

Муниципальное бюджетное общеобразовательное учреждение
Злынковская СОШ № 1
Брянская область

ПРОЕКТНАЯ РАБОТА
ДЕКОРАТИВНЫЙ СВЕТИЛЬНИК
«ЭЛЕКТРОННЫЙ ОГОНЬ»

Автор:

Лутченков Кирилл Иванович,
ученик 10 класса МБОУ Злынковской СОШ № 1

Руководитель:

Степура Андрей Александрович,
учитель физики и информатики,
педагог ДО Центра цифрового
и гуманитарного образования «Точка роста»
на базе МБОУ Злынковской СОШ № 1

2023 г.

Оглавление

	стр
I. Паспорт проектной работы.....	3
1. Общие сведения.....	3
2. Введение.....	3
3. Целевая аудитория, заказчик.....	6
4. Этапы и методы работы над проектом.....	6
5. Необходимые ресурсы.....	7
6. Смета.....	7
II. Описание процесса исполнения	8
III. Выводы	15
IV. Список использованных источников информации.....	17
V. Приложения.....	18

I. Паспорт проектной работы

1. Общие сведения	
Тема проектной работы	Декоративный светильник «Электронный огонь»
Направление	Техническое
Автор	Лутченков Кирилл Иванович, ученик 10 класса МБОУ Злынковской СОШ №№ 1
Руководитель	Степура Андрей Александрович, учитель физики и информатики, педагог ДО
Место выполнения проекта (наименование ОО)	Центр цифрового и гуманитарного образования «Точка роста» на базе МБОУ Злынковской СОШ № 1
Тип проекта	По виду деятельности - технический. По количеству участников – индивидуальный. По продолжительности – среднесрочный. По предметно-содержательной области – межпредметный (информатика /технология/ физика).
Срок выполнения проекта	2022 – 2023 учебный год (сентябрь – февраль)
2. Введение	
Актуальность	Иногда нам хочется украсить комнату чем-то новым, технологичным, эффектным,- тем, что завораживает внимание. Для этого, например, можно приобрести умную колонку «Алиса», а к ней добавить светодиодную лампочку из экосистемы умного дома. Но это потребует значительной суммы денег, а функционал такой лампочки может не совсем соответствовать нашим желаниям. Поэтому лучше изготовить необычный светильник своими руками. Он

	<p>будет работать в заданном режиме и радовать своего владельца разнообразием функциональных возможностей! Затратив намного меньше денег, в итоге можно получить, то, что актуально всегда:</p> <ul style="list-style-type: none"> - новые знания и полезные практические навыки в программировании микросхемных плат, сборке электрических цепей, выполнении паяльных работ, в использовании мультиметра и других приборов и инструментов; - удовлетворение от процесса и результата своего труда.
Цель работы	Изготовление управляемого декоративного светодиодного светильника с 18 эффектами свечения
Задачи	<ol style="list-style-type: none"> 1. Изучить особенности работы управляемых светодиодных ламп по специальным тематическим источникам информации. 2. Подобрать необходимые для сборки светильника электронные компоненты, расходные материалы, плафон. 3. Изготовить корпусную основу светильника с подставкой. 4. Изучить информацию по созданию программного обеспечения для управления светильником. 5. Выполнить вёрстку кода, провести его промежуточное тестирование. 6. Выполнить полную сборку изделия. 7. Проверить качество сборки, выявить и устранить недочёты конструкции.

	<p>8. Протестировать функционирование кода в процессе работы светильника.</p> <p>9. Презентовать изделие.</p>
Объект проектирования	Управляемый декоративный светодиодный светильник с 18 эффектами свечения
Решаемая проблема	Уменьшение себестоимости декоративного украшения дома
Риски проекта и пути их минимизации	<p>Риск 1: недостаток знаний и умений для программирования управления светильником.</p> <p>Пути минимизации риска: самообразование, прохождение онлайн-курсов по программированию, консультации с руководителем проекта.</p> <p>Риск 2: получение травмы при нарушении техники безопасности в процессе выполнения проекта.</p> <p>Пути минимизации риска: строгое соблюдение требований техники безопасности.</p>
Положительные эффекты	<p>Всё зависит от того, как владелец захочет использовать светильник. Во-первых, будут обеспечены положительные эмоции. Например: хочется утром «красиво» проснуться, - устанавливается режим «Рассвет». В доме нет камина, - выручит режим «Огонь»! В любом случае, светильник сможет украсить любую комнату и будет создавать хорошее настроение.</p> <p>Во-вторых, к положительным эффектам можно отнести приобретение новых знаний и умений в процессе подготовки и реализации проектной идеи.</p> <p>В-третьих, достигается экономия средств из семейного бюджета.</p>

