Realtek Kenel 2.6 SDK User Guide

Date: 2011/06/03

Version: 1.29

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. 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

→ rtl819x-sdk-v2.x/rtl819x

Directory bootcode_rtl8196c_98 deposit bootloader source code of RTL8196C, RTL8198 and RTL8954C for Kenel 2.6 SDK while rtl819x-sdk-v2.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 ('/').

1.1. Compile Environment

Red Hat Enterprise [Recommend]

Fedora 9

Ubuntu 8.10/9.10

1.2. Compile bootloader

Steps to compile bootloader are as follows:

Step 1, enter bootloader directory.

cd /RLXLinux/bootcode_rtl8196c_98

Step 2, choose default configure file.

For 8196C:	*cp def-rtl8196c-config .config*
For 8198:	*cp def-rtl8198-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-sdk-v2.x/rtl819x/toolchain.

There are two toolchains.

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

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

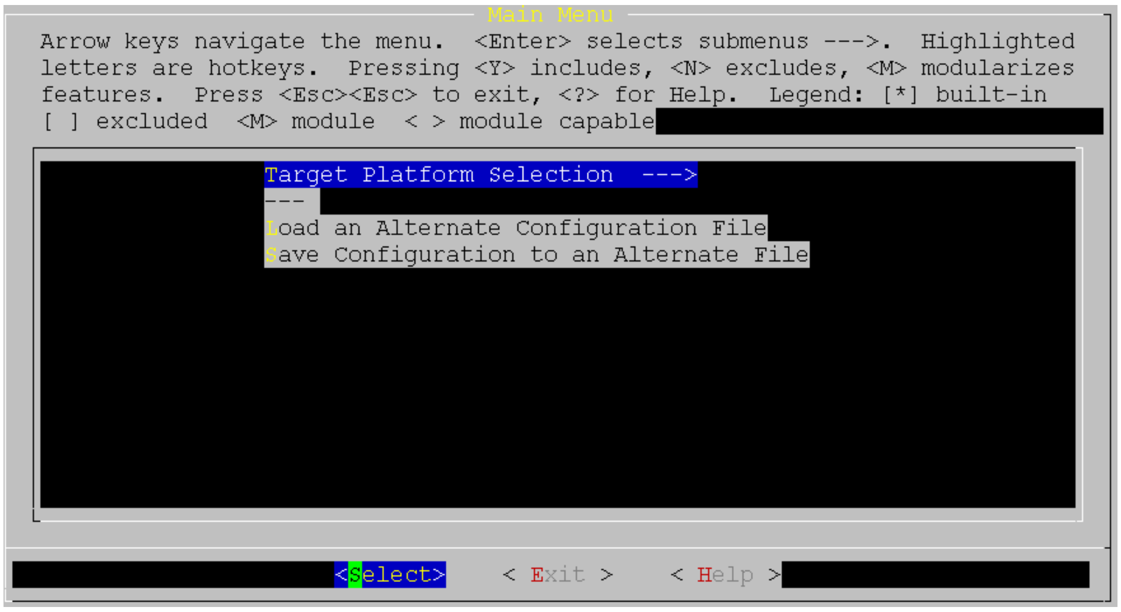
b. add the exact toolchain to the environment variant PATH.

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



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

                    Target Platform Selection  --->
                    ---
                    Load an Alternate Configuration File
                    Save Configuration to an Alternate File

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

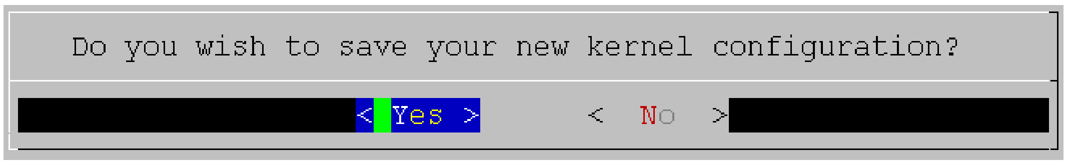
ii) Choose '<Exit>' and click 'Enter'

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.



```
Do you wish to save your new kernel configuration?

                  < Yes >      <  No  >
```

iii) Choose '<Yes>' and click 'Enter'

Step 4, complie bootloader.

make

1.3. Compile rlxlinux

Steps to compile applications and linux are as follows:

Step 1, enter rtl819x-sdk-v2.x/rtl819x directory.

Step 2, configure rlxlinux.

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

ii) Do some settings as follows:

'Selected Target' to choose rtl8196C, rtl8198 or rtl8954C supported at present;

'Selected Kernel' to choose linux kernel;

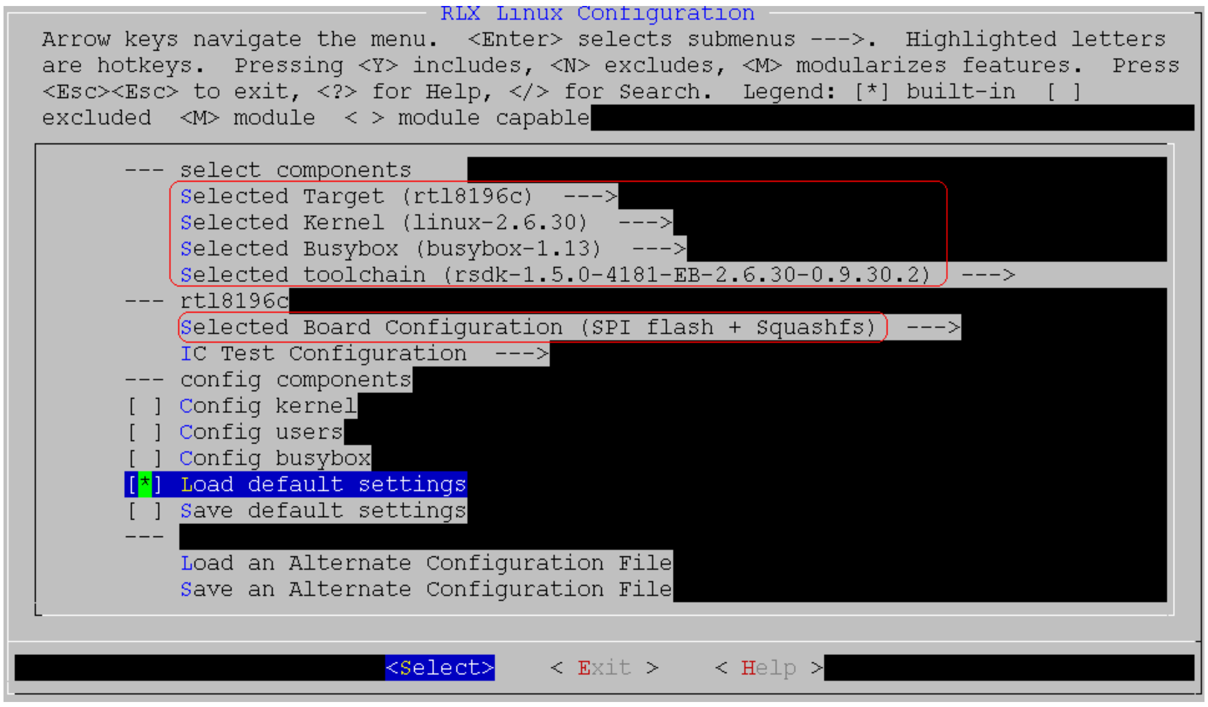
'Selected Busybox' to choose busybox;

'Selected toolchain' to choose toolchain to compile rlxlinux;

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

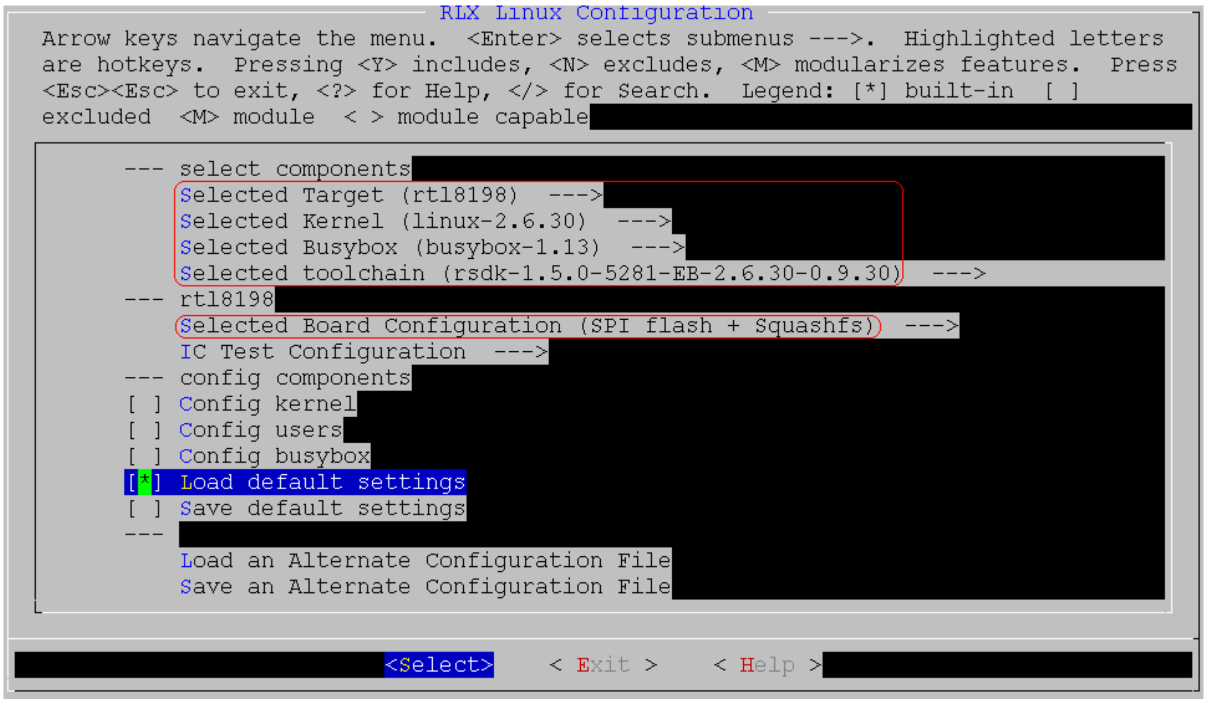
and squash file system.

Example I, settings for RTL8196C demo board as follows:



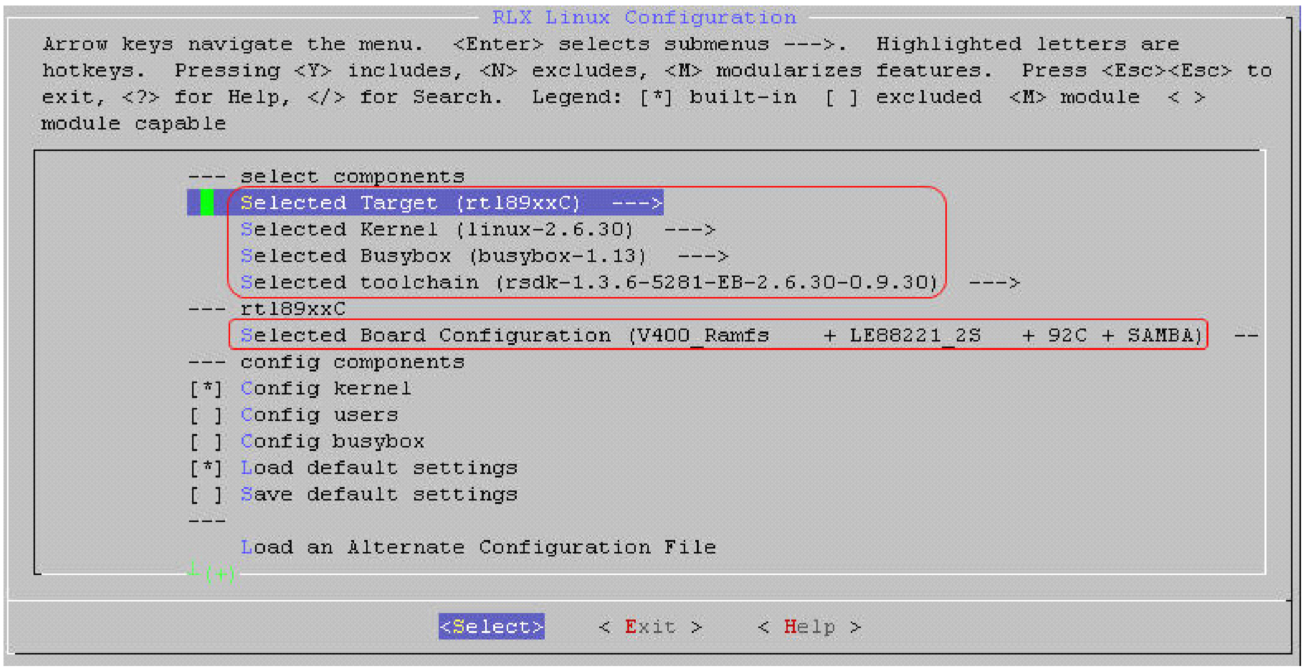
```
                        RLX Linux Configuration
  Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted letters
  are hotkeys.  Pressing <Y> includes, <N> excludes, <M> modularizes features.  Press
  <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend: [*] built-in  [ ]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

        --- select components
            Selected Target (rtl8196c)  --->
            Selected Kernel (linux-2.6.30)  --->
            Selected Busybox (busybox-1.13)  --->
            Selected toolchain (rsdk-1.5.0-4181-EB-2.6.30-0.9.30.2)  --->
        --- rtl8196c
            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

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

Example II, settings for RTL8198 demo board as follows:



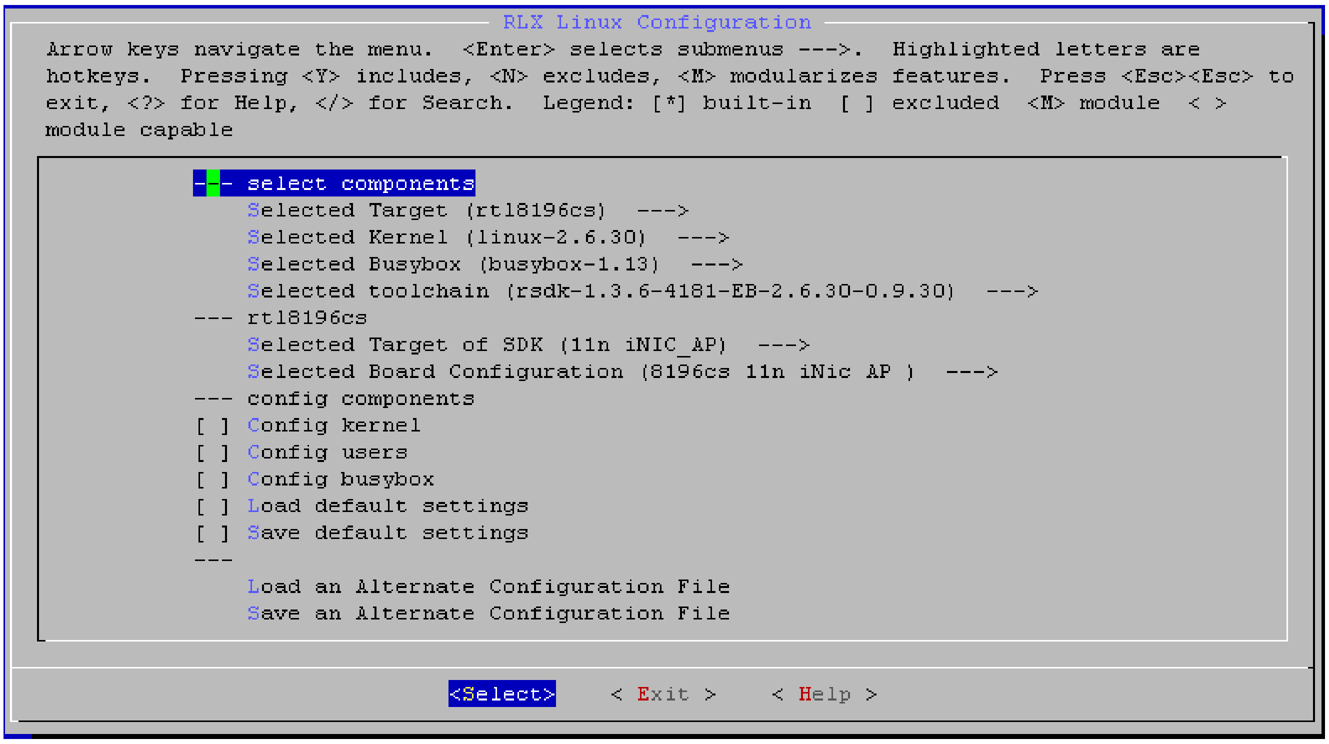
```
                        RLX Linux Configuration
  Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted letters
  are hotkeys.  Pressing <Y> includes, <N> excludes, <M> modularizes features.  Press
  <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend: [*] built-in  [ ]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

        --- select components
            Selected Target (rtl8198)  --->
            Selected Kernel (linux-2.6.30)  --->
            Selected Busybox (busybox-1.13)  --->
            Selected toolchain (rsdk-1.5.0-5281-EB-2.6.30-0.9.30)  --->
        --- rtl8198
            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

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

Example III, settings for RTL8954C demo board as follows:



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

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

                    <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 >
```

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

Note:

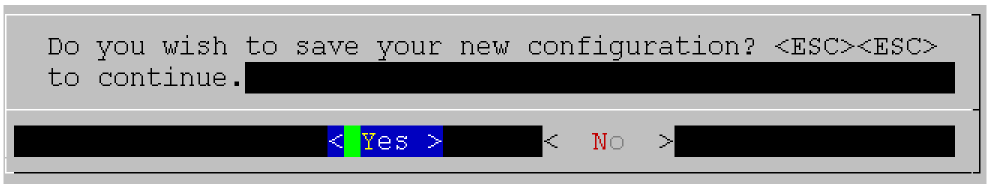
Above we just use configure applications and linux kernel in default way.

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'.



```
Do you wish to save your new configuration? <ESC><ESC>
to continue.

