



Kernel 2.6 SDK User Guide

Realtek Kernel 2.6 SDK User Guide

Date: 2012/12/19

Version: 1.75

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

Kernel 2.6 SDK User Guide.....	1
Realtek Kernel 2.6 SDK User Guide.....	1
1. Kenel 2.6 SDK introduction	1
2. Compile Kenel 2.6 SDK.....	2
2.1. <i>Compile Environment.....</i>	2
2.2. <i>Compile bootloader.....</i>	2
2.3. <i>Compile rlxlinux.....</i>	3
2.4. <i>Compile rlxlinux detailed.....</i>	13
3. Kenel 2.6 SDK images upload	15
3.1 <i>Topology to upload image.....</i>	15
3.2 <i>Upload bootloader image</i>	15
3.3 <i>Upload rlxlinux image</i>	16
3.4 <i>Upload default setting data (Optional).....</i>	18
3.5 <i>Upload rlxlinux image via webpage (Optional)</i>	18
3.6 <i>Upload rlxlinux image with config file via webpage (Optional).....</i>	19
4. Features configure.....	21
4.1 <i>PCI support.....</i>	21
4.2 <i>USB support.....</i>	22
4.3 <i>Samba support.....</i>	24
4.4 <i>USB 3G support.....</i>	34
4.5 <i>WAPI support.....</i>	39
4.6 <i>Mesh support (not support now)</i>	41
4.7 <i>VLAN support.....</i>	41
4.8 <i>I2C UART support.....</i>	48
4.9 <i>DLNA support.....</i>	51
4.10 <i>Pocket AP support.....</i>	55
4.11 <i>Wireless client mode 802.1x support</i>	56
4.12 <i>Hardware NAT</i>	57
4.13 <i>Iptables rule to ACL rule translation support.....</i>	60
4.14 <i>Hardware QoS support.....</i>	61
4.15 <i>IPv6 support.....</i>	65
4.16 <i>64K/sector SPI flash support.....</i>	71
4.17 <i>SPI Flash Support.....</i>	73
4.18 <i>92C/D support.....</i>	76
4.19 <i>GDB server support.....</i>	82
4.20 <i>HTTP file server support.....</i>	83
4.21 <i>Hostapd support.....</i>	91

4.22	<i>MP support</i>	94
4.23	<i>AP mode support</i>	97
4.24	<i>Wireless configuration file support</i>	98
4.25	<i>WPS under wireless configuration file support</i>	99
4.26	<i>Domain name query support</i>	100
4.27	<i>rtk_voip support</i>	105
4.28	<i>Realtek Flash Dual Image support</i>	105
4.29	<i>IEEE 802.3az EEE (Energy Efficient Ethernet) support</i>	106
4.30	<i>IGMP/MLD support</i>	106
4.31	<i>SNMP support</i>	108
4.32	<i>UVC support</i>	109
4.33	<i>Telnetd and Login support in busybox</i>	111
4.34	<i>Dynamic Frequency Selection (DFS) support</i>	112
4.35	<i>Fastpath Filter</i>	115
4.36	<i>Fastpath SPI (stateful packet inspection) support</i>	119
4.37	<i>ASAL support</i>	120
4.38	<i>Romeperf support</i>	121
4.39	<i>Boa support</i>	124
4.40	<i>uWiFi support</i>	131
4.41	<i>ALL NAT</i>	137
4.42	<i>Proc filesystem for debug</i>	138
4.43	<i>TR069 Support</i>	141
4.44	<i>ALG Support</i>	141
4.45	<i>Cone Nat support</i>	142
5.	FLASH tools	145
5.1	<i>flash</i>	145
5.2	<i>cvcfg-gw</i>	145
5.3	<i>compweb</i>	146
5.4	<i>cvimg</i>	146
5.5	<i>mgbin</i>	146
6.	Proc file format	148
6.1	<i>br_igmpProxy</i>	148
6.2	<i>br_igmpsnop</i>	148
6.3	<i>custom_Passthru</i>	148
6.4	<i>enable_dos</i>	149
6.5	<i>fast_I2tp</i>	150
6.6	<i>fast_nat</i>	150
6.7	<i>fast_pptp</i>	151

6.8	<i>filter_table</i>	151
6.9	<i>gpio</i>	152
6.10	<i>pptp_conn_ck</i>	153
6.11	<i>qos</i>	153
6.12	<i>rf_switch</i>	154
6.13	<i>rtk_vlan_support</i>	154
6.14	<i>mib_vlan</i>	154
6.15	<i>sw_nat</i>	155
6.16	<i>hw_nat</i>	155
6.17	<i>br_wlanblock</i>	156
6.18	<i>br_igmpVersion</i>	157
6.19	<i>br_igmpquery</i>	157
6.20	<i>br_mCastFastFwd</i>	157
6.21	<i>br_mldquery</i>	158
6.22	<i>br_mldsnoop</i>	158
6.23	<i>eee</i>	159
6.24	<i>gc_overflow_timeout</i>	159
6.25	<i>load_default</i>	159
6.26	<i>reInitSwitchCore</i>	160
6.27	<i>br_mldVersion</i>	160
6.28	<i>jate</i>	160
6.29	<i>gpio_ctrl</i>	161
7. FAQ	163
7.1	<i>How to modify the mappings of LAN/WAN port?</i>	163
7.2	<i>How to change the baudrate of console?</i>	163
7.3	<i>How to use vlan priority option of iptables?</i>	163
7.4	<i>Relationship of virtual AP and root AP?</i>	163
7.5	<i>How to customize icon of lld2d?</i>	164
7.6	<i>How to rename the AP when shown on Windows?</i>	164
7.7	<i>How to add MAC address filter based on ether driver API?</i>	164
7.8	<i>How to adjust CPU speed?</i>	167
7.9	<i>How to shutdown power of Ethernet ports?</i>	167
7.10	<i>How to disable Ethernet PHY?</i>	168
7.11	<i>Which channels are supported by RTL8192D?</i>	168
7.12	<i>How to set speed and duplex of physical port via APIs?</i>	170
7.13	<i>How to dial up pppoe for test when wan interface is changed?</i>	170
7.14	<i>How to enable/disable ipv6 passthru?</i>	174
7.15	<i>Please explain “Support multi-vlan in bridge/wisp mode” at kernel?</i>	175

7.16	<i>How to add/update flash MIB entry at the web-server?</i>	175
7.17	<i>How to modify flash MIB settings using the configure file uploaded via webpage?</i>	177
7.18	<i>How to modify webpages at the web-server?.....</i>	178
7.19	<i>How to build image which can run in ICE?.....</i>	179
7.20	<i>The image size and free memory info for RTL8198 + RTL8192C SDK.....</i>	180
7.21	<i>The summary of the HW feature for ICs till now.</i>	180
7.22	<i>Default values of RTL8196C SDK multicast and IPv6.</i>	180
7.23	<i>How to configure to process IGMP reserve address?.....</i>	181
7.24	<i>How to modify flash offset of webpages/rootfs/kernel etc via linux menuconfig?.....</i>	182
7.25	<i>For QoS using htb, something need to be noticed when set tc rules.....</i>	183
7.26	<i>How to configure RTL8192D internal PA?.....</i>	184
7.27	<i>How to configure RTL8192C External PA?</i>	184
7.28	<i>How to build image with firmware and config data (without HW setting)?</i>	185
7.29	<i>What is the firewall capability and the relation with hardware NAT?.....</i>	187
7.30	<i>How to enable/disable PPPoE passthru?.....</i>	187
7.31	<i>How to control dropping unknown multicast packet or not.....</i>	188
7.32	<i>How to use AP mib in another application.....</i>	189
7.33	<i>How to Support rtl8211ds for RTL8197D High Power demo board</i>	190
7.34	<i>How to adjust the DRAM size.....</i>	191
7.35	<i>How to add applications</i>	191
7.36	<i>How to Enable JFFS2 in the linux and user space for storage partition</i>	193

1. Kenel 2.6 SDK introduction

Kenel 2.6 SDK is a Linux development platform customized to provide a unified platform for embedded systems based on RLX processor cores.

Kenel 2.6 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 and RTL8954C for Kenel 2.6 SDK while rtl819x-sdk-v2.x/rtl819x stores applications and linux kernel.

Directory bootcode rtl819xD deposit bootloader source code of RTL8196D, RTL8197D for Kenel 2.6 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.

2. Compile Kenel 2.6 SDK

We assume that Kenel 2.6 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

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*

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*

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

0) Appoint the compile toolchain.

One methods to appoint the compile toolchain as follows:

a. find the toolchain at .../rtl819x/toolchain.

rsdk-1.3.6-4181-EB-2.6.30-0.9.30 is for RTL8196C and RTL8196CT.

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 and RTL8197D.

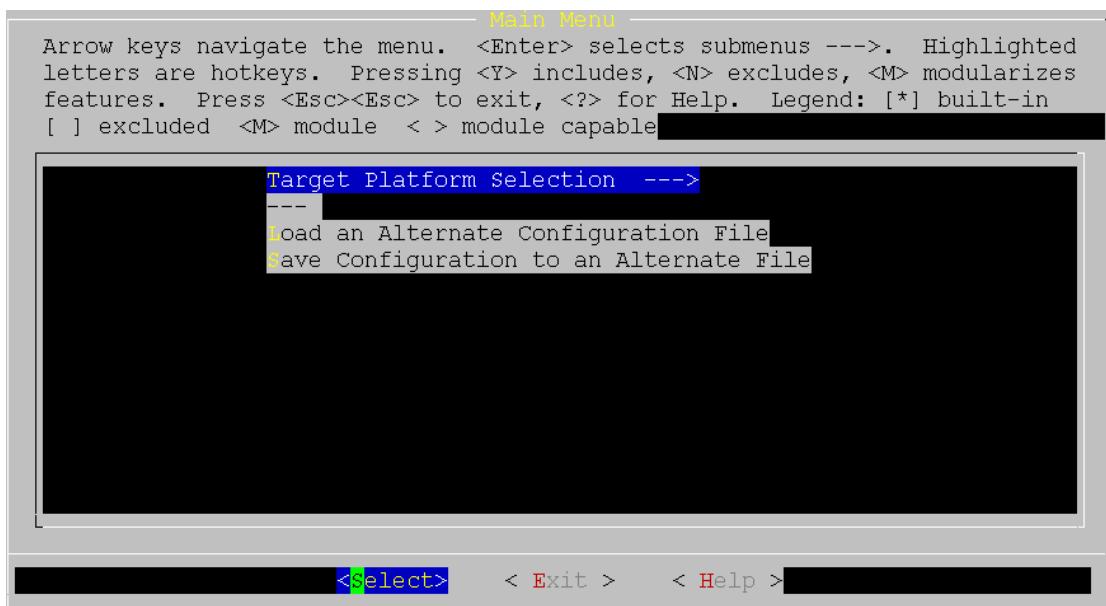
b. add the exact toolchain to the environment variant PATH.

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

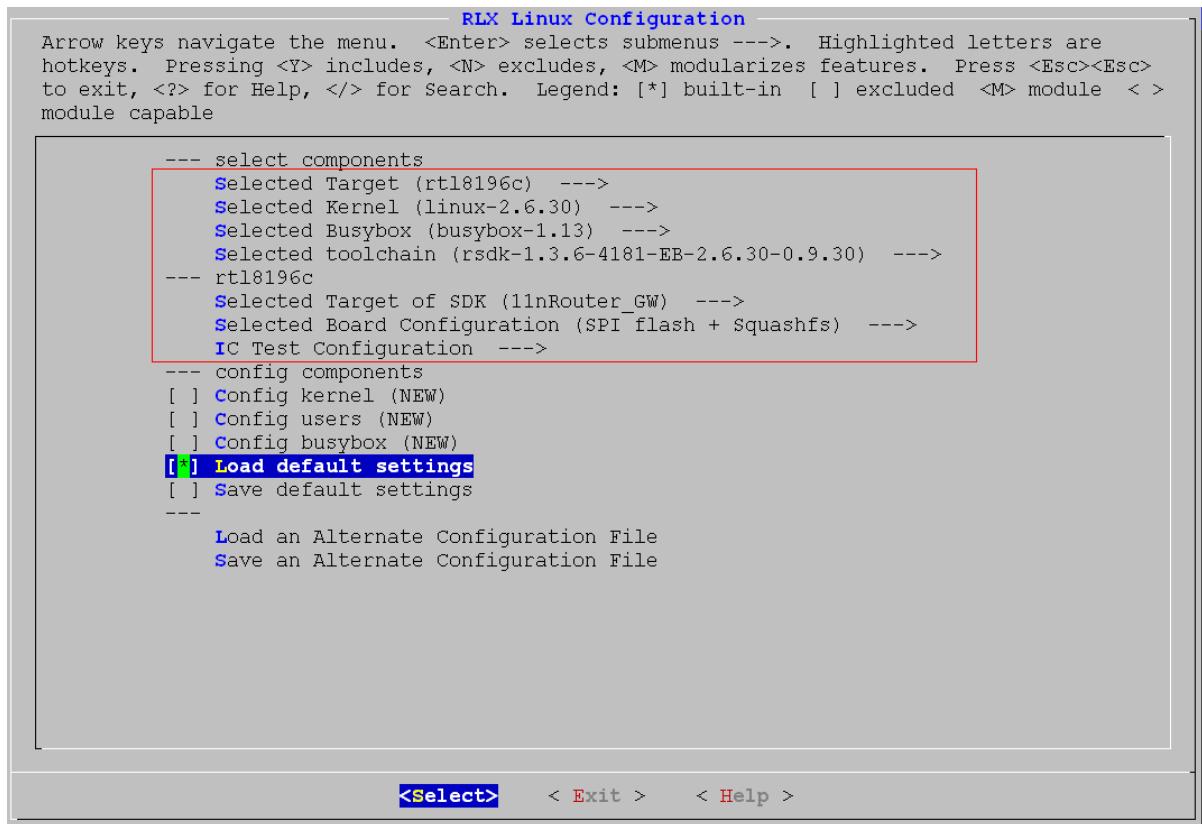
ii) Do some settings as follows:

'Selected Target' to choose rtl8196c, rtl8196ct, rtl8198, rtl8198t, rtl8954C and rtl819xD

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:



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 (rtl18198) --->
  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) --->
--- rtl18198
  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 (rtl189xxC) --->
  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) --->
--- rtl189xxC
  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
  < / >

<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]

[REDACTED]
<Select> < Exit > < Help > [REDACTED]
```

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]

[REDACTED]
<Select> < Exit > < Help > [REDACTED]
```

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 (rtl189xxD) --->
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) --->
--- rtl189xxD
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

----- RLX Linux Configuration -----
<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

----- Selected Board Configuration -----
<Select> < Help >

```

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

Note:

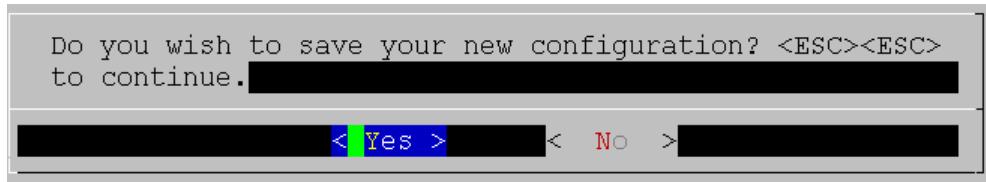
Above we just use configure applications and linux kernel in default way.

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’.



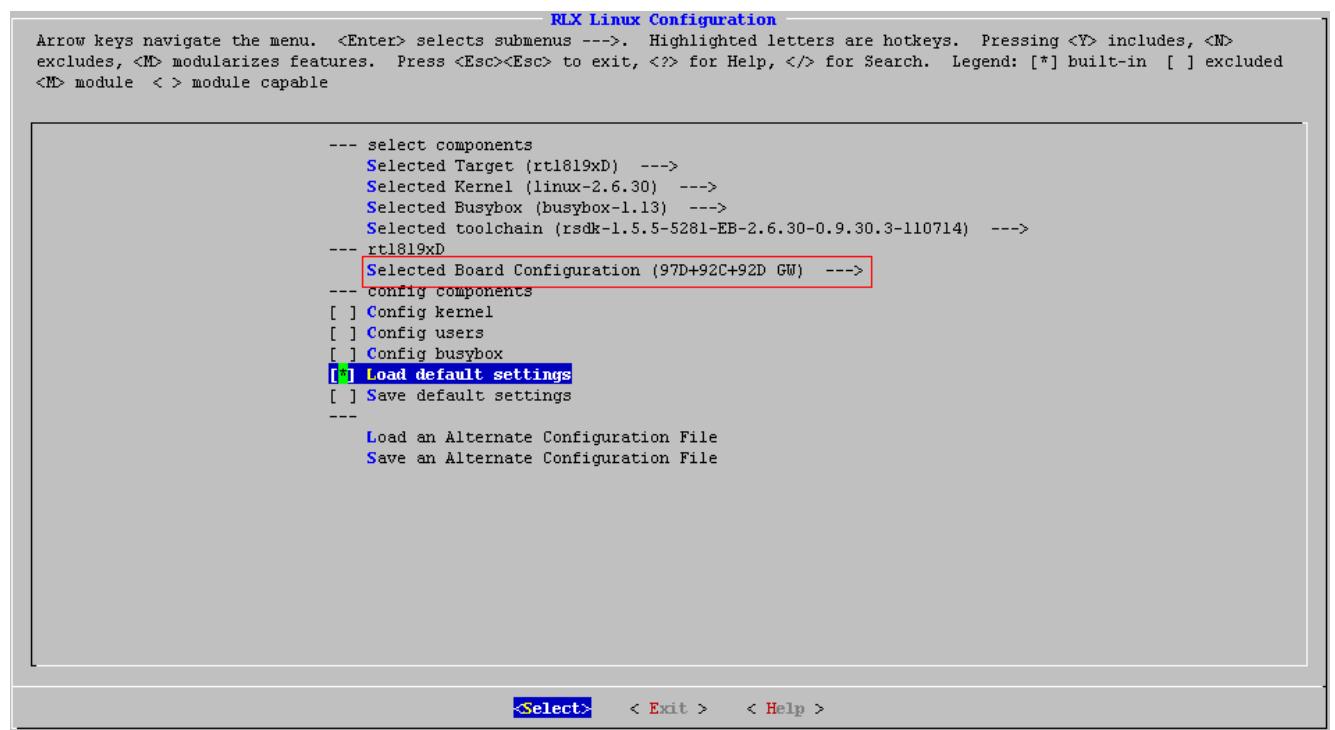
iv) Choose ‘<Yes>’ and click ‘Enter’

Step 3: compile rlxlinux.

Make

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

i) Load default for 97D+92C+92D



ii) Select 8812AE support in wifi 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 < > 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 8192C wireless support) --->
[ ] PCIe slot 0 Enable external high power PA
[*] Use PCIe slot 1 WiFi device
    PCIe Slot 1 device (Realtek 8812 wireless support) --->
[ ] PCIe slot 1 Enable external high power PA
[ ] Enable external LNA
[*] Private skb buffer management
[*] Virtual AP Support
[*] WDS Support
[ ] Config File support
[*] Realtek wps2.0 support
[*] Clock Source, Select is 40MHz, otherwise 25MHz

Select < Exit > < Help >

```

Example XII, setting for RTL8197D+8192C+8812AE +8367 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 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 >

```

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) module;
- (4) Compile linux dir (rtl819x/linux-2.6.30);
- (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 include 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.

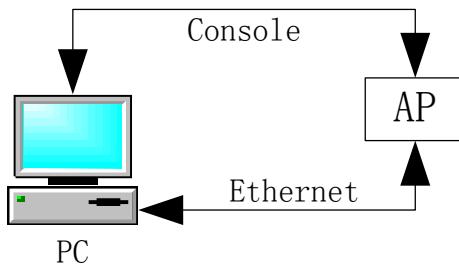
More info about these tools, please refer to [Chapter 5 FLASH tools](#).

3. Keneil 2.6 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)/btcode/ which is boot.bin.

Steps to upload bootloader image are as follows:

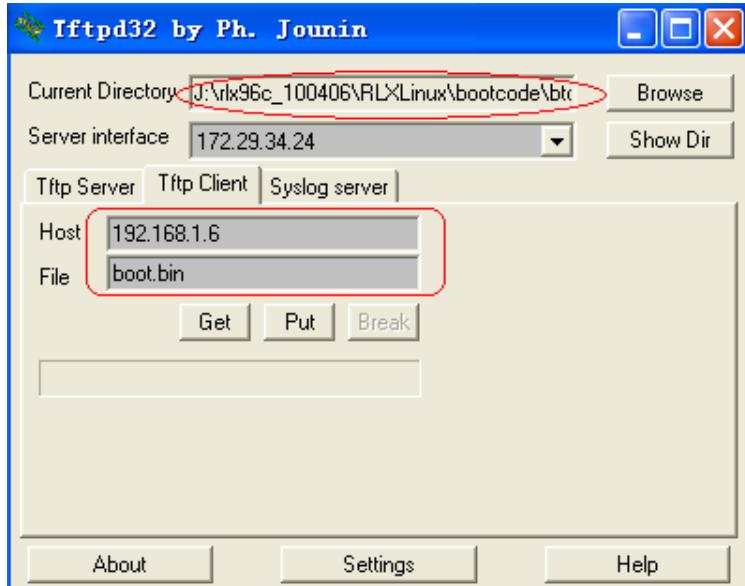
Step 1, reboot AP and click 'Esc' via console during AP booting, then AP enter bootloader environment. Output of AP's console is as follows:

```
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 tftp 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:

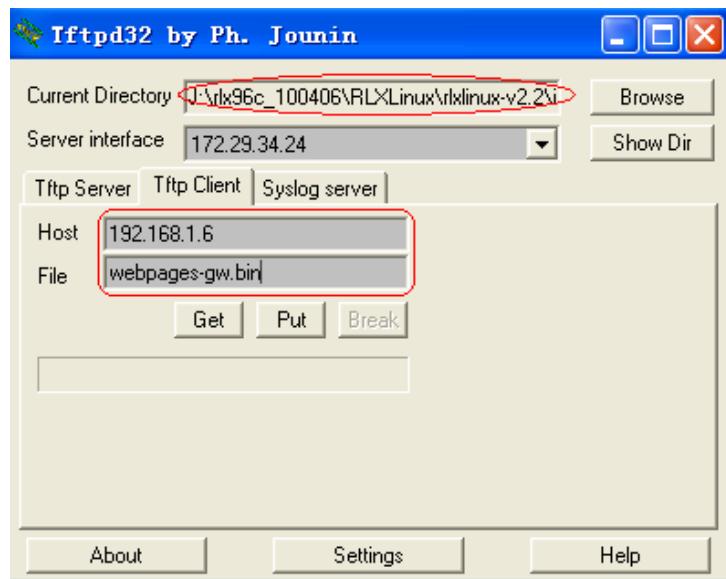
```
Booting...
=====
 SPI =====

---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.

Realtek WLAN AP Webserver - Windows Internet Explorer

http://192.168.1.254/home.asp

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

Setup Wizard

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.

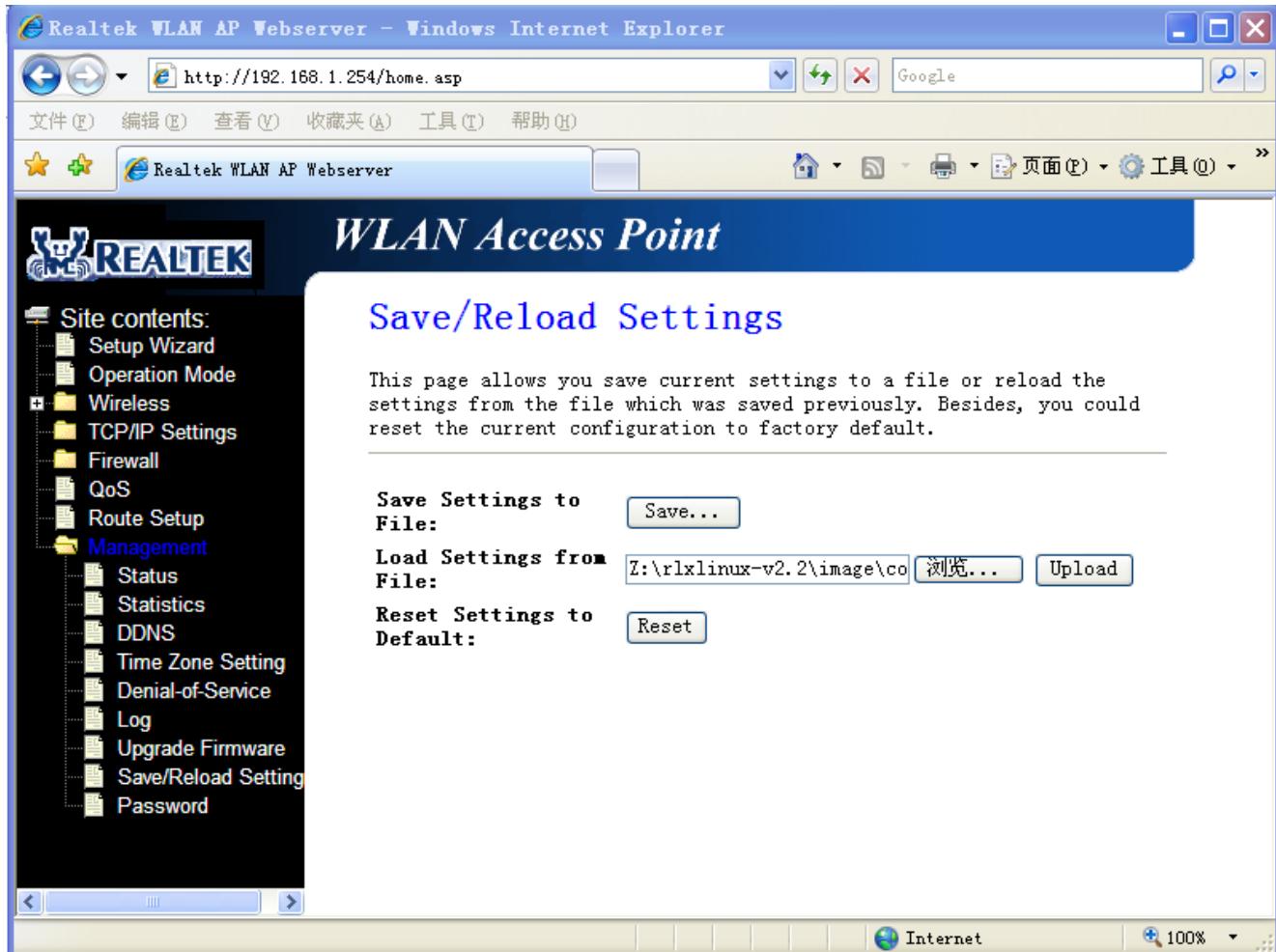
1. Setup Operation Mode
2. Choose your Time Zone
3. Setup LAN Interface
4. Setup WAN Time Zone
5. Select Wireless Band
6. Wireless LAN Setting
7. Wireless Security Setting

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.

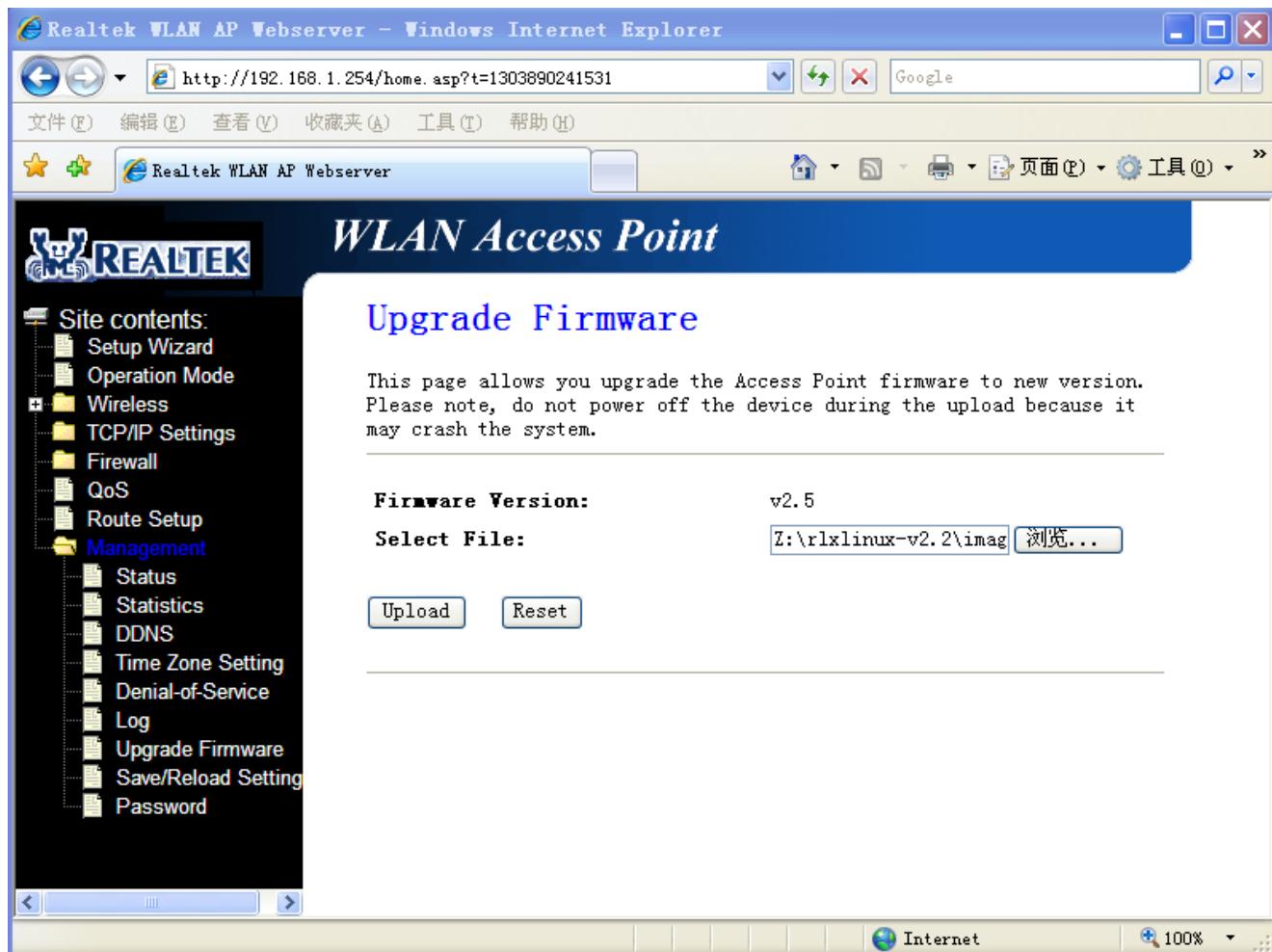
To upload default setting data, open AP webpage, select Management->Save/Reload Settings, browse and select config data file(ex: config-gw-98-92c-92d.dat) then upload.



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**

Site contents:

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

4. Features configure

4.1 PCI support

4.1.1 Linux kernel PCI not support

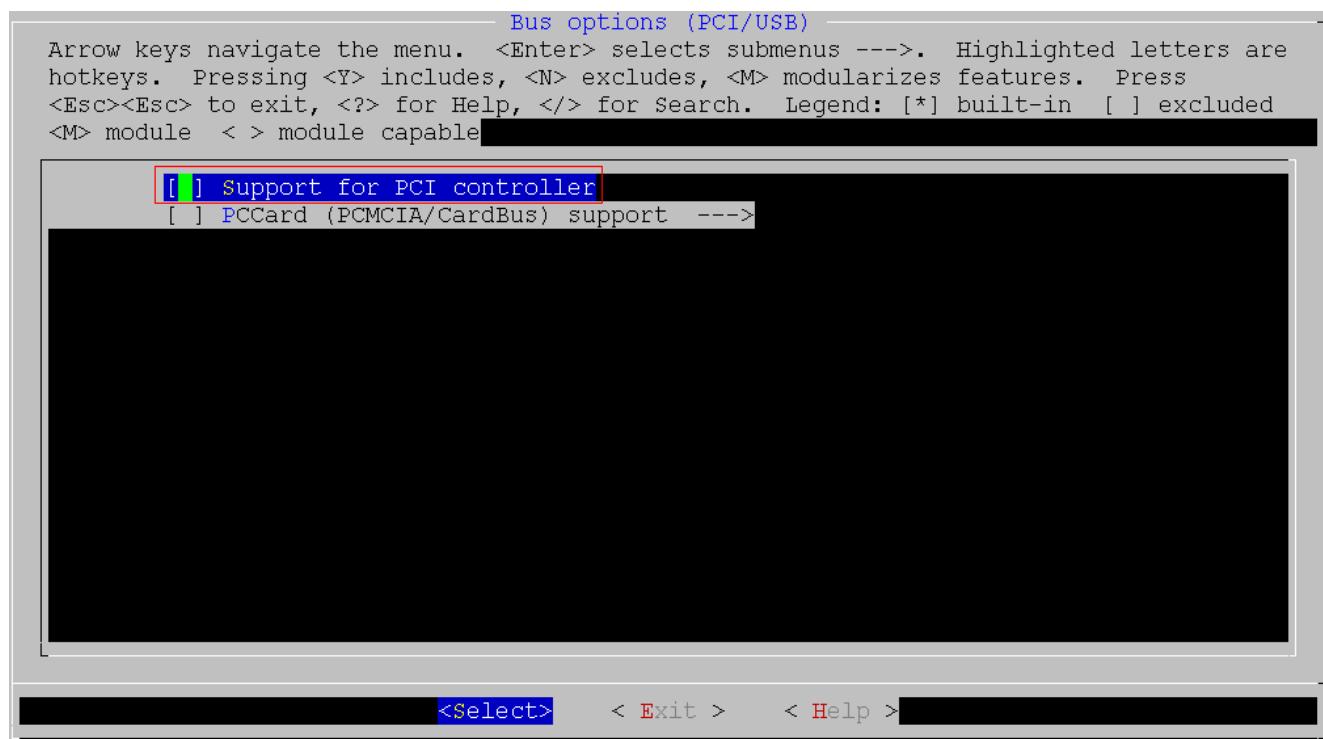
By kernel default settings, linux kernel PCI is not supported, because wireless driver use its own PCI driver. Linux kernel configure as follows.

