```
//------------------------------------------------------------------------------------- //
//      AVR e Arduino: Técnicas de Projeto, 2a ed. - 2012.                               //
//------------------------------------------------------------------------------------- //
//------------------------------------------------------------------------------------- //
//      DS1307 - RTC com LCD 16 x 2 utilizando I2C                                       //
//------------------------------------------------------------------------------------- //

#include "def_principais.h"
#include "LCD.h"
#include "DS1307.h"

unsigned char FLAGS;
//----- flags de leitura dos botoes -----
#define flag_menu 0
#define flag_mais 1
#define flag_menos 2
#define flag_enter 3
//----- flags para sistema de controle -----
#define menu_LED 4
#define on_off 5
#define alerta_on 6
#define luz_branca_on 7

extern unsigned char flag_pontos, cont, tempo,tempoRCT[7];

unsigned char horas=12,minutos=0,digitos[2],ledR[3],ledG[3],ledB[3];
unsigned char cont_mais_menos=1,cont_menu,cont_enter,branco=25,soneca,travo,RED=0,BLUE=0,GREEN=0,cont_LED=0, ↙
    piscaR=0,piscaG=0,piscaB=0,R=0,G=0,B=0,buzz_passo,buzz_ativo;

ISR(INT0_vect);

int main()
{
    DDRB    = 0b11111111;
    PORTB   = 0b11010011;       //pb2 pb3 saida do pwm-led (pb1)   pb5 - buzzer
    DDRC    = 0b11101111;
    PORTC   = 0xFF;
    DDRD    = 0b00001011;       //pd3 luzbranca
    PORTD   = 0b11110111;

    inic_TWI();
    ISR(INT0_vect);     // Interrupção
    EICRA = ((1<<ISC00) | (0<<ISC01));
    EIMSK = 0X01;
    sei();

    inic_LCD_4bits();
    inic_TWI();
    escreve_DS1307(0x00, 0x00);
    escreve_DS1307(0x07,0b00010000);    //habilita 1Hz no pino SOUT do DS1307

//------------------------------------------------------------------------------------------------------------↙
    -------------------//
//                                                  ||      P W M       ||                                  ↙
                      //
//------------------------------------------------------------------------------------------------------------↙
    -------------------//

TCCR2A = 0b00000011;//TCCR2A = 0b00000011;      //PWM não invertido no pino OC0A e OC0B esta desligado //    ↙
    Para ligar deve ficar assim TCCR2A = 0b10000011;
TCCR2B = 0b00000100;//TCCR2B = 0b00000100;      //liga TC0, prescaler = 64
OCR2A = 0;                  //controle do ciclo ativo do PWM 0C0A
OCR2B = 0;


TCCR1A = 0b00000001;        //PWM não invertido nos pinos OC1A e OC1B // Para ligar deve ficar assim TCCR1A =↙
     0b10100001;
TCCR1B = 0b00001011;        //liga TC1, prescaler = 64
OCR1A = 0;                  //controle do ciclo ativo do PWM 0C1A
OCR1B = 0;                  //controle do ciclo ativo do PWM 0C1B

//while(1);
//------------------------------------------------------------------------------------------------------------↙
    ----------------------
```

```
//---------------------------------usando a sequencia sem necessidade de interromper-----------------------
    ----------------------
clr_bit(FLAGS,flag_menu);
buzz_ativo=5;
desliga_PWM_2B();
desliga_PWM_2A();
desliga_PWM_1A();
desliga_PWM_1B();
while (1)
{
    if (tst_bit(FLAGS,on_off))                              //alarme armado, verifica os horarios
    {
        if (horas==tempoRCT[2])
        {
            if (soneca==tempoRCT[1])
            {
                if (tempoRCT[0]==0)
                {
                    set_bit(FLAGS,alerta_on);               //ativa o switch do despertador
                    cmd_LCD(0x8E,0);
                    cmd_LCD('*',1);

                    if (minutos==soneca)                    //modelo inicial de acordar com LEDs
                    {
                        BLUE=50;    liga_PWM_1A();
                        GREEN=10;   liga_PWM_2A();
                        desliga_PWM_1B();
                    }
                    else
                    {
                        GREEN=250;  liga_PWM_1B();  piscaG=3;
                        BLUE=150;   liga_PWM_1A();
                        RED=50;     liga_PWM_2A();  piscaR=2;
                    }
                    clr_bit(FLAGS,flag_menu);
