

Технологическая карта занятия/воспитательного мероприятия
по теме «Программирование адресных светодиодов с использованием библиотеки NeoPixel»
тема занятия ДО / воспитательного мероприятия

Автор-составитель: Савчук Алексей Михайлович
педагог дополнительного образования
ГУ ДО «Пермский краевой центр «Муравейник»
г. Пермь

Место проведения: лаборатория электроники и автоматики (каб.2, Пушкина,76)

Образовательная программа: Основы электроники.

Возраст обучающихся, год обучения: 12-14 лет.

Продолжительность: 3 ак. Часа.

Цель – знакомство с библиотекой Adafruit NeoPixel для управления адресными светодиодами.

Задачи:

- Рассмотреть принцип работы адресного светодиода WS2812b и схематическое подключение в составе сборок: лента, матрица, кольцо.
- Изучить библиотеку Adafruit NeoPixel, освоить приёмы использования основных её функций.
- Применить на практике полученные знания по работе с библиотекой путём её прямого использования.

Планируемые результаты:

- Освоение навыков программирования микроконтроллеров на языке программирования C++
- Умение читать и проектировать электронные схемы.
- Умение работать с информационной литературой и справочным материалом: характеристики, максимальные и минимальные параметры работоспособности, назначения выводов и прочая спецификация.
- Самостоятельная сборка и проверка электронных схем и программирование.

Форма проведения занятия/воспитательного мероприятия: лекция (1ч.), практическое занятие (2 ч.)

Педагогические технологии: технология развивающего обучения.

Методы:

теоретические: • беседа; • инструктаж; • демонстрация; • словесные;

наглядные: • демонстрация плакатов, схем, таблиц, фото; • использование технических средств;

практические: • практические задания;

по степени активности познавательной деятельности учащихся: • объяснительный; • иллюстративный; • проблемный.

Приёмы: практическое задание в форме кейса.

Оборудование:

При очном формате проведения занятия:

- Платформы на базе микроконтроллеров Atmel (по типу Arduino, Digipark или ESP8266) с USB кабелем
- Сборка из адресных светодиодов Ws2812b (матрица, кольцо, лента)
- Компьютер с доступным USB портом и предустановленной программной средой Arduino IDE.

При дистанционном формате проведения занятия:

- Компьютер с доступом в интернет

Предварительная подготовка:

- Подготовка демонстрационного материала в виде презентации для сопровождения лекции.
- При очном проведении: Проверка наличия и работоспособности электронных компонентов, установка среды Arduino IDE, проверка доступа в интернет.
- При дистанционном проведении: Проверка наличия аккаунта для входа в среду Tinkercad для каждого обучающегося. Выдача кода класса для регистрации участников.

Информационные источники:

1. Библиотека Adafruit NeoPixel https://github.com/adafruit/Adafruit_NeoPixel
2. Подключаем модули NeoPixel на базе WS2812B светодиодов к Arduino <https://lesson.iarduino.ru/page/podklyuchaem-moduli-neopixel-na-baze-ws2812b-svetodiiodov-k-arduino/>
3. Сайт Александра Климова "Освой Arduino играючи" Библиотека Adafruit NeoPixel https://developer.alexanderklimov.ru/arduino/libraries/adafruit_neopixel.php

Краткая аннотация к занятию/воспитательному мероприятию

Текущее занятие является частью проектной деятельности обучающихся 2 года обучения по ДОП «Электроника и автоматика». Первая часть занятия лекцию с использованием презентации. Рассматриваемые вопросы: подключение библиотеки, подготовка работы с ней и примеры использования. Вторая часть содержит индивидуальное практическое задание по работе с текущей библиотекой. Практическим итогом работы будут электронные поделки: святающийся смайлик, ночник-игрушка в форме звездочки, сердца, котёнка, космонавта и др.

