



ESP8266 AT 指令使用示例

Version 0.4

Espressif Systems IOT Team

Copyright (c) 2015

免责声明和版权公告

本文中的信息，包括供参考的URL地址，如有变更，恕不另行通知。

文档“按现状”提供，不负任何担保责任，包括对适销性、适用于特定用途或非侵权性的任何担保，和任何提案、规格或样品在他处提到的任何担保。本文档不负任何责任，包括使用本文档内信息产生的侵犯任何专利权行为的责任。本文档在此未以禁止反言或其他方式授予任何知识产权使用许可，不管是明示许可还是暗示许可。

Wi-Fi联盟成员标志归Wi-Fi联盟所有。

文中提到的所有商标名称、商标和注册商标均属其各自所有者的财产，特此声明。

版权归© 2015 乐鑫信息科技（上海）有限公司所有。保留所有权利。



Table of Contents

1.	前言.....	4
2.	使用指南	4
2.1.	单连接 TCP Client.....	5
2.2.	透传.....	6
2.3.	多连接 TCP Server.....	8
2.4.	UDP 传输	11
1.	固定远端的 UDP 通信	11
2.	远端可变的 UDP 通信	12
3.	问题反馈	14



1. 前言

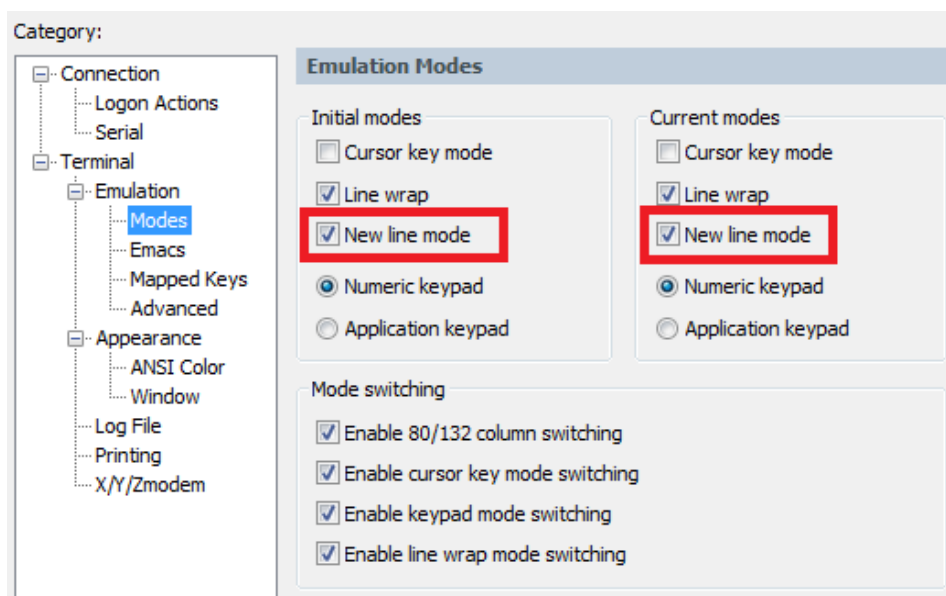
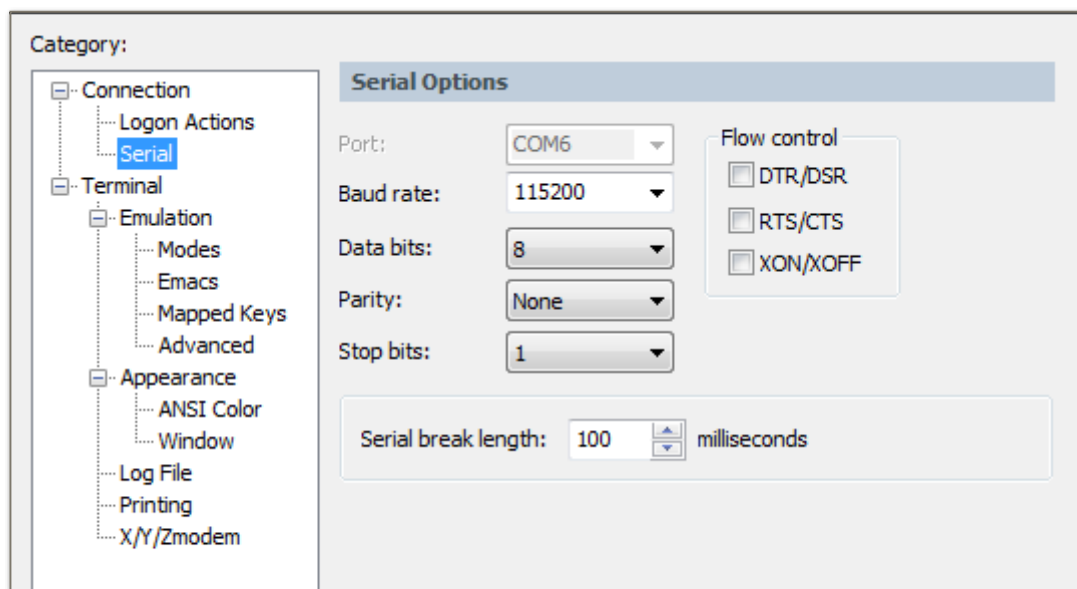
本文介绍几种常见的 Espressif AT 指令使用示例，更多 AT 指令请参考“Espressif AT指令集”。

