

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

**Учреждение образования  
«Гомельский государственный университет  
имени Франциска Скорины»**

Факультет физики и информационных технологий  
Кафедра общей физики

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА  
к курсовому проекту  
на тему**

**МОБИЛЬНЫЙ КЛИЕНТ-ПРОИГРЫВАТЕЛЬ АУДИО ФАЙЛОВ НА  
БАЗЕ ETHERNET**

ГГУ КП 1-39 03 02 027 ПЗ

Исполнитель:

Студент группы МС-32:

\_\_\_\_\_

Бондарчук А.Д.

Научный руководитель:

Старший преподаватель

Кафедры общей физики

\_\_\_\_\_

Подалов М. А.

Гомель 2023

## РЕФЕРАТ

Курсовой проект: 56 страниц, 3 главы, 24 рисунка, 16 источников.

### МОБИЛЬНЫЙ КЛИЕНТ-ПРОИГРЫВАТЕЛЬ, ESP32, УСИЛИТЕЛЬ, АУДИОФАЙЛЫ, ETHERNET

Предметом проектирования является мобильный клиент-проигрыватель аудио файлов на базе Ethernet.

**Цель работы:** освоить принципы работы клиент-проигрывателя аудио файлов на базе Ethernet, так же спроектировать простой и мобильный клиент-проигрыватель аудио файлов, работающий от общей электросети, и управлять удалено с помощью технологии Ethernet.

В процессе выполнения курсового проекта были решены следующие задачи:

- Выполнен анализ предметной области;
- Подобраны необходимые компоненты для клиент-проигрывателя;
- Разработана структурная электрическая схема клиент-проигрывателя;
- Разработана принципиальная электрическая схема проигрывателя;
- Разработан алгоритм функционирования клиент-проигрывателя;
- Разработан web-сервер для клиент-проигрывателя.

# СОДЕРЖАНИЕ

Введение.....	5
1 Анализ предметной области.....	6
1.1 Разбор аппаратных платформ.....	8
1.1.1 Платформа Arduino.....	8
1.1.2 Платформа ESP32.....	9
1.1.3 Платформа Raspberry Pi.....	10
1.1.4 Выводы и сравнение аппаратных платформ.....	11
1.2 Разбор модулей Ethernet.....	14
1.2.1 Классификация модулей Ethernet.....	14
1.2.2 Выводы и сравнение модулей Ethernet.....	15
1.3. Обзор Audio усилителей.....	16
1.3.1 Класс усилителя А.....	17
1.3.2 Класс усилителя АВ.....	18
1.3.3 Класс усилителя В.....	18
1.3.4 Класс усилителя D.....	19
1.3.5 Классы усилителя G, H, E, F.....	20
1.4 Используемые языки программирования.....	23
1.4.1 Среда разработки (IDE).....	23
1.4.2 Языки программирования.....	24
2 Разработка электрических схем мобильного клиент-проигрывателя аудиофайлов на базе Ethernet.....	29
2.1 Разработка структурной схемы мобильного клиент-проигрывателя аудио файлов на базе Ethernet.....	29
2.2.1 Обоснование базовых блоков структурной электрической схемы мобильного клиент-проигрывателя аудио файлов.....	29
2.2 Разработка принципиальной схемы мобильного клиент-проигрывателя аудио файлов на базе Ethernet.....	34
2.2.1 Обоснование выбора САПР для Разработки принципиальной электрической схемы мобильного клиент-проигрывателя аудио файлов на базе Ethernet.....	34
2.2.2 Описание используемых модулей и библиотечных элементов.....	37
3 Разработка алгоритма и функционирования мобильного клиент-проигрывателя аудио файлов на базе Ethernet.....	40
3.1 Описание алгоритма функционирования мобильного клиент-проигрывателя аудио файлов на базе Ethernet.....	40

3.2 Выбор локального веб-сервера .....	40
3.3 Разработка веб-сервера “MusCloud” .....	44
3.4 Исползования библиотек для ESP32 .....	46
3.4.1 Описание библиотеки Ethernet.h .....	46
3.4.2 Описание библиотеки SPI.h .....	47
3.4.3 Описание библиотеки SPIFFS.h .....	47
3.4.4 Описание библиотеки SPIFFS.h .....	48
Заключение .....	49
ПРИЛОЖЕНИИ А .....	52
Структурная электрическая схема мобильного клиент-мобильного .....	52
ПРИЛОЖЕНИЕ Б .....	53
Принципиальная электрическая схема электрического схема мобильного проигрывателя аудиофайлов .....	53
ПРИЛОЖЕНИЕ В .....	54
Блок-схема алгоритма работы мобильного клиент проигрывателя аудиофайлов на базе Ethernet .....	54
Спецификация курсового проекта .....	55
Ведомость к курсовому проекту .....	56

## Введение

В прошлом клиент-проигрыватели аудио файлов имели большую популярность среди широкого круга людей. Возвращаясь в наше время спрос на такие устройства сильно спал, но не из-за их потребности, а из-за их громоздкости, условий для работы таких клиент-проигрывателей. Сейчас они стали раритетом и желанным предметом для коллекционеров. На их место пришли мобильные клиент-проигрыватели из-за своих удобств. Современные клиент-проигрыватели есть у множества людей.

Мобильные клиент-проигрыватели, такие как музыкальные приложения, предлагают удобный способ слушать музыку на смартфонах и планшетах. Однако у них также могут возникать некоторые проблемы. Вот несколько распространенных проблематик, связанных с мобильными клиент-проигрывателями:

- **Качество звука:** Некоторые пользователи могут жаловаться на низкое качество звука в мобильных клиент-проигрывателях. Это может быть связано с ограничениями аудиокодеков, настройками устройства или с самим приложением. Пользователи хотят наслаждаться высококачественным звуком, поэтому проблемы с качеством могут быть раздражающими;
- **Отсутствие функциональности:** Некоторые мобильные клиент-проигрыватели могут не предлагать все функции, которые пользователи могут ожидать. Например, отсутствие возможности создавать плейлисты, неполные метаданные о треках или отсутствие возможности стриминга музыки могут быть разочаровывающими для пользователей.
- **Неудобный пользовательский интерфейс:** плохо разработанный пользовательский интерфейс может создавать проблемы для пользователей. Сложность в навигации по приложению, запутанность настроек и отсутствие интуитивного интерфейса могут вызывать раздражение и затруднять использование приложения;
- **Реклама и ограничения:** Многие мобильные клиент-проигрыватели предлагают бесплатные версии со встроенной рекламой или ограниченными функциями. Некоторые пользователи могут считать, что эти ограничения слишком сильные или реклама слишком навязчивая, что может повлиять на их удовлетворение от использования приложения.

## **1 Анализ предметной области**

Мобильный клиент-проигрыватель аудио файлов на базе Ethernet – это устройство или приложение, которое позволяет воспроизводить аудиофайлы на мобильных устройствах с использованием технологии Ethernet.

Такой клиент-проигрыватель может быть встроенным в устройство (например, портативный аудиоплеер) или представлять собой мобильное приложение, устанавливаемое на смартфоны или планшеты.

Так же он может поддерживать различные форматы аудиофайлов, такие как MP3, FLAC, AAC и другие, и предлагать функции управления воспроизведением, такие как пауза, перемотка, создание плейлистов и т.д. Он также может интегрироваться с другими функциями устройства, например, позволять управлять воспроизведением через наушники с пультом управления или интегрироваться с другими приложениями для обмена и синхронизации аудиофайлов.

Для корректной работы его нам потребуется определённые комплектующие, такие как: аппаратная платформа, модуль Ethernet, усилитель аудио файлов и динамик.

Разберем, что для чего используется:

Аппаратная платформа – это совокупность аппаратных компонентов, которые составляют основу компьютерной или электронной системы. Она включает в себя процессоры, память, входные-выходные порты, шины данных, устройства хранения данных, графические процессоры и другие компоненты, которые работают вместе для обеспечения функциональности и производительности системы.

Она может быть представлена различными уровнями сложности и масштабов. Например, для мобильных устройств это может быть система на кристалле (System-on-Chip, SoC), включающая процессор, графический процессор, контроллеры памяти, сетевые контроллеры и другие интегрированные компоненты.

Для компьютеров аппаратная платформа включает в себя материнскую плату, процессор, оперативную память, жесткий диск или твердотельный накопитель, видеокарту, звуковую карту, порты ввода-вывода и другие расширительные карты.

Также определяет возможности и ограничения системы, включая ее производительность, совместимость с программным обеспечением и поддержку различных функций и стандартов.

Примеры известных аппаратных платформ включают Intel x86 для компьютеров, ARM для мобильных устройств и NVIDIA Tegra для игровых консолей.

Модуль Ethernet — это физическое устройство или компонент, который добавляет функциональность Ethernet-соединения к другому устройству или системе. Он обычно представляет собой отдельную плату или модуль, который может быть установлен в разъем или слот на основной плате или другом устройстве.

Он выполняет функции, связанные с передачей данных по сети Ethernet. Он обеспечивает интерфейс для подключения к сети Ethernet и может поддерживать различные стандарты и скорости передачи данных, такие как 10/100/1000 Mbps или даже более высокие. Модуль Ethernet может включать физический разъем (например, RJ-45) для подключения сетевого кабеля.

Широко применяются в различных устройствах и системах, включая компьютеры, маршрутизаторы, коммутаторы, сетевые хранилища, промышленные устройства и другие. Они позволяют этим устройствам подключаться к локальным сетям или Интернету через Ethernet-инфраструктуру и обмениваться данными с другими устройствами в сети.

Модули Ethernet могут поддерживать различные функции, такие как автоматическое определение скорости соединения (auto-negotiation), поддержка виртуальных локальных сетей (VLAN), обработка сетевых протоколов, таких как TCP/IP, и технологии безопасности, такие как шифрование и аутентификация.

В некоторых случаях модуль Ethernet может быть встроенным или интегрированным в устройство, например, на материнской плате компьютера или в сетевом контроллере микроконтроллера.

Усилитель звука — это электронное устройство, которое увеличивает амплитуду звукового сигнала, усиливая его мощность и громкость. Он принимает слабый входной аудио сигнал и увеличивает его силу до уровня, достаточного для приведения в движение динамиков или других аудио устройств.

Он широко применяется в аудио системах, включая домашние кинотеатры, стереосистемы, наушники, музыкальные инструменты и профессиональное аудио оборудование. Они обеспечивают усиление аудио сигнала, чтобы он мог быть воспроизведен с достаточной громкостью и ясностью.

Состоят из нескольких основных компонентов, включая усилительные блоки, предусилители, блоки управления и выходные каскады. Входной сигнал проходит через усилительные блоки, которые увеличивают амплитуду сигнала. Предусилитель может также регулировать уровень сигнала и

применять различные эффекты и настройки. Блоки управления позволяют пользователю контролировать параметры звука, такие как баланс, тембр и громкость. Выходные каскады приводят усиленный сигнал к выходным устройствам, таким как динамики или наушники.

Также могут иметь различные характеристики и спецификации, такие как мощность, импеданс, частотный диапазон и коэффициент гармоник искажений. Они могут быть аналоговыми или цифровыми, и различаться по типу и классу работы. Выбор усилителя звука зависит от конкретных потребностей аудио системы и предпочтений пользователя.

Динамик — это устройство, которое преобразует электрический сигнал в звуковые колебания. Он является основным компонентом аудио системы и предназначен для воспроизведения звуковых сигналов.

Он состоит из нескольких ключевых элементов, включая динамический диффузор, катушку голосовой катушки, магнитную систему и корзину. Когда проходит электрический сигнал через голосовую катушку, создается переменное магнитное поле, которое взаимодействует с постоянным магнитом, расположенным в магнитной системе. Это вызывает колебания динамического диффузора, который передает акустические волны в окружающую среду и создает звуковое давление, известное как звук.

Динамики могут различаться по размеру, конструкции и характеристикам, таким как диапазон частот, мощность и чувствительность. Они могут быть одиночными или использоваться в парах или множестве для создания стереоэффекта или многоканального звучания.

Динамики широко применяются в различных устройствах и системах, включая аудио системы домашнего кинотеатра, автомобильные аудиосистемы, портативные колонки, наушники, профессиональное звуковое оборудование и т.д. Они воспроизводят различные звуковые частоты и обеспечивают звуковое воспроизведение для наслаждения музыкой, фильмами, играми и другими аудио контентом.

## **1.1 Разбор аппаратных платформ**

### **1.1.1 Платформа Arduino**

Arduino — это открытая платформа для разработки электронных устройств, основанная на микроконтроллерах. Она включает в себя аппаратную часть (микроконтроллер и периферийные устройства) и программное обеспечение (среду разработки и библиотеки) [1].



Она позволяет создавать различные электронные устройства, такие как роботы, контроллеры света, звука, движения и т.д. С помощью Arduino можно подключать различные датчики и управлять разными устройствами.

И имеет открытый исходный код, что позволяет пользователям создавать свои собственные проекты на базе платформы, а также делиться своими идеями и опытом с другими участниками сообщества. Благодаря простоте использования и доступности, Arduino стала очень популярной среди любителей электроники и программирования, а также в образовательных целях.



Рисунок 1.1.1 - Arduino UNO

### **1.1.2 Платформа ESP32**

ESP32 — это микроконтроллер, разработанный компанией Espressif Systems. Он является продолжением серии микроконтроллеров ESP8266 и имеет более широкие возможности и функции [2].

Основные характеристики ESP32 включают в себя двухъядерный процессор Xtensa LX6 с тактовой частотой до 240 МГц, поддержку беспроводной связи Wi-Fi и Bluetooth, а также большое количество периферийных интерфейсов, включая UART, I2C, SPI, ADC, DAC, PWM и другие.

