GigaDevice Semiconductor Inc.

GD32W51x Base Command User Guide

Application Note
AN081

Table of Contents

List of Figures

List of Tables

1. Basic user commands

Connect the test machine to the development board by using a USB cable, open the UART tool, and connect to the correct COM port. After the development board is powered on and started correctly, commands are transmitted through the UART tool, and the development board can complete corresponding operations according to the commands.

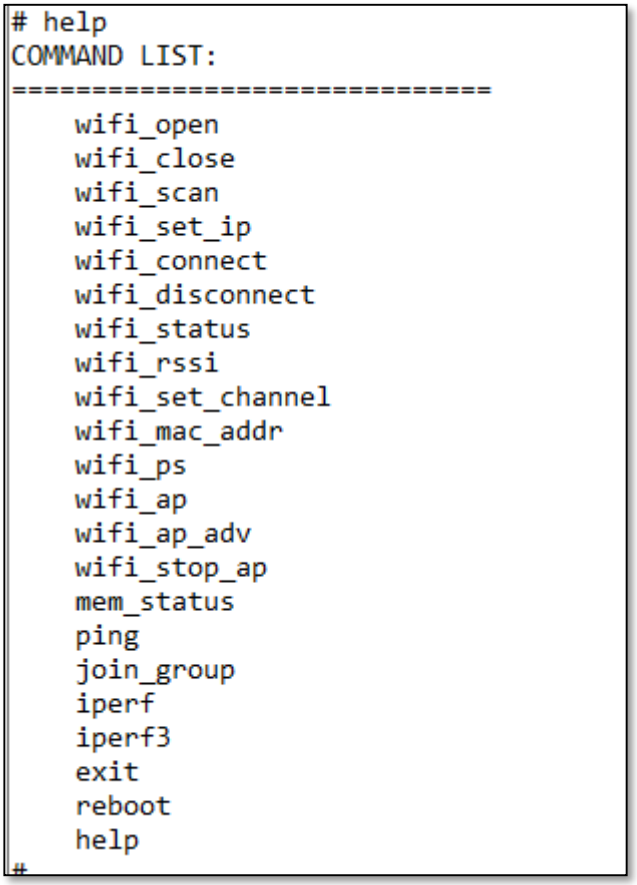
In this manual, < > indicates that this option is mandatory, and [] indicates that this option is optional. Note that commands are strictly distinguished between uppercase and lowercase letters.

1.1. help

This command has no options.

As shown in ***Figure 1-1. Help command***, The help command will list all the commands supported by the development board.

Figure 1-1. Help command



1.2. wifi_open

This command has no options.

This command is used to enable wifi. Other wifi-related commands take effect only when wifi is enabled. After the development board is started correctly, wifi is turned on by default, so there is no need to execute this command to turn on wifi repeatedly. This command is usually paired with wifi_close to turn wifi back on after wifi_close has turned it off.

As shown in ***Figure 1-2. wifi_open command***, after wifi is closed, execute wifi_open, wifi will be opened, and MAC address will be printed through the serial port; If wifi is enabled, the serial port will prompt that wifi is enabled.

Figure 1-2. wifi_open command

```
wifi_open
WiFi SW init OK.
WiFi RF init OK.
WiFi BB config OK.
WiFi RF calibration OK.
WiFi MAC address: 76:ba:ed:1e:00:1d
wifi netlink: device opened!
#
# wifi_open
wifi device had been opened!
```

1.3. wifi_close

This command has no options.

This command disables wifi. After that, some commands, such as wifi_scan and wifi_connect, cannot be executed.

Under different circumstances, the execution results of commands are different, as follows:

- If the development board is connected to the AP, the development board will be disconnected from the AP, and then the wifi will be turned off.
- If the development board is not connected to the AP, turn off the wifi directly;
- If the development board is in softAP mode and STA is connected to the development board, the connection will be disconnected and the wifi will be turned off.
- If the development board is in softAP mode, and no STA connection, then directly close the wifi;
- If wifi is disabled, the serial port will prompt that wifi is disabled.

1.4. reboot

This command has no options.

