



Realtek Wlan Simple Configuration

This document provides simple ways for a low power wi-fi device to associate to AP. These kinds of devices usually does not have rich interface, so there should be a way for setting ssid and related wi-fi configuration into device easily.

Table of Contents

1	Introduction	(
"	Solution \$* +evice AP mode , Smart Phone App.....	(
(Solution "" -endor Customi.e Simple Config.....	(
(.\$	STA mode , Promiscuous mode	(
/	Solution (* Realtek Simple Config	%
/\$	Simple Config Osage	1
/\$.\$	Install Android APP	1
/\$."	Config Progress	&
/\$. (Configuration without P)2 code	&
/\$./	Configuration with P)2 Code	3
/"	Simple Config Testing	3
/(Simple Config Resources	'
%	Solution (* Realtek WPS Config.....	\$#
%. \$	WPS Config Osage	\$#
%. \$.\$	Config Progress	\$#
%. \$."	Configuration with P4C	\$#
%. \$. (Configuration with P)2.....	\$\$
%. "	WPS Config Testing	\$\$

1 Introduction

This document provides simple ways for a low power wi-fi device to associate to AP. These kinds of devices usually does not have rich interface, so there should be a way for setting ssid and related wi-fi configuration into device easily.

2 Solution 1: Device AP mode + Smart Phone App

Device is configured at AP mode during configuration phase, and user connects to the device with smart phone and use an app to transfer ssid/key information to the device. After device receives ssid/key information, it is set to STA mode and tries to associate to the AP.

3 Solution 2: Vendor Customize Simple Config

3.1 STA mode + Promiscuous mode

A customized configuration implementation sometime may be required. That may use the data encoded in a certain pattern, and needs to collect these data for Wi-Fi configuration before device can connect to network. To support retrieving all the data on air, promiscuous mode is provided by Wi-Fi driver. The `wifi_set_promisc` function in `wificonf` API is provided to enable/disable the promiscuous mode of Wi-Fi driver and setup the callback function to handle the data information reported by the Wi-Fi driver under promiscuous mode. Enabling promiscuous mode will make network disconnected. The following is the definition of `wifi_set_promisc` function.

```
int
wifi_set_promisc (
    unsigned char enabled,
    void (*callback)(unsigned char*, unsigned char*, unsigned int),
    unsigned char len_used
)
```

The callback function set to promiscuous mode may be implemented to save the reported information to a user data space. These stored data could be used to support the Wi-Fi configuration. When Wi-Fi driver in promiscuous mode receives a data frame, the data frame may be not decrypted. Therefore, the data payload is not useful and reporting data payload will be meaningless. Wi-Fi driver will only report the destination MAC address, source MAC address and packet length of the Ethernet frame via this callback function. Therefore, the user implemented callback function should match the following definition. When the third parameter in `wifi_set_promisc` function is set to 0 to make not report packet length, the packet length reported to callback function will always be 0.

```
void
callback (
    unsigned char *da,
    unsigned char *sa,
    unsigned int len
)
```

A user space sample codes to work with promiscuous mode is provided in `promisc_test.c`. The “ATWM” command can be used to start the `promisc_test` function to show the stored data information reported by the Wi-Fi driver in promiscuous mode.

The `promisc_test` function will enable Wi-Fi driver promiscuous mode and switch Wi-Fi channel by `wifi_set_channel` function provided by `wificonf` API. It will continue to stay in channel 1 to 13 with the indicated duration in seconds for each channel. In `promisc_test` function, a user callback function example, named `promisc_callback`, is set by `wifi_set_promisc` function. The following is the execution of “ATWM” command with duration of 3 seconds for each channel with/without reporting each packet length.

```
# ATWM=3[with_len]
#ATWM match ATWM, search cnt 1
[ATWM]: _AT_WLAN_PROMISC_
```

Report packet length

```
Switch to channel<1>
DA: c4 6e 1f 58 42 bc, SA: 00 e0 4c 87 00 00, len=6
DA: 01 00 0c cc cc cd, SA: 24 e9 b3 23 f0 92, len=56
DA: ff ff ff ff ff ff, SA: 00 1f 3b 81 48 89, len=47

Switch to channel<2>
DA: ff ff ff ff ff ff, SA: 00 1f 3b 81 48 89, len=154

Switch to channel<3>

Switch to channel<4>
DA: 01 00 0c cc cc cd, SA: 24 e9 b3 23 f0 92, len=56

Switch to channel<5>
DA: 34 bd c8 ac 14 20, SA: d0 22 be 22 00 a9, len=6
DA: 34 bd c8 ac 14 20, SA: d0 22 be 22 00 a9, len=6
```

```
# ATWM=3
#M match ATWM, search cnt 1
[ATWM]: _AT_WLAN_PROMISC_
```

