

**Проектирование процедурного музыкального секвенсора
с точным таймингом в парадигме Zero-Dependency
методами промпт-инжиниринга**

Пайдершев Алексей Константинович

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В статье рассматривается процесс создания браузерного музыкального секвенсора с помощью большой языковой модели Qwen Studio. Исследование фокусируется на решении сложных инженерных задач, связанных с точным временным планированием, процедурным синтезом ударных и тональных звуков, а также динамическим изменением музыкальной структуры. Особое внимание уделено способности нейросети реализовывать оффлайн-рендеринг аудио и ручное конструирование бинарных WAV-файлов без использования внешних библиотек и аудио-сэмплов. Результаты показывают, что языковые модели успешно справляются с проектированием приложений, требующих жесткой синхронизации аудио- и визуальных потоков в рамках парадигмы Zero-Dependency.

Ключевые слова: Qwen Studio, нейросети, промпт-инжиниринг, Web Audio API, процедурный синтез, секвенсор, вайб-кодинг, Zero-Dependency.

Designing a procedural music sequencer with precise timing in the Zero-Dependency paradigm using prompt engineering methods

Paydershev Alexey Konstantinovich

Sholom-Aleichem Priamursky State University

Student

Abstract

The article examines the process of creating a browser-based music sequencer using the Qwen Studio large language model. The study focuses on solving complex engineering tasks related to precise temporal planning, procedural synthesis of drum and tonal sounds, and dynamic changes in musical structure. Special attention is paid to the neural network's ability to implement offline audio rendering and manual construction of binary WAV files without using external libraries and audio samples. The results show that language models successfully cope with designing applications that require strict synchronization of audio and visual streams within the Zero-Dependency paradigm.

Keywords: Qwen Studio, neural networks, prompt engineering, Web Audio API, procedural synthesis, sequencer, vibe coding, Zero-Dependency.

1 Введение

1.1 Актуальность

Разработка музыкальных веб-приложений традиционно сталкивается с фундаментальной проблемой: стандартные таймеры JavaScript обладают недопустимым дрейфом времени, что делает невозможным создание ритмичных композиций с помощью простых циклов. Для решения этой задачи профессиональные разработчики используют алгоритм «планирования с упреждением», который требует глубокого понимания асинхронных процессов. Кроме того, современные музыкальные инструменты должны быть гибкими: поддерживать различные музыкальные размеры, динамически изменять длину паттерна и позволять экспортировать результат в стандартные аудиоформаты. Актуальность данного исследования заключается в проверке способности больших языковых моделей самостоятельно проектировать подобные сложные, многопоточные системы в рамках жестких архитектурных ограничений, что позволяет оценить границы применимости генеративного подхода в задачах, требующих высокой точности и работы с бинарными данными.

1.2 Обзор исследования

Фундаментальные принципы конструирования интерактивных браузерных продуктов на базе HTML5 подробно раскрыты в исследованиях С. В. Рогатина и В. А. Антошкина, где авторы обосновывают универсальность нативных технологий [1]. Концепцию вайб-кодинга как педагогического и инженерного инструмента развивает А. Б. Рахмонов, выделяя его функциональные возможности для современного образовательного процесса [2]. Строгий инженерный контроль за результатами генерации предлагает И. В. Елманов, обосновывая необходимость методологии Spec Driven Development для верификации сгенерированного кода [3]. Применимость генеративных моделей в сложных предметных областях подтверждают работы Д. Р. Мухаметова, исследующего влияние ИИ на формирование политических образов [4], а также труд А. Д. Степанова и соавторов, демонстрирующий результативность нейросетевых алгоритмов в трансляционной кардиологии [5].

1.3 Цель исследования

Цель исследования – оценить эффективность применения структурированного промпт-инжиниринга для генерации веб-приложений, требующих точного аудиотайминга, процедурного синтеза звука и динамического управления состоянием пользовательского интерфейса в парадигме Zero-Dependency.

2 Материалы и методы

Методология исследования базируется на гибридном подходе, объединяющем структурированный промпт-инжиниринг и отказ от традиционных методов веб-разработки в пользу процедурной генерации. На

первом этапе формируются строгие архитектурные ограничения: вместо использования предзагруженных аудиофайлов и тяжелых библиотек утверждается применение исключительно встроенного Web Audio API.

Затем эти требования формируются в детализированную инструкцию для нейросети. Применяется метод ролевого позиционирования, а также жестко задаются математические и физические параметры звука: типы волн, настройки ADSR-огнивающей и алгоритмы планирования. В качестве платформы для написания кода применяется среда Qwen Studio. Ключевой особенностью методологии является итеративный подход: первоначальная генерация рассматривается как прототип, который последовательно дорабатывается через серию уточняющих промптов для добавления мелодических дорожек, музыкальных размеров и функции экспорта.

