ESP8266 AT Instruction Set

Version 0.23

Espressif Systems IOT Team

Copyright (c) 2015

Disclaimer and Copyright Notice

Information in this document, including URL references, is subject to change without notice.

THIS DOCUMENT IS PROVIDED AS IS WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NON-INFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION OR SAMPLE. All liability, including liability for infringement of any proprietary rights, relating to use of information in this document is disclaimed. No licenses express or implied, by estoppel or otherwise, to any intellectual property rights are granted herein.

The WiFi Alliance Member Logo is a trademark of the WiFi Alliance.

All trade names, trademarks and registered trademarks mentioned in this document are property of their respective owners, and are hereby acknowledged.

Table of Contents

1.   Preambles

This is the documentation for Espressif AT command Command set and usage.

Command set is divided into: Basic AT commands, WiFi function, AT commands, TCP / IP Toolbox AT commands.

Copy all files in folder "at" to folder "app" in esp_iot_sdk to compile.
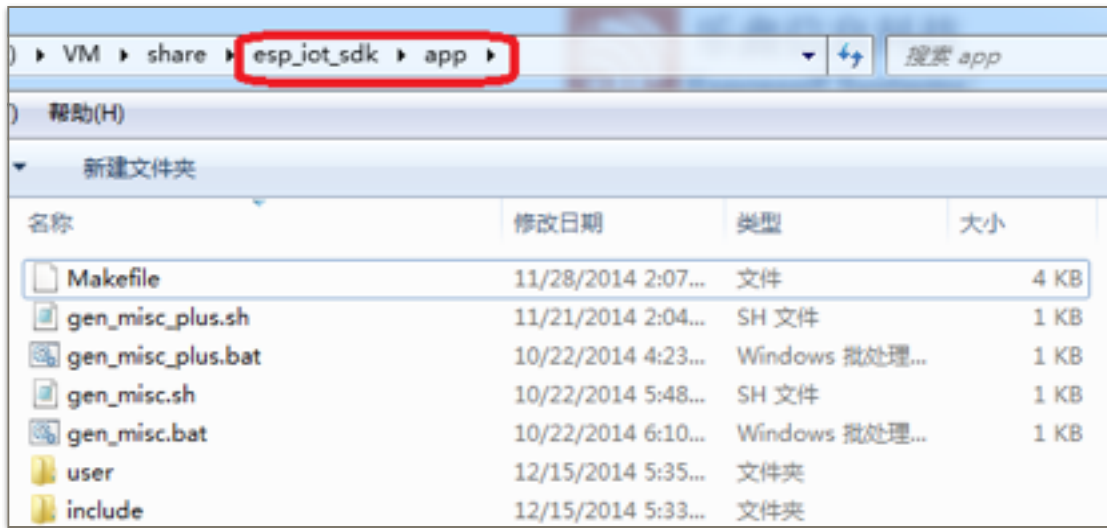
Download:

boot.bin, downloads to flash 0x00000

user1.bin, downloads to flash 0x01000

blank.bin, factory initialize, downloads to flash both 0x7E000 and the last sector but one in flash, e.g. 0xFE000 for 1MB flash

Notes:

• Please make sure that correct BIN (/esp_iot_sdk/bin/at) is already in the chip (ESP8266) before the AT commands listed in this documentation can be used.

• AT added some functions so AT bin size is larger than before, please 1024KB(or larger) flash

• If you need to compile your own AT, please select 1024KB or larger option in STEP 5

• How to use 1024KB(or larger) flash, please refer to BBS http://bbs.espressif.com/viewtopic.php?f=10&t=305

2. Command DescrIPtion

Each Command set contains four types of AT commands.

Type	Command Format	DescrIPtion
Test	AT+<x>=?	Query the Set command or internal parameters and its range values.
Query	AT+<x>?	Returns the current value of the parameter.
Set	AT+<x>=<…>	Set the value of user-defined parameters in commands and run.
Execute	AT+<x>	Runs commands with no user-defined parameters.

Notes:

1. Not all AT Command has four commands.

2. [] = default value, not required or may not appear

3. String values require double quotation marks, for example:

   AT+CWSAP="ESP756290","21030826",1,4

4. Baudrate = 115200

5. AT Commands has to be capitalized, and end with "/r/n"
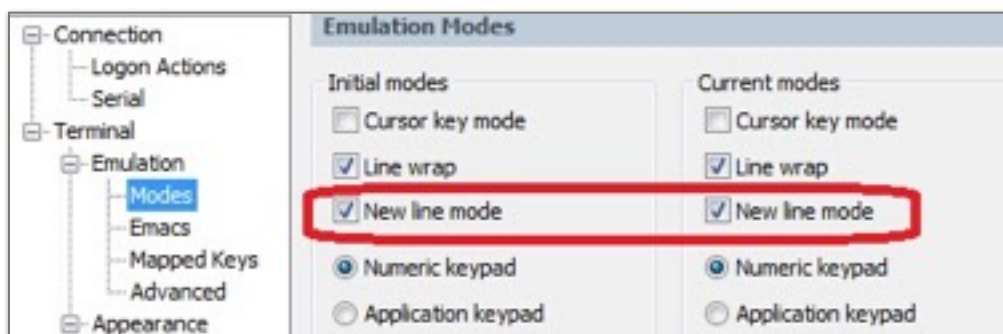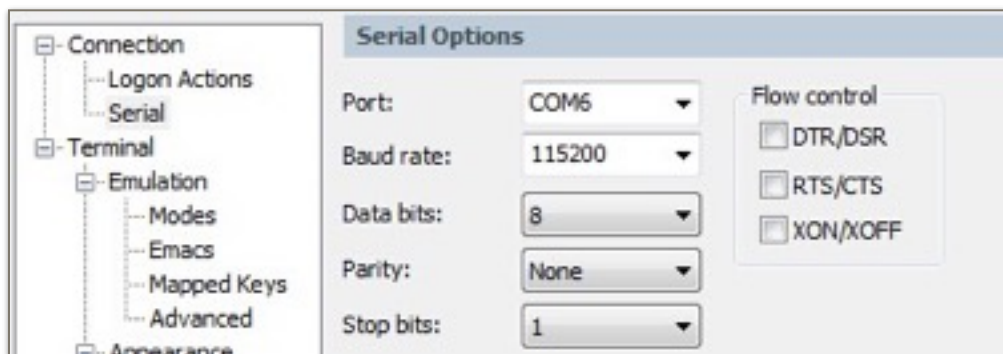
3. Basic AT Command Set

3.1. Overview

The ESP8266 wireless WiFi modules can be driven via the serial interface using the standard AT commands. Here is a list of some basic AT commands that can be used.

