

```

/*Dekodovani DFC signalu*/
// Projekt MMIA 2013
// Martin Dujicek, Milan Rejnus

#include <avr/io.h>
#include <avr/interrupt.h>
#include "lcd_h.h"
#include "lcd_c.c"
#include "twi_c.c"
#include "twi_h.h"
#include <stdio.h>
#define F_CPU 16000000UL
#include <util/delay.h>
#include <stdint.h>

#define ADRESA_RTC 0b1010000 //adresa I2C rtc obvodu

typedef enum{ FALSE, TRUE } boolean_t;
typedef struct{
    uint8_t minuta;           //0-59
    uint8_t hodina;          //0-23
    uint8_t den;             //1-31
    uint8_t den_week;       //1-7 (Po-Ne)
    uint8_t mesic;          //1-12
    uint8_t rok;            //desitky a jednotky let aktualniho století
    uint8_t sec_selc;       //SEC 0b01 / SELC 0b10
    uint8_t zmena;          //znacka prechodu SEC SELC hodinu dopredu
    uint8_t antena;         // 0 hlavni, 1 zalozni vysilaci antena
}cas_t;

char buffer[16];
volatile uint16_t pocitadlo=0, puls, perioda;//, start, stop;
volatile boolean_t hotovo=FALSE, zacatek_minuta = FALSE,prvni_puls = FALSE,perioda_ok=FALSE,dataOK=FALSE;
volatile uint8_t minuta[60];
cas_t cas,cas_minuly;

volatile uint8_t tik_1Hz=0,zobraz_rok=0,zobraz_sec_selc=0,zobraz_zmena=0;

//prototypy funkci
uint8_t bool_na_cislo(boolean_t );
void posliRTC(cas_t );

```

```

void vyctiRTC(void );
uint8_t prepocet_hod_min(uint8_t );
uint8_t prepocet_rok_den(uint8_t ,uint8_t );
uint8_t prepocet_tyden_mesic(uint8_t , uint8_t );

void main(){

    volatile static uint8_t i=0, parita=0;//i ukazuje na pozici v poli (minutovem telegramu DCF)
    cas_t posledni_OK;//zde je ulozen posledni platny prijaty cas

    posledni_OK.hodina=0;posledni_OK.minuta=0;posledni_OK.den=0;posledni_OK.mesic=0;

    //nastaveni c/c1 do rezimu "capture mode"
    TIMSK=(1<<TICIE1); //povolene preruseni od capt mode
    TCCR1B=(1<<ICES1)|(1<<ICNC1)|(1<<CS12)|(0<<CS11)|(1<<CS10);
        //cas/cit1, input capture mode,
        //nastupna hrana ICES1 v 1, sestupna ICES1 v 0
        //input capture noise canceler
        //preddelicka 1024 => tovf 4,2s; tik 64us

    DDRD = 0x00;//port D vstupni, PD6 vystup z prijimace
    DDRB = 0xFF;//port B, vystup na spinani SJC (system jednotheho casu) - PB0, PB1
    DDRC = 0x00;//port C vstupni,
    PORTC = 0xFF;// zapnuti pull up portu C

    //init I2C
    TWBR=3;//400kHz
    //TWBR=18;//twi bit rate 100kHz na 16MHz

    //init displej

    lcd_init();//inicializace
    lcd_clrscr();//smazani obsahu,casova prodleva
    lcd_puts("Init...");//vypsani retezce

    sei();                //globalni povoleni preruseni
    while(1){

        //prepinani zobrazovaneho udaje podle tlacitka na PC7 (soucasny cas / pomocne informace)

