

Министерство науки и высшего образования и науки Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение высшего  
образования



**«Московский государственный технический  
университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)**

---

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ  
КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К КУРСОВОЙ  
РАБОТЕ НА ТЕМУ:**

**«Синтезатор звуковых сигналов»  
по дисциплине «Микропроцессорные системы»**

Студент гр. ИУ6-72

\_\_\_\_\_  
(Подпись, дата)

А.А. Жидков  
(И.О.Фамилия)

Руководитель курсовой работы  
к.т.н., доц.

\_\_\_\_\_  
(Подпись, дата)

В.Я. Хартов  
(И.О.Фамилия)

Москва, 2019

**ЛИСТ**  
**ЗАДАНИЯ**

## РЕФЕРАТ

Расчетно-пояснительная записка 48 с., 28 рис., 13 ист., 2 прил.

МИКРОКОНТРОЛЛЕР, ATMEGA8515, ИНТЕРФЕЙС RS232,  
ВОСПРОИЗВЕДЕНИЕ ЗВУКА.

Объектом проектирования является разработка регистратора времени.

Цель работы:

- закрепление знаний, полученных при изучении дисциплины «Микропроцессорные системы», в процессе самостоятельной работы при синтезатора звуковых сигналов;
- развитие навыков и умений применять теоретические знания на практике при выполнении учебных проектов, а также по заказам промышленности и в порядке личной инициативы;
- освоение новых технологий проектирования при выполнении проектных работ.

Задачи, решаемые в процессе проектирования:

- анализ задания;
- разработка структурно-функциональной схемы системы;
- выбор радиоэлементов схемы;
- расчет потребляемой мощности устройства;
- разработка алгоритмов и соответствующей программы работы системы.

# ОГЛАВЛЕНИЕ

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ, СОКРАЩЕНИЯ.....	5
ВВЕДЕНИЕ.....	6
1 Анализ технического задания.....	7
1.1 Анализ требований.....	7
1.2 Описание функциональных элементов системы.....	9
1.2.1 Описание микроконтроллера ATmega8515.....	9
1.4.2. Клавиатура.....	11
1.4.3 Описание драйвера для взаимодействия с ПЭВМ.....	11
1.4.4 Описание работы функциональной схемы.....	13
2 Описание используемых элементов.....	15
2.1 Корпус ATmega8515 и его особенности.....	15
2.2 Динамик и выбор кнопок.....	17
2.3 Питание схемы.....	17
2.4 Основные элементы принципиальной схемы.....	19
3 Расчёт потребляемой мощности системы.....	20
4 Описание алгоритмов системы.....	21
5 Моделирование и отладка системы.....	35
5 Способ программирования МК.....	42
ЗАКЛЮЧЕНИЕ.....	45
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	46
ПРИЛОЖЕНИЕ А. Листинг программы	
ПРИЛОЖЕНИЕ Б. Спецификация радиоэлементов схемы	

## **ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ, СОКРАЩЕНИЯ**

ТЗ – техническое задание

МК – микроконтроллер

ATmega8515 – используемый микроконтроллер

UART – схема последовательного интерфейса передачи данных

SPI – последовательный периферийный интерфейс

Proteus ISIS – среда моделирования

MEL – режим работы МК в воспроизведении мелодии

SYNT – режим работы МК в воспроизведении нот

## **ВВЕДЕНИЕ**

На основании учебного плана кафедры ИУ6 поставлена задача разработать синтезатор звуковых сигналов. Устройство должно воспроизводить мелодии из FLASH памяти и отправленные устройству через терминал ПВЭМ.

Разработка состоит из трёх основных частей:

- конструкторской части, состоящей из анализа требований, проектирования структурно-функциональной схемы, разработки функциональной и принципиальной схем;
- описания алгоритмов работы системы и расчета потребляемой мощности;
- части с описанием симуляции системы и её тестирования.

# **1 Анализ технического задания**

## **1.1 Анализ требований**

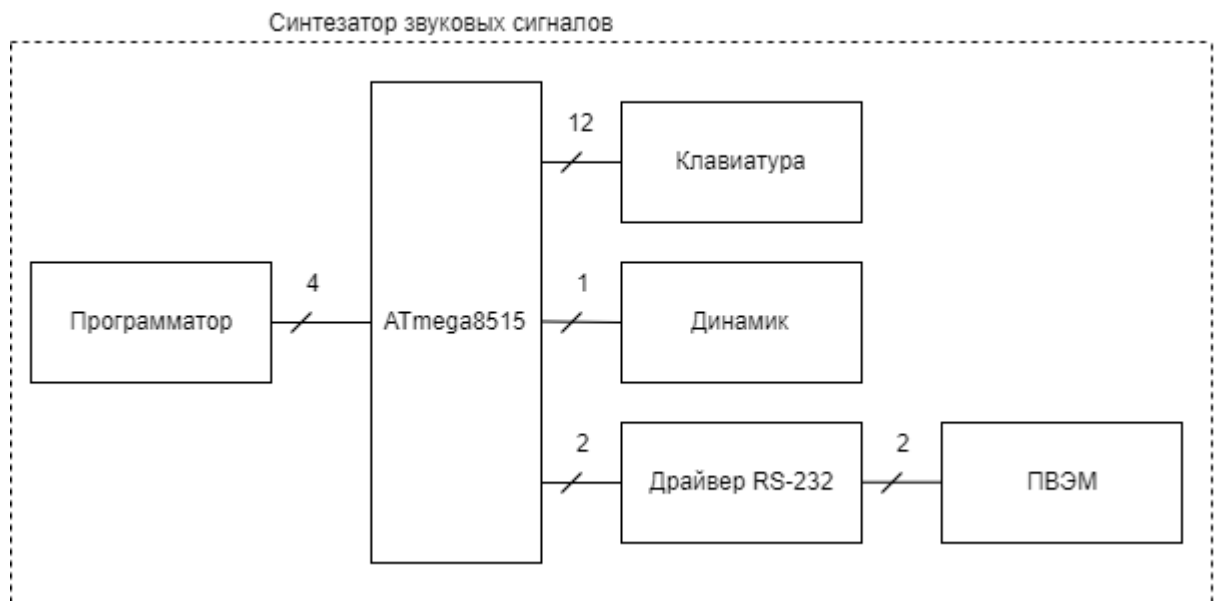
Согласно техническому заданию, необходимо разработать синтезатор звуковых сигналов. Воспроизведение звуковых сигналов будет осуществляться через динамик подключенный к одному из выходов МК.

Необходимо предусмотреть начальную установку кодов мелодий и передачу кодов мелодий с ПЭВМ по интерфейсу RS-232.

Исходя из вышеописанного разрабатываемая система должна представлять из себя аппаратно-программный модуль, содержащий по техническому заданию микроконтроллер семейства AVR ATmega8515.

В ходе работы было принято решение расширить функционал работы устройства добавив режим работы устройства, в котором у оператора будет возможность воспроизводить отдельные звуки с частотой соответствующей определенным нотам.

Помимо микроконтроллера система должна включать в себе: динамик, необходимый для воспроизведения звука; интерфейс RS-232, для передачи кодов мелодий с ПЭВМ; клавиатура, необходимая для проигрывания мелодий и для воспроизведения отдельных нот; кнопки для навигации и настройки режимов работы синтезатора, программатор, для программирования микроконтроллера. Структурно-функциональная схема МК-системы приведена на рисунке 1.



*Рисунок 1 – Структурно-функциональная схема*

МК находится в ожидании замыкания одной из кнопок. В зависимости от того, какая кнопка была нажата МК воспроизводит мелодию или звук с частотой соответствующей ноты, меняет режим работы, меняет темп или октаву воспроизводимой мелодии или ноты.

В результате анализа требований к МК-системе, можно выявить перечень блоков, которые необходимы для реализации устройства, соответствующего требованиям, указанным в техническом задании.

Структурно – функциональные элементы схемы:

- драйвер связи с ПЭВМ MAX232;
- клавиатура;
- динамик;
- светодиоды;
- программатор;
- микроконтроллер ATmega8515.



## **1.2 Описание функциональных элементов системы**

### **1.2.1 Описание микроконтроллера ATmega8515**

При разработке МК-системы был выбран микроконтроллер ATmega8515. Структурная схема микроконтроллера ATmega8515 приведена на рисунке 2.

Основной задачей программирования МК является управление модулями ядра AVR при помощи установки и сброса соответствующих битов в управляющих регистрах.

Для обмена сообщениями с ПЭВМ используется модуль USART, работа которого зависит от регистров UCSRA, UCSRB, UCSRC; для настройки скорости USART используется 16-разрядный регистр UBRR, а для приема-передачи данных – регистр UDR.

Для настройки портов ввода/вывода имеются 3 регистра: DDRx для установки направления линий (ввод/вывод), PORTx для настройки подтягивающих резисторов или значения на выходе МК, а также PINx для считывания значения на входе МК.

Устройство должно формировать звук, для этого используется 16-разрядный таймер T1, который настраивается на сброс по совпадению. Для настройки таймера T1 используются следующие управляющие регистры: TCCR1A, TCCR1B, TCNT1, TIMSK.



Кнопки для воспроизведения мелодий называются «C0»(до), «D»(ре), «E»(ми), «F»(фа), «G»(соль), «A»(ля), «B»(си), «C1»(до). Замыкая их будет проигрываться мелодия или звук с частотой соответствующей ноты. Кнопки «MEL» и «SYNT» отвечают за выбор режима в котором будет работать МК. При нажатии на кнопку «SYNT», МК будет работать в режиме синтезатора, в котором 8 кнопок используются для игры на инструменте. При нажатии на кнопку «MEL», МК будет работать в режиме воспроизведения мелодий, в котором каждая из 8 кнопок будет отвечать за воспроизведение определенной мелодии, хранящейся в памяти МК. Кнопки «+» и «-» отвечают за понижение или повышение октавы 8 нот воспроизводимых при нажатии на кнопки, или за повышение или понижение темпа проигрываемых мелодий. Действия МК будут зависеть от режима в котором он находится.

Для сброса МК используется кнопка «RESET».

Для упрощения навигации оператора в устройстве были предусмотрены 4 светодиода. Светодиоды «MEL» и «SYNT» показывают текущий режим в котором находится МК, а «+» и «-» будут загораться при нажатии на кнопки «+» и «-».

### **1.4.3 Описание драйвера для взаимодействия с ПЭВМ**

Как известно, коммуникационный последовательный интерфейс RS-232 работает в диапазоне напряжений (от -15В до +15В), которые не совместимы с логическими уровнями современных компьютеров.