Также поддерживает различные операционные системы, такие как FreeRTOS и Linux, что делает его очень гибким и мощным микроконтроллером для множества приложений, от простых IoT-устройств до сложных проектов с большим объемом данных и вычислений.

В целом, ESP32 является универсальным и высокопроизводительным микроконтроллером, который обладает множеством функций и широкими возможностями, что делает его популярным выбором для различных проектов, особенно в области IoT и мобильных приложений.



Рисунок 1.1.2 – ESP32-WROOM-32

### 1.1.3 Платформа Raspberry Pi

Raspberry Pi — это небольшой одноплатный компьютер, созданный в Великобритании Фондом Raspberry Pi. Он был разработан для обучения программированию, электронике и компьютерной науке, но также может использоваться для различных проектов, таких как умный дом, медиацентр, сервер и многое другое [3].

И имеет несколько моделей с разными характеристиками и ценами, начиная от простой модели Raspberry Pi Zero, который стоит около \$5, до более мощных моделей Raspberry Pi 4, которые могут стоить до \$75.

Каждая модель Raspberry Pi имеет процессор ARM, слот для карт памяти, порты USB, Ethernet, HDMI и аудио, а также GPIO (общий ввод/вывод), который позволяет подключать различные устройства и датчики.

Она работает на операционной системе Linux, и доступны различные дистрибутивы, такие как Raspbian, Ubuntu, и другие. Он также может работать на операционной системе Windows 10 IoT Core.

Raspberry Pi широко используется в образовании и в научно-исследовательских целях, а также в качестве бюджетного компьютера для проектов, где не требуется высокая производительность.



Рисунок 1.1.3 – Raspberry Pi

### **1.1.4 Выводы и сравнение аппаратных платформ**

Имея исходные данные, выбрать аппаратную платформу, на основе которой будет проектироваться мобильный клиент-проигрыватель.

Ранее были рассмотрены следующие аппаратные платформы:

- Arduino;
- ESP32;
- Raspberry Pi.

Разберем по отдельности их плюсы, а также и минусы. Для начала разберем аппаратную платформу Arduino [1].

Плюсы:

- Простота в использовании: Arduino прост в использовании и имеет простую среду программирования, что делает его подходящим для начинающих;
- Низкая стоимость: Arduino является относительно дешевым компонентом, что делает его более доступным для проектов с ограниченным бюджетом;
- Широкая поддержка и сообщество: Arduino имеет большое сообщество пользователей и разработчиков, что обеспечивает широкую поддержку и помощь;
- Большое количество доступных библиотек и дополнительных компонентов: на Arduino доступно большое количество библиотек и дополнительных компонентов, что делает его более гибким и многофункциональным.

#### Минусы:

- Ограниченный объем памяти: Arduino имеет ограниченный объем памяти, что делает его менее подходящим для выполнения задач, требующих больших объемов памяти;
- Ограниченные возможности встроенных интерфейсов: на Arduino доступны только основные интерфейсы, такие как USB, UART, SPI, I2C, что делает его менее гибким в сравнении с ESP32, который имеет большой набор интерфейсов;
- Ограниченный набор возможностей: Arduino ориентирован на выполнение простых задач, поэтому его возможности ограничены по сравнению с более мощными и продвинутыми платформами, такими как ESP32;
- Нет встроенной поддержки Wi-Fi и Bluetooth: на Arduino нет встроенной поддержки Wi-Fi и Bluetooth, что делает его менее подходящим для создания беспроводных проектов без дополнительных модулей.

Дальше перейдем к аппаратной платформе ESP32 [2].

#### Плюсы:

- Низкое энергопотребление: ESP32 потребляет очень мало энергии, что делает его более подходящим для использования в проектах, работающих на батарейках;

- Более компактный размер: ESP32 имеет более компактный размер, что делает его более подходящим для использования в маленьких пространствах;
- Низкая стоимость: ESP32 является относительно дешевым компонентом, что делает его более доступным для проектов с ограниченным бюджетом;
- Интеграция с Wi-Fi и Bluetooth: ESP32 имеет встроенную поддержку Wi-Fi и Bluetooth, что делает его более подходящим для создания беспроводных проектов.

Минусы:

- Ограниченная вычислительная мощность: ESP32 имеет ограниченную вычислительную мощность, что делает его менее подходящим для выполнения сложных задач, в сравнении с более мощным Raspberry Pi;
- Ограниченный объем памяти: ESP32 имеет ограниченный объем памяти, что делает его менее подходящим для выполнения задач, требующих больших объемов памяти, в сравнении с Raspberry Pi;
- Ограниченный набор интерфейсов: ESP32 имеет ограниченный набор интерфейсов, что делает его менее гибким в сравнении с Raspberry Pi, который имеет большой набор интерфейсов;
- Ограниченный набор возможностей: ESP32 ориентирован на выполнение простых задач, поэтому его возможности ограничены по сравнению с более мощным и продвинутым Raspberry Pi. Например, Raspberry Pi может выполнять задачи, требующие высокой вычислительной мощности, такие как машинное обучение и анализ данных.

И переходим к последней аппаратной платформе Raspberry Pi [3].

Плюсы:

- Большое сообщество: Raspberry Pi имеет большое сообщество пользователей и разработчиков, что обеспечивает широкую поддержку и помощь;
- Широкие возможности: Raspberry Pi имеет большие возможности для расширения, благодаря наличию большого количества доступных шилдов и модулей;

- Большой объем памяти: Raspberry Pi имеет более высокий объем памяти, что делает его более подходящим для выполнения задач, требующих больших объемов памяти.

Минусы:

- Более высокая стоимость: Raspberry Pi обычно стоит дороже, чем ESP32, что делает его менее доступным для некоторых проектов, особенно для простых и небольших.
- Большой размер: Raspberry Pi имеет более крупный размер и форм-фактор, чем ESP32, что делает его менее удобным для использования в некоторых проектах с ограниченным пространством.
- Низкая энергоэффективность: Raspberry Pi потребляет более высокое количество энергии по сравнению с ESP32, что делает его менее подходящим для проектов, работающих на батарейках или других источниках питания с ограниченной емкостью.
- Более сложная настройка: Raspberry Pi требует настройки операционной системы, что может быть сложно для начинающих разработчиков или тех, кто не имеет опыта работы с компьютерами и операционными системами.

В ходе сравнения мы выбрали аппаратную плату в виде ESP32. Так как удовлетворяет требования проекта и имеет возможность дальнейшего улучшения.

## **1.2 Разбор модулей Ethernet**

### **1.2.1 Классификация модулей Ethernet**

Ethernet-модули могут быть классифицированы по различным критериям. Вот несколько типовых классификаций модулей Ethernet:

Физический интерфейс:

- 10BASE-T: Поддержка скорости передачи данных до 10 Мбит/с по витой паре.
- 100BASE-TX: Поддержка скорости передачи данных до 100 Мбит/с по витой паре.

- 1000BASE-T: Поддержка скорости передачи данных до 1 Гбит/с по витой паре.
- 10GBASE-T: Поддержка скорости передачи данных до 10 Гбит/с по витой паре.
- 10GBASE-SR: Поддержка скорости передачи данных до 10 Гбит/с по оптоволоконному кабелю (SR - short range).
- 10GBASE-LR: Поддержка скорости передачи данных до 10 Гбит/с по оптоволоконному кабелю (LR - long range).

Форм-фактор:

- SFP (Small Form-factor Pluggable): Малый форм-факторный модуль с возможностью горячей замены, который поддерживает различные типы подключений и скоростей.
- SFP+: Улучшенная версия SFP, предназначенная для передачи данных на более высоких скоростях, включая 10 Гбит/с и 25 Гбит/с.
- QSFP (Quad Small Form-factor Pluggable): Модуль с четырьмя каналами для передачи данных на скорости до 40 Гбит/с.
- QSFP+: Улучшенная версия QSFP, предназначенная для передачи данных на скоростях 40 Гбит/с и 100 Гбит/с.
- XFP (10 Gigabit Small Form-factor Pluggable): Модуль, предназначенный для передачи данных на скорости 10 Гбит/с.

Медиа-тип:

- RJ-45: Подключение по витой паре, распространенное для Ethernet-соединений в офисных и домашних сетях.
- Оптоволокно: используется для передачи данных на большие расстояния или при высоких скоростях.

### **1.2.2 Выводы и сравнение модулей Ethernet**

В нашем случае идет сравнение двух модулей Ethernet: Arduino W550 [4] и ENC2860 [5].

Таблица 1 – Сравнение модулей Ethernet

Модуль Ethernet	Плюсы	Минусы
Arduino W550	Поддерживает стандарты Ethernet до 100 Мбит/с  Включает в себя контроллер Ethernet и физический уровень в одной микросхеме.	Стоит дороже чем ENC2860  И занимает больше места
ENC2860	Занимает довольно мало места  Стоимость довольно низкая	Требуется внешнего компонента для соединения с физическим уровнем сети.  Поддерживает стандарты Ethernet до 10 Мбит/с.

В ходе сравнения двух Ethernet модулей, мы остановили свой взгляд на Arduino W550. Из-за своих плюсов и доступности.

### 1.3. Обзор Audio усилителей

Усилитель мощности — это устройство, которое используется для управления громкоговорителями или другими нагрузками с низким сопротивлением. Его основная задача - усиление сигнала с минимальными искажениями при передаче высокого напряжения и большого тока на нагрузку в широком диапазоне частот.

Эффективность работы усилителя является важным показателем, поскольку влияет на энергопотребление и генерацию тепла. Усилитель мощности может быть самостоятельным устройством или встроенным в другое устройство.

Усилитель мощности состоит из каскадов усиления, которые могут быть однокаскадными или последовательностью связанных усилительных элементов, цепей нагрузки и связей с другими ступенями. Усилительные элементы могут быть различными, включая электронные лампы, транзисторы или туннельные диоды.



Отрицательная обратная связь используется для стабилизации работы усилителя и снижения искажений сигнала. Для этой цели могут использоваться различные элементы, такие как термисторы или частотно-зависимые компоненты.

Для регулирования степени усиления и частотной характеристики сигнала в различных частях усилителя могут применяться аттенюаторы, потенциометры и фильтры.

Усилители могут быть аналоговыми или цифровыми. В аналоговых усилителях входной сигнал усиливается аналоговыми каскадами, а выходной сигнал остается аналоговым. В цифровых усилителях входной сигнал после аналогового усиления подвергается аналого-цифровому преобразованию, и выходной сигнал становится цифровым.

Класс усилителя — это важная характеристика, которая указывает на режим работы усилительного элемента. Класс усилителя обозначается латинской буквой и определяет, какую часть сигнала он усиливает.

### **1.3.1 Клас усилителя А**

Класс А является простым классом усилителей, использующим один проводящий транзистор и минимизирующим нелинейные искажения сигнала. Хотя класс А редко применяется в выходных каскадах усилителей мощности из-за низкой эффективности и больших потерь энергии, он подходит для входных каскадов. В усилителях класса А часто используются дополнительные транзисторы для управления громкоговорителем на полной мощности. Однако усилители класса А выделяют много тепла, требуя громоздких систем охлаждения и имея высокое потребление электроэнергии.

Несмотря на низкую эффективность, усилители класса А считаются стандартом качества среди аудиофайлов, обеспечивая точное и почти без искажённое воспроизведение сигнала. Однако большинство пользователей не готовы принять низкую эффективность, повышенное выделение тепла, ограниченную мощность и высокое энергопотребление [6].

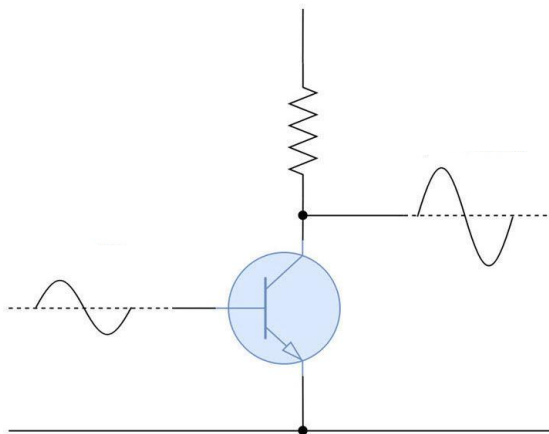


Рисунок 1.3.2 – Работа усилителя класса А

### 1.3.2 Клас усилителя АВ

Усилители класса АВ - наиболее распространенный тип усилителей мощности, который широко используется в домашних кинотеатрах и стереоусилителях. Они объединяют преимущества классов А и В, обеспечивая более высокую эффективность по сравнению с классом А и меньшие искажения по сравнению с классом В. Усилители класса АВ имеют более низкую температуру нагрева и простую схемотехнику, не требующую дорогих компонентов. Однако искажения в усилителях класса АВ все равно выше, чем в усилителях класса А [6].

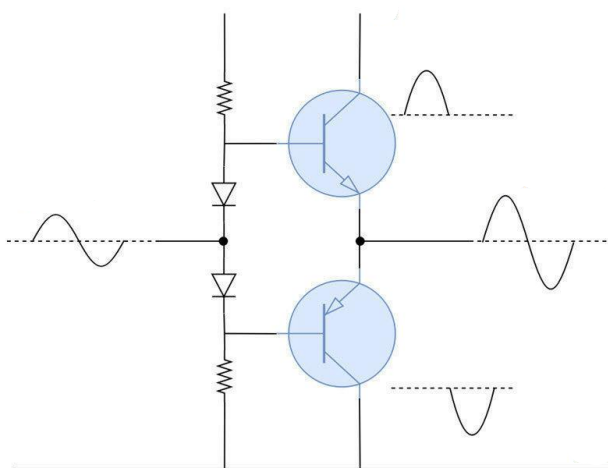


Рисунок 1.3.2 – Работа усилителя класса АВ

### 1.3.3 Клас усилителя В

Усилители класса В работают в двухтактном режиме, позволяя каждому выходному транзистору проводить только половину сигнала. Они обладают более высокой эффективностью по сравнению с классом А, но страдают от кроссоверных искажений, которые приводят к значительным искажениям сигнала и могут восприниматься слушателем как неприятные шумы. Именно по этой причине чистые усилители класса В уже не используются и были полностью заменены усилителями класса D [6].

