

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Вятский государственный университет»
Факультет автоматики и вычислительной техники
Кафедра электронных вычислительных машин

**Изучение принципов построения микроконтроллерных систем
на примере микроконтроллера AT89S8252**

Отчет по лабораторной работе № 1
дисциплины «Системы автоматизированного проектирования»

Выполнил студент группы ИВТ-42 _____ /Рзаев А. Э./
Выполнил студент группы ИВТ-42 _____ /Кодачигов А. А./
Выполнил студент группы ИВТ-42 _____ /Бессолицын А. А./
Выполнил студент группы ИВТ-42 _____ /Микрюков А. А./

Проверил доцент кафедры ЭВМ _____ /Скворцов А.А./

1 Задание на лабораторную работу

Ввод: клавиатура для ввода номера программы и периода смены состояния.

Вывод: Ж/К дисплей для вывода номера и названия текущей программы и текущего периода. Светодиоды всех цветов.

Структурная схема лабораторной установки представлена на рисунке 1.

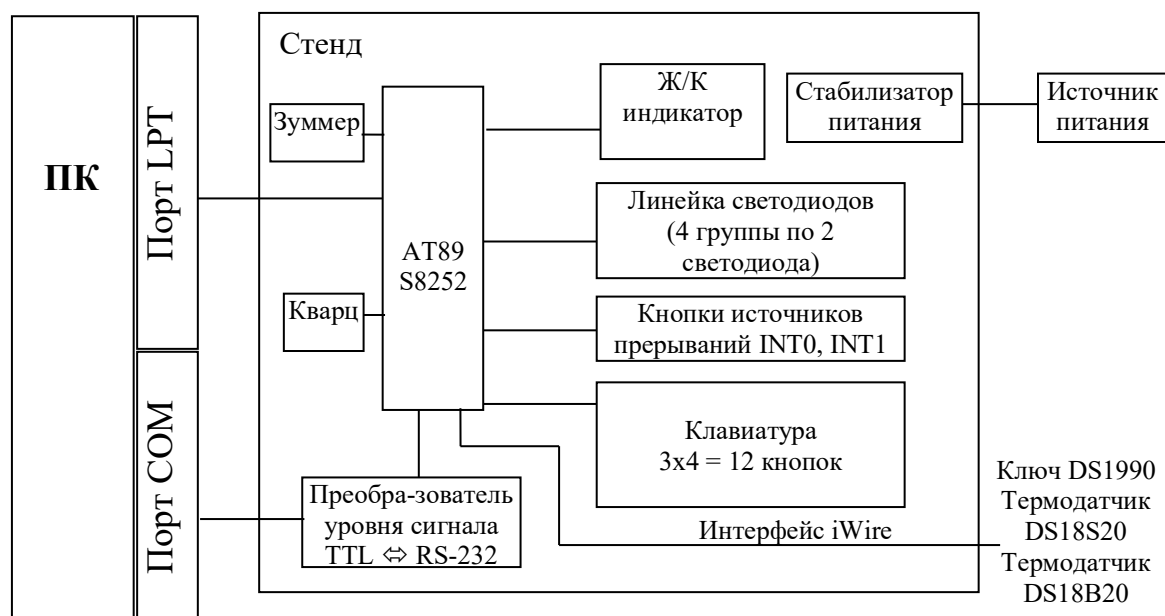


Рисунок 1 – Структурная схема лабораторной установки

1.1 Описание задачи

При включении питания на индикаторе отображаются записанные в энергонезависимой памяти номер программы и текущий период. Светодиоды должны включаться и выключаться в соответствии с текущей программой и периодом. При нажатии цифровой клавиши на клавиатуре устанавливается (и запоминается в EEPROM) текущая программа. Если программы с данным номером не существует, не должно быть никакой реакции. Клавиши "#" и "↵" модифицируют период. Допустимые значения периода: 0,1 – 1,0 с. Шаг изменения – 0,1 с. При изменении текущий период записывается в EEPROM.

Программы показаны в таблице 1.

