RAK476 UART-WIFI Module

Programming Manual V1.2

Shenzhen Rakwireless Technology Co., Ltd
www.rakwireless.com info@rakwireless.com

Content

1. Introduction

1.1 Brief introduction

This document mainly gives a detailed introduction to the AT command of RAK476, and provides simple, fast and efficient methods for the user to independently utilize WiFi module.

1.2 Outline

RAK476 is a Wi-Fi module fully conforming to the 802.11b/g /n wireless protocol, with an internal integration of complete TCP/IP protocol stack and containing ARM-CM3 MCU, WLAN MAC and a WLAN baseband of single current. The module includes onboard antenna, external antenna interface, and radio frequency output pin. With the integration of TCP/IP protocol stack, various protocols are supported, such as ARP, IP, ICMP, TCP, UDP, CLIENT DHCP, SERVER DHCP, DNS and so forth. It also supports the modes of AP and Station. RAK476 provides plenty of AT commands for a variety of applications and the user can easily and quickly implement netting and data transmission and receiving through the module. The module' s baud rate of serial interface supports a maximum of 921600bps, completely meeting the application of low-bitrate.

In terms of the network, RAK476 supports parameter preservation and the customer can command whether to start networking automatically to achieve one-step net connection and reduce the netting time of the system. Network parameter of wireless configuration module as well as firmware of wireless upgrade module are supported. Moreover, Easyconfig one-key-network mode are supported. In all, the difficulty of software development is greatly reduced.

RAK476 offers lots of configurable GPIO, which can be configured as SPI, UART, 12C to be applied in various applications and controls. The interior of RAK476 also includes 512KB of SRAM and 1MB of flash to achieve rich WIFI functions.

2. AT command

2.1 Serial port setting

The module starts up by the default UART parameter, the client host needs to configure the same parameter to guarantee normal communication, and automatic baud rate is not supported for the present. The UART default parameter is as follows:

Baud rate ----- 115200

Data bit ----- 8

Stop bit ----- 1

Parity check bit ----- NULL

Flow control ----- Disable

RAK476 module supports the following baud rates:

9600 bps

19200 bps

38400 bps

57600 bps

115200 bps

230400 bps

460800 bps

921600 bps

Note:

1. The maximum data length of single packet of UART interface communicating: sending-receiving are both 1,400Byte, mqtt is 1,200Byte.

2. The UART configuration parameter after modification is valid under the current status, and starts up by the default parameter 115200 after reset

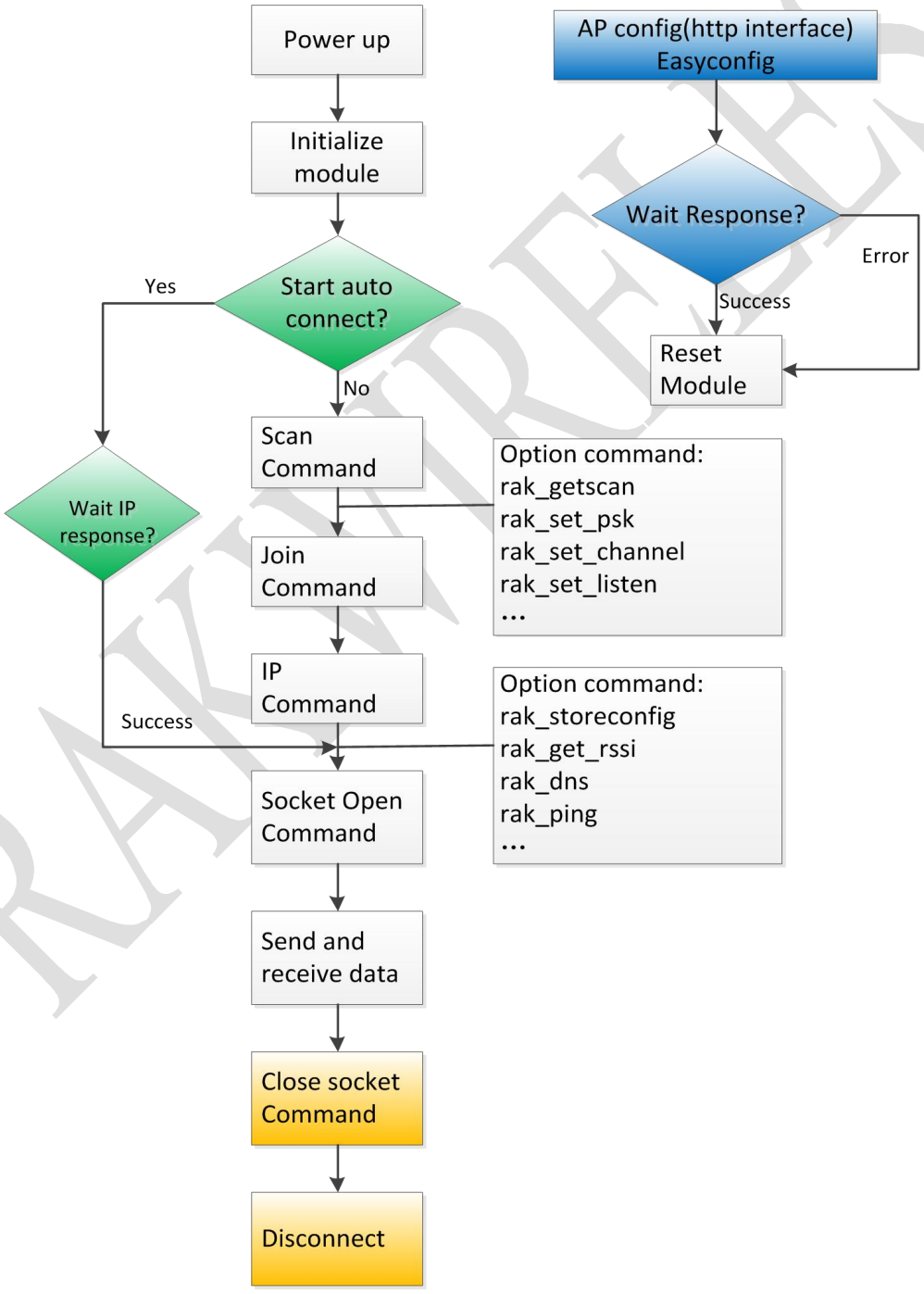
2.2 Operation flowchart

RAK476 UART command operation feature completes a few basic steps of WIFI communication, including network scanning, joining network and obtaining an IP address, and eventually establishing Socket communication. RAK476 provides a variety of convenient

operation to implement networking, so that customers can easily complete the network configuration, and concentrate on the management of socket and their own data protocols.

To realize automatic networking management, customers can take advantage of AP configuration and EasyConfig configuration module. The module will automatically store paramenters after a successful configuration, and these automatic networking commands can be used any time, letting the module automatically complete networking operation, and returning the results.

The basic operation of the process is as follows:



2.3 AT command format

Host to Module:

at+<command> = <parameter 1>, <parameter 2>, ... <parameter n> \ r \ n

All AT commands, including parameters, are in ASICII codes, for instance

at+psk = RAKwireless \ r \ n

at+connect = RAKwireless \ r \ n

1) After the implementation of each command, the module will send the return value in the following format

2) If the command is successfully implemented, the return value is

OK \ r \ n or OK <parameter 1> <parameter 2> ..... <parameter n> \ r \ n

Note:

Except for "OK", the other parameters are all hexadecimal, for example,

OK \ r \ n HEX = 4F 4B 0D 0A ---------- no parameter

OK @ \ r \ n HEX = 4F 4B 64 0D 0A ---------- parameter = 0x64

3) If the command execution fails, the return value is

ERROR <code>

Note:

ERROR is ASCII code, <code> is hexadecimal, for example,

ERROR? \r\n HEX = 45 52 52 4F 52 FE 0D 0A ---------- <code> = 0xFE

AT command format description

AT command begins with "at+" (all in lower case) and ends with "\ r \ n", the maximum command length is 80 bytes, and commands with beginnings in any other formats are wrong.

Note:

The aforementioned AT command format does not apply to the receiving data commands "at+ recv_data" and "at+ send_data", for more details please refer to at+recv_data, at+send_data.

ERROR CODE

Code	Description
-1	Parameter input error (parameter unrecognizable / missing parameter / too long command / other illicit parameters)
-10	Module is busy (wait till module processing is completed)
-12	Unknown error (internal storage, system, etc.)
-14	The system is upgrading, please operate later.
other	See specific commands for details

2.4 Information of boot up

The boot up time of RAK476 is about 700ms, the module automatically prints the boot information after normal start:

ASCII ------ Welcome to RAK476 \ r \ n

HEX -------- 57 65 6C 63 6F 6D 65 20 74 6F 20 52 41 4B 34 37 36 0D 0A

2.5 AT command summarization

AT operating command is divided into module management command, network operation command, mqtt and socket operation command and parameter perservation command. The following table is an encyclopedia of commands:

AT command	Description
Module management command	
at+ascii = <mode> \ r \ n	Opening ASCII display
at+mac \ r \ n	Query of module MAC address
at+version \ r \ n	Query of software version
at+pwrmode = <mode> \ r \ n	Setting power consumption mode of module
at+reset \ r \ n	Reset module
at+set_hostname = <name> \ r \ n	Setting module hostname
at+get_hostname \ r \ n	Acquiring module hostname
at+set_funcbitmap = <bitmap> \ r \ n	Setting special function bit
at+get_funcbitmap\r\n	Reading special function bit
at+data_mode\r\n	Setting data mode command
Network command	
at+scan = <channel>, <ssid> \ r \ n	Scanning wireless network
at+get_scan = <scan_num> \ r \ n	Reading the specified number of scanning results
at+psk = <passphrase> \ r \ n	Setting passwords for joining / creating network
at+channel = <channel> \ r \ n	Setting informaiton channel for joining / creating network
at+ap = <ssid> \ r \ n	Creating the name of AP Network

AT command	Description
at+connect = <ssid> \ r \ n	Connecting the specified network
at+ipstatic = <ip>, <mask>, <gateway>, <dns server1>, <dns server2> \ r \ n	The IP address information of static setting module
at+ipdhcp=<mode>\r\n	Opening DHCP Sever under dynamic allocation of IP or AP
at+easy_config \ r \ n	Module command of one-key-networking
at+con_status \ r \ n	Query of network connection status
at+ipconfig \ r \ n	Query of IP address information
at+rssi \ r \ n	Query of network signal strength
at+dns = <domain> \ r \ n	DNS name resolution
at+apconfig = <hidden>, <contry code> \ r \ n	Advanced settings of AP Network
at+disc \ r \ n	Disconnectting the currently connected network
Mqtt command	
at+mqtt_init = clientId, alive \ r \ n	Initialization of mqtt parameter
at+mqtt_auth = username, password \ r \ n	Setting authentication parameters
at+mqtt_con = svr_ip, svr_port \ r \ n	Connecting the Server
at+mqtt_discon=fd\r\n	Disconnecting Mqtt connection
at+mqtt_con_status\r\n	Query of mqtt connection status
at+ mqtt_sub=sub_topic\r\n	Subscribing themes
at+mqtt_unsub=sub_topic\r\n	Unsubscribing themes
at+mqtt_pub = pub_topic, retain \ r \ n	Setting push topics
socket command	
at+ltcp = <local_port> \ r \ n	Creating TCP Server
at+tcp = <dest_ip>, <dest_port>, <module_port> \ r \ n	Establishing TCP Client
at+ludp=<local port>\r\n	Creating UDP Server
at+udp=<dest_ip>,<dest_port>,<local_port>\r\n	Establishing UDP Client
at+multicast=<dest_ip>,<dest_port>,<local_port>\r\n	Creating UDP multicast communication
at+cls = <flag> \ r \ n	Closing opened socket handle
at+socket_status = <flag> \ r \ n	Query of socket status
at+read = <flag>, <data_length> \ r \ n	Reading data through way of query
at+send_data = <flag>, <dest_port>, <dest_ip>, <data_length>, <data_stream> \ r \ n	Sending data

AT command	Description
at+recv_data = <flag> ,<dest_port>, <dest_ip> ,<data_length> ,<data_stream> \ r \ n	Receiving data
Parameter perservation command	
at+storeconfig \ r \ n	Save the network parameter / tape parameter
at+get_storeconfig\r\n	Acquiring the saved network parameter
at+web_config = web_param \ r \ n	Modifying the network parameter of the built-in WEB server
at+get_webconfig\r\n	Acquiring the saved network parameter of built-in WEB
at+uartconfig = <baud rate>, <data bits>, <stop bits>, <parity>, <flow ctrol> \ r \ n	Modifying the UART parameter of module, current effect
at+store_uartconfig = <baud rate>, <data bits>, <stop bits>, <parity>, <flow ctrol> \ r \ n	Modifying the UART parameter of module, reset effect
at+auto_connect \ r \ n	Starting automatical net-connecting
at+start_web \ r \ n	Open WEB service, configuring module
Firmware upgrade	
at+upgrade\r\n	Into the firmware upgrade mode

2.6 Management command

2.6.1 Converting HEX into ASCII display

Command

at+ascii = <mode> \ r \ n

Description

The return value of all commands is displayed in ASCII after the command conversion, which facilitates user debugging and getting familiar with the AT command. There is no need to open while programming.

Parameter description

Parameter	Parameter value	Description
<Mode>	0	Disable conversion

	1	Enable conversion

Description of return value

Parameter	Format	Length (byte)	Description
The command is successfully implemented			
OK	ASCII	2	Starting successfully
\r\n	ASCII	2	Terminator
Failure of command execution			
ERROR	ASCII	5	Error
<CODE>	HEX	1	See ERROR list for details
\r\n	ASCII	2	Terminator
Remark			

2.6.2 Query of module MAC

Command

at+mac \ r \ n

Description

Query of MAC address information of module

Parameter description

NULL

Description of return value

Parameter	Format	Length (byte)	Description
The command is successfully implemented			
OK	ASCII	2	OK
MAC	HEX	6	MAC address
\r\n	ASCII	2	Terminator
Failure of command execution			
ERROR	ASCII	5	ERROR
<CODE>	HEX	1	See ERROR list for details
\r\n	ASCII	2	Terminator
Remark			

2.6.3 Query of version information

Command

at+version \ r \ n

Description

Query of module version,including software version and the version of the WLAN.

For example 0.0.0.1-1.0.1, software version is 0.0.0.1,WLAN version is1.0.1

Parameter description

NULL

Description of return value

Parameter	Format	Length (byte)	Description
The command is successfully implemented			
OK	ASCII	2	OK
	STRING		Character string
/r/n	ASCII	2	Terminator
Failure of command execution			
ERROR	ASCII	5	ERROR
<CODE>	HEX	1	See ERROR list for details
\r\n	ASCII	2	Terminator
Remark			

2.6.4 Setting power consumption mode

Command

at+pwrmode = <mode> \ r \ n

Description

RAK476 supports two kinds of power modes, as are shown in the following table

Mode	Control section	Wireless section	Methods of waking up	Average power consumption
0	Normal	Normal	Not required	80mA
1	Sleep_Mode	Power_Save	Not required	30-50mA

pwrmode = 0 ---------- mode 0

Module operates at maximum performance with the control section and wireless section fully open.

pwrmode = 1 ---------- mode 1

The control section enters shallow hibernation, while the wireless section keeps the current connection status and enters a low-power-consumption mode, with normal communication, but this will reduce the module performance, and the speed of sending and receiving data may decrease.

Note: The power mode under AP mode only supports Mode 0

Parameter description

Parameter	Parameter value	Description
<Mode>	0	Setting the power mode as 0
	1	Setting the power mode as 1

Description of return value

Parameter	Format	Length (byte)	Description
The command is successfully implemented			
OK	ASCII	2	Configuration succeeded
\r\n	ASCII	2	Terminator
Failure of command execution			
ERROR	ASCII	5	Error
<CODE>	HEX	1	See ERROR list for details
\r\n	ASCII	2	Terminator
Remark			

2.6.5 Reset module

Command

at+reset \r\n

Description

Module resets.

Parameter description

NULL

Description of return value

Parameter	Format	Length (byte)	Description
The command is successfully implemented			

OK	ASCII	2	Reset succeeded
\r\n	ASCII	2	Terminator
Failure of command execution			
ERROR	ASCII	5	Error
<CODE>	HEX	1	See ERROR list for details
\r\n	ASCII	2	Terminator
Remark			

2.6.6 Setting module hostname

Command

at+set_hostname = <name> \ r \ n

Description

When setting the host name of module, if the module returns "OK", it means the setting is successful and it will take effect after the module restarting. The default is rak_xxxxxx (_xxxxxx is the last six characters of MAC address), and the suffix is added automatically when setting the host name.

Parameter description

Parameter	Parameter value	Description
<Name>	Host name	The maximum name length is 32Byte (including suffixes)

For example:

at+set_hostname = RAK476 \ r \ n ---------- setting the hostname as RAK476_xxxxxx

Description of return value

Parameter	Format	Length (byte)	Description	
The command is successfully implemented				
OK	ASCII	2	Setting succeeded	
\r\n	ASCII	2	Terminator	
Failure of command execution				
ERROR	ASCII	5	Error	
<CODE>	HEX	1	0XFB=-5	Invalid storage area
\r\n	ASCII	2	Terminator	
Remark				

2.6.7 Acquiring module hostname

Command

at+get_hostname \ r \ n

Description

Acquiring the hostname of module, it indicates success when the module returns "OK" .

