

ESP8266 与 APP 通信

STA+SERVER 模式

目录

第一章 手机端准备.....	2
1.1 APP 安装.....	2
1.2 APP 参数.....	2
第二章 开发板、模块接线准备.....	4
第三章 代码说明.....	6
3.1 文件移植：.....	6
3.2 实施过程中所用 AT 指令顺序：.....	6
3.3 调用函数：.....	6
3.4 通信数据判断、执行.....	7
第四章 常见错误排查.....	9
4.1 无法连接至某 AP(WiFi)	9
4.2 能连接 AP(WiFi)，但无法连接 APP.....	9
4.3 使用小巧门.....	9
第五章 常用 AT 指令解释.....	11
5.1 AT 指令格式.....	11
5.2 常用 AT 指令解释.....	11

第一章 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：

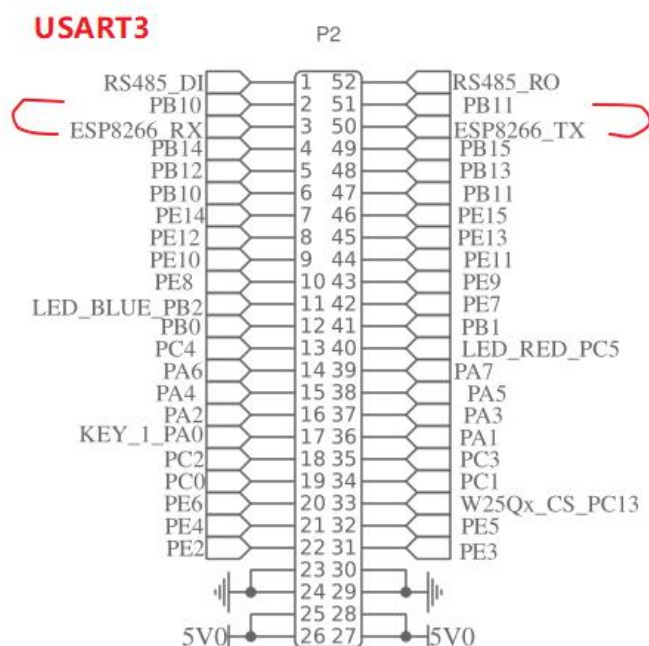
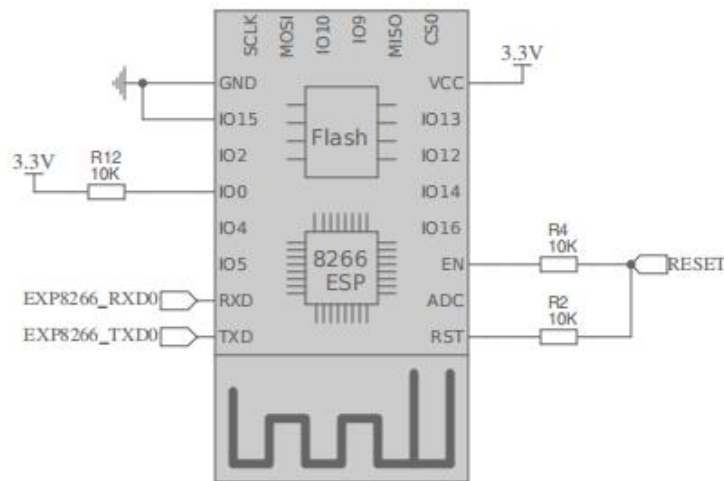


注意：此时，APP 右上角的连接状态为：连接，即处于未连接状态，无需点击；

第二章 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。

第三章 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。

```
// 监听，ESP8266所发过来的数据，并判断处理
if(ESP8266_CheckReceivedNum())
{
    static uint16_t CNT=0;
    printf("<<<<<< 来自ESP8266的第%d帧数据，共%d个字节：<<<<<<\r\n", ++CNT, xESP8266.ReceivedNum);
    printf("%s\r\n", xESP8266.ReceivedBuffer);

    // 判断收到的数据
    if(strstr((char*)xESP8266.ReceivedBuffer, "LED_RED_ON" )) LED_RED_ON;
    if(strstr((char*)xESP8266.ReceivedBuffer, "LED_RED_OFF" )) LED_RED_OFF;
    if(strstr((char*)xESP8266.ReceivedBuffer, "LED_BLUE_ON" )) LED_BLUE_ON;
    if(strstr((char*)xESP8266.ReceivedBuffer, "LED_BLUE_OFF")) LED_BLUE_OFF;
    if(strstr((char*)xESP8266.ReceivedBuffer, "LED_ALL_ON")) {LED_RED_ON; LED_BLUE_ON; }
    if(strstr((char*)xESP8266.ReceivedBuffer, "LED_ALL_OFF")) {LED_RED_OFF; LED_BLUE_OFF; }

    if(strstr((char*)xESP8266.ReceivedBuffer, "CLOSED")) printf("ESP8266与目标的通信连接已断开");
    ESP8266_CleanReceivedFlag(); // 处理完后，清零ESP8266的接收缓存，方便下一帧数据接收
}
}
```


第四章 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