Espressif: AT Instruction Set

Status	Released
Current version	V0.20
Author	Xu Jingjie
Completion Date	2014.11.28
Reviewer	
Completion Date	

[√] CONFIDENTIAL
[ ] INTERNAL
[ ] PUBLIC

Version Info

Date	Version	Author	Comments/Changes
2014.6.27	0.1	XuJingjie	Draft
2014.7.11	0.11	XuJingjie	Unvarnished transmission added
2014.8.12	0.15	XuJingjie	1、Added Timeout and IP settings for AP 2、Edited description for server functions 3、Support DNS
2014.9.25	0.18	XuJingjie	1、Added upgrade through network 2、Added CWLAP
2014.11.10	0.19	XuJingjie	Added UDP
2014.11.27	0.20	XuJingjie	1、Added set and get APIP/APMAC/STAIP /STAMAC 2、Added start and stop DHCP

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, NONINFRINGEMENT, 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 Wi-Fi Alliance Member Logo is a trademark of the Wi-Fi 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 Overview

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

Instruction set is divided into: Basic AT commands, Wifi function, AT commands, TCP / IP Toolbox AT commands.

Note: 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.

2 Instruction Description

Each instruction set contains four types of AT commands.

Type	Instruction 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.

Note:

1. Not all AT instruction has four commands.
2. [] = default value, not required or may not appear
3. String values require double quotation marks, for example: AT+CWSAP=“ESP756190”,”21030826”,1,4
4. Baud rate = 115200
5. AT instruction ends with “\r\n”

3 AT Instruction Listing

Instructions	Description
Basic	
AT	Test if AT startup
AT+RST	Restart
AT+GMR	View version info
AT+GSLP	Enter deep-sleep mode
ATE	AT commands echo
Wi-Fi	
AT+CWMODE	WIFI mode（station/softAP/station+softAP）
AT+CWJAP	Connect to AP
AT+CWLAP	Lists available APs
AT+CWQAP	Disconnect from AP
AT+CWSAP	Set parameters under AP mode
AT+CWLIF	Get stations' ip which are connected to ESP8266 softAP
AT+CWDHCP	Enable/Disable DHCP
AT+CIPSTAMAC	Set mac address of ESP8266 station
AT+CIPAPMAC	Set mac address of ESP8266 softAP
AT+CIPSTA	Set ip address of ESP8266 station
AT+CIPAP	Set ip address of ESP8266 softAP
TCP/IP	
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+CIPSTO	Set timeout when ESP8266 runs as TCP server
AT+CIUPDATE	For OTA（upgrade through network）
Data RX	
+IPD	Data received from network

4 Basic AT Instruction Set

4.1 Overview

Basic	
Instruction	**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

4.2 Instructions

4.2.1 AT – Test AT startup

AT – Test AT startup	
Type：execute Instruction： **AT**	Response： OK
	Param description：null

4.2.2 AT+RST – Restart module

AT+RST – Restart module	
Type ：execute Instruction： **AT+RST**	Response： OK
	Param description：null

4.2.3 AT+GMR – View version info

AT+GMR – View version info	
Type ： execute Instruction： **AT+GMR**	Response： <number> OK
	Param description： < number > version info，length：8 bytes

Note	For example，response is 0017xxxxxx，then 0017 means the AT version.

4.2.4 AT+GSLP – Enter deep-sleep mode

AT+GSLP – Enter deep-sleep mode	
Type ： set Instruction： **AT+GSLP=<time>**	Response： <time> OK
	Param description： < time > ms , set the sleep time of ESP8266 in ms. ESP8266 will wake up after X ms in deep-sleep.
Note	Hardware has to support deep-sleep wake up (XPD_DCDC connects to EXT_RSTB with 0R).

4.2.5 ATE – AT commands echo

ATE – AT commands echo	
Type ： set Instruction： **ATE**	Response： OK
	Param description： ATE0 ： Disable echo ATE1 ： Enable echo

5 WIFI functions

5.1 Overview

WIFI	
Instruction	**Description**
AT+CWMODE	WIFI mode（station/softAP/station+softAP）
AT+CWJAP	Connect to AP
AT+CWLAP	Lists available APs
AT+CWQAP	Disconnect from AP
AT+CWSAP	Set parameters under AP mode
AT+CWLIF	Get station’s ip which is connected to ESP8266 softAP
AT+CWDHCP	Enable/Disable DHCP
AT+CIPSTAMAC	Set mac address of ESP8266 station

AT+CIPAPMAC	Set mac address of ESP8266 softAP
AT+CIPSTA	Set ip address of ESP8266 station
AT+CIPAP	Set ip address of ESP8266 softAP

5.2 Instructions

5.2.1AT+CWMODE – WIFI mode

AT+CWMODE - WIFI mode（station/softAP/station+softAP）	
Type：test Function: Get value scope of wifi mode. Instruction： **AT+CWMODE=?**	Response： +CWMODE:(value scope of <mode>) OK
	Param description： <mode>1 means Station mode 2 means AP mode 3 means AP + Station mode
Type：query Function: Query ESP8266's current wifi mode. Instruction： **AT+CWMODE?**	Response： +CWMODE:<mode> OK
	Param description： The same as above.
Type：set Function: Set ESP8266 wifi mode Instruction： **AT+CWMODE=<mode>**	Response： OK
	Param description： The same as above.

