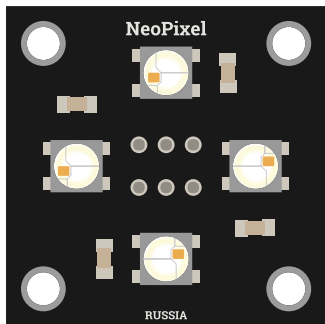


# Адресные светодиоды, модули NeoPixel



## Общие сведения:

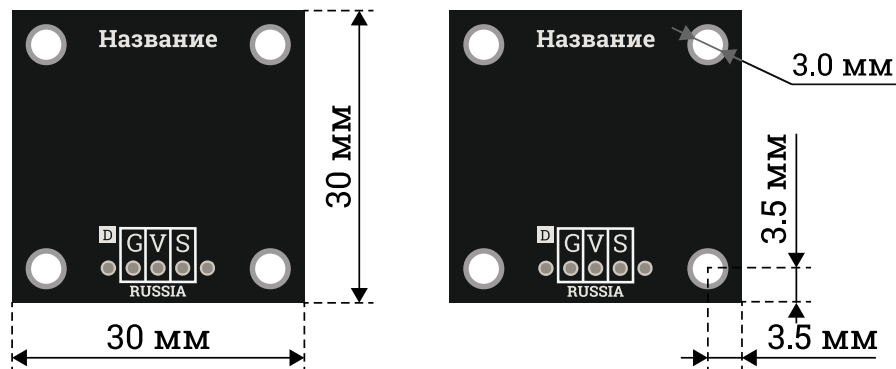
[Трех-модуль NeoPixel](#) - это 4 RGB-светодиода со встроенными драйверами в одном корпусе, установленные на одном модуле. Эти модули можно соединять в ленты NeoPixel, используя всего одну линию данных, а их «адресность» позволяет управлять цветом каждого светодиода на одной линии.

В этой статье речь пойдет о модулях NeoPixel, построенных на [адресных светодиодах WS2812B](#).

## Спецификация:

- Входное напряжение питания модуля: 5 В
- Ток потребляемый драйвером каждого светодиода: < 1 мкА
- Ток потребляемый каждым светодиодом: < 45 мА (по 15 мА на полную яркость свечения каждого цвета светодиода)
- Уровень логической «1» на входе IN: > 0,7 V<sub>cc</sub>
- Уровень логического «0» на входе IN: < 0,3 V<sub>cc</sub>
- Входная ёмкость: 15 пФ
- Задержка передачи данных: < 300 нс
- Время передачи одного бита данных: 1,25 мкс
- Длительность импульса для передачи бита «1»: 900 нс (±150 нс)
- Длительность импульса для передачи бита «0»: 350 нс (±150 нс)
- Время простоя линии данных перед применением новых цветов > 50 мкс
- Длина световой волны: 620 ... 630 нм (для красного цвета)
- Длина световой волны: 515 ... 530 нм (для зелёного цвета)
- Длина световой волны: 465 ... 470 нм (для синего цвета)
- Рабочая температура: -25 ... 80 °С
- Температура хранения: -55 ... 150 °С
- Габариты: 30x30x30 мм (с учётом колодки выводов)
- Вес: 9 г

Все модули линейки "Тема" выполнены в одном формате



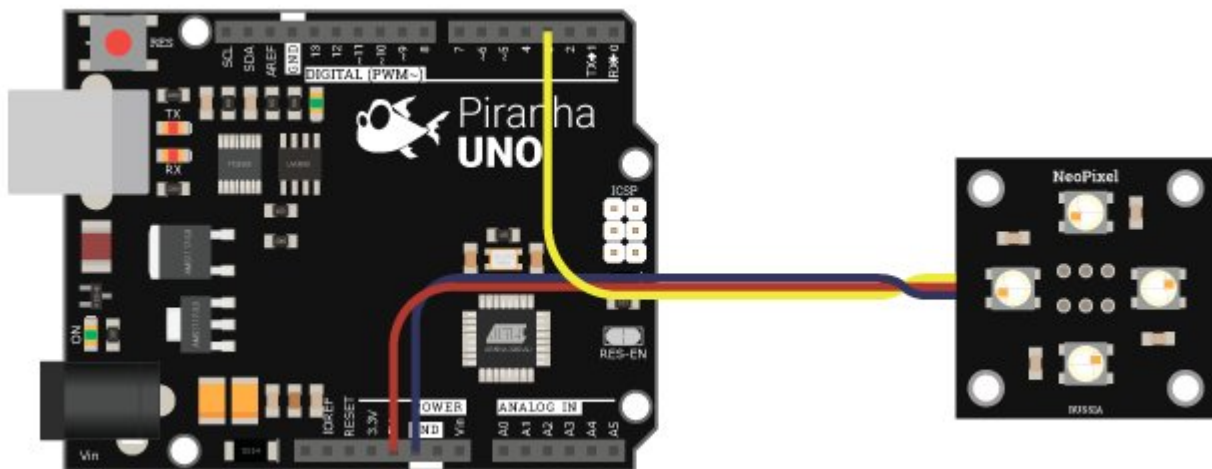
## Подключение:

Модули NeoPixel подключаются входом (IN) к любому выводу [Arduino](#).

Модуль удобно подключать 2 способами, в зависимости от ситуации:

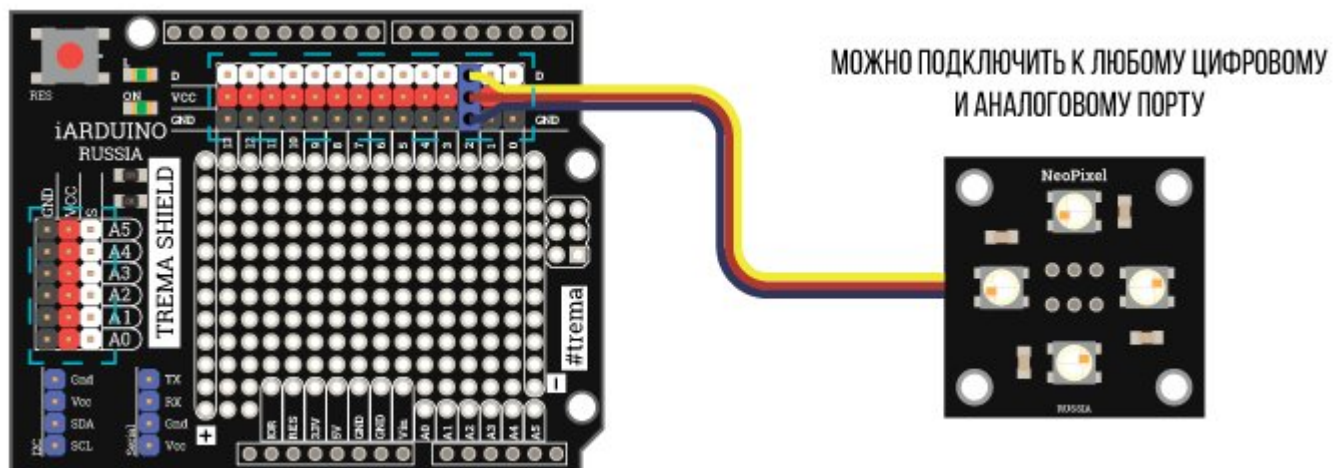
### Способ - 1 : Используя проводной шлейф и Piranha UNO

Используя провода «[Папа – Мама](#)», подключаем напрямую к контроллеру Piranha UNO.

