iComm Full Mac RTOS Command Guide

Copyright@2015 iComm Company.

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Revision | Date | Author | Description |
| 0.1 | Create | Firmware team | New create |
| 0.2 | 2015/09/30 | Firmware team | Update for new feature |
| 0.3 | 2015/10/23 | Firmware team | Update for iperf3’s limitation |
| 0.4 | 2015/11/16 | Firmware team | Update for RSSI info & auto channel selection |
| 0.5 | 2016/01/18 | Firmware team | Update for SmartConfig |
| 0.6 | 2016/01/18 | Firmware team | Update for AP mode WPA2 |
| 0.7 | 2016/01/21 | Firmware team | Update for STA mode |
| 0.8 | 2017/11/09 | MAC Team | Update for 5G |

Contents

[**1. Introduction 4**](#_Toc440892341)

[**2. Features 5**](#_Toc440892342)

[**2.1 Wireless LAN Interface 5**](#_Toc440892343)

[**2.2 Host Interface 5**](#_Toc440892344)

[**2.3 Network Protocols 5**](#_Toc440892345)

[**2.4 Management & Control 5**](#_Toc440892346)

[**2.5 WiFi-Chip Architecture 5**](#_Toc440892347)

[**3. Device Function Description 6**](#_Toc440892348)

[**3.1 Wi-Fi Mode 6**](#_Toc440892349)

[**3.1.1 WireLess Configure 6**](#_Toc440892350)

[**3.1.2 STA mode 6**](#_Toc440892350)

[**3.1.3 AP mode 10**](#_Toc440892351)

[**3.1.4 SmartConfig mode 12**](#_Toc440892352)

[**3.2 Application 13**](#_Toc440892353)

[**3.2.1.1 Ping 13**](#_Toc440892354)

[**3.2.1.2 Http Server 14**](#_Toc440892355)

[**3.2.1.3 iperf3 17**](#_Toc440892356)

[**3.2.1.4 Network manager 26**](#_Toc440892357)

[**3.2.1.5 System command 26**](#_Toc440892358)

[**3.3 General command 27**](#_Toc440892359)

[**3.4 Engineer mode 28**](#_Toc440892360)

1. **Introduction**

iComm’s Wi-Fi 6030 module is dedicated designed for Wireless product. This solution includes Wi-Fi driver, Embedded OS, TCP/IP stack and some applications. The following section will describe how to control this device by cli command.

1. **Features**
   1. Wireless LAN Interface
   * IEEE 802.11 b/g
   * 2.4 GHz/5GHz WLAN interface
   * Station (Open/WEP/WPA-Personal /WPA2-Personal)
   * Soft AP mode (Open/WEP/WPA2-Personal)
   1. Host Interface
   * Serial Interface (debug and data mode)
   * SPI Interface
   * I2C Interface
   * GPIO Interface
   1. Network Protocols
   * TCP
   * UDP
   * ARP
   * ICMP
   * IPv4
   * DHCP Client/Server
   * DNS client(option)
   1. Management & Control
   * Command line interface
   1. WiFi-Chip Architecture
   * iComm ARM7 processor
   * 192 KB SRAM
2. **Device Function Description**

Our solution provides console command to communicate and control iComm 6030 module. The following section describes how to use it

* 1. **Wi-Fi Mode**
     1. **Wireless Configure**

#### **5G mode enable**

|  |  |
| --- | --- |
| Syntax | Description |
| *cfg 5G [1/0]* | “1”: Enable the 5G mode  “0” Disable the 5G mode |

#### **HT20 only on AP mode**

|  |  |
| --- | --- |
| Syntax | Description |
| *cfg ap\_ht20\_only [1/0]* | “1”: AP mode HT20 only  “0” AP mode HT20/40 |

* + 1. **STA Mode**

Description:

In station mode, we provide some function control to let user experience iComm 6030 Wi-Fi module easily. The following commands could let user turn ON station mode, scan APs in different channel and join the specific one, leave the AP when user want to switch to another one. We also provide commands to let user check current system status easily.

Command:

The following command is available for iComm 6030 Wi-Fi module

#### **STA mode ON**

|  |  |
| --- | --- |
| Syntax | Description |
| *ctl sta on* | User could set station mode on |

**Example:**

The following example shows how to use the “STA mode ON” command:

wifi-host> ctl sta on

OK.

