
MyButtonClockLCD.c

```
#include "main.h"
```

```
//-----
```

```
#define BUTTONPORT PORTC
```

```
#define BUTTONPORT1 PORTC1
```

```
#define BUTTONPORT2 PORTC2
```

```
#define BUTTONPORT3 PORTC3
```

```
#define BUTTONPIN PINC
```

```
#define BUTTONPIN1 PINC1
```

```
#define BUTTONPIN2 PINC2
```

```
#define BUTTONPIN3 PINC3
```

```
#define BUTTONDDR DDRC
```

```
#define BUTTONDDR1 DDRC1
```

```
#define BUTTONDDR2 DDRC2
```

```
#define BUTTONDDR3 DDRC3
```

```
#define CLOCKMODE0 0
```

```
#define CLOCKMODEDATE 1
```

```

#define CLOCKMODEMONTH 2
#define CLOCKMODEYEAR 3
#define CLOCKMODEDAY 4
#define CLOCKMODEHOUR 5
#define CLOCKMODEMIN 6
#define CLOCKMODESEC 7
//-----
unsigned char sec,min,hour,day,date,month,year;
//-----
void port_ini(void)
{
    PORTD=0x00;
    DDRD=0xFF;
    BUTTONDDR &= ~((1<<BUTTONDDR3)|(1<<BUTTONDDR2)|(1<<BUTTONDDR1));//ножки
кнопок на вход
    BUTTONPORT |= (1<<BUTTONPORT3)|(1<<BUTTONPORT2)|(1<<BUTTONPORT1);//подтянем
резисторы к лапкам кнопок
}
//-----
void ModifyRTC(char index, char direction)

```

```
{  
    I2C_StartCondition();  
    I2C_SendByte(0b11010000);  
    switch(index){  
        case 1: //число даты  
            I2C_SendByte(4); //Переходим на 0x04 - байт числа даты  
            //вперед  
            if (direction==1)  
            {  
                if (month==2) //февраль  
                {  
                    if(year%4==0) //високосный год  
                    {  
                        if(date<29) I2C_SendByte(RTC_ConvertFromBinDec(date+1));  
                        else I2C_SendByte(RTC_ConvertFromBinDec(1));  
                    }  
                }  
            }  
            else  
            {  
                if(date<28) I2C_SendByte(RTC_ConvertFromBinDec(date+1));  
                else I2C_SendByte(RTC_ConvertFromBinDec(1));  
            }  
        }  
    }  
}
```

```

    }
}
else if ((month==4)|(month==6)|(month==9)|(month==11))
{
    if(date<30) I2C_SendByte(RTC_ConvertFromBinDec(date+1));
    else I2C_SendByte(RTC_ConvertFromBinDec(1));
}
else
{
    if(date<31) I2C_SendByte(RTC_ConvertFromBinDec(date+1));
    else I2C_SendByte(RTC_ConvertFromBinDec(1));
}
}
//назад
if (direction==0)
{
    if (month==2) //февраль
    {
        if(year%4==0) //високосный год
        {

```

```

        if(date>1) I2C_SendByte(RTC_ConvertFromBinDec(date-1));
        else I2C_SendByte(RTC_ConvertFromBinDec(29));
    }
    else
    {
        if(date>1) I2C_SendByte(RTC_ConvertFromBinDec(date-1));
        else I2C_SendByte(RTC_ConvertFromBinDec(28));
    }
}
else if ((month==4)|(month==6)|(month==9)|(month==11))
{
    if(date>1) I2C_SendByte(RTC_ConvertFromBinDec(date-1));
    else I2C_SendByte(RTC_ConvertFromBinDec(30));
}
else
{
    if(date>1) I2C_SendByte(RTC_ConvertFromBinDec(date-1));
    else I2C_SendByte(RTC_ConvertFromBinDec(31));
}
}

