

```

#include <avr/interrupt.h>
#include <avr/io.h>
#include <util/delay.h>
#include "lcd.h"
#include "ds1307.h"

#define F_CPU 1000000

volatile unsigned char count = 50, flag_int = 0, num_click_rega = 1,
num_click_seguranca = 1, flag_rega = 0, flag_seguranca = 0, relógio_click = 0,
tecla_posicao = 0;
unsigned char segundo = 0, minuto = 0, hora = 0, dia = 0, data = 0, mes = 0, ano =
0, variavel_hora, variavel_minuto, variavel_segundo, variavel_dia, horas, minutos,
segundos, dias, digito_hora, digito_minuto, digito_segundo, rega1_hora_inicio,
rega1_minuto_inicio, rega1_hora_fim, rega1_minuto_fim, rega2_hora_inicio,
rega2_minuto_inicio, rega2_hora_fim, rega2_minuto_fim, seguro1_hora_inicio,
seguro1_minuto_inicio, seguro1_hora_fim, seguro1_minuto_fim, seguro2_hora_inicio,
seguro2_minuto_inicio, seguro2_hora_fim, seguro2_minuto_fim;
unsigned char hora_inicio_rega, hora_fim_rega, minuto_inicio_rega, minuto_fim_rega,
hora_inicio_seguro, hora_fim_seguro, minuto_inicio_seguro, minuto_fim_seguro;

void inic_timer(void)
{
    DDRB = 0b11111111; // Port B
    como saida
    DDRC = 0b00001111; //bits de
0 a 3 como saida para leds e bits 4 e 5 como entrada para RTC
    PORTC = 0;
    //portc iniciado como 0
    PORTB = 1;
    //Portb iniciado a 1
    TCCR0A |= (1 << WGM01); // Normal port
operation, modo CTC
    TCCR0B |= (1 << CS01)|(1 << CS00); // Modo CTC,
prescaler 64
    TIMSK0 |= (1 << OCIE0A); // Activa o
disparo da interrupcao
    OCR0A = 155; // fclock
= 1MHz, prescaler 64, atraso 10ms
    sei(); //
Activa as interrupcoes globais
}

ISR (TIMER0_COMPA_vect)
{
    // Função que o CPU executa sempre que uma interrupcao ocorre

    count--; //
Decrementa o contador
    if(count == 0) //
Condição que verifica o estado do contador, caso seja 0, atraso de 500ms (50 *
10ms)
    {

```

```

        flag_int = 1;                                // Coloca
a flag_int = 1 representando o atraso de 500ms
        count = 50;                                //
Reinicia o contador
    }
}

void inicio_teclado(void)                            //Função de
inicialização do teclado
{
    DDRD = 0b11110000;                               //Define
as colunas (bits 4 a 7) como saidas e linhas (bits 0 a 3) como entradas
    PORTD = 0b00001111;                               //Ao
inicializa a 1 as linhas ativa os respectivos pull-up
}

void acerta_hora(void)                                //Função
acertar hora
{
    int teste = 0, tecla_numero = 0;                  //inicialização a 0 das
variaveis
    LCDWriteStringXY(0,0,"Acerte a hora: ");          //Imprime no LCD
    LCDWriteStringXY(9,1,"");
    teste = teclado();                                //Retorna
valor lido no teclado para variavel teste

    while(!teste)                                     //Enquanto
teste diferente de 0 le valor do teclado
    {
        tecla_numero = teclado();                     //retorna valor
lido no teclado para variavel tecla-numero

        if(tecla_numero == 10)                        //Se
tecla_numero = 10 (#) incrementa variavel relógio_click que funciona como cursor
        {
            relógio_click++;
        }

        if(tecla_numero == 12)                        //Atribui
para as respectivas variaveis o valor lido
        {
            horas = 0;
            minutos = 0;
            segundos = 0;
            digito_hora = 0;
            digito_minuto = 0;
            digito_segundo = 0;
        }
        if(tecla_numero == 1)
        {
            horas = 1;
            minutos = 1;
            segundos = 1;
            digito_hora = 1;
            digito_minuto = 1;

