

## 简介

HLK-WIFI-M03 是海凌科（香港）有限公司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。

HLK-WIFI-M03 在前两代产品的基础上进行了全面的软硬件升级，功能更加强大，使用更加简单：

- ✧ 全面支持串口透明数据传输模式，真正实现串口的即插即用
- ✧ 全新的 AT+指令集，全部基于 ASCII 格式，简单易懂，使用更容易
- ✧ 更加完备的 TCP/IP 协议栈，支持 DHCP 协议动态 IP 地址分配和 DNS 域名解析功能
- ✧ 内置 WEB 服务器，实现使用 IE 浏览器通过无线网络远程配置模块参数
- ✧ 更高的传输性能，最高可达 11KB/S 的发送速率

## 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

### 无线

- ✧ 支持 IEEE802.11b/g 无线标准
- ✧ 支持频率范围：2.412~2.484 GHz
- ✧ 支持两种无线网络类型：  
基础网（Infra）、自组网（Adhoc）h  
和 AP 模式

- ✧ 支持多种安全认证机制：  
WEP64/WEP128/TKIP/CCMP(AES)  
WEP/WPA-PSK/WPA2-PSK  
AP 模式下仅支持 WEP64/WEP128

- ✧ 支持快速联网
- ✧ 支持无线漫游

### 其它

- ✧ 支持多种网络协议：  
TCP/UDP/ICMP/DHCP/DNS/HTTP
- ✧ 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/无线连接

## 应用领域

- ✧ 智能公交网络，如无线刷卡机
- ✧ 小额金融支付网络，如无线 POS 机
- ✧ 工业设备联网，如无线传感器
- ✧ 物联网

## 目 录

<b>1</b>	<b>产品简介.....</b>	<b>5</b>
<b>2</b>	<b>产品综述.....</b>	<b>6</b>
2.1	技术规格.....	6
2.2	硬件说明.....	7
2.2.1	机械尺寸.....	7
2.2.2	外部接口.....	8
2.2.3	天线（选配）.....	9
2.2.4	串口转接板（选配）.....	9
2.2.5	无线适配器（选配）.....	10
2.3	主要功能.....	11
2.3.1	连接无线网络.....	11
2.3.2	串口即插即用.....	13
2.3.3	串口命令模式.....	14
2.3.4	TCP/IP协议栈.....	14
2.3.5	灵活的参数配置.....	14
2.3.6	多功能GPIO.....	15
2.3.7	固件在线升级.....	15
<b>3</b>	<b>快速开始向导.....</b>	<b>16</b>
3.1	准备工作.....	16
3.2	设置参数.....	16
3.3	开始测试.....	18
<b>4</b>	<b>产品设计指南.....</b>	<b>21</b>
4.1	硬件连接.....	21
4.2	工作模式.....	21
4.2.1	启动模式选择.....	21
4.2.2	自动工作模式.....	22
4.2.3	命令工作模式.....	25
4.3	参数详细说明.....	25
4.3.1	无线设置参数.....	25
4.3.2	网络设置参数.....	28

4.3.3	工作模式设置参数.....	29
4.3.4	高级无线设置参数.....	30
4.3.5	串口设置参数.....	34
4.3.6	透明模式设置参数.....	35
4.3.7	系统设置参数.....	36
4.4	出厂默认设置.....	37
4.5	典型应用范例.....	39
4.5.1	主动型串口设备联网.....	39
4.5.2	被动型串口设备联网.....	40
4.5.3	广播方式的串口设备联网.....	41
4.5.4	无线虚拟串口.....	43
4.5.5	多网络连接模式.....	45
<b>5</b>	<b>配置管理指南.....</b>	<b>48</b>
5.1	使用配置管理软件.....	48
5.1.1	简介.....	48
5.1.2	模块管理.....	48
5.1.3	配置参数.....	51
5.1.4	功能测试.....	56
5.1.5	系统信息.....	65
5.1.6	固件升级.....	65
5.2	使用WEB服务器.....	66
5.2.1	查询/修改配置参数.....	66
5.2.2	在线固件升级.....	69
5.3	使用超级终端程序.....	69
<b>6</b>	<b>AT+指令控制协议.....</b>	<b>72</b>
6.1	语法说明.....	72
6.1.1	语法格式.....	72
6.2	指令集.....	75
6.2.1	指令列表.....	75
6.2.2	网络控制类.....	77
6.2.3	系统控制类.....	81
6.2.4	参数设置类.....	84

6.3	例程代码.....	95
6.3.1	例程代码说明.....	95
6.3.2	命令模式下Wifi网卡的操作流程.....	95
6.3.3	示例代码主要的数据结构.....	96
6.3.4	串口数据操作.....	97
6.3.5	AT指令返回参数解析函数.....	101
6.3.6	字符串操作函数.....	102
6.3.7	几种AT指令发送解析示例.....	104

## 附录A 常见问题与解答 ..... 113

Q1: 模块为何搜索不到指定的网络? .....	113
Q2: 模块为什么连接不上网络? .....	113
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? .....	113
Q4: 模块已经显示加网成功, 为什么PING不通? .....	113
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? .....	113
Q6: 自动工作模式下, 为什么模块复位后没有自动联网? .....	113
Q7: 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? .....	114
Q8: 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? .....	114
Q9: 为什么配置程序搜索不到模块? .....	114
Q10: 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? .....	114
Q11: 为什么打不开WEB服务器网页? .....	114
Q12: 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? .....	114
Q13: 无线连接时为什么搜索不到指定的模块? .....	115
Q14: 在我的使用环境下, 为什么无线网络连接的质量总是很差? .....	115

## 附录B 文档修订记录 ..... 116

## 1 产品简介

HLK-WIFI-M03 是海凌科电子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，HLK-WIFI-M03 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。

HLK-WIFI-M03 在前两代产品的基础上进行了全面的软硬件升级，功能更加强大，使用更加简单，其主要功能特性包括：

### 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

### 无线

- ✧ 支持 IEEE802.11b/g 无线标准
- ✧ 支持频率范围：2.412~2.484 GHz
- ✧ 支持两种无线网络类型：
  - 基础网（Infra）和自组网（Adhoc）
- ✧ 支持软 AP，最多支持 4 个 station 连接
- ✧ 支持多种安全认证机制：

WEP64/WEP128/TKIP/CCMP(AES)

WEP/WPA-PSK/WPA2-PSK

AP 模式下仅支持 WEP64/WEP128

- ✧ 支持快速联网
- ✧ 支持无线漫游

### 其它

- ✧ 支持多种网络协议：

TCP/UDP/ICMP/DHCP/DNS/HTTP

- ✧ 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/无线连接

## 2 产品综述

### 2.1 技术规格

表 2-1 产品技术规格

	项 目	参 数
无线部分	无线标准	IEEE802.11b/g
	频率范围	2.412~2.484 GHz
	接收灵敏度	802.11b: -86 dBm @ 11Mbps (typical) 802.11g: -71 dBm @ 54Mbps (typical)
	数据速率	802.11b: 1, 2, 5.5, 11 Mbps 802.11g: 6, 9, 12, 18, 24, 36, 48, 54 Mbps
	调制方式	DSSS, OFDM, DBPSK, DQPSK, CCK, QAM16/64
	输出功率	802.11b: 18±2 dBm (typical) 802.11g: 15±1 dBm (typical)
	天线接口	IPX
硬件部分	接口类型	UART
	接口速率	1200~115200 bps
	工作电压	3.3±0.3 V
	工作电流	300mA (typical)
	存储温度	-40~+85 °C
	工作温度	-20°C~75
	外形尺寸	55.2×25.4×11.5 mm
	环保认证	RoHS
软件部分	网络类型	Infra/Adhoc/AP
	AP 模式	支持 4 个 STA 连接 仅支持 STA 和 AP 间通信, 不支持 STA 和 STA 间通信 加密认证方式支持 OPEN、WEP
	安全机制	WEP/WPA-PSK/WPA2-PSK
	加密类型	WEP64/WEP128/TKIP/CCMP (AES)
	工作模式	自动/命令

	串口命令	AT+指令集
	网络协议	TCP/UDP/ARP/ICMP/DHCP/DNS/HTTP
	最大 Socket 连接	15
	TCP 连接	最大连接数：8 最大 Client 数：8 最大 Server 数：3~4 本端 Server 最大接入 Client 数：4
	UDP 连接	最大连接数 5~6
	最高传输速率	11Kbytes (TCP)

## 2.2 硬件说明

### 2.2.1 机械尺寸

HLK-WIFI-M03 机械尺寸如下图所示，其中，图 2-1 为模块俯视图，图 2-2 为模块侧视图。



下图所示为 HLK-WIFI-M03 产品的外部接口示意图。

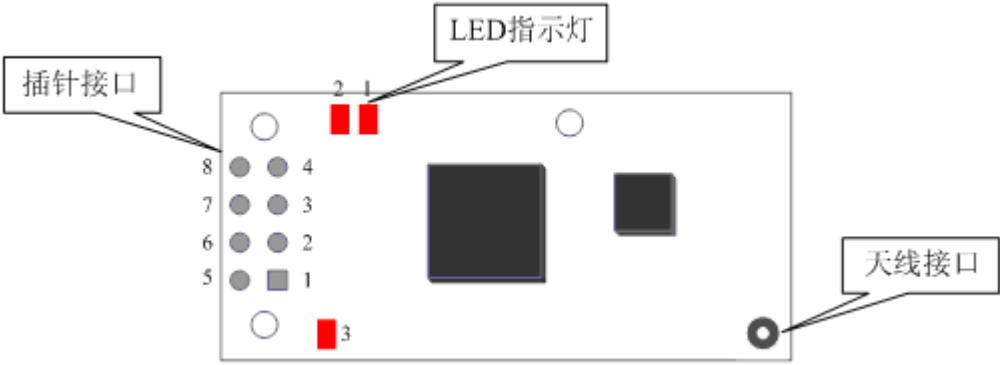


图 2-3 接口示意图

2.2.2.1 插针接口

本产品提供如上图所示的双列直插 8 针引脚，定义如下：

表 2-2 模块插针接口

号码	功 能	方 向	说 明
1	nRTS/nREADY	0	可选功能引脚，详细定义参见 2.3.6 节 同时连接至 LED1
2	nCTS/MODE/GPIO	I/O	可选功能引脚，详细定义参见 2.3.6 节 同时连接至 LED2
3	Link	0	可选功能引脚，无线传输指示 同时连接至 LED3
4	nRESET	I	可选功能引脚，复位，低电平有效
5	VDD	I	3.3V 电源输入引脚
6	RXD	I	串口数据接收
7	TXD	0	串口数据发送
8	GND	I	接地

注：

- 1、 外侧引脚（5~8）兼容 rs232m01/ttl001 单排插针接口）
- 2、 内侧引脚（1~4）为可选功能引脚，在不使用的情况下可以悬空

2.2.2.2 天线接口

本产品提供一个 IPX 天线接口。

2.2.3 天线（选配）



本产品必须连接符合 IEEE 802.11g / 802.11b 标准的 2.4G 天线，其参数要求如下表所示：

表 2-3 天线参数

号码	说 明
频率范围	2.4~2.4825 GHz
阻抗	50 Ω
电压驻波比	≦1.5

2.2.4 串口转接板（选配）

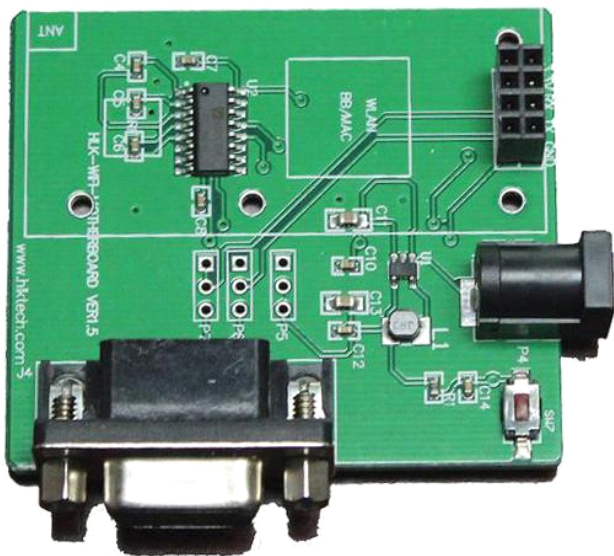


图 2-4 串口转接板

通过上图所示的转接板，可以将本产品与 PC 机的串口连接，从而使用我们提供的配置管理程序对产品的进行参数设置及功能测试。

➤ 接口说明

表 2-4 串口转接板接口说明

接 口	说 明
电源插口	接口，仅作为 5V 电源输入

DB9 接口	母型 9 针串口座，用于连接 PC 机串口
插针接口	2x4 插座，用于连接 WIFI 模块

➤ 跳线说明

表 2-5 串口转接板跳线说明

号码	说 明
1	无需跳线

注：连接 HLK-WIFI-M03 产品，使用图中所示默认选择即可

➤ LED 指示灯说明

表 2-6 串口转接板 LED 指示灯说明

号码	说 明
1	nRTS 信号指示，同模块 LED1
2	nCTS 信号指示，同模块 LED2
3	无线传输指示，同模块 LED3

➤ 按键说明

表 2-7 串口转接板按键说明

	说 明
复位按键	连接模块 nRESET 信号，用于复位模块

## 2.2.5 无线适配器（选配）



图 2-5 无线适配器

无线适配器，可以实现通过无线连接对本产品进行参数配置，这在无法使用插针接口连接 PC 串口的情况下尤其有用。另外，使用无线适配器还可以实现对本产品的参数进行批量修改的功能。使用方法详见第 3.2 节参数配置说明。

## 2.3 主要功能

### 2.3.1 连接无线网络

#### 2.3.1.1 网络拓扑

Wifi 无线网络包括两种类型的拓扑形式：基础网（Infra）和自组网（Adhoc），要说明无线网络的拓扑形式，需要首先了解两个基本概念：

AP，也就是无线接入点，是一个无线网络的创建者，是网络的中心节点。我们一般家庭或办公室使用的无线路由器就是一个 AP。

STA，站点，每一个连接到无线网络中的终端（如笔记本电脑、PDA 及其它可以联网的用户设备）都可称之为一个站点。

本模块作为一个 STA 接入到无线 AP 中，此外本模块亦可最为无线 AP 供 STA 的接入。

#### ✧ 基于 AP 组建的基础无线网络（Infra）

Infra：也称为基础网，是由 AP 创建，众多 STA 加入所组成的无线网络，这种类型的网络的特点是 AP 是整个网络的中心，网络中所有的通信都通过 AP 来转发完成。

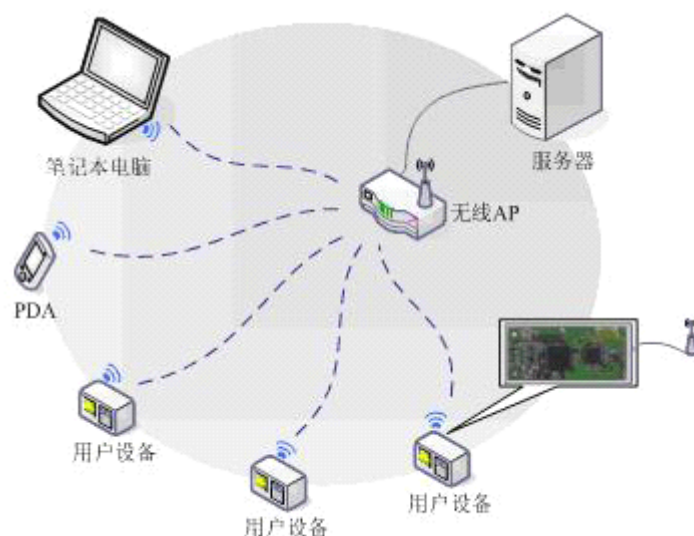


图 2-6 基础网络拓扑图

#### ✧ 基于自组网的无线网络（Adhoc）

Adhoc：也称为自组网，是仅由两个及以上 STA 自己组成，网络中不存在 AP，这种类型的网络是一种松散的结构，网络中所有的 STA 都可以直接通信。

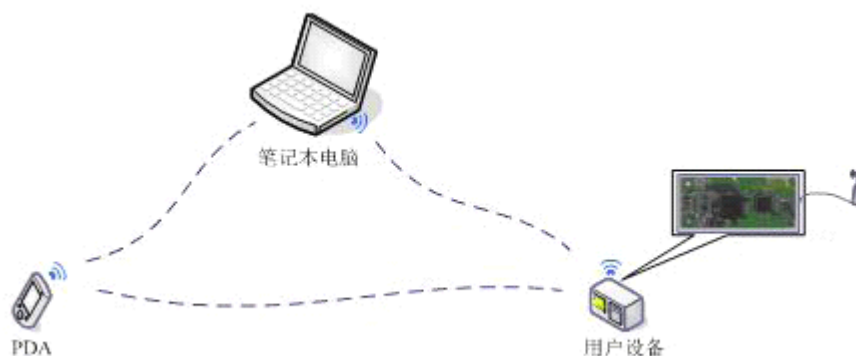


图 2-7 自组网络拓扑图

HLK-WIFI-M03 可以作为 STA 和 AP，支持上述两种类型的无线网络，但作为 **AP** 时仅支持 **STA** 和 **AP** 之间的数据通信，而不支持 **STA** 和 **STA** 之间的数据通信。

#### 2.3.1.2 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
- ✧ WEP128
- ✧ WPA-PSK/TKIP
- ✧ WPA-PSK/CCMP
- ✧ WPA2-PSK/TKIP
- ✧ WPA2-PSK/CCMP

#### 2.3.1.3 快速联网

本模块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 300 毫秒，从而达到加快联网速度的目的。

#### 2.3.1.4 地址绑定

本模块支持在联网过程中绑定目的网络 BSSID 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（也就是 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。

#### 2.3.1.5 无线漫游

本模块支持基于 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。

#### 2.3.1.6 区域设置

本模块支持无线的区域设置功能。用户可以通过设置无线信道列表参数来设定当前模块的扫描的信道范围。

#### 2.3.2 串口即插即用

本模块支持串口透明传输及自动工作模式，可真正实现串口即插即用，最大程度的降低用户使用的复杂度。在此模式下，用户只需预先设置好如下所示的联网必需的参数：

✧ 无线网络参数，包括

网络名称（Ssid）

安全模式

密钥（如果需要）

✧ 默认的 TCP/UDP 连接参数，包括

协议类型

连接类型

目的地址

目的端口

此后，模块每次上电后即可自动连接到预设的无线网络及服务器。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，按照使用普通串口的方式发送和接收数据就可以了。也就是说，用户原有的串口设备，直接把连接的串口线替换成本模块，用户设备无需作任何改动即可轻松实现数据的无线传输。

此工作模式下，模块支持用户可设置的自动组帧触发长度和触发的时间。自动组帧触发长度是指，模块从串口接收的指定长度的数据后，组成网络数据帧，并立即启动网络发送流

程。自动组帧触发时间是指，当从串口接收到的数据不足上述指定长度时，模块将在指定的超时时间后，强制将当前数据组成网络帧并启动发送。

此工作模式下，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。

### 2.3.3 串口命令模式

与自动工作模式不同，模块还提供了一种基于串口控制的命令工作模式，为用户提供了充分的灵活性，可以满足用户不同应用场合的特殊需求。

此工作模式下，模块根据用户通过串口下发的指令进行工作，用户可以通过指令对模块进行完全的控制，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，也是对用户来说最为灵活的使用方式。用户可以通过指令任意控制无线网络的连接、断开，也可以同时创建多个不同类型的 TCP/IP 连接，并保持通信。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（详见本文第六章），所有指令均基于 ASCII 编码，使用 Windows 超级终端程序即可直接对模块进行命令控制，方便用户调试和使用。

### 2.3.4 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。

- ✧ 支持最多 8 个 TCP Client 连接
- ✧ 支持最多 3 个 TCP Server 连接，每个 Server 最多接入 4 个 Client
- ✧ 支持最多 5~6 个 UDP 连接
- ✧ 支持 UDP 广播
- ✧ 最多 8 个 TCP 连接
- ✧ 支持 DHCP Client
- ✧ 支持 DNS Resolver
- ✧ 支持 HTTP，内置 WEB 服务器
- ✧ 支持 Socket 编程接口