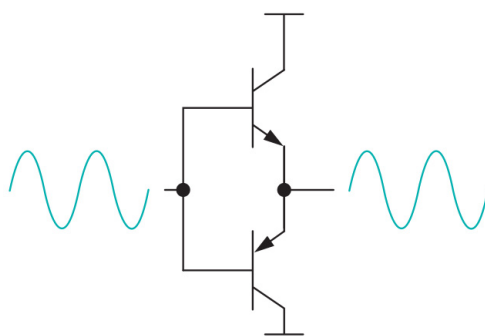


Рисунок 1.3.3 – Работа усилителя класса В

### 1.3.4 Клас усилителя D

Усилители класса D являются популярным и развивающимся типом усилителей благодаря высокой эффективности и отсутствию фоновых шумов. Они обладают компактными размерами, низким тепловыделением и не требуют прогрева перед работой, что делает их популярными в домашней аудиоаппаратуре, профессиональных и портативных устройствах.

Однако, из-за сложности схемотехники и использования высококачественных компонентов, усилители класса D могут быть дорогими. Качество таких усилителей зависит от типа и частоты модуляции. В усилителях класса D малой мощности обычно используется одна микросхема без дополнительных фильтров, и обратная связь компенсирует искажения сигнала и пульсации питания. Высокая частота модуляции помогает фильтровать "лишние" высокие частоты.

В профессиональных усилителях класса D могут быть встроены конвертеры PCM в DSD и цифровой сигнальный процессор (DSP), который компенсирует искажения, вызванные динамиком и характеристиками помещения, обеспечивая высокое качество звука [6].

Таким образом, усилители класса D имеют преимущества, такие как высокая эффективность, компактность и отсутствие фоновых шумов, но могут быть дорогими и иметь сложную схемотехнику. Они особенно полезны для профессионального использования и в портативной технике.

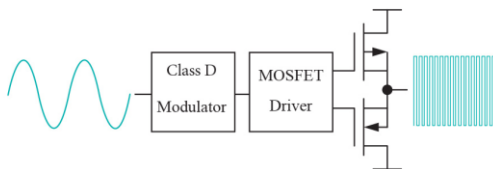


Рисунок 1.3.3 – Работа усилителя класса D

### 1.3.5 Классы усилителя G, H, E, F

Усилители классов G и H являются модификациями класса AB и включают дополнительные источники напряжения, позволяющие изменять напряжение питания выходного каскада в соответствии с уровнем сигнала, что повышает КПД. Класс H особенно подходит для компактных усилителей высокой мощности, применяемых в профессиональных акустических системах. Усилители классов E и F обладают еще более высоким КПД, но ограничены из-за возрастающих искажений. Преимуществами этих усилителей являются высокая эффективность, низкая теплоотдача и компактность. Однако их стоимость может быть высокой. В целом, усилители классов G, H, E и F представляют различные модификации классического AB-усилителя, каждая из которых имеет свои преимущества и недостатки, в зависимости от конкретных потребностей и задач [6].

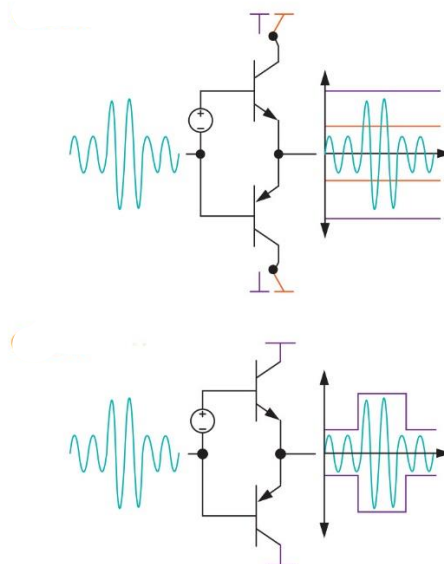


Рисунок 1.3.4 – Работа усилителя класса G, H

### 1.3.4 Выводы и сравнение классов усилителей аудио

В заключение, стоит отметить, что качество усилителя не связано напрямую с его классом, а зависит от выбранной схемотехники. Несмотря на то, что каждый класс имеет свои преимущества и недостатки, эффективность является одним из важных показателей, и наиболее высокая достигается в усилителях класса D.

Класс усилителя: А.

Плюсы:

- Минимальные искажения сигнала;
- Высокая точность воспроизведения.

Минусы:

- Низкая эффективность;
- Высокое потребление энергии;
- Повышенное выделение тепла;
- Малая мощность.

КПД, %: 50.

Класс усилителя: АВ.

Плюсы:

- КПД выше, чем в усилителях класса А;
- Искажения меньше, чем в усилителях класса В;

- Меньший нагрев, чем в усилителях класса А;
- Простая схемотехника.

Минусы:

- Недостаточно высокий КПД.

КПД, %: 50-70.

Класс усилителя: В.

Плюсы:

- Высокий КПД.

Минусы:

- искажения сигнала.

КПД, %: 70.

Класс усилителя: D.

Плюсы:

- Высокая эффективность;
- Компактные размеры и вес;
- Отсутствие фоновых шума;
- Низкая теплоотдача;
- Готовность к работе сразу же после включения.

Минусы:

- Зависимость качества звука от нагрузки на динамик;
- Сложная схемотехника;
- Цена.

КПД, %: 90.

Класс усилителя: H, G.

Плюсы:

- Высокая эффективность;
- Низкая теплоотдача;
- Компактность.

Минусы:

- Цена.

КПД, %: 50-70.

В ходе сравнения классов усилителя аудио файлов мы решили использовать готовый модуль усилитель LM386.

## 1.4 Используемые языки программирования

### 1.4.1 Среда разработки (IDE)

Для разработки программного обеспечения для ESP32 можно использовать различные среды разработки (IDE), в том числе:

- Arduino IDE — это одна из наиболее популярных сред разработки для ESP32. Это бесплатная среда разработки, которая позволяет создавать и загружать скетчи на плату ESP32. В Arduino IDE доступно большое количество библиотек, что упрощает создание проектов на ESP32;
- PlatformIO — это мультиплатформенная среда разработки, поддерживающая ESP32 и многие другие платформы. PlatformIO позволяет работать с Arduino IDE, а также поддерживает различные языки программирования, включая C++, Python и Rust;
- Visual Studio Code с расширением PlatformIO — это еще одна популярная среда разработки для ESP32. Она предоставляет богатый набор функций для разработки, включая подсветку синтаксиса, автодополнение кода и интеграцию с системами контроля версий;
- Espressif IDF — это официальная среда разработки от производителя ESP32, Espressif Systems. Она позволяет разработчикам создавать проекты с помощью языка C или C++, и содержит библиотеки и инструменты для разработки на ESP32.

Выбор конкретной среды разработки для ESP32 зависит от потребностей проекта и личных предпочтений разработчика. Каждая из этих сред разработки имеет свои преимущества и недостатки, поэтому необходимо определить, какие функции и возможности нужны для конкретного проекта, и выбрать среду разработки, которая наилучшим образом соответствует этим требованиям.

Мы остановили выбор на двух средах разработки (IDE) Visual Studio Code и Espressif IDF, сравним их:

Espressif IDF (IoT Development Framework) и Visual Studio Code (VS Code) — это две различные среды разработки, которые могут использоваться для разработки приложений на ESP32.

Espressif IDF — это официальная среда разработки от производителя ESP32, Espressif Systems. Она предоставляет инструменты и библиотеки для создания приложений на ESP32, написанных на языках C и C++. IDF включает

в себя набор инструментов для сборки проектов, отладки и загрузки программного обеспечения на ESP32. IDF также обеспечивает доступ к драйверам и библиотекам, необходимым для работы с различными периферийными устройствами на ESP32.

Visual Studio Code — это мощная и расширяемая среда разработки, которая позволяет разработчикам писать код на различных языках программирования, включая C и C++. Она содержит широкий набор расширений и плагинов, которые обеспечивают подсветку синтаксиса, автоматическое дополнение кода, отладку и многое другое. VS Code может быть настроен для работы с различными фреймворками и платформами, включая ESP32 [7].

Одним из способов использования VS Code для разработки приложений на ESP32 является установка расширения PlatformIO, которое позволяет создавать проекты на ESP32 и загружать программное обеспечение на плату.

В ходе сравнения мы решили, что для нас лучший вариант — это использовать Visual Studio Code. Так удобен в использовании в отличие от Espressif IDF, в котором нет выделения синтаксиса, а также важную роль имеет, что в этой же среде разработки будем писать код для backend и frontend сторон web-сайта.

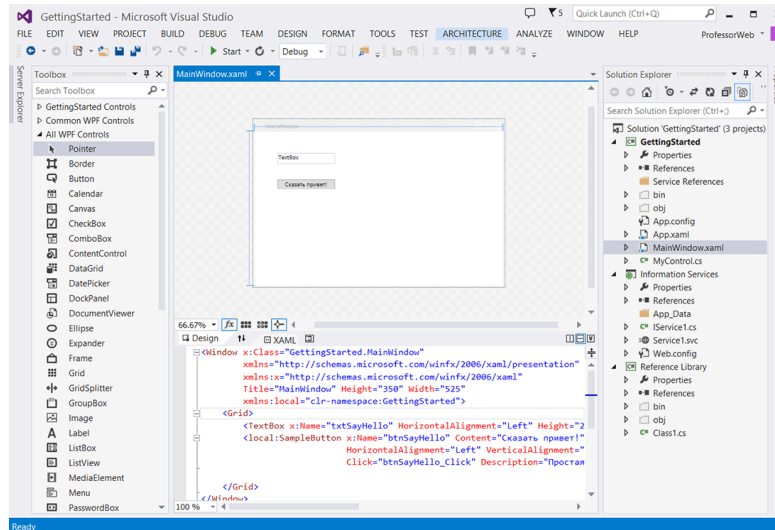


Рисунок 1.1.1 – Интерфейс Visual Studio Code

## 1.4.2 Языки программирования

Для программирования аппаратной платформы и web-сайт будем использовать следующие языки программирования:



C++ является мощным и гибким языком программирования, который широко используется для разработки приложений и системного программирования. Он предоставляет богатый набор возможностей, включая объектно-ориентированное программирование, обработку низкоуровневых операций, шаблоны, многопоточность и многое другое.

При программировании ESP32 на языке C++ с использованием Arduino IDE или PlatformIO, вы можете воспользоваться библиотеками и инструментами, предоставляемыми Arduino или ESP-IDF, чтобы упростить разработку и использование функциональности микроконтроллера ESP32.

C++ предоставляет возможности для эффективной работы с памятью, управления ресурсами и разработки сложных алгоритмов. Он также обеспечивает высокую производительность и позволяет эффективно использовать возможности ESP32.

Однако, при программировании на C++ необходимо быть осторожным с управлением памятью и избегать ошибок, таких как утечки памяти или неправильное освобождение ресурсов, чтобы избежать непредсказуемого поведения или сбоев в работе микроконтроллера.

Основные особенности языка программирования C++:

- Низкоуровневый язык программирования: C позволяет программистам непосредственно управлять аппаратными ресурсами, такими как память и процессор, что обеспечивает высокую производительность;
- Компилируемый язык программирования: Программы на C компилируются в машинный код, что обеспечивает высокую скорость выполнения программы;
- Статическая типизация: C является статически типизированным языком, что означает, что типы данных должны быть определены до компиляции программы;
- Мощные средства работы с памятью: C предоставляет мощные средства работы с памятью, включая указатели, что обеспечивает большую гибкость при работе с данными;
- Большое количество библиотек и инструментов: C имеет большое количество библиотек и инструментов, которые облегчают разработку программ;
- Портруемость: поскольку C является стандартом ISO, программы на C могут быть перенесены на различные платформы без изменения кода.

C является основой для многих других языков программирования, таких как C++, Java, Python и многих других. Язык C по-прежнему очень популярен

в области системного программирования и программирования микроконтроллеров, включая ESP32.

HTML (HyperText Markup Language) — это язык разметки, который используется для создания веб-страниц и веб-приложений. HTML описывает структуру и содержание документа, а также используется для определения внешнего вида и форматирования документа.

Основные особенности языка программирования HTML:

- Разметочный язык: HTML не является языком программирования в традиционном смысле, так как он не имеет операторов или переменных. Он используется только для разметки документов;
- Структурирование документов: HTML используется для описания структуры документа, включая заголовки, параграфы, списки, таблицы и другие элементы;
- Гиперссылки: HTML используется для создания гиперссылок, которые позволяют пользователям перемещаться между различными документами и веб-страницами;
- Формы: HTML используется для создания форм, которые позволяют пользователям отправлять данные на сервер;
- Встроенные элементы: HTML позволяет вставлять веб-графику, видео и аудио на страницы, а также использовать другие интерактивные элементы, такие как анимация и скрипты.

HTML развивается вместе с развитием веб-технологий, поэтому на протяжении времени появляются новые версии языка с новыми элементами и атрибутами. На текущий момент актуальной версией HTML является HTML5.