#### **Scan APs in different channel**

|  |  |
| --- | --- |
| Syntax | Description |
| *iw scan <0xffff> <0xffffffff>* | User could scan APs By this cmd. This first parameters(0xffff) map to the 2G band channel, and the second parameters(0xffffffff) map to the 5G band channel.  <0xffff>: ch1~ch14, Bit[0] is reserved, Bit[1]=ch1, Bit[2]=ch2, and so on  <0xffffffff>:Bit[0]=ch36, Bit[1]=ch40, Bit[2]=ch44, Bit[3]=ch48, Bit[4]=ch52, Bit[5]=ch56, Bit[6]=ch60, Bit[7]=ch64, Bit[8]=ch100, Bit[9]=ch104, Bit[10]=ch108, Bit[11]=ch112, Bit[12]=ch116, Bit[13]=ch120, Bit[14]=ch124, Bit[15]=ch128, BIT[16]=ch132, BIT[17]=136, BIT[18]=ch140, BIT[19]=ch144, BIT[20]=ch149, BIT[21]=153, BIT[22]=157, BIT[23]=161, BIT[24]=165, BIT[25~31] are reserved |

**Example:**

The following example shows how to use the “Scan APs in different channel” command:

If user wants to scan AP in channel 6:

wifi-host> iw scan 0x0040 0

BSSID: b8:3e:59:47:e5:a3

SSID: icomm-ap @Channel Idx: 6

proto: WPA2

Pairwise cipher=[CCMP]

Group cipher=[CCMP]

If user wants to scan AP from channel 6~9:

wifi-host> iw scan 0x03c0 0

If user wants to scan APs that locate on 2G Band

wifi-host> iw scan 0x3fff 0

If user wants to scan APs that locate on 5G Band

wifi-host> iw scan 0 0xffffffff

If user wants to scan APs that locate on 2G Band and 5G Band

wifi-host> iw scan

#### **List APs**

|  |  |
| --- | --- |
| Syntax | Description |
| *iw list* | User could list APs which have been scanned in station mode |

**Example:**

The following example shows how to use the “List AP” command:

wifi-host> iw list

BSSID: b8:3e:59:47:e5:a3

SSID: icomm-ap @Channel Idx: 1

proto: WPA2

Pairwise cipher=[CCMP]

Group cipher=[CCMP]

BSSID: b8:3e:59:47:e5:aa

SSID: icomm-ap-2 @Channel Idx: 2

proto: WPA2

Pairwise cipher=[CCMP]

Group cipher=[CCMP]

#### **Join AP**

|  |  |
| --- | --- |
| Syntax | Description |
| *iw join <AP\_SSID>[password]* | User could join the specific AP that is already in the AP list.  <AP\_SSID> is the specific AP’s SSID which user wants to join in.  [password] is <AP\_SSID>’s access password. If the <AP\_SSID> did not set security protocol (open mode), please let [password] empty.  Note:  The max length of [password] is 5 character in [WEP40] security protocol.  The limit of the [password] length is 8~63 character in [WPA/WPA2] security protocol. |

**Example:**

The following example shows how to use the “Join AP” command:

wifi-host> iw join icomm-ap 12345678

wmm\_used = 0

Joining " icomm-ap " using security type "wpa2".

Join success!! AID=2

#### **Join other AP**

|  |  |
| --- | --- |
| Syntax | Description |
| *iw join-other <AP\_SSID>[password]* | User could join the specific AP that is not in the AP list or is a hidden AP.  <AP\_SSID> is the specific AP’s SSID which user wants to join in.  [password] is <AP\_SSID>’s access password. If the <AP\_SSID> did not set security protocol (open mode), please let [password] empty.  Note:  The max length of [password] is 5 character in [WEP40] security protocol.  The limit of the [password] length is 8~63 character in [WPA/WPA2] security protocol. |

**Example:**

The following example shows how to use the “Join AP” command:

wifi-host> iw join-other icomm-ap 12345678

#### **Leave AP**

|  |  |
| --- | --- |
| Syntax | Description |
| *iw leave* | If user has joined the specific AP, and user want to leave it |

**Example:**

The following example shows how to use the “Leave AP” command:

wifi-host> iw leave

Leave received deauth from AP (reason=0) !!

#### **Get Status**

|  |  |
| --- | --- |
| Syntax | Description |
| *ctl status* | User could get current STA mode status outline |

**Example:**

The following example shows how to use the “Get Status” command:

wifi-host> ctl status

status:ON

Mode:Station, connected

self Mac addr: 60:11:13:86:b6:01

SSID:icomm-ap

AP Mac addr: 54:a0:50:e3:c1:7e

OK.

#### **Support Country**

|  |  |
| --- | --- |
| Syntax | Description |
| *iw reg* | Show the support Country |

#### **Set Country**

|  |  |
| --- | --- |
| Syntax | Description |
| *iw sreg [country code]* | Set the current Country |

#### **Get Country**

|  |  |
| --- | --- |
| Syntax | Description |
| *iw greg* | Get the current Country |

#### **Show available channels in the current country**

|  |  |
| --- | --- |
| Syntax | Description |
| *iw clist* | Show the available channels that are in the current country |

* + 1. **AP mode**

Description:

We provide a simple way to operate iComm 6030 Wi-Fi module’s excellent AP mode function. The following commands could let user turn ON AP mode and check current AP’s status. We also provide WEP security mechanism to make sure that user could transmit their private data safely.

