



AP-Router SDK User Guide

Realtek AP-Router SDK User Guide

Date: 2015/05/18

Version: 1.111

This document is subject to change without notice. The document contains Realtek confidential information and must not be disclosed to any third party without appropriate NDA.

Change History

Version	Date	Remarks
1.0	2010/04/06	Initial Release
1.1	2010/07/28	1) Change Jungle to kernel 2.6; 2) Add compile environment.
1.2	2010/08/03	Add Section 2.4 and add Chapter 4~6.
1.3	2010/08/20	1) Modify section 4.3; 2) Add section 4.4; 2) Add section 4.8~4.11; 3) Add chap 6; 4) Add section 7.3~7.5
1.4	2010/08/27	1) Modify section 4.11; 2) Add section 7.6~7.7
1.5	2010/09/21	Add section 4.12 and 6.16
1.6	2010/09/30	1) Modify section 6.8; 2) Add section 7.8~7.9
1.7	2010/10/22	Add section 4.12.4 and 4.13~4.14
1.8	2010/10/27	Add section 4.15~4.16
1.9	2010/11/05	Add section 4.17
1.10	2010/11/12	Add section 4.18 and 7.10
1.11	2010/11/16	Add section 4.19~4.22
1.12	2010/12/01	1) Modify section 4.18 for 92D; 2) Add section 4.23
1.13	2010/12/24	Add section 4.24~4.25 and 7.11
1.14	2011/01/06	Modify section 4.16 to add Bootloader configure for 8M flash.
1.15	2011/01/27	Add section 7.12
1.16	2011/01/28	Add section 4.26
1.17	2011/02/16	Add section 7.13~7.15
1.18	2011/03/07	Add section 4.5.2
1.19	2011/03/18	1) Add RTL8954C and add section 4.27 rtk_voip support. 2) Add step 3, 0) at section 2.2 3) Add section 4.28, 7.16~7.22
1.20	2011/03/24	Add section 4.29
1.21	2011/04/07	Add section 4.3.1.5, 4.30, 4.31, 7.23~7.24
1.22	2011/04/08	1) Modify section 3.3

		2) Modify 7.12 for force mode 3) Add section 7.25
1.23	2011/04/22	Modify section 4.18 92D already support WDS.
1.24	2011/04/27	1) Modify section 3.1 2) Modify section 3.3 3) Add section 3.4 4) Add section 3.5
1.25	2011/04/29	Add section 7.26
1.26	2011/05/05	Add section 4.32 and 7.27
1.27	2011/05/31	1) Modify 2.3 and add 4.22.4 to add iNIC. 2) Modify section 6.6
1.28	2011/05/31	Modify section 4.6 (not support Mesh)
1.29	2011/06/03	Add section 7.28
1.30	2011/06/09	Add section 4.33
1.31	2011/06/10	Add section 4.15.5
1.32	2011/06/10	Add section 4.34
1.33	2011/06/15	Add section 4.35
1.34	2011/06/20	Add section 7.29
1.35	2011/06/22	Add section 4.4.6
1.36	2011/06/24	Add section 4.36
1.37	2011/06/27	Add section 4.3.5 Add section 7.30
1.38	2011/06/29	Modify section 7.20
1.39	2011/7/7	Add section 4.3.6
1.40	2011/07/12	Add section 4.37
1.41	2011/08/01	1) Add section 4.14.2 2) Add section 4.38
1.42	2011/08/18	Add section 4.15.5 (2) (test result of 96C+92D)
1.43	2011/08/19	Modify section 4.14.3 (II) (ENABLE/DISABLE HW_QoS)
1.44	2011/08/26	1) Add section 4.39 2) Modify section 2.4.3, section 5, section 7.16~7.18 and section 7.28 for Boa

1.45	2011/09/05	Add section 4.40
1.46	2011/09/15	1) Add section 4.41 2) Modify section 4.7
1.47	2011/09/19	Add section 4.31.3
1.48	2011/09/30	Add RTL8196CT, RTL8198T and RTL819xD support
1.49	2011/10/11	Edit section 4.3.2, and add section 4.3.3
1.50	2011/10/25	Modify Section 4.31
1.51	2011/11/03	Add section 7.31 Add section 6.17~6.26
1.52	2011/11/07	Add section 4.42 Modify section 7.1
1.53	2011/11/09	Modify section 4.3
1.54	2011/11/10	Add section 7.32
1.55	2011/12/14	Add emch for ipv6 multicast forward support on section 4.15
1.56	2011/12/26	Modify section 5.2
1.57	2012/01/10	Add section 4.44, 4.45 and 7.33
1.58	2012/01/11	Modify section 2.3
1.59	2012/01/11	Modify section 4.7.2 Add section 7.34
1.60	2012/01/31	Add section 6.27
1.61	2012/02/20	Modify section 2.3
1.62	2012/03/06	Modify section 4.1.17
1.63	2012/03/21	Modify section 4.19.2
1.64	2012/04/24	Add section 6.28
1.65	2012/05/10	Add section 4.2.3
1.66	2012/05/17	Add BOA's ipv6 support. Add PPP's ipv6 support. Modify section 4.15.2
1.67	2012/05/22	Add ipv6 QoS support
1.68	2012/05/23	Add section 6.29
1.69	2012/06/13	Add section 7.35
1.70	2012/07/23	Modify section 4.15
1.71	2012/09/19	Add section 7.36
1.72	2012/10/02	Modify section 7.1
1.73	2012/10/03	Add section 3.6

1.74	2012/11/19	Modify section 4.12.3
1.75	2012/12/19	Add black list and white list to url filter
1.76	2013/1/11	Add 11AC special settings 7.37
1.77	2013/01/29	Add jumbo frame support and add sections of 4.48 and 6.30
1.78	2013/02/28	Modify section 2.2 and 2.3
1.79	2013/03/13	Add section 6.31
1.80	2013/03/15	Add section 7.38
1.81	2013/05/10	Modify section 2.3
1.82	2013/05/15	Add “8881A Selectable” settings in Section 2.3
1.83	2013/06/28	Add section 4.49 “11AC Logo Test by Sigma Support”
1.84	2013/07/10	Add “linux 802.1q vlan” in section 4.7.4
1.85	2013/09/27	Add section 7.39
1.86	2013/10/04	Add HS2 support in Section 4.50 and Modify Section 4.49 for 11AC/HS2 Logo Test by Sigma Support.
1.87	2013/10/14	Modify section 4.43
1.88	2013/10/17	Add multiple repeater and multiple mac clone in Section 4.51
1.89	2013/10/22	Add support Russian I2tp special features in Section 4.52
1.90	2013/10/30	Modify section 4.43
1.91	2013/11/12	Modify section 4.6 “Mesh Support”
1.92	2013/11/25	Modify section 4.43
1.93	2014/03/13	Add Realtek Simple Config Support
1.94	2014/03/28	Add 2.1 compile environment note
1.95	2014/05/05	Add SATA Interface Support.
1.96	2014/05/06	Add 2.4G-wlan0 5G-wlan1 Support
1.97	2014/05/06	Add Bluetooth and guest support
1.98	2014/05/08	Add 4.58 GIGA IOL Test
1.99	2014/06/11	Add 4.61 Realtek Giga Lite (Two-Pair 500Mbps) support
1.100	2014/06/13	Modify 4.21 Hostapd step2 how to make menuconfig and step3 ① test

		WPA2-psk
1.101	2014/08/04	Add support for both kernel 2.6 and kernel 3.10(section 1-4.20)
1.102	2014/08/19	Add 4.62 https boa
1.103	2014/08/19	Add 4.63/4.64 exfat file system
1.104	2014/09/24	Add section 7.44
1.105	2014/09/25	Add section 6.2 4)
1.106	2014/10/09	Add section 4.12.5 hardware nat init flow
1.107	2015/03/13	Add section 4.67
1.108	2015/03/31	Add Hardware Crypto Engine API support special features in Section 4.68
1.109	2015/04/28	Add WAC support Add HomeKit support
1.110	2015/04/30	Add FAQ samba disk 20M issue
1.111	2015/05/06	Refine to merge samba and usb support
1.112	2015/05/18	Refine the users_menuconfig
1.113	2015/06/09	Add rtl_fs support section 4.71 Add STA Control support section 4.72
1.114	2015/06/10	Add 802.11r/k support in section 4.73

AP-Router SDK User Guide	1
1. AP-Router SDK introduction	1
2. Compile AP-Router SDK	2
2.1. <i>Compile Environment.....</i>	2
2.2. <i>Compile bootloader.....</i>	2
2.3. <i>Compile rlxlinux.....</i>	4
2.4. <i>Compile rlxlinux detailed</i>	17
3. AP-Router SDK images upload	19
3.1 <i>Topology to upload image</i>	19
3.2 <i>Upload bootloader image</i>	19
3.3 <i>Upload rlxlinux image.....</i>	20
3.4 <i>Upload default setting data (Optional)</i>	22
3.5 <i>Upload rlxlinux image via webpage (Optional)</i>	22
3.6 <i>Upload rlxlinux image with config file via webpage (Optional).....</i>	23
4. Features configure	25
4.1 <i>PCI support</i>	25
4.2 <i>USB support.....</i>	26
4.3 <i>Samba support.....</i>	28
4.4 <i>USB 3G support.....</i>	36
4.5 <i>WAPI support.....</i>	42
4.6 <i>Mesh support</i>	44
4.7 <i>VLAN support.....</i>	53
4.8 <i>I2C UART support.....</i>	68
4.9 <i>DLNA support.....</i>	71
4.10 <i>Pocket AP support.....</i>	75
4.11 <i>Wireless client mode 802.1x support.....</i>	76
4.12 <i>Hardware NAT.....</i>	77
4.13 <i>Iptables rule to ACL rule translation support.....</i>	81
4.14 <i>Hardware QoS support</i>	82
4.15 <i>IPv6 support.....</i>	86
4.16 <i>64K/sector SPI flash support</i>	94
4.17 <i>SPI Flash Support.....</i>	96
4.18 <i>92C/D support</i>	99
4.19 <i>GDB server support</i>	105
4.20 <i>HTTP file server support</i>	106
4.21 <i>Hostapd support.....</i>	114

4.22	<i>MP support</i>	118
4.23	<i>AP mode support</i>	121
4.24	<i>Wireless configuration file support</i>	122
4.25	<i>WPS under wireless configuration file support</i>	123
4.26	<i>Domain name query support</i>	124
4.27	<i>rtk_voip support</i>	129
4.28	<i>Realtek Flash Dual Image support</i>	129
4.29	<i>IEEE 802.3az EEE (Energy Efficient Ethernet) support</i>	131
4.30	<i>IGMP/MLD support</i>	131
4.31	<i>SNMP support</i>	133
4.32	<i>UVC support</i>	134
4.33	<i>Telnetd and Login support in busybox</i>	136
4.34	<i>Dynamic Frequency Selection (DFS) support</i>	138
4.35	<i>Fastpath Filter</i>	141
4.36	<i>Fastpath SPI (stateful packet inspection) support</i>	145
4.37	<i>ALSA support</i>	145
4.38	<i>Romeperf support</i>	147
4.39	<i>Boa support</i>	150
4.40	<i>uWiFi support</i>	157
4.41	<i>ALL NAT</i>	163
4.42	<i>Proc filesystem for debug</i>	165
4.43	<i>TR069 Support</i>	168
4.44	<i>ALG Support</i>	176
4.45	<i>Cone Nat support</i>	177
4.46	<i>Miniigd IPv6 support</i>	178
4.47	<i>WPS supported for Virtual Client in repeater mode</i>	179
4.48	<i>Jumbo frame support</i>	179
4.49	<i>11AC/HS2R1 Logo Test by Sigma Support</i>	180
4.50	<i>Hotspot 2.0 Support</i>	182
4.51	<i>Hotspot 2.0 Release 1 Logo Test</i>	185
4.52	<i>Multiple repeater mode and multiple mac clone</i>	187
4.53	<i>Support Russian L2TP special features</i>	191
4.54	<i>WIFI-DIRECT (WIFI-P2P) support</i>	192
4.55	<i>Realtek Simple Config support</i>	197
4.56	<i>Realtek Bluetooth support</i>	198
4.57	<i>Realtek bridge guest zone support(for kernel 3.10)</i>	203
4.58	<i>2.4G -wlan0/5G-wlan1 Support(for kernel 3.10)</i>	205
4.59	<i>GIGA IOL Test</i>	206

4.60	<i>NAND FLASH support</i>	207
4.61	<i>IP-IN-IP SUPPORT</i>	209
4.62	<i>Realtek Giga Lite (Two-Pair 500Mbps) support (for kernel 3.10)</i>	210
4.63	<i>Https boa support</i>	211
4.64	<i>exfat-fuse file system</i>	214
4.65	<i>exfat-nofuse file system</i>	215
4.66	<i>UART1 support</i>	216
4.67	<i>System Memory size > 256MB support (98C only)</i>	216
4.68	<i>Hardware Crypto Engine API support</i>	217
4.69	<i>Apple WAC support</i>	219
4.70	<i>Apple HomeKit support</i>	221
4.71	<i>rtl_fs support</i>	223
4.72	<i>STA Control support</i>	225
4.73	<i>802.11r/k support</i>	227
5	FLASH tools	230
5.1	<i>flash</i>	230
5.2	<i>cvcfg-gw</i>	230
5.3	<i>compweb</i>	231
5.4	<i>cvimg</i>	231
5.5	<i>mgbin</i>	231
6	Proc file format	233
6.1	<i>br_igmpProxy</i>	233
6.2	<i>br_igmpsnoop</i>	233
6.3	<i>custom_Passthru</i>	234
6.4	<i>enable_dos</i>	235
6.5	<i>fast_l2tp</i>	236
6.6	<i>fast_nat</i>	236
6.7	<i>fast_pptp</i>	237
6.8	<i>filter_table</i>	237
6.9	<i>gpio</i>	238
6.10	<i>pptp_conn_ck</i>	239
6.11	<i>qos</i>	239
6.12	<i>rf_switch</i>	239
6.13	<i>rtk_vlan_support</i>	240
6.14	<i>mib_vlan</i>	240
6.15	<i>sw_nat</i>	241
6.16	<i>hw_nat</i>	241
6.17	<i>br_wlanblock</i>	242

6.18	<i>br_igmpVersion</i>	242
6.19	<i>br_igmpquery</i>	243
6.20	<i>br_mCastFastFwd</i>	243
6.21	<i>br_mldquery</i>	244
6.22	<i>br_mldsnoop</i>	244
6.23	<i>eee</i>	244
6.24	<i>gc_overflow_timeout</i>	245
6.25	<i>load_default</i>	245
6.26	<i>reInitSwitchCore</i>	245
6.27	<i>br_mldVersion</i>	246
6.28	<i>jate</i>	246
6.29	<i>gpio_ctrl</i>	247
6.30	<i>Jumbo frame support</i>	248
6.31	<i>port_status</i>	249
7	FAQ	251
7.1	<i>How to modify the mappings of LAN/WAN port?</i>	251
7.2	<i>How to change the baud rate of console?</i>	251
7.3	<i>How to use vlan priority option of iptables?</i>	251
7.4	<i>Relationship of virtual AP and root AP?</i>	252
7.5	<i>How to customize icon of lld2d?</i>	252
7.6	<i>How to rename the AP when shown on Windows?</i>	252
7.7	<i>How to add MAC address filter based on ether driver API?</i>	253
7.8	<i>How to adjust CPU speed?</i>	255
7.9	<i>How to shutdown power of Ethernet ports?</i>	256
7.10	<i>How to disable Ethernet PHY?</i>	257
7.11	<i>Which channels are supported by RTL8192D?</i>	257
7.12	<i>How to set speed and duplex of physical port via APIs?</i>	259
7.13	<i>How to dial up pppoe for test when wan interface is changed?</i>	259
7.14	<i>How to enable/disable ipv6 passthru?</i>	263
7.15	<i>Please explain “Support multi-vlan in bridge/wisp mode” at kernel?</i>	264
7.16	<i>How to add/update flash MIB entry at the web-server?</i>	264
7.17	<i>How to modify flash MIB settings using the configure file uploaded via webpage?</i>	266
7.18	<i>How to modify webpages at the web-server?</i>	267
7.19	<i>How to build image which can run in ICE?</i>	268
7.20	<i>The image size and free memory info for RTL8198 + RTL8192C SDK.</i>	269
7.21	<i>The summary of the HW feature for ICs till now.</i>	269
7.22	<i>Default values of RTL8196C SDK multicast and IPv6.</i>	270
7.23	<i>How to configure to process IGMP reserve address?</i>	270

7.24	<i>How to modify flash offset of webpages/rootfs/kernel etc via linux menuconfig?</i>	271
7.25	<i>For QoS using htb, something needs to be noticed when set tc rules.</i>	272
7.26	<i>How to configure RTL8192D internal PA?</i>	273
7.27	<i>How to configure RTL8192C External PA?</i>	273
7.28	<i>How to build image with firmware and config data (without HW setting)?</i>	274
7.29	<i>What is the firewall capability and the relation with hardware NAT?</i>	276
7.30	<i>How to enable/disable PPPoE passthru?</i>	276
7.31	<i>How to control dropping unknown multicast packet or not</i>	278
7.32	<i>How to use AP mib in another application</i>	278
7.33	<i>How to Support rtl8211ds for RTL8197D High Power demo board</i>	279
7.34	<i>How to adjust the DRAM size</i>	280
7.35	<i>How to add applications</i>	281
7.36	<i>How to Enable JFFS2 in the linux and user space for storage partition</i>	283
7.37	<i>What are the special settings for 11AC?</i>	284
7.38	<i>How to use Band Edge Limit Tables</i>	285
7.39	<i>RTL8367RB LED setting</i>	290
7.40	<i>How to enable GPIOB5 as Reset Button for some RTL8196EU-MCM reference design</i>	290
7.41	<i>How to enable SATA interface support on the RTL8198CS platform</i>	291
7.42	<i>How to use an gpio pin</i>	291
7.43	<i>How to build image without default config</i>	291
7.44	<i>How to support LDPC function for 8192ER</i>	293
7.45	<i>How to configure 8814AR/8194AR to use 2T2R setting</i>	293
7.46	<i>How to configure hw qos by rtk_cmd</i>	294
7.47	<i>How to fix samba “map network driver” disk size show 20M issue</i>	298

1. AP-Router SDK introduction

AP-Router SDK is a Linux development platform customized to provide a unified platform for embedded systems based on RLX processor cores.

AP-Router SDK's directory lists as follows:

RLXLinux

- bootcode rtl8196c_98 or bootcode rtl819xD
- rtl819x-sdk-v2.x/rtl819x or rtl819x-sdk-v3.x/rtl819x

Directory bootcode rtl8196c_98 deposit bootloader source code of RTL8196C, RTL8196CT, RTL8198, RTL8198T for AP-Router SDK while rtl819x-sdk-v2.x/rtl819x stores applications and linux kernel.

Directory bootcode rtl819xD deposit bootloader source code of RTL8196E, RTL8196D, RTL8197D for AP-Router SDK while rtl819x-sdk-v2.x/rtl819x stores applications and linux kernel.

Directory bootcode rtl8198c deposit bootloader source code of RTL8198C for AP-Router SDK while rtl819x-sdk-v3.x/rtl819x stores applications and linux kernel.

Note: RTL8198 and RTL8954C are SoC sharing the same SDK. The Major difference between RTL8198 and RTL8954C are BSP and VoIP capabilities. For more VoIP details please refer to VoIP SDK Manual.

RTL8198C is only supported by Kernel 3.10 now.

2. Compile AP-Router SDK

We assume that AP-Router SDK is stored at root directory ('/'), the source code directory as "rtl819x".

2.1. Compile Environment

Red Hat Enterprise [Recommend]

Fedora 9

Ubuntu 8.10/9.10

Note:

- 1) x64 environment need to install ia32-libs first to support compile tools.

On Red Hat/Fedora , run *yum install -y glibc-2.14.90-24.fc16.6.i686*

On ubuntu, run *sudo apt-get install ia32-libs*

- 2) Please use bash as default shell.

On some new linux distributions such as Ubuntu12.04, it may use dash as default shell(!! /bin/sh will see /bin/sh->dash). For Ubuntu, try *sudo dpkg-reconfigure dash* and select no.

2.2. Compile bootloader

Steps to compile bootloader are as follows:

Step 1, enter bootloader directory.

For RTL8196C, RTL8198, RTL8196CT, RTL8198T, RTL8954C: *cd bootcode rtl8196c_98*

For RTL8196D, RTL8197D: *cd bootcode rtl819xD*

For RTL8197D+8367RB: *cd bootcode rtl8197d_8367r*

For RTL8881A: *cd bootcode rtl8881A*

For RTL8198C: *cd bootcode rtl8198c*

Step 2, choose default configure file.

For 8196C and RTL8196CT: *cp def-rtl8196c-config .config*

For 8198 and RTL8198T: *cp def-rtl8198-config .config*

For 819xD: *cp def-rtl8196d-config .config*

For 8954C v200: *cp def-rtl89xxc-v200-config .config*

For 8954C v4xx:

cp def-rtl89xxc-v4xx-config .config

For 8198C

cp def-rtl8198c-8954e-config .config

Step 3, appoint the compile toolchain and make default configure file chosen effective.

0) Appoint the compile toolchain.

One method to appoint the compile toolchain as follows:

- find the toolchain at .../rtl819x/toolchain.

rsdk-1.3.6-4181-EB-2.6.30-0.9.30 is for RTL8196C, RTL8196CT, RTL8196E and RTL8196EU

rsdk-1.3.6-5281-EB-2.6.30-0.9.30 is for RTL8198, RTL8198T and RTL8954C.

rsdk-1.5.5-5281-EB-2.6.30-0.9.30.3-110714 is for RTL8196D, RTL8197D and RTL8881A.

- for rtl8198c ,please uncompress the toolchain

msdk-4.4.7-mips-EB-3.10-0.9.33-m32t-131227b.tgz

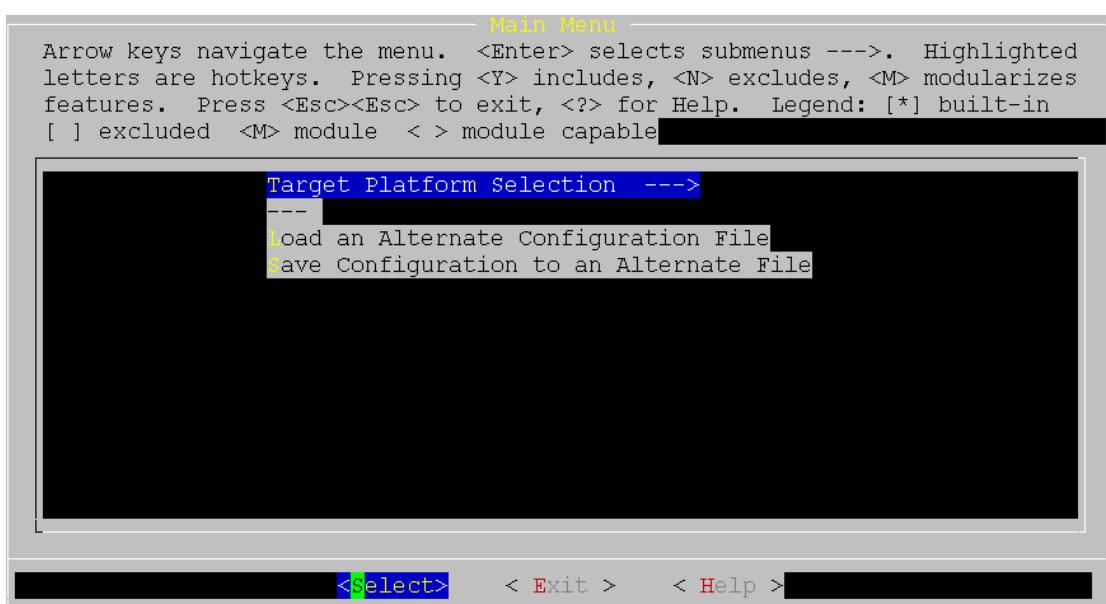
- add the exact toolchain to the environment variant PATH.

Here we take the toolchain for RTL8196C as an example.

Input command as follows:

PATH=\$PATH: .../rtl819x/toolchain/rsdk-1.3.6-4181-EB-2.6.30-0.9.30/bin

i) *make menuconfig* // To configure bootloader settings



ii) Choose '<Exit>' and click 'Enter'

Note:

Above we just use default configure file for bootloader settings.

If we want to configure bootloader different from default configure file,

select ‘Target Platform Selection’ and click ‘Enter’ to customize settings of bootloader.



iii) Choose ‘<Yes>’ and click ‘Enter’

Step 4, compile bootloader.

make

2.3. Compile rlxlinux

Steps to compile applications and linux are as follows:

Step 1, enter rtl819x directory.

Step 2, configure rlxlinux.

i) *make menuconfig* // To configure rlxlinux settings

ii) Do some settings as follows:

‘Selected Target’ to choose rtl8196c, rtl8196ct, rtl8198,
rtl8198t, rtl8954C, rtl819xD and rtl8198c supported at present;

‘Selected Kernel’ to choose linux kernel;

‘Selected Busybox’ to choose busybox;

‘Selected toolchain’ to choose toolchain to compile rlxlinux;

‘Selected Board Configuration’ to choose flash type (NOR or SPI)

and squash file system.

Example I, settings for RTL8196C demo board as follows:

```

RLX Linux Configuration
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are
hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc>
to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ] excluded <M> module < >
module capable

--- select components
  Selected Target (rtl8196c) --->
  Selected Kernel (linux-2.6.30) --->
  Selected Busybox (busybox-1.13) --->
  Selected toolchain (rsdk-1.3.6-4181-EB-2.6.30-0.9.30) --->
--- rtl8196c
  Selected Target of SDK (l1nRouter_GW) --->
  Selected Board Configuration (SPI flash + Squashfs) --->
  IC Test Configuration --->
--- config components
[ ] Config kernel (NEW)
[ ] Config users (NEW)
[ ] Config busybox (NEW)
[*] Load default settings
[ ] Save default settings
---
  Load an Alternate Configuration File
  Save an Alternate Configuration File

<Select>  < Exit >  < Help >

```

Example II, settings for RTL8198 demo board as follows:

```

RLX Linux Configuration
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are
hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc>
to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ] excluded <M> module < >
module capable

--- select components
  Selected Target (rtl8198) --->
  Selected Kernel (linux-2.6.30) --->
  Selected Busybox (busybox-1.13) --->
  Selected toolchain (rsdk-1.3.6-5281-EB-2.6.30-0.9.30) --->
--- rtl8198
  Selected Board Configuration (SPI flash + Squashfs) --->
  IC Test Configuration --->
--- config components
[ ] Config kernel (NEW)
[ ] Config users (NEW)
[ ] Config busybox (NEW)
[*] Load default settings
[ ] Save default settings
---
  Load an Alternate Configuration File
  Save an Alternate Configuration File

<Select>  < Exit >  < Help >

```

Example III, settings for RTL8954C demo board as follows:

```
RLX Linux Configuration
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are
hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to
exit, <?> for Help, </> for Search. Legend: [*] built-in [ ] excluded <M> module < >
module capable

--- select components
Selected Target (rtl89xxC) --->
Selected Kernel (linux-2.6.30) --->
Selected Busybox (busybox-1.13) --->
Selected toolchain (rsdk-1.3.6-5281-EB-2.6.30-0.9.30) --->
--- rtl89xxC
Selected Board Configuration (V400_Ramfs + LE88221_2S + 92C + SAMBA) --->
--- config components
[*] Config kernel
[ ] Config users
[ ] Config busybox
[*] Load default settings
[ ] Save default settings
---
Load an Alternate Configuration File
q(p)

<Select> < Exit > < Help >
```

Example IV, settings for RTL8196CS demo board (iNIC) as follows:

```
RLX Linux Configuration
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are
hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to
exit, <?> for Help, </> for Search. Legend: [*] built-in [ ] excluded <M> module < >
module capable

--- select components
Selected Target (rtl8196cs) --->
Selected Kernel (linux-2.6.30) --->
Selected Busybox (busybox-1.13) --->
Selected toolchain (rsdk-1.3.6-4181-EB-2.6.30-0.9.30) --->
--- rtl8196cs
Selected Target of SDK (11n iNIC_AP) --->
Selected Board Configuration (8196cs 11n iNic AP ) --->
--- config components
[ ] Config kernel
[ ] Config users
[ ] Config busybox
[ ] Load default settings
[ ] Save default settings
---
Load an Alternate Configuration File
Save an Alternate Configuration File

<Select> < Exit > < Help >
```

Example V, setting for RTL8196D demo board as follows:

```
RLX Linux Configuration
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are
hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc>
to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ] excluded <M> module < >
module capable

--- select components
Selected Target (rtl819xD) --->
Selected Kernel (linux-2.6.30) --->
Selected Busybox (busybox-1.13) --->
Selected toolchain (rsdk-1.5.5-5281-EB-2.6.30-0.9.30.3-110714) --->
--- rtl819xD
Selected Board Configuration (96D+92D GW) --->
--- config components
[ ] Config kernel
[ ] Config users (NEW)
[ ] Config busybox
[*] Load default settings
[ ] Save default settings
---
Load an Alternate Configuration File
Save an Alternate Configuration File

<Select> < Exit > < Help >
```

Example VI, setting for RTL8197D demo board as follows:

```
          RLX Linux Configuration
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are
hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc>
to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ] excluded <M> module < >
module capable

--- select components
Selected Target (rtl819xD) --->
Selected Kernel (linux-2.6.30) --->
Selected Busybox (busybox-1.13) --->
Selected toolchain (rsdk-1.5.5-5281-EE-2.6.30-0.9.30.3-110714) --->
--- rtl819xD
Selected Board Configuration (97D+92C+92D GW) --->

--- config components
[ ] Config kernel
[ ] Config users (NEW)
[ ] Config busybox
[*] Load default settings
[ ] Save default settings
---

Load an Alternate Configuration File
Save an Alternate Configuration File

<Select> < Exit > < Help >
```

Example VII, setting for RTL8196CT demo board as follows:

```

----- RLX Linux Configuration -----
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted
letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes
features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*]
built-in [ ] excluded <M> module < > module capable [REDACTED]

--- select components [REDACTED]
  Selected Target (rtl18196ct) --->
  Selected Kernel (linux-2.6.30) --->
  Selected Busybox (busybox-1.13) --->
  Selected toolchain (rsdk-1.3.6-4181-EB-2.6.30-0.9.30) --->
--- rtl18196ct [REDACTED]
  Selected Target of SDK (linRouter_GW) --->
  Selected Board Configuration (SPI flash + Squashfs) --->
--- config components [REDACTED]
  [ ] Config kernel [REDACTED]
  [ ] Config users [REDACTED]
  [ ] Config busybox [REDACTED]
  [*] Load default settings [REDACTED]
  [ ] Save default settings [REDACTED]
--- [REDACTED]
  Load an Alternate Configuration File [REDACTED]
  Save an Alternate Configuration File [REDACTED]

-----<Select> < Exit > < Help >-----

```

Example VIII, setting for RTL8198T demo board as follows:

```

----- RLX Linux Configuration -----
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted
letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes
features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*]
built-in [ ] excluded <M> module < > module capable [REDACTED]

--- select components [REDACTED]
  Selected Target (rtl18198t) --->
  Selected Kernel (linux-2.6.30) --->
  Selected Busybox (busybox-1.13) --->
  Selected toolchain (rsdk-1.3.6-5281-EB-2.6.30-0.9.30) --->
--- rtl18198t [REDACTED]
  Selected Board Configuration (SPI flash + Squashfs) --->
--- config components [REDACTED]
  [ ] Config kernel [REDACTED]
  [ ] Config users [REDACTED]
  [ ] Config busybox [REDACTED]
  [*] Load default settings [REDACTED]
  [ ] Save default settings [REDACTED]
--- [REDACTED]
  Load an Alternate Configuration File [REDACTED]
  Save an Alternate Configuration File [REDACTED]

-----<Select> < Exit > < Help >-----

```

Example IX, setting for RTL8196E demo board as follows:

```
RLX Linux Configuration
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are
hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc>
to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ] excluded <M> module < >
module capable

--- select components
Selected Target (rtl8196e) --->
Selected Kernel (linux-2.6.30) --->
Selected Busybox (busybox-1.13) --->
Selected toolchain (rsdk-1.3.6-4181-EB-2.6.30-0.9.30) --->
--- rtl8196e
Selected Board Configuration (96E+92C GW) --->
--- config components
[ ] Config kernel
[ ] Config users (NEW)
[ ] Config busybox
[*] Load default settings
[ ] Save default settings
---
Load an Alternate Configuration File
Save an Alternate Configuration File

<Select> < Exit > < Help >
```

Example X, setting for SDK i1.0 demo board as follows:

```

----- RLX Linux Configuration -----
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys.
Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help,
</> for Search. Legend: [*] built-in [ ] excluded <M> module < > module capable

--- select components
Selected Target (rtl89xxD) --->
Selected Kernel (linux-2.6.30) --->
Selected Busybox (busybox-1.13) --->
Selected toolchain (rsdk-1.5.5-5281-EB-2.6.30-0.9.30.3-110714) --->
--- rtl89xxD
Selected Board Configuration (V100_Squashfs + VE890HV_2S + 92C + SAMBA) --->
--- config components
[ ] Config kernel (NEW)
[ ] Config users (NEW)
[ ] Config busybox (NEW)
[*] Load default settings
[ ] Save default settings

Load an Alternate Configuration File
Save an Alternate Configuration File

----->
<Select> < Exit > < Help >

```

Please select one of the following Board Configuration according to your board.

```

----- Selected Board Configuration -----
Use the arrow keys to navigate this window or press the hotkey of
the item you wish to select followed by the <SPACE BAR>. Press
<?> for additional information about this option.

( ) V100_Squashfs + No VoIP + 92C
( ) V100_Squashfs + VE890HV_2S10 + 92C
( ) V100_Squashfs + VE890HV_2S10 + 92D
( ) V100_Squashfs + VE890HV_2S10 + 92C + SAMBA
(X) V100_Squashfs + VE890HV_2S + 92C + SAMBA
( ) V100_Squashfs + VE880_2S + 92C + SAMBA

----->
<Select> < Help >

```

Example XI, setting for RTL8197D+8192C+8812AE +8367RB demo board as follows:

```

RLX Linux Configuration
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys.
Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help,
</> for Search. Legend: [*] built-in [ ] excluded <M> module < > module capable

--- select_components
Selected Target (rtl819xD) --->
Selected Kernel (linux-2.6.30) --->
Selected Busybox (busybox-1.13) --->
Selected toolchain (rsdk-1.5.5-5281-EB-2.6.30-0.9.30.3-110714) --->
--- rtl819xD
Selected Board Configuration (97D+8367R+92C+8812 GW) --->
--- config components
[ ] Config kernel
[ ] Config users
[ ] Config busybox
[*] Load default settings
[ ] Save default settings
---
Load an Alternate Configuration File
Save an Alternate Configuration File

<Select> < Exit > < Help >

```

Example XII, setting for RTL8197D+8812AE+8367RB+92ER demo board as follows:

```

RLX Linux Configuration
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys.
Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help,
</> for Search. Legend: [*] built-in [ ] excluded <M> module < > module capable

--- select_components
Selected Target (rtl819xD) --->
Selected Kernel (linux-2.6.30) --->
Selected Busybox (busybox-1.13) --->
Selected toolchain (rsdk-1.5.5-5281-EB-2.6.30-0.9.30.3-110714) --->
--- rtl819xD
Selected Board Configuration (97D+8367R+92E+8812 GW) --->
--- config components
[ ] Config kernel
[ ] Config users
[ ] Config busybox
[*] Load default settings
[ ] Save default settings
---
Load an Alternate Configuration File
Save an Alternate Configuration File

<Select> < Exit > < Help >

```

Example XIII, setting for RTL8881A+88ER demo board as follows:

```
RLX Linux Configuration
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys.
Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help,
</> for Search. Legend: [*] built-in [ ] excluded <M> module < > module capable

--- select components
  Selected Target (rtl8881a) --->
  Selected Kernel (linux-2.6.30) --->
  Selected Busybox (busybox-1.13) --->
  Selected toolchain (rsdk-1.5.5-5281-EB-2.6.30-0.9.30.3-110714) --->
--- rtl819xD
  Selected Board Configuration (8881A+88E) --->
--- config components
  [ ] Config kernel (NEW)
  [ ] Config users
  [ ] Config busybox
  [ ] Load default settings
  [ ] Save default settings
---
  Load an Alternate Configuration File
  Save an Alternate Configuration File

<Select>  < Exit >  < Help >
```

Example XIV, setting for RTL8881A+92ER demo board as follows:

```

RLX Linux Configuration
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys.
Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help,
</> for Search. Legend: [*] built-in [ ] excluded <M> module < > module capable

--- select components
  Selected Target (rtl881a) --->
  Selected Kernel (linux-2.6.30) --->
  Selected Busybox (busybox-1.13) --->
  Selected toolchain (rsdk-1.5.5-5281-EB-2.6.30-0.9.30.3-110714) --->
--- rtl819xD
  Selected Board Configuration (8881A+92E) --->
--- config components
  [ ] Config kernel (NEW)
  [ ] Config users
  [ ] Config busybox
  [*] Load default settings
  [ ] Save default settings
---
  Load an Alternate Configuration File
  Save an Alternate Configuration File

<Select>  < Exit >  < Help >

```

Example XV, setting for RTL8192E+8192E 2M/16M demo board as follows:

```

RLX Linux Configuration
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys.
Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help,
</> for Search. Legend: [*] built-in [ ] excluded <M> module < > module capable

--- select components
  Selected Target (rtl8196e) --->
  Selected Kernel (linux-2.6.30) --->
  Selected Busybox (busybox-1.13) --->
  Selected toolchain (rsdk-1.3.6-4181-EB-2.6.30-0.9.30) --->
--- rtl8196e
  Selected Board Configuration (92E+92E GW 16M) --->
--- config components
  [ ] Config kernel
  [ ] Config users
  [ ] Config busybox
  [*] Load default settings
  [ ] Save default settings
---
  Load an Alternate Configuration File
  Save an Alternate Configuration File

<Select>  < Exit >  < Help >

```

Example XVI, setting for RTL8881A Selectable demo board as follows:

```
RLX Linux Configuration
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>
for Search. Legend: [*] built-in [ ] excluded <M> module < >

--- select components
Selected Target (rtl8881a) --->
Selected Kernel (linux-2.6.30) --->
Selected Busybox (busybox-1.13) --->
Selected toolchain (rsdk-1.5.5-5281-EB-2.6.30-0.9.30.3-110714
--- rtl819xD
Selected Board Configuration (8881A Selectable) --->
--- config components
[ ] Config kernel
[ ] Config users
[ ] Config busybox
[*] Load default settings
[ ] Save default settings
---
Load an Alternate Configuration File
Save an Alternate Configuration File

<Select> < Exit > < Help >
```

Example XVII, setting for RTL8198C+8812AE+92ER demo board as follows:

```
RLX Linux Configuration
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys.
Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help,
</> for Search. Legend: [*] built-in [ ] excluded <M> module capable

--- select components
Selected Target (rtl8198C_8954E) --->
Selected Kernel (linux-3.10) --->
Selected Busybox (busybox-1.13) --->
Selected toolchain (msdk-4.4.7-mips-EB-3.10-0.9.33-m32t-131227b) --->
--- rtl8198c
Selected Board Configuration (RTL8198C/8954E+8812+92E GW) --->
--- config components
[ ] Config kernel
[ ] Config users
[ ] Config busybox
[*] Load default settings
[ ] Save default settings
---
Load an Alternate Configuration File
Save an Alternate Configuration File

<Select> < Exit > < Help >
```

Example XVIII, setting for RTL8198CD+8812AE+92ER demo board as follows:

```
RLX Linux Configuration
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys.
Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help,
</> for Search. Legend: [*] built-in [ ] excluded <M> module < > module capable

--- select components
Selected Target (rtl8198C_8954E) --->
Selected Kernel (linux-3.10) --->
Selected Busybox (busybox-1.13) --->
Selected toolchain (msdk-4.4.7-mips-EB-3.10-0.9.33-m32t-131227b) --->
--- rtl8198c
Selected Board Configuration (RTL8198CD/8954E+8812+92E GW) --->
--- config components
[ ] Config kernel
[ ] Config users
[ ] Config busybox
[*] Load default settings
[ ] Save default settings
---
Load an Alternate Configuration File
Save an Alternate Configuration File

<Select> < Exit > < Help >
```

Example XIX, setting for RTL8198CS+8812AE+92ER demo board as follows:

```
RLX Linux Configuration
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys.
Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help,
</> for Search. Legend: [*] built-in [ ] excluded <M> module < > module capable

--- select components
Selected Target (rtl8198C_8954E) --->
Selected Kernel (linux-3.10) --->
Selected Busybox (busybox-1.13) --->
Selected toolchain (msdk-4.4.7-mips-EB-3.10-0.9.33-m32t-131227b) --->
--- rtl8198c
Selected Board Configuration (RTL8198CS/8954ES+8812+92E GW) --->
--- config components
[ ] Config kernel
[ ] Config users
[ ] Config busybox
[*] Load default settings
[ ] Save default settings
---
Load an Alternate Configuration File
Save an Alternate Configuration File

<Select> < Exit > < Help >
```

iii) Select ‘Load default settings’, then choose ‘<Exit>’ and click ‘Enter’

Note:

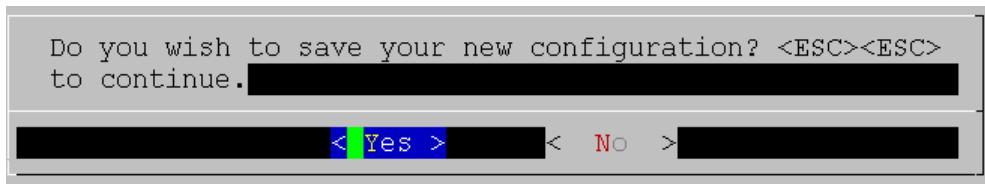
Above we just use configure applications and linux kernel in default way.

If we want to customize linux kernel, we can select ‘Config kernel’;

if we want to customize applications, we can select ‘Config users’;

if we want to customize busybox, we can select ‘Config busybox’;

then choose ‘<Exit>’ and click ‘Enter’.



iv) Choose ‘<Yes>’ and click ‘Enter’

Step 3: compile rlxlinux.

Make

2.4. Compile rlxlinux detailed

2.4.1 Compile rlxlinux process

Compile rlxlinux include several steps as follows.

- (1) Compile user dir (rtl819x/users);
- (2) Compile board dir (rtl819x/boards/.....) and compile user dir (rtl819x/users) romfs;
- (3) Compile linux dir (rtl819x/linux-2.6.30 or rtl819x/linux-3.10) module;
- (4) Compile linux dir (rtl819x/linux-2.6.30 or rtl819x/linux-3.10);
- (5) Create image dir (rtl819x/image) and compile board dir(rtl819x/boards/.....) image.

2.4.2 Images generated

After compile rlxlinux, several images are generated at image dir (rtl819x/image).

- (1) Full image include webpages, root file system and linux kernel, which can be uploaded via webpage or tftp.

fw.bin

- (2) Image includes webpages only, which can be uploaded via webpage or tftp.

webpages-gw.bin

- (3) Image include root file system only, which can be uploaded via webpage or tftp.

root.bin

(4) Image include linux kernel only, which can be uploaded via webpage or tftp.

linux.bin

(5) Configure file of default settings, which can be uploaded via webpage.

config-gw-* .dat

2.4.3 Tools to make images

To make images, tools such as cvcfg-gw, compweb, cvimg, mgbin are stored at rtl819x/users/goahead-2.1.1/LINUX or rtl819x/users/boa/tools if Boa web server is selected.

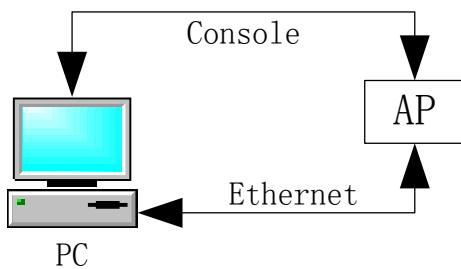
More info about these tools, please refer to [Chapter 5 FLASH tools](#).

3. AP-Router SDK images upload

3.1 Topology to upload image

Topology to upload image is as follows:

Connect AP's LAN port with PC to transmit data and connect AP's UART port with PC's serial port to send command.



3.2 Upload bootloader image

Via 2.1 bootloader compilation, bootloader image is generated at bootcode_rtl8196c_98 (bootcode_rtl819xD or bootcode_rtl8198c)/btcode/ which is boot. bin.

Steps to upload bootloader image are as follows:

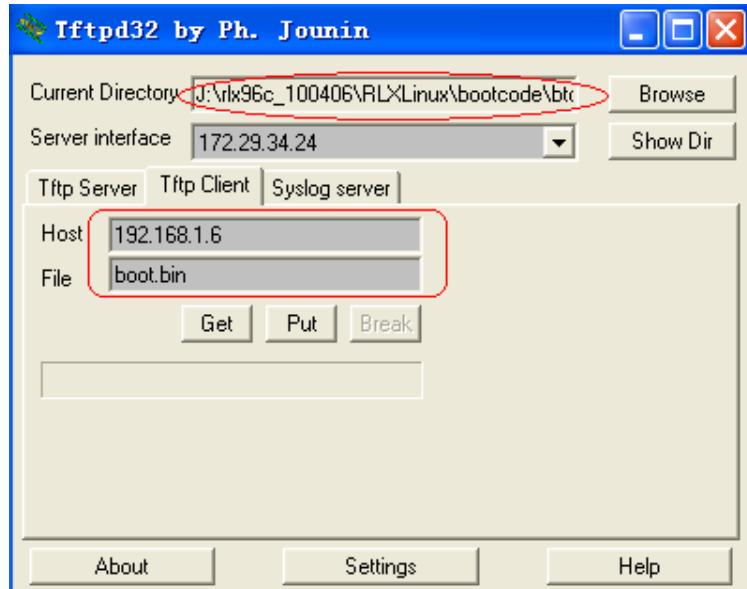
Step 1, reboot AP and click 'Esc' via console during AP booting, then AP enter bootloader environment. Output of AP's console is as follows:

```
Booting...
=====
      SPI =====

---RealTek(RTL8196C)at 2010.03.29-16:14+0800 version v1.1a [16bit](390MHz)
no sys signature at 00010000!
no sys signature at 00020000!
Set 8196C PHY Patch OK

---Ethernet init Okay!
<RealTek>
```

Step 2, upload bootloader image via AP's lan port using tftpd. Settings of tftpd are as follows:



Step 3, click ‘Put’ at tftp UI.

3.3 Upload rlxlinux image

Via 2.2 rlxlinux compilation, images are generated at rtl819x/image/ which are webpages-gw.bin, root.bin, linux.bin or fw.bin (which include webpages-gw.bin, root.bin, linux.bin). To upload rlxlinux images, we need to upload fw.bin or webpages-gw.bin, root.bin, linux.bin three images.

Steps to upload rlxlinux images are as follows:

Step 1, reboot AP and click ‘Esc’ via console during AP booting, then AP enter bootloader environment. Wait few seconds for Ethernet initialization. Output of AP’s console is as follows:

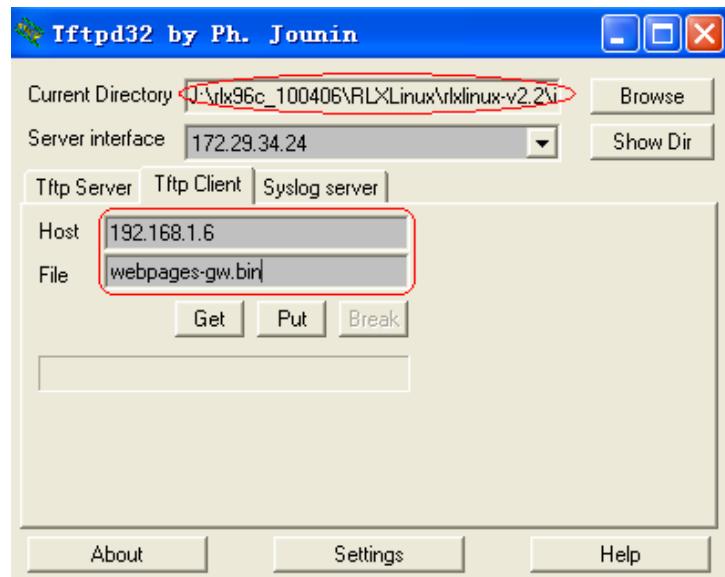
```
Booting...
===== SPI =====

---RealTek(RTL8196C)at 2010.03.29-16:14+0800 version v1.1a [16bit] (390MHz)
no sys signature at 00010000!
no sys signature at 00020000!
Set 8196C PHY Patch OK

---Ethernet init Okay!
<RealTek>[
```

Step 2, upload rlxlinux images via AP’s lan port using tftpd. Settings of tftp to upload webpages-gw.bin are as follows.

And settings of tftp to upload root.bin, linux.bin or fw.bin are the similar, just use root.bin, linux.bin or fw.bin as ‘File’ input at tftp UI instead.



Step 3, click ‘Put’ at tftp UI.

After fw.bin or webpages-gw.bin, root.bin and linux.bin are uploaded, AP boots up, and input IP (get it via console “ifconfig”, default 192.168.1.254) at the browser, the webpage will appear.

The setup wizard will guide you to configure access point for first time. Please follow the setup wizard step by step.

Welcome to Setup Wizard.

The Wizard will guide you through following steps. Begin by clicking on Next.

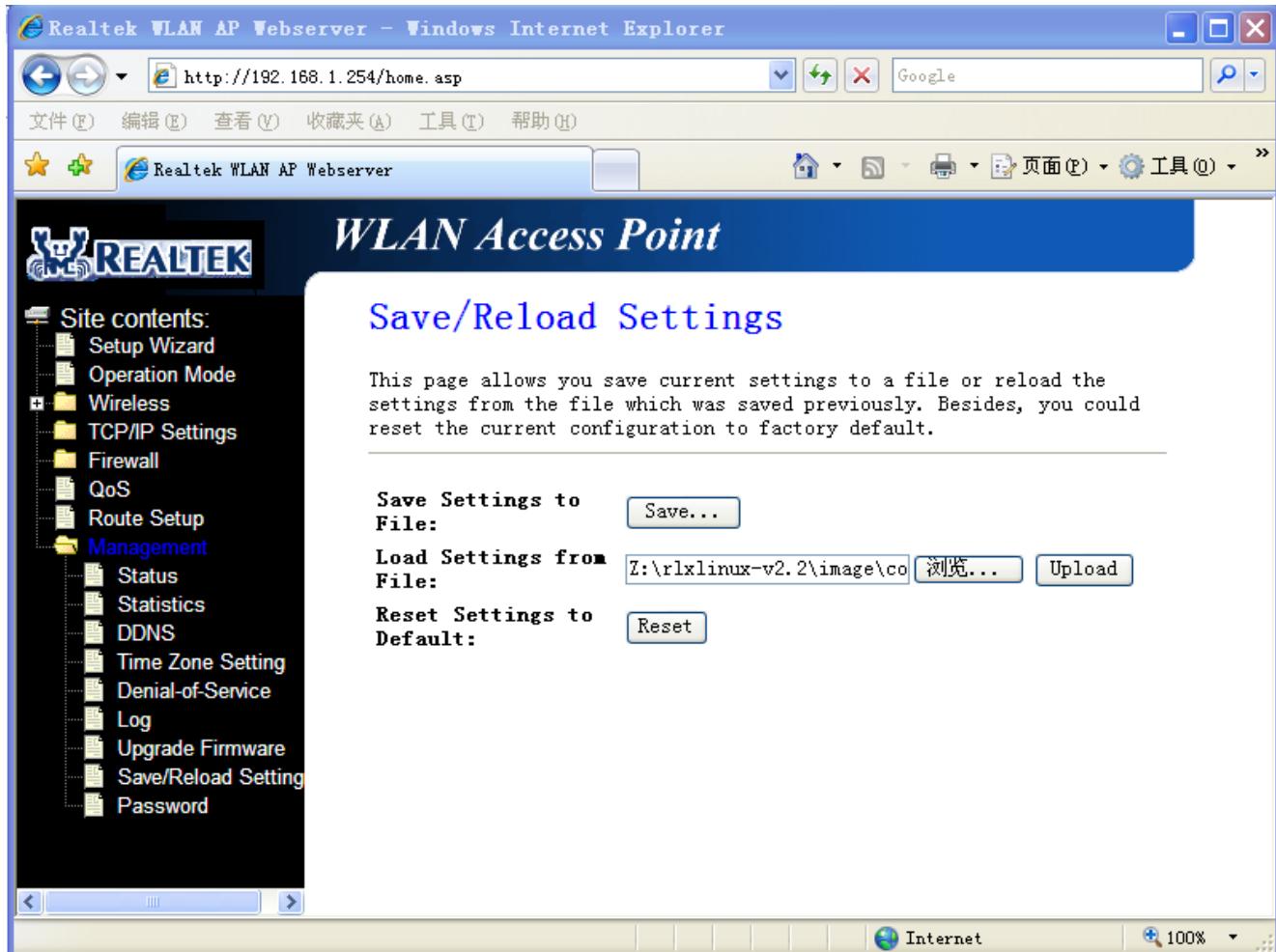
1. Setup Operation Mode
2. Choose your Time Zone
3. Setup LAN Interface
4. Setup WAN Time Zone
5. Select Wireless Band
6. Wireless LAN Setting
7. Wireless Security Setting

Next>>

3.4 Upload default setting data (Optional)

Via 2.2 rlxlinux compilation, default setting data are generated at rtl819x/image/, named like config-gw-98-92c-92d.dat.

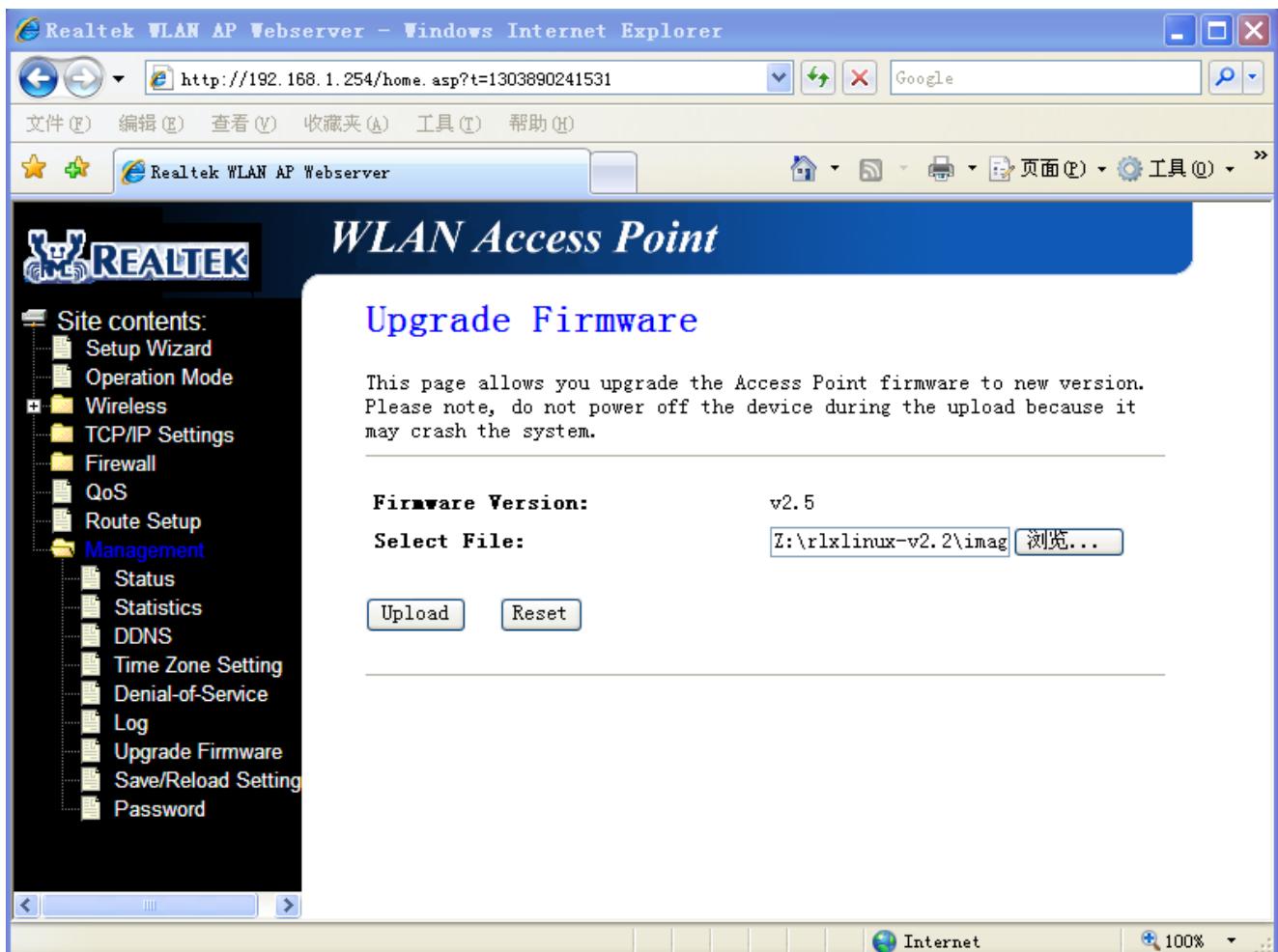
To upload default setting data, open AP webpage, select Management->Save/Reload Settings, browse and select config data file(ex: config-gw-98-92c-92d.dat) then upload.



After upload complete, AP will reboot.

3.5 Upload rlxlinux image via webpage (Optional)

Open AP webpage, select Management->Upgrade Firmware browse and select image file (ex: rtl819x/image/fw.bin), then upload.



After upload complete, AP will reboot.

3.6 Upload rlxlinux image with config file via webpage (Optional)

You can upload rlxlinux image with configuration file at one time via webpage by following step.

- a. Copy the tool "mgbn" to "rlxlinux-v2.2/image" first.
- b. Combine rlxlinux image and configuration file by "./mgbn -c -o all_fw.bin ./root.bin ./webpages.bin ./config-gw-xxx-xxx.dat ./linux.bin"
- c. The all_fw.bin is the rlxlinux image with configuration file.
- d. Upload the all_fw.bin via webpage like below.

Realtek WLAN AP Webserver - Windows Internet Explorer

http://192.168.1.254/home.asp?t=1303890241531

文件 (F) 编辑 (E) 查看 (V) 收藏夹 (A) 工具 (T) 帮助 (H)

Realtek WLAN AP Webserver

WLAN Access Point

Upgrade Firmware

This page allows you upgrade the Access Point firmware to new version. Please note, do not power off the device during the upload because it may crash the system.

Firmware Version: v2.5

Select File: Z:\rlxlinux-v2.2\image [浏览...]

Upload **Reset**

4. Features configure

4.1 PCI support

4.1.1 Linux kernel PCI not support

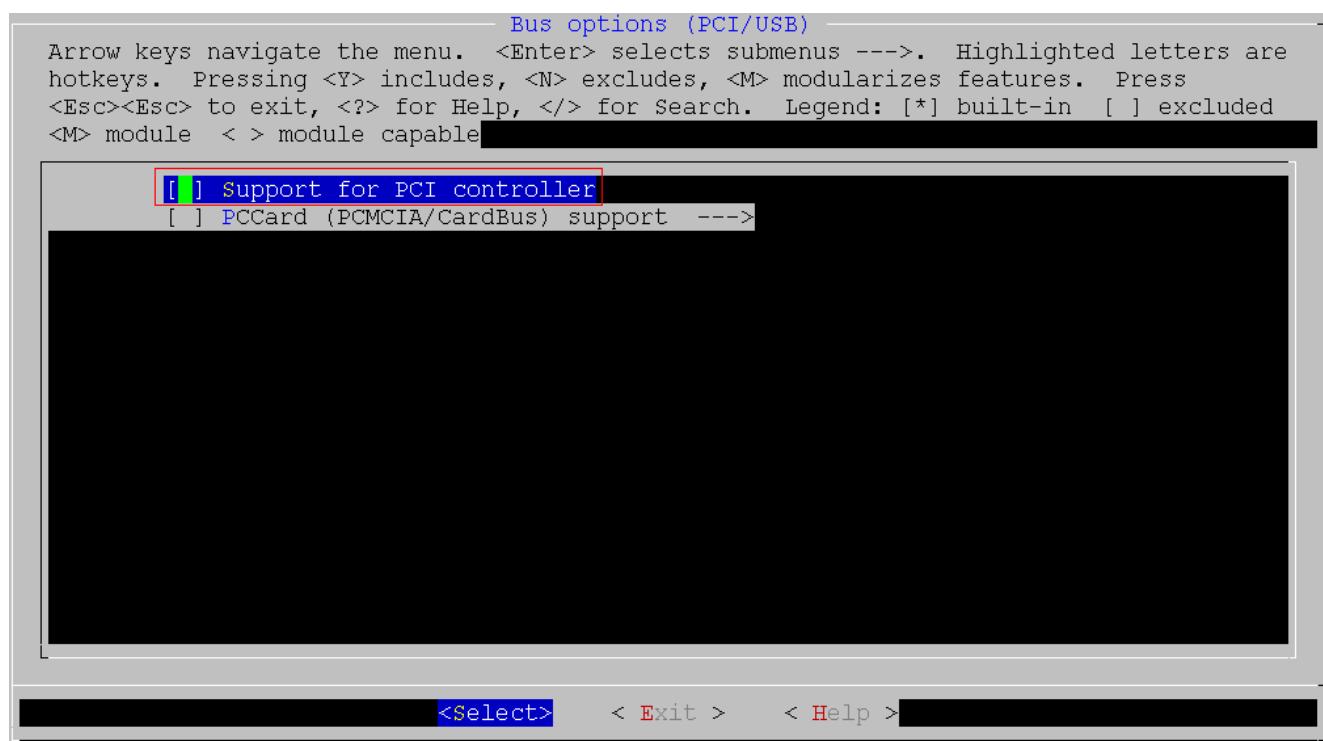
By kernel default settings, linux kernel PCI is not supported, because wireless driver use its own PCI driver. Linux kernel configure as follows.

```
make linux_menuconfig      // To configure linux kernel settings
```

Menuconfig:

Bus options (PCI/USB) --->

Support for PCI controller // Not selected



4.1.2 Linux kernel PCI support

If wireless driver use the PCI BIOS driver of linux kernel, linux kernel configure as follows.

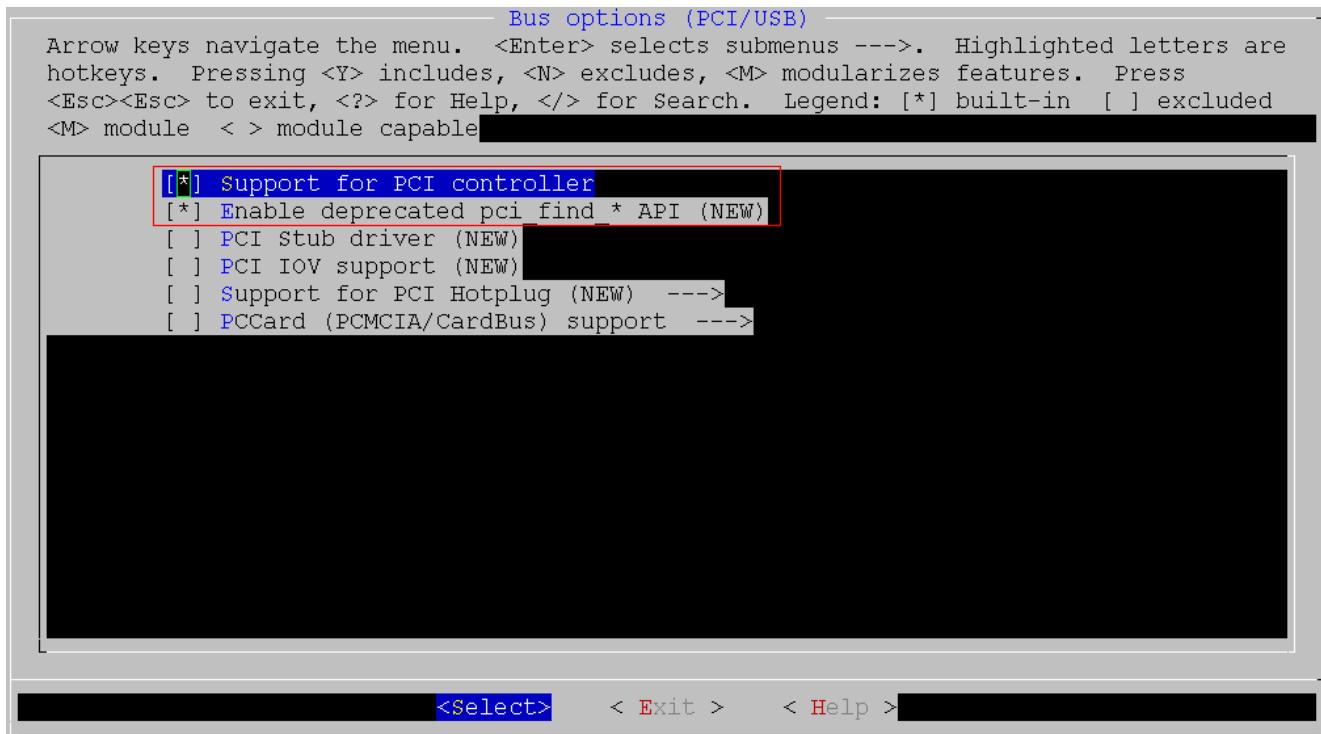
```
make linux_menuconfig      // To configure linux kernel settings
```

Menuconfig:

Bus options (PCI/USB) --->

Support for PCI controller // Selected

Enable deprecated pci find * API (NEW) // Selected



4.2 USB support

(1) If USB is not supported, linux kernel configure for USB as follows.

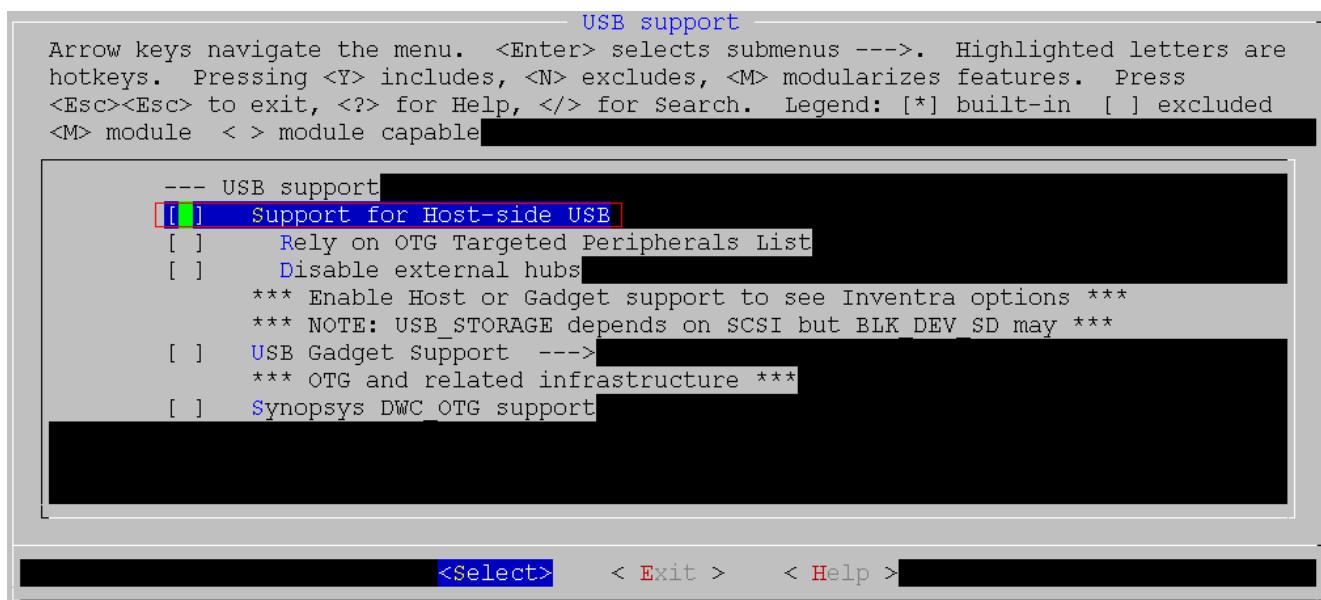
make linux_menuconfig // To configure linux kernel settings

Menuconfig:

Device Drivers --->

 USB support --->

 Support for Host-side USB // Not selected



(2) If USB is supported, linux kernel configure for USB as follows (Except of **RTL8196E**).

make linux_menuconfig // To configure linux kernel settings

Menuconfig:

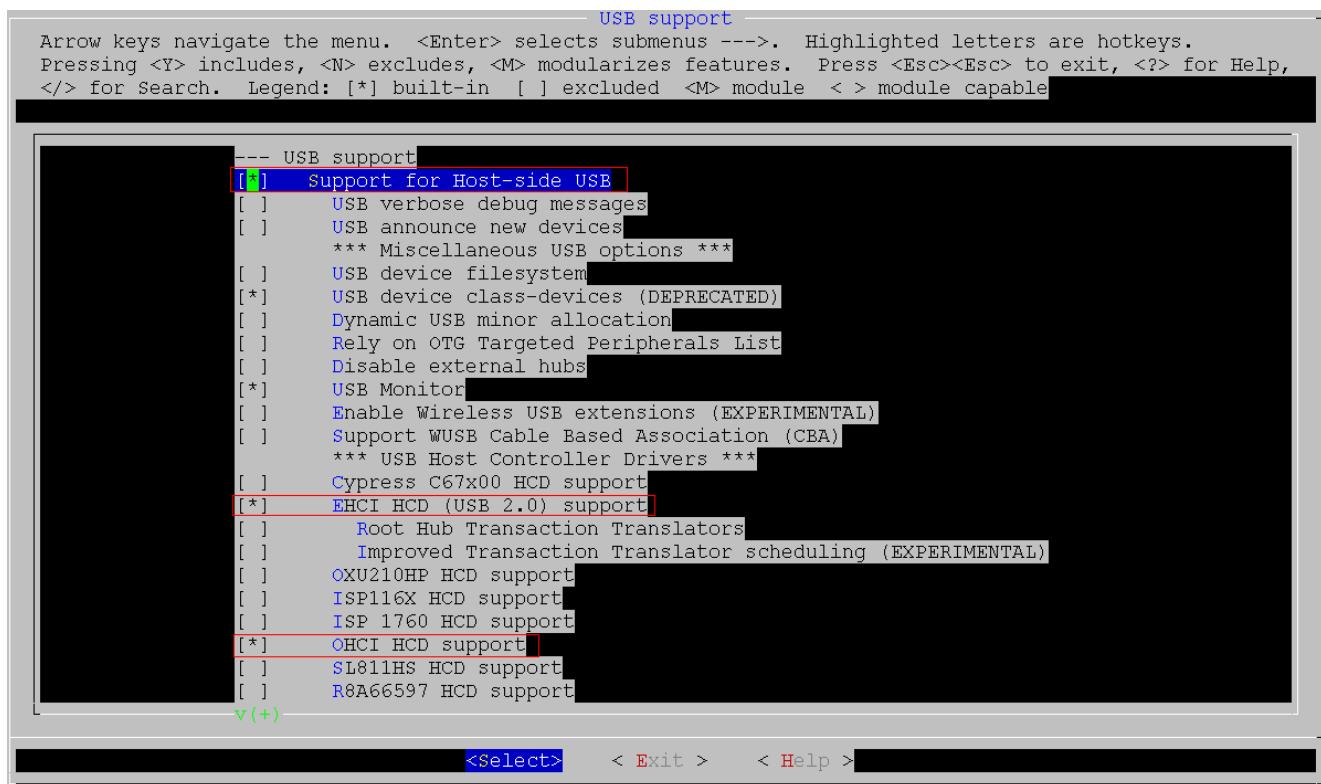
Device Drivers -->

USB support -->

Support for Host-side USB // selected

EHCI HCD (USB 2.0) support // To support USB 2.0, if selected

OHCI HCD support // To support USB 1.1, if selected



(3) If OTG USB is supported, linux kernel configure for OTG USB as follows in the (**RTL8196E**, **RTL8197D**).

make linux_menuconfig // To configure linux kernel settings

Menuconfig:

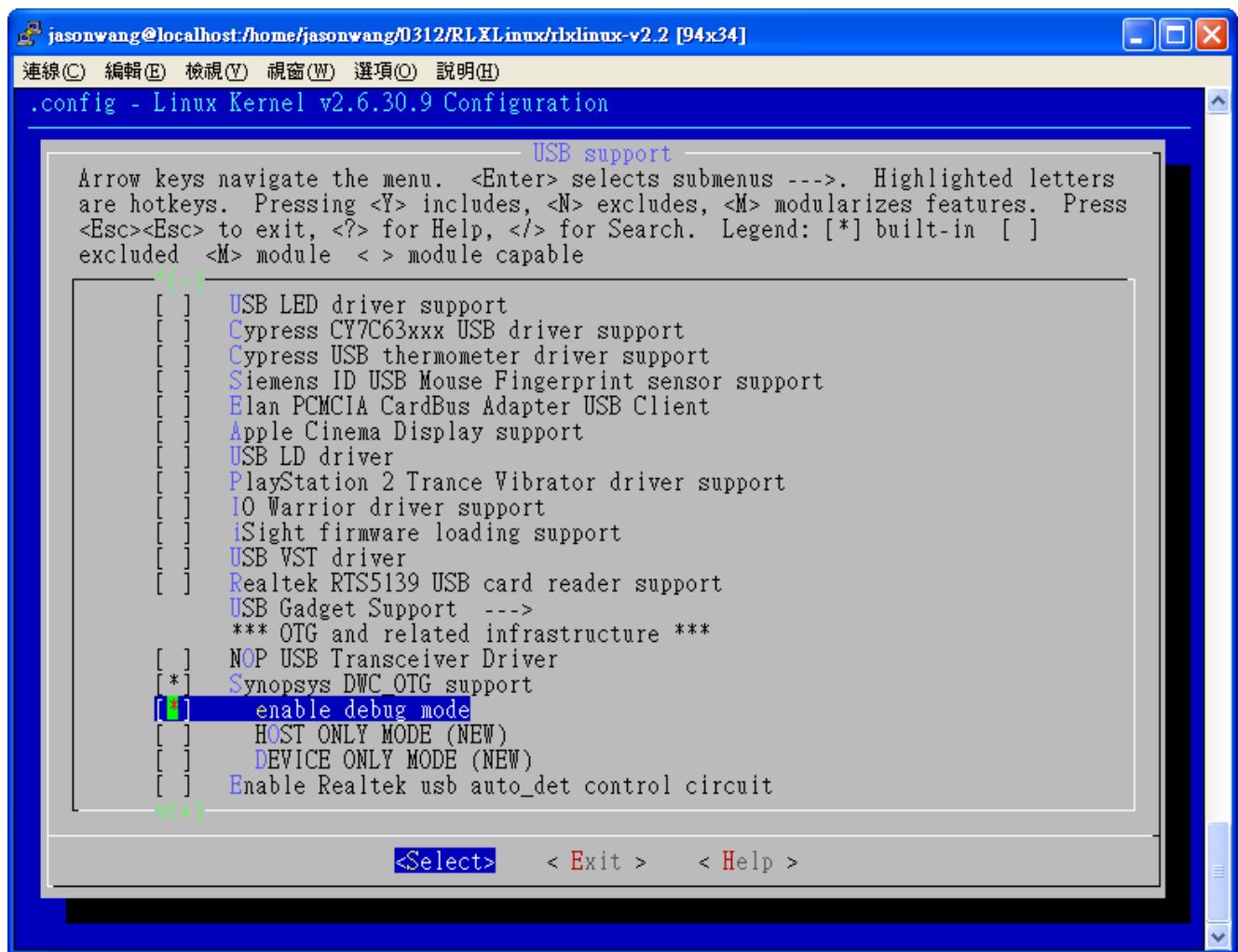
Device Drivers -->

USB support -->

Support for Host-side USB // selected

Synopsys DWC_OTG support

enable debug mode



4.3 Samba support

4.3.1 Samba kernel configure

4.3.1.1 SCSI support

Linux kernel configure for SCSI as follows.

```
make linux_menuconfig // To configure linux kernel settings
```

Menuconfig:

Device Drivers --->

SCSI device support --->

SCSI device support // selected

SCSI disk support // selected

```

SCSI device support
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are
hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press
<Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ] excluded
<M> module < > module capable [ ] RAID Transport Class
[*] SCSI device support [ ] SCSI target support
[*] legacy /proc/scsi/ support
*** SCSI support type (disk, tape, CD-ROM) ***
[*] SCSI disk support [ ] SCSI tape support
[ ] SCSI Onstream SC-x0 tape support
[ ] SCSI CDROM support
[ ] SCSI generic support
[ ] SCSI media changer support
*** Some SCSI devices (e.g. CD jukebox) support multiple LUNs ***
[ ] Probe all LUNs on each SCSI device
[ ] Verbose SCSI error reporting (kernel size +=12K)
[ ] SCSI logging facility
[ ] Asynchronous SCSI scanning
v(+)

```

<Select> < Exit > < Help >

4.3.1.2 File systems

Linux kernel configure for file systems as follows.

make linux_menuconfig // To configure linux kernel settings

Menuconfig (1):

File systems --->

Enable POSIX file locking API // selected

```

[ ] XFS filesystem support
[ ] OCFS2 file system support
[ ] Btrfs filesystem (EXPERIMENTAL) Unstable disk format
[*] Enable POSIX file locking API
[ ] Dnotify support
[ ] Inotify file change notification support

```

Menuconfig (2):

File systems --->

DOS/FAT/NT Filesystems --->

MSDOS fs support // selected

```

DOS/FAT/NT Filesystems
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are
hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press
<Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ] excluded
<M> module < > module capable
```

<Select> < Exit > < Help >

Menuconfig (3):

File systems --->

Native Language Support --->

Codepage 437 (United States, Canada) //Selected

NLS ISO 8859-1 (Latin 1; Western European Languages) // Selected

```

[*] Base native language support
(iso8859-1) Default NLS Option (NEW)
[*] Codepage 437 (United States, Canada)
[ ] Codepage 737 (Greek) (NEW)
[ ] Codepage 775 (Baltic Rim) (NEW)
[ ] Codepage 850 (Europe) (NEW)
[ ] Codepage 852 (Central/Eastern Europe) (NEW)
[ ] Codepage 855 (Cyrillic) (NEW)
[ ] Codepage 857 (Turkish) (NEW)
[ ] Codepage 860 (Portuguese) (NEW)
[ ] Codepage 861 (Icelandic) (NEW)
[ ] Codepage 862 (Hebrew) (NEW)
[ ] Codepage 863 (Canadian French) (NEW)
[ ] Codepage 864 (Arabic) (NEW)
[ ] Codepage 865 (Norwegian, Danish) (NEW)
[ ] Codepage 866 (Cyrillic/Russian) (NEW)
[ ] Codepage 869 (Greek) (NEW)
[ ] Simplified Chinese charset (CP936, GB2312) (NEW)
[ ] Traditional Chinese charset (Big5) (NEW)
[ ] Japanese charsets (Shift-JIS, EUC-JP) (NEW)
[ ] Korean charset (CP949, EUC-KR) (NEW)
[ ] Thai charset (CP874, TIS-620) (NEW)
[ ] Hebrew charsets (ISO-8859-8, CP1255) (NEW)
[ ] Windows CP1250 (Slavic/Central European Languages) (NEW)
[ ] Windows CP1251 (Bulgarian, Belarusian) (NEW)
[ ] ASCII (United States) (NEW)
[*] NLS ISO 8859-1 (Latin 1; Western European Languages)
[ ] NLS ISO 8859-2 (Latin 2; Slavic/Central European Languages) (NEW)
[ ] NLS ISO 8859-3 (Latin 3; Esperanto, Galician, Maltese, Turkish) (NEW)
```

4.3.1.3 USB support

Linux kernel configure for USB please see section 4.2, USB Support.