### 2.3.5 灵活参数配置

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）

使用专用的无线适配器与模块通过无线连接，然后运行我们提供的配置管理应用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：一是随时，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；二是随地，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；三是批量配置，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

✧ 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

### 2.3.6 多功能GPIO

本模块提供两个可以复用的多功能引脚：

表 2-8 GPIO 复用功能

引 脚		用 途
0	nRTS	输出，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	输出，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，高电平未连接
1	nCTS	输入，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
	MODE	输入，模块的启动模式选择，在模块启动阶段（上电/复位 300ms 以内），低电平表示进入正常工作模式，高电平表示进入配置模式（详见 4.2.1 节）
	GPIO	输入/输出，根据用户设置可以作为通用输入、输出管脚

### 2.3.7 固件在线升级

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 快速开始向导

### 3.1 准备工作


#### ■ 硬件

- ✧ HLK-WIFI-M03 模块
- ✧ 串口转接板
- ✧ 串口线

#### ■ 软件

- ✧ UART-WIFI 配置管理程序



- ✧ 串口调试软件，如串口调试助手、串口大师（ ComMaster.exe）等

- ✧ TCP/UDP 调试软件，如 TCP/UDP 测试工具（ TCPUDPDbg.exe）等

### 3.2 设置参数



- 1、启动 UART-WIFI 配置管理程序；
- 2、将 HLK-WIFI-M03 模块连接串口转接板，连接好串口线后打开电源，如下图所示：

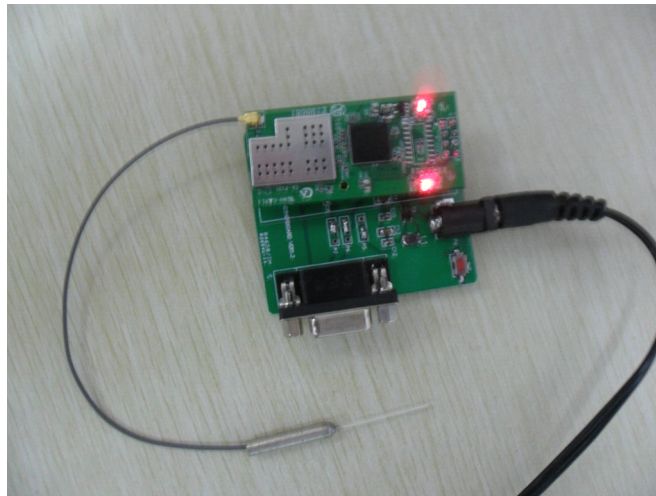


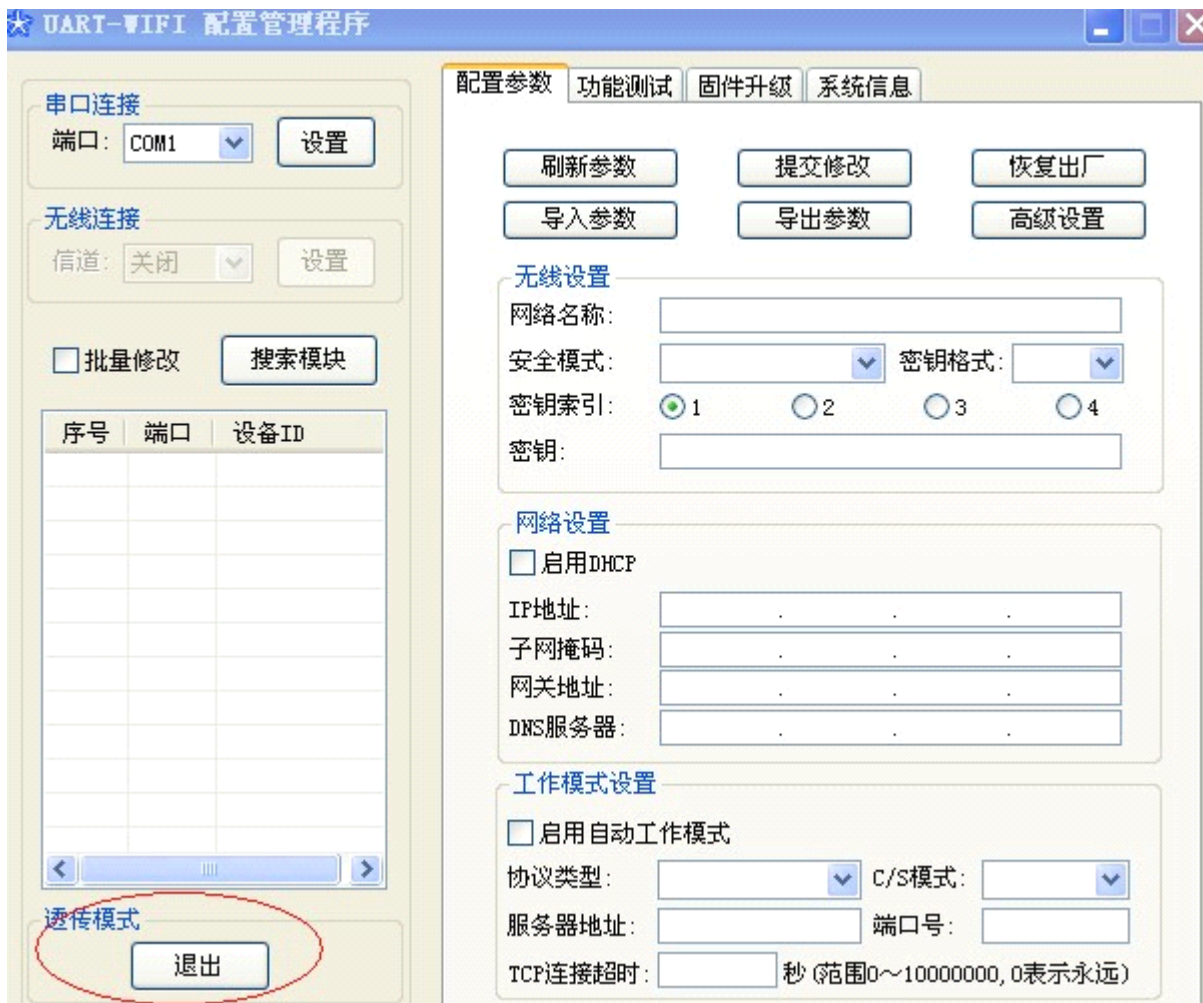
图 3-1 串口转接板连接图

直连串口线连接串口转接板和电脑串口



3、

4、点击“透传模式” “退出” 按钮！退出成功后



5、点击“搜索模块”，成功后如下图所示：



图 3-2 搜索模块成功示意图

- 6、根据您的所在无线网络的无线路由器（AP）设置参数修改无线设置中的参数，包括网络名称、加密方式、密钥等），网络设置中的 IP 地址相关参数；
- 7、修改工作模式设置，选中“启用自动工作模式”，并设置您需要自动创建的连接的参数；
- 8、参数修改完成后如下图所示，点击“提交修改”，并在弹出的对话框中选择“稍后手动复位”；



图 3-3 修改参数示意图

### 3.3 开始测试

- 1、启动 TCP/UDP 调试软件，本例中使用 TCP/UDP 测试工具（TCPUDPDbg.exe），创建一个创建了监听端口为 60000 的 TCP Server，如下图所示；

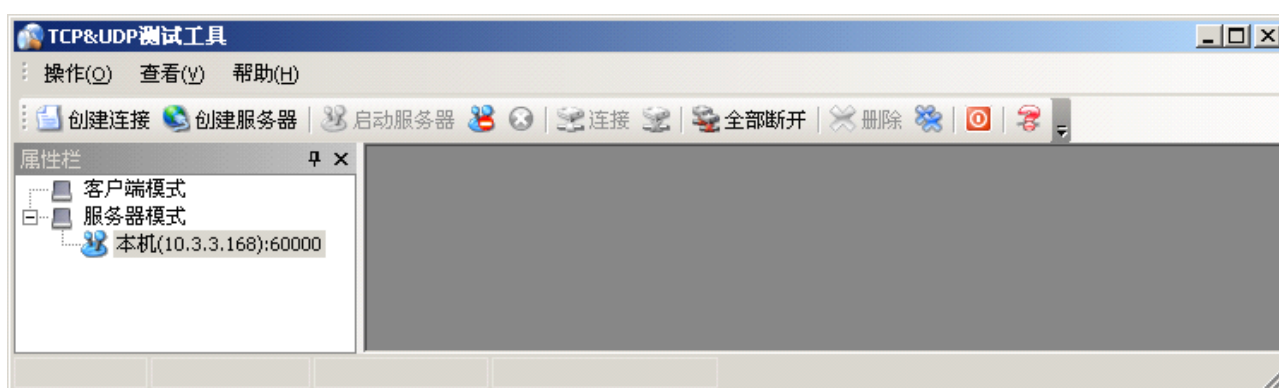


图 3-4 TCP 服务器示意图



- 2、启动串口调试软件，本例中使用串口大师（ComMaster.exe），按照下图所示进行设置，完成后打开串口（注意：串口大师软件高级选项中的流量控制选择“硬件 RTS/CTS 流控”后，按 OK 确认即设置成功。再次打开高级选项时流量控制可能依然显示为“NONE”，不用管它）；

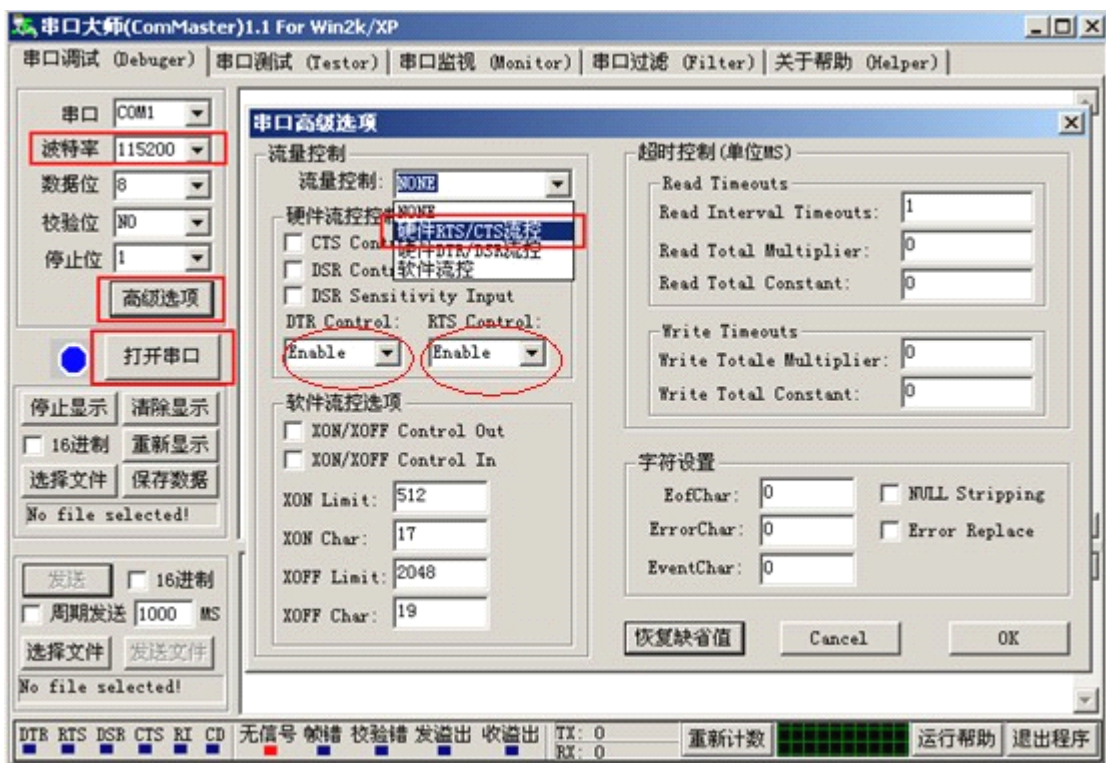


图 3-5 串口大师设置示意图

- 3、将串口转接板重新上电（或按下复位按键），HLK-WIFI-M03 模块复位并开始自动连接 AP 与 TCP 服务器，连接成功后可以通过串口大师与 TCP/UDP 测试工具进行传输测试，如下图所示；

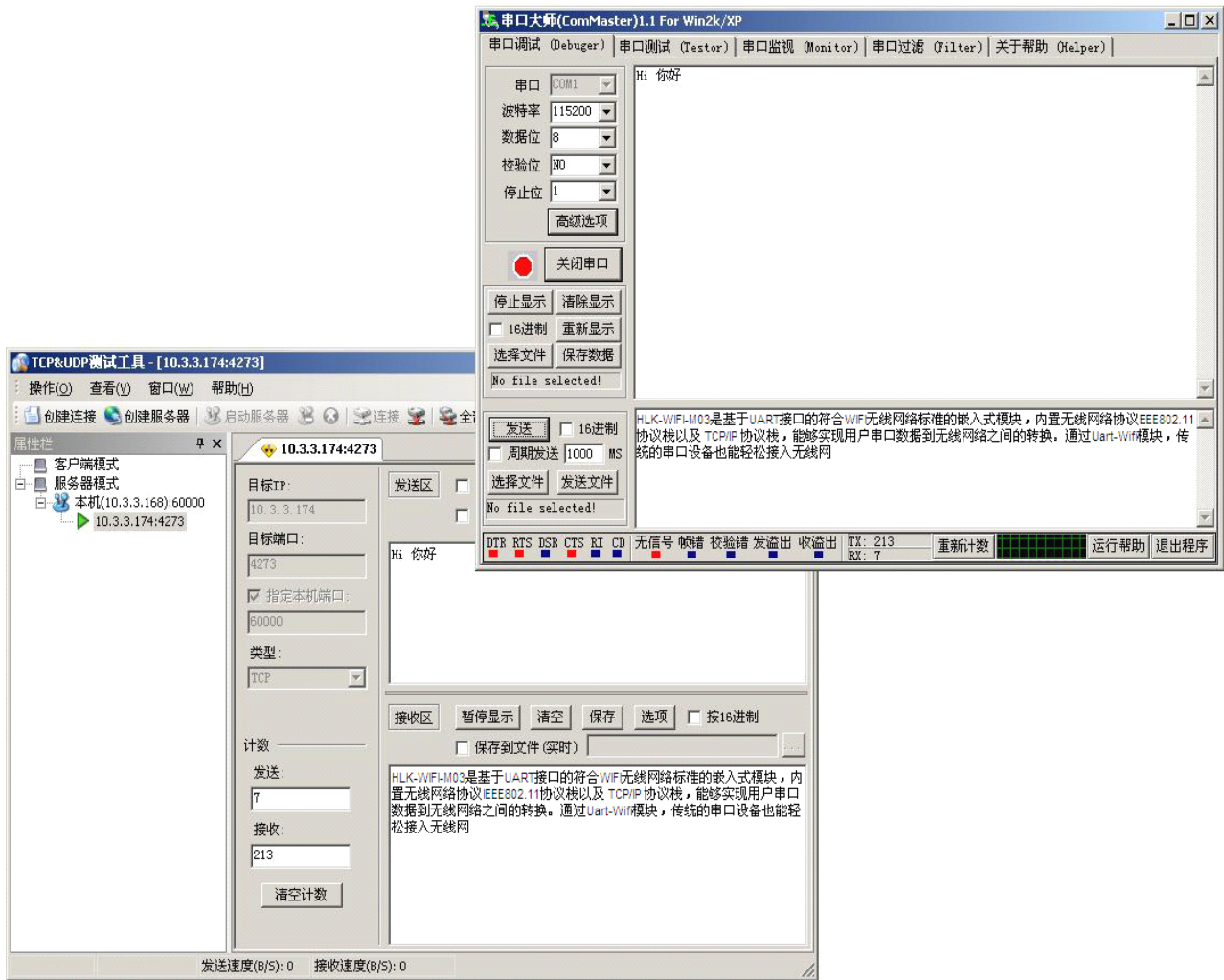


图 3-6 传输测试示意图



## 4 产品设计指南

### 4.1 硬件连接

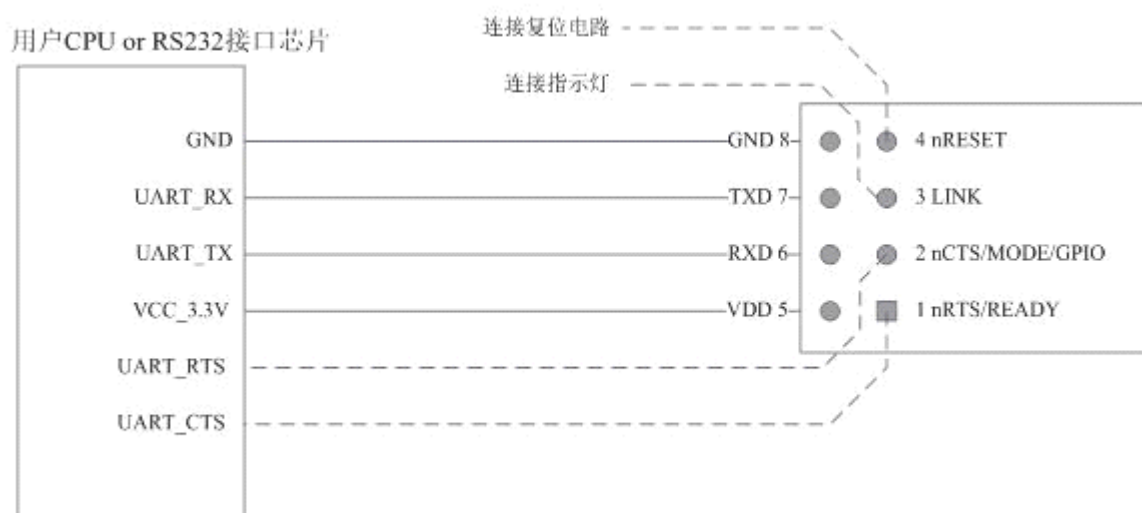


图 4-1 硬件连接图

HLK-WIFI-M03 模块提供双列直插 8 针引脚，其中外侧一排（5~8）引脚为必须连接，如上图所示，而且这些引脚完全兼容 HLK-WIFI-RS232M01/TTL001 单排 4 针接口。内侧一排（1~4）为可选功能引脚，连接如上图所示。如果不使用这些可选功能，只需将对应的引脚悬空即可。