Таблица 1 – Программы

Название	Порядок включения светодиодов (единица обозначает, что светодиод включен).
0. Случайный выбор	Через каждые несколько периодов случайно выбирается программа 1 - 8
1. Бегущая единица	1000 0100 0010 0001
2. Бегущая единица (обр.)	0001 0010 0100 1000
3. Чередование	0101 1010
4. Волна	0000 1000 1100 1110 1111 1110 1100 1000
5. Волна (обр.)	0000 0001 0011 0111 1111 0111 0011 0001
6. (Определяется самостоятельно)	1001 0110
7. (Определяется самостоятельно)	1100 0011
8. (Определяется самостоятельно)	1100 0110 0011

2 Модель управляющего автомата

Структурная схема системы управления представлена на рисунке 2.

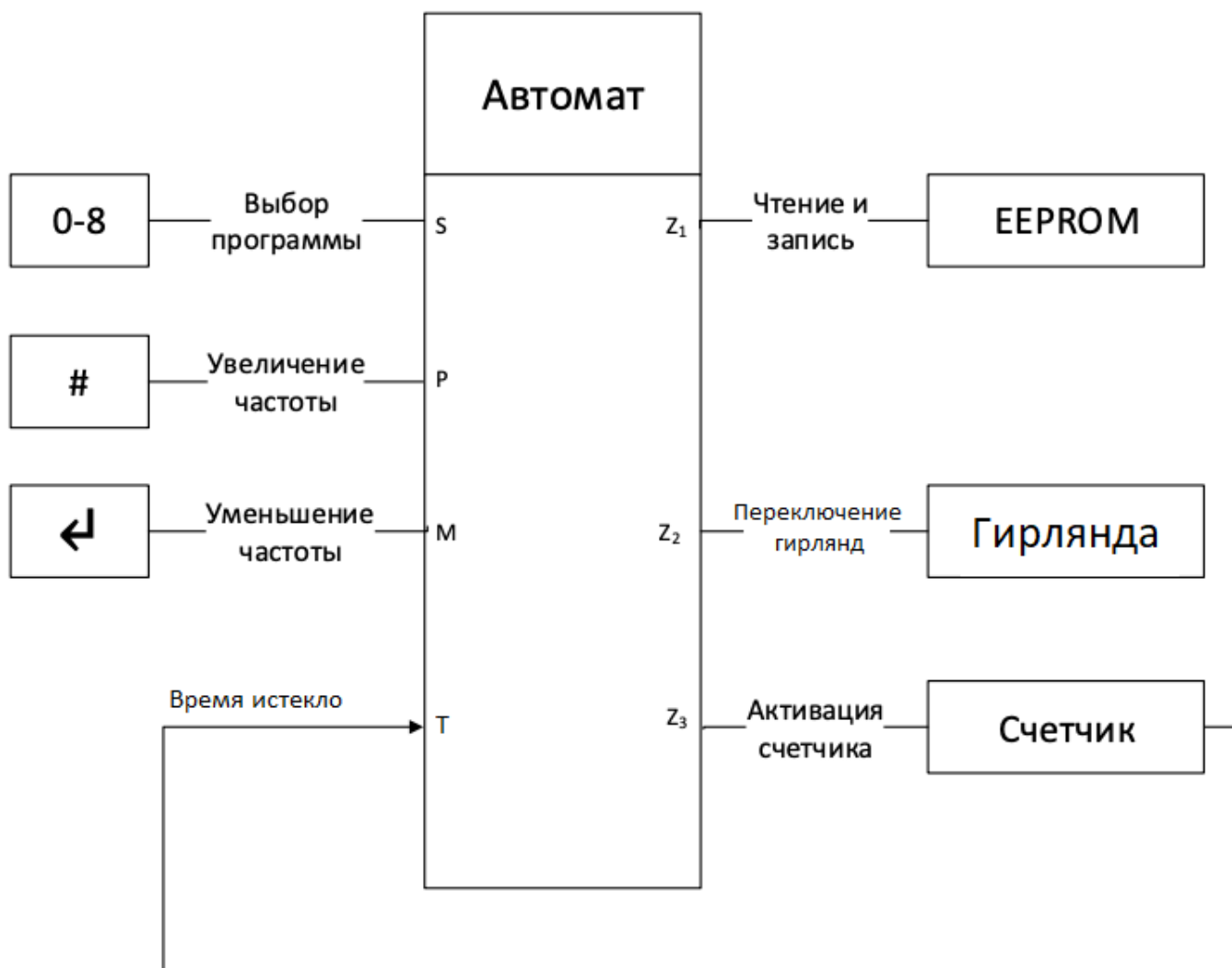


Рисунок 2 – Структурная схема системы управления

Переходы состояний и выходные сигналы представлены в таблице 2.

Таблица 2 – Переходы состояний и выходные сигналы