                    clr_bit(FLAGS,flag_enter);
                    clr_bit(FLAGS,flag_menos);
                    clr_bit(FLAGS,flag_mais);
                }
            }
        }
    }

    if (!tst_bit(PIND,MENU))
    {
        if (tst_bit(FLAGS,flag_menu))                       //sai do 1o menu
        {
            clr_bit(FLAGS,flag_menu);
            clr_bit(FLAGS,flag_enter);
            cont_menu=0;
            cont_enter=1;
        }
        else
        {
            set_bit(FLAGS,flag_menu);                       //liga o 1o menu
            cont_enter=0;
            flag_pontos=1;
            clr_bit(FLAGS,flag_enter);                      //desliga o enter, ou melhor, ele retorna
        }
        while(!tst_bit(PIND,MENU)) travado();               //verifica se o botao travou
        travo=0;
        cmd_LCD(0x01,0);
        /*if (tst_bit(FLAGS,flag_menu))                     /--/codigo inutil, porem sem ele da pau na
    simulacao
        {
            cmd_LCD(0x8D,0);
            escreve_LCD("5");
        }*/
    }
    if (!tst_bit(PIND,ENTER))
    {
        if (cont_menu==0)                                           //funcao para ligar/desligar as luzes
    brancas, se o 1o menu estiver inactivo
        {
```

```
            liga_desliga();                                 //aciona/desliga LED branco
        }
        set_bit(FLAGS,flag_enter);                      //entra no menu escolhido
        clr_bit(FLAGS,flag_menu);                       //trava o 1o menu
        cont_enter++;                                   //escolhe o menu de ajustar o alarme
        while(!tst_bit(PIND,ENTER)) travado();
        travo=0;
        cmd_LCD(0x01,0);
    }
    if (!tst_bit(PIND,MAIS))
    {
        if (tst_bit(FLAGS,luz_branca_on)&&(cont_menu==0))
        {
            OCR2B = branco + 25;
            branco = branco + 25;
        }
        cont_mais_menos++;                              //caminha no 1o menu
        set_bit(FLAGS,flag_mais);                       //seta horas/minutos do alarme
        while(!tst_bit(PIND,MAIS)) travado();
        travo=0;
        cmd_LCD(0x01,0);
    }
    if (!tst_bit(PIND,MENOS))
    {
        if (tst_bit(FLAGS,luz_branca_on)&&(cont_menu==0))
        {
            OCR2B = branco - 25;
            branco = branco - 25;
        }
        cont_mais_menos--;                              //caminha no 1o menu
        set_bit(FLAGS,flag_menos);                      //seta horas/minutos do alarme
        while(!tst_bit(PIND,MENOS)) travado();
        travo=0;
        cmd_LCD(0x01,0);
    }
//--------------------apenas testando botoes e acionando os respectivos flags-------------------------
//--------------------a seguir as relacoes e acoes dos flags-----------------------------------------
if ((cont>7||cont_enter>7))                             //controle do menu de ajuste do relogio (7 posicoes)
    {
        cont_enter = cont = 1;
        /*flag_pontos = 1;                              //habilita novamente os pontos e tracos*/
    }
if ((cont_mais_menos>3)&&(cont_mais_menos<254))         //controle do 1o menu
    {cont_mais_menos=1;}
if ((cont_mais_menos>254)||(cont_mais_menos<=0))
    {cont_mais_menos=3;}
if (tst_bit(FLAGS,flag_menu))                           //menu 1o ativo/travado
    {
        cont_menu=cont_mais_menos;              //atualiza o menu inicial segundo botao "enter" estiver inactivo, senao, ele trava a variavel

    }
//-----------------------------------------------------menu do alarme acionado-----------------------------------
    --------------------
            if(tst_bit(FLAGS,alerta_on))
            {
                    //cmd_LCD(0x8F,0);      --teste--
                    //cmd_LCD('X',1);
                    buzz_passo++;                           //PWM por programa para o BUZZER
                    if (buzz_passo==BUZ_rsl)                // BUZ_rsl em 25
                    {
                        buzz_passo=0;
                        set_bit(PORTB,BUZZER);
                    }
                    if (buzz_passo==buzz_ativo)             //ativo em 5, depois do soneca ativo = 10... depois 15...