### 4.2 工作模式

#### 4.2.1 启动模式选择

复用功能引脚 nCTS/MODE/GPIO（引脚 2）在模块上电复位阶段用来进行启动模式选择，时序如下图所示。

✧ 正常启动模式

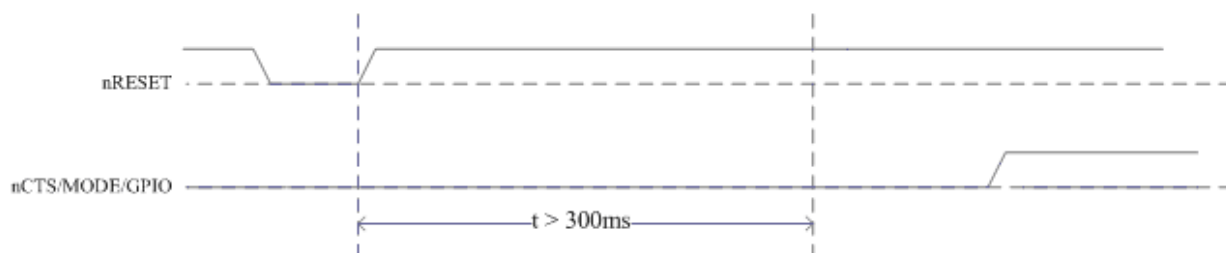


图 4-2 正常启动模式

需要说明的是，模块的 **nCTS/MODE/GPIO** 引脚进行了内部下拉，当用户将其悬空，模块即可自动进入正常启动模式。

在正常启动模式下，系统根据配置参数中预设的工作模式（自动/命令模式）运行。

#### ◇ 启动配置模式

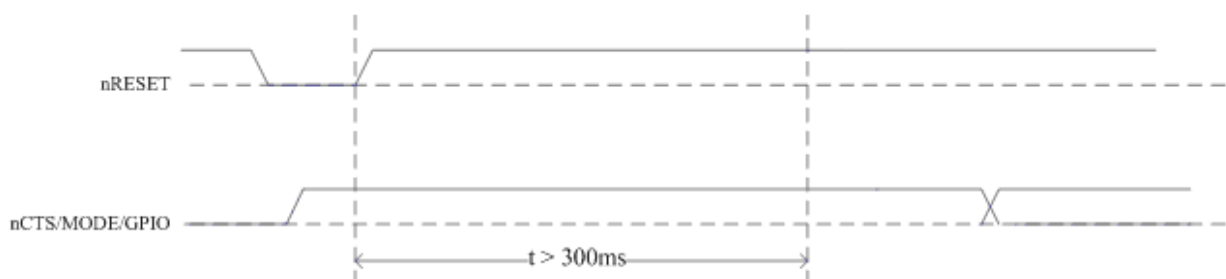


图 4-3 启动配置模式

在启动配置模式下，系统忽略配置参数中预设的工作模式参数，强制进入 AT+命令模式运行，此模式通常用于使用配置管理程序进行参数修改及功能测试。

### 4.2.2 自动工作模式

本模块支持自动工作模式，可真正实现串口即插即用，最大程度的降低用户使用的复杂度。在此模式下，用户只需预先设置好必需的参数，模块每次上电后即可自动连接到预设的无线网络及服务器。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，按照使用普通串口的方式发送和接收数据就可以了。也就是说，用户原有的串口设备，直接把连接的串口线替换成本模块，用户设备无需作任何改动即可轻松实现数据的无线传输。

#### 4.2.2.1 自动重试机制

在自动工作模式下，系统的连接网络完全自动运行，其工作流程如下图所示：

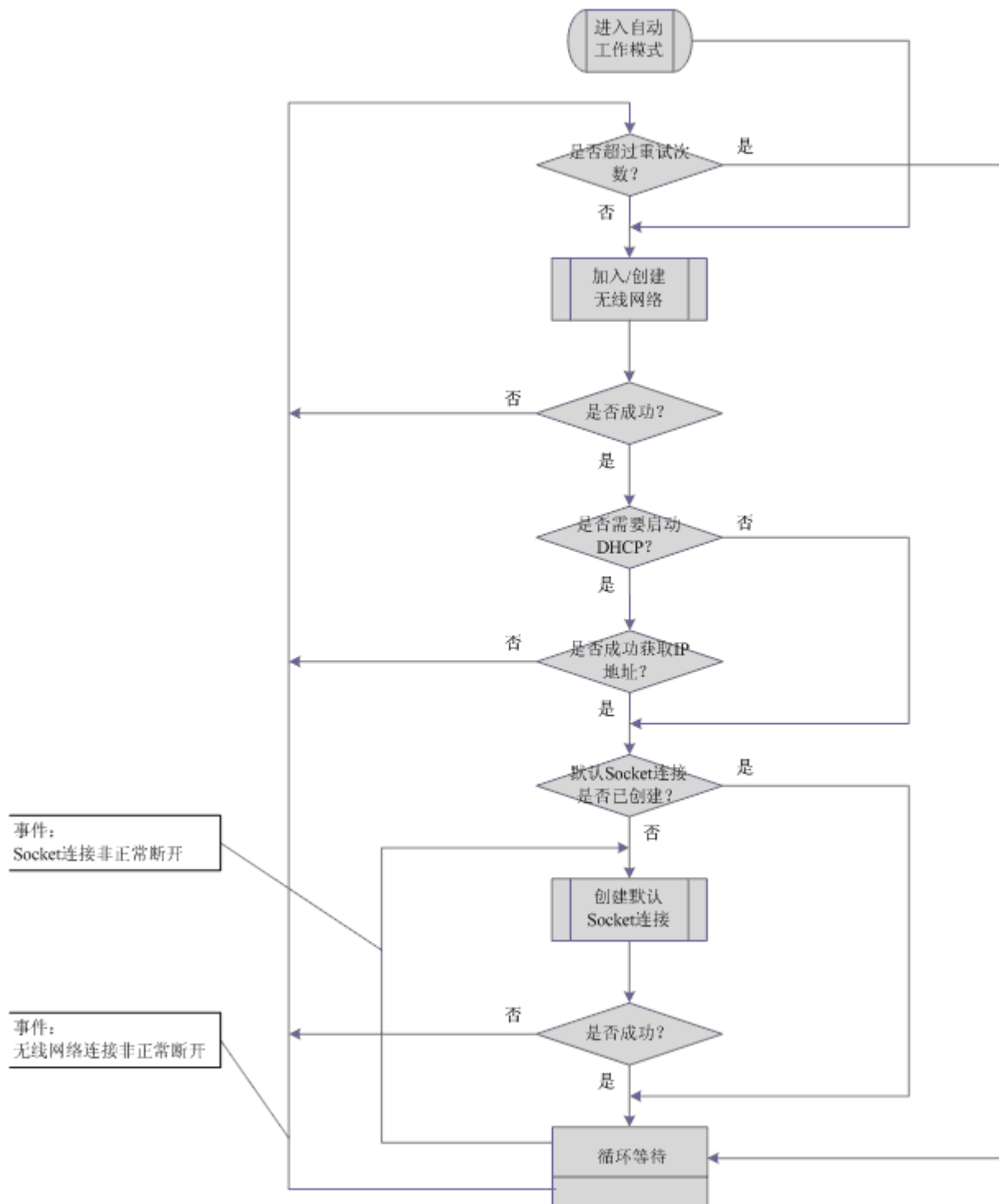


图 4-4 自动重试流程

说明：

- 1、系统默认自动重试次数设置为永远，也就是说，系统会在连接网络失败或网络断开的时候一直重新尝试连接。用户也可以将重试次数设置为有限值，此种情况下，一旦重试次数超限后，系统将不再尝试重连，只有系统复位后才能再次重启自动连接流程；
- 2、Socket 连接（此处特指 Tcp 连接）是虚连接，在无线网络连接非正常断开的情况下原有的 Socket 连接不会被删除，并且在无线网络连接恢复后继续使用；

#### 4.2.2.2 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。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。

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。

#### 4.2.2.3 硬件流控机制

模块在串口透明传输状态下，支持基于 RTS/CTS 的硬件流控机制。

- ✧ 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
  - 无线网络已连接
  - 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

#### 4.2.2.4 退出透明传输

在默认配置的自动工作模式下，模块的串口处于透明传输状态，此时，所有通过串口接收到的数据都转发到网络上，同样，从网络上接收到的数据，模块也都原样从发送到串口上。

这是一种单纯的数据传输状态，此时，如果要通过串口进行控制命令的传输，需要首先退出透明传输状态。同时，模块也退出自动工作模式。

模块退出透明传输状态的条件如下图所示。

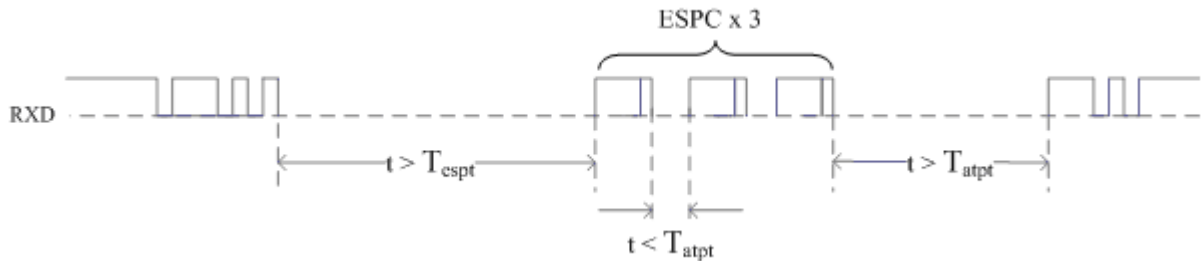


图 4-5 退出透明传输状态示意图

其中，

ESPC：逃逸字符

$T_{espt}$ ：逃逸时间

$T_{atpt}$ ：自动组帧周期

模块检测到符合上图所示的退出条件后，输出“+OK”，确认串口退出透明传输切换到 AT+命令状态。

#### 4.2.3 命令工作模式

模块还提供了一种基于串口控制的命令工作模式，用户不同应用场合的特殊需求。

此工作模式下，模块根据用户通过串口下发的指令进行工作，用户可以通过指令对模块进行完全的控制，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，也是对用户来说最为灵活的使用方式。用户可以通过指令任意控制无线网络的连接、断开，也可以同时创建多个不同类型的 TCP/IP 连接，并保持通信。

要使用此模式，请参考第 6 章 AT+指令控制协议。

### 4.3 参数详细说明

#### 4.3.1 无线设置参数

##### 4.3.1.1 网络名称

表 4-1 网络名称

参数名	Parameter Name	相关指令
网络名称	SSID	AT+SSID
说 明		
Wifi 网络的网络标识，长度为 1~32 个 ASCII 字符		

--

#### 4.3.1.2 安全模式

表 4-2 安全模式

参数名	Parameter Name	相关指令
安全模式	Encryption	AT+ENCRY
说 明		
<p>Wifi 网络的加密类型，可选项包括：</p> <ul style="list-style-type: none"><li>✧ OPEN</li><li>✧ WEP64</li><li>✧ WEP128</li><li>✧ WPA-PSK(TKIP)</li><li>✧ WPA-PSK(CCMP)</li><li>✧ WPA2-PSK(TKIP)</li><li>✧ WPA2-PSK(TKIP)</li><li>✧ WAPI（暂不支持）</li></ul>		

#### 4.3.1.3 密钥格式

表 4-3 密钥格式

参数名	Parameter Name	相关指令
密钥格式	Key Format	AT+KEY
说 明		
<p>Wifi 网络的密钥格式，可选项包括：</p> <ul style="list-style-type: none"><li>✧ HEX 格式</li><li>✧ ASCII 格式</li></ul> <p>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）。</p> <p>而对于 WPA/WPA2-PSK 方式，HEX 与 ASCII 方式则表示不同的含义，</p> <p>HEX 格式：表示 PMK</p> <p>ASCII 格式：表示 PSK</p>		

## 4.3.1.4 密钥索引

表 4-4 密钥索引

参数名	Parameter Name	相关指令
密钥索引	Key Index	AT+KEY
说 明		
Wifi 网络的密钥索引号，本参数仅在 WEP 加密方式下有效，可选范围为 1~4，其它加密方式下无意义		

## 4.3.1.5 密钥

表 4-5 密钥

参数名	Parameter Name	相关指令
密 钥	Encryption Key	AT+KEY
说 明		
Wifi 网络的密钥字符串，不同的加密方式及密钥格式，密钥有不同的长度要求： ✧ WEP64（十六进制）：10 个字符 ✧ WEP64（ASCII）：5 个字符 ✧ WEP128（十六进制）：26 个字符 ✧ WEP128（ASCII）：13 个字符 ✧ WPA/WPA2-PSK（十六进制）：64 个字符 ✧ WPA/WPA2-PSK（ASCII）：8~63 个字符		
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，可以使用本产品附带的 WPA 密钥转换工具（ <small>WPA Key Converter.exe</small> ）将 PSK 预先转换成 PMK，并设置到模块中。		

## 4.3.2 网络设置参数

### 4.3.2.1 DHCP

表 4-6 DHCP

参数名	Parameter Name	相关指令
DHCP	DHCP Enable	AT+NIP
说 明		
用于指定本端的网络地址类型，使能本参数后，模块将通过 DHCP 协议动态获取本端的网络地址。		

### 4.3.2.2 IP地址

表 4-7 IP 地址

参数名	Parameter Name	相关指令
IP 地址	Fixed IP Address	AT+NIP
说 明		
用于指定本端网络地址中的静态 IP 地址，仅在关闭自动获取网络地址的情况下有效。		

### 4.3.2.3 子网掩码

表 4-8 子网掩码

参数名	Parameter Name	相关指令
子网掩码	Subnet Mask	AT+NIP
说 明		
用于指定本端网络地址中的子网掩码，仅在关闭自动获取网络地址的情况下有效。		

### 4.3.2.4 网关地址

表 4-9 网关地址

参数名	Parameter Name	相关指令
网关地址	Gateway Address	AT+NIP
说 明		
用于指定本端网络地址中的网关地址，仅在关闭自动获取网络地址的情况下有效。		



### 4.3.2.5 DNS服务器

表 4-10 DNS 服务器

参数名	Parameter Name	相关指令
DNS 服务器	DNS Address	AT+NIP
说 明		
用于指定本端网络地址中的 DNS 服务器地址，仅在关闭自动获取网络地址的情况下有效。		

### 4.3.3 工作模式设置参数

#### 4.3.3.1 自动工作模式

表 4-11 自动工作模式

参数名	Parameter Name	相关指令
自动工作模式	Auto Mode Enable	AT+ATM
说 明		
用于指定模块是否启用自动工作模式		

#### 4.3.3.2 协议类型

表 4-12 协议类型

参数名	Parameter Name	相关指令
协议类型	Protocol	AT+ATRM
说 明		
用于指定自动工作模式下，模块创建的默认连接的协议类型，可选项包括： ✧ TCP ✧ UDP		

#### 4.3.3.3 CS模式

表 4-13 CS 模式

参数名	Parameter Name	相关指令
CS 模式	CS Mode	AT+ATRM
说 明		

用于指定自动工作模式下，模块创建的默认连接的 CS 模式，可选项包括：

- ✧ 客户端（Client）
- ✧ 服务器端（Server）

#### 4.3.3.4 服务器地址

表 4-14 服务器地址

参数名	Parameter Name	相关指令
服务器地址	Server Address	AT+ATRM
说 明		
用于指定自动工作模式下，当模块创建的默认连接 CS 模式为 Client 时，所要连接的服务器，输入格式可以是 IP 地址或域名。		

#### 4.3.3.5 TCP连接超时

表 4-15 TCP 连接超时

参数名	Parameter Name	相关指令
TCP 连接超时	Tcp Link TimeOut	AT+ATRM
说 明		
用于指定自动工作模式下，当模块创建的默认连接为 TCP Server 时，接入本服务器的 Client 在没有数据传输的情况下最大的连接时间，超过此时间将被自动踢掉。		

#### 4.3.3.6 端口号

表 4-16 端口号

参数名	Parameter Name	相关指令
端口号	Port Number	AT+ATRM
说 明		
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。		

### 4.3.4 高级无线设置参数

#### 4.3.4.1 网络类型

表 4-17 网络类型

参数名	Parameter Name	相关指令
网络类型	Network Mode	AT+WPRT
说 明		
<p>要加入 Wifi 网络的网络类型，可选项包括：</p> <ul style="list-style-type: none"> <li>✧ 基础网（Infra）</li> <li>✧ 自组网（Adhoc）</li> </ul> <p>系统缺省设置为 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 类型的网络时，可以修改此参数为 Adhoc。</p>		

## 4.3.4.2 网络不存在时自动创建

表 4-18 网络不存在时自动创建

参数名	Parameter Name	相关指令
网络不存在时自动创建	Create adhoc When network does not exist	AT+WATC
说 明		
<p>当 Wifi 网络类型为 Adhoc 时，指定如果网络不存在是否自动创建该网络。</p> <p>系统缺省设置为不启用。本参数仅当网络类型设置为 Adhoc 时才有效，如果用户需要创建一个 Adhoc 网络，可以设置本参数为启用。</p>		

## 4.3.4.3 BG模式

表 4-19 BG 模式

参数名	Parameter Name	相关指令
BG 模式	Wireless Standard	AT+WBGR
说 明		
<p>要加入 Wifi 网络的 BG 模式，可选项包括：</p> <ul style="list-style-type: none"> <li>✧ B/G 混合模式</li> <li>✧ B 模式</li> </ul> <p>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。</p>		

## 4.3.4.4 最高速率

表 4-20 最高速率

参数名	Parameter Name	相关指令
最高速率	Data Rate	AT+WBGR

说 明
<p>模块的无线最高发射速率，可选项包括：</p> <p>1M、2M、5.5M、11M、6M、9M、12M、18M、24M、36M、48M、54M</p> <p>当 BG 模式为 B 模式时，可选速率仅包含前 4 项。</p> <p>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。</p>

#### 4.3.4.5 BSSID

表 4-21 BSSID

参数名	Parameter Name	相关指令
BSSID	Ap BSSID	AT+BSSID
说 明		
<p>要加入的 Wifi 网络的 BSSID 地址设置，可选项包括：</p> <ul style="list-style-type: none"> <li>✧ 自动</li> <li>✧ 指定，需要输入 BSSID 地址</li> </ul> <p>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。</p>		

#### 4.3.4.6 信道

表 4-22 信道

参数名	Parameter Name	相关指令
信道	Channel Number	AT+CHL
说 明		
<p>要加入的 Wifi 网络的信道设置，可选项包括：</p> <ul style="list-style-type: none"> <li>✧ 自动</li> <li>✧ 信道 1~14</li> </ul> <p>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</p>		

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。

#### 4.3.4.7 自动重试

表 4-23 自动重试

参数名	Parameter Name	相关指令
自动重试	Auto Retry Count	AT+CHL
说 明		
本参数仅在自动工作模式下有效，表示的含义为当联网失败或在已经联网状态下网络非正常断开时，系统重试次数，可选项包括： ✧ 从不 ✧ 永远 ✧ 1~254 系统缺省设置为永远重试。		

#### 4.3.4.8 无线漫游

表 4-24 无线漫游

参数名	Parameter Name	相关指令
无线漫游	Wireless Roam	AT+WARM
说 明		
本参数用于指定是否启用无线漫游功能。 系统缺省设置为不启用。需要说明的是，当 BSSID 参数设置为指定时，无线漫游功能将失效。		

#### 4.3.4.9 信道列表

表 4-25 信道列表

参数名	Parameter Name	相关指令
信道列表	Channel List	AT+CHLL
说 明		
本参数用于指定无线区域设置，不同的国家和地区对 Wifi 的 2.4G 频段开放的范围不同，如美国为 1~11 信道，欧洲为 1~13 信道，日本则为 1~14 信道。 系统缺省设置为 1~14 所有信道。通常情况下用户无需修改此参数。本参数的另一个用途是，通过设定模块的扫描时的信道范围，来缩短信道扫描及联网时间。		

如，在已知我们所要连接的 AP 可能的工作信道为 1、6、11 的情况下，可以在信道列表中仅设置 1、6、11，这样，模块在联网时将只扫描这三个信道，从而缩短联网时间的目的。

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。

#### 4.3.5 串口设置参数

##### 4.3.5.1 波特率

表 4-26 波特率

参数名	Parameter Name	相关指令
波特率	Baud Rate	AT+UART
说 明		
串口波特率设置，可选项范围： ✧ 1200~115200bps 系统缺省设置为 115200bps。		

##### 4.3.5.2 数据位

表 4-27 数据位

参数名	Parameter Name	相关指令
数据位	Data Bits	AT+UART
说 明		
串口数据位长度设置，可选项包括： ✧ 8 位 ✧ 7 位 系统缺省设置为 8 位数据位。		

##### 4.3.5.3 停止位

表 4-28 停止位

参数名	Parameter Name	相关指令
停止位	Stop Bits	AT+UART
说 明		
串口停止位长度设置，可选项包括： ✧ 1 位 ✧ 2 位		

系统缺省设置为 1 位停止位。

#### 4.3.5.4 校验位

表 4-29 校验位

参数名	Parameter Name	相关指令
校验位	Parity	AT+UART
说 明		
串口校验位设置，可选项范围： ✧ 无校验 ✧ 奇校验 ✧ 偶校验 系统缺省设置为无校验。		

#### 4.3.6 透明模式设置参数

##### 4.3.6.1 自动组帧长度

表 4-30 自动组帧长度

参数名	Parameter Name	相关指令
自动组帧长度	Data Trigger Length	AT+ATLT
说 明		
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 64~1024。 系统缺省设置为 512 个字节。		