3 Результаты

Первый шаг исследования – написание специализированного промпта для генерации базового ударного секвенсора.

Промт: «Действуй как Senior Frontend разработчик. Твоя задача – написать полный, готовый к запуску код веб-приложения «Процедурный секвенсор». Приложение должно быть реализовано в одном файле и работать локально в браузере. Никаких сторонних библиотек и аудио-сэмплов – только нативные HTML, CSS и JavaScript с Web Audio API.

Все звуки должны синтезироваться процедурно через Web Audio API. Не загружай внешние аудиофайлы.

Синтезируй 5 звуков: Kick, Snare, Hi-Hat, Clap, Tom.

Логика секвенсора: 16 шагов, 5 дорожек, BPM (60-180), общая громкость.

Требования к интерфейсу: Тёмная тема. Слева панель управления (Запустить/Остановить, BPM, Громкость, Очистить/Рандом), справа сетка 5x16. Активные шаги подсвечиваются. Визуальный указатель должен двигаться по сетке во время воспроизведения.

Выведи полный код в одном блоке, чтобы я мог сохранить его как index.html и сразу запустить».

Результат генерации: <https://disk.yandex.ru/d/LgoiRAK36OpOgw>

В результате применения описанной методологии был успешно сгенерирован полностью рабочий прототип процедурного секвенсора (рис. 1). Нейросеть строго соблюдала ограничения, реализовав приложение в едином файле.

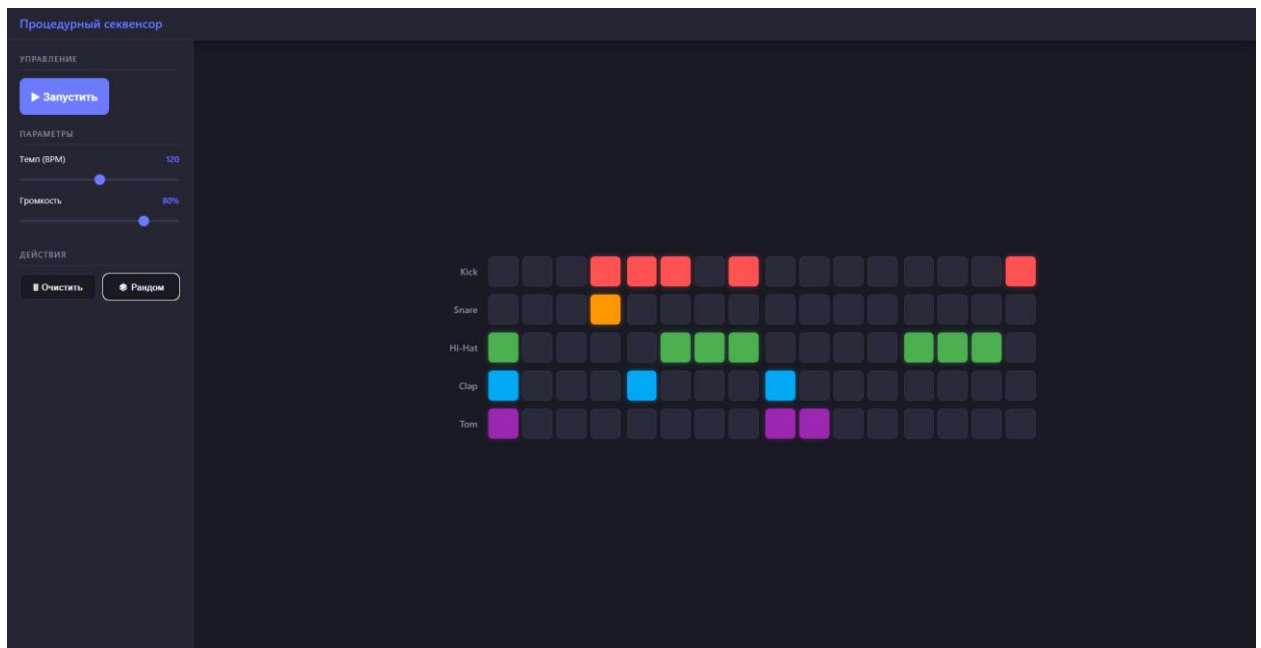


Рисунок 1. Интерфейс базового секвенсора

Ключевым достижением первичной генерации стала самостоятельная реализация временного планирования. ИИ разделил аудио-поток (планирование звуков на 100 мс вперед) и визуальный поток (обновление указателя воспроизведения), что обеспечило идеальный ритм без задержек. Кроме того, нейросеть корректно сгенерировала буфер белого шума и настроила математические огибающие для синтеза пяти различных ударных инструментов.

