

Quick Start Guide for starting Soft-AP mode

Realtek

(A) How to start Soft-AP mode:

- (1) Disable network management or other wireless tools, e.g. wpa_supplicant
- (2) Uncompress the driver and then compile the driver

./make

P.S. If the driver uses CFG80211, there are several steps below need to do:

I. If the driver package is for single interface

1. Uncomment the definition “`//#define CONFIG_IOCTL_CFG80211`” of the file “`include/autoconf.h`” to “`#define CONFIG_IOCTL_CFG80211`”
2. If the Linux kernel version is greater than 3.2.0 (`kernel >= 3.2.0`), user must uncomment the definition “`//#define RTW_USE_CFG80211_STA_EVENT`” of file `include/autoconf.h` to “`#define RTW_USE_CFG80211_STA_EVENT`”

II. If the driver package is for multiple interfaces

1. user should modify the definition in the “`autoconf_XXX_yyy_linux.h`” file but not “`include/autoconf.h`”. The “`XXX`” is IC type and the “`yyy`” is interface type. For example, the IC type is RTL8192C and the interface type is USB, the file name is “`autoconf_rtl8192c_usb_linux.h`”.

III. If the driver uses CFG80211 and the Linux kernel version $\geq 3.2.0$, the SOFTAP must use the

“`wpa_supplicant_8_jb_4.2_rtw_zzzzz.20130821.tar.gz`” or

“`wpa_supplicant_8_kk_4.4_rtw_zzzzz.20140220.tar.gz`” package. In contrast, the SOFTAP should use

“`wpa_supplicant_hostapd-0.8_rtw_zzzzz.20130812.tar.gz`” package for

WEXT. The `zzzz` is version number. If the driver using CFG80211 but `kernel < 3.2.0`, `wpa_supplicant` are not available in driver package so far, and please contact us.

(3) **insmod 8192cu.ko**

(4) **ifconfig wlan0 up**

(5) **ifconfig wlan0 192.168.0.1** (using the static ip for testing)

(6) Compile SOFTAP, unpack `wpa_supplicant_hostapd-0.8_rtw_20120803.zip` in the folder (`wpa_supplicant_hostapd-0.8\hostapd`)

./make

(7) start hostapd daemon:

./hostapd rtl_hostapd.conf -B

(B) Configure file for Soft-AP mode setting:

(1) rtl_hostapd.conf is the configure file for functions setting.

(2) The major variable setting in the rtl_hostapd.conf configure file,

(i) basic configuration

```
interface=wlan0
```

```
ssid=rtwap
```

```
# channel 1-14 is 2.4 GHz ; channel 36, 40, 44, 46, 48, 52, 56, 60,
```

```
# 64, 100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140, 149,
```

```
# 153, 157, 161 is 5GHz
```

```
# The channels that are available for use in a particular country differ
```

```
# according to the regulations of that country.
```

```
channel=6
```

```
# Operation mode (a = IEEE 802.11a, b = IEEE 802.11b, g = IEEE
```

```
# 802.11g, Default: IEEE 802.11b )
```

```
hw_mode=g
```

```
#If the wireless interface is included in a bridge,
```

```
#an additional configuration parameter, bridge, is needed
```

```
bridge=br0
```

```
# set "driver=rtl871xdrv" for WEXT, or "driver=nl80211" for
```

```
# CFG80211
```

```
driver=rtl871xdrv
```

(ii) security mode configuration

```
# This field is a bit field that can be used to enable WPA
```

```
# (IEEE 802.11i/D3.0)
```

```
# and/or WPA2 (full IEEE 802.11i/RSN):
```

```
# bit1 = IEEE 802.11i/RSN (WPA2) (dot11RSNAEnabled)
```

```
wpa=2
```

```
# wpa_passphrase=secret passphrase
```

```
wpa_passphrase=87654321
```

```
# Set of accepted key management algorithms
# (WPA-PSK, WPA-EAP, or both).
wpa_key_mgmt=WPA-PSK

# Set of accepted cipher suites (encryption algorithms)
# for pairwise keys
wpa_pairwise=CCMP
```

(iii) IEEE 802.11n related configuration

```
# ieee80211n: Whether IEEE 802.11n (HT) is enabled
# 0 = disabled (default)
# 1 = enabled
ieee80211n=1

# ht_capab: HT capabilities (list of flags)
# Supported channel width set: [HT40-] = both 20 MHz and 40 MHz
# with secondary channel below the primary channel;
# [HT40+] = both 20 MHz and 40 MHz with secondary channel upon
# the primary channel
# Note: There are limits on which channels can be used with HT40- and
# HT40+. Following table shows the channels that may be available for
# HT40- and HT40+ use per IEEE 802.11n Annex J:
# freq          HT40-          HT40+
# 2.4 GHz       5-13           1-7 (1-9 in Europe/Japan)
# 5 GHz         40,48,56,64    36,44,52,60
# Short GI for 20 MHz: [SHORT-GI-20] (disabled if not set)
# Short GI for 40 MHz: [SHORT-GI-40] (disabled if not set)
ht_capab=[SHORT-GI-20][SHORT-GI-40][HT40+]
```

(iv) Check the station connected to softap using hostapd_cli:

```
./hostapd_cli all_sta
```

(v) How to start WPS process as internal registrar?

1. for PIN code = 12345670

```
./hostapd_cli wps_pin any 12345670
```

2. for PBC

```
./hostapd_cli wps_pbc
```

(C) How to get the best channel?

1. Assume the WLAN interface is wlan0 and the IC is RTL8192DU-VS:

```
ifconfig wlan0 up  
iwlist wlan0 scan  
iwlist wlan0 scan  
iwlist wlan0 scan  
cat /proc/net/rtl8192du/wlan0/best_channel
```

Notes:

1. If your WLAN interface is not wlan0, please change it to your used interface. (ex: wlan51)
2. If your driver IC is not RTL8192DU-VS, please change the rtl8192du to your used IC. (e.g. rtl8192cu, rtl8188eu, etc.)
3. The result of “cat /proc/net/rtl.../wlan0/best_channel” has two parts. The first part is received packet count at all supported channels during site survey. Developers can use these statistics to decide your own best channel. The second part is the simple estimated best channels for 5G and 2.4G band for reference only. HT40+ rule is used to estimate the best channel when the primary channel is at 1, 6, 36, 44, etc., and the primary channel with minimum received packet count is selected.

(D) How to set Hidden SSID?

```
# Send empty SSID in beacons and ignore probe request frames that do not  
# specify full SSID, i.e., require stations to know SSID.  
# default: disabled (0)  
# 1 = send empty (length=0) SSID in beacon and ignore probe request for  
# broadcast SSID  
# 2 = clear SSID (ASCII 0), but keep the original length (this may be required  
# with some clients that do not support empty SSID) and ignore probe  
# requests for broadcast SSID  
ignore_broadcast_ssid=0
```

(E) How to set MAC address ACL

```
# Station MAC address -based authentication  
# Please note that this kind of access control requires a driver that uses  
# hostapd to take care of management frame processing and as such, this can be  
# used with driver=hostap or driver=nl80211, but not with driver=madwifi.
```

```
# 0 = accept unless in deny list, deny_mac_file is used to specify deny list.  
# 1 = deny unless in accept list, accept_mac_file is used to specify deny list.  
#macaddr_acl=1
```

```
# Accept/deny lists are read from separate files (containing list of  
# MAC addresses, one per line). Use absolute path name to make sure that the  
# files can be read on SIGHUP configuration reloads.  
accept_mac_file=/etc/hostapd.accept  
deny_mac_file=/etc/hostapd.deny
```

Realtek