
WRITE_A   MOVE. B Dx, PADR ; pi semo na vrata A
READ_A    MOVE. B PBDR, Dx ; beremo vrednost na vratih B
```

## 2.2 Besedni opis delovanja programa

V programu se najprej inicializirajo vrata A in B za koračni motor, nastavijo se pini na vratih ali so izhodni oz. vhodni. Sledi izpis navodil za uporabnika in sicer se izpišejo opici posameznih ukazov za koračni motorček. Preventivno očistimo register D1, katerega uporabljamo za smer oz. vrtenje motorja ter hitrost. Na spodnjih 16 bitov vpišemo začetno hitrost motorja ter nadaljujemo na neskončni zanki »INF\_ZANKA«. Kličemo podprogram za branje znaka iz tastatúre. Po vnesenem znaku program nadaljuje na preverjanju vpisanega znaka. V kolikor ne najde enakosti nadaljuje na čakanje za nov znak. V primeru pravilno vpisanega znaka – ukaza ustrezno veji v podprograme, kjer se biti v registru D1 ustrezno postavijo. Na koncu vedno preverimo ali je postavljen 16. bit v D1, ki nam pove ali moramo motor vrteti ali ne. V primeru, da je motor ugasnjen, nadaljujemo s preverjanjem vpisanega znaka v nasprotnem primeru vejimo na rutino kjer preverimo ali moramo motor vrteti v desno ali levo smer na način, da testiramo 17. bit v D1, kateri nam pove smer. Po preverjanju ustrezno glede na postavljen bit vejimo na rutino, ki v zaporedju pošilja ustrezno kombinacijo na izhod motorčka – med vsako kombinacijo se ustrezno zakasni z zanko »ZAKASNITEV«. Ta zakasnitev nam omogoča nastavitvev hitrosti vrtenja motorja, vrstni red pošiljanja kombinacij na izhod pa smer motorja. Vredno je še omeniti, da sta postavljeni meji hitrosti in sicer zgornja ter spodnja meja.

### 3. DIAGRAM POTEKA



Slika 1 Diagram poteka

## 4. PROGRAM

```

        section      .text

main:

PGCR EQU $400001      // Inicializacija vrat A in B, Port General Control Register
PADDR EQU $400005     // Port A Data Direction Register
PBDDR EQU $400007     // Port B Data Direction Register
PACR EQU $40000D     // Port A Control Register
PBCR EQU $40000F     // Port B Control Register
PADR EQU $400011     // Port A Data Register
PBDR EQU $400013     // Port B Data Register

INIT: MOVE.B    #$00,PGCR      // Izberemo MODE 0
      MOVE.B    #$80,PACR      // Izberemo SUBMODE 1X
      MOVE.B    #$80,PBCR      // Izberemo SUBMODE 1X
      MOVE.B    #%11111111,PADDR // vsi pini na vratih A so izhodi
      MOVE.B    #%00000000,PBDDR // vsi pini na vratih B so vhodi

      LEA      NAVODILA(PC),A1  // Izpis navodil za uporabnika
      MOVE.W   #$000D,-(A7)     //
      TRAP     #4               //

      LEA      NAVODILA_2(PC),A1 // nadaljna navodila za uporabnika.
      MOVE.W   #$000D,-(A7)     //
      TRAP     #4               //

      CLR.L    D1               // Preventivno ciscenje
      CLR.L    D4               //
      MOVE.L   #$500,D1         // Predhodno nastavljena zacetna hitrost vrtenja

INF_ZANKA:
      MOVE.W   #$0006,-(A7)     // Klic podprograma za branje iz tastature
      TRAP     #4               //

      CMP.W    #'0',D0          // Ali je vnesen znak enak "0" ?
      BNE.B    N1               // Ne, nadaljuj
      BSR.W    D1_0             // Da, D1[16] = 0
N1:
      CMP.W    #'1',D0          // Ali je vnesen znak enak "1" ?
      BNE.B    N2               // Ne, nadaljuj
      BSR.W    D1_1             // Da, D1[16] = 1
N2:
      CMP.W    #'L',D0          // Ali je vnesen znak enak "L" ?
      BNE.B    N3               // Ne, nadaljuj
      BSR.W    LEVO_S           // Da, D1[17] = 0
N3:
      CMP.W    #'D',D0          // Ali je vnesen znak enak "D" ?
      BNE.B    N4               // Ne, nadaljuj
      BSR.B    DESNO_S         // Da, D1[17] = 1
N4:
      CMP.W    #'+',D0          // Ali je vnesen znak enak "+" ?
      BNE.B    N5               // Ne, nadaljuj
      BSR.W    POSPESI         // Da, Povecaj vrednost .W D1
N5:
      CMP.W    #'-',D0          // Ali je vnesen znak enak "-" ?
      BNE.B    N7               // Ne, nadaljuj
      BSR.W    UPOCASNI        // Da, zmanjsaj vrednost .W D1
N7:
      CMP.W    #'1',D0          // Ali moram vrteti motor?
      BNE.B    INF_ZANKA       // Ne, pojdi cakati na nov ukaz

