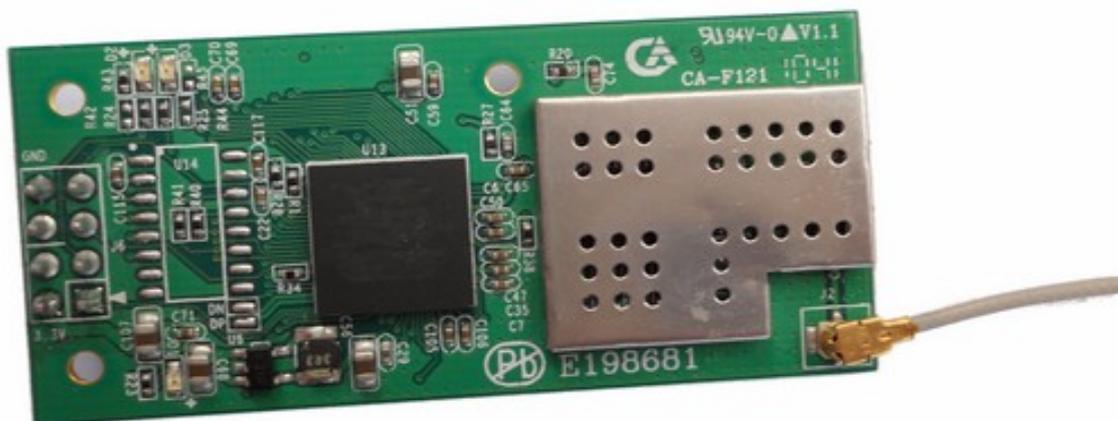


# UART WIFI TRANSPARENT MODULE



Copy Right Reserved By Elehouse

[www.elehouse.com](http://www.elehouse.com)



# Contents

|       |                                       |    |
|-------|---------------------------------------|----|
| 1     | INTRODUCTION.....                     | 6  |
| 2     | OVERVIEW.....                         | 8  |
| 2.1   | TECHNICAL SPECIFICATION.....          | 8  |
| 2.2   | HARDWARE INTRODUCTION .....           | 9  |
| 2.2.1 | Mechanical Dimension .....            | 9  |
| 2.2.2 | External Interface.....               | 10 |
| 2.2.3 | Antenna (optional) .....              | 11 |
| 2.2.4 | Serial Adapter (optional) .....       | 12 |
| 2.2.5 | Wireless adapter (optional) .....     | 13 |
| 2.3   | MAIN FEATURES .....                   | 14 |
| 2.3.1 | Wireless Network Connection.....      | 14 |
| 2.3.2 | Plug and Play Serial.....             | 16 |
| 2.3.3 | Command Mode of Serial.....           | 17 |
| 2.3.4 | TCP/IP Protocol stack.....            | 17 |
| 2.3.5 | Flexible Parameter Configuration..... | 18 |
| 2.3.6 | Multi-functional GPIO.....            | 19 |
| 2.3.7 | Firmware upgrade online .....         | 19 |
| 3     | QUICK START WIZARD .....              | 20 |
| 3.1   | PREPARATION.....                      | 20 |
| 3.2   | SETTING PARAMETERS.....               | 20 |
| 3.3   | TESTING .....                         | 22 |
| 4     | PRODUCT DESIGN GUIDE .....            | 24 |
| 4.1   | HARDWARE CONNECTION.....              | 24 |
| 4.2   | WORK MODE .....                       | 24 |
| 4.2.1 | Start-up mode selection .....         | 24 |
| 4.2.2 | Auto-work mode.....                   | 25 |
| 4.2.3 | Command work mode.....                | 29 |
| 4.3   | PARAMETER DETAILS.....                | 30 |
| 4.3.1 | Wireless parameters .....             | 30 |
| 4.3.2 | Network Parameters.....               | 32 |
| 4.3.3 | Parameters of work mode .....         | 33 |
| 4.3.4 | Advanced wireless parameters.....     | 35 |
| 4.3.5 | Port parameters .....                 | 39 |
| 4.3.6 | Parameters of transparent mode.....   | 40 |

---

|       |  |     |
|-------|--|-----|
| 4.3.7 | System parameters .....  | 41  |
| 4.4   | DEFAULT SETTING .....  | 43  |
| 4.5   | TYPICAL APPLICATION EXAMPLES.....  | 45  |
| 4.5.1 | The networking of active serial device .....                             | 45  |
| 4.5.2 | The networking of passive serial device .....                            | 46  |
| 4.5.3 | The networking with broadcast method of serial device .....              | 47  |
| 4.5.4 | Wireless virtual serial .....  | 49  |
| 4.5.5 | Multi-network connection mode.....                                       | 51  |
| 5     | CONFIGURATION MANAGEMENT MANUAL .....                                    | 54  |
| 5.1   | CONFIGURATION THROUGH CONFIGURATION MANAGEMENT SOFTWARE .....            | 54  |
| 5.1.1 | Introduction.....  | 54  |
| 5.1.2 | Module management.....   | 54  |
| 5.1.3 | Configuring parameter .....  | 56  |
| 5.1.4 | Function test.....   | 60  |
| 5.1.5 | System information.....  | 69  |
| 5.1.6 | Firmware upgrade .....   | 69  |
| 5.2   | CONFIGURATION THROUGH WEB SERVER.....                                    | 70  |
| 5.2.1 | Query/set configuration parameter .....                                  | 71  |
| 5.2.2 | Firmware upgrade on line .....   | 72  |
| 5.3   | CONFIGURATION THROUGH SUPER TERMINAL PROGRAM .....                       | 72  |
| 6     | AT+INSTRUCTION CONTROL PROTOCOL.....                                     | 75  |
| 6.1   | SYNTAX INTRODUCTION.....   | 75  |
| 6.1.1 | Syntax Format.....   | 75  |
| 6.2   | INSTRUCTION SET.....   | 78  |
| 6.2.1 | Instruction List.....  | 78  |
| 6.2.2 | Network Control Type .....   | 80  |
| 6.2.3 | System control type .....  | 85  |
| 6.2.4 | Parameter Setting Type .....   | 88  |
| 6.3   | SAMPLE CODE.....   | 99  |
| 6.3.1 | Description of sample code.....  | 99  |
| 6.3.2 | Operation flow of WiFi module in command mode .....                      | 99  |
| 6.3.3 | main data framework of sample code .....                                 | 100 |
| 6.3.4 | Operation of serial data.....  | 101 |
| 6.3.5 | The function used to parse the return parameters of AT instruction ..... | 105 |
| 6.3.6 | The function of operating character string.....                          | 107 |
| 6.3.7 | The parsing sample of several AT instructions sending.....               | 109 |

---

|  |     |
|--|-----|
| APPENDIX A FREQUENT QUESTIONS AND ANSWERS.....   | 118 |
| Q1: WHY CAN NOT THE MODULE SEARCH FOR THE SPECIFIED NETWORK? A: THE CHANNEL LIST USER SET DOESN'T CONTAIN THE CHANNEL WHICH THE SPECIFIED NETWORK WORK IN.                                     | 118 |
| Q2: WHY CAN NOT THE MODULE JOIN IN THE NETWORK?.....   | 118 |
| Q3: WHY DOES THE MODULE NEED ABOUT 10 SECONDS TO RESPONSE THE COMMAND OR CONNECT TO NETWORK AUTOMATICALLY, WHEN ENCRYPTION MODE IS SET TO WPA/WPA2?.....                                       | 118 |
| Q4: THE MODULE HAS CONNECTED TO THE NETWORK SUCCESSFULLY,WHY CAN'T IT SHOW "DESTINATION HOST UNREACHABLE" OR "REQUEST TIMED OUT" AFTER DOING THE OPERATION PING BETWEEN THE MODULE AND PC..... | 119 |
| Q5: IN AUTO-WORK MODE,THE ICMP CONNECTION SOMETIMES CONNECTS AND SOMETIMES DISCONNECTS AFTER THE MODULE HAS JOINED IN THE NETWORK,WHY?.....  | 119 |
| Q6: IN AUTO-WORK MODE,WHY DOES THE MODULE NOT CONNECT TO THE NETWORK AUTOMATICALLY AFTER RESETTING?.....   | 119 |
| Q7: IN AUTO-WORK MODE,DATA SENT TO THE MODULE THROUGH SERIAL WILL DELAY SOMETIME TO SEND TO THE NETWORK,WHY?.....  | 120 |
| Q8: IN AUTO-WORK MODE,WHY DOES THE MODULE NOT EXIT THE TRANSMISSION STATE, AFTER INPUTING THE ESCAPE CHARACTER?.....   | 120 |
| Q9: WHY CAN NOT THE CONFIGURATION PROGRAM SEARCH FOR THE TARGET MODULE?.....   | 120 |
| Q10: TO MODIFY THE PARAMETERS WITH CONFIGURATION PROGRAM,WHY DOES IT SHOW THE ERROR 'USED PROHIBITIVE OPERATING SYMBOL'?.....  | 120 |
| Q11: WHY CAN NOT WEB PAGES OF THE WEB SERVER OPEN?.....  | 120 |
| Q12: WHEN USER SETS THE SERIAL TO THE PAUD RATE IT DOES NOT SUPPORT, THE MODULE CAN'T BE CONNECTED TO, THEN HOW TO CORRECT IT ?.....   | 121 |
| Q13: IN WIRELESS CONNECTION STATE,WHY CAN NOT THE SPECIFIED MODULE BE SEARCHED FOR?.....   | 121 |
| Q14: IN SOME ENVIRONMENT THE MODULE USED, WHY IS THE QUALITY OF WIRELESS CONNECTION ALWAYS BAD,AND HOW TO IMPROVE THE WIRELESS TRANSMISSION?.....  | 122 |
| APPENDIX B REVISION HISTORY OF THE DOCUMENT.....   | 123 |

## 1 Introduction

VSD03 is the new third-generation embedded Uart-Wifi modules studied by VSD TECH. Uart-Wif is an embedded module based on the Uart serial, according with the WiFi wireless WLAN standards, It accords with IEEE802.11 protocol stack and TCP / IP protocol stack, and it enables the data conversion between the user serial and the wireless network module. through the Uart-Wifi module, the traditional serial devices can easily access to the wireless network.

VSD03 does a comprehensive hardware and software upgrades based on the products of the first two generations, now it's more functional and more Convenient to use, its main features include:

### Interface:

- 2 2\*4 pins of Interface: HDR254M-2X4
- 2 The range of baud rate: 1200~115200bps
- 2 RTS / CTS Hardware flow control
- 2 single 3.3V power supply

### Wireless

- 2 support IEEE802.11b / g wireless standards
- 2 support the range of frequency: 2.412~2.484 GHz
- 2 support two types of wireless networks:
  - Ad hoc and Infrastructure
- 2 support multiple security authentication mechanisms:
  - WEP64/WEP128/ TKIP/CCMP(AES)
  - WEP/WPA-PSK/WPA2-PSK
- 2 support quick networking
- 2 support wireless roam

### Others

- 2 support multiple network protocols
  - TCP/UDP/ICMP/DHCP/DNS/HTTP
- 2 support two types of work modes:
  - auto and command
- 2 Support the transparent transmission mode
- 2 support AT+ instruction set

- 2 support a variety of parameters configuration methods:
  - serial / WEB server / wireless configuration

### **Application**

- 2 intelligent bus network, such as wireless credit card machine
- 2 small financial payment network, such as wireless POS machine
- 2 industrial equipment networking, such as wireless sensor
- 2 Things

## 2 Overview

### 2.1 Technical Specification

Table2-1 Technical Specifications

|                  | Item                        | Parameters   |
|------------------|-----------------------------|--|
| <b>wireless</b>  | Wireless standard           | IEEE802.11b/g  |
|                  | Frequency range             | 2.412~2.484 GHz  |
|                  | Receiver sensitivity        | 802.11b: -86 dBm @ 11Mbps (typical)<br>802.11g: -71 dBm @ 54Mbps (typical) |
|                  | Transfer rate               | 802.11b: 1,2,5.5,11 Mbps<br>802.11g: 6,9,12,18,24,36,48,54 Mbps            |
|                  | Madulation                  | DSSS, OFDM, DBPSK, DQPSK, CCK, QAM16/64                                    |
|                  | Output power                | 802.11b: 18±2 dBm (typical)<br>802.11g: 15±1 dBm (typical)                 |
|                  | Antenna Interface           | IPX  |
| <b>Hardware</b>  | Serial type                 | UART   |
|                  | Serial rate                 | 1200~115200 bps  |
|                  | Operating voltage           | 3.3±0.3 V  |
|                  | Operating current           | 300mA (typical)  |
|                  | Storage temperature         | -40~+85 °C   |
|                  | Operating temperature       | 0~75 °C  |
|                  | Dimensions                  | 55.2×25.4×11.5 mm  |
|                  | Environmental certification | RoHS   |
| <b>Soft ware</b> | Network type                | Infra/Adhoc  |
|                  |                             | WEP/WPA-PSK/WPA2-PSK   |

|  |                          |   |
|--|--------------------------|---|
|  | Security                 | WEP64/WEP128/TKIP/CCMP(AES)   |
|  | Work mode                | auto/command  |
|  | Serial command           | AT+instruction set  |
|  | Network protocol         | TCP/UDP/ARP/ICMP/DHCP/DNS/HTTP  |
|  | Most sockets to connect  | 15  |
|  | TCP connection           | the most connection: 8<br>the most Client: 8<br>the most Server: 3~4<br>the most client to connect to server built in this module : 4 |
|  | UDP connection           | the most connection5~6  |
|  | The maximum sending rate | 11Kbytes /s(TCP)  |
|  |                          |   |

## 2.2 Hardware introduction

### 2.2.1 Mechanical Dimension

The mechanical dimension of VSD03 is shown as follow, thereinto, figure 2-1 is the top view of module, 2-2 is the side view of the module.

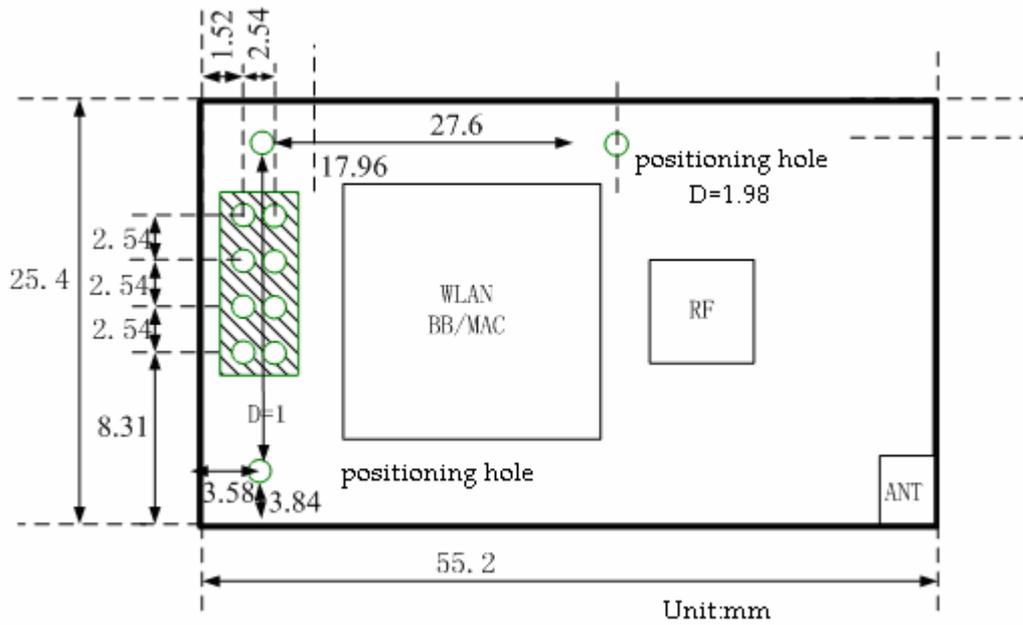


Figure 2-1 top view

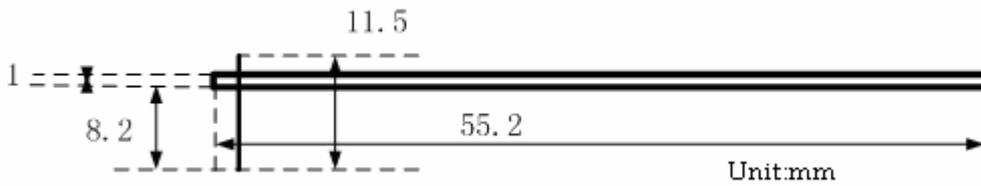


Figure 2-2 side view

### 2.2.2 External Interface

The following figure is the external interface diagram of VSD03 product.

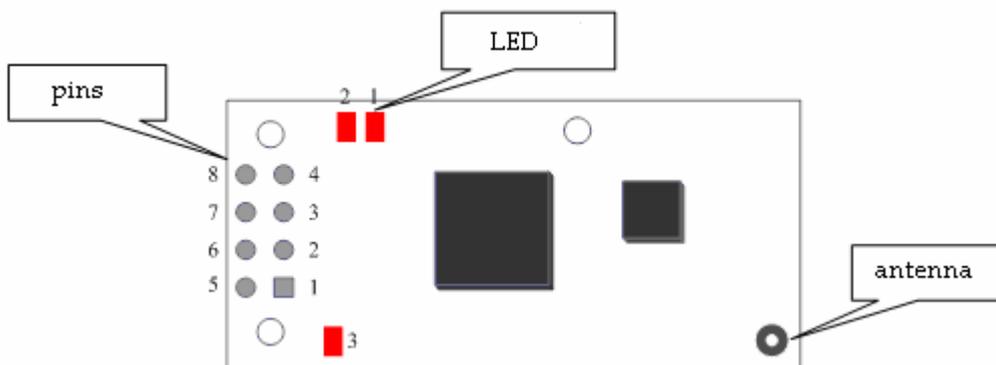


Figure 2-3 interface diagram

### 2.2.2.1 Pin Interfaces

The product supports 8-pin dual in-line as Figure 2-3, they are defined as follows:

Table 2-2 Pin Interfaces

| No | function           | direction | descreption  |
|----|--------------------|-----------|--|
| 1  | nRTS/nREADY        | O         | Optional features pin, detailed definition see Section 2.3.6,and it's connected to LED1        |
| 2  | nCTS/MODE/GPI<br>O | I/O       | Optional features pin, detailed definition see Section 2.3.6,and it's connected to LED1        |
| 3  | Link               | O         | Optional features pin,it's the designation of wireless transmission,and it's connected to LED3 |
| 4  | nRESET             | I         | Optional features pin,it's a reset signal and effective when it's low.                         |
| 5  | VDD                | I         | The pin of 3.3V input  |
| 6  | RXD                | I         | the pin receives data from the serial  |
| 7  | TXD                | O         | the pin sends data to the serial   |
| 8  | GND                | I         | ground   |

Notices:

- 1、 The lateral pins (5 ~ 8) can be used for the pins interface of TLG09UA01/02 in single row .
- 2、 The medial pins (1 to 4) is optional feature pins, when they are not in use they can be connected to nothing.

### 2.2.2.2 Antenna interface

This product provides one antenna interface:IPX.

### 2.2.3 Antenna (optional)

This product must be connected with the 2.4G antenna according to IEEE 802.11g / 802.11b standards, the parameters are required as follows:

Table 2-3 antenna parameters

| No | description |
|----|-------------|
|----|-------------|

|                 |                |
|-----------------|----------------|
| Frequency range | 2.4~2.4825 GHz |
| Impedance       | 50 Ω           |
| VSWR            | ≦ 1.5          |

### 2.2.4 Serial Adapter (optional)

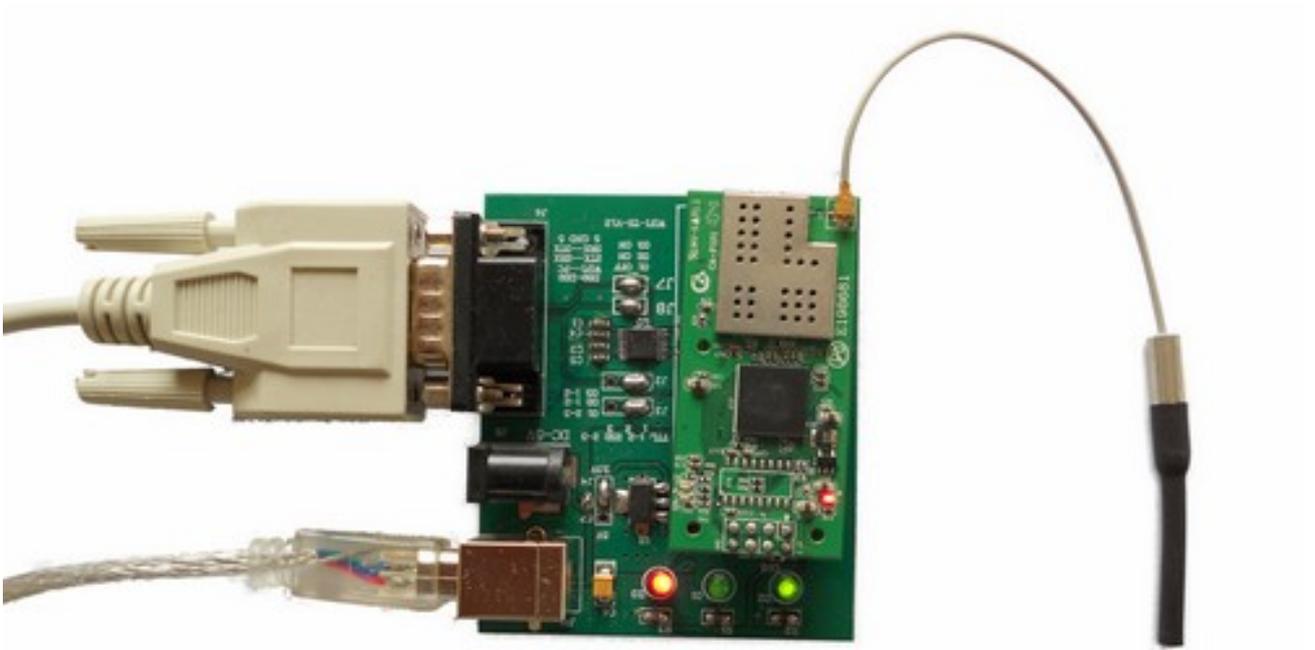


figure2-4 serial adapter

Through the adapter board shown above, you can connect this product with the serial port of PC, then user can configure the parameters with the configuration and management program and do the function tests.

Ø Interface description:

Table 2-4 description of interface on adapter board

| Interface | description  |
|-----------|--|
| USB       | B-type interface, only work in 5V input                |
| DB9       | female serial jack of 9-pin, and used to connect to PC |
| PIN       | 2x4 jack, and used to connect with WiFi module         |

Ø Skipping connection description:

Table 2-5 description of skipping connection on adapter board

| No | description  |
|----|--|
| 1  | The choice of power input.left: 5V; right: 3.3V      |
| 2  | TX signal choice. left: TTL; right: RS232            |
| 3  | RX signal choice. left: TTL; right: RS232            |
| 4  | CTS signal choice. Left: ground;right:connect to DB9 |
| 5  | RTS signal choice. right: connect to DB9             |

Note:when using VSD03 product,please use the default choices as figure.

### Ø LED light description

Table 2-6 LED light description

| No | description                       |
|----|-----------------------------------|
| 1  | nRTS signal , as LED1             |
| 2  | nCTS signal, as LED2              |
| 3  | Wireless transfer signal, as LED3 |

### Ø button description

Table 2-7 button description

|              | description   |
|--------------|---|
| Reset button | Connect to nRESET signal, and be used to reset the module |

## 2.2.5 Wireless adapter (optional)



Figure 2-5 Wireless adapter

Wireless adapter can be used to configure paraments of the product through the wireless connection, which is more useful when the products can't connect to serial port.In addition, parameter configuration of a group of products can be achieved by using the wireless adapter. the method for use is shown in Section 3.2.

## 2.3 Main features

### 2.3.1 Wireless Network Connection

#### 2.3.1.1 Network Topology

Wifi wireless network includes two types of topology: basic network (Infrastructure), and adhoc networks (Adhoc).to explain the wireless network topology, two basic concepts are needed to be understood first ,

AP, that is the wireless access point, the founder of a wireless network and the center of the network nodes. The wireless router we use at home or in office may be an AP.

STA, short for station, each terminal connects to a wireless network(such as laptops, PDA and other networking devices) can be called with a STA devices.

#### 2 Basic wireless network based on AP(Infrastructure)

Infrastructure:it's also called basic network.It's built by AP and many STAs which join in. The characters of network of this type is that AP is the center,and all communication is transmitted through the AP .

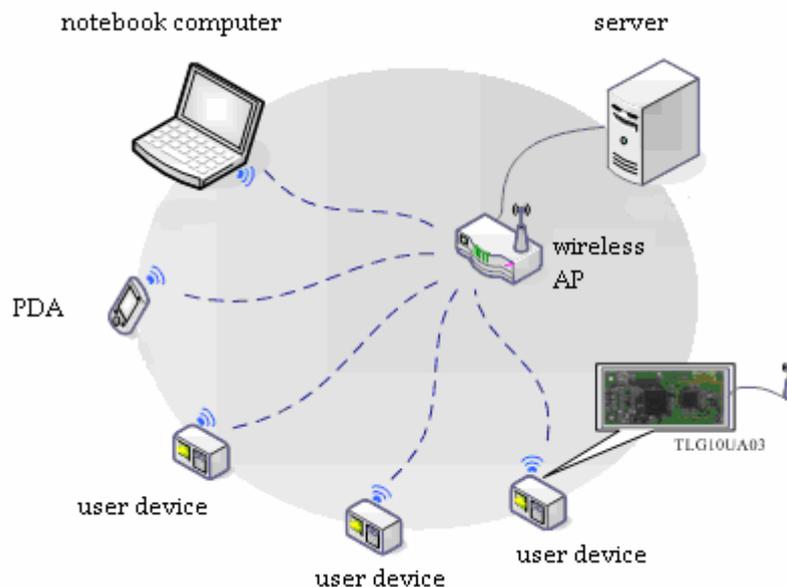


figure2-6 Infrastructure topology

#### 2 wireless network based on ad hoc network (Adhoc)

Adhoc: it's also called independent basic service set,and it's built by two or more STAs without AP, this type of network is a loose structure, all the STAs in the network can communicate directly.

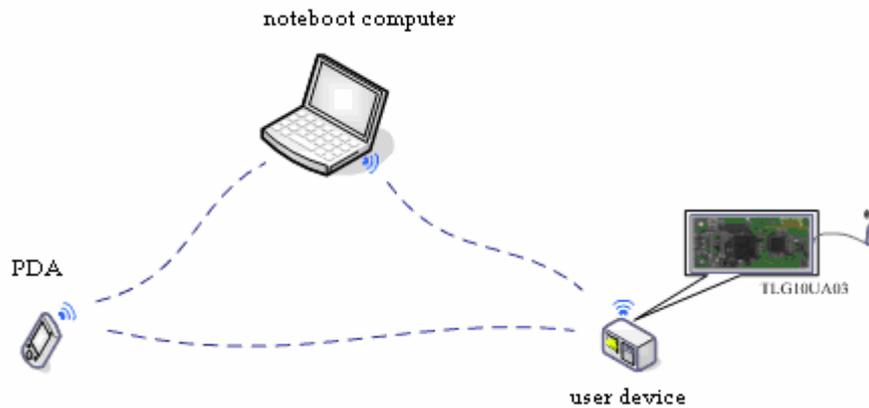


figure2-7 ad hoc network topology

VSD03 works as a STA, and supports the two types of wireless networks above.

