

Бюджетный Wi-Fi-модуль ESP8266: быстрый старт без регистрации и SMS

В статье рассматривается Wi-Fi-модуль ESP-WROOM-02 (ESP8266) компании Espressif Systems, приведены примеры из набора Arduino IDE.

Евгений Рахно
evgenij.rakhno@microchip.ua

Первоначально, в момент выхода на рынок, модуль ESP8266 представлял собой «черный ящик» и был предназначен для реализации подключения встроенных приложений к сети Wi-Fi. Взаимодействие с микроконтроллером осуществлялось при помощи AT-команд. Именно в таком виде модуль надежно закрепился на рынке, так как ближайшие конкуренты, предоставляющие подобный функционал, стоили, как минимум, вдвое дороже.

Однако спустя некоторое время Espressif пришла к выводу, что ресурсов микроконтроллера ESP8266 хватит не только на поддержку Wi-Fi, но и на программу пользователя, и предоставила разработчикам комплект документации на него. Так появился «легендарный ESP8266», столь любимый «ардуинщиками».

Политика продвижения данного модуля на мировой рынок была весьма необычной: «У нас есть модуль. Он такой, какой есть. Схемотехнику модуля мы берем на себя, а софт отдаем вам. Внутри модуля, непосредственно к регистрам, мы вас не пустим, но предоставим библиотеки для работы со всей периферией, имеющейся в модуле». И, судя по всему, такая маркетинговая стратегия сработала, ведь модули компании Espressif заняли довольно-таки устойчивое положение на рынке.

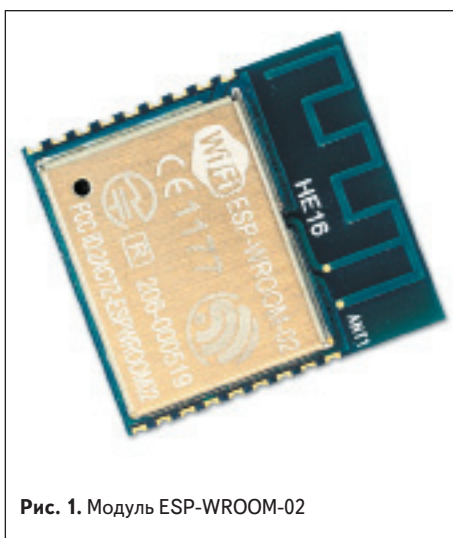


Рис. 1. Модуль ESP-WROOM-02

Физически ESP-WROOM-02 представляет собой SMD-модуль, в состав которого входят микроконтроллер ESP8266EX, микросхема SPI Flash-памяти емкостью 2 Мбайт и интегрированная PCB-антенна (чувствительность 2 дБи). Для обеспечения всех требований EMI компоненты модуля экранированы металлической крышкой. Геометрические размеры ESP-WROOM-02 составляют 18×20×3 мм (рис. 1).

Внутри ESP8266 находится 32-битный микроконтроллер Tensilica L106 и 16-битный RISC-микроконтроллер с ультранизким энергопотреблением, радиотракт (с этой точки зрения называть ESP8266EX микроконтроллером не совсем правильно, однако давайте согласимся с этой условностью). Тактовая частота процессора составляет 80 МГц и в режиме максимальной производительности может подниматься до 160 МГц. В настоящее время, по заявлениям производителя, поддержка сети Wi-Fi отнимает примерно 20% процессорного времени. Оставшееся время (около 80% MIPS) может быть занято под приложение пользователя. Необходимо отметить, что внутри модуля находится RTOS (операционная система реального времени), что значительно упрощает интеграцию пользовательских задач в его софт.

В распоряжении разработчика находятся следующие периферийные модули: GPIO, UART, I²C, HSPI, I²S, PWM, ADC, IR (Infrared remote). ESP8266 обеспечивает поддержку IPv4, TCP/UDP/HTTP/FTP. Надежность беспроводной передачи данных (Wi-Fi 802.11 b/g/n) обеспечивается применением технологий WPA/WPA2, WEP/TKIP/AES.

Модулем поддерживаются следующие режимы Wi-Fi-соединения: Wi-Fi-клиент, Wi-Fi-точка доступа (SoftAP), точка доступа плюс клиент. Напряжение питания — 2,7–3,6 В, ток потребления (средний) — 80 мА, ток потребления (макс.) — 500 мА. Диапазон рабочих температур –40...+85 °С (по спецзаказу –40...+125 °С).

