



开源共享

携手共进

深圳普中科技有限公司

官方网站: www.prechin.cn

技术论坛: www.prechin.net

技术 QQ: 2489019400

咨询电话: 0755-61139052

ESP8266-WIFI 模块使用教程

1.1 ESP8266 简介

ESP8266 是一个完整且成体系的 Wi-Fi 网络解决方案，能够搭载软件应用，或通过另一个应处理器卸载所有 Wi-Fi 网络功能。我们使用的 ESP8266 是串口型 WIFI，速度比较低，不能用来传输图像或者视频这些大容量的数据，主要应用于数据量传输比较少的场合，比如温度信息，一些传感器的开关量等。当然传输的数据量虽说少，但也能一次传输几千字节的数据，而且通信非常稳定，可以满足大多数应用。

1.2 ESP8266-WIFI 模块介绍

安信可推出的 ESP8266-WIFI 模块有很多，它们的使用方法都大同小异，我们普中 STM32 开发板预留了一个 ESP8266-WIFI 模块接口，大家只要将此模块插入开发板接口即可做此实验，无需额外连线。我们普中推出了 1 款 WIFI 模块，如图：



与通用的 WIFI 模块一样使用，这里我们就以通用的 WIFI 模块介绍。如图 1.2.1 所示：



图 1.2.1 ESP8266-WIFI 模块

WIFI 模块插在我们开发板上如图：

URXD: 串口接收管脚, 与开发板上串口 2 的 TXD 相连。

GPI00: GPI00 为高电平代表从 FLASH 启动, GPI00 为低电平代表进入系统升级状态, 此时可以经过串口升级内部固件, 这里我们不需要对此管脚操作。

GPI02: 此管脚为 ESP8266 引出的一个 IO 口, 这里我们不需要对此管脚操作。

GND: GND 管脚, 开发板上丝印已经标了。

其实我们不需要使用这么多管脚, 只需要使用 WIFI 模块的串口 UTXD、URXD 管脚、RST 和 CH_PD 管脚即可, 其他的不用管。

WSP8266-WIFI 模块接口与 STM32 芯片管脚连接图如下: (开发板内线路已连接好, 所以直接插上模块即可)

ESP8266-WIFI接口	STM32芯片管脚
VCC	3.3V
RST	PF6
CH_PD	PC7
UTXD	PB11 (串口3 RXD)
URXD	PB10 (串口3 TXD)
GPI00	PC0 (未使用)
GPI02	PG9 (未使用)
GND	GND

注意: 由于 COM3 模块中使用的串口也是串口 3, 开发板已经做好了切换短接片, 如果在做 WIFI 实验例程的时候一定要将 COM3 模块旁边的短接片短接到 COM3M 端, 这样串口 3 管脚就和 WIFI 模块串口相连。

ESP8266-WIFI 模块支持 STA/AP/STA+AP 三种工作模式。

STA 模式: ESP8266 模块通过路由器连接互联网, 手机或电脑通过互联网实现对设备的远程控制。

AP 模式: 默认模式 ATK_ESP8266 模块作为热点, 实现手机或电脑直接与模块通信, 实现局域网无线控制。

STA+AP 模式: 两种模式的共存模式, 即可以通过互联网控制可实现无缝切换, 方便操作。

ESP8266-WIFI 模块仅在单连接状态, 支持透传模式 (TCP Client 透传和 UDP 透传)。本教程所讲解的就是采用 TCP Client 透传, 其他的模式操作, 大家可以在此基础上继续深入了解。

1.3 ESP8266 常用指令

ESP8266-WIFI 模块采用的是 AT 指令操作, ESP8266 的指令非常多, 但是我们不需要全部掌握, 只需把常用的了解即可, 遇到不会的还可以查看“\ESP8266-WIFI 参考资料”ESP8266 AT 指令集文档, 里面详细介绍了各个指令的功能和使用方法。下面就简单介绍下这些指令。

1.3.1 基础 AT 指令

基础 AT 指令如图 1.3.1 所示:

AT	测试指令
AT+RST	重启模块
AT+GMR	查看版本信息
ATE	开关回显功能
AT+RESTORE	恢复出厂设置
AT+UART	设置串口配置

图 1.3.1 基础 AT 指令表

(1) AT 测试指令

AT 测试 AT 启动	
执行指令 AT	响应
	OK
	参数说明

(2) AT+RST

AT+RST 重启模块	
执行指令 AT+RST	响应
	OK
	参数说明

(3) AT+GMR

AT+GMR 查看版本信息	
执行指令 AT+GMR	响应
	<AT version>AT 版本
	<SDK version>SDK 版本
	<company>发布公司
	<date>发布时间
	OK

(4) ATE

ATE 开关回显功能	
执行指令 ATE	响应
	OK
	参数说明
	ATE0 关闭回显
	ATE1 开启回显

(5) AT+RESTORE

AT+RESTORE 恢复出厂设置功能	
执行指令 AT+RESTORE	响应 OK
说明	恢复出厂设置，将所有保存的参数恢复到出厂默认参数。 注意：恢复出厂设置模块会重启

