

Espressif ESP8266EX: AT COMMAND EXAMPLES

Status	Released
Current version	V0.4
Author	Fei Yu
Completion Date	2015.03.14
Reviewer	
Completion Date	

CONFIDENTIAL
 INTERNAL
 PUBLIC

Version Info

Date	Version	Author	Comments/Changes
2014.7.16	0.1	Fei Yu	Draft
2014.9.12	0.2	Fei Yu	Add unvarnished transmission (AT_v019)
2014.11.28	0.3	Fei Yu	Add UDP transmission (AT_v020)
2015.03.14	0.4	Fei Yu	Update UDP transmission (AT_v022)

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.

Copyright © 2013 Espressif Systems Inc. All rights reserved.

Table of Content

Version Info	2
Table of Content	3
1、 Foreword	4
2、 User Guide	5
2.1. Single connection as TCP client	6
2.2. Transparent transmission	8
2.3. Multiple connection as TCP server	11
2.4. UDP Transmission	13
2.4.1. UDP (remote ip, port won't change)	14
2.4.2. UDP (remote ip, port can change)	15
3、 Q&A	17

CONFIDENTIAL

1、 Foreword

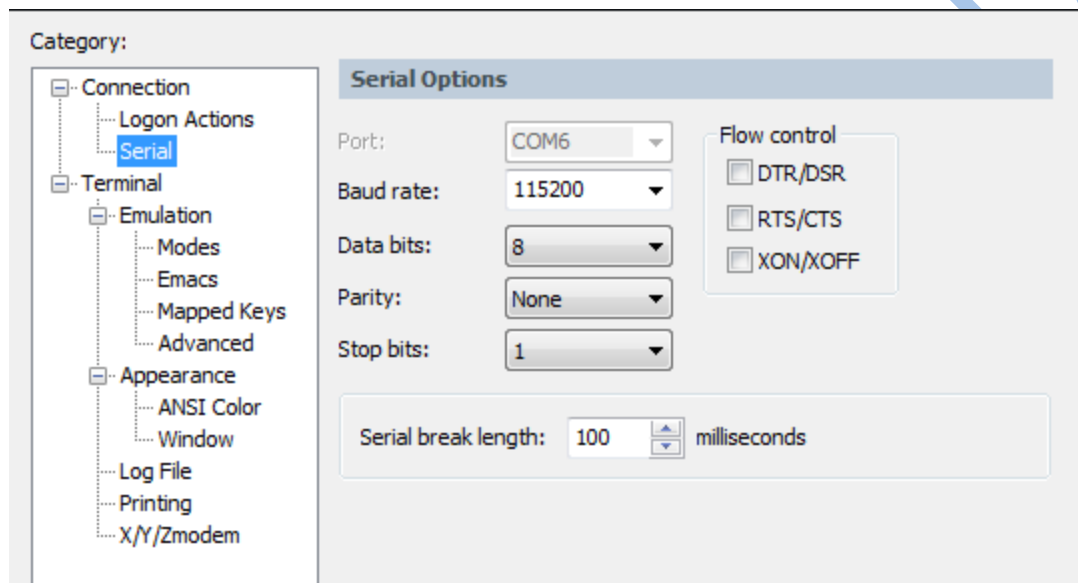
Herein contains examples for the usage of Espressif AT Commands. For the complete instruction set, please refer to Espressif AT Instruction Set documentation.

