GigaDevice Semiconductor Inc.

GD32W51x 基本指令用户指南

应用笔记

AN081

目录

图索引

表索引

1. 用户基本指令

使用 USB 线将测试机与开发板连接，打开 UART 工具，连接到正确的 COM 口。开发板上电并正确启动后，通过 UART 工具下发指令，开发板即可根据指令内容完成相应操作。

本手册中，指令后面< >代表该选项必填，[ ]代表该选项选填。注意指令严格执行大小写。

1.1. help

该指令没有选项。

如*图 1-1. help 指令*所示，help 指令会将开发板支持的所有指令列出。

图 1-1. help 指令

```
# help
COMMAND LIST:
==============================
    wifi_open
    wifi_close
    wifi_scan
    wifi_set_ip
    wifi_connect
    wifi_disconnect
    wifi_status
    wifi_rssi
    wifi_set_channel
    wifi_mac_addr
    wifi_ps
    wifi_ap
    wifi_ap_adv
    wifi_stop_ap
    mem_status
    ping
    join_group
    iperf
    iperf3
    exit
    reboot
    help
#
```

1.2. wifi_open

该指令没有选项。

wifi_open 用于打开 wifi，执行其他 wifi 相关命令时，需要在打开 wifi 的情况下才有效。开发板正确启动后，wifi 默认打开，因此不需要执行该指令来重复打开 wifi。该指令通常与 wifi_close

相配合，在 wifi_close 将 wifi 关闭后重新打开 wifi。

如*图 1-2. wifi_open 指令*所示，wifi 关闭后执行 wifi_open，wifi 将打开，同时串口打印 MAC 地址；如果 wifi 已打开，串口会提示 wifi 已打开。

图 1-2. wifi_open 指令

```
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

该指令没有选项。

wifi_close 可以关闭 wifi，此后一些指令将无法执行，如 wifi_scan、wifi_connect 等。

开发板处于不同情况下，指令执行结果不同，如下：

- 开发板已经与 AP 连接，则会将开发板与 AP 断连，然后关闭 wifi；
- 开发板未与 AP 连接，则直接关闭 wifi；
- 开发板为 softAP 模式，且有 sta 与开发板连接，则会断开该连接，再关闭 wifi；
- 开发板为 softAP 模式，没有 sta 连接，则直接关闭 wifi；
- wifi 已关闭，则串口会提示 wifi 已关闭。

1.4. reboot

该指令没有选项。

执行该指令后开发板将重启，串口会打印启动信息。该指令与 reset 按键作用类似。

1.5. wifi_scan

该指令没有选项。

执行该指令后串口会打印出开发板扫描到的 AP 信息，如*图 1-3. wifi_scan 指令*所示，包括 SSID，channel，加密方式，Network（BSS 类型），rate，RSSI 以及 BSSID。

图 1-3. wifi_scan 指令

```
# wifi_scan
# [Scanned AP list]
-----------------------------------------------------
SSID:         xiaomi_wifi6
Channel:      1
Security:     Open
Network:      Infrastructure
Rate:         144 Mbps
RSSI:         -76 dbm
BSSID:        8c:53:c3:d8:0d:fd

-----------------------------------------------------
SSID:         Jue
Channel:      6
Security:     WPA2
Network:      Infrastructure
Rate:         144 Mbps
RSSI:         -84 dbm
BSSID:        46:0a:0b:1c:16:95

-----------------------------------------------------
```

1.6. wifi_mac_addr

- Usage: wifi_mac_addr [MAC address]

该指令用于显示或临时变更 MAC 地址，该临时变更地址在芯片断电或 reset 后失效。

- wifi_mac_addr

串口会打印出开发板的当前 MAC 地址。

- wifi_mac_addr <MAC address>

<MAC address>为临时 MAC 地址，格式是 11:22:33:aa:bb:cc，例如：

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

在 EFUSE 没有配置 MAC 地址之前，SDK 内的 wifi MAC 地址是固定的。如果同时测试多个开发板，MAC 地址之间可能产生冲突，此时可以执行该指令临时改变 MAC 地址。

1.7. wifi_connect

- Usage: wifi_connect <SSID> [PASSWORD]

该指令用于连接 AP，此时开发板需处于 station 模式（开发板的默认模式）。

- wifi_connect <SSID>

用于连接没有加密的 AP。

- wifi_connect <SSID> <PASSWORD>

用于连接加密的 AP。

连接过程如*图 1-4. wifi_connect 指令*所示，串口打印出了连接过程信息，连接加密 AP 比连接未加密 AP 多了握手交互；同时，如果在已连接 AP 的情况下再执行 wifi_connect 指令，开发板会先与原 AP 断开，再连接新的 AP。

图 1-4. wifi_connect 指令

