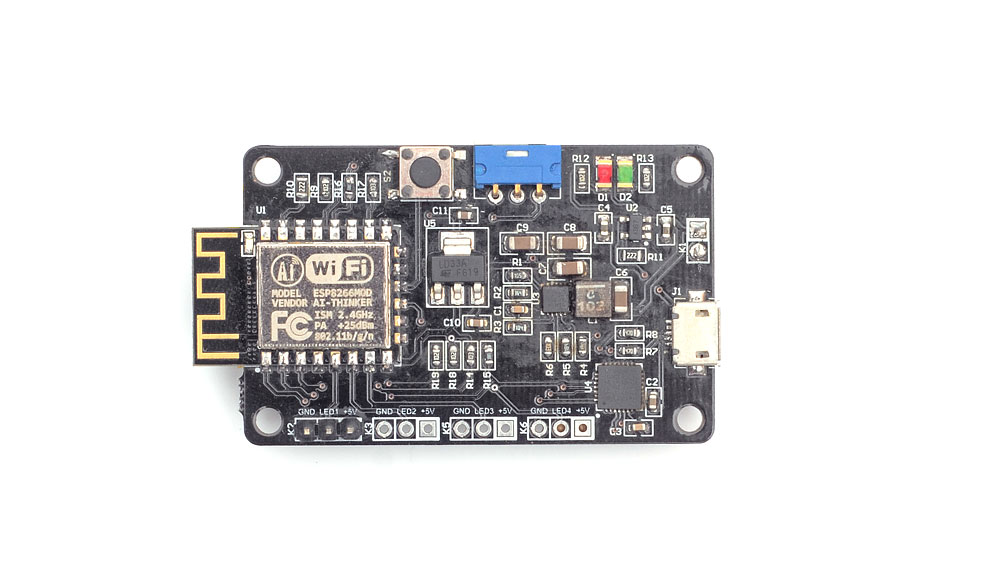
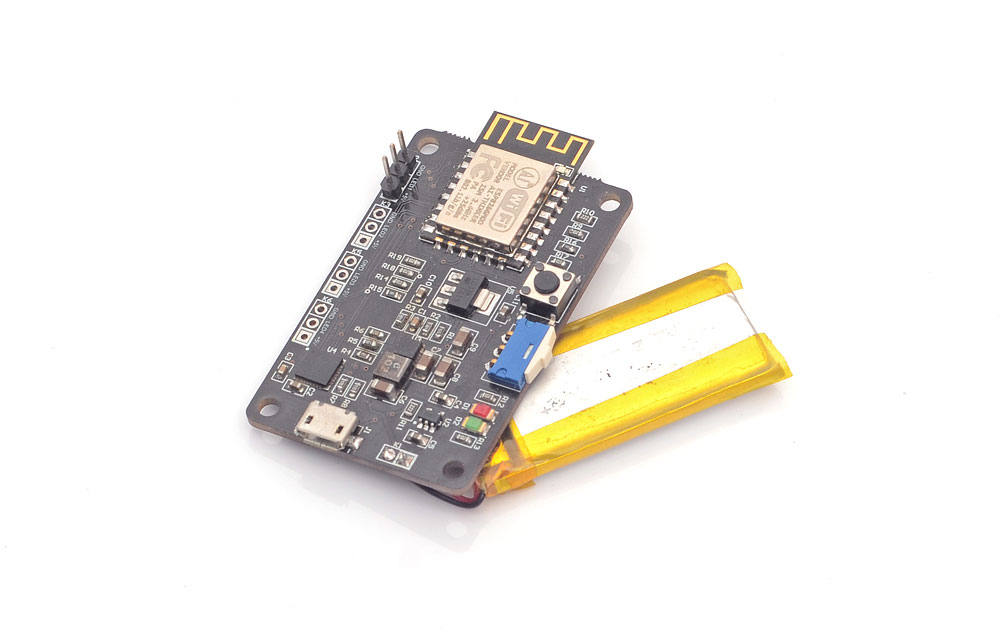
Микроконтроллер ESP8266 в кратчайшие сроки обрел популярность, причин тому несколько: высокая мощность, наличие Wi-Fi, возможность писать код в Arduino IDE и низкая цена – это лишь основные. На базе ESP выпускается огромное количество всевозможных модулей под различные типы задач, один из них – автономный контроллер адресных светодиодных лент Leduino-W.



