

Министерство образования Республики Беларусь
Учреждение образования «Белорусский государственный университет
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра программного обеспечения информационных технологий

Дисциплина: Современные платформы программирования (СПП)

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к курсовой работе
на тему:

«Онлайн-тренажер Р-180»

БГУИР КП 1-40 01 01 605 ПЗ

Студент: гр. 851006 Верещагин Н. В.

Руководитель: асс. Низовцов Д. В.

Минск 2021

Учреждение образования
«Белорусский государственный университет информатики и
радиоэлектроники»

Факультет компьютерных систем и сетей

УТВЕРЖДАЮ

Заведующий кафедрой ПОИТ

(подпись)

Лапицкая Н.В. 2021 г.

ЗАДАНИЕ

по курсовому проектированию

Студенту Верещагину Николаю Владимировичу

1. Тема работы «Онлайн-тренажер Р-180»
2. Срок сдачи студентом законченной работы 15.05.2021
3. Исходные данные к работе Документация по Node.js
4. Содержание расчётно-пояснительной записки (перечень вопросов, которые подлежат разработке)

Введение.

1. Анализ прототипов, литературных источников и формирование требований к проектируемому ПС;
 2. Разработка алгоритма;
 3. Разработка программного средства;
 4. Тестирование, экспериментальные исследования и анализ полученных результатов;
 5. Руководство пользователя программы;
- Заключение, список литературы, ведомость, приложения.

5. Перечень графического материала (с точным обозначением обязательных чертежей и графиков)

1. Схема программы на А1

6. Консультант по курсовому проекту Низовцов Д. В.

7. Дата выдачи задания 13.02.2021 г.

8. Календарный график работы над проектом на весь период проектирования (с обозначением сроков выполнения и процентом от общего объёма работы):

раздел 1 к 28.02.2021 – 15 % готовности работы;

разделы 2, 3 к 21.03.2021 – 30 % готовности работы;

раздел 4 к 18.04.2021 – 60 % готовности работы;

раздел 5 к 10.05.2021 – 90 % готовности работы;

оформление пояснительной записки и графического материала к 15.05.2021 –

100 % готовности работы. Защита курсового проекта с 17.05 по 10.06 2021 г.

РУКОВОДИТЕЛЬ _____ Д. В. Низовцов
(подпись)

Задание принял к исполнению Верецагин Н.В. _____ 13.02.2021 г.
(дата и подпись студента)

СОДЕРЖАНИЕ

Введение	5
1 Анализ предметной области	7
1.1 Обзор аналогов	7
1.2 Постановка задачи	12
2 Разработка программного средства	13
2.1 Структура программы	13
2.2 Интерфейс программного средства	13
2.3 Запись и передача звука на сервер	14
2.4 Обработка данных сервером	15
2.5 Меню программы	17
2.6 Воспроизведение полученных с сервера данных	18
2.7 Настройка частот	19
3 Тестирование программного средства	20
4 Руководство пользователя	22
4.1 Руководство по установке и запуску	22
4.2 Руководство по использованию	23
Заключение	31
Список использованных источников	32
Приложение 1	33
Приложение 2	34

ВВЕДЕНИЕ

21-й век – век информационных и телекоммуникационных технологий. Всё больше отраслей специализируются в компьютерной сфере, так как это наиболее актуальное и современное направление. Новейшие научные достижения безудержно преобразуют мир вокруг нас. Большим изменениям подверглась и сфера коммуникаций. Современные средства связи обладают огромными преимуществами перед радиосвязью, но, будучи очень уязвимыми, могут быть ликвидированы при первой возможности. Радиосвязь же может быть использована локально. Таким образом, знание работы с радиостанциями является необходимым навыком.

Первый патент на беспроводную связь получил в 1872-ом году американский радиолюбитель, стоматолог по профессии Малон Лумис, заявивший в 1866-ом году о том, что он открыл способ беспроволочной связи; в США изобретателем радио считают Дэвида Хьюза, а также Томаса Эдисона и Николу Теслу; в Германии – Генриха Герца; во Франции – Эдуарда Бранли; в ряде балканских стран – Николу Теслу; в Бразилии – Ланделя де Муру; в Англии – Оливера Джозефа Лоджа; в Индии – Джагадиша Чандру Боше; в России – А. С. Попова и Якова Наркевича-Иодко [1].

Создателем первой успешной системы обмена информацией с помощью радиоволн на западе считается итальянский инженер Гульельмо Маркони.

В СССР и в бывших союзных республиках изобретателем радиотелеграфии считается А. С. Попов. В опытах по радиосвязи, проведённых в физическом кабинете, а затем в саду Минного офицерского класса, прибор А. С. Попова обнаруживал излучение радиосигналов, посылаемых передатчиком, на расстоянии до 60-ти метров. На заседании Русского физико-химического общества в Петербурге 25-го апреля (седьмого мая) 1895-го года А. С. Попов продемонстрировал, как указано в протоколе заседания, «прибор, предназначенный для показывания быстрых колебаний в атмосферном электричестве». В СССР, с 1945-го года, седьмого мая стали отмечать как День радио.

Изобретение радиосвязи дало начало таким научно-техническим направлениям, как радиоастрономия, радиометрология, радионавигация, радиолокация, радиоразведка, радиопротиводействие.

Радио послужило мощнейшим стимулом в исследовании и развитии электричества, и стало основой электроники. А электроника, в свою очередь, породила вычислительную технику.

Ультракороткие волны (УКВ) – традиционное в СССР название диапазона радиоволн, объединяющего метровые, дециметровые, сантиметровые и миллиметровые волны (или диапазоны очень высоких частот – ОВЧ, ультравысоких частот – УВЧ, сверхвысоких частот – СВЧ и крайне высоких частот – КВЧ). То есть это все радиоволны, длина которых менее 10-ти метров.

Радиоволны – электромагнитные волны с частотами до трёх ТГц, распространяющиеся в пространстве без искусственного волновода. Радиоволны в электромагнитном спектре располагаются от крайне низких частот вплоть до инфракрасного диапазона. С учётом классификации Международным союзом электросвязи радиоволн по диапазонам, к радиоволнам относят электромагнитные волны с частотами от 0,03 Гц до трёх ТГц, что соответствует длине волны от 10-ти миллионов. километров до 0,1 миллиметра.

Существует огромное множество УКВ радиостанций, наиболее известные из них:

- Р-173;
- Р-123МТ;
- Р-111;
- Р-159;
- Р-180;
- Р-181-5НУ

Целью данного курсового проекта является разработка приложения «Онлайн-тренажёр Р-180».

1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1 Обзор аналогов

Существует ряд тренажёров радиостанций, каждый из них имеет свои преимущества и недостатки. На рисунке 1.1 представлен один из наиболее удобных для освоения тренажёров радиостанции Р-130, работающей в коротковолновом диапазоне.

Радиостанция имеет дискретную сетку частот и обеспечивает бес поисковое вхождение в связь и ведение связи без подстройки на фиксированных частотах с однотипными радиостанциями и радиостанциями Р-104М, Р-112, Р-123, Р-129, Р-134, Р-140, Р-143. Радиостанция допускает работу с танковым шлемофоном. Переключатель диапазонов Р-130 – механический, барабанного типа. Р-130 обеспечивает совместную работу на передачу с аппаратурой быстрогодействия в ЧТ, а также телефонную работу через переговорное устройство Р-124 и без него. Рабочая частота устанавливается тремя ручками «КИЛОГЕРЦЫ» – х1000, х100, х1 с десятичной системой счисления которая позволяет устанавливать частоту в полной темноте.



Рисунок 1.1 – Тренажёр радиостанции Р-130

В программном средстве есть пошаговые рекомендации по включению и настройке радиостанции, а также устройство блока выносного согласующего антенного устройства, которое изображено на рисунке 1.2. Также есть возможность самопроверки в самостоятельном режиме настройки устройства без подсказок приложения, таким образом, оно может быть использовано преподавательским составом.

Единственным недостатком данного приложения является неочевидная эксплуатация программного средства вместе с другими приложениями. Не подготовленный пользователь может совершить ряд ошибок, что испортит впечатление от получения навыков.



Рисунок 1.2 – Устройство блока выносного согласующего антенного устройства

Другим аналогом послужило приложения для создания тестов по различным радиостанциям, их тактико-технических характеристиках и использованию – MilitaryTest. Стандартная база данных программы включает такие устройства как: P-111, P-123, P-130 и другие.

Главное меню приложения представлено на рисунке 1.3.

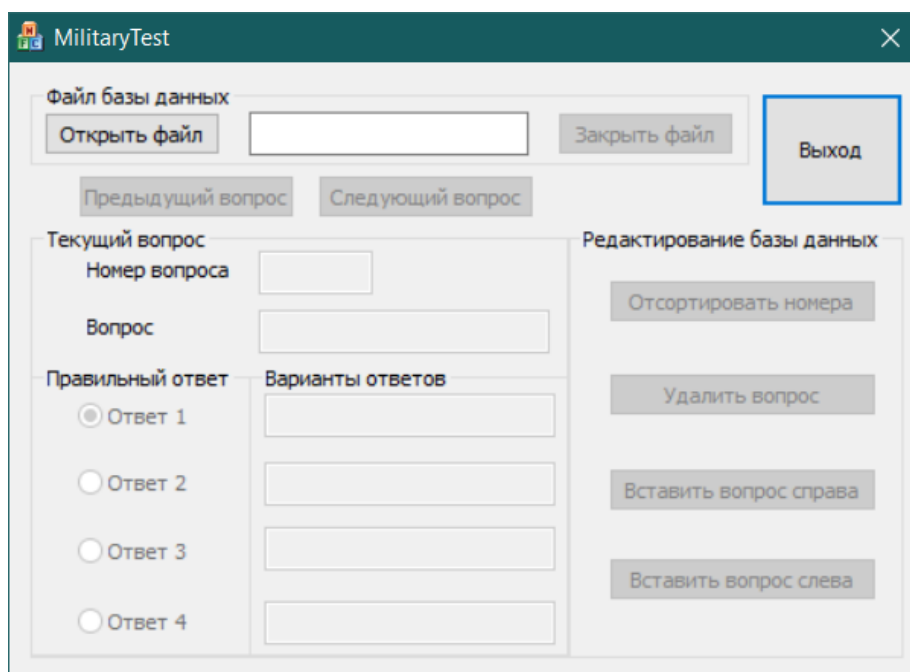


Рисунок 1.3 – Главное меню приложения MilitaryTest

К сожалению, на данный момент не существует полноценных аналогов тренажёра для Р-180.

Радиостанция Р-180 – УКВ диапазона, малой мощности второй подгруппы, приемопередающая, симплексная, предназначена для обеспечения связи в тактическом звене управления, несекретных переговоров и передачи данных с повышенной помехозащищенностью и скрытностью в тактическом звене управления Вооруженных Сил (взвод-рота).