##### 4.3.6.2 自动组帧周期

表 4-31 自动组帧周期

参数名	Parameter Name	相关指令
自动组帧周期	Data Trigger Timeout	AT+ATPT
说 明		
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 100~10000ms，最小精度 100ms。此外，本设置必须小于逃逸时间设置。 系统缺省设置为 500ms。		

### 4.3.6.3 逃逸字符

表 4-32 逃逸字符

参数名	Parameter Name	相关指令
逃逸字符	Escape Character	AT+ESPC
说 明		
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"+”。		

### 4.3.6.4 逃逸时间

表 4-33 逃逸时间

参数名	Parameter Name	相关指令
逃逸时间	Escape Trigger Period	AT+ESPT
说 明		
串口在透明传输模式下,当超过本参数指定的时间没有接收到任何数据时,将自动开始逃逸字符检测,可选范围为 100~10000ms,最小精度 100ms。此外,本设置必须大于自动组帧周期设置。 系统缺省设置为 2000ms。		

## 4.3.7 系统设置参数

### 4.3.7.1 系统密码

表 4-34 系统密码

参数名	Parameter Name	相关指令
系统密码	Login Password	AT+PASS
说 明		
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”。		

### 4.3.7.2 WEB服务器

表 4-35 WEB 服务器

参数名	Parameter Name	相关指令
WEB 服务器	Web Server	AT+WEBS
说 明		
用于指定是否启用内部 Web 服务器及服务器端口		



系统缺省设置为启用，端口 80。

#### 4.3.7.3 命令模式

表 4-36 命令模式

参数名	Parameter Name	相关指令
命令模式	Command Mode	AT+CMDM
说 明		
用于指定模块在命令模式下的控制协议，可选项包括： <ul style="list-style-type: none"> <li>✧ AT+指令</li> <li>✧ 兼容协议</li> </ul> 系统缺省设置为 AT+指令模式，用户通常无需修改本参数。兼容协议命令模式用于兼容 TLG09UA01/02 产品的控制协议，要使用该模式请联系技术支持获取更多帮助。		

#### 4.3.7.4 GPIO1 模式

表 4-37 GPIO1 模式

参数名	Parameter Name	相关指令
GPIO1 模式	GPIO Mode	AT+IOM
说 明		
用于指定 GPIO1 的工作模式，可选项包括： <ul style="list-style-type: none"> <li>✧ 系统功能</li> <li>✧ 输入</li> <li>✧ 输出</li> </ul> 系统缺省设置为系统功能。		

### 4.4 出厂默认设置

表 4-38 出厂默认设置

参数名	参数值	说 明
无线设置参数		
网络名称	"default"	无线网络 ssid 为 default
安全模式	OPEN	无线网络安全模式为开放网络
密钥格式	-	

密钥索引	—	
密钥	—	
网络设置参数		
DHCP	Enable	使能 DHCP 协议动态获取 IP 地址
IP 地址	—	
子网掩码	—	
网关地址	—	
DNS 服务器	—	
工作模式设置参数		
自动工作模式	Enable	使能自动工作模式
协议类型	TCP	默认连接： TCP 服务器，连接超时 120 秒
CS 模式	Server	
服务器地址	—	
TCP 连接超时	120	
高级无线设置参数		
网络类型	Infra	无线网络类型为基础网
网络不存在时 自动创建	—	—
BG 模式	B/G	无线网络模式为 BG 混合
最高速率	36M	无线最高发射速率 36M
BSSID	Auto	禁止绑定无线网络的 BSSID 地址
信道	Auto	自动检测目的网络的工作信道
自动重试	255	永远自动重试
无线漫游	Disable	禁止无线网络漫游
信道列表	0x3fff	1~14 所有信道
串口设置参数		
波特率	115200	串口数据格式 波特率 115200bps 8 位数据位/1 位停止位/无校验
数据位	8 位	
停止位	1 位	
校验位	None	
透明模式设置参数		
自动组帧长度	512	自动组帧长度为 512 字节
自动组帧周期	500ms	自动组帧周期 500ms
逃逸字符	0x2B	逃逸字符为 0x2B，即 ASCII 字符“+”

逃逸时间	2000ms	逃逸时间 2000ms
系统设置参数		
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 80
命令模式	AT+	系统命令模式为使用 AT+指令集
GPI01 模式	System	GPI01 设置为系统功能

## 4.5 典型应用范例

### 4.5.1 主动型串口设备联网

#### 4.5.1.1 应用场景

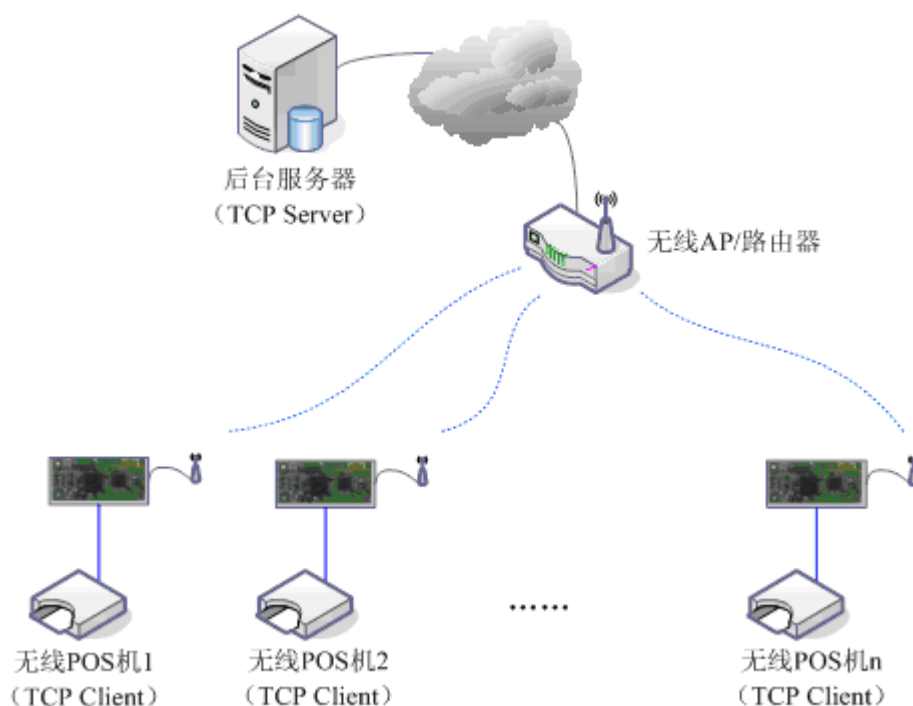


图 4-6 主动型串口设备联网

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上图所示。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。

#### 4.5.1.2 参考设置

## 无线 POS 机 X

### ✧ 无线设置

- 网络名称、加密类型、密钥等：参见无线 AP/路由器设置

### ✧ 网络设置

- 启用自动获得网络地址

### ✧ 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，如 192.168.0.100
- 端口号：后台服务器端口号，如 6000

## 4.5.2 被动型串口设备联网

### 4.5.2.1 应用场景

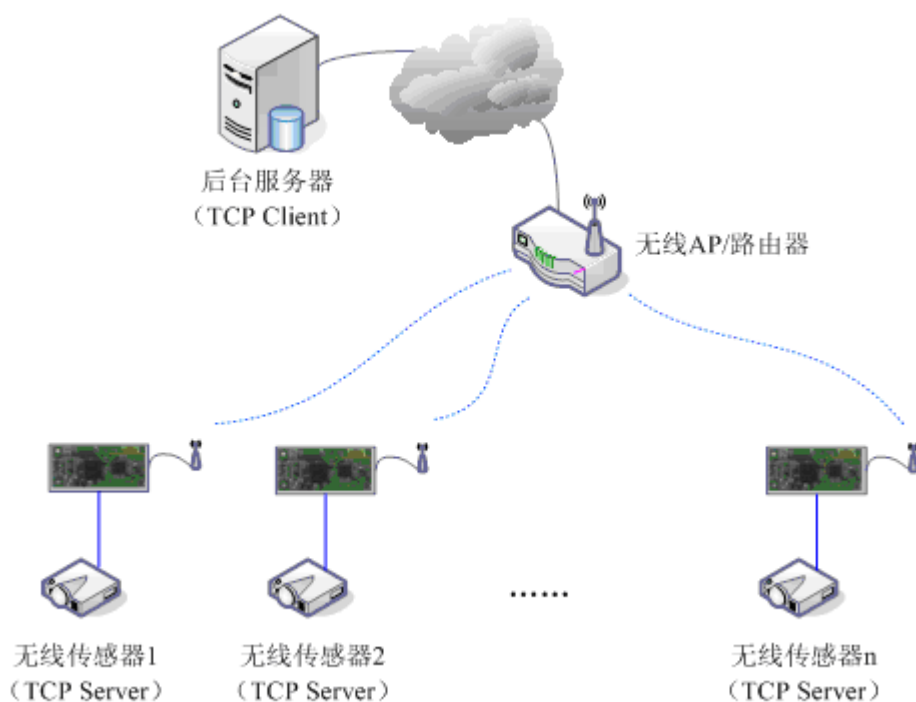


图 4-7 被动型串口设备联网

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，并进行请求或上传数据的方式。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。

#### 4.5.2.2 参考设置

##### 无线传感器 X

###### ✧ 无线设置

- 网络名称、加密类型、密钥等：参见无线 AP/路由器设置

###### ✧ 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

###### ✧ 工作模式设置

- 启用自动工作模式
- 协议类型：TCP
- C/S 模式：服务器
- 端口号：6000
- 连接超时：120

#### 4.5.3 广播方式的串口设备联网

##### 4.5.3.1 应用场景

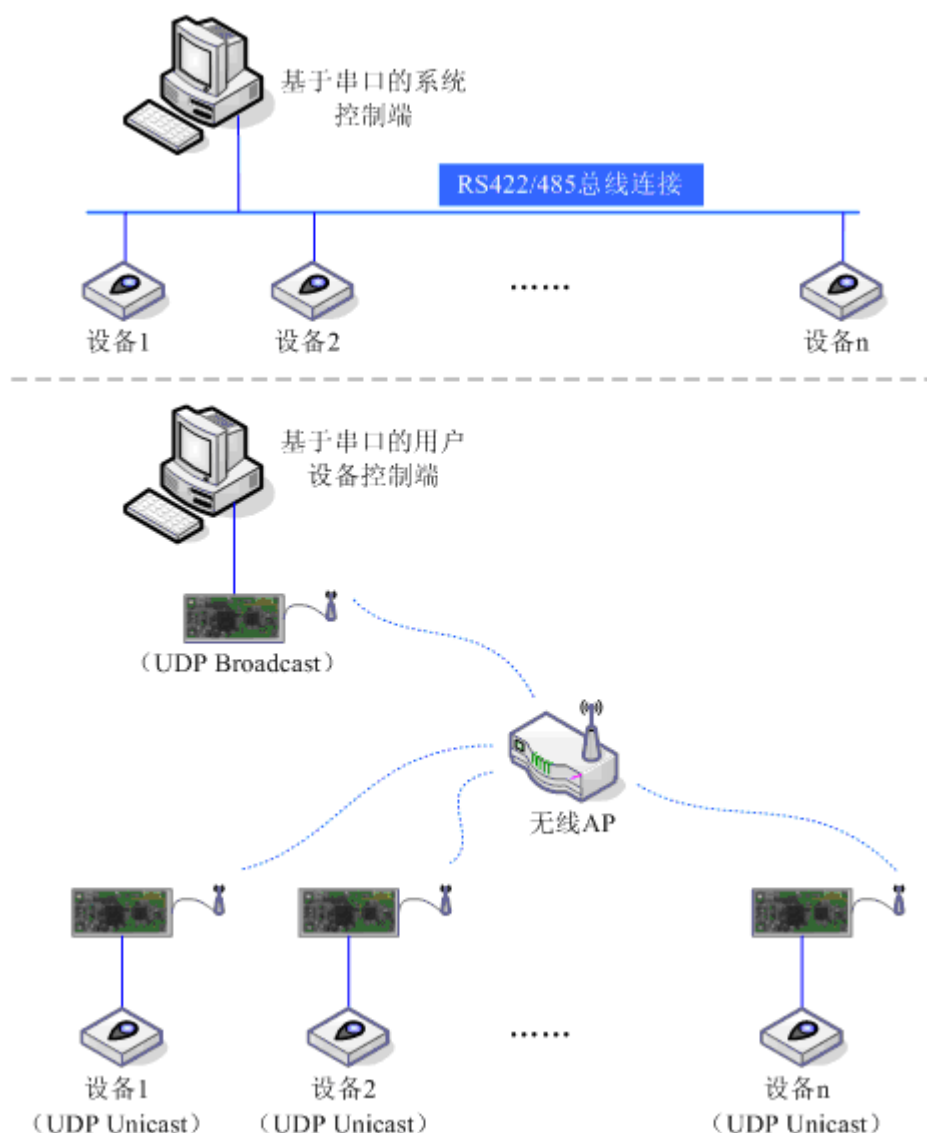


图 4-8 广播方式的串口设备联网

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”。

#### 4.5.3.2 参考设置

##### 设备控制端

##### ◇ 无线设置

- 网络名称、加密类型、密钥等：参见无线 AP/路由器设置

✧ 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

✧ 工作模式设置

- 启用自动工作模式
- 协议类型：UDP
- C/S 模式：广播
- 端口号：6000
- 连接超时：0

## 设备 X

✧ 无线设置

- 网络名称、加密类型、密钥等：参见无线 AP/路由器设置

✧ 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

✧ 工作模式设置

- 启用自动工作模式
- 协议类型：UDP
- C/S 模式：单播
- 端口号：6000
- 连接超时：0

### 4.5.4 无线虚拟串口

#### 4.5.4.1 应用场景

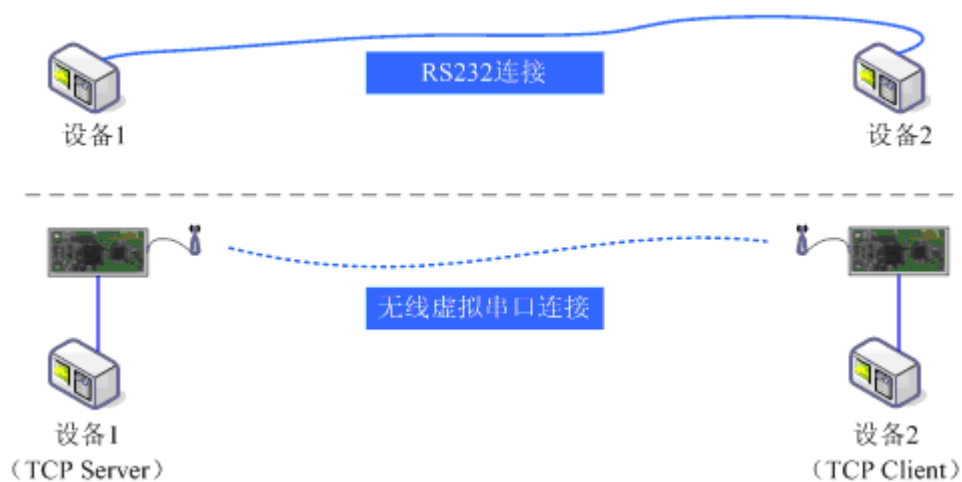


图 4-9 无线虚拟串口

#### 4.5.4.2 参考设置

##### 设备 1

##### ✧ 无线设置

- 网络类型：自组网络，使能“网络不存在时自动创建”
- 网络名称：my\_adhoc
- 加密类型：开放
- 其余默认

##### ✧ 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

##### ✧ 工作模式设置

- 启用自动工作模式
- 协议类型：TCP
- C/S 模式：服务器
- 端口号：6000
- 连接超时：0

##### 设备 2

##### ✧ 无线设置



- 网络类型：自组网络
- 网络名称：my\_adhoc
- 加密类型：开放
- 其余默认

✧ 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

✧ 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：192.168.0.1
- 端口号：6000

### 4.5.5 多网络连接模式

#### 4.5.5.1 应用场景

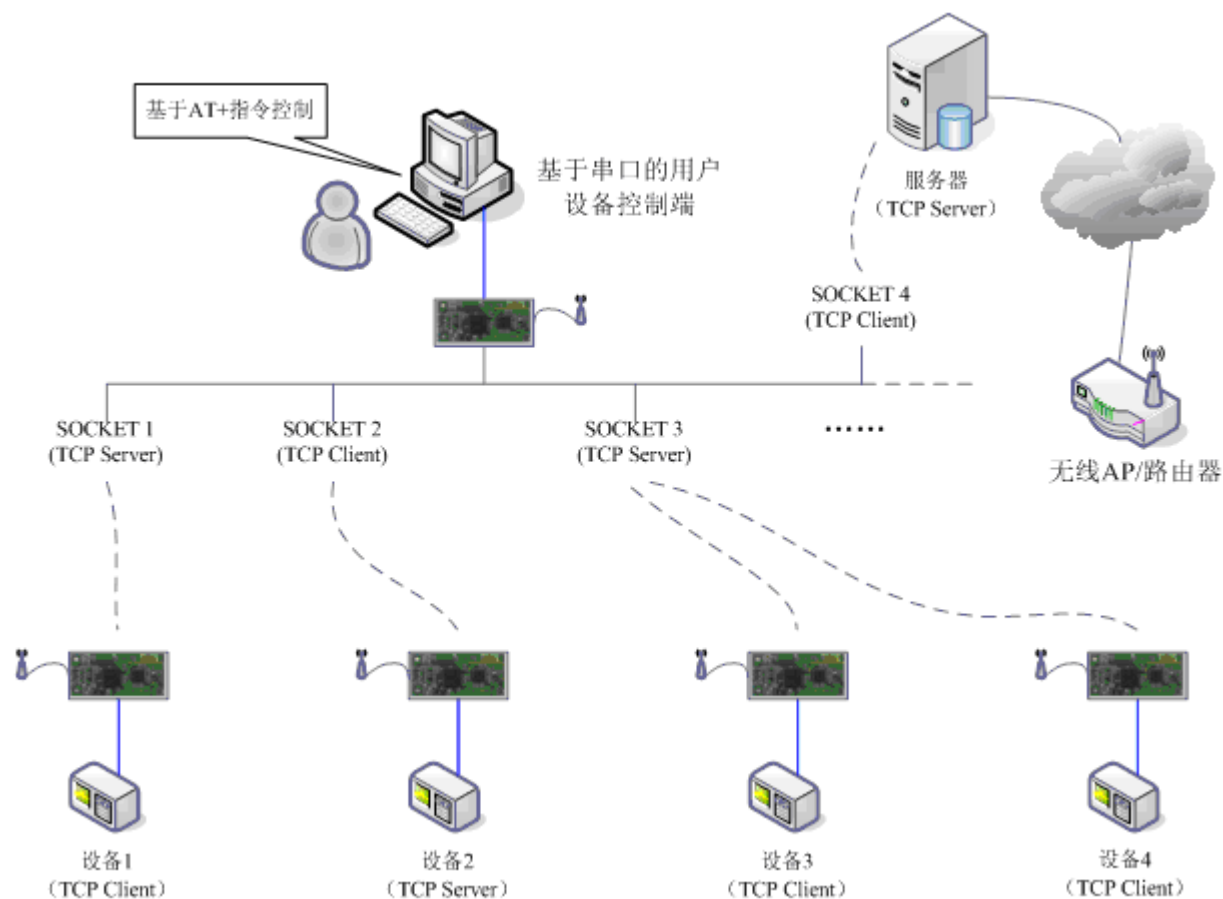


图 4-10 多网络连接模式

如上图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。

#### 4.5.5.2 参考设置

##### 设备控制端

##### ◇ 无线设置

- 网络名称、加密类型、密钥等：参见无线 AP/路由器设置

##### ◇ 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

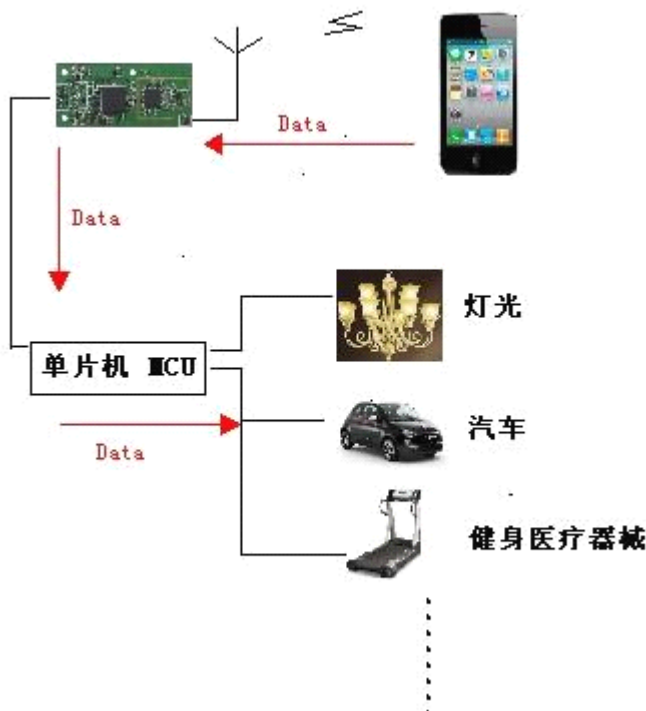
✧ 工作模式设置

- 不启用自动工作模式

#### 4.5.6; **iphone** 手机等智能设备应用（或者笔记本电脑 **OR** 带无线网卡的台式机）

随着 Apple 公司的 iphone 和 ipad 大热卖，Apple 粉丝的增多，iphone 周边产品的研发也有越来越多的公司进入。目前以开发 iphone 手机软件的公司每天都在增加中。

