



# WIFI (802.11b/g) 串口服务器应用说明

## USR-WIFIMOD-II

硬件版本: V3.0.0

文件版本: V2.4.4



Wi-Fi 串口服务器是一款用来将串口数据和 WIFI 网络数据互相转换和传输的设备，可以工作与 TCP Server, TCP Client 和 UDP 模式实现透明传输，也可以工作在 Sweb 模式通过浏览器网页实现复杂交互，使用简单，工作可靠。

关键字: WIFI, RS232, Wi-Fi, 802.11b/g, 串口服务器, 模块

济南有人科技有限公司为您提供完整的物联网解决方案，软件硬件，服务器，您只需要说明白需求，剩下的我们来，欢迎联系，0531-55507297 13864129404

注:

- ◆ 产品持续改进，本文档也在更新，请参阅文档尾部的更新记录；
- ◆ 串口服务器及其核心模块功能基本一致，如非额外注明，皆适用于本文档描述；

## 目 录

第一章、概述.....	3
第二章、快速进入.....	4
2.1 如何在出厂设置下配置 wifimodII.....	4
2.2 如何重置出厂设置.....	6
2.3 数据通道测试.....	7
第三章、无线网络模式选择和部署.....	10
3.1 点对点连接.....	10
3.2 基础模式连接.....	11
第四章、配置与使用.....	13
4.1 无线配置.....	13
4.2 工作方式配置.....	15
4.3 网络设置.....	17
4.4 串口配置.....	18
4.5 固件升级.....	20
4.6 配置维护.....	20
4.7 名称设置与节点发现协议.....	21
第五章、应用实例.....	23
5.1 应用实例之 UDP 模式.....	23
5.2 应用实例之 TCP Server 模式.....	24
5.3 应用实例之 TCP Client 模式.....	25
5.4 应用实例之公网 (Internet) 应用.....	26
第六章、工作模式与性能调整.....	28
6.1 UDP 模式特征和丢包处理.....	28
6.2 TCP 模式特征与速率调整.....	28
6.3 串口数据转发性能调整.....	29
6.4 Tcpc 模式的 keepalive 特性和重连特征.....	29
6.5 TcpS 模式与富客户端 Flash 或 SilverLight 应用.....	29
6.6 ADHOC (点对点) 模式下的 DHCP 功能.....	30
第七章、wifimodII 网页服务器专题.....	31
7.1 Sweb 概述.....	31
7.2 Sweb 模式核心知识点.....	34
7.3 Sweb 模式的页面访问.....	35
7.4 向串口上传文件.....	36
7.5 如何与串口交互数据.....	37
7.6 如何与串口交互二进制数据.....	39
7.7 高级内容 MIME 类型.....	42
第八章、SWEB 实践.....	43
8.1 预备工作.....	43
8.2 从最简单的网页开始.....	44
8.3 放置图片, 添加链接.....	46
8.4 应用 AJAX, 与串口动态交互.....	46
8.5 使用 JS 库和样式表.....	51
8.6 修改页面, 上传至用户空间.....	52

## 第一章、概述

WIFI 转串口服务器是一款用来将串口数据和 WIFI 网络数据互相转换和传输的设备，可以工作与 TCP Server, TCP Client 和 UDP 模式实现透明传输，也可以工作在 Sweb 模式通过浏览器网页实现复杂交互，使用简单，工作可靠。

产品特性:

1. 符合 IEEE 802.11b/g 标准，支持 AdHoc 点对点(不需要无线路由器，自组网)、基础模式
2. 支持数据安全标准 WEP64/128、WPA/RSN(WPA2)PSK
3. 支持静态 IP 设置，支持 DHCP 获取 IP，支持 AutoIP 地址获取
4. 透明传输支持 TCP Server、TCP Client 、UDP 三种工作模式
5. 特有的 Sweb 模式，模块内存储网页，通过通用的 HTTP Post 方法与串口进行数据交互
6. 支持从串口获取用户数据和文件，支持浏览器上传和下载串口数据，支持 AJAX
7. 1M 字节用户可用的空间，存放网页和数据等
8. 可以不通过无线路由器，自己组建网络，供其他 wifi 设备(手机或者笔记本电脑)访问
9. RS232 波特率 9600~115200，数据位和奇偶校验可设置
10. 通过网页配置各工作模式和参数，登录配置页面有用户名和密码认证，保障安全
11. 支持浏览器在线升级，增加功能和修正 Bug
12. 提供 TCPIP socket 编程的例子程序和 SWeb 网页设计例子，提供技术支持

物理与电器特性:

- 外型尺寸: 80mm x 84mm (不含天线)
- 储藏温度: -40~85 度
- 工作温度: -10~65 度
- 输入电源: 4.7v ~ 6V, 典型值 5V
- 最大电流: 300MA (54Mbps TX)
- 工作频段: 2.400~2.494Ghz
- 工作通道: 14 Channels
- 传输率: 1 2 5.5 6 9 12 18 24 36 48 54 Mbps
- 接收灵敏度: 10dBm
- 发送功率: 14dBm (1/2/5.5/11 Mbps)  
12dBm (6/9/12 Mbps)

装箱清单:

1. wifi 转串口服务器 1 台
2. 串口延长线(母对母,交叉) 1 条
3. 5V 电源适配器 1 个
4. 资料光盘 1 张



## 第二章、快速入门

### 2.1 如何在出厂设置下配置 wifimodII

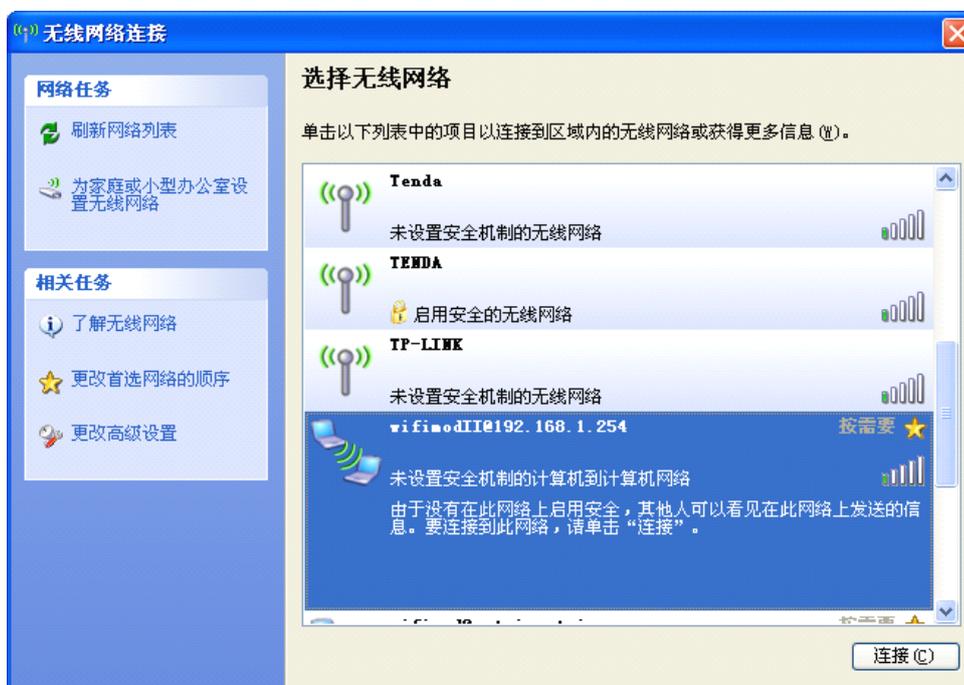
WifimodII 出厂配置如下：

无线配置：

点对点连接，ssid（网络名）为：“wifimodII@192.168.1.254”，名称中可以看出其缺省的 IP 地址为“192.168.1.254”，缺省配置为固定 IP 地址。

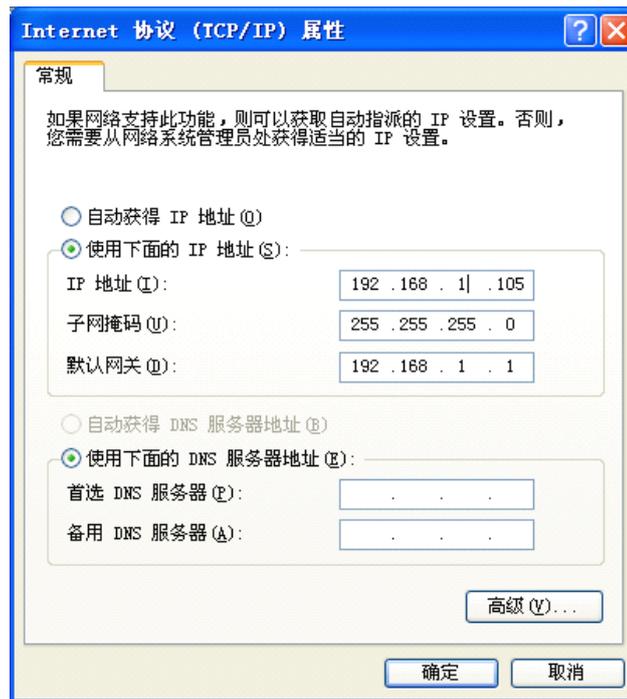
wifimodII 处在出厂设置的特征是上电后，Serial 和 Net 两个 LED 灯会以 2 秒为周期闪动。

若您使用带有 wifi 功能的笔记本电脑，在 XP 系统下，可以使用 windows 自带的无线扫描功能，找到 wifimodII 节点，显示如下图：



选中图中的节点，点击“连接”按钮便可以连接上模块了。计算机会提示您已连接，wifimodII 模块上 WLAN 灯也会亮起。

此时，需要改动您的计算机的 IP 地址，使其能够与 wifimodII 处在同一个网段中，建议您设置为 192.168.1.105（该 IP 为 wifimodII 出厂设置的默认目标 IP），配置窗口如下图：



配置 IP 完毕后，便可以登录 wifimodII 的界面了。打开浏览器，在地址栏输入：“192.168.1.254”，回车，便可以看到浏览器提示用户登录：输入缺省用户名 admin 与密码 admin 即可。



确认之后，便能够显示 wifimodII 内嵌 webServer 提供的网页内容了。



其中含有若干栏目，用户可以点击 查看状态、设置功能。  
具体的设置细节，请参加其他相关章节。

## 2.2 如何重置出厂设置

wifimodII 如果配置失误、或者忘记用户名密码导致无法连接，可以使用 wifimodII 的恢复配置流程，该操作将允许用户以缺省配置启动 wifimodII，用户登录后可以设置、保存新的配置。

工作原理：

wifimodII 在启动初期三秒，会配置串口为 115200，并监视串口输入，以判断用户是否需要恢复出厂配置，如果用户输入：“wifimod:RSTOR” 13 个字符（无双引号），本次启动便会使用出厂配置。但是如果断电再次启动，会以最近一次保存过的内容为启动配置。

操作之前，用户需要稍作准备，如下：

首先打开超级终端（或者其他串口调试工具），如下图配置，主要保证使用 115200 波特率，以及无硬件流控。



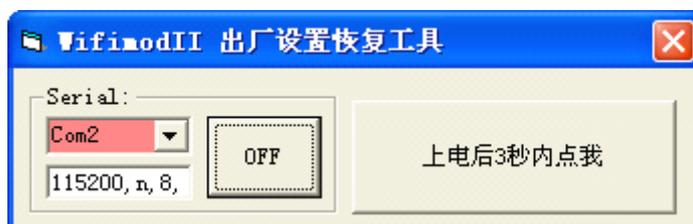
确定后，终端窗口待用。

之后，请拷贝“**wifimod:RSTOR**”双引号中间的文字，以便于可以复制进入终端，实现快速输入。

下面是具体的操作：

用户操作	wifimodII 动作
使用交叉串口线（2\3 交叉）连接电脑和模块；	
打开超级终端，配置完毕。	
拷贝字符串“ <b>wifimod:RSTOR</b> ”	
为模块上电	
	上电时，Serial 串口指示灯会点亮 3 秒，表示串口已经打开，等待用户输入。
迅速将“wifimod:RSTOR”字符串粘贴入超级终端的输入框（不会回显内容）。	
	得到正确的输入后，Serial 串口指示灯会立刻灭去，之后，会看到“Serial”和“Net”灯同步以两秒为周期闪动，表示进入出厂设置。