```

```
break;
case 2: //месяц
    I2C_SendByte(5); //Переходим на 0x05 - байт месяца
    //вперед
    if (direction==1)
    {
        if(month<12) I2C_SendByte(RTC_ConvertFromBinDec(month+1));
        else I2C_SendByte(RTC_ConvertFromBinDec(1));
    }
    //назад
    if (direction==0)
    {
        if(month>1) I2C_SendByte(RTC_ConvertFromBinDec(month-1));
        else I2C_SendByte(RTC_ConvertFromBinDec(12));
    }
break;
case 3: //год
    I2C_SendByte(6); //Переходим на 0x06 - байт года
    //вперед
    if (direction==1)
```

```
{
    if(year<99) I2C_SendByte(RTC_ConvertFromBinDec(year+1));
    else I2C_SendByte(RTC_ConvertFromBinDec(1));
}
//назад
if (direction==0)
{
    if(year>1) I2C_SendByte(RTC_ConvertFromBinDec(year-1));
    else I2C_SendByte(RTC_ConvertFromBinDec(99));
}
break;
case 4: //день недели
    I2C_SendByte(3); //Переходим на 0x03 - байт дня недели
    //вперед
    if (direction==1)
    {
        if(day<7) I2C_SendByte(RTC_ConvertFromBinDec(day+1));
        else I2C_SendByte(RTC_ConvertFromBinDec(1));
    }
    //назад
```

```
if (direction==0)
{
    if(day>1) I2C_SendByte(RTC_ConvertFromBinDec(day-1));
    else I2C_SendByte(RTC_ConvertFromBinDec(7));
}
break;
case 5: //часы
    I2C_SendByte(2); //Переходим на 0x02 - байт часов
    //вперед
    if (direction==1)
    {
        if(hour<23) I2C_SendByte(RTC_ConvertFromBinDec(hour+1));
        else I2C_SendByte(RTC_ConvertFromBinDec(0));
    }
    //назад
    if (direction==0)
    {
        if(hour>0) I2C_SendByte(RTC_ConvertFromBinDec(hour-1));
        else I2C_SendByte(RTC_ConvertFromBinDec(23));
    }
}
```



```
break;
case 6: //минуты
    I2C_SendByte(1); //Переходим на 0x01 - байт минут
    //вперед
    if (direction==1)
    {
        if(min<59) I2C_SendByte(RTC_ConvertFromBinDec(min+1));
        else I2C_SendByte(RTC_ConvertFromBinDec(0));
    }
    //назад
    if (direction==0)
    {
        if(min>0) I2C_SendByte(RTC_ConvertFromBinDec(min-1));
        else I2C_SendByte(RTC_ConvertFromBinDec(59));
    }
break;
case 7: //секунды
    I2C_SendByte(0); //Переходим на 0x00 - байт секунд
    //сброс секунд в 0
    if (direction==1)
```

```

        {
            I2C_SendByte(RTC_ConvertFromBinDec(0));
        }
        //сброс секунд в 30
        if (direction==0)
        {
            I2C_SendByte(RTC_ConvertFromBinDec(30));
        }
        break;
    }
    I2C_StopCondition();
}
//-----
char ButtonPress (char index)
{
    char result=0, butcount=0, butstate=0;

    switch(index){
        case 1:
            while (butstate==0)

```

```
{  
    if(!(BUTTONPIN&(1<<BUTTONPIN1)))//Кнопка 1 нажата  
    {  
        if(butcount < 5)  
        {  
            butcount++;  
        }  
        else  
        {  
            result = 1;  
            butstate=1;  
        }  
    }  
    else  
    {  
        if(butcount > 0)  
        {  
            butcount--;  
        }  
        else
```

```

        {
            butstate=1;
        }
    }
}
break;
case 2:
while (butstate==0)
{
    if(!(BUTTONPIN&(1<<BUTTONPIN2))){//Кнопка 2 нажата
    {
        if(butcount < 5)
        {
            butcount++;
        }
        else
        {
            result = 2;
            butstate=1;
        }
    }
}