Command:

The following command line is available for iComm 6030 Wi-Fi module

#### **AP mode ON**

|  |  |
| --- | --- |
| Syntax | Description |
| *ctl ap on <AP\_SSID> [channel ID] [wep/wpa2][password]* | User could set AP mode ON. User could set <AP\_SSID> with different characters or numbers. The max length of <AP\_SSID> is 32 characters, and it could not be empty. User could set [channel ID] from number 1 to 14, if user did not set channel ID, system will use auto channel selection mechanism to decide the best channel in current environment. If user want to enable WEP security, user could add [wep/wpa2] and [password] for other station to connect on.  Note:  iComm 6030 Wi-Fi module AP mode only support “open mode” and “WEP” currently.  <AP\_SSID> could not set “space” or “?” as SSID’s character.  [wep\_password] could only set 5 or 13 character currently |

**Example:**

The following example shows how to use the “AP mode ON” command:

Ex1: Configure a AP “icomm-ap” on channel 11 with WEP

wifi-host> ctl ap on icomm-ap 11 wep 12345

Ex1: Configure a AP with WEP, and do the auto channel selection.

wifi-host> ctl ap on icomm-ap wep 12345

#### **Get Status**

|  |  |
| --- | --- |
| Syntax | Description |
| *ctl status* | User could get current AP mode status outline |

**Example:**

The following example shows how to use the “Get Status” command:

wifi-host> ctl status

status:ON

Mode:AP

self Mac addr: 60:11:58:6b:84:aa

SSID:icomm-ap

channel:11

Station number:1

station Mac addr: 80:ea:96:2b:59:76

===================APStaInfo\_PrintStaInfo==================

STA:0

addr:80:ea:96:2b:59:76

aid:1

\_flag:00020547 : VALID AUTH ASSOC AUTHORIZED SHORT\_PREAMBLE WMM PS\_STA

idle:40 buf timeout:2000

RSSI = -33 (dBm)

BUF AC[0]:0

BUF AC[1]:0

BUF AC[2]:0

BUF AC[3]:0

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

STA:1

=>Invalid

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

OK.

* + 1. **SmartConfig mode**

Description:

SmartConfig is the special function that allows setting the SSID and password without having to type. It’s very useful on IOT devices.

Command:

The following command is available for iComm 6030 Wi-Fi module

#### **SmartConfig mode on**

|  |  |
| --- | --- |
| Syntax | Description |
| *ctl sconfig on* | There are no any other parameters in this cmd. After run this cmd, system change to SmartConfig mode, and ready to run SmartConfig solution. Now, iComm 6030 support WeChat’s AirKiss and iComm’s SmartLink. |

**Example:**

The following example shows how to use the “AP mode ON” command:

wifi-host> ctl sconfig on

#### **Run SmartConfig**

|  |  |
| --- | --- |
| Syntax | Description |
| *iw sconfig [solution] [2G channel mask] [5G channel mask]* | user can choose a SmartConfig solution to help you to get the SSID and password.  [solution]: “AIRKISS”,”SLINK”  [2G channel mask]: Total 16bits, each bits corresponds to a channel. Bit[0] is reserved, Bit[1]=ch1, Bit[2]=ch2, and so on  [5G channel mask]: Total 16bits, each bits corresponds to a channel  Bit[0]=ch36, Bit[1]=ch40, Bit[2]=ch44, Bit[3]=ch48, Bit[4]=ch52, Bit[5]=ch56, Bit[6]=ch60, Bit[7]=ch64, Bit[8]=ch100, Bit[9]=ch104, Bit[10]=ch108, Bit[11]=ch112, Bit[12]=ch116, Bit[13]=ch120, Bit[14]=ch124, Bit[15]=ch128, BIT[16]=ch132, BIT[17]=136, BIT[18]=ch140, BIT[19]=ch144, BIT[20]=ch149, BIT[21]=153, BIT[22]=157, BIT[23]=161, BIT[24]=165, BIT[25~31] are reserved |

**Example:**

The following example shows how to run SmartConfig

Run AirKiss, and scan 2G band channel only

wifi-host> iw sconfig AIRKISS 0xffff 0

Run AirKiss, and scan 5G band channel only

wifi-host> iw sconfig AIRKISS 0 0xffffffff

Run AirKiss, and scan 2G and 5G band channel

wifi-host> iw sconfig AIRKISS 0xffff 0xffffffff

* 1. **Application**
     + 1. **Ping**

Description:

We provide ping function to help user check status of connection

#### **ping**

|  |  |
| --- | --- |
| Syntax | Description |
| *ping [-c count] [-s size] [-d delay(ms)] <destination\_IP>* | User could check whether the specific destination is available. ***<***destination\_IP> is the IP address of destination, and it could not be empty |

**Example:**

The following example shows how to use the “ping” command:

wifi-host> ping 192.168.1.1

wifi-host> PING 192.168.1.1 32(60) bytes of data.