```
# 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

该指令没有选项。

执行该指令后开发板将与 AP 断开。

1.9. wifi_status

该指令没有选项。

执行该指令后串口将打印当前开发板的 wifi 状态。如*图 1-5. wifi_status 指令*所示，分为两个部分，WIFI Status 与 Network Interface Status。

WIFI Status 有四种，分别代表 Closed（wifi 关闭），Opened（wifi 打开），Connected（已连接 AP）及 AP Started（开启 softAP 模式，默认模式为 station 模式）。四种 status 附加了不同的信息，Connected 下是已连接 AP 的信息，包括 SSID、channel 和 bandwidth 等，AP Started 是自身作为 ap 的信息，Opened 及 Closed 没有附加信息。Network Interface Status 下是开发板的 MAC 地址、IP 及 Gateway，后两者只有 WIFI Status 是 Connected 和 AP Started 的时候有值。

图 1-5. wifi_status 指令

```
# wifi_status
WIFI Status: Opened

==============================

Network Interface Status
==============================
MAC:        [76:ba:ed:00:00:0c]
IP:         [0.0.0.0]
GW:         [0.0.0.0]
#
```

```
# wifi_status
WIFI Status: Connected
==============================
MODE:         STATION
SSID:         SmartLife-9568
CHANNEL:      1
BW:           20M
MODE:         G
SECURITY:     Open
BSSID:        82:7d:3a:04:95:68
RSSI:         -63 dbm

Network Interface Status
==============================
MAC:          [76:ba:ed:1e:00:12]
IP:           [192.168.175.4]
GW:           [192.168.175.1]
#
```

```
# wifi_status
WIFI Status: AP Started
==============================
MODE:                    AP
SSID:                    asdf
CHANNEL:          3
SECURITY:         WPA2
PASSWORD:         12345678

Network Interface Status
==============================
MAC:         [76:ba:ed:00:00:29]
IP:          [192.168.237.1]
GW:          [192.168.237.1]
#
```

1.10. wifi_rssi

该指令没有选项。

执行该指令可以获得开发板所连接 AP 的实时 RSSI 值，所以执行该指令前需要连接 ap。

1.11. mem_status

该指令没有选项。

执行该指令后串口将打印出当前的系统内存状态。

1.12. exit

该指令没有选项。

执行该指令后将离开指令操作模式，需要等待 10s 才能重新进入，继续执行指令。

1.13. ping

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

该指令用于进行 ping test。

其中，count 是 ping 包的数量；size 是包长度，单位是 byte；interval 是发包间隔，单位是 ms；total time 是总运行时间，单位是 s。默认情况下 count 为 5，size 为 120，interval 为 10，total time 不使用；如果使用 total time 选项，count 与 interval 选项将不起作用，interval 默认为 1000，count 与 total time 值相同。

ping 的使用方法如*图 1-6.* **ping 指令**所示，

图 1-6. ping 指令

```
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

ping stop 用于终止 ping test，如*图 1-7. ping stop 指令*所示，

图 1-7. ping stop 指令

```
# 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]

wifi_set_channel 用于设置开发板监听的 channel，使用方法如*图 1-8. wifi_set_channel 指令*所示，

- wifi_set_channel <channel>

只设置 channel，bandwidth 默认 0（20M），offset 默认为 0。

- wifi_set_channel <channel> <bandwidth>

设置 channel 和 bandwidth，offset 根据前两者自适应设置。

- wifi_set_channel <channel> <bandwidth> <offset>

设置 channel，bandwidth 与 offset。

图 1-8. wifi_set_channel 指令

```
# 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
```

注：当 STA 已经连接到某一个 AP 上时，如果此时将该 STA 使用此命令切换到一个与 AP 工作信道不一致的信道上时，这条命令也会顺利执行，但是此时该 STA 将会与 AP 断连，然后又会尝试重新连接，从而再次切换到原来的信道上来。

