Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет

информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Дисциплина: Схемотехника

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту

на тему

АНАЛИЗАТОР ЗВУКОВОГО СИГНАЛА

БГУИР КП 1-40 02 01 406 ПЗ

Студент: группы 150504,   
Горбачевский К. В.

Руководитель:   
Калютчик А. А.

Минск 2023

Учреждение образования

«Белорусский государственный университет информатики   
и радиоэлектроники»

Факультет компьютерных систем и сетей

УТВЕРЖДАЮ

Заведующий кафедрой ЭВМ

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

(подпись)

«\_\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2023 г.

**З А Д А Н И Е**

**по курсовому проектированию**

Студенту Горбачевскому Кириллу Витальевичу\_\_\_\_\_\_\_\_\_\_\_\_

(фамилия, имя, отчество)

1. Тема проекта Анализатор звукового сигнала

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

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

2. Срок сдачи студентом законченного проекта с 06.12.2023 по 09.12.2023\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3. Исходные данные к проекту:

1. Микроконтроллер – тактовая частота не менее 20 кГц, не менее 10 входов/выходов.\_\_\_\_\_\_\_\_\_\_

2. Источник питания – напряжение 9 В, максимальный выходной ток не менее 3 А.\_\_\_\_\_\_\_\_\_\_\_

3. Модуль усиления – напряжение питания 7 – 36 В, коэфициент усиления не меньше 10. \_

4. Дисплей – напряжение питания 3.3-5 В, разрешение не менее 128x64.

4. Содержание расчетно-пояснительной записки (перечень подлежащих разработке вопросов)

Введение\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1.Обзор литературы.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2.Разработка структуры микропроцессорного анализатора звукового сигнала.\_\_\_\_\_

3.Обоснование выбора узлов, элементов функциональной схемы устройства.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4. Разработка принципиальной электрической схемы устройства.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5. Разработка программного обеспечения. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Заключение.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Литература.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

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

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

5. Перечень графического материала (с точным указанием обязательных чертежей)

1. Структурная схема анлизатора звукового сигнала устройства (формат А4)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2. Функциональная электрическая схема анлизатора звукового сигнала (формат А4)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3. Принципиальная электрическая схема анлизатора звукового сигнала (формат А4)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

6. Консультант по проекту (с назначением разделов проекта)\_A.A. Калютчик \_\_\_\_\_\_\_

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

7. Дата выдачи задания 10.09.2023\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

8. Календарный график работы над проектом на весь период проектирования (с назначением сроков исполнения и трудоемкости отдельных этапов):

разделы 1,2 к 24.09 – 20 %;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

раздел 3 к 15.10 – 20 %;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

раздел 4 к 05.11 – 25 %;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

раздел 5 к 19.11 – 20 %;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

оформление пояснительной записки и графического материала к 06.12 – 15 %;\_\_\_\_\_\_

защита курсового проекта с 07.12 по 14.12.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

РУКОВОДИТЕЛЬ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_асистент каф. ЭВМ Калютчик А.А.

(подпись)

Задание принял к исполнению 10.09.2023\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_К. В. Горбачевский

(дата и подпись студента)

**СОДЕРЖАНИЕ**

Оглавление

[ВВЕДЕНИЕ 4](#_Toc166416694)

[1 ОБЗОР КОМПОНЕНТОВ 5](#_Toc166416695)

[1.1 Состав устройства 5](#_Toc166416696)

[1.2 Микроконтроллеры 5](#_Toc166416697)

[1.3 Операционные усилители 6](#_Toc166416698)

[1.4 Гнездо и кабель 7](#_Toc166416699)

[1.5 Дисплей 7](#_Toc166416700)

[1.6 Питание 8](#_Toc166416701)

[2 РАЗРАБОТКА СТРУКТУРЫ УСТРОЙСТВА 9](#_Toc166416702)

[2.1 Постановка задачи 9](#_Toc166416703)

[2.2 Определение компонентов структуры устройства 9](#_Toc166416704)

[2.3 Взаимодействие компонентов устройства 9](#_Toc166416705)

[2.4 Готовое устройство и корпус 10](#_Toc166416706)

[3 ОБОСНОВАНИЕ ВЫБОРА УЗЛОВ, ЭЛЕМЕНТОВ ФУНКЦИОНАЛЬНОЙ СХЕМЫ УСТРОЙСТВА 12](#_Toc166416707)

[3.1 Обоснование выбора микроконтроллеров 12](#_Toc166416708)

[3.2 Обоснование выбора операционного усилителя 12](#_Toc166416709)

[3.3 Обоснование выбора кабеля и гнезда 12](#_Toc166416710)

[3.4 Обоснование выбора дисплея 13](#_Toc166416711)

[3.5 Обоснование выбора питания 14](#_Toc166416712)

[4 РАЗРАБОТКА ПРИНЦИПИАЛЬНОЙ ЭЛЕКТРИЧЕСКОЙ СХЕМЫ УСТРОЙСТВА 15](#_Toc166416713)

[4.1 Расчёт мощности элементов схемы 15](#_Toc166416714)

[4.2 Микроконтроллер 15](#_Toc166416715)

[5 РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ 16](#_Toc166416716)

[5.1 Требования к разработке программного обеспечения 16](#_Toc166416717)

[5.2 Блок-схема алгоритма 16](#_Toc166416718)

[5.3 Исходный код программы для анализатора звукокого сигнала 17](#_Toc166416719)

[ЗАКЛЮЧЕНИЕ 19](#_Toc166416720)

[ПРИЛОЖЕНИЕ А 21](#_Toc166416721)

[ПРИЛОЖЕНИЕ Б 22](#_Toc166416722)

[ПРИЛОЖЕНИЕ В 23](#_Toc166416723)

[ПРИЛОЖЕНИЕ Г 24](#_Toc166416724)

[ПРИЛОЖЕНИЕ Д 25](#_Toc166416725)

[ПРИЛОЖЕНИЕ Е 30](#_Toc166416726)

[ПРИЛОЖЕНИЕ Ж 31](#_Toc166416727)

# ВВЕДЕНИЕ

Темой данного курсового проекта является разработка устройства, анализирующего аналоговый сигнал на базе микроконтроллера.

