

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра САУ**

**Лабораторная работа №5**  
**по дисциплине «МПУ»**  
**Вариант 7**

Студенты гр. 6

\_\_\_\_\_

Преподаватель

\_\_\_\_\_

Санкт-Петербург

2018

**Цель работы.** Программирование процедур вывода символьной информации на жидкокристаллический индикатор с использованием параллельного порта МК.

**Задание:** задано дискретное дифференциальное уравнение  $y(k + 1) = y(k) + T f[y(k), u(k)]$ , где  $u(k)$  и  $y(k)$  – значения входного воздействия и управляющего сигнала в дискретные моменты времени  $t = kT$ , где  $T$  – период прерывания (шаг расчета).

В данной лабораторной работе необходимо реализовать это соотношение в виде программы для расчета управляющего сигнала  $y(k)$  на пяти шагах при начальном условии  $y(0) = 0$  и известных значениях  $u(k)$  на каждом шаге. Для индикации полученных значений  $y(k)$  необходимо на каждом шаге выводить их на индикатор. Формат вывода: Результат:  $Y(k) = X$ , где  $X$  – выводимое число, а  $k$  – номер шага.

№ варианта	Функция	T	Значения $u(k)$				
			0	1	2	3	4
7	$f = -5y + u/4 - 1$	1	8	28	48	68	88

Таблица 1. Данные задания.



Рис.1 Отображение символов из DDRAM в окне индикатора.

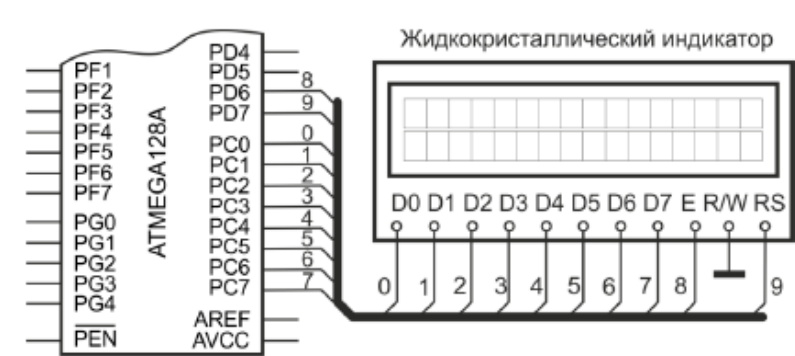
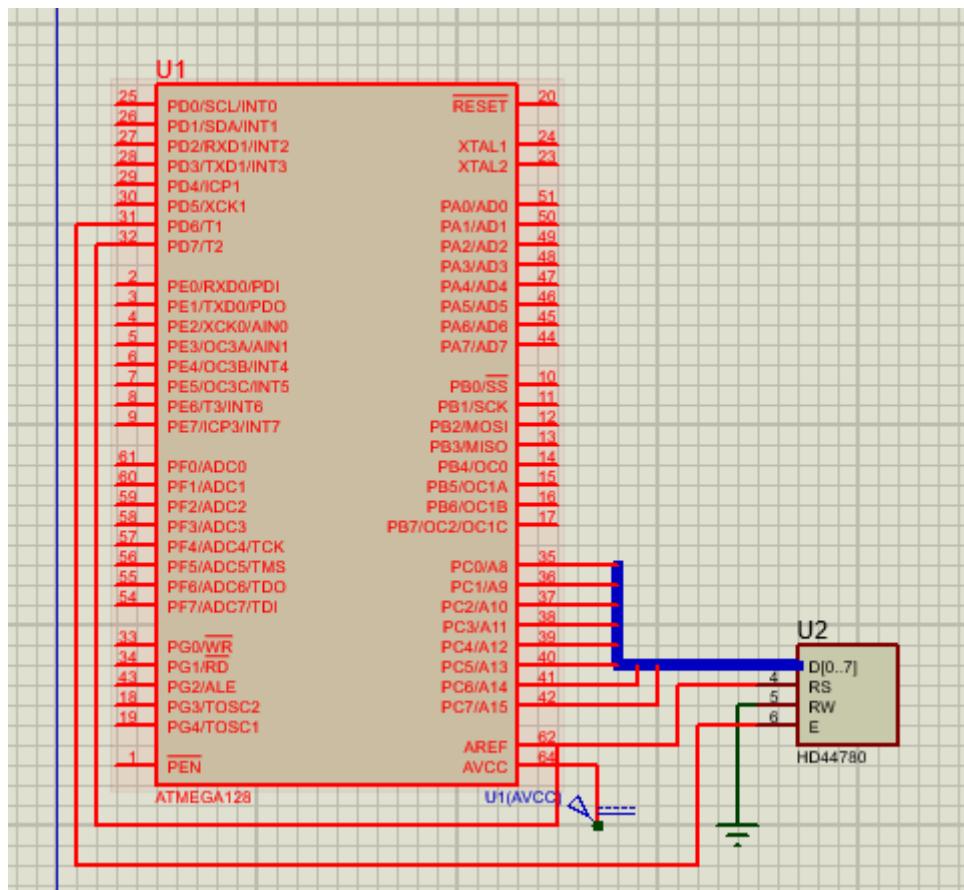