```

```
        digito_segundo = 1;
    }

    if(tecla_numero == 2)
    {
        horas = 2;
        minutos = 2;
        segundos = 2;
        digito_hora = 2;
        digito_minuto = 2;
        digito_segundo = 2;
    }
    if(tecla_numero == 3)
    {
        horas = 3;
        minutos = 3;
        segundos = 3;
        digito_hora = 3;
        digito_minuto = 3;
        digito_segundo = 3;
    }
    if(tecla_numero == 4)
    {
        horas = 4;
        minutos = 4;
        segundos = 4;
        digito_hora = 4;
        digito_minuto = 4;
        digito_segundo = 4;
    }
    if(tecla_numero == 5)
    {
        horas = 5;
        minutos = 5;
        segundos = 5;
        digito_hora = 5;
        digito_minuto = 5;
        digito_segundo = 5;
    }
    if(tecla_numero == 6)
    {
        horas = 6;
        minutos = 6;
        segundos = 6;
        digito_hora = 6;
        digito_minuto = 6;
        digito_segundo = 6;
    }
    if(tecla_numero == 7)
    {
        horas = 7;
        minutos = 7;
        segundos = 7;
        digito_hora = 7;
        digito_minuto = 7;
        digito_segundo = 7;
    }
}
```

```

if(tecla_numero == 8)
{
    horas = 8;
    minutos = 8;
    segundos = 8;
    digito_hora = 8;
    digito_minuto = 8;
    digito_segundo = 8;
}
if(tecla_numero == 9)
{
    horas = 9;
    minutos = 9;
    segundos = 9;
    digito_hora = 9;
    digito_minuto = 9;
    digito_segundo = 9;
}

if(tecla_numero!=0 && tecla_numero!=10 && relógio_click == 0)
//Testa se foi pressionada uma tecla, essa tecla é diferente do # e se
cursor esta na 1ª posição
{
    hora = convert_bcd_to_decimal(horas & 0x3f);
    //é atribuido para a variavel o valor já convertido de binario
para decimal do lido pelo teclado
    LCDWriteIntXY(0,1,hora,1);
    //Imprime no LCD a variavel
    hora = convert_decimal_to_bcd(hora);
}
if(tecla_numero!=0 && tecla_numero!=10 && relógio_click == 1)
//Testa se foi pressionada uma tecla, essa tecla é diferente do # e se
cursor esta na 2ª posição
{
    digito_hora = convert_bcd_to_decimal(digito_hora & 0x3f);
    //é atribuido para a variavel o valor já convertido de binario para
decimal do lido pelo teclado
    LCDWriteIntXY(1,1,digito_hora,1);
    //Imprime no LCD a variavel
    digito_hora = convert_decimal_to_bcd(digito_hora);

    variavel_hora = hora*10 + digito_hora;
    //Multiplica por 10 o primeiro valor da hora e soma ao
segundo
    ds1307_hour_write(variavel_hora,0,0);
    //Escreve no RTC o valor da hora
}

if(tecla_numero!=0 && tecla_numero!=10 && relógio_click == 2)
//Testa se foi pressionada uma tecla, essa tecla é diferente do # e se
cursor esta na 3ª posição
{
    minuto = convert_bcd_to_decimal(minutos);
    LCDWriteIntXY(3,1,minuto,1);
    minuto = convert_decimal_to_bcd(minuto);
}