Что же отличает Leduino-W от других плат на базе ESP8266? В первую очередь это наличие встроенного повышающего преобразователя питания с выходным напряжением 5 В и током до 3 А. Многим известно, что полностью заряженный Li-ion аккумулятор выдает около 4.2 В, но по мере разряда напряжение достаточно быстро падает до 3.7 В, а затем плавно до 3 В. При таком напряжении светодиодные ленты светят уже довольно тускло.

Благодаря повышающему преобразователю в Leduino-W лента будет светиться одинаково ярко на всем протяжении разрядки АКБ, а когда его напряжение опустится ниже 3 В, то сработает защита от переразряда. Включение/выключение питания осуществляется при помощи движкового переключателя, а зарядка АКБ – через MicroUSB разъем.

Через этот же разъем происходит и непосредственно прошивка ESP, т.к. на плате установлен USB-UART конвертер CP2102.



Пара слов об адресных светодиодах и чем такие ленты отличаются от обычных RGB лент. Если рассматривать обычную светодиодную RGB ленту, то все светодиоды на ней в один момент времени светятся одним и тем же цветом. В адресной ленте все иначе – каждый светодиод может светиться своим цветом абсолютно независимо от других. Из таких лент делают Ambilight подсветку, используют при создании всевозможных световых эффектов, в гирляндах – области применения безграничны.

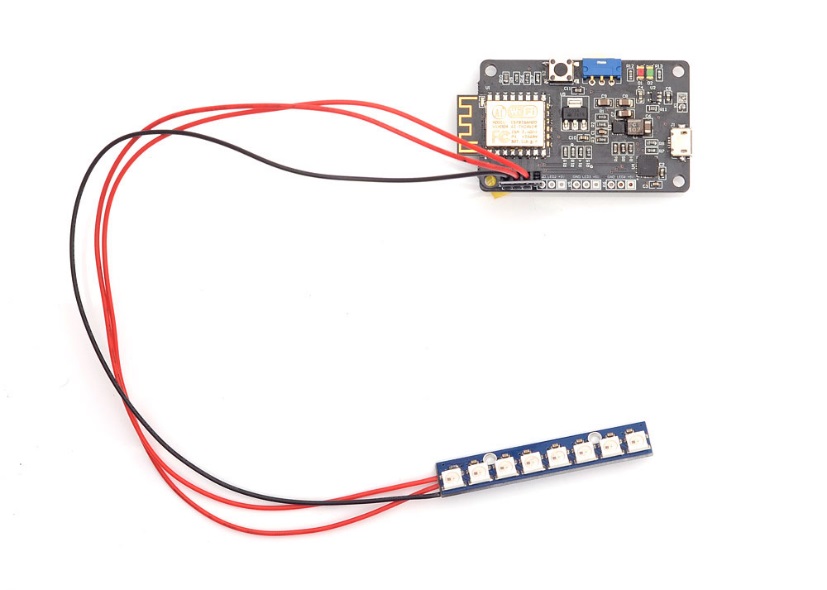
Если углубиться в теорию, то название «адресный» не совсем верно, т.к. у каждого светодиода нет своего собственного адреса (как, например, у устройств, работающих по шине I2C). Есть вход данных и выход, т.е. команда до самого дальнего светодиода идет через все предшествующие. По факту мы обращаемся не к конкретному адресу, а к порядковому номеру светодиода в цепочке.

Наиболее распространенный тип адресных светодиодов – WS2811 и WS2812B, на их базе выпускаются как светодиодные ленты, так и огромное количество разнообразных модулей. Мы рассмотрим основы работы с WS2812B на примере модуля, содержащего 8 таких светодиодов:

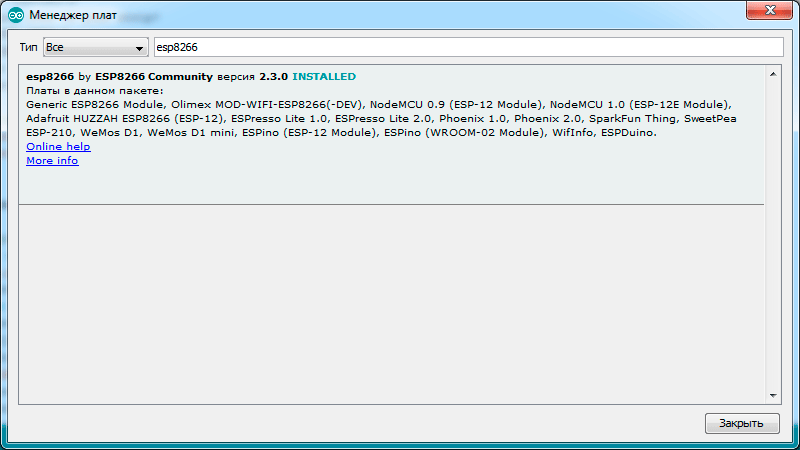


У модуля есть «направление», т.е. с одной стороны у нас вход для данных (DIN), а с другой – выход (DOUT). Т.к. это у нас первый модуль, то DIN подключаем непосредственно к контроллеру, а к DOUT мы может подключить еще один такой модуль (DOUT первого подключим к DIN второго).