用户也可以使用随带的工具软件辅助恢复：

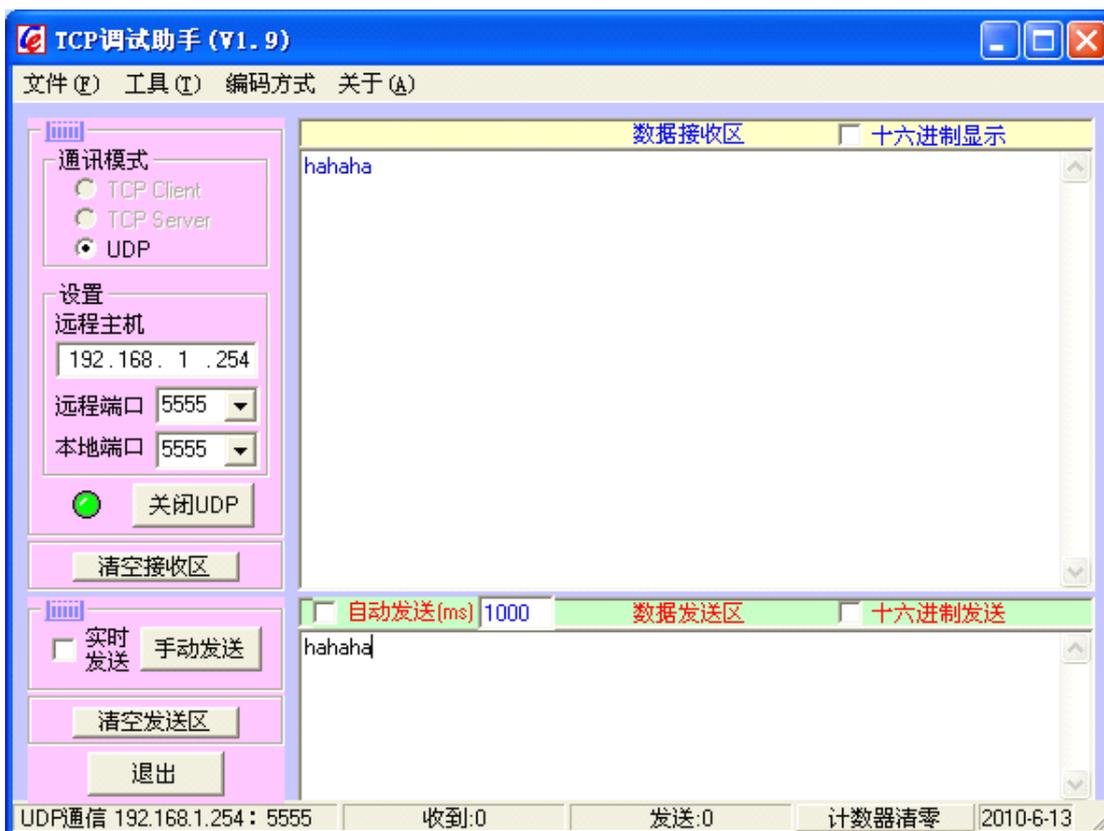


注意，回归出厂模式的操作并不删除之前用户已经保存的配置，如果不做新的配置保存，再次复位模块将依然使用最近一次保存的设置。

## 2.3 数据通道测试

当模块处在出厂配置下，用户笔记本在无线网络上连接到模块之后，可以将串口与模块相连，进行数据通讯测试。

需要运行“tcp 调试助手”小软件，如下图：

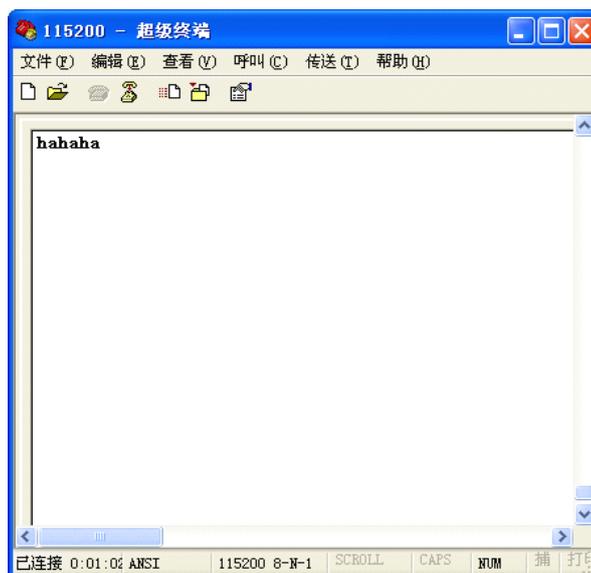


设置模式为 UDP；  
设置远程主机为：192.168.1.254；  
远程、本地端口皆设为：5555；  
点击“开启 UDP”按钮，绿灯亮起。

连接好串口线之后，用户可以打开超级终端，按出厂缺省配置，设置为 115200 波特率以及无流控。

“TCP 调试助手”软件中发送数据，可以看到超级终端上有数据出现，至此，说明数据通道打通。当然，也可以在串口超级终端敲击字符，在网络侧也会有相应的字符反应。

如下图：



WifimodII 设置为 TCP 服务器模式，测试过程与上面类似，“TCP 调试助手”软件需要设置为客户端，

发送数据前需要先连接。

若是设置为 TCP 客户机模式，需要“TCP 调试助手”软件预先设置为服务器端，并且开始侦听；模块配置完毕复位重启后，可以看到“TCP 调试助手”显示有连接接入，这时就可以进行数据传输测试了。

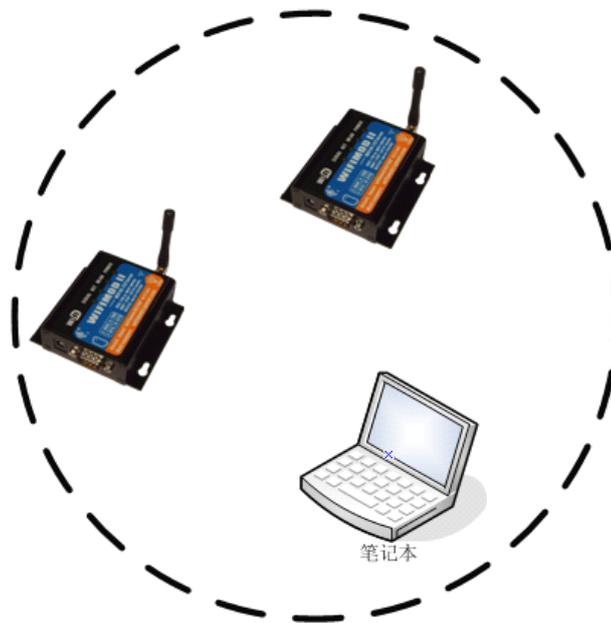
如果网络断开，wifimodII 会一直重复尝试连接直到连接上。

## 第三章、无线网络模式选择和部署

无线网络应用模式分为“点对点连接”或者“基础模式连接”wifimodII 同时支持两种模式。

### 3.1 点对点连接

该模式是一种低成本、方便连接方式。



点对点连接部署图

用户可以不依赖于无线路由器，直接使用笔记本连接模块。该模式应用成本低廉，但没有无线路由的协调，在 IP 分配上只能选用静态 IP 或者 AutoIP（配置页面有相关说明）。

其特点罗列如下：

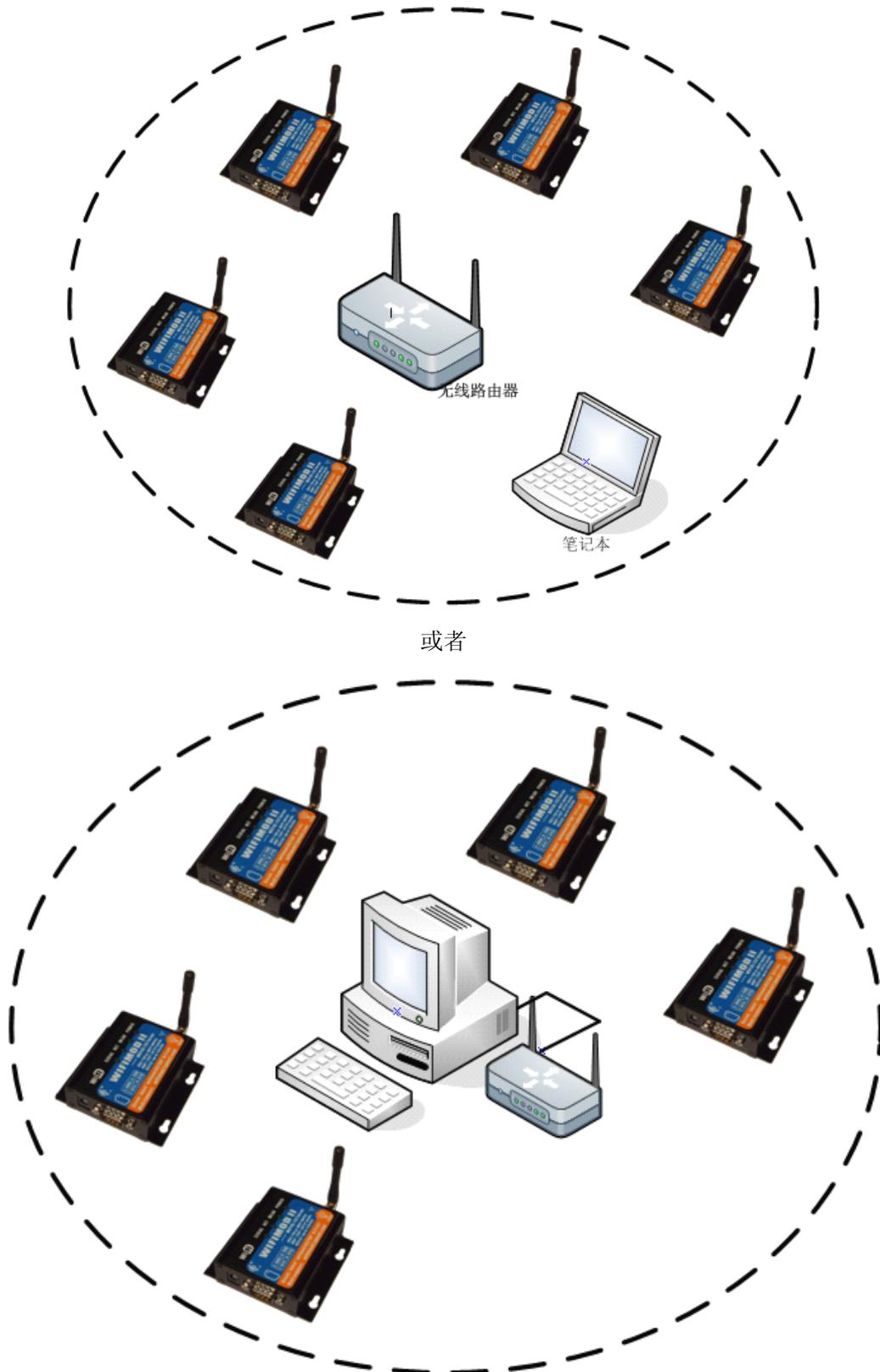
- 1、无需路由器；
- 2、可以支持多模块接入；
- 3、需要静态设置或者 AutoIP 自适应协调机制设置 IP；

采用点对点模式布局无线环境，其优势在于不需要无线路由器参与，在设备数量有限的情况下简单易行；

比如，模块出厂即为点对点模式，模块上电后，计算机可以直接扫描并接入模块创建的点对点网络，对其进行访问。

当设备端数目比较多，布局开阔时，或者用户已经部署了无线局域网，那么可以采用下面介绍的 基础模式连接。

### 3.2 基础模式连接



基础模式连接是以无线路由为核心的连接组网方式，借助于无线路由器，wifimodII 可以部署的更为灵

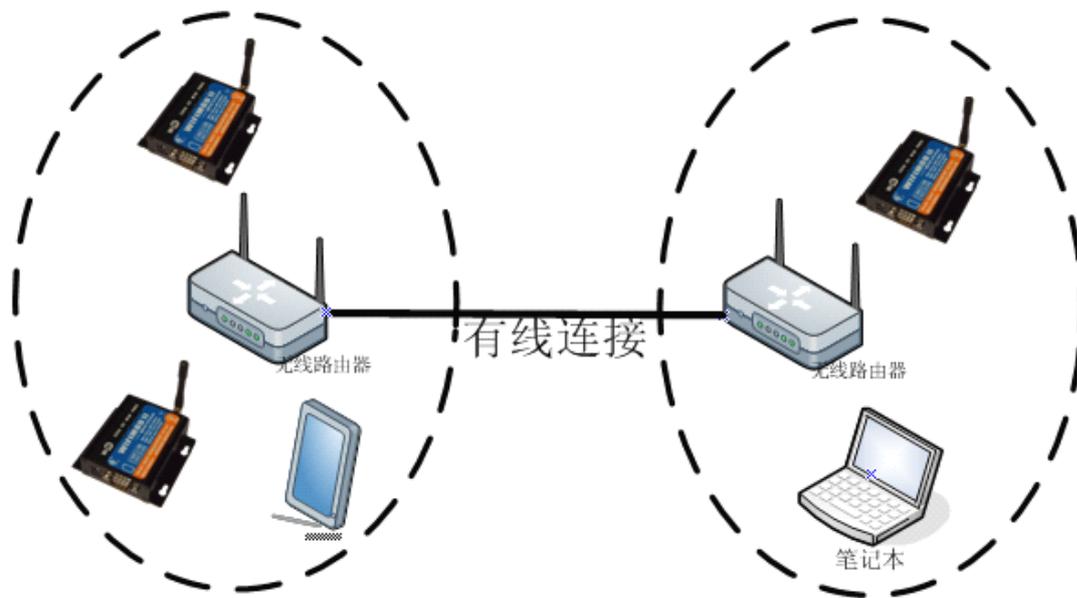
活，距离上也可以藉由路由器网桥功能予以延伸。

其特点罗列如下：

- 1、可以通过无线路由器扩展信号覆盖范围；
- 2、IP 地址可以通过路由器的 DHCP 功能动态分配；
- 3、可以设置高等级 WPA2 加密，保障网络数据安全；

一般家庭使用无线路由器组网的就是属于这种方式。

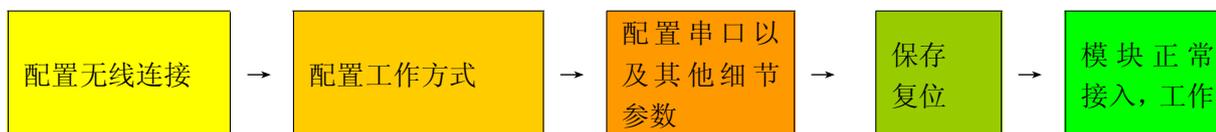
此外，在基础模式下，如果需要拓展覆盖区域，可以使用无线路由器桥接功能，拓扑图大致如下图。



本文档不描述这部分内容，用户可以查阅路由器说明和相关资料。

## 第四章、配置与使用

配置 wifimodII 的流程大致是：



### 4.1 无线配置

为 wifimodII 配置无线参数，最终目的是使模块能够接入用户无线网络中去。

#### 无线设置：

无线网络名称：	<input type="text" value="wifimodII@192.168.1.254"/> (max 32 ASCII)
接入模式：	<input type="text" value="点对点(Ad-Hoc)"/> ▼
RSSI低门限 (漫游尝试)：	<input type="text" value="0"/> (-90~-50, 0为禁用), 专家模式：请在指导下设置该参数

“RSSI 低门限”参数用于使能、调整模块的漫游功能。该功能在 2.4.2 版本后提供。

漫游 (Roaming) 是指：在同 SSID、多 AP 构造的 WDS (无线网络分布式系统) 环境中，模块可以根据信号强度适时的切换其所关联的 AP，保障数据传输。

设置该参数值，开启漫游功能后，模块会在信号降低到该阈值时尝试寻找信号更好的同名 AP，如果找到便会主动切换，该过程不影响和之前已接入 AP 的通讯，减少切换过程对通讯的影响，很大程度上提升模块移动时的传输效果。

参数设置为 0 时，禁止该功能；

该功能仅仅在基础模式下工作，AdHoc 网络中即使设置也不起作用；

单 AP 环境中，可以禁止该功能，减少资源占用；

模块与 AP 相对静止的应用环境中，可以禁止该功能，减少资源占用；

**专家指导下使用：若不清楚该功能对使用的影响，请勿设置该参数！！**

#### AdHoc 点对点模式：

模块出厂即为点对点模式。

在配置页面 “setup.htm”，配置为点对点网络的细节如下图：

**无线设置:**

无线网络名称:	<input type="text" value="wifimodII@192.168.1.254"/> (max 32 ASCII)
接入模式:	点对点(Ad-Hoc) ▼

**加密设置:** WEP ▼

WEP位数选择:	WEP128(128Bit) ▼
wepkey index:	<input checked="" type="radio"/> 0 <input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3
wep_key0:	<input type="text" value="12345678900000000000000000000000"/>
wep_key1:	<input type="text" value="23456789010000000000000000000000"/>
wep_key2:	<input type="text" value="34567890120000000000000000000000"/>
wep_key3:	<input type="text" value="45678901230000000000000000000000"/>

提示: +

**点对点模式可以选择WEP加密**

**Infra 基础模式:**

在配置页面 “setup.htm”，配置为基础网络模式的细节如下图:

**无线设置:**

无线网络名称:	<input type="text" value="wifimodII@192.168.1.254"/> (max 32 ASCII)
接入模式:	基础模式(Infrast) ▼