```

```
    }  
    else  
    {  
        if(butcount > 0)  
        {  
            butcount--;  
        }  
        else  
        {  
            butstate=1;  
        }  
    }  
}  
break;  
case 3:  
while (butstate==0)  
{  
    if(!(BUTTONPIN&(1<<BUTTONPIN3)))//Кнопка 3 нажата  
    {  
        if(butcount < 5)
```

```
        {  
            butcount++;  
        }  
    else  
    {  
        result = 3;  
        butstate=1;  
    }  
}  
else  
{  
    if(butcount > 0)  
    {  
        butcount--;  
    }  
    else  
    {  
        butstate=1;  
    }  
}
```

```

        }
        break;
    }
    return result;
}
//-----
int main(void)
{
    unsigned int tt=0; //переменная для хранения температуры
    unsigned char button1state=0; //статус кнопки 1
    unsigned char clockmode=CLOCKMODE0; //обычный режим показаний дисплея
    char blinkstate=0; //переменная для мигания определенным показателем на дисплее

    port_ini(); //Инициализируем порты
    I2C_Init(); //Инициализируем шину I2C
    LCD_ini(); //Инициализируем дисплей
    clearlcd(); //Очистим дисплей

    //Устанавливаем время
    //    I2C_StartCondition();

```

```
// I2C_SendByte(0b11010000);  
// I2C_SendByte(0); //Переходим на 0x00  
// I2C_SendByte(RTC_ConvertFromBinDec(0)); //секунды  
// I2C_SendByte(RTC_ConvertFromBinDec(31)); //минуты  
// I2C_SendByte(RTC_ConvertFromBinDec(20)); //часы  
// I2C_SendByte(RTC_ConvertFromBinDec(5)); //день недели  
// I2C_SendByte(RTC_ConvertFromBinDec(29)); //дата  
// I2C_SendByte(RTC_ConvertFromBinDec(1)); //месяц  
// I2C_SendByte(RTC_ConvertFromBinDec(16)); //год  
// I2C_StopCondition();
```

```
while(1)
```

```
{
```

```
    //Читаем время
```

```
    I2C_SendByteByADDR(0,0b11010000);    //переходим на адрес 0
```

```
    _delay_ms(50);
```

```
    I2C_StartCondition(); //Отправим условие START
```

```
    I2C_SendByte(0b11010001); //отправим в устройство бит чтения
```

```
    sec = I2C_ReadByte();
```

```
    min = I2C_ReadByte();
```



```
hour = I2C_ReadByte();
day = I2C_ReadByte();
date = I2C_ReadByte();
month = I2C_ReadByte();
year = I2C_ReadLastByte();
I2C_StopCondition(); //Отправим условие STOP
sec = RTC_ConvertFromDec(sec); //Преобразуем в десятичный формат
min = RTC_ConvertFromDec(min); //Преобразуем в десятичный формат
hour = RTC_ConvertFromDec(hour); //Преобразуем в десятичный формат
day = RTC_ConvertFromDec(day); //Преобразуем в десятичный формат
year = RTC_ConvertFromDec(year); //Преобразуем в десятичный формат
month = RTC_ConvertFromDec(month); //Преобразуем в десятичный формат
date = RTC_ConvertFromDec(date); //Преобразуем в десятичный формат
setpos(0,0); //Ставим курсор на начало координат
if(ButtonPress(1)==1)//Кнопка 1 нажата
{
    if (clockmode==CLOCKMODE0)
    {
        clockmode=CLOCKMODEDATE; //перейдем в режим перевода даты
        blinkstate=0; //сбросим счетчик мигания
    }
}
```

```
        button1state=1;//статус 1 кнопки
    }
}
if(clockmode!=CLOCKMODEDATE)
{
    sendcharlcd(date/10+0x30);//Преобразуем число в код числа
    sendcharlcd(date%10+0x30);//Преобразуем число в код числа
}
else //если режим перевода даты
{
    if (blinkstate==0)
    {
        sendcharlcd(' ');
        sendcharlcd(' ');
        blinkstate=1;
    }
    else
    {
        sendcharlcd(date/10+0x30);//Преобразуем число в код числа
        sendcharlcd(date%10+0x30);//Преобразуем число в код числа
    }
}
```

