USR-WIFIMOD-II user guide

CONTENTS

1. Introduce of WIFI module

1.1 Introduce

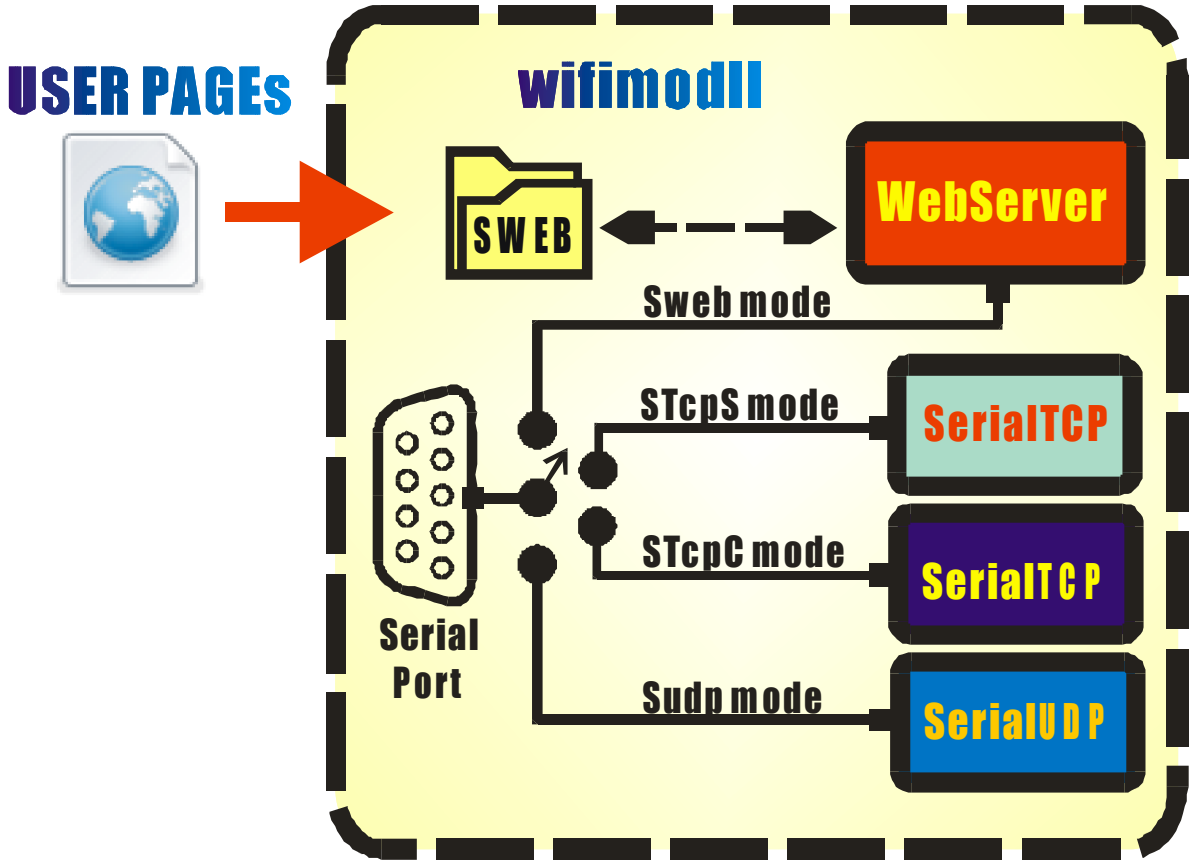
Almost all communication devices including PC are using serial transmission. In this type, devices send and receive data in the order of each byte. The serial communication is quite simple to implement but has weaknesses such as short distance and hard maintenance.

USR-WIFIMOD-II lets the serial devices connect to the WIFI mobile or Internet via wireless LAN (IEEE802.11b/g). To communicate with mobile phone or on Internet, devices should use TCP/IP protocol, so USR-WIFIMOD-II processes the converting serial data to TCP/IP.

WIFIMODII support HTTP server functions and provides web configuration interface.

Additionally, it provides 1 Mega Bytes user space, which can be used to store page files. User can easily upload custom pages to USR-WIFIMOD-II module.

In addition to traditional UDP\TCP Client \TCP Server Net to Serial usage. USR-WIFIMOD-II support SWEB (serial to web)mode, which provide a very handy way for controlling serial port via java script in web pages.



1.2 Parameter detail

1. IEEE 802.11b/g, support Ad-hoc point to point (no need to use AP).
2. Support for security standards WEP64/128, WPA/RSN(WPA2)PSK
3. Static IP setup, DHCP, Auto IP
4. Serial2Net transparent work mode: TCP Server, TCP Client , UDP , support DNS for domain target
5. Handy SWEB mode, 1Mega bytes storage, and Communication via HTTP Post
6. Support download user data and documents from serial port , support browser to up\download the serial data, support AJAX
7. 1M Bytes user flash space for user web page and data
8. Can work with wireless router(Infrastructure)
9. RS232 baud rate 1200~115200, Data bits and parity can be set
10. Setup all parameters via HTTP web page, there are password for page login for Security.
11. Support online upgrade, operated in browser.
12. Provide SWEB page design example,10+ templates.

1.3 Led status

NET&SERIAL LED blink in every 2seconds means it is work in factory default setting.

LED	Notes
POWER	Power status
WLAN	Indicate wireless connection. Blinks means trying. On means connected.
NET	Indicate TCP connection status. Blink means receiving data.
SERIAL	Indicate Serial is opened Blink means receiving data.

1.4 Power supply

Input power input is DC 5V~16V, maximum current is about 300mA, so the power adapter current should not less than 300mA.

Try to use high-quality power adapter supply to make sure the system work ok.

We supply one 110~220AC to 5V DC power adapter, positive outside and negative inside.

1.5 Packet list

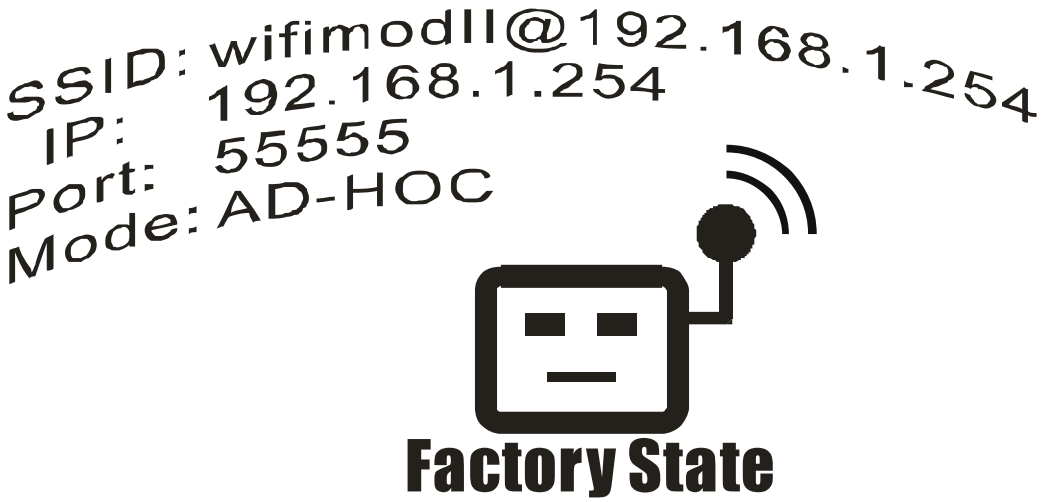
1. USR-WIFIMOD-II WIFI to RS232 convert module * 1
2. AC to DC Power adapter * 1
3. Female to Female Serial cross cable * 1
4. CD with user manual and software * 1

2. Configure and use USR-WIFIMOD-II

2.1 Connect to WIFI module

In factory state(initial configuration), USR-WIFIMOD-II behaves as below:

- It's NET&SERIAL LED Blink in every 2seconds.
- It will work in ADHOC mode,whose SSID is "wifimodII@192.168.1.254"
- It holds a static IP address "192.168.1.254"
- It works in Udp2Serial mode, and serial access port is 55555.

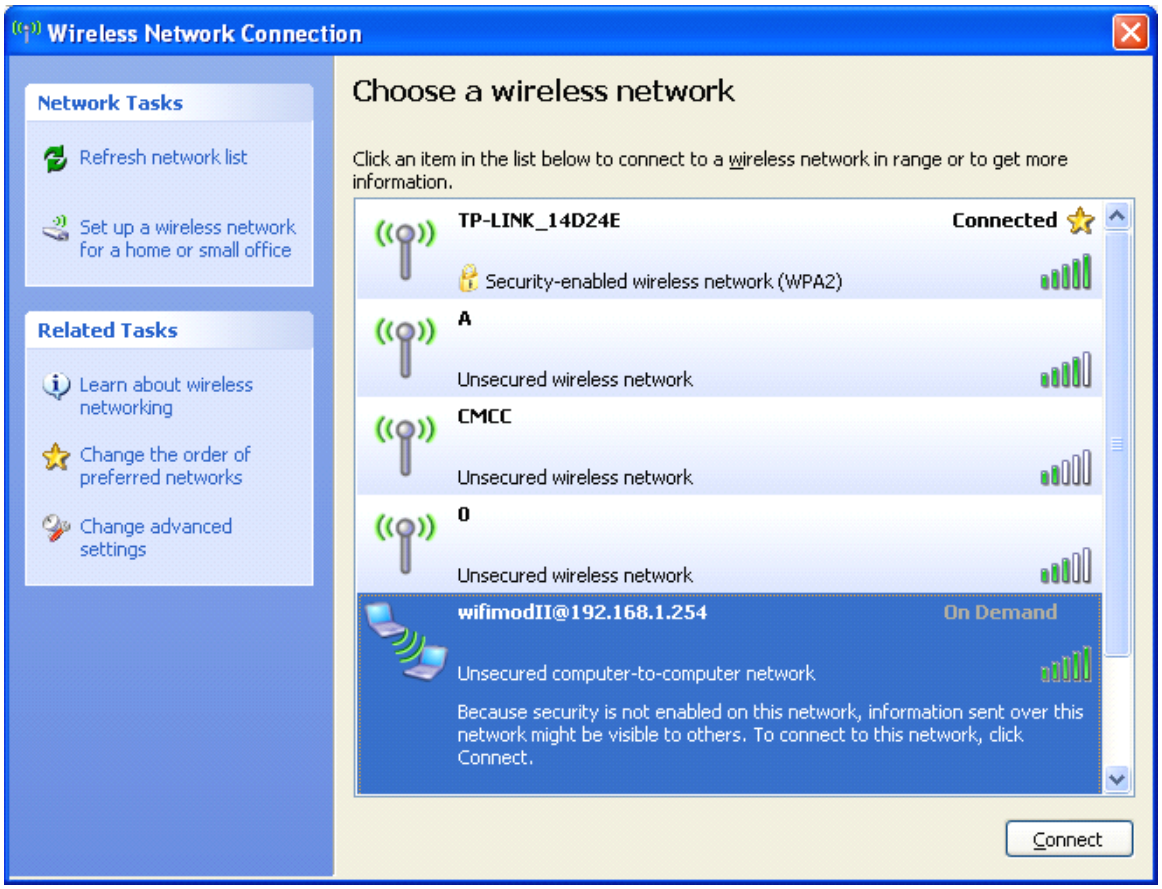


You can use laptop(with WIFI adapter) Ipad or Iphone to connect to WIFIMODII. with IP address properly set to same subnet (such as 192.168.1.105), you can use web browser to access the module.