**加密设置:** WPA\RSN PSK ▼

psk密钥:   显示

**基础模式支持 WEP、WPA和WPA2 PSK加密**

无线设置配置完毕后，保存设置，并复位模块，模块重启后，会看到 **WLAN 灯快速闪烁**，而后一直 **保持常亮**，这说明无线网络已经连接成功。

## 4.2 工作方式配置

在配置页面 “setup.htm”，配置工作模式的细节如下图：

**工作模式设置：**

工作模式：	SWEB(网页)模式
目标IP地址：	192 . 168 . 1 . 105
目标域名：	yizuoshe.com (0~32 ASCII)
目标端口号：	55555 (0~65535)
TCP连接动作：	无动作
Ping目标IP地址：	0 . 0 . 0 . 0

sweb.cgi 访问认证使能：

### 配置项说明：

配置项	说明	备注
工作模式	共四种模式：UDP、TCPc、TCPs 和 SWEB	
目标 IP 地址	连接目标 IP；	
目标域名	在 TCPc 模式下，目标 IP 设置为 “0.0.0.0” 时，该参数通过域名定义连接目标；	该配置项在 2.3.3 版本之后支持
目标端口号	连接目标端口号；其可以设置为 0，表示特殊含义。细节见 “工作模式详解”。	
TCP 连接动作	表示在 TCP 模式（客户机或服务器）下，连接建立后模块首先的动作（仅执行一次）。 <b>“发送组名称和名称”</b> 选项使模块可以主动向连接对方发送节点 ID 信息（16Bytes 组名称、16Bytes 名称）。 例如在模块在内网访问 Internet（公网）中的服务器时，服务器无法通过 IP 地址区分客户端，选中该功能，模块在连接到服务器之后会立即主动发送一个 32Bytes 的数据段（16bytes 组名称和 16bytes 名称），帮助服务器辨识模块。该数据只发送一次，完成后，模块依然是透传模式。	该配置项在 2.3.3 版本之后支持
Ping 目标 IP 地址	设置该 IP 地址，模块每隔 10 秒便会 ping 该 IP，连续 6 次未得到回应，便会复位自身。该设置用于保证模块长时接入网络的可靠性。例如此处可以设置为无线路由器的 IP 地址（例如 192.168.1.1）。此外还要确定目标未屏蔽 ping 功能。设置为 0.0.0.0 时关闭该功能。	该配置项在 2.4.2 版本之后支持
sweb.cgi 访	Sweb.cgi 为用户访问串口的接口，该设置指定是否需要用户登录	2.4.3 版本提供

问认证	认证。	
-----	-----	--

## 工作模式详解：

### UDP 模式：

通过设置“目标 IP 地址”和“目标端口号”，模块可以实现 **一对一绑定** 或者 **一对多过滤** 两种应用方式。

■ **一对一绑定** 要求设置“目标 IP 地址”和“目标端口号”两个参数为目标地址，两个参数皆需要设置为有效值（非 0）；模块将仅仅与预设目标一对一交互，该配置不能接收广播报文；

■ **一对多过滤** 可以实现对网络目标地址的过滤，即，“目标 IP 地址”和“目标端口号”作为过滤条件存在，任何一个设置为非空（非 0），网络端接收到数据后，将根据此参数确定是否和来源地址相匹配，如果相符，则转发数据到串口，若否，则放弃该报文。

例如，需要模块接收所有 192.168.1.105 的数据包，那么只需要设置“目标 IP 地址”为“192.168.1.105”，留空“目标端口号”即可。同理，若是需要接所有来源端口号为 55555 的报文，仅设置“目标端口号”为 55555，留空“目标 IP 地址”，便可以接受所有 IP 地址来源于 55555 端口的 UDP 数据报文了。

**串口侧接收到数据，将转发到最近一次访问模块的目标**，如果最近没有目标向模块发送过数据，那么串口数据将会被抛弃。

UDP 模式目前不支持通过域名解析对方 IP。

### TCP 服务器模式：

服务器模式为被动方式，模块不会主动连接目标。选择该模式，“目标端口号”会自动设置为缺省值 0，表示允许任意 IP 地址客户端连接模块；（同时仅允许一个连接接入）。

若用户手工将“目标端口号”设置为非 0，则服务器将仅仅允许“目标 IP 地址”定义的客户端连接。

TCP server 模式下，端口设置为 **4530**，将使能 SilverLight Socket 连接认证机制，用于支持用户浏览器中 SilverLight 程序与串口数据交互。

TCP server 模式下，端口设置为 **4540**，将使能 Flash XMLSocket 连接认证机制，用于支持用户浏览器中 Flash 与串口进行数据交互。

关于这两项内容，请参考“工作模式与性能调整”章节。

### TCP 客户机模式：

“目标 IP 地址”或者“目标域名”用于指示连接对象。若目标 IP 设置为 0.0.0.0，则模块会尝试解析“目标域名”获取 IP（需设置 DHCP 模式自动获取 DNS 或者静态设置 DNS，见“网络设置”部分）。

模块上电后会主动连接服务器，尝试 10 次未成功后会将重新连接的时间间隔从 15 秒调整到 60 秒；连接后断开，亦会重复该过程。

### SWEB 模式：

使用方法详见“sweb.htm”页面和 sweb 相关说明章节。

### 备注：

“目标域名”仅需要填写目标服务器的主域名，无需包含分支目录，例如“http://www.domain.com/sub/”可以填入“www.domain.com”。请勿使用汉字和双引号。

## 4.3 网络设置

配置网络参数的界面如下图：

**网络设置：**

获取IP方式：	静态设置
本地IP地址：	192 . 168 . 1 . 254
子网掩码：	255 . 255 . 255 . 0
网关IP地址：	192 . 168 . 1 . 1
DNS地址：	192 . 168 . 1 . 1

### 配置项说明：

配置项	说明	备注
获取 IP 方式	支持 3 种模式获取 IP 地址，分别是： <b>静态设置：</b> 用户主动为模块分配 IP，模块出厂时默认便是静态 IP，地址为：192.168.1.254； <b>DHCP 分配：</b> 模块需要通过 DHCP 服务器获取 IP 地址，一般无线路由器（AP）均有该功能，适用于多模块同时使用，管理者无需费心独立为其设置 IP； [1] <b>AutoIP 分配：</b> 该模式使用情况比较特殊。模块在没有 DHCP 服务器的情况下，会随机使用一个 169.254.0.0 ~ 169.254.255.255 之间的 IP 地址，一般模块不会遇到 IP 重复；若模块探测到有 IP 重复，会自动更换 IP。 不建议客户使用该模式。	
本地 IP 地址：	静态设置的 IP 地址	
子网掩码：		
网关 IP 地址：	一般设置为路由器的 IP	
DNS 地址：	如果使用了域名解析的功能，该处需要设置为 DNS 服务器的 IP 地址，一般设置为路由器的 IP 即可；	

注意：

[1] 如果在“无线设置”中选择了“adhoc 模式”，在网络设置中又选择了“DHCP 方式获取 IP”，第一个进入 adhoc 网络的模块会主动创建 DHCP 服务器，为后来的模块提供 IP 分配服务，细节请参考“ADHOC 模式下的 DHCP 功能”章节

## 4.4 串口配置

配置串口参数的界面如下图:

串口设置:

串口端口号:	<input type="text" value="55555"/>	(1~65535)
接收空闲时间 (HoldTime):	<input type="text" value="200"/>	(10~60000 ms)
接收打包尺寸 (LoadMax):	<input type="text" value="1460"/>	(16~1460 bytes)
串口波特率:	<input type="text" value="115200"/>	
传输字位长:	<input checked="" type="radio"/> 8Bits <input type="radio"/> 9Bits	
停止位长:	<input type="radio"/> 0.5Bit <input checked="" type="radio"/> 1Bit <input type="radio"/> 1.5Bits <input type="radio"/> 2Bits	
奇偶校验:	<input checked="" type="radio"/> 无 <input type="radio"/> 奇校验(Odd) <input type="radio"/> 偶校验(Even)	
流控选项:	<input type="text" value="无流控(0)"/>	

### 配置项说明:

配置项	说明	备注
串口端口号	<p>是指 wifimodII 设备上串口占用的缺省网络端口；该参数不可以被设置为 <b>12345</b>，该端口被用于节点发现协议；在 TCPc（客户机）模式下，该参数设置不起作用，wifimodII 在连接服务器时会自行选择合适的端口；</p> <p>TCP server 模式下，端口设置为 <b>4530</b>，将使能 SilverLight Socket 连接认证机制，用于支持用户浏览器中 SilverLight 程序与串口数据交互；</p> <p>TCP server 模式下，端口设置为 <b>4540</b>，将使能 Flash XMLSocket 连接认证机制，用于支持用户浏览器中 Flash 与串口进行数据交互；</p> <p>若不使用 SilverLight 或者 Flash 的网络连接功能，请避开使用 <b>4530</b> 和 <b>4540</b> 端口，以避免引导认证服务器加重系统负担；</p>	
接收空闲时间 (HoldTime)	定义空闲等待最长时间；串口缓冲数据，如果空闲超时，则打包转发；	
接收打包尺寸 (LoadMax)	定义串口缓冲容限；超限则打包转发；	
流控选择	<p>支持</p> <ul style="list-style-type: none"> <li>◆ 无流控</li> <li>◆ RTS（OUTPUT 接收允许通告）使能</li> <li>◆ CTS（INPUT 发送允许感知）使能</li> <li>◆ RTS 与 CTS 同时使能</li> </ul>	

	<p>◆ RS485 模式（该选项在 2.1.20 版本后支持）</p> <p>RS485 包含在流控选项中主要是因为 RTS 引脚充当 485 切换方向的信号，如果硬件上包含 485 接口芯片且作出正确的连接，可以选择该项以使能 485 的功能。</p> <p>注意：RTS 引脚缺省输出低电平。</p>	
传输位长	<p>为字节位长和校验位的总和。</p> <p><b>如果选择了校验，那么此处应该选择 9bits</b></p>	
其他	.....	

### 串口数据转发工作细节：

wifimodII 从串口接收的数据会预先存放在 FIFO 先入先出缓存（4Kbytes）中，网络端根据设置，定时、定量的从该缓存中取出数据转发到无线网络侧。

FIFO 中若已进入数据，那么其滞留的时间最长为 HoldTime 设置值；在滞留时间未到时，若积蓄长度超过 LoadMax 定义的值，现有数据也会被加以转发。

wifimodII 从无线网络侧接收到的数据也会预先存放到 FIFO（2Kbytes）中，只要 FIFO 中存在数据，模块将保持始终向串口发送，直到 FIFO 空为止。

如何提高转发效率，保证稳定性，减少数据丢失，请查看“工作模式与性能调整”章节。

## 4.5 固件升级

wifimodII 支持固件升级，“setup.htm”页面中，如下图所示可以上传升级文件。



升级操作有风险，需要保证文件的完整性，因此，提供了一个验证升级文件的功能，用户可以预先验证一下升级文件，以确保其正确性。如果，检查通过（返回 OK 页面），则可以使用下面升级固件功能，顺利上传完成后亦会返回 OK 页面。

此时，断开模块电源和串口连接，重新上电。静候 3 分钟左右，模块会自动完成升级过程，升级过程不会影响功能配置。升级完成后，模块会自动复位，重新接入到目标无线网络，可以通过 wlan 灯的特征加以识别升级是否完成。

## 4.6 配置维护



用户可以将当前模块的配置文件下载到本地，保存为本地文件。当然，也可以保存的配置文件恢复到模块。复位后新配置生效。

注意，用户名和密码设置无法备份与恢复。

（2.3.4 版本支持）

## 4.7 名称设置与节点发现协议

wifimodII 可以设置节点名称和组名称，这两个设置作用于节点发现协议。名称配置界面如下图：

名称设置：

名称：	wifimodII	(3~16 ASCII)
组名称：	wifimodIIGroup	(3~16 ASCII)

提示：+

wifimodII 节点名称设置组名称设置界面

节点发现协议：

简而言之，节点发现就是“广播询问，相应节点回复”的过程。

wifimodII 在 12345 端口开通了一个 UDP 发现协议服务器，可以响应查询该端口的报文。

查询报文与回复报文结构相同，如下表：

字段名称	空间	说明	备注
GroupName	16 字节	组名称	可以在查询报文中设置
NickName	16 字节	昵称	可以在查询报文中设置
IP	4 字节	IP 地址	回复报文中设置
mac	6 字节	Mac 地址	回复报文中设置
SNR	1 字节	SignalNoiseRatio	信噪比 (db)
NF	1 字节	NoiseFloor	负值
SN	8 字节	序列号	
Version	12 字节	固件版本信息	
servmode	4 字节	工作模式	2.4.4 版本之前为 8 字节
servport	2 字节	工作串口的端口号	2.4.4 及其之后添加
Padding	2 字节	填充字节，待用	
stat_sum	64 字节	其他状态信息	

用户发送查询报文时可以提供名称信息、组名称信息，节点接收到广播报文，根据名称设置，可以选择回复，也可以保持静默。这种功能可用于过滤节点，实现节点的分组化。

节点响应的条件如下：

响应的条件	说明
查询报文中 GroupName 为 "wifimodII"，且 NickName 为空	用于扫描全部 wifimodII 节点
查询报文中 GroupName 与本地 GroupName 相同，且 NickName 为空	用于扫描一个组
查询报文中 GroupName 与本地 GroupName 相同，且 NickName 与本地 NickName 相同；	用于明确寻找一个特定组中的特定节点

查询报文中 GroupName 为空，但 NickName 与本地 NickName 相同	以名称扫描一个节点
---	-----------

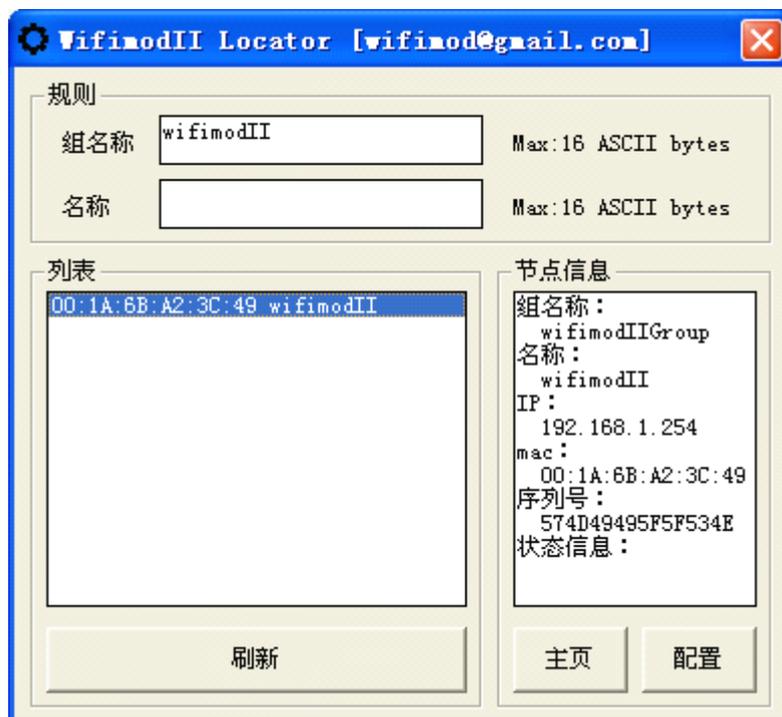
使用该服务，用户即使不知道节点的 IP 地址，也可以通过名称找到网络上响应的 wifimodII 节点。