месяца

```
        blinkstate=0;
    }
    if(ButtonPress(1)==1)//Кнопка 1 нажата
    {
        if(button1state==0) //опросим статус, чтобы сразу не перейти в режим перевода
        {
            clockmode=CLOCKMODEMONTH; //перейдем в режим перевода месяца
            button1state=1;
        }
    }
    if (clockmode==CLOCKMODEDATE) button1state=0;//сбросим статус
    if(ButtonPress(2)==2)//Кнопка 2 нажата
        ModifyRTC(1,0);//1-дата,0-назад
    if(ButtonPress(3)==3)//Кнопка 3 нажата
        ModifyRTC(1,1);//1-дата,1-вперед
    }
    sendcharlcd('.');
    if(clockmode!=CLOCKMODEMONTH)
    {
```

```
    sendcharlcd(month/10+0x30);//Преобразуем число в код числа
    sendcharlcd(month%10+0x30);//Преобразуем число в код числа
}
else //если режим перевода месяца
{
    if (blinkstate==0)
    {
        sendcharlcd(' ');
        sendcharlcd(' ');
        blinkstate=1;
    }
    else
    {
        sendcharlcd(month/10+0x30);//Преобразуем число в код числа
        sendcharlcd(month%10+0x30);//Преобразуем число в код числа
        blinkstate=0;
    }
    if(ButtonPress(1)==1)//Кнопка 1 нажата
    {
```

месяца

```
if(button1state==0) //опросим статус, чтобы сразу не перейти в режим перевода
{
    clockmode=CLOCKMODEYEAR; //перейдем в режим перевода года
    button1state=1;
}
}
if (clockmode==CLOCKMODEMONTH) button1state=0; //сбросим статус
if(ButtonPress(2)==2) //Кнопка 2 нажата
    ModifyRTC(2,0); //2-месяц, 0-назад
if(ButtonPress(3)==3) //Кнопка 3 нажата
    ModifyRTC(2,1); //2-месяц, 1-вперед
}
sendcharlcd('.');
if(clockmode!=CLOCKMODEYEAR)
{
    sendcharlcd('2');
    sendcharlcd('0');
    sendcharlcd(year/10+0x30); //Преобразуем число в код числа
    sendcharlcd(year%10+0x30); //Преобразуем число в код числа
}
```

```
}  
else //если режим перевода года  
{  
    if (blinkstate==0)  
    {  
        sendcharlcd(' ');  
        sendcharlcd(' ');  
        sendcharlcd(' ');  
        sendcharlcd(' ');  
        blinkstate=1;  
    }  
    else  
    {  
        sendcharlcd('2');  
        sendcharlcd('0');  
        sendcharlcd(year/10+0x30); //Преобразуем число в код числа  
        sendcharlcd(year%10+0x30); //Преобразуем число в код числа  
        blinkstate=0;  
    }  
    if(ButtonPress(1)==1) //Кнопка 1 нажата
```