4.3.1.4 General setup

make linux_menuconfig // To configure linux kernel settings

Menuconfig:

General setup --->

--- Configure standard kernel features (for small systems) --->

Support for hot-pluggable devices // selected

```
[*] Optimize for size
--*- Configure standard kernel features (for small systems) --->
[*] Strip assembler-generated symbols during link
[*] Support for hot-pluggable devices
[*] Enable support for printk log
[ ] Enable support for printk console
```

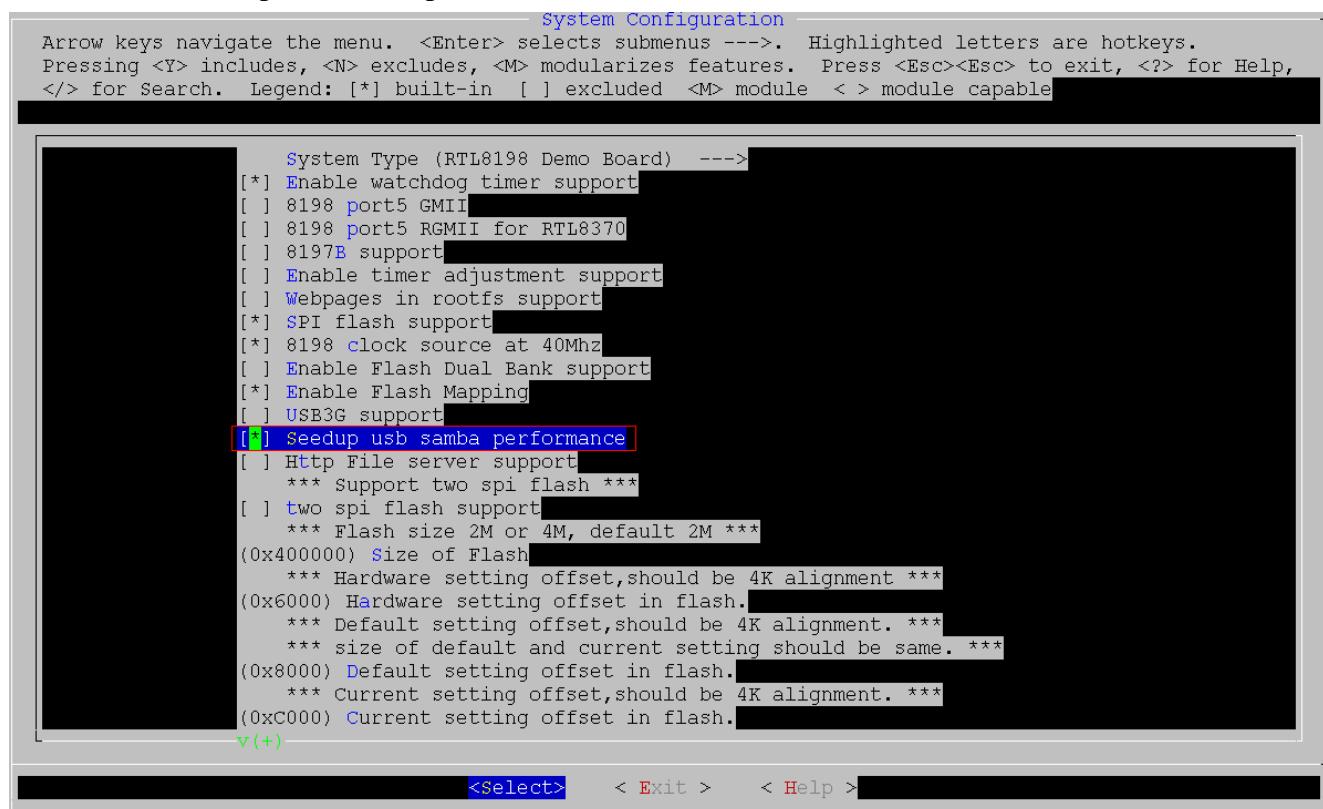
4.3.1.5 System configure

make linux_menuconfig // To configure linux kernel settings

Menuconfig:

System Configuration --->

[*] Seadup usb samba performance



4.3.2 Samba application configure

4.3.2.1 enable usbmount

make users_menuconfig // To configure application settings

Menuconfig:

usbmount // selected

```

[ ] uLinker autorun image tool
[ ] updatedd DDNS
[*] usbmount
[ ] usbdisk update image (NEW)
[ ] WiFi
[ ] vsftpd
[ ] wireless tools
[ ] wsc daemon
    Libraries Settings --->
    Debug & Test Settings --->
---
Load an Alternate Configuration File
Save an Alternate Configuration File

<Select> < Exit > < Help >

```

4.3.2.2 enable samba

make users_menuconfig // To configure application settings

Menuconfig:

[*] samba support ---> // selected

Note: when samba is selected here, default config file should be uploaded via webpage to enable samba.

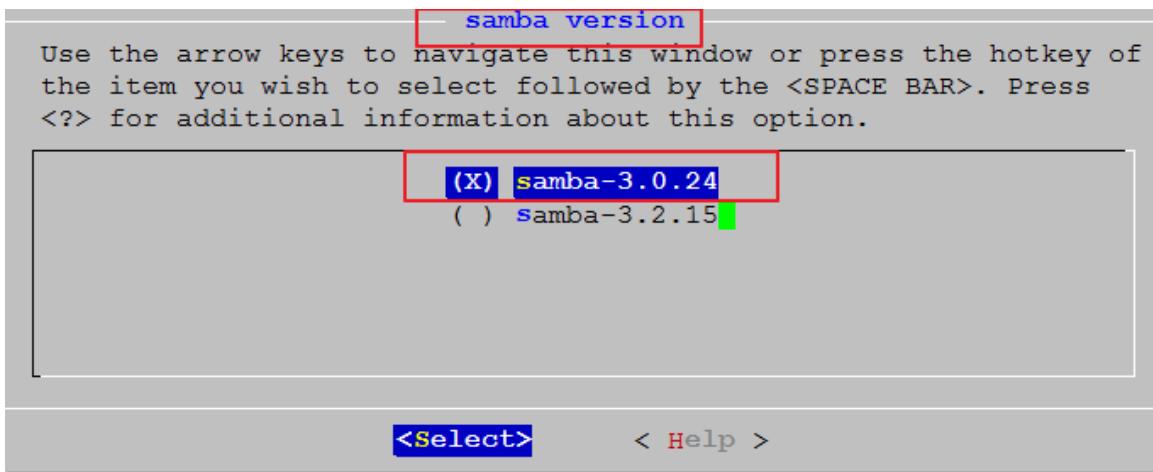
```

RLX Linux Configuration
the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, <X> for excluded <M> module < > module capable

--- (-)
[ ] oray ddns
[ ] pathsel
[ ] pmc utils
[ ] python support
[ ] reload
[ ] routed
[*] rtk_cmd daemon
[ ] RTK tcp2dev utility
[ ] rtk_voip
[ ] RTL8686 IPC User Utility
[*] samba support --->
[ ] scripts
[ ] simple config support

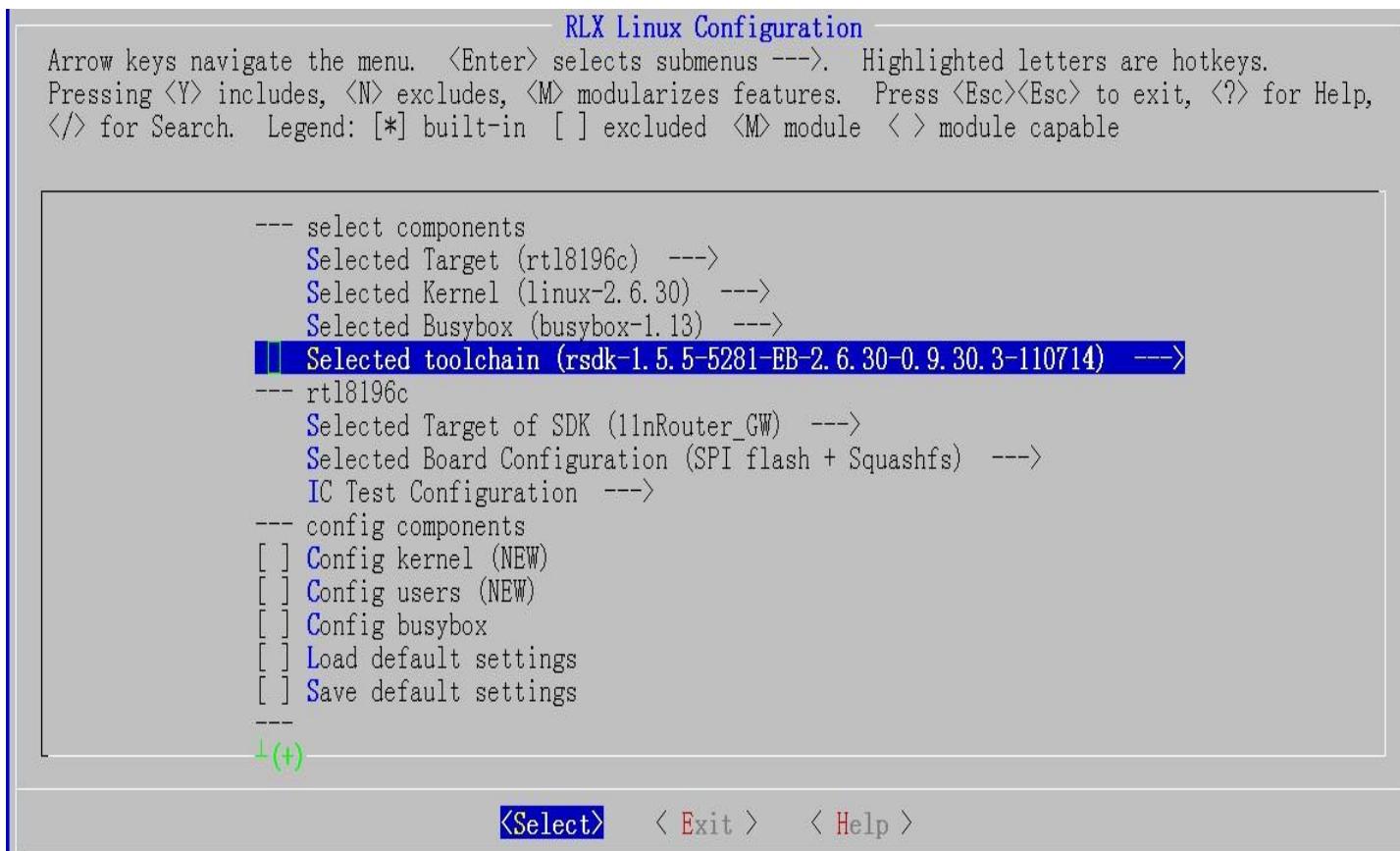
--- samba support
samba version (samba-3.0.24) --->

```



Note: For RTL8196D/RTL8197D, samba-3.2.15 has a better performance; while for RTL8198/RTL8196c, samba-3.0.24 is recommended.

To select samba-3.2.15, toolchain must choose rsdk-1.5.5.



4.3.3 Samba Enable on board

To enable Samba on board, the mib SAMBA_ENABLED must be set 1(default 0).

Two ways to set the mib.

a) via Console:

```
#flash all / grep SAMBA
DEF_SAMBA_ENABLED=0
```

```

SAMBA_ENABLED=0
#flash set DEF_SAMBA_ENABLED 1
#flash set SAMBA_ENABLED 1
#flash all / grep SAMBA
DEF_SAMBA_ENABLED=1
SAMBA_ENABLED=1

```

b) via webpage

- (1)Edit *users/boa/defconfig/config-gw-xx.txt* (use boa) or *users/goahead-2.1.1/LINUX/config-gw-xx.txt* (use goahead) to set *DEF_SAMBA_ENABLED* and *SAMBA_ENABLED* 1
- (2)Build the image, and find the config.dat file in *image/config-gw-xx.dat*
- (3)Upload the config.dat file via webpage Management→Save/Load Settings
- (4)After rebooting the mib will be set. Use *flash all / grep SAMBA* to check

4.3.4 Test Samba using USB flash disk

After system boots up, plug-in an USB Flash disk, you can read/write the USB flash in /tmp/usb directory.

4.3.5 How to add Samba user

After system boots up, start to add Samba user (testuser) as follows:

```

#echo "testuser:x:0:0:testuser:/dev/null" >> /var/passwd
#echo "[testuser]
comment = testuser's stuff
path = /var/log
valid users = testuser
public = no
writable = no
printable = no
create mask = 0765" > /var/smbuser.conf
# smbpasswd -a testuser
New SMB password:          // input new SMB password here
Retype new SMB password:   // re-input new SMB password here
Added user testuser.
# killall smbd
# smbd -D

```

Note: at present, only rtl8198 and rtl8954C SDK support this feature.

4.3.6 Factors related to test performance

There are several factors that have effect on the performance of samba test; the related factors are as follows:

- (1) When we test, enable the config: RTL_USB_IP_HOST_SPEEDUP in kernel config, the performance will be improved about 20%;

- (2) The usb disk used for test. Sata disk will improve the performance about 10%-20% compared with the ordinary USB flash disk;
- (3) The test card. New card will have better performance, 98-v631 can improve 15% compared with 98-v630;
- (4) The test computer. Different computer may result in different test performance;
- (5) The test software. CrystalDiskMark and fastcopy could have different performance as much as 10%. If choose CrystalDiskMark for test, the cycle number and disk size are related too, we choose 5/50M, which is better than just once.
- (6) Toolchain. The given test result is based on rsdk-1.3.6. The samba size will be smaller on rsdk-1.5.5, and the performance will be better.

4.3.7 How to enable ntfs file system for samba

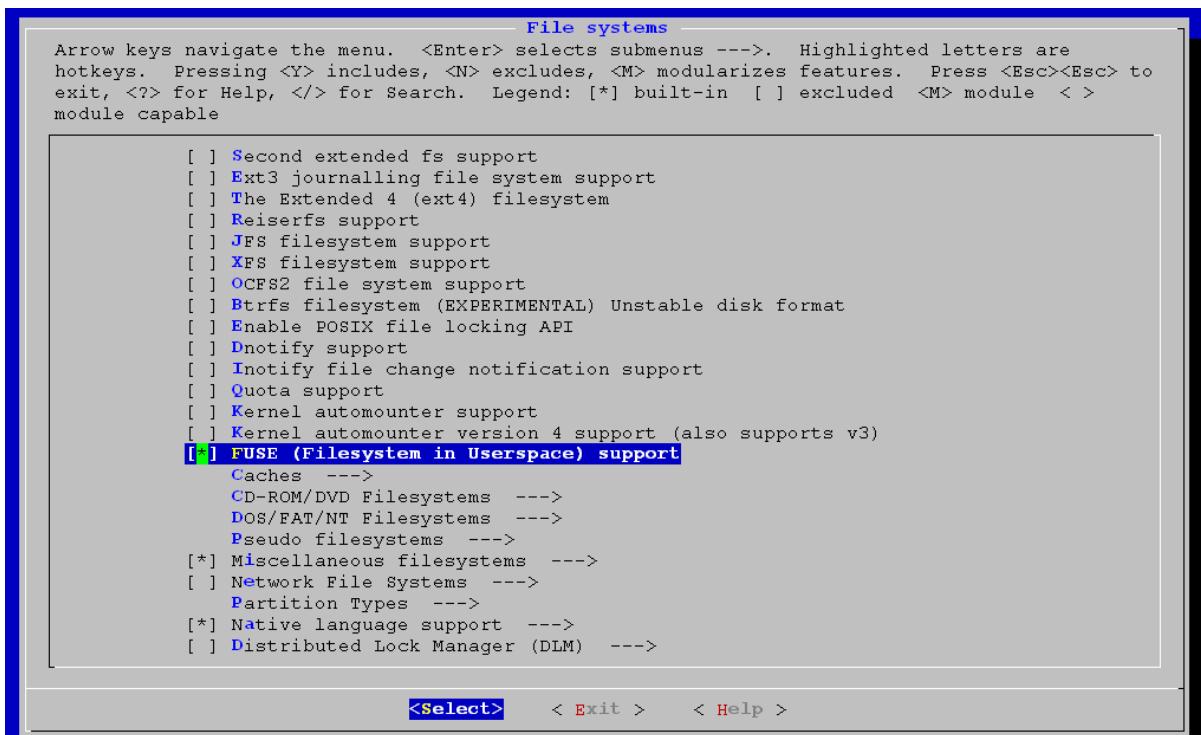
4.3.7.1 Linux kernel configures for ntfs file systems as follows.

make linux_menuconfig // To configure linux kernel settings

Menuconfig:

File systems --->

FUSE (Filesystem in Userspace) support // selected



4.3.7.2 users configures for ntfs file systems as follows.

make users_menuconfig // To configure application settings

Menuconfig:

ntfs3g // selected

```
----- RLX Linux Configuration -----
the menu. <Enter> selects submenus --->. Highlighted letters are hot-
les, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help,
cluded <M> module < > module capable

--- Applications Settings
  BT Configuration --->
  File_system Configuration --->
  IPv6 Configuration --->
  PPP Configuration --->
  RTK_inband Configuration --->
  USB3G Configuration --->
  [ ] alsound related
  [ ] auth
```

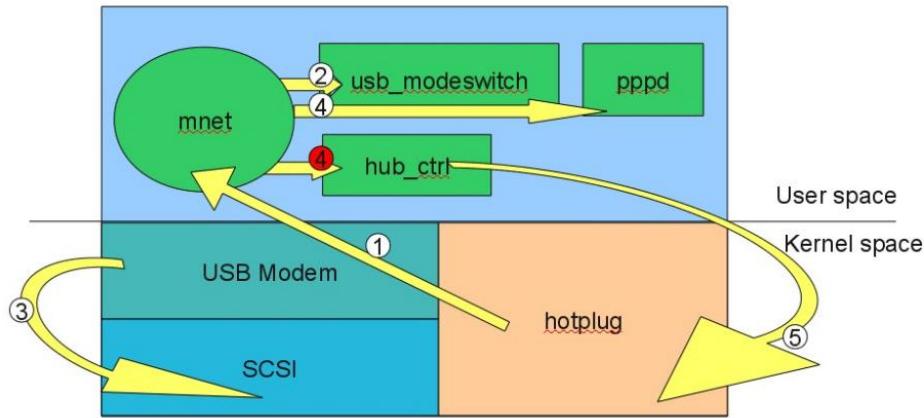
```
----- File_system Configuration -----
the menu. <Enter> selects submenus --->. Highlighted letters are hot-
les, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help,
cluded <M> module < > module capable

[ ] dosfsck
[ ] e2fsprogs
[ ] exfat-fuse
[ ] mkdosfs
[ ] NFS utility
[ ] ntfs3g
[ ] rti_fs
```

4.4 USB 3G support

4.4.1 Basic software flow

- ① Get the hotplug event.
- ② Find the right config for usb_modeswitch.
- ③ Monitor and control the activity of eject storage usb.
- ④ Prepare /var/usb3.option and /var/usb3g.chat and call pppd-chat.
- ⑤ If eject fail or call pppd fail, execute hub_ctrl to disconnect usb device, then run from beginning.



Ps.

mnet: a utility to control the above flow.

usb_modeswitch: a mode switching tool for controlling multiple device USB (turn from storage mode into modem mode).

4.4.2 Linux kernel configure

In System Configuration, please select USB3G support and change the size of Flash to 0x400000, and linux image offset to 0x130000 (make sure your board is 32M/4M sdram/flash, and bootloader is also compatible).

```

System Type (RTL8196C Demo Board) --->
[*] Enable watchdog timer support
[ ] Enable timer adjustment support
[ ] Webpages in rootfs support
[*] SPI flash support
[*] Enable Flash Mapping
[ ] Pocket router support
[ ] Domain name query support
[*] USB3G support
    *** please select all options in menu: USB3G Kernel Depends ***
    USB3G Kernel Depends --->
    *** Flash size 2M or 4M, default 2M ***
    *** USB3G is enable, please set to 0x400000 ***
(0x400000) Size of Flash
    *** Hardware setting offset,should be 4K alignment ***
(0x6000) Hardware setting offset in flash.
    *** Default setting offset,should be 4K alignment. ***
    *** size of default and current setting should be same. ***
(0x8000) Default setting offset in flash.
    *** Current setting offset,should be 4K alignment. ***
(0xC000) Current setting offset in flash.
    *** Webpage image offset,should be 4K alignment. ***
    *** size of web page is normally about 100K. ***
(0x10000) Webpages image offset in flash.
    *** Linux image offset,should be 4K alignment. ***
    *** this offset MUST between 0x10000~0x40000. ***
(0x30000) linux image offset in flash.
    *** Root image offset,should be 64K alignment. ***
    *** USB3G is enable, please set to 0x130000. ***
(0x130000) root image offset in flash.
(2) Kernel Stack Size Order Configuration

```

The usbfs filesystem for USB devices is traditionally mounted at /proc/bus/usb. It provides the /proc/bus/usb/devices file, as well as the /proc/bus/usb/BBB/DDD files. In linux-3.10 the usbfs

filesystem isn't used at all. Instead USB device nodes are created under /dev/usb/ or someplace similar. The "devices" file is available in debugfs, typically as /sys/kernel/debug/usb/devices.

Please select Debug Filesystem to support debugfs.

make linux_menuconfig

Kernel hacking --->

[*] Debug Filesystem

```
Kernel hacking
elects submenus --->. Highlighted letters are hotkeys. Pressing <Y> includes, <N> exclude
r Search. Legend: [*] built-in [ ] excluded <MD> module <> module capable

[ ] Show timing information on printks
(4) Default message log level (1-7)
[*] Enable __deprecated logic
[*] Enable __must_check logic
(1024) Warn for stack frames larger than (needs gcc 4.4)
[ ] Magic SysRq key
[*] Strip assembler-generated symbols during link
[ ] Generate readable assembler code
[ ] Enable unused/obsolete exported symbols
[!] Debug Filesystem
[ ] Run 'make headers_check' when building vmlinux
[ ] Enable full Section mismatch analysis
-- Kernel debugging
[ ] Debug shared IRQ handlers
[ ] Detect Hard and Soft Lockups
[ ] Panic on Oops
[ ] Detect Hung Tasks
[*] Collect scheduler debugging info
[ ] Collect scheduler statistics
[ ] Collect kernel timers statistics
[ ] Debug object operations
```

mount debugfs when init system:

```
mount -t sysfs none /sys
```

```
mount -t debugfs none_debug /sys/kernel/debug
```

4.4.3 User config

1) User space programs

```
--- USB3G support
[ ] comgt
[*] chat
[*] libusb
[*] usb-modeswitch
[*] mbpk_eject
[*] usb-modeswitch-data
[*] usutils
[*] hub-ctrl
[*] mnet
      Libraries
```

2) Busybox command

Linux System Utilities --->

```
[*] mdev
[*] Support /etc/mdev.conf
[*] Support subdirs/symlinks
[*] Support regular expressions substitutions when renaming device
[*] Support command execution at device addition/removal
[ ] Support loading of firmwares
```

4.4.4 Add new USB 3G dongle support

4.4.4.1 Modify mnet

While usb dongle has been inserted and attached as modem device, kernel appear several ttyUSB* in the /dev folder. The name of ttyUSB will append a number, for example: ttyUSB0. But the number doesn't always start from 0, mnet will find a proper one, and generate a ppp option file in the path /var/usb3g.option. Below is an example entry. In the most case, you doesn't need to change it.

```
-detach
noauth
/dev/ttyUSB0

115200
debug
defaultroute
ipcp-accept-local
ipcp-accept-remote
usepeerdns
crtsccts
lock
noccp
connect '/bin/chat -v -t10 -f /var/usb3g.chat'
```

Fig. Example of ppp option

```
ABORT 'NO DIAL TONE'
ABORT 'NO ANSWER'
ABORT 'NO CARRIER'
ABORT DELAYED
ABORT 'COMMAND NOT SUPPORT'

...
''' 'ATZ'

SAY 'show device infomation...\n'
'OK' 'ATI'

SAY 'show SIM CIMI...\n'
'OK' 'AT+CIMI'

SAY 'set APN...\n'
'OK' 'AT+CGDCONT=1,"IP","internet"'

SAY 'dial...\n\n'
'OK' 'ATD*99#'
'CONNECT' ''
```

Fig. Example of ppp chat

In some cases, for example: the dongle has special AT command, you may need to change the function void gen_ppp_option (char *ttyif) to generate /var/usb3g.option and /var/usb3g.chat for it.

4.4.4.2 Add new usb_modeswitch config file

Generally, mnet will detect the usb dongles' type and find a matched config for usb_modeswitch when inserted (i.e. hotplug occurred).

If an unsupported dongle is found, you can type command “lsusb” under the console to identify the vendor id and product id, and get the support from http://www.draisberghof.de/usb_modeswitch/#download.

```
# lsusb
lsusb: cannot open "/etc/usb.ids", No such file or directory
Bus 001 Device 001: ID 1d6b:0002
#
#
# cd /etc/usb_modeswitch.d/
# ls 12d1:1446
12d1:1446
#
```

Fig. find matched usb_modeswitch config

4.4.4.3 Modify option.c

If there is no supported config file, you need manually to add the vendor id and product id into linux-2.6.30/drivers/usb/serial/option.c (or linux-3.10/drivers/usb/serial/option.c) and also implement a suitable eject utility (ex: usb_modeswitch we used, it's kind of eject utility) for this dongle.

4.4.4.4 Add custom eject function

If there is a dongle doesn't support by usb_modeswitch, you can hook it into the follow of original usb_modeswitch, by add your own eject function in the structure, and add corresponding config.

```
struct custom_mode_db_s{
    int idx;
    char *modeName;
    void (*switch_func)(void);
};

enum {
    QISDA_MODE = 1,
    MBPK_MODE = 2,
};

struct custom_mode_db_s custom_mode_db[] = {
    { QISDA_MODE, "qisda", switchQisdaMode },
    { MBPK_MODE, "mobilepeak", ejectMobilepeakCDROM },
};

#####
# newer modems

DefaultVendor= 0x1da5
DefaultProduct=0xf000

TargetVendor= 0x1da5
TargetProduct= 0x4512

CustomMode="qisda"
CheckSuccess=20
```

4.4.5 USB 3G Connection

4.4.5.1 Manual connect/disconnect

If you are not using unlimited data plan, you can use manual dial. Please open the management web page [TCP/IP Setting] → [WAN Interface], and change “connection type” to “Manual” then

reboot, after that you can connect or disconnect on your will.

The screenshot shows the 'WLAN Access Point' management interface. On the left is a navigation tree with the following structure:

- Site contents:
 - Setup Wizard
 - Operation Mode
 - Wireless
 - TCP/IP Settings
 - LAN Interface
 - WAN Interface
 - Firewall
 - QoS
 - Route Setup
 - Management
 - Status
 - Statistics
 - DDNS
 - Time Zone Setting
 - Denial-of-Service
 - Log
 - Upgrade Firmware
 - Save/Reload Setting
 - Password

4.4.5.2 Connection status

If you want to know the running status of the mnet, you can get it via management web page [Management] → [Status], looking for the field “Attain IP Protocol” under the “WAN Configuration”.

We provide 5 statuses as follows:

- ① B3G Removed: there is no usb dongle on the DUT.
- ② 3G Modem Initializing...: the usb dongle has been inserted, and mnet is starting eject process.
- ③ 3G Dialing...: ejected and attached as modem device. Trying to establish a PPP connection.
- ④ 3G Connected: PPP connection has been established.
- ⑤ 3G Disconnected: PPP connection hangs up.

4.4.6 USB 3G data card list supported

At present, the USB 3G data cards are verified to be supported, as follows:

1. HUAWEI EC189 (CDMA)
2. HUAWEI E169U (WCDMA)
3. QISDA H21 (WCDMA)
4. ZTE MF637U (WCDMA)
5. ZTE AC2726 (WCDMA)
6. MOBILEPEAK Sample card (WCDMA)
7. HUAWEI K4505 (WCDMA)
8. Amoi H01 (WCDMA)
9. Vibo D200 (WCDMA)

4.4.7 The flash & SDRAM footprint for RTL8198 + RTL8192C

Enable USB 3G will lead to 226KBytes increase in flash footprint, and decrease 600Kbytes in SDRAM footprint.

RTL8198 + RTL8192C demo board v630 run SDK v2.5 image (2011/06/10), the test result as follows:

Configure \ Test entry	fw.bin (KB)	MemFree (KB)
Default	1962	16420
Enable USB 3G	2188	15820

4.5 WAPI support

4.5.1 Enable WAPI support

make linux_menuconfig // To configure linux kernel settings

Menuconfig:

Device Drivers --->

Network device support --->

Wireless LAN --->

WAPI Support // selected if WAPI support

Support local AS // selected if WAPI-CERT local AS support

Note:

1) For wireless AP Mode, WAPI support two encrypt: WAPI-PSK and WAPI-CERT, while WAPI-CERT support local AS and remote AS. If [WAPI Support] is selected and [Support local AS] is not selected, WAPI-PSK and WAPI-CERT remote AS are supported. If both [WAPI Support] and [Support local AS] are selected, WAPI-PSK, WAPI-CERT remote AS and WAPI-CERT local AS are supported.

2) For wireless Client Mode, WAPI only support WAPI-PSK.

4.5.2 Flash and SDRAM size for WAPI support

Flash and SDRAM size for WAPI support please refer to the table as follows, which is tested for RTL8198+92C (2010-11-19).

	Flash size (total)	Estimate SDRAM size for WAPI	Run-time free memory
Disable WAPI support	Kernel: linux.bin 815122. User: root.bin 1007634. Total: fw.bin 1933934.	None.	1) No security: 16136KB.
Enable WAPI support, but disable local AS	Kernel: linux.bin 831506. User: root.bin 1036306. Total:	Kernel: wapiCrypto.o 28044, wapi_wai.o 21044. User: WAPI certs related 64K.	1) No security: 15708KB. 2) WAPI-PSK: 15588KB. 3) WAPI-CERT(remote AS):

	fw.bin 1981512.	Total: 114624Byte.	15516KB.
Enable WAPI support and enable local AS	<p>Kernel: linux.bin 831506.</p> <p>User: root.bin 1630226.</p> <p>Total: fw.bin 2580110.</p>	<p>Kernel: wapiCrypto.o 28044,</p> <p>wapi_wai.o 21044.</p> <p>User: libcrypto.so.0.9.8 1485928,</p> <p>libssl.so.0.9.8 278900,</p> <p>openssl 402724,</p> <p>ecdsatest 20776,</p> <p>openssl.cnf 9675,</p> <p>readFileSize 4708,</p> <p>loadWapiFiles 10956,</p> <p>storeWapiFiles 17320,</p> <p>aeUdpClient 41136,</p> <p>aseUdpServer 31544,</p> <p>genUserCert.sh 2808,</p> <p>initCAFfiles.sh 1917,</p> <p>revokeUserCert.sh 1063,</p> <p>WAPI certs related 64K.</p> <p>Total: 2424079Byte.</p>	<p>1) No security: 15500KB.</p> <p>2) WAPI-PSK: 15440KB.</p> <p>3) WAPI-CERT(remote AS): 15312KB.</p> <p>4) WAPI-CERT(local AS): 13316KB.</p>

4.6 Mesh support

4.6.1 How to setup Mesh Network

To setup mesh network, choose AP+MESH or MESH mode on *Basic Settings* page shown as bellow figure.

WLAN Access Point

Wireless Basic Settings -wlan2

This page is used to configure the parameters for wireless LAN clients which may connect to your Access Point. You can change the network parameters.

Disable Wireless LAN Interface

Band: 2.4 GHz (B+G+N) ▾

Mode: AP+MESH ▾ Multiple AP

Network Type: Infrastructure ▾

SSID: RTK map 2g.5 Add to Profile

Channel Width: 40MHz ▾

Control Sideband: Upper ▾

Channel Number: 11 ▾

Figure 1

Choose Channel Width, Control Sideband, and a Channel Number.

Click the menu item, *Mesh settings*, of home page to expand *Mesh settings* page shown as bellow figure. This page is greyed out by default. Make sure to correctly select wireless operation mode as AP+MESH or MESH mode on *Basic Settings* to unlock this page.

Type in a Mesh ID and select a security suite.

WLAN Access Point

Wireless Mesh Network Setting

Mesh network uses wireless media to communicate with other APs, like the Ethernet does. To do this, under AP+MESH/MESH mode.

Enable Mesh

Mesh ID: RTK-mesh0
Encryption: WPA2 (AES) ▾
Pre-Shared Key Format: Passphrase ▾
Pre-Shared Key:

Apply Changes

Reset

Set Access Control

Show Advanced Information

Figure 2, Page of setup mesh network by band

After configuration is completed, please apply changes and reboot the device.

All the mesh nodes should be configured as **same band, Channel Width, Control Sideband, Channel Number, Mesh ID, and security suite**. If the configuration is correct, mesh network will construct automatically.

As Figure 2 demonstrate, there is a mesh network enabled with Mesh ID,*RTK-mesh0*, and take WPS2/AES as security suite.

The security settings are independent to WiFi applied. Be sure of all the mesh nodes use the exactly pre-shared key or the mesh nodes will be connected but data is unreachable.

MESH mode is for users who need to setup a device only capable of mesh network connectivity but without Wireless Access Point function, while AP+MESH mode has both mesh network connectivity and Wireless Access Point function.

4.6.2 Mesh Network Information

Users click the button named *Show Advanced Information* on web page as Figure 2, the browser tab is flushed by another web page as Figure 3. Users could read all mesh information through scroll bar.

WLAN Access Point

Wireless Mesh Network Information

These information is only for more technically advanced users who have a sufficient knowledge about wireless mesh network

Neighbor Table

MAC Address	Mode	Tx Packets	Rx Packets	Tx Rate (Mbps)	RSSI	Expired Time (s)
006d6573681d	11a	1430	2413	144.5	28	15
006d6573684d	11a	298	1338	130	61	15

Routing Table

Destination Mesh Point	Next-hop Mesh Point	Portal Enable	Metric	Hop Count
My-self	---	yes	---	---
006d6573681d	006d6573681d	yes	158	1
006d6573684d	006d6573684d	yes	76	1
006d6573683d	006d6573684d	yes	197	2

Portal Table

PortalMAC
006d6573682d
006d6573684d
006d6573681d
006d6573683d

Station List

MAC Address	Mode	Tx Packet	Rx Packet	Tx Rate (Mbps)	Power Saving	Expired Time (s)
None	---	---	---	---	---	---

Proxy Table

Owner	Client
006d6573682d	0018e708f6af
006d6573682d	00235a47f868
006d6573682d	006d6573682c
006d6573681d	00e04c8196c1

Refresh

Figure 3, Web Page of Mesh Information

4.6.2.1 Neighbor Table

This table shows basic information of all mesh nodes connected. Any mesh nodes configured with the same settings exceed proper distance, or behind some block that extremely degrade signal strength, or limited by Access Control List will not be present.

4.6.2.2 Routing Table

All mesh nodes discovered a routing path to reach are showed as this table. If some mesh node broadcast that it supports to be a *Portal*, the *Portal Enable* attribute recorded as “ recorded as “yes”. The *Metric* attributes explain cost of reaching this mesh node and *Hop Count* describe the path distance.

For example in Figure 3, *006d6573681d* and *006d6573684d* are 1-hop reachable. Path to reach *006d6573681d* is better than *006d6573684d* because of lower metric. *006d6573683d* is 2-hop reachable, and any packets with destination relative to *006d6573683d* will be forwarded to the Next-hop Mesh Node, *006d6573684d*.

4.6.2.3 Portable Table

All mesh nodes broadcast that it supports to be a Portal, which means they are capable to forward traffic to PC plugged in its Ethernet port.

4.6.2.4 Station List

Basic information of all WiFi clients connected to Access Point interface are listed.

4.6.2.5 Proxy Table

Data of this table is constructed through traffic learning. The learning happens to all packets received with mesh network formatted, or which are forwarded from its own Ethernet LAN/WiFi.

4.6.2.6 Refresh

Flush this page for latest information by click this button.

4.6.3 Layer-II Filtering in Mesh Network

If there is requirement to limit traffic go-through some link in mesh network, users could access web page as Figure 4 below. Click the button named *Set Access Control* at page Figure 2 shows will display this page.

There are three policies for limit traffic. **Disable** means turn off this functionality. All rules stored will not be cleaned up unless users demand. **Allow** makes packets destination of MAC(s) in rule(s) pass through, or **Deny** drop packets fit rules.

Rules append could be removed through clicking two buttons, *Delete Selected* or *Delete All*. Delete Selected will delete those rules users checked at check box. Delete All will remove all entries.

Reset will clean up all changes ever done, including MAC address just input and policy just choose.

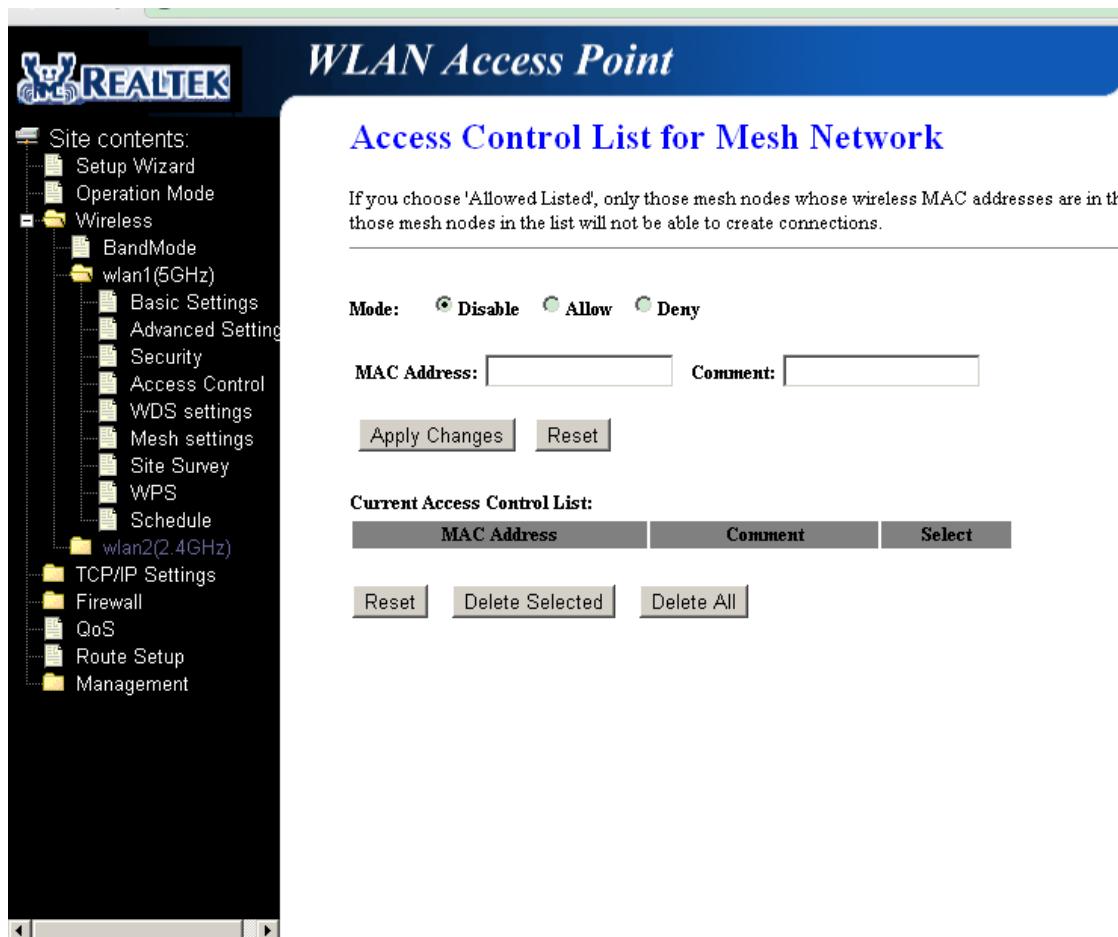


Figure 4, Access Control List for Mesh

This L-II filtering is independent to *Access Control* for WiFi.

4.6.4 Constraint of Mesh Network Application

- 1). Because Mesh Network will be L2 connected to form a LAN, please make sure that there is only one DHCP server/Gateway in one mesh network. Or multiple DHCP servers will deliver IP to clients within the same LAN and suffer IP confliction.
- 2). Although each mesh node is capable to connect 15 other mesh nodes, it is suggested to deploy 5 mesh nodes at most within an area such as at home. Or too much media access contention will drop throughput dramatically. In general, it is more than enough to deploy 3 nodes.
- 3). Universal Repeater Mode can be enabled in AP+MESH and MESH mode. However only one mesh node in the mesh network can have this function enabled. Multiple repeaters may cause the channel of mesh to fluctuate.

4.6.5 How to enable Mesh using SDK

To enable mesh network support, it is necessary to configure kernel and user-space application of SDK through menuconfig, as following demonstrated.

- First, configure the kernel.

Select “Config kernel”

```
--- select components
Selected Target (rt18196e) --->
Selected Kernel (linux-2.6.30) --->
Selected Busybox (busybox-1.13) --->
Selected toolchain (rsdk-1.3.6-4181-EB-2.6.30-0.9.30) --->
--- rt18196e
Selected Board Configuration (96E+92E GW) --->
--- config components
[ ] Config kernel
[ ] Config users
[ ] Config busybox
[ ] Load default settings
[ ] Save default settings
```

, and then exist.

Select option of “Device Drivers”.

```
System Configuration --->
Kernel type --->
General setup --->
[ ] Enable loadable module support --->
[*] Enable the block layer --->
Bus options (PCI/USB) --->
Power management options --->
[*] Networking support --->
[ ] Device Drivers --->
File systems --->
Kernel hacking --->
Security options --->
[*] Cryptographic API --->
Library routines --->
```

And choose to configure “Network device support”.

```
Generic Driver Options --->
[ ] Connector - unified userspace <-> kernelspace linker --->
[*] Memory Technology Device (MTD) support --->
[ ] Parallel port support --->
[*] Block devices --->
[ ] Misc devices --->
    SCSI device support --->
[ ] Serial ATA (prod) and Parallel ATA (experimental) drivers --->
[ ] Multiple devices driver support (RAID and LVM) --->
[*] Network device support --->
[ ] ISDN support --->
[ ] Telephony support --->
    Input device support --->
    ...
```

After choose “Wireless LAN”,

```
[ ] MAC-VLAN support (EXPERIMENTAL)
[ ] EQL (serial line load balancing) support
[ ] Universal TUN/TAP device driver support
[ ] Virtual ethernet pair device
[ ] Ethernet (10 or 100Mbit) --->
[ ] Ethernet (1000 Mbit) --->
[ ] Ethernet (10000 Mbit) --->
[*] Wireless LAN --->
    *** Enable WiMAX (Networking options) to see the WiMAX dri
[ ] Wan interfaces support --->
[*] PPP (point-to-point protocol) support
[ ]     PPP multilink support (EXPERIMENTAL)
[ ]     PPP filtering
[*]     PPP support for async serial ports
[*]     DOD support for sun3 ttu ports
```

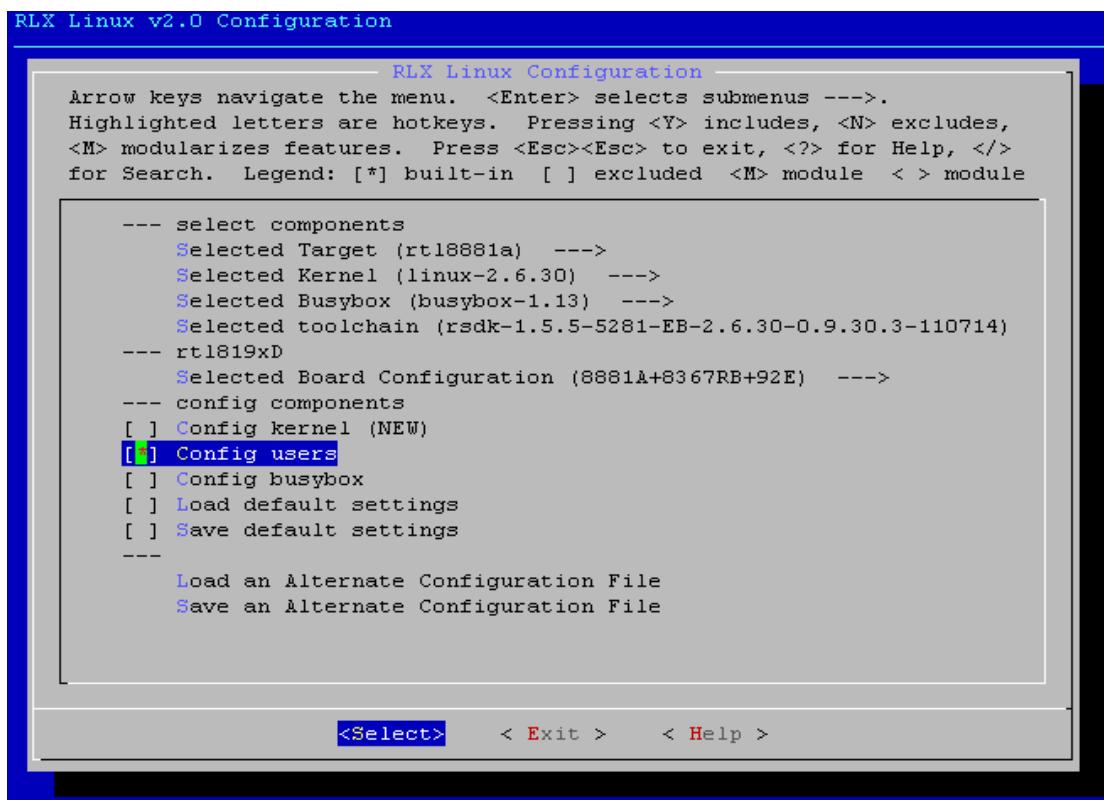
You will see the page as below.

```
[ ] Wireless LAN (pre-802.11)
[ ] Wireless LAN (IEEE 802.11)
[ ] Realtek 8190 wireless support
[ ] Realtek 8192SE wireless support
[*] RTL8192C/D 802.11b/g/n support
[*] Use PCIe slot 0 WiFi device
    PCIe Slot 0 device (Realtek 8188E wireless support)
[ ] PCIe slot 0 Enable external high power PA
[*] Private skb buffer management
[*] Virtual AP Support
[*] Client Mode Support
[*]     Repeater Mode support
[ ] Client Mode 802.1x Support
[*]     Multiple AP profile Support
[*] WDS Support
[ ] WAPI Support
[ ] Config File support
[ ] Wireless Tools v29 support
[*] Realtek wps2.0 support
[ ] Band Edge Limit support for 92C/92D/8812/88E/92E/8881A
[*] RTL Mesh Support
```

Select the option “RTL Mesh Support” then save and exit to store the configuration.

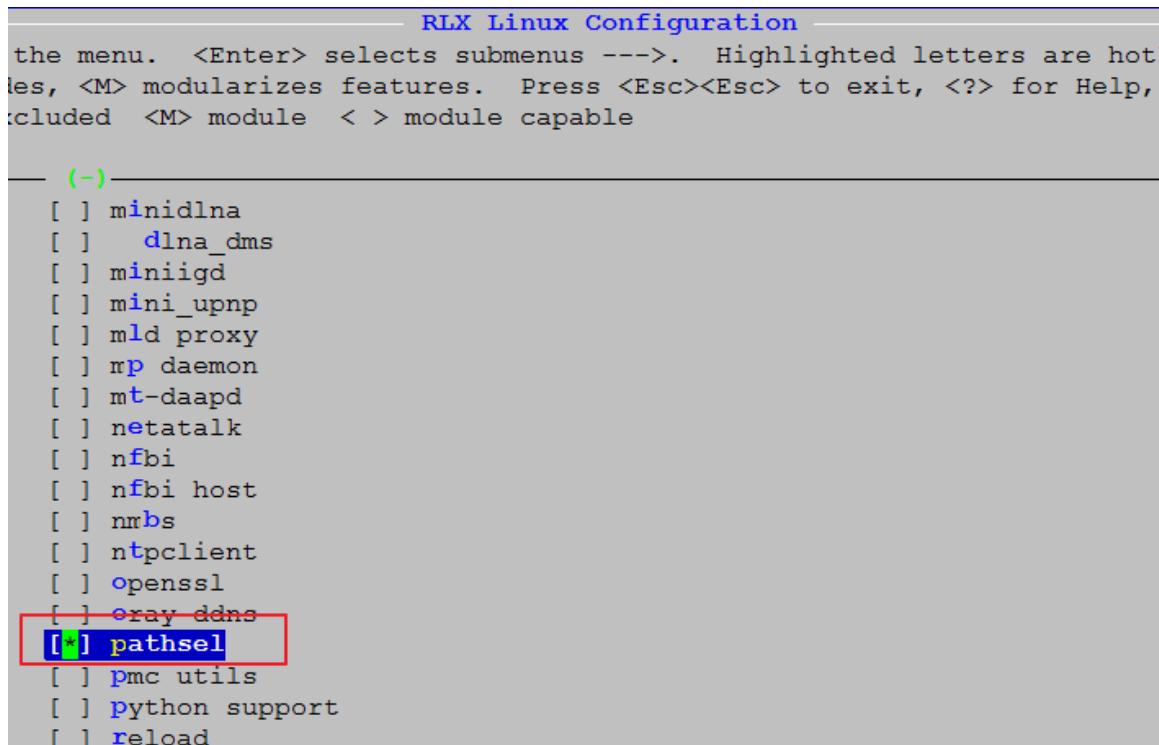
- Second, turn on “pathsel” application as well, which is under “Config users” option.

Select “Config users” and then exist.



It will bring up “Applications” submenu.

Enable pathsel application.



Save and exit to store the configuration, and then compile SDK to generate mesh image.

4.7 VLAN support

4.7.1 RTK VLAN

4.7.1.1. What's RTK VLAN?

RTK Vlan is an enhanced vlan feature based on 802.1Q, and it is processed by software. In RTK vlan, user can per-port configures the following features:

- Disable/enable RTK vlan feature
- vlan tag/ untag when packet tx from the port
- different/same vlan id for ports
- priority for every port
- cfi for every port
- if wan is vlan tagged, the tag vlan id is the source vid

4.7.1.2. How to implement in Linux

For example, rtk vlan is configured as following diagram.

Enable	Ethernet/Wireless	WAN/LAN	Tag	VID(1~4090)	Priority	CPI
<input checked="" type="checkbox"/>	Ethernet Port1	LAN	<input checked="" type="checkbox"/>	4090	7	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	Ethernet Port2	LAN	<input checked="" type="checkbox"/>	4090	0	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	Ethernet Port3	LAN	<input checked="" type="checkbox"/>	500	3	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Ethernet Port4	LAN	<input type="checkbox"/>	1	0	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Wireless Primary AP	LAN	<input type="checkbox"/>	1	0	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Virtual AP1	LAN	<input type="checkbox"/>	1	0	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Virtual AP2	LAN	<input type="checkbox"/>	1	0	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Virtual AP3	LAN	<input type="checkbox"/>	1	0	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Virtual AP4	LAN	<input type="checkbox"/>	1	0	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	Ethernet Port5	WAN	<input checked="" type="checkbox"/>	1	0	<input checked="" type="checkbox"/>

Both port1 and port2 are the member port of vlan 4090. It means that port1 and port2 can communicate with each other, port1 and port 3 can NOT communicate with each other.

The filed tag means that the ports only received vlan tagged packets whose vid is 4090 and should transmit with vlan tagged(vid is 4090).

If filed tag is disable, both vlan tagged (vid is 4090) and vlan untagged can be received by the port.

If packet received from port1 and with vlan tagged (vid is 4090) will be transmitted to wan through wan port, packet should be vlan tagged (vid is 4090) when wan port is vlan tag enable and packet should be vlan untagged when wan port is vlan tag disable.

4.7.1.3. How to enable/disable RTK vlan feature

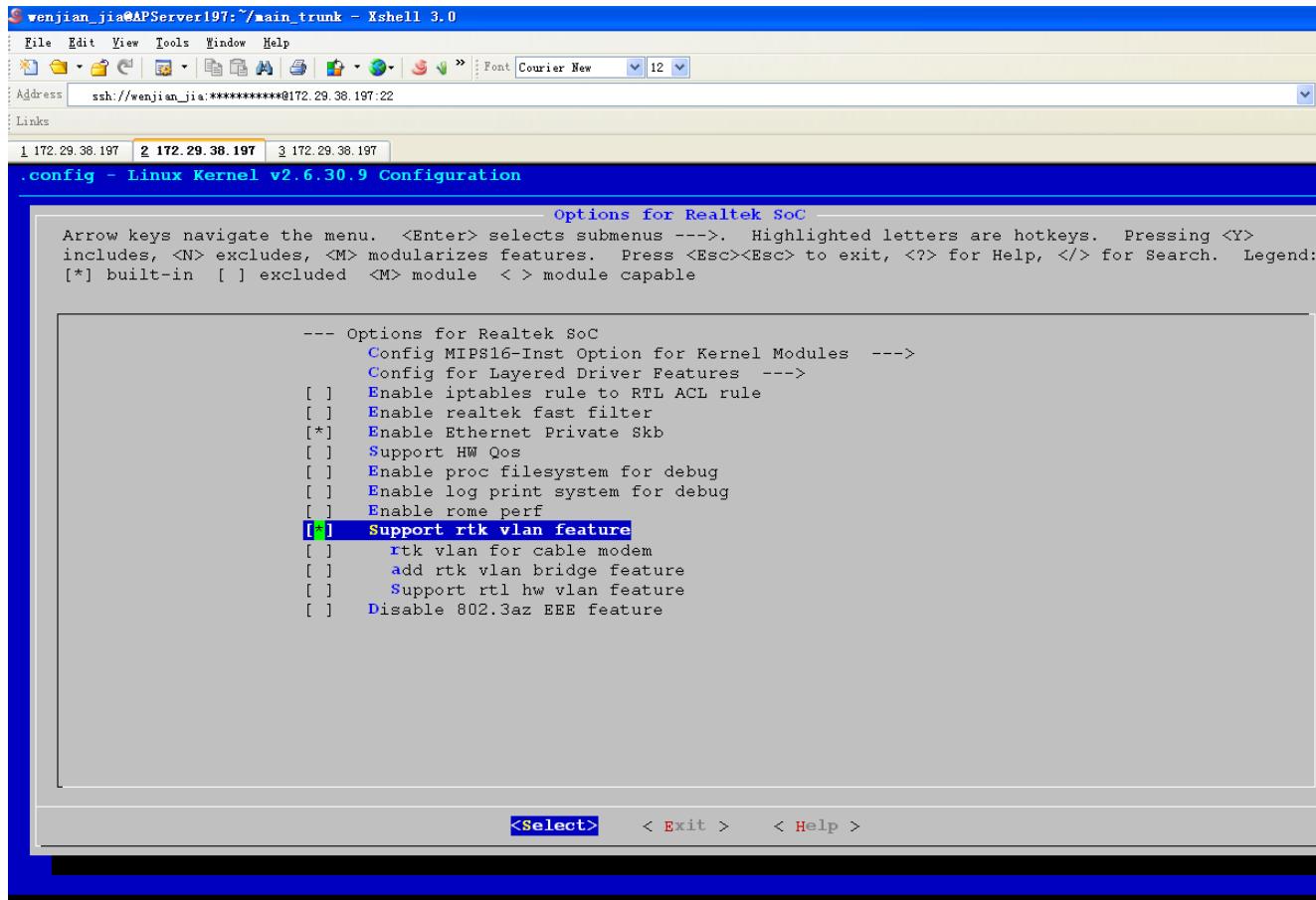
1. Menuconfig:

Device Drivers --->

[*] Network device support --->

[*] Options for Realtek SoC --->

[*] Support rtk vlan feature



2. RTK vlan feature

User can use command echo "0">>/proc/rtk_vlan_support to disable rtk vlan feature, and use command echo "1">>/proc/rtk_vlan_support to enable rtk_vlan_feature.

4.7.2 RTK VLAN BRIDGE FEATURE

Some customers require AP support VOIP, IPTV and so on. So we add a feature to enable lan port can bridge with wan port to forward VOIP and IPTV stream based on RTK VLAN support. Then it is still use CPU software to process vlan.

4.7.2.1 How to enable/disable RTK vlan feature

1. Menuconfig:

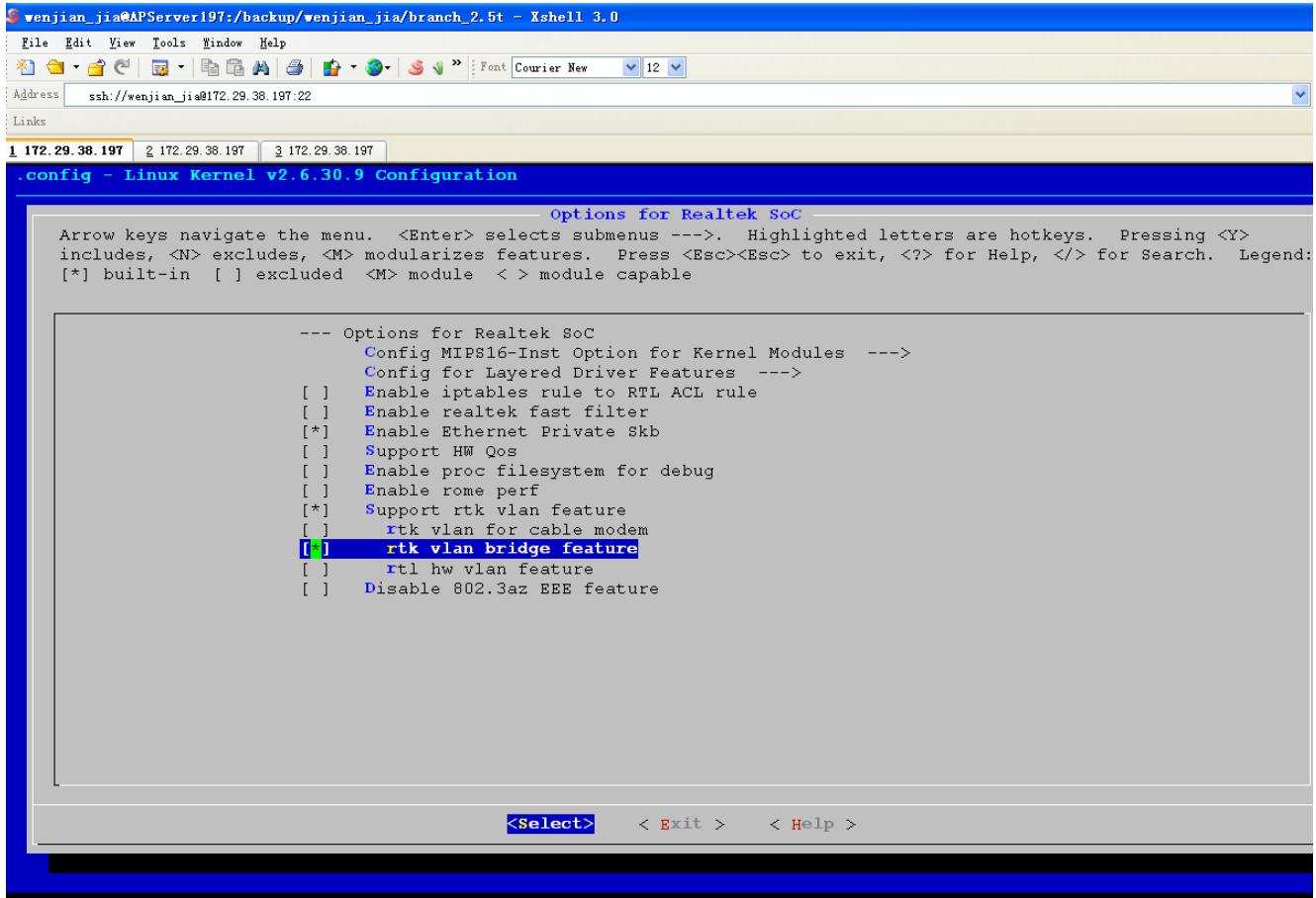
Device Drivers --->

[*] Network device support --->

[*] Options for Realtek SoC --->

[*] Support rtk vlan feature

[*] rtk_vlan_bridge feature



wenjian_jia@APServer197:/backup/wenjian_jia/branch_2.5t - Xshell 3.0

File Edit View Tools Window Help

Address ssh://wenjian_jia@172.29.38.197:22

Links

1 172.29.38.197 2 172.29.38.197 3 172.29.38.197

.config - Linux Kernel v2.6.30.9 Configuration

Options for Realtek SoC

Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [] excluded <M> module < > module capable

```
--- Options for Realtek SoC
      Config MIP816-Inst Option for Kernel Modules --->
      Config for Layered Driver Features --->
[ ]   Enable iptables rule to RTL ACL rule
[ ]   Enable realtek fast filter
[*]  Enable Ethernet Private Skb
[ ]   Support HW Qos
[ ]   Enable proc filesystem for debug
[ ]   Enable rome perf
[*]  Support rtk vlan feature
[ ]     rtk vlan for cable modem
[ ]     rtk_vlan_bridge feature
[ ]     rtl hw vlan feature
[ ]     Disable 802.3az EEE feature
```

<Select> < Exit > < Help >

2. RTK vlan bridge feature

User can use command echo “0”>/proc/rtk_vlan_support for disable rtk vlan feature, and use command echo “1”>/proc/rtk_vlan_support for enable rtk_vlan_feature.

4.7.2.2 How to implement in Linux

For example, If we want port4 to bridge with wan port, then we can config port as bridge forwarding rule, just as follows:

VLAN Settings

Entries in below table are used to config vlan settings. VLANs are created to provide the segmentation services traditionally provided by routers. VLANs address issues such as scalability, security, and network management.

Enable VLAN

Enable	Ethernet/Wireless	WAN/LAN	Forwarding Rule	Tag	VID (1~4090)	Priority	CFI
<input checked="" type="checkbox"/>	Ethernet Port1	LAN	NAT	<input checked="" type="checkbox"/>	200	0	
<input checked="" type="checkbox"/>	Ethernet Port2	LAN	NAT	<input checked="" type="checkbox"/>	200	0	
<input checked="" type="checkbox"/>	Ethernet Port3	LAN	NAT	<input checked="" type="checkbox"/>	200	0	
<input checked="" type="checkbox"/>	Ethernet Port4	LAN	Bridge	<input checked="" type="checkbox"/>	100	0	
<input type="checkbox"/>	Wireless 1 Primary AP	LAN	NAT	<input type="checkbox"/>	1	0	
<input type="checkbox"/>	Wireless 1 Virtual AP1	LAN	NAT	<input type="checkbox"/>	1	0	
<input type="checkbox"/>	Wireless 1 Virtual AP2	LAN	NAT	<input type="checkbox"/>	1	0	
<input type="checkbox"/>	Wireless 1 Virtual AP3	LAN	NAT	<input type="checkbox"/>	1	0	
<input type="checkbox"/>	Wireless 1 Virtual AP4	LAN	NAT	<input type="checkbox"/>	1	0	
<input checked="" type="checkbox"/>	Ethernet Port5	WAN	NAT	<input type="checkbox"/>	1	0	

Config as this web-page, port4 will be bridge with wan, and port0-3 will be nat with wan.

And Ethernet Port4 “Tag” is enabled; the packets from port4 to wan port will be tagged with vid

100. That means if you want the packets from lan port to wan port to be tagged, you just need to enable “Tag” on lan port.

Besides, if you want the packets generated by AP, such as the packets sent by pppoe from AP also tagged, then you need to enable “Tag” on Wan port.

Enable	Ethernet/Wireless	WAN/LAN	Forwarding Rule	Tag	VID (1~4090)	Priority	CFI
<input checked="" type="checkbox"/>	Ethernet Port1	LAN	NAT	<input checked="" type="checkbox"/>	200	0	<input type="checkbox"/>
<input checked="" type="checkbox"/>	Ethernet Port2	LAN	NAT	<input checked="" type="checkbox"/>	200	0	<input type="checkbox"/>
<input checked="" type="checkbox"/>	Ethernet Port3	LAN	NAT	<input checked="" type="checkbox"/>	200	0	<input type="checkbox"/>
<input checked="" type="checkbox"/>	Ethernet Port4	LAN	Bridge	<input checked="" type="checkbox"/>	100	0	<input type="checkbox"/>
<input type="checkbox"/>	Wireless 1 Primary AP	LAN	NAT	<input type="checkbox"/>	1	0	<input type="checkbox"/>
<input type="checkbox"/>	Wireless 1 Virtual AP1	LAN	NAT	<input type="checkbox"/>	1	0	<input type="checkbox"/>
<input type="checkbox"/>	Wireless 1 Virtual AP2	LAN	NAT	<input type="checkbox"/>	1	0	<input type="checkbox"/>
<input type="checkbox"/>	Wireless 1 Virtual AP3	LAN	NAT	<input type="checkbox"/>	1	0	<input type="checkbox"/>
<input type="checkbox"/>	Wireless 1 Virtual AP4	LAN	NAT	<input type="checkbox"/>	1	0	<input type="checkbox"/>
<input checked="" type="checkbox"/>	Ethernet Port5	WAN	NAT	<input checked="" type="checkbox"/>	1	0	<input type="checkbox"/>

[Apply Changes](#) [Reset](#)

Then pppoe of DUT will send packets tagged with vid = 1 to dial up.

4.7.3 HW VLAN FEATURE

HW vlan feature also support forwarding rule as bridge and nat between lan port and wan port, it uses hardware ASIC to process vlan. Only rtl8196ct and rtl8198t support hw vlan feature.

4.7.3.1 How to enable/disable RTK vlan feature

1. Menuconfig:

Device Drivers --->

[*] Network device support --->

[*] Options for Realtek SoC --->

[*] Support rtk vlan feature

[*] Support rtl hw vlan feature

```

wenjian_jia@APServer197:~/main_trunk - Xshell 3.0
File Edit View Tools Window Help
Address ssh://wenjian_jia:*****@172.29.38.197:22
Links
1 172.29.38.197 2 172.29.38.197 3 172.29.38.197
.config - Linux Kernel v2.6.30.9 Configuration

      Options for Realtek SoC
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys. Pressing <Y>
includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend:
[*] built-in [ ] excluded <M> module < > module capable

--- Options for Realtek SoC
      Config MIPS16-Inst Option for Kernel Modules --->
      Config for Layered Driver Features --->
[ ]   Enable iptables rule to RTL ACL rule
[ ]   Enable realtek fast filter
[*]  Enable Ethernet Private Skb
[ ]   Support HW Qos
[ ]   Enable proc filesystem for debug
[ ]   Enable log print system for debug
[ ]   Enable rome perf
[*]  Support rtk vlan feature
[ ]     rtk vlan for cable modem
[ ]     add rtk vlan bridge feature
[*]  Support rtl hw vlan feature
[ ]   Disable 802.3az EEE feature

<select>  < Exit >  < Help >

```

2. RTL HW vlan feature

As rtl HW vlan feature has almost the same function with rtk vlan & rtk vlan bridge feature. The difference is rtl HW vlan using hardware ASIC to process vlan, and rtk vlan bridge feature use CPU.

Besides, only rtl8196ct & rtl8198t support rtl HW vlan feature. So if you want to use HW vlan, just select “Support rtk vlan feature” & “Support rtl HW vlan feature” in menuconfig.

4.7.3.2 How to implement in Linux

The default forwarding rule for lan ports is nat, so if you want all lan port just do nat with wan port, then you need not to configure lan port on web any more. Anyway, if you want to configure a lan port, for example port0, to bridge with wan, then you can configure web as follow:

VLAN Settings

Entries in below table are used to config vlan settings. VLANs are created to provide the segmentation services traditionally provided by routers. VLANs address issues such as scalability, security, and network management.

Enable VLAN

Enable	Ethernet/Wireless	WAN/LAN	Forwarding Rule	Tag	VID (1~4096)	Priority	CFI
<input checked="" type="checkbox"/>	Ethernet Port1	LAN	Bridge	<input checked="" type="checkbox"/>	3022	2	<input type="checkbox"/>
<input type="checkbox"/>	Ethernet Port2	LAN	NAT	<input type="checkbox"/>	3030	7	<input type="checkbox"/>
<input type="checkbox"/>	Ethernet Port3	LAN	NAT	<input type="checkbox"/>	500	0	<input type="checkbox"/>
<input type="checkbox"/>	Ethernet Port4	LAN	NAT	<input type="checkbox"/>	1	3	<input type="checkbox"/>
<input type="checkbox"/>	Wireless 1 Primary AP	LAN	NAT	<input type="checkbox"/>	1	0	<input type="checkbox"/>
<input type="checkbox"/>	Wireless 1 Virtual AP1	LAN	NAT	<input type="checkbox"/>	1	0	<input type="checkbox"/>
<input type="checkbox"/>	Wireless 1 Virtual AP2	LAN	NAT	<input type="checkbox"/>	1	0	<input type="checkbox"/>
<input type="checkbox"/>	Wireless 1 Virtual AP3	LAN	NAT	<input type="checkbox"/>	1	0	<input type="checkbox"/>
<input type="checkbox"/>	Wireless 1 Virtual AP4	LAN	NAT	<input type="checkbox"/>	1	0	<input type="checkbox"/>
<input checked="" type="checkbox"/>	Ethernet Port5	WAN	NAT	<input checked="" type="checkbox"/>	8	0	<input type="checkbox"/>

Apply Changes **Reset**

Then port0 can bridge with wan port to forward IPTV & VOIP with vid = 3022. And other lan ports nat with wan, the packets from these lan port should be untag, and these packets go out to wan port will tagged with vid = 8.

4.7.4 LINUX 8021Q VLAN FEATURE

This feature is used to support kernel 802.1q vlan.

4.7.4.1 How to enable/disable linux vlan feature

1. enable 【make linux vlan support RTK AP router】 and 【802.1Q VLAN Support】 in kernel menuconfig

Location:

-> Networking support (NET [=y])

-> Networking options

[*] 802.1Q VLAN Support

[*] Make linux vlan support RTK AP router

```

[ ] The IPv6 protocol --->
[ ] Security Marking
[*] Network packet filtering framework (Netfilter) --->
[ ] The DCCP Protocol (EXPERIMENTAL) --->
[ ] The SCTP Protocol (EXPERIMENTAL) --->
[ ] The TIPC Protocol (EXPERIMENTAL) --->
[ ] Asynchronous Transfer Mode (ATM)
[*] 802.1d Ethernet Bridging
[ ] Enable RealTek STP
[ ] Enable RealTek Hardware STP
[ ] Distributed Switch Architecture support --->
[*] 802.1Q VLAN Support
[*] Make linux vlan support RTK AP router
[ ] GVRP (GARP VLAN Registration Protocol) support
[ ] DECnet Support
[ ] ANSI/IEEE 802.2 LLC type 2 Support
[ ] The IPX protocol
[ ] Appletalk protocol support
[ ] CCITT X.25 Packet Layer (EXPERIMENTAL)
[ ] LAPB Data Link Driver (EXPERIMENTAL)
[ ] Acorn Econet/AUN protocols (EXPERIMENTAL)
[ ] WAN router
[ ] Phonet protocols family
[*] QoS and/or fair queueing --->
[ ] Data Center Bridging support
[ ] Network testing --->

```

2. and **un-select** “Support rtk vlan feature” in kernel menuconfig.