Basic	
Command	**DescrIPtion**
AT	Test AT startup
AT+RST	Restart module
AT+GMR	View version info
AT+GSLP	Enter deep-sleep mode
ATE	AT commands echo or not
AT+RESTORE	Factory Reset
AT+UART	UART configuration, [@deprecated]
AT+UART_CUR	UART current configuration
AT+UART_DEF	UART default configuration, save to flash
AT+SLEEP	Sleep mode

3.2. Commands

1. AT – Test AT startup

The type of this command is "executed". It's used to test the setup function of your wireless WiFi module.

AT – Test AT startup	
Response	OK
Parameters	null

2. AT+RST – Restart module

The type of this command is "executed". It's used to restart the module.

AT+RST - Restart module	
Response	`OK`
Parameters	`null`

3. AT+GMR – View version info

This AT command is used to check the version of AT commands and SDK that you are using, the type of which is "executed". The version information is made up of a string of numbers, the first four numbers refer to the version of AT commands, while the last four refer to the version of SDK.

AT+GMR - View version info	
Response	`<number>` `OK`
Parameters	`< number >  version info, length: 8 bytes`
Notes	`For example, response is 0017xxxx, then 0017 refer to the version of AT commands.`

4. AT+GSLP – Enter deep-sleep mode

This command is used to invoke the deep-sleep mode of the module, the type of which is "set". A minor adjustment has to be made before the module enter this deep sleep mode, i.e., connect XPD_DCDC with EXT_RSTB via **OR**.

AT+GSLP=<time>	
Response	`<time>` `OK`
Parameters	`The time unit of < time >  is ms. ESP8266 will wake up after X ms and then enter the deep sleep mode.`

5. ATE – AT commands echo

This command ATE is an AT trigger command echo. It means that entered commands can be echoed back to the sender when ATE command is used. Two parameters are possible. The command returns "OK" in normal cases and "ERROR" when a parameter other than 0 or 1 was specified.

ATE – AT commands echo	
Response	OK
Parameters	ATE0: Switch echo off ATE1: Switch echo on

6.　AT+RESTORE – Factory reset

This command is used to restore the factory default settings of the module. The chip will be restarted when this command is executed.

AT+RESTORE – Factory reset	
Response	OK
Notes	Restore factory default settings. The chIP will restart.

7.　AT+UART – UART configuration

This command sets the UART configuration and writes the new configuration to the flash. It is stored as the default parameter and will also be used as the default baudrate henceforth. [THIS API IS DEPRECATED.]

AT+UART=<baudrate>,<databits>,<stopbits>,<parity>,<flow control> This API is deprecated, please use AT+UART_CUR or AT+UART_DEF instead.		
Example	AT+UART=115200,8,1,0,3	
Response	OK	
Parameters	<baudrate>	Baudrate range: 110 to 115200*40 (4.608 Mega)
	<databits>	5: 5 bits data 6: 6 bits data 7: 7 bits data 8: 8 bits data
	<stopbits>	1: 1 bit stop bit 2: 1.5 bit stop bit 3: 2 bit stop bit
	<parity>	0: None 1: Odd 2: EVEN
	<flow control>	0: disable flow control 1: enable RTS 2: enable CTS 3: enable both RTS and CTS

AT+UART=\<baudrate>,\<databits>,\<stopbits>,\<parity>,\<flow control>		
This API is deprecated, please use AT+UART_CUR or AT+UART_DEF instead.		
Notes	• This configuration will also store the baud rate as the default rate in the user parameter area in the Flash for boot up. • Flow control needs hardware support: MTCK is UART0 CTS and MTDO is UART0 RTS.	

8. AT+UART_CUR – current UART configuration

This command sets the current UART configuration; it does not write to the flash device. Hence there is no change in the default baudrate.

AT+UART_CUR=\<baudrate>,\<databits>,\<stopbits>,\<parity>,\<flow control>		
Example	AT+UART_CUR=115200,8,1,0,3	
Response	OK	
Parameters	\<baudrate>	Baudrate range: 110 to 115200*40 (4.608 Mega)
	\<databits>	5: 5 bits data 6: 6 bits data 7: 7 bits data 8: 8 bits data
	\<stopbits>	1: 1 bit stop bit 2: 1.5 bit stop bit 3: 2 bit stop bit
	\<parity>	0: None 1: Odd 2: EVEN
	\<flow control>	0: disable flow control 1: enable RTS 2: enable CTS 3: enable both RTS and CTS
Notes	• This configuration will **NOT** be stored in the flash; unlike the AT+UART command. • Flow control needs hardware support: MTCK is UART0 CTS and MTDO is UART0 RTS.	

9.   AT+UART_DEF – default UART configuration

This command sets the UART configuration and save it to flash. It is stored as the default parameter and will also be used as the default baudrate henceforth.

AT+UART_DEF=<baudrate>,<databits>,<stopbits>,<parity>,<flow control>		
Example	AT+UART_DEF=115200,8,1,0,3	
Response	OK	
Parameters	<baudrate>	Baudrate range: 110 to 115200*40 (4.608 Mega)
	<databits>	5: 5 bits data 6: 6 bits data 7: 7 bits data 8: 8 bits data
	<stopbits>	1: 1 bit stop bit 2: 1.5 bit stop bit 3: 2 bit stop bit
	<parity>	0: None 1: Odd 2: EVEN
	<flow control>	0: disable flow control 1: enable RTS 2: enable CTS 3: enable both RTS and CTS
Notes	• This configuration will be stored in the flash user parameter area. • Flow control needs hardware support: MTCK is UART0 CTS and MTDO is UART0 RTS.	

10. AT+SLEEP – sleep mode

This command sets ESP8266 sleep mode. It can only be used in station mode，default to be modem-sleep mode .

AT+SLEEP – sleep mode	
Command	**AT+SLEEP?**
Response	+SLEEP:<sleep mode> OK
Parameters	<sleep mode> 0 means disable sleep mode <sleep mode> 1 means light-sleep mode <sleep mode> 2 means modem-sleep mode
Command	**AT+SLEEP=<sleep mode>**
Response	OK
Parameters	The same as above.

4.   WiFi Functions Overview

WiFi Connectivity Functions Invoked by AT commands	
Command	**DescrIPtion**
AT+CWMODE	WIFI mode（sta/AP/sta+AP），[@deprecated]
AT+CWMODE_CUR	WIFI mode（sta/AP/sta+AP） Won't save to Flash
AT+CWMODE_DEF	WIFI default mode（sta/AP/sta+AP） Save to Flash
AT+CWJAP	Connect to AP, [@deprecated]
AT+CWJAP_CUR	Connect to AP, won't save to Flash
AT+CWJAP_DEF	Connect to AP, save to Flash
AT+CWLAP	Lists available APs
AT+CWQAP	Disconnect from AP
AT+CWSAP	Set configuration of ESP8266 softAP [@deprecated]
AT+CWSAP_CUR	Set configuration of ESP8266 softAP Won't save to Flash.
AT+CWSAP_DEF	Set configuration of ESP8266 softAP Save to Flash.
AT+CWLIF	Get station's IP which is connected to ESP8266 softAP
AT+CWDHCP	Enable/Disable DHCP, [@deprecated]
AT+CWDHCP_CUR	Enable/Disable DHCP, won't save to Flash
AT+CWDHCP_DEF	Enable/Disable DHCP, save to Flash
AT+CWAUTOCONN	Connect to AP automatically when power on
AT+CIPSTAMAC	Set mac address of ESP8266 station [@deprecated]
AT+CIPSTAMAC_CUR	Set mac address of ESP8266 station Won't save to Flash.