```

```

if(bit_is_set(PINC,PINC7)){
    vyctiRTC();
}
else{
    //zobrazeni alternativnich informaci na displej po stisku tlacitka na PC7
    //zobrazeni casu posledni synchronizace
    sprintf(buffer, "%2d:%2d %2d.%2d.",posledni_OK.hodina,posledni_OK.minuta,posledni_OK.den,posledni_OK.mesic);
    lcd_firstline();
    lcd_puts(buffer);//vypis na lcd
}

//rozhodovani zda byla prijata minutova synchr. znacka
if(perioda_ok==TRUE){
    perioda_ok=FALSE;
    if(perioda > 28000){
        zacatek_minuta = TRUE;
        prvni_puls = TRUE;
        i=0;
    }
}

if(hotovo==TRUE){

    if(zacatek_minuta==TRUE){

        //po prijati pulzu, vyhodnoceni log. hodnoty prijateho bitu (100 nebo 200 ms)
        if((puls > 2250) && (puls < 4000)){minuta[i]=1;i++;}
        if((puls > 1100) && (puls < 2250)){minuta[i]=0;i++;}

        if(bit_is_clear(PINC,PINC7)){
            //zobrazeni alternativnich informaci na displej po stisku tlacitka na PC7
            sprintf(buffer, "puls = %u",puls);
            lcd_secondline();
            lcd_puts(buffer);//vypis na lcd
        }

        //kompletni minuta+++++++
        //zpracovani datagramu DCF
        if(i==59){
            zacatek_minuta=FALSE;

```

```

//kontrola parity
parita=0;
for(i=21;i<29;i++){//parita u minut
    parita = parita ^ bool_na_cislo(minuta[i]);
}
if(parita==0){//pokud sedi suda parita minuty, jde dal

    for(i=29;i<36;i++){//parita u hodin
        parita = parita ^ bool_na_cislo(minuta[i]);
    }
    if(parita==0){//pokud sedi suda parita hodiny, jde dal
        for(i=36;i<59;i++){//parita u data
            parita = parita ^ bool_na_cislo(minuta[i]);
        }
        if(parita==0){//pokud sedi suda parita data, jde dal
            //Vsechny parity OK
            cas_minuly=cas;//ulozeni predchozi hodnoty pro porovnaní

```

→

```
cas.minuta=minuta[21]*1+minuta[22]*2+minuta[23]*4+minuta[24]*8+minuta[25]*10+minuta[26]*20+minuta[27]*40;
```

```
cas.hodina=minuta[29]*1+minuta[30]*2+minuta[31]*4+minuta[32]*8+minuta[33]*10+minuta[34]*20;
```

```
cas.den=minuta[36]*1+minuta[37]*2+minuta[38]*4+minuta[39]*8+minuta[40]*10+minuta[41]*20;
```

```
cas.den_week=minuta[42]*1+minuta[43]*2+minuta[44]*4;
```

```
cas.mesic=minuta[45]*1+minuta[46]*2+minuta[47]*4+minuta[48]*8+minuta[49]*10;
```

```
cas.rok=minuta[50]*1+minuta[51]*2+minuta[52]*4+minuta[53]*8+minuta[54]*10+minuta[55]*20+minuta[56]*40+minuta[57]*80;
```

```
cas.sec_selc=minuta[17]*2+minuta[18];
```

```
cas.zmena=minuta[16];
```

```
cas_minuly.minuta+=1;//predchozi pripocitam o 1 pro porovnaní s aktualnim udajem
```

```

    if(cas_minuly.minuta==60){
        cas_minuly.minuta=0;
        cas_minuly.hodina+=1;
        if(cas_minuly.hodina==24) cas_minuly.hodina=0;
    }

```

→

```

    if(cas_minuly.minuta==cas.minuta && cas_minuly.hodina==cas.hodina &&
cas_minuly.rok==cas.rok && cas_minuly.sec_selc==cas.sec_selc && cas_minuly.zmena==cas.zmena){
        //posilani aktualniho zkontrolovaneho casu do RTC

