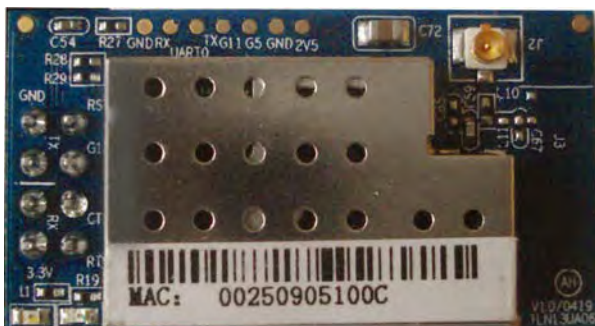


HC-21 WiFi 串口通信模块

用户手册 V1.1



简介

HC-21 (TLN13UA06) 是汇承科技推出的全新的第三代嵌入式 UART-WiFi 模块产品。

UART-WiFi 是基于UART接口的符合 WiFi 无线网络标准的嵌入式模块，内置无线网络协议 IEEE802.11 协议栈以及 TCP/IP 协议栈，能够实现用户串口数据到无线网络之间的转换。通过 UART -WiFi模块，传统的串口设备也能轻松接入无线网络。

此外，HC-21 (TLN13UA06) 模块除作为STA接入传统的 Infra 网络和 Ad-hoc 网络外，本模块还支持软 AP 模式，网卡工作在 AP 模式时能够提供智能手机等手持终端的接入功能，同时为改善 AP 模式下的用户体验，本模块在实现软 AP的同时实现了 DHCP Server 和 DNS Server，以提供智能终端的零配置接入，但需要注意的是，本模块所实现 DNS Server 仅提供本模块域名的解析功能。

相比前两代产品，HC-21 (TLN13UA06) 在软硬件方面进行了全面升级，功能更加强大，使用更加简单：

- ◇ 全面更优的串口透明数据传输模式，真正实现串口的即插即用
- ◇ 更小的功耗：430mW（典型）
- ◇ 更小的尺寸：20mm×37mm
- ◇ 提供服务器软件、手机 APP 应用软件设计定制服务，应用软件平台可灵活可选。

产品特性

接口

- ◇ 双排插针式接口：HDR254M-2X4
- ◇ 支持波特率范围：1200~115200bps
- ◇ 支持硬件 RTS/CTS 流控
- ◇ 单 3.3V 供电，可提供不小于 300mA 的直流电源

无线

- ◇ 支持 IEEE802.11b/g/i 无线标准
- ◇ 支持两种无线网络类型：
基础网（Infra）和自组网（Ad-hoc）
- ◇ 支持软 AP，最多支持 4 个 STA 连接
- ◇ 支持多种安全认证机制：
OPEN/WEP64/WEP128、TKIP/CCMP(AES)
OPEN/WEP/WPA-PSK/WPA2-PSK
- ◇ 支持快速联网（指定信道与 BSSID）
- ◇ 支持无线漫游
- ◇ 支持协议节能（PS-POLL 方式）
- ◇ 支持网络不存在时自动创建

其它

- ◇ 支持多种网络协议：
TCP/UDP/ARP/ICMP/DHCP/DNS/HTTP
- ◇ 支持 DHCP Server、DNS Server
- ◇ 支持自动和命令两种工作模式
- ◇ 支持串口透明传输模式
- ◇ 支持 AT+控制指令集
- ◇ 支持多种参数配置方式：
串口/WEB 服务器/无线配置卡
- ◇ 内置友好的 Web 配置页面
- ◇ 通过 CE / FCC / WiFi / ROHS 认证

应用领域

- ◇ 智能家电、智能家居
- ◇ 医疗监护、智能玩具
- ◇ 汽车电子、工业控制
- ◇ 智能电网、物联网



目录

1. 产品简介.....	4
1.1 技术规格.....	4
1.2 硬件介绍.....	5
1.2.1 机械尺寸.....	5
1.2.2 插针接口.....	6
1.2.3 电器连接注意事项.....	6
1.2.4 LED 指示灯说明.....	6
1.2.5 外部天线.....	7
1.3 软件介绍.....	7
1.3.1 自动工作模式.....	7
1.3.1.1 自动组帧机制.....	7
1.3.1.2 硬件流控机制.....	8
1.3.1.3 退出串口透明传输模式.....	8
1.3.2 命令工作模式.....	8
2. 功能描述.....	9
2.1 网络拓扑.....	9
2.1.1 自组无线网络 (Ad-hoc)	9
2.1.2 基础无线网络 (Infra)	9
2.2 安全机制.....	9
2.3 快速联网.....	9
2.4 地址绑定.....	10
2.5 无线漫游.....	10
2.6 TCP/IP 协议栈.....	10
2.7 灵活参数配置.....	10
2.8 固件升级.....	11
3. 快速使用向导.....	11
3.1 快速组建测试平台.....	11
3.1.1 组建原理.....	11
3.1.2 搭建平台.....	11
3.1.3 收发测试.....	12
3.1.3.1 默认模块参数测试.....	12
3.1.3.2 配置模块参数测试.....	12
3.1.3.3 手机平台测试.....	13
3.2 参数配置.....	14
3.2.1 通过网页浏览器配置.....	14
3.2.2 通过配置管理程序配置.....	14
3.2.2.1 通过串口连接.....	14
3.2.2.2 通过 WIFI 连接.....	14
3.2.2.3 UART-WIFI 配置管理程序使用说明.....	15
3.2.3 通过 AT 指令配置.....	16
3.3 编程使用原理.....	16
3.3.1 串口端编程.....	16
3.3.2 网络端编程.....	16



4. 典型应用实例.....	16
4.1 主动型串口设备联网.....	16
4.2 被动型串口设备联网.....	17
4.3 广播方式的串口设备联网.....	17
4.4 无线虚拟串口.....	18
4.5 多网络连接模式.....	18
4.6 软 AP 模式.....	19
5. AT 指令协议集.....	19
5.1 语法说明.....	19
5.1.1 处理流程.....	19
5.1.2 语法格式.....	20
5.1.3 格式范例.....	21
5.1.4 错误代码.....	21
5.2 指令集.....	21
5.2.1 指令列表.....	21
5.2.2 网络控制类.....	22
5.2.3 系统控制类.....	25
5.2.4 参数设置类.....	26
5.2.5 出厂默认设置.....	32
5.3 例程代码.....	33
5.3.1 例程代码说明.....	33
5.3.2 命令模式下 WiFi 网卡的操作流程.....	33
5.3.3 示例代码主要的数据结构.....	34
5.3.3.1 AT 命令结构体.....	34
5.3.3.2 串口 AT 指令发送标志.....	34
5.3.3.3 串口等待 AT 指令回复标志.....	34
5.3.4 串口数据操作.....	34
5.3.4.1 串口发送指令示例.....	34
5.3.4.2 串口接收 AT 指令回复示例.....	34
5.3.5 AT 指令返回参数解析函数.....	37
5.3.6 字符串操作函数.....	38
5.3.7 AT 指令发送解析示例.....	39
5.3.7.1 简单控制类指令发送示例.....	39
5.3.7.2 参数设置类指令发送示例.....	41
5.3.7.3 较复杂返回参数的 AT 指令示例.....	43
附录 常见问题与解答.....	46



1. 产品简介

1.1 技术规格

表 1-1 产品技术规格

	项目	参数
无线参数	无线标准	IEEE802.11 b/g/i
	频率范围	2.412 GHz ~ 2.484 GHz
	天线接口	IPX 微型天线接口 / 板载 PCB 天线
	调制方式	DSSS, OFDM, DBPSK, DQPSK, CCK, QAM16/64
	数据速率	802.11b: 1, 2, 5.5, 11 Mbps 802.11g: 6, 9, 12, 18, 24, 36, 48, 54 Mbps
	发射功率	802.11b: 15±2 dBm @ 11M (typical) 802.11g: 10±2 dBm @ 54M (typical)
	接收灵敏度	802.11b: -82 dBm @ 11Mbps (typical) 802.11g: -68 dBm @ 54Mbps (typical)
硬件参数	接口类型	标准 3.3V TTL 电平 UART 接口
	接口数率	1200bps ~ 115200bps
	工作电压	3.3±0.3 V
	平均功耗	420mw (典型)
	工作湿度	5%~90% (无凝结)
	存储温度	-55℃~+125℃
	工作温度	-20℃~+70℃
	外形尺寸	20mm×37mm
软件参数	网络类型	Infra / Ad-hoc / AP
	安全制式	WEP / WPA-PSK / WPA2-PSK
	工作模式	WEP64 / WEP128 / TKIP / CCMP(AES)
	串口指令	AT+指令集
	网络协议	TCP / UDP / ARP / ICMP / DHCP / DNS / HTTP
	Socke 连接	最大连接数: 15
	TCP 连接	最大连接数: 16 最大 Client 数: 16 最大 Server 数: 4 本端 Server 最大接入 Client 数: 4
	UDP 连接	最大连接数: 16

1.2.2 插针接口

本产品提供图 1-3 接口示意图所示的双列直插 8 针引脚，引脚功能定义如表 1-2 所示：

表 1-2 模块插针接口

8	GND	Power	电源地
7	TXD	Out	串口数据发送脚
6	RXD	In	串口数据接收脚
5	VDD	Power	3.3V 电源输入脚（LED②连接此脚）
4 (多功能)	nRTS	Out	RTS 信号，在自动工作模式下，作为本端的 RTS 信号可以直接与对端的 CTS 信号连接.
	nREADY	Out	在命令工作模式下，表示模块的无线网络连接状态，低电平表示已连接，高电平表示未连接
3 (多功能)	nCTS	In	CTS 信号，在自动工作模式下，作为本端的 CTS 信号可以直接与对端的 RTS 信号连接
	GPIO	In/Out	根据用户设置可以作为通用输入、输出管脚
2	Link	Out	无线传输指示（LED②连接此脚）
1	NRESET	In	复位，低电平有效

1.2.3 电器连接注意事项



图 1-4 硬件连接图

- ◇ 模块供电电压为直流 3.3V，接口信号电压为标准串口 TTL 电平。
- ◇ 模块最大工作电流约 250mA，在设计供电电路时需注意满足模块功耗。
- ◇ nRESET 引脚可悬空，但仍建议将此引脚与 MCU 连接，以便必要时可对模块进行复位操作。

注：引脚（8~5）用于标准 TTL 电平串口通讯，引脚（4~1）为可选功能引脚。

在不使用多功能的情况下将引脚悬空即可，不影响模块正常工作。

1.2.4 LED 指示灯说明

LED①：电源指示灯。

LED②：Link 指示灯。在 WIFI 未联网的情况下为低频率闪烁，在 WIFI 联网情况下为高频率闪烁。



1.2.5 外部天线

HC-21 (TLN13UA06) 模块提供板载 PCB 微带天线和外置 IPX 全向天线接口以适用于不同环境下的应用。在使用外置 IPX 全向天线时，需采用符合 IEEE 802.11 标准的 2.4GHz 天线，天线的参数要求如表1-3所示：

表1-3 外置天线匹配参数

频带 Frequency Range(MHz)	2.4GHz ~ 2.5GHz
驻波比 V.S.W.R	<1.5
极化 Polarization	Vertical
输入阻抗 Impedance(Ω)	50
接头形式 Connector Type	IPX
工作温度 Operating Temperature($^{\circ}\text{C}$)	-40~+70
储藏温度 Reposition Temperature($^{\circ}\text{C}$)	-55~+85

1.3 软件介绍

1.3.1 自动工作模式

自动工作模式即串口透明传输模式，也是模块的默认工作模式。此模式可实现串口即插即用功能，最大程度降低用户使用难度。用户可根据自己的实际情况对模块参数进行配置，参数将保存在模块 Flash 中，掉电数据不会丢失。上电启动后模块按预设的配置自动连接至指定无线网络及服务器，并按预配置的串口参数连接串口设备。在此模式下，用户按自己的需求配置好以下各类参数（未使用的功能，使用默认配置即可）：

- ◇ 无线网络参数，包括网络名称（SSID）、安全模式、密钥。
- ◇ TCP/UDP 连接参数，包括协议类型、连接类型、目的地址、目的端口。
- ◇ 串口参数，包括波特率、数据位、检验位、停止位、硬件流控。

由于在自动工作模式下，模块的串口始终工作在透明数据传输状态，因此用户只需把它看做一条虚拟的串口线，依照使用普通串口的方式发送和接收数据就即可。也就是说，用户直接在原有的串口设备上，把连接的串口线替换成本模块，原串口设备无需作任何改动即可轻松实现数据的无线传输。

1.3.1.1 自动组帧机制

在透明传输状态下，在串口上所有的数据都是以字节流的形式进行传输的，而在网络上，数据却是以固定的帧格式的形式传输，因此，数据在串口与网络之间流动时就需要一个流与帧之间的转换过程。对于从网络到串口的方向的数据传输来说，数据由帧转换成流是非常容易的。而相反的从串口到网络的数据传输方向，即数据由流转换成帧格式的时候，就需要一定的规则来约束。详细配置参见AT+ATLT。

本模块使用两个参数来指定转换规则：

◇ 自动组帧长度：当模块从串口接收到的数据长度达到了自动组帧长度时，模块将触发组帧操作，并将组好的数据帧发送到网络上。

◇ 自动组帧周期：在某些情况下，如发送的数据量比较小，导致缓冲区中的数据一直填不满一个数据块，导致缓冲区中的数据无法正常发送出去。这时候就需要一个超时机制来保证少量数据也能正常发送，这个超时时间就是自动组帧周期。