                   < Yes >      <  No  >
```

iv) Choose '<Yes>' and click 'Enter'

Step 3: compile rlxlinux.

make

1.4. Compile rlxlinux detailed

2.4.1 Compile rlxlinux process

Compile rlxlinux include several steps as follows.

(1) Compile user dir (rtl819x-sdk-v2.x/rtl819x/users);

(2) Compile board dir (rtl819x-sdk-v2.x/rtl819x/boards/rtl8198, rtl819x-sdk-v2.x/rtl819x/boards/rtl8196c or rtl819x-sdk-v2.x/rtl819x/boards/rtl89xxc) and compile user dir (rtl819x-sdk-v2.x/rtl819x/users) romfs;

(3) Compile linux dir (rtl819x-sdk-v2.x/rtl819x/linux-2.6.30) module;

(4) Compile linux dir (rtl819x-sdk-v2.x/rtl819x/linux-2.6.30);

(5) Create image dir (rtl819x-sdk-v2.x/rtl819x/image) and compile board dir (rtl819x-sdk-v2.x/rtl819x/boards/rtl8198 or rtl819x-sdk-v2.x/rtl819x/boards/rtl8196c) image.

2.4.2 Images generated

After compile rlxlinux, several images are generated at image dir (rtl819x-sdk-v2.x/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-98.dat or config-gw-96c.dat

2.4.3 Tools to make images

To make images, tools such as cvcfg-gw, compweb, cvimg, mgbin are stored at rtl819x-sdk-v2.x/rtl819x/users/goahead-2.1.1/LINUX.

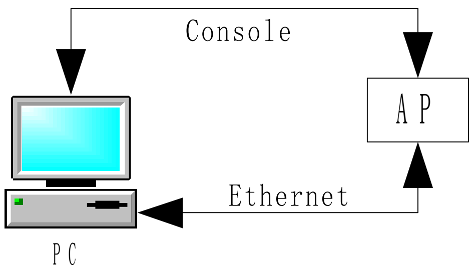
More info about these tools, please refer to Chapter 5 FLASH tools.

3. Kenel 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.

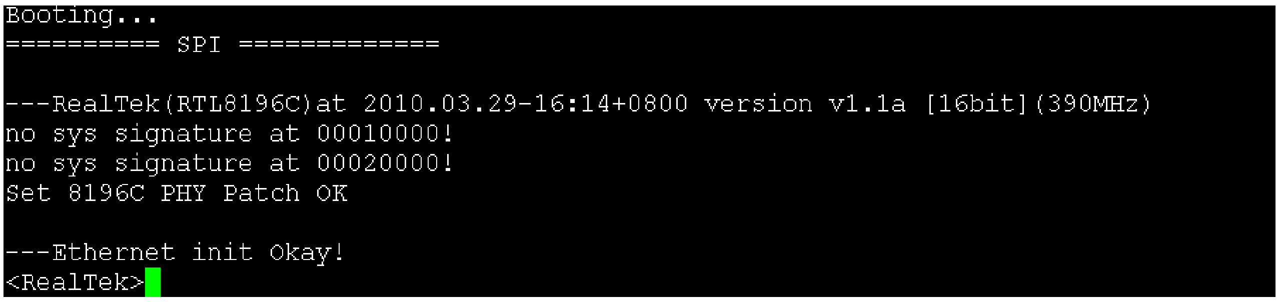


3.2 Upload bootloader image

Via 2.1 bootloader compilation, bootloader image is generated at /RLXLinux/bootcode_ rtl8196c_98/btcode/ which is boot.bin .

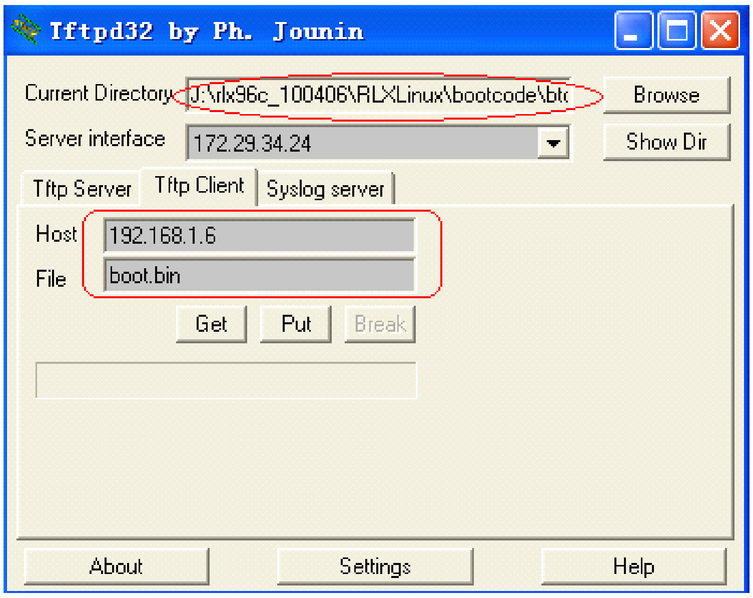
Steps to upload bootloader image are as follows:

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:


Current Directory: J:\rlx96c_100406\RLXLinux\bootcode\btc
Server interface: 172.29.34.24
Host: 192.168.1.6
File: boot.bin

Step 3, click 'Put' at tftp UI.

3.3 Upload rlxlinux image

Via 2.2 rlxlinux compilation, images are generated at rtl819x-sdk-v2.x/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.

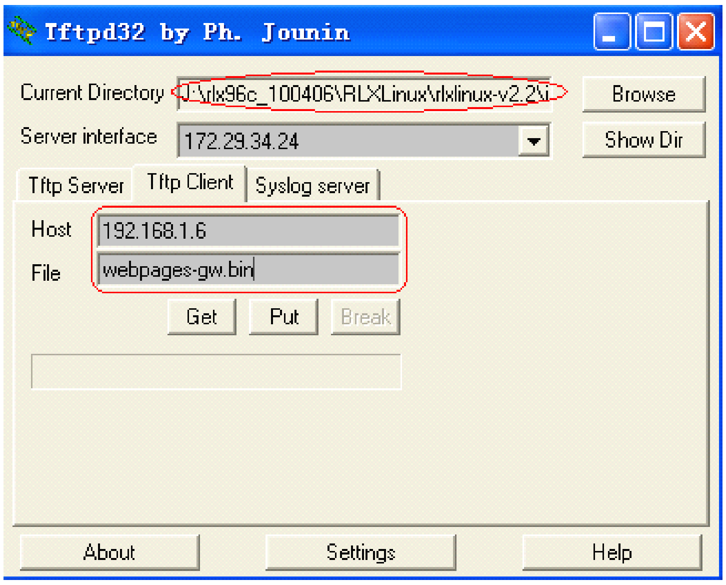
Steps to upload rlxlinux images are as follows:

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.



Current Directory: J:\rlx96c_100406\RLXLinux\rlxlinux-v2.2\i

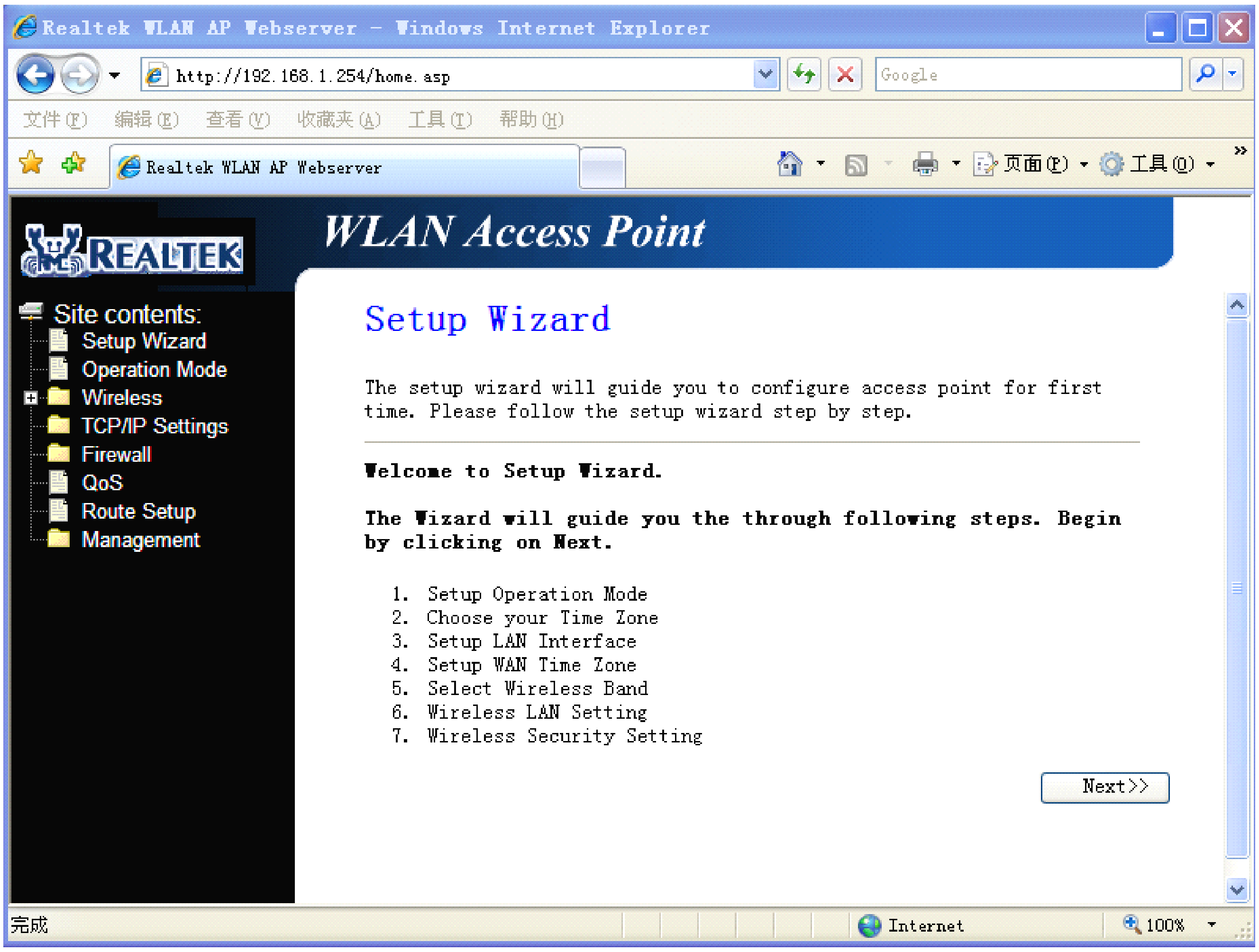
Server interface: 172.29.34.24

Host: 192.168.1.6

File: webpages-gw.bin

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.



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.

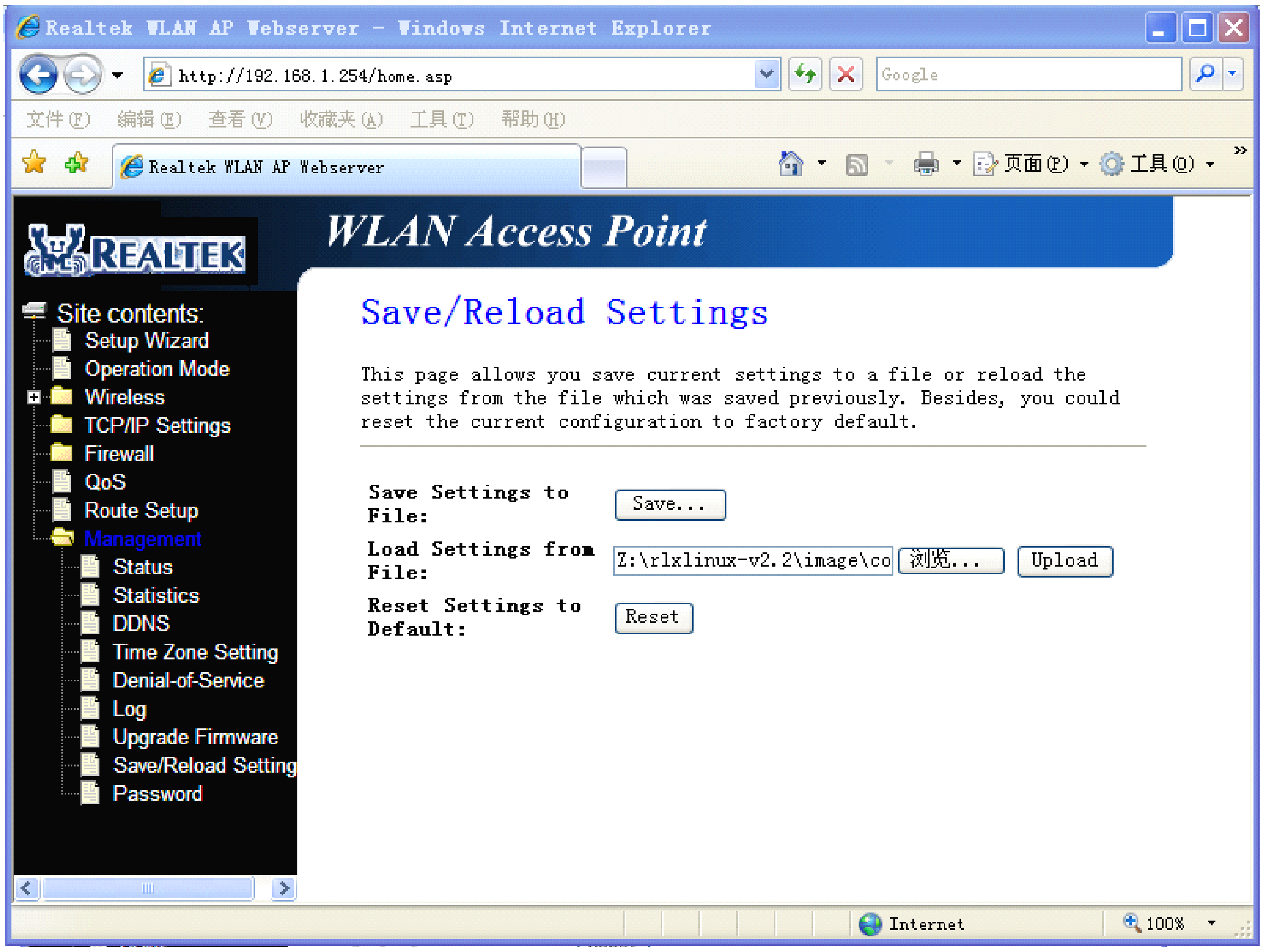
The Wizard will guide you the through following steps. Begin by clicking on Next.

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

3.4 Upload default setting data (Optional)

Via 2.2 rlxlinux compilation, default setting data are generated at rtl819x-sdk-v2.x/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.



Save/Reload Settings

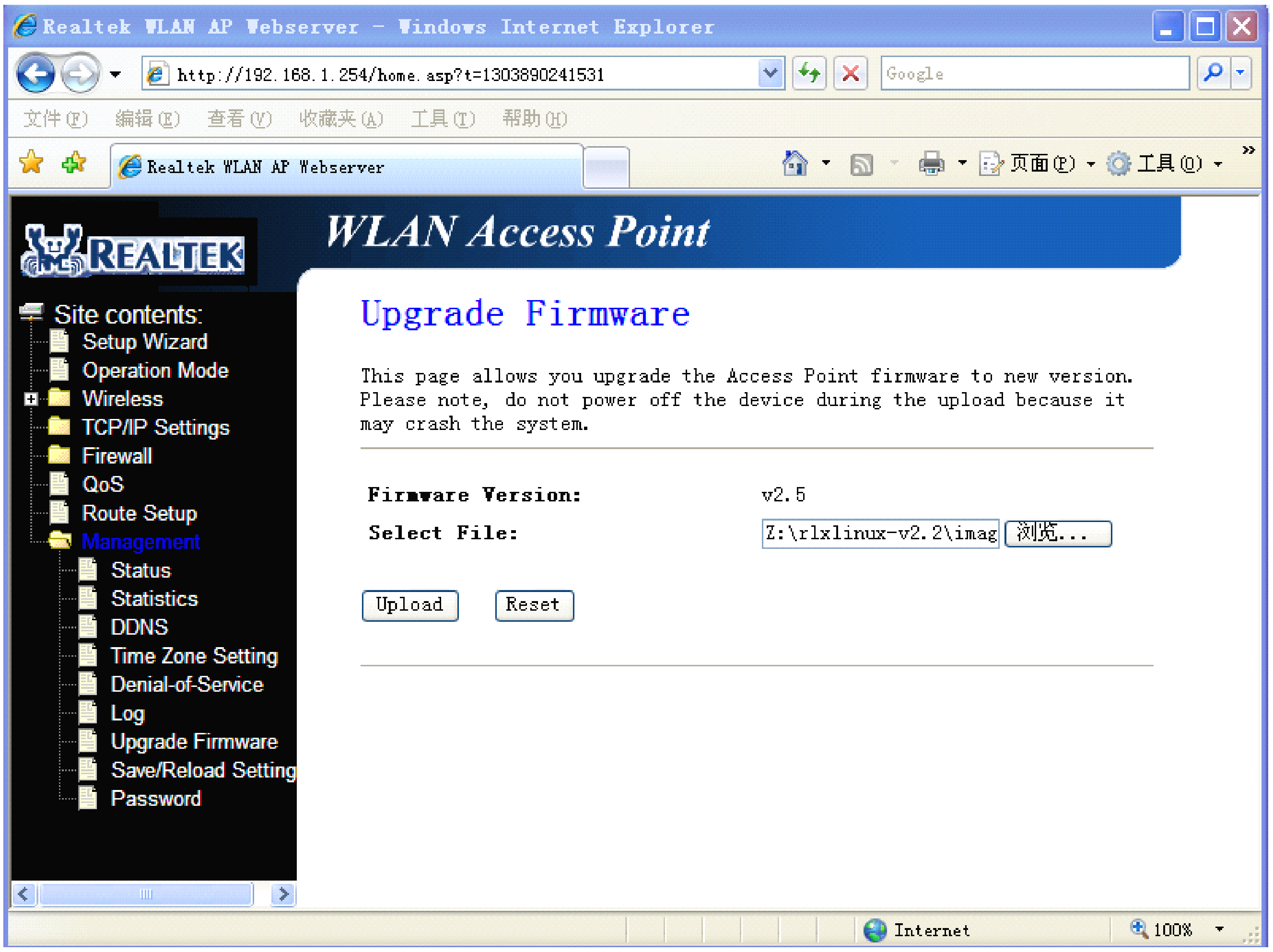
This page allows you save current settings to a file or reload the settings from the file which was saved previously. Besides, you could reset the current configuration to factory default.

Save Settings to File:	Save...
Load Settings from File:	Z:\rlxlinux-v2.2\image\co 浏览... Upload
Reset Settings to Default:	Reset

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-sdk-v2.x/rtl819x/image/fw.bin), then upload.



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\imag

After upload complete, AP will reboot.

4. Features configure

4.1 PCI support

4.1.1 Linux kernel PCI not support

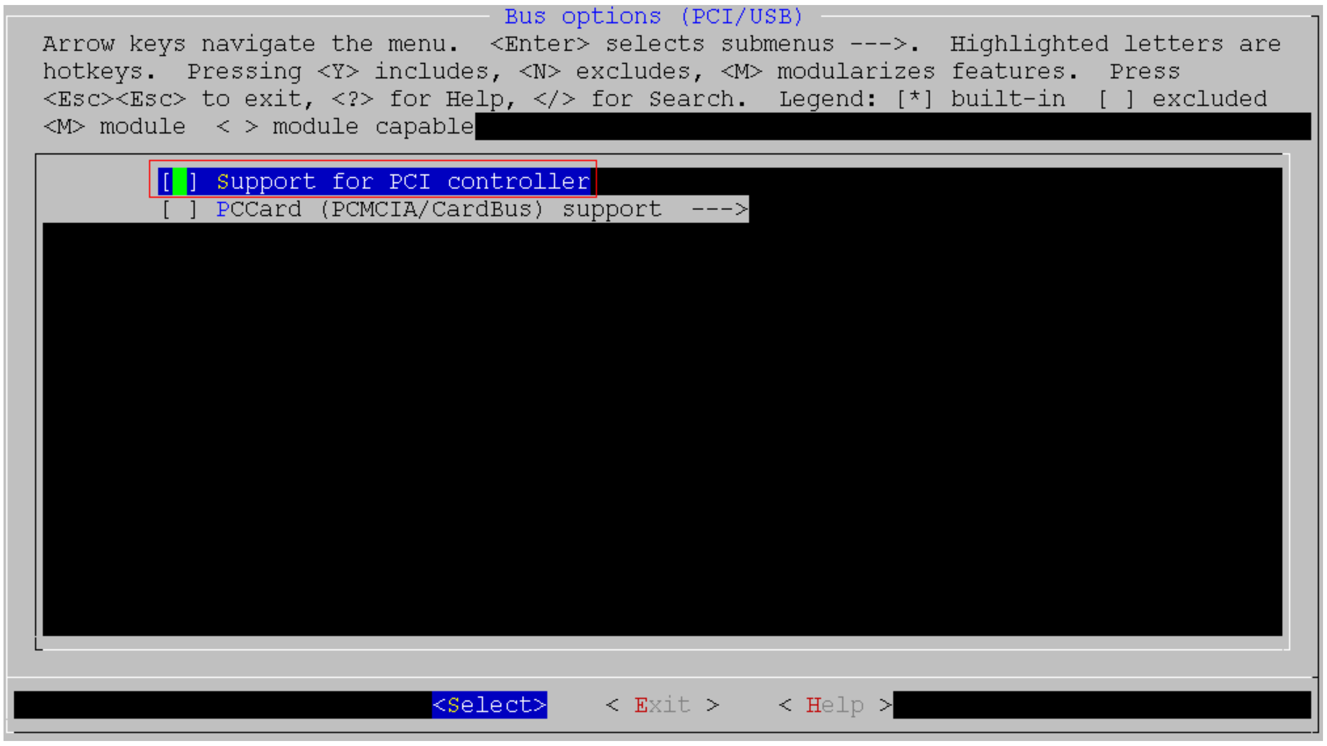
By kernel default settings, linux kernel PCI is not supported, because wireless driver use its own PCI driver. Linux kernel configure as follows.

make linux_menuconfig // To configure linux kernel settings

Menuconfig:

Bus options (PCI/USB) --->

Support for PCI controller // Not selected



```
                     Bus options (PCI/USB)
 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 PCI controller
        [ ] PCCard (PCMCIA/CardBus) support  --->

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

4.1.2 Linux kernel PCI support

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

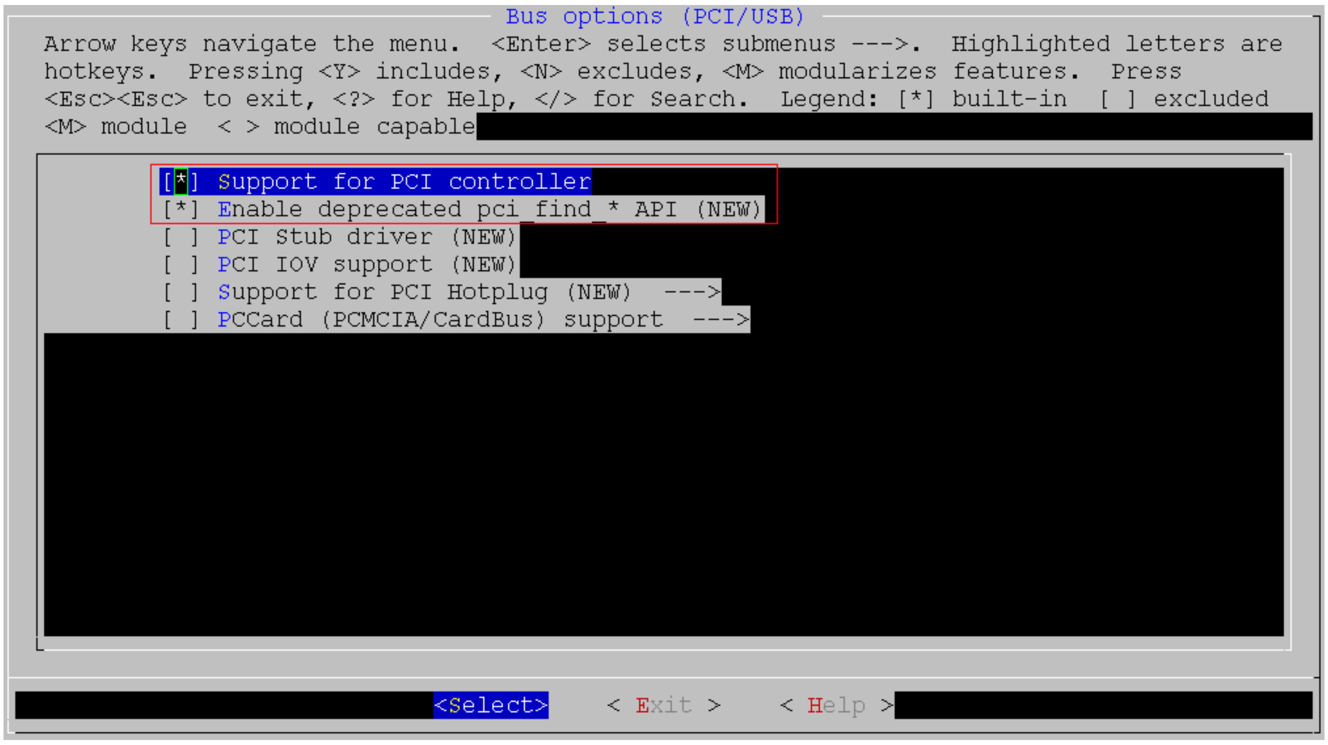
make linux_menuconfig // To configure linux kernel settings

Menuconfig:

Bus options (PCI/USB) --->

Support for PCI controller // Selected

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


```
                          Bus options (PCI/USB)
  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 PCI controller
          [*] Enable deprecated pci_find_* API (NEW)
          [ ] PCI Stub driver (NEW)
          [ ] PCI IOV support (NEW)
          [ ] Support for PCI Hotplug (NEW)  --->
          [ ] PCCard (PCMCIA/CardBus) support  --->

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

4.2 USB support

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

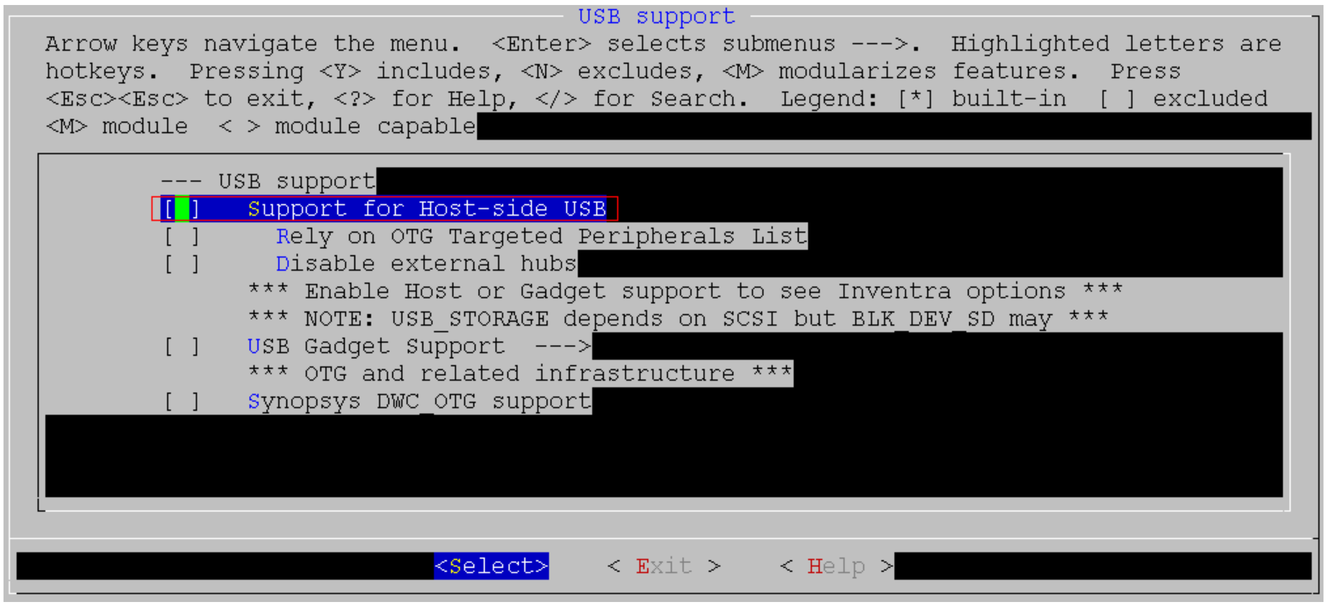
make linux_menuconfig // To configure linux kernel settings

Menuconfig:

Device Drivers --->

USB support --->

Support for Host-side USB // Not selected


```
                             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
          [ ]     Rely on OTG Targeted Peripherals List
          [ ]     Disable external hubs
                *** Enable Host or Gadget support to see Inventra options ***
                *** NOTE: USB_STORAGE depends on SCSI but BLK_DEV_SD may ***
          [ ]   USB Gadget Support  --->
                *** OTG and related infrastructure ***
          [ ]   Synopsys DWC_OTG support

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

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

make linux_menuconfig // To configure linux kernel settings

Menuconfig:

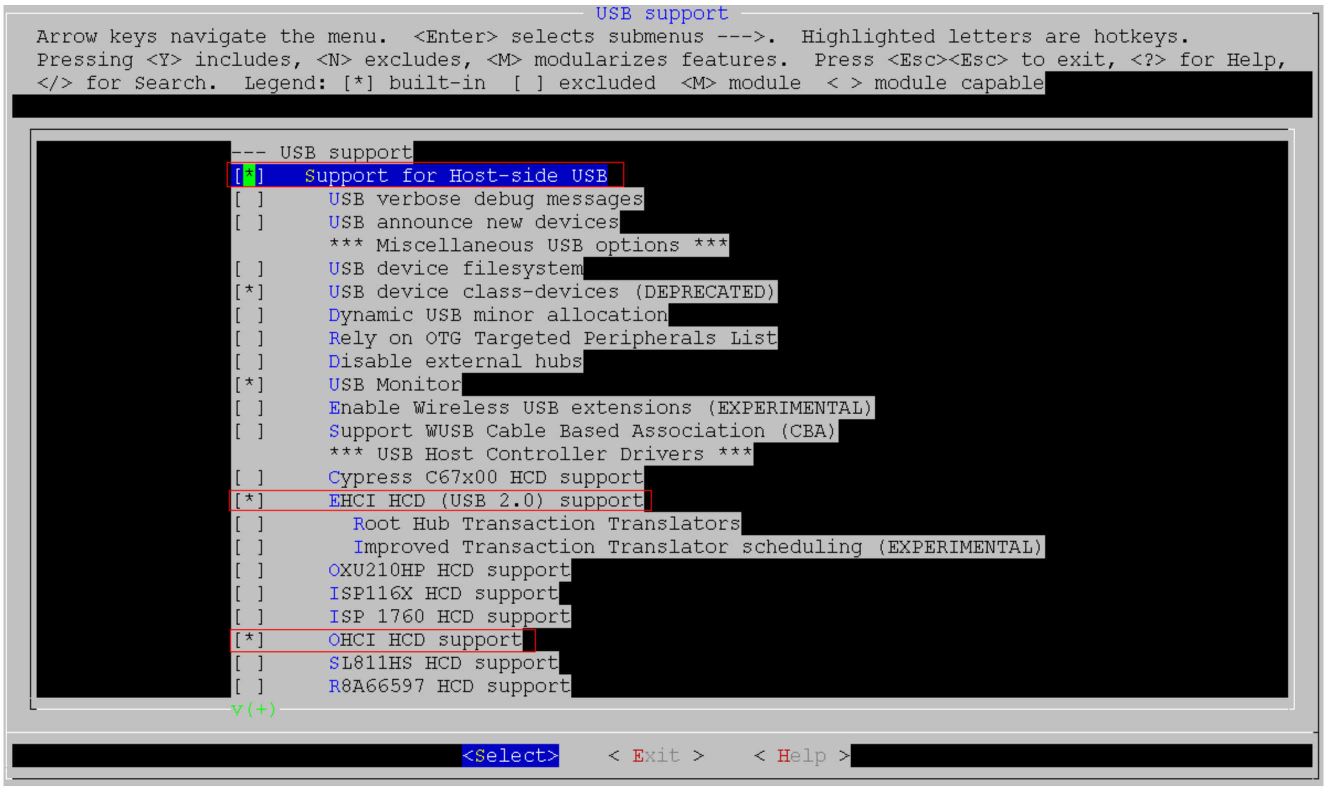
Device Drivers --->

USB support --->

Support for Host-side USB // selected

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

OHCI HCD support // To support USB 1.1, if 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
                  [ ]     USB announce new devices
                          *** Miscellaneous USB options ***
                  [ ]     USB device filesystem
                  [*]     USB device class-devices (DEPRECATED)
                  [ ]     Dynamic USB minor allocation
                  [ ]     Rely on OTG Targeted Peripherals List
                  [ ]     Disable external hubs
                  [*]     USB Monitor
                  [ ]     Enable Wireless USB extensions (EXPERIMENTAL)
                  [ ]     Support WUSB Cable Based Association (CBA)
                          *** USB Host Controller Drivers ***
                  [ ]     Cypress C67x00 HCD support
                  [*]     EHCI HCD (USB 2.0) support
                  [ ]       Root Hub Transaction Translators
                  [ ]       Improved Transaction Translator scheduling (EXPERIMENTAL)
                  [ ]     OXU210HP HCD support
                  [ ]     ISP116X HCD support
                  [ ]     ISP 1760 HCD support
                  [*]     OHCI HCD support
                  [ ]     SL811HS HCD support
                  [ ]     R8A66597 HCD support
                  v(+)