```

```

        posliRTC(cas);
        posledni_OK=cas;
        zobraz_rok=cas.rok;
        zobraz_sec_selc=cas.sec_selc;
        zobraz_zmena=cas.zmena;
        lcd_secondline();
        lcd_puts("Data OK");
    }
    }else{lcd_firstline();lcd_puts("neOK datum");}
}else{lcd_firstline();lcd_puts("neOK hod");}

}else{lcd_firstline();lcd_puts("neOK min");}
}
//+++++
}
}
hotovo = FALSE;
}
}
}

```

```

//vektor preruseni inp compare mode TC1
ISR( TIMER1_CAPT_vect ){

    if (bit_is_set(TCCR1B,ICES1)){//prisla nastupna hrana
        perioda = ICR1;
        if(perioda>12000)perioda_ok =TRUE;
        TCNT1 = 0;//start=0

        TCCR1B^=(1<<ICES1);//prepnuti na sestupnou, pro zachyceni STOP
    }
    else{//prisla sestupna
        puls = ICR1;
        TCCR1B^=(1<<ICES1);//prepne na nastupnou, pro zachyceni start

        if(puls>1000){
            hotovo = TRUE;
        }
    }
}

```

```
}
```

```
//fce, ktera z TRUE/FALSE udela 1/0
```

```
uint8_t bool_na_cislo(booleant_t x){  
    static uint8_t y=0;
```

```
    if(x==TRUE)y=1;  
    else y=0;
```

```
    return y;
```

```
}
```

```
//posilani casu do RTC
```

```
void posliRTC(cas_t cas_k_poslani){  
    uint8_t statusreg_rtc=0;
```

```
    //vycteni status registru z obvodu rtc a jeho zastaveni (doporuceno v datasheetu)
```

```
    twi_start();  
    twi_address_w(ADRESA_RTC);  
    twi_write(0x00);  
    twi_start();  
    twi_address_read(ADRESA_RTC);  
    statusreg_rtc=twi_readNACK();  
    twi_stop();
```

```
    statusreg_rtc |= 0b10000000;
```

```
    twi_start();  
    twi_address_w(ADRESA_RTC);  
    twi_write(0x00);  
    twi_write(statusreg_rtc);  
    twi_write(0x00);//1/10s, 1/100s  
    twi_write(0x00);//1s  
    twi_write(prepcet_hod_min(cas_k_poslani.minuta));//min  
    twi_write(prepcet_hod_min(cas_k_poslani.hodina));//hod  
    twi_write(prepcet_rok_den(cas_k_poslani.den,cas_k_poslani.rok));//den, rok  
    twi_write(prepcet_tyden_mesic(cas_k_poslani.mesic,cas_k_poslani.den_week));//mesic, den v tydnu  
    twi_stop();
```

```
    //opetovne povoleni citani rtc
```

```
    twi_start();  
    twi_address_w(ADRESA_RTC);
```

```

    twi_write(0x00); //naadresovani minut
    twi_start();
    twi_address_read(ADRESA_RTC);
    statusreg_rtc=twi_readNACK();
    twi_stop();

    statusreg_rtc &= 0b01111111;

    twi_start();
    twi_address_w(ADRESA_RTC);
    twi_write(0x00); //naadresovani minut
    twi_write(statusreg_rtc);
    twi_stop();

    return;
}

//vycteni hodnot z RTC a zobrazeni udaju na displeji
/*
    |-----|
    | hod:min:sec den_v_tydnu SEC/SELC |
    | den.mesic.rok ant. prechod_casu |
    |-----|
*/
void vyctiRTC(){
    uint8_t vycteno[5], vterina_minula=0;
    cas_t cas_prevedeny;

    vterina_minula=vycteno[0];

    twi_start();
    twi_address_w(ADRESA_RTC);
    twi_write(0x02); //nadresovani minut
    twi_start();
    twi_address_read(ADRESA_RTC);
    vycteno[0]=twi_read();
    vycteno[1]=twi_read();
    vycteno[2]=twi_read();
    vycteno[3]=twi_read();
    vycteno[4]=twi_readNACK();
    twi_stop();

