

## 3.1. Лабораторная работа: Начинаем работу с Wi-Fi модулем

Site: [Samsung Innovation Campus](#)

Course: Internet of Things

Book: 3.1. Лабораторная работа: Начинаем работу с Wi-Fi модулем

Printed by: Антон Файтельсон

Date: Saturday, 21 October 2023, 7:36 PM

## Table of contents

3.1.1. Введение

3.1.2. Коммуникация с ESP8266 через последовательный порт

3.1.3. Обновление прошивки ESP8266

3.1.4. Примеры AT-команд

### 3.1.1. Введение

ESP8266 - это самая популярная и недорогая плата WiFi для начинающих. Появившись в 2014 году, она взорвала рынок своей сверх-низкой ценой и её начали встраивать абсолютно везде. Мы будем ее использовать как учебный пример модуля WiFi - это не значит, что вы будете гарантированно встраивать ее в дальнейшем в свои решения. Какой конкретно чип использовать - выбирается под задачу и ограничения.

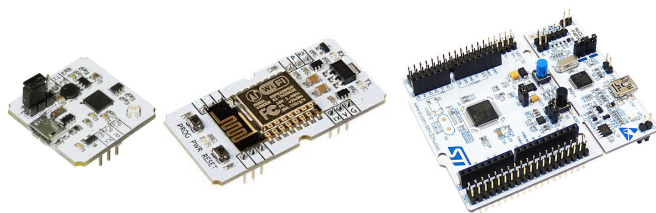
Для ESP8266 существует свой собственный SDK и ее тоже можно программировать, поскольку она представляет собой слабый микроконтроллерный модуль с небольшим количеством свободных выводов и весьма ограниченным набором периферии. Так что минимальные задачи можно выполнять и только средствами одного этого чипа. Но мы не будем погружаться в эти детали, поскольку учебная программа не предполагает изучения множества различных SDK, да это и не нужно. Ведь Wi-Fi-модуль можно использовать и как просто интерфейс связи, пользуясь его встроенными AT-командами. Сегодня мы получим необходимые начальные навыки работы с ESP8266 и посмотрим, как использовать ее в связке с внешним микроконтроллером.

План всех лабораторных на эту тему:

1. Подключаемся к ESP8266 и общаемся с ней через терминал
2. Проверяем актуальность прошивки и обновляем ее при необходимости
3. Соединяем ESP8266 с STM32Nucleo, проверяем соединение
4. Смотрим пример WiFi под Mbed
5. Смотрим пример Sockets под Mbed

#### Необходимое оборудование

- Плата STM32Nucleo
- Модуль связи WiFi ESP8266
- Переходник USB-UART
- WiFi-точка доступа (роутер)
- Соединительные провода (4 штуки)



В данном примере показана работа на примере модулей производства "Амперки", но всё то же самое можно сделать и на любом другом оборудовании.

### 3.1.2. Коммуникация с ESP8266 через последовательный порт

Для начала мы попробуем без всяких посредников пообщаться с платой напрямую. Мы передадим ей несколько простых AT-команд и увидим, что плата действительно подключается к WiFi-точке доступа.

Подключите ESP8266 к USB-UART переходнику, соединив 4 провода:

ESP8266	Переходник USB-UART
TX	RX
RX	TX
V	3.3V
GND	GND

Запустите терминал на скорости 115200. Не обращайте внимания, что в терминале появится “мусор” - почему-то плата его выдает при каждом старте.

Введите вначале команду AT - в ответ получите ответ OK. Если ошибетесь и введете несуществующую команду, плата выдаст ответ ERROR. Каждую команду завершайте следующими символами: возврат каретки (CTRL+M) и перенос строки (CTRL+J). Это те самые, которые в ASCII имеют коды \n и \r.

Затем введите AT+GMR - получите информацию о плате.

```
AT
OK
AT + GMR

ERROR
AT+GMR
AT version:1.2.0.0(Jul  1 2016 20:04:45)
SDK version:1.5.4.1(39cb9a32)
Ai-Thinker Tehnology Co. Ltd.
Dec  2 2016 14:21:16
OK
█
```

Как видите, в данном примере версия прошивки от 2016 года, номер версии AT 1.2.0.0. В то же время, самая свежая версия на сайте производителя называется ESP8266 AT Bin V1.7.4.0 и она от 2020 года. В зависимости от этой проверки, пропустите или выполните следующий раздел.