                    {                                       // 5 de 25, depois 10 de 25 ...
                        //buzz_passo=0;
                        clr_bit(PORTB,BUZZER);
                    }
                    if((tst_bit(FLAGS,on_off))&&((!tst_bit(FLAGS,flag_menos))&&(!tst_bit(FLAGS, flag_mais))))
                    {
```

```
                            cmd_LCD(0x8C,0);
                            escreve_LCD("+2min");
                        }
                        if ((tst_bit(FLAGS,flag_menu))||(tst_bit(FLAGS,flag_enter)))                //desliga
    o alarme por 2min
                        {
                            clr_bit(FLAGS,alerta_on);
                            clr_bit(PORTB,BUZZER);
                            desliga_PWM_1A();
                            desliga_PWM_1B();
                            desliga_PWM_2A();
                            if (tst_bit(FLAGS,on_off))
                            {
                                soneca=soneca+2;
                                buzz_ativo = buzz_ativo +5;

                            }
                            //_delay_ms(200);
                        }
                        if ((tst_bit(FLAGS,flag_menos))||(tst_bit(FLAGS,flag_mais)))                //permite
    desligar o alarme definitivamente
                        {
                            clr_bit(FLAGS,flag_mais);
                            clr_bit(FLAGS,flag_menos);
                            cpl_bit(FLAGS,on_off);
                            cmd_LCD(0x8C,0);
                            escreve_LCD("OFF");
                            //_delay_ms(200);
                        }
                }
//-------------------------------------------------------------------fim do alarme-----------------------
    --------------
//-----------------------------------------------------------PISCA LED---------------------------------
    --------------
if (R <= RED)
    {R = R + piscaR;}                       //efeito de piscar o led em progressao
else
    {R = 0;}
OCR2A = R;                                  // Valor do ciclo ativo do TC2A
if (G <= GREEN)
    {G = G + piscaG;}
else
    {G = 0;}
OCR1B = G;
if (B <= BLUE)
    {B = B + piscaB;}
else
    {B = 0;}
OCR1A = B;
//-----------------------------------------------------FIM DO PWM PISCA LED----------------------------
    ----------------
//  else
/*      cont_menu=0;            //menu_off da um flag para que o botao menu, saia do 1o menu e volte a
    apresentar somente horas*/
if (!tst_bit(FLAGS,menu_LED))
{
    //cmd_LCD(0x80,0); if (tst_bit(FLAGS,luz_branca_on)) {escreve_LCD("*");} //* indicando sinal de luz
    branca ligado
    //ja fiz esta merda, esta eu outro lugar...