месяца

```
{
    if(button1state==0) //опросим статус, чтобы сразу не перейти в режим перевода
    {
        clockmode=CLOCKMODEDAY; //перейдем в режим перевода дня недели
        button1state=1;
    }
}
if (clockmode==CLOCKMODEYEAR) button1state=0; //сбросим статус
if(ButtonPress(2)==2) //Кнопка 2 нажата
ModifyRTC(3,0); //3-год, 0-назад
if(ButtonPress(3)==3) //Кнопка 3 нажата
ModifyRTC(3,1); //3-год, 1-вперед
}
sendcharlcd(' ');
if(clockmode!=CLOCKMODEDAY)
{
    sendcharlcd('-');
    sendcharlcd(day+0x30); //Преобразуем число в код числа
    sendcharlcd('-');
```

```
}  
else //если режим перевода года  
{  
    if (blinkstate==0)  
    {  
        sendcharlcd(' ');  
        sendcharlcd(' ');  
        sendcharlcd(' ');  
        blinkstate=1;  
    }  
    else  
    {  
        sendcharlcd('-');  
        sendcharlcd(day+0x30); //Преобразуем число в код числа  
        sendcharlcd('-');  
        blinkstate=0;  
    }  
    if(ButtonPress(1)==1) //Кнопка 1 нажата  
    {  
        if(button1state==0)
```



```
        {
            clockmode=CLOCKMODEHOUR; //перейдем в режим перевода часов
            button1state=1;
        }
    }
    if (clockmode==CLOCKMODEDAY) button1state=0; //сбросим статус
    if(ButtonPress(2)==2) //Кнопка 2 нажата
        ModifyRTC(4,0); //4-день недели, 0-назад
    if(ButtonPress(3)==3) //Кнопка 3 нажата
        ModifyRTC(4,1); //4-день недели, 1-вперед
}
setpos(0,1); //Ставим курсор на начало координат
if(clockmode!=CLOCKMODEHOUR)
{
    sendcharlcd(hour/10+0x30); //Преобразуем число в код числа
    sendcharlcd(hour%10+0x30); //Преобразуем число в код числа
}
else //если режим перевода часов
{
    if (blinkstate==0)
```

```

{
    sendcharlcd(' ');
    sendcharlcd(' ');
    blinkstate=1;
}
else
{
    sendcharlcd(hour/10+0x30); //Преобразуем число в код числа
    sendcharlcd(hour%10+0x30); //Преобразуем число в код числа
    blinkstate=0;
}
if(ButtonPress(1)==1) //Кнопка 1 нажата
{
    if(button1 state==0) //опросим статус, чтобы сразу не перейти в режим перевода
        {
            clockmode=CLOCKMODEMIN; //перейдем в режим перевода минут
            button1 state=1;
        }
}

```

месяца

```
    if (clockmode==CLOCKMODEHOUR) button1state=0;//сбросим статус
    if(ButtonPress(2)==2)//Кнопка 2 нажата
        ModifyRTC(5,0);//5-часы,0-назад
    if(ButtonPress(3)==3)//Кнопка 3 нажата
        ModifyRTC(5,1);//5-часы,1-вперед
}
sendcharlcd(':');
if(clockmode!=CLOCKMODEMIN)
{
    sendcharlcd(min/10+0x30);//Преобразуем число в код числа
    sendcharlcd(min%10+0x30);//Преобразуем число в код числа
}
else //если режим перевода минут
{
    if (blinkstate==0)
    {
        sendcharlcd(' ');
        sendcharlcd(' ');
        blinkstate=1;
    }
}
```

```

else
{
    sendcharlcd(min/10+0x30);//Преобразуем число в код числа
    sendcharlcd(min%10+0x30);//Преобразуем число в код числа
    blinkstate=0;
}
if(ButtonPress(1)==1)//Кнопка 1 нажата
{
    if(button1state==0) //опросим статус, чтобы сразу не перейти в режим перевода
    {
        clockmode=CLOCKMODESEC; //перейдем в режим синхронизации секунд
        button1state=1;
    }
}
if (clockmode==CLOCKMODEMIN) button1state=0;//сбросим статус
if(ButtonPress(2)==2)//Кнопка 2 нажата
ModifyRTC(6,0);//6-минуты,0-назад
if(ButtonPress(3)==3)//Кнопка 3 нажата
ModifyRTC(6,1);//6-минуты,1-вперед

```

месяца

```
}
sendcharlcd(':');
if(clockmode!=CLOCKMODESEC)
{
    sendcharlcd(sec/10+0x30);//Преобразуем число в код числа
    sendcharlcd(sec%10+0x30);//Преобразуем число в код числа
}
else //если режим синхронизации секунд
{
    if (blinkstate==0)
    {
        sendcharlcd(' ');
        sendcharlcd(' ');
        blinkstate=1;
    }
    else
    {
        sendcharlcd(sec/10+0x30);//Преобразуем число в код числа
        sendcharlcd(sec%10+0x30);//Преобразуем число в код числа
        blinkstate=0;
    }
}
```

```
}  
if(ButtonPress(1)==1)//Кнопка 1 нажата  
{  
    if(button1 state==0)  
    {  
        clockmode=CLOCKMODE0; //перейдем в обычный режим хода  
        button1 state=1;  
    }  
}  
if (clockmode==CLOCKMODESEC) button1 state=0;//сбросим статус  
if(ButtonPress(2)==2)//Кнопка 2 нажата  
ModifyRTC(7,0);//7-минуты,0-установка 30  
if(ButtonPress(3)==3)//Кнопка 3 нажата  
ModifyRTC(7,1);//7-минуты,1-установка 0  
}  
sendcharlcd(' ');  
tt = converttemp(dt_check()); //измеряем температуру  
sendcharlcd(tt/10+0x30);//Преобразуем число в код числа  
sendcharlcd(tt%10+0x30);//Преобразуем число в код числа  
sendcharlcd('*');
```

```
        sendcharlcd('C');  
    }  
}
```