Сразу отметим, что приведенные выше значения актуальны для модуля ESP-WROOM-02 и могут отличаться от параметров ESP других моделей. Дело в том, что Espressif является производителем микроконтроллера ESP8266EX («производитель» — это еще одна

условность, так как компания не имеет своих фабрик, и все кристаллы выпускаются на производственных мощностях сторонних компаний (т. н. *fabless*), а конечной целью любого коммерческого предприятия является получение прибыли. С этой точки зрения ESP8266EX — это продукт, который можно и нужно продавать, и компания, имеющая соответствующие знания и квалификацию, может обратиться к дистрибьюторам Espressif и приобрести «микроконтроллер» ESP8266EX (естественно, существует MOQ, однако принципиальных проблем с приобретением ESP8266EX нет).

Благодаря высокой степени интеграции любой разработчик может собрать модуль ESP-WROOM-02, всего лишь добавив кварц, микросхему памяти, антенну и несколько пассивных элементов. В Интернете можно легко найти более двух-трех десятков модулей, подобных ESP-WROOM-02, и, на первый взгляд, между ними нет никакой разницы. Но, как говорится, «дьявол кроется в деталях»: поставили хорошую внешнюю антенну — модуль получился лучше, но дороже. Поставили дешевый кварц — модуль получился чуть хуже и чуть дешевле. Есть опыт в разработке PCB-антенн и не пожалели денег на «правильный» текстолит — получили хороший модуль по приемлемой цене. Взяли дешевый FR4, не сделали экранирование и скопировали чертеж антенны из Интернета — получили «дешевый Китай», который не пройдет ни одну сертификацию. Кстати, если говорить о сертификации, то «оригинальный ESP8266» ESP-WROOM-02 имеет следующие сертификаты: FCC/CE/TELEC/KCC/SRRC/IC/NCC.

Чем же модуль ESP8266 так полюбился «ардуинщикам»? Во-первых, встроенный Wi-Fi плюс стеки (TCP/UDP/HTTP/FTP). Во-вторых, очень хорошая программная поддержка, хорошая документированность. В-третьих, быстрый старт. Вот как раз о быстром старте мы и поговорим.

Чтобы ознакомиться с модулем ESP-8266, необходимо установить среду Arduino. Узнав о таких возможностях модуля, как дальность связи, уровни энергопотребления, функционал, можно либо остаться на Arduino, либо перейти к более «взрослым» средам разработки, но начинать мы предлагаем именно с Arduino.

Для установки Arduino нужно зайти на сайт www.arduino.cc и на вкладке **SOFTWARE** найти «Arduino IDE». Далее необходимо следовать всем инструкциям этого сайта и установить Arduino IDE на компьютер. После установки запустите Arduino IDE и пошагово следуйте инструкциям, приведенным ниже:

- Нажмите **File** → **Preferences** и на вкладке **Settings** найдите строку «Additional Board Manager URLs» (рис. 2). В этой строке необходимо прописать адрес: http://arduino.esp8266.com/stable/package_esp8266com_index.json.
- Добавьте ESP8266 в Arduino (рекомендую именно такую последовательность действий): **Tools** → **Board: ...** → **Boards Manager...** (рис. 3).
- В поиске наберите ESP8266. В нижнем окне появится esp8266 by ESP8266 Community (рис. 4).
- Нажмите **Install** и ждите окончания установки.
- Выберите отладочную плату: **Tools** → **Board: (...)** → **Generic ESP8266 Module** (рис. 5) (если у вас уже есть отладочная плата для ESP8266, выберите ее).
- Следующий шаг: **Sketch** → **Include Library** → **Manage Libraries...** (рис. 6).