```

```

        if(tecla_numero!=0 && tecla_numero!=10 && relógio_click == 3)
            //Testa se foi pressionada uma tecla, essa tecla é diferente do # e se
            cursor esta na 4ª posição
            {
                digito_minuto = convert_bcd_to_decimal(digito_minuto);
                //é atribuido para a variavel o valor já convertido de binario para
                decimal do lido pelo teclado
                LCDWriteIntXY(4,1,digito_minuto,1);
                //Imprime no LCD a variavel
                digito_minuto = convert_decimal_to_bcd(digito_minuto);

                variavel_minuto = minuto*10 + digito_minuto;
                //Multiplica por 10 o primeiro valor do minuto e soma ao segundo
                ds1307_minute_write(variavel_minuto);
                //Escreve no RTC o valor da minuto

            }

        if(tecla_numero!=0 && tecla_numero!=10 && relógio_click == 4)
            //Testa se foi pressionada uma tecla, essa tecla é diferente do # e se
            cursor esta na 5ª posição
            {
                segundo = convert_bcd_to_decimal(segundos);
                //é atribuido para a variavel o valor já convertido de
                binario para decimal do lido pelo teclado
                LCDWriteIntXY(6,1,segundo,1);
                //Imprime no LCD a variavel
                segundo = convert_decimal_to_bcd(segundo);

            }

        if(tecla_numero!=0 && tecla_numero!=10 && relógio_click == 5)
            //Testa se foi pressionada uma tecla, essa tecla é diferente do # e se
            cursor esta na 6ª posição
            {
                digito_segundo = convert_bcd_to_decimal(digito_segundo);
                //é atribuido para a variavel o valor já convertido de binario para
                decimal do lido pelo teclado
                LCDWriteIntXY(7,1,digito_segundo,1);
                //Imprime no LCD a variavel
                LCDWriteStringXY(8,1," ");
                digito_segundo = convert_decimal_to_bcd(digito_segundo);
                variavel_segundo = segundo*10 + digito_segundo;
                //Multiplica por 10 o primeiro valor do segundos e soma ao
                segundo
                ds1307_second_write(variavel_segundo);
                //Escreve no RTC o valor da segundo

            }

        if(tecla_numero!=10 && relógio_click == 6)
            //Testa se foi pressionada uma tecla e se cursor esta na
            7ª posição
            {
                if(variavel_hora>23 || variavel_minuto>59 ||
                variavel_segundo>59) //Testa se os valores inseridos estao dentro dos limites
                {

```

```

        variavel_hora = 0;
        //Casos valores estejam fora dos limites
inicializa o relógio tudo a 0 e pede para inserir novos valores
        variavel_minuto = 0;
        variavel_segundo = 0;
        LCDClear();
        LCDWriteStringXY(0,0,"Insira");
        LCDWriteStringXY(0,1,"Novamente");
        _delay_ms(1000);
        ds1307_hour_write(variavel_hora,0,0);
        ds1307_minute_write(variavel_minuto);
        ds1307_second_write(variavel_segundo);

    }

    relógio_click = 0;
    //coloca cursor na posição 0

    return;
    //Sai da função

}

}

}

void acerta_dia(void)
    //Função acerta dia
{
    int teste = 0;
    LCDClear();
    LCDWriteStringXY(0,0,"Acerte o dia: ");
    //Imprime mensagem no LCD

    teste = teclado();
    //Atribui na variável teste o valor lido no
    teclado
    int tecla_numero = 0;

    while(!teste)
    //Enquanto teste diferente de 0 le valor do
    teclado
    {
        tecla_numero = teclado();
        //Atribui na variável teste o valor lido no
    teclado

        if(tecla_numero == 10)
            //Atribui ao tecla_numero o valor lido do
    teclado
        {
            return;
        }
        if(tecla_numero == 1)
        {
            dias = 1;
        }
    }
}

```

```

if(tecla_numero == 2)
{
    dias = 2;
}
if(tecla_numero == 3)
{
    dias = 3;
}
if(tecla_numero == 4)
{
    dias = 4;
}
if(tecla_numero == 5)
{
    dias = 5;
}
if(tecla_numero == 6)
{
    dias = 6;
}
if(tecla_numero == 7)
{
    dias = 7;
}

ds1307_day_write(dias);
//Escreve no RTC o dia
dia = ds1307_read_day();
//Le o dia do RTC

switch(dia)
//Testa variavel dia, caso seja 1
imprime segunda e assim sucessivamente ate domingo
{
    case 1:

        LCDWriteStringXY(0,1,"Segunda ");
        break;

    case 2:

        LCDWriteStringXY(0,1,"Terca ");
        break;

    case 3:

        LCDWriteStringXY(0,1,"Quarta ");
        break;

    case 4:

        LCDWriteStringXY(0,1,"Quinta ");
        break;

    case 5:

        LCDWriteStringXY(0,1,"Sexta ");
        break;