#### 2.3.1.2 Security

The module supports multiple wireless encryption mechanisms, and enables to protect the security of user's data transmission, the mechanisms include,

- 2 WEPA64
- 2 WEPA128
- 2 WPA-PSK/TKIP
- 2 WPA-PSK/CCMP
- 2 WPA2-PSK/TKIP
- 2 WPA2-PSK/CCMP

#### 2.3.1.3 Quick networking

The module supports quick networking by specifying channel number. In the usual course of wireless networking, devices would first scan automatically on the current channel, in order to search for the network (or Ad hoc) built by the target AP. This module provides working channel configuration, when the channel of the target network is known, users can specify the working channel directly, the networking time will be reduced from 2 seconds to about 300 milliseconds, then quick networking is achieved.

#### 2.3.1.4 Address binding

This module supports the feature of binding the BSSID address of target network. According to the provisions of 802.11 protocol, different wireless networks can have a

same network name (ie SSID / ESSID), but must correspond to a unique BSSID address. Illegal intruders can create a wireless network with the same SSID / ESSID, it will make STAs in the network to join to the illegal AP, thereby and then network leakage happens. Users can prevent STA from joining to illegal network by binding the BSSID address, to improve wireless network security.

#### 2.3.1.5 Wireless Roam

This module supports 802.11 wireless roam. Wireless Roam is that, in order to expand the coverage of a wireless network, multi-AP co-found the wireless network with a same SSID/ESSID, each AP is used to cover different area, STA joined in the network can choose a proximate AP (namely signal is the strongest) to access according to different location, and mobile STA will switch among the different APs .

When wireless roam is enabled, the module will automatically initiate a scan in every 100 seconds, and according to the current scan result, module selects a recent AP to access. During the wireless roaming, the network may be a brief interruption, therefore, it is recommended to close as no need of roam.

What should be noted is that, when BSSID address is binded, wireless roaming is not effective anymore.

#### 2.3.1.6 Region Setting

The module supports wireless region setting. Users can set the wireless channel list to determine the channels of scanning range.

### 2.3.2 Plug and Play Serial

The module supports serial transparent transmission and auto-work mode, it really achieve the serial plug and play, and it does the greatest degree to reduce the use complexity. In this mode, users only need to pre-set the required network parameters as follow:

- ② The paraments of wireless network, include:
  - network name(SSID);
  - security mode;
  - Key(if it's need);
- ② default TCP/UDP paraments, include:
  - protocol type
  - connetion type
  - server address
  - port number

Since then, after its power is on, module can automatically connect to the default wireless network and server each time. As in the auto mode, the module's serial always

work in the transparent transfer state, so users only need to think of the module as a virtual serial cable, and send or receive data as using a simple serial. In other words, the serial cable of users' original serial devices is directly replaced with the module, user equipments can be easy for wireless data transmission without any changes.

In this mode, the module supports the trigger length and trigger time of auto forming a frame, Trigger length means that the module receives from the serial a specified length of data, then forms a wireless data frame, and immediately start the network sending process. Trigger time means, when the data received from the serial is less than the specified length, the module will force to form a frame with the current data and start to send after the specified time.

In this mode, the module serial supports hardware flow control, provides normal RTS /CTS signals, it's strongly recommended when doing large amounts of data, hardware flow control should be enabled, so as to fully ensure reliable data transmission. In the applications which doesn't need flow control, users can simply leave RTS / CTS pin vacant.

### **2.3.3 Command Mode of Serial**

Module also provides the command mode based on serial control, which is different from auto-work mode, this mode provides the user with sufficient flexibility, and meet users' specific needs of different applications.

In this mode, the module works according to user command through the serial, user can control the module completely with the command, including modifying parameters, controlling networking, controlling TCP/IP connection and data transmission. This is a high-level use, and also the most flexible to use. Users can discretionarily control the connection and disconnection of the wireless network with command, also can create several different types of TCP / IP connections at the same time and keep them transferring. However, this model also requires the user has full understanding of control protocol of the module's AT+Instruction set, and have a basic use knowledge of wireless network and TCP/IP network.

This module beds a set of AT+Instruction used for system control and parameter configuration command (see Article VI), all instructions are based on ASCII code, using Windows Super Terminal program can directly control module with command, it's convenient for debugging and using.

### **2.3.4 TCP/IP Protocol stack**

This module contains a complete TCP / IP protocol stack, and it supports the TCP/UDP /ICMP/ARP/DHCP/DNS/HTTP protocol, in addition, it also supports programming

interfaces of socket based on AT+Instruction set.

- 2 supports 8 TCP client connections at most
- 2 supports 3 TCP server connections at most, and the most number of clients which can join in one server is 4
- 2 supports 5~6 UDP connections at most
- 2 supports UDP broadcasting
- 2 supports 8 TCP connections at most
- 2 supports DHCP Client
- 2 supports DNS Resolver
- 2 supports HTTP,built-in WEB server
- 2 supports programming interfaces of socket

### 2.3.5 Flexible Parameter Configuration

This module commonly works based on the default parameters, configured parameters are saved in the flash memory, parameters can be saved when power is down, the user can use a variety of methods to modify the configuration parameters, they include:

- 2 *on the basis of wireless connection, using the configuration management program (recommended)*

Use the dedicated wireless adapter and connect it with the module through wireless connection, and then run the configuration management application we provide, you can search and configure the parameters of the module, the method of operation and interface are the same with those when serial connection is used. Advantages of this approach are: the first is **anytime**, that means module doesn't need to connect to any wireless network, as long as the module is power-on, parameters can be modified; the second is **anywhere**, without any physical cable connection, even if the module has been integrated into the user equipment, parameters can also be modified; the third is **group configuration**, it means a group of modules can be uniformly modified once, which is more useful to do a unified factory configuration to devices for users,it can greatly improve operation efficiency.

- 2 *on the basis of serial connection,using the configuration management program*

It needs an adapter plate to connect the module to the serial of the PC, and then run the configuration management application, this approach has the advantage of intuitive interface and easy operation

- 2 *on the basis of serial connection,using Windows HyperTerminal program*

It needs an adapter plate to connect the module to the serial of the PC, and then run the Windows HyperTerminal program, and do the parameters configuration with AT+Instruction set, this approach is the most flexible, but it needs users are familiar with AT+Instruction set.

<sup>2</sup> on the basis of network connection, using IE browser program

This approach requires the module has connected to a wireless network. On the PC connected to the same wireless network, use IE browser to connect the WEB server of the module. This approach has the advantages of simple operation and intuitive interface.

### 2.3.6 Multi-functional GPIO

This module provides two multi-functional pins which can be multiplexed:

Table 2-8 GPIO Multi-functions

| pin |        | function   |
|-----|--------|--|
| 0   | nRTS   | Output, RTS signal, in auto-work mode, it as the RTS signal of this side can directly connect with the CTS signal of the other side  |
|     | nREADY | Output. In command mode, it indicates the state of wireless network connection of the module, low level is connected, high is disconnected   |
| 1   | nCTS   | Input. CTS signal, in auto-work mode, it as the CTS signal of this side can be directly connected to the RTS signal of the other side  |
|     | MODE   | Input. it means the startup mode selection of the module, In the start-up phase (power-on / reset within 300ms), low level means common work mode, high level means configuration mode (referred in Section 4.2.1) |
|     | GPIO   | Input/output, it can works as general input or output pin according to user configuration  |

### 2.3.7 Firmware upgrade online

The module supports firmware upgrade online, there are two methods to upgrade the firmware:

<sup>2</sup> on the basis of serial connection, using the configuration management program

- 2 on the basis of network connection, using the IE browser program

### 3 Quick Start Wizard

#### 3.1 Preparation

##### n Hardware

- 2 VSD03 module
- 2 serial adapter board
- 2 serial cable

##### n Software

- 2 UART-WIFI configuration management program
- 2 serial debugging software, such as serial debugging assistant, serial master



- 2 TCP/UDP debugging software, such as TCP/UDP test tool (  TCPUDPDbg.exe ) and so on.

#### 3.2 Setting parameters

- 1、 Run the UART-WIFI configuration management program.
- 2、 Connect VSD03 module with Serial adapter board, and then turn the power on after the serial cable has been connected, as shown below:

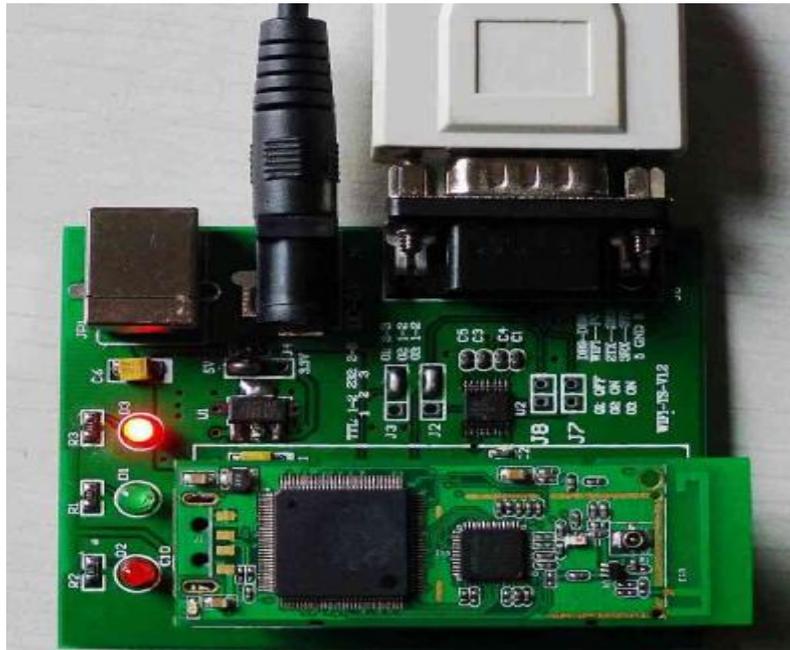


Figure 3-1 connection between with serial adapter board and UART-WIFI module

3、click "search modules", as shown below:

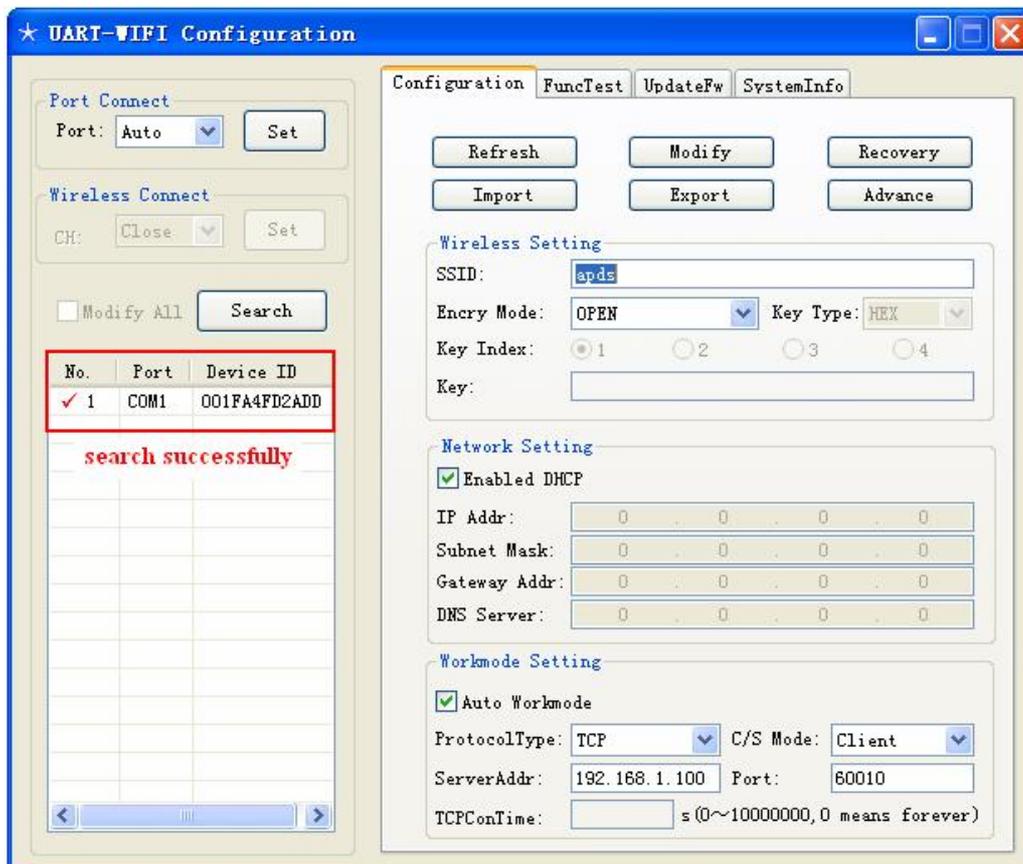


Figure 3-2 Search modules successfully

4. According to the wireless router (AP) that you connect to, you can modify the parameters of wireless settings, including network name, encryption, keys and the IP address of network settings.
5. Modify the working mode settings, select the "enable auto-work mode" option and set the parameters according to what you want.
6. After modifying parameters as shown below, click "submit" and select "manually reset later" in the pop-up dialog box.

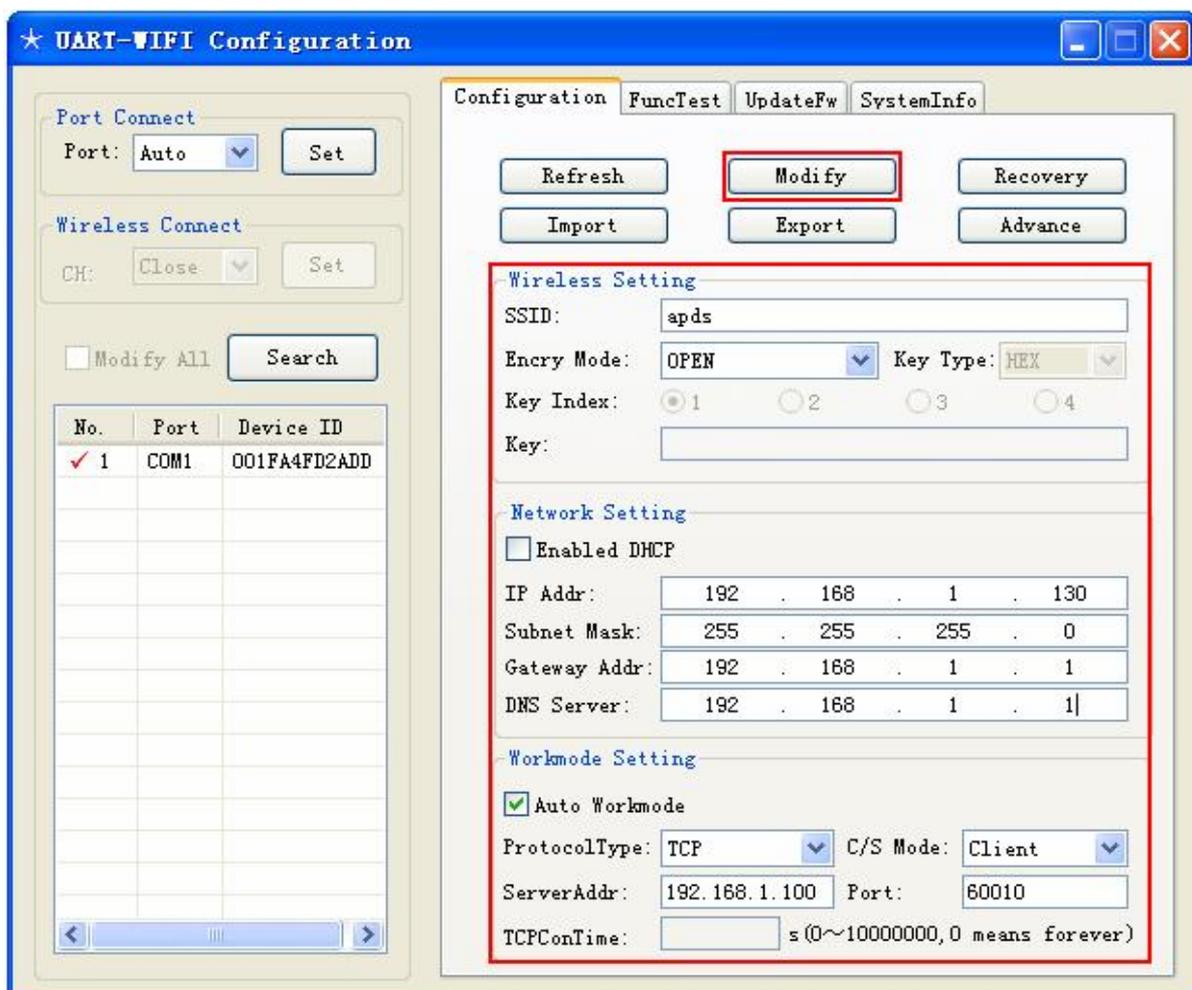


Figure3-3 Modify parameters of UART-WIFI module

### 3.3 Testing

1. Run TCP / UDP debugging software, create a TCP Server whose detecting port is 60000.

- 2、 Run serial debugging software, such as RealTerm.exe in this case. Modify parameters.

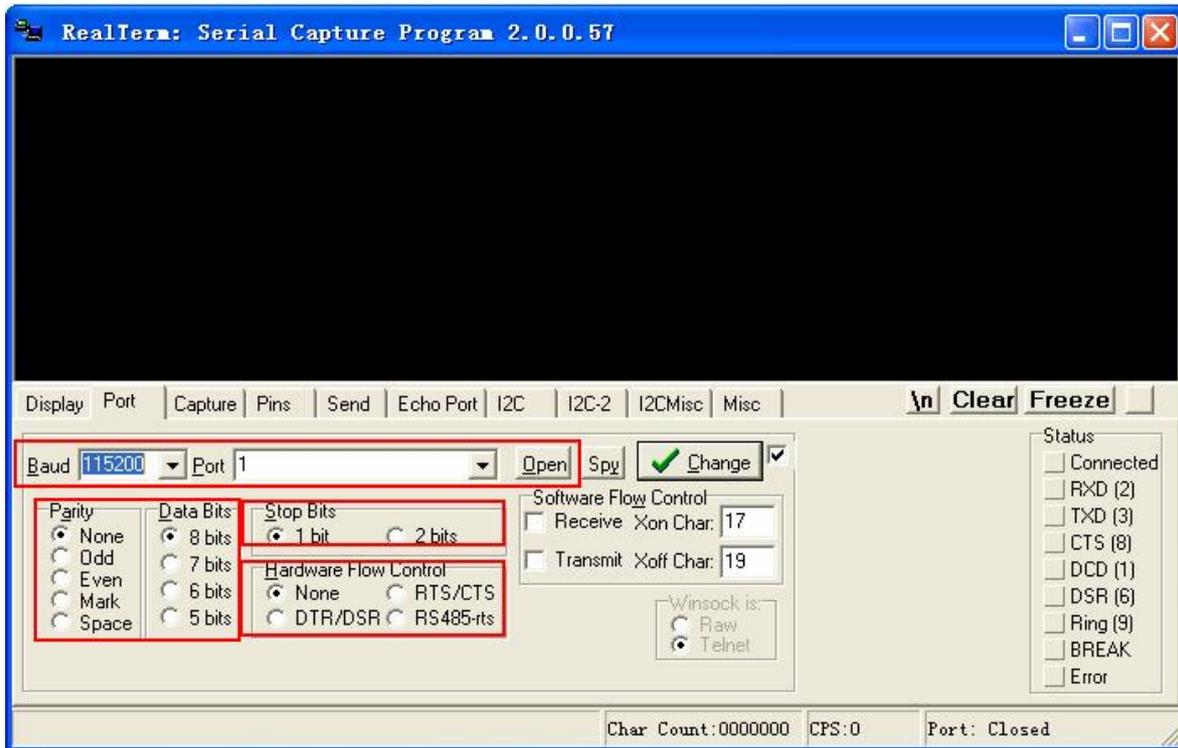


Figure3-5 modify the parameters of ComMaster

- 3、 Repower the serial adapter plate(or press the reset button), and the VSD03 module will reset and automatically connect to the network and TCP server. Then you can begin your transmission test between ComMaster and the TCP/UDP test tools.

## 4 Product Design Guide

### 4.1 Hardware Connection

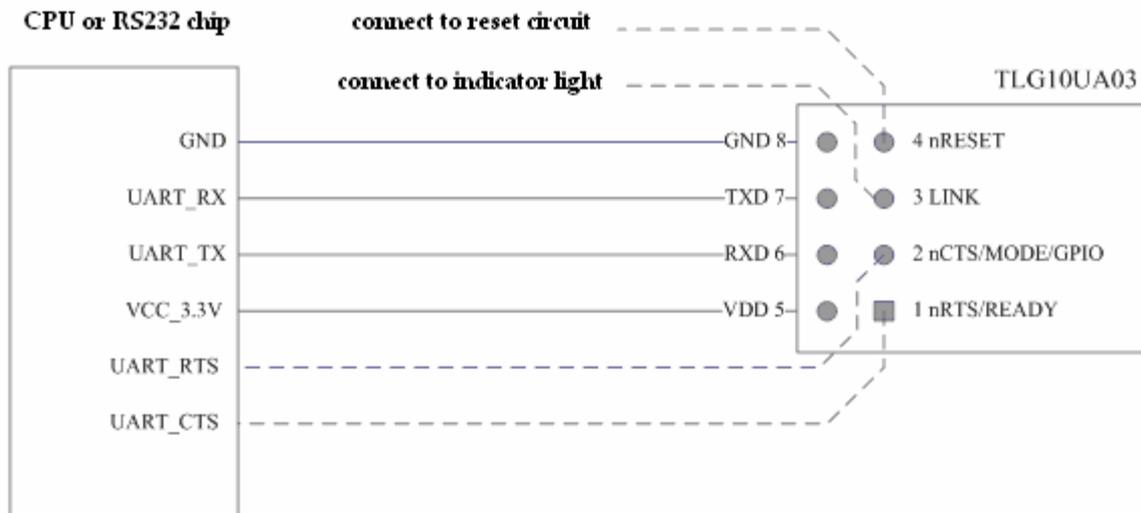


figure4-1 Hardware Connection

LG10UA03 module provides 8-pin dual in-line, the lateral row pins (5 ~ 8) must be connected, as shown above, and these pins are compatible with single-row 4-pins of TLG9UA01/02. Inside row pins (1 ~ 4) are optional feature pins, connection as shown above. If they are not used, you can leave the corresponding pins vacant. .

### 4.2 Work mode

#### 4.2.1 Start-up mode selection

Multi-functional pins nCTS/MODE/GPIO(pin 2) is used to select start-up mode at the power-on and reset phase, timing is shown below.

² Normal startup mode

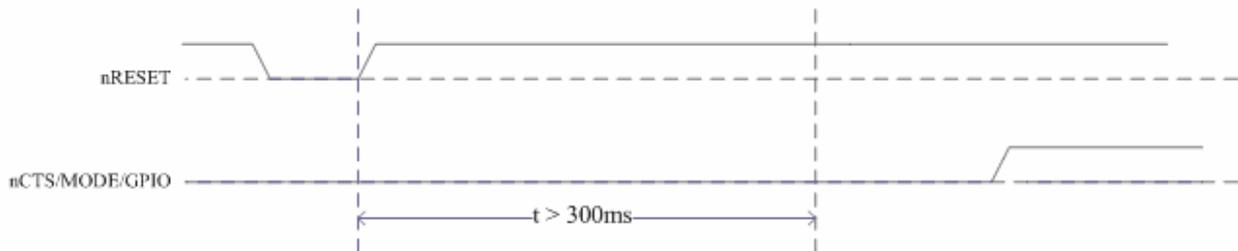


figure4-2 Normal startup mode

What is noted is that nCTS/MODE/GPIO pin of the module do a internal pull-down, when it's disconnected, module accesses to the normal startup mode automatically.

In the normal startup mode, the system runs according to the default work mode(auto or command) in the parameters.

## 2 launching configuration mode

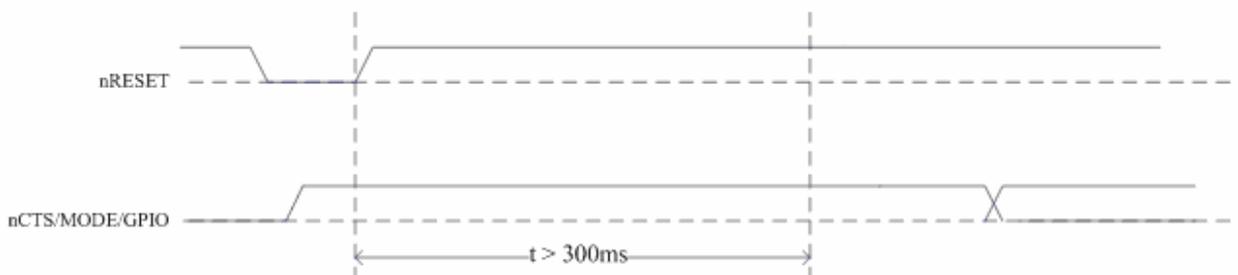


figure4-3 launching configuration mode

In launching configuration mode, the system ignores the default work mode parameters, and force to operate in the AT+Instruction mode, this mode is commonly used to modify the parameters with the configuration management program and to do the function tests.

### 4.2.2 Auto-work mode

The module supports auto-work mode, and it really achieves a plug and play serial, and it reduces user complexity furthest. In this mode, user should only configure the necessary parameters. After power is on each time, module can automatically connect to the default wireless network and server. As in the automatic mode, the module's serial always work in the transparent transmission mode, so users only need to think of it as a virtual serial cable, and send and receive data as using a simple serial. In other words, the

serial cable of users' original serial devices is directly replaced with the module, user equipments can be easy for wireless data transmission without any changes.