state	П 0	П 1	П 2	П 3	П 4	П 5	П 6	П 7	П 8
П 0	П 0	S ₁ /Z ₂	S ₂ /Z ₂	S ₃ /Z ₂	S ₄ /Z ₂	S ₅ /Z ₂	S ₆ /Z ₂	S ₇ /Z ₂	S ₈ /Z ₂
П 1	S ₀ /Z ₂	П 1	S ₂ /Z ₂	S ₃ /Z ₂	S ₄ /Z ₂	S ₅ /Z ₂	S ₆ /Z ₂	S ₇ /Z ₂	S ₈ /Z ₂
П 2	S ₀ /Z ₂	S ₁ /Z ₂	П 2	S ₃ /Z ₂	S ₄ /Z ₂	S ₅ /Z ₂	S ₆ /Z ₂	S ₇ /Z ₂	S ₈ /Z ₂
П 3	S ₀ /Z ₂	S ₁ /Z ₂	S ₂ /Z ₂	П 3	S ₄ /Z ₂	S ₅ /Z ₂	S ₆ /Z ₂	S ₇ /Z ₂	S ₈ /Z ₂
П 4	S ₀ /Z ₂	S ₁ /Z ₂	S ₂ /Z ₂	S ₃ /Z ₂	П 4	S ₅ /Z ₂	S ₆ /Z ₂	S ₇ /Z ₂	S ₈ /Z ₂
П 5	S ₀ /Z ₂	S ₁ /Z ₂	S ₂ /Z ₂	S ₃ /Z ₂	S ₄ /Z ₂	П 5	S ₆ /Z ₂	S ₇ /Z ₂	S ₈ /Z ₂
П 6	S ₀ /Z ₂	S ₁ /Z ₂	S ₂ /Z ₂	S ₃ /Z ₂	S ₄ /Z ₂	S ₅ /Z ₂	П 6	S ₇ /Z ₂	S ₈ /Z ₂
П 7	S ₀ /Z ₂	S ₁ /Z ₂	S ₂ /Z ₂	S ₃ /Z ₂	S ₄ /Z ₂	S ₅ /Z ₂	S ₆ /Z ₂	П 7	S ₈ /Z ₂
П 8	S ₀ /Z ₂	S ₁ /Z ₂	S ₂ /Z ₂	S ₃ /Z ₂	S ₄ /Z ₂	S ₅ /Z ₂	S ₆ /Z ₂	S ₇ /Z ₂	П 8

Принципиальная схема лабораторной установки представлена на рисунке 3.