AT+CIPSTAMAC_DEF	Set mac address of ESP8266 station Save to Flash.
AT+CIPAPMAC	Set mac address of ESP8266 softAP [@deprecated]
AT+CIPAPMAC_CUR	Set mac address of ESP8266 softAP Won't save to Flash.
AT+CIPAPMAC_DEF	Set mac address of ESP8266 softAP Save to Flash.
AT+CIPSTA	Set IP address of ESP8266 station, [@deprecated]
AT+CIPSTA_CUR	Set IP address of ESP8266 station Won't save to Flash.
AT+CIPSTA_DEF	Set IP address of ESP8266 station Save to Flash.
AT+CIPAP	Set IP address of ESP8266 softAP, [@deprecated]
AT+CIPAP_CUR	Set IP address of ESP8266 softAP Won't save to Flash.
AT+CIPAP_DEF	Set IP address of ESP8266 softAP Save to Flash.

4.1.   Commands

1.   AT+CWMODE – WiFi mode

The function of this AT command is to get the value scope of WiFi mode, including station mode, softAP mode, and station+softAP mode, enquiry about the information of WiFi mode, or set the WiFi mode.

AT+CWMODE – WiFi mode	
[@deprecated]. Please use AT+CWMODE_CUR or AT+CWMODE_DEF instead.	
Command	`AT+CWMODE=?`

Response	`+CWMODE:( value scope of <mode>)`
	
	`OK`

Parameters	`Please refer to AT command settings.`
Command	`AT+CWMODE?`

Response	`+CWMODE:<mode>`
	
	`OK`

Parameters	`Please refer to AT command settings.`
Command	`AT+CWMODE=<mode>`
Response	`OK`

Parameters	`<mode>1 means Station mode`
	`<mode>2 means softAP mode`
	`<mode>3 means AP + Station mode`

Notes	`This setting will be stored in the flash system parameter area. It won't` `be erased even when the power is off and restarted.`

2.   AT+CWMODE_CUR – current WiFi mode

There are three WiFi working modes: Station mode, softAP mode, and the co-existence of Station mode and softAP mode. This command is used to acquire the existing WiFi mode, or to set a customised WiFi mode.

AT+CWMODE_CUR – Set WIFI mode（sta/AP/sta+AP）, won't save to Flash	
Example	**AT+CWMODE_CUR=3**
Command	**AT+CWMODE_CUR=?**
Response	+CWMODE_CUR:(value scope of <mode>) OK
Parameters	<mode>1 means Station mode <mode>2 means softAP mode <mode>3 means AP + Station mode
Command	**AT+CWMODE_CUR?**
Response	+CWMODE_CUR:<mode> OK
Parameters	<mode>1 means Station mode <mode>2 means softAP mode <mode>3 means AP + Station mode
Command	**AT+CWMODE_CUR=<mode>**
Response	OK
Parameters	<mode>1 means Station mode <mode>2 means softAP mode <mode>3 means AP + Station mode
Notes	This configuration will not store in Flash.

3.   AT+CWMODE_DEF – default WiFi mode

AT+CWMODE_DEF - WIFI mode (sta/AP/sta+AP), save to Flash	
Example	AT+CWMODE_DEF=3
Command	**AT+CWMODE_DEF=?**
Response	+CWMODE_DEF:(value scope of <mode>) OK
Parameters	<mode>1 means Station mode <mode>2 means softAP mode <mode>3 means AP + Station mode
Command	**AT+CWMODE_DEF?**
Response	+CWMODE_DEF:<mode> OK
Parameters	<mode>1 means Station mode <mode>2 means softAP mode <mode>3 means AP + Station mode
Command	**AT+CWMODE_DEF=<mode>**
Response	OK
Parameters	<mode>1 means Station mode <mode>2 means softAP mode <mode>3 means AP + Station mode
Notes	This configuration will store in Flash system parameter area.

4.   AT+CWJAP – Connect to AP

AT+CWJAP – Connect to AP	
[@deprecated]. Please use AT+CWJAP_CUR or AT+CWJAP_DEF instead.	

Example	`AT+CWJAP ="abc", "0123456789"`
	`If SSID is "ab/,c" and password is "0123456789"/"`
	`AT+CWJAP ="ab///,c", "0123456789"///"`

Command	`AT+ CWJAP?`

Response	`+ CWJAP:<ssid>`
	
	`OK`

Parameters	`<ssid>  string, AP's SSID`
Command	`AT+ CWJAP=<ssid>,<pwd>`
Response	`OK` `ERROR`

Parameters	`<ssid>  string, AP's SSID`
	`<pwd>  string, MAX: 64 bytes ASCII`
	
	`This command needs station mode enable.`
	`Escape character syntax is needed if "SSID" or "password" contains`
	`any special characters`
	`(',' 、 '"'and'/')`

Notes	`This configuration will store in Flash system parameter area.`

5. AT+CWJAP_CUR – Connect to AP，for current

AT+CWJAP_CUR – Connect to AP, won't save to Flash	
Example	AT+CWJAP_CUR ="abc","0123456789" If SSID is "ab/,c" and password is "0123456789"/" AT+CWJAP_CUR="ab///,c","0123456789/"//"
Command	**AT+CWJAP_CUR?**
Response	+CWJAP_CUR:<ssid> OK
Parameters	<ssid> string, AP's SSID
Command	**AT+CWJAP_CUR=<ssid>,< pwd >**
Response	OK ERROR
Parameters	<ssid> string, AP's SSID <pwd> string, MAX: 64 bytes ASCII This command needs station mode enable. Escape character syntax is needed if "SSID" or "password" contains any special characters (','、'"'and'/')
Notes	This configuration will not store in Flash .

6. AT+CWJAP_DEF – Connect to AP, save as default

AT+CWJAP_DEF - Connect to AP and save AP info to flash	
Example	AT+CWJAP_DEF="abc","0123456789" If SSID is "ab/,c" and password is "0123456789"/" AT+CWJAP_DEF="ab///,c","0123456789"//"
Command	**AT+CWJAP_DEF?**
Response	+ CWJAP_DEF:<ssid> OK
Parameters	<ssid> string, AP's SSID
Command	**AT+ CWJAP_DEF=<ssid>,< pwd >**
Response	OK ERROR
Parameters	<ssid> string, AP's SSID <pwd> string, MAX: 64 bytes ASCII This command needs station mode enable. Escape character syntax is needed if "SSID" or "password" contains any special characters (',',、'"'and'/')
Notes	This configuration will store in Flash system parameter area.