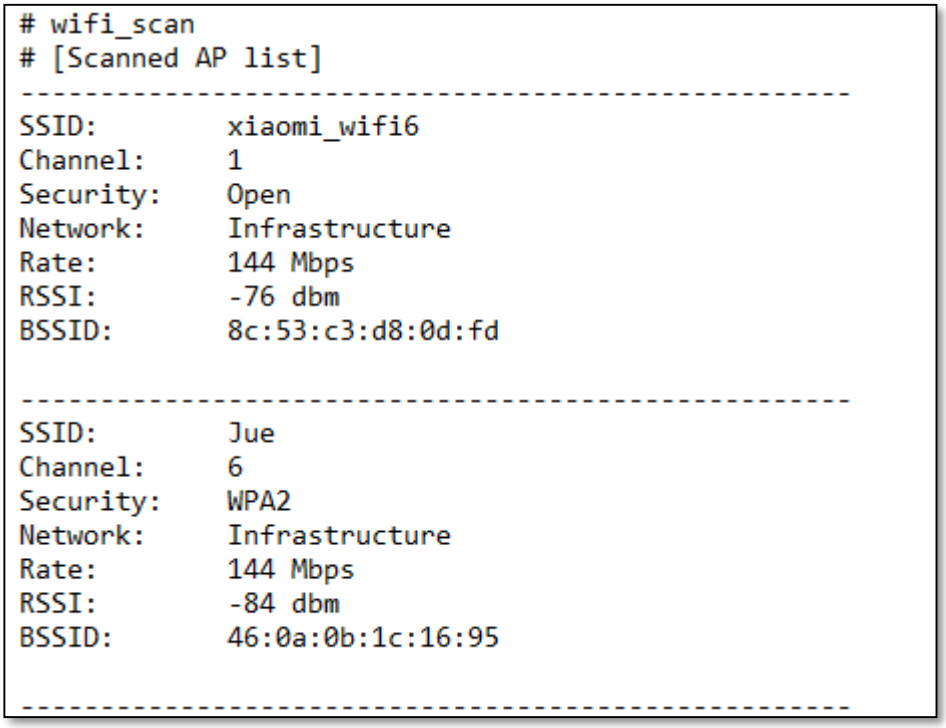
After this command is executed, the development board restarts, and the serial port displays startup information. This command is similar to the reset button.

1.5. wifi_scan

This command has no options.

After this command is executed, the serial port prints the AP information scanned by the development board, As shown in ***Figure 1-3. wifi scan command***, It includes SSID, channel, encryption mode, Network (BSS type), Rate, RSSI and BSSID.

Figure 1-3. wifi_scan command



1.6. wifi_mac_addr

- Usage: wifi_mac_addr [MAC address]

This command is used to display or temporarily change MAC addresses, the temporary change address becomes invalid after the chip is powered off or reset.

- wifi_mac_addr

The serial port will print the current MAC address of the development board.

- wifi_mac_addr <MAC address>

<MAC address> is a temporary MAC address in the format of 11:22:33:aa: bb :cc, for example:

- wifi_mac_addr 76:ba:ed:12:13:14

Before EFUSE is configured with a MAC address, the wifi MAC address in the SDK is fixed. If multiple development boards are tested at the same time, MAC addresses may conflict with each other. In this case, this command could change MAC addresses temporarily.

1.7. wifi_connect

- Usage: wifi_connect <SSID> [PASSWORD]

When this command is used to connect to the AP, the development board must be in station mode (the default mode of the development board).

- wifi_connect <SSID>

Used to connect to an AP that is not encrypted.

- wifi_connect <SSID> <PASSWORD>

Used to connect to an AP that is encrypted.

The connection process is shown in ***Figure 1-4. wifi_connect command***. The serial port prints the connection process information, there is one more handshake interaction when connecting to an encrypted AP than an unencrypted AP. At the same time, if the wifi_connect command is executed when the AP is already connected, the development board will disconnect from the original AP and then connect to the new AP.

Figure 1-4. wifi_connect command

```
# wifi_connect tplink886
# STA: Auth Request sent with algm 0x00 and seq 1.
STA: Auth response received with status 0.
STA: Assoc Request sent to 80:89:17:c2:e2:72.
STA: Assoc Response received with status 0.
wifi netlink: indicate connect, link_status is 2.
wifi netlink: connected to ap: tplink886
wifi netlink: Got IP 192.168.1.189

# wifi_connect totolink_n150_2 12345678
STA: Send Deauth to AP with reason 3.
STA: Indicate disconnect.
Disconnect from up layer
wifi netlink: disconnect with ap tplink886
# STA: Auth Request sent with algm 0x00 and seq 1.
STA: Auth response received with status 0.
STA: Assoc Request sent to b8:55:10:49:93:6c.
STA: Assoc Response received with status 0.
STA: Receive Eapol 4-1.
STA: Send Eapol 4-2.
STA: Receive Eapol 4-3.
STA: Set PTK to HW for b8:55:10:49:93:6c
STA: Send Eapol 4-4.
wifi netlink: indicate connect, link_status is 2.
wifi netlink: connected to ap: totolink_n150_2
wifi netlink: Got IP 192.168.0.83
```

1.8. wifi_disconnect

This command has no options.

After this command is executed, the development board will be disconnected from the AP.