HTML часто используется вместе с языками программирования, такими как CSS и JavaScript, чтобы создавать более сложные веб-страницы и веб-приложения.

CSS (Cascading Style Sheets) — это язык стилей, который используется для определения внешнего вида и форматирования веб-страниц и веб-приложений, написанных на языках разметки HTML и XML.

Основные особенности языка программирования CSS:

- Объектно-ориентированное программирование (ООП): C++ поддерживает основные принципы ООП, такие как инкапсуляция, наследование и полиморфизм. Это позволяет создавать модульные и расширяемые программы;
- Высокая производительность: C++ предоставляет доступ к низкоуровневым операциям и управлению памятью, что позволяет

разработчикам оптимизировать программы для достижения высокой производительности;

- Шаблоны (Templates): C++ поддерживает шаблоны, которые позволяют создавать обобщенные типы данных и алгоритмы. Это обеспечивает гибкость и возможность повторного использования кода;
- Наследование: CSS также использует механизм наследования, который позволяет наследовать стили от родительских элементов;
- Многопоточность: C++ предоставляет средства для работы с многопоточностью, позволяя создавать параллельные программы и эффективно использовать многопроцессорные системы.

С помощью CSS можно управлять многими аспектами внешнего вида веб-страниц, такими как цвет, шрифты, размеры, отступы, границы, фоны и другие. Также можно определять анимации, переходы и другие эффекты для создания интерактивных и привлекательных веб-приложений.

CSS развивается вместе с развитием веб-технологий, поэтому на протяжении времени появляются новые версии языка с новыми возможностями и свойствами. На текущий момент актуальной версией CSS является CSS3.

JavaScript — это интерпретируемый язык программирования, который часто используется для создания динамических веб-страниц и веб-приложений. Он может выполняться как на стороне клиента (в браузере), так и на стороне сервера (с использованием Node.js).

Некоторые из основных особенностей JavaScript:

- Динамическая типизация: переменные в JavaScript не нужно объявлять заранее с указанием типа данных, так как тип переменной может изменяться во время выполнения программы;
- Функциональное программирование: JavaScript поддерживает функциональное программирование, позволяющее использовать функции как значения и передавать их в другие функции;
- Объектно-ориентированное программирование: в JavaScript есть поддержка объектов и классов, позволяющая создавать объекты и методы;
- Обработка событий: JavaScript позволяет реагировать на события, такие как клики мыши, нажатия клавиш, изменения содержимого элементов HTML и т.д.;

- Асинхронный код: JavaScript поддерживает асинхронное программирование, позволяя выполнять несколько операций одновременно без блокировки потока выполнения программы.

JavaScript используется для создания динамического поведения веб-страниц, таких как интерактивные элементы, анимации, проверка форм и т.д. Он также используется для создания многих типов веб-приложений, включая одностраничные приложения (SPA), игры, приложения для обработки изображений и видео, веб-сервисы и другие.

JavaScript является одним из наиболее популярных языков программирования в мире и имеет широкое сообщество разработчиков, которые создают новые библиотеки и фреймворки для упрощения разработки веб-приложений.

## **2 Разработка электрических схем мобильного клиент-проигрывателя аудиофайлов на базе Ethernet**

### **2.1 Разработка структурной схемы мобильного клиент-проигрывателя аудио файлов на базе Ethernet**

#### **2.2.1 Обоснование базовых блоков структурной электрической схемы мобильного клиент-проигрывателя аудио файлов**

В рамках курсового проекта для разработки мобильного клиент-проигрывателя аудио файлов, была взята за основу аппаратная платформа ESP32 микроконтроллер ESP32S DevKit.

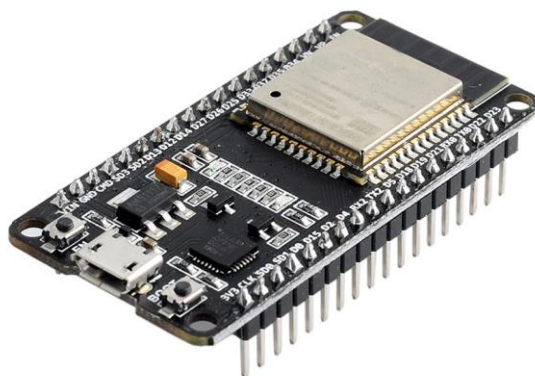


Рисунок 2.2.1 – ESP32S DevKit

ESP32S DevKit — это плата разработки, основанная на микроконтроллере ESP32. Вот некоторые особенности и характеристики ESP32S DevKit:

- Микроконтроллер: ESP32, двухядерный процессор Tensilica LX6 с тактовой частотой до 240 МГц;

- Wi-Fi: Встроенный Wi-Fi модуль, поддерживающий стандарты 802.11 b/g/n;
- Bluetooth: Встроенный Bluetooth модуль, поддерживающий Bluetooth Classic и Bluetooth Low Energy (BLE);
- Оперативная память (RAM): 520 Кб (включая 8 Кб буфера для запуска);
- Память для программного обеспечения (Flash): 4 Мбайт;
- GPIO (общего назначения ввода-вывода): 34 пина GPIO, которые могут быть настроены для различных целей, таких как ввод/вывод данных, прерывания, PWM, SPI, I2C и другие;
- UART: 3 UART-интерфейса для подключения и обмена данными с другими устройствами;
- I2C: 2 I2C-интерфейса для подключения и обмена данными с датчиками и другими периферийными устройствами;
- SPI: 4 SPI-интерфейса для обмена данными с другими устройствами, такими как дисплей, флэш-память и другие;
- ADC: 12-битный аналого-цифровой преобразователь с 18 каналами;
- Частота ядра: Максимальная частота процессора - 240 МГц;
- Питание: Рабочее напряжение 2,2 В - 3,6 В. Поддерживается подача питания через USB или внешний источник;
- Разъемы: Микро USB порт, разъем для подключения антенны Wi-Fi/BLE, разъемы для подключения дополнительных модулей и расширительных плат;
- Размеры: Плата имеет компактный размер, обычно около 54 мм x 25 мм;
- Операционные системы: поддерживает различные операционные системы, включая FreeRTOS, Arduino и другие.

Далее подключения через Ethernet мы используем Arduino W550 [4].



Рисунок 2.2.2 – Arduino W5500

Arduino W5500 представляет собой Ethernet Shield (расширительную плату) для платформы Arduino, которая оснащена контроллером Ethernet W5500. Вот основные особенности Arduino W5500:

- Контроллер Ethernet: Встроенный контроллер Ethernet W5500, поддерживающий стандарты Ethernet и TCP/IP протоколы;
- Скорость передачи данных: поддерживает скорость передачи данных до 10/100 Мбит/с;
- Интерфейс: подключается к Arduino через шину SPI;
- Память: Встроенная память (буфер) размером 8 Кб для обработки данных Ethernet;
- Разъемы: Разъемы для подключения сетевого кабеля Ethernet (RJ-45) и Arduino;
- Поддержка протоколов: поддерживает протоколы TCP, UDP, ICMP, IPv4, ARP, IGMP и другие;
- встроенный стек TCP/IP: обеспечивает возможность подключения к сети Ethernet и передачи данных через протокол TCP/IP;
- Совместимость: совместим с платформой Arduino и может быть использован с различными моделями Arduino, такими как Arduino Uno, Arduino Mega и другими;
- Поддержка библиотек: существуют библиотеки и примеры кода для удобного программирования и использования Arduino W5500;
- Использование ресурсов: Ethernet Shield использует SPI интерфейс и не занимает все пины Arduino, оставляя другие пины доступными для подключения других устройств

Arduino W5500 Ethernet Shield позволяет Arduino-платформе подключаться к сети Ethernet и обмениваться данными по сети, что открывает возможности для разработки проектов, связанных с Интернетом, удаленного управления и сетевого взаимодействия.

Для усиления звука мы выбрали модуль усилитель LM386 [8].



**Рисунок 2.2.3 – Модуль усилителя LM386**

LM386 — это маломощный аудио усилитель на основе операционного усилителя, который широко используется для усиления аудио сигналов. Вот некоторые особенности LM386:

- Низкое потребление энергии: LM386 имеет низкое потребление энергии, что делает его идеальным для портативных и батарейных устройств;
- Простота использования: Усилитель LM386 имеет простую схему и минимальное количество внешних компонентов, что упрощает его включение в проекты;
- Работа от одного источника питания: LM386 может работать от одного источника питания, что упрощает его подключение и использование;
- Регулируемый усиление: с помощью некоторых внешних компонентов, усиление LM386 может быть легко настроено и регулировано в зависимости от требуемого уровня усиления;
- Низкий уровень искажений: LM386 обеспечивает достаточно низкий уровень искажений, что важно для достижения качественного звука;
- Встроенная защита: LM386 имеет встроенную защиту от перегрузок и короткого замыкания, что обеспечивает безопасность и долговечность устройства;
- Настройка частотной характеристики: с помощью некоторых внешних компонентов, частотная характеристика LM386 может быть настроена для подходящего диапазона частот;



- Малый размер: LM386 имеет компактный размер, что облегчает его включение в различные проекты с ограниченным пространством.

LM386 широко используется в различных приложениях, включая портативные радиоприемники, громкоговорители, музыкальные инструменты и другие устройства, требующие небольшой мощности и усиления аудио сигналов.

Динамик мы будем использовать 0,5 ГДШ-2-8 [9].



**Рисунок 2.2.4 – Динамик 0,5 ГДШ-2-8**

0,5 ГДШ-2-8 – это устройство предназначено для использования в компактных портативных акустических системах, которые встроены в бытовую радиоаппаратуру. Они предназначены для работы в помещении и на открытом воздухе, и относятся к третьей группе сложности. Вот его некоторые характеристики:

- Диапазон частот: 315 – 5000 Гц;
- Неравномерность АЧХ: 12 дБ;
- Чувствительность: 90 дБ;
- Номинальное среднее звуковое давление: 0,21 Па;
- Коэффициент гармоник на частотах 630 – 4000 Гц: 5 %
- Номинальное сопротивление: 8 Ом;
- Паспортная мощность: 0,5 Вт;
- Долговременная мощность: 2 Вт;
- Кратковременная мощность: 3 Вт;
- Резонанс: 290 Гц ( $\pm 60$  Гц);
- Полная добротность: 1,5 – 3;

- Размеры:  $\varnothing 63 \times 22$  мм;
- Вес: 130 г.

Несмотря на то, что данный динамик относительно устарел, но благодаря своим характеристиками и доступности имеет место быть в нашем курсовом проекте.

## **2.2 Разработка принципиальной схемы мобильного клиент-проигрывателя аудио файлов на базе Ethernet**

### **2.2.1 Обоснование выбора САПР для Разработки принципиальной электрической схемы мобильного клиент-проигрывателя аудио файлов на базе Ethernet**

Для начала разберем что такое САПР, для дальнейшего понимания и выбора конкретной программы.

САПР — это аббревиатура от "Система автоматизированного проектирования". Это программное обеспечение, которое позволяет инженерам, архитекторам, дизайнерам и другим специалистам создавать и редактировать различные модели и проекты на компьютере [10].

С помощью САПР можно создавать 2D и 3D модели объектов и систем, проводить анализ различных параметров, таких как силы, теплообмен, динамика и т.д. САПР также позволяет проектировать электрические схемы, печатные платы, архитектурные проекты, механизмы и многое другое.

САПР является важным инструментом в современном проектировании и производстве, позволяя значительно сократить время на проектирование и улучшить качество продукции. Он также помогает избежать ошибок, которые могут возникнуть в процессе ручного проектирования, и обеспечивает более точный и точный расчет и анализ различных параметров.

На данный момент существует множество САПР для различных отраслей промышленности и проектирования. Рассмотрим некоторые из них:

- AutoCAD - один из самых популярных и широко используемых САПР в мире. Он используется в архитектуре, машиностроении, электротехнике и других отраслях.
- SolidWorks - САПР для трехмерного моделирования, широко используется в машиностроении и промышленном дизайне.
- CATIA - САПР для создания трехмерных моделей, используется в авиастроении, автомобильной и других отраслях.

- Siemens NX - САПР для интегрированного проектирования и производства, используется в авиастроении, машиностроении, электронике и других отраслях.
- Altium Designer - САПР для проектирования электронных схем и печатных плат.
- Revit - САПР для архитектурного проектирования, используется для создания 3D-моделей зданий и сооружений.
- ANSYS - САПР для численного моделирования и анализа различных параметров, используется в инженерии, механике, электротехнике и других отраслях.
- KiCad — Это один из наиболее популярных САПР для электронной инженерии и электротехники.

Это лишь некоторые примеры САПР, которые используются в различных отраслях. Каждый из них имеет свои особенности и возможности, и выбор определенного САПР зависит от конкретной задачи и отрасли.

В нашем случае подходит KiCad и AutoCAD, разберём их более подробно.

KiCad — это бесплатное и открытое программное обеспечение (Open Source), предназначенное для проектирования электронных схем и печатных плат.

KiCad предоставляет множество инструментов для создания и редактирования электрических схем и печатных плат. Он включает в себя библиотеки компонентов, которые можно использовать при создании схем, а также инструменты для создания пользовательских компонентов.

Одной из особенностей KiCad является его интеграция с программным обеспечением для создания 3D-моделей. Это позволяет инженерам создавать виртуальные модели своих печатных плат и проверять их взаимодействие с другими компонентами и системами.

KiCad также имеет функциональность для генерации списка материалов (BOM) и автоматического трассирования печатных плат, что упрощает процесс проектирования и производства.

