**Курсовой проект по СИФО ЭВМ “RGB LED-панель с тремя модулями”**студентов группы 550504 Козелько Н. С. и Дорошко В. Э.

Описание проекта:

Собрать LED-панель из RGB светодиодов, которая будет отображать текст, вводимый в приложении на компьютере. А также протестировать устройство в условиях, при которых оно должно работать. И решить возникшие проблемы, если они появятся.

Цель проекта:

Получить новые знания и умения работы с Arduino, а также проверить уже полученные знания на практике.

|  |  |  |
| --- | --- | --- |
| План работы | | |
| Неделя | Дата | Работа |
| 1 | 05.03 - 11.03 | Выбор, заказ и отслеживание деталей. |
| 2 | 12.03 - 18.03 | Разработка платы Sprint Layout. Проектирование устройства в Proteus. И предусмотрение проблем, которые могут возникнуть при сборке. |
| 3 | 19.03 - 25.03 | Разработка ПО. Травление платы. |
| 4 | 26.03 - 01.04 | Разработка ПО. Тестирование в Proteus. Травление плат. |
| 5 | 02.04 - 08.04 | Разработка корпуса модулей. Переход к пайке |
| 6 | 09.04 - 15.04 | Пайка. Изготовление корпусов. |
| 7 | 16.04 - 22.04 | Программация и тестирование устройства. |
| 8 | 23.04 - 29.04 | Устранение возникших проблем/ошибок. Тестирование устройства. Написание пояснительной записки. Изготовление корпусов. |
| 9 | 30.04 - 06.05 | Оформление пояснительной записки. |

Примеры похожих работ довольно несложно найти в интернете:





Но в этих примерах нет модульности. Есть один модуль определенного размера, и уже с ним идет работа.

После изучения области, в которой нам придется работать, было решено использовать необычные светодиоды, а smd светодиоды.

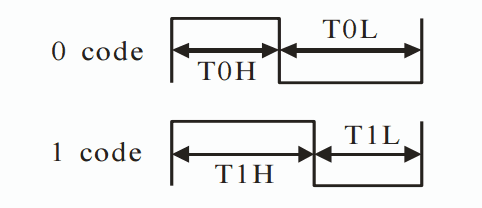


Выгода их использования заключается в том, что при работе с обычными светодиодами нам необходимо работать с каждым светодиодом по отдельности. А при работе с smd светодиодами WS2812B нам необходимо работать напрямую только с первым светодиодом в цепи, пропуская всю нужную информацию через него. Немного подробнее о том, как они работают:

В течение первых 50 ms происходит инициализация ленты путем заземления управляющей шины. После этого контроллером отправляется пачка из пакетов по 24 бита (8 бит на каждый цветовой канал), содержащих информацию о яркости, каждый пакет предназначается для одного диода. Данные в пачке ничем не разделяются, т.е. каждый следующий пакет идет непосредственно за предыдущим. Получив всю пачку (длина пачки равна 24 бита x количество диодов в ленте), контроллер первого диода «откусывает» от нее свой пакет, использует информацию по назначению, а остальную пачку ретранслирует вперед. Таким образом, до последнего диода доходит пачка из одного 24-битного пакета.

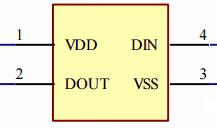
Принцип работы WS2812B — официальная информация из даташита.

Физически в WS2812B имеется 3 излучающих светодиода (красный, синий и зеленый) и ШИМ-драйвера ,управляющие их яркостью. ШИМ-драйвера 8-ми битные, то есть для каждого из цветов возможны 256 градаций яркости и, соответственно, для того чтобы установить яркости для каждого из 3-х светодиодов, нужно передать пикселю 8х3=24 бит (3 байта) информации. Протокол передачи информации светодиоду однолинейный с фиксированной скоростью. Единички и нули информации о яркости кодируются длительностью высокого и низкого уровня сигнала в линии.

****

Время передачи одного бита составляет 0.8+0.45=1.25 мкС — это довольно быстро. Время передачи всего пакета из 24 бит для одного пикселя WS2812B составляет 24\*1.25=30 мкС. Для 1000 штук — 1000\*30=30 мС (что, например, позволяет обновлять по одной линии панно 30х30 пикселей с частотой 30 раз в секунду!).

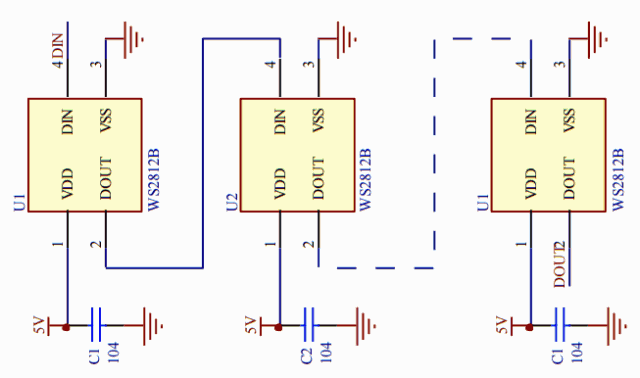
Каждый из пикселей WS2812B имеет 2 вывода питания (VDD, VSS), вход (DIN) и выход (DOUT).



На вход DIN подается информация (24бита) для установки нового цвета. Информация о цвете передается побитно (начиная со старшего бита) последовательно для каждой из составляющей цветов G, R, B.

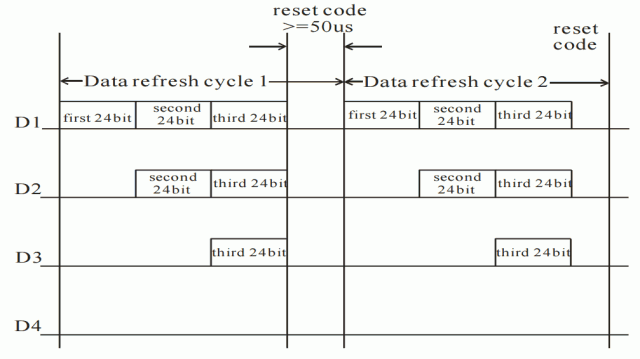
ws2812b передача бит

Пиксели соединяются в цепочку следующим образом:



Запись значений цвета цепочки пикселей происходит следующим способом:

Первые 24 бита, поданные на DIN, записывает себе во временную память (цвет пока остается неизменным с предыдущего раза) первый пиксель. Последующие биты первый пиксель пропускает через себя и выдает на выход DOUT. Второй пиксель повторяет действия первого (оставляя себе первые дошедшие до него 24 бита) и так по цепочке. Для того чтобы значения цветов из временной памяти пикселей стали активными, должна быть выдержана пауза в передаче (reset code) в течение 50мкС. После этой паузы цикл можно повторять снова.

****

То есть в аппаратной части проекта нам необходимо будет сделать:

* цепочку из светодиодов;
* разместить их на каждой плате по 100 штук;
* сделать механизм соединения цепочек светодиодов из двух разных плат в одну цепочку;
* сделать корпус для каждого модуля;
* разработать блок питания для модулей.

В программной части проекта надо будет реализовать:

* скетч, который будет работать со светодиодами через их нумерацию и предоставлять API для управления через последовательный порт;
* desktop-приложение для более удобного управления модулями через предоставляемое API на последовательном порту.

Ход работы (будет обновляться в процессе работы)

|  |  |  |
| --- | --- | --- |
| 1 неделя | 05.03 - 11.03 | Выбор, заказ и отслеживание деталей. |

Данный проект будет представлять из себя изготовление трех светодиодных панелей, блока управления панелями и разработка ПО.

