Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

«Южно-Уральский государственный университет»

Факультет «Компьютерных технологий, управления и радиоэлектроники»

Кафедра «Электронные вычислительные машины»

Разработка игры для развития памяти на базе микроконтроллера

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К КУРСОВОЙ РАБОТЕ

по дисциплине: «Микропроцессорные системы»

Руководитель

В.А. Руднев

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2016 г.

Выполнил

студент группы КТУР-418

И. А. Ельцов

«\_\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2016 г.

Рекомендуемая оценка

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Челябинск, 2016

ОГЛАВЛЕНИЕ

[1 АНАЛИЗ РЕШАЕМОЙ ЗАДАЧИ 2](#_Toc450002773)

[1.1 Формулировка задания 2](#_Toc450002774)

[1.3 Обзор аналогов 2](#_Toc450002775)

[2 ТЕХНИЧЕСКОЕ ЗАДАНИЕ 3](#_Toc450002776)

[2.1 Требования к функционалу (с точки зрения пользователя) 3](#_Toc450002777)

[2.2 Требования к функционалу (с точки зрения разработчика) 3](#_Toc450002778)

[2.3 Конструктив (с точки зрения пользователя) 3](#_Toc450002779)

[2.4 Конструктив (с точки зрения разработчика) 3](#_Toc450002780)

[2.5 Требования к надежности 3](#_Toc450002781)

[2.6 Требования к безопасности 3](#_Toc450002782)

[2.7 Требования к программному обеспечению 3](#_Toc450002783)

[2.8 Требования к аппаратному обеспечению 4](#_Toc450002784)

[2.9 Стоимость 4](#_Toc450002785)

[3 ПРОЕКТИРОВАНИЕ 4](#_Toc450002786)

[3.1 Выбор элементной базы 4](#_Toc450002787)

[3.2 Структурное проектирование 5](#_Toc450002788)

[4. СИНТЕЗ ПРИНЦИПИАЛЬНОЙ СХЕМЫ 5](#_Toc450002789)

[5. РАЗВОДКА ПЕЧАТНОЙ ПЛАТЫ 5](#_Toc450002790)

[6 ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ МИКРОКОНТРОЛЛЕРА 7](#_Toc450002791)

[7 ПРОВЕРКА НА СООТВЕТСТВИЕ ТРЕБОВАНИЯМ ТЗ 7](#_Toc450002792)

[ВЫВОД 8](#_Toc450002793)

[СПИСОК ЛИТЕРАТУРЫ 9](#_Toc450002794)

[ПРИЛОЖЕНИЕ А 10](#_Toc450002795)

1 АНАЛИЗ РЕШАЕМОЙ ЗАДАЧИ

1.1 Формулировка задания

Разработать устройство для развития и тренировку памяти. Устройство подключается к ПК через интерфейс RS-232. Устройство демонстрирует игроку последовательность случайных цифр с помощью семисегментного индикатора, игрок впоследствии должен повторить набор цифр с помощью матричной клавиатуры. Если игрок ошибается, то на ПК показывается результат игры (величина численной последовательности), если нет – игра добавляет новое случайное число.

1.3 Обзор аналогов

В настоящее время существует множество аналогов данному устройству. Большинство таких устройств не имеет взаимодействия с компьютером. Рассмотрим одно из них – устройство «Саймон», внешний вид которого изображен на рисунке 1.



*Рис. 1.* Электронная игра «Саймон»

Устройство представляет собой классическую одноименную игру. Внешне игра представляет собой круглый диск с четырьмя большими разноцветными кнопками-панелями, издававшими разные звуки. Задача игры — запомнить продемонстрированную игроку последовательность этих вспышек и звуков (начать игру можно было с нажатия любой кнопки, при этом игра была задумана так, чтобы мелодия звучала гармонично независимо от того, какая кнопка была нажата первой) и впоследствии воспроизвести её в рамках «раунда» игры. Игра реализована на базе микропроцессора. Российская стоимость игры составляет 2299 рублей.

