

```

#include<stdio.h>
#include<conio.h>
#include<dos.h>
#include<math.h>

#define PI 3.141592654
#define SLIPGAIN 2450
#define TORQMAX 123207680
#define IQNOM 118
#define IDNOM 95
#define KI 6461
#define KP 50565
#define QTR1 20480000
#define QTR2 40960000
#define QTR3 61440000
#define QTR4 81920000
#define BUFFER_SIZE 42000

union dugi {
    long lo;
    struct {
        int lower,upper;
    }in;
};

union krat {
    int rec;
    struct {
        unsigned char lower,upper;
    }bajt;
}enc;

union poin {
    unsigned int rec;
    struct {
        unsigned char lower,upper;
    }bajt;
}poin;

union dugi tetasyn,sinarglong,cosarglong,refrtorque;
union dugi speed;
union krat tim0,tim1;
long slipinc,refrtorqueold=0,deltor;
long gainslip,speedref=0,speedold=0;
long kkp,kki,refspeed;
int ia,ib,id,iq,iqabs,iqneg,encold=0;
int dell_9,delttime,count,inc;
int idsin,idcos,iqsin,iqcos;
int argsin,argcoss;
int num1,num2,num3,speeddac;
char ch1,ch2,delay;
float fspeed;

unsigned char tabela[BUFFER_SIZE];

void interrupt far intlc()
{
    ++count;

    if(count==10)
    {
        outp(0x302,0x0f); /* 0000 1111 */
                        /* freeze encoder latch */
                        /* set A/D blank&convert */
                        /* gates #0 #1 frozen */

        delay=99;
    }
}

```

```

tim0.bajt.lower=inp(0x30c); /* cntr#0 lsb */
delay=98;
tim0.bajt.upper=inp(0x30c); /* cntr#0 msb */
delay=97;
tim1.bajt.lower=inp(0x30d);
delay=96;
tim1.bajt.upper=inp(0x30d);

enc.bajt.lower=inp(777);
enc.bajt.upper=inp(776);

outp(0x30c,0xff); /* reload cnt#0 lsb */
delay=99;
outp(0x30c,0xff); /* reload cnt#0 msb */

outp(0x302,0x01); /* 0000 0001 */
/* start AD conversion */
/* latch enabled */
/* gates frozen until #1 */

inc=enc.rec-encold;
encold=enc.rec;
if(inc>1250) inc=inc-2500;
if(inc<-1250) inc=inc+2500;

dell_9=dell_9+inc;

if(dell_9==0) {speed.lo=0; goto reg; };
if(dell_9>0) {--dell_9; speed.lo=116; };
if(dell_9<0) {++dell_9; speed.lo=-116;};
if(dell_9==0){ goto reg;};

/* ako je broj impulsa >=2 , racuna se brzina po punoj formuli */

delttime=tim1.rec-tim0.rec;
speed.in.upper=dell_9*32;
if(delttime==0) deltime=18000;
speed.lo=speed.lo/delttime;
if((speed.lo-speedold)>1440) speed.lo=speedold+1440;
if((speedold-speed.lo)>1440) speed.lo=speedold-1440;

reg;;

/* PI pozicioni regulator brzine spl time=10ms limit=2 Tn */

deltor=kki*(speedref-speed.lo)+kkp*(speedold-speed.lo);
speedold=speed.lo;
/* limiter na +/- 2 Tn */

reftorque.lo=reftorqueold+deltor;
if(reftorque.lo>TORQMAX){reftorque.lo=TORQMAX;};
if(reftorque.lo<-TORQMAX){reftorque.lo=-TORQMAX;};
reftorqueold=reftorque.lo;
iq=(reftorque.in.upper)>>3;
count=0;
goto transf;
};

if(count==1)
{
outp(0x302,0x0e); /* 0000 1110 */
/* ADC b&c high pc1 hi pc2 lo */
/* latch frozen , first next */
/* encoder pulse sets gates */
enc.bajt.lower=inp(777);
enc.bajt.upper=inp(776);