Радиостанция выполнена в виде моноблочной конструкции, не требующей защиты от вибрации и принудительной вентиляции.

Габаритные размеры радиостанции в составе приемопередатчика и блок аккумуляторов, включая выступающие части, составляют 75х60х270 мм. Масса радиостанции составляет 1,63 кг.

К радиостанции подключаются микрофонно-телефонная гарнитура и одна из антенн: AD-44/CW-TA-30-512, AD-44/CW-AS-30-512. Масса антенн 0,12 кг и 0,152 кг соответственно. Масса микрофонно-телефонной гарнитуры 0,319 кг.

Основные тактико-технические характеристики:

Диапазоны рабочих частот:

- от 30 до 108 МГц;
- от 108 до 146 МГц;
- от 146 до 174 МГц.

Радиостанция обеспечивает обмен:

- речевой информацией в аналоговом режиме;
- речевой информацией в цифровом режиме;
- цифровыми данными.

Радиостанция обеспечивает одно-двухчастотную симплексную радиосвязь при обмене аналоговой и цифровой информацией.

Радиостанция обеспечивает следующие режимы работы:

- режим фиксированной рабочей частоты (ФРЧ) во всех диапазонах;
- режим псевдослучайной перестройки рабочей частоты (ППРЧ) в любом из поддиапазонов при обмене цифровой информацией.

Количество каналов радиостанции: 100.

В радиостанции применяются следующие виды модуляции:

- фазовая (частотная (класс излучения G3E)) в режиме ФРЧ во всех диапазонах;
- однополосная с полной несущей на верхней боковой полосе (класс излучения H3E) в режиме ФРЧ во всех диапазонах;
- четырехпозиционная частотная (класс излучения F1W) в режимах ФРЧ, ППРЧ, во всех диапазонах.

Пиковая мощность передатчика радиостанции для класса излучения H3E находится в диапазоне от 4 до 7 Вт.

Мощность несущей передатчика радиостанции классов излучения G3E, F1W:

- номинальная от 0,4 до 1,1 Вт;
- повышенная от 3 до 5 Вт.

Чувствительность приемника в аналоговом режиме при отношении сигнал/шум 12 дБ: не хуже 1,5 мкВ; чувствительность приемника в цифровом режиме при относительном уровне битовых ошибок 5 % не более минус 101 дБм и минус 98 дБм при частотном разnose между каналами 12,5 кГц и 25 кГц соответственно.

Напряжение питания радиостанции составляет $10,8 \pm 1,8$ В и обеспечивается литий-ионной аккумуляторной батареей.

Ниже представлен общий вид аналога радиостанции на рисунке 1.4, а также вид передней панели радиостанции на рисунке 1.5 [2].



Рисунок 1.4 – Общий вид радиостанции Р-180



Рисунок 1.5 – Вид передней панели радиостанции Р-180

1.2 Постановка задачи

В рамках данного курсового проекта планируется разработка программного средства «Онлайн-тренажёр Р-180».

Будут разработаны алгоритмы записи, передачи, воспроизведение данных и интерактивное меню.

В программном средстве планируется реализовать следующие функции:

- настройка частот приема и передачи;
- меню радиостанции;
- одновременная работа большого числа пользователей;
- работа в телеграфном режиме;
- работа в телефонном режиме;
- работа в амплитудной и частотной модуляциях;
- выбор и передача рабочей частоты приема и передачи;
- запись переговоров пользователей;
- изменение фона;
- настройки доступов радиостанции;
- изменение паролей доступа;
- настройка подсветки клавиш;
- автоблокировка радиостанции;
- выключение экрана по истечению времени;
- тест для проверки готовности радиостанции.
- состояние выбранного режима;
- состояния подключенных модулей и текущих режимов;
- состояние частоты передачи и приема;
- алгоритмы записи, передачи и воспроизведения данных;
- взаимодействие структурных блоков приложения.

Главной задачей является передать в полном объеме процесс эксплуатации радиостанцией, а также приблизиться к максимально точной ее копии.

Для разработки программного средства будут использоваться языки программирования Node.js, JavaScript, CSS, HTML операционная система Windows 10.

2 РАЗРАБОТКА ПРОГРАММНОГО СРЕДСТВА

2.1 Структура программы

Данное приложение включает в себя две составляющие: серверную часть, обеспечивающая обмен данными между пользователями, и клиентскую часть, в которой требуется использовать четыре структурных блока:

- clientLogicBroadcasting – модуль, отвечающий за логику обработки сообщений, полученных с сервера, также для отправки аудиопакеты на сервер для последующей их передачи другим пользователям;
- menuRadiostation – модуль, отвечающий за состояния радиостанции;
- menuLogic – модуль, отвечающий за логику меню;
- clientControls – модуль, отвечающий за основные контроллеры управления радиостанцией.

2.2 Интерфейс программного средства

Внешний вид и удобность в использовании являются одними из главных критериев качества программного средства. Поэтому взаимодействие приложения с пользователем необходимо организовать максимально интуитивно и просто.

Интерфейс программного средства полностью соответствует оригинальному, взаимодействие с ним происходит также, как и на оригинальной радиостанции. Он получился компактным и позволяет работать сразу с несколькими приложениями одновременно с одного устройства. Главное окно программы представлено на рисунке 2.1.



Рисунок 2.1 – Главное окно программы

Кроме основного интерфейса, необходимо было уделить особое внимание мелочам в верхнем меню состояния радиостанции: время, аудиовыход, мощность, сеть, состояние батареи и тип вещания.

В меню состояний время – HTML – блок, который каждую секунду берет текущее время и вписывает его, код представлен на рисунке 2.2.

```
async startTime() {  
    menuRadiostation.statusTime = true;  
    menuRadiostation.divTime.textContent = getTime(new Date());  
  
    this.timeInterval = setInterval(async () => {  
        menuRadiostation.divTime.textContent = getTime(new Date());  
    }, 1000);  
}
```

Рисунок 2.2 – Код отображения текущего времени

Все остальные элементы меню состояния – картинки, которые меняются при работе с радиостанцией.

2.3 Запись и передача звука на сервер

Радиостанция Р-180 работает в симплексной и двухчастотной-симплексной связях. Симплексная связь — связь, при которой информация передаётся только в одном направлении, при двухчастотной симплексной связи, прием и передача ведутся на разных частотах. Это значит, что вещать одновременно может только одна радиостанция, а другие в этот момент могут ее слушать.

Для того чтобы установить непрерывную передачу звука, отсылаются аудиоданные. Эмпирическим путем было установлено, что оптимальная продолжительность аудио равна 500 миллисекунд, при меньшей продолжительности связь будет прерывистой, при большей продолжительности будет большая задержка в вещании. Также для записи переговоров параллельно ведется цельная запись аудио, которая в конце вещания передается на сервер. Решение о параллельной записи было принято из-за сложности склеивания аудиодорожек, такую запись можно скачать для повторного воспроизведения. Во время вещания заблокирована функция приема. Алгоритм записи и передачи звука на сервер представлен на рисунке 2.3.

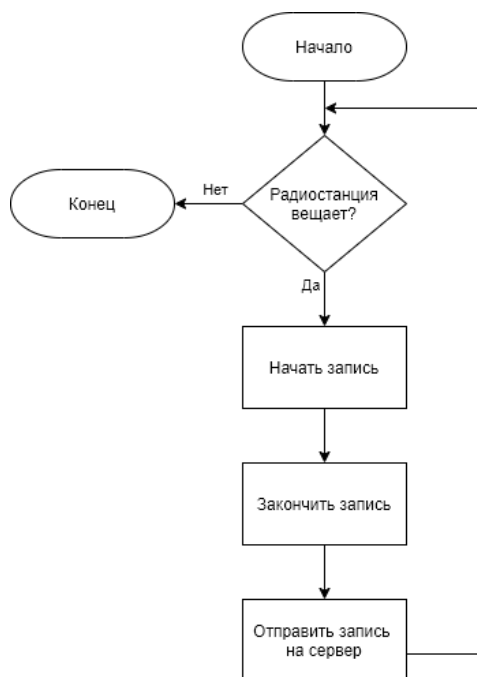


Рисунок 2.3 – Алгоритм записи и передачи звука на сервер

Для того чтобы передать телеграфный звук, используется другой подход, т.к. здесь сильно влияет задержка и минимальное время аудиодорожки. Чтобы реализовать нормальную поддержку телеграфной связи, было решено изначально вместе со страницей загружать аудиодорожку телеграфного звука. Чтобы передать телеграфный звук, передаются данные в которых указан тип вещания и если это телеграфное вещание, то воспроизводится телеграфный звук, до той поры, пока не придут данные о конце вещания. Эмпирическим путем было выявлено, что 100 миллисекунд – это оптимальное время.

2.4 Обработка данных сервером

Сервер должен: принимать аудиоданные, отдавать аудиоданные, а также отдавать элементы страницы (HTML, JS, CSS, картинки и т.д.).

Для того чтобы было проще пользователю, при не правильном вводе абсолютной ссылки, сервер автоматически его перенаправит на нужную страницу, код представлен на рисунке 2.4 [3].

```

app.use("/", (req, res) => {
  res.redirect("/html/index.html");
});

```

Рисунок 2.4 – Код перенаправления сервером пользователя на нужную ссылку

Код отвечающий за отправку элементов страницы пользователю представлен на рисунке 2.5.

```
app.use(express.static(__dirname.slice().replace(/\\[^\\]*$/, "/, "")));
```

Рисунок 2.5 – Код отправки элементов страницы пользователю

Сервер вещает, откуда на него поступают данные от пользователей, код показан на рисунке 2.6.

```
io.on("connection", (socket) => {  
  socket.on("stream", async (audio) => {  
    socket.broadcast.emit("stream", audio);  
  });  
});
```

Рисунок 2.6 – Код вещания полученных данных сервером

Сервер умеет вести запись разговоров пользователей, чтобы начать вести запись, было принято решение при запуске передавать параметром текст о ведении записи, после чего сервер будет передавать аудиозаписи переговоров, также, как и при вещании.