Современная технологическая эпоха обогатила мир музыки и музыкантов инновационными средствами и инструментами для творчества. Однако, несмотря на множество цифровых достижений в области музыкального оборудования, аналоговые инструменты, такие как гитары, остаются популярными среди музыкантов разного уровня мастерства.

Важным аспектом игры на аналоговых инструментах, в том числе на гитаре, является настройка для достижения чистых и гармоничных звуков. В данном контексте представляется актуальной тема создания устройств, позволяющих музыкантам легко и точно настраивать свои инструменты.

Целью данного курсового проекта является разработка и реализация анализатора аналогового сигнала на базе микроконтроллера, предназначенного для настройки гитары.

Данное устройство может быть использовано не только для анализа сигнала исходящего от электрогитары. Прибор сможет определять частоту (тон) любого сигнала, передаваемого по кабелю Jack 6.3

В данной работе будет рассмотрен весь процесс разработки и реализации анализатора аналогового сигнала, начиная с выбора аппаратной платформы и заканчивая созданием программного обеспечения для обработки сигнала. Также будут рассмотрены различные методы анализа и алгоритмы, применяемые для определения настроенности струн инструмента.

В качестве основного вычислительного элемента устройства будет использована плата Arduino UNO. Выбор данной платы мотивирован тем, что она уже у меня была. Разработка программного обеспечения будет происходить в интегрированной среде разработки для Windows Arduino IDE 1.8.16. В данной среде есть все необходимое для написания программного обеспечения с последующей загрузки на плату.

Разработка курсового проекта будет происходить поэтапно. В первую очередь необходимо подобрать элементы устройства, учитывая их надежность, стоимость, функциональность и размеры. Затем необходимо собрать устройство и разработать программное обеспечение для корректной обработки информации и поддержания связи между элементами схемы.

# ОБЗОР КОМПОНЕНТОВ

## Состав устройства

Как сказано ранее, разрабатываемое микропроцессорное устройство выполняет функции анализа аналогового сигнала, а если быть точнее - измерения частоты аналогового сигнала и определения тона. Для решения этих задач в состав устройства должны входить:

- микроконтроллер

- усилитель

- гнездо для кабеля

- кабель

- дисплей для вывода информации

- питание

- корпус

## Микроконтроллеры

Существует огромное разнообразие плат с разными микроконтроллерами. Все они отличаются размерами, параметрами, предустановленными интерфейсами и выполняемыми задачами. Для сравнения была выбрана плата Arduino UNO и аналоги других производителей. Результаты сравнения приведены в таблице 1.1.

Таблица 1.1 — Сравнение микроконтроллеров

|  |  |  |  |
| --- | --- | --- | --- |
| Параметры сравнения | Arduino UNO | Raspberry Pi Zero | NodeMcu Lua |
| Микроконтроллер | ATmega328 | ARM Cortex-A53 | ESP8266 |
| Входное напряжение | 6 – 20 В | 6 – 28 В | 3.6 – 20 В |
| Флэш-память | 32 Кб | порт для microCD | 4 мб |
| ОЗУ | 2 Кб | 512 Мб | 20 Кб |
| Тактовая частота | 16 МГц | 1 ГГц | 80-160 МГц |
| Разрядность | 8 бит | 32 бит | 32 бит |
| Цифровые  входы/выходы | 14 шт | 26 шт | 11 шт |
| Аналоговые  входы/выходы | 6 шт | 0 шт | 1 шт |
| Выходное напряжение | 3.3В, 5 В | 3.3В, 5 В | 3.3В, 5 В |
| Рабочая температура | от -25 до +85 ºС | от -40 до +85 ºС | от -25 до +85 ºС |

Продолжение таблицы 1.1

Для получения более подробной информации о рассмотренных микроконтроллерах использовались источники [4, 5, 6].

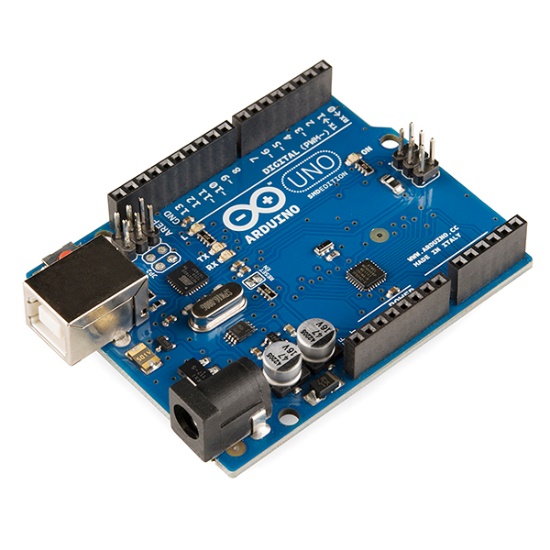


Рисунок 1.1 – Микроконтроллер Arduino Uno Rev3

## Операционные усилители

Сам по себе сигнал, исходящий из гитары, очень слабый и для анализа требует усиления.