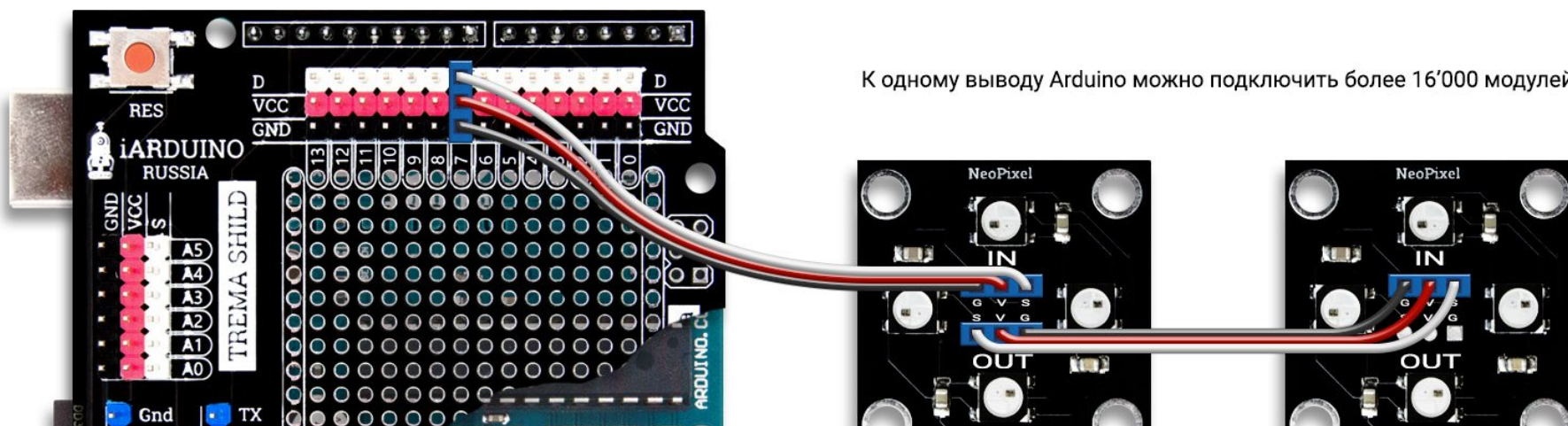


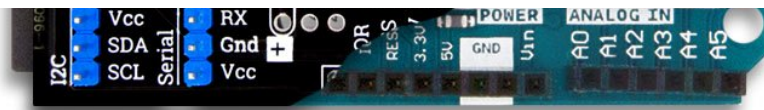
## Способ - 2 : Используя проводной шлейф и Shield

Используя 3-х проводной шлейф, к Trema Shield, Trema-Power Shield, Motor Shield, Trema Shield NANO и тд.



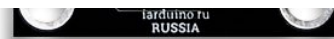
При подключении нескольких модулей их можно соединить друг с другом (выход OUT каждого модуля со входом IN следующего), а вход IN первого модуля к любому выводу [Arduino](#). Или подключить вход IN каждого модуля к своему выводу [Arduino](#).





Trema Shield

Arduino Uno



Трема-модуль  
NeoPixel



Трема-модуль  
NeoPixel

Нумерация (адрес) [светодиодов](#) является сквозной и начинается от ближайшего к выводу [Arduino](#). Например, если Вы подключили к одному выводу [Arduino](#) два модуля из 4 светодиода, то первый светодиод первого модуля будет иметь адрес 0, а первый светодиод второго модуля будет иметь адрес 4.

## Питание:

Входное напряжение 5 В постоянного тока, подаётся на выводы Vcc и GND модуля NeoPixel.

## Подробнее о модуле:

[Адресные светодиоды WS2812B](#) это три светодиода RGB (Red - красный, Green - зелёный и Blue - синий) и драйвер (микросхема) для управления этими светодиодами, в одном SMD (Surface Mounted Device - прибор монтируемый поверхностно) корпусе.

Корпус каждого светодиода имеет 4 вывода: два вывода данных (IN - вход и OUT - выход) и два вывода питания (Vcc и GND). В модулях NeoPixel выходы (OUT) предыдущих светодиодов соединены со входами (IN) следующих создавая цепочку светодиодов. Но и сами модули NeoPixel также имеют 4 вывода: два вывода данных (IN, OUT) и два вывода питания (Vcc, GND), и модули также можно соединять в цепочку. Таким образом к одному выводу [Arduino](#) можно подключить «неограниченное» количество модулей NeoPixel.

Цвет каждого [адресного светодиода](#) задаётся 3 байтами (каждый байт указывает яркость одного цвета: 1-G, 2-R, 3-B). После того как первый [адресный светодиод](#) получил на вход (IN) первые 3 байта, он сохраняет их в своей ОЗУ, а остальные байты пропускает со входа (IN) на выход (OUT). Второй [адресный светодиод](#) сохранит в своей ОЗУ следующие 3 байта а остальные пропустит и т.д. Если в линию данных перестали поступать биты (на линии установлен уровень логического «0» дольше 50 мкс), светодиоды установят свои цвета. Каждый бит данных имеет длительность 1,25 мкс и состоит из импульса и паузы. Бит «0» отличается от бита «1» длительностью импульса: • бит «0» - импульс 350 нс, пауза 900 нс; • бит «1» - импульс 900 нс, пауза 350 нс.

Для работы с модулем предлагаем воспользоваться библиотекой [iarduino\\_NeoPixel](#), которая позволяет работать с модулями NeoPixel если суммарное количество [адресных светодиодов WS2812B](#) не превышает 65534 шт.