If you have any question about AT, please contact us support-at@espressif.com

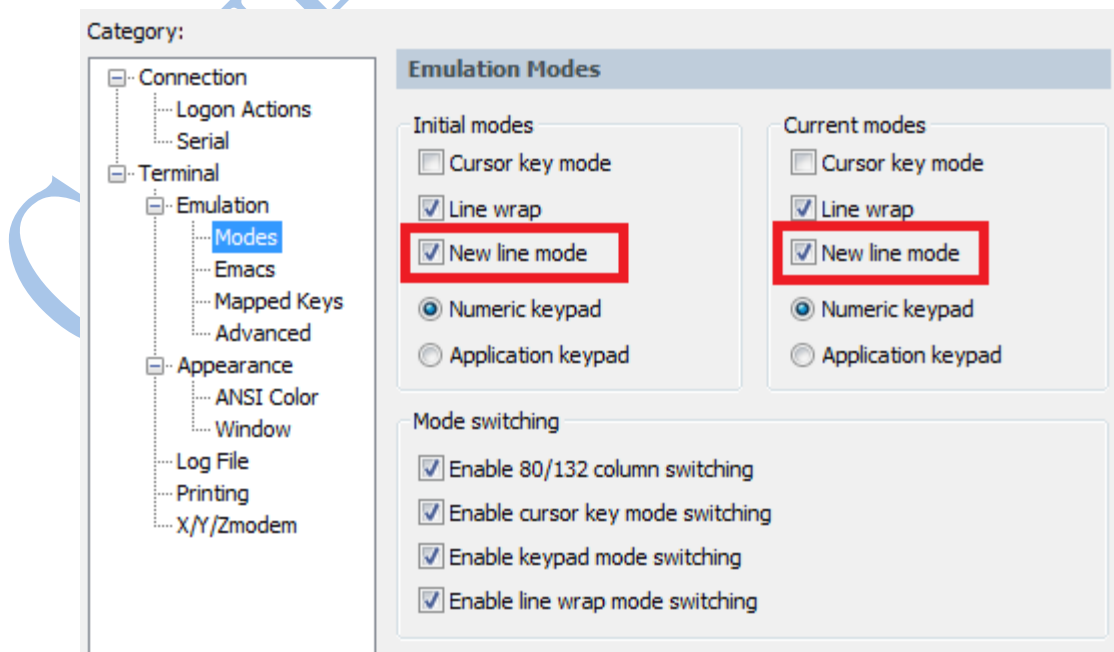
CONFIDENTIAL

2、 User Guide

- (1) First flash in blank.bin (contains default Wi-Fi parameter settings) into the ESP8266EX device, then flash in the BIN program that supports AT commands (\esp_iot_sdk\bin\at).
- (2) Power on device and set serial baud rate to 115200. Enter AT commands.



Pay attention to the new line mode, AT command need “\r\n” to be the end.



2.1. Single connection as TCP client

- 1) Set wifi mode:

```
AT+CWMODE=3 // softAP+station mode
```

Response: OK

- 2) Connect to router

```
AT+CWJAP="ssid","password" // ssid and password of router
```

Response: OK

- 3) Query device's IP

```
AT+CIFSR
```

Response: 192.168.3.106 // Device got an IP from router.

- 4) PC connects to the same router which ESP8266 connects to. Using a network tool (eg: "NetAssist.exe") on the computer to create a server.



- 5) ESP8266EX connect to server as a client

```
AT+CIPSTART="TCP","192.168.3.116",8080 //protocol、 server IP &
port
```

Response: OK

6) Send data

```
AT+CIPSEND=4 // set data length which will be sent, such as 4 bytes
```

```
>DGFY // enter the data, no CR
```

Response: SEND OK

Note: If the number of bytes sent is bigger than the size defined (n), will reply busy, and after sending n number of bytes, reply SEND OK.

7) Receive data:

```
+IPD,n:xxxxxxxx // received n bytes, data=xxxxxxxx
```

2.2. Transparent transmission

In AT Demo, transparent transmission only enables when it is “single connection as client” mode.

Here takes ESP8266EX station as an example, you can take ESP8266EX softAP as the same way according to document “4A-AT-Espressif AT Instruction Set”.

1) Set wifi mode:

```
AT+CWMODE=3 // softAP+station mode
```

Response: OK

2) Connect to router

```
AT+CWJAP="ssid","password" // ssid and password of router
```

Response: OK

3) Query device's IP

```
AT+CIFSR
```

Response: 192.168.101.105 // Device's ip that got from router.

4) PC connects to the same router which ESP8266 connects to. Using a network tool (eg: "NetAssist.exe") on the computer to create a server.