(1, 0, 1) 40 bytes from 192.168.1.1: icmp\_seq=13 ttl=64 time=3 ms

(2, 0, 2) 40 bytes from 192.168.1.1: icmp\_seq=14 ttl=64 time=4 ms

(3, 0, 3) 40 bytes from 192.168.1.1: icmp\_seq=15 ttl=64 time=3 ms

(4, 0, 4) 40 bytes from 192.168.1.1: icmp\_seq=16 ttl=64 time=3 ms

recv\_count:4, miss\_count:0, ping\_count:4

rtt min/avg/max = 3/3/4 ms

* + - 1. **Http Server**

Description:

We support webpage operation that user could use simply via web browser. User could check current iComm 6030 module status, full function control in AP/ STA mode, and basic function operation when user is on the webpage.

#### **Login page**

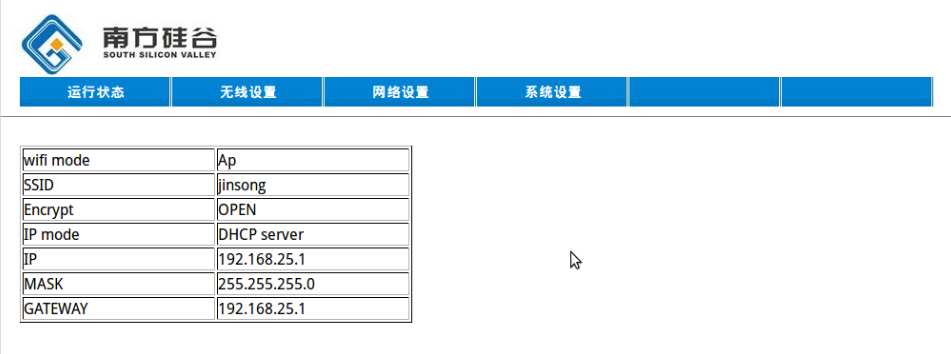


Default user and password is none.

Notes:

In AP mode, the default system IP address is 192.168.25.1

#### **Running status**



This page shows network and wireless information.

|  |  |
| --- | --- |
| **Wifi mode** | AP or STA |
| **SSID** | AP’s SSID |
| **Encrypt** | OPEN/WEP/WPA/WPA2 |
| **IP mode** | DHCP client / DHCP server / Static IP |
| **IP** | IP address |
| **Mask** | IP mask |
| **Gateway** | IP gateway |

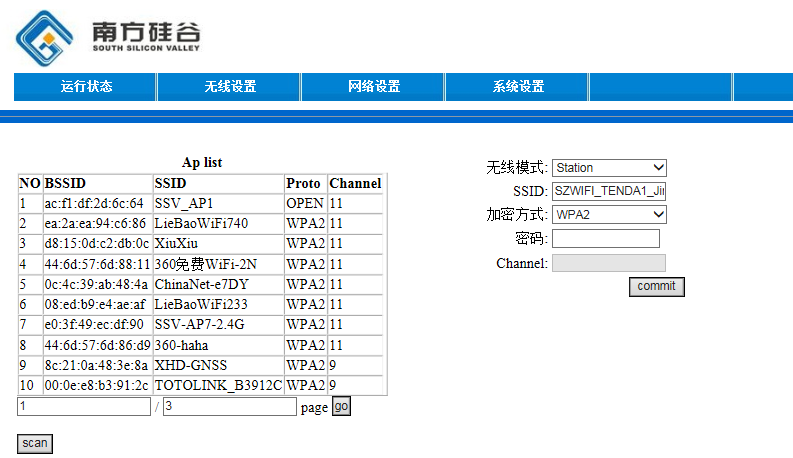
#### **Wireless setting**

1. AP mode

User can set AP mode, SSID, open encrypt, channel.

Note:

In AP mode, it can only support “open” and “WEP” encrypt mode, and AP list table is blank.



1. STA mode

User can set STA mode, join AP SSID, encrypt mode, password. Channel can’t be set. In addition, the left area is AP list table which is scanned, and user can also scan manual on this page.

Note:

AP mode -> STA mode, user need wait a second because the device needs to do AP list scanning for a while.

#### **Network setting**

This page show the network information, included IP mode (DHCP client/server, Static), IP mask, and gateway.

#### **System setting**



This function page shows nothing temporarily.