KiCad поддерживает множество форматов файлов, включая Gerber, Excellon, SVG и DXF, что обеспечивает совместимость с большинством производственных процессов и программных средств.

KiCad является бесплатным и открытым программным обеспечением, что означает, что его исходный код доступен для всех и может быть изменен

и распространен в соответствии с лицензией GNU GPL. Это позволяет пользователям вносить свои вклады в развитие программного обеспечения и улучшать его функциональность в соответствии с их потребностями.

AutoCAD — это программное обеспечение для 2D и 3D-проектирования и черчения, которое используется в различных областях, таких как архитектура, машиностроение, электротехника и другие. Оно было разработано компанией Autodesk и является одним из самых популярных САПР в мире.

AutoCAD обладает мощным функционалом, который позволяет создавать и редактировать различные типы чертежей, моделей и проектов. Среди его возможностей:

- Создание 2D- и 3D-моделей с использованием различных инструментов, таких как линии, дуги, круги, эллипсы, полилинии и т.д.
- Использование библиотеки объектов для быстрого создания элементов, таких как окна, двери, мебель, электронные компоненты и т.д.
- Автоматическое построение объектов на основе указанных параметров, например, размеров и углов.
- Редактирование и модификация объектов, включая их перемещение, копирование, масштабирование и т.д.
- Работа с точками привязки и пространствами, такими как модельное, листовое и бумажное пространство.
- Создание анимаций и визуализаций проектов.
- Интеграция с другими программными средствами, такими как Autodesk Revit и Autodesk Inventor.

AutoCAD поддерживает множество форматов файлов, включая DWG, DXF, PDF и DGN, что обеспечивает совместимость с другими САПР и программами для обработки данных.

AutoCAD доступен для Windows и macOS, а также имеет мобильное приложение для работы на смартфонах и планшетах.

В ходе разбора этих двух САПР мы выяснили, что больше для нашего курсового проекта KiCad, так он является бесплатным для использования в отличие от AutoCAD и имеет более нужную специфику.

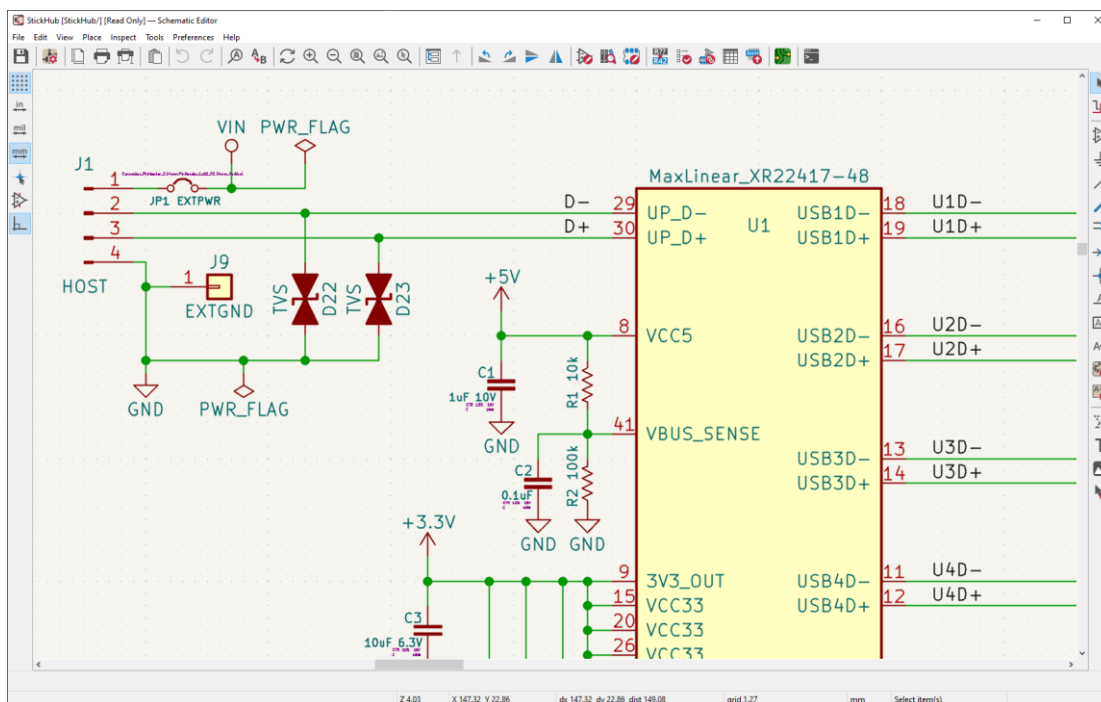


Рисунок 2.2.5 – Интерфейс KidCad

## 2.2.2 Описание используемых модулей и библиотечных элементов

Ранее была описана структурная схема мобильного клиент-проигрывателя ауди файлов на базе Ethernet, где все устройство разбито на основные компоненты для его работы:

- ESP32S DevKit;
- Arduino W5500;
- LM386;
- 0,5 ГДШ-2-8.

Каждый отдельный компонент мобильного клиент-проигрывателя на базе Ethernet представляет собой отдельную модель или плату и имеет свою принципиальную электрическую схему.

Управлять мы будем с помощью микропроцессора ESP32 установленный на отладочной плате ESP32S DevKit, который представлен на общей принципиальной схеме представлен в виде компонента ESP32 WROOM из библиотеки ESP32 взята из интернета.

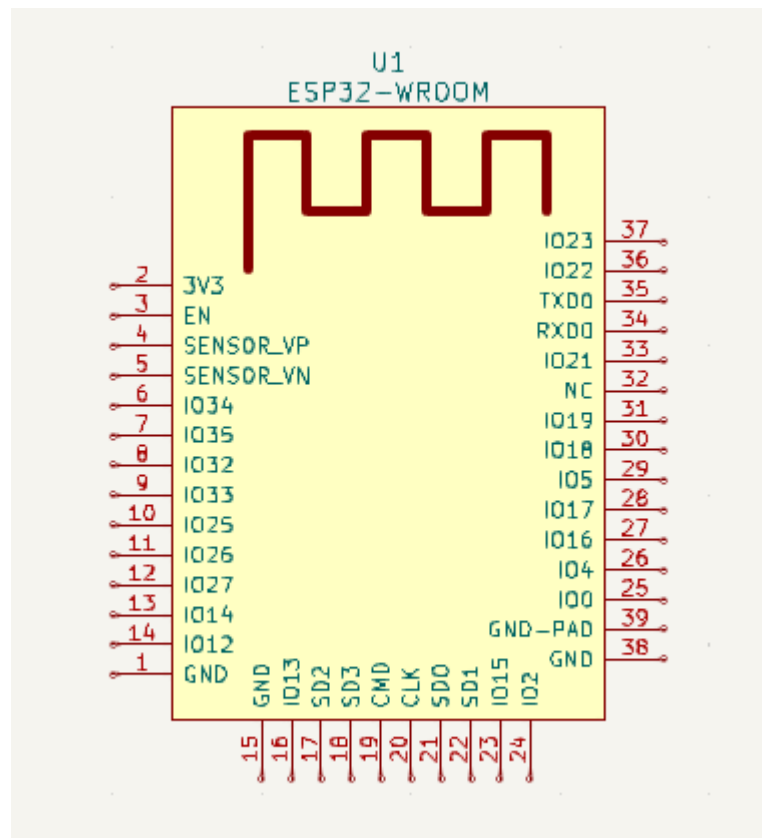


Рисунок 2.2.6 – ESP32 WROOM

Для подключения к локальной сети, мы будем использовать готовый модуль Arduino W5500, который нам позволит подключиться к сети с помощью технологии Ethernet. Взяв его с стандартной библиотеки KidCad.

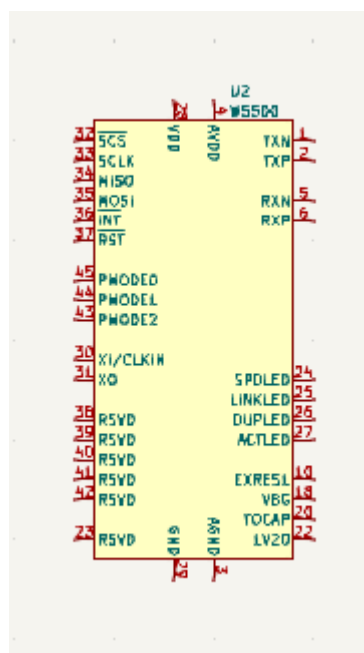


Рисунок 2.2.7 – W5500

Для усиления звука мы будем использовать LM386, так данного усилителя нет в библиотеки KidCad и нет возможности ее добавить, мы соберем ее самостоятельно для общей принципиальной схемы.

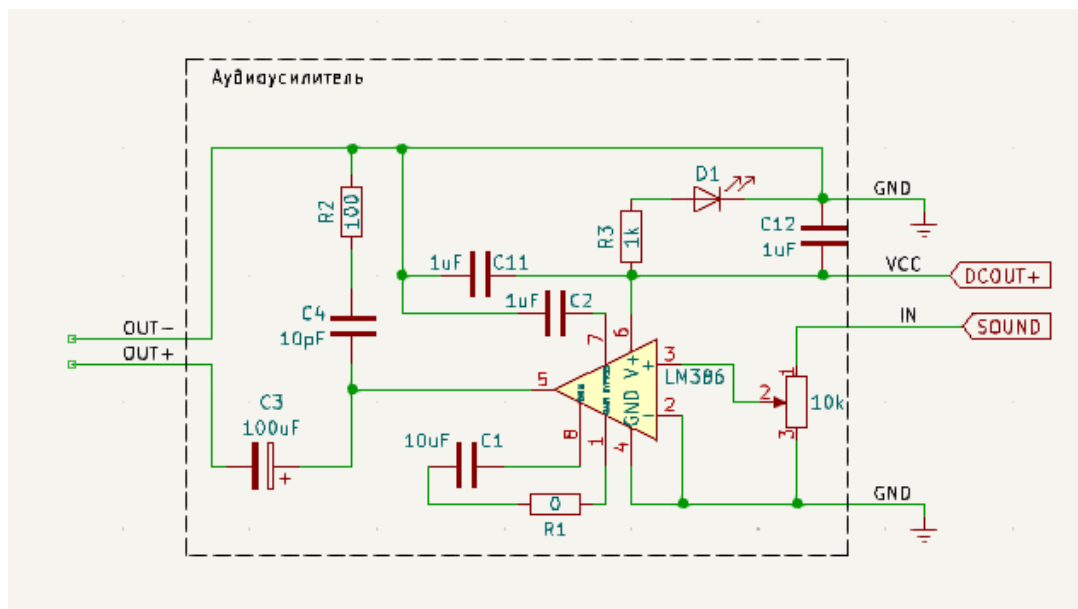


Рисунок 2.2.7 – Модуль усилитель LM386

Для обозначения динамика 0,5 ГДШ-2-8, мы будем использовать простое обозначение громкоговорителя, взятого из библиотеки KidCad. Так для воспроизведения аудио файлов, в дальнейшем возможно будет изменить конфигурацию, так он легко отключатся от мобильного клиент-проигрывателя.

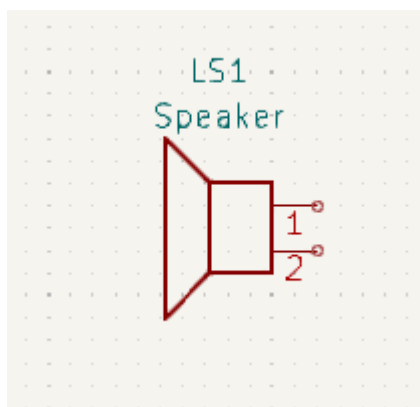


Рисунок 2.2.8 – Speaker LS1

## **3 Разработка алгоритма и функционирования мобильного клиент-проигрывателя аудио файлов на базе Ethernet**

### **3.1 Описание алгоритма функционирования мобильного клиент-проигрывателя аудио файлов на базе Ethernet**

Стоит начать с технических требований к алгоритму. Надо определить функции, которые должны выполняться.

В рамках курсового проекта у мобильного клиент проигрывателя на базе Ethernet будет собственный web-сервер под названием “MusicCloud”, следовательно все управление будет осуществляться через него.

Исходя из выше сказанного наш функционал должен быть таким:

- Включении и выключение аудио файла;
- Увеличение и уменьшение громкости воспроизведения;
- Переключение аудио файлов;
- Возможность перемешать аудио файлов.

### **3.2 Выбор локального веб-сервера**

Веб-сервер — это программное обеспечение или компьютер, который предоставляет доступ к веб-сайтам, приложениям или другим ресурсам через Интернет. Он обрабатывает запросы от клиентов (обычно веб-браузеров) и отправляет им ответы, содержащие запрошенную информацию.

Роль веб-сервера заключается в обработке HTTP-запросов от клиентов и предоставлении им необходимых данных. Когда пользователь вводит веб-адрес в браузере, браузер отправляет запрос на этот адрес, и веб-сервер отвечает, предоставляя запрошенную веб-страницу, файлы, изображения или другие ресурсы [11].

Он может обрабатывать различные типы содержимого, включая статические файлы (например, HTML, CSS, JavaScript, изображения) и динамически генерируемые страницы (например, с помощью серверных скриптов или приложений). Он может также поддерживать различные протоколы, такие как HTTP, HTTPS, FTP и другие.

Различные веб-серверы могут использовать разные программные платформы и технологии, такие как Apache HTTP Server, Nginx, Microsoft IIS, Node.js, и другие. Они могут быть установлены на отдельном сервере или



интегрированы в другое программное обеспечение, например, веб-фреймворк или приложение для обслуживания контента.