我司的串口 wifi 模块，型号 HLK-WIFI-M03，支持 AD-HOC 模式，透明传输，上电后我们的 M03 模块可以发起一个 AD-HOC 的网络，iphone 加入这个网络，之后就可以用 iphone 控制您的设备了。具体连接方式见下图



数据从 iphone 经过我们的串口 wifi 模块透明传输到客户设备的 MCU 中，然后 MCU 做相关的动作控制灯光或者汽车或者其他设备。

#### HLK-WIFI-M03 的参考设置

✧ 无线设置（高级设置里面）

- 网络类型: 自组网络，使能“网络不存在时自动创建”勾选
- 网络名称: my\_adhoc(可修改成任何名称)
- 加密类型: 开放
- 其余默认

✧ 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

✧ 工作模式设置

- 启用自动工作模式
- 协议类型：TCP
- C/S 模式：服务器
- 端口号：6000
- 连接超时：0

Iphone/IPAD 等的网络配置

✧ 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

然后 Iphone 上运行一个 TCP 协议的控制程序即可完成 data 的透明传输。

## 5 配置管理指南

### 5.1 使用配置管理软件

#### 5.1.1 简介

我们提供了一个可以方便的管理 HLK-WIFI-M03 模块的配置管理程序，用于配置查询/配置、功能测试等，其主要功能包括：

- ✧ 模块管理
  - 通过串口连接
  - 通过无线连接
- ✧ 参数配置
  - 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
- ✧ 功能测试
  - 无线网络命令测试
  - Socket 命令测试
  - AT+指令测试
- ✧ 固件升级
- ✧ 系统信息查询

#### 5.1.2 模块管理

##### 5.1.2.1 通过串口连接

- 1、将 HLK-WIFI-M03 模块连接串口转接板，连接好串口线后打开电源；
- 2、启动 UART-WIFI 配置管理程序，从端口列表中选择模块的连接端口号，如果不确定可以选择自动；（最好不要选择自动）
- 3、点击“搜索模块”，（如果搜索失败，则应该先退出透传模式）如果连接端口选择“自动”且计算机有多个串口时，可能会花费较长的搜索时间，完成后如下图所示：

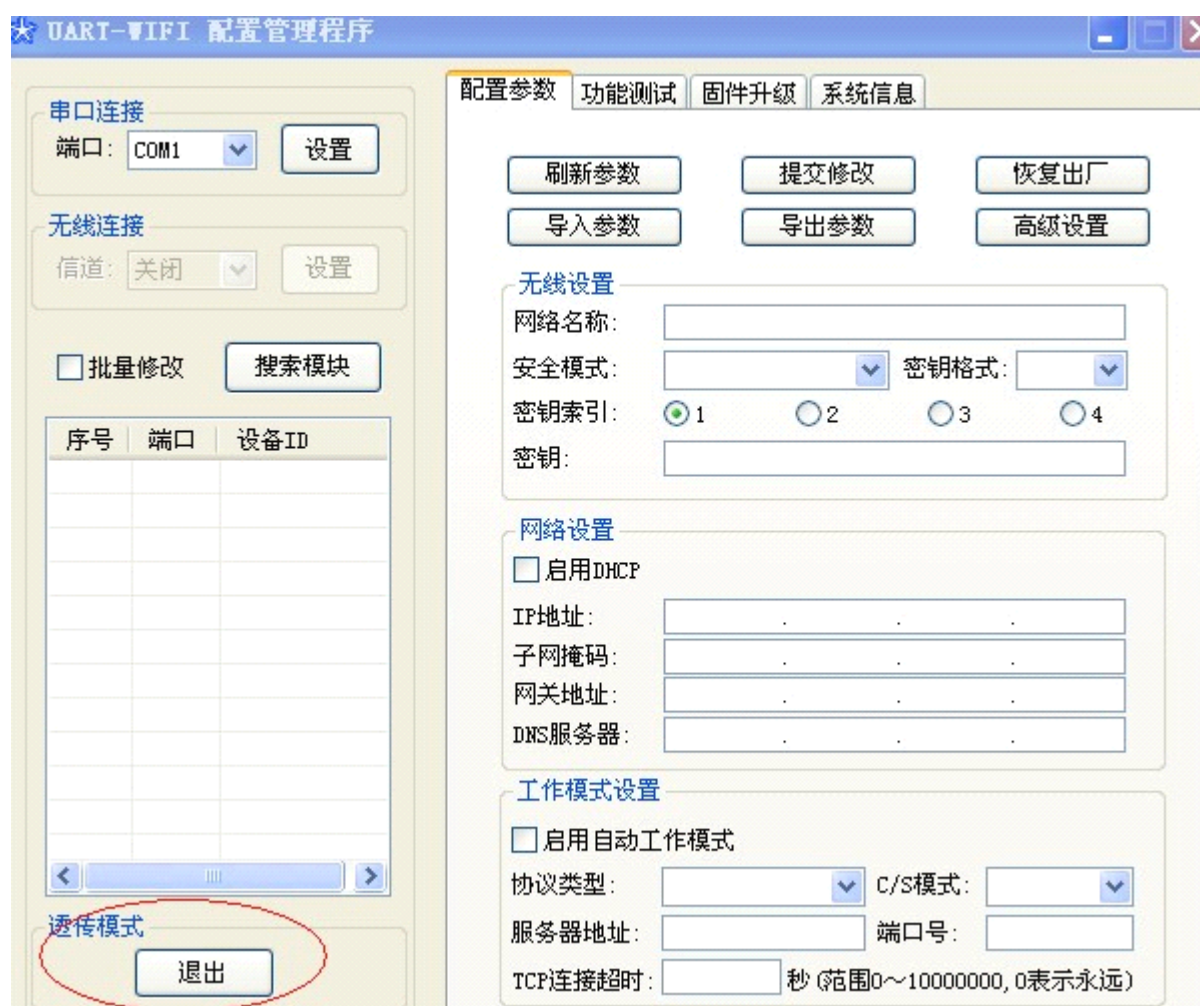


图 5-1 串口连接示意图

### 5.1.2.2 通过无线连接

注意：要使用本功能需要选配无线适配器并且正确安装无线驱动程序。

- 1、将如图 2-5 所示的无线适配器插入计算机；
- 2、安装无线驱动程序，成功后如下图所示；

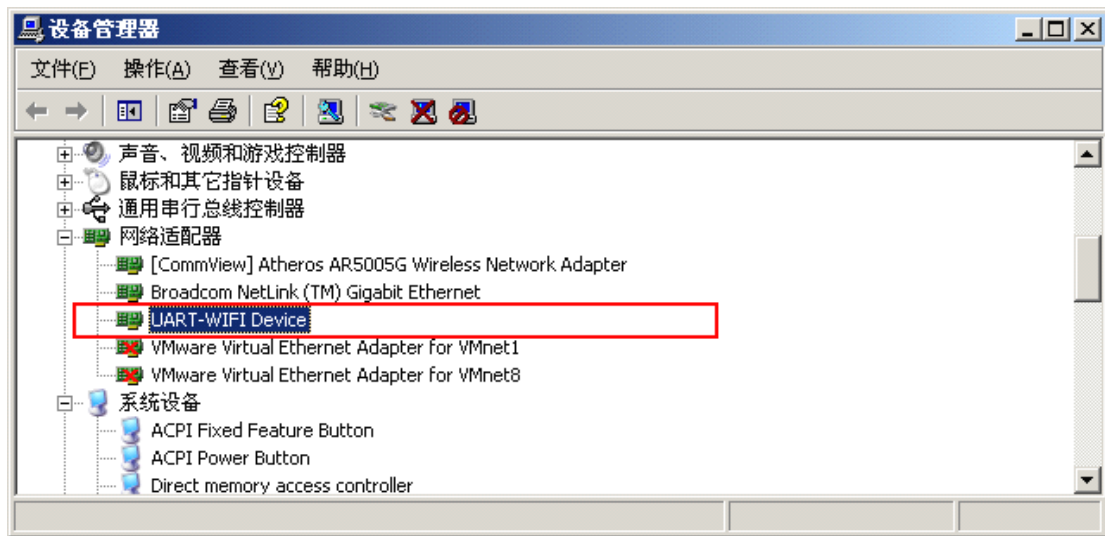


图 5-2 无线适配器驱动安装

- 3、启动 UART-WIFI 配置管理程序，如果驱动程序安装成功，无线连接设置框将处于激活状态；
- 4、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）；
- 5、从信道列表中选择希望扫描的信道，如果选择自动，将扫描 1~14 所有信道；
- 6、点击“搜索模块”，如果信道选择“自动”可能会花费较长的搜索时间，完成后如下图所示；



图 5-3 无线连接示意图

### 5.1.3 配置参数

本页为用户提供了快速查询/修改模块配置参数的功能。



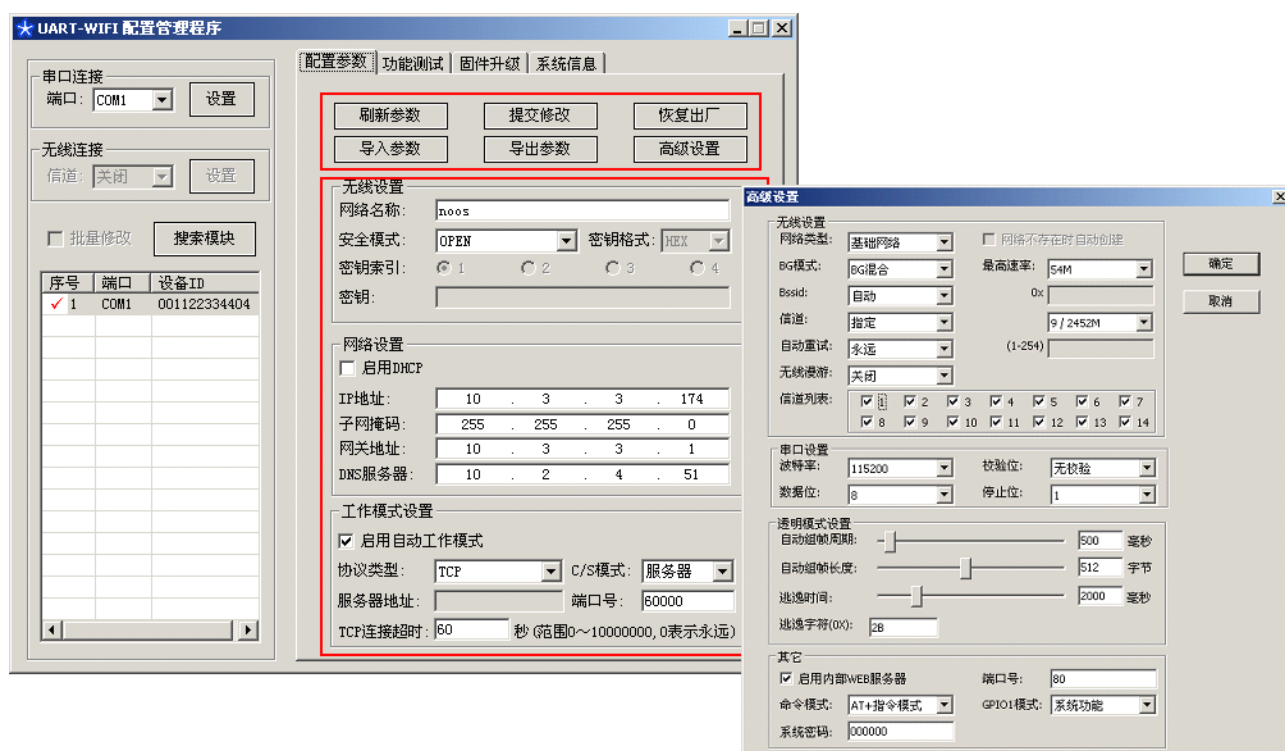


图 5-4 配置参数示意图

#### 5.1.3.1 刷新参数

用户点击“刷新参数”按钮可以从无线模块重新读取配置参数，并刷新到显示界面。

#### 5.1.3.2 提交修改

用户在通过显示界面修改完配置参数后（包括高级设置中的参数），需要点击“提交修改”按钮才能将修改后的参数保存到无线模块中。参数修改后需要复位无线模块新的参数才能生效，用户可以在修改完成后弹出的对话框中选择立刻复位或稍后手动复位，如下图所示。

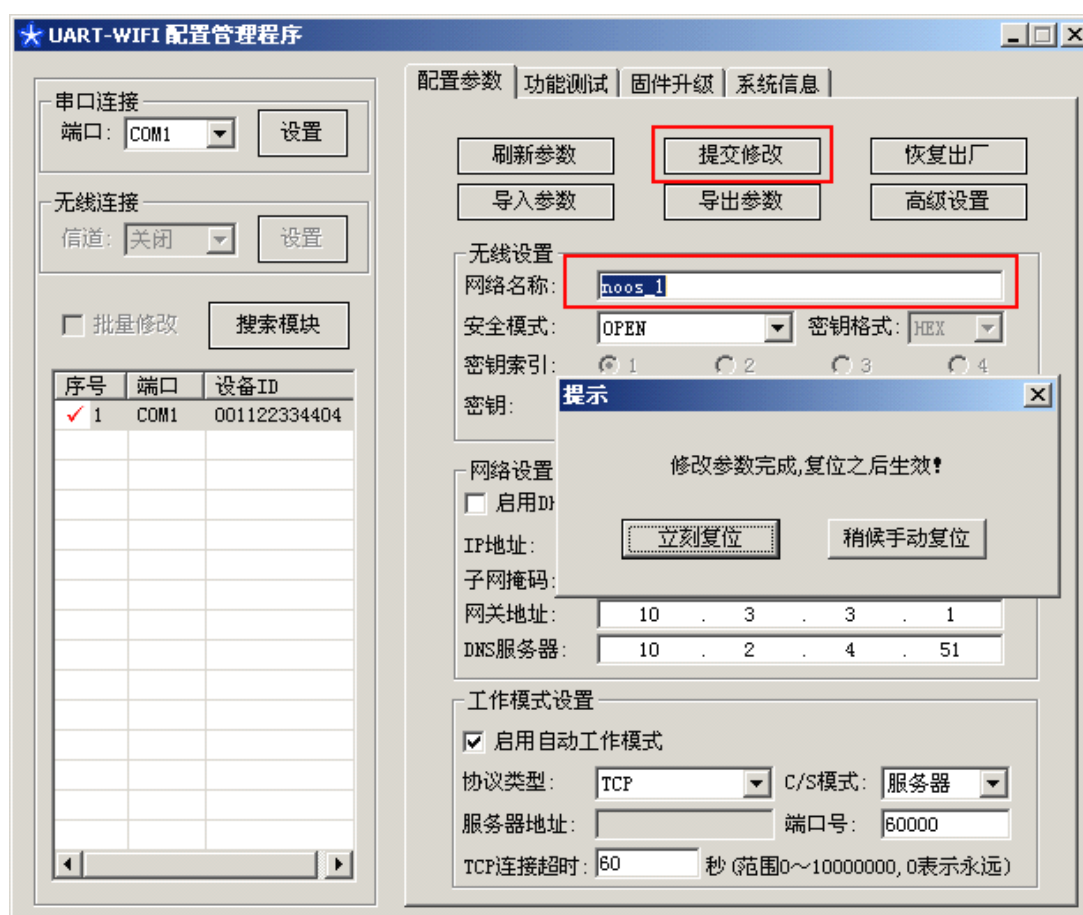


图 5-5 提交修改示意图

### 5.1.3.3 恢复出厂设置

用户点击“恢复出厂”按钮可以将无线模块的配置参数恢复到出厂状态。恢复完成后，需要首先复位模块以使参数生效。



图 5-6 恢复出厂设置示意图

#### 5.1.3.4 导入/导出参数

用户可以通过点击“导出参数”按钮显示界面中的所有配置参数保存到指定的配置文件中。此后通过点击“导入参数”按钮直接使用该文件导入之前保存的所有参数。需要说明的是，使用导入参数功能仅是将参数导入到显示界面中，只有“提交修改”后导入的参数才真正修改到无线模块。

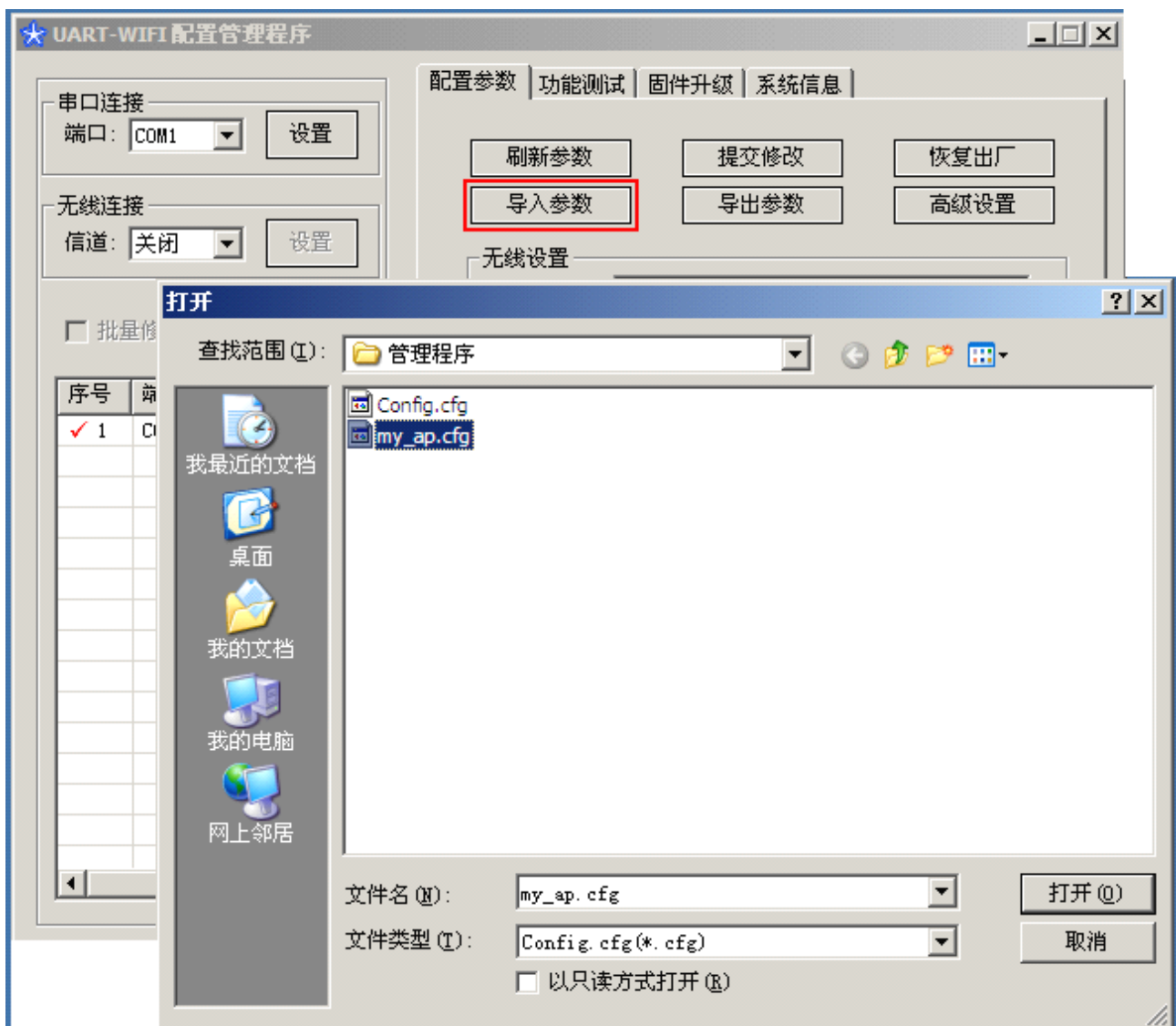


图 5-7 导入参数示意图

#### 5.1.3.5 批量修改

当配置管理程序搜索到多个无线模块时，将激活批量修改功能。用户选中该功能后，在点击“提交修改”时，将会自动的将参数修改应用到所有无线模块，如下图所示。这一功能在需要对大量模块进行统一的出厂配置时，可以简化操作步骤，提高工作效率。