7.   AT+CWLAP – List available APs

AT+CWLAP - Lists available APs	

Example	AT+CWLAP
	    List of all available AP's detected by ESP8266
	AT+CWLAP="wifi","ca:d7:19:d8:a6:44",6
	    Find AP with specific SSID and MAC at specific channel.
	AT+CWLAP="wifi"
	    Find AP with specific SSID

Command	**AT+CWLAP=<ssid>,<mac>,<ch>**
Response	+CWLAP:<ecn>,<ssid>,<rssi>,<mac>,<ch> OK ERROR
Parameters	<ecn> 0 OPEN <ecn> 1 WEP <ecn> 2 WPA_PSK <ecn> 3 WPA2_PSK <ecn> 4 WPA_WPA2_PSK <ssid> string, SSID of AP <rssi> signal strength <mac> string, MAC address
Command	**AT+CWLAP**
Response	+CWLAP:<ecn>,<ssid>,<rssi>,<mac>,<ch> OK ERROR
Parameters	The same as above

8.   AT+CWQAP – Disconnect from AP

AT+CWQAP - Disconnect from AP	
Command	AT+CWQAP=?
Response	OK
Parameters	null
Command	AT+ CWQAP
Response	OK
Parameters	null

9.　AT+CWSAP – Configuration of softAP mode

AT+ CWSAP – Configuration of softAP mode [@deprecated]. Please use AT+CWSAP_CUR or AT+CWSAP_DEF instead.	
Example	AT+CWSAP="ESP8266","1234567890",5,3
Command	**AT+CWSAP?**
Response	+CWSAP:<ssid>,<pwd>,<chl>,<ecn>
Parameters	<ssid> string, ESP8266 softAP' SSID <pwd> string, MAX: 64 bytes ASCII <chl> channel id <ecn> 0 OPEN <ecn> 2 WPA_PSK <ecn> 3 WPA2_PSK <ecn> 4 WPA_WPA2_PSK
Command	**AT+ CWSAP=<ssid>,<pwd>,<chl>,<ecn>**
Response	OK ERROR
Parameters	The same as above.
Notes	This CMD is only available when softAP mode enable, and need to follow by AT+RST to make it works. ESP8266 softAP don't support WEP. This configuration will store in Flash system parameter area.

10. AT+CWSAP_CUR – Current config of softAP mode

AT+CWSAP_CUR - Set configuration of softAP mode, won't save to Flash	
Example	AT+CWSAP_CUR="ESP8266","1234567890",5,3
Command	**AT+CWSAP_CUR?**
Response	+CWSAP_CUR:<ssid>,<pwd>,<chl>,<ecn>
Parameters	<ssid> string, ESP8266 softAP' SSID <pwd> string, MAX: 64 bytes ASCII <chl> channel id < ecn >0 OPEN < ecn >2 WPA_PSK < ecn >3 WPA2_PSK < ecn >4 WPA_WPA2_PSK
Command	**AT+CWSAP_CUR=<ssid>,<pwd>,<chl>,<ecn>**
Response	OK ERROR
Parameters	The same as above.
Notes	This command is only available when softAP mode enable, and need to follow by AT+RST to make it works. ESP8266 softAP don't support WEP. This configuration will not store in Flash.

11. AT+CWSAP_DEF – Default config of softAP mode

AT+ CWSAP_DEF - Set configuration of softAP mode, save to Flash	
Example	AT+CWSAP_DEF="ESP8266","1234567890",5,3
Command	**AT+ CWSAP_DEF?**
Response	+ CWSAP_DEF:<ssid>,<pwd>,<chl>,<ecn>
Parameters	<ssid> string, ESP8266 softAP' SSID <pwd> string, MAX: 64 bytes ASCII <chl>channel id < ecn >0 OPEN < ecn >2 WPA_PSK < ecn >3 WPA2_PSK < ecn >4 WPA_WPA2_PSK
Command	**AT+CWSAP_DEF=<ssid>,<pwd>,<chl>,<ecn>**
Response	OK ERROR
Parameters	The same as above.
Notes	This command is only available when softAP mode enable, and need to follow by AT+RST to make it works. ESP8266 softAP don't support WEP. This configuration will store in Flash system parameter area.

12. AT+CWLIF – IP of stations

This command is used to get the IP of stations that are connected to ESP8266 softAP.

AT+ CWLIF- IP of stations which are connected to ESP8266 softAP	
Response	<IP addr>,<mac> OK
Parameters	<IP addr> IP address of stations which are connected to ESP8266 softAP <mac> mac address of stations which are connected to ESP8266 softAP

13. AT+CWDHCP – Enable/Disable DHCP

AT+ CWDHCP - Enable/Disable DHCP	
[@deprecated]. Please use AT+CWDHCP_CUR or AT+CWDHCP_DEF instead.	
Command	`AT+CWDHCP?`
Response	`DHCP disabled or enabled now?`

Parameters	`Bit0: 0 – softap dhcp disable`
	`       1 – softap dhcp enable`
	`bit1: 0 – station dhcp disable`
	`       1 – station dhcp enable`

Command	`AT+CWDHCP=<mode>,<en>`
Response	`OK`

Parameters	`<mode>`
	`0 : set ESP8266 softAP`
	`1 : set ESP8266 station`
	`2 : set both softAP and station`
	`<en>`
	`0 : Disable DHCP`
	`1 : Enable DHCP`

Notes	`This configuration will store in Flash user parameter area.`
	`This configuration interact with static IP related AT commands(AT`
	`+CIPSTA related and AT+CIPAP related):`
	`If enable DHCP, static IP will be disabled;`
	`If enable static IP, DHCP will be disabled;`
	`This will depends on the last configuration.`

14. AT+CWDHCP_CUR – Enable/Disable DHCP

AT+CWDHCP_CUR – Enable/Disable DHCP, won't save to Flash	
Example	`AT+CWDHCP_CUR=0,1`
Command	**AT+CWDHCP_CUR=<mode>,<en>**
Response	`OK`

Parameters	`<mode>`
	`0 : set ESP8266 softAP`
	`1 : set ESP8266 station`
	`2 : set both softAP and station`
	`<en>`
	`0 : Disable DHCP`
	`1 : Enable DHCP`

Notes	`This configuration will not store in Flash.`
	`This configuration interact with static IP related AT commands(AT+CIPSTA related and AT+CIPAP related):`
	`If enable DHCP, static IP will be disabled;`
	`If enable static IP, DHCP will be disabled;`
	`This will depends on the last configuration.`

15. AT+CWDHCP_DEF – Enable/Disable DHCP and save to Flash

AT+CWDHCP_DEF - Enable/Disable DHCP and save to Flash	
Example	`AT+CWDHCP_CUR=0,1`
Command	**`AT+CWDHCP_DEF=<mode>,<en>`**
Response	`OK`

Parameters	`<mode>`
	`0 : set ESP8266 softAP`
	`1 : set ESP8266 station`
	`2 : set both softAP and station`
	`<en>`
	`0 : Disable DHCP`
	`1 : Enable DHCP`

Notes	This configuration will store in Flash user parameter area. This configuration interacts with static IP related AT commands(AT+CIPSTA related and AT+CIPAP related): If enable DHCP, static IP will be disabled; If enable static IP, DHCP will be disabled; This will depend on the last configuration.