Зная описание проекта, можно выделить элементы, которые нужны для этого устройства:

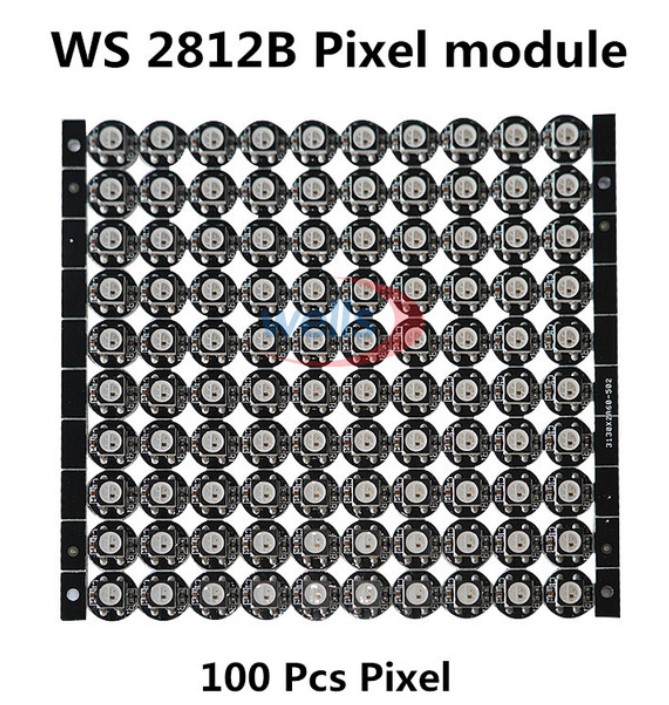
* RGB светодиоды (WS2812b)
* конденсаторы 100нФ \* 300 шт.
* Arduino Uno R3
* двусторонний текстолит
* хлорное железо
* моторчик со сверлом
* материал для боковых стенок модуля
* материал для задней стенки модуля
* материал для передней стенки модуля

Поиски в интернет-магазинах привели к покупке данных элементов:

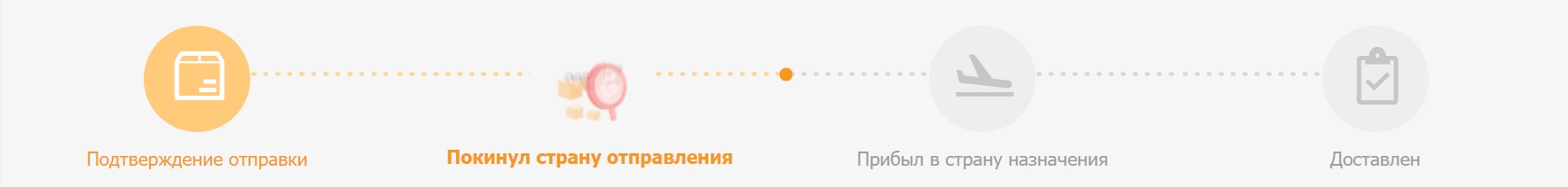
Список необходимых деталей (цены на 05.03)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Название | Количество | Цена | | Стоимость | | Заказано | Куплено/ получено |
| USD | BYN | USD | BYN |
| Arduino Uno R3 | 1 |  |  |  |  | 1 | 1\* |
| RGB светодиоды (WS2812b) | 300 |  | 0.24 |  | 74.40 | 310 | 310 |
| Конденсаторы 100нФ | 300 |  | 0,03 |  | 9 |  | 300 |
| RGB светодиоды (WS2812b) | 100 |  |  |  | 17 | 100 | 100 |
| Оргстекло |  |  |  |  |  |  |  |
| Двусторонний фольгированнный текстолит | 20\*20см |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| Итого: | | | | USD | BYN |  | |
|  | 100.40 |

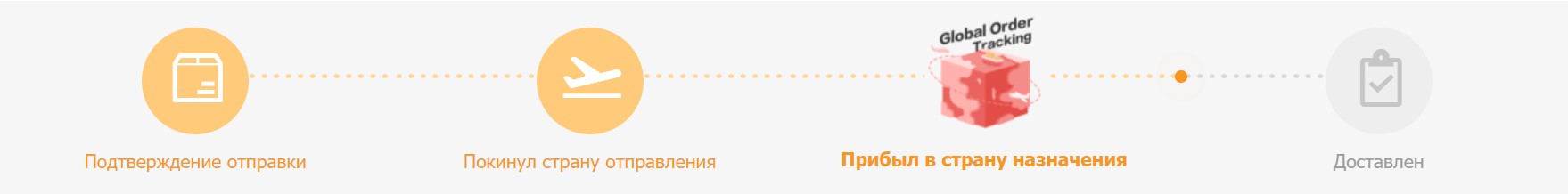
Основные детали были куплены в магазине г. Минска для того, чтобы быстро приступить к работе. Также запасные 100 светодиодов были заказаны с AliExpress.



Ход отслеживания светодиодов с Китая:

Состояние на 25 марта:  


Состояние на 7 апреля:

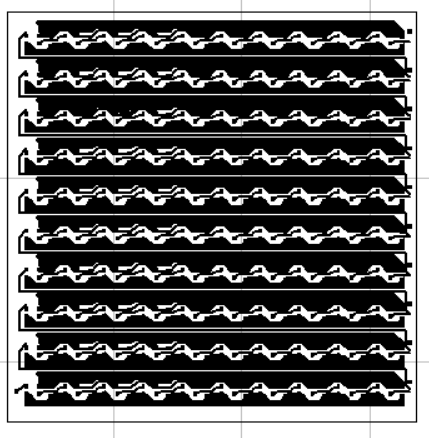
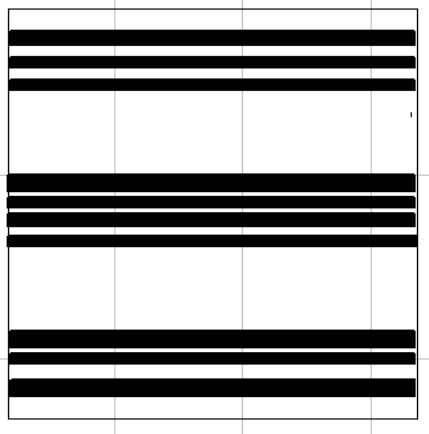


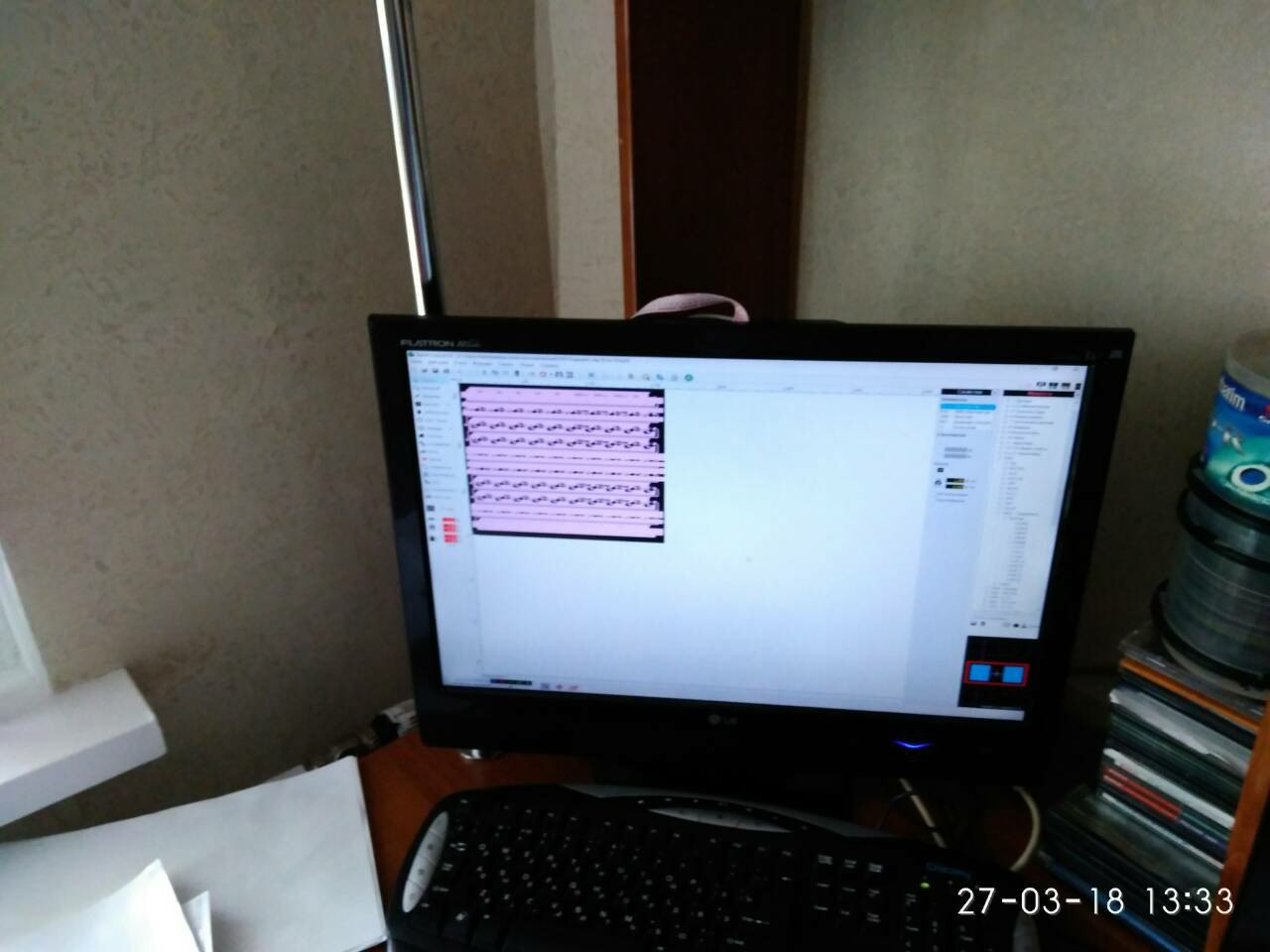
11 апреля пришло извещение, что посылка уже ждет на почте.

