

# Радиосеть управления освещением с использованием технологии Wifi (esp8266) Часть 3



Roman Kurmaev, 19 мая 2016г.

[Ссылка на первую часть.](#)

[Ссылка на вторую часть.](#)

## Разработка алгоритма управления освещением с помощью esp8266.

Как уже было сказано в первой части данной работы. Непосредственно управлять свечением светодиодной ленты можно посредством ШИМ выходов esp8266. Исходя из этого, пользователь должен иметь возможность управлять ШИМ выходами esp8266. Прошивка NodeMCU позволяет сделать это с помощью следующих команд:

Настройка ШИМ

```
pwm.setup(1,500,512);
```

где 1 – номер вывода esp8266, 500 - частота дискретизации ШИМ, 512 – количество уровней ШИМ

Запуск режима ШИМ

```
pwm.start(1)
```

где 1- номер вывода esp8266

```
pwm.setduty(1,217)
```

где 1- номер вывода esp8266 , 217 – уровень ШИМ (от 0 до 512).

Очевидно, что вводить в консоль данные команды, каждый раз, когда нужно изменить свечение ленты нерентабельно. Устройство должно получать команды от пользователя более простым (для пользователя) путем.

В настоящий момент широкое распространение получили два способа взаимодействия с устройствами IoT:

1. Посредством «облачных» сервисов взаимодействующих с устройством посредством UDP (User Datagram Protocol).
2. С помощью web - интерфейса взаимодействующим с устройством с помощью TCP(Transmission Control Protocol).

Каждый из представленных способов имеет свои преимущества и недостатки.

В данном проекте в качестве способа взаимодействия с пользователем будет использоваться Web интерфейс. Выбор обусловлен в основном недоверием к сторонним сервисам и простотой реализации web – интерфейса.

Esp8266 имеет 2 возможных режима работы:

1. Soft AP - esp8266 создает собственную Wi – Fi сеть выполняя функции точки доступа;
2. Station – esp82266 подключается к уже существующей сети Wi – Fi.

В данном проекте в качестве режима работы esp8266 выбран режим Station. Выбор обусловлен наличием всего 1-го модуля, большим радиусом приема (по сравнению с Soft AP), возможностью взаимодействия по сети Internet.

Исходя из вышесказанного модуль после подачи на него питания , должен выполнить следующие действия:

1. Зарегистрироваться в существующей сети Wi – Fi;
2. Запустить веб – сервер;
3. Инициализировать выходы ШИМ
4. Ожидать авторизацию/команду пользователя.

Наглядно действия можно представить в виде блок схемы (Рис.1)

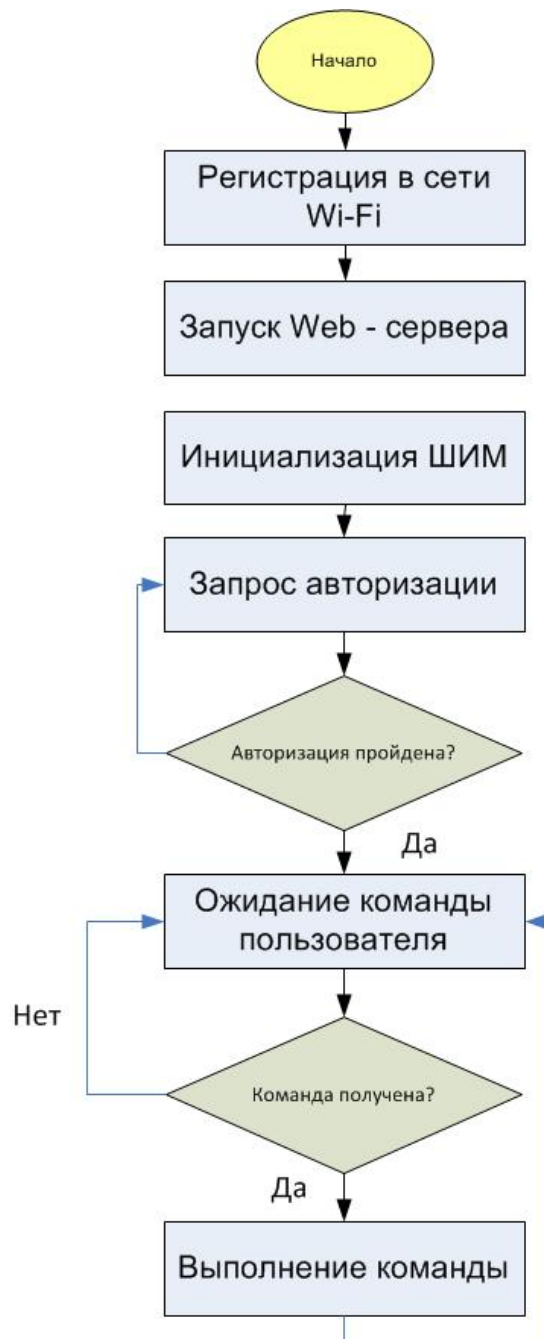


Рисунок 1. Алгоритм работы esp8266.

Для доступа к web - интерфейсу пользователю необходимо знать ip адрес устройства в сети.

Устройство может получать IP адрес от Wi-Fi роутера посредством DHCP. При этом адрес привязан к MAC - адресу данного устройства.

Для примера в сети:

192.168.0.102 – адрес esp8266

192.168.0.103 – адрес устройства пользователя.

Тогда взаимодействие пользовательского терминала с esp8266 можно представить на следующей диаграмме (рис.2).



Рисунок 2. Взаимодействие пользовательского терминала с esp8266.