С другой стороны, традиционная TTL логика компьютера работает между 0В...+ 5В. В современных схемах с малым энергопотреблением логические уровни могут быть в диапазоне от 0В и до + 3,3В или даже ниже. Уровни сигналов RS-232 слишком высоки для электронной логики современных компьютеров поэтому, чтобы получать последовательные

данные с интерфейса RS-232, напряжение должно быть понижено и инвертировано для лог. 0 и лог. 1.

Для согласования сигнала RS-232 с логикой внешних устройств зачастую используют интегральную микросхему из семейства MAX232. В качестве драйвера связи с ПЭВМ, а именно согласования интерфейсов (логических уровней сигналов) UART со стороны МК и RS232 со стороны ПЭВМ, был выбран драйвер MAX232, внутренняя схема которого представлена на рисунке 3.

Основные технические характеристики:

- корпус: SOIC-16-3.9;
- количество передатчиков: 2;
- количество приёмников: 2;
- максимальная скорость передачи данных: 150 кбит/с;
- напряжение питания: 5 В;
- особенности: 0.1 мкФ;
- рабочая температура: 0...70°C.

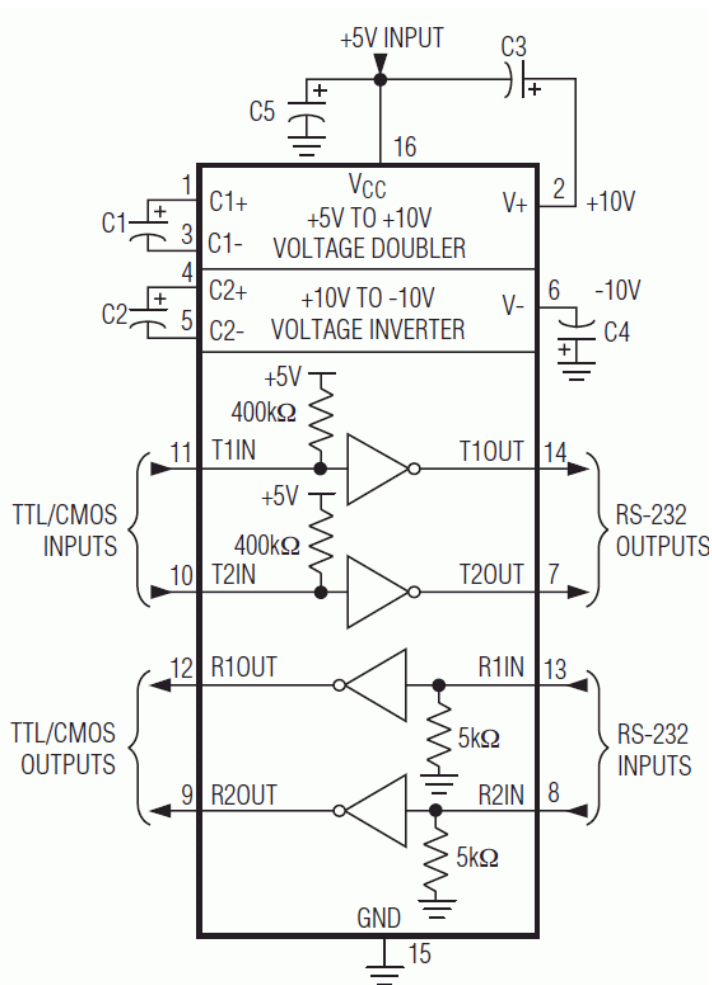


Рисунок 3 – Внутреннее устройство микросхемы MAX232

#### 1.4.4 Описание работы функциональной схемы

На функциональной схеме показана подробная структура микроконтроллера и динамика 3 ГДШ-2, которые описывают внутренние процессы работы системы.

Динамик 3 ГДШ-2 подключается к МК следующим образом: сигнал с порта PD5 через резистор подключается к транзистору KT3102. Транзисторный каскад необходим для нагрузочной способности, таким образом сигнал с выхода PD5 усилится, что даст звуку достаточную громкость.

Клавиши синтезатора («C0»(до), «D»(ре), «E»(ми), «F»(фа), «G»(соль), «A»(ля), «B»(си), «C1»(до)) подключаются к PB[0...7].

Кнопки «MEL», «SYNT», «+», «-» подключаются к PC[0...3].

Линии PD0 (RXD) и PD1(TXD) подключаются к драйверу RS232 – MAX232, по линии RXD идет прием данных от ПЭВМ, а по линии TXD – отправка.

В качестве внешнего источника тактирования используется кварцевый резонатор 1 МГц.

## **2 Описание используемых элементов**

### **2.1 Корпус ATmega8515 и его особенности**

Atmega8515 является КМОП 8-битным микроконтроллером построенным на расширенной AVR RISC архитектуре. Используя команды, исполняемые за один машинный такт, контроллер достигает производительности в 1 MIPS на рабочей частоте 1 МГц, что позволяет разработчику эффективно оптимизировать потребление энергии за счёт выбора оптимальной производительности.

AVR ядро сочетает расширенный набор команд с 32 рабочими регистрами общего назначения. Все 32 регистра соединены с АЛУ, что обеспечивает доступ к двум независимым регистрам на время исполнения команды за один машинный такт.

ATmega8515 содержит 8 Кбайт внутрисистемной программируемой FLASH памяти программ, допускающей чтение во время записи, 512 байт EEPROM, 32 рабочих регистра, три гибких таймера-счётчика с модулем сравнения, внутренние и внешние прерывания, последовательный программируемый интерфейс USART, байт-ориентированный двухпроводный последовательный интерфейс,

В режиме Выкл. процессор сохраняет содержимое всех регистров, замораживает генератор тактовых сигналов, приостанавливает все другие функции кристалла до прихода внешнего прерывания или поступления внешней команды reset.

В режиме ожидания работает один тактовый генератор, при остановке остальных функций контроллера. Благодаря быстрому переходу в нормальный режим работы в том числе и по внешнему прерыванию Atmega8515 успешно приспосабливается к внешним условиям работы и требует меньше энергии, чаще оказываясь в режиме Выкл. В расширенном

режиме ожидания в рабочем состоянии находятся основной генератор и асинхронный генератор.

Микросхемы выпускаются при использовании Atmel технологии энергонезависимой памяти высокой плотности. Встроенная ISP FLASH память позволяет перепрограммировать область программной памяти внутрисистемно через последовательный SPI интерфейс стандартным программатором, или используя загрузочную программу из энергонезависимой памяти, работающую в AVR ядре. Комбинация расширенной 8-и битной RISC архитектуры ЦПУ и твердотельной FLASH памяти обеспечивают Atmega8515 высокую гибкость и экономическую эффективность во встраиваемых системах управления.

Конфигурация выводов микроконтроллера для варианта корпуса PDIP приведена на рисунке 4.

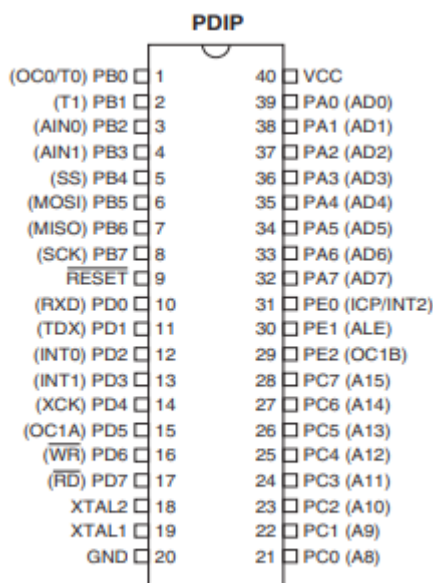


Рисунок 4 – Конфигурация выводов ATmega8515 PDIP

Микроконтроллер в данной работе используется в качестве центрального управляющего устройства, к которому подключены клавиатура, динамик, светодиоды, и драйвер MAX232.



## **2.2 Динамик и выбор кнопок**

В качестве кнопок для клавиатуры и кнопки сброса МК были выбраны кнопки без фиксации PBS-10B-2. Они миниатюрны, удобны в монтаже, недороги и надежны.

В качестве динамика был выбран 3 ГДШ-2-8. Данный динамик подходит по диапазону частот и амплитудно частотным характеристикам. Диапазон частот динамика: 100-12500 Гц. Неравномерность АЧХ: 12дБ. Паспортная мощность: 3 Вт.

## **2.3 Питание схемы**

В АТТІNY2313 встроена схема контроля питающего напряжения, которая запускает микроконтроллер только при безопасном уровне питающего напряжения.

При подключении питания к микроконтроллеру необходимо использовать сглаживающие конденсаторы емкостью 0,1 мкФ.

В данной работе используется LM7805 для преобразования входного напряжения +12 В от внешнего блока питания в +5 В для подачи питания на МК.

LM7805 стабилизатор напряжения с интегральной схемой с фиксированным напряжением. Это устройство используется с внешними компонентами для получения регулируемых выходных напряжения и тока.

Схема работы стабилизатора LM7805 изображена на рисунке 5. У стабилизатора 3 вывода, на первый вывод подается входное напряжение (+12В), второй вывод – это земля, а третий вывод — это выходное напряжение (+5В). Схема подключения питания к МК приведена на рисунке 6.

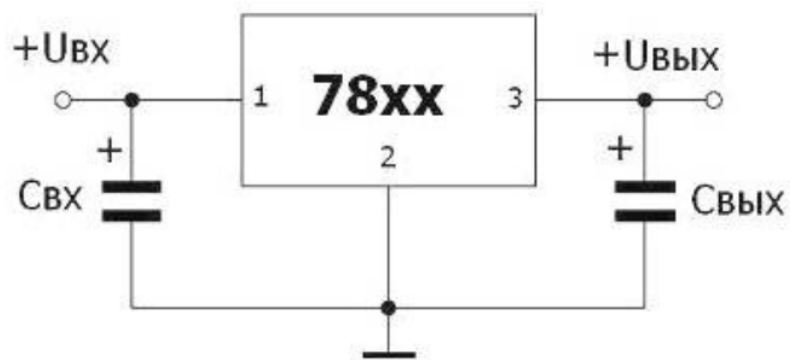


Рисунок 5 – Схема работы стабилизатора

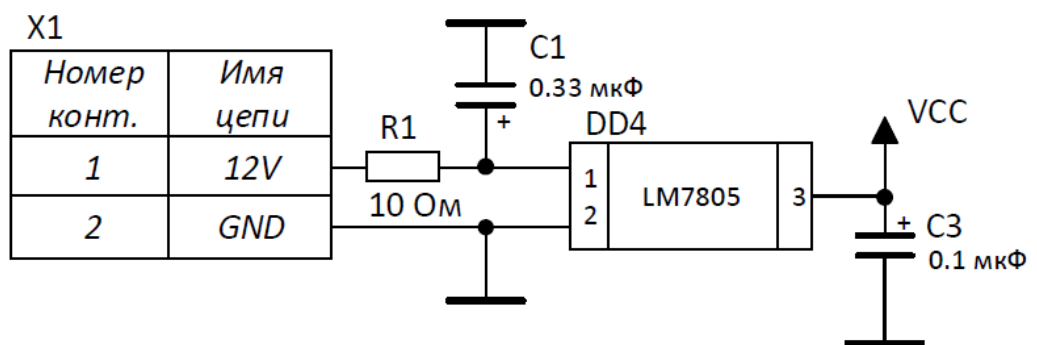


Рисунок 6 – Схема питания

## **2.4 Основные элементы принципиальной схемы**

По результатам анализа возможностей реализации требований к конечному устройству можно сформулировать перечень структурных элементов, из которых состоят блоки функциональной схемы:

- Микроконтроллер ATmega8515;
- Динамик 3 ГДШ-2;
- Драйвер преобразования напряжение MAX232;
- Программатор;
- Стабилизатор напряжения LM7805;
- Клавиатура, состоящая из 13 кнопок.

