

## An ESP8266 Firmware with On-Module HTTP WEB Server & Local DNS

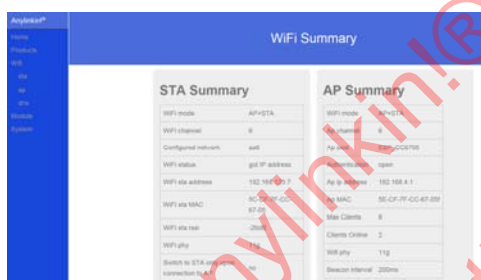
### FEATURES

#### •ESP8266 WIFI AP+STA

- Latest ESP8266® SDK 2.0.0 base
- STA, AP, & STA+AP modes supported
- Supporting SmartConfig & Airkiss

#### •HTTP WEB SERVER On-module

- Accommodated by a HTTPD Server
- Web with HTML, JS, JSON, CGI, etc
- Compatible with most mainstream browsers
- Simple & neat UI for WEB operations



#### •Local Domain System On-module

- A small and fast Local DNS system
- Allowing for max 5 user-defined domain records
- Allowing for fuzzy hit and adaptive and pre-defined hit
- Allowing for domain name hit to AP IP address
- Simple to set your own LAN DNS system

#### •IO resources control via WEB UI

- Allowing for IO resources query and control via the WEB Server UI
- Supporting operations to IO resources of nearly all types of IO control including GPIO, PWM, I<sup>2</sup>C, etc.
- Supporting UART control and communication
- Supporting Flash control and data

dump

- Supporting system info query (such as chip id, mac, flash info, version etc) and control (such as reset, sleep etc)

#### •Configurations Query and Setting via WEB Interface

- Allowing for WIFI configurations query, settings, storage and load of those parameters
- For STA mode, such as AP scanning, selecting or inputting AP SSID and password and connecting to AP
- For AP mode, such as Channel No, Authentication, SSID and password, AP's IP address, Clients lists, etc

#### •Advanced Features

- Captive Portal supported for both Android and IOS access
- Automatic launching web page (e.g. home page) upon STA connecting to the AP of ESP8266 module

#### •Hardware Platform Compatibility

- Compatible with all ESP8266 platforms
- Small Flash size required: >=1MBytes

### APPLICATIONS

#### •Those based on ESP8266 Modules

- 2.4-GHz 802.11b/g/n System
- Home/Building Automation, e.g. Smart Home, Lighting Systems etc
- Industrial Control and Monitoring
- Low-Power Wireless Sensor Networks
- Consumer Electronics
- Health Care

## TABEL OF CONTENTS

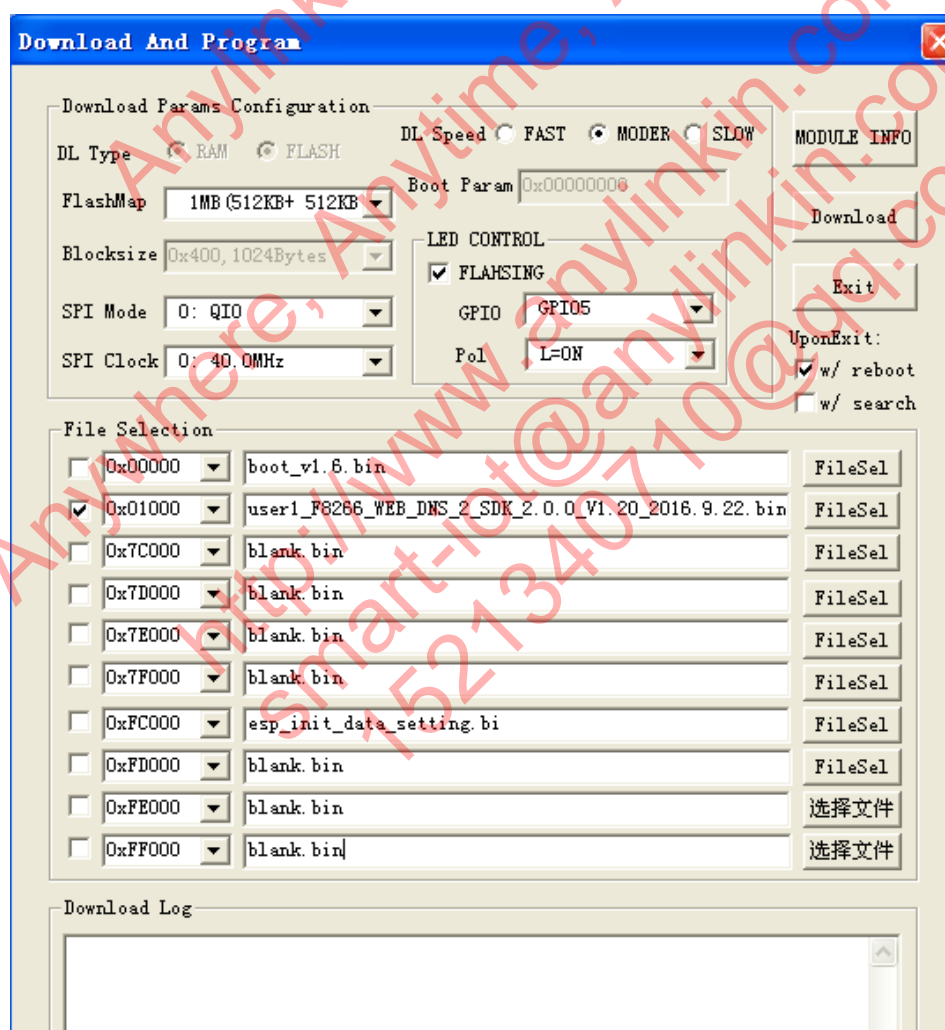
1. DESCRIPTIONS.....	3
2. IMAGE TYPE AND BURNING .....	3
3. OPERATIONS FROM WEB USER INTERFACE .....	4
3.1 OPEN THE SERVER.....	4
3.2 PRODUCTS INFORMATION.....	6
3.3 WIFI OPERATIONS.....	7
3.3.1 WIFI SUMMARY PAGE .....	7
3.3.1.1 WIFI SUMMARY - STA.....	7
3.3.1.2 WIFI SUMMARY - AP.....	7
3.3.2 WIFI AP QUERY AND CONFIGURATION PAGE .....	8
3.3.3 WIFI STA QUERY AND CONFIGURATION PAGE .....	9
3.3.3.1 AP SCANNING .....	9
3.3.3.2 SELECT AP OR INPUT AP INFO TO CONNECT .....	10
3.3.3.3 ACCESS THE WEB SERVER VIA ESP8266 STA MODE .....	11
3.3.4 LOCAL DNS QUERY AND CONFIGURATION PAGE .....	12
3.3.4.1 CUSTOMIZE LOCAL DNS SYSTEM.....	13
3.3.4.2 RULES TO CUSTOMIZE LOCAL DNS SYSTEM.....	14
3.4 MODULE RESOURCES OPERATIONS .....	16
3.4.1 MODULE RESOURCES ENTRY PAGE .....	16
3.4.2 GPIO OPERATIONS .....	17
3.4.3 PWM OPERATIONS.....	18
3.4.4 I2C OPERATIONS.....	19
3.4.5 UART OPERATIONS.....	20
3.5 SYSTEM OPERATIONS.....	21
ORDERING INFORMATION.....	22