Также нейросеть верно выполнила поставленную задачу по созданию боковой панели (рис. 2). Все кнопки и ползунки работают. Нажатие на кнопку «рандом» случайным образом заполняет поля секвенсора (рис. 3).

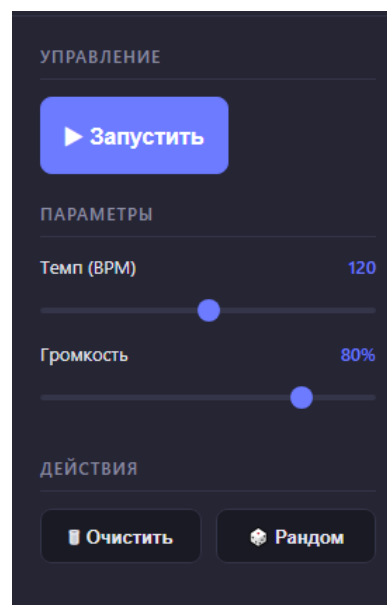


Рисунок 2. Боковая панель

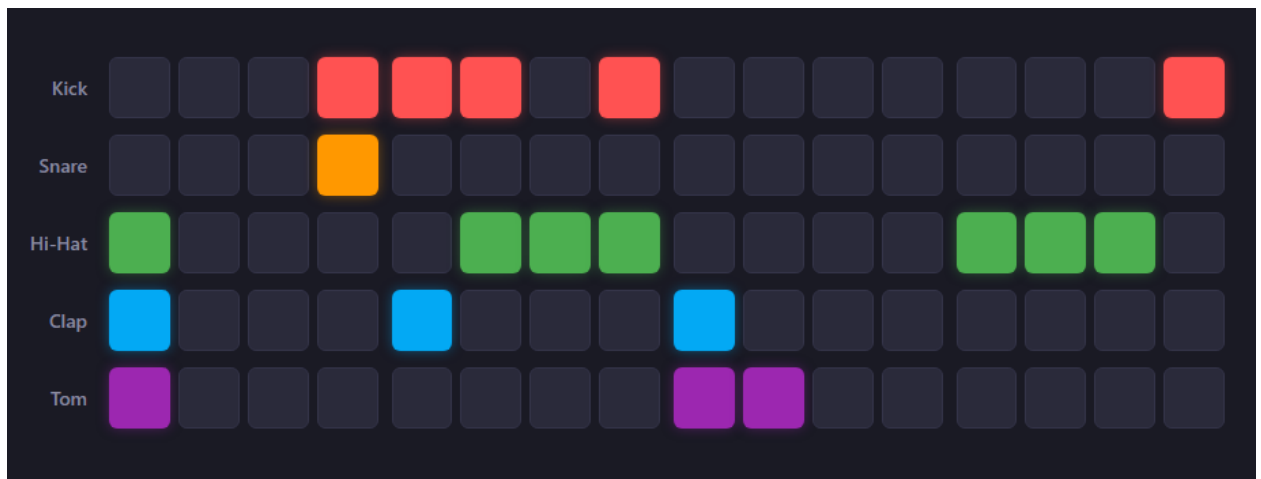


Рисунок 3. Случайное заполнение

Второй этап потребовал расширения функционала приложения, а именно добавления дорожки Synth с возможностью выбора нот.

Промт для доработки: «Добавь ещё одну звуковую дорожку Synth, которая воспроизводит звук синтезатора. К дорожке Synth добавь возможность выбирать ноты (при нажатии на шаг всплывает меню с выбором ноты из 12 нот второй октавы)».

Результат: <https://disk.yandex.ru/d/evlpTWXrksKKag>

Эта задача потребовала от ИИ изменить архитектуру данных. Если для ударных матрица шагов хранила простые значения (да/нет), то для Synth ячейки должны были хранить объекты с частотой и названием ноты. Нейросеть корректно адаптировала планировщик, добавив ветвление для тональных звуков, и реализовала процедурный синтез (рис. 4).