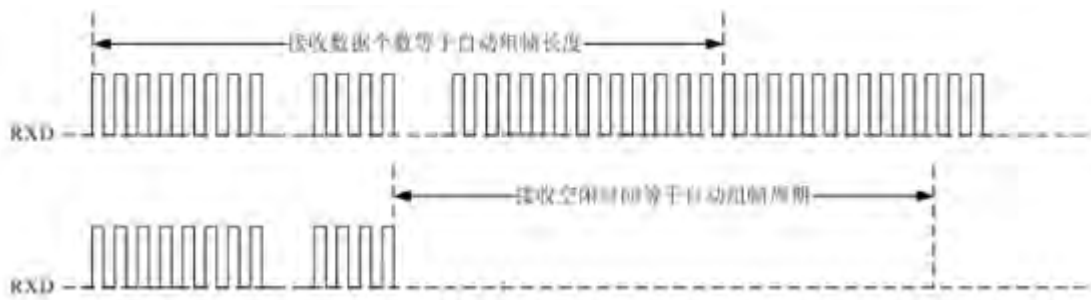


图 1-7 自动组帧示意图

上述两个参数共同组成了自动组帧的两个触发条件，自动组帧长度越大，系统的传输性能越高，自动组帧周期越短，系统的传输实时性越高，因此在实际应用中，可以根据具体的需要来调整两个参数值，从而获得最佳的传输效果。详细配置参见 AT+ATPT。

1.3.1.2 硬件流控机制

在串口透明传输模式下，模块串口传输支持硬件流控方式，提供标准的 RTS/CTS 信号，强烈建议用户在进行大数据量传输时启用硬件流控功能，这样可以充分保证数据的可靠传输。对于不需要流控功能的应用场合，用户只需把 RTS/CTS 引脚悬空即可。

基于 RTS/CTS 的硬件流控使用机制如下：

- ◇ RTS：模块接收使能信号，低电平有效，当同时满足以下条件时，RTS 信号有效
 - ◆ 条件一：无线网络已连接
 - ◆ 条件二：UART 接收缓冲区中的数据长度低于 3/4 满
- ◇ CTS：主机接收使能信号，低电平有效，模块在检测到 CTS 信号无效后立即停止向串口发送数据，直到 CTS 变为有效

1.3.1.3 退出串口透明传输模式

在串口透明传输模式下，模块仅将收发数据进行 TCP 协议与串口协议的转换处理，是一种纯粹的数据传输形式，不做其他任何数据解析，因此收发数据无法对模块进行配置操作。此时如需对模块进行配置，需要先退出透明传输模式，模块退出透明传输模式条件如图 1-8 所示。

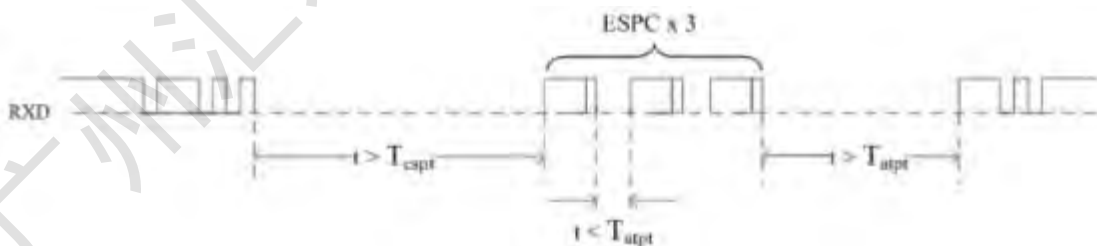


图 1-8 退出透明传输模式示意图

其中：ESCP 表示逃逸字符（默认是“+”）、 T_{escpt} 表示逃逸时间、 T_{atpy} 表示自动组帧周期。详细配置参见 AT+ESCP。模块检测到符合图 1-8 所示的退出条件后，反馈输出“+OK”，确认切换到命令工作模式。

1.3.2 命令工作模式

命令工作模式下可对模块串口发送 AT 指令进行参数配置，包括修改配置参数、控制联网、控制 TCP/IP 连接、数据传输等。这是一种高级的使用方式，用户可依据不同的应用场景进行相应的配置修改。但是，此模式也需要用户对模块的 AT+ 指令控制协议有充分的了解，并且具备基本的无线网络以及 TCP/IP 网络的使用知识。

本模块内置了一套用于系统控制及参数配置指令的 AT+ 指令集，所有指令均基于 ACSII 编码，使用 Windows 系统自带的超级终端程序或串口助手等工具，即可直接对模块 UART 端进行命令控制，方便用户调试和使用。

详细配置方式请参考第 5 章 AT 指令协议集。

2. 功能描述

2.1 网络拓扑

WiFi 无线网络包括两种拓扑形式：基础网（Infra）和自组网（Ad-hoc），要说明无线网络的拓扑形式，首先了解两个基本概念：

◇ AP：即无线接入点，是一个无线网络的创建端，也是网络的中心节点。我们家庭或公司使用的无线路由器就是一个 AP，其他无线终端可通过 AP 相互连接。

◇ STA：即无线站点，是一个无线网络的终端。如笔记本，手机及其他可无线上网的设备。

2.1.1 自组无线网络（Ad-hoc）

自组网（Ad-hoc）仅由两个及以上 STA 组成的无线网络结构，网络中的 STA 都可以直接通信。

注：HC-21（TLN13UA06）模块可以作为 STA 接入到 AP 中，也可作为 AP 供 STA 的接入。但作为 AP 时，仅支持 STA 和 AP 之间的数据通信，不支持 STA 和 STA 之间直接数据通信。如图 2-1 所示。



图 2-1 自组无线网络拓扑图

2.1.2 基础无线网络（Infra）

基础网（Infra）是由 AP 创建，众多 STA 加入所组成的无线网络结构。这种类型的网络特点是 AP 作为整个网络的中心，网络中所有的数据通信都通过 AP 来转发完成。如图 2-2 所示。



图 2-2 基础无线网络拓扑图

2.2 安全机制

本模块支持多种无线网络加密方式，能充分保证用户数据的安全传输，包括：

- ◇ WEP64
- ◇ WPA-PSK/TKIP
- ◇ WPA2-PSK/TKIP
- ◇ WEP128
- ◇ WPA-PSK/CCMP
- ◇ WPA2-PSK/CCMP



2.3 快速联网

本模块作为 STA 时支持通过指定信道号的方式来进行快速联网。在通常的无线联网过程中，会首先对当前的所有信道自动进行一次扫描，来搜索准备连接的目的 AP 创建的（或 Adhoc）网络。本模块提供了设置工作信道的参数，在已知目的网络所在信道的条件下，可以直接指定模块的工作信道，此时可以将无线联网时间从 2 秒降低至约 100 毫秒，从而达到加快联网速度的目的。

2.4 地址绑定

本模块作为 STA 时支持在联网过程中绑定目的网络 BSSID 地址（即 MAC 地址）的功能。根据 802.11 协议规定，不同的无线网络可以具有相同的网络名称（也就 SSID/ESSID），但是必须对应一个唯一的 BSSID 地址。非法入侵者可以通过建立具有相同的 SSID/ESSID 的无线网络的方法，使得网络中的 STA 联接到非法的 AP 上，从而造成网络的泄密。通过 BSSID 地址绑定的方式，可以防止 STA 接入到非法的网络，从而提高无线网络的安全性。

2.5 无线漫游

本模块作为 STA 时支持基于 802.11 协议的无线漫游功能。无线漫游指的是为了扩大一个无线网络的覆盖范围，由多个 AP 共同创建一个具有相同的 SSID/ESSID 的无线网络，每个 AP 用来覆盖不同的区域，接入到该网络的 STA 可以根据所处位置的不同选择一个最近（即信号最强）的 AP 来接入，而且随着 STA 的移动自动的在不同的 AP 之间切换。

在开启无线漫游后，模块将每隔 100 秒自动发起一次扫描过程，并根据扫描结果选择一个当前最近的 AP 连接。在无线漫游期间，网络可能短暂中断，因此，在不需要漫游功能的情况下，建议将其关闭。

需要说明的是，启用 BSSID 地址绑定后，将无法使用无线漫游功能。

2.6 TCP/IP 协议栈

本模块内置一个完整的 TCP/IP 协议栈，支持 TCP/UDP/ICMP/ARP/DHCP/DNS/HTTP 协议，此外还支持基于 AT+ 指令的 SOCKET 编程接口，支持 DHCP Server / Client，支持 DNS Server / Resolver，支持 HTTP，内置 WEB 服务器。

2.7 灵活的参数配置

本模块主要基于预设的参数进行工作，配置参数保存在内部的 Flash 存储器中，可以掉电保存，用户可以使用多种方式对模块的配置参数进行修改，包括：

◇ 基于无线连接，使用配置管理程序（推荐）

使用无线适配器（即无线网卡）与模块通过无线连接，然后运行汇承科技提供的配置管理程序，即可搜索并对模块的参数进行配置，操作方法及界面与使用串口连接时完全相同。这种方式的优点：

①随时，即模块无需连接任何到无线网络，只要模块处于上电状态即可进行参数配置；

②随地，无需任何物理线缆连接，即使模块已经集成到用户设备内部，同样可以进行参数配置；

③批量配置，即可以实现一次性对一批模块进行统一的配置，这在用户需要对出厂的设备进行统一的出厂设置时尤其有用，可以大幅度提高工作效率。



◇ 基于串口连接，使用配置管理程序

需要使用转接板将模块与 PC 机的串口进行连接，然后运行我们提供的配置管理应用程序，这种方式的优点是界面直观，操作简便。

◇ 基于网络连接，使用 IE 浏览器程序

这种方式需要模块在已经连接到无线网络的情况下使用，在一台连接到同一个无线网络中的 PC 机上，使用 IE 浏览器连接本模块内置的 WEB 服务器即可。这种方式的优点是操作简便，界面直观。

◇ 基于串口连接，使用 Windows 下的超级终端程序

需要使用转接板将模块与 PC 机的串口进行连接，然后运行 windows 下的超级终端程序，使用 AT+ 指令对参数进行配置，这种方式最为灵活，但是需要用户对 AT+ 指令集比较熟悉。

参数配置详细说明请查看第 3 章节。

2.8 固件升级

本模块支持固件在线升级功能，可以通过以下两种方式进行固件的升级：

◇ 基于串口连接，使用配置管理程序

◇ 基于网络连接，使用 IE 浏览器程序

3.快速使用向导

3.1 快速组建测试平台

3.1.1 组建原理

模块的核心功能是将串口数据与 TCP/IP 数据进行相互转换传输。故而模块组建测试平台，必须将模块的串口端与计算机串口端连接，模块 WIFI 网络端也需要与计算机通过无线网卡或无线路由器建立连接。

在模块串口端与 WIFI 端同时与电脑设备连接上的前提下，可通过串口调试工具获取从 WIFI 端发送来的数据，同时也可以通过 TCP/UDP 工具获取从串口端发送来的数据。

3.1.2 搭建平台

搭建测试平台需要准备以下工具，其中我公司提供串口转接板和调试软件下载。



◇ 配置管理程序：UART-WIFI 配置管理程序，是由我公司提供的一个快捷配置管理模块的应用程序。用于配置查询/配置、功能测试等，详细功能参见 3.2 节。



◇ 串口调试工具：用于模块串口端数据收发的调试工具，网络上随处可见的小工具，如：串口大师、串口助手、串口调试等。本章节将使用“串口大师”作为演示。



◇ TCP/UDP 工具：用于模块网络端数据收发的调试工具，网络上随处可见的小工具。本章节将使用“TCP&UDP 测试工具”作为演示。



◇ 个人计算机：用于提供串口与模块串口端连接。运行相应调试软件，监测调试串口端与网络端相关数据。一台同时具备无线网卡与串口输出的电脑即可完成测试。



◇ 无线适配器：台式机没有无线网卡时使用，提供无线网络功能，用于连接模块网络端。通常使用市面上常见的 USB 无线网卡即可。



◇ 串口转接板：用于模块 TTL 电平的串口数据与电脑 RS232 电平的串口数据间转换。板子有 DC 座通过 5V 直流供电，模块经 MAX232 芯片转换电平后连接到电脑串口端。



◇ 串口延长线：用于连接电脑主板上串口 DB 公头与串口转接板上 DB 母头。市面上常见。

注①：需要注意的是 UART 串口包含 TTL 电平的串口和 RS232 电平的串口。串口连接时，模块的串口端为 3.3V 的 TTL 电平，电脑为+12V~-12V的 RS232 电平。两者之间不能直接通讯，需要通过 MAX232芯片转换电平后，才可以通讯，串口开发板正是用于此处。用户可自行制作此串口转接板。

3.1.3 收发测试

3.1.3.1 默认模块参数测试

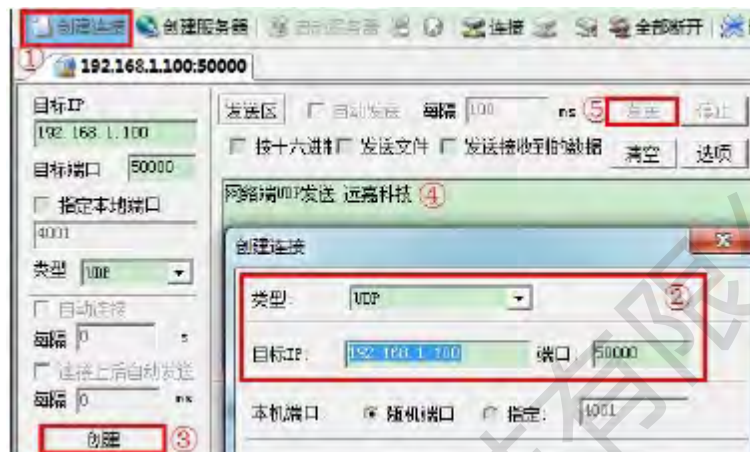


图 3-1 TCP & UDP 测试工具设置示意图

本节详细介绍在不对模块进行任何操作的前提下，即使用模块的出厂默认设置，快速实现对模块的收发测试。将模块直接插到串口转接板上，按下列步骤操作即可。

步骤 A：将搭建平台准备的各项工具按各自功能连接好。

步骤 B：模块默认 AP 模式，将类似路由器功能发射一个名为“wifi-socket”的 WIFI 信号，使用无线网卡搜索到这个 WIFI 信号，连接。

步骤 C：a、打开“串口大师”软件。选择电脑使用的 COM 口，波特率为“9600”、数据位“8”、校验位“NO”、停止位“1”都是默认设置。

b、“高级选项”按钮中“流量控制”下拉框，选择“硬件 RTS/CTS 流控”，预防数据丢失。

c、点击“打开串口”。即可进入下一步骤。

步骤 D：打开“TCP&UDP 测试工具”软件。按上图中步骤操作，

a、点击图 3-1 中①所示区域“创建连接”，弹出创建连接对话框。

b、在图 3-1 中②所示区域填写参数。选择类型“UDP”、填写目标 IP“192.168.1.100”、端口“50000”。此处参数为模块的出厂默认设置，可暂不理睬。点击“创建”按钮。

c、点击图 3-1 中③所示的“创建”按钮，创建此连接。

d、在“发送区”填写你要发送的数据。

e、点击图 3-1 中⑤所示的“发送”按钮，数据将通过网络段发送给模块串口端。

此时在“串口大师”的接收区，可看到“TCP&UDP 测试工具”发来的数据。

3.1.3.2 配置模块参数测试

本节详细介绍在不具备无线网卡时，测试环境有已知 WIFI 信号的情况下，快速实现对模块的收发测试。将模块直接插到串口转接板上，按下列步骤操作即可。

步骤 A：将搭建平台准备的各项工具按各自功能连接好。

步骤 B：a、打开“UART-WIFI 配置管理程序”，在“串口连接”中选择电脑使用的 COM 口，“设置”中确认波特率为“9600”、数据位“8”、校验位“无效验”、停止位“1”。

b、点击“退出透传模式”按钮。退出成功后，点击“搜索模块”。

步骤 C: 如图 3-2 所示, 设置“工作模式”为基础网-STA, 填写所在环境的WIFI帐号和密码, 如公司的WIFI帐号与密码。注意安全模式必须和发射WIFI信号的路由器一致。