Очевидными недостатками этого модуля являются:

* отсутствие возможности подключения к ПК;
* малое количество клавиш;
* отсутствие беззвучного режима;
* высокая стоимость.

К основным недостаткам можно также отнести:

* высокую стоимость;
* невозможность подключения к ПК.

Таким образом, можно сделать вывод о необходимости разработки собственного аналога существующим устройствам с добавлением необходимого функционала.

2 ТЕХНИЧЕСКОЕ ЗАДАНИЕ

2.1 Требования к функционалу (с точки зрения пользователя)

Разрабатываемое устройство и программное обеспечение должно предоставлять возможность видеть последовательность чисел, повторять это значение с помощью клавиатуры и наблюдать результат игры с ПК.

2.2 Требования к функционалу (с точки зрения разработчика)

Отправка данных на ПК осуществляется через интерфейс RS-232. Программное обеспечение устройства должно корректно отправлять данные на ПК, считывать значения с матричной клавиатуры и выводить последовательность цифр на семисегементный индикатор.

2.3 Конструктив (с точки зрения пользователя)

Устройство в пластиковом корпусе с разъёмом DE-9, матричной клавиатурой 3х3 и семисегментным индикатором.

2.4 Конструктив (с точки зрения разработчика)

Модуль состоит из следующих компонентов, смонтированных на печатной плате:

* Микроконтроллер;
* Разъем DE-9 для подключения к компьютеру;
* Микросхема MAX-232;
* Матричная клавиатура 3х3;
* Семисегментный индикатор;
* Разъем SPI для программирования микроконтроллера.

2.5 Требования к надежности

Отключения питания устройства не должно приводить его в негодность. Так как устройство предназначено для бытового применения, особых требований к надежности не предъявляется.

2.6 Требования к безопасности

Так как устройство предназначено для бытового применения, особых требований к безопасности не предъявляется.

2.7 Требования к программному обеспечению

Программное обеспечение контроллера должно корректно отправлять результат игры на компьютер по интерфейсу RS-232, генерировать случайную последовательность чисел с выводом на семисегментный индикатор и корректно обрабатывать нажатия клавиш матричной клавиатуры 3х3.

2.8 Требования к аппаратному обеспечению

Устройство должно быть выполнено на базе микроконтроллера фирмы Atmel. Микроконтроллер должен иметь достаточное число ножек для подключения датчиков, иметь объем памяти программ, достаточный для размещения управляющей программы и объем энергонезависимой памяти для хранения результата работы случайного генератора чисел. Также, устройство должно иметь возможность подключения программатора, для изменения программы микроконтроллера.

2.9 Стоимость

Общая стоимость конечного устройства не должна превышать 2000 рублей.

3 ПРОЕКТИРОВАНИЕ

3.1 Выбор элементной базы

Устройство работает на базе микроконтроллера ATmega16, т.к. он:

1. обладает достаточным объемом памяти;
2. TQFP корпус удобен для монтажа на печатную плату и подключения к нему множества элементов;
3. имеет аппаратную реализацию интерфейсов SPI и UART.

Микросхема MAX-232 необходима для преобразования сигналов последовательного порта RS-232 в сигналы, пригодные для микроконтроллера.

Семисегментый индикатор SC04-12YWA для отображения процесса игры.

Линейный стабилизатор LM7805 на 5 Вольт для преобразования напряжения элемента питания.

Выпрямительный диод 1N4001 для подключения элемента питания.

Кнопки 8551MAVBE2 для реализации матричной клавиатуры.