例如，一个宾馆，其中每个房间都设置一个节点，那么可以使用楼层作为组名称，假设以“F1、F2...”命名，房间号直接作为节点的名称，例如，“101、102、201、202...”等。

wifimodII 被设置为 DHCP 自动获取 IP 之后，我们无法知道某个楼层某个房间的节点确切的 IP 地址，也就没法去访问它。

使用发现协议，不仅可以一次扫描出所有的节点，得到其 IP 地址；也可以分楼层扫描，或者直接询问房间号。

下图是使用 VB 开发该协议的一个实例。

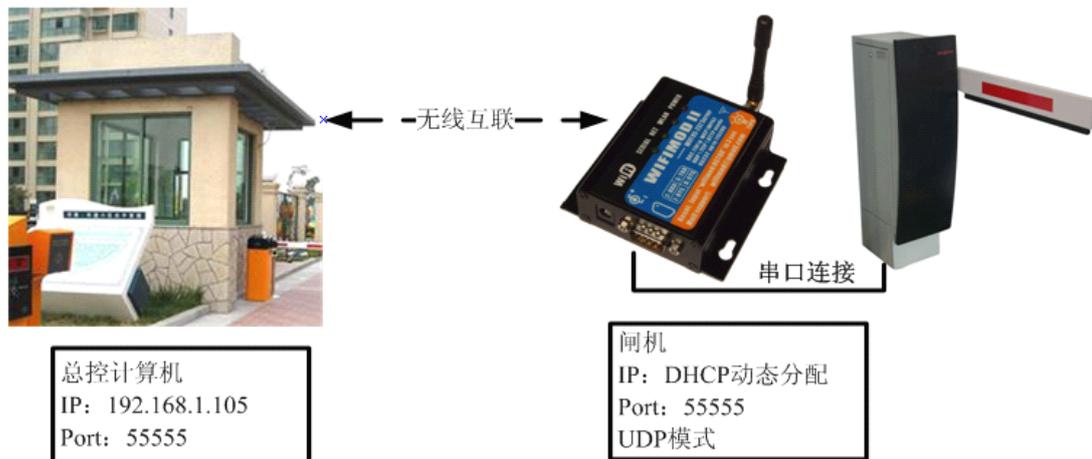


wifimodII 节点发现工具

该程序已经可以实现组合式扫描，可以直接应用与节点发现。此外，其源代码开放给用户，用户可以在此基础之上定制自己需要的功能，将发现过程纳入自己的管理体系中。

## 第五章、应用实例

### 5.1 应用实例之 UDP 模式



某小区部署无线网络实现闸机设备智能联网

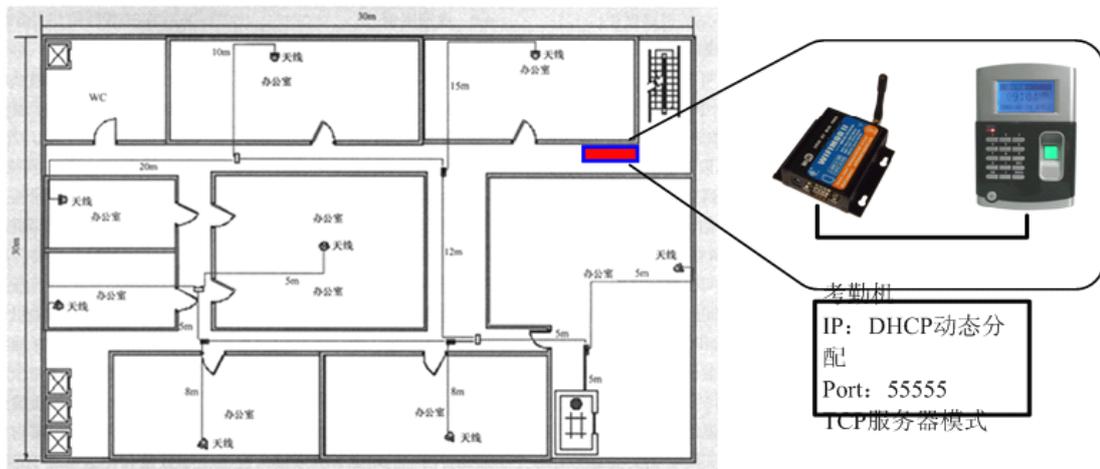
上图所示的应用场景中，车辆闸机在进出车辆刷卡时，需要向总控室发送刷卡数据（长度小于 1K），总控室计算机得到信号，经过处理后，发送信号给闸机开闸放行。

上述流程中，传输的数据量很少，仅仅是几个报文的交互；因此，可以采用 UDP 模式。没有 TCP 连接的额外开销，实时性也能得到保证。

注意：

虽然几率较小，但是无线传输会有丢失 UDP 报文的情况，因此，闸机需要实现发送重试机制，发送刷卡信号给总控室计算机之后，应该短时间内得到回复，否则应该尝试再次发送。

## 5.2 应用实例之 TCP Server 模式

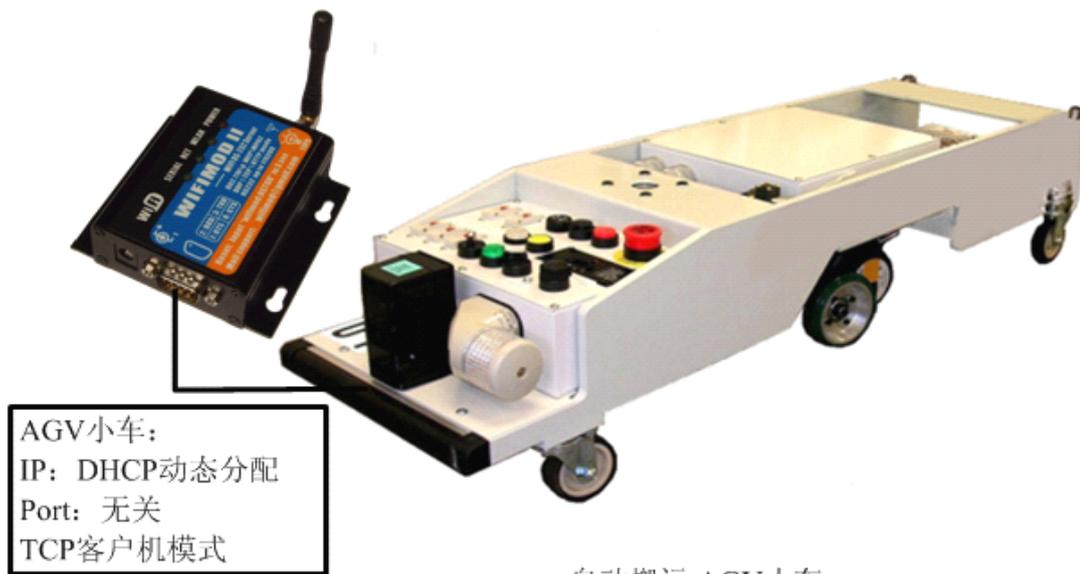


某公司办公区域设置考勤机接入无线网

上图展示的是 wifimodII 将串口指纹考勤机接入无线网的应用场景。指纹考勤机本身具有存储能力，员工的报到完毕后，数据存放在机器内部，考勤人员需要访问它获取其中的报到信息。

考勤机为被动访问，而且需要向访问者提供考勤数据（约 1000~2000Kbytes），因此，设置为 TCP 服务器的方式是合理的，访问者登录服务器，请求下载数据，下载完成后注销。TCP 连接保障可以保证数据的完整和正确性。

## 5.3 应用实例之 TCP Client 模式



自动搬运 AGV 小车

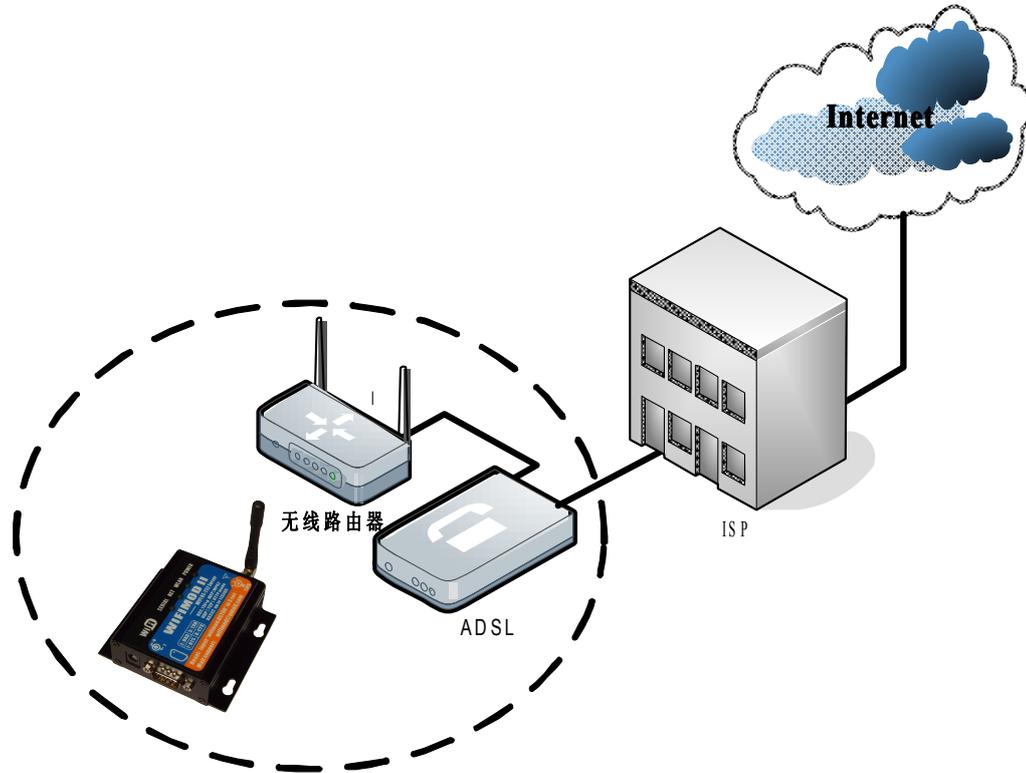
智能仓储系统中 AGV 无人搬运车无线联网

上图展示的是 wifimodII 在物流系统中为无人搬运车提供无线接入的应用场景。AVG 小车在开机之后，会自动相应响应控制系统的指派，完成搬运工作。

小车开机上电后，应该主动与总控计算机相连，因此，采用 TCP 客户机模式是比较合理的安排。wifimodII 上电后，会主动连接总控制端；总控制界面上，也可以非常清楚的看到主动登录的小车节点，查看其状态，为其指派任务。若小车出现故障，如掉电，总控也可以察觉到丢失连接，进而可以及时解决问题。

## 5.4 应用实例之公网（Internet）应用

wifimodII 不仅适用于局域网内建立组网，也可以通过无线路由器连接远程服务器，建立基于 Internet 的组网方式。



系统框架图

这种组网方式利用 wifi 网络和宽带资源，将热点覆盖下的 wifimodII 模块连接到云端服务器上，实现远程监控应用。

**wifimodII 可以直接设置远程服务器的域名：**

通过域名访问，用户不用选择固定 IP 地址的服务器，wifimodII 支持 DNS 地址解析，实时获得域名的真实 IP。

**wifimodII 还支持连接后进行身份声明：**

模块含有“组名称”和“名称”设置选项，用户可以预设该值，为每一个模块设置不同名称。TCP 客户端模式时，使能这种特征后，一旦连接建立，模块便主动发送“组名称”和“名称”到对方。这样，服务器端便可以分辨出模块，区别对待了。这种声明过程仅仅发生在连接建立之后，完成该过程后，模块依然处在透传模式。

**名称设置：**

组名称：	wifimodIIGroup	(3~16 ASCII)
名称：	wifimodII	(3~16 ASCII)

提示：+

## 组名称、名称设置

TCP 客户机模式访问广域网服务器 实例设置参见下面图示：

**工作模式设置：**

工作模式：	TCPC(客户机)模式
目标IP地址：	0 . 0 . 0 . 0
目标域名：	domain.com (0~32 ASCII)
目标端口号：	55555 (0~65535)
TCP连接后动作：	发送名称和组名称

提示：+

说明：

如果需要域名设置，须将目标 IP 设置为 0.0.0.0；

如果需要身份声明，需要选择“发送名称和组名称”选项；

**网络设置：**

获取IP方式：	DHCP自动分配
本地IP地址：	192 . 168 . 1 . 54
子网掩码：	255 . 255 . 255 . 0
网关IP地址：	192 . 168 . 1 . 1
DNS地址：	192 . 168 . 1 . 1

提示：+

模块连接远程服务器，采用预设名称进行身份声明，无需使用 IP 地址来区分模块。若处在 AP 无线网络中，就可以不用设置静态 IP 地址，采用自动分配即可。

## 第六章、工作模式与性能调整

### 6.1 UDP 模式特征和丢包处理

鉴于无线通讯天生的易受干扰，易丢报文的特点，需要用户注意使用方式，以提升转发过程的稳定性。

UDP 模式非常灵活，不需要握手建立连接的过程，转发效率比较高，尤其对于小数据量、快速转发的情形尤其适用。

另一方面，其协议中没有提供接收确认的能力，传输过程中无论是发送还是接收都不可避免会遇到报文丢失的情况。

对于无线网络丢失报文的情况，有两种方案建议客户参考：

一、是用户可以自行设计数据接收回应的规则；

二、如果数据分段，某些情况下可以容忍数据段丢失的情况，那么可以设置串口打包尺寸（LoadMax）参数为数据段的长度，这样即使数据包丢失，也可以保证整段丢失，不会影响上下段其他数据部分。

此外，还要注意在 UDP 方式下，网络侧发送数据速度不可以超过串口波特率。wifimodII 串口发送 FIFO 长度为 2K bytes，网络侧发送数据填满 FIFO 之后，其他将会被抛弃。

### 6.2 TCP 模式特征与速率调整

TCP 协议包含接收确认和重传功能，可以保证数据完整性。该模式下，需要预先建立连接，而后才可以交互数据。

这种模式比较适合需要保证数据内容的情形，例如，下载上传数据文件。

该模式工作细节如下：

TCP 模式下（无论主从）建立连接之前，串口接收数据均会被抛弃；

TCP 客户机模式会在上电后主动连接设定的服务器，尝试 4 次，若失败或者无响应。其会等待约 1 分钟，而后重试；

连接建立后，才可以交互数据；

PC 通过无线网络向 wifimodII 发送数据，会填满 wifimodII 串口发送 FIFO 缓存，其将会回应 PC 接收窗口关闭，从而自动降低信道上的数据速率。PC 侧的底层协议栈也会“被迫”积蓄发送数据，若发送程序不停的写入，其最终会得到写入错误的信息。因此，使用 TCP 方式向 wifimodII 发送数据，可以很好的保持收发两端的“速率压力”平衡，保证数据的完整性。