Parameter description

NULL

For example:

at+get_hostname \ r \ n ---------- acquiring hostname

Description of return value

Parameter	Format	Length (byte)	Description	
The command is successfully implemented				
OK	ASCII	2	Setting succeeded	
<Name>	X HEX HEX HEX	32	Host name	
\r\n	ASCII	2	Terminator	
Failure of command execution				
ERROR	ASCII	5	Error	
<CODE>	HEX	1	0XFB=-5	Invalid storage area
\r\n	ASCII	2	Terminator	
Remark				

2.6.8 Setting special function bit

Command

at+set_funcbitmap = <bitmap> \ r \ n

Description

Set some special function bits for the current module, and they will take effect after reset.

Parameter description:

bitmap length 4byte

bitmap [0] - whether to enable the query and reception function, the default setting is 0.

0 - Disable query reception

1 - Enable query reception

bitmap [1] -bitmap [31]: Preservation

Description of return value

Parameter	Format	Length (byte)	Description	
The command is successfully implemented				
OK	ASCII	2	Configuration succeeded	
\r\n	ASCII	2	Terminator	
Failure of command execution				
ERROR	ASCII	5	Error	
<CODE>	HEX	1	0XFB=-5	Invalid storage area
\r\n	ASCII	2	Terminator	
Remark				

2.6.9 Reading special function bit

Command

at+get_funcbitmap\r\n

Description

When acquiring the special function bit of module, the acquiring is successful when the module returns "OK".

Parameter description

NULL

For example:

at+get_funcbitmap \ r \ n ---------- acquiring special function bit

Description of return value

Parameter	rmat Format Format Format	Length (byte)	Description	
The command is successfully implemented				
OK	ASCII	2	Setting succeeded	
<bitmap>	HEX	4	bit [0]	Enable bit of query and reception
			bit [1] -bit [31]	Preservation
\r\n	ASCII	2	Terminator	
Failure of command execution				
ERROR	ASCII	5	Error	

<CODE>	HEX	1	0XFB=-5	Invalid storage area
r\n \r\n	ASCII	2	Terminator	
Remark				

2.6.10 Setting data mode command

Command

at+data_mode\r\n

Description

Invoke this command after the establishment of socket. Only one socket can exist when entering the data mode; if the socket is disconnected, the module will exit the data mode automatically and receive the return information. (With the TCP is in connection, the module will both automatically exit the date mode whether the module disconnects the socket actively or passively. While the UDP is in connection, the module will automatically exit the data mode only when the module actively disconnects the socket)

Actively exit the data mode:

1. Upper computer (master MCU) sends "+++", to request to enter the command mode.

2. Timing 200ms and waiting for the module to return "U" (0x55), if the module gives no return within the specified time, then send "+++" again at the end of the 200ms timing to request to enter the command mode until the module successfully returns "U" (0x55) which indicates that the module is ready to enter the command mode and is waiting for the final confirmation (waiting 3S).

3. After receiving "U" (0x55), the upper computer (master MCU) sends the module the final confirmation information, "U" (0x55), within 3S. The module returns "OK" after reception and enters the command mode. If the module fails in reception within 3S, it will exit the Ready status and rewait for the request command. Repeat steps 1- 3.

Parameter description

NULL

Description of return value

Parameter	Format	Length (byte)	Description
The command is successfully implemented			
OK	ASCII	2	Configuration succeeded

\r\n	ASCII	2	Terminator	
Failure of command execution				
ERROR	ASCII	5	Error	
<CODE>	HEX	1	0XFE =-2	Fail to enter the data mode
\r\n	ASCII	2	Terminator	
Remark				

2.7 Network command

2.7.1 Scanning wireless network

Command

at+scan = <channel>, <ssid> \ r \ n

Description

Scan the wireless network through this command, and get the wireless information including encrypted information, information channel, signal strength, BSSID and so forth. If there is no need to get the specified SSID information, this command can be ignored. When implementing the connect command, the internal of module automatically performs scanning.

Parameter description

The scan command includes two parameters, of which <channel> is scanning specified informaiton channel with the value range of 1-13, if set as 0, then all information channels will be scanned. <ssid> is scanning specified SSID, and this parameter is optional.

Note:

If to specify infromation channel for scanning, then the scanning time can be reduced!

Parameter	Parameter value	Description
<channel>	0-13	Specify informaiton channel (1-13) for scanning, and scan all channels when the value is 0.
<ssid>	Network name	Specify SSID (optional)

For example:

at+scan = 0 \ r \ n ---------- scan all information channels

at+scan = 0, RAKwireless \ r \ n ---------- scan wireless network named "RAKwireless" among all channels

at+scan = 8, RAKwireless \ r \ n ---------- scan wireless network named "RAKwireless" in informaiton channel 8

at+scan = 6 \ r \ n ---------- scan all SSID in the information channel 6

Description of return value

If the command is successfully implemented, return "OK" and the number of scanned wireless networks (a maximum of 20); if the user needs to use the network information, which can be acquired by the command of "at+get_scan".

Note:

When at+ascii = 1, the module will return all information, only to facilitate debugging and observation

Parameter	Format	Length (byte)	Description	
The command is successfully implemented				
OK	ASCII	2	Network has been scanned	
<SCAN NUM>	HEX	1	Number of wireless network	
\r\n	ASCII	2	Terminator	
Failure of command execution				
ERROR	ASCII	5	Error	
<CODE>	HEX	1	0XFE=-2	The specified ssid is not found
\r\n	ASCII	2	Terminator	
Remark				

2.7.2 Reading scanning information

Command

at+get_scan = <scan_num> \ r \ n

Description

Read the scanning information by command, and this command must be invoked after the command of "at+scan" - scanning wireless network.

Note:

If there is no need to get the wireless network information, this command can be omitted!

After the scanning information have all been read, if to read again, the module will return error -2, you need to invoke the command of "at+scan" for rescanning!

Parameter description

<scan_num> is the number of read scanning information, if the parameter is greater than the actual number of scanning, the command will return the actual number of scanning.

Parameter	Parameter value	Description

<scan_num>	Greater than 0	Read number of scanning information

For example:

at+get_scan = 10 \ r \ n ---------- read 10 wireless network information

Description of return value

Parameter	Format	Length (byte)	Description							
The command is successfully implemented										
OK	ASCII	2	The information acquired is correct							
<SSID>	HEX	33	SSID							
<BSSID>	HEX	6	BSSID							
<CHANNEL>	HEX	1	Information channel							
<RSSI>	HEX	1	Signal strength (negative)							
<Security Mode>	HEX	1	Encryption method							
			b7	b6	b5	b4	b3	b2	b1	b0
			WPA2	WPA	WEP	802.1X	PSK	WEP	TKIP	CCMP
\r\n	ASCII	2	Terminator							
Failure of command execution										
ERROR	ASCII	5	Error							
<CODE>	HEX	1	0XFE=-2	1, The scanning information has all been read						
\r\n	ASCII	2	Terminator							
Remark	b7-b5: encryption method b4-b3: encryption type b2-b0: Encryption Algorithm									

2.7.3 Setting password

Command

at+psk = <passphrase> \ r \ n

Description

If the module is operating in the station mode, this command is used to enter the network password, RAK476 supports WPA-PSK, WPA2-PSK and WPA-PSK + WPA2-PSK encryption, among which WPA-PSK and WPA2-PSK support TKIP, mixed encryption algorithm of CCMP and TKIP+CCMP.

If the module is working in AP mode, this command is used to set up network password.

In AP mode, network encryption method is WAP2-PSK-CCMP by default, and other encryption methods are not supported. Password must be ASCII characters of 8-63 bytes, or

hexadecimal characters of 64 bytes (0-9, a-f).

Note:

1, if the network for connection is OPEN, this command can be omitted!

2, RAK476 does not support character password contains a comma (,)

Parameter description

Parameter	Parameter value	Description
<passphrase>	Password	Input or set password

For example:

at+psk = RAKwireless \ r \ n ---------- 8-byte character password, under the mode of WPA2 or WPA

Description of return value

Parameter	Format	Length (byte)	Description
The command is successfully implemented			
OK	ASCII	2	Setting succeeded
\r\n	ASCII	2	Terminator
Failure of command execution			
ERROR	ASCII	5	Error
<CODE>	HEX	1	See ERROR list for details
\r\n	ASCII	2	Terminator
Remark			

2.7.4 Setting information channel

Command

at+channel = <channel> \ r \ n

Description

Setting network infromaiton channel of AP mode, this parameter must be invoked before the establishment of network.

Parameter description

Parameter	Parameter value	Description
<channel>	1-13	Setting information channel

For example:

at+channel = 1 \ r \ n Setting network information channel as channel 1

Description of return value

Parameter	Format	Length (byte)	Description
The command is successfully implemented			
OK	ASCII	2	OK
\r\n	ASCII	2	Terminator
Failure of command execution			
ERROR	ASCII	5	ERROR
<CODE>	HEX	1	See ERROR list for details
\r\n	ASCII	2	Terminator
Remark			

2.7.5 Creating wireless access point

Command

at+ap = <ssid> \ r \ n

Description

The user entering this command can create a wireless access point (AP), other wireless devices can be connected to this network to send and receive data. The maximum number of clients in connection is 3.

Before entering command, the user can configure other wireless parameters using "at+psk" PASSWORD SETTING, "at+apconfig" ADVANCE PARAMETER, "at+channel" SETTING INFORMATON CHANNEL, the default channel of network is 6 (2437MHZ).

After the establishment is completed, the user needs to configure static IP (at+ipstatic), then startup DHCPserver, and invoke the command of "at+ipdhcp = 1".

Parameter description

The maximum length of SSID is 32 bytes

Parameter	Parameter value	Description
SSID> <ssid>	SSID	The maximum support length of SSID to be connected is 32

For example: at+ap = RAKwireless \ r \ n ---------- establishing SSID as "RAKwireless"

Description of return value

Parameter	Format	Length (byte)	Description
The command is successfully implemented			
OK	ASCII	2	Connection succeeded
\r\n	ASCII	2	Terminator
Failure of command execution			
ERROR	ASCII	5	Error
<CODE>	HEX	1	0XFE=-2 Creation failed
\r\n	ASCII	2	Terminator
Remark			

2.7.6 Connecting wireless network

Command

at+connect = <ssid> \ r \ n

Description

This command is used to connect to the specified network, if the network is encrypted network, the command can be invoked only after the command of "at+psk " is input. If the network password is empty, then "at+scan" and "at+psk" need not to be invoked.

Parameter description

The maximum length of SSID is 32 bytes

Parameter	Parameter value	Description
<ssid>	SSID	The maximum support length of SSID to be connected is 32

For example:

at+connect = RAKwireless \ r \ n ---------- the connection to SSID is network of "RAKwireless"

Description of return value

Parameter	Format	Length (byte)	Description
The command is successfully implemented			
OK	ASCII	2	Connection succeeded
\r\n	ASCII	2	Terminator
Failure of command execution			

ERROR	ASCII	5	Error	
<CODE>	HEX	1	0XFE=-2	No SSID is found
<CODE>	HEX	1	0XFD=-3	Connection failed
\r\n	ASCII	2	Terminator	
Remark				

2.7.7 Setting static IP address

Command

at+ipstatic=<ip>,<mask>,<gateway>,<dnsserver1>,<dnsserver2>\r\n

Description

Setting static IP address

Parameter description

Parameter	Parameter value	Description
<IP>	0.0.0.0-255.255.255.255	IP address setting , suggest set into the class C IP address
<NETMASK>	0.0.0.0-255.255.255.255	Subnet mask setting
<GATEWAY>	0.0.0.0-255.255.255.255	Gateway setting
<DNS SERVER1>	0.0.0.0-255.255.255.255 (can be 0)	DNS server 1 setting
<DNS SERVER2>	0.0.0.0-255.255.255.255 (can be 0)	DNS server 2 setting

For example:

at+ipstatic = 192.168.9.5,255.255.255.0,192.168.9.1,192.168.9.1,0 \ r \ n

Configuration module IP = 192.168.9.5

Subnet mask = 255.255.255.0

Gateway = 192.168.9.1

dns server 1 = 192.168.9.1

dns server 2 = 0

Description of return value

Parameter	Format	Length (byte)	Description
The command is successfully implemented			
OK	ASCII	2	IP address is configured successfully
\r\n	ASCII	2	Terminator
Command execution error			

ERROR	ASCII	5	Error	
<CODE>	HEX	1	0XFE=-2	Error of setting IP address
\r\n	ASCII	2	Terminator	
Remark				

2.7.8 Setting DHCP mode

Command

at+ipdhcp=<mode>\r\n

Description

Setting DHCP function of module

Parameter description

If <mode> = 0, the module operates in DHCP CLIENT mode, which will get the IP address and other information from the DHCP SERVER.

If <mode> = 1, the module will automatically set the DHCP SERVER parameter, including the IP address pool range and lease time, this function can be used only after the "at+ipstatic" command has been sent, vailid under the AP mode.

Parameter	Parameter value	Description
<Mode>	0	=0 DHCP CLIENT
	1	=1 DHCP SERVER

For example:

at+ipdhcp = 0 \ at r \ n ---------- the module working in DHCP ClIENT mode

at+ipdhcp = 1 \ In r \ n ---------- the module working in DHCP SERVER mode

Description of return value

Parameter	Format	Length (byte)	Description
The command is successfully implemented			
OK	ASCII	2	The command is successfully implemented
<MAC>	HEX	6	Module MAC address
<IP>	HEX	4	Module IP address
<NETMASK>	HEX	4	Module subnet mask
<GATEWAY>	HEX	4	Gateway

<DNS1>	HEX	4	DNS server 1	
<DNS2>	HEX	4	DNS server 2	
\r\n	ASCII	2	Terminator	
Command execution error				
ERROR	ASCII	5	Error	
<CODE>	HEX	1	0XFC=-4	Address acquiring timeout (<mode> = 0 valid)
\r\n	ASCII	2	Terminator	
Remark				

2.7.9 EasyConfig net-connection

Command

at+easy_config \ r \ n

Description

Open module EasyConfig function through this command. With the mobile APP software, finish the module' s joining the specified network. When it is successful, the user can choose to save the parameters of the current network to the internal module through the command of "at+storeconfig". After reset, the automatic net-connection command "At+auto_connect" can be used (see the command description for details).

Description of return value

Parameter	Format	Length (byte)	Description
The command is successfully implemented			
OK	ASCII	2	Connected to the network
<MAC>	HEX	6	Module MAC address
<IP>	HEX	4	Module IP address
<NETMASK>	HEX	4	Module subnet mask
<GATEWAY>	HEX	4	Gateway
<DNS SERVER1>	HEX	4	DNS server 1
<DNS SERVER2>	HEX	4	DNS server 2
\r\n	ASCII	2	Terminator
Failure of command execution			
ERROR	ASCII	5	Error

<CODE>	HEX	1	0XFE =-2	AP not found
			0XFD=-3	Connection failed
			0XFC =-4	Obtaining IP address failed
			0XFA=-6	Easy config failure
\r\n	ASCII	2	Terminator	
Remark				

2.7.10 Query of wireless network connection status

Command

at+con_status \ r \ n

Description

If the module is operating in Station mode, this command is used for wireless network connection status of module.

If the module is operating in AP mode, this command is used to judge the connection status of the device.

Parameter description

NULL

Description of return value

Parameter	Format	Length (byte)	Description
The command is successfully implemented			
OK	ASCII	2	The command is successfully implemented
1	ASCII	1	1: Wireless network already connected / device already connected 0: Not connected
\r\n	ASCII	2	Terminator
Failure of command execution			
ERROR	ASCII	5	Error
<CODE>	HEX	1	See ERROR list for details
\r\n	ASCII	2	Terminator
Remark			

2.7.11 Query of IP information of module

Command

at+ipconfig \ r \ n

Description

Query of the IP information of the current module, including MAC address, IP address, subnet mask, gateway, DNS server

Parameter description

NULL

Description of return value

Parameter	Format	Length (byte)	Description
The command is successfully implemented			
OK	ASCII	2	Query succeeded
<MAC>	HEX	6	Module MAC address
<IP>	HEX	4	Module IP address
<NETMASK>	HEX	4	Module subnet mask
<GATEWAY>	HEX	4	Gateway
<DNS SERVER1>	HEX	4	DNS server 1
<DNS SERVER2>	HEX	4	DNS server 2
\r\n	ASCII	2	Terminator
Command execution error			
ERROR	ASCII	5	Error
<CODE>	HEX	1	0XFC = -4 Query failed
\r\n	ASCII	2	Terminator
Remark			

2.7.12 Query of network signal strength

Command

at+rssi \ r \ n

Description

Query of the wireless network strength, valid under STA mode.