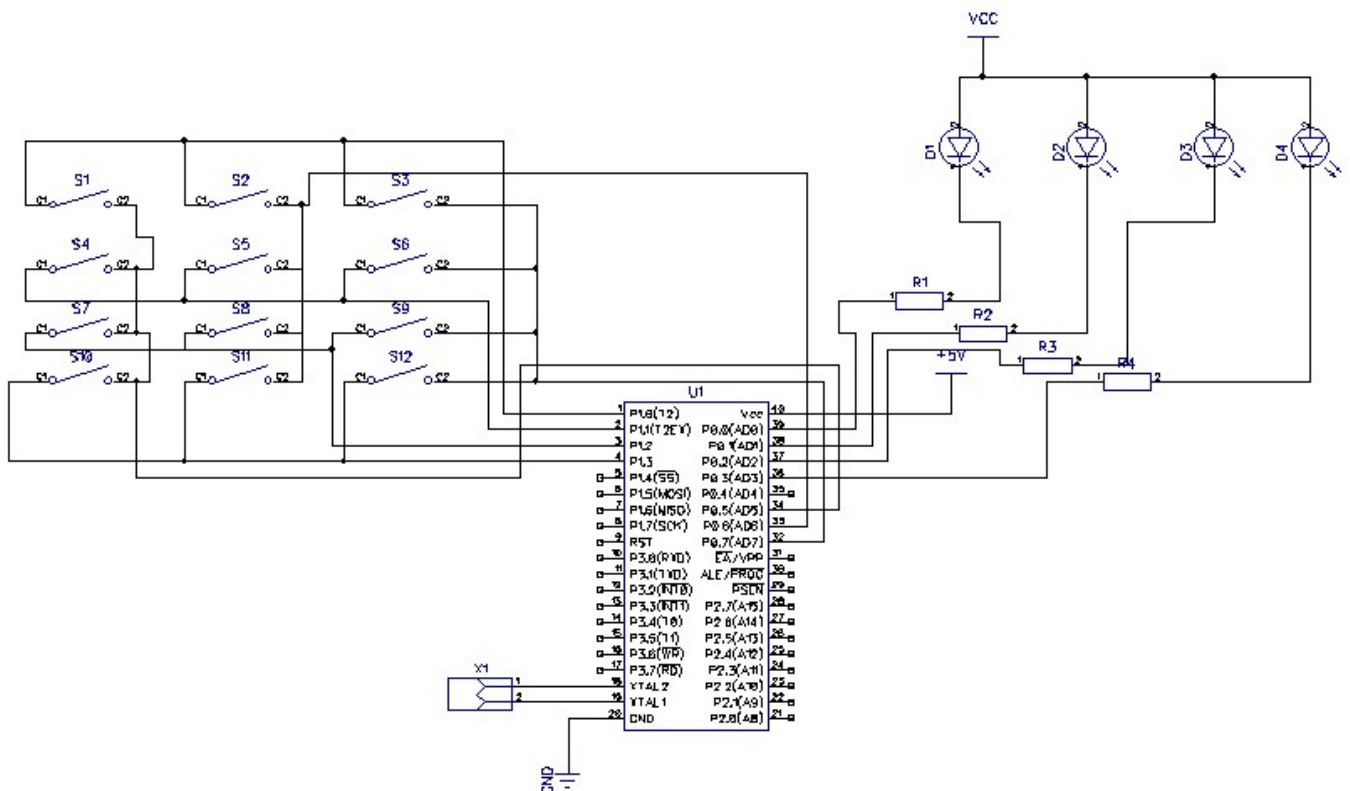


Рисунок 3 – Принципиальная схема лабораторной установки

Схема печатной платы представлена на рисунке 4.