5) Device connect to server

```
AT+CIPSTART="TCP","192.168.101.110",8080 // protocol、server IP & port
```


Response: OK

Linked

6) Enable transparent transmission mode

AT+CIPMODE=1

Response: OK

7) Start send

AT+CIPSEND

Response: >

Note: From now on, data received from uart will be transparent transmitted to server.

```
AT+CIFSR
192.168.101.105
AT+CIPSTART="TCP", "192.168.101.110", 8080

OK
Linked
AT+CIPMODE=1

OK
AT+CIPSEND

>
```



8) Stop send

Data packet contains only “+++” exits transparent transmission.

NOTE: Input +++ directly by keyboard, may not be continually, suggest using tool as below:



Input : +++

New Line Mode : **don't** select

Click “Send”

Note: We send “+++” to exit transparent transmission mode, back to normal AT command mode, TCP connection is still maintain, we can use command “AT+CIPSEND” back to transparent transmission mode.

9) Delete TCP connection

AT+CIPCLOSE

Response: CLOSED OK

2.3. Multiple connection as TCP server

It has to be multiple connection when ESP8266EX runs as server, then there can be more than one client connects to ESP8266EX.

For example,

- 1) Set wifi mode:

```
AT+CWMODE=3 // softAP+station mode
```

Response: OK

- 2) Enable multiple connection

```
AT+CIPMUX=1
```

Response: OK

- 3) Setup server

```
AT+CIPSERVER=1 // default port = 333
```

Response: OK

- 4) PC connects to ESP8266EX softAP as station, then PC connects to ESP8266EX server as client.



NOTE: ESP8266EX acting as server has a timeout mechanism. When connection is established and no data is transmitted within a period of time, it will disconnect the client. Please setup a recurring packet transmission

every 2s on the computer to ensure connection is maintained.



5) Send data

// id number of connection is default to be 0.

```
AT+CIPSEND=0,4 // send 4 bytes to connection NO.0
```

```
>iopd // enter the data, no CR
```

Response: SEND OK

Note: If the number of bytes sent is bigger than the size defined (n), will reply busy, and after sending n number of bytes, reply SEND OK.

6) Receive data:

```
+IPD,0,n:xxxxxxxx // received n bytes, data = xxxxxxxxxx
```

7) Delete TCP connection

```
AT+CIPCLOSE=0 // Delete NO.0 connection.
```

Response: 0,CLOSED OK

2.4. UDP Transmission

AT+CIPSTART can create a UDP transmission, more details in documentation “Espressif AT Instruction Set”.

- 1) Set wifi mode:

```
AT+CWMODE=3 // softAP+station mode
```

Response: OK

- 2) Connect to router

```
AT+CWJAP="ssid","password" // ssid and password of router
```

Response: OK

- 3) Query device's IP

```
AT+CIFSR
```

Response: +CIFSR:STAIP,"192.168.101.104" // IP address of ESP8266 station

- 4) PC connects to the same router which ESP8266 connects to. Using a network tool (eg: "NetAssist.exe") on the computer to create UDP .



Then we example two different kinds of UDP transmission。

2.4.1. UDP (remote ip, port won't change)

In UDP transmission, remote ip and port can change or not is decided by the last parameter of "AT+CIPSTART", 0 means changeless, usually we give it a link id.

- 1) Enable multiple connection

```
AT+CIPMUX=1
```

Response: OK

- 2) Create a UDP transmission, for example, id is 4.

```
AT+CIPSTART=4,"UDP","192.168.101.110",8080,1112,0
```

Response: 4,CONNECT OK

Note:

"192.168.101.110",8080 here is the remote ip and port of UDP transmission which create on PC in step 4;

1112 is the local port of ESP8266, user-define, if user does not define it, it will be a random value;

0 means destination peer entity of UDP will not change. For example, in this case, if another PC also creates a UDP entity and sends data to ESP8266 port 1112, ESP8266 can receive these data, but when we send data with command "AT+CIPSEND=4,X", it will still be sent to the first PC. If this parameter is not 0, it will send to the new PC.

