# Quick Start Guide for starting Soft-AP mode

| Version | Date | Change list | Notes |
|---|---|---|---|
| V0.7 | 2022/04/21 | Add description about softAP cannot be started at 5G band when NO_IR status is not cleared. | |
| V0.6 | 2021/10/12 | Add AX description and hostapd.conf examples in appendix for reference. | |
| V0.5 | 2021/06/02 | Update ACS information | |
| V0.4 | 2021/06/01 | Add command "chan_switch" usage | |
| V0.3 | 2020/11/24 | Add VHT configuration of hostapd | |
| V0.2 | 2019/09/18 | Add Realtek proprietary ACS (Automatic Channel Selection) | |
| V0.1 | | Initial version | |

(A) How to start Soft-AP mode:

(1) Disable network management or other wireless tools, e.g. wpa_supplicant

**P.S.** If now is in concurrent mode. The wlan0 interface used by STA mode still needs its wpa_supplicant. For more information, please refer the document “Realtek_WiFi_concurrent_mode_Introduction.doc”

(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 >= 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**

I. The default channel plan setting is worldwide (WW) and that means 5G band will be all NO_IR. With this setting, SoftAP mode cannot be started at 5G band until the NO_IR status is cleared. For more detail, please check with module or IC vendor.

(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=nl80211
```

(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) IEEE 802.11ac related configuration

```
# ieee80211ac: Whether IEEE 802.11ac (VHT) is enabled
# 0 = disabled (default)
# 1 = enabled
# Note: You will also need to enable WMM for full VHT functionality.
# Note: hw_mode=a is used to specify that 5 GHz band is used with
VHT.
ieee80211ac=1

# 0 = 20 or 40 MHz operating Channel width
# 1 = 80 MHz channel width
# 2 = 160 MHz channel width
# 3 = 80+80 MHz channel width
vht_oper_chwidth=1

# center freq = 5 GHz + (5 * index)
# So index 42 gives center freq 5.210 GHz
# which is channel 42 in 5G band
# You don’t need to set this if you use 20/40MHz
vht_oper_centr_freq_seg0_idx=42
```

For example:

● VHT20

```
ieee80211ac=1
vht_oper_chwidth=0
```

● VHT40

```
ieee80211ac=1
ht_capab=[HT40+]
vht_oper_chwidth=0
```

● VHT80

```
ieee80211ac=1
ht_capab=[HT40+]
vht_oper_chwidth=1
vht_oper_centr_freq_seg0_idx=42
```

(v) IEEE 802.11ax related configuration

```
#ieee80211ax: Whether IEEE 802.11ax (HE) is enabled
# 0 = disabled (default)
# 1 = enabled
```

**ieee80211ax**=1

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

**./hostapd_cli all_sta**

(vii) 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) ACS (Automatic Channel Selection) in pure linux, you can choose **one** of hostapd ACS or realtek proprietary ACS.

1. How to use hostapd ACS (Driver version >= 5.11)
   a) Add compiler flag in your platform setting and rebuild the driver.
      EXTRA_CFLAGS += -D**CONFIG_RTW_HOSTAPD_ACS**
   b) Enable ACS in hostapd build config file.
      **CONFIG_ACS**=y
   c) The major variable setting in the rtl_hostapd.conf configure file
      (i) hw mode configuration
          **hw_mode** = b or g or a
      (ii) channel configuration
          # If CONFIG_ACS build option is enabled, the channel can be selected automatically at run time by setting channel=acs_survey or channel=0, both of which will enable the ACS survey based algorithm.
          **channel**=0 or **channel**=acs_survey

2. Realtek driver has ability to switch to low interference channel (best channel) automatically. It relies on third party software to trigger Realtek proprietary ACS via proc filesystem.
   How to use Realtek proprietary ACS (Driver version >= 5.7)
   a) Add compiler flag in your platform setting and rebuild the driver
      EXTRA_CFLAGS += -D**CONFIG_RTW_ACS**
   b) Trigger ACS via proc filesystem
      When softAP was started, echo acs > /proc/net/rtl…./wlan0/survey_info, to trigger driver processing ACS mechanism to scan all supported channels.

After scan, driver switches to the best channel automatically.

The usage of survey_info

(i) Driver processes ACS scan and then switch to the best channel.

```
# echo acs > /proc/net/rtl…./wlan0/survey_info
```

(ii) Driver processes normal scan.

