**ESP8266与APP通信**

STA+SERVER模式

--------------------------------------------------------------

**目录**

[第一章 手机端准备](#_Toc8200_WPSOffice_Level1) [2](#_Toc8200_WPSOffice_Level1)

[1.1 APP安装](#_Toc22333_WPSOffice_Level2) [2](#_Toc22333_WPSOffice_Level2)

[1.2 APP参数](#_Toc6750_WPSOffice_Level2) [2](#_Toc6750_WPSOffice_Level2)

[第二章 开发板、模块接线准备](#_Toc22333_WPSOffice_Level1) [4](#_Toc22333_WPSOffice_Level1)

[第三章 代码说明](#_Toc6750_WPSOffice_Level1) [6](#_Toc6750_WPSOffice_Level1)

[3.1 文件移植：](#_Toc6862_WPSOffice_Level2) [6](#_Toc6862_WPSOffice_Level2)

[3.2 实施过程中所用AT指令顺序：](#_Toc31155_WPSOffice_Level2) [6](#_Toc31155_WPSOffice_Level2)

[3.3 调用函数：](#_Toc17239_WPSOffice_Level2) [6](#_Toc17239_WPSOffice_Level2)

[3.4 通信数据判断、执行](#_Toc24232_WPSOffice_Level2) [7](#_Toc24232_WPSOffice_Level2)

[第四章 常见错误排查](#_Toc6862_WPSOffice_Level1) [9](#_Toc6862_WPSOffice_Level1)

[4.1 无法连接至某AP(WiFi）](#_Toc25197_WPSOffice_Level2) [9](#_Toc25197_WPSOffice_Level2)

[4.2 能连接AP(WiFi），但无法连接APP](#_Toc23201_WPSOffice_Level2) [9](#_Toc23201_WPSOffice_Level2)

[4.3 使用小巧门](#_Toc6506_WPSOffice_Level2) [9](#_Toc6506_WPSOffice_Level2)

[第五单 常用AT指令解释](#_Toc31155_WPSOffice_Level1) [11](#_Toc31155_WPSOffice_Level1)

[5.1 AT指令格式](#_Toc4370_WPSOffice_Level2) [11](#_Toc4370_WPSOffice_Level2)

[5.2 常用AT指令解释](#_Toc20357_WPSOffice_Level2) [11](#_Toc20357_WPSOffice_Level2)

# 手机端准备

## 1.1 APP安装

安卓手机，下载APP：“TCP连接”, 安装完成后，图标如下图。



## 1.2 APP参数

手机**先**连接WiFi，**再**打开APP, 记录APP界面中所示: **IP地址**、**端口号**：

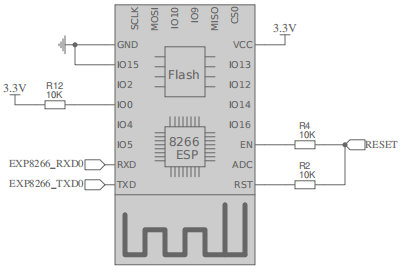


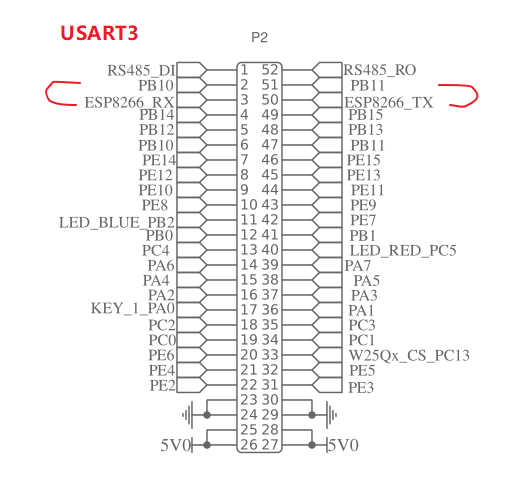
**注意**：此时，APP右上角的连接状态为：连接，即处于未连接状态，无需点击；

# 开发板、模块接线准备

使用串口USART通信，发送AT指令(字符串)，可方便地对ESP8266执行各种常用操作。下面是ESP8266作为通信模块AT指令模式下常用的接线方式：

1：使用带ESP8266的开发板。目前不少开发板，已板载ESP8266模块，如魔女开发板系列，根据电路，选择跳线帽位置，即可接通USART与ESP8266间电路。





如上图所见，示例所用开发板，其右侧排针顶部，已预留USART3的快捷连接位，只需要使用跳线帽，短接PB10与ESP8266\_RX、PB11与ESP8266\_TX， 可使ESP8266方便地连接USAR3所用引脚；



2：使用独立的ESP8266时，亦可按上面图示接线，然后通过UART接入TTL转USB模块，插入电脑，可使用电脑与APP间通信。

独立ESP8266接USB转TTL的具体接线方法，这里略过。

# 代码说明

## 3.1 文件移植：

* **复制文件**：bsp\_ESP8266.c和bsp\_ESP8266.h两个文件，复制至目标工程文件夹;
* **添加文件**：keil左侧文件管理器中，双击某文件夹，以添加bsp\_ESP8266.c到工程中;
* **添加路径**：点击魔术棒工具按钮，在“c/c++”->“Include Paths”里，添加刚才所复制粘贴的bsp\_ESP8266文件的存放路径(指路径，即选择文件夹，不是选择文件）；
* **文件包含**：在所需使用ESP8266功能的文件中，添加：#include “bsp\_ESP8266.h”