1.9. wifi_status

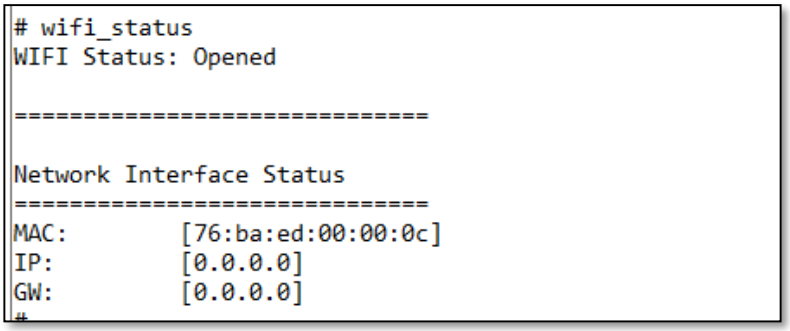
This command has no options.

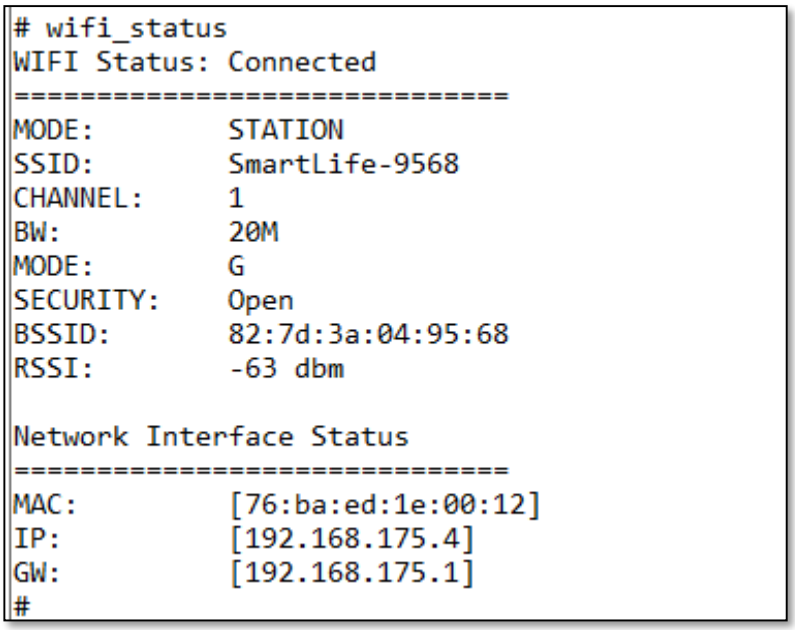
After this command is executed, the serial port will print the wifi status of the current development board. As shown in ***Figure 1-5. wifi status command***, it is divided into two parts, WIFI Status and Network Interface Status.

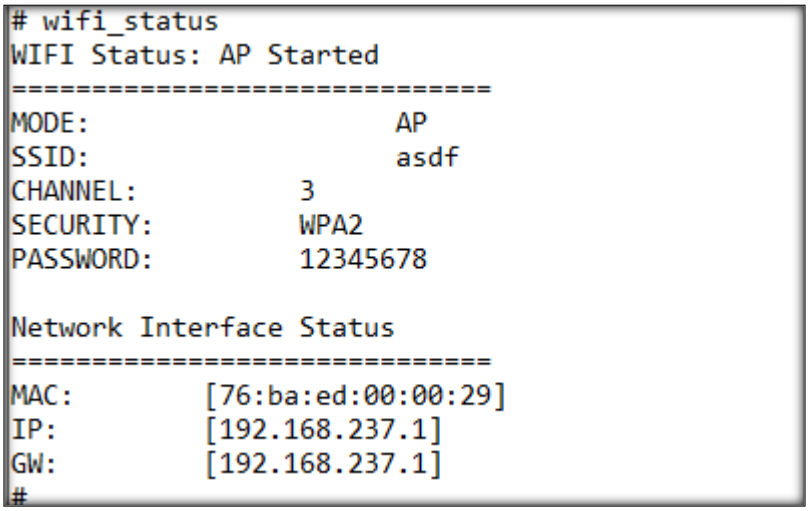
There are four status of WIFI Status: Closed (WIFI is closed), Opened (WIFI is opened), Connected (AP is connected), and AP Started (softAP mode is enabled, the default mode is station mode). The four status add different information. Connected indicates information about connected APs, including SSID, channel, and bandwidth. AP Started indicates

information about its AP. Opened and Closed have no additional information. Network Interface Status indicates the MAC address, IP, and Gateway of the development board. The last two values are valid only when the WIFI Status is Connected and AP Started.

Figure 1-5. wifi_status command







1.10. wifi_rssi

This command has no options.

The real-time RSSI value of the AP connected to the development board can be obtained by executing this command. Therefore, the AP needs to be connected before executing this command.

1.11. mem_status

This command has no options.

After this command is executed, the serial port prints the current system memory status.

1.12. exit

This command has no options.

After this command is executed, the system leaves the command operation mode. Need to Wait 10 seconds before entering again and continuing to execute commands.