Резисторы:

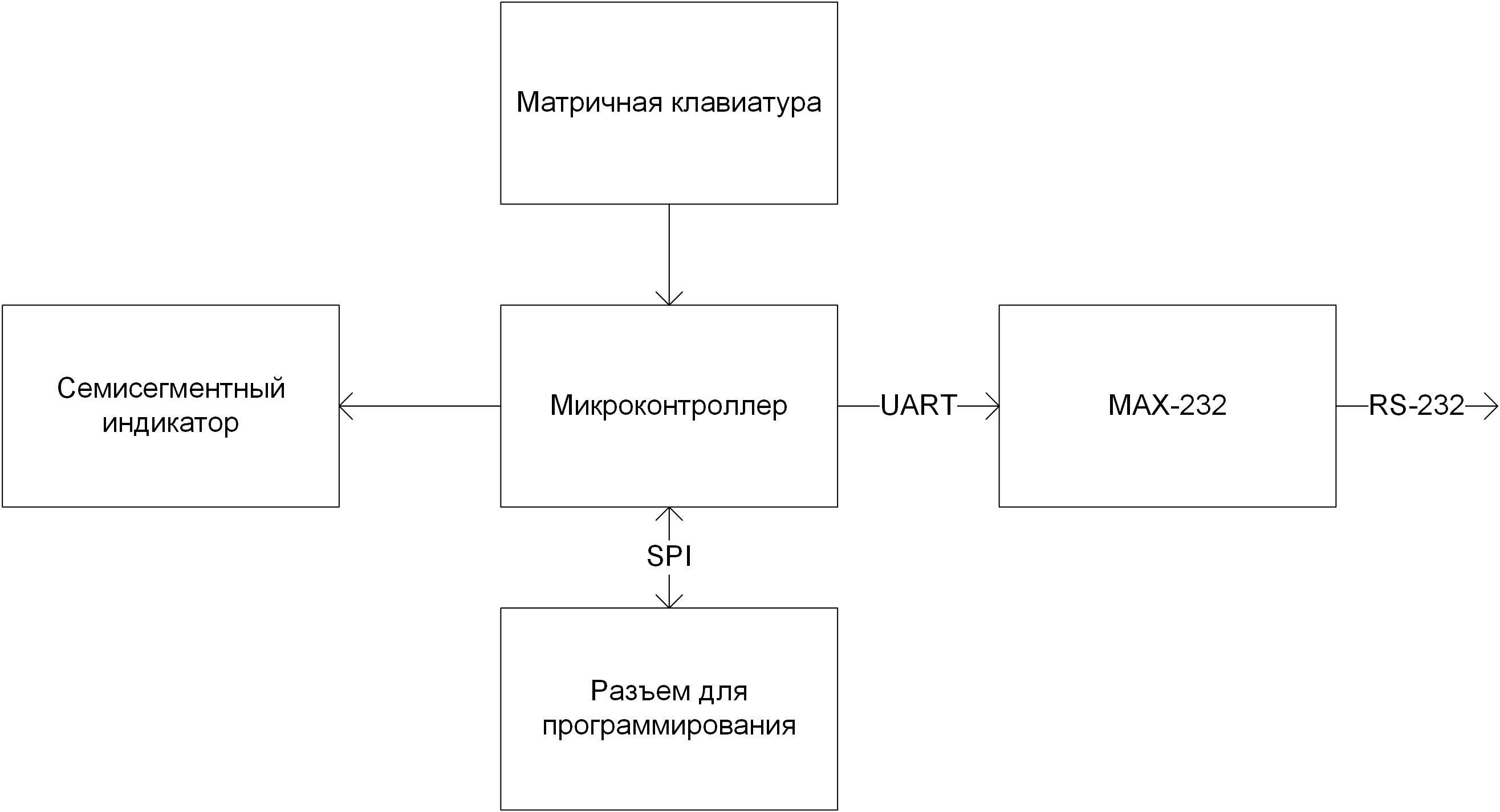
* 100 Ом – токоограничивающие резисторы для семисегментного индикатора.