Parameter description

NULL

Description of return value

Parameter	Format	Length (byte)	Description	
The command is successfully implemented				
OK	ASCII	2	OK	
<RSSI>	HEX	1	Signal strength (negative) for example: -50, the lower the signal strength, the smaller the return value.	
\r\n	ASCII	2	Terminator	
Failure of command execution				
ERROR	ASCII	5	ERROR	
<CODE>	HEX	1	0XFE=-2	No network connection or module working in AP mode
\r\n	ASCII	2	Terminator	
Remark				

2.7.13 Domain name resolution

Command

at+dns=<domain> \r\n

Description

Convert domain name to the corresponding IP address, this command must be configured with usable DNS server address

Parameter description

NULL

Description of return value

Parameter	Format	Length (byte)	Description	
The command is successfully implemented				
OK	ASCII	2	Query succeeded	
<IP>	HEX	4	IP address	
\r\n	ASCII	2	Terminator	
Command execution error				
ERROR	ASCII	5	Error	
<CODE>	HEX	1	0XFD=-3	DNS resolution failed or no network connection

\r\n	ASCII	2	Terminator
Remark			

2.7.14 Advanced parameters

Command

at+apconfig = <hidden>, <contry code> \ r \ n

Description

This command is used to set wireless access point parameter, including the country code, and whether to hide the access point.

Parameter description

Parameter	Parameter value	Description
<hidden>	0	Set the network as visible (optional parameter)
	1	Set the network as hidden (optional parameter)
<contry code >	country code	Country code, such as China (CN) CN (China) can join 1 to 13 information channel router JP (Japan) can join 1 to 14 information channel router US (America) can join 1 to 11 information channel router

Description of return value

Parameter	Format	Length (byte)	Description
The command is successfully implemented			
OK	ASCII	2	Setting succeeded
\r\n	ASCII	2	Terminator
Failure of command execution			
ERROR	ASCII	5	Error
<CODE>	HEX	1	See ERROR list for details
\r\n	ASCII	2	Terminator
Remark			

2.7.15 Disconnectting wireless network

Command

at+disc \ r \ n

Description

Disconnectting the current wireless connection

Parameter description

NULL

Description of return value

Parameter	Format	Length (byte)	Description
The command is successfully implemented			
OK	ASCII	2	Disconnect succeeded
\r\n	ASCII	2	Terminator
Failure of command execution			
ERROR	ASCII	5	Error
<CODE>	HEX	1	0XFE=-2 Network not connected
\r\n	ASCII	2	Terminator
Remark			

2.8 Mqtt command

MQTT (Message Queuing Telemetry Transport) is an instant messaging protocol developed by IBM that supports all platforms. It is a communication protocol which almost can link all networked objects with the external, and it is used as a sensor.

The RAK476 module applies the version of MQTT V3, and the module only supports one MQTT connection.

2.8.1 Initialization of mqtt parameter

Command

at+mqtt_init=<clientId>,<alive>\r\n

Description

This command is used to initialize the necessary parameters of mqtt connection, the command needs to be invoked before the command of "at+mqtt_con" 2.8.3Connecting the Server

Parameter description

mqtt_init contains two parameters, <clientId> is the client ID of connecting to the Server , the maximum length of clientId allowes 50 bytes. <alive> is the alive time setting with the Server, unit of second (S), alive range is 30-300 seconds

Parameter	Parameter value	Description
<clientId>	1-50Byte	The client ID connected to the Server
<alive>	30-300S	The keep-alive time setting with the Server, unit of second (S)

For example:

at+mqtt_init= ID,100\r\n

Description of return value

Parameter	Format	Length (byte)	Description
The command is successfully implemented			
OK	ASCII	2	Setting succeeded
\r\n	ASCII	2	Terminator
Failure of command execution			
ERROR	ASCII	5	Error
<CODE>	HEX	1	See ERROR list for details
\r\n	ASCII	2	Terminator
Remark			

2.8.2 Setting authentication parameters

Command

at+mqtt_auth=<username>,<password>\r\n

Description

The authentication parameter setting the mqtt connecting the Server may not be used if the Server does not need it, this command needs to be invoked before the connect command.

Parameter description

mqtt_auth contains two parameters, <username> is the user name used in the Server authentication, and the maximum length of username allowes 64 bytes. <password> is the password used in the Server authentication, the maximum length of the password allows 64

bytes.

Parameter	Parameter value	Description
<username>	0-64Byte	The username used in the Server authentication.
<password>	0-64Byte	The password used in the Server authentication.

For example:

at+mqtt_auth= RAK,RAKwireless\r\n

Description of return value

Parameter	Format	Length (byte)	Description
The command is successfully implemented			
OK	ASCII	2	Setting succeeded
\r\n	ASCII	2	Terminator
Failure of command execution			
ERROR	ASCII	5	Error
<CODE>	HEX	1	See ERROR list for details
\r\n	ASCII	2	Terminator
Remark			

2.8.3 Connecting the Server

Command

at+mqtt_con=<svr_ip>,<svr_port>\r\n

Description

Implement mqtt to connect the Server.

Parameter description

Mqtt_con contains three parameters, <svr_ip> is domain name or IP address of the Server. <svr_port> is the Server port.

Parameter	Parameter value	Description
<svr_ip>	0.0.0.0-255.255.255.255	The domain name or IP address of the Server

<svr_port>		The Server port

For example:

at+mqtt_auth = 192.168.9.5,2500, \ r \ n

Parameter	Format	Length (byte)	Description	
The command is successfully implemented				
OK	ASCII	2	Connected the Server	
<fd>	HEX		Descriptor This description is a fixed value (0x14), does not take up 0-7	
\r\n	ASCII	2	Terminator	
Failure of command execution				
ERROR	ASCII	5		
<CODE>	HEX	1	0XFE =-2	Creation failed
			0XFD=-3	bind failed
			0XFC=-4	connect failed
			0XFB=-5	Authentication failed
			0XFB=-6	Request failed
\r\n	ASCII	2	Terminator	
Remark				

2.8.4 Disconnecting Mqtt connection

Command

at+mqtt_discon=<fd>\r\n

Description

Disconnect the mqtt server.

Parameter description

mqtt_discon contains a parameter, the descriptor returned when fd establishes connection.

Parameter	Parameter value	Description
<fd>	20(0x14)	The descriptor returned in connection establishment.

For example:

at+mqtt_discon=20\r\n

Description of return value

Parameter	Format	Length (byte)	Description
The command is successfully implemented			

OK	ASCII	2	Setting succeeded	
\r\n	ASCII	2	Terminator	
Failure of command execution				
ERROR	ASCII	5	Error	
<CODE>	HEX	1	0XFB=-6	Disconnect failed
\r\n	ASCII	2	Terminator	
Remark				

2.8.5 Query of mqtt connection status

Command

at+mqtt_con_status\r\n

Description

This command is used to inquire the connection status of devices with the mqtt Server.

Parameter description

NULL

Description of return value

Parameter	Format	Length (byte)	Description
The command is successfully implemented			
OK	ASCII	2	The command is successfully implemented
1	ASCII	1	1: Connected 0. Not connected
\r\n	ASCII	2	Terminator
Failure of command execution			
ERROR	ASCII	5	Error
<CODE>	HEX	1	See ERROR list for details
\r\n	ASCII	2	Terminator
Remark			

2.8.6 Subscription of theme

Command

at+mqtt_sub=<sub_topic>\r\n

Description

For theme subscription, such topics will be pushed to the module when received by the Server, for example, clientId/in indicates that the module has subscribed data (input data) sent by other clients.

Parameter description

mqtt_sub contains a parameter, sub_topic: the theme the module has subscribed.

Parameter	Parameter value	Description
<sub_topic>	1-64Byte	Themes subscribed by the module

For example:

at+mqtt_sub=abc123\r\n

Description of return value

Parameter	Format	Length (byte)	Description	
The command is successfully implemented				
OK	ASCII	2	Setting succeeded	
\r\n	ASCII	2	Terminator	
Failure of command execution				
ERROR	ASCII	5	Error	
<CODE>	HEX	1	0XFB=-6	Request failed
\r\n	ASCII	2	Terminator	
Remark				

2.8.7 Unsubscribing themes

Command

at+mqtt_unsub=<sub_topic>\r\n

Description

After unsubscribing themes, if the Server receives such topics, they will not be pushed to the module.

Parameter description

mqtt_unsub contains a parameter, sub_topic is the theme the module has subscribed.

Parameter	Parameter value	Description

<sub_topic>	1-64Byte	Themes subscribed by the module

For example:

at+mqtt_unsub=abc123\r\n

Description of return value

Parameter	Format	Length (byte)	Description
The command is successfully implemented			
OK	ASCII	2	Setting succeeded
\r\n	ASCII	2	Terminator
Failure of command execution			
ERROR	ASCII	5	Error
<CODE>	HEX	1	0XFB=-6 Request failed
\r\n	ASCII	2	Terminator
Remark			

2.8.8 Setting push topics

Command

at+mqtt_pub=<pub_topic>,<retain>\r\n

Description

Subscribe push topics.

Parameter description

at+mqtt_pub contains two parameters, sub_topic: themes subscribed by the module, retain: whether the Server preserves information.

Parameter	Parameter value	Description
<pub_topic>	1-64Byte	Themes subscribed by the module
<retain>	0	The Server retains information
	1	The Server does not retain information

For example:

at+mqtt_pub=abc123,1\r\n

Description of return value

Parameter	Format	Length (byte)	Description
The command is successfully implemented			
OK	ASCII	2	Setting succeeded
\r\n	ASCII	2	Terminator

Failure of command execution				
ERROR	ASCII	5	Error	
<CODE>	HEX	1	0XFB=-6	Request failed
\r\n	ASCII	2	Terminator	
Remark				

2.9 Socket command

2.9.1 Creating TCP SERVER

Command

at+ltcp = <local_port> \ r \ n

Description

The module, as a TCP server, also creates listening port, if the operation is successful, the module will return a hexadecimal identifier, used for managing the connection. To establish a TCP server, up to 7 clients can be connected. Create TCP server at different ports, up to 4 are allowed to be created. Close the descriptor of TCP server, client connected to this TCP sever will all be closed.

Parameter description

Parameter	Parameter value	Description
<local_port>	1-65535	Create local listening port

For example:

at+ltcp = 25000 \ r \ n ---------- create TCP SERVER

Description of return value

Parameter	rmat Format Format Format	Length (byte)	Description
The command is successfully implemented			
OK	ASCII	2	Creation Succeeded
<FLAG>	HEX	1	=0x08~0x0B Port identifier, used for port management
\r\n	ASCII	2	Terminator

Failure of command execution				
ERROR	ASCII	5	Error	
<CODE>	HEX	1	0XFE= -2	Creating local port error (the number of local ports created is more than 4)
			0XFD= -3	Binding local port error (repeat the creation of local ports)
			0XFB= -5	Listener error
\r\n	ASCII	2	Terminator	
Remark	Port identifier (0x08 ~ 0x0B) is only used to delete TCP Sever. The port identifier of the communication sending data is the port identifier (0x00~0x07) that is returned when the TCP-Client is connected.			

2.9.2 Creating TCP CLIENT

Command

at+tcp=<dest_ip>,<dest_port>,<local port>\r\n

Description

This command is module creating TCP CLIENT and connecting with the remote TCP SERVER, if the operation is successful, the module will return a hexadecimal identifier, used to manage the connection, this command can create a maximum of eight connections.

Parameter description

Parameter	Parameter value	Description
<dest ip>	0.0.0.-255.255.255.255	Destination IP address
<dest port>	1-65535	Destination port
<local port>	0-65535	Local port (can be omitted), 0 indicates randomly assigning ports

For example:

at+tcp = 192.168.9.5,25000,25001, \ r \ n ---------- use the fixed port to connect the Server

at+tcp = 192.168.9.5,25000 \ r \ n ---------- use the default port to connect the Server

Description of return value

Parameter	Format	Length (byte)	Description
The command is successfully implemented			
OK	ASCII	2	Connection succeeded

<FLAG>	HEX	1	=0x00~0x07 Port identifier, used for port management	
\r\n	ASCII	2	Terminator	
Failure of command execution				
ERROR	ASCII	5	Error	
<CODE>	HEX	1	0XFE= -2	Creating local port error
			0XFD= -3	Binding local port error
			0XFC= -4	Connecting TCP SERVER error
\r\n	ASCII	2	Terminator	
Remark				

2.9.3 Establishing UDP connection

Command

at+udp=<dest_ip>,<dest_port>,<local_port>\r\n

Description

This command is creating a UDP port on the module and setting the remote IP address and port number, if the establishment is successful, the module will return a hexadecimal identifier, used to manage the connection, this command can create up to eight connections.

Parameter description

Parameter	Parameter value	Description
<dest ip>	0.0.0.0-255.255.255.255	Destination IP address
<dest port>	1-65535	Destination port
<local port>	1-65535	Local port

For example:

at+udp = 192.168.9.5,25000,25001 \ r \ n ---------- connecting the destination port

at+udp = 192.168.9.255,25000,25001 \ r \ n ---------- broadcast

Description of return value

Parameter	Format	Length (byte)	Description
The command is successfully implemented			
OK	ASCII	2	Setting succeeded
<FLAG>	HEX	1	=0x00~0x07 Port identifier, used for port management
\r\n	ASCII	2	Terminator
Failure of command execution			

ERROR	ASCII	5	Error	
<CODE>	HEX	1	0XFE= -2	Creating local port error (the number of local ports created is more than 8)
<CODE>	HEX	1	0XFD= -3	Binding local port error
<CODE>	HEX	1	0XFC= -4	Connection to the destination port error
\r\n	ASCII	2	Terminator	
Remark				

2.9.4 Creating UDP SERVER

Command

at+ludp=<local port>\r\n

Description

This command is creating a UDP designated port listener at the local, a hexadecimal identifier will be retured after the creation is successful, waiting to receive remote port data. If the remote port sends data to this port, the received data will include IP and port information of the opposite side. If the module needs to reply to the opposite side, it can specify the IP and port number information of the opposite side when sending data, thus the reply is done. UDP server application is fairly flexible, which can receive unicast and broadcast messages and can also actively send data to the designated IP and port.

Parameter description

Parameter	Parameter value	Description
<local port>	1-65535	Create local port

For example:

at+ludp = 25000 \ r \ n ---------- create local port 25,000

Description of return value

Parameter	Format	Length (byte)	Description
The command is successfully implemented			
OK	ASCII	2	Creation Succeeded
<FLAG>	HEX	1	=0x00~0x07 Port identifier, used for port management

\r\n	ASCII	2	Terminator	
Failure of command execution				
ERROR	ASCII	5	Error	
<CODE>	HEX	1	0XFE= -2	Creating local port error (the number of local ports created is more than 8)
<CODE>	HEX	1	0XFD= -3	Binding local port error
\r\n	ASCII	2	Terminator	
Remark				

2.9.5 Creating UDP multicast

Command

at+multicast=<dest_ip>,<dest_port>,<local_port>\r\n

Description

This command is creating a UDP multicast socket on the module, the multicast IP which has joined the routing can be specified to implement data communication within the group, generally suitable for applications of one main with multi-followers within the group.

Parameter description

Parameter	Parameter value	Description
<dest ip>	224.0.0.1-239.255.255.255	Destination multicast IP address
<dest port>	1-65535	Destination port
<local port>	1-65535	Local port

For example:

at+multicast = 224.255.255.252,25001,25000 \ r \ n ---------- join multicast IP 224.255.225.252

Description of return value

Parameter	Format	Length (byte)	Description
The command is successfully implemented			
OK	ASCII	2	Creation Succeeded
<FLAG>	HEX	1	=0x00~0x07 Port identifier, used for port management
\r\n	ASCII	2	Terminator
Failure of command execution			

ERROR	ASCII	5	Error	
<CODE>	HEX	1	0XFE= -2	Creating local port error
<CODE>	HEX	1	0XFD= -3	Binding local port error
\r\n	ASCII	2	Terminator	
Remark				

2.9.6 Closing created port

Command

at+cls = <flag> \ r \ n

Description

Shut down the port listener or connection according to the corresponding identifier.

Parameter description

Parameter	Parameter value	Description
< flag >	0X00-0X07	Shut down the corresponding port

For example:

at+cls = 0 \ r \ n ---------- close the connection with port identifier 0

Description of return value

Parameter	Format	ength (byte) Length (byte)	Description	
The command is successfully implemented				
OK	ASCII	2	Closure succeeded	
\r\n	ASCII	2	Terminator	
Failure of command execution				
ERROR	ASCII	5	Error	
<CODE>	HEX	1	0XFE= -2	The specified port does not exist
<CODE>	HEX	1	0XFD= -3	Close failed
\r\n	ASCII	2	Terminator	
Remark				

2.9.7 Query of socket status

Command

at+socket_status = <flag> \ r \ n

Description

The current socket status, socket details can be inquired through commands.

Parameter description

Parameter	Parameter value	Description
< flag >	0-11	Connection identifier (ASCII)

Description of return value

Parameter	Format	Length (byte)	Description
The command is successfully implemented			
OK	ASCII	2	Data sent successfully
<VALID_NUM>	HEX	1	The number of currently available SOCKET
<FLAG>	HEX	1	Current SOCKET descriptor
<DEST_PORT>	HEX	2	Current SOCKET destination port number
<DEST_ADDR>	HEX	4	Current SOCKET destination address
\r\n	ASCII	2	Terminator
Remark			

2.9.8 Reading data through way of query

Command

at+read = <flag>, <data_length> \ r \ n

Description

This command is used to read data through query way, if there is no data to read, the range length is 0.