#### 4.2.2.1 Automatic retry mechanism

In auto-work mode, the connection of the system runs completely and automatically, and its working process is shown below:

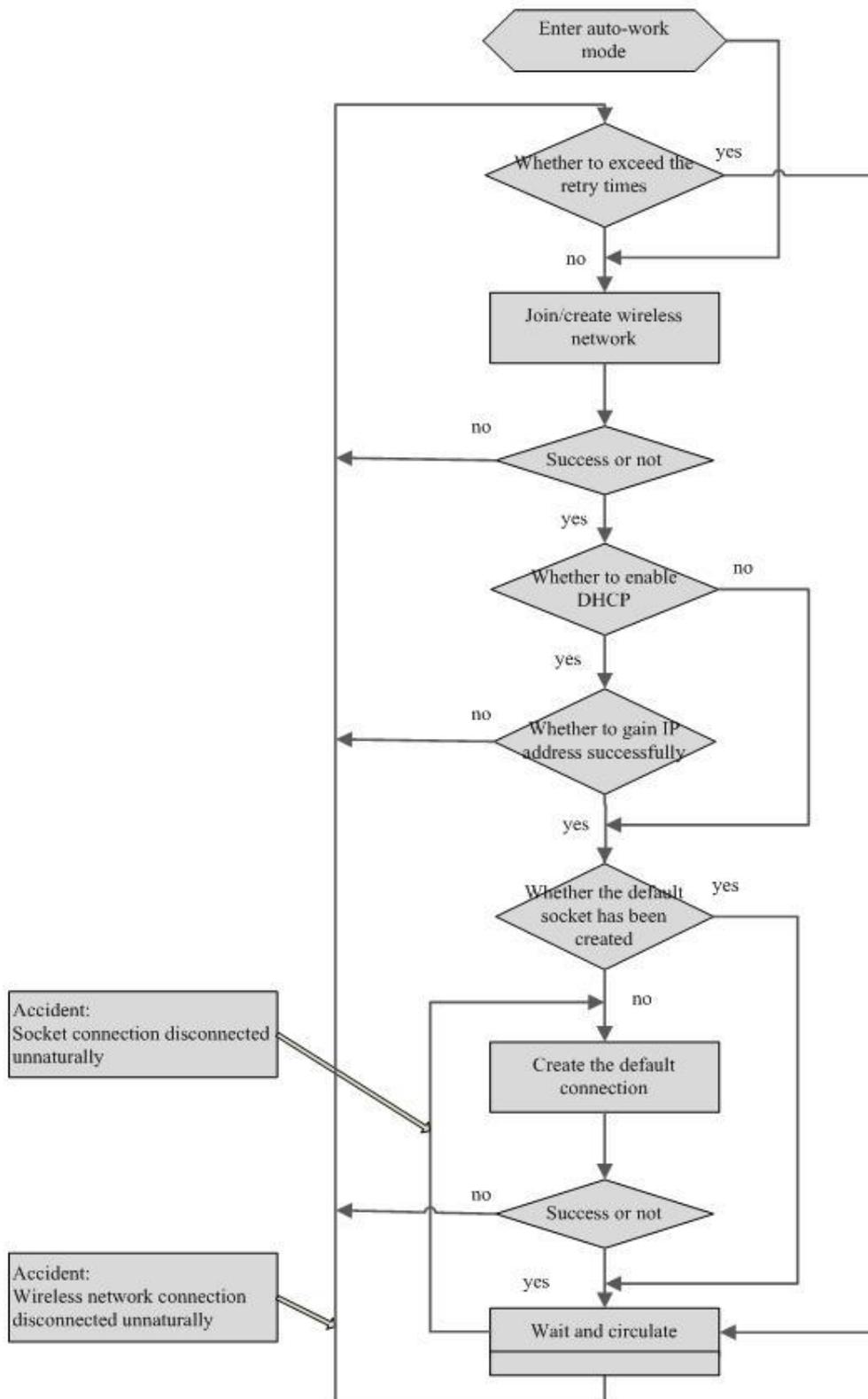


figure4-4 Automatic retry mechanism

Note:

1、 the default Automatic retry time of the system is set to forever,that is,the ssysyem will

retry to connect and acquire when the networking is failed or network is disconnected. Users can also set a limited number of retries value, in this case, once the retry number is over the limit, the system will not try to reconnect any more, except resetting the system;

2. Socket connection (here especially means Tcp connection) is a virtual connection, in the case of non-normal disconnection of the wireless network, the original Socket connection will not be deleted, and restored when the wireless network connection is resumed;

#### 4.2.2.2 Auto framing system

In transparent transmission mode, all data in the serial transfers in the form of byte stream, and in the network, the data transfers at a fixed frame format, therefore, the data between the serial and network need a conversion process between a stream and a frame.

For the data transmission in the direction from the network to the serial, data conversion from a frame into a stream is very easy. but in the contrary direction, that is data is converted from the stream to frame format, it needs certain rules to restrain. The module uses two parameters to specify the conversion rules:

- 2 Auto framing length: When the data length the module received reach the length of auto framing, the module will trigger the framing operation, and send the data frame to the network.
- 2 Auto framing period: In certain cases, such as the data sent is little, and cause the data in buffer can fill up a data block, then the data in buffer is not normally sent. At this time timeout mechanism is needed to ensure the small amount of data can be sent, the timeout is auto framing period.

Two parameters above together form the two trigger terms of auto framing. the longer the length of auto framing, the higher the transmission performance of the system, the shorter the auto framing period, the higher real-time transmission of the system, so in practice the two parameters can be adjusted according to the needs to obtain the best effect of transmission.

#### 4.2.2.3 Hardware Flow Control Mechanism

Module supports hardware flow control mechanism in transparent transmission mode.

- 2 RTS: it indicates the enable signal of module receiver, and it's effective when it's low, if

the following conditions are met simultaneously,RTS signal is effective.

- u Wireless network is connected
  - u The length of data in Uart receiver buffer is no longer than 3/4 of full
- 2 CTS: it indicates the enable signal of host receiver,and it's effective when it's low, once the module detects the ineffective CTS signal,it will immediately stop sending data to the serial until CTS becomes effective.

#### 4.2.2.4 Exit transparent transmission

In auto-work mode with default configuration, the module's serial is in transparent transmission state.At this time, all data through the serial receiving are forwarded to the network, and similarly,data received from the network, the module also sent to serial. This is a simple data transfer state,at this time, if user wants to do the transmission in command through the serial,firstly needs to exit the transparent transmission status. Meanwhile, the module also exit auto-work mode.

The conditions of Module exiting transparent transmission status is shown as following figure.

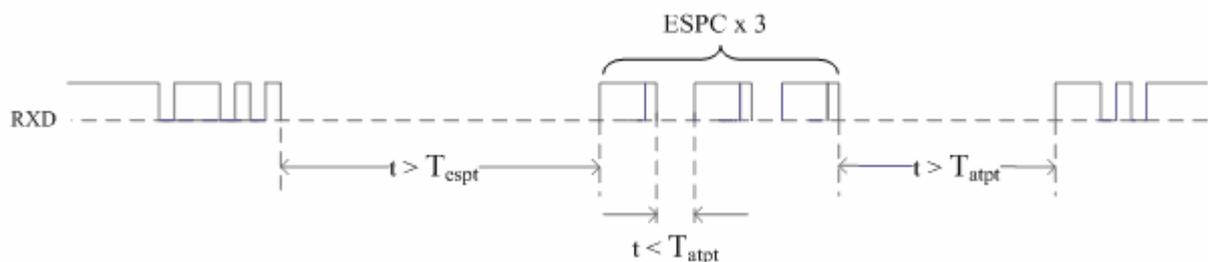


figure4-5 Sketch map of exiting transparent transmission status

Thereinto,

ESPC: escape character

$T_{espt}$ : Escape time

$T_{atpt}$ : Auto framing period

After having detected the exiting condition as shown above,the module puts out "+OK",and confirms the serial exiting transparent transmission and switching to the AT+Instruction state

#### 4.2.3 Command work mode

The module also provides a command mode based on serial control,to use for different applications of users' specific requirements.

In this mode, the module works according to user commands issued through the serial,

user can completely control the module with command, including parameters configuration, networking control, TCP/IP connection control and data transmission. This is an advanced usage, and also the most flexible for users use. Users can discretionarily control the connection and disconnection of the wireless network by order, also can create several different types of TCP/IP connection at the same time and keep them transmitting.

To use this mode, please refer to Chapter 6 AT+Instruction control protocol.

### 4.3 Parameter Details

#### 4.3.1 Wireless parameters

##### 4.3.1.1 SSID

Table 4-1 SSID

| Parameter name   | Parameter | Correlative Command |
|--|-----------|---------------------|
| SSID   | SSID      | AT+SSID             |
| Description  |           |                     |
| network identity of Wifi network, the length is 1 to 32 ASCII characters |           |                     |

##### 4.3.1.2 Encryption mode

Table 4-2 Encryption mode

| Parameter name   | Parameter  | Correlative Command |
|--|------------|---------------------|
| Encryption mode  | Encryption | AT+ENCRY            |
| Description  |            |                     |
| encryption types of Wifi network, options include: <ul style="list-style-type: none"> <li>2 OPEN</li> <li>2 WEP64</li> <li>2 WEP128</li> <li>2 WPA-PSK(TKIP)</li> <li>2 WPA-PSK(CCMP)</li> <li>2 WPA2-PSK(TKIP)</li> <li>2 WPA2-PSK(CCMP)</li> <li>2 WAPI (not supported)</li> </ul> |            |                     |

##### 4.3.1.3 Key Format

Table 4-3 Key Format

| Parameter name | Parameter | Correlative Command |
|----------------|-----------|---------------------|
|----------------|-----------|---------------------|

| Key Format   | Key Format | AT+KEY |
|--|------------|--------|
| Description  |            |        |
| <p>Key format of WiFi network, options include:</p> <ul style="list-style-type: none"> <li>≥ HEX format</li> <li>≥ ASCII format</li> </ul> <p>For WEP encryption, the difference between HEX and ASCII formats is the display format of key, such as WEP64 encryption, its key can be 5 ASCII characters (such as 12345) or 10 hexadecimal numbers (such as 3132333435).</p> <p>For WPA/WPA2-PSK encryption, HEX and ASCII formats indicate different meanings,</p> <ul style="list-style-type: none"> <li>HEX format: PMK</li> <li>ASCII format: PSK</li> </ul> |            |        |

#### 4.3.1.4 Key Index

Table 4-4 Key Index

| Parameter name   | Parameter | Correlative Command |
|--|-----------|---------------------|
| Key Index  |           | AT+KEY              |
| Description  |           |                     |
| <p>key index number of WiFi network, this parameter is only effective in WEP encryption mode, optional range is 1 to 4, it's no sense in other encryption mode</p> |           |                     |

#### 4.3.1.5 Key

Table 4-5 Key

| Parameter name  | Parameter      | Correlative Command |
|---|----------------|---------------------|
| Key   | Encryption Key | AT+KEY              |
| Description   |                |                     |
| <p>Key string of Wifi network, the keys of different encryption modes and key formats have different length provisions:</p> <ul style="list-style-type: none"> <li>≥ WEP64 (HEX): 10 characters</li> <li>≥ WEP64 (ASCII): 5 characters</li> <li>≥ WEP128 (HEX): 26 characters</li> <li>≥ WEP128 (ASCII): 13 characters</li> <li>≥ WPA/WPA2-PSK (HEX): 64 characters</li> <li>≥ WPA/WPA2-PSK (ASCII): 8~63 characters</li> </ul> |                |                     |

### Tips

In WPA/WPA2-PSK (TKIP/CCMP) encryption mode, as the key format is ASCII, the module must first do the conversion to the key of ASCII format when it's power-on, this process increases about 10 seconds to the starting time. And When using WPA/WPA2 encryption, directly using PMK of hexadecimal format can eliminate this delay. However, some APs does not provide PMK (which can be directly input) of hexadecimal format in WPA/WPA2 mode, in this case, you can use the WPA key conversion tool (



and set to the module.

## 4.3.2 Network Parameters

### 4.3.2.1 DHCP

Table 4-6 DHCP

| Parameter name  | Parameter   | Correlative Command |
|---|-------------|---------------------|
| DHCP  | DHCP Enable | AT+NIP              |
| Description   |             |                     |
| Used to indicate the network address type of the client, if this parameter is enabled, the module will dynamically obtain the network address of the client according to the DHCP protocol. |             |                     |

### 4.3.2.2 IP address

Table 4-7 IP address

| Parameter name   | Parameter        | Correlative Command |
|--|------------------|---------------------|
| IP address   | Fixed IP Address | AT+NIP              |
| Description  |                  |                     |
| Used to specify static IP address of the local network address of the client, it's only effective in the case of exiting DHCP. |                  |                     |

### 4.3.2.3 Subnet Mask

Table 4-8 Subnet Mask

| Parameter name | Parameter   | Correlative Command |
|----------------|-------------|---------------------|
| Subnet Mask    | Subnet Mask | AT+NIP              |
| Description    |             |                     |

Used to specify the subnet mask of the local network address, it's only effective in the case of exiting DHCP.

#### 4.3.2.4 Gateway address

Table4-9 Gateway address

| Parameter name   | Parameter       | Correlative Command |
|--|-----------------|---------------------|
| Gateway address  | Gateway Address | AT+NIP              |
| Description  |                 |                     |
| Used to specify the gateway address of the local network address, it's only effective in the case of exiting DHCP. |                 |                     |

#### 4.3.2.5 DNS Server

Table 4-10 DNS Server

| Parameter name   | Parameter   | Correlative Command |
|--|-------------|---------------------|
| DNS Server   | DNS Address | AT+NIP              |
| Description  |             |                     |
| Used to specify the DNS server address of the local network address, t's only effective in the case of exiting DHCP. |             |                     |

### 4.3.3 Parameters of work mode

#### 4.3.3.1 Auto-work mode

Table 4-11 Auto-work mode

| Parameter name  | Parameter        | Correlative Command |
|---|------------------|---------------------|
| Auto-work mode  | Auto Mode Enable | AT+ATM              |
| Description   |                  |                     |
| Used to specify whether auto-work mode of the module is enabled or not. |                  |                     |

#### 4.3.3.2 Protocol Type

Table 4-12 Protocol Type

| Parameter name   | Parameter | Correlative Command |
|--|-----------|---------------------|
| Protocol Type  | Protocol  | AT+ATRM             |
| Description  |           |                     |
| Used to specify the protocol type of the default connection module creates in the auto-work mode, options include: |           |                     |

|  |
|--|
| <ul style="list-style-type: none"> <li>2 TCP</li> <li>2 UDP</li> </ul> |
|--|

#### 4.3.3.3 CS mode

Table 4-13 CS mode

| Parameter name   | Parameter | Correlative Command |
|--|-----------|---------------------|
| CS mode  | CS Mode   | AT+ATRM             |
| <b>Description</b>   |           |                     |
| <p>Used to specify the CS mode of the default connection module creates in the auto-work mode, options include:</p> <ul style="list-style-type: none"> <li>2 Client</li> <li>2 Server</li> </ul> |           |                     |

#### 4.3.3.4 Server Address

Table 4-14 Server Address

| Parameter name  | Parameter      | Correlative Command |
|---|----------------|---------------------|
| Server Address  | Server Address | AT+ATRM             |
| <b>Description</b>  |                |                     |
| <p>In the auto-work mode, when CS mode of the default connection module creates is Client, it's used to specify the server module connects to, its input format may be IP address or domain name.</p> |                |                     |

#### 4.3.3.5 TCP connection timeout

Table 4-15 TCP connection timeout

| Parameter name  | Parameter        | Correlative Command |
|---|------------------|---------------------|
| TCP connection timeout  | Tcp Link TimeOut | AT+ATRM             |
| <b>Description</b>  |                  |                     |
| <p>In the auto-work mode, when CS mode of the default connection module creates is TCP server, it's used to specify the maximum connection time of the client which joined in the server in the condition of no data transfer, if the no transfer connection is more than this time, the client will be kicked out automatically.</p> |                  |                     |

## 4.3.3.6 Port number

Table 4-16 Port number

| Parameter name  | Parameter   | Correlative Command |
|---|-------------|---------------------|
| Port Number   | Port Number | AT+ATRM             |
| Description   |             |                     |
| Used to specify the port number of the default connection module creates in the auto-work mode. |             |                     |

## 4.3.4 Advanced wireless parameters

## 4.3.4.1 Network mode

Table 4-17 Network mode

| Parameter name   | Parameter    | Correlative Command |
|--|--------------|---------------------|
| Network mode   | Network Mode | AT+WPRT             |
| Description  |              |                     |
| <p>Used to specify the network mode of the Wifi network, options include:</p> <ul style="list-style-type: none"> <li>≥ Infrastructure</li> <li>≥ Ad hoc</li> </ul> <p>The default setting is Infrastructure. In most cases,if what we want to join to is wireless network AP creates,using the default setting is just contented. Only when user needs to join Adhoc,the parameter may be modified to Adhoc.</p> |              |                     |

## 4.3.4.2 Creating adhoc when network does not exist

Table 4-18 Create adhoc when network does not exist

| Parameter name  | Parameter                                | Correlative Command |
|---|--|---------------------|
| Create adhoc When network does not exist  | Create adhoc When network does not exist | AT+WATC             |
| Description   |  |                     |
| <p>Used to specify whether to automatically create network or not if the network does not exist ,When Wifi network type is adhoc.</p> <p>The default setting of system is disabled. This parameter is only effective if the network type is set to Adhoc, if users need to create an Adhoc network, set this parameter to Enable.</p> |  |                     |

## 4.3.4.3 BG mode

Table 4-19 BG mode

| Parameter name   | Parameter         | Correlative Command |
|--|-------------------|---------------------|
| BG mode  | Wireless Standard | AT+WBGR             |
| Description  |                   |                     |
| <p>Used to specify the BG mode of the Wifi network, options include:</p> <ul style="list-style-type: none"> <li>≥ B/G fixed mode</li> <li>≥ B mode</li> </ul> <p>The default setting of system is B/G fixed mode, ordinarily user doesn't need to modify this parameter.</p> |                   |                     |

## 4.3.4.4 The maximum rate

Table 4-20 The maximum rate

| Parameter name  | Parameter | Correlative Command |
|---|-----------|---------------------|
| The maximum rate  | Data Rate | AT+WBGR             |
| Description   |           |                     |
| <p>The maximum wireless sending rate, options include:<br/>1M、2M、5.5M、11M、6M、9M、12M、18M、24M、36M、48M、54M<br/>When BG mode is chosed to be B mode, options include only the first 4 items.</p> <p>The default setting of system is 36M, ordinarily user doesn't need to modify this parameter. when the use environment is harsh,and wireless signal is worse,user can modify this parameter to reduce the sending rate in order to improve the reliability of wireless transmission.</p> |           |                     |

## 4.3.4.5 BSSID

Table 4-21 BSSID

| Parameter name   | Parameter | Correlative Command |
|--|-----------|---------------------|
| BSSID  | AP BSSID  | AT+BSSID            |
| Description  |           |                     |
| <p>Used to specify the BSSID address of WiFi network, options include:</p> <ul style="list-style-type: none"> <li>≥ Auto</li> <li>≥ Fixed, it needs to input BSSID address</li> </ul> <p>The default setting of system is Auto, ordinarily user doesn't need to modify</p> |           |                     |

this parameter. When join to WiFi network, module can detect BSSID address of the network according with SSID and join in. When several networks (APs create) with the same SSID is existed, in order to enhance network security, user can fix the target AP by modifying this parameter to fix BSSID address. Once the parameter is fixed, the module will never connect to the network without suited BSSID.

#### 4.3.4.6 Channel

Table 4-22 Channel

| Parameter name   | Parameter      | Correlative Command |
|--|----------------|---------------------|
| Channel  | Channel Number | AT+CHL              |
| Description  |                |                     |
| <p>Used to specify the channel of WiFi network, options include:</p> <ul style="list-style-type: none"> <li>≥ Auto</li> <li>≥ Channel 1~14</li> </ul> <p>The default setting of system is Auto, ordinarily user doesn't need to modify this parameter. When join to WiFi network, the module will automatically scan all channels to detect the target network. The time module needs to scan each channel is 150ms, and the completion of all channel scan takes about 2s. In order to reduce the scanning time and improve networking speed, user can specify channel number through modifying this parameter. What should be noted, Once user set it to fixed channels, the module scans only the specified channels, if the target AP has changed the channel number, module will never connect to the network successfully.</p> |                |                     |

#### 4.3.4.7 Auto Retry

Table 4-23 Auto Retry

| Parameter name   | Parameter        | Correlative Command |
|--|------------------|---------------------|
| Auto Retry   | Auto Retry Count | AT+CHL              |
| Description  |                  |                     |
| <p>This parameter is only effective in auto-work mode, it's used to specify the auto retry number when networking is failed or network disconnects unnaturally, options include:</p> <ul style="list-style-type: none"> <li>≥ Never</li> </ul> |                  |                     |

|   |
|---|
| <p>2 Ever</p> <p>2 1~254</p> <p>The default setting of system is Ever</p> |
|---|

#### 4.3.4.8 Wireless roam

Table 4-24 wireless roam

| Parameter name   | Parameter     | Correlative Command |
|--|---------------|---------------------|
| Wireless roam  | Wireless Roam | AT+WARM             |
| Description  |               |                     |
| <p>Used to specify whether to enable the function of wireless roam or not.</p> <p>The default setting of system is disabled. What should be noted, when the parameter of BSSID is set to fixed, wireless roam will become no effect.</p> |               |                     |

#### 4.3.4.9 Channel List

Table 4-25 Channel List

| Parameter name  | Parameter    | Correlative Command |
|---|--------------|---------------------|
| Channel List  | Channel List | AT+CHLL             |
| Description   |              |                     |
| <p>Used to specify the wireless region settings, the ranges of the 2.4G frequency bands different countries and regions open is different, such as the United States is 1 ~ 11 channels, Europe is 1 to 13 channels, and Japan is 1 to 14 channels.</p> <p>The system default setting is 1~14 channels. ordinarily user doesn't need to modify this parameter. Another use of this parameter is reducing the time of scanning and networking by modifying the channel range. For example, we have known the AP may be connected to is in 1,6,11 channels, the channel list can be set only 1,6,11, so that the module will only scan these three channels in the process of networking, thus the purpose of shortening the networking time is reached.</p> <p>Another should be noted is that parameter of channel has higher priority than the parameter of channel list, that is, when the parameter of channel is set to the specified channel, even if this channel is not included by the channel list, networking will still succeed.</p> |              |                     |

### 4.3.5 Port parameters

#### 4.3.5.1 Baud Rate

Table 4-26 Baud Rate

| Parameter name  | Parameter | Correlative Command |
|---|-----------|---------------------|
| Baud Rate   | Baud Rate | AT+UART             |
| Description   |           |                     |
| Baud rate setting of serial,options include:<br>≥ 1200~115200bps<br>The default setting of system is 115200bps。 |           |                     |

#### 4.3.5.2 Data Bits

Table 4-27 Data Bits

| Parameter name   | Parameter | Correlative Command |
|--|-----------|---------------------|
| Data Bits  | Data Bits | AT+UART             |
| Description  |           |                     |
| Length setting of serial data bits,options include:<br>≥ 8 bits<br>≥ 7 bits<br>The default setting of system is 8 bits |           |                     |

#### 4.3.5.3 Stop Bits

Table 4-28 Stop Bits

| Parameter name  | Parameter | Correlative Command |
|---|-----------|---------------------|
| Stop Bits   | Stop Bits | AT+UART             |
| Description   |           |                     |
| Length setting of serial stop bits,options include:<br>≥ 1 bit<br>≥ 2 bits<br>The default setting of system is 1 bit. |           |                     |

#### 4.3.5.4 Parity Bit

Table 4-29 Parity Bit

| Parameter name | Parameter | Correlative Command |
|----------------|-----------|---------------------|
| Parity Bit     | Parity    | AT+UART             |

| Description  |
|--|
| Parity Bit setting of serial,options include:<br>2 no parity<br>2 odd parity<br>2 even parity<br>The default setting of system is no parity. |

### 4.3.6 Parameters of transparent mode

#### 4.3.6.1 Auto framing length

Table 4-30 Auto framing length

| Parameter name  | Parameter           | Correlative Command |
|---|---------------------|---------------------|
| Auto framing length   | Data Trigger Length | AT+ATLT             |
| Description   |                     |                     |
| When serial is in transparent transmission mode, and the data received reaches the length this parameter specifys,module will force to frame a network frame and trigger the sending process, the optional range is 64 ~ 1024 bytes.<br>The default setting of system is 512 bytes. |                     |                     |

#### 4.3.6.2 Auto framing period

Table 4-31 Auto framing period

| Parameter name  | Parameter            | Correlative Command |
|---|----------------------|---------------------|
| Auto framing period   | Data Trigger Timeout | AT+ATPT             |
| Description   |                      |                     |
| When serial is in transparent transmission mode, the data received is less than the length of framing, if waiting time is more than this parameter specifys , module will force to frame a network frame and trigger the sending process. The timeout period of auto framing has a optional range of 100 ~ 10000ms, the minimum accuracy is 100ms. In addition, this setting must be less than the setting of escape time.<br>The default setting of system is 500ms. |                      |                     |

#### 4.3.6.3 Escape Character

Table 4-32 Escape Character

| Parameter name | Parameter | Correlative Command |
|----------------|-----------|---------------------|
|----------------|-----------|---------------------|

| Escape Character   | Escape Character | AT+ESPC |
|--|------------------|---------|
| <b>Description</b>   |                  |         |
| <p>This parameter is specified the escape character used to exit the transparent transmission mode for serial, such as "+", when system is in the state of detecting escape character, and the data receives is three and only three consecutive escape characters, it will exit the transparent transmission mode.</p> <p>The default setting of system is 0x2B, that is characters "+" in ASCII.</p> |                  |         |

#### 4.3.6.4 Escape time

Table 4-33 Escape time

| Parameter name   | Parameter             | Correlative Command |
|--|-----------------------|---------------------|
| Escape time  | Escape Trigger Period | AT+ESPT             |
| <b>Description</b>   |                       |                     |
| <p>When serial is in transparent transmission mode, the time received no data is more than this parameter specifies, it will automatically detect the escape characters. Optional range is 100 ~ 10000ms, the minimum accuracy is 100ms. In addition, the setting must be greater than the setting of auto framing period.</p> <p>The default setting of system is 2000ms.</p> |                       |                     |

### 4.3.7 System parameters

#### 4.3.7.1 System password

Table 4-34 System password

| Parameter name  | Parameter      | Correlative Command |
|---|----------------|---------------------|
| System password   | Login Password | AT+PASS             |
| <b>Description</b>  |                |                     |
| <p>The login password for accessing the module through WEB server or wireless configuration.</p> <p>The default setting of system is "000000" .</p> |                |                     |

#### 4.3.7.2 WEB server

Table 4-35 WEB server

| Parameter name | Parameter  | Correlative Command |
|----------------|------------|---------------------|
| WEB Server     | Web Server | AT+WEBS             |

| Description  |
|--|
| <p>Used to specify whether to enable the WEB server or not and the server port.</p> <p>The default setting of system is enabled, the default port is 80.</p> |

#### 4.3.7.3 Command Mode

Table 4-36 Command Mode

| Parameter name  | Parameter    | Correlative Command |
|---|--------------|---------------------|
| Command Mode  | Command Mode | AT+CMDM             |
| Description   |              |                     |
| <p>Used to specify the control protocol of the module in command mode, options include:</p> <ul style="list-style-type: none"> <li>≥ AT+Instruction</li> <li>≥ Compatible Protocol</li> </ul> <p>The default setting of system is AT+Instruction mode, ordinarily user doesn't need to modify this parameter. The command mode of compatible protocol is used to be compatible with the control protocol of TLG09UA01/02 products. If user wants to use this mode, please call for the Technical Support for more help.</p> |              |                     |