1.15. join_group

- Usage: join_group <MulticastIP>

执行该指令后开发板将加入一个多播组。执行该指令前开发板必须已连接到 AP。例如：

- join_group 224.0.0.5

此时，使用 sniffer 可以抓到开发板发出的 IGMP 协议包。

1.16. wifi_ps

Usage: wifi_ps <0 or 1, 2>

0：禁用 power save 模式；

1：启用 power save 模式，wifi sleep and CPU not sleep；

2：启用 power save 模式，wifi sleep and CPU in deep sleep；

1.17. iperf3

iperf3 指令调用 iperf3 进行网络速度测试。下面是 iperf3 的相关选项（注意大小写）。

1.17.1. iperf3 -v

串口打印出 SDK 的 iperf3 版本信息。

1.17.2. iperf3 -h

如*图 1-9. iperf3 -h 指令*所示，串口将打印出 iperf3 指令相关选项。

图 1-9. iperf3 -h 指令

```
# 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

本次 iperf3 运行在服务端模式，同时监听 TCP/UDP。其他选项为默认值。

- iperf3 -s -p <port>

设置服务端监听的端口，port 范围 0-65535，默认 5201。

举例：iperf3 -s -p 5003

服务端在 5003 端口监听。

- iperf3 -s -i <interval>

设置串口打印的测试结果的周期（Interval 这一列），单位为 second（秒），范围是 0.1-60 以及 0。当设置为 0 时代表不打印周期性报告，只输出最终的测试结果。默认是 4。

举例：iperf3 -s -i 0.5，

串口打印的测试结果周期为 0.5s。

1.17.4. iperf3 -c <host> [options]

- iperf3 -c <host>

本次 iperf3 运行在 tcp 客户端模式，其他选项均为默认值。host 为 iperf3 服务端的 IP 地址。

- iperf3 -c <host> -u

使用该选项后 iperf3 运行在 udp 客户端模式，否则默认 tcp。该选项通常与-b 选项联合使用，指定发送的数据带宽。

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

设置客户端去连接的端口，与服务端监听的端口相同。

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

-i 选项设置与服务端相同。

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

bandwidth 单位为 bits/sec，格式为：data[KMG]。如 50K、50k 或 50000，表示带宽设置为 50Kbits/sec；当 bandwidth 为 0 时，表示没有限制。udp 默认 1 Mbit/sec，tcp 无限制。

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

bandwidth 后面不加“/number”时，iperf3 会根据每个数据包的长度，算出达到指定带宽每秒需要发送的数据包数量，然后每个数据包以平均时间间隔发送。

举例：iperf3 -c 192.168.3.132 -u -b 200k

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

bandwidth 后面加“/number”时，进入 burst mode，iperf3 会一次性连续发送指定数量（number）的数据包，中间没有间隔，但每一批次之间有间隔，且间隔均匀。

举例：iperf3 -c 192.168.3.132 -u -b 200k/60

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

设置数据传输的时间，以秒为单位，默认值为 10。

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

设置读写 buffer 的长度，单位为 byte，格式为：data[KMG]，与-n 选项相同。udp 模式下该值不超过 1472，tcp 模式下不超过 1460。

备注：无论是 iperf3 -s [options]还是 iperf3 -c <host> [options]，以上 options 可搭配使用。

1.17.5. iperf3 stop

该指令可以终止 iperf3 测试。

1.17.6. iperf3 test example

- 开发板与测试机连接同一个 AP，然后查看自身 IP。
  - 开发板使用 wifi_connect 指令连接 AP，wifi_status 指令查看 IP。
- 测试机打开 iperf3 指令窗口，开始测试。
  - server 端先执行指令：iperf3 -s -p <port> -i <interval>
  - client 端随即执行指令：iperf3 -c <host> -l <length> -p <port> -i <interval> -u -b <bandwidth/number> -t <time>
  - 其中，-l、-p、-i、-u、-b、-t 选项可选。-p 选项必须 server 与 client 同时使用且值相同；-i 选项两端可不同时使用且值可不同；
  - 例如：
  - 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
- server 端执行指令后会在窗口看到打印信息，告诉我们 server 已打开且在对应 port 监听，client 端执行指令后测试机与开发板会同时打印测试信息。

1.18. iperf

iperf 指令调用 iperf2 进行网络速度测试。iperf 默认运行在 tcp 模式，如果测试 udp 必须使用-u 选项。下面是指令的相关选项（注意大小写）。

1.18.1. iperf -h

如*图 1-10. iperf -h 指令*所示，串口将打印出 iperf2 相关指令选项。

图 1-10. iperf -h 指令

```
# 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

本次 iperf2 运行在 tcp 服务端模式，其他选项为默认值。

- iperf -s -u

本次 iperf2 运行在 udp 服务端模式，其他选项为默认值。

- iperf -s -i <interval>

设置串口打印的测试结果的周期（Interval 这一列），单位为 second（秒），范围是 1-3600 之间的整数(非整数向下取整)。默认是 1。

- iperf -s -l <length>

设置读写缓冲区的长度，单位是 byte，默认是 1460bytes，udp 最大值为 2380，tcp 为 4380。在实际测试中，udp 建议值为 1472，tcp 为 1460。