Содержание взаимодействия педагога и обучающихся

№	Этап занятия / воспитательного мероприятия, продолжительность, мин.	Деятельность педагога (с обозначением используемых технологий, методик, приёмов; применяемого технического оснащения)	Деятельность обучающихся (с обозначением видов и форм активности)	Планируемый результат, комментарий
1	Организационный (5 мин)	Приветствие обучающихся, оповещение обучающихся о начале лекции. Подготовка мультимедийного оборудования. Проверка количества обучающихся, выяснение отсутствия обучающихся (если имеются), отметка обучающихся в журнале.	Приветствие, готовность к занятию собственных рабочих мест (тетрадей, ручек и пр.)	Включение в занятие
2	Теоретический этап (40 мин)	Проведение лекции по теме занятия. Демонстрация презентации по текущей теме. Ответы на возникающие в процессе лекции вопросы.	Конспектирование лекции. Участие в обсуждении.	Конспект лекционного материала. Ознакомление обучающихся с новой библиотекой Adafruit NeoPixel. Новые знания для обеспечения связи микроконтроллера и адресных светодиодов.
3	Промежуточный этап (10 мин)	<ol style="list-style-type: none"> 1. Проведение гимнастических упражнений в кабинете для сохранения здоровья обучающихся. 2. Проветривание кабинета без детей. 	<ol style="list-style-type: none"> 1. Гимнастика для пальцев рук. Упражнения для глаз по типу «влево-вправо, вверх-вниз». 2. Прогулка по коридору без телефонов, живое общение 	Снятие напряжения и утомления с обучающихся, подготовка кабинета к практическому этапу.
4	Практический этап (75 мин)	Выдача практического задания, помощь в выполнении, контроль за выполнением задания и оставшимся временем до конца занятия.	Выполнение и сдача практического задания. Задание для конкретного обучающегося определяется индивидуально, и может быть либо с использованием микроконтроллера и	Практическая работа в виде цвето-световой иллюминации.

			светодиодной сборки, либо эти же элементы используются в виртуальной среде TinkerCad. Ответы на вопросы.	
5	Заключительный этап (10 мин)	Контроль и оценивание выполненного задания.	Представление выполненного задания. Практическим итогом работы будут такие электронные поделки как светящийся смайлик, ночник-игрушка в форме звёздочки, сердца, котёнка, космонавта и др. Уборка рабочих мест.	Подведение итогов занятия.
6	Рефлексия (5 мин)	Опрос для самооценивания. Поощрение прогрессивных обучающихся и рекомендации для тех, кто не успел или не справился с заданием.	Ответы на вопросы педагога. Самооценка достигнутых результатов.	Самооценка обучающимися проделанной работы.

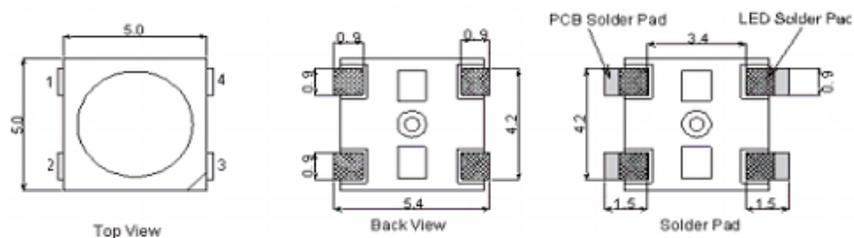
Datasheet (информационный листок об элементе, спецификация) для адресного светодиода



WS2812B

Intelligent control LED
integrated light source

Mechanical Dimensions



PIN function

NO.	Symbol	Function description
1	VDD	Power supply LED
2	DOUT	Control data signal output
3	VSS	Ground
4	DIN	Control data signal input

Absolute Maximum Ratings

Parameter	Symbol	Ratings	Unit
Power supply voltage	V_{DD}	+3.5~+5.3	V
Input voltage	V_I	-0.5~VDD+0.5	V
Operation junction temperature	T_{opt}	-25~+80	°C
Storage temperature range	T_{stg}	-40~+105	°C