## 1. DESCRIPTIONS

The F8266WIFI®-WEB+DNS is a firmware for ESP8266 modules accommodating an on-module HTTP WEB server and an on-module Local Domain System that users could customize. From the WEB user interface, users could perform many operations within or via the ESP8266 modules, such as query, control, and communication.

## 2. IMAGE TYPE AND BURNING

Although only a small 1Mbytes of SPI Flash with Flash Map = 2 (that is 512KB+512KB) will be required for the image, images with different types of Flash Map are also available upon request. Different Flash Map may introduce different flash burning. Please refer to Espressif SDK user manual for details. Here below is an example to burn an image of Flash Map = 2.

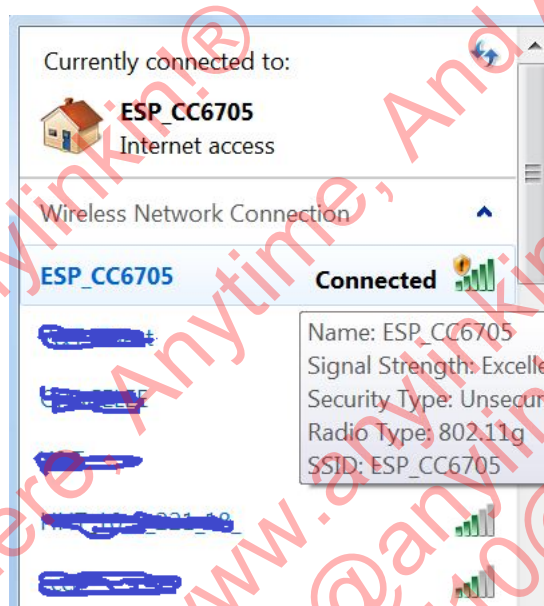


## 3. OPERATIONS FROM WEB USER INTERFACE

### 3.1 OPEN THE SERVER

**Step 1** Power on the ESP8266 Module which has been burnt with the firmware Image, and Connect a STA to the AP with the AP SSID.

Notes 1: If you have not configured the AP parameters before, the default AP SSID is ESP\_XXXXXX, where the XXXXXX is the last 6 digits of the MAC address. See below picture as an example.



The Default AP SSID could be changed from default by user. Please refer to [3.3.2 WIFI AP QUERY AND CONFIGURATION PAGE](#) for details.

**Step 2** Open a browser in the STA to access the on-module WEB server in the ESP8266 AP by input `http://[AP's IP or AP's DNS name]`

Notes 1: If you have not configured the AP parameters before, the default AP's IP Address is 192.168.4.1. That is, you could input <http://192.168.4.1> in the browser URL address to access the home page of the on-module web server.