1.13. ping

- Usage: ping <target_ip | stop> [-n count] [-l size] [-i interval] [-t total time]

This command is used to perform a ping test.

In the parameters of the command, count is the number of ping packets. size indicates the packet length in byte. interval is the interval for sending packets in ms. total time Indicates the total running time in seconds. By default, count is 5, size is 120, interval is 10, and total time is not used. If you use the total time option, the count and interval options do not work. interval defaults to 1000, and count is the same as total time.

The usage of ping is shown in ***Figure 1- 6. ping command***.

Figure 1- 6. ping command

```
16:04:22.596  # ping 192.168.1.1
16:04:22.599  # [ping_test] PING 192.168.1.1 120 bytes of data
16:04:22.647  [ping_test] 120 bytes from 192.168.1.1: icmp_seq=1 time=19 ms
16:04:22.648  [ping_test] 120 bytes from 192.168.1.1: icmp_seq=2 time=1 ms
16:04:22.649  [ping_test] 120 bytes from 192.168.1.1: icmp_seq=3 time=2 ms
16:04:22.698  [ping_test] 120 bytes from 192.168.1.1: icmp_seq=4 time=4 ms
16:04:22.700  [ping_test] 120 bytes from 192.168.1.1: icmp_seq=5 time=1 ms
16:04:22.702  [ping_test] 5 packets transmitted, 5 received, 0% packet loss
16:04:22.703  [ping_test] delay: min 1 ms, max 19 ms, avg 5 ms
16:04:23.769
16:04:31.693  # ping 192.168.1.1 -n 3
16:04:31.694  # [ping_test] PING 192.168.1.1 120 bytes of data
16:04:31.697  [ping_test] 120 bytes from 192.168.1.1: icmp_seq=1 time=1 ms
16:04:31.698  [ping_test] 120 bytes from 192.168.1.1: icmp_seq=2 time=1 ms
16:04:31.702  [ping_test] 120 bytes from 192.168.1.1: icmp_seq=3 time=1 ms
16:04:31.742  [ping_test] 3 packets transmitted, 3 received, 0% packet loss
16:04:31.743  [ping_test] delay: min 1 ms, max 1 ms, avg 1 ms
16:04:32.457
16:04:39.214  # ping 192.168.1.1 -n 3 -l 1000
16:04:39.217  # [ping_test] PING 192.168.1.1 1000 bytes of data
16:04:39.218  [ping_test] 1000 bytes from 192.168.1.1: icmp_seq=1 time=1 ms
16:04:39.265  [ping_test] 1000 bytes from 192.168.1.1: icmp_seq=2 time=1 ms
16:04:39.266  [ping_test] 1000 bytes from 192.168.1.1: icmp_seq=3 time=1 ms
16:04:39.270  [ping_test] 3 packets transmitted, 3 received, 0% packet loss
16:04:39.272  [ping_test] delay: min 1 ms, max 1 ms, avg 1 ms
16:04:39.826
16:05:02.193  # ping 192.168.1.1 -n 3 -l 500 -i 5000
16:05:02.194  # [ping_test] PING 192.168.1.1 500 bytes of data
16:05:02.196  [ping_test] 500 bytes from 192.168.1.1: icmp_seq=1 time=1 ms
16:05:07.231  [ping_test] 500 bytes from 192.168.1.1: icmp_seq=2 time=6 ms
16:05:12.209  [ping_test] 500 bytes from 192.168.1.1: icmp_seq=3 time=3 ms
16:05:12.211  [ping_test] 3 packets transmitted, 3 received, 0% packet loss
16:05:12.215  [ping_test] delay: min 1 ms, max 6 ms, avg 3 ms
16:05:15.208
16:11:03.842  # ping 192.168.1.1 -n 3 -l 500 -i 5000 -t 5
16:11:03.844  # [ping_test] PING 192.168.1.1 500 bytes of data
16:11:03.845  [ping_test] 500 bytes from 192.168.1.1: icmp_seq=1 time=8 ms
16:11:04.859  [ping_test] 500 bytes from 192.168.1.1: icmp_seq=2 time=3 ms
16:11:05.876  [ping_test] 500 bytes from 192.168.1.1: icmp_seq=3 time=1 ms
16:11:06.843  [ping_test] 500 bytes from 192.168.1.1: icmp_seq=4 time=1 ms
16:11:07.860  [ping_test] 500 bytes from 192.168.1.1: icmp_seq=5 time=1 ms
16:11:07.861  [ping_test] 5 packets transmitted, 5 received, 0% packet loss
16:11:07.867  [ping_test] delay: min 1 ms, max 8 ms, avg 2 ms
```

- ping stop

This command is used to stop ping test, as shown in ***Figure 1-7. ping stop command***.