Parameter description

Parameter	Parameter value	Description
< flag >	0-7	Connection identifier (ASCII)
<data_length>	1-1400	The data length needs to be read

Description of return value

Parameter	Format	Length (byte)	Description
The command is successfully implemented			
OK	ASCII	2	Data sent successfully
data_length> <data_length>	HEX	2	The data length of actual return
<data_stream>	HEX	data_length> <data_length>	data
\r\n	ASCII	2	Terminator
Failure of command execution			
ERROR	ASCII	5	Data sending failed
<CODE>	HEX	1	0XFE=-2 socket does not exist
\r\n	ASCII	2	Terminator
Remark			

2.9.9 Sending data

Command

at+send_data = <flag>, <dest_port>, <dest_ip>, <data_length>, <data_stream> \ r \ n

Description

Send data to the target connection (port identifier) via command, the maximum data length is 1004, of which <data_stream> can be data in any format, the module will retain the original smaple of data, without any treatment. If the connection is TCP connection, where the target IP and target port can be ignored, filling in 0. When the connection is UDP, fill in 0 if there is no specification, if needing to be sent to a specified target as LUDP, fill in the target IP and destination port number.

Parameter description

Parameter	Parameter value	Description
< flag>	0-7	Connection identifier (ASCII)
<dest port>	1-65535	Destination port
<dest ip>	0.0.0.-255.255.255.255	Destination IP address
<data_length>	1-1400	Data length, a maximum of 1004 bytes (ASCII)
<data_stream>	data	The data (HEX) to be sent

For example:

at+send_data = 0,0,0,4, ABCD \ r \ n ---------- send 4 bytes of data to connection with identifier 0, the data content is "ABCD"

Description of return value

Parameter	Format	Length (byte)	Description	
The command is successfully implemented				
OK	ASCII	2	Data sent successfully	
\r\n	ASCII	2	Terminator	
Failure of command execution				
ERROR	ASCII	5	Data sending failed	
<CODE>	HEX	1	0XFE=-2	The specified socket does not exist
			0XFD=-3	Data sending error
\r\n	ASCII	2	Terminator	
Remark				

2.9.10 Receiving data

Command

at+recv_data = <flag>,<dest_port>,<dest_ip>,<data_length>,<data_stream> \ r \ n

or

at+recv_data=< socket_status>,<flag>,<dest_port>,<dest_ip>\r\n

Description

This command is the data the module sending to the host, including the destination port, destination IP, port identifier, data length, data. This command can also be used to receive TCP connection and disconnect information.

If socket type is udp, receive udp per packet is 1024Byte, part longer than 1024Byte will be discarded,so need set udp sender the max bytes per packet.

Parameter

NULL

Description of return value of data reception

Parameter	Format	Length (byte)	Description
Data successfully received			
<CMD>	ASCII	13	Command header
< flag>	HEX	1	=0X00-0X07 Port identifier

<dest_port>	HEX	2	Destination port
<dest_ip>	HEX	4	Destination IP
<data_length>	HEX	2	Data length
<data_stream>	HEX	<data_length>	data
\r\n	ASCII	2	Terminator
Data reception failed			
<CMD>	ASCII	13	Command header
<CODE>	HEX	1	See ERROR list for details
\r\n	ASCII	2	Terminator
Remark			

TCP connection status

Parameter	Format	Length (byte)	Description
TCP connection			
<CMD>	ASCII	13	Command header
< socket_status >	HEX	1	=0X80 TCP Client connection
< flag>	HEX	1	=0X00-0X07 Port identifier
<DEST_PORT>	HEX	2	Destination port
<dest_ip>	HEX	4	Destination IP
\r\n	ASCII	2	Terminator
TCP disconnect			
<CMD>	ASCII	13	Command header
< socket_status >	HEX	1	=0X81 TCP Client disconnection
< flag>	HEX	1	=0X00-0X07 Port identifier
<DEST_PORT>	HEX	2	Destination port
<dest_ip>	HEX	4	Destination IP
\r\n	ASCII	2	Terminator
Remark			

Network connection status

Parameter	Format	Length (byte)	Description
Network connection			
<CMD>	ASCII	13	Command header
< net_status >	HEX	1	=0X82 Net connection succeeded

\r\n	ASCII	2	Terminator
Remark	AP Mode: When there is client terminal connecting this AP hotspot, this packet is received.		
Network disconnect			
<CMD>	ASCII	13	Command header
< net_status >	HEX	1	=0X83 Network disconnection
\r\n	ASCII	2	Terminator
Remark	RAK476 disconnect will inform the host. AP Mode: When there is client terminal disconnecting this AP hotspot, this packet is received. STA mode: after this module successfully connecting the router, if network disconnection appears, this packet will be received.		

2.10 Parameter perservation

2.10.1 Configuring UART parameter

Command

at+uartconfig=<baud rate>,<data bits>,<stopbits>,<parity>,<flow ctrol>\r\n

Description

The configuration of UART parameter is valid under the current status, and start up by the default parameter 115200 after reset.

Parameter description

This command contains five parameters, including the baud rate, data bit, stop bit, parity check bit, and flow control switch

Parameter	Parameter value	Description
<baud rate>	See 2.1 for details	Baud rate
<data bits>	7	Data bit of 7
	8	Data bit of 8
<stop bits>	1	Stop bit of1
	2	Stop bit of 2
<parity>	0	No parity check bit
	1	Odd parity check
	2	Even parity check

<flow ctrol>	0	Disable
	1	Enable

For example:

at+uartconfig=115200,8,1,0,1\r\n------- set the UART parameter as the baud rate =115200

Data bit = 8

Stop bit = 1

Partiy check bit = no odd-even parity check

Flow control = Enable

Description of return value

Parameter	Format	Length (byte)	Description
The command is successfully implemented			
OK	ASCII	2	Configuration succeeded
\r\n	ASCII	2	Terminator
Failure of command execution			
ERROR	ASCII	5	Error
<CODE>	HEX	1	See ERROR list for details
\r\n	ASCII	2	Terminator
Remark			

2.10.2 Configure storage UART parameters

Command

at+store_uartconfig=<baud rate>,<data bits>,<stopbits>,<parity>,<flow ctrol>\r\n

Description

The UART parameter is configured, the reset is valid, and the UART parameter is modified after the reset.

Parameter description

This command contains five parameters, including the baud rate, data bit, stop bit, parity check bit, and flow control switch

Parameter	Parameter value	Description
<baud rate>	See 2.1 for details	Baud rate
<data bits>	7	Data bit of 7

	8	Data bit of 8
<stop bits>	1	Stop bit of1
	2	Stop bit of 2
<parity>	0	No parity check bit
	1	Odd parity check
	2	Even parity check
<flow ctrol>	0	Disable
	1	Enable

For example:

at+store_uartconfig=115200,8,1,0,1\r\n------- set the UART parameter as the baud rate =115200

Data bit = 8

Stop bit = 1

Partiy check bit = no odd-even parity check

Flow control = Enable

Description of return value

Parameter	Format	Length (byte)	Description
The command is successfully implemented			
OK	ASCII	2	Configuration succeeded
\r\n	ASCII	2	Terminator
Failure of command execution			
ERROR	ASCII	5	Error
<CODE>	HEX	1	See ERROR list for details
\r\n	ASCII	2	Terminator
Remark			

2.10.3 Saving configuration parameter

Command

at+storeconfig\r\n or at+storeconfig=param_struct\r\n

Description

Used to save user parameters, including password, SSID, IP address and scan information, this

command can be with or without parameter structure. Without parameters, they can be successfully saved after the correct implementation of scan, connect and IP obtaining

Parameter description

See the return value description in 2.10.3 Reading configuration parameter

For example:

at+storeconfig=param_struct\r\n

param_struct is HEX data： 00 00 00 01 00 06 00 01 52 41 4B 34 37 36 5F 57 45 42 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 07 A8 C0 00 FF FF FF 01 07 A8 C0 00 00 00 00 00 00 00 00 00 43 4E 00

```
typedef struct {
uint32_t feature_bitmap;
uint8_t net_type;
uint8_t channel;
uint8_t sec_mode;
uint8_t dhcp_mode;
char ssid[33];
char psk[65];
uint8_t dummy[2];
rw_IpConfig_t ip_param;
uint8_t ap_hidden;
uint8_t countryCode[3];
}rak_cfg_t;

typedef struct {
uint32_t addr;
uint32_t mask;
uint32_t gw;
uint32_t svr1;
uint32_t svr2;
}rw_IpConfig_t;
```

Description of return value

Parameter	Format	Length (byte)	Description
The command is successfully implemented			

OK	ASCII	2	Configuration succeeded	
\r\n	ASCII	2	Terminator	
Failure of command execution				
ERROR	ASCII	5	Error	
<CODE>	HEX	1	0XFB=-5	Invalid storage area
\r\n	ASCII	2	Terminator	
Remark				

2.10.4 Reading configuration parameter

Command

at+get_storeconfig\r\n

Description

Used to read the saved parameters of the user, to return the structure content of saved parameters.

Parameter description

NULL

Description of return value

Parameter	Format	Length (byte)	Description
The command is successfully implemented			
OK	ASCII	2	Configuration succeeded
feature_bitmap	HEX	4	Characteristic switch
net_type	HEX	1	Network type 0:STA, 1:AP
channel	HEX	1	Network information channel
sec_mode	HEX	1	Encryption mode
dhcp_mode	HEX	1	Acquisition mode of network IP address 0: static set 1: DHCP client
SSID	ASCII	33	Create / join network name
PSK	ASCII	65	Create / join network password
dummy	HEX	2	Null data
addr	HEX	4	IP address

mask	HEX	4	Subnet mask	
gw	HEX	4	Default gateway	
dnsrv1	HEX	4	DNS server 1	
dnsrv2	HEX	4	DNS server 2	
hidden	HEX	1	Advanced settings of AP Network whether to hide SSID	
country	ASCII	3	Advanced settings of AP Network country code settings CN (China) can join 1 to 13 information channel router JP (Japan) can join 1 to 14 information channel router US (America) can join 1 to 11 information channel router	
\r\n	ASCII	2	Terminator	
Failure of command execution				
ERROR	ASCII	5	Error	
<CODE>	HEX	1	0XFB=-5	Invalid storage area
\r\n	ASCII	2	Terminator	
Remark				

2.10.5 Modifying the startup parameter of WEB

Command

at+web_config = web_param \ r \ n

Description

Modify the network parameter when the module boots up WEB, the original factory establishes AP by default, the customer can modify this part of the parameter according to the actual application or requirement.

Parameter description

See the return value description in 2.10.5 Reading WEB startup parameter

For example:

at+web_config=web_param\r\n

web_param is HEX data： 00 00 00 01 00 06 00 01 52 41 4B 34 37 36 5F 57 45 42 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 07 A8 C0 00 FF FF FF 01 07

A8 C0 00 00 00 00 00 00 00 00 00 43 4E 00 61 64 6D 69 6E 00 00 00 00 00 00 00 00 00 00 00 61 64 6D 69 6E 00 00 00 00 00 00 00 00 00 00 00 00 00

```
typedef struct {
uint32_t feature_bitmap;
uint8_t net_type;
uint8_t channel;
uint8_t sec_mode;
uint8_t dhcp_mode;
char ssid[33];
char psk[65];
uint8_t dummy[2];
rw_IpConfig_t ip_param;
uint8_t ap_hidden;
uint8_t countryCode[3];
}rak_cfg_t;

typedef struct {
uint32_t addr;
uint32_t mask;
uint32_t gw;
uint32_t svr1;
uint32_t svr2;
}rw_IpConfig_t;

typedef struct {
char user_name[17];
char user_psk[17];
uint8_t dummy[2];
}rak_name_t;
```

Description of return value

Parameter	rmat Format Format Format	Length (byte)	Description	
The command is successfully implemented				
OK	ASCII	2	Reset succeeded	
\r\n	ASCII	2	Terminator	
Failure of command execution				
ERROR	ASCII	5	Error	
<CODE>	HEX	1	0XFB=-5	Invalid storage area
\r\n	ASCII	2	Terminator	
Remark				

2.10.6 Reading the startup parameter of WEB

Command

at+get_webconfig\r\n

Description

Read the network parameter of starting WEB configuration, return the structure parameter.

Parameter description

NULL

Description of return value

Parameter	Format	Length (byte)	Description
The command is successfully implemented			
OK	ASCII	2	Configuration succeeded
feature_bitmap	HEX	4	Characteristic switch
net_type	HEX	1	Network type 0:STA, 1:AP
channel	HEX	1	Network information channel
sec_mode	HEX	1	Encryption mode
dhcp_mode	HEX	1	Acquisition mode of network IP address 0: static set 1: DHCP client
SSID	ASCII	33	Create / join network name
PSK	ASCII	65	Create / join network password
dummy	HEX	2	Null data
addr	HEX	4	IP address
mask	HEX	4	Subnet mask
gw	HEX	4	Default gateway
dnsrv1	HEX	4	DNS server 1
dnsrv2	HEX	4	DNS server 2
hidden	HEX	1	Advanced settings of AP Network whether to hide SSID
country	ASCII	3	Advanced settings of AP Network country code settings CN (China) can join 1 to 13 information channel router JP (Japan) can join 1 to 14 information channel router US (America) can join 1 to 11 information channel router

user_name	ASCII	17	Webpage authentication user name	
user_psk	ASCII	17	Webpage authentication password	
dummy	HEX	2	Null data	
\r\n	ASCII	2	Terminator	
Failure of command execution				
ERROR	ASCII	5	Error	
<CODE>	HEX	1	0XFB=-5	Invalid storage area
\r\n	ASCII	2	Terminator	
Remark				

2.10.7 Starting WEB configuration

Command

at+start_web \ r \ n

Description

Open embedded WEB service, the module will start with the default WEB parameter, generally in the AP mode. After the user has joined, the browser can be used for wireless configuration of module parameter.

Parameter description

NULL

Return value

Parameter	Format	Length (byte)	Description	
The command is successfully implemented				
OK	ASCII	2	Start WEB successfully	
\r\n	ASCII	2	Terminator	
Failure of command execution				
ERROR	ASCII	5	Error	
<CODE>	HEX	1	0XFB=-5	Invalid storage area
\r\n	ASCII	2	Terminator	
Remark				

2.10.8 Starting automatical net-connecting

Command

at+auto_connect \ r \ n

Description

This command is used when the module is saving parameters, the module interior automatically connects the internet to complete the scanning, joining and IP allocation. After success, return MAC and IP information

Parameter description

NULL

Return value

Parameter	Format	Length (byte)	Description	
The command is successfully implemented				
OK	ASCII	2	Query succeeded	
<MAC>	HEX	6	Module MAC address	
<IP>	HEX	4	Module IP address	
<NETMASK>	HEX	4	Module subnet mask	
<GATEWAY>	HEX	4	Gateway	
<DNS SERVER1>	HEX	4	DNS server 1	
<DNS SERVER2>	HEX	4	DNS server 2	
\r\n	ASCII	2	Terminator	
Command execution error				
ERROR	ASCII	5	Error	
<CODE>	HEX	1	0XFE =-2	AP not found
			0XFD=-3	Connection failed
			0XFC=-4	Obtaining IP address failed
			0XFB=-5	Invalid storage area
\r\n	ASCII	2	Terminator	
Remark				

2.11 Firmware upgrade

2.11.1 Into the firmware upgrade mode

Command

at+upgrade\r\n

Description

This command is used to set the module in upgrade mode.

1.PC(master MCU) sends,at+upgrade \ r \ n, request in upgrade mode.

2.Until the command is successful, module returns "OK", PC (master MCU) send "u", confirm in upgrade mode.If there is no return "OK", execute the command module return to step 1.

3.Until the module back to "OKC ", using xmodem send module firmware.If there is no return "OKC ", return to step 1-3.

Parameter description

NULL

Example:

Reference3.5.3Upgrade of uart

Description of return value

Parameter	Format	Length (byte)	Description
The command is successfully implemented			
OK	ASCII	2	Starting successfully
\r\n	ASCII	2	Terminator
Failure of command execution			
ERROR	ASCII	5	Error
<CODE>	HEX	1	See ERROR list for details
\r\n	ASCII	2	Terminator
Remark			

3. AT command sample flow

3.1 Test Conditions

Hardware: RAK476 demoboard

micro usb line

Router: TL-WDR4310

Software:

TCP/UDP test tool

Serial port debugging assistant

RAK4X6 Windows Config Tool

RAK476 IOS/Andriod APP

Serial port baud rate setting: 115200

Parity check bit: NULL

Data bit: 8

Stop bit: 1

Flow control: Disable

Note: Three colors respectively represent command header (red), data body (green), terminator (blue)

3.2 Creating AP and Establishing TCP_SEVER

This part gives an example process of AT command, set the module to AP mode, and establish TCP Sever, PC(C) connects to the module AP, and create TCP Client to communicate with the module.

