```
#define F_CPU 16000000UL
#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>
#include <time.h>
#include <rtc3231.h>
#include <eepromEasy.h>

int disinit = 1;
int changemin = 0;
int menu = 0;
int submenu = 0;

int EditHour = 0;
int EditMin = 0;
int EditSec = 0;
int editok = 0;
int ringok = 0;

int EditSet = 0;

int Delset = 0;

int AlarmMin = 0;
int AlarmHour = 0;

int numOfEntries;
int MemMin[30];
int MemHour[30];

#define control_bus PORTB
#define controlbus_direction DDRB
#define data_bus PORTB
#define databus_direction DDRB
#define Bellbus_direction DDRD
#define Bellbus_port PORTD

#define rs 0
#define en 1
#define d4 2
#define d5 3
#define d6 4
#define d7 5
#define Ringpin 1

struct rtc_time RTCH;
struct rtc_date RTCD;
struct tm times;
struct tm Deltimes;

char Timedis[16];
char Datedis[16];
char Timeeditdis[16];
char Dateeditdis[16];
char Timedeldis[16];


void LCD_CmdWrite( char a)
{
    if(a & 0x80) data_bus|=(1<<d7); else data_bus&= ~(1<<d7);
    if(a & 0x40) data_bus|=(1<<d6); else data_bus&= ~(1<<d6);
    if(a & 0x20) data_bus|=(1<<d5); else data_bus&= ~(1<<d5);
    if(a & 0x10) data_bus|=(1<<d4); else data_bus&= ~(1<<d4);
    control_bus &=~(1<<rs);control_bus |=(1<<en);
    _delay_ms(2);
    control_bus &=~(1<<en);

    _delay_ms(2);

    if(a & 0x08) data_bus|=(1<<d7); else data_bus&= ~(1<<d7);
    if(a & 0x04) data_bus|=(1<<d6); else data_bus&= ~(1<<d6);
    if(a & 0x02) data_bus|=(1<<d5); else data_bus&= ~(1<<d5);
    if(a & 0x01) data_bus|=(1<<d4); else data_bus&= ~(1<<d4);
    control_bus &=~(1<<rs);control_bus |=(1<<en);
    _delay_ms(2);
    control_bus &=~(1<<en);

    _delay_ms(2);
}
void LCD_DataWrite( char a)
{

    if(a & 0x80) data_bus|=(1<<d7); else data_bus&= ~(1<<d7);
    if(a & 0x40) data_bus|=(1<<d6); else data_bus&= ~(1<<d6);
    if(a & 0x20) data_bus|=(1<<d5); else data_bus&= ~(1<<d5);
    if(a & 0x10) data_bus|=(1<<d4); else data_bus&= ~(1<<d4);
    control_bus |=(1<<rs)|(1<<en);
    _delay_ms(2);
    control_bus &=~(1<<en);
    _delay_ms(2);

    if(a & 0x08) data_bus|=(1<<d7); else data_bus&= ~(1<<d7);
    if(a & 0x04) data_bus|=(1<<d6); else data_bus&= ~(1<<d6);
    if(a & 0x02) data_bus|=(1<<d5); else data_bus&= ~(1<<d5);
    if(a & 0x01) data_bus|=(1<<d4); else data_bus&= ~(1<<d4);
    control_bus |=(1<<rs)|(1<<en);
    _delay_ms(2);
    control_bus &=~(1<<en);
    _delay_ms(2);

}
void LCD_Init()
{
    controlbus_direction |= ((1<<rs)|(1<<en));
    databus_direction |= ((1<<d7)|(1<<d6)|(1<<d5)|(1<<d4));
    _delay_ms(2);
    LCD_CmdWrite(0x01); // clear display
    LCD_CmdWrite(0x02); // back to home
    LCD_CmdWrite(0x28); // 4bit,2line,5x8 pixel
```

```c
    LCD_CmdWrite(0x06); // entry mode,cursor increments by cursor shift
    LCD_CmdWrite(0x0c); // display ON,cursor OFF
    LCD_CmdWrite(0x80); // force cursor to begin at line1

}
void LCD_Init2()
{
    controlbus_direction |= ((1<<rs)|(1<<en));
    databus_direction |= ((1<<d7)|(1<<d6)|(1<<d5)|(1<<d4));
    _delay_ms(2);
    LCD_CmdWrite(0x01); // clear display
    LCD_CmdWrite(0x02); // back to home
    LCD_CmdWrite(0x20); // 4bit,1line,5x8 pixel
    LCD_CmdWrite(0x06); // entry mode,cursor increments by cursor shift
    LCD_CmdWrite(0x0c); // display ON,cursor OFF
    LCD_CmdWrite(0x80); // force cursor to begin at line1

}
void LCD_Disp(const char *p)
{
    while(*p!='\0')
    {
        LCD_DataWrite(*p);
        p++; _delay_us(800);
    }
}
void LCD_setCursor(int a, int b)
{
    int i=0;
    switch(b){
        case 0:LCD_CmdWrite(0x80);break;
        case 1:LCD_CmdWrite(0xC0);break;
    }

    for(i=0;i<a;i++)
    LCD_CmdWrite(0x14);
}

void Menu_setclock()
{

    LCD_Init2();
    LCD_setCursor(0,0);
    LCD_Disp("   SET  CLOCK   ");
    //LCD_setCursor(79,0);
    //LCD_Disp(" ");

}
void Menu_deletealarm()
{

    LCD_Init2();
    LCD_setCursor(0,0);
    LCD_Disp("  DELETE ALARM  ");
    //LCD_setCursor(79,0);
    //LCD_Disp(" ");
```

```c
}
void Menu_setalarm()
{

    LCD_Init2();
    LCD_setCursor(0,0);
    LCD_Disp("   SET  ALARM   ");
    //LCD_setCursor(79,0);