如遇到 AT 使用问题，请发邮件至 support-at@espressif.com

2. 使用指南

- (1) AT bin 位于 `\esp_iot_sdk\bin\at`，请参考“readme”进行烧录。
- (2) 设备上电。PC 打开串口工具，波特率设置为115200，输入 AT 指令。

注意，AT 指令必须大写，以换行符结束。





2.1. 单连接 TCP Client

- 设置 WiFi 模式:

```
AT+CWMODE=3      // softAP+station mode
Response :OK
```

- 连接路由:

```
AT+CWJAP="SSID", "password"    // SSID and password of router
Response :OK
```

- 查询设备 IP 地址 :

```
AT+CIFSR
Response :192.168.3.106    // Device got an IP from router.
```

- PC 与 ESP8266 连接同一个路由, 在 PC 上使用网络调试助手, 创建一个 TCP 服务器。





- ESP8266 作为 client 连接到 TCP server:

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

- 发送数据:

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

>DGFY // enter the data, no CR
Response :SEND OK
```

注意: 若输入的字节数目超过了指令设定的长度 (n), 则会响应 busy, 并发送数据的前 n 个字节, 发送完成后响应 **SEND OK**.

- 接收数据:

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

2.2. 透传

AT Demo 仅在 ESP8266 作为单连接时, 支持透传。

以下为 ESP8266 作为 station 实现透传的举例, ESP8266 作为 softAP 可参考文档“[Espressif AT 指令集](#)”同理实现透传。

- 设置 WiFi 模式:

```
AT+CWMODE=3 // softAP+station mode
Response :OK
```

- 连接路由:

```
AT+CWJAP="SSID", "password" // SSID and password of router
Response :OK
```

- 查询设备 IP 地址:

```
AT+CIFSR
Response :192.168.101.105 // Device's IP that got from router.
```

- PC 与 ESP8266 连接同一个路由, 在 PC 上使用网络调试助手, 创建一个 TCP 服务器。



- 设备作为 TCP client 连接 TCP server:

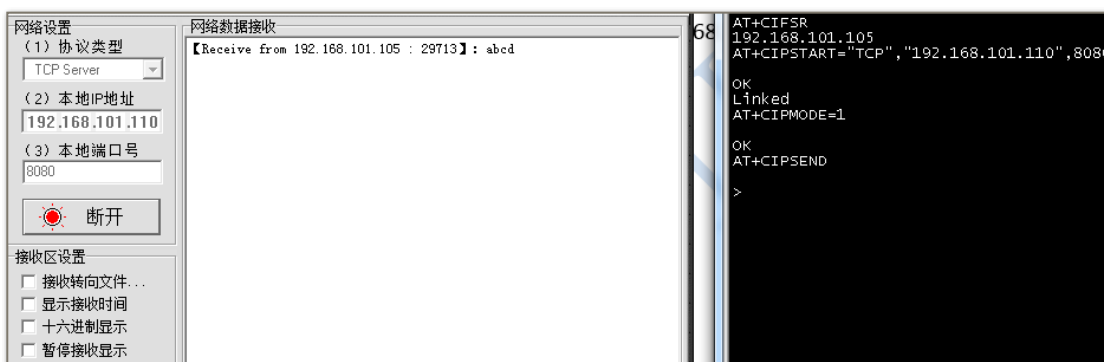
```
AT+CIPSTART="TCP", "192.168.101.110", 8080 // protocol、server IP & port
Response :OK
          Linked
```

- 使能透传模式:

```
AT+CIPMODE=1
Response :OK
```

- 发送数据 :

```
AT+CIPSEND
Response: > //From now on, data received from UART will be
transparent transmitted to server.
```



- 退出发送数据:



在透传发送数据过程中，若识别到单独的一包数据“+++”，则退出透传发送。

如果直接用键盘打字输入“+++”，有可能时间太慢，不被认为是连续的三个“+”，建议可使用如下工具：



字符串输入框：+++

发送新行：不勾选

点击“发送”

注意：“+++”退出透传发送数据，回到正常AT指令模式，TCP连接仍然是保持的，也可以再发AT+CIPSEND指令，开始透传。

- 断开TCP连接:

```
AT+CIPCLOSE
Response :CLOSED   OK
```

2.3. 多连接 TCP Server

目前 AT Demo ESP8266 作为服务器，必须建立多连接，即可以连接多个 TCP client。



以下为 ESP8266 作为 softAP，建立 TCP 服务器的举例:

- 设置 WiFi 模式：

```
AT+CWMODE=3      // softAP+station mode
Response :OK
```

- 使能多连接:

```
AT+CIPMUX=1
Response :OK
```

- 建立 TCP server:

```
AT+CIPSERVER=1    // default port = 333
Response :OK
```

- PC 连入 ESP8266 softAP，PC 作 TCP client 连接设备。



注意: ESP8266 作为 TCP server 有超时机制，如果连接建立后，一段时间内无数据来往，ESP8266 TCP server 会将 TCP client 踢掉。请在 PC 网络工具连上 ESP8266 后建立一个 2s 的循环数据发送，用于保持连接。

- 发送数据:

```
// ID number of connection is defaulted to be 0.
AT+CIPSEND=0, 4  // send 4 bytes to connection NO.0
```



```
>iopd          // enter the data, no CR  
Response :SEND OK
```

注意：若输入的字节数目超过了指令设定的长度 (n)，则会响应 busy，并发送数据的前 n 个字节，发送完成后响应 SEND OK.

- 接收数据:

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

- 断开 TCP 连接:

```
AT+CIPCLOSE=0 // Delete N0.0 connection.  
Response :0, CLOSED OK
```



2.4. UDP 传输

UDP 传输不区分 server 或者 client，由指令 **AT+CIPSTART** 建立传输关系，更多指令说明请参考文档“[Espressif AT指令集](#)”。

- 设置 WiFi 模式：

```
AT+CWMODE=3      // softAP+station mode
Response :OK
```

- 连接路由：

```
AT+CWJAP="SSID", "password"    // SSID and password of router
Response :OK
```

- 查询设备 IP 地址：

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

- PC 与 ESP8266 连接同一个路由，在 PC 上使用网络调试助手，创建 UDP 传输。



下面介绍两种 UDP 通信的示例：

1. 固定远端的 UDP 通信

UDP 通信的远端固定，由“**AT+CIPSTART**”指令的最后参数 0 决定，分配一个连接号给这个固定连接，通信双方不会被其他设备替代。



- 使能多连接:

```
AT+CIPMUX=1
Response :OK
```

- 创建 UDP 传输，例如，分配连接 ID 为 4。

```
AT+CIPSTART=4, "UDP", "192.168.101.110", 8080, 1112, 0
Response :4, CONNECT OK
```

注意：

"192.168.101.110", 8080 为 UDP 传输的远端 IP 和远端 port，也就是 PC 建立的 UDP 配置；

1112 为 ESP8266 的 UDP 本地端口，用户可自行设置，如不设置则为随机值；

0 表示当前 UDP 传输建立后，UDP 远端不会被其他设备更改；即使有其他设备通过 UDP 协议发数据到 ESP8266 UDP 端口 1112，ESP8266 4 号 UDP 传输的远端也不会被替换，使用指令 "AT+CIPSEND=4, X" 发送数据，仍然是当前确定的 PC 端收到。

- 发送数据:

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

>DGFYQ // enter the data, no CR
Response :SEND OK
```

注意：若输入的字节数目超过了指令设定的长度 (n)，则会响应 busy，并发送数据的前 n 个字节，发送完成后响应 SEND OK.

