

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Министерство образования и науки Российской Федерации | | | | |
| федеральное государственное бюджетное образовательное учреждение  высшего образования | | | | |
| «Иркутский государственный университет» | | | | |
| (ФГБОУ ВО «ИГУ») | | | | |
| Факультет бизнес-коммуникаций и информатики | | | | |
| Кафедра естественнонаучных дисциплин | | | | |
|  |  | | |
|  | |  | | |
|  | |  | | |
|  | |  | | |
| **ОТЧЕТ ПО УЧЕБНОЙ ПРАКТИКЕ**  **Б2.В.03(У) Научно-исследовательская работа**  **(получение первичных навыков научно-исследовательской работы)**  **по направлению 09.03.03 «Прикладная информатика» (прикладной бакалавриат)**  **Профиль «Прикладная информатика (разработка программного обеспечения)»** | | | | |
|  | | | | |
|  | |  | | |
|  | | | Студент 3 курса очной формы обучения | |
|  | | | группа 14321-ДБ | |
|  | | | Сизых Владислав Сергеевич | |
|  | | |  | |
|  | | | Руководитель: к.т.н., доцент  \_\_\_\_\_\_\_\_\_\_ Иван Сергеевич Петрушин | |
|  | | |  | |
|  | | |  | |
|  | | | Работа защищена: | |
|  | | | «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2023 г. | |
|  | | | С оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_ | |
|  | | | Протокол № \_\_\_\_\_\_ | |
|  | | |  | |
|  | | | | |

**СОДЕРЖАНИЕ**

[Введение 3](#_Toc139550101)

[Концепция проекта 5](#_Toc139550102)

[1. Изучение аналогов 5](#_Toc139550103)

[2. Проектирование светильника, а также подбор комплектующих 7](#_Toc139550104)

[3. Проектирование функциональных возможностей светильника 9](#_Toc139550105)

[Вывод по главе 12](#_Toc139550106)

[Моделирование корпуса 14](#_Toc139550107)

[1. Создание (моделирование) 3д-модели корпуса для модуля светильника 14](#_Toc139550108)

[Вывод по главе 16](#_Toc139550109)

[Написание кода 18](#_Toc139550110)

[1. Подбор библиотек для проекта 18](#_Toc139550111)

[2. Проектирование логики кода 18](#_Toc139550112)

[3. Пайка комплектующих и сборка модулей 19](#_Toc139550113)

[Вывод по главе 21](#_Toc139550114)

[Заключение 23](#_Toc139550115)

[Список используемых источников 24](#_Toc139550116)

[Приложения 26](#_Toc139550117)

# Введение

В современном мире дизайн интерьера становится все более важным и влияет на наше настроение и комфорт в помещении. Особое внимание уделяется освещению, которое не только создает атмосферу, но и является важным элементом декора. Именно поэтому модульные светильники стали популярным решением для многих дизайнеров и владельцев жилых и коммерческих помещений. В ходе практической работы будет рассмотрен процесс создания модульного светильника, его преимущества и способы применения в различных интерьерах.

Цель работы – разработать многофункциональный модульный светильник и описать процесс его создания, включая написание кода для управления его функциями и создание 3D-модели корпуса.

Для достижения цели были поставлены следующие **задачи**:

1. Изучение аналогов.
2. Проектирование светильника, а также подбор комплектующих.
3. Проектирование функциональных возможностей светильника.
4. Создание (моделирование) 3д-модели корпуса для модуля светильника.
5. Проектирование кода с учетом возможности горячей замены любого модуля светильника, а также подбор библиотек.
6. Пайка комплектующих и сборка модулей.
7. Написание веб-интерфейса для комфортного взаимодействия пользователя со светильником.

**Объект исследования**: многофункциональный модульный светильник.

**Предмет исследования**: процесс проектирования и создания многофункционального модульного светильника, включая написание кода для управления его функциями и создание 3D-модели корпуса.

Результаты практической работы могут быть полезны следующим группам пользователей:

1. Производителям светильников: результаты исследования могут помочь им в разработке и выпуске новых модульных светильников с улучшенными функциями и возможностями.
2. Дизайнерам интерьеров: результаты исследования могут помочь им в выборе и использовании модульных светильников для создания эффективного и эстетически привлекательного освещения в различных помещениях.
3. Инженерам и программистам: результаты исследования могут быть полезны при разработке кода для управления функциями модульных светильников, а также при создании 3D-моделей корпусов.
4. Потребителям: результаты исследования могут помочь им в выборе и покупке модульных светильников, учитывая их функциональность, энергоэффективность и дизайн.

# Концепция проекта

## Изучение аналогов

В ходе изучения аналогов было выявлено что самые качественные и функциональные модульные светильники производит компания nanoleaf.



Рисунок 1 – Вариант модульного светильника от компании nanoleaf

Эти же светильники оказались и самыми дорогими на текущий момент. Каждый модуль представляет собою равносторонний треугольник со стороной – 24 см. Вместе можно соединить до 30 модулей. Питание при этом будет подаваться только к одной любой панели от контроллера, а от нее будет поступать ко всем остальным. Возможность включать систему и переключаться между режимами предоставляется с помощью контроллера, расположенного на кабеле питания. А если нужно тонко настроить цвет, яркость или контраст светильника, то нужно использовать мобильное приложение (iOS или Android). При этом предоставляется возможность настройки света для каждой отдельной панели. Series. Приложение связывается со светильником при помощи сети Wi-Fi. Дополнительно можно менять температуру цвета, создавать анимации, и задавать режимы дневного/ночного света, что является важным функционалом, позволяющим настроить светильник под себя.