### 3 Расчёт потребляемой мощности системы

Для определения потребляемой мощности МК воспользуемся графиком тока через вывод микроконтроллера в зависимости от частоты и питающего напряжения, приведенного на рисунке 7.

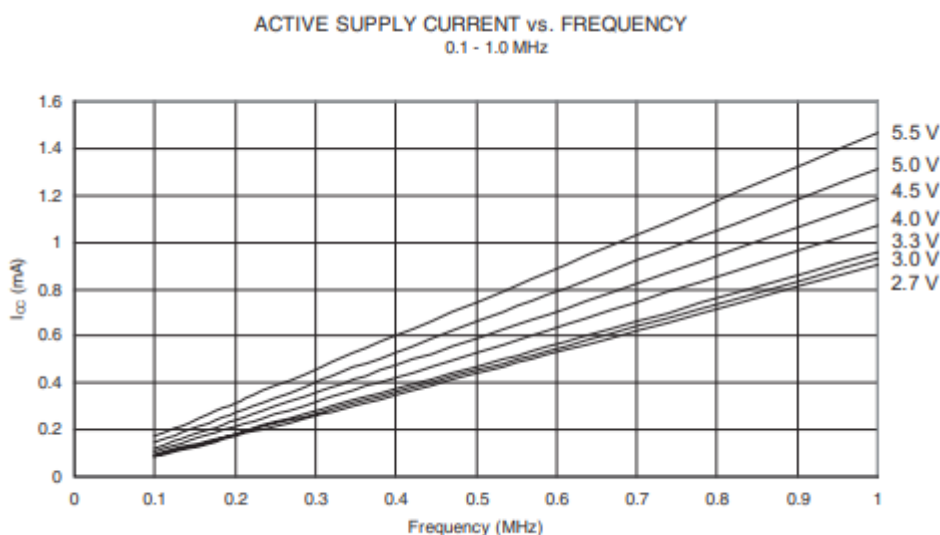


Рисунок 7 – График зависимости тока через вывод МК в зависимости от частоты и питающего напряжения

Для расчета мощности воспользуемся формулой:

$$P_{ATmega8515} = I_{CC} * U_{VCC} * N = 1,3 * 10^{-3} * 5 * 20 = 0.13 \text{ Вт} ,$$

где  $I_{CC}$  – ток на выводах МК = 1,3 мА (при 1 МГц, 5В),  $U_{VCC}$  – напряжение питания = 5 В,  $N$  – число задействованных выводов = 20.

Рассчитаем мощность каждой отдельной микросхемы по формуле  $P_{\text{микросхемы}} = I_{CC} * U_{VCC}$ :

$$P_{LM7805} = 5 \text{ мА} * 12 \text{ В} = 60 \text{ мВт}$$

$$P_{MAX232} = 10 \text{ мА} * 5 \text{ В} = 50 \text{ мВт}$$

$$P_{3 ГДШ-2} = 0.625 A * 5 B = 3.1 Bm$$

$$P_{КТ 3102} = 2.5 mA * 5 B = 12.5 мВm$$

Суммарная мощность получается:

$$P = P_{ATmega 8515} + P_{LM 8705} + P_{3 ГДШ-2} + P_{КТ 3102} + P_{МАХ 232} = 3.352 Bm$$

## 4 Описание алгоритмов системы

Разработанная система состоит из алгоритмов: основной работы, инициализации микроконтроллера, проигрывания ноты, проигрывания мелодии, обработки USART прерывания, сохранение мелодии в память из буфера.

Алгоритм основной работы приведен на рисунке 8 а описание по пунктам представлено ниже:

1. вызов подпрограммы инициализации микроконтроллера;
2. записывание значения порта РС в переменную;
3. опрашивание нажатия кнопки;
4. перевод системы в состояние ввода пароля;
5. записываем значение порта РВ в переменную;
6. опрос нажатия кнопок;
7. в зависимости от режима проигрывается мелодия или нота;
8. переход к пункту 2:

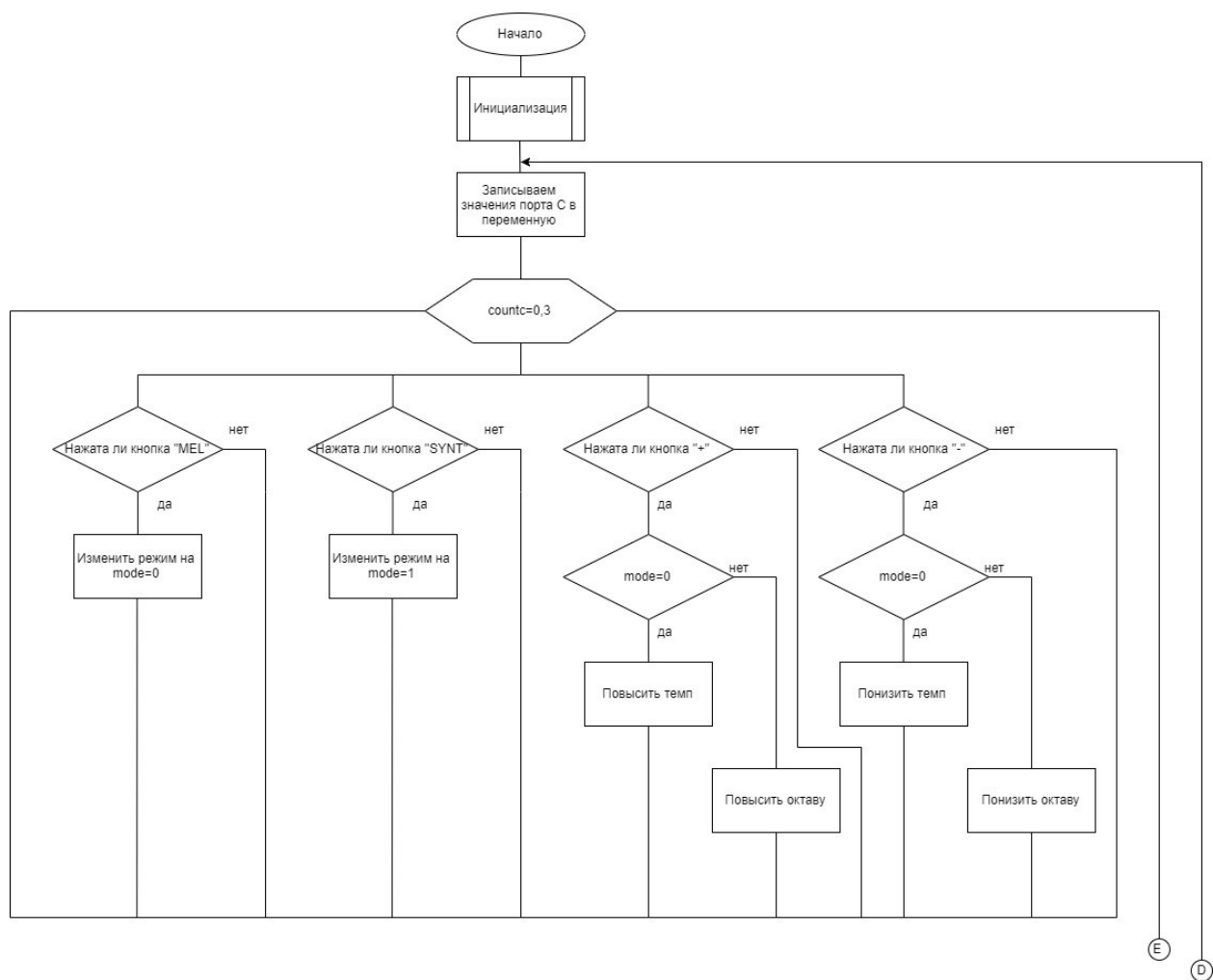


Рисунок 8 – Блок-схема алгоритма основной работы

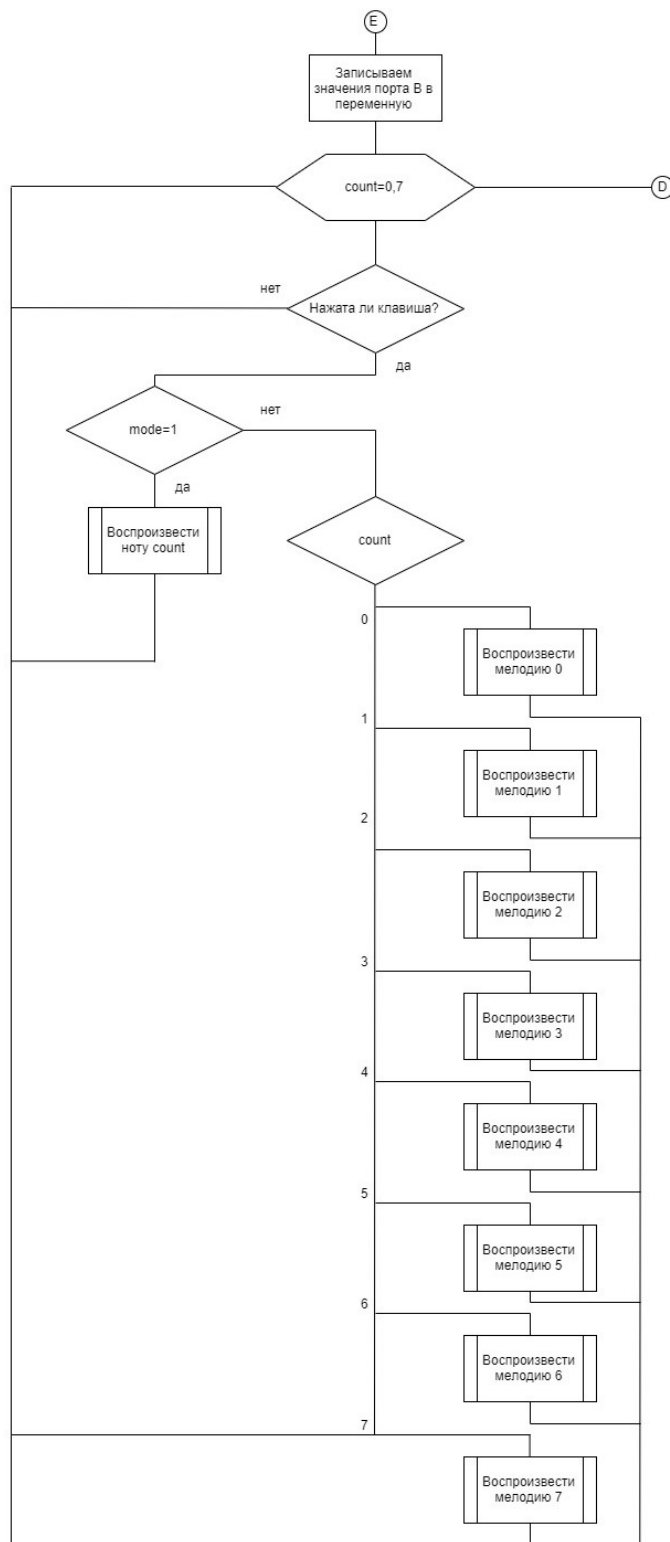
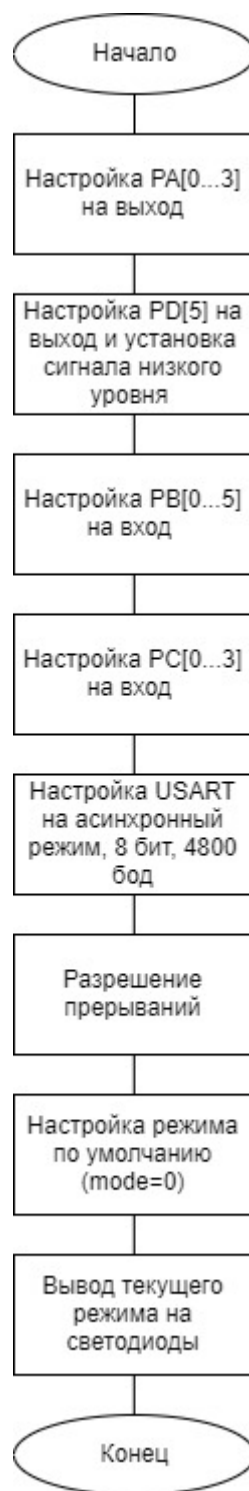


Рисунок 8 – Блок-схема алгоритма основной работы (продолжение)

Алгоритм инициализации микроконтроллера приведен на рисунке 9 и может быть описан следующим образом:

1. настройка линий PA0, PA1, PA2, PA3 на выход;
2. настройка линий PD5 на выход и установка сигнала низкого уровня, для того чтобы через динамик постоянно не шел ток;
3. настройка всех линий порта PB на вход;
4. настройка PC0, PC1, PC2, PC3, на вход;
5. настройка модуля USART на работу в асинхронном режиме со скоростью 4800 бод;
6. разрешение прерываний;
7. настройка режима по умолчанию, чтобы МК сразу начал работу в одном из режимов;
8. вывод текущего режима на светодиоды;

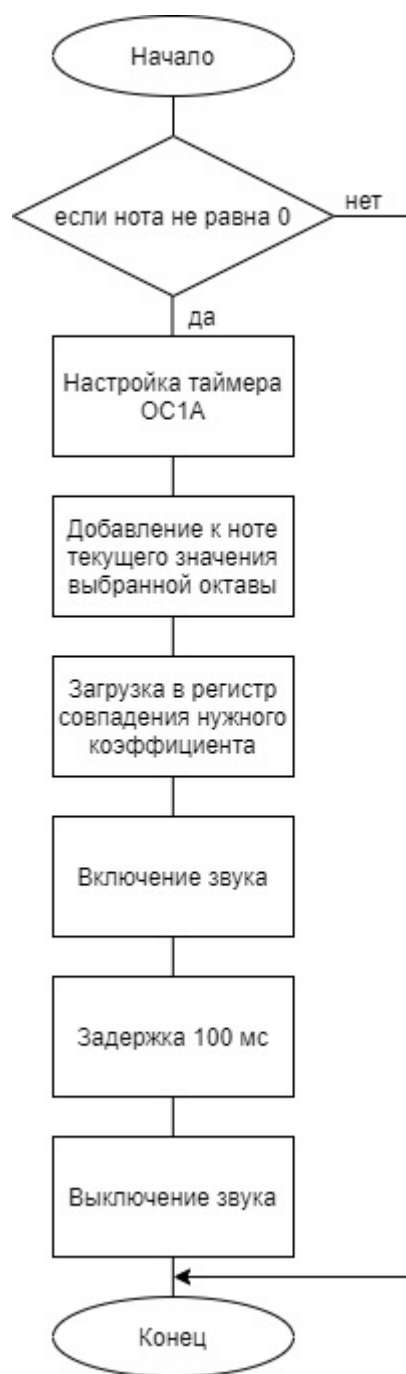




*Рисунок 9 – Блок-схема алгоритма инициализации микроконтроллера*

Алгоритм отправки проигрывания ноты приведен на рисунке 10 и может быть описан следующим образом:

1. проверка значения ноты, если нота равна нулю, то ее проигрывать не нужно;
2. настройка таймера OC1A;
3. добавление к ноте текущего значения октавы, с помощью кнопок «+» и «-» можно задать октаву. Изначально значение октавы равно 0;
4. включение звука;
5. задержка 100 мс;
6. выключение звука.

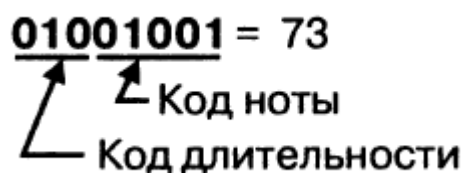


*Рисунок 10 – Блок-схема алгоритма проигрывания ноты*

Любая мелодия состоит из нот. Каждая нота имеет свой тон (частоту) и длительность звучания. Весь музыкальный ряд делится на октавы. Каждая октава делится на 12 нот. Семь основных нот (белые клавиши) и пять дополнительных (черные клавиши). В настоящее время все 12 нот одной октавы равнозначны. Частоты любых двух соседних нот отличаются друг от друга в одинаковое количество раз. При этом частоты одноименных нот в двух соседних октавах отличаются ровно в два раза.

В музыке применяют не произвольную длительность, а длительность, выраженную долями от целой (1,  $\frac{1}{2}$ ,  $\frac{1}{4}$ ,  $\frac{1}{8}$ ,  $\frac{1}{16}$ ,  $\frac{1}{32}$ ...и т. д.). В зависимости от темпа реальная длительность целой ноты меняется. Для сохранения мелодии необходимо соблюдать лишь соотношения между длительностями

Для экономии памяти удобнее каждую ноту кодировать одним байтом. Три старших бита используются для кодирования длительности ноты, а оставшиеся пять – для кодирования ее тона (рисунок 11).



*Рисунок 11 – Кодирование ноты*

Цепочка таких кодов будет представлять закодированную мелодию. Для определения конца мелодии используется код 255. В памяти МК хранятся массивы с коэффициентами длительности и коэффициентами совпадения для таймера. В зависимости от кода ноты и длительности МК выбирает из этих массивов необходимые коэффициенты для проигрывания мелодии.

Алгоритм проигрывания мелодии приведен на рисунке 12 и состоит из следующих пунктов:

1. инициализация переменных необходимых для работы алгоритма;
2. настройка таймера OC1A;
3. присвоения указателю адреса начала мелодии;
4. проверка на нажатии клавиши порта PB, если клавиша не нажата то переход к пункту 16;
5. проверка на конец мелодии, если мелодия закончилась, то возврат к пункту 3;
6. присвоение переменным соответствующего коэффициента совпадения и задержки для паузы между нотами;
7. если нота равна 0, то это пауза. Переход к пункту 10;
8. загрузка в регистр совпадения нужного коэффициента;
9. включение звука;
10. цикл длительности звучания ноты. Длительность ноты зависит от текущего темпа. Повторение пункта 11 необходимое количество раз;
11. задержка 10мс;
12. выключение звука
13. цикл задержки между нотами. Необходим для того чтобы ноты было проще отличить друг от друга, также зависит от темпа. Повторить пункт 14 необходимое количество раз;
14. задержка 10мс;
15. сдвиг указателя на следующий элемент. Таким образом реализовано продвижение по массиву мелодии;
16. выключение звука.