Рис.2 Схема подключения ЖКИ и МК

Proteus



Блок-схема



### Код программы.

```

/*
Variant 7
lab 5
*/

#include <io.h>
#include <mega128a.h>
#include <delay.h>

#define CMD 1
#define DATA 0
#define DISPLAY 1
#define CURSOR 0
#define RHIGHT 1
#define LEFT 0
#define ON 1
#define OFF 0
#define EIGHT 1
#define FOUR 0

```

```

#define ONE 0
#define TWO 1
#define EIGHT 1
#define FOUR 0
#define LOWERCASE 0
#define UPPERCASE 1
#define RS 7 // выбор регистра
#define E 6 // строб передачи

//прототипы функции для работы с дисплеем
void LCD_message(char cmd,int type );
void clear_display() ;
void default_display() ;
void shift_direction(int ID,int S);
void switch_display(int B,int D,int C);
void shift(int choose,int direction);
void display_setting(int data_bus_width,int line_number,int font);
void LCD_init(void) ;
void LCD_string(unsigned char *str);
void LCD_int(int value);
void LCD_data_string(int data, unsigned char i) ;

unsigned char Code(unsigned char symb);
unsigned char Digit (unsigned int d, unsigned char m);

void main(void)
{

    unsigned char T=1;
    int i, y, u[5]={8, 28, 48, 68, 88};

    DDRC = 0xFF;          // все разряды PORTC на выход
    DDRD|= ((1<<E)|(1<<RS)); // разряды PORTD на выход

    LCD_init(); //инициализация
    clear_display(); //Очистить дисплей и установить курсор в нулевую позицию (адрес 0)
    LCD_string("РЕЗУЛЬТАТ:");

    while (1)
    {
        clear_display() ;
        y=0;

        for (i=0; i<5; i++)
        {
            y = y + T * (-5*y+u[i]/4-1); //расчет заданной функции
            LCD_data_string(y, i); //вывод результат строкой i - аргумент функции y - значение
            delay_ms(1000);
        }
    }

}

//Вывод числового значения (max 5 десятичных разрядов и больше нуля)

```

```

void LCD_int(int value)
{
    int i ;
    unsigned char flag_first_num = 0 ;
    unsigned char number;
    if (value<0) LCD_message(0b10010110,DATA);//выводим минус
    for(i=1; i<=5; i++)
    {
        number = Digit(value,i);
        if(number != 0) //появление перового символа
        {
            flag_first_num = 1;
        }

        if(flag_first_num == 1)
        {

            LCD_message(number+'0',DATA);//выводим цифру
        }

    }
}

```

```

void LCD_message(char message,int type)
{
    //[]-----[]
    //| Назначение: запись кодов в регистр команд ЖКИ |
    //| Входные параметры: message - сообщение |
    //| Входные параметры: type - тип сообщения (код или данные) |
    //[]-----[]

    if(type) // 1
    PORTD &= ~(1<<RS); // выбор регистра команд RS=0
    else
    PORTD |= (1<<RS); // выбор регистра данных RS=1
    PORTC=message; // записать команду в порт PORTC
    PORTD|= (1<<E); // \ сформироватьна
    delay_us(5); // |выводе E строб 1-0
    PORTD&= ~(1<<E); // / передачи команды
    delay_ms(100); // задержка для завершения записи

}

```

```

void LCD_init(void)
{
    //[]-----[]
    //| Назначение: инициализация ЖКИ |
    //[]-----[]
    delay_ms (100); // задержка для установления
        // напряжения питания
    LCD_message(0x30,CMD); // \ вывод
    LCD_message(0x30,CMD); //| трех
    LCD_message(0x30,CMD); // / команд 0x30
    LCD_message(0x38,CMD); //8 разр.шина, 2 строки, 5 ? 7 точек
    LCD_message(0x0E,CMD); // включить ЖКИ и курсор, без мерцания
}