Пример выполненной работы



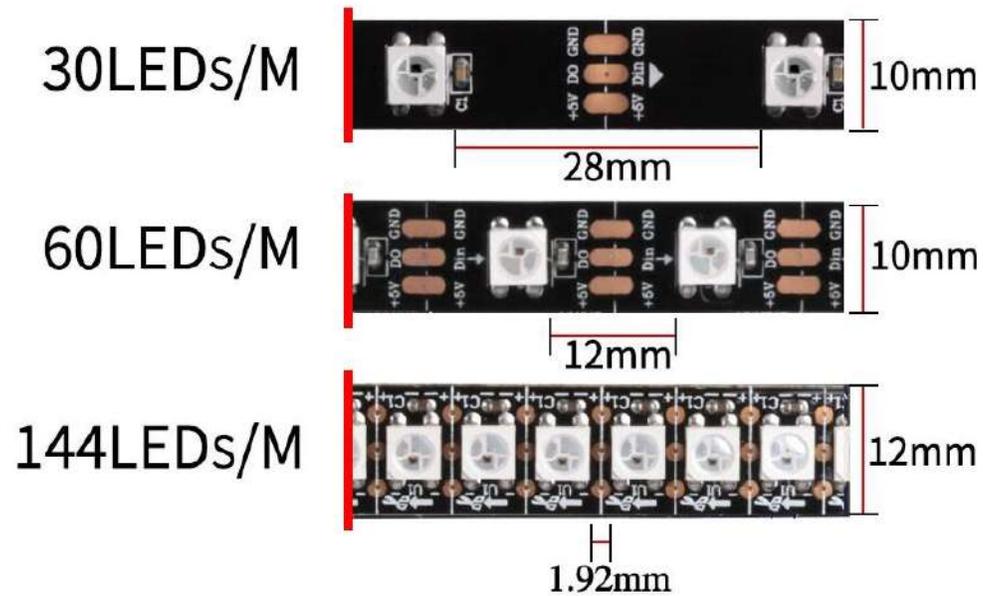
NeoPixel



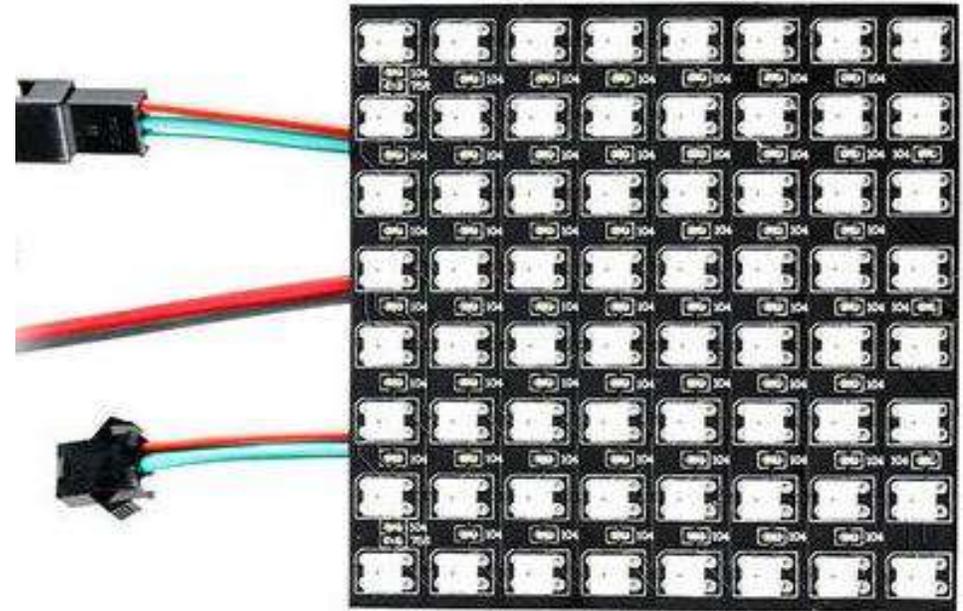
Знакомство с адресным светодиодом.
Работа с библиотекой для управления
адресных светодиодов WS2812b

Варианты готового исполнения (заводского)

