# WM_W60X_SDK_AT Command User Manual V1.0.3

## Document History

| Version | Completion Date | Revision Record | Author | Auditor | Approver |
|---|---|---|---|---|---|
| V1.0.0 | 2018-03-29 | Initial release | PF Zhang | Kevin | Kevin |
| V1.0.1 | 2018-08-01 | Modify LKSTT2, SSID2, QMAC2 etc. | PF Zhang | Kevin | Kevin |
| V1.0.2 | 2018-10-25 | Modify Onemode Command etc. | Laychin | WM | WM |
| V1.0.3 | 2018-12-15 | Modify W600 to W60X | Cuiych | Ray | |
| | | | | | |
| | | | | | |

# Content

1  Introduction

1.1  Overview

This document is used to introduce the Winner Micro's AT instruction communication protocol of embedded Wi-Fi System on Chip (SoC). AT instructions are command sets based on ASCII commands, which communicate with Wi-Fi chip through UART interface.

1.2  Control protocol

1.2.1  User Command

User commands are used to configure parameters and network transmission functions. The command list is as follows:

| | Instruction name | Function Description |
|---|---|---|
| 1. | (null) | Empty instruction |
| 2. | APCHL | Set/Query wireless channel number under SoftAP mode |
| 3. | APENCRY | Set/Query encryption mode of wireless network under SoftAP mode |
| 4. | APKEY | Set/Query key of wireless network under SoftAP mode |
| 5. | APLKSTT | Query the network connection status under SoftAP mode |
| 6. | APMAC | Query the physical address under SoftAP mode |
| 7. | APNIP | Set/Query local IP address under SoftAP mode |
| 8. | APSSID | Set/Query SSID under SoftAP mode |
| 9. | ATM | Set/Query operation mode of the module |
| 10. | ATRM | Set/Query socket connection message automatically created by the module in automatic operation mode |
| 11. | BSSID | Set/Query BSSID address of target AP |
| 12. | BRDSSID | Enable/Disable AP SSID broadcast |
| 13. | CHL | Set/Query target wireless channel number |
| 14. | CHLL | Set/Query wireless channel list |
| 15. | CNTPARAM | Query network configuration parameters. |

| 16. | CUSTDATA | Receiving user data via one-shot configuration tools. |
|---|---|---|
| 17. | DNS | Set NIC domain |
| 18. | E | Switch to command echo |
| 19. | ENTM | Set the module into the serial transparent transmission mode |
| 20. | ENTS | Set the module into power-saving mode |
| 21. | FWUP | Set URL included firmware's name and start OTA update. |
| 22. | HTTPC | HTTP post or get data function. |
| 23. | KEY | Set/Query key of the wireless network |
| 24. | LKSTT | Query the network connection status(IP,netmask,dns) |
| 25. | NIP | Set/Query local IP address |
| 26. | ONESHOT | Enable/Disable one shot configuration function |
| 27. | ONEMODE | Set one shot configuration mode(UDP, socket or webserver) |
| 28. | PASS | Set/Query system password |
| 29. | PING | Start ping operation |
| 30. | PMTF | Update all parameters from RAM to Flash |
| 31. | PORTM | Set wireless NIC port mode(UART,HSPI) |
| 32. | QMAC | Query MAC address |
| 33. | QVER | Query SDK version information |
| 34. | RSTF | Reset to Factory data |
| 35. | SKCLS | Close socket link |
| 36. | SKCT | Create socket link |
| 37. | SKGHBN | Get server's IP |
| 38. | SKRCV | Receive data through socket |
| 39. | SKRPTM | Enable/Disable the function of active reporting data received by socket |
| 40. | SKSDF | Set default socket connection using to data communication |
| 41. | SKSND | Send data through socket connection |
| 42. | SKSRCIP | Query socket data source IP address |
| 43. | SKSTT | Query status of connected socket |

| 44. | SLIST | Query STAs connected to AP created by module |
|---|---|---|
| 45. | SSID | Set/Query name of SSID |
| 46. | UART | Set/Query serial configuration(baud rate, stop bits, data bits and parity) |
| 47. | WATC | Set/Query creating ADHOC network automatically |
| 48. | WEBS | Set or Query web server's configuration |
| 49. | WJOIN | Join AP or Create AP |
| 50. | WLEAV | Disassociated from wireless network |
| 51. | WPRT | Set/Query type of wireless network |
| 52. | WPSM | Turn on/off power-saving mode |
| 53. | WSCAN | Scan AP |
| 54. | WWPS | Set/query WPS function |
| 55. | Z | Reset system |

## 1.2.2  Error Code

The error code supported by this system is defined as follows, which applies to response message in AT command protocol:

| ASCII | Meaning |
|---|---|
| - | Success |
| -1 | Invalid command format |
| -2 | Command is not supported |
| -3 | Invalid operation symbol |
| -4 | Invalid parameter |
| -5 | Operation not permitted |
| -6 | Lack of memory |
| -7 | Flash error |
| -8 | System busy |
| -10 | Failed to join the network |
| -11 | No available socket |
| -12 | Invalid socket |
| -13 | Socket connection failed |
| -62 | Socket send data failed |

| -63 | Socket receive data failed |
|-----|---------------------------|
| -64 | Undefine error |

## 2    AT Command Protocol

### 2.1    Syntax Introduction

This chipset uses AT+ command protocol as the user control protocol. AT+ command protocol is a command format based on ASCII command style. It's syntax format and management process is described as follows.

### 2.1.1    Syntax Format

■    Format description

All AT+ commands and the parameters are case-insensitive except for the "SSID" and "Key". The parameters are separated by ",".

<>: mandatory, indicates the part must be included at the command

[]:    optional, indicates the part is optional.


■    Command message

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

AT+: Command message prefix

CMD: Command string

[op]: Command operator, when the command requires parameter, it can specify the operation type of parameters, including:

= Parameter/returned value prefix characters;

=! Synchronize the modified parameter to flash in the command of parameter setting type;

=? Query the current setting in the command of parameter setting type.

<CR>: Enter. It is 0x0d in ASCII.

■ Response message

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

+  Response message prefix

RSP   Response string

OK   Success

ERR   Failure

<CR>   Enter. It is 0x0d in ASCII

<LF>   Newline. It is 0x0a in ASCII

■ Data type

String: String, enclosed by double quotation marks, but the content does not include the quotation marks, such as "this is a string"