Требования к изделию	<ul style="list-style-type: none"> - Функциональность. Светильник должен работать во всех планируемых режимах. - Эстетичный внешний вид. Изделие должно украсить интерьер комнаты. - Качество исполнения. Изделие должно быть надёжным, не иметь внешних и внутренних дефектов. - Удобство в использовании. - Экологичность изделия. - Малое энергопотребление. - Безопасность использования.
3. Целевая аудитория, заказчики	
Целевая аудитория, заказчики	<p>Заказчик конкретного изделия – семья.</p> <p>К целевой аудитории относится любой человек, желающий иметь аналогичный светильник для украшения интерьеров дома.</p>
4. Этапы и методы работы над проектом	
Этапы работы над проектом	<p><i>Подготовительный этап:</i> выбор темы проекта, определение решаемой проблемы, первичный сбор информации, детализация темы и проблемы.</p> <p><i>Аналитический этап:</i> постановка цели и задач проекта, разработка паспорта проекта, планирование работы, подбор и изучение источников специализированной информации, оценка и подбор необходимых ресурсов. Оценка проектной идеи на реализуемость. Корректировка.</p> <p><i>Практический этап:</i> реализация проекта, оформление продукта, составление отчётной документации по проекту.</p> <p><i>Презентационный этап:</i> защита проекта.</p>

	<i>Оценочный этап:</i> подведение итогов работы, самооценка и оценка результатов проекта.
Методы работы	Теоретические и практические
5. Необходимые ресурсы	
Материальные ресурсы	<p>Компьютер с Windows, электричество.</p> <p>Wemos mini (плата управления), адресная матрица, гнездо 5.5×2.1, сенсорная кнопка, БП 5V (минимум 3А на каждые 100 LED), реле, сантех: (труба D50 длина 250, труба D40 короткая, заглушка D40).</p> <p>Плафон для лампы, клеевой пистолет и клеевые стержни к нему, паяльник, припой, мультиметр, монтажные проводки.</p>
Нематериальные ресурсы	<p>Программное обеспечение для ОС Windows (стандартные приложения, Arduino IDE, MIT app inventor - для создания приложения и встроенная в него среда разработки). Язык программирования «С++» и дополнительные библиотеки для Arduino (FastLED-3.2.9, GyverButton, NTPClient-master, WiFi, WiFiManager).</p>
6. Смета	
Смета	<ul style="list-style-type: none"> • Диодная матрица WS2812В 16x16 – 850 руб • Блок питания 5В/3А – 374 руб • Модуль управления <i>NodeMcU v2/V3</i> – 153 руб или (***)возможная замена) <i>Wemos D1 mini</i> - 124 руб • Пластмассовая ваза – 490 руб • Пластиковая труба диаметр 50мм – длина 250мм, заглушка диаметр 150мм – 350 рублей • Электричество – 70 копеек

	<ul style="list-style-type: none"> • Новые лезвия для ножа – 96 руб • Провода – 15 руб <p><i>Всего: 2 328,7 рублей</i></p> <p>Используемое в данном проекте программное обеспечение распространяется свободно и бесплатно. Компьютер, клеевой пистолет и клеевые стержни к нему, паяльник, мультиметр частично предоставлены школой, частично – из личной мастерской.</p>
--	--

II. Описание процесса исполнения

Предпроектная идея

Лампа под названием «Электронный огонь»! На первый взгляд кажется, что это что-то странное и необычное. Как огонь может быть электронным? Ответ – легко! По большому счёту всё очень просто: в основе подобного изделия - набор электронных плат Arduino, светодиодная матрица и несколько проводов. (Alex Gyver, 2019).

В умном доме, в обычном простом доме или квартире, - везде и всегда найдется место красивому и стильному светильнику. Такой светильник может использоваться как дополнительный источник света или как ночник.

Конечно же на рынке есть разные, в т.ч. дорогие варианты таких изделий, их цена начинается от 3-х тысяч рублей. Все модели имеют свои преимущества, но и недостатков у таких гаджетов тоже хватает. Обычно при хорошей «начинке» страдает программная часть. Разработчики не раскрывают весь потенциал платформ, на которых сделано их детище, не стремятся значительно разнообразить красивые эффекты или полезные функции, которых иногда очень не хватает. Они делают это постепенно, стараясь получить как можно больше прибыли от продажи.