Frequency band: 2.4GHZ

Information channel: channel 1

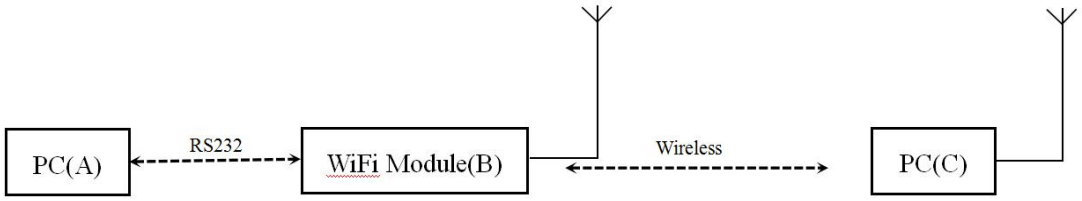
AP Name: rak_ap

AP password: rakwireless

Encryption mode: WAP2-PSK-CCMP

Module IP address: 192.168.9.4

Country code: CN



AT command flow is as follows

Starting-up returns

57 65 6C 63 6F 6D 65 20 74 6F 20 52 41 4B 34 37 33 0D 0A

Setting information channel

Send: at+channel=1\r\n

Return: 4F 4B 0D 0A

Set the AP password

Send: at+psk=rakwireless\r\n

Return: 4F 4B 0D 0A

Creating SSID as the wireless access point of rak_ap

Send: at+ap=rak_ap\r\n

Return: 4F 4B 0D 0A

Setting module static IP as 192.168.9.4

Send: at+ipstatic=192.168.9.4,255.255.255.0,192.168.9.1,0,0\r\n

Return: 4F 4B 0D 0A

Setting the module to automatically set the DHCP SEVER parameter

Send: at+ipdhcp=1\r\n

Return: 4F 4B 0D 0A

The module creates a TCP Sever with local port of 25000

Send: at+ltcp=25000\r\n

Return: 4F 4B 08 0D 0A

When PC is connected to the module ap, the module returns:

61 74 2B 72 65 63 76 5F 64 61 74 61 3D 82 0D 0A

Through PC create target ip as 192.168.9.4, target port as TCP Client of 25000, and connects to TCP Sever created by the module, the module returns:

61 74 2B 72 65 63 76 5F 64 61 74 61 3D 80 00 18 CA 02 09 A8 C0 0D 0A

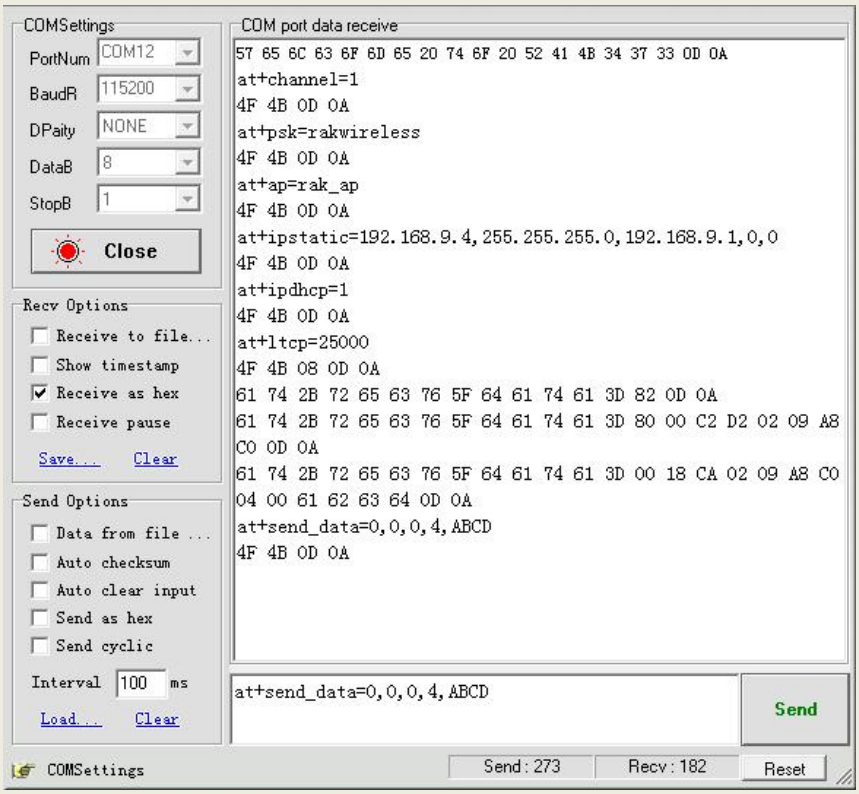
TCP Client sends a string of "ABCD" to TCP Sever, the module returns:

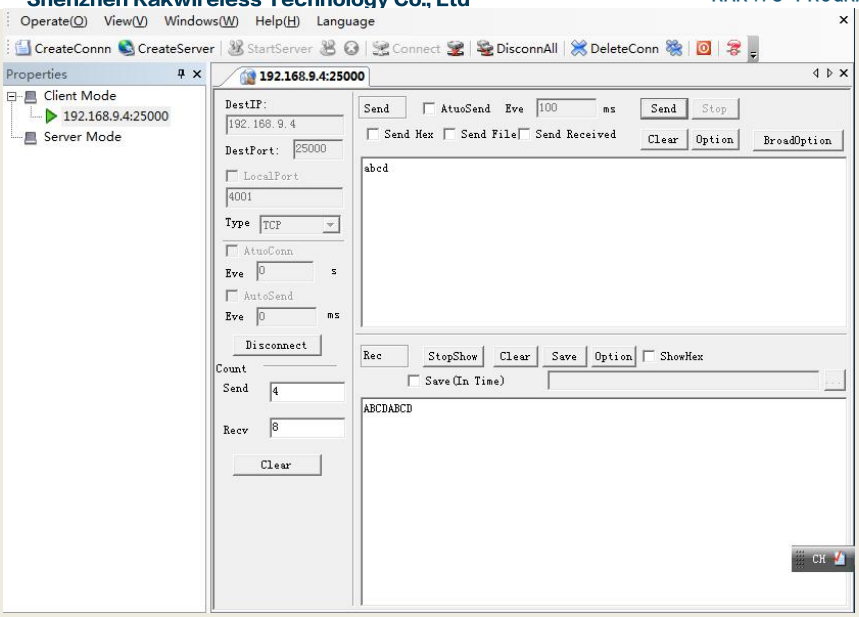
61 74 2B 72 65 63 76 5F 64 61 74 61 3D 00 18 CA 02 09 A8 C0 04 00 61 62 63 64 0D 0A

TCP Sever sends a string of “ABCD” to TCP Client

Send： at+send_data=0,49729,192.168.9.2,4,ABCD\r\n

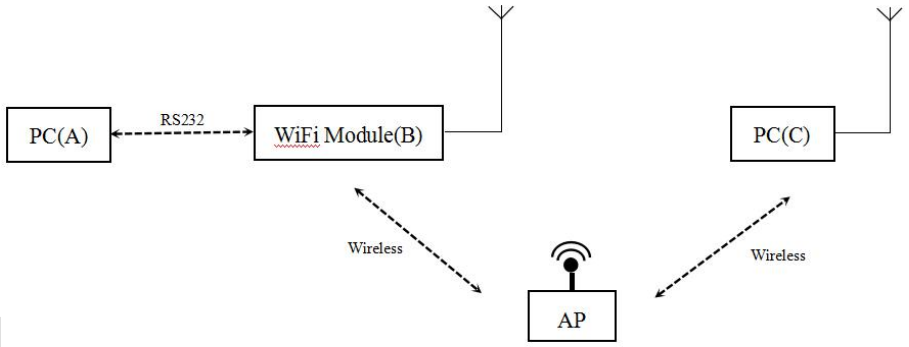
Return： 4F 4B 0D 0A





3.3 Conneting Router(STA) and Establishing TCP Client

This part gives a sample process of AT command, connect the module with the router with SSID of RAK, PSK of rakwireless, then establish TCP sever. PC (C) establishes TCP Client and connects to the module terminal TCP SEVER to transmit data.



Frequency band: 2.4GHZ

AP name: rak_sta

AP password: rakwireless

Encryption mode: WPA2-PSK

IP address of WiFi Moudle (B): DHCP acquiring

RAK_2.4GHz

AT command flow is as follows:

Starting-up returns

57 65 6C 63 6F 6D 65 20 74 6F 20 52 41 4B 34 37 33 0D 0A

Scan wireless network with SSID of rak_sta in all information channels

Send： at+scan=0,rak_sta\r\n

Return： 4F 4B 01 0D 0A

Setting wireless password rakwireless

Send： at+psk=rakwireless\r\n

Return： 4F 4B 0D 0A

The module connects the wireless network with SSID of rak_sta

Send： at+connect=rak_sta\r\n

Return： 4F 4B 0D 0A

Boot up module DHCP Client, obtain module IP address

Send： at+ipdhcp=0\r\n

Return： 4F 4B 9C 44 3D 00 06 52 7F 01 A8 C0 00 FF FF FF 01 01 A8 C0 01 01 A8 C0 00 00 00 00 0D 0A

at PC terminal establish TCP Sever with local port of 9000, and start up the server. The module terminal establishes TCP Client, and then connects to the TCP Sever of PC terminal

Send： at+tcp=192.168.1.106,9000,25000,0\r\n

Return： 4F 4B 00 0D 0A

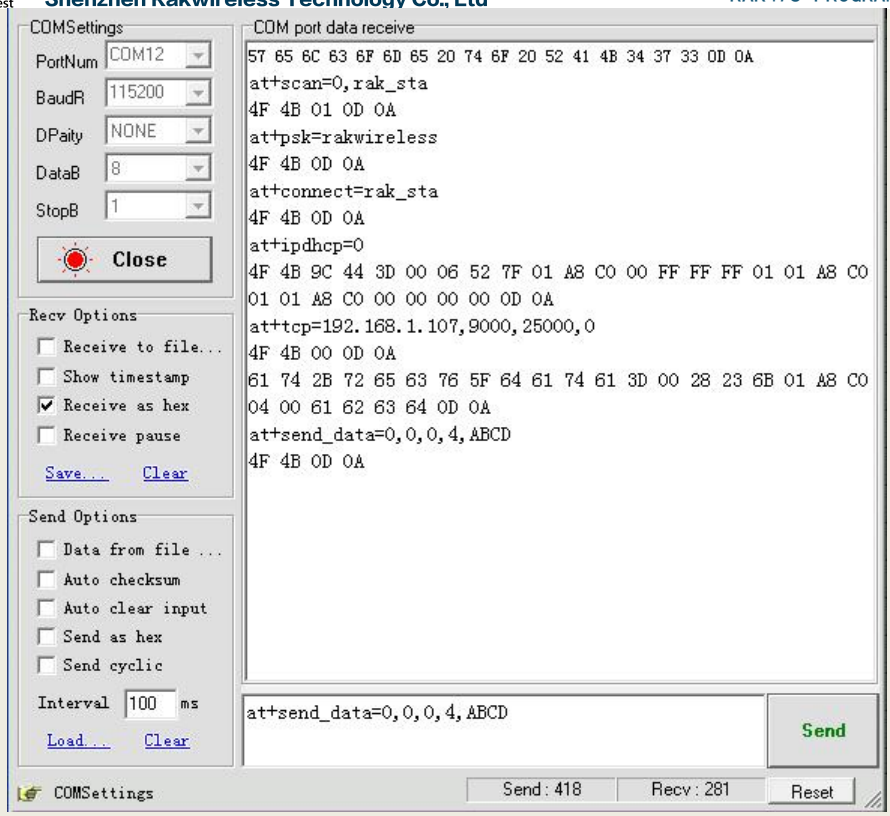
TCP Sever of PC terminal sends a string of "abcd" to TCP Client, the module returns

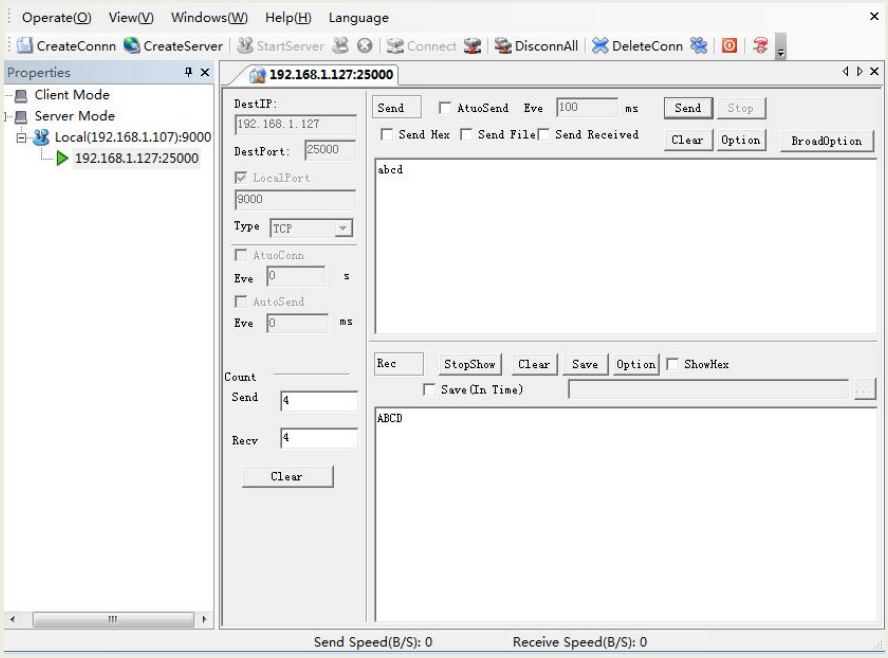
61 74 2B 72 65 63 76 5F 64 61 74 61 3D 00 28 23 6B 01 A8 C0 04 00 61 62 63 64 0D 0A

TCP Client of module terminal sends TCP Sever of PC terminal a string of "ABCD"

Send： at+send_data=0,9000,192.168.1.106,4,ABCD\r\n

Return： 4F 4B 0D 0A





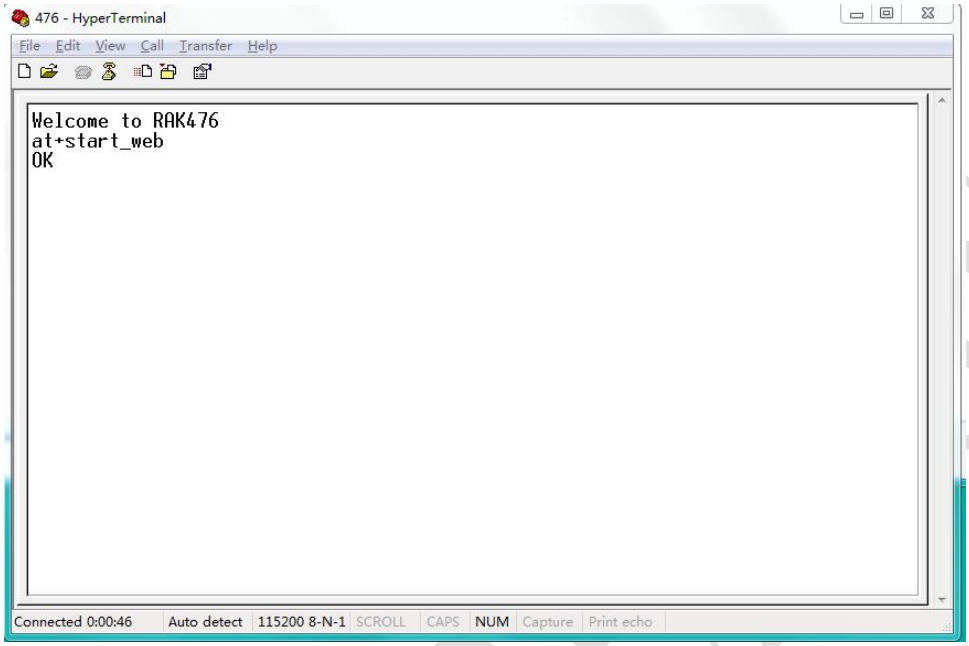
3.4 Network configuration

This part mainly introduces several network configurations of RAK476, RAK476 mainly includes two kinds of network configuration modes, that is, AP and Easyconfig.

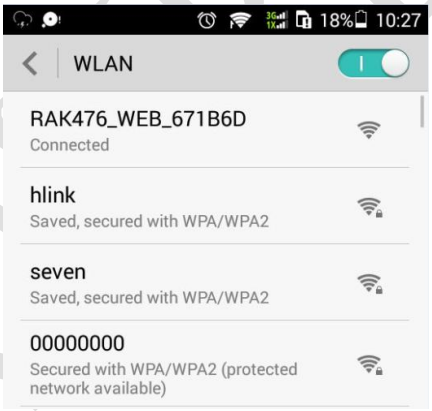
3.4.1 AP network configuration

This section introduces how to use the mobile APP to configure the module to the specified router under the AP mode.

