

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

Кафедра дискретной математики и информационных технологий

**РАЗРАБОТКА ПРОГРАММНО-АППАРАТНОГО КОМПЛЕКСА ДЛЯ
УПРАВЛЕНИЯ АДРЕСНЫМИ СВЕТОДИОДНЫМИ ЛЕНТАМИ
WS2812B НА БАЗЕ ОДНОПЛАТНОГО КОМПЬЮТЕРА RASPBERRY PI**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 421 группы

направления 09.03.01 — Информатика и вычислительная техника

факультета КНиИТ

Давиденко Алексея Алексеевича

Научный руководитель

ассистент

А. А. Трунов

Заведующий кафедрой

доцент, к. ф.-м. н.

Л. Б. Тяпаев

Саратов 2021

ВВЕДЕНИЕ

Невозможно себе представить современную жизнь без искусственных источников света. Эти источники продолжают играть важнейшую роль в технологическом прогрессе и применяются во всех сферах, например в сфере развлечений.

Сочетание различных световых сигналов может влиять на настроение человека, делая его более спокойным, активным, радостным или грустным. В паре со звуковыми сигналами, это может оказывать более сильное воздействие, что активно применяется в сфере развлечений, например, в проведении световых шоу, концертов, презентаций компаний и других областях.

Одним из основных видов искусственных источников света являются светодиоды, полупроводниковые приборы, создающие оптическое излучение при пропускании через него электрического тока в прямом направлении. Адресные же светодиоды, в свою очередь, являются подтипом светодиодов, ключевой особенностью которых является возможность управления каждым светодиодом, находящимся в ленте адресных светодиодов, по отдельности за счёт встроенного контроллера. Но также в работе с адресными светодиодами существует трудность, заключающаяся в необходимости более сложного управляющего контроллера, чем для обычных светодиодов.

Звуками, в свою очередь, также можно управлять. Получая на устройстве считывания некий звуковой поток, можно его проанализировать, получить набор его характеристик, и, преобразовав, трансформировать его в другой звуковой поток, либо, например, в световой поток.

Реализовать трансформацию звукового потока в световой можно на различных платформах: на персональных компьютерах, плат с микроконтроллерами, либо одноплатных компьютерах Raspberry Pi. Ввиду своих дороговизны, низкой энергоэффективности и избыточной производительности, не оптимально производить эту операцию на персональных компьютерах. Arduino являются подходящим вариантом для реализации такой трансформации, но, ввиду недостаточности производительности, также не являются оптимальным решением, если необходимо в рамках одной платы реализовать несколько проектов. Наиболее оптимальным является использование одноплатных компьютеров Raspberry Pi, которые объединяют в себе удобство в использовании персональных компьютеров, использование одного из дистрибутивов Linux в качестве операционной

системы, а так же энергоэффективность и компактность плат с микроконтроллерами.

Актуальность работы заключается в том, что в различных сферах требуется использование света и музыки для достижения поставленной цели. В разработках, связанных с концепцией «умный дом», например, световые сигналы дополняют играющую музыку, включенную пользователем.

Целью выпускной квалификационной работы является разработка и тестирование программно-аппаратного комплекса, позволяющего управлять адресными светодиодными лентами с помощью звука, а так же веб-приложения, позволяющего управлять комплексом.

Для достижения цели были поставлены следующие задачи:

- Обзор и анализ характеристик и особенностей работы светодиодов, подходящих для реализации поставленной цели;
- изучение способов работы со звуковыми потоками;
- обзор и анализ характеристик платформ, подходящих для реализации поставленной цели;
- обзор средств для разработки пользовательских интерфейсов;
- обзор и анализ протоколов связи, подходящих для реализации поставленной цели;
- изучение особенностей работы с одноплатными компьютерами Raspberry Pi;
- разработка комплекса, позволяющего управлять адресными светодиодными лентами с помощью звука;
- разработка веб-приложения для управления программно-аппаратным комплексом;
- тестирование разработанного программно-аппаратного комплекса для управления светодиодными лентами при помощи веб приложения.

Бакалаврская работа состоит из списка обозначений и сокращений, введения, двух разделов, заключения, списка использованных источников и четырёх приложений. Общий объём работы – 70 страниц, из них 44 страницы – основное содержание, включая 36 рисунков и 4 таблицы, список использованных источников информации – 23 наименований.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Первый раздел «Обзор используемых технологий» посвящён обзору характеристик светодиода WS2812b, особенностей записи и обработки звука в системах Linux, обзор и анализ средств и способов разработки пользовательских интерфейсов, а также протоколов, используемых для обеспечения клиент-серверного взаимодействия.