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

4.3 Samba support

4.3.1 Samba kernel configure

4.3.1.1 SCSI support

Linux kernel configure for SCSI as follows.

make linux_menuconfig // To configure linux kernel settings

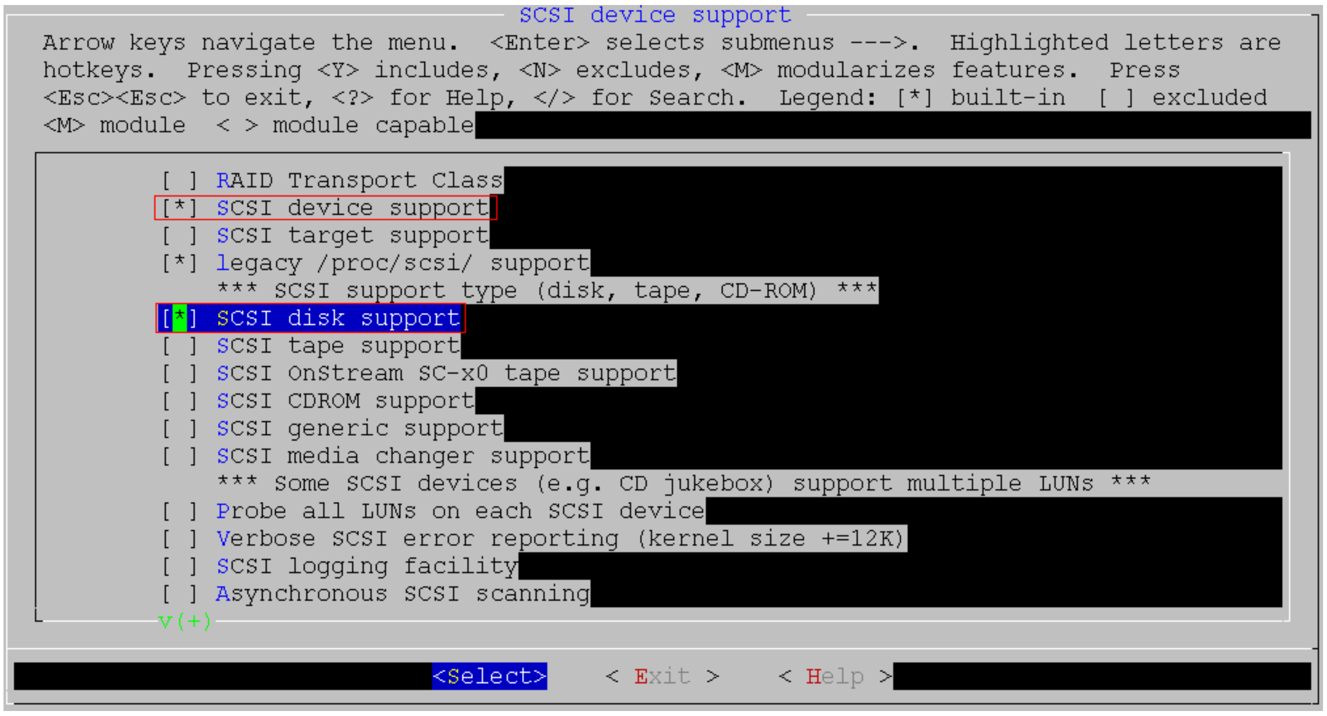
Menuconfig:

Device Drivers --->

SCSI device support --->

SCSI device support // selected

SCSI disk support // selected



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

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

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

4.3.1.2 File systems

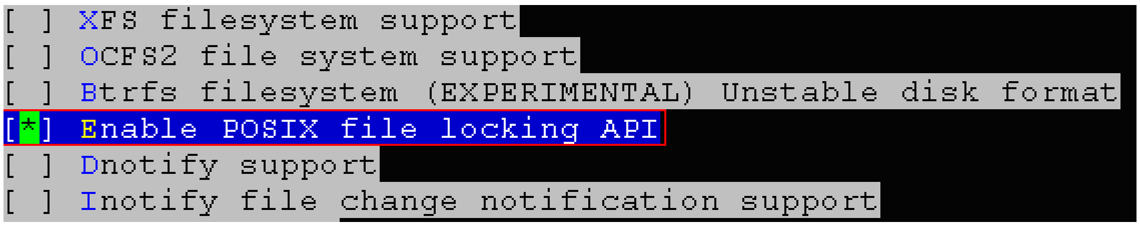
Linux kernel configure for file systems as follows.

make linux_menuconfig // To configure linux kernel settings

Menuconfig (1):

File systems --->

Enable POSIX file locking API // selected



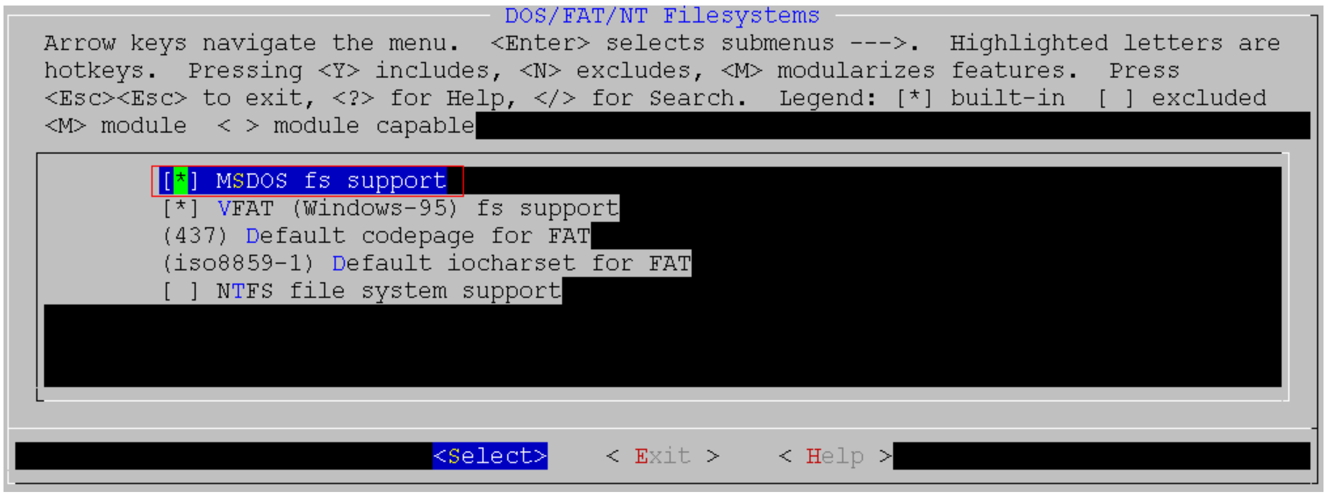
```
[ ] XFS filesystem support
[ ] OCFS2 file system support
[ ] Btrfs filesystem (EXPERIMENTAL) Unstable disk format
[*] Enable POSIX file locking API
[ ] Dnotify support
[ ] Inotify file change notification support
```

Menuconfig (2):

File systems --->

DOS/FAT/NT Filesystems --->

MSDOS fs support // selected



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

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

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

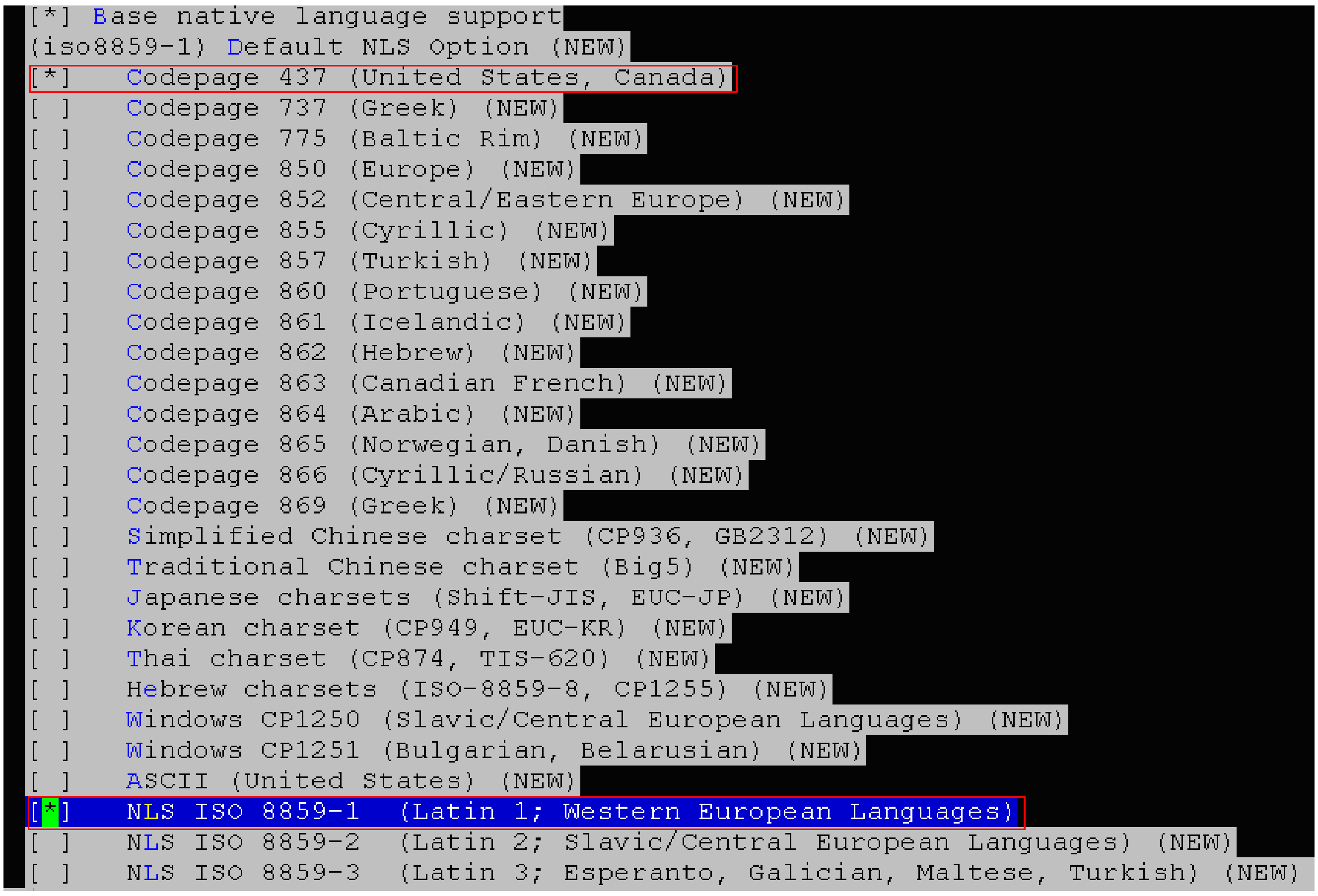
Menuconfig (3):

File systems   --->

Native Language Support   --->

Codepage 437 (United States, Canada) //Selected

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



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

4.3.1.3 USB support

Linux kernel configure for USB as follows.

make linux_menuconfig   // To configure linux kernel settings

Menuconfig:

Device Drivers --->

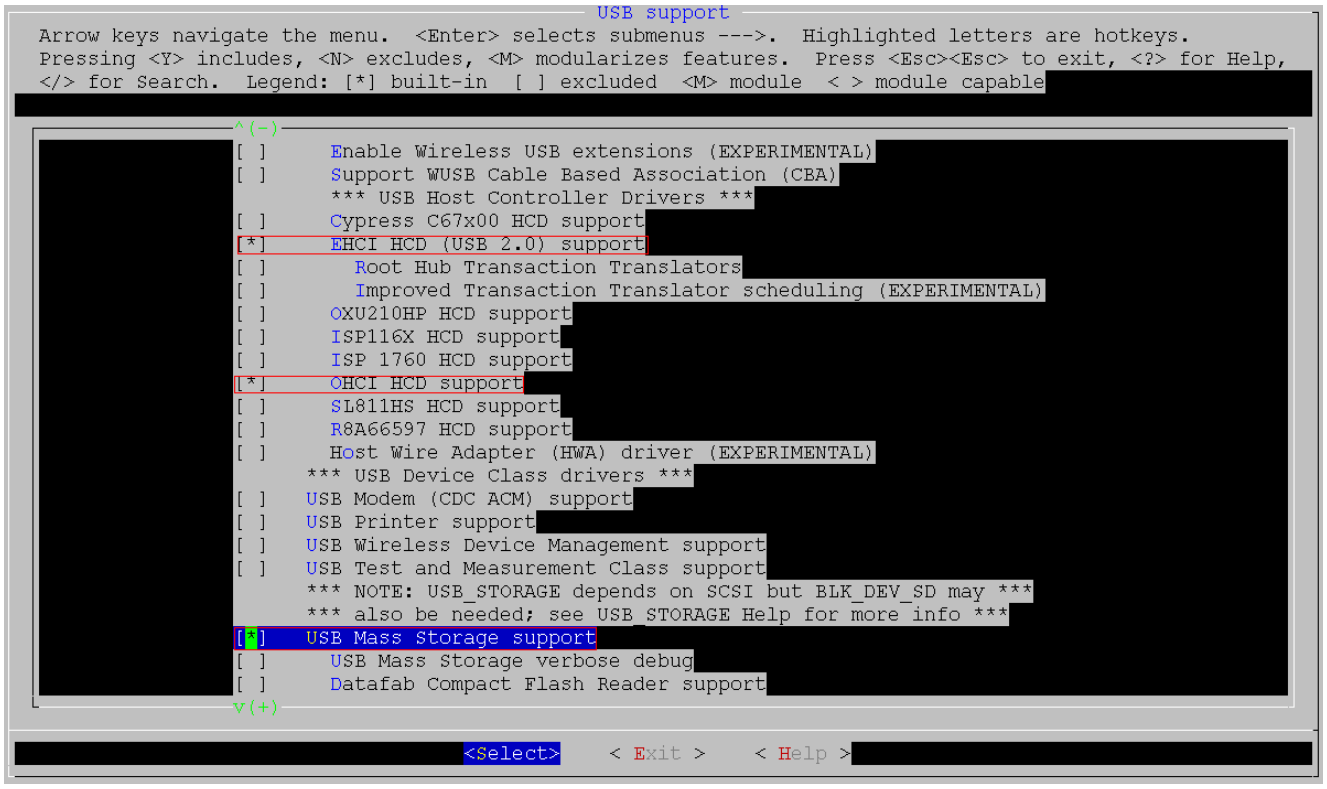
USB support --->

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

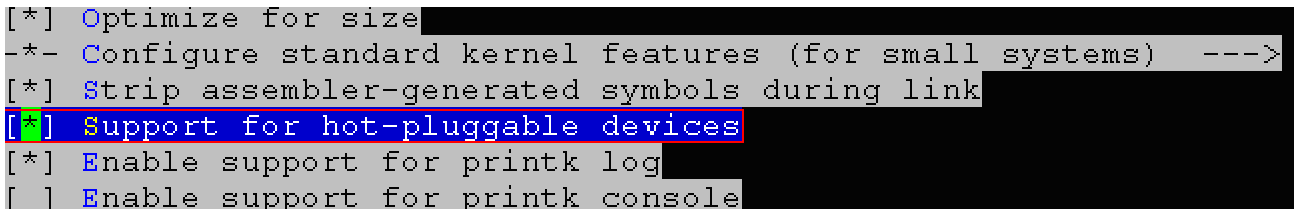
make linux_menuconfig   // To configure linux kernel settings

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
```

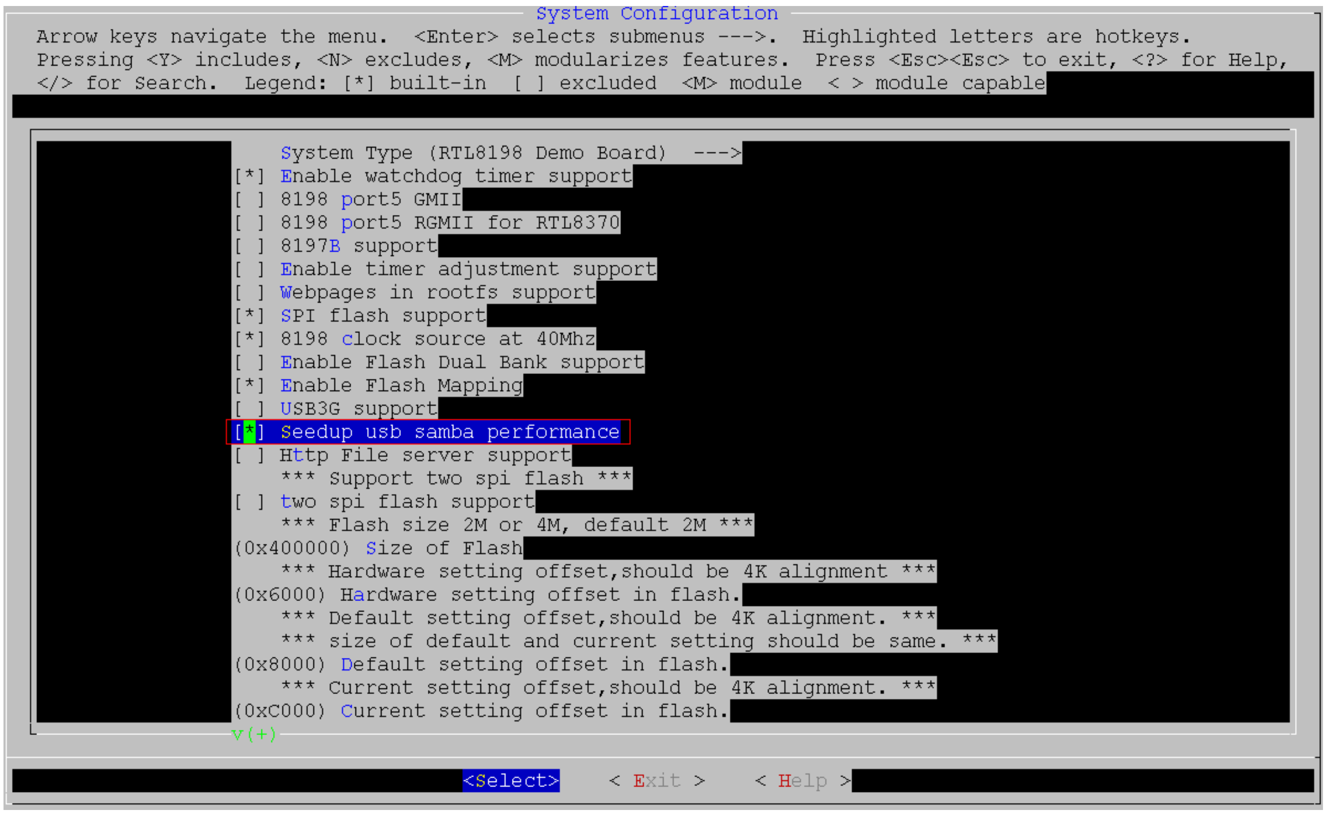
4.3.1.5 System configure

make linux_menuconfig   // To configure linux kernel settings

Menuconfig:

System Configuration --->

[*] Seedup usb samba performance



```
                              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 (RTL8198 Demo Board)  --->
               [*] Enable watchdog timer support
               [ ] 8198 port5 GMII
               [ ] 8198 port5 RGMII for RTL8370
               [ ] 8197B support
               [ ] Enable timer adjustment support
               [ ] Webpages in rootfs support
               [*] SPI flash support
               [*] 8198 clock source at 40Mhz
               [ ] Enable Flash Dual Bank support
               [*] Enable Flash Mapping
               [ ] USB3G support
               [*] Seedup usb samba performance
               [ ] Http File server support
                   *** Support two spi flash ***
               [ ] two spi flash support
                   *** Flash size 2M or 4M, default 2M ***
               (0x400000) Size of Flash
                   *** Hardware setting offset,should be 4K alignment ***
               (0x6000) Hardware setting offset in flash.
                   *** Default setting offset,should be 4K alignment. ***
                   *** size of default and current setting should be same. ***
               (0x8000) Default setting offset in flash.
                   *** Current setting offset,should be 4K alignment. ***
               (0xC000) Current setting offset in flash.
               v(+)

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

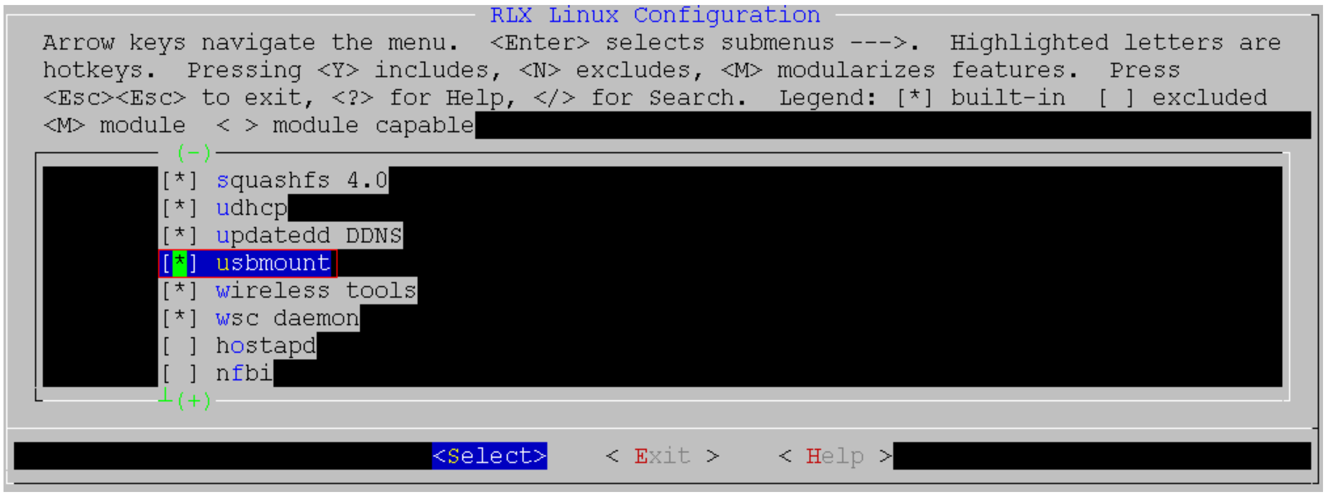
4.3.2 Samba application configure

4.3.2.1 enable usbmount

make users_menuconfig // To configure application settings

Menuconfig:

usbmount // 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
           (-)
          [*] squashfs 4.0
          [*] udhcp
          [*] updatedd DDNS
          [*] usbmount
          [*] wireless tools
          [*] wsc daemon
          [ ] hostapd
          [ ] nfbi
          ┴(+)