Конденсаторы:

* 1 мкФ – типовые электролитические конденсаторы для подключения микросхемы MAX-232;
* 0,33/0,1 мкФ – типовое подключение стабилизатора LM7805.
* 10 мкФ – электролитический конденсатор для подключения элемента питания.

3.2 Структурное проектирование

На рисунке 2 представлена структурная схема соединения ключевых узлов модуля.



*Рис. 2.* Структурная схема

К микроконтроллеру подключена микросхема MAX-232, позволяющая преобразовывать сигналы, отправляемые от микроконтроллера к компьютеру по интерфейсу RS-232. Программирование осуществляется через SPI-разъем. Микроконтроллер напрямую посылает сгенерированный сигнал на вход семисегментного индикатора и принимает сигналы с матричной клавиатуры.

4. СИНТЕЗ ПРИНЦИПИАЛЬНОЙ СХЕМЫ

DD1 – микроконтроллер ATmega16.

DD2 – микросхема MAX-232 для преобразования логических уровней.

XS1 – разъем для подключения к устройству по интерфейсу RS-232.

XS2 – разъем IDC 10p для подключения программатора.

XS3 – разъем для подключения элемента питания 9В.

DA1 – микросхема преобразования напряжения на 5В.

R1 – резистор для подключения пьезоизлучателя к микроконтроллеру.

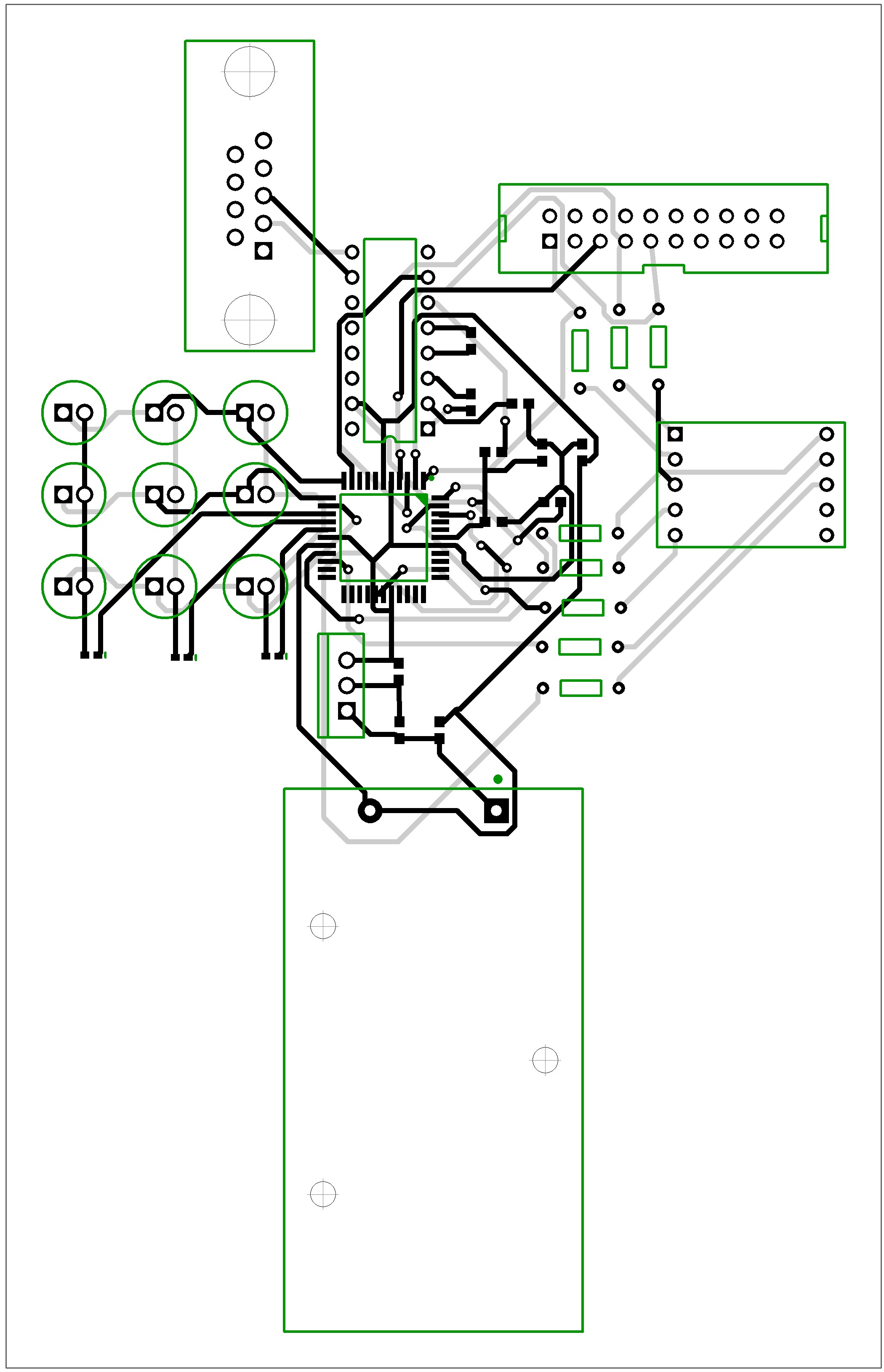
С1 – конденсатор для подключения элемента питания.