### 3.1.3. Обновление прошивки ESP8266

Все нижеизложенное - небольшой пересказ руководства: <https://os.mbed.com/teams/ESP8266/wiki/Firmware-Update>

Что нужно подготовить из программного обеспечения:

- Самая новая версия прошивки - перейдите на [сайт](#) и скачайте ее. На приведенной картинке самая новая прошивка - это 1.7.5, но к моменту, как вы будете это читать, возможно, уже выйдет новая версия



AT

Found 33 results

Expand all +

Collapse all -

Title	Platform	Version	Release Date	Download
+ ESP8266 NonOS AT Bin	Bin	Deprecated	2021.12.28	↓
+ ESP8266 NonOS AT Bin V1.7.5	Bin	V1.7.5	2021.10.18	↓
+ ESP32-C3 AT Bin	Bin	Latest	2021.08.12	↓
+ ESP32 AT Bin	Bin	Latest	2021.07.05	↓
+ ESP32-S2 AT Bin	Bin	Latest	2020.08.21	↓

Прошивку со словом IDF в названии лучше не трогать, хотя у нее и указана более свежая версия - она не подходит для большинства отладочных плат на основе ESP, так как в ней используются разные выводы для прошивки и для коммуникации

- Для прошивки вам понадобится программа <https://github.com/espressif/esptool>, установите ее: **pip install esptool**.

Из аппаратного обеспечения у вас должна быть плата ESP8266, подключенная к компьютеру через USB-UART переходник, и корректно отвечающая на AT-команды из предыдущего раздела.

Для начала проверьте в утилите esptool, обнаруживается ли ваша плата. Переведите плату в режим программирования удержанием кнопки PROG и однократным нажатием RESET (удерживая при этом кнопку PROG).

```
esptool.py --port /dev/ttyUSB0 --baud 115200 flash_id
```

Порт может быть другой - проверьте в /dev/, как именно он называется. Если всё правильно, то выдача этой программы будет такая:

```
esptool.py v2.8
Serial port /dev/ttyUSB0
Connecting...
Detecting chip type... ESP8266
Chip is ESP8266EX
Features: WiFi
Crystal is 26MHz
MAC: 2c:f4:32:81:6a:00
Uploading stub...
Running stub...
Stub running...
Manufacturer: 20
Device: 4016
Detected flash size: 4MB
```

Программа корректно определила чип и теперь можно загружать прошивку. Опять переведите плату в режим программирования, перейдите в папку с прошивкой.

Здесь немного отступление в сторону, почему мы выбираем тот или иной вариант прошивки? Во-первых, есть 2 варианта собранной прошивки: на 1Мб (урезанная, для слабых устройств) и на 2Мб. Мы выбираем второй, поскольку у нас как раз плата с 2Мб памяти. Во-вторых, там есть деление на AT-версию (то, что нужно нам), и на AT+SDIO (для использования линий ввода-вывода). Второе нам не нужно, поэтому берем вариант AT, на 2 Мб. В Readme-файле есть инструкции, какие файлы и по каким адресам загружать.

Итак, выполните:

```
esptool.py --port /dev/ttyUSB0 --baud 115200 write_flash --flash_size 2MB-c1 \
0x00000 boot_v1.7.bin 0x1000 at/1024*1024/user1.2048.new:5.bin 0x1FC000 esp_init_data_default_v08.bin \
0x7E000 blank.bin 0x1FE000 blank.bin 0x1FB000 blank.bin
```