当然，wifimodII 向网络侧发送数据，也需要接收端能够及时处理，否则同样的原理会使 wifimodII 串口侧接收 FIFO 填满而丢失数据。

## 6.3 串口数据转发性能调整

场景：串口设备 → wifimodII → 无线网络

wifimodII 发送无线网络报文频率有限，实测稳定发送最高可以达到 150Packet/s，按 1460bytes/packet 来算，其带宽远远大于串口 115200 速率。

Traffic	Captured	Displayed	Marked
Packets	1136	1136	0
Between first and last packet	5.116 sec		
Avg. packets/sec	222.049		
Avg. packet size	1020.819 bytes		
Bytes	1159650		
Avg. bytes/sec	226671.410		
Avg. MBit/sec	1.813		

无线网络发送速度测试统计（来自于 wireshark 截图）

但是，有时用户数据为高频率、小数据段形态，需要很高的转发反应速度（例如使用 UDP 方式），这时可以通过调小数据打包长度（LoadMax）来提高转发速度。

鉴于保证模块长期稳定工作的需要，建议用户不要使网络侧发包频率超过 50Packets/s。推荐采用数据积蓄转发的方式来完成工作，尽量利用 packet 内部 1460bytes 的载荷。

例如，用户设备串口输出 数据段长度为 32bytes，周期为 5ms。需要模块长期以 32bytes/packet @200packets/s 来转发数据，不仅转发性能无法满足，每个报文仅仅存放 32 字节，也是对无线带宽的浪费。

这时可以设置 LoadMax 为 256，这样可以将 8 个数据段放在一个报文中，数据报文转发间隔也由原先的 5ms 变为 40ms，从总体上看，数据流速一样，但是合理利用了网络数据包的负荷，也助于提升设备的稳定性。这种情况下，HoldTime 值可以调整稍大于 40ms，以防止“轧边”情况出现。

## 6.4 Tcpc 模式的 keepalive 特性和重连特征

tcpc 模式用于主动连接远端服务器，模块具有 keepalive 特征，可以在连接空闲时确保数据链路正常以及服务器没有意外崩溃。

其细节如下：连接空闲一段时间（缺省为 250 秒）后，模块会主动发送 keepalive 探测报文，如果得到回复，模块会假设服务器正常，并如此循环在连接空闲时探测服务器。如果探测报文没有得到回应，模块尝试若干次之后，会断开当前连接，进入试探接入的循环。（注：在有数据传输时 keepalive 特征不动作）

模块重连尝试的特征如下：

模块在上电之后便会进入尝试连接循环，直到连接上为止。连接断开后也同样进入这个循环。

尝试连接细节如下：检查设置目标 IP 地址是否有效，否则会尝试解析目标域名，解析成功 IP 地址后才会连接目标 IP 地址。前 10 次尝试过程间隔 15 秒，10 次之后的尝试间隔为 60 秒。

## 6.5 TcpS 模式与富客户端 Flash 或 SilverLight 应用

STcpS 工作模式使模块作为服务器的角色存在于网络上。任何 TCP 客户端都可以通过相应 IP 和端口

连接到该服务器上，这就是 C/S 架构。

其中有一种 Client 比较特殊，其就是寄居于浏览器框架下的小程序，诸如“ActiveX 控件”、“JavaApplet”、“Flash”、或者“SilverLight”。其中前两者由于其局限性不做涉及，Flash 和 SilverLight 是我们讨论的重点。这两者都具有编写网络客户端的能力，用户可以编写客户端程序，放置在模块用户空间中，以实现用“web server 部署应用，tcp Server 提供数据”的高级应用。

Flash 和 SilverLight 以其媒体表现能力强而著名，用户可以利用其开发工具设计华丽的用户界面，利用其语言编写逻辑，之后放置到模块存储空间，实现对最终用户的服务。

这种方式可以将主要的界面设计放在特定的开发工具中进行，例如，Flash 可以使用 FlashCS3（或 Flex）来开发，SilverLight 程序可以使用 VS2010 来设计。避免对 Javascript 的过分依赖。

应用这种能力的主要步骤如下：

- 1、编写 Flash、SilverLight 网络程序，注意模块提供的端口号分别只有“4540”和“4530”。
- 2、本地测试，查看小程序是否能够连接 PC 本机的 TCP 服务器；
- 3、上传小程序和相应页面到模块空间；
- 4、设置模块工作模式为 STcpS 模式，串口端口设置为“4540（Flash）”和“4530（SilverLight）”。
- 5、复位模块，访问其页面，完工。

\* 购买该功能的用户，请联系模块提供商索取更详尽的应用说明。

## 6.6 ADHOC（点对点）模式下的 DHCP 功能

ADHOC 模式（点对点）方式是一种非常简单的组网方式，无需无线路由器，多个节点可以很方便的建立网络互相沟通。

这种组网方式也决定了网络的动态性——节点可以随时加入、退出。加入后就可以和其他节点互发数据，退出便杳无音讯。

ADHOC 节点位置平等，没有像 AP 一样的角色能够为其他节点分配 IP 地址。因此，使用 ADHOC 网络往往需要用户将模块设置为静态 IP 地址。

为了使用的方便性，wifimodII 模块在 2.4.2 版本后提供了实验功能，使 ADHOC 组网方式下可以提供 DHCP 动态 IP 地址分配功能，细节如下：

模块需要同时实施 **使用 ADHOC 网络** 以及 **DHCP 获取 IP** 两个设置。

ADHOC 网络由第一个上电的模块创建，当创建网络的节点发现同网络中只有自己后，其会使用缺省静态设置的 IP 地址作为自身 IP，并且引导一个 DHCP 服务器，为后来的模块或者接入相同 ADHOC 网络的笔记本、手机等节点分配 IP。

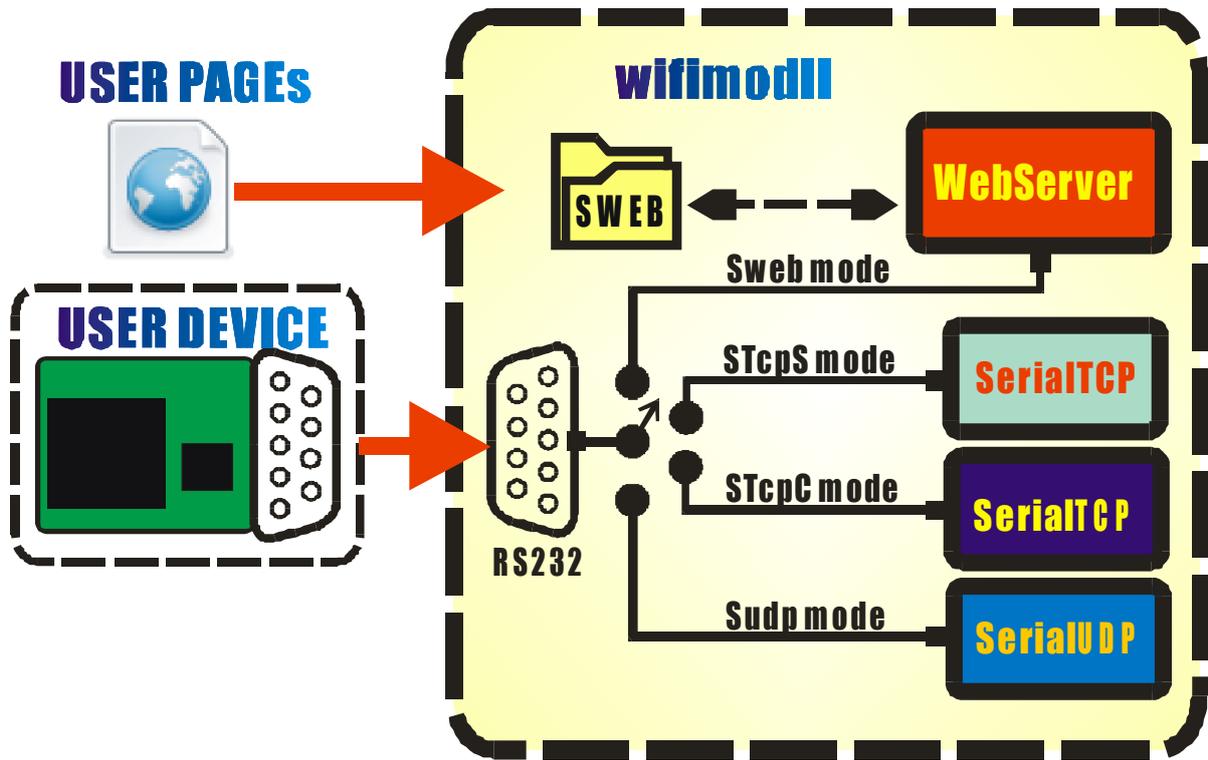
这种方式适用于无 AP，多节点动态组网的环境，例如，使用手机去控制配置了 wifimodII 模块的玩具，模块上电后，手机能够直接扫描到模块，接入后便可以得到 IP 分配。这种方式无需手机使用者设置 IP 地址，保证良好的使用体验。

但是同时也要注意该特征的局限性：

- 一、 创建网络的节点需要始终上电，后上电的节点无法察觉创始节点的退出，也无法推选出合适的后继 DHCP 服务器，维持之后的服务；
- 二、 目前模块内置 DHCP 服务器仅仅支持分配 16 个 IP 地址，无法为超过 16 个节点的 ADHOC 网络提供服务；

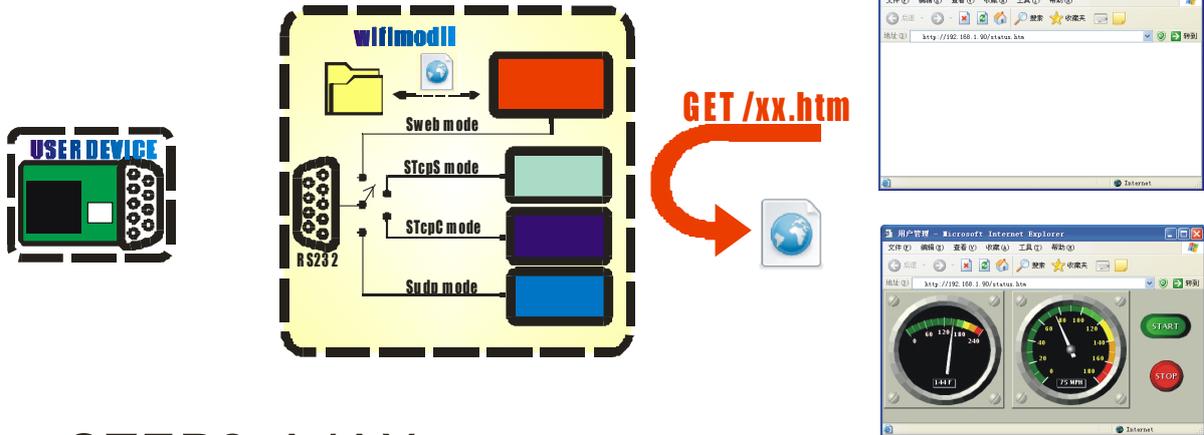


# SWEB APPLICATION DEMO

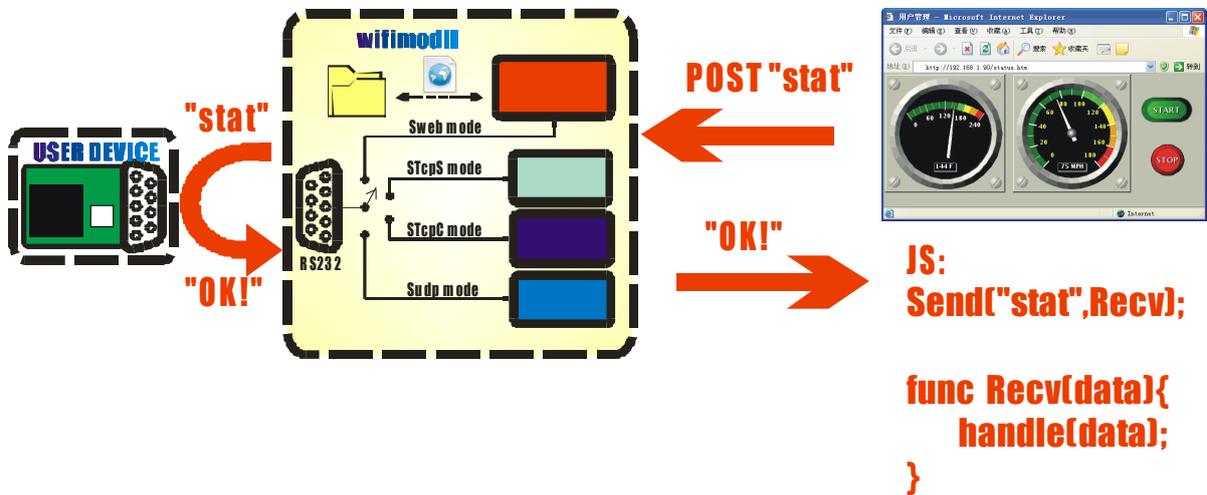


工作方式见下图:

# STEP1: Get Page



# STEP2: AJAX



wifimodII 在“/SWEB”目录下附带了 sweb 功能的演示页面，同时提供了 VB 编写的串口模拟程序，用于仿真一个用户串口设备，用户可以查看代码，了解 sweb 协议的简单和易于上手。



## 7.2 Sweb 模式核心知识点

- ◆ sweb 模式是网页服务器独占串口的工作模式;
- ◆ 浏览器中通过 “/sweb/” 路径可以访问到用户文件。(2.1.19 版本后)
- ◆ 不管是否 sweb 模式, 用户都可以访问 “/sweb” 目录下的文件;
- ◆ sweb 模式下, 浏览器对设备 IP 的访问会重定向到/sweb/index.htm 文件。
- ◆ sweb 模式下, 浏览器通过"/sweb.cgi"访问串口。例如, 访问"http://192.168.1.254/sweb.cgi", 页面会返回串口缓存数据;
  - ◆ 支持 3 种参数: “file” 请求文件; “wait” 设置串口等候时间; “clean” 指示清空串口接收缓存; (后详)
  - ◆ GET 缺省用于索取文件数据, 若不设置 wait 参数, 将使用参数配置中缺省值 HoldTime 替代; POST 多用于发送数据, 不设置 wait 参数表示不读取串口, 若设置 wait=0 表示不作等待, 发送完毕后直接读取缓存;

clean 参数用于清空串口接收缓存, 设置 clean 参数, 可以消除之前杂乱数据的影响, 适用于 “命令、回答” 工作方式;

确切过程为: 清缓存; 发送提示; 等待接收; 读缓存; 返回页面;

wait 参数用于指定等待串口的时间, 单位是 ms。

## 7.3 Sweb 模式的页面访问

wifimodII 提供存储空间可用来存放用户页面文件；除了从本地读取数据之外，wifimodII 也支持从串口获取文件。

### 访问主页：

wifimodII 的 sweb 模式下，对缺省主页（http://IP）的访问将重定向到用户空间“/sweb/index.htm”上。用户可以修改该页面，实现自定义的用户界面。