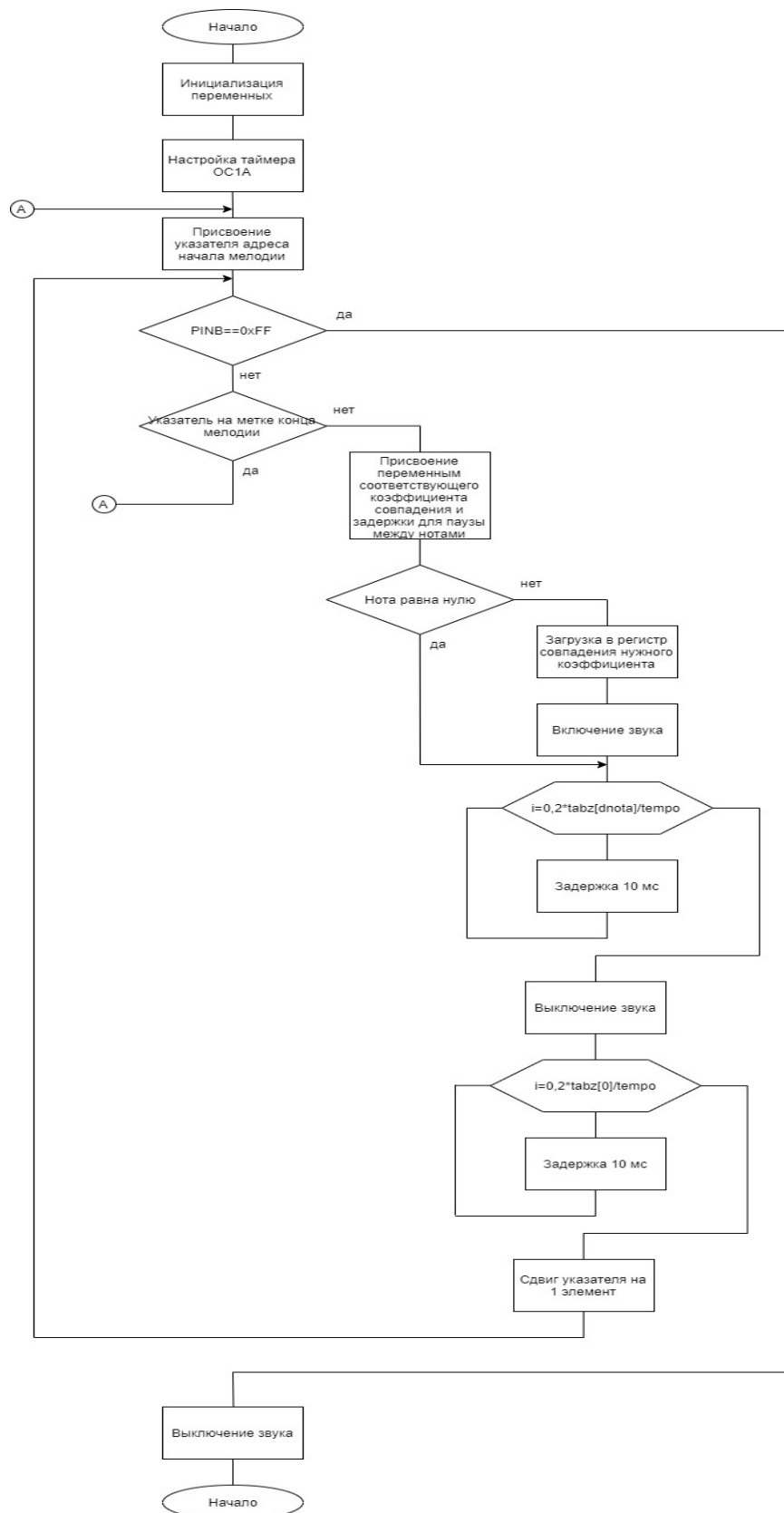


Рисунок 12 – Алгоритм проигрывания мелодии

Алгоритм обработки USART прерывания представлен на рисунке 13 и описывается следующий образом:

1. присвоение переменной значения регистра UDR;
2. проверка введенного символа, чтобы избежать ошибок разрешено вводить лишь некоторые символы. Если символ введен не верно, то на терминал передается сообщение поясняющее ошибку и на этом обработка прерывания завершится;
3. проверка текущего режима работы МК, если МК находится в режиме «MEL», то вызовется функция проигрывания ноты и на этом прерывания завершится;
4. буфер, в который записываются данные пришедшие по USART, проверяется на оставшееся место, если массив заполнен, то в терминал передается сообщение об ошибке и далее переход к пункту 9;
5. если пришел символ воспроизведения мелодии, то переход к пункту 6, если нет то к пункту 8;
6. вызов функции записи из буфера в память МК;
7. обнуление буферного массива и завершение обработки прерывания и переход к пункту 9;
8. проверка корректности ввода данных, в случае возникновения ошибок, сообщение оповещающее о некорректном вводе данных будет передано на терминал.
9. вывод сообщения на терминал, которое содержит уже введенную ранее мелодию.

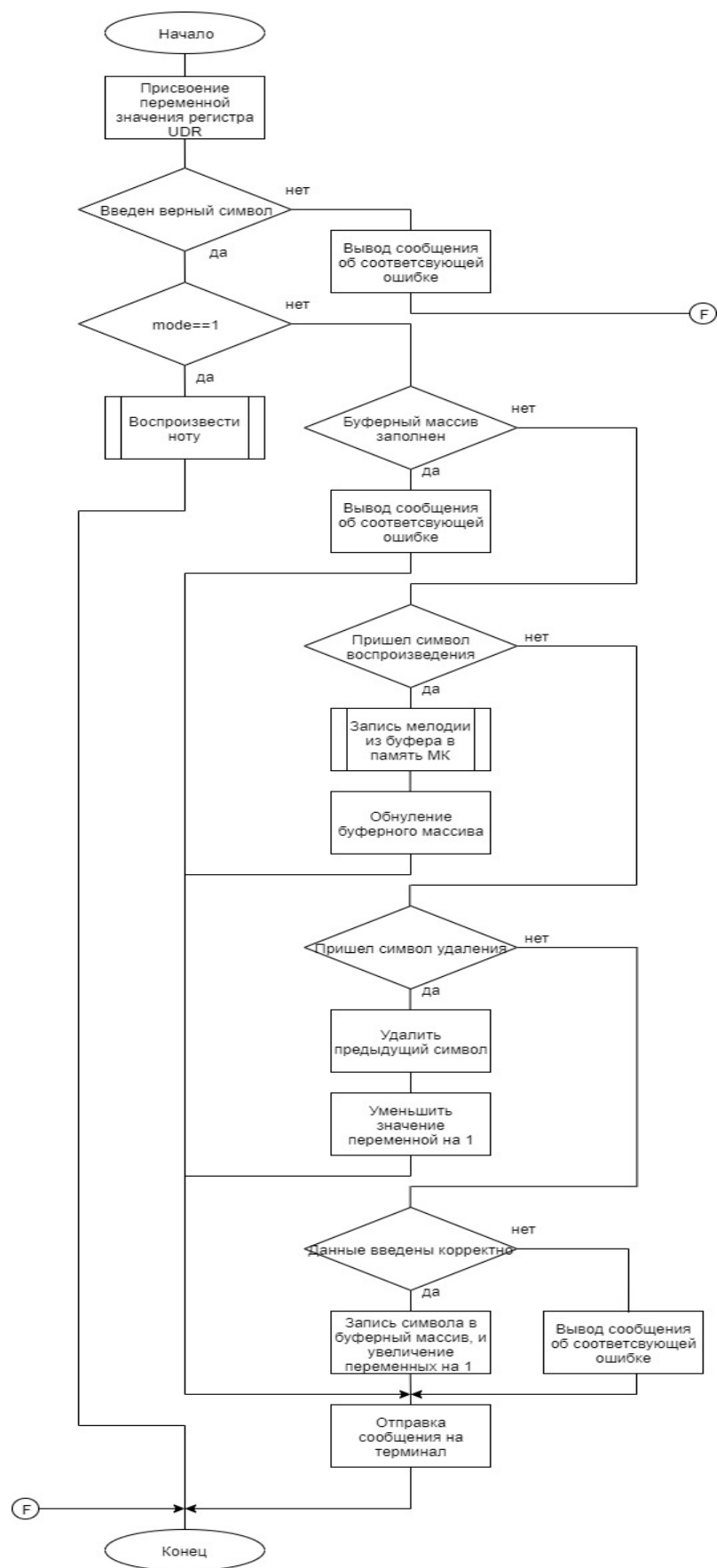


Рисунок 13 – Алгоритм обработки USART прерывания



Алгоритм записи буфера в память МК приведен на рисунке 14 и состоит из следующих пунктов:

1. инициализация переменных;
2. цикл для прохождения по буферному массиву;
3. если текущий символ не «,», то переход к пункту 2;
4.  $b-3$  позволяет вернуться в буферном массиве на 3 элемента назад;
5. цикл для записи элементов буфера в одну строку. Повторяется 3 раза;
6.  $b+4$  чтобы перейти к следующему символу в массиве после запятой;
7. перевод строки в числовое значение;
8. в зависимости от текущего значения `buf` будет выбрана мелодия из памяти МК которая будет перезаписана;
9. запись числа как код символа в символьный массив мелодии;
10. в зависимости от значения `buff` будет добавлен символ конца мелодии (0xFF) к мелодии;
11. увеличение значения `buff` на 1;
12. если `buf==4` , то обнуляется значение `buf`.