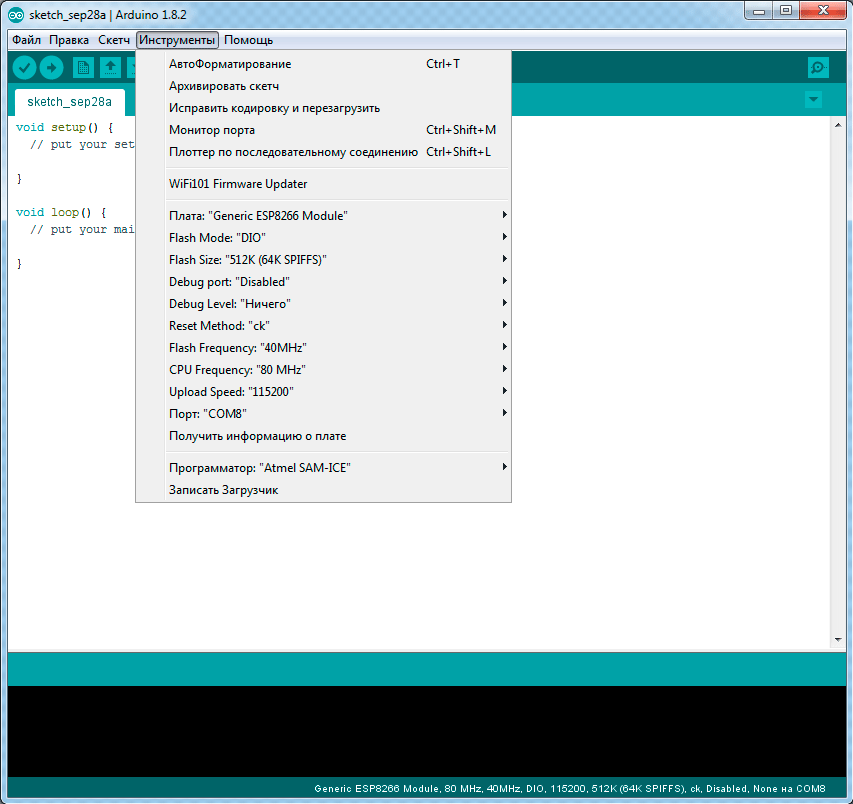
На Leduino-W есть 4 независимых канала для адресных светодиодов, мы будем использовать первый, в Arduino IDE он соответствует Pin’у с номером 14.



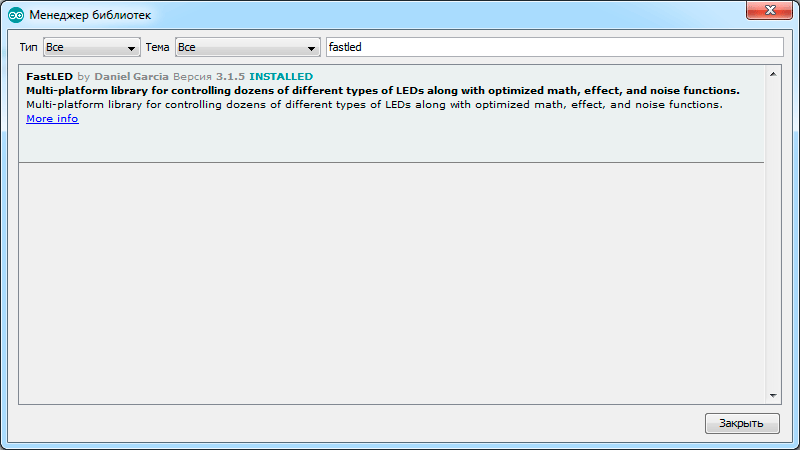
Теперь открываем Arduino IDE и устанавливаем пакет для работы с микроконтроллерами ESP8266, для этого переходим в Инструменты – Плата – Менеджер плат… и в поиск вписываем «ESP8266».