Светодиод WS2812b является одним из широко используемых типов адресных светодиодов. Он представляет из себя светодиод, состоящий из контроллера и RGB чипа, интегрированных в светодиод SMD 5050, обработчик цифровых сигналов и контроллер восстановления и усиления сигнала [1]. Для передачи информации в ленте светодиодов используется линейное кодирование NRZ. При этом цвет каждого из светодиодов кодируется 24 битами информации, где первые 8 бит соответствуют зелёной составляющей цвета, следующие 8 бит – красной, а последние – синей.

Для обработки звука в Linux-системах применяются два вида звуковых подсистем: ALSA и OSS. ALSA предоставляет функциональность аудио драйвера и цифрового интерфейса музыкальных инструментов в Linux, а также тесно связана с ядром Linux [2]. Для записи и воспроизведения звука в Linux-системах, использующих ALSA в качестве звуковой подсистемы, используют утилиты arecord и aplay. Arecord и aplay являются утилитами для командной строки и предназначены для записи и воспроизведения звука соответственно. Эти утилиты поддерживают несколько форматов входных и файлов, несколько форматов данных, разное количество каналов, частоту дискретизации и другие параметры [3, 4].

Для разработки пользовательских интерфейсов существует множество библиотек и фреймворков: React, Angular, Vue и другие. Наиболее популярным из них является библиотека React [5].

React является JavaScript-библиотекой для создания пользовательских интерфейсов. С помощью React можно разрабатывать одностраничные приложения используя JSX для шаблонов. Также в качестве разработки можно выбрать язык TypeScript. Основной концепцией React является разбиение элементов приложения на компоненты [6]. Компоненты React представляет из себя участки кода, соответствующие части веб-страницы приложения. По умолчанию точкой входа React приложения является контейнер div с идентификатором Root. В

этом контейнере содержится всё веб приложение, представляющее из себя композицию компонентов или JSX элементов, состоящих из одного или нескольких компонентов или JSX-элементов. Данные в компоненте хранятся в виде «пропсов». Эти «пропсы» можно использовать для передачи динамически изменяемых параметров от родительских компонентов дочерним [6].

В библиотеке React нет встроенной поддержки маршрутизации страниц. Для предоставления веб-приложению этой возможности необходимо использовать сторонние библиотеки. Наиболее популярной из таких библиотек является ReactRouter. Эта библиотека предоставляет компоненты, позволяющие добавить в веб приложение компоненты трёх видов: маршрутизаторы, маршруты и элементы навигации [7]. Маршрутизация веб-приложения позволяет пользователю видеть, в каком месте приложения он находится в данный момент времени, и облегчить процесс использования приложения.

При сложной логике приложения становится сложно управлять параметрами приложения, используемыми в нескольких компонентах. Для решения этой проблемы в React-приложениях используются два подхода: использование Redux (или другого менеджера состояний), или использование встроенной в React технологии React Context. Использование Redux необходимо, когда невозможно эффективно управлять состоянием, используя лишь возможности React. Так как для небольших приложений можно передавать данные во все необходимые компоненты без потерь в эффективности и увеличения сложности логики приложения, применение Redux необходимо лишь для более сложных приложений. В простых приложениях более предпочтительно использование React Context. Эта технология позволяет использовать динамически изменяемые параметры приложения в нескольких компонентах, в том числе и на разных уровнях вложенности [6]. При этом необходимо создание контекста с используемыми параметрами, использование компонента, предоставляющего использование параметров контекста всем дочерним компонентам, а также компонента, создающего подписку на контекст, для каждого компонента, в котором необходимо использование контекста.

Для стилизации пользовательских интерфейсов применяется каскадные таблицы стилей, или CSS. С помощью CSS можно изменить внешний вид элементов и их отображение на странице. Для выстраивания элементов на странице наиболее часто используются модели раскладки flex и grid. Основное различие

этих моделей раскладки заключается в том, что в модели flex элементы позиционируются в одном направлении, то есть, либо в строке, либо в столбце, в то время как grid позволяет позиционировать в двух направлениях: элементы можно позиционировать и в строке, и в столбце [8]. Одной и той же раскладки элементов на странице можно добиться используя как flex модель, так и grid. Различие будет состоять только в количестве дополнительных стилей и использованных контейнерах.