```
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 driver of linux kernel, linux kernel configure as follows.

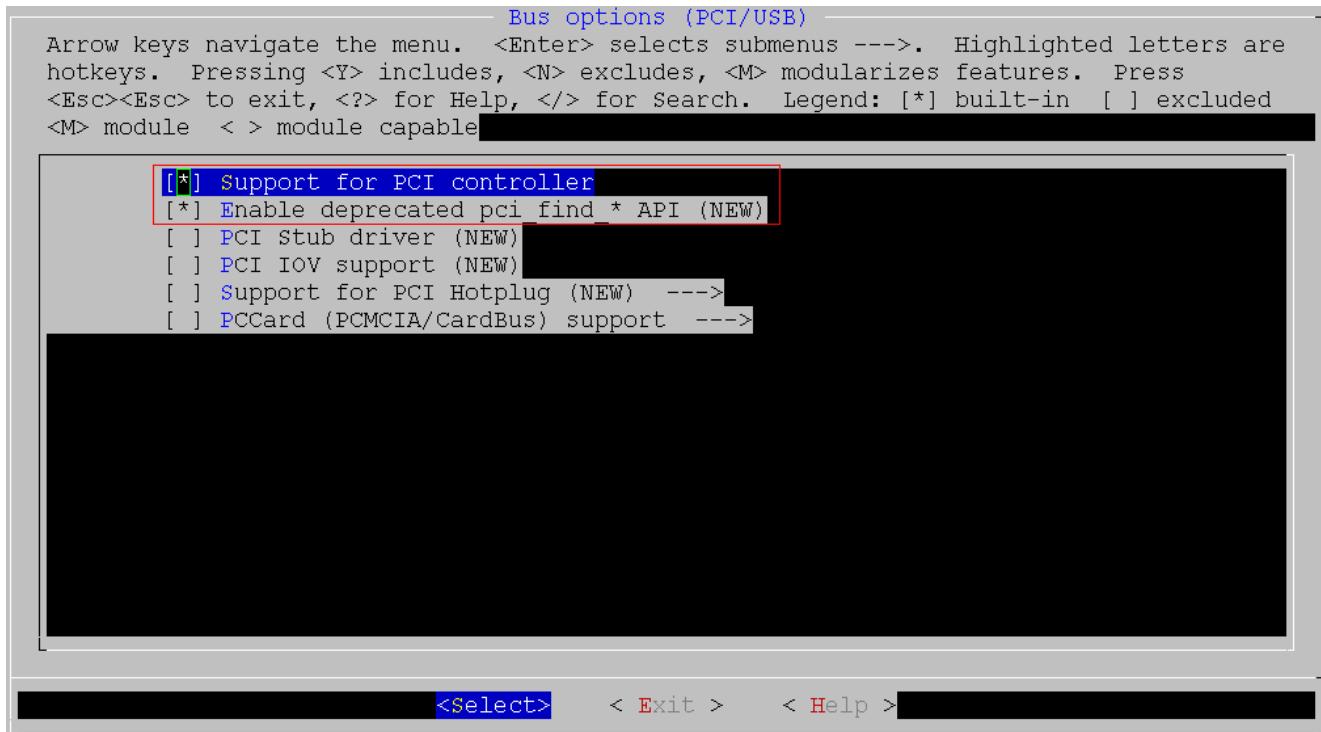
```
make linux_menuconfig      // To configure linux kernel settings
```

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.

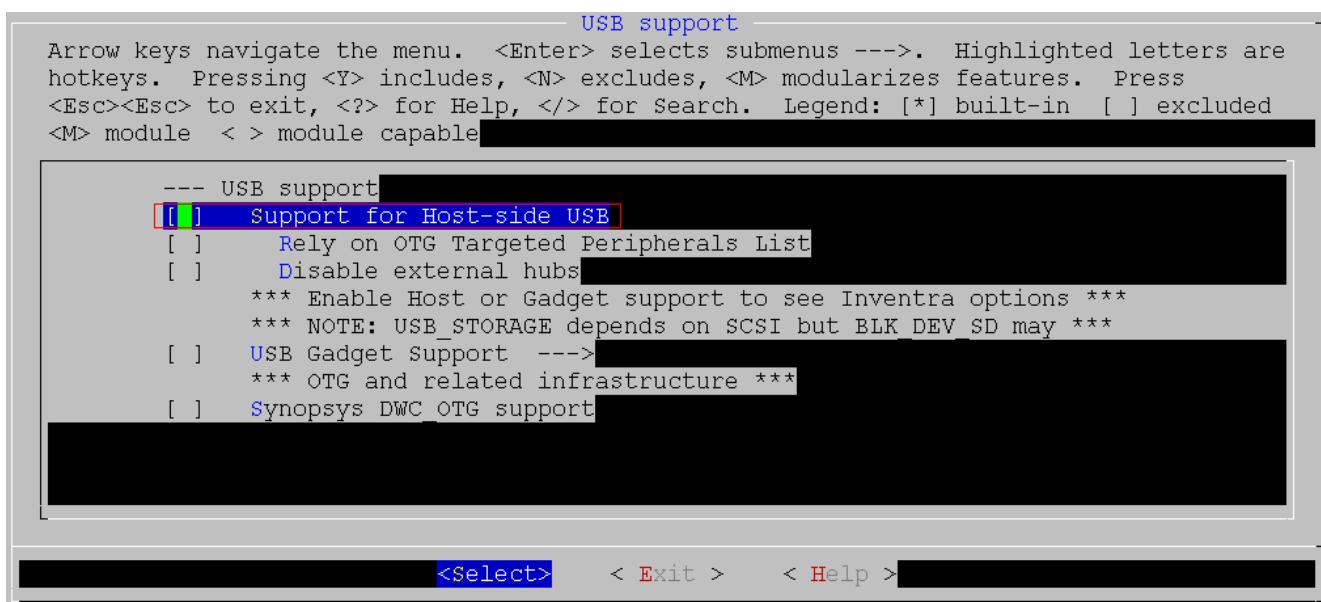
make linux_menuconfig // To configure linux kernel settings

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:

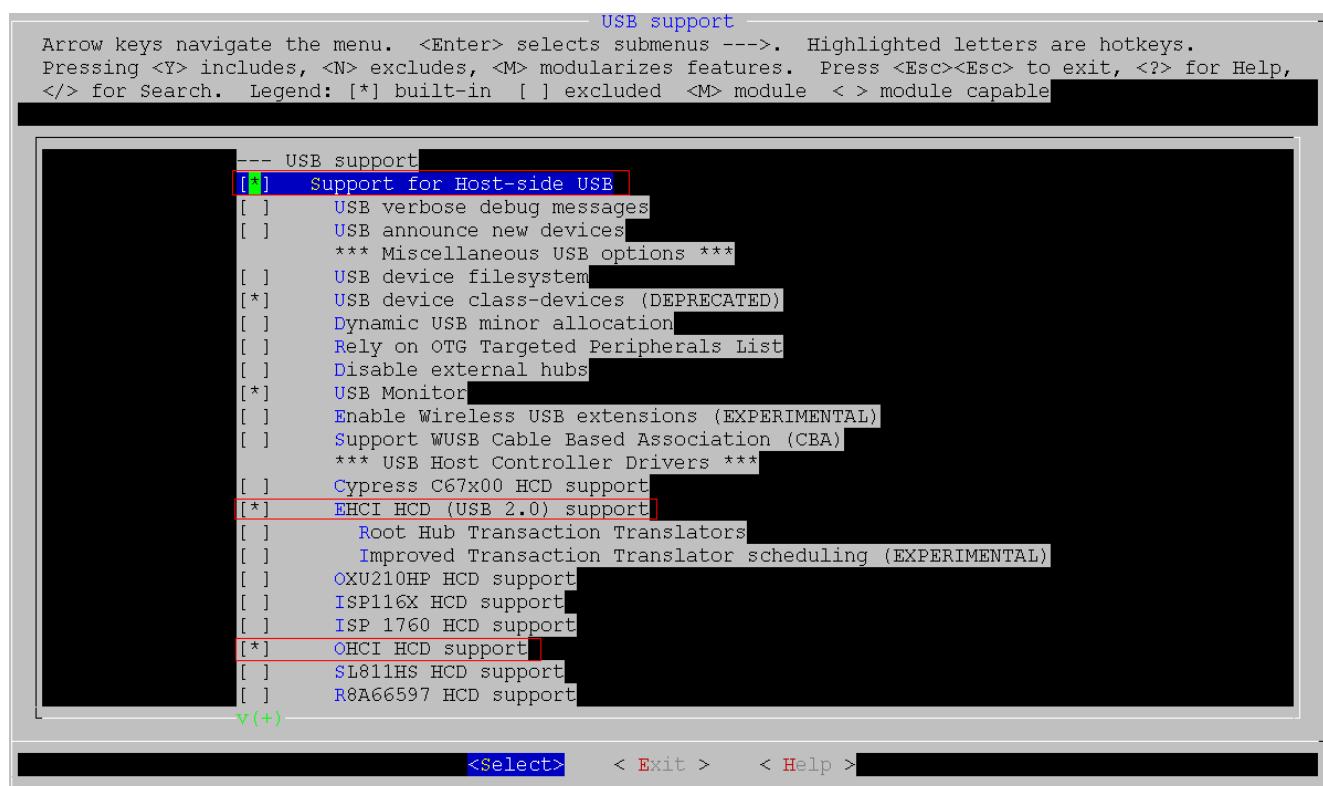
Device Drivers -->

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

make linux_menuconfig // To configure linux kernel settings

Menuconfig:

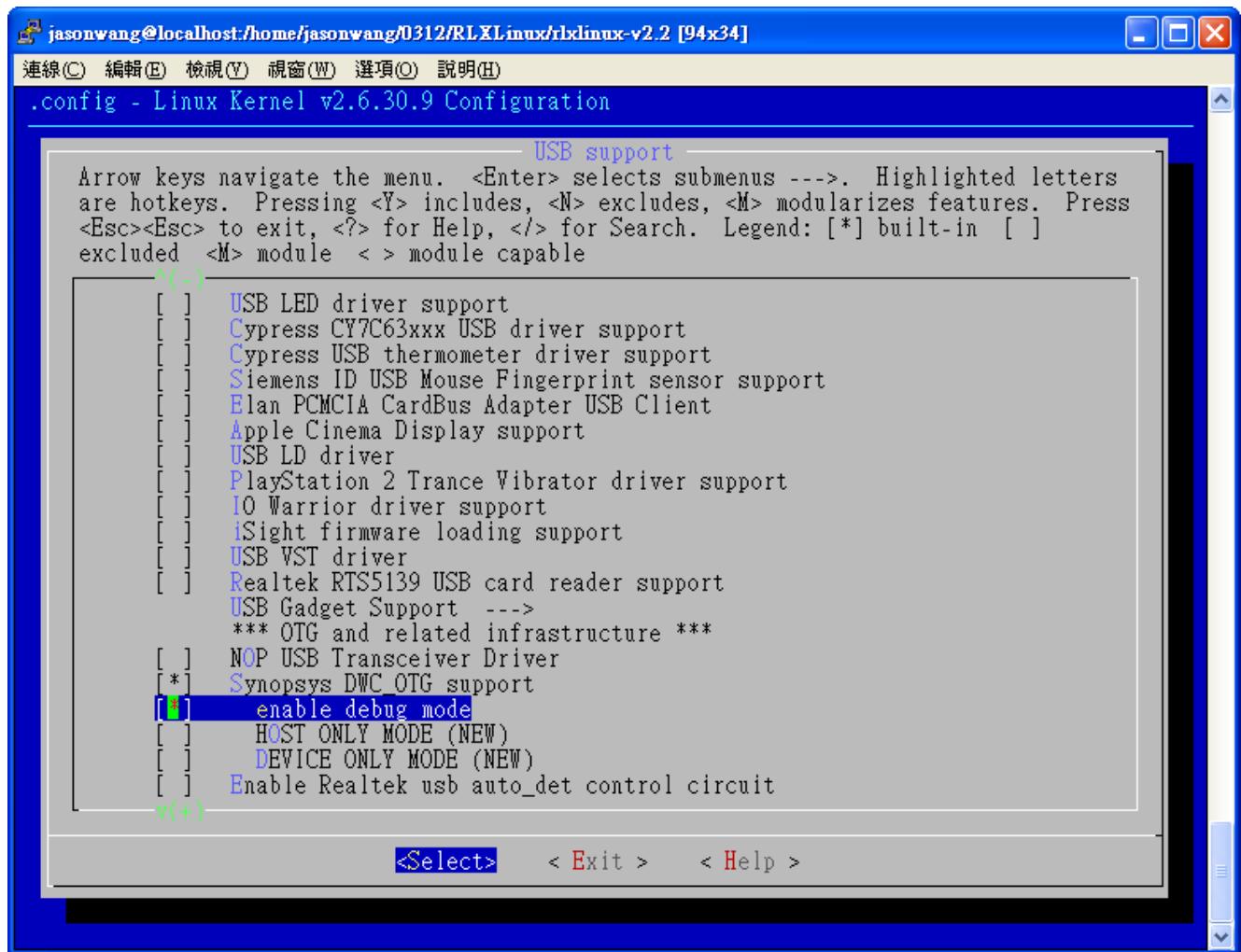
Device Drivers -->

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 config for file systems as follows.

make linux_menuconfig // To configure linux kernel settings

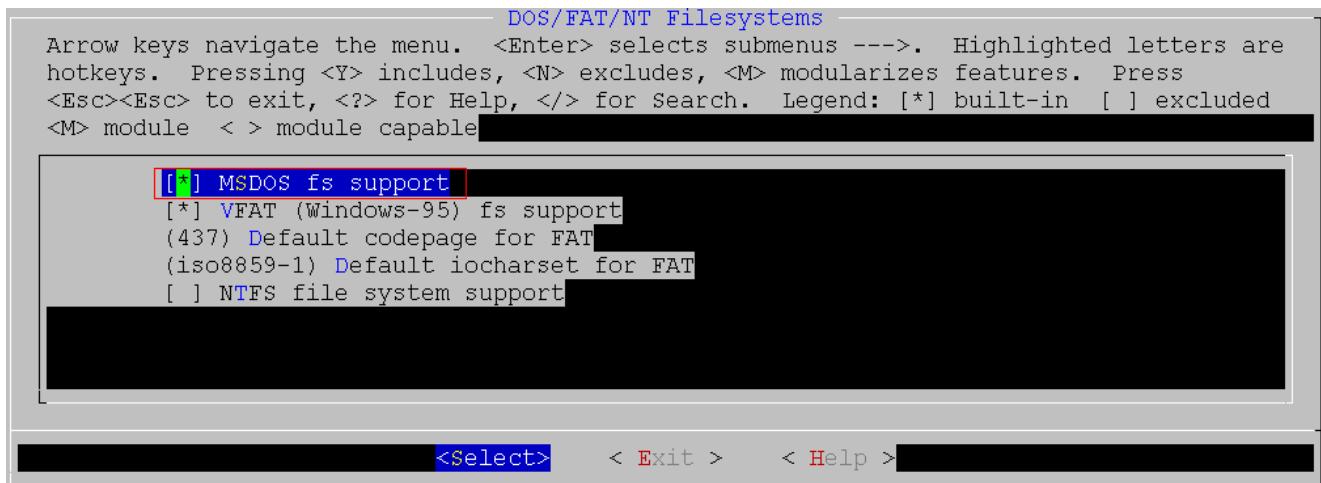
Menuconfig (1):

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



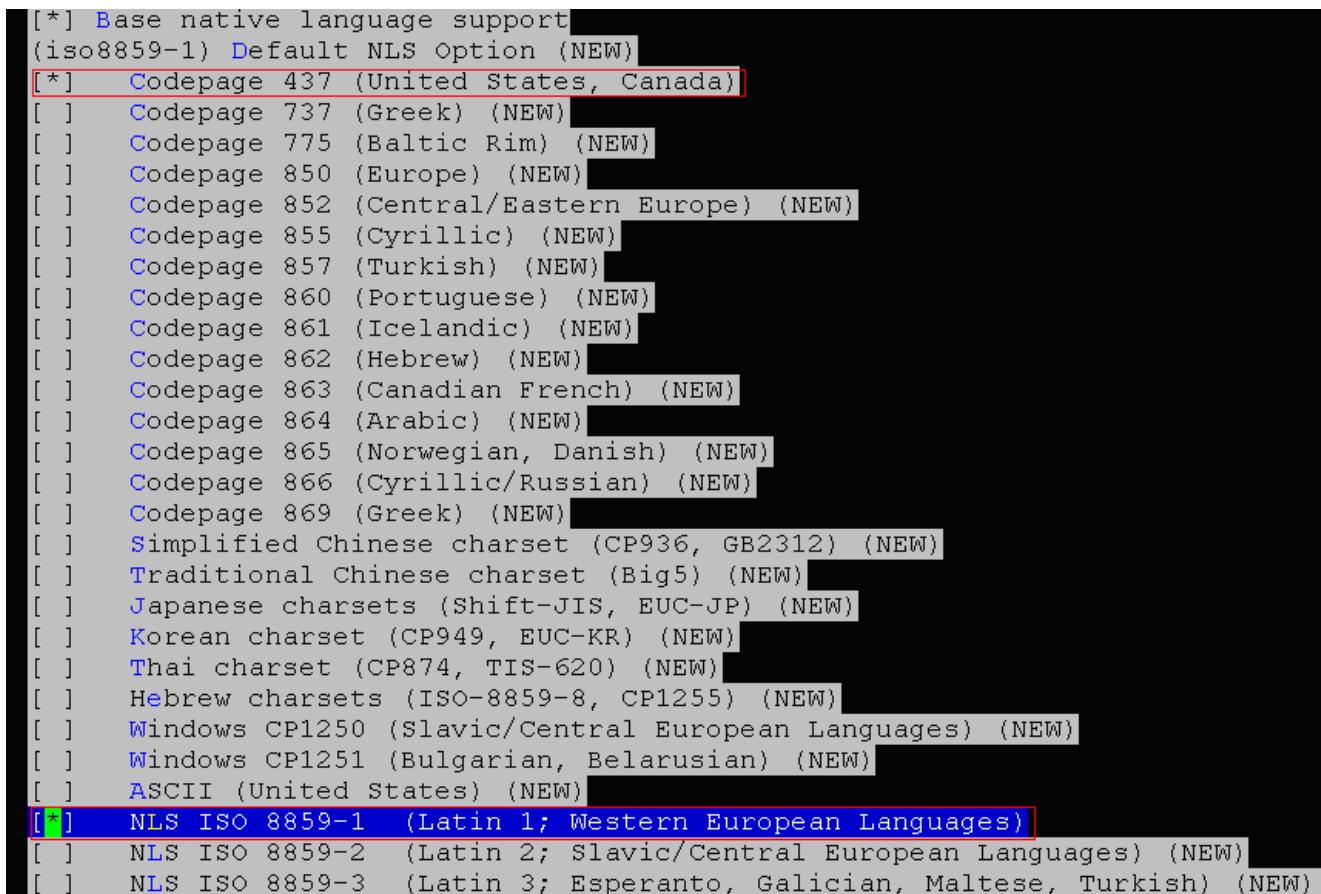
Menuconfig (3):

File systems --->

Native Language Support --->

Codepage 437 (United States, Canada) //Selected

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



4.3.1.3 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

```
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 < > module capable
```

```
^ (-)
[*]   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)
      *** USB Device Class drivers ***
[*]   USB Modem (CDC ACM) support
[*]   USB Printer support
[*]   USB Wireless Device Management support
[*]   USB Test and Measurement Class support
      *** 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
[*]   Datafab Compact Flash Reader support
v (+)

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

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
```

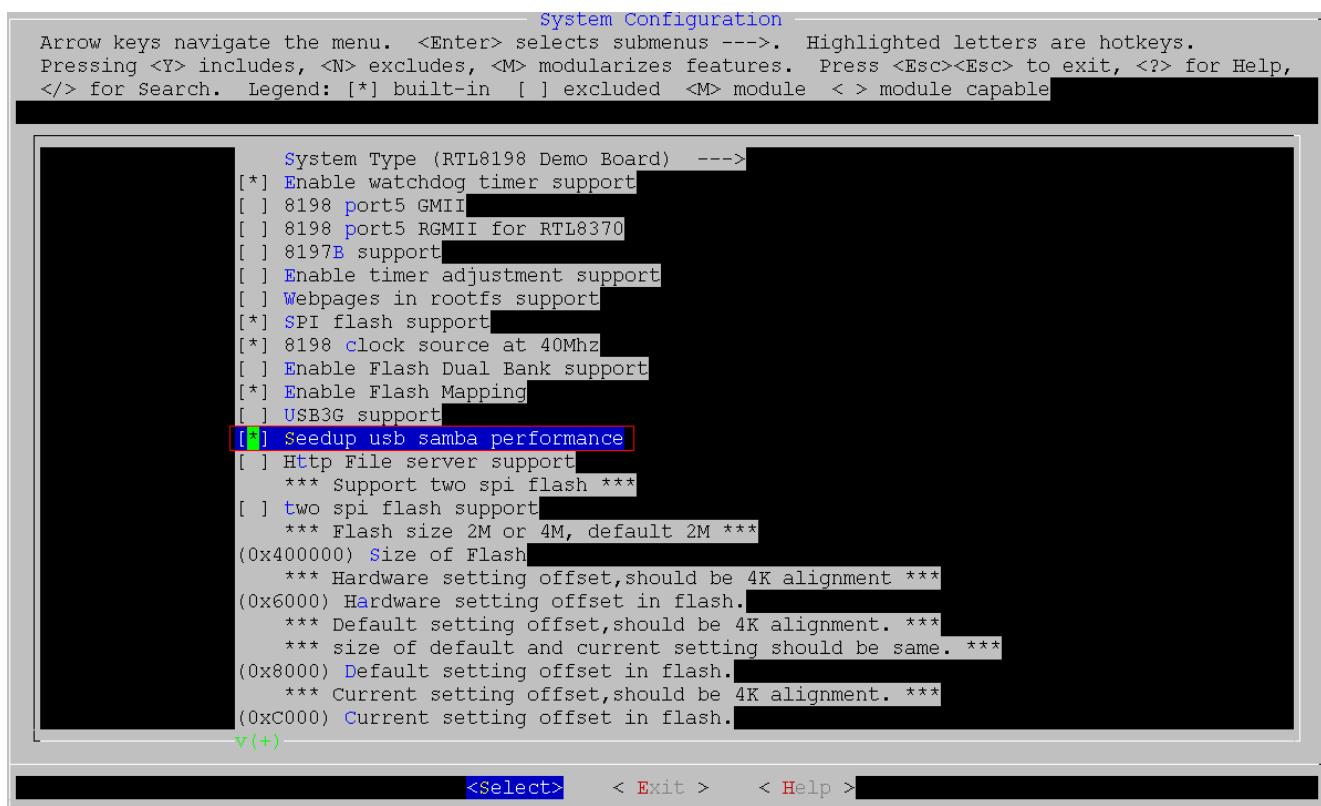
4.3.1.5 System configure

make linux_menuconfig // To configure linux kernel settings

Menuconfig:

System Configuration --->

[*] Seedup usb samba performance



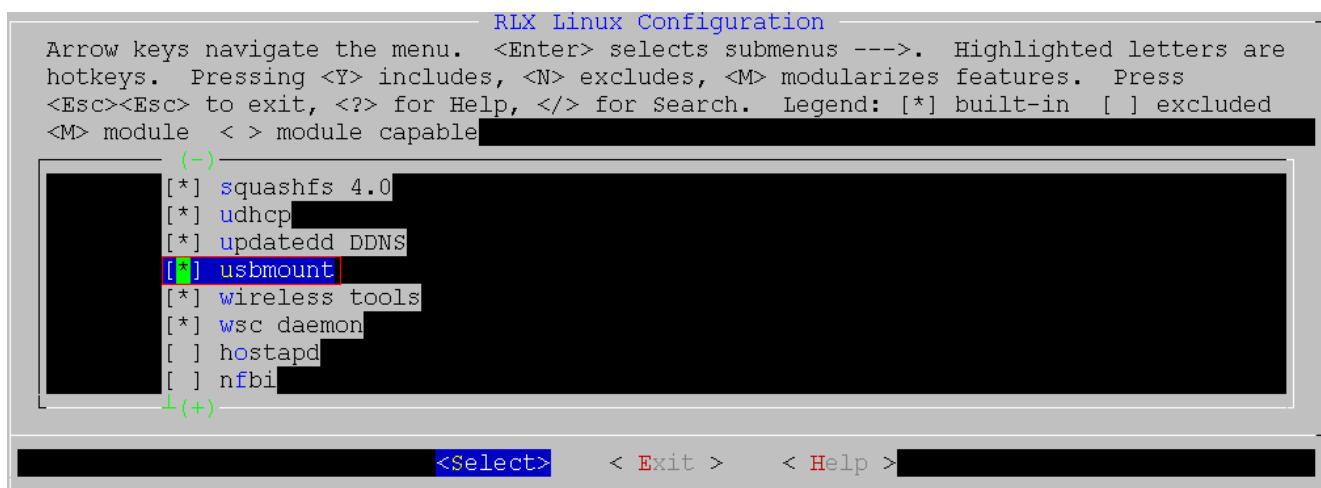
4.3.2 Samba application configure

4.3.2.1 enable usbmount

make users_menuconfig // To configure application settings

Menuconfig:

usbmount // selected



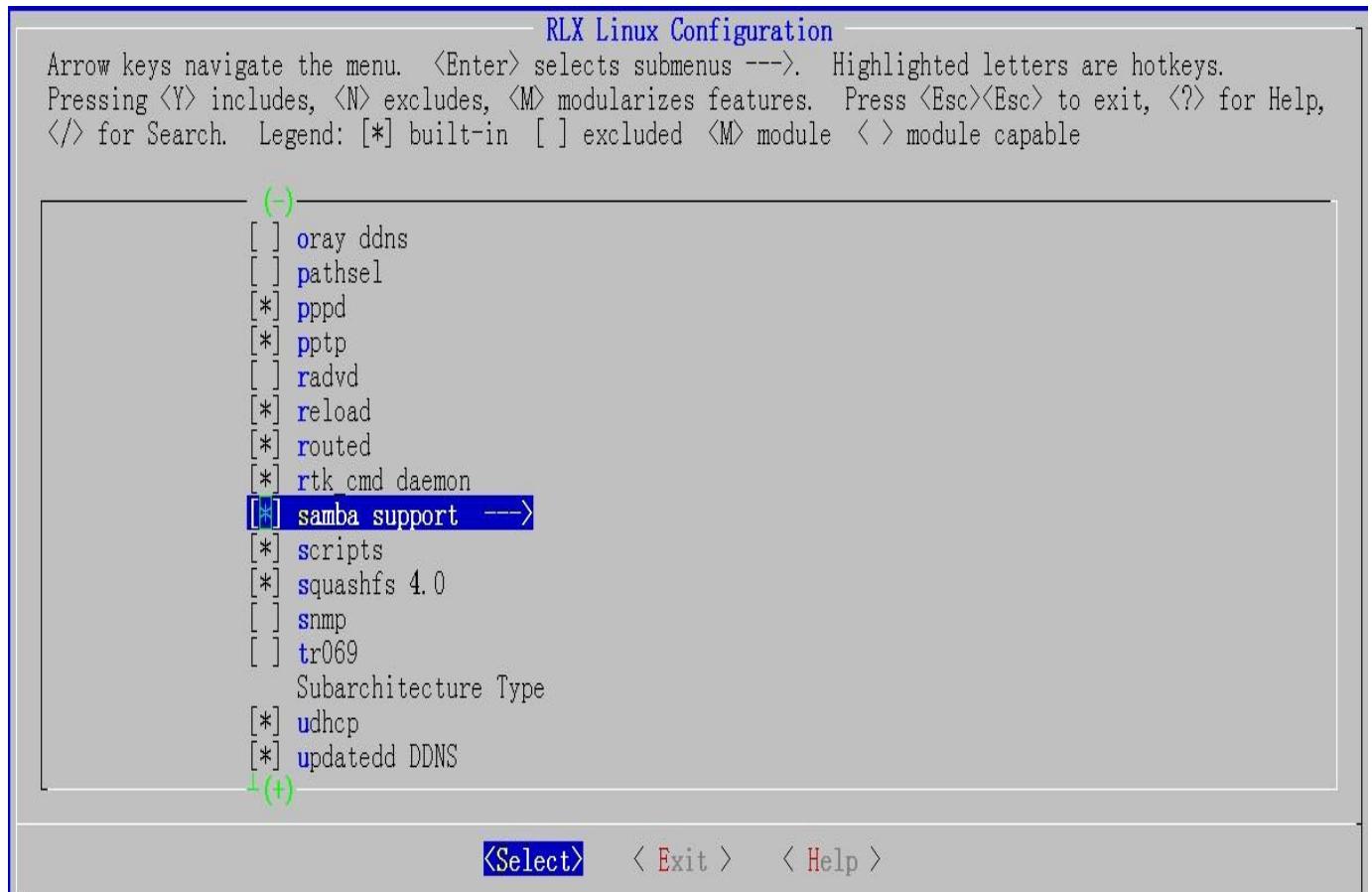
4.3.2.2 enable samba

```
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.



--- samba support

samba version (samba-3.0.24) --->

samba version

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.

(X) samba-3.0.24
() samba-3.2.15

<Select> < Help >

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.

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-1.5.5-5281-EB-2.6.30-0.9.30.3-110714) --->
--- rtl8196c
Selected Target of SDK (1lnRouter_GW) --->
Selected Board Configuration (SPI flash + Squashfs) --->
IC Test Configuration --->
--- config components
[ ] Config kernel (NEW)
[ ] Config users (NEW)
[ ] Config busybox
[ ] Load default settings
[ ] Save default settings
---
```