Для того чтобы поддерживать ведение связи на любых устройствах, таких как мобильный телефон, надо установить защищенное соединение. Для создания защищенного соединения на сервере, необходим сертификат, но он стоит денег. Если создать сертификат без подтверждения, то сервер будет работать по протоколу https, но пользователей при подключении к радиостанции уведомят о небезопасности перехода на сайт. Данное решение позволяет вести переговоры по радиостанции с большого количества устройств. Код создания сервера по протоколу https представлен на рисунке 2.7 [4].

```
const express = require("express");  
const app = express();  
const fs = require("fs");  
const https = require("https");  
const options = {  
  key: fs.readFileSync("localhost.key"),  
  cert: fs.readFileSync("localhost.crt"),  
};  
const server = https.createServer(options, app);  
const io = require("socket.io")(server);  
  
const port = 1000;  
server.listen(port);
```

Рисунок 2.7 – Код создания сервера по протоколу https

2.5 Меню программы

Для создания меню будут нужны три переменные: текущее меню, текущая цифра пункта в этом меню и код нажатой клавиши. При нажатии кнопки вызывается функция меню, куда передается текущее меню, цифра пункта и код клавиши, после чего определяется действие на текущей странице. Алгоритм меню представлен на рисунке 2.8.

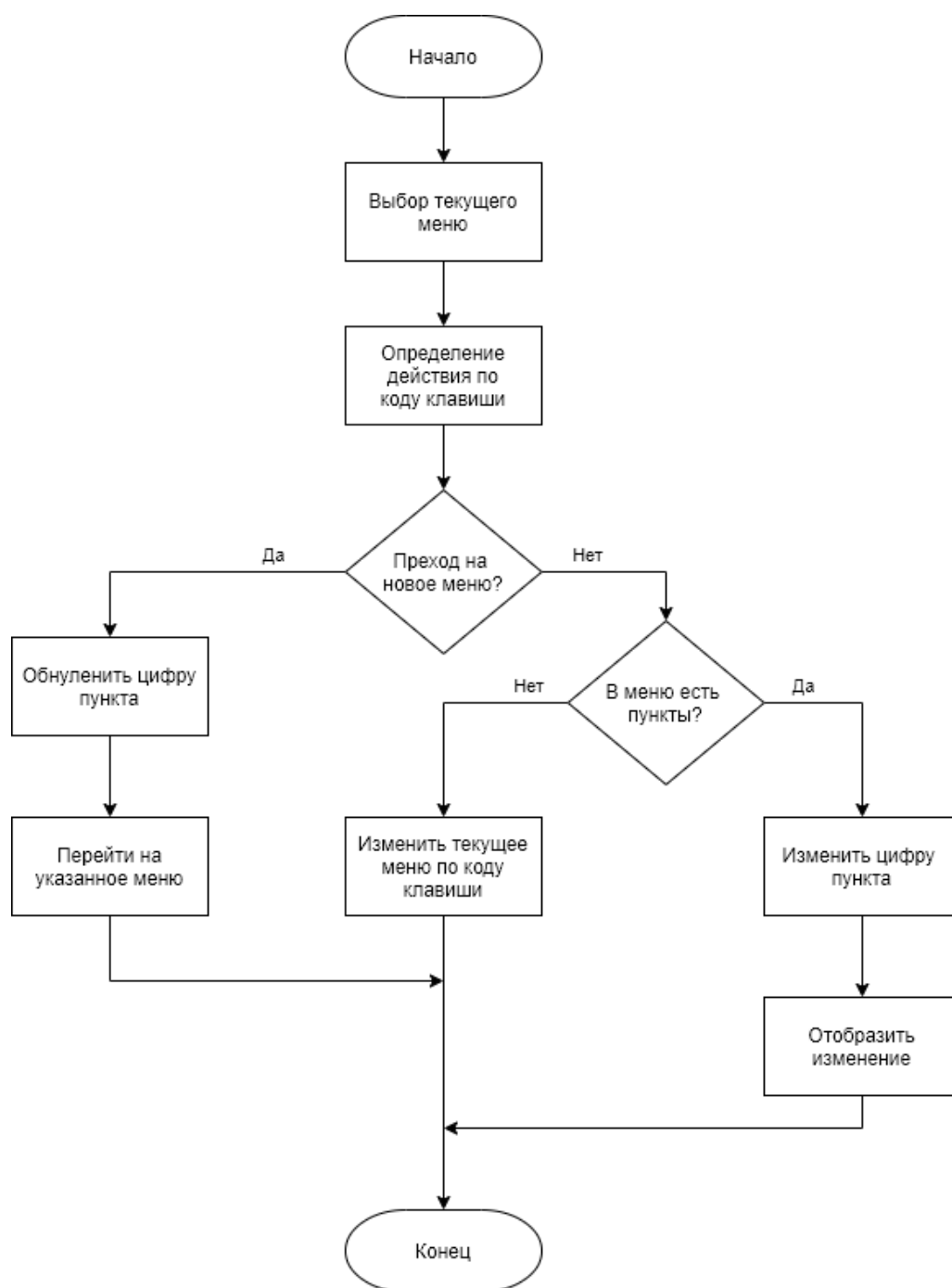


Рисунок 2.8 – Алгоритм меню

2.6 Воспроизведение полученных с сервера данных

После того как пользователь получает данные от сервера, необходимо определиться к какому виду они относятся (вещание, запись, телеграфный звук). Если полученные данные являются аудиовещанием, то из них создается новое аудио и воспроизводится, код показан на рисунке 2.9 [5].

```
socket.on("stream", async (stream) => {  
    const audioBlob = new Blob(Array(stream.audioChunks));  
    const audioUrl = URL.createObjectURL(audioBlob);  
    const audio = new Audio(audioUrl);  
    audio.volume =  
        (stream.speakVolume * menuRadiostation.volume) / (24 * 24);  
    await audio.play();  
});
```

Рисунок 2.9 – Код воспроизведения полученного вещания

Если полученные данные являются записью, то создается новый элемент в журнале аудиозаписей, с меткой времени, частоты и модуляции по которой было передано сообщение. Алгоритм представлен на рисунке 2.10.



Рисунок 2.10 – Создание новой аудиозаписи

Если полученные данные являются телеграфным звуком, то запускается уже подгруженное его аудио, если телеграфный вызов больше длины аудиозаписи, то аудиозапись повторяется. В конце телеграфного вызова приходит сообщение о его конце и воспроизведение прекращается, после чего время воспроизведения переводится в начало аудиозаписи.

2.7 Настройка частот

Настройка частот приема и передачи, а также модуляции с шириной канала работают по каналам. При настройке частоты, заданная частота должна вмещаться в диапазон радиостанции, а также быть кратной минимальной ширине канала. Если во время настройки канала сменить частоту, то радиостанция должна продолжить настройку по предыдущему. Если в конце выбрать большую ширину канала, и заданная частота не будет кратна ей, то частота автоматически сменится на ближайшую кратную. Алгоритм представлен на рисунке 2.11.

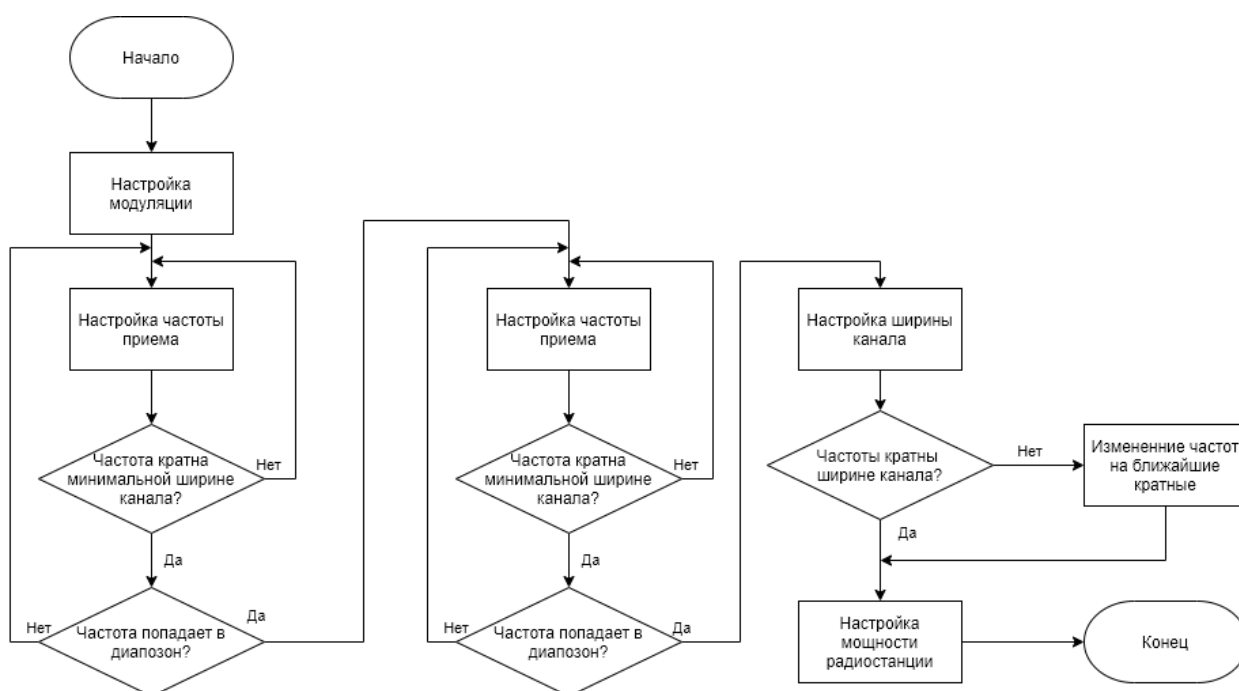


Рисунок 2.11 – Алгоритм настройки частот

3 ТЕСТИРОВАНИЕ ПРОГРАММНОГО СРЕДСТВА

В ходе тестирования приложения не было выявлено недостатков программного средства. Была составлена таблица 3.1, показывающая ожидаемые и реальные результаты, полученные при заданных условиях, она представлена ниже.

Таблица 3.1 – Ожидаемые и реальные результаты тестирования

№	Тестовые случаи	Ожидаемый результат	Полученный результат
1.	Передача аудиоданных на сервер	Успешная передача данных за конечное время	Тест пройден
2.	Подключение к серверу клиента	Успешное подключение	Тест пройден
3.	Смена рабочей частоты	Изменение частоты	Тест пройден
4.	Смена режима передачи	Смена посылаемых данных	Тест пройден
5.	Непрерывная запись голоса	Безотказная работа приложения	Тест пройден
6.	Отключить батарею во время работы	Выключение радиостанции	Тест пройден
7.	Отключение сервера во время работы клиента	Остановка передачи данных	Тест пройден
8.	Включение режима автоблокировки	Автоблокировка радиостанции через 7 секунд	Тест пройден
9.	Отключение антенны во время передачи	Остановка передачи	Тест пройден
10.	Настройка громкости приема	Изменение громкости приема	Тест пройден
11	Настройка выключения экрана через 5 секунд	Выключение экрана через 5 секунд	Тест пройден
12	Включение подсветки клавиатуры	Включение подсветки	Тест пройден
13	Сменить канал	Смена канала	Тест пройден

Разработка онлайн тренажёра Р-180 велась с использованием системы контроля версий GitHub, позволившая сохранять состояние программы на

каждом отдельном этапе по ходу добавления нового функционала или изменения уже существующего. Появление новых точек возврата происходит посредством группировки изменённых файлов, затем они объединяются под общим именем «коммита», в котором кратко изложена суть изменений. Также можно добавлять к каждому этапу новые файлы, или удалять устаревшие варианты. После накопления определённого количества групп изменений, их следует отправить на удалённый репозиторий, где видна вся история приложения и разница между каждым новым «коммитом».