若非 sweb 模式，缺省主页是设备本身页面，等同于“http://192.168.1.254/index.htm”

### 访问用户空间文件：

用户空间（/sweb 目录）的文件可以通过 类如“http://192.168.1.254/sweb/abc.xyz”的方式来访问。

### 通过串口请求页面：

客户在浏览器中向串口请求文件，链接形式如：

```
http://192.168.1.254/sweb.cgi?file=usr.htm&clean=1&wait=2000
```

“file=usr.htm”指示请求的文件名为“usr.htm”（可以包含路径）。

“clean=1”指示在串口获取反馈之前，先清空接收缓存；

“wait=2000”参数指示在输出提示信息之后，串口空闲等待最长时间是 2000ms；

在浏览器地址栏中输入上面的地址，wifimodII 会清空串口接收缓存，并从串口输出“swebget usr.htm\r\n”字符串（“\r\n”为换行回车符），在此之后，串口接收到的数据将被返回给浏览器，直到 2000ms 内接收不到数据为止；

#### 注意浏览器的缓存机制：

浏览器默认缓存已经访问过的页面，这种特征可以大大减少对相同文件的请求数量；尤其是 sweb 模式下通过串口请求用户文件时，利用缓存机制，可以减少冗余数据量，加快浏览器的反应速度。

wifimodII 能够配合浏览器实现支持该特征，在 sweb 模式下，浏览器从串口请求相同文件都会被通告使用缓存页面，以减少对串口的占用；因为这个原因，在调试 sweb 模式时，浏览器刷新用户页面时串口可能不会出现请求信息，如果文件变化，需要重新读取数据，可以使用 Ctrl+F5（IE）或 Shift+F5（FireFox）来实现强制页面刷新。这个细节在做基于串口的页面开发时需要注意；在正常使用时，由于用户文件一般不多变，不会因为缓存机制出现问题。

sweb.cgi 请求一个带后缀名的文件，访问其他格式的文件，wifimodII 发还数据时，浏览器端将根据不同的后缀做出不同的处理，例如 htm、jpg 等页面元素将直接显示，若 rar、hex 之类的后缀则会提示下载。借助这个过程，可以实现将串口发回的数据从浏览器下载直接保存到本地文件。

## 7.4 向串口上传文件

sweb 模式下可以通过表单向串口上传文件，表单是以 POST 方式提交。

如下表单代码：

```
<form action="sweb.cgi" enctype="multipart/form-data" method="post">
  选择文件上传
  <input type="file" name="files">
  <input type="submit" value="Send">
</form>
```

在页面中将显示为：



选择文件上传  浏览... Send

点击 Send，便可以将文件数据发送到串口。

需要注意：

form 表单中，action 目标必须设置为 “sweb.cgi”；

form 表单中，“enctype="multipart/form-data" method="post"” 两个属性不能改动；

form 表单中，仅允许一个 type="file"的文件控件和 submit 控件，不能包含其他 input 控件；

## 7.5 如何与串口交互数据

sweb 模式下与串口交互数据，主要是以 js 脚本编程方式，使用 AJAX+POST 方法，在浏览器后台控制数据的发送和接收，并在前台显示。这种方式交互数据，不会受到浏览器缓存机制影响。

### 快速进入：

- ◆ AJAX 方式交换数据不会刷新整个页面，js 脚本可以完全控制数据交换流程，处理后局部的改变页面内容；
- ◆ AJAX 请求必须由浏览器端发起，类似于一种“命令、回答”的交互方式；
- ◆ 浏览器使用 POST 方式向 `sweb.cgi` 发起请求来实施 AJAX 过程，`wifimodII` 会对该请求作出反应，一方面处理请求数据（转发到串口），另一方面根据参数设置来返回数据；
- ◆ AJAX 方式传输数据，默认的编码方式是 `utf-8`，中文都会以 `utf-8` 编码形式送到串口，若串口终端不支持 `utf-8` 解码显示，则会出现乱码；若终端能接收另存为文件，在“windows 记事本”下打开，便可以正常显示（“windows 记事本”支持 `utf-8` 解码）；
- ◆ 同理，在终端上输入的中文字符如果不是以 `utf-8` 的编码送出，浏览器接收到数据，也会出现乱码。如果终端支持发送文件，那么可以尝试发送一个 `utf-8` 编码的文件，这样浏览器端可以解析出中文。
- ◆ UTF-8 中 ASCII 字符依然为单字节且编码不变，但中文字符使用 3 字节表示；

为了方便用户使用，`wifimodII` 出厂时已经对 `ajax` 过程作了包装，以 `sweb.js` 文件存放在 `/sweb` 目录下。

其中已经定义的函数如下：

```
function sweb_send_utf8(dat,cb_ok,cb_err)
```

`dat`: 发送字符串；

`cb_ok`: 发送成功后回调函数；

`cb_err`: 发送失败回调函数；

该函数将请求 `wifimodII` 仅仅发送数据到串口，不作停留直接返回。注意，该函数以 UTF8 编码发送数据，ASCII 表中可打印字符以单字节编码发出到串口，无法发送纯二进制数据流。

```
function sweb_send_read_utf8(dat,wait,clear,cb_ok,cb_err)
```

`dat`: 发送字符串；

`wait`: 等待时间参数，单位 ms

`clear`: 向串口发送数据之前，是否清空接收缓存；

`cb_ok`: 发送成功后回调函数；

`cb_err`: 发送失败回调函数；

该函数将请求 `wifimodII` 发送数据到串口，等候指定时间，返回串口接收数据。

例如：

```
//发送请求字符串“This is request!”
sweb_send_read_utf8(“This is request!”,500,0,cb_ok,cb_err);

function cb_ok(data)
{
    alert(dat);    //提示串口接收到的数据;
}
function cb_err()
{
    alert (“---Connection Break! --- ”);    //若出错，提示连接断开
}
```

```
function sweb_send_read_utf8(dat,wait,clean,cb_ok,cb_err)
```

dat: 发送字符串;

wait: 等待时间参数, 单位 ms

clean: 向串口发送数据之前, 是否清空接收缓存;

cb\_ok: 发送成功后回调函数;

cb\_err: 发送失败回调函数;

该函数将请求 wifimodII 发送数据到串口, 等候指定时间, 返回串口接收数据。

例如:

```
//发送请求字符串“This is request!”, 清理之前的缓存, 等候 100ms
sweb_send_read_json(“This is request!”,100,1,cb_ok,cb_err);
//串口返回 json 格式数据字符串, 例如 “{“val”:100}”
function cb_ok(ret)
{
    alert(“ret value:” + ret.val);
    //ret 为解析好的对象, 可以直接通过 “ret.val” 访问 val 成员。
}
function cb_err()
{
    alert (“---Connection Break! --- ”);    //若出错，提示连接断开
}
```

## 7.6 如何与串口交互二进制数据

由于 AJAX 过程对用户输入的字符采用 utf8 方式进行编码，所以，我们没有办法随心所欲的将我们经常使用的二进制数据发送到模块的串口，为了解决这个问题，可以使用 wifimodII 内置的 URLCode 编解码器来完成这个工作。

URLCode 编码方式很简单，其类似于一种字符替代机制。" "（空格）替换为"+", 英文字母、数字、"-",".","\_","~"不变，之外其他所有字符，都使用"%XY"方式直接设置其 16 进制编码值即可。

例如，“hello 中国”编码后为“hello+%e4%b8%ad+%e5%9b%bd”。

wifimodII 模块支持 URL 编解码，利用这种功能，可以在浏览器和串口间传递任意二进制数据。

不采用 URLCode 方式的数据交互过程如下：

浏览器发送栏	utf8 编码	➔	wifimodII 网络接收	串口输出
abc 中文	61 62 63 (abc)	➔	61 62 63	61 62 63
	E4 B8 AD (中)		E4 B8 AD	E4 B8 AD
	E6 96 87 (文)		E6 96 87	E6 96 87

采用 URLCode 方式的数据交互过程如下：

浏览器发送栏	utf8 编码	➔	wifimodII 网络接收	串口输出
“%E4%B8%AD %E6%96%87”	“%E4%B8%AD %E6%96%87”	➔	“%E4%B8%AD %E6%96%87”	E4 B8 AD E6 96 87
浏览器接收栏	utf8 解码	⬅	wifimodII 编码	串口输入
“%00%01%02 %03%04%FF”	“%00%01%02 %03%04%FF”	⬅	“%00%01%02 %03%04%FF”	00 01 02 03 04 FF

设置 req=20，类如：“sweb.cgi?req=20”，便可以激活 wifimodII 内置的 URLCode 编解码器。

模块在网络侧接收到的数据都会被作为 URLCode 编码数据进行解码，之后发送到串口；

而从串口接收到的数据会经过 URLCode 编码再发送到浏览器。

由于 URLCode 编码后的字符全部是 ASCII 与 UTF8 中交集字符部分，因此我们便可以在浏览器侧发送任何二进制数据到模块的串口，而串口的任何数据也都可以被转发到浏览器。

### 应当考虑的情况：

- ◆ 为了实现二进制数据任意转发，URLCode 编码把原先一字节的数据变为三个字节，加大了传输负担。
- ◆ 用户需要保证发送给模块的 URLCode 字符串的规范性，否则，模块可能会输出异常的数据；
- ◆ 用户在网页侧需要对接收到类似于“%01%FF”的字符串进行解析，才能得到二进制数据。（可以采用 swab.js 中提供的转换函数）

Swab.js API:

```
function swab_send_urlencode (dat,cb_ok,cb_err)
```

dat: 发送字符串;

cb\_ok: 发送成功后回调函数;

cb\_err: 发送失败回调函数;

该函数将提示 wifimodII 接收 URLCode 数据, 并将数据解码转发到串口, 不作停留直接返回。用户可以根据需要发送纯二进制数据流到串口。

例如:

```
//发送请求 URLCode 编码的二进制串"%00%FFabcd", 清理之前的缓存, 等候 100ms
sweb_send_read_urlencode ("%00%FFabcd", cb_ok,cb_err);
//串口将输出 0x00 0xFF 0x61 0x62 0x63 0x64

function cb_ok(ret)
{
    alert("send ok!");
}
function cb_err()
{
    alert ("---Connection Break! --- ");    //若出错, 提示连接断开
}
```

```
function sweb_send_read_urlencode (dat,wait,clean,cb_ok,cb_err)
```

dat: 待发送 URLCode 字符串;

wait: 等待时间参数, 单位 ms

clean: 向串口发送数据之前, 是否清空接收缓存;

cb\_ok: 发送成功后回调函数;

cb\_err: 发送失败回调函数;

该函数将请求 wifimodII 解析 URLCode 数据, 并转发到串口, 等候指定时间, 模块将编码串口返回数据, 转发给浏览器。

例如:

```
//发送请求字符串"%00%FFabcd", 清理之前的缓存, 等候 100ms
sweb_send_read_urlencode ("%00%FFabcd",100,1,cb_ok,cb_err);
//串口将输出 0x00 0x FF 0x61 0x62 0x63 0x64
//串口若在 100ms 内输入 0x00 0x01 0x02 0x03

function cb_ok(ret)
{
    alert("ret value: " + ret);
    //将输出 "ret value: %00%01%02%03"
}
function cb_err()
{
    alert ("---Connection Break! --- ");    //若出错, 提示连接断开
}
```

实际应用:

发送静态二进制数据 **0x11 0x22 …… 0x99 0x00:**

```
function SendBinVal ()
{
    swweb_send_urlencode(“%11%22%33%44%55%66%77%88%99%00”);
}
```

二进制串表示为 url 串的最简单办法就是每两个 Hex 字符前加上一个百分号。也就是说发送 “%”

发送动态调整的二进制数据:

```
function SendBinVal (id, val)
{
    var seg = new Array(3);
    seg[0] = 170;    //0xAA
    seg[1] = id;    //不能超过 255
    seg[2] = val;   //不能超过 255
    swweb_send_urlencode(array2url(seg));
}
```

上面的函数有 id, val 两个参数, 构造一个 3 单元的数组, **注意每个单元赋值不能超过 255**。单元赋值后, 通过 **array2url()** 函数 (该函数为 swweb 内置 API), 将数组转换为 url 字符串。然后通过 **swweb\_send\_urlencode()** 函数发送出去。

接收二进制数据:

注意 urlencode 标准中, 可显示的 ASCII 字符是不被编码的, 因此,

模块串口接收到: **0x12 0x34 0x56 0x78 0x9A 0xBC 0xDE 0xF0**

转发到网页中的会是: **%124Vx%9a%bc%de%f0**

**0x34: ‘4’**

**0x56: ‘V’**

**0x78: ‘x’**

这个问题可以使用 swweb.js 中的 **url2array()** 函数来解决, 这个函数将 url 字符串作为输入, 输出解析好的数组 (单元为字节数据), 例如:

```
url2array(“%124Vx%9a%bc%de%f0”);
```

返回数组, 8 个单元分别是 0x12 0x34 0x56 0x78 0x9A 0xBC 0xDE 0xF0

## 7.7 高级内容 MIME 类型

### MIME 类型指定

或许您需要指定串口返回数据时在 HTTP 回应头部指示的 MIME 类型，该要求可以通过指定 req 参数来实现，类如：“sweb.cgi?req=xx”，能够指定的类型列表如下：

req 设置值	MIME 类型
1	application/json
2	text/xml
3	application/octet-stream
4	image/gif
5	image/png
6	image/jpeg
7	image/bmp
8	image/x-icon
9	application/x-javascript
10	audio/x-pn-realaudio
11	text/css
12	application/x-shockwave-flash

### 更多内容

wifimodII 模块/sweb.htm 页面中包含了应用教学部分，用户可以按照其指导一步一步熟悉如何使用 swweb 模式。

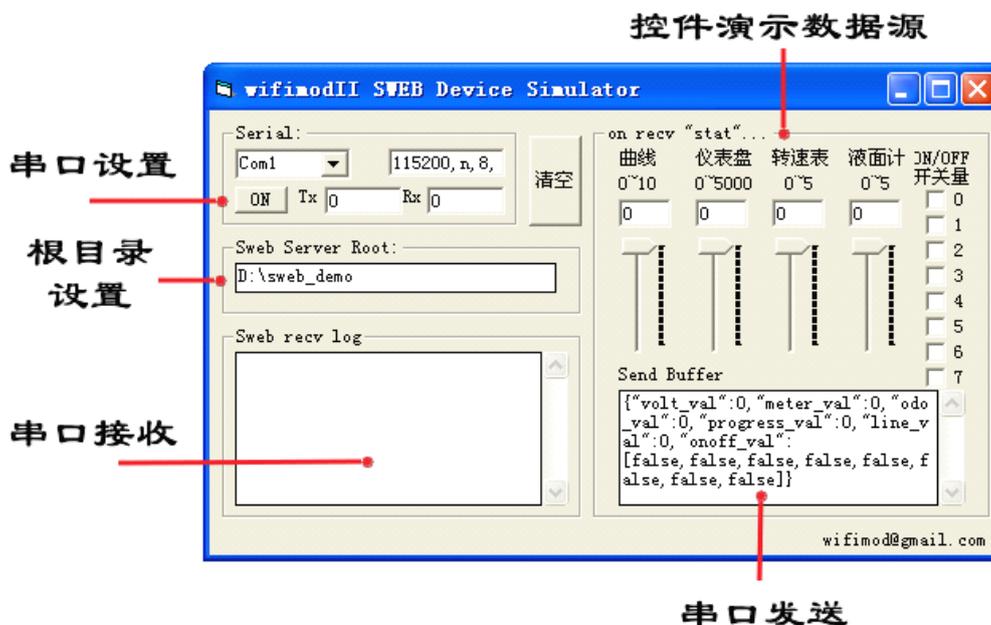
此外，在/sweb 目录下，也已经预先提供了很多 swweb 演示页面，用于展示 swweb 模式全部的功能，用户可以参考其源代码。