По причине выхода из строя своих Arduino Uno R3, во избежания слишком больших затрат, было решено взять Arduino Uno у одногруппников.

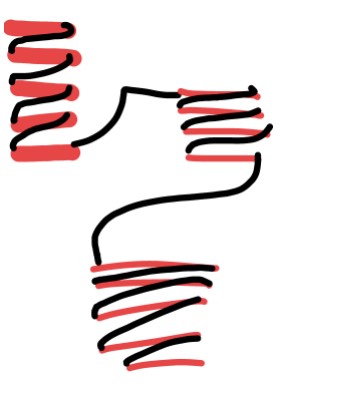
|  |  |  |
| --- | --- | --- |
| 2 неделя | 12.03 - 18.03 | Разработка платы Sprint Layout. Проектирование устройства в Proteus. И предусмотрение проблем, которые могут возникнуть при сборке. |

Плата в Sprint Layout





Размер платы составляет 16\*16 см. Плата отражена зеркально, в печатном виде.  
  
Рисунок платы был выбран не случайно. С таким рисунком платы будет проще писать скетч для проекта (нумерация светодиодов на панели будет идти, как обычно, слева направо и сверху вниз).

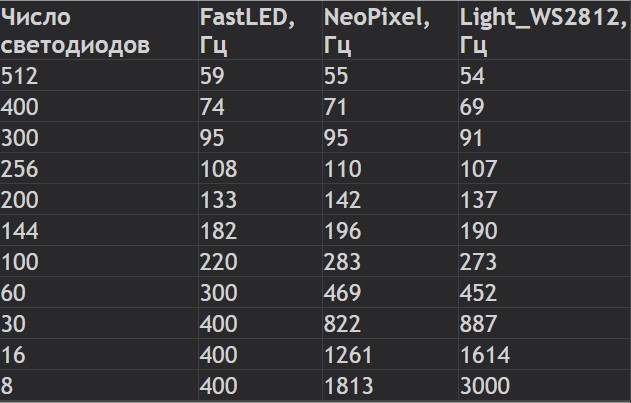


И такая нумерация будет для каждого модуля в отдельности. Что также упрощает работу при написании скетча при анимации сдвига, что будет рассмотрено позже.

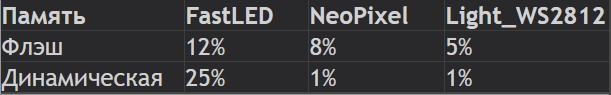
|  |  |  |
| --- | --- | --- |
| 3 неделя | 19.03 - 25.03 | Разработка ПО.Травление платы. |

Ни для кого не секрет, что для управления адресными светодиодами с Ардуины существует две популярные библиотеки: FastLED (независимый разработчик) и NeoPixel (Adafruit). Также еще есть LightWS2812, но написание скетча с её помощью — не самое удобное занятие.

В интернете можно найти сравнение скоростей обновления матрицы светодиодов при использовании данных библиотек:



Также можно найти сравнения по используемой памяти:



По результатам видно, что FastLED медленнее NeoPixel при любом количестве светодиодов, а LightWS2812 показывает просто безумную скорость на маленьком количестве светодиодов. Если в проекте используется более 100 светодиодов, то без разницы, какую библиотеку использовать. FastLED предлагает кучу удобных инструментов и быструю 8-битную математику, чего нет в NeoPixel, а в LightWS2812 нет вообще ничего вспомогательного.

Так как NeoPixel занимает меньше памяти (а мы ограничены имеющейся памятью у Arduino Uno R3) и показалась более удобной в использовании, то выбор пал на неё.

Ссылка на репозиторий выбранной библиотеки:

<https://github.com/adafruit/Adafruit_NeoPixel>

На сайте разработчика данной библиотеки можно найти уже готовые матрицы светодиодов:  
 <https://www.adafruit.com>

На desktop-приложении было решено писать на C# и использовать WPF для GUI. Также на данном языке упрощена работа с обменом информации через последовательный порт.

**C#** (произносится си шарп) — [объектно-ориентированный](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) [язык программирования](https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F). Разработан в [1998](https://ru.wikipedia.org/wiki/1998)—[2001 годах](https://ru.wikipedia.org/wiki/2001_%D0%B3%D0%BE%D0%B4) группой инженеров компании [Microsoft](https://ru.wikipedia.org/wiki/Microsoft) под руководством [Андерса Хейлсберга](https://ru.wikipedia.org/wiki/%D0%A5%D0%B5%D0%B9%D0%BB%D1%81%D0%B1%D0%B5%D1%80%D0%B3,_%D0%90%D0%BD%D0%B4%D0%B5%D1%80%D1%81) и Скотта Вильтаумота как язык разработки приложений для платформы [Microsoft .NET Framework](https://ru.wikipedia.org/wiki/.NET_Framework). Впоследствии был стандартизирован как [ECMA](https://ru.wikipedia.org/wiki/ECMA)-334 и [ISO](https://ru.wikipedia.org/wiki/ISO)/[IEC](https://ru.wikipedia.org/wiki/IEC) 23270.

C# относится к семье языков с [C-подобным синтаксисом](https://ru.wikipedia.org/wiki/C-%D0%BF%D0%BE%D0%B4%D0%BE%D0%B1%D0%BD%D1%8B%D0%B9_%D1%81%D0%B8%D0%BD%D1%82%D0%B0%D0%BA%D1%81%D0%B8%D1%81), из них его синтаксис наиболее близок к [C++](https://ru.wikipedia.org/wiki/C%2B%2B) и [Java](https://ru.wikipedia.org/wiki/Java). Язык имеет [статическую типизацию](https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D0%B0%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D1%82%D0%B8%D0%BF%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F), поддерживает [полиморфизм](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BB%D0%B8%D0%BC%D0%BE%D1%80%D1%84%D0%B8%D0%B7%D0%BC_(%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)), [перегрузку операторов](https://ru.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D0%B5%D0%B3%D1%80%D1%83%D0%B7%D0%BA%D0%B0_%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BE%D0%B2) (в том числе операторов явного и неявного приведения типа), [делегаты](https://ru.wikipedia.org/wiki/%D0%94%D0%B5%D0%BB%D0%B5%D0%B3%D0%B0%D1%82_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)), атрибуты, [события](https://ru.wikipedia.org/wiki/%D0%A1%D0%BE%D0%B1%D1%8B%D1%82%D0%B8%D0%B9%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5), [свойства](https://ru.wikipedia.org/wiki/%D0%A1%D0%B2%D0%BE%D0%B9%D1%81%D1%82%D0%B2%D0%BE_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)), [обобщённые](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D0%BE%D0%B1%D1%89%D1%91%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) типы и методы, [итераторы](https://ru.wikipedia.org/wiki/%D0%98%D1%82%D0%B5%D1%80%D0%B0%D1%82%D0%BE%D1%80), [анонимные функции](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%BE%D0%BD%D0%B8%D0%BC%D0%BD%D0%B0%D1%8F_%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D1%8F) с поддержкой [замыканий](https://ru.wikipedia.org/wiki/%D0%97%D0%B0%D0%BC%D1%8B%D0%BA%D0%B0%D0%BD%D0%B8%D0%B5_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)), [LINQ](https://ru.wikipedia.org/wiki/Language_Integrated_Query), [исключения](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B0_%D0%B8%D1%81%D0%BA%D0%BB%D1%8E%D1%87%D0%B5%D0%BD%D0%B8%D0%B9), [комментарии](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%80%D0%B8%D0%B8_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)) в формате [XML](https://ru.wikipedia.org/wiki/XML).

Переняв многое от своих предшественников — языков [C++](https://ru.wikipedia.org/wiki/C%2B%2B), [Pascal](https://ru.wikipedia.org/wiki/%D0%9F%D0%B0%D1%81%D0%BA%D0%B0%D0%BB%D1%8C_(%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)), [Модула](https://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D1%83%D0%BB%D0%B0-2), [Smalltalk](https://ru.wikipedia.org/wiki/Smalltalk) и, в особенности, [Java](https://ru.wikipedia.org/wiki/Java) — С#, опираясь на практику их использования, исключает некоторые модели, зарекомендовавшие себя как проблематичные при разработке программных систем, например, C# в отличие от C++ не поддерживает [множественное наследование](https://ru.wikipedia.org/wiki/%D0%9C%D0%BD%D0%BE%D0%B6%D0%B5%D1%81%D1%82%D0%B2%D0%B5%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BD%D0%B0%D1%81%D0%BB%D0%B5%D0%B4%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) классов (между тем допускается [множественное наследование интерфейсов](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_(%D0%BE%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)#%D0%9C%D0%BD%D0%BE%D0%B6%D0%B5%D1%81%D1%82%D0%B2%D0%B5%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BD%D0%B0%D1%81%D0%BB%D0%B5%D0%B4%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_%D0%B8_%D1%80%D0%B5%D0%B0%D0%BB%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81%D0%BE%D0%B2)).