#### 4.3.7.4 GPIO1 mode

Table 4-37 GPIO1 mode

| Parameter name  | Parameter | Correlative Command |
|---|-----------|---------------------|
| GPIO1 mode  | GPIO Mode | AT+IOM              |
| Description   |           |                     |
| <p>Used to specify the work mode of GPIO1, options include:</p> <ul style="list-style-type: none"> <li>≥ System function</li> <li>≥ Input</li> <li>≥ Output</li> </ul> <p>The default setting of system is system function.</p> |           |                     |

## 4.4 Default setting

Table 4-38 default setting

| parameter                                | value     | description  |
|--|-----------|--|
| <b>Wireless setting</b>                  |           |  |
| SSID                                     | "default" | The ssid of wireless network is "default"  |
| Encryption mode                          | OPEN      | The encryption mode of wireless network is open                                      |
| Key format                               | -         |  |
| Key Index                                | -         |  |
| Key                                      | -         |  |
| <b>Network setting</b>                   |           |  |
| DHCP                                     | Enable    | Enable DHCP protocol to dynamically obtain IP address                                |
| IP address                               | -         |  |
| Subnet Mask                              | -         |  |
| Gateway address                          | -         |  |
| DNS server                               | -         |  |
| <b>Work mode setting</b>                 |           |  |
| Auto-work mode                           | Enable    | Enable the auto-work mode  |
| Protocol type                            | TCP       | Default connection:<br>When it's set to TCP server, the default timeout is 120second |
| CS mode                                  | Server    |  |
| Server address                           | -         |  |
| TCP connection timeout                   | 120       |  |
| <b>Advanced wireless setting</b>         |           |  |
| Network mode                             | Infra     | Network mode is Infrastructure   |
| Create adhoc when network does not exist | -         | -  |
| BG mode                                  | B/G       | Wireless mode is BG fixed  |
| Maximum rate                             | 36M       | The Maximum sending rate is 36M  |
| BSSID                                    | Auto      | Forbid to fix the BSSID address of wireless network                                  |

|  |          |   |
|--|----------|---|
| Channel                                | Auto     | Detect the operating channel of wireless network automatically  |
| Auto retry                             | 255      | Retry for ever automatically  |
| Wirelss roam                           | Disable  | Forbid to do the wireless roam  |
| Channel list                           | 0x3fff   | 1~14 channels   |
| <b>Serial setting</b>                  |          |   |
| Baud rate                              | 115200   | The data format of serial is set to:<br>Baud rate is 115200bps;<br>8 bits of data bits<br>1 bit of stop bits<br>No parity of parity bit |
| Data bits                              | 8 bits   |   |
| Stop bits                              | 1 bit    |   |
| Parity bit                             | None     |   |
| <b>The setting of transparent mode</b> |          |   |
| Auto framing length                    | 512      | The length is 512 bytes   |
| Auto framing period                    | 500ms    | The period is 500ms   |
| Escape character                       | 0x2B     | The character is 0x2B,that is "+" in ASCII  |
| Escape time                            | 2000ms   | The period is 2000ms  |
| <b>System setting</b>                  |          |   |
| System password                        | "000000" | Login password of system is "000000"  |
| WEB server                             | Enable   | Enable WEB server, port is 80   |
| Command mode                           | AT+      | The system command is AT+Instruction set  |
| GPIO1 mode                             | System   | The setting of GPIO1is system function  |

## 4.5 Typical application examples

### 4.5.1 The networking of active serial device

#### 4.5.1.1 Application scene

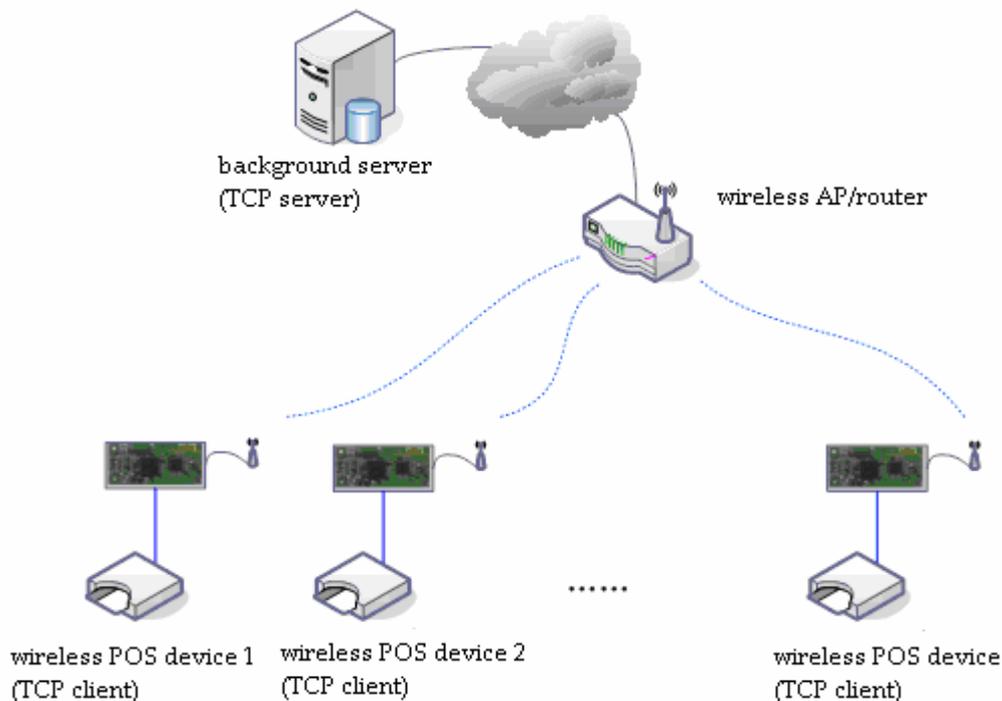


Figure 4-6 The networking of active serial device

The networking of active serial device means the device sponsors the networking and data exchange with background server (upload or download). A typical active-type equipment is such as wireless POS machine, after each credit card transaction and then begins to connect background server, and upload the transaction data. The topology of the networking of active serial device is as shown above. Thereinto, the background server works as a TCP Server, the device connects to the network through wireless AP or router, and works as a TCP Client.

#### 4.5.1.2 Reference Setting

##### Wireless POS machine X

- 2 Wireless setting
  - Ø SSID, encryption mode, key and so on (refer to the setting of wireless AP or router)
- 2 Network setting

- Ø Enable DHCP
- 2 Work mode setting
  - Ø Enable auto-work mode
  - Ø Protocol type: TCP
  - Ø C/S mode: client
  - Ø Server address: IP address of background server,such as 192.168.0.100
  - Ø Port number: port number of background server,such as 6000

## 4.5.2 The networking of passive serial device

### 4.5.2.1 Application scene

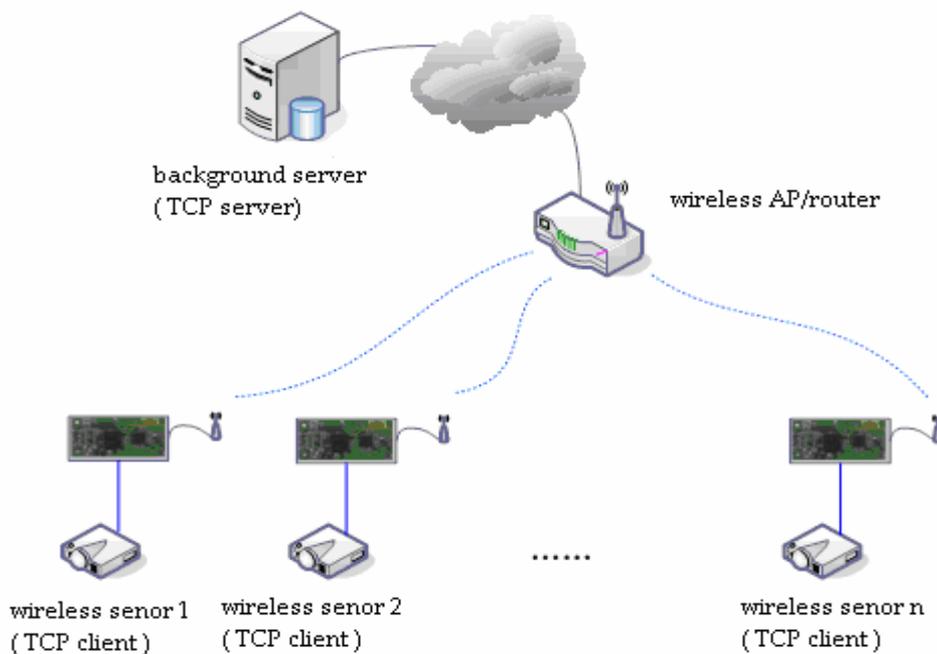


Figure4-7 The networking of passive serial device

The networking of passive serial device means, all devices in the system are passive to wait for a connecting, only the background server sponsors the networking and does request or data download. Typical applications are such as some wireless sensor network, each sensor terminals always are doing the real-time data collection, and the data collected does not immediately upload, but temporarily stores in the devices. The background servers periodically connect devices actively, and request to upload or download data. At this time, background server works as a TCP Client, while the device works as a TCP Server.

#### 4.5.2.2 Reference setting

##### Wireless sensor X

- 2 Wireless setting
  - Ø SSID,encryption mode,key and so on(refer to the setting of wireless AP or router)
- 2 Network setting
  - Ø IP address: 192.168.0.X
  - Ø Subnet mask: 255.255.255.0
  - Ø Gateway address: 192.168.0.1
  - Ø DNS address: 192.168.0.1
- 2 Work mode setting
  - Ø Enable auto-work mode
  - Ø Protocol type: TCP
  - Ø C/S mode: server
  - Ø Port number: 6000
  - Ø Connection timeout: 120s

#### 4.5.3 The networking with broadcast method of serial device

##### 4.5.3.1 Application scene

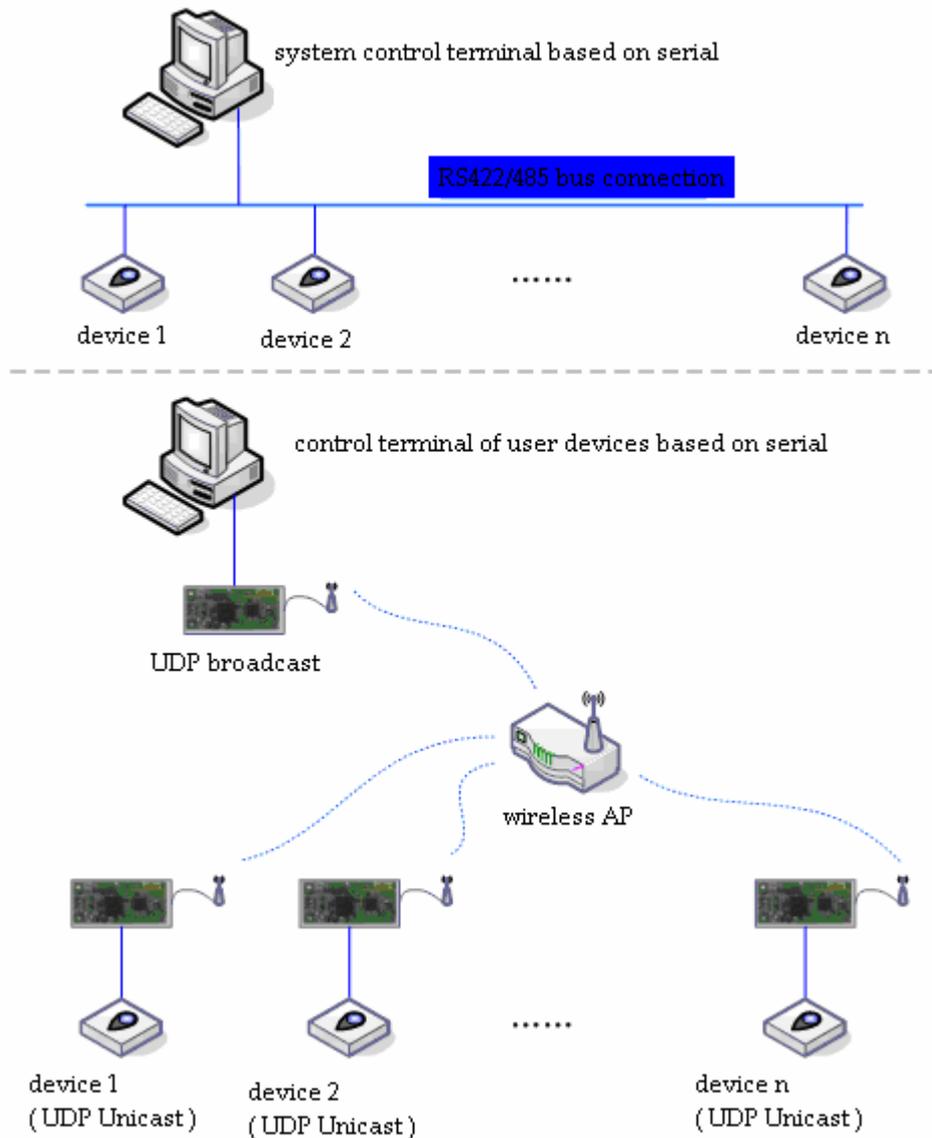


figure4-8 The networking with broadcast method of serial device

The networking with broadcast type of serial device is a networking mode based on UDP protocol. In the network, the host works in UDP broadcast mode, all devices work in UDP unicast mode. This method creates a passive network essentially. The most typical application is upgrading the wireless network for the traditional network based on the connection of RS422/485 bus. If user would use this connection mode, all the devices and their system control terminals in the original system needn't to get any changes of software, user directly replaces the original hardware connection of RS422/485 with this module, and it can be truly "plug and play".

#### 4.5.3.2 Reference setting

##### Control terminal of device

- 2 Wireless setting
  - Ø SSID, encryption mode, key and so on (refer to the setting of wireless AP or router)
- 2 Network setting
  - Ø IP address: 192.168.0.Y
  - Ø Subnet mask: 255.255.255.0
  - Ø Gateway address: 192.168.0.1
  - Ø DNS address: 192.168.0.1
- 2 Work mode setting
  - Ø Enable auto-work mode
  - Ø Protocol type: UDP
  - Ø C/S mode: broadcast
  - Ø Port number: 6000
  - Ø Connection timeout: 0

##### Device X

- 2 Wireless setting
  - Ø SSID, encryption mode, key and so on (refer to the setting of wireless AP or router)
- 2 Network setting
  - Ø IP address: 192.168.0.X
  - Ø Subnet Mask: 255.255.255.0
  - Ø Gateway address: 192.168.0.1
  - Ø DNS address: 192.168.0.1
- 2 Work mode setting
  - Ø Enable auto-work mode
  - Ø Protocol type: UDP
  - Ø C/S mode: unicast
  - Ø Port number: 6000
  - Ø Connection timeout: 0

#### 4.5.4 Wireless virtual serial

##### 4.5.4.1 Application scene

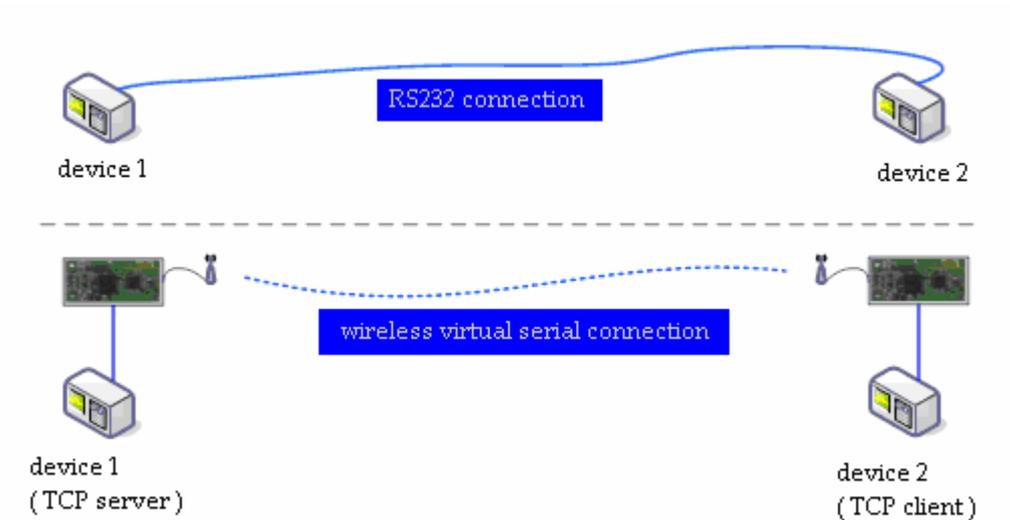


figure4-9 Wireless virtual serial

#### 4.5.4.2 Reference setting

##### Device 1

- 2 Wireless setting
  - Ø Network mode: adhoc, and enable "Create adhoc when network does not exist"
  - Ø SSID: my\_adhoc
  - Ø Encryption mode: open
  - Ø Others is default
- 2 Network setting
  - Ø IP address: 192.168.0.1
  - Ø Subnet Mask: 255.255.255.0
  - Ø Gateway address: 192.168.0.1
  - Ø DNS address: 192.168.0.1
- 2 Work mode setting
  - Ø Enable auto-work mode
  - Ø Protocol type: TCP
  - Ø C/S mode: server
  - Ø Port number: 6000
  - Ø Connection timeout: 0

##### Device 2

- 2 Wireless setting

- Ø Network mode: adhoc
- Ø SSID: my\_adhoc
- Ø Encryption mode: open
- Ø Others is default

2 Network setting

- Ø IP address: 192.168.0.2
- Ø Subnet Mask: 255.255.255.0
- Ø Gateway address: 192.168.0.1
- Ø DNS address:192.168.0.1

2 Setting of work mode

- Ø Enable auto-work mode
- Ø Protocol type: TCP
- Ø C/S mode: client
- Ø Port number: 6000
- Ø Server address:192.168.0.1

#### **4.5.5 Multi-network connection mode**

##### 4.5.5.1 Application scene

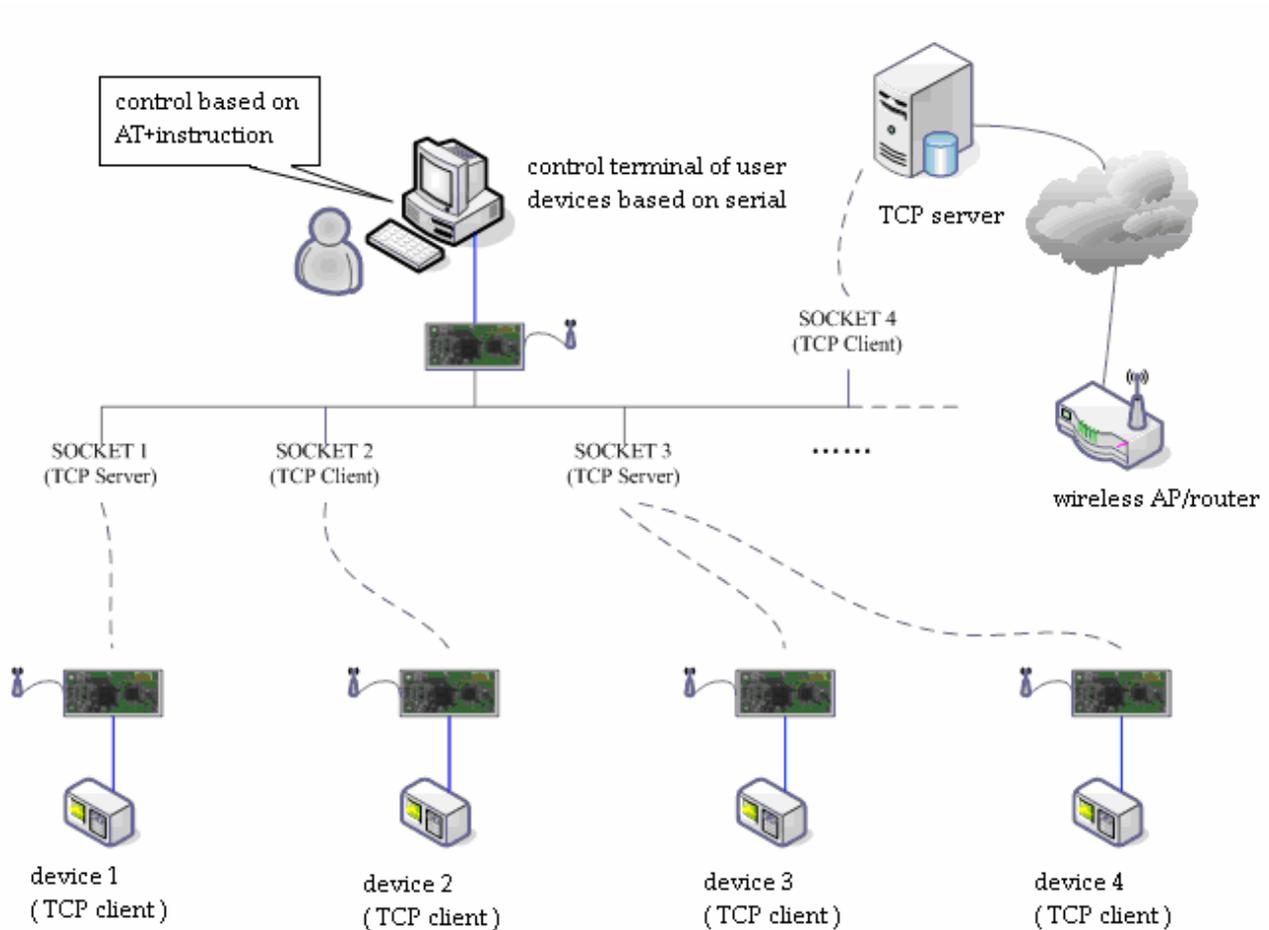


figure4-10 Multi-network connection mode

As shown above, it's the network topology diagram of multi-network connection based on this modules, this system does not be used for a particular application, it is not practical, but only be used to describe the function of this module as an analog system.

In the system above the module of device control terminal works in command mode, user inputs to the AT+Instruction to control it create four TCP connections, including 2 Clients and 2 Servers. There are a number of clients access to each Server. Control terminal can communicate with all devices and server in the above figure through different socket connection.

#### 4.5.5.2 Reference setting

##### Device control terminal

##### 2 Wireless setting

- Ø SSID, encryption mode, key and so on (refer to the setting of wireless AP or router)

2 Network setting

- Ø IP address: 192.168.0.X
- Ø Subnet Mask: 255.255.255.0
- Ø Gateway address: 192.168.0.1
- Ø DNS address:192.168.0.1

2 Work mode setting

- Ø Disable auto-work mode

## 5 Configuration management manual

### 5.1 Configuration through configuration management software

#### 5.1.1 Introduction

We offer a configuration hypervisor that can manage VSD03 module expediently, for configuration querying、config、function test etc, its main function includes,

- ² module management
  - n managemant through serial
  - n managemant through wireless connection
- ² parameter configuration
  - n modify/query module parameters
  - n import/export parameters
  - n modify parameters for a group of modules
  - n resume to default setting
- ² function test
  - n wireless network command test
  - n socket command test
  - n AT+ instruction test
- ² firmware upgrade
- ² system information query

#### 5.1.2 Module management

##### 5.1.2.1 Management through serial connection

- 1、 Connect the VSD03 module to the serial adpater, then turn it power-on;
- 2、 Open the UART-WIFI configuration hypervisor, select the port number of connection from port list, you can select 'auto' unless you make sure;
- 3、 Click 'search module', if the connection port is selected to 'auto' and the computer has more than one serial, it maybe expend a long time for searching, The following sketch map will show after search completed:

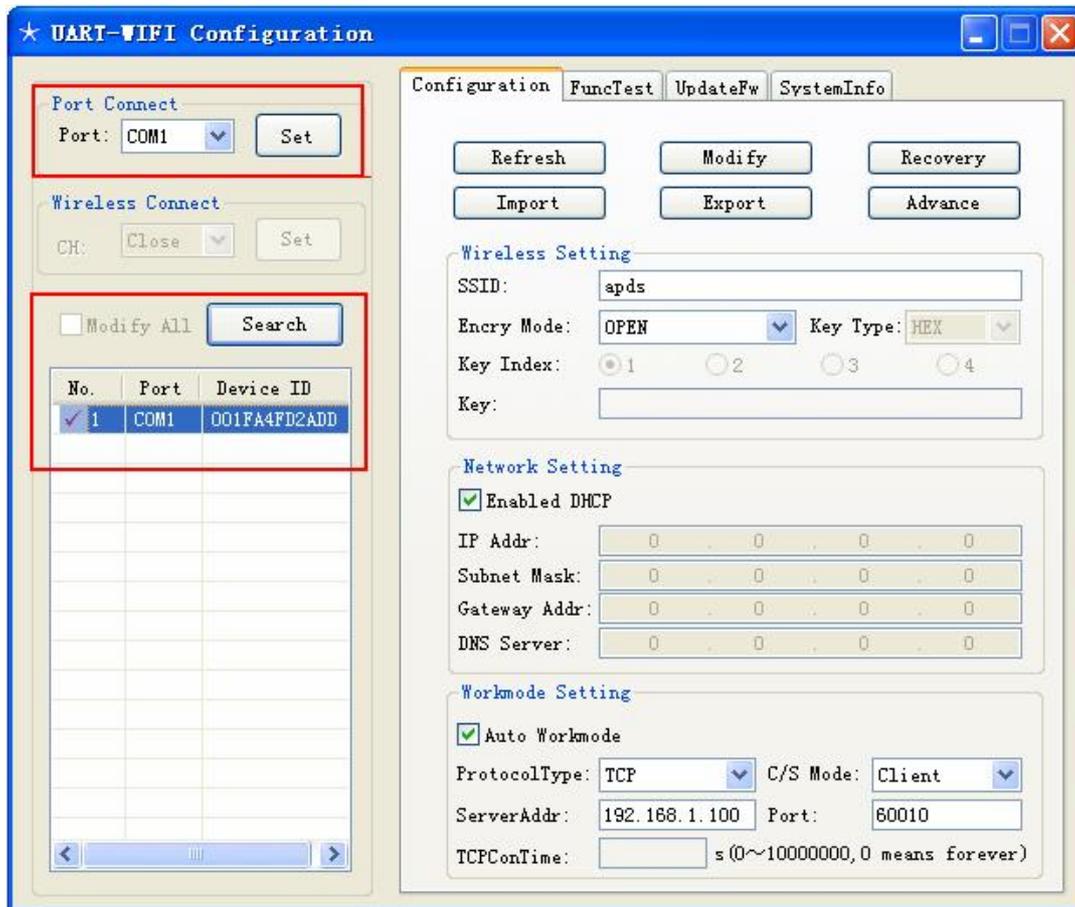


figure5-1 serial connection sketch map

### 5.1.2.2 Management through wireless connection

Note: Using this function needs to choose correct wireless adapter and install wireless driver correctly.