Остальные аналоги продаются в основном на aliexspress. Их можно найти, введя в поисковике nanoleaf. Там очень много вариантов светильников, которые могут стоить в разы дешевле, но предоставлять меньше функционала, быть менее качественными и не такими красивыми как nanoleaf. Большинство из них управляется также через Wi-Fi, но есть более дешевые версии с управлением при помощи Bluetooth. Некоторые варианты модульных светильников обладают встроенным микрофоном, позволяющим светильнику реагировать на музыку. Также на просторах aliexpress можно встретить модульные светильники в форме шестиугольника:



Рисунок 2 – Модульный светильник в форме шестиугольника

Подробное изучение аналогов позволило узнать какие технологии можно применить в модульном светильнике и облегчило стадию проектирования своего светильника за счет анализа как хороших вариантов модульных светильников, так и не очень.

## Проектирование светильника, а также подбор комплектующих

В качестве формы модульного светильника был выбран равносторонний треугольник со стороною 15 см. Связано это с тем, что из треугольников можно собрать наиболее эстетичные формы и по отзывам в интернете, именно форма треугольника является оптимальной на текущий момент.

Для взаимодействия светильника с пользователем было решено использовать веб-интерфейс, который будет запускаться на контроллере, который будет вынесен в отдельный корпус. Этот контроллер не будет входить в модуль. Он вынесен отдельно и будет отвечать за питание всей системы, а также управление светильниками. В дальнейшем этот контроллер будет называться master, так как он будет отвечать за обнаружение модулей и обеспечение коммуникации между ними.

Для возможности управления подсветкой каждого отдельного модуля, и также возможности сбора любой произвольной фигуры из частей светильника, в каждом модуле необходим свой контроллер (slave). Этот контроллер будет прослушивать сообщения главного контроллера (master), и воспроизводить подсветку в своем модуле, согласно инструкциям.

Согласно размерам корпуса, было решено использовать адресную светодиодную ленту плотностью 60 светодиодов на метр и длинною 12 светодиодов. Лента прокладывается по периметру светильника для наиболее плотного и красивого свечения модуля.

Далее необходимо было выбран коннекторы для соединения модулей между собой. Так, каждому модулю было решено сделать по три коннектора на каждую сторону. Один коннектор типа **папа** используется для подключения модуля к другим модулям. И два коннектора типа **мама** для подключения к нашему модулю других модулей. Именно при организации коннекторов таким образом не возникнет проблем при соединении модулей между собою и создании различных фигур из светильников.

Для хорошей сцепки модулей между собою и фиксации при их соединении было решено использовать неодимовые магниты. Также они используются для фиксации крышки модуля к корпусу модуля.

Также еще внутри каждого модуля нужна макетная плата для удобного расположения элементов внутри модуля и качественной связки между собою при помощи пайки.

Теперь, когда спроектирован модуль, необходимо продумать коннекторы для взаимодействия модулей друг с другом.

Проработка коннектора между модулями светильника является важным шагом в разработке, поскольку он обеспечивает правильное соединение и функционирование модулей. В данном случае, коннектор будет размещен вертикально, чтобы избежать возможности перепутать полярность при подключении.

Коннектор будет иметь 5 контактов, каждый из которых выполняет определенную функцию. Два контакта - 5V и GND - предназначены для питания модулей. Контакт 5V обеспечивает постоянное напряжение питания 5 Вольт, а контакт GND - заземление. Два следующих контакта - RX и TX - предназначены для коммуникации между модулями. Контакт RX (приемник) используется для получения данных от других модулей, а контакт TX (передатчик) - для отправки данных другим модулям. Это позволяет модулям светильника взаимодействовать друг с другом и передавать необходимую информацию.

Последний контакт - WD (watch dog) - предназначен для обнаружения подключения нового модуля. Он используется для определения, когда новый модуль был подключен к светильнику.

Все контакты коннектора должны быть обеспечены надежным соединением. Проработка коннектора также должна учитывать электрические и механические характеристики модулей и обеспечивать их правильное взаимодействие.

Общая схема получившегося коннектора есть на рисунке ниже.

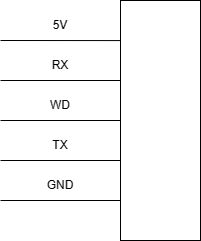


Рисунок 3 – Схема коннектора для соединения модулей между собой

После проработки коннектора, его можно интегрировать в общую конструкцию светильника и приступить к тестированию и оптимизации работы модулей вместе.

## Проектирование функциональных возможностей светильника

Готовый модульный светильник должен обладать следующими функциональными возможностями:

* Настройка подсветки при помощи веб-интерфейса (регулировка яркости, температуры цвета, возможность настройки режимов свечения светильника и переключения анимации).
* Возможность реагирования на звук музыки.
* Возможность настройки дневного/ночного режима.
* Возможность горячей замены модулей светильника.

Возможность горячей замены подразумевает, что во время работы светильника можно подключить еще один модуль и он при этом сразу начнет работать, без перезагрузки системы (горячее подключение). Или можно отключить любой модуль, и система продолжит исправно работать (горячее отключение).