### Разработка Web-интерфейса.

Web –интерфейс должен обладать следующими качествами:

1. Высокая скорость загрузки;
2. Минимальная нагрузка на esp8266;
3. Корректное отображение на устройстве пользователя;
4. Интерфейс должен быть интуитивно понятен пользователю.

Минимальный интерфейс содержит:

1. 3 формы для ввода уровня ШИМ;
2. Кнопка отправки команды;
3. Отображение текущего состояния выводов ШИМ.

Enter value for red:	<input type="text" value="512"/>
Enter value for green:	<input type="text" value="512"/>
Enter value for blue:	<input type="text" value="512"/>
<input type="button" value="Save"/>	

Рисунок 3. Минимально достаточный Web –интерфейс.

Стоит заметить, что для увеличения скорости загрузки, размер страницы не должен превышать 4096 байт. Это ограничение связано с максимальным размером HTTP запроса.

Страница в формате HTML см. в приложениях к данной статье.

Разработка lua скрипта.

Разрабатываемый скрипт можно разбить на несколько блоков:

Блок инициализации – устанавливает режимы ШИМ на выводах, выполняет подключение к сети Wi – Fi, запускает Web – сервер;

Блок авторизации – выполняет проверку авторизации пользователя;

Блок обработки команд - принимает и обрабатывает команды пользователя.

Первый блок выполняется только 1 раз при подаче питания на устройство. Второй блок повторяет свое выполнение до тех пор, пока пользователь не будет авторизован. Третий блок повторяется в бесконечном цикле.

Код программы см. в приложении к статье.

### Реализация конечного устройства

Конечное устройство включает в себя:

1. Блок питания 12 v – для питания светодиодной ленты;
2. Преобразователь напряжения 12v-3.3 v;
3. Esp8266;
4. Светодиодную RGB ленту;
5. Три МДП транзистора.

Наличие МДП транзисторов обусловлено тем что максимальное напряжение на выводах esp8266 составляет всего три вольта, что не достаточно для яркого свечения RGB ленты.

Схему конечного устройства можно представить следующим образом(Рис.4).

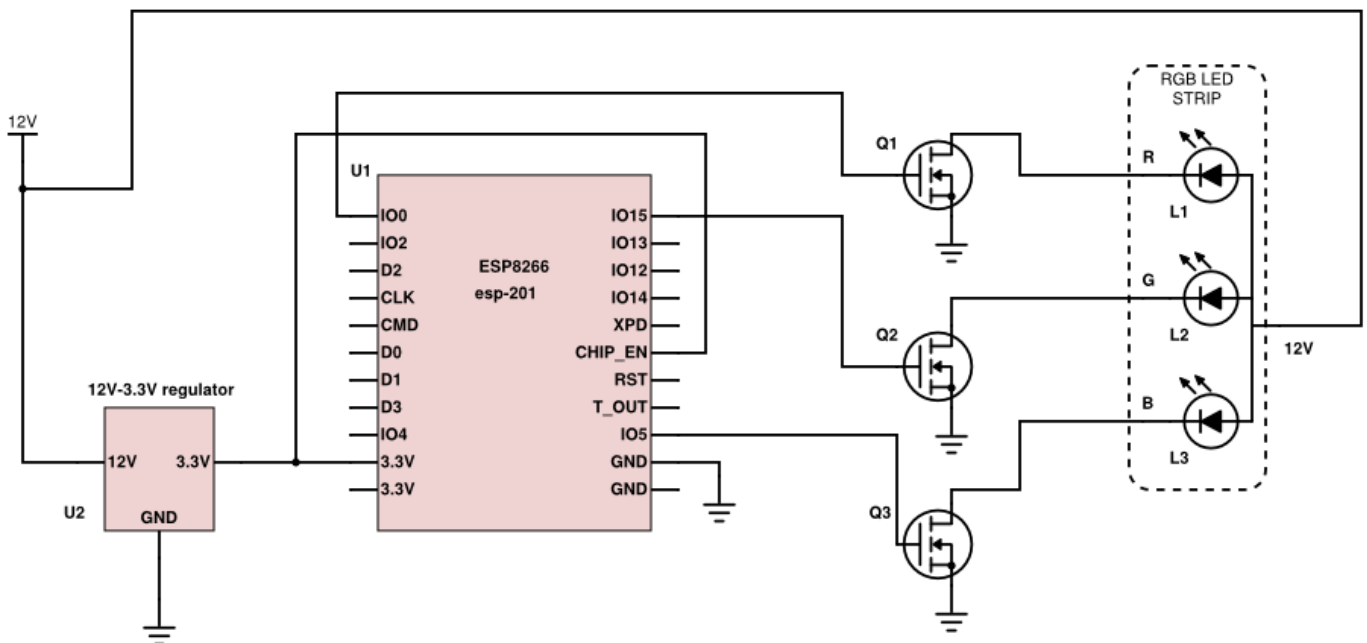


Рис.4 Схема конечного устройства.

### Используемая литература:

1. Спецификация esp8266:  
[https://nurdspace.nl/images/e/e0/ESP8266\\_Specifications\\_English.pdf](https://nurdspace.nl/images/e/e0/ESP8266_Specifications_English.pdf);
  2. Распиновка ESP8266, различные модификации модулей на базе ESP8266:  
<http://esp8266.ru/modules-esp8266/>;
  3. Модуль NodeMCU DevKit v0.9:  
<http://www.chipdip.ru/news/nodemcu-devkit-v0.9-module/>;
1. Comparison of ESP8266 NodeMCU development boards:  
<http://frightanic.com/iot/comparison-of-esp8266-nodemcu-development-boards/>