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

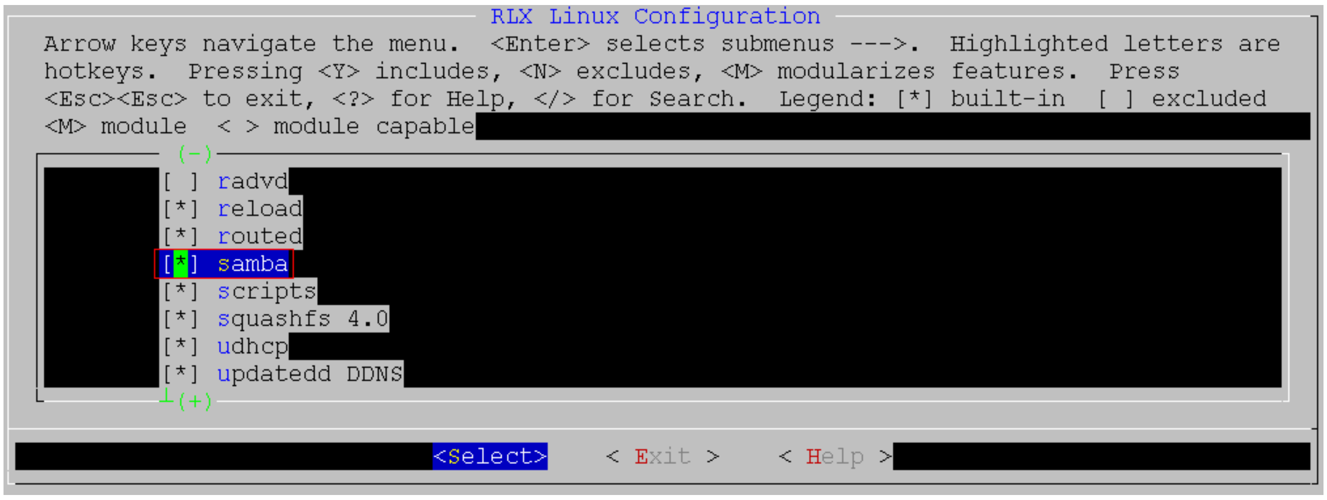
4.3.2.2 enable samba

make users_menuconfig // To configure application settings

Menuconfig:

samba // selected

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



```
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
            (-)
            [ ] radvd
            [*] reload
            [*] routed
            [*] samba
            [*] scripts
            [*] squashfs 4.0
            [*] udhcp
            [*] updatedd DDNS
            (+)

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

4.3.3 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.

4.3.4 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.

4.4 USB 3G support

4.4.1 Basic software flow

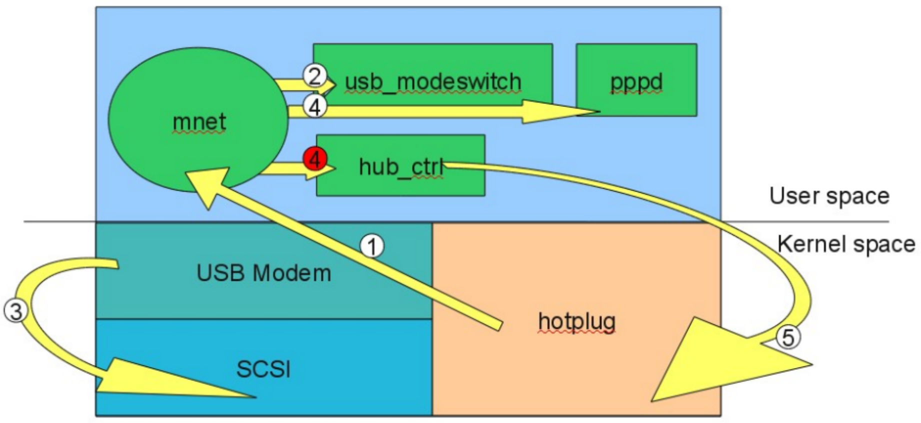
① Get the hotplug event.

② Find the right config for usb_modeswitch.

③ Monitor and control the activity of eject storage usb.

④ Prepare /var/usb3.option and /var/usb3g.chat and call pppd-chat.

⑤ If eject fail or call pppd fail, execute hub_ctrl to disconnect usb device, then run from beginning.



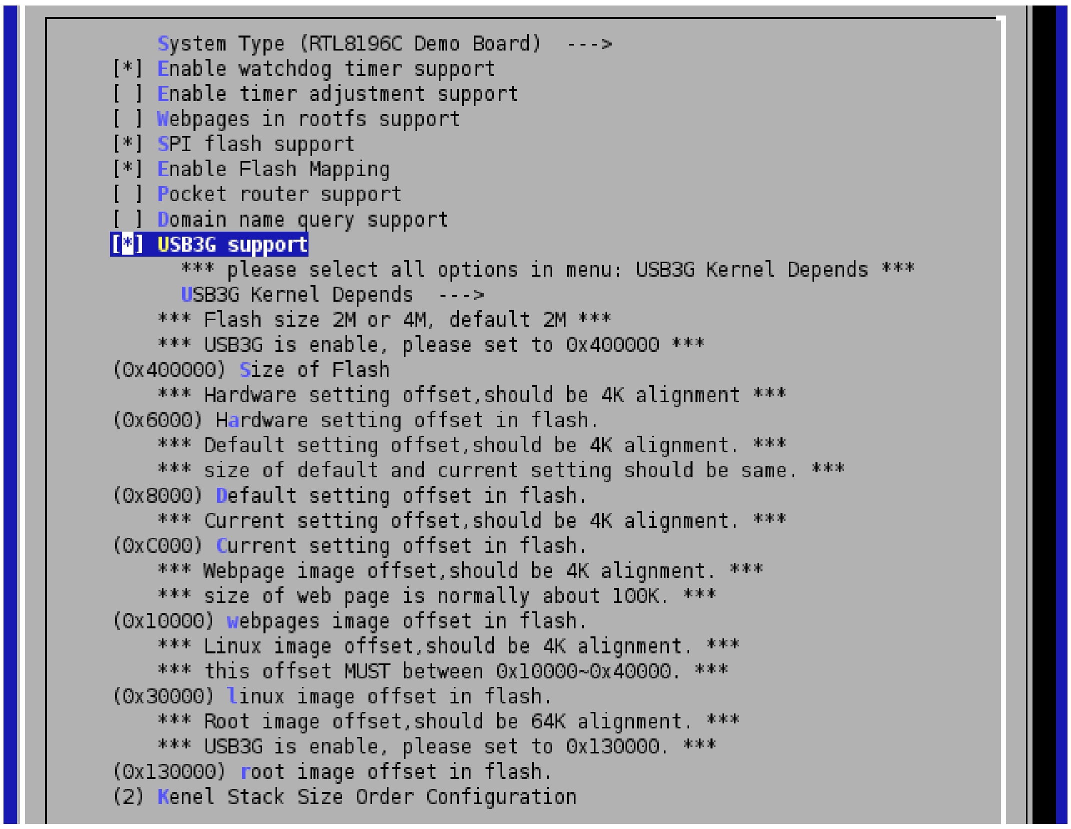
Ps.

mnet: a utility to control the above flow.

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

4.4.2 Linux kernel configure

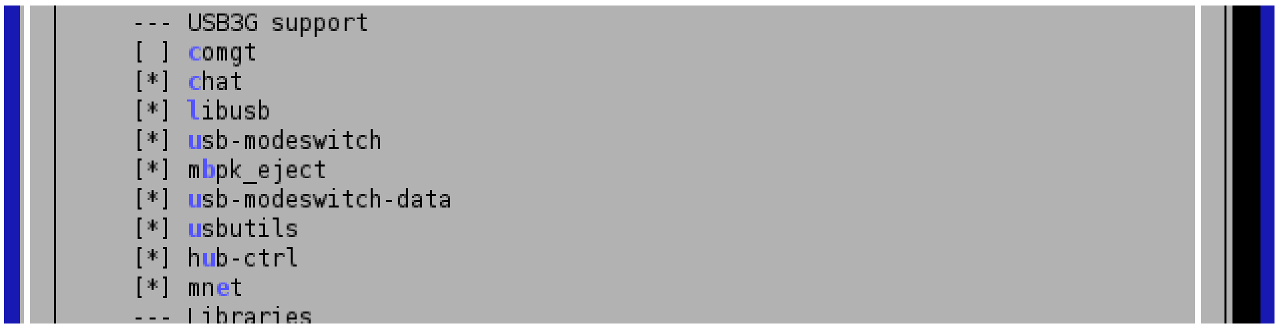
In System Configuration, please select USB3G support and change the size of Flash to 0x400000, and linux image offset to 0x130000 (make sure your board is 32M/4M sdram/flash, and bootloader is also compatible).



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

4.4.3 User configure

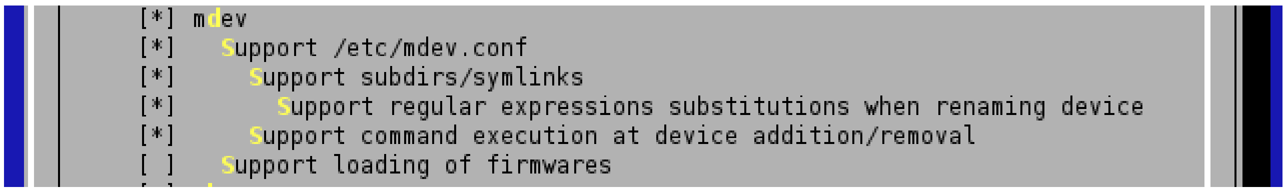
1) User space programs



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

2) Busybox command

Linux System Utilities --->

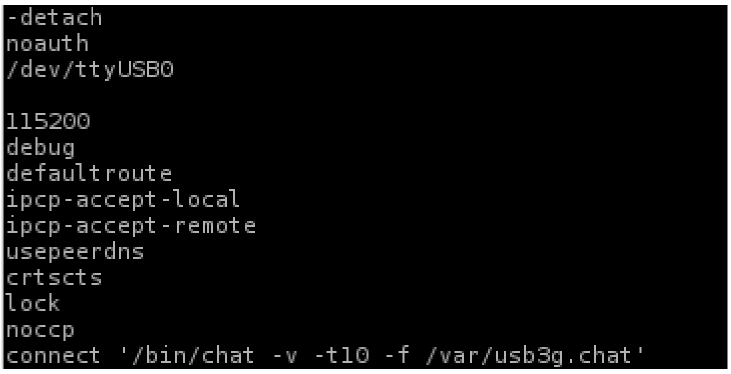


```
[*] mdev
[*]   Support /etc/mdev.conf
[*]     Support subdirs/symlinks
[*]       Support regular expressions substitutions when renaming device
[*]     Support command execution at device addition/removal
[ ]   Support loading of firmwares
```

4.4.4 Add new USB 3G dongle support

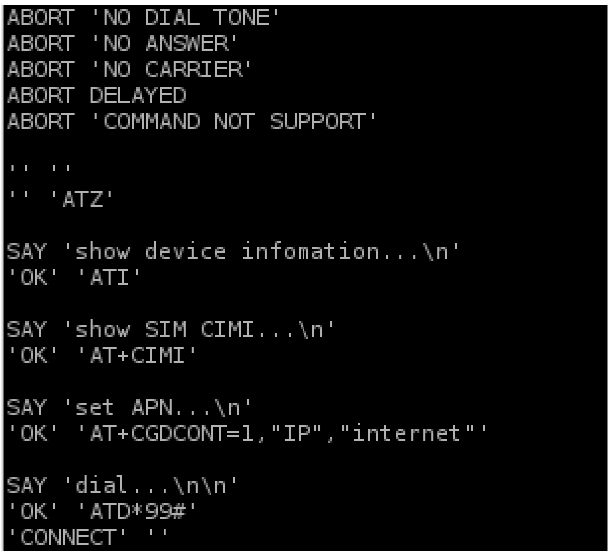
4.4.4.1 Modify mnet

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' ''
```

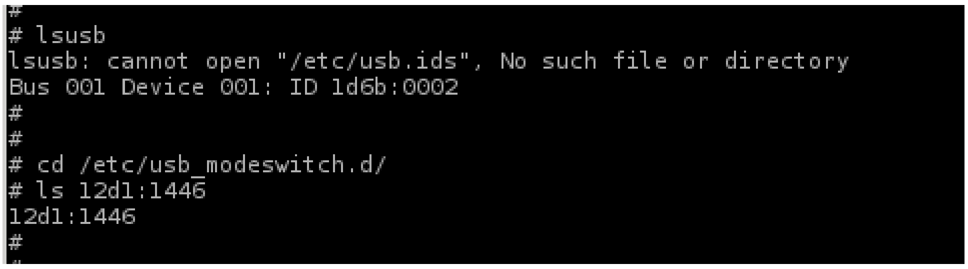
Fig. Example of ppp chat

In some cases, for example: the dongle has special AT command, you may need to change the function void gen_ppp_option(char *ttyitf) 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
#
```

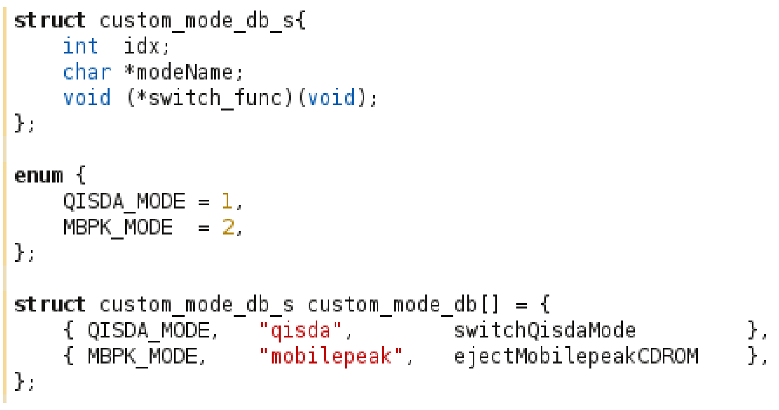
Fig. find matched usb_modeswitch config

4.4.4.3 Modify option.c

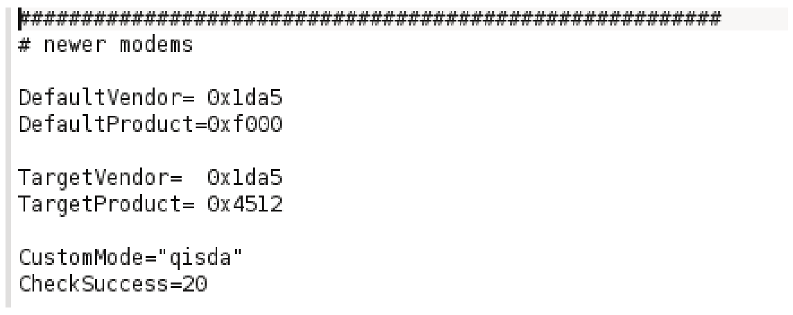
If there is no supported config file, you need manually to add the vendor id and product id into linux-2.6.30/drivers/usb/serial/option.c, and also implement a suitable eject utility (ex:usb_modeswitch we used, it's kind of eject utility) for this dongle.

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    },
};
```

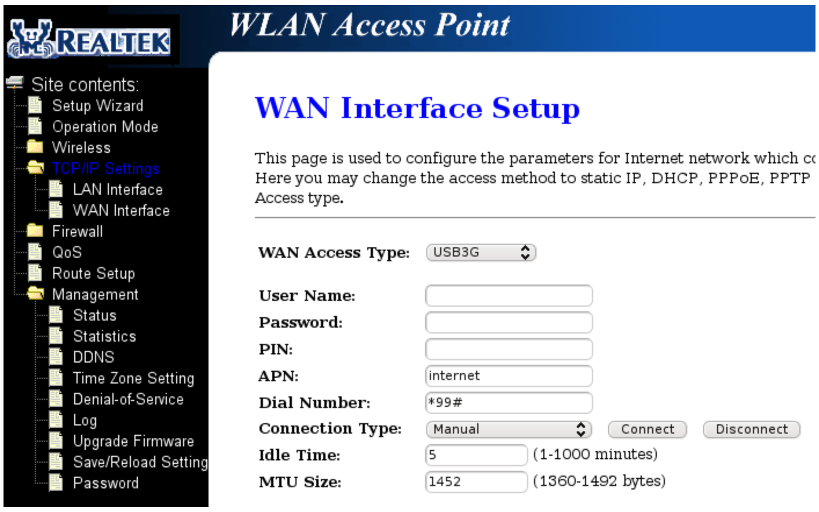


```
########################################################
# newer modems

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.



WAN Interface Setup

This page is used to configure the parameters for Internet network which co
Here you may change the access method to static IP, DHCP, PPPoE, PPTP
Access type.

WAN Access Type:	USB3G
User Name:	
Password:	
PIN:	
APN:	internet
Dial Number:	*99#
Connection Type:	Manual
Idle Time:	5 (1-1000 minutes)
MTU Size:	1452 (1360-1492 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:

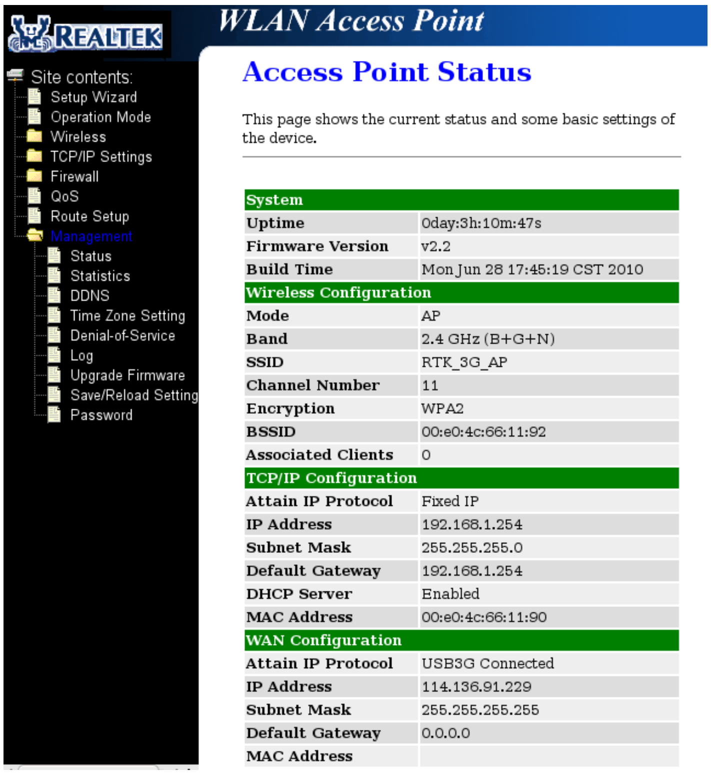
① B3G Removed: there is no usb dongle on the DUT.

② 3G Modem Initializing...: the usb dongle has been inserted, and ment is starting eject process.

③ 3G Dialing...: ejected and attached as modem device. Trying to establish a PPP connection.

④ 3G Connected: PPP connection has been established.

⑤ 3G Disconnected: PPP connection hang up.



Access Point Status

This page shows the current status and some basic settings of the device.

System	
Uptime	0day:3h:10m:47s
Firmware Version	v2.2
Build Time	Mon Jun 28 17:45:19 CST 2010
Wireless Configuration	
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
Associated Clients	0
TCP/IP Configuration	
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
MAC Address	00:e0:4c:66:11:90
WAN Configuration	
Attain IP Protocol	USB3G Connected
IP Address	114.136.91.229
Subnet Mask	255.255.255.255
Default Gateway	0.0.0.0
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)

4.5 WAPI support

4.5.1 Enable WAPI support

make linux_menucofig // 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.	None.	1) No security: 16136KB.

	User: root.bin 1007634. Total: fw.bin 1933934.		
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, readFileSize 4708, loadWapiFiles 10956, storeWapiFiles 17320, aeUdpClient 41136, aseUdpServer 31544, genUserCert.sh 2808, initCAFiles.sh 1917, revokeUserCert.sh 1063,	1) No security: 15500KB. 2) WAPI-PSK: 15440KB. 3) WAPI-CERT(remote AS): 15312KB. 4) WAPI-CERT(local AS): 13316KB.

		WAPI certs related 64K. Total: 2424079Byte.	

4.6 Mesh support (not support now)

make linux_menucofig // 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

make linux_menucofig // To configure linux kernel settings

Menuconfig:

```
Device Drivers --->
    Network device support --->
        Options for Realtek SoC --->
        Support rtk vlan feature     // selected if vlan support
```

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 menucofig // To configure bootloader settings

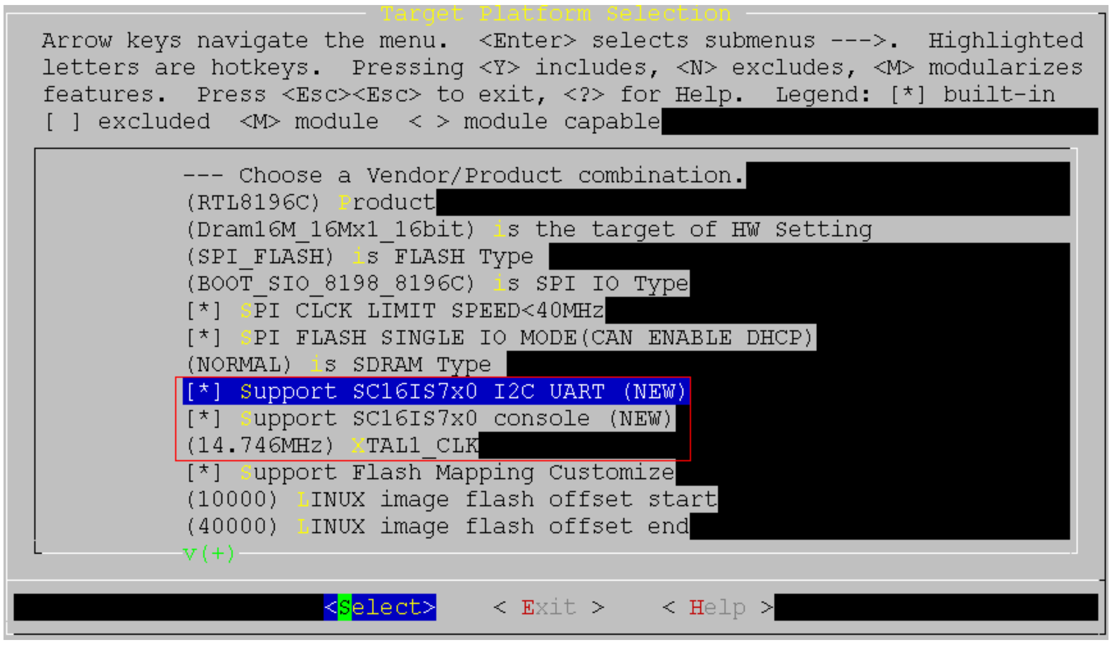
Menuconfig:

Target Platform Selection --->

Support SC16IS7x0 I2C UART // selected

Support SC16IS7x0 console // selected

(14.746MHz) XTAL1_CLK // XTAL1_CLK should be set based on hardware



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

          --- Choose a Vendor/Product combination.
          (RTL8196C) Product
          (Dram16M_16Mx1_16bit) is the target of HW Setting
          (SPI_FLASH) is FLASH Type
          (BOOT_SIO_8198_8196C) is SPI IO Type
          [*] SPI CLCK LIMIT SPEED<40MHz
          [*] SPI FLASH SINGLE IO MODE(CAN ENABLE DHCP)
          (NORMAL) is SDRAM Type
          [*] Support SC16IS7x0 I2C UART (NEW)
          [*] Support SC16IS7x0 console (NEW)
          (14.746MHz) XTAL1_CLK
          [*] Support Flash Mapping Customize
          (10000) LINUX image flash offset start
          (40000) LINUX image flash offset end
          v(+)

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

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

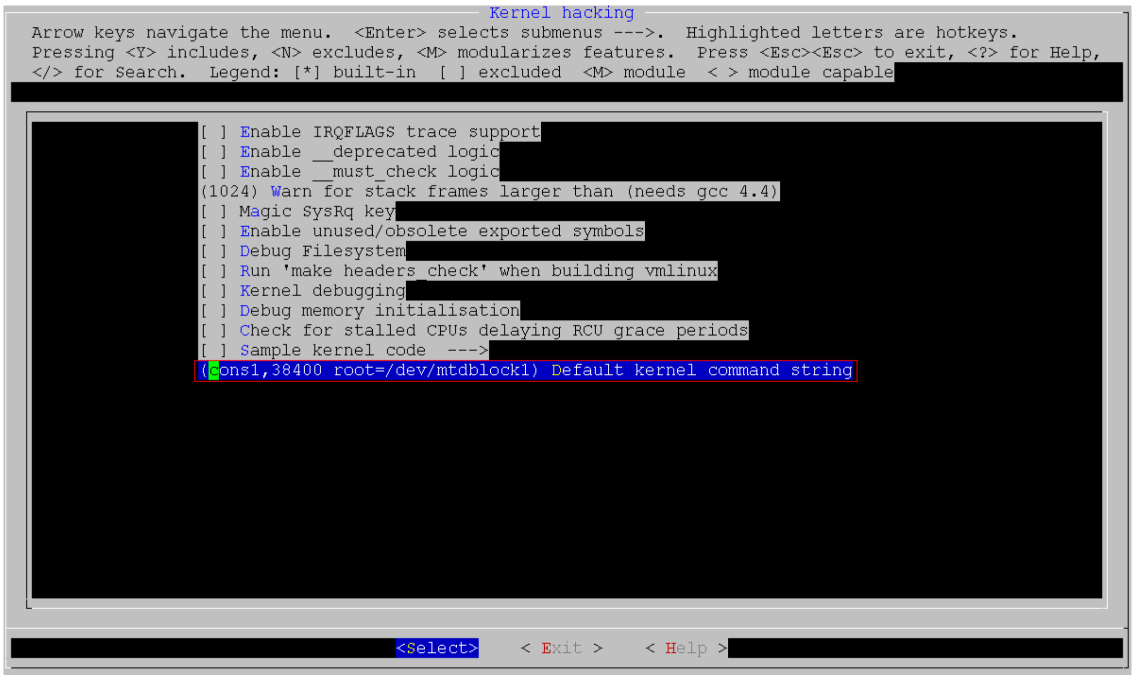
1) Set ttys1 as console tty.

make linux_menucofig // To configure linux kernel settings

Menuconfig:

Kernel hacking --->

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



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

                [ ] Enable IRQFLAGS trace support
                [ ] Enable __deprecated logic
                [ ] Enable __must_check logic
                (1024) Warn for stack frames larger than (needs gcc 4.4)
                [ ] Magic SysRq key
                [ ] Enable unused/obsolete exported symbols
                [ ] Debug Filesystem
                [ ] Run 'make headers_check' when building vmlinux
                [ ] Kernel debugging
                [ ] Debug memory initialisation
                [ ] Check for stalled CPUs delaying RCU grace periods
                [ ] Sample kernel code  --->
                (console1,38400 root=/dev/mtdblock1) Default kernel command string

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

2) Modify the 8250/16500 serial port number.

make linux_menucofig // To configure linux kernel settings

Menuconfig:

Device Drivers --->

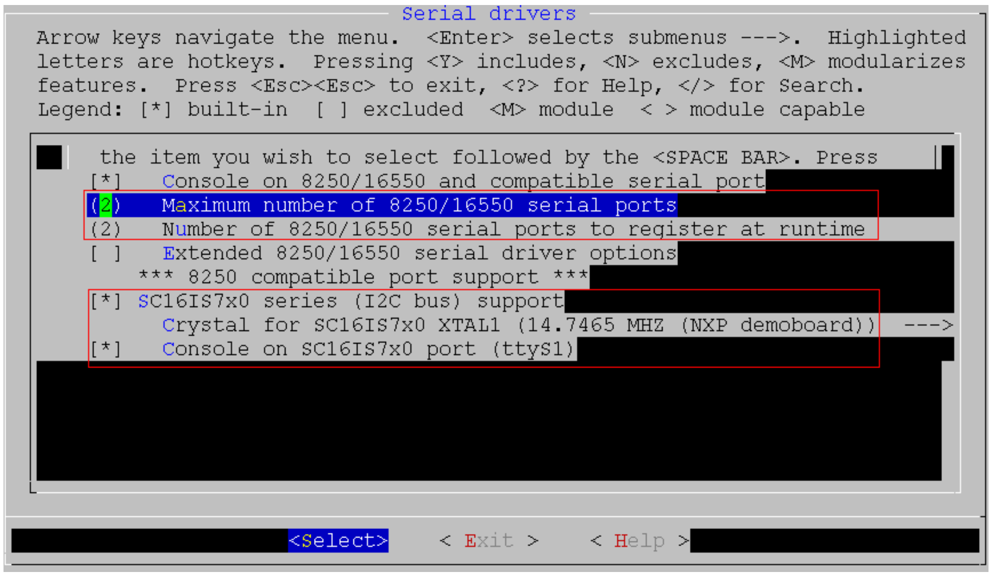
Character devices --->

Serial drivers --->

Maximum number of 8250/16550 serial ports = 2

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

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



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

     the item you wish to select followed by the <SPACE BAR>. Press
   [*]   Console on 8250/16550 and compatible serial port
   (2)   Maximum number of 8250/16550 serial ports
   (2)   Number of 8250/16550 serial ports to register at runtime
   [ ]   Extended 8250/16550 serial driver options
         *** 8250 compatible port support ***
   [*] SC16IS7x0 series (I2C bus) support
         Crystal for SC16IS7x0 XTAL1 (14.7465 MHZ (NXP demoboard))  --->
   [*]   Console on SC16IS7x0 port (ttyS1)

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

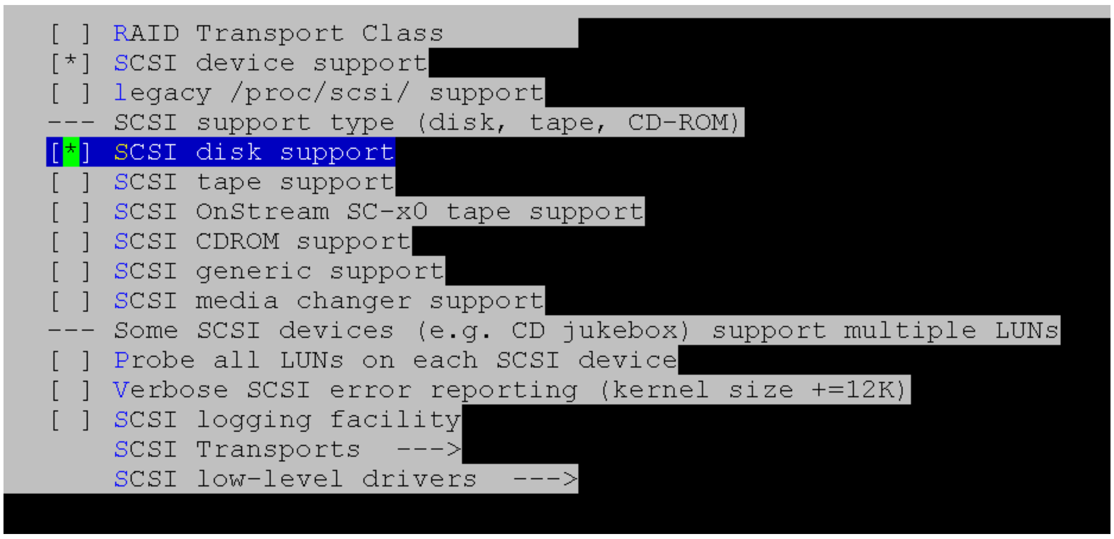
4.9 DLNA support

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

4.9.1 Linux kernel configure

4.9.1.1 SCSI support

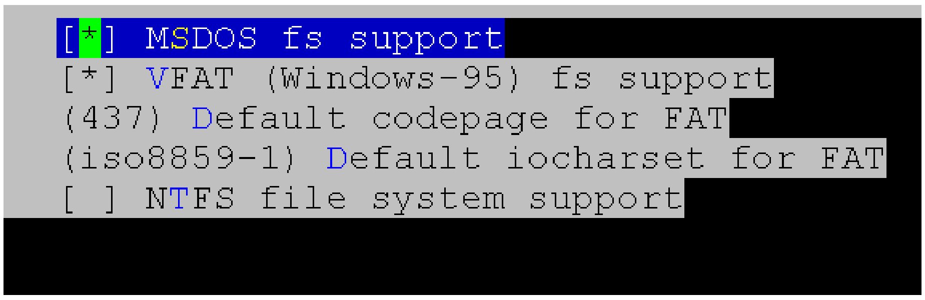
Device Drivers --->
SCSI device support --->



```
[ ] 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  --->
```

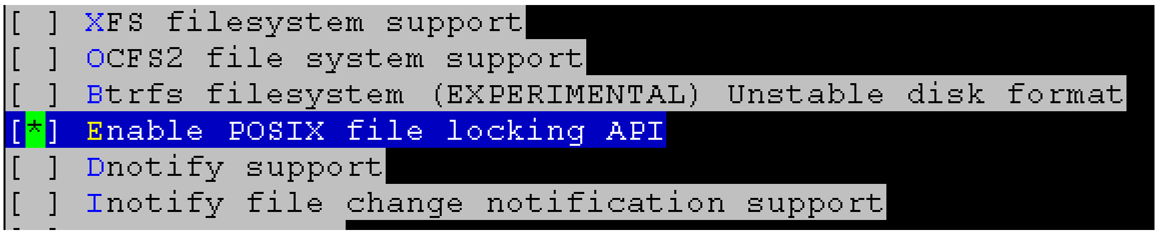
4.9.1.2 File systems configure

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



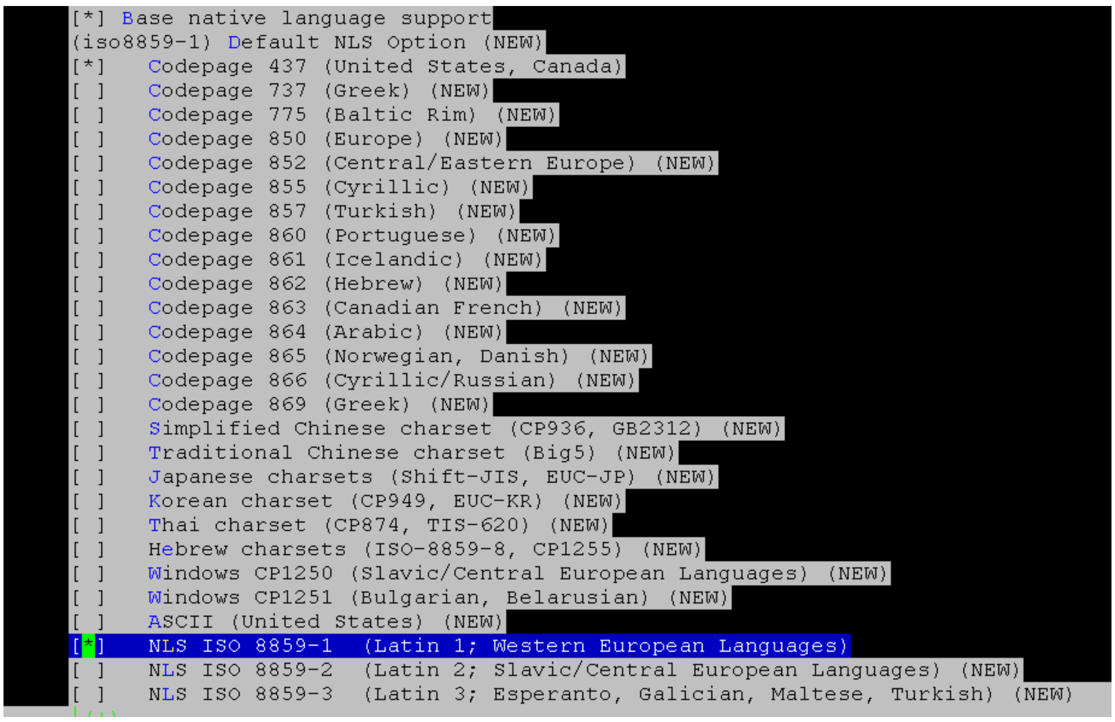
```
[*] MSDOS fs support
[*] VFAT (Windows-95) fs support
(437) Default codepage for FAT
(iso8859-1) Default iocharset for FAT
[ ] NTFS file system support
```

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)
```

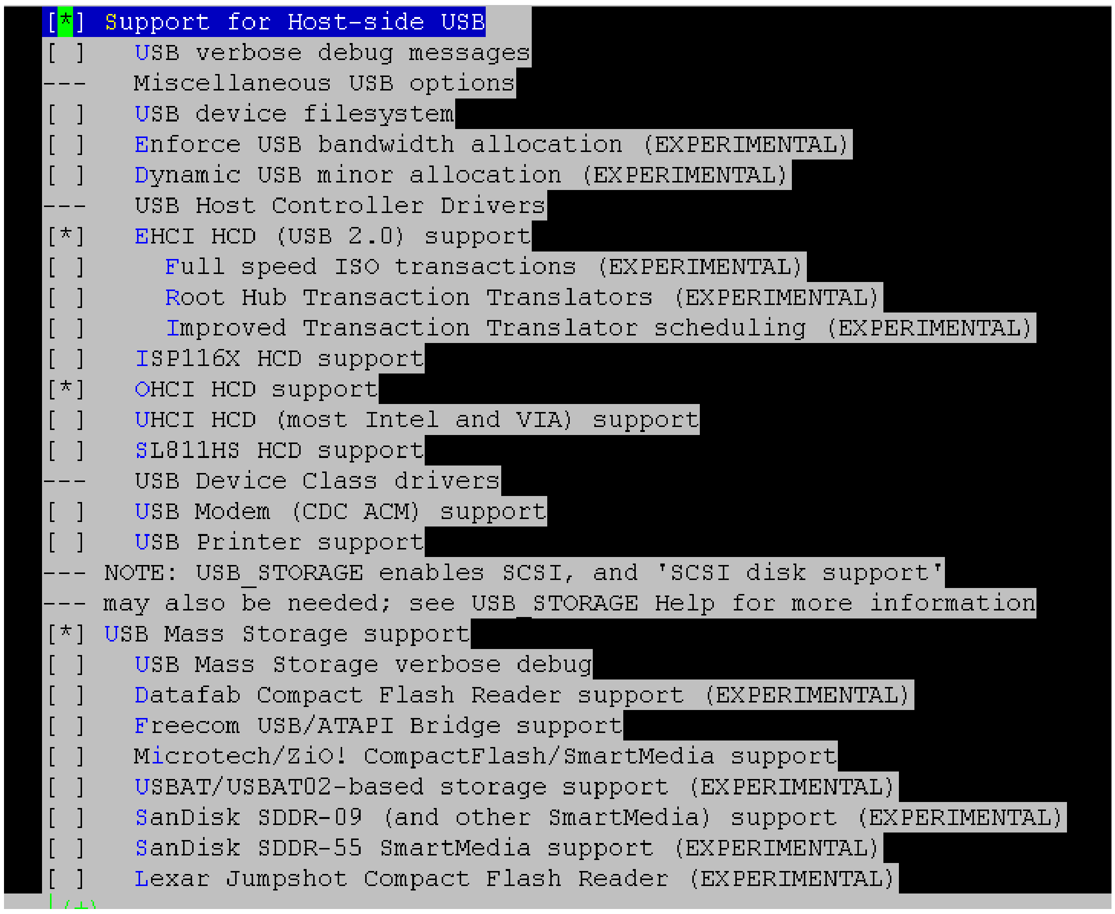
4.9.1.3 Enable USB support

Linux kernel configure for USB as follows.

make linux_menuconfig // To configure linux kernel settings

Menuconfig:

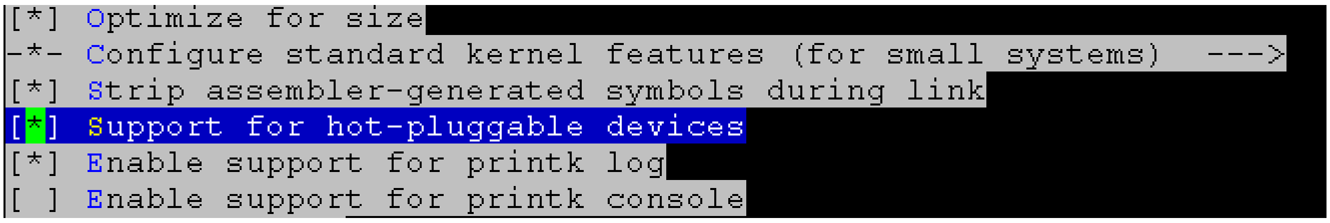
```
Device Drivers --->
    USB support --->
        Support for Host-side USB // selected
            EHCI HCD (USB 2.0) support // selected
            OHCI HCD support    // selected
            USB Mass Storage support   // selected
```



```
[*] 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

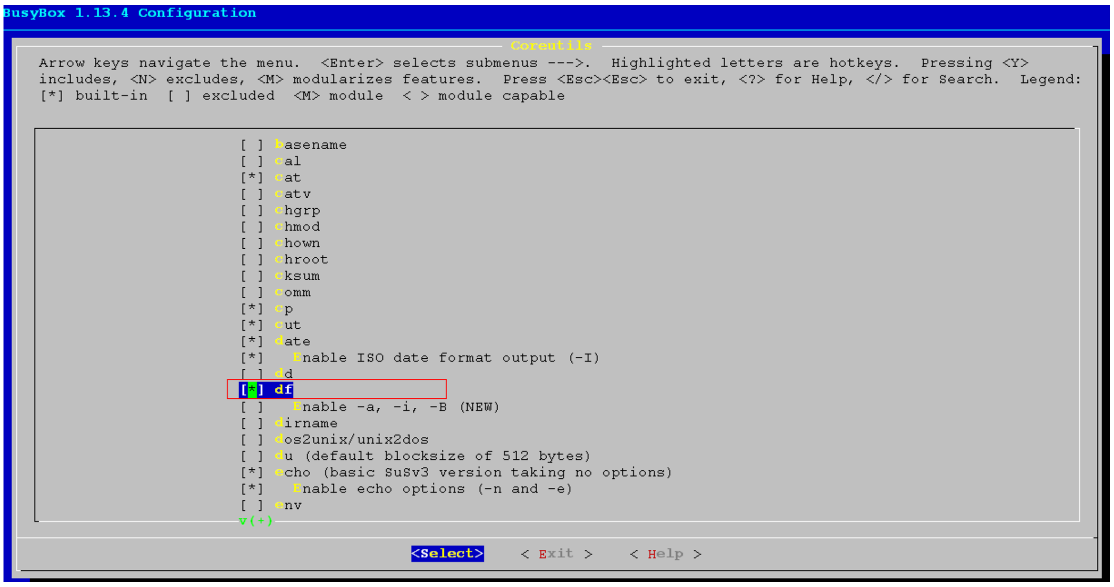
4.9.2.1 busybox configure

make menuconfig and select Config busybox

Menuconfig:

Coreutiles--->

df support // selected



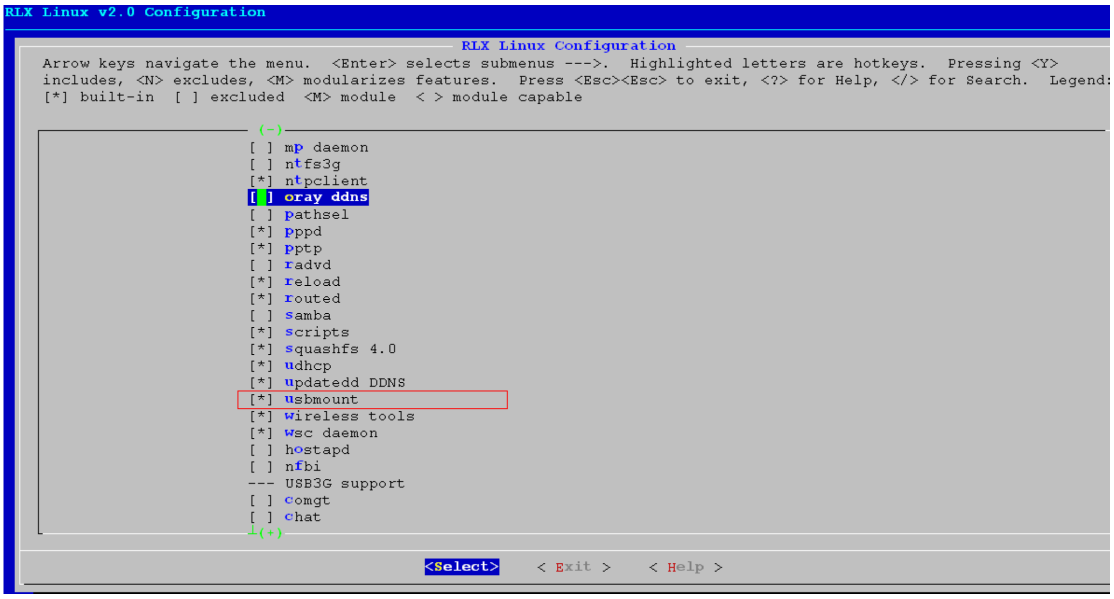
```
BusyBox 1.13.4 Configuration
                                  Coreutils
  Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted letters are hotkeys.  Pressing <Y>
  includes, <N> excludes, <M> modularizes features.  Press <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend: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

                         [ ] basename
                         [ ] cal
                         [*] cat
                         [ ] catv
                         [ ] chgrp
                         [ ] chmod
                         [ ] chown
                         [ ] chroot
                         [ ] cksum
                         [ ] comm
                         [*] cp
                         [*] cut
                         [*] date
                         [*]   Enable ISO date format output (-I)
                         [ ] dd
                         [*] df
                         [ ]   Enable -a, -i, -B (NEW)
                         [ ] dirname
                         [ ] dos2unix/unix2dos
                         [ ] du (default blocksize of 512 bytes)
                         [*] echo (basic SuSv3 version taking no options)
                         [*]   Enable echo options (-n and -e)
                         [ ] env
                         v(+)

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

4.9.2.2 enable usbmount

make users_menuconfig // To configure application settings

Menuconfig:

usbmount // selected



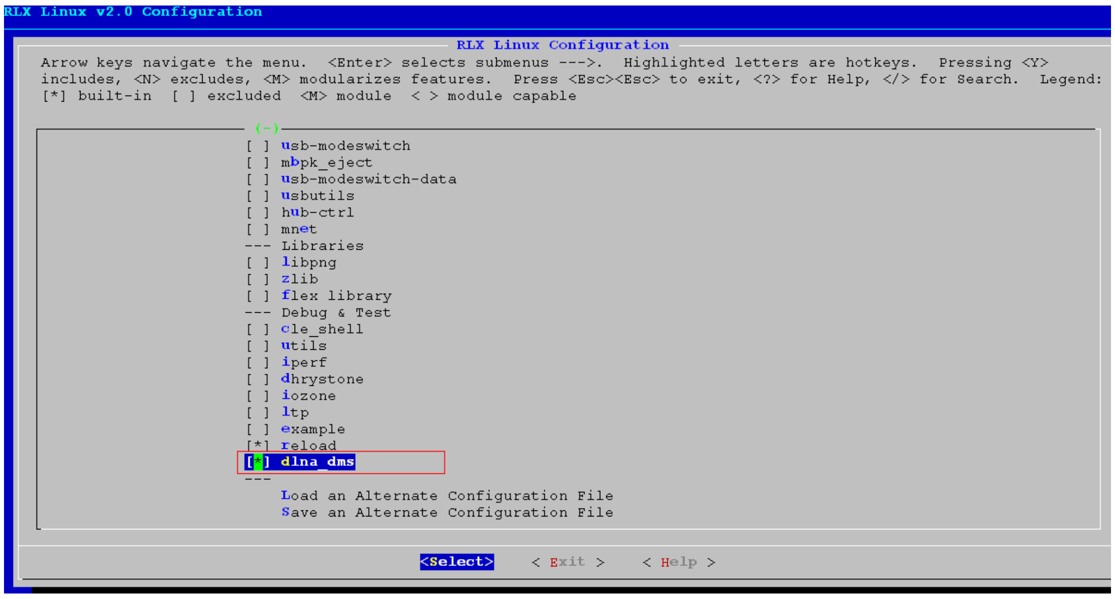
```
RLX Linux v2.0 Configuration
                                  RLX Linux Configuration
  Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted letters are hotkeys.  Pressing <Y>
  includes, <N> excludes, <M> modularizes features.  Press <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend:
  [*] built-in  [ ] excluded  <M> module  < > module capable

                          (-)
                         [ ] mp daemon
                         [ ] ntfs3g
                         [*] ntpclient
                         [ ] oray ddns
                         [ ] pathsel
                         [*] pppd
                         [*] pptp
                         [ ] radvd
                         [*] reload
                         [*] routed
                         [ ] samba
                         [*] scripts
                         [*] squashfs 4.0
                         [*] udhcp
                         [*] updatedd DDNS
                         [*] usbmount
                         [*] wireless tools
                         [*] wsc daemon
                         [ ] hostapd
                         [ ] nfbi
                         --- USB3G support
                         [ ] comgt
                         [ ] chat
                         ↓(+)

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

4.9.2.3 enable dlna

make users_menuconfig // To configure application settings

Menuconfig:

Dlna_dms // selected



```
RLX Linux v2.0 Configuration
                                          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

                      (-)
                      [ ] usb-modeswitch
                      [ ] mbpk_eject
                      [ ] usb-modeswitch-data
                      [ ] usbutils
                      [ ] hub-ctrl
                      [ ] mnet
                      --- Libraries
                      [ ] libpng
                      [ ] zlib
                      [ ] flex library
                      --- Debug & Test
                      [ ] cle_shell
                      [ ] utils
                      [ ] iperf
                      [ ] dhrystone
                      [ ] iozone
                      [ ] ltp
                      [ ] example
                      [*] reload
                      [*] dlna_dms
                      ---
                          Load an Alternate Configuration File
                          Save an Alternate Configuration File

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

4.9.3 Test dlna using USB flash disk

(1) First of all, you should read and modify /boards/rtl8198/etc.default/ushare.conf.

(2) After system boots up, plug-in an USB Flash disk. The directory to be shared is /var/tmp/usb/sda6/Media (you can modify this directory in ushare.conf).

(3) Write this command: 'ushare –f etc/ushare.conf &' in console to enable dlna as a daemon.