```

```

        case 6:

            LCDWriteStringXY(0,1,"Sabado ");
            break;

        case 7:

            LCDWriteStringXY(0,1,"Domingo ");
            break;

    }

}

}

void rega_manual(void) // bit 0 -
portao, bit - 1 luz, bit - 2 persianas, bit - 3 alarme
{
    if(num_click_rega == 1) //testa o
click, caso seja o primeiro liga o led da rega e imprime a mensagem a indicar a
função em funcionamento
    {
        LCDClear();
        LCDWriteStringXY(9,1,"Rega_M");
        PORTC |= (1 << PORTC3);
        flag_rega = 1; //atribui
o valor 1 à flag da rega, significando que está a atuar esta função
    }
    else if(num_click_rega == 2) // volta a testar o
click, caso este seja o segundo click desliga o led da rega e imprime uma mensagem
de saída
    {
        LCDWriteStringXY(9,1,"Sair ");
        PORTC &= ~(1<<PORTC3);
        _delay_ms(500);
        _delay_ms(500);
        LCDWriteStringXY(9,1,"");
        LCDClear();
        flag_rega = 0; //atribui o
valor 0 à flag da rega para simbolizar que a rega manual já não está a actuar
    }
}

void seguranca_manual(void) // o
comportamento desta função é idêntico a função rega manual, com a diferença de que
este modo de funcionamento utiliza 3 leds
{
    // bit 0 - portao, bit 1 - persianas, bit 2 - alarme, bit 3 - rega
    if(num_click_seguranca == 1)
    {
        LCDWriteStringXY(9,1,"Segurança_M ");
        PORTC |= (1<<PORTC0)|(1<<PORTC1)|(1<<PORTC2);
        flag_seguranca = 1;

    }
    else if(num_click_seguranca == 2)

```



```

    {
        LCDWriteStringXY(9,1,"Sair");
        PORTC &= ~(1<<PORTC0)& ~(1<<PORTC1)& ~(1<<PORTC2);
        _delay_ms(500);
        _delay_ms(500);
        LCDWriteStringXY(9,1,"");
        flag_seguranca = 0;
    }
}

void seguranca_automatica(void) //função segurança
automatica, a segurança é ligada e desligada (representada por 3 leds) conforme o
utilizador define a hora de activa e desactivar a segurança
{
    if((hora*100 + minuto) >= (hora_inicio_seguro*100 + minuto_inicio_seguro) &&
(hora*100 + minuto) < (hora_fim_seguro*100 + minuto_fim_seguro)) //Caso a hora e os
minutos em que se encontra o relógio estejam entre a hora de inicio e de fim da
segurança definidos na função "acerta_automatica", então o modo de segurança vai ser
activado e o led vai acender. Caso contrário a segurança não actua ou é desligada
caso estivesse ligada.
    {
        PORTC |= (1<<PORTC0)|(1<<PORTC1)|(1<<PORTC2);
        LCDWriteStringXY(9,1,"Segurança_A");
    }
    else
    {
        LCDWriteStringXY(9,1,"");
        PORTC &= ~(1<<PORTC0)& ~(1<<PORTC1)& ~(1<<PORTC2);
    }
}

void rega_automatica(void) //
esta função é idêntica à segurança_automatica, com a diferença de que o modo de
funcionamento da rega apenas utiliza um led para a representar, apenas funciona a
segunda, quarta e sexta feira
{
    if((hora*100 + minuto) >= (hora_inicio_rega*100 + minuto_inicio_rega) &&
(hora*100 + minuto) < (hora_fim_rega*100 + minuto_fim_rega) && (dia == 1 || dia == 3
|| dia == 5) && flag_rega == 0 && flag_seguranca == 0)
    {
        PORTC |= (1<<PORTC3);
        LCDWriteStringXY(9,1,"Rega_A");
        _delay_ms(500);
    }
    else
    {
        LCDWriteStringXY(9,1,"");
        PORTC &= ~(1<<PORTC3);
    }
}
}