在“SWEB 实践”章节，用户可以在教程指导下，逐步进入网页设计流程。

有人科技可以为客户收费提供定制网页功能服务。

# 第八章、SWEB 实践

应用 WifimodII SWEB Device Simulator



该工具使用 VB 编写，用于在 PC 上模拟一个串口设备，展示 wifimodII 模块 sweb 模式的功能。代码结构简单，用户可以很快上手，不仅可以利用其学习 sweb 模式的简单协议，还可以用它来测试编写的网页，设计完毕的网页可以在测试完毕后，再上传到 wifimodII 内部的用户空间中。

## 8.1 预备工作

连接 wifimodII 与电脑的串口，将 wifimodII 上电，采用点对点或者基础模式，使 PC 机可以与 wifimodII 通讯。

注意：

观察 wlan 灯，其点亮表示 wifimodII 已经接入无线网络；

若是在点对点网络环境下，需要设置 PC 机的 IP 地址与 wifimodII 保持在同一网段；

若是在基础模式下，需要知道 wifimodII 的 IP 地址，这里假设为缺省地址 192.168.1.254

下面设置必要的配置：

如下图，配置为 sweb 模式。

工作模式设置:

工作模式:	SWEB(网页)模式
目标IP地址:	192 . 168 . 1 . 105
目标端口号:	55555 (0~65535)

提示: +

如下图, 配置 HoldTime 为 1 秒;

串口设置:

串口端口号:	55555 (1~65535)
接收空闲时间 (HoldTime):	1000 (10~60000 ms)
接收打包尺寸 (LoadMax):	1460 (16~1460 bytes)
串口波特率:	115200
传输字位长:	<input checked="" type="radio"/> 8Bits <input type="radio"/> 9Bits
停止位长:	<input type="radio"/> 0.5Bit <input checked="" type="radio"/> 1Bit <input type="radio"/> 1.5Bits <input type="radio"/> 2Bits
奇偶校验:	<input checked="" type="radio"/> 无 <input type="radio"/> 奇校验(Odd) <input type="radio"/> 偶校验(Even)
流控选项:	无流控(0)

提示: +

保存配置, 复位系统。

PC 机上运行 “sweb\_simulator.exe” 程序, 设置其根目录 (缺省为程序所在目录);

## 8.2 从最简单的网页开始

打开记事本, 建立一个文本文件, 其中输入:

```
<!DOCTYPE html PUBLIC "-//W3C/DTD XHTML 1.0 Strict/EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<HTML xmlns="http://www.w3.org/1999/xhtml">
  <HEAD>
    <META http-equiv="Content-Type" content="text/html; charset=gb2312">
    <TITLE>wifimodII Sweb Demo</TITLE>
  </HEAD>
```

```
<BODY>
    Hello, sweb! 我的第一张网页!
</BODY>
</HTML>
```

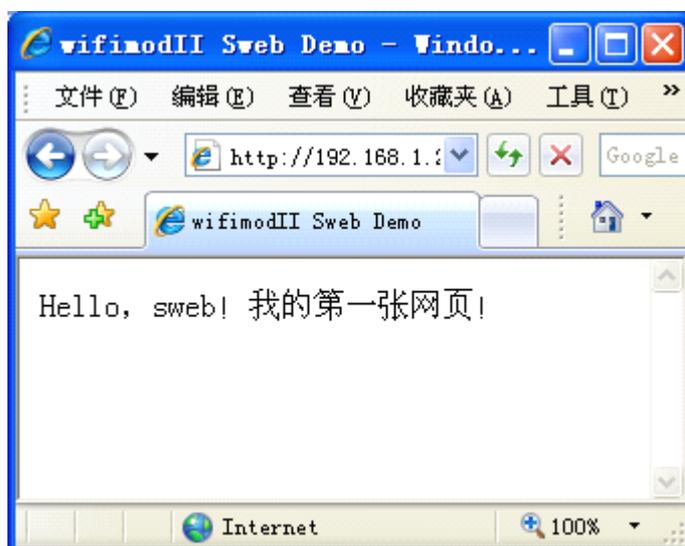
保存文件为 step1.htm，将其放置在程序设置的根目录下。

打开浏览器，地址栏输入 192.168.1.254，回车，这时显示的页面可能是设备主页面，这是浏览器缓存机制在起作用，若是 IE 或者 Chrome 浏览器可以用 CTRL+刷新按钮 刷新页面，Firefox 使用 SHIFT+刷新按钮，如果没有错误，应该可以看到页面被重定向到 “/sweb/index.htm”，该文件为出厂配置。

如果要看到我们刚才编写的页面，可以在浏览器中输入：

http://192.168.1.254/sweb.cgi?file=\step1.htm&clean=1&wait=2000

不出意外，可以看到浏览器从 sweb\_simulator 程序获得网页数据，展示如下：



运行过程：

浏览器		wifimodII		sweb_simulator
http://192.168.1.254/sweb.cgi? file=\step1.htm&clean=1& wait=2000	➔	清空缓存； 发送、等待；	➔	swebget \ step1.htm \r\n
step1.htm	➔	step1.htm 数据	➔	查找 Root_path + \step1.htm， 若存在该文件，则返回其 数据；

**注意：**

wifimodII 内部的用户空间文件系统文件名为 8.3 格式，即文件名长度不超过 8 字节，后缀不超过 3 字节。若以后需要将该文件上传到 wifimodII 内部空间，需要注意这个限制，保证文件名称的规范。

因此，请将页面保存为 xxx.htm，后缀为 htm，不能保存为 html。

在页面调试阶段，由于浏览器的缓存作用，第二次输入地址，往往不会产生页面请求，这时需要采用上面提到的刷新页面的方式，如下表：

浏览器	刷新方式
IE & Chrome	CTRL + 刷新按钮
FireFox	SHIFT + 刷新按钮

## 8.3 放置图片，添加链接

同样原理，我们也可以从串口获得其他网页元素，例如，图片。将下面一句添加在 step1.htm 文件中，保存为 step2.htm，文本如下：

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<HTML xmlns="http://www.w3.org/1999/xhtml">
  <HEAD>
    <META http-equiv="Content-Type" content="text/html; charset=gb2312">
    <TITLE>wifimodII Sweb Demo</TITLE>
  </HEAD>
  <BODY>
    从串口获取的图片
    <IMG src="/sweb.cgi?file=\cat.jpg&clean=1&wait=1000" />
    <a href="/sweb.cgi?file=\step1.htm&clean=1&wait=1000">返回 step1.htm</a>
  </BODY>
</HTML>
```

浏览器中输入：

<http://192.168.1.254/sweb.cgi?file=\step2.htm&clean=1&wait=2000>

查看结果。

演示中还添加了超级链接，实践中可能会遇到浏览器缓存的功能影响，要注意使用“完全刷新”来激活浏览器发起新的请求。

### 注意：

使用 web\_simulator 工具向浏览器提供媒体元素，在速度上受限于串口的波特率，请注意该因素合理选择图片尺寸。

## 8.4 应用 AJAX，与串口动态交互

从现在开始，我们将开始涉及到网页中 Javascript 脚本，掌握网页编程技术可以使我们的 web 界面更加智能，更加动感。

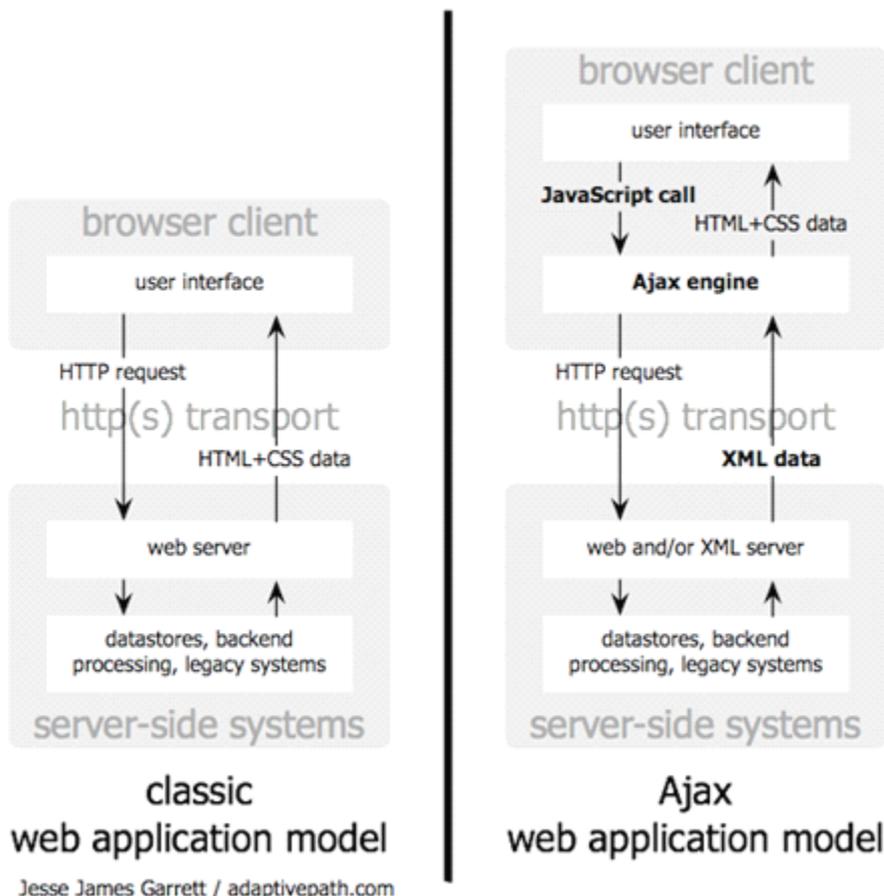
在页面中操作串口，不可避免的需要动态的与服务器（wifimodII）交互，我们将使用 AJAX 技术来完成这项工作。

Ajax 是 Asynchronous JavaScript and XML（以及 DHTML 等）的缩写。从功能上解释，这是一种在页面中使用 JS 脚本与服务器交互的技术，这种方式可以动态获取服务器端的数据，不用重新刷新整个页

面。

Ajax 基本上就是把 JavaScript 技术和 XMLHttpRequest 对象放在 Web 表单和服务端之间。JavaScript 通过 XMLHttpRequest 对象向服务器发送请求。用户屏幕上的页面也不会闪烁、消失或延迟。换句话说, JavaScript 代码在幕后发送请求, 用户甚至不知道请求的发出。更好的是, 请求是异步发送的, 就是说 JavaScript 代码(和用户)不用等待服务器的响应。因此用户可以继续输入数据、滚动屏幕和使用应用程序。然后, 服务器将数据返回 JavaScript 代码, 后者决定如何处理这些数据。

各部分的配合流程可以见下图:



关于 AJAX 技术, 可以参阅下面提供的资料, 有兴趣的用户可以深入学习:

IBM: Ajax 入门简介

<http://www.ibm.com/developerworks/cn/xml/wa-ajaxintro1.html>

这里直接给出页面代码, 用户可以将其拷贝保存为 step3.htm。

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<HTML xmlns="http://www.w3.org/1999/xhtml">
  <HEAD>
    <META http-equiv="Content-Type" content="text/html; charset=gb2312">
    <TITLE>wifimodII Sweb Demo</TITLE>
    <SCRIPT language="JavaScript" type="text/javascript">

```

```
function getXMLHttpRequest() {
  try {
    req = new XMLHttpRequest();
  } catch(err1) {
    try {
      req = new ActiveXObject("Msxml2.XMLHTTP");
    } catch (err2) {
      try {
        req = new ActiveXObject("Microsoft.XMLHTTP");
      } catch (err3) {
        req = false;
      }
    }
  }
  return req;
}
var http = getXMLHttpRequest();
function wmii_sweb_post() {
  var dest = 'sweb.cgi?wait=1000&clean=1';
  http.open("POST", dest, true);
  http.onreadystatechange = useHttpResponse;
  http.send("stat");
}
function useHttpResponse() {
  if (http.readyState == 4) {
    if(http.status == 200) {
      alert(http.responseText);
    }else{
      alert("Peer return NOT OK!");
    }
  }
}
</SCRIPT>
</HEAD>
<BODY>
  点击按钮将会向 串口发送 “stat”
  <br>
  <input type="button" value="发送 stat 命令" onClick="wmii_sweb_post()">
  <br>
  <a href="/sweb.cgi?file=\step2.htm&clean=1&wait=1000">返回 step2.htm</a>
  <br>
  <a href="/sweb.cgi?file=\step4.htm&clean=1&wait=1000">继续 step4.htm</a>
</BODY>
</HTML>
```

浏览器中输入:

`http://192.168.1.254/sweb.cgi?file=\step3.htm&clean=1&wait=2000`

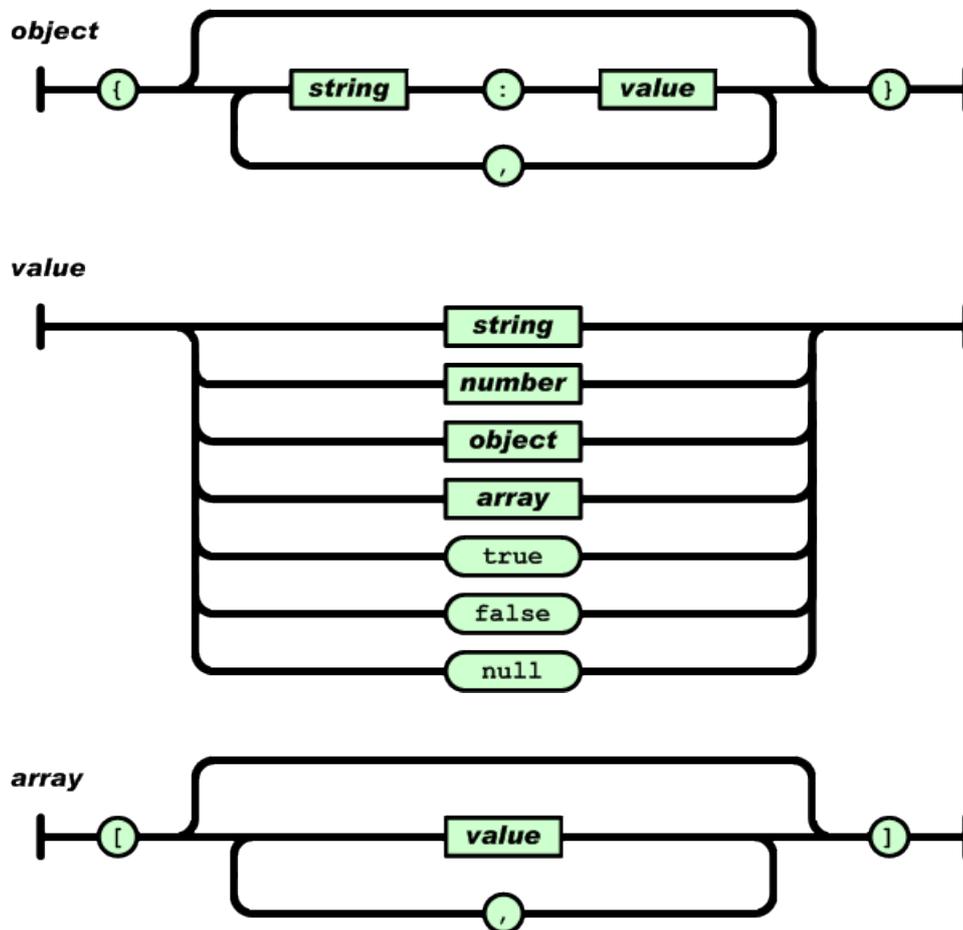
查看结果。

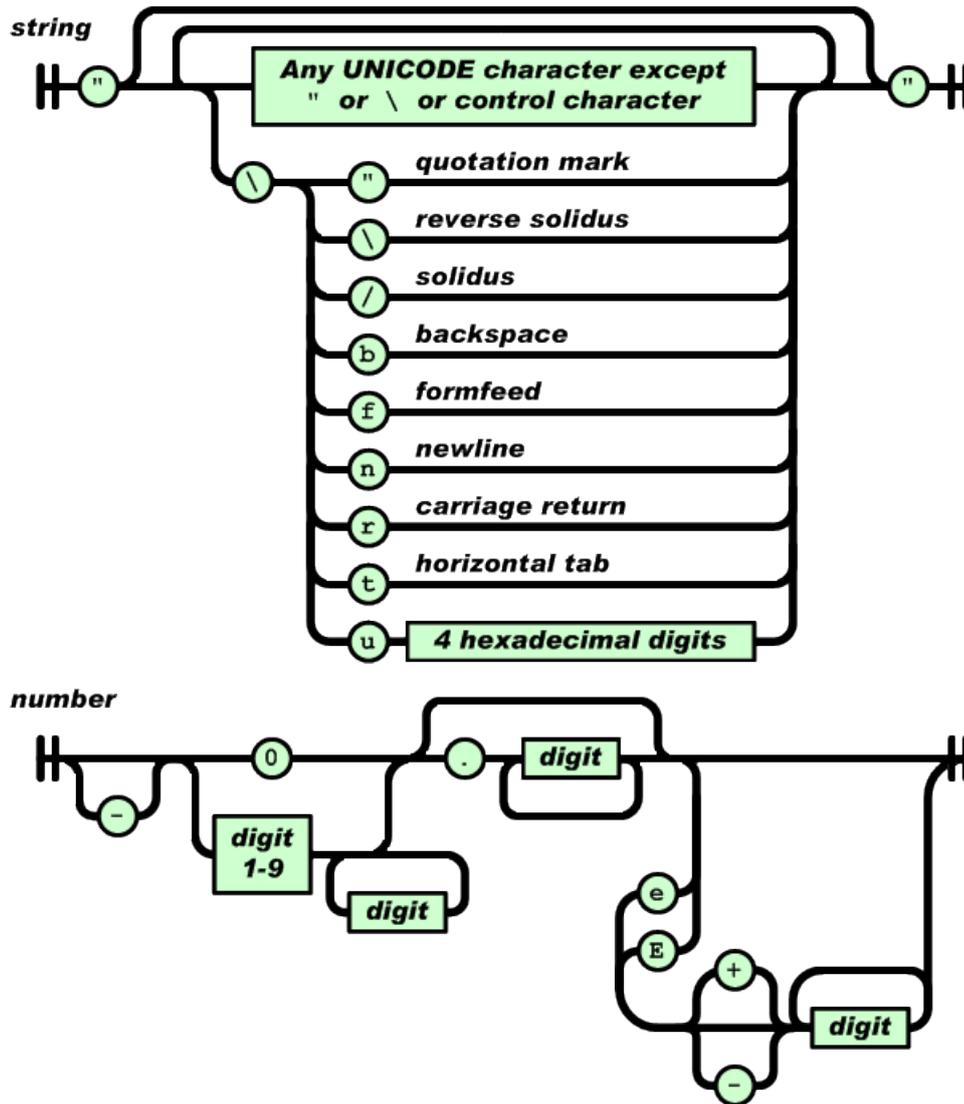
点击按钮, 可以看到 wifimodII 接收到浏览器的 POST 请求, 并将“stat”命令转发给 sweb\_simulator 程序; 相应的, sweb\_simulator 程序会发回一串字符串, 这串数据包含了一些状态信息, 如下:

```
{"volt_val":0,"meter_val":0,"odo_val":0,"progress_val":0,"line_val":0,"onoff_val":[false,false,false,false,false,false,false,false]}
```

这串信息表达了 sweb\_simulator 程序窗口右侧的控件所处状态的数值, 使用的是 JSON 格式。

JSON 格式是 web 领域用于浏览器和服务器间交互数据最常用的一种数据格式, 其定义了一个数据对象, 语法定义如下:





详见 <http://www.json.org/>

该格式定义以名值对作为主要的数据负载形式，可以包含字符串、整形数据、浮点数据以及布尔型数据。字符串形式上符合 javascript 语法，因此可以直接被 Javascript 解释为数据对象。

wifimodII 出厂 “/sweb” 目录下的演示页面演示了对JSON格式返回数据的用法，用户可以尝试访问 [http://192.168.1.254/sweb/sweb\\_B1.htm](http://192.168.1.254/sweb/sweb_B1.htm) 查看效果。

JSON 格式是互联网上最为通用的一种数据交互格式，应用这种格式，用户可以方便的放置设备的状态和数据，负载开销小，符合阅读习惯，也可以实现复杂的层次结构，在网页侧也能轻易的还原为结构化数据，因此，非常推荐用户使用这种格式。

## 8.5 使用 JS 库和样式表

到目前为止，我们已经体验到如何利用 Javascript 脚本和 wifimodII 的串口进行动态交互了。但是我们写的代码依然显得有些混杂，页面也显得比较简陋。

JS 脚本可以单独放在一个文件中，形成代码库，在 htm 文件内部可以引用该库。事实上，在互联网上有很多开放源码的优秀的 JS 库，这些库面向用户界面、AJAX 等各个方面，可以说覆盖了互联网应用的各个角落，而且经过住实际应用的考验，其性能优秀包装良好，对 web 设计者来说不啻于是一笔巨大的财富。

在这些 js 库中，首先向用户推荐的是 jquery，它的好处如下：

- ◆ 使用者众多，尤其是国内有大量的书籍和网上的材料供你参考；有大量的功能插件基于该库，各种方面的都有涉及，覆盖需求非常广泛。
- ◆ 其尺寸小巧，压缩后不到 100K；
- ◆ AJAX 以及页面常用功能封装良好，调用接口丰富到位，基本可以满足 web 设计各方面的需要；

如何使用这个 js 库，这里篇幅有限，不能囊括这部分内容了，不过建议用户可以边用边学，利用这里提供的实例，细细分析，慢慢咀嚼，加以外部材料的学习，一定可以短期内了解其特性把握其用法。

wifimodII 出厂配置中，用户空间（“/sweb”）中提供了 sweb.js 文件，其中包含了 jquery 的压缩版，用户可以通过 <http://192.168.1.254/sweb/sweb.js> 下载保存该文件，了解其内容。

若用户需要在页面中引用 sweb.js 的功能，需要在 html 文件中添加下面语句：

```
<SCRIPT language="javascript" src="/sweb/sweb.js"></SCRIPT>
```

引用该文件之后，我们便可以不用编写 AJAX 复杂操作代码，上一小节中发送“stat”命令的操作可以简单的写成下面的语句：

```
sweb_send_read_json("stat",1000,1,cb_stat_ok,cb_stat_err);
```

此外，为了页面的美观起见，使用 CSS 样式表是必不可少的。CSS 样式表可以定义页面中文字格式、版面的布局，良好的 CSS 可以预先设计整个网站的风格，划分好各个区域，使 web 开发者可以一次性的解决美工问题，把注意力集中在 web 内容上。

wifimodII 为用户提供了一套网站模板，其主要是由 step4.htm 和用户空间（“/sweb”）中的 style.css 两个文件组成，整个模板完整的提供了标题、页面链接、主内容区、侧边内容区和页脚区域，其中没有使用到任何图片，可以非常明显的减少页面请求时的负担，使加载过程更加的快捷。如下图：



step4.htm 中提供了对 sweb.js 和 style.css 的引用，也提供了使用 sweb.js 库的演示，用户可以以这个页面为基础，修改内容做出自己的页面。

至于 CSS 样式表如何使用，也请用户自行学习了解，不过只要稍微注意一下模板中的应用，就可以很快入门。

step4.htm 已经存放于 sweb\_simulator 程序所在目录，访问下面地址：

<http://192.168.1.254/sweb.cgi?file=\step4.htm&clean=1&wait=2000>

便可以看到上面图示中的页面。

#### 注意：

直接在 PC 中打开 step4.htm 将会看到没有布局的页面，是因为本地无法找到 style.css；

对 sweb.js 和 style.css 的引用是指向 wifimodII 内部/sweb 目录，因此，这两个文件不通过串口提供，浏览器可以非常快速的获得这两个文件。

## 8.6 修改页面，上传至用户空间

使用 sweb\_simulator 工具调试页面，不可避免会受到串口速率限制影响，如若引用的资源尺寸过大，可能会导致页面加载速度很慢。

因此，用户可以将比较大的资源文件预先上传到“/sweb”空间中，使用如下类似语句做出引用：

```
<IMG src="/sweb/cat.jpg" />
```

调试成功的页面若想要上传到/sweb 用户空间，需要注意：其中形同“sweb.cgi?file=\step4.htm”的串口媒介链接 需要被改正为相对地址链接，如果本 html 文件与待引用的资源在相同目录，那么可以直接引用其名称，例如：

```
<SCRIPT language="javascript" src="/sweb/sweb.js"></SCRIPT>
```

可以改进为:

```
<SCRIPT language="javascript" src="sweb.js"></SCRIPT>
```

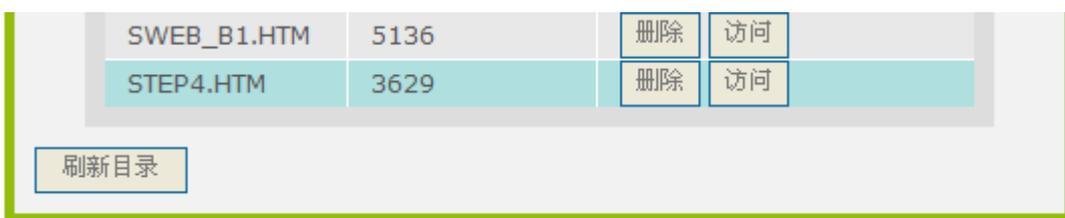
再此外，建议用户在设计自己页面的时候，不要过于追求页面的华丽和丰富，采用大量的图片或其他资源文件。不仅加载速度缓慢，一个页面中包含过多对其他资源的引用会导致浏览器发布过多 TCP 连接来请求资源，wifimodII 同时最多支持 3 个 TCP 连接请求，过多的连接会降低服务器的性能，可能会造成某些资源文件请求失败。

建议将常用的 js 脚本尽量复用，放置在一个 js 文件中；类似的，css 样式表也最好集中于一个文件；图片等资源文件尽量减少其尺寸，减少数量；

文件被修改完毕后，便可以在设备的配置页面选择上传，如下图：



上传完毕后，点击“刷新列表”，可以看到新上传的文件。



这时，便可以通过<http://192.168.1.254/sweb/step4.htm>链接，访问wifimodII本地的页面了。

## 第九章、附录

wifimodII 提供 4 个 LED 指示灯，

指示灯	说明
POWER	红色； 表示当前电源情况；
WLAN	绿色； 表示当前是否接入无线网络；闪动表示接入过程中； 点亮为已接入
NET	绿色； 表示当前是否建立 TCP 连接；闪动表示有数据流动；
SERIAL	绿色； 表示当前串口是否被打开；闪动表示有数据流动；

LED 灯闪动模式说明：

出厂模式：
NET 与 SERIAL 灯会同步以两秒为周期闪动，表示进入出厂模式。出厂配置为： 无线网络模式：AdHoc（点对点模式）； 无线网络名：wifimodII@192.168.1.254； IP 地址：192.168.1.254；
Flash 芯片故障：
NET 与 SERIAL 灯会同步以一秒为周期闪动
其他故障：
NET 与 SERIAL 灯始终灭

修改记录：

时间	事件
20100523	初稿
20100603	添加故障模式下的 LED 状态
20100612	添加完善 “无线网络模式选择和部署” 章节 以及四种模式应用场景；
20100725	添加 TCPc 模式下本地 Port 自动选择说明，以及 sweb 相关说明；
20100906	添加丰富了 sweb 相关章节内容；
20100921	添加 “工作模式与性能调整” 章节，完善了串口配置部分中 工作细节的描述； 添加 “sweb 实践章节”
20101106	添加 “工作模式详解” 部分；

	添加“应用实例之公网 (Internet) 应用”部分;
20101117	添加“stcpc 模式的 keepalive 特性和重连特征”部分;
20101121	添加“配置维护”章节
20101217	添加“MIME 类型指定”“交互二进制数据”章节 添加“STcpS 模式与富客户端 Flash 或 SilverLight 应用”
20110310	2.4.2 版本更新 添加了“网络配置”小节; 添加了“ADHOC 网络下的 DHCP 功能”章节; 添加“漫游 (Roaming) 参数”设置说明; 添加“Ping 目标 IP 地址”设置及其功能的说明; 更新“UDP 模式一对多的逻辑”; 添加“RTS 引脚缺省输出低电平”描述;
20110605	发现协议添加 SNR NF 两个信息;
20110621	2.4.3 版本 添加 sweb.cgi 访问认证 设置说明
20110623	2.4.4 版本 修改发现协议细节, 增加了工作端口信息;

## 联系方式:

公 司: 济南有人科技有限公司  
地 址: 济南市高新区奥体中路盛福花园 D5-3-302  
电 话: 0531-55507297 13864129404  
网 址: [www.usr.cn](http://www.usr.cn)  
邮 箱: 销售 [sales@usr.cn](mailto:sales@usr.cn) 技术 [tec@usr.cn](mailto:tec@usr.cn)  
QQ 联系: 835475229 286629322