



## **AT Command Application Note**

---

This document provides information for controlling Ameba through external UART.

---

## Table of Contents

1	System Architecture .....	5
2	Command Format .....	5
3	AT command .....	5
3.1	AT command list.....	5
3.2	AT command list.....	7
3.2.1	COMMON.....	7
3.2.1.1	‘help’ Print help message .....	7
3.2.1.2	‘AT??’ Print Log History .....	7
3.2.1.3	‘AT--’ Exit Log Service.....	7
3.2.2	WLAN .....	7
3.2.2.1	‘ATW0’ Wlan Set Network SSID .....	7
3.2.2.2	‘ATW1’ Wlan set Network Passphrase .....	7
3.2.2.3	‘ATW2’ Wlan Set Key ID.....	8
3.2.2.4	‘ATWC’ Wlan Join a Network.....	8
3.2.2.5	‘ATWD’ Wlan Disconnect from Network.....	8
3.2.2.6	‘ATW3’ Wlan Set Access Point SSID.....	8
3.2.2.7	‘ATW4’ Wlan Set Access Point Security Key .....	8
3.2.2.8	‘ATW5’ Wlan Set Access Point Channel.....	8
3.2.2.9	‘ATWA’ Wlan Activate Access Point .....	9
3.2.2.10	‘ATWB’ Wlan Activate Access Point mode and Station mode .....	9
3.2.2.11	‘ATW?’ Wlan Show WiFi information .....	9
3.2.2.12	‘ATWS’ Wlan Scan for Network Access Point .....	9
3.2.2.13	‘ATWR’ Wlan Get RSSI of Associated Network Access Point .....	9
3.2.2.14	‘ATWM’ Wlan Wi-Fi promisc .....	9
3.2.2.15	‘ATWE’ Wlan Start Web Server.....	10
3.2.2.16	‘ATWQ’ Wlan Wi-Fi Simple Config.....	10
3.2.2.17	‘ATWP’ Wlan Power on/off wifi module .....	10
3.2.2.18	‘ATWI’ Wlan ping test.....	10
May 12, 2015		2

---

3.2.2.19	'ATWO' Wlan OTA update .....	10
3.2.2.20	'ATWT' Wlan TCP throughput test .....	10
3.2.2.21	'ATWU' Wlan UDP test .....	11
3.2.2.22	'ATWL' Wlan SSL client .....	11
3.2.2.23	'ATWW' Wlan Wi-Fi Protected Setup .....	11
3.2.2.24	'ATWZ' Wlan IWPRIV .....	11
3.2.3	System .....	11
3.2.3.1	'ATSC' System Clear OTA Signature .....	11
3.2.3.2	'ATSR' System Recover OTA Signature .....	12
4	Common AT command .....	13
4.1	help .....	13
4.2	Log history .....	13
4.3	Exit .....	13
5	WIFI AT Command Usage .....	14
5.1	Disable/Enable WI-FI .....	14
5.2	Network Connection .....	14
5.3	Wi-Fi Information .....	16
5.4	Start AP .....	17
5.5	Start STA+AP .....	19
5.6	Ping .....	20
5.7	TCP RX/TX Throughput Test .....	21
5.7.1	TCP Test .....	21
5.7.1.1	Receive Throughput Test .....	21
5.7.1.2	Transmit Throughput Test .....	22
5.7.1.3	Transmit and Receive Throughput Test .....	23
5.8	Start Web Server .....	24
5.9	Wi-Fi Simple Config .....	24
5.10	Wi-Fi Protected Setup .....	24
5.11	Start STA+AP .....	24

---

5.12	Set MAC address.....	25
6	System AT Command Usage.....	25
6.1	Clear OTA Signature .....	25
6.2	Restore OTA Signature .....	26

# 1 System Architecture

Realtek Low Power Wi-Fi SoC can be a standalone system with Wi-Fi internet capability or a Wi-Fi interface that connect to an existing MCU.



Realtek CM3 attaches to MCU through UART or SPI, and MCU control Realtek CM3 through AT command.

## 2 Command Format

Command	Delimiter	Payload	Delimiter
AT CMD(4 chars)	=	Req Data	\r
AT CMD(4 chars)	\r		

Response Formats			
Delimiter	return	delimiter	payload
\r\n	OK	\r\n	Data
\r\n	Error type	\r\n	Usage