лента



матрица

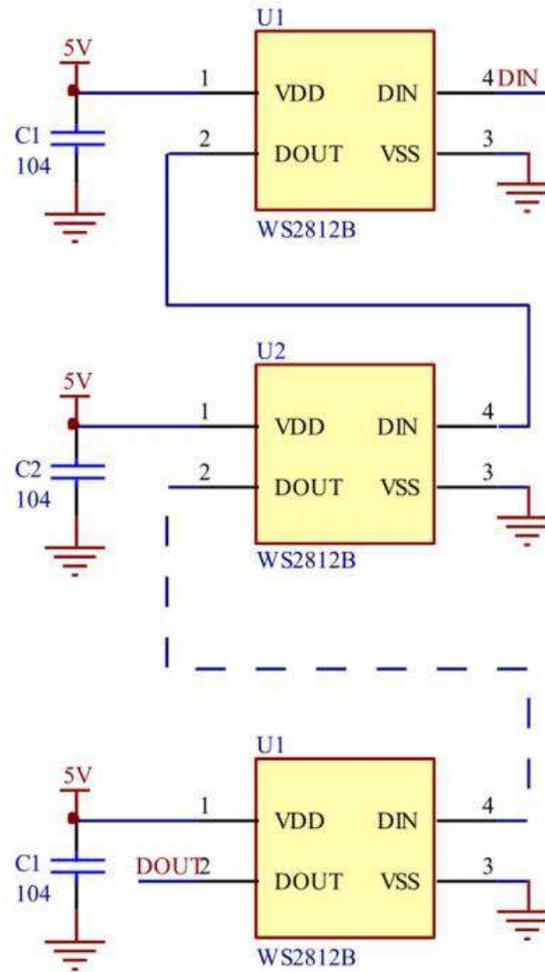


Назначение выводов светодиода



- VCC – положительный полюс питания светодиодом
- GND – отрицательный полюс питания светодиодом
- DIN (Data In) – информационный вход для подачи сигнала
- DOUT(Data Out) – информационный выход для передачи сигнала на следующий светодиод

Схема подключения светодиодов



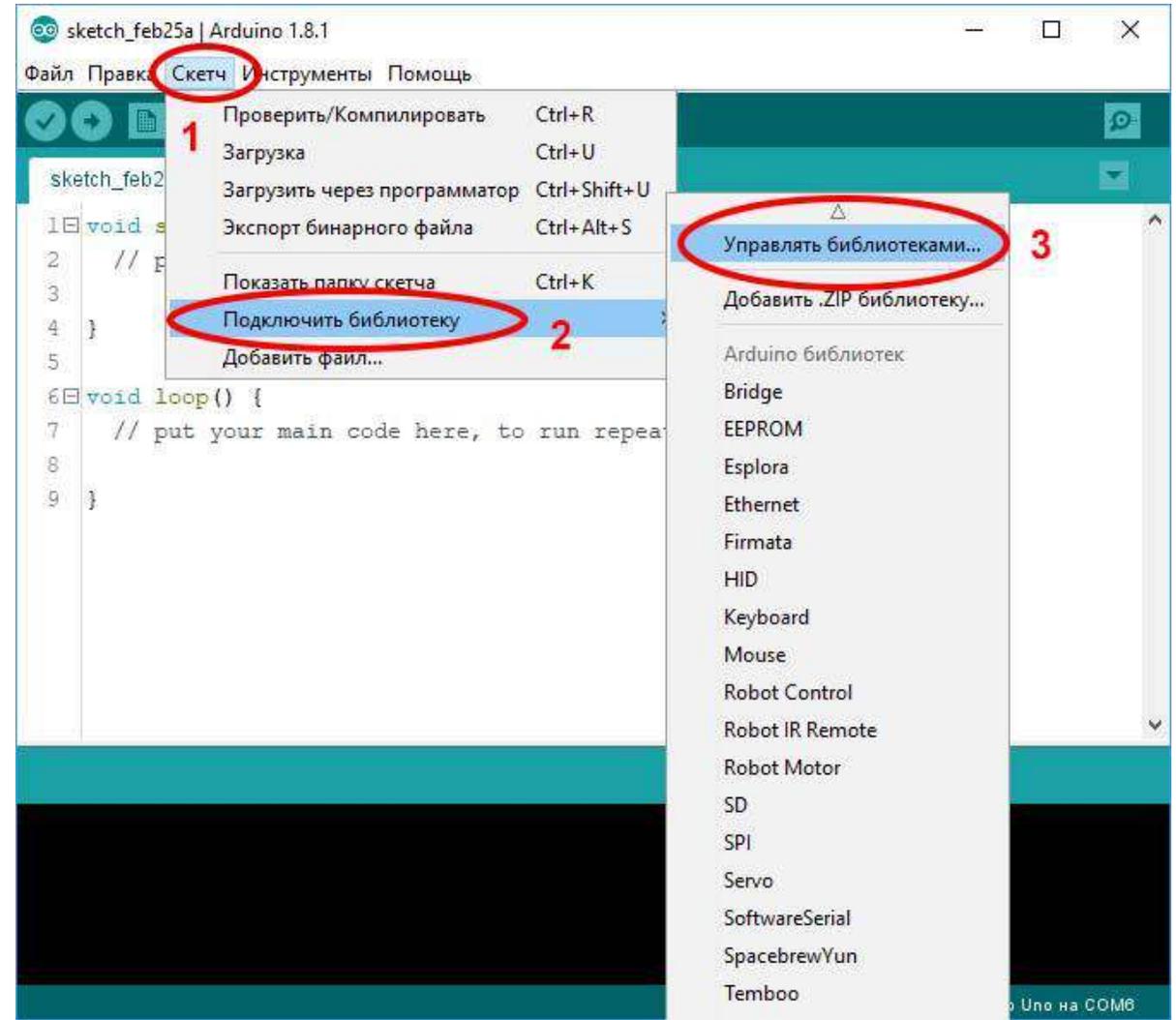
DOUT последнего светодиода остаётся неподключенным