步骤 D: 如图3-3所示, 设置“协议类型”为TCP, “C/S模式”为客户端。服务器地址填写打开“TCP&UDP测试工具”的电脑 IP, 笔者的电脑 IP 为“192.168.1.11”。端口号任意设置, 与后续步骤中“TCP&UDP 测试工具”设置的端口号一致即可。此处端口号设置为“60000”。

步骤 E: 确认无误后, 点击“提交修改”。弹出确认重启。弹出无响应。都点击确认即可。

步骤 F: 如果无线设置中安全模式未出错, 模块已经连接至无线路由器。可通过“UART-WIFI 配置管理程序”中“功能测试”选项下“扫描”功能确认是否正常连接。如图 3-4 所示。

步骤 G: 打开“串口大师”。同 3.1.3.1 中步骤 C。

步骤 H: 打开“TCP&UDP 测试工具”, 点击“创建服务器”, 输入本机端口“60000”即步骤 D 中设置给模块的端口值。点击“启动服务器”。将出现图 3-5 所示界面。

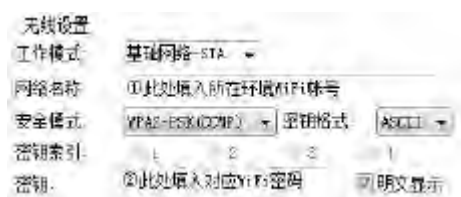


图 3-2 设置模块接入 WIFI 参数



图 3-3 设置模块服务器参数



图 3-4 确认模块联网状态

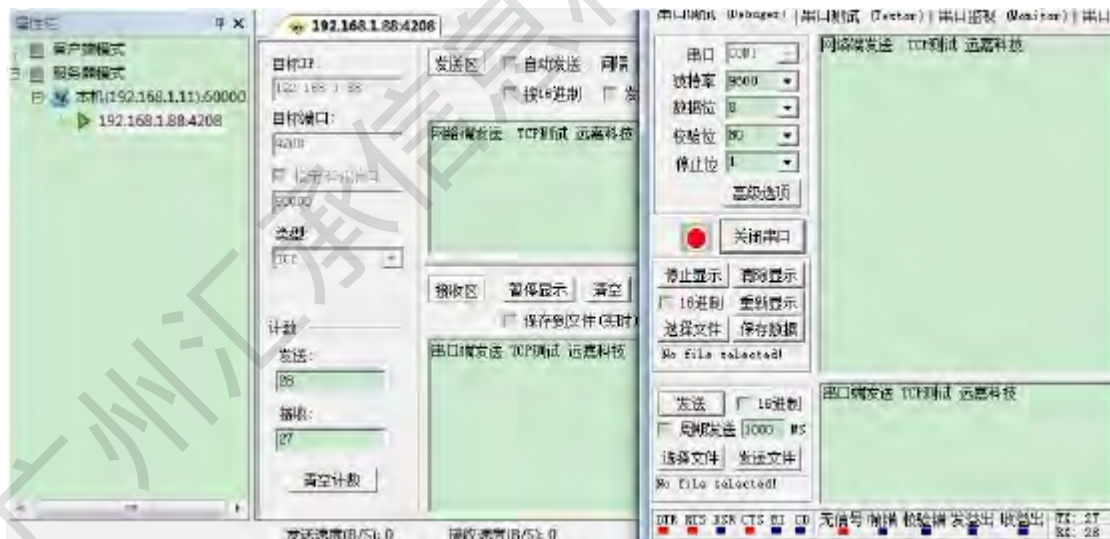


图 3-5 TCP&UDP 测试工具成功创建连接后收发示意图

此时在“TCP&UDP 测试工具”和“串口大师”的发送区与接受区可以收发数据。即实现了双向通讯传输的功能, 同时也可看出, 模块两端是透明传输, 数据内容并未改变。

3.1.3.3 手机平台测试

用户可通过手机平台对模块进行数据收发测试。使用手机端“TCP&UDP 测试工具”建立服务器时, 记录创建的服务器地址与端口。如“127.0.0.1: 50000”。将服务器地址与端口号填入模块配置中, 提交修改, 重启模块后, 即可在电脑端串口大师, 接收到手机端“TCP&UDP 测试工具”发来的数据。原理即步骤与 3.1.3.2 节大同小异, 在次不再赘述。



3.2 参数配置

3.2.1 通过网页浏览器配置

模块内置 WEB 服务器。支持使用网页浏览器，通过访问 IP 地址的形式配置模块参数，方法如下：

- ① 保持访问设备与模块处于 WIFI 网络连接状态。
- ② 在网页浏览器地址栏输入模块默认的 WEB 服务器地址 192.168.1.100。部分浏览器需输入模块 WEB 服务器默认端口号，否则页面打不开，如 192.168.1.100:80。AP 模式下，可通过模块当前的域名访问网页，例如模块默认域名 Local.hed.com.cn。
- ③ 初次登录 WEB 服务器，需输入帐号密码。模块默认帐号“admin”密码“000000”。
- ④ 进入配置界面后，对模块各项参数进行修改后，需点击“Save”。完成全部配置后，需点击“Restart System”按钮。类似路由器的配置。各项配置详细说明参考 3.2.2 节。

3.2.2 通过配置管理程序配置

UART-WIFI 配置管理程序是我公司为用户迅速掌握模块使用方法，快捷配置管理模块参数而开发的专用管理软件。通过本软件，可快速实现查询/修改模块参数、参数导入/导出、批量修改模块参数、恢复出厂设置、无线网络命令测试、Socket 命令测试、AT+指令测试、固件升级、系统版本信息查询等功能。

3.2.2.1 通过串口连接

- ① 将模块插到串口转接板，连接串口延长线至电脑串口端，连接 5V 直流电源给转接板供电。
- ② 打开“UART-WIFI 配置管理程序”，在“串口连接”中选择电脑使用的 COM 口，“设置”中确认波特率为“9600”、数据位“8”、校验位“无效验”、停止位“1”。
- ③ 点击“搜索模块”。如提示“串口被占用”，点击“退出透传模式”按钮或检查串口正常后再尝试。搜索模块成功后，软件将会显示模块当前配置参数信息。

3.2.2.2 通过 WIFI 连接

通过 WIFI 可连接多个模块，从而实现快速对模块批量操作。但使用本功能需要无线网卡并正确安装模块无线驱动程序，且同时对多个模块上电使其工作。

- ① 无线网卡工作正常后，安装模块无线驱动程序，在电脑网络配置器中可看到“UART-WIFI Device”
- ② 启动“UART-WIFI 配置管理程序”，如果驱动安装成功，“无线连接”区域将处于激活状态。
- ③ 点击“设置”按钮，输入系统密码（只有系统登录密码匹配的模块才会被扫描到）。
- ④ 选择扫描信道，如果选择字段，将扫描所有信道，搜索模块时可能花费较长的搜索时间。
- ⑤ 点击“搜索模块”按钮。将显示所有被扫描到的模块。

3.2.2.3 UART-WIFI 配置管理程序使用说明

本节详细说明配置管理程序中各区域的功能作用。程序界面示意图如 3-6 所示：



图3-6 配置管理软件示意图

配置参数

- ◇ 刷新参数：从模块重新读取配置参数，并刷新显示到程序界面。
- ◇ 提交修改：修改参数后，点击此按钮将新参数保存至模块中，同时复位模块，使新参数生效。
- ◇ 恢复出厂：点击此按钮，模块配置恢复出厂状态。同时复位模块，使参数生效。
- ◇ 导出参数：可将当前显示中的配置参数保证到指定路径的自定义配置文件。方便保存配置参数。
- ◇ 导入参数：将导出的配置文件导入至程序。需“提交修改”后，参数才修改至模块中。
- ◇ 高级设置：本项包含模块详细参数修改，如无需求，默认即可。详细说明参见表 5-5。

功能测试

功能测试选项提供了快速测试 AT+指令的功能，如果用户使用自动工作模式可以跳过本节内容。另外，本软件通过无线连接到模块时，将无法使用本页功能。

- ◇ 加网/断网：加入或断开无线网络。
- ◇ 扫描：扫描模块当前环境下的无线网络。显示网络类型、网络名称、MAC 地址、信道、信号强度等。状态：监测模块当前网络连接状态。显示连接与否、IP 地址、网关、DNS 服务器等信息。
- ◇ 复位：对模块复位操作，用于恢复到上电初始时的网络参数配置。
- ◇ 建立：创建 Socket，配置相应参数后，返回一个 Socket 号，则创建成功。Socket 号默认监听状态。
- ◇ 关闭：输入需要关闭的 Socket 号，删除之前建立的 Socker 连接。
- ◇ 查询：输入创建时返回的 Socket 号，将返回 Socket 当前的状态。
- ◇ 发送：通过 socket 发送数据。在弹出的对话框中输入需要的 socket 号（类型为 TCP 服务器的 socket 无法直接进行数据收发，需使用其客户端连接 socket 号）及需要发送的数据。
- ◇ 接受：通过 socket 接收数据。在弹出的对话框中输入需要的 socket 号（类型为 TCP 服务器的 socket 无法直接进行数据收发，需使用其客户端连接 socket 号）及期望接收的数据个数。在接收数据之前，请先使用“查询”命令确认该 socket 接收缓冲区中是否有数据。

固件升级

当本软件通过无线 WIFI 连接到模块时，将无法使用本页功能。用户通过本页可以更新模块固件程序。

警告！升级前请务必仔细阅读升级注意事项。升级失败将导致模块损坏无法再次使用。

系统信息

用户可通过本页可查询模块的 MAC 地址、硬件版本、固件版本等信息。

3.2.3 通过 AT 指令配置

本节简要阐述通过 AT 指令对模块进行配置。仅使用自动工作模式的用户可跳过本节。对于希望测试 AT+ 指令功能的用户，可使用 Windows 操作系统自带的超级终端程序直接输入 AT+ 指令，也可以通过各种串口工具发送 AT+ 指令操作模块，进而达到配置模块的目的。此处仅介绍可由串端口直接配置，不做详细说明，AT 指令使用方法请参见第 5 章节。

3.3 编程使用原理

3.3.1 串端口编程

串端口是串口协议中标准的 TTL 电平通讯。正如本文一再复述的，模块在自动工作模式下，发送数据是模块内部自动处理转换协议后送到 WIFI 网络端的。其中自动处理的过程用户无需理会，仅将模块作为一个普通的串口设备即可。关于各种平台下收发数据的串口协议代码，网络上俯拾皆是。本公司未曾且没必要开发设计任何开发协议包或 DOME 程序。

3.3.2 网络端编程

网络端是标准的 TCP/IP 数据包协议，此协议负责数据的各项处理。而 WIFI 网络仅负责网络的连接形式，因此 TCP/IP 数据包协议与 WIFI 网络本身并无关联。用户可通过 OCX 或者 API 函数实现通讯编程，如 winsock.ocx。网络通信使用 TCP Server, TCP Client 或者 UDP 任意一种工作方式，模块内可设置工作方式，仅需注意软件端与模块对应关联即可。如 Server 对应 Client, UDP 对于 UDP。相应网络知识请自行学习掌握。

4.典型应用实例

4.1 主动型串口设备联网

主动型串口设备联网指的是由设备主动发起连接，并与后台服务器进行数据交互（上传或下载）的方式。典型的主动型设备，如无线 POS 机，在每次刷卡交易完成后即开始连接后台服务器，并上传交易数据。主动型串口设备联网的拓扑结构如图 4-1 所示。其中，后台服务器作为 TCP Server 端，设备通过无线 AP/路由器接入到网络中，并作为 TCP Client 端。



图 4-1 主动型串口设备联网

4.1.1 模块参考设置

无线设置：输入无线 AP/路由器的网络名称和密码

网络设置：启用 DHCP 自动获得网络地址

工作模式设置：启用自动工作模式

协议类型：TCP

C/S 模式：客户端

服务器地址：后台服务器 IP 地址

端口号：后台服务器端口号，如 6000

4.2 被动型串口设备联网

被动型串口设备联网指的是，在系统中所有设备一直处于被动的等待连接状态，仅由后台服务器主动发起与设备的连接，并进行请求或下传数据的方式。典型的应用，如某些无线传感器网络，每个传感器终端始终实时的在采集数据，但是采集到的数据并没有马上上传，而是暂时保存在设备中。而后台服务器则周期性的每隔一段时间主动连接设备，并请求上传或下载数据。此时，后台服务器实际上作为 TCP Client 端，而设备则是作为 TCP Server 端。