## 3 AT command

### 3.1 AT command list

AT Command	Description
<b>LOG Common Command</b>	
AT??	Print cmd history
AT--	Exit Log service
<b>WLAN</b>	
ATW0	Network set SSID
ATW1	Network set passphrase
ATW2	Network set Key ID
ATW3	Set Access Point SSID
ATW4	Set Access Point Security Key
ATW5	Set Access Point Channel
ATWA	Activate Access Point
ATWB	Start STA+AP
ATWC	Join a network
ATWD	Disconnect from a network
ATWE	Start web server
ATWI	Ping test
ATWL	SSL client
ATWM	Wlan Wi-Fi promisc
ATWP	Power on/off wifi module
ATWQ	Wi-Fi Simple Config
ATWR	Get RSSI of Associated Network Access Point
ATWS	Scan for Network Access Point
ATWT	TCP T/RX throughput test
ATWU	UDP
ATWW	Wi-Fi Protected Setup
ATWZ	Wlan iwpriv
ATW?	Show network information
<b>System</b>	
ATSC	Clear OTA signature
ATSR	Recover OTA signature

## 3.2 AT command list

### 3.2.1 COMMON

#### *3.2.1.1 'help' Print help message*

Description: Print some commands description and usage  
Command Format: AT??<CR>  
Default Value: None  
Response: TBD

#### *3.2.1.2 'AT??' Print Log History*

Description:  
Command Format: AT??<CR>  
Default Value: None  
Response: TBD

#### *3.2.1.3 'AT--' Exit Log Service*

Description:  
Command Format: AT--<CR>  
Default Value: None  
Response: TBD

### 3.2.2 WLAN

#### *3.2.2.1 'ATW0' Wlan Set Network SSID*

Description:  
Command Format: ATW0=SSID<CR>  
Default Value: None  
Response: None

#### *3.2.2.2 'ATW1' Wlan set Network Passphrase*

Description:  
Command Format: ATW1=password<CR>  
Default Value: None  
Response: None

### ***3.2.2.3 'ATW2' Wlan Set Key ID***

Description:

Command Format: ATW2=Key\_ID<CR>

Default Value: None

Response: None

### ***3.2.2.4 'ATWC' Wlan Join a Network***

Description:

Command Format: ATWC<CR>

Default Value: None

Response: TBD

### ***3.2.2.5 'ATWD' Wlan Disconnect from Network***

Description:

Command Format: ATWD<CR>

Default Value: None

Response: TBD

### ***3.2.2.6 'ATW3' Wlan Set Access Point SSID***

Description:

Command Format: ATW3=AP\_SSID<CR>

Default Value: None

Response: None

### ***3.2.2.7 'ATW4' Wlan Set Access Point Security Key***

Description:

Command Format: ATW4=key<CR>

Default Value: None

Response: None

### ***3.2.2.8 'ATW5' Wlan Set Access Point Channel***

Description:

Command Format: ATW5=channel<CR>

Default Value: None

Response: None



---

### ***3.2.2.9 'ATWA' Wlan Activate Access Point***

Description:

Command Format: ATWA<CR>

Default Value: None

Response: TBD

### ***3.2.2.10 'ATWB' Wlan Activate Access Point mode and Station mode***

Description:

Command Format: ATWB<CR>

Default Value: None

Response: TBD

### ***3.2.2.11 'ATW?' Wlan Show WiFi information***

Description:

Command Format: ATW?<CR>

Default Value: None

Response: TBD

### ***3.2.2.12 'ATWS' Wlan Scan for Network Access Point***

Description:

Command Format: ATWS<CR>

ATWS=num\_channels[channel1, channel2,...]

Default Value: None

Response: TBD

### ***3.2.2.13 'ATWR' Wlan Get RSSI of Associated Network Access Point***

Description:

Command Format: ATWR <CR>

Default Value: None

Response: TBD

### ***3.2.2.14 'ATWM' Wlan Wi-Fi promisc***

Description:

Command Format: ATWM=DURATION\_SECONDS [with\_len]<CR>

Default Value: None

Response: TBD

---

### ***3.2.2.15 'ATWE' Wlan Start Web Server***

Description:

Command Format: ATWE<CR>

Default Value: None

Response: TBD

### ***3.2.2.16 'ATWQ' Wlan Wi-Fi Simple Config***

Description:

Command Format: ATWQ=pin\_code<CR>

Default Value: None

Response: TBD

### ***3.2.2.17 'ATWP' Wlan Power on/off wifi module***

Description:

Command Format: ATWP=0/1<CR>

Default Value: None

Response: TBD

WiFi Power	
Off	0
On	1

### ***3.2.2.18 'ATWI' Wlan ping test***

Description: The parentheses “[]” is required to define repeat count

Command Format: ATWI=xxxx.xxxx.xxxx.xxxx[y/loop]<CR>

Default Value: Count = 5

Response: TBD

### ***3.2.2.19 'ATWO' Wlan OTA update***

Description:

Command Format: ATWO=IP[PORT] <CR>

ATWO= REPOSITORY[FILE\_PATH]<CR>

Default Value: None

Response: TBD

### ***3.2.2.20 'ATWT' Wlan TCP throughput test***

Description:

Command Format: ATWT=[-c/c,xxxx.xxxx.xxxx.xxxx,buf\_len,count] <CR>

ATWT=[-s/s]<CR>

---

	ATWT=[stop]<CR>
Default Value:	None
Response:	TBD

#### ***3.2.2.21 'ATWU' Wlan UDP test***

Description:	
Command Format:	ATWU=[-c/c,xxxx.xxxx.xxxx.xxxx,buf_len,count] <CR> ATWU=[-s/s]<CR>
Default Value:	None
Response:	TBD

#### ***3.2.2.22 'ATWL' Wlan SSL client***

Description:	The parentheses "[ ]" is required to define user name and password if needed
Command Format:	ATWL=SSL_SERVER_HOST[SRP_USER_NAME,SRP_PASSWORD]<CR>
Default Value:	None
Response:	TBD

#### ***3.2.2.23 'ATWW' Wlan Wi-Fi Protected Setup***

Description:	
Command Format:	ATWW=pbcc/pin<CR>
Default Value:	None
Response:	TBD

#### ***3.2.2.24 'ATWZ' Wlan IWPRIV***

Description:	
Command Format:	ATWZ=command[parameter]<CR>
Default Value:	None
Response:	TBD

### **3.2.3 System**

#### ***3.2.3.1 'ATSC' System Clear OTA Signature***

Description:	Clear OTA signature so that boot code load default image.
Command Format:	ATSC<CR>
Default Value:	None
Response:	None

---

### ***3.2.3.2 'ATSR' System Recover OTA Signature***

Description: Recover OTA signature so that boot code load upgraded image(ota image).

Command Format: ATSR<CR>

Default Value: None

Response: None

---

## 4 Common AT command

### 4.1 help

The help command can be used to get description and usage of supported commands.

```
# help
WLAN AT COMMAND SET:
=====
1. Wlan Scan for Network Access Point
   # ATWS
2. Connect to an AES AP
   # ATW0=SSID
   # ATW1=PASSPHRASE
   # ATWC
3. Create an AES AP
   # ATW3=SSID
   # ATW4=PASSPHRASE
   # ATW5=CHANNEL
   # ATWA
4. Ping
   # ATWI=xxx.xxx.xxx.xxx
[MEM] After do cmd, available heap 42752
```

### 4.2 Log history

The “AT??” command prints history of commands which have been made, in order to confirm command information as expected.

```
# AT??
#AT?? match AT??, search cnt 1
[AT]log history:

  ATW3=realtek
  ATW5=1
  ATWA
  ATW?
[MEM] After do cmd, available heap 47896
```

### 4.3 Exit

The “AT--” command makes leaving from UART interactive mode. The stack used by interactive task is released to get more memory.

```
# AT--
AT-- match AT--, search cnt 1
Leave LOG SERVICE
```

## 5 WIFI AT Command Usage

UART interactive mode provides some commands to control Wi-Fi. Users can also implement their commands and add them into command table. The following is the description of built-in commands.

### 5.1 Disable/Enable Wi-Fi

The “ATWP=0/1” commands are used to initialize and de-initialize Wi-Fi driver correspondingly. Before using the functionality of Wi-Fi driver, it needs to be initialized. After Wi-Fi driver is initialized, it will be in station mode. The following are the output when executing “ATWP” commands.

```
# ATWP=0
ATWP match ATWP, search cnt 1
[ATWP]: _AT_WLAN_POWER_OFF]

LwIP DHCP: dhcp stop.
Deinitializing WIFI ...lextra_bus_dma_interrupt(80)
WIFI deinitialized
[MEM] After do cmd, available heap 89080
```

```
# ATWP=1
ATWP match ATWP, search cnt 1
[ATWP]: _AT_WLAN_POWER_ON]

reg 002: 0x3 WIFI ...
reg 01F: 0xea
reg 0b0: 0x0
reg 0b4: 0x0
reg 11c: 0

[_freertos_usleep_os] _freertos_usleep_os: Please Implement micro-second delay
WIFI initialized
[MEM] After do cmd, available heap 47264
```

### 5.2 Network Connection

The “ATWC” command can be used to connect to an access point. To process the connection, an SSID should be set first. Meanwhile a password must be set except in open mode, and a key id is also required for WEP mode.

To disconnect AP, type “ATWD”.