Dec: Decimal Number, such as 10

Hex: Hexadecimal Number, such as a

Ip: IP address string, such as 192.168.0.1

MAC: 12 consecutive hexadecimal numbers, such as 001EE3A80102

2.1.1.1  Process Flow

AT+ command protocol uses the form of command + response. The majority of commands need the response returned by the receiver to complete the command process. If the previous command is processing and a new command is received again, it will be silently discarded without returning any message, as shown below.
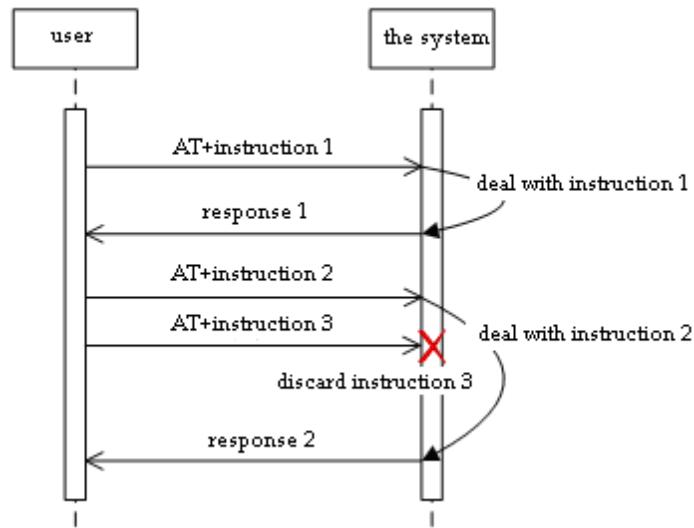
For certain special commands, such as AT+SKSND, AT+SKRCV, binary data shall be transmitted after the command or response, at this time, the data receiver temporarily turns into the transparent transmission mode and starts to receive binary data flow until it receives the data length specified by the <size> field of command or message or waits for timeout, and then it automatically exits transparent transmission status, the process is shown as follows.
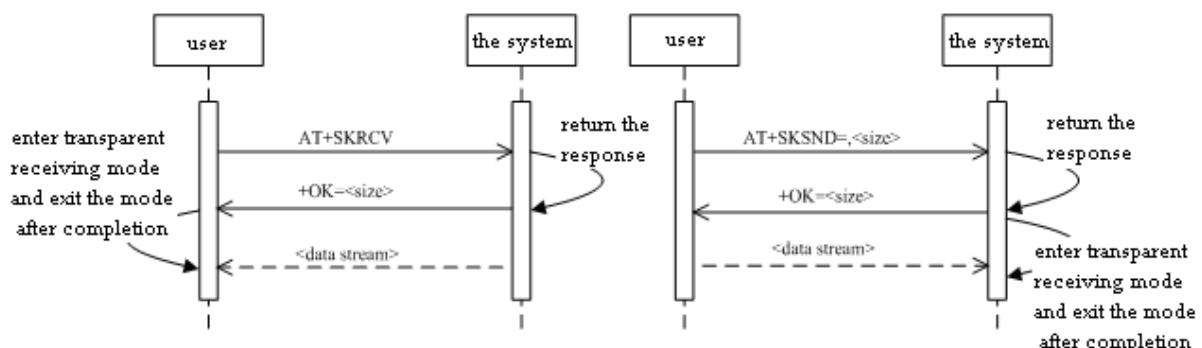


2.1.1.2   Format Examples

Example 1: Successful response message

```
AT+
+OK
```

Example 2: Failure response message

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

Example 3: Command with parameters

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

Example 4: Command with parameter sync to Flash operator <!>

```
AT+SSID=!Test_AP
+OK
```

Example 5: Query Command

```
AT+SSID=?
+OK=Test_AP
```

2.2    Command Sets

2.2.1    System control type

2.2.1.1    AT+

**Function:**

Null command, to confirm whether there is a normal response of the program.

**Format (ASCII):**

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

**Parameter:**

None

### 2.2.1.2 AT+Z

**Function:**

Reset system.

**Format (ASCII):**

```
AT+Z<CR>

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

**Parameter:**

None

### 2.2.1.3 AT+E

**Function:**

Switch to command echo.

**Format (ASCII):**

```
AT+E<CR>

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

**Parameter:**

None

### 2.2.1.4 AT+ENTS

**Function:**

Make system enter in different power-saving mode.

**Format (ASCII):**

```
AT+ENTS=[ps_type],[wake_type],[delay_time],[wake_time]<CR>

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

**Parameter:**

ps_type: power saving type

| Value | Meaning |
|-------|---------|
| 0 | Wi-Fi Sleep |

| 1 | Standby |
|---|---------|

wake_type:

For ps_type = 1

| Value | Meaning |
|-------|---------|
| 0 | gpio |
| 1 | Timer0 |

For ps_type = 0

| Value | Meaning |
|-------|---------|
| 0 | Open Wi-Fi |
| 1 | Close Wi-Fi |

delay_time: delay time, valid only when wakeup source is timer0, unit ms, 100 ～10000ms

wake_time: wake time, valid only when wakeup source is timer0, unit ms, 1000 ～65535ms

## 2.2.1.5 AT+ENTM

**Function:**

Make the module enter the transparent transmission mode. In the transparent transmission mode, the system will exit this mode while receiving escape character in line with the trigger conditions.

Note: Before entering the transparent transmission mode, 1)ensure networking, 2)create a socket, 3)set the default connection of the transparent transmission is the current created socket.

**Format (ASCII):**

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

**Parameter:**

None

## 2.2.1.6 AT+RSTF

**Function:**

Return to factory settings in FLASH. The system needs to restart and then the settings will

become effective.

**Format (ASCII):**

```
AT+RSTF<CR>

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

**Parameter:**

None

### 2.2.1.7   AT+PMTF

**Function:**

All the parameters stored in memory will be updated to FLASH to ensure that USER

defined parameter won't disappear after power down.

**Format (ASCII):**

```
AT+PMTF<CR>

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

**Parameter:**

None

### 2.2.1.8   AT+QMAC

**Function:**

Acquire MAC address of the module.

**Format (ASCII):**

```
AT+QMAC<CR>

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

**Parameter:**

mac address: 12 consecutive hexadecimal numbers, the format is 001EE3A34455

### 2.2.1.9   AT+APMAC

**Function:**

Acquire MAC address of SoftAP, valid in SoftAP and APSTA mode.

**Format(ASCII):**

AT+APMAC<CR>

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

**Parameter:**

mac address：12 consecutive hexadecimal numbers, the format is 001EE3A34455

2.2.1.10 AT+QVER

**Function:**

Acquire the system version information, including hardware version and firmware version.

**Format (ASCII):**

AT+QVER<CR>

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

**Parameter:**

hard: hardware version information, string format, such as "H1.00.00.1029"

firm: firmware version information, string format, such as " F0.02.02@ 18:25:25 Jul 28 2010"

time: time of firmware creation

date: date of firmware creation

2.2.1.11 AT+FWUP

**Function:**

Set URL included firmware's name and start up OTA. Attention, the chipset should have been connected with Server during OTA operation.

**Format(ASCII):**

AT+FWUP=<URL><CR>

Return<URL><CR><LF><CR><LF>

**Parameter:**

URL: The URL included firmware's name during OTA update, such as http://192.168.104.100:80/gnu-linux-W600_SEC.img。

**For example:**

```
AT+FWUP=http://192.168.104.100:80/gnu-linux-W600_SEC.img
```
Location: http://192.168.104.100/gnu-linux-W600_SEC.img

content_length: 318824

download 1460 / 318824

download 2484 / 318824

## 2.2.2 Parameter setting type

### 2.2.2.1 AT+NIP

**Function:**

When the module is in STA mode, this command is used to set/query the local IP address. It should be noted that when DHCP is used, the actual dynamic IP address of the module can't be queried by using this command. AT+LKSTT command can be used to query.

**Format (ASCII):**

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

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

**Parameter:**

type: address type

| Value | Meaning |
|---|---|
| 0 | Use DHCP to dynamically allocate |
| 1 | Use a static IP address |

ip: IP address, data format is "192.168.1.22", without the quotation marks

netmask: subnet mask, data format is in line with IP address

gateway: gateway address, data format is in line with IP address

dns: DNS address, data format is in line with IP address

### 2.2.2.2 AT+APNIP

**Function:**

When the chipset is in SoftAP mode, this command is used to set/query the local IP address and enable/disable DHCP server. When enable DHCP server, this command also can query the IP

address.