```

```

LCD_message(0x06,CMD); //инкремент курсора, без сдвига экрана
LCD_message(0x01,CMD); // очистить экран, курсор в начало
}

void LCD_data_string(int data, unsigned char i)
{

    unsigned char STR[5]={'Y','(', ')', '=', '\0'};

    unsigned char k=0;

    while (k<5)
    {

        if (k==2) LCD_message(i+'0',DATA); // вывод аргумента функции

        if (k==4)
        {
            LCD_int(data);
            break;
        }

        LCD_message(STR[k],DATA); //оформление
        k++;
        delay_ms(50);

    }

}

//вывод строковой информации на дисплей
void LCD_string(unsigned char* str)
{
    while(*str != '\0')
    {
        LCD_message(Code(*str++),DATA);
        // *str извлечение элемента по адресу в указателе str
        // str++ увеличение значение указателя на 1 (т.е. переход к следующему элементу массива)
    }
}

unsigned char Digit (unsigned int d, unsigned char m)
{
    //[]-----[]
    // Назначение: выделение цифр из разрядов пятиразрядного |
    // десятичного положительного числа |
    // Входные параметры: |
    // d - целое десятичное положительное число |
    // m - номер разряда (от 1 до 5, слева направо) |
    // Функция возвращает значение цифры в разряде m числа d |
    //[]-----[]
    unsigned char i = 5, a;
    while(i)
    {

```

```

    a = d%10; //если d < 0 то a<0
    if(i-- == m) break;
    d /= 10;
}
return(a);
}

```

```

unsigned char Code(unsigned char symb)

```

```

{

    //[ ]-----[ ]
    //[ Назначение: перекодировка символов кириллицы |
    //[ Входные параметры: symb – символ ASCII |
    //[ Функция возвращает код отображения символа |
    //[ ]-----[ ]
    unsigned char TabCon[] =
    {0x41,0xA0,0x42,0xA1,0xE0,0x45,0xA3,0xA4,0xA5,0xA6,0x4B,
0xA7,0x4D,0x48,0x4F,0xA8,0x50,0x43,0x54,0xA9,0xAA,0x58,
0xE1,0xAB,0xAC,0xE2,0xAD,0xAE,0x62,0xAF,0xB0,0xB1,0x61,
0xB2,0xB3,0xB4,0xE3,0x65,0xB6,0xB7,0xB8,0xB9,0xBA,0xBB,
0xBC,0xBD,0x6F,0xBE,0x70,0x63,0xBF,0x79,0x5C,0x78,0xE5,
0xC0,0xC1,0xE6,0xC2,0xC3,0xC4,0xC5,0xC6,0xC7}; //коды символов в кириллице
    return (symb >= 192 ? TabCon[symb-192]: symb);
}

```

```

//очистить дисплей и установить курсор в нулевую позицию(адрес 0 )

```

```

void clear_display()

```

```

{
    LCD_message(1,CMD);
};

```

```

//установить курсор в нулевую позицию, дисплей относительно буфера DDRAM в начальную позицию

```

```

void default_display()

```

```

{
    LCD_message(2,CMD);
};

```

```

//установить направление сдвига курсора при записи кода в DDRAM

```

```

// и разрешить(запретить) сдвиг окна вместе с курсором

```

```

// ID = RIGHT

```

```

// ID = LEFT

```

```

// S = ON разрешить сдвиг окна вместе с курсором

```

```

// S = OFF запретить сдвиг окна вместе с курсором

```

```

void shift_direction(int ID,int S)

```

```

{
    LCD_message((1<<2)|(ID<<1)|S,CMD);
}

```

```

//Включить(выключить) индикатор,зажечь(погасить) курсор.Сделать курсор мигающим

```

```

// B = ON курсор мигает

```

```

// B = OFF курсор не мигает

```

```

// D = ON включить индикатор

```

```

// D = OFF выключить индикатор

```

```

// C = ON зажечь курсор

```

```

// C = OFF погасить курсор

```



```

void switch_display(int B,int D,int C)
{
    LCD_message((1<<3)|(D<<2)|(C<<1)|B,CMD);
}

//сдвиг курсора или дисплея вправо или влево
// choose = DISPLAY
// choose = CURSOR
// direction = RIGHT
// direction = LEFT
void shift(int choose,int direction)
{
    LCD_message((1<<4)|(choose<<3)|(direction<<2),CMD);
}

//установить разрядность шины данных, количество строк, шрифт
//data_bus_width = EIGHT 8 бит
//data_bus_width = FOUR 4 бит
//line_number = ONE 1 строка
//line_number = TWO 2 строки
//font = LOWERCASE строчные
////font = UPPERCASE заглавные
void display_setting(int data_bus_width,int line_number,int font)
{
    LCD_message((1<<5)|(data_bus_width<<4)|(line_number<<3)|(font<<2),CMD);
}

```

**Вывод:** Программирование процедур вывода символьной информации на жидкокристаллический индикатор с использованием параллельного порта МК.