

```

/*
vystupy z PICu
RA0-ADC- hladina baterie 1(clanek 1)
RA1-ADC- hladina baterie 1(clanek 2)
RA2-ADC- hladina baterie 2(clanek 1)
RA3-ADC- hladina baterie 2(clanek 2)
RA5-ADC- hladina sol. panelu na vstupu

RC0-optoclen-rizeni vypinani na vstupu
RC1-rizeni vstupniho menice - zmena frekvence
RC4-SDI-I2C
RC3-SCL-I2C

RD0-prepinani clanku na baterii 1(clanek 1) -vstup
RD1-prepinani clanku na baterii 1(clanek 2) -vstup
RD2-prepinani clanku na baterii 2(clanek 1) -vstup
RD3-prepinani clanku na baterii 2(clanek 2) -vstup
RD4-optoclen vystup baterie 1
RD5-optoclen vystup baterie 2
RD6-dioda zapnuti

RE0-ADC-baterie 1 -BAT1+
RE1-ADC-baterie 2 -BAT2+
RE2-ADC-baterie 12 -BAT--
*/
int prijmani=0; //promena pro prijem zpravy z I2C
double Ibat1,Ibat2,Iinabij; //proudy vystupnich baterii -pro vypocty
int hlso=1; //promena pro rozhodnuti nabijeni z panelu (hladina
solaru)
int dvebat=0; //promena pro rozhodnuti letu se zatizenim dvou
baterii
int hel=0,hl=0; //pomocne promenna pro mereni
double odpor=0; //promenna pro vypocet proudu
double
clanek1=0,clanek2=0,clanek3=0,clanek4=0,panel=0,bat1=0,bat2=0,bat12=
0;// pro hodnoty ADC

void prvni_spusteni(){ // nastavi prvotni hodnoty
    TRISD=0b00000000; //nastavení vstupu/vystupu
    TRISC=0b00000000; //nastavení vstupu/vystupu
    PORTD.RD6=1; //rozsviceni LEDky pro info ze
je zaplo
    PORTD.RD5=1; //vypnuti 2 baterie na vystupu
    PORTD.RD4=0; //zapnuti 1 baterie na vystupu
    PORTC.RC0=1; //vypnuti vstupniho optoclenu
    PORTD.RD0=1; //vypnuti prepínání nabíjení
bat1 clan1
    PORTD.RD1=1; //vypnuti prepínání nabíjení
bat1 clan2
    PORTD.RD2=0; //zapnuti prepínání nabíjení
bat2 clan1
    PORTD.RD3=0; //zapnuti prepínání nabíjení
bat2 clan2
}

void vypnuti_nabijeni(){ //vypne nabíjení baterii

```

```

    PORTC.RC0=1; //vypnuti vstupniho optoclenu
}

void spinaci_frekvence(){ //metoda pro urceni frekvence nabijeni
//koment pro demo-limit
    // int i; //promenna pro cyklus
/*    if(panel>=2.0&& panel<3.2) //podminka pro frekvenci
spinani
    {
        for( i=0;i<10;i++) //cyklus pro spinani
            {
                if( PORTD.RC0==1)
                {
                    PORTC.RC1=0;
                }
                else
                {
                    PORTC.RC1=1;
                }
                delay_us(100);
            }
    }

    if(panel>=3.2&& panel<4.0) //podminka pro frekvenci spinani
    {
        for( i=0;i<10;i++) //cyklus pro spinani
            {
                if( PORTD.RC0==1)
                {
                    PORTC.RC1=0;
                }
                else
                {
                    PORTC.RC1=1;
                }
                delay_us(200);
            }
    }

    if(panel>=4.0&& panel<5.1) //podminka pro frekvenci spinani
    {
        for( i=0;i<10;i++) //cyklus pro spinani
            {
                if( PORTD.RC0==1)
                {
                    PORTC.RC1=0;
                }
                else
                {
                    PORTC.RC1=1;
                }
                delay_us(300);
            }
    }

    if(panel>=5.1&& panel<6.2) //podminka pro frekvenci spinani