    //LCD_Disp(" ");

}

void Edit_hour()
{
    if (menu == 1)
    {
        times.tm_hour = (int)RTCH.hour;
    }
    else if (menu == 2)
    {
        times.tm_hour = AlarmHour;
    }

    LCD_Init();
    LCD_setCursor(0,1);
    strftime(Timeeditdis,16,"       %H        ", &times);
    LCD_Disp(Timeeditdis);
    LCD_setCursor(0,0);
    LCD_Disp("   (EDIT)HOUR   ");
}
void Edit_minute()
{
    if (menu == 1)
    {
        times.tm_min = (int)RTCH.min;
    }
    else if (menu == 2)
    {
        times.tm_min = AlarmMin;
    }

    LCD_Init();
    LCD_setCursor(0,1);
    strftime(Timeeditdis,16,"       %M        ", &times);
    LCD_Disp(Timeeditdis);
    LCD_setCursor(0,0);
    LCD_Disp("   (EDIT)Min    ");
}
void Edit_sec()
{
    if (menu == 1)
    {
        times.tm_sec = (int)RTCH.sec;
    }
    LCD_Init();
    LCD_setCursor(0,1);
    strftime(Timeeditdis,16,"       %S        ", &times);
    LCD_Disp(Timeeditdis);
    LCD_setCursor(0,0);
    LCD_Disp("   (EDIT)SEC    ");
}

void Edit_clock()
{
    EditHour = 0;
    EditMin = 0;
    EditSec = 0;
    LCD_Init();
    LCD_setCursor(0,1);
    rtc3231_read_time(&RTCH);
    Afterwrite();
    times.tm_sec = (int)RTCH.sec;
    times.tm_min = (int)RTCH.min;
    times.tm_hour = (int)RTCH.hour;

    switch (submenu)
    {
        case 1:
        strftime(Timeeditdis,16,"       %H        ", &times);
        LCD_Disp(Timeeditdis);
        LCD_setCursor(0,0);
        LCD_Disp("      HOUR      ");
        submenu = 1;
        EditSec = 0;
        EditMin = 0;
        EditHour = 1;
        break;

        case 2:
        strftime(Timeeditdis,16,"       %M        ", &times);
        LCD_Disp(Timeeditdis);
        LCD_setCursor(0,0);
        LCD_Disp("      Min       ");
        submenu = 2;
        EditMin = 1;
        EditHour = 0;
        EditSec = 0;
        break;

        case 3:
        strftime(Timeeditdis,16,"       %S        ", &times);
        LCD_Disp(Timeeditdis);
        LCD_setCursor(0,0);
        LCD_Disp("      SEC       ");
        submenu = 3;
        EditSec = 1;
        EditMin = 0;
        EditHour = 0;
        break;
```

```c
    }
}
void AlarmEdit_clock()
{
    EditHour = 0;
    EditMin = 0;


    LCD_Init();
    LCD_setCursor(0,1);


    times.tm_min = AlarmMin;
    times.tm_hour = AlarmHour;

    switch (submenu)
    {
        case 1:
        strftime(Timeeditdis,16,"       %H         ", &times);
        LCD_Disp(Timeeditdis);
        LCD_setCursor(0,0);
        LCD_Disp("      HOUR      ");
        submenu = 1;
        EditMin = 0;
        EditHour = 1;
        EditSet = 0;
        break;

        case 2:
        strftime(Timeeditdis,16,"       %M         ", &times);
        LCD_Disp(Timeeditdis);
        LCD_setCursor(0,0);
        LCD_Disp("      Min       ");
        submenu = 2;
        EditMin = 1;
        EditHour = 0;
        EditSet = 0;
        break;


        case 3:
        LCD_setCursor(0,0);
        LCD_Disp("      SET       ");
        submenu = 3;
        EditMin = 0;
        EditHour = 0;
        EditSet = 1;
        break;
    }
}

void SelectMenu()
{
    switch (menu)
    {
        case 0:
        disinit = 1;
        Home();

        break;


        case 1:
        _delay_ms(250);
        Menu_setclock();
        break;

        case 2: Menu_setalarm();
        break;

        case 3:
        _delay_ms(250);
        Menu_deletealarm();
        break;
    }
}

void Ring()
{
    ringok = 0;
    Bellbus_port |= (1 << Ringpin);
    _delay_ms(4000);
    Bellbus_port &= !(1 << Ringpin);

}

void Home()
{
    changemin = (int)RTCH.min;
    rtc3231_read_time(&RTCH);
    Afterwrite();
    rtc3231_read_date(&RTCD);
    Afterwrite();
    times.tm_sec = (int)RTCH.sec;
    times.tm_min = (int)RTCH.min;
    times.tm_hour = (int)RTCH.hour;
    times.tm_mday = (int)RTCD.day;
    times.tm_mon = (int)RTCD.month;
    times.tm_year = (int)RTCD.year;
    times.tm_wday = (int)RTCD.wday;
    times.tm_yday = 25;
    times.tm_isdst = 2;

    strftime(Datedis,16,"    %x    ", &times);