```

```

outp(0x302,0x00); /* start ADC release latch */
/* pc0 stays 0 to set gates */
inc=enc.rec-encold;
encold=enc.rec;
if(inc>1250) inc=inc-2500;
if(inc<-1250) inc=inc+2500;
dell_9=0;

speeddac=(speedold>>10)+128; /* izbacivanje brzine */
outp(778,speeddac); /* na port ppi#3 c */
/* 128 +/- 70 za om n */
goto transf; /* = +/- 5.45 V */
};

```

```

/* for count== 2..9 */

```

```

outp(0x302,0x0e); /* 0000 1110 */
enc.bajt.lower=inp(777); /* pc0 stays low */
enc.bajt.upper=inp(776);

outp(0x302,0x00); /* 0000 0000 */

inc=enc.rec-encold;
encold=enc.rec;
if(inc>1250) inc=inc-2500;
if(inc<-1250) inc=inc+2500;

dell_9=dell_9+inc;

```

```

transf:;

```

```

/* obrtna transformacija (id,iq)=>(ia,ib) sin je tabeliran 0-90 */

```

```

if(iq<0) {iqabs=-iq;iqneg=1;}
else {iqabs=iq ;iqneg=0;};

```

```

tetasyn.in.upper=tetasyn.in.upper+inc;
slipinc=gainslip*iq;
tetasyn.lo=tetasyn.lo+slipinc;
if(tetasyn.lo>QTR4){tetasyn.lo=tetasyn.lo-QTR4;};
if(tetasyn.lo<0){tetasyn.lo=tetasyn.lo+QTR4;};

```

```

if(tetasyn.lo<QTR2)
{ if(tetasyn.lo<QTR1)
{
sinarglong.lo=tetasyn.lo;
cosarglong.lo=QTR1-tetasyn.lo;
argsin=sinarglong.in.upper>>1;
argcos=cosarglong.in.upper>>1;

poin.bajt.lower=id;poin.bajt.upper=argsin;
idsin=tabela[poin.rec];

poin.bajt.lower=id;poin.bajt.upper=argcos;
idcos=tabela[poin.rec];

poin.bajt.lower=iqabs;poin.bajt.upper=argsin;
iqsin=tabela[poin.rec];

poin.bajt.lower=iqabs;poin.bajt.upper=argcos;
iqcos=tabela[poin.rec];
}
}

```

```

else
{
    sinarglong.lo=QTR2-tetasyn.lo;
    cosarglong.lo=tetasyn.lo-QTR1;
    argsin=sinarglong.in.upper>>1;
    argcos=cosarglong.in.upper>>1;

    poin.bajt.lower=id;poin.bajt.upper=argsin;
    idsin=tabela[poin.rec];

    poin.bajt.lower=id;poin.bajt.upper=argcos;
    idcos=-tabela[poin.rec];

    poin.bajt.lower=iqabs;poin.bajt.upper=argsin;
    iqsin=tabela[poin.rec];

    poin.bajt.lower=iqabs;poin.bajt.upper=argcos;
    iqcos=-tabela[poin.rec];
};
}

else
{ if(tetasyn.lo<QTR3)
    {
        sinarglong.lo=tetasyn.lo-QTR2;
        cosarglong.lo=QTR3-tetasyn.lo;
        argsin=sinarglong.in.upper>>1;
        argcos=cosarglong.in.upper>>1;

        poin.bajt.lower=id;poin.bajt.upper=argsin;
        idsin=-tabela[poin.rec];

        poin.bajt.lower=id;poin.bajt.upper=argcos;
        idcos=-tabela[poin.rec];

        poin.bajt.lower=iqabs;poin.bajt.upper=argsin;
        iqsin=-tabela[poin.rec];

        poin.bajt.lower=iqabs;poin.bajt.upper=argcos;
        iqcos=-tabela[poin.rec];
    }

    else
    {
        sinarglong.lo=QTR4-tetasyn.lo;
        cosarglong.lo=tetasyn.lo-QTR3;
        argsin=sinarglong.in.upper>>1;
        argcos=cosarglong.in.upper>>1;

        poin.bajt.lower=id;poin.bajt.upper=argsin;
        idsin=-tabela[poin.rec];

        poin.bajt.lower=id;poin.bajt.upper=argcos;
        idcos=tabela[poin.rec];

        poin.bajt.lower=iqabs;poin.bajt.upper=argsin;
        iqsin=-tabela[poin.rec];

        poin.bajt.lower=iqabs;poin.bajt.upper=argcos;
        iqcos=tabela[poin.rec];
    };

};

if(iqneg){iqsin=-iqsin;iqcos=-iqcos;};

ia=(idcos-iqsin)>>1;          /* DAC: 0=-10 V */