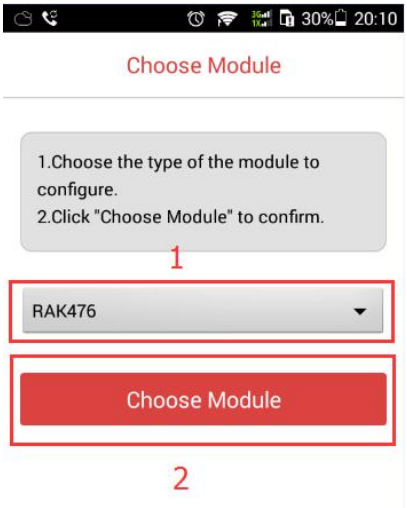
Reset module,Send at+start_web\r\n to establish a hot spot, as follows :



Mobile phone connected to the module to establish a hot spot, as shown below :

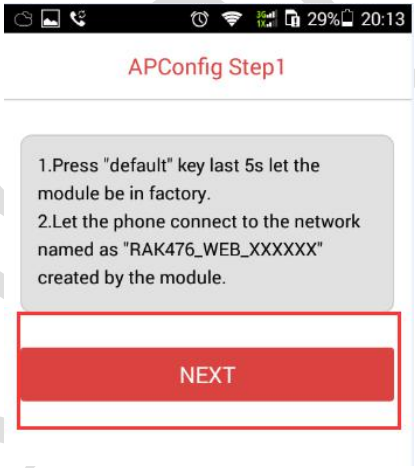


Open the phone APP RAK47XConfigure ,Select module type is RAK476 ,Click Choose Module :

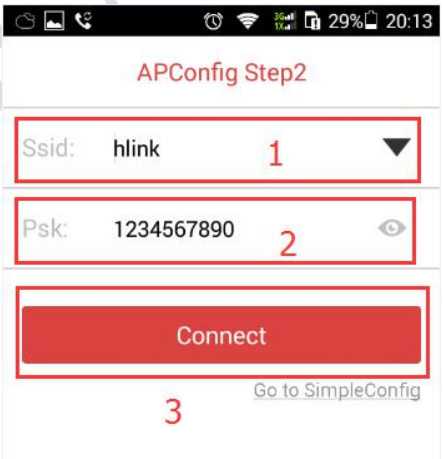


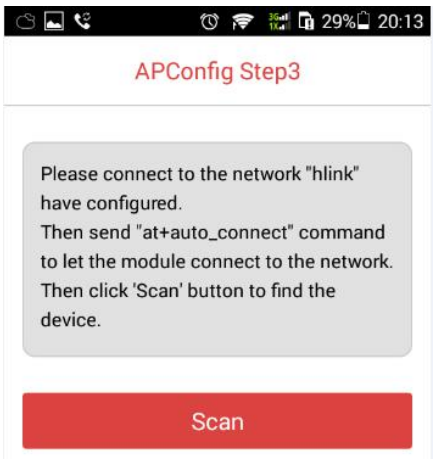
Enter the Config option, select the APConfig configuration :

If the phone has been connected to the module to establish a hot spot, click NEXT :

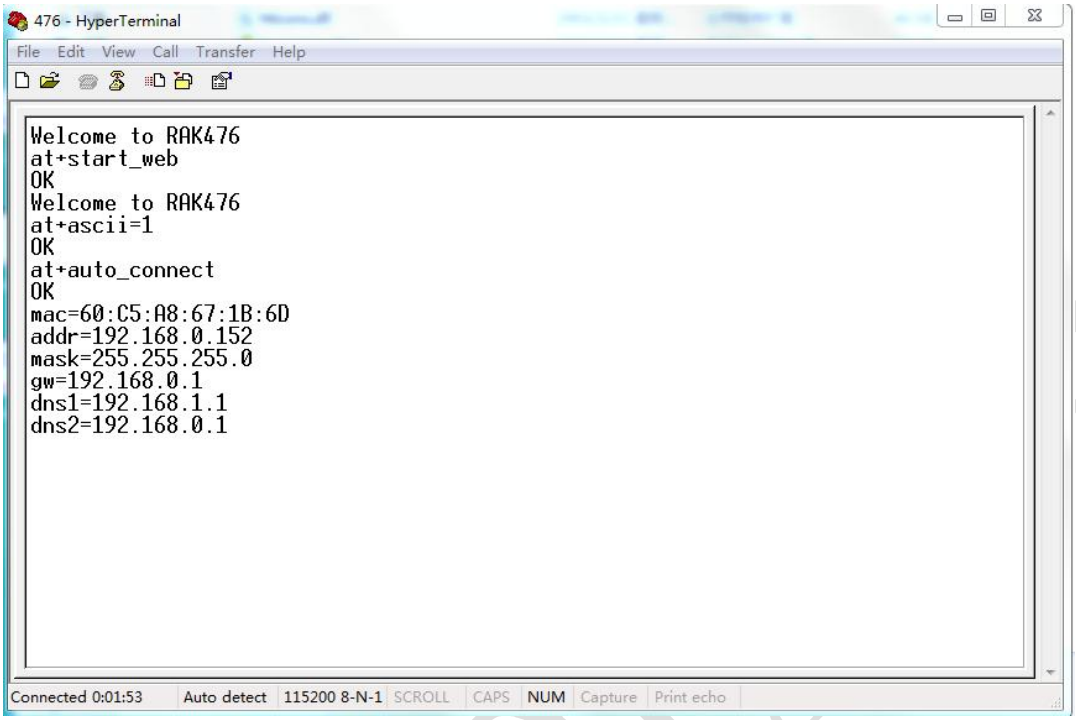


Choose to connect the router, enter the password, click Connect, module received SSID and password automatically saved:

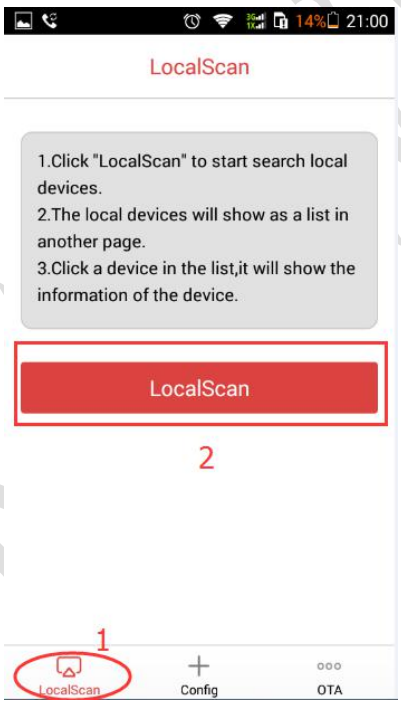


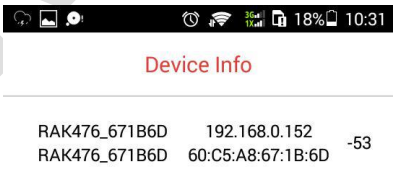


At this point the module will automatically restart, send at+auto_connect\r\n automatic networking :



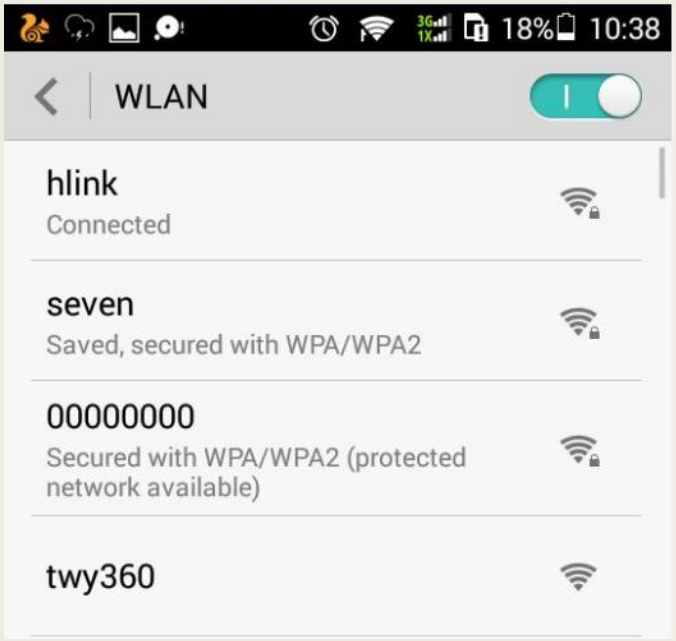
Mobile phone connected to the module connected to the router, select the LocalScan option, scan module :





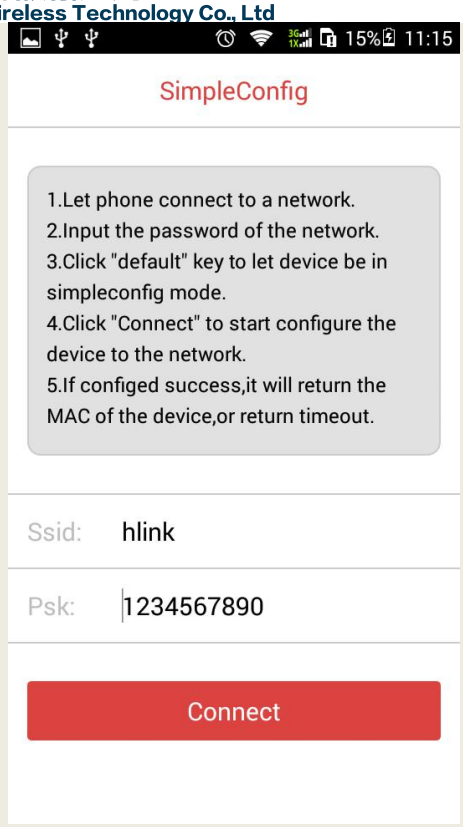
3.4.2 Easyconfig configuration

Mobile phone connected router :

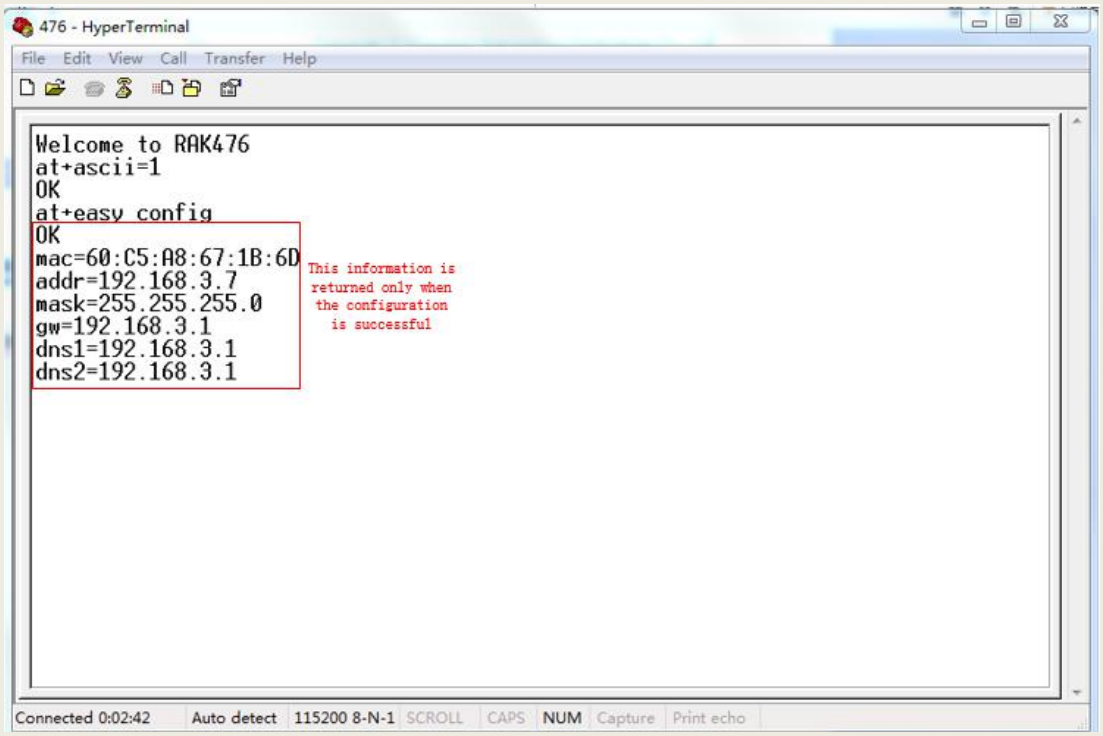


Open the phone APP RAK47XConfigure,Select SimpleConfig options :

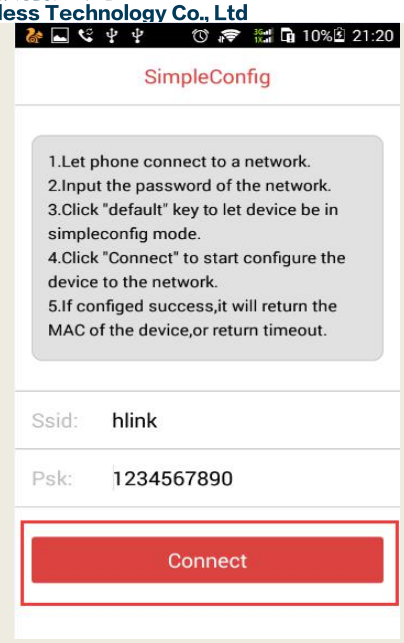
Ssid auto fill, enter Psk :



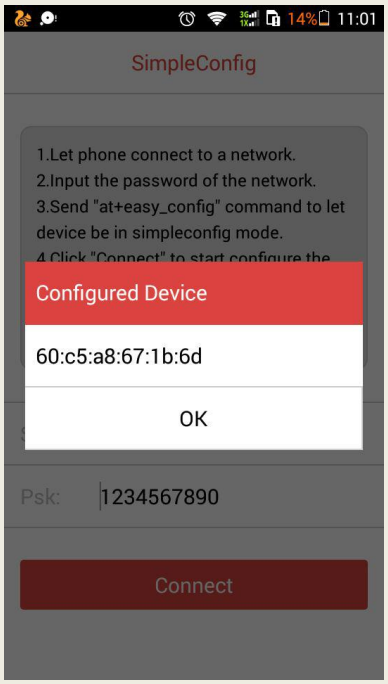
Reset module, send at+easy_config\r\n to allow the module to enter the wait state, the development board link indicator light slow flash :



Click Connect to start configuration :



Configuration successfully returned to the module's MAC address :



3.5 Firmware upgrade

3.5.1 Using mobile APP upgrade

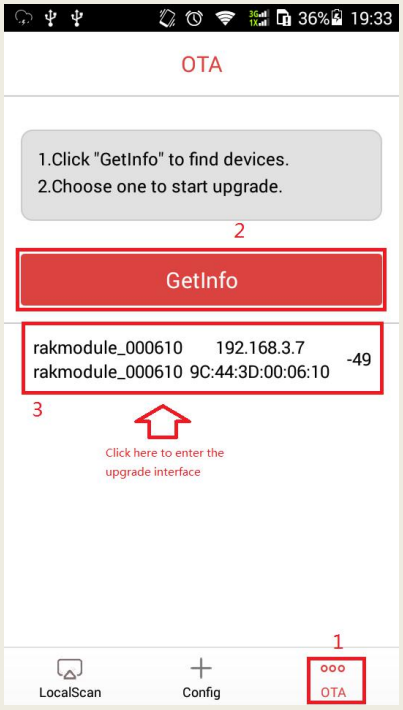
Module support in the AP mode, STA mode upgrade.

The following main introduction module work in the STA mode upgrade, upgrade the firmware can not erase the previous configuration information, so that you can continue to connect before

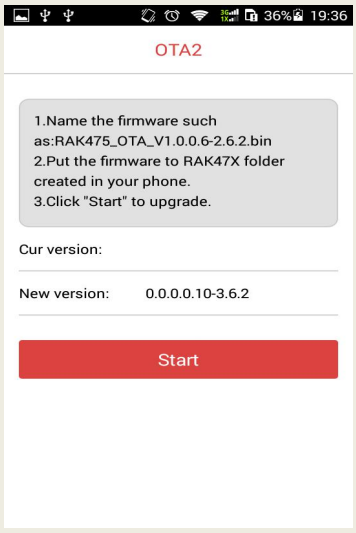
the upgrade configuration after the router, do not need to re configure.

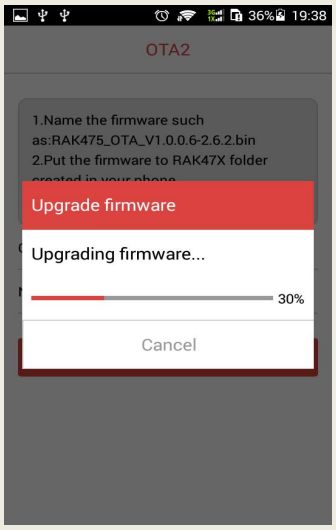
First, make sure that the phone and the module are connected to the same router.

Open the phone APP, select the module type RAK476,Select the OTA option, click CetInfo, access to the module Mac and IP address and other information,Click the module information area, enter the upgrade interface.



Module upgrade firmware saved to the phone RAK47X\RAK476 folder , File name format reference RAK475_OTA_V1.0.0.6-2.6.2.bin,Cur version (STA mode no),New version is Upgrade firmware on mobile phone,Click Start to start the upgrade.