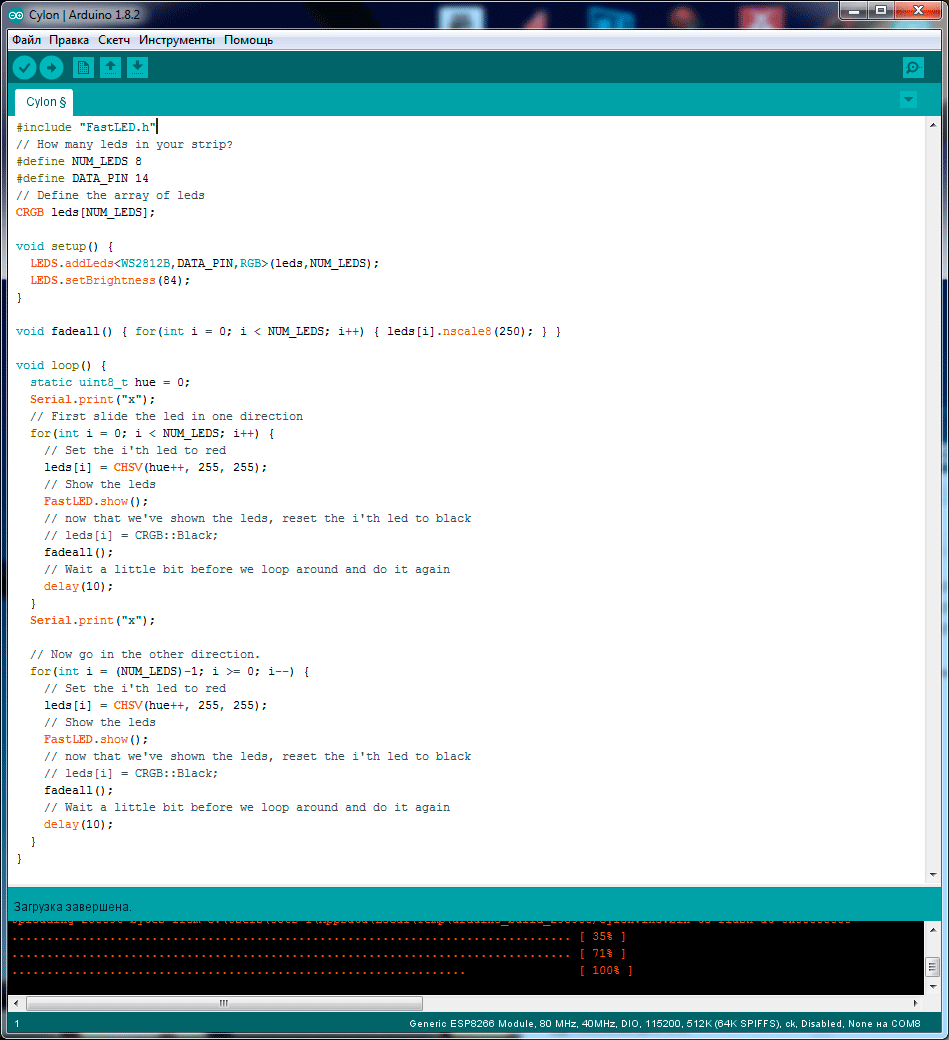
И устанавливаем пакет «esp8266», после чего необходимо выбрать нужную плату и установить ее параметры как на изображении ниже:



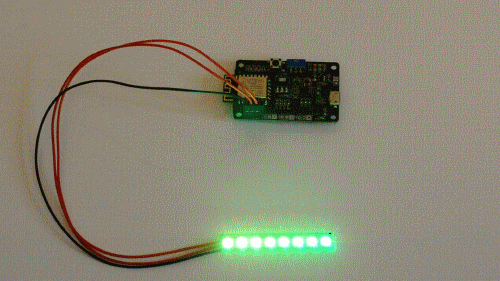
Готово! Теперь скачиваем библиотеку для работы с WS2812B, их существует большое количество, для примера мы разберем «FastLed», для этого переходим в Скетч – Подключить библиотеку – Управлять библиотеками, после чего в поиске пишем «fastled» и устанавливаем библиотеку.



Теперь можно приступать к экспериментам! Начнем со встроенных примеров, например, Cyclon. В коде зададим правильное количество светодиодов (у нас их 8), DATA\_PIN (первый канал у нас на 14-ой «ноге»), а также тип светодиодов – WS2812B.