```

```

        ib=(idsin+iqcos)>>1;          /* 128=0 V */
        ia=ia+128; ib=ib+128;        /* 255=10 V */
        outp(772,ia);                /* iaref na DAC */
        outp(773,ib);                /* ibref na DAC */
        outp(774,((iq>>1)+128));     /* iq na DAC */
    }

/* kraj interrupt rutine */

/* inicijalizacioni program i background petlja za monitoring */
main(void)
{

/* citanje tabele iz fajla TABELA.DAT */

FILE *fp;

fp = fopen("TABELA.DAT", "rb");
if(fp == (FILE *)NULL)
    { printf("Error opening file TABELA.DAT \nAborting...\n");
      return(-1);
    };
if(fread(tabela, BUFFER_SIZE, 1, fp) < 1)
    { printf("Error reading file TABELA.DAT\nAborting...");
      fclose(fp);
      return(-1);
    };
fclose(fp);

gainslip=SLIPGAIN; kkp=KP; kki=KI; iq=0;
id=IDNOM; count=0; inc=0; speed.lo=0; speedold=0;
tetasyn.lo=0; reftorqueold=0; speedref=0;
encold=0; num1=100; num2=0;

/* inicijalizacija portova 771<=9a pa=in pb=in pcu=in pcl=out */
/* 775<=80 pa=out pb=out pc=out */
/* 779<=92 pa=in pb=in pc=out */
    outp(779,0x92);
    outp(775,0x80);
    outp(771,0x9a);

/* inicijalizacija sistemskog tajmera - interrupt #0 svakih 1ms */

    outp(0x021,0x01); /* disable interrupt #0 */
    outp(0x043,0x36); /* select cntr#0 mode 3 */
    outp(0x040,0xa8); /* load lsb */
    outp(0x040,0x04); /* load msb */
    outp(0x021,0x00); /* enable all interrupts */

/* inicijalizacija tajmera #0 i #1 u eksternom 8253 */

    outp(0x30f,0x30); /* cntr #0 mode 0 */
    delay=33;
    outp(0x30f,0x74); /* cntr #1 mode 2 */
    delay=32;
    outp(0x30c,0xff); /* load #0 lsb */
    delay=31;
    outp(0x30c,0xff); /* load #0 msb */
    delay=30;
    outp(0x30d,0xff); /* load #1 lsb */
    delay=29;
    outp(0x30d,0xff); /* load #1 msb */

```

```

printf("\n regulator brzine vektorski kontrolisanog");
printf("\n asinhronog motora ");
printf("\n PI regulator kp= %ld ki=%ld change ? y/n ",kkp,kki);
scanf("%c",&chl);
if(chl=='y') { printf("\n kp= ? ");scanf("%ld",&kkp);
               printf("\n ki= ? ");scanf("%ld",&kki);
               }

_dos_setvect(0x1c,int1c);

lab1:; ++num1;
if(num1>20){num1=0; printf("\n speed ? "); scanf("%ld",&refspeed);
            if(refspeed==9999) goto exit;
            if(refspeed>2000) refspeed=2000;
            if(refspeed<-2000) refspeed=-2000;
            speedref=refspeed*48;
            };
            fspeed=((float) speedold)/48.0;

            printf("\n iq %d %% speed %f rpm ",((iq*100)/118),fspeed);
goto lab1;

exit:;
        outp(0x021,0x01); /* mask interrupt */
}

```