Рассмотрим не сколько таких:

- OpenServer — это локальный веб-сервер среды разработки для операционных систем Windows. Он предоставляет удобный способ установки и настройки веб-сервера, баз данных (например, MySQL), PHP и других необходимых компонентов для разработки веб-приложений на локальной машине;
- XAMPP: это один из самых популярных вариантов локальных веб-серверов. XAMPP включает в себя веб-сервер Apache, базу данных MySQL, интерпретаторы PHP и Perl, а также другие инструменты, необходимые для разработки веб-приложений. Он доступен для Windows, macOS, Linux и других операционных систем;
- WampServer: это еще один популярный локальный веб-сервер для Windows. WampServer включает в себя Apache, MySQL и PHP, обеспечивая все необходимые компоненты для разработки и тестирования веб-приложений на локальном компьютере
- MAMP: это локальный веб-сервер для macOS. Он содержит Apache, MySQL и PHP, и предоставляет удобный способ развертывания веб-приложений на локальной машине под управлением macOS;
- Laragon: это легковесный и простой в использовании локальный веб-сервер для разработки веб-приложений на платформе Windows. Laragon включает Apache, PHP, MySQL и другие инструменты, и предлагает удобную установку и управление окружением разработки.

Мы остановим свой взгляд на OpenServer, так удобен в использовании, также имеется опыт с работой с данным программным обеспечением.

### **3.2.1 Настройка локального сервера для ESP32**

Создав просто сайт, который мы рассмотрим не много подробнее в следующем пункте. Недостаточно, чтобы мы могли зайти на него с других устройств для прослушивания. И управлением MP3 проигрывателям.

Так как память ESP32 иметь большой объём, а также написания страницы внутри ESP32, довольно сложное и объективно не уместное в нашем случае действие.

И так перейдем к настройке OpenServer:  
Узнаем IP нашего ПК, на котором храниться наш web-server. Это можно узнать через командную строку вписав команду ipconfig

```
Выбрать Командная строка
Microsoft Windows [Version 10.0.19045.2965]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Users\Алексей>ipconfig

Настройка протокола IP для Windows

Адаптер Ethernet Ethernet:

    Состояние среды. . . . . : Среда передачи недоступна.
    DNS-суффикс подключения . . . . . :

Адаптер Ethernet Ethernet 2:

    DNS-суффикс подключения . . . . . :
    Локальный IPv6-адрес канала . . . : fe80::e57:875f:e1e9:9ea3%9
    IPv4-адрес. . . . . : 192.168.56.1
    Маска подсети . . . . . : 255.255.255.0
    Основной шлюз. . . . . :

Адаптер беспроводной локальной сети Подключение по локальной сети* 1:

    Состояние среды. . . . . : Среда передачи недоступна.
    DNS-суффикс подключения . . . . . :

Адаптер беспроводной локальной сети Подключение по локальной сети* 2:

    Состояние среды. . . . . : Среда передачи недоступна.
    DNS-суффикс подключения . . . . . :

Адаптер беспроводной локальной сети Беспроводная сеть:

    DNS-суффикс подключения . . . . . :
    Локальный IPv6-адрес канала . . . : fe80::ed28:5f1a:5847:bfc3%3
    IPv4-адрес. . . . . : 192.168.0.107
    Маска подсети . . . . . : 255.255.255.0
    Основной шлюз. . . . . : 192.168.0.1

C:\Users\Алексей>
```

Рисунок 3.2.1 – Командная строка

Запомним наш IP “192.168.0.107” он нам в дальнейшем понадобится для входа на наш сайт.

Дальше заходим в OpenServer-е в настройки и вставляем наш IP.

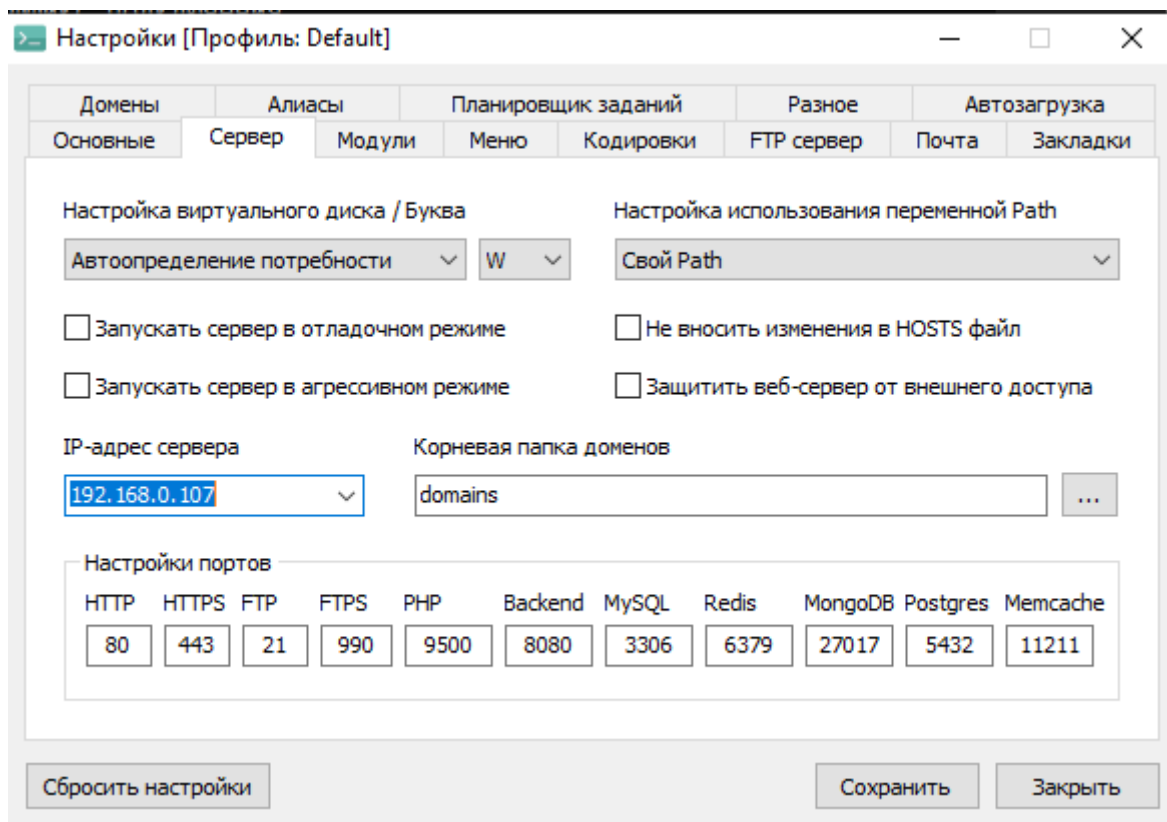


Рисунок 3.2.2 – Настройка сервера

Дальше заходим на вкладку Алиасы и вставляем и опять наш IP, и указываем Название нашего проекта.

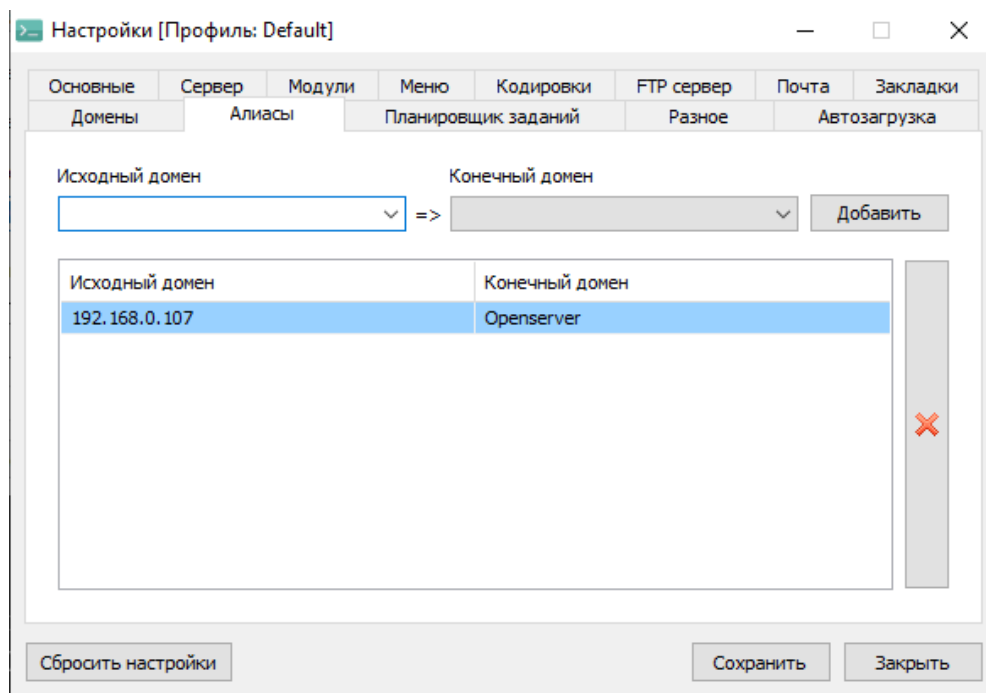


Рисунок 3.2.3 – Настройка Алиасы

Далее сохраним настройки. После этого OpenSever перезапустится. И при попытке зайти на наш сайт через другое устройства мы все так же не сможем. Это из-за того, что Брендмаузер.

Решается это тем, что мы создадим входящие правило с портами 80,80,80.

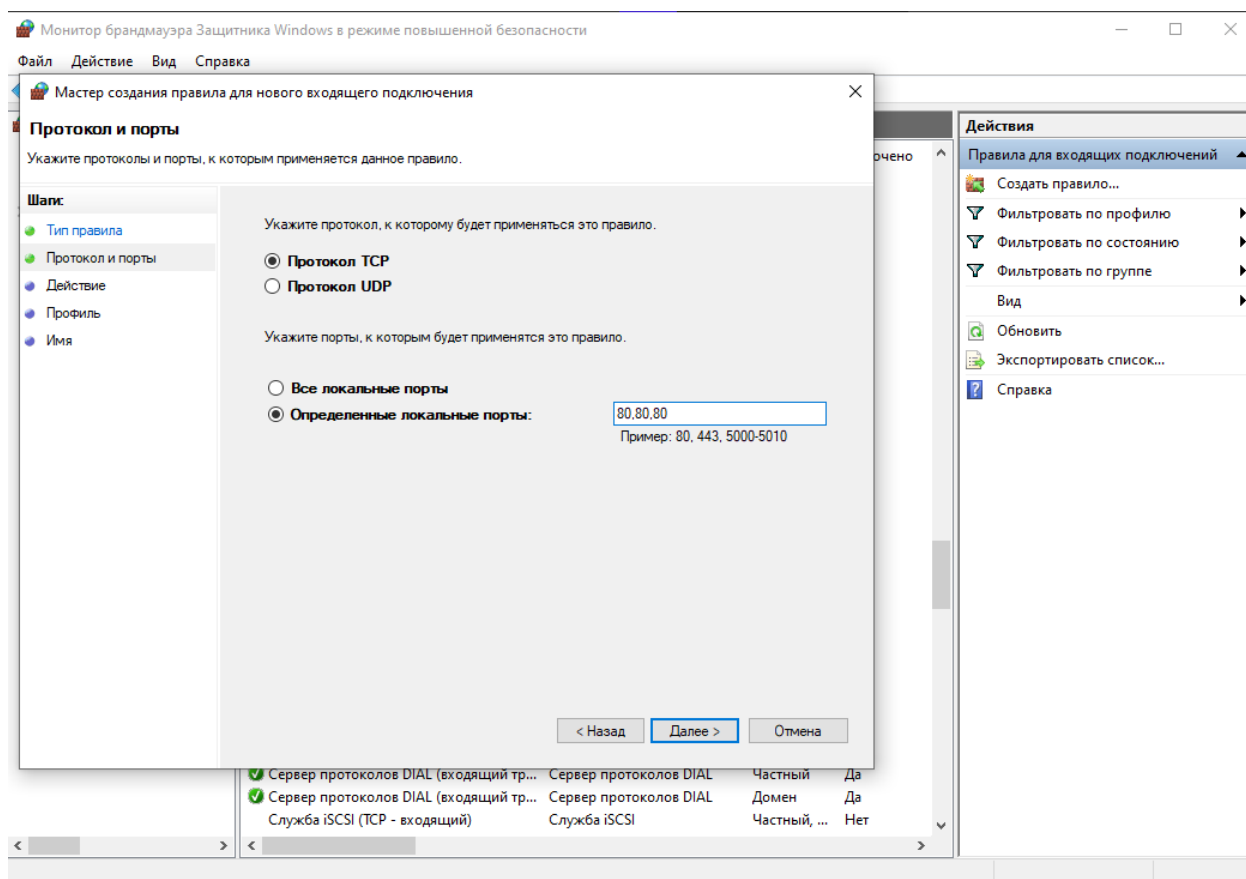


Рисунок 3.2.4 – Создание правила входящего

### 3.3 Разработка веб-сервера “MusiCloud”

Для нашего веб-сервера “MucisCloud” мы создадим две html страницы:

- Страница с приветствием;
- Страница с MP3 плеером.

Дальше при создании страницы с MP3 плеером, для управления мы будем использовать JavaScript. С помощью которого мы сможем выполнять такие функции, как:

- Воспроизведение аудио файла;

- Переключение аудио файла;
- Зацикливание аудио файла;
- Перемешивания аудио файлов;
- Управлением уровнем громкости.