**Format(ASCII):**

```
AT+APNIP=[!?][type],[ip],[netmask],[gateway],[dns]<CR>

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

**Parameter:**

type: address type

| Value | Meaning |
|-------|---------|
| 0 | Use DHCP to dynamically allocate/enable DHCP Server |
| 1 | Use a static IP address/disable DHCP Server |

ip: IP address, data format is "192.168.1.22", without the quotation marks

netmask: subnet mask, data format is in line with IP address

gateway: gateway address, data format is in line with IP address

dns: DNS address, data format is in line with IP address

### 2.2.2.3   AT+DNS

**Function:**

Set/query NIC module domain name, the setting is effective only when module is in AP mode.

**Format (ASCII):**

```
AT+DNS=[!?][dnsname]<CR>

+OK[=dnsname]<CR><LF><CR><LF>
```

**Parameter:**

dns name: NIC module domain name, 1~31 characters and surrounded by double quotation marks.

### 2.2.2.4   AT+ATM

**Function:**

Set/query the operating mode of the module.

**Format (ASCII):**

```
AT+ATM=[!?][mode]<CR>

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

**Parameter:**

mode: operating mode

| Value | Meaning |
|-------|---------|
| 0 | Automatic mode |
| 1 | Command mode |

### 2.2.2.5 AT+ATRM

**Function:**

Set/query the socket connection information automatically created by the module in automatic operating mode.

**Format (ASCII):**

AT+ATRM=[!?][protocol],[cs],[host/timeout],[port]<CR>

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

**Parameter:**

protocol: protocol type,

| Value | Meaning |
|-------|---------|
| 0 | TCP |
| 1 | UDP |

cs: C/S mode,

| Value | Meaning |
|-------|---------|
| 0 | Client |
| 1 | Server |

host/timeout: According to the protocol and cs value, the meanings are respectively as follows.

| cs | protocol | Meaning |
|----|----------|---------|
| 0 | X | The target server name, input domain name or ip address, such as "192.168.1.100" or "www.sina.com.cn" |
| 1 | 0 | TCP connection timeout period, the client |

| | | connecting to this server will be automatically kicked off if it does not send any data in this period. The effective value range is 1~10000000, unit: second, 0 means never, the default setting is 120 seconds |
|---|---|---|
| 1 | 1 | No meanings |

port: port number

### 2.2.2.6  AT+SSID

**Function:**

Set/query the wireless network name, that's ssid.

**Format (ASCII):**

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

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

**Parameter:**

ssid: the wireless network name, 1~32 characters surrounded by double quotation marks.

### 2.2.2.7  AT+APSSID

**Function:**

Set/query the softAP name of the module created in AP or APSTA mode, that's ssid.

**Format (ASCII):**

```
AT+APSSID=[!?][ssid]<CR>

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

**Parameter:**

ssid: the wireless network name, 1~32 characters surrounded by double quotation marks.

### 2.2.2.8  AT+APENCRY

**Function:**

Set/query wireless network encryption mode in SoftAP. It should be specially noted that other

encryption modes except OPEN mode need to set correct network key cooperating with AT+KEY

command.

**Format (ASCII):**

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

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

**Parameter:**

encry mode: encryption mode

| Value | Meaning |
|-------|---------|
| 0 | OPEN |
| 1 | WEP64 |
| 2 | WEP128 |
| 3 | WPA-PSK(TKIP) |
| 4 | WPA-PSK(CCMP/AES) |
| 5 | WPA2-PSK(TKIP) |
| 6 | WPA2-PSK(CCMP/AES) |

## 2.2.2.9  AT+KEY

**Function:**

Set/query the network key. It should be noted that before using this command to set network

key, the encryption mode must be firstly set by the command AT+ENCRY.

**Format (ASCII):**

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

+OK[=format,index,key]<CR><LF><CR><LF>

**Parameter:**

format: key format

| Value | Meaning |
|-------|---------|
| 0 | HEX |
| 1 | ASCII |

index: The key index number, 1~4 is used in WEP encryption key, and it's set 0 in other encryption

mode.

key: Key string, it's surrounded by double quotation marks. According to different encryption modes, key size and key format are defined as follows,

| Encryption mode | Key format | |
|---|---|---|
| | HEX | ASCII |
| WEP64 | 10 hexadecimal characters (Note 1) | 5 ASCII characters (Note 2) |
| 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 |

Note 1: 16 hexadecimal characters represents 0 ~ 9, a ~ f (case insensitive), such as "11223344dd"

Note 2: ASCII character refers to numbers 0~9 and characters a ~ z (case-sensitive) in the standard ASCII character set specified by the International Organization for Standardization (ISO), such as "14u6E"

### 2.2.2.10 AT+APKEY

**Function:**

Set/query the network key in SoftAP mode. It should be noted that before using this command to set network key, the encryption mode must be firstly set by the command AT+ENCRY.

**Format(ASCII):**

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

**Parameter:**

format: key format

| Value | Meaning |
|---|---|
| 0 | HEX |
| 1 | ASCII |

index: The key index number, 1~4 is used in WEP encryption key, and it's set 0 in other encryption mode.

key: Key string, it's surrounded in double quotation marks. According to different encryption

modes, key size and key format are defined as follows,

| Encryption mode | Key format | |
| --- | --- | --- |
| | HEX | ASCII |
| WEP64 | 10 hexadecimal characters [Note 1] | 5 ASCII characters [Note 2] |
| 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 |

Note 1: 16 hexadecimal characters represents 0 ~ 9, a ~ f (case insensitive), such as "11223344dd"

Note 2: ASCII character refers to numbers 0~9 and characters a ~ z (case-sensitive) in the standard ASCII character set specified by the International Organization for Standardization (ISO), such as "14u6E"

### 2.2.2.11 AT+BSSID

**Function:**

Set/query specified BSSID address of the target AP. This setting is effective only in the infrastructure network when the module is in STA mode.

**Format (ASCII):**

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

**Parameter:**

mode: BSSID mode

| Value | Meaning |
| --- | --- |
| 0 | Automatic |
| 1 | Specified |

bssid: network BSSID, the length is 12 hexadecimal numbers, the format is 001EE3A34455

### 2.2.2.12 AT+CHL

**Function:**

Set/query the specified wireless channel mode.

**Format (ASCII):**

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

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

**Parameter:**

mode: channel mode

| Value | Meaning |
|---|---|
| 0 | Automatic |
| 1 | Specified |

channel: wireless channel number, the effective range is 1 ~ 14.

### 2.2.2.13 AT+APCHL

**Function:**

Set/query specified wireless channel mode in SoftAP mode.

**Format(ASCII):**

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

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

**Parameter:**

mode：channel mode

| Value | Meaning |
|---|---|
| 0 | Automatic |
| 1 | Specified |

channel: wireless channel number, the effective range is 1 ~ 14.

### 2.2.2.14 AT+CHLL

**Function:**

Set/query wireless channel list. The parameter in wireless channel list is used to specify working channel range of the module, channels not in the list will not be scanned. Reasonably using this parameter may speed up the module scanning and networking speed.

**Format (ASCII):**

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

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

**Parameter:**

channel list: wireless channel list, in hexadecimal format, beginning from the lowest bit, each

bit represents a channel, the default setting is 3fff, that represents all the 1~14 channels

2.2.2.15 AT+WPRT

**Function:**

Set/query wireless network mode.

**Format (ASCII):**

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

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

**Parameter:**

type: network mode

| Value | Meaning |
|-------|---------|
| 0 | infra network (the module as STA) |
| 1 | adhoc network (**not supported currently**) |
| 2 | infra network (the module as AP) |
| 3 | infra network (the module as APSTA) |

2.2.2.16 AT+WATC

**Function:**

Set/query whether to automatically create the enabled adhoc network. This setting is effective

only when the wireless network type is set to adhoc, which means whether to create an adhoc

network with the same SSID when networking is failed.

**Format (ASCII):**

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

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

**Parameter:**

enable: The symbol of enabling

| Value | Meaning |
|-------|---------|
| 0 | Disabled |
| 1 | Enabled |

## 2.2.2.17 AT+WPSM

**Function:**

Turn on/off automatic power-saving mode after module connects to AP.

**Format (ASCII):**

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

**Parameter:**

enable:

| Value | Meaning |
|-------|---------|
| 0 | Disabled |
| 1 | Enabled |

## 2.2.2.18 AT+BRDSSID

**Function:**

Enable/disable AP SSID broadcast, and this setting is only valid when the module is in AP mode.

**Format (ASCII):**

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

**Parameter:**

enable: Whether to enable SSID broadcast or not

| Value | Meaning |
|-------|---------|
| 0 | Disable AP SSID broadcast |

| 1 | Enable AP SSID broadcast |
|---|---|

### 2.2.2.19 AT+UART

**Function:**

Set/query UART interface parameter (only effective in UART1).

**Format (ASCII):**

AT+UART=[!?][baudrate],[databit],[stopbit],[parity],[flowcontrol]<CR>

+OK[=baudrate,databit,stopbit,parity]<CR><LF><CR><LF>

**Parameter:**

baud rate: baud rate, the valid range is from 1200～115200，

| Value | Meaning |
|---|---|
| 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: data bits

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

stop bit: stop bits

| Value | Meaning |
|---|---|
| 0 | 1 bit |
| 1 | Not support |
| 2 | 2 bits |

parity: parity

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

## 2.2.2.20 AT+WEBS

**Function:**

Set/query whether to enable WEB management server.

**Format (ASCII):**

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

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

**Parameter:**

enable:

| Value | Meaning |
|-------|---------|
| 0 | Disabled |
| 1 | Enabled |

port: server port number, the default setting is 80.

## 2.2.2.21 AT+PASS

**Function:**

Set/query system login password for webserver.

**Format (ASCII):**

AT+PASS=[!?][password]<CR>

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

**Parameter:**

pass: 6 ASCII characters

2.2.2.22 AT+CUSTDATA

**Function:**

Receiving user data via one-shot configuration tool.

**Format(ASCII):**

AT+CUSTDATA[=?]<CR>

+OK=data<CR><LF><CR><LF>

**Parameter:**

None

2.2.2.23 AT+CNTPARAM

**Function:**

Query network configuration parameters.

**Format (ASCII):**

AT+CNTPARAM[=?]<CR>

+OK=<bssid_en>,[bssid],[ssid],<key><CR><LF><CR><LF>

**Parameter:**

return：bssid_en

| Value | Meaning |
|-------|---------|
| 0 | Disable BSSID connecting |
| 1 | Enable BSSID connecting |

Enable BSSID,return 1 , bssid, key

Disable BSSID,return 0, ssid, key

2.2.3　Network control class

2.2.3.1　AT+PORTM

**Function:**

Set/query the module interface mode.

**Format (ASCII):**

AT+PORTM=[!?][mode]<CR>

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

**Parameter:**

mode: interface mode

| Value | Meaning |
|-------|---------|
| 0 | Low-speed UART |
| 1 | High-speed UART |
| 2 | H-SPI |

### 2.2.3.2 AT+SKGHBN

**Function:**

Get IP of the extranet domain name such as www.xxx.com.

**Format (ASCII):**

AT+ SKGHBN =[!?][ URL] <CR>

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

**Parameter:**

URL: extranet domain name

IP: ip address.

### 2.2.3.3 AT+WWPS

**Function:**

Set/query WPS function

**Format (ASCII):**

AT+WWPS=[!?][mode],[pin]<CR>

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

**Parameter:**

mode: WPS function selection

| Value | Meaning |
|-------|---------|
| 0 | Button method |

| 1 | PIN method |
|---|---|

pin_len: the length of PIN code (only for HEX format command)

pin: wireless router PIN code, hexadecimal format

### 2.2.3.4 AT+WJOIN

**Function:**

This command is only valid when the module joins wireless network as STA or create a wireless network as access point (AP). If the current network mode is adhoc, and no specified SSID network can be detected, the module will automatically create a new network. If it is already networking, then the network connection information will be directly returned.

**Format (ASCII):**