Для облегчения разработки стилей компонентов веб-приложений можно использовать CSS-модули. CSS модули используют подход при котором HTML разметка задаётся в JavaScript файле и стилизация применяется только к компоненту, заданному в этом JS файле. Такой подход гарантирует, что все стили одного компонента: находятся в одном месте и применяются исключительно к этому компоненту и никакому другому.

Для обеспечения взаимодействия типа «клиент-сервер» необходимо использовать один из протоколов передачи данных. Основным протоколом передачи данных во Всемирной паутине является HTTP [9]. При клиент-серверном взаимодействии по протоколу HTTP, только клиент может быть инициатором запроса, а сервер – инициатором ответа. Протокол WebSocket предоставляет возможность как клиенту, так и серверу отправлять сообщения, не основываясь на истории предыдущих запросов. При этом WebSocket обладает следующими особенностями: протокол двунаправленный (и клиент, и сервер могут независимо обмениваться сообщениями), коммуникация полнодуплексная (обмен сообщениями происходит независимо и одновременно), а также используется лишь одно TCP соединение [9]. При этом, для установки связи по протоколу WebSocket, необходимо установить сначала HTTP соединение, которое затем будет закрыто и использоваться только один TCP канал.

Во втором разделе «Разработка программно-аппаратного комплекса» описывается процесс разработки серверной и клиентской составляющих программно-аппаратного комплекса, а также тестирования разработанного комплекса.

Разработка серверной составляющей комплекса включает в себя разработку программ, предоставляющих:

- Возможность управления светодиодной лентой.
- Возможность записи и обработки звука.

— Программного интерфейса (API) для управления комплексом с помощью веб-приложения.

Для реализации поставленной задачи была выбрана платформа Raspberry Pi. Ввиду своей распространённости и доступности, была выбрана модель Raspberry Pi 3 model B. Операционная система RPI располагается на внешней microSD-карте, на которую можно записать один из дистрибутивов Linux. Официально можно установить одну из следующих операционных систем [10]: Raspberry Pi OS, Ubuntu, Manjaro, RISC OS. Для разработки был выбран дистрибутив Manjaro ввиду наличия у аналогов критических недостатков.

Для управления лентой светодиодов WS2812b необходимо использование широтно-импульсной модуляции. Для этого можно использовать использовать расположенные на плате Raspberry Pi GPIO контакты. При этом для программной генерации ШИМ можно использовать все контакты GPIO, а для аппаратной – только контакты GPIO12, GPIO13, GPIO18, GPIO19. Недостатком программной генерации ШИМ является большие затраты аппаратных ресурсов на генерацию и кодирование импульсов. Для генерации ШИМ был выбран контакт GPIO18, а для заземления ленты был выбран контакт 6 GPIO.

Был составлен макет программно-аппаратного комплекса, и, согласно макету, была собрана и протестирована работоспособность системы.

В качестве основного языка разработки серверной составляющей комплекса был выбран язык JavaScript. Для взаимодействия с операционной системой используется среда выполнения JS – NodeJS (Node).

Для управления адресными светодиодными лентами с помощью языка программирования Node, наибольшей популярностью обладают 3 пакета: node-pixel, rpi-ws281x и rpi-ws281x-native. При разработке был использован пакет rpi-ws281x-native ввиду более подробной документации и использования при выполнении нативных привязок GPIO. При работе с этим пакетом необходимо указание количества светодиодов в ленте, и порт GPIO, к которому подключена лента. Данные о ленте хранятся в виде массива чисел, где каждому элементу массива соответствует определённый светодиод. Обращение к светодиодам происходит по индексам. Для отображения необходимого состояния ленты используется метод render(), для очистки ленты – reset(), а для завершения работы с лентой – finalize().

Был разработан модуль, позволяющий управлять светодиодной лентой.

При этом был реализован эффект, при котором новые цвета на ленте исходят из центра ленты в стороны.

Для записи и воспроизведения звука используются утилиты `arecord` и `aplay`. Для запуска этих утилит из основного процесса сервера комплекса, необходимо создание дочерних процессов, управляемых из основного процесса. Для этого в Node используется модуль «`child_process`». При этом существует несколько методов для запуска дочерних процессов: `exec`, `execFile`, `fork` и `spawn`. Для запуска процессов, выполняющих команды `aplay` и `arecord`, был использован метод `spawn`.

Для преобразования полученного звукового потока в световой, был разработан и реализован модуль обработки звука.

Для предоставления программного интерфейса управления комплексом был выбран протокол `WebSocket`. Для создания `WebSocket`-сервера, необходимо создать HTTP сервер, а затем установить `WebSocket` соединение. Для создания HTTP-сервера используется стандартная библиотека Node `http`. Затем, для создания самого `WebSocket`-сервера, необходимо использовать библиотеку, предоставляющую возможность установления `WebSocket` соединения. В качестве такой библиотеки была выбрана библиотека «`websocket`».