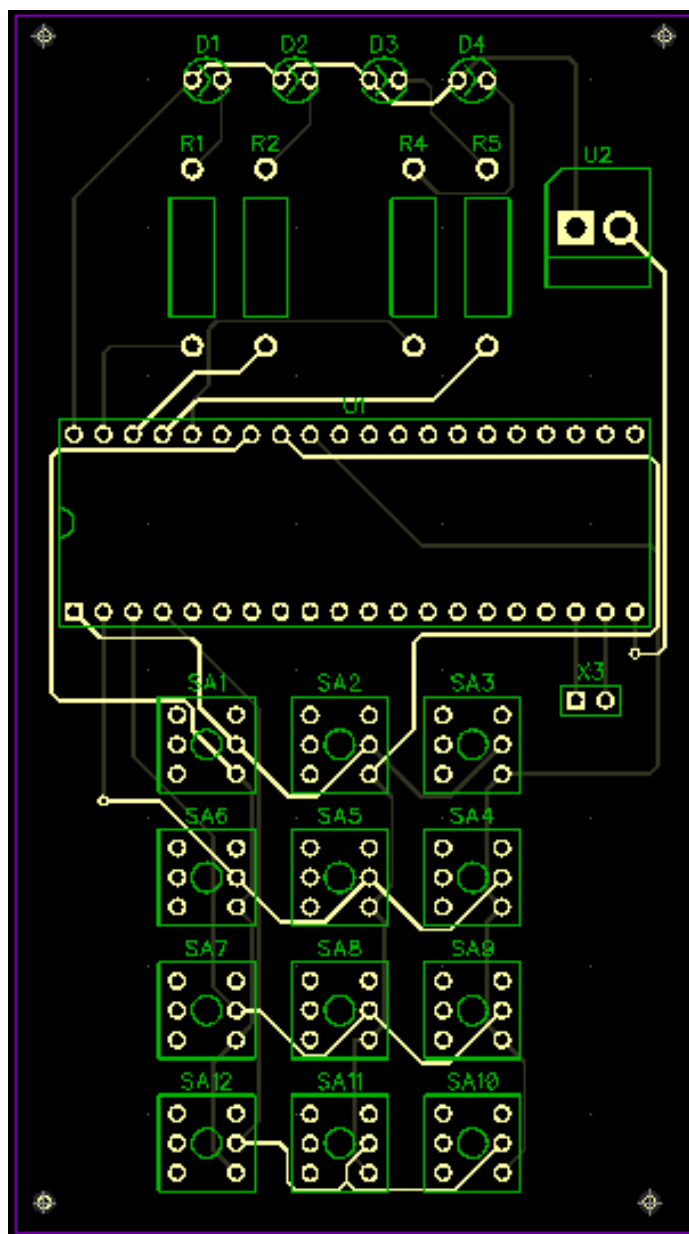


Рисунок 4 – Схема печатной платы

Корпус устройства представлена на рисунке 5.