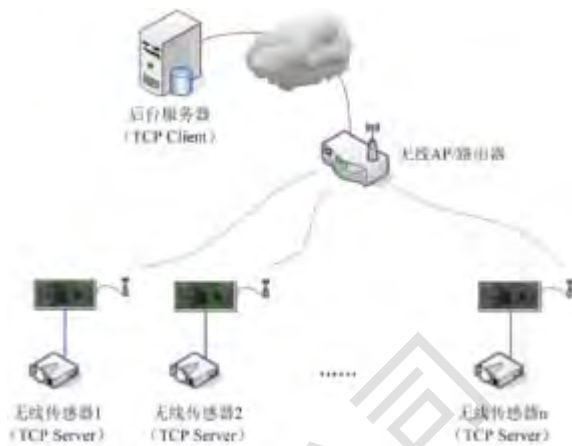


图 4-2 被动型串口设备联网

4.2.1 模块参考设置

无线设置：输入无线 AP/路由器的网络名称和密码。

网络设置： IP 地址：192.168.0.X 子网掩码：255.255.255.0
 网关地址：192.168.0.1 DNS 地址：192.168.0.1

模式设置：启用自动工作模式

协议类型：TCP

端口号：6000

C/S 模式：服务器

连接超时：120

4.3 广播方式的串口设备联网

广播方式的串口设备联网，是一种基于 UDP 协议的组网方式，网络中的主控端工作在 UDP 广播方式下，所有的设备端均工作在 UDP 单播方式下。这种方式本质上也是一种被动型的网络。最典型的应用为对传统的基于 RS422/485 总线连接的网络进行无线网络升级，采用这种基于 UDP 广播的组网方式，原有系统中的所有设备及设备的系统控制端均无需做任何软件改动，直接使用本模块替换原有的 RS422/485 硬件连接即可，真正实现“即插即用”。

4.3.1 模块参考设置

设备控制端

无线设置：输入无线 AP/路由器的网络名称和密码。

网络设置： IP 地址：192.168.0.Y
 子网掩码：255.255.255.0
 网关地址：192.168.0.1
 DNS 地址：192.168.0.1

模式设置：启用自动工作模式

协议类型：UDP

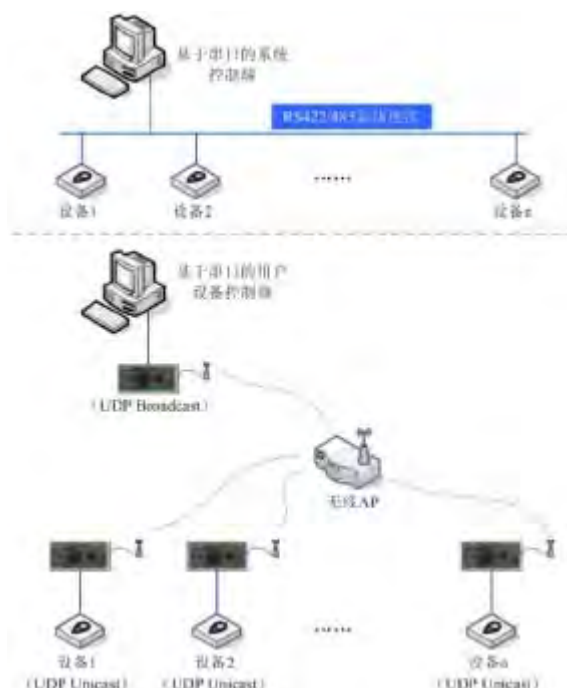
C/S 模式：广播

端口号：6000

设备 X

无线设置：输入无线 AP/路由器的网络名称和密码。

网络设置： IP 地址：192.168.0.X
 子网掩码：255.255.255.0
 网关地址：192.168.0.1
 DNS 地址：192.168.0.1



4-3 广播方式的串口设备联网

模式设置：启用自动工作模式图
协议类型：UDP
C/S 模式：单播
端口号：6000

4.4 无线虚拟串口

4.4.1 模块参考设置

设备 1

使能“网络不存在时自动创建”

无线设置：网络类型：自组网络
网络名称：my_adhoc
加密类型：开放

网络设置：IP 地址：192.168.0.1
子网掩码：255.255.255.0
网关地址：192.168.0.1
DNS 地址：192.168.0.1

模式设置：启用自动工作模式
协议类型：TCP
端口号：6000

C/S模式：服务器
连接超时：0

设备 2

无线设置：网络类型：自组网络
网络设置：IP 地址：192.168.0.2
子网掩码：255.255.255.0
网关地址：192.168.0.1
DNS 地址：192.168.0.1

模式设置：启用自动工作模式
协议类型：TCP
服务器地址：192.168.0.1
端口号：6000

网络名称：my_adhoc
子网掩码：255.255.255.0
DNS 地址：192.168.0.1

C/S 模式：客户端
端口号：6000

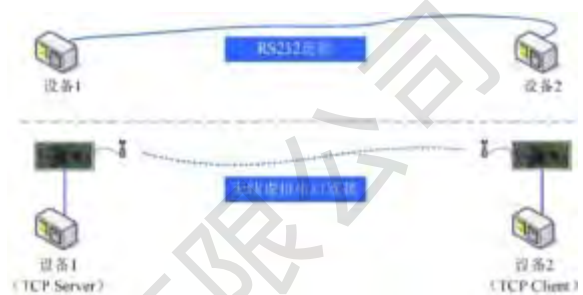


图 4-4 无线虚拟串口

4.5 多网络连接模式

如图 4-5 所示的一个基于本模块的多网络连接功能的网络拓扑结构图，此系统并不针对某种特定的应用，也不具有实用性，而仅作为一个模拟系统来用于功能说明。上述系统的设备控制端模块工作在命令模式，由用户通过输入 AT+ 指令来控制，其同时建立了四个 TCP 连接，包括 2 个 Client 和 2 个 Server。而每个 Server 又有若干 Client（即设备）接入。通过不同的 socket 连接，设备控制端可以与图中所示的所有设备及服务器进行通信。

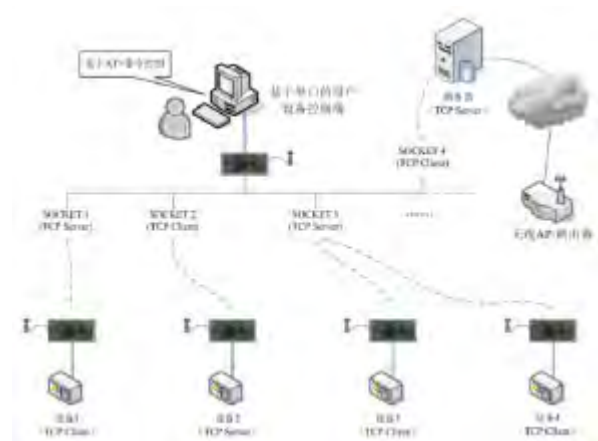


图 4-5 多网络连接模式

4.5.1 模块参考设置

设备控制端

无线设置：输入无线 AP/路由器的网络名称和密码。

网络设置：IP 地址：192.168.0.X 子网掩码：255.255.255.0
网关地址：192.168.0.1 DNS 地址：192.168.0.1

模式设置：不启用自动工作模式

4.6 软 AP 模式

如图 4-6 所示为一个本模块作为 AP 时的网络拓扑结构图，此系统并不针对某种特定的应用，也不具有实用性，而仅作为一个模拟系统来用于功能说明。当本模块工作在 AP 模式时，最多可接入4个智能终端，终端的 IP 地址可通过本模块所提供的DHCP Serve模块进行动态分配，也可静态设置，本模块DHCP Server所分配的 IP 地址租期统一为2小时，智能终端要定时的进行续约，智能终端和本模块建立无线连接后即可进行用户数据的通信，本模块工作在 AP模式下时，用户数据通信方式同 STA 模式下，详见上文。此外，本模块工作在AP 模式时，将自动使能 DNS Server 对本模块的域名进行解析，网卡域名可通过参数配置，例如，若模块域名为“local.hed.com.cn”时，智能终端通过浏览器键入：<http://local.hed.com.cn> 访问本模块配置网页。



图 4-6 AP 模式

4.6.1 模块参考设置

无线设置：网络类型：软 AP

网络名称: soft ap

加密类型：开放

SSID 广播：使能

网络设置: IP 地址: 192.168.0.1

子网掩码: 255.255.255.0

网关地址: 192.168.0.1

DNS 地址: 192.168.0.1

DHCP Server: 开启, IP 范围为本网卡 IP 地址加 1 至本网卡地址加 8, 即, 192.168.0.2~192.168.0.9, 租约为 2 小时

模块域名: local.hed.com.cn

模式设置：启用自动工作模式

5. AT指令协议集

如果用户仅使用无线模块的自动工作模式可以跳过本章内容。

5.1 语法说明

本模块使用 AT+指令协议作为用户控制协议。AT+指令协议采用一套基于 ACSII 的命令行格式指令集，下面对其处理流程和语法格式进行描述。

5.1.1 处理流程

AT+指令协议采用命令+响应的形式,绝大多数指令需要接收方在处理完成后返回响应消息,如果在前一个命令处理过程中,再次接收到新的命令,则将其静静丢弃,不返回任何消息,如图5-1所示。

对于某些特殊命令，如 AT+SKSND、AT+SKRCV，需要在指令或响应详细后面传输二进制数据，此时，命令传输的接收方暂时进入透明传输状态，开始接收二进制数据流，直到接收到在命令或消息的<size>字段中规定长度的数据或等待超时后，自动退出透明传输状态，其流程如图 5-2 所示。

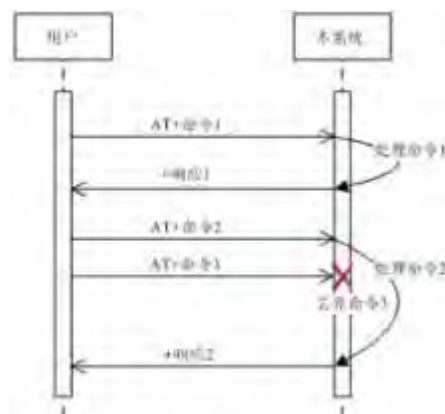


图 5-1 AT+指令处理流程

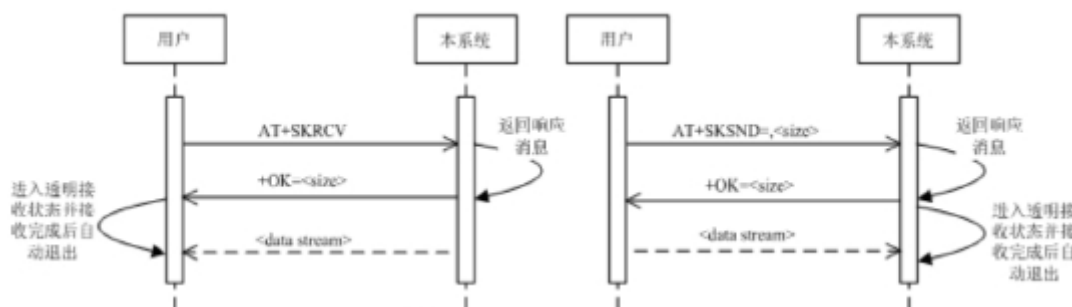


图 5-2 AT+指令特殊处理流程

5.1.2 语法格式

◇ 格式说明

<>: 表示必须包含的部分

[]: 表示可选的部分

◇ 命令消息

AT+<CMD>[op][para1],[para2],[para3],[para4]...<CR>

AT+: 命令消息前缀

CMD: 指令字符串

[op]: 指令操作符, 当命令需要带参数时, 可以指定参数的操作类型, 包括,

=, 参数/返回值前导符

!=, 在设置参数类命令中, 表示将修改同步至 flash

=?, 在设置参数类命令中, 查询当前设置

[para-n]: 查询时返回参数或出错时返回错误码

<CR>: 结束符, 回车, ACSII 字符 0x0d

◇ 响应消息

+<RSP>[op][para1],[para2],[para3],[para4]...<CR><LF><CR><LF>

+: 响应消息前缀

RSP: 响应字符串

OK 成功

ERR 失败

<CR>: 回车, ACSII 字符 0x0d

<LF>: 换行, ACSII 字符 0x0a

◇ 数据类型

String: 字符串, 以双引号包围, 内容不含引号, 如: "this is a string"

Dec: 十进制数字, 如: 10

Hex: 16 进制数字, 如: a

IP: ip 地址串, 如: 192.168.0.1

MAC: 由 12 个 16 进制数字组成, 如 001EE3A80102



5.1.3 格式范例

发送: AT+	[发送一个空指令]
返回: +OK	[返回接收成功]
发送: AT+WJOIN	[发送“加入网络”指令]
返回: +ERR=-10	[返回“加入网络失败”错误码]
发送: AT+UART=9600,1,1,0	[对串口配置参数]
返回: +OK	[返回配置成功]
发送: AT+ATPT=!500	[赋值自动组帧周期配置]
返回: +OK	[返回配置成功]
发送: AT+ATPT=?	[查询自动组帧周期配置]
返回: +OK=500	[返回查询值]

5.1.4 错误代码

表 5-1 错误码含义列表

	含义说明
-1	无效的指令格式
-2	指令不支持
-3	无效的操作符
-4	无效的参数
-5	操作不允许
-6	内存不足
-7	FLASH 错误
-10	加入网络失败
-11	无可用的 socket
-12	无效的 socket
-13	Socket 连接失败
-100	未定义错误

5.2 指令集

5.2.1 指令列表

表 5-2 AT+指令列表

	功能描述
(null)	空指令
ATLT	设置/查询数据自动组帧数据长度
ATM	设置/查询模块工作模式
ATPT	设置/查询数据自动组帧周期
ATRM	设置/查询自动工作模式下模块自动创建的 socket 连接信息
BSSID	设置/查询指定 AP 的 bssid 地址
BRDSSID	使能和关闭 AP SSID 广播
CHL	设置/查询指定无线信道方式
CHLL	设置/查询无线信道列表
CMDM	设置/查询系统的默认命令模式
DNS	设置网卡域名