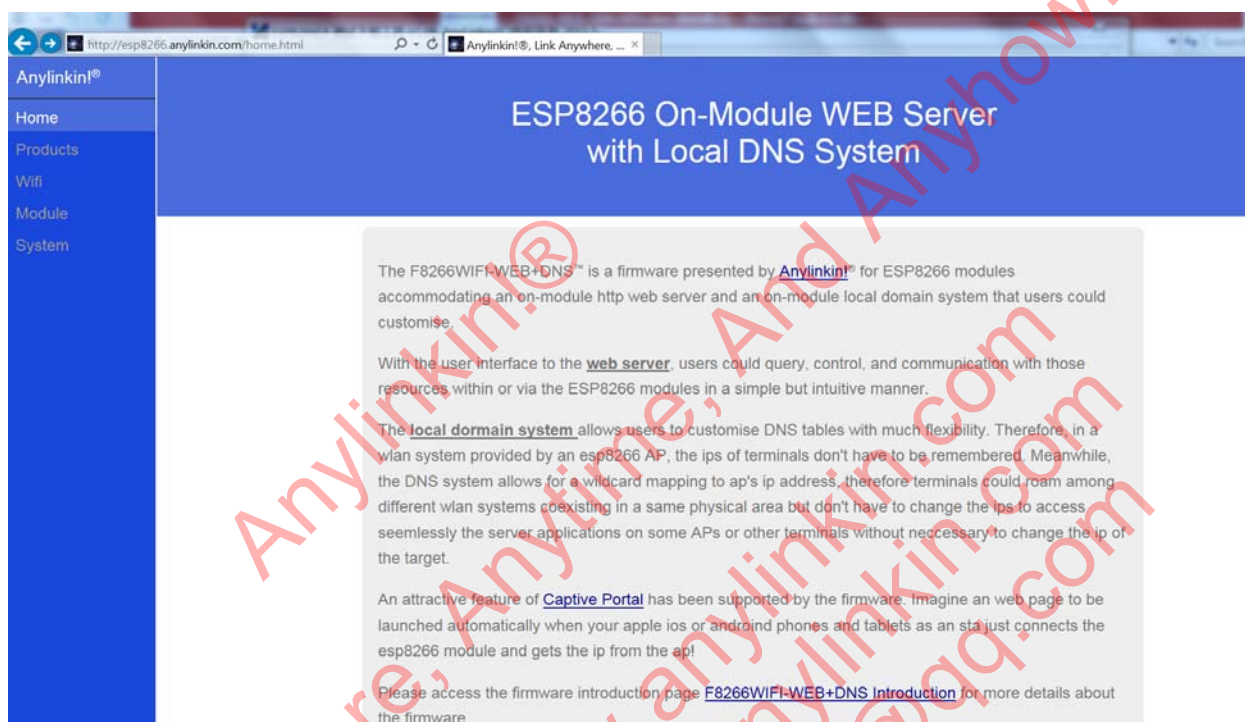
The Default AP's IP Address could be changed from default by user. Please refer to [3.3.2 WIFI AP QUERY AND CONFIGURATION PAGE](#) for details.

Notes 2: If you have not configured the Local DNS system before, there is a default AP DNS Name which is `esp8266.anylinkin.com`. That is, you could input <http://esp8266.anylinkin.com> in the browser URL address to access the

home page of the on-module web server.

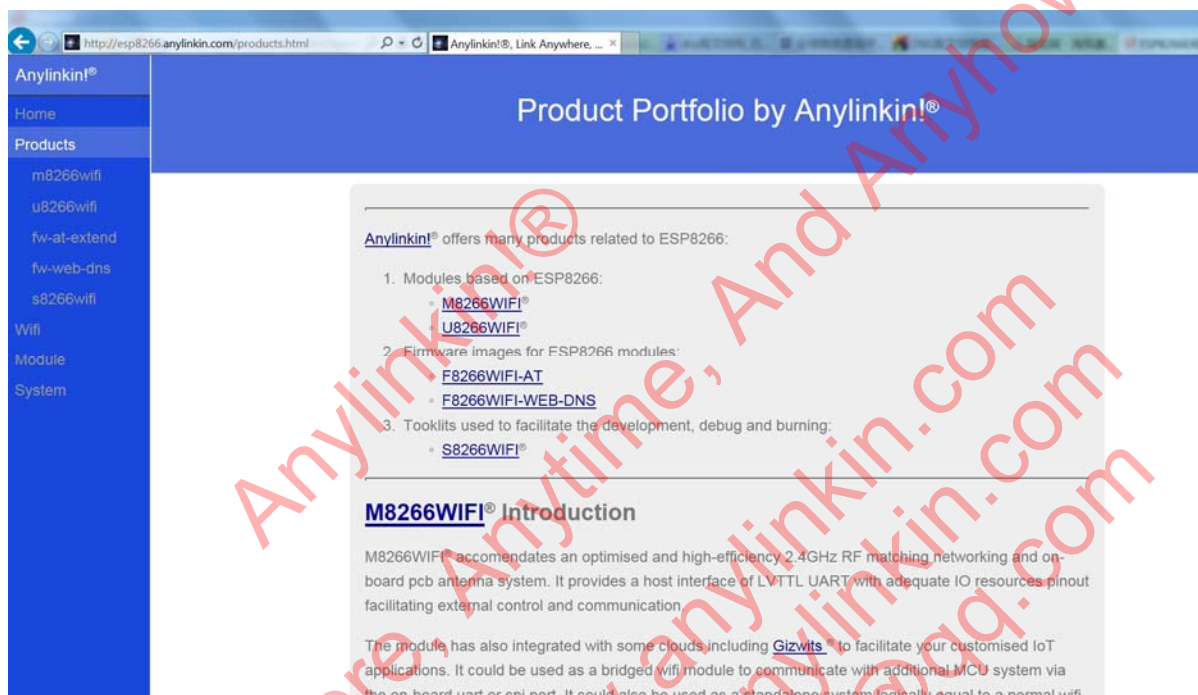
The Default AP DNS Name could be changed from default by user. Please refer to [3.3.2 WIFI AP QUERY AND CONFIGURATION PAGE](#) for details.