Путем тщательной проверки тестами, было выявлено, что ошибок в работе программы нет.

4 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

4.1 Руководство по установке и запуску

Перед началом работы в радиостанции необходимо скачать Node.js по ссылке: <https://nodejs.org/en/download/>. После чего скачать само программное средство. Чтобы начать общение по радиостанции, необходимо включить сервер. Это делается следующим образом:

- открыть меню пуск;
- найти там «Командная строка»;
- после чего перейти к папке с радиостанцией, чтобы это сделать надо пользоваться командой «cd». Чтобы перейти в новую папку напишите «cd название папки», пример «cd folder», чтобы перейти на другой диск напишите «Имя диска:», пример «D:», чтобы перейти на папку назад «cd ..». Составные имена папок должны быть в кавычках;
- после перехода в папку радиостанции зайдите в папку «node js»;
- чтобы запустить сервер напишите «node server.js», чтобы запустить сервер с записью переговоров «node server.js recording».

Теперь сервер запущен, осталось только узнать IP сервера. Чтобы это сделать, снова откройте командную строку, и напишите там ipconfig. На экране появятся данные, среди которых должна быть строка IPv4. Рядом с ней вы увидите код формата 192.168.1.40 — это и есть IP-адрес сервера, рисунок 4.1.

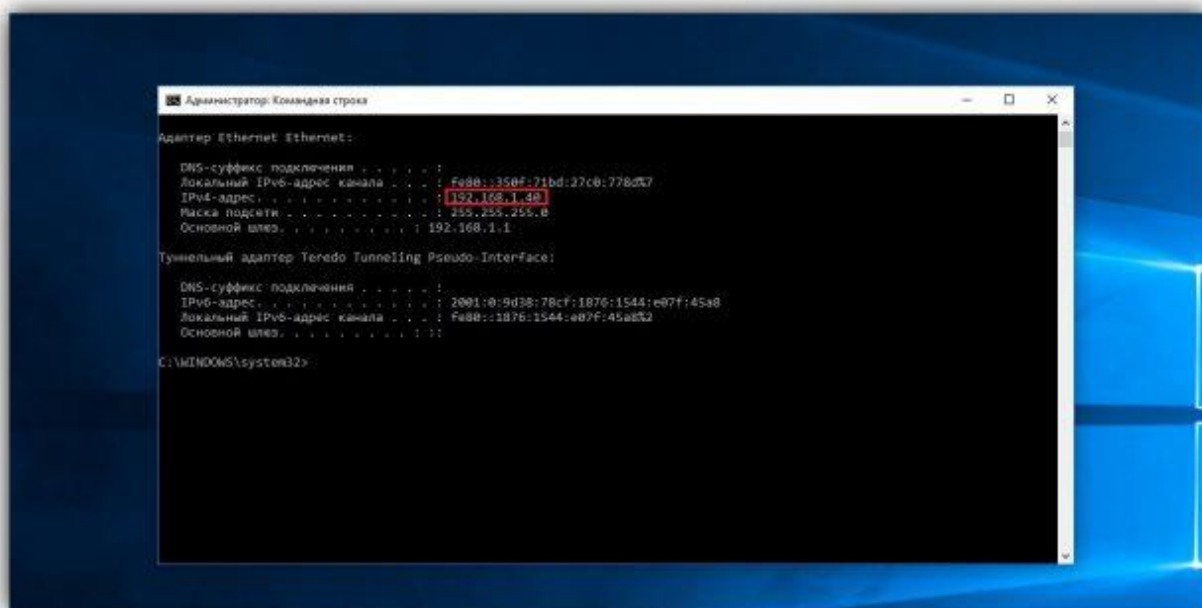


Рисунок 4.1 – Результат поиска IP

После нахождения IP, чтобы перейти к радиостанции надо ввести «https://IP:1000», пример «https://192.168.43.82:1000». Чтобы зайти в радиостанцию с компьютера на котором стоит сервер, напишите «https://localhost:1000».

4.2 Руководство по использованию

Интерфейс программного средства может показаться неудобным или неочевидным, но разработан он на основе настоящей радиостанции. Такое решение позволяет получить максимум практических навыков в работе с радиостанцией и приведёт к минимуму ошибок при эксплуатации реальных аналогов.

4.2.1. Работа с радиостанцией

Чтобы включить радиостанцию нужно подключить все устройства, для этого нажмите на кнопку, после чего подключите все устройства, рисунок 4.2.



Рисунок 4.2 – Подключение устройств к радиостанции

После подключения устройств нажмите на кнопку включения. Вы попадете на главный экран радиостанции. С главного экрана можно перейти в: меню (буква «М»), настройку громкости приема (цифра «1»), тест радиостанции (цифра «0»). Чтобы сменить окно показа частоты на прием или передачу (цифра «2»), сменить мощность радиостанции (цифра «7»).

В меню (рисунок 4.3), можно перейти к: настройке каналов, параметрам, информации о радиостанции, настройке доступа и тесту. Чтобы перейти к

нижнему пункту нажми цифру «8», чтобы перейти к верхней цифре «2», чтобы перейти по пункту нажмите клавишу «Enter».

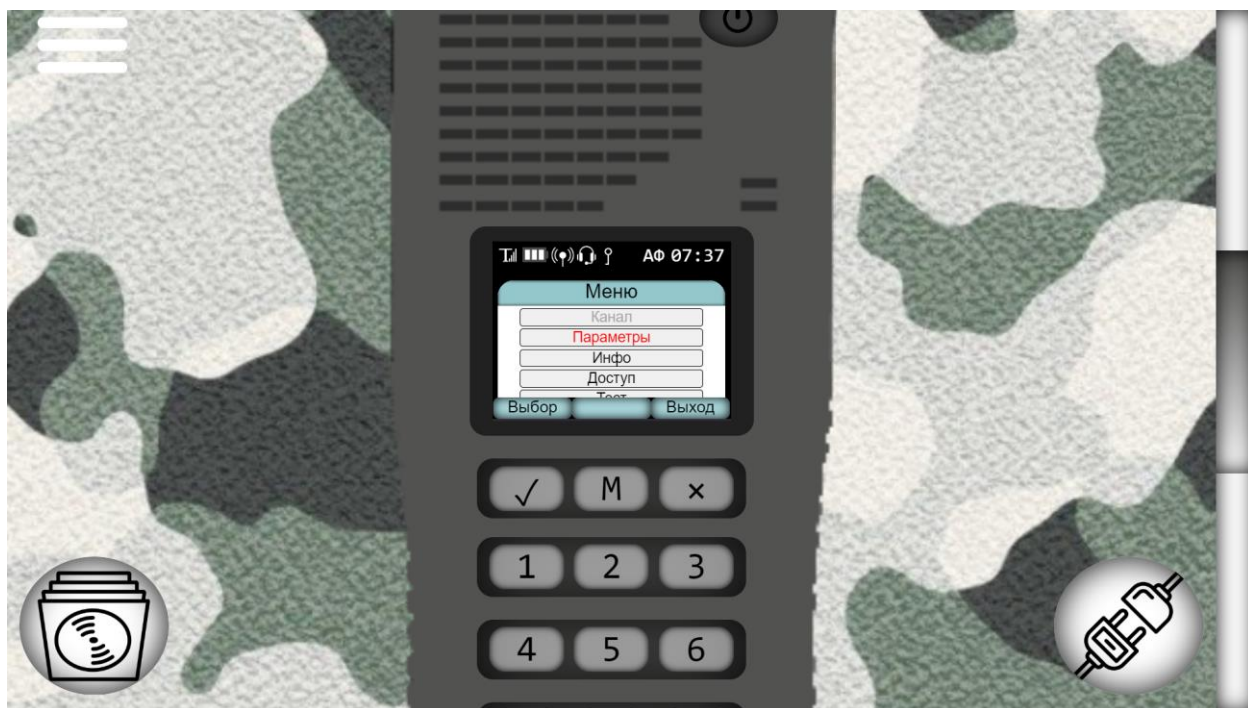


Рисунок 4.3 – Меню радиостанции

В параметрах (рисунок 4.4) вы можете настроить: громкость микрофона (рисунок 4.5), автоблокировку (рисунок 4.6), подсветку клавиатуры (рисунок 4.7), а также выключение экрана (рисунок 4.8).