E	切换串口字符回显
ENCRY	设置/查询无线网络安全模式
ENTM	进入串口透明传输模式
ENTS	进入睡眠模式
ESPC	设置/查询用于退出串口透明传输模式的逃逸字符
ESPT	设置/查询用于退出串口透明传输模式的逃逸时间
IOC GPIO	控制
IOM	设置/查询 GPIO 模式
KEY	设置/查询无线网络密钥
LKSTT	查询网络连接状态
NIP	设置/查询本端 IP 地址
PASS	设置/查询系统密码
PMTF	将内存中所有参数更新到 Flash
QMAC	查询物理地址
QVER	查询版本信息
RSTF	恢复出厂设置
SKCLS	关闭 Socket 连接
SKCT	创建 Socket 连接
SKRCV	通过 Socket 连接接收数据
SKSDF	设置默认发送 Socket 连接
SKSND	通过 Socket 连接发送数据
SKSTT	查询 Socket 连接状态
SSID	设置/查询无线网络名称
UART	设置/查询串口数据格式
WARC	设置/查询自动工作模式下的自动重试次数
WARM	设置/查询无线漫游
WATC	设置/查询自动创建 Adhoc 网络
WBGR	设置/查询无线网络 BG 模式和最大发射速率
WEBS	设置/查询 Web 服务器
WAPST	启动/退出 AP
WJOIN	加入/创建无线网络
WLEAV	断开无线网络
WPRT	设置/查询无线网络类型
WPSM	打开/关闭节能模式
WSCAN	扫描网络
Z	复位

5.2.2 网络控制类

AT+WJOIN

功能：该指令仅在模块作为 STA 时有效，用于加入/创建无线网络。如果当前网络类型为 adhoc，且未检测到指定 SSID 的网络，则自动创建该网络。如果当前网络已经处于联网状态，则直接返回网络连接信息。



格式: AT+WJOIN<CR>

+OK=<bssid>,<type>,<channel>,<b_encry>,<ssid>,<rssi><CR><LF><CR><LF>

参数: bssid: 网络 BSSID, 长度为 12 的十六进制数, 格式为 001EE3A34455

type: 网络类型。其中 0 表示 infra 网络、1 表示 adhoc 网络。

channel: 信道号。

b_encry: 加密模式。其中 0 表示开放模式、1 表示加密模式。

SSID: 无线网络名称, 1~32 个 ASCII 字符, 双引号包围。

rssi: 网络信号强度, 不含负号, 单位 dB, 即 50 表示信号强度为-50dB。

AT+WLEAV

功能: 该指令仅在模块作为 STA 时有效, 用于断开当前无线网络。

格式: AT+WLEAV<CR>

+OK<CR><LF><CR><LF>

参数: 无

AT+WAPST

功能: 该指令仅在模块作为 AP 时有效, 用于启动和关闭软 AP。

格式: AT+WAPST=[!?][start]<CR>

+OK[=start]<CR><LF><CR><LF>

参数: start: 启动/关闭软 AP。其中 0 表示关闭软 AP、1 表示启动软 AP。

AT+WSCAN

功能: 该指令仅在模块作为 STA 时有效, 用于扫描无线网络, 完成后返回。

格式: AT+WSCAN<CR>

+OK=<bssid>,<type>,<channel>,<b_encry>,<ssid>,<rssi><CR><LF>

<bssid>,<type>,<channel>,<b_encry>,<ssid>,<rssi><CR><LF>

<CR><LF>

参数: 同 AT+WJOIN。

AT+LKSTT

功能: 查询本端网络连接状态。

格式: AT+LKSTT<CR>

+OK[=status,ip,netmask,gateway,dns]<CR><LF><CR><LF>

参数: status: 连接状态 其中 0 表示断开、1 表示连接。

ip: ip 地址, 数据格式为"192.168.1.22", 不含引号。

netmask: 子网掩码, 数据格式同 ip 地址。

gateway: 网关地址, 数据格式同 ip 地址。

dns: DNS 地址, 数据格式同 ip 地址。

AT+SKCT

功能: 建立 socket。在 client 模式, 等待连接完成 (成功或失败) 后返回;

在 server 模式下, 创建完成后直接返回。

格式: AT+SKCT=[protocol],[cs],[host_timeout],[port]<CR>

+OK=<socket><CR><LF><CR><LF>



参数: protocol: 协议类型, 其中 0 表示 TCP 协议、1 表示 UDP 协议。
cs: C/S 模式, 其中 0 表示 Client、1 表示 Server。
port: 端口号, 用于指定自动工作模式下, 当模块创建的默认连接的端口号。
socket: socket 号。
host_timeout: 根据 protocol 及 cs 两者状态决定, 其含义如表 5-3 所示。

表 5-3 host_timeout 不同状态下定义说明

	protocol	含义
0	任意值	目的服务器名称, 可以输入域名或 ip 地址, 如"192.168.1.100"或"www.wavesen.com"
1	0	TCP 连接超时时间, 即连接到本服务器的客户端超过本时间不发送任何数据后即被自动踢掉, 有效取值范围 1~10000000, 单位: 秒, 0 表示永远不, 缺省 120 秒
1	1	无意义

AT+SKSND

功能: 通过指定的 socket 发送数据, 完成后返回。此命令使用二进制格式发送数据, 用户应在接收到模块的响应消息(+OK)之后再开始发送原始数据。模块接收完指定长度的数据后自动结束数据传输阶段, 并将数据发送到网络上, 多余的数据将被丢弃。否则, 模块在等待超时(1s)后, 强制结束数据传输阶段并将已经接收到的数据发送到网络上。

格式: AT+SKSND=<socket>,<size><CR>
+OK=<actualsize><CR><LF><CR><LF>
[data steam]

参数: socket: socket 号
size: 准备发送的数据长度, 字节数
actualsize: 允许发送的数据长度, 字节数
data steam: 原始数据

AT+SKRCV

功能: 读取指定 socket 的接收缓冲区中的数据, 完成后返回。接收到此命令后, 模块将在发送完成相应消息(+OK)后使用二进制格式传送指定长度数据。

格式: AT+SKRCV=<socket>,<maxsize><CR>
+OK=<size><CR><LF><CR><LF>
[data stream]

参数: socket: socket 号
maxsize: 可接收的最大数据长度
size: 实际接收到的数据长度
data steam: 原始数据

AT+SKSTT

功能: 获取指定的 socket 状态, 返回值的第一行表示的是用户指定的 socket 的状态信息, 如果该 socket 类型为 TCP 服务器, 那么从下一行开始每一行表示一个接入的 client 的 socket 状态。

格式: AT+SKSTT=<socket><CR>
+OK=<socket>,<status>,[host],[port],[rx_data]<CR><LF>
[socket],[status],[host],[port],[rx_data]<CR><LF>
...
<CR><LF>



参数: socket: socket 号

status: socket 状态, 其中 0 表示断开状态、1 表示监听状态、2 表示连接状态

host: 对端 ip 地址

port: 对端端口号

rx_data: 接收 buffer 中数据长度

AT+SKCLS

功能: 关闭指定的 socket。

格式: AT+SKCLS=<socket><CR>

+OK<CR><LF><CR><LF>

参数: socket: socket 号

AT+SKSDF

功能: 设置系统默认发送的 socket。当用户需要在命令模式下进入透明传输模式时, 使用本命令可以指定将串口的透明数据发送的目的地。

格式: AT+SKSDF=<socket><CR>

+OK<CR><LF><CR><LF>

参数: socket: socket 号

5.2.3 系统控制类

AT+

功能: 空指令。

格式: AT+<CR>

+OK<CR><LF><CR><LF>

参数: 无

AT+Z

功能: 复位系统。

格式: AT+Z<CR>

+OK<CR><LF><CR><LF>

参数: 无

AT+E

功能: 切换串口指令回显。

格式: AT+E<CR>

+OK<CR><LF><CR><LF>

参数: 无

AT+ENTS

功能: 系统进入睡眠状态。系统在睡眠状态下接收到任意一个 at+ 指令后自动被唤醒。

格式: AT+ENTS<CR>

+OK<CR><LF><CR><LF>

参数: 无

注: 模块暂不支持该指令, 如需进入省电状态, 请断开模块电源!



AT+ENTM

功能：串口进入透明传输模式。系统在透明传输模式下接收到符合触发条件的逃逸字符时退出此模式。

格式：AT+ENTM<CR>

+OK<CR><LF><CR><LF>

参数：无

AT+RSTF

功能：恢复 FLASH 中的出厂设置。恢复后的设置需系统重启后才能生效。

格式：AT+RSTF<CR>

+OK<CR><LF><CR><LF>

参数：无

AT+PMTF

功能：将保存在内存中的参数全部更新至 FLASH。

格式：AT+PMTF<CR>

+OK<CR><LF><CR><LF>

参数：无

AT+IOC

功能：GPIO 输入/输出控制。当 GPIO1 设置为输入模式（AT+IOM 的 mode=1）时，允许读取 IO 状态，当GPIO1 设置为输出模式（AT+IOM 的 mode=2）时，允许设置 IO 状态。

格式：AT+IOC=[?][status]<CR>

+OK[=status]<CR><LF><CR><LF>

参数：status: IO 状态，其中 0 表示高电平、1 表示低电平

AT+QMAC

功能：获取模块的物理地址。

格式：AT+QMAC<CR>

+OK=<mac address><CR><LF><CR><LF>

参数：mac address: 长度为 12 的十六进制数，格式为 001EE3A34455

AT+QVER

功能：获取系统版本信息，包括硬件版本和固件版本。

格式：AT+QVER<CR>

+OK=<hard,firm><CR><LF><CR><LF>

参数：hard: 硬件版本信息，字符串格式，如"H1.00.00.1029"

firm: 固件版本信息，字符串格式，如"F0.02.02@18: 25: 25 Jul 28 2010"

5.2.4 参数设置类

AT+NIP

功能：当模块作为 STA 时，该指令用于设置/查询本端 ip 地址。需要说明的是，当地址类型设置为 DHCP时，使用本命令无法查询模块实际动态分配到的 ip 地址信息。查询可以使用 AT+LKSTT 命令。当模块作为 AP 时，该指令用于设置/查询本端的 ip 地址以及使能和禁用 DHCP Server，即 AP 模式下指令“type”域的含义通 STA 模式时不同，在使能 DHCP Server 时，任然可以通过该指令查询模块的 ip 地址设置。



格式: AT+NIP=[!?][type],[ip],[netmask],[gateway],[dns]<CR>

+OK[=type,ip,netmask,gateway,dns]<CR><LF><CR><LF>

参数: type: 地址类型, 其中 0 表示使用 DHCP 动态分配/使能 DHCP Server

1 表示使用静态 IP 地址/禁用 DHCP Server

ip: ip 地址, 数据格式为"192.168.1.22", 不含引号。仅在模块作为 STA 且禁用 DHCP 时有效。

netmask: 子网掩码, 数据格式同 ip 地址。仅在禁用 DHCP 时有效。

gateway: 网关地址, 数据格式同 ip 地址号。仅在禁用 DHCP 时有效。

dns: DNS 地址, 数据格式同 ip 地址, 仅在禁用 DHCP 时有效。

AT+DNS

功能: 设置/查询网卡模块域名, 该设置仅在模块作为 AP 时有效。

格式: AT+DNS=[!?][dns name]<CR>

+OK[=dns name]<CR><LF><CR><LF>

参数: dns name: 网卡模块域名, 1~31 个字符, 双引号包围

AT+ATM

功能: 设置/查询模块工作模式。

格式: AT+ATM=[!?][mode]<CR>

+OK[=mode]<CR><LF><CR><LF>

参数: mode: 工作模式, 其中 0 表示自动工作模式、1 表示命令工作模式

AT+ATRM

功能: 设置/查询自动工作模式下模块自动创建的 socket 连接信息。

格式: AT+ATRM=[!?][protocol],[cs],[host_timeout],[port]<CR>

+OK[=protocol,cs,host,port]<CR><LF><CR><LF>

参数: 同 AT-SKCT 下定义

AT+SSID

功能: 设置/查询无线网络名称, 即 ssid。

格式: AT+SSID=[!?][ssid]<CR>

+OK[=ssid]<CR><LF><CR><LF>

参数: ssid: 无线网络名称, 1~32 个字符, 双引号包围

AT+ENCRY

功能: 设置/查询无线网络安全模式。需要特别说明的是除了 OPEN 模式以外, 其它安全模式都需要配合 AT+KEY 指令设置正确的网络密钥, 此外 AP 或者 AD-HOC 模式时, 模块仅支持设置 OPEN、WEP64、WEP128。

格式: AT+ENCRY=[!?][encry mode]<CR>

+OK[=encry mode]<CR><LF><CR><LF>

参数: encry mode: 安全模式。其中 0 表示 OPEN

4 表示 WPA-PSK(CCMP/AES)

1 表示 WEP64

5 表示 WPA2-PSK(TKIP)

2 表示 WEP128

6 表示 WPA2-PSK(CCMP/AES)

3 表示 WPA-PSK(TKIP)

**AT+KEY**

功能：设置/查询网络密钥。需要说明的是，在使用本命令设置网络密钥之前必须首先使用 AT+ENCRY 命令设置网络安全模式。

格式：AT+KEY=[!?][format],[index],[key]<CR>
+OK[=format,index,key]<CR><LF><CR><LF>

参数：format: 密钥格式，其中 0 表示 HEX、1 表示 ACSII。

对于 WEP 加密，HEX 与 ASCII 格式的区别仅在于密钥的显示格式不同，如 WEP64 加密，其输入密钥可以为5个ASCII 字符（如12345）或10个16进制数字（31323333435）。而对于 WPA/WPA2-PSK 方式，HEX 与 ASCII 方式则表示不同的含义。