```
AT+WJOIN<CR>
+OK=<bssid>,<type>,<channel>,<encry>,<ssid>,<rssi><CR><LF><CR><LF>
```

**Parameter:**

bssid: Network BSSID with the length of 12 hexadecimal numbers in the format of 001EE3A34455

type: network mode

| Value | Meaning |
|-------|---------|
| 0 | Infra network(STA) |
| 1 | Adhoc network (No Used currently) |
| 2 | infra network (AP) |

channel: channel number

encry: encryption mode

| Value | Meaning |
|-------|---------|
| 0 | Open |
| 1 | Encryption |

ssid: wireless network name, 1~32 characters, surrounded by double quotation marks

rssi: network signal strength, none minus, its unit is Db, i.e., 50 indicates that the strength is -50Db.

2.2.3.5   AT+WLEAV

**Function:**

This command is only valid when the module is in STA mode, and it can be used to disconnect the current wireless network.

**Format (ASCII):**

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

**Parameter:**

None

Note: When in SoftAP or APSTA mode, the AT command is AT+WLEAVE=2 to destroy the SoftAP.

2.2.3.6   AT+WSCAN

**Function:**

This command is valid only when the module is in STA mode, and it can be used to scan wireless network and return the access point's list after scan completion.

**Format (ASCII):**

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

**Parameter:**

The same with AT+WJOIN

2.2.3.7   AT+LKSTT

**Function:**

Query the network connection status.

**Format (ASCII):**

AT+LKSTT<CR>

+OK[=status,ip,netmask,gateway,dns1,dns2]<CR><LF><CR><LF>

**Parameter:**

status: connection status

| Value | Meaning |
|-------|---------|
| 0 | Disconnect |
| 1 | Connect |

ip: ip address, data format is "192.168.1.22", without the quotation marks

netmask: subnet mask, data format is in line with ip address

gateway: gateway address, data format is in line with ip address

dns1: DNS address, data format is in line with ip address

dns2: DNS address, data format is in line with ip address

### 2.2.3.8  AT+APLKSTT

**Function:**

Query the soft ap network connection status of the module created in SoftAP or APSTA mode.

**Format(ASCII):**

AT+APLKSTT<CR>

+OK[=status,ip,netmask,gateway,dns1,dns2]<CR><LF><CR><LF>

**Parameter:**

status: connection status

| Value | Meaning |
|-------|---------|
| 0 | Disconnect |
| 1 | Connect |

ip: ip address, data format is "192.168.1.22", without the quotation marks

netmask: subnet mask, data format is in line with ip address

gateway: gateway address, data format is in line with ip address

dns1: DNS address, data format is in line with ip address

dns2: DNS address, data format is in line with ip address

2.2.3.9 AT+SLIST

**Function:**

Query the STA information already connected. Valid only in ap or apsta mode.

**Format (ASCII):**

AT+SLIST<CR>

+OK[=sta_number,sta_mac1,sta_ip1,sta_mac2,sta_ip2…]<CR><LF><CR><LF>

**Parameter:**

sta_number: the number of STA already connected.

sta_mac: STA MAC address already connected

sta_ip: STA IP address already connected

2.2.3.10 AT+SKCT

**Function:**

Create the socket. In client mode, waiting for connection completion (success or failure) and then return; in server mode, directly return after the completion of creation.

**Format (ASCII):**

AT+SKCT=[protocol],[cs],[host_timeout],<remote_port>,<local_port><CR>

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

**Parameter:**

protocol: protocol type,

| Value | Meaning |
|-------|---------|
| 0 | TCP |
| 1 | UDP |

cs：C/S mode,

| Value | Meaning |
|-------|---------|
| 0 | Client |
| 1 | Server |

host/timeout: According to the protocol and cs, the meanings are respectively as follows.

| cs | protocol | Meaning |
|----|----------|---------|

| 0 | X | The target server name, input domain name or ip address, such as "192.168.1.100" or "www.sina.com.cn" |
|---|---|---|
| 1 | 0 | TCP connection timeout period, the client connecting to this server will be automatically kicked off if it does not send any data in this period. The effective value range is1~10000000, unit: second, 0 means never, the default setting is 120 seconds |
| 1 | 1 | No meanings |

remote_port: remote port number

local_port: local port number

socket: socket number

### 2.2.3.11 AT+SKSND

**Function:**

Send data by the specified socket and return after the completion. This command sends binary data. Users should begin to send original data after receiving the response (+OK) of module. The module will automatically exit transmission process after receiving the specified data length and send data to the network, the redundant data will be rejected. Otherwise, the module will be forced to constrainedly exit transmission mode and send the received data to the network when the module has been waiting for timeout period (1 second).

**Format (ASCII):**

AT+SKSND=<socket>,<size><CR>

+OK=<actualsize><CR><LF><CR><LF>

[data steam]

**Parameter:**

socket: socket number

size: the prepared sending data length, the number of bytes

actualsize: the allowed sending data length, the number of bytes

data steam: original data

### 2.2.3.12 AT+SKRCV

**Function:**

Read data form receive buffer of the target socket, and return after the completion. After receiving this command, the module will transmit the specified size binary data and respond to the correspondent message (+OK) after finishing.

**Format (ASCII):**