- 1、 Plug the wireless adapter in computer, it's shown as the figure 2-5;
- 2、 Install wireless driver, the successful result is shown as the following sketch map;



Figure5-2 Install wireless driver

- 3、 Open the UART-WIFI configuration hypervisor, if the driver is installed successfully, setting frame of wireless connection will be in activation status;
- 4、 Click 'setting' button, input system password (only the module of which the system password matchs with the setting can be scanned);

- 5、 Select scan channel from channel list. It will scan all channels from 1 to 14 ,if 'auto' is selected;
- 6、 Click 'search module', if channel is selected to 'auto', it will expend a long time to search, the completed result is shown as the following sketch map:

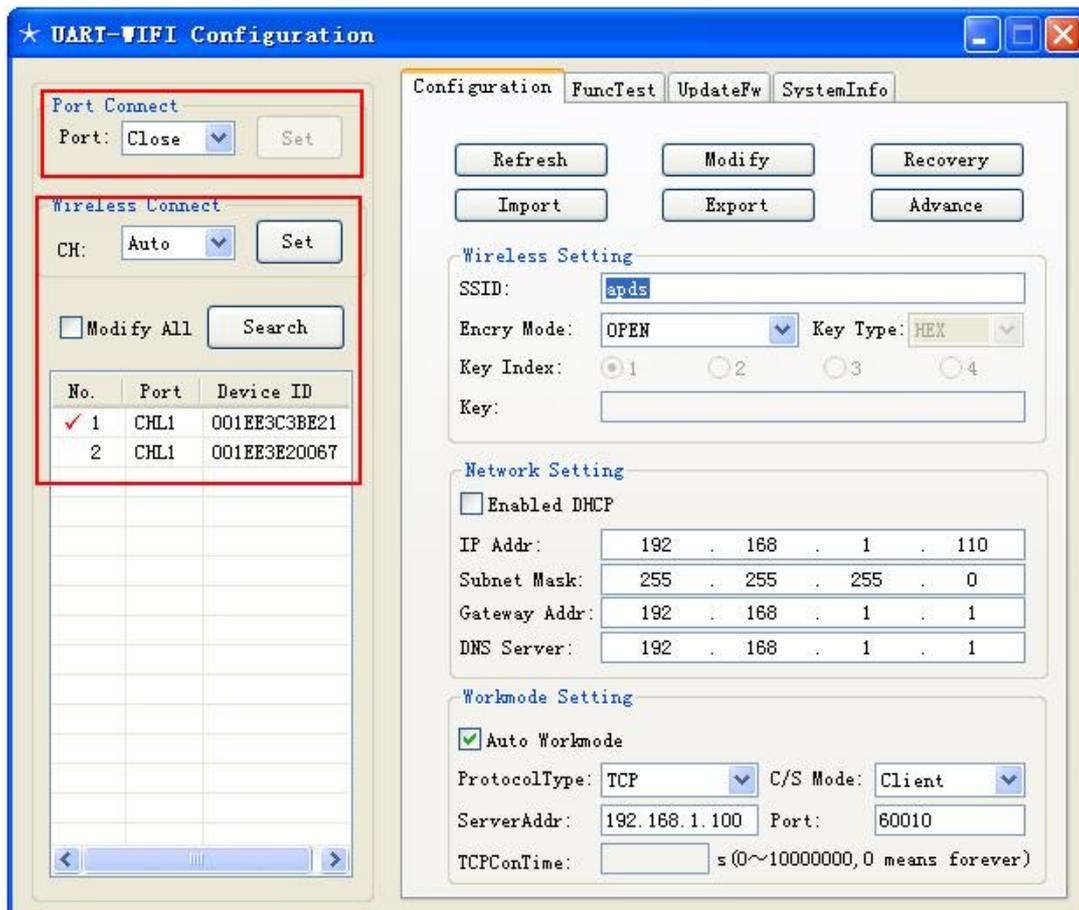


Figure5-3 sketch map of wireless connection

### 5.1.3 Configuring parameter

This section offers the function of fast querying/modifying configuration parameter of module for user.

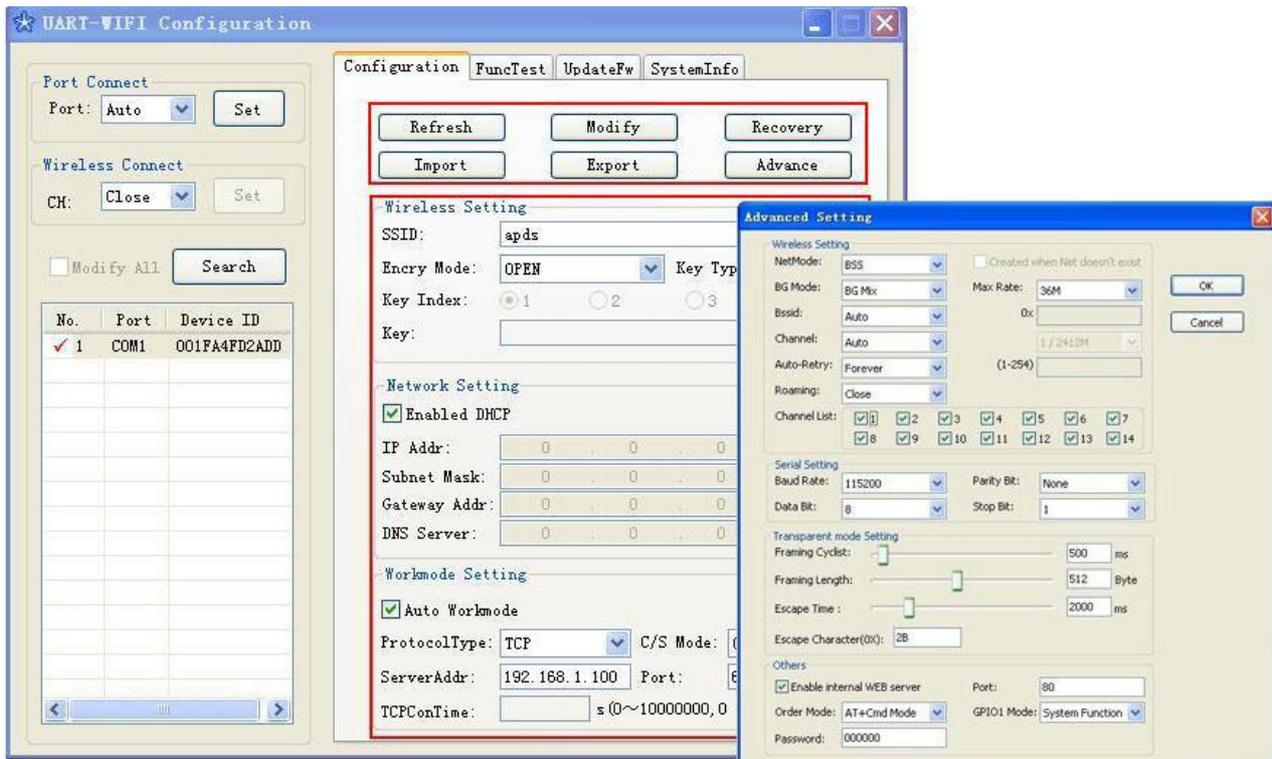


figure5-4 configure parameters

### 5.1.3.1 Refresh parameter

User can refresh the configuration parameter from wireless module and show in the display interface through clicking the 'refresh parameter' button.

### 5.1.3.2 Submit modification

User can click 'submit modification' button after modifying configuration parameter (including parameter in advanced setting) in the display interface to save the parameters into wireless module. New parameters will become effective until wireless module has been reset, user can select to reset at once or manually reset later in pop-up dialog box after modifying, it's shown as the following sketch map,

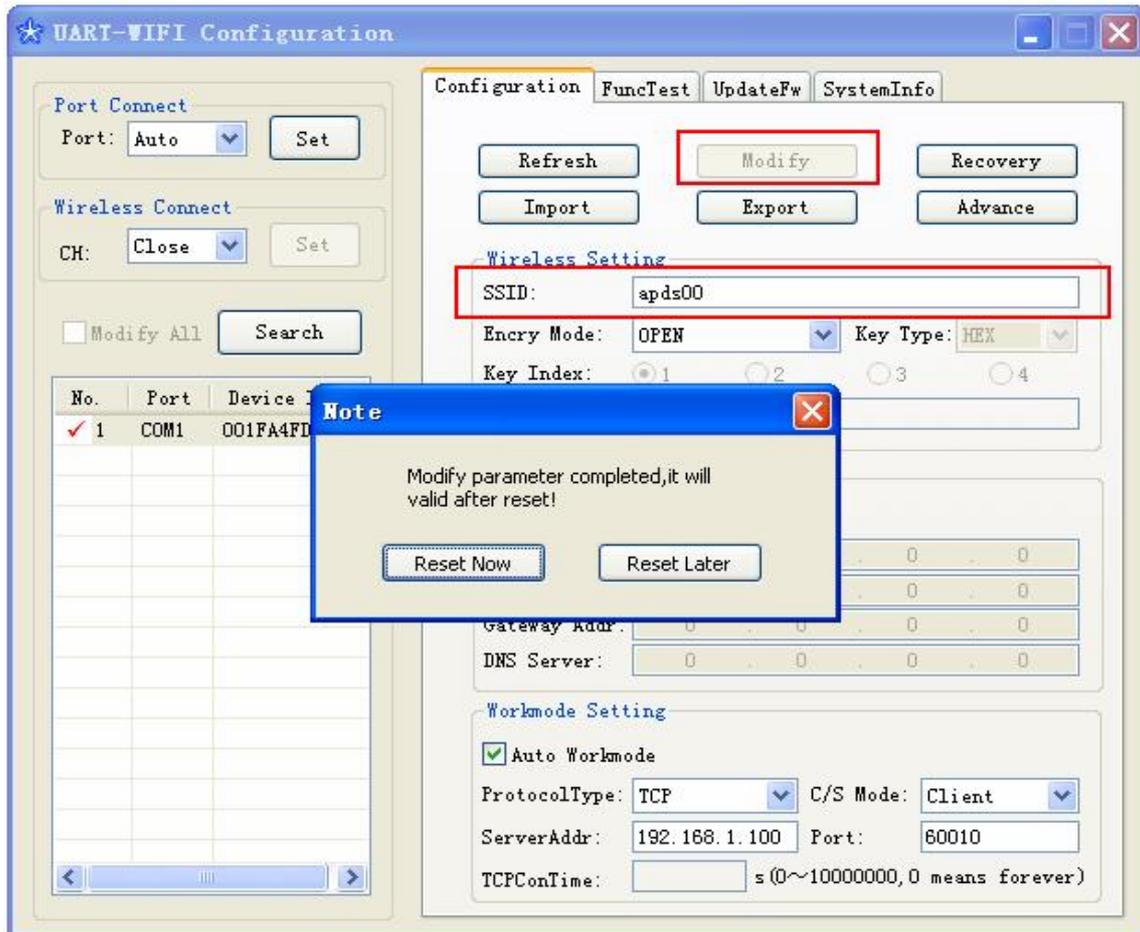


figure5-5 sketch map of submitting modification

### 5.1.3.3 Restore to factory setting

User clicks " " button to restore the configuration parameters of the wireless module into factory status. After restoring, the factory setting becomes effect until the module has been reset.

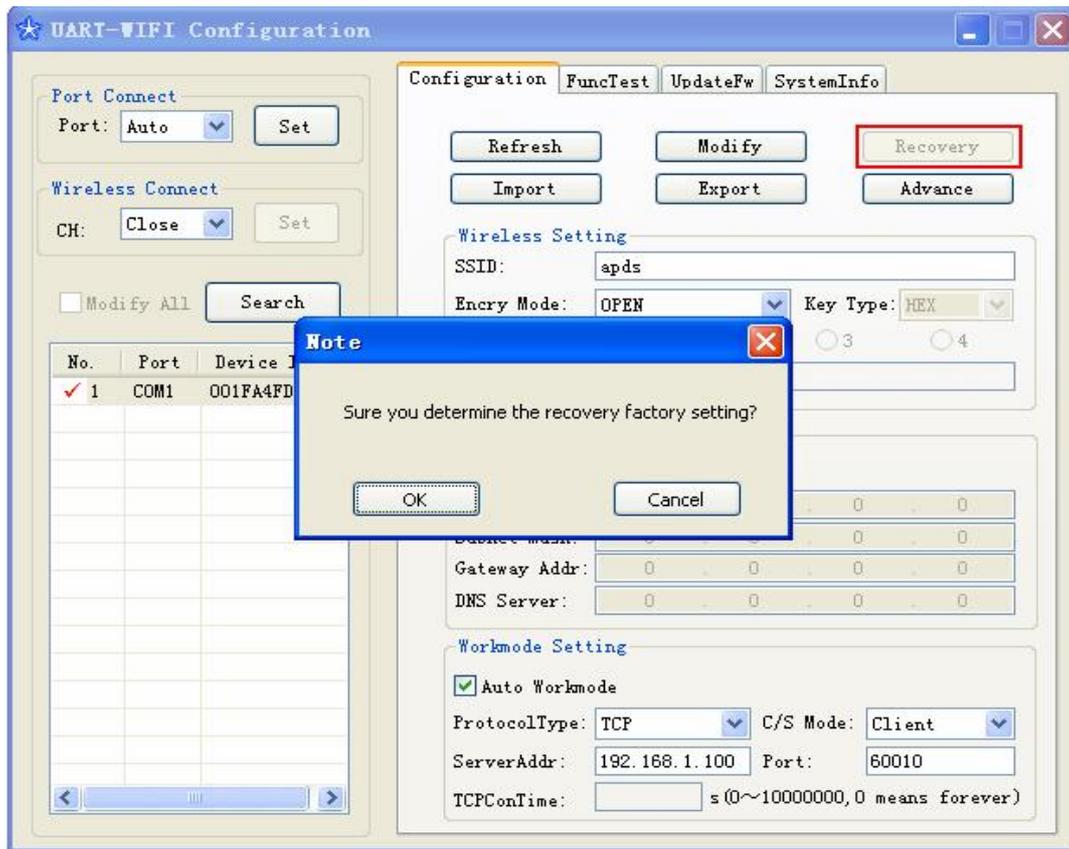


figure5-6 sketch map of restoring to factory setting

#### 5.1.3.4 Import/export parameters

User can save all configuration parameters in display interface to the specified configuration file through clicking the 'export parameters' button, and apply all parameter which are saved in configuration file through clicking 'import parameter' button. What should be noted is that, using import parameter function only imports parameters to the display interface, the imported parameters can be wrote in wireless module actually until clicking the 'submit modification'.

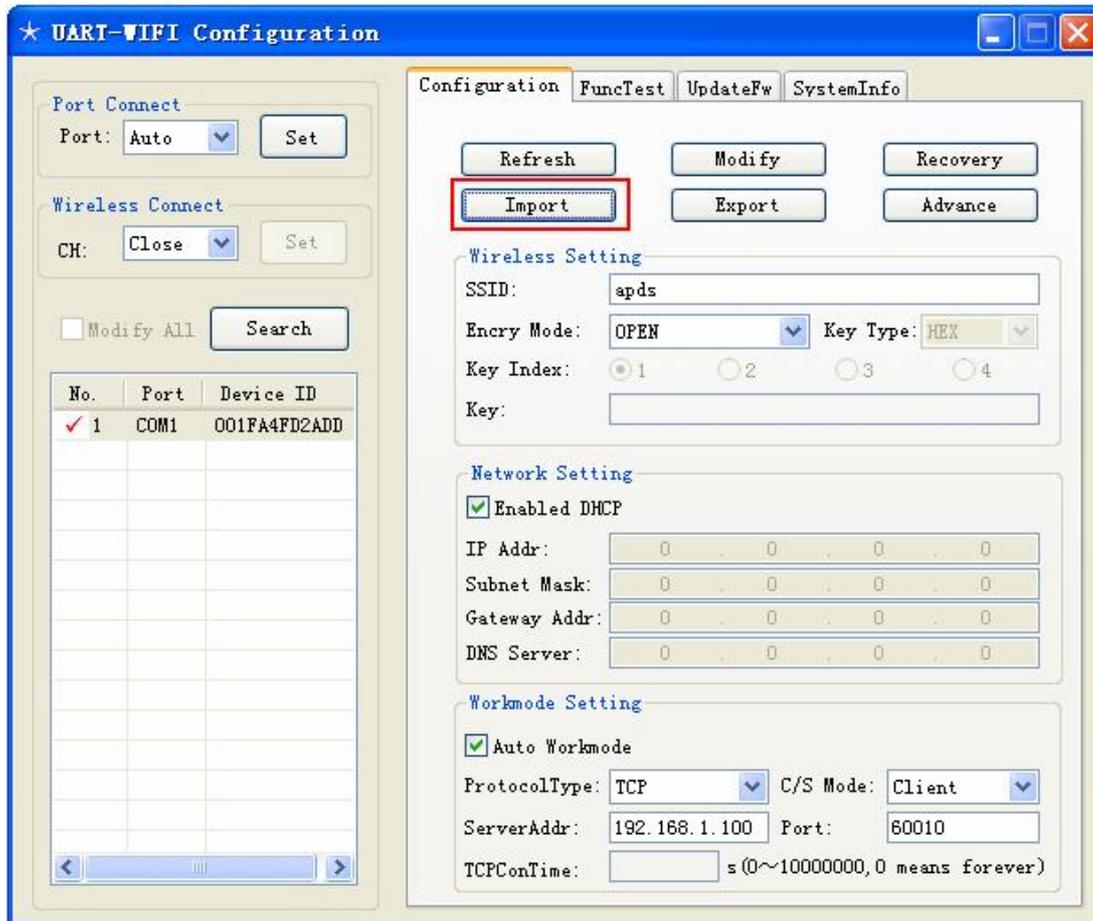


figure5-7 sketch map of Importing parameters

#### 5.1.3.5 Group modification

Group modification will be activated when configuration hypervisor has searched for more than one wireless module. If user selects the function, the modified parameters will be applied to all of the wireless modules automatically through clicking 'modify all'. This function can predigest operation steps and enhance work efficiency when a lot of modules need to be restored to the default configuration.

#### 5.1.4 Function test

This section offers the function of fast testing AT+ instruction for user, the user can jump over this section when using auto-work mode. Additionally, this function can not work while the module is configured through wireless connection.

User can send command through shortcut button of usual instruction or inputting AT+ instruction directly, it is shown as the following sketch map,



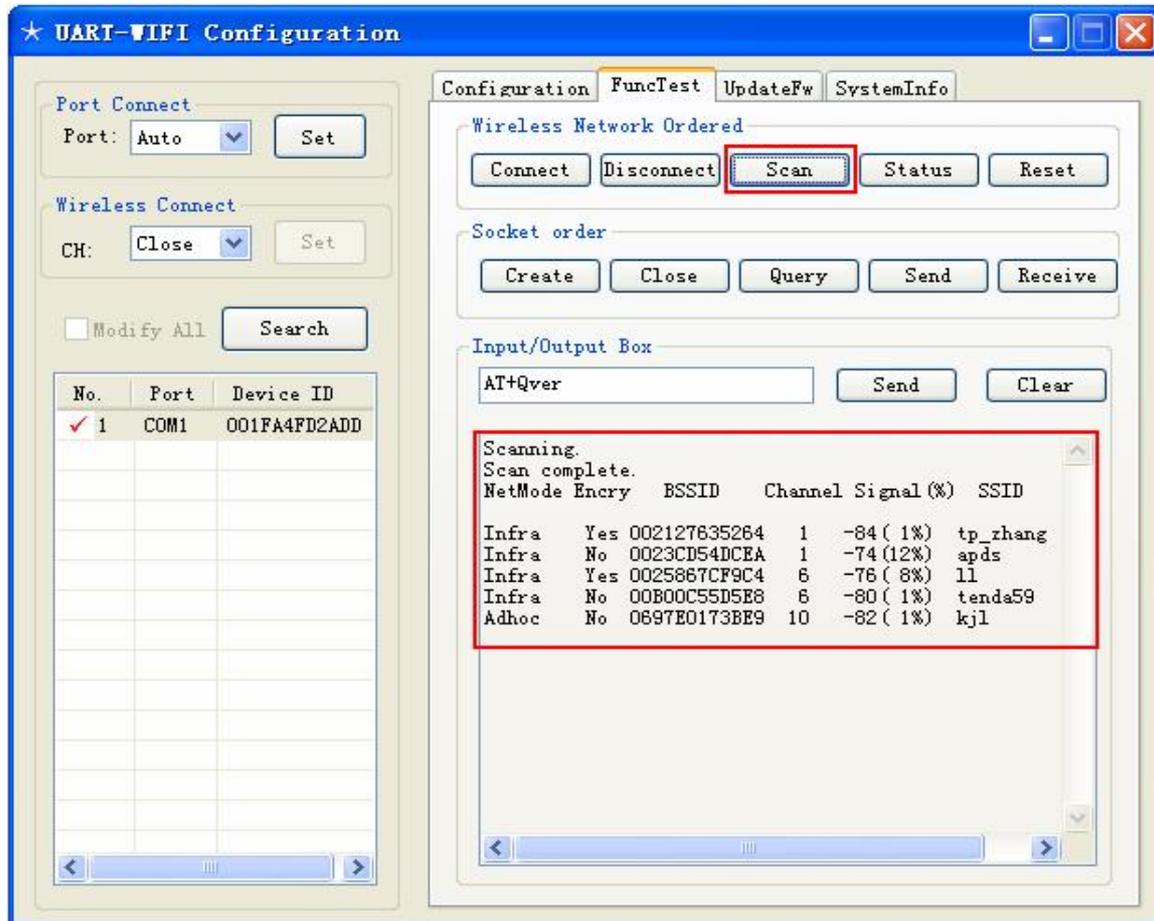


figure5-9 network scanning

#### 5.1.4.2 Join/Disconnect network

User can click the " " button to join or disconnect the wireless network as the following sketch map, the result will show in the following display frame.

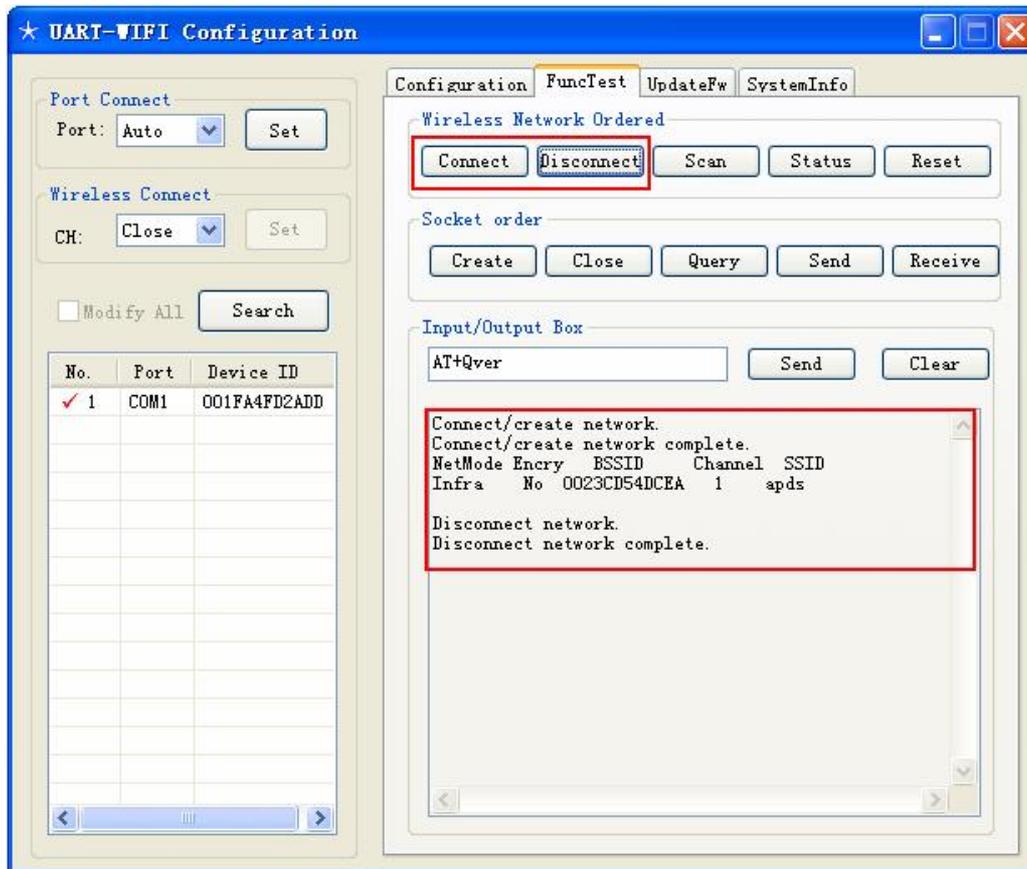


figure5-10 join/disconnect network

#### 5.1.4.3 Query network status

User can click the 'state' button to query network connection status as following sketch map, the result will show in the following display frame.

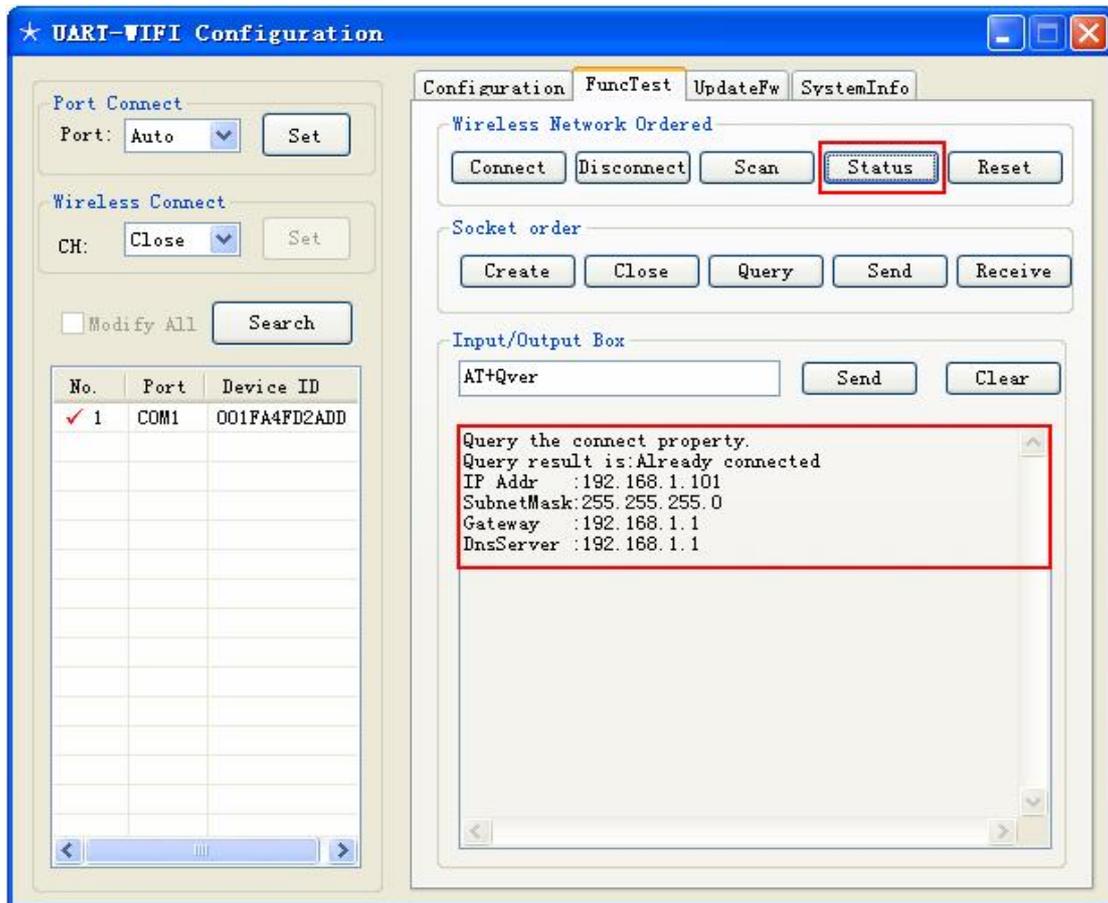


figure5-11 query network status

#### 5.1.4.4 Create socket

User can click 'create' button to create socket as the following sketch map. Click the 'Yes' button after having configured each parameter. It will return a socket number after creating successfully.