Лог будет выглядеть так:

```
esptool.py v2.8
Serial port /dev/ttyUSB0
Connecting...
Detecting chip type... ESP8266
Chip is ESP8266EX
Features: WiFi
Crystal is 26MHz
MAC: 2c:f4:32:81:6a:00
Uploading stub...
Running stub...
Stub running...
Configuring flash size...
Flash params set to 0x0050
Compressed 4080 bytes to 2936...
Wrote 4080 bytes (2936 compressed) at 0x00000000 in 0.3 seconds (effective 123.0 kbit/s)...
Hash of data verified.
Compressed 413444 bytes to 296955...
Wrote 413444 bytes (296955 compressed) at 0x00001000 in 26.4 seconds (effective 125.5 kbit/s)...
Hash of data verified.
Compressed 128 bytes to 75...
Wrote 128 bytes (75 compressed) at 0x001fc000 in 0.0 seconds (effective 84.0 kbit/s)...
Hash of data verified.
Compressed 4096 bytes to 26...
Wrote 4096 bytes (26 compressed) at 0x0007e000 in 0.0 seconds (effective 4703.9 kbit/s)...
Hash of data verified.
Compressed 4096 bytes to 26...
Wrote 4096 bytes (26 compressed) at 0x001fe000 in 0.0 seconds (effective 4671.8 kbit/s)...
Hash of data verified.
Compressed 4096 bytes to 26...
Wrote 4096 bytes (26 compressed) at 0x001fb000 in 0.0 seconds (effective 4640.7 kbit/s)...
Hash of data verified.
```

```
Leaving...
Hard resetting via RTS pin...
```

После чего перезагрузите плату и посмотрите в консоли, какая там теперь версия. Должно быть примерно так:

```
AT+GMR
AT version:1.7.4.0(May 11 2020 19:13:04)
SDK version:3.0.4(9532ceb)
compile time:May 27 2020 10:12:22
Bin version(Wroom 02):1.7.4
OK
```

Дата прошивки должна совпадать с актуальной на сайте.

### 3.1.4. Примеры AT-команд

Чтобы немного освоиться с логикой работы WiFi-модуля, давайте сделаем следующее: через AT-команды переведем плату в режим клиента (по умолчанию она находится в режиме точки доступа), и попробуем подключиться к одной из сетей.

Команда AT+CWMODE касается режима работы платы. Если ввести ее со знаком вопроса в конце - AT+CWMODE? - то выведется, в каком режиме плата находится сейчас:

```
AT+CWMODE?  
+CWMODE:2
```

Режимов всего 3:

1 - Режим клиента

2 - Режим точки доступа

3 - Двойной режим: и клиент, и точка доступа

Мы видим, что по умолчанию она сейчас в режиме точки доступа. Нам в настоящий момент нужен режим клиента, поэтому выберем режим 1, командой AT+CWMODE=1:

```
AT+CWMODE=1  
OK
```

Команда AT+CWLAP выведет листинг имеющихся точек доступа:

```
AT+CWLAP  
+CWLAP: (3, "MGTS_GECN_6736", -92, "d4:0:e3:21:94:7a", 1, 0, 0, 4, 4, 7, 1)  
+CWLAP: (4, "OnLine_10", -35, "7c:39:53:8:a4:06", 7, 8, 0, 5, 3, 7, 1)  
+CWLAP: (3, "RT-WiFi_8", -91, "80:26:89:19:2c:3a", 11, 25, 0, 4, 4, 7, 1)
```

Эта команда не будет срабатывать, если плата находится в режиме точки доступа (что естественно).

Чтобы подключиться к точке доступа, используйте команду AT+CWJAP, передав в нее в качестве параметров название сети и пароль от нее (всё в кавычках):

```
AT+CWJAP="my-test-wifi", "1234test".
```

Если всё получится, то увидите следующее:

```
AT+CWJAP="OnLine_10", "  
WIFI CONNECTED  
WIFI GOT IP  
OK
```

Теперь плата запомнила эту сеть и будет подключаться к ней каждый раз при старте.

Если хотите, чтобы плата забыла сеть, выдайте ей такую команду:

```
AT+CWQAP
```

В ней буква Q означает Quit (а J в предыдущей команде, как несложно догадаться, означает Join). Сделайте это прямо сейчас, чтобы мы могли проверить, как работает поиск сети и подключение из MBed далее.

Выполнив эту задачу, увидите надпись:

```
WIFI DISCONNECT
```

[Reset user tour on this page](#)