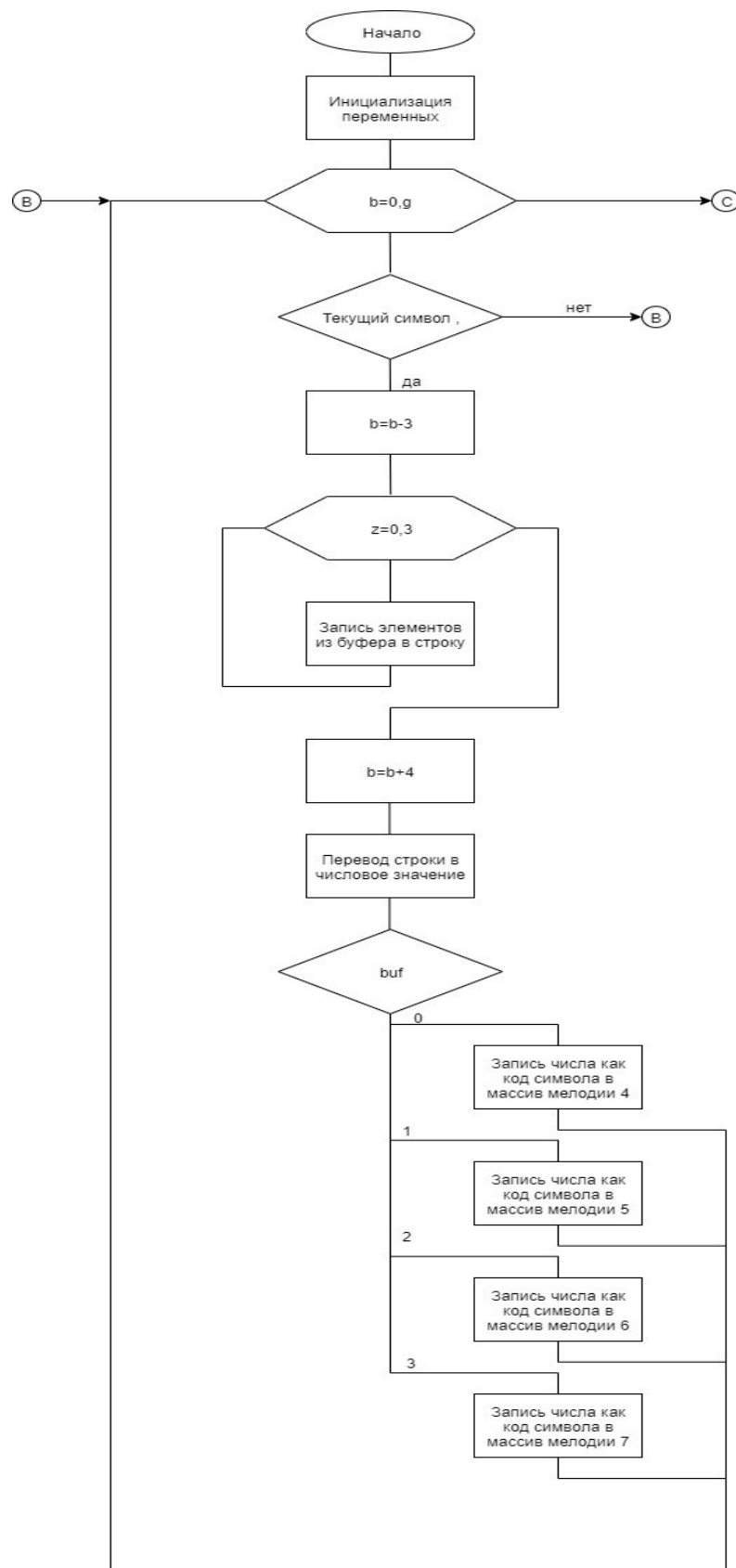


Рисунок 14 – Блок-схема алгоритма записи буфера в память МК

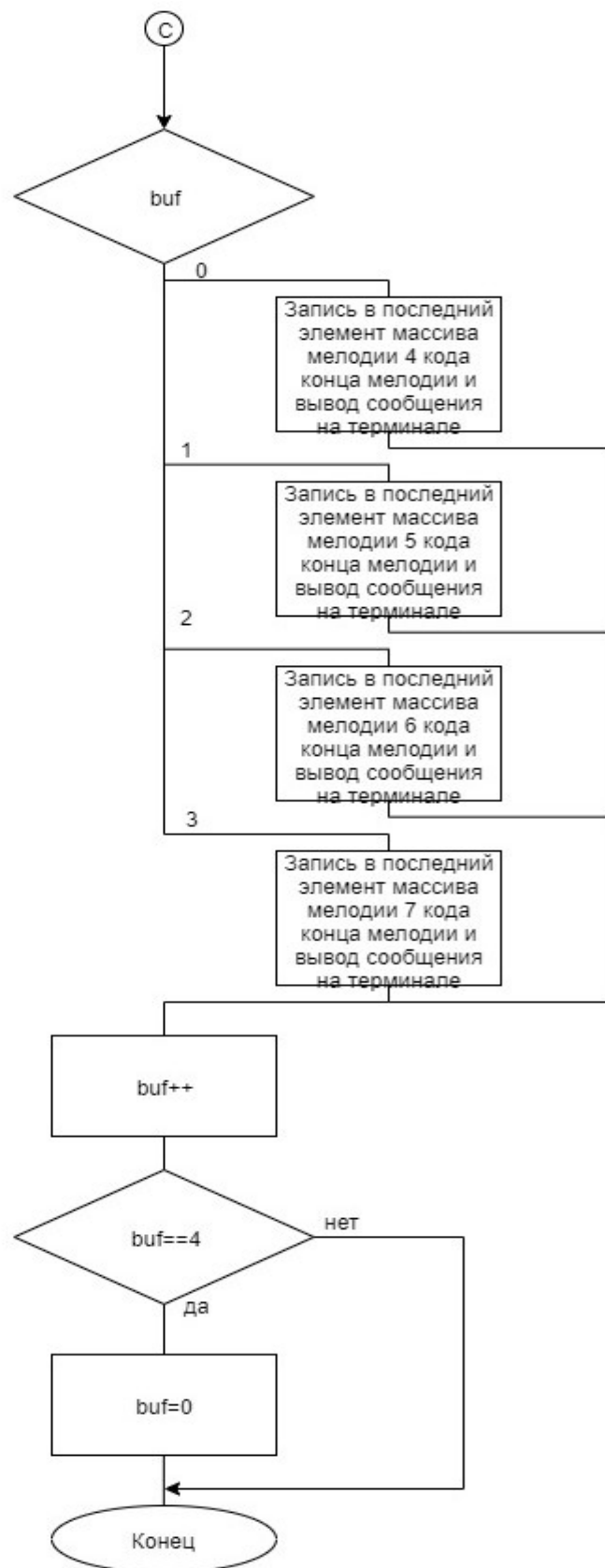


Рисунок 14 – Блок-схема алгоритма записи буфера в память МК  
(продолжение)

## Расчеты.

МК при помощи выхода таймера OC1A создает прямоугольную волну определенной частоты. Для воспроизведения ноты определенной частоты, необходимо учитывать специфику работы таймера в режиме сброса при совпадении. При достижении TCNT1 значения в OCR1A, значение счетчика TCNT1 сбрасывается и выход OC1A меняется на противоположный. Для подходящего периода прямоугольной волны необходимо чтобы частота срабатывания таймера OC1A была в два раза больше чем частота воспроизводимой ноты. Пример для воспроизведения ноты D (150 Гц) потребуется частота:

$$f = \frac{F_{\text{МК}}}{OCR1A * K} = 298 \text{ Гц}$$

где  $F_{\text{МК}}$  – частота работы МК (1 МГц), OCR1A – значение регистра сравнения, K – предделитель. Выберем OCR1A = 3356, а предделитель = 1. Полученная частота срабатывания таймера почти в двое больше частоты ноты. Такая погрешность допустима, потому что очень мала и не заметна на слух.

Скорость работы модуля USART в бодах рассчитывается по формуле:

$$baud = \frac{F_{\text{МК}}}{16 * (UBRR + 1)}$$

где  $F_{\text{МК}}$  – частота работы МК (1 МГц), UBRR – значение регистра. Для организации скорости 4800 бод выберем UBRR = 12.

## 5 Моделирование и отладка системы

Моделирование и отладка проводилась при помощи программного комплекса ISIS Proteus и компилятора AtmelStudio для программирования на языке C.

Модель системы, собранная в Proteus, приведена на рисунке 15.

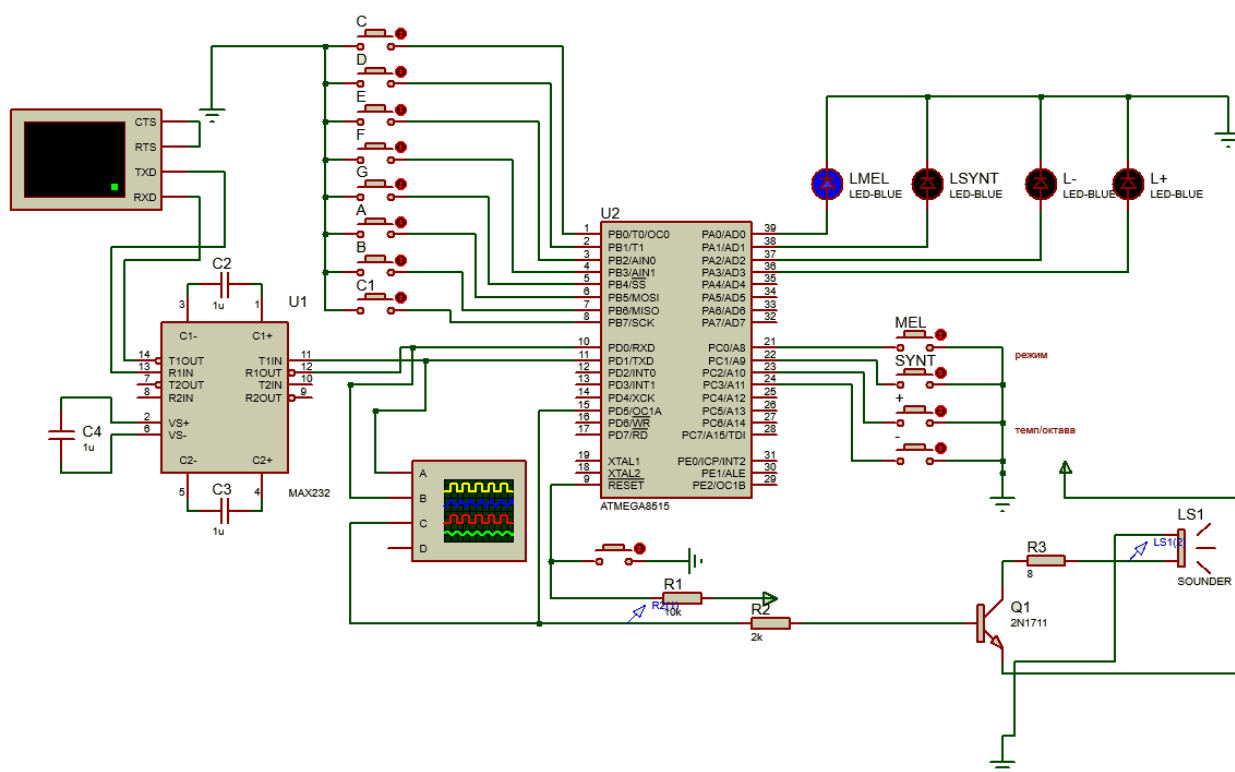
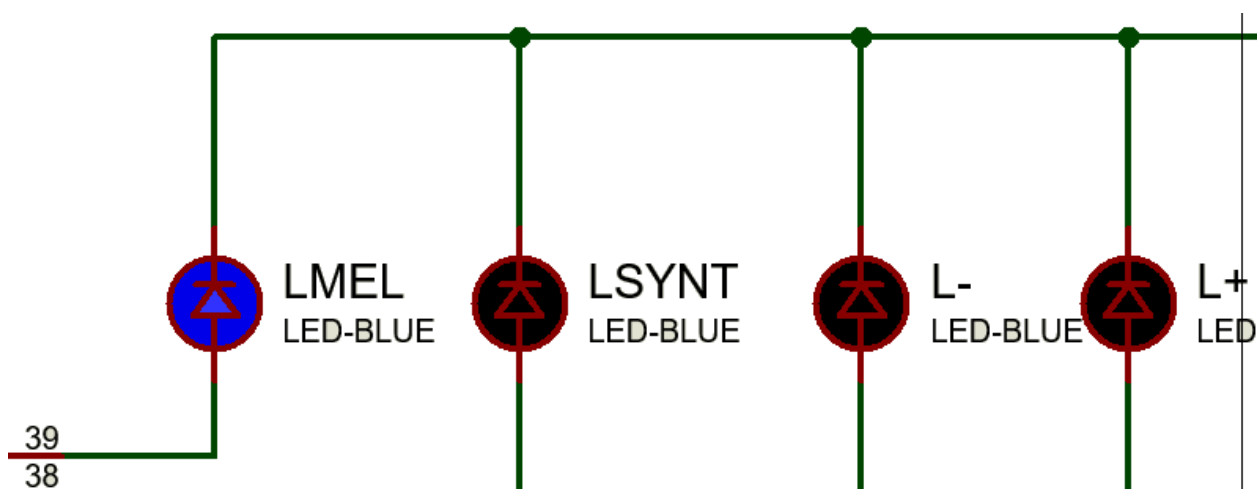


Рисунок 15 – Модель системы

Для отладки обмена информацией с ПЭВМ используется специальный инструмент Virtual terminal, который позволяет управлять приемом и передачей данных по протоколу UART (аналог RS-232).

При замыкании одной из клавиш («C0»(до), «D»(ре), «E»(ми), «F»(фа), «G»(соль), «A»(ля), «B»(си), «C1»(до)) микроконтроллер проигрывает ноту или мелодию в зависимости от выбранного режима.