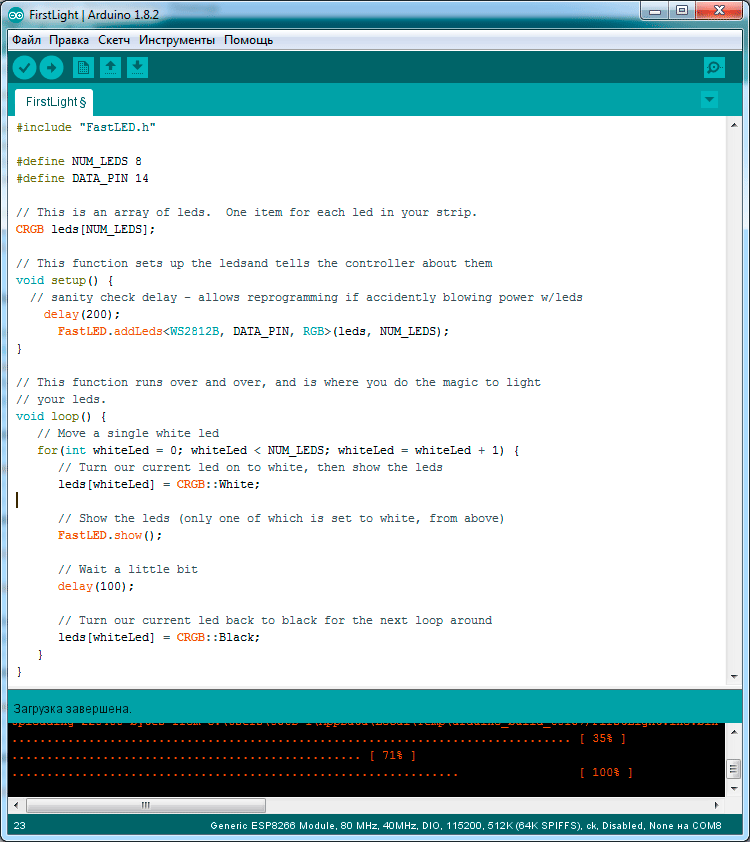


Прошиваем и смотрим на результат:

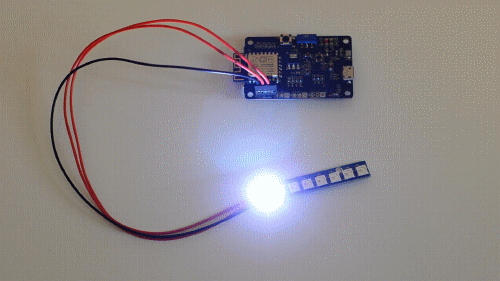


Пока что явных видимых отличий от обычной ленты нет, т.к. контроллер плавно перебирает все цвета по кругу.

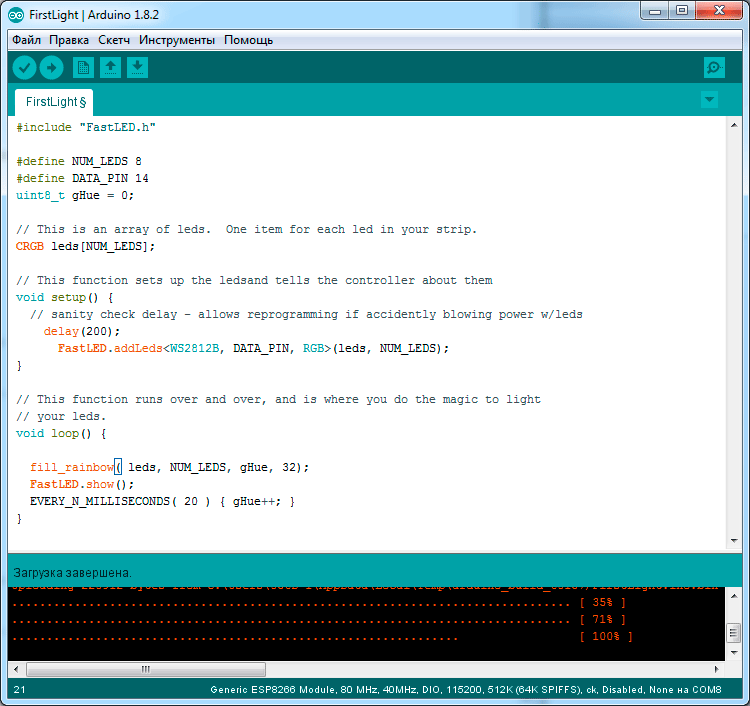
Теперь попробуем пример FirstLight, опять же не забываем поменять настройки количества и типа светодиодов, а также задать правильный DATA\_PIN.