```
AT+SKRCV=<socket>,<maxsize><CR>

+OK=<size><CR><LF><CR><LF>

[data steam]
```

**Parameter:**

socket: socket number

maxsize: the maximum length of the receivable data

size: the length of the actual received data

data steam: original data

### 2.2.3.13 AT+SKSTT

**Function:**

Get the specified socket status, the first line of the return value means user-specified socket status information. If the socket type is TCP server, then from the next line, each line represents a socket status of a connected client.

**Format (ASCII):**

```
AT+SKSTT=<socket><CR>

+OK=<socket>,<status>,[host],[HostPort],[LocalPort],[rx_data]<CR><LF>

    [socket],[status],[host],[HostPort],[ LocalPort],[rx_data]<CR><LF>

    ...

    <CR><LF>
```

**Parameter:**

socket：socket number

status：socket status

| Value | Meaning |
|-------|---------|
| 0 | Disconnection |
| 1 | Detecting |
| 2 | Connection |

host: IP address of the opposite terminal

HostPort: port number of the opposite terminal

LocalPort: the local port number

rx_data: data size in the receiving buffer

## 2.2.3.14 AT+SKCLS

**Function:**

Close the specified socket.

**Format (ASCII):**

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

**Parameter:**

socket: socket number

## 2.2.3.15 AT+SKSDF

**Function:**

Set the default sending socket of the system. If users need to enter the transparent transmission mode in command mode, this command can be used to specify the destination that transparent serial data transfers to or receives from.

**Format (ASCII):**

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

**Parameter:**

socket: socket number

### 2.2.3.16 AT+SKSRCIP

**Function:**

Get current socket receiving data source IP address.

**Format (ASCII):**

```
AT+SKSRCIP=?<CR>

+OK=[host]<CR><LF><CR><LF>
```

**Parameter:**

Host: IP address of data source

### 2.2.3.17 AT+SKRPTM

**Function:**

Turn on/off socket active reporting and receiving data function.

**Format (ASCII):**

```
AT+SKRPTM=<mode><CR>

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

**Parameter:**

Mode：

| Value | Meaning |
|-------|---------|
| 0 | Close |
| 1 | Open |

### 2.2.3.18 AT+ONEMODE

**Function:**

Set one-shot config mode.

**Format(ASCII):**

```
AT+ONEMODE=[!?][mode]<CR>

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

**Parameter:**

Mode:

| Value | Meaning |
|-------|---------|
| 0 | One-shot config mode |
| 1 | SoftAP config mode |
| 2 | Web config mode |

2.2.3.19 AT+ONESHOT

**Function:**

Turn on/off one-shot configuration function.

Format (ASCII):

```
AT+ONESHOT=<status><CR>

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

**Parameter:**

status:

| Value | Meaning |
|-------|---------|
| 0 | Close |
| 1 | Open |

2.2.3.20 AT+HTTPC

**Function:**

Http post/get function.