图 5-8 批量修改示意图

#### 5.1.4 功能测试

本页为用户提供了快速测试 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，当本软件通过无线连接到模块时，将无法使用本页功能。

用户既可以通过常用指令的快捷按钮，也可以通过直接输入 AT+指令字符串的方式向模块发送命令，如下图所示。

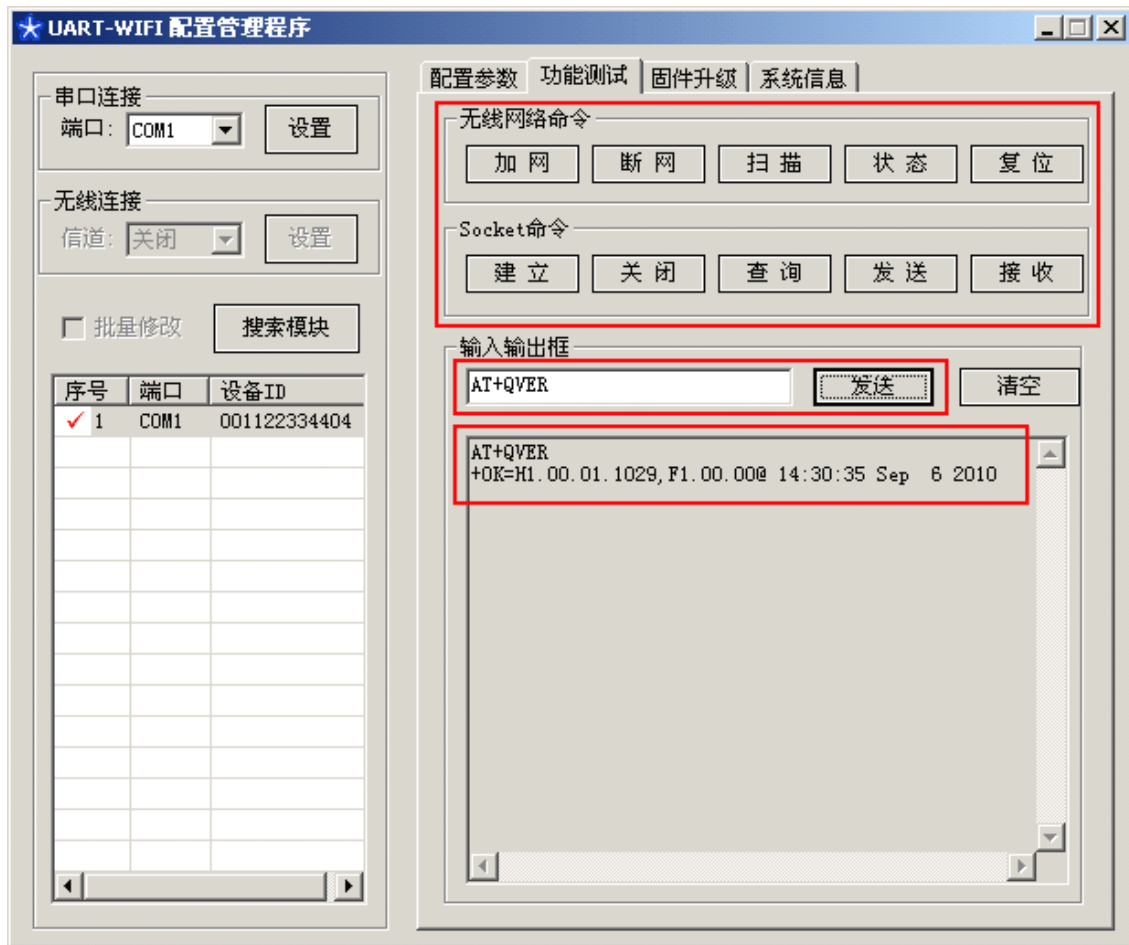


图 5-9 功能测试页

#### 5.1.4.1 扫描网络

用户可点击“扫描”按钮来扫描网络，如下图所示。扫描结果，如网络类型、加密、网络 MAC、信道、信号强度等会在下面显示框中显示。

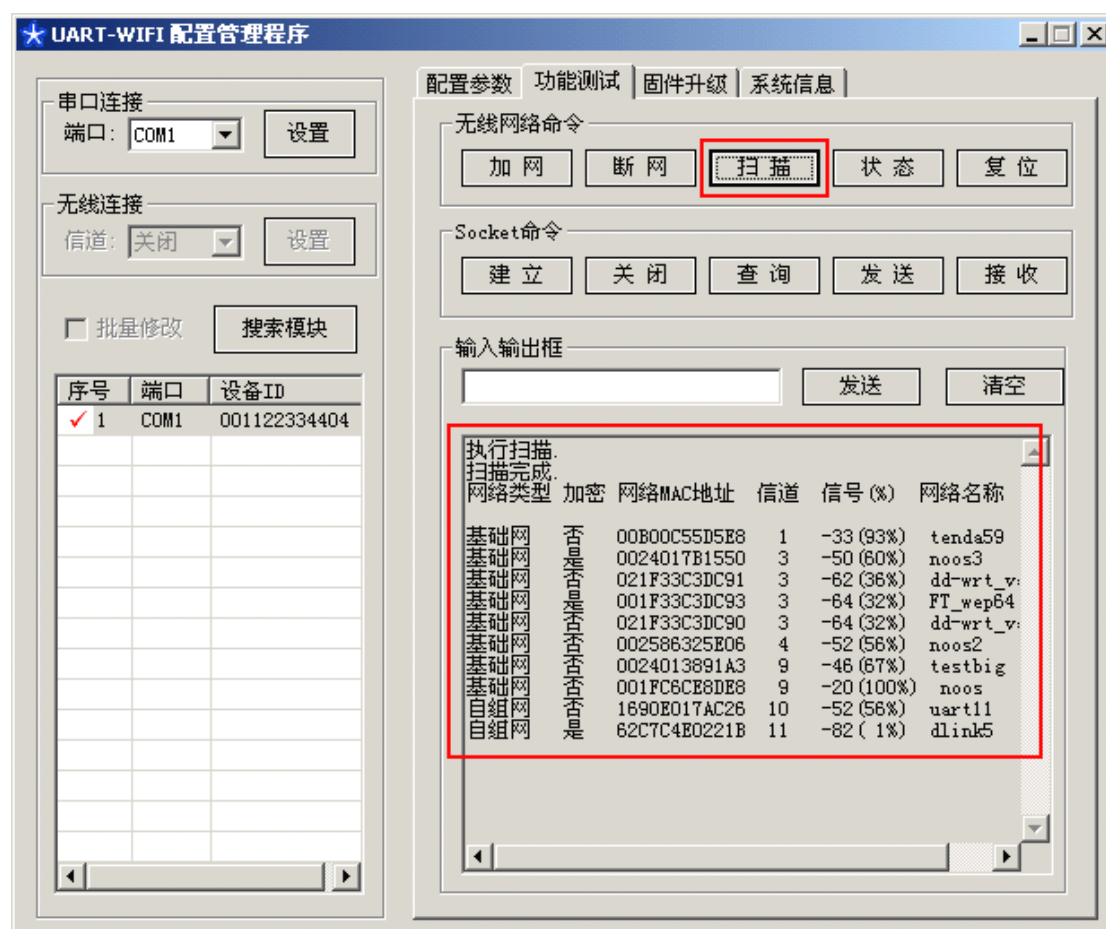


图 5-10 扫描网络示意图

#### 5.1.4.2 加入/断开网络

用户可点击“加网”、“断网”按钮加入或断开无线网络，如图所示，结果都会在下面显示框中显示。





图 5-11 加入/断开网络示意图

#### 5.1.4.3 查询网络状态

用户可点击“状态”按钮可以查询网络连接状态，如图所示。查询结果会在下面显示框中显示。





图 5-12 查询网络状态示意图

#### 5.1.4.4 socket建立

用户可点击“建立”按钮创建 socket，如图所示，配置好各项参数，点击“确定”按钮，创建成功后将返回一个 socket 号。

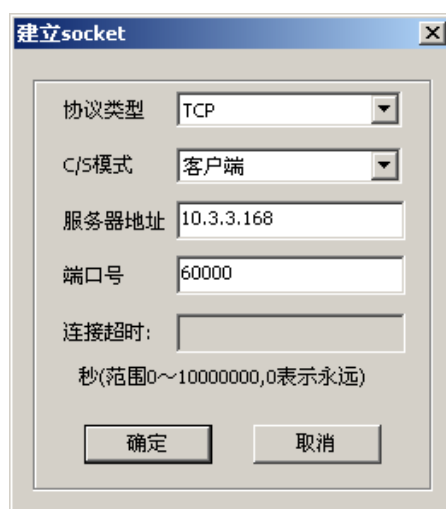


图 5-13 Socket 建立参数输入

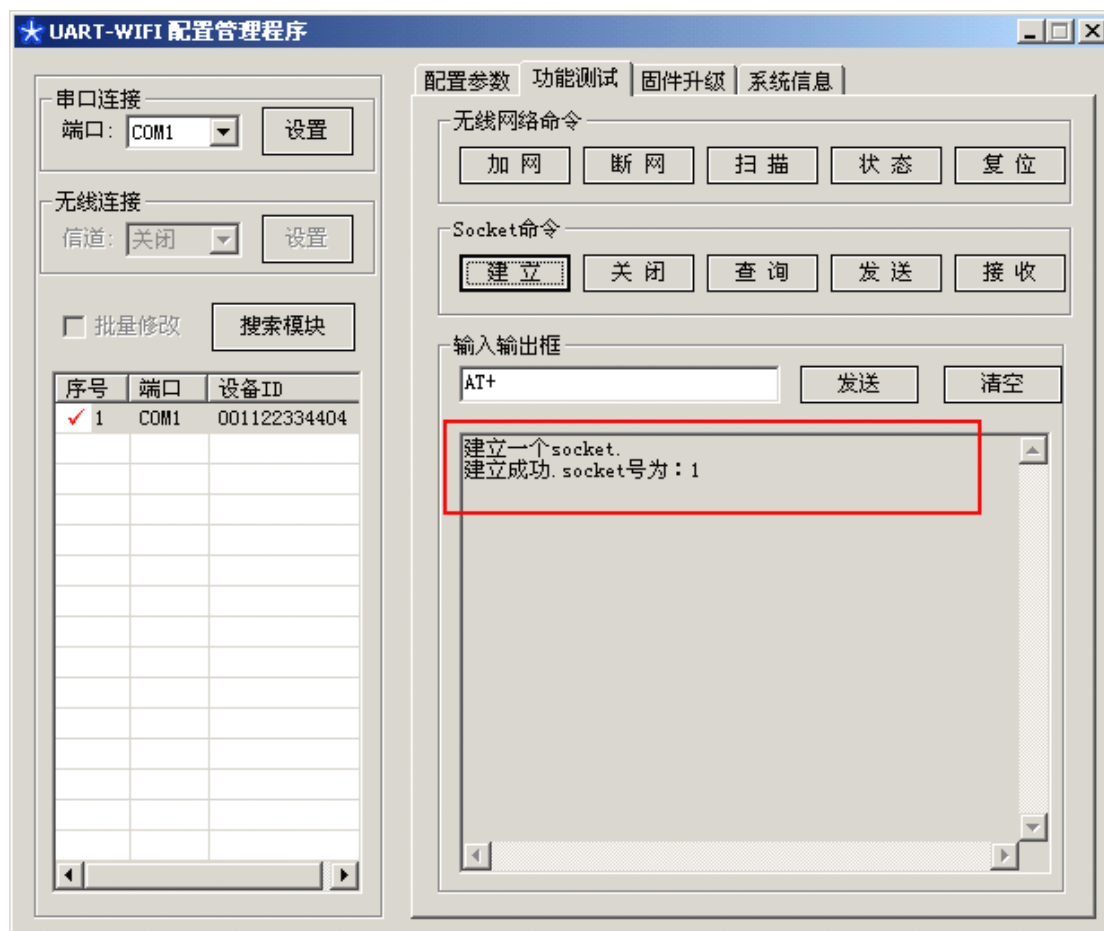


图 5-14 Socket 建立成功示意图

#### 5.1.4.5 socket关闭

用户可点击“关闭”按钮，输入需要关闭的 socket 号，就可以删除之前建立的 socket 连接。

#### 5.1.4.6 socket查询

用户可点击“查询”按钮查询 socket，输入需要查询的 socket 号，即创建时返回的 socket 号，点击“确认”按钮即可，显示框会返回 socket 当时的状态。



图 5-15 Socket 查询示意图

说明:

1、上图所示的为查询两个 socket 返回的结果，二者显示的内容有所不同。这是因为第 2 个 socket 类型为 TCP 服务器，创建成功后处于监听状态。查询该 socket 是不仅包含了其自身的状态，还包括了连接到该服务器的 Client 对应的 socket 连接信息（这些 socket 由系统自动创建），如图中所示的 socket3 和 4；

#### 5.1.4.7 socket发送

用户可点击“发送”按钮通过 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，而是应该使用其客户端连接 socket 号）及需要发送的数据，点击“确认”按钮即可，显示框会返回发送完成信息。

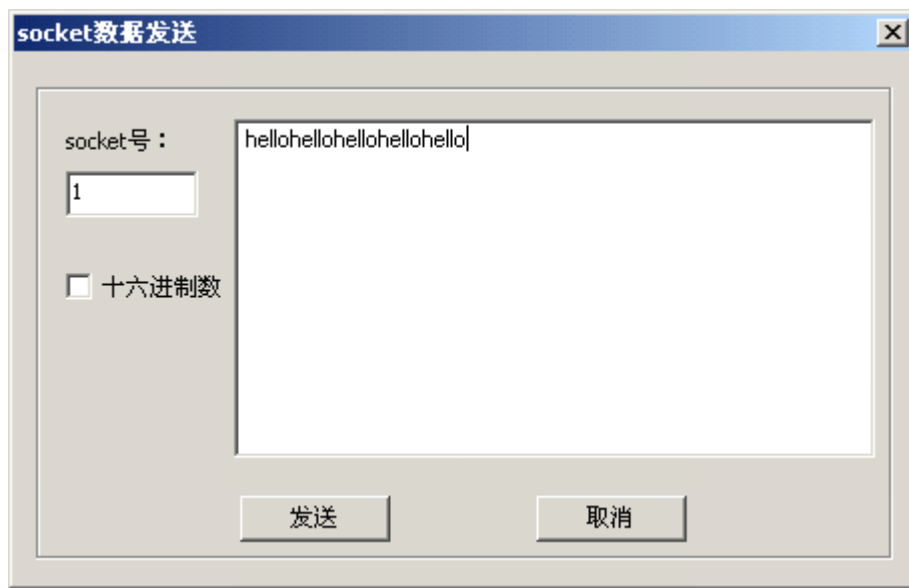


图 5-16 Socket 数据发送示意图

#### 5.1.4.8 socket接收

用户可点击“接收”按钮通过 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，而是应该使用其客户端连接 socket 号）及期望接收的数据个数，点击“接收”按钮，数据即会显示到接收窗口中。