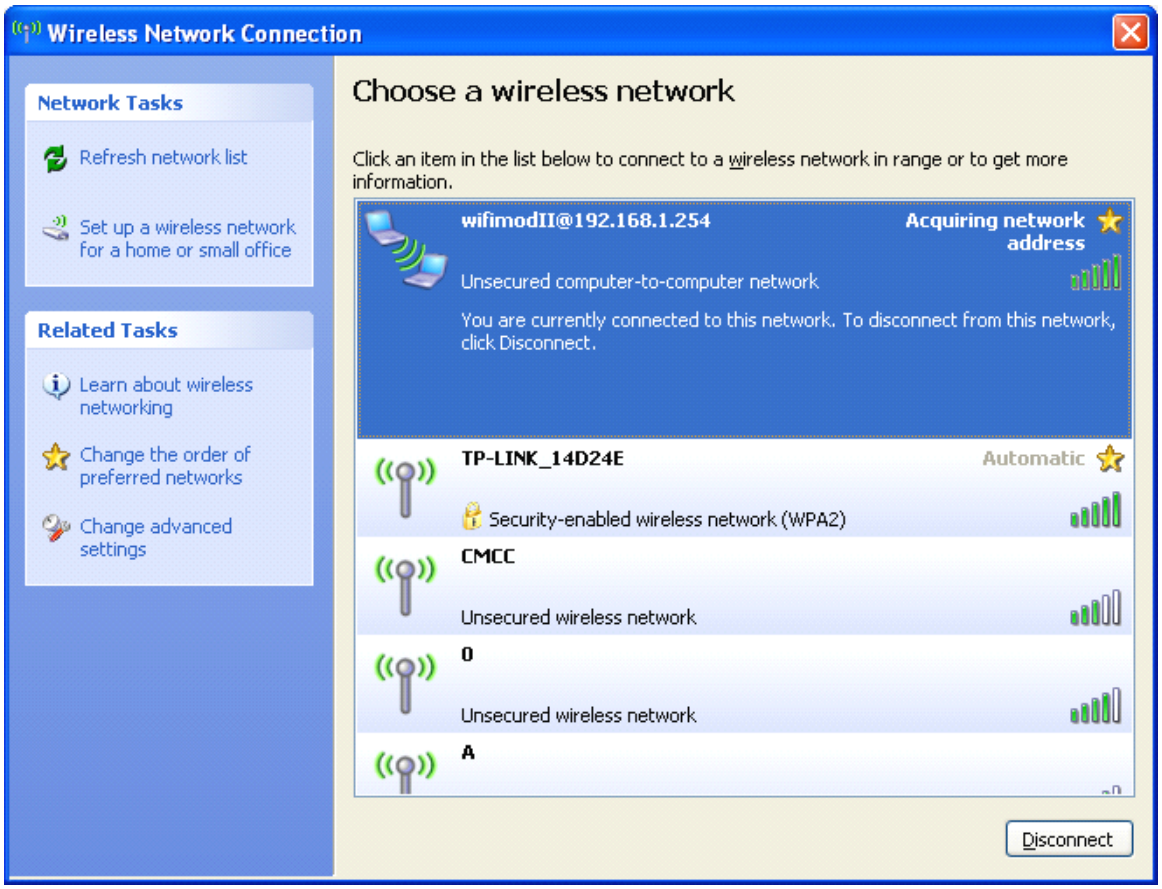
Steps:

Power on the USR-WIFI-mod-II, the two leds serial and net flash every 2 seconds, show the WIFI module work in factory setting.

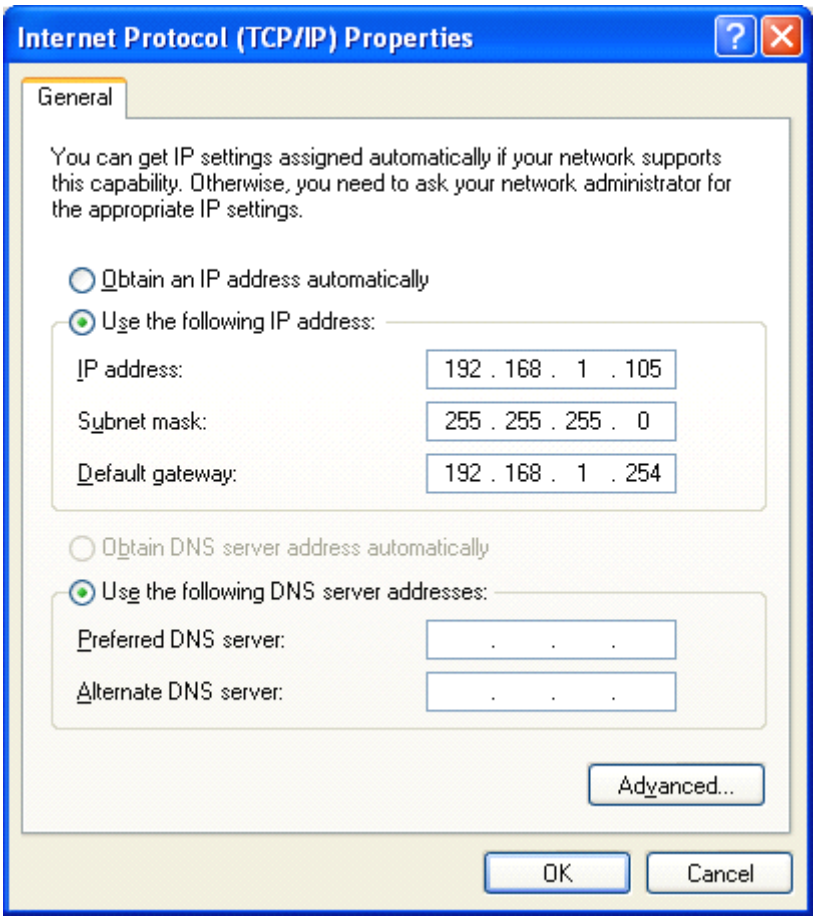
Find one PC with WIFI support, open your PC wireless network manager, Refresh network list, you can see "wifimodII@192.168.1.254" as follow picture. It is a computer-to-computer network, now connect to it.



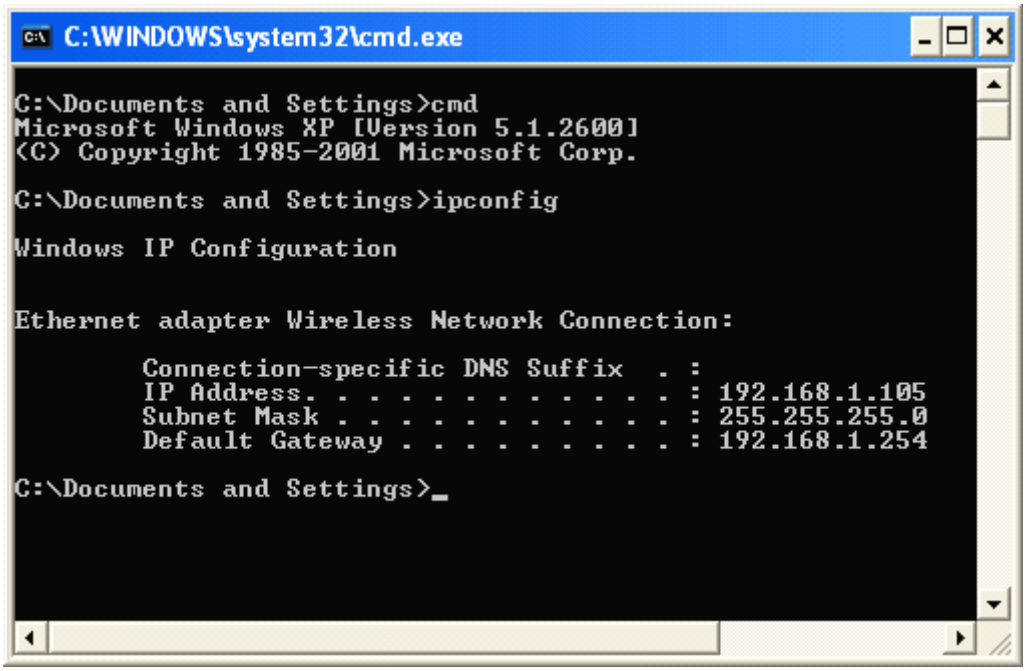
Then it shows Acquiring network address.



As you can see , this link can not auto get IP address, so set one IP by your self, Wireless Network Connection status -> Properties -> Internet properties(TCP/IP)-> fill the tables like follow picture.

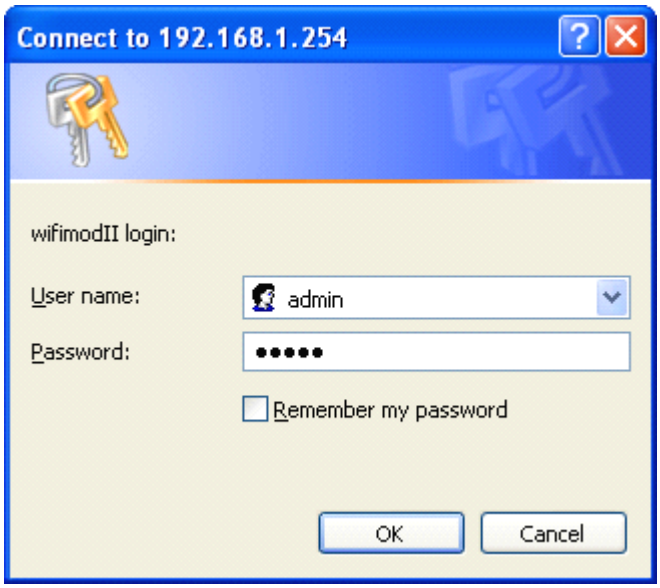


After all OK, you can confirm the new IP setting, just use a command "ipconfig" in cmd.exe



Now you can use Internet Explorer Browser to visit the web page in WIFIMOD-II, type http://192.168.1.254 .

You will be asked to login, fill User name and Password both with "admin", the default setting for WIFIMOD-II.



After login, you can change to the english version page, on the page you can visit and change and save settings.

2.2 WIFI network mode choose

As you can see i 2.1, USR-WIFIMOD-II network is work at computer-to-computer (Ad-hoc) mode at default factory setting, it can also work at Infrastructure mode with wireless router.

WIFIMODII supports IEEE802.11b/g. The IEEE802.11b and IEEE802.11g support 11Mbps and 54Mbps transmission rate respectively. There are two types of wireless LAN networks – ad-hoc and infrastructure.

Basic Requisite

SSID

It is an identifier to identify the particular wireless LAN. So the same SSID should be configured to all stations to communicate in the same wireless network. In case of infrastructure mode, user has to set his station's SSID as same as AP's. The maximum value of this parameter is 31 bytes.