```
# echo 1 > /proc/net/rtl…./wlan0/survey_info
```

(iii) Driver dumps all AP info from scan queue.

```
# cat /proc/net/rtl…./wlan0/survey_info
```

### (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
```

### (F) How to initiate channel switch announcement

* **chan_switch** <cs_count> <freq> [sec_channel_offset=] [center_freq1=]

[center_freq2=] [bandwidth=] [blocktx] [ht | vht]
* sec_channel_offset - Secondary channel offset for HT40
    * 0 = HT40 disabled,
    * -1 = HT40 enabled, secondary channel below primary,
    * 1 = HT40 enabled, secondary channel above primary
* center_freq1 - Segment 0 center frequency in MHz, valid for both HT and VHT.
* bandwidth - Channel bandwidth in MHz (20, 40, 80, 160)

For example

**VHT 80M** (need ieee80211n=1 && ieee80211ac=1)

```
# ./hostapd_cli chan_switch 5 5200 sec_channel_offset=-1 bandwidth=80
center_freq1=5210 vht
```

**HT 40M** (need ieee80211n=1)

```
# ./hostapd_cli chan_switch 5 5180 sec_channel_offset=1 bandwidth=40 ht
```

**20M** depends on hostapd HT/VHT capability

```
# ./hostapd_cli chan_switch 5 5180
```

Note that:

a. You can’t chan_switch to VHT capability if hostapd.conf set ieee80211ac=0.
b. You can’t chan_switch to bandwidth=40 if hostapd.conf doesn’t set ht_capab= [HT40-/+].
c. P2P-GO can’t chan_switch to DFS channel even if you enable DFS-master. (with RTK wpa_supplicant_8_O_8.x)

(G) How to delete hostapd

```
# rm /var/run/hostapd/wlan0
```

# Appendix:

- N mode AP at ch6@BW20 with open security

Configuration:

```
interface=wlan0
ssid=ht_ch6_bw20
hw_mode=g
channel=6
ieee80211n=1

driver=nl80211
beacon_int=100
```

Result:

```
$ cat /proc/net/rtl8852be/wlan0

AP Vendor Realtek
SSID=ht_ch6_bw20
sta's macaddr:00:e0:4c:0b:b1:d5
cur_channel=6, cur_bwmode=0(20MHz), cur_ch_offset=0
wireless_mode=0xd(B/G/N), rtsen=0, cts2slef=0 hw_rts_en=0
state=0x11, aid=0, macid=0
qos_en=1, ht_en=1, init_rate=0
bwmode=0, ch_offset=0, sgi_20m=1,sgi_40m=1
```

- N mode AP at ch6@BW40 with open security

Configuration:

```
interface=wlan0
ssid=ht_ch6_bw40
hw_mode=g
channel=6
ieee80211n=1
ht_capab=[HT40-]
```

```
driver=nl80211
beacon_int=100
```

Result:

```
$ cat /proc/net/rtl8852be/wlan0

AP Vendor Realtek
SSID=ht_ch6_bw40
sta's macaddr:00:e0:4c:0b:b1:d5
cur_channel=6, cur_bwmode=1(40MHz), cur_ch_offset=0
wireless_mode=0xd(B/G/N), rtsen=0, cts2slef=0 hw_rts_en=0
state=0x11, aid=0, macid=0
qos_en=1, ht_en=1, init_rate=0
bwmode=0, ch_offset=0, sgi_20m=1,sgi_40m=1
```

Notes:

STA may not work at BW40 in your environment (still at 20MHz bandwidth). It is due to hostapd found overlap BSS nearby. You will see following message in hostapd log.

```
Found overlapping 20 MHz HT BSS: 2c:4d:54:6f:b7:60 freq=2412
Overlapping 20 MHz BSS is found
20/40 MHz operation not permitted on channel pri=6 sec=2 based on
overlapping BSSes
Fallback to 20 MHz

Check list of neighboring BSSes (from scan) to see whether 40 MHz is
allowed per IEEE Std 802.11-2012, 10.15.3.2.
```

- AC mode AP at ch36@BW20 with open security

Configuration:

```
interface=wlan0
ssid=ac_ch36_bw20
hw_mode=a
channel=36
```

```
ieee80211n=1
ieee80211ac=1

driver=nl80211
beacon_int=100
```

