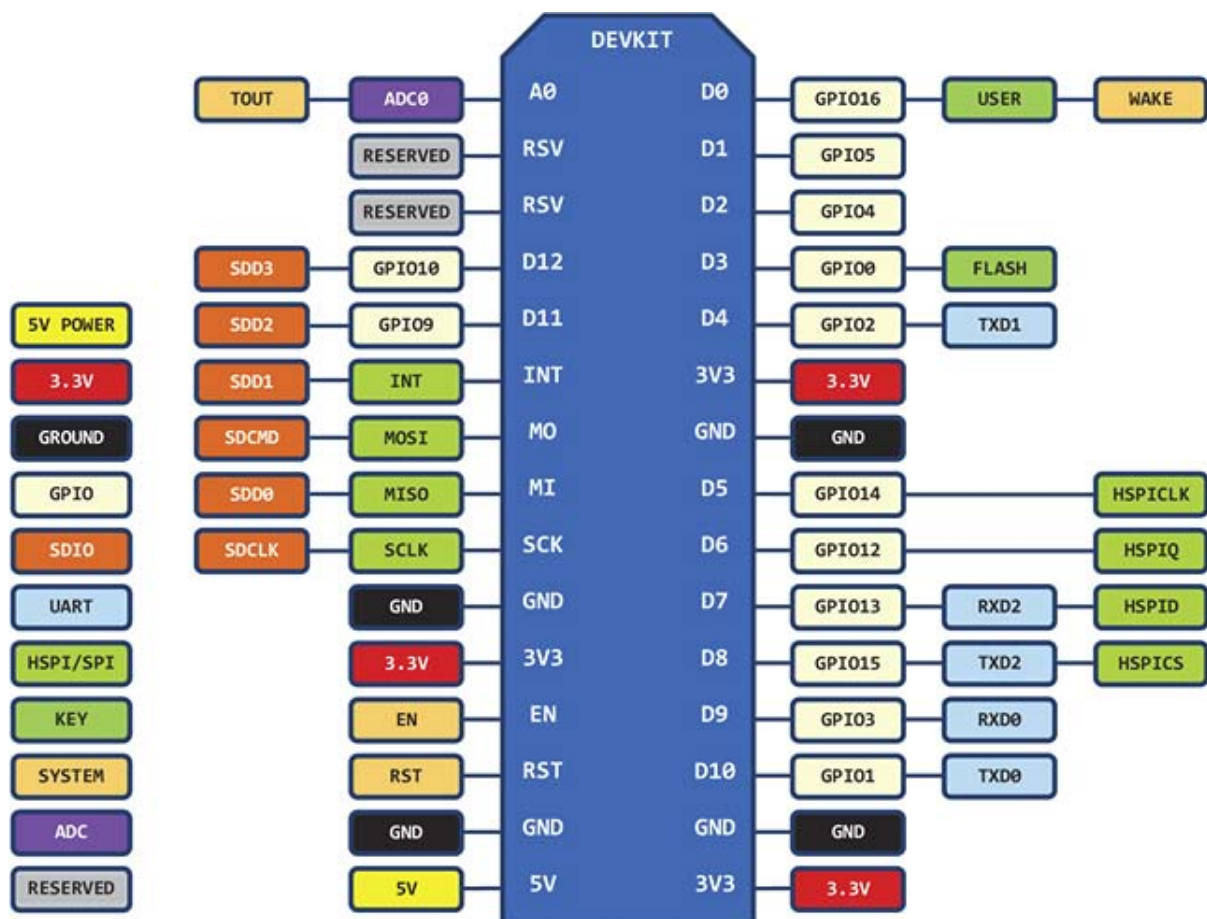


Рассмотрим еще одну плату на основе популярного модуля ESP8266. Это плата Интернет вещей NodeMcu v3 Lua WI-FI ESP8266 CH340.



[NodeMCU](#) — это полноценная платформа для создания устройств IoT (Интернет вещей) на основе модуля ESP8266, который умеет принимать и посылать данные в локальную сеть или интернет через Wi-Fi. Плата построена на основе одной из последних модификаций модулей ESP8266 – модуля ESP-12E. Назначение контактов платы NodeMCU представлено на рисунке.