main.h

```
#ifndef MAIN_H_  
#define MAIN_H_  
  
#define F_CPU 8000000UL  
  
#include <avr/io.h>  
#include <avr/interrupt.h>  
#include <util/delay.h>  
#include <stdio.h>  
#include <stdlib.h>  
  
#include "lcd.h"  
#include "twi.h"  
#include "RTC.h"
```

```
#include "DS18B20.h"
```

```
#endif /* MAIN_H_ */
```

```
#endif /* MAIN_H_ */
```

```
twi.c
```

```
#include "twi.h"
```

```
void I2C_Init(void)
```

```
{  
    TWBR=0x20;//скорость передачи (при 8 мГц получается 100 кГц, что и необходимо для общения  
с ds1307)  
}
```

```
void I2C_StartCondition(void)
```

```
{
```



```

    TWCR = (1<<TWINT)|(1<<TWSTA)|(1<<TWEN);
    while (!(TWCR & (1<<TWINT)));//подождем пока установится TWIN
}

void I2C_StopCondition(void)
{
    TWCR = (1<<TWINT)|(1<<TWSTO)|(1<<TWEN);
}

void I2C_SendByte(unsigned char c)
{
    TWDR = c;//запишем байт в регистр данных
    TWCR = (1<<TWINT)|(1<<TWEN);//включим передачу байта
    while (!(TWCR & (1<<TWINT)));//подождем пока установится TWIN
}

void I2C_SendByteByADDR(unsigned char c,unsigned char addr)
{
    I2C_StartCondition(); // Отправим условие START
    I2C_SendByte(addr); // Отправим в шину адрес устройства + бит чтения-записи
}

```

```

    I2C_SendByte(c); // Отправим байт данных
    I2C_StopCondition(); // Отправим условие STOP
}

unsigned char I2C_ReadByte(void)
{
    TWCR = (1<<TWINT)|(1<<TWEN)|(1<<TWEA);
    while (!(TWCR & (1<<TWINT))); //ожидание установки бита TWIN
    return TWDR; //читаем регистр данных
}

unsigned char I2C_ReadLastByte(void)
{
    TWCR = (1<<TWINT)|(1<<TWEN);
    while (!(TWCR & (1<<TWINT))); //ожидание установки бита TWIN
    return TWDR; //читаем регистр данных
}

```

twi.h

```
#ifndef TWI_H_
#define TWI_H_

#include "main.h"

void I2C_Init (void); //инициализация i2c
void I2C_StartCondition(void); //Отправим условие START
void I2C_StopCondition(void); //Отправим условие STOP
void I2C_SendByte(unsigned char c); //передача байта в шину
void I2C_SendByteByADDR(unsigned char c,unsigned char addr); //передача байта в шину на
устройство по адресу
unsigned char I2C_ReadByte(void); //читаем байт
unsigned char I2C_ReadLastByte(void); //читаем последний байт
```

```
#endif /* TWI_H_ */
```

=====

RTC.c

```
#include "RTC.h"
```

```
unsigned char RTC_ConvertFromDec(unsigned char c)
{
    unsigned char ch = ((c>>4)*10+(0b00001111&c));
    return ch;
}
```

```
unsigned char RTC_ConvertFromBinDec(unsigned char c)
{
    unsigned char ch = ((c/10)<<4)|(c%10);
    return ch;
}
```