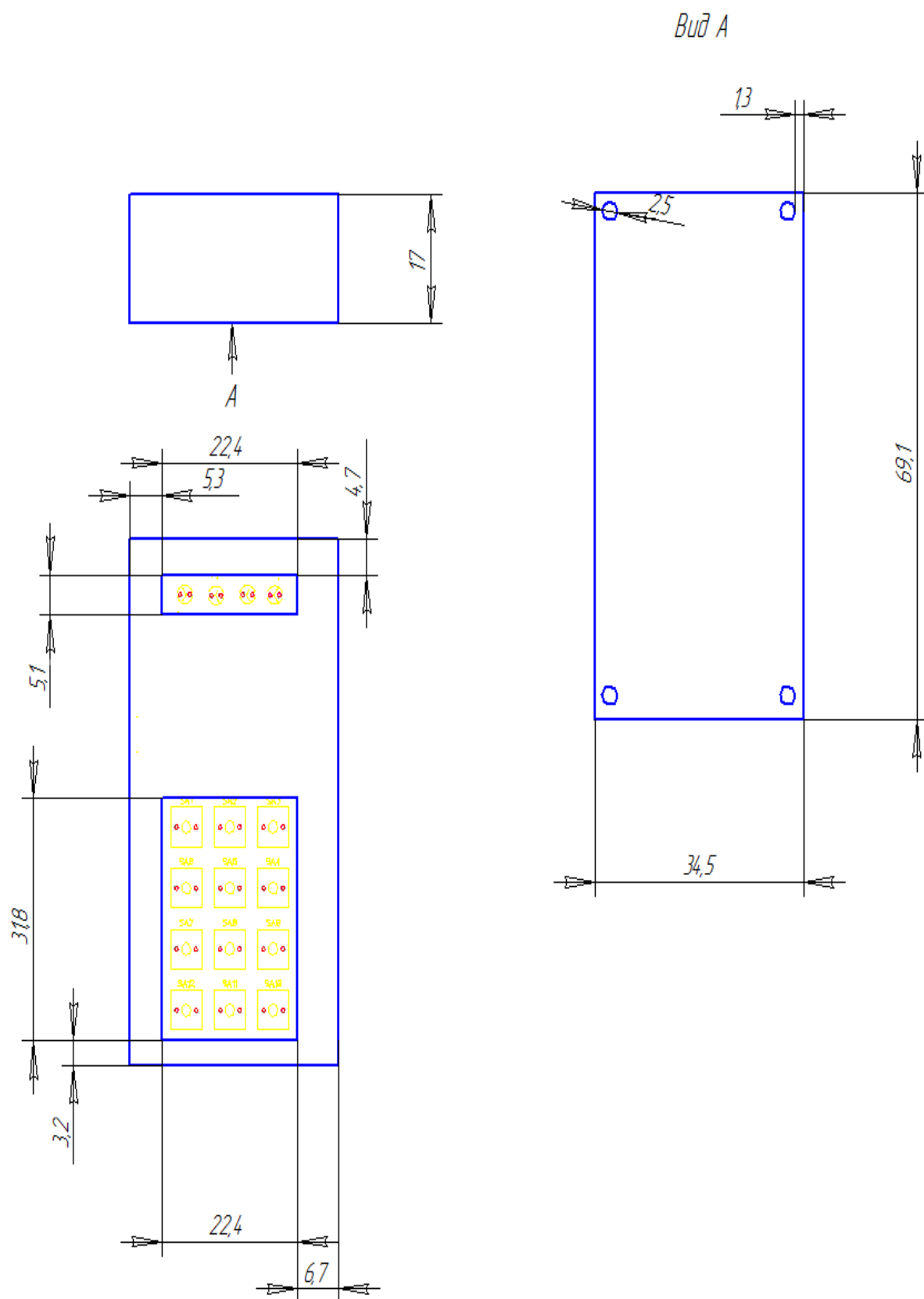


Рисунок 5 – Корпус устройства

Выводы

В результате лабораторной работы:

- а) изучены принципы построения микроконтроллерных систем на примере микроконтроллера AT89S8252;
- б) разработана модель автомата, управляющего гирляндой на 4 цвета;
- в) на базе этой модели разработана программа управления гирляндой на 4 цвета на языке C в среде разработки Micro C Pro for 8051;
- г) программа успешно скомпилирована, отлажена в симуляторе среды разработки и на лабораторной установке;
- д) изучены схемы, принцип работы и программирование микроконтроллера и периферийных устройств: клавиатура, светодиодная линейка, ЖК индикатор.

Проблемы, возникшие в результате работы, и их решения представлены в таблице 3.

Таблица 3 – Проблемы, возникшие в результате работы, и их решения

Проблема	Решение
Размер программы превышает 2 кбайт, ошибка «Demo limit»	Уменьшить количество состояний. Теперь в программе всего 9 состояний, каждое из которых обозначает одну из программ.
Не хватает памяти	Записать программы в виде двоичных кодов.