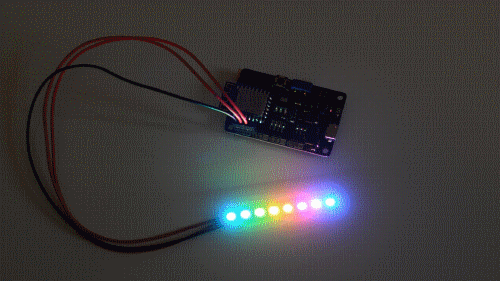
В результате у нас по модулю «бегает» белый цвет. Из кода видно, что в цикле for мы перебираем по очереди все светодиоды, при этом он сначала загорается белым, далее задержка 100 мс, после который он гасится.



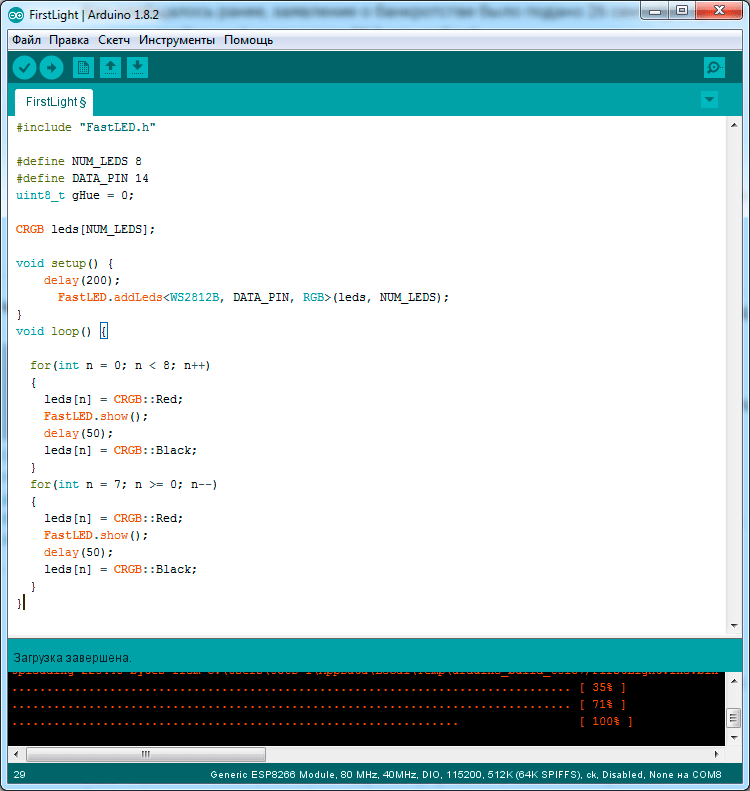
Испробуем встроенные функции, например, сделаем сдвигающуюся радугу при помощи fill\_rainbow – указываем массив светодиодов, их количество, цветовую составляющую и шаг. Т.к. у нас всего 8 светодиодов и мы хотим получить полную радугу, то 256 возможных значений для gHue мы поделим на 8, отсюда и получается шаг в 32 единицы. Каждые 20 мс происходит сдвиг цвета на 1 единицу.



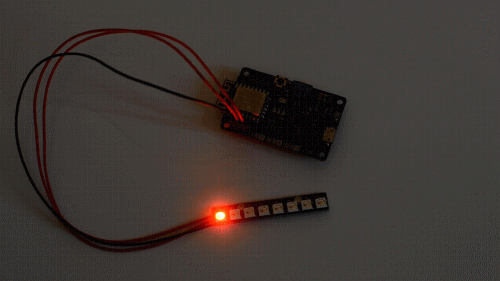
И результат:



Напоследок вспомним Рыцаря дорог, машину по прозвищу Кит и тот самый «бегающий красный огонек» на радиаторной решетке. Делаем два цикла – в одном поочередно сдвигаем активный светодиод в одну сторону, а затем в другом цикле возвращаем обратно.



Чем не Кит?



Мы рассмотрели только базовые примеры работы с адресными светодиодами для того, чтобы показать их суть – возможность каждому светодиоду светиться своим цветом независимо от других. В связке с Leduino-W такие светодиодные ленты могут стать отличным украшением для всевозможных костюмов и реквизита в светодиодных шоу благодаря работе от аккумулятора и наличию Wi-Fi.