Вместе с тем добиться расширения возможностей, демонстрируемых светильником по нашему усмотрению, не так уж сложно. Например, можно

добавить их в собственное приложение, разработанное в специальном конструкторе.

Да и в целом собственными руками создать «с нуля» необычный светильник с программным управлением, - интересная и вполне реализуемая идея, которая и легла в основу данного проекта.

Информационная и практическая подготовка

Светодиодные светильники давно стали достаточно популярными, поэтому неудивительно, что появилось немало желающих попробовать свои силы не только в самостоятельном изготовлении подобных изделий, но и в изменении их свойств согласно своим запросам и пожеланиям. Однако информации о плюсах и минусах, достигнутых в итоге таких работ, на просторах интернета не так уж и много. Хотелось бы учесть уже накопленный опыт, но многие авторы не фиксируют свои результаты, не хотят их документировать и не делятся ими с другими людьми. Поэтому, когда возникла идея самостоятельного создания управляемого декоративного светодиодного светильника с 18 эффектами свечения, сначала пришлось серьёзно задуматься, - с чего начать.

Для начала было решено подробнее ознакомиться с пайкой, т.к. это то действие, которое будет необходимо выполнять в течение почти всей работы над проектом от самого начала и до конца. Разобраться во всех нюансах помогла книга Г.И. Гуревича «Как паять». Из неё была получена информация об основных материалах и инструментах, применяемых для пайки при проведении электромонтажных и электроремонтных работ, о технологии пайки разных материалов мягкими и твёрдыми припоями, о правилах организации рабочего места при выполнении паяльных работ. Большим плюсом пайки является то, что соединение с её помощью деталей из однородных и разнородных материалов позволяет достаточно быстро

получить высокопрочное соединение, простое по способу и форме, без коробления и деформации соединяемых элементов (Гуревич, 1964).

Для реализации проекта необходимо знание основ программирования. Теоретическая и практическая подготовка по программированию была получена путём самообразования на основе изучения материалов специализированной онлайн-платформы Stepic и YouTube-канала «Заметки ардуинщика».

По книге У. Соммера «Программирование микроконтроллерных плат Arduino/Freduino» были изучены структура и функционирование микроконтроллеров, среда программирования Arduino (структура программы, команды, операторы и функции, способы ввода/вывода данных), подобраны инструменты и комплектующие, необходимые для реализации проектной идеи (Соммер, 2012).

Советы и практические примеры о том, как проектировать, отлаживать и изготавливать электронные устройства в домашних условиях, как работать с микросхемами, были рассмотрены по книге «Занимательная электроника» (Ревич, 2017) и пособию «Arduino «На пальцах»». Последний источник заинтересовал тем, что автор предлагает «...не знающему ничего о микроконтроллерах, научиться программировать платформу Arduino с нуля» (Менщиков, 2017).

Ну и, конечно, очень полезной находкой стала обучающая видеозапись по созданию светодиодного светильника на YouTube-канале Alex Gyver (Alex Gyver, 2019).

Электронные компоненты, расходные материалы, инструменты

Все необходимые компоненты можно разделить на электронную начинку и материалы для корпуса.

Требуемая электроника была заказана на AliExpress и Wildberries:

- диодная матрица WS2812B 16x16 – 850 рублей
- блок питания 5В/3А – 374 рубля
- модуль управления *NodeMCU v2/V3* – 153 рубля
- сенсорная кнопка – 160 рублей

Материал для изготовления корпуса светильника был куплен в строительном магазине, в качестве плафона для светильника приобретена пластмассовая ваза понравившейся формы и подходящего размера.

- Пластмассовая ваза – 490 рублей
- Пластиковая труба (диаметр 50 мм, длина 250 мм), заглушка (диаметр 150 мм) - 350 рублей.
- Паяльник, паяльная кислота и припой
- Канцелярский нож
- Сверло типа «Хвост ласточки»
- Шуруповёрт, бита PH2
- Клеевой пистолет, клеевые палочки-расходники
- Монтажные провода
- Провода сечением 1,5 мм
- Саморезы (2 шт)
- Циркуль
- Линейка

- Лучковая пила по металлу

Изготовление светильника

Для изготовления корпуса будущего светильника в магазине «Мастерок» были приобретены: кусок канализационной трубы диаметром 50 мм, заглушка диаметром 150 мм (*Рисунок 1*), пластмассовая ваза белого цвета со средней способностью пропускать свет, два самореза длиной по 40 мм.