```

```

void acerta_automatico(void)
    //Função de definir inicio e fim das funções automáticas
{
    int teste = 0, tecla_numero = 0;
    //Esta função é idêntica a função acerta hora, com a particularidade de que
    os valores lidos do teclado não são escritos no RTC, mas são guardados em variáveis
    que vão ser usadas para comparar com os valores de hora atual nas funções segurança
    e rega automática
    LCDWriteStringXY(0,0,"Inicio da rega: ");
    LCDWriteStringXY(0,1,"00:00");

    teste = teclado();

    while(!teste)
    {
        tecla_numero = teclado();

        if(tecla_numero == 10)
        {
            relógio_click++;
        }

        if(tecla_numero == 12)
        {
            horas = 0;
            minutos = 0;
            dígito_hora = 0;
            dígito_minuto = 0;
        }

        if(tecla_numero == 1)
        {
            horas = 1;
            minutos = 1;
            dígito_hora = 1;
            dígito_minuto = 1;
        }

        if(tecla_numero == 2)
        {
            horas = 2;
            minutos = 2;
            dígito_hora = 2;
            dígito_minuto = 2;
        }

        if(tecla_numero == 3)
        {
            horas = 3;
            minutos = 3;
            dígito_hora = 3;
            dígito_minuto = 3;
        }

        if(tecla_numero == 4)
        {
            horas = 4;

```

```

        minutos = 4;
        digito_hora = 4;
        digito_minuto = 4;
    }
    if(tecla_numero == 5)
    {
        horas = 5;
        minutos = 5;
        digito_hora = 5;
        digito_minuto = 5;
    }
    if(tecla_numero == 6)
    {
        horas = 6;
        minutos = 6;
        digito_hora = 6;
        digito_minuto = 6;
    }
    if(tecla_numero == 7)
    {
        horas = 7;
        minutos = 7;
        digito_hora = 7;
        digito_minuto = 7;
    }
    if(tecla_numero == 8)
    {
        horas = 8;
        minutos = 8;
        digito_hora = 8;
        digito_minuto = 8;
    }
    if(tecla_numero == 9)
    {
        horas = 9;
        minutos = 9;
        digito_hora = 9;
        digito_minuto = 9;
    }
}

    if(tecla_numero!=0 && tecla_numero!=10 &&relogio_click == 0)           //0
princípio de funcionamento e igual a acerta hora como já tinha sido referido, mas de
realçar que tem de ser preenchidas mais posições do cursor (17) para início e fim
das duas funções
    {
        rega1_hora_inicio = convert_bcd_to_decimal(horas);
        LCDWriteIntXY(0,1,rega1_hora_inicio,1);
    }

    if(tecla_numero!=0 && tecla_numero!=10 && relogio_click == 1)
    {
        rega2_hora_inicio = convert_bcd_to_decimal(digito_hora);
        LCDWriteIntXY(1,1,rega2_hora_inicio,1);
    }

    if(tecla_numero!=0 && tecla_numero!=10 && relogio_click == 2)
    {