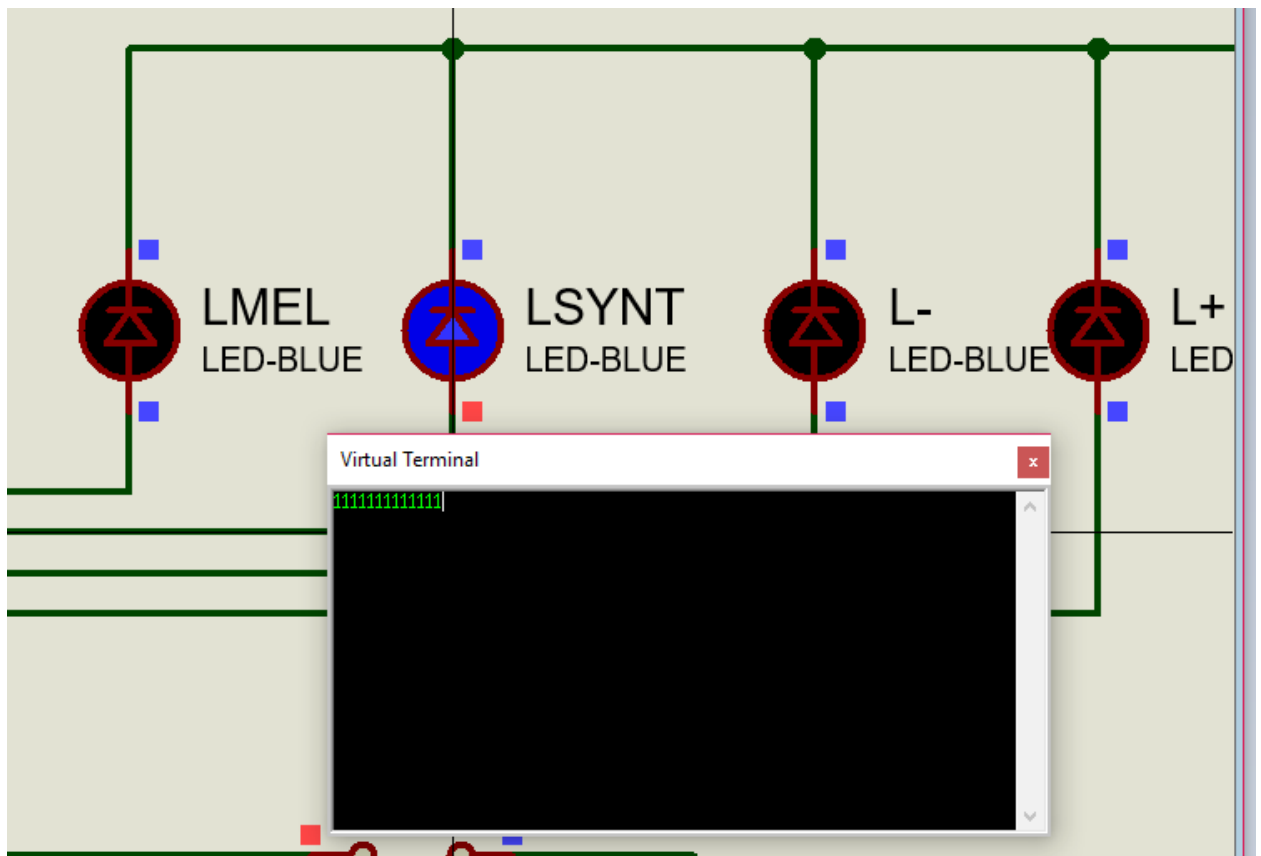
При нажатии на кнопки «SYNT» или «MEL» МК меняет режим работы, и отображает на светодиодах режим в котором находится (рисунок 16).



*Рисунок 16 – Отображение светодиодов*

При нажатии на кнопки «+» или «-» МК либо меняет октаву, либо меняет темп в зависимости от выбранного режима. В момент нажатия мигают светодиод соответствующей клавиши, означая что кнопка была нажата .

Находясь в режиме SYNT при получении с терминал ПЭВМ байта, МК запускает обработчик прерывания и проигрывает ноту соответствующую полученному байту (рисунок 17).



*Рисунок 17 – Проигрывание ноты в режиме SYNT*

Находясь в режиме MEL при получении с терминал ПЭВМ байта, МК запускает обработчик прерывания и в зависимости от полученного байта либо записывает его в буфер, либо вызывает функцию записи из буфера в память, либо отправляет на терминал ПЭВМ сообщение о соответствующей ошибке (рисунок 18, рисунок 19).



```
Virtual Terminal
melody:
14
melody:
142
melody:
142,
melody:
142,1
melody:
142,14
melody:
142,143
melody:
142,143
melody is on PB4
```

*Рисунок 18 – Запись в память мелодии из буфера в режиме MEL*



```
Virtual Terminal
error!
Put numbers 1..7
1
melody:
12
melody:
123
melody:
1233
error!
expected ','
melody:
123
```

*Рисунок 19 – Вывод сообщения об ошибке в режиме MEL*

ISIS Proteus ограничен в анализе звуковых сигналов. Программа Ableton Live 10 suite поможет в анализе звуковых частот. При проигрывании аудио записанного при нажатии всех кнопок («C0»(до), «D»(ре), «E»(ми), «F»(фа), «G»( соль), «A»(ля), «B»(си), «C1»(до)) в режиме SYNT были получены следующие результаты



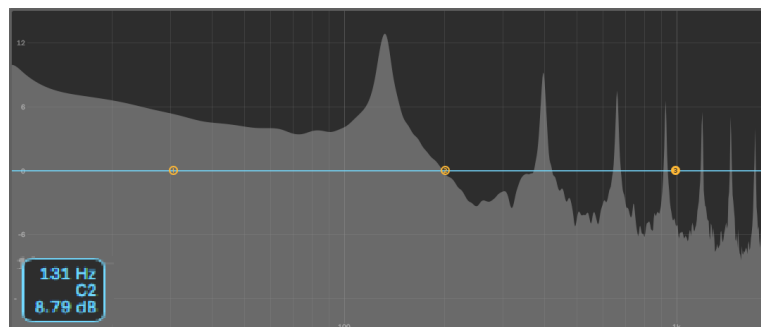


Рисунок 20 – Окно эквалайзера при нажатии на «C»

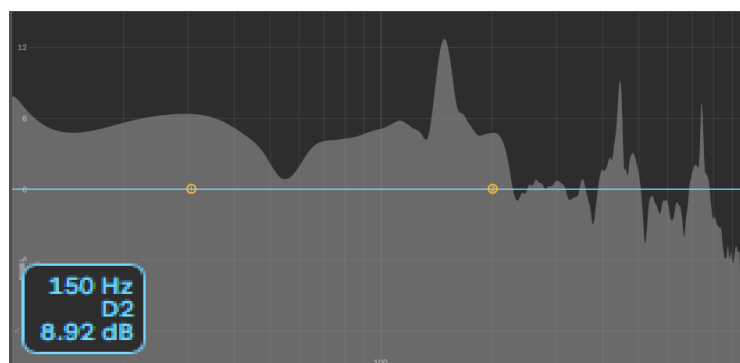


Рисунок 21 – Окно эквалайзера при нажатии на «D»

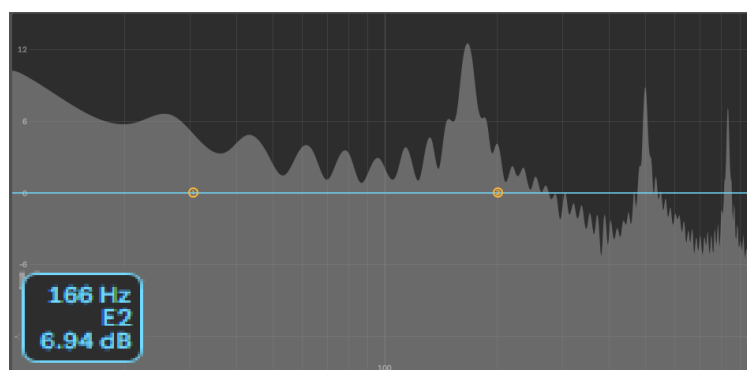


Рисунок 22 – Окно эквалайзера при нажатии на «E»

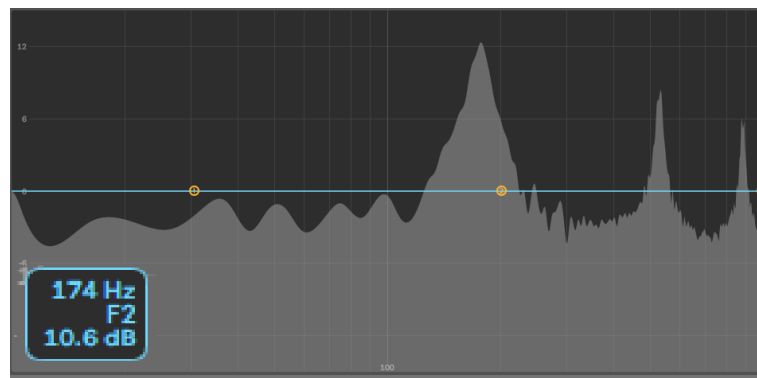


Рисунок 23 – Окно эквалайзера при нажатии на «F»

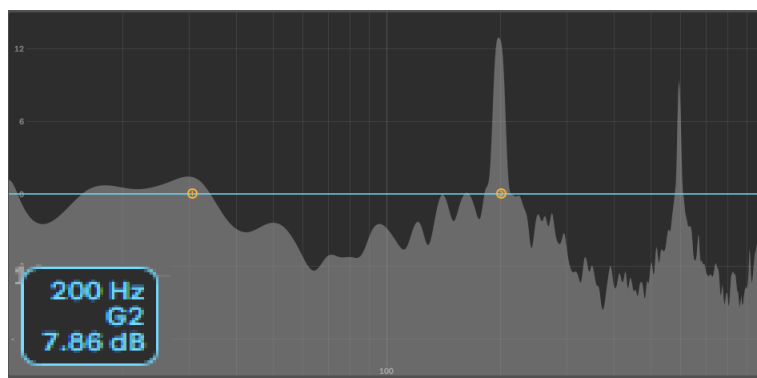


Рисунок 24 – Окно эквалайзера при нажатии на «G»