С помощью линейки и канцелярского ножа был отмерен кусок трубы длиной 240 мм в соответствии с высотой вазы (будущего плафона), далее необходимая часть отрезана лучковой пилой по металлу.

Из оставшейся части трубы выпилена полоска длиной 50 мм и шириной 15 мм, которая позже припаяна разогретым паяльником к верхней части заготовки корпуса светильника. Данный элемент должен обеспечить возможность крепления сенсорного датчика, необходимого для управления светильником в оффлайн-режиме.

После примерки размещения диодной матрицы на трубе выполнена разметка места под вывод проводов. Отверстие необходимого размера пришлось вырезать паяльником, т.к. это был наиболее удобный вариант.

Затем на заглушке циркулем был начерчен контур отверстия, соответствующего диаметру трубы, само отверстие также вырезано с помощью паяльника. В полученное отверстие вставлена ранее изготовленная заготовка корпуса светильника, и далее детали запаяны в монолитную конструкцию (*Рисунок 2*).

После данных манипуляций необходимо было приступить к пайке, используя заранее выбранную схему (*Рисунок 3*)

Работа с паяльником требует правильной подготовки рабочего места и чёткого выполнения существующих правил техники безопасности, что и соблюдалось в ходе всей работы над проектом. Так как навыки пайки

имелись, в процессе работы трудностей не возникло, все операции удалось произвести достаточно легко, правильно и безопасно. Необходимые элементы были спаяны воедино согласно выбранной схеме (*Рисунок 4*).

После этого было выполнено тестирование матрицы. Так как в редких случаях на них бывает заводской брак, нужно было убедиться, что матрица находится в рабочем состоянии. На платах используемого типа при подключении по нужной схеме загораются контрольные светодиоды, что свидетельствует том, что они работают, как и сама плата управления. Тестирование подтвердило, что проблем нет, всё функционирует (*Рисунок 5*).

Далее подготовленной матрицей была обёрнута заготовка корпуса, сама матрица закреплена на ней пластиковыми стяжками, чтобы при необходимости можно было быстро модернизировать или заменить отдельные элементы системы в случае выхода их из строя. Через ранее подготовленное для этого отверстие все провода выведены внутрь и вверх, также вверх выведена и сенсорная кнопка (*Рисунок 6*).

После этого произведено крепление сенсорной кнопки на верхней части изделия с помощью клеевого пистолета, предварительно зафиксировав кнопку для удобства в нужном положении на двухсторонний скотч.

Программирование работы светильника

Чтобы запрограммировать процесс работы светильника, были применены ранее полученные базовые знания языка программирования «Arduino C». Это высокоуровневый язык программирования, который был смешан с языком «C+». Код получился достаточно несложным и читаемым, для большего понимания добавлялись комментарии.

Для уменьшения времени разработки кода было решено сначала установить несколько готовых библиотек: «FastLed» (версии 3.2.9),

«GyverButton», «NTPClient-master» , «WiFi», «WiFiManager». Этим программная часть проекта была значительно упрощена, т.к самому писать библиотеки было бы нерационально, если уже существующие могут обеспечить достижение планируемого результата. Положительным оказалось и то, что объём кода за счёт добавления готовых библиотек сократился примерно в 2 раза.

В функции была добавлена поддержка сторонних приложений для управления светильником. Также количество эффектов доведено до 18 и добавлена почти тонкая настройка каждого из них (*Рисунки 7-11*).

Список эффектов:

1. Конфетти
2. Огонь
3. Радуга вертикальная
4. Радуга горизонтальная
5. Переход
6. Безумие
7. Облака
8. Магма
9. Плазма
10. Радуга
11. Райская птица
12. Зебра
13. Лес
14. Океан
15. Цвет (любой на выбор)
16. Снегопад
17. Матрица (Из одноимённого фильма)
18. Светляки

Добавлена функция под названием «Рассвет»: при достижении установленного времени (при условии, что работа осуществляется через приложение) сначала загорится белый свет, а затем будет постепенно повышаться яркость. Также был добавлен вывод своего текста.

Светильник подключается достаточно просто, но варианты подключения стали более разнообразными. Так, благодаря добавленным библиотекам, появилась возможность заставить светильник работать в режиме точки доступа. В этом режиме можно через WiFi лампы управлять её функциями или подключить к домашнему WiFi и использовать те же функции без необходимости каждый раз подключаться к точке доступа самого изделия.