Для реализации этой возможности необходимо найти подходящую библиотеку для коммуникации микроконтроллеров, которая позволит выдавать адрес не во время прошивки устройства, а прямо во время его работы. Получается, система должна обладать динамической адресацией. При попытке поиска библиотеки для коммуникации были найдены следующие:

* Serial. Позволяет наладить коммуникацию между двумя микроконтроллерами при помощи двух пинов. Но не предусматривает общение более 2-ух устройств.
* I2C. Предусматривает общение более 2 устройств. Но адрес устройства нельзя изменить во время работы системы, так как он задается во время прошивки устройства. А значит отпадает горячая замена, добавляется много ограничений. Также, согласно документации, у этого способа подключения более низкая скорость передачи и плохая надежность по сравнению с Serial.
* SPI. Этот способ также предусматривает общение более 2 устройств, но при этом у главного микроконтроллера должен быть провод на каждого соседа (для адресации данных). Уже только из-за этого ограничения не получится использовать данный протокол общения.
* GyverBus. Библиотека-обертка с адресами на основе Serial. Но после продолжительных тестов выяснилось, что версия не стабильна, а документация неполная. Также библиотека заброшена, ввиду отсутствия обновлений за последние несколько лет.

В течение продолжительного поиска подходящей библиотеки не было найдено ни одной. Поэтому было решено написать свою библиотеку с возможностью динамической адресации, широковещательными сообщениями, а также внедрить контроль целостности данных, используя функцию хеширования crc-8.

Библиотека получила название SerialBus, так как в основе своей использует протокол Serial, а точнее SoftwareSerial, который позволяет общаться по Serial на любых цифровых пинах микроконтроллера.

Работа библиотеки подразумевает 1-го главного микроконтроллера (master) и несколько ведомых микроконтроллеров (slave). Для работы библиотеки нужно 3 пина, 2 из которых соединяются, 1 остается нетронутым (используется программно внутри библиотеки).

Соединение микроконтроллеров довольно легкое: RX пин мастера соединяется со всеми пинами TX ведомых. А пин TX мастера соединяется со всеми пинами RX ведомых.

При разработке библиотеки были учтены следующие важные моменты:

* Так как библиотека может компилироваться разными компиляторами (например, Arduino или esp), то нужно было учитывать, что некоторые примитивные типы имеют разный размер. Несоблюдение этой особенности могло было бы нарушить совместимость между Arduino и esp.
* Библиотека внутри себя использует фейковый пин. Это связано с особенностью работы протокола Serial: ведомый контроллер при прослушивании выставляет высокий уровень сигнала на пин TX, тем самым блокируя любую возможность общения остальных ведомых микроконтроллеров с мастером. Поэтому при инициализации Serial общения, библиотека использует фейковый пин, а на настоящем пине TX выставляет статус INPUT\_PULLUP (подтягивание ко внутреннему резистору с питанием), который не будет мешать сигналам других ведомых контроллеров доходить до мастера. В случае, если ведомому контроллеру нужно отправить сообщение мастеру, то он производит замену Serial-объекта (для доступа к настоящему пину TX).
* Внедрение контроля целостности данных на основе хеш-функции crc-8.
* Возможность отправки сообщения, доставка которого подтверждается его получателем. Это позволяет делать ping/pong нужного ведомого контроллера из пользовательского кода в несколько строчек.
* Возможность отправки широковещательного сообщения: главный контроллер может отправить сообщение всем ведомым сразу.

Все это позволило, не усложняя схему, реализовать быстрое и надежное общение микроконтроллеров по общей шине.