      BTST.L   #17,D1           // Da, vrteti moram motor, kam? Ali je D1[17] = 1 ?
      BNE.B    N8               // Ni enak, pojdi preveriti ali je D1[17] = 0
      BSR.W    LEVO             // Da, D1[17] = 1, pojdi vrteti motor v desno
N8:
      BTST.L   #17,D1           // Ali je D1[17] = 1?
      BEQ.B    N9               // Da, pojdi naprej
      BSR.W    DESNO           // Ne, D1[17] = 0, pojdi vrteti motor v levo
N9:
      JMP     N7               // Pojdi preveriti ali je se potrebno vrteti motor

D1_1:
      BSET    #16,D1           // Subrutina - D1[16] = 1
      RTS

```



## Internetne tehnologije

```
D1_0:
    BCLR    #16,D1    // Subrutina - D1[16] = 0
    RTS

POSPESI:
    MOVE.W  D1,D4    // Shranim v buffer
    ADD.W   #$250,D1 // Pristejem $250 D1 in s tem skrajsam zakasnitev - pospesim motor
    CMP.W   #$34D0,D1 // Ali je vrednost D1 enaka oz vecja $34D0 ? (34D0 je max za motor)
    BLT.W   N_POS    // Ce je vrednost manjsa ali enaka 34D0 pojdi naprej
    MOVE.W  #$34D0,D1 // v nasprotnem primeru povrni vrednost iz D4 v D1 - zgornja meja
N_POS:
    RTS          // Poveca vrednost .W D1

UPOCASNI:
    MOVE.W  D1,D4    // Shranim v Buffer
    SUB.W   #$250,D1 // Odstejem $250 D1 in s tem podaljsam zakasnitev - upocasnim motor
    CMP.W   #$250,D1 // Primerjaj vrednost $250 in vrednost v D1 (.W!)
    BGT.W   N_UPO    // Ce je vrednost vecja ali enaka $250 pojdi naprej
    MOVE.W  #$250,D1 // v nasprotnem primeru povrni vrednost iz D4 v D1 - spodnja meja
N_UPO:
    RTS          // Zmanjsa vrednost .W D1

DESNO_S:
    BSET    #17,D1    // Subrutina - smer motorja v desno
    RTS     // D2 = 1
LEVO_S:
    BCLR    #17,D1    // Subrutina - smer motorja v levo
    RTS     // D2 = 0

DESNO:
    MOVE.B  #%1010,PADR // Vrtenje motorja
    MOVE.W  D1,D4    // Inicializacija D4 za zakasnitev
    BSR.W   ZAKASNITEV // Klic subrutine - Branch to Subrutine
    MOVE.B  #%0110,PADR //
    MOVE.W  D1,D4    //
    BSR.W   ZAKASNITEV //
    MOVE.B  #%0101,PADR //
    MOVE.W  D1,D4    //
    BSR.W   ZAKASNITEV //
    MOVE.B  #%1001,PADR //
    MOVE.W  D1,D4    //
    BSR.W   ZAKASNITEV //
    MOVE.W  #$0004,-(A7) // Je prislo do sprememb? (pritisak na tipko)
    TRAP    #4        //
    BTST    #0,D0     //
    BEQ     DESNO     // Ne, nadaljuj z vrtenjem
    JMP     INF_ZANKA // Da, pojdi ponovno preveriti pogoje
LEVO:
    MOVE.B  #%1001,PADR // Vrtenje motorja
    MOVE.W  D1,D4    // Inicializacija D4 za zakasnitev
    BSR.W   ZAKASNITEV // Klic subrutine - Branch to Subrutine
    MOVE.B  #%0101,PADR //
    MOVE.W  D1,D4    //
    BSR.W   ZAKASNITEV //
    MOVE.B  #%0110,PADR //
    MOVE.W  D1,D4    //
    BSR.W   ZAKASNITEV //
    MOVE.B  #%1010,PADR //
    MOVE.W  D1,D4    //
    BSR.W   ZAKASNITEV //
    MOVE.W  #$0004,-(A7) // Je prislo do sprememb? (pritisak na tipko)
    TRAP    #4        //
    BTST    #0,D0     //
    BEQ     LEVO      //
    JMP     INF_ZANKA //

ZAKASNITEV:
    ADD.W   #1,D4    // Zakasnitev s pomocjo pristevanja
    CMP.W   #$44D0,D4 // je ze vrednost 2000?
    BNE.W   ZAKASNITEV // Ne, nadaljuj s pristevanje
    RTS     // Vrnitev iz subrutine

NAVODILA:
    DC.B   26,"Ukazi za vtenje motorja: ",$A,$D
NAVODILA_2:
    DC.B   61,"0~Ustavi, 1~Pozeni, L~Levo, D~Desno, +~Hitreje, --Pocasneje", $A,$D

    END
```