    switch (cont_menu)                      //menu inicial, sendo o default o padrao do funcionamento do
    relogio
    {
        case 1:                                                             //menu de ajustar relógio
                    sei();
                    mostra_pontos(flag_pontos);
                    mostra_tempo();
                    if (!tst_bit(FLAGS,flag_enter))                         //enter em zero
                    {   cmd_LCD(0x8A,0);
                        escreve_LCD("m +- <");
                        cmd_LCD(0xC0,0);
                        escreve_LCD(" Ajustar Relogio");
                    }
                    else                                                    //executa o ajuste do tempo
    estilo padrao/original
```

```
                {
                    mostra_tempo_parte2();
                    mostra_pontos_parte2(flag_pontos);
                    flag_pontos = 0;                                    //avisa para nao imprimir mais os
 pontos e tracos
                    cont = cont_enter;                                  //conta o nr. de vezes que o
botao SELEÇAO foi pressionado
                    mostra_pontos_de_ajuste(cont);
                    alerta_display(cont);                               //coloca a seta no local para
ajuste do tempo

                    if (flag_pontos==0)
                    {
                        if(tst_bit(FLAGS,flag_mais))
                        {
                            clr_bit(FLAGS,flag_mais);
                            ajusta_tempo2(cont);                            //ajusta RTC botao mais
                        }
                        if(tst_bit(FLAGS,flag_menos))
                        {
                            clr_bit(FLAGS,flag_menos);
                            ajusta_tempo2_menos(cont);                      //ajusta RTC botao menos
                        }
                    }//if !flag
                } //fim do else
                break;
    case 2:     //menu ajuste do alarme
                cmd_LCD(0x8A,0);
                escreve_LCD("m +- <");
                cmd_LCD(0xC0,0);
                escreve_LCD(" Ajustar Alarme ");
                if (tst_bit(FLAGS,flag_enter))
                {                                       //executa o ajuste do alarme segundo algumas
funcoes criadas
                    cli();
                    //flag_pontos=0;
                        ident_num(minutos,digitos);     //transforma a variável horas em bits
                    cmd_LCD(0x84,0);                //posiciona
                    cmd_LCD(digitos[1],1);          //primeiro digito
                    cmd_LCD(digitos[0],1);          //segundo digito
                    _delay_ms(10);
                        ident_num(horas,digitos);       //transforma a variável minutos em bits
                    cmd_LCD(0x81,0);                //posiciona
                    cmd_LCD(digitos[1],1);          //primeiro digito
                    cmd_LCD(digitos[0],1);          //segundo digito
                    _delay_ms(10);
                    cmd_LCD(0x87,0);
                    if (tst_bit(FLAGS,on_off))
                    escreve_LCD("ON ");
                    else
                    escreve_LCD("OFF");
                    switch (cont_enter)                     //entra no menu do alarme para alterar o
valor
                        {
                            case 4:
                                    cmd_LCD(0x80,0); cmd_LCD(0x7E,1);
                                    cmd_LCD(0x83,0); cmd_LCD(':',1);
                                    horas_();                               //altera as horas do alarme
                                    break;
                            case 5:
                                    cmd_LCD(0x83,0); cmd_LCD(0x7E,1);
                                    minutos_();                             //altera os minutos do alarme
                                    break;
                            case 3:
                                    cmd_LCD(0x83,0); cmd_LCD(':',1);
                                    cmd_LCD(0x86,0); cmd_LCD(0x7E,1);
                                    if((tst_bit(FLAGS,flag_mais))||(tst_bit(FLAGS,flag_menos)))
                                    {
                                        clr_bit(FLAGS,flag_mais);
                                        clr_bit(FLAGS,flag_menos);
                                        cpl_bit(FLAGS,on_off);                      //altera o estado on/
off
                                    }
                                    break;
```

```
                                default:
                                        cont_enter=3;
                                        break;
                                }//----end switch-----
                        }//----end if -----
                        else
                        {
                            //flag_pontos=1;
                            sei();
                            mostra_pontos(flag_pontos);