16. AT+CWAUTOCONN – Auto connect to AP or not

AT+CWAUTOCONN - Connect to AP automatically or not	
Example	`AT+CWAUTOCONN=1`
Command	**`AT+CWAUTOCONN=<enable>`**
Response	`OK`

Parameters	`<enable>`
	`0 : do not auto-connect to AP when power on`
	`1 : connect to AP automatically when power on`
	
	`Default is enable, ESP8266 station will connect to AP automatically when power on.`

Notes	This configuration will store in Flash system parameter area.

17. AT+CIPSTAMAC – Set mac address of station

AT+ CIPSTAMAC – Set mac address of ESP8266 station [@deprecated]. Use AT+CIPSTAMAC_CUR or AT+CIPSTAMAC_DEF instead.	
Example	`AT+CIPSTAMAC="18:fe:35:98:d3:7b"`
Command	**AT+CIPSTAMAC?**

Response	`+CIPSTAMAC:<mac>`
	
	`OK`

Parameters	`<mac>` string, mac address of ESP8266 station
Command	**AT+CIPSTAMAC=<mac>**
Response	`OK`
Parameters	`<mac>` string, mac address of ESP8266 station
Notes	`This configuration will store in Flash user parameter area.`

18. AT+CIPSTAMAC_CUR – Set mac address of station

AT+ CIPSTAMAC_CUR – Set mac address of ESP8266 station, won't save to Flash	
Example	`AT+CIPSTAMAC_CUR="18:fe:35:98:d3:7b"`
Command	**AT+CIPSTAMAC_CUR?**

Response	`+CIPSTAMAC_CUR:<mac>`
	
	`OK`

Parameters	`<mac>  string, mac address of ESP8266 station`
Command	**AT+CIPSTAMAC_CUR=<mac>**
Response	`OK`
Parameters	`<mac>  string, mac address of ESP8266 station`
Notes	`This configuration will not store in Flash.`

19. AT+CIPSTAMAC_DEF – Set mac address of station, save as default

AT+ CIPSTAMAC_DEF – Set mac address of ESP8266 station, save to Flash	
Example	`AT+CIPSTAMAC_DEF="18:fe:35:98:d3:7b"`
Command	**AT+CIPSTAMAC_DEF?**

Response	`+CIPSTAMAC_DEF:<mac>`
	
	`OK`

Parameters	`Param description` `<mac>  string, mac address of ESP8266 station`
Command	**AT+CIPSTAMAC_DEF=<mac>**
Response	`OK`
Parameters	`<mac>  string, mac address of ESP8266 station`
Notes	This configuration will store in Flash user parameter area.

20. AT+CIPAPMAC – Set mac address of softAP

AT+ CIPAPMAC – Set mac address of ESP8266 softAP [@deprecated]. Use AT+CIPAPMAC_CUR or AT+CIPAPMAC_DEF instead.	
Example	`AT+CIPAPMAC="1a:fe:36:97:d5:7b"`
Command	**AT+CIPAPMAC?**

Response	`+CIPAPMAC:<mac>`
	
	`OK`

Parameters	`<mac>  string, mac address of ESP8266 softAP`
Command	**AT+CIPAPMAC=<mac>**
Response	`OK`
Parameters	`<mac>  string, mac address of ESP8266 softAP`
Notes	`This configuration will store in Flash user parameter area.`

21. AT+CIPAPMAC_CUR – Set mac address of softAP

AT+CIPAPMAC_CUR – Set mac addr of ESP8266 softAP, won't save to Flash	
Example	`AT+CIPAPMAC_CUR="1a:fe:36:97:d5:7b"`
Command	**AT+CIPAPMAC_CUR?**

Response	`+CIPAPMAC_CUR:<mac>`
	
	`OK`

Parameters	`<mac>   string, mac address of ESP8266 softAP`
Command	**AT+CIPAPMAC_CUR=<mac>**
Response	`OK`
Parameters	`<mac>   string, mac address of ESP8266 softAP`
Notes	`This configuration will `not` store in Flash.`

22. AT+CIPAPMAC_DEF – Set mac address of softAP and save as default

AT+ CIPAPMAC_DEF – Set mac address of ESP8266 softAP, save to Flash	
Example	`AT+CIPAPMAC_DEF="1a:fe:36:97:d5:7b"`
Command	**AT+CIPAPMAC_DEF?**

Response	`+CIPAPMAC_DEF:<mac>`
	
	`OK`

Parameters	`<mac>   string, mac address of ESP8266 softAP`
Command	**AT+CIPAPMAC_DEF=<mac>**
Response	`OK`
Parameters	`<mac>   string, mac address of ESP8266 softAP`
Notes	`This configuration will store in Flash user parameter area.`

23. AT+CIPSTA – Set IP address of station

AT+ CIPSTA – Set IP address of ESP8266 station [@deprecated]. Please use AT+CIPSTA_CUR or AT+CIPSTA_DEF instead.	
Example	AT+CIPSTA="192.168.6.100","192.168.6.1","255.255.255.0"
Command	**AT+CIPSTA?**
Response	+CIPSTA:<IP> OK
Parameters	<IP> string, IP address of ESP8266 station
Command	**AT+CIPSTA=<IP>[,<gateway>,<netmask>]**
Response	OK
Parameters	<IP> string, IP address of ESP8266 station [<gateway>] gateway [<netmask>] netmask
Notes	This configuration will store in Flash user parameter area. This configuration interacts with DHCP related AT commands(AT+CWDHCP related): If enable static IP, DHCP will be disabled; If enable DHCP, static IP will be disabled; This will depend on the last configuration.

24. AT+CIPSTA_CUR – Set IP address of station

AT+CIPSTA_CUR – Set IP address of ESP8266 station, won't save to Flash	
Example	AT+CIPSTA_CUR="192.168.6.100","192.168.6.1","255.255.255.0"
Command	**AT+CIPSTA_CUR?**
Response	+CIPSTA_CUR:<IP> OK
Parameters	Param description <IP> string, IP address of ESP8266 station
Command	**AT+CIPSTA_CUR=<IP>[,<gateway>,<netmask>]**
Response	OK
Parameters	<IP> string, IP address of ESP8266 station [<gateway>] gateway [<netmask>] netmask
Notes	This configuration will not store in Flash. This configuration interacts with DHCP related AT commands(AT+CWDHCP related): If enable static IP, DHCP will be disabled; If enable DHCP, static IP will be disabled; This will depend on the last configuration.