Для данной схемы выбран операционный усилитель TL082, но существует куда больше усилителей, выполняющих разные задачи. Результаты сравнения приведены в таблице 1.2.

Таблица 1.2 — Сравнение усилителей

|  |  |  |  |
| --- | --- | --- | --- |
| Параметры сравнения | TL082 | OPA2134 | LT1803 |
| Коэффициент усиления | 86 | 104-120 | 50 |
| Полоса пропускания | 5 MHz | 8 MHz | 85 MHz |
| Рабочая температура | от -40 до +125 ºС | от -55 до +125 ºС | от -40 до +85 ºС |
| Кол-во каналов | 2 | 2 | 1 |

Для получения точной информации о данных датчиках использовалась техническая спецификация [8, 9, 10].

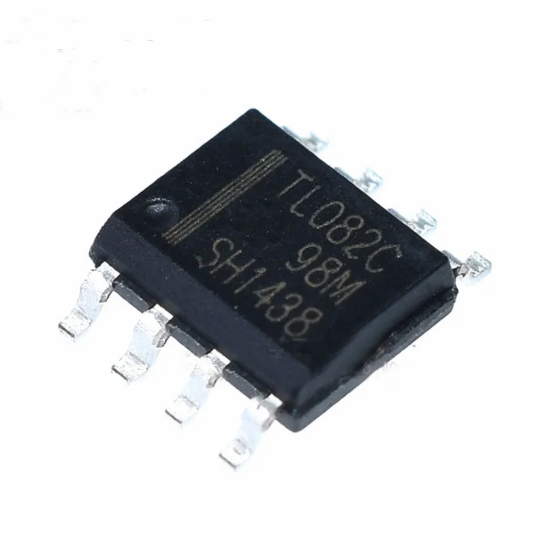


Рисунок 1.2 – Операционный усилитель TL082C

## Гнездо и кабель

Почти все профессиональное музыкальное оборудование взаимодействует между собой посредством кабелей Jack 6.3мм. Для устройства представленного в данной курсовой работе был выбран именно такой формат кабеля и гнезда. Существует два типа кабелей Jack 6.3: mono и stereo. В данной курсовой работе этот параметр не имеет значения так сигнал, выходящий из гитары является одноканальным.

a б

Рисунок 1.3 – Коммутация: а – Jack 6.3 кабель, б – Jack 6.3 гнездо

## Дисплей

Для того чтобы устройством было комфортно пользоваться и снимать данные в схеме предусмотрен небольшой I2C OLED дисплей с разрешением 128х64. OLED дисплей не требует для себя отдельной схемы питания. Его питание обеспечивается выходами GND и 5V микроконтроллера

Сигнал, обработанный программой в символы/числовые значения, будет передан на дисплей с выходов микроконтроллера A4 и A5 или SCL, SDA.

Дисплей поддерживает желтый и синий цвета.

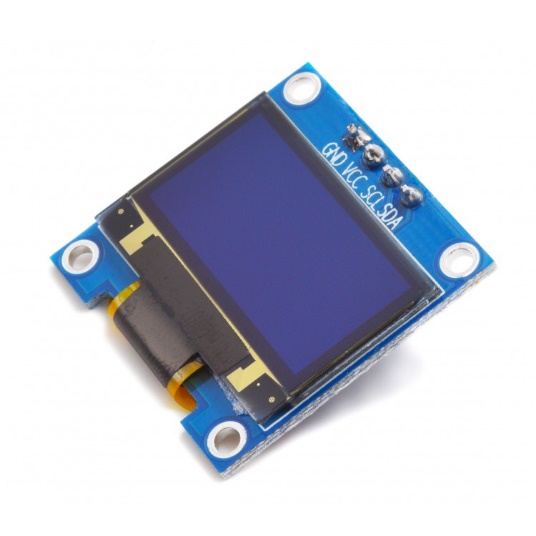


Рисунок 1.4 – OLED дисплей 128x64

## Питание

Для работы разрабатываемого устройства требуется питание. В данной схеме использовалась батарейк 6F22 (“Крона”). Модуль усиления требует для себя питание равное 18В, в том время как допустимое питание микроконтроллера колеблется от 7 до 12В. Для питания модуля усиления два элемента питания были соеденены параллельно. А для питания микроконтроллера Arduino Uno питание было отведено от одного элемента питания. Таким образом входное напряжение микроконтроллера равное 9В является приемлемым для работы данного устройства.

# РАЗРАБОТКА СТРУКТУРЫ УСТРОЙСТВА

## 2.1 Постановка задачи

Для того, чтобы составить структуру разрабатываемого устройства, необходимо выделить функции, которые будет выполнять устройство, затем определить компоненты и связь между ними исходя из данных функций. Результаты можно посмотреть на структурной схеме, представленной в приложении А.

В рамках данного курсового проекта необходимо разработать анализатор звукового сигнала. Для реализации было выбрано устройство, определяющее частоту сигнала (тон), исходящего от электрогитары и вывод этой информации на дисплей с выходов микроконтроллера. Вывод на дисплей должен отображать отклонение от ноты для точной настройки инструмента с погрешностью в 3Гц.

## 2.2 Определение компонентов структуры устройства

Компоненты структуры устройства выбираются исходя из функций, определенных в постановке задачи. Проанализировав выделенные функции, были определены следующие компоненты, представленные ниже.