С2-С3 – конденсаторы схемы типового подключения стабилизатора напряжения.

С4-C7 – конденсаторы схемы типового подключения микросхемы MAX-232 к микроконтроллеру.

5. РАЗВОДКА ПЕЧАТНОЙ ПЛАТЫ

Спроектированная плата является двусторонней, в центре располагается микроконтроллер и микросхема MAX-232, в правой части расположен пьезоилучатель, в левой части разъемы IDC и DE-9. В нижней части платы располагается разъем для элемента питания. Печатная плата изображена на рисунке 3.



*Рис. 3.* Лицевая сторона печатной платы

6 ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ МИКРОКОНТРОЛЛЕРА

Для демонстрации результата на ПК в микроконтроллере используется аппаратная реализация интерфейса UART со следующими параметрами: скорость 9600 бод, размер пакета – 8 бит, 1 стоп-бит, бит четности – none. Отправка данных выполняется с помощью функции sendByte.

Для реализации генерации случайных чисел используется энергонезависимая память и функции rand и srand из библиотеки stdlib, а в качестве источника энтропии было определено время задержки нажатия клавиш пользователя.

При включении микроконтроллера инициализируется начальные переменные. При окончании игры передается сигнала на ПК с результатом.

7 ПРОВЕРКА НА СООТВЕТСТВИЕ ТРЕБОВАНИЯМ ТЗ

1) Функциональные требования.

Спроектированное устройство работает на базе микроконтроллера, и имеет разъем DE-9 для подключения к компьютеру. Устройство демонстрирует случайные числа, реагирует на нажатие клавиш и передает результат игры на ПК. Можно считать, что устройство удовлетворяет функциональным требованиям ТЗ.

2) Конструктивные требования.

Прибор имеет все перечисленные ключевые компоненты и разъемы (микроконтроллер, матричная клавиатура, семисегментный индикатор, разъем RS-232, разъем ISP) и исполнено в пластиковом корпусе – устройство удовлетворяет конструктивным требованиям ТЗ.

3) Требования к надежности.

Отключение питания не приводит в негодность устройство – устройство удовлетворяет требованиям надежности ТЗ.

4) Требования к безопасности.

Требований к безопасности не предъявлялось.

5) Требования к программному обеспечению.

Устройство работает на базе микроконтроллера ATmega16, смонтировано на печатной плате, оснащено интерфейсом RS-232 и обладает достаточным размером памяти.

6) Требования к аппаратному обеспечению.