---

**WPA2 mode**

Command sequence: (refer to 3.2.1)

#ATW0=SSID  
#ATW1=passphrase  
#ATWC

```
# ATW0=rtk
ATW0 match ATW0, search cnt 2
[ATW0]: _AT_WLAN_SET_SSID_ [rtk]

[MEM] After do cmd, available heap 47264

# ATW1=12345678
ATW1 match ATW1, search cnt 1
[ATW1]: _AT_WLAN_SET_PASSPHRASE_ [12345678]

[MEM] After do cmd, available heap 47264

# ATWC
ATWC match ATWC, search cnt 2
[ATWC]: _AT_WLAN_JOIN_NET_

Joining BSS ...RTL8195A[Driver]: set ssid [rtk]
RTL8195A[Driver]: start auth
RTL8195A[Driver]: auth success, start assoc
RTL8195A[Driver]: association success(res=2)

wifi_handshake_done_hdl 31
CCConnected after 1261ms.
RTL8195A[Driver]: set group key to hw: alg:4(WEP40-1 WEP104-5 TKIP-2 AES-4) keyid:1
RTL8195A[Driver]: set pairwise key to hw: alg:4(WEP40-1 WEP104-5 TKIP-2 AES-4)

IP address : 192.168.1.100

GGGot IP after 2782ms.

[MEM] After do cmd, available heap 46616
```

**#ATWD**

```
# ATWD
ATWD match ATWD, search cnt 1
[ATWD]: _AT_WLAN_DISC_NET_

Deassociating AP ...
ioctl[SIOCGIWESSID] ssid = NULL, not connected
WIFI disconnected

[MEM] After do cmd, available heap 47376
```

**WEP mode**

Command sequence: (refer to 3.2.1)

#ATW0=SSID  
#ATW1=Password  
#ATW2=Key id  
#ATWC

The WEP key can be 5 ASCII characters for WEP 40 or 13 ASCII characters for WEP 104. The key ID should be 0, 1, 2 or 3. The following is an example to connect network by using WEP 40 with key ID 0.

```
# ATW0=rtk
ATW0 match ATW0, search cnt 2
[ATW0]: _AT_WLAN_SET_SSID_ [rtk]

[MEM] After do cmd, available heap 47480

# ATW1=12345
ATW1 match ATW1, search cnt 1
[ATW1]: _AT_WLAN_SET_PASSPHRASE_ [12345]

[MEM] After do cmd, available heap 47480

# ATW2=0
ATW2 match ATW2, search cnt 2
[ATW2]: _AT_WLAN_SET_KEY_ID_ [0]

[MEM] After do cmd, available heap 47480

# ATWC
ATWC match ATWC, search cnt 2
[ATWC]: _AT_WLAN_JOIN_NET_

Joining BSS ...RTL8195A[Driver]: set ssid [rtk]
RTL8195A[Driver]: set group key to hw: alg:1(WEP40-1 WEP104-5 TKIP-2 AES-4) keyid:0
RTL8195A[Driver]: start auth
RTL8195A[Driver]: auth success, start assoc
RTL8195A[Driver]: association success(res=1)

wifi_connected_hdl 31
CCConnected after 1286ms.

IP address : 192.168.1.100
GSGot IP after 1801ms.

[MEM] After do cmd, available heap 46616
```

## 5.3 Wi-Fi Information

The “ATW?” command can be used to get the information of Wi-Fi driver, including some Wi-Fi statistic, setting, status and memory usage. The following is an example of the output of “ATW?” command when Wi-Fi is disabled. The Wi-Fi status information shows nothing about the Wi-Fi module.

```
# ATW?
ATW? match ATW?, search cnt 1
[ATW?]: _AT_WLAN_INFO_

[MEM] After do cmd, available heap 102752
```

The following is the output of “ATW?” command when Wi-Fi driver is enabled and disconnected. The Wi-Fi status shows the Wi-Fi driver is running without SSID connected. The wlan statistic includes the memory usage that wlan heap used.



```
# ATW?
ATW? match ATW?, search cnt 1
[ATW?]: _AT_WLAN_INFO_

WIFI wlan0 Status: Running
=====
[rltk_wlan_statistic] tx stat: tx_packets=4, tx_dropped=0, tx_bytes=884
[rltk_wlan_statistic] rx stat: rx_packets=10, rx_dropped=10, rx_bytes=4186
[rltk_wlan_statistic] min_free_heap_size=46096, current heap free size=47480
[rltk_wlan_statistic] max_skbbuf_used_num=20, skbbuf_used_num=16
[rltk_wlan_statistic] max_skbdata_used_num=20, skbdata_used_num=16
[rltk_wlan_statistic] max_timer_used_num=7
ioctl[SIIOCGIWESSID] ssid = NULL, not connected

WIFI wlan0 Setting:
=====
MODE => STATION
SSID =>
CHANNEL => 3
SECURITY => OPEN
PASSWORD =>

Interface <wlan0>
=====
MAC => 00:e0:4c:87:00:00
IP => 192.168.1.100
GW => 192.168.1.254

[MEM] After do cmd, available heap 47480
```