Светильник может работать и в оффлайн режиме, для этого требуется ранее установленная сенсорная кнопка. Чтобы выключить или включить лампу, требуется кратковременно нажать на кнопку один раз. А чтобы переключать эффекты, нужно кратковременно с небольшой паузой нажать на кнопку два или три раза для перехода на следующий или предыдущий эффект. Одно кратковременное нажатие и удержание вызовет изменение яркости, два – изменение скорости эффекта, три – масштаб.

После написания кода состоялась его проверка, показавшая, что проблем с работоспособностью нет (*Рисунок 12*).

Работа полностью завершённого изделия продемонстрирована в школе, сам светильник в настоящее время украшает комнату в доме автора проекта.

III. Выводы

В результате выполнения проекта достигнуты следующие результаты:

1) расширены знания в области проектирования, изготовления и налаживания бытовой электроники и электротехники, приобретен практический опыт в программировании работы диодной матрицы, сборке

электрических цепей, выполнении паяльных работ, в использовании приборов и инструментов,

2) изучены теоретические основы функционирования светодиодных управляемых ламп и механизм создания программного обеспечения для управления светодиодным светильником,

3) сформированы новые практические умения в области применения базовых знаний языка программирования «Arduino C»: выполнена вёрстка кода для управления светодиодным светильником, проведено промежуточное и окончательное тестирование работоспособности кода,

4) из легкодоступных материалов и компонентов себестоимостью 2 328,7 руб. изготовлен качественный управляемый декоративный светодиодный светильник «Электронный огонь» с 18 красивыми эффектами свечения (*Рисунок 13*), характеризующийся:

- хорошей функциональностью,
- эстетичным внешним видом,
- высоким качеством исполнения,
- малым энергопотреблением,
- экологичностью,
- удобством и безопасностью в использовании,

5) приобретены навыки планирования и организации проектной деятельности и оформления отчётной документации.

IV. Список использованных источников информации

Гуревич Г.И. Как паять. - Москва: Издательство «Энергия», 1964. – 56 с.

Менщиков Ю. Arduino “На пальцах”/ Издательство БГУФРИКТ. – Минск, 2017. – 62с.

Ревич Ю.В. Занимательная электроника. – 3-е изд., перераб. и доп. – СПб.: БВХ-Петербург, 2015. – С.487-540.

Соммер У. Программирование микроконтроллерных плат Arduino/Freduino. – СПб.: БХВ-Петербург, 2012. – 256 с.

Alex Gyver. (2019) Такой светильник не купить в магазине! Электронный огонь. [Видеозапись по созданию изделия] // YouTube. 15 июня (https://www.youtube.com/watch?v=771-Okf0dYs&t=823s&ab_channel=AlexGyver)

V. Приложения



Рисунок 1. Варианты сантехнических изделий, подходящих для изготовления корпуса светильника