25. AT+CIPSTA_DEF – Set IP address of station and save as default

AT+CIPSTA_DEF – Set IP address of ESP8266 station, save to Flash	
Example	AT+CIPSTA_DEF="192.168.6.100","192.168.6.1","255.255.255.0"
Command	**AT+CIPSTA_DEF?**
Response	+CIPSTA:<IP> OK
Parameters	<IP> string, IP address of ESP8266 station
Command	**AT+CIPSTA_DEF=<IP>[,<gateway>,<netmask>]**
Response	OK
Parameters	<IP> string, IP address of ESP8266 station [<gateway>] gateway [<netmask>] netmask
Notes	This configuration will store in Flash user parameter area. This configuration interacts with DHCP related AT commands(AT+CWDHCP related): If enable static IP, DHCP will be disabled; If enable DHCP, static IP will be disabled; This will depend on the last configuration.

26. AT+ CIPAP – Set IP address of softAP

AT+ CIPAP – Set IP address of ESP8266 softAP [@deprecated]. Please use AT+CIPAP_CUR or AT+CIPAP_DEF instead.	
Example	AT+CIPAP="192.168.5.1"
Command	**AT+CIPAP?**
Response	+CIPAP:<IP> OK
Parameters	<IP> string, IP address of ESP8266 softAP
Command	**AT+CIPAP=<IP>**
Response	OK
Parameters	<IP> string, IP address of ESP8266 softAP
Notes	This configuration will store in Flash user parameter area. This configuration interacts with DHCP related AT commands(AT +CWDHCP related): If enable static IP, DHCP will be disabled; If enable DHCP, static IP will be disabled; This will depend on the last configuration.

27. AT+CIPAP_CUR – Set IP address of softAP

AT+CIPAP_CUR – Set IP address of ESP8266 softAP, won't save to Flash	
Example	AT+CIPAP_CUR="192.168.5.1"
Command	**AT+CIPAP_CUR?**
Response	+CIPAP_CUR:<IP> OK
Parameters	<IP>　string, IP address of ESP8266 softAP
Command	**AT+CIPAP_CUR=<IP>**
Response	OK
Parameters	<IP>　string, IP address of ESP8266 softAP
Notes	This configuration will not store in Flash. This configuration interacts with DHCP related AT commands(AT+CWDHCP related): If enable static IP, DHCP will be disabled; If enable DHCP, static IP will be disabled; This will depend on the last configuration.

28. AT+CIPAP_DEF – Set IP address of softAP, save as default

AT+ CIPAP_DEF – Set IP address of ESP8266 softAP, save to Flash	
Example	AT+CIPAP_DEF="192.168.5.1"
Command	**AT+CIPAP_DEF?**
Response	+CIPAP_DEF:<IP> OK
Parameters	<IP>　string, IP address of ESP8266 softAP
Command	**AT+CIPAP_DEF=<IP>**
Response	OK
Parameters	<IP>　string, IP address of ESP8266 softAP
Notes	This configuration will store in Flash user parameter area. This configuration interacts with DHCP related AT commands(AT+CWDHCP related): If enable static IP, DHCP will be disabled; If enable DHCP, static IP will be disabled; This will depend on the last configuration.

29. AT+CWSTARTSMART – Start SmartConfig

AT+CWSTARTSMART – Start SmartConfig	
Example	`AT+CWMODE=1` `AT+CWSTARTSMART=1`
Command	`AT+CWSTARTSMART =<type>`
Response	`OK`

Parameters	`< type>  SmartConfig protocol type`
	`1: ESP_TOUCH`
	`2: AirKiss`

Notes	• You can apply for more documents about our SmartConfig from Espressif. • ESP8266 station has to be enable • Message "Smart get wifi info" means Smart Config succeed, then you can use "AT+CIFSR" to check whether it got IP from router or not. • ESP8266 can't do anything during SmartConfig so please wait till it succeed or use command "AT+CWSTOPSMART" to stop SmartConfig.

30. AT+CWSTOPSMART – stop SmartConfig

AT+CWSTOPSMART stop SmartConfig	
Command	`AT+CWSTOPSMART`
Response	`OK`
Notes	No matter SmartConfig succeed or not, please always call "AT +CWSTOPSMART" to release the buffer it took.

5.   TCP/IP Related AT Commands

5.1.   Overview

TCP/IP	
Command	**Description**
AT+ CIPSTATUS	Get connection status
AT+CIPSTART	Establish TCP connection or register UDP port
AT+CIPSEND	Send data
AT+CIPCLOSE	Close TCP/UDP connection
AT+CIFSR	Get local IP address
AT+CIPMUX	Set multiple connections mode
AT+CIPSERVER	Configure as server
AT+CIPMODE	Set transmission mode
AT+SAVETRANSLINK	Save transparent transmission link to Flash
AT+CIPSTO	Set timeout when ESP8266 runs as TCP server
AT+CIUPDATE	Upgrade firmware through network
AT+PING	Function PING

5.2.   TCP/IP

1.   AT+CIPSTATUS – Information about connection

AT+CIPSTATUS – Information about connection	
Command	**AT+CIPSTATUS**
Response	STATUS:<stat> +CIPSTATUS:<ID>,<type>,<remote_IP>,<remote_port> ,<local_port>,<te type>
Parameters	<stat> <stat> 2: Got IP <stat> 3: Connected <stat> 4: Disconnected <id> id of the connection (0~4), for multi-connect <type> string, "TCP" or "UDP" <remote_IP> string, remote IP address. <remote_port> remote port number <local_port> ESP8266 local port number <tetype> <tetype> 0: ESP8266 runs as client <tetype> 1: ESP8266 runs as server

2. AT+CIPSTART – Start connection

AT+CIPSTART – Establish TCP connection or register UDP port, start connection	
Example	AT+CIPSTART="TCP","192.168.101.110",1000 Refer to "Espressif AT Command Examples"
Command	**AT+CIPSTART=?**
Response	1) If AT+CIPMUX=0 +CIPSTART:(<type>),(<IP address>),(<port>)[,(<local port>),(<mode>)] +CIPSTART:(<type>),(<domain name>),(<port>)[,(<local port>),(<mode>)] OK 2) If AT+CIPMUX=1 +CIPSTART:(id),(<type>),(<IP address>),(<port>)[,(<local port>),(<mode>)] +CIPSTART: (id), (<type>),(<domain name>),(<port>)[,(<local port>),(<mode>)]
Parameters	null
Single connection	(+CIPMUX=0) **AT+CIPSTART=<type>,<addr>,<port>[,<UDP local port>,<UDP mode>] [,<TCP keep alive>]**
Multiple connection	(+CIPMUX=1) **AT+CIPSTART=<id>,<type>,<addr>,<port>[,<UDP local port>,<UDP mode>][,<TCP keep alive>]**
Response	OK or ERROR If connection already exists, returns ALREAY CONNECT
Parameters	<id> 0–4 , id of connection <type> string, "TCP" or "UDP" <addr> string, remote IP <port> string, remote port [<UDP local port>] for UDP only [<UDP mode>] In UDP transparent transmission, it has to be 0. [<UDP mode>] 0 : destination peer entity of UDP will not change. [<UDP mode>] 1 : destination peer entity of UDP can change once. [<UDP mode>] 2 : destination peer entity of UDP is allowed to change. Note: [<UDP mode>] can only be used when [<UDP local port>] is set. [<TCP keep alive>] default 0. unit: 500 milliseconds. [<TCP keep alive>] 0 : disable TCP keep-alive [<TCP keep alive>] 1 ~ 7200 : TCP keep-alive interval