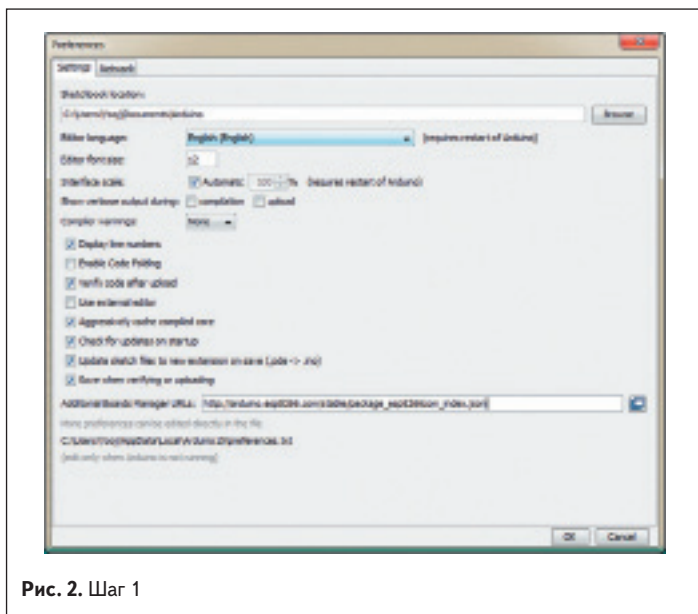


Рис. 2. Шаг 1

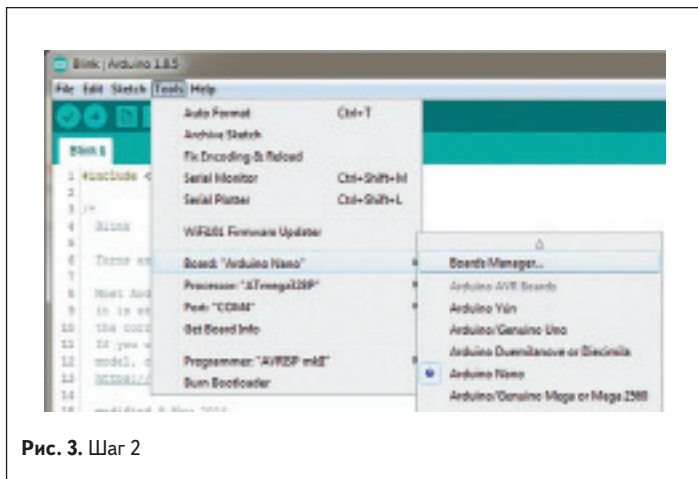


Рис. 3. Шаг 2



Рис. 4. Шаг 3

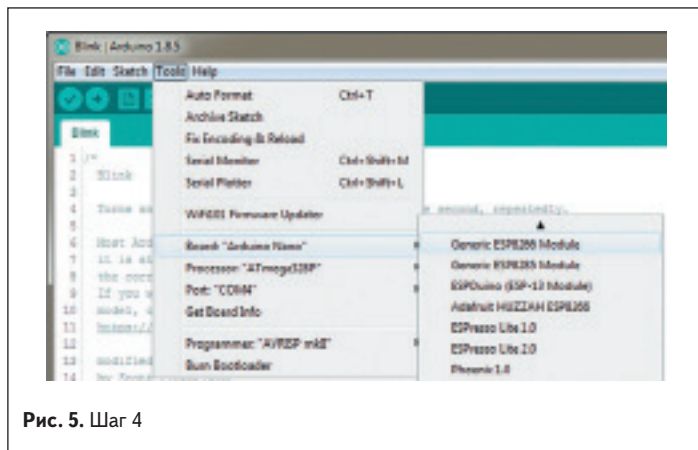


Рис. 5. Шаг 4

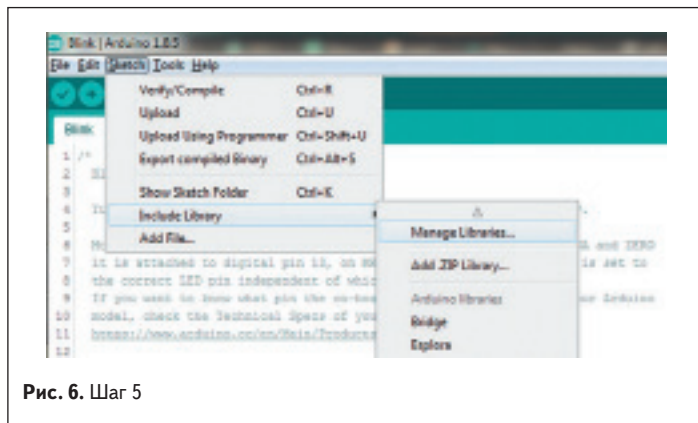


Рис. 6. Шаг 5

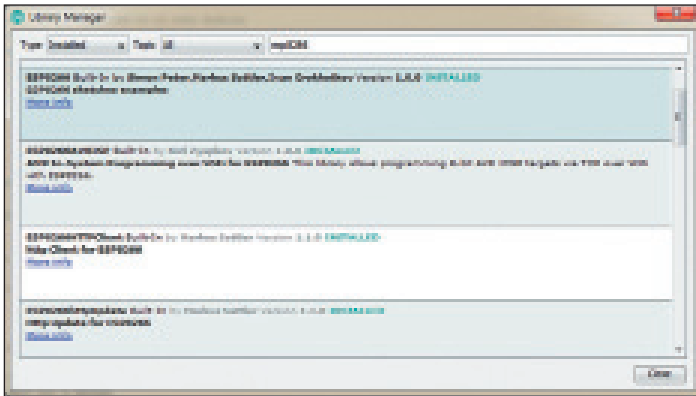


Рис. 7. Шаг 6

- Снова наберите в поиске ESP8266 и найдите библиотеку ESP8266 BuiltIn by Simon Peter, Markus Sattler, Ivan Grokhotkov (рис. 7). Если она не установлена, установите ее.
- Для выбора и покупки отладочной платы нажмите **Tools** → **Board:** «...» → Все, что есть в разделе ESP8266, поддерживается Arduino IDE.
- После того как плата приобретена и подключена к компьютеру, установлены все необходимые драйверы (для преобразователя USB-UART), запустите Arduino IDE. На вкладке **Tools** → **Port:** ... → ... выберите COM-порт, к которому подключен ваш отладочный комплект (рис. 8).
- А теперь самое интересное — «быстрый старт»: **Examples** → **Basics** → **Blink** (рис. 9). Запустится еще одна среда Arduino с примером «Blink» (старую можно закрыть). Нажмите **Verify/Upload** — начнется компиляция и загрузка кода в модуль ESP8266 (возможно, понадобится нажать Reset или Flash). Выглядеть это будет примерно так, как на рис. 10, а светодиод на отладочной плате начнет мигать.

Приятная особенность Arduino заключается в том, что все примеры, идущие в комплекте с данной средой разработки, всегда работают.