Format (ASCII):

```
AT+HTTPC=<url>,<verb>,<postdata><CR>

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

**Parameter:**

url: remote http server-side address;

verb:

| Value | Meaning |
|-------|---------|

| 0 | http get |
|---|---|
| 1 | - |
| 2 | http post |
| 3 | http put |

post data:

Upload data to http when verb is 2, 3.

2.2.3.21 AT+PING

**Function:**

Start to PING.

**Format(ASCII):**

AT+PING=<host,interval,count,operate><CR>

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

**Parameter:**

Host: The IP or domain's name on the other end. The IP address is dotted decimal format.

Interval: ping packet interval, in milliseconds

Count: ping packet count, 0 means continuous PING

Operate: 1 means start ping, 0 means stop ping

3    Common Operations

3.1    Create SoftAP process

（1）WPRT    Set wireless NIC operating mode as AP

AT+WPRT=2

（2）SSID    Set wireless NIC network name of STA as MyAp

AT+APSSID="MyAp"

（3）ENCRY    Set wireless NIC encryption mode as WEP64

AT+ENCRY=1

Parameter:open：0，WEP64：1，WEP128：2

（4）KEY    Set wireless NIC key as 12345

AT+KEY=1,1,"12345"

Parameter 1: key format, 0 represents HEX, 1 represents ASCII

Parameter 2: index: key index number, 1～4 for WEP encryption key, other encryption method is fixed at 0

Parameter 3: wireless key. For example: 12345

（5）APNIP                Set IP address and subnet mask

AT+APNIP=1,"192.168.1.1","255.255.255.0","192.168.1.1","192.168.1.1"

Parameter 1: address type, 0 refers to using DHCP dynamic allocation, 1 refers to the static address

Parameter 2: IP: 192.168.1.1

Parameter 3: netmask: 255.255.255.0

Parameter 4: gateway: 192.168.1.1