Location:

```

-> Device Drivers
    -> Network device support (NETDEVICES [=y])
        -> Options for Realtek SoC (RTL_819X_SWCORE [=y])

```

```

--- Options for Realtek SoC
    Config MIPS16-Inst Option for Kernel Modules --->
    Config for Layered Driver Features --->
[ ] Enable iptables rule to RTL ACL rule
[ ] Enable realtek fast filter
[*] Enable Ethernet Private Skb
[*]   Enable Dump Ethernet Private Skb Info
[ ] Support HW Qos
[*] Enable proc filesystem for debug
[ ] Enable log print system for debug
[ ] Enable rome perf
[ ] Support rtk vlan feature
[*] Disable 802.3az EEE feature
[ ] Enable Dump Socket Buffer Info

```

3. If enable linux vlan in support hw nat/hw multicast mode please select “Enable RTL Hardware NAPT” in kernel menuconfig.

Location:

- > Device Drivers
- > Network device support (NETDEVICES [=y])
- > Options for Realtek SoC (RTL_819X_SWCORE [=y])
- > Config for Layered Driver Features
- > Hardware Features Selection (<choice> [=y])

```
(X) Enable RTL Hardware NAPT
( ) Enable RTL Hardware Multicast Only
( ) Enable RTL Hardware switch Only
```

4. enable 【 iproute2 full support】 in user menuconfig

NOTE: you can also use vconfig command to config, enable vconfig in busybox menuconfig.

```
RLX Linux Configuration
the menu. <Enter> selects submenus --->. Highlighted letters are hot
keys, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help,
excluded <M> module < > module capable
```

(-) [] curl
[] curl-7.34.0
[] discover
[] dnrd
[*] firmware write daemon
[] FT Daemon support
[] gdbserver
[] gproxy
[] hostapd 2.0
[] iapp
[] igmp proxy
[] ioh
[*] iproute2
[*] iproute2 full support
[] iptables

5. disable 【IP】 in busybox menuconfig(note: use iproute2 to config vlan instead of busybox ip tool)

```

[ ] Enable IPv6 support
[ ] Verbose resolution errors
[ ] arp
[ ] arping
[ ] brctl
[ ] dnsd
[ ] ether-wake
[ ] fakeidentd
[ ] ftpget
[ ] ftpput
[*] hostname
[ ] httpd
[*] ifconfig
[*]   Enable status reporting output (+7k)
[ ]   Enable slip-specific options "keepalive" and "outfill"
[ ]   Enable options "mem_start", "io_addr", and "irq"
[*]   Enable option "hw" (ether only)
[ ]   Set the broadcast automatically
[ ] ifenslave
[ ] ifupdown
[ ] inetd
[*] ip
[ ] ipcalc
[ ] nameif
[ ] nc
[ ] netstat

```

4.7.4.2 VLAN related proc

1. /proc/net/vlan/vlanEnable: to control linux vlan feature enable or disable

The sdk version greater than 3.4.7 support the following format:

When echo “1 0” >/proc/net/vlan/vlanEnable, just enable linux vlan, not support hw nat and hw multicast, the vlan set to be the wan vlan role will be represented as “eth1.x” in linux networking stack so your firewall rules (Routing rules ,iptables, NAPT ..etc) for internet servicet should use “eth1.x” as the wan interface.

When echo “1 1” >/proc/net/vlan/vlanEnable, enable linux vlan, support hw nat and hw multicast, the vlan set to be the wan vlan role will be represented as “eth1” in linux networking stack so your firewall rules (Routing rules ,iptables, NAPT ..etc) for internet servicet should always use “eth1” as the only wan interface. The wan interface limitation is due to rtk HW NAT and driver design.

When echo “0 0” >/proc/net/vlan/vlanEnable, disable linux vlan

The sdk version less than 3.4.7 support the following format:

```

echo 1 > /proc/net/vlan/vlanEnable, enable linux vlan
echo 0 > /proc/net/vlan/vlanEnable, disable linux vlan

```

2. /proc/net/vlan/groupIndex: to add vlan groupIndex

When add new vlan group, groupIndex should be set first. Its value should be the same as vlan group's vid you want to add.

3. /proc/net/vlan/vlanGroup: to add vlan group(indicated by vid), group member and tagged/untagged member

flag, memberMask, tagMemberMask, vid

1, 111f, 1110, 9

When flag is 1, it will add this new group. When flag is 0, the old vlan group will be deleted.

4. /proc/net/vlan/pvid : to set each member's pvid

9,9,9,9,0,0,0,0,0,0,9,9,9,9,9,9,9,9

Pvid's index mapping is the same as VLAN group's member mask.

5. the mapping relationship between "MemberMask" and "physical port or wlan interface"

Only 21 bits are used, 11 bits are reserved for future use.

```
#define ETH_P0_MASK_BIT      0
#define ETH_P1_MASK_BIT      1
#define ETH_P2_MASK_BIT      2
#define ETH_P3_MASK_BIT      3
#define ETH_P4_MASK_BIT      4
#define ETH_P5_MASK_BIT      5
#define ETH_P6_MASK_BIT      6
#define ETH_P7_MASK_BIT      7
#define ETH_P8_MASK_BIT      8
#define WLAN0_MASK_BIT       9
#define WLAN0_VA0_MASK_BIT   10
#define WLAN0_VA1_MASK_BIT   11
#define WLAN0_VA2_MASK_BIT   12
```

```
#define WLAN0_VA3_MASK_BIT    13  
#define WLAN0_VXD_MASK_BIT    14  
#define WLAN1_MASK_BIT        15  
#define WLAN1_VA0_MASK_BIT    16  
#define WLAN1_VA1_MASK_BIT    17  
#define WLAN1_VA2_MASK_BIT    18  
#define WLAN1_VA3_MASK_BIT    19  
#define WLAN1_VXD_MASK_BIT    20
```

5. /proc/net/vlan/wanVlanId

The sdk version greater than 3.4.7 support this proc.

This proc is used to specify the vlan id of the wan. If the vlan is created and one of its member port is for internet wan service (NAT), the vid need to be specified in this proc before you created the wan vlan by /proc/net/vlan/groupIndex and /proc/net/vlan/vlanGroup. The wan vid information is a must for our HWNAT related tables and driver.

Command:

```
echo "filed1" > /proc/net/vlan/wanVlanId  
field1: wan's vid (NAT role) . 1~4095 .
```

Example: Set wlan group vid=2, its member ports are port3,port4(internet service) ,CPU port. Only Cpu port is tagged . Because Port4 is NAT function involve, you have to set this vid to /proc/net/vlan/wanVlanId before using /proc/net/vlan/groupIndex and /proc/net/vlan/vlanGroup to create this wan wlan.

```
echo 2 > /proc/net/vlan/wanVlanId  
echo 2 >/proc/net/vlan/groupIndex  
echo "1,58,40,2" >/proc/net/vlan/vlanGroup
```

NOTE: This field valid only when echo “1 1” > /proc/net/vlan/vlanEnable(means:enable linux wlan, support hw nat and hw multicast), and in this mode the wlan set to be the wan wlan role will be represented as “eth1” in linux networking stack , so your firewall rules (Routing rules ,iptables, NAPT ..etc) for internet service should always use “eth1” as the only wan interface. The wan interface limitation is due to rtk HW NAT and driver design.

4.7.4.3 VLAN setup example

1. enable linux vlan **not** support hw nat and hw multicast mode example

Example A (wan type as static, wan vlan group=10 tag, ethernet lan vlan group=5 untag)

```
ip link add link eth1 name eth1.10 type vlan id 10  
ifconfig eth1.10 up  
ifconfig eth1.10 192.168.2.177  
ifconfig eth1 0.0.0.0
```

```
ip link add link eth0 name eth0.5 type vlan id 5  
ifconfig eth0.5 up
```

```
brctl delif br0 eth0  
brctl addif br0 eth0.5
```

The sdk version greater than 3.4.7 using: echo "1 0" > /proc/net/vlan/vlanEnable

The sdk version less than 3.4.7 using: echo 1 > /proc/net/vlan/vlanEnable

```
echo 5 > /proc/net/vlan/groupIndex  
echo 1,FFE0,0,5 > /proc/net/vlan/vlanGroup
```

```
echo 10 > /proc/net/vlan/groupIndex  
echo 1,110,110,10 > /proc/net/vlan/vlanGroup
```

```
echo 5,5,5,5,10,5,5,5,0,0,0,0,0,0,0,0,0,0,0 > /proc/net/vlan/pvid
```

Please check the firewall settings(The following rules only for test).

```
iptables -F -t nat  
iptables -t nat -A POSTROUTING -o eth1.10 -j MASQUERADE  
iptables -F  
iptables -P FORWARD ACCEPT  
iptables -P INPUT ACCEPT
```

With this configuration, the packets go out of the lan side without tag, those packets go out of the eth1.10 interface will tag with vlanid=10.

Example B (wan type as static, eth1 untag, eth1.10 and eth1.11 tag)

```
ip link add link eth1 name eth1.10 type vlan id 10  
ifconfig eth1.10 up
```

```
ip link add link eth1 name eth1.11 type vlan id 11  
ifconfig eth1.11 up
```

```
ifconfig eth1 192.168.2.177  
ifconfig eth1.10 192.168.3.177  
ifconfig eth1.11 192.168.4.177
```

The sdk version greater than 3.4.7 using: echo "1 0" > /proc/net/vlan/vlanEnable

The sdk version less than 3.4.7 using: echo 1 > /proc/net/vlan/vlanEnable

```
echo 10 > /proc/net/vlan/groupIndex  
echo 1,110,110,10 > /proc/net/vlan/vlanGroup
```

```
echo 11 > /proc/net/vlan/groupIndex  
echo 1,110,110,11 > /proc/net/vlan/vlanGroup
```

Please check the firewall settings(The following rules only for test).

```
iptables -F -t nat
iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE
iptables -t nat -A POSTROUTING -o eth1.10 -j MASQUERADE
iptables -t nat -A POSTROUTING -o eth1.11 -j MASQUERADE
iptables -F
iptables -P FORWARD ACCEPT
iptables -P INPUT ACCEPT
```

With this configuration, the packets go out of the eth1 interface without tag, those packets go out of the eth1.10 interface will tag with vlanid=10, and those packets go out of the eth1.11 interface will tag with vlanid=11.

2. enable linux vlan in support hw nat/hw multicast mode example

Example A (all ports(wan and lan) are untagged) We choose that ports from 0 to 3 are LAN ports, and LAN vid is 1 (for example, it can be different). This ports are untagged. port 4 — WAN port and his vid is 2. This port is untagged configure switch:

```
echo "1 1">> /proc/net/vlan/vlanEnable  
echo 2 >/proc/net/vlan/wanVlanId
```

```
echo 1 >/proc/net/vlan/groupIndex  
echo "1,4f,40,1" > /proc/net/vlan/vlanGroup  
  
echo 2 >/proc/net/vlan/groupIndex  
echo "1,50,40,2" >/proc/net/vlan/vlanGroup  
  
echo 1 1 1 1 2 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 > /proc/net/vlan/pvid
```

```
vconfig add eth0 1  
ifconfig eth0.1 up  
brctl delif br0 eth0  
brctl addif br0 eth0.1
```

WAN interface:

ifconfig eth1 up → wan vlan (internet service) always represent as eth1

Example B (we need all that shown in example A and second tagged wan with vid 555) , the tagged wan is used to management vlan application.

configure switch:

```
echo "1 1">> /proc/net/vlan/vlanEnable  
echo 2 >/proc/net/vlan/wanVlanId
```

```
echo 1 >/proc/net/vlan/groupIndex  
echo "1,4f,40,1" > /proc/net/vlan/vlanGroup  
  
echo 2 >/proc/net/vlan/groupIndex  
echo "1,50,40,2" >/proc/net/vlan/vlanGroup
```

```
vconfig add eth0 1  
ifconfig eth0.1 up  
vconfig add eth0 555  
ifconfig eth0.555 up  
brctl delif br0 eth0  
brctl addif br0 eth0.1
```

WAN interface:

ifconfig eth1 up → wan vlan (internet service) always represent as eth1

NOTE: vlanid =8 and vlanid =9 are reserved for DUT, please don't use it.

4.8 I2C UART support

4.8.1 Enable I2C UART in bootloader

4.8.1.1 Modify code for GPIO pin in bootloader

According hardware designed; please modify the GPIO pin mux which is used by I2C UART.

The GPIO pin used by I2C UART is defined in boot/serial_sc16is7x0/8250_sc16is7x0.c

For example, following configuration means I2C UART use GPIO pin D3~D6.

```
#define SC16IS7X0_RESET I2C_GPIO_ID('D', 3) // RESET = D3
```

```
#define SC16IS7X0_SCI_I2C GPIO_ID('D', 4) // SCII ≡ D4
```

```
#define SC16IS7X0_SDA I2C_GPIO_ID('D', 5) // SDA = D5
```

```
#define SC16IS7X0_IRO I2C_GPIO_ID('D', 6) // IRO = D6
```

1.2 bootloader menuconfig

1.2 bootloader menuconfig

Choose the correct configuration related to I2C UART in below table.

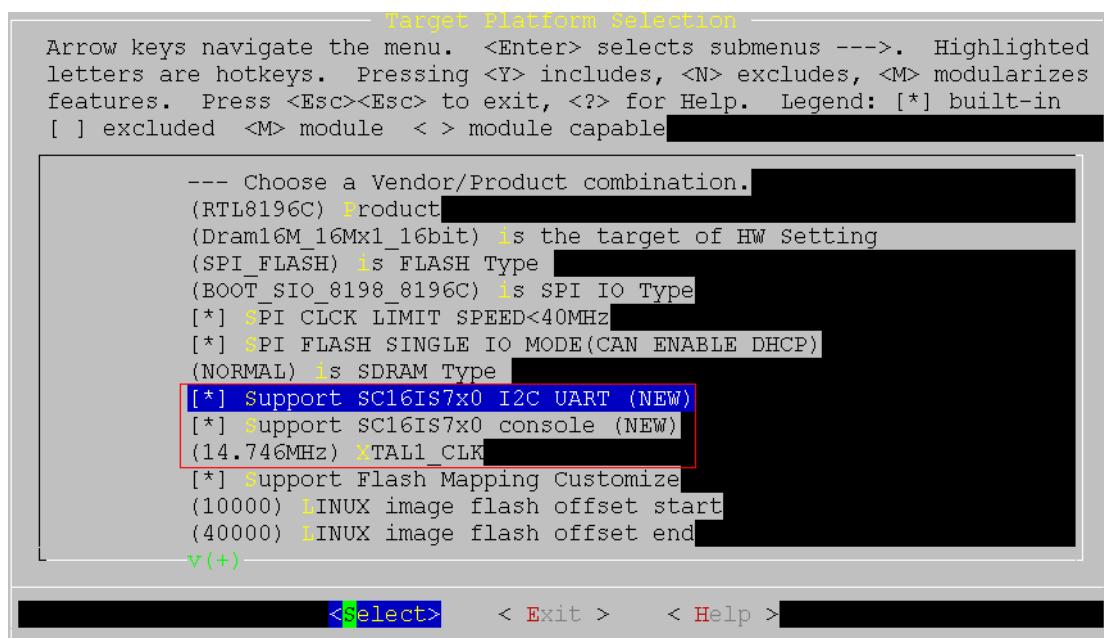
make menu

Target Plate Selection

```

Support SC16IS7x0 I2C UART // selected
Support SC16IS7x0 console // selected
(14.746MHz) XTAL1_CLK // XTAL1_CLK should be set based on hardware

```



4.8.2 Enable I2C UART in kernel

4.8.2.1 Modify code for GPIO pin in kernel(only valid at kernel 2.6.30)

According hardware designed, please modify the gpio pin mux which used by I2C uart. The gpio pin used by i2c uart is defined in linux-2.6.30/drivers/serial/8250_sc16is7x0.c

For example, following configuration means i2c uart use gpio pin D3~D6.

```

#define SC16IS7X0_RESET      I2C_GPIO_ID('D', 3)    // RESET = D3
#define SC16IS7X0_SCL        I2C_GPIO_ID('D', 4)    // SCL = D4
#define SC16IS7X0_SDA        I2C_GPIO_ID('D', 5)    // SDA = D5
#define SC16IS7X0_IRQ        I2C_GPIO_ID('D', 6)    // IRQ = D6

```

4.8.2.2 kernel menuconfig

1) Set ttys1 as console tty.

```
make linux_menuconfig // To configure linux kernel settings
```

Menuconfig:

Kernel hacking --->

(console=ttyS1, 38400 root=/dev/mtdblock1) Default kernel command string

```

Kernel hacking
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys.
Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help,
</> for Search. Legend: [*] built-in [ ] excluded <M> module < > module capable

[ ] Enable IRQFLAGS trace support
[ ] Enable __deprecated logic
[ ] Enable __must_check logic
(1024) Warn for stack frames larger than (needs gcc 4.4)
[ ] Magic SysRq key
[ ] Enable unused/obsolete exported symbols
[ ] Debug Filesystem
[ ] Run 'make headers check' when building vmlinux
[ ] Kernel debugging
[ ] Debug memory initialisation
[ ] Check for stalled CPUs delaying RCU grace periods
[ ] Sample kernel code --->
(Console1,38400 root=/dev/mtdblock1) Default kernel command string

<Select> < Exit > < Help >

```

2) Modify the 8250/16500 serial port number.

`make linux_menuconfig` // To configure linux kernel settings

Menuconfig:

Device Drivers --->

Character devices --->

Serial drivers --->

Maximum number of 8250/16550 serial ports = 2

Number of 8250/16550 serial ports to register at runtime = 2

Select SC16IS7x0 series (I2C bus) support and select correct Crystal for I2C based on hardware.

```

serial drivers
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys.
Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help,
</> for Search. Legend: [*] built-in [ ] excluded <M> module < > module capable

the item you wish to select followed by the <SPACE BAR>. Press
[*] Console on 8250/16550 and compatible serial port
(1) Maximum number of 8250/16550 serial ports
(2) Number of 8250/16550 serial ports to register at runtime
[ ] Extended 8250/16550 serial driver options
*** 8250 compatible port support ***
[*] SC16IS7x0 series (I2C bus) support
    Crystal for SC16IS7x0 XTAL1 (14.7465 MHZ (NXP demoboard)) --->
[*] Console on SC16IS7x0 port (ttys1)

<Select> < Exit > < Help >

```

4.9 DLNA support

Note: recommend to enable DLNA on 96D/97D/98 platform with at least 64M ram.

4.9.1 Linux kernel configure

4.9.1.1 SCSI support

Device Drivers --->
SCSI device support --->

```
[ ] RAID Transport Class  
[*] SCSI device support  
[ ] legacy /proc/scsi/ support  
--- SCSI support type (disk, tape, CD-ROM)  
[!] SCSI disk support  
[ ] SCSI tape support  
[ ] SCSI OnStream SC-x0 tape support  
[ ] SCSI CDROM support  
[ ] SCSI generic support  
[ ] SCSI media changer support  
--- Some SCSI devices (e.g. CD jukebox) support multiple LUNs  
[ ] Probe all LUNs on each SCSI device  
[ ] Verbose SCSI error reporting (kernel size +=12K)  
[ ] SCSI logging facility  
    SCSI Transports --->  
    SCSI low-level drivers --->
```

4.9.1.2 File systems configure

File systems --->
DOS/FAT/NT Filesystems --->

```
[*] MSDOS fs support  
[*] VFAT (Windows-95) fs support  
(437) Default codepage for FAT  
(iso8859-1) Default iocharset for FAT  
[ ] NTFS file system support
```

File systems --->

```
[ ] XFS filesystem support  
[ ] OCFS2 file system support  
[ ] Btrfs filesystem (EXPERIMENTAL) Unstable disk format  
[*] Enable POSIX file locking API  
[ ] Dnotify support  
[ ] Inotify file change notification support
```

Native Language Support --->

```
[*] Base native language support [iso8859-1] Default NLS Option (NEW)
[*]   Codepage 437 (United States, Canada)
[ ]   Codepage 737 (Greek) (NEW)
[ ]   Codepage 775 (Baltic Rim) (NEW)
[ ]   Codepage 850 (Europe) (NEW)
[ ]   Codepage 852 (Central/Eastern Europe) (NEW)
[ ]   Codepage 855 (Cyrillic) (NEW)
[ ]   Codepage 857 (Turkish) (NEW)
[ ]   Codepage 860 (Portuguese) (NEW)
[ ]   Codepage 861 (Icelandic) (NEW)
[ ]   Codepage 862 (Hebrew) (NEW)
[ ]   Codepage 863 (Canadian French) (NEW)
[ ]   Codepage 864 (Arabic) (NEW)
[ ]   Codepage 865 (Norwegian, Danish) (NEW)
[ ]   Codepage 866 (Cyrillic/Russian) (NEW)
[ ]   Codepage 869 (Greek) (NEW)
[ ]   Simplified Chinese charset (CP936, GB2312) (NEW)
[ ]   Traditional Chinese charset (Big5) (NEW)
[ ]   Japanese charsets (Shift-JIS, EUC-JP) (NEW)
[ ]   Korean charset (CP949, EUC-KR) (NEW)
[ ]   Thai charset (CP874, TIS-620) (NEW)
[ ]   Hebrew charsets (ISO-8859-8, CP1255) (NEW)
[ ]   Windows CP1250 (Slavic/Central European Languages) (NEW)
[ ]   Windows CP1251 (Bulgarian, Belarusian) (NEW)
[ ]   ASCII (United States) (NEW)
[*] NLS ISO 8859-1 (Latin 1; Western European Languages)
[ ] NLS ISO 8859-2 (Latin 2; Slavic/Central European Languages) (NEW)
[ ] NLS ISO 8859-3 (Latin 3; Esperanto, Galician, Maltese, Turkish) (NEW)
```

4.9.1.3 Enable USB support

Linux kernel configure for USB as follows.

make linux_menuconfig // To configure linux kernel settings

Menuconfig:

Device Drivers --->

USB support --->

Support for Host-side USB // selected

EHCI HCD (USB 2.0) support // selected

OHCI HCD support // selected

USB Mass Storage support // selected

```
[*] Support for Host-side USB
[ ]   USB verbose debug messages
--- Miscellaneous USB options
[ ]   USB device filesystem
[ ]   Enforce USB bandwidth allocation (EXPERIMENTAL)
[ ]   Dynamic USB minor allocation (EXPERIMENTAL)
--- USB Host Controller Drivers
[*]   EHCI HCD (USB 2.0) support
[ ]     Full speed ISO transactions (EXPERIMENTAL)
[ ]     Root Hub Transaction Translators (EXPERIMENTAL)
[ ]     Improved Transaction Translator scheduling (EXPERIMENTAL)
[*]   ISP116X HCD support
[*]   OHCI HCD support
[ ]   UHCI HCD (most Intel and VIA) support
[ ]   SL811HS HCD support
--- USB Device Class drivers
[ ]   USB Modem (CDC ACM) support
[ ]   USB Printer support
--- NOTE: USB_STORAGE enables SCSI, and 'SCSI disk support'
--- may also be needed; see USB STORAGE Help for more information
[*]   USB Mass Storage support
[ ]     USB Mass Storage verbose debug
[ ]     Datafab Compact Flash Reader support (EXPERIMENTAL)
[ ]     Freecom USB/ATAPI Bridge support
[ ]     Microtech/Zio! CompactFlash/SmartMedia support
[ ]     USBAT/USBAT02-based storage support (EXPERIMENTAL)
[ ]     SanDisk SDDR-09 (and other SmartMedia) support (EXPERIMENTAL)
[ ]     SanDisk SDDR-55 SmartMedia support (EXPERIMENTAL)
[ ]     Lexar Jumpshot Compact Flash Reader (EXPERIMENTAL)
```

4.9.1.4 General setup

General setup -->

```
[*] Optimize for size
-- Configure standard kernel features (for small systems) --->
[*] Strip assembler-generated symbols during link
[*] Support for hot-pluggable devices
[*] Enable support for printk log
[ ] Enable support for printk console
```

4.9.2 Application configure

4.9.2.1 busybox configure

make menuconfig and select Config busybox

Menuconfig:

Coreutils-->

df support // selected

```

BusyBox 1.13.4 Configuration

Coreutils
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys. Pressing <Y>
includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend:
[*] built-in [ ] excluded <M> module < > module capable

[ ] dasename
[ ] dd
[*] df
[ ] enable -a, -i, -B (NEW)
[ ] dirname
[ ] dos2unix/unix2dos
[ ] du (default blocksize of 512 bytes)
[*] echo (basic $u$V3 version taking no options)
[*] enable echo options (-n and -e)
[ ] env
v(+)

<select> < Exit > < Help >

```

4.9.2.2 enable usbmount

make users_menuconfig // To configure application settings

Menuconfig:

usbmount // selected

```

RLX Linux Configuration

the menu. <Enter> selects submenus --->. Highlighted letters are hot
ies, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help,
cluded <M> module < > module capable

-- (-) --
[*] pathsel
[ ] pmc utils
[ ] python support
[ ] reload
[ ] routed
[*] rtk_cmd daemon
[ ] RTK tcp2dev utility
[ ] rtk	voip
[ ] RTL8686 IPC User Utility
[*] samba support --->
[ ] scripts
[ ] simple config support
[ ] snmp
[ ] squashfs 4.0
[ ] squashfs 4.2
[ ] tr069
[ ] udhcp
[ ] ulinker autorun image tool
[ ] updateddd DDNS
[ ] upImage
[ ] usbmount
[ ] usbdisk update image
[ ] uWiFi

```

4.9.2.3 enable dlna

make users_menuconfig // To configure application settings

Menuconfig:

```

Dlna_dms // selected

[ ] mdadm-3.2.5
[ ] minidlna
[!] dlna_dms
[*] libiconv
[ ] miniigd
[ ] mini_upnp
[ ] mld proxy
[ ] mp daemon
[ ] mt-daapd
└ (+)

<Select> < Exit > < Help >

```

4.9.3 Test dlna using USB flash disk

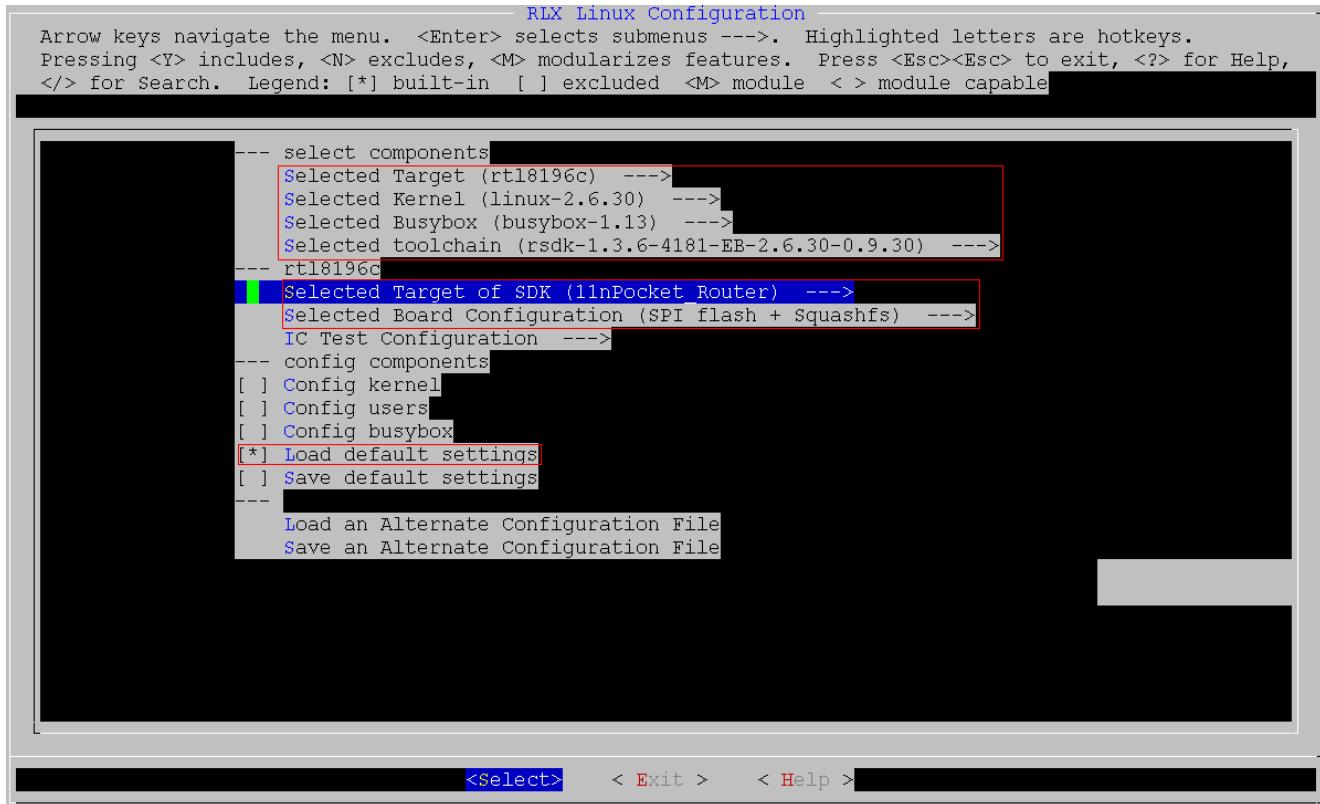
- (1) First of all, you should read and modify /boards/rtl8198/etc.default/ushare.conf.
- (2) After system boots up, plug-in an USB Flash disk. The directory to be shared is /var/tmp/usb/sda6/Media (you can modify this directory in ushare.conf).
- (3) Write this command: ‘ushare -f etc/ushare.conf &’ in console to enable dlna as a daemon.
- (4) Then you can enjoy the multimedia in your flash disk with PS3 or XBOX and so on.

4.10 Pocket AP support

At present, pocket AP is supported at RTL8196C(kernel 2.6.30). Configure pocket AP is as follows:

make menuconfig // To configure linux kernel settings

And the linux kernel configurations refer to the pic as follows.



4.11 Wireless client mode 802.1x support

Note: at present, wireless client mode 802.1x test as follows:

- 1) md5 / peap-mschapv2 / tls with linux radius server
- 2) md5 / peap-mschap v2 / tls with windows 2003 radius server

Enable wireless client mode 802.1x support as follows:

make linux_menuconfig // To configure linux kernel settings

Menuconfig:

Device Drivers --->

 Network device support --->

 Wireless LAN --->

 Client Mode 802.1x Support // selected

```

----- Wireless LAN -----
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted
letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes
features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*]
built-in [ ] excluded <M> module < > module capable [REDACTED]

[ ] Wireless LAN (pre-802.11)
[ ] Wireless LAN (IEEE 802.11)
[ ] Realtek 8190 wireless support
[ ] Realtek 8192SE wireless support
[*] RTL8192C/D 802.11b/g/n support
[*] 8198 clock source at 40Mhz
[*] Realtek 8192C wireless support
[*] Private skb buffer management
[*] Client Mode Support
[*] Client Mode 802.1x Support
[*] Repeater Mode support
[*] WDS Support
[*] Virtual AP Support
[ ] Efuse Support
[*] WAPI Support
[ ] support local AS
[ ] Config File support
[ ] Wireless Tools v29 support
[ ] Enable PCIE power saving support

v(+)

<Select> < Exit > < Help >

```

4.12 Hardware NAT

4.12.1 Hardware NAT description

Hardware NAT can be regarded as cache of Linux IP conntrack in hardware, without any relationship to iptables rules.

Hardware NAT table entry is tried to add when Linux PS (Protocol Stack) create IP conntrack.

Hardware NAT table entry is tried to delete when Linux PS destroy IP conntrack.

When Linux PS IP conntrack timeout, it will be checked whether there is no traffic leading to timeout or hardware offload leading to fake timeout.

Note: No API of hardware NAT is offered. All modifications of hardware NAT are patched to Linux kernel. So operations on Linux IP conntrack will modify hardware NATP entry synchronously. For example:

```

/* flush all ip conntrack & hw nat table entry */
void flushAllNaptSession (void)
{
    struct net *net;
    for_each_net (net) {
        nf_conntrack_flush (net, 0, 0); //clean conntrack table
    }
}

```

4.12.2 Proc files related to hardware NAT

- 1) Name: fast_nat

Path: /proc/fast_nat

Description: Flags for fast path control

Input Format:

```
echo "$FLAG" > /proc/fast_nat
```

Input Para:

- * the unit is non-zero: fastpath enabled
- * the unit is 0: fastpath disabled
- * "echo 0 > /proc/fast_nat": disable fastpath.
- * "echo 1 > /proc/fast_nat": enable fastpath.
- * "echo 2 > /proc/fast_nat": clean the conntrack table.

Output Format:

\$FLAG

Output Para:

- * 10: disable fastpath.
- * 11: enable fastpath.
- * 12: clean the conntrack table.

2) hw_nat

Path: /proc/hw_nat

Description: Flags for hardware NAT control

Input Format:

```
echo "$FLAG" > /proc/hw_nat
```

Input Para:

- * the unit is non-zero: hardware NAT enabled
- * the unit is 0: hardware NAT disabled
- * "echo -1 > /proc/hw_nat": hardware NAT disabled, change to gateway mode. It will flush hw l3 table and hw ip table, when echo 1 > /proc/hw_nat again, hw nat can not work again. This value is only used when br0 or eth1's ip set as A or B class network address.
- * "echo 0 > /proc/hw_nat": hardware NAT disabled, change to gateway mode. It will not flush hw l3 table and hw ip table, when echo 1 > /proc/hw_nat again, hw nat can work again.
- * "echo 1 > /proc/hw_nat": hardware NAT enabled, change to gateway mode.
- * "echo 2 > /proc/hw_nat": Change to bridge mode.
- * "echo 3 > /proc/hw_nat": Change to WISP mode.
- * "echo 4 > /proc/hw_nat": simply disabled the hardware NAT.
- * "echo 5 > /proc/hw_nat": simply disabled the hardware NAT.
- * "echo 8 > /proc/hw_nat": simply disabled the hardware NAT.

* “echo 9 > /proc/hw_nat”: init hardware NAT parameters. (Must init before hardware NAT works)

Output Format:

\$FLAG

Output Para:

- * 0: gateway mode & hardware NAT disabled.
- * 1: gateway mode & hardware NAT enabled.
- * 2: bridge mode.
- * 3: WISP mode.
- * 4: hardware NAT disabled.
- * 5: hardware NAT disabled.
- * 8: hardware NAT disabled.
- * 9: hardware NAT parameters have already initialized.
- * others: no means

3) Name: nf_conntrack

Path: /proc/net/nf_conntrack

Description: this proc can indicate whether hardware NAT is applied to each IP conntrack.

4.12.3 Hardware NAT characteristic & limitation

Following is the characteristic & limitation about hardware nat

- 1) support static/dhcp/pppoe wan type, not support vpn tunnel (PPTP/L2TP/multi-PPPoE wan type).
- 2) not support IP conntracks which need do ALG.
- 3) when URL filter enabled, must disable hardware nat.
- 4) when enable software qos, must disable hardware nat.
- 5) when enable rtk vlan, must disable hardware nat.
- 6) not support IP fragment packets.
- 7) support ipv4, not support ipv6.
- 8) support tcp/udp, not support icmp....
- 9) not support encryption packets (e.g.: IPSec).
- 10) not support A or B class network address.
- 11) support server port / trigger port / DMZ / port mapping etc, and hardware NAT need not do anything special for the features related to iptables rules because hardware NAT is independent of the iptables rules.
- 12) won't affect alg when enable hardware nat.

13) total 1024 hardware nat entries, bi-direction support 512 connection.

Note: If wan type changes to PPTP/L2TP/multi-PPPoE or URL filter/QoS function is enabled, hardware NAT must be manually disabled by echo correct value to "/proc/hw_nat". Others like ALG or IP fragment will be automatically processed, no other settings involved.

For these hw nat limitaiton, you can check the function update_hwnat_setting() in users/boa/system/set_firewall.c.

4.12.4 Enable/Disable hardware NAT support

For RTL8198, enable/disable hardware NAT support as follows:

make linux_menuconfig // To configure linux kernel settings

Menuconfig:

Device Drivers --->

Network device support --->

Options for Realtek SoC --->

Config for Layered Driver Features --->

Hardware Features Selection (Enable RTL Hardware NAPT) --->

/* (Default) Selected to enable hardware NAT */

Enable RTL Hardware NAPT

/* If selected, enable hardware L2 and multicast function */

Enable RTL Hardware Multicast Only

/* If selected, enable hardware L2 function */

Enable RTL Hardware switch Only

Note: if hardware NAT is disabled, hardware QoS (refer to section 4.14) should be disabled too.



4.12.5 Hardware NAT init flow

If macro CONFIG_RTK_VLAN_SUPPORT enabled, execute the following commands after "echo 0 > /proc/rtk_vlan_support".

Step1:

Init hardware tables by command:

```
echo -1 > /proc/hw_nat
```

Step2:

Enable hardware nat by command:

```
echo 1 >/proc/hw_nat
```

Step3:

Configure Lan/Wan interface's ip address

For example:

```
ifconfig br0 192.168.1.254
```

```
Ifconfig eth1 192.168.2.100
```

The linux kernel will automatically add Lan/Wan routing rules, and then sync to hardware routing table

Step4:

Configure hardware ip table by setting iptables nat rule.

For example:

```
iptables -t nat -I POSTROUTING -o eth1 -j MASQUERADE
```

```
(or iptables -t nat -I POSTROUTING -o ppp0 -j MASQUERADE )
```

Step5:

Configure hardware default route by command

For example:

```
route add default gw 192.168.2.200 dev eth1
```

```
(or route add default gw 192.168.2.200 dev ppp0 )
```

Step6:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Allow the linux kernel to do IP packet forwarding from one network interface to another network interface.

4.13 Iptables rule to ACL rule translation support

In order to offload cpu when firewall is enabled by iptables rules, the feature of iptables rule to

ACL rule translation can be enabled.

Enable iptables rule to ACL rule translation support as follows:

make linux_menuconfig // To configure linux kernel settings

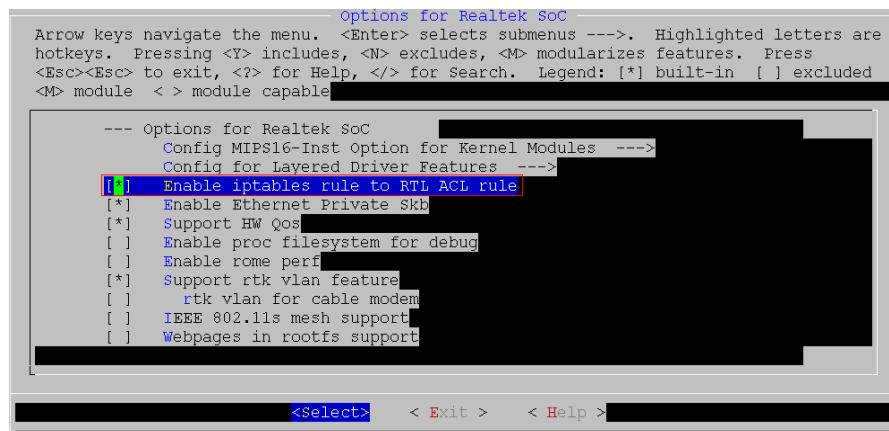
Menuconfig:

Device Drivers --->

Network device support --->

Options for Realtek SoC --->

Enable iptables rule to RTL ACL rule // (Default) Selected



Note: if iptables rule to ACL rule is enabled, hardware NAT is recommend.

4.14 Hardware QoS support

4.14.1 How to enable Hardware QoS

In order to offload cpu when QoS is enabled, hardware QoS can be used.

Hardware QoS is based on hardware NAT. So if we want to use hardware QoS, hardware NAT should be enabled at first.

After hardware NAT (refer to section 4.12.4) is enabled, enable hardware QoS as follows:

make linux_menuconfig // To configure linux kernel settings

Menuconfig:

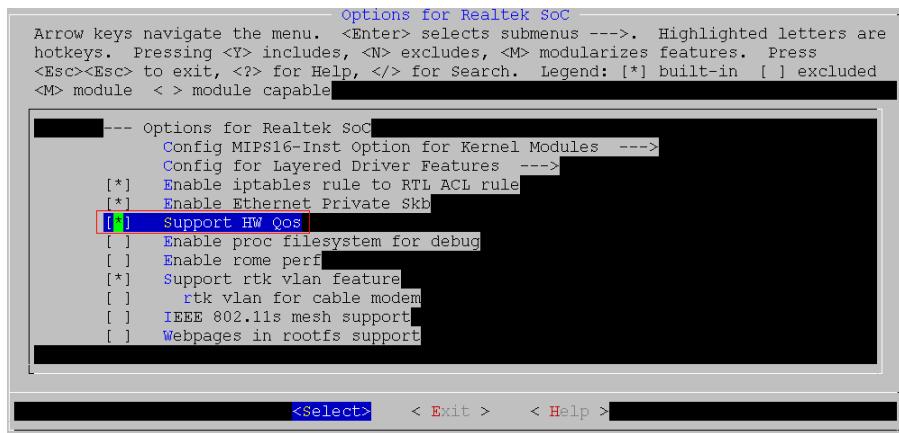
Device Drivers --->

Network device support --->

Options for Realtek SoC --->

Support HW Qos // Selected

Note: For more details of hardware QoS, please refer to Realtek_QoS_v*.pdf.



4.14.2 Configure HW QoS by rtk_cmd

4.14.2.1 Function illustration

The customer could input “rtk_cmd qos” to implement the configuration for hardware qos.

NOTE: The function is only supported in trunk now.

4.14.2.2 Introduction of Command

1. ENABLE/DISABLE HW_QoS:

Enable HW QoS, input the following command in console please:

rtk_cmd qos Flow_Control ENABLE

Disable HW QoS, input the following command in console please:

rtk_cmd qos Flow_Control DISABLE

If you want to show the flow control configure, input the following command in console please:

rtk_cmd qos SHOW FLOW_CONTROL_CONFIG

2. QUEUE_NUMBER:

Set queue number of each port, input the following command in console:

rtk_cmd qos Queue_Num PORT_NUM1 QUEUE_NUM1 PORT_NUM2 QUEUE_NUM2...

Input para:*PORT_NUM1 QUEUE_NUM1 PORT_NUM2 QUEUE_NUM2...*

Para *PORT_NUM1,PORT_NUM2* mean the port you want to set ;

Para *QUEUE_NUM,QUEUE_NUM2* mean the queue number you want to use.

e.g. if you want to use 2 queues of port0, and use 3queues of port1, then please input the command in console:(please follow the format in sample)

rtk_cmd qos Queue_Num port0 2 port1 3

If you want to check the queue number of each port, please input the command in console :

rtk_cmd qos SHOW QUEUE_NUMBER

3. PRIORITY_DECISION:

ASIC support 5 kinds of priority policy: based on port, dscp, 802.1p, acl or nat. Please choose one kind policy and input the value of decision. Each decision takes up 4 bits. e.g. if you want to set port 15, dscp 1 802.1p 2, acl 3, nat 4, please input the following command in console:

```
rtk_cmd qos Priority_Decision port 15 vlan 1 dscp 2 acl 3 nat 4
```

Note: The value between port and 15, vlan and 1 need a blank .

If you want to check the priority of policies, please input the following command in console:

```
rtk_cmd qos SHOW PRIORITY_DECISION
```

4. PRIORITY_ASSIGN:

Now, rtk_cmd qos support 3 kinds policy: Port_Based, Dscp_Based, 802.1p_Based.

1) Port_Based:

The priority is assigned by port. If you want to set system priority of port0= 3, system priority of port1=4, system priority of port2=5, please input the follow command in console:

```
rtk_cmd qos Priority_Assign Port_Based port0 3 port1 4 port2 5
```

2) Dscp_Based:

The priority is assigned by dscp. If you want to set system priority=1 while dscp=10, system priority=2 while dscp=61, then input the follow command in console:

```
rtk_cmd qos Priority_Assign Dscp_Based dscp10 1 dscp61 2
```

If you want to display the priority of each dscp, please input the command in console:

```
rtk_cmd qos SHOW DSCP_BASED_PRI
```

3) 802.1p_Based:

If you want to set system priority=2 while vlan priority=1, set system priority=3while vlan priority=2, then please input the following command in console:

```
rtk_cmd qos Priority_Assign Vlan_Based vlan1 2 vlan2 3
```

If you want to check the system priority correspond to each vlan priority, please input the following command in console:

```
rtk_cmd qos SHOW VLAN_BASED_PRI
```

5. PRIORITY_TO_QID:

Before set the qid correspond to each system priority, please make sure that the queue number of each port has been set. If set priority 1 corresponding to qid 1, priority 2 corresponding to qid 2, please input the following command in console:

```
rtk_cmd qos Priority_to_Qid pri1 1 pri2 2
```

If you want to display the mapping relationship between system priority and qid, please input the following command in console:

```
rtk_cmd qos SHOW PRIORITY_TO_QID
```

6. QUEUE_TYPE:

Queue could be set to two types: STRICT or WEIGHTED.

1) STRICT

If you want to set queue0, queue1 and queue2of port1 to STRICT type, please input the following command in console:

```
rtk_cmd qos Queue_Type STRICT port0 q0 q1 q2
```

If you want to check that which queues of port have been set to STRICT type, please input:

```
rtk_cmd qos SHOW QUEUE_TYPE_STRICT
```

2) WEIGHTED

If you want to set queue0 of port1 to WEIGHTED type, and the weight=15; set queue1 to WEIGHTED type and weight=18, then please input the following command in console:

```
rtk_cmd qos Queue_Type WEIGHTED port1 q0 15 q1 18
```

If you want to check which queues of port have been set to WEIGHTED type, and what the weight values are, then please input:

```
rtk_cmd qos SHOW QUEUE_TYPE_WEIGHTED
```

7. REMARK:

ASIC support 2 kinds of remark: VLAN remark and DSCP remark.

1) VLAN REMARK

If you want to remark vlan priority=2 when port=0, system priority=1, please input in console:

```
rtk_cmd qos Remark VLAN port0 pri1 2
```

If you want to check the information of vlan remark, please input in console:

```
rtk_cmd qos SHOW VLAN_REMARK
```

2) DSCP REMARK:

If you want to remark dscp=61, when port=0, system priority=2, please input in console:

```
rtk_cmd qos Remark DSCP port0 pri2 61
```

If you want to check the information of dscp remark, please input in console:

```
rtk_cmd qos SHOW DSCP_REMARK
```

8. QUEUE_RATE:

If you want to set the following parameters of queue0 in port0: ppr=7, burst size=225, apr=16381, please input in console:

```
rtk_cmd qos Rate port0 q0 ppr 7 burst 255 apr 16383
```

If you want to check the value of ppr, burst size and apr of each queue in each port, please input in console:

```
rtk_cmd qos SHOW QUEUE_RATE
```

4.14.2.3 NOTE ITEMS

When input the following value in console, please pay attention to the range respective:

1. vlan priority: 3bit;
2. system priority: 3bit;
3. port: 3bit;
4. dscp value: 6bit;
5. weight: 7bit;
6. ppr: 3bit;
7. burst size: 8bit;
8. apr: 14bit

4.15 IPv6 support

In order to add IPv6 support, Linux kernel, IPv6 daemon and Busybox need to be configured.

4.15.1 Linux kernel for IPv6 configure

Configure Linux kernel for IPv6 as follows:

```
make linux_menuconfig // To configure linux kernel settings
```

Menuconfig:

1) Networking support --->

 Networking options--->

 The ipv6 protocol---> // Selected and enter

```

Networking options
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are
hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press
<Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ] excluded
<M> module < > module capable
^(-)
[ ] IP: IPsec tunnel mode
[ ] IP: IPsec BEET mode
[ ] Large Receive Offload (ipv4/tcp)
[ ] INET: socket monitoring interface
[ ] TCP: advanced congestion control --->
[ ] TCP: MD5 Signature Option support (RFC2385) (EXPERIMENTAL)
[*] The IPv6 protocol --->
[ ] Security Marking
[*] Network packet filtering framework (Netfilter) --->
[ ] The DCCP Protocol (EXPERIMENTAL) --->
[ ] The SCTP Protocol (EXPERIMENTAL) --->
[ ] The TIPC Protocol (EXPERIMENTAL) --->
[ ] Asynchronous Transfer Mode (ATM)
v(+)

<Select> < Exit > < Help >

```

2) Networking support --->

Networking options--->

The ipv6 protocol--->

IPv6: ready logo patch	// Selected
IPv6: Enable RFC 4429 Optimistic DAD (EXPERIMENTAL)	// Selected
IPv6: multicast routing (EXPERIMENTAL)	// Selected

Netfiler--->

IPv6 netfilter configure--->

ip6tables support--->	// Selected
packet mangling--->	// Selected
packet filtering --->	//Selected

```

The IPv6 protocol
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are
hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press
<Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ] excluded
<M> module < > module capable

--- The IPv6 protocol
[*] IPv6: ready logo patch
[ ] IPv6: Privacy Extensions (RFC 3041) support
[ ] IPv6: Router Preference (RFC 4191) support
[*] IPv6: Enable RFC 4429 Optimistic DAD (EXPERIMENTAL)
[ ] IPv6: AH transformation
[ ] IPv6: ESP transformation
[ ] IPv6: IPComp transformation
[ ] IPv6: Mobility (EXPERIMENTAL)
[*] IPv6: IPsec transport mode
[*] IPv6: IPsec tunnel mode
[*] IPv6: IPsec BEET mode
[ ] IPv6: MIPv6 route optimization mode (EXPERIMENTAL)
[*] IPv6: IPv6-in-IPv4 tunnel (SIT driver)
[ ] IPv6: IP-in-IPv6 tunnel (RFC2473)
[ ] IPv6: Multiple Routing Tables
[*] IPv6: multicast routing (EXPERIMENTAL)
[ ] IPv6: PIM-SM version 2 support (EXPERIMENTAL)

<Select> < Exit > < Help >

```

```

IPv6: Netfilter Configuration
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are
hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes
features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend:
[*] built-in [ ] excluded <M> module < > module capable

[*] IPv6 connection tracking support
[ ] IPv6 Userspace queueing via NETLINK (OBSOLETE)
[*] IPv6 tables support (required for filtering)
[ ] "ah" match support
[ ] "eui64" address check
[ ] "frag" Fragmentation header match support
[ ] "hbh" hop-by-hop and "dst" opts header match support
[ ] "hl" hoplimit match support
[ ] "ipv6header" IPv6 Extension Headers Match
[ ] "mh" match support
[ ] "rt" Routing header match support
[ ] "HL" hoplimit target support
[ ] LOG target support
[*] Packet filtering
[ ] REJECT target support
[*] Packet mangling

```

4.15.2 IPv6 daemon configure

Configure IPv6 daemon as follows:

make users_menuconfig // To configure application settings

Menuconfig:

1) dhcpcv6 // Selected

2) radvd // Selected

3) ecmh // Selected

```
----- RLX Linux Configuration -----
the menu. <Enter> selects submenus --->. Highlighted letters are hot
keys, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help,
excluded <M> module < > module capable
```

```
--- Applications Settings
  BT Configuration --->
  File_system Configuration --->
  IPv6 Configuration --->
  PPP Configuration --->
  RTK_inband Configuration --->
  USB3G Configuration --->
  [ ] alsa related
  [ ] auth
```

```
----- IPv6 Configuration -----
the menu. <Enter> selects submenus --->. Highlighted letters are hot
keys, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help,
excluded <M> module < > module capable
```

```
[ ] 6rd
[*] dhcpv6
[ ] dnsmasq
[ ] ds-lite
[*] ecmh
[ ] ndp proxy
[*] radvd
```

4) PPP support IPv6 //Selected

```
----- RLX Linux Configuration -----
the menu. <Enter> selects submenus --->. Highlighted letters are hot
keys, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help,
excluded <M> module < > module capable
```

```
--- Applications Settings
  BT Configuration --->
  File_system Configuration --->
  IPv6 Configuration --->
  PPP Configuration --->
  RTK_inband Configuration --->
  USB3G Configuration --->
  [ ] alsa related
```

```
    PPP Configuration
the menu. <Enter> selects submenus --->. Highlighted letters are hot
ies, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help,
xcluded <M> module < > module capable

[ ] 12tpd
[ ] Multi PPPoE Support
[ ] ppp reject error session patch
[*] pppd
[!] IPv6 over PPP support
[ ] pppoe_disc_flow_patch
[ ] pptp
```

5) BOA support IPv6 //Selected

```
    RLX Linux Configuration
the menu. <Enter> selects submenus --->. Highlighted letters are hot
ies, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help,
xcluded <M> module < > module capable

--- Applications Settings
  BT Configuration --->
  File_system Configuration --->
  IPv6 Configuration --->
  PPP Configuration --->
  RTK_inband Configuration --->
  USB3G Configuration --->
[ ] alsa related
[ ] auth
[*] bluez
  Bluez Version (Bluez 4.96) --->
[ ] RTK bluetooth firmware selection
[*] boa
[ ] boa CGI Support (NEW)
[!] boa IPv6 Support
[ ] boa new UI Support (NEW)
[ ] brctl
```

6) ip6tables support //Selected

```
----- RLX Linux Configuration -----
the menu. <Enter> selects submenus --->. Highlighted letters are hot
keys, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help,
excluded <M> module < > module capable

-- (-)
[*] boa
[ ]   boa CGI Support (NEW)
[*]   boa IPv6 Support
[ ]   boa new UI Support (NEW)
[ ] brctl
[ ] busybox
[ ] CPU utilization
[ ] curl
[ ] curl-7.34.0
[ ] discover
[ ] dnrd
[*] firmware write daemon
[ ] FT Daemon support
[ ] gdbserver
[ ] gproxy
[ ] hostapd 2.0
[ ] iapp
[ ] igmp proxy
[ ] ioh
[ ] iproute2
[*] iptables
[!] ip6tables
[ ]   l7filter_pat (NEW)
[ ] iwcontrol
```

7) dnsmasq support //Selected

```
----- IPv6 Configuration -----
the menu. <Enter> selects submenus --->. Highlighted letters are hot
keys, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help,
excluded <M> module < > module capable

[ ] 6rd
[*] dhcpv6
[!] dnsmasq
[ ] ds-lite
[*] ecmh
[ ] ndp proxy
[*] radvd
```

8) mldproxy support //Selected

```

----- RLX Linux Configuration -----
the menu. <Enter> selects submenus --->. Highlighted letters are hot-
keys, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help,
<cluded <M> module <> module capable

(-)
[*] firmware write daemon
[ ] FT Daemon support
[ ] gdbserver
[ ] gproxy
[ ] hostapd 2.0
[ ] iapp
[ ] igmp proxy
[ ] ioh
[ ] iproute2
[*] iptables
[*] ip6tables
[ ] l7filter_pat (NEW)
[ ] iwcontrol
[ ] lltdd
[ ] login on console
[ ] MAP-E
[ ] mdadm-3.2.5
[ ] minidlna
[*] dlna_dms
[*] libiconv
[ ] miniigd
[ ] mini_upnp
[*] mld proxy
[!] support mld proxy multiwan
[ ] ntp daemon

```

4.15.3 Busybox for IPv6 configure

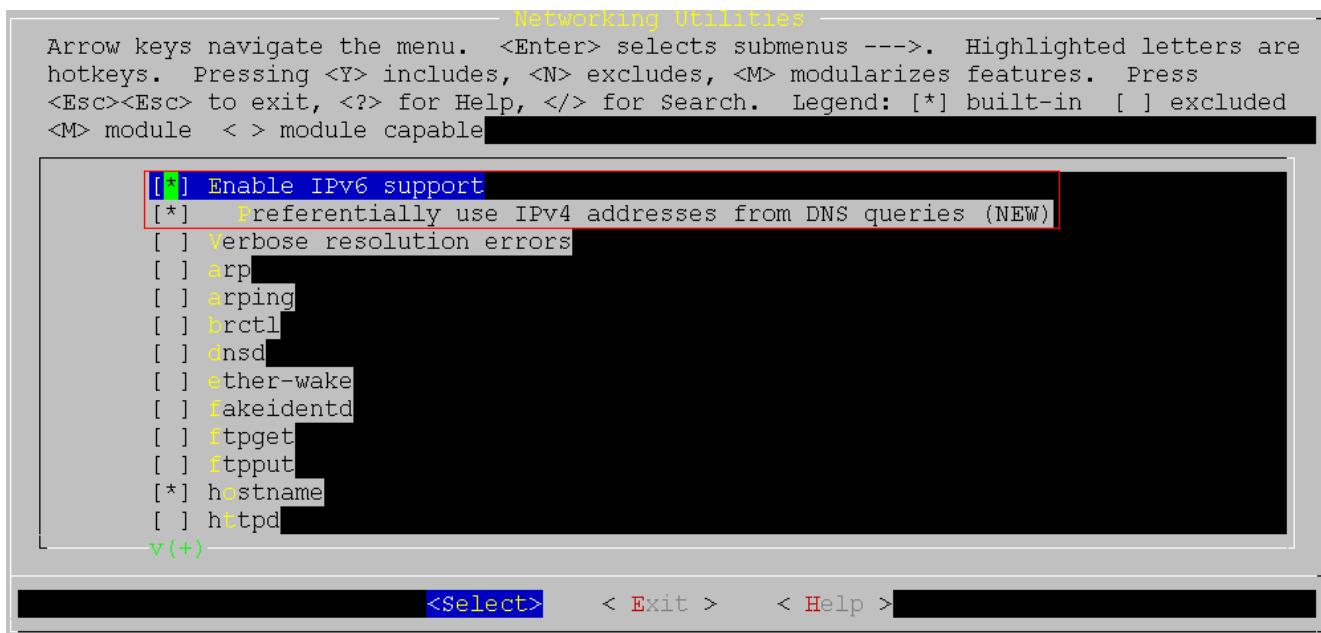
Enter the directory of Busybox and make menuconfig to configure Busybox for IPv6 as follows:

Menuconfig:

1) Networking Utilities --->

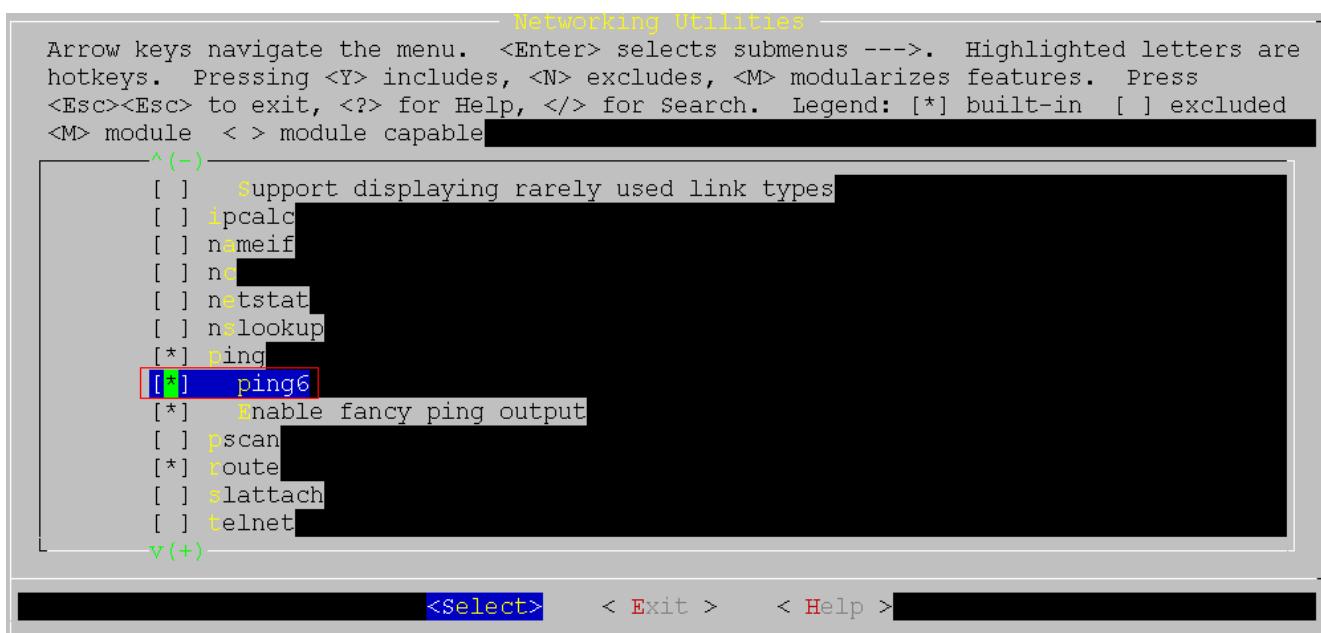
Enable IPv6 support // Selected

Preferentially use IPv4 addresses from DNS queries (NEW) // Selected



2) Networking Utilities --->

ping6 // Selected



4.15.4 Test IPv6

Add IPv6 support as above and update image to our AP.

In order to check whether IPv6 works, input command at AP console as follows:

`ifconfig br0 // To check that ipv6 link local address of br0 should exist if IPv6 works.`

4.15.5 The flash & SDRAM footprint for demo board.

1) RTL8198 + RTL8192C

Enable IPv6 will lead to 262KBytes increase in flash footprint, and increase 596Kbytes in SDRAM footprint.

RTL8198 + RTL8192C demo board v630 run SDK v2.5 image (2011/06/10), the test result as follows:

Configure \ Test entry	fw.bin (KB)	MemFree (KB)
Default	1962	16420
Enable IPv6	2224	15824

2) RTL8196C+RTL8192D

RTL8196C+RTL8192D demo board v400 run image (svn version: 6977 2011/08/18), the test result as follows:

Configure\Test entry	Root.bin(KB)	Linux.bin(KB)	fw.bin(KB)	Memfree(KB)
Enable ipv6	1201	885	2203	15664

4.16 64K/sector SPI flash support

In order to add 64K/sector SPI flash support, both Linux kernel and Bootloader need to be re-configured.

1) Linux kernel configure as follows (here we take a 64K/sector 8M SPI flash as an example):

make linux_menuconfig // To configure linux kernel settings

Menuconfig:

System Configuration --->

(0x800000) Size of Flash	// 8M Flash
(0x10000) Hardware setting offset in flash.	// user defined, but 64K alignment needed
(0x20000) Default setting offset in flash.	// user defined, but 64K alignment needed
(0x30000) Current setting offset in flash.	// user defined, but 64K alignment needed
(0x40000) webpages image offset in flash.	// user defined, but 64K alignment needed
(0x100000) linux image offset in flash.	// user defined, but 64K alignment needed
(0x300000) root image offset in flash.	// user defined, but 64K alignment needed

```

System Configuration
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys.
Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help,
</> for Search. Legend: [*] built-in [ ] excluded <M> module <> module capable
^(-)
[ ] Enable timer adjustment support
[ ] Webpages in rootfs support
[*] SPI flash support
[ ] Enable Flash Dual Bank support
[*] Enable Flash Mapping
[ ] USB3G support
[ ] Seadup usb samba performance
*** Flash size 2M or 4M, default 2M ***
(0x800000) Size of Flash
*** Hardware setting offset,should be 4K alignment ***
(0x10000) Hardware setting offset in flash.
*** Default setting offset,should be 4K alignment. ***
*** size of default and current setting should be same. ***
(0x20000) Default setting offset in flash.
*** Current setting offset,should be 4K alignment. ***
(0x30000) Current setting offset in flash.
*** Webpage image offset,should be 4K alignment. ***
*** size of web page is normally about 100K. ***
(0x40000) webpages image offset in flash.
*** Linux image offset,should be 4K alignment. ***
*** this offset MUST between 0x10000~0x40000. ***
(0x100000) linux image offset in flash.
*** Root image offset,should be 64K alignment. ***
(0x300000) root image offset in flash.
(2) Kernel Stack Size Order Configuration

<Select> < Exit > < Help >

```

2) Bootloader configure as follows (based on the Linux kernel configurations above):

make menuconfig // To configure bootloader settings

Menuconfig:

```

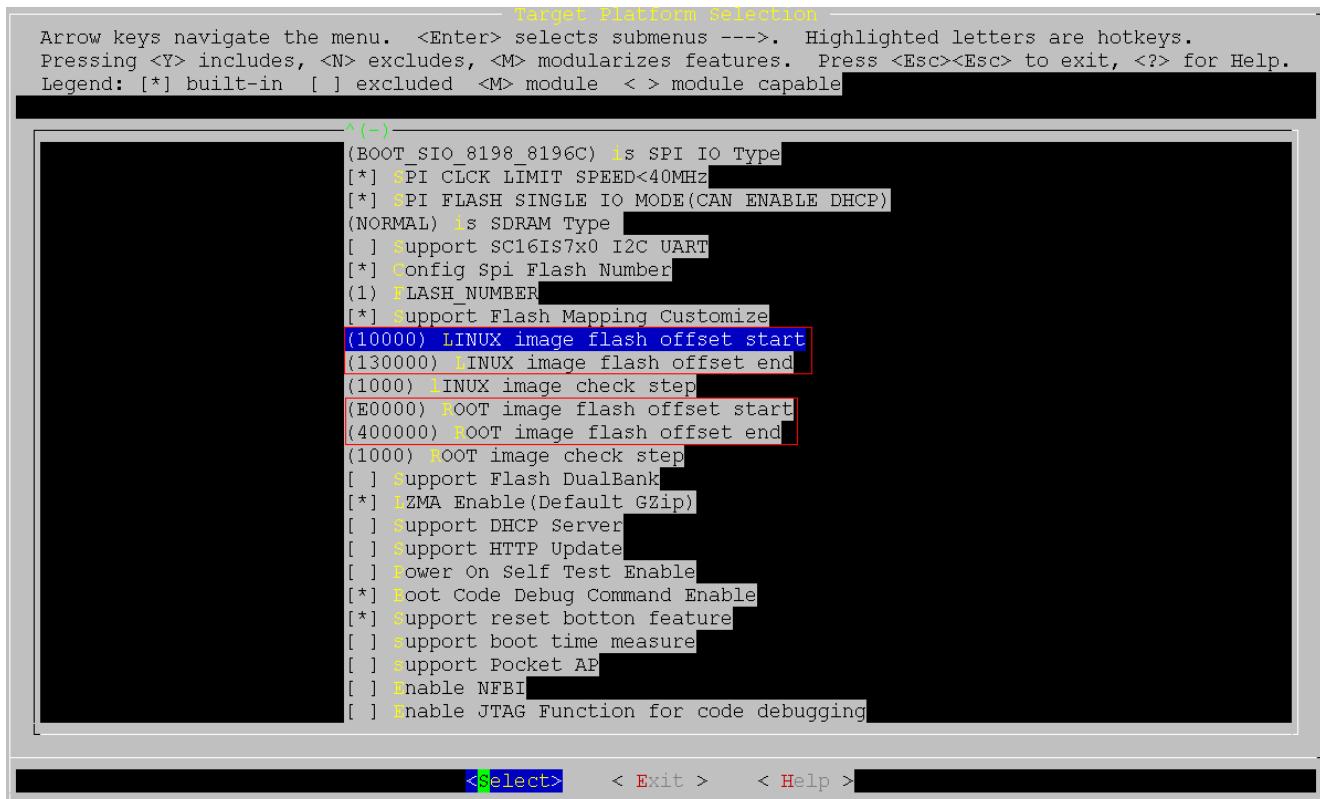
[*] Support Flash Mapping Customize
/* Linux image flash offset range 0x10000 ~ 0x130000 which Bootloader will auto check
should cover 0x100000 (Linux image offset in flash) set by Linux kernel above */

(10000) LINUX image flash offset start
(130000) LINUX image flash offset end

/* Root image flash offset range 0xE0000 ~ 0x400000 which Bootloader will auto check
should cover 0x300000 (Root image offset in flash) set by Linux kernel above */

(E0000) ROOT image flash offset start
(400000) ROOT image flash offset end

```



4.17 SPI Flash Support

4.17.1 Two SPI Flash Support

Add two SPI flash support need to modify both bootloader configuration and Linux kernel configuration.

4.17.1.1 Configure bootloader for two SPI flash support

Enter bootloader directory and *make menuconfig* to configure bootloader settings as follows:

Target Platform Selection --->

```

//Selected to add more than one SPI flash support

[*] Config Spi Flash Number (NEW)

//Input number of SPI flash to support, such as 2 here

(2) FLASH_NUMBER (NEW)

```

```

----- Target Platform Selection -----
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters
are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features.
Press <Esc><Esc> to exit, <?> for Help. Legend: [*] built-in [ ] excluded
<M> module < > module capable
^ (-)
[RTL8196C) Product
(Dram16M_16Mx1_16bit) is the target of HW Setting
(SPI_FLASH) is FLASH Type
(BOOT_SIO_8198_8196C) is SPI IO Type
[*] SPI CLK LIMIT SPEED<40MHz
[*] SPI FLASH SINGLE IO MODE(CAN ENABLE DHCP)
(NORMAL) is SDRAM Type
[ ] Support SC16IS7x0 I2C UART
[*] Config Spi Flash Number (NEW)
(2) FLASH NUMBER (NEW)
[*] Support Flash Mapping Customize
(10000) LINUX image flash offset start
(40000) LINUX image flash offset end
(1000) LINUX image check step
(E0000) ROOT image flash offset start
(130000) ROOT image flash offset end
(1000) ROOT image check step
[ ] Support Flash DualBank
v (+)

<Select> < Exit > < Help >

```

4.17.1.2 Configure Linux kernel for two SPI flash support

Enter Linux kernel directory and *make menuconfig* to configure Linux kernel settings as follows:

System Configuration --->

```

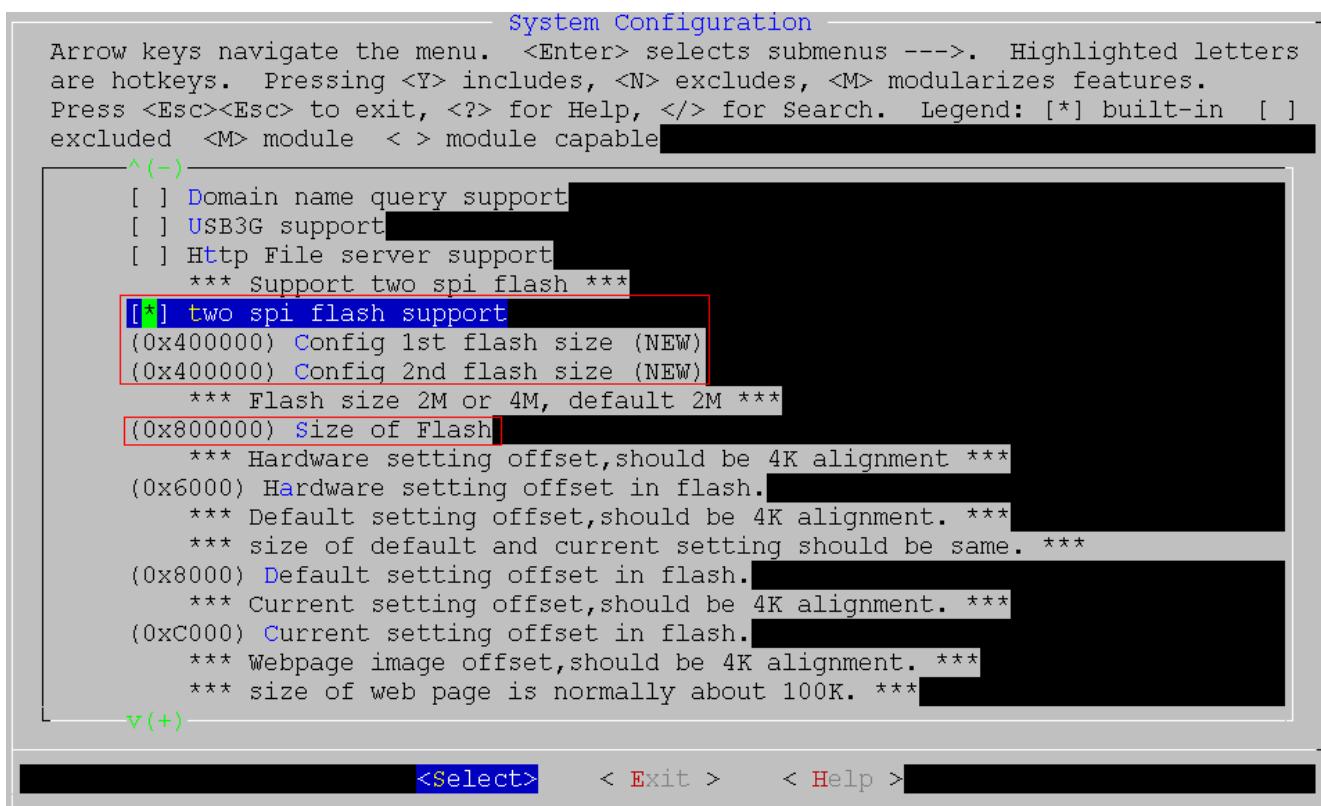
[*] two spi flash support      // Selected to add two SPI flash support

(0x400000) Config 1st flash size (NEW)    //Input the 1st flash size, such as 4MB here

(0x400000) Config 2nd flash size (NEW)    //Input the 2nd flash size, such as 4MB here

(0x800000) Size of Flash      // Input total size of two SPI flash, such as 8MB here

```



4.17.2 Add New SPI Flash

If want to add a new SPI flash, bootloader and kernel source should be modified as below

4.17.2.1 Add New SPI Flash in bootcoder

Find the “spi_common.c” file in bootloader, and search “struct spi_flash_known spi_flash_registered [] “, fill in the structure with new SPI flash’s info. The chip info can Get from datasheet.

```
struct spi_flash_known
{
    unsigned int      uiChipId;
    unsigned int      uiDistinguish;
    unsigned int      uiCapacityId;
    unsigned int      uiBlockSize;
    unsigned int      uiSectorSize;
    unsigned int      uiPageSize;
    char*            pcChipName;
    unsigned int      chipClock;

    FUNC_ERASE           pfErase;
    FUNC_READ            pfRead;
    FUNC_SETQEBIT        pfQeBit;
    FUNC_PAGEWRITE       pfPageWrite;
```

};

Where “chipClock” is the frequency the SPI flash good work at. The callback functions Should be fill with common function used in the file if the SPI flash is generic. Maybe “pfQeBit” Need to be re-implemented according to NEW SPI flash’s datasheet.

4.17.2.2 Add New SPI Flash in linux

The same operation as done in bootloader.