```

```

{
for( i=0;i<10;i++) //cyklus pro spinani
{
if( PORTD.RC0==1)
{
PORTC.RC1=0;
}
else
{
PORTC.RC1=1;
}
delay_us(400);
}
}

if(panel>=6.2&& panel<7.0) //podminka pro frekvenci spinani
{
for( i=0;i<10;i++) //cyklus pro spinani
{
if( PORTD.RC0==1)
{
PORTC.RC1=0;
}
else
{
PORTC.RC1=1;
}
delay_us(700);
}
} /*
} //nutny koment pro demo limit

void prepinari_clanku_prvni(){ //prepinani clanku na prvni
baterii
int i; //pomocna promenna
for(i=0;i<10;i++) //for pro opakovani prepinari
{
if( PORTD.RD0==1) //podminka pro prepinari
{
PORTD.RD0=0; //zapne prvni clanek
PORTD.RD1=1; //vypne druhy clanek
spinaci_frekvence();//spina nabijeni
}
else
{
PORTD.RD0=1; //vypne prvni clanek
PORTD.RD1=0; //zapne druhy clanek
spinaci_frekvence();//spina nabijeni
}
delay_ms(300);
}
}

void prepinari_clanku_druha(){ //prepinani clanku na druhe
baterii
int i; //pomocna promenna

```

```

    for( i=0;i<10;i++) //for pro opakovani prepnaní
    {
        if( PORTD.RD2==1) //podmínka pro prepnaní
        {
            PORTD.RD2=0; //zapne první článěk
            PORTD.RD3=1; //vypne druhý článěk
            spinaci_frekvence();//spína nabíjení
        }
        else
        {
            PORTD.RD2=1; //vypne první článěk
            PORTD.RD3=0; //zapne druhý článěk
            spinaci_frekvence();//spína nabíjení
        }
        delay_ms(300);
    }
}

void prepnaní_clanku() //podle nabíjené baterie prepína články
{
    if(PORTD.RD5==0) //zapnuta 2 baterie
    {
        prepnaní_clanku_druha() ; //metoda pro prepnaní článku
    }
    if(PORTD.RD4==0) //zapnuta 1 baterie
    {
        prepnaní_clanku_první() ; //metoda pro prepnaní článku
    }
}

void zapnutí_nabíjení(){ //zapne nabíjení baterii
    PORTC.RC0=0; //zapnutí vstupního optočlenu
}

void měření_panel() { //změří hodnotu solárního panelu
    hl=ADC_Read(5);
    panel=((hl*8.4)/1024); //přepočítá hodnotu podle maximálního
napětí 8.4V
}

void hlídání_hlady_solar(){//určí hodnotu pro nabíjení baterii
    měření_panel(); //změří hladinu panelu
    if(panel>1.0) //podmínka pro určení nabíjení
    {
        hlso=1; //nabíjí
    }
    else
    {
        hlso=0; //nenabíjí
    }
}

void měření_výstupu(){ //metoda pro počítání proudu na výstupu
    odpor=0.03; //hodnota odporu na MOSFetu
    hl=ADC_Read(6);
    bat1==((hl*8.4)/1024); //hladina 1baterie
}

```

```

hl=ADC_Read(7);
bat2==(hl*8.4)/1024); //hladina 2baterie
hl=ADC_Read(8);
bat12==(hl*8.4)/1024); //hladina -baterii
Ibat1=(bat1-bat12)/odpor; // vypocet proudu z baterie 1
Ibat2=(bat2-bat12)/odpor; // vypocet proudu z baterie 2
}

void pocitani_proudu(){ //metoda pro pocitani proudu na vstupu
double konst=3.52; //konstanta pro vypocet
double Uout=4.2; //vystupni napeti
double Uin=panel; //vstupni napeti
if(panel<4.2) //podle vstupniho napeti vypocita proud
{
Inabij= (Uin/(konst))*((Uout-Uin)/Uout); //vzorecek ze
vstupniho nabijeni
}
else //podle vstupniho napeti vypocita proud
{
Inabij= (Uout/(konst))*((Uin-Uout)/Uin); //vzorecek ze
vstupniho nabijeni
}
}

void mereni_baterii(){ //zmeri hodnoty clanku baterii
PORTD.RD0=0; //nastavi na vystup jen 1 clanek 1 baterie
PORTD.RD1=1;
PORTD.RD2=1;
PORTD.RD3=1 ;
clanek4=ADC_Read(0); //zmeri 1 clanek 1 baterie

PORTD.RD0=1; //nastavi na vystup jen 2 clanek 1 baterie
PORTD.RD1=0 ;
clanek4=ADC_Read(1); //zmeri 2 clanek 1 baterie

PORTD.RD0=1 ; //nastavi na vystup jen 1 clanek 2 baterie
PORTD.RD1=1 ;
PORTD.RD2=0 ;
PORTD.RD3=1 ;
clanek4=ADC_Read(2); //zmeri 1 clanek 2 baterie

PORTD.RD2=1 ; //nastavi na vystup jen 2 clanek 2 baterie
PORTD.RD3=0;
clanek4=ADC_Read(3); //zmeri 2 clanek 2 baterie

}

void prepnuti_mezi_baterkama() //hodnoty napeti kdy prepnou mezi
sebou pro nabijeni
{
double hodprvni=0,hoddruhy=0;
hodprvni=clanek1+clanek2;
hoddruhy=clanek3+clanek4;
if(hodprvni+0.5<=hoddruhy) //pomer napeti kdy prepne baterku
{
PORTD.RD4=1; //prvni baterie
}
}