Figure 1-7. ping stop command

```
# ping 192.168.1.1 -n 3 -l 500 -i 5000 -t 50
# [ping_test] PING 192.168.1.1 500 bytes of data
[ping_test] 500 bytes from 192.168.1.1: icmp_seq=1 time=1 ms
[ping_test] 500 bytes from 192.168.1.1: icmp_seq=2 time=1 ms
[ping_test] 500 bytes from 192.168.1.1: icmp_seq=3 time=1 ms
[ping_test] 500 bytes from 192.168.1.1: icmp_seq=4 time=1 ms
ping stop
# [ping_test] 4 packets transmitted, 4 received, 0% packet loss
[ping_test] delay: min 1 ms, max 1 ms, avg 1 ms
```

1.14. wifi_set_channel

- Usage: wifi_set_channel <channel> [bandwidth] [offset]

This command is used to set the channel that the development board listens to. The usage method is shown in ***Figure 1-8. wifi_set_channel command***.

- wifi_set_channel <channel>

Set only channel. The default value of bandwidth is 0 (20M) and offset is 0.

- wifi_set_channel <channel> <bandwidth>

Set channel and bandwidth, offset according to the former two adaptive settings.

- wifi_set_channel <channel> <bandwidth> <offset>

Set channel, bandwidth, and offset.

Figure 1-8. wifi_set_channel command

```
# wifi_set_channel
Usage: wifi_set_channel <channel> [bandwidth] [offset]
      channel: 1 - 14
      bandwidth: 0: 20M, 1: 40M, default 0
      offset: 1: 2nd channel above, 3: 2nd channel below (only use for 40M, 20M will ignore)
#
# wifi_set_channel 7
set primary channel to 7, bandwidth is 0, channel offset is 0
#
# wifi_set_channel 11 1
set primary channel to 11, bandwidth is 1, channel offset is 3
#
# wifi_set_channel 5 1 3
set primary channel to 5, bandwidth is 1, channel offset is 3
```

Note: When the STA is already connected to an AP, if you use this command to switch the STA to a channel that is inconsistent with the working channel of the AP, the command will be executed smoothly. However, the STA will be disconnected from the AP, and then try to reconnect to the AP, and switch to the original channel again.

1.15. join_group

- Usage: join_group <MulticastIP>

After executing this command the development board will join a multicast group. The development board must be connected to the AP before executing this command. Such as:

- join_group 224.0.0.5

At this point, using sniffer can catch the development board issued IGMP protocol package.

1.16. wifi_ps

Usage: wifi_ps <0 or 1, 2>

0: The power Save mode is disabled;

1: Enable the Power Save mode, wifi sleep and CPU not sleep;

2: Enable the Power Save mode, wifi sleep and CPU in deep sleep;

1.17. iperf3

This command calls iperf3 to test the network speed. Here are the relevant options for iperf3 (note the letter case).

1.17.1. iperf3 -v

The serial port will print the iperf3 version information of the SDK

1.17.2. iperf3 -h

As is shown in ***Figure 1-9. iperf3 –h command***, the serial port will print out options related to the iperf3 command.

Figure 1-9. iperf3 –h command

```
# iperf3 -h
Usage: iperf3 <-s|-c hostip|stop|-h|-v> [options]
Server or Client:
  -p, --port      #         server port to listen on/connect to
  -i, --interval  #         seconds between periodic bandwidth reports
Server specific:
  -s, --server              run in server mode
Client specific:
  -c, --client    <host>    run in client mode, connecting to <host>
  -u, --udp                 use UDP rather than TCP
  -b, --bandwidth #[KMG][/#] target bandwidth in bits/sec (0 for unlimited)
                            (default 105 Mbit/sec for UDP, unlimited for TCP)
                            (optional slash and packet count for burst mode)
  -t, --time      #         time in seconds to transmit for (default 0 secs)
  -l, --len       #[KMG]    length of buffer to read or write

[KMG] indicates options that support a K/M/G suffix for kilo-, mega-, or giga-
```

1.17.3. iperf3 -s [options]

- iperf3 -s

This time iperf3 is running in server mode and listening for TCP/UDP. The other options are the default value.

- iperf3 -s -p <port>

Set the port on which the server listens, the value range of port from 0 to 65535, the default value is 5201.

For example: iperf3 -s -p 5003

The server listens on port 5003.

- iperf3 -s -i <interval>

Set the serial port printing of test results interval(The Interval column) in seconds. The value range from 0.1 to 60, or 0. When the value is set to 0, the periodic report is not printed and only the final test results are output. The default value is 4.

For example: iperf3 -s -i 0.5

The test result is printed over the serial port every 0.5 seconds.

1.17.4. iperf3 -c <host> [options]

- iperf3 -c <host>

This time iperf3 is running in tcp client mode and all other options are default, host is the IP address of the iperf3 server.