**Step 3** The home page of the web server comes out as below



## 3.2 PRODUCTS INFORMATION

If you would feel interested with parts of products presented by [Anylinkin!®](http://www.anylinkin.com), just press the “Products” from the left frame area, the Product Portfolio page will be launched out as below. Be kindly reminded that the features of the web server with local domain system could be found in the page as well.





## 3.3 WIFI OPERATIONS

Wifi operations include wifi information summary, AP and STA configurations.

### 3.3.1 WIFI SUMMARY PAGE

Press the “Wifi” from the left frame area, the WiFi Summary page will be launched out as below. Two columns is listed, one for the information of STA mode of the ESP8266 module, and the other for AP.

STA Summary	
WiFi mode	AP+STA
WiFi channel	6
Configured network	aa6
WiFi status	got IP address
WiFi sta address	192.168.123.7
WiFi sta MAC	5C-CF-7F-CC-67-05
WiFi sta rssi	-26dB
WiFi phy	11g
Switch to STA only upon connection to AP	no

AP Summary	
WiFi mode	AP+STA
Ap channel	6
Ap ssid	ESP_QC6705
Authentication	open
Ap ip address	192.168.4.1
Ap MAC	5E-CF-7F-CC-67-05f
Max Clients	8
Clients Online	2
Wifi phy	11g
Beacon interval	200ms

#### 3.3.1.1 WIFI SUMMARY - STA

#### 3.3.1.2 WIFI SUMMARY - AP

### 3.3.2 WIFI AP QUERY AND CONFIGURATION PAGE

Press the “ap” under the “Wifi” from the left frame area, the AP Configuration page will be launched out as below.

**AP Configuration**

**AP Setting**

WiFi mode	AP+STA
Ap channel	CH6
Authentication	open
Ap ssid	ESP_CC6705
Ap passwd	password
Ap ip address	192.168.4.1
Max Clients	8
Clients	2
Wifi phy	11g
Ap MAC	5E-CF-7F-CC-67-05F
Beacon interval	200 ms
Ap stop upon sta connected	no

**STA Clients**

No.	MAC Addr	IP Addr
1	CC-3D-...	192.168.4.2
2	F4-8B-...	192.168.4.3

Refresh

Set

The current configuration of AP mode has been displayed in the left and the connected STA clients are listed in the right. There are two clients connecting to the ESP8266 AP in this example, which are the laptop and mobile we used for testing now.

Change the values in the left table, and press Set, the AP parameters could be updated and valid at once and saved inside the ESP8266 module which could be loaded next time upon boot-up. Here, you may change the authentication, SSID, password, AP' IP, Channel, etc.

Kindly reminded that, since ESP8266 share the RF channel for both AP and STA when in STA+AP mode and STA's channel predominate the AP's one, therefore when the module connects to other AP or router(which means that the channel has been selected), the AP channel will be the same with STA channel and can't be set. However, your setting value



will still be saved into the Flash of the ESP8266 module and be loaded next time.

### 3.3.3 WIFI STA QUERY AND CONFIGURATION PAGE

In the [3.3.2 WIFI AP QUERY AND CONFIGURATION PAGE](#), we connect our laptop or mobile to the web server via the AP IP address (or domain name) when the ESP8266 functions as the AP. If you expect to access the on-module web server via the STA IP address of the ESP8266 module, you will have to connect the ESP8266 as an STA to another AP or router at first. Meanwhile, you will have to connect the laptop or mobile to that AP or router as well so that they are in a same WLAN with the ESP8266 module.

Press the “sta” under the “Wifi” from the left frame area, the STA Configuration page will be launched out as below.

#### 3.3.3.1 AP SCANNING

Initially, the ESP8266 module of STA mode will start to scan the Aps nearby and the page will be displayed as below which indicating the module is scanning Aps nearby.



Once the scanning gets completed, the APs scanned will be listed as below

Anylinkin!®

Home

Products

Wifi

sta

ap

dns

Module

System

STA Configuration

### Wifi Association

To connect to a WiFi network, please select one of the detected networks, enter the password, and hit the connect button...

Network SSID

- ☐ -81dB
- ☐ -83dB
- ☐ -75dB
- ☐ -82dB
- ☐ -90dB
- ☐ -79dB
- ☐ -91dB
- ☐ -19dB aa6

Kindly noticed that there is an AP called “aa6”, which is the testing router we are using and to which the ESP8266 module in AP+STA mode has connected as an STA.

### 3.3.3.2 SELECT AP OR INPUT AP INFO TO CONNECT

In the bottom of the scanned ap list, there are additional two input Editboxes of SSID and password. In order to connecting the ESP8266 module in STA mode to an AP, you may select one the the scanned Aps, or you may input other SSID in the SSID EditBox. And input the password in the password EditBox, then hit the button *Connect* to connect ESP8266 module to the selected APs or routers.

☐ Input other SSID...