**Windows Presentation Foundation** (WPF) — система для построения клиентских приложений [Windows](https://ru.wikipedia.org/wiki/Windows) с визуально привлекательными возможностями взаимодействия с пользователем, графическая (презентационная) подсистема в составе [.NET Framework](https://ru.wikipedia.org/wiki/.NET_Framework) (начиная с версии [3.0](https://ru.wikipedia.org/wiki/.NET_Framework_3.0)), использующая язык [XAML](https://ru.wikipedia.org/wiki/XAML).

WPF предустановлена в [Windows Vista](https://ru.wikipedia.org/wiki/Windows_Vista) ([.NET Framework](https://ru.wikipedia.org/wiki/.NET_Framework) 3.0), [Windows 7](https://ru.wikipedia.org/wiki/Windows_7) ([.NET Framework](https://ru.wikipedia.org/wiki/.NET_Framework) 3.5 SP1), [Windows 8](https://ru.wikipedia.org/wiki/Windows_8) (.NET Framework 4.0 и 4.5), [Windows 8.1](https://ru.wikipedia.org/wiki/Windows_8.1) (.NET Framework 4.5.1). С помощью WPF можно создавать широкий спектр как автономных, так и запускаемых в [браузере](https://ru.wikipedia.org/wiki/%D0%91%D1%80%D0%B0%D1%83%D0%B7%D0%B5%D1%80) приложений.

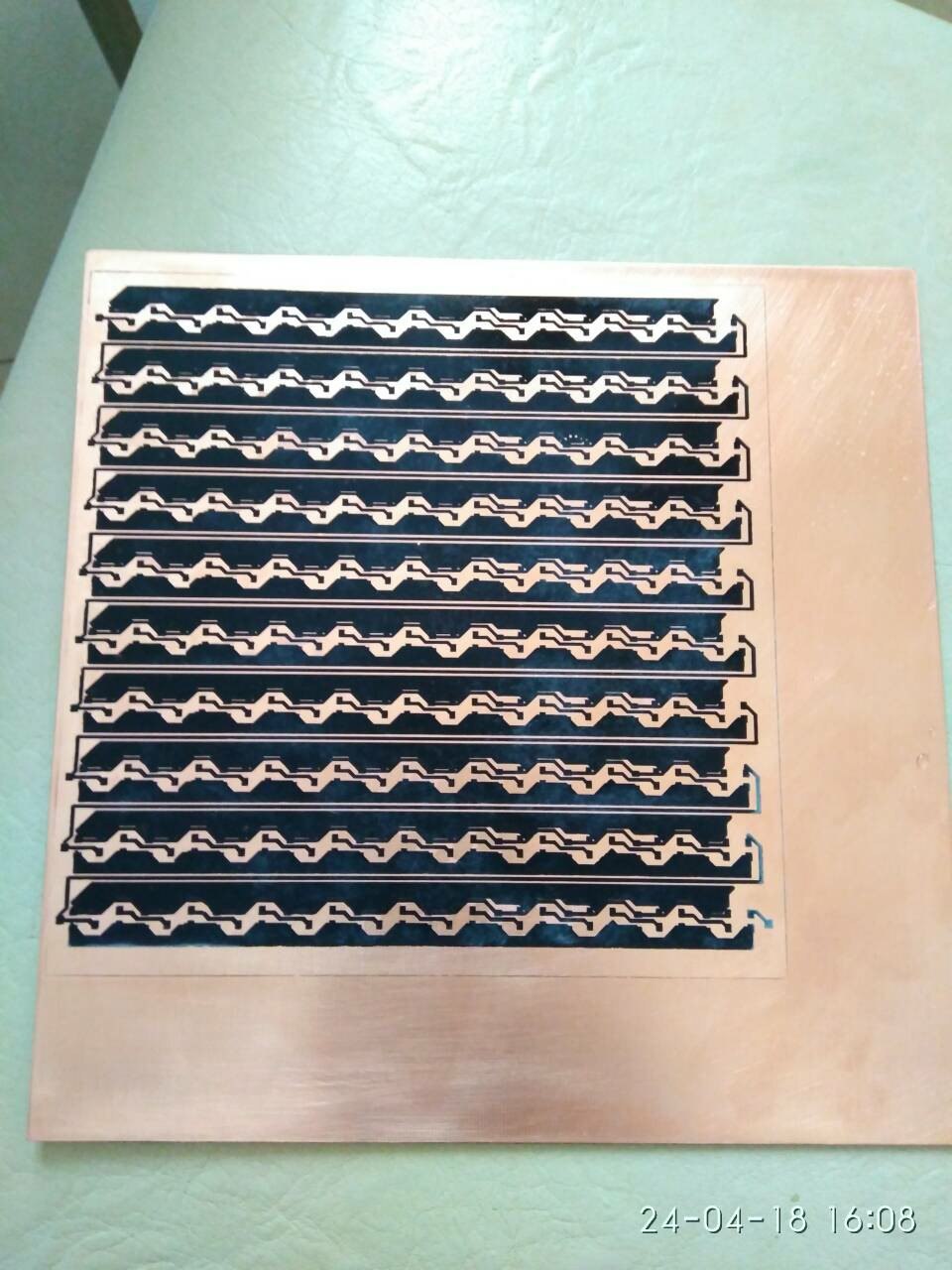
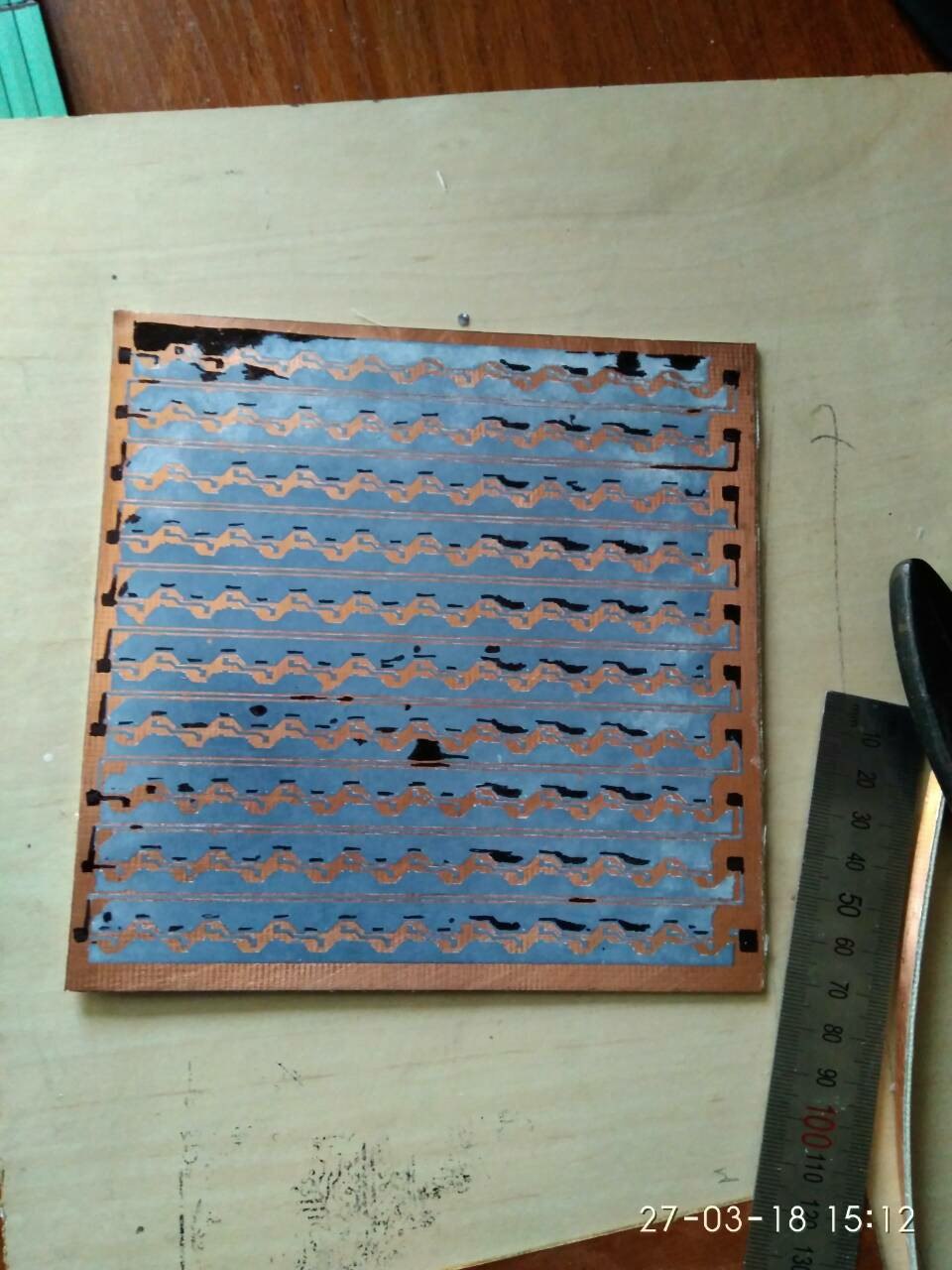


Для проверки корректной работы выбранной библиотеки было изготовлено тестовое устройство (линия светодиодов).