1) Микроконтроллер — ключевой компонент всей схемы. Выполняет функцию обработки поступающей информации и формирует выходное значение.

2) Питание — источник постоянного напряжения 9V.

3) Модуль усиления сигнала – компонент схемы, усиливающий слабый аналаговый сигнал для дальнейшего анализа микроконтроллером.

4) Модуль отображения – компонент схемы, представляющий результат анализа звукового сигнала в понятной человеку форме (в виде цифр/букв/символов).

5) Корпус устройства – корпус для комфортного использования устройства, также выполняющий функцию защиты и инкапсуляции.

## 2.3 Взаимодействие компонентов устройства

При колебании струн электрогитары, электромагнитный сигнал поступает на вход модуля усиления по кабелю Jack 6.3. Усиленный сигнал с выхода модуля усиления анализируется микроконтроллером и значение частоты исходного сигнала выводятся на OLED дисплей в виде нотного обозначения.

Модуль питания взаимодействует со всеми элементами схемы напрямую или через модуль усиления, благодаря ему осуществляется питание всех необходимых элементов.

## 2.4 Готовое устройство и корпус

Для реализации курсового проекта были куплены все нужные элементы и детали для изготовления устройства. Модуль усиления размещен на макетной плате навесным способом и связан с микроконтроллер посредством нескольких проводов, главный из которых, A0, является анализируемым сигналом. Готовое устройство без корпуса представлено на рисунке 2.1

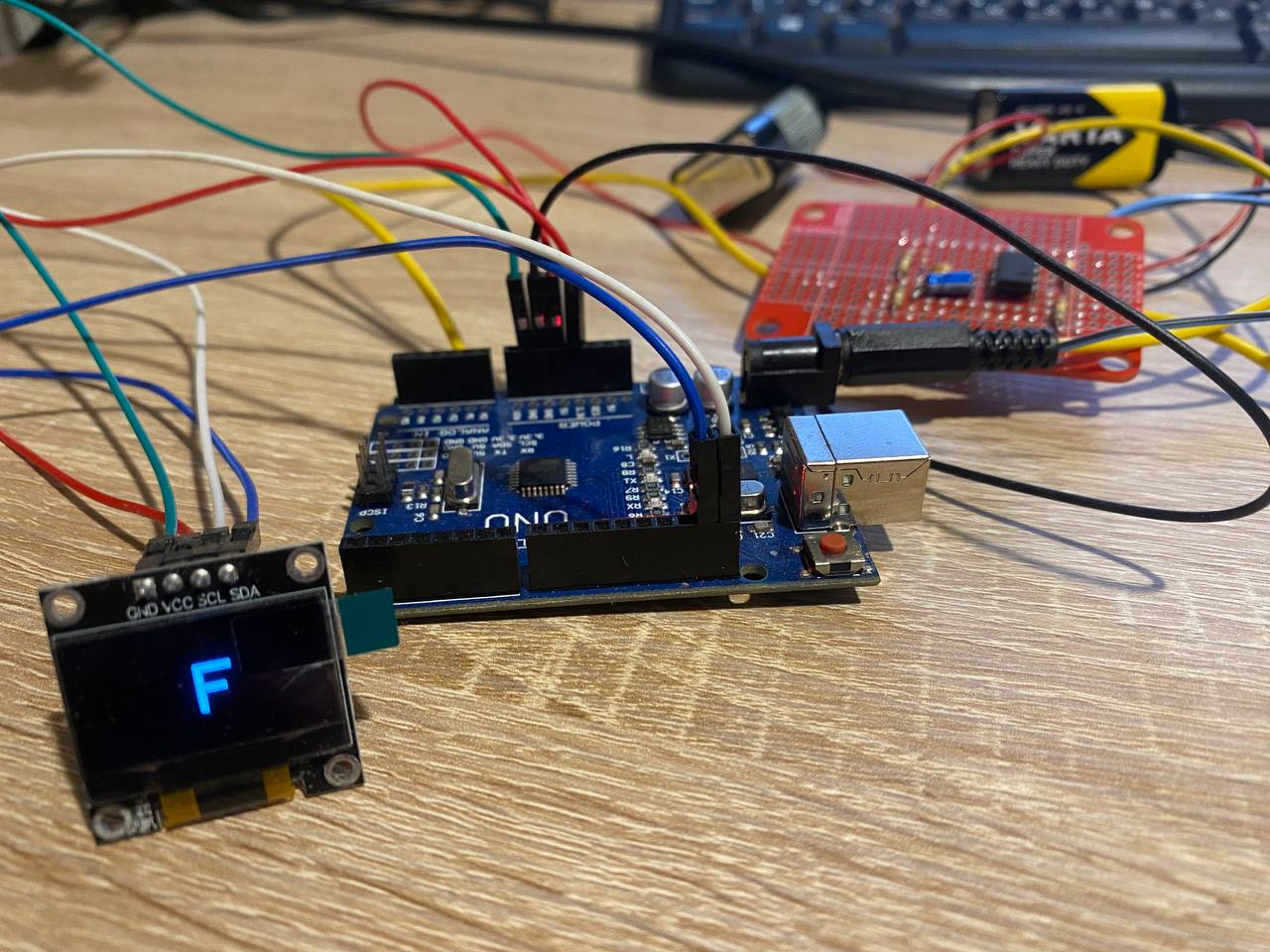


Рисунок 2.1 – Анализатор звукового сигнала

Для удобства использовани и защиты устройства был разработан полиуретановый корпус, имеющий один аудио-вход Jack 6.3 сбоку на корпусе, отверстие для дисплея на крышке, а также кнопку включения/выключения питания, которая также располагается на крышке. Корпус повышает мобильность устройства, удобство использования, а также гарантирует безопасность схемы при падении. Устройство в корпусе изображено на рисунке 2.2



Рисунок 2.1 – Анализатор звукового сигнала в корпусе

# ОБОСНОВАНИЕ ВЫБОРА УЗЛОВ, ЭЛЕМЕНТОВ ФУНКЦИОНАЛЬНОЙ СХЕМЫ УСТРОЙСТВА

## Обоснование выбора микроконтроллеров

В данной курсовом проекте в качестве контроллера могла быть использована любая плата из представленных в таблице 1.1 так как устройство не требует больших затрат в памяти и мощности для корректной работы.