WiFi password, if applicable:

password

OK, if the ESP8266 module as an STA has connected to another AP or routers and gets an valid IP, it will be displayed in the summary [3.3.1 WIFI SUMMARY PAGE](#). Write down

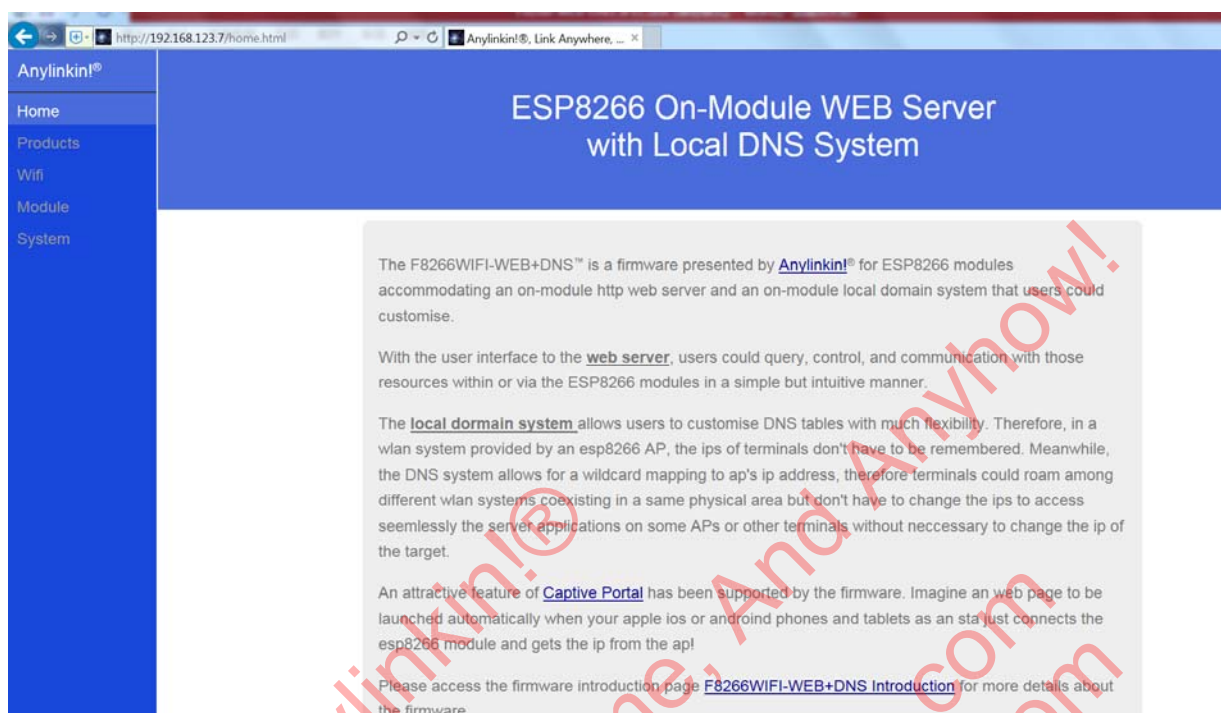
the STA IP address of the module, which will be used to access the web server later. In the snapshot of the example, the ESP8266 as an SAT has joined the WLAN of “aa6” and gets an ip of “192.168.123.7”. We will use this STA IP to access the ESP8266 on-module web server next.

### 3.3.3.3 ACCESS THE WEB SERVER VIA ESP8266 STA MODE

Before that, we need change the testing laptop and mobile to connect the router “aa6” at first, so that the testing laptop, mobile, and the ESP8266 module of STA mode will be in the same WLAN.



Then input <http://192.168.123.7> (i.e. the STA IP Address got from the router) into the address bar of your browser and web page as below will be displayed



Yes, the home page comes out which means that we connect to the on-module web server via the router of “aa6” successfully. Via the STA mode or AP mode to go ahead with following operations make no much difference. And we will go back with operations via the ESP8266 AP mode.

### 3.3.4 LOCAL DNS QUERY AND CONFIGURATION PAGE

Press the “dns” under the “Wifi” from the left frame area, the DNS Configuration page will be launched out as below.



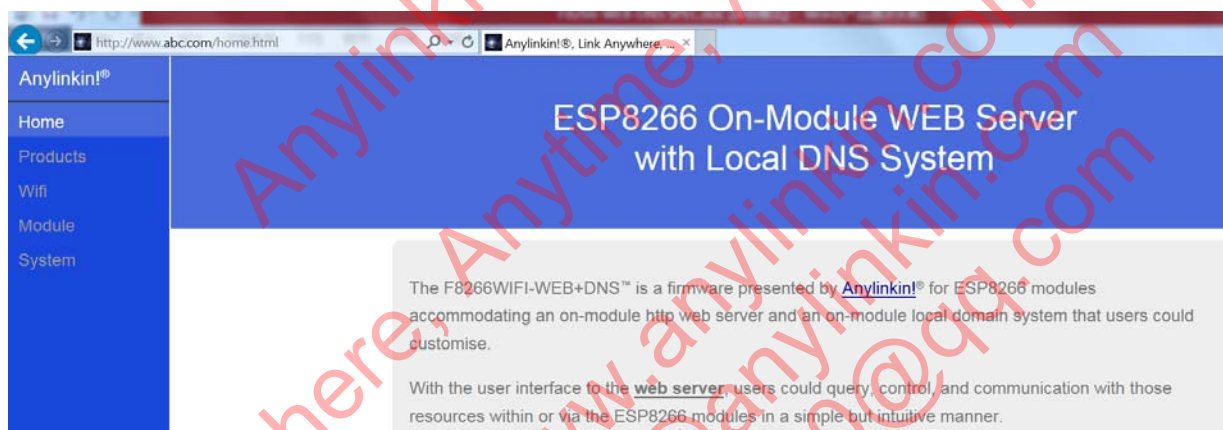
Yes, you will find that there is already a pair of domain name="esp8266.anylinkin.com" and ip="ap", which is the default initial value built into the firmware, which would be useful when you don't know the module AP's IP address. You could access the on-module web server via the same domain name even when the AP's IP address has been changed. Sure, you could change, add, or remove the values here as well.