```

```

        rega1_minuto_inicio = convert_bcd_to_decimal(minutos);
        LCDWriteIntXY(3,1,rega1_minuto_inicio,1);
    }

    if(tecla_numero!=0 && teclado_numero!=10 && relógio_click == 3)
    {

        rega2_minuto_inicio = convert_bcd_to_decimal(digito_minuto);
        LCDWriteIntXY(4,1,rega2_minuto_inicio,1);
        LCDWriteStringXY(5,1,"    ");

    }

    if(relógio_click == 4)
    {

        LCDWriteStringXY(0,0,"Fim da rega:    ");
        LCDWriteStringXY(0,1,"00:00    ");

        rega1_hora_fim = convert_bcd_to_decimal(digito_minuto);
        LCDWriteIntXY(0,1,rega1_hora_fim,1);

    }
    if(tecla_numero!=0 && teclado_numero!=10 && relógio_click == 5)
    {

        rega2_hora_fim = convert_bcd_to_decimal(digito_minuto);
        LCDWriteIntXY(1,1,rega2_hora_fim,1);

    }
    if(tecla_numero!=0 && teclado_numero!=10 && relógio_click == 6)
    {

        rega1_minuto_fim = convert_bcd_to_decimal(digito_minuto);
        LCDWriteIntXY(3,1,rega1_minuto_fim,1);

    }
    if(tecla_numero!=0 && teclado_numero!=10 && relógio_click == 7)
    {

        rega2_minuto_fim = convert_bcd_to_decimal(digito_minuto);
        LCDWriteIntXY(4,1,rega2_minuto_fim,1);
        LCDWriteStringXY(5,1,"    ");

    }

    if(relógio_click == 8)
    {

        hora_inicio_rega = rega1_hora_inicio*10 + rega2_hora_inicio;
        hora_fim_rega = rega1_hora_fim*10 + rega2_hora_fim;

        minuto_inicio_rega = rega1_minuto_inicio*10 +
rega2_minuto_inicio;
        minuto_fim_rega = rega1_minuto_fim*10 + rega2_minuto_fim;

    }

```

```

if(relogio_click == 9)
{
    LCDClear();
    LCDWriteStringXY(0,0,"Inicio seguran ca");
    LCDWriteStringXY(0,1,"00:00");

    seguro1_hora_inicio = convert_bcd_to_decimal(horas);
    LCDWriteIntXY(0,1,seguro1_hora_inicio,1);
}

if(tecla_numero!=0 && tecla_numero!=10 && relogio_click == 10)
{
    seguro2_hora_inicio = convert_bcd_to_decimal(digito_hora);
    LCDWriteIntXY(1,1,seguro2_hora_inicio,1);
}

if(tecla_numero!=0 && tecla_numero!=10 && relogio_click == 11)
{
    seguro1_minuto_inicio = convert_bcd_to_decimal(minutos);
    LCDWriteIntXY(3,1,seguro1_minuto_inicio,1);
}

if(tecla_numero!=0 && tecla_numero!=10 && relogio_click == 12)
{
    seguro2_minuto_inicio = convert_bcd_to_decimal(digito_minuto);
    LCDWriteIntXY(4,1,seguro2_minuto_inicio,1);
    LCDWriteStringXY(5,1,"");
}

if(relogio_click == 13)
{
    LCDWriteStringXY(0,0,"Fim seguran a: ");
    LCDWriteStringXY(0,1,"00:00");

    seguro1_hora_fim = convert_bcd_to_decimal(digito_minuto);
    LCDWriteIntXY(0,1,seguro1_hora_fim,1);
}

if(tecla_numero!=0 && tecla_numero!=10 && relogio_click == 14)
{
    seguro2_hora_fim = convert_bcd_to_decimal(digito_minuto);
    LCDWriteIntXY(1,1,seguro2_hora_fim,1);
}

if(tecla_numero!=0 && tecla_numero!=10 && relogio_click == 15)
{
    seguro1_minuto_fim = convert_bcd_to_decimal(digito_minuto);
    LCDWriteIntXY(3,1,seguro1_minuto_fim,1);
}

if(tecla_numero!=0 && tecla_numero!=10 && relogio_click == 16)

```

```

    {

        seguro2_minuto_fim = convert_bcd_to_decimal(digito_minuto);
        LCDWriteIntXY(4,1,seguro2_minuto_fim,1);
        LCDWriteStringXY(5,1,"    ");

    }

    if(relogio_click == 17)
    {

        hora_inicio_seguro = seguro1_hora_inicio*10 +
seguro2_hora_inicio;
        hora_fim_seguro = seguro1_hora_fim*10 + seguro2_hora_fim;

        minuto_inicio_seguro = seguro1_minuto_inicio*10 +
seguro2_minuto_inicio;
        minuto_fim_seguro = seguro1_minuto_fim*10 + seguro2_minuto_fim;

        relogio_click = 0;

        return;

    }

}

}

int teclado(void)                                //Função teclado
{
    int tecla_numero = 0;                        //Inicializa variavel a 0

    //Primeira coluna

    PORTD = 0b11101111;                          //Selecciona a coluna
    _delay_ms(50);
    if (!(PIND & 0b00000001))                    //Vai testar as quatro linha
possiveis, quando encontra a linha e coluna precionada guarda o valor da tecla numa
variavel
    {
        //LCDWriteStringXY(1,1,"1");            // Este processo é feito para
todas as colunas
        tecla_numero = 1;

    }
    if (!(PIND & 0b00000010))
    {
        //LCDWriteStringXY(1,1,"4");
        tecla_numero = 4;
    }
    if (!(PIND & 0b00000100))
    {
        //LCDWriteStringXY(1,1,"7");
        tecla_numero = 7;
    }
}