Not report packet length

```
Switch to channel<1>
DA: 01 00 5e 00 00 fb, SA: c4 6e 1f 58 42 bc, len=0
DA: 01 00 0c cc cc cd, SA: 24 e9 b3 23 f0 92, len=0
DA: 01 00 5e 00 00 fb, SA: c4 6e 1f 58 42 bc, len=0
DA: 01 00 5e 00 00 fb, SA: c4 6e 1f 58 42 bc, len=0
DA: ff ff ff ff ff ff, SA: ac cf 5c 61 eb 27, len=0
DA: 01 00 0c cc cc cd, SA: 24 e9 b3 23 f0 92, len=0

Switch to channel<2>

Switch to channel<3>
DA: 01 00 0c cc cc cd, SA: 24 e9 b3 23 f0 92, len=0
DA: 01 00 0c cc cc cd, SA: 24 e9 b3 23 f0 92, len=0

Switch to channel<4>
DA: 20 10 7a 7c bb 34, SA: 5c f3 fc 3e a9 b8, len=0
```

4 Solution 3: Realtek Simple Config

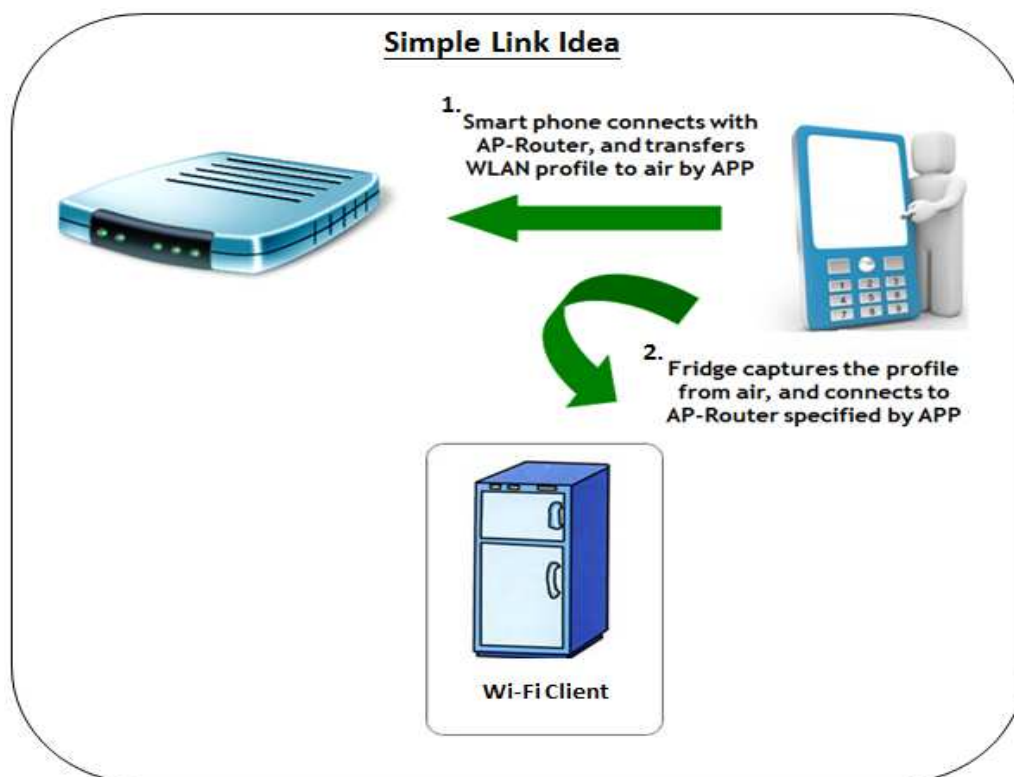
It provides a way for device to associate to AP easily. The user experience is clicking a button on the Simple Config app, and everything is done. The concept of simple Config is shown

```
"#$%-#&-#"
```

```
"#$%-#(-#"
```

```
%
```

as the following figure.



To setup the environment: First, install simple config app on android phone, and second, configure device in log service mode (UART console mode). The detail of application usage and platform setting are introduced in the following subsections.

4.1 Simple Config Usage

4.1.1 Install Android APP

Simple Config is an android apk file. It can be installed on any Android devices. To install Simple Config app, copy the apk file to the phone (via usb or micro sd), and then click on the apk file to install it. Please get apk file SimpleConfigWizard_107.apk in tools\simple_config_wizard\Android\.