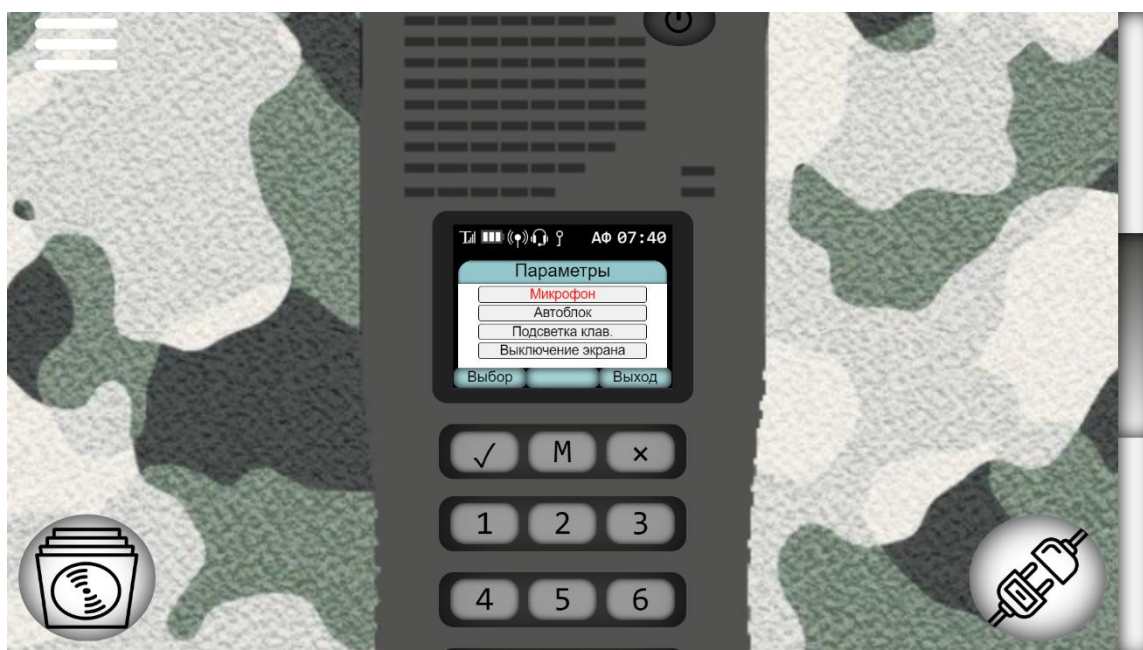


Рисунок 4.4 – Параметры радиостанции



Рисунок 4.5 – Настройка громкости микрофона



Рисунок 4.6 – Настройка автоблокировки



Рисунок 4.7 – Настройка подсветки клавиатуры

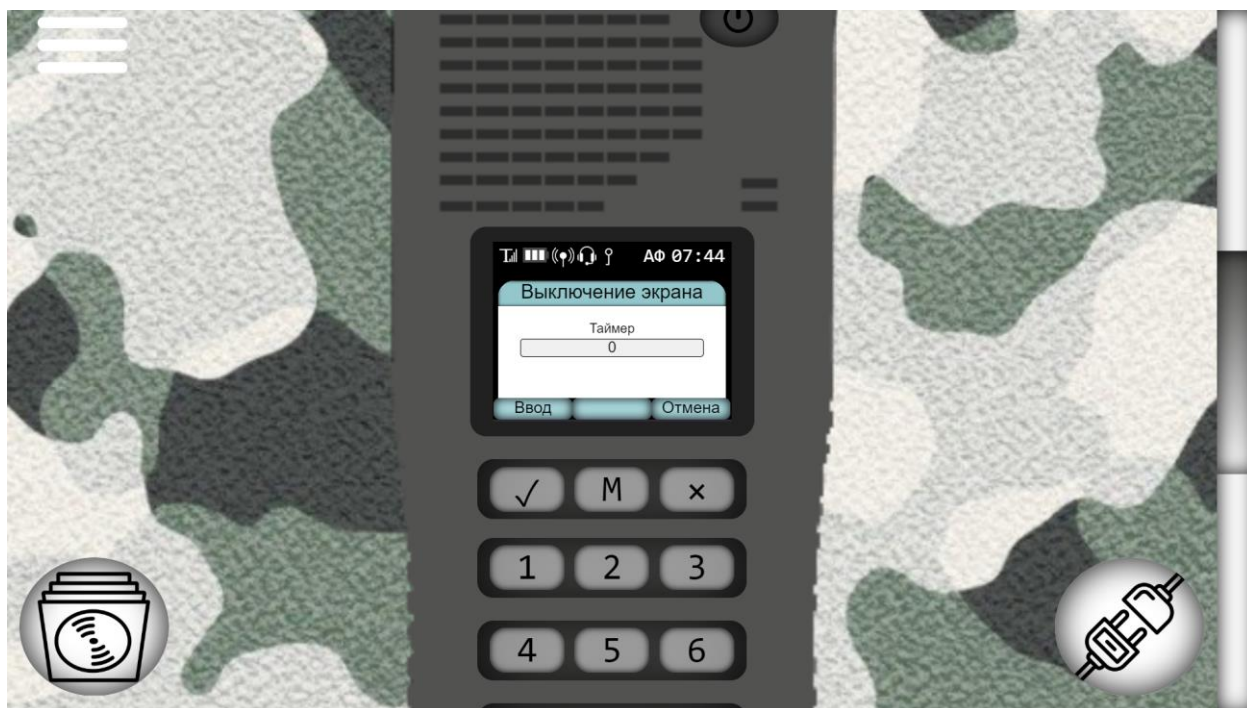


Рисунок 4.8 – Настройка таймера выключения экрана

В информации о радиостанции вы можете посмотреть температуру (рисунок 4.9) и версию радиостанции (рисунок 4.10).



Рисунок 4.9 – Текущая температура радиостанции

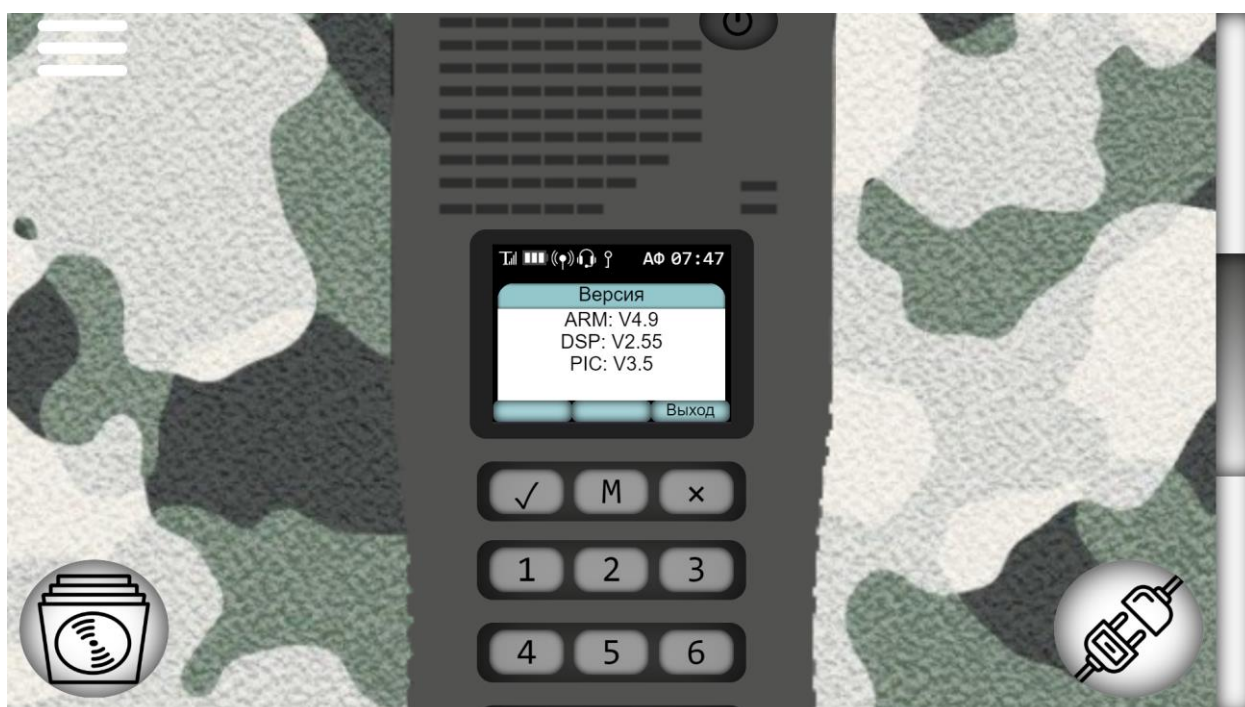


Рисунок 4.10 – Текущая версия радиостанции

В пункте «Доступ» вы можете выбрать режим настройки радиостанции (рисунок 4.11), для того чтобы вы смогли настроить ее на частоту, сменить пароль к радиостанции (изначальный пароль «0000»), войти в режим пользователя или режим обзора.



Рисунок 4.11 – Вход в режим настройки радиостанции

В пункте «Тест» (рисунок 4.12) вы можете проверить готовность радиостанции к работе.

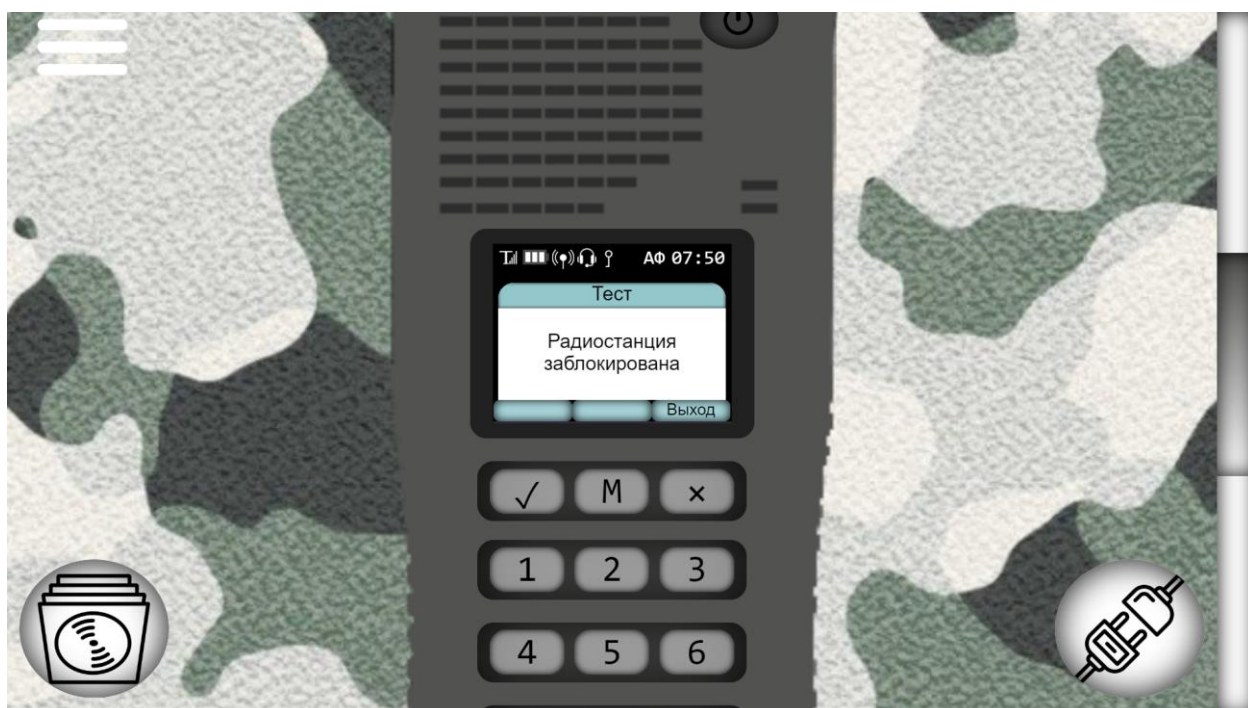


Рисунок 4.12 – Тест готовности радиостанции

С помощью пункта «Канал» вы можете настроить радиостанцию.

Для смены канала наведите на основание антенны, там находится переключатель каналов, нажмите на одну из стрелок для смены.

4.2.2. Порядок настройки частоты

Чтобы начать настройку радиостанции, необходимо зайти в «Доступ» и выбрать «Настройка», после чего мы получим доступ к настройке радиостанции. После чего надо будет выбрать модуляцию, частоту приема и передачи (в диапазоне 30-174МГц), выбрать ширину канала и мощность, рисунок 4.13. После чего нажмите на «ТНГ» или «ПРД», для передачи голоса. «ВЫЗ» или цифра «3» в главном меню для передачи телеграфного звука. Чтобы открыть контроллеры управления передачей, нажмите на кнопку в верхнем левом углу.