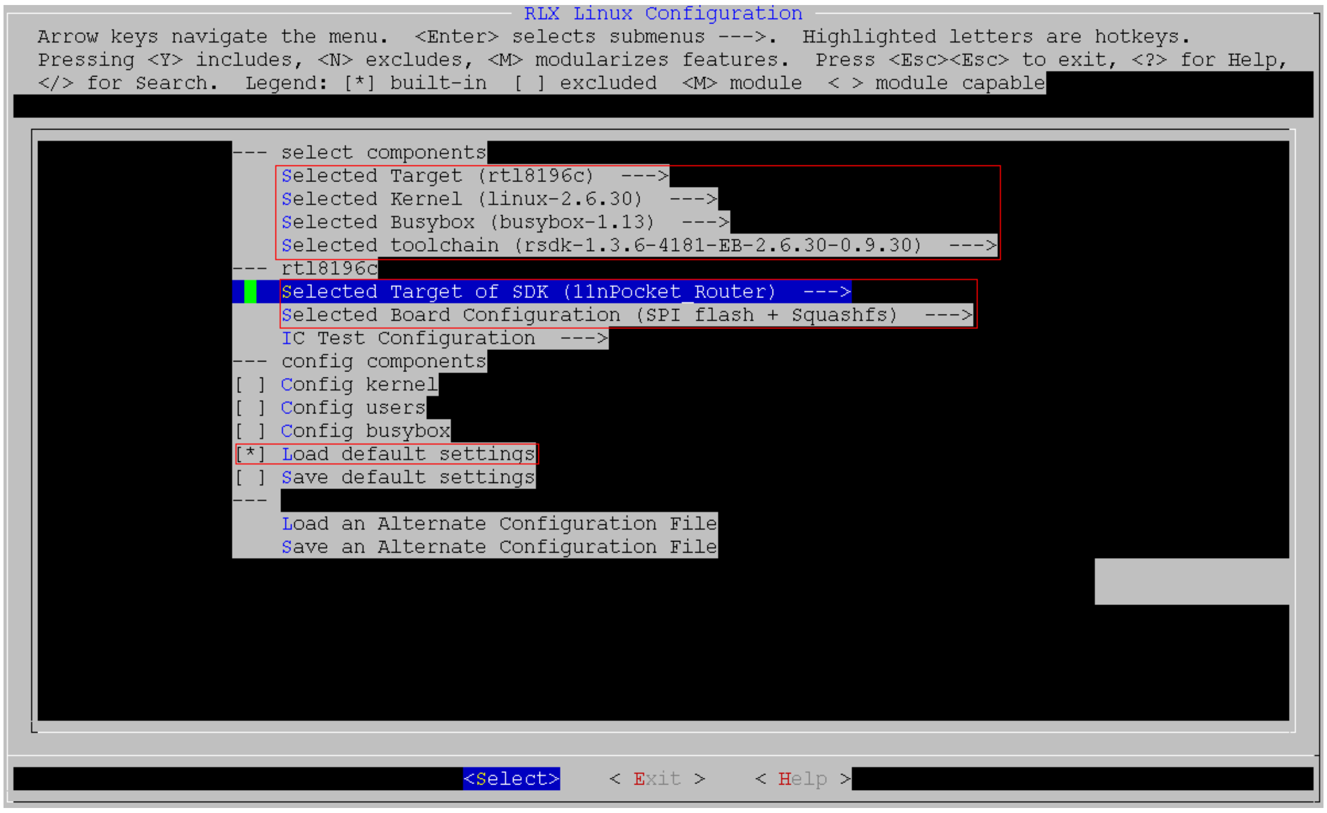
(4) Then you can enjoy the multimedia in your flashdisk with PS3 or XBOX and so on.

4.10 Pocket AP support

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

make menucofig // To configure linux kernel settings

And the linux kernel configuration refer to the pic 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 (11nPocket_Router)  --->
                    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

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

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_ menucofig // To configure linux kernel settings

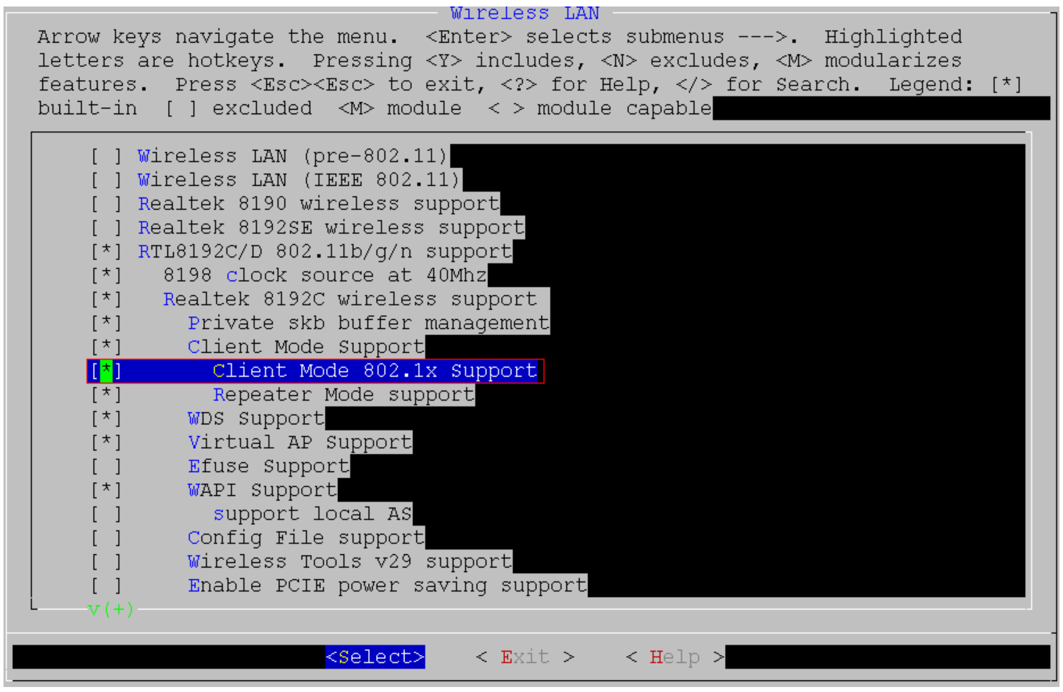
Menuconfig:

Device Drivers --->

Network device support --->

Wireless LAN --->

Client Mode 802.1x Support // selected



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

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

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

4.12 Hardware NAT

4.12.1 Hardware NAT description

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

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

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

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

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

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

4.12.2 Proc files related to hardware NAT

1) Name: fast_nat

Path: /proc/fast_nat
Description: Flags for fast path control
Input Format:
```
echo “$FLAG” > /proc/fast_nat
```
Input Para.:
* the unit is non-zero: fastpath enabled
* the unit is 0: fastpath disabled
* “echo 0 > /proc/fast_nat” : disable fastpath.
* “echo 1 > /proc/fast_nat” : enable fastpath.
* “echo 2 > /proc/fast_nat” : clean the conntrack table.

Output Format:
```
$FLAG
```
Output Para.:
* 10: disable fastpath.
* 11 : enable fastpath.
* 12 : clean the conntrack table.

2) hw_nat

Path: /proc/hw_nat

Description: Flags for hardware NAT control
Input Format:
```
echo “$FLAG” > /proc/hw_nat
```
Input Para.:
* the unit is non-zero: hardware NAT enabled
* the unit is 0: hardware NAT disabled
* “echo 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 limitation

Hardware NAT can't handle with the cases as follows:

1) PPTP/L2TP/multi-PPPoE wan type.
2) IP conntracks which need do ALG.
3) URL filter enabled or QoS enabled.
4) IP fragment packets.
5) 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.

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_ menucofig          // 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

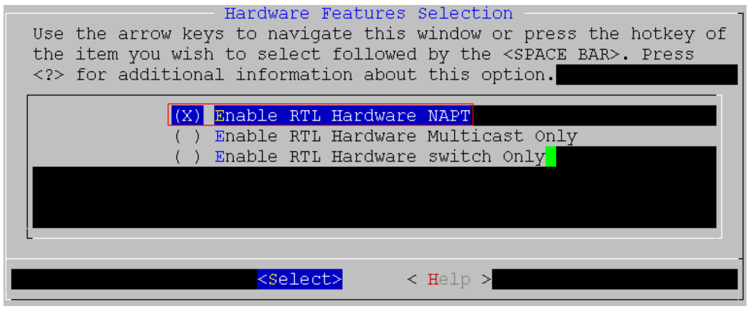
/* If selected, enable hardware L2 and multicast function */

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.



```
                    Hardware Features Selection
 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) Enable RTL Hardware NAPT
              ( ) Enable RTL Hardware Multicast Only
              ( ) Enable RTL Hardware switch Only

                      <Select>     < Help >
```

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_ menucofig // To configure linux kernel settings

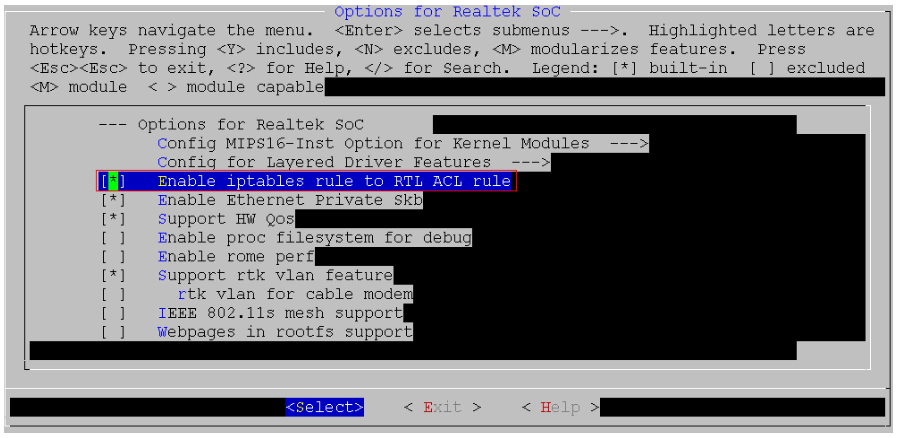
Menuconfig:

Device Drivers --->

Network device support --->

Options for Realtek SoC --->

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



```
                          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 Ethernet Private Skb
        [*]   Support HW Qos
        [ ]   Enable proc filesystem for debug
        [ ]   Enable rome perf
        [*]   Support rtk vlan feature
        [ ]     rtk vlan for cable modem
        [ ]   IEEE 802.11s mesh support
        [ ]   Webpages in rootfs support

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

4.14 Hardware QoS support

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

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

After hardware NAT (refer to section 4.12.4) and iptables rule to ACL rule translation (refer to section 4.13) are enabled, enable hardware QoS as follows:

make linux_ menucofig // To configure linux kernel settings

Menuconfig:

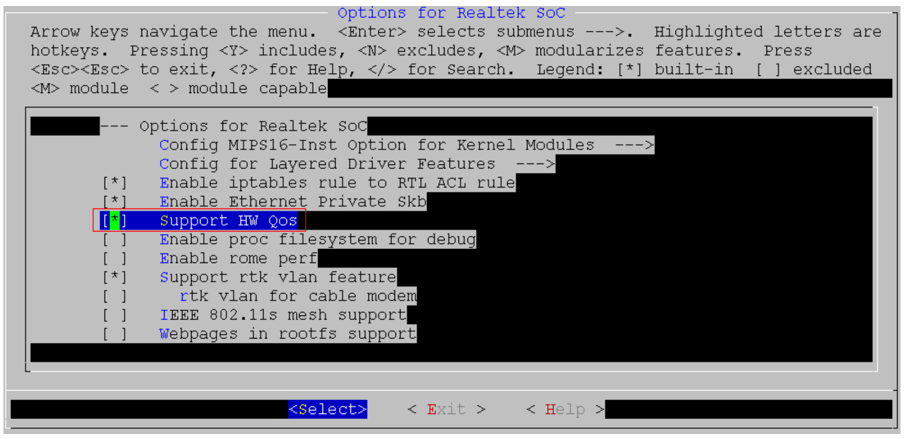
Device Drivers --->

Network device support --->

Options for Realtek SoC --->

Support HW Qos // Selected

Note: if hardware QoS is disabled, iptables rule to ACL rule translation support should be disabled too. For more details of hardware QoS, please refer to Realtek_QoS_v*.pdf.



```
                         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 Ethernet Private Skb
        [*]    Support HW Qos
        [ ]    Enable proc filesystem for debug
        [ ]    Enable rome perf
        [*]    Support rtk vlan feature
        [ ]      rtk vlan for cable modem
        [ ]    IEEE 802.11s mesh support
        [ ]    Webpages in rootfs support

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

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:

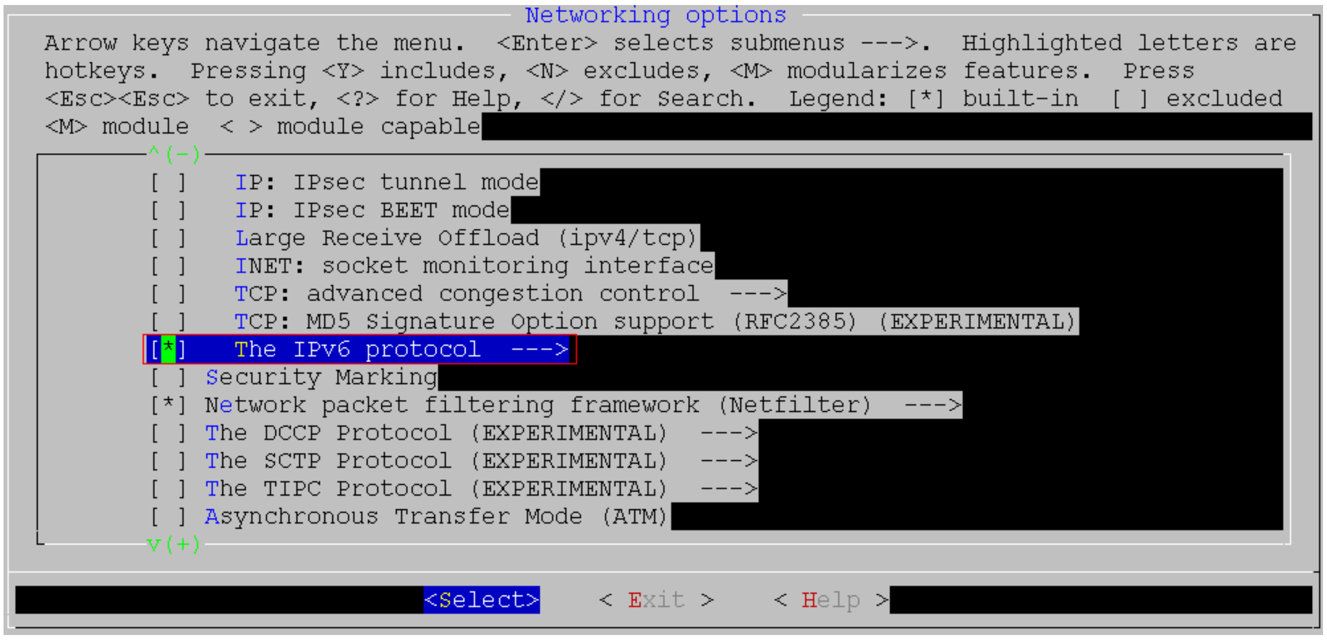
make linux_ menucofig // To configure linux kernel settings

Menuconfig:

1) Networking support --->

Networking options--->

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



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

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

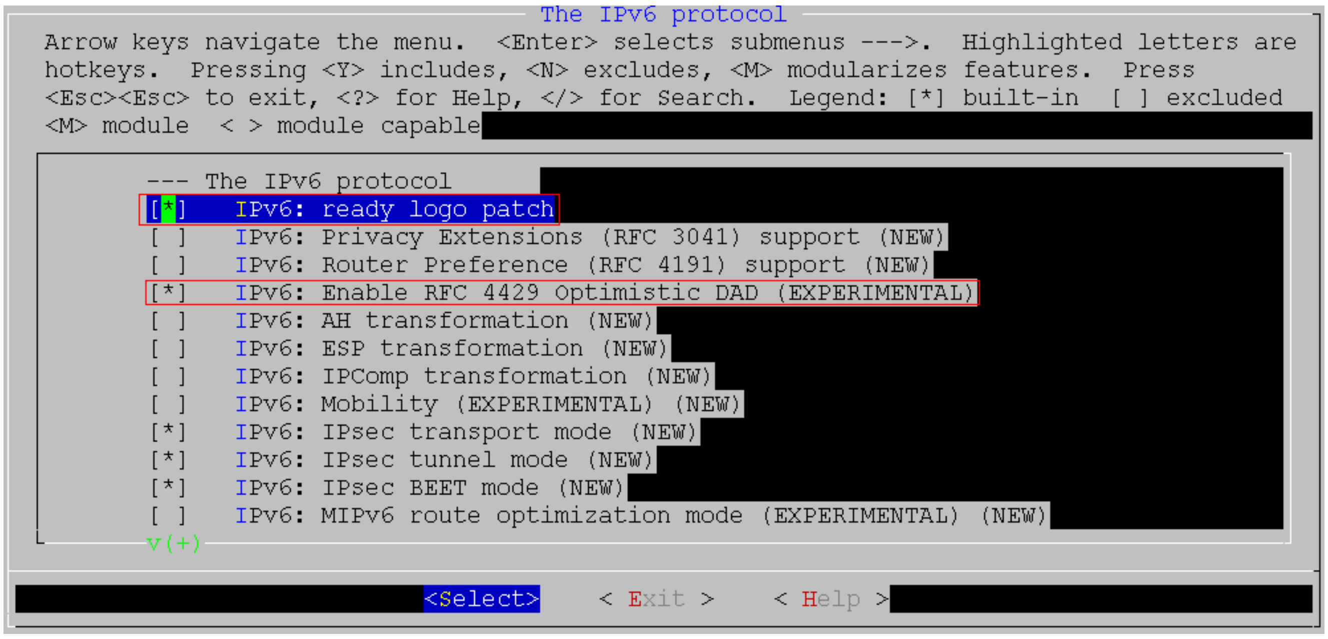
2) Networking support --->

Networking options--->

The ipv6 protocol--->

IPv6: ready logo patch // Selected

IPv6: Enable RFC 4429 Optimistic DAD (EXPERIMENTAL) // Selected



```
                          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 (NEW)
        [ ]   IPv6: Router Preference (RFC 4191) support (NEW)
        [*]   IPv6: Enable RFC 4429 Optimistic DAD (EXPERIMENTAL)
        [ ]   IPv6: AH transformation (NEW)
        [ ]   IPv6: ESP transformation (NEW)
        [ ]   IPv6: IPComp transformation (NEW)
        [ ]   IPv6: Mobility (EXPERIMENTAL) (NEW)
        [*]   IPv6: IPsec transport mode (NEW)
        [*]   IPv6: IPsec tunnel mode (NEW)
        [*]   IPv6: IPsec BEET mode (NEW)
        [ ]   IPv6: MIPv6 route optimization mode (EXPERIMENTAL) (NEW)
        v(+)

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

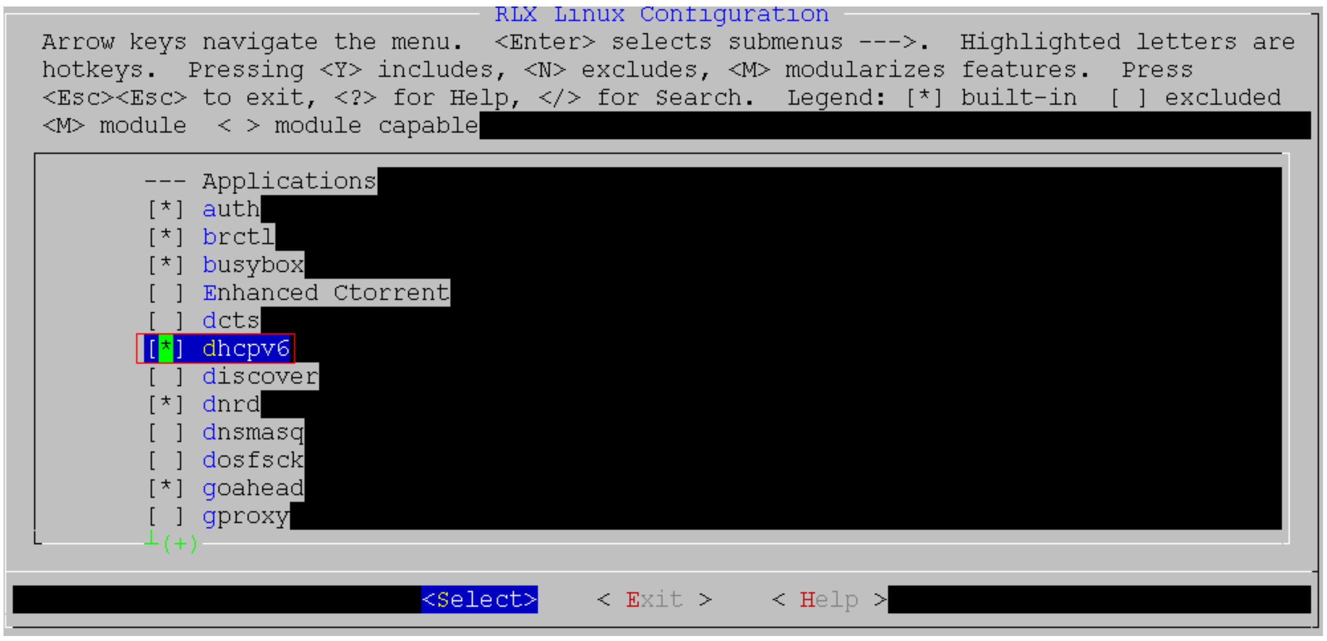
4.15.2 IPv6 daemon configure

Configure IPv6 daemon as follows:

make users_ menucofig // To configure application settings

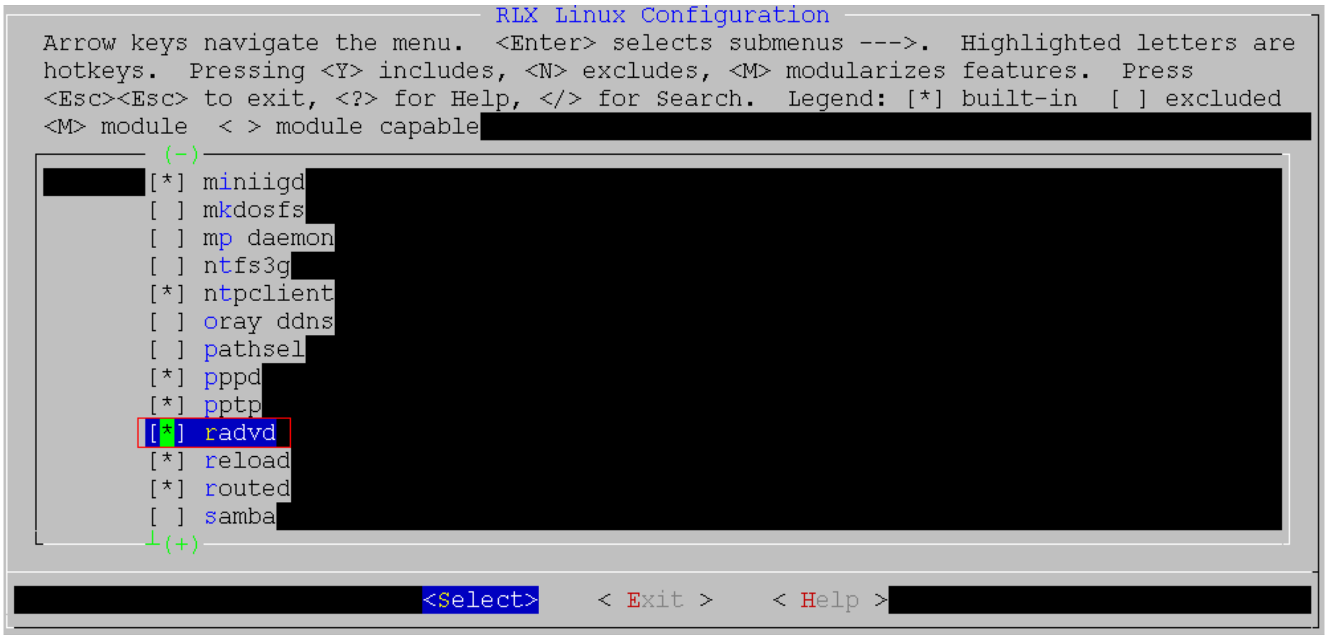
Menuconfig:

1) dhcpv6 // 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

          --- Applications
          [*] auth
          [*] brctl
          [*] busybox
          [ ] Enhanced Ctorrent
          [ ] dcts
          [*] dhcpv6
          [ ] discover
          [*] dnrd
          [ ] dnsmasq
          [ ] dosfsck
          [*] goahead
          [ ] gproxy
          ┴(+)

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

2) radvd // 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
           (-)
          [*] miniigd
          [ ] mkdosfs
          [ ] mp daemon
          [ ] ntfs3g
          [*] ntpclient
          [ ] oray ddns
          [ ] pathsel
          [*] pppd
          [*] pptp
          [*] radvd
          [*] reload
          [*] routed
          [ ] samba
          ┴(+)

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

4.15.3 Busybox for IPv6 configure

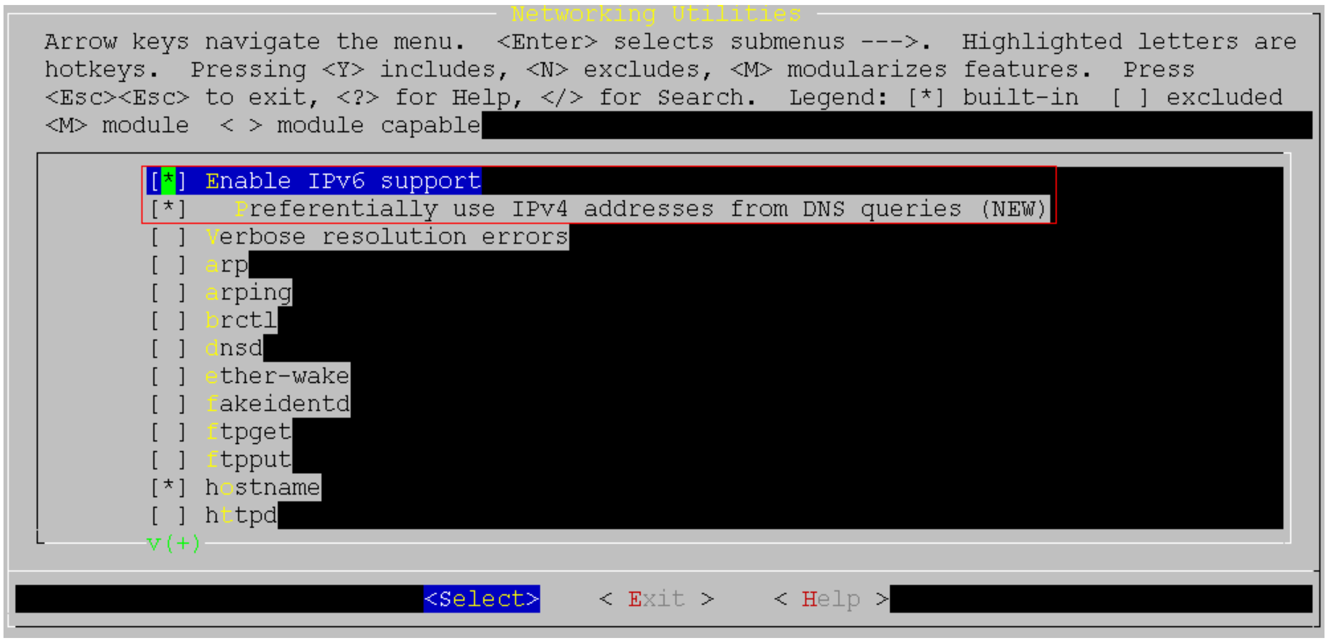
Enter the directory of Busybox and make menuconfig to configure Busybox for IPv6 as follows:

Menuconfig:

1) Networking Utilities --->

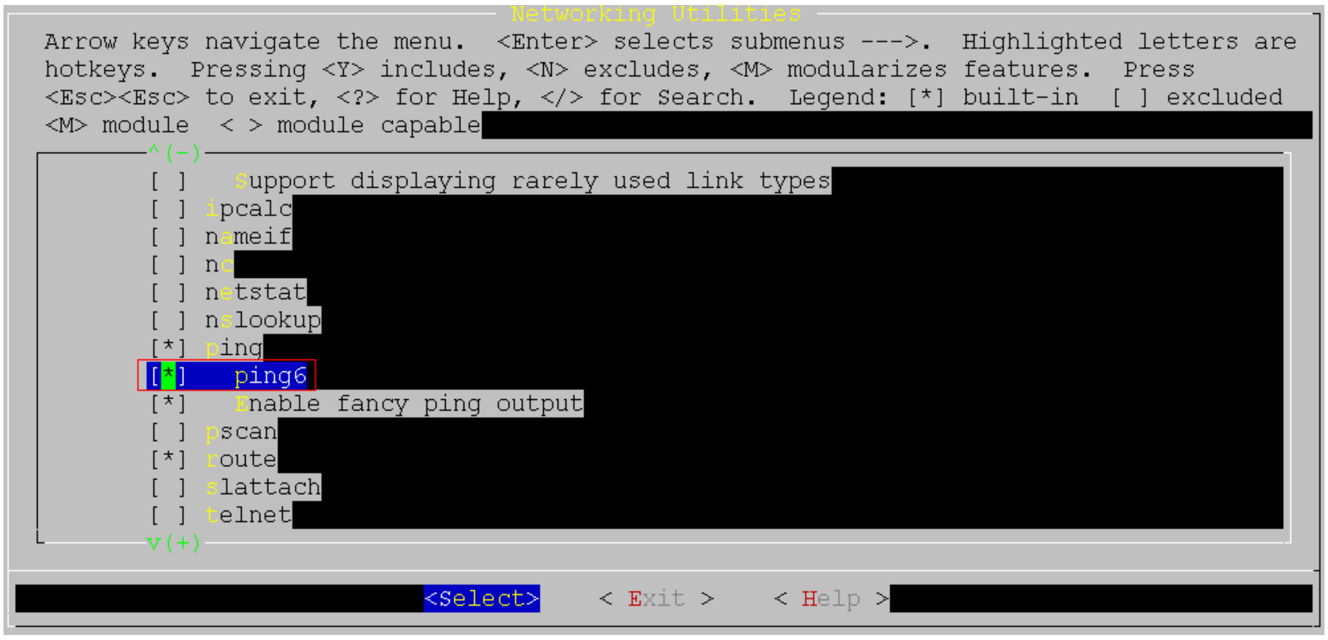
Enable IPv6 support // Selected

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


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

        [*] Enable IPv6 support
        [*]   Preferentially use IPv4 addresses from DNS queries (NEW)
        [ ] Verbose resolution errors
        [ ] arp
        [ ] arping
        [ ] brctl
        [ ] dnsd
        [ ] ether-wake
        [ ] fakeidentd
        [ ] ftpget
        [ ] ftpput
        [*] hostname
        [ ] httpd
        v(+)

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

2) Networking Utilities --->

ping6 // Selected