3. AT+CIPSEND – Send data

AT+CIPSEND – Send data	
Command	**AT+CIPSEND=?**
Response	OK
Parameters	null
Single connection	(+CIPMUX=0) **AT+CIPSEND=<length>**
Multiple connection	(+CIPMUX=1) **AT+CIPSEND=<id>,<length>**
UDP Transmission	**AT+CIPSEND=[<id>,]<length>[,<remote IP>,<remote port>]**
Response	Wrap return ">" after set command. Begins receive of serial data, when data length is met, starts transmission of data. If connection cannot be established or gets disconnected during send, returns ERROR If data is transmitted successfully, returns SEND OK
Parameters	<id> ID no. of transmit connection <length> data length, MAX 2048 bytes
Command	**AT+CIPSEND**
Response	Wrap return ">" after execute command. Enters unvarnished transmission, 20ms interval between each packet, maximum 2048 bytes per packet. When single packet containing "+++" is received, it returns to normal command mode. This command can only be used in transparent transmission mode which require to be single connection mode. For UDP transparent transmission, <UDP mode> has to be 0 in command "AT+CIPSTART"
Notes	Please refer to "Espressif AT Command Examples" for more examples.

4. AT+CIPSENDEX – Send data

AT+CIPSENDEX – Send data	
Command	`AT+CIPSENDEX=?`
Response	`OK`
Parameters	`null`
Single connection	`(+CIPMUX=0)` `AT+CIPSENDEX=<length>`
Multiple connection	`(+CIPMUX=1)` `AT+CIPSENDEX=<id>,<length>`
UDP Transmission	`AT+CIPSENDEX=[<id>,]<length>[,<remote IP>,<remote port>]`
Response	Wrap return ">" after set command. Begins receive of serial data, when data length or "\0" is met, starts transmission of data. So if sending "\0" is needed, please send it as "\\0" If connection cannot be established or gets disconnected during send, returns ERROR If data is transmitted successfully, returns SEND OK
Parameters	`<id>` ID no. of transmit connection `<length>` data length, MAX 2048 bytes

5. AT+CIPCLOSE – Close TCP or UDP connection

AT+CIPCLOSE – Close TCP or UDP connection	
Command	`AT+CIPCLOSE=?`
Response	`OK`
Multiple connection	`AT+CIPCLOSE=<id>`

Response	`No errors, returns`
	`OK`
	
	`If connection <id> is disconnected, returns`
	`Link is not`

Parameters	`<id>` ID no. of connection to close, when id=5, all connections will be closed. (id=5 has no effect in server mode)
Single connection	`AT+CIPCLOSE`

Response	`OK`
	`or`
	`If no such connection, returns`
	`ERROR`
	
	`Prints UNLINK when there is no connection`

6. AT+CIFSR – Get local IP address

AT+CIFSR – Get local IP address	
Command	`AT+CIFSR=?`
Response	`OK`
Command	`AT+ CIFSR`

Response	`+ CIFSR:<IP address>`
	`+ CIFSR:<IP address>`
	
	`OK`
	`ERROR`

Parameters	`<IP address>`
	`IP address of ESP8266 softAP`
	`IP address of ESP8266 station`

7. AT+CIPMUX – Enable multiple connections

AT+ CIPMUX – Enable multiple connections or not	
Example	`AT+CIPMUX=1`
Command	**AT+CIPMUX?**

Response	```+ CIPMUX:<mode>```
	
	```OK```

Parameters	```<mode>0 single connection``` ```<mode>1 multiple connection```
Command	**AT+CIPMUX=<mode>**

Response	```OK```
	
	```If already connected, returns```
	```Link is builded```

Parameters	```The same as above.```

Notes	```1. "AT+CIPMUX=1" can only be set when transparent transmission```
	```disabled ("AT+CIPMODE=0")```
	```2. This mode can only be changed after all connections are```
	```disconnected. If server is started, reboot is required.```

8. AT+CIPSERVER – Configure as TCP server

AT+ CIPSERVER – Configure as TCP server	
Example	AT+ CIPMUX=1 AT+ CIPSERVER=1,1001
Command	**AT+ CIPSERVER=<mode>[,<port>]**
Response	OK
Parameters	<mode>0 Delete server (need to follow by restart) <mode>1 Create server <port> port number, default is 333
Notes	1. Server can only be created when AT+CIPMUX=1 2. Server monitor will automatically be created when Server is created. 3. When a client is connected to the server, it will take up one connection，be gave an id.

9.   AT+CIPMODE – Set transfer mode

AT+ CIPMODE – Set transfer mode	
Example	`AT+CIPMODE=1`
Command	**AT+ CIPMODE?**

Response	`+ CIPMODE:<mode>`
	
	`OK`

Parameters	`<mode>0  normal mode`
	`<mode>1  unvarnished transmission mode,  only for TCP single`
	`connection.`

Command	**AT+CIPMODE=<mode>**

Response	`OK`
	
	`If already connected, returns`
	`Link is builded`

Parameters	`<mode>0  normal mode`
	`<mode>1  unvarnished transmission mode,  only for TCP single`
	`connection.`

Notes	`This configuration would not save into Flash.`

10. AT+SAVETRANSLINK – Save transparent transmission link to Flash

AT+SAVETRANSLINK – Save transparent transmission link to Flash	
Example	AT+SAVETRANSLINK=1,"192.168.6.110",1002,"TCP"
Command	**AT+SAVETRANSLINK =<mode>,<IP>,<port>[,<type>]**
Response	OK or ERROR
Parameters	<mode>0 normal mode <mode>1 transparent transmission mode <IP> remote IP <port> remote port [<type>] TCP or UDP, default to be "TCP"
Notes	1. This command will save the transparent transmission mode and its TCP link into Flash user parameter area. 2. As long as the IP, port numerical conformance to specification, we will save them to Flash

11. AT+CIPSTO – Set TCP server timeout

AT+ CIPSTO – Set TCP server timeout	
Example	AT+ CIPMUX=1 AT+ CIPSERVER=1,1001 AT+CIPSTO=10
Command	**AT+CIPSTO?**
Response	+ CIPSTO:<time> OK
Parameters	Param description The same as below.
Command	**AT+CIPSTO=<time>**
Response	OK
Parameters	Param description < time> TCP server timeout, range 0~7200 seconds
Notes	ESP8266 as TCP server, will disconnect to TCP client that didn't communicate with it even if timeout. If AT+CIPSTO=0, it will never timeout. We don't recommend that.