(6) AT+UART

AT+UART 设置串口配置	
执行指令 AT+UART=<baudrate>,<databits>,<stopbits>,<parity>,<flow control>	响应 OK
	参数说明 <baudrate> 串口波特率
	<databits> 数据位 5: 5 bit 数据位 6: 6 bit 数据位 7: 7 bit 数据位 8: 8 bit 数据位 <stopbits> 停止位 1: 1 bit 停止位 2: 1.5 bit 停止位 3: 2 bit 停止位 <parity> 校验位 0: None 1: Odd 2: EVEN <flow control> 流控 0: 不使能流控 1: 使能 RTS 2: 使能 CTS 3: 同时使能 RTS 和 CTS
说明	1. 本设置将保存在 Flash user parameter 区，重新上电后 仍生效。 2. 使用流控需要硬件支持流控，MTCK 为 UART0 CTS, MTDO 为 UART0 RTS 3. 波特率支持范围：110~921600
举例	AT+UART=921600,8,1,0,0

1.3.2 WIFI 功能 AT 指令

WIFI 功能 AT 指令如图 1.3.2 所示：

指令	描述
AT+CWMODE	选择 WIFI 应用模式
AT+CWJAP	加入 AP
AT+CWLAP	列出当前可用 AP
AT+CWQAP	退出与 AP 的连接
AT+CWSAP	设置 AP 模式下的参数
AT+CWLIF	查看已接入设备的 IP
AT+CWDHCP	设置 DHCP 开关
AT+CWAUTOCONN	设置 STA 开机自动连接到 wifi
AT+CIPSTAMAC	设置 STA 的 MAC 地址
AT+CIPAPMAC	设置 AP 的 MAC 地址
AT+CIPSTA	设置 STA 的 IP 地址
AT+CIPAP	设置 AP 的 IP 地址
AT+CWSMARTSTART	启动智能连接
AT+CWSMARTSTOP	停止智能连接

图 1.3.2 WIFI 功能 AT 指令表

里面具体的指令大家可以查看“\ESP8266-WIFI 参考资料”ESP8266 AT 指令集文档，里面详细介绍了各个指令的功能和使用方法，这里就不截图出来。

1.4 ESP8266-WIFI 实验

了解了 WIFI 模块的 AT 指令格式及功能，下面我们就可以使用 STM32 串口与 WIFI 模块串口进行通信了。本教程我们要实现 STA 模式的 TCP Client 透传。简单理解就是配置 WIFI 模块为 STA 模式，让 WIFI 模块连接路由器或者笔记本发出的 wifi，然后在网络调试助手上设置好连接 WIFI 的 IP 地址和端口，这时 STM32 串口 3 发送的数据就可以经过 WIFI 模块传输到网络调试助手上显示，即实现了 TCP Client 透传。

这里需要注意：WIFI 模块默认的时候串口波特率是 115200, 8 位数据，1 位停止位，因此我们在配置 STM32 串口的时候也要按这个模式配置串口，否则不能进行数据传输。将 COM3 模块旁的短接片短接到 COM3M 端。

1.4.1 ESP8266-WIFI 数据透传实验

本实验所实现的功能：在局域网中，WIFI 模块配置为 STA 模式，做客户端，电脑做服务端，客户端往服务端不断发数据，每间隔 1 秒发送一串“www.prechin.cn”字符，发送的字节数为 14 字节。

要实现此功能程序，首先要初始化 ESP8266-WIFI 模块所用的 IO 口及时钟，其中包括串口配置，初始化后即进入 STA 模式透传，配置好 STA 模式，连接由电脑或路由器发出的 wifi 后，开始传输数据。

(1) 修改 WIFI 名称、密码及 IP

在下载程序到开发板前，需要看下你所在的局域网的 WIFI 名称和你电脑的 IP，然后打开工程，编译后找到 sta_tcpclent_test.h 文件（在 sta_tcpclent_test.c 内），把里面的宏参数修改为你所在的局域网的 WIFI 名称和你电脑的 IP，代码如下：

```
#define User_ESP8266_SSID "PUZHONG" //要连接的热点的名称
```

```
#define User_ESP8266_PWD "puzhong168" //要连接的热点的密码
```

```
#define User_ESP8266_TCPServer_IP "192.168.191.1" //要连接的服务器的 IP
```

```
#define User_ESP8266_TCPServer_PORT "8080" //要连接的服务器的端口
```

我们程序里面的这 4 个宏参数，是针对我们使用的局域网 WIFI 名称、密码和电脑 IP，如果你要实现与你的电脑进行透传，就需要将此部分进行修改。至于怎么查看你电脑 IP，请百度。