Если загрузка в модуль прошла успешно, но светодиод не мигает, то либо ваша отладочная плата вообще не имеет светодиода, либо он подключен к другому порту ввода/вывода, но для перехода к следующему примеру крайне необходимо добиться работы этого примера.

Будем считать, что у вас все получилось, и светодиод наконец-то замигал.

Теперь перейдем ко второму примеру — мигание светодиода через Интернет.

Нажмите **File** → **Examples** → **ESP8266WiFi** → **WiFiWebServer**. В тексте кода найдите строки:

```
const char* ssid = «yourssid»;
const char* password = «yourpassword»;
```

Затем «yourssid» замените на имя вашего роутера, а «yourpassword» — на пароль к нему.

Компилируйте и заливайте. Затем нажмите **Tools** → **Serial Monitor** (рис. 11). В результате вы получите сообщение, что модуль успешно подключился к Wi-Fi-роутеру, и ему был присвоен IP 192.168.1.152.

Откройте браузер. Для включения светодиода введите в браузере `http://192.168.1.152/gpio/0`, для выключения — `http://192.168.1.152/gpio/1`.

У вас теперь есть простенькое устройство, управляя которым можно через Интернет.

Перейдем к третьему примеру — Wi-Fi-клиент.

Отключаем первую плату (WiFiWebServer) и подключаем вторую. После этого запускаем Arduino IDE: **File** → **Examples** → **ESP8266WiFi** → **WiFiWebClientBasic**.

В вызове функции `WiFiMulti.addAP(«SSID», «passpasspass»)` вводим логин и пароль такие же, как во втором примере.