Channel

IEEE802.11b/g wireless LAN stations communicate through the ISM (Industrial, Scientific, and Medical) band whose frequency is about 2.4GHz. IEEE802.11 specification divides this band into 14 channels every 5MHz. If user installs more than one wireless LAN networks in the same area, the channels should be apart more than 4 channels to avoid interferences.

WIFIMODII module can arrange channels automatically.

Authentication and Security

Authentication

A wireless LAN station should get authentication from the AP in the infrastructure mode. There are the Open system and the Shared key for the authentication methods.

WEP (Wired Equivalent Privacy)

The WEP is a secure protocol for wireless LAN. There are two kinds of WEP method – 64 bits and 128 bits key.

WPA (Wi-Fi Protected Access)

WPA is a security standard for users of device equipped with Wi-Fi wireless connection. It is an improvement on and is expected to replace the original Wi-Fi security standard, Wired Equivalent Privacy (WEP). There are two modes about the user authentication in WPA security. The one is Enterprise which has authentication server and the other one is PSK (Pre-Shared Key) which does not have any server. WIFIMODII supports WPA-PSK (TKIP).

WPA 2

To final security of Wireless LAN, IEEE 802.11i, a standard about Wireless LAN, has suggested the Counter Mode with Cipher Block Changing Message Authentication Code Protocol (CCMP) for replacing the TKIP. CCMP uses Advanced Encryption Standard (AES), it is the WPA 2 that adopts the using the method. WPA 2 has also both Enterprise and PSK mode. WIFIMODII supports WPA 2-PSK (AES).

WIFIMODII supports WEP, WPA-PSK (TKIP) and WPA2-PSK (AES)

2.2.1 Ad-hoc Mode

Wireless stations communicate each other without the AP. So user can make a system more simply. It is proper if there's no wired LAN requirement and it is a small network. Some people call it peer-to-peer mode.

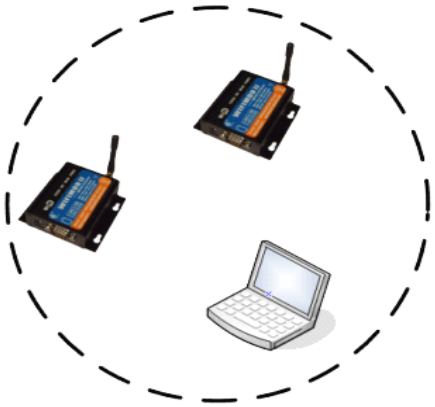
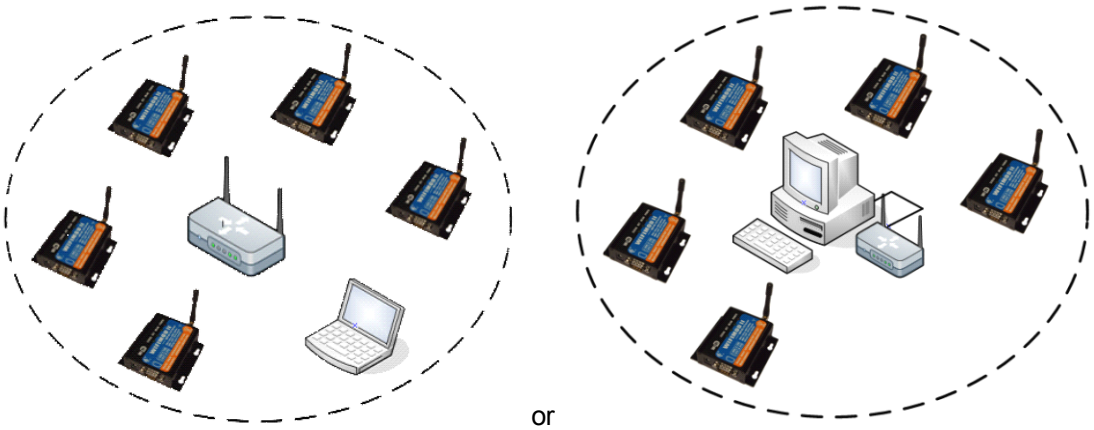


Figure 2-1 ad-hoc

Key feature:

- AP not essential, low cost.
- More than one node can join in.(but limited)
- Static IP needed (if DHCP server unavailable)
- Only WEP security supported.

2.2.2 infrastructure Mode



or

Figure 2-2 infrastructure

The wireless LAN stations communicate through an Access Point (AP). So, at least one AP is needed to make the infrastructure network. The wireless LAN station can talk to wired network hosts because AP relays between wireless LAN stations as well as between wireless LAN station and wired LAN (Ethernet) host.

Key feature:

- AP needed, more range covered & transmission stable.
- More than one node can join in.(but limited)
- Dynamic IP can be used (Most AP have DHCP server function)
- WEP\WPA\RSN security supported.

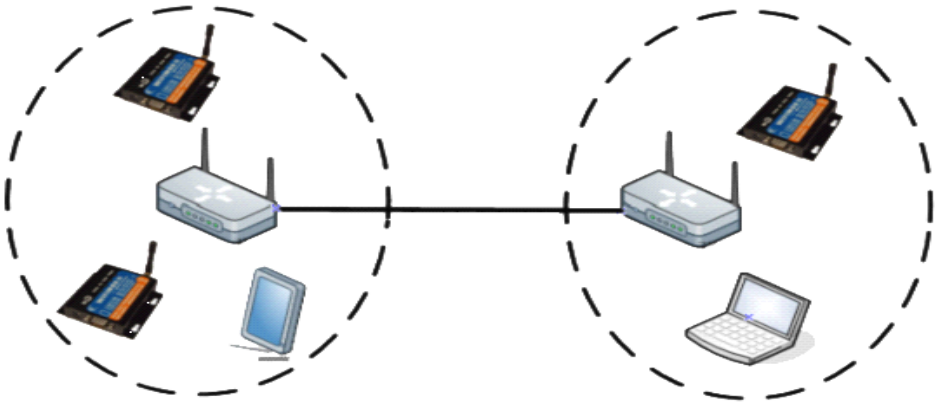


Figure 2-3 Wireless distribution system

2.3 Configurate module

Module provide web interface for configuration.On setup page, “save” button will submit user’s configuration, and save it to file system as a file.

Module will read user configurate file ONLY when it boot up. So a **reboot is necessary after click the “save” bottom on setup page**.

All the following settings can be found on the setup web page.

2.3.1 wireless setting

Network name	SSID
link mode	Ad-Hoc or Infrastructure
RSSI low limit	[1]
Encryption settings	According to destination AP security settings.

[1] In circumstances with more than one AP having same SSID, module select AP with best signal strength.When association complete,module bound to this AP(by AP mac add). If Roaming enabled,module keep checking AP signal all the time. When this value drop below user defined threshold, module start to rescan in background(lead to some performance loss),until it find a new AP(same SSID)with better signal. So, if not in a multi-AP environment,and module is not keep moving, do not enable it.

TIP: WLAN LED blinks means that it is trying to associate to wireless network.When it keeps on, means module is on-line.

2.3.2 Work mode setting:

Work mode	Module's service selection. UDP/TCPS/TCPC [1]/SWEB[2]/UDPro(test)[3]
Destination IP	[4]
Destination domain	When in tcpc mode, dest IP set to 0.0.0.0, this will indicate tcp connection destination. e.g. "www.domain.com"
Destination port	[4]
After TCP connect	[5]
Ping IP	[6]

[1] TCP Client mode: module try to connect to Destination IP after link to WIFI network, in the first 10 times, every 15 Seconds try to connect if not connected, after the 10 times, the retry Frequency change to every 60 Seconds. if connection is Disconnected, it will retry connect. Destination IP and Destination Port is the target for TCP Client to connect, if

Destination IP set to 0.0.0.0, WIFIMODII will Resolve the target domain. this is based on DNS setting.

[2] SWEB mode: see details in "SWEB" section.

[3] UDP_Pro mode: UDP_Pro is an advanced UDP mode, the data segment exchanged on serial port include 16 bytes head, having IP\Port\length info.

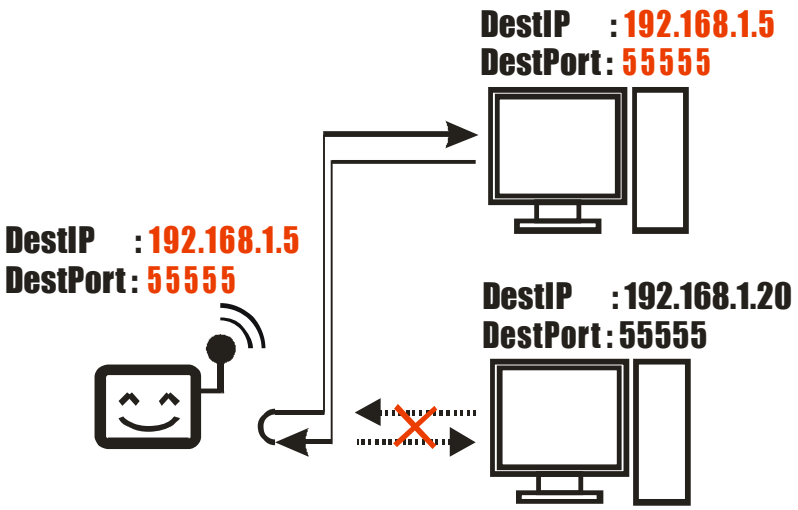
[4] Dest IP & Dest Port, play a role in specifying or filtering the target.
In UDP mode: if destination IP and Port be set,module will bounded to the target. If dest IP or dest Port be set to 0,it means any IP or any Port can sent data to module's serial port.To some extent,dest IP&Port can be used as a filter. If one of them is 0,the module will record every access to serial port, and data from serial port will be forwarded to the most recent visitor.
In TCP Server: WIFIMODII work as a TCP Server, if the Destination Port set to 0, it allow any IP connect to it, else specified allowed only.

[5] In TCP Client mode, module will send its name&group name just after connection established.This can be used by server to identify module.

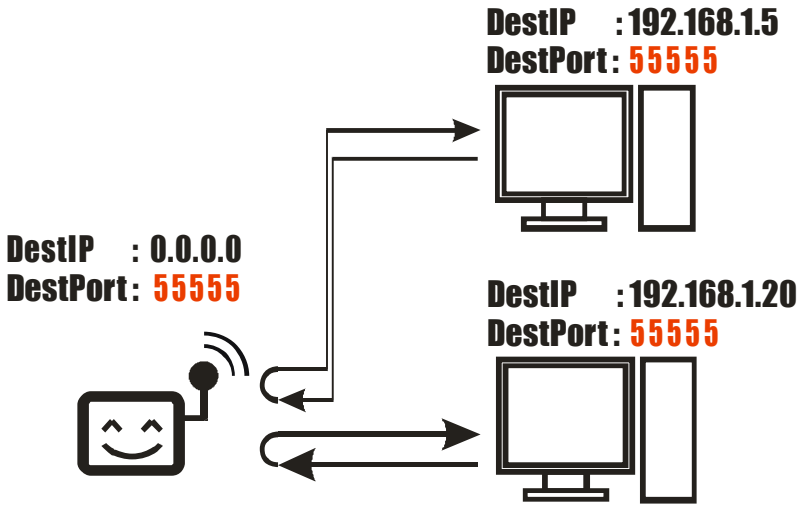
[6] When specified, "ping keep-alive" function will be enabled. Module will periodically ping the specified IP (router's IP recommended), if the other did not respond within 1 minute, the module will reset itself. This feature is used to keep wireless connection for long time work.