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。

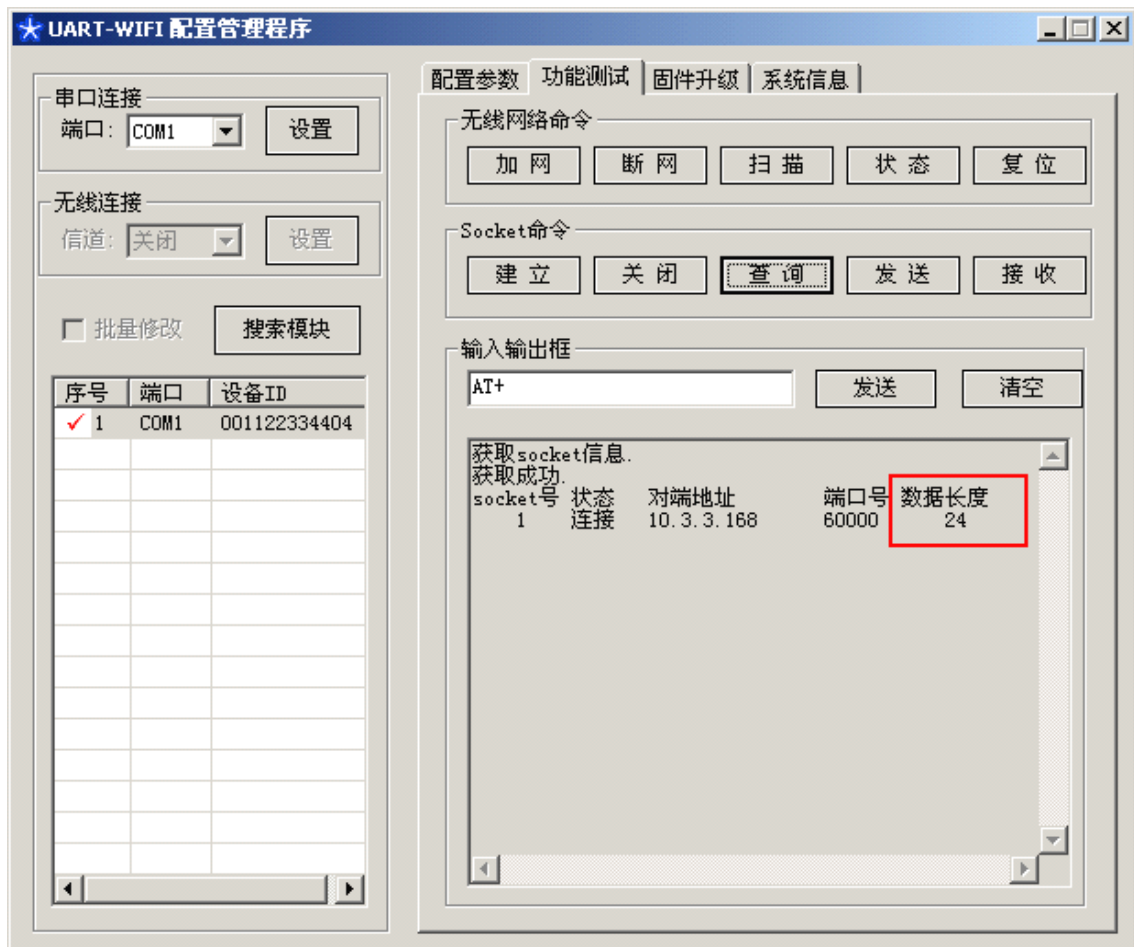


图 5-17 查询 Socket 接收缓冲区数据长度示意图

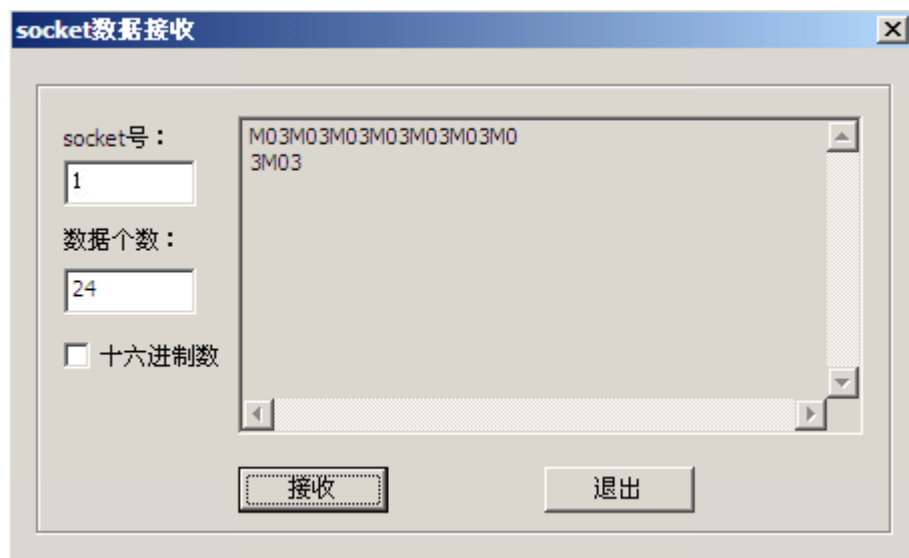


图 5-18 Socket 接收示意图



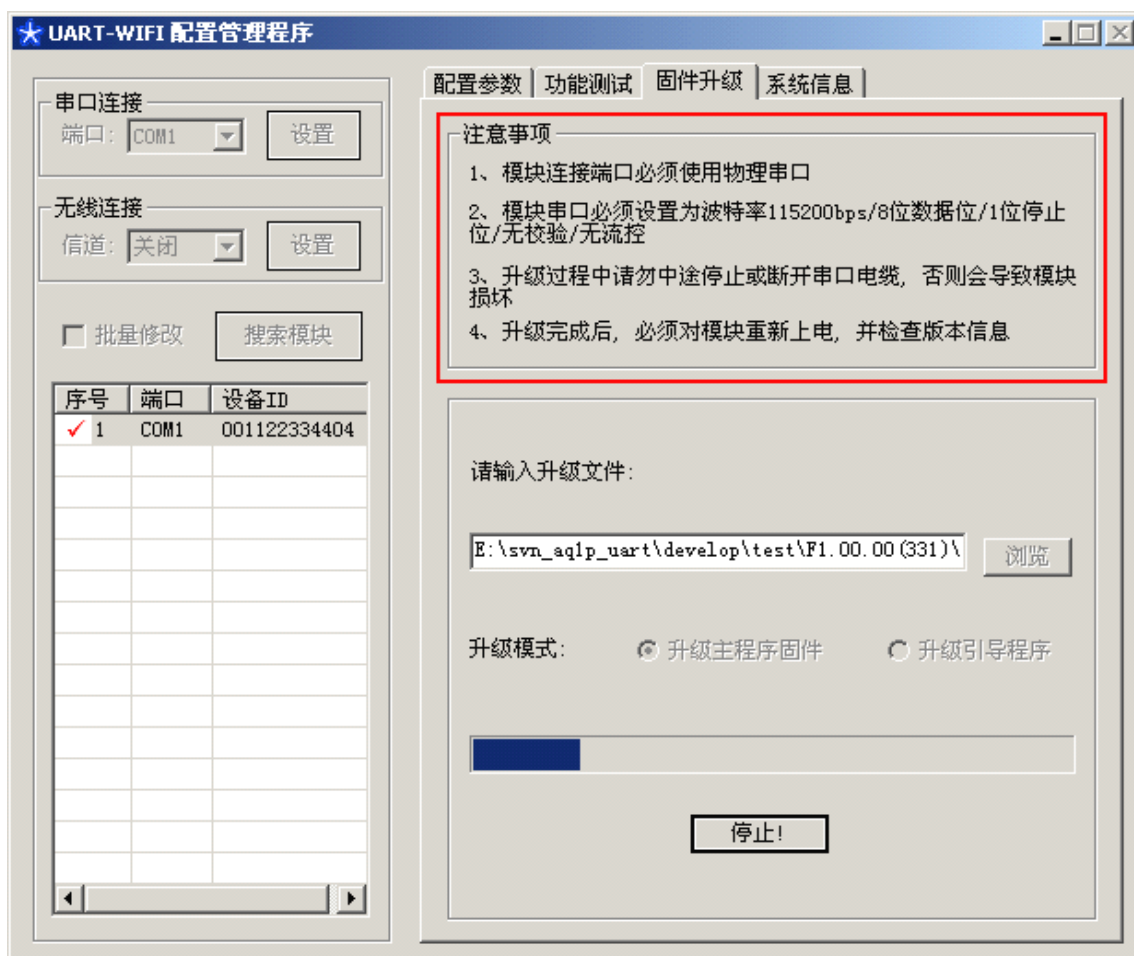


图 5-20 固件升级示意图

## 5.2 使用WEB服务器

本产品内置 WEB 服务器，支持使用 IE 浏览器通过网页进行参数配置功能，使用方法如下：

注意事项：

- 1、要使用网页配置，必须保证模块已经成功连接到网络；
- 2、WEB服务器默认端口号为 80，因此在浏览器的地址栏仅输入地址即可，如果用户修改web服务器端口，必须输入设置的端口号，否则页面打不开，如端口设为 8080 时，web服务器的访问地址为 <http://192.168.1.20:8080>；
- 3、请尽量使用 Windows IE 浏览器访问本服务器；

### 5.2.1 查询/修改配置参数

- 1、打开 IE 浏览器，在地址栏输入无线模块 IP 地址后连接，将弹出如下图所示的登录框，输

入用户名 admin，密码为无线模块的登录密码（出厂默认设置为 000000）；



图 5-21 WEB 登录

- 2、网页的配置页面如下图所示，如果要修改无线模块的配置参数，只需在修改完成后点击相应的“save”按钮即可提交修改；



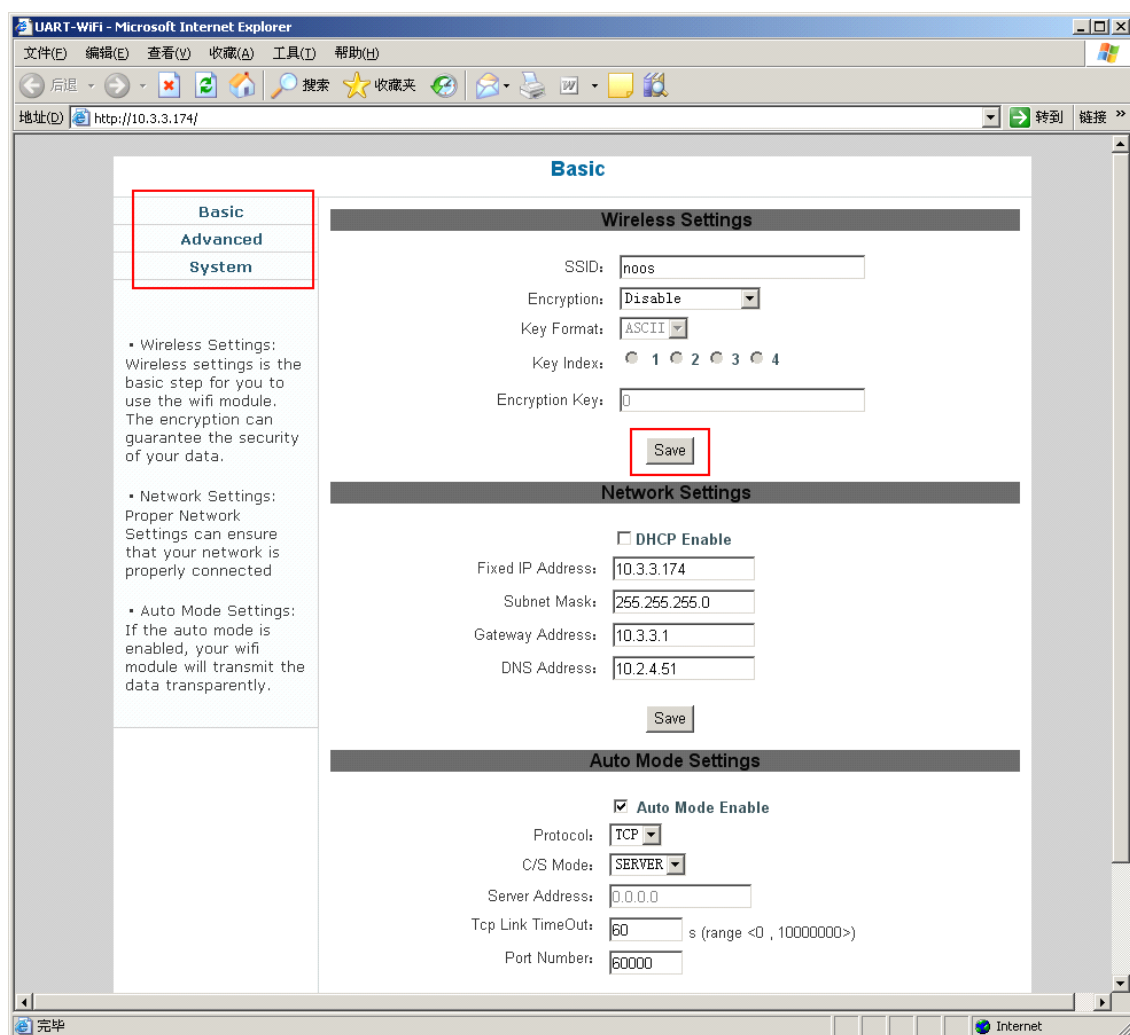


图 5-22 参数配置页面

- 3、提交完成后，无线模块将自动复位，网页也将在 10 秒后自动刷新（如果用户新修改的参数使得模块无法正常联网，会导致网页刷新失败），如下图所示：

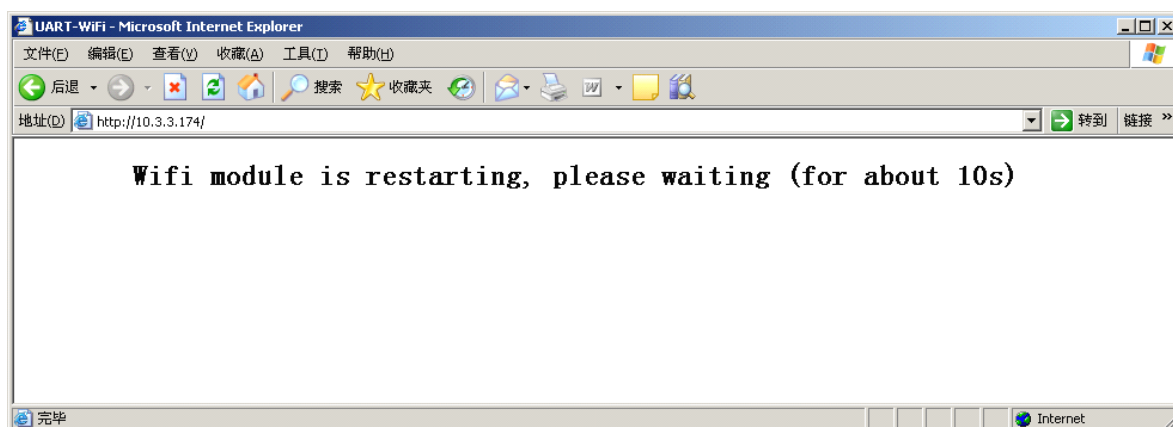


图 5-23 网页自动刷新页面

### 5.2.2 在线固件升级

本产品支持通过网页在线升级固件程序，如下图所示：

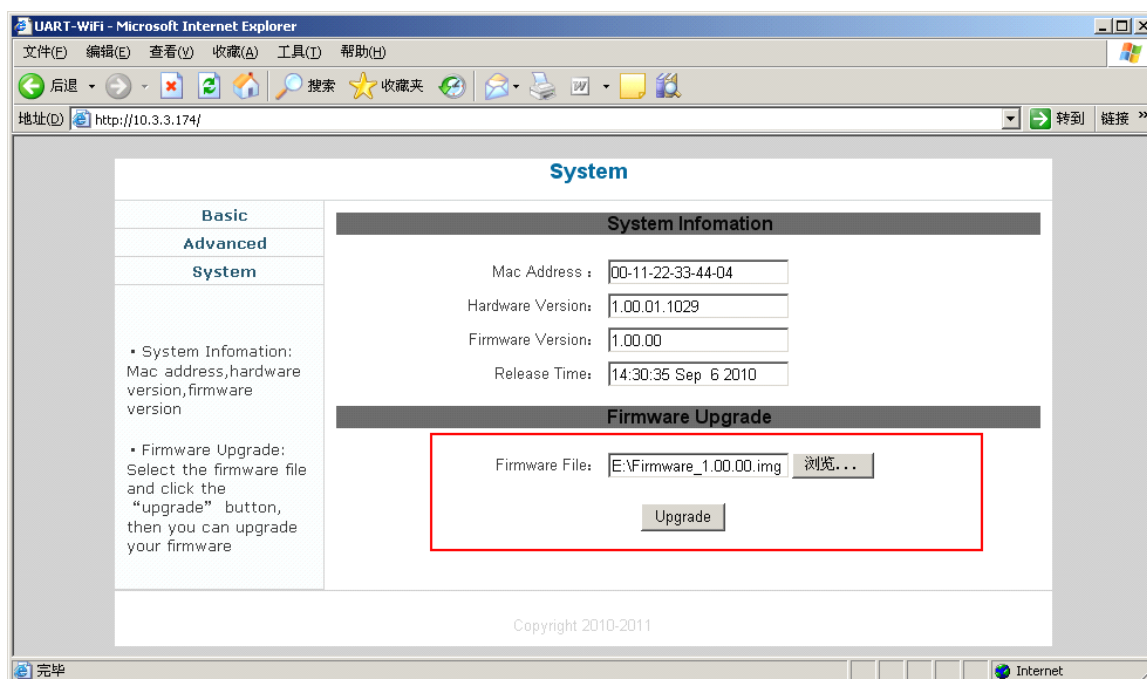


图 5-24 固件在线升级页面

## 5.3 使用超级终端程序

如果用户仅使用无线模块的自动工作模式可以跳过本节内容。

对于希望测试 AT+指令功能的用户，还可以使用 Windows 操作系统自带的超级终端程序直接输入 AT+指令，对无线模块进行参数设置及功能测试。需要说明的是，前文中提到的串口调试软件（串口精灵、串口大师等）均可以用于 AT+指令测试，本节仅以超级终端为例进行说明。

- 1、打开超级终端（windows 开始—程序—附件—通讯—超级终端），输入连接名称和图标，选择连接的串口，选择串口参数（与无线模块设置一致），需要特别说明的是，要使用 AT+指令模式，流控选项必须设置为“无”，如下图所示；

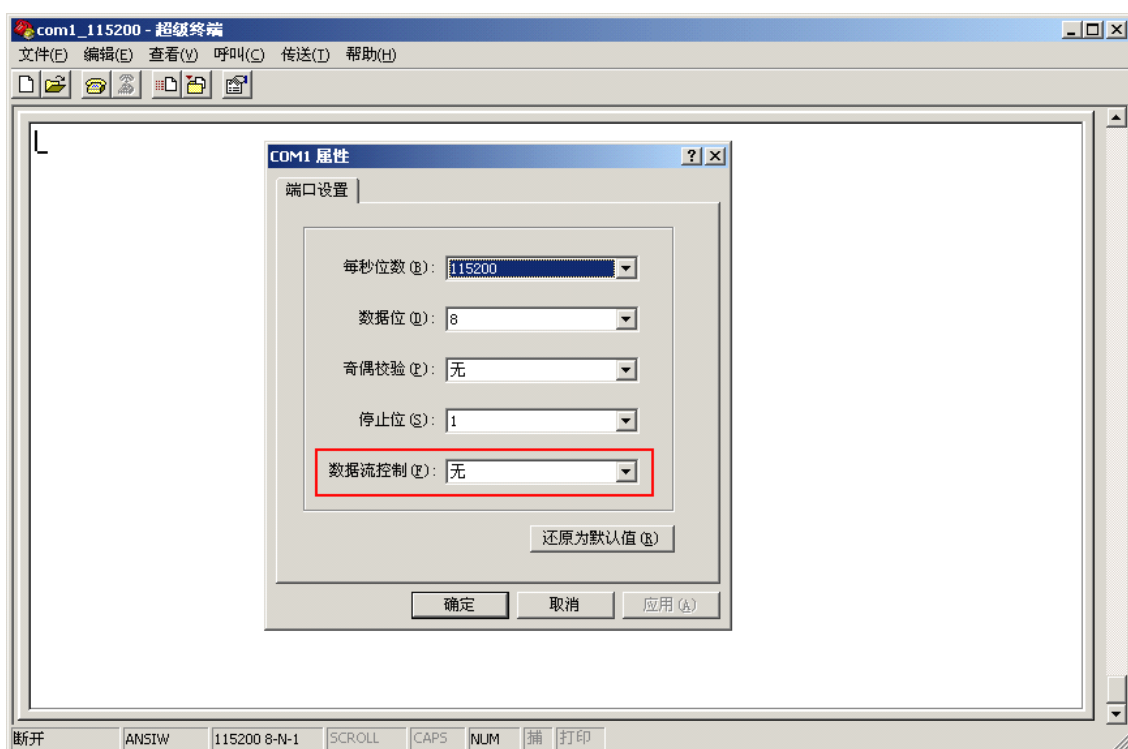


图 5-25 超级终端配置示意图

- 2、如果无线模块已经启动配置模式（参见 4.2.1 节）或工作模式设置为命令模式，直接转到步骤 5，否则，需要首先按照下面的步骤使模块退出透明传输状态（参见 4.2.2.4 节）；
- 3、打开记事本编辑逃逸字符串（默认逃逸字符为”+”）并保存，如下图所示；

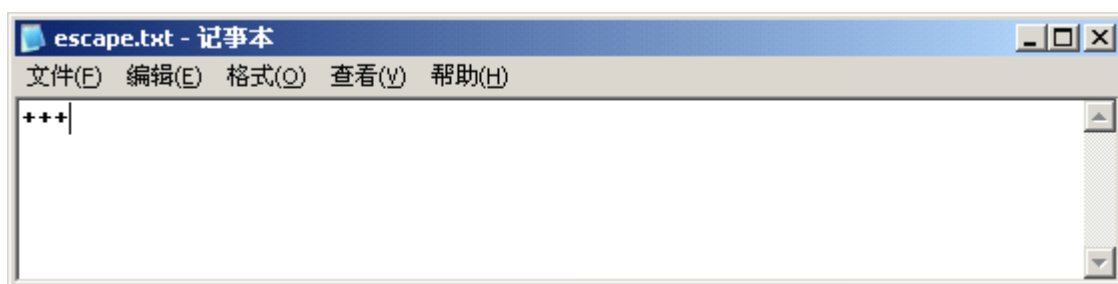


图 5-26 编辑逃逸字符文件

- 4、等待“逃逸时间参数”指定的时间（默认设置为 2 秒），从“传送”菜单中选择“发送文本文件”，发送逃逸字符文件，如下图所示。如果成功则模块返回“+OK”消息；