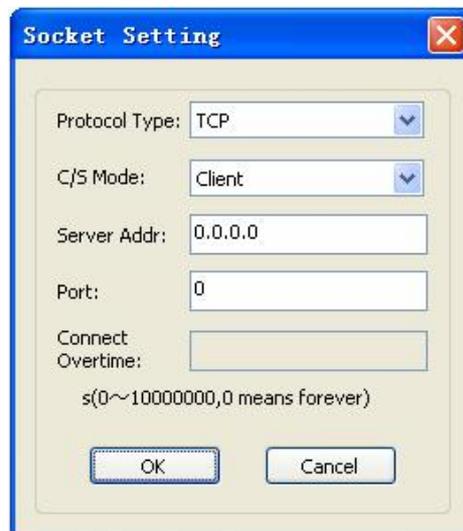


figure5-12 interface of creating socket

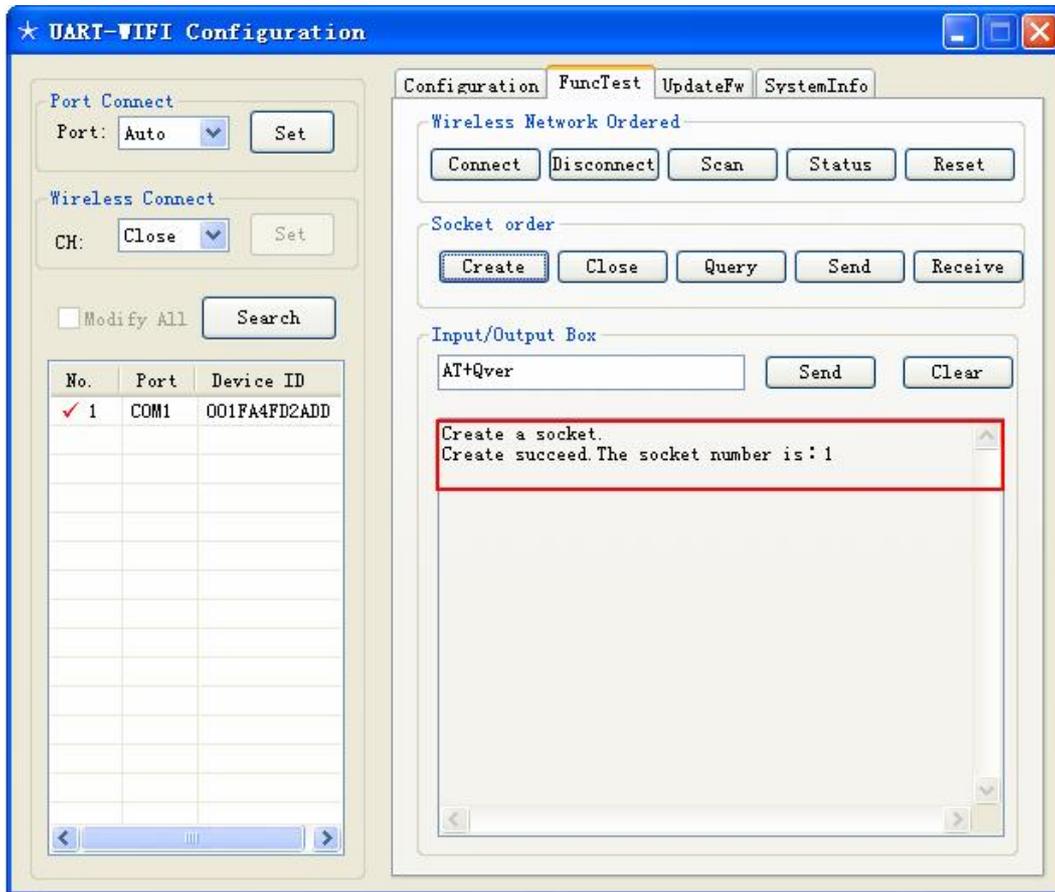


figure5-13 interface of the result of creating socket

#### 5.1.4.5 close socket

User can delete the socket connection through clicking 'close' button and inputing the socket number which is expected to close.

#### 5.1.4.6 query socket status

User can query socket status through clicking 'query' button and inputing the socket number which is wanted, the state of socket will return in the display frame after clicking 'yes' button.

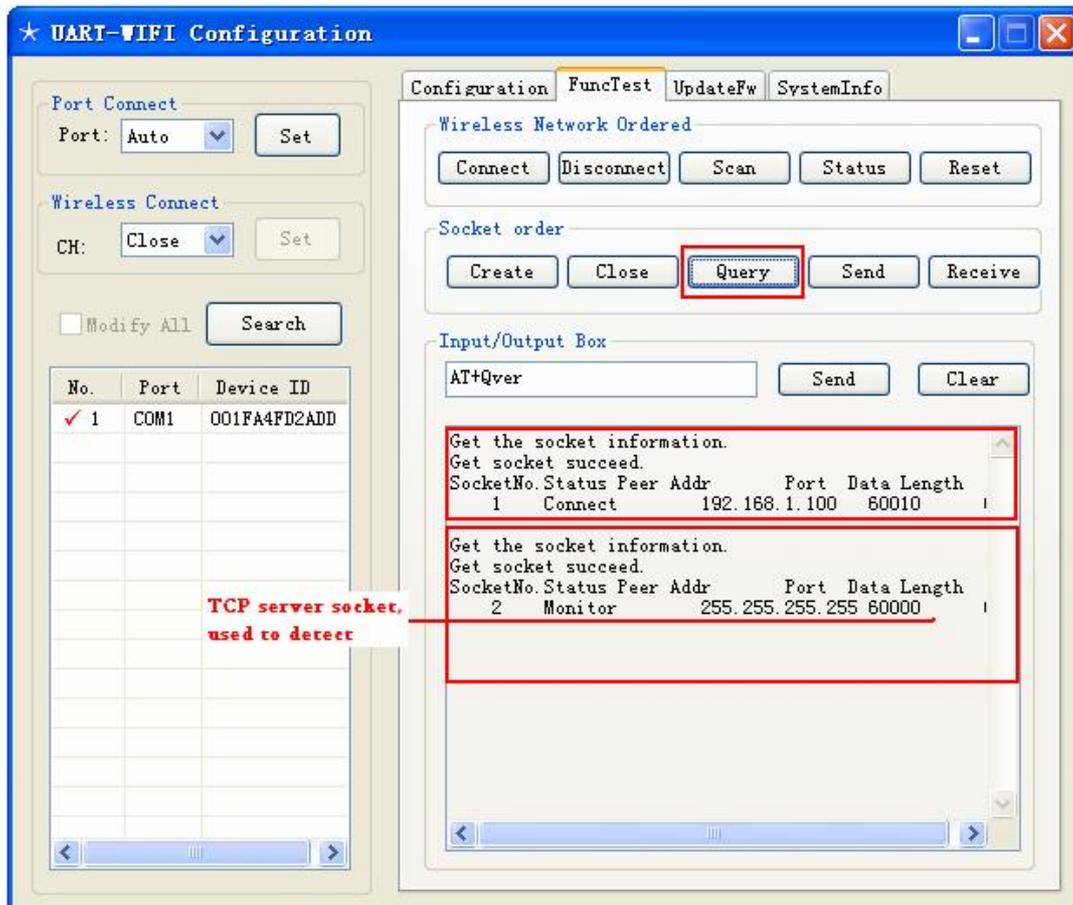


figure5-14 socket status querying

Description,

- 1、As shown above, the querying results return from two sockets are different. Because the second socket type is TCP server, which is in monitor state after creating successfully, its result does not only contain itself state, but also includes the connection information of Clients which are connecting to the server (these sockets are created by system automatically), as socket3 and socket4 shown in sketch map.

#### 5.1.4.7 Socket sending

User can send data through socket by clicking 'send' button and inputting socket number (the socket of TCP server can't be used to send or receive data directly, what is input should be the socket number of the connection between clients and the server) and data that need to send, the sending accomplishment information will return in the display frame after clicking 'yes' button.

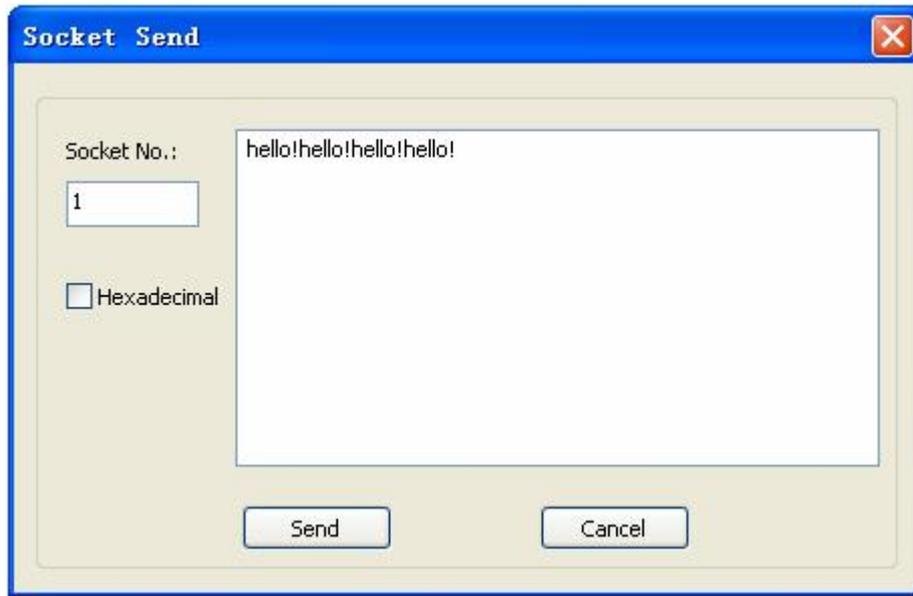


figure5-15 send data through socket

#### 5.1.4.8 Socket receiving

User can receive data through socket by clicking 'receive' button and inputing socket number (the socket of TCP server can't be used to send or receive data directly, what is input should be the socket number of the connection between clients and the server) and data size that expected to receive in pop-up dialog box, the data will show in the receive frame after clicking 'receive' button shown as figure5-18.

Before receiving data, please use 'query' command to confirm whether the socket receiver buffer has saved data in it at first.

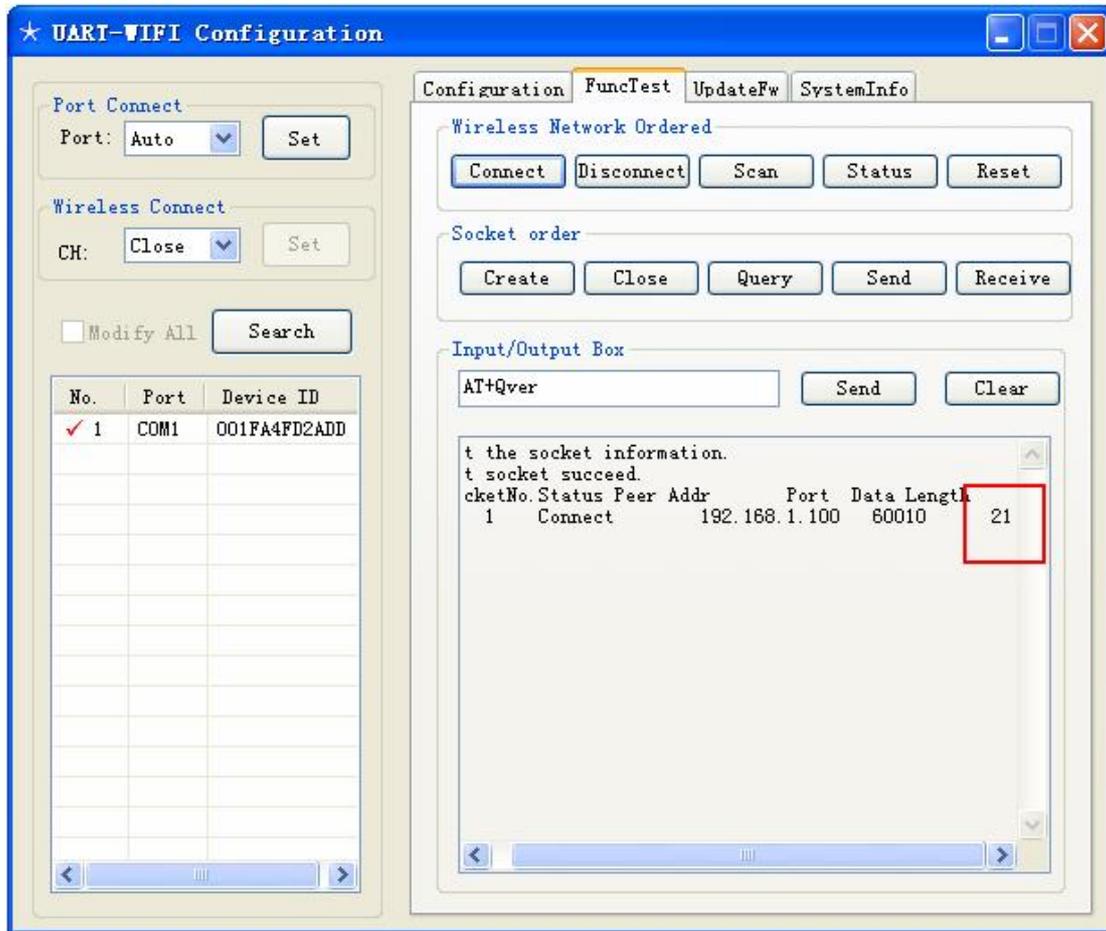


figure5-16 query the receiver buffer

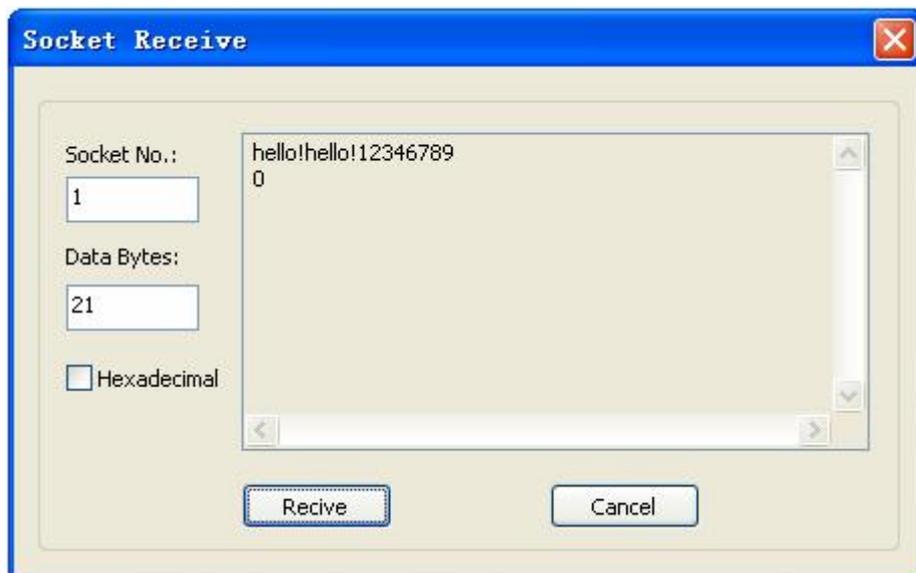


figure5-17 socket receiving

### 5.1.5 System information

User can query MAC address、hardware version information and firmware version information of the module in this section, the following sketch map is the show.

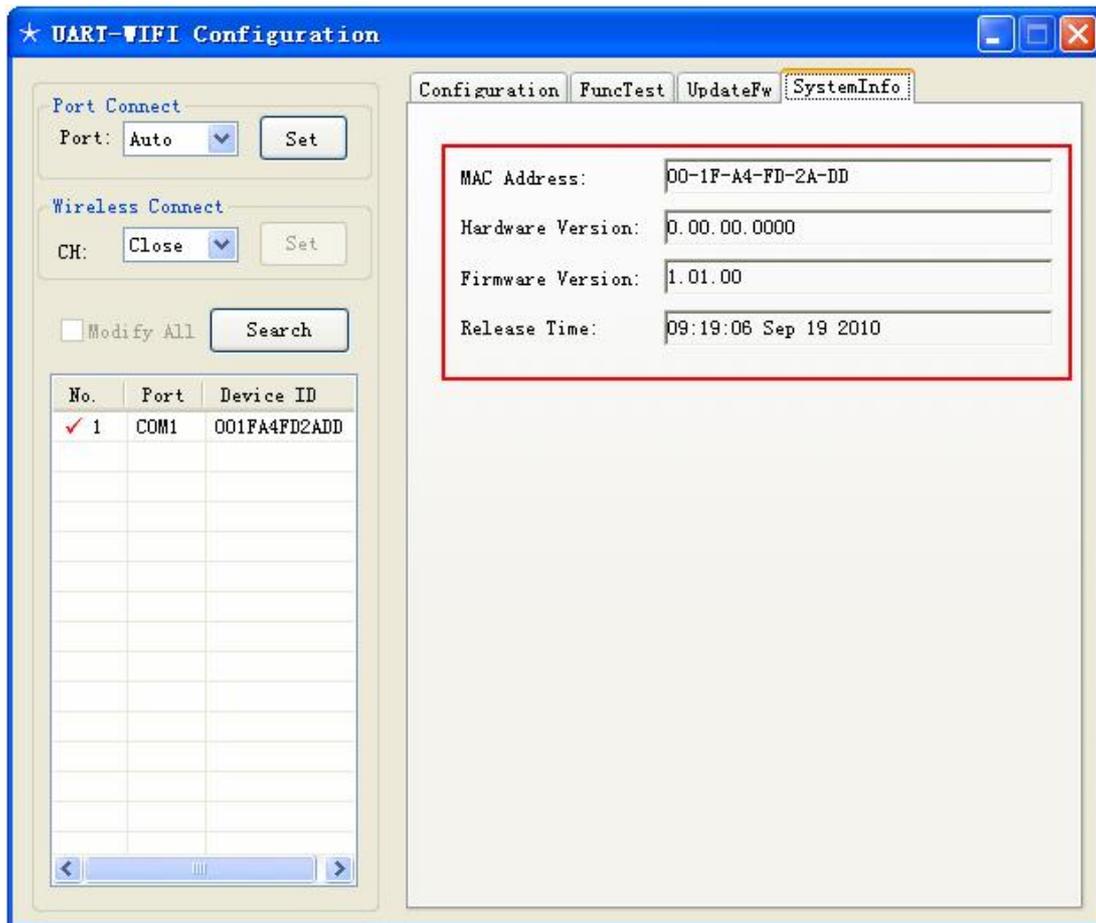


figure5-18 system information

### 5.1.6 Firmware upgrade

This function will be activate when the module has connected to the wireless network.

User can acquire the function of module firmware upgrade in this section, as the following sketch map shows ( what should be noted is that, please read the notice carefully before upgrading, otherwise, failure will make the module destroyed ) .

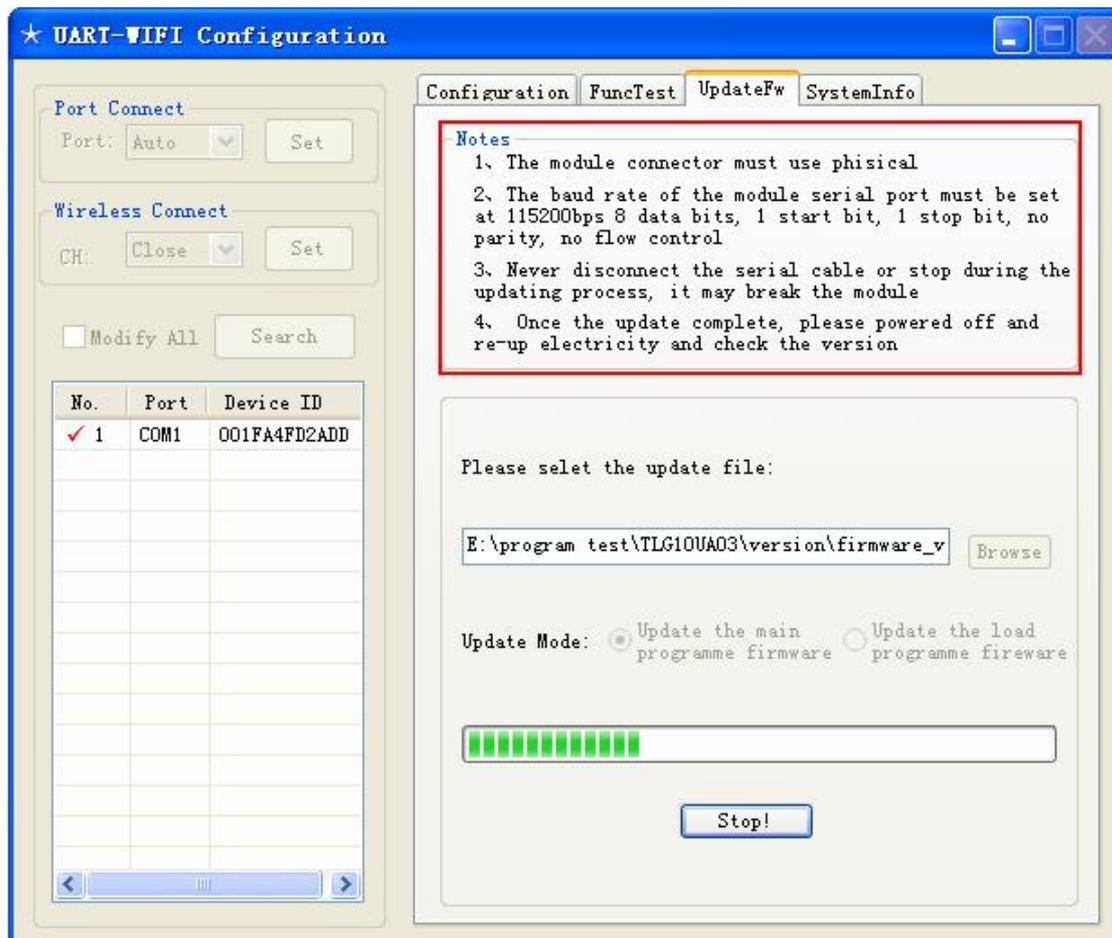


figure5-19 firmware upgrade

## 5.2 Configuration through WEB server

WEB server has been embedded in this product, it can support the function of parameters configuration with IE browser, the method is as follow,

Notices,

- 1、 Make sure the module has connected to the network successfully before using web page configuration;
- 2、 The default port number of the WEB server is 80, so it only needs to input address in address column of browser, if user has modified the web server port, user must input port number, otherwise the net can not be opened, for example when the port number is set to 8080, the access address of the web server is <http://192.168.1.20:8080>;
- 3、 Please use Windows IE browser to access the server as well as you can;

### 5.2.1 Query/set configuration parameter

1. Open IE browser, connect after inputting IP address (and the port number sometimes) of wireless module in address column, the login interface will appear, input user name 'admin' and login password of module (default password is 000000);
2. The configuration web page is shown as the following sketch map, if user want to modify configuration parameters of wireless module, you need to click 'save' button to submit the modification;

figure5-20 web page of parameter modification

3. After completing the submission, the wireless module will restart automatically, the web page will be also refreshed in ten seconds.

### 5.2.2 Firmware upgrade on line

The product supports firmware upgrade online through web page, it's shown as following sketch map,

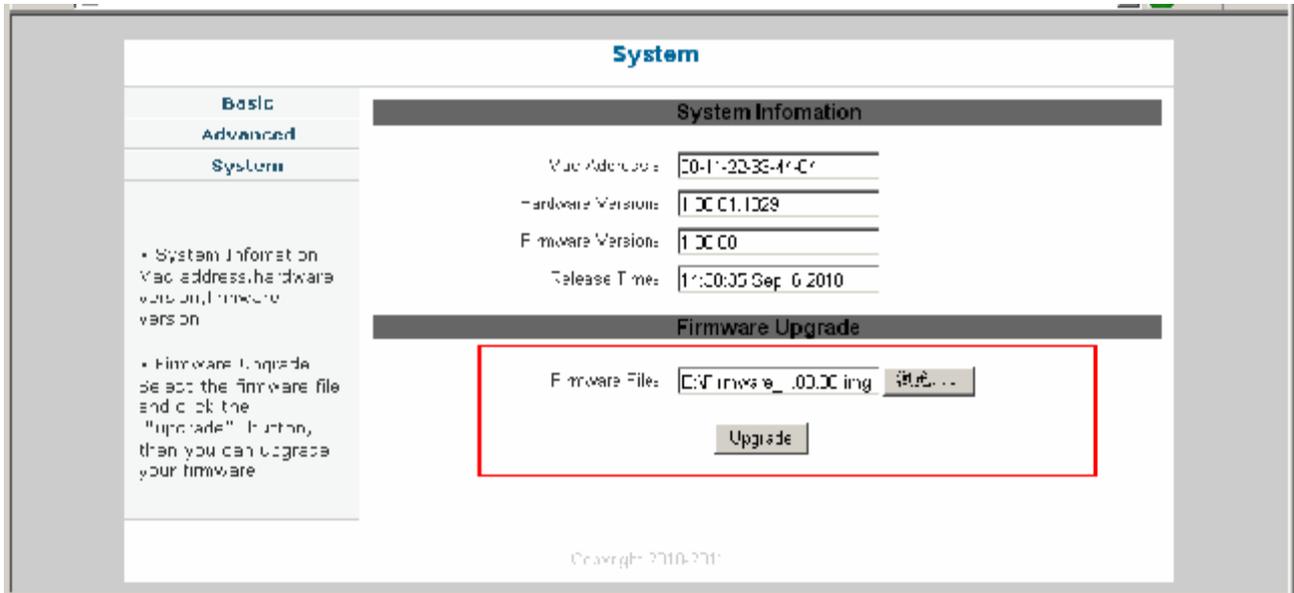


figure5-21 online upgrade of firmware

### 5.3 Configuration through super terminal program

User can ignore the section if only using the wireless module in auto-work mode.

For the user who hopes to test the function of AT+instruction, Windows Super Terminal which is embedded in Windows operation system can be used to input AT+ instruction directly for parameter configuration of wireless module and function test. What should be described is that, the serial debugging softwares (serial genius、serial master, etc) that referred in preamble all can be used for AT+instruction test, this section just takes super terminal program as an example.