- 接收数据:

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

- 断开 UDP 传输:

```
AT+CIPCLOSE=4
Response :4, CLOSED OK
```

2. 远端可变的 UDP 通信

- 创建 UDP 传输，最后参数为 "2".

```
AT+CIPSTART="UDP", "192.168.101.110", 8080, 1112, 2
Response :CONNECT OK
```

注意：

"192.168.101.110", 8080 为 UDP 传输的远端 IP 和远端 port，就是前述 PC 建立的 UDP 配置；

1112 为 ESP8266 的 UDP 本地端口，用户可自行设置，如不设置则为随机值；



2 表示当前 UDP 传输建立后，UDP 传输远端仍然会更改；UDP 传输远端会自动更改为最近一个与 ESP8266 UDP 通信的远端。

- 发送数据:

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

>DGFYQ          // enter the data, no CR
Response :SEND OK
```

注意: 若输入的字节数目超过了指令设定的长度 (n), 则会响应 busy, 并发送数据的前 n 个字节, 发送完成后响应 **SEND OK**.

- 若需要发 UDP 包给其他 UDP 远端, 只需指定对方 IP 和 port 即可。

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

>abcdef         // enter the data, no CR
Response :SEND OK
```

- 接收数据:

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

- 断开 UDP 通信:

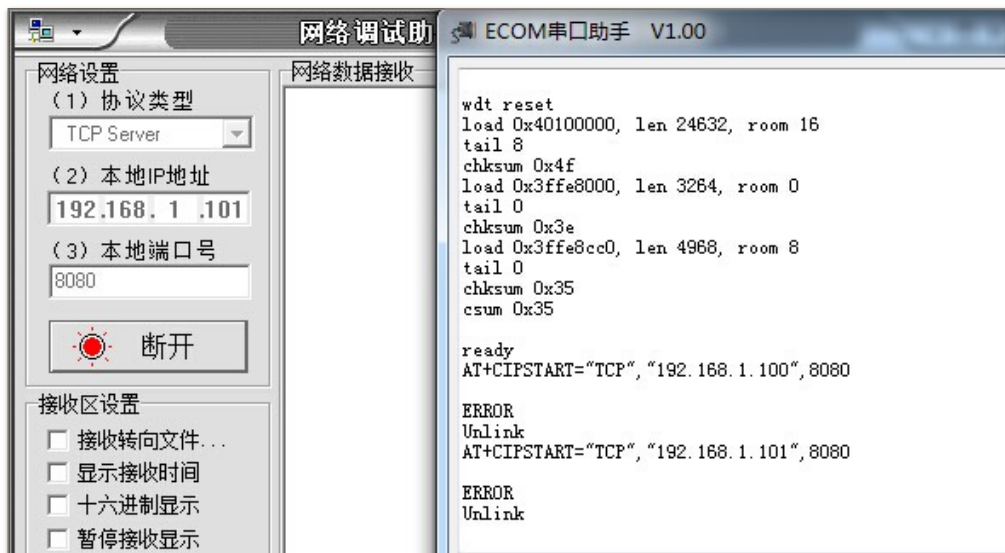
```
AT+CIPCLOSE
Response :CLOSED OK
```



3. 问题反馈

如遇到 AT 使用异常，请发邮件至 support-at@espressif.com，附录如下信息：

- AT 软件的版本号: 指令 “AT+GMR” 可获取版本信息
- 硬件模块的信息: 例如，安信可 ESP-01
- 测试指令步骤说明或截图，例如



- 如能提供 log 打印信息，请附上异常 log 信息，例如

```
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)
```