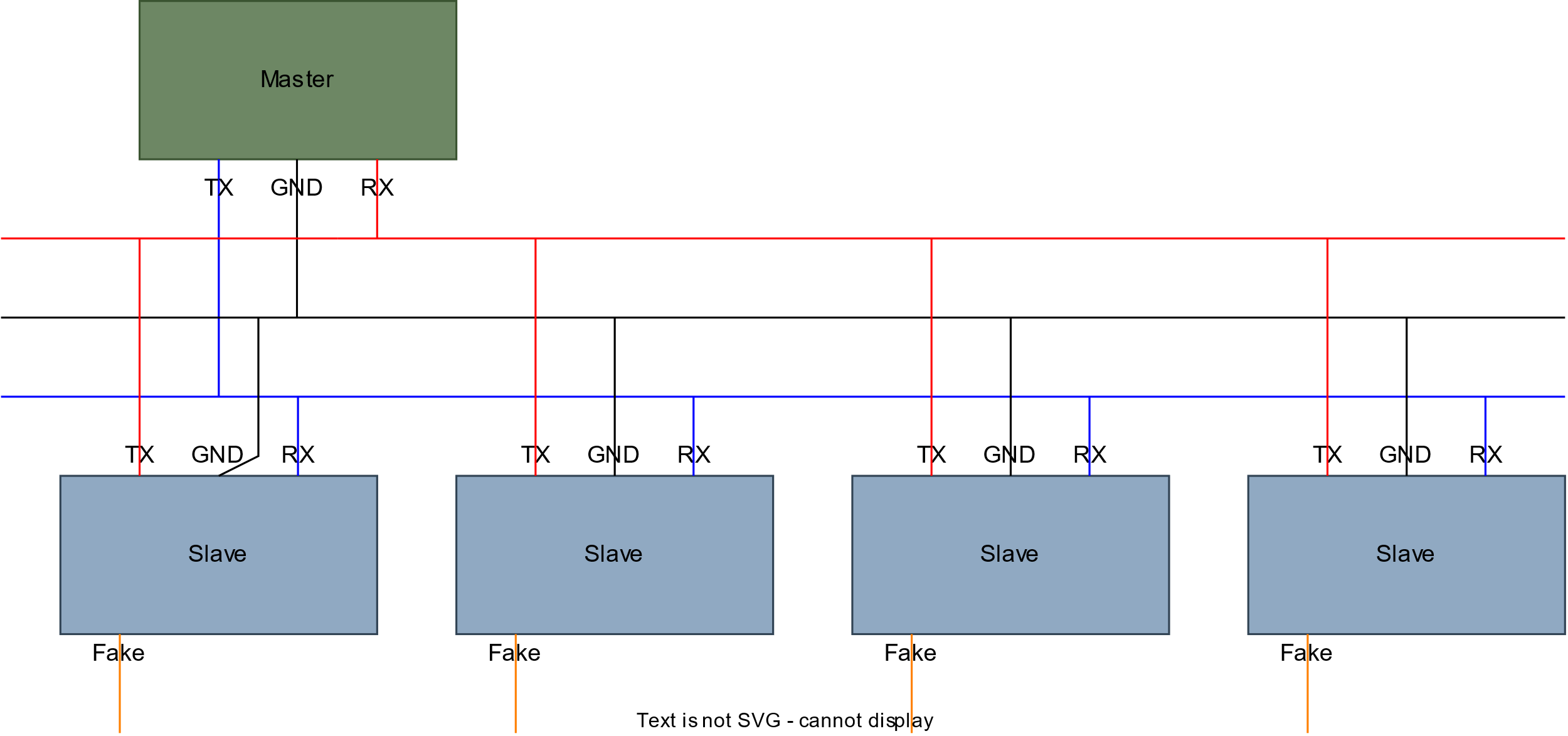


Рисунок 4 – Схема подключения устройств при общении по SerialBus

## Вывод по главе

В данной главе была определена концепция модульного светильника, которая основана на идее создания универсальной системы освещения, которая позволяет гибко настраивать и контролировать световые эффекты. Основной элемент светильника - адресная лента, которая состоит из множества светодиодов, каждый из которых может быть индивидуально управляемым. Это позволяет создавать различные цветовые сцены, эффекты переливающегося света и динамические паттерны.

Для управления адресной лентой был выбран микроконтроллер esp8266, который обладает достаточной вычислительной мощностью и возможностью подключения к Wi-Fi сети. Это позволяет управлять светильником через веб-интерфейс, используя любой браузер на мобильном телефоне.

Для обеспечения коммуникации между микроконтроллерами была разработана специальная библиотека, которая позволяет передавать данные о цвете и яркости светодиодов по протоколу Software Serial. Это обеспечивает надежную и быструю передачу информации между модулями светильника.

В дальнейшей работе будет создана модель светильника в программе Fusion 360. Затем будет разработан дизайн светильника, учитывая требования проекта и доступные ресурсы, такие как материалы и технологии производства.

Итоговый модульный светильник будет представлять собой современное и эстетичное изделие, которое можно будет использовать как в домашней обстановке, так и в коммерческих помещениях.

# Моделирование корпуса

## Создание (моделирование) 3д-модели корпуса для модуля светильника

Для моделирования была выбрана бесплатная программа fusion 360, ввиду нескольких причин:

* Высокая популярность.
* Множество обучающих видеороликов на просторах Интернета.
* Очень дружелюбный интерфейс.

Процесс разработки 3D корпуса для модульного светильника в программе Fusion 360 начинается с создания нового проекта и выбора соответствующего шаблона. Затем происходит создание будущей формы корпуса при помощи инструментов создания скетча. Скетч – что-то вроде чертежа, то есть при помощи примитивных фигур, таких как прямоугольник или многоугольник, создается 2д чертеж, указываются его размеры, а также задаются базовые ограничивающие правила (принцип параллельности прямых или перпендикулярности, например).

После получения готового скетча при помощи инструментов для создания твердотельной геометрии, создается основная форма корпуса. После этого можно приступить к добавлению деталей и функциональных элементов. Например, можно добавить крепления для установки неодимовых магнитов или сделать отверстия для установки коннекторов модуля.

Далее следует провести необходимые исправления и оптимизации модели. В программе Fusion 360 доступны различные инструменты для проверки геометрии и обнаружения ошибок, такие как инструменты для анализа структуры и инструменты для проверки наличия перекрытий или неправильных размеров.