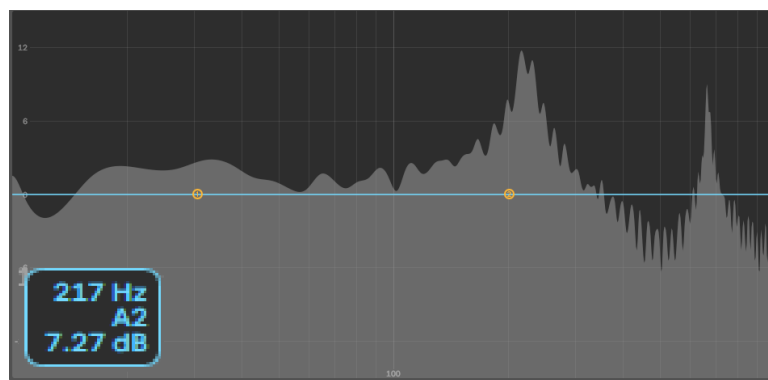
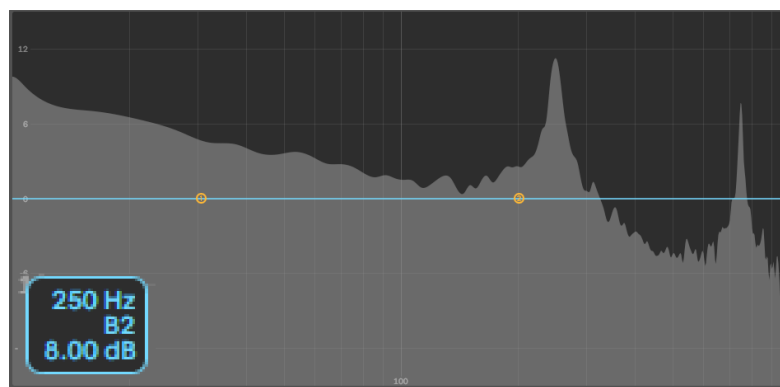
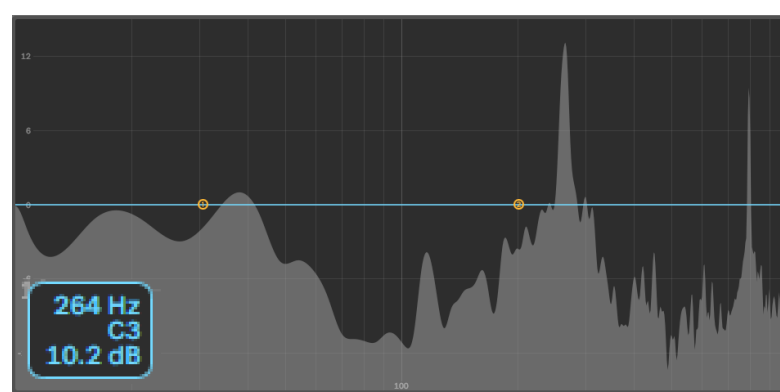


Рисунок 25 – Окно эквалайзера при нажатии на «A»



*Рисунок 26 – Окно эквалайзера при нажатии на «B»*



*Рисунок 27 – Окно эквалайзера при нажатии на «C1»*

Из полученных результатов можно сделать вывод, о том что синтезатор настроен корректно и воспроизведение частоты звуков соответствует частоте нот.

ISIS Proteus поддерживает классический функционал отладки программы. Если поставить на паузу симуляцию, то появится возможность проводить пошаговое выполнение программы и видеть внутреннее представление данных, памяти и регистров. При помощи точек останова и выполнения программы по шагам можно отладить сложные ошибки программы.

## 5 Способ программирования МК

Для программирования МК используется внутрисхемное программирование (ISP) при помощи канала SPI. Оно осуществляется путем послыки 4-байтовых команд (рисунок 28) на вывод MOSI МК, в который один или два байта определяют тип операции, остальные – адрес, записываемый байт, установочные биты и биты защиты, пустой байт. При выполнении операции чтения считываемый байт снимается через вывод MISO. В каждой команде указывается адрес записываемой ячейки и записываемое значение.

При программировании микроконтроллера производится установка конфигурационных битов (FUSE Bits). При выполнении операции стирания памяти, выполняемой перед перепрограммированием, конфигурационные биты и байты идентификации остаются без изменений, в отличие от ячеек Flash памяти и памяти EEPROM, которым присваивается значение «1».

Конфигурационные биты доступны только при программировании. ATmega8515 имеет 2 конфигурационных бита:

- SPIEN – при программировании в «1» запрещает параллельное программирование;
- FSTRT – задает время задержки старта AVR-микроконтроллера после сброса.

Чтобы завершить программирование и перевести МК в рабочий режим, достаточно подать на RESET логический уровень «1». Контроллер запустится и будет работать по новой программе.

**Table 94.** Serial Programming Instruction Set

Instruction	Instruction Format				Operation
	Byte 1	Byte 2	Byte 3	Byte4	
Programming Enable	1010 1100	0101 0011	xxxx xxxx	xxxx xxxx	Enable Serial Programming after RESET goes low.
Chip Erase	1010 1100	100x xxxx	xxxx xxxx	xxxx xxxx	Chip Erase EEPROM and Flash.
Read Program memory	0010 H000	0000 aaaa	bbbb bbbb	oooo oooo	Read H (high or low) data o from Program memory at word address a:b.
Load Program memory Page	0100 H000	0000 xxxx	xxxb bbbb	iiii iiii	Write H (high or low) data i to Program memory page at word address b. Data low byte must be loaded before Data high byte is applied within the same address.
Write Program memory Page	0100 1100	0000 aaaa	bbbх xxxx	xxxx xxxx	Write Program memory Page at address a:b.
Read EEPROM Memory	1010 0000	00xx xxxa	bbbb bbbb	oooo oooo	Read data o from EEPROM memory at address a:b.
Write EEPROM Memory	1100 0000	00xx xxxa	bbbb bbbb	iiii iiii	Write data i to EEPROM memory at address a:b.
Read Lock bits	0101 1000	0000 0000	xxxx xxxx	xxoo oooo	Read Lock bits. "0" = programmed, "1" = unprogrammed. See Table 81 on page 177 for details.
Write Lock bits	1010 1100	111x xxxx	xxxx xxxx	11ii iiii	Write Lock bits. Set bits = "0" to program Lock bits. See Table 81 on page 177 for details.
Read Signature Byte	0011 0000	00xx xxxx	xxxx xxbb	oooo oooo	Read Signature Byte o at address b.
Write Fuse bits	1010 1100	1010 0000	xxxx xxxx	iiii iiii	Set bits = "0" to program, "1" to unprogram. See Table 84 on page 179 for details.
Write Fuse High Bits	1010 1100	1010 1000	xxxx xxxx	iiii iiii	Set bits = "0" to program, "1" to unprogram. See Table 83 on page 178 for details.
Read Fuse bits	0101 0000	0000 0000	xxxx xxxx	oooo oooo	Read Fuse bits. "0" = programmed, "1" = unprogrammed. See Table 84 on page 179 for details.
Read Fuse High Bits	0101 1000	0000 1000	xxxx xxxx	oooo oooo	Read Fuse high bits. "0" = programmed, "1" = unprogrammed. See Table 83 on page 178 for details.
Read Calibration Byte	0011 1000	00xx xxxx	0000 0000	oooo oooo	Read Calibration Byte

*Рисунок 28 – Формат байтов команд для программирования*

Примечание:

- a - адрес старших разрядов;
- b - адрес младших разрядов;
- H - 0 - мл. байт, 1 - ст. байт;
- o - ВЫВОД ДАННЫХ;
- i - ВВОД ДАННЫХ;

–  $x$  - произвольное значение.

Для программирования микроконтроллера был выбран программатор AVR-ISP500. AVR-ISP500 — это быстрый и надежный USB-AVR-программатор, который работает как с студией Atmel, так и с AVRDUDE. Он распознается как программист STK500 и использует протокол связи STK500v2.

Программирование памяти программ и данных осуществляется через последовательный интерфейс SPI. Для перевода микроконтроллера в режим программирования необходимо подать низкий уровень напряжения на вход RESET, выждать не менее 20мс, после чего послать инструкцию разрешения программирования на вывод MOSI интерфейса SPI. Так как микроконтроллер может быть запрограммирован только тогда, когда сигнал RESET на входе микроконтроллера равен нулю. Поэтому сигнал TRST подключается к сигналу RESET микроконтроллера.

## **ЗАКЛЮЧЕНИЕ**

В результате проделанной работы был спроектирован и разработан звуковой синтезатор нот и мелодий.

Проигрывание нот осуществляется путем нажатия клавиш. В зависимости от выбранного режима устройство будет выдавать звуковые сигналы соответствующей мелодии или ноты. С помощью кнопок настройки можно поменять режим работы устройства, прибавить или убавить октаву, а так же повысить или понизить темп воспроизводимой мелодии. При получении байтов с терминала ПВЭМ устройство будет либо проигрывать ноты, либо записывать мелодию. Устройство в режиме ввода мелодии предоставляет оператору обратную связь, что облегчает пользование устройством.

Разработаны схемы МК-системы, описаны разработанные алгоритмы работы МК, реализована симуляция работы системы, произведена отладка с помощью симулятора Proteus.

Основная аппаратная часть устройства реализована в пределах используемого микроконтроллера, обладающего значительными возможностями ввода-вывода.

Разработанное устройство может быть использовано обучающих и развлекательных целях.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Хартов, В.Я. Микропроцессорные системы: учебник для студ. технических ВУЗов. – Издательство: Академия, 2014. – 368с.
- 2 Хартов, В.Я. Микроконтроллеры AVR. Практикум для начинающих: учеб. пособие. – 2- изд., испр. И доп. – М.: Издательство: МГТУ им. Н.Э. Баумана, 2012. – 280с.
- 3 Хартов В.Я. Проектирование и отладка программ для микроконтроллеров AVR фирмы ATMEL. Изд-во МГТУ им. Н.Э.Баумана, 2004 г
- 4 Белов А.В. Разработка устройств на микроконтроллерах AVR. Изд-во Наука и Техника, 2013 г
- 5 Маслов В. Методические указания по оформлению РПЗ
- 6 Описание (datasheet) микроконтроллера ATmega8515 - <https://static.chipdip.ru/lib/059/DOC000059786.pdf>
- 7 Описание (datasheet) LM016L. Режим доступа: <https://www.alldatasheet.com/datasheet-pdf/pdf/146552/HITACHI/LM016L.html>
- 8 Описание (datasheet) MAX232. Режим доступа: <http://www.ti.com/lit/ds/symlink/max232.pdf>
- 9 Описание микросхемы MAX232 и её работы. Режим доступа: <http://www.joyta.ru/7033-mikroshema-max232-opisanie-sxema-vklyucheniya-datasheet/>
- 10 Программирование микроконтроллеров AVR. – Режим доступа: [https://myrobot.ru/stepbystep/mc\\_programming.php](https://myrobot.ru/stepbystep/mc_programming.php)
- 11 ГОСТ 2.743-91 Обозначения условные в графических схемах. Элементы цифровой техники.
- 12 ГОСТ 2.701-84 Правила выполнения схем.
- 13 ГОСТ 2.702-75 Правила выполнения электрических схем.