4.1.2 Config Progress

Two methodologies are provided by Simple Config. The first one methodology ***configures device without PIN*** code and the second methodology ***configures device with PIN code***. The role of PIN code is just like the KEY.

After triggering simple config process, device and app on phone stays in config mode for **one** minutes. The time between triggering simple config on device and app on phone should be as short as possible.

If user selects the methodology of configuring without PIN code, default PIN code is used to handle the hash, encryption and decryption during the simple Config process. The process is described in 4.1.3

If user selects the methodology of configuring with PIN code, the inputted PIN code will be used during the hash, encryption and decryption during the simple Config process. The process is described in 4.1.4

For now, the default pin code is "57289961" and the PIN code length be limited to 8 ~ 64-digit numerics. If the user enter the PIN code length less than eight ,the "ATWQ" command will auto padding to 8-digit numeric by zero. Ex: "1234" will be padding to "12340000". If the PIN code length is greater than 64. It only takes the first 64 digit numerics as the input.

4.1.3 Configuration without PIN code

Note: The custom pin code must be **entered via "ATWQ" command** first. Please refer to the usage guideline in 4.2.

For the operations on simple config wizard android apk, please refer to document:
tools\simple_config_wizard\Android\
Realtek_Android_Simple_Config_Wizard_User_Guide_V1.pdf

4.1.4 Configuration with PIN Code

PIN code is just like a password for each device. In the SDK, PIN code is pre-configured in .h file, while in real implementation; it can be pre-configured in flash, or in the format of QR code.

For the operations on simple config wizard android apk, please refer to document:

tools\simple_config_wizard\Android\

Realtek_Android_Simple_Config_Wizard_User_Guide_V1.pdf

4.2 Simple Config Testing

<pre>/* For Simple Link */ #define CONFIG_INCLUDE_SIMPLE_CONFIG 1</pre>

To enable simple config, set CONFIG_INCLUDE_SIMPLE_CONFIG flag in platform_opts.h and use the command “ATWQ”. Our platform will run in configuration without PIN code.

If user wants to configure the app with PIN code, they can use the command “ATWQ” follow by custom PIN code. The entered PIN code into the Android APP has to be the same with the command line PIN code otherwise the profile cannot be decoded correctly and simple config process gets a fail result. Once Simple config is triggered (by command line), device retrieves AP's profile from encoded data packets. After getting correct profile, the device connects to the AP according to profile information. The following figures show the command usage that device associate to AP by configuration with PIN Code then use the ping test command to try the connection between device and AP.


```
# ATWQ=12345678
# Command ATWQ
ATWQ match ATWQ, search cnt 1
[ATWQ]: _AT_WLAN_SIMPLE_CONFIG_

ioctl[SIIOCGIWESSID] ssid = NULL, not connected
ioctl[SIIOCGIWESSID] ssid = NULL, not connected
ioctl[SIIOCGIWESSID] ssid = NULL, not connected
===== Connection Information =====

wifi->security_type = RTW_SECURITY_OPEN
wifi.password =
wifi.password_len = 0
wifi.ssid = rtk
wifi.ssid_len = 3

===== start to connect target AP =====
RTL8195A[Driver]: set ssid [rtk]
RTL8195A[Driver]: start auth
RTL8195A[Driver]: auth success, start assoc
RTL8195A[Driver]: association success(res=1)

IP address : 192.168.1.100

WIFI wlan0 Setting:
=====
MODE => STATION
SSID => rtk
CHANNEL => 1
SECURITY => OPEN
PASSWORD =>

[MEM] After do cmd, available heap 52936

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

[ping_test] PING 192.168.1.254 128(148) bytes of data
[ping_test] 128 bytes from 192.168.1.254: icmp_seq=1 time=2 ms
[ping_test] 128 bytes from 192.168.1.254: icmp_seq=2 time=2 ms
[ping_test] 128 bytes from 192.168.1.254: icmp_seq=3 time=2 ms
[ping_test] 128 bytes from 192.168.1.254: icmp_seq=4 time=3 ms
[ping_test] 128 bytes from 192.168.1.254: icmp_seq=5 time=2 ms
[MEM] After do cmd, available heap 52936
```

4.3 Simple Config Resources

Please check tools\simple_config_wizard

- Android: Release_SimpleConfigWizard_v107
- iOS: Release_iOS_SimpleConfigWizard_v103

5 Solution 3: Realtek WPS Config

It provides a way for device to associate to AP easily. The user's experience is clicking a button on the AP or entering the device PIN code on AP's website, and trigger WPS command then everything is done.

To setup the environment: configure device in log service mode (UART console mode). The detail of application usage and platform setting are introduced in the following subsections.

