

ESP8266 AT 指令 使用示例

版本 1.3

版权 © 2017

关于本手册

本文档提供 ESP8266_NONOS_SDK 的 AT 指令使用示例。文档结构如下：

章	标题	内容
第 1 章	概述	介绍 AT 固件及使用配置
第 2 章	单连接 TCP Client	示例 ESP8266 作为 TCP Client 建立单连接通信
第 3 章	UDP 传输	示例如何建立 UDP 通信
第 4 章	透传	示例如何进行透传通信
第 5 章	多连接 TCP Server	示例 ESP8266 作为 TCP Server 通信
第 6 章	问题反馈	AT 指令相关问题的反馈途径与方式。

发布说明

日期	版本	发布说明
2017.08	V1.3	文档格式更新。

目录

1. 概述.....	1
2. 单连接 TCP Client	2
3. UDP 传输.....	4
3.1. 固定远端的 UDP 通信.....	4
3.2. 远端可变的 UDP 通信.....	6
4. 透传.....	8
4.1. TCP Client 单连接透传	8
4.2. UDP 透传.....	10
5. 多连接 TCP Server	12
6. 问题反馈	14



1.

概述

本文档提供 ESP8266_NONOS_SDK 的 AT 指令的几种常见使用示例，更多 AT 指令说明请参考文档 [ESP8266 AT Instruction Set](#)。

- 下载 ESP8266 AT Bin: <http://www.espressif.com/en/support/download/at>
 - 参考 README.md 进行烧录
- PC 串口工具，用于向 ESP8266 发送 AT 指令
 - 波特率设置为 115200
 - AT 指令要求以新行（CR LF）结尾，串口工具应支持“新行模式”

⚠ 注意：

- AT 指令必须为大写英文字母。



2. 单连接 TCP Client

1. 配置 WiFi 模式

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

响应：

```
OK
```

2. 连接路由器

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

响应：

```
OK
```

3. 查询 ESP8266 设备的 IP 地址

```
AT+CIFSR
```

响应：

```
+CIFSR:APIP,"192.168.4.1"  
+CIFSR:APMAC,"1a:fe:34:a5:8d:c6"  
+CIFSR:STAIP,"192.168.3.133"  
+CIFSR:STAMAC,"18:fe:34:a5:8d:c6"  
OK
```

4. PC 与 ESP8266 设备连接同一路由器，在 PC 端使用网络调试工具，建立一个 TCP 服务器。

- 假设，PC 创建的服务器 IP 地址为 192.168.3.116，端口为 8080。

5. ESP8266 设备作为 TCP client 连接到上述服务器

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

响应：

```
OK
```

6. ESP8266 设备向服务器发送数据

```
AT+CIPSEND=4      // set data length which will be sent, such as 4 bytes  
>test             // enter the data, no CR
```

响应：

```
Recv 4 bytes
```



SEND OK

⚠ 注意：

- 发送数据时，如果输入的字节数超过了设置长度 (n) ：
 - 系统将提示 *busy*，并发送数据的前 n 个字节，发送完成后响应 *SEND OK*。
 - 超出长度的部分数据被认为是无效数据，不被接受。

7. 当 ESP8266 设备接收到服务器发来的数据，将提示如下信息：

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



3. UDP 传输

UDP 传输不区分 server 或者 client，由指令 AT+CIPSTART 建立传输。

1. 配置 WiFi 模式

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

响应：

```
OK
```

2. 连接路由器

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

响应：

```
OK
```

3. 查询 ESP8266 设备的 IP 地址

```
AT+CIFSR
```

响应：

```
+CIFSR:APIP,"192.168.4.1"  
+CIFSR:APMAC,"1a:fe:34:a5:8d:c6"  
+CIFSR:STAIP,"192.168.101.133"  
+CIFSR:STAMAC,"18:fe:34:a5:8d:c6"  
OK
```

4. PC 与 ESP8266 设备连接同一路由器，在 PC 端使用网络调试工具，建立一个 UDP 传输。

- 假设，PC 创建的 UDP 自身 IP 地址为 192.168.101.116，端口为 8080。

5. 后文将基于前述步骤，介绍两种 UDP 通信的示例。

3.1. 固定远端的 UDP 通信

UDP 通信的远端固定，由 AT+CIPSTART 指令的最后一个参数设置为 0 决定。系统将分配一个连接号给这个固定连接，UDP 通信双方不会被其他设备替代。

1. 使能多连接

```
AT+CIPMUX=1
```

响应：



```
OK
```

2. 创建 UDP 传输。例如，分配连接号为 4，指令如下：

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

响应：

```
4,CONNECT
```

```
OK
```

说明：

示例指令中的参数说明如下：

- "192.168.101.110", 8080 为 UDP 传输的远端 IP 和端口，即前文步骤 4 中 PC 建立的 UDP 端口；
- 1112 为 ESP8266 本地的 UDP 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 的第 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，仍然是当前固定的 PC 端收到。