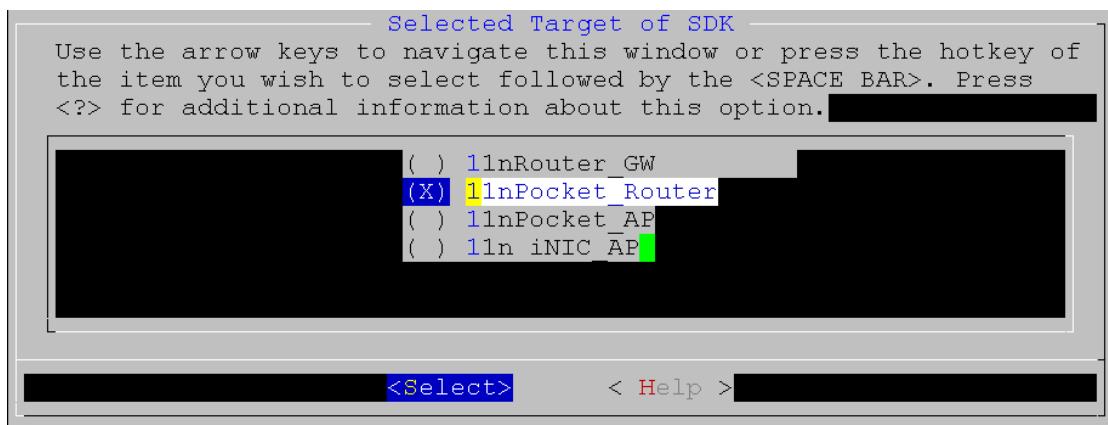
4.18 92C/D support

4.18.1 92C support for Pocket AP SDK

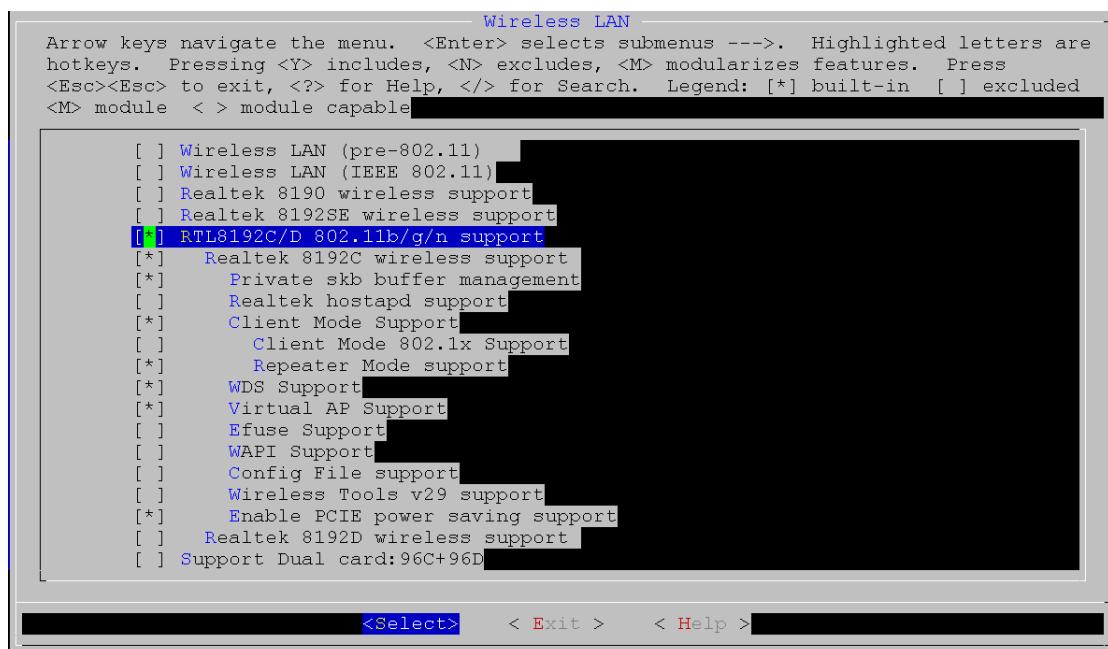
1) Select SDK for Pocket AP 92C:

Selected Target of SDK --->

And selected “11nPocket_Router”



2) Enable 92C Driver

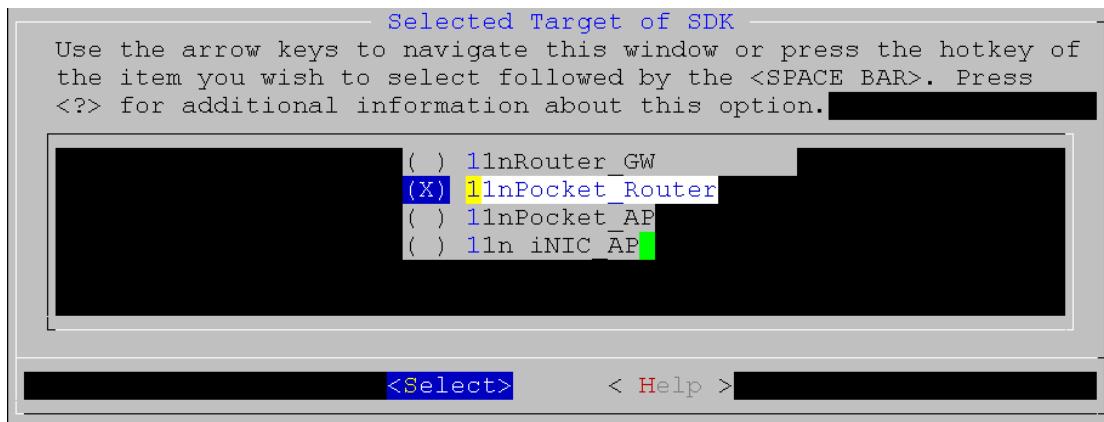


4.18.2 92D support for Pocket AP SDK

1) Select SDK for Pocket AP 92D:

Selected Target of SDK --->

And selected “11nPocket_Router”



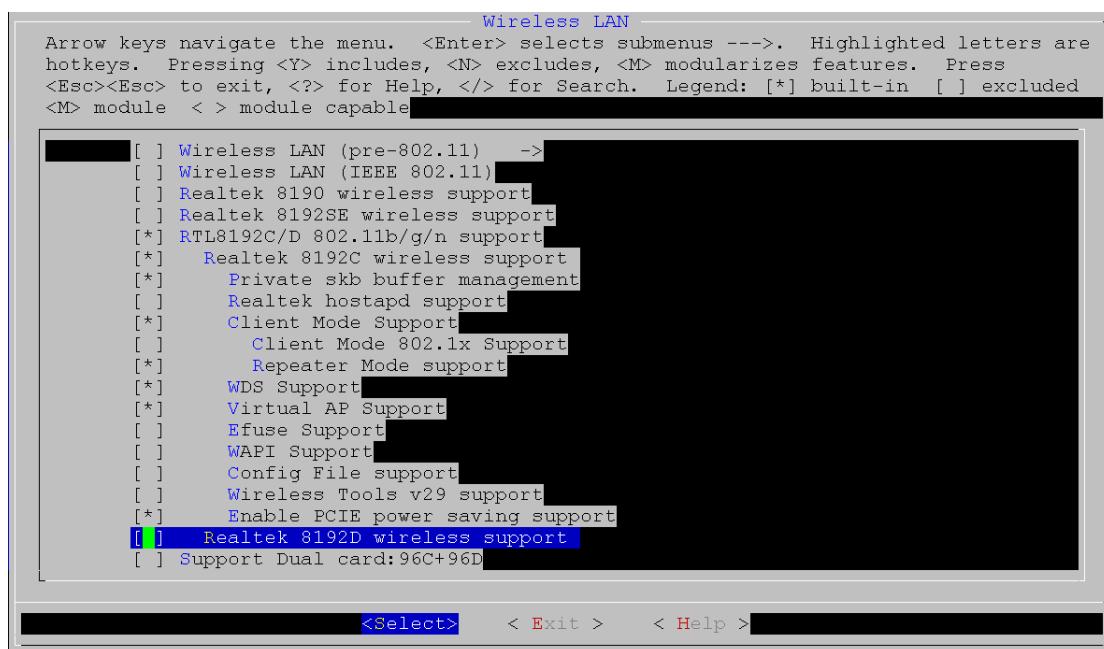
2) Enable 92D Driver

Config kernel --->

Device Drivers --->

Network device support --->

Wireless LAN --->



Selected “Realtek 8192D wireless support”

```

----- Wireless LAN -----
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are
hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press
<Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ] excluded
<M> module < > module capable

[ ] Wireless LAN (pre-802.11) ->
[ ] Wireless LAN (IEEE 802.11)
[ ] Realtek 8190 wireless support
[ ] Realtek 8192SE wireless support
[*] RTL8192C/D 802.11b/g/n support
[*] Realtek 8192D wireless support
[*] Private skb buffer management
[*] RTL8192D dual-MAC-dual-PHY mode (NEW)
[*] Virtual AP Support
[*] Client Mode Support
[!] WDS Support
[ ] Efuse Support
[ ] Smart Concurrent Support (NEW)
[ ] Support Dual card:96C+96D

[Select] < Exit > < Help >

```

If “RTL8192D dual-MAC-dual-PHY mode” is selected, only supports 92Dvc:

```

----- Wireless LAN -----
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are
hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press
<Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ] excluded
<M> module < > module capable

[ ] Wireless LAN (pre-802.11) ->
[ ] Wireless LAN (IEEE 802.11)
[ ] Realtek 8190 wireless support
[ ] Realtek 8192SE wireless support
[*] RTL8192C/D 802.11b/g/n support
[*] Realtek 8192D wireless support
[*] Private skb buffer management
[*] RTL8192D dual-MAC-dual-PHY mode (NEW)
[*] Virtual AP Support
[*] Client Mode Support
[!] WDS Support
[ ] Efuse Support
[ ] Smart Concurrent Support (NEW)
[ ] Support Dual card:96C+96D

[Select] < Exit > < Help >

```

If “RTL8192D dual-MAC-dual-PHY mode” is canceled, supports both 92Dvs and 92Dvc:

```

----- Wireless LAN -----
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are
hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press
<Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ] excluded
<M> module < > module capable

[ ] Wireless LAN (pre-802.11)
[ ] Wireless LAN (IEEE 802.11) [REDACTED]
[ ] Realtek 8190 wireless support
[ ] Realtek 8192SE wireless support
[*] RTL8192C/D 802.11b/g/n support
[*] Realtek 8192D wireless support
[*] Private skb buffer management
[*] RTL8192D dual-MAC-dual-PHY mode [SELECTED]
[Virtual AP Support]
[*] Client Mode Support
[ ] WDS Support
[ ] Efuse Support
[ ] Smart Concurrent Support (NEW)
[ ] Use USB Power (NEW)
[ ] Support Dual card: 92C+92D

----- Select ----- Exit ----- Help -----

```

Save and Make.

4.18.3 92D support for RTL8198 SDK

Enable 92D Driver

Config kernel --->

Device Drivers --->

Network device support --->

Wireless LAN --->

Enable 92D:

```

----- Wireless LAN -----
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are
hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press
<Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ] excluded
<M> module < > module capable

[ ] Wireless LAN (pre-802.11)
[ ] Wireless LAN (IEEE 802.11) [REDACTED]
[ ] Realtek 8190 wireless support
[ ] Realtek 8192SE wireless support
[*] RTL8192C/D 802.11b/g/n support
[*] 8198 clock source at 40Mhz
[*] Realtek 8192D wireless support [SELECTED]
[Private skb buffer management]
[*] RTL8192D dual-MAC-dual-PHY mode (NEW)
[*] Virtual AP Support
[*] Client Mode Support
[*] WDS Support
[ ] Efuse Support
[ ] Smart Concurrent Support (NEW)
[ ] Support Dual card: 96C+96D

----- Select ----- Exit ----- Help -----

```

If “RTL8192D dual-MAC-dual-PHY mode” is selected, only supports 92Dvc:

```

----- Wireless LAN -----
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are
hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press
<Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ] excluded
<M> module < > module capable

[ ] Wireless LAN (pre-802.11) ->
[ ] Wireless LAN (IEEE 802.11)
[ ] Realtek 8190 wireless support
[ ] Realtek 8192SE wireless support
[*] RTL8192C/D 802.11b/g/n support
[*] Realtek 8192D wireless support
[*] Private skb buffer management
[*] RTL8192D dual-MAC-dual-PHY mode (NEW)
[*] Virtual AP Support
[*] Client Mode Support
[!] WDS Support
[ ] Efuse Support
[ ] Smart Concurrent Support (NEW)
[ ] Support Dual card:96C+96D

----- Select ----- Exit ----- Help -----

```

If “RTL8192D dual-MAC-dual-PHY mode” is canceled, supports both 92Dvs and 92Dvc:

```

----- Wireless LAN -----
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are
hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press
<Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ] excluded
<M> module < > module capable

[ ] Wireless LAN (pre-802.11)
[ ] Wireless LAN (IEEE 802.11)
[ ] Realtek 8190 wireless support
[ ] Realtek 8192SE wireless support
[*] RTL8192C/D 802.11b/g/n support
[*] Realtek 8192D wireless support
[*] Private skb buffer management
[!] RTL8192D dual-MAC-dual-PHY mode
[*] Virtual AP Support
[*] Client Mode Support
[ ] WDS Support
[ ] Efuse Support
[ ] Smart Concurrent Support (NEW)
[ ] Use USB Power (NEW)
[ ] Support Dual card:92C+92D

----- Select ----- Exit ----- Help -----

```

Save and make.

4.18.4 92C support for RTL8198 SDK

Enable 92C Driver

Config kernel --->

Device Drivers --->

Network device support --->

Wireless LAN --->

Enable 92C:

```

----- Wireless LAN -----
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are
hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press
<Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ] excluded
<M> module < > module capable

[ ] Wireless LAN (pre-802.11)
[ ] Wireless LAN (IEEE 802.11)
[ ] Realtek 8190 wireless support
[ ] Realtek 8192SE wireless support
[*] RTL8192C/D 802.11b/g/n support
[*] 8198 clock source at 40Mhz
[*] Realtek 8192C wireless support
[*] Private skb buffer management
[ ] Realtek hostapd support
[*] Client Mode Support
[ ] Client Mode 802.1x Support
[*] Repeater Mode support
[*] WDS Support
[*] Virtual AP Support
[ ] Efuse Support
[ ] WAPI Support
[ ] Config File support
[ ] Wireless Tools v29 support
[ ] Enable PCIE power saving support
[ ] Enable both of the 2 pcie slot for bi-8192C support

v (+)

<Select> < Exit > < Help >

```

Save and make.

4.18.5 92C and 92D support for RTL8198 SDK

Enable 92D Driver and 92C Driver

Config kernel --->

Device Drivers --->

Network device support --->

Wireless LAN --->

Select “Support Dual card: 92C+92D”

```

----- Wireless LAN -----
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are
hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press
<Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ] excluded
<M> module < > module capable

^(-)
[ ] Realtek 8192SE wireless support
[*] RTL8192C/D 802.11b/g/n support
[*] 8198 clock source at 40Mhz
-**- Realtek 8192C wireless support
[*] Private skb buffer management
[ ] Realtek hostapd support
[*] Client Mode Support
[ ] Client Mode 802.1x Support
[*] Repeater Mode support
[ ] Efuse Support
[ ] Config File support
[ ] Wireless Tools v29 support
-**- Realtek 8192D wireless support
[*] Private skb buffer management
[*] Virtual AP Support
[*] Client Mode Support
[*] WDS Support
[ ] Efuse Support
[ ] Smart Concurrent Support
[*] Support Dual card:92C+92D

<Select> < Exit > < Help >

```

This configuration can support both 92Dvs and 92Dvc.

Save and make.

4.19 GDB server support

4.19.1 Enable GDB server

```
make users_menuconfig // To configure application settings
```

Menuconfig:

```
[*] gdbserver // selected
```

```
RLX Linux Configuration
the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, <M> module < > module capable

( - )
[ ] dnrd
[*] firmware write daemon
[ ] FT Daemon support
[!] gdbserver
[ ] gproxy
[ ] hostapd 2.0
[ ] i...
```

4.19.2 Use GDB server

4.19.2.1 Debugging an Already-running Process (over network)

1) remote step

```
gdbserver <host ip: port> --attach <pid of debugged process>
```

2) host step

- ① export PATH=YOUR_PATH/rxlxlinux-v2.2/users/gdb/gdb-host/bin/:\$PATH
- ② cd <source file dir of debugged program>
- ③ mips-linux-gdb <debugged program> /*enter gdb*/
- ④ set solib-search-path <lib path> /* Set the search path for loading non-absolute shared library symbol files. */
- ⑤ target remote <remote ip:port>

4.19.2.2 Starting and debugging your program (over network)

1) remote step

```
gdbserver <host ip: port> <debugged program>
```

2) host step

- ① export PATH=YOUR_PATH/rxlxlinux-v2.2/users/gdb/gdb-host/bin/:\$PATH
- ② cd <source file dir of debugged program>
- ③ mips-linux-gdb <debugged program> /*enter gdb*/
- ④ set solib-search-path <lib path> /* Set the search path for loading non-absolute shared library symbol files. */

⑤ target remote <remote ip:port>

4.19.2.3 Note for GDB server usage

On the host machine, GDB should compiled as target = mips-linux.

Host should ping remote succeed.

On the GDB host machine, an un-stripped copy of the debugging program is needed, since GDB needs the symbol and debugging information.

GDB can communicate with the target (GDB server) over a serial line, or over an IP network using TCP or UDP.

At present gdbserver6.8 is used.

4.20 HTTP file server support

4.20.1 HTTP File Server Introduction

HTTP File Server is used to Brower/upload/Delete files on USB storage device which connected to Realtek RTL819x Wireless Home Gateway.

4.20.2 Configuration for HTTP File Server

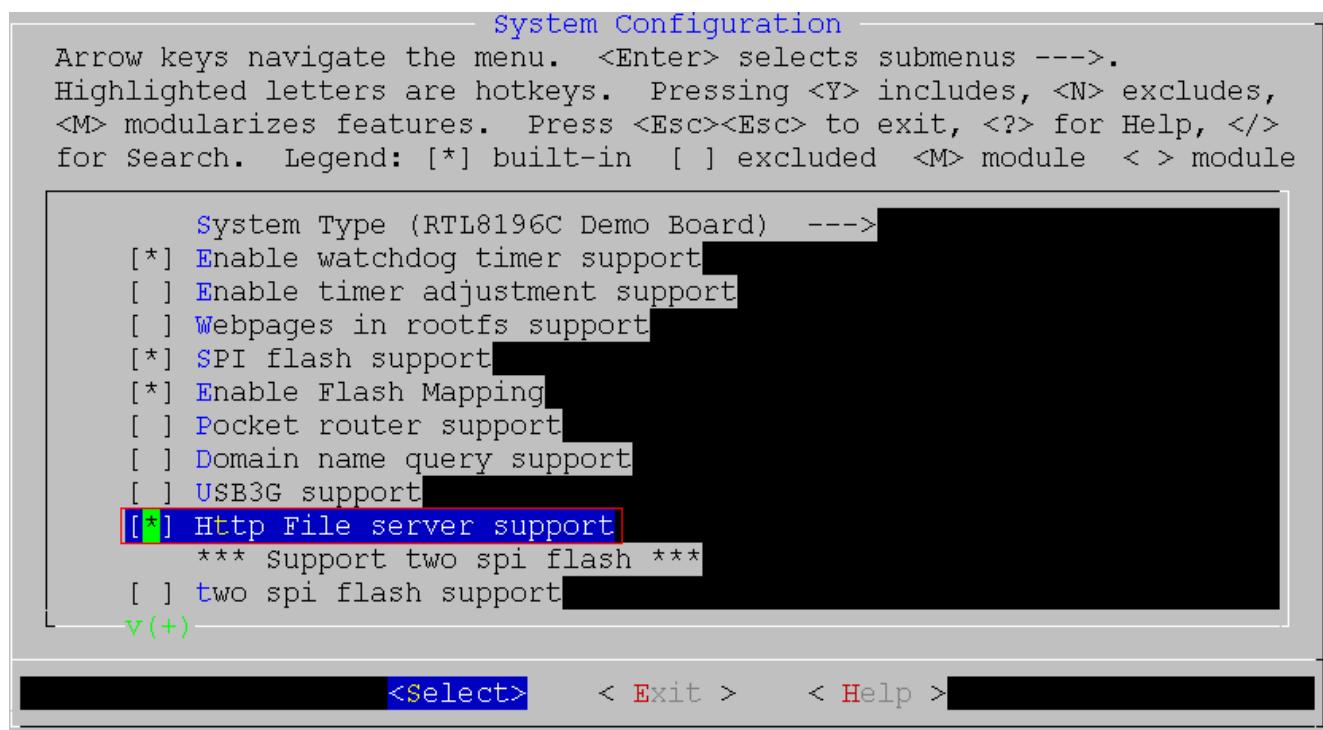
1) Kernel configure for HTTP File Server

make linux_menuconfig // To configure kernel settings

Menuconfig:

System Configuration --->

[*] Http File server support // Selected



General setup --->

[*] Support for hot-pluggable devices // Selected

```
[*] Prompt for development and/or incomplete code/drivers
() Local version - append to kernel release
[ ] Automatically append version information to the version string
[ ] Support for paging of anonymous memory (swap)
[ ] System V IPC
[ ] POSIX Message Queues
[ ] BSD Process Accounting
[ ] Export task/process statistics through netlink (EXPERIMENTAL)
[ ] Auditing support
RCU Subsystem --->
[ ] Kernel .config support
(12) Kernel log buffer size (16 => 64KB, 17 => 128KB)
[ ] Group CPU scheduler
[ ] Control Group support --->
[ ] Create deprecated sysfs layout for older userspace tools
[ ] Kernel->user space relay support (formerly relayfs)
[ ] Namespaces support
[ ] Initial RAM filesystem and RAM disk (initramfs/initrd) support
[*] Optimize for size
--*- Configure standard kernel features (for small systems) --->
[*] Strip assembler-generated symbols during link
[*] Support for hot-pluggable devices
[*] Enable support for printk log
[ ] Enable support for printk console
-->/etc
```

Device Drivers --->

SCSI device support --->

[*] SCSI device support // Selected

[*] legacy /proc/scsi/ support (NEW) // Selected

[*] SCSI disk support // Selected

[*] SCSI low-level drivers (NEW) ---> // Selected

```

SCSI device support
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys.
Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?>
for Help, </> for Search. Legend: [*] built-in [ ] excluded <M> module capable

[ ] RAID Transport Class
[*] SCSI device support
[ ] SCSI target support (NEW)
[*] legacy /proc/scsi/ support (NEW)
*** SCSI support type (disk, tape, CD-ROM) ***
[*] SCSI disk support
[ ] SCSI tape support (NEW)
[ ] SCSI OnStream SC-x0 tape support (NEW)
[ ] SCSI CDROM support (NEW)
[ ] SCSI generic support (NEW)
[ ] SCSI media changer support (NEW)
*** Some SCSI devices (e.g. CD jukebox) support multiple LUNs ***
[ ] Probe all LUNS on each SCSI device (NEW)
[ ] Verbose SCSI error reporting (kernel size +=12K) (NEW)
[ ] SCSI logging facility (NEW)
[ ] Asynchronous SCSI scanning (NEW)
  SCSI Transports --->
[*] SCSI low-level drivers (NEW) --->
[ ] SCSI Device Handlers (NEW) --->
[ ] OSD-Initiator library (NEW)

<Select>    < Exit >    < Help >

```

Device Drivers --->

[*] USB support --->

- [*] Support for Host-side USB // Selected
- [*] USB device class-devices (DEPRECATED) (NEW) // Selected
- [*] USB Monitor (NEW) // Selected
- [*] EHCI HCD (USB 2.0) support // Selected
- [*] OHCI HCD support // Selected
- [*] USB Mass Storage support // Selected

```

USB support
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys.
Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help,
</> for Search. Legend: [*] built-in [ ] excluded <M> module capable
```

-- USB support

- [*] Support for Host-side USB
- [] USB verbose debug messages (NEW)
- [] USB announce new devices (NEW)
 - *** Miscellaneous USB options ***
- [] USB device filesystem (NEW)
- [*] USB device class-devices (DEPRECATED) (NEW)
 - [] Dynamic USB minor allocation (NEW)
 - [] Rely on OTG Targeted Peripherals List (NEW)
 - [] Disable external hubs (NEW)
- [*] USB Monitor (NEW)
 - [] Enable Wireless USB extensions (EXPERIMENTAL) (NEW)
 - [] Support WUSB Cable Based Association (CBA) (NEW)
 - *** USB Host Controller Drivers ***
 - [] Cypress C67x00 HCD support (NEW)
- [*] EHCI HCD (USB 2.0) support
 - [] Root Hub Transaction Translators (NEW)
 - [] Improved Transaction Translator scheduling (EXPERIMENTAL) (NEW)
 - [] OXU210HP HCD support (NEW)
 - [] ISP116X HCD support (NEW)
 - [] ISP 1760 HCD support (NEW)
- [*] OHCI HCD support
 - [] SL811HS HCD support (NEW)
 - [] R8A66597 HCD support (NEW)
 - [] Host Wire Adapter (HWA) driver (EXPERIMENTAL) (NEW)

v (+)

<Select> < Exit > < Help >

```

USB support
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys.
Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help,
</> for Search. Legend: [*] built-in [ ] excluded <M> module capable
```

-- USB support

- ^ (-)
 - [] R8A66597 HCD support (NEW)
 - [] Host Wire Adapter (HWA) driver (EXPERIMENTAL) (NEW)
 - *** USB Device Class drivers ***
 - [] USB Modem (CDC ACM) support (NEW)
 - [] USB Printer support (NEW)
 - [] USB Wireless Device Management support (NEW)
 - [] USB Test and Measurement Class support (NEW)
 - *** NOTE: USB STORAGE depends on SCSI but BLK_DEV_SD may ***
 - *** also be needed; see USB STORAGE Help for more info ***
- [*] USB Mass Storage support
 - [] USB Mass Storage verbose debug (NEW)
 - [] Datafab Compact Flash Reader support (NEW)
 - [] Freecom USB/ATAPI Bridge support (NEW)
 - [] ISD-200 USB/ATA Bridge support (NEW)
 - [] USBAT/USBAT02-based storage support (NEW)
 - [] SanDisk SDDR-09 (and other SmartMedia, including DPCM) support (NEW)
 - [] SanDisk SDDR-55 SmartMedia support (NEW)
 - [] Lexar Jumpshot Compact Flash Reader (NEW)
 - [] Olympus MAUSEB-10/Fuji DPC-R1 support (NEW)
 - [] Support for Rio Karma music player (NEW)
 - [] SAT emulation on Cypress USB/ATA Bridge with ATACB (NEW)
 - [] The shared table of common (or usual) storage devices (NEW)
 - *** USB Imaging devices ***
 - [] USB Mustek MDC800 Digital Camera support (NEW)
 - [] Microtek X6USB scanner support (NEW)

v (+)

<Select> < Exit > < Help >

File systems --->

- [*] Enable POSIX file locking API // Selected
- [*] FUSE (Filesystem in Userspace) support // Selected

File systems

Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [] excluded <M> module < > module capable

```

[ ] Second extended fs support
[ ] Ext3 journaling file system support
[ ] The Extended 4 (ext4) filesystem
[ ] Reiserfs support
[ ] JFS filesystem support
[ ] XFS filesystem support
[ ] OCFS2 file system support
[ ] Btrfs filesystem (EXPERIMENTAL) Unstable disk format
[*] Enable POSIX file locking API
[ ] Dnotify support
[ ] Inotify file change notification support
[ ] Quota support
[ ] Kernel automounter support
[ ] Kernel automounter version 4 support (also supports v3)
[*] FUSE (Filesystem in Userspace) support
    Caches --->
    CD-ROM/DVD Filesystems --->
    DOS/FAT/NT Filesystems --->
    Pseudo filesystems --->
[*] Miscellaneous filesystems --->
[ ] Network File Systems --->
v(+)

```

<Select> < Exit > < Help >

File systems --->

DOS/FAT/NT Filesystems --->

- [*] MSDOS fs support // Selected
- [*] VFAT (Windows-95) fs support // Selected
- (437) Default codepage for FAT (NEW)
- (utf8) Default iocharset for FAT // Set Default iocharset for FAT is “utf8”
- [*] NTFS file system support // Selected, but Disable NTFS write support

DOS/FAT/NT Filesystems

Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [] excluded <M> module < > module capable

- [*] MSDOS fs support
- [*] VFAT (Windows-95) fs support
- (437) Default codepage for FAT (NEW)
- (utf8) Default iocharset for FAT
- [*] NTFS file system support
- [] NTFS debugging support (NEW)
- [] NTFS write support (NEW)

<Select> < Exit > < Help >

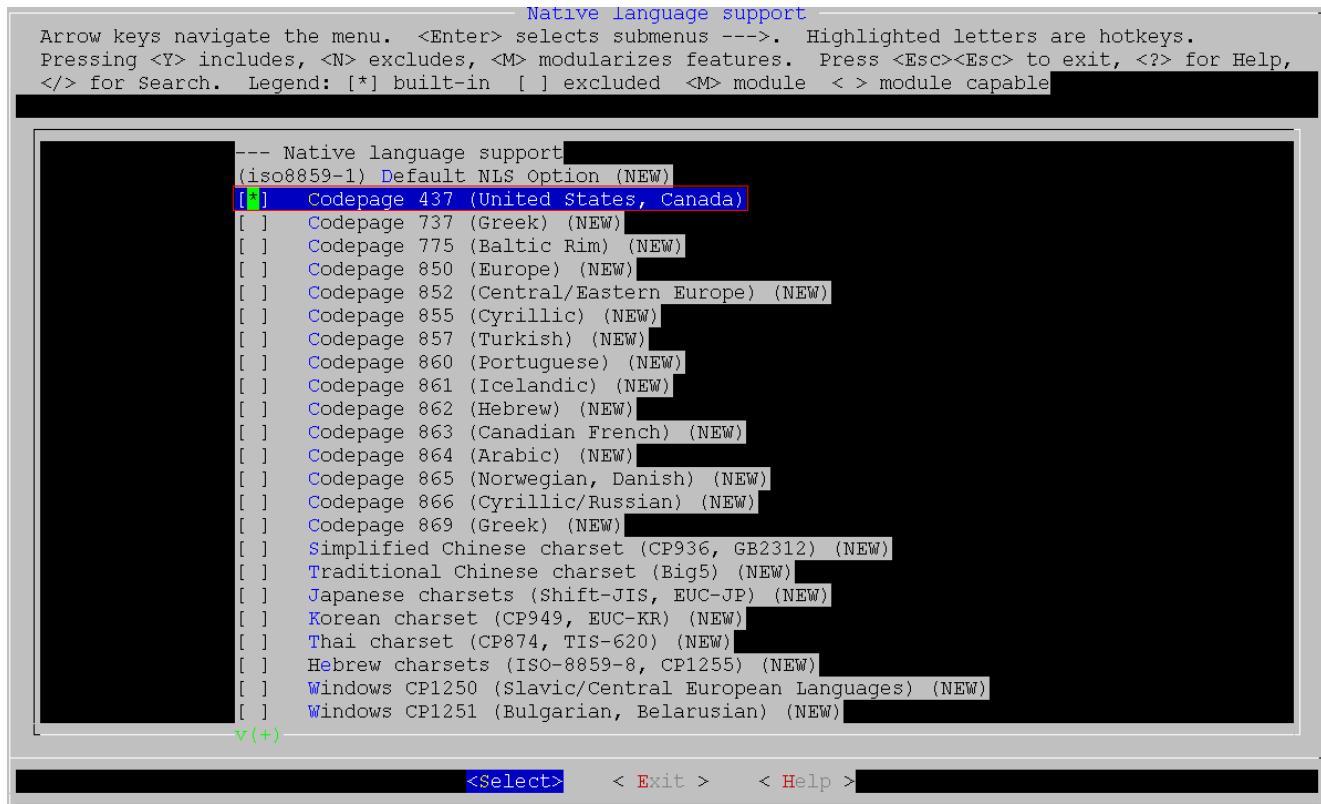
File systems --->

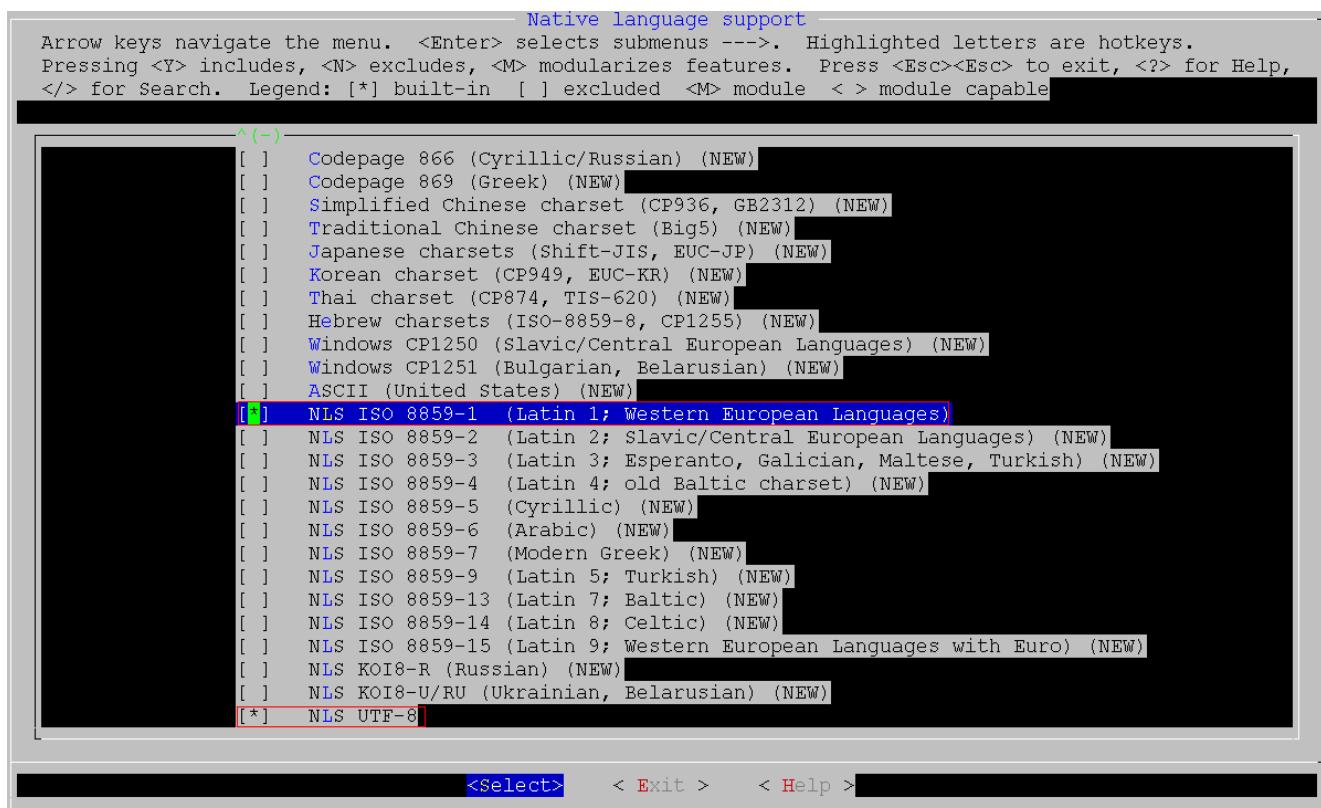
[*] Native language support --->

[*] Codepage 437 (United States, Canada) // Selected

[*] NLS ISO 8859-1 (Latin 1; Western European Languages) // Selected

[*] NLS UTF-8 // Selected



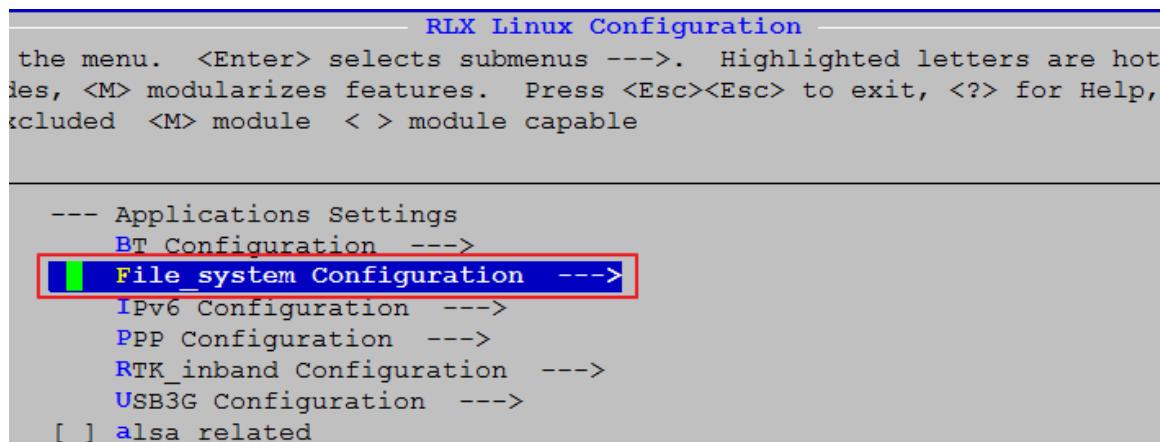


2) Application configure for HTTP File Server

make users_menuconfig // To configure application settings

Menuconfig:

[*] ntfs3g // selected



File system Configuration

```
the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, included <M> module < > module capable
```

- [] dosfsck
- [] e2fsprogs
- [] exfat-fuse
- [] mkdosfs
- [] NFS utility
- [*] ntfs3g**
- [] rtl_fs

[*] usbmount // selected

RLX Linux Configuration

```
the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, included <M> module < > module capable
```

— (-)

- [] pmc utils
- [] python support
- [] reload
- [] routed
- [*] rtk_cmd daemon
- [] RTK tcp2dev utility
- [] rtk voip
- [] RTL8686 IPC User Utility
- [*] samba support --->
- [] scripts
- [] simple config support
- [] snmp
- [] squashfs 4.0
- [] squashfs 4.2
- [] tr069
- [] udhcp
- [] ulinker autorun image tool
- [] updatedd DDNS
- [] upImage
- [*] usbmount**
- [] usbdisk update image

4.20.3 WEB GUI Manual

Home page:

Realtek uWiFi



Shared Partitions:

No shared partition available.

The Hyperlink “Settings” is for Router configuration, and if partition is available, the partition name will be showed in “Shared Partitions”

Realtek uWiFi



Shared Partitions:

[sda1](#)

User can enter the hyperlink in shared partition for file access.

User can click the hyperlink “Name”, “Last modified”, and “Size” for sort.

Index of /sda1

Name	Last modified	Size	
Parent Directory			
123/	28-Oct-2010 16:07:00	-	<input type="button" value="Remove"/>
abc/	28-Oct-2010 18:18:14	-	<input type="button" value="Remove"/>
1.txt	20-Oct-2010 15:05:14	25.68K	<input type="button" value="Remove"/>

Select File:

4.21 Hostapd support

Add hostapd support as follows:

Step 1, modify the related files.

- ① users/hostapd-0.6.10/hostapd/Makefile, comment lines as follows:

```
# CFLAGS += -DINBAND_CTRL  
# CFLAGS += -I../../inband_lib  
# LIBS += ../../inband_lib/inband.a
```

- ② users/script/cinit/init.sh, modified as follows:

```
# sysconf init $*  
  
echo 1 > /proc/rtk_vlan_support  
  
echo "hostapd test ... "  
  
brctl addbr br0  
  
brctl addif br0 eth0  
  
brctl addif br0 eth2
```

```
brctl addif br0 eth3  
brctl addif br0 eth4  
brctl addif br0 wlan0  
ifconfig br0 192.168.1.254  
ifconfig eth1 192.168.2.167  
ifconfig eth0 up  
ifconfig eth2 up  
ifconfig eth3 up  
ifconfig eth4 up  
ifconfig wlan0 hw ether 00:44:55:66:98:81  
cp /etc/hostapd.* /tmp
```

- ③ Copy net80211 folder (linux-2.6.30/include/net80211 or linux-3.10/include/net80211) under include the folder of toolchain, ex:
toolchain/rsdk-1.3.6-5281-EB-2.6.30-0.9.30/include/net80211.

Step 2, configure kernel and application.

- ① Select "Realtek hostapd support" at kernel menuconfig.

Device Drivers --->

 Network device support --->

 Wireless LAN --->

 [*] Realtek hostapd support

```

Wireless LAN
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys.
Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help,
</> for Search. Legend: [*] built-in [ ] excluded <M> module < > module capable

[ ] Wireless LAN (pre-802.11)
[ ] Wireless LAN (IEEE 802.11)
[ ] Realtek 8190 wireless support
[ ] Realtek 8192SE wireless support
[*] RTL8192C/D 802.11b/g/n support
[*] Realtek 8192C wireless support
[*] Private skb buffer management
[*] Realtek hostapd support
[*] Client Mode Support
[ ] Client Mode 802.1x Support
[*] Repeater Mode support
[*] WDS Support
[*] Virtual AP Support
[ ] Efuse Support
[ ] WAPI Support
[ ] Config File support
[ ] Wireless Tools v29 support
[*] Enable PCIE power saving support
[ ] Realtek 8192D wireless support
[ ] Support Dual card:96C+96D

<Select> < Exit > < Help >

```

- ② Select "openssl" & "hostapd in host controlled mode" and disable auth & wsc at users menuconfig.

```

[ ] auth
[*] openssl
    --- Choose openssl version.
    openssl (openssl-0.9.8b) --->
[*] hostapd in host controlled mode
[ ] wsc daemon

```

```

RLX Linux Configuration
the menu. <Enter> selects submenus --->. Highlighted letters are hot
les, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help,
cluded <M> module < > module capable

-- (-)
[ ] nfibi host
[ ] nmbs
[ ] ntpclient
[*] openssl
    openssl (openssl-0.9.8b) --->
[!] hostapd in host controlled mode
[ ] wapi_utils (NEW)
[ ] oray ddns

```

```

      RLX Linux Configuration
the menu. <Enter> selects submenus --->. Highlighted letters are hot-
des, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help,
xcluded <M> module < > module capable

--- Applications Settings
  BT Configuration --->
  File_system Configuration --->
  IPv6 Configuration --->
  PPP Configuration --->
  RTK_inband Configuration --->
  USB3G Configuration --->
[ ] alsafe related
[ ] auth
[ ] bluez
[ ] WiFi
[ ] vsftpd
[ ] wireless tools
[ ] wsc daemon
  Libraries Settings --->
  Debug & test Settings --->
---
Load an Alternate Configuration File
Save an Alternate Configuration File

<Select>  < Exit >  < Help >

```

Step 3, test hostapd.

① Test wpa2-psk

Firstly at AP console input command as follows:

```

killall hostapd
hostapd /etc/hostapd_psk.wpa2-ccmp-5g &

```

Secondly wireless client connect to our AP (SSID: hostapd_2_0-wpa2-ccmp) using wpa2-psk (password: 12345678).

② Test WPS push button

Firstly at AP console input command as follows:

```

killall hostapd
hostapd -B etc/hostapd_0.wps_upnp &
hostapd_cli wps_pbc

```

Secondly wireless client connect to our AP using WPS push button.

③ Test WPS pin

Firstly at AP console input command as follows:

```

killall hostapd

hostapd -B etc/hostapd_0.wps_upnp &
hostapd_cli wps_pin <uuid><sta_pin_code>
the uuid can be "any" or like format "63041253-1019-2006-1228-00D24C819881"

```

Secondly wireless client connect to our AP using WPS pin.

④ Test wpa2-802.1x

Firstly at AP console input command as follows:

```

killall hostapd

hostapd /tmp/hostapd.radius_wpa_test &

```

Secondly wireless client connect to our AP using wpa2-802.1x

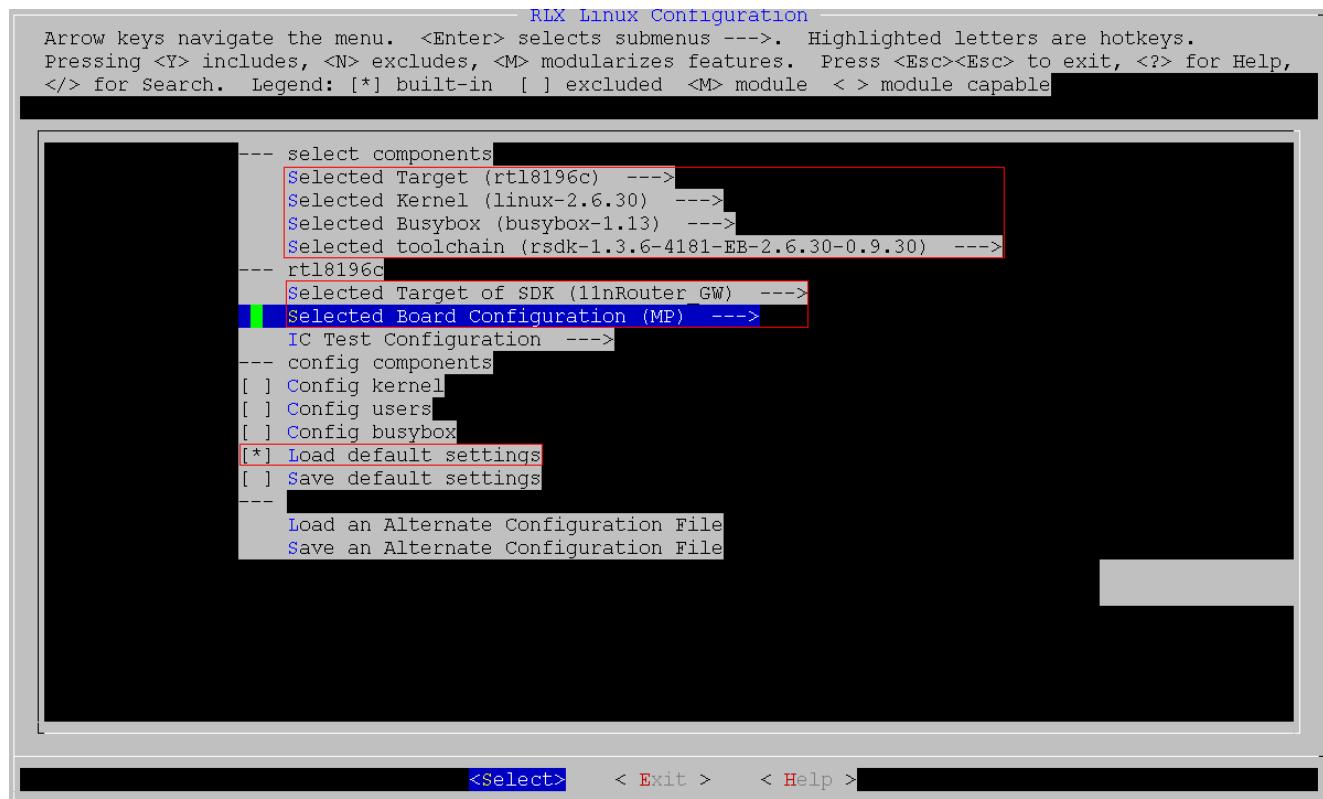
Step 4, customize hostapd.

If user want to customize hostapd, the configure file of hostapd at the folder of users/hostapd-0.6.10/hostapd/conf can be modified as user need.

4.22 MP support

4.22.1 RTL8196C MP support

For RTL8196C MP, uses “*make menuconfig*” to configure RTL8196C SDK settings as follows.



After configuring the settings as above, please refer to section 4.18 and configure the right

wireless driver.

4.22.2 RTL8198 MP support

For RTL8198 MP, uses “*make menuconfig*” to configure RTL8198 SDK settings as follows.

```
RLX Linux Configuration
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys.
Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help,
</> for Search. Legend: [*] built-in [ ] excluded <M> module < > module capable

--- select components
  Selected Target (rtl8198) --->
  Selected Kernel (linux-2.6.30) --->
  Selected Busybox (busybox-1.13) --->
  Selected toolchain (rsdk-1.3.6-5281-EB-2.6.30-0.9.30) --->
--- rtl8198
  Selected Board Configuration (MP) --->
    IC Test Configuration --->
--- config components
  [ ] Config kernel (NEW)
  [ ] Config users
  [ ] Config busybox
  [*] Load default settings
  [ ] Save default settings
--- 
  Load an Alternate Configuration File
  Save an Alternate Configuration File

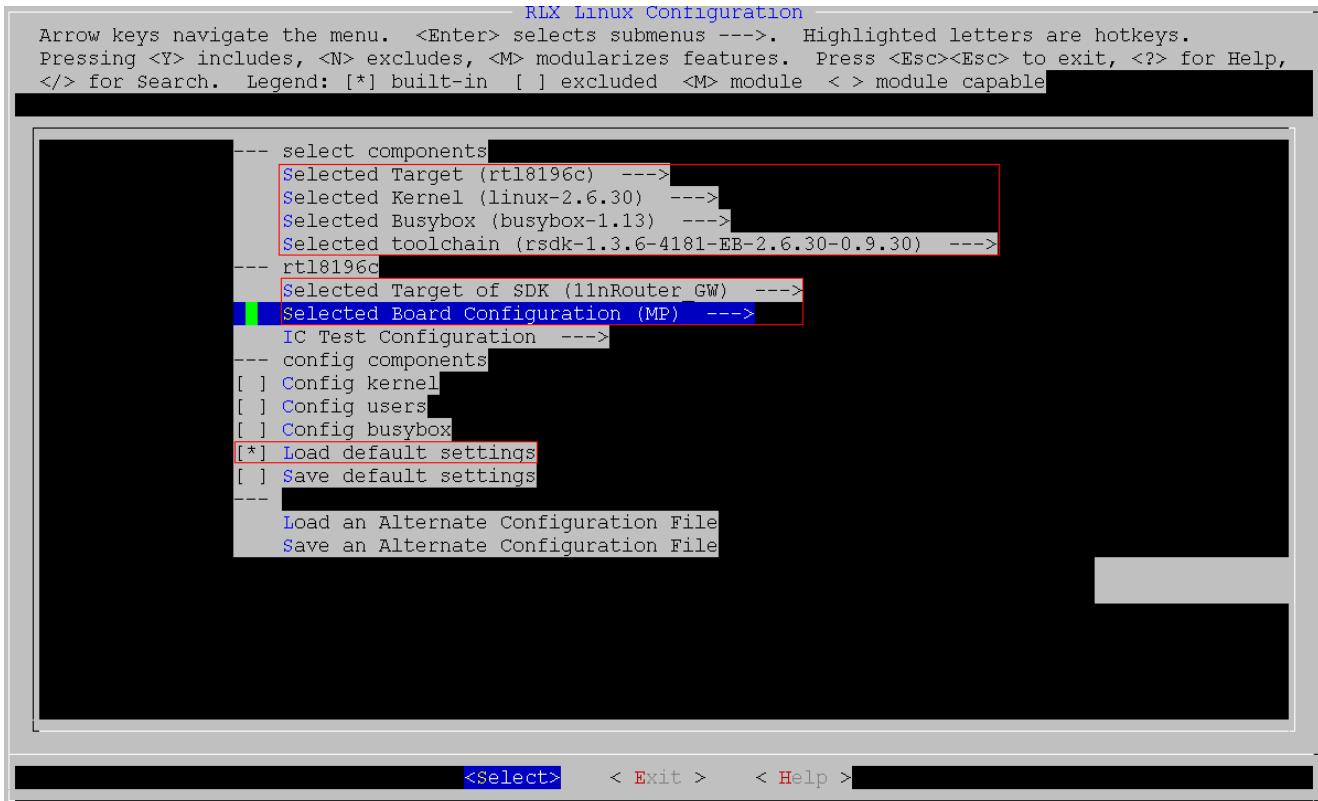
<Select>  < Exit >  < Help >
```

After configuring the settings as above, please refer to section 4.18 and configure the right wireless driver.

4.22.3 POCKET AP MP support

Since POCKET AP MP is based on RTL8196C MP, POCKET AP MP configure as follows:

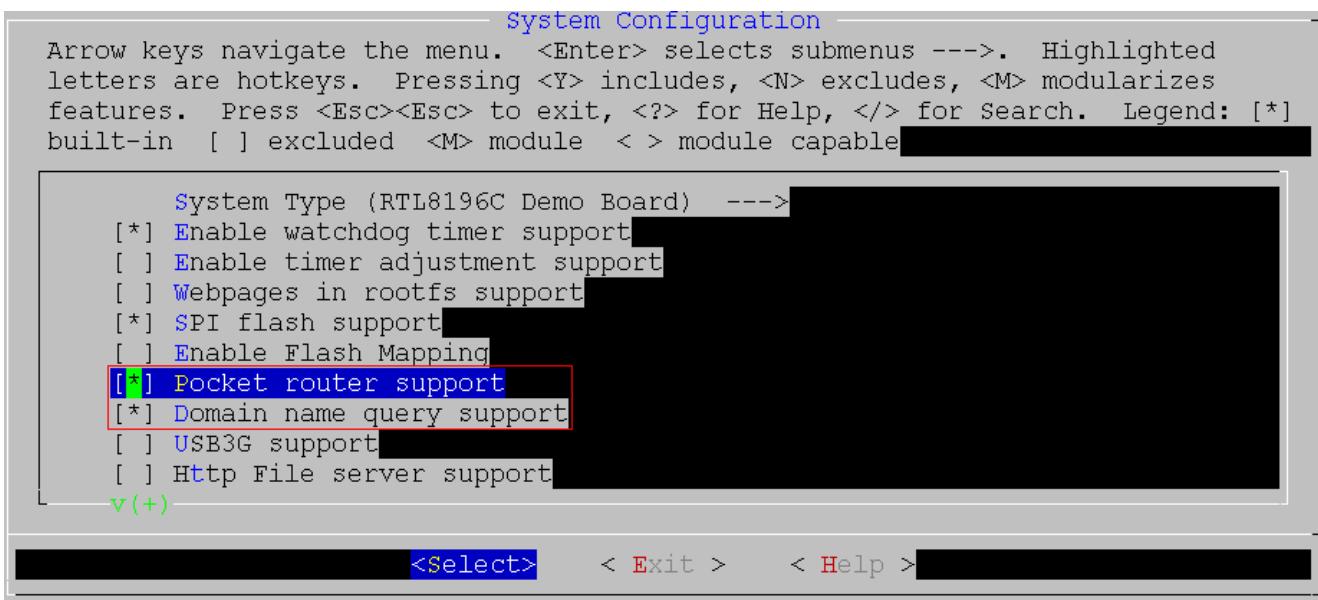
Step 1, use “*make menuconfig*” to configure SDK settings as follows.



Step 2, use “*make linux_menuconfig*” to config kernel settings as follows.

System Configuration --->

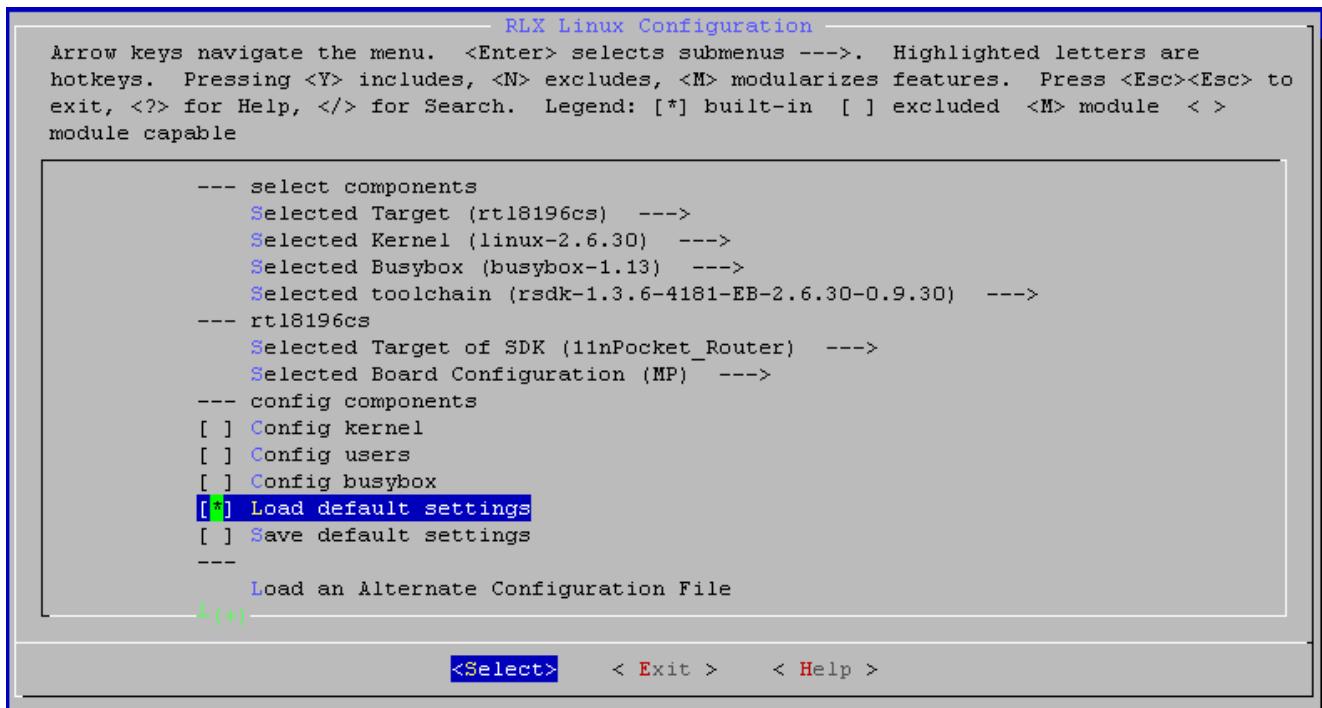
- [*] Pocket router support // Selected
- [*] Domain name query support // Selected



After configuring the settings as above, please refer to section 4.18 and configure the right wireless driver.

4.22.4 RTL8196CS (iNIC) MP support

For RTL8196CS (iNIC) MP, uses “*make menuconfig*” to configure RTL8198 SDK settings as follows.



```
RLX Linux Configuration
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ] excluded <M> module < > module capable

--- select components
    Selected Target (rtl8196cs) --->
    Selected Kernel (linux-2.6.30) --->
    Selected Busybox (busybox-1.13) --->
    Selected toolchain (rsdk-1.3.6-4181-EB-2.6.30-0.9.30) --->
--- rtl8196cs
    Selected Target of SDK (linnPocket_Router) --->
    Selected Board Configuration (MP) --->
--- config components
[ ] Config kernel
[ ] Config users
[ ] Config busybox
[*] Load default settings
[ ] Save default settings
--- 
    Load an Alternate Configuration File
L (+)

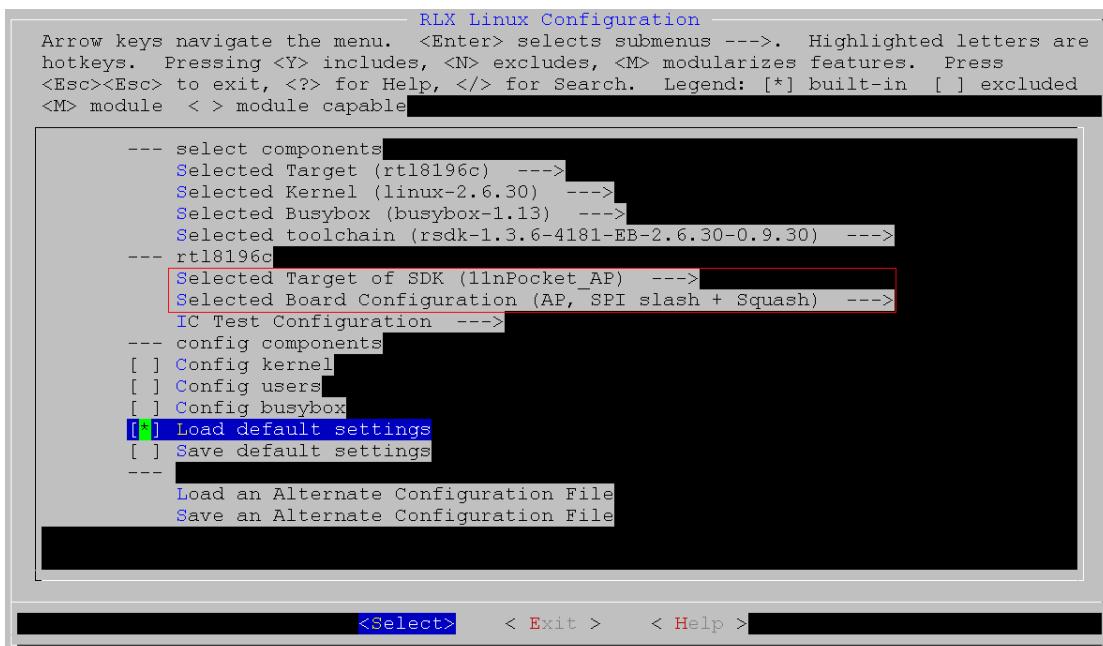
<Select> < Exit > < Help >
```

4.23 AP mode support

AP mode is that our platform is used as an access point rather than a router.

4.23.1 AP mode for Pocket AP SDK

To configure AP mode for Pocket AP SDK, please use “*make menuconfig*” to configure SDK settings as follows.



```
RLX Linux Configuration
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ] excluded <M> module < > module capable

--- select components
    Selected Target (rtl8196c) --->
    Selected Kernel (linux-2.6.30) --->
    Selected Busybox (busybox-1.13) --->
    Selected toolchain (rsdk-1.3.6-4181-EB-2.6.30-0.9.30) --->
--- rtl8196c
    Selected Target of SDK (linnPocket_AP) --->
    Selected Board Configuration (AP, SPI slash + Squash) --->
    IC Test Configuration --->
--- config components
[ ] Config kernel
[ ] Config users
[ ] Config busybox
[*] Load default settings
[ ] Save default settings
--- 
    Load an Alternate Configuration File
    Save an Alternate Configuration File

<Select> < Exit > < Help >
```

4.23.2 AP mode for RTL8198 SDK

To configure AP mode for RTL8198 SDK, please use “*make menuconfig*” to configure SDK settings as follows.

The screenshot shows the RLX Linux Configuration menu. At the top, there is a header with instructions for navigating the menu using arrow keys, enter, esc, and search. Below the header, the menu lists several configuration options:

- select components
 - Selected Target (rtl8198) --->
 - Selected Kernel (linux-2.6.30) --->
 - Selected Busybox (busybox-1.13) --->
 - Selected toolchain (rsdk-1.3.6-5281-EB-2.6.30-0.9.30) --->
- rtl8198
 - Selected Board Configuration (AP - SPI flash, Squashfs) ---> (highlighted with a red box)
 - IC Test Configuration --->
- config components
 - [] Config kernel
 - [] Config users
 - [] Config busybox
 - [] Load default settings (highlighted with a green box)
 - [] Save default settings
- - Load an Alternate Configuration File
 - Save an Alternate Configuration File

At the bottom of the menu, there are navigation keys: <Select>, <Exit>, and <Help>.

4.24 Wireless configuration file support

The wireless driver can be configured via a configuration file each time an interface is up.

Note: this method is different from the original method. The original method: applications use iwpriv to set MIBs to wireless driver. This method: after the configuration file is prepared by applications, the wireless driver will use the configuration file to set MIBs each time an interface is up.

1) Kernel configuration

Use “*make linux_menuconfig*” to config kernel settings as follows.

Device Drivers --->

[*] Network device support --->

Wireless LAN --->

[*] Config File support

Note: at present, wireless configuration file is only supported for RTL8192C.

```

----- Wireless LAN -----
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys.
Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help,
</> for Search. Legend: [*] built-in [ ] excluded <M> module <> module capable

[ ] Wireless LAN (pre-802.11)
[ ] Wireless LAN (IEEE 802.11) [ ]
[*] Realtek 8190 wireless support [ ]
[*] Realtek 8192SE wireless support [ ]
[*] RTL8192C/D 802.11b/g/n support [ ]
[*] Realtek 8192C wireless support (NEW) [ ]
[*] Private skb buffer management [ ]
[*] Realtek hostapd support (NEW) [ ]
[*] Client Mode Support [ ]
    Client Mode 802.1x Support (NEW) [ ]
[*] Repeater Mode support (NEW) [ ]
[*] WDS Support [ ]
[*] Virtual AP Support [ ]
[*] Efuse Support [ ]
[*] WAPI Support [ ]
[*] support local AS [ ]
[*] Config File support [ ]
[*] Wireless Tools v29 support (NEW) [ ]
[*] Enable PCIE power saving support (NEW) [ ]
[*] Enable external high power PA (NEW) [ ]
[*] Enable external LNA (NEW) [ ]
[*] Enable both of the 2 pcie slot for bi-8192C support (NEW) [ ]
[*] Realtek 8192D wireless support [ ]
[*] Support Dual card:92C+92D [ ]

-----> <Select> < Exit > < Help >

```

2) Configuration file

Path: /etc/Wireless/RTL8192CD.dat

Syntax: 'wlan_interface'_'mib_command', e.g. wlan0_ssid=xxxx.

Notes:

- ① Add '#' in front of comment lines.
- ② Space is NOT allowed between 'wlan_interface' and 'mib_command'.
- ③ If the user needs to configure MIB values with special characters, e.g. '#', the value of 'mib_command' MUST be quoted E.g. wlan0_ssid="#XXXXXX@##\$\$%%"
- ④ 'wlan_interface': wlan interface, e.g., wlan0, wlan0-va0. However, please DO NOT configure WDS interfaces because WDS is configured in wlan0 interface.
- ⑤ 'mib_command': MIB commands, e.g., ssid=xxxx, please refer to table "MIB command table" and table "Extended MIB command table".
- ⑥ MIB value should be also configured for each virtual interface separately.
- ⑦ Each time an interface is up, the configuration file will be loaded.

4.25 WPS under wireless configuration file support

For WPS, please refer to "Realtek_WPS_user_guide_V1.3.pdf" for detail explanation and usages. In this section, it's described that how to use WPS when wireless configuration file is supported. For wireless configuration file support, please refer to section 2.24.

1) WPS daemon (wscd)

When the wireless configuration file is supported, wscd should also refer to the wireless

configuration file and the parameter of wscd should be generated based on the wireless configuration file. Please refer to “Realtek_WPS_user_guide_V1.3.pdf” for detailed information of WPS parameters.

For the command line of wscd, one example is as follows:

```
wscd -start -c /var/RTL8192CD.dat -w wlan0 -fi /var/wscd-wlan0 fifo -daemon
```

// /var/RTL8192CD.dat is the wireless configuration file

2) How to save wireless profile generated

When WPS changes from un-configured state to configured state, wscd will save the wireless profile to a temp file and the profile need to be written to FLASH.

3) How to trigger PBC method

By command line or GPIO interface as follows:

① wscd -sig_pbc

② wscd will read /proc/gpio in one second timer, if its value>0, wscd will do PBC method.

4) How to trigger PIN method

By command line as follows:

```
iwpriv wlan0 set_mib pin=xxxxxxxx
```

Note:

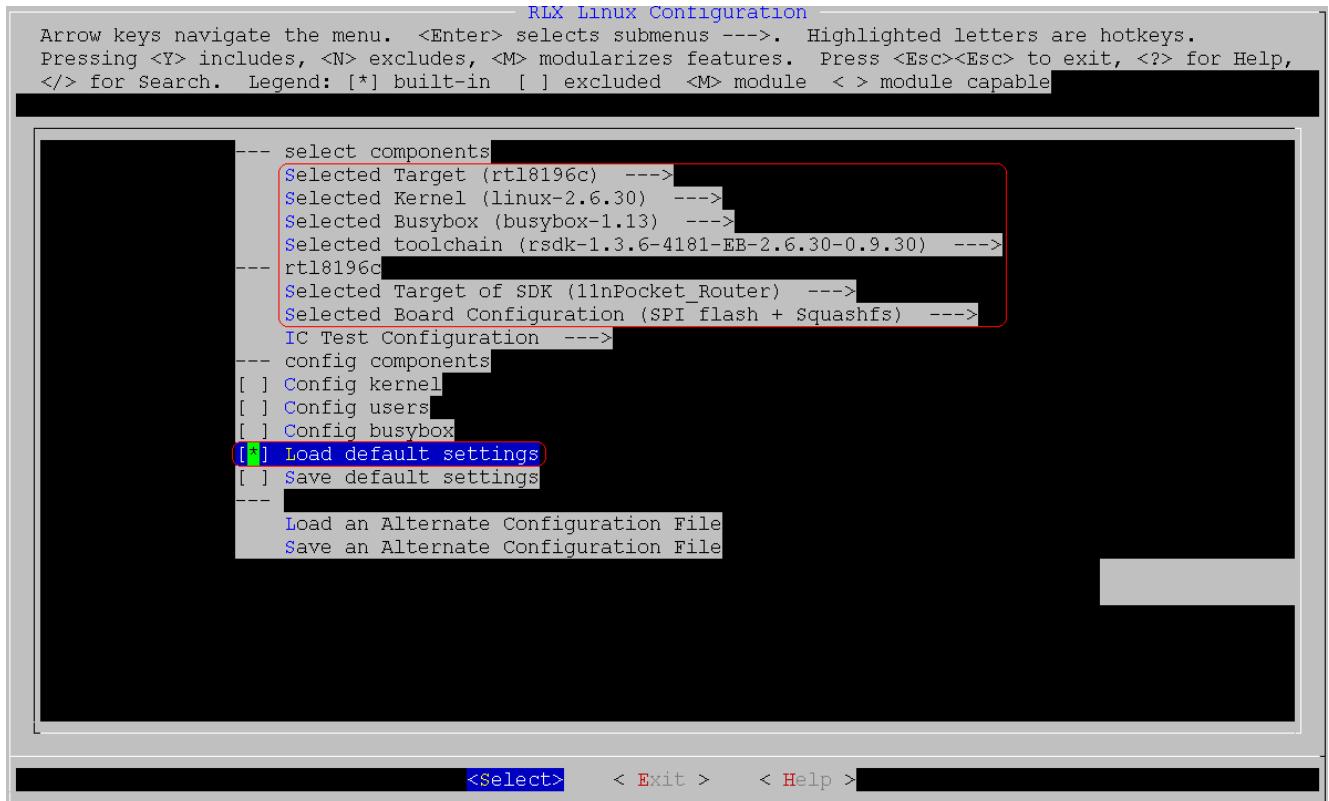
The iwcontrol tool is also needed for WPS. Please make sure the FIFO is created and opened in wscd, and iwcontrol use the same FIFO, otherwise wscd will hang.

4.26 Domain name query support

At present, the feature of domain name query is only supported for POCKET AP SDK.

Step 1, choose POCKET AP SDK first of all.

make menuconfig as follows.



Step 2, choose domain name query support.

make linux_menuconfig as follows:

System Configuration --->

[*] Domain name query support

```

System Configuration
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys.
Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help,
</> for Search. Legend: [*] built-in [ ] excluded <M> module <> module capable

System Type (RTL8196C Demo Board) --->
[*] Enable watchdog timer support
[ ] Enable timer adjustment support
[ ] Webpages in rootfs support
[*] SPI flash support
[*] Enable Flash Mapping
[*] Pocket router support
[ ] Pocket AP support
[*] Domain name query support
[ ] USB3G support
[ ] Http File server support
    *** Support two spi flash ***
[ ] two spi flash support
    *** Flash size 2M or 4M, default 2M ***
(0x200000) Size of Flash
    *** Hardware setting offset,should be 4K alignment ***
(0x6000) Hardware setting offset in flash.
    *** Default setting offset,should be 4K alignment. ***
    *** size of default and current setting should be same. ***
(0x8000) Default setting offset in flash.
    *** Current setting offset,should be 4K alignment. ***
(0xC000) Current setting offset in flash.
    *** Webpage image offset,should be 4K alignment. ***
    *** size of web page is normally about 100K. ***
(0x10000) webpages image offset in flash.

V(+)

```

<Select> < Exit > < Help >

Step 3, test domain name query.

- 1) Set Domain Name (for example: Realtek) at [LAN Interface] webpage of AP;
- 2) If AP works at AP mode, AP webpage can be accessed via <http://realtekap.com/home.asp>
- 3) If AP works at client mode, AP webpage can be accessed via <http://realtekcl.com/home.asp>

Realtek WLAN AP Webserver - Microsoft Internet Explorer

文件(F) 编辑(E) 查看(V) 收藏(A) 工具(T) 帮助(H)

后退() 前进() 停止() 搜索() 收藏夹() | 开始() 搜索() 转到() 链接()

地址() http://192.168.1.254/ Google

REALTEK WLAN Access Point

Site contents:

- Setup Wizard
- Operation Mode
- Wireless
 - BandMode
 - wlan1(5GHz)
 - Basic Settings
 - Advanced Setting
 - Security
 - WDS settings
 - Site Survey
 - WPS
 - Schedule
 - wlan2(2.4GHz)
 - TCP/IP Settings
 - LAN Interface
 - Management

LAN Interface Setup

This page is used to configure the parameters for local area network which connects to the LAN port of your Access Point. Here you may change the setting for IP addresss, subnet mask, DHCP, etc..

IP Address: 192.168.1.254

Subnet Mask: 255.255.255.0

Default Gateway: 0.0.0.0

DHCP: Auto

DHCP Client Range: 192.168.1.100 – 192.168.1.200

Static DHCP: Set Static DHCP

Domain Name: Realtek

802.1d Spanning Tree: Disabled

Clone MAC Address: 000000000000

Internet

Realtek WLAN AP Webserver - Microsoft Internet Explorer

文件(F) 编辑(E) 查看(V) 收藏(A) 工具(T) 帮助(H)

地址(0) http://realtekap.com/home.asp

Google

开始 | 搜索 | 收藏夹 | 转到 | 链接 »

REALTEK

WLAN Access Point

LAN Interface Setup

This page is used to configure the parameters for local area network which connects to the LAN port of your Access Point. Here you may change the setting for IP address, subnet mask, DHCP, etc..

IP Address: 192.168.1.254

Subnet Mask: 255.255.255.0

Default Gateway: 0.0.0.0

DHCP: Auto

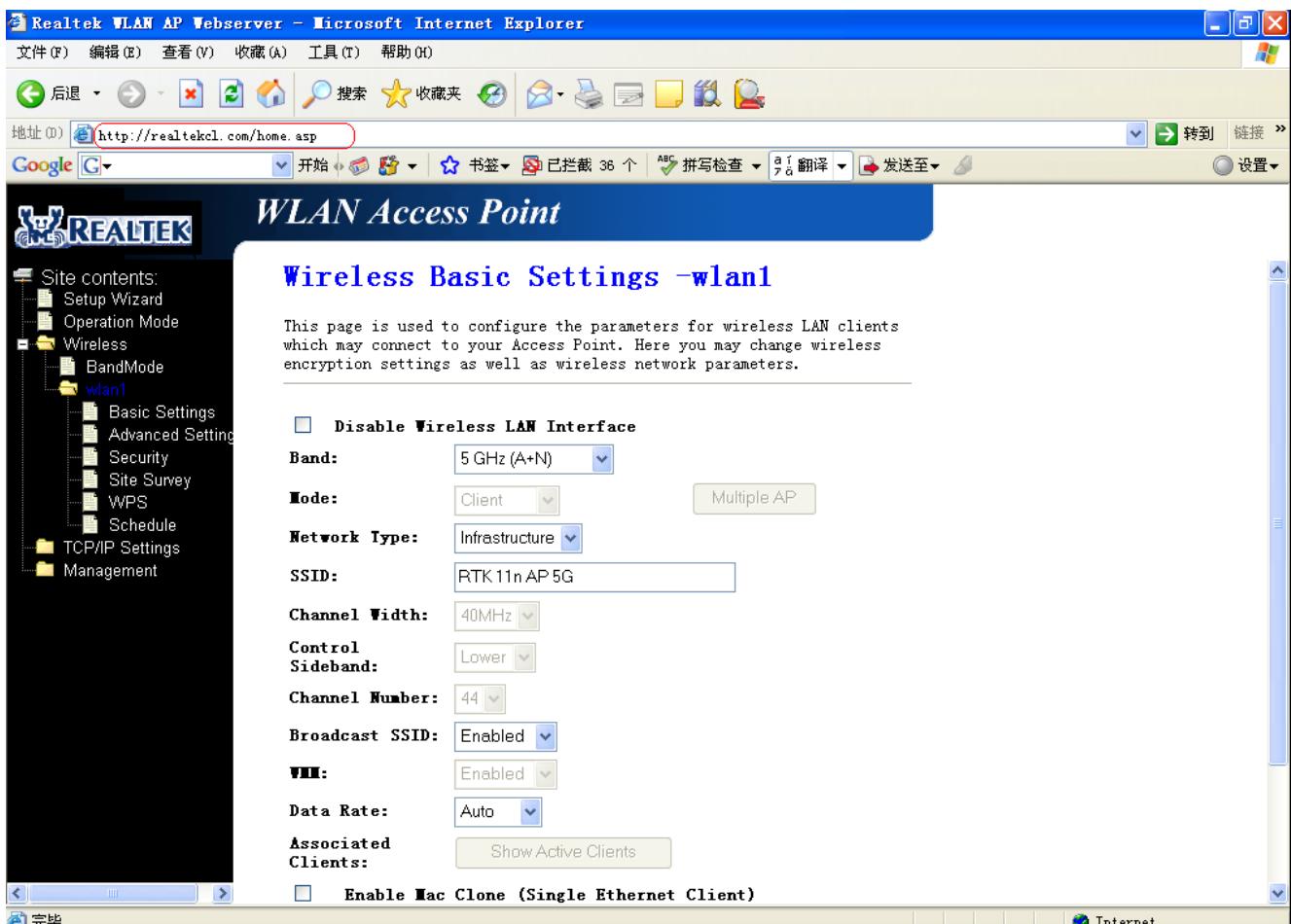
DHCP Client Range: 192.168.1.100 – 192.168.1.200

Static DHCP:

Domain Name: Realtek

802.1d Spanning Tree: Disabled

Clone MAC Address: 000000000000



4.27 rtk voip support

rtk(voip) is only available in RTL8954C. Please refer to VoIP SDK Manual for more detail.