Устройство собрано на базе микроконтроллера ATmega16. Также устройство имеет разъем для подключения программатора. Требования к аппаратному обеспечению выполняются.

7) Требования к стоимости.

Расчет стоимости устройства приведен ниже. Общая стоимость полученного устройства не превышает 2000 рублей, а значит, требования к стоимости устройства удовлетворены.

Таблица 1. Стоимость компонентов.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Номер | Название элемента | Цена (р.) | Кол-во | Стоимость |
| 1 | ATMEGA162-16AU | 194.35 | 1 | 194.35 |
| 2 | MAX232 | 48.75 | 1 | 48.75 |
| 3 | SC04-12YWA | 15.00 | 1 | 15.00 |
| 4 | CA050M1R00REB-0405 | 1.12 | 4 | 4.48 |
| 5 | C2012X7R1H104K | 1.80 | 1 | 1.80 |
| 6 | CC0805KKX7R7BB334 | 1.82 | 1 | 1.82 |
| 7 | CE016M0010REB-0405 | 1.57 | 1 | 1.57 |
| 8 | CF-25 (С1-4) | 0.80 | 7 | 5.60 |
| 9 | DBS-9F Z+H | 28.40 | 1 | 28.40 |
| 10 | BH9VA | 25.00 | 1 | 25.00 |
| 11 | L7805CV | 16.20 | 1 | 16.20 |
| 12 | 1N4001 | 2.25 | 5 | 11.25 |
| 13 | IDC-10M | 23.87 | 1 | 23.87 |
| 14 | IT-1102WB | 0.91 | 9 | 8.19 |
|  |  |  | Сумма | 386.30 |

Суммарная стоимость компонентов не превышает ограничение, выставленное в ТЗ – модуль удовлетворяет требованиям к стоимости ТЗ.

ВЫВОД

В результате выполнения данной работы было спроектировано устройство для развития памяти.

Разработанный девайс подключается к компьютеру по интерфейсу RS-232. Устройство принимает номер от ПК и воспроизводит соответствующую композицию. Для генерации случайных чисел используется энергонезависимая память.

Устройство удовлетворяет всем заявленным требованиям технического задания. Работу над устройством можно считать завершенной успешно.

СПИСОК ЛИТЕРАТУРЫ

1. Simon Electronic Game, 1978 [Электронный ресурс]. URL: http://americanhistory.si.edu/collections/search/object/nmah\_1302005
2. 8-bit Microcontroller with 16K Bytes In-System Programmable Flash (ATmega16 Datasheet). [Электронный ресурс]. URL: http://www.atmel.com/images/2466s.pdf
3. AVR. Учебный курс. Использование ШИМ. [Электронный ресурс]. URL: http://easyelectronics.ru/avr-uchebnyj-kurs-ispolzovanie-shim.html
4. Белов, А.В. Микроконтроллеры AVR: от азов программирования до создания практических устройств. Книга + CD с видеокурсами, листин гами, программами, драйверами, справочниками. / А.В. Белов **–** СПб.: Наука и Техника, 2016. **–** 544 с.
5. AVR. Учебный курс. Передача данных через UART. [Электронный ресурс]. URL: http://easyelectronics.ru/avr-uchebnyj-kurs-peredacha-dannyx-cherez-uart.html.
6. Группа компаний промэлектроника. [Электронный ресурс]. URL: http://www.promelec.ru/shop/main/

ПРИЛОЖЕНИЕ А

Исходный код программной части устройства

Исходный код файла main.c

#define F\_CPU 8000000UL

#include <avr/io.h>

#include <avr/eeprom.h>

#include <avr/interrupt.h>

#include <util/delay.h>

#include <stdlib.h>

#include <string.h>

//Задержка между миганиями значенями на семисегментном индикаторе

#define DELAY 400

uint8\_t display[3][3] = {{0xF9,0x99,0xF8}, //1 4 7

{0xA4,0x92,0x80}, //2 5 8

{0xB0,0x82,0x90}}; //3 6 9

int makeRandom(uint16\_t upper)