HEX 格式：表示 PMK。ASCII 格式：表示 PSK。

index: 密钥索引号，1~4 用于 WEP 加密密钥，其它加密方式固定为 0。

key: 密钥字符串，以双引号包围，不同的安全模式，密钥使用长度及格式定义如表 5-4 所示：

表 5-4 密钥

	密钥格式	
	HEX	ACSII
WEP64	10 个 16 进制字符 ^①	5 个 ACSII 字符 ^②
WEP128	26 个 16 进制字符	13 个 ACSII 字符
WPA-PSK(TKIP)	64 个 16 进制字符	8~63 个 ACSII 字符
WPA-PSK(CCMP/AES)	64 个 16 进制字符	8~63 个 ACSII 字符
WPA2-PSK(TKIP)	64 个 16 进制字符	8~63 个 ACSII 字符
WPA2-PSK(CCMP/AES)	64 个 16 进制字符	8~63 个 ACSII 字符

①注：16 进制字符指 0~9、a~f（不区分大小写），如"11223344dd"

注②：ACSII 字符指国际标准化组织（ISO）规定的标准 ACSII 字符集中的数字 0~9 与字母 a~z（区分大小写），如"14u6E"。

小技巧：

在 WPA/WPA2-PSK（TKIP/CCMP）安全模式下，密钥格式为 ASCII 格式时，模块在上电复位时需要首先对 ASCII 格式的密钥进行转换，这个过程需要增加大约 10 秒左右的模块启动时间，而在设置WPA/WPA2 的密钥时直接使用 16 进制格式的 PMK 可以消除这个延迟。但是，某些 AP 在设置中并不提供直接输入 WPA/WPA2 的 HEX 格式的 PMK 方式，因此，在这种情况下，可以使用本产品附WPA Key带的WPA密钥转换工具（Converter.exe）将 PSK 预先转换成PMK，并设置到模块中。

注：路由器密码加密方式暂不能用 HEX 方式。

AT+BSSID

功能：设置/查询指定 AP 的 bssid 地址。本设置仅在模块作为 STA 时 infra 网络下有效。

格式：AT+BSSID=[!?][mode],[bssid]<CR>
+OK[=mode,bssid]<CR><LF><CR><LF>

参数：mode: BSSID 模式。其中 0 表示自动模式、1 表示指定 BSSID

bssid: 网络 BSSID，长度为 12 的十六进制数，格式为 001EE3A34455

系统缺省设置为自动，通常情况下用户无需修改此参数，当用户加入 WiFi 网络时，模块会根据网络名称自动探测到该网络的 BSSID 地址，并连接。当同时存在多个 AP 创建的同名网络时，可以通过修改本参数指定 BSSID 地址，来绑定连接的目的 AP，从而提高网络的安全性。一旦用户指定 BSSID 地址后，将无法连接到 BSSID 不匹配的网络中。该指令仅在模块作为 STA 时有效。



AT+CHL

功能：设置/查询指定无线信道方式。

格式：AT+CHL=[!][mode],[channel]<CR>

+OK[=mode,channel]<CR><LF><CR><LF>

参数：mode：信道模式。其中 0 表示自动模式、1 表示指定信道

channel：无线信道号，有效范围 1~14

系统缺省设置为自动，通常情况下用户无需修改此参数，当用户加入 WiFi 网络时，模块会自动对所有信道进行扫描来探测目的网络。模块扫描每个信道的时间为 150ms，完成所有信道扫描需要大约 2s。用户可以通过设置本参数来指定信道号，来缩短信道扫描时间，从而加快联网速度。需要说明的是，一旦用户设置为指定信道后，模块将仅扫描指定信道，如果目的 AP 改变信道号，则模块将无法成功连接到网络。

AT+CHLL

功能：设置/查询无线信道列表。无线信道列表参数用于规定模块的工作信道范围，不包含在列表中的信道将不会被扫描，合理的使用本参数，可以加快模块的扫描以及联网速度。

格式：AT+CHLL=[!][channel list]<CR>

+OK[=channel list]<CR><LF><CR><LF>

参数：channel list：无线信道列表，采用 16 进制格式，从最低位开始，每一位表示一个信道，缺省为 3fff，表示 1~14 所有信道另外需要说明的是，信道参数优先级高于本参数，也就是说，当信道参数设置为指定信道时，该信道即使不在本参数所规定的信道列表中依然可以联网成功。

AT+WPRT

功能：设置/查询无线网络类型。

格式：AT+WPRT=[!][type]<CR>

+OK[=type]<CR><LF><CR><LF>

参数：type：网络类型，其中 0 表示 infra 网络（模块作为 STA 时）

1 表示 adhoc 网络

2 表示 infra 网络（模块作为 AP 时）

系统缺省设置为模块作为 STA 时 Infra 网络类型，在大多数情况下，我们要连接的都是 AP 创建的无线网络，使用缺省设置即可。仅当用户需要加入 Adhoc 类型的网络或者需要本模块作为 AP 时，可以修改此参数为 Adhoc 或者 AP。

AT+WATC

功能：设置/查询是否使能自动创建 adhoc 网络功能。本设置仅在无线网络类型设置为 adhoc 时有效，表示当加入网络失败时是否自动创建同名的 adhoc 网络。

格式：AT+WATC=[!][enable]<CR>

+OK[=enable]<CR><LF><CR><LF>

参数：enable：使能标志，其中 0 表示不使能、1 表示使能。

AT+WPSM

功能：打开/关闭能自动节能模式。

格式：AT+WPSM=[!][enable]<CR>

+OK[=enable]<CR><LF><CR><LF>

参数：enable：使能标志，其中 0 表示不使能、1 表示使能。



AT+WARM

功能：设置/查询是否使能无线网络漫游功能，本设置仅在模块作为 STA 时 infra 网络下有效。

格式：AT+WARM=[!?][enable]<CR>

+OK[=enable]<CR><LF><CR><LF>

参数：enable：使能标志，其中 0 表示不使能、1 表示使能。

AT+WARC

功能：设置/查询无线网络断开或加网失败后的自动重试功能。本参数仅在模块作为 STA 时 infra 网络下且模块工作在自动工作模式时有效。

格式：AT+WARC=[!?][count]<CR>

+OK[=count]<CR><LF><CR><LF>

参数：count：重试次数，其中 0 表示不重试、1~254 表示重试次数、255 表示永远重试。

AT+WBGR

功能：设置/查询无线网络 BG 模式及最高发射速率。

格式：AT+WBGR=[!?][bg mode],[max rate]<CR>

+OK[=bg mode,max rate]<CR><LF><CR><LF>

参数：bg mode：BG 模式，其中 0 表示 B/G 混合模式、1 表示 B 模式。

系统缺省设置为 B/G 混合模式，通常情况下用户无需修改此参数。

max rate：最高发送速率，在 B 模式下，仅 0~3 有效。其中：

0 代表 1 Mbps	4 表示 6 Mbps	8 表示 24 Mbps
1 表示 2 Mbps	5 表示 9 Mbps	9 表示 36 Mbps
2 表示 5.5 Mbps	6 表示 12 Mbps	10 表示 48 Mbps
3 表示 11 Mbps	7 表示 18 Mbps	11 表示 54 Mbps

系统缺省设置为 36M，通常情况下用户无需修改此参数。当用户的使用环境比较恶劣，无线信号比较差时，可以修改本参数来降低无线发射速率，从而提高无线传输的可靠性。

AT+BRDSSID

功能：使能/禁止 AP SSID 广播，该设置仅在模块作为 AP 时有效。

格式：AT+BRDSSID=[!?][enable]<CR>

+OK[=enable]<CR><LF><CR><LF>

参数：enable：是否使能 SSID 广播。其中 0 表示禁止 AP SSID 广播、1 表示使能 AP SSID 广播。

AT+UART

功能：设置/查询 uart 接口数据格式。

格式：AT+UART=[!?][baud rate],[data bit],[stop bit],[parity]<CR>

+OK[=baud rate,data bit,stop bit,parity]<CR><LF><CR><LF>

参数：baud rate：波特率，有效值范围 1200~115200，其中：

115200 表示 115200 bps	9600 表示 9600 bps
57600 表示 57600 bps	4800 表示 4800 bps
38400 表示 38400 bps	2400 表示 2400 bps
19200 表示 19200 bps	1200 表示 1200 bps

data bit：数据位。其中 0 表示 8 位、1 表示 7 位

stop bit：停止位。其中 0 表示 1 位、1 表示不支持、2 表示 2 位

parity：校验。其中 0 表示无校验、1 表示奇校验、2 表示偶校验

**AT+ATPT**

功能：设置/查询数据自动组帧周期。本参数仅在串口透明传输模式下有效。

格式：AT+ATPT=[!?][period]<CR>

+OK[=period]<CR><LF><CR><LF>

参数：period：自动组帧周期，50~10000ms，单位 ms，最小步长 50ms 串口在透明传输模式下，当接收到数据的长度不足组帧长度时，强制自动帧的超时时间，可选范围为 50~10000ms，最小精度 50ms。系统缺省设置为 500ms。本设置必须小于逃逸时间设置。

AT+ATLT

功能：设置/查询数据自动组帧数据长度。本参数仅在串口透明传输模式下有效。

格式：AT+ATLT=[!?][length]<CR>

+OK[=length]<CR><LF><CR><LF>

参数：length：自动组帧长度，32~1024，单位：字节

串口在透明传输模式下，当接收到数据达到本参数指定长度时，强制组成网络帧并触发发送流程，可选范围为，32~1024。系统缺省设置为 512 个字节。

AT+ESPC

功能：设置/查询逃逸字符。本参数用于退出串口透明传输模式。

格式：AT+ESPC=[!?][escape]<CR>

+OK[=escape]<CR><LF><CR><LF>

参数：escape：逃逸字符。

本参数指定用于退出串口透明传输模式字符，如"+", 系统在逃逸字符检测状态下，当接收到有且只有 3 个连续的逃逸字符时，将退出串口透明传输模式。系统缺省设置为 0x2B，即 ASCII 字符"+"。

AT+ESPT

功能：设置/查询逃逸时间。本参数用于退出串口透明传输模式。

格式：AT+ESPT=[!?][time]<CR>

+OK[=time]<CR><LF><CR><LF>

参数：time：逃逸时间，100~10000ms，单位 ms，最小步长 100ms。

AT+WEBS

功能：设置/查询内置是否使能 WEB 管理服务器。

格式：AT+WEBS=[!?][enable],[port]<CR>

+OK[=enable,port]<CR><LF><CR><LF>

参数：enable：使能标志。其中 0 表示不使能、1 表示使能。

port：服务器端口号，缺省为 80。

AT+PASS

功能：设置/查询系统登录密码。

格式：AT+PASS=[!?][pass]<CR>

+OK[=pass]<CR><LF><CR><LF>

参数：pass：6 个 ACSII 字符。

当通过 WEB 服务器或无线配置连接模块时的登录密码。系统缺省设置为“000000”。

**AT+IOM**

功能：设置/查询 GPIO1 工作模式。

格式：AT+IOM=[!?][mode]<CR>

+OK[=mode]<CR><LF><CR><LF>

参数：mode：工作模式。其中 0 表示系统功能、1 表示输入、2 表示输出。

AT+CMDM

功能：设置/查询系统的默认命令模式。

格式：AT+CMDM=[!?][mode]<CR>

+OK[=mode]<CR><LF><CR><LF>

参数：mode：命令模式。其中 0 表示 AT+指令、1 表示兼容协议。

5.2.5 出厂默认设置

表 5-5 HC-21 (TLN13UA06) 模块出厂默认设置

	参数名	参数值	说明
无线	工作模式	基础网络-AP	其他两种模式：基础网络-SAT、对等网络。 此处如此设置，模块将创建一个名为 wifi-socket 的无线 WiFi 信号，开发网络无需密匙。
	网络名称	wifi-socket	
	安全模式	OPEN	
网络	DHCP	Enable	AP 模式下，DHCP 协议动态获取 IP 地址功能默认启用。此时连接 wifi-socket 的设备使用浏览器，可进入 WEB 服务器配置模块。 网卡域名参数仅在 AP 模式下有效，所配置域名长度不大于 32 字节。
	Ip 地址	192.168.1.100	
	子网掩码	255.255.255.0	
	网关地址	192.168.1.1	
	DNS 服务器	192.168.1.1	
	网卡域名	Local.hed.com.cn	
工作模式	自动工作模式	Enable	启用自动工作模式，即透明传输模式。 模块上电后的默认工作模式为自动工作模式，即串口透明传输模式，因此对模块进行配置时，需要先退出透明传输模式，进入命令模式。 使用 PC 端配置软件搜索模块，提示串口被占用或设备未连接时，可尝试点击“退出透传模式”。
	协议类型	UDP	
	CS 模式	服务器	
	服务器地址	-	
	端口号	50000	
	TCP 连接超时	0	
高级	网络不存在时自动创建	Disable	默认网络不存在时自动创建失效。
	BG 模式	B/G	无线网络模式为 BG 混合。
	最高速率	24M	无线最高发射速率 54Mbps。
	BSSID	Auto	禁止绑定无线网络的 BSSID 地址。
	信道	Auto	自动监测目的网络的工作信道。
	自动重试	从不	从不自动重试。
	无线漫游	关闭	禁止无线网络漫游。
	信道列表	0x3FFF	1~14 所有信道。
	SSID 广播	Enable	使能 SSID 广播，该参数仅在 AP 模式下有效。
串口	波特率	9600	模块默认波特率：9600bps。 串口数据格式：8 位数据位/1 位停止位/无效验位 使用 PC 端配置软件搜索模块，提示串口被占用或设备未连接时，可检查波特率是否正确。
	数据位	8 位	
	停止位	1 位	
	校验位	无效验位	



透明模式	自动组帧长度	512	自动组帧长度为 512 字节。
	自动组帧周期	0	自动组帧周期 0 代表自动。
	逃逸时间	2000ms	逃逸时间必须大于自动组帧周期。
	逃逸字符	0x2B	即 ACSII 字符“+”。
系统	系统密码	“000000”	系统登录密码为字符串“000000”。
	WEB 服务器	Enable	启用 WEB 服务器。
	端口号	80	连接时示例：192.168.1.100: 80。
	命令模式	AT+	系统命令模式为使用 AT+指令集。
	GPIO1 模式	System	GPIO1 设置为系统功能。