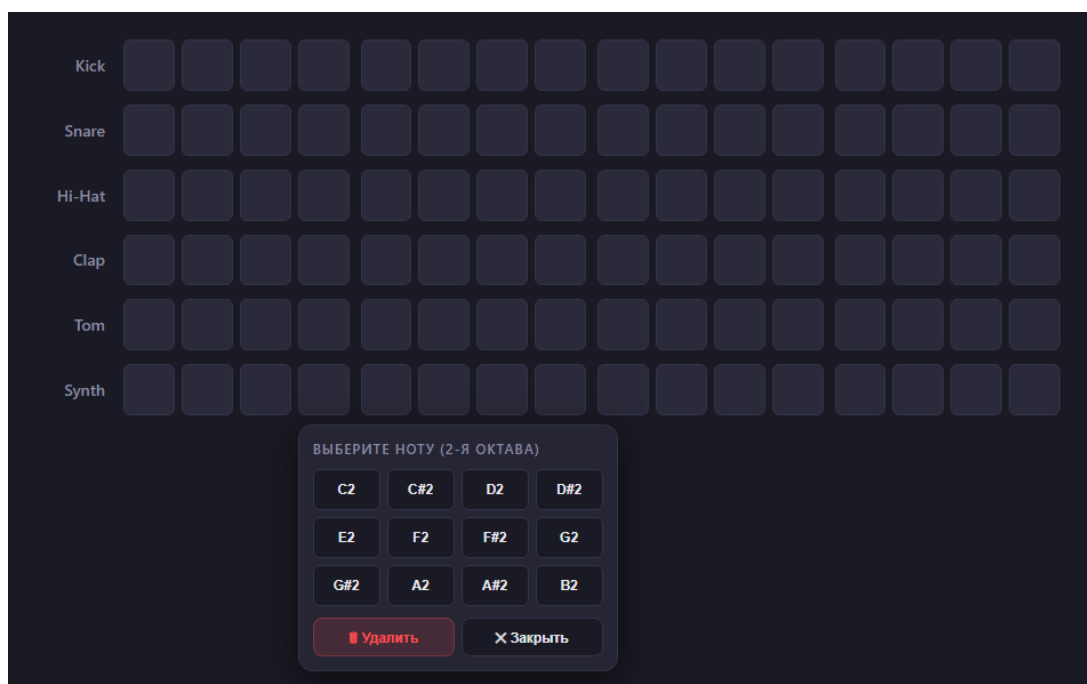


Рисунок 4. Всплывающее меню выбора ноты для дорожки Synth

Третий этап усложнил задачу: приложение должно было поддерживать нестандартные музыкальные размеры и переменное количество шагов, а также экспорт аудио.

Промт для доработки: «Добавь следующее:

- Возможность добавлять шаги (слайдер от 8 до 32).
- Выбор размера (3/4, 4/4, 7/8 и тд.) с визуальной группировкой тактов.
- Сохранение аудиофайла в формате WAV».

Результат: https://disk.yandex.ru/d/_ia5MC3Yr7GYVA

ИИ успешно реализовал динамическую перестройку DOM-дерева. При изменении слайдера шагов или выборе размера (например, 7/8) сетка перерисовывалась, а шаги визуально группировались в такты с помощью CSS-границ (рис. 5). При этом существующие активные ноты не терялись, что доказывает грамотное управление состоянием приложения.

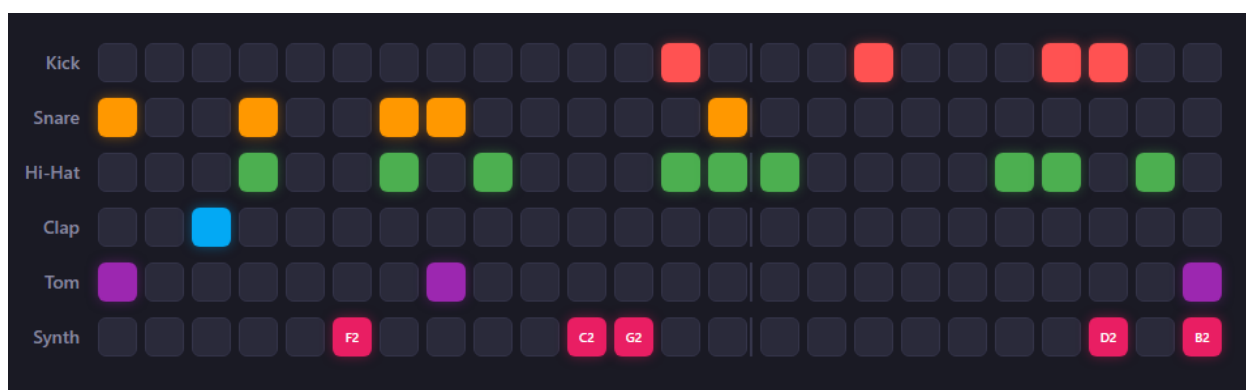


Рисунок 5. Визуальная группировка тактов для размера 7/8

Самой сложной задачей третьего этапа стала реализация экспорта. Поскольку использование сторонних библиотек запрещено парадигмой Zero-Dependency, ИИ пришлось написать конвертер бинарных данных вручную. Теперь файл можно скачивать и передавать (рис. 6-7).