Контроллер Raspberry PI Zero является чем-то большим чем просто микроконтроллером. Данный “мини-компьютер” имеет процессор на 1ГГц и 512мб оперативной памяти что в сотни раз больше чем аналогичные характеристики других микроконтроллеров. И именно по этой причине Raspberry PI Zero не подходит для решения относительно простой задачи анализа звукового сигнала и его пераметры будут излишни.

NodeMcu Lua с другой стороны самый близкий к Arduino контроллер, со встроенным WI-FI модулем. Он превосходит Arduino Uno по основным характеристикам и в разы меньше по размерам. Но для разработки полу-аналаговой схемы работать с таким контроллем не комфортно из-за малых габаритов и отсутствия отдельного входа питания, который в Arduino Uno присутсвует.

В связи со всеми вышеперечисленными причинами, был выбран контроллер Arduino Uno который уже имелся в наличии.

## Обоснование выбора операционного усилителя

Операционные усилители семейства TL являются самыми популярными и доступными операционными усилителями. Стоимость TL082 в несколько раз меньше стоимости OPA2134 при почти одинаковых характеристиках, а усилитель LT1803 хоть и обладает более лучшими характеристиками, тем не менее почти не встречается в продаже.

TL082 имеет подходящие параметры для использования при изготовлении устройства.

## Обоснование выбора кабеля и гнезда

Для данных компонентов почти не существует альтернатива. Электрогитара имеет выход Jack 6.3-mono. Кабель расчитан на прямое подключение к гитаре без использования переходников которые сильно влияют на шум сигнала. По таким соображение в качестве входной точки схемы было выбрано гнездно польностью сочетающиеся с кабелем – Jack 6.3-mono. Это стандартное гнездо, используемое во всех музыкальных приборах начиная от тюнеров и комбо-усилителей, заканчивая профессиональным студийным оборудованием, колонками и наушниками. Отличие стерео Jack от моно Jack изображено на рисунке 3.1.

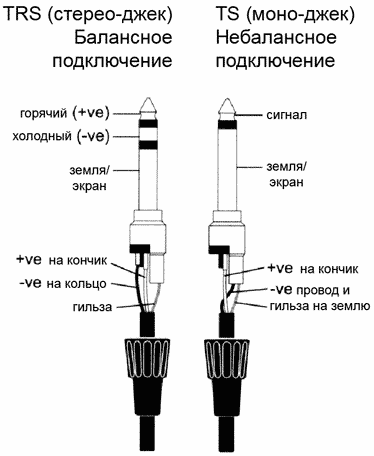


Рисунок 3.1 – Отличие mono и stereo Jack 1/4"

Стерео — двухканальный формат звука. В отличие от моно, стереосигнал передаёт информацию по двум независимым каналам: данные в левом и правом канале могут различаться.

Классическое подключение гитары подразумевает mono Jack, так как гитара зачастую записывается в mono и выход гитары это mono Jack.

## Обоснование выбора дисплея

В данной курсовой работе предусмотрен вывод результата анализа звукового сигнала на дисплей в качестве цифр, буква и символов. Для данной задачи был выбран OLED дисплей небольшого разрешения 128x64 пикселя. Стоимость OLED дисплея в несколько раз превышает стоимость ЖК-дисплея с аналогичным разрешением из-за различных типов матриц и способа передачи изображения. Но OLED дисплей обладает лучшими характеристиками цветопередачи и отзывчивости, а также прост в использовании. Данный дисплей может работать при питании от 3.3 до 5.5В, данное питание могут обеспечить выходы микроконтроллера Arduino UNO R3

## Обоснование выбора питания

Для данной схемы предусмотрено питание от двух батареек 6F22 (“Крона”) отдельно для микроконтроллера и схемы с усилителем. При использовании батареек прибор становиться мобильным и появляется возможность его использования без подключения к сети постоянного напряжения. Микроконтроллер Arduino UNO не требует для себя питания конкретного значения питания. Примлемое входное напряжение Arduino UNO R3 находится в диапазоне он 5 до 12В, что обеспечивает долговечность службы микроконтроллера, но подаче на микроконтроллер даже 20В он все еще будет работоспособен. Всвязи со всеми вышеперечисленными причинами, в качестве источника питания были выбраны батарейки типа 6F22 (“Крона”), обеспечивающие достаточное напряжение как для схемы с усилением, так и для микроконтроллера.

# РАЗРАБОТКА ПРИНЦИПИАЛЬНОЙ ЭЛЕКТРИЧЕСКОЙ СХЕМЫ УСТРОЙСТВА

## Расчёт мощности элементов схемы

Потребляемая мощность разрабатываемого устройства равна сумме мощностей, потребляемых его элементами. Расчет мощности элементов схемы устройства управления и самого устройства представлены в таблице 4.1 и 4.2 соответственно.