3. 发送数据

```
AT+CIPSEND=4,7          // Send 7 bytes to transmission NO.4
```

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

响应：

```
Recv 7 bytes
```

```
SEND OK
```

注意：

- 发送数据时，如果输入的字节数超过了设置长度 (n)：
 - 系统将提示 *busy*，并发送数据的前 n 个字节，发送完成后响应 *SEND OK*。
 - 超出长度的部分数据被认为是无效数据，不被接受。

4. 接收数据。当 ESP8266 设备接收到服务器发来的数据，将提示如下信息：

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

5. 断开 UDP 传输

```
AT+CIPCLOSE=4
```

响应：

```
4,CLOSED
```

```
OK
```




3.2. 远端可变的 UDP 通信

当使用 AT+CIPSTART 指令创建 UDP 通信，将最后一个参数设置为 2 时，UDP 通信的远端可改变。

1. 创建 UDP 传输。

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

响应：

```
CONNECT
OK
```

📖 说明：

示例指令中的参数说明如下：

- "192.168.101.110", 8080 为 UDP 传输的远端 IP 和端口，即前文 PC 建立的 UDP 端口；
- 1112 为 ESP8266 本地的 UDP 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。

2. 发送数据

```
AT+CIPSEND=7          // Send 7 bytes
>UDPtest              // enter the data, no CR
```

响应：

```
Recv 7 bytes
SEND OK
```

⚠ 注意：

- 发送数据时，如果输入的字节数超过了设置长度 (n)：
 - 系统将提示 *busy*，并发送数据的前 n 个字节，发送完成后响应 *SEND OK*。
 - 超出长度的部分数据被认为是无效数据，不被接受。

3. 发送数据到其他指定远端。例如，发数据到 192.168.101.111，端口 1000。

```
AT+CIPSEND=7, "192.168.101.111", 1000      // Send 7 bytes
>UDPtest                                     // enter the data, no CR
```

响应：

```
Recv 7 bytes
SEND OK
```



4. 接收数据。当 ESP8266 设备接收到服务器发来的数据，将提示如下信息：

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

5. 断开 UDP 传输

```
AT+CIPCLOSE
```

响应：

```
CLOSED
```

```
OK
```



4.

透传

ESP8266 AT 默认仅在 TCP client 单连接或 UDP 传输模式时，支持透传。

4.1. TCP Client 单连接透传

以下为 ESP8266 作为 station 实现 TCP client 单连接透传的举例，ESP8266 作为 softAP 可同理实现透传。

1. 配置 WiFi 模式

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

响应：

```
OK
```

2. 连接路由器

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

响应：

```
OK
```

3. 查询 ESP8266 设备的 IP 地址

```
AT+CIFSR
```

响应：

```
+CIFSR:APIP,"192.168.4.1"  
+CIFSR:APMAC,"1a:fe:34:a5:8d:c6"  
+CIFSR:STAIP,"192.168.3.133"  
+CIFSR:STAMAC,"18:fe:34:a5:8d:c6"  
OK
```

4. PC 与 ESP8266 设备连接同一路由器，在 PC 端使用网络调试工具，建立一个 TCP 服务器。

- 假设，PC 创建的服务器 IP 地址为 192.168.3.116，端口为 8080。

5. ESP8266 设备作为 TCP client 连接到上述服务器

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

响应：

```
OK
```



6. 使能透传模式

```
AT+CIPMODE=1
```

响应：

```
OK
```

7. ESP8266 设备向服务器发送数据

```
AT+CIPSEND
```

```
> //From now on, data received from UART will be transmitted to server automatically.
```

⚠ 注意：

- 发送数据时，如果输入的字节数超过了设置长度 (n)：
 - 系统将提示 *busy*，并发送数据的前 n 个字节，发送完成后响应 *SEND OK*。
 - 超出长度的部分数据被认为是无效数据，不被接受。