4.28 Realtek Flash Dual Image support

This feature modify dual-image update algorithm in boot loader that it will copy bank2 image to bank1 when bank1 image is corrupted.

This feature support two kinds of mode, static mode and toggle mode. Static mode is the default dual image mode.

4.28.1 Dual image logistic

always boot from bank1 if it was not corrupt.

image will be backup from bank2 if bank1 was corrupt.

always write to bank1 in any type of user upgrading.

always write identical content to bank2 from bank1 if bank1 was fine.

4.28.2 Static mode

- Bootcode menuconfig

please run 'menuconfig' in boot loader root directory, and then select dual image support as follow:

[*] Support Flash DualBank

(Static) Mode

(400000) Second Bank offset

Please note, you may need modify Second Bank offset according to your hardware platform.

- **2nd bank image generate**

You must make sure the image in 2nd bank has been updated correctly before testing this function. You may use the following commands in boot loaderconsole to read flash from bank1, and write to bank2 in first time as:

```
flr 80800000 30000 3d0000
```

```
flw 430000 80800000 3d0000
```

Above commands is assumed the offset of Linux image is put at Flash 0x30000, and your the total flash size is 8M bytes (each bank size is 4M bytes).

4.28.3 Toggle mode

- **Bootcode menuconfig**

please run 'menuconfig' in boot loader root directory, and then select dual image support as follow:

[*] Support Flash DualBank

(Toggle) Mode

(400000) Second Bank offset

Please note, you may need modify Second Bank offset according to your hardware platform

- **Kernel menuconfig**

In order to enable the Dual Image support, you need to select the item "Enable Flash Dual Bank support" and enter offset of the second bank in the menuconfig. For example, you have a 8M flash and want to put the first 4M to bank1 and the last 4M to bank2, then you need to configure the linux menuconfig as follows:

System Configuration --->

[*]Enable Flash Dual Bank support

[*] Enable Flash Dual Bank Webpage Backup Support

*** Second Bank Offset ***

(0x400000) offset of Flash

[*] Enable Flash Mapping

*** Flash size 2M or 4M, default 2M ***

(0x400000) Size of Flash

*** Hardware setting offset should be 4K alignment ***

(0x6000) Hardware setting offset in flash.

Note: even though the real flash size is 8M, the item "Size of Flash" you need to enter is 4M
(Because the item means "Size of Bank" in Dual Image case.)

● Web pages

There are some options will be added in the "Upgrade Firmware" web page if you are running a "Dual Image" supported firmware. You can Enable/Disable Dual Image support or reboot from the backup bank in web page.

4.29 IEEE 802.3az EEE (Energy Efficient Ethernet) support

RTL8196C/RTL8198 support the IEEE 802.3az Energy Efficient Ethernet. This feature is enabled in RTL8196C and RTL8198 by default and can not be modified by end users. However, customers can disable this feature via linux menuconfig and the menuconfig is shown as follows:

make menuconfig:

-> Config kernel

-> Device Drivers

-> Network device support

-> Options for Realtek SoC

-> Disable 802.3az EEE feature

Verification:

The link partner needs also support EEE feature when does the verification. If such kind of link partner can not be found, the RTL8196C/RTL8198 can be used instead. Please connect a current meter to measure the DUT's current first. The DUT's current consumption will be different for EEE enabled image (current consumption value is lower) and EEE disable image (current consumption value is higher) when we plug in a RJ45 cable which connect to an EEE enabled link partner.

4.30 IGMP/MLD support

4.30.1 IGMP/MLD introduce

For IGMP support, note as follows:

IGMP proxy supports IGMP v1/v2/v3,

IGMP snooping supports IGMP v1/v2/v3,

MLD snooping supports MLD v1/v2,

IGMP/MLD snooping supports fast leave,

Support 128 hardware multicast entries.

IGMP proxy is independent with IGMP snooping, but hardware multicast depends on IGMP snooping.

4.30.2 How to enable IGMP proxy

make users_menuconfig //To configure applications

Note: if only “igmp proxy” is selected, IGMPv2 is supported; if both “igmp proxy” and “support igmpv3 proxy” are selected, IGMPv3 is supported.

```
RLX Linux Configuration
the menu. <Enter> selects submenus --->. Highlighted letters are hot
des, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help,
xcluded <M> module < > module capable

( - )
[*] firmware write daemon
[ ] FT Daemon support
[*] gdbserver
[ ] gproxy
[ ] hostapd 2.0
[ ] iapp
[*] igmp proxy
[*] support igmp proxy multiwan
[ *] support igmpv3 proxy
[ ] ioh
[ ] ioh
[ ] ioh
```

4.30.3 How to enable IGMP/MLD snooping

vim linux-2.6.30/drivers/net/rtl819x/Kconfig (or *linux-3.10/drivers/net/rtl819x/Kconfig*)

Default IGMP/MLD snooping is enabled.

```
config RTL_IGMP_SNOOPING
    bool
    default y

config RTL_MLD_SNOOPING
    bool
    default y
    depends on RTL_IGMP_SNOOPING
```

4.30.4 How to enable hardware multicast

make linux_menuconfig //To configure linux kernel

Linux kernel menu as follows:

Device Drivers --->

[*] Network device support --->

[*] Options for Realtek SoC --->

Config for Layered Driver Features --->

Hardware Features Selection (Enable RTL Hardware Multicast Only)

Or Hardware Features Selection (Enable RTL Hardware NAPT)

4.31 SNMP support

4.31.1 Applications configure

make users_menuconfig //To configure applications

```
RLX Linux Configuration
the menu. <Enter> selects submenus --->. Highlighted letters are hot
les, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help,
cluded <M> module < > module capable

( - )
[ ] pmc utils
[ ] python support
[ ] reload
[ ] routed
[*] rtk_cmd daemon
[ ] RTK tcp2dev utility
[ ] rtk	voip
[ ] RTL8686 IPC User Utility
[*] samba support --->
[ ] scripts
[ ] simple config support
[*] snmp
[ ] squashfs 4.0
[ ] squashfs 4.2
```

4.31.2 Kernel configure

1) *make linux_menuconfig //To configure linux kernel*

Linux kernel menu as follows:

General setup --->

[*] System V IPC

2) #define SUPPORT_SNMP_MIB at file drivers/net/wireless/rtl8192cd/8192cd_cfg.h

4.31.3 Webpage setting

After load image, please set SNMP in webpage to enable SNMP.

The screenshot shows the Realtek WLAN Access Point configuration interface. The top navigation bar is blue with the text "WLAN Access Point". On the left, there is a sidebar titled "Site contents:" with various menu items: Setup Wizard, Operation Mode, Wireless, TCP/IP Settings, Firewall, QoS, SNMP (which is selected and highlighted), and Management.

The main content area is titled "SNMP Setting". It contains a brief description: "SNMP is a application for network managment". Below this, there is a section with a checked checkbox labeled "Enable SNMP".

Name :	Realtek
Location :	AP
Contact :	Router
Read/Write Community :	private
Read-Only Community :	public

At the bottom of the form are two buttons: "Apply Change" and "Reset".

Note: The Read/Write community and Read-Only Community Should NOT be the same

4.31.4 SNMP test

Rebuild image, upload image, and use MIBBROWSER to test.

4.32 UVC support

To support UVC (USB VIDEO CLASS), **support USB first** (refer to section 4.2).

(1) If UVC is not supported, linux kernel configure for UVC as follows.

`make linux_menuconfig` // To configure linux kernel settings

Menuconfig:

Device Drivers --->

Multimedia devices --->

[] Video For Linux // Not selected

```

.config - Linux Kernel v2.6.30.9 Configuration

Multimedia devices
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys.
Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help,
</> for Search. Legend: [*] built-in [ ] excluded <M> module < > module capable

*** Multimedia core support ***
[!] Video For Linux
[ ] DVB for Linux
*** Multimedia drivers ***
[ ] DAB adapters

<Select> < Exit > < Help >

```

(2) If UTC is supported, linux kernel configure for UVC as follows.

make linux_menuconfig // To configure linux kernel settings

Menuconfig:

Device Drivers --->

Multimedia devices --->

- [*] Video For Linux // selected
- [*] Video capture adapters ---> // selected and enter
 - [*] V4L USB devices ---> // selected and enter
 - [*] USB Video Class (UVC) // selected

```

.config - Linux Kernel v2.6.30.9 Configuration

V4L USB devices
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys.
Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help,
</> for Search. Legend: [*] built-in [ ] excluded <M> module < > module capable

--- V4L USB devices
[*] USB Video Class (UVC)
[*] GSPCA based webcams --->
[*] Hauppauge HD PVR support
[*] USB 3com HomeConnect (aka vicam) support (EXPERIMENTAL)
[*] USB IBM (Xirlink) C-it Camera support
[*] USB Konica Webcam support
[*] USB Logitech Quickcam Messenger
[*] USB ET61X[12]51 PC Camera Controller support
[*] USB OV511 Camera support
[*] USB SE401 Camera support
[*] USB SN9Clxx PC Camera Controller support
[*] USB STV680 (Pencam) Camera support
[*] USB ZC0301[P] Image Processor and Control Chip support
[*] USB Philips Cameras
[*] USB Philips Cameras input events device support
v(+)

<Select> < Exit > < Help >

```

4.33 Telnetd and Login support in busybox

Enter the busybox menu, and then make menuconfig.

make menuconfig // To configure bootloader settings

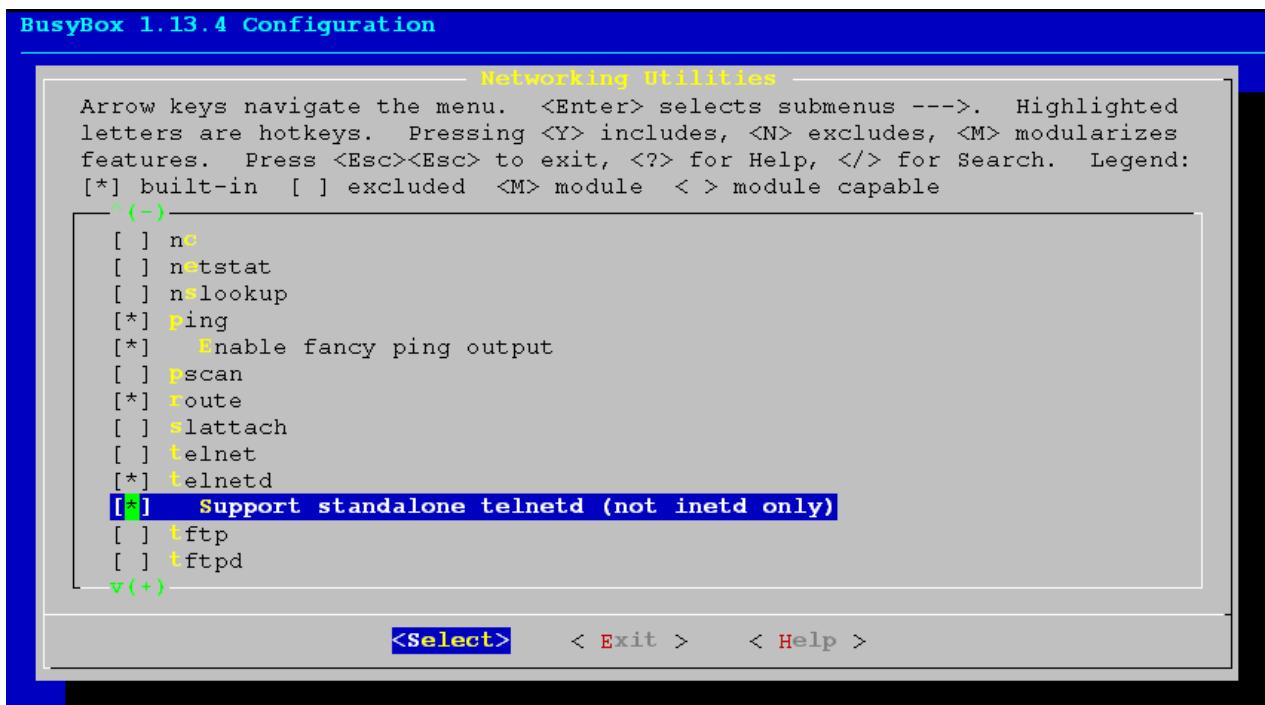
4.33.1 telnetd support menuconfig:

menuconfig:

Networking Utilities--->

[*] telnetd // selected

[*] Support standalone telnetd (not inetd only) // selected



```
## select items in menuconfig:  
make menuconfig  
-> Config busybox  
  -> Networking Utilities --->  
    [*] telnetd  
    [*] Support standalone telnetd (not inetd only)  
  -> Login/Password Management Utilities --->  
    [*] adduser  
    [*] login  
    [*] passwd  
    [*] Check new passwords for weakness  
## update files:
```

In /boards/rtlxxxx/Makefile, add:

```
$(ROMFSINST) /etc/passwd /etc/passwd_orig  
$(ROMFSINST) /etc/group /etc/group_orig  
$(ROMFSINST) -s /var/passwd /etc/passwd  
$(ROMFSINST) -s /var/group /etc/group
```

In /boards/rtlxxxx/etc.default/passwd, add:

```
leo:$1$HG1TCmqu$u8GmNt6f5LUcYM3.HIOkU/:1000:1000:Linux  
User,,,:/home/leo:/bin/sh
```

in /boards/rtlxxxx/etc.default/init.d/rcS, add:

```
cp /etc/passwd_orig /var/passwd  
cp /etc/group_orig /var/group  
telnetd&
```

after the new image upgraded, you can use a LAN PC to telnet 192.168.1.254 with login name "leo" and password "leo".

you can use console command "adduser xxx" and "passwd xxx" to add a user for telnet login account.

4.33.2 login support menuconfig

menuconfig:

(1) Login/Password Management Utilities-->

```
[*] login // selected
```

```

busybox 1.13.4 Configuration
----- Login/Password Management Utilities -----
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted
letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes
features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend:
[*] built-in [ ] excluded <M> module < > module capable

[ ] Support for shadow passwords
[ ] Use internal password and group functions rather than system function
[ ] Use internal DES and MD5 crypt functions
[ ] addgroup
[ ] delgroup
[ ] adduser
[ ] deluser
[ ] getty
[ ] Support utmp file (NEW)
[ ] Support wtmp file (NEW)
[*] login
[ ] Support for PAM (Pluggable Authentication Modules) (NEW)
[ ] Support for login scripts (NEW)
v(+)

<Select> < Exit > < Help >

```

(2) Login/Password Management Utilities--->

[*] passwd // selected

[*]Check new passwords for weakness //selected

```

busybox 1.13.4 Configuration
----- Login/Password Management Utilities -----
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted
letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes
features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend:
[*] built-in [ ] excluded <M> module < > module capable

[ ] Support for shadow passwords
[ ] Use internal password and group functions rather than system function
[ ] Use internal DES and MD5 crypt functions
[ ] addgroup
[ ] delgroup
[ ] adduser
[ ] deluser
[ ] getty
[ ] login
[*] passwd
[*] Check new passwords for weakness (NEW)
[ ] cryptpw
[ ] chpasswd
v(+)

<Select> < Exit > < Help >

```

4.34 Dynamic Frequency Selection (DFS) support

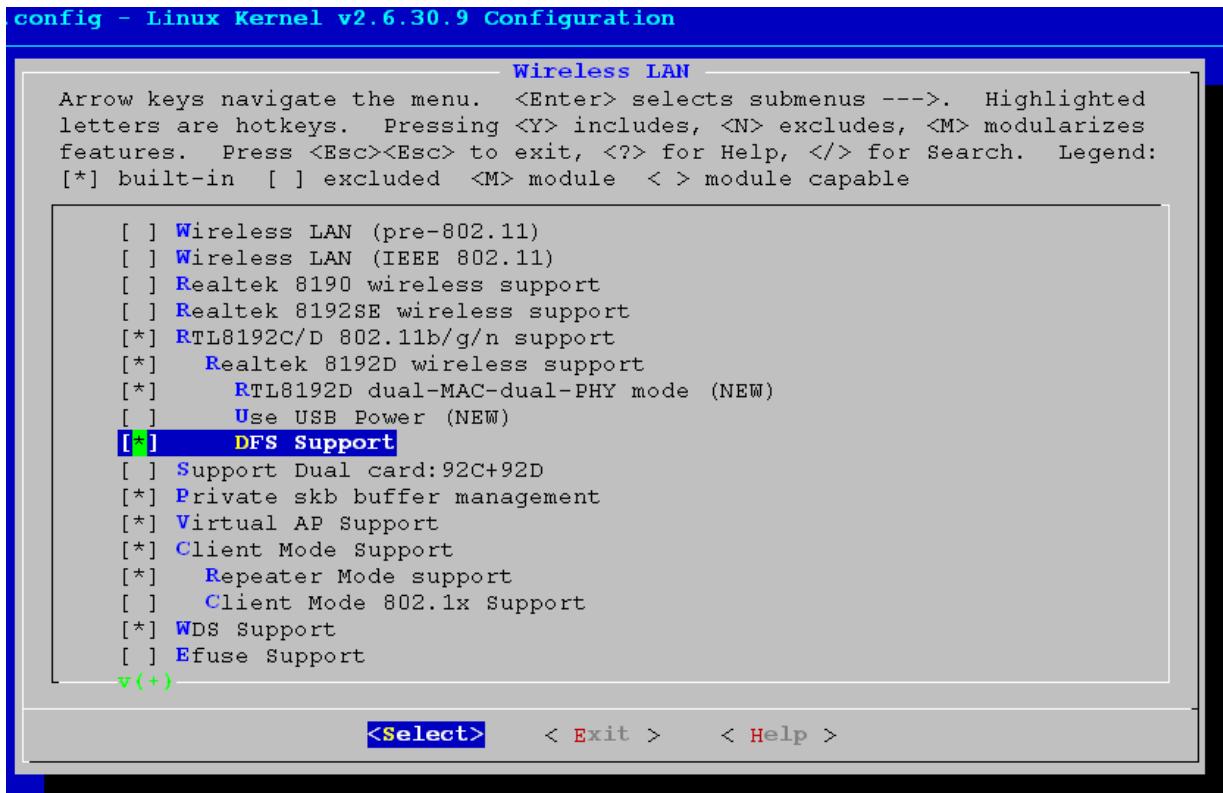
1) kernel configure

make linux_menuconfig //To configure linux kernel

Linux kernel menu as follows:

Device Drivers --->	//enter
[*]Network device support --->	//selected and enter

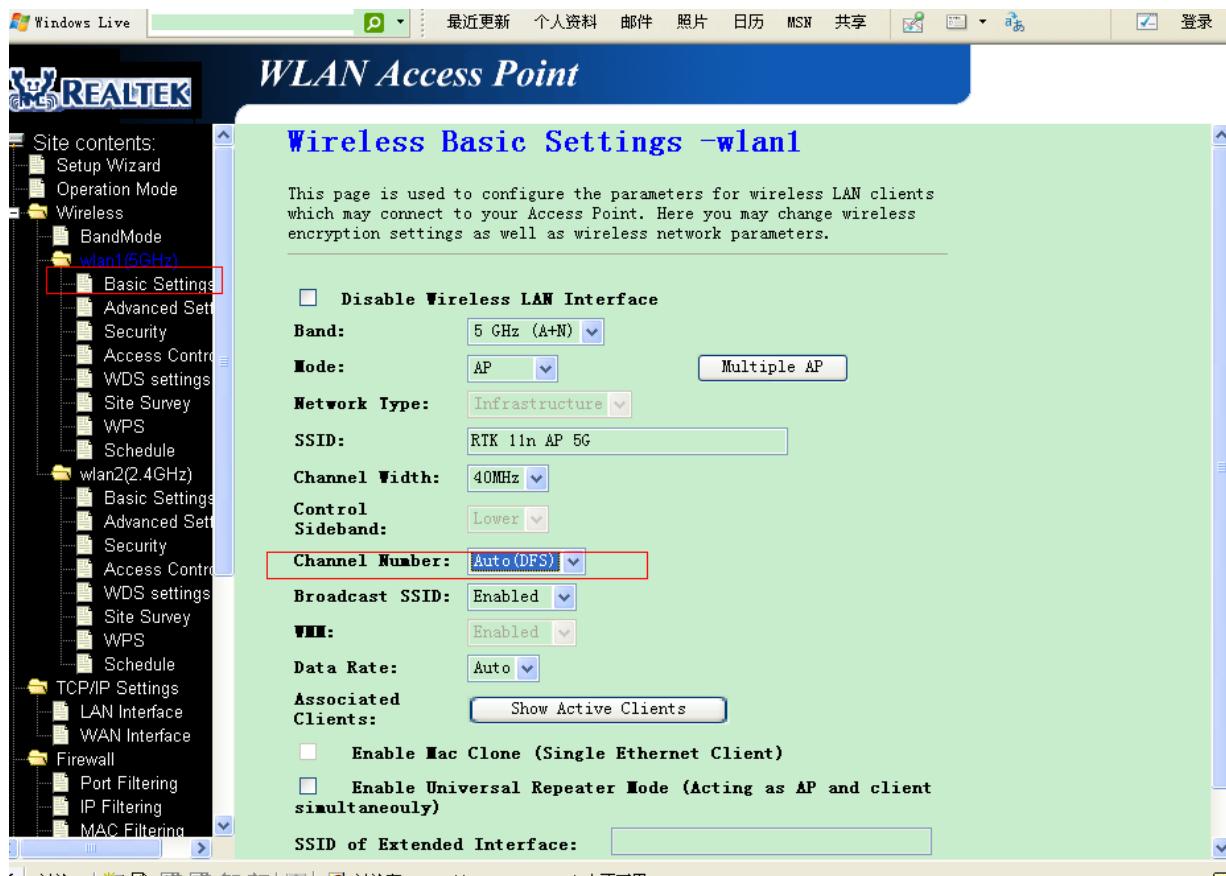
Wireless LAN	--->	//enter
[*]RTL8192C/D 802.11b/g/n support		//selected
[*]Realtek 8192D wireless support		//selected
[*]DFS Support		//selected



- 2) To obey regulation, DFS channels can ONLY be selected by auto-channel selection. The user can see “Auto (DFS)” on the channel column on web UI..

web UI menu as follows:

```
Wireless --->
  Basic Settings --->
    Channel Number --->
      Auto (DFS) //selected
```



- 3) If the user want to force the DUT set in a DFS channel for evaluation purpose, one should set console command with “**flash set WLAN0_CHANNEL <channel #>**”, and then reboot.

Note: Alternatively, the user can use <http://192.168.1.254/syscmd.asp> to input the command.

5G Channel Plan

regulation domain (mib regdomain value)	supported channels – DFS enabled	supported channels – DFS disabled
FCC (1)	36,40,44,48,52,56,60,64,100,104,108, 112,116,136,140,149,153,157,161,165	36,40,44,48,149,153,157,161,165
IC (2)	36,40,44,48,52,56,60,64,149,153,157, 161	36,40,44,48,149,153,157,161
ETSI (3)	36,40,44,48,52,56,60,64,100,104,108, 112,116,120,124,128,132,136,140	36,40,44,48
SPAIN (4)	36,40,44,48,52,56,60,64,100,104,108, 112,116,120,124,128,132,136,140	36,40,44,48
FRANCE (5)	36,40,44,48,52,56,60,64,100,104,108, 112,116,120,124,128,132,136,140	36,40,44,48
MKK (6)	36,40,44,48,52,56,60,64,100,104,108, 112,116,120,124,128,132,136,140	36,40,44,48
ISREAL (7)	36,40,44,48,52,56,60,64,100,104,108,	36,40,44,48

	112,116,120,124,128,132,136,140	
MKK1 (8)	34,38,42,46	34,38,42,46
MKK2 (9)	36,40,44,48	36,40,44,48
MKK3 (10)	36,40,44,48,52,56,60,64	36,40,44,48
NCC (11)	56,60,64,100,104,108,112,116,136,140, 149,153,157,161,165	56,60,64,149,153,157,161,165

4.35 Fastpath Filter

4.35.1 what's Fastpath Filter

'rtk_cmd filter' is a administration tool for IPv4 packet filtering and NAT. Fastpath filter(rtк_cmd filter) is used to set up, maintain, and inspect the packet filter rules before fastpath in Realtek SDK.

4.35.2 How to enable/disable Fastpath Filter feature

Rtk_cmd feature is the basic feature, so rtk_cmd must be enabled if you want enable fastpath filter. Users use rtk_cmd to add fastpath filter rules.

1) Enable rtk_cmd support

make users_menuconfig

[*] rtk_cmd daemon

```

RLX Linux Configuration
the menu. <Enter> selects submenus --->. Highlighted letters are hot
keys, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help,
<cluded <M> module < > module capable

( - )
[ ] pmc utils
[ ] python support
[ ] reload
[ ] routed
[ *] rtk_cmd daemon
[ ] RTK tcp2dev utility
[ ] rtk	voip

```

2) Enable fast filter support

make linux_menuconfig

Device Drivers --->

[*] Network device support --->

[*] Options for Realtek SoC --->

[*] Enable realtek fast filter

```

Options for Realtek SoC
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys. Pressing <Y>
includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend:
[*] built-in [ ] excluded <M> module <> module capable

--- Options for Realtek SoC
      Config MIPS16-Inst Option for Kernel Modules --->
      Config for Layered Driver Features --->
[ ] Enable iptables rule to RTL ACL rule
[*] Enable realtek fast filter
[*] Enable Ethernet Private Skb
[ ] Enable proc filesystem for debug
[ ] Enable rome perf
[*] Support rtk vlan feature
[ ]     rtk vlan for cable modem
[ ] Disable 802.3az EEE feature

```

4.35.3 How to use Fastpath Filter

1) Fastpath Filter Rule Format

Realtek fastpath filter rule format is:

`rtk_cmd filter action filter_rule_option`

- action:
 - add //add rule
 - delete //delete rule
 - flush //flush filter table, clean all rules
 - enableLog // enable log
 - disableLog //disable log
 - show
- filter_rule_option:
 - [--mac-src mac[/mask]]
 - ◆ Source Mac or Mac mask; MAC format:xx:xx:xx:xx:xx:xx or FF/xx:FF/xx:FF/xx:FF/xx:FF/xx, the default mask is FF:FF:FF:FF:FF:FF
 - [--mac-dst mac[/mask]]
 - ◆ destination Mac or Mac mask; MAC format:xx:xx:xx:xx:xx:xx or FF/xx:FF/xx:FF/xx:FF/xx:FF/xx, the default mask is FF:FF:FF:FF:FF:FF

- [--mac-range-src smac/emac]
 - ◆ source mac range, xx:xx:xx:xx:xx:xx/xx:xx:xx:xx:xx:xx
- [--mac-range-dst smac/emac]
 - ◆ destination mac range, xx:xx:xx:xx:xx:xx/xx:xx:xx:xx:xx:xx
- [--ip-src ip[/mask]]
 - ◆ source IP; IP format, 192.168.1.0/255.255.255.0, the default mask is 255.255.255.255
- [--ip-dst ip[/mask]]
 - ◆ Destination IP; IP format, 192.168.1.0/255.255.255.0, the default mask is 255.255.255.255
- [--ip-range-src sip:eip]
 - ◆ Source IP range
- [--ip-range-dst sip:eip]
 - ◆ Destination IP range
- [--port-src port]
 - ◆ Source port
- [--port-dst port]
 - ◆ Destination port
- [--port-range-src sport:eport]
 - ◆ Source port range
- [--port-range-dst sport:eport]
 - ◆ Destination port range
- [--protocol tcp/udp/tcp_udp]
 - ◆ protocol
- [--schedule day_mask:all_hours[:stime:etime]]
 - ◆ day_mask is a char
- [--url-key key_word]
 - ◆ url or key word
- [--phy-port-source port_mask]

- ◆ Physical port
- [--priority priority]
 - ◆ Priority;0~7, 0 is the highest priority, the default priority is 6
- [--mark mark_value]
 - ◆ Mark value
- [--policy drop/fastpath/linux_protocol_stack/mark/omit]
 - ◆ The action if this rule is matched, the default action is drop.

2) Example

- Filter destination mac address “00:e0:4c:11:22:55”

rtk_cmd filter add --mac-dst 00:e0:4c:11:22:55
- Filter source mac address “00:e0:4c:11:22:55”

rtk_cmd filter add --mac-src 00:e0:4c:11:22:55
- Filter destination mac address range, between “00:E0:4C:11:22:34” and “00:E0:4C:11:22:44”

rtk_cmd filter add --mac-range-src 00:E0:4C:11:22:34/00:E0:4C:11:22:44
- Filter key/url “www.biadu.com”

rtk_cmd filter add --url-key www.biadu.com
- Filter source ip “192.168.1.100”

rtk_cmd filter add --ip-src 192.168.1.100
- Filter destination ip “192.168.1.100”

rtk_cmd filter add --ip-dst 192.168.1.100
- Filter source ip range, between “192.168.1.10” and “192.168.1.20”

rtk_cmd filter add --ip-range-src 192.168.1.10:192.168.1.20
- Filter source port “10398”

rtk_cmd filter add --port-src 10398
- Filter destination port “10398”

rtk_cmd filter add --port-dst 10398
- Filter source port range, between “10397” and “10399”

rtk_cmd filter add --port-range-src 10397:10399
- Filter protocol tcp and udp

```
rtk_cmd filter add --protocol tcp_udp
```

- Physical port 5's packet need add mark 12, and this rule priority is 3

```
rtk_cmd filter add --phy-port-source 5 --priority 3 --mark 12 --policy mark
```

4.36 Fastpath SPI (stateful packet inspection) support

4.36.1 What's Fastpath SPI?

SPI (stateful packet inspection): Do packet filter by checking ACK and sequence fields of TCP packets to avoid attacks. Please refer the function `tcp_in_window()` in kernel ipv4 module for detail.

4.36.2 How to enable/disable Fastpath SPI feature

It is base on based on Realtek fastpath and is disabled by default setting.

```
make linux_menuconfig // To configure linux kernel settings
```

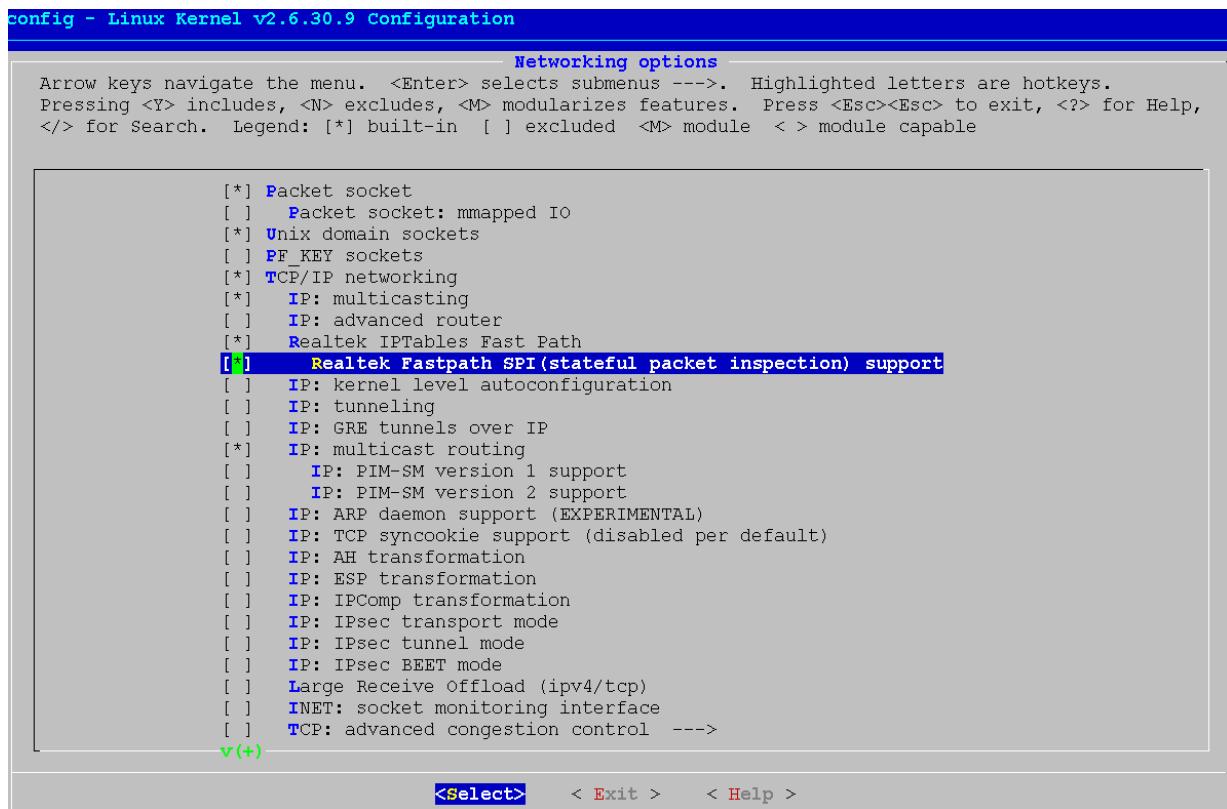
Menuconfig:

Networking support --->

Networking options --->

[*] Realtek IPTables Fast Path

[*] Realtek Fastpath SPI (stateful packet inspection) support



4.37 ALSA support

4.37.1 What's ALSA

The Advanced Linux Sound Architecture (ALSA) provides audio and MIDI functionality to the

Linux operating system.

4.37.2 How to enable/disable ALSA feature

First, Enable USB support (see section 4.2).

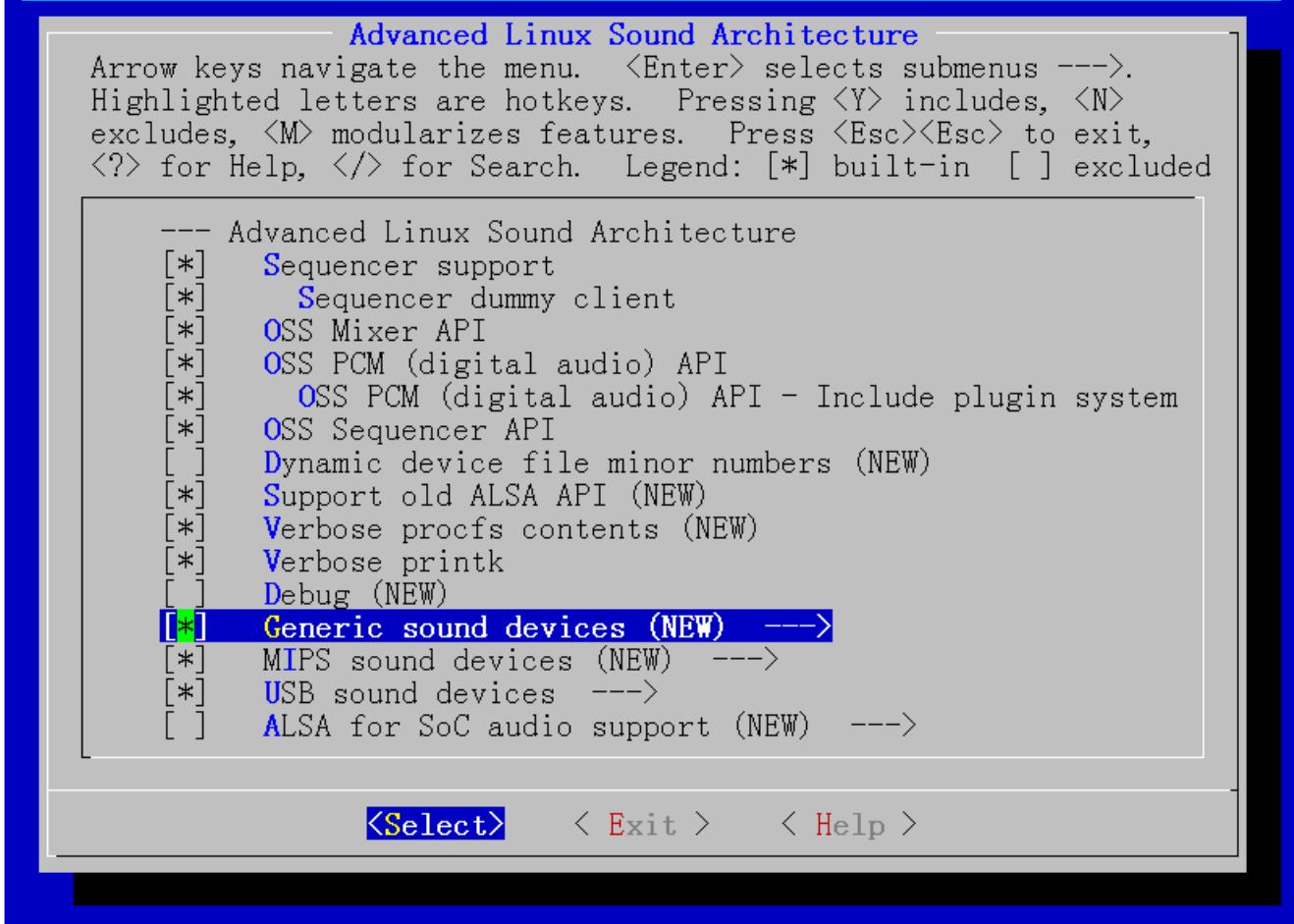
Then, do

```
make linux_menuconfig // To configure linux kernel settings
```

Menuconfig:

Device Drivers --->

```
<*> Sound card support --->
    <*> Advanced Linux Sound Architecture --->
        [*] Sequencer support
            [*] Sequencer dummy client
        [*] OSS Mixer API
        [*] OSS PCM (digital audio) API - Include plugin system
        [*] OSS Sequencer API
        [*] Support old ALSA API
        [*] Verbose procfs contents
        [*] Verbose printk
        [*] Generic sound devices --->
            <*> MOTU MidiTimePiece AV multiport MIDI
            <*> UART16550 serial MIDI driver
            <*> Generic MPU-401 UART driver
        [*] USB sound devices --->
            <*> USB Audio/MIDI driver
            <*> Native Instruments USB audio devices
```



Then, do *cat test.wav > /dev/dsp* on console will hear the voice.

4.38 Romeperf support

4.38.1 What's romeperf

Romeperf function could use to count the number of a function be called (totalNum) , average cycle(Average) and total(accCycle) cycle in statistics.

4.38.2 How to enable/disable romeperf feature

make linux_menuconfig

-> Device Drivers

-> Network device support (NETDEVICES [=y])

-> Options for Realtek SoC (RTL_819X_SWCORE [=y])

-> [*] Enable rome perf

```

options for Realtek SoC
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are
hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to
exit, <?> for Help, </> for Search. Legend: [*] built-in [ ] excluded <M> module < >
module capable

--- Options for Realtek SoC
  Config MIPS16-Inst Option for Kernel Modules --->
  Config for Layered Driver Features --->
  [ ] Enable iptables rule to RTL ACL rule
  [ ] Enable realtek fast filter
  [*] Enable Ethernet Private Skb
  [ ] Support HW Qos
  [ ] Enable proc filesystem for debug
  [!] Enable rome perf
  [*] Support rtk vlan feature
  [ ]     rtk vlan for cable modem

<Select>  < Exit >  < Help >

```

4.38.3 How to use romeperf

- 1) Input the index corresponding to the function for test in rtl8651_romeperfInit (path:drivers/net/rtl819x/romeperf.c). For example:

```
romePerfStat[89].desc = "ip_finish_output3"
```

The function ip_finish_output3 correspond to index 89 in global array romePerfStat.

- 2) Call rtl8651_romeperfEnterPoint (89) in the beginning of function ip_finish_output3 and call rtl8651_romeperfExitPoint (89) in the ending of function ip_finish_output3. Note that the parameter 89 is corresponding to the index value 89 in function rtl8651_romeperfInit
- 3) After the system initiation, input the command in console:

```
echo FLAG1 FLAG2> /proc/rtl865x/perf_dump
```

For example:

```
echo 89 89 > /proc/rtl865x/perf_dump
```

The information of index=89 will be dumped:

<i>index</i>	<i>description</i>	<i>accCycle</i>	<i>totalNum</i>	<i>Average</i>
[89]	ip_finish_output3	2001104	135	14822
		231705	135	1716
		30048	135	222

Note:

FLAG1 in Echo command indicate the start index, and FLAG2 indicate the end index.

If you want to dump the romeperf information of index between 79 and 89, please input:

```
echo 79 89 > /proc/rtl865x/perf_dump.
```

Besides, input the following command:

```
echo flush > /proc/rtl865x/perf_dump
```

And the romeperf will be reset.

4.39 Boa support

This section describes Boa supported features and some useful programming information, which are helpful to develop your own Boa web server.

4.39.1 How to enable Boa web server on the Realtek SDK

1) On the top directory of SDK, execute “make menuconfig” or “make config”.

```
$ make menuconfig
```

2) The configuration menu is shown as follows.

Select "Config kernel" and "Config users", and then Exit the menuconfig.

--- select components

 Selected Target (rtl8196c) --->

 Selected Kernel (Router SDK) --->

 Selected Busybox (busybox-1.13) --->

 Selected toolchain (rsdk-1.3.6-4181-EB-2.6.30-0.9.30) --->

--- rtl8196c

 Selected Target of SDK (11nRouter_GW) --->

 Selected Board Configuration (SPI flash + Squashfs) --->

 IC Test Configuration --->

--- config components

 [*] Config kernel

 [*] Config users

 [] Config busybox

 [] Load default settings

 [] Save default settings

 Load an Alternate Configuration File

 Save an Alternate Configuration File

3) If you turn on APMIB_SHARED feature (APMIB_SHARED = 1, users/boa/Makefile), you have to do the step. Otherwise, skip this step.

In kernel config, enter "General setup" menu to enable "System V IPC" option.

- [*] Prompt for development and/or incomplete code/drivers
- () Local version - append to kernel release
- [] Automatically append version information to the version string
- [] Support for paging of anonymous memory (swap)
- [*] System V IPC
- [] POSIX Message Queues
- [] BSD Process Accounting

4) In users config, enable "boa" application.

--- Applications

- [*] auth
- [*] brctl
- [*] busybox
- [] login on console
- [] Enhanced Ctorrent
- [] dcts
- [] dhcipv6
- [] discover
- [*] dnrd
- [] dnsmasq
- [] dosfsck
- [] gdbserver
- [*] boa
- [] gproxy

4.39.2 Directory Structure

users

```
|-- boa
  |-- apmib           <----- apmib library
```

```

|-- defconfig      <----- default apmib configuration
|-- html          <----- HTML pages and graphics files
|-- src           <----- source code of Boa web server
|-- system        <----- source code for system and network init
|-- tools         <----- tools on x86
`-- utils         <----- flash utility

```

4.39.3 ASP function

When a user requests to get a web page, Boa web server will load the web page file and output it to the browser via HTTP session. If the web page with “*.htm” or “*.asp” extension file name, before outputting it to the browser, Boa web server will parse it to find out all ASP functions. An ASP function is surrounded by tag “<%” and “%>”. And if ASP functions found, it will execute the corresponding callback functions to dynamically generate web pages.

ASP function has a simple syntax: <%function (...); %>. The function argument must be a C string which surrounded by double quotes.

For example,

```

<%getInfo ("countDownTime");%>
var opmode=<% getIndex("wlanMode"); %>;
<% getDHCPModeCombobox (); %>

```

ASP functions are written in the web page file. And its corresponding callback functions are defined on the table “root_asp” as showed below. You can see the source file “asp_page.c” for details.

```

asp_name_t root_asp[] = {
    {"getInfo", getInfo},
    {"getIndex", getIndex},
    {"wirelessClientList", wirelessClientList},
    {"wlSiteSurveyTbl", wlSiteSurveyTbl},
    ...
    ...
    {NULL, NULL}
};

```

Please also see section 7.18 for more information.

4.39.4 Form function

When a user clicks the Apply button on the Web GUI to change some configuration settings, the action represents that the browser would send a POST request message containing a form action string within the request URI to the Boa web server. Once the web server receives the request, it will parse the request URI to get the form action string and then call the corresponding form callback function to do something for users.

The form action string is defined in the html file. It has to be prefixed with “/boafrm/”. For example, “tcpipwan.htm”

```
<body onload="Load_Setting();">  
<blockquote>  
<h2><font color="#0000FF">WAN Interface Setup</font></h2>  
<form action=/boafrm/formWanTcpipSetup method=POST name="tcpip">  
....  
....
```

The corresponding form callback functions are defined on the table “root_form” as showed below. You can see the source file “asp_page.c” for details.

```
form_name_t root_form[] = {  
    {"formWlanSetup", formWlanSetup},  
    {"formWlanRedirect", formWlanRedirect},  
    {"formWep", formWep},  
    ...  
    ...  
    {NULL, NULL}  
};
```

Please also see section 7.18 for more information.

4.39.5 No Ejscript Support

Ejscript is a server-side script language. Boa web server does not support this feature. If your web pages have this kind of script, you should re-write them. In general, you can use JavaScript and asp function to implement the same functionality provided with Ejscript. There are many examples in our

Boa web server. Please see source codes for details.

For example, in “users/goahead-2.1.1/web-gw/tcpipwan.asp”, the following Ejscript codes in blue are not supported on Boa.

```
<td width="30%" height=40><font size=2><b>WAN Access Type:</b></td>

<td width="70%"><font size=2><select size="1" name="wanType" onChange="wanTypeSelection(this)">

<option <% if (getIndex("wanDhcp")==0) write("selected"); %> value="fixedIp">Static IP</option>

<option <% if (getIndex("wanDhcp")==1) write("selected"); %> value="autoIp">DHCP Client</option>

<option <% if (getIndex("wanDhcp")==3) write("selected"); %> value="ppp">PPPoE</option>

<option <% if (getIndex("wanDhcp")==4) write("selected"); %> value="pptp">PPTP</option>

....<

<td width="70%"><font size=2><select size="1" name="pppConnectType" onChange="pppTypeSelection(0)">

<% var type = getIndex ("pppConnecType");

    if (type == 0 ) {

        write ("<option selected value=\"0\">Continuous</option>");

        write ("<option value=\"1\">Connect on Demand</option>");

        write ("<option value=\"2\">Manual</option>");

    }

    if (type == 1) {

        write ("<option value=\"0\">Continuous</option>");

        write ("<option selected value=\"1\">Connect on Demand</option>");

        write ("<option value=\"2\">Manual</option>");

    }

    if (type == 2) {

        write ("<option value=\"0\">Continuous</option>");

        write ("<option value=\"1\">Connect on Demand</option>");

        write ("<option selected value=\"2\">Manual</option>");

    }  %>

</select>&nbsp;&nbsp;

<input type="submit" value="Connect" name="pppConnect" onClick="return pppConnectClick(0)">&nbsp;&nbsp;
```

```

<input type="submit" value="Disconnect" name="pppDisconnect" onClick="return pppConnectClick(1)">
<% if ( getIndex("pppConnectStatus") ) write("\n<script> setPPPConnected(); </script>\n"); %>
</td>

```

In “users/goahead-2.1.1/web-gw/ wlbasic.asp”,

```

var countrystr= "<% getInfo ("country_str"); %>";
var wlan_idx= <% write (getIndex("wlan_idx")); %> ;
var opmode=<% write (getIndex("opMode")); %> ;
var WiFiTest=<% write (getIndex("WiFiTest")); %> ;
var ssid_2g=<% write (getIndex("2G_ssid")); %>;
var ssid_5g=<% write (getIndex("5G_ssid")); %>;
var info_country=<% getInfo ("info_country");%>;
.....
<SCRIPT>
<%   val = getIndex("RFType");
      write ("RFType[wlan_idx]="+ val + ";\n");
      val = getIndex ("wlanMode");
      write ("\tAPMode[wlan_idx]="+ val + ";\n");
      val = getIndex ("band");
      if (val > 0) val = val-1;
      write ("\tbandIdx[wlan_idx]="+ val + ";\n");
%>
.....

```

4.39.6 Boa configuration file

Boa configuration file “boa.conf” is put in the directory “users/boa/src” and it should be installed into the directory “/etc/boa/” of the root file system. Please check the target “romfs:” in the “users/boa/Makefile” for details. Since the file is self-documented, it’s easy to understand the meaning of each setting. So we just list some useful settings here for reference.

Port 80	# The port Boa listens on.
---------	----------------------------

```

PidFile /var/run/webs.pid          # where to put the pid of Boa process
DocumentRoot /web                  # The root directory of the HTML documents
MimeType /etc/boa/mime.types       #This is the file that is used to generate mime type pairs
                                   # and Content-Type fields for boa.
DefaultType text/html             #MIME type used if the file extension is unknown, or there
                                   # is no file extension.

```

4.39.7 no-cache

To prevent the browser from caching the web page, you can add the following HTML Meta Tags to your page directly.

```

<meta http-equiv="Pragma" content="no-cache">
<meta HTTP-equiv="Cache-Control" content="no-cache">
<meta HTTP-EQUIV="Expires" CONTENT="Mon, 01 Jan 1990 00:00:01 GMT">

```

Additionally, you can add a asp function “<% getIndex("no-cache"); %>” instead.

For example, in “users/boa/html/tcpipwan.htm”

```

<html>
<! Copyright (c) Realtek Semiconductor Corp., 2003. All Rights Reserved. ->
<head>
<meta http-equiv="Content-Type" content="text/html">
<% getIndex("no-cache"); %>
<title>WAN Interface Setup </title>
<style>
.....

```

4.39.8 Migration from old web server to Boa web server

To port your code of old web server to Boa web server, the following action items are needed.

1) asp and form function codes should be porting to “users/boa/src/”. The definition of asp and form functions should be added to the table “root_asp” and “root_form” in “asp_page.c” respectively.

- 2) web pages should be porting to “users/boa/html/”.
 - 3) form action name should be renamed to be prefixed with “/boafrm/”. So web pages should be modified.
 - 4) review your web page to find out all Ejscript code and rewrite them.
 - 5) The header file “users/boa/src/port.h” defines some useful macros. Include the header file “port.h” to your source code to get more compatibility.
 - 6) The prototype of asp callback function on the old web server is different from Boa. You should modify them to be the same as Boa.
 - 7) Some header files included in your source code are not existed in Boa now. You should remove or comment out those statements. Additionally, you need to include "boa.h" and "asp_page.h" in your source as below.
- ```
#include "boa.h"
#include "asp_page.h"
```
- 8) There are a lot of porting examples in the directory “users/boa/src” and “users/boa/html”. If you have a problem, we recommended that you should compare old source code and Boa source code first and then you can quickly find out the answer.

## **4.40 uWiFi support**

### **4.40.1 uWiFi Introduction**

uWiFi is used to negotiate with uWiFi manager that is installed in Windows PC. Users can use uWiFi to change the share method of USB storage device. Either share by HTTP or by uWiFi method. If the method is http, then users can Brower/upload/Delete files on USB storage device which connected to Realtek RTL819x Wireless Home Gateway via http protocol. If the method is uWiFi, then users can use uWiFi manager to connect the storage device.

### **4.40.2 Configuration for uWiFi support**

#### 1) Kernel configure for uWiFi

All of the settings of HTTP File server should be included, the additional configuration is:

Device Drivers --->

[\*] Staging drivers ---> //Select

```

DEVICE DRIVERS
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are
hotkeys. Pressing <> includes, <N> excludes, <M> modularizes features. Press
<Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [] excluded
<M> module < > module capable

[] Character devices --->
[] I2C support --->
[] SPI support --->
[] Dallas's 1-wire support --->
[] Power supply class support --->
[] Hardware Monitoring support --->
[] Generic Thermal sysfs driver --->
[] Watchdog Timer Support --->
 Sonics Silicon Backplane --->
 Multifunction device drivers --->
 Voltage and Current Regulator Support --->
 Multimedia devices --->
 Graphics support --->
[] Sound card support --->
[*] USB support --->
[] MMC/SD/SDIO card support --->
[] Sony MemoryStick card support (EXPERIMENTAL) --->
[] LED Support --->
[] Accessibility support --->
[] Real Time Clock --->
[] DMA Engine support --->
[] Auxiliary Display support --->
[] Userspace I/O drivers --->
[*] Staging drivers --->

```

[\*] RTK USB uWiFi host driver

//Select

```

STAGING DRIVERS
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are
hotkeys. Pressing <> includes, <N> excludes, <M> modularizes features. Press
<Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [] excluded
<M> module < > module capable

[*] Staging drivers
[] Exclude Staging drivers from being built
[*] RTK USB uWiFi host driver
[] USB IP support (EXPERIMENTAL)
[] Line Echo Canceller support
[] Asus OLED driver
[] Frontier Transport and Alphatrack support
 Android --->
 Distributed storage
[] POHMELES filesystem support
[] Brontes 3d Frame Grabber
[] Plan 9 style capability device implementation
[] Intel Management Engine Interface (MEI) Support

```

## 2) Application configure for uWiFi:

All of the settings of HTTP File server should be included, the additional configuration is:

[\*] uWiFi // selected  
[\*] nmbs // selected

```

----- RLX Linux Configuration -----
the menu. <Enter> selects submenus --->. Highlighted letters are hot
keys, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help,
<excluded <M> module < > module capable

(-)
[] nfbi host
[*] nmbs
[] ntpclient
[*] openssl
 openssl (openssl-0.9.8b) --->
[] wapi_utils (NEW)
[] oray ddns
[*] pathsel
[] pmc utils
[] python support
[] reload
[] routed
[*] rtk_cmd daemon
[] RTK tcp2dev utility
[] rtk	voip
[] RTL8686 IPC User Utility
[*] samba support --->
[] scripts
[] simple config support
[*] snmp
[] squashfs 4.0
[] squashfs 4.2
[] tr069
[] udhcp
[] ulinker autorun image tool
[] updateddd DDNS
[] upImage
[] usbmount
[*] uWiFi
[] vsftpd
[] v

```

#### 4.40.3 WEB GUI Manual

Home page:

##### Realtek uWiFi



##### Shared Partitions:

No shared partition available.

##### Storage Device Management:

No USB storage device available.

[Apply Changes](#) | [Reset](#)

The Hyperlink “Settings” is for Router configuration, user can set share method for storage device that list in Storage Device Management table. In default, the method is in “uWiFi” mode.



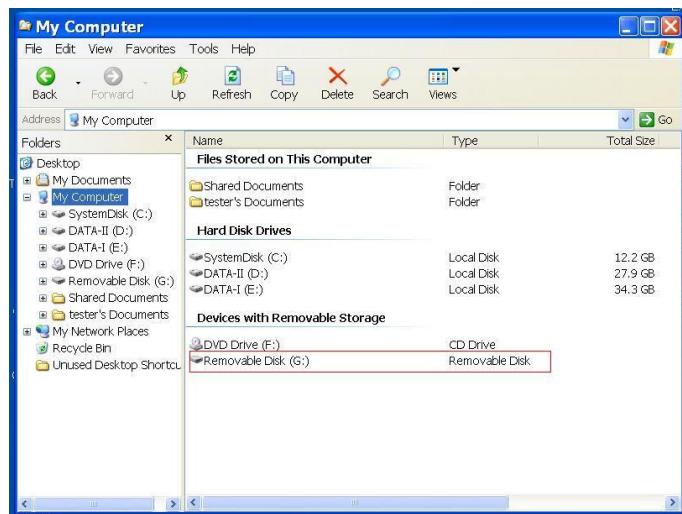
If the user selects share method “uWiFi”, then uWiFi Manager can see storage device icon. uWiFi Manager can be used to “Connect” if the current state is Disconnect.



When user selects “connect”, uWiFi manager will connect to storage device.



User can see the new drive is attached to PC



uWiFi attachment information will be showed in web home page if storage device is attached by uWiFi mode.

Realtek uWiFi

[Settings](#)

---

Shared Partitions:  
No shared partition available.

---

Storage Device Management:

| USB Device                   | Share Method                                                                                   |
|------------------------------|------------------------------------------------------------------------------------------------|
| JetFlash Mass Storage Device | <input type="radio"/> Http <input checked="" type="radio"/> uWiFi (In used by zzaadd-6toqmew9) |

[Apply Changes](#) [Reset](#)

uWiFi Manager can be used to “Disconnect” if the current state is Connect.



When user selects “Disconnect”, uWiFi manager will disconnect storage device.



If the user selects share method “Http”, the partition name will be showed in “Shared Partitions” when it is available.

A screenshot of the Realtek uWiFi configuration interface. The top bar has a "Settings" link. Below it, under "Shared Partitions", there is a link to "sda1". Under "Storage Device Management", there is a table with one row: "USB Device: JetFlash Mass Storage Device" and "Share Method: Http". At the bottom are "Apply Changes" and "Reset" buttons.

User can enter the hyperlink in shared partition for file access.

User can click the hyperlink “Name”, “Last modified”, and “Size” for sort.

## Index of /sda1

| Name                             | Last modified        | Size   |                        |
|----------------------------------|----------------------|--------|------------------------|
| <a href="#">Parent Directory</a> |                      |        |                        |
| <a href="#">123/</a>             | 28-Oct-2010 16:07:00 | -      | <a href="#">Remove</a> |
| <a href="#">abc/</a>             | 28-Oct-2010 18:18:14 | -      | <a href="#">Remove</a> |
| <a href="#">1.txt</a>            | 20-Oct-2010 15:05:14 | 25.68K | <a href="#">Remove</a> |

Select File:

### 4.40.4 Lan NetBIOS Name query service support:

In Lan settings, user can set LAN NetBIOS name for Realtek RTL819x Wireless Home Gateway. The default value is “uWiFi”.

## LAN Interface Setup

This page is used to configure the parameters for local area network which connects to the LAN port of your Access Point. Here you may change the setting for IP address, subnet mask, DHCP, etc.

|                       |                                 |
|-----------------------|---------------------------------|
| IP Address:           | 192.168.1.254                   |
| Subnet Mask:          | 255.255.255.0                   |
| Default Gateway:      | 0.0.0.0                         |
| DHCP:                 | Auto                            |
| DHCP Client Range:    | 192.168.1.100 - 192.168.1.200   |
| DHCP Lease Time:      | 480 (1 ~ 10080 minutes)         |
| Static DHCP:          | <a href="#">Set Static DHCP</a> |
| Domain Name:          | Realtek                         |
| LAN BIOS NAME:        | uWiFi                           |
| 802.1d Spanning Tree: | Disabled                        |
| Clone MAC Address:    | 000000000000                    |

[Apply Changes](#) [Reset](#)

Users can use LAN BIOS Name to connect Realtek RTL819x Wireless Home Gateway.

The screenshot shows the 'Realtek uWiFi' configuration page in Internet Explorer. The 'LAN BIOS NAME' field is highlighted with a red border. The page includes sections for Shared Partitions (empty), Storage Device Management (listing a JetFlash Mass Storage Device shared via uWiFi), and buttons for Apply Changes and Reset.

## 4.41 ALL NAT

### 4.41.1 what's Multiple wan (All Nat)

There are 2 network interfaces with different ip address. Different service packet will be routed/napted to different network interface, all connection should be forwarded by hardware napt.

And only rtl8198t & rtl8196ct support this feature!

### 4.41.2 How to enable/disable Multiple wan

1. menuconfig:

Device Drivers --->

[\*] Network device support --->

[\*] Options for Realtek SoC --->

## Config for Layered Driver Features --->

[\*] Enable multiple wan in hw nat mode

The screenshot shows a terminal window titled '.config - Linux Kernel v2.6.30.9 Configuration'. The menu is titled 'Config for Layered Driver Features'. It includes instructions for navigating with arrow keys, pressing Y for yes, N for no, M for modular, Esc for exit, ? for help, and / for search. A legend indicates that [\*] means built-in, [ ] means excluded, <M> means modularized, and <> means module capable.

Options listed under 'Hardware Features Selection (Enable RTL Hardware NAPT) --->' include:

- [\*] enable linux ps vs rtl819x driver netif mapping
- [\*] Enable multiple wan in hw nat mode (highlighted with a red box)
- [ ] Rt1865x lan port restriction
- [ ] HW Qos support vlan priority
- [\*] Support multi-vlan in bridge/wisp mode
- [ ] Enable RTL local public

At the bottom of the terminal window, there are navigation keys: <Select>, < Exit >, and < Help >.

## 2. Network interface

ifconfig -a, interface ethX is created for multiple wan feature.

ex. eth1:192.168.2.11/24, eth6:192.168.3.11/24

## 3. ip configuration

configure ip address for this interface. please use advanced route if the two wan interface in same subnet.

ifconfig eth1 192.168.2.11 netmask 255.255.255.0

ifconfig eth6 192.168.3.11 netmask 255.255.255.0

## 4. Iptables rule configuration

configure iptables rule:

iptables -F

iptables -P INPUT ACCEPT

iptables -P FORWARD ACCEPT

iptables -t nat -F

```
iptables -t nat -A POSTROUTING -s 192.168.1.0/25 -o eth1 -j MASQUERADE
iptables -t nat -A POSTROUTING -s 192.168.1.128/25 -o eth6 -j MASQUERADE
```

## 5. adv route configuration

add/del rtl adv route to proc/adv\_rt, the format is:

"add/del srcip\_start srcip\_end dstip\_start dstip\_end extIp nexthop outinterface" which means that packet whose source ip address in srcip range and destip address in dst range will be forwarded to nexthop, and the napt ip address is extIp.

Ex.

```
echo "add 192.168.1.128 192.168.1.255 192.168.3.0 192.168.3.255 192.168.3.11 192.168.3.253
eth6">/proc/adv_rt
```

Note: When test with eth6, you should set the wan PC ip as nexthop(192.168.3.253).

## 4.42 Proc filesystem for debug

There are several proc filesystem and command provided for debug.

### 4.42.1 How to enable proc filesystem for debug

make linux\_menuconfig

Device Drivers --->

[\*] Network device support --->

[\*] Options for Realtek SoC --->

[\*] Enable proc filesystem for debug

```

----- options for Realtek SoC -----
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are
hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press
<Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [] excluded
<M> module < > module capable

--- Options for Realtek SoC
 CConfig MIPS16-Inst Option for Kernel Modules --->
 CConfig for Layered Driver Features --->
 [] EEnable iptables rule to RTL ACL rule
 [] EEnable realtek fast filter
 [*] EEnable Ethernet Private Skb
 [I] EEnable proc filesystem for debug
 [] EEnable log print system for debug
 [] EEnable rome perf
 [*] SSupport rtk vlan feature
 [] rtk vlan for cable modem
 [] add rtk vlan bridge feature
 [] DDisable 802.3az EEE feature

<Select> < Exit > < Help >

```

#### 4.42.2. Proc illustration:

CASE 1:

When Enable proc filesystem for debug is selected, Marco CONFIG\_RTL\_PROC\_DEBUG=y, all the proc entry defined in rtl865x\_proc\_debug.c will be established in /proc/rtl865x/..

CASE 2:

When Enable proc filesystem for debug isn't selected, Marco CONFIG\_RTL\_PROC\_DEBUG=n, only the following proc entry will be established: (Marco: CONFIG\_RTL\_DEBUG\_TOOL, default y)

|             |              |        |             |
|-------------|--------------|--------|-------------|
| asicCounter | fc_threshold | memory | phyReg      |
| diagnostic  | mac          | mmd    | port_status |

#### 4.42.3. Command illustration:

The following command is enabled when

CONFIG\_RTL\_DEBUG\_TOOL||CONFIG\_RTL\_PROC\_DEBUG

1). dw

Description:

command to read value from memory

Input Format:

dw <Address> <Len>

Output Format :

Address : data

2). ew

Description:

command to write value to memory

Input Format:

ew Address value

Output Format :

Address : value

3). phyr

Description:

read value from phy register

Input Format:

phyr phyId pageId regId

Output Format :

extRead phyId, pageId, regId, regData

4). phyw

Description:

write value to phy register

Input Format:

phyw <phyId> <pagId> <regId> <regData>

Output Format :

extWrite phyId, pageId, regId, regData

5) .mmd\_cmdr

Description:

read EEE information by mmd proc entry

Input Format:

mmd\_cmdr phyID deviceID addressID

Output Format :

read phyId, devId, regId, regData

6) .mmd\_cmdw

Description:

write value to phy register by mmd proc entry

Input Format:

mmd\_cmdw <phyId> <devId> <regId> <regData>

Output Format :

extWrite phyId, devId, regId, regData

## **4.43 TR069 Support**

### **4.43.1 Specification**

Comply with:

1. TR-069 CPE WAN Management Protocol v1.1

Version: Issue 1 Amendment 2

Version Date: December 2007

2. TR-098 Internet Gateway Device Data Model for TR-069

Issue: 1 Amendment 2

Issue Date: September 2008

3. TR-181 Device Data Model for TR-069

Issue: 2 Amendment 2

Issue Date: February 2011

4. TR-104 VoiceService:1.0 Service Object definition for TR-069

Issue Date: September 2009

### **4.43.2. User program**

- I. /users/cwmp-tr069/cwmpClient: parameter data model
- II. /users/cwmp-tr069/gsoaplib: gsoap library
- III. /users/flatfsd: flatfs utility

### **4.43.3. RPC Method**

GetRPCMethods  
SetParameterValues  
GetParameterValues  
GetParameterNames  
SetParameterAttributes  
GetParameterAttributes  
AddObject  
DeleteObject  
Reboot  
Download

Upload  
FactoryReset  
ScheduleInform  
Inform

#### **4.43.4. Parameter Data Model**

- TR-098 Amendment 2

Internet Gateway Device Data Model v1.1:

- Baseline:1 Profile
- EthernetWAN:1 Profile
- EthernetLAN:1 Profile
- WiFiLAN:1 Profile
- Time:1 Profile
- IPPing:1 Profile

- TR-181 Issue 2 Amendment 2

Device Data Model v2.2:

- Baseline:1 Profile
- DeviceAssociation:1 Profile
- GatewayInfo:1 Profile
- EthernetInterface:1 Profile
- EthernetLink:1 Profile
- WiFiRadio:1 Profile
- WiFiSSID:1 Profile
- WiFiAccessPoint:1 Profile
- Bridge:1 Profile
- IPInterface:1 Profile
- IPInterface:2 Profile
- IPv6Interface:1 Profile
- DHCPv4Client:1 Profile
- DHCPv6Client:1 Profile

- VendorLogFiles: 1 Profile
  - IPPing:1 Profile
  - UDPConnReq:1 Profile
- TR-104
- Please see Realtek\_TR069\_TR098\_TR181\_SupportList.pdf

#### 4.43.5. ACS Server Tested

- Comtrend
- OpenACS
- CDRouter

#### 4.43.6. Compile user program

- Kernel menuconfig

Set flatfs is supported in default, in “System Configuration”

Flatfs image offset is located in the latest 128K of flash memory

```
----- System Configuration -----
Arrow keys navigate the menu. <Enter> selects submenus -->. Highlighted letters are hotkeys. Pressing
<Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for
Search. Legend: [*] built-in [] excluded <M> module capable < > module capable

^(-)
*** Flash size 2M or 4M, default 2M ***
(0x400000) Size of Flash
 *** Hardware setting offset,should be 4K alignment ***
(0x6000) Hardware setting offset in flash.
 *** Default setting offset,should be 4K alignment. ***
 *** size of default and current setting should be same. ***
(0x8000) Default setting offset in flash.
 *** Current setting offset,should be 4K alignment. ***
(0xC000) Current setting offset in flash.
 *** Webpage image offset,should be 4K alignment. ***
 *** size of web page is normally about 100K. ***
(0x10000) webpages image offset in flash.
 *** Linux image offset,should be 4K alignment. ***
 *** this offset MUST between 0x10000~0x40000. ***
(0x30000) linux image offset in flash.
 *** Root image offset,should be 64K alignment. ***
(0x130000) root image offset in flash.
 *** Flatfs image offset,should be 64K alignment. ***
(0x3E0000) FLATFS image offset in flash.
(3) Kelen Stack Size Order Configuration
v(+)

<Select> <Exit > < Help >
```

Device Drivers --->

Memory Technology Device (MTD) support --->

```
-- Memory Technology Device (MTD) support
[] Debugging
[] MTD concatenating support
[*] MTD partitioning support
[] RedBoot partition table parsing
[] Command line partition table parsing
[] TI AR7 partitioning support
*** User Modules And Translation Layers ***
[*] Direct char device access to MTD devices
```

- Users menuconfig

```
----- RLX Linux Configuration -----
the menu. <Enter> selects submenus --->. Highlighted letters are hot
keys, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help,
<cluded <M> module < > module capable

--- (-)
[] bird-1.4.0 (NEW)
[*] rtk_cmd daemon
[] RTK tcp2dev utility
[] rtk_voip
[] RTL8686 IPC User Utility
[*] samba support --->
[] scripts
[] simple config support
[*] snmp
[] squashfs 4.0
[] squashfs 4.2
[*] tr069
-- Flatfs support for TR069
 Subarchitecture Type (no support any SSL) --->
[] support TR143 in TR069 (NEW)
[*] support TR181 in TR069
[] support TR069 Annex G (NEW)
[!] support TR181 V6 part in TR069
[] udhcp
[] ulinker autorun image tool
```

In default, flatfs is enabled. TR143 is supported but it is not verified in this stage. Hence TR143 default is disabled. If only tr069 option is selected, the TR-098 data model is supported. If tr069 and support TR181 in TR069 options are selected, the TR-181 data model is supported. If tr069, support TR181 in TR069 and support TR181 V6 part in TR069 options are selected, the TR181 data model is supported and IPv6 related part is included. If support TR181 V6 part in TR069 option is selected, please enable IPv6 support according to section 4.15.

```

RLX Linux Configuration
the menu. <Enter> selects submenus --->. Highlighted letters are hot
ties, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help,
<cluded <M> module <> module capable

-- (-)
[] bird-1.4.0 (NEW)
[*] rtk_cmd daemon
[] RTK tcp2dev utility
[*] rtk_voip
[] E8C (NEW)
[] RTL8686 IPC User Utility
[*] samba support --->
[] scripts
[] simple config support
[*] snmp
[] squashfs 4.0
[] squashfs 4.2
[*] tr069
--*- Flatfs support for TR069
Subarchitecture Type (no support any SSL) --->
[] support TR104 in TR069 (NEW)
[] support TR143 in TR069 (NEW)
[*] support TR181 in TR069
[] support TR069 Annex G (NEW)
[*] support TR181 v6 part in TR069
[] unknown

```

If rtk\_voip is enabled, the support TR104 in TR069 option will appear. The supported ICs are only RTL8954C and RTL8972D.

#### 4.43.7. Enable/Disable TR069

Enable/Disable Tr-069 from WEB GUI:



## TR-069 Configuration

This page is used to configure the TR-069 CPE. Here you may change the setting for the ACS's parameters.

|                           |                                                                         |
|---------------------------|-------------------------------------------------------------------------|
| TR069:                    | <input type="radio"/> Disabled <input checked="" type="radio"/> Enabled |
| ACS:                      |                                                                         |
| URL:                      | http://172.21.146.44/cpe/?pd128                                         |
| User Name:                | sd9_e8                                                                  |
| Password:                 | 1234                                                                    |
| Periodic Inform Enable:   | <input type="radio"/> Disabled <input checked="" type="radio"/> Enabled |
| Periodic Inform Interval: | 0                                                                       |
| Connection Request:       |                                                                         |
| User Name:                | 123456                                                                  |
| Password:                 | 123456                                                                  |
| Path:                     | /t069                                                                   |
| Port:                     | 4567                                                                    |

Certificat Management:

CA Certificat:

Tr069 daemon is started when system booting (in /etc/init.d/rcS script). If you modify ACS url or account, Tr069 daemon will re-start after you apply changes.

You also have to open the port number to accept ACS connection request in firewall setting.

### 4.43.8. Enable/Disable TR069 Debug Message Output

In console, use “flash” utility to enable/disable debug message

Enable Debug Message output:

```
flash set CWMP_FLAG 97
flash get CWMP_FLAG
CWMP_FLAG=97

```

Disable Debug Message output:

```
flash set CWMP_FLAG 96
flash get CWMP_FLAG
CWMP_FLAG=96

```

CWMP\_FLAG bit 0:

1: enable debug message output

0: disable debug message output

When you modify “CWMP\_FLAG” by “flash” utility, you have to restart “cwmpClient” daemon, or reboot system to take effect.

#### 4.43.9. Usage of Flatfs

TR-069 daemon (cwmpClient) use flatfs to save notify list to flash memory.

The path is “/var/cwmp\_config”. The file name “CwmpNotify.txt”. Below is the format:

ParameterName Notification AccessList

Example:

```
cat ./CWMPNotify.txt
InternetGatewayDevice.DeviceSummary 5 1
InternetGatewayDevice.DeviceInfo.SpecVersion 5 1
InternetGatewayDevice.DeviceInfo.HardwareVersion 5 1
InternetGatewayDevice.DeviceInfo.SoftwareVersion 6 1
InternetGatewayDevice.DeviceInfo.ProvisioningCode 6 1
InternetGatewayDevice.ManagementServer.ConnectionRequestURL 6 1
InternetGatewayDevice.ManagementServer.ParameterKey 5 1
InternetGatewayDevice.WANDevice.1.WANConnectionDevice.1.WANIPConnection.1.ExternalIPAddress 6 1
InternetGatewayDevice.WANDevice.1.WANConnectionDevice.2.WANIPConnection.1.ExternalIPAddress 6 1
InternetGatewayDevice.WANDevice.1.WANConnectionDevice.3.WANIPConnection.1.ExternalIPAddress 6 1
InternetGatewayDevice.WANDevice.1.WANConnectionDevice.4.WANIPConnection.1.ExternalIPAddress 6 1
InternetGatewayDevice.LANDevice.1.WLANConfiguration.1.SSID 2 1
[]
```

Notification:

0 = Notification off.

1 = Passive notification.

2 = Active notification.

4 = Forced notification

AccessList

0 = off

1 = Subscriber

Configuration of Flatfs:

In Kernel Configuration should set the offset for flatfs.

In default the offset is the latest 128K of flash memory. And device node is necessary for flatfs in /dev directory.