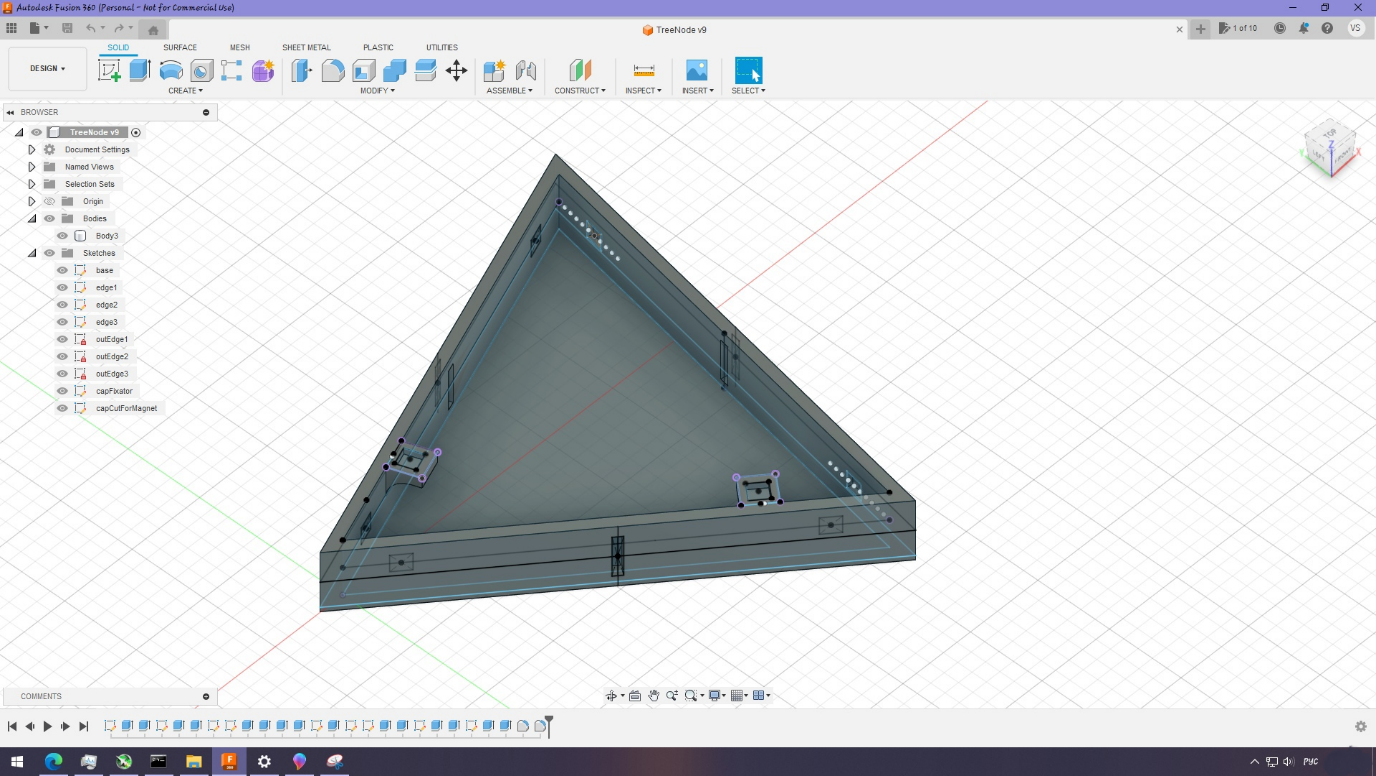


Рисунок 5 – Вид готового корпуса в программе fusion 360

После завершения моделирования корпуса можно приступить к проработке деталей и текстур. В программе Fusion 360 можно добавить текстуры, изменить цвета и материалы, чтобы создать реалистичное представление светильника.

Когда модель корпуса готова, ее нужно экспортировать нужный формат, а именно STL, для последующей печати на 3d принтере.

Печать модели корпуса заняла около 8 часов, и итоговый результат показан на рисунке 6.