- iperf3 -c <host> -u

With this option iperf3 runs in UDP client mode, otherwise TCP is default. This option is usually used in conjunction with the -b option to specify the data bandwidth to be sent.

- iperf3 -c <host> -p <port>

Set the port on which the client connects to the same port on which the server listens.

- iperf3 -c <host> -i <interval>

The -i option is set the same as that on the server.

- iperf3 -c <host> -b <bandwidth/number>

The unit of bandwidth is bits/sec and the format is data[KMG]. For example, 50K, 50k, or 50000, the bandwidth is 50Kbits/sec. When bandwidth is 0, there is no limit. By default, it is 1 Mbit/ sec for UDP and unlimited for TCP.

- iperf3 -c <host> -b <bandwidth>

If "/number" is not added after bandwidth, iperf3 calculates the number of packets that need to be sent per second to reach the specified bandwidth based on the length of each packet, and then sends each packet at an average interval.

For example:iperf3 -c 192.168.3.132 -u -b 200k

- iperf3 -c <host> -b <bandwidth/number>

When "/number" is added to the end of bandwidth, iperf3 will enter burst mode and continuously send a specified number of data packets at one time. There is no interval between them, but there is an interval between each batch and the interval is even.

For example:iperf3 -c 192.168.3.132 -u -b 200k/60

- iperf3 -c <host> -t <time>

Set the data transfer time in seconds. The default value is 10.

- iperf3 -c <host> -l <length>

Set the length of the read and write buffer (unit: byte) in the format of data[KMG], which is the same as the -n option. The value cannot exceed 1472 in udp mode and 1460 in tcp mode.

Note: Either iperf3-s [options] or iperf3 –c <host> [options], the options above can be used together.

1.17.5. iperf3 stop

This command can stop the iperf3 test.

1.17.6. iperf3 test example

- The development board and the test machine connect to the same AP, and then check

their IP.

- The development board uses wifi_connect command to connect to the AP, and wifi_status command to view the IP address.

- The test machine opens the iperf3 command window and starts the test.
  - The server executes the command first:iperf3 -s -p <port> -i <interval>
  - The client executes the command immediately:iperf3 -c <host> -l <length> -p <port> -i <interval> -u -b <bandwidth/number> -t <time>.
  - The -l, -p, -i, -u, -b, and -t options are optional. The -p option must be used by both server and client and has the same value. The -i option can be used at the different time and the value can be different.
  - For example:
  - iperf3 -s -p 5004 -i 1
  - iperf3 -c 192.168.1.104 -l 1460 -p 5004 -i 2 -t 20 //TCP
  - iperf3 -c 192.168.1.104 -l 1472 -p 5004 -i 4 -t 30 -u -b 50M //UDP

- When the command is executed by the server, a print message will be seen in the window to tell us that the server is open and listening in the corresponding port. When the command is executed by the client, the test machine and the development board will print the test information at the same time.

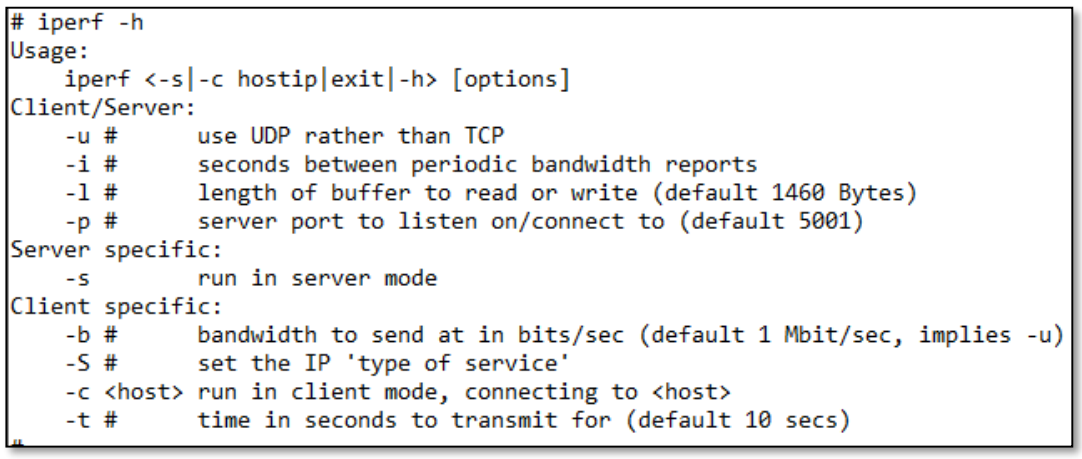
1.18. iperf

The iperf calls iperf2 command to test the network speed. The iperf runs in TCP mode by default. If testing UDP, the -u option must be used. Here are the relevant options for the command (note the letter case).

1.18.1. iperf –h

As is shown in ***Figure 1-10. iperf –h command***, The serial port will print out the iperf2-related command options.

