
MyADCLCD.c

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

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

```
void port_ini(void)
```

```
{
```

```
    PORTD=0x00;
```

```
    DDRD=0xFF;
```

```
}
```

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

```
int main(void)
```

```
{
```

```
    float n;
```

```
    adc_value=0;
```

```
    adc_counter=0;
```

```
    adc_tmp=0;
```

```
    port_ini(); //Инициализируем порты
```

```
    LCD_ini(); //Инициализируем дисплей
```

```
    ADC_Init(); //Инициализируем АЦП
```

```
sei();//разрешаем глобально прерывания
```

```
clearlcd(); //Очистим дисплей
```

```
while(1)
```

```
{
```

```
    setpos(0,0);
```

```
    sendcharlcd(adc_value/1000+0x30);//Преобразуем число в код числа
```

```
    sendcharlcd(((adc_value%1000)/100+0x30));//Преобразуем число в код числа
```

```
    sendcharlcd((adc_value%100)/10+0x30);//Преобразуем число в код числа
```

```
    sendcharlcd(adc_value%10+0x30);//Преобразуем число в код числа
```

```
    setpos(8,0);
```

```
    n= (float) adc_value / 400;
```

```
    sendcharlcd((unsigned char) n +0x30);//Преобразуем число в код числа
```

```
    sendcharlcd('.');//Преобразуем число в код числа
```

```
    sendcharlcd(((unsigned char) (n*10))%10 +0x30);//Преобразуем число в код числа
```

```
    sendcharlcd(((unsigned char) (n*100))%10 +0x30);//Преобразуем число в код числа
```

```
    _delay_ms(500);
```

```
}
```

```
}
```

```
=====
```

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>
```

```
unsigned int adc_value,adc_counter,adc_tmp;
```

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

```
#include "adc.h"
```

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

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_ */
```

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(0b00000110, 0); //курсор (хоть он у нас и невидимый) будет двигаться влево
    _delay_ms(1);
}
//-----
void clearlcd(void)
{

```

```
    sendbyte(0b00000001, 0);  
    _delay_us(1500);  
}  
//-----  
void str_lcd (char str1[])  
{  
    wchar_t n;  
    for(n=0;str1[n]!='\0';n++)  
        sendcharlcd(str1[n]);  
}  
//-----
```

adc.h

```
#ifndef ADC_H_  
#define ADC_H_
```

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

```
void ADC_Init(void);
```



```
#endif /* ADC_H_ */
```

adc.c

```
#include "adc.h"
```

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

```
extern unsigned int adc_value,adc_counter,adc_tmp;
```

```
char high_adc=0, low_adc=0;
```

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

```
// Прерывание для АЦП
```

```
ISR(ADC_vect)
```

```
{
```

```
    low_adc = ADCL;
```

```
    high_adc = ADCH; //Верхняя часть регистра ADC должна быть считана последней, иначе не  
продолжится преобразование
```

```
    if(adc_counter<20) {adc_tmp += high_adc*256+low_adc; adc_counter++;}
```

```
    else {adc_value = adc_tmp/20; adc_counter=0;adc_tmp=0;}
```

```
}
```

```
//-----  
void ADC_Init(void)  
{  
    ADCSRA |= (1<<ADEN) // Разрешение использования АЦП  
    |(1<<ADSC) //Запуск преобразования  
    |(1 << ADFR) // Непрерывный режим работы АЦП  
    |(1<<ADPS2)|(1<<ADPS1)|(1<<ADPS0)//Делитель 128 = 64 кГц  
    |(1 << ADIE); // Разрешение прерывания от АЦП  
    ADMUX |= (1<<REFS1)|(1<<REFS0); //Внутренний Источник ОН 2,56в, вход ADC0  
}  
//-----
```