The following is the output of “ATW?” Command when Wi-Fi is connected. Wi-Fi setting shows the Wi-Fi driver is in station mode and connecting to a SSID. The connection information in Wi-Fi setting also includes current channel and security.

```
# ATW?
ATW? match ATW?, search cnt 1
[ATW?]: _AT_WLAN_INFO_

WIFI wlan0 Status: Running
=====
[rltk_wlan_statistic] tx stat: tx_packets=4, tx_dropped=0, tx_bytes=884
[rltk_wlan_statistic] rx stat: rx_packets=2, rx_dropped=2, rx_bytes=1236
[rltk_wlan_statistic] min_free_heap_size=46096, current heap free size=46616
[rltk_wlan_statistic] max_skbbuf_used_num=20, skbbuf_used_num=16
[rltk_wlan_statistic] max_skbdata_used_num=20, skbdata_used_num=16
[rltk_wlan_statistic] max_timer_used_num=7

WIFI wlan0 Setting:
=====
MODE => STATION
SSID => rtk
CHANNEL => 3
SECURITY => WEP
KEY INDEX => 0
PASSWORD =>

Interface <wlan0>
=====
MAC => 00:e0:4c:87:00:00
IP => 192.168.1.100
GW => 192.168.1.254

[MEM] After do cmd, available heap 46616
```

## 5.4 Start AP

The Wi-Fi driver can be switched from station mode to AP mode. The wifi\_ap command can be used to start a Wi-Fi AP with indicated SSID, channel and password. If password is not given, this command starts AP in open mode. Otherwise, it starts AP with WPA2 security.

---

Command sequence: (refer to 3.2.1)

#ATW3=SSID

#ATW4=Password (no need for OPEN mode)

#ATW5=Channel

#ATWA

```
# ATW3=bonjour
ATW3 match ATW3, search cnt 2
[ATW3]: _AT_WLAN_AP_SET_SSID_ [bonjour]

[MEM] After do cmd, available heap 47480

# ATW5=1
ATW5 match ATW5, search cnt 1
[ATW5]: _AT_WLAN_AP_SET_CHANNEL_ [channel 1]

[MEM] After do cmd, available heap 47480

# ATWA
ATWA match ATWA, search cnt 1
[ATWA]: _AT_WLAN_AP_ACTIVATE_

LwIP_DHCP: dhcp stop.
Deinitializing WIFI ...lextra_bus_dma_interrupt(80)

WIFI deinitialized
reg 002: 0x3 WIFI ...
reg 01F: 0xea
reg 0b0: 0x0
reg 0b4: 0x0
reg 11c: 0

[_freertos_usleep_os] _freertos_usleep_os: Please Implement micro-second delay

WIFI initialized
Starting AP ...
bonjour started

[MEM] After do cmd, available heap 47840
```

The following is the output of “ATW?” command when AP mode. The Wi-Fi setting shows the Wi-Fi driver is operating in AP mode with SSID, channel, security.

```
# ATW?
ATW? match ATW?, search cnt 1
[ATW?]: _AT_WLAN_INFO_

WIFI wlan0 Status: Running
=====
[rltk_wlan_statistic] tx stat: tx_packets=0, tx_dropped=0, tx_bytes=0
[rltk_wlan_statistic] rx stat: rx_packets=0, rx_dropped=0, rx_bytes=0
[rltk_wlan_statistic] min_free_heap_size=46936, current heap free size=47896
[rltk_wlan_statistic] max_skbbuf_used_num=17, skbbuf_used_num=16
[rltk_wlan_statistic] max_skbdata_used_num=17, skbdata_used_num=16
[rltk_wlan_statistic] max_timer_used_num=8

WIFI wlan0 Setting:
=====
MODE => AP
SSID => bonjour
CHANNEL => 1
SECURITY => OPEN
PASSWORD =>

Interface (wlan0)
=====
MAC => 00:e0:4c:87:00:00
IP => 192.168.1.1
GW => 192.168.1.1

[MEM] After do cmd, available heap 47896
```

To switch back from AP to STA mode, set Wi-Fi connection command set (refer to 5.2).

## 5.5 Start STA+AP

The Wi-Fi driver can start station mode and AP mode concurrently. The “ATWB” command can be used to start a Wi-Fi AP with indicated SSID, channel and password and start a station mode together. If password is not given, this command starts AP in open mode. Otherwise, it starts AP with WPA2 security. And the Wi-Fi connection command set (refer to 5.2) is used to connect with an AP.