Figure 1-10. iperf –h command



```
# iperf -h
Usage:
    iperf <-s|-c hostip|exit|-h> [options]
Client/Server:
    -u #      use UDP rather than TCP
    -i #      seconds between periodic bandwidth reports
    -l #      length of buffer to read or write (default 1460 Bytes)
    -p #      server port to listen on/connect to (default 5001)
Server specific:
    -s        run in server mode
Client specific:
    -b #      bandwidth to send at in bits/sec (default 1 Mbit/sec, implies -u)
    -S #      set the IP 'type of service'
    -c <host> run in client mode, connecting to <host>
    -t #      time in seconds to transmit for (default 10 secs)
#
```

1.18.2. iperf -s [options]

- iperf -s

This time iperf2 runs in tcp server mode, and the other options are the default values.

- iperf -s -u

This time iperf2 runs in udp server mode, and the other options are the default values.

- iperf -s -i <interval>

Set the interval (The Interval column) of the test result printed over the serial port, the unit is second. The value is an integer(Non-integers are rounded down) ranging from 1 to 3600. The default is 1.

- iperf -s -l <length>

Set the length of the read/write buffer in byte, the default value is 1460bytes. The maximum value is 2380 for udp and 4380 for tcp. In real tests, the recommended values are 1472 for UDP and 1460 for TCP.

- iperf -s -p <port>

Set the port on which the server listens. Port range from 0 to 65535 and default value is 5001.

1.18.3. iperf -c <host> [options]

- iperf -c <host>

This time iperf2 is running in tcp client mode and the other options are the default values, host is the IP address of the iperf2 server.

- iperf -c <host> -u

This time iperf2 is running in udp client mode and the other options are the default values, host is the IP address of the iperf2 server.

- iperf -c <host> -i <interval>
- iperf -c <host> -l <length>

The -l and -i options are the same as those on the server.

- iperf -c <host> -p <port>

Set the port on which the client connects to the same port on which the server listens.

- iperf -c <host> -b <bandwidth>

The unit of bandwidth is bits/ sec and the format is data[KMG]. For example, 50K, 50k, or 50000, the bandwidth is 50Kbits/ sec. When bandwidth is 0, there is no limit. The default value is 1 Mbit/ sec. This parameter is available only in UDP mode.

- iperf -c <host> -t <time>

Set the total transfer time. The default value is 10 seconds.

- iperf -c <host> -S <number>

Set the service type of the out of stack packet. The number range from 0 to 255. It can be a hexadecimal (0x prefix) or a decimal, such as 0x16 = 22.

1.18.4. iperf exit

This command terminates the iperf2 test.

1.18.5. iperf2 test example

- The development board and the test machine connect to the same AP, and then check their IP.
  - The development board uses the wifi_connect command to connect to the AP, and wifi_status command to view the IP address.
  - The test machine opens the iperf2 command window and starts the test.
  - Server executes the command first:
  - iperf -s -p <port> -i <interval> -l <length> //TCP
  - iperf -s -p <port> -i <interval> -l <length> -u //UDP
  - Client executes the command immediately:
  - iperf -c <host> -l <length> -p <port> -i <interval> -b <bandwidth/number> -t <time> -S <number>//TCP
  - iperf -c <host> -l <length> -p <port> -i <interval> -u -b <bandwidth/number> -t <time> -S <number>//UDP
  - The -l, -p, -i, -u, -b, -t, and -s options are optional.
  - **!!Note**: The -p option must be used by both server and client and have the same

value; The -i option can be used at the diffirent time on the server and the client, and the value can be different. The -u option must be used by both server and client.

- For example:
- iperf -s -p 5004 -i 1 //TCP
- iperf -s -p 5004 -i 1 –u //UDP
- iperf3 -c 192.168.1.104 -l 1460 -p 5004 -i 2 -t 20 -S 0xe0 //TCP
- iperf3 -c 192.168.1.104 -l 1472 -p 5004 -i 4 -t 30 -S 0xe0 -u -b 50M //UDP

- When the command is executed by the server, a print message will be seen in the window to tell us that the server is open and listening in the corresponding port. When the command is executed by the client, the test machine and the development board will print the test information at the same time.

1.19. wifi_ap

- Usage: wifi_ap <SSID> <PASSWORD> <CHANNEL>

This command is used to set the board to softap mode. The usage is shown in ***Figure 1-11. wifi_ap command***, SSID does not support Chinese characters, and PASSWORD is encrypted by WPA2 by default.