```

```

    PORTD.RD5=0; //druha baterie
}
if(hoddruhy+0.5<=hodprvni) //pomer napeti kdy prepne baterku
{
    PORTD.RD4=0; //prvni baterie
    PORTD.RD5=1; //druha baterie
}
}

void I2C_odeslani(){ //vyslani hodnot do PICu -nucený koment- demo
limit-zkrácená metoda pro odeslani hodnot do hlavniho PICu
/* // odesle hodnotu panelu a informace o stavu baterie
I2C1_Start(); // zacatek odesilani
I2C1_Wr(0x9A); // adresa zarizeni registru
I2C1_Wr(0x9A); // registr kam posilam
I2C1_Wr(panel); // poslana data
I2C1_Stop(); // konec odesilani

    if((clanek1+clanek2)<3.7 && (clanek3+clanek4)<3.7 )
//podminka pro hlidani low hladiny baterii
    {
I2C1_Start(); // zacatek odesilani
I2C1_Wr(0x9C); // adresa zarizeni
I2C1_Wr(0x9A); // registr kam posilam
I2C1_Wr('L'); // poslana data
I2C1_Stop(); // ikonec odesilani
    }
*/
} //nuceny koment -demo limit

void I2C_prijem (){ // prijem, jestli chci dve baterie

I2C1_Start(); // zacatek I2C
I2C1_Wr(0x9A); // adresa zarizeni
I2C1_Wr(0x02); // z jakeho registru
I2C1_Repeated_Start(); // opakovany start
I2C1_Wr(0x9B); // odeslani ze chci cist
prijmani = I2C1_Rd(0u); // co ctu a kam zapisuju
I2C1_Stop(); // konec

    Delay_100ms();

    if(prijmani==0b1111111) //podminka pro vstupni hlidani,
jestli chce dve baterie
    {
        dvebat=1; //zapne dve baterie
    }
    else
    {
        dvebat=0; //vypne dve baterie
    }
}

void odeslani(){ //metoda co spoji mereni a odeslani do hlavniho
PICu

```

```

mereni_panel();//zmeri hodnotu panelu
mereni_baterii();//zmeri hladinu baterii
mereni_vystupu();//zmeri proudy na vystupu
pocitani_proudu();//vypocita proud nabijeni baterii
I2C_odeslani();// odesle zmerene hodnoty
}

void spinani_druhe_baterie(){ //spina druhou baterii
mereni_vystupu();// zmeri proudy
while(dvebat==1||((clanek1+clanek2)<3.7 &&
(clanek3+clanek4)<3.7)) //podminka hlidani vypnuti dvou baterii
(dostanu signal nebo hladina je low)
{
I2C_prijem (); //kontrola, jestli nevypnul spinani druhe
baterie
mereni_vystupu();// zmeri proudy
if(Ibat1>Ibat2) //vyvazovani baterie
{
PORTD.RD5=0; //zapne druhou baterii
}
if(Ibat1<Ibat2)
{
PORTD.RD5=1; //vypne druhou baterii
}
}
PORTD.RD5=1; //vypnuti 2 baterie na vystupu
po ukonceni
}

void main() //hlavni metoda
{
I2C1_Init(100000); //inicializa I2C 10000 -
frekvence
ADC_Init(); //inicializace ADC
prvni_spusteni(); //metoda pro prvni spustění
odeslani(); // zmeri a odesle hodnoty

while(1) //opakovani
{
I2C_prijem();//hlidani jestli dve baterky zapnout
if(dvebat==1) //kdyz vstupni signal bude 1 -sepnuti
obou baterii
{
vypnuti_nabijeni(); //vypne nabijeni
spinani_druhe_baterie(); //dve baterie zaplo
}
else
{
hlidani_hladsolar();// zmeni hladinu panelu
if(hlso==1) //kdyz sviti
{
zapnuti_nabijeni(); //zapne nabijeni
odeslani(); // zmeri, odesle hodnoty
prepnuti_mezi_baterkama();// prepne baterie
prepinani_clanku(); //nabiji baterii
}
}
}
}

```

```
        else //kdyz nesviti
        {
            vypnuti_nabijeni();           //vypnout nabijeni
        }
    }
}
```