```
                          Networking Utilities
 Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted letters are
 hotkeys.  Pressing <Y> includes, <N> excludes, <M> modularizes features.  Press
 <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend: [*] built-in  [ ] excluded
 <M> module  < > module capable
        ^(-)
        [ ]   Support displaying rarely used link types
        [ ] ipcalc
        [ ] nameif
        [ ] nc
        [ ] netstat
        [ ] nslookup
        [*] ping
        [*]   ping6
        [*]   Enable fancy ping output
        [ ] pscan
        [*] route
        [ ] slattach
        [ ] telnet
        v(+)

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

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.

4.16 64K/sector SPI flash support

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

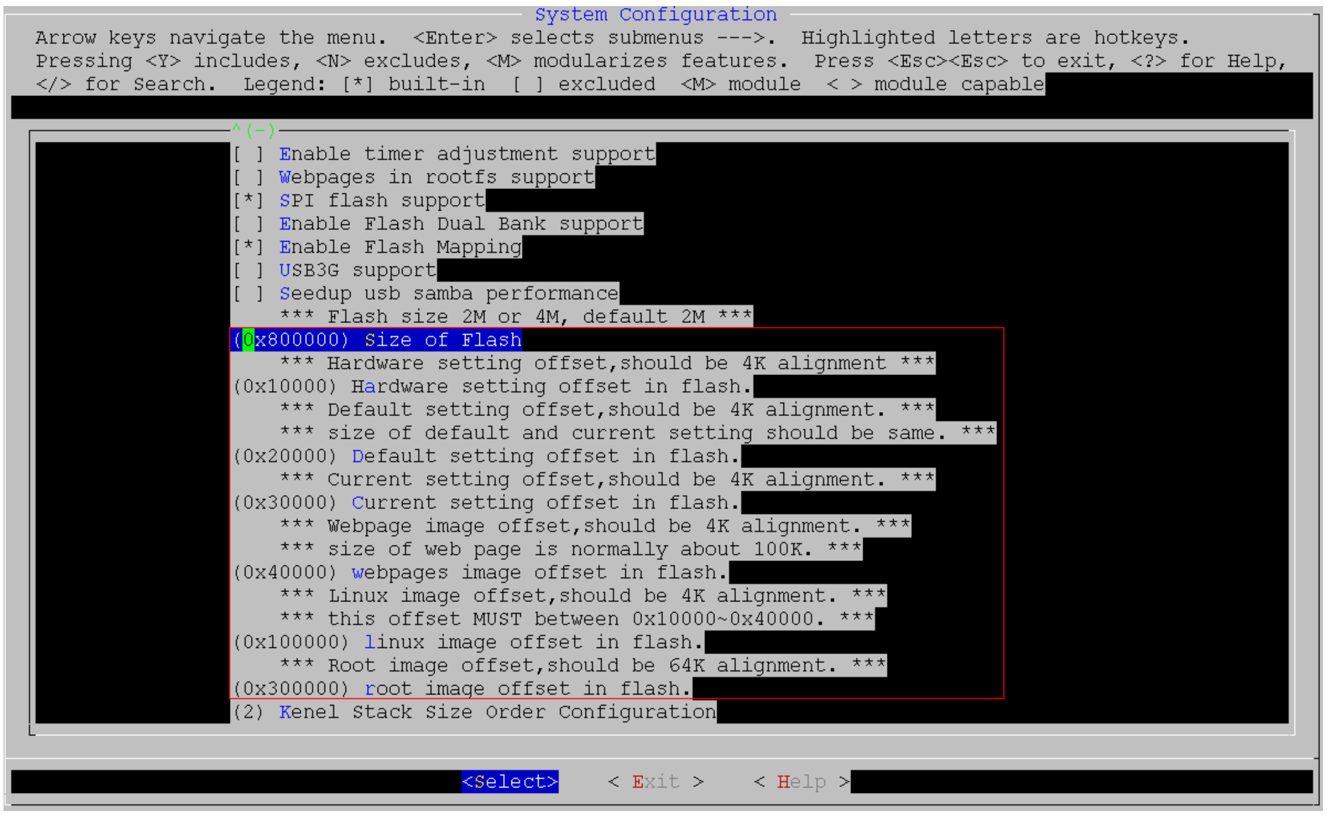
1) Linux kernel configure as follows (here we take a 64K/sector 8M SPI flash as an example) :

make linux_ menucofig   // To configure linux kernel settings

Menuconfig:

System Configuration   --->

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



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

        ^(-)
        [ ] Enable timer adjustment support
        [ ] Webpages in rootfs support
        [*] SPI flash support
        [ ] Enable Flash Dual Bank support
        [*] Enable Flash Mapping
        [ ] USB3G support
        [ ] 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) Kenel Stack Size Order Configuration

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

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

make menucofig // 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 */

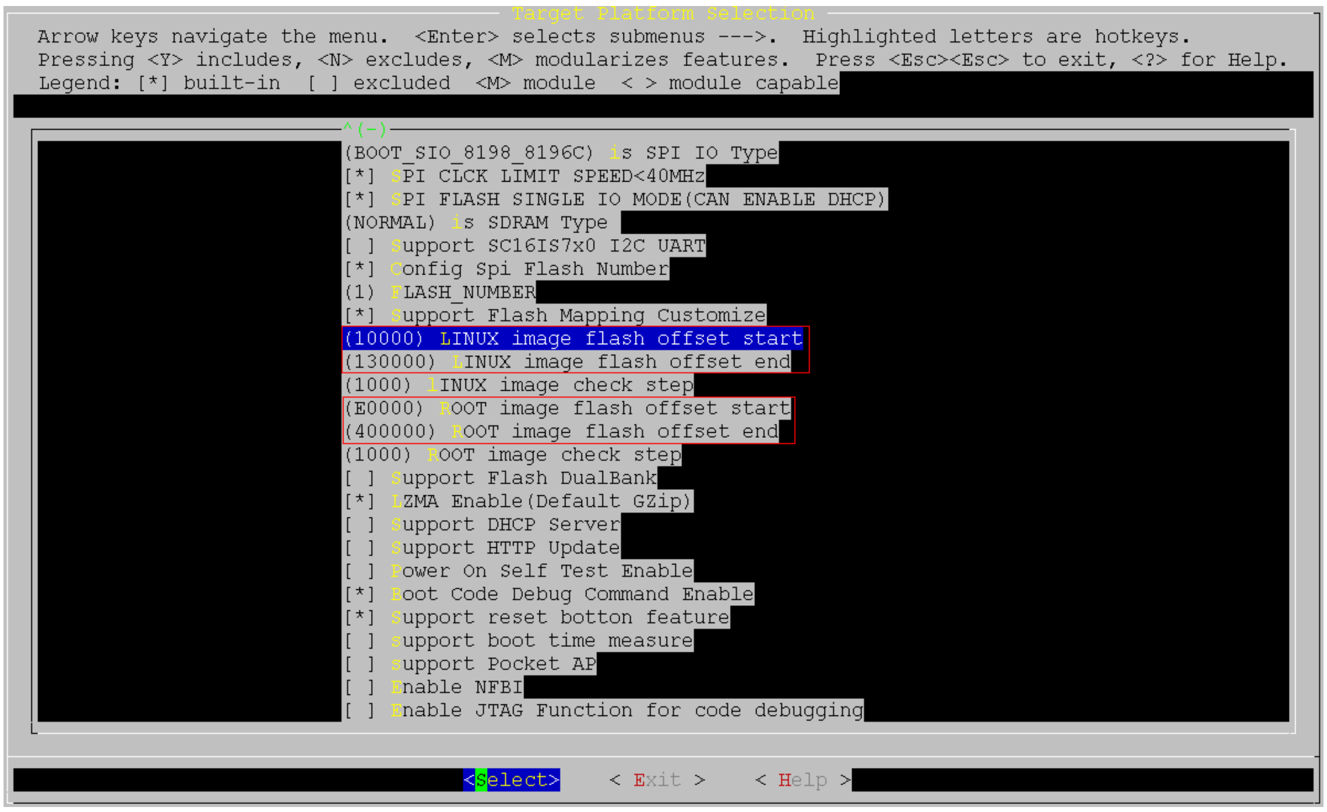
(10000) LINUX image flash offset start

(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



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

                        ^(-)
                        (BOOT_SIO_8198_8196C) is SPI IO Type
                        [*] SPI CLCK LIMIT SPEED<40MHz
                        [*] SPI FLASH SINGLE IO MODE(CAN ENABLE DHCP)
                        (NORMAL) is SDRAM Type
                        [ ] Support SC16IS7x0 I2C UART
                        [*] Config Spi Flash Number
                        (1) FLASH_NUMBER
                        [*] Support Flash Mapping Customize
                        (10000) LINUX image flash offset start
                        (130000) LINUX image flash offset end
                        (1000) LINUX image check step
                        (E0000) ROOT image flash offset start
                        (400000) ROOT image flash offset end
                        (1000) ROOT image check step
                        [ ] Support Flash DualBank
                        [*] LZMA Enable(Default GZip)
                        [ ] Support DHCP Server
                        [ ] Support HTTP Update
                        [ ] Power On Self Test Enable
                        [*] Boot Code Debug Command Enable
                        [*] Support reset botton feature
                        [ ] support boot time measure
                        [ ] support Pocket AP
                        [ ] Enable NFBI
                        [ ] Enable JTAG Function for code debugging

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

4.17 Two SPI flash support

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

4.17.1 Configure bootloader for two SPI flash support

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

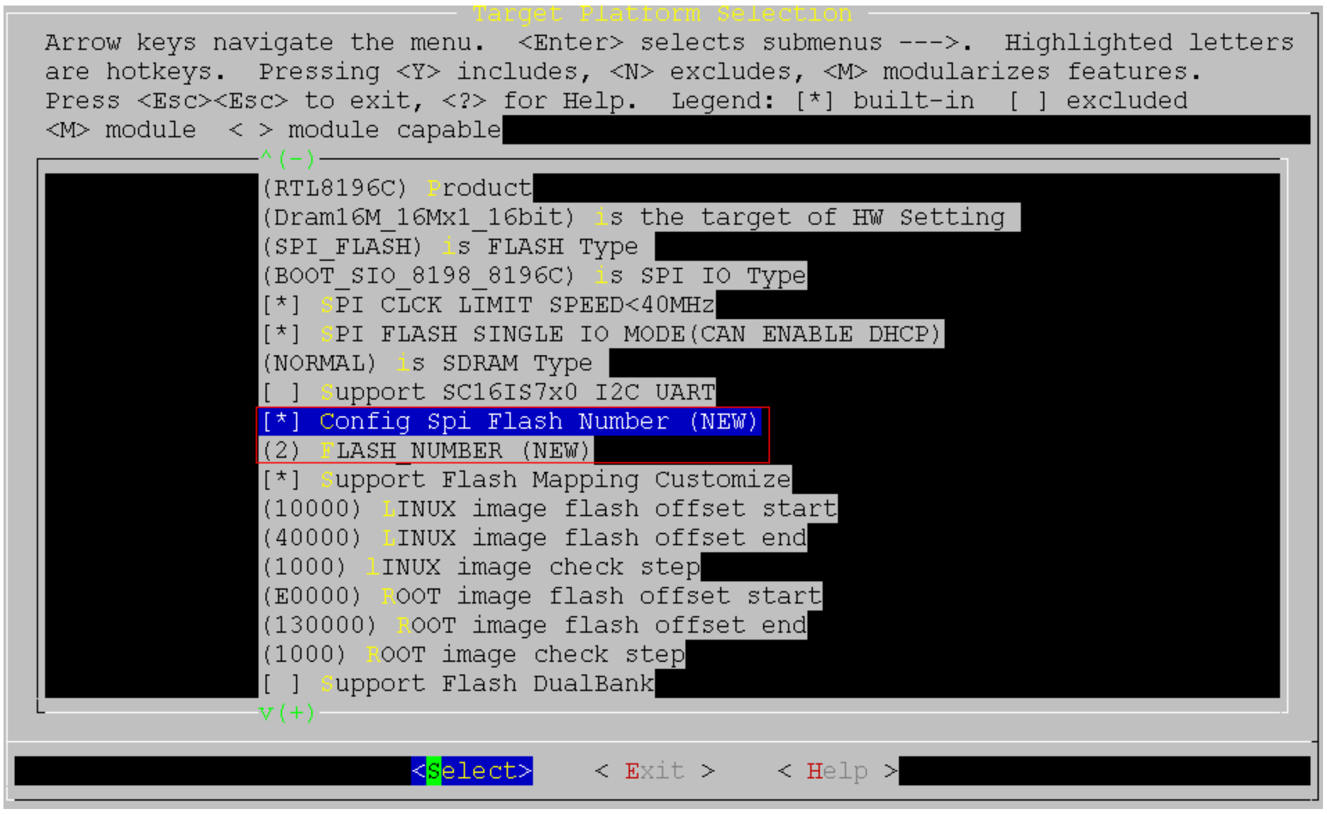
Target Platform Selection --->

//Selected to add more than one SPI flash support

[*] Config Spi Flash Number (NEW)

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

(2) FLASH_NUMBER (NEW)



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

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

4.17.2 Configure Linux kernel for two SPI flash support

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

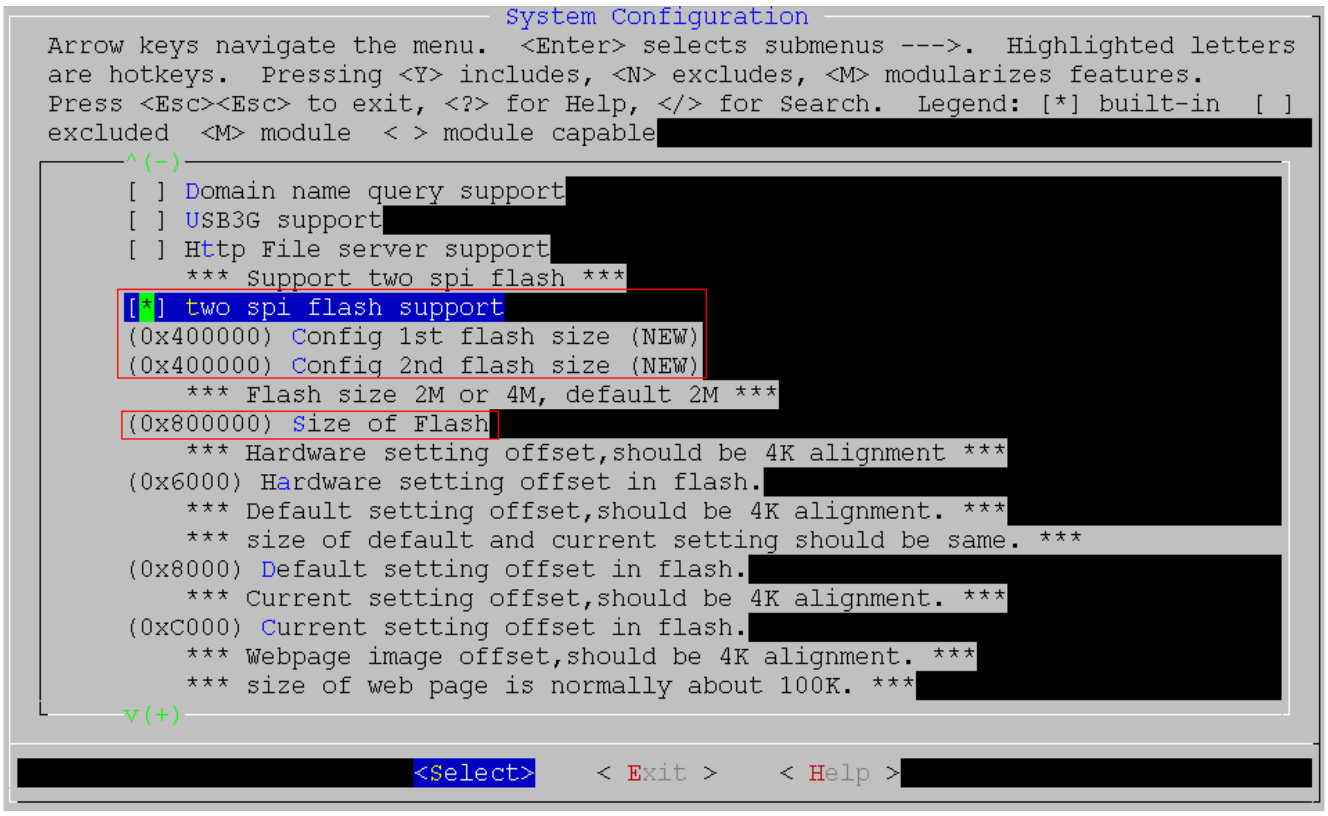
System Configuration   --->

[*] 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


```
                          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
          (0x400000) Config 1st flash size (NEW)
          (0x400000) Config 2nd flash size (NEW)
              *** Flash size 2M or 4M, default 2M ***
          (0x800000) 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. ***
          v(+)

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

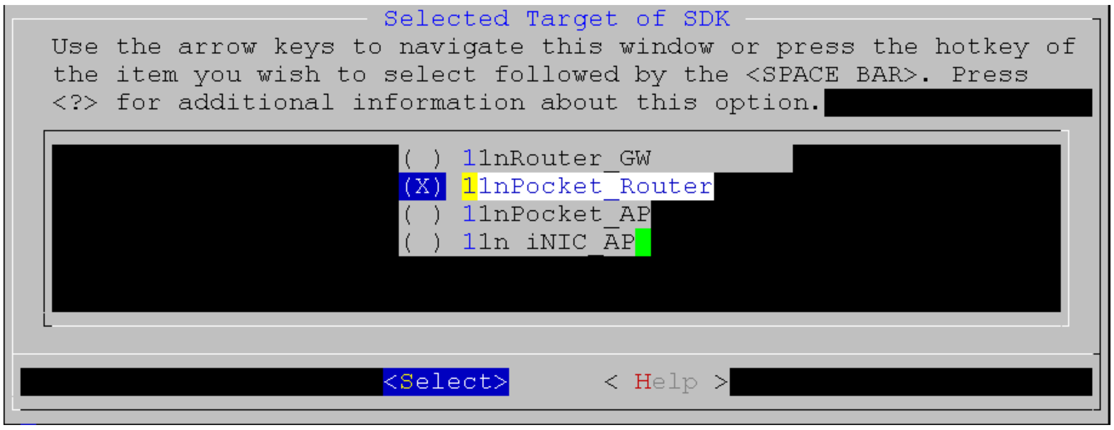
4.18 92C/D support

4.18.1 92C support for Pocket AP SDK

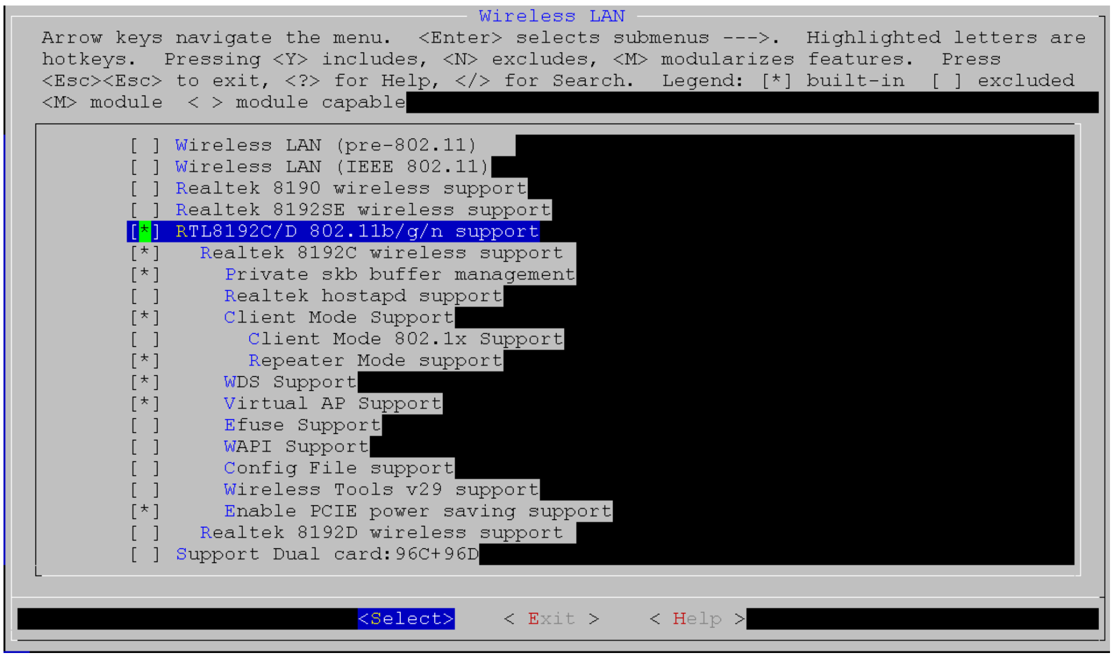
1) Select SDK for Pocket AP 92C:

Selected Target of SDK ---->

And selected "11nPocket_Router"


```
                         Selected Target of SDK
  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.