↓(+) ↴

<Select> < Exit > < Help >

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 Consle:

```
#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.

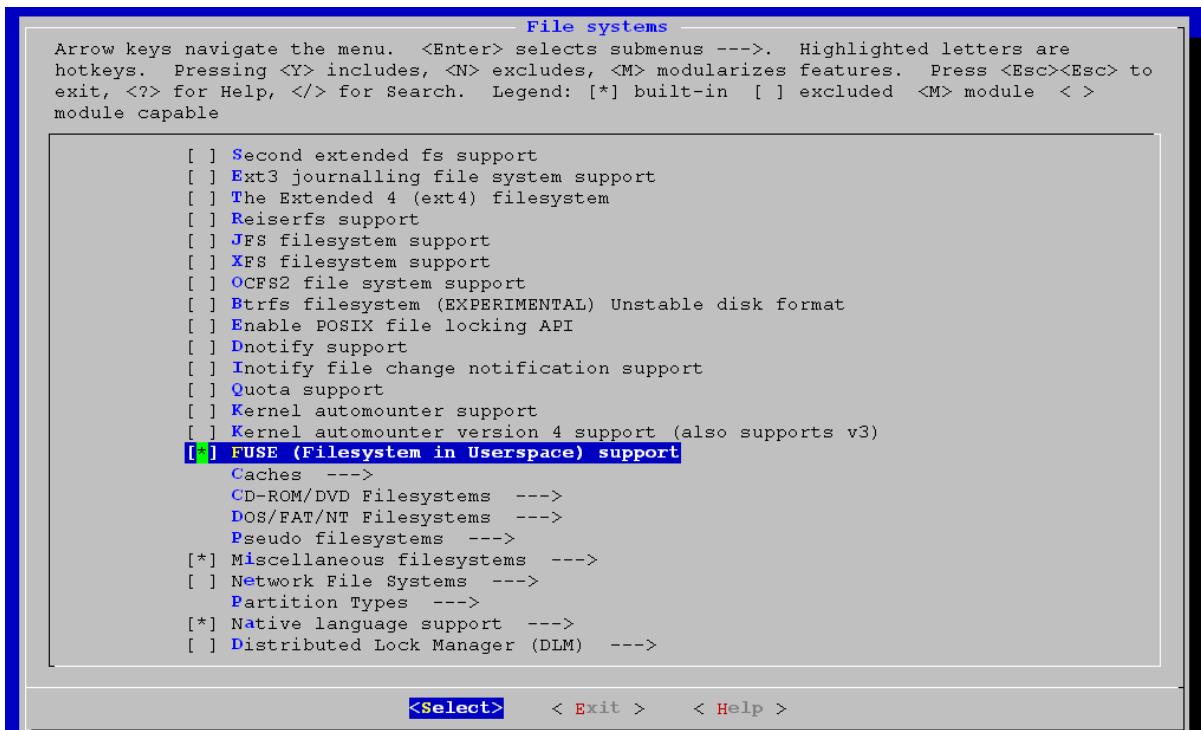
4.3.7 How to enable ntfs file system for samba

4.3.7.1 Linux kernel configure 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 configure for ntfs file systems as follows.

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

Menuconfig:

```
ntfs3g // selected
```

```
ntfs-3g version (ntfs-3g-2010.10.2) --->
```

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

```
(-)
[*] iapp
[*] igmp proxy
[ ]   support igmpv3 proxy
[*] iproute2
[*] iptables
[ ]   l7filter_pat
[*] iwcontrol
[*] l2tpd
[*] lltdd
-* mini_upnp
[*] miniigd
[ ] mkdosfs
[ ] mp_daemon
[*] ntfs3g
      ntfs-3g version (ntfs-3g-2010.10.2) --->
[*] ntpclient
(+)
```

<Select> < Exit > < Help >

Note:

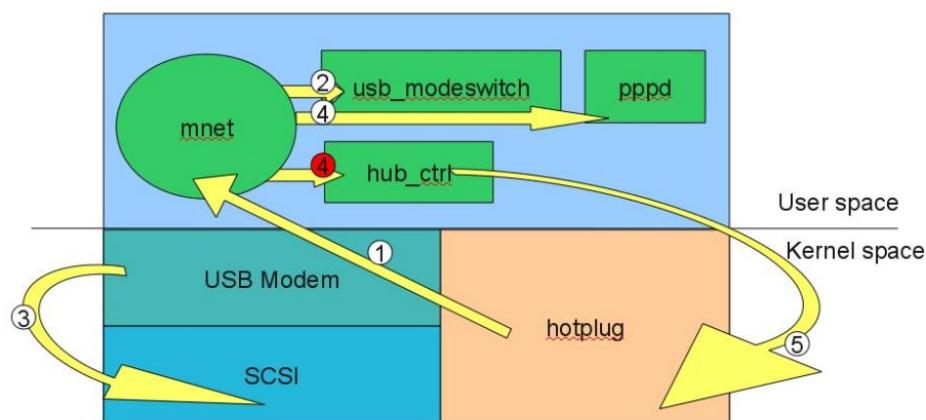
For ntfs-3g-2009.3.8 , has a balanced performance on ntfs read/write; while for ntfs-3g-2010.10.2 ,has a high performance on ntfs read(+15% on rtl8196c) and low on write(-65% on rtl8196c)



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 sram/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
```

4.4.3 User configure

1) User space programs

```
--- USB3G support
[ ] comgt
[*] chat
[*] libusb
[*] usb-modeswitch
[*] mbpk_eject
[*] usb-modeswitch-data
[*] usbutils
[*] 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 a 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
crtscts
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 dongle's type and find a matched config for usb_modeswitch when inserted (ie. 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, 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 Realtek WLAN Access Point management interface. The left sidebar contains a tree view of site contents: Site contents, Setup Wizard, Operation Mode, Wireless, TCP/IP Settings (selected), LAN Interface, WAN Interface, Firewall, QoS, Route Setup, Management (selected), Status, Statistics, DDNS, Time Zone Setting, Denial-of-Service, Log, Upgrade Firmware, Save/Reload Setting, and Password.

The main content area is titled "WAN Interface Setup". It displays configuration fields for WAN access type (set to "USB3G"), user name, password, PIN, APN (internet), dial number (*99#), connection type (Manual), idle time (5 minutes), and MTU size (1452 bytes).

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 status as follows:

- ① B3G Removed: there is no usb dongle on the DUT.
- ② 3G Modem Initializing...: the usb dongle has been inserted, and the system is starting the 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 hang up.

The screenshot shows the Realtek WLAN Access Point management interface. The left sidebar contains a tree view of site contents: Site contents, Setup Wizard, Operation Mode, Wireless, TCP/IP Settings, Firewall, QoS, Route Setup, Management (selected), Status, Statistics, DDNS, Time Zone Setting, Denial-of-Service, Log, Upgrade Firmware, Save/Reload Setting, and Password.

The main content area is titled "Access Point Status". It displays current status and basic settings of the device. The "System" section shows Uptime (0day:3h:10m:47s), Firmware Version (v2.2), and Build Time (Mon Jun 28 17:45:19 CST 2010). The "Wireless Configuration" section shows Mode (AP), Band (2.4 GHz (B+G+N)), SSID (RTK_3G_AP), Channel Number (11), Encryption (WPA2), BSSID (00:e0:4c:66:11:92), and Associated Clients (0). The "TCP/IP Configuration" section shows Attain IP Protocol (Fixed IP), IP Address (192.168.1.254), Subnet Mask (255.255.255.0), Default Gateway (192.168.1.254), DHCP Server (Enabled), and MAC Address (00:e0:4c:66:11:90). The "WAN Configuration" section shows Attain IP Protocol (USB3G Connected), IP Address (114.136.91.229), Subnet Mask (255.255.255.255), Default Gateway (0.0.0.0), and MAC Address.

4.4.5 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)

4.4.6 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: fw.bin 1981512.	Kernel: wapiCrypto.o 28044, wapi_wai.o 21044. User: WAPI certs related 64K. Total: 114624Byte.	1) No security: 15708KB. 2) WAPI-PSK: 15588KB. 3) WAPI-CERT(remote AS): 15516KB.
Enable WAPI support and enable local AS	Kernel: linux.bin 831506. User: root.bin 1630226. Total: fw.bin 2580110.	Kernel: wapiCrypto.o 28044, wapi_wai.o 21044. User: libcrypto.so.0.9.8 1485928, libssl.so.0.9.8 278900, openssl 402724, ecdsatest 20776, openssl.cnf 9675,	1) No security: 15500KB. 2) WAPI-PSK: 15440KB. 3) WAPI-CERT(remote AS): 15312KB. 4) WAPI-CERT(local AS): 13316KB.

		<p>readFileSize 4708, loadWapiFiles 10956, storeWapiFiles 17320, aeUdpClient 41136, aseUdpServer 31544, genUserCert.sh 2808, initCAFiles.sh 1917, revokeUserCert.sh 1063, WAPI certs related 64K.</p> <p>Total: 2424079Byte.</p>	
--	--	---	--

4.6 Mesh support (not support now)

make linux_menuconfig // To configure linux kernel settings

Menuconfig:

Device Drivers --->

 Network device support --->

 Options for Realtek SoC --->

 IEEE 802.11s mesh support // selected if mesh support

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	CFI
<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's 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

```

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. 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 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
[ ]     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.

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.

4.8 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_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
```

4.8.1.2 bootloader menuconfig

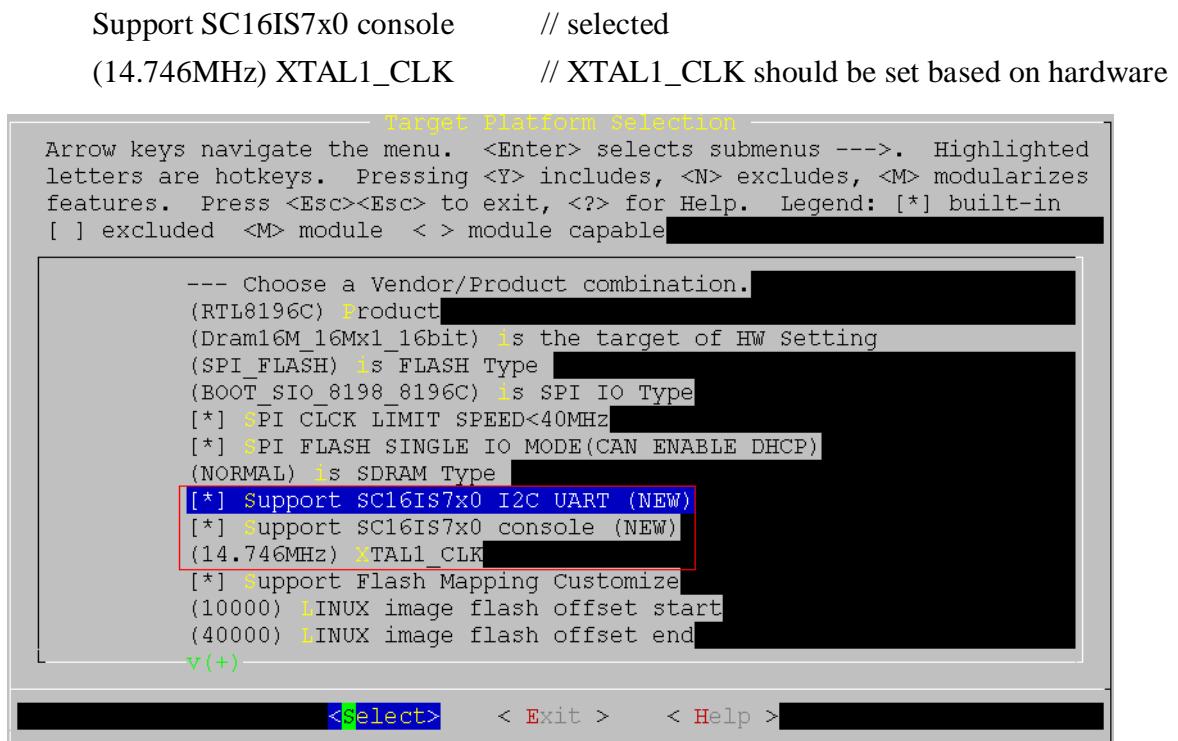
Choose the correct configuration related to I2C UART in bootloader's menuconfig.

make menuconfig // To configure bootloader settings

Menuconfig:

Target Platform Selection --->

Support SC16IS7x0 I2C UART // selected



4.8.2 Enable I2C UART in kernel

4.8.2.1 Modify code for GPIO pin in kernel

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
```

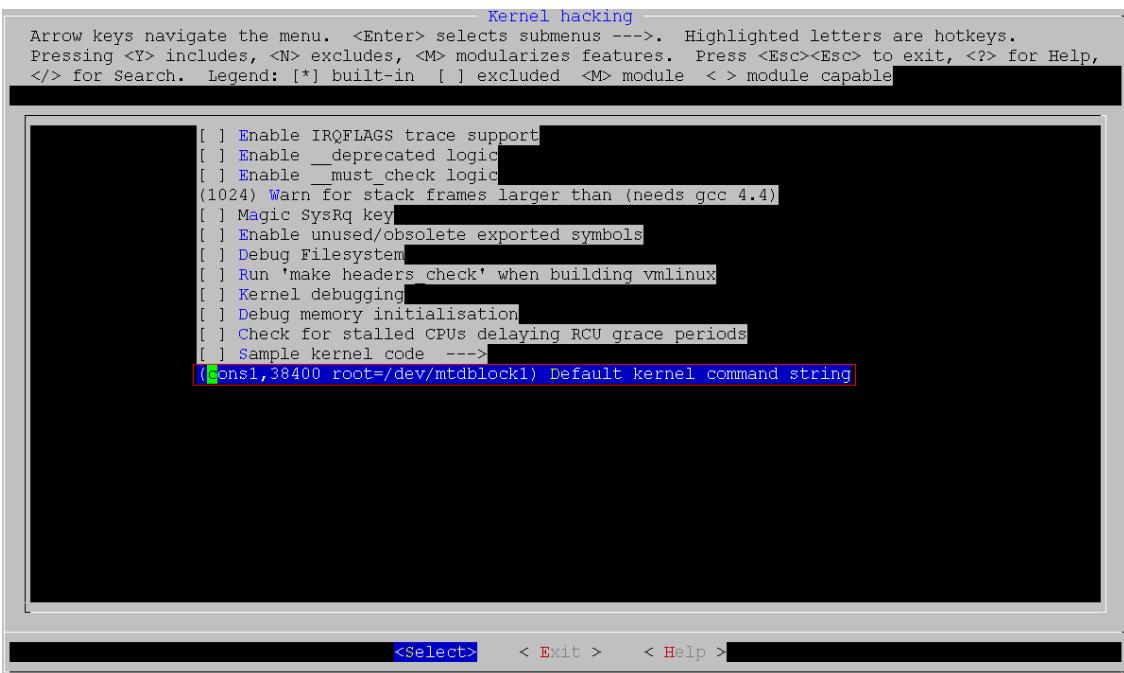
4.8.2.2 kernel menuconfig

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



2) Modify the 8250/16500 serial port number.

`make linux_menuconfig` // To configure linux kernel settings

Menuconfig:

Device Drivers --->

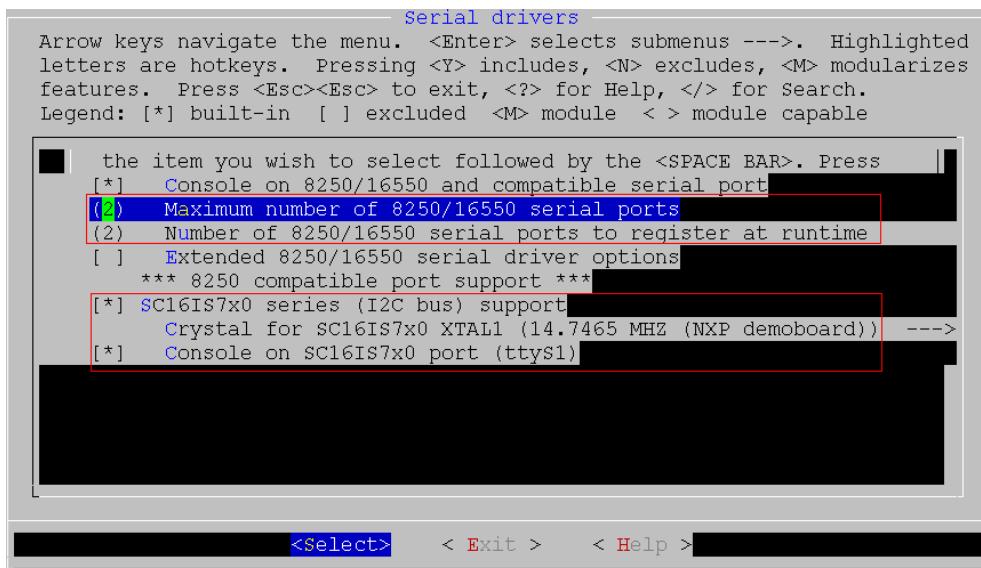
Character devices --->

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.



4.9 DLNA support

Note: at present, DLNA is only supported for RTL8198.

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

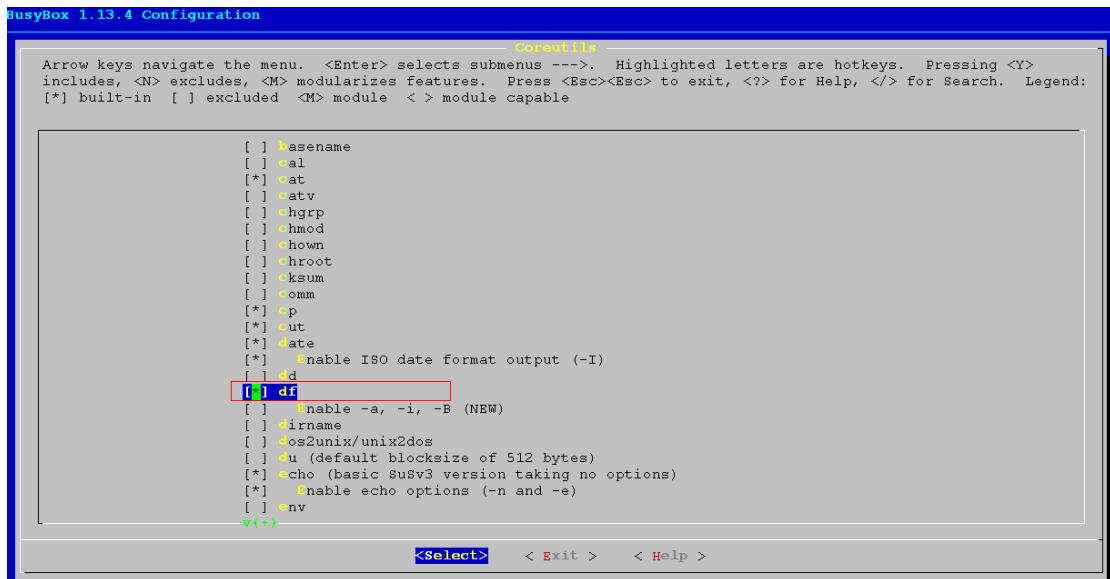
4.9.2.1 busybox configure

make menuconfig and select Config busybox

Menuconfig:

Coreutils-->

df support // selected

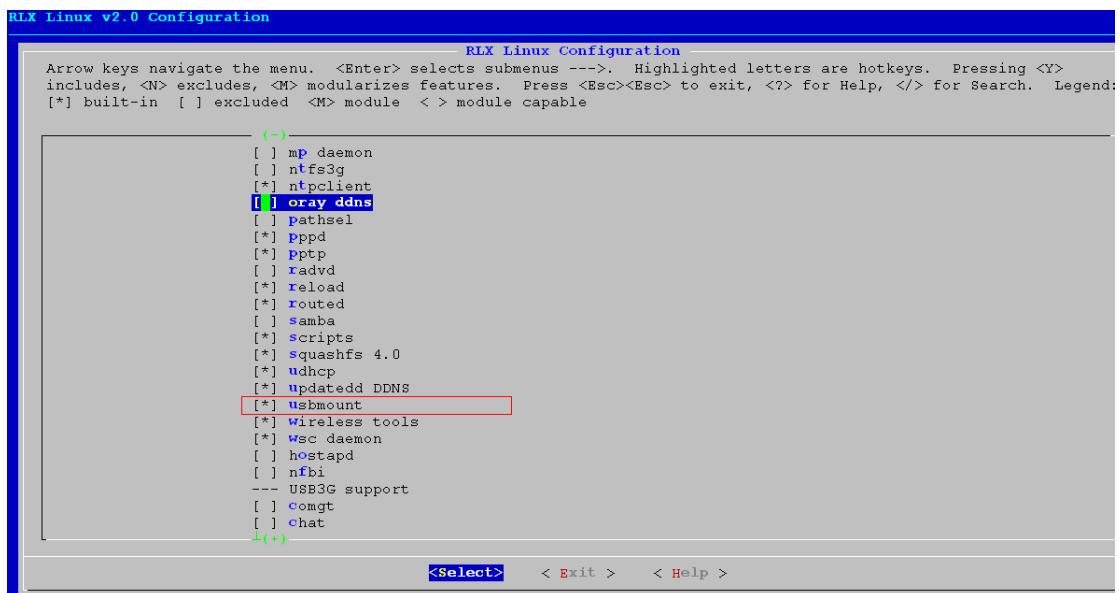


4.9.2.2 enable usbmount

make users_menuconfig // To configure application settings

Menuconfig:

usbmount // selected

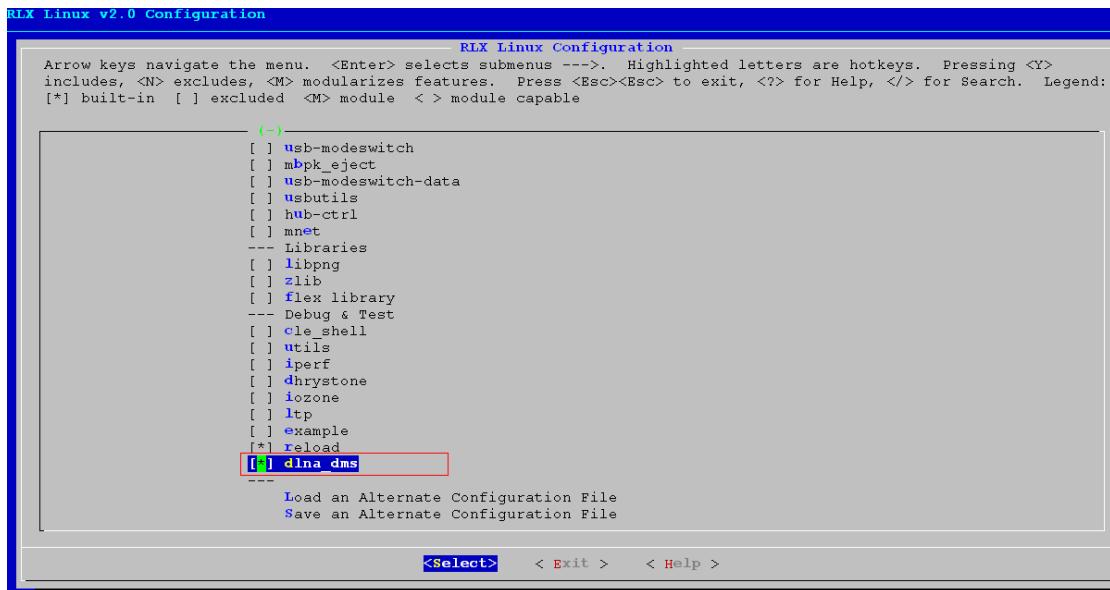


4.9.2.3 enable dlna

make users_menuconfig // To configure application settings

Menuconfig:

Dlna_dms // selected



4.9.3 Test dlna using USB flash disk

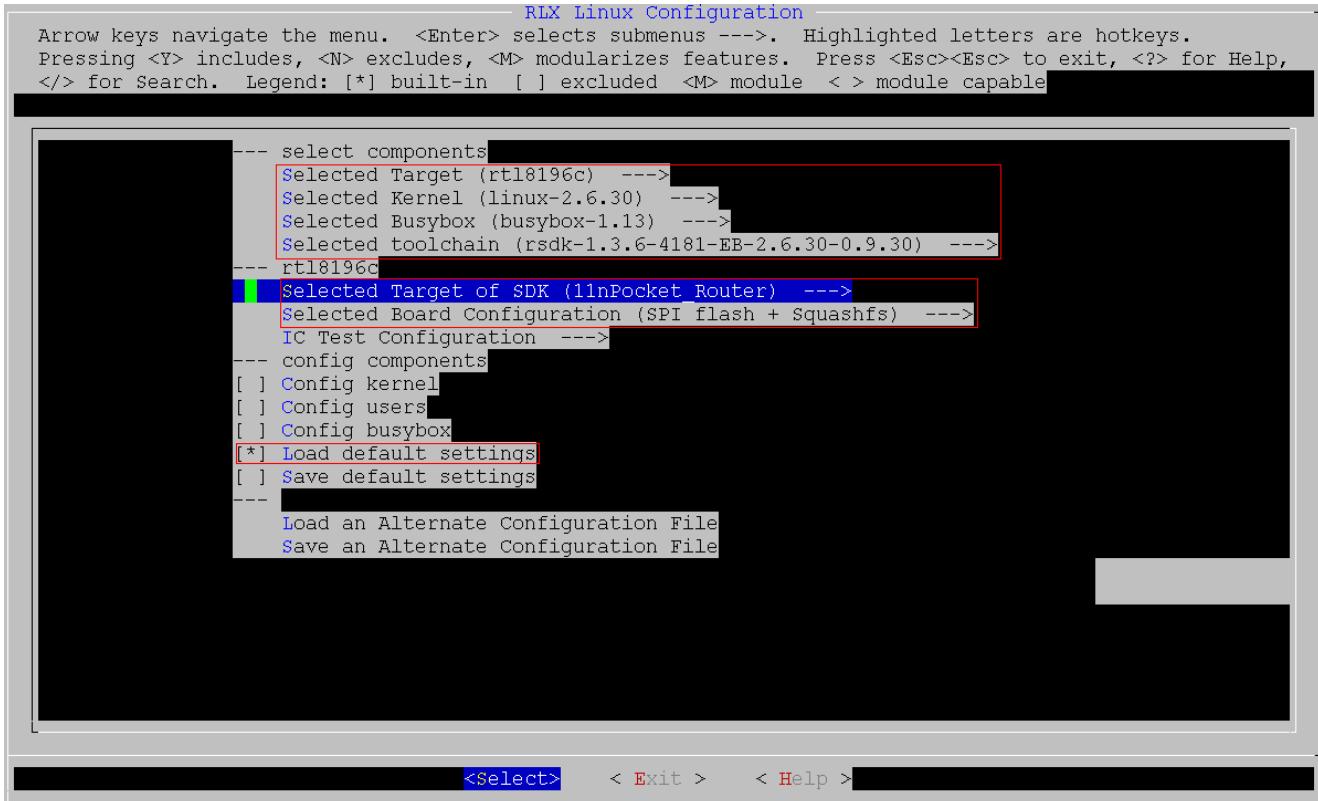
- (1) First of all, you should read and modify /boards/rtl8198/etc.default/ushare.conf.
- (2) After system boots up, plug-in an USB Flash disk. The directory to be shared is /var/tmp/usb/sda6/Media (you can modify this directory in ushare.conf).
- (3) Write this command: ‘ushare –f etc/ushare.conf &’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

4.10 Pocket AP support

At present, pocket AP is supported at RTL8196C. Configure pocket AP is as follows:

make menuconfig // To configure linux kernel settings

And the linux kernel configuration 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

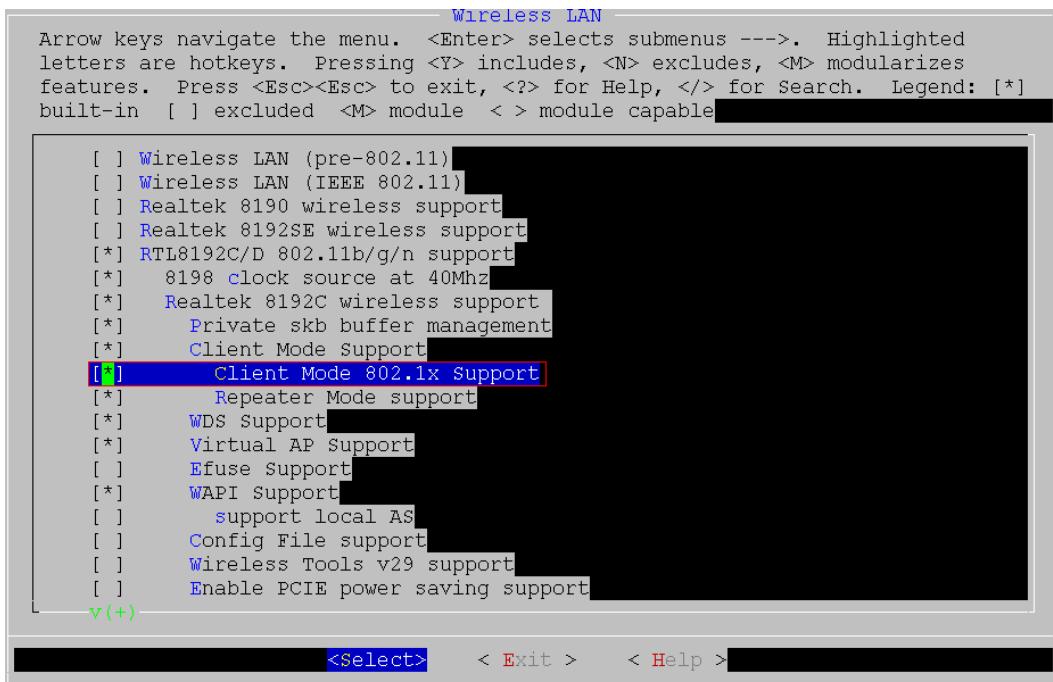
Menuconfig:

Device Drivers -->

 Network device support -->

 Wireless LAN -->

 Client Mode 802.1x Support // selected



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 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 has 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) IP conntracks which need do ALG.
- 3) when URL filter enabled, must disable hardware nat.
- 4) when enable 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(eg:ipsec)
- 10) 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.
- 11) won't affect alg when enable hardware nat
- 12) total 1024 hardware nat entries, bi-direction support 512 connection

Note: If wan type changed 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 fragement will be automatically processed, no other settings involved.

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

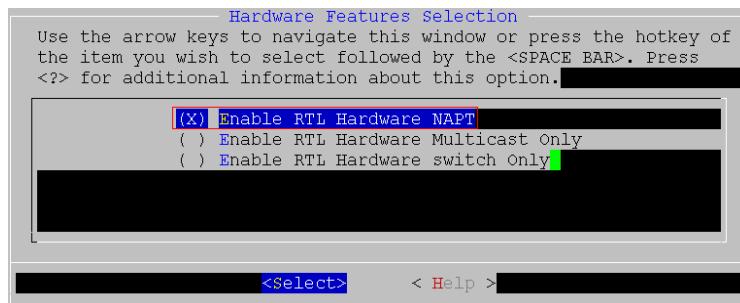
/* If selected, enable hardware L2 and multicast function */

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.