RTC.h

```
#ifndef RTC_H_
#define RTC_H_

#include "main.h"
```

```
unsigned char RTC_ConvertFromDec(unsigned char c); //перевод двоично-десятичного числа в десятичное
```

```
unsigned char RTC_ConvertFromBinDec(unsigned char c); //перевод десятичного числа в двоично-десятичное
```

```
#endif /* RTC_H_ */
```

```
lcd.c
```

```
#include "lcd.h"
```

```
//-----  
void sendhalfbyte(unsigned char c)
```

```
{
```

```
    c<<=4;
```

```
    e1; //включаем линию E
```

```
    _delay_us(50);
```

```
    PORTD&=0b00001111; //стираем информацию на входах DB4-DB7, остальное не трогаем
```

```
    PORTD|=c;
```

```
    e0; //выключаем линию E
```

```

    _delay_us(50);
}
//-----
void sendbyte(unsigned char c, unsigned char mode)
{
    if (mode==0) rs0;
    else      rs1;
    unsigned char hc=0;
    hc=c>>4;
    sendhalfbyte(hc); sendhalfbyte(c);
}
//-----
void sendcharlcd(unsigned char c)
{
    sendbyte(c,1);
}
//-----
void setpos(unsigned char x, unsigned y)
{
    char adress;

```

```

    adress=(0x40*y+x)|0b10000000;
    sendbyte(adress, 0);
}
//-----
void LCD_ini(void)
{
    _delay_ms(15); //Ждем 15 мс (стр 45)
    sendhalfbyte(0b00000011);
    _delay_ms(4);
    sendhalfbyte(0b00000011);
    _delay_us(100);
    sendhalfbyte(0b00000011);
    _delay_ms(1);
    sendhalfbyte(0b00000010);
    _delay_ms(1);
    sendbyte(0b00101000, 0); //4бит-режим (DL=0) и 2 линии (N=1)
    _delay_ms(1);
    sendbyte(0b00001100, 0); //включаем изображение на дисплее (D=1), курсоры никакие не
    включаем (C=0, B=0)
    _delay_ms(1);

```

```

    sendbyte(0b000000110, 0); //курсор (хоть он у нас и невидимый) будет двигаться влево
    _delay_ms(1);
}
//-----
void clearlcd(void)
{
    sendbyte(0b000000001, 0);
    _delay_us(1500);
}
//-----
void str_lcd (char str1[])
{
    wchar_t n;
    for(n=0;str1[n]!='\0';n++)
        sendcharlcd(str1[n]);
}
//-----

```

lcd.h

```
#ifndef LCD_H_  
#define LCD_H_
```

```
#include "main.h"
```

```
//-----
```

```
void LCD_ini(void);
```

```
void setpos(unsigned char x, unsigned y);
```

```
void str_lcd (char str1[]);
```

```
void clearlcd(void);
```

```
void sendcharlcd(unsigned char c);
```

```
//-----
```

```
#define e1 PORTD|=0b00001000 // установка линии E в 1
```

```
#define e0 PORTD&=0b11110111 // установка линии E в 0
```

```
#define rs1PORTD|=0b00000100 // установка линии RS в 1 (данные)
```

```
#define rs0PORTD&=0b11111011 // установка линии RS в 0 (команда)
```

```
//-----
```

```
#endif /* LCD_H_ */
```

```
=====
```