以上为模块出厂设置，详细参数可使用配置管理软件查看。查看方法参考第 3 章。

5.3 例程代码

本节主要针对命令模式下对 WiFi 模块的控制给出上位机的示例代码。如果模块运行在自动模式，将不需要这部分内容。

5.3.1 例程代码说明

本例程代码是基于 C 语言编写的例程代码，可以将其移植到其他的系统中。本例程代码主要是对 WiFi 模块的操作给出一个操作的过程示例。在移植时需要根据自己系统的状况进行相应的修改。

5.3.2 命令模式下 WiFi 网卡的操作流程

通常在命令行模式下对 WiFi 模块的操作的流程如图 5-3 所示。在使用 WiFi 模块时，可以根据系统的情况决定是否进行命令行的操作。

5.3.3 示例代码主要的数据结构

5.3.3.1 At 命令结构体

```
typedef struct_TATCP_COMMAND{
char *CmdName;
INT Flag;
char *Cmdbuf;
INT *Cmdbuflen;
}TATCP_CMD;
```

说明：CmdName：At 指令命令代号。

Flag：该命令是否有数据返回标志。

Cmdbuf：该命令所含命令内容

Cmdbuflen：命令长度

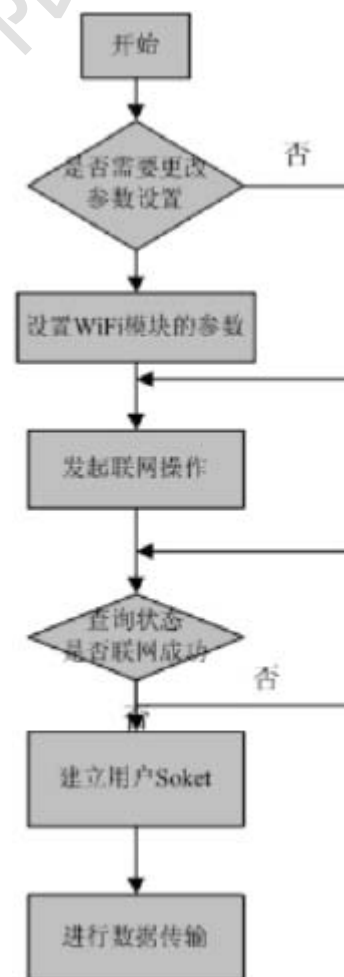


图 5-3 WiFi 网卡操作流程

5.3.3.2 串口 AT 指令发送标志

```
unsigned char AtCmdSending;
```

说明：该标志说明有一条 At 指令正在发送或等待回复。不能继续下发另外，一条指令。

如果用户系统使用基于操作系统的编程，可以使用信号量的机制实现。



5.3.3.3 串口等待 AT 指令回复标志

unsigned char ReciveAtCmdrep;

说明：该标志代表等待某条 AT 指令的回复，如果使用操作系统的话，可以使用。

5.3.4 串口数据操作

在串口发送每条 CMD 后，需要等待该命令的回复。如果本条命令没有回复，需要增加超时，将强制退出该命令。应当保证 At 发送命令时，串口只有一条在等待回复的 At 指令。

5.3.4.1 串口发送指令示例

```
int AtCmdSend(TATCP_CMD*AtCmd)
{
    unsigned char AtCmdbuf[512];
    int ret=0;
    ////
    ///AtCmd_Check();///AtCmdCheck
    ////
    strcpy(AtCmdbuf,"AT+");
    strcat(AtCmdbuf,AtCmd->Cmdbuf);
    ReciveAtCmdrep=0;
    ComSendstr(AtCmdbuf);
    AtcmdTimerId=StartTimer(MSG_CMD_TIMEOUT_CFM,TIMER_ATCMD_TIMEOUT);
    while(ReciveAtCmdrep!=1)
    {
        if(Atcmdtimeout==1)
        {
            ret=-1;
            break;
        }
    }
    StopTimer(AtcmdTimerId);
    return ret;
}
```

说明：本示例代码中需要系统另外提供超时处理的函数。在串口发送 At Cmd 后启动定时器，在给定的时间内将 Atcmdtimeout 置为 1。以便实现 WiFi 模块超时后的处理。

5.3.4.2 串口接收 AT 指令回复示例

用于串口接收 WiFi 模块回传的响应信息，在上位机发送完成每条 AT 指令后需要等待 WiFi 模块响应或者超时（请将超时时间设置大于 500ms）。本例子以 51 体系下串口中断为例。

1、串口接收中断函数：

```
void COMM_INT()interrupt 4 using 3
{
    unsigned char cBuf;
    if(RI)//接收数据
    {
        RI=0;
```



```
        cBuf=SBUF;
        ProcessUARTData(cBuf);
    }
    else if(TI)
    {
        TI=0;
        if(iSendCounter<iUartSendLen)
            SBUF=cpSendBufPtr[iSendCounter++];
    }
}
```

说明：ProcessUARTData（cBuf）用来处理串口接收到的数据。

2、串口接收到数据处理函数，主要数据结构描述：

```
#define WIFIBUFLEN 1024
typedef struct WiFiRecBuffer
{
    unsigned char gcATReceData[WIFIBUFLEN];//
    unsigned int gcATReceCount;//接收字节长度
}WiFiRecBuffer;
struct WiFiRecBuffer xdata ATReceBuffer[1];//51 系统外部 ram
```

3、串口接收 AT 返回状态宏定义，该定义主要用于控制串口接收 AT 指令返回的流程。

```
#define WIFI_WAIT_SYN 0X00////串口初始状态
#define WIFI_WAIT_RESP 0x01////串口等待相应字符
#define WIFI_RECEDATA 0x02////等待数据
#define WIFI_WAIT_OVER 0x03////等待接收结束字符
#define WIFI_WAIT_ERRCODE 0x04////等待接收错误代码
#define WIFI_SYN 0X2B////+收到同步字符状态
```

4、串口接收 At 指令回复解析函数 ProcessUARTData（cBuf）

```
void ProcessUARTData(IN unsigned char UartData)
{
    int i=0;
    switch(gcWiFiCommunState)
    {
        case WIFI_WAIT_SYN: //接收等待
            if(UartData==WIFI_SYN)
            {
                gcWiFiCommunState=WIFI_WAIT_RESP;
            }
            break;
        case WIFI_WAIT_RESP:
            AtResp[AtResplen++]=UartData;
            if(strcmp(AtResp,"OK")==0)
            {
```



```
        AtRespState=1;///response ok
        gcWiFiCommunState=WIFI_RECEDATA;
        AtResplen=0;
    }
    else if(strcmp(AtResp,"ERR")
    {
        AtRespState=0;///response ERR
        gcWiFiCommunState=WIFI_WAIT_ERRCODE;
        AtResplen=0;
    }
    else
    {
        printf("Recive unknown words\n");
        AtRespState=0;///response ERR
        gcWiFiCommunState=WIFI_WAIT_SYN;
        AtResplen=0;
    }
    break;
case WIFI_RECEDATA:
case WIFI_WAIT_OVER:
case WIFI_WAIT_ERRCODE:
    ATReceBuffer[0].gcATReceData[ATReceBuffer[0].gcATReceCount]=UartData;
    ATReceBuffer[0].gcATReceCount++;
    if(strstr(ATReceBuffer[0].gcATReceData,"\r\n\r\n")!=NULL)
    {
        ReciveAtCmdrep=1;
        gcWiFiCommunState=WIFI_WAIT_SYN;
        ATReceBuffer[0].gcATReceCount=0;
    }
    else if(ATReceBuffer[0].gcATReceCount==WIFIBUFLEN)
    {
        printf("At cmd recive error\n");
        gcWiFiCommunState=WIFI_WAIT_SYN;
    }
    else
    {
        printf("Wiat next byte\n");
    }
    break;
default:
    gcWiFiCommunState=WIFI_WAIT_SYN;
    break;
}
}
```



说明：此函数用来处理串口接收到的数据，在串口每接收到一个字节的数据时都应当调用此函数进行判断处理。函数在完全接收到 AT 指令的回复后，函数会置位 ReciveAtCmdrep，此时在 AT 发送进程中可以根据此位进行判断，然后根据每条 AT 指令不同的回复进行解析处理。

5.3.5 AT 指令返回参数解析函数

此函数用来解析含有返回参数的 AT 指令的返回参数，将每条返回的指令的参数进行拆分处理。函数示例如下：

```
INT parse_line(char*line,char*argv[])
{
    INT nargs=0;
    while(nargs<ATCP_MAXARGS){
        /*skip any white space*/
        while((*line=="")||(*line=='\t')){
            ++line;
        }
        if(*line=='\0'){/*end of line,no more args*/
            argv[nargs]=NULL;
            return(nargs);
        }
        if(*line==""){/*argument such as"xxxx"*/
            line++;/*skip 1st">*/
            argv[nargs++]=line;/*begin of argument string*/
            /*find end of string*/
            while(*line&&(*line!="")){
                ++line;
            }
            if(*line=='\0'){/*end of line,but 2nd">is not found*/
                argv[nargs]=NULL;
                return(-1);
            }
            *line++='\0';/*terminate current arg*/
            /*skip any white space*/
            while((*line=="")||(*line=='\t')){
                ++line;
            }
            if(*line==','){
                line++;/*skip<,>*/
            }
            else if(*line=='\0'){/*end of line,no more args*/
                argv[nargs]=NULL;
                return(nargs);
            }
            else{/*unexpected char*/
                argv[nargs]=NULL;
                return(-1);
            }
        }
    }
}
```



```
    }
}
else{
    argv[nargs++]=line;/*begin of argument string*/
    /*find end of string*/
    while(*line&&(*line!=',')){
        ++line;
    }
    if(*line=='\0'){/*end of line,no more args*/
        argv[nargs]=NULL;
        return(nargs);
    }
    *line++='\0';/*terminate current arg*/
}
}
return(-1);
}
```

函数说明：char*line 指向串口接收到的数据，char*argv[]指向存放返回参数的指针，用于后面的参数解析。

5.3.6 字符串操作函数

下面几个函数主要应用于解析 At 返回指令中的字符串参数，将字符串参数转换为 CPU所示的十六进制或者十进制。

```
INT atodec(char Ch)
{
    INT dec=-1;
    if(Ch>='0'&&Ch<='9')
        dec=Ch-'0';
    return dec;
}
INT strtodec(INT*dec,char*str)
{
    INT i=-1;
    INT dd=0,sign=1;
    if(*str=='-')
    {
        str++;
        sign=-1;
    }
    while(*str)
    {
        i=atodec(*str++);
        if(i<0){return-1;}
        dd=dd*10+i;
    }
}
```



```
*dec=dd*sign;
return(i<0?-1: 0);
}
INT atohex(char Ch)
{
    INT hex=-1;
    if(Ch>='0'&&Ch<='9')
        hex=Ch-'0';
    else if(Ch>='a'&&Ch<='f')
        hex=Ch-'a'+0xa;
    else if(Ch>='A'&&Ch<='F')
        hex=Ch-'A'+0xa;
    return hex;
}
INT strtohexarray(INT8U array[],INT cnt,char*str)
{
    INT hex;
    INT8U tmp,*des;
    des=array;
    while(cnt-->0)
    {
        hex=atohex(*str++);
        if(hex<0)
            return-1;
        else{
            tmp=(hex<<4)&0xf0;
        }
        hex=atohex(*str++);
        if(hex<0)
            return-1;
        else{
            tmp=tmp|(hex&0x0f);
        }
        *des++=(INT8U)tmp;
    }
    return(*str=='0'?0: -1);
}
```

5.3.7 AT 指令发送解析示例

5.3.7.1 简单控制类指令发送示例

此命令仅仅为控制网卡的动作，不需要在 AT 指令中携带大量的参数数据，并且此类指令在收到 resp 后即为执行正确。



1、以控制 WiFi 模块加网操作进行示例：

WiFi 模块加网操作使用到的数据结构如下：

```
#define RespParm 1
#define NoRespParm 0
#define ATC_CH_SET '='
#define ATC_CH_SAVE '!'
#define ATC_CH_QUERY '?'
typedef struct _JionResult{
    unsigned char BSSID[12];
    unsigned char Type;///0 infra 1---adhoc
    unsigned char channel;
    unsigned char encry;
    char ssid[32];
    int rssi;
}TJION_RESULT;
```

2、WiFi 模块加入网络控制的主函数如下所示，调用此函数系统将挂起，直到超时或收到命令的回复为止。

```
void WiFi_Jion(void)
{
    TATCP_CMD WiFiJoin;
    TJION_RESULT Jionr;
    unsigned char ErrorCode=0;
    char*arg=NULL;
    INT argc=0;
    int argpoint=0;
    INT err=0;
    char*argv[ATCP_MAXARGS];
    char*ATCmd="WJOIN\n";
    memset(WiFiJoin,0,sizeof(TATCP_CMD));
    memset(ATReceBuffer[0].gcATReceData,0,sizeof(WiFiRecBuffer));
    WiFiJoin.Flag=RespParm;
    WiFiJoin.Cmdbufl=ATCmd;
    WiFiJoin.Cmdbuflen=strlen(ATCmd);
    AtCmdSend(WiFiJoin);
    if(AtRespState==1)
    {
        arg=strchr(ATReceBuffer[0].gcATReceData,ATC_CH_SET);
        if(arg)
        {
            {
                argc=parse_line(arg,argv);
                if(argc<=0)
                {
                    printf("Recive Error parm\n");
                }
            }
        }
    }
}
```



```
        return;
    }
}
if(argc!=6)
{
    printf("recvie parm number error\n");
}
else
{
    if(strtohexarray(Jionr.BSSID,6,argv[0])<0)
    {
        printf("Error Bssid\n");
        return;
    }
    if(strtodec(&(Jionr.Type),argv[1])<0)
    {
        return;
    }
    if(strtodec(&(Jionr.channel),argv[2])<0)
    {
        return;
    }
    if(strtodec(&(Jionr.encry),argv[3])<0)
    {
        return;
    }
    strcpy((char*)Jionr.Ssid,argv[4]);
    if(strtodec(&(Jionr.rssi),argv[5])<0)
    {
        return;
    }
}
}
else
{
    ErrorCode=*(arg+2);
    printf("reciver error At cmd resp ERROR Code:  %d",ErrorCode);
}
}
```