Parameter 5: DNS: 192.168.1.1

（6）PMTF      Save parameters to flash

AT+PMTF

（7）Z            Reset wireless NIC

AT+Z

（8）Delay 1 second

（9）WJOIN        Create wireless network MyAp

AT+WJOIN

（10）SLIST         Query connected STAs' information

AT+SLIST

## 3.2　Scan AP process

AT cmd for wireless NIC scanning AP: AT+WSCAN

## 3.3　STA joins AP process

（1）WPRT      Set operating mode as STA

AT+WPRT=0

（2）SSID　　　　Set the target AP name , such as WinnerMicro

AT+SSID="WinnerMicro"

（3）KEY　　　　　Set the target AP wireless key 12345678

AT+KEY=1,0,"12345678"

Parameter 1: key format, 0 represents HEX, 1 represents ASCII

Parameter 2: index: key index number, 1～4 for WEP encryption key, other encryption method is fixed at 0

Parameter 3: wireless key. For example: 12345678

（4）NIP　　　　　Start DHCP

AT+NIP=0

（5）PMTF　　　　Save parameters to flash

AT+PMTF

（6）Z　　　　　　Reset wireless NIC

AT+Z

（7）Delay 1 second

（8）WJOIN　　　　　Join wireless network WinnerMicro

AT+WJOIN

3.4　Create APSTA mode process

（1）WPRT　　　　Set operating mode as APSTA

AT+WPRT=3

（2）SSID　　　　　Set the target AP's name to be connected, such as WinnerMicro

AT+SSID="WinnerMicro"

（3）KEY　　　　Set the target AP's key number, such as 12345678

AT+KEY=1,0,"12345678"

Parameter 1: key format, 0 means HEX, 1 means ASCII

Parameter 2: index: key's reference number, 1~4 used for WEP encryption, 0 for other encryptions.

Parameter 3: wireless key, such as 12345678

（4）APSSID　　　Create SoftAP's name, such as MYSoftAP

AT+APSSID="MYSoftAP"

（5）APENCRY　　　Set encryption type by SoftAP, such as WPA2-TKIP

AT+APENCRY=5

（6）APKEY　　　Set the key used by SoftAP, such as 87654321 in ASCII