Таблица 4.1 – Расчет мощности элементов схемы устройства

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Блок | U, В | I, мА | Кол-во | P, мВт |
| Микроконтроллер Arduino UNO R3 | 5 | 22 | 1 | 110 |
| Модуль усиления | 18 | 28 | 1 | 504 |
| Модуль отображения | 3.3-5 | 16 | 1 | 80 |
| Суммарная мощность, мВт | | | | 694 |

В реализованной схеме главными компонентами являются: микроконтроллер Arduino UNO R3, модуль усиления на базе операционного усилителя TL082 и OLED дисплей 0.96 с разрешением 128х64.

Таким образом потребляемая мощность будет равна:

Р = 5 ∙ 22 + 18 ∙ 28 + 5 ∙ 16 = 694 мВт.

Учитывая поправочный коэффициент в 20%, максимальная потребляемая мощность составит 832 мВт.

Рассчитаем потребляемый ток:

## 4.2 Микроконтроллер

Информация о выбранном микроконтроллере Arduino UNO представлена в пункте 3.1 раздела 3.

В схеме с устройством к аналоговому входу A0 подключен выход с модуля усиления, который представляет собой аналоговый сигнал для анализа. Дисплей подключается к пинам GND, 5V, A4, A5 микроконтроллера

Данный микроконтроллер питается от напряжения 9 B.

# РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

## Требования к разработке программного обеспечения

Разработанное микропроцессорное устройство анализирует аналаговый усиленный сигнал, исходящий от гитары. Под анализом сигнала в данном проекте подразумевается определение частоты сигнала, а затем определение ноты.

Устройство работает следующим образом. В линейный вход (гнездо Jack 6.3) вставляется шнур, который соединяет выход гитары с входом модуля усиления. На устройство подается питание путем подключения двух батареек типа “Крона”. Устройство готово к работе. При взаимодействии со струнами гитары, возникает звуковая волна, которая считывается магнитными звукоснимателя внутри гитары и звуковой сигнал передается далее по кабелю.

Так как звуковой сигнал от гитары очень слыбый, ведь для работы электро-гитары не требуется питание, то для анализа этого сигнала его нужно усилить. Сигнал усиливается в схеме с операционным усилителем TL082 и делее подается на вход A0 микроконтроллера Arduino.

## Блок-схема алгоритма

Блок-схема — это схематичное представление процесса, системы или компьютерного алгоритма. Блок-схемы часто применяются в разных сферах деятельности, чтобы документировать, изучать, планировать, совершенствовать и объяснять сложные процессы с помощью простых логичных диаграмм.

Рассмотрим блок-схему алгоритма программного обеспечения данного курсового проекта, представленную в приложении Г.

Исполняемая программа работает на основе прерываний поддерживающихся по умолчанию в микроконтроллере Arduino UNO R3. Прерывание срабатывает при изменении уровня сигнала на входе A0 микроконтроллера. На блок-схеме этот шаг описывает блок 1 в виде условия.

Если уровень сигнала не изменился программа входит в режим ожидания и на экране дисплея отображается предыдущий уровень сигнала. В противном случае запускается функция прерывания, которая высчитывает новое значение частоты сигнала, данная функция отображена блоком 2 на блок-схеме. Если частота сигнала со входа A0 совпадает с частотой ноты приведенной в таблице 5.1 с точностью до 3Гц, следует блок 3 – отображения информации на экран дисплея, а затем конец программы.

Если частота сигнала со входа A0 отличается от частоты ноты из таблицы более чем на 3Гц, на дисплей помимо звучащей ноты, выводится ее отклонение в виде стрелки вверх или стрелки вниз для более точной настройки инструмента.

## Исходный код программы для анализатора звукокого сигнала

Данное устройство является основным в разрабатываемом курсовом проекте и здесь реализована большая часть логики. Исходный код программного обеспечение можно найти в приложении Д

Функция void setup() функция инициализации основный переменных, используемых в программе. Также в данном функции устанавливается начальное значение дисплея, сдвиг курсора, цвет текста и другие параметры.

Функция void loop()является главной, здесь раз в 70 миллисекунд вызывается функция считывания сигнала с порта A0

Функция void printNote() функция реализует вывод на экран I2C OLED дисплея переданной информации. В данном случае это буквенное обозначение ноты (E, A, A#). Данная функция очищает дисплей от предыдущего значения, устанавливает новую позицию курсора и выводит на экран новое значение. При таком алгоритме, если сигнал на входе A0 микроконтроллера не изменяется на экране дисплея будет постоянно отображаться последняя сыгранная ноту, до тех пор, пока сигнал не изменится.

Функция String getGuitarNoteByFrequency() реализует логику преобразования частотного значения аналогового сигнала, считанного со входа A0 микроконтроллера. Функция принимает одно значения – частоту сигнала, и по этой частоте определяет ноту. Возвращаемое значение функции – буквенное обозначение ноты.