Figure 1-11. wifi_ap command

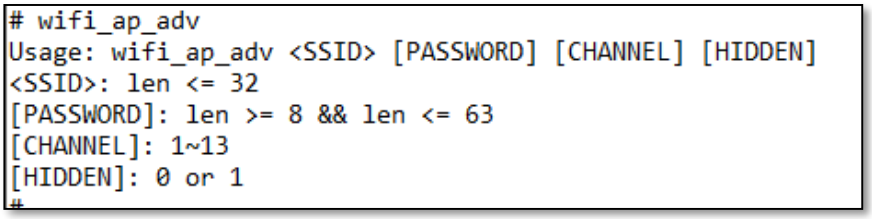
```
# wifi_ap
Usage: wifi_ap <SSID> <PASSWORD> <CHANNEL>
<SSID>: The length should be less than 32.
<PASSWORD>: The length should be between 8 and 63.
<CHANNEL>: 1~13.
#
# wifi_ap asdf 12345678 1
wifi netlink: starting softap ...
wifi netlink: softap asdf started!
```

1.20. wifi_ap_adv

Usage: wifi_ap_adv <SSID> [PASSWORD] [CHANNEL] [HIDDEN]

This command sets the development board to softap mode. The usage is shown in ***Figure 1-12. wifi_ap_adv command***, if password is not set, the AP encryption mode is open. If password is set, the AP encryption mode is WPA2. Channel defaults to 1; The parameter hidden is 0 by default. When hidden is 1, the AP does not broadcast SSID and other devices cannot detect the AP.

Figure 1-12. wifi_ap_adv command



Note: This command can be optional in square brackets, but the order can not be wrong, for example: If PASSWORD is not configured, CHANNEL and HTDDEN cannot be configured. Commands are parsed in order, if the PASSWORD field cannot be parsed successfully, subsequent parameters cannot be parsed successfully.

1.21. wifi_stop_ap

This command has no options.

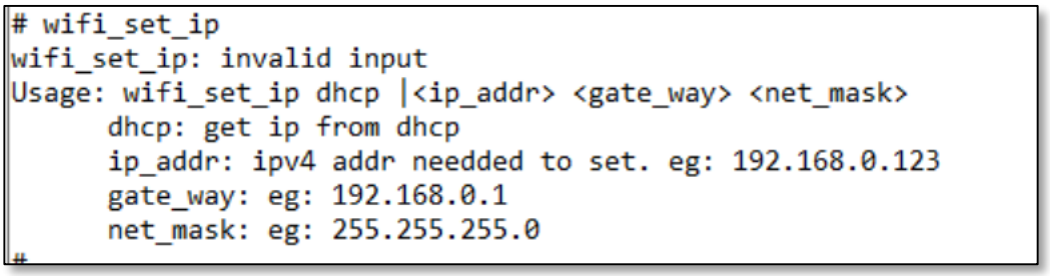
This command is used to stop softAP mode and convert it to station mode.

1.22. wifi_set_ip

Usage: wifi_set_ip dhcp |<ip_addr> <gate_way> <net_mask>

This command is used to set the static IP address of the development board or set it to the DHCP mode,The usage is shown in ***Figure 1-13. wifi_set_ip command***.

Figure 1-13. wifi_set_ip command



1.23. AT

After this command is executed, the system enters the AT command mode. In this mode, only a series of commands related to AT can be executed, and other commands cannot be executed.

For details about the AT command, please refer to < GD32W51x AT Command User Guide>.

2. Revision history

Table 2-1. Revision history

Revision No.	Description	Date
1.0	Initial Release	Nov.23, 2021

Important Notice

This document is the property of GigaDevice Semiconductor Inc. and its subsidiaries (the "Company"). This document, including any product of the Company described in this document (the “Product”), is owned by the Company under the intellectual property laws and treaties of the People’s Republic of China and other jurisdictions worldwide. The Company reserves all rights under such laws and treaties and does not grant any license under its patents, copyrights, trademarks, or other intellectual property rights. The names and brands of third party referred thereto (if any) are the property of their respective owner and referred to for identification purposes only.

The Company makes no warranty of any kind, express or implied, with regard to this document or any Product, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The Company does not assume any liability arising out of the application or use of any Product described in this document. Any information provided in this document is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Except for customized products which has been expressly identified in the applicable agreement, the Products are designed, developed, and/or manufactured for ordinary business, industrial, personal, and/or household applications only. The Products are not designed, intended, or authorized for use as components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, atomic energy control instruments, combustion control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or Product could cause personal injury, death, property or environmental damage ("Unintended Uses"). Customers shall take any and all actions to ensure using and selling the Products in accordance with the applicable laws and regulations. The Company is not liable, in whole or in part, and customers shall and hereby do release the Company as well as it’s suppliers and/or distributors from any claim, damage, or other liability arising from or related to all Unintended Uses of the Products. Customers shall indemnify and hold the Company as well as it’s suppliers and/or distributors harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of the Products.

Information in this document is provided solely in connection with the Products. The Company reserves the right to make changes, corrections, modifications or improvements to this document and Products and services described herein at any time, without notice.