### 3.3.4.1 CUSTOMIZE LOCAL DNS SYSTEM

As an example, we remove the domain name "esp8266.anylinkin.com" but add two customized domain records as below. Input the value and press "Set":

Domain name	Domain ip
<input type="text" value="www.abc.com"/>	<input type="text" value="ap"/>
<input type="text" value="www.efg.com"/>	<input type="text" value="192.168.4.1"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>

Now access the <http://www.abc.com> and you may find that the home page comes out which means that the just-added domain name has been validated by our on-module local DNS system:



Reboot the ESP8266 module and then browse the DNS page and you will find that the updated values are kept.

### 3.3.4.2 RULES TO CUSTOMIZE LOCAL DNS SYSTEM

Also kindly noticed that:

- The record in a row is invalid when the name field in that row is blank.
- However, if the name field is not blank while the ip field is blank or filled with "ap", then the domain name will be parsed as the AP's IP address of the ESP8266 module.

This feature would be helpful. In many case the module AP IP address may not be fixed or predefined or unique the same. For example, your system may have several AP provided the ESP8266 modules and STAs are grouped to connect different APs for purpose and the AP IP address should be different. However, you expect to access different web server in different WLAN without need to change the AP IP address when a STA roams. You may define a same domain name which will parse to the



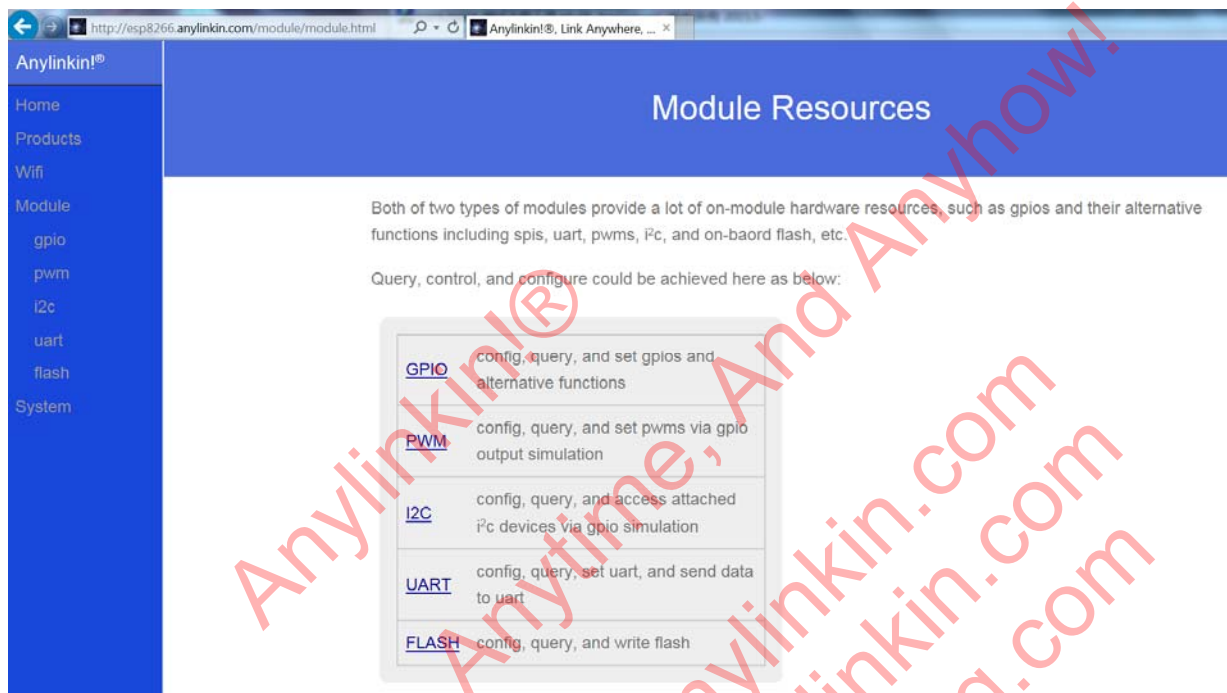
corresponding AP' IP address (by inputting "AP" in the ip field) in each ESP8266 modules which provides the WLANs, then the STA will automatically connecting the web server via the corresponding AP's IP address upon roaming.

Now try to access the <http://esp8266.anylinkin.com> again, you will get a failure page notice since the esp8266.anylinkin.com has been removed. (kindly reminded that some buffers of dns data in the testing laptop may delay to be flushed and possible you may have to reboot the laptop or mobile to ensure the previously successful domain records to be invalid:p).