- iperf -s -p <port>

设置服务端监听的端口。port 范围 0-65535，默认 5001。

1.18.3. iperf -c <host> [options]

- iperf -c <host>

本次 iperf2 运行在 tcp 客户端模式，其他选项为默认值。host 为 iperf2 服务端的 IP 地址。

- iperf -c <host> -u

本次 iperf2 运行在 udp 客户端模式，其他选项为默认值。host 为 iperf2 服务端的 IP 地址。

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

-l、-i 选项设置与服务端相同。

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

设置客户端去连接的端口，与服务端监听的端口相同。

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

bandwidth 单位为 bits/sec，格式为：data[KMG]。如 50K、50k 或 50000，表示带宽为 50Kbits/sec；当 bandwidth 为 0 时，表示没有限制。默认为 1 Mbit/sec。只在 UDP 模式使用。

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

设置传输的总时间。默认是 10 秒。

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

设置出栈数据包的服务类型。number 范围为 0-255，可以使用 16 进制（0x 前置符）或 10 进制，如 0x16 = 22。

1.18.4. iperf exit

该指令终止 iperf2 测试。

1.18.5. iperf2 test example

- 开发板与测试机连接同一个 AP，然后查看自身 IP。
  - 开发板使用 wifi_connect 指令连接 AP，wifi_status 指令查看 IP。
  - 测试机打开 iperf2 指令窗口，开始测试。
  - server 端先执行指令：
  - iperf -s -p <port> -i <interval> -l <length> //TCP
  - iperf -s -p <port> -i <interval> -l <length> -u //UDP
  - client 端随即执行指令：
  - 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
  - 其中，-l、-p、-i、-u、-b、-t、-S 选项可选。
  - !! 注意：-p 选项必须 server 与 client 同时使用且值相同；-i 选项两端可不同时使用且值可不同；-u 选项必须 server 与 client 同时使用。
  - 例如：
  - 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

- server 端执行指令后会在窗口看到打印信息，告诉我们 server 已打开且在对应 port 监听，client 端执行指令后测试机与开发板会同时打印测试信息。

1.19. wifi_ap

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

wifi_ap 用于将开发板设置为 softap 模式，使用方法如*图 1-11. wifi_ap 指令*所示，其中，SSID 不支持中文字符，PASSWORD 默认为 WPA2 加密。

图 1-11. wifi_ap 指令

```
# 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]

wifi_ap_adv 将开发板设置为 softap 模式，使用方法如*图 1-12. wifi_ap_adv 指令*所示，其中，password 不设置时 ap 为 open 加密方式，设置后为 WPA2 加密方式；channel 默认为 1；hidden 默认为 0，hidden 为 1 时该 ap 不广播 SSID，其他设备无法扫描到该 AP。

图 1-12. wifi_ap_adv 指令

```
# wifi_ap_adv
Usage: wifi_ap_adv <SSID> [PASSWORD] [CHANNEL] [HIDDEN]
<SSID>: len <= 32
[PASSWORD]: len >= 8 && len <= 63
[CHANNEL]: 1~13
[HIDDEN]: 0 or 1
#
```

注：该命令虽然方括号内的可以选填（即可填可不填），但是顺序不能错，例如：如果 PASSWORD 选择不填，那么后面的 CHANNEL 和 HTDDEN 则也无法再填，因为命令是按序解析的，如果不能顺利解析到 PASSWORD 字段，则后面的参数也会无法顺利解析出来。

1.21. wifi_stop_ap

该指令没有选项，用于终止 softAP 模式，转为 station 模式。

1.22. wifi_set_ip

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

该指令用于设置开发板的静态 IP 或设置为 DHCP 模式，使用方法如*图 1-13. wifi_set_ip 指令*所示。

图 1-13. wifi_set_ip 指令

```
# wifi_set_ip
wifi_set_ip: invalid input
Usage: wifi_set_ip dhcp |<ip_addr> <gate_way> <net_mask>
      dhcp: get ip from dhcp
      ip_addr: ipv4 addr needded to set. eg: 192.168.0.123
      gate_way: eg: 192.168.0.1
      net_mask: eg: 255.255.255.0
#
```

1.23. AT

执行该指令后将进入 AT 指令模式，该模式下只能执行 AT 相关的一系列指令，无法执行其他类型指令。

AT 指令相关内容详见《GD32W51x AT 指令用户指南》。

2. 版本历史

表 2-1. 版本历史

版本号	说明	日期
1.0	首次发布	2021 年 11 月 23 日

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.