In default /dev/mtd1, /dev/mtd2, /dev/mtd3, /dev/mtd4 are generated when the rootfs is created during compile firmware. /dev/mtd4 is used by “flatfsd” to create flatfs and sync content to flash

memory.

“flatfsd -r”: read, or create flatfs from flash memory to main memory

“flatfsd -s”: sync the data in main memory to flash memory, such the data will not disappear after system reboot

#### 4.43.10. Certificate

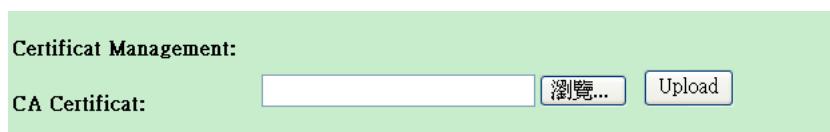
The default certificates are placed on the below location:

\users\cwmp-tr069\_v2\cwmpClient\cacert.pem

\users\cwmp-tr069\_v2\cwmpClient\client.pem

They can be changed for user’s default certificate.

Certificate can also be upload from TR069 web page.



#### 4.43.11. TR-069 Annex G – Connection Request via NAT Gateway

```
RLX Linux Configuration
the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, <cluded <M> module <> module capable

--- (-)
[] bird-1.4.0 (NEW)
[*] rtk_cmd daemon
[] RTK tcp2dev utility
[*] rtk_voip
[] E8C (NEW)
[] RTL8686 IPC User Utility
[*] samba support --->
[] scripts
[] simple config support
[*] snmp
[] squashfs 4.0
[] squashfs 4.2
[*] tr069
-- Flatfs support for TR069
 Subarchitecture Type (no support any SSL) --->
[] support TR104 in TR069 (NEW)
[] support TR143 in TR069 (NEW)
[*] support TR181 in TR069
[!] support TR069 Annex G
[*] support TR181 V6 part in TR069
```

Connection Request via NAT Gateway feature can be enabled by selecting “support TR069 Annex G” in users\_menuconfig. The related Data Model is UDPConnReq:1 Profile. The message flows can reference the Annex G of the TR-069\_Amendment-2.pdf. This feature can be used if and only if the

ACS server also supports the TR-069 Annex G.

#### **4.43.12. Reference documents**

Realtek\_TR-069\_How\_to\_Add\_the(TR-69)\_Parameters.pdf

Realtek\_TR069\_TR098\_TR181\_SupportList.pdf

### **4.44 ALG Support**

#### **4.44.1 DESCRIPTION**

SDK support the following alg:

1. SIP;
2. FTP;
3. TFTP;
4. H323;
5. PPTP;
6. L2TP;
7. RTSP;
8. IPSEC;

#### **4.44.2 How to enable ALG**

*make linux\_menuconfig*

Take enable IPSEC ALG as an example:

```

Core Netfilter Configuration
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys. Pressing <Y>
includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend:
[*] built-in [] excluded <M> module <> module capable

` (-)
[] DCCP protocol connection tracking support (EXPERIMENTAL)
[] SCTP protocol connection tracking support (EXPERIMENTAL)
[] UDP-Lite protocol connection tracking support
[] Amanda backup protocol support
[*] FTP protocol support
[] H.323 protocol support
[] IRC protocol support
[] NetBIOS name service protocol support
[!] IPSEC protocol support
[*] RTSP protocol support
[*] L2TP protocol support
[*] PPTP protocol support
[] SANE protocol support (EXPERIMENTAL)
[*] SIP protocol support
[] TFTP protocol support
[] Connection tracking netlink interface
[] Transparent proxying support (EXPERIMENTAL)
-- Netfilter Xtables support (required for ip_tables)
[] "CLASSIFY" target support
[] "CONNMARK" target support
[] "DSCP" and "TOS" target support
[] "HL" hoplimit target support
[*] "MARK" target support

v(+)
<Select> < Exit > < Help >

```

Other ALG could be enabled similarly.

**Note:** The feature is only supported on main trunk now.

## 4.45 Cone Nat support

### 4.45.1 How to Enable Cone Nat via config

*make linux\_menuconfig*

-> Networking support (NET [=y])

-> Networking options

-> Network packet filtering framework (Netfilter) (NETFILTER [=y])

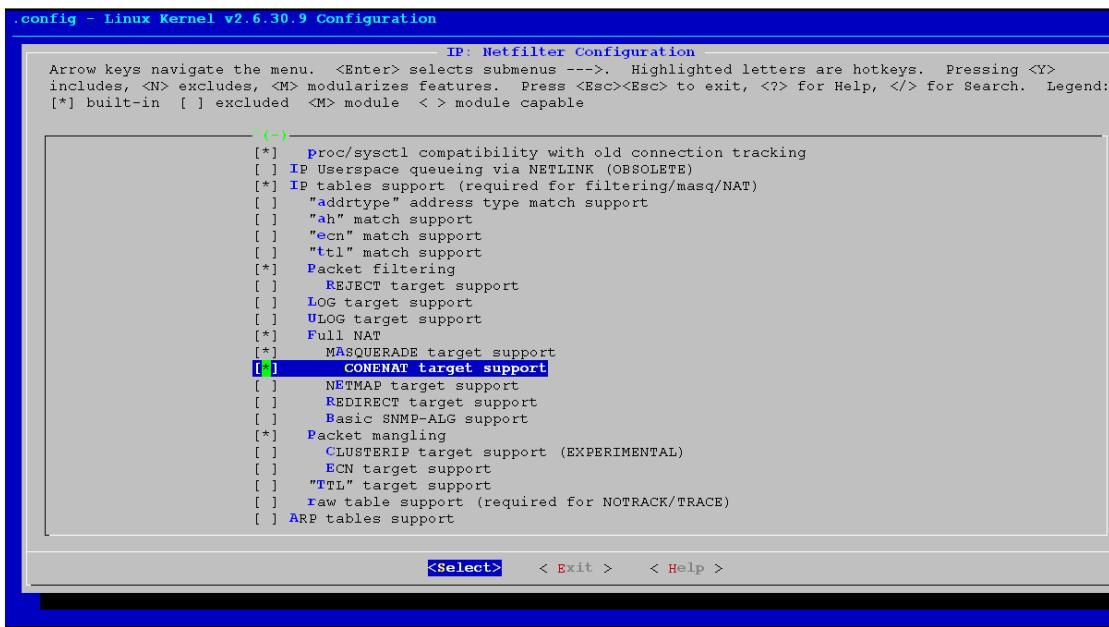
-> IP: Netfilter Configuration

-> IP tables support (required for filtering/masq/NAT) (IP\_NF\_IPTABLES [=y])

-> Full NAT (NF\_NAT [=y])

-> MASQUERADE target support (IP\_NF\_TARGET\_MASQUERADE [=y])

-> IP\_NF\_TARGET\_CONENAT [=y]



#### 4.45.2 How to Enable Cone Nat via proc

/proc/conenat

This proc is used to select the nat type :

1. echo 0 > /proc/conenat, it is symmetric nat;
2. echo 1 > /proc/conenat, it is full cone nat;
3. echo 2 > /proc/conenat, it is restricted cone nat;
4. echo 3 > /proc/conenat, it is port restricted cone nat.

**Note:** The feature is only supported on main trunk now.

#### 4.46 Miniigd IPv6 support

##### 4.46.1 How to Enable Miniigd ipv6 support via config

*make users\_menuconfig*

---Applications

[\*] miniigd

[\*] miniigd ipv6 support

```

----- RLX Linux Configuration -----
the menu. <Enter> selects submenus --->. Highlighted letters are hot
keys, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help,
<Excluded <M> module < > module capable

--- (-) ---
[] mdadm-3.2.5
[] minidlna
[*] dlna_dms
[*] libiconv
[*] miniigd
[] miniigd certification support (NEW)
[*] miniigd ipv6 support
[] mini_upnp

```

## 4.47 WPS supported for Virtual Client in repeater mode

Original WPS is only support for root interface whether wireless is AP mode, Client mode or enable repeater in AP mode. But if you need WPS is support for virtual client (wlanX-vxd) in repeater mode. You can open the flowing compiler flag, the WPS will switch to support virtual client (wlanX-vxd).

1. Turn on compiler flag “CONFIG\_REPEATERO\_WPS\_SUPPORT” and “CONFIG\_WPS\_EITHER\_AP\_OR\_VXD” in apmib.h

When Repeater is disabled, the WPS is working for root AP/Client interface.

When Repeater is enabled, the WPS is working for virtual client interface.

2. Only Turn on compiler flag “CONFIG\_REPEATERO\_WPS\_SUPPORT” in apmib.h

When Repeater is disabled, the WPS is working for root AP/Client interface.

When Repeater is enabled, If virtual client interface is connected, the WPS is working for root interface, If virtual client interface is not connected, the WPS is working for virtual client interface.

## 4.48 Jumbo frame support

### 4.48.1 How to enable jumbo frame support via config

*make linux\_menuconfig*

---device driver

[\*] network device support

[\*] options for realtek soc

[\*] Jumbo frame support

```

Options for Realtek SoC
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted
letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes
features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend:
[*] built-in [] excluded <M> module < > module capable

--- Options for Realtek SoC
 Config MIPS16-Inst Option for Kernel Modules --->
 Config for Layered Driver Features --->
[] Enable iptables rule to RTL ACL rule
[] Enable realtek fast filter
[*] Enable Ethernet Private Skb
[] Enable Dump Ethernet Private Skb Info
[] Enable proc filesystem for debug
[*] Enable JUMBO frame support
[] Enable log print system for debug
[] Enable rome perf
[*] Support rtk vlan feature
[] Support rtk vlan wan tag feature
[] rtk vlan for cable modem
[*] add rtk vlan bridge feature
[] Disable 802.3az EEE feature
v (+)

```

## 4.49 11AC/HS2R1 Logo Test by Sigma Support

### 4.49.1 How to enable jumbo frame support via config

1. Enable Telnetd and Login support in busybox, please follow the instruction of **section 4.33**, it's strongly recommended that set user name "root" without any password. (default setting of AP\_ControlAgent application)

2. Enable Sigma Utility

*make users\_menuconfig*

--- Debug & Test

[\*] sigma utils

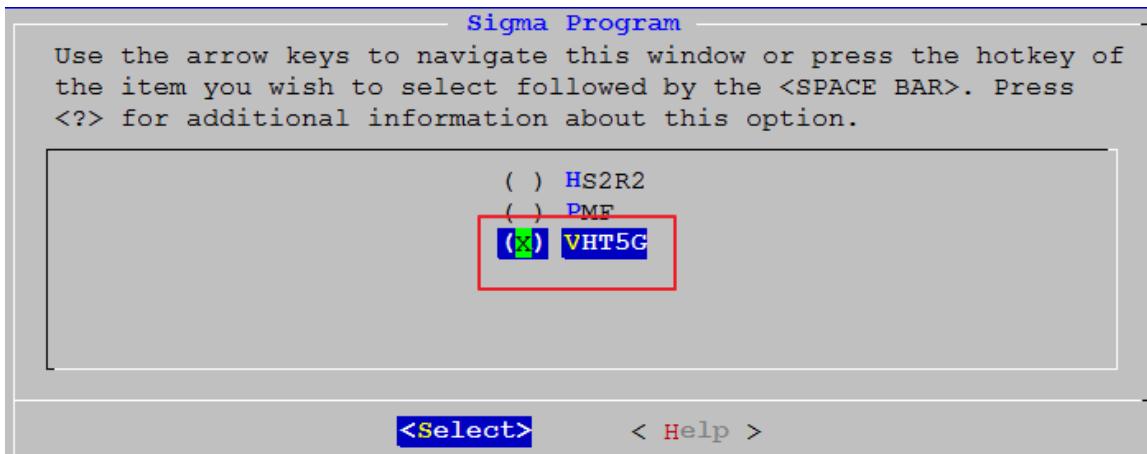
```
----- RLX Linux Configuration -----
the menu. <Enter> selects submenus --->. Highlighted letters are hot-
keys, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help,
*cluded <M> module < > module capable

--- (-) ---
[] pmc utils
[] python support
[] reload
[] routed
[*] rtk_cmd daemon
[] RTK tcp2dev utility
[*] rtk	voip
[] E8C (NEW)
[] RTL8686 IPC User Utility
[*] samba support --->
[] scripts
[] simple config support
[*] snmp
[] squashfs 4.0
[] squashfs 4.2
[] tr069
[] udhcp
[] ulinker autorun image tool
[] updateddd DDNS
[] upImage
[] usbmount
[*] uWiFi
[] vsftpd
[] wireless tools
[] wsc daemon
Libraries Settings --->
[] Debug & test Settings --->
```

```
----- Debug & test Settings -----
the menu. <Enter> selects submenus --->. Highlighted letters are hot-
keys, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help,
*cluded <M> module < > module capable

[] 802.11k
[] cle_shell
[*] Debug tool1
[] dhrystone
[] example
[] iozone
[] iperf
[] ltp
[*] sigma utils
[] Sigma Program (VHT5G) --->
[] utils
```

Select Sigma Program (VHT5G for AC or HS2R1 for HS2.0 Release 1)



#### 4.49.2 How to launch Telnetd

In default, Telnetd daemon will not be launched automatically, user can try command in console  
`telnetd &`

or, user can add such command line to file “sdk/users/script/cinit/init.sh”

`sysconfinit $*`

`telnetd &`

#### 4.49.3 How to use Realtek AP\_ControlAgent for Sigma test

Loading an image with support of sigma utility, login & telentd (root user with no password), APUT can be collocated with AP\_ControlAgent to run sigma test.

Inside of AP\_ControlAgent package we provided, there are documents in the folder named \Document to instruct user how to config & launch this utility.

About the whole setup of 11ac sigma environment, please follow the documents from WFA website. (<http://www.wi-fi.org/>)

### 4.50 Hotspot 2.0 Support

#### 1) CONFIG FILES

Before building image, an engineer needs to modify HS2.0 ANQP Information in CONFIG Files (SDK/ users/hs2/ hs2\_wlan0.conf or SDK/ users/hs2/ hs2\_wlan1.conf). Please refer to WFA Hotspot 2.0 release 1 spec and test plan for how to set the values in CONFIG files. When sigma auto test, CONFIG file will be set automatically by Control Agent.

#### 2) kernel configure

`make linux_menuconfig` //To configure linux kernel

Linux kernel menu as follows:

```
Device Drivers ---> //enter
[*]Network device support ---> //selected and enter
Wireless LAN ---> //enter
 [*]RTL8192C/D 802.11b/g/n support //selected
 [*]Realtek 8192D wireless support //selected
 [*]Realtek HS2.0 Support //selected
```

```
.config - Linux Kernel v2.6.30.9 Configuration
```

```
----- Wireless LAN -----
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>
for Search. Legend: [*] built-in [] excluded <M> module < >
^(-)
[] Efuse Support
[] WAPI Support
[] Config File support
[] Wireless Tools v29 support
[] Realtek P2P support
[*] Realtek wps2.0 support
[*] Clock Source, Select is 40MHz, otherwise 25MHz
[] Band Edge Limit support for 92C/92D/8812/88E/92E/8881A
[] RTL Mesh Support
[*] Realtek HS2.0 support

<Select> < Exit > < Help >
```

## 2) The user can enable/disable HS2 on web UI.

In Hotspot 2.0 spec, AP supports HS2 function only when the security type of AP is set to 802.1X. So, before enabling HS2, users need to set Security Type to 802.1X mode.

web UI menu as follows:

```
Wireless --->
wlan1 (5GHz) / wlan2(2.4GHz) --->
 Security --->
```



## WLAN Access Point

- Site contents:
  - > Setup Wizard
  - > Operation Mode
  - > Wireless
    - > BandMode
    - > wlan1(5GHz)
      - > Basic Settings
      - > Advanced
      - > Security
      - > Access Control
      - > WDS settings
      - > Site Survey
      - > WPS
      - > Schedule
    - > wlan2(2.4GHz)
  - > TCP/IP Settings
  - > Firewall
  - > QoS
  - > Route Setup
  - > Management

### Wireless Security Setup -wlan1

This page allows you setup the wireless security. Turn on WEP or WPA by using Encryption Keys could

Select SSID: Root AP - NEC AP 5G

Encryption:

Authentication Mode:  Enterprise (RADIUS)  Personal (Pre-Shared Key)

WPA2 Cipher Suite:  TKIP  AES

RADIUS Server IP Address:

RADIUS Server Port:

RADIUS Server Password:

The ON/OFF for HS2 setting

web UI menu as follows:

Wireless --->  
wlan1 (5GHz) / wlan2(2.4GHz) --->  
Advanced --->  
enable / disable HS2

The screenshot shows the 'WLAN Access Point' interface. On the left is a navigation tree with options like Site contents, Setup Wizard, Operation Mode, Wireless (selected), BandMode, wlan1(5GHz), wlan2(2.4GHz), TCP/IP Settings, Firewall, QoS, Route Setup, and Management. The 'Wireless' section is expanded, showing sub-options: Basic Settings, Advanced, Security, Access Control, WDS settings, Site Survey, WPS, Schedule, and wlan1(5GHz). Under wlan1(5GHz), 'HS2' is listed. The main right panel is titled 'Wireless Advanced Settings -wlan1'. It contains several configuration fields with radio button options: Fragment Threshold (2346), RTS Threshold (2347), Beacon Interval (100), IAPP (Enabled), HS2 (Enabled, highlighted with a red border), Protection (Enabled), Aggregation (Enabled), Short GI (Enabled), WLAN Partition (Enabled), STBC (Enabled), LDPC (Enabled), TX Beamforming (Enabled), Multicast to Unicast (Enabled), and RF Output Power (100%, 70%, 50%, 35%, 15%). At the bottom are 'Apply Changes' and 'Reset' buttons.

## 4.51 Hotspot 2.0 Release 1 Logo Test

- 1) Please follow 4.50-2 to enable Hotspot 2.0 during compile time
- 2) For logo test, there are several features have to be enabled as well: (Or follow 4.33)
  - I. Remote login DUT through ssh protocol
    - i. Enable telnetd through menuconfig of busyboxPlease reference snapshot below to enter menuconfig of busybox

```

--- select components
 Selected Target (rtl819xD) -->
 Selected Kernel (linux-2.6.30) -->
 Selected Busybox (busybox-1.13) -->
 Selected toolchain (rsdk-1.5.5-5281-EB-2.6.30-0.9.30.3-110714) -->
--- rtl819xD
 Selected Board Configuration (97D+8367R+92C+8812 GW) -->
--- config components
[] Config kernel (NEW)
[] Config users (NEW)
[?] Config busybox
[] Load default settings
[] Save default settings

 Load an Alternate Configuration File
 Save an Alternate Configuration File

```

, save and exit. Then enable *login* service within “Login/Password Management Utilities” and *telnetd* service within “Networking Utilities”. Please reference snapshots below.

```

[] adduser
[] deluser
[] getty
[] support utmp file
[] support wtmp file
[?] login
[] Support for PAM (Pluggable Authentication Modules)
[] Support for login scripts
[*] Support for /etc/nologin
[*] Support for /etc/security
[] passwd
[] cryptpw
[] chnassmd

```

```

[*] route
[] zattach
[] telnet
[?] telnetd
[*] Support standalone telnetd (not inetd only)
[] ftp
[] ftppd
[] traceroute

```

And modify the default administration control configuration to enable *root* remote login. The configuration file is located at SDK/boards/rtl819xD/etc.default/passwd (The location is for example while platform is RTL8197D SoC with both 8812 and 92E wireless chips). Change the line from “root:x:0:0:root:/bin/sh” to “root::0:0:root:/bin/sh”

Finally, recompile SDK. To make sure, after firmware upload to DUT, telnetd should be located through console command. And /etc/passwd carry the correct configuration.

ii. Enable configuration relay utility *sigma\_util*

Enter menuconfig of user space applications as below:

```

--- select components
 Selected Target (rtl819xD) --->
 Selected Kernel (linux-2.6.30) --->
 Selected Busybox (busybox-1.13) --->
 Selected toolchain (rsdk-1.5.5-5281-EB-2.6.30-0.9.30.3-110714) --->
--- rtl819xD
 Selected Board Configuration (97D+8367R+92C+8812 GW) --->
--- config components
[] CConfig kernel
[*] Config users
[] CConfig busybox
[] LLoad default settings
[] SSave default settings

 Load an Alternate Configuration File
 Save an Alternate Configuration File

```

and Exit.

Select *sigma\_utils* to support *HS2R1*

```

[] dlna_dms
[] minidlna
[] mt-daapd
[*] sigma utils
 Sigma Program (HS2R1) --->
[] capwap

```

*Note, if ever build SDK to support 802.11 AC logo test, make sure to rebuild directory of /users/sigma\_utils.*

## II. Release Control Agent to Lab

*Note, please make sure test Lab modify control agent follow rules of:*

1. HESSID do not include colon
2. AP's MAC Address keep FF:FF:FF:FF:FF:FF

## 4.52 Multiple repeater mode and multiple mac clone

### Multiple repeater mode

Turn on compile flag “CONFIG\_RTL\_MULTI\_REPEAT\_MODE\_SUPPORT” from config file or make menuconfig as below Fig.

Turn on the feature from **web page** as follows

Step 1: In web UI, enter “basic setting”->“multiple AP/multiple repeater mode”

**Wireless Basic Settings - wlan1**

This page is used to configure the parameters for wireless LAN clients which may connect to your Access Point. Here you may change wireless encryption settings as well as wireless network parameters.

Disable Wireless LAN Interface

Band: 5 GHz (A+N)

Mode: AP  
MultipleAP-MultipleRepeater (highlighted with a red box)

Network Type: Infrastructure

SSID: RTK 11n AP 5G

Add to Profile

Channel Width: 40MHz

Control Sideband: Auto

Channel Number: 44

Broadcast SSID: Enabled

WMM: Enabled

Data Rate: Auto

Step 2: enable the check box of “Enabled multiple-repeater mode”, then root and vap0 interfaces, e.g. wlan0 and wlan0-va0, will be enabled at AP mode and vap1 and vap2, e.g. wlan0-va1 and wlan0-va2, will be enabled under STA mode.

Note: Don't enable wlan0-va3 and wlan0-vxd when multiple-repeater is enabled;

| No. | Enable                              | Band        | SSID         | Data Rate | Broadcast SSID | WMM     | Access  | Tx Restrict (Mbps) | Rx Restrict (Mbps) | Active Client List | WLAN mode |
|-----|-------------------------------------|-------------|--------------|-----------|----------------|---------|---------|--------------------|--------------------|--------------------|-----------|
| AP1 | <input checked="" type="checkbox"/> | 5 GHz (A+N) | RTK 11n AP V | Auto      | Enabled        | Enabled | LAN+WAN | 0                  | 0                  | Show               | AP        |
| AP2 | <input checked="" type="checkbox"/> | 5 GHz (A+N) | RTK 11n AP V | Auto      | Enabled        | Enabled | LAN+WAN | 0                  | 0                  | Show               | STA       |
| AP3 | <input checked="" type="checkbox"/> | 5 GHz (A+N) | RTK 11n AP V | Auto      | Enabled        | Enabled | LAN+WAN | 0                  | 0                  | Show               | STA       |
| AP4 | <input type="checkbox"/>            | 5 GHz (A+N) | RTK 11n AP V | Auto      | Enabled        | Enabled | LAN+WAN | 0                  | 0                  | Show               | AP        |

Enable Multiple Universal Repeater Mode (Acting as AP and client simultaneously); root as AP, vif0 be disable, vap1 as AP, vap2 as STA, vap3 as STA root and vap2 at repeater1, vap1 and vap3 at repeater2

Apply Changes Reset

Step3: Fill SSID text fields at AP2, i.e. repeater 1's STA interface (R1\_STA) and AP3, i.e. repeater2's STA interface (R2\_STA) where the SSID texts should be the SSIDs that R1\_STA and R2\_STA try to connect, if the target remote AP has enabled encryption, setting security at security

page.

**WLAN Access Point**

**Multiple APs**

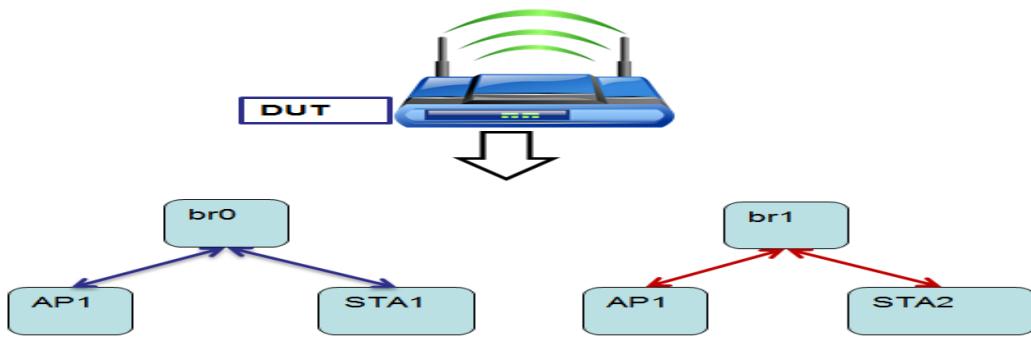
This page shows and updates the wireless setting for multiple APs.

| No. | Enable                              | Band        | SSID         | Data Rate | Broadcast SSID | WMM     | Access  | Tx Restrict (Mbps) | Rx Restrict (Mbps) | Active Client List | WLAN mode |
|-----|-------------------------------------|-------------|--------------|-----------|----------------|---------|---------|--------------------|--------------------|--------------------|-----------|
| AP1 | <input checked="" type="checkbox"/> | 5 GHz (A+N) | RTK 11n AP V | Auto      | Enabled        | Enabled | LAN+WAN | 0                  | 0                  | Show               | AP        |
| AP2 | <input checked="" type="checkbox"/> | 5 GHz (A+N) | RemoteAP1    | Auto      | Enabled        | Enabled | LAN+WAN | 0                  | 0                  | Show               | STA       |
| AP3 | <input checked="" type="checkbox"/> | 5 GHz (A+N) | RemoteAP2    | Auto      | Enabled        | Enabled | LAN+WAN | 0                  | 0                  | Show               | STA       |
| AP4 | <input type="checkbox"/>            | 5 GHz (A+N) | RTK 11n AP V | Auto      | Enabled        | Enabled | LAN+WAN | 0                  | 0                  | Show               | STA       |

Enable Multiple Universal Repeater Mode (Acting as AP and client simultaneously); root as AP, vif0 be disable, vap1 as AP, vap2 as STA, vap3 as STA root and vap2 at repeater1, vap1 and vap3 at repeater2

**Apply Changes** **Reset**

Step4: Under multiple repeater mode, we need to setup dual bridge interface and binding the AP1 and STA1 under br0, AP2 and STA2 under br1, the topology as below figure



#### Multiple mac clone:

Turn on this feature by enabled define "RTL\_MULTI\_CLONE\_SUPPORT" from config file or make menuconfig.

Support chip: 8192E

Multiple mac clone feature is usually enabled at repeater mode; when STAs connect to repeater AP and the traffic pass through from RSTA (repeater STA) to remote AP, RSTA will use a mac address different with RSTA's mac to connect to remote AP, the cloned mac address is followed as the following rule:

E.g. RSTA's mac Addr is 00e04c112233 then 1<sup>st</sup> cloned mac addr should be 00e000112233, and 2<sup>nd</sup> mac addr should be 00e001112233, at 8192E the most cloned mac mount is 16.

#### **4.53 Support Russian L2TP special features**

#### **4.53.1 Russian L2TP special features included the following:**

1. When wan type is DHCP+L2TP, and DHCP completes but L2TP fails, then it uses eth1 as the wan interface, and the default gateway and firewall are correct;
  2. When L2TP disconnects, it should reconnects successfully;

3. When L2TP daemon is killed manually, use eth1 as wan interface, and the default gateway and firewall are correct;
4. When unplug the network cable of wan interface after L2TP connected, it must disconnect and restore the case that using eth1 as wan interface;
5. Russian ISP L2TP server's IP is the same with the PPP server's IP, so it should add a route to L2TP server through eth1 interface;
6. Support DHCP option 33, 121, 249 to set static IP, and option 57 to set maximum message size;
7. DNS servers got from DHCP procedure are internal, DNS servers got from L2TP procedure are external, and the internal DNS servers have high priority to parse the DNS query from client;
8. Russia has high network bandwidth, the speed of BT testing reaches to 5MB/s. So we send the LCP-Echo-Reply packet in fastpath to avoid L2TP disconnect due to UDP socket congestion.

#### **4.53.2 How to enable the patches to support these features**

*make linux\_menuconfig*

- > Networking support (NET [=y])
- > Networking options
  - > Support Russia features ([=n])

#### **4.53.3 What's the platform could be supported**

All the patches are limited by platform, so our existed platforms are supported.

#### **4.54 WIFI-DIRECT (WIFI-P2P) support**

Turn on compile flag “CONFIG\_RTL\_P2P\_SUPPORT” from config file or make menuconfig as below Fig.WIFI-DIRECT only support for 2.4G band chip for example 92e,92c,88e,8881a selectable now.

-make menuconfig-> [ ] Config kernel (NEW)-> Device Drivers-> Network device support-> Wireless LAN-> [\*] Realtek P2P



When enabled wifi-direct need to turn system configuration

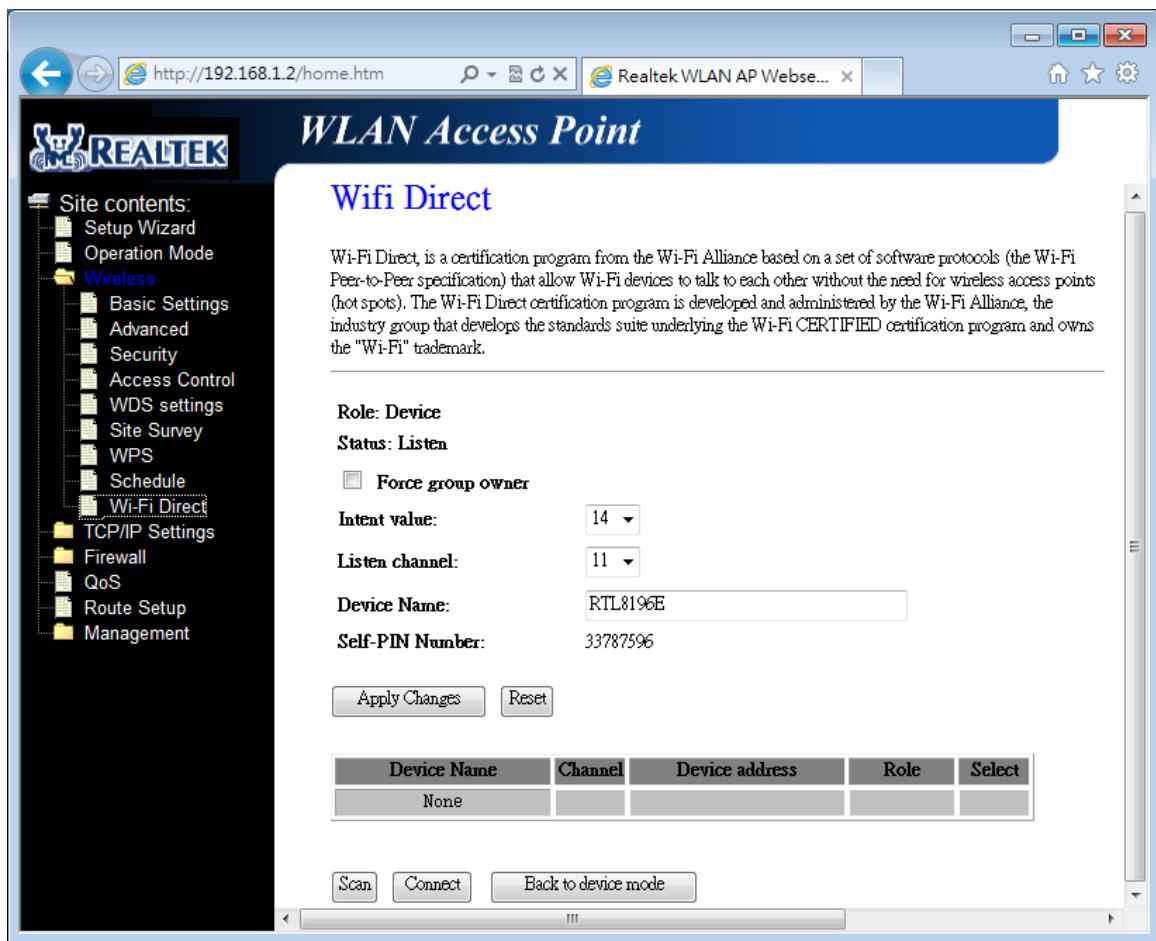
(0x31000) linux image offset in flash.

(0x160000) root image offset in flash.

The screenshot shows a terminal window titled "172.21.69.111 - PuTTY". Inside, it displays the "Linux Kernel v2.6.30.9 Configuration" menu. The user is navigating through the "System Configuration" section. The current option highlighted is "(0x160000) root image offset in flash.", which is described as "Root image offset,should be 64K alignment. \*\*\*". Other options visible include "linux image offset,should be 4K alignment. \*\*\*" at address 0x31000, and "Flatfs image offset,should be 128K alignment. \*\*\*" at address 0x100000. The menu also includes sections for "System Type (96D+92D GW)", "Watchdog timer support", "USB3G support", "Webpage support", and "Flash Mapping". A legend at the top right indicates that "x" marks selected items and "x x" marks excluded items.

Turn on P2P mode by set flash mib” flash set WLAN0\_MODE 8”, or configure from web page as below

The screenshot shows a web browser window titled "Realtek WLAN AP Webse...". The main title bar says "WLAN Access Point". The left sidebar menu includes "Site contents:", "Setup Wizard", "Operation Mode", "Wireless" (selected), "Advanced", "Security", "Access Control", "WDS settings", "Site Survey", "WPS", "Schedule", "Wi-Fi Direct", "TCP/IP Settings", "Firewall", "QoS", "Route Setup", and "Management". The "Wireless" section is expanded, showing "Basic Settings", "Advanced", "Security", "Access Control", "WDS settings", "Site Survey", "WPS", "Schedule", and "Wi-Fi Direct". Under "Basic Settings", the "Mode" dropdown is set to "P2P". Other settings visible include "Band: 2.4 GHz (B+G+N)", "Network Type: P2P" (selected from a dropdown), "SSID: P2P", "Channel Width: 20MHz", "Control Sideband: Upper", "Channel Number: 11", and "Broadcast SSID: Enabled". A note at the top states: "This page is used to configure the parameters for wireless LAN clients which may connect to your Access Point. Here you may change wireless encryption settings as well as wireless network parameters."



P2P control page as above fig

- The check box “force to group owner” to switch DUT’s role to GO mode force, then push “apply change” button to apply.
- change Intent value or listen channel or device name and push “apply change” button to apply.
- The button “back to device mode” can force switch role to DEV mode
- Under DEV mode push the button “scan” to do P2P discovery and result will list in the table of web page
- After P2P discovery you choose some DEV or GO from discovery result and push the button “connect” to do P2P connect.

When DUT under GO mode or DEV mode, DUT can do P2P connection by passive mode that without web UI intervened.

## 4.55Realtek Simple Config support

Realtek Simple Config is a method to simplify the WLAN profile configuration procedure:Use APP on smart-mobile/tablet/PC... to assist Wi-Fi device to join the WLAN network.

### 4.55.1 How to Enable Simple Config

#### 1) kernel configure

*make linux\_menuconfig* //To configure linux kernel

Linux kernel menu as follows:

Device Drivers ---> //enter  
[\*]Network device support ---> //selected and enter  
Wireless LAN ---> //enter  
[\*]RTL8192C/D 802.11b/g/n support //selected  
[\*] Realtek Simple Config Support //selected

```
Wireless LAN
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys.
Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help,
</> for Search. Legend: [*] built-in [] excluded <M> module < > module capable

^ (-)
[*] Use PCIe slot 0 WiFi device
 PCIe Slot 0 device (Realtek 8192EE wireless support) --->
[] PCIe slot 0 Enable external high power PA
[*] PCIe slot 0 Enable Tx Beamforming
[] PCIe slot 0 Enable Antenna Diversity
[*] Private skb buffer management
[*] Virtual AP Support
[*] Client Mode Support
[*] Repeater Mode support
[] multiple Repeater Mode support
[] multiple clone support
[] Client Mode 802.1x Support
[*] Multiple AP profile Support
[*] WDS Support
[] Efuse Support
[] WAPI Support
[] Config File support
[] Wireless Tools v29 support
[] MP quick PSD support
[] Realtek P2P support
[*] Realtek wps2.0 support
[*] Band Edge Limit support for 92C/92D/8812/88E/92E/8881A
[] RTL Mesh Support
[] Enable WLAN Dos Filter
[] Realtek HS2.0 support
[*] Realtek Simple Config Support
v (+)

<Select> < Exit > < Help >
```

#### 2) user application configure

*make users\_menuconfig* //To configure user application

```

RLX Linux Configuration
the menu. <Enter> selects submenus --->. Highlighted letters are hot
keys, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help,
selected <M> module < > module capable

-- (-) --
[] pmc utils
[] python support
[] reload
[] routed
[*] rtk_cmd daemon
[] RTK tcp2dev utility
[*] rtk	voip
[] E8C (NEW)
[] RTL8686 IPC User Utility
[*] samba support --->
[] scripts
[!] simple config support
[*] snmp

```

#### 4.55.2 How to test Simple Config

The device MUST be in client mode, and the MIB SC\_ENABLED MUST be set to 1.

Please refer to [Android\\_Simple\\_Config\\_User\\_Guide\\_v0.x.pdf](#) or

[IOS\\_Simple\\_Config\\_User\\_Guide\\_v0.x.pdf](#) for more details.

#### 4.56 Realtek Bluetooth support

Realtek Bluetooth BSP and Driver are supported by the SDK.

For 96e, toolchain rsdk-4.4.7-4181-EB-2.6.30-0.9.30-m32u-140129([for kernel 2.6.30](#)) or  
rsdk-4.6.4-4181-EB-3.10-u0.9.33-m32-150324 ([for kernel 3.10](#)) should be used.

For 96D/97D/8881A, toolchain rsdk-1.5.10-5281-EB-2.6.30-0.9.30-m32ub-130429([for kernel 2.6.30](#)) or rsdk-4.6.4-5281-EB-3.10-0.9.33-m32ub-20141111([for kernel 3.10](#)) should be used.

Please decompress the toolchain tar ball at SDK toolchain directory.

##### 4.56.1 Enable Bluetooth Support

###### Kernel menuconfig

```
[*] Enable the block layer --->
 Bus options (PCI, PCMCIA, EISA, ISA, TC) --->
 Executable file formats --->
 Power management options --->
[*] Networking support --->
 Device Drivers --->
 Firmware Drivers --->
 File systems --->
 ...
--- Networking support
 Networking options --->
[] Amateur Radio support --->
[] CAN bus subsystem support --->
[] IrDA (infrared) subsystem support --->
[*] Bluetooth subsystem support --->
[] RxRPC session sockets
-*-
[] Wireless --->
[] WiMAX Wireless Broadband support --->
[] RF switch subsystem support --->
[] Plan 9 Resource Sharing Support (9P2000) --->
[] CAIF support --->
[] Ceph core library
[] NFC subsystem support --->
```

Bluetooth subsystem support for kernel 3.10:

```
--- Bluetooth subsystem support
[*] RFCOMM protocol support
[*] RFCOMM TTY support
[*] BNEP protocol support
[*] Multicast filter support
[*] Protocol filter support
[*] Bluetooth device drivers --->
```

Bluetooth subsystem support for kernel 2.6.30:

```
--- Bluetooth subsystem support
[*] L2CAP protocol support
[] SCO links support
[*] RFCOMM protocol support
[*] RFCOMM TTY support
[*] BNEP protocol support
[*] Multicast filter support
[*] Protocol filter support
[*] Bluetooth device drivers --->
```

```
[] HCI USB driver
[*] RTK HCI USB driver
[*] HCI UART driver
[] UART (H4) protocol support
[*] BCSP protocol support
[] Atheros AR300x serial support
[*] Realtek H5 protocol support
[] HCILL protocol support
[] Three-wire UART (H5) protocol support
[] HCI BCM203x USB driver
[] HCI BPA10x USB driver
[] HCI BlueFRITZ! USB driver
[] HCI VHCI (Virtual HCI device) driver
[] Marvell Bluetooth driver support
```

### Users menuconfig:

Select bluez and the bluetooth chip

```
RLX Linux Configuration
the menu. <Enter> selects submenus --->. Highlighted letters are hot
keys, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help,
<Excluded> <M> module < > module capable

--- Applications Settings
 BT Configuration --->
 File_system Configuration --->
 IPv6 Configuration --->
 PPP Configuration --->
 RTK_inband Configuration --->
 USB3G Configuration --->
 [] alsa related
 [] auth
 [*] bluez
 Bluez Version (Bluez 4.96) --->
 [*] RTK bluetooth firmware selection
 [*] SERIAL bluetooth
 Serial Bluetooth chip (rtl8761a) --->
 [*] USB Bluetooth chip
 [*] rtl8761a
```

Add mdev in by busybox menconfig if usb bluetooth used

```
Linux Ext2 FS Progs --->
Linux Module Utilities --->
Linux System Utilities --->
Miscellaneous Utilities --->
Networking Utilities --->
Print Utilities --->
```

```
[] ipcs
[] losetup
[*] mdev
[*] Support /etc/mdev.conf
[*] Support subdirs/symlinks
[*] Support regular expressions substitutions when renaming device
[*] Support command execution at device addition/removal
[*] Support loading of firmwares
[] mkswap
```

#### 4.56.2 Enable USB for USB Bluetooth Support

USB config for XHCI (USB 3.0):

```
[*] USB support --->
-- USB support
[] OHCI support
[] EHCI support
[*] XHCI support
[*] Support for Host-side USB
[] USB verbose debug messages
[] USB announce new devices
 *** Miscellaneous USB options ***
[*] Enable USB persist by default
[] Dynamic USB minor allocation
[] Rely on OTG Targeted Peripherals List
[] Disable external hubs
[] USB Monitor
[] Support WUSB Cable Based Association (CBA)
 *** USB Host Controller Drivers ***
[] Cypress C67x00 HCD support
[*] xHCI HCD (USB 3.0) support
[*] Support for XHCI USB controller on platform
[] Debugging for the xHCI host controller
[] OXU210HP HCD support
[] ISP116X HCD support
[] ISP 1760 HCD support
[] ISP1362 HCD support
[] UHCI HCD (most Intel and VIA) support
[] SL811HS HCD support
v (+)
```

## USB config for EHCI/OHCI/OTG (USB 2.0):

(For 96E, you should enable DWC\_OTG support)

```
-- USB support
[*] Support for Host-side USB
[] USB verbose debug messages
[] USB announce new devices
 *** Miscellaneous USB options ***
[] USB device filesystem
[*] USB device class-devices (DEPRECATED)
[] Dynamic USB minor allocation
[] Rely on OTG Targeted Peripherals List
[] Disable external hubs
[*] USB Monitor
[] Enable Wireless USB extensions (EXPERIMENTAL)
[] Support WUSB Cable Based Association (CBA)
 *** USB Host Controller Drivers ***
[] Cypress C67x00 HCD support
[*] EHCI HCD (USB 2.0) support
[] Root Hub Transaction Translators
[] Improved Transaction Translator scheduling (EXPERIMENTAL)
[] OXU210HP HCD support
[] ISP116X HCD support
[] ISP 1760 HCD support
[*] OHCI HCD support
[] SL811HS HCD support
[] R8A66597 HCD support
[] Host Wire Adapter (HWA) driver (EXPERIMENTAL)
```

## 4.56.3 Enable UART for UART Bluetooth Support (now only 96e/97DL can support UART

### Bluetooth)

Device Drivers --->

Character devices --->

Serial drivers --->

Note: Number of 8250/16550 serial port to register at runtime should be changed to 2

```
[*] 8250/16550 and compatible serial support
[*] Console on 8250/16550 and compatible serial port
(2) Maximum number of 8250/16550 serial ports
(2) Number of 8250/16550 serial ports to register at runtime
[] Extended 8250/16550 serial driver options
 *** 8250 compatible port support ***
[*] 819x RTL UART1 support
[] SC16IS7x0 series (I2C bus) support
```

## 4.56.4 How to check if bluetooth work

If usb bluetooth chip used, the following command should be issued before bring the device up:

```
echo /bin/mdev > /proc/sys/kernel/hotplug
mount -t sysfs sysfs /sys
echo 1 > /sys/class/firmware/timeout
```

If uart bluetooth chip used, the following command should be issued to bring up the device:

```
hciattach -n -s 115200 /dev/ttyS1 rtk_h5 115200 &
```

verify by bluez command:

```
hciconfig hci0 up
hciconfig hci0 iscan /*can be seen by other bluetooth device */
hcitool scan /*scan other bluetooth device*/.
l2ping xx:xx:xx:xx:xx:xx /*if can ping through*/
```

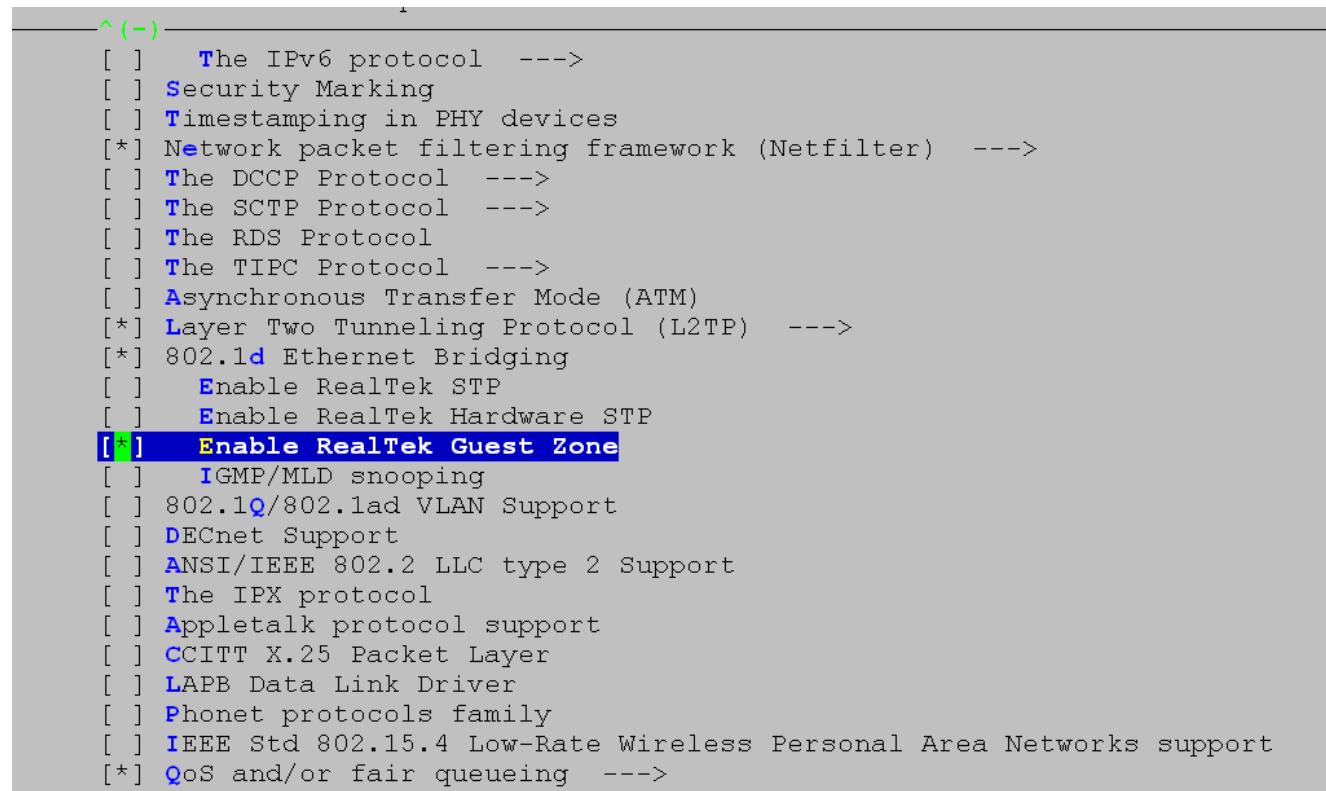
## 4.57 Realtek bridge guest zone support(for kernel 3.10)

### 4.57.1 How To enable bridge guest zone

#### Kernel config

[\*] Networking support --->

Networking options --->



```
^ (-)
[] The IPv6 protocol --->
[] Security Marking
[] Timestamping in PHY devices
[*] Network packet filtering framework (Netfilter) --->
[] The DCCP Protocol --->
[] The SCTP Protocol --->
[] The RDS Protocol
[] The TIPC Protocol --->
[] Asynchronous Transfer Mode (ATM)
[*] Layer Two Tunneling Protocol (L2TP) --->
[*] 802.1d Ethernet Bridging
[] Enable RealTek STP
[] Enable RealTek Hardware STP
[!] Enable RealTek Guest Zone
[] IGMP/MLD snooping
[] 802.1Q/802.1ad VLAN Support
[] DECnet Support
[] ANSI/IEEE 802.2 LLC type 2 Support
[] The IPX protocol
[] Appletalk protocol support
[] CCITT X.25 Packet Layer
[] LAPB Data Link Driver
[] Phonet protocols family
[] IEEE Std 802.15.4 Low-Rate Wireless Personal Area Networks support
[*] QoS and/or fair queueing --->
```

#### 4.57.2 How to configure bridge guest zone

commands:

|                 |                         |                                                             |
|-----------------|-------------------------|-------------------------------------------------------------|
| setzone         | <bridge> <device> <val> | set zone type (0: host, 1: guest, 2: gateway) for interface |
| setzoneisolate  | <bridge> <value>        | set zone isolation (0: no, 1: yes)                          |
| setguestisolate | <bridge> <value>        | set guest isolation (0: no, 1: yes)                         |
| chkguestmac     | <bridge> <verb> <mac>   | check if mac addrs of client is come from guest zone        |
| setlockclient   | <bridge> <mac>          | set mac addrs of locked client list                         |
| showguestinfo   | <bridge>                | show zone and locked client info                            |
| setgatewaymac   | <bridge> <mac>          | set mac addr of gateway                                     |

We divide all bridge ports to 3 kinds. The first one is host port. The traffic of devices connecting to host ports is not restricted. The second one is guest port. The traffic of devices connecting to guest ports is restricted to and from gateway only. The third one is gateway port. The gateway device is expected to connect to the gateway port.

Using brctl utility with command “setzone” can set port type. For example, we assign SSID4 as guest port and we connect gateway device on eth0. The commands are:

```
brctl setzone br0 wlan1-va2 1
brctl setzone br0 eth0 2
```

Other ports that are not set are host ports by default.

If we don't wish the traffic communicated between host ports and guest ports, we can use command “setzoneisolate” to give restriction. The command is:

```
brctl setzoneisolate br0 1
```

If we don't wish the traffic communicated between guest ports, we can use command “setguestisolate” to give restriction. The command is:

```
brctl setguestisolate br0 1
```

Because we want to let device connecting to guest ports communicate with gateway device, but we don't want to let device connecting to guest ports communicate with devices which are also connecting to the same port as gateway device, we need to know the MAC address of gateway device. Then if a packet is from guest port and going to gateway device, we will allow it. But if a packet is from guest port and going to other devices connecting to gateway port, we will drop it. Using command “setgatewaymac” to give the MAC address of gateway device:

```
brctl setgatewaymac br0 001a92783ecf
```

There is also a command “showguestinfo” showing the settings or status. Using this command can get the current status:

```
brctl showguestinfo br0
```

The content printed is as following:

```
zone isolation: 1
guest isolation: 1
eth0: gateway
wlan0: host
wlan1: host
wlan0-va0: host
wlan0-va1: host
wlan0-va2: host
wlan1-va0: host
wlan1-va1: host
wlan1-va2: guest
locked client no: 0
gateway mac=00:1a:92:78:3e:cf
```

The other two commands of “chkguestmac” and “setlockclient” are not used in WLAN partition function.

We take an example. In some kind of usage scenario, there are 4 SSIDs on 5G band and 4 SSIDs on 2.4G band. The device is configured as bridge mode. All Ethernet ports are configured as LAN, including eth0 and eth1. A gateway device is connecting to eth1. We wish smart phones connecting to a restricted area, i.e. SSID4 and SSID8. These phones can communicate with gateway device, but can't communicate with devices connecting to other SSIDs or eth0. For this kind of application, we can configure bridge module after bridge module is activated as following:

```
brctl setzone br0 wlan0-va2 1
brctl setzone br0 wlan1-va2 1
brctl setzone br0 eth1 2
brctl setzoneisolate br0 1
brctl setguestisolate br0 1
brctl setgatewaymac br0 001a92783ecf
```

## 4.58 2.4G -wlan0/5G-wlan1 Support(for kernel 3.10)

Support 2.4G at wlan0 and 5G at wlan1.

#### 4.58.1 How To enable 2.4G -wlan0/5G-wlan1

Kernel config

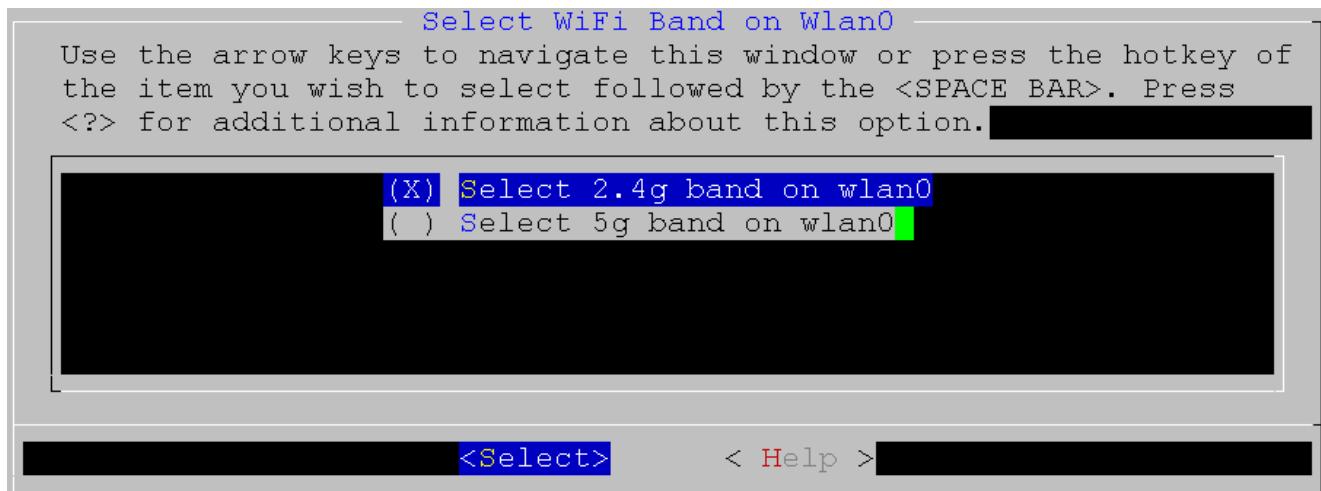
Device Drivers --->

[\*] Network device support --->

Wireless LAN --->

Select WiFi Band on Wlan0 (Select 2.4g band on wlan0) --->

(X) Select 2.4g band on wlan0



#### 4.58.2 How to Test 2.4G -wlan0/5G-wlan1

In order to check whether 2.4G at wlan0, you can open Web UI or get wlan related mib value at AP console .

### 4.59 GIGA IOL Test

For RTL8198 platform:

Set to test mode 1: echo test 1 > /proc/ phyRegTest

Set to test mode 4 for port 0 ~ 4: echo test 4 [port] > /proc/ phyRegTest

You need to reboot the DUT when you finish any one of 1000M test mode then redo next one.

For RTL8198C platform:

Set to test mode 1: echo test 1 > /proc/ phyRegTest

Set to test mode 4 for port 0 ~ 4: echo test 4 [port] > /proc/ phyRegTest

Set to normal mode: echo test 0 > /proc/ phyRegTest

For RTL8197D+RTL8367RB or RTL8881AB+RTL8367RB platform:

`echo 8367test [phy_id] [test_mode] > /proc/rtl865x/phyReg`

phy\_id: 0/1/2/3/4

test\_mode: 0/1/4 (0: normal mode/1: mode1/4:mode4)

If you want to change to different port, you need to set previous port to normal mode first.

For example:

i) 1st power on

ii) phyid 3 test mode 1

`echo 8367test 3 1 > /proc/rtl865x/phyReg`

iii) phyid 3 test mode 4

`echo 8367test 3 4 > /proc/rtl865x/phyReg`

iv) before you change to phyid2, you must set phyid3 to normal mode

`echo 8367test 3 0 > /proc/rtl865x/phyReg`

v) phyid 2 test mode 1

`echo 8367test 2 1 > /proc/rtl865x/phyReg`

## 4.60 NAND FLASH support

### 4.60.1 nand flash support bootloader.

example for 8198C:

8198C nand flash board have it's own independent bootloader config.just config the bootloader as <2.2 section> describe. The key config step describe blow:

step1: enter bootloader source code.

step2: select config file:

`cp def-rtl8198c-8954e-nand-config .config`

step3: appoint toolchain:

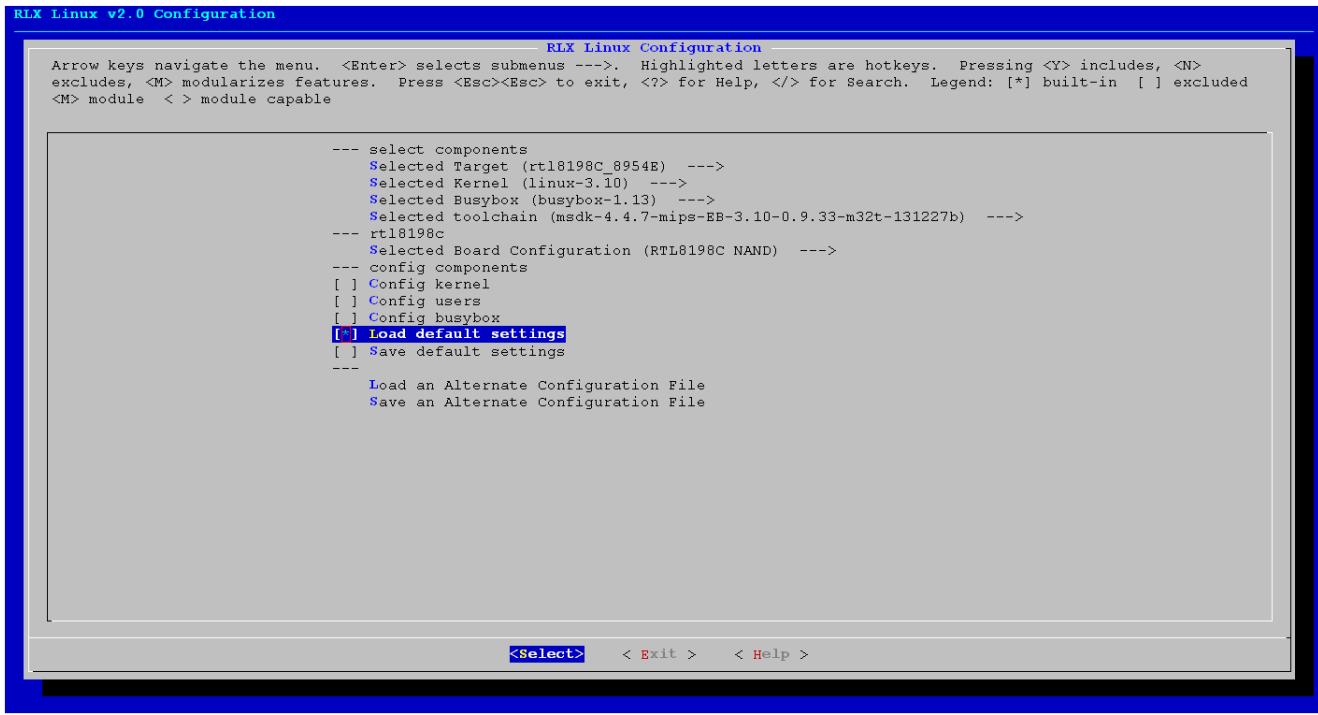
`PATH=$PATH: ./rtl819x/toolchain/msdk-4.4.7-mips-EB-3.10-0.9.33-m32t-131227b/bin`

step4: run command: `make menuconfig`. Load the default config

step5: choose ‘<Exit>’ and click ‘Enter’, save the new configuration. than *make*.

the bootloader for 8198C nand flash will generate in *bicode/boot.bin*

#### 4.60.2 compile nand flash support rlxlinux image.



sdk provide independent configuration for nand flash 8198C board,just run command *make menuconfig*, select config like the picture above show, and select <*load default setting*>,than *make*. the detail config process description same with <2.3 section>,Image will generate in folder *image/* directory. you can upload image like the <3.3 section> describe.

#### 4.60.3 How to generate image for nand flash e-WriterPro

Image for nand flash e-WriterPro is the image can be burned to nand flash directorly by nand flash e-WriterPro. the image generator tool location is <users/bch6\_eccenc>,run command like below:

First, compiler the image generate tool for nand flash e-WriterPro.

a> cd users/bch6\_eccenc

b> make

c> image generator tool will generate with name “eccenc”

second,generate image for nand flash e-WriterPro.

d> prepare three files blow, copy the three files to the directory same with eccenc program.

*boot* (note: not *boot.bin*, locate in *rtl819x-bootcode-SDK-v3.4.7-98C/btcode/* directory),

*linux.bin* (locate in *rtl819x/image/* directory)

*squashfs.o* (locate in *rtl819x/target/* directory).

```
f> ./eccenc --chunk-size 2048 --chunk-per-block 128 -bs0 23 -bdo 2000 boot 0 linux.bin
```

```
3145728 squashfs.o 9437184
```

*g>* the image for nand flash e-WriterPro will generate with name “*burn.bin.ecc*”.

## 4.61 IP-IN-IP SUPPORT

### 4.61.1 How To enable IP-IN-IP

**Kernel config**

**make linux\_menuconfig**

```
[*] Networking support --->
```

```
 Networking options --->
```

```
 [*] IP: tunneling
```

**Users config**

**make menuconfig**

```
[*] Config busybox
```

```
 Networking Utilities --->
```

```
 [*] ip
```

```
 [*] ip address
```

```
 [*] ip link
```

```
 [*] ip route
```

```
 [*] ip tunnel
```

```
 [*] ip rule
```

**make users\_menuconfig**

```
[] iproute2 // disable iproute2
```

### 4.61.2 How to configure ip-in-ip

LAN PC<-----> DUT <----- Tunnel -----> Ubuntu server

DUT Confiure:

```
ifconfig eth1 dut_ip_addr
```

```
ip tunnel add tun0 mode ipip remote server_ip_addr local dut_ip_addr
```

```
ip link set tun0 mtu 1400 up
ip address add tun_dut_ip brd 255.255.255.255 peer tun_server_ip dev tun0
route del default
route add default gw tun_server_ip
//set firewall to accept tunnel packet
iptables -I INPUT -p 4 -j ACCEPT
iptables -I INPUT -p icmp -j ACCEPT
iptables -A FORWARD -p icmp -j ACCEPT
iptables -A FORWARD -i tun0 -m state --state RELATED,ESTABLISHED -j ACCEPT
```

ubuntu server configure:

```
ifconfig eth1 server_ip_addr
ip tunnel add tun0 mode ipip remote dut_ip_addr local server_ip_addr
ip link set tun0 mtu 1400 up
ip address add tun_server_ip brd 255.255.255.255 peer tun_dut_ip dev tun0
route del default
route add default gw tun_dut_ip
```

#### 4.61.3 How to test

1. LAN PC ping tun\_server\_ip
2. LAC PC run chariot to test thruput by tunnel encapsulation

### 4.62 Realtek Giga Lite (Two-Pair 500Mbps) support (for kernel 3.10)

For RTL8198C/RTL8198CS/RTL8954E/RTL8954ES SoC, the Realtek proprietary giga lite protocol allows the Ethernet port to link on 500Mbps instead of 100Mbp with two-pair cable. It can only take effect when two link partners support this protocol.

Disable giga lite in the release image: the giga lite default setting is disabled in the release SDK.

Enable giga lite in the release image: please enable “Enable Giga Lite support” item.

Path: make linux\_menuconfig

```
-> Device Drivers
 -> Network device support
 -> Options for Realtek SoC
 -> Enable Giga Lite support
```

Enable/disable giga lite in the run time stage:

Enable: echo gigalite 1 > /proc/rtl865x/phyReg

Disable: echo gigalite 0 > /proc/rtl865x/phyReg

Link speed check: use “cat /proc/rtl865x/port\_status” console command to check the link speed is 10M, 100M, 500M or 1Gbps..

## 4.63Https boa support

### 4.63.1 How To enable https boa

make users\_menuconfig

[\*] openssl

--- Choose openssl version.

openssl (openssl-0.9.8i) --->

```
RLX Linux Configuration
the menu. <Enter> selects submenus --->. Highlighted letters are hot
keys, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help,
:cluded <M> module < > module capable

--- (-)
[] ntpclient
[*] openssl
[] openssl (openssl-0.9.8i) --->
[] wapi_utils
[] oray ddns
[*] pathsel
```

[\*] boa

[\*]    boa SSL Support

```

 RLX Linux Configuration
the menu. <Enter> selects submenus --->. Highlighted letters are hot
keys, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help,
<cluded <M> module <> module capable

--- Applications Settings
 BT Configuration --->
 File_system Configuration --->
 IPv6 Configuration --->
 PPP Configuration --->
 RTK_inband Configuration --->
 USB3G Configuration --->
 [] alsa related
 [] auth
 [] bluez
 [*] boa
 [] boa CGI Support (NEW)
 [*] boa IPv6 Support
 [] boa new UI Support (NEW)
 [*] boa SSL Support
 [] brctl

```

#### 4.63.2 How To Generate a certificate

- 1) download ssl.ca-0.1.tar.gz
- 2) # tar zxvf ssl.ca-0.1.tar.gz
- 3) # cd ssl.ca-0.1
- 4) # ./new-root-ca.sh

No Root CA key round. Generating one

Generating RSA private key, 1024 bit long modulus

.....+++++

....+++++

e is 65537 (0x10001)

Enter pass phrase for ca.key: (enter password)

Verifying - Enter pass phrase for ca.key: (verify password)

.....

Self-sign the root CA...

Enter pass phrase for ca.key: (enter password)

.....

.....

Country Name (2 letter code) [MY]:CN

State or Province Name (full name) [Perak]:JS

Locality Name (eg, city) [Sitiawan]:SZ

Organization Name (eg, company) [My Directory Sdn Bhd]:AAA

Organizational Unit Name (eg, section) [Certification Services Division]:AAA

Common Name (eg, MD Root CA) []:AAA

Email Address []:aa@aa.com

5) # ./new-server-cert.sh server

.....

.....

Country Name (2 letter code) [MY]:CN

State or Province Name (full name) [Perak]:JS

Locality Name (eg, city) [Sitiawan]:SZ

Organization Name (eg, company) [My Directory Sdn Bhd]:AAA

Organizational Unit Name (eg, section) [Secure Web Server]:AAA

Common Name (eg, www.domain.com) []:AAA

Email Address []:aa@aa.com

6) # ./sign-server-cert.sh server

CA signing: server.csr -> server.crt:

Using configuration from ca.config

Enter pass phrase for ./ca.key: (enter password)

Check that the request matches the signature

Signature ok

The Subject's Distinguished Name is as follows

countryName :PRINTABLE:'CN'

stateOrProvinceName :PRINTABLE:'JS'

localityName :PRINTABLE:'SZ'

organizationName :PRINTABLE:'AAA'

organizationalUnitName:PRINTABLE:'AAA'

```

commonName :PRINTABLE:'AAA'
emailAddress :IA5STRING:'aa@aa.com'

Certificate is to be certified until Jul 16 12:55:34 2005 GMT (365 days)

Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]y

Write out database with 1 new entries

Data Base Updated

CA verifying: server.crt <-> CA cert

server.crt: OK

7) Copy server.key&server.crt to ~/users/boa/, and rename to privateKey.key&certificate.crt.

8) build image

```

## 4.64exfat-fuse file system

### 4.64.1 How To enable exfat-fuse

make linux\_menuconfig

[\*] FUSE (Filesystem in Userspace) support

```

[*] Inotify support for userspace
[] Filesystem wide access notification
[] Quota support
[] Kernel automounter version 4 support (also supports v3)
[*] FUSE (Filesystem in Userspace) support
[] Character device in Userspace support
 Caches --->
 CD-ROM/DVD Filesystems --->

```

make users\_menuconfig

[\*] exfat-fuse

```

File_system Configuration
the menu. <Enter> selects submenus --->. Highlighted letters are hot
keys, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help,
xcluded <M> module < > module capable

[] dosfsck
[] e2fsprogs
[*] exfat-fuse
[] mkdosfs
[] NFS utility

```

[\*] fuse library

**Libraries Settings**

the menu. <Enter> selects submenus --->. Highlighted letters are hot  
des, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help,  
<Excluded <M> module < > module capable

---

[ ] flex library  
[ \* ] fuse library  
[ ] Libpcap library  
[ ] libpng  
[ ] zlib

#### **4.64.2 How To mount exfat-fuse**

```
mkdir /tmp/usb/sda1
mount.exfat-fuse /dev/sda1 /tmp/usb/sda1
```

## 4.65exfat-nofuse file system

#### **4.65.1 How To enable exfat-nofuse**

make linux\_menuconfig

## File systems --->

## DOS/FAT/NT Filesystems --->

[\*] exFAT fs support

#### 4.65.2 How To mount exfat-fuse

```
mkdir /tmp/usb/sda1
```

```
mount -t exfat /dev/sda1 /tmp/usb/sda1
```

## 4.66 UART1 support

Config method:

```
make linux_menuconfig
```

```
Device Drivers --->
Character devices --->
Serial drivers --->
```

Note: Number of 8250/16550 serial port to register at runtime should be changed to 2

```
[*] 8250/16550 and compatible serial support
[*] Console on 8250/16550 and compatible serial port
(2) Maximum number of 8250/16550 serial ports
(2) Number of 8250/16550 serial ports to register at runtime
[] Extended 8250/16550 serial driver options
*** 8250 compatible port support ***
[+]
[+] 819x RTL UART1 support
[] SC16IS7x0 series (I2C bus) support
```

Test Method:

- Find an pc ,use tools such as SecureCRT connected with DUT uart1on pc, choose an baudrate such as 38400.
- Enter command on DUT : stty -F /dev/ttyS1 38400 (the baudrate you set)
- Enter command on DUT : echo "any world you enterd" > /dev/ttyS1
- Check SecureCRT if can get the right(any world you enterd) world you enterd on DUT

How to config stty:

Make menuconfig:

```
Config busybox --->
Coreutils --->
[] stty
```

## 4.67 System Memory size > 256MB support (98C only)

- If the memory size of your platform is bigger than 256MB, for example 512MB, then you should follow below steps to enable high memory support in the kernel menuconfig.

Step1: Enable High Memory Support in kernel\_menuconfig

*System Configuration* --->

[\*] Support System Memory bigger than > 256MB

*Kernel type* →

- [\*] High Memory Support
- [ ] Enable bounce buffers

## 4.68 Hardware Crypto Engine API support

### 4.68.1 What's Hardware Crypto Engine API?

We have two ways to implement the API:

1. The Standard Register Procedure: The Advantage is user can specify which algorithm (SW or HW) he want.
2. Hack the Crypto API: The Advantage is all test/app code can be reused.

As the time issue, we select the Item2 to implement. Features support currently:

1. "md5"
2. "sha1"
3. "ecb(des)"
4. "ecb(des3\_ede)"
5. "ecb(aes)"
6. "cbc(des)"
7. "cbc(des3\_ede)"
8. "cbc(aes)"
9. "ctr(aes)"

Note: The Linux Crypto API is layer design. So "hmac(md5) / hmac(sha1) / rfc3686(ctr(aes)) / cts(cbc(aes)) ..." will be affected. Only 8881A、8198 and 8198C chip support this feature.

### 4.68.2 How to enable Hardware Crypto Engine API

Hardware Crypto Engine API is based on Linux Cryptographic API. So if we want to use Hardware Crypto Engine API, Linux Cryptographic API should be enabled at first.

Enable Hardware Crypto Engine API as follows:

```
make linux_menuconfig // To configure linux kernel settings
```

Menuconfig:

Location:

- > Cryptographic API
- > Hardware crypto devices
- [\*] Driver Realtek Crypto Engine

#### 4.68.3 How to test

##### ***Step 1***

Enable test related macro in linux\_menuconfig.

```
make linux_menuconfig // To configure linux kernel settings
```

Menuconfig:

Location:

- > Cryptographic API
- > Hardware crypto devices
- [\*] Driver Realtek Crypto Engine
- [\*] Driver Realtek Crypto Engine Test //for test
- [\*] Realtek Crypto Engine Test by Linux Selftest case //for test

##### ***Step 2***

Disable watchdog by run the following command, avoid testing for long time cause the system reboot:

```
echo "write 0xb800311C 0xA5600000" >/proc/rtl865x/memory
```

##### ***Step 3***

Init the test code by run the following command:

```
echo 10 >/proc/rtk_ipsec_test
```

##### ***Step 4***

Test the code by run the following command:

```
echo "field" >/proc/rtk_ipsec_test
```

field: current support: 10/99/11~16/19/21/31/40~46

11~14 as throughput related test

15 as DES related test  
16 and 31 as AES related test  
19 and 21 as random related test  
40~46 test by linux selftest case

Example:

echo 11 >/proc/rtk\_ipsec\_test means DES and 3DES throughput test.

#### 4.68.4 The matters need attention

```
1. int32 rtl_ipsecEngine(uint32 modeCrypto, uint32 modeAuth,
 uint32 cntScatter, rtl_ipsecScatter_t *scatter, void *pCryptResult,
 uint32 lenCryptoKey, void* pCryptoKey,
 uint32 lenAuthKey, void* pAuthKey,
 void* pIv, void* pPad, void* pDigest,
 uint32 a2eo, uint32 enl)
```

The address of the parameter of **rtl\_ipsecEngine** marked by red color must be un-cached address, please refer the test sample code.

2. Please disable watchdog when run test case, otherwise may trigger watchdog cause the system restart.

#### 4.69 Apple WAC support

Apple's Wireless Accessory Configuration feature is designed to allow consumers to simply set up their wireless accessories with the network credentials already stored on their iOS or OSX device.

Pre-existing MFi hardware is necessary.