                            mostra_tempo();
                        }
                        break;
                case 3:     //menu de acionamento de LEDs sequenciais
                        sei();
                        cmd_LCD(0x8A,0);
                        escreve_LCD("m +- <");
                        cmd_LCD(0xC0,0);
                        escreve_LCD(" Acionar LEDs ->");
                        mostra_pontos(flag_pontos);
                        mostra_tempo();
                        if (tst_bit(FLAGS,flag_enter))
                        {
                            set_bit(FLAGS,menu_LED);            //executa menu de sequencias diferentes de LEDs, ↙
    com outro switch
                            cont_LED=1;
                            clr_bit(FLAGS,flag_enter);
                            clr_bit(FLAGS,flag_mais);
                            clr_bit(FLAGS,flag_menos);
                            cmd_LCD(0x01,0);
                        }
                        break;
                default:
                        sei();
                        cont_menu=0;
                        cont_mais_menos=1;
                        mostra_pontos(flag_pontos);                         //superior pontos
                        mostra_pontos_parte2(flag_pontos);                  //inferior pontos
                        mostra_tempo();                                     //superior tempo
                        mostra_tempo_parte2();                              //inferior tempo
                        if (tst_bit(FLAGS,luz_branca_on))                   //luz branca ligada
                        { cmd_LCD(0x8E,0); cmd_LCD('*',1); }
                        break;
    }//-fim do switch --------
}//-----fim do if menu_ativo--
else
{
    if (!tst_bit(FLAGS,flag_enter))
    {
        if (tst_bit(FLAGS,flag_mais))
        {
            cont_LED++;
            clr_bit(FLAGS,flag_mais);
            if ((cont_LED>5)&&(cont_LED<250))
            cont_LED = 1;
        }
        if (tst_bit(FLAGS,flag_menos))
        {
            cont_LED--;
            clr_bit(FLAGS,flag_menos);
            if (cont_LED<1)
            cont_LED = 5;
        }
    }
    cli();
    cmd_LCD(0x8A,0);
    escreve_LCD("m +- <");
    switch(cont_LED)
    {
            case 1:
                    {
                    if (tst_bit(FLAGS,flag_enter))
                    {
                            cmd_LCD(0x80,0);
```

```
                        escreve_LCD(">> RED <<");
                        cmd_LCD(0xC0,0);
                        escreve_LCD("RATIO: ");
                        PWM_display(&RED);                      //funcao de alterar o valor do RED para ↙
    mais ou menos
                        R = RED;
                            ident_num(RED,ledR);                //char de 3 digitos
                        cmd_LCD(0xC7,0);
                        cmd_LCD(ledR[2],1);
                        cmd_LCD(ledR[1],1);
                        cmd_LCD(ledR[0],1);
                            if ((cont_enter%2)==0)                  //verificacao somente para ON/OFF
                            {
                                cmd_LCD(0xCC,0);
                                escreve_LCD(" ON");
                                liga_PWM_2A();
                            }
                            else
                            {
                                cmd_LCD(0xCC,0);
                                escreve_LCD("OFF");
                                desliga_PWM_2A();
                            }
                    }
                    else
                    {
                        cmd_LCD(0x80,0);
                        escreve_LCD("SET RED");
                        cmd_LCD(0xC0,0);
                        escreve_LCD("SET GREEN");
                    }
                    break;
                    }
            case 2:
                    {
                    if (tst_bit(FLAGS,flag_enter))
                    {
                        cmd_LCD(0x80,0);
                        escreve_LCD(">>GREEN<<");
                        cmd_LCD(0xC0,0);
                        escreve_LCD("RATIO: ");
                        PWM_display(&GREEN);                    //funcao de alterar o valor do GREEN para ↙
    mais ou menos
                        G = GREEN;
                            ident_num(GREEN,ledG);
                        cmd_LCD(0xC7,0);
                        cmd_LCD(ledG[2],1);
                        cmd_LCD(ledG[1],1);
                        cmd_LCD(ledG[0],1);
                        if ((cont_enter%2)==0)
                        {
                            cmd_LCD(0xCC,0);
                            escreve_LCD(" ON");
                            liga_PWM_1B();
                        }
                        else
                        {
                            cmd_LCD(0xCC,0);
                            escreve_LCD("OFF");
                            desliga_PWM_1B();
                        }