Details for destination setting:

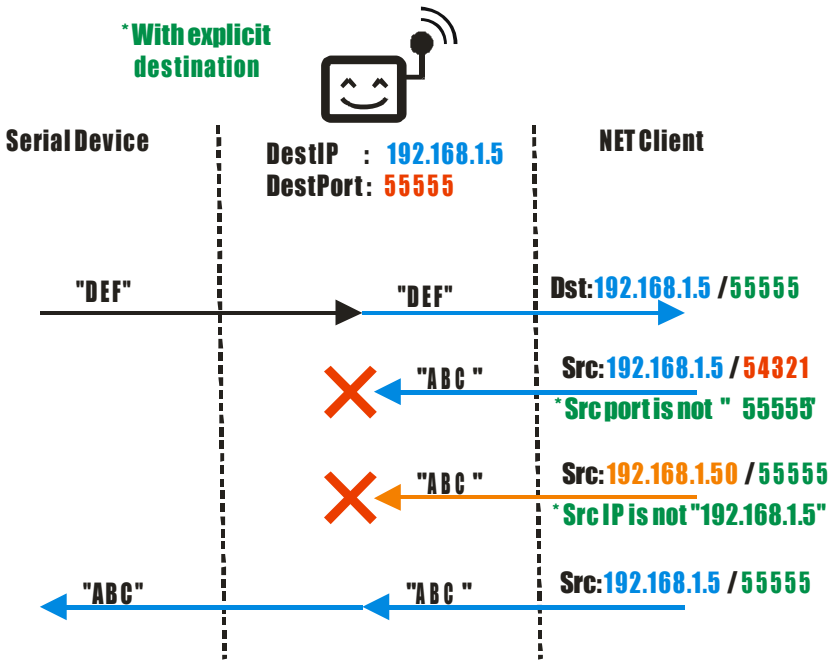
UDP mode

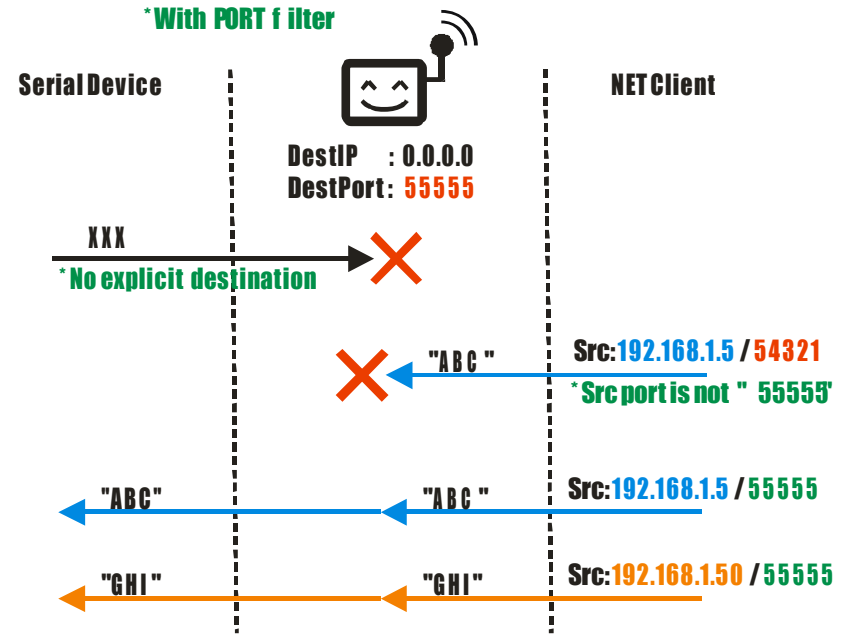


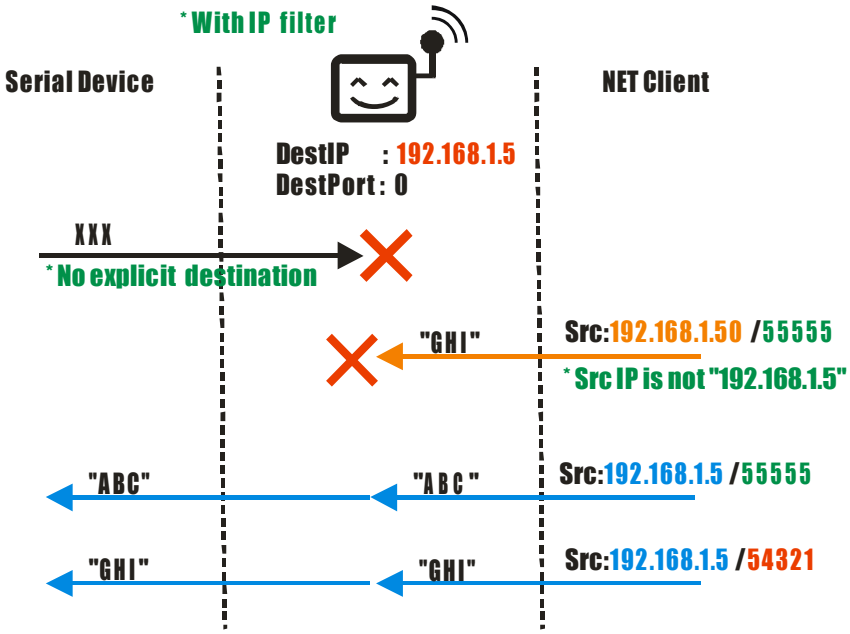
When destination been set explicitly

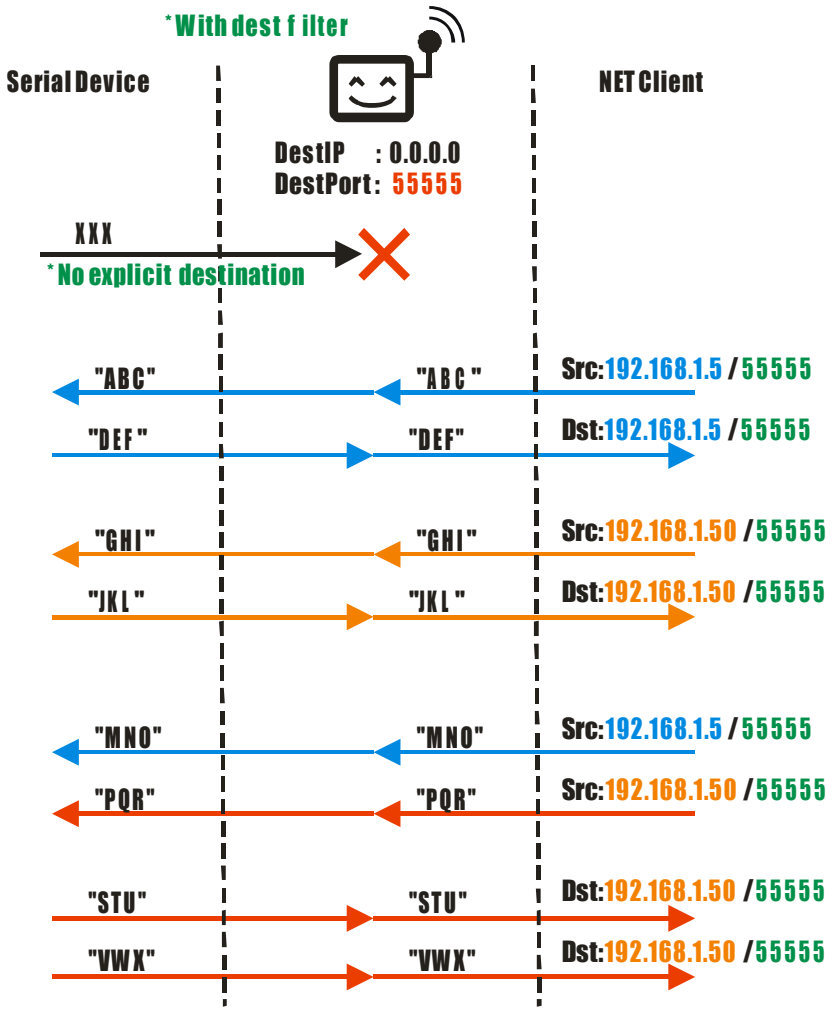


When destination IP not limited




*With PORT f ilter
Serial Device
NET Client
DestIP : 0.0.0.0
DestPort: 55555
XXX
*No explicit destination
"ABC"
Src:192.168.1.5 /54321
*Src port is not " 55555"
"ABC"
"ABC"
Src:192.168.1.5 /55555
"GHI"
"GHI"
Src:192.168.1.50 /55555


*With IP filter
Serial Device
NET Client
DestIP : 192.168.1.5
DestPort: 0
XXX
*No explicit destination
"GHI"
Src:192.168.1.50 /55555
*Src IP is not "192.168.1.5"
"ABC"
"ABC"
Src:192.168.1.5 /55555
"GHI"
"GHI"
Src:192.168.1.5 /54321



2.3.3 Network setting

IP mode	How to set module's IP address
Module IP	Used if static IP mode is selected.
Subnet mask	
Gateway IP	
DNS	DNS IP

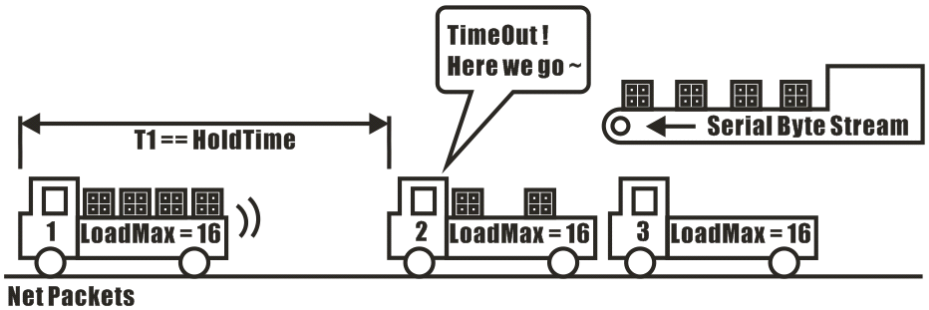
2.3.4 Serial port setting

Module Port on TCPIP	Net port used to access module's serial port.[1]
Receive free time (HoldTime)	[2]
Receive packet size(LoadMax)	[2]
Baud rate	
data bits	**If Parity set to EVEN or ODD,this should be set to 9bits**
stop bits	
Data Parity	
Data flow control	If use RS485, here must choose RS485 control.