Result:

```
$ cat /proc/net/rtl8852be/wlan0/ap_info

AP Vendor Realtek
SSID=ac_ch36_bw20
sta's macaddr:00:e0:4c:0b:b1:d5
cur_channel=36, cur_bwmode=0(20MHz), cur_ch_offset=0
wireless_mode=0x12(A/AC), rtsen=0, cts2slef=0 hw_rts_en=0
state=0x11, aid=0, macid=0
qos_en=1, ht_en=1, init_rate=0
bwmode=0, ch_offset=0, sgi_20m=1,sgi_40m=1
```

- AC mode AP at ch36@BW40 with open security

  Configuration:

```
interface=wlan0
ssid=ac_ch36_bw40
hw_mode=a
channel=36
ieee80211n=1
ht_capab=[HT40-][HT40+]

ieee80211ac=1

driver=nl80211
beacon_int=100
```

Result:

```
$ cat /proc/net/rtl8852be/wlan0/ap_info

AP Vendor Realtek
```

```
SSID=ac_ch36_bw40
sta's macaddr:00:e0:4c:0b:b1:d5
cur_channel=36, cur_bwmode=1(40MHz), cur_ch_offset=1
wireless_mode=0x12(A/AC), rtsen=0, cts2slef=0 hw_rts_en=0
state=0x11, aid=0, macid=0
qos_en=1, ht_en=1, init_rate=0
bwmode=1, ch_offset=1, sgi_20m=1,sgi_40m=1
```

- AC mode AP at ch36@BW80 with open security

Configuration:

```
interface=wlan0
ssid=ac_ch36_bw80
hw_mode=a
channel=36
ieee80211n=1
ht_capab=[HT40-][HT40+]

ieee80211ac=1
vht_oper_chwidth=1
vht_oper_centr_freq_seg0_idx=42

driver=nl80211
beacon_int=100
```

Result:

```
$ cat /proc/net/rtl8852be/wlan0/ap_info

AP Vendor Realtek
SSID=ac_ch36_bw80
sta's macaddr:00:e0:4c:0b:b1:d5
cur_channel=36, cur_bwmode=2(80MHz), cur_ch_offset=1
wireless_mode=0x12(A/AC), rtsen=0, cts2slef=0 hw_rts_en=0
state=0x11, aid=0, macid=0
qos_en=1, ht_en=1, init_rate=0
bwmode=2, ch_offset=1, sgi_20m=1,sgi_40m=1
```

- AX mode AP at ch36@BW80 with open security

Configuration:

```
interface=wlan0
ssid=ax_ch36_bw80
hw_mode=a
channel=36
ieee80211n=1
ht_capab=[HT40-][HT40+]

ieee80211ac=1
vht_oper_chwidth=1
vht_oper_centr_freq_seg0_idx=42

ieee80211ax=1

driver=nl80211
beacon_int=100
```

Result:

```
$ cat /proc/net/rtl8852be/wlan0/ap_info

AP Vendor Realtek
SSID=ax_ch36_bw80
sta's macaddr:00:e0:4c:0b:b1:d5
cur_channel=36, cur_bwmode=2(80MHz), cur_ch_offset=1
wireless_mode=0x22(A/AX), rtsen=0, cts2slef=0 hw_rts_en=0
state=0x11, aid=0, macid=0
qos_en=1, ht_en=1, init_rate=0
bwmode=2, ch_offset=1, sgi_20m=1,sgi_40m=1
```

- AX mode AP at ch36@BW80 with WPA2 security

Configuration:

```
interface=wlan0
ssid=ax_ch36_bw80
```

```
hw_mode=a
channel=36
ieee80211n=1
ht_capab=[HT40-][HT40+]

ieee80211ac=1
vht_oper_chwidth=1
vht_oper_centr_freq_seg0_idx=42

ieee80211ax=1

wpa=2
wpa_passphrase=12345678
wpa_key_mgmt=WPA-PSK
wpa_pairwise=CCMP

driver=nl80211
beacon_int=100
```

Result:

```
$ cat /proc/net/rtl8852be/wlan0/ap_info

AP Vendor Realtek
SSID=ax_ch36_bw80
sta's macaddr:00:e0:4c:0b:b1:d5
cur_channel=36, cur_bwmode=2(80MHz), cur_ch_offset=1
wireless_mode=0x22(A/AX), rtsen=0, cts2slef=0 hw_rts_en=0
state=0x11, aid=0, macid=0
qos_en=1, ht_en=1, init_rate=0
bwmode=2, ch_offset=1, sgi_20m=1,sgi_40m=1
```