Рисунок 2. Готовая основа для сборки светильника

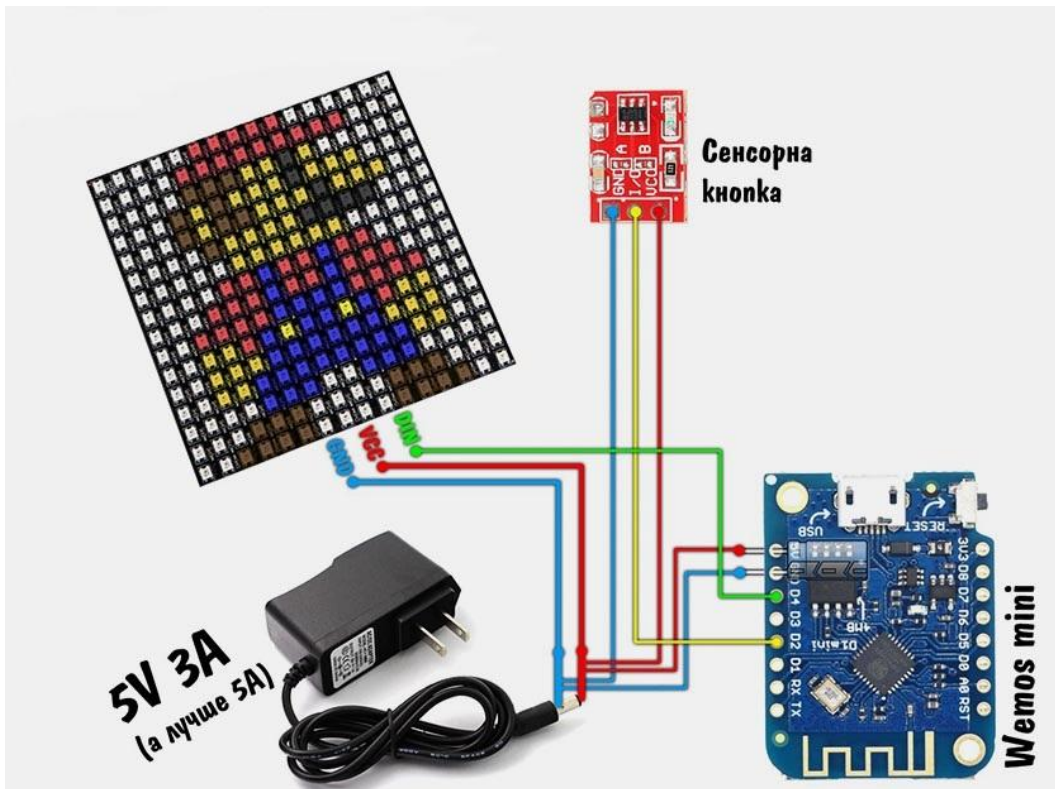


Рисунок 3. Схема для спайки компонентов (* выбрана из открытых Интернет-источников)



Рисунок 4. Внешний вид полностью пропаянной схемы

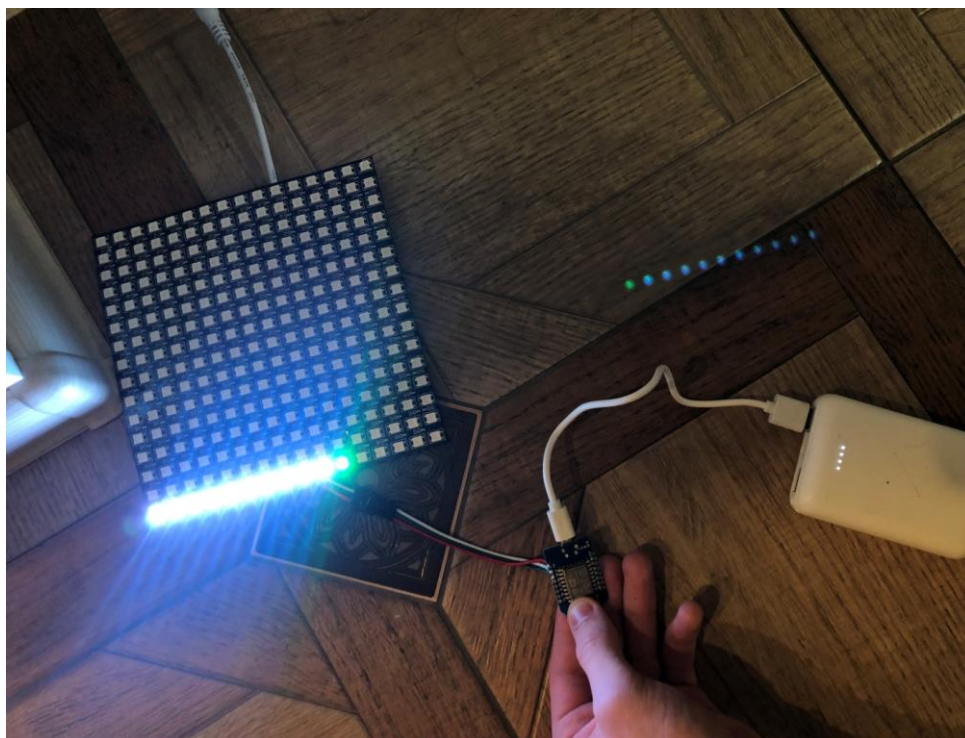


Рисунок 5. Первая тестовая проверка матрицы с использованием внешнего источника питания



Рисунок 6. Крепление готовой матрицы на корпусе

```
GyverLamp_v1.5.5 | Arduino 1.8.19
Файл Правка Скетч Инструменты Помощь

GyverLamp_v1.5.5$ button asprom effectTicker effects fonts.h noiseEffects parsing runningText time timer2Minim.h utility

#define COLOR_ORDER GRB // порядок цветов на ленте. Если цвет отображается некорректно - меняйте. Начать можно с RGB

#define MATRIX_TYPE 0 // тип матрицы: 0 - зигзаг, 1 - параллельная
#define CONNECTION_ANGLE 0 // угол подключения: 0 - левый нижний, 1 - левый верхний, 2 - правый верхний, 3 - правый нижний
#define STRIP_DIRECTION 0 // направление ленты из угла: 0 - вправо, 1 - вверх, 2 - влево, 3 - вниз

// ----- ESP -----
#define ESP_MODE 1
// 0 - точка доступа
// 1 - локальный
byte IP_AP[] = {192, 168, 4, 66}; // статический IP точки доступа (менять только последнюю цифру)

// ----- AP (точка доступа) -----
#define AP_SSID "AutoAP"
#define AP_PASS "12345678"
#define AP_PORT 8888

// ----- Менеджер WiFi -----
#define AC_SSID "AutoConnectAP"
#define AC_PASS "12345678"

// ===== ДЛЯ РАЗРАБОТЧИКОВ =====
#define LED_PIN D4 // пин ленты
#define BTN_PIN D2
#define MODE_AMOUNT 18
```