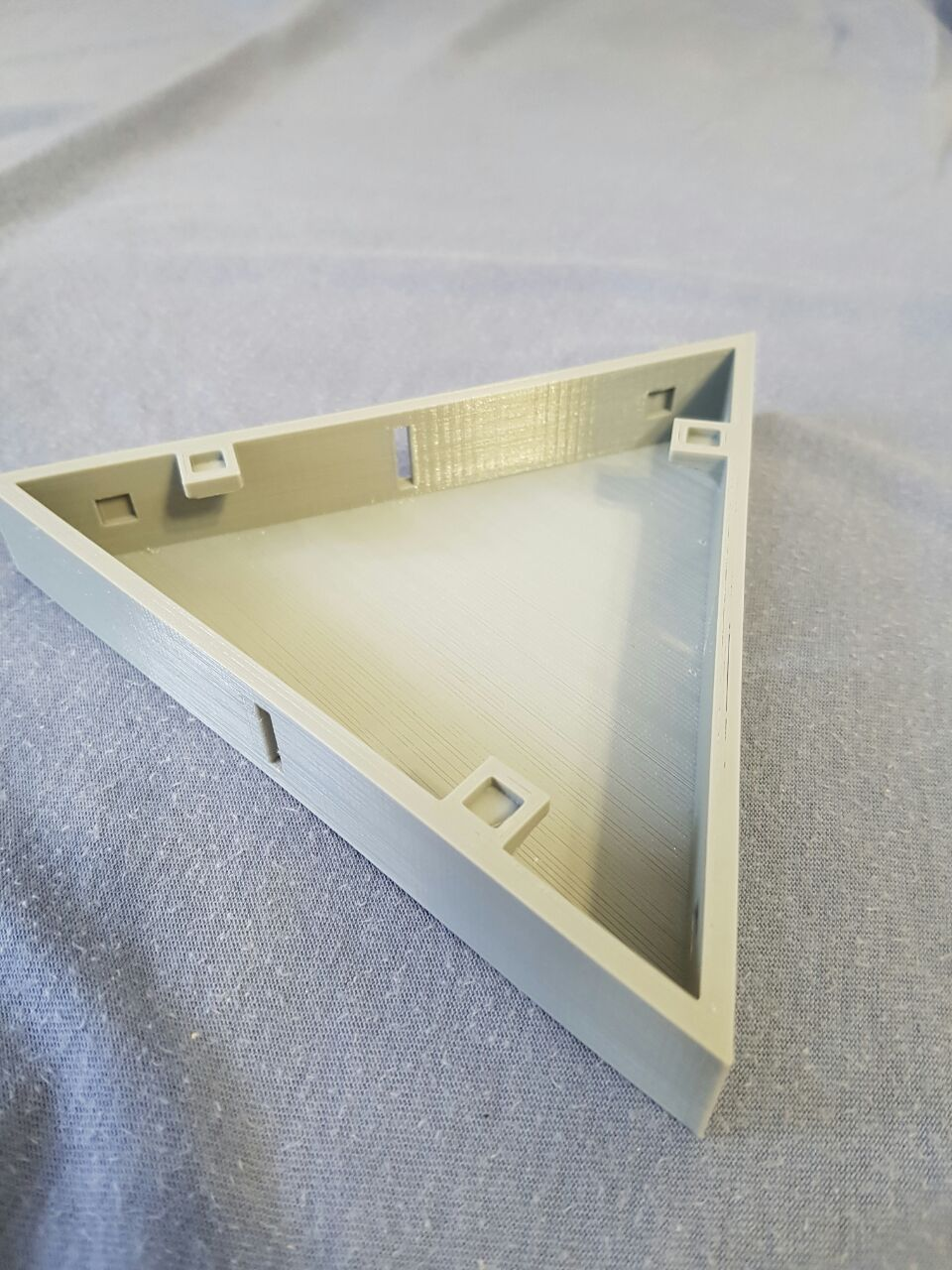


Рисунок 6 – Вид готового напечатанного корпуса

## Вывод по главе

В данной главе была выполнена работа по созданию 3D модели корпуса модульного светильника в программе Fusion 360. Это позволило визуализировать внешний вид и функциональность светильника, а также провести необходимые проверки на соответствие требованиям проекта.

После создания 3D модели, был использован 3D принтер для печати корпуса светильника. Это позволило получить физический прототип, который можно использовать для дальнейших тестов и оценки эргономики и эстетики светильника.

Использование 3D моделирования и 3D печати позволяет значительно сократить время и затраты на разработку и производство прототипов. Такой подход также обеспечивает большую гибкость в изменении дизайна и функциональности светильника, что является важным преимуществом при разработке модульных систем.

Таким образом, создание 3D модели и печать корпуса на 3D принтере являются важными этапами в разработке модульного светильника, которые позволяют оценить его внешний вид, функциональность и эргономику перед переходом к производству в больших масштабах.

# Написание кода

## Подбор библиотек для проекта

Важным критерием при выборе библиотек, которые будут использоваться в проекте, является их популярность и актуальность. Очень важно, чтобы библиотека получала необходимые обновления, которые часто выходят вместе с обновлениями версии ядра прошивок на микроконтроллеры. А благодаря популярности библиотеки, у нее вероятней всего отсутствуют баги и прочие неполадки, а также есть много решенных проблем на форумах пользователей.

Для проекта со светильником были выбраны следующие библиотеки, которые удовлетворяют этим критериям больше своих аналогов:

* FastLED – Библиотека для работы с адресной лентой. Имеет очень большое и сплоченное комьюнити. А также регулярно обновляется. На форумах есть множество примеров кода, демонстрирующего различные анимации и режимы работы ленты.
* ESPAsyncWebSrv – библиотека для создания веб-сервера на базе микроконтроллера esp8266. Позволяет быстро и удобно реализовать веб-интерфейс, при помощи которого можно предоставить пользователю доступ к управлению светильником.
* SerialBus. Библиотека, реализованная мною. Используется ввиду отсутствия каких-либо аналогов в Интернете.

## Проектирование логики кода

Самая трудная проблема при реализации данных модульных светильников – реализация горячей замены. Продумать конечный автомат для данной логики , который не будет тормозить выполнение другой логики и не создаст задержки в общении микроконтроллеров – сложная задача.

Таким образом, отсеяв несколько вариантов, остался один. Его алгоритм работы таков:

* Для главного микроконтроллера (ГМ): ГМ запускается инициализируя нужные пины в функции setup(). В конце этой функции реализован бесконечный цикл, выход из которого будет выполнен только при условии, что МК удалось подсоединиться хотя-бы к одному ведомому микроконтроллеру. После успешного подключения к ведомому контроллеру, логика переходит к функцию loop(). Здесь, ГМ регулярно опрашивает уже подключенные ведомые элементы, чтобы убедиться, что они подключены (ping/pong), а также чтобы убедится, подсоединяли ли к ним модули в последнее время. И в случае, если присоединяли, ГМ проведет операцию выдачи адреса ведомому элементу. В случае если при опросе ведомого элемента, ГМ не получил ответа pong, то он лишает текущего микроконтроллера адреса. Если оказалось, что все модули отключились от ГМ, он перезагрузиться и будет ждать новых подключений. Также с момента запуска, ГМ запускает веб-сервер, и обрабатывает запросы, приходящие с мобильного устройства пользователя на протяжении всей работы светильника.
* Для ведомого микроконтроллера (ВМ): при включении, ВМ попадает в функцию setup(), и инициализирует пины, создает объекты для работы в дальнейшем. В конце функции реализован бесконечный цикл, выход из которого происходит только при условии, что ВМ получил адрес от ГМ. Дальше он попадает в функцию loop(), в которой регулярно проверяет наличие сообщений от мастера и обрабатывает пришедшие запросы. Также с момента включения и на протяжении всей работы, ВМ управляет адресной светодиодной лентой, используя прерывания.

## Пайка комплектующих и сборка модулей

В качестве микроконтроллера, используемого в проекте, была выбрана esp8266. Был выбран именно он по ряду причин:

* Wi-Fi возможности: ESP8266 обладает встроенным Wi-Fi модулем, что позволяет легко подключаться к интернету и обмениваться данными с другими устройствами. Это особенно полезно для проектов, связанных с Интернетом вещей (IoT), мониторингом и управлением удаленных устройств.
* Высокая производительность: ESP8266 оснащен 32-разрядным ядром с достаточной производительностью.
* Низкая стоимость: ESP8266 является относительно недорогим микроконтроллером, что делает его доступным для широкого круга разработчиков и проектов с ограниченным бюджетом.
* Широкая поддержка и сообщество: ESP8266 имеет большое сообщество разработчиков, которые активно создают и делятся библиотеками, примерами кода и решениями. Это облегчает разработку проектов и решение возникающих проблем.