(2) 设置串口调试助手与网络调试助手

修改完这部分程序后，将工程编译下载到开发板内（记得要插上你的 WIFI 模块）。打开串口调试助手和网络调试助手，此软件在“\调试工具”目录下。将串口调试助手波特率设置为 115200，数据位 8，停止位 1，**特别注意，由于此串口助手刚打开会把我们开发板一直拉到复位状态，所以需要勾选下 DTR，然后再取消掉，即可有信息返回到串口助手上**，如图 1.4.1 所示：

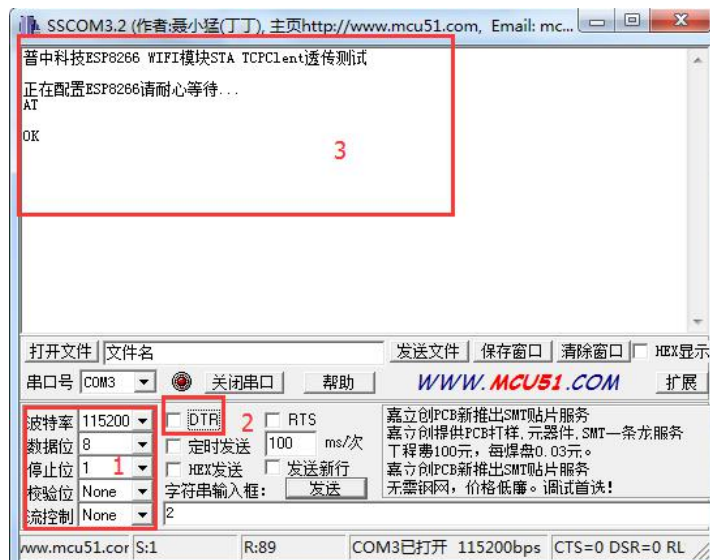
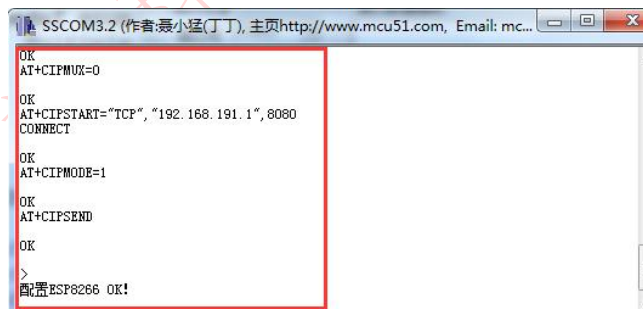


图 1.4.1 串口调试助手设置

然后打开网络调试助手，选择协议类型为 TCP Server，本地 IP 地址和端口设置为前面 sta_tcpclnt_test.h 文件内配置的 IP 地址及端口（要保证前面配置的是你的电脑 IP 地址和端口），选择开始监听，当串口助手上显示配置 ESP8266 OK，如下：



然后就会接收到由开发板串口 2 发送的“www.prechin.cn”信息，间隔时间为 1 秒。如果没有接收到，请重新复位下开发板，直到连接成功，接收到数据。

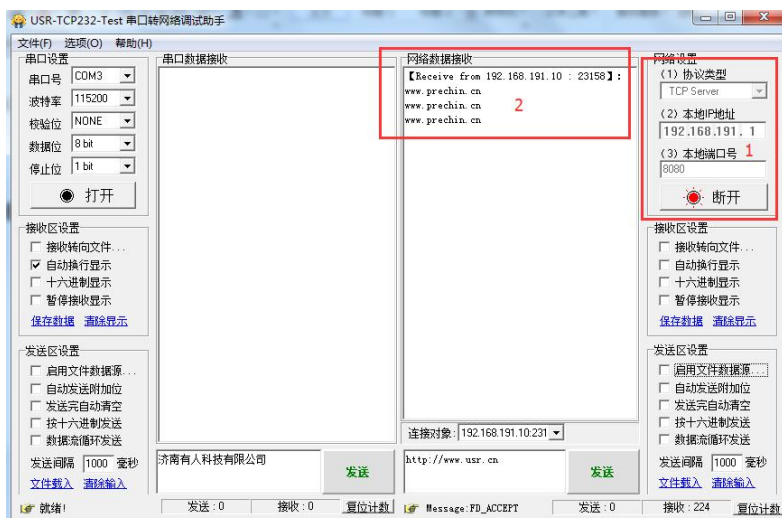


图 1.4.2 网络调试助手设置

具体代码可以打开“\ESP8266-WIFI 实验例程\ESP8266-WIFI 数据透传实验”查看。

1.4.2 ESP8266-WIFI 温度传输实验

其实这个实验和上一个实验基本上是一样的，只不过这个实验传输的是 DS18B20 温度传感器检测的温度值，将温度数据通过 WIFI 模块发送到网络调试助手上进行显示（要将 DS18B20 温度传感器插到开发板上 DS18B20 接口处，注意方向）。具体的操作步骤同上，实验现象如图 1.4.3 所示：



图 1.4.3 温度数据显示

具体代码可以打开“\ESP8266-WIFI 实验例程\ESP8266-WIFI 温度传输实验”查看。