Command sequence: (refer to 3.2.1)

Start AP:

#ATW3=SSID

#ATW4=Password (no need for OPEN mode)

#ATW5=Channel

#ATWB

Connect to an AP:

#ATW0=SSID

#ATW1=Password

#ATW2=Key\_id(only needed for WEP mode)

#ATWC

## 5.6 Ping

The “ATWI” command continues sending 5 ping packets, each in one second, to an indicated IP address. Please note that if DHCP client is not enabled, it is required to pre-configured default IP in main.h. It is useful when testing the network connection.

```
# ATWI=192.168.1.254
#ATWI match ATWI, search cnt 1
[ATWI]: _AT_WLAN_PING_TEST_
arg: 192.168.1.254hello
[ATWI]Target address: 192.168.1.254
[ATWI]Repeat Count: 5

[ping_test] PING 192.168.1.254 120(148) bytes of data

[ping_test] 128 bytes from 192.168.1.254: icmp_seq=1 time=11 ms
[ping_test] 128 bytes from 192.168.1.254: icmp_seq=2 time=10 ms
[ping_test] 128 bytes from 192.168.1.254: icmp_seq=3 time=9 ms
[ping_test] 128 bytes from 192.168.1.254: icmp_seq=4 time=12 ms
[ping_test] 128 bytes from 192.168.1.254: icmp_seq=5 time=7 ms
[MEM] After do cmd, available heap 46616
```

To ping y packets, type “ATWI=xxx.xxx.xxx.xxx[y]”(no any space, and [] is required)

```
# ATWI=192.168.1.254[2]
#ATWI match ATWI, search cnt 1
[ATWI]: _AT_WLAN_PING_TEST_
arg: 192.168.1.254[2]hello
[ATWI]Target address: 192.168.1.254
[ATWI]Repeat Count: 2

[ping_test] PING 192.168.1.254 120(148) bytes of data

[ping_test] 128 bytes from 192.168.1.254: icmp_seq=1 time=9 ms
[ping_test] 128 bytes from 192.168.1.254: icmp_seq=2 time=13 ms
[MEM] After do cmd, available heap 46616
```

To ping continuously, type “ATWI=xxx.xxx.xxx.xxx[loop]”. Please note that currently, exiting infinite ping loop by UART command is not supported yet.

```
# ATWI=192.168.1.254[loop]
#ATWI match ATWI, search cnt 1
[ATWI]: _AT_WLAN_PING_TEST_
arg: 192.168.1.254[loop]hello
[ATWI]Target address: 192.168.1.254
[ATWI]Repeat Count: loop

[ping_test] PING 192.168.1.254 120(148) bytes of data

[ping_test] 128 bytes from 192.168.1.254: icmp_seq=1 time=10 ms
[ping_test] 128 bytes from 192.168.1.254: icmp_seq=2 time=9 ms
[ping_test] 128 bytes from 192.168.1.254: icmp_seq=3 time=7 ms
[ping_test] 128 bytes from 192.168.1.254: icmp_seq=4 time=10 ms
[ping_test] 128 bytes from 192.168.1.254: icmp_seq=5 time=8 ms
[ping_test] 128 bytes from 192.168.1.254: icmp_seq=6 time=5 ms
[ping_test] 128 bytes from 192.168.1.254: icmp_seq=7 time=6 ms
[ping_test] 128 bytes from 192.168.1.254: icmp_seq=8 time=14 ms
[ping_test] 128 bytes from 192.168.1.254: icmp_seq=9 time=5 ms
[ping_test] 128 bytes from 192.168.1.254: icmp_seq=10 time=8 ms
[ping_test] 128 bytes from 192.168.1.254: icmp_seq=11 time=15 ms
[ping_test] 128 bytes from 192.168.1.254: icmp_seq=12 time=13 ms
[ping_test] 128 bytes from 192.168.1.254: icmp_seq=13 time=13 ms
[ping_test] 128 bytes from 192.168.1.254: icmp_seq=14 time=12 ms
[ping_test] 128 bytes from 192.168.1.254: icmp_seq=15 time=16 ms
[ping_test] 128 bytes from 192.168.1.254: icmp_seq=16 time=13 ms
[ping_test] 128 bytes from 192.168.1.254: icmp_seq=17 time=17 ms
[ping_test] 128 bytes from 192.168.1.254: icmp_seq=18 time=9 ms
```

---

## 5.7 TCP RX/TX Throughput Test

### 5.7.1 TCP Test

TCP transmit and receive throughput can be measured by iperf.exe tool which you can get from \$sdk/tools/iperf.exe.