```

```

if (!(PIND & 0b00001000))
{
    //LCDWriteStringXY(1,1,"*");
    tecla_numero = 10;
}

//Segunda coluna

PORTD = 0b11011111;
_delay_ms(50);
if (!(PIND & 0b00000001))
{
    //LCDWriteStringXY(1,1,"2");
    tecla_numero = 2;
}
if (!(PIND & 0b00000010))
{
    //LCDWriteStringXY(1,1,"5");
    tecla_numero = 5;
}
if (!(PIND & 0b00000100))
{
    //LCDWriteStringXY(1,1,"8");
    tecla_numero = 8;
}
if (!(PIND & 0b00001000))
{
    //LCDWriteStringXY(1,1,"0");
    tecla_numero = 12;
}

//Terceira coluna

PORTD = 0b11011111;
_delay_ms(50);
if (!(PIND & 0b00000001))
{
    //LCDWriteStringXY(1,1,"3");
    tecla_numero = 3;
}
if (!(PIND & 0b00000010))
{
    //LCDWriteStringXY(1,1,"6");
    tecla_numero = 6;
}
if (!(PIND & 0b00000100))
{
    //LCDWriteStringXY(1,1,"9");
    tecla_numero = 9;
}
if (!(PIND & 0b00001000))
{
    //LCDWriteStringXY(1,1,"#");
    Se tecla # for pressionada chama função acerta automático //
    acerta_automatico();
}

//Quarta coluna

```

```

    PORTD = 0b01111111;
    _delay_ms(50);
    if (!(PIND & 0b00000001))
        //Se a tecla A for pressionada liga ou desliga mediante o estado do click a
        rega manual
        {
            rega_manual();
            if(num_click_rega == 1)
            {
                num_click_rega = 2;
            }
            else
            {
                num_click_rega = 1;
            }
            //LCDWriteStringXY(1,1,"A");
        }
    if (!(PIND & 0b00000010))
    {
        seguranca_manual();
        //Se a tecla B for pressionada liga ou desliga mediante o estado do click a
        segurança manual
        {
            if(num_click_seguranca == 1)
            {
                num_click_seguranca = 2;
            }
            else
            {
                num_click_seguranca = 1;
            }
            //LCDWriteStringXY(1,1,"B");
        }
    if (!(PIND & 0b00000100))
    {
        acerta_dia();
        // Se tecla C for pressionada chama função acerta dia
    }
    if (!(PIND & 0b00001000))
    {
        acerta_hora();
        // Se tecla D for pressionada chama função acerta
        hora
    }
    return (tecla_numero);
}

int main(void)
    //Função MAIN
{
    inicio_teclado();
    //Chama função de inicialização do teclado
    inic_timer();
    //Chama função de inicialização do timer
    twi_init();
    //Inicializa comunicação two wire interface (I2C)