##### 4.69.1 How To enable WAC

```
make linux_menuconfig
[*] Networking support --->
 Networking options --->
 [*] The IPv6 protocol --->
```

```
[*] Networking options
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>
for Search. Legend: [*] built-in [] excluded <M> module < > module
[*] IP: ESP transformation
[*] IP: IPComp transformation
[*] IP: IPsec transport mode
[*] IP: IPsec tunnel mode
[*] IP: IPsec BEET mode
[*] Large Receive Offload (ipv4/tcp)
[*] INET: socket monitoring interface
[*] TCP: advanced congestion control --->
[*] TCP: MD5 Signature Option support (RFC2385) (EXPERIMENTAL)
[*] The IPv6 protocol --->
[] Security Marking
[*] Network packet filtering framework (Netfilter) --->
```

Device Drivers --->

## Character devices --->

Serial drivers --->

[\*] Support I2C-GPIO Driver for MFi Coprocessor

```
Serial drivers
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>
for Search. Legend: [*] built-in [] excluded <M> module < > module
[*] 8250/16550 and compatible serial support
[*] Console on 8250/16550 and compatible serial port
(2) Maximum number of 8250/16550 serial ports
(1) Number of 8250/16550 serial ports to register at runtime
[] Extended 8250/16550 serial driver options
*** 8250 compatible port support ***
[!] Support I2C-GPIO Driver for MF1 Coprocessor
```

*make users\_menuconfig*

## System Configuration --->

## [\*] Apple WAC Support

RLX Linux Configuration  
Arrow keys navigate the menu. <Enter> selects submenus ---.  
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,  
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>  
for Search. Legend: [\*] built-in [ ] excluded <M> module < > module

( - )

- [ ] dnsmasq
- [ ] dosfsck
- [ ] exfat-fuse
- [ ] gdbserver
- [\*] boa
- [ ]   boa IPv6 Support
- [ ] uWiFi
- [ ] alsal related
- [ ] Apple WAC Support
- [ ]   Apple Homekit Support
- [ ] gproxy
- [ ] Multi PPPoE Support

*make menuconfig*

## [\*] Config busybox

## Networking Utilities --->

[\*] Enable IPv6 support

[\*] ping6

```
Networking Utilities
Arrow keys navigate the menu. <Enter> selects submenus ---.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>
for Search. Legend: [*] built-in [] excluded <M> module < > module
[*] Enable IPv6 support
[*] Preferentially use IPv4 addresses from DNS queries (NEW)
[] Verbose resolution errors
[] arp
[] arping
```

```
Networking Utilities
Arrow keys navigate the menu. <Enter> selects submenus ---.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>
for Search. Legend: [*] built-in [] excluded <M> module < > module
[*] Support displaying rarely used link types
[] ipcalc
[] nameif
[] nc
[] netstat
[] nslookup
[*] ping
[**] ping6
[*] Enable fancy ping output
[] pscan
```

#### **4.69.2 How to test WAC**

1. Wlan Configuration: Mode~AP Encrypt~disabled
  2. First your iphone or MacBook connect to a remote AP.
  3. Site survey the AP with your iphone or MacBook.
  4. Click next.

If WAC finished, the AP will change to client or repeater mode and connect to the remote AP, and your iphone or Macbook will display successful message.

#### **4.70 Apple HomeKit support**

Apple's HomeKit allows consumers to communicate with and control connected accessories in the home. You can enable users to discover HomeKit accessories in their home and configure them, or you can create actions to control those devices.

Pre-existing MFi hardware is necessary.

#### 4.70.1 How To enable HomeKit

You should enable WAC before Homekit.

*make linux\_menuconfig*

System Configuration --->

[\*] Apple MFi program support

[\*] Apple Homekit mtd block support

```
System Configuration
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>
for Search. Legend: [*] built-in [] excluded <M> module < > module
System Type (96D+92D GW) --->
[*] Enable watchdog timer support
[] Support RTL8367R for RTL881A
[] Support RTL8211F for RTL881A
[] Enable timer adjustment support
[] Webpages in rootfs support
[*] SPI flash support
[*] 819xD clock source at 40Mhz
[] Enable Flash Dual Bank support
[] USB3G support
[*] Seedup usb samba performance
[*] Apple MFi program support
[*] Apple Homekit mtd block support
[] Http File server support
*** Support two spi flash ***
```

*make users\_menuconfig*

System Configuration --->

[\*] Apple WAC Support

[\*] Apple Homekit Support

```
RLX Linux Configuration
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>
for Search. Legend: [*] built-in [] excluded <M> module < > module
(-)
[] dnsmasq
[] dosfsck
[] exfat-fuse
[] gdbserver
[*] boa
[] boa IPv6 Support
[] uWiFi
[] alsa related
[*] Apple WAC Support
[*] Apple Homekit Support
Choose Homekit Services (Homekit Switch Service) --->
[] gproxy
[] Multi PPPoE Support
```

#### 4.70.2 How to Test HomeKit

In order to user homekit, the AP and your MacBook should connect to the same remote AP.

With Apple's software HAT, you can discover and configure the AP.

## 4.71 rtl\_fs support

rtl\_fs is Realtek proprietary high performance file-system supporting for NTFS, exFAT and HFS+ volume. It will be built as a module and automatically inserted into your kernel when your system was booted up.

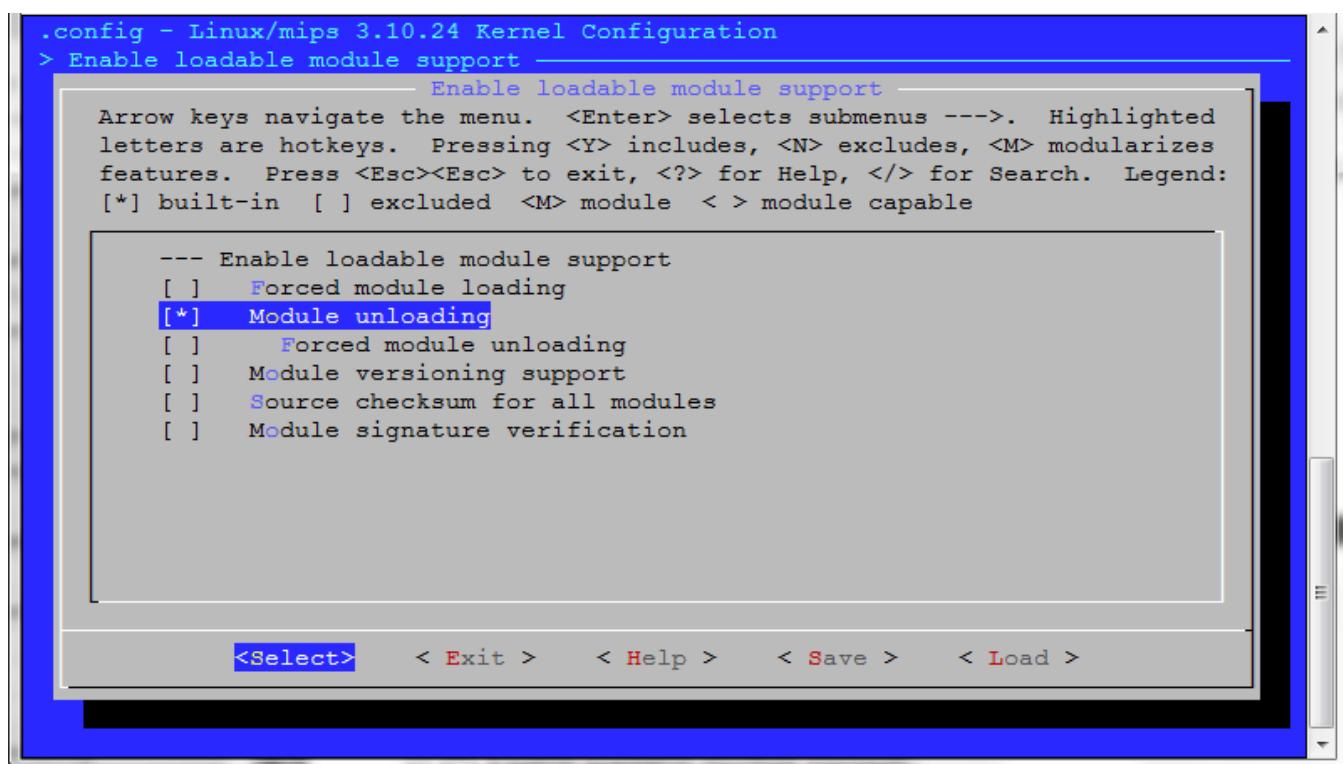
How to enable rtl\_fs? There are three parts of options you need to configure.

1. For kernel:

*make linux\_menuconfig*

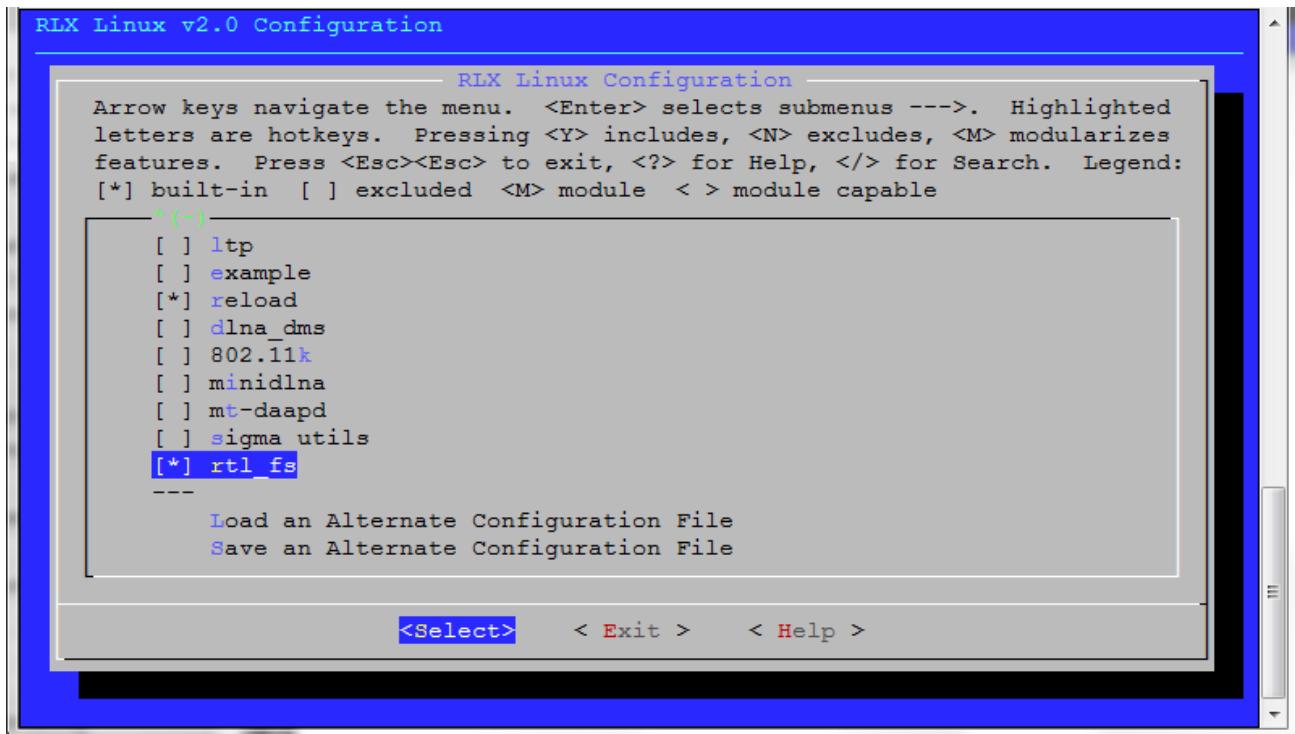
-> [\*] Enable loadable module support

-> [\*] Module unloading



2. For users:

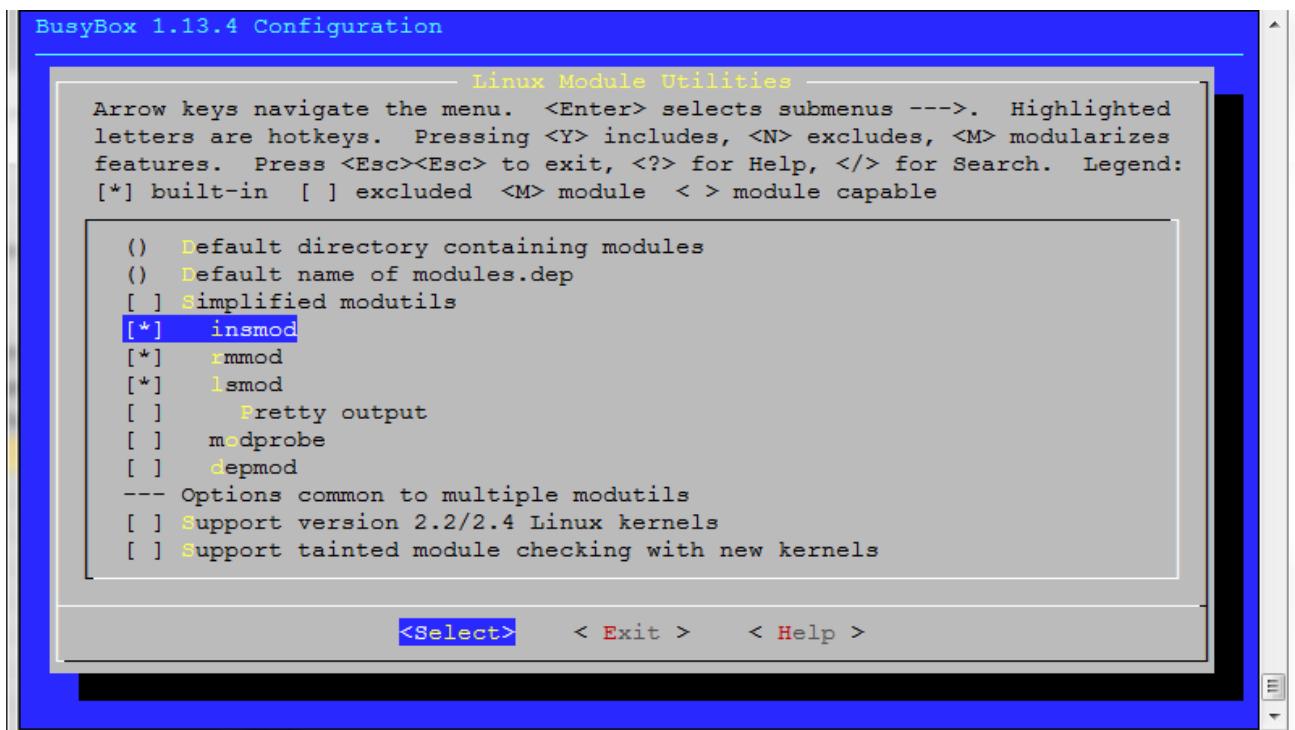
*make users\_menuconfig*



### 3. For busybox command:

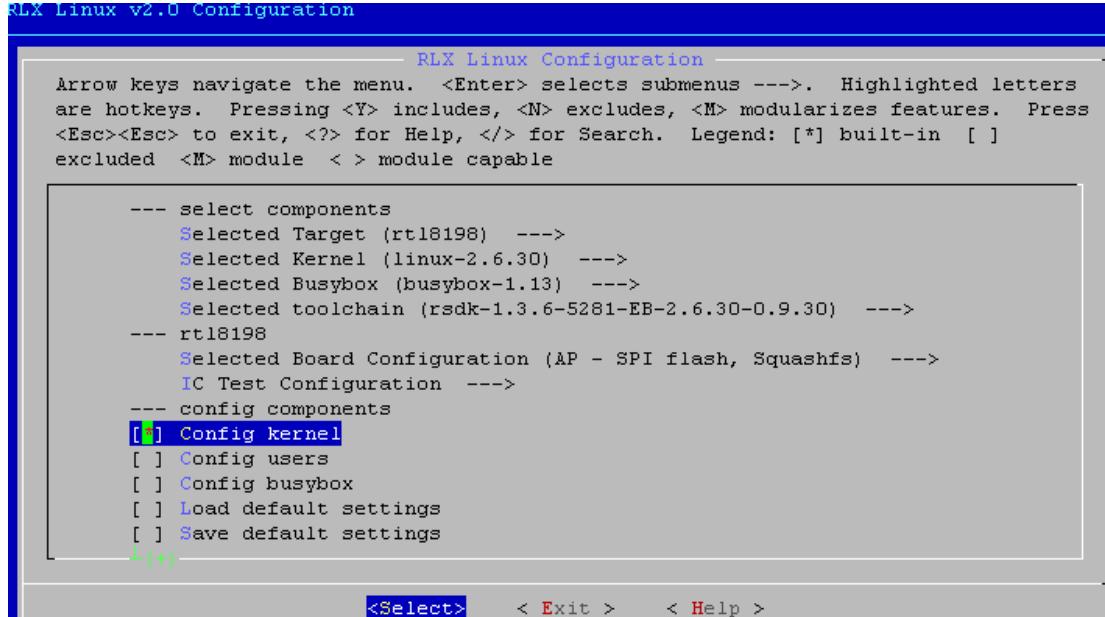
Linux Module Utilities --->

- [\*] insmod
- [\*] rmmod
- [\*] lsmod



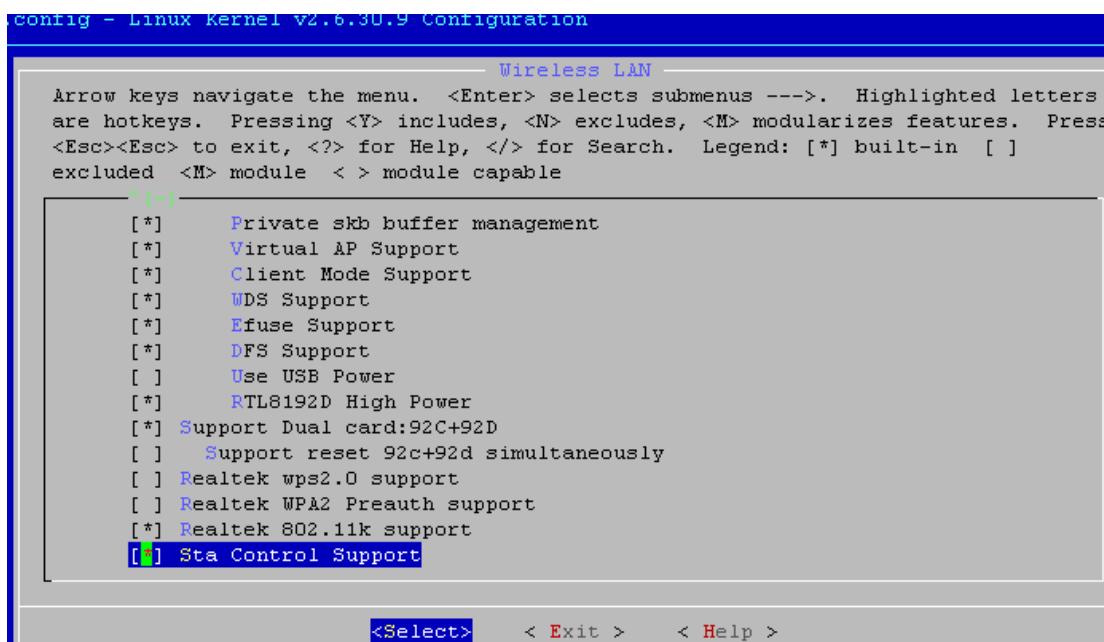
## 4.72 STA Control support

STA Control is Realtek proprietary feature, that works on dual-band platform. It detects whether a wireless client is dual-band capable, and if it is, steer the client to the pre-configured prefer-band. To enable this feature, please select the “Sta Control Support” in menuconfig, as following demonstrated. Select “Config kernel”



, and then exist.

Select option of “Device Drivers”->“Network device support”->“Wireless LAN”, you would see bellow figure.



Make sure “Sta Control Support” is selected. Compile SDK to generate new image.

After the new image is uploaded, an “Enable STA Control” check box would appear in the WLAN1 advanced setting page.

| WLAN Access Point     | SETUP                                                                                                                                                          | WLAN1 | WLAN2 | TCP/IP |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|-------|--------|
| <b>BASIC SETTING</b>  | Fragment Threshold: 2346 (256-2346)                                                                                                                            |       |       |        |
| <b>ADVANCED</b>       | RTS Threshold: 2347 (0-2347)                                                                                                                                   |       |       |        |
| <b>SECURITY</b>       | Beacon Interval: 100 (20-1024 ms)                                                                                                                              |       |       |        |
| <b>ACCESS CONTROL</b> | IAPP: <input checked="" type="radio"/> Enabled <input type="radio"/> Disabled                                                                                  |       |       |        |
| <b>WDS SETTING</b>    | Protection: <input type="radio"/> Enabled <input checked="" type="radio"/> Disabled                                                                            |       |       |        |
| <b>SITE SURVEY</b>    | Aggregation: <input checked="" type="radio"/> Enabled <input type="radio"/> Disabled                                                                           |       |       |        |
| <b>WPS</b>            | Short GI: <input checked="" type="radio"/> Enabled <input type="radio"/> Disabled                                                                              |       |       |        |
| <b>SCHEDULE</b>       | WLAN Partition: <input type="radio"/> Enabled <input checked="" type="radio"/> Disabled                                                                        |       |       |        |
| <b>802.1R SETTING</b> | STBC: <input checked="" type="radio"/> Enabled <input type="radio"/> Disabled                                                                                  |       |       |        |
|                       | LDPC: <input checked="" type="radio"/> Enabled <input type="radio"/> Disabled                                                                                  |       |       |        |
|                       | TX Beamforming: <input checked="" type="radio"/> Enabled <input type="radio"/> Disabled                                                                        |       |       |        |
|                       | Multicast to Unicast: <input checked="" type="radio"/> Enabled <input type="radio"/> Disabled                                                                  |       |       |        |
|                       | TDLS Prohibited: <input type="radio"/> Enabled <input checked="" type="radio"/> Disabled                                                                       |       |       |        |
|                       | TDLS Channel Switch Prohibited: <input type="radio"/> Enabled <input checked="" type="radio"/> Disabled                                                        |       |       |        |
|                       | RF Output Power: <input checked="" type="radio"/> 100% <input type="radio"/> 70% <input type="radio"/> 50% <input type="radio"/> 35% <input type="radio"/> 15% |       |       |        |
|                       | <input checked="" type="checkbox"/> Enable STA Control <input type="button" value="Prefer 5GHz"/>                                                              |       |       |        |
|                       | <input type="button" value="Apply Changes"/> <input type="button" value="Reset"/>                                                                              |       |       |        |

After checking the “Enable STA Control” option and selecting the prefer-band, please apply changes and reboot the device.

In order to have the STA Control function work properly, please make sure the SSID and security setting for both WLAN1 and WLAN2 are same.

## 4.73 802.11r/k support

802.11r introduces a new concept of roaming where the initial handshake with the new AP is done before the client roams to the target AP. The initial handshake allows the clients and APs to do the PTK calculation in advance. This mechanism allows time-consuming operations to be removed from the time critical re-association process.

802.11k is an assisted feature that can as well reduce the time required to roam. By providing neighbor AP list and their channel to clients, it allows clients to quickly determine which AP it should roam next. Clients do not need to probe all channels to find candidate AP for roaming.

To enable this feature, select “Config kernel”->“Device Drivers”->“Network device support”->“Wireless LAN”

```
.config - Linux/mips 3.10.24 Kernel Configuration
> Device Drivers > Network device support > Wireless LAN ━━━━━━━━
 ━━━━━━━━ Wireless LAN ━━━━━━━━
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys.
Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [] excluded <M> module < > module capable

^ (-)
[] HOST Clock Source, Select is 40MHz, otherwise 25MHz
[*] Device Clock Source, Select is 40MHz, otherwise 25MHz
[*] Band Edge Limit support
[] Use kernel thread to process TX power tracking
[*] RTL Mesh Support
[] Support Cross-Band Access
[] RTL A4_STA Support
[] Enable WLAN DoS Filter
[] Realtek HS2.0 support
[] Monitor mode support
[*] TDLS Support
[] Sta Control Support
[] Realtek Simple Config Support
[*] IEEE 802.11K Support
[*] IEEE 802.11W Support
[!] IEEE 802.11R Support
Select WiFi Band on Wlan0 (Select 5g band on wlan0) --->

<Select> < Exit > < Help > < Save > < Load >
```

Make sure both “IEEE 802.11R Support” and “IEEE 802.11K Support” are selected.

Besides, select option of “Users”->“openssl”->“openssl-0.9.8b”

```
RLX Linux Configuration
Arrow keys navigate the menu. <Enter> selects submenus --->. Hotkeys are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [] excluded <M> module < > module capable
^ (-)
[*] ntpclient
[] NFS utility
[!] openssl
--- Choose openssl version.
 openssl (openssl-0.9.8b) --->
[] wapi_utils (NEW)
[] oray ddns
[*] pathsel
```

and select option of "Users"->"FT Daemon support"

```
RLX Linux Configuration
Arrow keys navigate the menu. <Enter> selects submenus --->. Highl
letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modul
features. Press <Esc><Esc> to exit, <?> for Help, </> for Search.
built-in [] excluded <M> module < > module capable
[] nmbs
[+] FT Daemon support
[] u linker autorun image tool
[] python support
--- Choose python version:
[] simple config support
--- USB3G support
[] comgt
[] chat
```

Compile SDK to generate new image.

After the new image is uploaded, click list view of “802.11R SETTING”, Fast BSS Transition configuration menu is as below,

The screenshot shows the Realtek Linux Configuration interface. The left sidebar has a vertical list of menu items: Realtek, SETUP, WLAN1, WLAN2 (highlighted in yellow), TCP/IP, FIREWALL, and MANAGEMENT. Below this, there are several sub-menu items: BASIC SETTING, ADVANCED, SECURITY, ACCESS CONTROL, WDS SETTING, SITE SURVEY, WPS, SCHEDULE, and 802.11R SETTING (highlighted in yellow). The main content area is titled "Fast BSS Transition (802.11r)". It contains the following configuration fields:

- SSID Type: Root AP - RTK 11n AP 2.4G
- Encryption: None
- IEEE 802.11r: Disable (radio button selected)
- Mobility Domain ID: 0
- Support over DS: Disable (radio button selected)
- Key expiration timeout: 0 (1..65535 minutes, 0:disable)
- Reassociation timeout: 0 (1000..65535 seconds, 0:disable)
- NAS identifier: 0 (1-48 characters)
- Support Key-Push: Disable (radio button selected)

At the bottom of this section is a blue "Apply Changes" button. Below this is another section titled "Key Holder Configuration" with a "SSID Type" dropdown set to "Root AP - RTK 11n AP 2.4G".

In upper half of this menu, it is used to configure wlan driver settings.

**Mobility Domain ID:** The MDID is needed to set with the same ID, which means they are within the same mobility domain.

**Key expiration timeout:** Key expiration timeout in minutes

**Reassociation timeout:** Reassociation timeout in TUs

**NAS identifier:** R0 Key Holder identifier. Each APs has its own ID.

In lower half of this menu, it is used to configure for other APs in the same Mobility Domain.

| MAC address       | NAS identifier   | 128-bit key / passphrase         | Op Class | Channel | Select                   |
|-------------------|------------------|----------------------------------|----------|---------|--------------------------|
| RTK 11n AP 2.4G   |                  |                                  |          |         |                          |
| 00:e0:4c:12:22:6e | ap1.r0kh-id.fqdn | 000102030405060708090a0b0c0d0e0f | 81       | 6       | <input type="checkbox"/> |
| 00:e0:4c:72:55:66 | ap3.r0kh-id.fqdn | 000102030405060708090a0b0c0d0e0f | 81       | 11      | <input type="checkbox"/> |

**MAC address:** The other AP's mac address within the mobility domain

**NAS identifier:** The other AP's R0 Key Holder identifier within the mobility domain

Following two fields are used for 802.11k neighbor list.

**Op Class:** The other AP's operation class. Please refer to the Table 2 of the development guide "Realtek\_802.11k.doc"

**Channel:** The other AP's channel.

## 5 FLASH tools

FLASH tools such as flash, cvcfg-gw, compweb, cvimg, mgbin are stored at  
rtl819x/users/goahead-2.1.1/LINUX or rtl819x/users/boa/tools if Boa web server is selected.

### 5.1 flash

Read/write MIB in Flash.

Usage:

| Command                                                       | Description                                   |
|---------------------------------------------------------------|-----------------------------------------------|
| flash all                                                     | Dump all MIB name and value in Flash          |
| flash get [wlan-idx] <MIB name>                               | Get MIB value from Flash                      |
| flash set [wlan-idx] <MIB name> <MIB value>                   | Set MIB value into Flash                      |
| flash set [wlan-idx] <MIB name><br><action> <MIB table entry> | Set MIB table entry into Flash                |
| flash reset                                                   | Reset current settings (use default settings) |
| flash default-sw                                              | Reset Current and Default settings            |
| flash default                                                 | Reset HW, Current, and Default settings       |
| flash set_mib                                                 | Get MIB from flash and set to WLAN interface  |

Examples:

```
flash all | grep HW // find all MIB in HW settings
flash get WLAN0_SSID // get SSID in wlan0
flash set WLAN0_SSID test // set SSID to test in wlan0
flash set WDS add 00e04c8196cc comment 0 // add 00e04c8196cc to WDS, 0 is autorate
```

### 5.2 cvcfg-gw

MIB binary and text conversion tool.

Usage:

```
cvcfg-gw [-r] [-no_hw] <input> <output>
```

Options:

- r: generate raw configuration with padding 0
- no\_hw: do not generate hw setting
- in: input file path //convert binary file to txt must use this option
- ot: output file path //convert binary file to txt must use this option

Examples:

```
cvcfg-gw config-gw-96c.txt config-gw-96c.dat //convert txt to binary file
cvcfg-gw -r -no_hw config-gw-96c.txt config.bin //convert txt to binary file generate no hw
cvcfg-gw -in config-gw-96c.dat -ot config-gw-96c.txt //convert binary file to txt
```

### **5.3 compweb**

Convert WEB pages to be the downloadable binary.

Usage:

```
compweb <mode> <fileList> <output name>
```

Options:

- mode: gw

Examples:

```
find .../web-gw -name "*.*" > web_files
compweb gw web_files webpages-gw.bin
```

### **5.4 cvimg**

Convert image to be the downloadable binary.

Usage:

```
cvimg <image type> <image> <output file> <load address> <flash offset>
```

Options:

- Image type: root / boot / linux
- Load address: used in linux type
- Flash offset: burn address on flash

Examples:

```
cvimg root squashfs-lzma.o root.bin 0x100000 0x100000
cvimg boot boot boot.bin 0 0
cvimg linux nfjfrom linux.bin 80500000 30000
```

### **5.5 mgbin**

Merge all binary into one image.

Usage:

```
mgbin [-s] [-c] [-a] -o outputfile bootcode config webpages linux root
```

Options:

- s: do byte swap
- c: cascade. May use this option to merge image for web upload
- a: add all tag in header

Examples:

- 1) Create full image (rtl8196c-gw.bin) for TFTP update.  

```
cvcfg-gw -r config-gw-96c.txt config-gw-96c.bin
mgbin -o rtl8196c-gw.bin boot.bin config-gw-96c.bin webpages-gw.bin linux.bin root.bin
Note: add -s if need swap
```
- 2) Create total image (rtl8196c-gw.bin) without boot loader and HW settings for TFTP update.  

```
cvcfg-gw -r -no_hw config-gw-96c.txt config-gw-96c.bin
mgbin -o rtl8196c-gw.bin config-gw-96c.bin webpages-gw.bin linux.bin root.bin
```

3) Create total image (rtl8196c-gw.bin) for WEB upgrade.

```
mgbn -o rtl8196c-gw.bin webpages-gw.bin linux.bin root.bin
```

4) Create total image (rtl8196c-gw.bin) with config dates for WEB upgrade.

```
cvcfg-gw config-gw-96c.txt config-gw-96c.dat
```

```
mgbn -c -o rtl8196c-gw.bin config-gw-96c.dat webpages-gw.bin linux.bin root.bin
```

## 6 Proc file format

### 6.1 br\_igmpProxy

Path: /proc/br\_igmpProxy

Description: Enable/Disable IGMP proxy function.

Input Format:

Echo “\$FLAG” > /proc/br\_igmpProxy

Input Para:

\* 1: Enable IGMP proxy.

\* 0: Disable IGMP proxy.

Output Format:

\$FLAG

Output Para.:

\* 1: IGMP proxy daemon is running.

\* 0: IGMP proxy daemon is not running.

### 6.2 br\_igmpsnoop

Path: /proc/br\_igmpsnoop

1) Description: Enable/Disable IGMP snooping.

Input Format:

Echo “\$FLAG” > /proc/ br\_igmpsnoop

Input Para:

\* 1: Enable IGMP snooping.

\* 0: Disable IGMP snooping.

Output Format:

\$FLAG

Output Para:

\* 1: IGMP snooping enabled.

\* 0: IGMP snooping disabled.

2) Description: Configure reserved multicast address not processed by multicast fastfwd.

Input Format:

Echo “reserve \$McastAddr” > /proc/ br\_igmpsnoop

Input Para:

\*McastAddr: The reserved multicasts address

Output Format:

None

Output Para:

None

3) Description: Enable/Disable fast leave

Input Format:

Echo fastleave “\$FLAG1” “\$FLAG2” > /proc/ br\_igmpsnoop

Input Para:

FLAG1: 1: Enable IGMP fast leave; 0: Disable IGMP snooping.

FLAG2: last member aging time when disable fast leave

Output Format:

None

Output Para:

None

4) Description: configure src mac address of multicast packet forward by HW

Input Format:

echo mcastMac “\$Enabled” “\$oriMac” “\$replaceMac” > proc/br\_igmpsnoop

Input Para:

Enabled: 1:enabled configure src mac process;0: disable configure src mac process.

oriMac: original src mac address before configure

replaceMac: the configured src mac address for hw multicast

Output Format:

mcastMac: [\$Enabled] oriMac: \$oriMac replaceMac: \$replaceMac

Output Para:

Enabled: 1:enabled configure src mac process;0: disable configure src mac process.

oriMac: original src mac address before configure

replaceMac: the configured src mac address for hw multicast

### 6.3 custom\_Passthru

1) Path: /proc/ custom\_Passthru

Description: Enable/Disable IPv6/PPPoE pass through for gateway mode.

Input Format:

Echo “\$FLAG” > /proc/ custom\_Passthru

Input Para.:

\* bit 0 for IPv6 pass through control and bit 1 for PPPoE pass through control.

\* 0: Disable both IPv6 and PPPoE pass through.

\* 1: Enable IPv6 pass through and disable PPPoE pass through.

\* 2: Enable PPPoE pass through and disable IPv6 pass through.

\* 3: Enable both IPv6 and PPPoE pass through.

Output Format:

\$FLAG

Output Para:

- \* 0: Both IPv6 and PPPoE pass through disabled.
- \* 1: IPv6 pass through enabled and PPPoE pass through disabled.
- \* 2: PPPoE pass through enabled and PPPoE pass through disabled.
- \* 3: Both IPv6 and PPPoE pass through enabled.

2) Path: /proc/custom\_Passthru\_wlan

Description: Enable/Disable IPv6/PPPoE pass through for WISP mode.

Usage of /proc/custom\_Passthru\_wlan is the same as /proc/ custom\_Passthru above.

Please refer to the 7.14 and 7.30 for the entire configure sequence of pass through.

## 6.4 enable\_dos

Path: /proc/enable\_dos

Description:

Input Format:

Echo “\$FLAG” > /proc/enable\_dos

Input Para.:

- \* 1st parameter: operation mode.
- \* 2nd parameter: br0 IP address.
- \* 3rd parameter: br0 subnet mask.
- \* 4th parameter: item number.
- \* 5th parameter: whole SYN threshold.
- \* 6th parameter: whole FIN threshold.
- \* 7th parameter: whole UDP threshold.
- \* 8th parameter: whole ICMP threshold.
- \* 9th parameter: per SYN threshold.
- \* 10th parameter: per FIN threshold.
- \* 11th parameter: per UDP threshold.
- \* 12th parameter: per ICMP threshold.
- \* 13th parameter: block time.

Output Format:

\$FLAG

Output Para.:

- \* NULL: disable dos.
- \* 1st parameter: operation mode.
- \* 2nd parameter: br0 IP address.
- \* 3rd parameter: br0 subnet mask.
- \* 4th parameter: item number.

- \* 5th parameter: whole SYN threshold.
- \* 6th parameter: whole FIN threshold.
- \* 7th parameter: whole UDP threshold.
- \* 8th parameter: whole ICMP threshold.
- \* 9th parameter: per SYN threshold.
- \* 10th parameter: per FIN threshold.
- \* 11th parameter: per UDP threshold.
- \* 12th parameter: per ICMP threshold.
- \* 13th parameter: block time.

## 6.5 fast\_l2tp

Path: /proc/fast\_l2tp

Description:

Input Format:

```
echo "$FLAG" > /proc/fast_l2tp
```

Input Para:

- \* 1: the file value is 1 when the WAN interface type is set to L2TP
- \* 0: the file value is 0 when the WAN interface type is not set to L2TP

Output Format:

\$FLAG

Output Para:

- \* 1: the file value is 1 when the WAN interface type is set to L2TP
- \* 0: the file value is 0 when the WAN interface type is not set to L2TP

## 6.6 fast\_nat

Path: /proc/fast\_nat

Description:

Cases 1), if CONFIG\_RTL\_NF\_CONNTRACK\_GARBAGE\_NEW is not enabled at kernel configure, /proc/fast\_nat is only used for fastpath control.

Cases 2), if CONFIG\_RTL\_NF\_CONNTRACK\_GARBAGE\_NEW is enabled at kernel configure, /proc/fast\_nat is used for fastpath control and fastpath garbage collection rx threshold control.

Input Format:

```
echo "$FLAG" > /proc/fast_nat
```

Input Para.:

- \* 0: fastpath disabled
- \* 2: flush conntrack at kernel
- \* non-zero and not equal to 2: fastpath enabled
- \* larger than 2: assign it to the fastpath garbage collection rx threshold. Note: this is only for

case 2) above.

Output Format:

\$FLAG

Output Para.:

For case 1), output “fastpath: [%d]”

- \* 10: disable fastpath.
- \* 11: enable fastpath.
- \* 12: clean the conntrack table.

For case 2), output “fastpath %s, GC\_RX\_Count %d, Status: %d”

- \* fastpath enabled or disabled
- \* GC\_RX\_Count is the fastpath garbage collection rx threshold
- \* Status is the status for new garbage collection

## 6.7 fast\_pptp

Path: /proc/fast\_pptp

Description:

Input Format:

echo “\$FLAG” > /proc/fast\_nat

Input Para.:

- \* 1: the file value is 1 when the WAN interface type is set to PPTP
- \* 0: the file value is 0 when the WAN interface type is not set to PPTP

Output Format:

\$FLAG

Output Para.:

- \* 1: the file value is 1 when the WAN interface type is set to PPTP
- \* 0: the file value is 0 when the WAN interface type is not set to PPTP

## 6.8 filter\_table

Path: /proc/filter\_table

Description: url filter control

Input Format:

echo “\$STR” > /proc/filter\_table

Input Para.:

\*1: “flush”

Clear filter table, no parameter.

\*2: “init \$list\_id

Initial the list with the id (\$list\_id)

\$ list\_id: several filter lists are created in filter table, list\_id is the id of list

#define IP\_RANGE\_TABLE 1

```

#define MAC_TABLE 2
#define URL_KEY_TABLE 3
#define SCHEDULT_TABLE 4
#define CONTENT_FILTER 5

*3: "add:$resv#$list_id $flag $num key1 key2 ..."

Add url rule
$resv: reserved, set 0 recommended.
$list_id: several filter lists are created in filter table, list_id is the id of list
$flag: flag of filter rules
 bit0: is the first condition? 1:0 //If this list is the first condition, bit0 is set 1,
because only url list exist
 bit1: have next condition? 1:0 [next table condition] //If there is continuous
condition, bit1 is set 1
 bit2: have "and" logic condition?1:0
 bit3: default action: 1 block;0 forward //action of match: drop or forward
 bit4~7: the index of "and" logic rule
 bit8: all match flag 1: all, 0: not all
 bit9: NULL flag, 1:NULL, 0: not NULL
$num: number of keyword

*4: "enable_log"
Enable log of filter related. It will be shown at log webpage of AP.

*5: "white"
Set URL filter mode to white list.

*6: "black"
Set URL filter mode to black list.

```

Output Format:

```

type=$list_id num=$item_num+1
index=$list_index url=$key1
...
index=$list_index url=$item_num

```

Output Para.:

## 6.9 gpio

Path: /proc/gpio

Description: The /proc/gpio was used for led control and push button. It's depends on the hardware.

Input Format:

Input Para.

Output Format:

Output Para.:

## **6.10 pptp\_conn\_ck**

Path: /proc/ pptp\_conn\_ck

Description: for pptp dial-on-demand wan type, if wan is not dialed up, wan ip(10.64.64.\*) is invalid, the packets from lan to wan will not go thru fast pptp, but go up to linux kernel to trigger pptp dialing up.

Input Format:

```
echo $FLAG > /proc/pptp_conn_ck
```

Input Para:

- \* 3: wan type is pptp dial-on-demand and wan ip is 10.64.64.\* , the packets from lan to wan will go up to linux kernel to trigger pptp dialing up

- \* others: disable this function

Output Format:

```
$FLAG
```

Output Para:

- \* 3: wan type is pptp dial-on-demand and wan ip is 10.64.64.\* , the packets from lan to wan will go up to linux kernel to trigger pptp dialing up

- \* others: disable this function

## **6.11 qos**

Path: /proc/ qos

Description: Enable/Disable fastpath qos control.

Input Format:

```
Echo “$FLAG” > /proc/ qos
```

Input Para:

- \* 0: QoS disabled

- \* non-zero value: QoS enabled

Output Format:

```
$FLAG
```

Output Para:

- \* 0: QoS disabled

- \* non-zero value: QoS enabled

## **6.12 rf\_switch**

Path: /proc/ rf\_switch

Description: check the current state of wireless.

Input Format:

none

Input Para.:

none

Output Format:

\$FLAG

Output Para:

\* 0: the current state of wireless is enabled.

\* 1: the current state of wireless is disabled.

## 6.13 rtk\_vlan\_support

Path: /proc/ rtk\_vlan\_support

Description: enable/disable rtk vlan feature

Input Format:

echo \$FLAG >/proc/rtk\_vlan\_support

Input Para:

0: disable rtk vlan feature

1: enable rtk vlan feature

Output Format:

\$FLAG

Output Para:

0: the current state of rtk vlan feature is disabled

1: the current state of rtk vlan feature is enabled

## 6.14 mib\_vlan

Path: /proc/ethx/mib\_vlan

Description: set the vlan feature of each interface. Each port is mapped to one interface ethx; from left to right is eth0, eth2, eth3, eth4, eth1.

Input Format:

echo "1 \$is\_lan \$vlan \$tag \$vid \$priority \$cfi" > /proc/eth0/ mib\_vlan

Input Para:

is\_lan = 1; //1:lan, 0: wan

vlan = 1; //1:enable rtk vlan in this port, 0: disable rtk vlan feature in this port

tag = 0; // 1: only received vlan tagged packet and transmit packet with vlan tagged, 0: received both vlan tagged and vlan untagged packet and transmit packet with UNTAGGED

vid = vlan id; //vlan id: 1~4095.

priority = x; //priority value in 802.1Q (0~7) if transmit packet with vlan tagged.

cfi=0; //cfi value in 802.1Q tag if transmit packet with vlan tagged.

Output Format:

\$Global\_vlan \$is\_lan \$vlan \$tag \$vid \$priority \$cfi

Output Para.:

Global\_vlan; // 1: rtk vlan feature is enable in whole system; 0: rtk vlan feature is disable in whole system

is\_lan; //1:lan, 0: wan

vlan; //1:enable rtk vlan in this port, 0: disable rtk vlan feature in this port

tag; // 1: only received vlan tagged packet and transmit packet with vlan tagged, 0: received both vlan tagged and vlan untagged packet and transmit packet with UNTAGGED

vid; //vlan id: 1~4095.

priority; //priority value in 802.1Q (0~7) if transmit packet with vlan tagged.

cfi; //cfi value in 802.1Q tag if transmit packet with vlan tagged.

## 6.15 sw\_nat

Note: proc sw\_nat exist when CONFIG\_RTL\_HARDWARE\_NAT is not defined.

Path: /proc/sw\_nat

Description: when hardware NAT is disabled, change the op mode of AP.

Input Format:

echo "\$FLAG" /proc/sw\_nat

Input Para:

\* "echo 0 > /proc/sw\_nat" : change the op mode of AP to Gateway mode.

\* "echo 1 > /proc/sw\_nat" : change the op mode of AP to Bridge mode.

\* "echo 2 > /proc/sw\_nat" : change the op mode of AP to WISP mode.

Output Format:

\$FLAG

Output Para:

\* 0: the op mode of AP is Gateway mode.

\* 1: the op mode of AP is Bridge mode.

\* 2: the op mode of AP is WISP mode.

## 6.16 hw\_nat

Note: proc hw\_nat exist when CONFIG\_RTL\_HARDWARE\_NAT is defined.

Path: /proc/hw\_nat

Description: Flags for hardware NAT control

Input Format:

echo "\$FLAG" > /proc/hw\_nat

Input Para:

\* the unit is non-zero: hardware NAT enabled

\* the unit is 0: hardware NAT disabled

\* "echo 0 > /proc/hw\_nat" : hardware NAT disabled, change to gateway mode.

- \* “echo 1 > /proc/hw\_nat” : hardware NAT enabled, change to gateway mode.
- \* “echo 2 > /proc/hw\_nat” : Change to bridge mode.
- \* “echo 3 > /proc/hw\_nat” : Change to WISP mode.
- \* “echo 4 > /proc/hw\_nat” : simply disabled the hardware NAT.
- \* “echo 5 > /proc/hw\_nat” : simply disabled the hardware NAT.
- \* “echo 8 > /proc/hw\_nat” : simply disabled the hardware NAT.
- \* “echo 9 > /proc/hw\_nat” : init hardware NAT parameters. (Must init before hardware NAT works)

Output Format:

\$FLAG

Output Para:

- \* 0: gateway mode & hardware NAT disabled.
- \* 1: gateway mode & hardware NAT enabled.
- \* 2: bridge mode.
- \* 3: WISP mode.
- \* 4: hardware NAT disabled.
- \* 5: hardware NAT disabled.
- \* 8: hardware NAT disabled.
- \* 9: hardware NAT parameters have already initialized.
- \* others: no means

## 6.17 br\_wlanblock

Path: /proc/br\_wlanblock

Description: Enable/Disable wlan block.

Input Format:

Echo “\$FLAG” > /proc/ br\_wlanblock

Input Para:

- \* 1: Enable wlan block relay.
- \* 0: Disable wlan block relay.

Output Format:

\$FLAG

Output Para:

- \* 1: wlan block relay enabled.
- \* 0: wlan block relay disabled.

## 6.18 br\_igmpVersion

Path: /proc/br\_igmpVersion

Description: The IGMP version support.

Input Format:

Echo “\$FLAG” > /proc/ br\_igmpVersion

Input Para:

\* 2: IGMP version 2.

\* 3: IGMP version 3.

Output Format:

\$FLAG

Output Para:

\* 2: IGMP version 2.

\* 3: IGMP version 3.

## 6.19 br\_igmpquery

Path: /proc/br\_igmpquery

Description: Enable/Disable IGMP query.

Input Format:

Echo “\$FLAG” > /proc/ br\_igmpquery

Input Para:

\* 1: Enable IGMP query.

\* 0: Disable IGMP query.

Output Format:

\$FLAG

Output Para:

\* 1: IGMP query enabled.

\* 0: IGMP query disabled.

## 6.20 br\_mCastFastFwd

Path: /proc/ br\_mCastFastFwd

Description: Enable/Disable ipMulticastFastFwd.

Input Format:

Echo “\$FLAG1 \$FLAG2” > /proc/ br\_mCastFastFwd

Input Para.:

1)\$FLAG1:

\*1: ipMulticastFastFwd enabled

\*0: ipMulticastFastFwd disabled

2)\$FLAG2:

\*1: needCheckMfc enabled

\*0: needCheckMfc disabled

Output Format:

\$FLAG1,\$FLAG2

Output Para:

1) \$FLAG1:  
    \*1: ipMulticastFastFwd enabled (ip multicast fast forward)  
    \*0: ipMulticastFastFwd disabled

2) \$FLAG2:  
    \*1: needCheckMfc enabled (need check multicast forwarding cache)  
    \*0: needCheckMfc disabled

## 6.21 br\_mldquery

Path: /proc/ br\_mldquery

Description: Enable/Disable mld query.

Input Format:

Echo “\$FLAG” > /proc/ br\_mldquery

Input Para.:

- \* 1: Enable mld query.
- \* 0: Disable mld query.

Output Format:

\$FLAG

Output Para.:

- \* 1: mld query enabled.
- \* 0: mld query disabled.

## 6.22 br\_mldsnoop

Path: /proc/ br\_mldsnoop

Description: Enable/Disable mld snooping.

Input Format:

Echo “\$FLAG” > /proc/ br\_mldsnoop

Input Para.:

- \* 1: Enable mld snoop.
- \* 0: Disable mld snoop.

Output Format:

\$FLAG

Output Para.:

- \* 1: mld snooping enabled.
- \* 0: mld snooping disabled.

## 6.23 eee

Path: /proc/eee

Description: Enable/Disable eee.

Input Format:

Echo “\$FLAG” > /proc/ eee

Input Para:

\* 1: Enable eee.

\* 0: Disable eee.

Output Format:

\$FLAG

Output Para:

\* 1: eee enabled.

\* 0: eee disabled.

## 6.24 gc\_overflow\_timeout

Path: /proc/gc\_overflow\_timeout

Description: gc\_overflow\_timeout.

Input Format:

Echo “\$FLAG” > /proc/ gc\_overflow\_timeout

Input Para:

\$FLAG: rtl\_gc\_overflow\_timout

Output Format:

\$FLAG1 \$FLAG2

Output Para:

\$FLAG1: rtl\_gc\_overflow\_timout.

\$FLAG2: Hz(100).

## 6.25 load\_default

Path: /proc/load\_default

Description: Enable/Disable load\_default.

Input Format:

Echo “\$FLAG” > /proc/ load\_default

Input Para:

\* 1: Enable load\_default.

\* 0: Disable load\_default.

Output Format:

Output Para:

## 6.26 reInitSwitchCore

Path: /proc/ reInitSwitchCore

Description: do rtl865x\_reinitSwitchCore or not.

Input Format:

Echo “\$FLAG” > /proc/ reInitSwitchCore

Input Para:

- \* 0: do nothing
- \* non-zero value: rtl865x\_reinitSwitchCore

Output Format:

none

Output Para:

none

Note: not supported in kernel 3.10.

## 6.27 br\_mldVersion

Path: /proc/ br\_mldVersion

Description: the MLD Version supported.

Input Format:

Echo “\$FLAG” > /proc/ br\_mldVersion

Input Para:

- \* 1: MLD version 1.
- \* 2: MLD version 2.

Output Format:

\$FLAG

Output Para:

- \* 1: MLD version 1.
- \* 2: MLD version 2.

## 6.28 jate

Path: /proc/jate

Description: this file is added for JATE (the Japan Approvals Institute for Telecommunications Equipment) test. This feature is disabled by default and will be enabled after defined “CONFIG\_RTL\_JATE\_TEST” compile flag in /linux-2.6.30/drivers/net/rtl819x/rtl\_nic.c(for kernel 2.6) or /linux-3.10/drivers/net/rtl819x/rtl\_nic.c(for kernel 3.10).

Input Format:

Echo “\$FLAG” > /proc/jate

Input Para.:

- \* 1: port setting command:

echo port <port number> <mode> > /proc/jate

the valid value of port number is 0 ~ 5, 5 mean all ports.

the valid value of mode is 10H/10F/100H/100F/Auto.

for example:

```
echo port 0 10H > /proc/jate
echo port 1 10F > /proc/jate
echo port 2 100H > /proc/jate
echo port 3 100F > /proc/jate
echo port 5 Auto > /proc/jate
```

\* 2: packet transmitted command:

```
echo tx <port mask> <packet type> <packet output time> > /proc/jate
```

the valid value of port mask are 1 ~ 0x1f, 1 means port 0, 2 means port 1, 0x1f means all ports

the valid value of packet type are 0~2, 0: random data pattern, 1: all0 data pattern; 2: all1 data pattern

the valid value of packet output time is from 1 to 1000 (unit: second)

for example:

```
echo tx 1 0 > /proc/jate
echo tx 2 1 > /proc/jate
echo tx 1f 2 > /proc/jate
```

Output Format:

none

Output Para.:

none

## 6.29 gpio\_ctrl

Path: /proc/wlan0/gpio\_ctrl

Description: GPIO control for 8188ER, 8192CE, 8188RE, 8192D

Input Format:

```
Echo $CMD $GPIO $ACT > /proc/wlan0/gpio_ctrl
```

Input Para:

1) \$CMD:

- \* config: Config GPIO pins to be input or output
- \* set: Write value to specified GPIO pin

2) \$GPIO:

- \* 0~11: Specify GPIO pin number

3) \$ACT:

- \* r/w: When \$CMD is config, specify GPIO pin to be input or output

\* 0/1: When \$CMD is set, write value to GPIO pin

Output Format:

GPIO\$M \$N

Output Para:

1) \$M:

\* 0~11: The GPIO pin which is specified as input

2) \$N:

\* 0/1: The GPIO status of the GPIO pin

for example:

|                                         |                                  |
|-----------------------------------------|----------------------------------|
| echo config 4 w > /proc/wlan0/gpio_ctrl | (Config GPIO 4 to be output pin) |
| echo set 4 1 > /proc/wlan0/gpio_ctrl    | (Set GPIO 4 output high)         |
| echo set 4 0 > /proc/wlan0/gpio_ctrl    | (Set GPIO 4 output low)          |

## 6.30 Jumbo frame support

Path: /proc/jumbo\_frame\_support

Description: It controls jumbo frame setting on switch core.

Input Format:

Echo \$FLAG \$SIZE > /proc/jumbo\_frame\_support

Input Para:

1) \$ FLAG:

\* 0: disable jumbo frame support

\* 1: enable jumbo frame support

2) \$ SIZE:

\* 9000: Specify max length of jumbo frame supported.

\* 16000: Specify max length of jumbo frame supported.

Output Format:

None

Output Para:

None

for example:

1) Set AP to support 9K jumbo frame

echo 1 9000 > /proc/jumbo\_frame\_support

/\*Don't forget to change interface's mtu accordingly when enable jumbo frame\*/

ifconfig eth0 mtu 9216

ifconfig br0 mtu 9216

ifconfig eth1 mtu 9216

2) Set AP to support 16K jumbo frame

```
echo 1 16000 > /proc/jumbo_frame_support
```

/\*Don't forget to change interface's mtu accordingly when enable jumbo frame\*/

```
ifconfig eth0 mtu 16370
```

```
ifconfig br0 mtu 16370
```

```
ifconfig eth1 mtu 16370
```

3) disable jumbo frame support

```
echo 0 0 > /proc/jumbo_frame_support
```

/\*Don't forget to change interface's mtu accordingly when disable jumbo frame\*/

```
ifconfig br0 mtu 1500
```

```
ifconfig eth0 mtu 1500
```

```
ifconfig eth1 mtu 1500
```

## 6.31 port\_status

Path: /proc/rtl865x/port\_status

Description: display Ethernet port status (cat /proc/rtl865x/port\_status) or set Ethernet port speed and duplex mode (echo ... > /proc/rtl865x/port\_status).

Input Format:

```
echo port $PORT_MASK $MODE > /proc/rtl865x/port_status
```

Input Para:

1) \$PORT\_MASK:

decimal format

bit 0 ~ 4: port 0 ~ 4

2) \$MODE:

10\_half: 10M half-duplex

10\_full: 10M full-duplex

100\_half: 100M half-duplex

100\_full: 100M full-duplex

1000\_half: 1000M half-duplex

1000\_full: 1000M full-duplex

the other character: auto

Output Format:

None

Output Para:

None

for example:

```
echo port 17 10_full > /proc/rtl865x/port_status
==> set port 0 and port 4 to "10M full-duplex" (17 = 0x11, bit 4 and bit 0 = 1)
```

## 7 FAQ

### 7.1 How to modify the mappings of LAN/WAN port?

Modify linux-2.6.30/include/net/rtl/rtl865x\_netif.h(for kernel 2.6) or linux-3.10/include/net/rtl/rtl865x\_netif.h (for kernel 3.10), and define RTL\_WANPORT\_MASK and RTL\_LANPORT\_MASK as needed.

RTL\_LANPORT\_MASK\_x (x is 1,2,3,4) should be adjust also if CONFIG\_RTL\_MULTI\_LAN\_DEV enabled.

For example: change wan port to port 0

```
#define RTL_WANPORT_MASK 0x01
#define RTL_LANPORT_MASK 0x11E
#define RTL_LANPORT_MASK_1 0x10
#define RTL_LANPORT_MASK_2 0x08
#define RTL_LANPORT_MASK_3 0x04
#define RTL_LANPORT_MASK_4 0x02
```

In 8197D+8367RB platform, please also change the “RTL8367R\_WAN” definition in /linux-2.6.30/drivers/net/rtl819x/AsicDriver/rtl865xc\_asicregs.h(for kernel 2.6) and /linux-2.6.30/drivers/net/rtl819x/rtl8367r/rtk\_api.c(for kernel 2.6).

or:

/linux-3.10/drivers/net/rtl819x/AsicDriver/rtl865xc\_asicregs.h(for kernel 3.10) and /linux-3.10/drivers/net/rtl819x/rtl8367r/rtk\_api.c(for kernel 3.10).

### 7.2 How to change the baud rate of console?

Modify bootloader code file btcode/start.h and boot/init/utility.h, define BAUD\_RATE as needed.

### 7.3 How to use vlan priority option of iptables?

Firstly, Linux kernel configure as follows:

```
① Networking support --->
 Networking options --->
 Network packet filtering framework (Netfilter) --->
 Core Netfilter Configuration --->
 "VLAN" match support //Selected
```

② Device Drivers --->

Network device support --->

Options for Realtek SoC --->

Support HW Qos //Selected

Config for Layered Driver Features --->

HW Qos support vlan priority //Selected

Secondly , one example of iptables rule using vlan priority option as follows:

// Mapping all the packets with vlan priority 3 to skb MARK 13

iptables -A PREROUTING -t mangle -m vlanpriority --prio-value 3 -j MARK --set-mark 13

## 7.4 Relationship of virtual AP and root AP?

Relationship of virtual AP and root AP is as follows:

- ① Since there is only one wireless IC, virtual AP and root AP use the same hardware.
- ② Boot sequence of our AP is that root AP is initialized firstly and virtual AP is initialized then.
- ③ Hardware settings of virtual AP such as band and channel should be subset of root AP hardware settings. Since our hardware support multiple data rate at the same, data rate of virtual AP can be set different from root AP.
- ④ Since virtual AP is also the software AP of our AP, the software settings of virtual AP such as SSID, WMM, ACCESS, ENCRYPTION etc can be set different from root AP.

## 7.5 How to customize icon of lld2d?

Steps to customize icon of lld2d as follows:

- ① Add Icon, and install to /etc/
- ② Modify lld2d.conf, set "jumbo-icon" value, and install lld2d.conf to /etc/

One example to customize icon of lld2d as follows:

- ① Add Icon: realsil\_gw.ico, Install to /etc/realsil\_gw.ico
- ② Modify lld2d.conf: jumbo-icon = /etc/realsil\_gw.ico

And install lld2d.conf to /etc/lld2d.conf

## 7.6 How to rename the AP when shown on Windows?

Please input command “flash set DEVICE\_NAME ‘name’ ” in console and then reboot AP.

One example as follows: If you want AP show the name ‘BARLX48’ on Windows, do command “flash set DEVICE\_NAME BARLX48” in console, and then reboot AP.

## 7.7 How to add MAC address filter based on ether driver API?

1) Solution 1: Use ACL rule to do source or destination MAC address filter.

Program APIs as follows:

```
int32 rtl865x_add_acl(rtl865x_AclRule_t *rule, char *netifName,int32 priority);
int32 rtl865x_del_acl(rtl865x_AclRule_t *rule, char *netifName,int32 priority);
```

Example code for solution 1 as follows:

```
int32 retval;
rtl865x_AclRule_t rule;

// To add acl rule for smac(00:1e:c9:3b:b3:44) filter
bzero((void*)&rule,sizeof(rule));
rule.ruleType_ = RTL865X_ACL_MAC;
rule.actionType_ = RTL865X_ACL_DROP;
rule.pktOpApp_ = RTL865X_ACL_ALL_LAYER;
rule.srcMac_.octet[0]=0x00;
rule.srcMac_.octet[1]=0x1e;
rule.srcMac_.octet[2]=0xc9;
rule.srcMac_.octet[3]=0x3b;
rule.srcMac_.octet[4]=0xb3;
rule.srcMac_.octet[5]=0x44;

rule.srcMacMask_.octet[0]=0xFF;
rule.srcMacMask_.octet[1]=0xFF;
rule.srcMacMask_.octet[2]=0xFF;
rule.srcMacMask_.octet[3]=0xFF;
rule.srcMacMask_.octet[4]=0xFF;
rule.srcMacMask_.octet[5]=0xFF;

retval= rtl865x_add_acl(&rule, "br0", RTL865X_ACL_SYSTEM_USED);

// To del acl rule for smac(00:1e:c9:3b:b3:44) filter
bzero((void*)&rule,sizeof(rule));
rule.ruleType_ = RTL865X_ACL_MAC;
rule.actionType_ = RTL865X_ACL_DROP;
rule.pktOpApp_ = RTL865X_ACL_ALL_LAYER;
rule.srcMac_.octet[0]=0x00;
```

```

rule.srcMac_.octet[1]=0x1e;
rule.srcMac_.octet[2]=0xc9;
rule.srcMac_.octet[3]=0x3b;
rule.srcMac_.octet[4]=0xb3;
rule.srcMac_.octet[5]=0x44;

rule.srcMacMask_.octet[0]=0xFF;
rule.srcMacMask_.octet[1]=0xFF;
rule.srcMacMask_.octet[2]=0xFF;
rule.srcMacMask_.octet[3]=0xFF;
rule.srcMacMask_.octet[4]=0xFF;
rule.srcMacMask_.octet[5]=0xFF;

retval= rtl865x_del_acl(&rule, "br0", RTL865X_ACL_SYSTEM_USED);

```

2) Solution 2: Use source block via hardware L2 table to do source MAC address filter.

Program APIs as follows:

```

int32 rtl865x_addFilterDatabaseEntryExtension (uint16 fid, rtl865x_filterDbTableEntry_t * L2entry);
int32 rtl865x_delFilterDatabaseEntry (uint16 l2Type, uint16 fid, ether_addr_t * macAddr);

```

Example code for solution 2 as follows:

```

int32 retval;
uint32 fid;
rtl865x_filterDbTableEntry_t l2temp_entry;
ether_addr_t macAddr;

// To add hardware L2 table entry for smac(00:1e:c9:3b:b3:44) filter
fid=RTL_LAN_FID;
memset((void *)&l2temp_entry, 0, sizeof(l2temp_entry));
l2temp_entry.l2type = RTL865x_L2_TYPEII;
l2temp_entry.process = FDB_TYPE_SRCBLK;
l2temp_entry.memberPortMask = RTL_LANPORT_MASK;
l2temp_entry.auth = FALSE;
l2temp_entry.SrcBlk = TRUE;

l2temp_entry.macAddr.octet[0]=0x00;
l2temp_entry.macAddr.octet[1]=0x1e;

```

```

l2temp_entry.macAddr.octet[2]=0xc9;
l2temp_entry.macAddr.octet[3]=0x3b;
l2temp_entry.macAddr.octet[4]=0xb3;
l2temp_entry.macAddr.octet[5]=0x44;

retval=rtl865x_addFilterDatabaseEntryExtension(fid, &l2temp_entry);

// To del hardware L2 table entry for smac(00:1e:c9:3b:b3:44) filter
fid=RTL_LAN_FID;

macAddr.octet[0]=0x00;
macAddr.octet[1]=0x1e;
macAddr.octet[2]=0xc9;
macAddr.octet[3]=0x3b;
macAddr.octet[4]=0xb3;
macAddr.octet[5]=0x44;

retval = rtl865x_delFilterDatabaseEntry(RTL865x_L2_TYPEII, fid, &macAddr);

```

## 7.8 How to adjust CPU speed?

For hardware designer, please refer to Fig 7.8.1 and adjust signal MA11~9 (CPUClkSel[2:0]) to select CPU clock from 250MHz, 270 MHz, 290 MHz, 310 MHz, 330 MHz, 350 MHz, 370 MHz and 390 MHz.

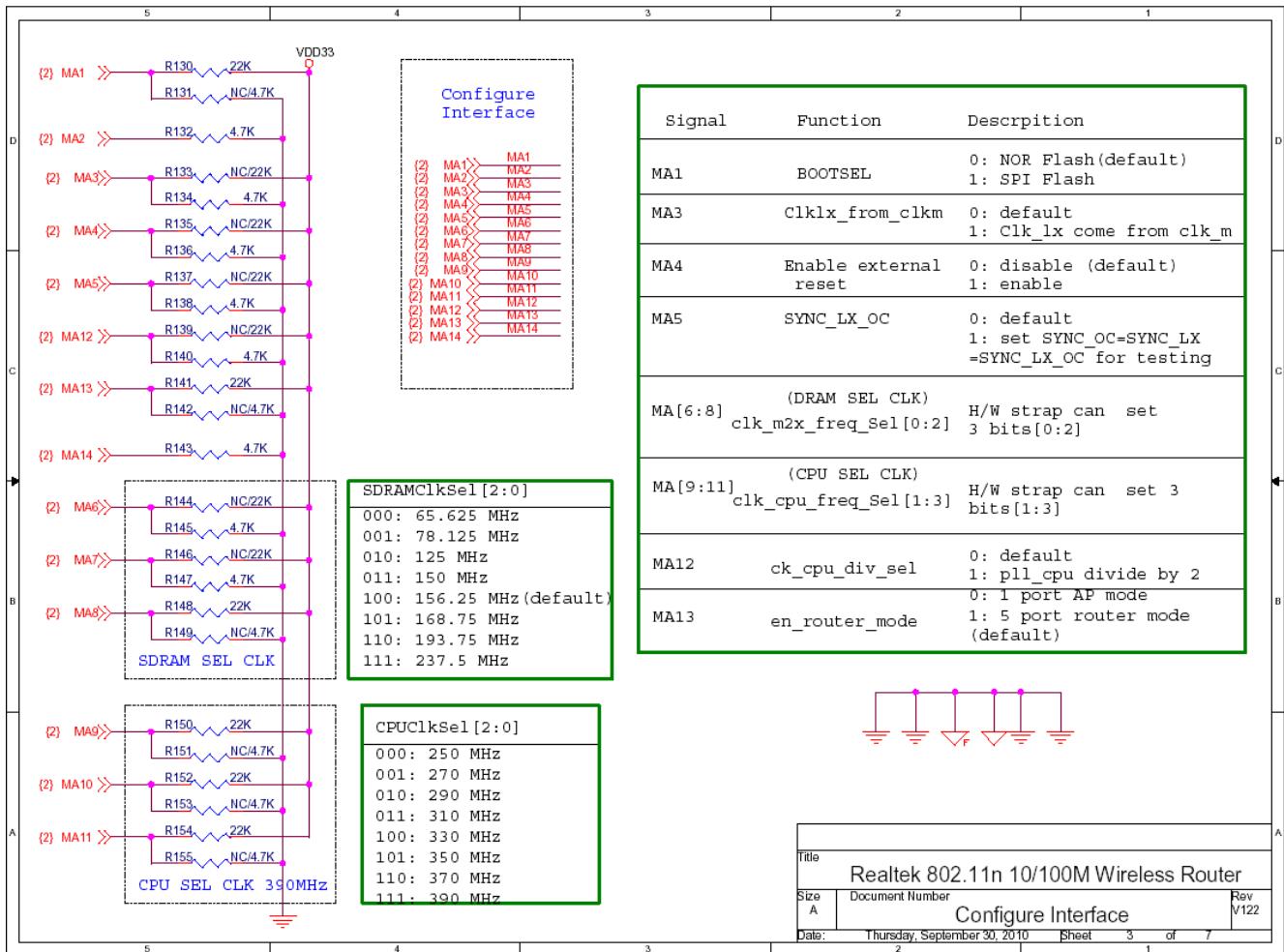


Fig 7.8.1 Realtek 802.11n 10/100M Wireless Router Interface Circuit

## 7.9 How to shutdown power of Ethernet ports?

1) For hardware designer, please refer to Fig 7.8.1 above and modify as follows:

Set MA13=0: port0~3 closed and only port4 opened.

Set MA13=1: port0~4 opened.

2) For software designer, “#define CONFIG\_RTL\_PHY\_POWER\_CTRL” in linux-2.6.30/include/net/rtl/rtl\_nic.h(for kernel 2.6) or linux-3.10/include/net/rtl/rtl\_nic.h(for kernel 3.10), rebuild image and update it to AP.

After AP boots up, control the power of Ethernet ports as follows:

```
echo "portmask phystats" > /proc/phyPower // 0: power off; 1: power on.
```

For example:

```
echo "0x1f 1" > /proc/phyPower // set port0~port5 power on, don't change other port status.
```

```
echo "0x03 1" > /proc/phyPower // set Ethernet port0 and port1 power on, don't change other port status.
```

```
echo "0x03 0" > /proc/phyPower // set Ethernet port0 and port1 power off, don't change other port status.
```

## 7.10 How to disable Ethernet PHY?

Solution:

- 1) bit0 of Port Configuration Register set to 0.
- 2) Use API rtl865x\_disableDevPortForward at linux-2.6.30/driver/net/rtl819x/rtl\_nic.c(**for kernel 2.6**) or linux-3.10/driver/net/rtl819x/rtl\_nic.c(**for kernel 3.10**)

## 7.11 Which channels are supported by RTL8192D?

Answer:

- 1) Channels of 2.4G Band supported by RTL8192D are listed as follows.

Note:

at present, there aren't so many channel number supported at the webpage of AP, the reason is that the webpage is just for demo. The webpage of AP can be customized to support all the channel number which are supported by RTL8192D.

| 2.4G Band      |                         | Regulatory Domains |             |               |       |        |       |               | RTL8192D |     |
|----------------|-------------------------|--------------------|-------------|---------------|-------|--------|-------|---------------|----------|-----|
| Channel Number | Channel Frequency (MHz) | US (FCC)           | Canada (IC) | Europe (ETSI) | Spain | France | Japan | Japan (MPHPT) | 20M      | 40M |
| 1              | 2412                    | v                  | v           | v             |       |        |       | v             | v        |     |
| 2              | 2417                    | v                  | v           | v             |       |        |       | v             | v        |     |
| 3              | 2422                    | v                  | v           | v             |       |        |       | v             | v        | v   |
| 4              | 2427                    | v                  | v           | v             |       |        |       | v             | v        | v   |
| 5              | 2432                    | v                  | v           | v             |       |        |       | v             | v        | v   |
| 6              | 2437                    | v                  | v           | v             |       |        |       | v             | v        | v   |
| 7              | 2442                    | v                  | v           | v             |       |        |       | v             | v        | v   |
| 8              | 2447                    | v                  | v           | v             |       |        |       | v             | v        | v   |
| 9              | 2452                    | v                  | v           | v             |       |        |       | v             | v        | v   |
| 10             | 2457                    | v                  | v           | v             | v     | v      |       | v             | v        | v   |
| 11             | 2462                    | v                  | v           | v             | v     | v      |       | v             | v        | v   |
| 12             | 2467                    |                    |             | v             |       | v      |       | v             | v        | v   |
| 13             | 2472                    |                    |             | v             |       | v      |       | v             | v        |     |
| 14             | 2484                    |                    |             |               |       |        | v     |               | v        |     |

- 2) Channels of 5G Band supported by RTL8192D are listed as follows.

Note:

at present, there aren't so many channel number supported at the webpage of AP, the reason is that the webpage is just for demo. The webpage of AP can be customized to support all the channel number which are supported by RTL8192D.

| 5G Band        |                         | Regulatory Domains |         |                     |         |                           |         | RTL8192D |     |
|----------------|-------------------------|--------------------|---------|---------------------|---------|---------------------------|---------|----------|-----|
| Channel Number | Channel Frequency (MHz) | US (FCC 15.407)    |         | Europe (CE 301 893) |         | Japan (MIC Item 19-3, 19- |         | 20M      | 40M |
|                |                         | (20MHz)            | (40MHz) | (20MHz)             | (40MHz) | (20MHz)                   | (40MHz) |          |     |
| 34             | 5170                    |                    |         |                     |         |                           |         |          |     |
| 36             | 5180                    | v                  |         | v                   |         | v                         |         | v        |     |
| 38             | 5190                    |                    | v       |                     | v       |                           | v       |          | v   |
| 40             | 5200                    | v                  |         | v                   |         | v                         |         | v        |     |
| 42             | 5210                    |                    | Turbo   |                     |         |                           |         |          |     |
| 44             | 5220                    | v                  |         | v                   |         | v                         |         | v        |     |
| 46             | 5230                    |                    | v       |                     | v       |                           | v       |          | v   |
| 48             | 5240                    | v                  |         | v                   |         | v                         |         | v        |     |
| 50             | 5250                    |                    | Turbo   |                     |         |                           |         |          |     |
| 52             | 5260                    | v                  |         | v                   |         | v                         |         | v        |     |
| 54             | 5270                    |                    | v       |                     | v       |                           | v       |          | v   |
| 56             | 5280                    | v                  |         | v                   |         | v                         |         | v        |     |
| 58             | 5290                    |                    | Turbo   |                     |         |                           |         |          |     |
| 60             | 5300                    | v                  |         | v                   |         | v                         |         | v        |     |
| 62             | 5310                    |                    | v       |                     | v       |                           | v       |          | v   |
| 64             | 5320                    | v                  |         | v                   |         | v                         |         | v        |     |
| 100            | 5500                    | v                  |         | v                   |         | v                         |         | v        |     |
| 102            | 5510                    |                    | v       |                     | v       |                           | v       |          | v   |
| 104            | 5520                    | v                  |         | v                   |         | v                         |         | v        |     |
| 106            | 5530                    |                    |         |                     |         |                           |         |          |     |
| 108            | 5540                    | v                  |         | v                   |         | v                         |         | v        |     |
| 110            | 5550                    |                    | v       |                     | v       |                           | v       |          | v   |
| 112            | 5560                    | v                  |         | v                   |         | v                         |         | v        |     |
| 114            | 5570                    |                    |         |                     |         |                           |         |          |     |
| 116            | 5580                    | v                  |         | v                   |         | v                         |         | v        |     |
| 118            | 5590                    |                    | v       |                     | v       |                           | v       |          | v   |
| 120            | 5600                    | v                  |         | v                   |         | v                         |         | v        |     |
| 122            | 5610                    |                    |         |                     |         |                           |         |          |     |
| 124            | 5620                    | v                  |         | v                   |         | v                         |         | v        |     |
| 126            | 5630                    |                    | v       |                     | v       |                           | v       |          | v   |
| 128            | 5640                    | v                  |         | v                   |         | v                         |         | v        |     |
| 130            | 5650                    |                    |         |                     |         |                           |         |          |     |
| 132            | 5660                    | v                  |         | v                   |         | v                         |         | v        |     |
| 134            | 5670                    |                    | v       |                     | v       |                           | v       |          | v   |
| 136            | 5680                    | v                  |         | v                   |         | v                         |         | v        |     |
| 138            | 5690                    |                    |         |                     |         |                           |         |          |     |
| 140            | 5700                    | v                  |         | v                   |         | v                         |         | v        |     |
| 149            | 5745                    | v                  |         |                     |         | v                         |         | v        |     |
| 151            | 5755                    |                    | v       |                     |         |                           | v       |          | v   |
| 153            | 5765                    | v                  |         |                     |         | v                         |         | v        |     |
| 155            | 5775                    |                    |         |                     |         |                           |         |          |     |
| 157            | 5785                    | v                  |         |                     |         | v                         |         | v        |     |
| 159            | 5795                    |                    | v       |                     |         |                           | v       |          | v   |
| 161            | 5805                    | v                  |         |                     |         | v                         |         | v        |     |
| 163            | 5815                    |                    |         |                     |         |                           |         |          |     |
| 165            | 5825                    | v                  |         |                     |         | v                         |         | v        |     |

## 7.12 How to set speed and duplex of physical port via APIs?

Answer:

In order to set speed and duplex of physical port, the related APIs are as follows:

- 1) int32 rtl865xC\_setAsicEthernetForceModeRegs (uint32 port, uint32 enForceMode, uint32 forceLink, uint32 forceSpeed, uint32 forceDuplex);
- 2) int32 rtl8651\_setAsicEthernetPHYSpeed (uint32 port, uint32 speed);
- 3) int32 rtl8651\_setAsicEthernetPHYDuplex( uint32 port, uint32 duplex );
- 4) int32 rtl8651\_setAsicEthernetPHYAutoNeg( uint32 port, uint32 autoneg);
- 5) int32 rtl8651\_setAsicEthernetPHYAdvCapalitity(uint32 port, uint32 capalitity);
- 6) int32 rtl8651\_restartAsicEthernetPHYNway (uint32 port);

One example to set port 2 at forcemode duplex 100Mbps as follows:

```
#define SPEED10M 0
#define SPEED100M 1
#define SPEED1000M 2

enum
{
 PORT_DOWN=0,
 HALF_DUPLEX_10M,
 HALF_DUPLEX_100M,
 HALF_DUPLEX_1000M,
 DUPLEX_10M,
 DUPLEX_100M,
 DUPLEX_1000M,
 PORT_AUTO
};

/* Customized */
uint32 port=2;
int forceMode=TRUE;
int forceLink=TRUE;
int forceLinkSpeed=SPEED100M;
int forceDuplex=TRUE;
uint32 advCapability=(1<<DUPLEX_100M);

rtl865xC_setAsicEthernetForceModeRegs(port, forceMode, forceLink, forceLinkSpeed,
forceDuplex);
rtl8651_setAsicEthernetPHYSpeed(port,forceLinkSpeed);
rtl8651_setAsicEthernetPHYDuplex(port,forceDuplex);
//Note: if use force mode, auto negotiation must be disabled.
rtl8651_setAsicEthernetPHYAutoNeg (port,FALSE);
rtl8651_setAsicEthernetPHYAdvCapalitity (port,advCapability);
rtl8651_restartAsicEthernetPHYNway (port);
```

## 7.13 How to dial up pppoe for test when wan interface is changed?

Answer:

One example, interface usb\_3g is used as wan interface to dial up pppoe.