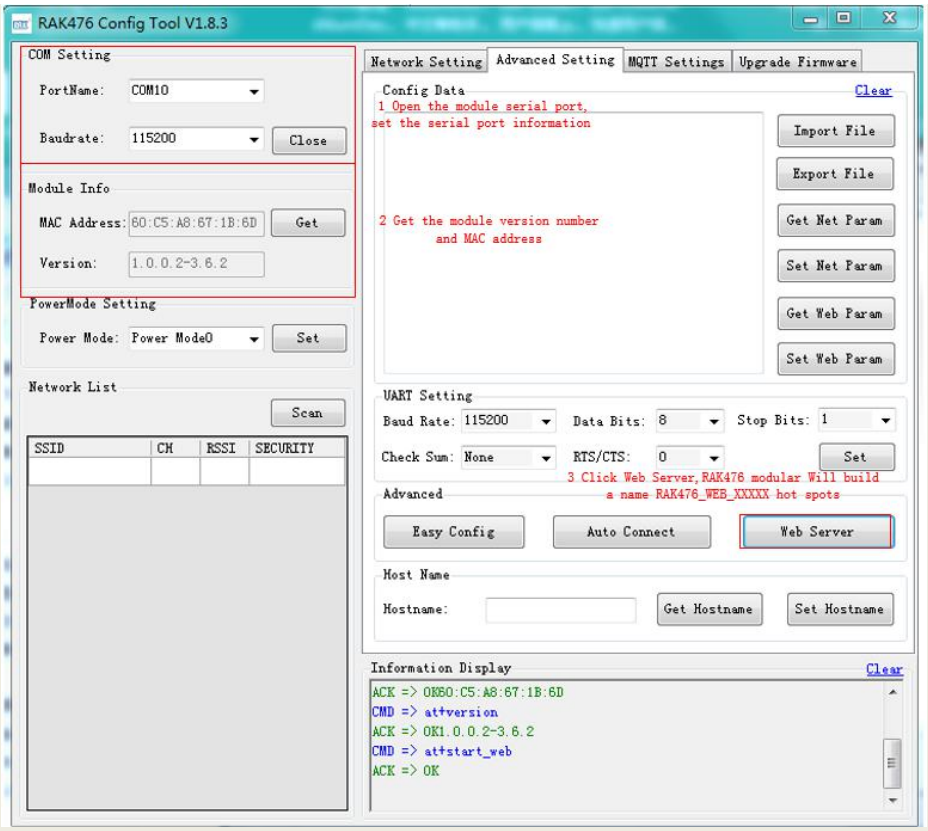
In the upgrade process, Development board link light flashes quickly, stop flashing, said after the flash firmware. Now,Module will automatically restart ,received Welcome to RAK476\r\n ,

Executes the at+version\r\n command to see if the version number is changed.

3.5.2 Upgrade using PC tools RAK476 Config Tool

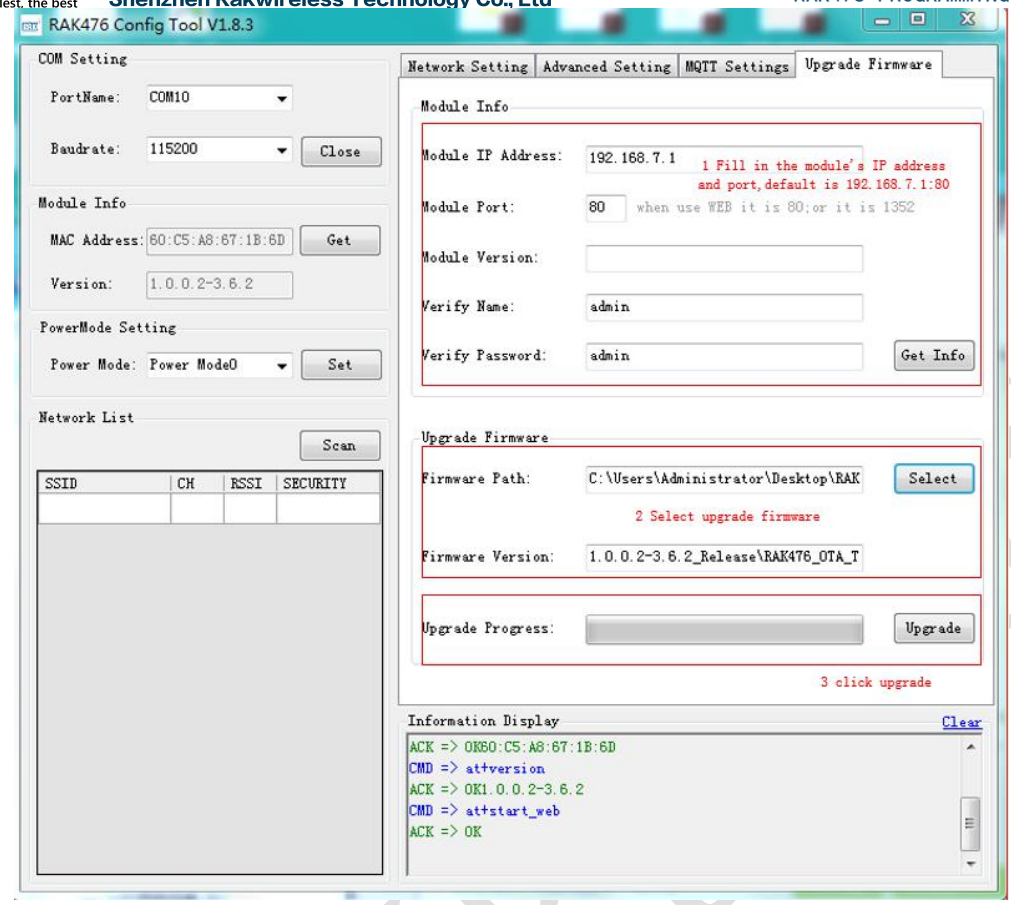
1. AP mode firmware upgrade

Module create hot spots



Computer connected to the RAK476 hot spots

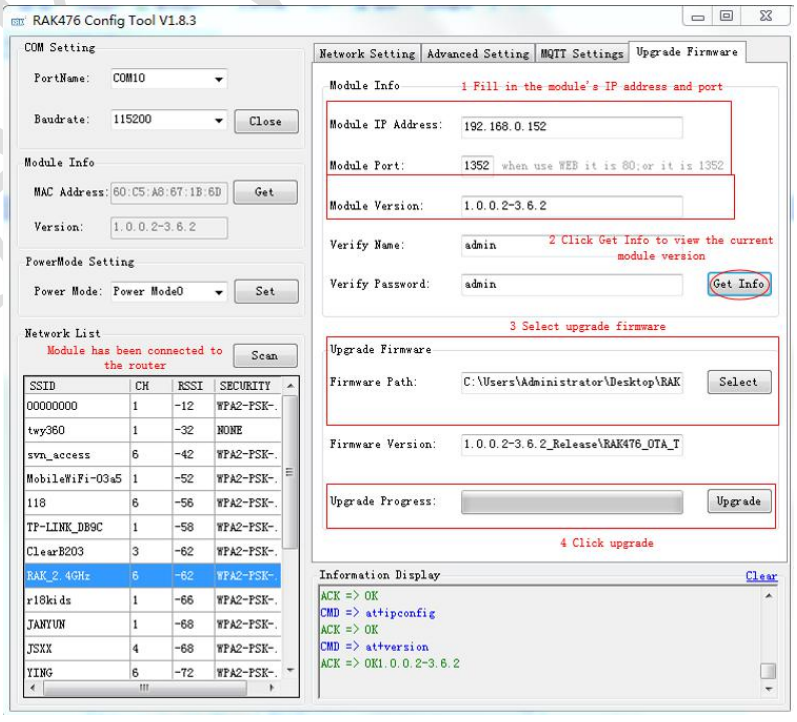
Use tool upgrade settings are as follows



2. STA module firmware upgrade

This section describes how to use the firmware upgrade interface for firmware upgrade,Reference 3.4 network configuration, the module configuration to the specified router.

PC and module are connected to the same router, the module upgrade port is fixed to 1352 in STA mode, and the IP address is obtained from the route randomly.



3.5.3 Upgrade of uart

This section introduces how to use the uart to upgrade the firmware

Starting-up returns :

Welcome to RAK476

Enter the upgrade mode

Send： at+start_web\r\n

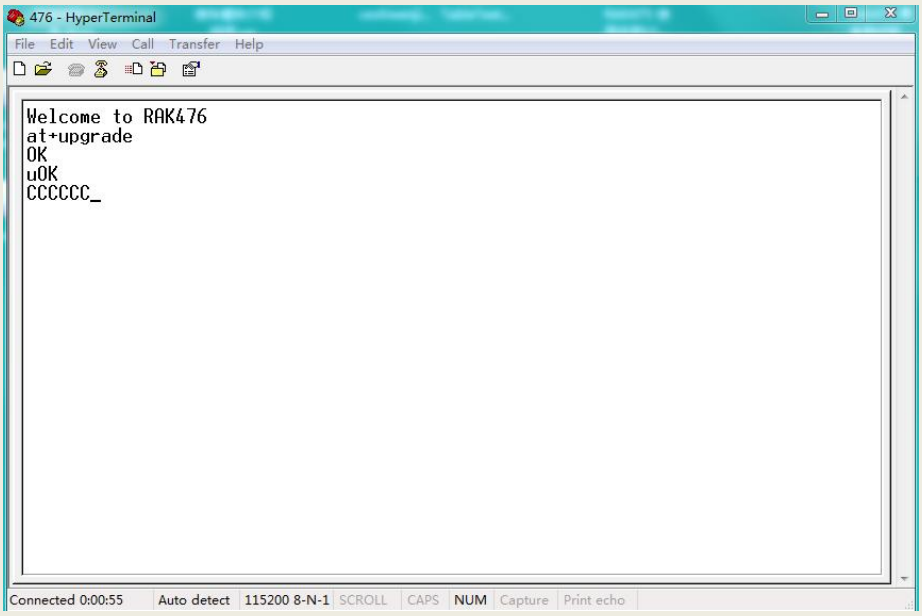
Return： 4F 4B 0D 0A

Send confirmation

Send： u

Return： OK

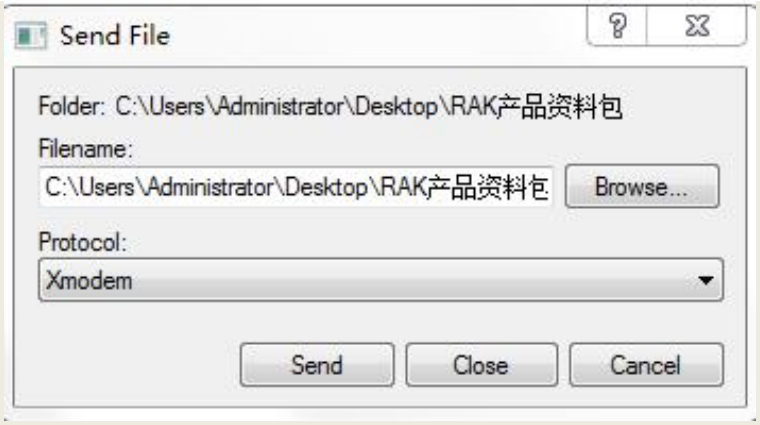
CCCC

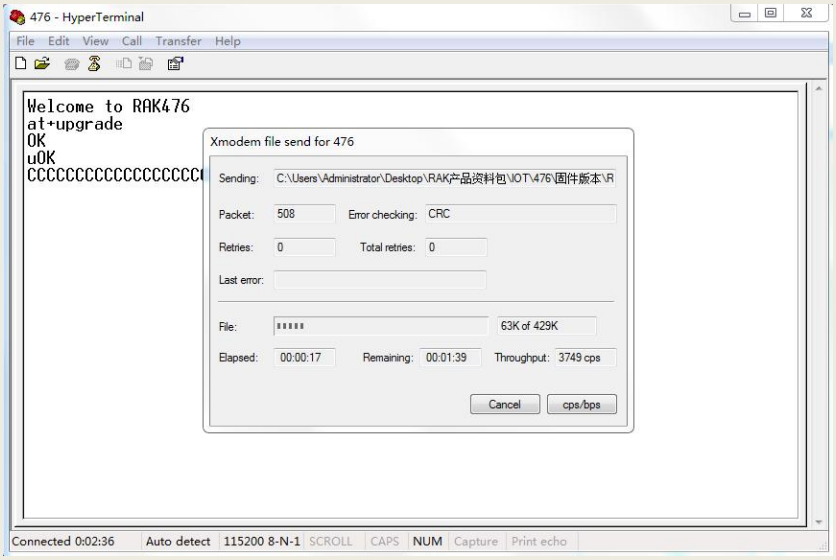


1.Select the file type of Xmodem

2.Choose the need to upgrade the firmware

3.Click "send", start upgrade the firmware





3.6 Saving parameter for fast net-connection

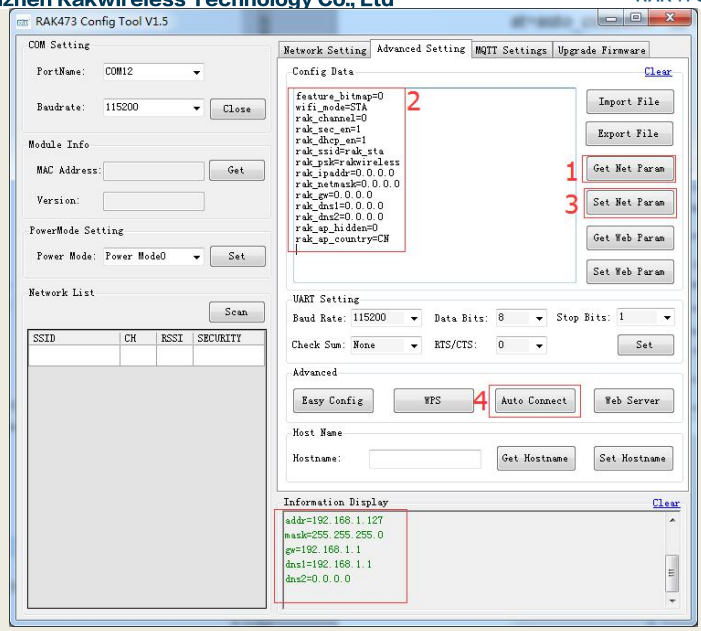
This part introduces how to use tools to save net connecting parameters, every tim the user restarts the module, only needs to invoke "at auto_connect", to quickly connect the module to the designated router.

1, Acquire network parameters

2, Modify network parameters

3, Set network parameters

4, Start atuomatical net-connecting(or send at+auto_connect)



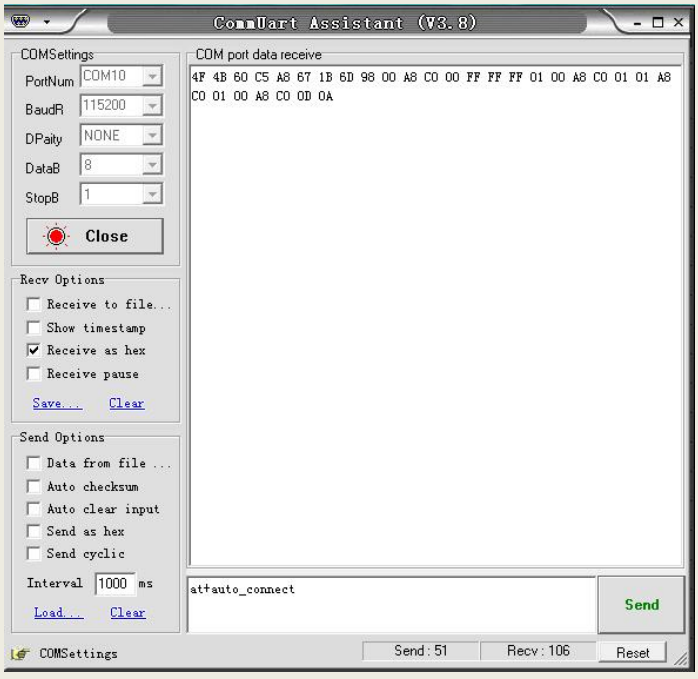
Starting-up returns

57 65 6C 63 6F 6D 65 20 74 6F 20 52 41 4B 34 37 33 0D 0A

Start automatical net-connecting

Send: at+auto_connect

Return: 4F 4B 60 C5 A8 67 1B 6D 98 00 A8 C0 00 FF FF FF 01 00 A8 C0 01 01 A8 C0 01 00 A8 C0 0D 0A



3.7 Examples of tcp client

Refer to 3.3 Creating STA and establishing TCP Client to communicate with PC

3.8 Examples of mqtt

This part gives a sample process of AT command, set the mqtt parameters of module A and B, and then connect module A and B to the Server for communication

In the CMD interface of PC terminal , input "C:\Program Files\mosquitto\mosquitto.exe" -p 1880 -v , and press “enter" key to start a virtual server. (C:\Program Files\mosquitto\mosquitto.exe represents the file path of "mosquitto.exe" , and 1880 indicates the port number of the Server)

Module A parameters

clientId:DXOE

alive:30

username:dhiogaekd

password:dsalhgdsgg

svr_ip:192.168.1.113

svr_port:1880

sub_topic:modula

pub_topic:modulb

Module B parameters

clientId:CSCJ

alive:30

username:shgdsgklg

password:ddkahglehg

svr_ip:192.168.1.113

svr_port:1880

sub_topic:modulb

pub_topic:modula

The specific operating procedures are as follows:

Operation procedure of Module A

Starting-up returns

57 65 6C 63 6F 6D 65 20 74 6F 20 52 41 4B 34 37 33 0D 0A

Refer to 3.4 Network Configuration to configure the module to the specified router

Initialization of mqtt parameter

Send: at+mqtt_init=DXOE,30\r\n

Return: 4F 4B 0D 0A

Setting authentication parameters

Send： at+mqtt_auth=dhiogaekb,dsalhgdsgg\r\n

Return： 4F 4B 0D 0A

Connecting the Server

Send： at+mqtt_con=192.168.1.113,1880\r\n

Return： 4F 4B 0D 0A

Set subscription themes

Send: at+mqtt_sub=moduleaat+mqtt_sub=modulea\r\n

Return： 4F 4B 0D 0A

Setting push topics

Send： at+mqtt_pub=moduleb,1\r\n

Return： 4F 4B 0D 0A

Operation procedure of Module A

Configure parameters for module B according to the parameter of module B and refering to the configuration mode of module A.