```



```

LCDInit(LS_NONE);
//Inicializa o LCD

LCDClear();
//Limpa o LCD

LCDWriteStringXY(3,0,"Programador");
//O LCD liga com a mensagem Programador Semanal
LCDWriteStringXY(5,1,"Semanal");
_delay_ms(1500);
LCDClear();

LCDWriteStringXY(0,0,"A - Rega_M");
//Menu de informações
LCDWriteStringXY(0,1,"B - Seguranca_M");
_delay_ms(1500);
LCDClear();

LCDWriteStringXY(0,0,"C - Acerta dia");
LCDWriteStringXY(0,1,"D - Acerta hora");
_delay_ms(1500);
LCDClear();

LCDWriteStringXY(0,0,"# - Acerta Auto");
_delay_ms(1500);
LCDClear();

ds1307_second_write(0);
// Escrever o valor dos segundos no DS1307
ds1307_minute_write(0);
// Escrever o valor dos minutos no DS1307
ds1307_hour_write(0,0,0);
// Escrever o valor das horas no DS1307 no formato 24h
ds1307_day_write(1);
// Escrever o dia no DS1307
ds1307_date_write(1);
// Escreve a data no DS1307
ds1307_month_write(1);
// Escreve o mes no DS1307
ds1307_year_write(15);
// Escreve o ano no DS1307

while(1)
{
    teclado();

    if(flag_int == 1)
    {
        // Condição que testa o valor da flag_int

        // Caso esta seja verdadeira ocorreu o atraso de 500ms
        flag_int = 0;
    }
}
// Reset flag_int

```

```

        PORTB ^= (1 << PORTB0);
    // Liga ou desliga led consoante o seu estado anterior
    }

    segundo = ds1307_read_second();
    // Le o valor dos segundos no DS1307 e guarda na variavel
    minuto = ds1307_read_minute();
    // Le o valor dos minutos no DS1307 e guarda na variavel

    hora = ds1307_read_hour();
    // Le o valor dos hora no DS1307 e guarda na variavel
    dia = ds1307_read_day();
    // Le o valor dos dias no DS1307 e guarda na variavel
    data = ds1307_read_date();
    // Le o valor dos data no DS1307 e guarda na variavel
    mes = ds1307_read_month();
    // Le o valor dos mes no DS1307 e guarda na variavel
    ano = ds1307_read_year();
    // Le o valor dos ano no DS1307 e guarda na variavel

    hora = convert_bcd_to_decimal((hora & 0x3f));           //Converte
o valor da variavel, que guarda o a hora lida do rtc, de binario para decimal e
imprime no lcd

                                // O mesmo vai suceder-se para os minutos, segundos, dia,
data, hora, ano
    LCDWriteIntXY(0,1,hora,2);
    LCDWriteStringXY(2,1,":");

    minuto = convert_bcd_to_decimal(minuto);

    LCDWriteIntXY(3,1,minuto,2);
    LCDWriteStringXY(5,1,":");

    segundo = convert_bcd_to_decimal(segundo);

    LCDWriteIntXY(6,1,segundo,2);
    LCDWriteStringXY(8,1," ");

    data = convert_bcd_to_decimal(data);
    LCDWriteIntXY(0,0,data,2);
    LCDWriteStringXY(2,0,":");

    mes = convert_bcd_to_decimal(mes);
    LCDWriteIntXY(3,0,mes,2);
    LCDWriteStringXY(5,0,":");

    ano = convert_bcd_to_decimal(ano);
    LCDWriteIntXY(6,0,ano,2);
    LCDWriteStringXY(8,0," ");

    switch(dia)
    {
        case 1:

            LCDWriteStringXY(9,0,"Segunda");

```

```

        break;

    case 2:

        LCDWriteStringXY(9,0,"Terca");
        break;

    case 3:

        LCDWriteStringXY(9,0,"Quarta");
        break;

    case 4:

        LCDWriteStringXY(9,0,"Quinta");
        break;

    case 5:

        LCDWriteStringXY(9,0,"Sexta");
        break;

    case 6:

        LCDWriteStringXY(9,0,"Sabado");
        break;

    case 7:

        LCDWriteStringXY(9,0,"Domingo");
        break;

    }

    if(flag_rega == 0 && flag_seguranca == 0)
        //Garante que so entra nas funcoes automáticas se as funcoes manuais nao
        estiverem ativas
        {
            rega_automatica();
            //Chama funcao rega automática
            seguranca_automatica();
            //Chama funcao segurança automática
        }

    }

}

```