                    }
                    else
                    {
                        cmd_LCD(0x80,0);
                        escreve_LCD("SET GREEN");
                        cmd_LCD(0xC0,0);
                        escreve_LCD("SET BLUE");
                    }
                    break;
                    }
            case 3:
                    {
                    if (tst_bit(FLAGS,flag_enter))
```

```
                    {
                        cmd_LCD(0x80,0);
                        escreve_LCD(">> BLUE<<");
                        cmd_LCD(0xC0,0);
                        escreve_LCD("RATIO: ");
                        PWM_display(&BLUE);                 //funcao de alterar o valor do BLUE para mais ou ↙
    menos
                        B = BLUE;
                            ident_num(BLUE,ledB);
                        cmd_LCD(0xC7,0);
                        cmd_LCD(ledB[2],1);
                        cmd_LCD(ledB[1],1);
                        cmd_LCD(ledB[0],1);
                        if ((cont_enter%2)==0)
                        {
                            cmd_LCD(0xCC,0);
                            escreve_LCD(" ON");
                            liga_PWM_1A();
                        }
                        else
                        {
                            cmd_LCD(0xCC,0);
                            escreve_LCD("OFF");
                            desliga_PWM_1A();
                        }
                    }
                    else
                    {
                        cmd_LCD(0x80,0);
                        escreve_LCD("SET BLUE");
                        cmd_LCD(0xC0,0);
                        escreve_LCD("SET PISCA LEDS");
                    }
                    break;
                    }
            case 4:
                    {
                        if (tst_bit(FLAGS,flag_enter))
                        {
                            cmd_LCD(0x80,0);
                            escreve_LCD("PISCA LEDS");
                            cmd_LCD(0xC0,0);
                            escreve_LCD(" R:");
                            cmd_LCD(0xC5,0);
                            escreve_LCD(" G:");
                            cmd_LCD(0xCA,0);
                            escreve_LCD(" B:");
                            cmd_LCD(0xCF,0);
                            escreve_LCD(" ");
                                ident_num(piscaR,digitos);
                            cmd_LCD(0xC3,0);
                            //cmd_LCD(digitos[1],1);
                            cmd_LCD(digitos[0],1);
                                ident_num(piscaG,digitos);
                            cmd_LCD(0xC8,0);
                            //cmd_LCD(digitos[1],1);
                            cmd_LCD(digitos[0],1);
                                ident_num(piscaB,digitos);
                            cmd_LCD(0xCD,0);
                            //cmd_LCD(digitos[1],1);
                            cmd_LCD(digitos[0],1);
                            switch (cont_enter)                     //submenu
                            {
                                case 1:
                                            PiscaRGB(&piscaR);
                                            break;
                                case 2:
                                            PiscaRGB(&piscaG);
                                            break;
                                case 3:
                                            PiscaRGB(&piscaB);
                                            break;
                                default :
                                            if (cont_enter<1)
```

```
                                        {
                                            cont_enter=3;
                                        }
                                        if (cont_enter>3)
                                        {
                                            cont_enter =1;
                                        }
                                        break;
                                }   //fim do switch de submenu

                            }
                            else
                            {
                            cmd_LCD(0x80,0);
                            escreve_LCD("PISCA LEDS");
                            cmd_LCD(0xC0,0);
                            escreve_LCD("SAIR            ");
                            }
                            break;
                        }
                    case 5:
                        {
                            if (tst_bit(FLAGS,flag_enter))
                            {
                                clr_bit(FLAGS,menu_LED);
                                clr_bit(FLAGS,flag_menu);
                                clr_bit(FLAGS,flag_enter);
                                clr_bit(FLAGS,flag_mais);
                                clr_bit(FLAGS,flag_menos);
                                cont_LED=1;