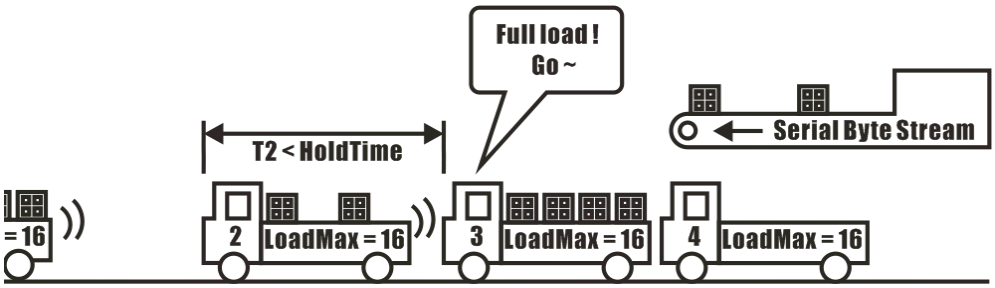
[1] In most cases,there is no need to change this parameter.If you need to change, be sure **NOT set to 12345**, this is used for node discovery protocol. **Also avoid 4530 or 4540**, if you do not use silver light or Flash socket technology. In TCP server mode, set this to 4530, will enable Silver Light Socket authentication mechanism. This is used to support user's silver light program using TCP socket to connect module.Port 4540 is the same,but for Flash XML Socket authentication.

[2] Module keep caching Serial input data until 2 conditions occur. First,incoming data interval longer than Hold Time; Sencond, cached data longer than Load Max. These two conditions will lead cached data packing and forwarding.

Load Max & Hold Time details:



If serial RX cache hold data exceed Hold Time…

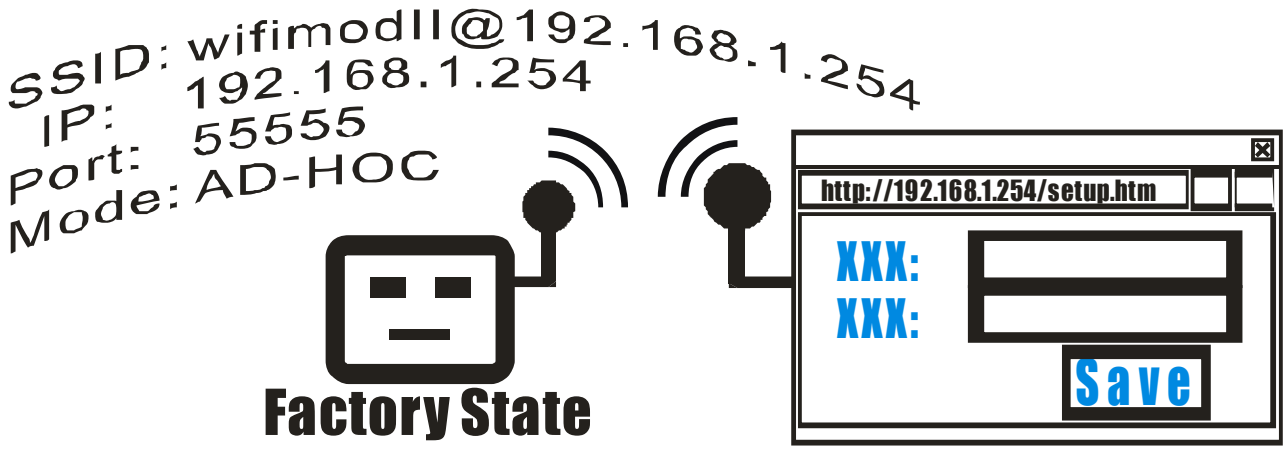


If serial RX cache get enough data…

2.4 Recover to factory settings

Factory default mode:

WIFIMODII initial wireless mode is **AD-HOC,** user can connected to module and configurate it in web browser.



When in default mode, module open 55555 port for UDP to Serial service

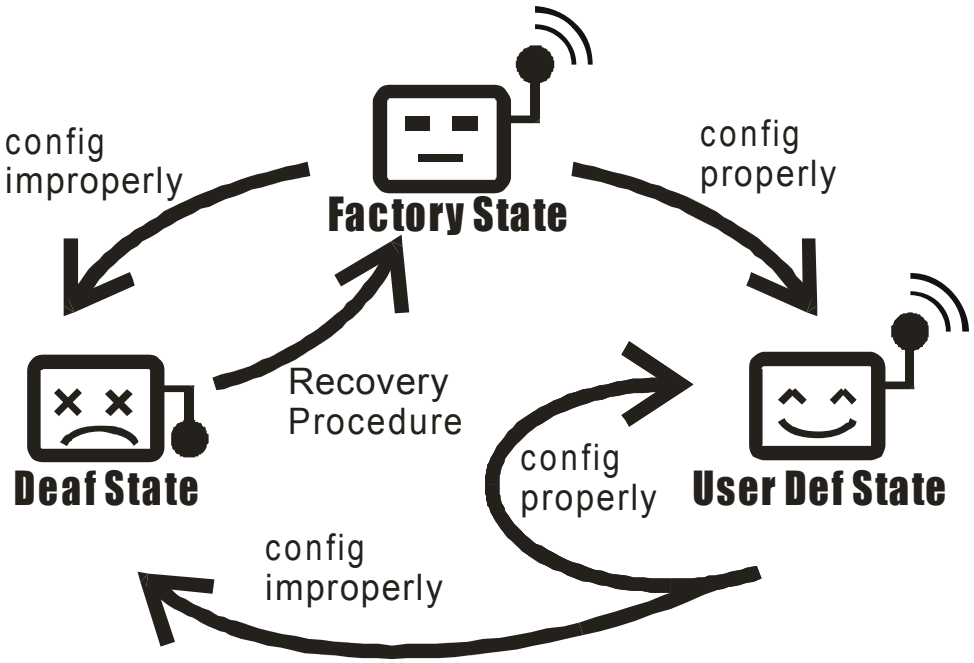
Deaf state:

Module may enter into DEAF state if configurate improperly, in this state, user can not connect to module wireless,and can not access module's setup page.

By sending specific string to module's serial port,user can restore module to the factory default state. This will be described in detail.

User defined state:

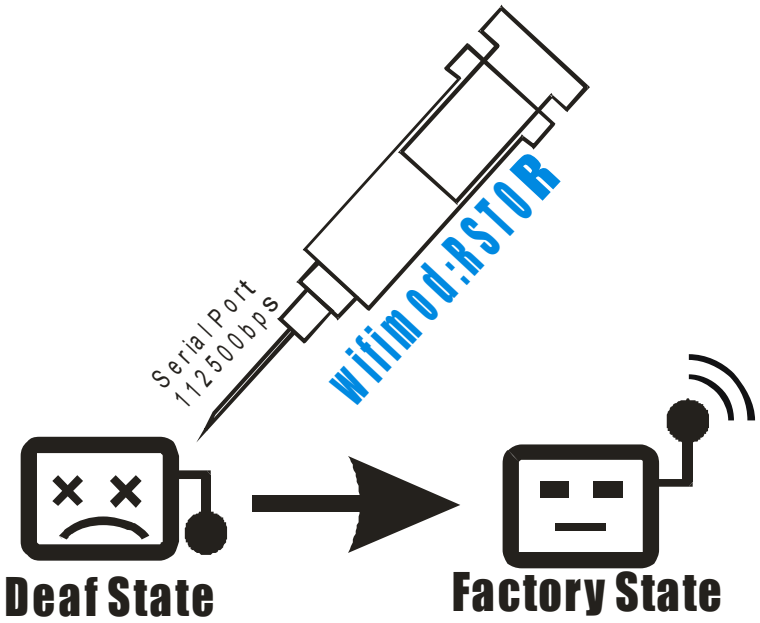
If properly configurate, module will work in required mode, and user could access module wireless.



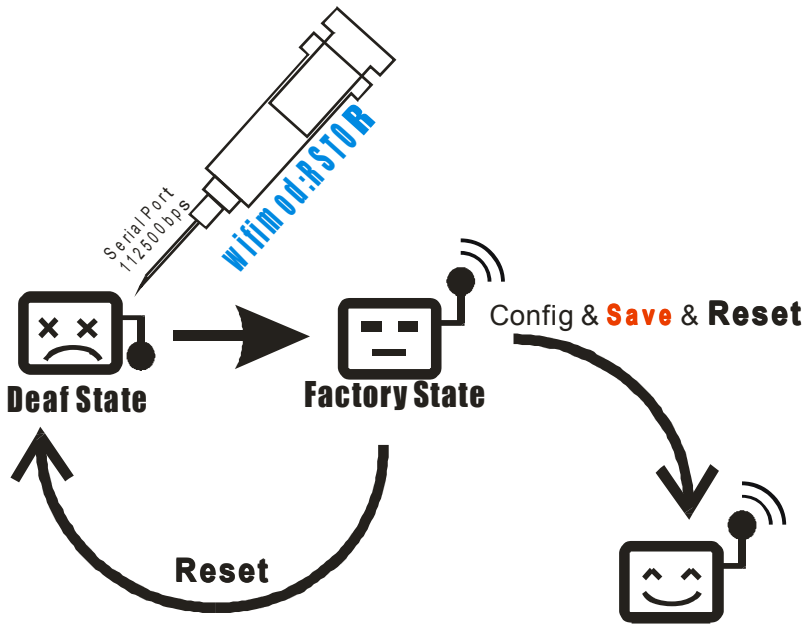
There are two ways to reset module to factory default state:

- **By sending specific string via serial port**

By sending "**wifimod:RSTOR**"(without quotes) to module's serial port, module will boot in factory default mode. And remember,these **must be done within 3 seconds after its reset**(or Power on).



The restore procedure just boot module in default mode, giving user a chance to re-configurate it.But configurate information is not modified.Without saving,a reboot will lead module to the last configurate state.



new configuration must be saved before reset.

Restore tool

- **By control module's specified Pin**

By LED_NET signal(core module pin), user can reset system to factory default state.

Keeping this signal low for 3 seconds when power on or reset, module will boot to factory default mode.