4.13 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
```

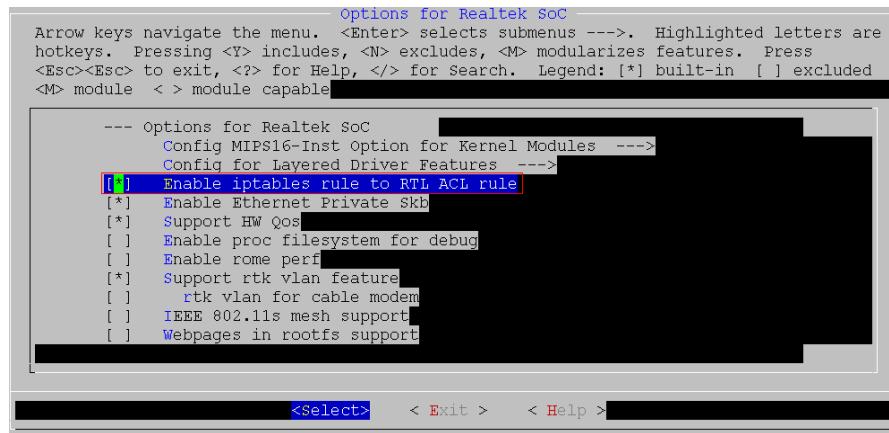
Menuconfig:

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:

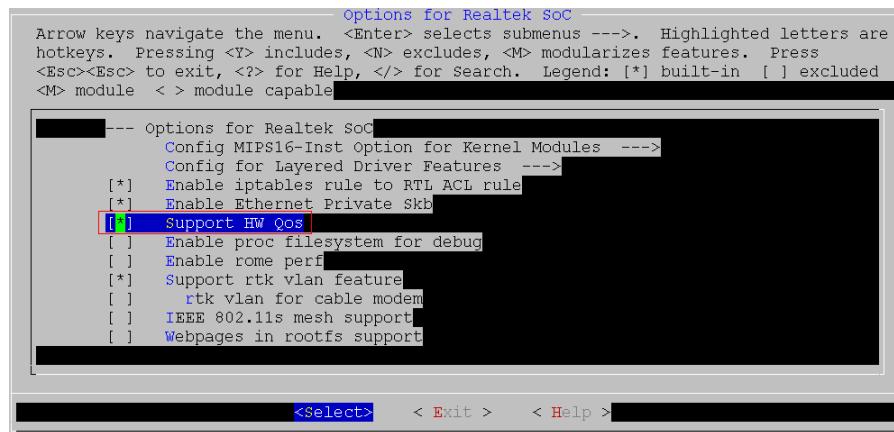
Device Drivers --->

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.

4.12.2.2 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 configre, input the following command in console please:

rtk_cmd qos SHOWFLOW_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:

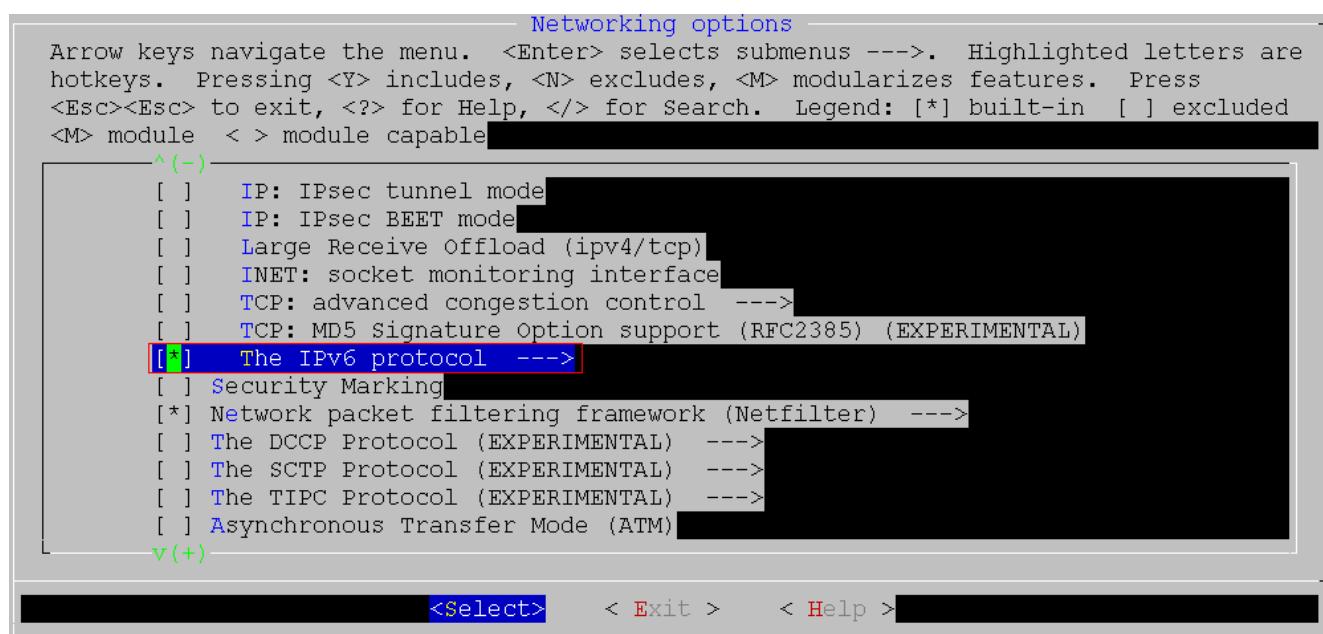
make linux_menuconfig // To configure linux kernel settings

Menuconfig:

1) Networking support --->

Networking options--->

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



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
[ ] IP6 Userspace queueing via NETLINK (OBSOLETE)
[*] IP6 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

```

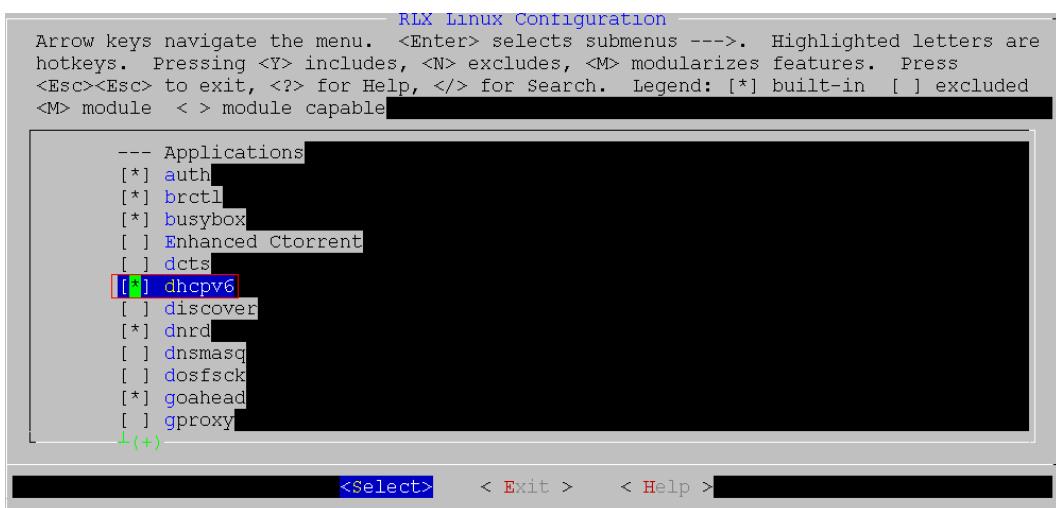
4.15.2 IPv6 daemon configure

Configure IPv6 daemon as follows:

make users_menuconfig // To configure application settings

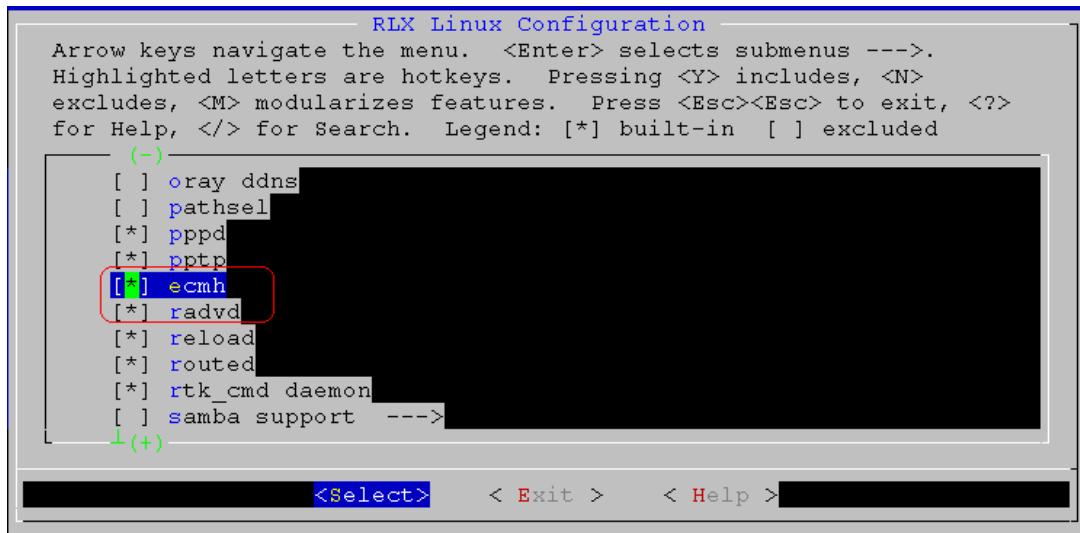
Menuconfig:

1) dhcpcv6 // Selected

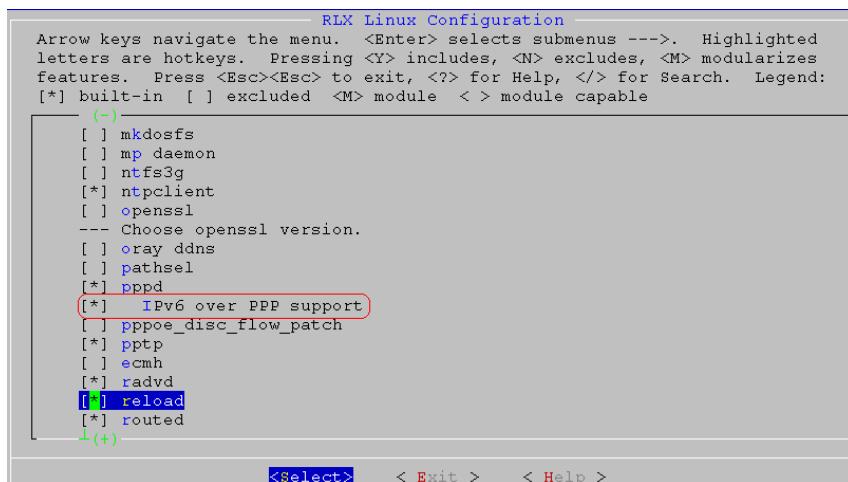


2) radvd // Selected

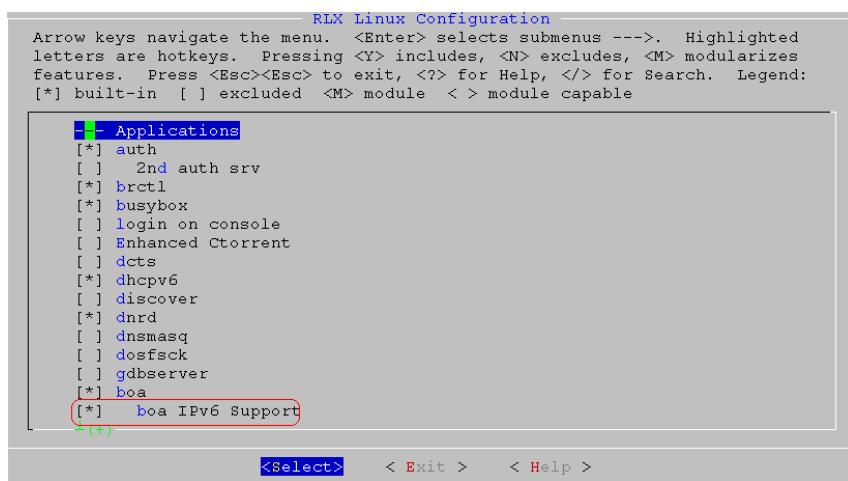
3) ecmh // Selected



4) PPP support IPv6 //Selected



5) BOA support IPv6 //Selected



6) ip6tables support //Selected

```

----- 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
(=)
[*] iapp
[*] igmp proxy
[ ] support igmp proxy multiwan
[ ] support igmpv3 proxy
[*] mld proxy
[ ] support mld proxy multiwan
[*] iproute2
[*] iptables
[*] ip6tables
[ ] l7filter_pat
[*] iwcontrol
[*] l2tpd
[*] lltdd
-- mini_upnp
[*] minii_gd
[ ] mkdosfs
(+)

<Select> < Exit > < Help >

```

7) dnsmasq support //Selected

```

----- 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
--- Applications
[*] auth
[ ] 2nd auth srv
[*] brctl
[*] busybox
[ ] login on console
[ ] Enhanced Ctorrent
[ ] dcts
[*] dhcpcv6
[ ] discover
[*] dnrd
[*] dnsmasq
[ ] dosfsck
[ ] gdbserver
[*] boa
[*] boa IPv6 Support
[ ] uWiFi
(+)

<Select> < Exit > < Help >

```

8) mldproxy support //Selected

```

----- 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
(=)
[*] dhcpcv6
[ ] discover
[*] dnrd
[ ] dnsmasq
[ ] dosfsck
[ ] gdbserver
[*] boa
[*] boa IPv6 Support
[ ] uWiFi
[ ] alsa related
[ ] gproxy
[ ] Multi PPPoE Support
[*] iapp
[*] igmp proxy
[ ] support igmp proxy multiwan
[ ] support igmpv3 proxy
[*] mld proxy
[ ] support mld proxy multiwan
[*] iproute2
(+)

<Select> < Exit > < Help >

```

4.15.3 Busybox for IPv6 configure

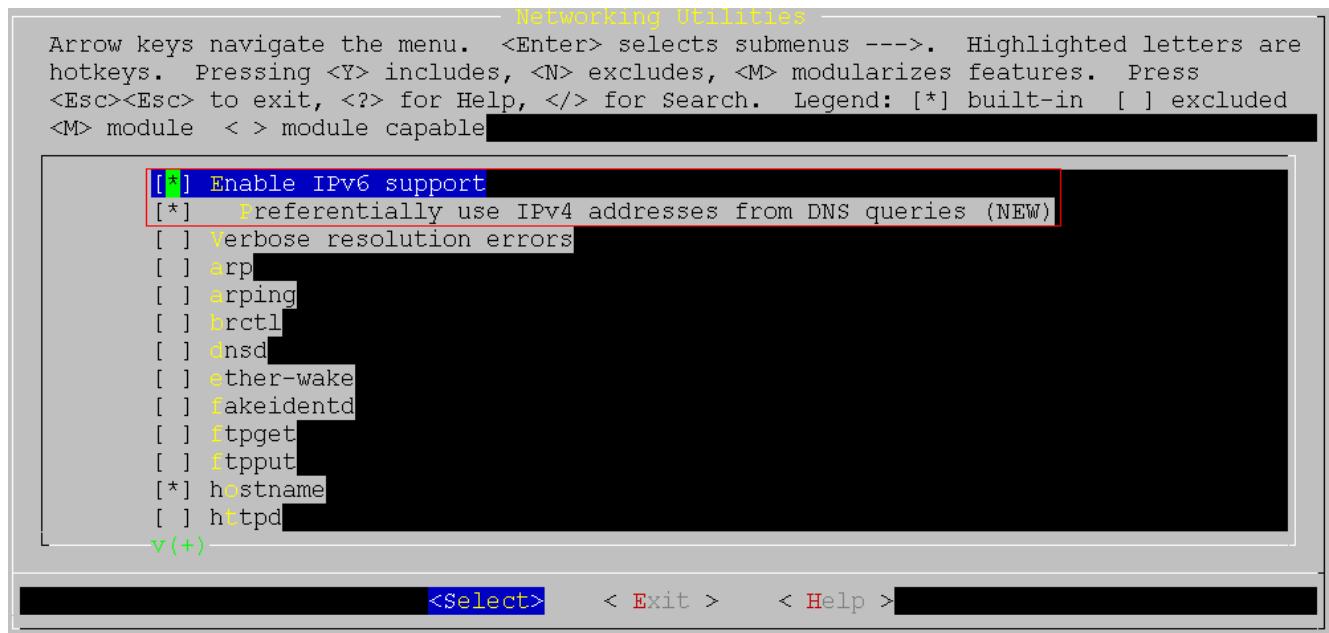
Enter the directory of Busybox and make menuconfig to configure Busybox for IPv6 as follows:

Menuconfig:

1) Networking Utilities --->

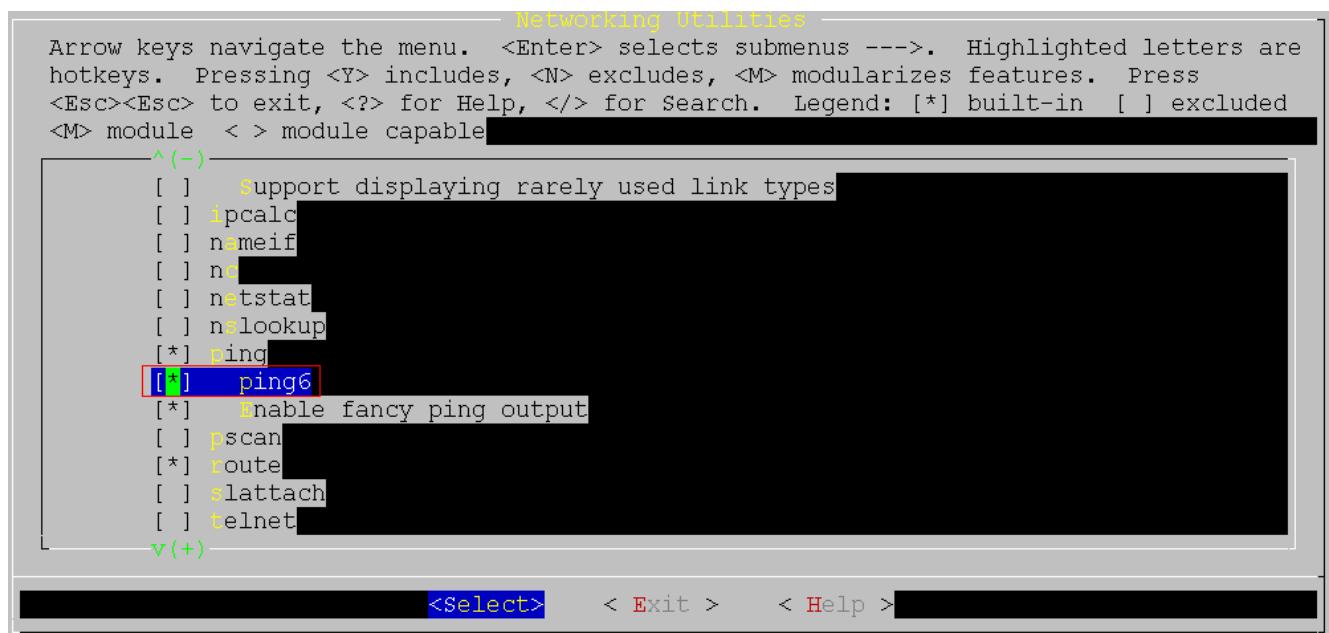
Enable IPv6 support // Selected

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
[*] SPI flash support
[*] Enable Flash Dual Bank support
[*] Enable Flash Mapping
[*] USB3G support
[*] Seedup 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 */

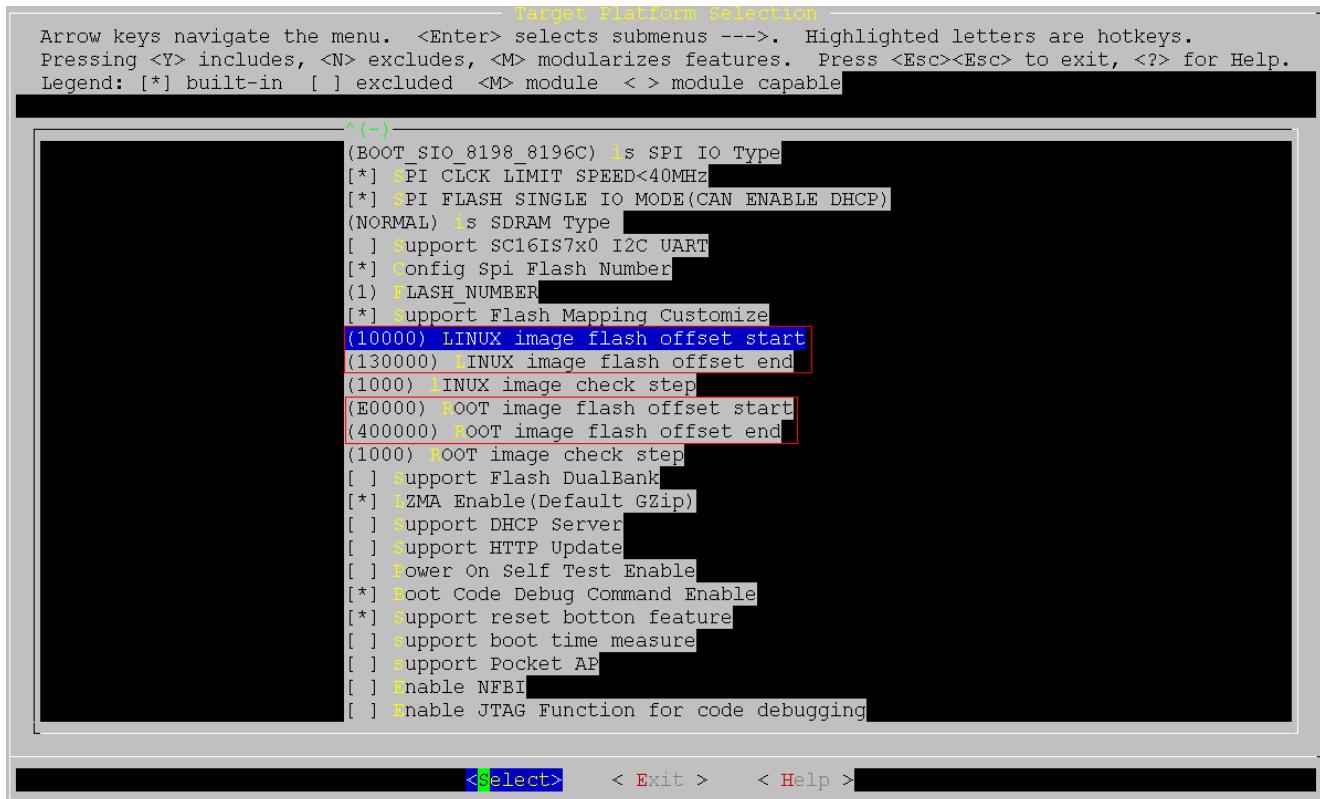
(10000) LINUX image flash offset start

(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:

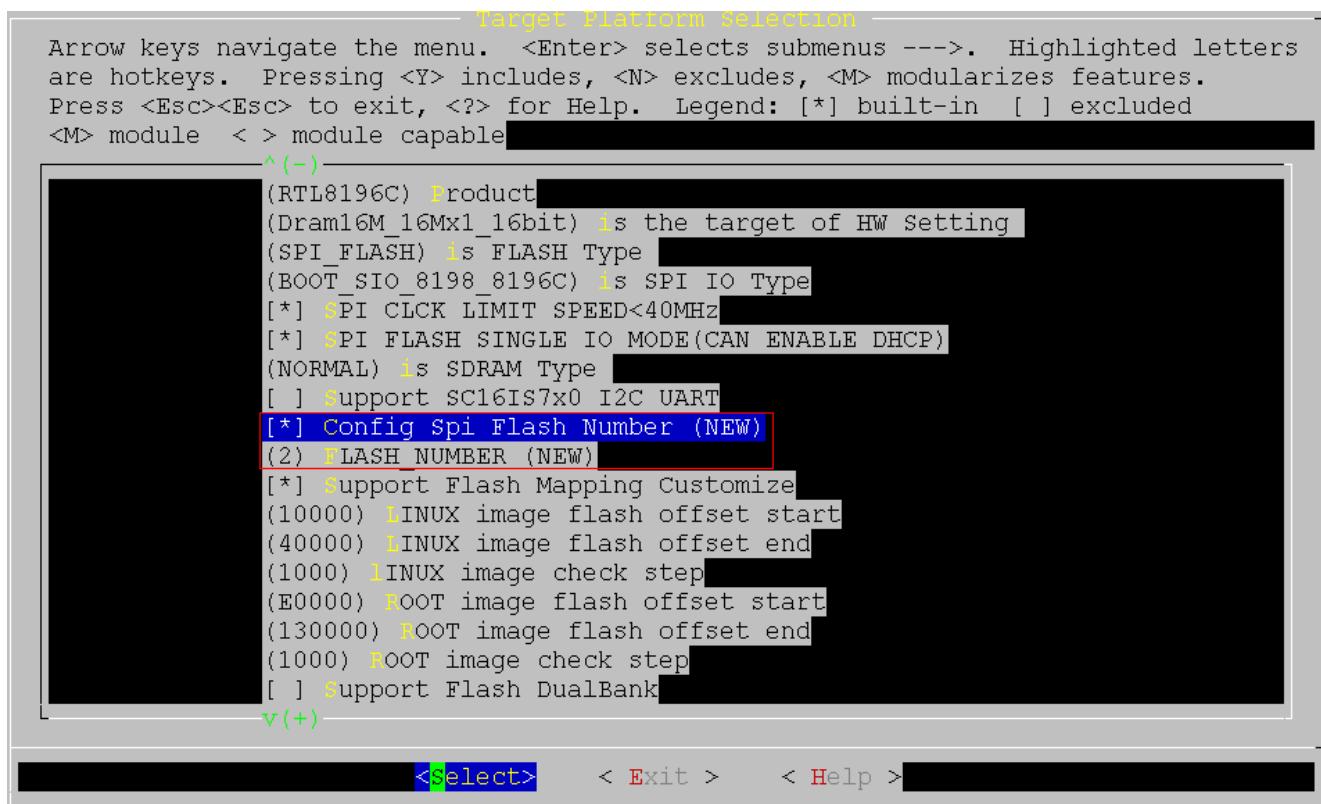
Target Platform Selection --->

//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)



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:

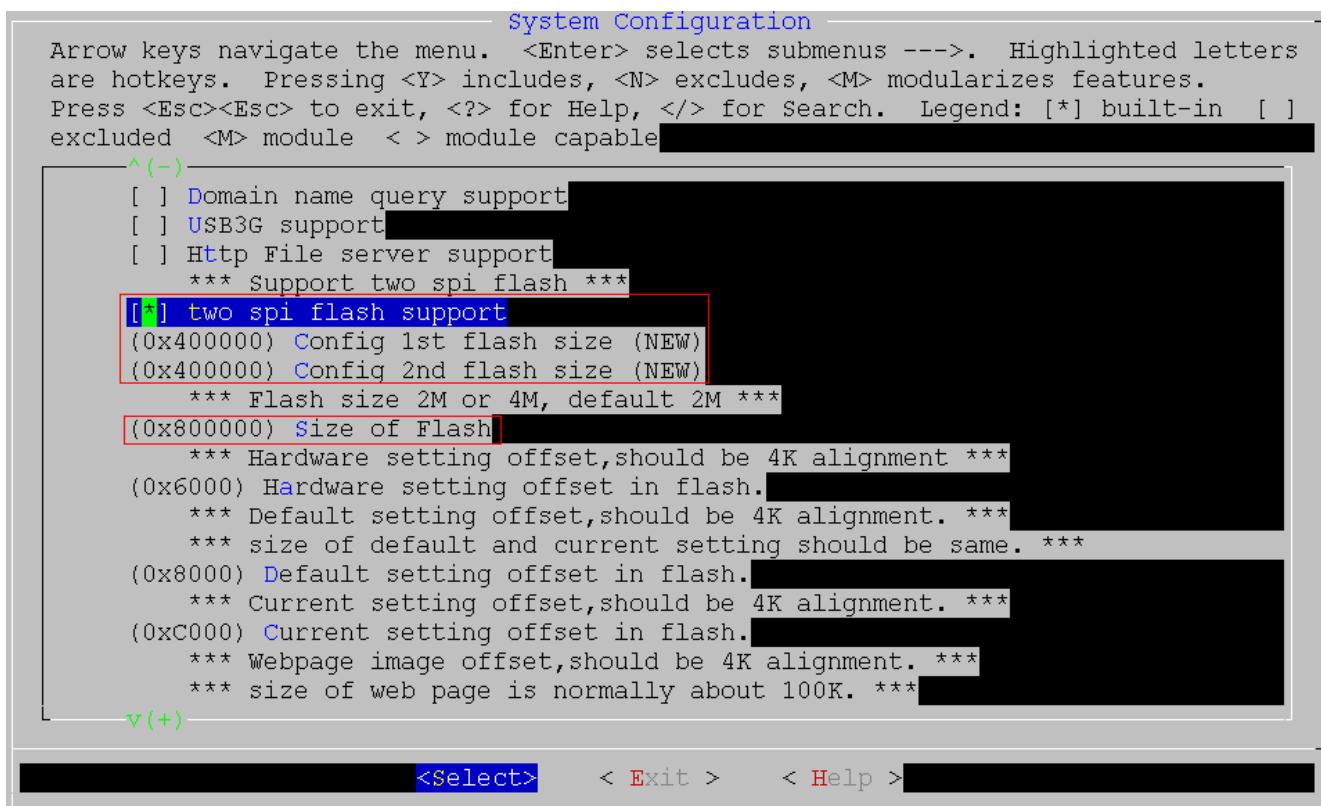
System Configuration --->

```
[*] 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.

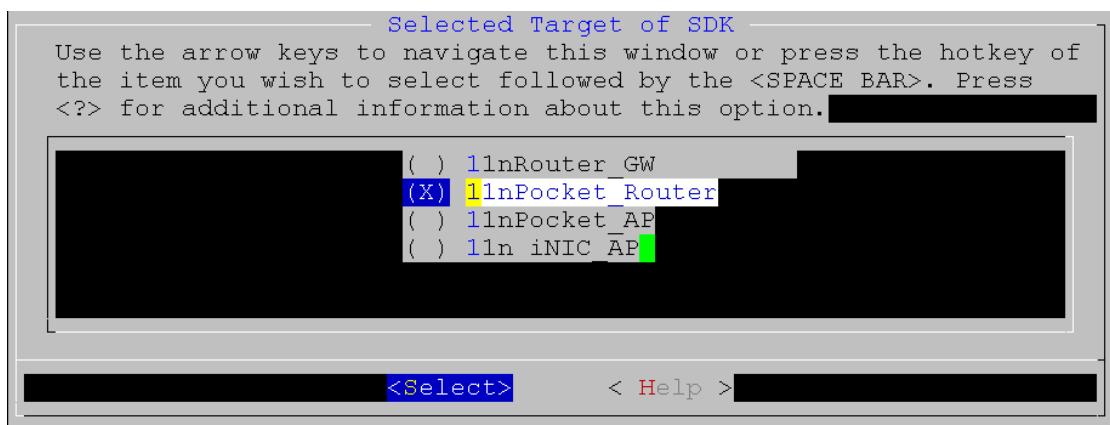
4.18 92C/D support

4.18.1 92C support for Pocket AP SDK

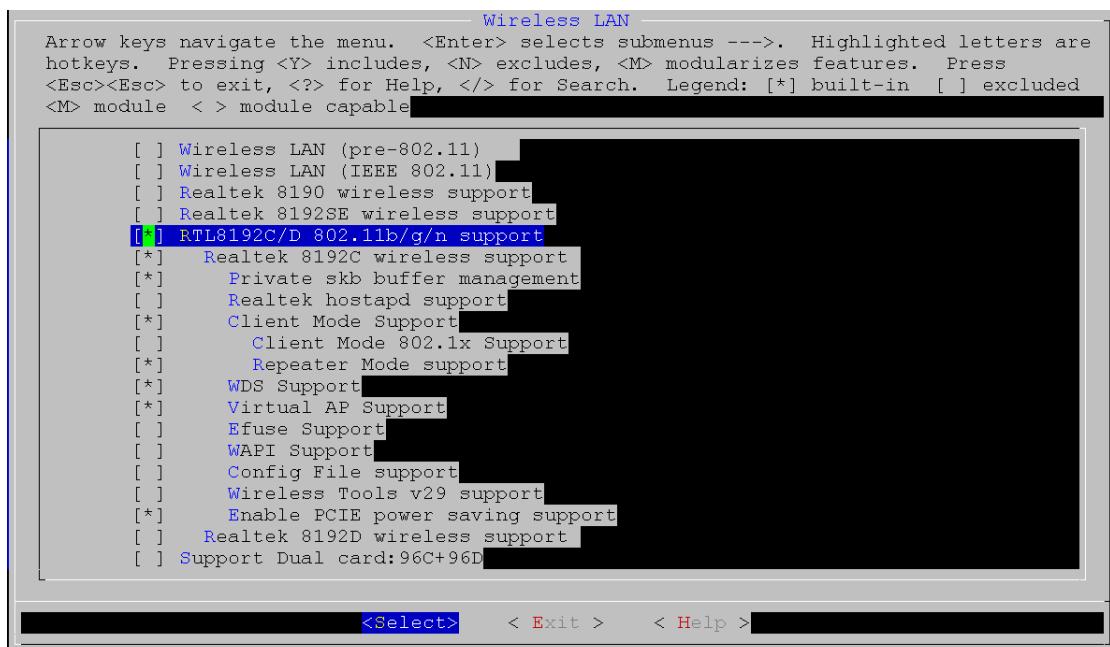
1) Select SDK for Pocket AP 92C:

Selected Target of SDK --->

And selected “11nPocket_Router”



2) Enable 92C Driver

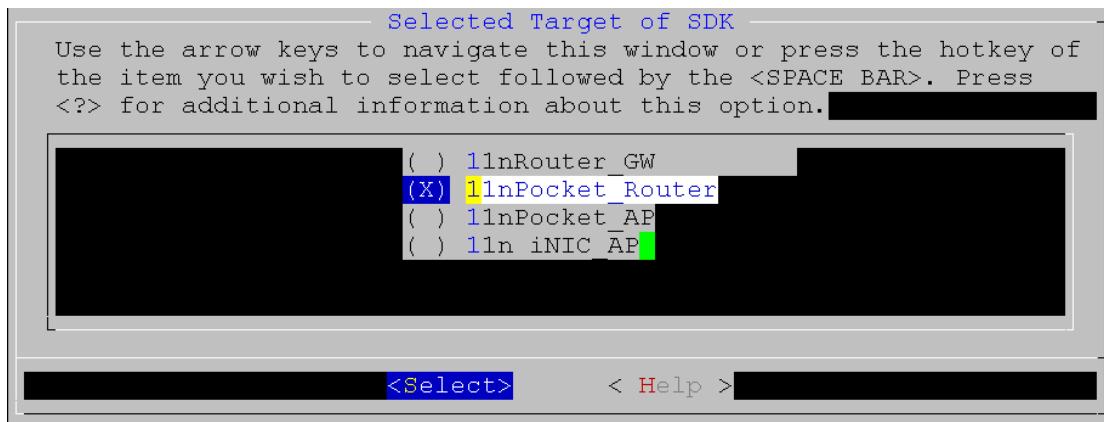


4.18.2 92D support for Pocket AP SDK

1) Select SDK for Pocket AP 92D:

Selected Target of SDK --->

And selected “11nPocket_Router”



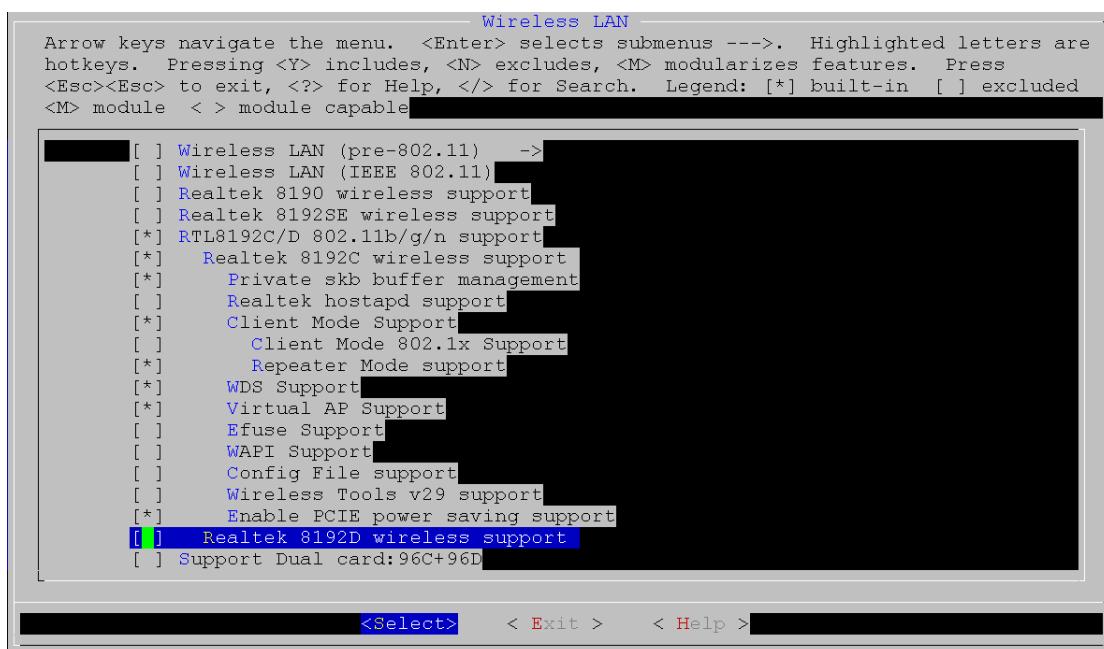
2) Enable 92D Driver

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 [REDACTED]
[*] 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 [REDACTED]
[*] 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
[*] RTL8192D 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.

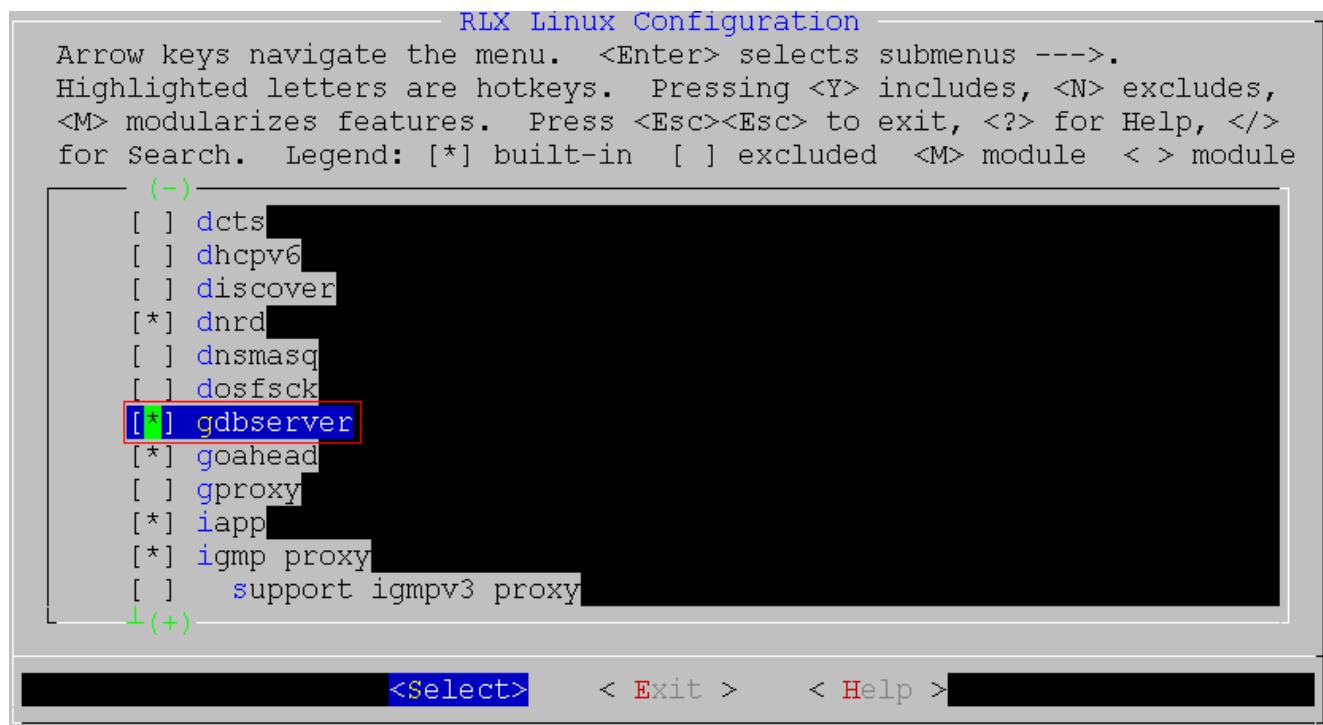
4.19 GDB server support

4.19.1 Enable GDB server

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

Menuconfig:

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



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/rlxlinux-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/rlxlinux-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-striped 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

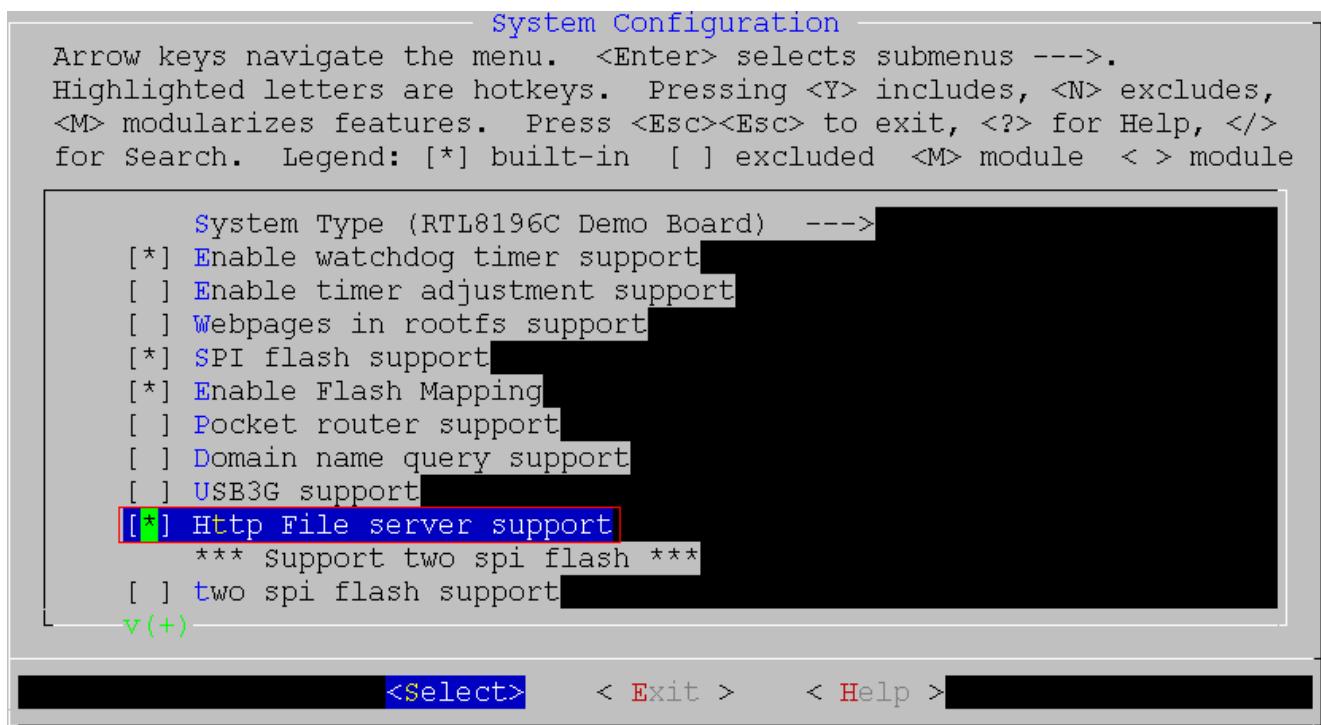
1) Kernel configure for HTTP File Server

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
```

Device Drivers --->

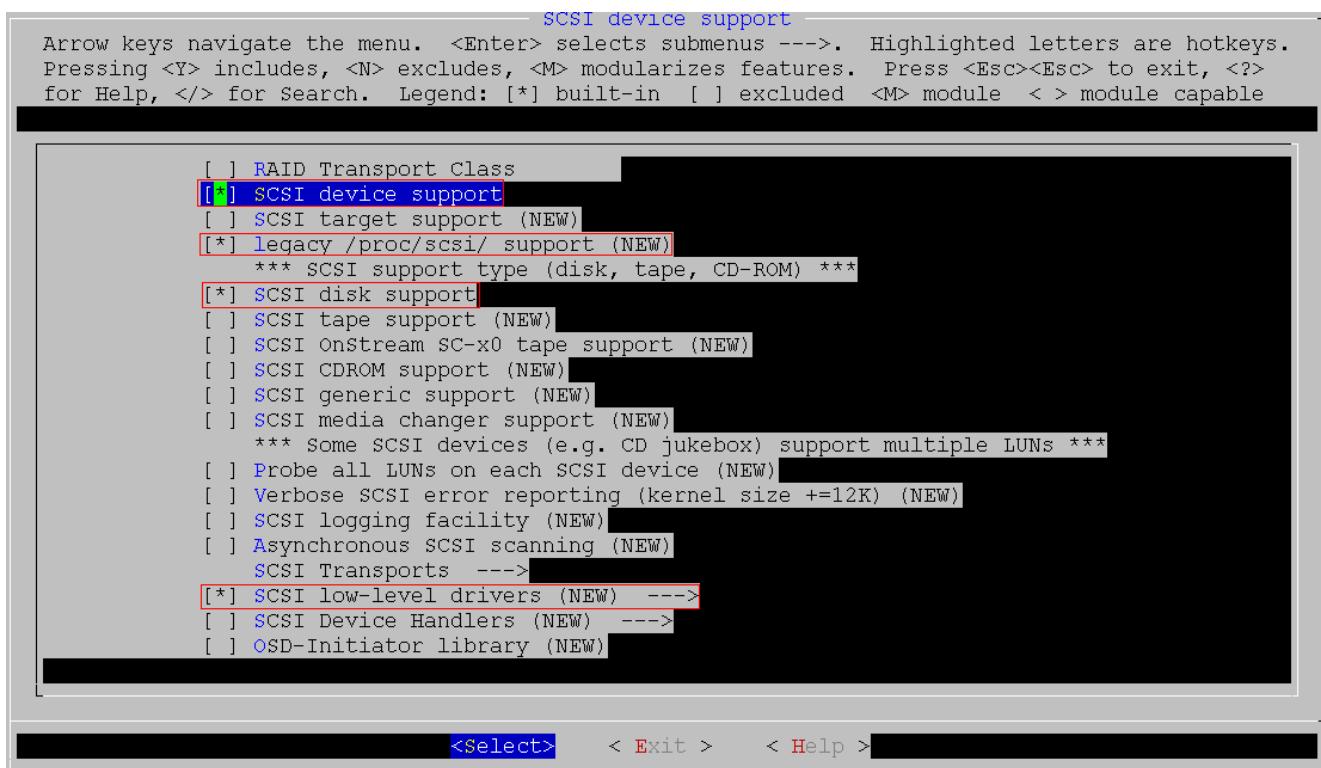
SCSI device support --->

[*] SCSI device support // Selected

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

[*] SCSI disk support // Selected

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



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 < > 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 < > module capable

^ (-)
[ ] 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 MAUSE-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 journalling 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)

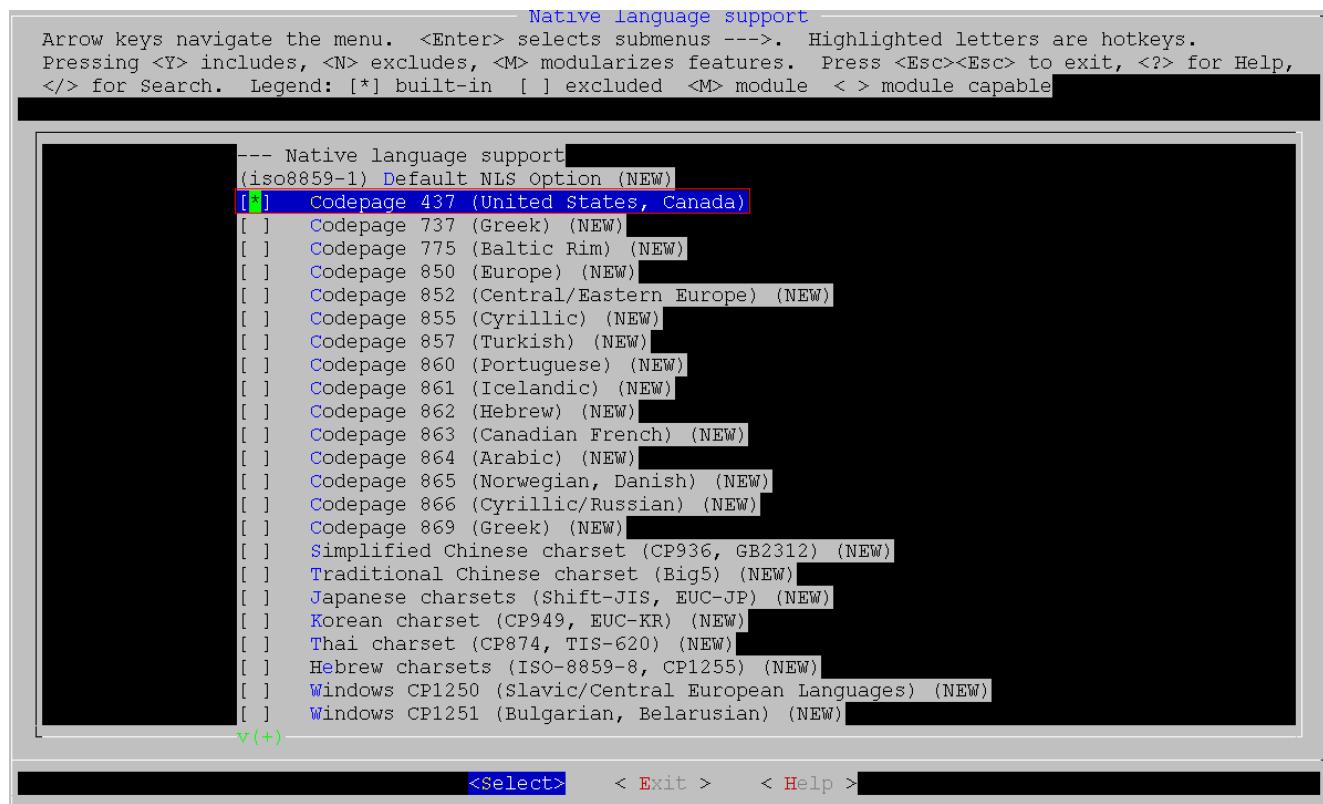
```

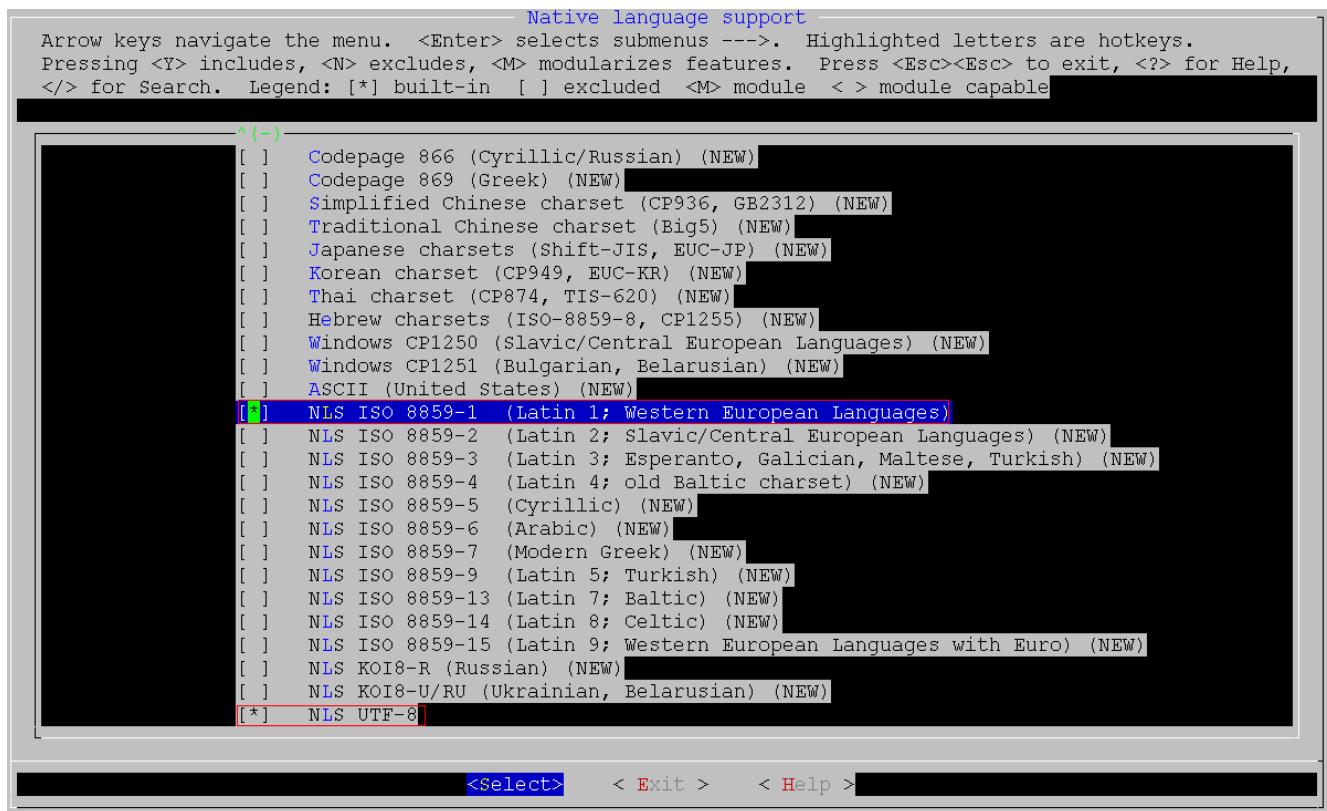
<Select> < Exit > < Help >

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



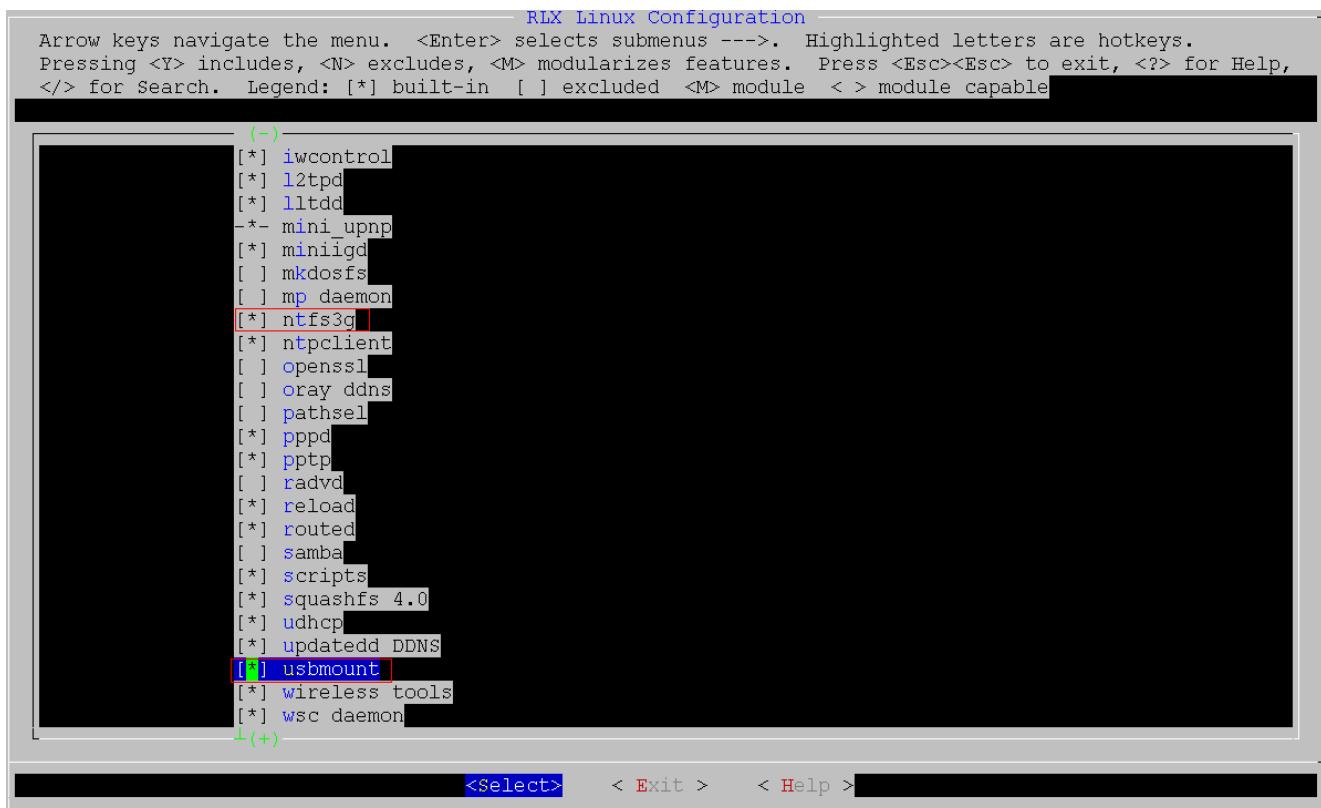


2) Application configure for HTTP File Server

make users_menuconfig // To configure application settings

Menuconfig:

- [*] ntfs3g // selected
- [*] usbmount // selected



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	-	Remove
abc/	28-Oct-2010 18:18:14	-	Remove
1.txt	20-Oct-2010 15:05:14	25.68K	Remove

Select File: [瀏覽...](#) [Upload](#)

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:

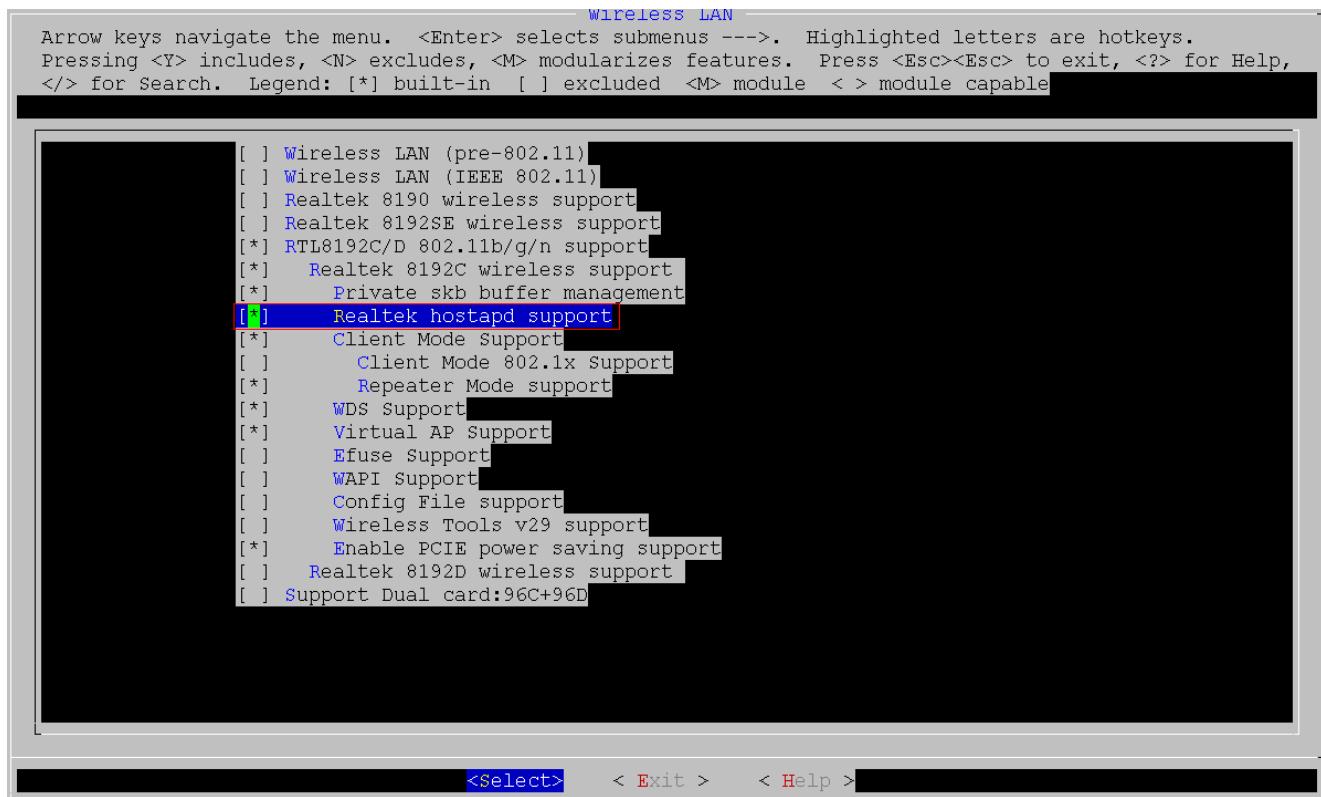
```
# sysconfinit $*  
  
echo 1 > /proc/rtk_vlan_support  
  
echo "hostapd test ... "  
  
brctl addbr br0  
  
brctl addif br0 eth0  
  
brctl addif br0 eth2  
  
brctl addif br0 eth3  
  
brctl addif br0 eth4  
  
brctl addif br0 wlan0  
  
ifconfig br0 192.168.1.254  
  
ifconfig eth1 192.168.2.167  
  
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
```

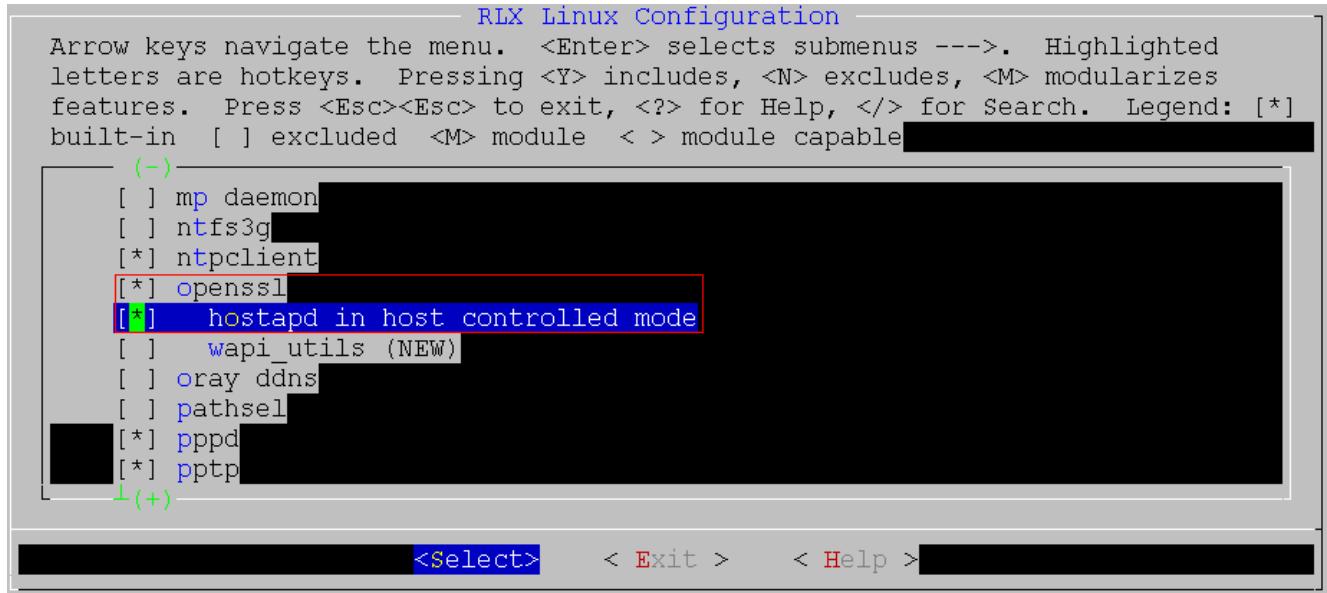
- ③ Copy net80211 folder (linux-2.6.30/include/net80211) under include the folder of toolchain,ex: toolchain/rsdk-1.3.6-5281-EB-2.6.30-0.9.30/include/net80211.

Step 2, configure kernel and application.

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



- ② Select "openssl" & "hostapd in host controlled mode" at users menuconfig.



Step 3, test hostapd.

① Test wpa2-psk

Firstly at AP console input command as follows:

```
killall hostapd
hostapd /tmp/hostapd.test_new_conf &
```

Secondly wireless client connect to our AP (SSID: Realtek_HAPD) using wpa2-psk(password: 1234567890123).

② Test WPS push button

Firstly at AP console input command as follows:

```
killall hostapd
hostapd /tmp/hostapd.test_new_conf &
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 /tmp/hostapd.test_new_conf &
hostapd_cli wps_pin any <pin number>
```

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

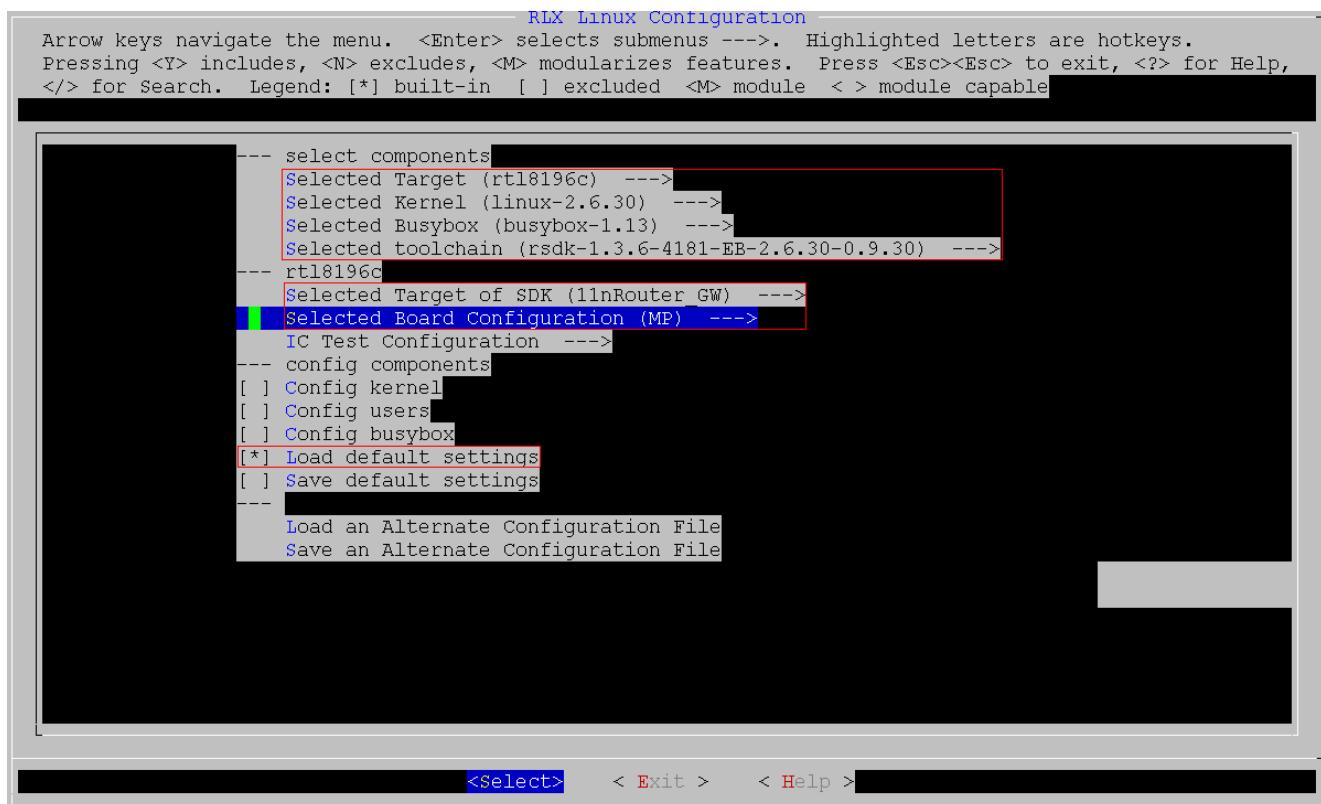
Step 4, customize hostapd.

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

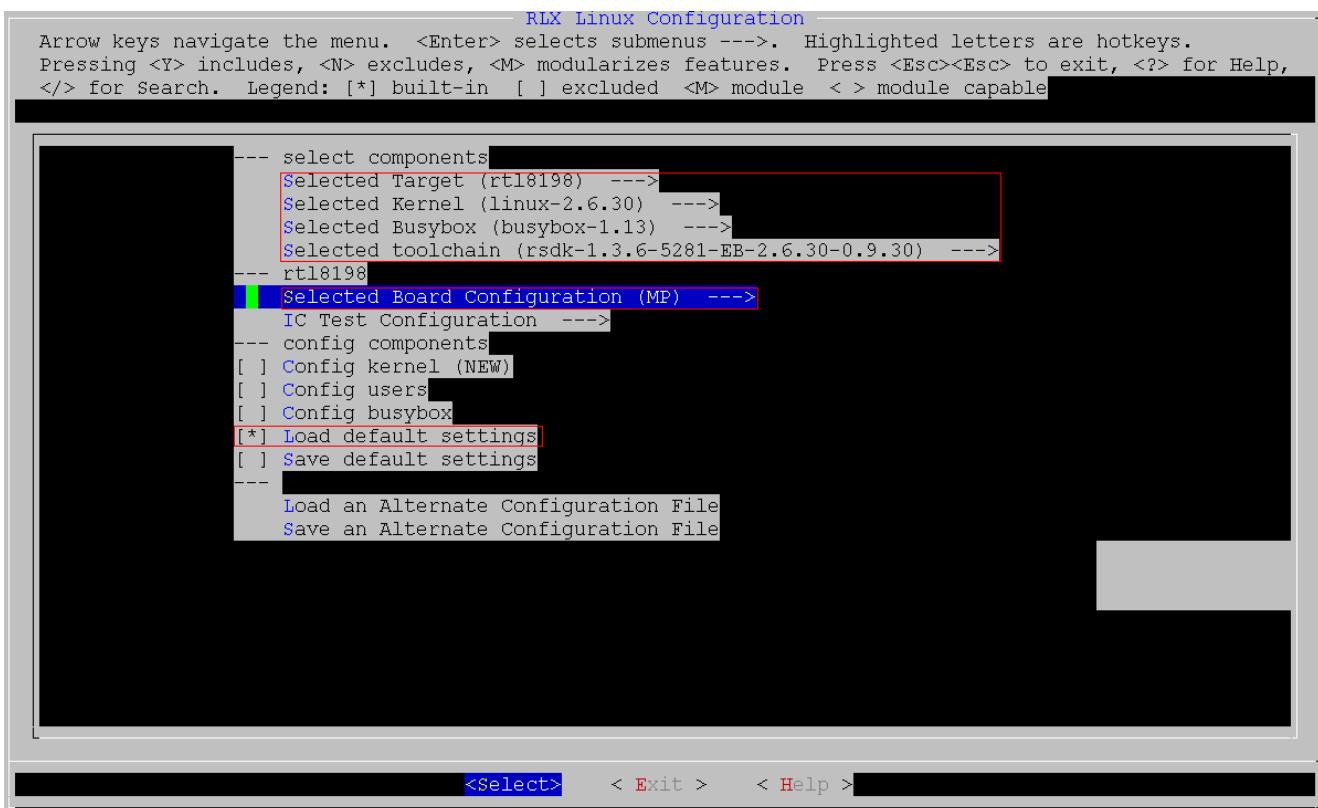
For RTL8196C MP, use “*make menuconfig*” to configure RTL8196C SDK settings as follows.



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, use “*make menuconfig*” to configure RTL8198 SDK settings as follows.

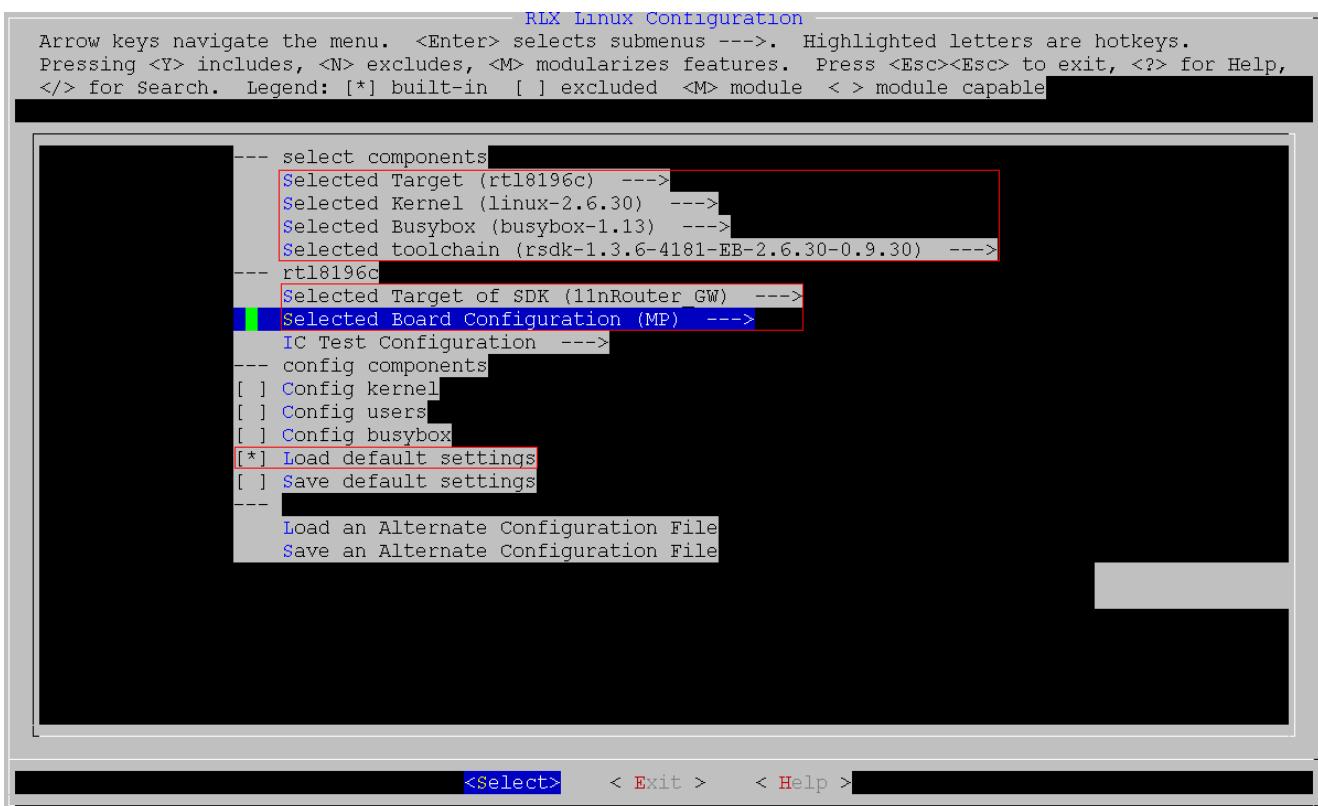


After configuring the settings as above, please refer to section 4.18 and configure the right wireless driver.

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.

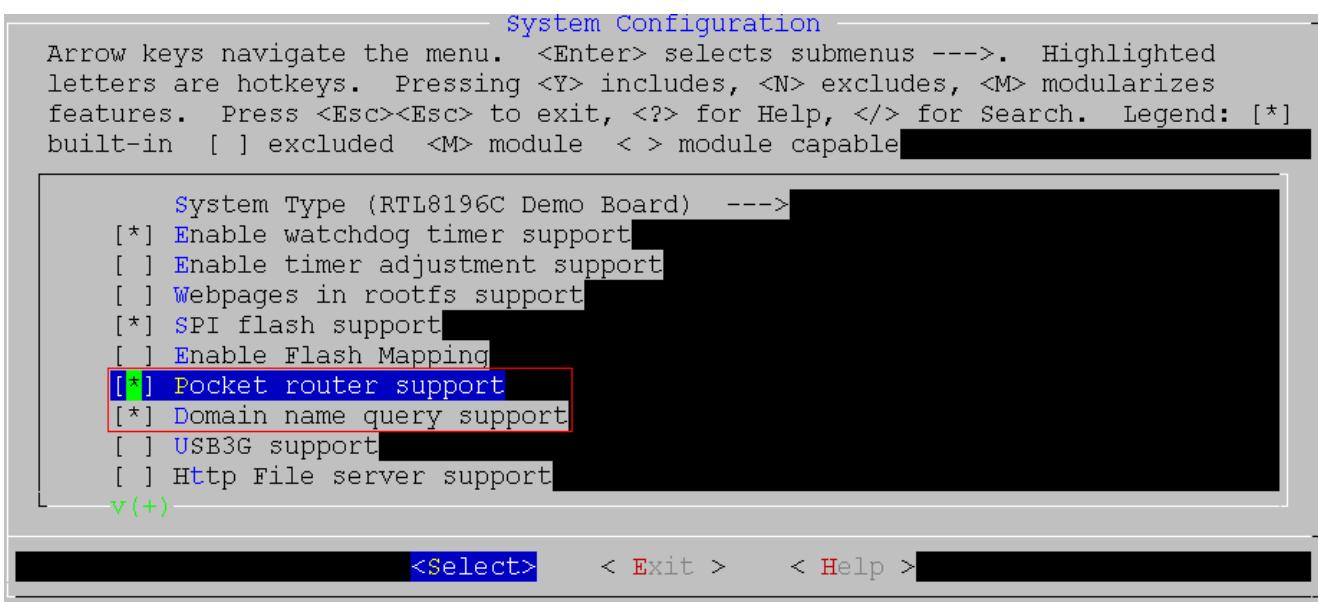


Step 2, use “*make linux_menuconfig*” to config kernel settings as follows.

System Configuration --->

[*] Pocket router support // Selected

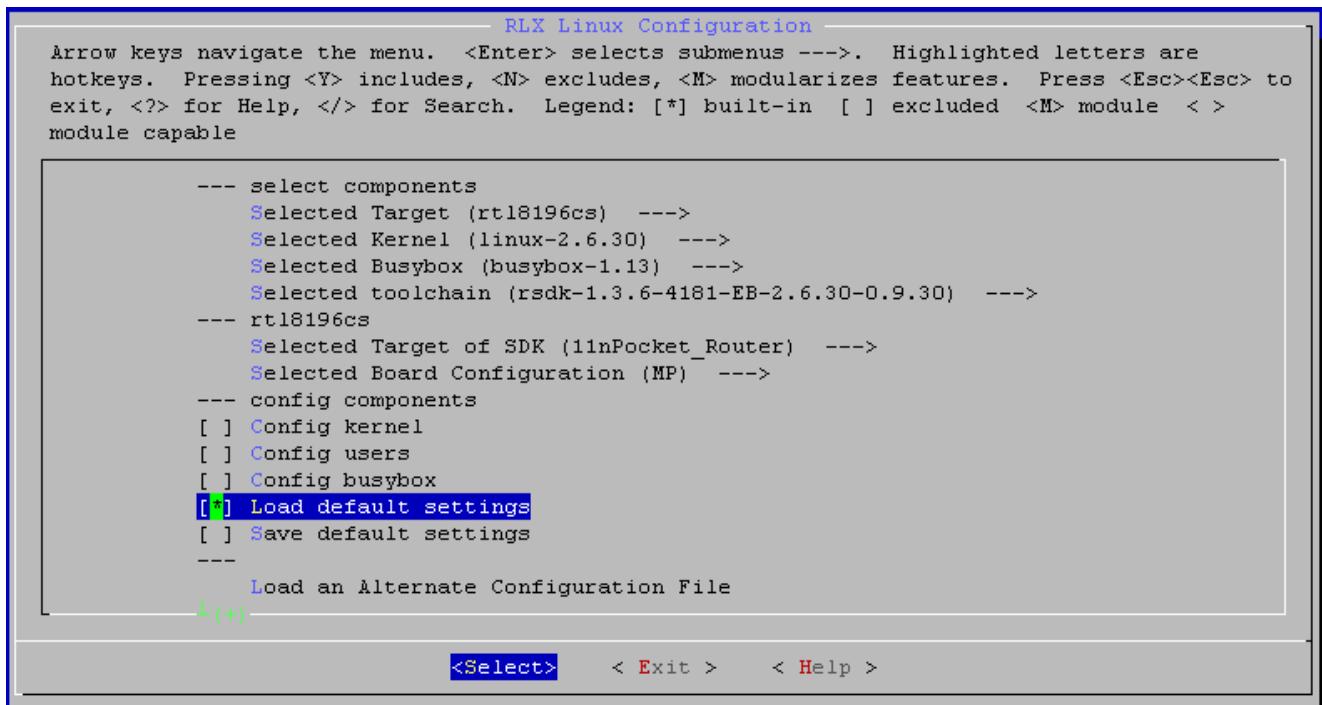
[*] Domain name query support // Selected



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, use “*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(+)

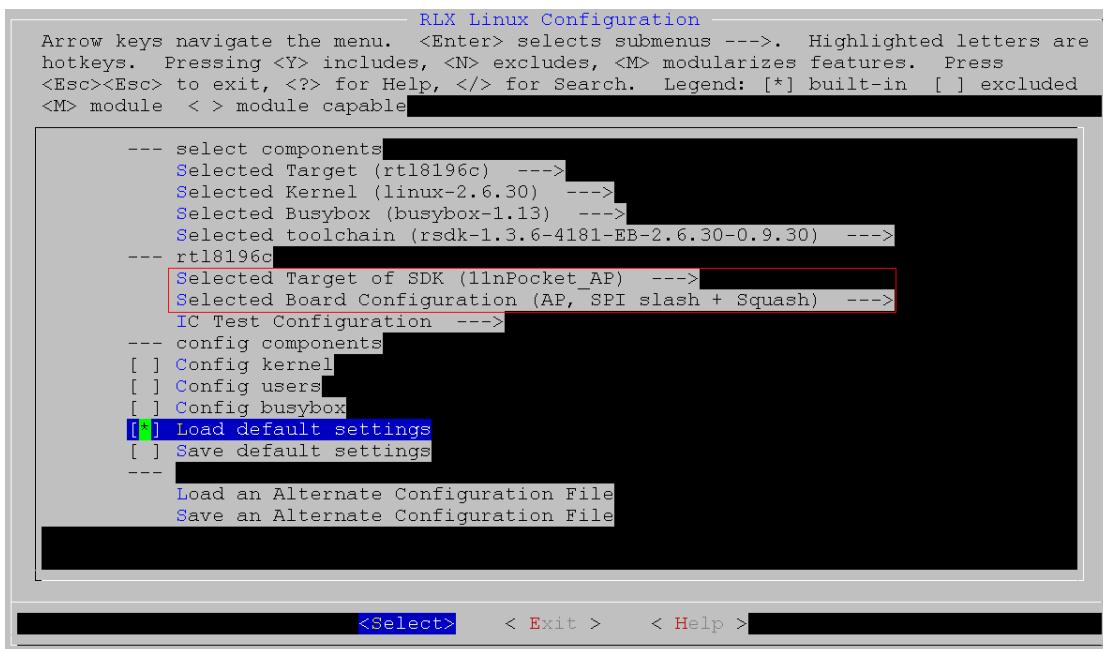
<Select> < Exit > < Help >
```

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 for submenus, and hotkeys like Y, N, M, Esc, and Help. Below the header, the menu lists several configuration 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) --->
- Selected Board Configuration (AP - SPI flash, Squashfs) ---> (This option is highlighted with a red border.)
- IC Test Configuration --->
- Config components
 - [] Config kernel
 - [] Config users
 - [] Config busybox
- [] Load default settings (This option is highlighted with a green border.)
- [] Save default settings

At the bottom of the menu, there are two additional options:
Load an Alternate Configuration File
Save an Alternate Configuration File

At the very bottom of the window, there is a footer with navigation keys: <Select>, <Exit>, <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:

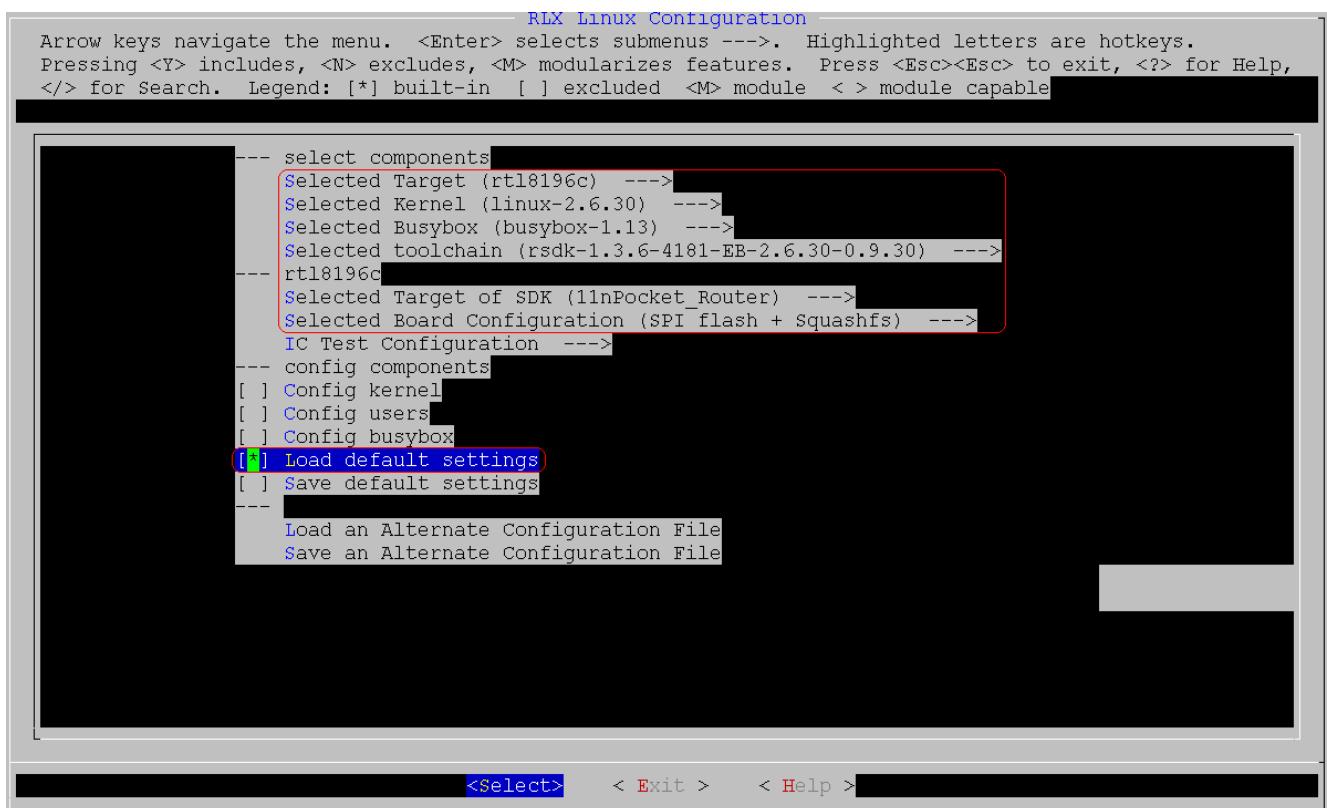
The iwcontrol tool is also needed for WPS. Please make sure the FIFO is created and opened in wscd, and iwcontrol use the same FIFO, otherwise wscd will hang.

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()

地址() http://192.168.1.254/ 转到() 链接()

Google() 开始() Windows() 书签() 已拦截 36 个 ABC 拼写检查() 翻译() 发送至() 设置()

WLAN Access Point

REALTEK

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 Show Client

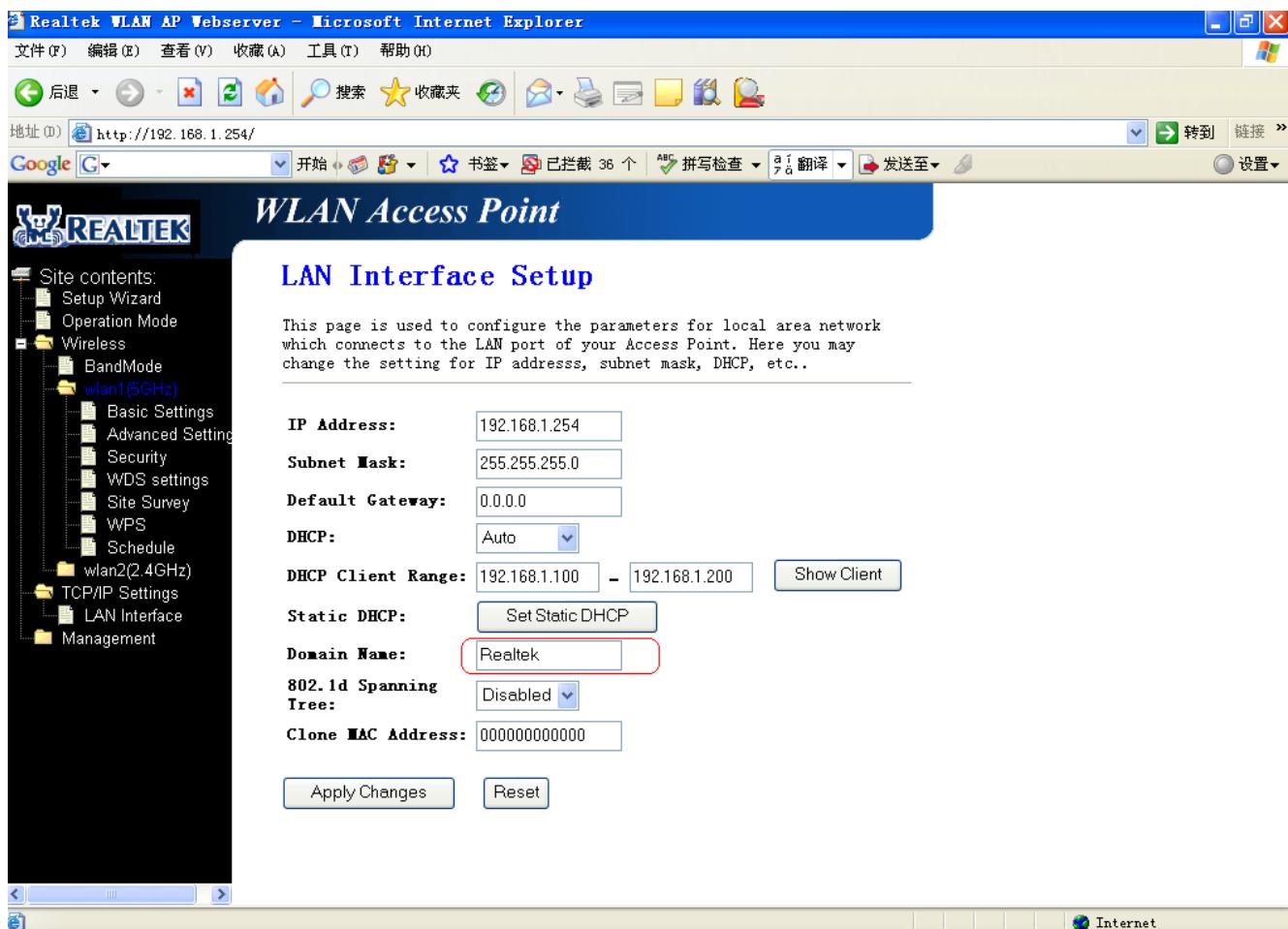
Static DHCP: Set Static DHCP

Domain Name: Realtek

802.1d Spanning Tree: Disabled

Clone MAC Address: 000000000000

Apply Changes Reset



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 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 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

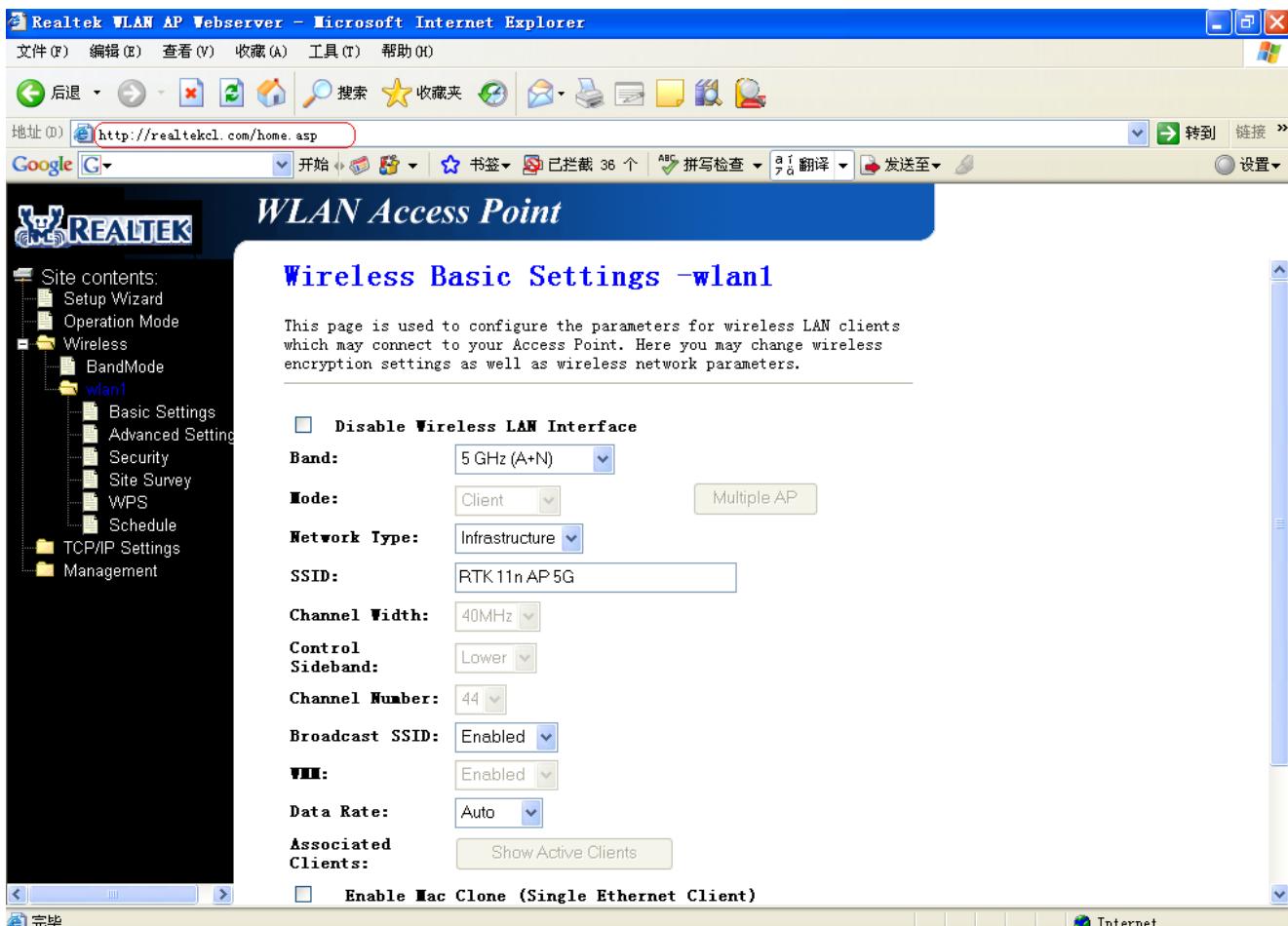
Static DHCP:

Domain Name: Realtek

802.1d Spanning Tree: Disabled

Clone MAC Address: 000000000000

完毕 Internet



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

4.28.1 Bootcode menuconfig

In order to enable the Dual Image support , you need to select the item "Support Flash DualBank" and enter the offset of the second bank in the bootcode 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 bootcode menuconfig as follows.

[*]Support Flash DualBank

(400000) Second Bank offset

4.28.2 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

*** 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.)

4.28.3 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 need also support EEE feature when do 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,

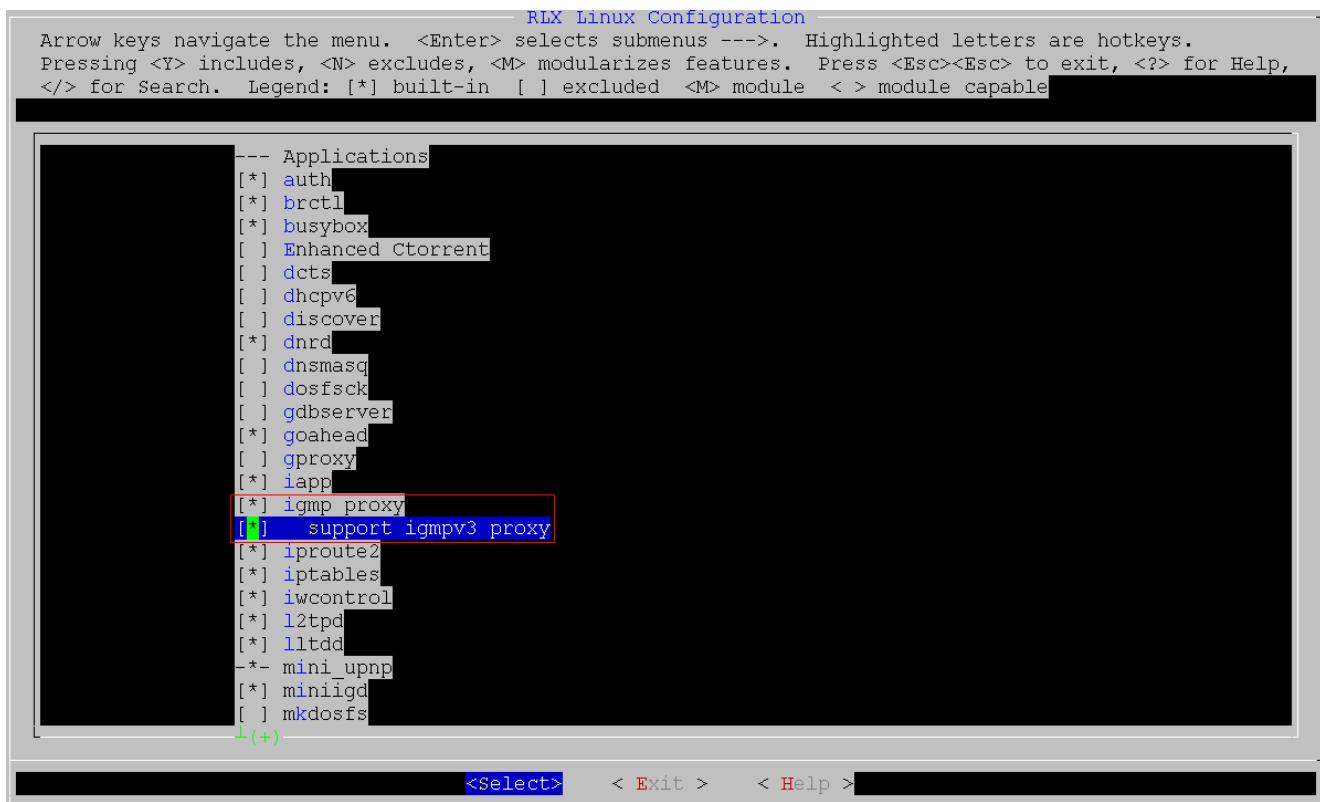
Support 128 hardware multicast entries.