{

return (uint16\_t)((double)rand() / ((double)RAND\_MAX + 1) \* upper);

}

uint8\_t toDisplayNum(uint8\_t num)

{

uint8\_t i,j;

i = num / 3;

j = num % 3;

return display[i][j];

}

void comboPort(uint8\_t num)

{

PORTC = num & 0b00000011;

PORTB = num & 0b11111100;

}

void initUART()

{

//UBRR=f/(16\*band)-1 f=8000000Гц band=9600,

UBRRH=0;

UBRRL=51;

UCSRA=0x00;

//Разрешение работы передатчика

UCSRB=(1<<TXEN);

//Установка формата посылки: 8 бит данных, 1 стоп-бит

UCSRC=(1<<URSEL)|(1<<UCSZ1)|(1<<UCSZ0);

}

void sendByte(uint8\_t b)

{

//Устанавливается, когда регистр свободен

while(!(UCSRA & (1 << UDRE)));

UDR = b;

}

void sendStr(uint8\_t \*s)

{

while (\*s) sendByte(\*s++);

}

void sendNum(uint8\_t num)

{

if (num > 99) sendByte(num / 100 + '0');

if (num > 9) sendByte((num % 100) / 10 + '0');

sendByte(num % 10 + '0');

}

int main(void)

{

uint8\_t i, j, randomNumber, tempNumber, buttonCount, n = 0;

uint8\_t portState[3] = {0xDF,0xBF,0x7F};

uint8\_t inputState[3] = {0x04,0x08,0x10};

uint8\_t\* arr;

uint8\_t\* backup\_arr;

//Источник энтропии - счетчик задержки нажатия клавиш

uint16\_t delayCounter = eeprom\_read\_word((uint16\_t\*)1);

DDRC = 0xFF;

DDRB = 0xFF;

DDRD = 0xE3;

PORTD = 0xFF;

initUART();

while (1)

{

arr = (uint8\_t \*)malloc((n + 1) \* sizeof(uint8\_t));

if (n > 0) {

memcpy(arr, backup\_arr, n);

free(backup\_arr);

}

tempNumber = eeprom\_read\_byte((uint8\_t\*)0);

srand(delayCounter);

tempNumber ^= (makeRandom(1000)%255);

eeprom\_write\_byte((uint8\_t\*)0, (uint8\_t)(tempNumber % 8));

randomNumber = eeprom\_read\_byte((uint8\_t\*)0);

arr[n] = toDisplayNum(randomNumber);

for (i = 0; i < n + 1; i++)

{

comboPort(arr[i]);

\_delay\_ms(DELAY);

comboPort(0x00);

\_delay\_ms(DELAY);

}

buttonCount = 0;

while(buttonCount < n + 1)

{

for(i = 0; i < 3; i++)

{

PORTD = portState[i];

for(j = 0; j < 3; j++)

{

if(!(PIND & inputState[j]))

{

while((PIND & inputState[j]) != inputState[j]);

if(display[i][j] == arr[buttonCount]){

comboPort(display[i][j]);

buttonCount++;

} else {

sendStr("Game over! Your score: ");

sendNum(n);

sendByte(13);

//Мигаем GG

while(1)

{

comboPort(0xC2);

\_delay\_ms(DELAY);

comboPort(0x00);

\_delay\_ms(DELAY);

}

}

}

delayCounter++;

}

}

}

backup\_arr = (uint8\_t\*)malloc((n + 1) \* sizeof(uint8\_t));

memcpy(backup\_arr, arr, n + 1);

free(arr);

\_delay\_ms(DELAY);

comboPort(0x00);

\_delay\_ms(DELAY \* 2);

eeprom\_write\_word((uint16\_t\*)1, (uint16\_t)(delayCounter));

n++;

}

}