(Please refer to detailed module-related guide)

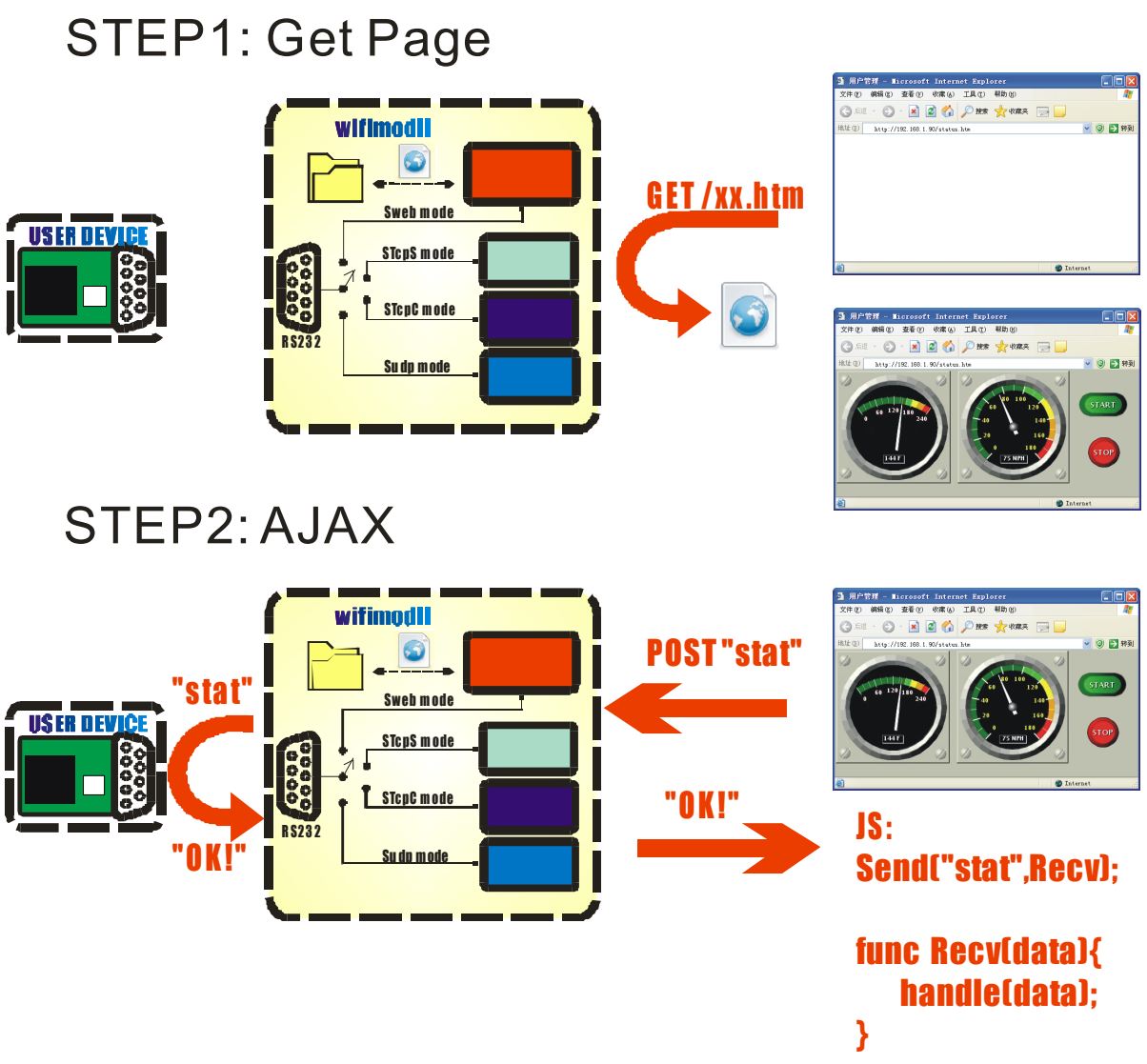
3. Work mode settings

3.1 UDP mode

3.2 TCP Server mode

3.3 TCP Client mode

3.4 SWEB mode



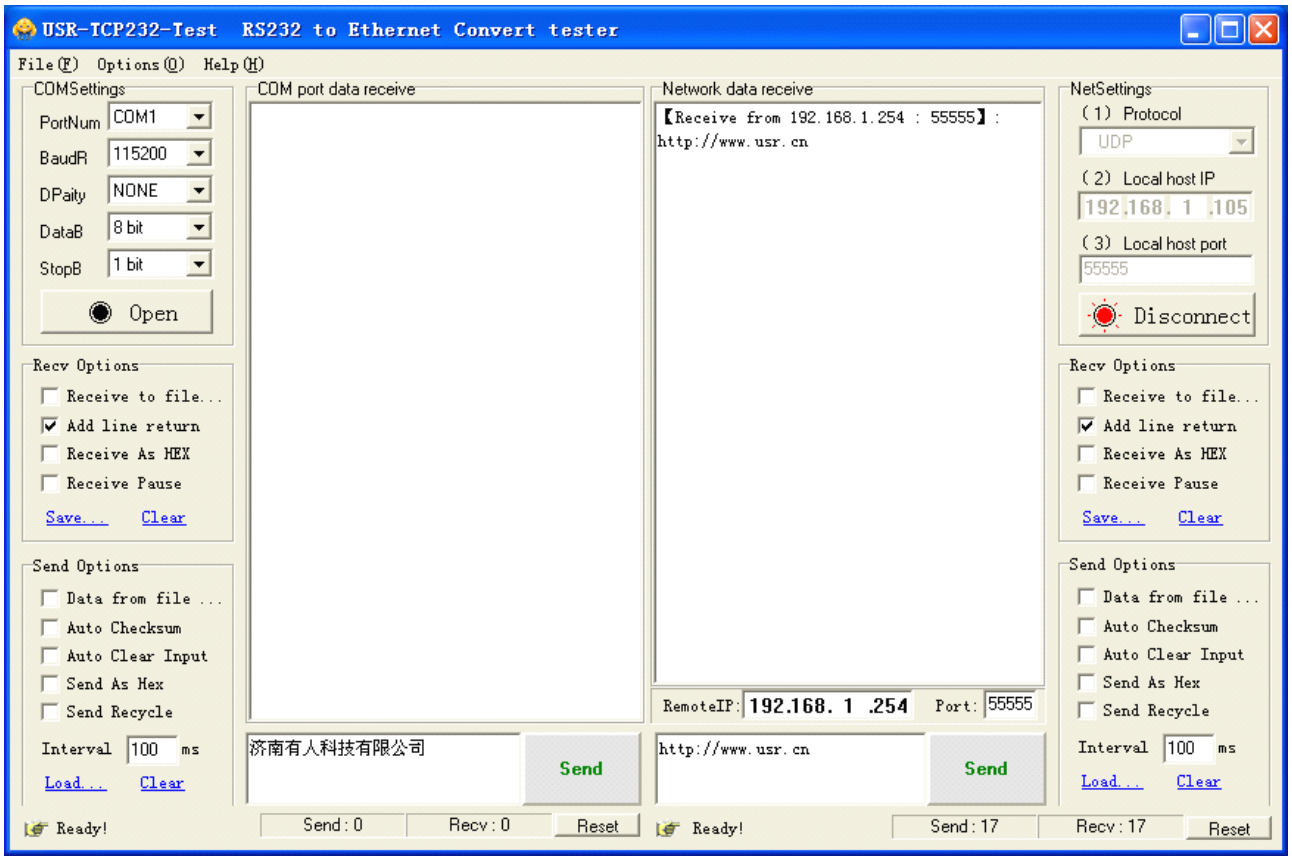
3.5 Virtual COM Port mode

4. Use sample test

4.1 UDP communication test (it is factory mode)

Test require:

- Module in factory default state with it's serial port Rx（DB9:Pin2） & Tx（DB9:Pin3）short connected.
- Laptop connected to module(See above section).
- Using TCP/IP debugging tools in UDP mode, send any text to 192.168.1.254:55555



4.2 Virtual COM Port test

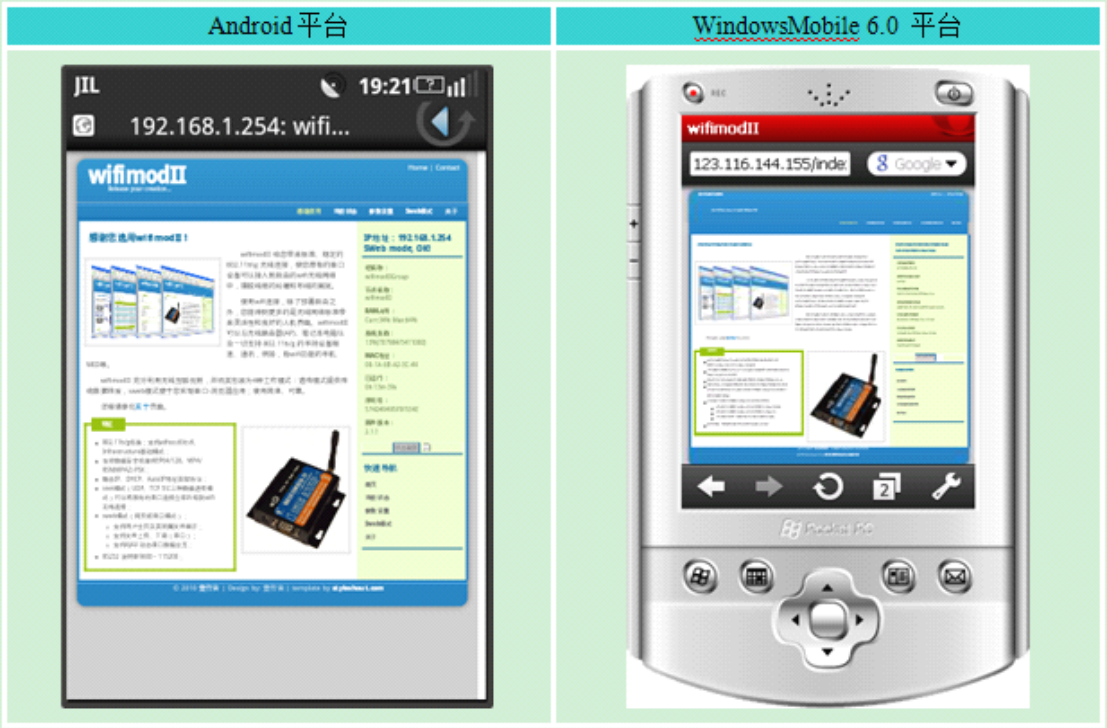
4.3 SWEB Test

Android	Windows Mobile 6.0
	

5. WEB Server in WIFIMODII

5.1 Sweb summary

Web server is an important part of WIFIMODII design,which has made the full display on the device homepage, status and function, reflecting its effect and function.In addition, WIFI also make many test and optimization to support WIFI mobile device browser, mobile phones can also get the same effect as PC.