                                cmd_LCD(0x01,0);
                            }
                            else
                            {
                                cmd_LCD(0x80,0);
                                escreve_LCD("SAIR?");
                            }
                            break;
                        }
                    default:
                        {
                            cont_LED=1;
                            break;
                        }
                }//--fim do switch--------

}//-----fim do else-----------
}//-----fim do while(1)-------
}//-----fim do int main-------

//----- funcao liga_desliga LED branco----------------------------------
 void liga_desliga ()
{
    while(!tst_bit(PIND,ENTER))
    {
        if (cont_enter<300)
        {
            cont_enter++;
            _delay_ms(10);
            if (cont_enter>=200)
                {
                //cmd_LCD(0x80,0);
                //escreve_LCD("<><><><><><><><>");  //desliga a luminaria
                //cmd_LCD(0xC0,0);
                //escreve_LCD("<><><><><><><><>");
                cpl_bit(FLAGS,luz_branca_on);       //flag dos botoes mais/menos
                    if (tst_bit(FLAGS,luz_branca_on))
                    {   liga_PWM_2B();
                    OCR2B = branco;     }
                    else
                    {   desliga_PWM_2B();
                        OCR2B = 0;
                    branco = 25;        }
                return;
```

```
                }
        }
        else
        { cont_enter=0; }
        //atraso, 3s para desligar os leds
    }

}
 //---avisa que o botao travou-------------------------------------------------------------
void travado()
{
    travo++;
    _delay_ms(10);
    if (travo>250)
    {
        cmd_LCD(0x80,0);
        escreve_LCD("   POR FAVOR,   ");
        cmd_LCD(0xC0,0);
        escreve_LCD(" SOLTE O BOTAO! ");
    }
}
//---funcao de ajuste do alarme-----------------------------------------------------------
void minutos_()                 //incrementa minutos no display
{
                if (tst_bit(FLAGS,flag_mais))
                {
                    minutos++;
                    clr_bit(FLAGS,flag_mais);
                    if (minutos > 59)
                    minutos = 0;
                }
                if (tst_bit(FLAGS,flag_menos))
                {
                    minutos--;
                    clr_bit(FLAGS,flag_menos);
                    if (minutos > 254)
                    minutos = 59;
                }
        soneca=minutos;
}
void horas_()
{
                if (tst_bit(FLAGS,flag_mais))
                {
                    horas++;
                    clr_bit(FLAGS,flag_mais);
                    if (horas > 23)
                    horas = 0;
                }
                if (tst_bit(FLAGS,flag_menos))
                {
                    horas--;
                    clr_bit(FLAGS,flag_menos);
                    if (horas > 254)
                    horas = 23;
                }
}
//------------------------Incremento da variavel controle dos LEDs PWM------------------
void PWM_display(unsigned char* COR)
{
    //cmd_LCD(0xCA,0); escreve_LCD("");
    if (tst_bit(FLAGS,flag_mais))
    {
        //  cmd_LCD(0xCA,0); escreve_LCD("+");      --teste--
        *COR=*COR+25;                       //falta jogar o valor setado para o carregador PWM
        clr_bit(FLAGS,flag_mais);
    }

    if (tst_bit(FLAGS,flag_menos))
    {
        *COR=*COR-25;                   //o valor da variavel RGB qndo estoura 256 nao vira 0xx
        clr_bit(FLAGS,flag_menos);
    }
```

```
}
void PiscaRGB (unsigned char* COR)
{
    if (tst_bit(FLAGS,flag_mais))
    {
        *COR = *COR+1;
        clr_bit(FLAGS,flag_mais);
        if ((*COR>3)&&(*COR<255))
        {
            *COR = 0;
        }
    }
    if (tst_bit(FLAGS,flag_menos))
    {
        *COR=*COR-1;
        clr_bit(FLAGS,flag_menos);
        if (*COR>250)
        {
            *COR = 3;
        }
    }
}
//---------------------------------------------------------------------------------
ISR(INT0_vect)
{
    unsigned char sreg;

    sreg = SREG;                //salva SREG porque a rotina pode alterar o seu valor    ----> Devido a      ↙
    interrupção do I²C

    clr_bit(EIMSK,INT0);        //desabilita INTO para que ele não chame a si mesmo    ----> Devido a        ↙
    interrupção do I²C
    sei();                      //habilita a interrupção geral, agora INT1 pode interromper INT0    ---->    ↙
    Devido a interrupção do I²C

    ler_convert_tudo(0x00);     //gasta um bom tempo para ler o RTC! Mais lento na simulação quando entra    ↙
    aqui
    //mostra_tempo();
    //mostra_pontos(flag_pontos);   //flag indica se deve ligar os pontos e tracos
    _delay_ms(2);

    set_bit(EIMSK,INT0);        //habilita novamente a interrupção INT0    ----> Devido a interrupção do I²C

    SREG = sreg;                //restaura o valor de SREG que pode ter sido alterado    ----> Devido a      ↙
    interrupção do I²C
}
```