* + - 1. **iperf3**

Description:

iperf3 is a TCP and UDP network bandwidth measurement tool. iperf3 could measure maximum achievable bandwidth on IP networks. It supports tuning of various parameters related to different timing, protocols and buffers; it will report the test result of bandwidth, loss, and other accuracy measurement data. iperf3 is based on client/server mode. The parameter “-s” indicates the iperf3 thread run in server mode, and “-c” run in client mode. iperf3 provide general options which could be used in both server and client mode, and specific option for each server mode or client mode only. User could get more detail in following descriptions. iComm 6030 Wi-Fi module could run two iperf3 threads at same time, which could run in server mode and client mode separately at one time, especially notice that two thread should run in different TCP/UDP ports via option “-p <port>”, and both threads could be stopped by the command “iperf3 stop”. iperf3 is NOT backwards compatible with the original iperf.

#### **Iperf3**

|  |  |
| --- | --- |
| Syntax | Description |
| *iperf3 -c <server> [options]* | Start iperf3 test which run in client mode. <server> is the MAC address of server which is running in server mode. [option] include different parameter for client mode test, user could check detail by using “iperf3 -h” |
| *iperf3 -s* | Start iperf3 test as a server |
| *iperf3 stop* | Stop iperf3 |
| *iperf3 [-h/--help] [-v/--version]* | Get help / version information |

**Example:**

The following examples show how to use the “iperf3” command:

To get iperf3 usage in details:

wifi-host> iperf3 -h

Usage: iperf3 [-s|-c host] [options]

iperf3 [-h|--help] [-v|--version]

Server or Client:

-p, --port # server port to listen on/connect to

-i, --interval # seconds between periodic bandwidth reports

-V, --verbose more detailed output

-d, --debug emit debugging output

-v, --version show version information and quit

-h, --help show this message and quit

stop kill the iperf threads

list show the running iperf threads

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 1 Mbit/sec for UDP, unlimited for TCP)

(optional slash and packet count for burst mode)

-t, --time # time in seconds to transmit for (default 10 secs)

-n, --bytes #[KMG] number of bytes to transmit (instead of -t)

-k, --blockcount #[KMG] number of blocks (packets) to transmit (instead of -t or -n)

-l, --len #[KMG] length of buffer to read or write

(default 4 KB for TCP, 1472 Bytes for UDP)

-N, --no-delay set TCP no delay, disabling Nagle's Algorithm

-P, --parallel # number of parallel client streams to run

-T, --title str prefix every output line with this string

iperf3 run in server mode:

wifi-host> iperf3 -s

-----------------------------------------------------------

Server listening on 5201, socket:0

-----------------------------------------------------------

iperf3 run in client mode:

wifi-host> iperf3 –c 192.168.25.101

Connecting to host 192.168.25.101, port 5201

[ 1] local 192.168.25.1 port 49154 connected to 192.168.25.101 port 5201

[ ID] Interval Transfer Bandwidth

[ 1] 0.00- 1.00 sec 376 KBytes 3.08 Mbits/sec

[ 1] 1.00- 2.00 sec 490 KBytes 4.02 Mbits/sec

[ 1] 2.00- 3.00 sec 559 KBytes 4.58 Mbits/sec

[ 1] 3.00- 4.00 sec 523 KBytes 4.29 Mbits/sec

[ 1] 4.00- 5.00 sec 478 KBytes 3.91 Mbits/sec

[ 1] 5.00- 6.00 sec 472 KBytes 3.87 Mbits/sec

[ 1] 6.00- 7.00 sec 525 KBytes 4.30 Mbits/sec

[ 1] 7.00- 8.00 sec 488 KBytes 3.99 Mbits/sec

[ 1] 8.00- 9.00 sec 489 KBytes 4.01 Mbits/sec

[ 1] 9.00- 10.00 sec 436 KBytes 3.57 Mbits/sec

- - - - - - - - - - - - - - - - - - - - - - - - -

[ ID] Interval Transfer Bandwidth

[ 1] 0.00- 10.00 sec 4.72 MBytes 3.96 Mbits/sec sender

[ 1] 0.00- 10.00 sec 4.72 MBytes 3.96 Mbits/sec receiver

iperf Done.

In client mode, option “-t <seconds>” could change test duration.

Example: Set test duration to 5 seconds:

wifi-host> iperf3 –c 192.168.25.101 –t 5

wifi-host>

Connecting to host 192.168.25.101, port 5201

[ 1] local 192.168.25.1 port 49156 connected to 192.168.25.101 port 5201

[ ID] Interval Transfer Bandwidth

[ 1] 0.00- 1.00 sec 542 KBytes 4.44 Mbits/sec

[ 1] 1.00- 2.00 sec 412 KBytes 3.38 Mbits/sec

[ 1] 2.00- 3.00 sec 461 KBytes 3.77 Mbits/sec

[ 1] 3.00- 4.00 sec 482 KBytes 3.95 Mbits/sec

[ 1] 4.00- 5.00 sec 405 KBytes 3.32 Mbits/sec

- - - - - - - - - - - - - - - - - - - - - - - - -

[ ID] Interval Transfer Bandwidth

[ 1] 0.00- 5.00 sec 2.25 MBytes 3.77 Mbits/sec sender

[ 1] 0.00- 5.00 sec 2.25 MBytes 3.77 Mbits/sec receiver

iperf Done.