Рисунок 7. Главная страница

```
GyverLamp_v1.5.5 - effects.ino | Arduino 1.8.19
Файл Правка Скетч Инструменты Помощь

GyverLamp_v1.5.5$ button asprom effectTicker effects fonts.h noiseEffects parsing runningText time timer2Minim.h utility

// ===== ЭФФЕКТЫ =====

// ----- конфетти -----
void sparklesRoutine() {
  for (byte i = 0; i < modes[0].scale; i++) {
    byte x = random(0, WIDTH);
    byte y = random(0, HEIGHT);
    if (getPixColorXY(x, y) == 0)
      leds[getPixelNumber(x, y)] = CHSV(random(0, 255), 255, 255);
  }
  fader(70);
}

// функция плавного угасания цвета для всех пикселей
void fader(byte step) {
  for (byte i = 0; i < WIDTH; i++) {
    for (byte j = 0; j < HEIGHT; j++) {
      fadePixel(i, j, step);
    }
  }
}

void fadePixel(byte i, byte j, byte step) { // новый фейдер
  int pixelNum = getPixelNumber(i, j);
  if (getPixColor(pixelNum) == 0) return;

  if (leds[pixelNum].r >= 30 ||
      leds[pixelNum].g >= 30 ||
      leds[pixelNum].b >= 30) {
```

Рисунок 8. Страница эффектов. «Конфетти».

```

GyverLamp_v1.5.5 - effects.ino | Arduino 1.8.8
Файл Правка Скетч Инструменты Помощь

GyverLamp_v1.5.5 button eeprom effectTicker effects fonts.h noiseEffects parsing

// ===== ЭФФЕКТЫ =====

// ----- конфетти -----
void sparklesRoutine() {
  for (byte i = 0; i < modes[0].scale; i++) {
    byte x = random(0, WIDTH);
    byte y = random(0, HEIGHT);
    if (getPixColorXY(x, y) == 0)
      leds[getPixelNumber(x, y)] = CHSV(random(0, 255), 255, 255);
  }
  fader(70);
}

// функция плавного угасания цвета для всех пикселей
void fader(byte step) {
  for (byte i = 0; i < WIDTH; i++) {
    for (byte j = 0; j < HEIGHT; j++) {
      fadePixel(i, j, step);
    }
  }
}

void fadePixel(byte i, byte j, byte step) { // новый фейдер
  int pixelNum = getPixelNumber(i, j);
  if (getPixColor(pixelNum) == 0) return;

  if (leds[pixelNum].r >= 30 ||
      leds[pixelNum].g >= 30 ||
      leds[pixelNum].b >= 30) {
    leds[pixelNum].fadeToBlackBy(step);
  } else {
    leds[pixelNum] = 0;
  }
}

// ----- ОГОНЬ -----
// эффект "огонь"
#define SPARKLES 1 // вылетающие угольки вкл выкл
unsigned char line[WIDTH];
int pcnt = 0;

//these values are substracted from the generated values to give a shape to the animation
const unsigned char valueMask[8][16] PROGMEM = {
  {32, 0, 0, 0, 0, 0, 0, 0, 32, 32, 0, 0, 0, 0, 0, 0, 32},
  {64, 0, 0, 0, 0, 0, 0, 0, 64, 64, 0, 0, 0, 0, 0, 0, 64},
  {96, 32, 0, 0, 0, 0, 32, 96, 96, 32, 0, 0, 0, 0, 32, 96},
  {128, 64, 32, 0, 0, 32, 64, 128, 128, 64, 32, 0, 0, 32, 64, 128},
  {160, 96, 64, 32, 32, 64, 96, 160, 160, 96, 64, 32, 32, 64, 96, 160}.
};