The local domain system are not definitely be used under the ESP8266 AP mode, or only related to the AP's IP address. Like the web server, it could be valid when the ESP8266 module is in STA mode. And we could access the on-module DNS system via the STA IP address of the module as well. However, if you uses the ESP8266 module in STA mode, then, you may have to set the DNS searching address of those STAs who uses the domain system to the STA IP address of the ESP8266 module who contains the DNS system. Normally this setting should be added inside another APs or routers to which the ESP8266 module of STA mode, and STAs connected. Since in most case STAs will get the DNS search ip address from the AP including the AP's IP address, therefore when to access the local dns system on ESP8266 module in AP mode, you don't have to set the DNS searching address in your STA.

## 3.4 MODULE RESOURCES OPERATIONS

### 3.4.1 MODULE RESOURCES ENTRY PAGE



### 3.4.2 GPIO OPERATIONS

The esp8266ex provides total 16 gpios which could be configured as normal gpio functions of input or output, and the alternative functions. e.g. the pin#26 for gpio1 could be configured as uart0\_bxd, spi\_cs1, gpio input, gpio output, or clk\_rtc. As per gpio output, normal or open drain output could be selected, while for other function selections, pullup or float are the options.

Press the *GET* button to get a picture of current GPIO settings. If a gpio pin has been configured as normal gpio, the status will be displayed as the current input status read in if it is in input mode, or as the current output level.

Press the *SET* button with the expected options the checkboxes selected to set gpio pins, e.g. select function with *gpio\_in*, mode with *normal*, and status with *low* for gpio5, and then press the *SET* button will output a low level on the pin.

IO#	fun	mode	status
<input type="checkbox"/> IO0	tbd	pullup	
<input type="checkbox"/> IO1	uart0_bxd	float	
<input type="checkbox"/> IO2	uart1_bxd	float	
<input type="checkbox"/> IO3	uart0_rxd	float	
<input type="checkbox"/> IO4	gpio_in	float	high
<input type="checkbox"/> IO5	gpio_out	normal	low
<input type="checkbox"/> IO6	spi_clk	float	
<input type="checkbox"/> IO7	spi_miso	float	
<input type="checkbox"/> IO8	spi_mosi	float	
<input type="checkbox"/> IO9	spi_hold	float	
<input type="checkbox"/> IO10	spi_wp	float	
<input type="checkbox"/> IO11	spi_cs0	float	
<input type="checkbox"/> IO12	i2c_data	pullup	
<input type="checkbox"/> IO13	i2c_bck	pullup	
<input type="checkbox"/> IO14	gpio_in	pullup	low
<input type="checkbox"/> IO15	gpio_in	pullup	low

GET SET

### 3.4.3 PWM OPERATIONS

The esp8266ex soc provides PWM output capability via a hardware timer to control the GPIO output, simulating a PWM output de facto. In the table below, PWM $n$  means a PWM simulated on GPIO $n$ . Max 8 channels supported simultaneously. Meanwhile all these PWMs share a same hardware time which means they share a same period as well.

To query current PWM usage, just press *GET* button. Those PWMs currently used will be labled with an enabled checkbox besides the PWM $n$  together with the duties in percentage.

To config PWM output, just input a period, enable the checkboxes besides the PWM $n$  and input the duty, then press the *SET* button.

The settings will be cleared after a reboot.

**PWMs Status and Config**

Period:  us

PWM#	Duty%
<input type="checkbox"/> PWM0	<input type="text" value="0"/>
<input type="checkbox"/> PWM1	<input type="text" value="0"/>
<input type="checkbox"/> PWM2	<input type="text" value="0"/>
<input type="checkbox"/> PWM3	<input type="text" value="0"/>
<input type="checkbox"/> PWM4	<input type="text" value="0"/>
<input type="checkbox"/> PWM5	<input type="text" value="0"/>
<input type="checkbox"/> PWM6	<input type="text" value="0"/>
<input type="checkbox"/> PWM7	<input type="text" value="0"/>
<input type="checkbox"/> PWM8	<input type="text" value="0"/>
<input type="checkbox"/> PWM9	<input type="text" value="0"/>
<input type="checkbox"/> PWM10	<input type="text" value="0"/>
<input type="checkbox"/> PWM11	<input type="text" value="0"/>
<input type="checkbox"/> PWM12	<input type="text" value="0"/>
<input type="checkbox"/> PWM13	<input type="text" value="0"/>
<input type="checkbox"/> PWM14	<input type="text" value="0"/>
<input type="checkbox"/> PWM15	<input type="text" value="0"/>

### 3.4.4 I2C OPERATIONS

Anylinkin!®  
Link Anywhere, Anytime, And Anyhow!  
<http://www.anylinkin.com>  
[smart-iot@anylinkin.com](mailto:smart-iot@anylinkin.com)  
[1521340710@qq.com](mailto:1521340710@qq.com)

### 3.4.5 UART OPERATIONS

Anylinkin!®  
Link Anywhere, Anytime, And Anyhow!  
<http://www.anylinkin.com>  
[smart-iot@anylinkin.com](mailto:smart-iot@anylinkin.com)  
[1521340710@qq.com](mailto:1521340710@qq.com)



## 3.5 SYSTEM OPERATIONS

Anylinkin!®

Home

Products

Wifi

Module

System

System Settings

### System Information

Chip Id	0x00CC6705
Wifi Mac	5C-CF-7F-CC-67-05
Uart Baud	115200bps
RST Code	normal
Flash Size	2MB
Flash Map	2, 1024KB(512KB+512KB)
SDK Version	2.0.0(656edbf)
App Version	
Compilation	Sep 24 2016 14:50:20
Authorisation	no

Refresh

### System Reset

To reset the system, please select one of the reset type, and then hit the reset button...