In client mode, iperf3 will use TCP to measure the bandwidth by default, option “-u” could set UDP test

wifi-host> iperf3 –c 192.168.25.101 -u

Connecting to host 192.168.25.101, port 5201

[ 1] local 0.0.0.0 port 49154 connected to 192.168.25.101 port 5201

[ ID] Interval Transfer Bandwidth Total Datagrams

[ 1] 0.00- 1.00 sec 116 KBytes 954 Kbits/sec 81

[ 1] 1.00- 2.00 sec 116 KBytes 954 Kbits/sec 81

[ 1] 2.00- 3.00 sec 116 KBytes 954 Kbits/sec 81

[ 1] 3.00- 4.00 sec 116 KBytes 954 Kbits/sec 81

[ 1] 4.00- 5.00 sec 116 KBytes 954 Kbits/sec 81

[ 1] 5.00- 6.00 sec 116 KBytes 954 Kbits/sec 81

[ 1] 6.00- 7.00 sec 116 KBytes 954 Kbits/sec 81

[ 1] 7.00- 8.00 sec 116 KBytes 954 Kbits/sec 81

[ 1] 8.00- 9.00 sec 116 KBytes 954 Kbits/sec 81

[ 1] 9.00- 10.00 sec 116 KBytes 954 Kbits/sec 81

- - - - - - - - - - - - - - - - - - - - - - - - -

[ ID] Interval Transfer Bandwidth Jitter Lost/Total Datagrams

[ 1] 0.00- 10.00 sec 1.14 MBytes 954 Kbits/sec 0.617 ms 0/810 (0%)

[ 1] Sent 810 datagrams

iperf Done.

In client mode UDP test, there are some default options, including UDP payload length (1472 bytes) and bandwidth (954Kbps). In order to change the values, please use options “-b”.

Example: Measure the UDP performance with bandwidth 5Mbps:

wifi-host> iperf3 –c 192.168.25.101 –u –b 5M

Connecting to host 192.168.25.101, port 5201

[ 1] local 0.0.0.0 port 49154 connected to 192.168.25.101 port 5201

[ ID] Interval Transfer Bandwidth Total Datagrams

[ 1] 0.00- 1.00 sec 579 KBytes 4.75 Mbits/sec 403

[ 1] 1.00- 2.00 sec 579 KBytes 4.75 Mbits/sec 403

[ 1] 2.00- 3.00 sec 579 KBytes 4.75 Mbits/sec 403

[ 1] 3.00- 4.00 sec 579 KBytes 4.75 Mbits/sec 403

[ 1] 4.00- 5.00 sec 581 KBytes 4.76 Mbits/sec 404

[ 1] 5.00- 6.00 sec 578 KBytes 4.73 Mbits/sec 402

[ 1] 6.00- 7.00 sec 579 KBytes 4.75 Mbits/sec 403

[ 1] 7.00- 8.00 sec 579 KBytes 4.75 Mbits/sec 403

[ 1] 8.00- 9.00 sec 578 KBytes 4.73 Mbits/sec 402

[ 1] 9.00- 10.00 sec 579 KBytes 4.75 Mbits/sec 403

- - - - - - - - - - - - - - - - - - - - - - - - -

[ ID] Interval Transfer Bandwidth Jitter Lost/Total Datagrams

[ 1] 0.00- 10.00 sec 5.66 MBytes 4.74 Mbits/sec 0.739 ms 0/4029 (0%)

[ 1] Sent 4029 datagrams

iperf Done.

Once the UDP begins, the iperf3 client will show information of the throughput. The client will not know whether the packets are received by the server or not until the test is finished normally because the client and server exchange the testing result at the end of the test. So, in order to get the intermediate network bandwidth and status during testing, the output of both the client and the server should be checked. For the TCP test, the output of both the client and the server will be alike, because TCP has the reliable transmission control mechanism. The iperf3 output on the Linux host:

Accepted connection from 192.168.1.105, port 49158

[ 5] local 192.168.25.101 port 5201 connected to 192.168.25.1 port 49155

[ ID] Interval Transfer Bandwidth Jitter Lost/Total Datagrams

[ 5] 0.00-1.00 sec 450 KBytes 3.69 Mbits/sec 0.786 ms 0/313 (0%)

[ 5] 1.00-2.00 sec 579 KBytes 4.75 Mbits/sec 0.574 ms 0/403 (0%)

[ 5] 2.00-3.00 sec 579 KBytes 4.75 Mbits/sec 0.671 ms 0/403 (0%)