12. AT+CIUPDATE – Update through network

AT+ CIUPDATE – update through network	
Command	`AT+ CIUPDATE`

Response	`+CIPUPDATE:<n>`
	
	`OK`

Parameters	`<n> 1  found server`
	`<n>  2  connect server`
	`<n>  3  got edition`
	`<n>  4  start update`

Notes	`Firmware upgrade depends on network condition.` `It will return ERROR if upgrade fail, please wait a while.` `1.` `If using Espressif AT BIN (`\esp_iot_sdk\bin\at`), "`AT+CIUPDATE`"` `   will download new AT BIN from Espressif Cloud。` `2.` `If using user-compiled AT BIN, users need to make their own "`AT` ` +CIUPDATE`" to upgrade, Espressif provide a demo in `\esp_iot_sdk` ` `\example\at`` `3.` `AT BINs on server have to be named as "user1.bin" and` `   "user2.bin"`

13. AT+PING – Function Ping

AT+PING – Function Ping	
Example	`AT+PING="192.168.1.1"` `AT+PING="www.baidu.com"`
Command	**AT+PING=<IP>**

Response	`+<time>`
	
	`OK`
	`Or`
	`ERROR // means ping fail`

Parameters	`Param description`
	`<IP> : string, host IP or domain name`
	`<time> : response time of ping`

14. +IPD – Receive network data

+IPD – Receive network data	
Single connection	**(+CIPMUX=0)** **+IPD,<len>:<data>**
Multiple connection	**(+CIPMUX=1)** **+IPD,<id>,<len>:<data>**

Parameters	`<id>   id no. of connection`
	`<len>  data length`
	`<data>  data received`

Notes	`When the module receives network data, it will send the data` `through the serial port using +IPD command.`

6.   Appendix

ESP8266 AT commands below will save configuration into Flash:

| Command	Example
Save in Flash user parameter area	
AT+UART_DEF	AT+UART_DEF=115200,8,1,0,3
AT+CWDHCP_DEF	AT+CWDHCP_DEF=1,1
AT+CIPSTAMAC_DEF	AT+CIPSTAMAC_DEF="18:fe:35:98:d3:7b"
AT+CIPAPMAC_DEF	AT+CIPAPMAC_DEF="1a:fe:36:97:d5:7b"
AT+CIPSTA_DEF	AT+CIPSTA_DEF="192.168.6.100"
AT+CIPAP_DEF	AT+CIPAP_DEF="192.168.5.1"
AT+SAVETRANSLINK	AT+SAVETRANSLINK =1,"192.168.6.10",1001
Save in Flash system parameter area	
AT+CWMODE_DEF	AT+CWMODE_DEF=3
AT+CWJAP_DEF	AT+CWJAP_DEF="abc", "0123456789"
AT+CWSAP_DEF	AT+CWSAP_DEF="ESP8266","12345678",5,3
AT+CWAUTOCONN	AT+CWAUTOCONN=1

Notes:

- We will check the new setting with original configuration from flash first，only if the configuration changes，we will write it to flash.

- To 512KB flash, default setting:

    user parameter area is 0x3C000 ~ 0x40000，16KB;

    system parameter area is 0x7C000~0x80000, 16KB

- To 1MB flash (or larger than 1MB), default setting:

    user parameter area is 0x7C000 ~ 0x80000, 16KB;

    system parameter area is the last 16KB of flash.

7. Q&A

If you have any questions about AT Commands, please contact us (support-at@espressif.com) with information as follows:

- **Version info of AT** : Using "AT+GMR" to get the version info.

  Hardware Module info: example Ai-thnker ESP-01

- **Screenshot or steps of the test steps,** for example:
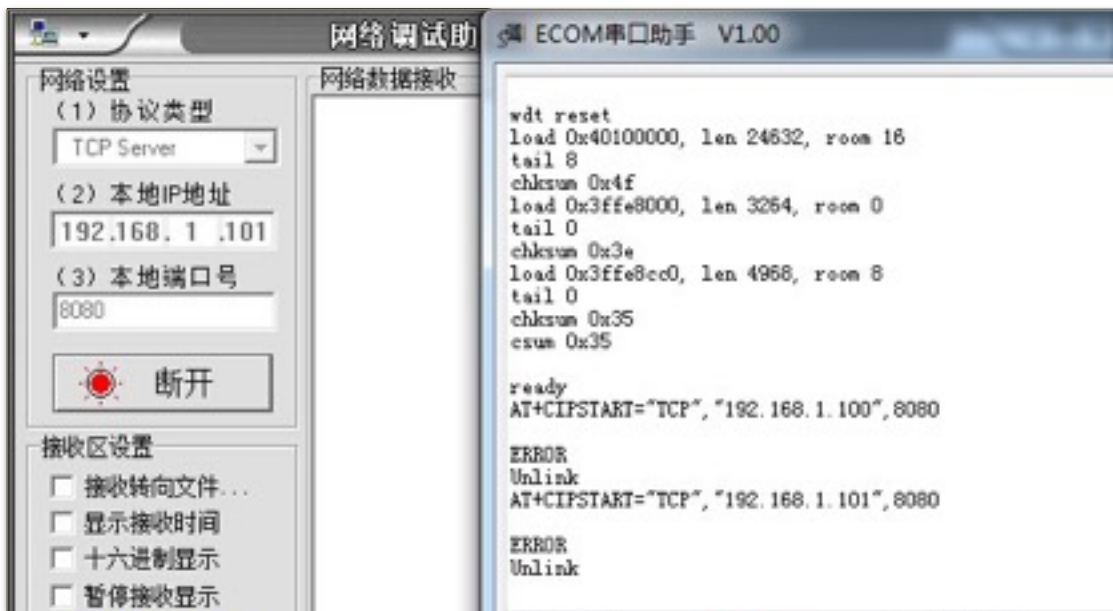
- **Log**:

```
ets Jan  8 2013,rst cause: 1, boot mode: (3,3)

load 0x40100000, len 26336, room 16
tail 0
chksum 0xde
load 0x3ffe8000, len 5672, room 8
tail 0
chksum 0x69
load 0x3ffe9630, len 8348, room 8
tail 4
chksum 0xcb
csum 0xcb
SDK version: 0.9.1
addr not ack when tx write cmd
mode : sta(18: fe: 34: 97: d5: 7b) + softAP(1a: fe: 34: 97: d5: 7b)
```