И вот, когда наконец был выбран микроконтроллер и напечатан корпус, можно было приступать к пайке запчастей. Процесс довольно трудоемкий и долгий.

Процесс пайки компонентов включал в себя несколько этапов:

1. Подготовка: перед началом пайки необходимо подготовить все необходимые инструменты и материалы. Это включает в себя паяльник, припой, флюс, держатель для паяльника, нож (для оголения проводов), пинцеты, щипцы и другие инструменты, которые могут понадобиться для работы.

2. Подготовка компонентов: компоненты, которые будут паяться, нужно очистить от окислов и загрязнений.

3. Непосредственно процесс пайки: в процессе пайки применялась паяльная кислота (флюс) для удаления окислы и загрязнения с поверхности контактов, а также улучшения качества пайки. При пайке очень важно избегать перегрева компонента или платы, что может привести к повреждению, особенно при пайке светодиодной ленты.

4. После пайки: после сборки модуля необходимо обязательно проверить отсутствие коротких замыканий на плате, а также корректность соединений. Для этого использовался мультиметр.

Результат сборки одного модуля продемонстрирован на рисунке ниже.

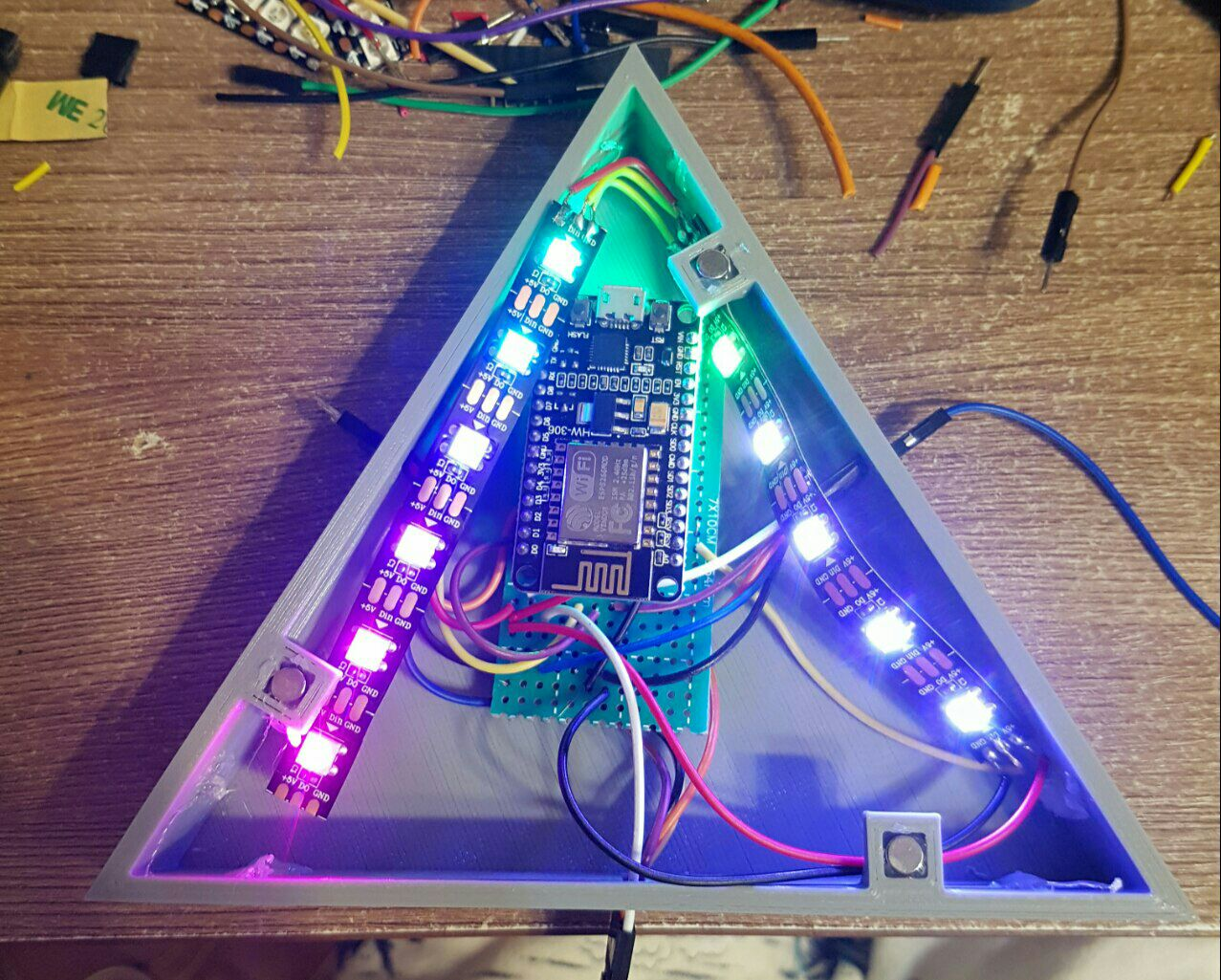


Рисунок 7 – Собранный и запущенный модуль изнутри

Всего было собрано четыре таких модуля.

## Вывод по главе

В данной главе была проведена работа по проектированию логики работы модульного светильника и подбору необходимых библиотек. Затем был написан код на языке C++ для микроконтроллера ESP8266, который обеспечивает совместимость с аналогичными микроконтроллерами Arduino.

Разработка логики работы светильника включала определение функциональности, такой как включение и выключение, обеспечение связи между модулями, реализация горячей замены (горячее включение и горячее отключение), регулировка яркости и цветовой температуры.