Подробнее про установку библиотеки читайте в нашей [инструкции](#)..

## Примеры:

### Установка цвета для каждого из 4 светодиодов модуля NeoPixel:

```
#include <iarduino_NeoPixel.h> // Подключаем библиотеку iarduino_NeoPixel для работы со светодиодами Neo
iarduino_NeoPixel led(7,4); // Объявляем объект LED указывая (№ вывода Arduino к которому подключён м
void setup(){
    led.begin(); // Иницируем работу с модулем NeoPixel
}
void loop(){
    led.setColor(0, 255,0,0); // Устанавливаем красный цвет для 0 светодиода (0,R,G,B)
    led.setColor(1, 0,255,0); // Устанавливаем зелёный цвет для 2 светодиода (2,R,G,B)
    led.setColor(2, 0,0,255); // Устанавливаем синий цвет для 2 светодиода (2,R,G,B)
    led.setColor(3, 0xFFFF00); // Устанавливаем жёлтый цвет для 3 светодиода (3,RGB)
    led.write(); // Записываем
    delay(2000); // Ждём 2 секунды, чтоб увидеть четыре разных цвета
    led.setColor(NeoPixelAll, 255,255,255); // Устанавливаем белый цвет для всех светодиодов (все,RGB)
    led.write(); // Записываем
    delay(2000); // Ждём 2 секунды, чтоб увидеть белый цвет
}
```

### Перелив всех цветов радуги на модуле NeoPixel из 4 светодиодов:

```
#include <iarduino_NeoPixel.h> // Подключаем библиотеку iarduino_NeoPixel для работы со светодиодами Ne
iarduino_NeoPixel led(7,4); // Объявляем объект LED указывая (№ вывода Arduino к которому подключён
    uint8_t j; // Объявляем переменную для хранения значения сдвига спектра цветов для
    uint8_t k; // Объявляем переменную для хранения положения сдвига спектра цвета для
    uint8_t r, g, b; // Объявляем переменную для хранения цветов RGB для каждого светодиода
const uint8_t z=5; // Определяем константу указывающую задержку в мс (чем выше значение, те
```

```

void setup(){
  Serial.begin(9600); // Инициуем передачу данных в монитор последовательного порта
  if(led.begin()) {Serial.println("Ok");} // Инициуем работу со светодиодами и выводим сообщение об успешной инициализации
  else {Serial.println("Err");} // Если инициализация невозможна (не хватает памяти для хранения цветов)
}
void loop(){
  j++; // Смещаем спектр цветов для всех светодиодов
  for(uint16_t i=0; i<led.count(); i++){ // Проходим по всем светодиодам
    k=((uint16_t)(i*256/led.count()+j)); // Определяем положение очередного светодиода на смещённом спектре цветов
    if(k<85) { b=0; r=k*3; g=255-r;}else // Перелив от зелёного к красному, через жёлтый
    if(k<170){k-=85; g=0; b=k*3; r=255-b;}else // Перелив от красного к синему, через фиолетовый
    {k-=170; r=0; g=k*3; b=255-g;} // Перелив от синего к зелёному, через голубой
    led.setColor(i, r,g,b); // Устанавливаем выбранный цвет для очередного светодиода
  } led.write(); // Записываем цвета всех светодиодов
  delay(z); // Устанавливаем задержку
}

```

Данный пример может работать и с модулями на большее количество светодиодов, тогда Вам нужно указать их количество во второй строке скетча. Скорость перелива зависит от значения константы z, чем больше её значение, тем медленнее перелив.

## Описание основных функций библиотеки:

### Подключение библиотеки:

```

#include <iarduino_NeoPixel.h> // Подключаем библиотеку iarduino_NeoPixel для работы с модулями
iarduino_NeoPixel led( №_ВЫВОДА , КОЛИЧЕСТВО_СВЕТОДИОДОВ ); // Объявляем объект led для работы с модулями подключёнными к ука

```

### Функция begin();

- Назначение: Инициализация работы с модулями NeoPixel.
- Синтаксис: begin();

- Параметры: Нет.
- Возвращаемые значения: bool, 1 (успешная инициализация) или 0 (отказ инициализации, не хватает памяти).
- Примечание: Вызывается 1 раз в коде setup.
- Пример:

```
if(led.begin()) {Serial.println("Ok" );} // Инициуем работу с модулями NeoPixel и выводим сообщение об успешной инициализации
else           {Serial.println("Err");} // Не хватает памяти для хранения цветов всех светодиодов, выводим сообщение об ошибке
```