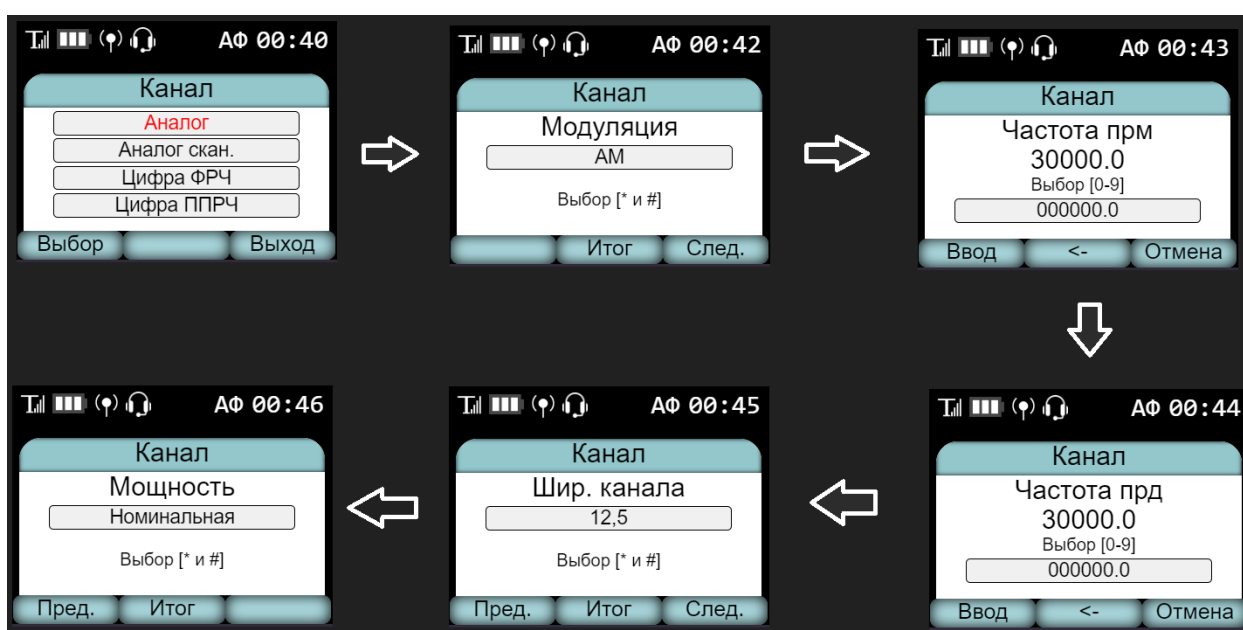


Рисунок 4.13 – Порядок настройки радиостанции

4.2.3. Расширенные возможности

Для тех кому надоел старый интерфейс есть возможность его сменить, чтобы это сделать нажмите три раза по фоновой картинке и вставьте ссылку на новую, рисунок 4.14.



Рисунок 4.14 – Смена фона

Также при включенном режиме записи переговоров, можно их повторно прослушать или загрузить, рисунок 4.15.

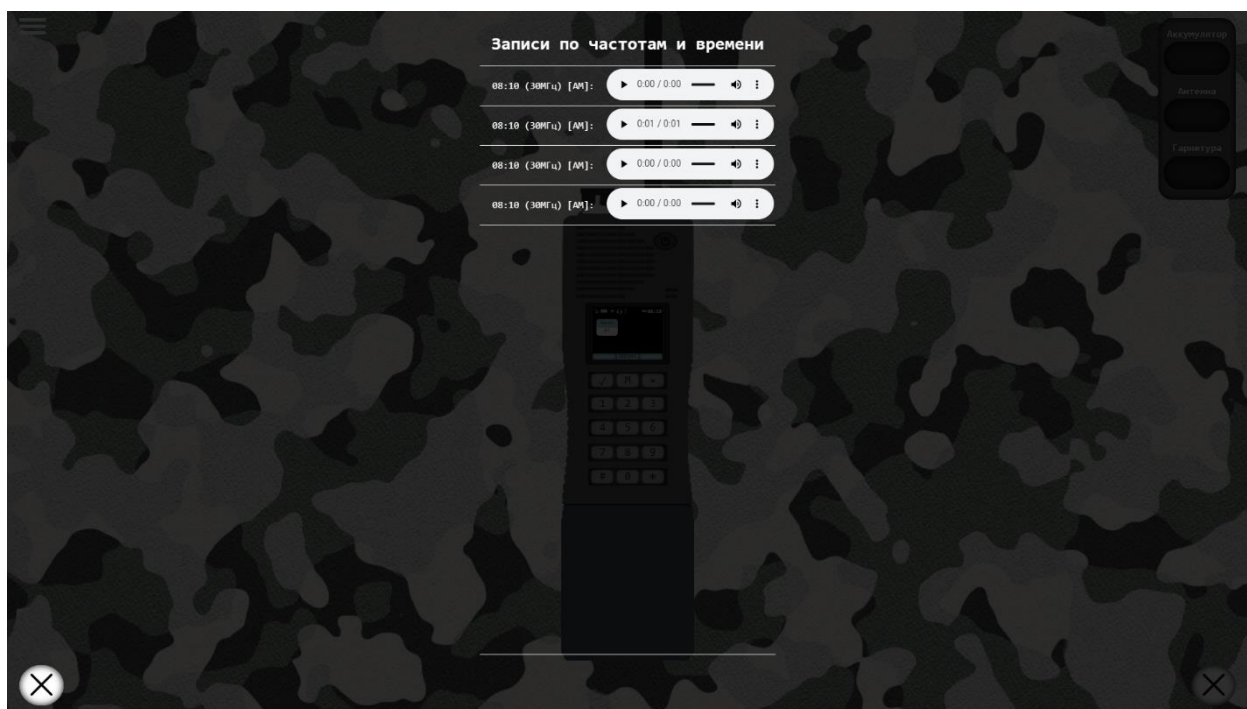


Рисунок 4.15 – Журнал аудиозаписей

ЗАКЛЮЧЕНИЕ

Развитие телекоммуникаций не прекращает расти, человечество с каждым годом достигает новых скоростей и качества передачи данных, но новые знания вырабатываются на основе предыдущих. Поэтому так важно не только знать теоретические основы радиосвязи, но и применять их на практике.

В рамках данного курсового проекта было разработано программное средство «Онлайн-тренажёр Р-180», которое обеспечит полезное и увлекательное времяпрепровождение. Работа была выполнена в полном объеме, были реализованы все поставленные функции.

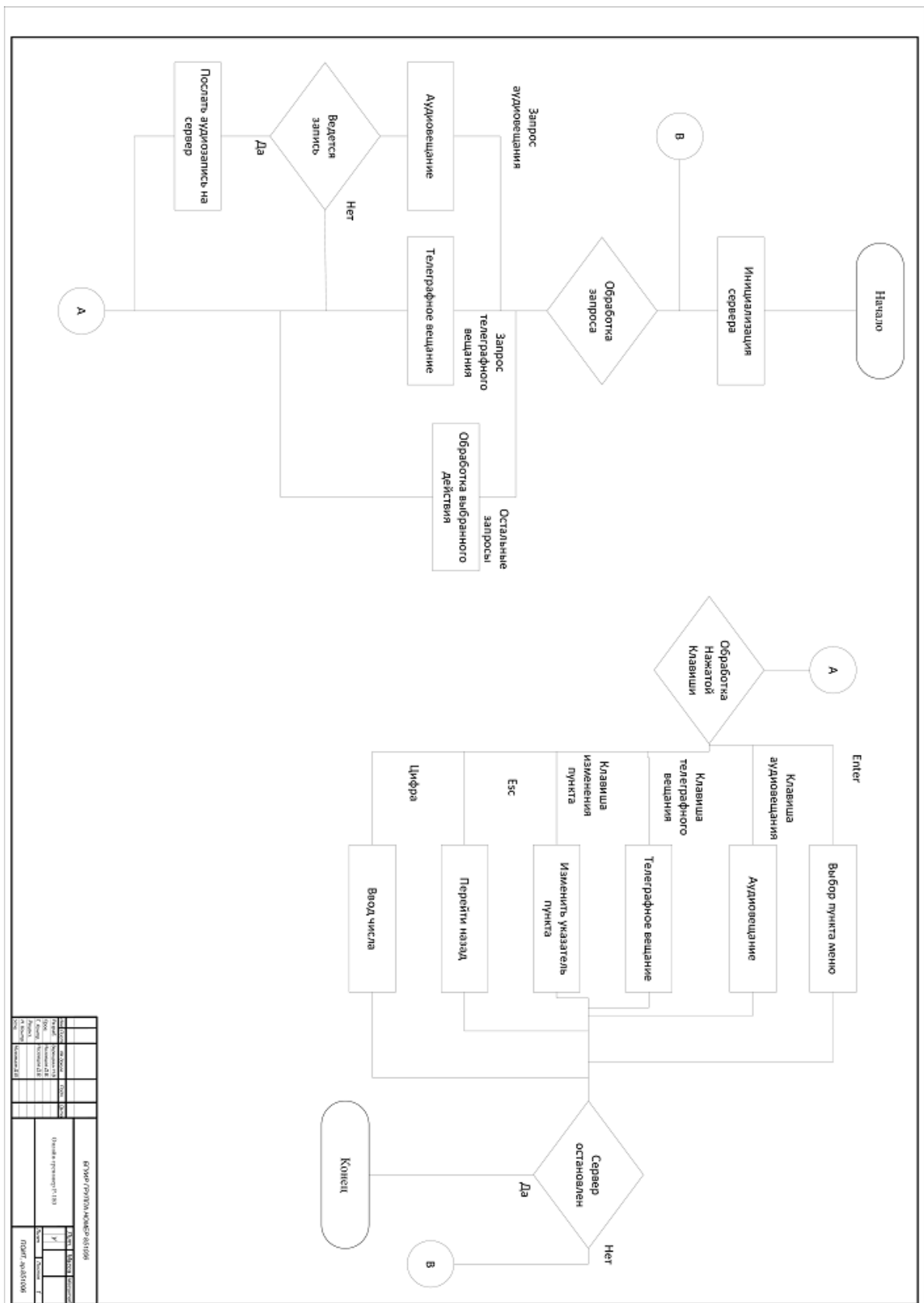
Существует много возможностей для дальнейшего улучшения приложения. Одним из самых простых направлений является добавление подробного описания настройки и включения приложения. Также можно добавить: рабочую «3d» модель радиостанции, таблицу «ТДР-84» и таблицу «Альфа». Клиентская часть может быть дополнена другими радиостанциями, а серверная часть – интерфейсом с таблицей подключённых пользователей, их позывных и качества сети.