    strftime(Timedis,16,"    %I:%M %p    ", &times);

    if ((changemin != (int)RTCH.min)|(disinit == 1))
    {
        LCD_Init();
        LCD_setCursor(0,1);
        LCD_Disp(Timedis);
        LCD_setCursor(0,0);
        LCD_Disp(Datedis);
        changemin = (int)RTCH.min;
        disinit = 0;
        ringok = 1;
    }
    int q;
    numOfEntries = EEPROM_ReadByte(0x00);

    if  (ringok == 1)
    {
        for (q=0;q < numOfEntries; q ++)
        {

            if ((EEPROM_ReadByte((uint8_t *)(1+2*q)) == (int)RTCH.hour) &&
              (EEPROM_ReadByte((uint8_t *)(2 + 2*q)) == (int)RTCH.min))
            {
                Ring();
            }


        }
    }


    _delay_ms(1000);

}
void Afterwrite(void)
{
    i2c_start_condition();
    i2c_send_byte(RTC_WADDR);
    i2c_send_byte(0x00);
    i2c_stop_condition();
}

void DisplayDel()
{
    Delset = 0;
    char indexstr[3];
    numOfEntries = EEPROM_ReadByte(0x00);
    LCD_Init();
    LCD_setCursor(0,1);
    strftime(Timedeldis,16,"     %H:%M      ", &Deltimes);
    LCD_Disp(Timedeldis);
    LCD_setCursor(0,0);
    itoa(submenu,indexstr,10);
    LCD_Disp(indexstr);

}

void  DelSetting()
{
    LCD_Init();
    LCD_setCursor(0,0);
    LCD_Disp("     DELETE     ");
}

void Save()
{

    EEPROM_WriteByte((uint8_t *)(2*numOfEntries +1),(uint8_t *)AlarmHour);
    EEPROM_WriteByte((uint8_t *)(2*numOfEntries +2),(uint8_t *)AlarmMin);
    numOfEntries += 1;
    EEPROM_WriteByte(0x00,(uint8_t *)numOfEntries);
    int k;

    if (numOfEntries > 0)
    {
        for (k=0;k < numOfEntries; k ++)
        {
            MemHour[k] = EEPROM_ReadByte((uint8_t *)(1+2*k));
            MemMin[k] = EEPROM_ReadByte((uint8_t *)(2 + 2*k));
        }
    }

}

void Delete()
{
    EEPROM_WriteByte((uint8_t *)(2*numOfEntries -1),0xFF);
    EEPROM_WriteByte((uint8_t *)(2*numOfEntries),0xFF);
    numOfEntries -= 1;
    EEPROM_WriteByte(0x00,(uint8_t *)numOfEntries);
    int m;

    if (numOfEntries > 0)
    {
        for (m=0;m < numOfEntries; m ++)
        {
            MemHour[m] = EEPROM_ReadByte((uint8_t *)(1+2*m));
            MemMin[m] = EEPROM_ReadByte((uint8_t *)(2 + 2*m));
        }
    }
}

int main()
{
    //Ring();
    ringok = 1;
    Bellbus_direction |= (1 << Ringpin);
    PCICR = (1 << PCIE2);
    sei();
    PCMSK2 = (1 << PCINT18) | (1 << PCINT19)| (1 << PCINT20)| (1 << PCINT21);

    i2c_init();
    rtc3231_init();

    numOfEntries = EEPROM_ReadByte(0x00);
    if (numOfEntries == 0xFF)
    numOfEntries = 0;
    int p;

    if (numOfEntries > 0)
    {
        for (p=0;p < numOfEntries; p ++)
        {
            MemHour[p] = EEPROM_ReadByte((uint8_t *)(1+2*p));
            MemMin[p] = EEPROM_ReadByte((uint8_t *)(2 + 2*p));
        }
    }



    while(1)
    {
        if (menu == 0)
        Home();
    }

}

ISR(PCINT2_vect)
{

    if (PIND & 0b00000100) // <----
    {
    }
    else{
        if (submenu == 0)
        {
            menu -= 1;
            if (menu == -1)
            {
                menu = 3;
            }
            SelectMenu();

        }
        if (submenu != 0)
        {
            if (menu == 1)
            {
                if (editok == 0)
                {
                    submenu -= 1;
                    if (submenu == 0)
                    submenu = 3;
                    Edit_clock();
                }
                else if ( editok == 1)
                {
                    if (EditHour == 1)
                    {
                        RTCH.hour -= 1;
                        if (RTCH.hour == -1)
                        RTCH.hour = 23;
                        Edit_hour();
                    }

                    if (EditMin == 1)
                    {
                        RTCH.min -= 1;
                        if (RTCH.min == -1)
                        RTCH.min = 59;
                        Edit_minute();
                    }

                    if (EditSec == 1)
                    {
                        RTCH.sec -= 1;
                        if (RTCH.sec == -1)
                        RTCH.sec = 59;