Для красивого оформления нашего веб-сервера, мы напишем код на css. И в итоге получим такой результат:

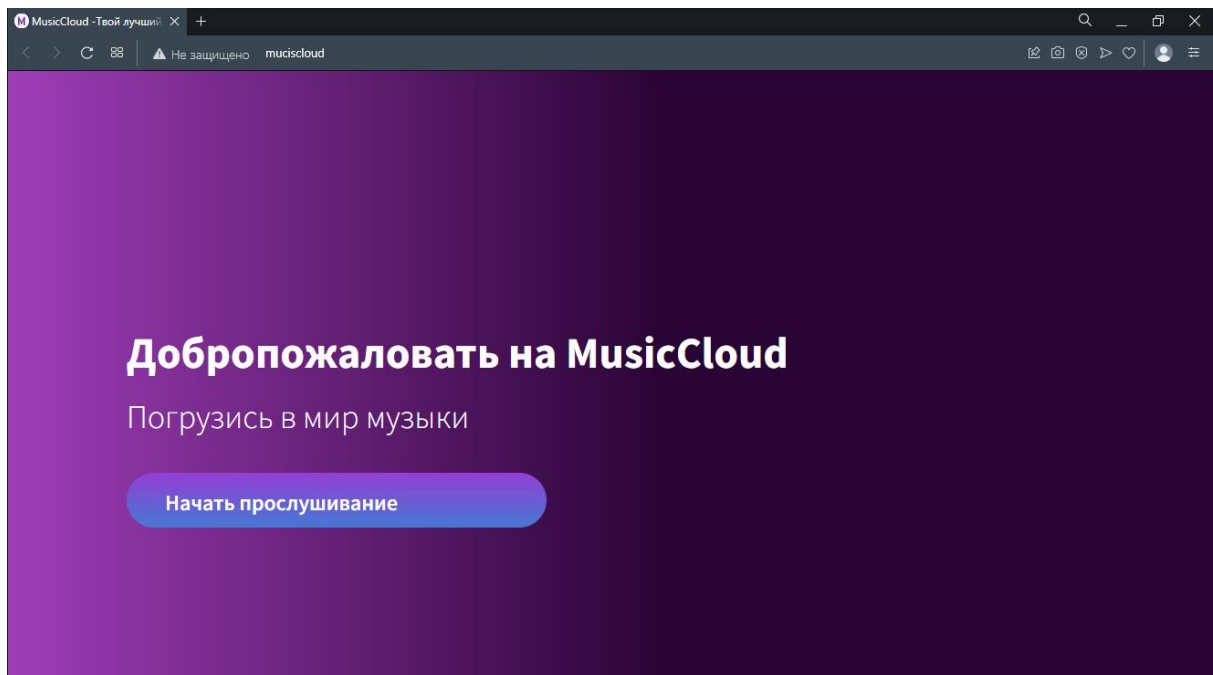
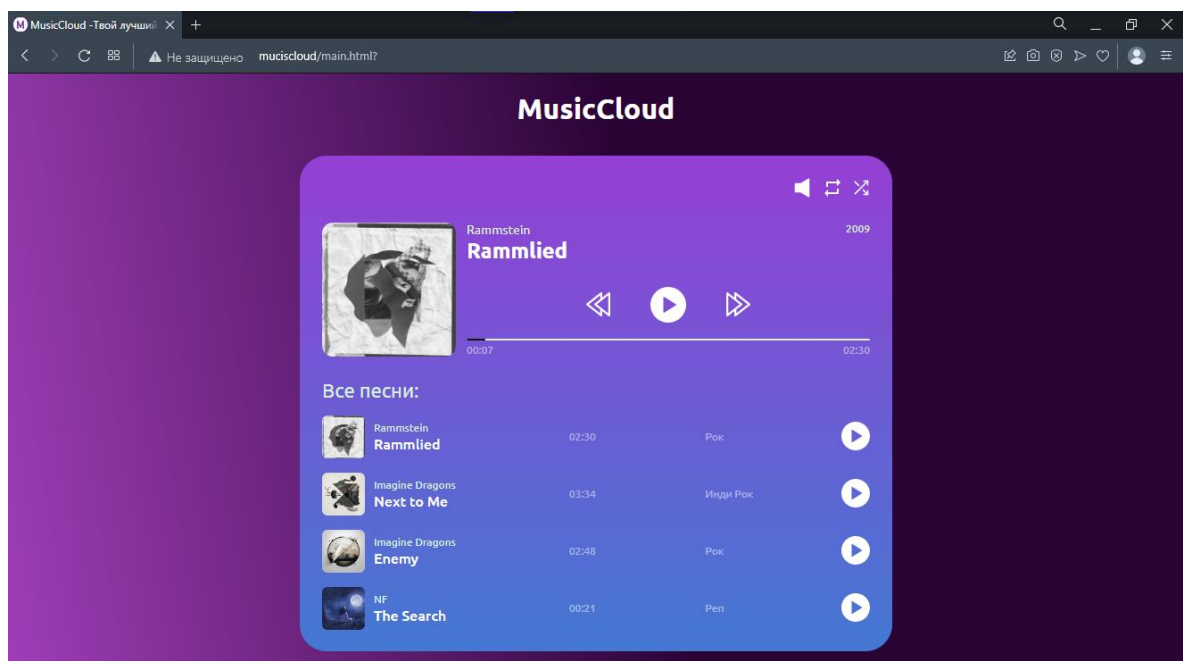


Рисунок 3.3.1 – Страница приветствия



### 3.4 Использование библиотек для ESP32

Для корректной работы нашего мобильного клиент-проигрывателя аудио файлов на базе Ethernet, были использованы библиотеки:

- Ethernet.h
- SPI.h
- SPIFFS.h
- FS.h

#### 3.4.1 Описание библиотеки Ethernet.h

В библиотеке Ethernet.h [12] обычно содержатся определения и функции для работы с сетевым интерфейсом Ethernet на Arduino или других подобных платформах. Она обеспечивает возможность подключения и обмена данными по сети, используя протокол Ethernet.

Некоторые из основных функций и классов, доступных в библиотеке Ethernet.h, могут включать:

- Ethernet: класс, представляющий сетевой интерфейс Ethernet. Он позволяет настраивать параметры сетевого подключения, устанавливать и разрывать соединения, а также передавать данные;
- EthernetClient: класс, представляющий клиентское соединение Ethernet. Он позволяет устанавливать соединение с удаленным хостом и передавать данные в обоих направлениях;
- EthernetServer: класс, представляющий серверное соединение Ethernet. Он позволяет создавать серверный сокет и принимать входящие соединения от удаленных клиентов;
- EthernetUDP: класс, предоставляющий возможность работы с протоколом UDP (User Datagram Protocol) через Ethernet. Он позволяет отправлять и принимать UDP-пакеты;
- Ethernet.begin(): функция, используемая для инициализации сетевого интерфейса Ethernet;
- Ethernet.maintain(): функция, используемая для обновления состояния сетевого интерфейса Ethernet и обработки входящих пакетов;

- `Ethernet.localIP()`: функция, возвращающая IP-адрес, присвоенный устройству в локальной сети.

### **3.4.2 Описание библиотеки SPI.h**

Библиотека `SPI.h` [13] является стандартной библиотекой Arduino и предоставляет функции для работы с интерфейсом SPI (Serial Peripheral Interface). Интерфейс SPI позволяет обмениваться данными между микроконтроллером Arduino и другими устройствами, такими как датчики, дисплеи, EEPROM и другие периферийные устройства.

Основные функции и классы, предоставляемые библиотекой `SPI.h`, включают:

- `SPI.begin()`: инициализирует библиотеку SPI и начинает работу с интерфейсом SPI;
- `SPI.beginTransaction()`: запускает транзакцию SPI, обычно вызывается перед началом обмена данными;
- `SPI.transfer()`: отправляет и принимает байт данных по шине SPI;
- `SPI.endTransaction()`: завершает транзакцию SPI;
- `SPI.end()`: Завершает работу с библиотекой SPI и освобождает ресурсы.
- `SPISettings`: Класс, используемый для настройки параметров передачи данных по SPI, таких как скорость передачи, порядок бит и режим тактирования.

### **3.4.3 Описание библиотеки SPIFFS.h**

Библиотека `SPIFFS.h` [14] предоставляет возможность работать с файловой системой SPIFFS (SPI Flash File System) на микроконтроллерах, таких как ESP8266 и ESP32. SPIFFS — это файловая система, которая хранит файлы на внешней флеш-памяти, подключенной через интерфейс SPI.

С помощью библиотеки `SPIFFS.h` вы можете выполнять следующие операции:

- Инициализация файловой системы SPIFFS с помощью `SPIFFS.begin()`;
- Открытие файла для чтения с помощью `SPIFFS.open()`;
- Открытие файла для записи с помощью `SPIFFS.open()`;

- Чтение данных из файла с помощью методов `read()` или `readBytes()`;
- Запись данных в файл с помощью методов `write()` или `writeBytes()`;
- Закрытие файла с помощью метода `close()`;
- Переименование файла с помощью метода `rename()`;
- Удаление файла с помощью метода `remove()`;
- Получение информации о файле с помощью метода `info()`;
- Создание новой директории с помощью метода `mkdir()`;
- Удаление директории с помощью метода `rmdir()`;
- Перемещение в начало файла с помощью метода `seek()`.

### 3.4.4 Описание библиотеки SPIFFS.h

Библиотека `FS.h` [15] является общей библиотекой файловых систем (File System) для платформ на основе Arduino, таких как ESP8266 и ESP32. Она предоставляет базовые функции для работы с файлами и директориями на различных типах файловых систем, включая SPIFFS, LittleFS и FAT.

`FS.h` является абстрактным интерфейсом, который определяет общие операции с файлами и директориями, такие как открытие, чтение, запись, удаление и перемещение. Фактическая реализация зависит от используемой файловой системы, которую вы выбираете.

Некоторые из основных функций и классов, доступных в библиотеке `FS.h`, включают:

- `FS.begin()`: Инициализирует выбранную файловую систему;
- `FS.open()`: Открывает файл для чтения или записи;
- `File`: Класс, представляющий файл. Имеет методы для чтения и записи данных, перемещения указателя в файле и других операций с файлами;
- `File.available()`: проверяет наличие данных, доступных для чтения из файла;
- `File.read()`: читает байт из файла;
- `File.write()`: записывает байт в файл;
- `File.seek()`: устанавливает позицию указателя файла;
- `File.close()`: закрывает файл;
- `FS.exists()`: проверяет существование файла или директории;
- `FS.remove()`: удаляет файл или директорию;
- `FS.rename()`: переименовывает файл или директорию;
- `FS.mkdir()`: создает директорию;



## **Заключение**

В ходе работы выполнения курсового проекта по разработке мобильного-клиент проигрывателя на базе Ethernet, был произведена детальная работа над анализом предметной области и принципы работы клиент-проигрывателя аудио файлов на базе Ethernet. В результате было спроектировано и создано устройство, которое позволяет пользоваться воспроизведением аудио файлов с помощью технологии Ethernet.

В процессе работы были рассмотрены различные аппаратные платформы и микроконтроллеры, различные аудио усилители, модули Ethernet, ознакомились с классами усилителей, так же с технологией Ethernet. Был изучен принцип работы аппаратной платформы ESP32, так алгоритмы работы веб-серверов.

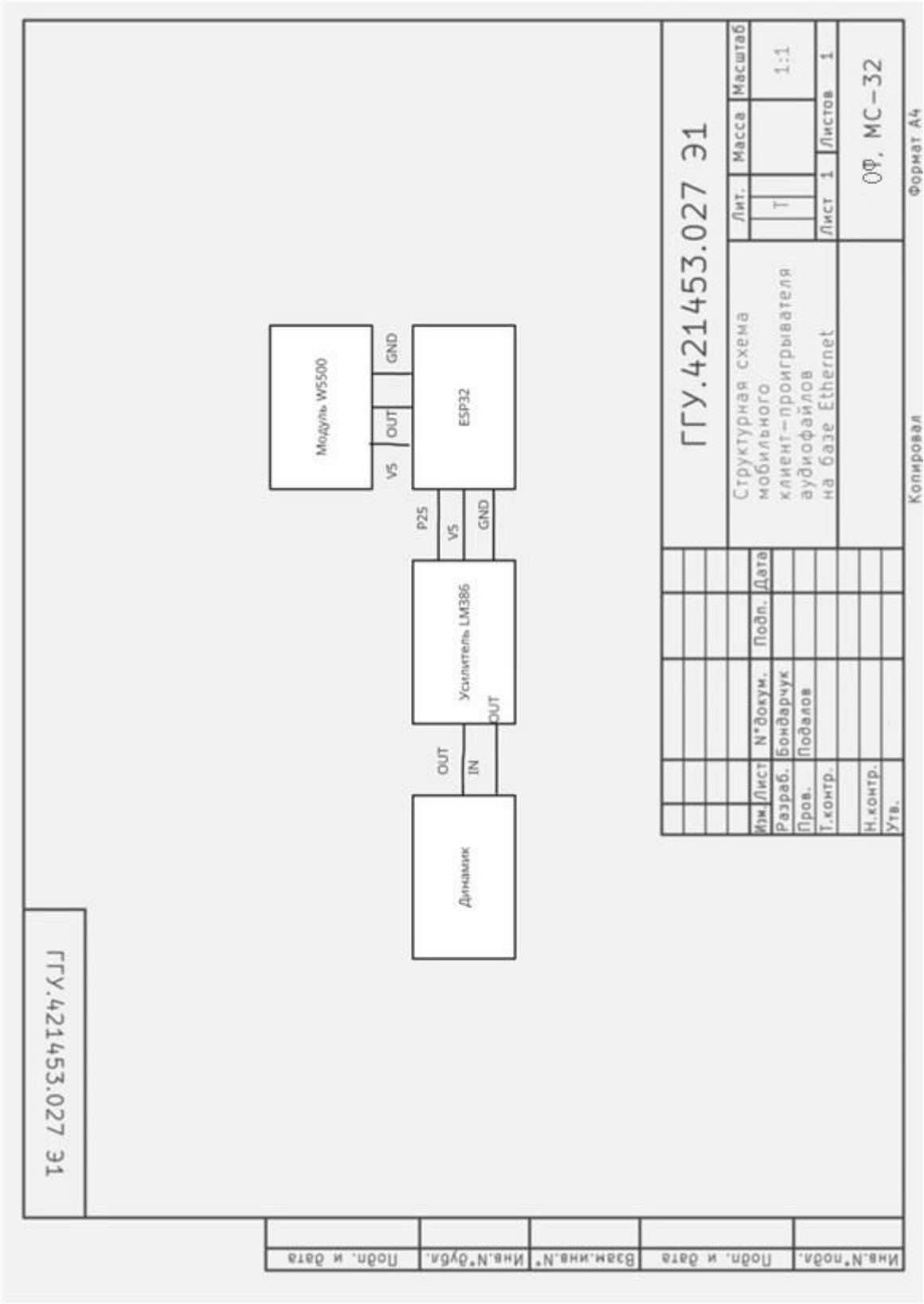
В итоге были разработаны принципиальная электрическая схема, структурная электрическая схема и алгоритмы работы мобильного клиент-проигрывателя аудио файлов на базе Ethernet.

## СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Arduino [Электронный ресурс] // Свободная энциклопедия Википедия. – URL: <https://ru.wikipedia.org/wiki/Arduino> – Дата доступа: 26.03.2023.
2. ESP32 [Электронный ресурс] // Свободная энциклопедия Википедия. – URL: <https://ru.wikipedia.org/wiki/ESP32> - Дата доступа: 27.03.2023.
3. Raspberry Pi [Электронный ресурс] // Свободная энциклопедия Википедия. – URL: [https://ru.wikipedia.org/wiki/Raspberry\\_Pi](https://ru.wikipedia.org/wiki/Raspberry_Pi) - Дата доступа: 27.03.2023.
4. Модуль w5500 [Электронный ресурс] // IARDUINO. – URL: <https://iarduino.ru/shop/Expansion-payments/setevoy-modul-w5500-tsr-ip-ethernet.html> - Дата доступа 13.04.2023.
5. Модуль ENC2860 [Электронный ресурс] // chipdip. – URL: <https://www.chipdip.by/product/enc28j60-h> - Дата доступа: 13.04.2023.
6. Классы усилителей [Электронный ресурс] // POP-MUSIC. - URL: <https://pop-music.ru/articles/klassy-usiliteley/> - Дата доступа: 14.04.2023.
7. VS code [Электронный ресурс] // Visual Studio Code. – URL: <https://code.visualstudio.com/> - Дата доступа: 14.04.2023.
8. Модуль LM386 [Электронный ресурс] // Compact Tool. – URL: <https://compacttool.ru/modul-audio-usilitel-zvuka-lm386-monofonicheskiy-2vt> - Дата доступа: 15.04.2023.
9. Динамик 0,5 ГДШ-2-8 [Электронный ресурс] // LDSOUND. – URL: <https://ldsound.info/05-gdsh-2-8/> - Дата доступа: 15.04.2023.
10. САПР [Электронный ресурс] // Свободная энциклопедия Википедия. - URL: [https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0\\_%D0%B0%D0%B2%D1%82%D0%BE%D0%BC%D0%B0%D1%82%D0%B8%D0%B7%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B3%D0%BE\\_%D0%BF%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D0%B0%D0%B2%D1%82%D0%BE%D0%BC%D0%B0%D1%82%D0%B8%D0%B7%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BF%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F) – Дата обращения: 28.04.2023.
11. OpenServer [Электронный ресурс] // OsPanel. – URL: <https://ospanel.io/> - Дата обращения 07.05.2023.
12. Ethernet.h [Электронный ресурс] // ARDUINO.UA. – URL: <https://doc.arduino.ua/ru/prog/Ethernet> - Дата обращения 08.05.2023.

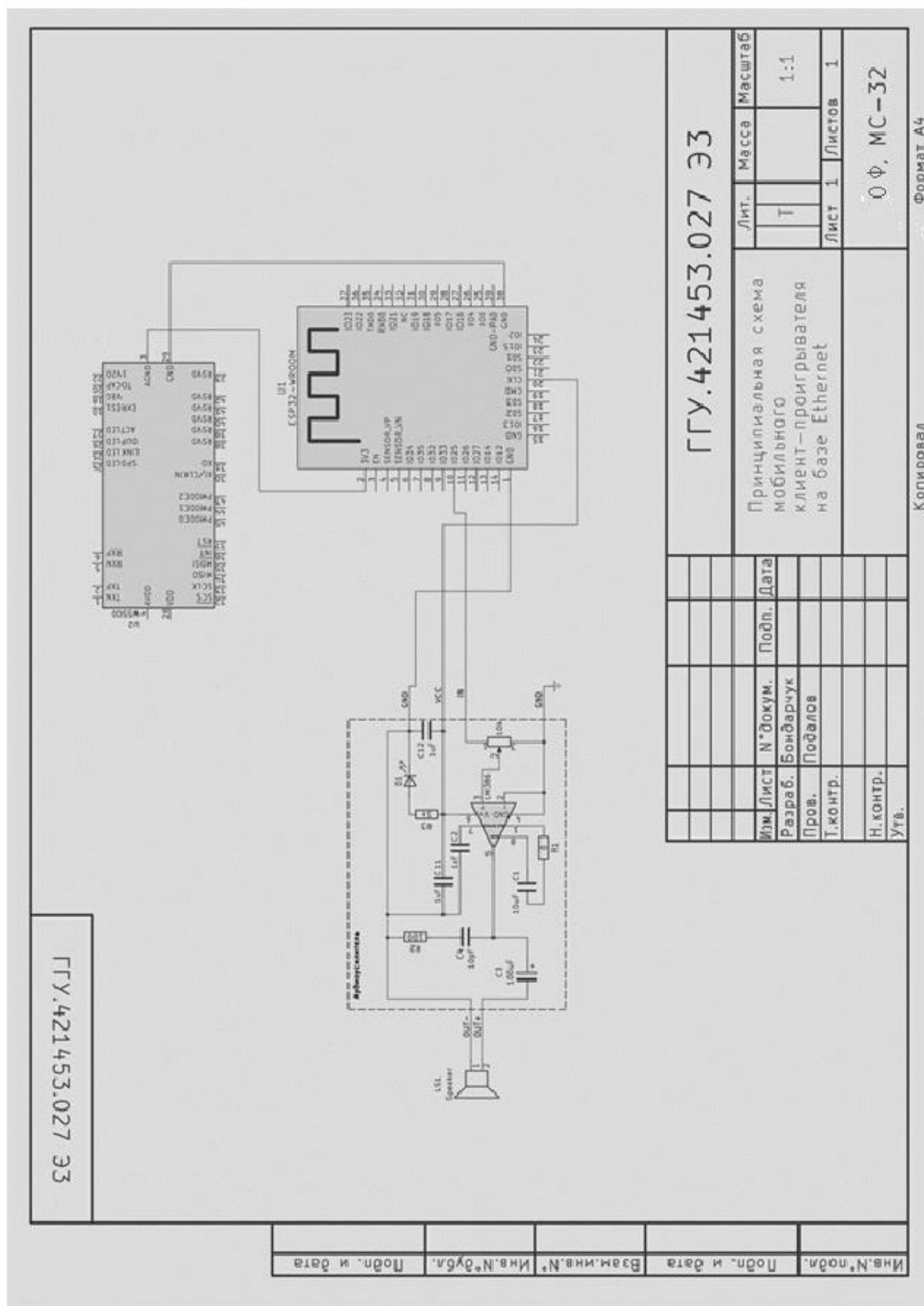
13. SPI.h [Электронный ресурс] // ARDUINO.UA. – URL: <https://doc.arduino.ua/ru/prog/SPI> - Дата обращения 08.05.2023.
14. SPIFFS.h [Электронный ресурс] // habr.com. – URL: <https://habr.com/ru/articles/483280/> - Дата обращения 08.05.2023.
15. FS.h [Электронный ресурс] // habr.com. – URL: <https://habr.com/ru/articles/483280/> - Дата обращения 08.05.2023.
16. AlekseiBondarchuk [Электронный ресурс]// GitHub. – URL: <https://github.com/AlekseiBondarchuk/ethernet-basedmobileaudioclient> - Дата обращения. 16.05.2023.

# ПРИЛОЖЕНИИ А Структурная электрическая схема мобильного клиент-мобильного



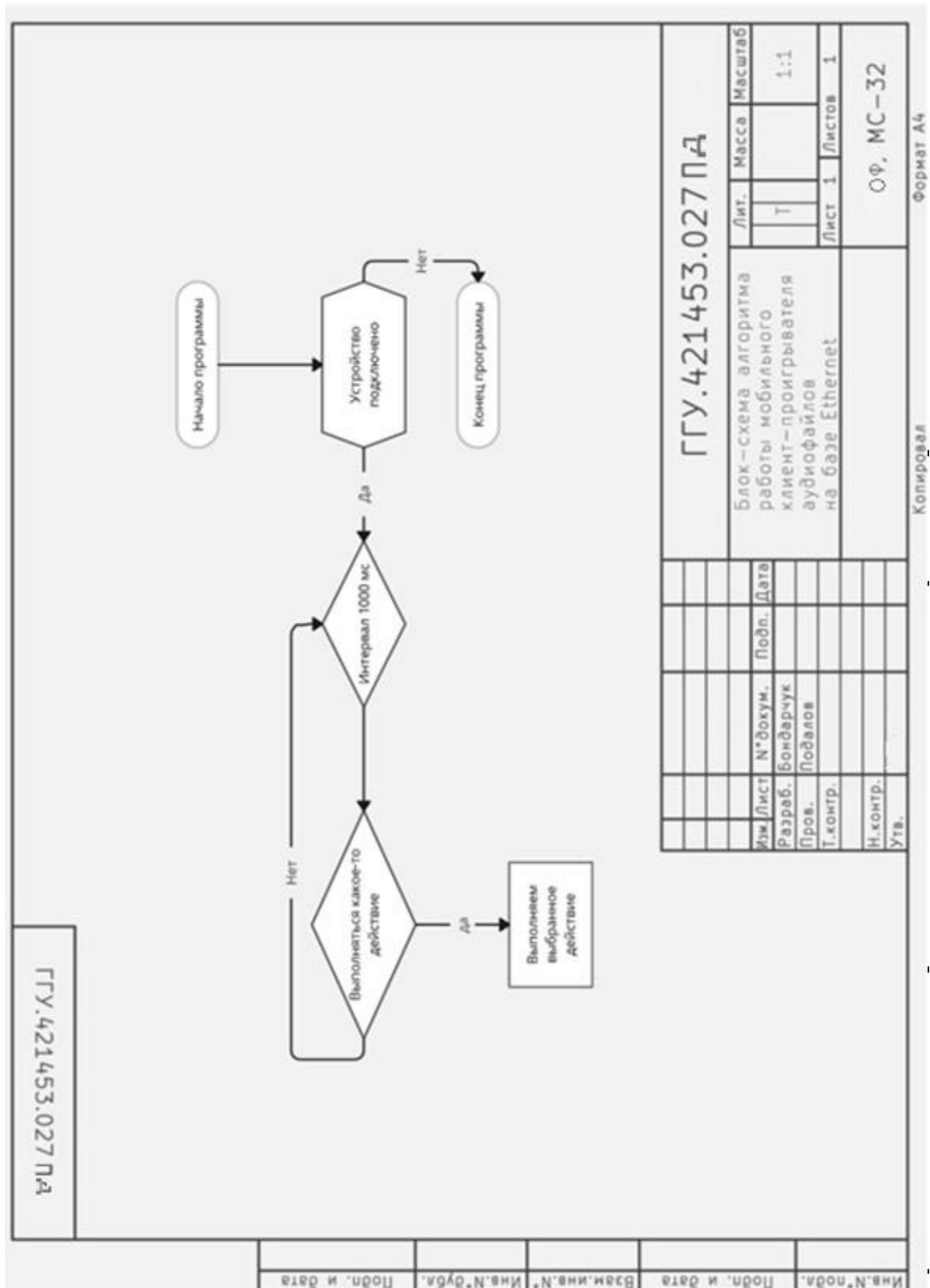
# ПРИЛОЖЕНИЕ Б

## Принципиальная электрическая схема электрического схема мобильного проигрывателя аудиофайлов



## ПРИЛОЖЕНИЕ В

### Блок-схема алгоритма работы мобильного клиент проигрывателя аудиофайлов на базе Ethernet



## Спецификация курсового проекта

Перв. примен.		ГГУ.421453.027		Формат	Зона	Поз.	Обозначение	Наименование	Кол.	Примечание
Справ. №		ГГУ.421453.027		А4						
Подп. и дата		ГГУ.421453.027		А4						
Взам. инв. №		ГГУ.421453.027		А4						
Подп. и дата		ГГУ.421453.027		А4						
Инв. № подл.		ГГУ.421453.027		А4						

ГГУ.421453.027				
Изм.	Лист	№ докум	Подпись	Дата
Разраб.		Бондарчук		
Пров.		Подалов		
Н.контр				
Итв.				

Мобильный декодер штрих-кодов		
Литера	Лист	Листов
Т		1
ОФ, МС-32		

## Ведомость к курсовому проекту

[illegible]