В функции loop находим строку

```
const char * host = «192.168.1.1»;
```

и меняем этот адрес на значение из второго примера. В нашем случае это

```
const char * host = «192.168.1.152»;
```

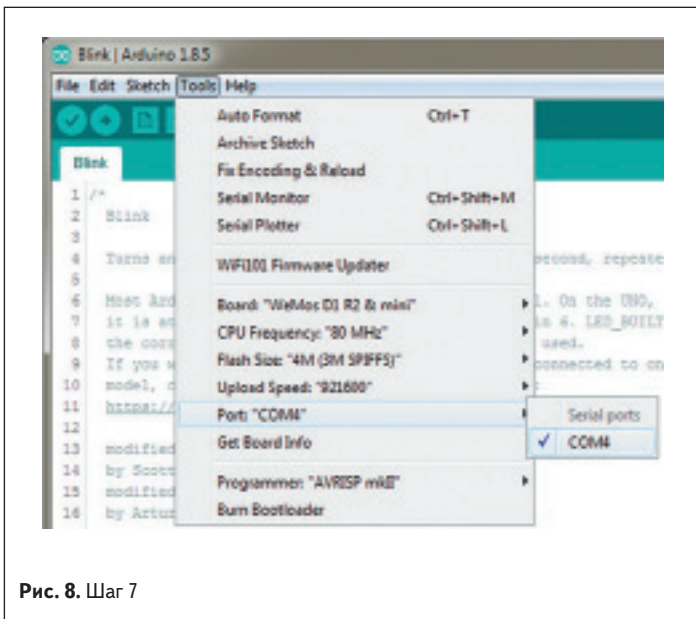


Рис. 8. Шаг 7

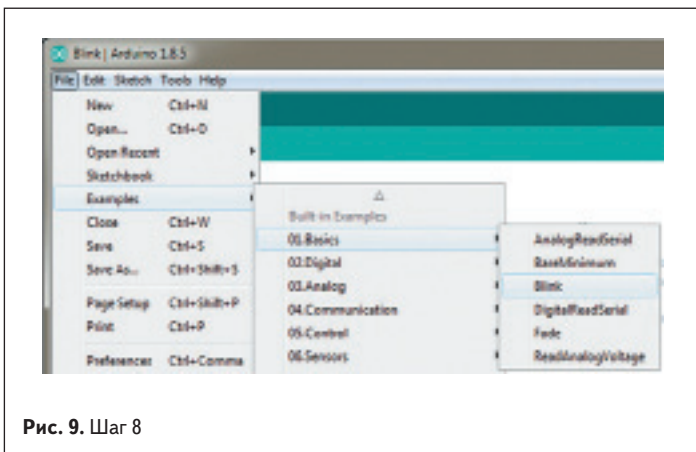


Рис. 9. Шаг 8

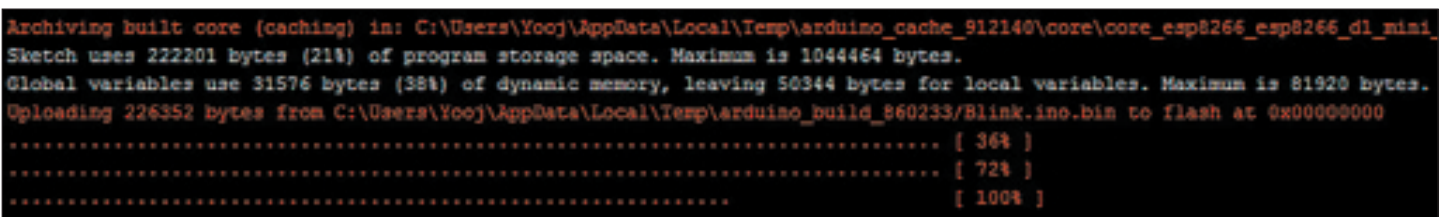


Рис. 10. Шаг 9

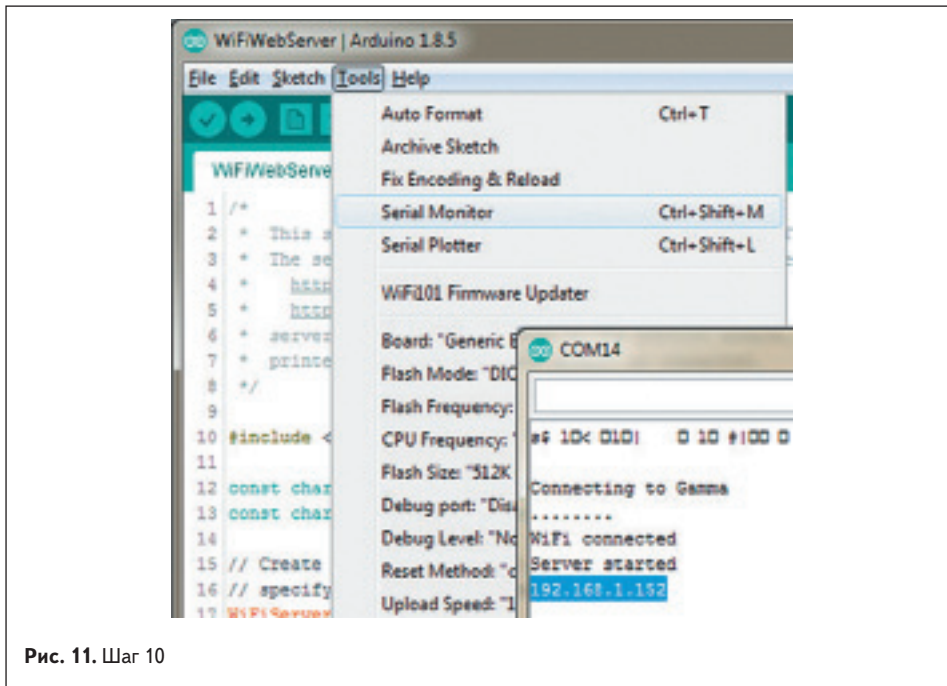


Рис. 11. Шаг 10

а у вас адрес будет другой. Далее в функции **loop** находим строки

```
//read back one line from server
String line = client.readStringUntil('r');
client.println(line);
```

и удаляем их. Выделяем фрагмент, начиная с `if (!client.connect...)` и до конца функции, копируем его в буфер обмена и вставляем затем в конец функции. Должно получиться так, как показано на рис. 12. Находим первую строку

```
client.print(«Send this data to server»);
```

и заменяем ее на

```
client.print(«GET /gpio/0 HTTP/1.1»);
```

Находим вторую строку

```
client.print(«Send this data to server»);
```

и заменяем ее на

```