Подбор необходимых библиотек позволил использовать готовые решения для работы с различными компонентами светильника, такими как адресная светодиодная лента, Wi-Fi модуль микроконтроллера.

Таким образом, была разработана логика работы модульного светильника и написан код для микроконтроллера.

# Заключение

Подводя итоги, в данной работе был разработан и создан модульный светильник, который имеет ряд преимуществ перед традиционными светильниками. Модульная конструкция позволяет легко заменять и добавлять компоненты, что делает светильник гибким и удобным в использовании.

Основной целью работы было разработать и создать многофункциональный модульный светильник с возможностью горячей замены модулей. Полученный результат является положительным и соответствует поставленной цели. Модульный светильник обладает высокой яркостью и равномерностью освещения, а также имеет низкое энергопотребление. Кроме того, светильник обладает долгим сроком службы.

В заключение, создание и кодирование модульного светильника является интересным и полезным процессом, который требует внимания к деталям и аккуратности. Был получен бесценный опыт создания проекта с нуля, в ходе проекта была создана 3д модель корпуса, спаяны компоненты, написан код для работы светильника, а также проверка его работоспособности.

Исходный код или файл 3д-модели можно посмотреть в репозитории, ссылка на который есть в приложении Б.

Ход работы над практикой можно посмотреть в Task list на сайте notion, ссылка на который есть в приложении В.

# Список используемых источников

1. Объединение нескольких плат Arduino в единую сеть [Электронный ресурс], URL: https://vk.com/@robonora-obedinenie-neskolkih-plat-arduino-v-edinuu-set (дата посещения: 13.06.2023).

2. Используем протокол ModBus с Arduino [Электронный ресурс], URL: https://arduino-tech.ru/2018/08/20/ispolzuem-protokol-modbus-s-arduino/ (дата посещения: 14.06.2023).

3. Умные световые панели nanoleaf [Электронный ресурс], URL: https://nanoleaf.ru/p/nanoleaf-aurora-smarter-kit (дата посещения: 16.06.2023).

4. Связь Нескольких Arduino По Проводу (GyverBus) [Электронный ресурс], URL: https://alexgyver.ru/gyverbus/ (дата посещения: 17.06.2023).

5. Аналог световой панели nanoleaf на сайте aliexpress [Электронный ресурс], URL: https://aliexpress.ru/item/4001365662617.html?sku\_id=12000020662848097&spm=a2g2w.productlist.search\_results.6.5ec14aa6qDoK6U (дата посещения: 18.06.2023).

6. I2C vs SPI vs UART – Introduction and Comparison of their Similarities and Differences [Электронный ресурс], URL: https://www.totalphase.com/blog/2021/12/i2c-vs-spi-vs-uart-introduction-and-comparison-similarities-differences/ (дата посещения: 19.06.2023).

7. КРАТКОЕ МЕТОДИЧЕСКОЕ ПОСОБИЕ ПО AUTODESK FUSION 360 [Электронный ресурс], URL: https://xn--80appmj0e.xn--p1ai/AUTODESK\_FUSION\_360.pdf (дата посещения: 20.06.2023).

8. Гайд По Адресной Светодиодной Ленте [Электронный ресурс], URL: https://alexgyver.ru/ws2812\_guide/ (дата посещения: 2.07.2023).

9. ESP8266 Web Server using SPIFFS (SPI Flash File System) – NodeMCU [Электронный ресурс], URL: https://randomnerdtutorials.com/esp8266-web-server-spiffs-nodemcu/ (дата посещения: 4.07.2023).

10. ESP8266 Timer and Ticker Example [Электронный ресурс], URL: https://circuits4you.com/2018/01/02/esp8266-timer-ticker-example/ (дата посещения: 5.07.2023).

# Приложения

Приложение А

**Ежедневные записи студента по практике**

|  |  |  |
| --- | --- | --- |
| Дата | Описание работы, выполненной студентом | Отметка руководителя о выполнении |
| 12.06 | Ознакомительная лекция по практике |  |
| 13.06 | Планирование задач и оформление notion для более продуктивной работы. |  |
| 14.06 – 15.06 | Разработка концепции проекта |  |
| 16.06 – 17.06, 19.06 | Обучение создания 3д моделей в программе fusion 360 |  |
| 20.06 – 21.06 | Проектирование корпуса и размещение элементов |  |
| 22.06 – 24.06 | Проектирование способа взаимодействия компонент |  |
| 26.06 – 27.06 | Подбор нужных библиотек и проверка работоспособности и пригодности каждой из них |  |
| 28.06 – 29.06 | Анализ способов соединения микроконтроллеров в единую сеть - кольцо, шина, звездочка и т.д. |  |
| 30.06-1.07,3.07 | Написание библиотеки для объединения МК в общую шину при помощи Serial. Подбор, поиск и сбор комплектующих. Заполнение отчета |  |
| 4.07-7.07 | Обновление библиотеки в связи с особенностями работы serial. Пайка схем. Написание кода для взаимодействия модулей во время выполнения (реализация горячей замены). Создание презентации. |  |
| 08.07 | Защита практик |  |

**Приложение Б**

**Ссылка на материалы (исходный код, файл с 3д-моделью)**

<https://github.com/CodeR-na-r1/TriangleNodesARGB/>

**Приложение В**

**Ссылка на notion (Task List по выполнению проекта)**

https://painted-cadmium-edb.notion.site/d2fce76c042f4b5c818a90c46f33241c?v=cedc8d05a25e44d0b224f3074def3c7a&pvs=4