Было начато изготовление плат:







|  |  |  |
| --- | --- | --- |
| 4 неделя | 26.03 - 01.04 | Разработка ПО. Тестирование в Proteus |

Для будущего тестирования изготовленных плат был написан тестовый скетч, который зажигает по очереди каждый светодиод в цепи. Код предложен в приложении.

Также была разработана модель из 100 светодиодов в Proteus для проверки скетчей. Проект на github.

|  |  |  |
| --- | --- | --- |
| 5 неделя | 02.04 - 08.04 | Разработка корпуса модулей. Переход к пайке |

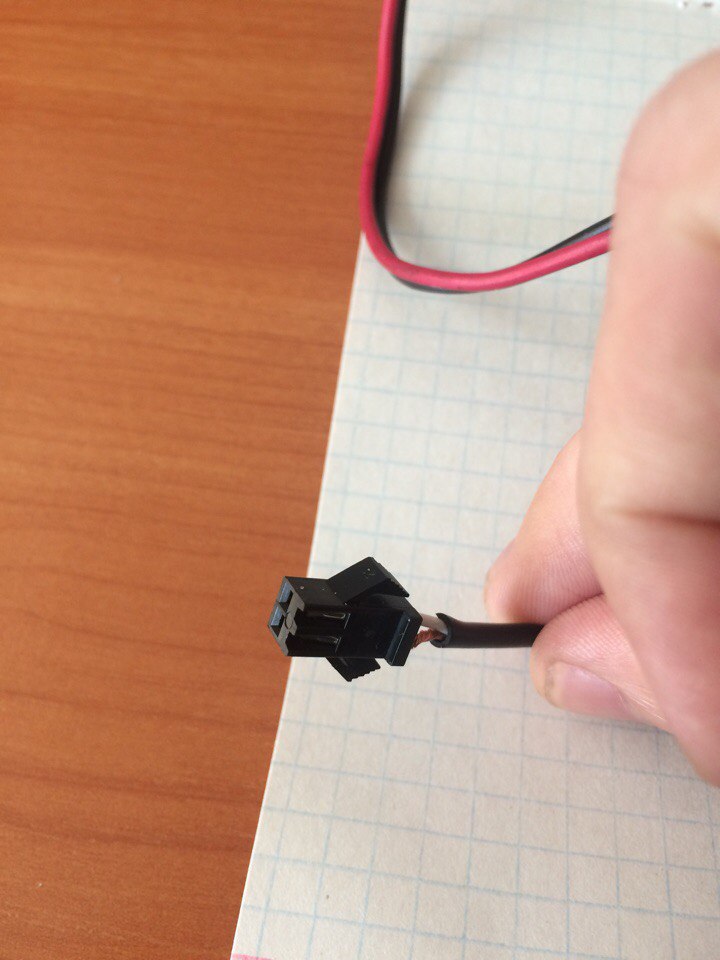
Для корпуса было выбрано матовое оргстекло для передней стенки, в качестве стенок боковых - вспененная пластмасса, а в качестве задней - тектстолит . Размер одного модуля 16Х16 см.

Была начата пайка плат для модулей:



|  |  |  |
| --- | --- | --- |
| 6 неделя | 09.04 - 15.04 | Пайка. Изготовление корпусов. |

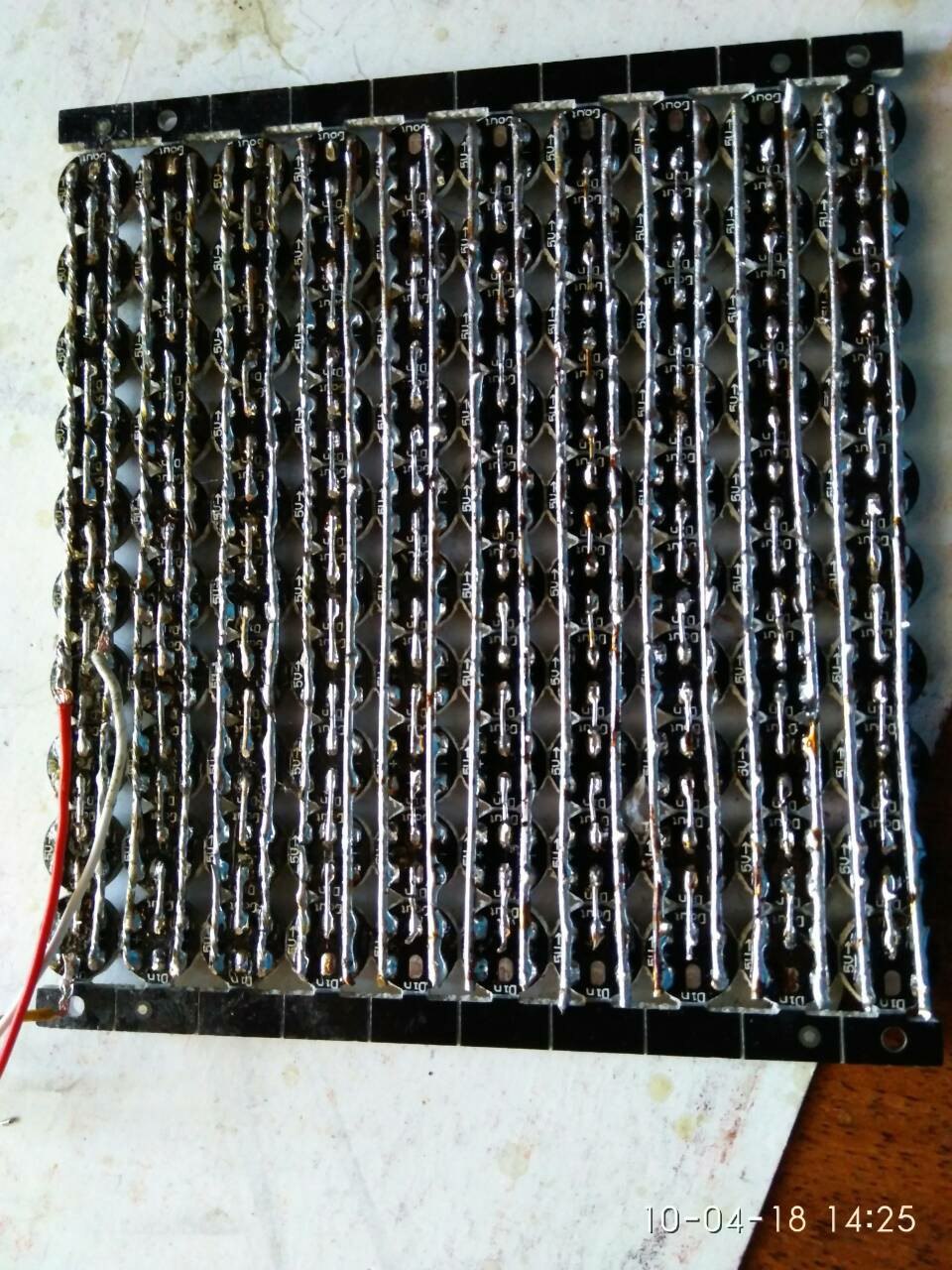
Для удобного соединения нескольких модулей было принято решение использовать фишки для подключения:

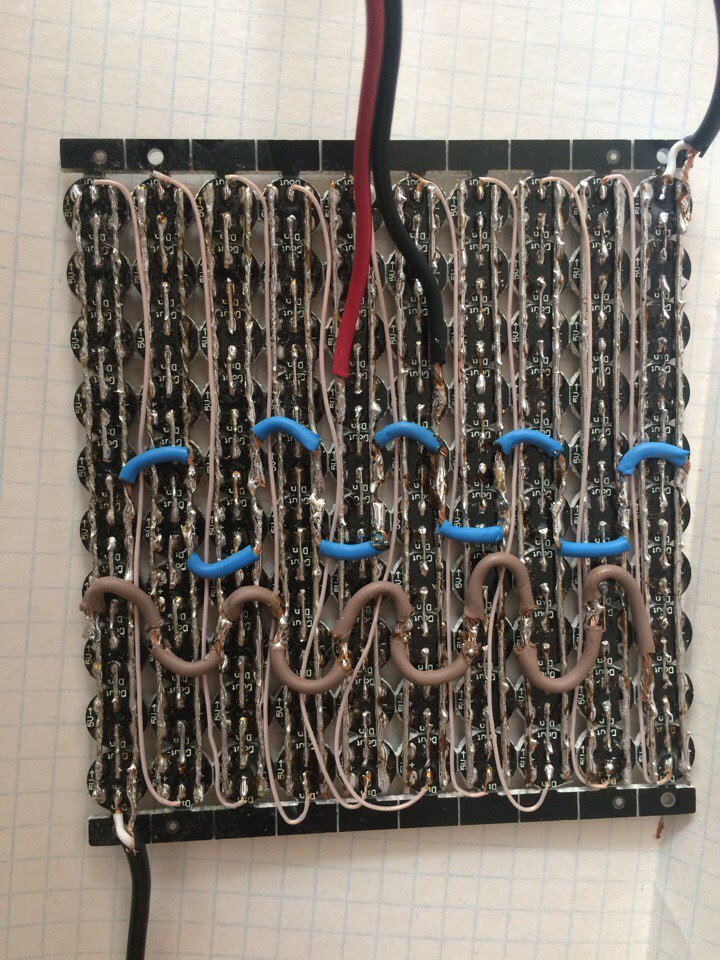


Питание 3-ех модулей будет не под силу Arduino Uno R3. Поэтому для решения вопроса с питанием было решено использовать блок питания ПК.