5.3.7.2 参数设置类指令发送示例

参数设置类指令主要用于上位机对 WiFi 模块进行参数设置，AT 指令中含有较多的参数设置。



1、以设置密钥进行示例：

在密钥设置中所使用的主要数据结构：

```
typedef struct _WiFiKey{  
    unsigned char PSK[64];  
    unsigned char KeyLength;  
    unsigned char KeyIndex;  
    unsigned char KeyFormat;  
}WiFiKey;
```

2、设置密钥的主函数如下：

```
void WiFi_SetKey(unsigned char NeedFlash)  
{  
    WiFiKey KeySet;  
    WiFiKey KeyResult;  
    TATCP_CMD ATWiFiJoin;  
    unsigned char ErrorCode=0;  
    char*arg=NULL;  
    INT argc=0;  
    char WireFlash;  
    int argpoint=0;  
    INT err=0;  
    char*argv[ATCP_MAXARGS];  
    char*ATCmd="KEY=";  
    char Atbuf[128];  
    memset(WiFiKey,0,sizeof(WiFiKey));  
    memset(ATReceBuffer[0].gcATReceData,0,sizeof(WiFiRecBuffer));  
    memset(Atbuf,0,128);  
    strcpy(WiFiKey.PSK,"12345678");  
    WiFiKey.KeyLength=8;  
    WiFiKey.KeyIndex=1;  
    WiFiKey.KeyFormat=1;  
    if(NeedFlash)  
    {  
        WireFlash='!';  
    }  
    else  
    {  
        WireFlash=NULL;  
    }  
    sprintf(Atbuf,"%s%c%d,%d,\"%s\\\"\\n\",ATCmd,WireFlash,WiFiKey.KeyFormat,  
    WiFiKey.KeyIndex,WiFiKey.PSK);  
    ATWiFiJoin.Flag=1;  
    ATWiFiJoin.Cmdbuf=Atbuf;  
    ATWiFiJoin.Cmdbuflen=strlen(Atbuf);  
    AtCmdSend(ATWiFiJoin);
```

```
if(AtRespState==1)
{
    arg=strchr(ATReceBuffer[0].gcATReceData,ATC_CH_SET);
    if(arg)
    {
        {
            argc=parse_line(arg,argv);
            if(argc<=0)
            {
                printf("Recive Error parm\n");
                return;
            }
        }
        if(argc!=3)
        {
            printf("recvie parm number error\n");
        }
        else
        {
            if(strtodec(&(KeyResult.KeyFormat),argv[0])<0)
            {
                return;
            }
            if(strtodec(&(KeyResult.KeyIndex),argv[1])<0)
            {
                return;
            }
            strcpy((char*)KeyResult.Psk,argv[2]);
        }
    }
}
else
{
    {
        ErrorCode=*(arg+2);
        printf("reciver error At cmd resp ERROR Code: %d",ErrorCode);
    }
}
```

5.3.7.3 较复杂返回参数的 AT 指令示例

此类的 AT 指令有较多的参数返回值，需要解析的参数较多。以扫描为例示意此类指令的操作。

1、分离参数组函数

参数组分离函数用来将 AT 指令返回的较长的参数进行分组。



```
INT parse_line_group(char*line,char*argv_group[])
{
    INT nargs=0;
    while(nargs<ATCP_MAXARGS){
        /*skip any white space*/
        while((*line=="")||(*line=="\t")){
            ++line;
        }
        if(*line=="\0"){/*end of line,no more args*/
            argv_group[nargs]=NULL;
            return(nargs);
        }
        {
            line++;
            argv_group[nargs++]=line;/*begin of argument group*/
            /*find end of string*/
            while(*line!='\r'&&*(line+1)!='\n'){
                ++line;
            }
            if(*line=="\0"){/*end of line,but 2nd">"is not found*/
                argv_group[nargs]=NULL;
                return(nargs);
            }
            *line++="\0";/*terminate current arg group*/
            *line++="\0";
            if(*line=="\r"&&*(line+1)=="\n"){
                *line++="\0";/*terminate current arg group*/
                *line++="\0";
                return(nargs);
            }
        }
    }
    return(-1);
}
```

2、模块扫描命令发送示例

在发送扫描命令后，WiFi 模块会返回非常长的参数区。最长可以达到 2K 之多。因此我们将这部分作为示例，发送扫描命令 AT 指令如下，

```
void WiFiScan(void)
{
    TATCP_CMD WiFiScan;
    unsigned char ErrorCode=0;
    char*arggroup=NULL;
    char*arg=NULL;
    INT arggroupc=0;
```



```
INT argc=0;
int  argpoint=0;
INT err=0;
Int  i=0;
char*argv[ATCP_MAXARGS];
char*arggroupv[ATCP_MAXARGS];
char*ATCmd="WSCAN\n";
memset(WiFiJoin,0,sizeof(TATCP_CMD));
memset(ATReceBuffer[0].gcATReceData,0,sizeof(WiFiRecBuffer));
WiFiScan.Flag=RespParm;
WiFiScan.Cmdbuf=ATCmd;
WiFiScan.Cmdbuflen=strlen(ATCmd);
AtCmdSend(WiFiScan);
If(ATRespState==1)
{
    arggroup=strchr(ATReceBuffer[0].gcATReceData,ATC_CH_SET);
    //arg=strchr(ATReceBuffer[0].gcATReceData,ATC_CH_SET);
    if(arggroup)
    {
        {
            arggroupc=parse_line_group(arggroup,arggroupv);
            if(argc<=0)
            {
                printf("Recive Error parm\n");
                return;
            }
        }
        for(i=0;i<arggroupc;i++)
        {
            argc=parse_line(&arggroupv[arggroupc],argv);
            if(argc<=0)
            {
                printf("Recive Error parm\n");
                return;
            }
        }
        else
        {
            if(strtohexarray(Scanresult[arggroupc].BSSID,6,argv[0])<0)
            {
                printf("Error Bssid\n");
                return;
            }
            if(strtodec(&(Scanresult[arggroupc].Type),argv[1])<0)
            {
                return;
            }
        }
    }
}
```



```
    }
    if(strtodec(&(Scanresult[arggroupc].channel),argv[2])<0)
    {
        return;
    }
    if(strtodec(&(Scanresult[arggroupc].encry),argv[3])<0)
    {
        return;
    }
    strcpy((char*)Scanresult[arggroupc].Ssid,argv[4]);
    if(strtodec(&(Scanresult[arggroupc].rssi),argv[5])<0)
    {
        return;
    }
}
}
else
{
    ErrorCode=*(arg+2);
    printf("reciver error At cmd resp ERROR Code: %d",ErrorCode);
}
}
```

附录 常见问题与解答

Q1: 模块为何搜索不到指定的网络?

A: 用户参数中信道列表设置不包含指定网络所在的信道。

Q2: 模块为什么连接不上网络?

A: 请仔细检查模块设置的SSID、密钥、信道等参数与 AP 的设置是否一致模块是否设置了指定BSSID, 且该 BSSID 与该网络的 BSSID 不一致。

Q3: 模块设置为 WPA/WPA2 安全模式时, 为什么在复位后需要 10 多秒才能响应指令或自动连上网络?

A: 当安全模式为 WPA/WPA2, 而且密钥格式设置为 ACSII 格式时, 模块在上电复位时需要首先将对 ACSII 格式的密钥进行转换, 这个过程需要增加大约 10 秒左右的模块启动时间, 将密钥格式设置为使用 16 进制格式可以消除这个延迟, 具体方法参见 4.3.1.5 节。

Q4: 模块已经显示加网成功, 为什么 ping 不通?

A: 请检查模块的网络设置, 即 IP 地址等是否正确模块的安全模式设置是否与 AP 的设置完全一致, 因为在有些安全模式下(如 AP 设置为 WEP 加密, 开放鉴权), 即使密钥设置不正确, AP 依然能连接成功, 但是却无法正常通信可能有正在运行的软件或防火墙阻止 ping 操作。



Q5: 自动工作模式下, 为什么模块联网后总是一会儿能 ping 通, 一会儿又 ping 不通?

A: 模块在自动模式下联网成功后会自动创建默认的网络连接, 如果默认网络连接的目标 TCP 服务器配置不正确或没有开启, 那么模块会在连接失败后自动断开无线网络, 并在延时 10 秒后尝试重新连接。于是就会出现模块时而能 ping 通时而 ping 不通的情况。正确配置 TCP 服务器并开启即可解决此问题。

Q6: 自动工作模式下, 为什么通过串口发送到模块的数据, 延迟一段时间才被发送到网络上?

A: 在串口透明传输状态下, 数据组帧发送有两个触发条件: 触发时间和触发长度, 如果数据较小达不到组帧的长度要求, 将会在等待触发时间超时后才强制组帧并发送, 详见 1.3.1.1 小节。

Q7: 自动工作模式下, 输入逃逸字符后为什么无法退出透明传输状态?

A: 退出透明传输状态有严格的条件限制, 详见 1.3.3.1 小节。

Q8: 为什么配置程序搜索不到模块?

A: 配置程序的串口设置与模块的串口设置不一致模块处于自动工作模式下, 请先推出自动工作模式, 再进行相关的搜索。

Q9: 使用配置程序修改参数, 为什么返回使用了禁止的操作符错误?

A: 请首先切换到功能测试页点击“断网”, 然后再提交参数修改。

Q10: 为什么打不开 web 服务器网页?

A: web 服务器是否被禁用 Web 服务器端口设置是否正确, 如果设置端口为除 80 外的其它端口, 那么在使用浏览器访问时需要在地址栏中输入端口号, 如 `http://192.168.1.100:2000`。

Q11: 用户设置了自身不支持的波特率, 导致的再也无法连接模块该怎么办?

A: 可以通过 bootloader 来恢复出厂设置, 步骤如下:

- 1、打开串口调试软件, 如串口大师, 串口设置为 115200/8 位数据位/1 位停止位/无校验/无流控;
- 2、在软件的发送数据栏中输入“at+” (不含引号, 含回车), 选择周期发送, 周期设为 50ms;
- 3、复位模块, 如果此时模块返回“+OK”, 停止周期发送;
- 4、发送指令“AT+QVER”, 查看返回信息, 如果包含 B1.xx 字段, 则成功进入 bootloader, 否则, 重复步骤 2;
- 5、发送指令“AT+RSTF” (无返回信息);
- 6、等待 1 秒后将模块复位即可将模块恢复到出厂设置状态。

Q12: 无线连接时为什么搜索不到指定的模块?

A: 无线连接的网卡与指定模块之间的信号不好; 无线连接设置中的系统密码与指定模块的系统密码不一致; 无线连接设置中 MAC 地址限定。

Q13: 在我的使用环境下, 为什么无线网络连接的质量总是很差?

A: 导致无线网络连接质量不好的原因很多, 如 AP 的位置布置不合理、AP 与模块之间的距离过远、有明显的屏蔽或遮挡、环境中 2.4G 频段的干扰等等。有些环境问题是无法从根本上改变的, 只能寻求一些减小环境带来的不良影响的方法。

您可以尝试进行以下的改变:

- 1、尽量将 AP 布置在空旷、没有明显遮挡的位置, 尤其要避免放置在大面积的金属附近、墙角等这些地方;



- 2、修改 AP 设置，将工作信道设成远离当前的信道，如 1 改到 6，6 改到 13 等；
- 3、修改模块高级无线设置参数中的最高速率，降低到 11M 甚至更低。

Q14: 为什么在模块工作在 Sever 模式下时，TCP 连接会周期性的断开？

A: WiFi 模块工作在 Server 模式下时，如果终端 Client 在 Timeout 周期内没有任何数据的收发。模块会主动的与 Client 断开。如果用户需要，可将 Timeout 时间设置为 0，此时 Sever 不会周期性的与 Client 断开，但是此时会占用 WiFi 模块的 socket 资源。

Q15: 模块无法进行收发数据，如何判断模块的物理层的好坏？

A: 请根据 Q12 中的描述，执行其前 3 步。如果能够正常的看到回应。应该说明 WiFi 模块基本上硬件不会有问题。否则，请更换为新的模块。

Q16: 是否可以通过管脚 link 来判断模块是否已经联网成功？

A: WiFi 模块的 link 引脚仅仅是指示模块的 WiFi 是否存在数据的收发，不能够判断 WiFi 模块的联网状态。如果判断模块的联网状态请使用 RTS 引脚判断模块的联网状态。

Q17: 模块上电后没有任何反应，具体是什么原因？

A: 首先，确定模块的供电电源，模块的供电电源为 3.3V，供电电源最少应该提供 350mA 的供电电流。

然后，确认模块上的 LED 灯是否正常亮起。

再次，如果模块没有反应，请手动对模块进行 reset。

最后，根据 Q16 的描述判断模块是否已经物理层损坏。

Q18: 模块发送距离较近时，如何处理？

A: 在模块发送距离较近时，可以通过将模块设置为 11b 的工作模式，并将模块的速率调整为 1M 的速率。

Q19: 在使用网页进行模块的固件升级时，请使用 windows 自带的 IE 浏览器进行。不要使用诸如 360 等浏览器。

Q20: 模块启动问题？

A: UA06 模块在初始启动的过程中需要对内部一些硬件进行初始化的处理，启动大概需要 1-2s 的时间，用户需要在设计系统的时候予以注意。

Q21: 网页参数配置需要注意事项？

A: 网页进行参数配置时，需要将所用到的参数进行一次性的全部配置好以后，然后在 system 页面中点击 Restart System 后，系统才能够使用正确的参数进行工作。