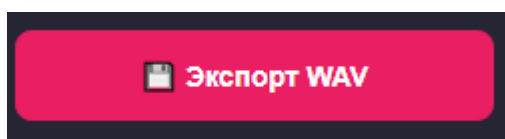


Рисунок 6. Кнопка экспорта и скачивания WAV-файла

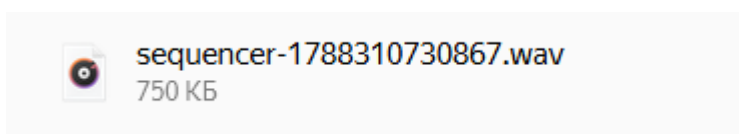


Рисунок 7. Загруженный файл

Тестирование показало, что сгенерированные файлы корректно открываются в стандартных аудиоплеерах, а размер файла точно соответствует длительности паттерна и частоте дискретизации.

4 Вывод

Проведенное исследование демонстрирует, что большие языковые модели способны успешно решать задачи, связанные с жестким временным планированием и процедурной генерацией мультимедийного контента. Эксперимент по созданию процедурного секвенсора выявил несколько важных закономерностей.

Во-первых, ИИ безупречно справился с алгоритмом временного планирования, который является стандартом индустрии для браузерного аудио. Это подтверждает способность модели понимать асинхронную природу JavaScript и разделять потоки рендеринга.

Во-вторых, добавление тональной дорожки и динамических музыкальных размеров показало, что нейросеть умеет гибко трансформировать структуры данных и перестраивать DOM-дерево без потери состояния, что является нетривиальной задачей для фронтенд-разработки.

В-третьих, ручная реализация экспорта в WAV доказывает, что в парадигме Zero-Dependency ИИ может выступать в роли полноценного системного инженера, способного работать с низкоуровневыми форматами данных без опоры на готовые пакеты.

Автоматизированная генерация кода с помощью ИИ фундаментально меняет процесс проектирования интерфейсов. Разработчик переходит от ручного написания строк кода к роли постановщика задач и верификатора архитектуры. Если задача опирается на строгие правила, математику и структурированные данные, искусственный интеллект позволяет в кратчайшие сроки создавать надежные, независимые от фреймворков и реактивные веб-приложения.

Библиографический список

1. Рогатин С. В., Антошкин В. А. Разработка компьютерных игр по технологии html5 //Информатика и прикладная математика. 2015. №. 21. С. 12-16. URL: <https://elibrary.ru/item.asp?id=25030492>
2. Рахмонов А. Б. Вайб-кодинг как инструмент цифровой педагогики: функциональные возможности и механизмы реализации //The Lingua Spectrum. 2025. Т. 12. №. 2. С. 552-556. URL: <https://lingvospektr.uz/index.php/lngsp/article/view/1329>
3. Елманов И. В. Инженерный подход к ии: использование spec driven development для валидации vibe coding //Вестник науки. 2025. Т. 4. №. 12 (93). С. 716-723. URL: <https://cyberleninka.ru/article/n/inzhenernyy-podhod-k-ii-ispolzovanie-spec-driven-development-dlya-validatsii-vibe-coding>
4. Мухаметов Д. Р. Политический вайб-маркетинг: возможности искусственного интеллекта для продвижения образа будущего //Известия Тульского государственного университета. Гуманитарные науки. 2025. №. 3. С. 27-33. URL: <https://cyberleninka.ru/article/n/politicheskiy-vayb-marketing-vozmozhnosti-iskusstvennogo-intellekta-dlya-prodvizheniya-obraza->

budushego

5. Степанов А. Д., Богданов Л. А., Шишкова Д. К., Кутихин А. Г. Применение инструментов на основе искусственного интеллекта в экспериментальной и трансляционной кардиологии: обзор //Российский кардиологический журнал. 2026. Т. 31. №. S2. С. 114-122. URL: <https://cyberleninka.ru/article/n/primeneniye-instrumentov-na-osnove-iskusstvennogo-intellekta-v-eksperimentalnoy-i-translyatsionnoy-kardiologii-obzor>