## 3.2 实施过程中所用AT指令顺序：

注意, AT指令不区分大小写，以字符串形式+"\r\n"结尾，经串口发送;

1：恢复出厂设置： AT+RESTORE

2：设置工作模式：AT+CWMODE=1 // 1\_STA, 2\_AP, 3\_STA+AP

3：模块重启上电：AT+RST //

4：设置多连接 ：AT+CIPMUX=0 // 0\_单连接，1\_最多5个连接

5：加入AP热点 ：AT+CWJAP=SSID, PassWord

6：TCP 模式连接：AT+CIPSTART="TCP",IP, 端口号

7: 设置传输模式：AT+CIPMODE=1 // 0\_普通， 1\_透明

8: 开始传输数据：AT+CIPSEND // 在透明模式下，无需参数

## 3.3 调用函数：

示例代码中，已把常用AT指令的发送操作，封装成函数。串口的使用，就如普通的串口通信一般，并没有特殊的地主。

函数调用及次序如下：

ESP8266\_Init (UART4, 115200); // ESP8266 初始化ESP8266\_JoinAP("ZHOU", "55025865502586"); // 加入某WiFi热点,

ESP8266\_TCPConnect("192.168.3.33",1234); // 以TCP通信模式，

ESP8266\_EnablePassThrough(); // 开始透明传输

至此，具体的移植操作已完成。编译后，根据错误提示，稍作修改，即可使用STM32开发板，与手机APP通讯。

所附示例代码，在烧录代码后，将自动建立与手机APP的连接。

APP方发送数据，开发板接收后判断各种执行操作，并于串口软件中输出所接收的数据。

同样，开发板发送的数据，APP方，也能于聊天窗口中显示。

注意：“TCP连接”这个APP，不支持中文字的发送(手机与Keil所常设置的汉字编码不同)。

## 3.4 通信数据判断、执行

APP端，可如下图，预先设定各种发送设置，ESP8266收到数据后，经UART发送到STM32芯片的主程序，主程序收到数据后，根据预先设定的格式，可判断后做出各种执行动作，如LED开关，断电器通信，电机控制等。



下图代码示例：接收字符串后，判断执行条件，以操作不同的灯光开关。



# 常见错误排查

## 4.1 无法连接至某AP(WiFi）

* 检查SSID,密码是否正确。
* 距离。WiFi所用的2.4G频率，信号衰退是十分明显的，即使用三根天线的路由器，隔了三四面墙后，信号也剩无几了。调试时，ESP8266距离需要连接的路由器，最好控制在五米以内。排查办法，把烧录好代码的板子，靠近些路由方向(不要超过十米），重新复位，看看能否连接，此方法可以判断是信号问题，还是代码问题；
* 如果时而能连接，里时不能连接，把ESP8266的朝向，更换一下。ESP8266采用板载的PCB天线，信号是比较弱的。这种情况，最好的解决方法，还是更靠近路由；
* 不要使用手机所分享的热点。很多时候会死活连接不上。

## 4.2 能连接AP(WiFi），但无法连接APP

* 手机必须先连接WiFi,再打开APP、
* 只有APP的右上角，显示“连接”状态时，才能连接；如果显示“断开”,表示已打开某连接，还没有关闭。这种情况，最常见于代码调试时，刚才还是连接中的，烧录代码后，就不能连接了。解决方法：点击APP右上角的“断开”，板子重新复位即可。

接线方式

## 4.3 使用小巧门

* 上电后不能立即操作，约稍延时500ms左右待启动完成才能工作；
* 透传必须在单连接模式下：AT+CIPMUX=0
* 查看电脑端口号。如何查看哪些端口号被占用了：快捷键WIN+R, 打开cmd, 输入netstat -ano 。：端口号可自定义，取值范围0~65535。尽量不要使用1023以下的端口号，因为知名端口常使用这段号。建议使用30000~39999间的端口号，这段号比较少用，可有效减少冲突

# 第五单 常用AT指令解释

**5.1**  AT指令格式

* 不区分大小写；
* 指令以'\r'+'\n'结尾，即回车键

## 5.2 常用AT指令解释

AT // 测试

AT+RST // 重启模块

AT+RESTORE // 恢复出厂设置

AT+GMR

AT+CWMODE=? // 1\_STA, 2\_AP, 3\_STA+AP

AT+CWLIF // 查看已接入设备的IP

AT+CIPSTO // 设置服务器超时时间，对方长时间未动作就自动踢除

AT+CIPSTATUS // 查看当前的连接状态

AT+CIPSEND // 发送数据 分两种情况：

当CIPMUX=0: AT+CIPSEND=10; 字节数

当CIPMUX=1: AT+CIPSEND=0,10;ID,字节数

AT+CIPSTART // 新建一个连接

AT+CIPSTART=2,"TCP","192.168.4.1", 554433