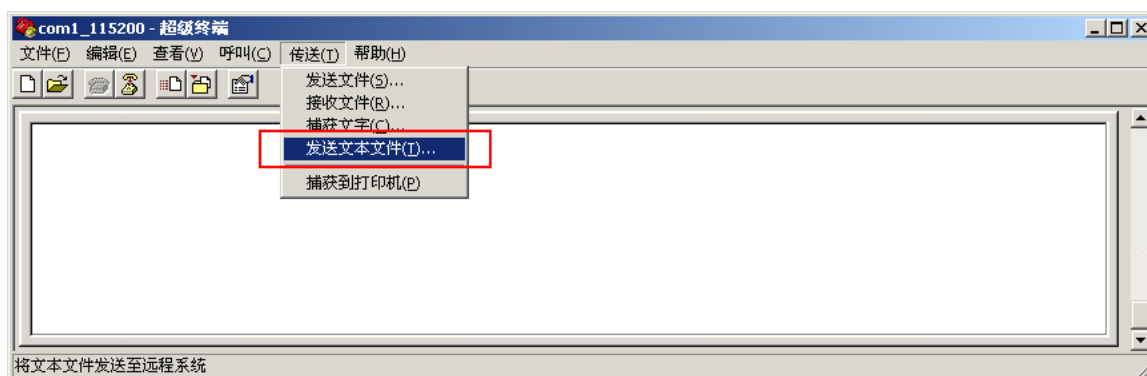


图 5-27 发送逃逸字符文件

- 5、输入“AT+E”指令打开串口输入回显（在回显打开之前，用户输入的字符都无法在超级终端窗口中显示）；
- 6、至此，用户可以使用超级终端程序测试绝大部分 AT+指令，如扫描网络命令，输入“AT+WSCAN”，如下图所示；

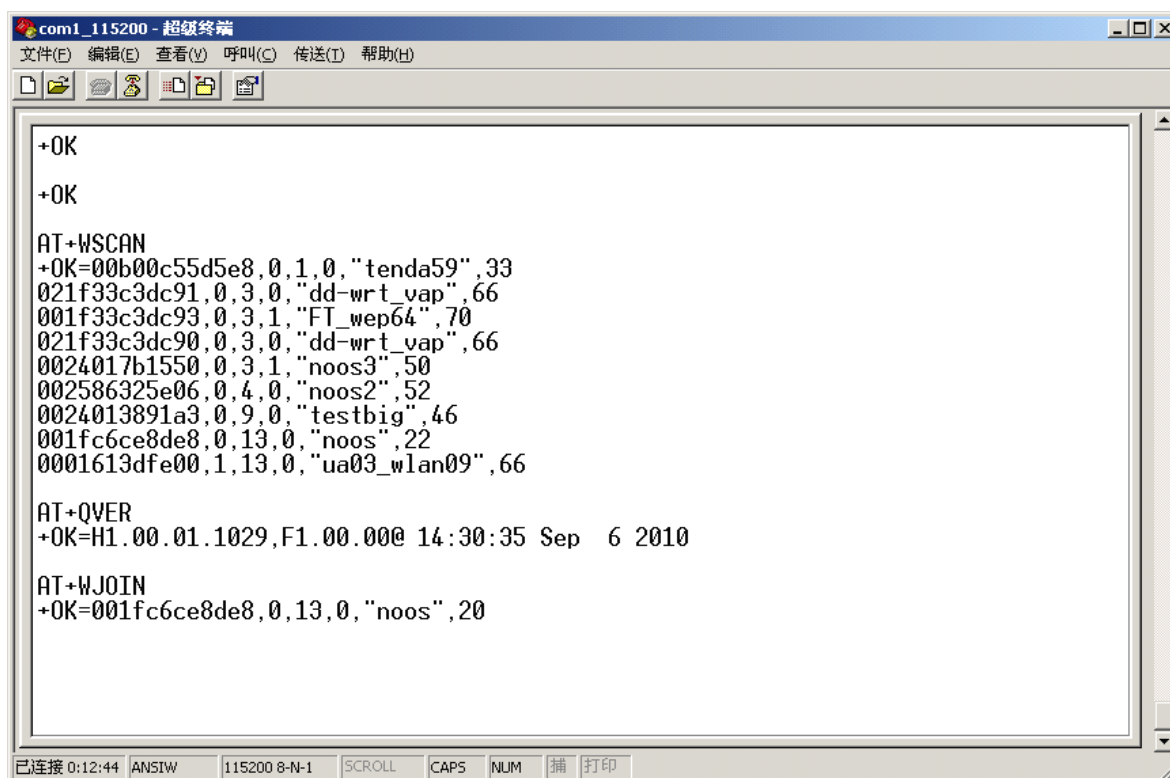


图 5-28 指令测试示意图

## 6 AT+指令控制协议

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。

### 6.1 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 ascii 的命令行格式指令集，下面对其语法格式和处理流程进行描述。

#### 6.1.1 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

<CR>：回车，ascii 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，ascii 字符 0x0d

<LF>：换行，ascii 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

#### 6.1.1.1 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，如下图所示。

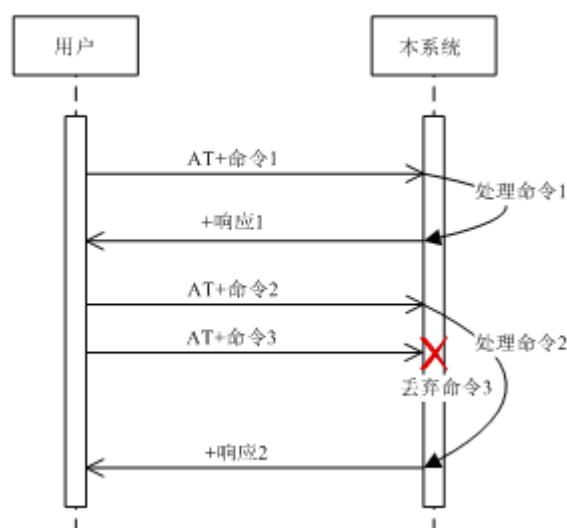


图 6-1 AT+指令处理流程

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，其流程如下图所示。

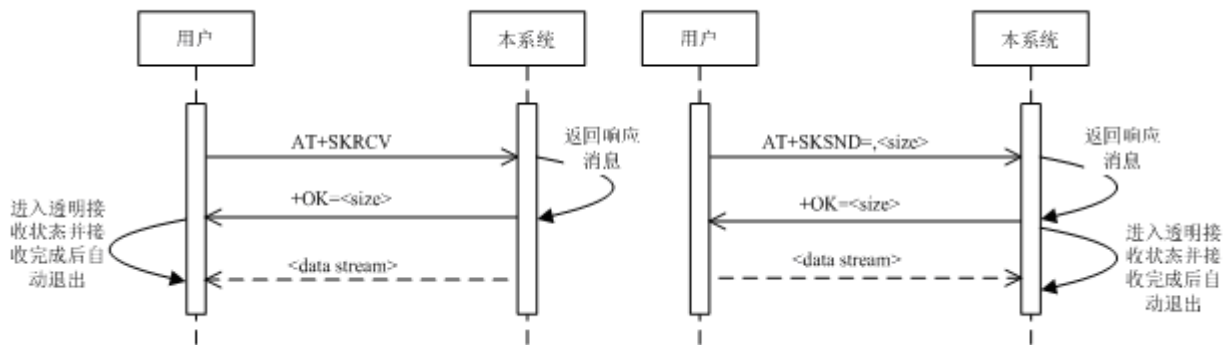


图 6-2 AT+指令特殊处理流程

#### 6.1.1.2 格式范例

范例 1：返回成功消息

```
AT+  
+OK
```

范例 2：返回错误消息

```
AT+WJOIN  
+ERR=-10
```

范例 3：使用输入参数

```
AT+UART=9600, 1, 1, 0  
+OK
```

范例 4：使用参数同步到 Flash 操作符<!>

```
AT+ATPT=!500  
+OK
```

范例 5：使用查询操作符<?>

```
AT+ATPT=?  
+OK=500
```

### 6.1.1.3 错误代码

表 6-1 错误代码

值	含 义
-1	无效的命令格式
-2	命令不支持
-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

## 6.2 指令集

### 6.2.1 指令列表

表 6-2 AT+指令列表

指令名称	用 途
<a href="#">(null)</a>	空指令
<a href="#">ATLT</a>	设置/查询数据自动组帧数据长度
<a href="#">ATM</a>	设置/查询模块工作模式
<a href="#">ATPT</a>	设置/查询数据自动组帧周期
<a href="#">ATRM</a>	设置/查询自动工作模式下模块自动创建的 socket 连接信息
<a href="#">BSSID</a>	设置/查询指定 AP 的 bssid 地址
<a href="#">CHL</a>	设置/查询指定无线信道方式
<a href="#">CHLL</a>	设置/查询无线信道列表
<a href="#">CMDM</a>	设置/查询系统的默认命令模式
<a href="#">E</a>	切换串口字符回显
<a href="#">ENCRY</a>	设置/查询无线网络安全模式
<a href="#">ENTM</a>	进入串口透明传输模式
<a href="#">ENTS</a>	进入睡眠模式



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

## 6.2.2 网络控制类

### 6.2.2.1 AT+WJOIN

功能：

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

表 6-3 网络类型

值	含 义
0	infra 网络
1	adhoc 网络

channel: 信道号

b\_encry: 加密模式

表 6-4 加密模式

值	含 义
0	开放
1	加密

ssid: 无线网络名称，1~32 个字符，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

#### 6.2.2.2 AT+WLEAV

**功能：**

断开当前无线网络。

**格式：**

AT+WLEAV<CR>

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

**参数：**

无

#### 6.2.2.3 AT+WSCAN

**功能：**

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

**6.2.2.4 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

表 6-5 连接状态

值	含 义
0	断开
1	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

**6.2.2.5 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,

表 6-6 协议类型

值	含 义
---	-----

0	TCP
1	UDP

cs: C/S 模式,

表 6-7 CS 模式

值	含 义
0	Client
1	Server

host\_timeout: 根据 protocol 及 cs, 其含义分别如下

表 6-8 host\_timeout

cs	protocol	含 义
0	X	目的服务器名称, 可以输入域名或 ip 地址, 如"192.168.1.100"或"www.sina.com.cn"
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

port: 端口号

socket: socket 号

#### 6.2.2.6 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

#### 6.2.2.7 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

#### 6.2.2.8 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

表 6-9 Socket 状态

值	含 义
0	断开
1	监听

2	连接
---	----

host: 对端 ip 地址

port: 对端端口号

rx\_data: 接收 buffer 中数据长度

#### 6.2.2.9 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

#### 6.2.2.10 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

#### 6.2.3 系统控制类

##### 6.2.3.1 AT+

##### 功能:

空指令。

##### 格式:

AT+<CR>

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

##### 参数:

无

#### 6.2.3.2 AT+Z

**功能：**

复位系统。

**格式：**

AT+Z<CR>

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

**参数：**

无

#### 6.2.3.3 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

#### 6.2.3.4 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

#### 6.2.3.5 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

#### 6.2.3.6 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

#### 6.2.3.7 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

#### 6.2.3.8 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

表 6-10 IO 状态

值	含 义
0	高电平
1	低电平



#### 6.2.3.9 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

#### 6.2.3.10 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”

### 6.2.4 参数设置类

#### 6.2.4.1 AT+NIP

**功能：**

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。

**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

表 6-11 地址类型

值	含 义
0	使用 DHCP 动态分配
1	使用静态 IP 地址

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 地址号

dns: DNS 地址，数据格式同 ip 地址

#### 6.2.4.2 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

表 6-12 工作模式

值	含 义
0	自动工作模式
1	命令工作模式

#### 6.2.4.3 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:

protocol: 协议类型,

表 6-13 协议类型

值	含 义
0	TCP
1	UDP

cs: C/S 模式,

表 6-14 CS 模式

值	含 义
0	Client
1	Server

host\_timeout: 根据 protocol 及 cs, 其含义分别如下

表 6-15 host\_timeout

cs	protocol	含 义
0	X	目的服务器名称，可以输入域名或 ip 地址，如“192.168.1.100”或“www.sina.com.cn”
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

port: 端口号

#### 6.2.4.4 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

#### 6.2.4.5 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。

##### 格式:

AT+ENCRY=[!?][encry mode]<CR>

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

##### 参数:

encry mode: 安全模式

表 6-16 安全模式

值	含 义
0	OPEN
1	WEP64

2	WEP128
3	WPA-PSK (TKIP)
4	WPA-PSK (CCMP/AES)
5	WPA2-PSK (TKIP)
6	WPA2-PSK (CCMP/AES)

#### 6.2.4.6 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

表 6-17 密钥格式

值	含 义
0	HEX
1	ASCII

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，根据不同的安全模式，密钥使用的长度与格式要求定义如下，

表 6-18 密钥

安全模式	密钥格式	
	HEX	ASCII
WEP64	10 个 16 进制字符 <sup>(注 1)</sup>	5 个 ASCII 字符 <sup>(注 2)</sup>
WEP128	26 个 16 进制字符	13 个 ASCII 字符
WPA-PSK (TKIP)	64 个 16 进制字符	8~63 个 ASCII 字符
WPA-PSK (CCMP/AES)	64 个 16 进制字符	8~63 个 ASCII 字符
WPA2-PSK (TKIP)	64 个 16 进制字符	8~63 个 ASCII 字符
WPA2-PSK (CCMP/AES)	64 个 16 进制字符	8~63 个 ASCII 字符

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

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

#### 6.2.4.7 AT+BSSID

##### 功能:

设置/查询指定 AP 的 bssid 地址。本设置仅在 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

表 6-19 BSSID 模式

值	含 义
0	自动
1	指定

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

#### 6.2.4.8 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

表 6-20 信道模式

值	含 义
0	自动
1	指定

channel: 无线信道号, 有效范围 1~14

#### 6.2.4.9 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

#### 6.2.4.10 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

表 6-21 网络类型

值	含 义
0	infra 网络
1	adhoc 网络

#### 6.2.4.11 AT+WATC

**功能:**

设置/查询是否使能自动创建 adhoc 网络功能。本参数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

表 6-22 使能标志

值	含 义
0	不使能
1	使能

#### 6.2.4.12 AT+WARM

**功能:**

设置/查询是否使能无线网络漫游功能。

**格式:**

AT+WARM=[!?][enable]<CR>

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

**参数:**

enable: 使能标志

表 6-23 使能标志

值	含 义
0	不使能
1	使能

## 6.2.4.13 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。本参数仅在自动工作模式下有效。

**格式:**

AT+WARC=[!?][count]<CR>

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

**参数:**

count: 重试次数

表 6-24 重试次数

值	含 义
0	不重试
1~254	重试次数
255	永远重试

## 6.2.4.14 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

表 6-25 BG 模式

值	含 义
0	B/G 混合
1	B

max rate: 最高发送速率，在 B 模式下，仅 0~3 有效。

表 6-26 最高发送速率

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

#### 6.2.4.15 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，

表 6-27 波特率

值	含 义
115200	115200 bps
57600	57600 bps
38400	38400 bps
19200	19200 bps
9600	9600 bps



4800	4800 bps
2400	2400 bps
1200	1200 bps

data bit: 数据位

表 6-28 数据位

值	含 义
0	8 位
1	7 位

stop bit: 停止位

表 6-29 停止位

值	含 义
0	1 位
1	不支持
2	2 位

parity: 校验

表 6-30 校验

值	含 义
0	无校验
1	奇校验
2	偶校验

#### 6.2.4.16 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, 100~10000ms, 单位 ms, 最小步长 100ms

#### 6.2.4.17 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，64~1024，单位字节

**6.2.4.18 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，格式为使用 2 个十六进制数形式表示的 ASCII 字符，如 ASCII 字符“+”应表示为“2B”。

**6.2.4.19 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

**6.2.4.20 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

表 6-31 使能标志

值	含 义
0	不使能

1	使能
---	----

port：服务器端口号，缺省为 80

6.2.4.21 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 ASCII 字符

6.2.4.22 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

表 6-32 工作模式

值	含 义
0	系统功能
1	输入
2	输出

6.2.4.23 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

表 6-33 命令模式

值	含 义
0	AT+指令
1	兼容协议

### 6.3 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。

#### 6.3.1 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。

#### 6.3.2 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下图所示。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。

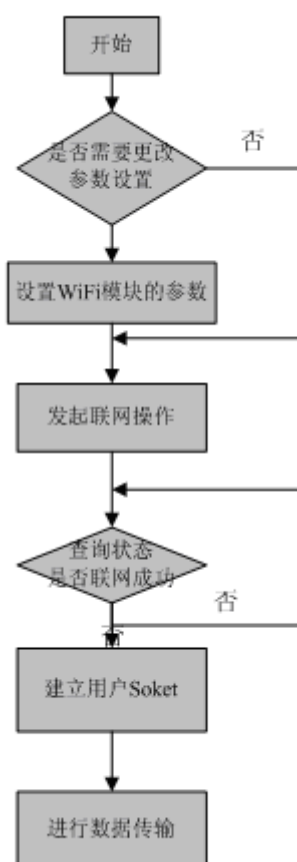


图 6-3 Wifi 网卡操作流程

### 6.3.3 示例代码主要的数据结构

#### 6.3.3.1 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

#### 6.3.3.2 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。

#### 6.3.3.3 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

#### 6.3.4 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。

##### 6.3.4.1 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。

#### 6.3.4.2 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。

### 6.3.5 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。

### 6.3.6 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?-1);
}
```

### 6.3.7 几种AT指令发送解析示例

#### 6.3.7.1 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 (strtouarray(Jionr.BSSID, 6, argv[0]) < 0)
        {
            printf("Error Bssid\n");
            return ;
        }
        if (strtoec(&(Jionr.Type), argv[1]) < 0)
        {
            return ;
        }
        if(strtoec(&(Jionr.channel), argv[2]) < 0)
        {
            return;
        }
        if(strtoec(&(Jionr.ency), argv[3])<0)
        {
            return;
        }
        strepy((char *)Jionr.Ssid, argv[4]);
        if(strtoec(&(Jionr.rssi), argv[5])<0)
        {
            return;
        }
    }
}
```

```
    }  
    else  
    {  
        ErrorCode=*(arg+2);  
        printf("reciver error At cmd resp ERROR Code:%d",ErrorCode);  
    }  
}
```

#### 6.3.7.2 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    WiretFlash;  
    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)
{
    WiretFlash='!';
}
else
{
    WiretFlash=NULL;
}
sprintf(Atbuf,"%s%c%d,%d,\"%s\"\n",ATCmd,WiretFlash,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);
}
}

```

#### 6.3.7.3 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 (strtoull(Scanresult[arggroupc].BSSID, 6, argv[0]) < 0)
    {
        printf("Error Bssid\n");
        return ;
    }
    if (strtoul(&(Scanresult[arggroupc].Type), argv[1]) < 0)
    {
        return ;
    }
    if(strtoul(&(Scanresult[arggroupc].channel), argv[2]) < 0)
    {
        return;
    }
    if(strtoul(&(Scanresult[arggroupc].encry), argv[3])<0)
    {
        return;
    }
    strcpy((char *)Scanresult[arggroupc].Ssid, argv[4]);
    if(strtoul(&(Scanresult[arggroupc].rssi), argv[5])<0)
    {
        return;
    }
}
}
}
else
{
    ErrorCode=*(arg+2);
    printf("receiver error At cmd resp ERROR Code:%d",ErrorCode);
}
}
```

## 附录A 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 ASCII 格式时，模块在上电复位时需要首先将对 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，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，为什么模块复位后没有自动联网？

A: 模块启动后根据 MODE 引脚状态，强制进入配置模式，详见 4.2.1 节

Q7: 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 4.2.2.2 小节;

Q8: 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 4.2.2.4 小节;

Q9: 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, 请复位模块, 并确保 LED1 和 LED2 同时点亮后重新搜索

Q10: 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

Q11: 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>

Q12: 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；

Q13：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

Q14：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 有 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；

**Tel: 13420973470 QQ: 370925732 龚生**

邮箱 [David@hlktech.com](mailto:David@hlktech.com)



附录B 文档修订记录

版本号	修订范围	日 期
1.00	Draft 版本	2010-9-10