- 3) Send data

```
AT+CIPSEND=4,5 // Send 5 bytes to transmission NO.4
```

```
>DGFYQ // enter the data, no CR
```

Response: SEND OK

Note: If the number of bytes sent is bigger than the size defined (n), will reply busy, and after sending n number of bytes, reply SEND OK.

4) Receive data:

```
+IPD,4,n:xxxxxxxx // received n bytes, data=xxxxxxxx
```

5) Delete transmission NO.4

```
AT+CIPCLOSE=4
```

```
Response: 4,CLOSED OK
```

2.4.2. UDP (remote ip, port can change)

1) Create a UDP transmission, last parameter to be 2.

```
AT+CIPSTART="UDP","192.168.101.110",8080,1112,2
```

```
Response: CONNECT OK
```

Note:

"192.168.101.110",8080 here is the remote ip and port of UDP transmission which create on PC in step 4;

1112 is the local port of ESP8266, user-define, if user does not define it, it will be a random value;

2 means destination peer entity of UDP will change automatically to be the last one which send UDP packet to ESP8266. For example, in this case, if another PC also creates a UDP entity and sends data to ESP8266 port 1112, ESP8266 can receive these data and when it send data back with command "AT+CIPSEND=X", it will still be sent to the new PC.

2) Send data

```
AT+CIPSEND=5 // Send 5 bytes
```

```
>DGFYQ // enter the data, no CR
```

```
Response: SEND OK
```

Note: If the number of bytes sent is bigger than the size defined (n), will reply busy, and after sending n number of bytes, reply SEND OK.

- 3) If you want to send data to any other UDP destination, just set a specific ip and port.

```
AT+CIPSEND=6,"192.168.101.111",1000 // Send 6 bytes
```

```
>abcdef // enter the data, no CR
```

```
Response: SEND OK
```

- 4) Receive data:

```
+IPD,n:xxxxxxxx // received n bytes, data=xxxxxxxx
```

- 5) Delete UDP transmission

```
AT+CIPCLOSE
```

```
Response: CLOSED OK
```

CONFIDENTIAL

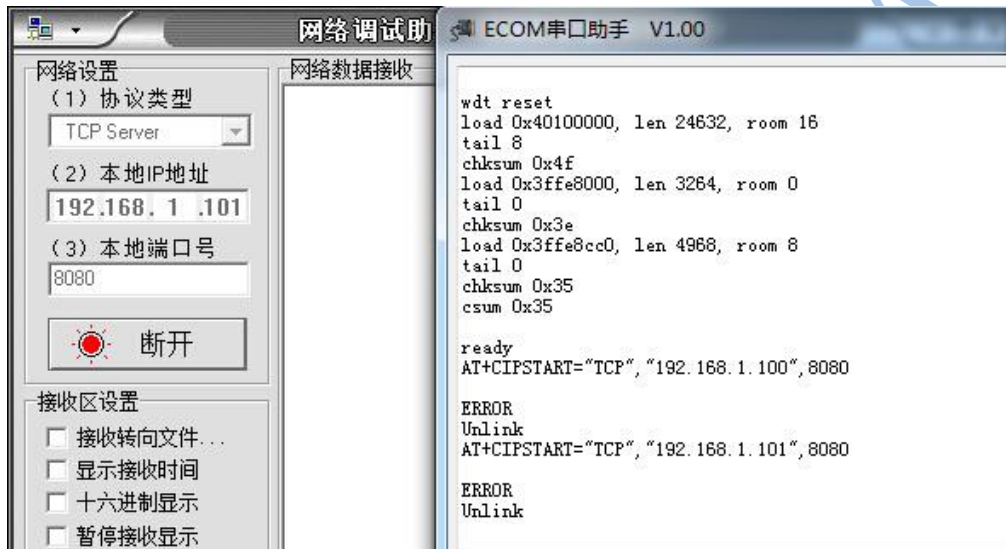
3、 Q&A

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

- (1) Version info or AT: Using “AT+GMR” to get the version info.

Hardware Module info: example AITHINK ESP-01

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



- (3) If possible, please offer the test log, such as:

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)