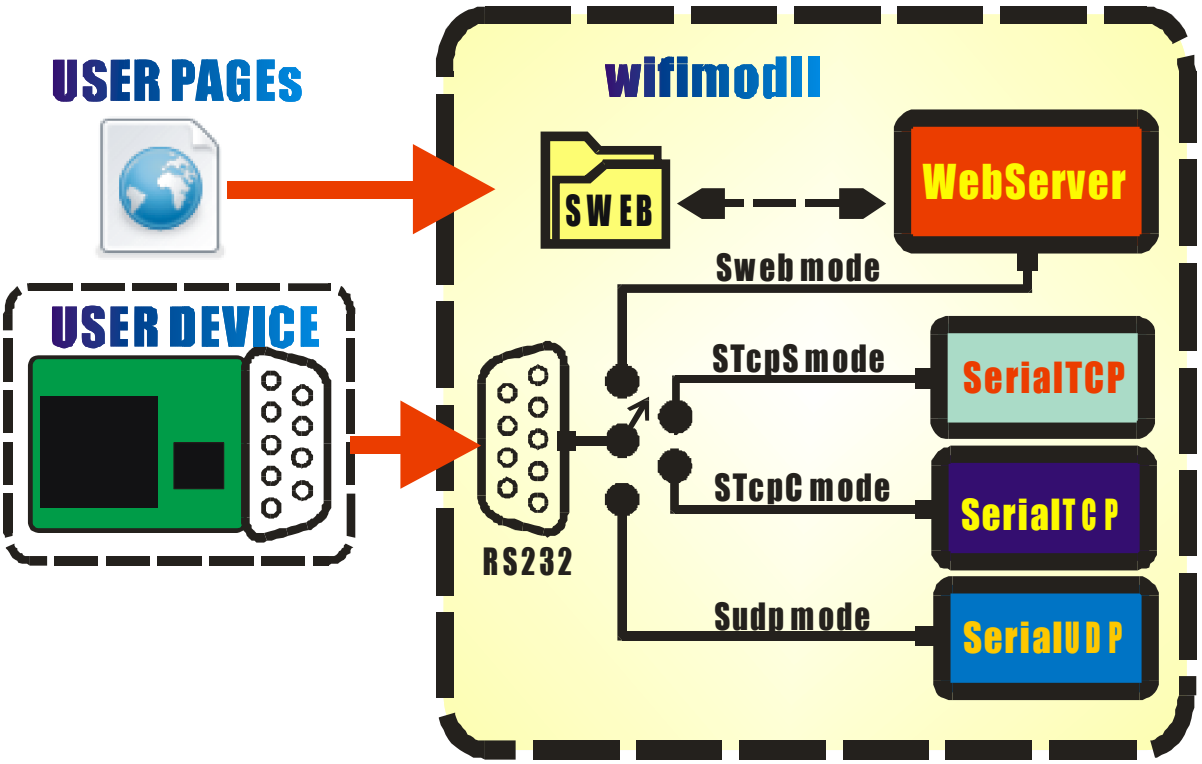


The provide of user storage space makes it possible for some applications to custom page, to write JS script logic, to place Flash, to deploy Java Applet, and sweb model makes the page logical access come into reality.

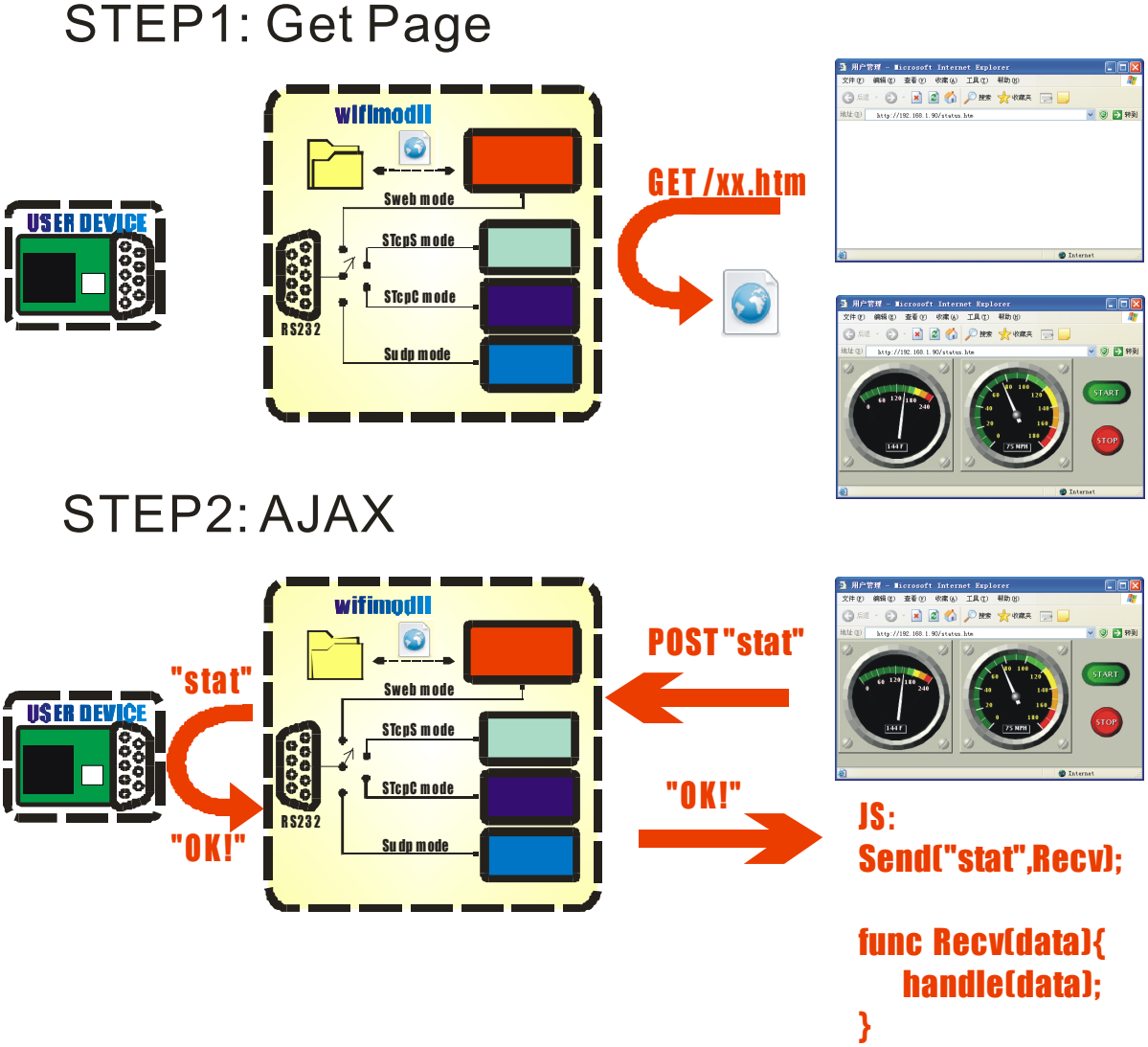
The natural principle of this Web interface application is that the server (module) disposably provides interface framework (HTML\CSS\IMG), logic (js), and many times provides dynamic data (ajax mode).

Users need to connect hardware equipment serial ports; prepare static data (design web pages and pages logic JS), plan dynamic data format (serial equipment plan command protocol). As below:

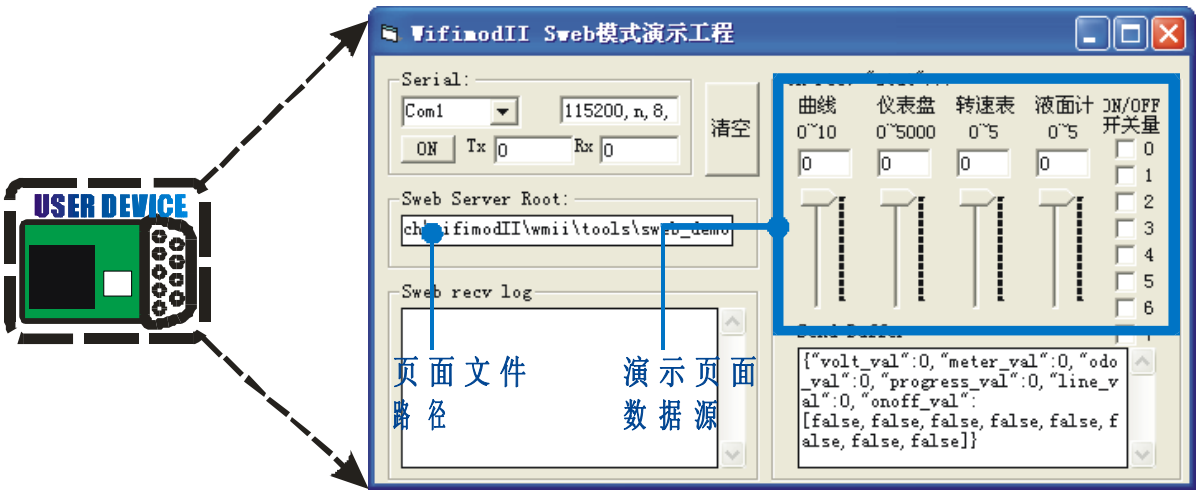
SWEB APPLICATION DEMO



Working mode as below:



WIFIMODII In "/ SWEB" directory supplies the SWEB function demo page, at the same time provides serial simulation program written by VB, used for the simulation of a serial users' port equipment, the user can view the code, to understand SWEB agreement is simple and easy to fit in.



5.2 Sweb mode core knowledge

- Sweb mode is a web server monopoly serial work mode
- The browser through the "/ sweb/" path can access to user file. (2.1.19 version)
- Whether sweb mode, the user can access "/ sweb" directory file;
- Sweb mode, the browser on the equipment IP access will redirect to/sweb/index. HTML file.
- Sweb mode, the browser through the "/ sweb. cgi" access com. For example, visit "http://192.168.1.254/sweb.cgi", page will return the com cache data;
- Support three parameters: "file" requested file; "Wait" set serial waiting time. "Clean" instructions to empty serial receive buffer; (later details)
- GET the default used for file data, if don't set wait parameters, will use the parameters in the configuration of the default HoldTime instead; POST usually is used to send data, not setting the wait parametric representing not read serial port, if setting wait = 0 means not waiting, the cache is directly read after sended;

Clean parameters used for empty serial receiving cache, setting clean parameters can eliminate the influence of clutter data before, it is suitable for "command, answer" work style,

The exact process: clear the cache; Send tip; Wait for receiving; Read the cache;Return to the page;

Wait parameter is used to specify the time of waiting for the serial port, the unit is ms.

5.3 Page views of sweb mode

WIFIMODII provide storage space that can be used to store user page file; In addition to read local data, but WIFIMODII also support from the serial port for file.

Visit the home page:

The WIFIMODII Sweb mode, the default home page (http://IP) visit will redirect to user space "/ sweb/index HTM.". The user can modify the page to realize custom user interface.

If it is not sweb mode, the default home page is the device page itself, equivalent to "http://192.168.1.254/index.htm"

Access to user space files:

The user space(/sweb directory)files can be visited through the way like "http://192.168.1.254/sweb/abc.xyz".

Through the serial port to the requested page:

Customers in the browser to com request file, link forms such as:

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

"File = usr. HTM" indicating request file named "usr HTM." (can contain path).

"Clean = 1" instructs before the feedback of the serial port, empty receiving cache first;

"Wait = 2000" parameter indicates after outputting message, com idle waiting for the longest time is 2000 ms;

In the browser address bar input the address above, WIFIMODII will empty serial receiving buffer, and output "swebget usr. HTM \ r \ n" string from the serial port (" \ r \ n "for the new line operator), after that,the data serial port receives will be returned to the browser, until in 2000 ms no more data be received

Pay attention to the browser's cache mechanism:

The default browser cache has visited pages, this feature can greatly reduce the number of the same file request; Especially sweb mode through the serial port users requests file, using the cache mechanism can reduce redundant data volume, speed up the browser's reaction speed.

WIFIMODII can realize this feature with the browser support, in sweb mode, the browser from serial requests the same file will be circular to use cache pages, in order to reduce serial be taken up; For this reason, when debug sweb mode, the browser refreshes user page in the serial port may not be requested information, if the file changes, you need to read the data, can use the Ctrl + F5 (IE) or Shift + F5 (Fire Fox) to force the page to refresh. The details in the page based on serial port development need to be paid attention to; In normal use, because the customer file generally does not change, the cache mechanism will not cause problems.