AT+APKEY=1,0,"87654321"

（7）APNIP　　　Set IP address and subnet mask

AT+APNIP=1,"192.168.1.1","255.255.255.0","192.168.1.1","192.168.1.1"

（8）PMTF　　　Save parameters to flash

AT+PMTF

（9）Z　　　Reset

AT+Z

（10）Delay 1 second

（11）WJOIN　　　Join the wireless network WinnerMicro and create softAP

AT+WJOIN

## 3.5　How to disconnect AP by STA

AT cmd for wireless NIC disconnecting AP: AT+WLEAV

## 3.6　How to destroy SoftAP

AT cmd for destroying SoftAP: AT+WLEAV=2

## 3.7　How to check the current status by STA

AT cmd for wireless NIC querying current NIC status: AT+LKSTT

### 3.8    How to check the current status by SoftAP

AT cmd for view the status by SoftAP mode: AT+APLKSTT

### 3.9    How to use socket function

（1）Build a PC-side TCP server. If using TCP debugging assistant, TCP server address is 192.168.253.1, listening port 1000.
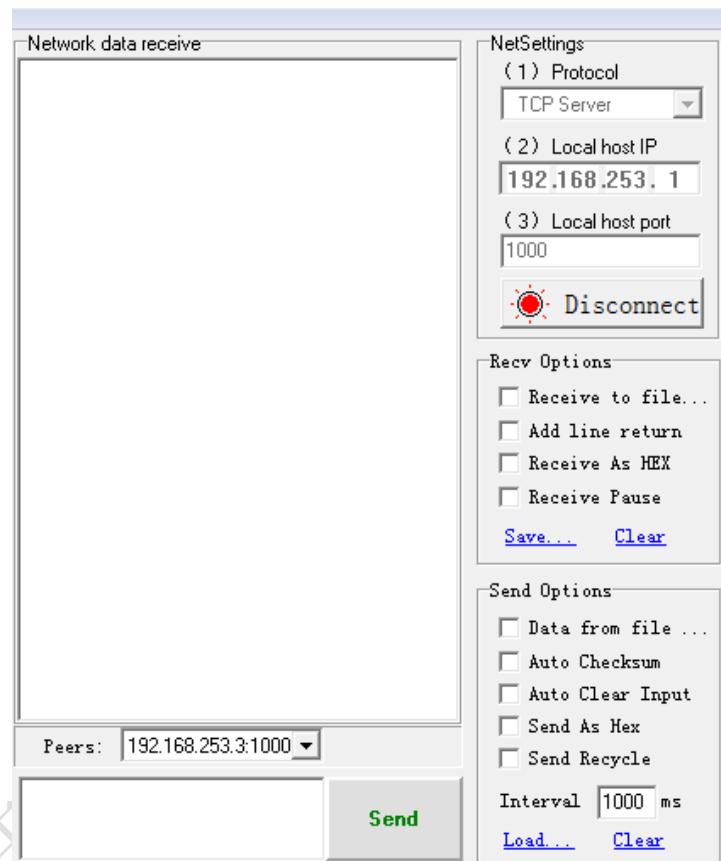


（2）Set up Socket:

      TX:     AT+SKCT=0,0,"192.168.253.1",1000,1000

      RX:     +OK=1      ---> 1 is socket number
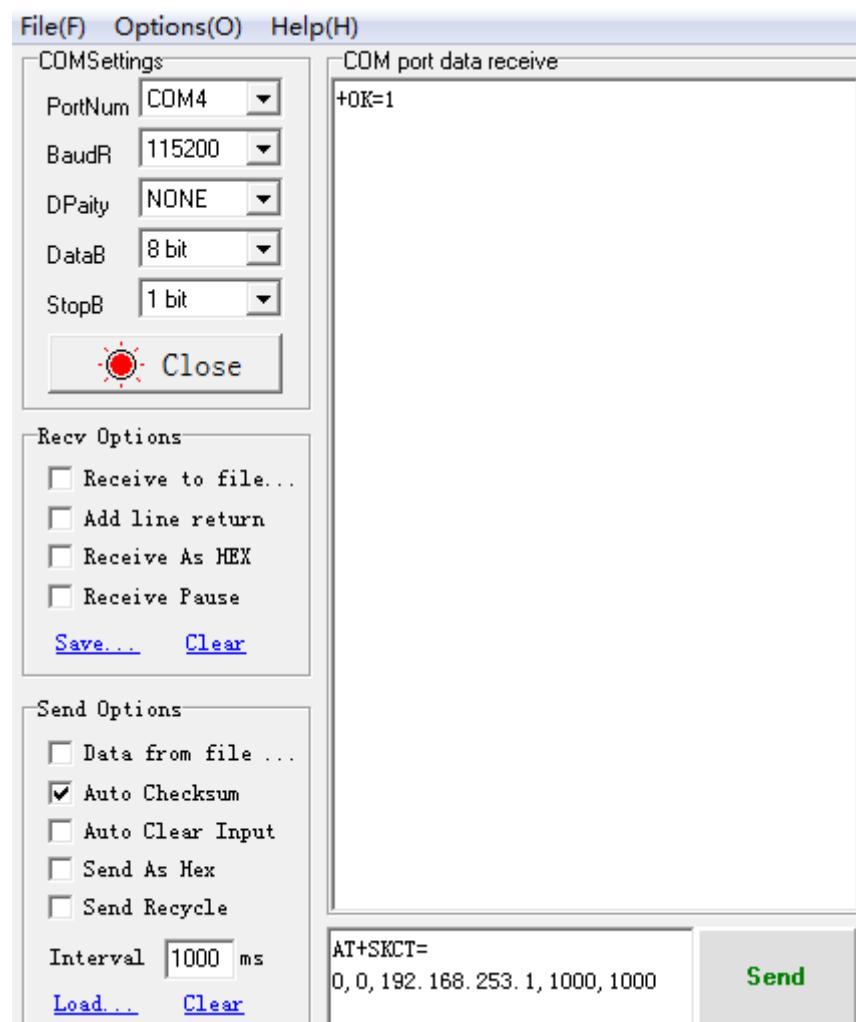
（3）Sending data:

TX:     AT+SKSND=1,5

        kevin

RX：    +OK=5
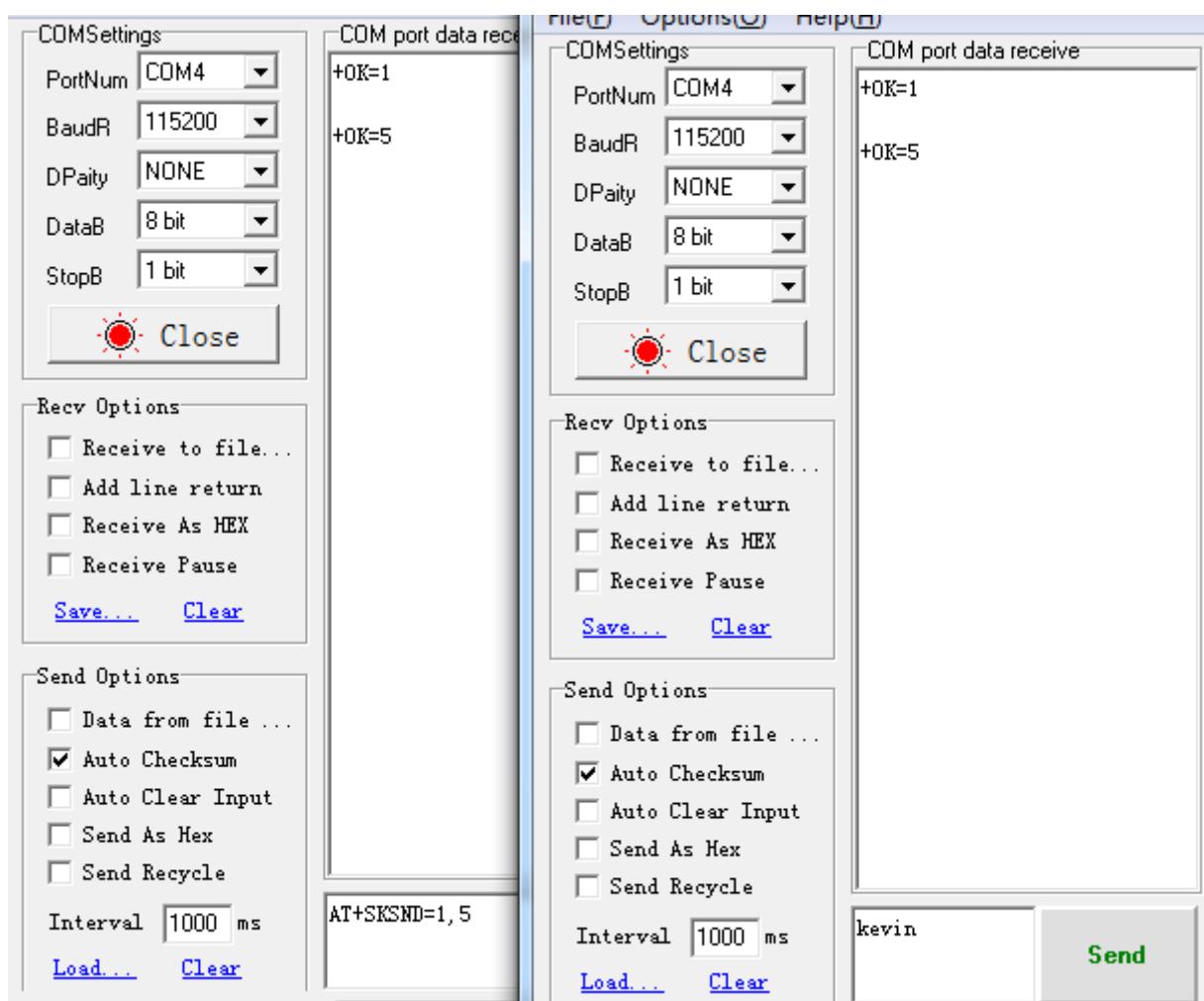
The data interface received by TCP server is as follows:
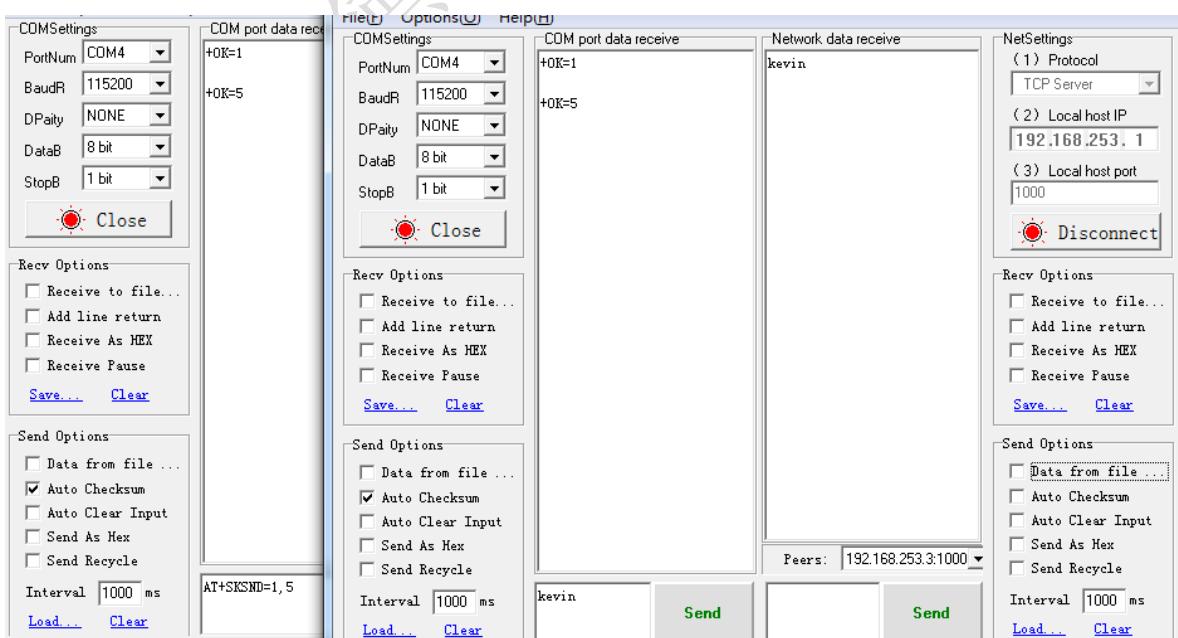


（4）Receiving data:

Input sending data *hello* in the TCP debugging assistant interface, and click *Sending*.
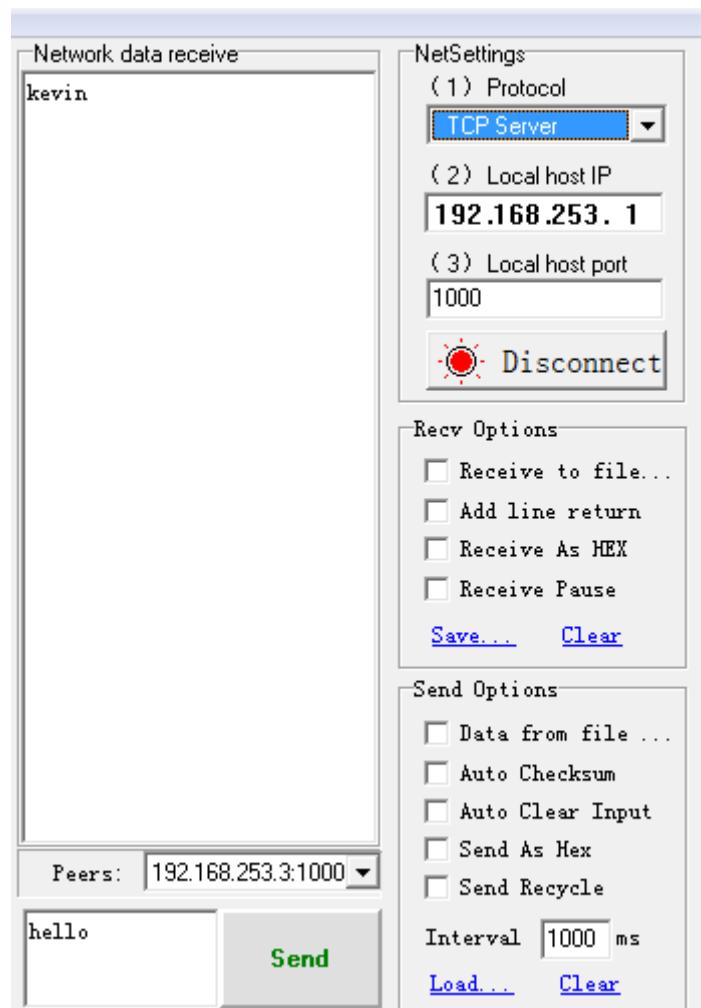


TX： AT+SKRCV=1,5

RX： +OK=5

hello
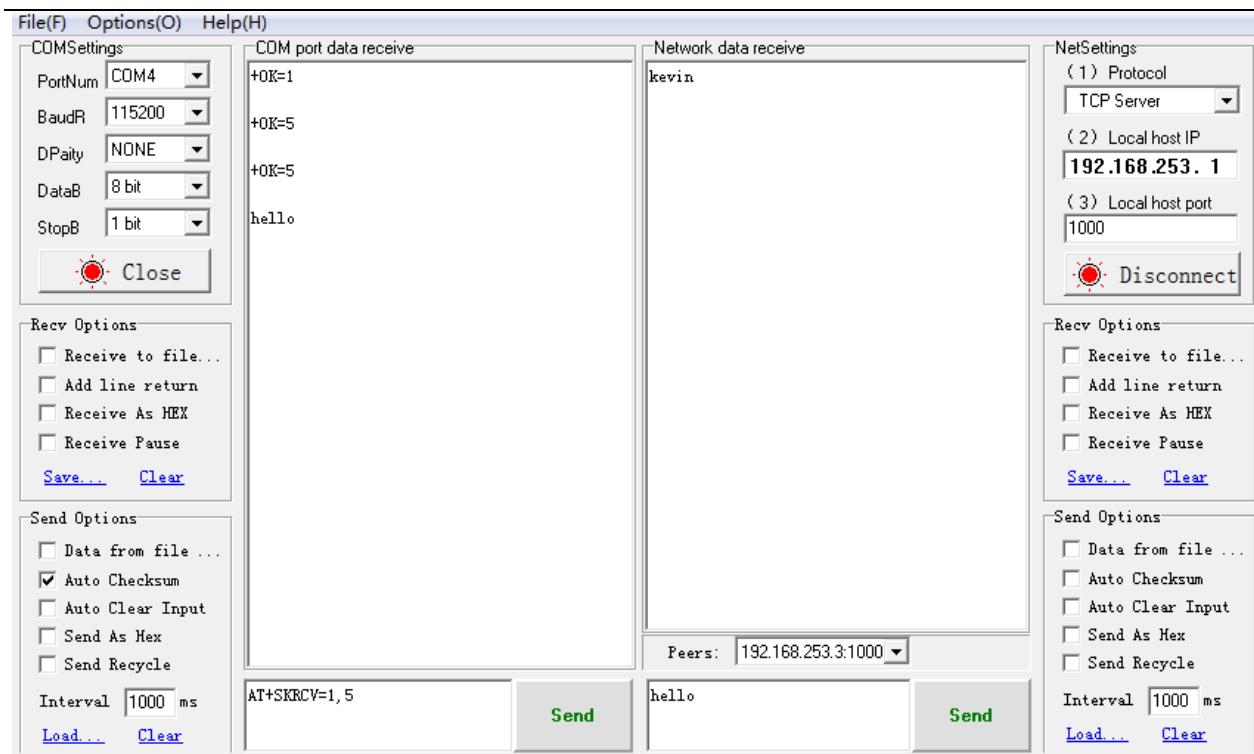
（5）Query Socket status:

TX： AT+SKSTT=1

RX： +OK=1,2,"192.168.253.1",1000,1000,0
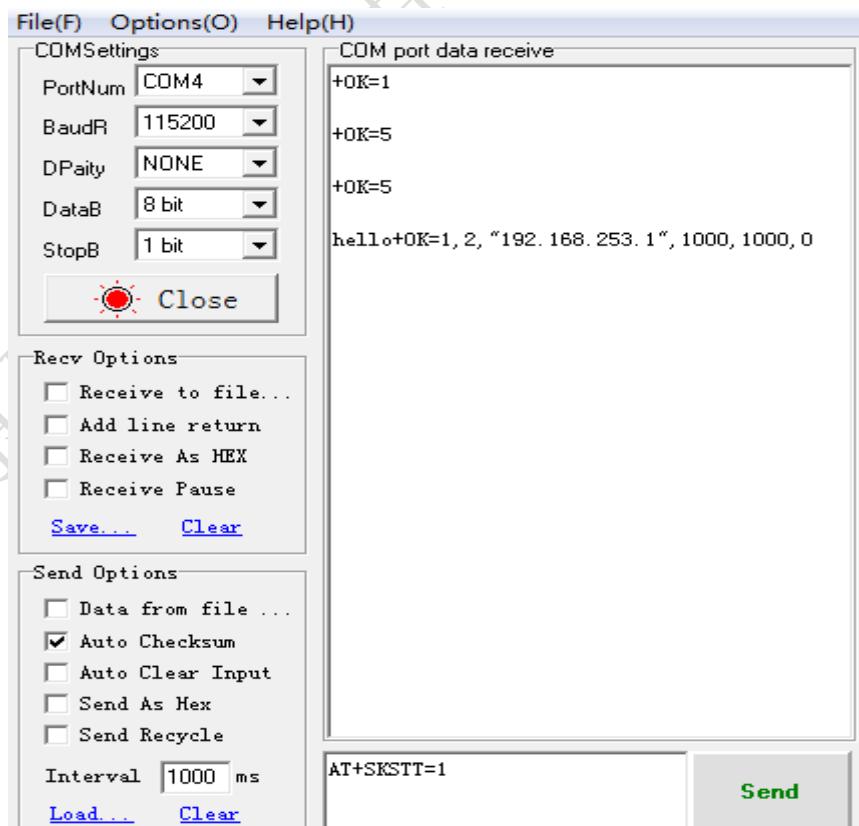


（6）Close Socket connection

TX： AT+SKCLS=1

RX： +OK