#### 5.7.1.1 Receive Throughput Test

Receive test measures receive throughput of the development board. Start TCP server in the development board, listen to port 5001 and wait for connection from iperf client. Iperf on the Windows platforms connects to the TCP server via AP and transmits data to it. Iperf client running on the Windows platforms computes bytes of data transmitted, and print it out every 1 second. A sample session is illustrated as bellow:

Type the following command to start TCP server on the console of development board:

```
# ATWT=[-s]
```

The “-s” command-line option starts a TCP server.

```
# ATWT=[-s]
ATWT match ATWT, search cnt 1
[ATWT]: _AT_WLAN_TCP_TEST_

[MEM] After do cmd, available heap 46016

#
TCP: Start tcp Server!
TCP: Create server socket 0

TCP: Bind successfully.
TCP: Listen port 5001
```

Type the following command to start Iperf client on Windows platforms:

```
~:> iperf .exe -c 192.168.1.100 -i 1 -t 60
```

The “-c” command-line option means starting a TCP client and connecting to “192.168.1.100”, “-i” is seconds between periodic bandwidth reports, “-t” is time in seconds to transmit for (default 10 seconds).

```
sd9@sd9-ThinkPad-T410:~$ iperf -c 192.168.1.100 -i 1 -t 60
-----
Client connecting to 192.168.1.100, TCP port 5001
TCP window size: 43.8 KByte (default)
-----
[ 3] local 192.168.1.103 port 40280 connected with 192.168.1.100 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 3] 0.0- 1.0 sec   384 KBytes  3.15 Mbits/sec
[ 3] 1.0- 2.0 sec   256 KBytes  2.10 Mbits/sec
[ 3] 2.0- 3.0 sec   256 KBytes  2.10 Mbits/sec
[ 3] 3.0- 4.0 sec   256 KBytes  2.10 Mbits/sec
[ 3] 4.0- 5.0 sec   256 KBytes  2.10 Mbits/sec
```

### 5.7.1.2 Transmit Throughput Test

Transmit test measures the transmission throughput of the development board. Start TCP Client in the development board and connect to Iperf server on the Windows platforms via AP. Iperf server works on the default port 5001 and should not be changed since TCP client is fixed to connect with this port. TCP client send 10000 packets with length 1460 one time as default. Iperf server running on the Windows platforms computes bytes of data received, and print it out every 1 second. A sample session is illustrated as below:

Type the following command to start Iperf server on Windows platforms:

```
~:> iperf.exe -s -i 1
```

The “-s” command-line option starts a TCP server, “-i” is seconds between periodic bandwidth reports.

```
sd9@sd9-ThinkPad-T410:~$ iperf -s -i 1
-----
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)
-----
[ 4] local 192.168.1.103 port 5001 connected with 192.168.1.100 port 4100
[ ID] Interval      Transfer    Bandwidth
[ 4] 0.0- 1.0 sec   211 KBytes  1.73 Mbits/sec
[ 4] 1.0- 2.0 sec   182 KBytes  1.49 Mbits/sec
[ 4] 2.0- 3.0 sec   185 KBytes  1.51 Mbits/sec
[ 4] 3.0- 4.0 sec   149 KBytes  1.22 Mbits/sec
[ 4] 4.0- 5.0 sec   214 KBytes  1.75 Mbits/sec
[ 4] 5.0- 6.0 sec   214 KBytes  1.75 Mbits/sec
[ 4] 6.0- 7.0 sec   205 KBytes  1.68 Mbits/sec
[ 4] 7.0- 8.0 sec   173 KBytes  1.42 Mbits/sec
[ 4] 8.0- 9.0 sec   199 KBytes  1.63 Mbits/sec
[ 4] 9.0-10.0 sec   229 KBytes  1.87 Mbits/sec
[ 4] 10.0-11.0 sec  196 KBytes  1.61 Mbits/sec
[ 4] 11.0-12.0 sec  211 KBytes  1.73 Mbits/sec
```

Type the following command to start TCP client on the development board:

```
# ATWT=[-c,192.168.0.100,1460,10000]
```

The “-c” command-line option starts a TCP client, “192.168.0.100” is IP address of the Windows platforms, “1460” is the length of packet to be transmitted, “10000” is the number of packets transmitted to Iperf Server. Please note that packet length is no more than 4300 .

```
# ATWT=[-c,192.168.1.103,1460,10000]
ATWT match ATWT, search cnt 1
[ATWT]: _AT_WLAN_TCP_TEST_

[MEM] After do cmd, available heap 46016

#
TCP: Start tcp client!
TCP: ServerIP=192.168.1.103 port=5001.
TCP: Create socket 0.
TCP: Connect server successfully.[]
```

Stop TCP test by typing the following command:

```
#ATWT=[stop]
```

```
ATWT=[stop]
ATWT match ATWT, search cnt 1
[ATWT]: _AT_WLAN_TCP_TEST_

[MEM] After do cmd, available heap 44264

#
TCP: Sent u packets successfully.
TCP: Tcp client stopped![]
```

### **5.7.1.3 Transmit and Receive Throughput Test**

The concurrent throughput test measures receive and transmit throughput concurrently. The development board run “ATWT=[-s]” to start a TCP server and communicate with iperf client on Windows platform, run “ATWT=[-c,192.168.0.100,1460,100000]” to start a TCP client and communicate with iperf server on Windows platform. A sample session is illustrated as bellow:

Step 1: Start Iperf server on Windows platforms:

```
~:> iperf.exe -s -i 1
```

Step 2: Start TCP server on the development board:

```
# ATWT=[-s]
```

Step 3: Start Iperf client on Windows platforms:

```
~:> iperf.exe -c 192.168.1.103 -i 1 -t 100
```

Step 4: Start TCP client on the development board:

```
# ATWT=[-c,192.168.1.100,1460,100000]
```



```

^Csd9@sd9-ThinkPad-T410:~$ iperf -s -l 1
.....
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)
.....
[ 4] local 192.168.1.103 port 5001 connected with 192.168.1.100 port 4097
[ ID] Interval      Transfer    Bandwidth
[ 4] 0.0- 1.0 sec    180 KBytes  1.48 Mbits/sec
[ 4] 1.0- 2.0 sec    117 KBytes  960 Kbits/sec
[ 4] 2.0- 3.0 sec    195 KBytes  1.60 Mbits/sec
[ 4] 3.0- 4.0 sec    96.7 KBytes 792 Kbits/sec
[ 4] 4.0- 5.0 sec    196 KBytes 1.61 Mbits/sec
[ 4] 5.0- 6.0 sec    168 KBytes 1.38 Mbits/sec

^Csd9@sd9-ThinkPad-T410:~$ iperf -c 192.168.1.100 -l 1 -t 60
.....
Client connecting to 192.168.1.100, TCP port 5001
TCP window size: 43.8 KByte (default)
.....
[ 3] local 192.168.1.103 port 40283 connected with 192.168.1.100 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 3] 0.0- 1.0 sec    128 KBytes  1.05 Mbits/sec
[ 3] 1.0- 2.0 sec    128 KBytes  1.05 Mbits/sec
[ 3] 2.0- 3.0 sec    128 KBytes  1.05 Mbits/sec

```

## 5.8 Start Web Server

The “ATWE” command can be used to start webserver. Web server works only after Wi-Fi driver switched to AP mode or concurrent AP mode. After client associated with the AP and get right IP address, the client PC can open web browser and enter <http://192.168.1.1> (<http://192.168.1.1> in AP mode or <http://192.168.43.1> in concurrent AP mode) to get or set AP settings. For details, please refer to the document UM0014 Realtek web server user guide.pdf.

## 5.9 Wi-Fi Simple Config

This “ATWQ” command provides a simple way for device to associate to AP. For details, please refer to the document AN0011 Realtek wlan simple configuration.pdf.

## 5.10 Wi-Fi Protected Setup

The “ATWW” command provides another simple way for device to associate to AP. After pressing WPS button on the AP, execute “ATWW=pbw” in the command line, then the device will automatically associate with the AP. PIN method also supported. Please refer to the document AN0011 Realtek wlan simple configuration.pdf for more detail.

## 5.11 Start STA+AP

The Wi-Fi driver can start station mode and AP mode concurrently. The “ATWB” command can be used to start a Wi-Fi AP with indicated SSID, channel and password and start a station mode together. If password is not given, this command starts AP in open mode. Otherwise, it starts



---

AP with WPA2 security. And the Wi-Fi connection command set (refer to 5.2) is used to connect with an AP.

Command sequence: (refer to 3.2.1)

Start AP:

#ATW3=SSID

#ATW4=Password (no need for OPEN mode)

#ATW5=Channel

#ATWB

Connect to an AP:

#ATW0=SSID

#ATW1=Password

#ATW2=Key\_id(only needed for WEP mode)

#ATWC

## 5.12 Set MAC address

The ATWZ command can be used to read/write MAC address. There are two examples for reading and writing MAC address as below:

Read MAC address:

#ATWZ=read\_mac

Write MAC address:

#ATWZ=write\_mac[00e04c870102]

## 6 System AT Command Usage

### 6.1 Clear OTA Signature

Read back OTA signature value. The value of 81958711 at first time shows OTA image is *valid*.

After clear the signature, read back OTA signature again and it is 00000000.

```
#ATSC

[ATSC]: _AT_SYSTEM_CLEAR_OTA_SIGNATURE_
OTA offset = 0x00044000
Signature = 81958711
Signature = 00000000
Clear OTA signature success.
```

## 6.2 Restore OTA Signature

Read back OTA signature value. The value of 00000000 at first time shows OTA image is *invalid*. After set OTA signature to valid, (that is, 81958711), write this value to flash and read back again for double check.

```
#ATSR

[ATSR]: _AT_SYSTEM_RECOVER_OTA_SIGNATURE_
OTA offset = 0x00044000
Signature = 00000000
Signature = 81958711
Recover OTA signature success.
```