Sweb.cgi request a document with suffix name, to access to other format files, wifimodII reimbursement data, the browser will make different treatment according to different suffix, such as HTM, JPG and so on pages elements will be directly displayed, if rar, hex and so on suffix will prompt to download.

With the aid of this process, it can realize directly saving the com back data to local files through the browser downloads.

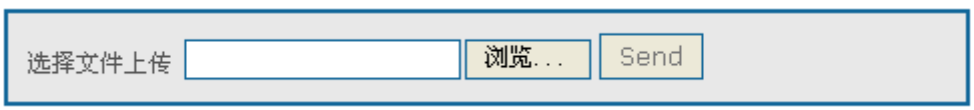
5.4 Upload files to serial port

Sweb mode can upload through the form to com file, the form is submitted in the POST way.

As following form code:

```
<form action="sweb.cgi" enctype="multipart/form-data" method="post">
Select upload files
<input type="file" name="files">
<input type="submit" value="Send">
</form>
```

The page will display:



Attention:

Form form, action target must be set as "sweb. cgi";

Form form, "enctype =" multipart/form - data "method =" post "" two attributes can't be changed;

Form form, only allow one type = "file" document control and submit control, cannot contain other input control;

5.5 Interact with serial data

Under sweb mode exchanging data with serial port is mainly through js script programming mode, using the method of AJAX + POST, in the browser background control data sending and receiving, and in the foreground to show. This way exchange data, will not be effected by the browser cache mechanism.

Quick access:

◆ AJAX's way to exchange data will not refresh the page, js script can completely control data exchange process, partly change page content after processing;

◆ AJAX requests must be launched by the browser, similar to a "command, answer" interactive way;

◆ The browser uses POST way to sweb. cgi to launch request to the implementation of the AJAX process, WIFIMODII will respond, on the one hand, the request processing request data (forward to com), on the other hand according to the parameters set to return data;

◆ AJAX transmission data, default encoding is utf-8, Chinese will be sent to the serial port in utf-8 coding form, if the serial port terminal does not support utf-8 decoding display, it will appear messy code; If the terminal can receive save as files,open in the "Windows notepad" way, can normally display("Windows notepad "support utf-8 decoding);

◆ Similarly, if the Chinese character inputted in the terminal is not sent out in utf-8 coding, the data browser receives also can appears messy code. If the end support to send the file, then it can try to send a utf-8 code files, so that the browser end to analyze the Chinese.

◆ Utf-8 in ASCII character is still single-byte and codes the same, but the Chinese character use 3 bytes;

In order to facilitate users to use, WIFIMODII factory has packed the ajax process, saved as sweb. Js file stored in/ sweb directory.

Which has been defined function is as follows:

function sweb_send_utf8(dat, cb_ok, cb_err)

Dat: send string;

Cb_ok: callback functions after sent successfully;

Cb_err: callback functions after failing to send;

This function will ask WIFIMODII just to send data to the serial port, directly back not to stay. Note that this function sends data in UTF8 coding, characters that can be printed in ASCII table is sent to the serial port in single-byte encoding,can't send pure binary data flow.

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

Dat: send string;

Wait, wait for time parameter, the unit is ms

Clean: before send data to serial port,whether clear the receiving cache;

Cb_ok: callback functions after sent successfully;

Cb_err:callback functions after failing to send;

This function will ask WIFIMODII to send data to the serial port, waiting for the specified time, return to com to receiving data.

Such as:

```
//send requested string“This is request!”
sweb_send_read_utf8(“This is request!”,500,0,cb _ok,cb _err);

function cb_ok(data)
{
    alert(dat);          //Prompt com to receive data;
}
function cb_err()
{
    alert ("---Connection Break! --- ");          //If the error, indicating that the disconnect
}
```

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

Dat: send string;

Wait, wait for time parameter, the unit is ms

Clean: before send data to serial port,whether clear the receiving cache;

Cb_ok: callback functions after sent successfully;

Cb_err:callback functions after failing to send;

This function will ask WIFIMODII to send data to the serial port, waiting for the specified time, return to com to receiving data.

Such as:

```
//send requested string“This is request!”, clear previous cache, wait for 100ms
sweb_send_read_json(“This is request!”,100,1,cb_ok,cb_err);
//serial return json form data string, such as “{“val”:100}”
function cb_ok(ret)
{
    alert(“ret value:” + ret. val);
    //Ret is parsed object, can directly through the "ret. Val" access val members
}
function cb_err()
{
    alert ("---Connection Break! --- ");          //If the error, indicating that the disconnect
}
```

5.6 Interact with serial binary data

Due to the AJAX process uses utf8 encoding mode for characters user input , so, we can't follow one's inclinations send the binary data we often use to the module of the serial port, in order to solve this problem, can use WIFIMODII built-in URL Code codec to complete the work.

URL Code coding method is very simple, it is similar to a character substitution mechanism. " " (space) replace with"+", English letters, Numbers, "-", ". ", "_", "~" as usual, besides all the other characters, use "% XY" manner directly set its hexadecimal code value.

For example, "hello China" code as "hello +%e4%b8%ad% e5%9b %bd".

WIFIMODII module support URL decoding, using this function can send arbitrary binary data between the browser and serial transmission.

Do not use URL Code mode data interaction process as follows:

Browser sending bar	Utf8 encoding	→	WIFIMODII network reception	Serial output
Abc Chinese	61 62 63 (abc) E4 B8 AD (中) E6 96 87 (文)	→	61 62 63 E4 B8 AD E6 96 87	61 62 63 E4 B8 AD E6 96 87

The way of URL Code data interaction process as follows:

Browser sending bar	Utf8 encoding	→	WIFIMODII network reception	Serial output
“%E4%B8%AD %E6%96%87”	“%E4%B8%AD %E6%96%87”	→	“%E4%B8%AD %E6%96%87”	E4 B8 AD E6 96 87

Browser sending bar	Utf8 encoding	←	WIFIMODII network reception	Serial input
“%00%01%02 %03%04%FF”	“%00%01%02 %03%04%FF”	←	“%00%01%02 %03%04%FF”	00 01 02 03 04 FF

Set req = 20, like: "sweb. cgi? Req = 20", can activate WIFIMODII built-in URL Code codec.

Module in the network side receives the data will be as URL Code coded data decoding, and then sent to the serial port;

Data receives from a serial port will go through URL Code coding and sent to the browser.

Because URL Code code characters totally are ASCII and UTF8 of intersection character part, so we can in the browser side send any binary data to the serial port module, serial port and any data can also be transmitted to the browser.

Shall be taken into consideration:

◆ In order to realize arbitrary binary data forwarding, URL Code coding former one byte of data into three bytes, increase the transmission burden.

◆ Users need to ensure that string sent to the module URL Code normative, otherwise, the module may output abnormal data;

◆ Users in the web side need to analytic string received similar to "% 01% FF" ,then can obtain the binary data. (can use the transition function provided by sweb.Js)

Sweb.js API:

function sweb_send_urlencode (dat,cb_ok,cb_err)

Dat: send string;

Cb_ok: callback functions after sent successfully;

Cb_err:callback functions after failing to send;

This function will prompte wifimodII receive URLCode data and send the data decoding forward to serial ports, directly back not to stay. The user can according to need send pure binary data flow to the serial port.

Such as:

```
//Send request URLCode coded binary string "%00%FFabcd",clear the cache before, waiting for 100 ms
Sweb_send_read_urlencode (" % 00% FFabcd ", cb_ok, cb_err);
//serial will output 0x00 0xFF 0x61 0x62 0x63 0x64


function cb_ok(ret)
{
      alert(“send ok!”);
}
function cb_err()
{
      alert ("---Connection Break! --- ");          //If the error, indicating that the disconnect
}
```

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

Dat: send string;

Wait, wait for time parameter, the unit is ms

Clean: before send data to serial port,whether clear the receiving cache;

Cb_ok: callback functions after sent successfully;

Cb_err:callback functions after failing to send;

This function will ask WIFIMODII to analytic URL Code data, and then move to serial ports, waiting for the specified time, module will coding com return data, transmitted to the browser.

Such as:

```
//send requested string"%00%FFabcd",clear cache before,wait for 100ms
sweb_send_read_urlencode ("%00%FFabcd",100,1,cb_ok,cb_err);
//serial will output 0x00 0x FF 0x61 0x62 0x63 0x64
//if serial input 0x00 0x01 0x02 0x03 in 100ms

function cb_ok(ret)
{
    alert("ret value: " + ret);
    //will output "ret value: %00%01%02%03"
}
function cb_err()
{
    alert ("---Connection Break! --- ");          //If the error, indicating that the disconnect
}
```

Application:

Send static binary data 0 x11 0 x22... 0 x99 0 x00:

```
function SendBinVal ()
    {
        sweb_send_urlencode("%11%22%33%44%55%66%77%88%99%00");;
    }
```

Binary string expressed as url list of the most simple way is add a character percent before every two Hex, that is sent "%"

Send the dynamic adjustment of binary data:

```
function SendBinVal (id, val)
        {
            var seg = new Array(3);
            seg[0] = 170;           //0xAA
            seg[1] = id;        //no more than 255
            seg[2] = val;       //no more than 255
            sweb_send_urlencode(array2url(seg));;
        }
```

The function above has id, val two parameters, structure a 3 unit array, pay attention to that each unit assignment can't be more than 255. Unit assignment, through the array2url () function (this function is sweb built-in API), the array changed into url string. Then send it out through the sweb_send_urlencode () function .

Receiving binary data:

Note urlencode standard, characters the ASCII can display are not to be coded, therefore,

Module serial port to receive:**0x12 0x34 0x56 0x78 0x9A 0xBC 0xDE 0xF0**

Forward to web pages will be:**%124Vx%9a%bc%de%f0**

0x34：‘4’

0x56：‘V’

0x78：‘x’

This problem can use sweb the js url2array () function to solve, this function use url string as input, output analytic array (unit for byte data), such as:

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

Returns an array, eight unit were 0x12 0x34 0x56 0x78 0x9A 0xBC 0xDE 0xF0

5.7 Senior content MIME types

MIME type specified

Maybe you need to specify the serial data back to the HTTP response head indicating MIME types, this request may realize through the designated req parameters, like: "sweb. cgi? Req = xx", specify the type list as follows:

Req setting	MIME type
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

More content

WIFIII module/sweb. HTM page includes the application of teaching, the user can step by step according to the guidance of learn how to use sweb mode.

In addition, in/sweb directory, also had previously offers many sweb footholds, used to display the function of all sweb mode, the user can refer to its source code.

In the "SWEB practice" section, the user can under the tutorial guidance gradually into the web design process.

We can supply customer designing web function service.

6. Contact us

Company: Jinan USR IOT Technology Co., Ltd

Address: 1-523, Huizhan Guoji Cheng, Gaoxin Qu, Jinan, Shandong, China

Tel: 86-531-55507297 86-531-88826739-803

Web: www.tcp232.net