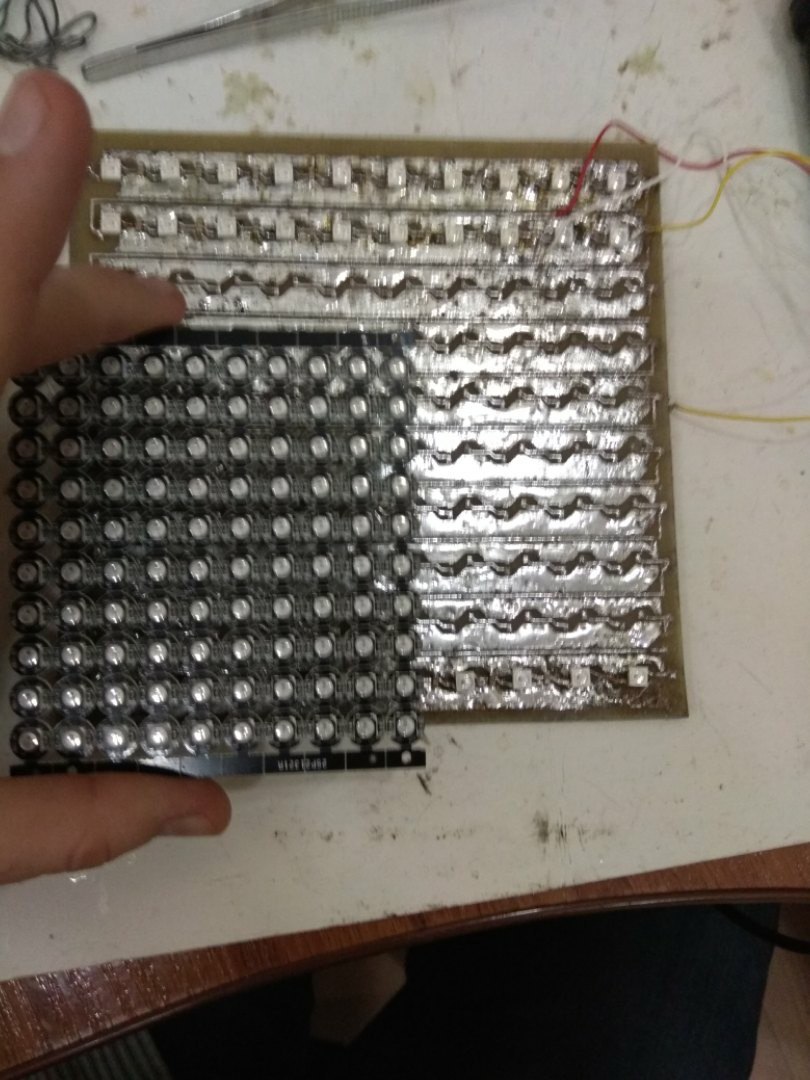
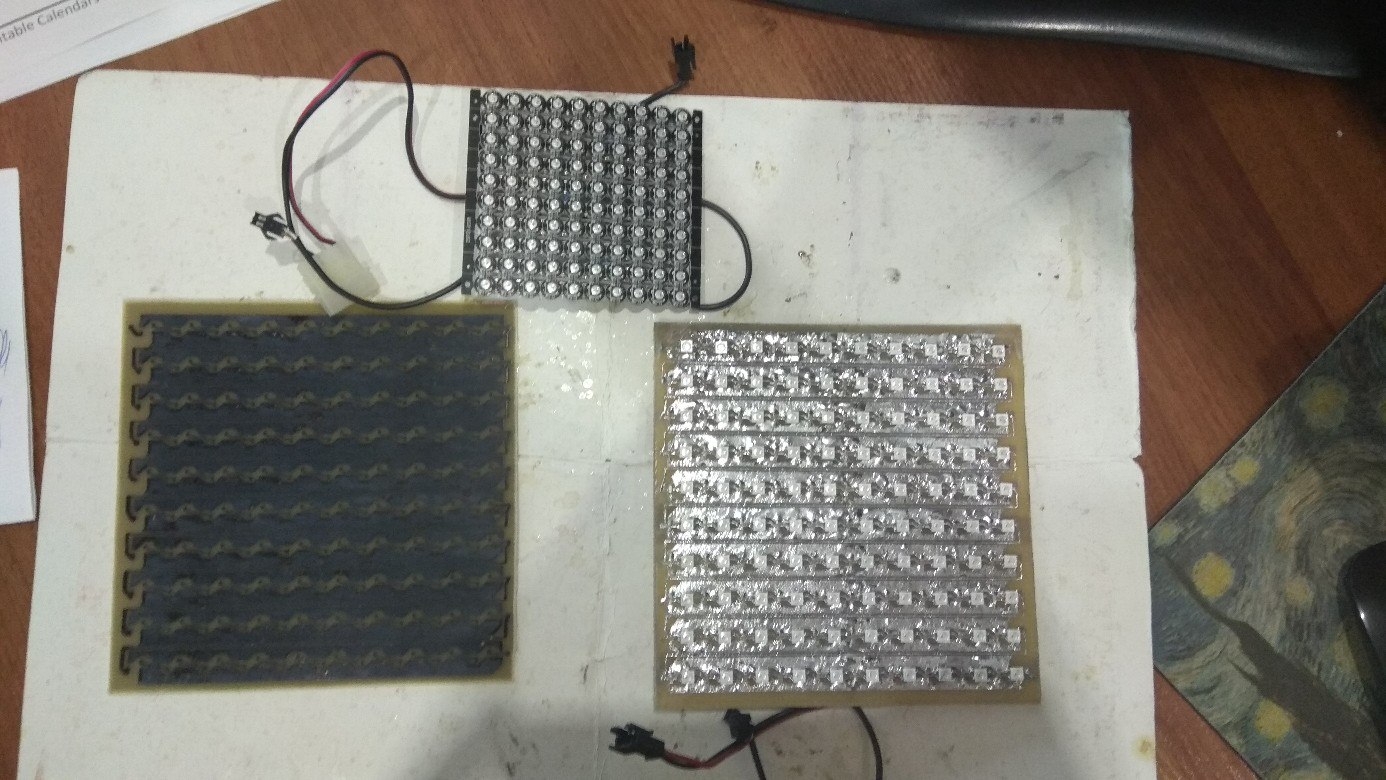


Когда пришли светодиоды с AliExpress, то их решено было спаять в одну плату для предварительного тестирования финального теста.