client.print(«GET /gpio/1 HTTP/1.1»);
```

Компилируем, запускаем. Включаем первую плату (с примером **WiFiWebServer**) и смотрим на мониторе ее сообщения (нас интересует, не изменился ли IP-адрес). Если он не изменился, то все хорошо; если изменился, надо будет внести соответствующие правки в код **WiFiWebClientBasic**).

Суть данного примера: клиент посылает Get-запрос (`/gpio/0` или `/gpio/1`), и, в зависимости от содержимого этого запроса, светодиод на плате **WiFiWebServer** включается либо выключается

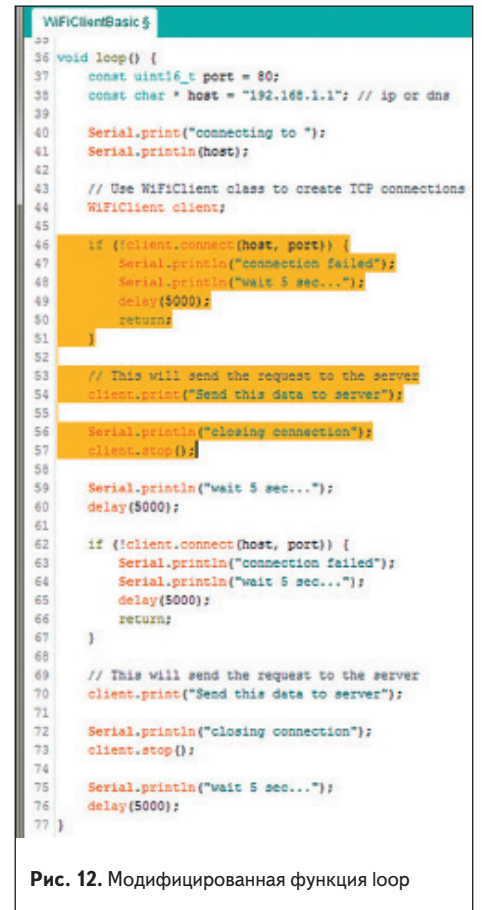


Рис. 12. Модифицированная функция loop

(для тренировки попробуйте самостоятельно изменить код таким образом, чтобы программа клиента (**WiFiWebClientBasic**) анализировала ответ сервера, и светодиоды на обеих платах работали синхронно).

Таким образом, у нас получился «комплект», состоящий из **WiFiServer** и **WiFiClient**, а дальнейшее самостоятельное изучение примеров, входящих в комплект Arduino IDE, лишь укрепит вашу уверенность в том, что после появления Espressif Wi-Fi стал намного доступнее. ■