Использование данного приложения позволит не только провести время с пользой, но и стимулирует работу памяти, а также даст полезные навыки в работе с радиостанцией.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- [1] www.spb.kp.ru [Электронный портал]. – Электронные данные. – Режим доступа: <https://www.spb.kp.ru/daily/26954.5/4007387>
- [2] agat-system.by [Электронный портал]. – Электронные данные. – Режим доступа: <http://agat-system.by/produktsiya/produktsiya-spetsialnogo-naznacheniya/sredstva-radiosvyazi/nosimaya-radiostantsiya-ukv-diapazona-r-180.html>
- [3] metanit.com [Электронный портал]. – Электронные данные. – Режим доступа: <https://metanit.com/web/nodejs/4.13.php>
- [4] nodejs.org [Электронный портал]. – Электронные данные. – Режим доступа: <https://nodejs.org/api/https.html>
- [5] stackoverflow.com [Электронный портал]. – Электронные данные. – Режим доступа: <https://stackoverflow.com/questions/34934862/how-to-replay-an-audio-blob-in-javascript>

ПРИЛОЖЕНИЕ 1 Схема алгоритма работы



ПРИЛОЖЕНИЕ 2

Исходный код программы

Файл clientLogicBroadcasting.js:

```
const socket = io();
const recordLength = 500;
const beepLength = 100;
const [broadcastButton] = document.getElementsByClassName("broadcast_button");
const [_broadcastButton] = document.getElementsByClassName("broadcast");
const [callButton] = document.getElementsByClassName("call");
const beep = new Audio("../music/beep.mp3");

const userFrequencysOut = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0];
const userFrequencysIn = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0];
const userModulation = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0];
const userChannelWidth = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0];
let channel = 0;

function recordAudio(data) {
  return new Promise((resolve) => {
    navigator.mediaDevices.getUserMedia =
      navigator.mediaDevices.getUserMedia ||
      navigator.mediaDevices.webkitGetUserMedia ||
      navigator.mediaDevices.mozGetUserMedia ||
      navigator.mediaDevices.msGetUserMedia;

    navigator.mediaDevices.getUserMedia({ audio: true }).then((stream) => {
      const mediaRecorder = new MediaRecorder(stream);
      mediaRecorder.addEventListener("dataavailable", (event) => {
        socket.emit(data, {
          audioChunks: event.data,
          frequency: userFrequencysOut[channel],
          beep: false,
          speakVolume: menuRadiostation.speakVolume,
          modulation: userModulation[channel],
        });
      });
    });

    const start = () => {
      mediaRecorder.start();
    };

    const stop = () => {
      mediaRecorder.stop();
    };

    resolve({ start, stop });
  });
}
```

```

function stopBeep() {
  beep.pause();
  beep.currentTime = 0;
  menuRadiostation.beepOff();
}

socket.on("stream", async (stream) => {
  if (
    !menuRadiostation.statusBroadcasting &&
    menuRadiostation.statusWorking &&
    menuRadiostation.statusAntenna
  ) {
    if (
      stream.frequency === userFrequenciesIn[channel] &&
      stream.modulation === userModulation[channel]
    )
      try {
        if (stream.beep) {
          if (!menuRadiostation.statusBeep) {
            beep.volume = menuRadiostation.volume / 24;
            beep.play();
            menuRadiostation.beepOn();
          } else {
            if (beep.currentTime > 595) beep.currentTime = 0.142821;
          }
        } else {
          if (menuRadiostation.statusBeep) {
            stopBeep();
          } else {
            const audioBlob = new Blob(Array(stream.audioChunks));
            const audioUrl = URL.createObjectURL(audioBlob);
            const audio = new Audio(audioUrl);
            audio.volume =
              (stream.speakVolume * menuRadiostation.volume) / (24 * 24);
            await audio.play();
          }
        }
      } catch {}
  }
});

function getTime(date) {
  let hours = String(date.getHours());
  let minutes = String(date.getMinutes());
  if (hours.length < 2) hours = "0" + hours;
  if (minutes.length < 2) minutes = "0" + minutes;
  return hours + ":" + minutes;
}

socket.on("recording", async (record) => {

```

```

const [divParent] = document.getElementsByClassName("records_block");
const block = document.createElement("div");
const audioTag = document.createElement("audio");
const text = document.createElement("p");
const date = new Date();

```

```

const audioBlob = new Blob(Array(record.audioChunks), {
  type: "audio/mp3",
});
const audioUrl = URL.createObjectURL(audioBlob);
audioTag.controls = true;
audioTag.src = audioUrl;

```

```

let modulation = "";
if (record.modulation === 0) modulation = "AM";
if (record.modulation === 1) modulation = "ЧМ";

```

```

text.textContent =
  getTime(date) +
  " (" +
  String((record.frequency + 2400) / 80) +
  "MHz) [" +
  modulation +
  "].";

```

```

block.className = "audio_block";
block.appendChild(text);
block.appendChild(audioTag);
divParent.appendChild(block);
});

```

```

(async () => {
  broadcastButton.addEventListener("touchstart", isBroadcasting);
  broadcastButton.addEventListener("touchend", isNotBroadcasting);
  _broadcastButton.addEventListener("touchstart", isBroadcasting);
  _broadcastButton.addEventListener("touchend", isNotBroadcasting);
  callButton.addEventListener("touchstart", isBroadcastingBeep);
  callButton.addEventListener("touchend", isNotBroadcastingBeep);

```

```

  broadcastButton.addEventListener("mousedown", isBroadcasting);
  broadcastButton.addEventListener("mouseup", isNotBroadcasting);
  _broadcastButton.addEventListener("mousedown", isBroadcasting);
  _broadcastButton.addEventListener("mouseup", isNotBroadcasting);
  callButton.addEventListener("mousedown", isBroadcastingBeep);
  callButton.addEventListener("mouseup", isNotBroadcastingBeep);

```

```

const recorder = await recordAudio("stream");
const record = await recordAudio("record");

```

```

function isBroadcasting() {
  if (

```

```

    menuRadiostation.statusAntenna &&
    menuRadiostation.statusWorking &&
    !menuRadiostation.blocking
  ) {
    if (menuRadiostation.statusBeep) stopBeep();
    menuRadiostation.broadcastingOn();
    broadcasting();
  }
}

function isNotBroadcasting() {
  menuRadiostation.broadcastingOff();
}

menuRadiostation.isBroadcastingBeep = isBroadcastingBeep;
menuRadiostation.isNotBroadcastingBeep = isNotBroadcastingBeep;

function isBroadcastingBeep() {
  if (
    menuRadiostation.statusAntenna &&
    menuRadiostation.statusWorking &&
    !menuRadiostation.blocking
  ) {
    if (menuRadiostation.statusBeep) stopBeep();
    menuRadiostation.broadcastingOn();
    broadcastingBeep();
  }
}

function isNotBroadcastingBeep() {
  menuRadiostation.broadcastingOff();
}

function broadcasting() {
  if (menuRadiostation.statusBroadcasting) {
    recorder.start();
    record.start();

    const interval = setInterval(async () => {
      await recorder.stop();
      if (!menuRadiostation.statusBroadcasting) {
        clearInterval(interval);
        await record.stop();
      } else await recorder.start();
    }, recordLength);
  }
}

function broadcastingBeep() {
  const interval = setInterval(async () => {
    if (!menuRadiostation.statusBroadcasting) {

```

```

    socket.emit("stream", {
      audioChunks: 0,
      frequency: userFrequencyOut[channel],
      beep: false,
      modulation: userModulation[channel],
    });
    clearInterval(interval);
  } else
    socket.emit("stream", {
      audioChunks: 0,
      frequency: userFrequencyOut[channel],
      beep: true,
      modulation: userModulation[channel],
    });
  }, beepLength);
}
})();

```

```

console.error = () => {};

```

Файл server.js:

```

const express = require("express");
const app = express();
const fs = require("fs");
const https = require("https");
const options = {
  key: fs.readFileSync("localhost.key"),
  cert: fs.readFileSync("localhost.crt"),
};
const server = https.createServer(options, app);
const io = require("socket.io")(server);

const port = 1000;
const property = process.argv;

let flagRecording = false;
if (property.length > 2 && property[2].toLowerCase() === "recording") {
  flagRecording = true;
  console.log("Сервер ведет аудиозапись");
}

app.use(express.static(__dirname.slice().replace(/\\/[^\\]*$/, "")));

app.use("/", (req, res) => {
  res.redirect("/html/index.html");
});

io.on("connection", (socket) => {
  socket.on("stream", async (audio) => {
    socket.broadcast.emit("stream", audio);
  });
});

```

```

if (flagRecording)
  socket.on("record", async (audio) => {
    if (!audio.beep && audio.audioChunks !== 0)
      socket.broadcast.emit("recording", {
        audioChunks: audio.audioChunks,
        frequency: audio.frequency,
        modulation: audio.modulation,
      });
  });
});

```

```

console.log(`Сервер запущен на порте: ${port}`);
server.listen(port);

```

Файл index.html:

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <link rel="shortcut icon" href="../img/logo.png" type="image/x-icon" />
    <link rel="stylesheet" href="../css/main.css" />
    <link rel="stylesheet" href="../css/menu.css" />
    <title>Радиостанция Р-180</title>
  </head>
  <body class="body_hidden_thumb">
    <div class="loading"></div>

    <div class="open_controls_button" title="Контроллеры управления">
      <div></div>
      <div></div>
      <div></div>
    </div>

    <div class="controls_close">
      <div class="broadcast_button_block">
        <div class="broadcast_button_top_bolt"></div>
        <p class="broadcast_button_text">ТНГ</p>
        <div class="broadcast_button"></div>
        <div class="broadcast_button_bottom_bolt"></div>
      </div>
      <div class="controls_call_broadcast_close">
        <p class="call_text">
          БЫЗ
        </p>
        <div class="call"></div>
        <p class="broadcast_text">
          ПРД
        </p>
        <div class="broadcast"></div>
      </div>
    </div>
  </body>
</html>

```

```

</div>
</div>

<div
  class="connect_controls_button"
  title="Подключение и отключение устройств"
></div>

<div class="connect_controls_block_close">
  <p>Аккумулятор</p>
  <div class="battery_button"></div>
  <p>Антенна</p>
  <div class="antenna_button"></div>
  <p>Гарнитура</p>
  <div class="headset_button"></div>
</div>

<div class="records_button" title="Записи"></div>

<div class="records_block_not_active">
  <h1 class="records_head_text">Записи по частотам и времени</h1>
  <div class="records_block"></div>
</div>