[ 5] 3.00-4.00 sec 579 KBytes 4.75 Mbits/sec 0.686 ms 0/403 (0%)

[ 5] 4.00-5.00 sec 579 KBytes 4.75 Mbits/sec 0.539 ms 0/403 (0%)

[ 5] 5.00-6.00 sec 581 KBytes 4.76 Mbits/sec 0.717 ms 0/404 (0%)

[ 5] 6.00-7.00 sec 578 KBytes 4.73 Mbits/sec 0.783 ms 0/402 (0%)

[ 5] 7.00-8.00 sec 579 KBytes 4.75 Mbits/sec 0.922 ms 0/403 (0%)

[ 5] 8.00-9.00 sec 579 KBytes 4.75 Mbits/sec 0.552 ms 0/403 (0%)

[ 5] 9.00-10.00 sec 578 KBytes 4.73 Mbits/sec 0.726 ms 0/402 (0%)

[ 5] 10.00-10.21 sec 129 KBytes 4.97 Mbits/sec 0.739 ms 0/90 (0%)

- - - - - - - - - - - - - - - - - - - - - - - - -

[ ID] Interval Transfer Bandwidth Jitter Lost/Total Datagrams

[ 5] 0.00-10.21 sec 5.66 MBytes 4.65 Mbits/sec 0.739 ms 0/4029 (0%)

-----------------------------------------------------------

Server listening on 5201

-----------------------------------------------------------

To run the second iperf3 thread, option “-p <port>” should be used to assign a different transport number for the corresponding iperf3 client and server.

wifi-host> iperf3 –c 192.168.25.101 –p 5200

Connecting to host 192.168.25.101, port 5200

[ 1] local 192.168.25.1 port 49160 connected to 192.168.25.101 port 5200

[ ID] Interval Transfer Bandwidth

[ 1] 0.00- 1.00 sec 471 KBytes 3.85 Mbits/sec

[ 1] 1.00- 2.00 sec 388 KBytes 3.18 Mbits/sec

[ 1] 2.00- 3.00 sec 489 KBytes 4.01 Mbits/sec

[ 1] 3.00- 4.00 sec 496 KBytes 4.06 Mbits/sec

[ 1] 4.00- 5.00 sec 479 KBytes 3.92 Mbits/sec

[ 1] 5.00- 6.00 sec 532 KBytes 4.36 Mbits/sec

[ 1] 6.00- 7.00 sec 529 KBytes 4.33 Mbits/sec

[ 1] 7.00- 8.00 sec 224 KBytes 1.83 Mbits/sec

[ 1] 8.00- 9.00 sec 462 KBytes 3.78 Mbits/sec

[ 1] 9.00- 10.00 sec 412 KBytes 3.38 Mbits/sec

- - - - - - - - - - - - - - - - - - - - - - - - -

[ ID] Interval Transfer Bandwidth

[ 1] 0.00- 10.00 sec 4.38 MBytes 3.67 Mbits/sec sender

[ 1] 0.00- 10.00 sec 4.38 MBytes 3.67 Mbits/sec receiver

Meanwhile, the same iperf3 port should be assigned to the iperf3 server, like “iperf3 –s –p 5200” on the host of 192.168.25.101. There are no other restraints except for the different port value for the two iperf3 threads.

wifi-host> iperf3 -s -p 5200

-----------------------------------------------------------

Server listening on 5200

-----------------------------------------------------------

In client mode, option “-P <streams>” can be used to test with several streams in parallel:

Note:

If target platform is under win32 environment and is running on server mode, it could not be allowed to use several streams in parallel when user needs to run UDP TX iperf3 testing, but it works on TCP TX.

wifi-host> iperf3 –c 192.168.25.101 –P 2

In this case, the iperf client will create two TCP connections and try to transmit traffic at its best. The two streams share the same iperf thread and have the same transmission direction.