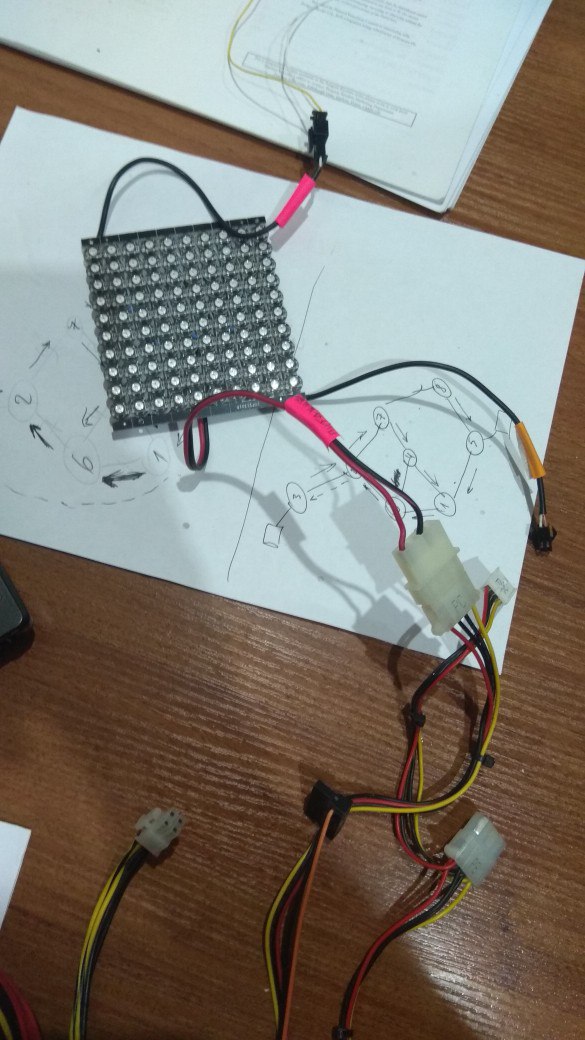


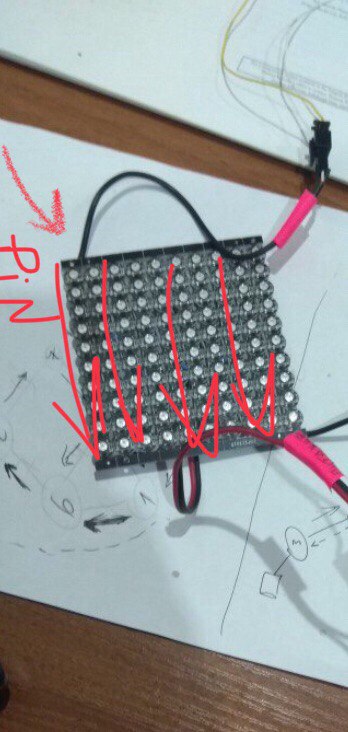
Сравнение получившейся платы с обычной платой:



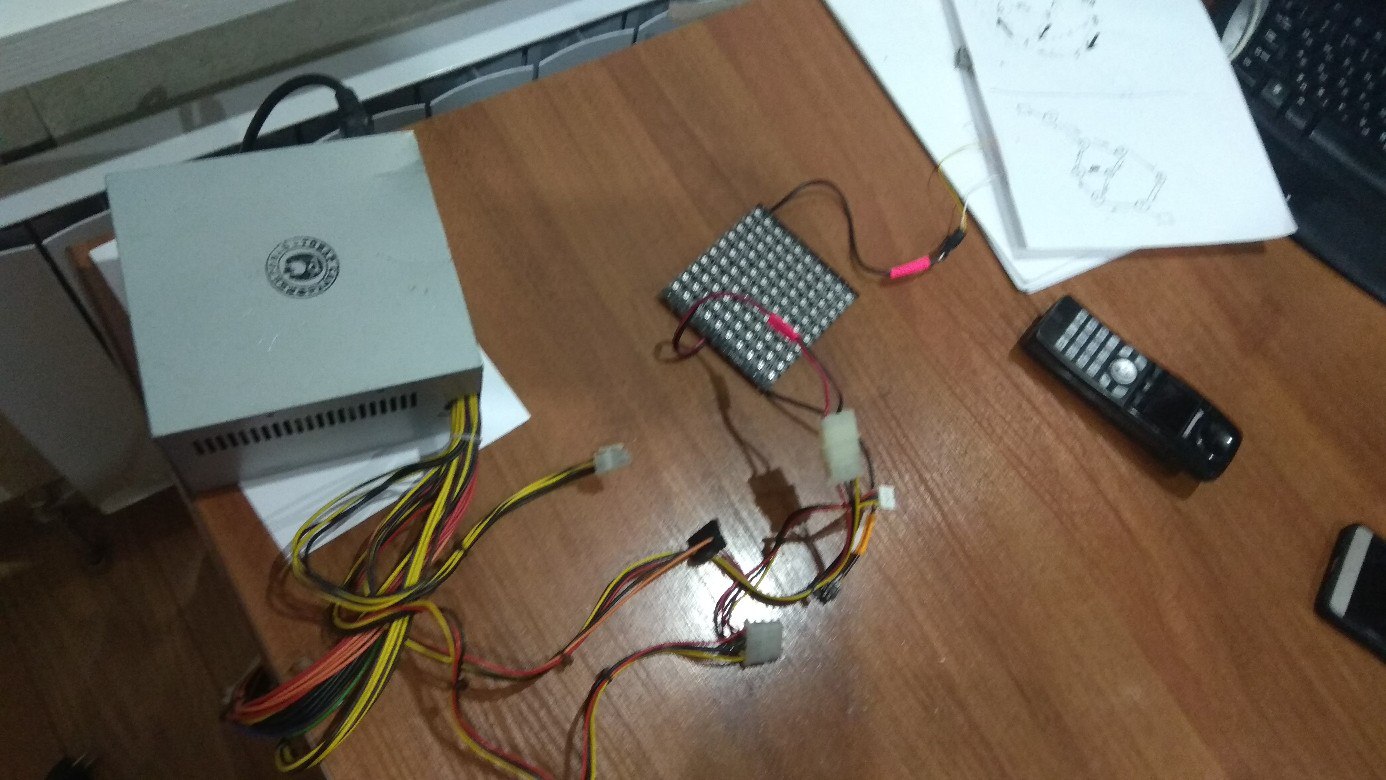
|  |  |  |
| --- | --- | --- |
| 7 неделя | 16.04 - 22.04 | Программация и тестирование устройства. |

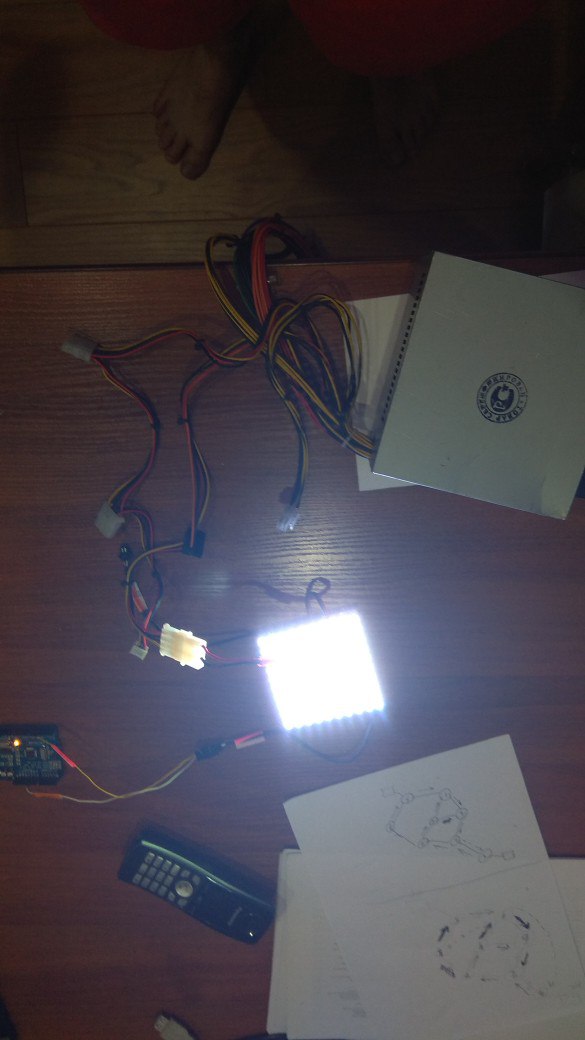
При запуске тестового проекта было выявлено, что, хоть и при правильном печатании платы, была допущена ошибка при пайке матрицы с aliexpress. Получившаяся плата была зеркально отражена предполагаемой:





То есть первый светодиод в цепи находился не в верхнем левом углу, как предполагалось, а в нижнем левом. После исправления данной ошибки можно было приступить к тестированию платы и написанию финального скетча.

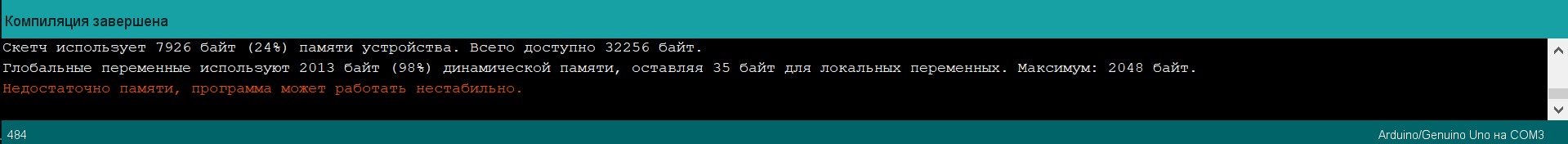




При проектировании управляющего скетча было решено выделить дополнительно в памяти массив 10Х10 для использования его как буфера. В него загружалась необходимая информация о состоянии светодиодов, после чего все состояния сдвигались.

Репозиторий решения на Github.com:   
https://github.com/NikitaKozelko/ModuleLedPanel

При написании алфавита в скетч и его запуске было показано, что памяти на плате не хватит для хранения всего алфавита.