5.2.2AT+CWJAP – Connect to AP

AT+CWJAP – Connect to AP	
Type：query Function: Query AP's info which is connect by ESP8266. Instruction： **AT+ CWJAP?**	Response： + CWJAP:<ssid> OK
	Param description： <ssid> string, AP's SSID

Type：set Function: Set AP's info which will be connect by ESP8266. Instruction： **AT+ CWJAP =<ssid>,< pwd >**	Response： OK ERROR
	Param description： <ssid> string, AP's SSID <pwd> string, MAX: 64 bytes

5.2.3AT+CWLAP – List available APs

AT+CWLAP - Lists available APs	
Type：set Function: Search available APs with specific conditions. Instruction： **AT+ CWLAP =** **<ssid>,< mac >,<ch>**	Response： + CWLAP: <ecn>,<ssid>,<rssi>,<mac> OK ERROR
	Param description： The same as below.
Type ： execute Function: Lists all available APs. Instruction： **AT+CWLAP**	Response： + CWLAP: <ecn>,<ssid>,<rssi>,<mac> OK ERROR
	Param description： < ecn >0 OPEN 1 WEP 2 WPA_PSK 3 WPA2_PSK 4 WPA_WPA2_PSK <ssid> string, SSID of AP <rssi> signal strength <mac> string, MAC address

5.2.4AT+CWQAP – Disconnect from AP

AT+CWQAP - Disconnect from AP	
Type：test Function: Only for test Instruction：	Response： OK
	Param description：

AT+CWQAP=?	
Type ： execute Function: Disconnect from AP. Instruction： **AT+ CWQAP**	Response： OK Param description：

5.2.5AT+CWSAP – Configuration of softAP mode

AT+ CWSAP – Configuration of softAP mode	
Type：Query Function: Query configuration of softAP mode. Instruction： **AT+ CWSAP?**	Response： + CWSAP:<ssid>,<pwd>,<chl>,<ecn> Param description： The same as below.
Type：Set Function: Set configuration of softAP mode. Instruction： **AT+ CWSAP=** **<ssid>,<pwd>,<chl>,** **<ecn>**	Response： OK ERROR Note: This CMD is only available when softAP mode enable, and need to follow by AT+RST to make it works. Param description： <ssid> string, ESP8266 softAP' SSID <pwd> string, MAX: 64 bytes <chl> channel id < ecn >0 OPEN 2 WPA_PSK 3 WPA2_PSK 4 WPA_WPA2_PSK

5.2.6AT+CWLIF – IP of stations

AT+ CWLIF – ip of stations which are connected to ESP8266 softAP	
Type ： execute Function: Get ip of stations which are connected to ESP8266 softAP	Response： <ip addr> OK Param description：

Instruction: **AT+CWLIF**	<ip addr> ip address of stations which are connected to ESP8266 softAP

5.2.7AT+CWDHCP – Enable/Disable DHCP

AT+ CWDHCP – Enable/Disable DHCP	
Type : set Function: Enable/Disable DHCP. Instruction: **AT+CWDHCP=<mode>,<en>**	Response: OK
	Param description: <mode> 0 : set ESP8266 softAP 1 : set ESP8266 station 2 : set both softAP and station <en> 0 : Enable DHCP 1 : Disable DHCP

5.2.8AT+CIPSTAMAC – Set mac address of station

AT+ CIPSTAMAC – Set mac address of ESP8266 station	
Type : query Function: Get mac address of ESP8266 station. Instruction: **AT+CIPSTAMAC?**	Response: +CIPSTAMAC:<mac> OK
	Param description: <mac> string, mac address of ESP8266 station
Type : set Function: Set mac address of ESP8266 station. Instruction: **AT+CIPSTAMAC=<mac>**	Response: OK
	Param description: <mac> string, mac address of ESP8266 station

5.2.9AT+CIPAPMAC – Set mac address of softAP

AT+ CIPAPMAC – Set mac address of ESP8266 softAP	
Type ： query Function: Get mac address of ESP8266 softAP. Instruction： **AT+CIPAPMAC?**	Response： +CIPAPMAC:<mac> OK Param description： <mac> string, mac address of ESP8266 softAP
Type ： set Function: Set mac address of ESP8266 softAP. Instruction： **AT+CIPAPMAC=<mac>**	Response： OK Param description： <mac> string, mac address of ESP8266 softAP