Библиотека Adafruit NeoPixel

Добавить библиотеку Adafruit NeoPixel в Arduino IDE можно двумя способами:

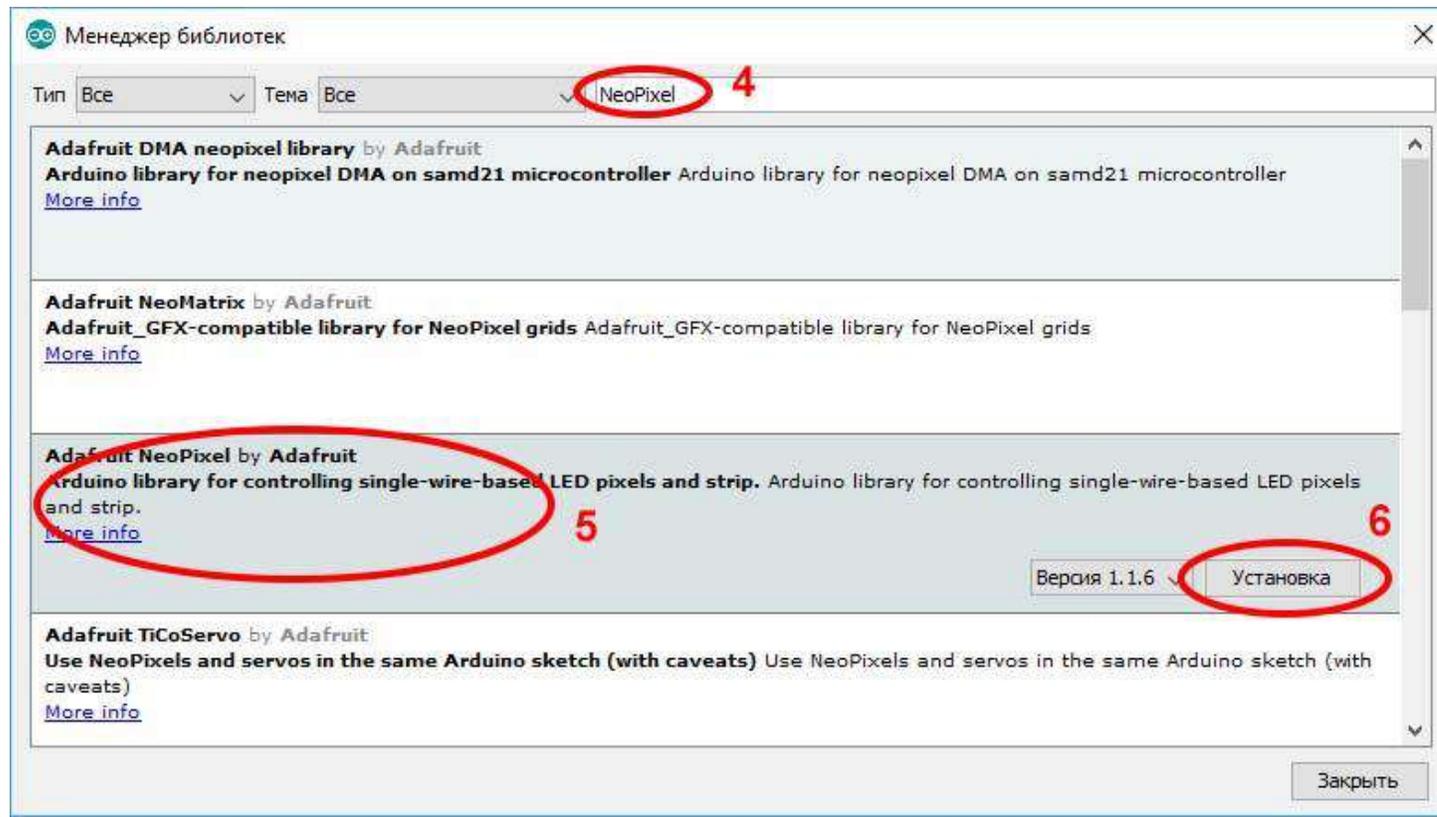
1. Через «Менеджер библиотек»