                        Edit_sec();
                    }
                }
            }
            if (menu == 2)
            {
                if (editok == 0)
                {
                    submenu -= 1;
                    if (submenu == 0)
                    submenu = 3;
                    AlarmEdit_clock();
                }
                else if(editok ==1)
                {
                    if (EditHour == 1)
                    {
                        AlarmHour -= 1;
                        if (AlarmHour == -1)
                        AlarmHour = 23;
                        Edit_hour();
                    }

                    if (EditMin == 1)
                    {
                        AlarmMin -= 1;
                        if (AlarmMin == -1)
                        AlarmMin = 59;
                        Edit_minute();
                    }

                }
            }
            if (menu == 3)
            {
                submenu -= 1;
                if (submenu == 0)
                submenu = numOfEntries + 1;
                if (submenu < numOfEntries + 1)
                {
                    Deltimes.tm_hour = MemHour[submenu -1];
                    Deltimes.tm_min = MemMin[submenu - 1];
                    DisplayDel();
                }

                else if(submenu == numOfEntries + 1)
                {
                    Delset = 1;
                    DelSetting();
                }
            }

        }
    }

    if (PIND & 0b00001000) // OK
    {
    }
    else {

        if ((menu>0) && (submenu == 0))
        submenu = 1;

        if (menu == 1)
        {
            if  (submenu != 0)
            {
                if (editok == 0)
                {
                    if (EditHour == 1)
                    {
                        editok = 1;
                        Edit_hour();
                    }
                    else if(EditMin == 1)
                    {
                        editok = 1;
                        Edit_minute();
                    }
                    else if (EditSec == 1)
                    {
                        editok = 1;
                        Edit_sec();
                    }
                    else
                    {
                        Edit_clock();
                    }
                }
                else if(editok == 1)
                {
                    rtc3231_write_time(&RTCH);
                    editok = 0;
                    if (EditHour == 1)
                    {
                        EditHour = 0;
                        submenu = 1;
                        Edit_clock();
                    }
                    if (EditMin == 1)
                    {
                        EditMin = 0;
                        submenu = 2;
                        Edit_clock();
                    }
                    if (EditSec == 1)
                    {
                        EditSec = 0;
                        submenu = 3;
                        Edit_clock();
                    }
                }
            }
        }
        if (menu == 2)
        {
            if  (submenu != 0)
            {
                if (editok == 0)
                {
                    if (EditHour == 1)
                    {
                        editok = 1;
                        Edit_hour();
                    }
                    else if(EditMin == 1)
                    {
                        editok = 1;
                        Edit_minute();
                    }

                    else if (EditSet == 1)
                    {
                        Save();
                        submenu = 1;
                        EditSet = 0;
                        menu = 2;
                        AlarmMin = 0;
                        AlarmHour = 0;
                        AlarmEdit_clock();
                    }
                    else
                    {
                        AlarmEdit_clock();
                    }
                }