<div class="radiostation">
  <div class="radiostation_antenna_block">
    <div class="radiostation_antenna_not_active"></div>
  </div>
  <div class="radiostation_body">
    <div class="arrow_block">
      <div class="arrow_left"></div>
      <div class="arrow_right"></div>
    </div>
    <div class="radiostation_power_button"></div>
    <div class="radiostation_display">
      <!-- Info block -->
      <div id="top_block" class="not_active">
        <div class="network"></div>
        <div class="battery"></div>
        <div class="power"></div>
        <div class="sound_type"></div>
        <div class="key"></div>
        <div class="operating_mode">АФ</div>
        <div class="time"></div>
      </div>
      <!-- Menu display -->
      <div id="active_zone" class="not_active">
        <!-- Start menu-->
        <div class="not_active">
          <div class="common_block_channel">
            <div class="block_channel">

```



```

    <div id="top_text_channel">Канал</div>
    <div id="bottom_text_channel"></div>
  </div>
  <div class="not_active" id="frequency_call">
    <div id="common_top_text_channel">Адрес вызова (А)</div>
    <div id="bottom_text_frequency"></div>
  </div>
</div>
</div>
<!-- Main menu-->
<div class="not_active">
  <div class="menu_block" id="main_menu">
    <div class="top_text_menu">Меню</div>
    <div class="menu_point" id="block">Канал</div>
    <div class="menu_point" id="current_position">Параметры</div>
    <div class="menu_point" id="">Инфо</div>
    <div class="menu_point" id="">Доступ</div>
    <div class="menu_point" id="">Тест</div>
  </div>
</div>
<!-- Setting channel -->
<div class="not_active">
  <div class="menu_block" id="settings_channel">
    <div class="top_text_menu">Канал</div>
    <div class="menu_point" id="current_position">Аналог</div>
    <div class="menu_point" id="">Аналог скан.</div>
    <div class="menu_point" id="">Цифра ФРЧ</div>
    <div class="menu_point" id="">Цифра ППРЧ</div>
  </div>
</div>
<!-- Parameters menu -->
<div class="not_active">
  <div class="menu_block" id="parameters_menu">
    <div class="top_text_menu">Параметры</div>
    <div class="menu_point" id="current_position">Микрофон</div>
    <div class="menu_point" id="">Автоблок</div>
    <div class="menu_point" id="">Подсветка клав.</div>
    <div class="menu_point" id="">Выключение экрана</div>
  </div>
</div>
<!-- Info menu -->
<div class="not_active">
  <div class="menu_block" id="info_menu">
    <div class="top_text_menu">Инфо</div>
    <div class="menu_point" id="current_position">Температура</div>
    <div class="menu_point" id="">Версия</div>
  </div>
</div>
<!-- Access menu -->
<div class="not_active">
  <div class="menu_block" id="access_menu">

```

```

<div class="top_text_menu">Доступ</div>
<div class="menu_point" id="current_position">Пользователь</div>
<div class="menu_point" id="">Обзор</div>
<div class="menu_point" id="">Настройка</div>
<div class="menu_point" id="block">Смена пароля</div>
</div>
</div>
<div class="not_active">
<div class="menu_block" id="test_menu">
<div class="top_text_menu">Тест</div>
<div class="text_info"><br /></div>
<div class="text_info" id="test_menu_text"></div>
</div>
</div>
<!-- Temperature menu -->
<div class="not_active">
<div class="menu_block" id="temperature_menu">
<div class="top_text_menu">Температура</div>
<div class="text_info">Текущая<br /><br />35 C</div>
</div>
</div>
<!-- Version menu -->
<div class="not_active">
<div class="menu_block" id="version_menu">
<div class="top_text_menu">Версия</div>
<div class="text_info">
ARM: V4.9<br />DSP: V2.55<br />PIC: V3.5
</div>
</div>
</div>
</div>
<!-- Microphone menu -->
<div class="not_active">
<div class="menu_block" id="microphone_menu">
<div class="top_text_menu">Микрофон</div>
<div class="text_info">Уровень</div>
<div class="text_info" id="graph_speak_volume">
[|||||||]
</div>
<div class="text_info"><br /></div>
<div class="text_info" id="digit_speak_volume">24</div>
</div>
</div>
<!-- Auto blocking -->
<div class="not_active">
<div class="menu_block" id="autoblocking_menu">
<div class="top_text_menu">Автоблокировка</div>
<div class="text_info">Вкл/Выкл</div>
<div class="menu_point" id="text_blocking">Выкл</div>
<div class="little_letter_text_info">
<br />
Выбор [* и #]

```

```

    </div>
  </div>
</div>
<!-- Keyboard light -->
<div class="not_active">
  <div class="menu_block" id="keyboard_light_menu">
    <div class="top_text_menu">Подсветка клав.</div>
    <div class="text_info">Вкл/Выкл</div>
    <div class="menu_point" id="text_light_keyboard">Выкл</div>
    <div class="little_letter_text_info">
      <br />
      Выбор [* и #]
    </div>
  </div>
</div>
</div>
<!-- Turn off display after time -->
<div class="not_active">
  <div class="menu_block" id="timer_display">
    <div class="top_text_menu">Выключение экрана</div>
    <div class="little_letter_text_info"><br />Таймер</div>
    <div class="menu_point" id="text_timer">0</div>
  </div>
</div>
<!-- View mode -->
<div class="not_active">
  <div class="menu_block">
    <div class="top_text_menu">Обзор</div>
    <div class="menu_point_bold">Пароль</div>
    <div class="menu_point">####</div>
    <div class="menu_point_little">Введите пароль</div>
    <div class="menu_point" id="password_view"></div>
  </div>
</div>
<!-- Seting mode -->
<div class="not_active">
  <div class="menu_block">
    <div class="top_text_menu">Настройка</div>
    <div class="menu_point_bold">Пароль</div>
    <div class="menu_point">####</div>
    <div class="menu_point_little">Введите пароль</div>
    <div class="menu_point" id="password_setting"></div>
  </div>
</div>
<!-- Change password -->
<div class="not_active">
  <div class="menu_block" id="">
    <div class="top_text_menu">Смена пароля</div>
    <div class="menu_point_bold">Пароль</div>
    <div class="menu_point">####</div>
    <div class="menu_point_little">Введите новый пароль</div>
    <div class="menu_point" id="password_change"></div>
  </div>
</div>

```

```

</div>
</div>
<!-- Choose modulation -->
<div class="not_active">
  <div class="menu_block" id="">
    <div class="top_text_menu">Канал</div>
    <div class="text_info">Модуляция</div>
    <div class="menu_point" id="analogue_modulation"></div>
    <div class="little_letter_text_info">
      <br />
      Выбор [* и #]
    </div>
  </div>
</div>
</div>
<!-- Choose frequency in -->
<div class="not_active">
  <div class="menu_block" id="">
    <div class="top_text_menu">Канал</div>
    <div class="text_info">Частота прм</div>
    <div
      class="text_info"
      id="old_input_in_frequency_analogue"
    ></div>
    <div class="little_letter_text_info">
      Выбор [0-9]
    </div>
    <div class="menu_point" id="input_in_frequency_analogue">
      000000.0
    </div>
  </div>
</div>
</div>
<!-- Choose frequency out -->
<div class="not_active">
  <div class="menu_block" id="">
    <div class="top_text_menu">Канал</div>
    <div class="text_info">Частота прд</div>
    <div
      class="text_info"
      id="old_input_out_frequency_analogue"
    ></div>
    <div class="little_letter_text_info">
      Выбор [0-9]
    </div>
    <div class="menu_point" id="input_out_frequency_analogue">
      000000.0
    </div>
  </div>
</div>
</div>
<!-- Channel width -->
<div class="not_active">
  <div class="menu_block" id="">

```

```

<div class="top_text_menu">Канал</div>
<div class="text_info">Шир. канала</div>
<div class="menu_point" id="chanel_width_analogue">12,5</div>
<div class="little_letter_text_info">
  <br />
  Выбор [* и #]
</div>
</div>
</div>
</div>
<!-- Power -->
<div class="not_active">
  <div class="menu_block" id="">
    <div class="top_text_menu">Канал</div>
    <div class="text_info">Мощность</div>
    <div class="menu_point" id="power_analogue">Номинальная</div>
    <div class="little_letter_text_info">
      <br />
      Выбор [* и #]
    </div>
  </div>
</div>
</div>
<!-- Volume menu -->
<div class="not_active">
  <div class="menu_block" id="volume_menu">
    <div class="top_text_menu">Громкость</div>
    <div class="text_info">Уровень</div>
    <div class="text_info" id="graph_volume">
      [|||||]
    </div>
    <div class="text_info"><br /></div>
    <div class="text_info" id="digit_volume">24</div>
  </div>
</div>
</div>
<!-- Bottom button block -->
<div id="bottom_block" class="not_active">
  <div id="left_bottom_button"></div>
  <div id="center_bottom_button"></div>
  <div id="right_bottom_button"></div>
</div>
</div>
<div class="radiostation_button_block">
  <div class="radiostation_button" id="button_ok">✓</div>
  <div class="radiostation_button" id="button_menu">M</div>
  <div class="radiostation_button" id="button_cancel">×</div>
</div>
<div class="radiostation_button_block">
  <div class="radiostation_button" id="button_1">1</div>
  <div class="radiostation_button" id="button_2">2</div>
  <div class="radiostation_button" id="button_3">3</div>

```

```

</div>
<div class="radiostation_button_block">
  <div class="radiostation_button" id="button_4">4</div>
  <div class="radiostation_button" id="button_5">5</div>
  <div class="radiostation_button" id="button_6">6</div>
</div>
<div class="radiostation_button_block">
  <div class="radiostation_button" id="button_7">7</div>
  <div class="radiostation_button" id="button_8">8</div>
  <div class="radiostation_button" id="button_9">9</div>
</div>
<div class="radiostation_button_block">
  <div class="radiostation_button" id="button_lattice">#</div>
  <div class="radiostation_button" id="button_0">0</div>
  <div class="radiostation_button" id="button_star">*</div>
</div>
</div>
<div class="radiostation_accum_not_active"></div>
</div>

<script type="text/javascript" src="../js/modules/socket.io.js"></script>
<script src="../js/clientLogicBroadcasting.js"></script>
<script src="../js/clientControls.js"></script>
<script src="../js/menuRadiostation.js"></script>
<script src="../js/menuLogic.js"></script>
<script src="../js/menuLogic.js"></script>
</body>
</html>

```

Обозначение					Наименование					Дополнительные сведения				
					Текстовые документы									
БГУИР КП 1–40 01 01 605 ПЗ					Пояснительная записка					47 с.				
					Графические документы									
ГУИР 851006 01 ПД					Схема программы на А2					Формат А2				