Module A with push theme of "moduleb" pushes data to module B with subscription

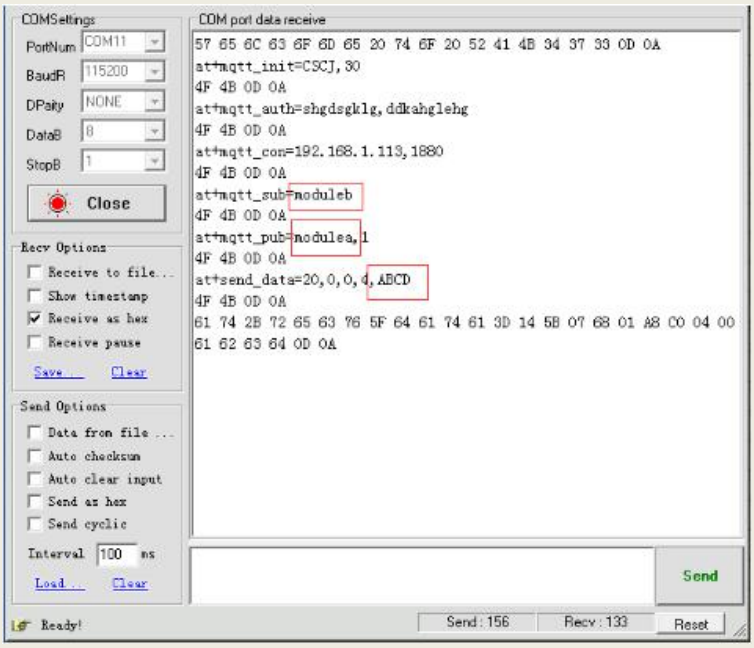
theme of "moduleb"

Send： at+send_data=20,0,0,4,abcd\r\n

Return： 4F 4B 0D 0A

Module B received:

61 74 2B 72 65 63 76 5F 64 61 74 61 3D 00 28 23 6B 01 A8 C0 04 00 61 62 63 64 0D 0A



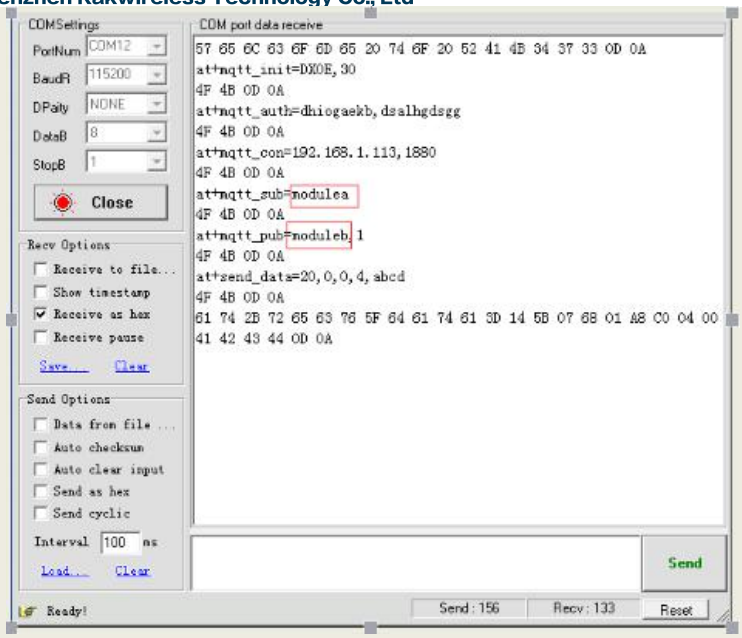
Module B with push theme of "modulea" pushes data to module A with subscription theme of "modulea"

Send： at+send_data=20,0,0,4,ABCD\r\n

Return： 4F 4B 0D 0A

Module A received:

61 74 2B 72 65 63 76 5F 64 61 74 61 3D 00 BB 01 66 01 A8 C0 06 00 41 42 43 44



3.9 Examples of data mode

This part introduces how to enter the data mode to conduct socket data transmission, and to actively exit the mode data.

The specific operating procedures are as follows:

Refer to 3.3Conneting Router(STA) and Establishing TCP Client

Enter the data mode

Send： at+data_mode\r\n

Return： 4F 4B 0D 0A

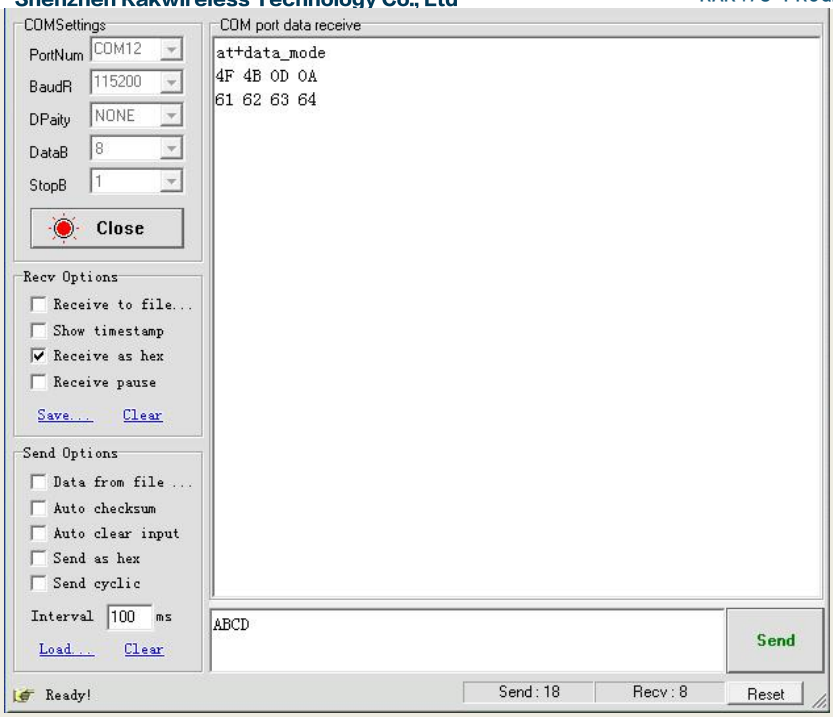
TCP Sever of PC terminal sends a string of "abcd" to TCP Client, the module returns

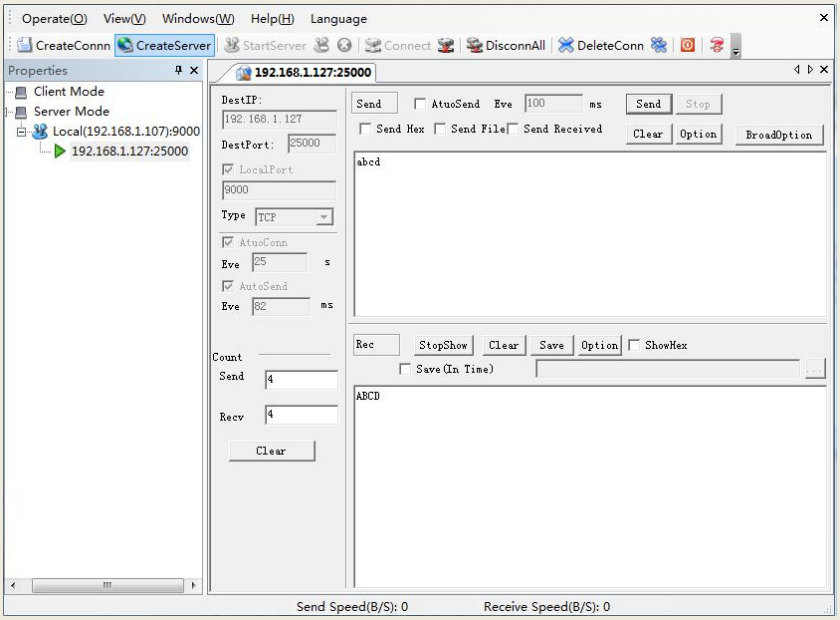
61 62 63 64

TCP Client of module terminal sends TCP Sever of PC terminal a string of "ABCD"

TCP Sever of PC terminal received "ABCD"

You can comapare this format of sending and receiving data with the format of sending and receiving data in 3.3Conneting Router(STA) and Establishing TCP Client, to see the effect of data mode.





Request to exit the data mode

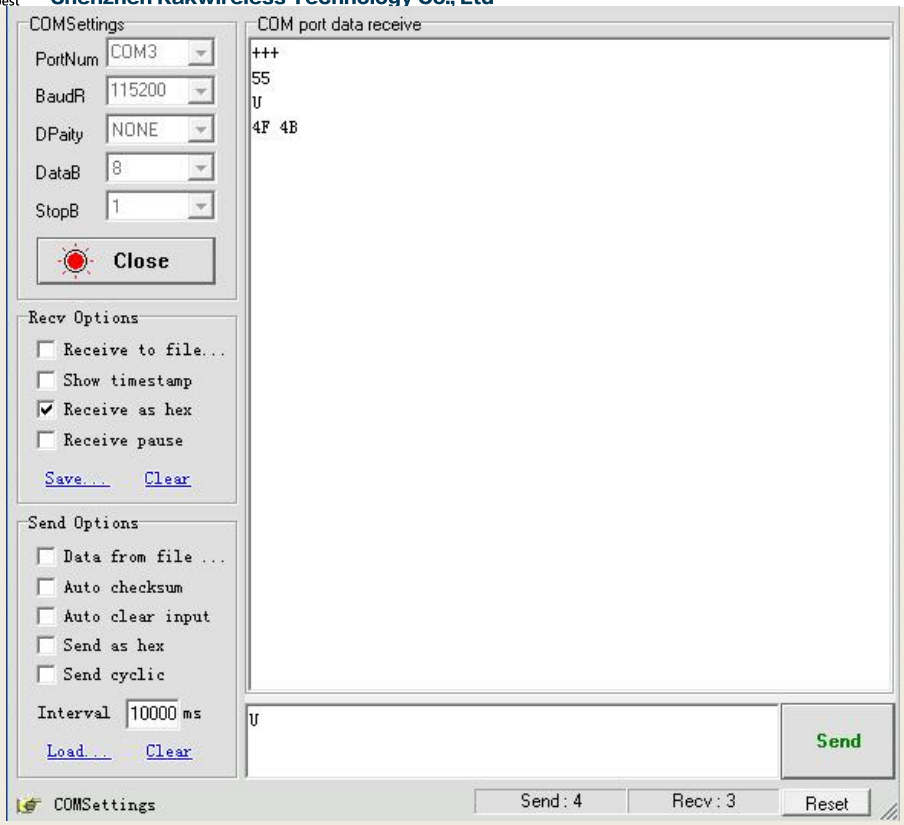
Send： +++

Return： 55

Within 3S send "U" to the module to confirm the exit of data mode

Send： U

Return： 4F 4B



3.10 Reading data through way of query

This part gives a sample procedure of the AT command, after the connection of TCP Client established in the module terminal and TCP Sever established in PC terminal, the module terminal reads the data sent by TCP Sever of PC terminal through way of query.

The specific operating procedures are as follows:

Set enable query to return, after the setting, reset the module to take effect.

Send： at+set_funcbitmap=1

Return： 4F 4B 0D 0A

Reset return

57 65 6C 63 6F 6D 65 20 74 6F 20 52 41 4B 34 37 33 0D 0A

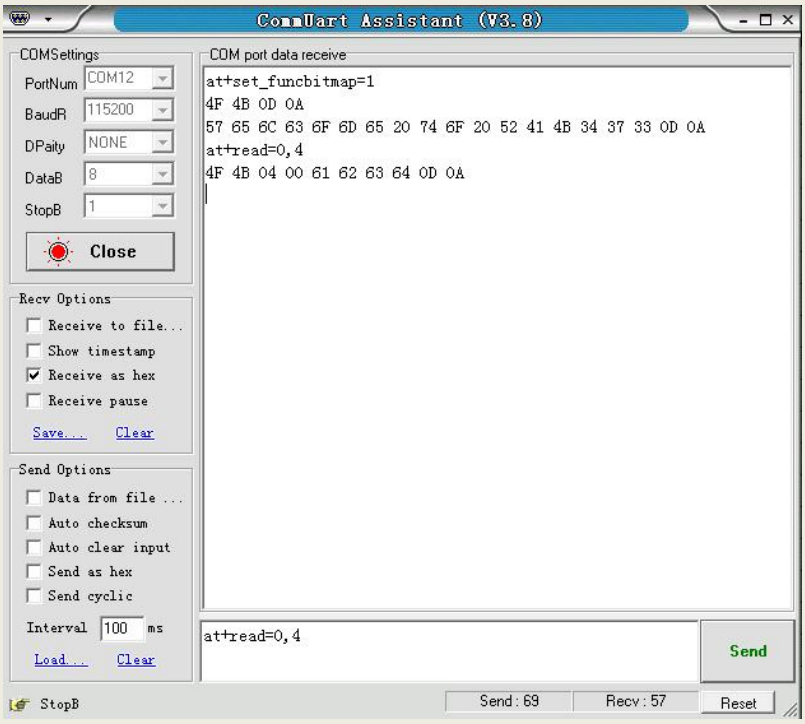
Refer to 3.3Conneting Router(STA) and Establishing TCP Client

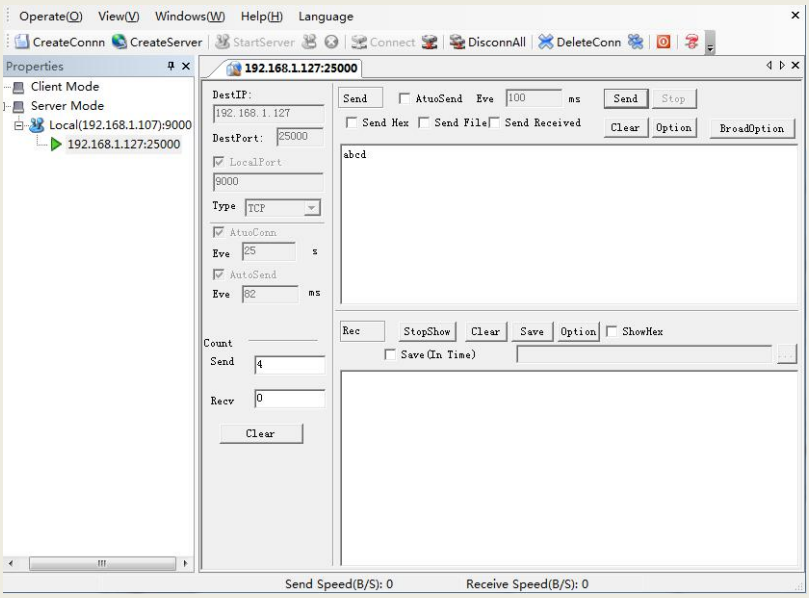
TCP Sever of PC terminal sends a string of "abcd" to TCP Client, the module terminal sends a

command to read data

Send： at+read=0,4\r\n

Return： 4F 4B 04 00 61 62 63 64 0D 0A





4. Sale and service

Shanghai

FAE allan.jin@rakwireless.com

Tel：021- 61553990

Fax：021- 54721038

Address: Room B205, Green light kechuang garden, 2588 Lane, Hongmei South road, Minhang District, Shanghai

Shenzhen

FAE steven.tang@rakwireless.com

Tel: 0755- 26506594

Fax: 0755- 86152201

Address: 802 Yongfu Building, No. 106 Yongfu Road, Baoan District, Shenzhen

5. Modification records

Version	Date	Modification content
V1.0	2015/12/1	Creating document
V1.1	2016/5/10	Modify the flow control description, change to enable and disable
V1.2	2016/6/30	New additions to MQTT、configuration store UART parameter command ，modify some errors