Точность определения ноты составляет 3Гц. Если частота входного сигнала и частоты ноты по таблице различаются больше чем на 3Гц, на экран дисплея вместе с нотой будет показано ее отклонение в виде стрелки. Если частота ноты отличается больше чем на 10Гц, будет выведен полутон. Нота, лежащяя между извлекаемой и следующей в октаве. Например (A#)

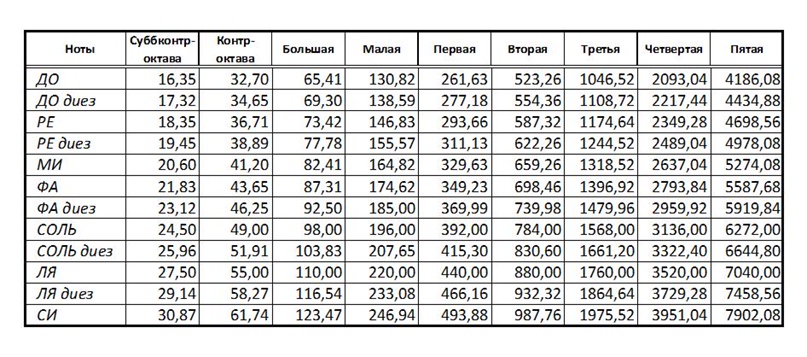
Все ноты имеют свою частоту. Частоты нот представлены на рисунке 5.1

Рисунок 5.1 – Частота звучания музыкальных нот

Функция void ISR()- функция прерывания микроконтроллера Arduino Uno R3. Данная функция осуществляет основную логику программы. При изменении значения на входе A0 микроконтроллера высчитывается новое значение частоты сигнала, которое далее передается в функцию printNote.

Если значение на входе не изменяется, функция loop ничего не выполняет.

# ЗАКЛЮЧЕНИЕ

В результате работы над данным курсовым проектом было разработано работоспособное микропроцессорное устройство со своим программным обеспечением. Устройство анализирует аналоговый сигнал, передающийся от гитары к модулю усиления и далее на микроконтроллер, где осуществляется обработка этого сигнала и вывод информации на дисплей в виде обозначения ноты. Данный проект был спроектирован в соответствии с поставленными задачами, весь функционал был реализован в полном объеме.

Разработанное микропроцессорное устройство обладает следующими достоинствами: низкая стоимость компонентов, простота реализации и сборки. Недостатки данного устройства: сильная энергозависимость, необходимость подключения гитары через кабель.

В дальнейшем планируется усовершенствование данного курсового проекта. Одним из таких улучшений является оптимизация алгоритма анализа полученных данных, улучшение питания, а также создание более дружественного интерфейса.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

[1]. Вычислительные машины, системы и сети: дипломное проектирование (методическое пособие) [Электронный ресурс] : Минск БГУИР 2019. – Электронные данные. – Режим доступа: <https://www.bsuir.by/m/12_100229_1_136308.pdf>

[2]. Документация Arduino [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://docs.arduino.cc/>

[3]. Геддес, М. 25 крутых проектов с Arduino / М. Геддес ; [пер. с англ. М. А. Райтмана]. – Москва : Эксмо, 2019. – 272 c.

[4]. Arduino UNO [Электронный ресурс]. – Электронные данные. – Режим доступа: http://arduino.ru/Hardware/ArduinoBoardUno – Дата доступа: 11.09.2023

[5]. TL082 Datasheet [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://www.alldatasheet.com/view.jsp?Searchword=Tl082%20datasheet> – Дата доступа: 09.10.2023

[6]. OLED дисплеи [Электронный ресурс]. – Электронные данные. – Режим доступа: https://amperkot.by/page/amperkot-oled-displays/ – Дата доступа: 17.10.2023

[7]. Аналоговая микросхемотехника [Электронный ресурс]: Минск БГУИР 2003. – Электронные данные. – Режим доступа: <https://www.bsuir.by/m/12_100229_1_85477.pdf> – Дата доступа: 12.10.2023

[8]. Программирование Arduino [Электронный ресурс] - Электронные данные. – Режим доступа: <https://arduino.ru/Reference> – Дата доступа: 1.11.2023

[9]. Пайка своими руками: основы для начинающих - [Электронный ресурс] - Электронные данные. – Режим доступа: https://vopros-remont.ru/elektrika/pajka/ – Дата доступа: 4.11.2023

# ПРИЛОЖЕНИЕ А

(обязательное)

**Схема структурная**

# ПРИЛОЖЕНИЕ Б

(обязательное)

**Схема функциональная**

# ПРИЛОЖЕНИЕ В

(обязательное)

**Схема принципиальная**

# ПРИЛОЖЕНИЕ Г

(обязательное)

**Блок-схема программы**

# ПРИЛОЖЕНИЕ Д

(обязательное)

**Листинг кода**

#include <SPI.h>

#include <Wire.h>

#include <Adafruit\_GFX.h>

#include <Adafruit\_SSD1306.h>

#define SCREEN\_WIDTH 128 // OLED display width, in pixels

#define SCREEN\_HEIGHT 64 // OLED display height, in pixels

Adafruit\_SSD1306 display(SCREEN\_WIDTH, SCREEN\_HEIGHT, &Wire, -1);

//clipping indicator variables

boolean clipping = 0;

//data storage variables

byte newData = 0;

byte prevData = 0;

unsigned int time = 0;//keeps time and sends vales to store in timer[] occasionally

int timer[10];//storage for timing of events

int slope[10];//storage for slope of events

unsigned int totalTimer;//used to calculate period

unsigned int period;//storage for period of wave

byte index = 0;//current storage index

float frequency;//storage for frequency calculations

int maxSlope = 0;//used to calculate max slope as trigger point

int newSlope;//storage for incoming slope data

//variables for decided whether you have a match

byte noMatch = 0;//counts how many non-matches you've received to reset variables if it's been too long

byte slopeTol = 3;//slope tolerance- adjust this if you need

int timerTol = 10;//timer tolerance- adjust this if you need

//variables for amp detection

unsigned int ampTimer = 0;

byte maxAmp = 0;

byte checkMaxAmp;

byte ampThreshold = 5;//raise if you have a very noisy signal

void setup(){

if(!display.begin(SSD1306\_SWITCHCAPVCC, 0x3C)) { // Address 0x3D for 128x64

Serial.println(F("SSD1306 allocation failed"));

for(;;);