## Функция setColor();

- Назначение: Указание цвета для адресного светодиода.
- Синтаксис: setColor( №\_СВЕТОДИОДА , ЦВЕТ [ , ЯРКОСТЬ ] );
- Параметры:
  - №\_СВЕТОДИОДА - число от 0 до 65534, или NeoPixelAll если требуется установить цвет сразу всем светодиодам.
  - ЦВЕТ - можно указать одним числом (как в графических редакторах), например: 0xFF00FF - фиолетовый. А можно указать тремя параметрами (числами) через запятую, R, G, B, например: 0xFF, 0xFF, 0x00 - жёлтый.
  - ЯРКОСТЬ - необязательный параметр от 0 до 255. Если указать 127 - то указанный цвет не претерпит изменений. Чем ближе число к 0, тем темнее будет указанный цвет, а чем ближе число к 255, тем светлее будет указанный цвет (вплоть до белого).
- Возвращаемые значения: Нет.
- Примечание: Функция устанавливает цвет в массиве цветов, но их запись в модули осуществляется функцией write().
- Пример:

```
led.setColor(0, 0xFF00FF); // Установить фиолетовый цвет для 0 светодиода
led.setColor(1, 0xFF, 0x00, 0xFF); // Установить фиолетовый цвет для 1 светодиода
led.setColor(2, 0xFF, 0x00, 0xFF, 200); // Установить фиолетовый цвет для 2 светодиода, но сделать его более светлым
led.setColor(NeoPixelAll, 0, 255, 0); // Установить зелёный цвет для всех светодиодов
```

## Функция write();

- Назначение: Запись (применение) всех установленных цветов в модули NeoPixel.



- Синтаксис: `write()`;
- Параметры: Нет.
- Возвращаемые значения: Нет.
- Примечание: Если для светодиода новый цвет не установлен, то он сохранит старый цвет.
- Пример:

```
led.write(); // Запись всех установленных цветов в модули NeoPixel
```

### Функция `count()`;

- Назначение: Получение количества используемых светодиодов.
- Синтаксис: `count()`;
- Параметры: Нет.
- Возвращаемые значения: Количество светодиодов, от 0 до 65534.
- Примечание: Функция возвращает число светодиодов, которое было указано при объявлении объекта библиотеки. Эту функцию можно использовать для создания циклов. Если Вы пожелаете изменить количество светодиодов, то Вам будет достаточно сделать это в строке объявления объекта, а не править весь скетч.
- Пример:

```
for(uint16_t i=0; i<led.count(); i++){ // Цикл для прохода по всем светодиодам
...
}
```

### Необязательная функция `getPointer()`;

- Назначение: Получение указателя на массив цветов светодиодов.
- Синтаксис: `getPointer()`;
- Параметры: Нет.
- Возвращаемые значения: `uint8_t*`, Указатель на массив цветов светодиодов.
- Примечание: Это необязательная функция которая возвращает указатель на массив цветов светодиодов. Используя этот массив Вы сами

можете формировать цвета в обход функции setColor(). А так же можете считывать ранее установленные цвета.

- байты 0, 1, 2 массива, это цвета 0 светодиода (G-зелёный, R-красный, B-синий).
  - байты 3, 4, 5 массива, это цвета 1 светодиода (G-зелёный, R-красный, B-синий) и т.д.
  - размер массива зависит от количества светодиодов указанных при объявлении объекта библиотеки.
- Пример:

```
uint8_t *ptrLED = led.getPointer();           // Получаем указатель на массив цветов светодиодов
ptrLED[0]=255; ptrLED[1]=0; ptrLED[2]=0;     // Устанавливаем зелёный цвет для 0 светодиода
ptrLED[3]=0; ptrLED[4]=255; ptrLED[5]=0;     // Устанавливаем красный цвет для 1 светодиода
ptrLED[6]=0; ptrLED[7]=0; ptrLED[8]=255;    // Устанавливаем синий цвет для 2 светодиода
led.write();                                 // Записываем установленные цвета в модуль.
```

## Применение:

- Декоративное освещение;

## Ссылки:

- [Библиотека `arduino\_NeoPixel`, позволяющая работать с адресными светодиодами и модулями NeoPixel.](#)
- [Wiki - Установка библиотек в Arduino IDE](#)