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.



4.30.3 How to enable IGMP/MLD snooping

vim linux-2.6.30/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 multicate

```
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 --->

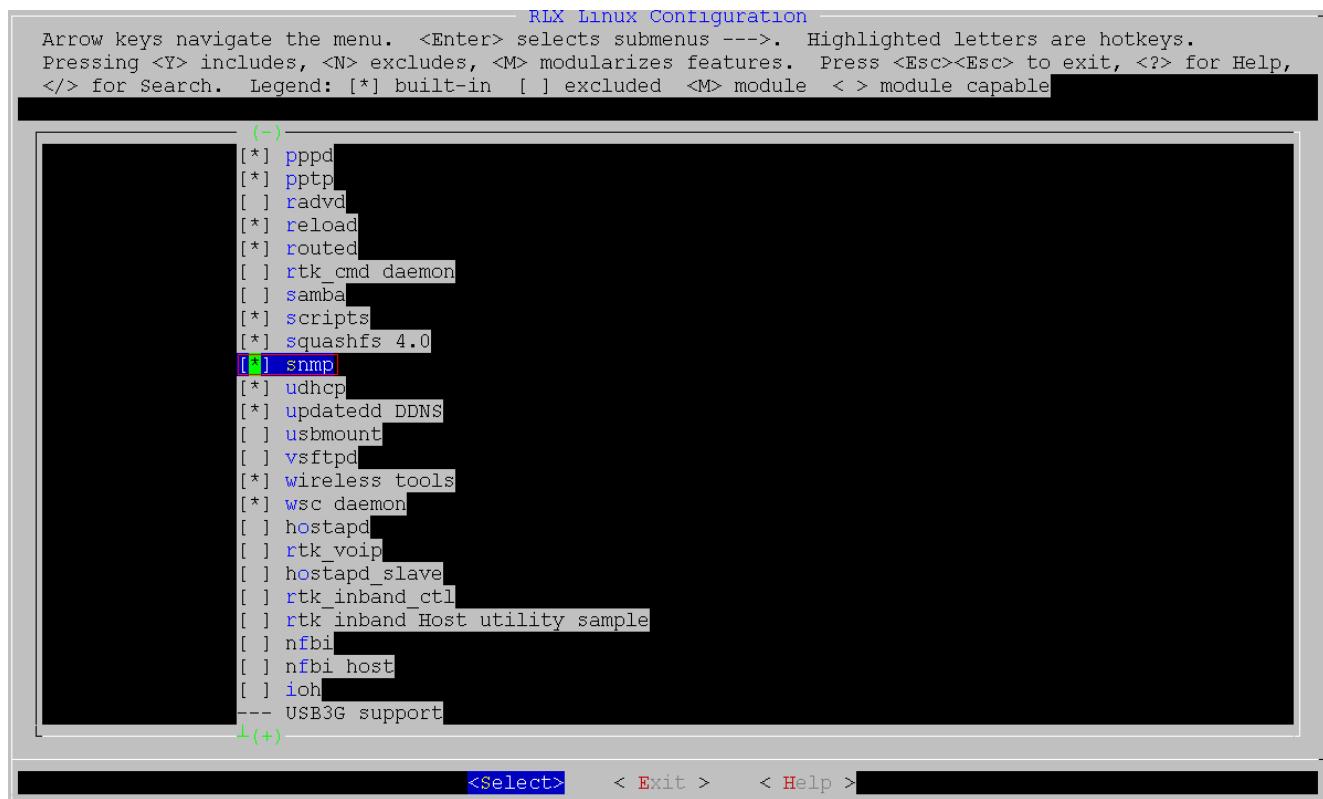
Hardware Features Selection (Enable RTL Hardware Multicast Only)

Or Hardware Features Selection (Enable RTL Hardware NAPT)

4.31 SNMP support

4.31.1 Applications configure

```
make users_menuconfig //To configure applications
```



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 linux-2.6.30/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 bar has the title "WLAN Access Point". On the left, there's a sidebar with "Site contents:" and a list of options: Setup Wizard, Operation Mode, Wireless, TCP/IP Settings, Firewall, QoS, SNMP (which is selected), and Management. The main content area is titled "SNMP Setting". It contains a sub-header "SNMP is a application for network managment". Below that is a section with a checked checkbox labeled "Enable SNMP". There are five input fields with the following values: Name : Realtek, Location : AP, Contact : Router, Read/Write Community : private, and Read-Only Community : public. At the bottom 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.

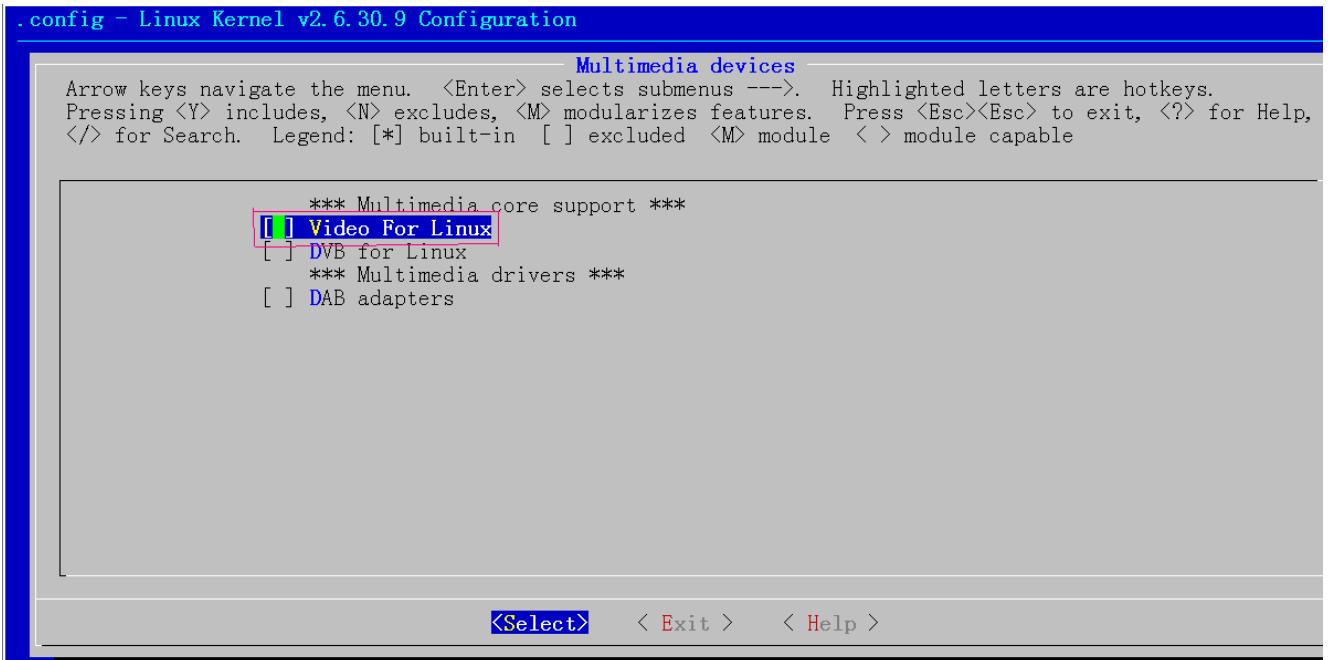
```
make linux_menuconfig // To configure linux kernel settings
```

Menuconfig:

Device Drivers --->

Multimedia devices --->

[] Video For Linux // Not selected



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

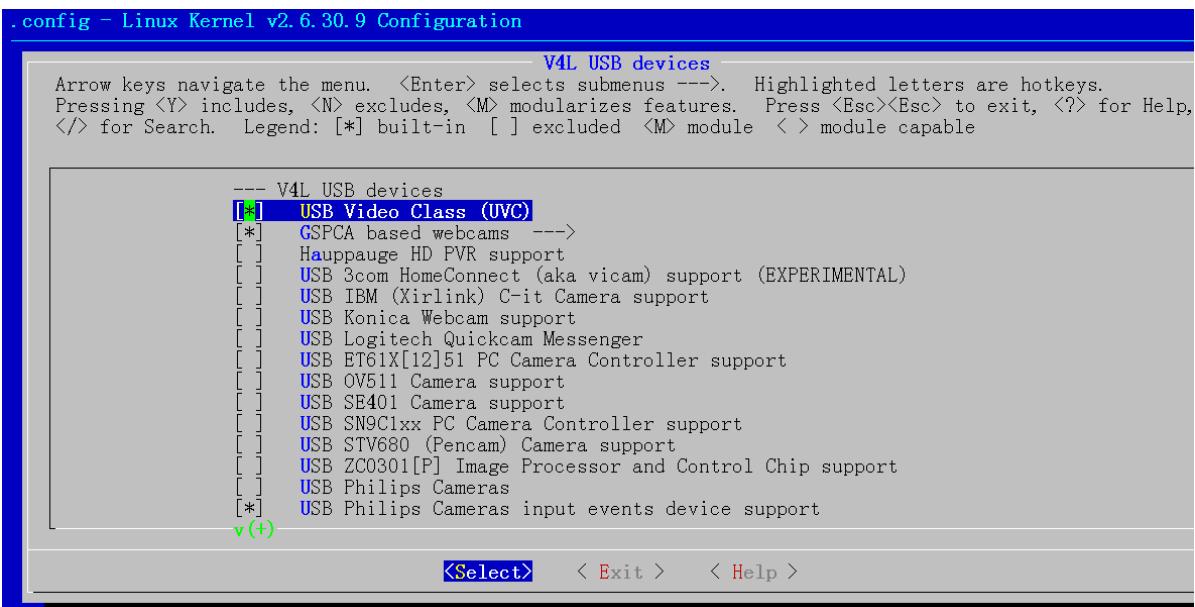
make linux_menuconfig // To configure linux kernel settings

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



4.33 Telnetd and Login support in busybox

Enter the busybox menu, and then make menuconfig.

make menuconfig // To configure bootloader settings

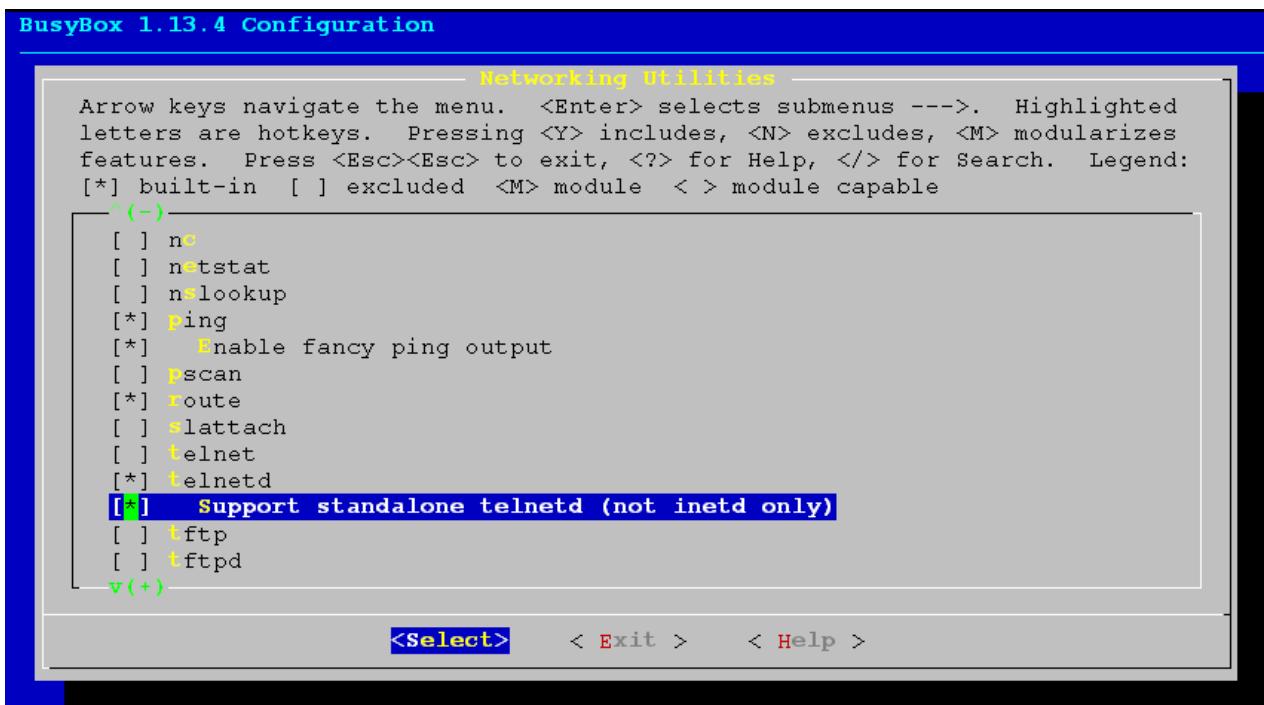
4.33.1 telnetd support menuconfig:

menuconfig:

Networking Utilities--->

[*] telnetd // selected

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



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

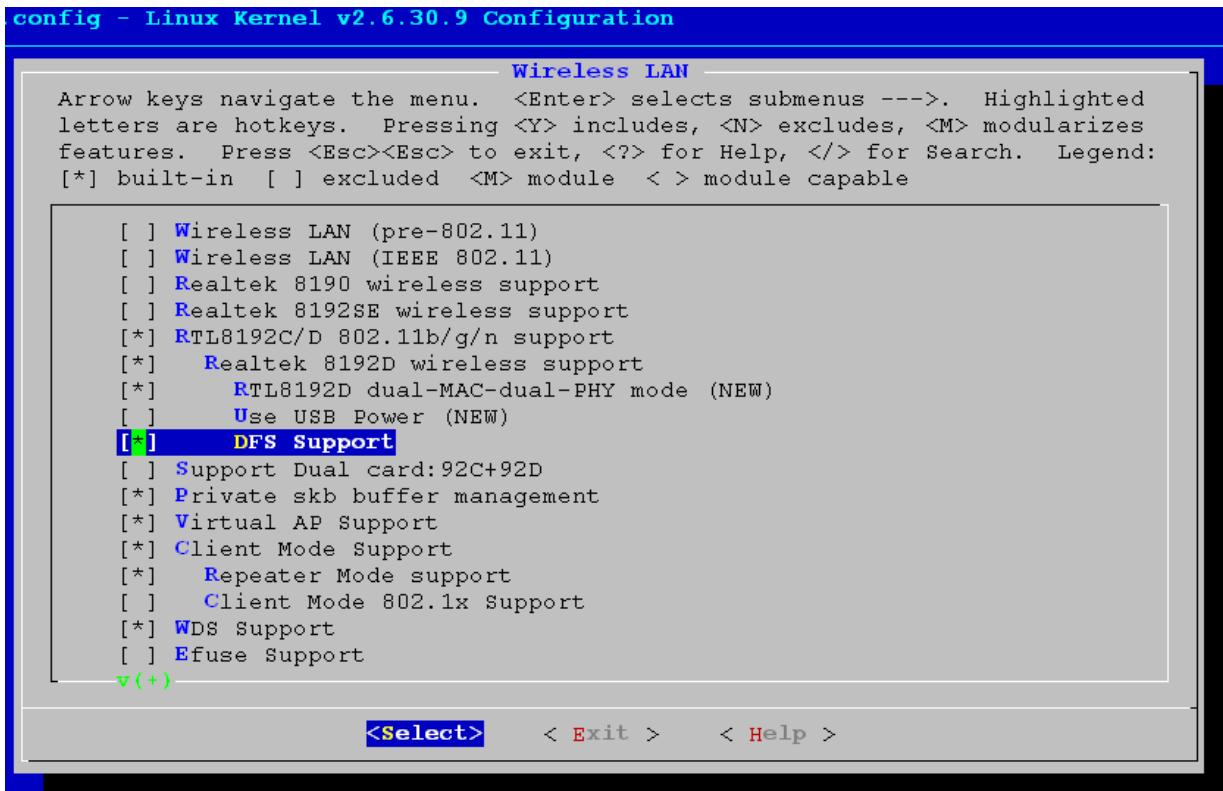
1) kernel configure

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



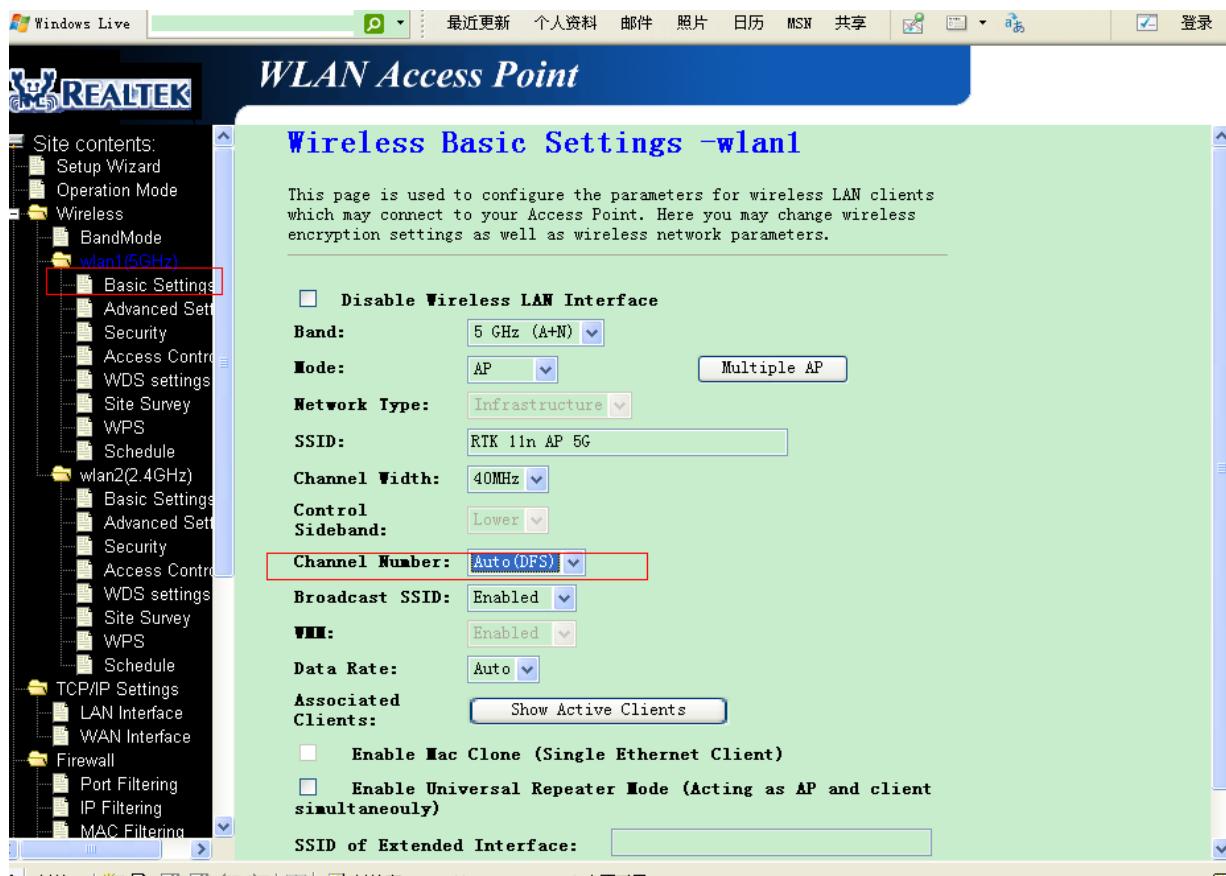
- 2) To obey regulation, DFS channels can ONLY be selected by auto-channel selection. The user can see “Auto (DFS)” on the channel column on web UI..

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.

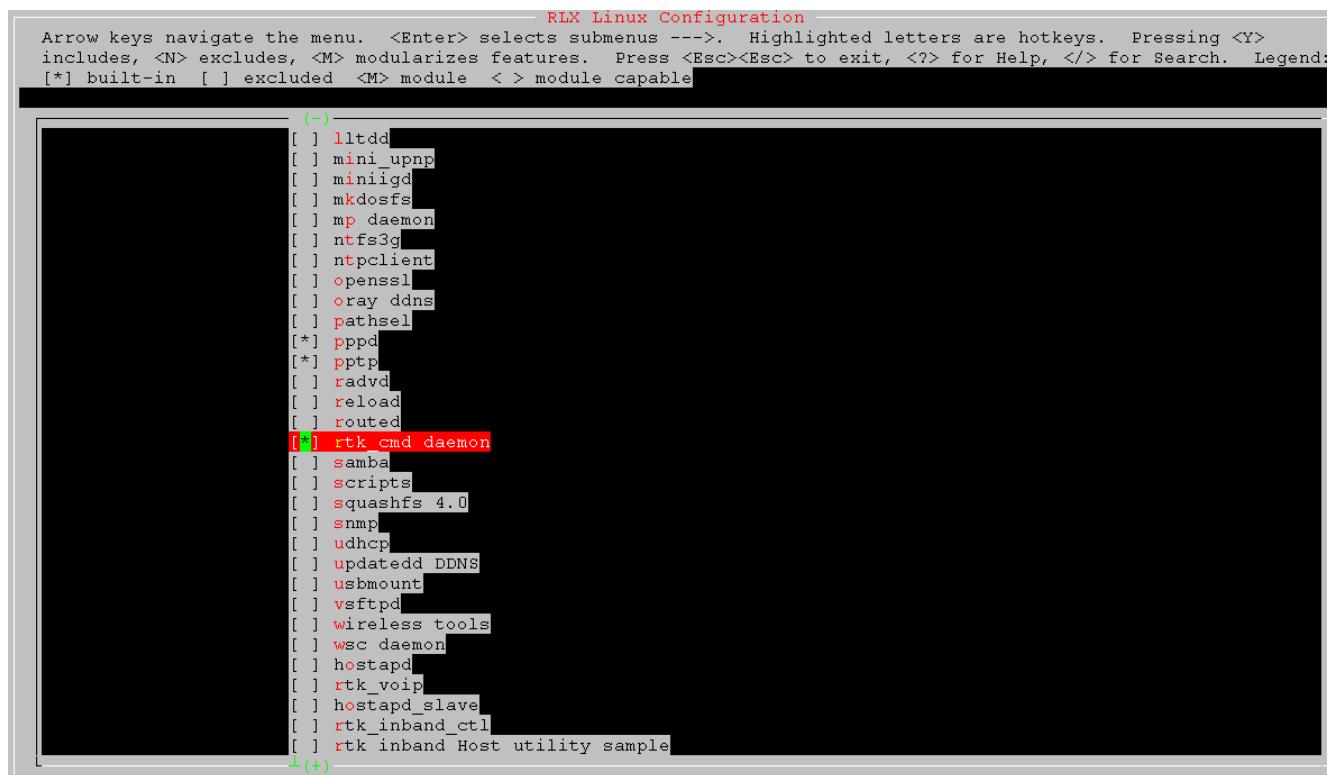
4.35.2 How to enable/disable Fastpath Filter feature

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



- 2) Enable fast filter support

make linux_menuconfig

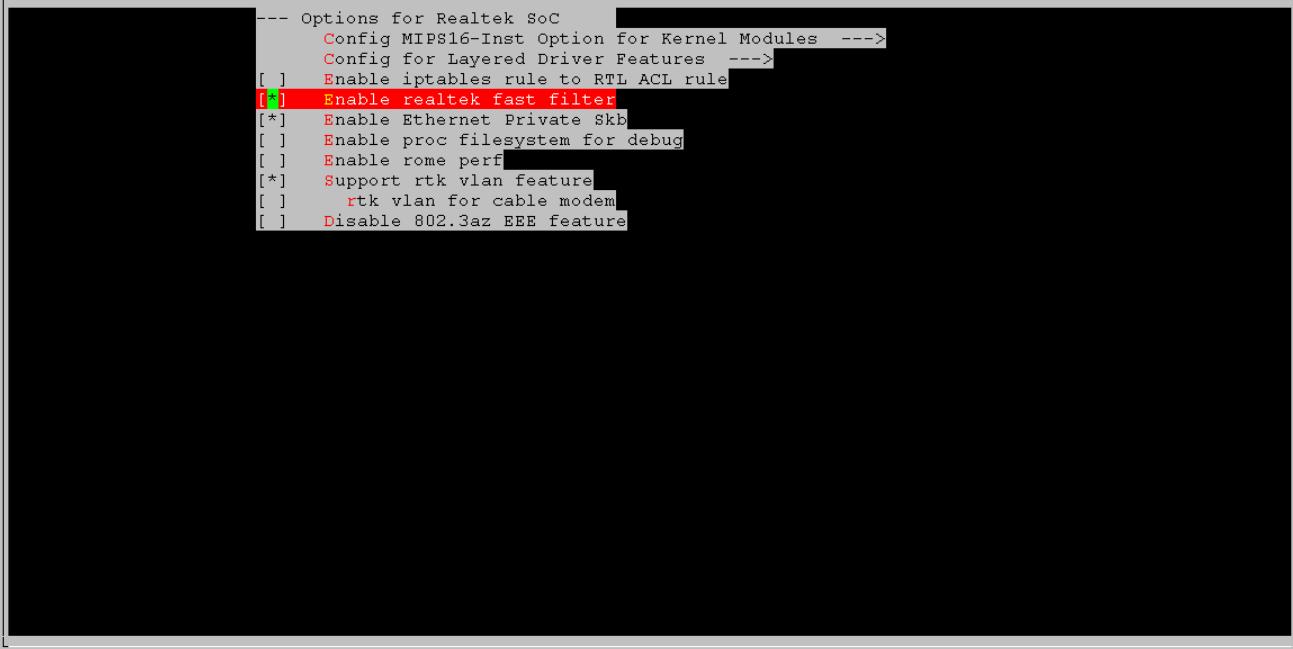
Device Drivers --->

[*] 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 MIFS16-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

```
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 upd

```
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
```

```

config - Linux Kernel v2.6.30.9 Configuration
                                               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

[ *] Packet socket
[ ]   Packet socket: mmapped IO
[*] Unix domain sockets
[ ]   PF_KEY sockets
[*] TCP/IP networking
[ *]   IP: multicasting
[ ]   IP: advanced router
[*]   Realtek IPTables Fast Path
[ *]   Realtek Fastpath SPI(stateful packet inspection) support
[ ]     IP: kernel level autoconfiguration
[ ]     IP: tunneling
[ ]     IP: GRE tunnels over IP
[*]     IP: multicast routing
[ ]       IP: PIM-SM version 1 support
[ ]       IP: PIM-SM version 2 support
[ ]     IP: ARP daemon support (EXPERIMENTAL)
[ ]     IP: TCP syncookie support (disabled per default)
[ ]     IP: AH transformation
[ ]     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 --->
v(+)

<Select>    < Exit >    < Help >

```

4.37 ASAL support

4.37.1 What's ASAL

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

4.37.2 How to enable/disable ASAL feature

First, Enable USB support (see section 4.2).

Then, do

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

Menuconfig:

Device Drivers --->

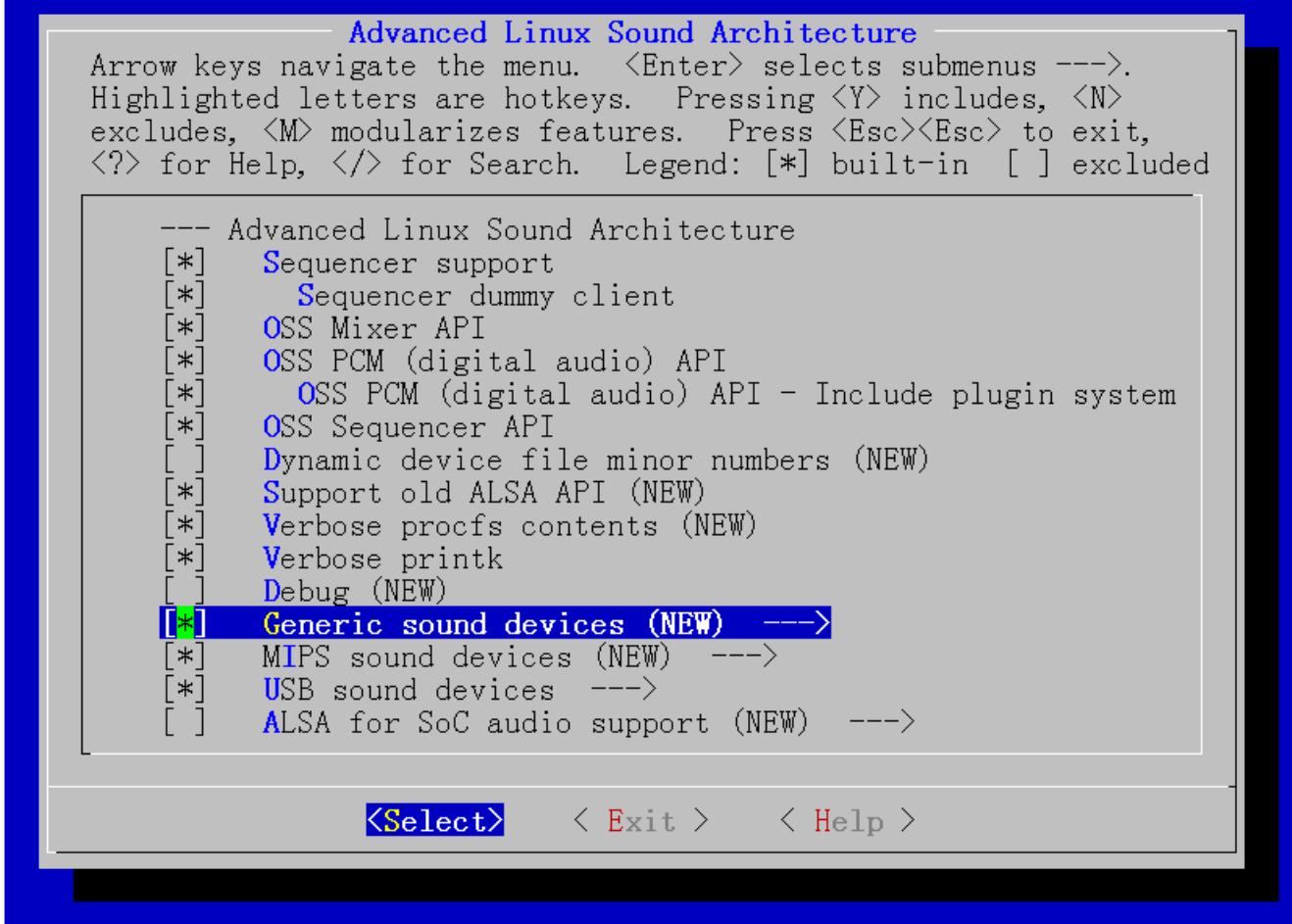
- <*> Sound card support --->
- <*> Advanced Linux Sound Architecture --->
- [*] Sequencer support
 - [*] Sequencer dummy client
- [*] OSS Mixer API
- [*] OSS PCM (digital audio) API - Include plugin system
- [*] OSS Sequencer API
- [*] Support old ALSA API
- [*] Verbose procfs contents

```

[*] 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

```

.config - Linux Kernel v2.6.30.9 Configuration



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

The screenshot shows a terminal window displaying the kernel configuration menu for the Realtek SoC. The title bar says "Options for Realtek SoC". Below it is a detailed description of the menu navigation. The main list contains several options, with the "Enable rome perf" option highlighted by a red box and marked with a green asterisk (*). At the bottom of the screen, there are navigation keys: <Select>, < Exit >, and < Help >.

```
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:linux-2.6.30/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 totalNum</i>	<i>Average</i>
[89]	<i>ip_finish_output3</i>	2001104	135 14822
		231705	135 1716
		30048	135 222
		1308804	135 9694

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 (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 (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
- [] dhcpcv6
- [] 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("pppConnectType");

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 rename 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 usful 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, <> 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, <> 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
[] BOHMELES 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

```

[] discover
[*] dnrd
[] dnsmasq
[] dosfsck
[] gdbserver
[*] goahead
[] boa
[*] uWiFi
[] gproxy
[*] iapp

```

[\*] nmbs // selected