- 1、Open super terminal program (Windowsstart—program—accessories—communication—super terminal program), input connection name and icon, select serial number and parameters (the same as the wireless module), what should be described is that the option of flow control must be set to none in command mode, the setting is shown as following sketch map,

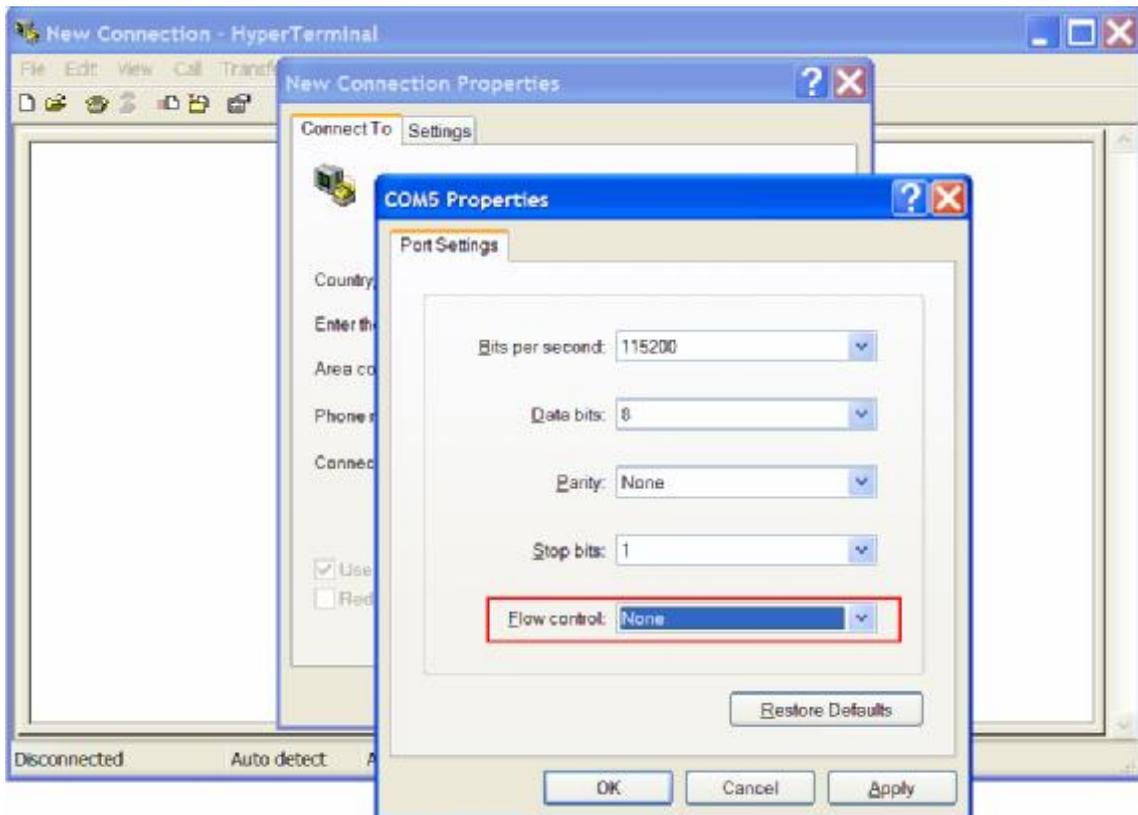
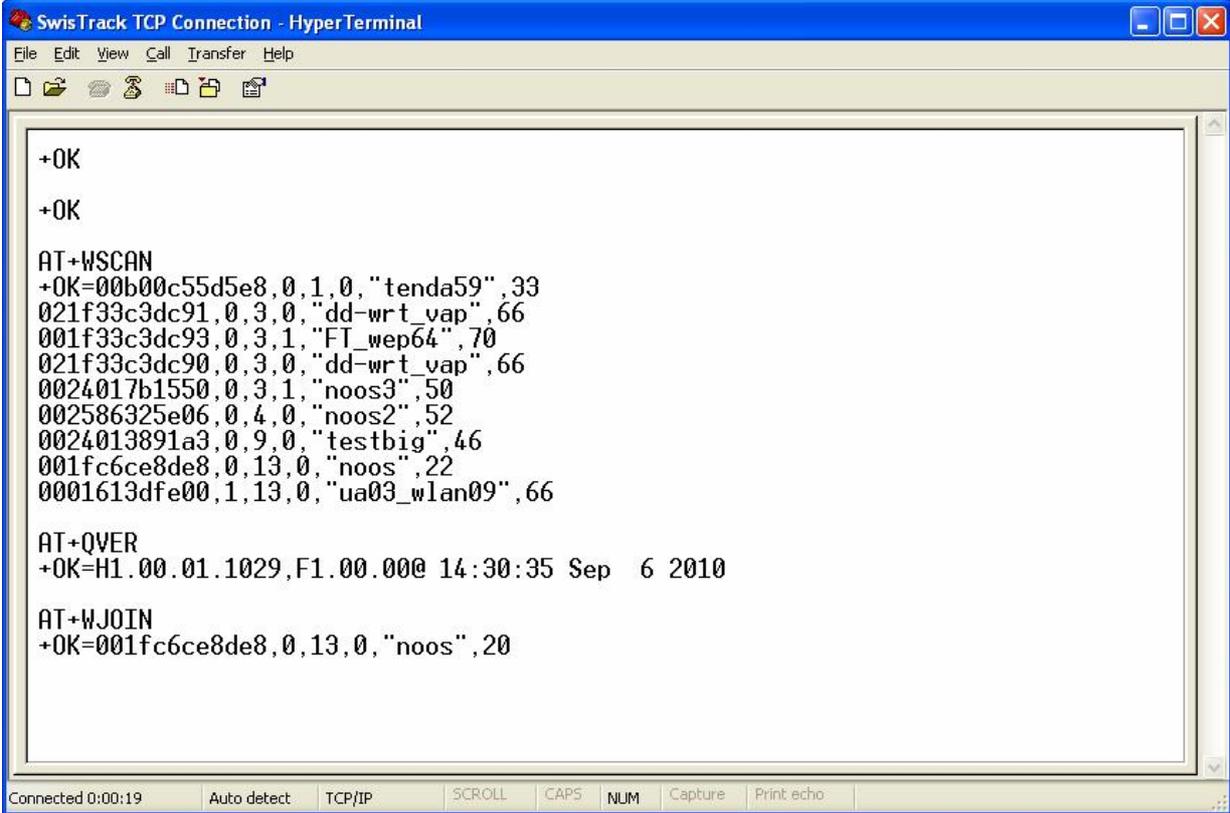


figure5-22 the configuration of super terminal

- 2、 If the wireless module has started configuration mode (see the 4.2.1 section) or the work mode is set to command mode, please jump to step 5 directly, otherwise, it needs to make the module exiting the transparent transmission mode according to following approaches at first (refer to section 4.2.2.4);
- 3、 Open a notepad to edit escape character string (the default escape character is '+' ) and save.
- 4、 Wait for the time specified by the parameter of escape time (its default setting is 2 second), select 'send text file' in the 'transmit' menu, send escape character string file, the module return '+OK' information if success;
- 5、 Input 'AT+E' instruction to open input return display (before it, the character that user input can not be display in the super terminal interface);
- 6、 Thus, user can use super terminal program to test the most of AT+ instruction set, for example, the command of network scan can be insteaded of inputting 'AT+WSCAN', it's shown as the following sketch map.



```
+OK
+OK
AT+WSCAN
+OK=00b00c55d5e8,0,1,0,"tenda59",33
021f33c3dc91,0,3,0,"dd-wrt_vap",66
001f33c3dc93,0,3,1,"FT_wep64",70
021f33c3dc90,0,3,0,"dd-wrt_vap",66
0024017b1550,0,3,1,"noos3",50
002586325e06,0,4,0,"noos2",52
0024013891a3,0,9,0,"testbig",46
001fc6ce8de8,0,13,0,"noos",22
0001613dfe00,1,13,0,"ua03_wlan09",66

AT+QVER
+OK=H1.00.01.1029,F1.00.00@ 14:30:35 Sep  6 2010

AT+WJOIN
+OK=001fc6ce8de8,0,13,0,"noos",20
```

Connected 0:00:19 | Auto detect | TCP/IP | SCROLL | CAPS | NUM | Capture | Print echo

figure5-23 instruction test

## 6 AT+Instruction Control Protocol

User can ignore this chapter if only using auto-work mode.

### 6.1 Syntax Introduction

This module uses AT+Instruction protocol as user control protocol. AT+Instruction protocol is based on the instruction of ascii command style, The description of syntax format and management process is as follow.

#### 6.1.1 Syntax Format

##### n Format Description

< > means the parts must be included

[ ] means the optional part

##### n Command Message

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

'AT+' means the leader of Command message

'CMD' means command string

'[op]' means the symbol of Command operator, when the command requires parameters, it can specify the operation type of parameters, it includes,

'=' means the leader symbol of parameters or returns

'=!' means to synchronize the modified parameter to flash in the command of parameters setting type

'=?' means to query the current setting in the command of parameters setting type

<CR> means enter, it's 0x0d in ASCII

##### n Response message

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

'+' means the leader of response message

RSP means response string

'OK' means success

'ERR' means fail

<CR> means enter, it's 0x0d in ASCII

<LF> means newline, it's 0x0a in ASCII

## n Data Type

String is enveloped with double quotation marks, but its content does not include the quotation mark, such as "this is a string".

Dec means number in decimalist, such as 10.

Hex means number in hexadecimal, such as a.

IP means IP address string, such as 192.168.0.1.

MAC means MAC address, it's formed with 12 hexadecimal numbers, such as 001EE3A80102.

### 6.1.1.1 Process Flow

AT+Instruction protocol uses the form of command+response, most of instructions need a response message after the receiver has processed the instruction, if the receiver receive a new command acquire in the process of processing the previous command, it will throw the command away stealthily without any response message, the process is shown as following figure.

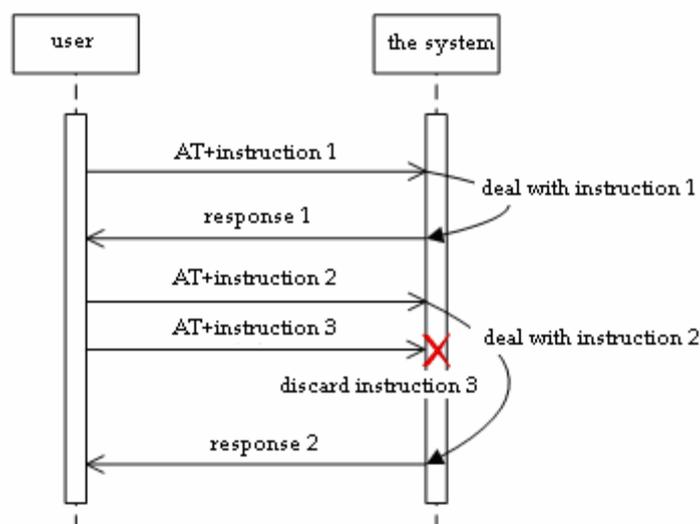


Figure6-1 Process flow of AT+Instruction

Some special commands, such as AT+SKSND and AT+SKRCV, need to transmit binary data behind command or response, at this moment, data receiver becomes into diaphanous transmission status provisionally, and begins to receive binary numbers flow, until it has received the data, of which length is prescribed in the <SIZE> field of command or

message, or waits for timeout, then it exits transparent transmission mode, the flow is shown as follow,

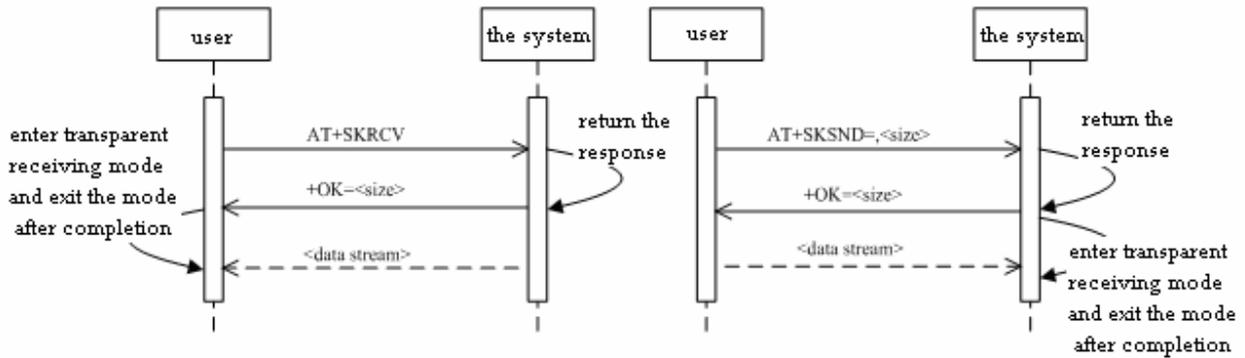


figure6-2 AT+Instruction special process flow

### 6.1.1.2 Format Examples

Example 1: responding success message

```
AT+
+OK
```

Example 2: responding failed message

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

Example 3: Using input parameter

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

Example 4: Using the symbol <!>

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

Example 5: Using the symbol <?>

AT+ATPT=?  
+OK=500

### 6.1.1.3 Error Code

Table 6-1 Error Code

| Value | meanings                     |
|-------|------------------------------|
| -1    | Ineffective Command Format   |
| -2    | Non supported                |
| -3    | Ineffective Operation Symbol |
| -4    | Ineffective Parameter        |
| -5    | Not Permitted                |
| -6    | Memory Lack                  |
| -7    | FLASH Error                  |
| -10   | Join Failed                  |
| -11   | No Available socket          |
| -12   | Ineffective Socket           |
| -13   | Socket Connect Failed        |
| -100  | Undefined                    |

## 6.2 Instruction Set

### 6.2.1 Instruction List

Table 6-2 AT+Instruction list

| name                   | use  |
|------------------------|--|
| <a href="#">(null)</a> | Null   |
| <a href="#">ATLT</a>   | Set or query data length of automatic framing  |
| <a href="#">ATM</a>    | Set or query work mode of module   |
| <a href="#">ATPT</a>   | Set or query period of automatic framing   |
| <a href="#">ATRM</a>   | Set or query connection message of socket module creates automatically in auto-work mode |
| <a href="#">BSSID</a>  | Set or query bssid address of target AP  |

|                              |  |
|------------------------------|--|
| <a href="#"><u>CHL</u></a>   | Set or query target wireless channel number                |
| <a href="#"><u>CHLL</u></a>  | Set or query wireless channel list                         |
| <a href="#"><u>CMDM</u></a>  | Set or query default command mode of system                |
| <a href="#"><u>E</u></a>     | Switch serial character to show back                       |
| <a href="#"><u>ENCRY</u></a> | Set or query security mode of wireless network             |
| <a href="#"><u>ENTM</u></a>  | Set module into transparent transmission mode              |
| <a href="#"><u>ENTS</u></a>  | Enter sleeping mode  |
| <a href="#"><u>ESPC</u></a>  | Set or query escape character                              |
| <a href="#"><u>ESPT</u></a>  | Set or query escape time                                   |
| <a href="#"><u>IOC</u></a>   | GPIO Control   |
| <a href="#"><u>IOM</u></a>   | Set or query GPIO mode                                     |
| <a href="#"><u>KEY</u></a>   | Set or query key of wireless network                       |
| <a href="#"><u>LKSTT</u></a> | Query network connection state                             |
| <a href="#"><u>NIP</u></a>   | Set or query local IP address                              |
| <a href="#"><u>PASS</u></a>  | Set or query system password                               |
| <a href="#"><u>PMTF</u></a>  | Update parameters in memory to Flash                       |
| <a href="#"><u>QMAC</u></a>  | Query physical address                                     |
| <a href="#"><u>QVER</u></a>  | query version  |
| <a href="#"><u>RSTF</u></a>  | Return to manufacturer setting                             |
| <a href="#"><u>SKCLS</u></a> | Close socket connection                                    |
| <a href="#"><u>SKCT</u></a>  | Create socket connection                                   |
| <a href="#"><u>SKRCV</u></a> | Receive data through socket connection                     |
| <a href="#"><u>SKSDF</u></a> | Set default sending socket connection                      |
| <a href="#"><u>SKSND</u></a> | send data through socket connection                        |
| <a href="#"><u>SKSTT</u></a> | query socket connection state                              |
| <a href="#"><u>SSID</u></a>  | Set or query SSID  |
| <a href="#"><u>UART</u></a>  | Set or Query serial data format                            |
| <a href="#"><u>WARC</u></a>  | Set or query automatic retry times in auto-work mode       |
| <a href="#"><u>WARM</u></a>  | Set or query wireless roam                                 |
| <a href="#"><u>WATC</u></a>  | Set or query whether to create Adhoc network automatically |
| <a href="#"><u>WBGR</u></a>  | Set or query wireless BG mode and maximum send rate        |
| <a href="#"><u>WEBS</u></a>  | Set or query web service                                   |
| <a href="#"><u>WJOIN</u></a> | Join in or create network                                  |
| <a href="#"><u>WLEAV</u></a> | Disconnect network   |

|                       |                            |
|-----------------------|----------------------------|
| <a href="#">WPRT</a>  | Set or query wireless mode |
| <a href="#">WSCAN</a> | Scan for network           |
| <a href="#">Z</a>     | Reset                      |

## 6.2.2 Network Control Type

### 6.2.2.1 AT+WJOIN

#### Function:

Join/Create a wireless network.If the type of current network is adhoc,and the module can not search for the same ssid network,then it automatically creates a new network,or it has been connected,then return the message of network connection.

#### Format:

AT+WJOIN<CR>

+OK=<ssid>,<type>,<channel>,<b\_encry>,<ssid>,<rsi><CR><LF><CR><LF>

#### Parameter:

*ssid* means network BSSID,it is formed with twelve hexadecimal Numbers,and the format is 001EE3A34455

*type* means Network Type

Table 6-3 Network Type

| Value | meanings      |
|-------|---------------|
| 0     | Infra network |
| 1     | adhoc network |

*channel* means Channel Number

*b\_encry* means Encryption Mode

Table 6-4 Encryption Mode

| Value | Meanings   |
|-------|------------|
| 0     | Open       |
| 1     | Encryption |

*ssid* means Wireless network name,it's formed with 1~32 Character and embedded by double quotation marks

*rsi* means Network Signal Strength,it does not include minus,its unit is DBm, For example 50 means that the strength is -50 DBm.

### 6.2.2.2 AT+WLEAV

#### Function:

Disconnect the current network。

**Format:**

```
AT+WLEAV<CR>
+OK<CR><LF><CR><LF>
```

**Parameter:**

None

### 6.2.2.3 AT+WSCAN

**Function:**

Scan and return after finishing.

**Format:**

```
AT+WSCAN<CR>
+OK=<bssid>,<type>,<channel>,<b_encry>,<ssid>,<rssi><CR><LF>
    <bssid>,<type>,<channel>,<b_encry>,<ssid>,<rssi><CR><LF>
    .....
    <CR><LF>
```

**Parameter:**

Same as AT+WJOIN

### 6.2.2.4 AT+LKSTT

**Function:**

Query the network connection status。

**Format:**

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

**Parameter:**

Status means Connection Status

Table 6-5 Connection Status

| Value | Meanings   |
|-------|------------|
| 0     | Disconnect |
| 1     | connect    |

*ip* means IP address,its format is "192.168.1.22",it does not include quotation marks

*netmask* means Subnet Mask,its format is same as IP address

*gateway* means Gateway address,its format is same as IP address

*dns* means DNS address,its format is same as IP address

## 6.2.2.5 AT+SKCT

**Function:**

Create a socket connection. In client mode, module wait for connection is over (success or failure) to return, and in server mode it return after creating a network.

**Format:**

AT+SKCT=[*protocol*],[*cs*],[*host\_timeout*],<*port*><CR>  
+OK=<*socket*><CR><LF><CR><LF>

**Parameter:**

*protocol*: Protocol Type,

Table 6-6 Protocol Type

| Value | Meanings |
|-------|----------|
| 0     | TCP      |
| 1     | UDP      |

*cs*: C/S mode,

Table 6-7 C/S mode

| Value | Meanings |
|-------|----------|
| 0     | Client   |
| 1     | Server   |

the meanings of *host\_timeout* is shown as follows according to protocol and C/S mode.

Table 6-8 host\_timeout

| cs | protocol | Meanings   |
|----|----------|--|
| 0  | X        | Name of target server, it can be filled with domain name or ip address. For example "192.168.1.100" or " <a href="http://www.sina.com.cn">www.sina.com.cn</a> "  |
| 1  | 0        | TCP connection timeout period, it means client will be automatically kicked out from server if it send nothing in effective period; Effective value range is 1~10000000, Unit: second, 0 means never, default setting is 120 seconds |
| 1  | 1        | No Meanings  |

*port* means Port number

*socket* means socket number

#### 6.2.2.6 AT+SKSND

##### **Function:**

Send data through the specified socket and return when it is over. This command send binary data, User should send original data after receiving the response(+OK) of module. Module automatically exits transmission after receiving the data of specified length and send data to the network, the redundant data will be rejected. Otherwise, module will be forced to constrainedly exit transmission mode and send the data has been received to the network when the module has waiting for timeout period(1 second).

##### **Format:**

```
AT+SKSND=<socket>,<size><CR>
+OK=<actualsize><CR><LF><CR><LF>
[data stream]
```

##### **Parameter:**

*socket* means socket number

*size* means the size of the data which prepare to be send, its unit is type.

*actualsize* means the size of the data which actually is send, its unit is type.

*data stream* means original data.

#### 6.2.2.7 AT+SKRCV

##### **Function:**

Read data form RX buffer of the target socket ,and return after hasing finished. After receiving this command and sending the response(+OK) is over ,module will transmit the data of specified size in binary format

##### **Format:**

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

##### **Parameter:**

*socket* means socket number

*maxsize* means maximal data size allows to receive

*size* means actually size of the receiving data

*data stream* means original data

## 6.2.2.8 AT+SKSTT

**Function:**

Aquire the state of the specified socket, the first line of return value means the state of the message of the socket,if the socket mode is TCP server,then beginning from the first line, each line shows a socket state of a client which is joined in the server

**Format:**

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

**Parameter:**

*socket* means socket number

*status* means socket state

Table 6-9 Socket status

| Value | Meanings     |
|-------|--------------|
| 0     | Disconnected |
| 1     | Detecting    |
| 2     | Connected    |

*host* means IP address of the opposite terminal

*port* means port number of the opposite terminal

*rx\_data* means data size in RX buffer

## 6.2.2.9 AT+SKCLS

**Function:**

Close the specified socket

**Format:**

```
AT+SKCLS=<socket><CR>
+OK<CR><LF><CR><LF>
```

**Parameter:**

*socket* means socket number

## 6.2.2.10 AT+SKSDF

**Function:**

Set the default sending socket of system.If user needs to enter transparent transmtion

mode in command mode, using this command can send the transparent data from serial port to the destination.

**Format:**

```
AT+SKSDF=<socket><CR>
+OK<CR><LF><CR><LF>
```

**Parameter:**

socket means socket number

### 6.2.3 System control type

#### 6.2.3.1 AT+

**Function:**

Null command

**Format:**

```
AT+<CR>
+OK<CR><LF><CR><LF>
```

**Parameter:**

None

#### 6.2.3.2 AT+Z

**Function:**

Reset system

**Format:**

```
AT+Z<CR>
+OK<CR><LF><CR><LF>
```

**Parameter:**

None

#### 6.2.3.3 AT+E

**Function:**

Switch serial command to return

**Format:**

```
AT+E<CR>
+OK<CR><LF><CR><LF>
```

**Parameter:**

None

#### 6.2.3.4 AT+ENTS

**Function:**

Enable system to enter to sleeping mode. System will be awaked automatically after receiving any at+instruction in sleeping state.

**Format:**

```
AT+ENTS<CR>  
+OK<CR><LF><CR><LF>
```

**Parameter:**

None

#### 6.2.3.5 AT+ENTM

**Function:**

Enable the serial to enter to transparent transmission mode. System exit this mode when it has received escape character according with escape term in transparent transmission mode

**Format:**

```
AT+ENTM<CR>  
+OK<CR><LF><CR><LF>
```

**Parameter:**

None

#### 6.2.3.6 AT+RSTF

**Function:**

Return to manufacturer settings in flash. System need to restart and then the settings become effective.

**Format:**

```
AT+RSTF<CR>  
+OK<CR><LF><CR><LF>
```

**Parameter:**

None

#### 6.2.3.7 AT+PMTF

**Function:**

Parameters which are saved in the memory are updated to the flash

**Format:**

AT+PMTF<CR>  
+OK<CR><LF><CR><LF>

**Parameter:**

None

6.2.3.8 AT+IOC

**Function:**

GPIO input or output control. IO status is allowed to read when GPIO1 is set to input mode( the mode of AT+IOM is 1), IO status is allowed to write when GPIO1 is set to output mode( the mode of AT+IOM is 2)

**Format:**

AT+IOC=[?][*status*]<CR>  
+OK[=*status*]<CR><LF><CR><LF>

**Parameter:**

*status* means IO Status

Table 6-10 IO Status

| Value | Meanings   |
|-------|------------|
| 0     | High level |
| 1     | Low level  |

6.2.3.9 AT+QMAC

**Function:**

Aquire physical address of the module.

**Format:**

AT+QMAC<CR>  
+OK=<*mac address*><CR><LF><CR><LF>

**Parameter:**

*mac address* is twelve hexadecimal Numbers, format is 001EE3A34455

6.2.3.10 AT+QVER

**Function:**

Aquire version of system.the version message includes hardware version and firmware version.

**Format:**

AT+QVER<CR>  
 +OK=<*hard,firm*><CR><LF><CR><LF>

**Parameter:**

*hard* means hardware version,its format is string,such as " H1.00.00.1029"

*firm* means firmware version, its format is string ,such as " F0.02.02@ 18:25:25 Jul 28 2010"

**6.2.4 Parameter Setting Type**

## 6.2.4.1 AT+NIP

**Function:**

Set or query local IP address.The dynamic IP address can not be queried if address type is set to DHCP.AT+LKSTT command can be used to query.

**Format:**

AT+NIP=[!/?][*type*],[*ip*],[*netmask*],[*gateway*],[*dns*]<CR>  
 +OK[=*type,ip,netmask,gateway,dns*]<CR><LF><CR><LF>

**Parameter:**

*type* means address type

Table 6-11 address type

| Value | Meanings                                |
|-------|---|
| 0     | Use DHCP to assign IP address dynamicly |
| 1     | assign static IP address                |

*ip* means IP address,data format is such as "192.168.1.22",not including quotation marks

*netmask* means subnet mask, data format is the same as IP address

*gateway* means Gateway address,ata format is the same as IP address

*dns* means DNS address,data format is the same as IP address

## 6.2.4.2 AT+ATM

**Function:**

Set or query work mode of module

**Format:**

AT+ATM=[!/?][*mode*] <CR>  
 +OK[=*mode*]<CR><LF><CR><LF>

**Parameter:**

*mode* means work mode

Table 6-12 work mode

| Value | Meanings       |
|-------|----------------|
| 0     | Auto-work mode |
| 1     | Command mode   |

#### 6.2.4.3 AT+ATRM

##### Function:

Set or query the message of the connection the module creates automatically in auto-work mode.