5.2.10 AT+ CIPSTA – Set ip address of station

AT+ CIPSTA – Set ip address of ESP8266 station	
Type ： query Function: Get ip address of ESP8266 station. Instruction： **AT+CIPSTA?**	Response： +CIPSTA:<ip> OK Param description： <ip> string, ip address of ESP8266 station
Type ： set Function: Set ip address of ESP8266 station. Instruction： **AT+CIPSTA=<ip>**	Response： OK Param description： <ip> string, ip address of ESP8266 station

5.2.11 AT+ CIPAP – Set ip address of softAP

AT+ CIPAP – Set ip address of ESP8266 softAP	
Type ： query Function:	Response： +CIPAP:<ip>

Get ip address of ESP8266 softAP. Instruction： **AT+CIPAP?**	OK
	Param description： <ip> string, ip address of ESP8266 softAP
Type ： set Function: Set ip address of ESP8266 softAP. Instruction： **AT+CIPAP=<ip>**	Response： OK
	Param description： <ip> string, ip address of ESP8266 softAP

6 TCP/IP Related

6.1 Overview

TCP/IP	
Instruction	**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+CIPSTO	Set timeout when ESP8266 runs as TCP server

6.2 TCP/IP

6.2.1AT+ CIPSTATUS – Information about connection

AT+ CIPSTATUS – Information about connection	
Type ： execute Function: Get information about connection. Instruction: **AT+ CIPSTATUS**	Response： STATUS:<stat> + CIPSTATUS:<id>,<type>,<addr>,<port>,<tetype> OK
	Param description： <stat> 2: Got IP 3: Connected 4: Disconnected <id> id of the connection (0~4), for multi-connect <type> string, “TCP” or “UDP” <addr> string, IP address. <port> port number <tetype> 0: ESP8266 runs as client 1: ESP8266 runs as server

6.2.2AT+CIPSTART – Start connection

AT+CIPSTART – Establish TCP connection or register UDP port, start connection	
Type ： test Function: Get the information of param. Instruction: **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>)]
	Param description： null
Type ： Set Function: Start a connection as client. Instruction: 1)Single connection **(+CIPMUX=0)** **AT+CIPSTART=** **<type>,<addr>,<port>** **[,(<local port>),(<mode>)]** 2)Multiple connection (+CIPMUX=1) **AT+CIPSTART=** **<id><type>,<addr>,<port>** **[,(<local port>),(<mode>)]**	Response： OK or ERROR If connection already exists, returns ALREAY CONNECT
	Param description： <id> 0-4 , id of connection <type> string, “TCP” or “UDP” <addr> string, remote ip <port> string, remote port [<local port>] for UDP only [<mode>] for UDP only 0 : destination peer entity of UDP will not change. 1 : destination peer entity of UDP can change once. 2 : destination peer entity of UDP is allowed to change. Note: [<mode>] can only be used when [<local port>] is set.

6.2.3AT+CIPSEND – Send data

AT+CIPSEND – Send data	
Type ： test Function: Only for test. Instruction: **AT+CIPSEND=?**	Response： OK Param description： null
Type ： Set Function: Set length of the data that will be sent. For normal send. Instruction: 1)For single connection: (+CIPMUX=0) **AT+CIPSEND=<length>** 2) For multiple connection: (+CIPMUX=1) **AT+CIPSEND=** **<id>,<length>**	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 Note: This CMD Param description： <id> ID no. of transmit connection <length> data length, MAX 2048 bytes
Type ： execute Function: Send data. For unvarnished transmission mode. Instruction: **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 command mode. This command can only be used in unvarnished transmission mode which require to be single connection mode.

6.2.4AT+CIPCLOSE – Close TCP or UDP connection

AT+CIPCLOSE – Close TCP or UDP connection	
Type ： test Function:	Response：

Only for test. Instruction: **AT+CIPCLOSE=?**	OK
Type ： Set Function: Close TCP or UDP connection. Instruction: For multiply connection mode **AT+CIPCLOSE=<id>**	Response： No errors, returns OK If connection <id> is disconnected, returns Link is not Param description： <id> ID no. of connection to close, when id=5, all connections will be closed. (id=5 has no effect in server mode)
Type ： execute Instruction: For single connection mode **AT+CIPCLOSE**	Response： OK or If no such connection, returns ERROR Prints UNLINK when there is no connection

6.2.5AT+CIFSR – Get local IP address

AT+CIFSR – Get local IP address	
Type ： Test Function: Only for test. Instruction: **AT+CIFSR=?**	Response： OK
Type ： Execute Function: Get local IP address. Instruction: **AT+ CIFSR**	Response： + CIFSR:<IP address> + CIFSR:<IP address> OK ERROR Param description： <IP address> IP address of ESP8266 softAP IP address of ESP8266 station

6.2.6AT+ CIPMUX – Enable multiple connections

AT+ CIPMUX – Enable multiple connections or not	
Type : Query Function: Get param config. Instruction: **AT+ CIPMUX?**	Response: + CIPMUX:<mode> OK
	Param description: The same as below.
Type : Set Function: Set connection mode. Instruction: **AT+ CIPMUX=<mode>**	Response: OK If already connected, returns Link is builded
	Param description: <mode>0 single connection 1 multiple connection
Note	This mode can only be changed after all connections are disconnected. If server is started, reboot is required.