```

[*] wsc daemon
[] hostapd
[] rtk	voip
[] hostapd_slave
[] rtk_inband_ctl
[] rtk inband Host utility sample
[] nfbi
[] nfbi host
[] ioh
[*] nmbs

```

#### 4.40.3 WEB GUI Manual

Home page:

**Realtek uWiFi**

[!\[\]\(cc87be3f83672c4af2aa12bb0512a05d\_img.jpg\) Settings](#)

---

**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.

**Realtek uWiFi**

[!\[\]\(646b5de9823abd8ce77bb056c8158b8e\_img.jpg\) 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 |

---

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

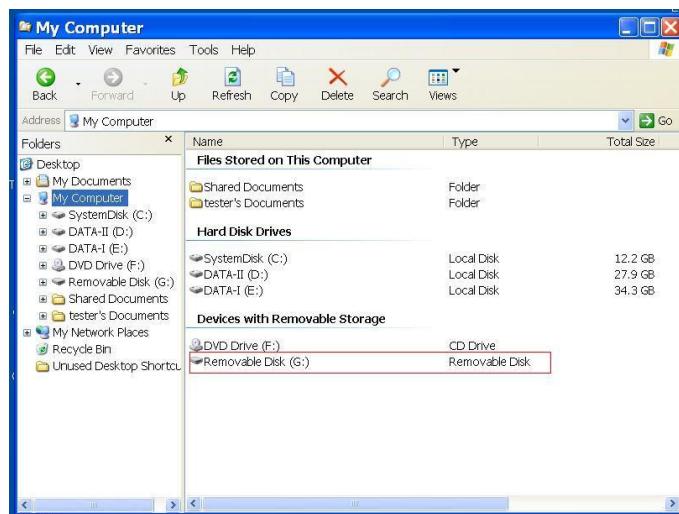
If the user selects share method “uWiFi”, then uWiFi Manager can see storage device icon. uWiFi Manager can be used to “Connect” if the current state is Disconnect.



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

---

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) |

---

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 web interface. The title bar says "Realtek uWiFi". Under "Shared Partitions", there is a link to "sda1". In the "Storage Device Management" section, there is a table:

| USB Device                   | Share Method                                                      |
|------------------------------|-------------------------------------------------------------------|
| JetFlash Mass Storage Device | <input checked="" type="radio"/> Http <input type="radio"/> uWiFi |

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:  [瀏覽...](#)

### 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 addresss, subnet mask, DHCP, etc..

|                       |                                                                                                                                    |
|-----------------------|------------------------------------------------------------------------------------------------------------------------------------|
| IP Address:           | <input type="text" value="192.168.1.254"/>                                                                                         |
| Subnet Mask:          | <input type="text" value="255.255.255.0"/>                                                                                         |
| Default Gateway:      | <input type="text" value="0.0.0.0"/>                                                                                               |
| DHCP:                 | Auto <input type="button" value="▼"/>                                                                                              |
| DHCP Client Range:    | <input type="text" value="192.168.1.100"/> - <input type="text" value="192.168.1.200"/> <input type="button" value="Show Client"/> |
| DHCP Lease Time:      | <input type="text" value="480"/> (1 ~ 10080 minutes)                                                                               |
| Static DHCP:          | <input type="button" value="Set Static DHCP"/>                                                                                     |
| Domain Name:          | <input type="text" value="Realtek"/>                                                                                               |
| LAN BIOS NAME:        | <input type="text" value="uWiFi"/>                                                                                                 |
| 802.1d Spanning Tree: | Disabled <input type="button" value="▼"/>                                                                                          |
| Clone MAC Address:    | <input type="text" value="000000000000"/>                                                                                          |

Users can use LAN BIOS Name to connect Realtek RTL819x Wireless Home Gateway.

The screenshot shows the "Realtek uWiFi" configuration page in a Windows Internet Explorer browser. The URL in the address bar is <http://uwifi/uWiFi.asp>. The main content area displays the "Realtek uWiFi" logo and a "Settings" icon. Under "Shared Partitions", it states "No shared partition available.". In the "Storage Device Management" section, there is a table with one entry: "USB Device" (JetFlash Mass Storage Device) and "Share Method" (Http, uWiFi). At the bottom of the page are "Apply Changes" and "Reset" buttons.

## 4.41 ALL NAT

### 4.41.1 what's Multiple wan(All Nat)

There are 2 network interface 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:

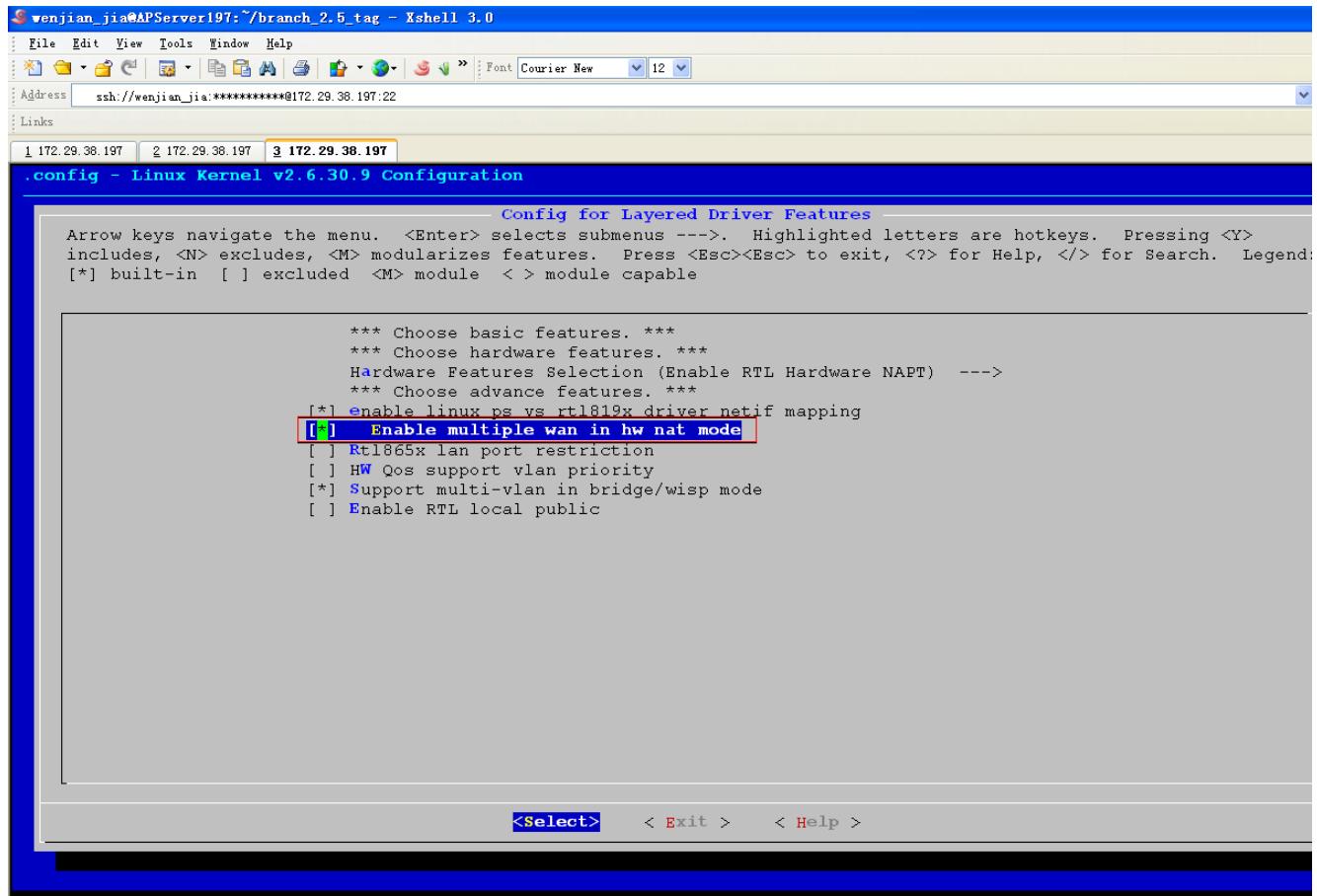
Device Drivers --->

[\*] Network device support --->

[\*] Options for Realtek SoC --->

Config for Layered Driver Features --->

[\*] Enable multiple wan in hw nat mode



### 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
 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 log print system for debug
[] Enable rome perf
[*] Support rtk vlan feature
[] rtk vlan for cable modem
[] add rtk vlan bridge feature
[] Disable 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

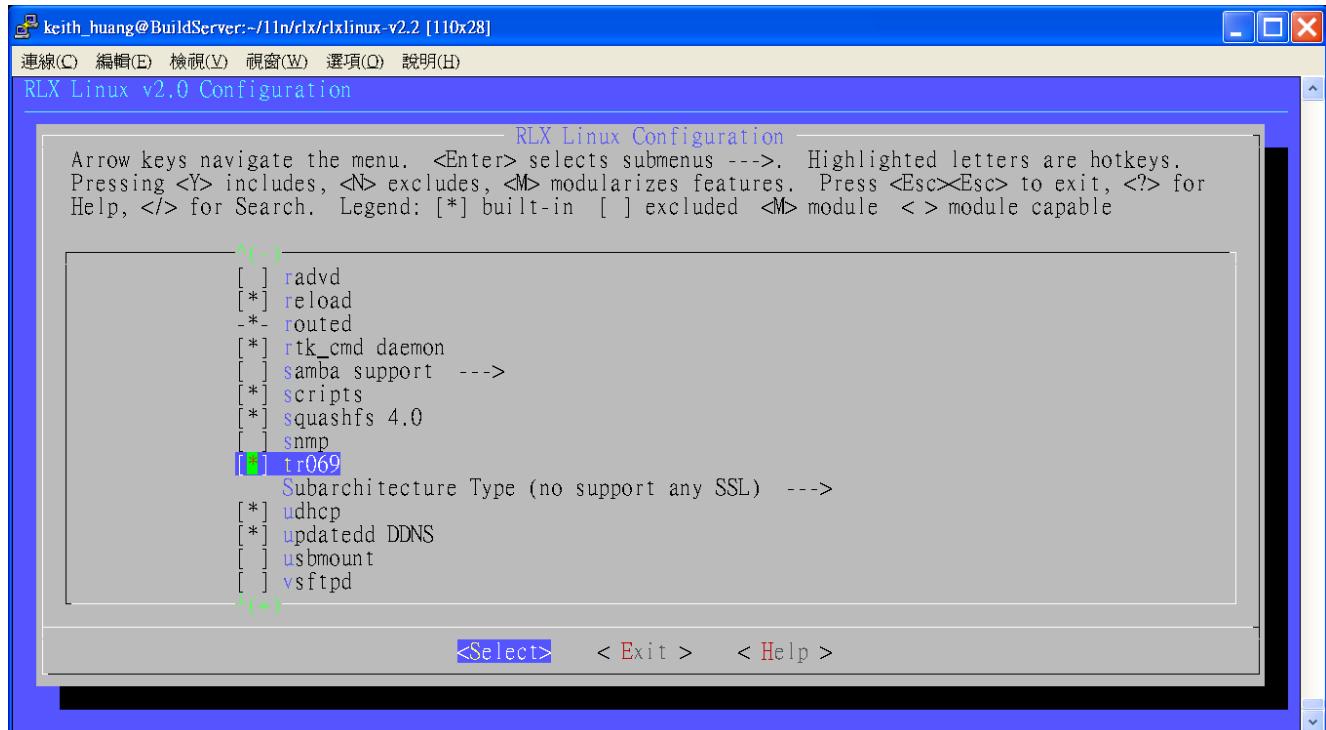
## 4.43 TR069 Support

### 4.43.1 How to enable proc filesystem for debug

make users\_menuconfig

Device Drivers --->

[\*] tr069



### 4.43.2. Proc illustration:

Modify related setting in WEB page.

## 4.44 ALG Support

### 4.44.1 DESCRIPTION

SDK support the following alg:

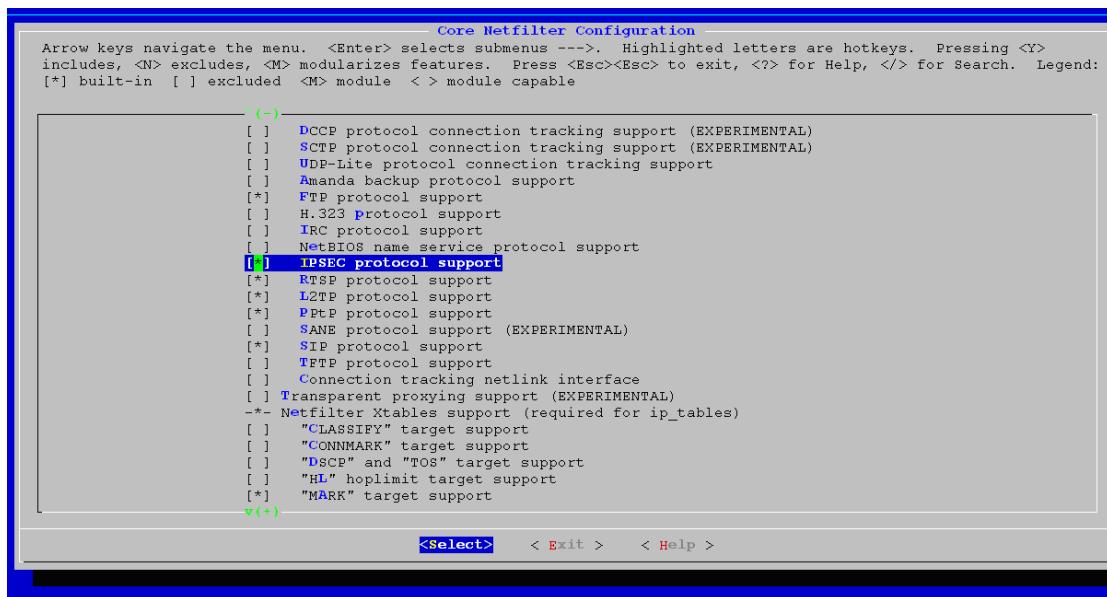
1. SIP;
2. FTP;
3. TFTP;
4. H323;
5. PPTP;
6. L2TP;
7. RTSP;

## 8. IPSEC;

### 4.44.2 How to enable ALG

*make linux\_menuconfig*

Take enable IPSEC ALG as an example:



Other ALG could be enable 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]

```

.config - Linux Kernel v2.6.30.9 Configuration
IP: 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

^(-)
[*] proc/sysctl compatibility with old connection tracking
[] IP Userspace queueing via NETLINK (OBSOLETE)
[*] IP tables support (required for filtering/masq/NAT)
[] "addrtype" address type match support
[] "ah" match support
[] "ecn" match support
[] "ttl" match support
[*] Packet filtering
[] REJECT target support
[] LOG target support
[] ULOG target support
[*] Full NAT
[*] MASQUERADE target support
[!] CONENAT target support
[] NETMAP target support
[] REDIRECT target support
[] Basic SNMP-ALG support
[*] Packet mangling
[] CLUSTERIP target support (EXPERIMENTAL)
[] ECM target support
[] TTL target support
[] raw table support (required for NOTRACK/TRACE)
[] ARP tables support

<Select> < Exit > < Help >

```

## 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

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 igmpv3 proxy
  - [\*] mld proxy
  - [ ] support mld proxy multiwan
  - [\*] iproute2
  - [\*] iptables
  - [\*] ip6tables
  - [ ] l7filter\_pat
  - [\*] iwcontrol
  - [\*] l2tpd
  - [\*] lltdd
  - \*- mini\_upnp
  - [\*] miniigd
  - [\*] miniigd ipv6 support
  - [ ] mkdosfs
  - [ ] mp daemon
- (+)

<Select> < Exit > < Help >

## 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 nfjrom 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 datas 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

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.

### 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:

Case 1), if CONFIG\_RTL\_NF\_CONNTRACK\_GARBAGE\_NEW is not enabled at kernel configure, /proc/fast\_nat is only used for fastpath control.

Case 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 wantype, 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 : wantype 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 : wantype 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 has 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 forwardig 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

## **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.

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 are 0 ~ 5, 5 means all ports.

the valid value of mode are 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 are 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 pin 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)          |

## 7. FAQ

### 7.1 How to modify the mappings of LAN/WAN port?

Modify linux-2.6.30/include/net/rtl [/rtl865x\\_netif.h](#), and define RTL\_WANPORT\_MASK and RTL\_LANPORT\_MASK as needed.

In 8197D+8367RB platform, please also change the “RTL8367R\_WAN” definition in /linux-2.6.30/drivers/net/rtl819x/AsicDriver/rtl865xc\_asicregs.h and /linux-2.6.30/drivers/net/rtl819x/rtl8367r/rtk\_api.c.

### 7.2 How to change the baudrate 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.

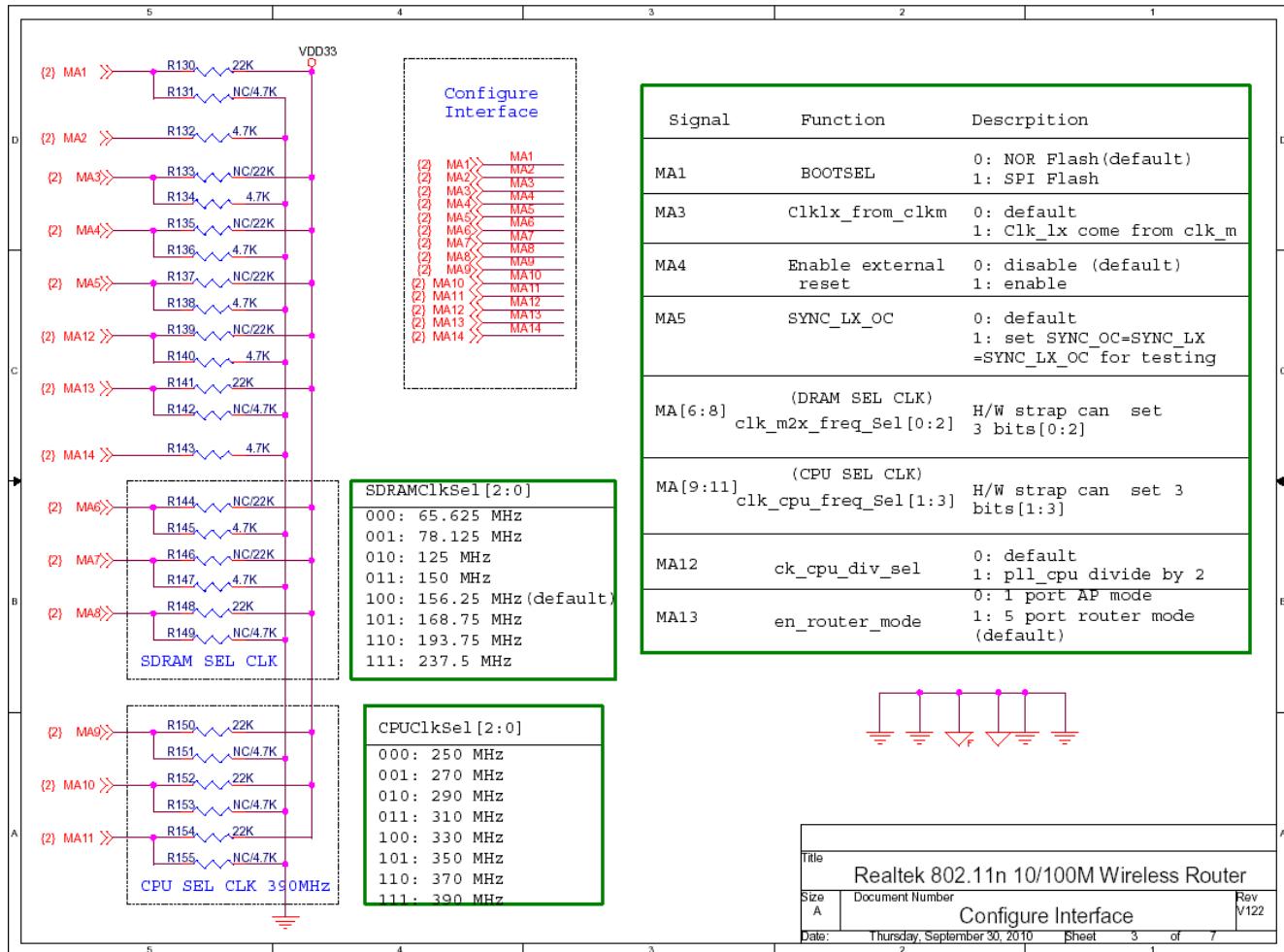


Fig 7.8.1 Realtek 802.11n 10/100M Wireless Router Interface Circuit

## 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, rebuild image and update it to AP.

After AP boots up, control ther 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

## 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\_setAsicEthernetPHYAdvCapalility(uint32 port, uint32 capalility);
- 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_setAsicEthernetPHYAdvCapalility(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_DHCPC 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_DHCPC`

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

Step 2, remove CONFIG\_RTL865X\_BICOLOR\_LED code in rtl\_nic.c

```
#if 0

#ifndef CONFIG_RTL865X_BICOLOR_LED
#define CONFIG_RTL865X_BICOLOR_LED

REG32(LEDSCR) |= (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(LEDSCR, READ_MEM32(LEDSCR) | 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(LEDSCR, 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 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 specific 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 0xFFFFFFF" > /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 Size Order Configuration

- \*\*\* Build rootfs options \*\*\*
- File system to mount root (squash fs) --->

<Select> < Exit > < Help >

## 7.25 For QoS using htb, something need 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:

	<i>HW_WLANX_11N_TRSWPAPE_C9</i>	<i>HW_WLANX_11N_TRSWPAPE_CC</i>
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:

- 1) linux-2.6.30/drivers/net/rtl819x/I2Driver/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
```

```

#ifndef 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 form 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 pwlan0 up
brctl addif br0 pwlan0
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 multilcast 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 libary

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

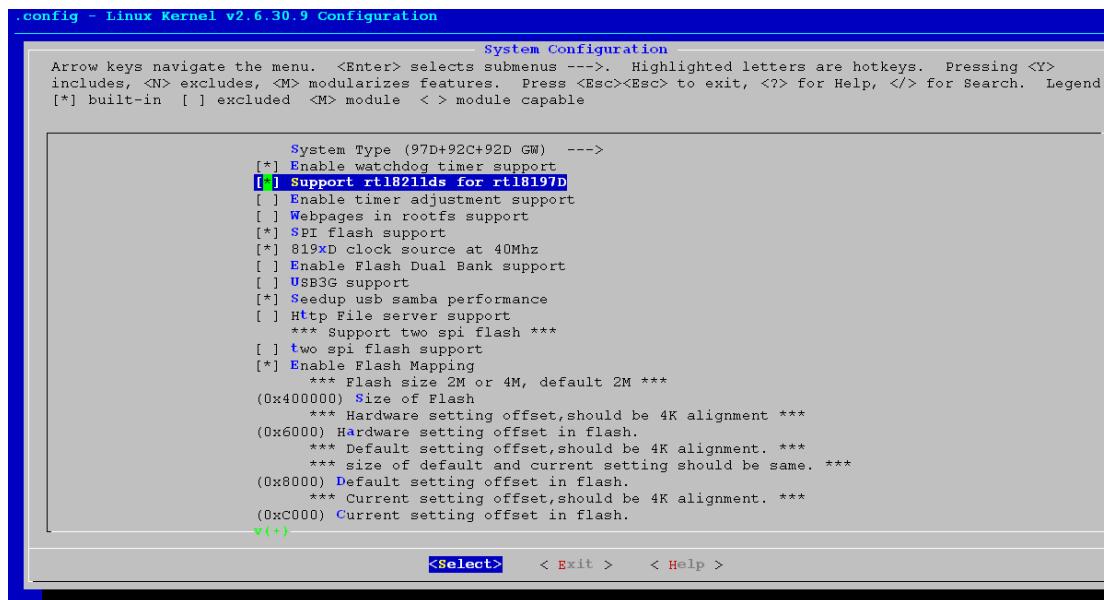
Answer:

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



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.

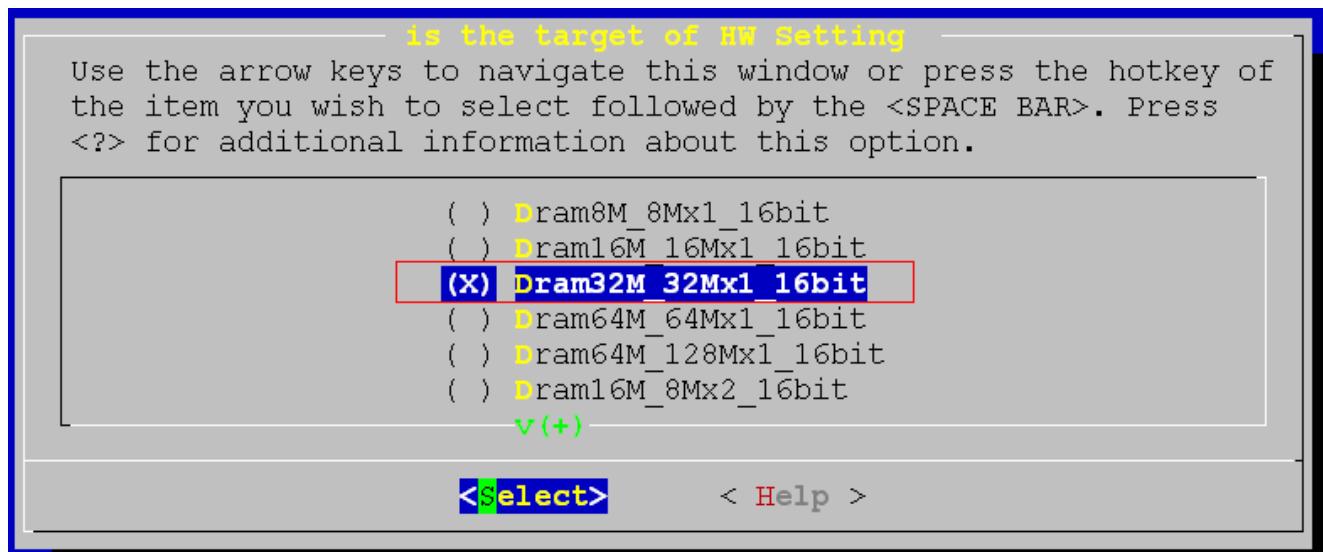
cd bootloader and

*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 udhcpd support:

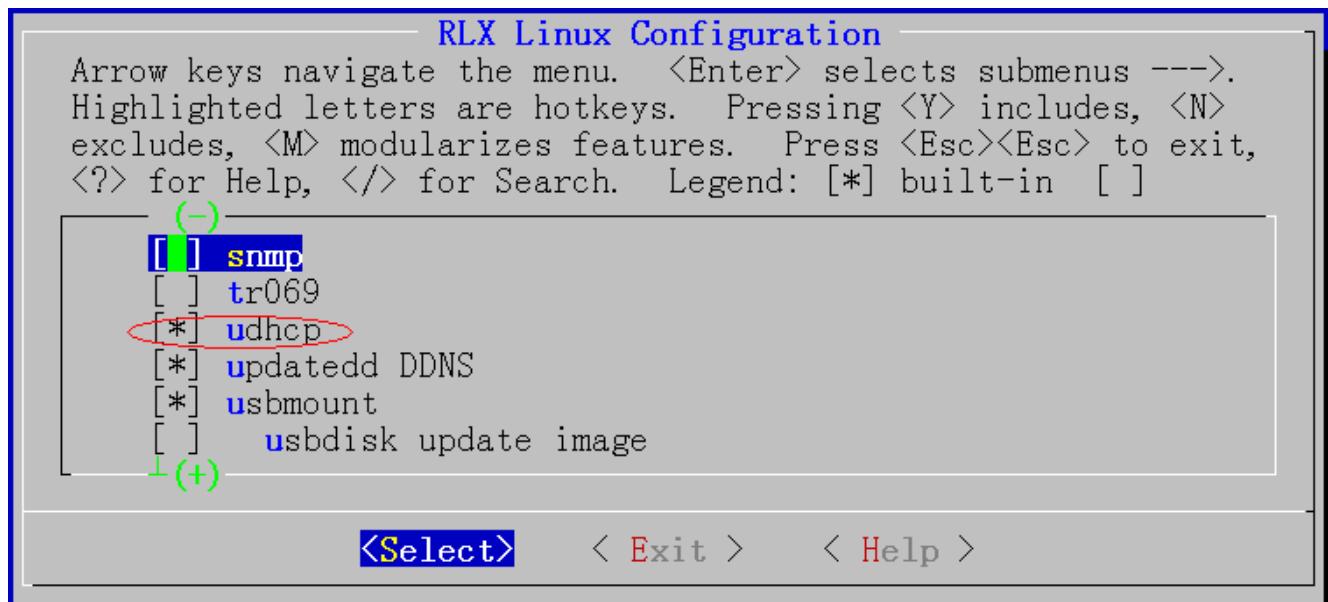
*config APP\_UDHCPD*

*bool "udhcp"*

*help*

*An embedded DHCP client and server*

Show:



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
}

```

```

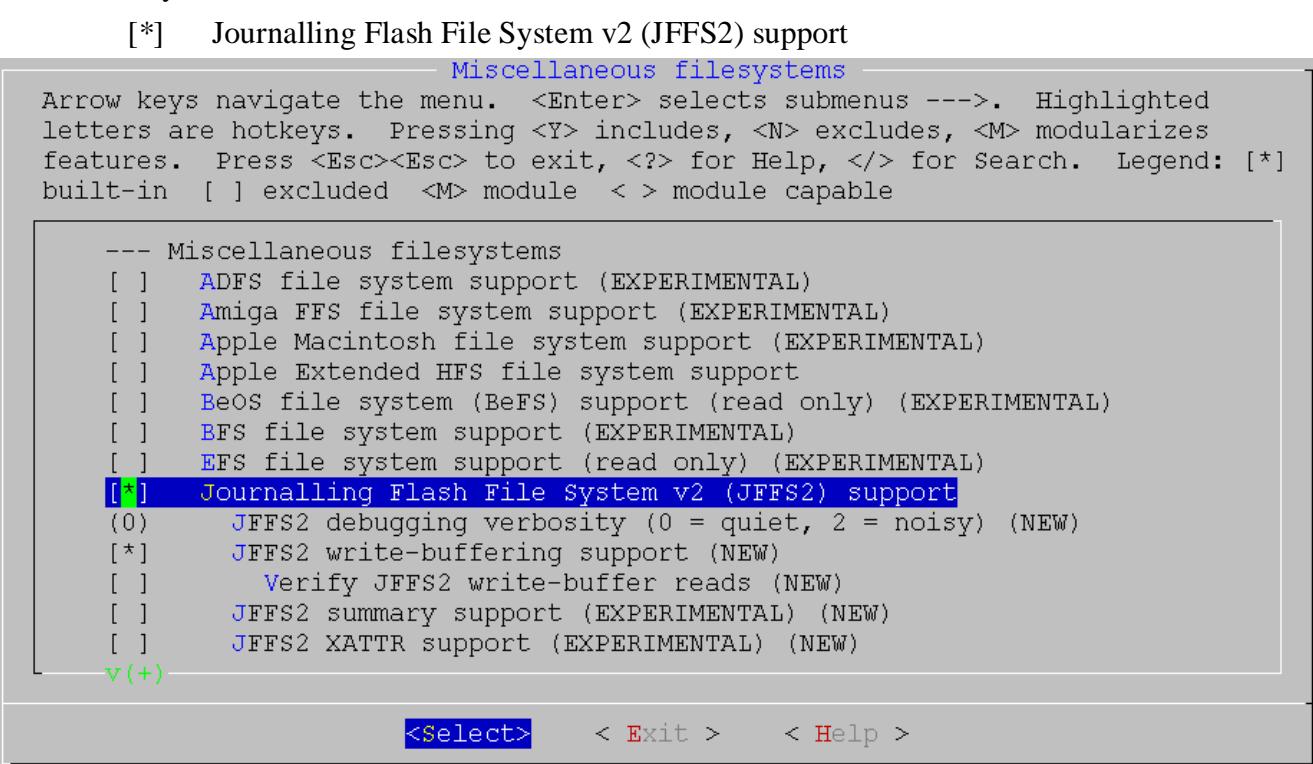
 size: (CONFIG_JFFS2_SIZE),
 offset: (WINDOW_SIZE - CONFIG_JFFS2_SIZE),
 } //mtd2
}

```

Step 2, please enable the following configurations.

*make linux\_menuconfig*

->File systems --->



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.*