Для этого в среде Arduino IDE открываем «Менеджер библиотек» перейдя по следующим вкладкам меню: «Скетч» – «Подключить библиотеку» – «Управлять библиотеками...»



Библиотека Adafruit NeoPixel

В открывшемся окне «Менеджер библиотек» в поле «Отфильтровать результаты поиска...» вписать название библиотеки «NeoPixel». В получившемся списке выбрать «Adafruit NeoPixel by Adafruit» и нажать на кнопку «Установка».



Библиотека Adafruit NeoPixel

2. Самостоятельная установка

- Для самостоятельной установки библиотеки, необходимо перейти на страницу библиотеки в GitHub.

https://github.com/adafruit/Adafruit_NeoPixel

- В правом углу нажать на кнопку «Clone or download» - «Download ZIP».
- После скачивания библиотеки, запустить Arduino IDE и перейти по следующим меню: «Скетч» – «Подключить библиотеку» - «Добавить .ZIP библиотеку...», выбрать скачанный архив и нажать на кнопку «Открыть».

Работа с библиотекой Adafruit NeoPixel.

adafruit/ **Adafruit_NeoPixel**



Arduino library for controlling single-wire LED pixels
(NeoPixel, WS2812, etc.)

 67

Contributors

 57

Issues

 3k

Stars

 1k

Forks



Подключение

Для того чтобы подключить библиотеку к своему скетчу необходимо в самом его начале кода прописать следующее:

```
#include "Adafruit_NeoPixel.h"
```

Эта строчка подключит библиотеку и даст возможность использовать ее в своем проекте.

Создание основного объекта библиотеки

Для создания объекта прописываем следующее:

```
Adafruit_NeoPixel pixel = Adafruit_NeoPixel(5, 6, NEO_GRB + NEO_KHZ800);
```

Где:

pixel – имя объекта библиотеки (их можно создать несколько с разным именем, учитывая количество управляемых контуров схемы)

Параметр 1 - Количество пикселей в ленте.

Параметр 2 - Порт, к которому подключена лента.

Параметр 3 - Дополнительные флаги (по умолчанию не изменяются)

Инициализация (подготовка к готовности)

Для инициализации наших светодиодов и дальнейшей работы с ними, необходимо в секции `setup()` вызвать две команды ранее созданного объекта ленты.

```
void setup()
```

```
{
```

```
pixel.begin(); // эта функция настроит порт Arduino и выставит значения по умолчанию.
```

```
pixel.show(); // эта функция передаст команду на выключение всем светодиодам, так как не были заданы цвета.
```

```
}
```

Управление. Установка цвета.

- Для того чтобы задать любому пикселю определенный цвет, существует функция `setPixelColor()`

Первый вариант использования функции:

```
pixel.setPixelColor(n, red, green, blue);
```

Где:

- `n` – номер светодиода с сборке (счёт светодиодов ведётся от «0»)
- `red` - число от 0 до 255 определяющее яркость красного светодиода в пикселе.
- `green` - число от 0 до 255 определяющее яркость зеленого светодиода в пикселе.
- `blue` - число от 0 до 255 определяющее яркость синего светодиода в пикселе.

К примеру, чтобы зажечь 2-й светодиод белым светом средней яркости в сборке нужно прописать `pixel.setPixelColor(1, 127, 127, 127)` или `pixel.setPixelColor(1, 0, 0, 0)` чтобы вовсе его отключить.