wifi-host> iperf3 -c 192.168.1.100 -P 2 -t 5

wifi-host>

Connecting to host 192.168.1.100, port 5201

[ 1] local 192.168.25.1 port 49165 connected to 192.168.25.1 port 5201

[ 2] local 192.168.25.1 port 49166 connected to 192.168.25.1 port 5201

[ ID] Interval Transfer Bandwidth

[ 1] 0.00- 1.00 sec 386 KBytes 3.17 Mbits/sec

[ 2] 0.00- 1.00 sec 373 KBytes 3.06 Mbits/sec

[SUM] 0.00- 1.00 sec 759 KBytes 6.22 Mbits/sec

- - - - - - - - - - - - - - - - - - - - - - - - -

[ 1] 1.00- 2.00 sec 475 KBytes 3.89 Mbits/sec

[ 2] 1.00- 2.00 sec 489 KBytes 4.01 Mbits/sec

[SUM] 1.00- 2.00 sec 964 KBytes 7.90 Mbits/sec

- - - - - - - - - - - - - - - - - - - - - - - - -

[ 1] 2.00- 3.00 sec 364 KBytes 2.98 Mbits/sec

[ 2] 2.00- 3.00 sec 361 KBytes 2.96 Mbits/sec

[SUM] 2.00- 3.00 sec 724 KBytes 5.93 Mbits/sec

- - - - - - - - - - - - - - - - - - - - - - - - -

[ 1] 3.00- 4.00 sec 515 KBytes 4.22 Mbits/sec

[ 2] 3.00- 4.00 sec 518 KBytes 4.24 Mbits/sec

[SUM] 3.00- 4.00 sec 1.01 MBytes 8.46 Mbits/sec

- - - - - - - - - - - - - - - - - - - - - - - - -

[ 1] 4.00- 5.00 sec 506 KBytes 4.15 Mbits/sec

[ 2] 4.00- 5.00 sec 493 KBytes 4.04 Mbits/sec

[SUM] 4.00- 5.00 sec 999 KBytes 8.19 Mbits/sec

- - - - - - - - - - - - - - - - - - - - - - - - -

[ ID] Interval Transfer Bandwidth

[ 1] 0.00- 5.00 sec 2.19 MBytes 3.68 Mbits/sec sender

[ 1] 0.00- 5.00 sec 2.19 MBytes 3.68 Mbits/sec receiver

[ 2] 0.00- 5.00 sec 2.18 MBytes 3.66 Mbits/sec sender

[ 2] 0.00- 5.00 sec 2.18 MBytes 3.66 Mbits/sec receiver

[SUM] 0.00- 5.00 sec 4.37 MBytes 7.34 Mbits/sec sender

[SUM] 0.00- 5.00 sec 4.37 MBytes 7.34 Mbits/sec receiver

In case that a two-way throughput test is required, the following iperf3 commands will meet this demand.

wifi-host> iperf3 –s –p 5200

wifi-host> iperf3 –c 192.168.25.101

Meanwhile, you should run iperf3 commands on the peer host. Suppose wifi-host runs at AP mode and is assigned an IP address 192.168.25.1, the peer iperf3 commands would be “iperf3 –c 192.168.25.1 –p 5200” and “iperf3 –s”. In this example, both the iperf3 threads measure the throughput with one TCP stream separately. Note that the iperf3 servers should be activated in advance of the corresponding iperf3 client.

The following example shows the information of the ongoing iperf3 threads:

wifi-host> iperf3 list

iperf task1: server, server port: 5201

In order to break the test, “iperf3 stop” can be executed.

wifi-host> iperf3 stop

The iperf threads stopped!

* + - 1. **Network manager**

#### **Netmgr Show**

|  |  |
| --- | --- |
| Syntax | Description |
| *netmgr show* | User could see DHCP information, included DHCP server allocated client IP and DHCP/DHPCD on/off status. |

**Example:**

The following example shows how to use the “Netmgr Show” command:

wifi-host> netmgr show

Dhcpd: on

Dhcpc: off

------------------------------------------------------------

| MAC | IP |

------------------------------------------------------------

[d8:fc:93:2d:8b:7e] --- [192.168.25.101]

* + - 1. **System command**
  1. **General command**

Description:

We provide some general commands to help user get basic information. User could use “ifconfig” to get station’s IP and MAC address, user could use “?” to get the information of what kind of command user could use.

Command:

The following command line is available for iComm 6030 Wi-Fi module

#### **ifconfig**

|  |  |
| --- | --- |
| Syntax | Description |
| *ifconfig* | User could get current network configuration |

**Example:**

The following example shows how to use the “ifconfig” command:

wifi-host> ifconfig

wlan0 Link encap:Ethernet HWaddr 60:11:7a:62:ef:ea

inet addr:0.0.0.0 Bcast:255.255.255.255 Mask:0.0.0.0

UP BROADCAST LINK-DOWN MTU:1500 GW:0.0.0.0

lo Link encap:Ethernet HWaddr 00:00:00:00:00:00

inet addr:127.0.0.1 Bcast:127.255.255.255 Mask:255.0.0.0

UP POINT-TO-POINT LINK-DOWN MTU:0 GW:127.0.0.1

#### **Help**

|  |  |
| --- | --- |
| Syntax | Description |
| *?* | User could know what kind of commands they can use in iComm 6030 Wi-Fi module at station mode |

**Example:**

The following example shows how to use the “Help” command:

wifi-host> ?

Usage:

ifconfig Network interface configuration

ping ping

iperf3 throughput testing via tcp or udp

netapp Network application utilities

netmgr Network Management utilities

iw Wireless utility

ctl wi-fi interface control (AP/station on or off)

r Read SoC

w Write SoC

dump Do dump.

s dump tcp/ip status

sys Components info

spi check spi status

* 1. **Engineer mode**

[] option

<> choose one