ResetType: 

Power On Reset  
nResetPin Reset  
Software Reset  
Watchdog Reset

Reset

### System Sleep

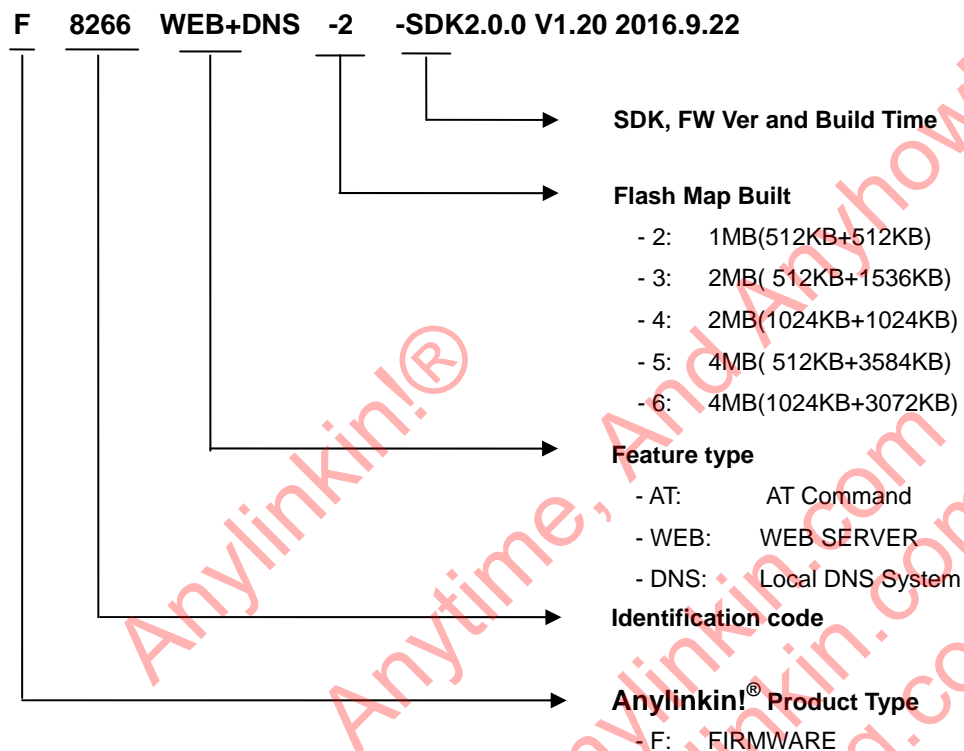
To bring the system into sleep, please select one of the sleep type, input a sleep period, select a wakeup option, and then hit the sleep button...

SleepPeriod:  millicsec(s)

WakeOption:  ☒

Sleep

## ORDERING INFORMATION



### •Purchase

@eBay

<http://www.ebay.com/itm/282191379273>

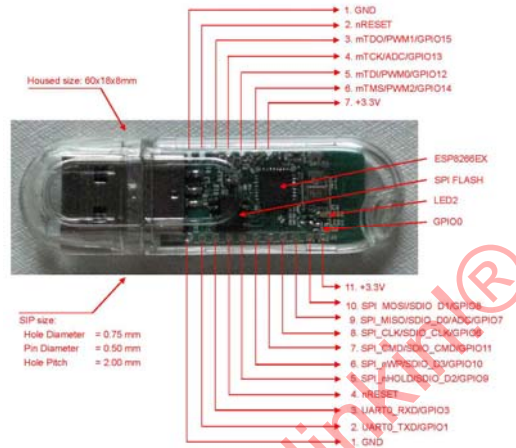
@taobao

<https://item.taobao.com/item.htm?id=523127335413>

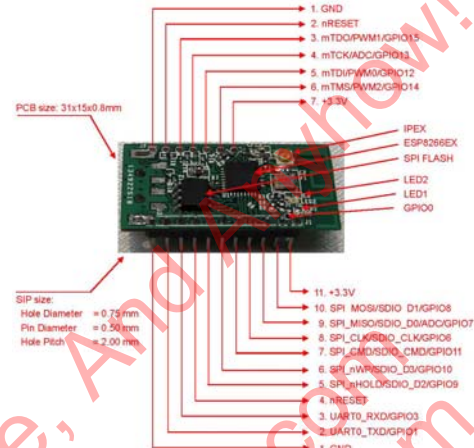


**Welcome** to use the customized One-Press-Programming USB WIFI modules and the debug and download toolkit, which bring about much conveniences redevelopment and burning. Modules could be purchased from the below URLs and more detailed specifications could be provided upon request.

#### ESP8266EX Module, USB Connector



#### ESP8266EX Module, SIP Connector



#### Purchase @ eBay

<http://www.ebay.com/itm/282162857143>



<http://www.ebay.com/itm/282162896341>