```

Сохранено.

```

exit status 1

```

Рисунок 9. Страница эффектов

```

GyverLamp_v1.5.5 - effects.ino | Arduino 1.8.8
Файл Правка Скetch Инструменты Помощь

GyverLamp_v1.5.5 button eeprom effectTicker effects fonts.h noiseEffects parsing

byte hue;
// ----- радуга -----
void rainbowVertical() {
  hue += 2;
  for (byte j = 0; j < HEIGHT; j++) {
    CHSV thisColor = CHSV((byte)(hue + j * modes[2].scale), 255, 255);
    for (byte i = 0; i < WIDTH; i++)
      drawPixelXY(i, j, thisColor);
  }
}
void rainbowHorizontal() {
  hue += 2;
  for (byte i = 0; i < WIDTH; i++) {
    CHSV thisColor = CHSV((byte)(hue + i * modes[3].scale), 255, 255);
    for (byte j = 0; j < HEIGHT; j++)
      drawPixelXY(i, j, thisColor); //leds[getPixelNumber(i, j)] = thisColor;
  }
}
// ----- ЦБЕТА -----
void colorsRoutine() {
  hue += modes[4].scale;
  for (int i = 0; i < NUM_LEDS; i++) {
    leds[i] = CHSV(hue, 255, 255);
  }
}
// ----- ЦБЕТ -----
void colorRoutine() {
  for (int i = 0; i < NUM_LEDS; i++) {
    leds[i] = CHSV(modes[14].scale * 2.5, 255, 255);
  }
}
// ----- снегопад -----
void snowRoutine() {
  // сдвигаем всё вниз
  for (byte x = 0; x < WIDTH; x++) {
    for (byte y = 0; y < HEIGHT - 1; y++) {
      drawPixelXY(x, y, getPixColorXY(x, y + 1));
    }
  }

  for (byte x = 0; x < WIDTH; x++) {
    // заполняем случайно верхнюю строку
    // а также не даём двум блокам по вертикали вместе быть
    if (getPixColorXY(x, HEIGHT - 2) == 0 && (random(0, modes[15].scale) == 0))

```

Сохранено.

```

exit status 1

```

Рисунок 10. Страница эффектов. Радуга. Снегопад.

```
© GyverLamp_v1.5.5 - effects.ino | Arduino 1.8.8
Файл Правка Скetch Инструменты Помощь

GyverLamp_v1.5.5 button eeprom effectTicker effects fonts.h noiseEffects parsing

for (byte x = 0; x < WIDTH; x++) {
  // заполняем случайно верхнюю строку
  // а также не даём двум блокам по вертикали вместе быть
  if (getPixColorXY(x, HEIGHT - 2) == 0 && (random(0, modes[15].scale) == 0))
    drawPixelXY(x, HEIGHT - 1, 0xE0FFFF - 0x101010 * random(0, 4));
  else
    drawPixelXY(x, HEIGHT - 1, 0x000000);
}
}

// ----- МАТРИЦА -----
void matrixRoutine() {
  for (byte x = 0; x < WIDTH; x++) {
    // заполняем случайно верхнюю строку
    uint32_t thisColor = getPixColorXY(x, HEIGHT - 1);
    if (thisColor == 0)
      drawPixelXY(x, HEIGHT - 1, 0x00FF00 * (random(0, modes[16].scale) == 0));
    else if (thisColor < 0x002000)
      drawPixelXY(x, HEIGHT - 1, 0);
    else
      drawPixelXY(x, HEIGHT - 1, thisColor - 0x002000);
  }

  // сдвигаем всё вниз
  for (byte x = 0; x < WIDTH; x++) {
    for (byte y = 0; y < HEIGHT - 1; y++) {
      drawPixelXY(x, y, getPixColorXY(x, y + 1));
    }
  }
}

// ----- СВЕТЛЯКИ -----
#define LIGHTERS_AM 100
int lightersPos[2][LIGHTERS_AM];
int8_t lightersSpeed[2][LIGHTERS_AM];
CHSV lightersColor[LIGHTERS_AM];
byte loopCounter;

int angle[LIGHTERS_AM];
int speedV[LIGHTERS_AM];
int8_t angleSpeed[LIGHTERS_AM];

void lightersRoutine() {
  if (loadingFlag) {
    loadingFlag = false;
    randomSeed(millis());
  }
}

Сохранено.
exit status 1
*** [1] ***
*** [1] ***
```

Рисунок 11. Страница эффектов. Матрица. Светляки.



Рисунок 12. Проверка работы кода. Эффект «Огонь».



Рисунок 13. Фотографии готового светильника в процессе работы