Был разработан модуль, предоставляющий API, с помощью которого можно удалённо управлять состоянием программно-аппаратного комплекса.

При разработке клиентской составляющей комплекса использовалась библиотека `React`. Для создания `React`-приложения использовалась утилита «`create-react-app`» с настроенной поддержкой `TypeScript`. Затем была улучшена структуризация приложения для оптимизации разработки и сборки приложения и реализовано веб-приложение с использованием технологий `ReactRouter`, `React Context` и `CSS`-модулей.

В результате разработки веб-приложения был реализован сайт, с помощью которого можно взаимодействовать с комплексом управления светодиодными лентами.

Для тестирования разработанного программно-аппаратного комплекса использовалось разработанное веб-приложение.

Была проверена достоверность предоставляемой API информации о состоянии комплекса, а также возможность удалённого изменения состояния комплекса с помощью веб-приложения. Во всех тестах изменение поведения ком-

плекса было ожидаемым и корректным, что показывает эффективность и работоспособность разработанного программно-аппаратного комплекса.

ЗАКЛЮЧЕНИЕ

Проекты, связанные с концепцией «умный дом», являются одними из наиболее молодых и популярных направлений разработки. Одну из важнейших ролей в таких проектах является создание и использование искусственных источников света.

В ходе выполнения выпускной квалификационной работы был разработан программно-аппаратный комплекс для управления светодиодными лентами с помощью звука, а также веб-приложение для управления комплексом. Впоследствии разработанный программно-аппаратный комплекс был протестирован при помощи разработанного веб-приложения. В ходе тестов комплекс показал свою эффективность и корректность.

Преимуществом данной разработки является возможность управления всем программно-аппаратным комплексом используя веб-приложение. Есть возможность управления состояниями светодиодной ленты и микрофона, а также их параметрами: яркостью свечения светодиодов, скоростью эффекта бегущих из центра огней, нижней границей звучания и максимальным процентом зашумлённости.

Также был произведён анализ предметной области, получены знания о способах записи, обработки и воспроизведения звуковых потоков, способы модуляции сигналов, изучены особенности работы с одноплатными компьютерами Raspberry Pi, изучены способы клиент-серверного взаимодействия, а также средства для разработки пользовательских интерфейсов.

Таким образом, цель и задачи выпускной квалификационной работы были выполнены.

Основные источники информации:

- 1 Worldsemi [Электронный ресурс]. — URL: <http://www.world-semi.com/Certifications/WS2812B.html> (Дата обращения 02.05.21). Загл. с экрана Яз. Англ.
- 2 ALSA project [Электронный ресурс]. — URL: https://www.alsa-project.org/wiki/Main_Page (Дата обращения 02.05.21). Загл. с экрана Яз. Англ.

- 3 die.net [Электронный ресурс]. — // <https://www.die.net/> – arecord
URL: <https://linux.die.net/man/1/arecord> (Дата обращения 03.05.21).
Загл. с экрана Яз. Англ.
- 4 die.net [Электронный ресурс]. — // <https://www.die.net/> – aplay
URL: <https://linux.die.net/man/1/aplay> (Дата обращения 03.05.21).
Загл. с экрана Яз. Англ.
- 5 State of JS [Электронный ресурс]. — URL: <https://stateofjs.com/> (Дата
обращения 20.02.21). Загл. с экрана Яз. англ.
- 6 React [Электронный ресурс]. — URL: <https://reactjs.org/> (Дата
обращения 21.02.21). Загл. с экрана Яз. англ.
- 7 REACT ROUTER [Электронный ресурс]. — URL: <https://reactrouter.com/> (Дата обращения 24.02.21). Загл. с экрана Яз. англ.
- 8 MDN Web Docs [Электронный ресурс]. — URL: <https://developer.mozilla.org/ru/docs/> (Дата обращения 02.03.21). Загл. с экрана Яз. рус.
- 9 Medium [Электронный ресурс]. — // <https://medium.com/> – HTTP and
Websockets: Understanding the capabilities of today’s web communication
technologies URL: <https://github.com/css-modules/css-modules> (Дата
обращения 07.03.21). Загл. с экрана Яз. англ.
- 10 Raspberry Pi [Электронный ресурс]. — URL: <https://www.raspberrypi.org/help/> (Дата обращения 04.05.21). Загл. с экрана Яз. Англ.