For test, pppoe script (named pppoe\_test.sh for example) can be used as follows:

```
#!/bin/sh

WAN=$2

OPTIONS=/etc/ppp/options
PAPFILE=/etc/ppp/pap-secrets
CHAPFILE=/etc/ppp/chap-secrets
RESOLV=/etc/ppp/resolv.conf
LINKFILE=/etc/ppp/link
PPPFILE=/var/run/ppp
FIRSTFILE=/etc/ppp/first
FIRSTDEMAND=/etc/ppp/firstdemand
CONNECTFILE=/etc/ppp/connectfile
DNRDPIDFILE=/var/run/dnrd.pid

Customized here for test
eval `flash set WAN_DHCP 3`

eval `flash set PPP_USER_NAME "zj2"`

eval `flash set PPP_PASSWORD "123"`

eval `flash set PPP_IDLE_TIME 300`

eval `flash set PPP_CONNECT_TYPE 0`

eval `flash set PPP_MTU_SIZE 1460`

eval `flash set DNS_MODE 0`

eval `flash set DNS1 0.0.0.0`

eval `flash set DNS2 0.0.0.0`

eval `flash set DNS3 0.0.0.0`

eval `flash get WAN_DHCP`

eval `flash get PPP_USER_NAME`

eval `flash get PPP_PASSWORD`

eval `flash get PPP_IDLE_TIME`

eval `flash get PPP_CONNECT_TYPE`

eval `flash get PPP_MTU_SIZE`

eval `flash get DNS_MODE`

eval `flash get DNS1`

eval `flash get DNS2`

eval `flash get DNS3`

ifconfig $WAN 0.0.0.0
if [$1 = 'connect']; then
 ENABLE_CONNECT=1
else
 ENABLE_CONNECT=0
fi
if [-n "$PPP_USER_NAME"] ; then
 echo "name \"$PPP_USER_NAME\" > $OPTIONS
 echo "############################ > $PAPFILE
 echo "\"$PPP_USER_NAME\" * \"$PPP_PASSWORD\" >> $PAPFILE
 echo "############################ > $CHAPFILE
 echo "\"$PPP_USER_NAME\" * \"$PPP_PASSWORD\" >> $CHAPFILE
fi
echo "noauth" >>$OPTIONS
echo "nomppc" >>$OPTIONS
echo "noipdefault" >> $OPTIONS
echo "hide-password" >> $OPTIONS
echo "defaultroute" >> $OPTIONS
```

```

echo "persist" >> $OPTIONS
echo "ipcp-accept-remote" >> $OPTIONS
echo "ipcp-accept-local" >> $OPTIONS
echo "nodetach" >> $OPTIONS
echo "usepeerdns" >> $OPTIONS
echo "mtu $PPP_MTU_SIZE" >> $OPTIONS
echo "mru $PPP_MTU_SIZE" >> $OPTIONS
echo "lcp-echo-interval 20" >> $OPTIONS
echo "lcp-echo-failure 3" >> $OPTIONS
echo "wantype $WAN_DHCP" >> $OPTIONS
echo "holdoff 10" >> $OPTIONS

if [-n "$PPP_SERVICE_NAME"]; then
 echo "plugin /etc/ppp/plugins/libplugin.a rp_pppoe_service $PPP_SERVICE_NAME $WAN" >>
$OPTIONS
else
 echo "plugin /etc/ppp/plugins/libplugin.a $WAN" >> $OPTIONS
fi

PID_FILE=/var/run/ppp0.pid
DNRD_PID=/var/run/dnrd.pid

if [! -f $DNRD_PID]; then
 DNS="--cache=off"
 if [$DNS_MODE != 1]; then
 dnrd $DNS -s 168.95.1.1
 fi
 if [$DNS_MODE = 1]; then
 if ["$DNS1" != '0.0.0.0']; then
 DNS="$DNS -s $DNS1"
 fi
 if ["$DNS2" != '0.0.0.0']; then
 DNS="$DNS -s $DNS2"
 fi
 if ["$DNS3" != '0.0.0.0']; then
 DNS="$DNS -s $DNS3"
 fi
 dnrd $DNS
 fi
fi

if [-r "$PPPFILE"]; then
 rm $PPPFILE
fi

killall -15 pppd 2>/dev/null

if [-r "$CONNECTFILE"]; then
 rm -f $CONNECTFILE
fi

if [$PPP_CONNECT_TYPE = 0] ; then
{
 while [true]; do
 if [$WAN_DHCP != 3] // [$PPP_CONNECT_TYPE != 0]; then
 break
}

```

```

fi
if [! -r "$CONNECTFILE"] && [$PPP_CONNECT_TYPE = 0]; then
 echo "pass" > $CONNECTFILE
 if [! -f $FIRSTFILE]; then
 echo "pass" > $FIRSTFILE
 fi
 pppd
fi
sleep 5
done
rm -f $FIRSTFILE
} &
fi

if [$PPP_CONNECT_TYPE = 1]; then
{
 echo "demand" >> $OPTIONS
 echo "idle $PPP_IDLE_TIME" >> $OPTIONS

 while [true]; do

 if [$WAN_DHCP != 3] // [$PPP_CONNECT_TYPE != 1]; then
 break
 fi
 if [! -r "$CONNECTFILE"] && [$PPP_CONNECT_TYPE = 1]; then
 echo "pass" > $CONNECTFILE
 if [! -f $FIRSTDEMAND]; then
 echo "pass" > $FIRSTDEMAND
 fi
 pppd
 if [-f $DNRDPIDFILE]; then
 PID=`cat $DNRDPIDFILE`
 kill -9 $PID
 rm -f $DNRDPIDFILE
 fi
 dnrd --cache=off -s 168.95.1.1
 fi
 sleep 5
 done
 rm -f $FIRSTDEMAND
 } &
 fi

 if [$PPP_CONNECT_TYPE = 2]; then
 if [$ENABLE_CONNECT = 1]; then
 pppd &
 fi
 fi
}

```

Test steps as follows:

Step 1): add pppoe\_test.sh as above in users/script/cinit/.

Step 2): make sdk and update image.

Step 3): after AP boots up, at AP console, input command as follows:

*/bin/sh -x /bin/pppoe\_test2.sh all usb\_3g &*

Test result: pppoe should dial up via wan interface usb\_3g.

Debug:

At AP console, cd /etc/ppp/ and check settings for pppoe as follows:

1) chap-secrets and pap-secrets is secret settings, one example as follows:

```
cat chap-secrets
#####
"zj2" * "123"
cat pap-secrets
#####
"zj2" * "123"
```

2) options is config settings for pppoe, one example as follows:

```
cat options
name "zj2"
noauth
nomppc
noipdefault
hide-password
defaultroute
persist
ipcp-accept-remote
ipcp-accept-local
nodetach
usepeerdns
mtu 1460
mru 1460
lcp-echo-interval 20
lcp-echo-failure 3
wantype 3
holdoff 10
plugin /etc/ppp/plugins/libplugin.a eth1
```

## 7.14 How to enable/disable ipv6 passthru?

Answer:

1.1) At gateway mode, enable ipv6 passthru as follows:

```
ifconfig peth0 up
brctl addif br0 peth0
echo "1">>/proc/custom_Passthru
```

1.2) At gateway mode, disable ipv6 passthru as follows:

```
echo "0">>/proc/custom_Passthru
brctl delif br0 peth0
ifconfig peth0 down
```

2.1) At WISP mode, enable ipv6 passthru as follows:

```
ifconfig pwlan0 up
brctl addif br0 pwlan0
echo "1" > /proc/custom_Passthru_wlan
```

Note: for WISP mode, in order to use ipv6 passthru, nat2.5 should also be enabled, otherwise

ipv6 passthru will fail. Enable nat2.5 as follows:

```
ifconfig wlan0 down
iwpriv wlan0 set_mib nat25_disable=0
ifconfig wlan0 up
```

2.2) At WISP mode, disable ipv6 passthru as follows:

```
echo "0" > /proc/custom_Passthru_wlan
brctl delif br0 pwlan0
ifconfig pwlan0 down
```

## 7.15 Please explain “Support multi-vlan in bridge/wisp mode” at kernel?

Answer:

Menu “Support multi-vlan in bridge/wisp mode” is at linux menuconfig as follows:

```
Device Drivers -->
[*] Network device support -->
[*] Options for Realtek SoC -->
 Config for Layered Driver Features -->
 [*] Support multi-vlan in bridge/wisp mode
```

“Support multi-vlan in bridge/wisp mode” is implemented by macro CONFIG\_RTL\_IVL\_SUPPORT.

When AP is changed to bridge/WISP mode, the original wan port is added into br0.

If CONFIG\_RTL\_IVL\_SUPPORT is enabled, the original wan port vlan id is set to 8 and the other lan port vlan id is set to 9, so the traffic between the lan port and the original wan port will be software forwarded via CPU.

If CONFIG\_RTL\_IVL\_SUPPORT is disabled, the original wan port and the other lan port is set to the same vlan id, so the traffic between the lan port and the original wan port will be hardware forwarded.

## 7.16 How to add/update flash MIB entry at the web-server?

Answer:

### 1) Introduction of flash MIB

The flash MIB is used for system configuration.

The flash MIB can be accessed via webpage or flash tool. The flash tool is a method to manage the flash MIB via console.

In flash layout, there are three MIB settings: HW Settings, Default Settings and Current Settings.

#### -- HW Settings

The HW settings are the parameter settings of the hardware, such as RF power index. Usually, they are set via MP tools.

#### -- Default Settings

The Default settings are backup settings. When you click "Reset to Default" button on webpage,

or input console command "flash default-sw", the Current settings will be replaced by the Default settings.

The Program-default settings are hardcode settings at the function writeDefault() in flash.c.

#### -- Current Settings

The Current settings are current user settings.

### 2) Steps to add flash MIB entry

Step 1, in users/goahead-2.1.1/LINUX/apmib.h or users/boa/apmib/apmib.h if Boa web server is selected, create new MIB define. The MIB ID defined by users should be 16000~32767 (1~15999 for Realtek) .

Ex:

```
#define MIB_WLAN_11N_ONOFF_TKIP 660
```

Step 2, in users/goahead-2.1.1/LINUX/mibdef.h or users/boa/apmib/mibdef.h if Boa web server is selected, create new member in "config\_setting" or in "config\_wlan\_setting". There are flags MIB\_IMPORT for "config\_setting" and MIB\_CONFIG\_WLAN\_SETTING\_IMPORT for "config\_wlan\_setting". Please insert the new MIB entry to the proper table.

Ex:

```
#ifdef MIB_IMPORT
...
MIBDEF (unsigned char, wlan11nOnOffTKIP, , WLAN_11N_ONOFF_TKIP,
BYTE_T, APMIB_T, 0, 0)
...
#endif /*MIB_IMPORT*/
```

Step 3, in users/goahead-2.1.1/LINUX/flash.c or users/boa/utils/flash.c if Boa web server is selected, add program-default value for new MIB entry at function writeDefault().

Step 4, in users/goahead-2.1.1/LINUX/upmib.h or users/boa/apmib/upmib.h if Boa web server is selected, upgrade MIB VERSION value if step 5 exist.

Ex:

```
{MIB_MIB_VER, "MIB_VER", "2"},
```

Step 5, in users/goahead-2.1.1/LINUX/upmib.h or users/boa/apmib/upmib.h if Boa web server is selected, set new MIB default value in UPMIB\_T new\_mib[] before {0} if needed.

Ex:

```
{MIB_WLAN_11N_ONOFF_TKIP, "WLAN_11N_ONOFF_TKIP", "1"}
```

### 3) Steps to update the current setting of flash MIB entry

Step 1, in users/goahead-2.1.1/LINUX/upmib.h or users/boa/apmib/upmib.h if Boa web server is selected, upgrade MIB VERSION value if step 2 exist.

Ex:

```
{MIB_MIB_VER, "MIB_VER", "3"},
```

Step 2, in users/goahead-2.1.1/LINUX/upmib.h or users/boa/apmib/upmib.h if Boa web server is

selected, update the current setting of the existed MIB entry in UPMIB\_T update\_mib[] before {0} if needed.

Ex:

```
{MIB_IP_ADDR, "IP_ADDR", "192.168.1.1"}
```

4) Note

\* Any of following conditions may cause the HW settings or Default settings become program-default settings.

- a. The Setting signature in flash is not equal to setting signature in firmware.
- b. The Setting version in flash is not equal to setting version in firmware.

\* Any of following conditions may cause the Current settings replaced by Default settings.

- a. The Setting signature in flash is not equal to setting signature in firmware.
- b. The Setting version in flash is not equal to setting version in firmware.

\* The id in struct upmib is the same as the defination in apmib.h. The name in struct upmib is the same as the defination in mibdef.h

\* The new MIB value will be "0" or "" if you don't change the MIB version, and the specific MIB value won't be updated.

**\* Users can delete the unnecessary MIB in UPMIB\_T update\_mib[] and UPMIB\_T new\_mib[] at the next firmware version. If the MIB version is changed, All the MIB in the two struct will be upgrade when upload firmware.**

## 7.17 How to modify flash MIB settings using the configure file uploaded via webpage?

Answer:

The configure files are stored at users/goahead-2.1.1/LINUX or users/boa/defconfig if Boa web server is selected which are named after config-\*.txt. For example, config-gw-96c.txt is the configure file for 96C+92C demo board. At the end of make image, the configure file will be converted into config-\*.dat (ex: config-gw-96c.dat) which can be uploaded via webpage.

According to section 7.16, in flash layout, there are three MIB settings: HW Settings, Default Settings and Current Settings.

Similarly, there are HW Settings, Default Settings and Current Settings in the configure file.

HW Settings are named after HW\_\*, such as: HW\_NIC0\_ADDR.

Default Settings are named after DEF\_\*, such as: DEF\_IP\_ADDR.

The rest MIB entries are Current Settings.

We can modify any of the flash MIB setting in the configure file, then rebuild image and the configure file config-\*.dat will be generated.

After the configure file config-\*.dat is uploaded via webpage, all the flash MIB settings will be replaced by the configure file.

## 7.18 How to modify webpages at the web-server?

Answer:

The steps to modify webpages at the web-server as follows:

Step 1, in users/goahead-2.1.1/web-gw, or users/boa/html if Boa web server is selected.

- a. Add new page.
- b. Add new link at code.asp or code.htm if Boa web server is selected
- c. Create new page as \*.asp or \*.htm if Boa web server is selected

Step 2, in users/goahead-2.1.1/web-gw /\*.asp or users/boa/html/\*.htm if Boa web server is selected, dynamic values are generated and MUST be within "<% getInfo('value name'); %>".

And in users/goahead-2.1.1/LINUX/fmget.c or users/boa/src/fmget.c if Boa web server is selected, add the return value by apmib\_get() in "getInfo" function.

Step 3, in users/goahead-2.1.1/web-gw /\*.asp or in users/boa/html/\*.htm if Boa web server is selected, add form action which post the settings of webpages to c code.

Ex: in users/goahead-2.1.1/web-gw/tcpipwan.asp, add source code as follows,

```
<form action=/goform/formWanTcpipSetup method=POST name="tcpip">
...
</form>
```

Boa example: in users/boa/html/tcpipwan.htm, add source code as follows,

```
<form action=/boafrm/formWanTcpipSetup method=POST name="tcpip">
...
</form>
```

And in users/goahead-2.1.1/LINUX/fm\*.c, find or add new goform function (ex: formWanTcpipSetup) to handle the settings post from webpage. If Boa web server is selected, in users/boa/src/fm\*.c, find or add new boafrm function (ex: formWanTcpipSetup) to handle the settings post from webpage. Usually, save the new setting to flash by apmib\_set().

Step 5, re-build the image.

Note: the webpages will be decompressed in /var/webs when system boots up.

## 7.19 How to build image which can run in ICE?

Answer:

The steps to build image which can run in ICE as follows.

Step 1, defined CONFIG\_USING\_JTAG in linux-2.6.30/drivers/char/rtl\_gpio.c(**for kernel 2.6**) or linux-3.10/drivers/char/rtl\_gpio.c(**for kernel 3.10**)

Step 2, remove CONFIG\_RTL865X\_BICOLOR\_LED code in rtl\_nic.c

```
#if 0
 #if defined(CONFIG_RTL865X_BICOLOR_LED)
 #ifdef BICOLOR_LED_VENDOR_BXXX
 REG32(LEDCR) |= (1 << 19); // 5 ledmode set to 1 for bi-color LED
 REG32(PABCNR) &= ~0x001f0000; /* set port port b-4/3/2/1/0 to gpio */
 REG32(PABDIR) |= 0x001f0000; /* set port port b-4/3/2/1/0 gpio direction-output */
 #else
 //8650B demo board default: Bi-color 5 LED
 WRITE_MEM32(LEDCR, READ_MEM32(LEDCR) | 0x01180000); // bi-color LED
 #endif
 /* config LED mode */
 WRITE_MEM32(SWTAA, PORT5_PHY_CONTROL);
 WRITE_MEM32(TCR0, 0x000002C2); //8651 demo board default: 15 LED boards
 WRITE_MEM32(SWTACR, CMD_FORCE | ACTION_START); // force add
 #else /* CONFIG_RTL865X_BICOLOR_LED */
 /* config LED mode */
 WRITE_MEM32(LEDCR, 0x00000000); // 15 LED
 WRITE_MEM32(SWTAA, PORT5_PHY_CONTROL);
 WRITE_MEM32(TCR0, 0x000002C7); //8651 demo board default: 15 LED boards
 WRITE_MEM32(SWTACR, CMD_FORCE | ACTION_START); // force add
 #endif /* CONFIG_RTL865X_BICOLOR_LED */
#endif
```

Step 3, modify ARCH\_CPU\_SLEEP default value to 'N' in boards/rtl8196c/config.in

```
config ARCH_CPU_SLEEP
```

```
 bool
```

```
 default n
```

make menuconfig and load default value.

Step 4, close watchdog at linux kernel menuconfig

System Configuration --->

Enable watchdog timer support []

Step 5, re-build image.

Step 6, load linux-2.6.30/vmlinux(**for kernel 2.6**) or linux-3.10/ vmlinux (**for kernel 3.10**) into ICE and run it.

## 7.20 The image size and free memory info for RTL8198 + RTL8192C SDK.

Answer:

RTL8198 + RTL8192C demo board v610 run SDK v2.5 image (2011-06-07), the test result as follows.

Configure \ Test entry	Config.dat(B)	fw.bin (KB)	MemFree (KB)
Default	9197	2006	16420
Enable samba	9252	2900	13736
Enable DLNA	9252	2949	15056

SDK v2.5 boot loader size

Configure	Boot. Bin(B)
96c	22006
96c_92d	22006
98	24158

## 7.21 The summary of the HW feature for ICs till now.

Answer:

96C: HW multicast/Dump Switch

98: HW NAT/HW Multicast

96CT/98T: HW NAT/Multicast/Qos & All Nat

97: HW Multicast/Qos

## **7.22 Default values of RTL8196C SDK multicast and IPv6.**

Answer:

Default values of RTL8196C SDK multicast and IPv6 as follows.

```
/proc/br_mldsnoop: 1
/proc/br_mldquery: 1
/proc/br_igmpsnoop: 1
/proc/br_igmpquery: 1
Ipv6: disable
```

## **7.23 How to configure to process IGMP reserve address?**

Answer:

1) proc UI

```
/proc/br_igmpDb
```

2) command format

```
echo "opCode devicename ipversion groupAddr forwardPortMask" > /proc/br_igmpDb
(i) opCode: add/del
(ii) deviceName:eth*/br0/all
(iii) ipversion: ipv4 or ipv6
(iv) groupAddr: special multicast address
(v) forwardPortMask: user specified forward port mask
```

3) examples

example 1:

```
flood special address 224.1.2.3 if no client join
echo "add all ipv4 224.1.2.3 0xffffffff" > /proc/br_igmpDb
```

example 2:

```
block special address 224.1.2.3 if no client join
echo "add all ipv4 224.1.2.3 0" > /proc/br_igmpDb
```

example 3:

```
del special address 224.1.2.3 record
echo "del all ipv4 224.1.2.3 0" > /proc/br_igmpDb
```

example 4:

```
default to flood ipv4 unknown multicast
echo "add all ipv4 0.0.0.0 0xffffffff" > /proc/br_igmpDb
```

example 5:

```
default to block ipv4 unknown multicast
echo "add all ipv4 0.0.0.0 0x0" > /proc/br_igmpDb
```

example 6:

```
default to flood ipv6 unknown multicast
```

```
echo "add all ipv6 0x0-0-0-0 0xFFFFFFFF" > /proc/br_igmpDb
```

example 7:

default to block ipv6 unknown multicast

```
echo "add all ipv6 0x0-0-0-0 0x0" > /proc/br_igmpDb
```

Notice ipv6 address must use the following format:

0xAAAAAAA-BBBBBBBB-CCCCCCC-DDDDDDDD

#### 4) Note

If user doesn't specify the special multicast address or the default unknown multicast forwarding rule, the default process is as follows:

239.255.255.250(upnp) is to be flooded,

225.1.1.1(voip phone) is to be flooded,

default to block unknown ipv4 multicast,

default to flood unknown ipv6 multicast.

## 7.24 How to modify flash offset of webpages/rootfs/kernel etc via linux menuconfig?

Answer:

```
make linux_menuconfig //To configure linux menuconfig
```

One example as follows:

System Configuration --->

(0x200000) Size of Flash //customized, should be <= real flash size

(0x6000) Hardware setting offset in flash. //customized, should not overwrite default settings in flash

(0x8000) Default setting offset in flash. //customized, should not overwrite current settings in flash

(0xC000) Current setting offset in flash. //customized, should not overwrite webpage image in flash

(0x10000) webpages image offset in flash. //customized, should not overwrite linux image in flash

(0x30000) linux image offset in flash. //customized, should not overwrite root image in flash

(0x100000) root image offset in flash. //customized, should not oversize the end of the flash

```

System Configuration
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys.
Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help,
</> for Search. Legend: [*] built-in [] excluded <M> module < > module capable

^(-)
[] Domain name query support
[] USB3G support
[] Http File server support
 *** Support two spi flash ***
[] two spi flash support
 *** Flash size 2M or 4M, default 2M ***
(0x200000) Size of Flash
 *** Hardware setting offset,should be 4K alignment ***
(0x6000) Hardware setting offset in flash.
 *** Default setting offset,should be 4K alignment. ***
 *** size of default and current setting should be same. ***
(0x8000) Default setting offset in flash.
 *** Current setting offset,should be 4K alignment. ***
(0xC000) Current setting offset in flash.
 *** Webpage image offset,should be 4K alignment. ***
 *** size of web page is normally about 100K. ***
(0x10000) webpages image offset in flash.
 *** Linux image offset,should be 4K alignment. ***
 *** this offset MUST between 0x10000~0x40000. ***
(0x30000) linux image offset in flash.
 *** Root image offset,should be 64K alignment. ***
(0x100000) root image offset in flash.
(3) Kernel Stack Order Configuration
 *** Build rootfs options ***
File system to mount root (squash fs) --->

```

<Select> < Exit > < Help >

## 7.25 For QoS using htb, something needs to be noticed when set tc rules.

Answer:

For QoS using htb to set “Guaranteed minimum bandwidth” rules, the sum of all children rate (except the default child rate) should be less than its parent rate.

One example is as follows:

```

tc qdisc del dev eth1 root
iptables -F -t mangle
tc qdisc add dev eth1 root handle 2:0 htb default 2 r2q 64
tc class add dev eth1 parent 2:0 classid 2:1 htb rate 10000kbit ceil 10000kbit quantum 30000
tc class add dev eth1 parent 2:1 classid 2:2 htb rate 1kbit ceil 10000kbit prio 256 quantum 30000
tc qdisc add dev eth1 parent 2:2 handle 102: sfq perturb 10
iptables -A PREROUTING -t mangle -m iprange --src-range 192.168.1.88-192.168.1.89 -j MARK
--set-mark 13
tc class add dev eth1 parent 2:1 classid 2:13 htb rate 5000kbit ceil 10000kbit prio 2 quantum 30000
tc qdisc add dev eth1 parent 2:13 handle 113: sfq perturb 10
tc filter add dev eth1 parent 2:0 protocol ip prio 100 handle 13 fw classid 2:13
iptables -A PREROUTING -t mangle -m iprange --src-range 192.168.1.60-192.168.1.60 -j MARK
--set-mark 14
tc class add dev eth1 parent 2:1 classid 2:14 htb rate 5000kbit ceil 10000kbit prio 2 quantum 30000
tc qdisc add dev eth1 parent 2:14 handle 114: sfq perturb 10
tc filter add dev eth1 parent 2:0 protocol ip prio 100 handle 14 fw classid 2:14

```

```
echo 1 > /proc/qos
```

For the example above, the sum of all children rate except the default child (2:13 rate 5000kbit + 2:14 rate 5000kbit) <= its parent rate (2:1 rate 10000kbit).

## 7.26 How to configure RTL8192D internal PA?

Answer:

There are two flash hardware MIB settings for the internal PA type of each interface on RTL8192D. Please check by the following commands:

```
flash get HW_WLAN0_11N_TRSWPAPE_C9
flash get HW_WLAN0_11N_TRSWPAPE_CC
flash get HW_WLAN1_11N_TRSWPAPE_C9
flash get HW_WLAN1_11N_TRSWPAPE_CC
```

The corresponding MIBs at wlan driver can be checked by *iwpriv* commands as follows:

```
iwpriv wlanX get_mib trsw_pape_C9
iwpriv wlanX get_mib trsw_pape_CC
```

The mapping of the flash MIBs and the PA types of RTL8192D is illustrated as the following table:

	HW_WLANX_11N_TRSWPAPE_C9	HW_WLANX_11N_TRSWPAPE_CC
Type 1: 5G TRSW + Ext. PA, 2G TR co-matched (Default)	0	0
Type 2 : 5G TRSW + Int. PA, 2G TR co-matched	170 (=0xAA)	160 (=0xA0)
Type 3: 5G SP3TSW + Int.PA, 2G TR co-matched	170 (=0xAA)	175 (=0xAF)
Type 4: 5G TRSW + Int PA, 2G TRSW + Int PA	0	160 (=0xA0)

If the wlan driver can not find the matching settings, the RTL8192D driver will be initialized as the Type 1: External PA.

## 7.27 How to configure RTL8192C External PA?

Answer:

RTL8192C use internal PA as the default setting. If the external PA is wanted to be used, the linux kernel config need to be configured as follows.

```
make linux_menuconfig //To configure linux menuconfig
```

Device Drivers --->

[\*] Network device support --->

    Wireless LAN --->

        [\*]     Enable external high power PA

        [\*]     Enable external LNA

```

----- Wireless LAN -----
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letter
includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?>
[*] built-in [] excluded <M> module < > module capable

[] Wireless LAN (pre-802.11)
[] Wireless LAN (IEEE 802.11)
[] Realtek 8190 wireless support
[] Realtek 8192SE wireless support
[*] RTL8192C/D 802.11b/g/n support
[*] Realtek 8192C wireless support
[] Realtek hostapd support
[] Enable PCIE power saving support
[*] Enable external high power PA
[*] Enable external LNA
[] Enable Antenna Diversity
[] Enable both of the 2 pcie slot for bi-8192C support
[] Realtek 8192D wireless support
[] Support Dual card:92C+92D
[*] Private skb buffer management
[*] Virtual AP Support
[*] Client Mode Support
[*] Repeater Mode support
[] Client Mode 802.1x Support
[*] WDS Support
[] Efuse Support
[] WAPI Support
[] Config File support
[] Wireless Tools v29 support

```

## 7.28 How to build image with firmware and config data (without HW setting)?

Answer:

First, build CONFIG-DATA with no HW setting. Modify the file :

users/goahead-2.1.1/LINUX/Makefile or users/boa/defconfig/Makefile if Boa web server is selected.

`./cvcfg-ap $(CONFIG_FILE) $(CONFIG_DAT)`

and

`./cvcfg-gw $(CONFIG_FILE) $(CONFIG_DAT)`

Add the argument `-no_hw`

`./cvcfg-ap -no_hw $(CONFIG_FILE) $(CONFIG_DAT)`

and

`./cvcfg-gw -no_hw $(CONFIG_FILE) $(CONFIG_DAT)`

Then, add config data to firmware image(fw.bin). Modify the file board/rtl8198(or

rtl8196c ,rtl8196ct etc. Depend on CPU)/Makefile

```
egrep "^CONFIG_APP_BOA=y" $(DIR_USERS)/.config > BOA.test; \
if [-s BOA.test] ; then \
 cp $(DIR_USERS)/boa/html/$(WEBIMAGE_BIN) \
$(DIR_IMAGE)/$(WEBIMAGE_BIN); \
 $(MGBIN) -c -o $(FW_BIN) $(ROOT_BIN) $(WEBPAGE_BIN) $(LINUX_BIN); \
 cd $(DIR_USERS)/boa/defconfig; \
 mv *.dat $(DIR_ROOT)/boards/rtl8198/image; \
 cd -; \
else \
 egrep "^CONFIG_APP_GOAHEAD=y" $(DIR_USERS)/.config > GOAHEAD.test; \
 if [-s GOAHEAD.test] ; then \
 cp $(DIR_USERS)/goahead-2.1.1/LINUX/$(WEBIMAGE_BIN) \
$(DIR_IMAGE)/$(WEBIMAGE_BIN); \
 $(MGBIN) -c -o $(FW_BIN) $(ROOT_BIN) $(WEBPAGE_BIN) $(LINUX_BIN); \
 cd $(DIR_USERS)/goahead-2.1.1/LINUX; \
 mv *.dat $(DIR_ROOT)/boards/rtl8198/image; \
 cd -; \
 fi; \
fi; \
```

Move the sentence back and add the argument *image*/\*.*dat*

```
egrep "^CONFIG_APP_BOA=y" $(DIR_USERS)/.config > BOA.test; \
if [-s BOA.test] ; then \
 cp $(DIR_USERS)/boa/html/$(WEBIMAGE_BIN) \
$(DIR_IMAGE)/$(WEBIMAGE_BIN); \
 cd $(DIR_USERS)/boa/defconfig; \
 mv *.dat $(DIR_ROOT)/boards/rtl8198/image; \
 cd -; \
 $(MGBIN) -c -o $(FW_BIN) image/*.dat $(ROOT_BIN) $(WEBPAGE_BIN) \
$(LINUX_BIN); \
else \
 egrep "^CONFIG_APP_GOAHEAD=y" $(DIR_USERS)/.config > GOAHEAD.test; \
 if [-s GOAHEAD.test] ; then \
 cp $(DIR_USERS)/goahead-2.1.1/LINUX/$(WEBIMAGE_BIN)
```

```

$(DIR_IMAGE)/$(WEBIMAGE_BIN); \
 cd $(DIR_USERS)/goahead-2.1.1/LINUX; \
 mv *.dat $(DIR_ROOT)/boards/rtl8198/image; \
 cd -; \
 $(MGBIN) -c -o $(FW_BIN) image/*.dat $(ROOT_BIN) $(WEBPAGE_BIN)
$(LINUX_BIN); \
 fi; \
fi; \

```

At last, rebuild, and the image with firmware and config data (without HW setting) fw.bin will be generated.

**Note:** Do not upload the image fw.bin with config data via tftp and console (section 3.3), upload it via webpage! (section 3.5)

## 7.29 What is the firewall capability and the relation with hardware NAT?

Answer:

Firewall is based on iptables rule except URL filter.

Iptables based firewall rules don't have any relationship with hardware NAT ( section 4.12 ) ; the URL filter is conflict to hardware NAT. The hardware NAT will be disabled when the URL filter is enabled from WEB UI.

All the iptables rules are support in kernel, and Port/IP/MAC/URL filter are supported in WEB UI.

We do the URL filter in fast path. So the fast path MUST be enabled when URL filter is enabled.

All packets will be trapped to CPU when hardware NAT is disabled. When hardware NAT is enabled, the packets will be trapped to CPU if it match a firewall rule, otherwise the packets will be transmit by hardware.

**Note:** hardware NAT is support by 98/98T/96CT, 96C don't support this feature.

## 7.30 How to enable/disable PPPoE passthru?

Answer:

This feature is depend on RTL\_CUSTOM\_PASSTHRU which enabled by default setting.

PPPoE pass through (CONFIG\_RTL\_CUSTOM\_PASSTHRU\_PPPOE) is disabled by default setting. Users can enable it by modify the following codes:

For kernel 2.6.30

- 1) linux-2.6.30/drivers/net/rtl819x/12Driver/rtl865x\_fdb.h
 

```
#define CONFIG_RTL_CUSTOM_PASSTHRU
//#define CONFIG_RTL_CUSTOM_PASSTHRU_PPPOE
→
#define CONFIG_RTL_CUSTOM_PASSTHRU
#define CONFIG_RTL_CUSTOM_PASSTHRU_PPPOE
```
- 2) linux-2.6.30/drivers/net/wireless/rtl8192cd/8192cd\_cfg.h

```

#define CONFIG_RTL_CUSTOM_PASSTHRU
#if defined(CONFIG_RTL_CUSTOM_PASSTHRU)
//#define CONFIG_RTL_CUSTOM_PASSTHRU_PPPOE
→
#define CONFIG_RTL_CUSTOM_PASSTHRU
#if defined(CONFIG_RTL_CUSTOM_PASSTHRU)
#define CONFIG_RTL_CUSTOM_PASSTHRU_PPPOE

```

#### For kernel 3.10

```

1) linux-3.10/drivers/net/rtl819x/l2Driver/rtl865x_fdb.h
#define CONFIG_RTL_CUSTOM_PASSTHRU
//#define CONFIG_RTL_CUSTOM_PASSTHRU_PPPOE
→
#define CONFIG_RTL_CUSTOM_PASSTHRU
#define CONFIG_RTL_CUSTOM_PASSTHRU_PPPOE
2) linux-3.10/drivers/net/wireless/rtl8192cd/8192cd_cfg.h
#define CONFIG_RTL_CUSTOM_PASSTHRU
#if defined(CONFIG_RTL_CUSTOM_PASSTHRU)
//#define CONFIG_RTL_CUSTOM_PASSTHRU_PPPOE
→
#define CONFIG_RTL_CUSTOM_PASSTHRU
#if defined(CONFIG_RTL_CUSTOM_PASSTHRU)
#define CONFIG_RTL_CUSTOM_PASSTHRU_PPPOE

```

SDKv2.5 don't support enable/disable PPPoE pass through from WEB UI now. Please do the following commands when init system.

1.1) At gateway mode, enable PPPoE passthru as follows:

```

ifconfig peth0 up
brctl addif br0 peth0
echo "2">>/proc/custom_Passthru or echo "3">>/proc/custom_Passthru

```

1.2) At gateway mode, disable PPPoE passthru as follows:

```

echo "0">>/proc/custom_Passthru or echo "1">>/proc/custom_Passthru
brctl delif br0 peth0
ifconfig peth0 down

```

2.1) At WISP mode, enable PPPoE passthru as follows:

```

ifconfig wlan0 up
brctl addif br0 wlan0
echo "2" >/proc/custom_Passthru_wlan or echo "3" >/proc/custom_Passthru_wlan

```

Note: for WISP mode, in order to use PPPoE passthru, nat2.5 should also be enabled, otherwise PPPoE passthru will fail. Enable nat2.5 as follows:

```

ifconfig wlan0 down
iwpriv wlan0 set_mib nat25_disable=0
ifconfig wlan0 up

```

2.2) At WISP mode, disable PPPoE passthru as follows:

```

echo "0" >/proc/custom_Passthru_wlan or echo "1" >/proc/custom_Passthru_wlan

```

```
brctl delif br0 pwlan0
ifconfig pwlan0 down
```

Note1: Please refer to sector 6.3 for more details about custom\_Passthru/custom\_Passthru\_wlan.

Note2: Please refer to sector 7.14 for more details about ipv6 pass through.

### 7.31 How to control dropping unknown multicast packet or not

- 1) User could have a choice to drop or forward unknown multicast to unicast packet by the following command in console.

example1:

```
/*to drop unknown multicast to unicast packet*/
```

```
iwpriv wlan0 set_mib mc2u_drop_unknown=1
```

example2:

```
/*to forward unknown multicast to unicast packet*/
```

```
iwpriv wlan0 set_mib mc2u_drop_unknown=0
```

- 2) User could have a choice to drop or forward unknown multicast packet

example1:

```
/*to drop unknown multicast packet*/
```

default to block ipv4 unknown multicast.

```
echo "add all ipv4 0.0.0.0 0x0" > /proc/br_igmpDb
```

example2:

```
/*to forward unknown multicast packet*/
```

```
echo "add all ipv4 0.0.0.0 0xFFFFFFFF" > /proc/br_igmpDb
```

### 7.32 How to use AP mib in another application

Answer:

If user's application (not goahead or boa) need to get/set mib through apmib library. The following steps should be followed.

#### 1) Build APMIB as dynamic library

Modify the Makefile in goahead/LINUX  
users/goahead-2.1.1/LINUX/Makefile

Set “APMIB\_SHARED = 1” in the Makefile.

for example:

```
#SNMPD support
ifeq ($(CONFIG_APP_SNMP), y)
APMIB_SHARED = 1
else
APMIB_SHARED = 0
endif
```

## 2) Linking APMIB dynamic library

In the application, the header file apmib.h should be included.

In the application's Makefile, the apmib.so should be linked.

## 3) Using APMIB dynamic library

Sample code for using the library.

```
int main()
{
 /*first need to init apmib*/
 apmib_init();

 /*then call apmib_get()*/
 Apmib_get(MIB_ID,value);

 /*if apmib_set() is needed*/
 Apmib_set(MIB_ID,value);

 /*if want to write back to flash*/
 apmib_update(CURRENT_SETTING);
}
```

## 7.33 How to Support rtl8211ds for RTL8197D High Power demo board

Answer:

With rtl8211ds, rtl8197d will have an Giga wan port. And rtl8211ds is mapping to phy port0.

*Make linux\_menuconfig*  
-> System Configuration  
-> Support rtl8211ds for rtl8197D

```

.config - Linux Kernel v2.6.30.9 Configuration
System Configuration

Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend:
[*] built-in [] excluded <M> module capable

System Type (97D+92C+92D GW) --->
[*] Enable watchdog timer support
[+] Support rtl8211ds for rtl8197D
[] Enable timer adjustment support
[] Webpages in rootfs support
[*] SPI flash support
[*] 819xD clock source at 40Mhz
[] Enable Flash Dual Bank support
[] USB3G support
[*] Speedup usb samba performance
[] Http File server support
*** Support two spi flash ***
[] two spi flash support
[*] Enable Flash Mapping
*** Flash size 2M or 4M, default 2M ***
(0x400000) Size of Flash
*** Hardware setting offset,should be 4K alignment ***
(0x6000) Hardware setting offset in flash.
*** Default setting offset,should be 4K alignment. ***
*** size of default and current setting should be same. ***
(0x8000) Default setting offset in flash.
*** Current setting offset,should be 4K alignment. ***
(0xC000) Current setting offset in flash.

v(+)

```

< Select > < Exit > < Help >

Note: 1) The function is only supported on SDK3.1 now;

2) If you are not sure How to support High Power, please refer to section 7.27 .

## 7.34 How to adjust the DRAM size

Answer:

The size of DRAM depends on bootcode. If 32M is chosen by bootloader, although the size of ram is 64M in fact, the size will be considered to be 32M by system. So we need to do configuration under bootloader.

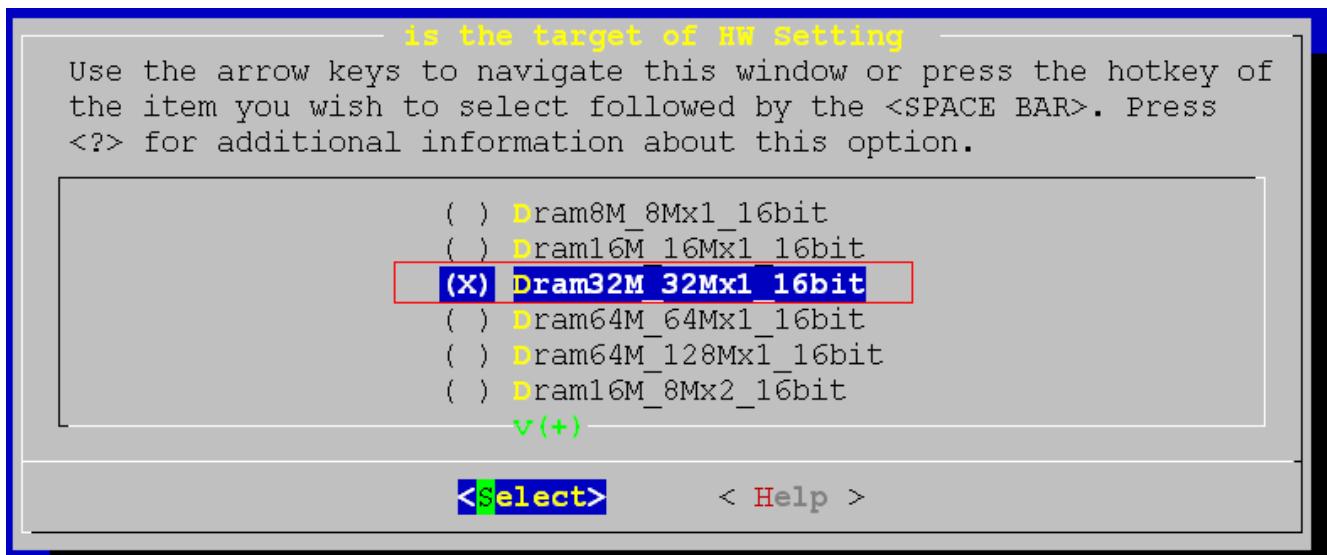
cd bootloader and

*make menuconfig*

->Target Platform Selection //enter

->(Dram32M\_32Mx1\_16bit) is the target of HW Setting //enter

->Dram32M\_32Mx1\_16bit //selected



## 7.35 How to add applications

Answer:

1. Add application to users directory

Take *udhcp* for example:

Copy *udhcp-0.9.9-pre* to users directory

2. Edit users/Kconfig

Add udhcpcd support:

*config APP\_UDHCPD*

*bool "udhcp"*

*help*

*An embedded DHCP client and server*

Show:

The screenshot shows a terminal window with the title "RLX Linux Configuration". The menu displays various options, with "udhcpd" highlighted in blue and enclosed in a red rectangle. Other options include pmc utils, python support, reload, routed, rtk\_cmd daemon, RTK tcp2dev utility, rtk voip, E8C (NEW), RTL8686 IPC User Utility, samba support, scripts, simple config support, snmp, squashfs 4.0, squashfs 4.2, and tr069.

When selected, users/.config have line:

*CONFIG\_APP\_UDHCPD=y*

### 3. Edit users/Makefile

Add udhcpd support, add line

*app\_\$(CONFIG\_APP\_UDHCPD) += udhcp-0.9.9-pre*

*udhcp-0.9.9-pre* is the directory name, and *CONFIG\_APP\_UDHCPD* is defined in .config

### 4. Add romfs target to application's Makefile

Edit application's Makefile to fit Cross Compiling, and add romfs support to collect target binary files and config files, use ROMFSINST to install them to romfs.

*romfs:*

```

$(ROMFSINST) udhcpd /bin/udhcpd
mkdir -p $(DIR_ROMFS)/usr/share
mkdir -p $(DIR_ROMFS)/usr/share/udhcpc
$(ROMFSINST) -s /var/udhcpc /etc/
rm $(DIR_ROMFS)/usr/share/udhcpc/*.*deconfig
$(ROMFSINST) -s /var/udhcpc/br0.deconfig /usr/share/udhcpc/br0.deconfig
$(ROMFSINST) -s /var/udhcpc/eth0.deconfig /usr/share/udhcpc/eth0.deconfig
$(ROMFSINST) -s /var/udhcpc/eth1.deconfig /usr/share/udhcpc/eth1.deconfig

```

```

$(ROMFSINST) -s /var/udhcpc/wlan0.deconfig /usr/share/udhcpc/wlan0.deconfig
$(ROMFSINST) -s /var/udhcpc/wlan1.deconfig /usr/share/udhcpc/wlan1.deconfig
$(ROMFSINST) -s /var/udhcpc/wlan0-vxd.deconfig
/usr/share/udhcpc/wlan0-vxd.deconfig
$(ROMFSINST) -s /var/udhcpc/wlan1-vxd.deconfig
/usr/share/udhcpc/wlan1-vxd.deconfig
$(ROMFSINST) -s udhcpd /bin/udhcpc
$(ROMFSINST) -s /var/udhcpd /etc

```

5. Make users\_menuconfig, select the application, and build image.

At last run the application on board and hope it works.

## 7.36 How to Enable JFFS2 in the linux and user space for storage partition

Answer:

Step 1, add the partitions for /dev/mtd2 in the linux-2.6.30/drivers/mtd/maps/rtl819x\_flash.c.

Please add the item to the mtd\_partition array , "rtl8196\_parts1".

Adjust the CONFIG\_JFFS2\_SIZE to fit the capacity of the flash.

For example:

```

static struct mtd_partition rtl8196_parts1[] = {
{
 name: "boot+cfg+linux",
 size: (CONFIG_RTL_ROOT_IMAGE_OFFSET-0),
 offset: 0x00000000,
},//mtd0
{
 name: "root fs",
 size: (WINDOW_SIZE - CONFIG_RTL_ROOT_IMAGE_OFFSET-
CONFIG_JFFS2_SIZE),
 offset: (CONFIG_RTL_ROOT_IMAGE_OFFSET),
}, //mtd1
//adding the mtd2
{
 name: "jffs2 ", //for JFFS2 use
 size: (CONFIG_JFFS2_SIZE),
 offset: (WINDOW_SIZE - CONFIG_JFFS2_SIZE),
} //mtd2
}

```

Step 2, please enable the following configurations.

*make linux\_menuconfig*

->File systems --->

```
[*] Journalling Flash File System v2 (JFFS2) support
 └── Miscellaneous filesystems
 Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted
 letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes
 features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*]
 built-in [] excluded <M> module < > module capable

 --- Miscellaneous filesystems
 [] ADFS file system support (EXPERIMENTAL)
 [] Amiga FFS file system support (EXPERIMENTAL)
 [] Apple Macintosh file system support (EXPERIMENTAL)
 [] Apple Extended HFS file system support
 [] BeOS file system (BeFS) support (read only) (EXPERIMENTAL)
 [] BFS file system support (EXPERIMENTAL)
 [] EFS file system support (read only) (EXPERIMENTAL)
 [*] Journalling Flash File System v2 (JFFS2) support
 (0) JFFS2 debugging verbosity (0 = quiet, 2 = noisy) (NEW)
 [*] JFFS2 write-buffering support (NEW)
 [] Verify JFFS2 write-buffer reads (NEW)
 [] JFFS2 summary support (EXPERIMENTAL) (NEW)
 [] JFFS2 XATTR support (EXPERIMENTAL) (NEW)
 v (+)

 <Select> < Exit > < Help >
```

Step 3, mount JFFS2 partition on the target

Before accessing JFFS2 file system, you must input the following command on the target as:

*mount -t jffs2 /dev/mtd2 /TARGET\_DST\_FOLDER.*

## 7.37 What are the special settings for 11AC?

Answer:

The most settings are the same as 11N. The followings are wlan mib for 11AC:

1. band  
bit 6 for 11AC. For 11AC+11N+11A, band = 76
2. use40M  
2 for 80M mode. For 11AC 80M mode, use40M = 2
3. Tx power settings, including:  
pwrdiff\_20BW1S\_OFDM1T\_A  
pwrdiff\_40BW2S\_20BW2S\_A  
pwrdiff\_5G\_20BW1S\_OFDM1T\_A  
pwrdiff\_5G\_40BW2S\_20BW2S\_A  
pwrdiff\_5G\_80BW1S\_160BW1S\_A  
pwrdiff\_5G\_80BW2S\_160BW2S\_A

pwrdiff\_20BW1S\_OFDM1T\_B  
pwrdiff\_40BW2S\_20BW2S\_B  
pwrdiff\_5G\_20BW1S\_OFDM1T\_B  
pwrdiff\_5G\_40BW2S\_20BW2S\_B  
pwrdiff\_5G\_80BW1S\_160BW1S\_B  
pwrdiff\_5G\_80BW2S\_160BW2S\_B

Please refer to MP document for detail of Tx power calibration.

#### 4. supportedvht

Set Tx/Rx MCS map of VHT capability element which carried in beacon, probe response, association request, and association response.

0xffffa: Support 2SS MCS0~9, 1SS MCS0~9

0xffff5: Support 2SS MCS0~8, 1SS MCS0~8

0xffffe: Support 1SS MCS0~9, 2SS not support

0xffffc: Support 1SS MCS0~7, 2SS not support

#### 5. vht\_txmap

Set VHT Tx rate map for rate adaptive algorithm

0xfffff: Support 2SS MCS9~0, 1SS MCS9~0

0x7ffdः: Support 2SS MCS8~0, 1SS MCS8~0

0x3fcff: Support 2SS MCS7~0, 1SS MCS7~0

0xfcff: Support 2SS MCS5~0, 1SS MCS7~0

The following are some new flash settings for 11AC for Tx power calibration:

TX\_POWER\_DIFF\_20BW1S\_OFDM1T\_A  
TX\_POWER\_DIFF\_40BW2S\_20BW2S\_A  
TX\_POWER\_DIFF\_5G\_20BW1S\_OFDM1T\_A  
TX\_POWER\_DIFF\_5G\_40BW2S\_20BW2S\_A  
TX\_POWER\_DIFF\_5G\_80BW1S\_160BW1S\_A  
TX\_POWER\_DIFF\_5G\_80BW2S\_160BW2S\_A  
TX\_POWER\_DIFF\_20BW1S\_OFDM1T\_B  
TX\_POWER\_DIFF\_40BW2S\_20BW2S\_B  
TX\_POWER\_DIFF\_5G\_20BW1S\_OFDM1T\_B  
TX\_POWER\_DIFF\_5G\_40BW2S\_20BW2S\_B  
TX\_POWER\_DIFF\_5G\_80BW1S\_160BW1S\_B  
TX\_POWER\_DIFF\_5G\_80BW2S\_160BW2S\_B

Please refer to MP document for detail of Tx power calibration.

## 7.38 How to use Band Edge Limit Tables

Band Edge Limit tables are used to control the amount of increased (even decreased) TPI (**Tx Power Index**) for these specific channels.

- **Band Edge Limit Table Format for 92c/92d**

In \rtl8192cd\data\ or \rtl8192cd\data\_92d , there are three files for Band Edge Limit Tables.  
Ex. In \rtl8192cd\data\:

TXPWR_LMT_92c_1.txt	Region Domain = FCC
TXPWR_LMT_92c_2.txt	Region Domain = CE
TXPWR_LMT_92c.txt	Others

The Format of Band Edge Limit tables is like this

```
//=====
// Table 2: FCC, OFDM (6M,9M,12M,18M,24M,36M,48M,54M)
// start from here
table 2
1 15.5 15// band edge
11 13.5 15// band edge

// 5G band 1 & 2
36 16 15// band edge
```

Channel	Limit	Target
1	15.5	15
11	13.5	15
36	16	15

## ● Band Edge Limit Table Format for 8812

The Band Edge Limit Tables for 8812 chip are in \rtl8192cd\data\_8812\TXPWR\_LMT\_8812\_new.txt

In TXPWR\_LMT\_8812\_new.txt, there are 13 Band Edge Limit Tables.

The Format of Band Edge Limit tables is like this

```
//
//Table 13: ======
//
5G, 80M, 2T, VHT, //(MCS0~MCS9)
START
#3# FCC ETSI MKK
//5G Band 1
CH42 14 15 15
//5G Band 2
CH58 13 15 15
//5G Band 3
CH106 14 15 15
CH122 16 15 15
//5G Band 4
CH155 17 15 NA
END
```

Channel	RegDomain 1	RegDomain 2	RegDomain 3
42	14	15	15
58	13	15	15
106	14	15	15
122	16	15	15
155	17	15	NA

PS 1: NA means NA is non-release channel

PS 2: Power Limit Value is real dBm.

PS 2. The power limit value is defined by special restrictions for wireless operation in some countries.

## ● How to Add a Band Edge Limit Table for New Regulatory Domain (92c / 92d)

If chip is 92c, add a new Band Edge Limit in \rtl8192cd\data, format of file name is TXPWR\_LMT\_92c\_##.txt where ## is TX Power Limit Table Index (txpwr\_lmt\_index)

Ex. Add TXPWR\_LMT\_92c\_3.txt for CCC (China) domain (txpwr\_lmt\_index=3)

Add TXPWR\_LMT\_92c\_4.txt for KCC (Korea) domain (txpwr\_lmt\_index=4)

.....

If chip is 92c, add a new Band Edge Limit in \rtl8192cd\data\_92d, format of file name is TXPWR\_LMT\_92d\_##.txt where ## is TX Power Limit Table Index

Ex. Add TXPWR\_LMT\_92d\_3.txt for CCC (China) domain (txpwr\_lmt\_index=3)

Add TXPWR\_LMT\_92d\_4.txt for KCC (Korea) domain (txpwr\_lmt\_index=4)

.....

## ● How to Add a Band Edge Limit Table for New Regulatory Domain (8812)

Add new power limit to data\_8812/TXPWR\_LMT\_8812\_new.txt in each table

For example, in table 13:

```
//
//Table 13: ======
//
5G, 80M, 2T, VHT, //(MCS0~MCS9)
START
#4# FCC ETSI MKK NEW
//5G Band 1
CH42 14 15 15 16
//5G Band 2
CH58 13 15 15 16
//5G Band 3
CH106 14 15 15 16
CH122 16 15 15 15
//5G Band 4
CH155 17 15 NA 16
```

```
END
```

In this example,

FCC Domain (txpwr\_lmt\_index=1)  
ETSI Domain (txpwr\_lmt\_index=2)  
MKK Domain (txpwr\_lmt\_index=3)  
NEW Domain (txpwr\_lmt\_index=4)

## ● How to assign Band Edge Limit Table for Driver

As the examples in previous two sections:

When Regulatory Domain is KCC in wlan1 (92c / 92d), then executing the following command:

```
iwpriv wlan1 set_mib txpwr_lmt_index=4
```

When Regulatory Domain is ETSI in wlan0 (8812), then executing the following command:

```
iwpriv wlan0 set_mib txpwr_lmt_index=2
```

Note that if there are two wlan interfaces, the values of txpwr\_lmt\_index MIB in the two interfaces all need to be configured.

There are two cases for the txpwr\_lmt\_index settings

Case 1: user does not set the value of txpwr\_lmt\_index (Default value, i.e. txpwr\_lmt\_index = 0),

IF regdomain = 1 (FCC), FCC TX Power Limit will be applied.

IF regdomain = 3 (ETSI), ETSI TX Power Limit will be applied.

IF regdomain = others, Default TX Power Limit will be applied

Case 2: user set the value of txpwr\_lmt\_index (txpwr\_lmt\_index > 0)

the specified TX Power Limit will be applied.

Note:

1. RegionDomain 1 in data\_8812/TXPWR\_LMT\_8812\_new.txt is default TX Power Limit for 8812.

data\TXPWR\_LMT\_92c.txt is default TX Power Limit for 92c.

## 7.39 RTL8367RB LED setting

Realtek demo boards (include 8197D+8367RB and 8881AB+8367RB) use two LEDs for each Ethernet port. If customer uses only one LED for each port, please define RTL8367RB\_USE\_ONE\_LED\_PER\_PORT in \linux-2.6.30\drivers\net\rtl819x\rtl8367r\rtk\_api.c.

**Table. RTL8367RB LED Control Register (0x1B03)**

Reg.bit	Name	Description	Mode	Default
n.15	Reserved			
n.14	cfg_set_data_led_en	control cfg_set_data_led enable/disable control 1:enable 0:disable	W/R	0B
n.13-n.12	cfg_set_data_led	write to these two bits will change LED_CFG set cfg_data_ledx_cfg as the following table set led0 led1 led2 00 0010 0011 0100 01 0110 0111 1000 10 0001 0110 1001 11 1000 0110 0111	W/R	00B
n.11-n.8	cfg_data_led2_cfg	led2 configuration (see led0 configuration).	W/R	100B
n.7-n.4	cfg_data_led1_cfg	led1 configuration (see led0 configuration).	W/R	011B
n.3-n.0	cfg_data_led0_cfg	led0 configuration 0000: LED_Off 0001: Dup/Col 0010: Link/Act, 0011: Spd1000 0100: Spd100, 0101: Spd10 0110: Spd1000/Act, 0111: Spd100/Act 1000: Spd10/Act 1001: Spd100(10)/Act 1010: LoopDetect, 1011: eee_cap/eee_lpi 1100: Spd1000(100)/Act, 1101: Spd1000(10)/Act 1110: Master, 1111: 1'b0/Act	W/R	010B

## 7.40 How to enable GPIOB5 as Reset Button for some RTL8196EU-MCM reference design

RTL8196E use GPIOA5 as reset button, but some reference design use GPIOB5 as reset button, this feature can be enabled by a configure option.

Menu “Enable GPIOB5 as Reset Button” is at linux menuconfig as follows:

Device Drivers -->

[\*] Network device support -->

[\*] Options for Realtek SoC -->

[\*] Enable GPIOB5 as Reset Button

## 7.41 How to enable SATA interface support on the RTL8198CS platform.

RTL8198CS has the SATA interface support and this feature can be enabled by a configure option.

Enable the SATA interface at linux menuconfig as follows:

Device Drivers --->

[\*] ATA/ATAPI/MFM/RLL support (DEPRECATED)

[\*] Serial ATA and Parallel ATA drivers --->

[\*] Platform AHCI SATA support

[\*] ATA SFF support (for legacy IDE and PATA)

## 7.42 How to use an gpio pin

Before using an gpio pin, four register need to be set.

1. set shared pin register

A certain pin may be versatile with different function, therefore, these shared pins need to be configured as gpio mode.

2. set gpio control register

The GPIO pins are shared with some peripheral pins, and the type of peripheral can affect the attributes of the shared pins. Set control register as gpio pin is needed.

3. set gpio direction register

Each GPIO pin may be configured as an input or output pin. Reset button is configured as input pin, LED pin is configured as output pin.

4. set gpio data register

For output pin(led pin as example), write this register will change the state of pin.

For input pin(button pin as example),this register value will change with the change of pin state.

Please refer to the datasheet for more information

## 7.43 How to build image without default config

There are some CPU+Wi-Fi IC don't have demo board and default configuration, users can change Wi-Fi according to there request.

Take 8198+8192E for example.

1. select 8198 and load default, select config kernel

The screenshot shows the RLX Linux Configuration interface. At the top, there is a header with usage instructions: "Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [\*] built-in [ ] excluded <M> module < > module capable". Below this, the main menu displays several options under the "rtl8198" target: "Selected Target (rtl8198) --->", "Selected Kernel (linux-2.6.30) --->", "Selected Busybox (busybox-1.13) --->", "Selected toolchain (rsdk-1.3.6-5281-EB-2.6.30-0.9.30) --->". Under "rtl8198", there are sub-options: "Selected Board Configuration (SPI flash + Squashfs) --->" and "IC Test Configuration --->". The "config components" section contains "Config kernel" (highlighted with a blue border), "Config users", "Config busybox", "Load default settings" (marked with an asterisk [\*]), and "Save default settings". At the bottom of the menu, there are two additional options: "Load an Alternate Configuration File" and "Save an Alternate Configuration File". At the very bottom of the window, there is a footer with navigation keys: "<Select> < Exit > < Help >".

2. enter linux kernel Wireless LAN menu and select as follows:

Device Drivers ---> //enter  
[\*]Network device support ---> //selected and enter  
    Wireless LAN           ---> //enter

```

Wireless LAN
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys.
Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help,
</> for Search. Legend: [*] built-in [] excluded <M> module < > module capable

[] Wireless LAN (pre-802.11)
[] Wireless LAN (IEEE 802.11)
[] Realtek 8190 wireless support
[] Realtek 8192SE wireless support
[*] RTL8192C/D 802.11b/g/n support
[*] Use PCIe slot 0 WiFi device
PCIe Slot 0 device (Realtek 8192EE wireless support) --->
[] PCIe slot 0 Enable external PA
[] PCIe slot 0 Enable external LNA
[*] PCIe slot 0 Enable Tx Beamforming (NEW)
[] PCIe slot 0 Enable Antenna Diversity (NEW)
[] Use PCIe slot 1 WiFi device
[] Realtek hostapd support
[*] Private skb buffer management
[*] Virtual AP Support
[*] Client Mode Support
[*] Repeater Mode support
[] multiple Repeater Mode support
[] multiple clone support (NEW)
[] Client Mode 802.1x Support
[*] Multiple AP profile Support
[*] WDS Support
[] Efuse Support
[] WAPI Support
[] Config File support
[] Wireless Tools v29 support
▼(+)

<Select> < Exit > < Help >

```

## 7.44 How to support LDPC function for 8192ER

We support LDPC function for 8192ER to get about 2dB gain for coverage or performance.

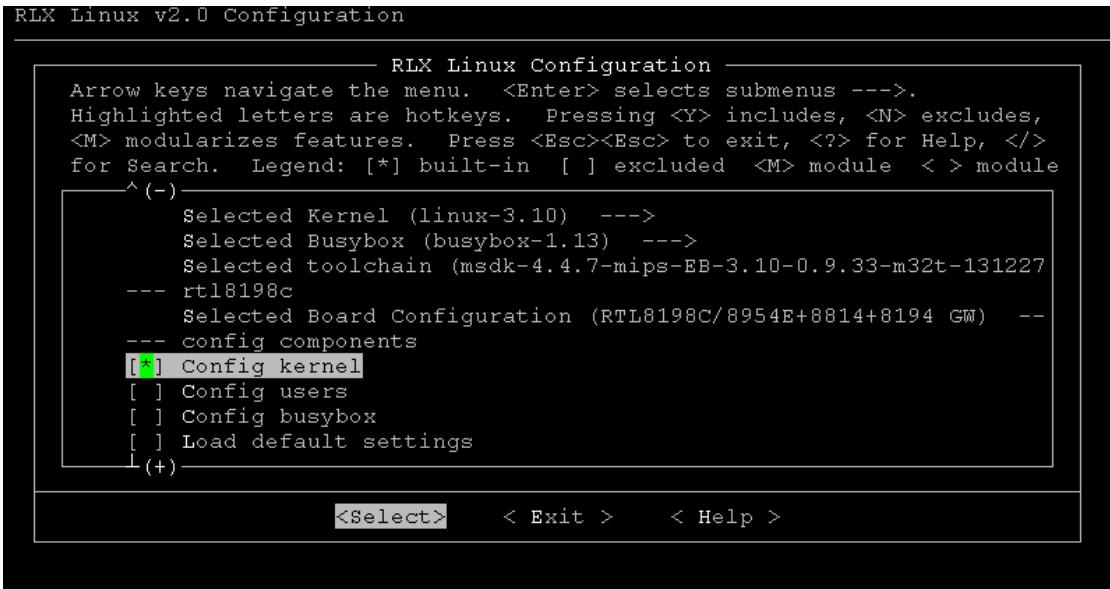
However we suggest to enable LDPC function for 8192ER on LDO board only. It is not suitable to enable LDPC function for 8192ER on SWR board for system stability.

To enable LDPC function for 8192ER, please follow the steps:

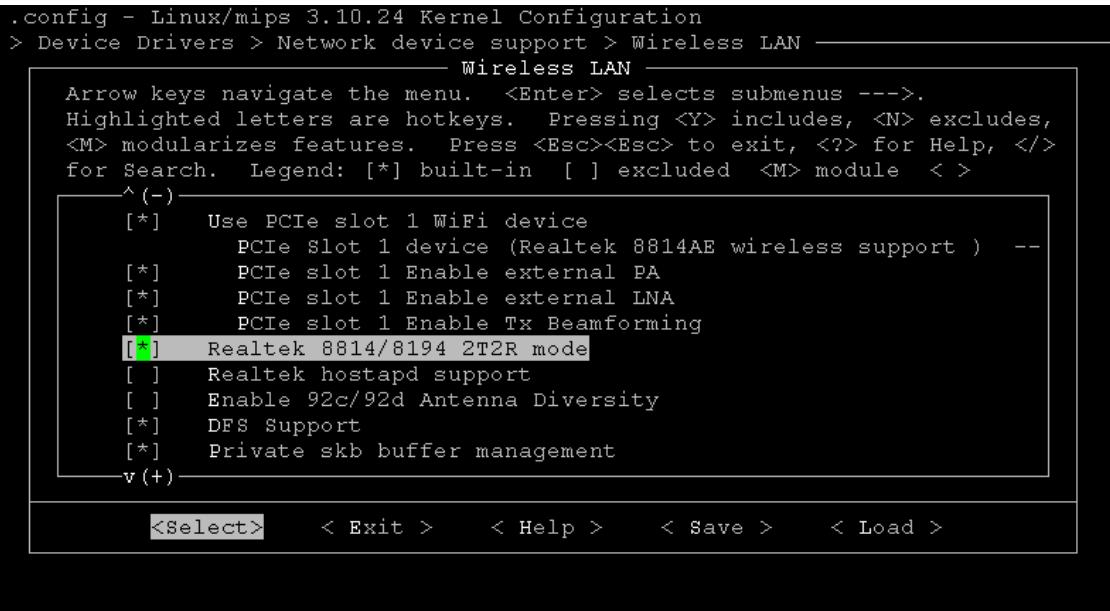
1. Find initWlan( ) in users/boa/utils/flash.c.
2. Find the code segment of getting MIB\_WLAN\_LDPC\_ENABLED to set to wlan mib.  
`apmib_get(MIB_WLAN_LDPC_ENABLED,(void *)&intVal);  
pmib->dot11nConfigEntry.dot11nLDPC = intVal;`
3. Add the following code after the above code segment.  
`sprintf((char *)buf2, "iwpriv %s set_mib ldpc_92e=%d", ifname, intVal);  
system((char *)buf2);`

## 7.45 How to configure 8814AR/8194AR to use 2T2R setting

- Configure 8814AR/8194AR to use 2T2R setting by Menuconfig
  - Step1: make menuconfig in SDK
  - Step2: select “RTL8198C/8954E+8814+8194 GW” option and load default settings



- Step3: make linux\_menuconfig and enter to Path > Device Drivers > Network device support > Wireless LAN
- Step4: enable “Realtek 8814/8194 2T2R mode” option as the following figure and save & exit



## 7.46 How to configure hw qos by rtk\_cmd

1. Enable the following config
  - a) CONFIG\_RTL\_HW\_QOS\_SUPPORT
 

```
make linux_menuconfig
->DeviceDrivers
```

->Network device support(NETDEVICES[=y])  
-> Options for Realtek SoC (RTL\_819X\_SWCORE [=y])  
[\*] Support HW Qos

b) CONFIG\_APP\_RTK\_CMD  
make users\_menuconfig  
[\*] rtk\_cmd daemon  
2. Rtk hw qos cmd

## 2.1 Function illustration

The customer could do “*rtk\_cmd qos*” to implement the configuration for hardware qos.

## 2.2 Introduction of Command

### 2.2.1 QUEUE\_NUMBER:

Set queue number of each port, input the following command in console :

*rtk\_cmd qos Queue\_Num PORT\_NUM1 QUEUE\_NUM1 PORT\_NUM2*

*QUEUE\_NUM2...*

Input para:

*PORT\_NUM1 QUEUE\_NUM1 PORT\_NUM2 QUEUE\_NUM2...*

Para *PORT\_NUM1,PORT\_NUM2* mean the port you want to set ;

Para *QUEUE\_NUM,QUEUE\_NUM2* mean the queue number you want to use.

e.g. if you want to use 2 queues of port0, and use 3queues of port1, then please input the command in console:(please follow the format in sample)

*rtk\_cmd qos Queue\_Num port0 2 port1 3*

If you want to check the queue number of each port, please input the command in console :

*rtk\_cmd qos SHOW QUEUE\_NUMBER*

### 2.2.2 PRIORITY\_DECISION:

ASIC support 5 kinds of priority policy: based on port, dscp, 802.1p, acl or nat. Please

choose one kind policy and input the value of decision. Each decision takes up 4 bits. e.g. if you want to set port 15, dscp 1 802.1p 2, acl 3, nat 4, please input the following command in console:

```
rtk_cmd qos Priority_Decision port 15 vlan 1 dscp 2 acl 3 nat 4
```

(Note that the value between port and 15, vlan and 1 need a blank。)

If you want to check the priority of policies, please input the following command in console :

```
rtk_cmd qos SHOW_PRIORITY_DECISION
```

### 2.2.3 PRIORITY\_ASSIGN:

Now, rtk\_cmd qos support 3 kinds policy: Port\_Based, Dscp\_Based, 802.1p\_Based。

#### 2.2.3.1 Port\_Based:

The priority is assigned by port. If you want to set system priority of port0= 3, system priority of port1=4, system priority of port2=5, please input the follow command in console:

```
rtk_cmd qos Priority_Assign Port_Based port0 3 port1 4 port2 5
```

#### 2.2.3.2 Dscp\_Based:

The priority is assigned by dscp. If you want to set system priority=1 while dscp=10, system priority=2 while dscp=61, then input the follow command in console:

```
rtk_cmd qos Priority_Assign Dscp_Based dscp10 1 dscp61 2
```

If you want to display the priority of each dscp, please input the command in console:

```
rtk_cmd qos SHOW_DSCP_BASED_PRI
```

#### 2.2.3.3 802.1p\_Based:

If you want to set system priority=2 while vlan priority=1, set system priority=3while vlan priority=2, then please input the following command in console:

```
rtk_cmd qos Priority_Assign Vlan_Based vlan1 2 vlan2 3
```

If you want to check the system priority correspond to each vlan priority, please input the following command in console:

```
rtk_cmd qos SHOW VLAN_BASED_PRI
```

#### 2.2.4 PRIORITY\_TO\_QID:

Before set the qid correspond to each system priority, please make sure that the queue number of each port has been set. If set priority 1 corresponding to qid 1, priority 2 corresponding to qid 2, please input the following command in console:

```
rtk_cmd qos Priority_to_Qid pri1 1 pri2 2
```

If you want to display the mapping relationship between system priority and qid, please input the following command in console:

```
rtk_cmd qos SHOW PRIORITY_TO_QID
```

#### 2.2.5 QUEUE\_TYPE:

Queue could be set to two types: STRICT or WEIGHTED.

##### 2.2.5.1 STRICT

If you want to set queue0, queue1 and queue2of port1 to STRICT type, please input the following command in console:

```
rtk_cmd qos Queue_Type STRICT port1 q0 q1 q2
```

If you want to check that which queues of port have been set to STRICT type, please input :

```
rtk_cmd qos SHOW QUEUE_TYPE_STRICT
```

##### 2.2.5.2 WEIGHTED

If you want to set queue0 of port1 to WEIGHTED type, and the weight=15; set queue1 to WEIGHTED type and weight=18, then please input the following command in console:

```
rtk_cmd qos Queue_Type WEIGHTED port1 q0 15 q1 18
```

If you want to check which queues of port have been set to WEIGHTED type, and what the weight values are, then please input:

```
rtk_cmd qos SHOW QUEUE_TYPE_WEIGHTED
```

#### 2.2.6 REMARK:

ASIC support 2 kinds of remark: VLAN remark and DSCP remark.

#### 2.2.6.1 VLAN REMARK

If you want to remark vlan priority=2 when port=0, system priority=1, please input in console:

```
rtk_cmd qos Remark VLAN port0 pri1 2
```

If you want to check the information of vlan remark, please input in console :

```
rtk_cmd qos SHOW VLAN REMARK
```

#### 2.2.6.2 DSCP REMARK:

If you want to remark dscp=61, when port=0, system priority=2, please input in console:

```
rtk_cmd qos Remark DSCP port0 pri2 61
```

If you want to check the information of dscp remark, please input in console:

```
rtk_cmd qos SHOW DSCP REMARK
```

#### 2.2.7 NOTE

When input the following value in console, please pay attention to the range respective:

- I. vlan priority: 3bit;
- II. system priority: 3bit;
- III. port: 3bit;
- IV. dscp value: 6bit;
- V. weight: 7bit;

### 7.47 How to fix samba “map network driver” disk size show 20M issue

When test samba performance, usually use samba disk address to “map network driver” to host pc.



In that case, map samba disk size will be root directory disk size.

For example:

Smb.conf

[mnt]

```
comment = Temporary file space
path = /tmp/usb/
```

...

[sda1]

```
comment = Temporary file space sda1
path = /tmp/usb/sda1
```

Map <\\192.168.1.254\mnt\sda1> to local disk, it will show directory "mnt" s' disk size, in dut system mount point is /tmp/usb, usually is 20M

Map <\\192.168.1.254\sda1> to local disk, it will show directory "sda1" s' disk size, in dut system mount point is /tmp/usb/sda1, usually is the real sda1 storage disk size.

In later(sdk 3.4.7) release sdk, usb storage device will be auto mount and the samba directory will be auto generated, just be care not map to [\\192.168.1.254\mnt\\\*](\\192.168.1.254\mnt\*) directory, and the issue will be avoid.