##### Format:

AT+ATRM=[!/?][protocol],[cs],[host\_timeout],[port]<CR>  
 +OK[=protocol,cs,host,port]<CR><LF><CR><LF>

##### Parameter:

*protocol* means protocol type,

Table 6-13 protocol type

| Value | Meanings |
|-------|----------|
| 0     | TCP      |
| 1     | UDP      |

*cs* means C/S mode,

Table 6-14 CS mode

| Value | Meanings |
|-------|----------|
| 0     | Client   |
| 1     | Server   |

*host\_timeout* means that shown as follow,

Table 6-15 host\_timeout

| cs | protocol | Meanings   |
|----|----------|--|
| 0  | X        | destination server name, you can input domain name or IP address, for example: "192.168.1.100" or "www.sina.com.cn"  |
| 1  | 0        | TCP connection timeout period, the clients that connect with the server send noting in this effective period will be kicked out, effective value range is 1~10000000, its Unit |

|   |   |  |
|---|---|--|
|   |   | is second,0 means never,default setting is 120 seconds |
| 1 | 1 | No meanings  |

*port* means port number

#### 6.2.4.4 AT+SSID

**Function:**

Set or query wireless network name, namely ssid

**Format:**

AT+SSID=[!?!][ssid]<CR>  
 +OK[=ssid]<CR><LF><CR><LF>

**Parameter:**

*ssid* means wireless network name,it's formed with 1~32 characters,and embedded in double quotation marks

#### 6.2.4.5 AT+ENCRY

**Function:**

Set or query wireless network encryption mode.What should be noted is that, every encryption modes need to set correct key to cooperate with AT+KEY command except OPEN mode.

**Format:**

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

**Parameter:**

encry mode means encryption mode

Table 6-16 encryption mode

| Value | Meanings          |
|-------|-------------------|
| 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

##### Function:

Set or query network key. What should be noted is that, before using this command to set network key, user must set the encryption mode with the command AT+ENCRY .

##### Format:

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

##### Parameter:

*format* means key format

Table 6-17 key format

| Value | Meanings |
|-------|----------|
| 0     | HEX      |
| 1     | ASCII    |

*index* means key index, 1~4 is used in wep key, in other encryption modes it's set 0  
*key* means key string, it's embedded in double quotation marks, according with different encryption modes, key size and key format is defined as follows

Table 6-18 key

| Encryption mode    | Key Format                                    |  |
|--------------------|---|--|
|                    | HEX   | ASCII                                  |
| WEP64              | 10 hexadecimal characters <sup>(Note 1)</sup> | 5 ASCII characters <sup>(Note 2)</sup> |
| WEP128             | 26 hexadecimal characters                     | 13 ASCII characters                    |
| WPA-PSK(TKIP)      | 64 hexadecimal characters                     | 8~63 ASCII characters                  |
| WPA-PSK(CCMP/AES)  | 64 hexadecimal characters                     | 8~63 ASCII characters                  |
| WPA2-PSK(TKIP)     | 64 hexadecimal characters                     | 8~63 ASCII characters                  |
| WPA2-PSK(CCMP/AES) | 64 hexadecimal characters                     | 8~63 ASCII characters                  |

Note1: hexadecimal characters means 0~9 and a~f (no division of majuscule and

lowercase ), for example "11223344dd"

Note2: ASCII character means the numbers (0~9) and characters (a~z and A~Z) in ASCII character set which is specified by ISO,for example "14u6E"

#### 6.2.4.7 AT+BSSID

##### Function:

Set or query bssid address of the target AP.The configuration is only effective in infrastructure network

##### Format:

AT+BSSID=[!?][mode],[bssid]<CR>

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

##### Parameter:

*mode* means BSSID Mode

Table 6-19 BSSID Mode

| Value | Meanings  |
|-------|-----------|
| 0     | auto      |
| 1     | specified |

*bssid* means Network BSSID,its size is twelve hexadecimal numbers,format is such as 001EE3A34455

#### 6.2.4.8 AT+CHL

##### Function:

Set or query the specified wireless channel

##### Format:

AT+CHL=[!?][mode],[channel]<CR>

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

##### Parameter:

*mode* means channel mode

Table 6-20 channel mode

| Value | Meanings  |
|-------|-----------|
| 0     | auto      |
| 1     | specified |

*channel* means wireless channel number, effective range is 1~14

## 6.2.4.9 AT+CHLL

**Function:**

Set or query wireless channel list. Wireless channel list is used to prescribe work channel range of module, the channel not included in the channel list will be not scanned, reasonably using this parameter can expedite the speed of scanning and networking of module.

**Format:**

```
AT+CHLL=[!?!][channel list]<CR>
+OK[=channel list]<CR><LF><CR><LF>
```

**Parameter:**

*channel list* means wireless channel list, it uses hexadecimal format, it begins from the lowest bit, each bit means a channel, the default setting is 3fff, that means all of 1~14 channels.

## 6.2.4.10 AT+WPRT

**Function:**

Set or query wireless network type.

**Format:**

```
AT+WPRT=[!?!][type]<CR>
+OK[=type]<CR><LF><CR><LF>
```

**Parameter:**

*type* means network type.

Table 6-21 network type

| Value | meanings               |
|-------|------------------------|
| 0     | Infrastructure network |
| 1     | Adhoc network          |

## 6.2.4.11 AT+WATC

**Function:**

Set or query whether to enable to create adhoc network automatically. This parameter is only effective when wireless network type is set to adhoc, it means whether to create a adhoc network with the same SSID when networking is failed.

**Format:**

```
AT+WATC=[!?!][enable]<CR>
+OK[=enable]<CR><LF><CR><LF>
```

**Parameter**

*enable* means the symbol of enabling.

Table 6-22 symbol of enabling

| Value | Meanings |
|-------|----------|
| 0     | disabled |
| 1     | enabled  |

## 6.2.4.12 AT+WARM

**Function:**

Set or query whether to enable the wireless roam.

**Format:**

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

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

**Parameter:**

*enable* means the symbol of enabling.

Table 6-23 enable symbol

| Value | meanings |
|-------|----------|
| 0     | disabled |
| 1     | enabled  |

## 6.2.4.13 AT+WARC

**Function:**

Set or query the function of auto retry after the wireless network is disconnected or networking is failed. This parameter is only effective in the auto-work mode.

**Format:**

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

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

**Parameter:**

*count* means retry times.

Table 6-24 retry times

| Value | Meanings             |
|-------|----------------------|
| 0     | No retry             |
| 1~254 | 1~254 times of retry |
| 255   | Retry for ever       |

#### 6.2.4.14 AT+WBGR

**Function:**

Set or query BG mode of the wireless network and the maximum sending rate.

**Format:**

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

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

**Parameter:**

*bg mode* mean BG mode

Table 6-25 BG mode

| Value | Meanings  |
|-------|-----------|
| 0     | B/G fixed |
| 1     | B         |

max rate means the maximum sending rate, in B mode, only 0 to 3 is effective

Table 6-26 The maximum sending rate

| Value | Meanings |
|-------|----------|
| 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

**Function:**

Set or query data format of uart serial.

**Format:**

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

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

**Parameter:**

*baud rate* means baud rate, effective range is from 1200 to 115200,

Table 6-27 baud rate

| Value  | Meanings   |
|--------|------------|
| 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* means data bits

Table 6-28 data bits

| Value | Meanings |
|-------|----------|
| 0     | 8 bits   |
| 1     | 7 bits   |

*stop bit* means stop bits

Table 6-29 stop bits

| Value | Meanings      |
|-------|---------------|
| 0     | 1 bit         |
| 1     | Not supported |
| 2     | 2 bits        |

*parity* means parity bit

Table 6-30 Even parity

| Value | Meanings    |
|-------|-------------|
| 0     | No parity   |
| 1     | Odd parity  |
| 2     | Even parity |

#### 6.2.4.16 AT+ATPT

**Function:**

Set or query the auto framing period of data.data is automatically set frame period. This

parameter is only effective when serial is in transparent transmission mode.

**Format:**

```
AT+ATPT=[!?!][period]<CR>
+OK[=period]<CR><LF><CR><LF>
```

**Parameter:**

*period* means auto framing period, its range is 100 ~ 10000ms, and its units is millisecond, the smallest step is 100ms.

#### 6.2.4.17 AT+ATLT

**Function:**

Set or query the data auto framing length of data. This parameter is only effective when serial is in transparent transmission mode.

**Format:**

```
AT+ATLT=[!?!][length]<CR>
+OK[=length]<CR><LF><CR><LF>
```

**Parameter:**

*length* means auto framing length, its range is 64 ~ 1024 bytes.

#### 6.2.4.18 AT+ESPC

**Function:**

Set or query escape characters. This parameter is used for serial to exit the transparent transmission mode.

**Format:**

```
AT+ESPC=[!?!][escape]<CR>
+OK[=escape]<CR><LF><CR><LF>
```

**Parameter:**

*escape* means escape characters,its format is two ASCII characters expressed in hexadecimal,such as ASCII character '+' should be expressed as '2B'.

#### 6.2.4.19 AT+ESPT

**Function:**

Set or query escape time.This parameter is used for serial to exit the transparent transmission mode.

**Format:**

```
AT+ESPT=[!?!][time]<CR>
```

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

**Parameter:**

*time* means escape time, its range is 100 ~ 10000ms, its unit is millisecond, the smallest step is 100ms.

#### 6.2.4.20 AT+WEBS

**Function:**

Set or query whether to enable the internal WEB server.

**Format:**

AT+WEBS=[!?!][*enable*],[*port*]<CR>

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

**Parameter:**

*enable* means the symbol of enabling.

Table 6-31 symbol of enabling

| Value | Meanings |
|-------|----------|
| 0     | disabled |
| 1     | enabled  |

*port* means server port number, the default setting is 80.

#### 6.2.4.21 AT+PASS

**Function:**

Set or query the system password.

**Format:**

AT+PASS=[!?!][*pass*] <CR>

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

**Parameter:**

*pass* means system password, it's formed with 6 ASCII characters.

#### 6.2.4.22 AT+IOM

**Function:**

Set or query GPIO1 mode.

**Format:**

AT+IOM=[!?!][*mode*]<CR>

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

**Parameter:**

*mode* means work mode.

Table 6-32 work mode

| Value | Meanings        |
|-------|-----------------|
| 0     | System function |
| 1     | Input           |
| 2     | Output          |

#### 6.2.4.23 AT+CMDM

##### **Function:**

Set or query default command mode of the system.

##### **Format:**

AT+CMDM=[!/?][*mode*]<CR>

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

##### **Parameter:**

*mode* means the type of command mode.

Table 6-33 Command Mode

| Value | Meaning            |
|-------|--------------------|
| 0     | AT+Instruction     |
| 1     | Compatible potocol |

## 6.3 Sample code

This section chiefly gives the sample codes of the PC for the control of WiFi module in command mode. If the module works in auto-work mode, it will be not need to refer to this section.

### 6.3.1 Description of sample code

The sample code is based on Language C, and it can be explanted to other systems. The sample code mainly gives an demonstration of process for the operation of WiFi module. while explanting, appropriate modification is needed according with own system situation.

### 6.3.2 Operation flow of WiFi module in command mode

Usually the operation flow of WiFi module in command mode is shown as the

following figure. While module is in use, user can decide whether to do the operation of the command line according with the state of system.

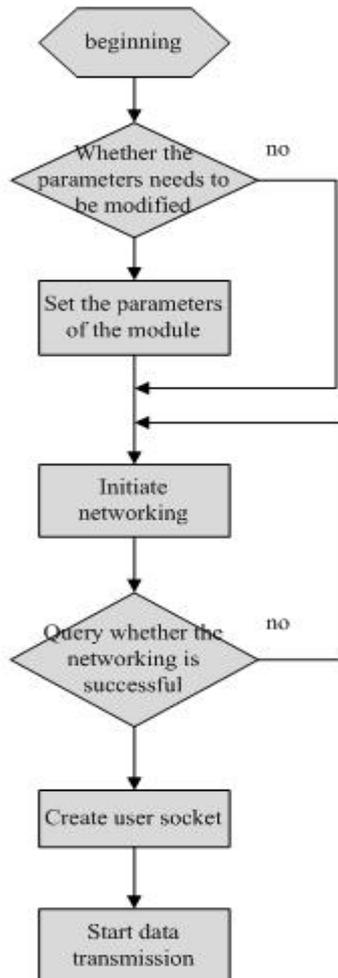


figure6-3 the operation of Wifi module

### 6.3.3 main data framework of sample code

#### 6.3.3.1 At command structure

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

Description,

'CmdName' means code name of the command of AT instruction.

'Flag' means whether the command has return symbol.

'Cmdbuf' means the content the command includes

'Cmdbuiflen' means the length of the command

#### 6.3.3.2 The sending symbol of serial's AT instruction

```
unsigned char AtCmdSending;
```

Description,

the symbol means an AT instruction is being sent or waiting for return. The module can't continue to send down another instruction. If user's system is based on operation system programming, it can be achieved with the mechanism of signal quantum.

#### 6.3.3.3 The symbol for serial waiting return of AT instruction

```
unsigned char ReciveAtCmdrep;
```

Description,

the symbol means the serial is waiting for the return of an AT instruction, if operation system is used, it can be used.

### 6.3.4 Operation of serial data

After the serial has sent a CMD, it needs to wait for return of the command. If there is no return for this command, timeout is needed, the system will force to exit the command after time out. It must be sure that, in the process of sending AT instruction, there is one AT instruction in serial which is waiting for return.

#### 6.3.4.1 The sample of serial sending instruction

```
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;
}

```

Description,

It needs the system supports added function of timeout processing in this sample code. The system will start the timer after serial sends a command, and set *Atcmdtimeout* to 1 in the specified time, in order to achieve the process after the module is time out.

#### 6.3.4.2 The sample of serial receiving the return of AT instruction

It is used to receive the response message of AT instruction, after having sent a AT instruction, the PC needs to wait for the response or timeout (more than 500 ms at least) of WiFi module. This sample is illustrated with the serial interrupt of 51 structure.

##### 1、The function of serial receiving interrupt,

```

void COMM_INT() interrupt 4 using 3
{
    unsigned char  cBuf;
    if(RI)// receive data
    {
        RI = 0;
        cBuf = SBUF;
        ProcessUARTData(cBuf);
    }
    else if (TI)
    {
        TI = 0;
        if(iSendCounter < iUartSendLen)
            SBUF = cpSendBufPtr[iSendCounter++];
    }
}

```

```

    }
}

```

Description.

ProcessUARTData (cBuf) is used to deal with the data the serial receives.

2、The function of data processing,main data structure is described is as follow,

```

#define WIFIBUFLEN  1024
typedef struct WiFiRecBuffer
{
    unsigned char    gcATReceData[WIFIBUFLEN]; //
    unsigned int    gcATReceCount;           ///data length
} WiFiRecBuffer;
struct WiFiRecBuffer  xdata  ATReceBuffer[1];  /// external ram of 51

```

3、Macro definition of AT instruction return status,the definition is mainly used to control the flow of serial receive the return of AT instruction.

```

#define WIFI_WAIT_SYN      0X00    /// serial initialization
#define WIFI_WAIT_RESP    0x01    /// serial waits for
                                   Corresponding character
#define WIFI_RECEDATA     0x02    ///wait for data
#define WIFI_WAIT_OVER    0x03    ///wait for end
#define WIFI_WAIT_ERRCODE 0x04    ///wait for error
#define WIFI_SYN          0X2B    ///receive the character

```

4、The function of parsing the return( ProcessUARTData (cBuf) )

```

void ProcessUARTData( IN unsigned char  UartData)
{
    int i=0;
    switch (gcWifiCommunState)
    {
        case WIFI_WAIT_SYN:           // wait to receive
            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;
}
}

```

Description,

This function is used to deal with the data which is received by the serial,it should be used to judge and deal with when the serial has received a byte of data.The function will set the bit of ReciveAtCmdrep when it has received the return of the AT instruction,at this time ,the bit will be judged in AT instruction sending course,and then the function will parse the different returns of AT instructions.

### 6.3.5 The function used to parse the return parameters of AT instruction

This function is used to parse the return parameters of AT instruction,and split the the return parameters.The sample is as follow.

```

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);
}

```

Description,

*char \*line* points to the data received by the serial, *char \*argv[]* points to the buffer used to save the return parameters, they are used the following parameters parsing.

### 6.3.6 The function of operating character string

The following functions is mainly used to parse the parameters of character string type in the return of AT instruction, and convert the parameters to hexadecimal or decimal which CPU display.

```

INT atodec(char Ch)
{
    INT dec = -1;
    if (Ch >= '0' && Ch <= '9')
        dec = Ch - '0';
    return dec;
}

INT strtodec(INT *dec, char *str)
{
    INT i = -1;
    INT dd = 0, sign = 1;
    if (*str == '-') {
        str++;
        sign = -1;
    }
    while(*str){

```

```
        i = atodec(*str++);
        if (i < 0) {return -1;}
        dd = dd*10 + i;
    }
    *dec = dd*sign;
    return (i<0?-1:0);
}

INT atohex(char Ch)
{
    INT hex = -1;
    if (Ch >= '0' && Ch <= '9')
        hex = Ch - '0';
    else if (Ch >= 'a' && Ch <= 'f')
        hex = Ch - 'a' + 0xa;
    else if (Ch >= 'A' && Ch <= 'F')
        hex = Ch - 'A' + 0xa;
    return hex;
}

INT strtohexarray(INT8U array[], INT cnt, char *str)
{
    INT hex;
    INT8U tmp, *des;
    des = array;
    while(cnt-- > 0)
    {
        hex = atohex(*str++);
        if (hex < 0)
            return -1;
        else{
            tmp = (hex<<4)&0xf0;
        }
        hex = atohex(*str++);
        if (hex < 0)
            return -1;
```

```

    else{
        tmp = tmp | (hex & 0x0f);
    }
    *des++ = (INT8U) tmp;
}
return (*str==0?0:-1);
}

```

### 6.3.7 The parsing sample of several AT instructions sending

#### 6.3.7.1 The sending sample of simple instruction of control type

This command is only used to control the operation of the module, it does not need to send a lot of parameters data in AT instruction, and receiving the response means it has been operated correctly.

1、Take the operation of controlling the WiFi module networking as an example, The data structure used in the operation of WiFi module networking is as follows,

```

#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---ad-hoc
    unsigned char channel;
    unsigned char encry;
    char ssid[32];
    int rssi;
}TJION_RESULT;

```

2、The main function used to control the networking of WiFi module is shown as follows, while using this function, the system will hang out until networking is time out or the module has received the response of the command.

```

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.Cmdbuf1 =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 (strtodec(&(Jionr.Type), argv[1]) < 0)
    {
        return ;
    }
    if(strtodec(&(Jionr.channel), argv[2]) < 0)
    {
        return;
    }
    if(strtodec(&(Jionr.ency), argv[3])<0)
    {
        return;
    }
    strcpy((char *)Jionr.Ssid, argv[4]);
    if(strtodec(&(Jionr.rssi), argv[5])<0)
    {
        return;
    }
}
}
else
{
    ErrorCode=*(arg+2);
    printf("reciver error At cmd resp ERROR Code:%d",ErrorCode);
}
}
}

```

#### 6.3.7.2 The sending sample of the instruction of parameter setting type

the instruction of parameter setting type is mainly used to set the parameters of WiFi module for PC, AT+Instruction set includes many command which is used to set parameters.

1、 Take the key setting as an example,

What used in key setting is mainly data structure,

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

2、 The main function of setting key is as follow,

```
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 The sample of complicated AT+instruction with return parameters

AT+instruction as this type has a response with more parameters, network scan is a typical example.

#### 1、The function of detaching the group of parameters

It is used to divide the long parameters in AT+instruction response into groups.

```

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、 Sample of module sending the scan command

After sending the scan command, WiFi module will return a very long parameters section. The longest section can reach as much as 2Kbytes. therefore we use this part as a example, the AT+instruction used to scan is as follow,

```

void WiFiScan(void)
{
    TATCP_CMD WiFiScan;
    unsigned char ErrorCode=0;
    char *arggroup = NULL;
    char *arg = NULL;
    INT    arggroupc = 0;
    INT    argc = 0;
    int argpoint=0;
    INT err = 0;
    int i=0;
    char *argv[ATCP_MAXARGS] ;
    char *arggroupv[ATCP_MAXARGS] ;
    char * ATCmd="WSCAN\n";
    memset(WiFiJoin,0,sizeof(TATCP_CMD));
    memset(ATReceBuffer[0].gcATReceData,0,sizeof(WiFiRecBuffer));
    WiFiScan.Flag=RespParm;
    WiFiScan.Cmdbuf =ATCmd;
}

```

```

WiFiScan.Cmdbuflen=strlen(ATCmd);
AtCmdSend(WiFiScan);
if(AtRespState==1)
{
    arggroup=strchr(ATReceBuffer[0].gcATReceData, ATC_CH_SET);
    //arg = strchr(ATReceBuffer[0].gcATReceData, ATC_CH_SET);
    if(arggroup)
    {
        {
            arggroupc = parse_line_group(arggroup, arggroupv);
            if (argc <= 0)
            {
                printf("Recive Error parm\n");
                return;
            }
        }
        for(i=0;i<arggroupc;i++)
        {
            argc = parse_line(&arggroupv[arggroupc], argv);
            if (argc <= 0)
            {
                printf("Recive Error parm\n");
                return;
            }
            else
            {
                if (strtohexarray(Scanresult[arggroupc].BSSID, 6, argv[0]) <
0)
                {
                    printf("Error Bssid\n");
                    return ;
                }
                if (strtodec(&(Scanresult[arggroupc].Type), argv[1]) < 0)
                {
                    return ;
                }
            }
        }
    }
}

```

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

## Appendix A Frequent Questions and Answers

**Q1:** Why can not the module search for the specified network?

A: The channel list user set doesn't contain the channel which the specified network work in.

**Q2:** Why can not the module join in the network?

A: Please go through the following reasons,  
Whether the parameters SSID,Key,Channel and so on are consistent with the AP sets or not.  
Whether the module is specified BSSID ,and the BSSID is not consistent with the AP sets.

**Q3:** Why does the module need about 10 seconds to response the command or connect to network automatically, when encryption mode is set to WPA/WPA2?

A: When encryption mode is set to WPA/WPA2,as the key format is ASCII, the module must first do the conversion to the key of ASCII format when it's power-on, this process increases about 10 seconds to the starting time.And When using WPA/WPA2 encryption, directly using PMK of hexadecimal format can eliminate this delay. The specific method is refered in Section 4.3.1.5.

**Q4:** The module has connected to the network successfully, why can't it show "Destination host unreachable" or "Request timed out" after doing the operation ping between the module and PC.

A: Please check the network setting, check whether the IP address and so on is correct or not.

Check whether the encryption mode is consistent with the AP sets or not, because in certain encryption mode, such as WEP encryption with open authentication, even if the key is not correct, module will join in the AP, but they can't communicate.

Maybe certain operating program or Windows Firewall prevents from the operation ping.

**Q5:** In auto-work mode, the ICMP connection sometimes connects and sometimes disconnects after the module has joined in the network, why?

A: The module will create a default network connection after having joined in network, if the TCP server which is the default target of the network connection, is configured unsuccessfully or not opened, the module will disconnect the wireless connection after failed TCP connection, and retry to connect in 10 seconds. And then it appears that ICMP connection sometimes connects and sometimes disconnects. To settle this question is correct configuration of TCP server and opening it.

**Q6:** In auto-work mode, why does the module not connect to the network automatically after resetting?

A: The module forces to go into the configuration mode according the state of MODE pin after start-up, The specific is referred in Section 4.3.1.5.

**Q7:** In auto-work mode,data sent to the module through serial will delay sometime to send to the network,why?

A: In auto-work mode,framing and sending data has two trigger qualification:trigger time and trigger length,if the length of data doesn't reach the trigger length,and it will wait to be framed and sent coercively after trigger time,the specific is refered in Section 4.2.2.2.

**Q8:** In auto-work mode,why does the module not exit the transmission state, after inputing the escape character?

A: To exit transparent transmission has strict limits, the specific is refered in Section 4.2.2.4.

**Q9:** Why can not the configuration program search for the target module?

A: The serial setting of configuration program is not consistent with the serial setting of the module;

The module is in auto-work mode,please reset it , be sure LED1 and LED2 is illumed,and then retry the search.

**Q10:** To modify the parameters with configuration program,why does it show the error 'used prohibitive operating symbol'?

A: Please disconnect the network connection first,and then refer the modified parameters.

**Q11:** Why can not web pages of the WEB server open?

A: The reasons maybe include,

whether the WEB server is forbidden

Whether the server port is set correctly or not, if the port is set to the number except 80, then it's needed to input the port number in the address bar while using a browser, such as <http://192.168.1.100:2000>.

**Q12:** When user sets the serial to the baud rate it does not support, the module can't be connected to, then how to correct it ?

A: Resume it to the manufacturer setting, the approach is as follows,

- 1、 Open the serial debugging software, such as serial master, set the serial parameters to 115200bps/8 bits of data bits/1 bit of stop bits/no parity of parity bit/no flow control.
- 2、 Input 'at+' (no contain quotation marks and contain enter) in the sending column of the software, choose periodic sending, the period is set to 50ms.
- 3、 Reset the module, if it return '+OK', then stop the periodic sending.
- 4、 Send the command 'AT+QVER' to look over the return message, if the message includes B1.xx field, then it joins in bootloader mode successfully, otherwise it's not in, retry from the 2th approach.
- 5、 Send the command 'AT+QVER' (no return).
- 6、 Wait for 1 second, and reset the module, then it can be resume to the manufacturer state.

**Q13:** In Wireless connection state, why can not the specified module be searched for?

A: The reasons includes,

The signal between the netcard and the specified module is weak.

System password of wireless connection setting is not consistent with the specified module.

MAC address in wireless connection setting is specified.

**Q14:** In some environment the module used, why is the quality of wireless connection always bad, and how to improve the wireless transmission?

A: The reasons which cause bad quality of wireless connection are multiple, such as those, the placement of AP is not reasonable, the distance between AP and module is too far, there are some obvious screens or shelter between AP and module, in the environment module works there are some disturbance of 2.4G band and so on. Some environmental problems can't be resolved fundamentally, it can be only reduced for adverse effects with some methods, user can try to do the changes as follow,

- 1、 Lay AP in open and no obvious shelter environment as much as possible, in particular, to avoid large areas of metal and the corners of wall.
- 2、 Modify the setting of AP, change the channel far from the current channel, such as those, change the channel from 1 to 6, or from 6 to 13 and so on.
- 3、 Modify the maximum rate of advanced wireless parameters, reduce it to 11Mbps or even lower.

## Appendix B Revision History of the document

| Version | Revised scope | Date                      |
|---------|---------------|---------------------------|
| 1.00    | Draft         | Sep 10 <sup>th</sup> 2010 |
|         |               |                           |
|         |               |                           |