5.1 WPS Config Usage

5.1.1 Config Progress

Two methodologies are provided by WPS Config. The first one methodology ***configures device with PBC*** and the second methodology ***configures device with PIN***. The role of PIN code is just like the KEY.

After triggering wps config process, device stays in config mode for **two** minutes (i.e walking time). During this period, device will try to find out one AP which be triggered WPS for connection. In PBC mode, the time between triggering wps config on device and press button on AP should be as short as possible. In PIN mode, the time between triggering wps config on device and enter the PIN code on AP's website should be as short as possible, too.

If user selects the methodology of configuring with PBC, default PIN code is used to handle the hash, encryption and decryption during the WPS Config process. The process is described in 4.1.3

If user selects the methodology of configuring with PIN, the inputted PIN code will be used during the hash, encryption and decryption during the WPS Config process. The process is described in 4.1.4. For now, the **device pin code** is "75798024" and the PIN code length must be 8-digit numerics.

5.1.2 Configuration with PBC

To enable Configuration with PBC, user should do the following steps. These steps do not need to follow the order. Users only need to ensure that all steps have been completed.

Step 1: Click the WPS button on AP to trigger AP in the WPS PBC mode.

Step 2: Trigger device WPS PBC command in log service mode.

5.1.3 Configuration with PIN

To enable Configuration with PIN, user should do the following steps. These steps do not need to follow the order. Users only need to ensure that all steps have been completed.

Step 1: Enter the device PIN code on AP's website to trigger AP in the WPS PIN mode.

Step 2: Trigger device WPS PIN command in log service mode.

5.2 WPS Config Testing

#define CONFIG_ENABLE_WPS 1

To enable WPS config, set CONFIG_ENABLE_WPS flag in platform_opts and use the command "ATWW=pbcc/pin" . Our platform will run in configuration with PBC/PIN.

For example, if user want to configure the device with PIN. They should use the command "ATWW" follow by "pin" string. The string follow by the wps command will indicate our platform to process the WPS in which mode.

The entered PIN code into the AP's website has to be the same with the command line popped device PIN code otherwise the WPS config process gets a fail result. Once WPS config is triggered (by command line), device retrieves AP's credential from WPS process. After getting correct AP's credential, the device connects to the AP according to credential information. The following figures show the command usage that device associate to AP by configuration with PIN.

```
# ATWW=pin
# Command ATWW
ATWW match ATWW, search cnt 2
[ATWW]: _AT_WLAN_WPS_

ioctl[ISIOCGIWESSID] ssid = NULL, not connected
[pin]device pin code = 75798024

[pin]find out first trigger wps AP = rtk
Start WPS PIN

[pin]target ssid = rtk
RTL8195A[Driver]: set ssid [rtk]
RTL8195A[Driver]: start auth
RTL8195A[Driver]: auth success, start assoc
RTL8195A[Driver]: association success(res=3)
RTL8195A[Driver]: WPS: Building Message M1
RTL8195A[Driver]: Receive WPS_M2
RTL8195A[Driver]: WPS: Building Message M3
RTL8195A[Driver]: Receive WPS_M4
RTL8195A[Driver]: WPS: Building Message M5
RTL8195A[Driver]: Receive WPS_M6
RTL8195A[Driver]: WPS: Building Message M7
RTL8195A[Driver]: Receive WPS_M8
RTL8195A[Driver]: WPS: Received M8
RTL8195A[Driver]: WPS: Building Message WSC_Done

[pin]get credential from queue
[pin]dev_cred.ssid = rtk
[pin]dev_cred.key =
[pin]dev_cred.auth_type = u
[pin]dev_cred.encr_type = u
wps_config_wifi_setting
wps_wifi.ssid = rtk
wps_wifi.ssid_len = 3
security_type = RTW_SECURITY_OPEN
wps_wifi.security_type = 0
wps_wifi.password =
wps_wifi.password_len = 0
===== wifi_certificate_info =====
```

```
===== wifi_certificate_info =====
```

```
wps_wifi.ssid = rtk
```

```
security_type = 0
```

```
wps_wifi.password =
```

```
ssid_len = 3
```

```
password_len = 0
```

```
RTL8195A[Driver]: set ssid [rtk]
```

```
RTL8195A[Driver]: start auth
```

```
RTL8195A[Driver]: auth success, start assoc
```

```
RTL8195A[Driver]: association success(res=3)
```

```
IP address : 192.168.1.100
```

```
WIFI wlan0 Setting:
```

```
=====
```

```
MODE => STATION
```

```
SSID => rtk
```

```
CHANNEL => 2
```

```
SECURITY => OPEN
```

```
PASSWORD =>
```

```
[MEM] After do cmd, available heap 52912
```