                else if(editok == 1)
                {

                    editok = 0;
                    if (EditHour == 1)
                    {
                        EditHour = 0;
                        submenu = 1;
                        AlarmEdit_clock();
                    }
                    if (EditMin == 1)
                    {
                        EditMin = 0;
                        submenu = 2;
                        AlarmEdit_clock();
                    }

                }
            }
        }

        if (menu == 3)
        {
            if (submenu != 0)
            {
                if (Delset == 1)
                {
                    Delete();
                    Delset = 0;
                    submenu = 0;
                    menu = 0;
                    SelectMenu();

                }
                else
                {
                    Deltimes.tm_hour = MemHour[submenu -1];
                    Deltimes.tm_min = MemMin[submenu - 1];
                    DisplayDel();
                }

            }
        }

    }

    if (PIND & 0b00010000) // ---->
    {
    }
    else {

        if (submenu != 0)
        {
            if (menu == 1)
            {
                if (editok == 0)
                {
                    submenu += 1;
                    if (submenu == 4)
                    submenu = 1;
                    Edit_clock();
                }
                else if(editok ==1)
                {
                    if (EditHour == 1)
                    {
                        RTCH.hour += 1;
                        if (RTCH.hour == 24)
                        RTCH.hour = 0;
                        Edit_hour();
                    }

                    if (EditMin == 1)
                    {
                        RTCH.min += 1;
                        if (RTCH.min == 60)
                        RTCH.min = 0;
                        Edit_minute();
                    }

                    if (EditSec == 1)
                    {
                        RTCH.sec += 1;
                        if (RTCH.sec == 60)
                        RTCH.sec = 0;
                        Edit_sec();
                    }
                }
            }
            if (menu == 2)
            {
                if (editok == 0)
                {
                    submenu += 1;
                    if (submenu == 4)
                    submenu = 1;
                    AlarmEdit_clock();
                }
                else if(editok ==1)
                {
                    if (EditHour == 1)
                    {
                        AlarmHour += 1;
                        if (AlarmHour == 24)
                        AlarmHour = 0;
                        Edit_hour();
                    }

                    if (EditMin == 1)
                    {
                        AlarmMin += 1;
                        if (AlarmMin == 60)
                        AlarmMin = 0;
                        Edit_minute();
                    }

                }
            }
            if (menu == 3)
            {
                submenu += 1;
                if (submenu > numOfEntries + 1)
                submenu = 1;
                if (submenu < numOfEntries + 1)
                {
                    Deltimes.tm_hour = MemHour[submenu -1];
                    Deltimes.tm_min = MemMin[submenu - 1];
                    DisplayDel();
                }
                else if(submenu == numOfEntries + 1)
                {
                    Delset = 1;
                    DelSetting();
                }

            }
        }
        if (submenu == 0)
        {
            menu += 1;
            if (menu == 4)
            {
                menu = 0;
            }
            SelectMenu();
        }

    }
    if (PIND & 0b00100000) //Back
    {
    }
    else {


        if (menu > 0)
        {
            menu = 0;
            submenu = 0;
            SelectMenu();
        }

    }
    sei();
    main();
    disinit = 1;
}
```