}

delay(500);

display.setTextSize(4);

display.setTextColor(WHITE);

// Display static text

Serial.begin(9600);

pinMode(13,OUTPUT);//led indicator pin

pinMode(12,OUTPUT);//output pin

cli();//diable interrupts

//set up continuous sampling of analog pin 0 at 38.5kHz

//clear ADCSRA and ADCSRB registers

ADCSRA = 0;

ADCSRB = 0;

ADMUX |= (1 << REFS0); //set reference voltage

ADMUX |= (1 << ADLAR); //left align the ADC value- so we can read highest 8 bits from ADCH register only

ADCSRA |= (1 << ADPS2) | (1 << ADPS0); //set ADC clock with 32 prescaler- 16mHz/32=500kHz

ADCSRA |= (1 << ADATE); //enabble auto trigger

ADCSRA |= (1 << ADIE); //enable interrupts when measurement complete

ADCSRA |= (1 << ADEN); //enable ADC

ADCSRA |= (1 << ADSC); //start ADC measurements

sei();//enable interrupts

}

ISR(ADC\_vect) {//when new ADC value ready

PORTB &= B11101111;//set pin 12 low

prevData = newData;//store previous value

newData = ADCH;//get value from A0

if (prevData < 127 && newData >=127){//if increasing and crossing midpoint

newSlope = newData - prevData;//calculate slope

if (abs(newSlope-maxSlope)<slopeTol){//if slopes are ==

//record new data and reset time

slope[index] = newSlope;

timer[index] = time;

time = 0;

if (index == 0){//new max slope just reset

PORTB |= B00010000;//set pin 12 high

noMatch = 0;

index++;//increment index

}

else if (abs(timer[0]-timer[index])<timerTol && abs(slope[0]-newSlope)<slopeTol){//if timer duration and slopes match

//sum timer values

totalTimer = 0;

for (byte i=0;i<index;i++){

totalTimer+=timer[i];

}

period = totalTimer;//set period

//reset new zero index values to compare with

timer[0] = timer[index];

slope[0] = slope[index];

index = 1;//set index to 1

PORTB |= B00010000;//set pin 12 high

noMatch = 0;

}

else{//crossing midpoint but not match

index++;//increment index

if (index > 9){

reset();

}

}

}

else if (newSlope>maxSlope){//if new slope is much larger than max slope

maxSlope = newSlope;

time = 0;//reset clock

noMatch = 0;

index = 0;//reset index

}

else{//slope not steep enough

noMatch++;//increment no match counter

if (noMatch>9){

reset();

}

}

}

if (newData == 0 || newData == 1023){//if clipping

clipping = 1;//currently clipping

//Serial.println("clipping");

}

time++;//increment timer at rate of 38.5kHz

ampTimer++;//increment amplitude timer

if (abs(127-ADCH)>maxAmp){

maxAmp = abs(127-ADCH);

}

if (ampTimer==1000){

ampTimer = 0;

checkMaxAmp = maxAmp;

maxAmp = 0;

}

}

void reset(){//clean out some variables

index = 0;//reset index

noMatch = 0;//reset match couner

maxSlope = 0;//reset slope

}

void checkClipping(){//manage clipping indication

if (clipping){//if currently clipping

clipping = 0;

}

}

double openStringFrequencies[] = {82.41, 110.00, 146.83, 196.00, 246.94, 329.63};

String stringNames[] = {"E", "A", "D", "G", "B", "E"};

bool in\_between(double a, double min, double max)

{

return a >= min && a <= max;

}

String semi\_ton(String note)

{

String semiton;

if(note == "E") {

semiton = "F";

}

else if(note == "B") {

semiton = "C";

}

else {

semiton = note + "#";

}

return semiton;

}

String note = "";

String getGuitarNoteByFrequency(double frequency) {

double accuracyTone = 1;

double accuracyDemiton = 3;

// Determine the note based on the provided frequency.

for (int i = 0; i < 6; i++) {

if(in\_between(frequency, openStringFrequencies[i] - accuracyTone, openStringFrequencies[i] + accuracyTone))

note = stringNames[i];

else if (in\_between(frequency, openStringFrequencies[i] + accuracyTone, openStringFrequencies[i] + accuracyDemiton))

note = stringNames[i] + "+";

else if (in\_between(frequency, openStringFrequencies[i] + accuracyDemiton, openStringFrequencies[i] + 2\*accuracyDemiton))

note = semi\_ton(stringNames[i]);

else if (in\_between(frequency, openStringFrequencies[i] - accuracyTone - accuracyDemiton, openStringFrequencies[i] - accuracyTone))

note = stringNames[i] + "-";

}

return note;

}

void printNote(float frequency) {

String note = getGuitarNoteByFrequency(frequency);

display.setCursor(60, 20);

display.clearDisplay();

display.print(note);

display.display();

}

void loop(){

checkClipping();

if (checkMaxAmp > ampThreshold){

frequency = 38462/float(period);//calculate frequency timer rate/period

//print results

Serial.print(frequency);

Serial.println(" hz");

printNote(frequency);

}

delay(150);

}

# ПРИЛОЖЕНИЕ Е

(обязательное)

**Перечень элементов**

# ПРИЛОЖЕНИЕ Ж

(обязательное)

**Ведомость документов**