```

```

//prevedeni z formatu ulozeneho v rtc an format pro zobrazeni
cas_prevedeny.minuta = vycteno[1];
cas_prevedeny.hodina = vycteno[2];
cas_prevedeny.rok = cas.rok;
cas_prevedeny.den = vycteno[3] & 0b00111111;
cas_prevedeny.den_week=(vycteno[4] & 0b11100000) / 32;
cas_prevedeny.mesic=vycteno[4] & 0b00011111;

//osetreni prechodu roku
if(cas_prevedeny.den==0x01 && cas_prevedeny.mesic==0x01 && cas_prevedeny.hodina==0x00 && cas_prevedeny.minuta==0x00)
    zobraz_rok=0;

if(vterina_minula != vycteno[0]){
    sprintf(buffer, "%2X:%2X:%2X ", cas_prevedeny.hodina, cas_prevedeny.minuta, vycteno[0]);
    lcd_firstline();
    lcd_puts(buffer); //vypis na lcd

    switch(cas_prevedeny.den_week){
        case 6: lcd_puts("Ne"); break;
        case 0: lcd_puts("Po"); break;
        case 1: lcd_puts("Ut"); break;
        case 2: lcd_puts("St"); break;
        case 3: lcd_puts("Ct"); break;
        case 4: lcd_puts("Pa"); break;
        case 5: lcd_puts("So"); break;
        default: lcd_puts("Po"); break;
    }

    switch(zobraz_sec_selc){
        case 1: lcd_puts(" SEC"); break;
        case 2: lcd_puts(" SELC"); break;
        default: break;
    }

    sprintf(buffer, "%2X.%2X. ", cas_prevedeny.den, cas_prevedeny.mesic);
    lcd_secondline();
    lcd_puts(buffer); //vypis na lcd

    if(zobraz_rok!=0){
        sprintf(buffer, "20%2d", zobraz_rok);
        lcd_puts(buffer); //vypis na lcd
    }
}

```



```

//zobrazeni udaje o hlavni / zalozni antene
if(cas.antena){
    lcd_gotoxy(13,1);
    lcd_puts("!");//vypis na lcd
}

//indikace chystaneho prechodu SEC <-> SELC
if(zobraz_zmena){
    lcd_gotoxy(15,1);
    lcd_puts("!");//vypis na lcd
}

//generovani signalu pro pripojene podruzne hodiny (System Jednotheho Casu)
if(vycteno[0]==0x01)
    PORTB=0;

if(vycteno[0]==0x59){
    if(cas_prevedeny.minuta % 2){
        PORTB=0x01;
    }
    else{
        PORTB=0x02;
    }
}
}

return;
}

//prepocet do formatu pro RTC - zapis
uint8_t prepocet_hod_min(uint8_t cas_in){

    uint8_t prepocteno=0;
    prepocteno = cas_in / 10;
    cas_in = cas_in - prepocteno*10;
    prepocteno*=16;
    prepocteno+=cas_in;

    return prepocteno;
}

```

```
uint8_t prepocet_rok_den(uint8_t cas_in, uint8_t roky){  
    uint8_t prepocteno=0;  
    prepocteno = cas_in / 10;  
    cas_in = cas_in - prepocteno*10;  
    prepocteno*=16;  
    prepocteno+=cas_in;  
  
    prepocteno += (roky%4)*64;  
  
    return prepocteno;  
}  
  
uint8_t prepocet_tyden_mesic(uint8_t cas_in, uint8_t den_v_tydnu){  
    uint8_t prepocteno=0;  
    prepocteno = cas_in / 10;  
    cas_in = cas_in - prepocteno*10;  
    prepocteno*=16;  
    prepocteno+=cas_in;  
  
    prepocteno += (den_v_tydnu-1)*32;  
  
    return prepocteno;  
}
```