DS18B20.c

```
#include "DS18B20.h"
```

```
char dt_testdevice(void) //dt - digital thermometer | определим, есть ли устройство на шине
{
    char stektemp=SREG; //сохраним значение стека
    cli(); //запрещаем прерывания
    char dt;
    DDRTMP |= 1<<BITTEMP; //притягиваем шину
    _delay_us(485); //задержка как минимум 480 микросекунд
    DDRTMP &= ~(1<<BITTEMP); //отпускаем шину
    _delay_us(65); //задержка как минимум 60 микросекунд
    if((PINTMP & (1<<BITTEMP)) == 0x00) //проверяем, ответит ли устройство
    {
        dt=1; //устройство есть
    }
    else dt=1; //устройства нет
    SREG=stektemp; //вернем показания стека на место
    _delay_us(420); //задержка минимум 480 микросекунд, но хватит и 420, т.к. это с учетом времени
    прошедших команд
    return dt;
}
```

```
}
```

```
// функция отправки бита на устройство
```

```
void dt_sendbit (char bt)
```

```
{
```

```
    char stektemp=SREG; //сохраним значение стека
```

```
    cli(); //запрещаем прерывания
```

```
    DDRTMP |= 1<<BITTEMP; //притягиваем шину
```

```
    _delay_us(2);
```

```
    if(bt)
```

```
        DDRTMP &= ~(1<<BITTEMP); //отпускаем шину
```

```
    _delay_us(65);
```

```
    DDRTMP &= ~(1<<BITTEMP); //отпускаем шину
```

```
    SREG=stektemp; //вернем показания стека на место
```

```
}
```

```
//функция отправки байта на устройство
```

```
void dt_sendbyte(unsigned char bt)
```

```
{
```

```
    char i;
```

```

for(i = 0; i < 8; i++)//посылаем отдельно каждый бит
{
    if((bt & (1<<i)) == 1<<i)//посылаем 1
        dt_sendbit(1);
    else//посылаем 0
        dt_sendbit(0);
}
}

```

//функция чтения одного бита с устройства

```

char dt_readbit (void)
{
    char stektemp=SREG;//сохраним значение стека
    cli(); //запрещаем прерывания
    char bt; //переменная хранения бита
    DDRTMP |= 1<<BITTEMP; //притягиваем шину
    _delay_us(2);
    DDRTMP &= ~(1<<BITTEMP); //отпускаем шину
    _delay_us(13);
    bt=(PINTMP & (1<<BITTEMP))>>BITTEMP; //читаем бит
}

```

```
    _delay_us(45);  
    SREG=stektemp; //вернем показания стека на место  
    return bt;  
}
```

//функция чтения байта с устройства

```
unsigned char dt_readbyte()  
{  
    char c=0;  
    char i;  
    for(i = 0; i < 8; i++)  
        c|= dt_readbit()<<i; //читаем бит  
    return c;  
}
```

//функция преобразования показаний датчика в температуру

```
int dt_check(void)  
{  
    unsigned char bt; //переменная для считывания бита  
    unsigned int tt=0;
```

```

if(dt_testdevice()==1)//если устройство нашлось
{
    dt_sendbyte(NOID);//пропустить идентификацию, тк у нас только одно устройство на шине
    dt_sendbyte(T_CONVERT);//измеряем температуру
    _delay_ms(100);//в 9битном режим преобразования - 93.75ms (tCONV/8)
    dt_testdevice();//снова используем те же манипуляции с шиной что и при проверке ее
    присутствия
    dt_sendbyte(NOID);
    dt_sendbyte(READ_DATA);//передадим байты устройству (в первых двух байтах
    температура)
    bt = dt_readbyte(); //читаем младший бит
    tt = dt_readbyte(); //читаем бит MS
    tt = (tt<<8)|bt;//сдвигаем старший влево, младший пишем на его место, тем самым
    получаем общий результат
}
return tt;
}

//преобразование температуры в единицы
char converttemp (unsigned int tt)

```

```
{  
    char t = tt >> 4; // сдвиг и отсечение старшего байта  
    return t;  
}
```

DS18B20.h

```
#ifndef DS18B20_H_  
#define DS18B20_H_  
  
#include "main.h"  
  
#define NOID 0xCC // Пропустить идентификацию  
#define T_CONVERT 0x44 // Код измерения температуры  
#define READ_DATA 0xBE // Передача байтов ведущему  
  
#define PORTTEMP PORTD  
#define DDRTEMP DDRD  
#define PINTEMP PIND  
#define BITTEMP 1
```

```
int dt_check(void); //функция преобразования показаний датчика в температуру  
char converttemp (unsigned int tt); //преобразование температуры в единицы
```

```
#endif /* DS18B20_H_ */
```