Приложение А

Листинг программы

```
#include "drv.h"

#define D1 P0_0_bit
#define D2 P0_1_bit
#define D3 P0_2_bit
#define D4 P0_3_bit

#define p0s0 0
#define p1s0 1
#define p2s0 2
#define p3s0 3
#define p4s0 4
#define p5s0 5
#define p6s0 6
#define p7s0 7
#define p8s0 8

#define key_up 13
#define key_down '#'

char ms = 0;
char state = 1;
char key = 0;
char t = 0;
unsigned int speed = 1000;
char T_FLAG = 0;
char zero_flag = 0;

char* prog1[] = { 8, 4, 2, 1 };
char* prog2[] = { 1, 2, 4, 8 };
char* prog3[] = { 5, 10 };
char* prog4[] = { 0, 8, 12, 14, 15, 14, 12, 8 };
char* prog5[] = { 0, 1, 3, 7, 15, 7, 3, 1 };
char* prog6[] = { 9, 6 };
char* prog7[] = { 12, 3 };
char* prog8[] = { 12, 6, 3 };

void DelayMs(unsigned int m) {
    unsigned char a;
    for (ms = 0; ms != m; ms++) {
        for (a = 0; a != 120; a++);
        WMCON.WDTRST = 1;
    }
}

int getRandProg() {
    return 1;
}

void changeProgram(unsigned char key) {
    if (key == 0) return;
    switch (key) {
        case '0':
            t = 2;
            zero_flag = 1;
    }
```

```

        break;
    case '1':
        state = p1s0;
        zero_flag = 0;
        break;
    case '2':
        state = 2;
        zero_flag = 0;
        break;
    case '3':
        state = 3;
        zero_flag = 0;
        break;
    case '4':
        state = 4;
        zero_flag = 0;
        break;
    case '5':
        state = 5;
        zero_flag = 0;
        break;
    case '6':
        state = 6;
        zero_flag = 0;
        break;
    case '7':
        state = 7;
        zero_flag = 0;
        break;
    case '8':
        state = 8;
        zero_flag = 0;
        break;
    }
}

void show(char* str[], int len, char speed) {
    char i = 0;
    char j = 8;
    char c = 0;
    while (i < len) {
        c = str[i];
        j = 8;
        while (j > 0) {
            if (j & c) {
                if (j == 1) D1 = 0;
                if (j == 2) D2 = 0;
                if (j == 4) D3 = 0;
                if (j == 8) D4 = 0;
            }
            else {
                if (j == 1) D1 = 1;
                if (j == 2) D2 = 1;
                if (j == 4) D3 = 1;
                if (j == 8) D4 = 1;
            }
            j >>= 1;
        }
    }
}

```

```

        delayMs(speed);
        i++;
    }
}

void main() {
    init();
    clear_lcd();

    while (1) {
        outd(state + '0');
        {
            int i;
            for (i = 1; i < 32; ++i) {
                outd(' ');
            }
        }
        key = ScanKbd();
        switch (state) {
            case p1s0:
                show(prog1, 4, speed);
                changeProgram(key);
                if (t > 0) t--;
                if ((t == 0) && (zero_flag)) {
                    t = 2;
                    state = getRandProg();
                }
                break;
            case p2s0:
                show(prog2, 4, speed);
                changeProgram(key);
                if (t > 0) t--;
                if ((t == 0) && (zero_flag)) {
                    t = 2;
                    state = getRandProg();
                }
                break;
            case p3s0:
                show(prog3, 2, speed);
                changeProgram(key);
                if (t > 0) t--;
                if ((t == 0) && (zero_flag)) {
                    t = 2;
                    state = getRandProg();
                }
                break;
            case p4s0:
                show(prog4, 8, speed);
                changeProgram(key);
                if (t > 0) t--;
                if ((t == 0) && (zero_flag)) {
                    t = 2;
                    state = getRandProg();
                }
                break;
            case p5s0:
                show(prog5, 8, speed);

```

```

    changeProgram(key);
    if (t > 0) t--;
    if ((t == 0) && (zero_flag)) {
        t = 2;
        state = getRandProg();
    }
    break;
case p6s0:
    show(prog6, 2, speed);
    changeProgram(key);
    if (t > 0) t--;
    if ((t == 0) && (zero_flag)) {
        t = 2;
        state = getRandProg();
    }
    break;
case p7s0:
    show(prog7, 2, speed);
    changeProgram(key);
    if (t > 0) t--;
    if ((t == 0) && (zero_flag)) {
        t = 2;
        state = getRandProg();
    }
    break;
case p8s0:
    show(prog8, 3, speed);
    changeProgram(key);
    if (t > 0) t--;
    if ((t == 0) && (zero_flag)) {
        t = 2;
        state = getRandProg();
    }
    break;
}
if ((key == key_up) && (speed > 100)) {
    speed -= 100;
}
if ((key == key_down) && (speed < 1000)) {
    speed += 100;
}
delayMs(speed);
}
}

```