8. 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。

- 如果使用键盘打字输入“+++”，可能耗时太长，不被认为是连续的三个“+”，建议使用串口工具一次性发送“+++”，并注意不要携带空格或换行符等不可见字符。
- 之后，请至少间隔 1 秒，再发下一条 AT 指令。

⚠ 注意：

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

9. 退出透传模式

```
AT+CIPMODE=0
```

响应：

```
OK
```

10. 断开 TCP 连接

```
AT+CIPCLOSE
```

响应：

```
CLOSED
```

```
OK
```



4.2. UDP 透传

以下为 ESP8266 作为 softAP 实现 UDP 透传的举例，ESP8266 作为 station 可同理实现透传。

1. 配置 WiFi 模式

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

响应：

```
OK
```

2. PC 连入 ESP8266 softAP



3. 在 PC 端使用网络调试工具，建立一个 UDP 传输。

- 假设，PC 创建的 UDP 自身 IP 地址为 192.168.4.2，端口为 1001。

4. ESP8266 与 PC 对应端口建立固定对端的 UDP 传输

```
AT+CIPSTART="UDP", "192.168.4.2", 1001, 2233, 0
```

响应：

```
OK
```

5. 使能透传模式

```
AT+CIPMODE=1
```

响应：

```
OK
```

6. ESP8266 设备向服务器发送数据

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

**⚠ 注意：**

- 发送数据时，如果输入的字节数超过了设置长度（ n ）：
 - 系统将提示 *busy*，并发送数据的前 n 个字节，发送完成后响应 *SEND OK*。
 - 超出长度的部分数据被认为是无效数据，不被接受。

7. 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。

- 如果使用键盘打字输入“+++”，可能耗时太长，不被认为是连续的三个“+”，建议使用串口工具一次性发送“+++”，并注意不要携带空格或换行符等不可见字符。
- 之后，请至少间隔 1 秒，再发下一条 AT 指令。

⚠ 注意：

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

8. 退出透传模式

```
AT+CIPMODE=0
```

响应：

```
OK
```

9. 断开 TCP 连接

```
AT+CIPCLOSE
```

响应：

```
CLOSED
```

```
OK
```



5. 多连接 TCP Server

ESP8266 AT 仅支持建立一个 TCP 服务器，且必须使能多连接，即允许连接多个 TCP client。

以下为 ESP8266 作为 softAP ， 建立 TCP 服务器的举例；如果是 ESP8266 作为 station，可在连接路由后，同理建立服务器。

1. 配置 WiFi 模式

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

响应：

```
OK
```

2. 使能多连接

```
AT+CIPMUX=1
```

响应：

```
OK
```

3. 建立 TCP server

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

响应：

```
OK
```

4. PC 连入 ESP8266 softAP



5. 在 PC 端使用网络调试工具，建立一个 TCP client，连接 ESP8266 的 TCP server。

**⚠ 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。因此，请在 PC TCP client 连上 ESP8266 TCP server 后建立一个 2s 的循环数据发送，用于保持连接。

6. 发送数据

```
// ID number of connection is defaulted to be 0.  
AT+CIPSEND=0, 4    // send 4 bytes to connection NO.0  
>test              // enter the data, no CR
```

响应：

```
Recv 4 bytes  
SEND OK
```

⚠ 注意：

- 发送数据时，如果输入的字节数超过了设置长度 (n)：
 - 系统将提示 *busy*，并发送数据的前 n 个字节，发送完成后响应 *SEND OK*。
 - 超出长度的部分数据被认为是无效数据，不被接受。

7. 接收数据。当 ESP8266 设备接收到服务器发来的数据，将提示如下信息：

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

8. 断开 UDP 传输

```
AT+CIPCLOSE=0
```

响应：

```
0,CLOSED  
OK
```




6.

问题反馈

如遇到 AT 使用异常，请发邮件至[乐鑫技术支持](#)，附上如下信息：

- 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：ESP-WROOM-02；
- 详细的测试步骤说明，例如：

```
AT+CWMODE_CUR=1
OK
AT+GMR
AT version:0.23.0.0(Apr 24 2015 21:11:01)
SDK version:1.0.1
compile time:Apr 24 2015 21:19:31
OK
AT+CIPSTAMAC_DEF="14:CF:11:22:33:05"
OK
```

- 如能提供 log 打印信息，请附上异常 log 信息，例如以下截屏：

```
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)
```



乐鑫 IoT 团队
www.espressif.com

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。蓝牙标志是 Bluetooth SIG 的注册商标。

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 © 2017 乐鑫所有。保留所有权利。