Управление. Установка цвета.

- Для того чтобы задать любому пикселю определенный цвет, существует функция `setPixelColor()`

Первый вариант использования функции:

```
pixel.setPixelColor(n, red, green, blue);
```

Где:

- `n` – номер светодиода с сборке (счёт светодиодов ведётся от «0»)
- `red` - число от 0 до 255 определяющее яркость красного светодиода в пикселе.
- `green` - число от 0 до 255 определяющее яркость зеленого светодиода в пикселе.
- `blue` - число от 0 до 255 определяющее яркость синего светодиода в пикселе.



К примеру, чтобы зажечь 2-й светодиод белым светом средней яркости в сборке нужно прописать `pixel.setPixelColor(1, 127, 127, 127)` или `pixel.setPixelColor(1, 0, 0, 0)` чтобы вовсе его отключить.

Управление. Альтернативная установка цвета.

В качестве переменной цвета можно использовать функцию `pixels.Color()`.

Пример:

```
uint32_t magenta = strip.Color(255, 0, 255); // Пурпурный.
```

```
Или uint32_t magenta = strip.Color(0xFF00FF); // Пурпурный.
```

```
pixels.setPixelColor(0, magenta)); // Назначаем для первого светодиода цвет  
“пурпурный»
```

```
0x00FF00
```

```
//Для справки:
```

```
uint16_t, unsigned int – тип беззнаковой 16-битной переменной
```

```
uint32_t, unsigned short – тип беззнаковой 32-битной переменной
```

Управление. Альтернативная установка цвета.

Ещё один вариант заполнения объекта цветом с использованием функции `pixels.fill()`.

Пример:

```
pixels.fill(c, i, n).
```

Заполнить `n` количество пикселей цветом `c` начиная со светодиода `i`.

Управление. Установка яркости.

Для того чтобы установить яркость, существует функция `pixel.setBrightness(n);`

Где:

- `n` – яркость светодиодов от 0 до 255

Пример: `strip.setBrightness(64);` //задается 1/4 от общей яркости

Нулевая яркость отключает светодиоды. При использовании максимальной яркости учитывайте потребление тока всех светодиодов и максимальную силу тока источника питания.

Управление. Вывод на светодиоды.

- Для того чтобы передать данные на светодиоды используется функция `pixel.show()`;

Эта функция заставляет светиться светодиоды. Функцию необходимо вызывать каждый раз после того, как вы изменили цвета или настройки ленты.

Управление. Гашение светодиодов.

- Для того чтобы «потушить» светодиоды используется функция `pixel.clear()`; с последующим вызовом `pixel.show()`;

Управление. Редкоиспользуемые функции.

- Функция `pixel.getPixelColor()` возвращает цвет пикселя в виде 32-битного числа. В параметре `n` задается номер пикселя, цвет которого необходимо вернуть.

Пример: `unsigned int color = strip.getPixelColor(n);`

- Функция `pixel.numPixels()` возвращает количество пикселей в ленте. Используется, как правило, в циклах или при работе с несколькими светодиодными лентами.

Пример: `unsigned short n = strip.numPixels();`

Использование цикла `for` для управления

Поскольку библиотека требует установки цвета для каждого светодиода, целесообразно использовать цикл для заполнения одним цветом всех светодиодов.

```
for (int i = 0; i < numPixels ; i++) // numPixels – кол-во светодиодов в сборке
{
  pixel.setPixelColor(i, strip.Color(255, 0, 0)); // Красный цвет.
}
pixel.show(); // Передаем цвета ленте.
delay(500); // Ждем 500 мс.
```

Таким образом, в этом примере каждые полсекунды будет загораться по одному светодиоду красным цветом. При удалении задержки все светодиоды мгновенно получают свой цвет и при вызове `pixel.show()` загорятся красным.

При вызове этого цикла с нулевым цветом для каждого светодиода, можно их все выключить.

Практическое задание

Используя представленные в этой презентации функции, попробуйте поуправлять адресными светодиодами в составе ленты или матрицы.

В качестве дополнения, рекомендуется познакомиться с новыми интересными эффектами по ссылке:

<https://duino.ru/blog/beguschie-ogni-na-digispark/>