Платформа оснащена 32 кБ flash-памяти, 2 кБ из которых отведено под, так называемый, bootloader. Он позволяет прошивать Arduino с обычного компьютера через USB. Эта память постоянна и не предназначена для изменения по ходу работы устройства. Её предназначение — хранение программы и сопутствующих статичных ресурсов. Также имеется 2 кБ SRAM-памяти, которые используются для хранения временных данных вроде переменных программы. По сути, это оперативная память платформы. SRAM-память очищается при обесточивании. Ещё имеется 1 кБ EEPROM-памяти для долговременного хранения данных. По своему назначению это аналог жёсткого диска для Arduino.

В итоге, от хранения каждой буквы в таком виде (в пример приведена буква а)

{0, 1, 0, 0, 0}, //a

{1, 1, 1, 0, 0},

{1, 0, 1, 0, 0},

{1, 0, 1, 0, 0},

{1, 0, 1, 0, 0},

{1, 0, 1, 0, 0},

{1, 1, 1, 0, 0},

{1, 0, 1, 0, 0},

{1, 0, 1, 0, 0},

{1, 0, 1, 0, 0},

Мы перешли к хранению букв в таком формате (в пример приведена буква а)

8, //{0, 1, 0, 0, 0}, //a

28, //{1, 1, 1, 0, 0},

20, //{1, 0, 1, 0, 0},

20, //{1, 0, 1, 0, 0},

20, //{1, 0, 1, 0, 0},

20, //{1, 0, 1, 0, 0},

28, //{1, 1, 1, 0, 0},

20, //{1, 0, 1, 0, 0},

20, //{1, 0, 1, 0, 0},

20, //{1, 0, 1, 0, 0},

Кардинального изменения не произошло. Если раньше мы хранили состояние светодиодов для каждой буквы, то сейчас мы просто изменили формат хранения. Мы представили состояние линии светодиодов как число в бинарном формате, а, значит, для его хранения мы могли бы перевести его в десятичный вид. Что мы и сделали.

Также в скетч были сохранены цифры и некоторые знаки пунктуации(, . ? !)

Для команды API был выбран следующий формат:

1-4 команда, 5-7 красный, 8-10 зеленый, 11-13 синий, 14-15 режим, 16-... текст.

Если была выбрана угловая команда UGLV, то вместо текста передавался поток состояния светодиодов для этого режима, потому что алгоритм сдвига в угловом режиме не определен.

Формат команды:

* команда:
  + TEXT для вывода текста
  + UGLV для вывода изображения в угловом режиме

- цвет:

- каждый цвет задается в формате от 0 до 255

- режим:

- определяет направление сдвига:

- 1: сдвигает слева направо в текстовом режиме

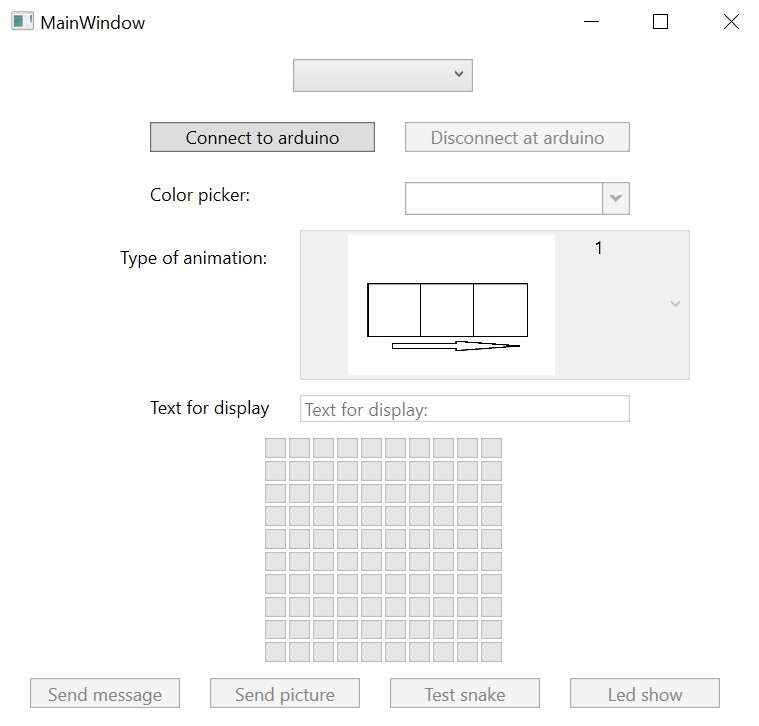
- 2: сдвигает справа налево в текстовом режиме

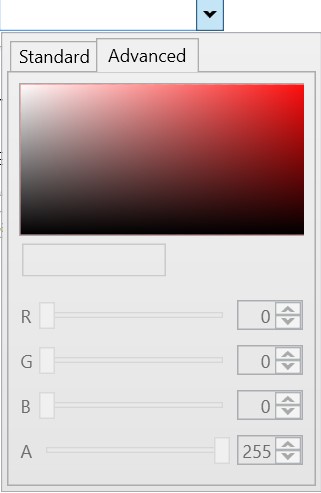
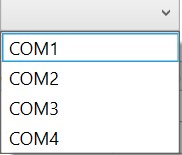
- 3: сдвигает изображение слева направо в угловом режиме

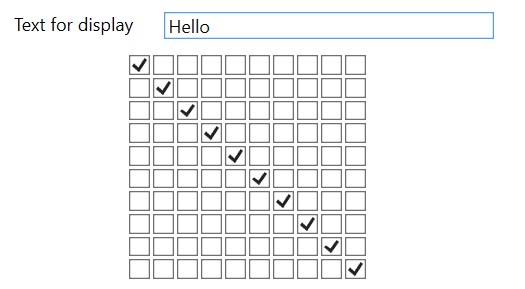
- 4: сдвигает изображение справа налево в угловом режиме

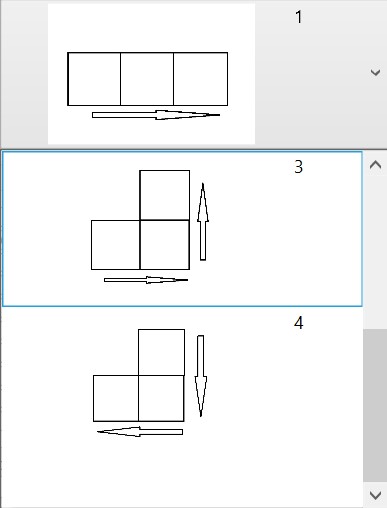
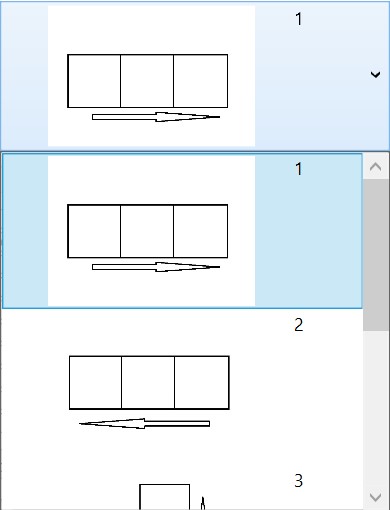
При декодировании изображения возникли проблемы. При декодировании строки длиной большей 100 символов крайняя часть строки удалялась при сохранении. Поэтому было решено передавать данные состояния строки светодиодов в двоичном виде.

При написании desktop-приложения на WPF для выбора цвета отображения информации был выбран color picker из wpf toolkit(расширение базовых элементов для WPF).









После написания скетча все платы были готовы. Значит, можно было приступить в полной проверке всего скетча.



|  |  |  |
| --- | --- | --- |
| 8 неделя | 23.04 - 29.04 | Устранение возникших проблем/ошибок. Тестирование устройства. Написание пояснительной записки. |

При тестирование было выявлено неправильное отображение передаваемых цветов. Этому послужила невнимательность при пайке светодиодов. Исправлено путем добавления одного конденсатора и допаивания ножек светодиодов.

|  |  |  |
| --- | --- | --- |
| 9 неделя | 30.04-06.05 | Оформление пояснительной записки. |

**Приложение:**

**Тестовый скетч для проверки модулей:**

**#include <Adafruit\_NeoPixel.h>**

**#define PIN 6 // номер порта, к которому подключен модуль**

**#define count\_led 200 // количество светодиодов**

**Adafruit\_NeoPixel pixels = Adafruit\_NeoPixel(count\_led, PIN, NEO\_GRB + NEO\_KHZ800); //first number change does distance between colors**

**void setup() {**

**pixels.begin();**

**for (int i=0; i<200; i++)**

**{**

**pixels.setPixelColor(i, pixels.Color(0,0,0));**

**}**

**pixels.show(); // Устанавливаем все светодиоды в состояние "Выключено"**

**}**

**void loop() {**

**for (int i=0; i<200; i++)**

**{**

**if (i != 0) pixels.setPixelColor(i-1, pixels.Color(0,0,0));**

**pixels.setPixelColor(i, pixels.Color(0,0,250));**

**pixels.show();**

**delay(50);**

**}**

**pixels.show();**

**}**