В платформу загружена прошивка NodeMCU. В неё встроен интерпретатор скриптового языка Lua. Интерпретатор этого языка распространяется свободно, с открытыми исходными текстами на языке Си. С помощью интерпретатора можно создавать файлы, загружать их в флеш-память ESP8266 и выполнять их. Также имеется Autorun (init.lua) для выполнения скриптов при запуске платы. С помощью lua-команд мы можно:

- подключаться к WiFi точке доступа;
- выступать в роли WiFi точки доступа;
- уходить в глубокий сон для снижения энергопотребления;
- привязать lua-функцию к кнопке на GPIO16;
- включать/выключать светодиод на GPIO16;
- создавать/записывать/читать/выполнять/искать/удалять/выводить список файлы во флеш памяти;
- автоматически находить открытую WiFi сеть и подключаться к ней (режим startsmart);
- выводить свой MAC адрес;
- управлять пользовательским таймером;
- управлять таймером WatchDog ;
- управлять (запись/чтение/триггер) выводами GPIO1 — GPIO5, GPIO10, GPIO12-GPIO15;
- PWM (ШИМ) GPIO1 — GPIO5, GPIO10, GPIO12-GPIO15;
- TCP/IP сокет;
- режим WEB сервера;
- I2C — адресация/запись/чтение;
- АЦП 10 бит на пине TOUT.

Полное описание команд можно посмотреть на github (https://github.com/nodemcu/nodemcu-firmware/wiki/nodemcu_api_ru).

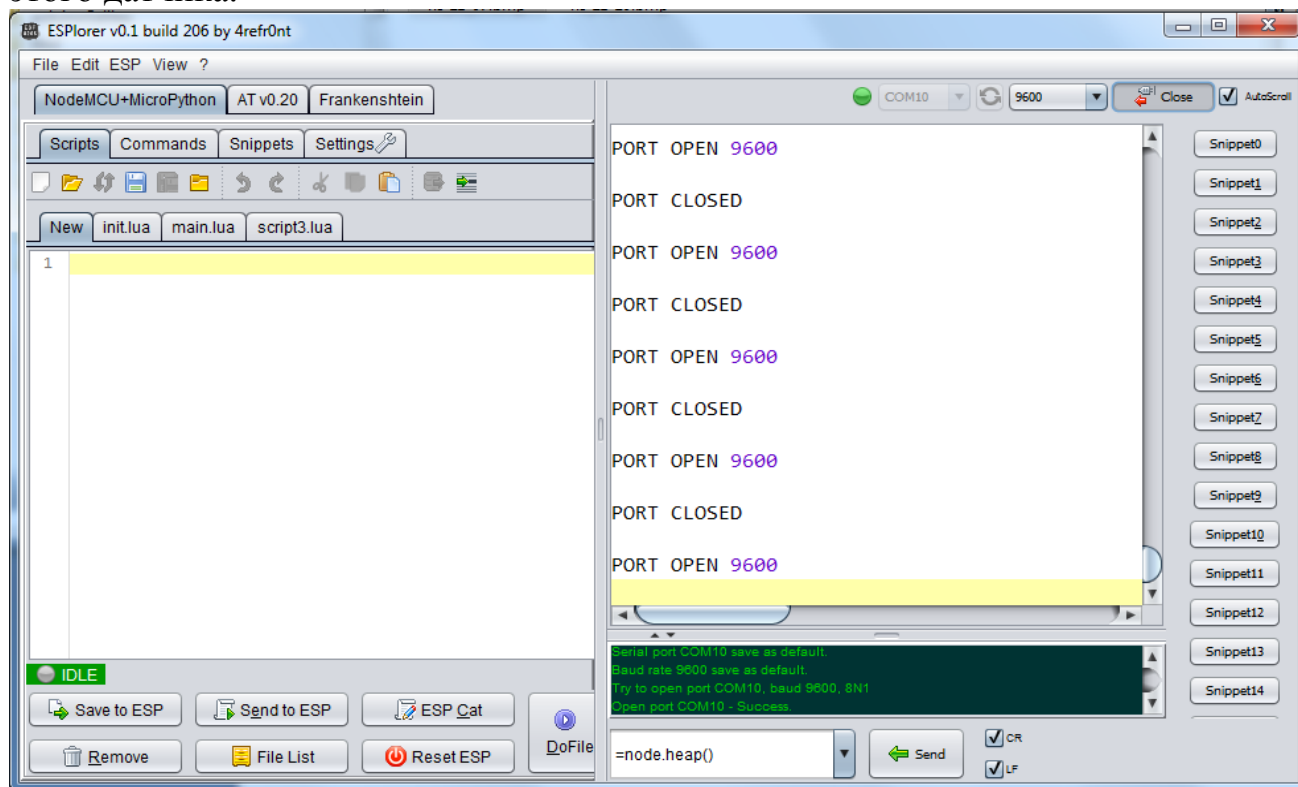
Программировать плату можно в среде Arduino IDE (версия 1.6.5 и выше), а можно использовать комплект средств разработки (SDK) – ESPlorer, который можно скачать со страницы <http://esp8266.ru/esplorer-ide-esp8266>. ESPlorer отличается от других программ для ESP8266 тем, что:

работает на множестве платформ;

- поддерживает нескольких открытых файлов;
- обеспечивает подсветку кода языков Lua и Python;
- имеет режимы Undo/Redo;
- поддерживает цветовые темы редакторов: dark, Eclipse, IDEA, Visual Studio;

- осуществляет автозавершение кода по нажатию комбинации клавиш <Ctrl>+<Space>;
- обеспечивает «умную» отправку файлов с ожиданием ответа;
- поддерживает несколько прошивок одновременно.

Скачаем и установим Explorer. Запустим, а затем подключим к плате датчик температуры ds18b20 и напишем скрипт на языке Lua для получения данных с этого датчика.



В прошивке NodeMCU присутствуют модули датчиков, которые можно подключать к скриптам командой `require()`. Количество таких модулей постоянно увеличивается, и они доступны для просмотра и загрузки на странице https://github.com/nodemcu/nodemcu-firmware/tree/master/luam_modules.

В листинге приведен пример скрипта для подключения модуля датчика температуры DS18B20, получения и вывода его значений.

```
-- подключение модуля
t = require("ds18b20")
-- GPIO карта модуля ESP-01
gpio0 = 3
gpio2 = 4
t.setup(gpio0)
addrs = t.addrs()
if (addrs ~= nil) then
    print("Total DS18B20 sensors: "..table.getn(addrs))
end
-- чтение температуры
```

```

print("Temperature: "..t.read().."C")
-- освободить память после использования
t = nil
ds18b20 = nil
package.loaded["ds18b20"]=nil

```

Результат выполнения этого скрипта при подключении датчика температуры к выводу GPIO0 приведен на рисунке.

The screenshot shows the ESP8266 Explorer IDE interface. The left pane displays the source code for a Lua script named 'ds18b20-1.lua'. The right pane shows the terminal output of the script's execution. The code in the left pane includes comments in Russian and Lua code to initialize the GPIO pins, read the temperature from a DS18B20 sensor, and free memory. The terminal output on the right shows the execution steps, including the detection of 0 sensors and the final temperature reading of 21.3753750°C.

```

-- GPIO карта модуля ESP-01
> gpio0 = 3
> gpio2 = 4
> t.setup(gpio0)
> addr = t.addr()
> if (addr ~= nil) then
>> print("Total DS18B20 sensors: "..table.getn(addr)
>> end
Total DS18B20 sensors: 0
-- чтение температуры
> print("Temperature: "..t.read().."C")
Temperature: 21.3753750C
-- освободить память после использования
> t = nil
> ds18b20 = nil
> package.loaded["ds18b20"]=nil

```

В несколько строк на Lua можно написать и скрипт сервера, что позволяет использовать эту плату для быстрого создания устройств IoT ("Интернет вещей"). Если вам больше привычна среда программирования Arduino, можно программировать в среде Arduino IDE (см. установку и настройку Arduino IDE для программирования плат ESP8266 в этом [обзоре](#)).