6.2.7AT+ CIPSERVER – Configure as TCP server

AT+ CIPSERVER – Configure as TCP server	
Type : Set Function: Set TCP server. Instruction: **AT+ CIPSERVER= <mode>[,<port>]**	Response: OK
	Param description: <mode> 0 Delete server (need to follow by restart) 1 Create server <port> port number, default is 333
Note	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.

6.2.8AT+ CIPMODE – Set transfer mode

AT+ CIPMODE – Set transfer mode

Type ： Query Function: Query transfer mode. Instruction: **AT+ CIPMODE?**	Response： + CIPMODE:<mode> OK
	Param description： The same as below.
Type ： Set Function: Set transfer mode. Instruction: **AT+CIPMODE=<mode>**	Response： OK If already connected, returns Link is builded
	Param description： <mode>0 normal mode 1 unvarnished transmission mode

6.2.9AT+ CIPSTO – Set server timeout

AT+ CIPSTO – Set server timeout	
Type ： Query Function: Query server timeout. Instruction: **AT+CIPSTO?**	Response： + CIPSTO:<time> OK
	Param description： The same as below.
Type ： Set Function: Set server timeout. Instruction: **AT+CIPSTO=<time>**	Response： OK
	Param description： < time> server timeout, range 0~7200 seconds

6.2.10 AT+ CIUPDATE – Update through network

AT+ CIUPDATE – update through network	
Type ： execute Function: Start upgrade. Instruction: **AT+ CIUPDATE**	Response： +CIPUPDATE:<n> OK
	Param description： <n> 1 found server 2 connect server 3 got edition

	4 start update

6.2.11 +IPD – Receive network data

+IPD – Receive network data	
1)Single connection: **(+CIPMUX=0)** **+IPD,<len>:<data>** 2) Multiple connection **(+CIPMUX=1)** **+IPD,<id>,<len>:<data>**	NOTE: When the module receives network data, it will send the data through the serial port using +IPD command Param description: <id> id no. of connection <len> data length <data> data received

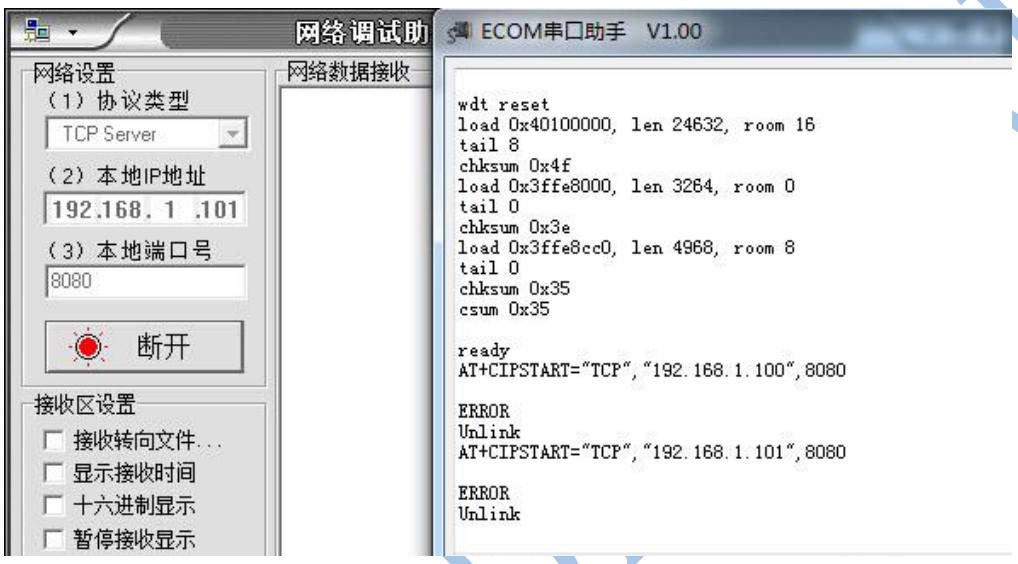
7 Q&A

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

(1) Version info of AT

Using “AT+GMR” to get the version info.

(2) Screenshot of the test steps, for example:



(3) 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)
```