                     ( ) 11nRouter_GW
                     (X) 11nPocket_Router
                     ( ) 11nPocket_AP
                     ( ) 11n iNIC AP

                         <Select>      < Help >
```

2) Enable 92C Driver



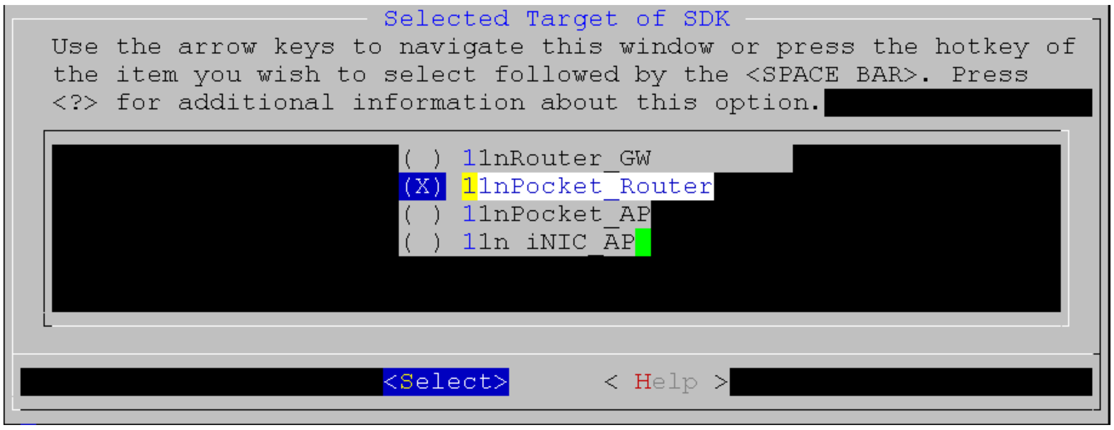
```
                          Wireless LAN
 Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted letters are
 hotkeys.  Pressing <Y> includes, <N> excludes, <M> modularizes features.  Press
 <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend: [*] built-in  [ ] excluded
 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

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

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

4.18.2 92D support for Pocket AP SDK

1) Select SDK for Pocket AP 92D:

Selected Target of SDK ---->

And selected "11nPocket_Router"



```
                     Selected Target of SDK
 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.

                   ( ) 11nRouter_GW
                   (X) 11nPocket_Router
                   ( ) 11nPocket_AP
                   ( ) 11n iNIC AP

                   <Select>      < Help >
```

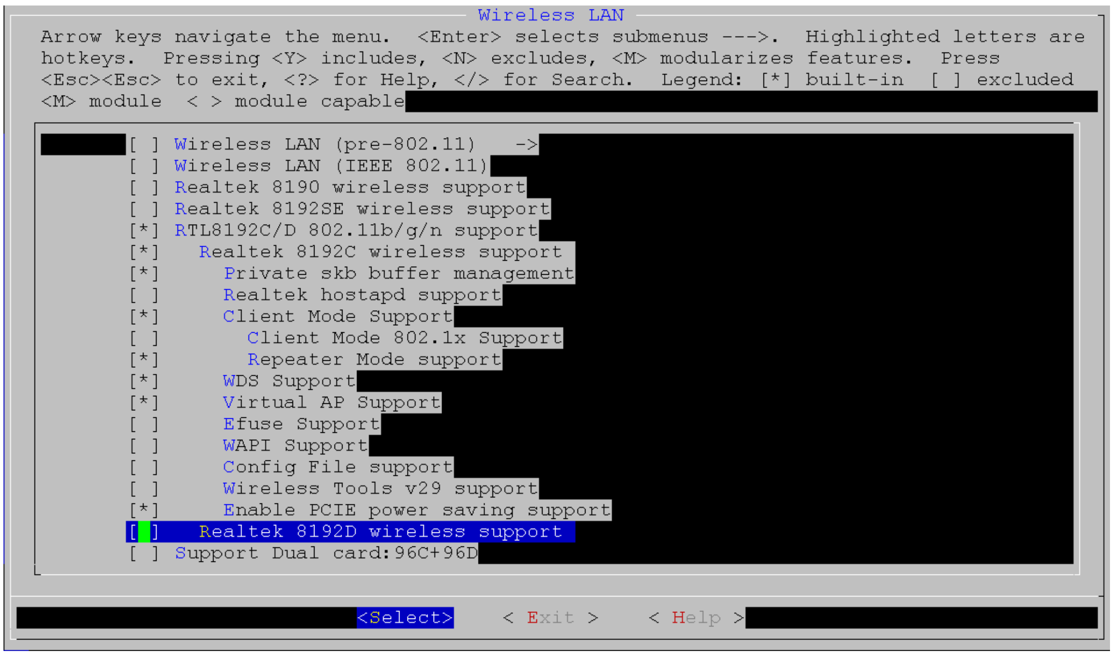
2) Enable 92D Driver

Config kernel --->

Device Drivers --->

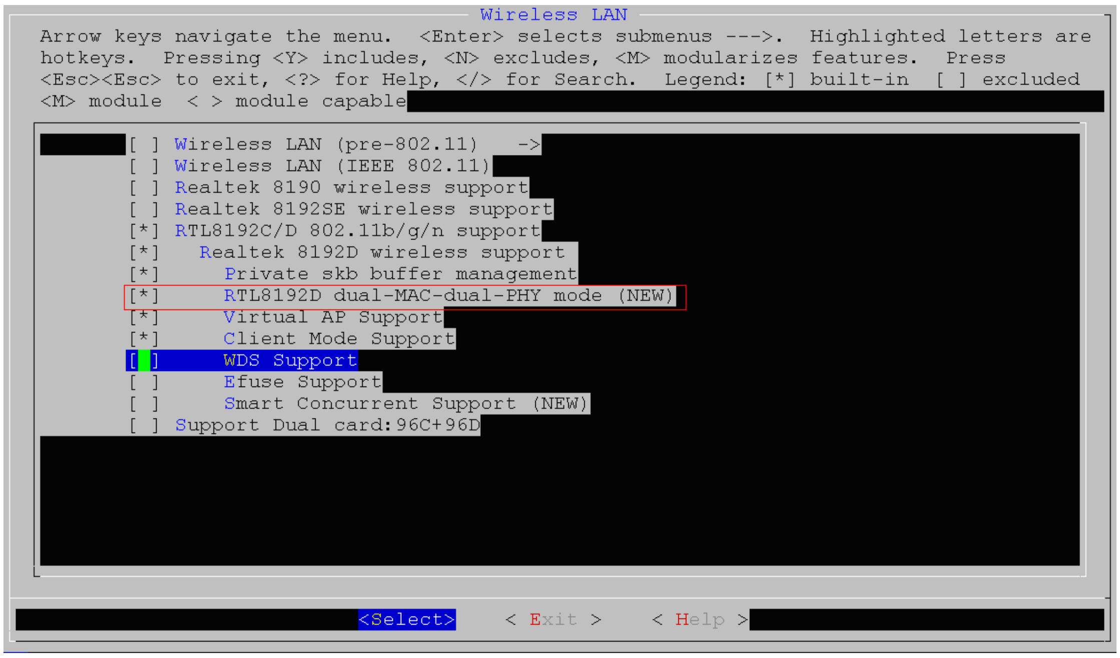
Network device support --->

Wireless LAN --->



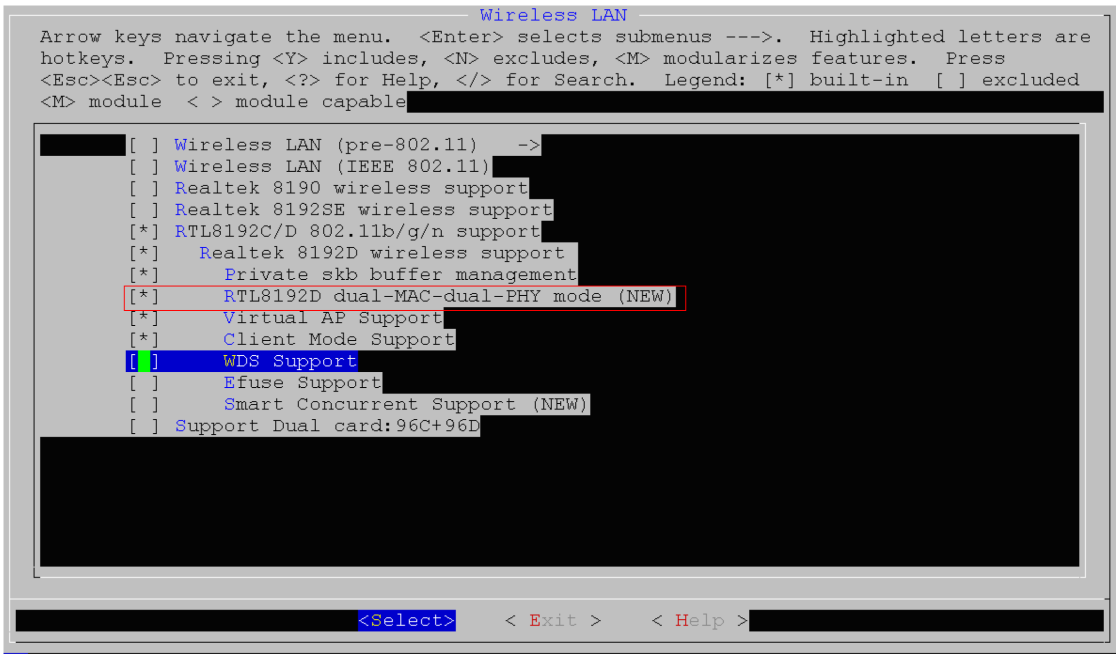
```
                              Wireless LAN
 Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted letters are
 hotkeys.  Pressing <Y> includes, <N> excludes, <M> modularizes features.  Press
 <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend: [*] built-in  [ ] excluded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

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

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

Selected "Realtek 8192D wireless support"



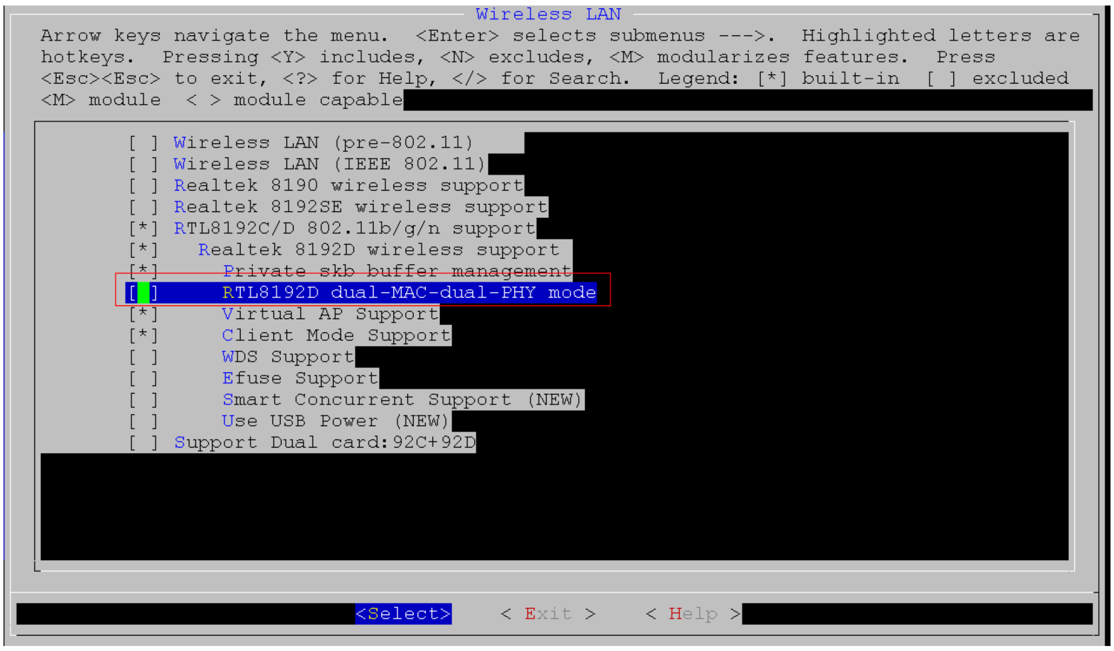
```
                              Wireless LAN
 Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted letters are
 hotkeys.  Pressing <Y> includes, <N> excludes, <M> modularizes features.  Press
 <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend: [*] built-in  [ ] excluded
 <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)
       [ ] Realtek 8190 wireless support
       [ ] Realtek 8192SE wireless support
       [*] RTL8192C/D 802.11b/g/n support
       [*]   Realtek 8192D wireless support
       [*]     Private skb buffer management
       [ ]     RTL8192D dual-MAC-dual-PHY mode
       [*]     Virtual AP Support
       [*]     Client Mode Support
       [ ]     WDS Support
       [ ]     Efuse Support
       [ ]     Smart Concurrent Support (NEW)
       [ ]     Use USB Power (NEW)
       [ ] Support Dual card:92C+92D

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

Save and Make.

4.18.3 92D support for RTL8198 SDK

Enable 92D Driver

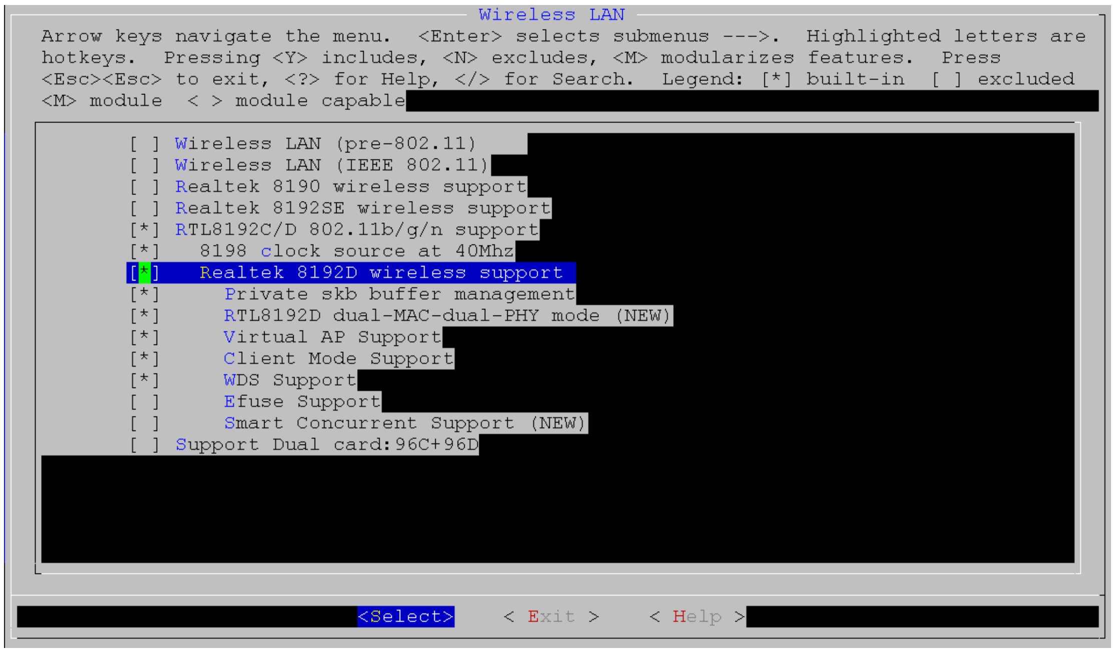
Config kernel --->

Device Drivers --->

Network device support --->

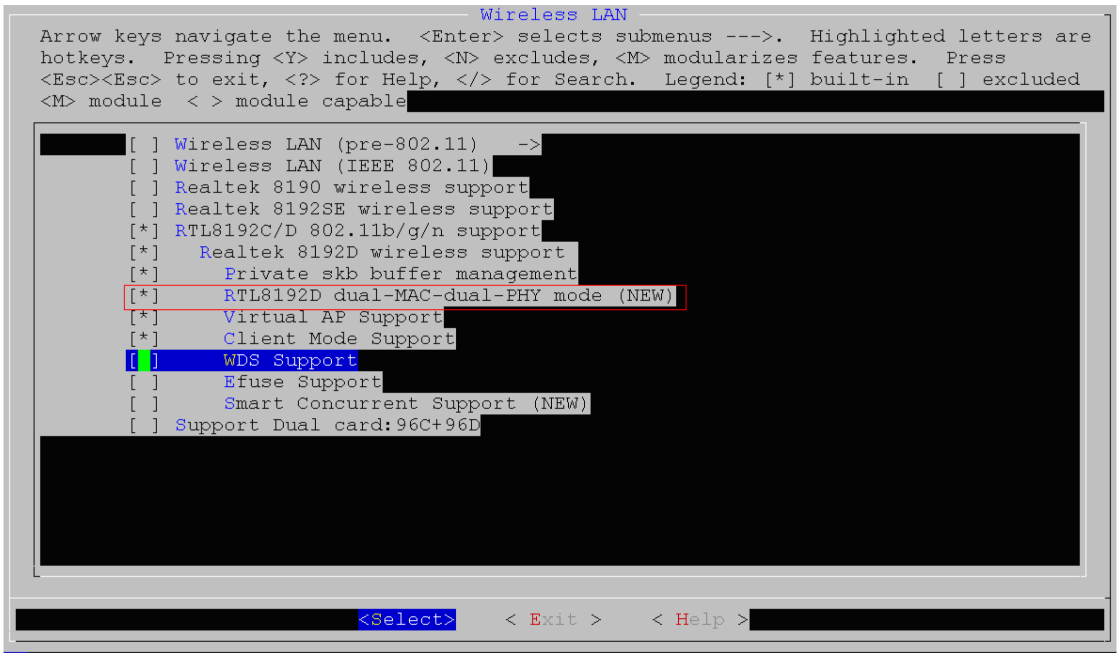
Wireless LAN --->

Enable 92D:



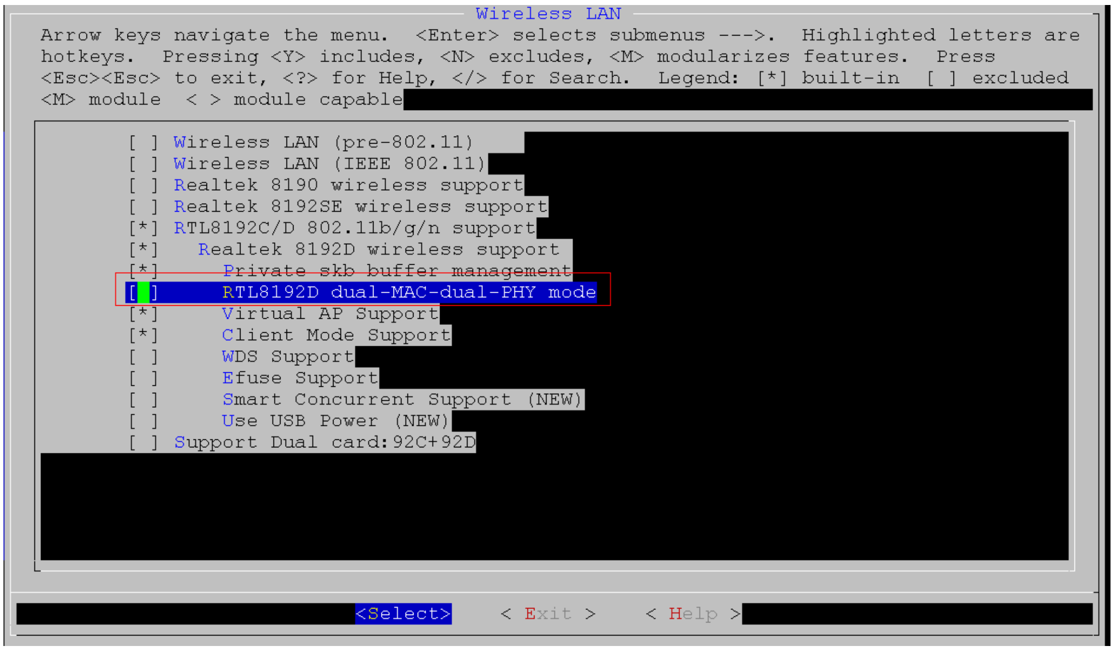
```
                                    Wireless LAN
  Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted letters are
  hotkeys.  Pressing <Y> includes, <N> excludes, <M> modularizes features.  Press
  <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend: [*] built-in  [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

         [ ] Wireless LAN (pre-802.11)
         [ ] Wireless LAN (IEEE 802.11)
         [ ] Realtek 8190 wireless support
         [ ] Realtek 8192SE wireless support
         [*] RTL8192C/D 802.11b/g/n support
         [*]   8198 clock source at 40Mhz
         [*]   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)
         [ ] Realtek 8190 wireless support
         [ ] Realtek 8192SE wireless support
         [*] RTL8192C/D 802.11b/g/n support
         [*]   Realtek 8192D wireless support
         [*]     Private skb buffer management
         [ ]     RTL8192D dual-MAC-dual-PHY mode
         [*]     Virtual AP Support
         [*]     Client Mode Support
         [ ]     WDS Support
         [ ]     Efuse Support
         [ ]     Smart Concurrent Support (NEW)
         [ ]     Use USB Power (NEW)
         [ ] Support Dual card:92C+92D

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

Save and make.

4.18.4 92C support for RTL8198 SDK

Enable 92C Driver

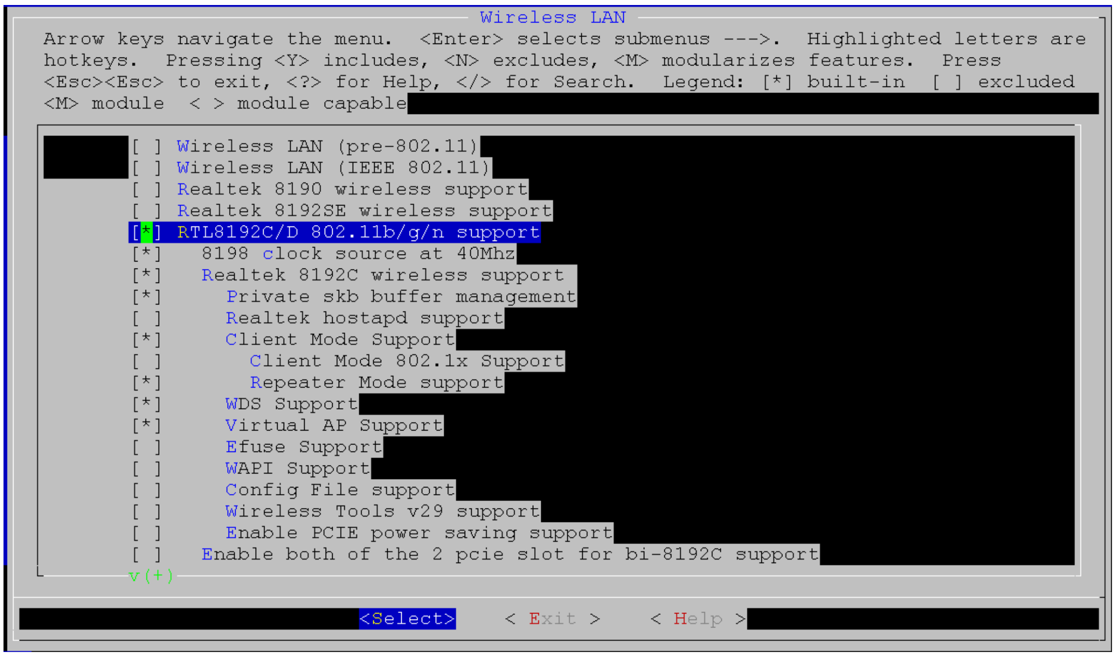
Config kernel ---->

Device Drivers ---->

Network device support ---->

Wireless LAN ---->

Enable 92C:



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

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

Save and make.

4.18.5 92C and 92D support for RTL8198 SDK

Enable 92D Driver and 92C Driver

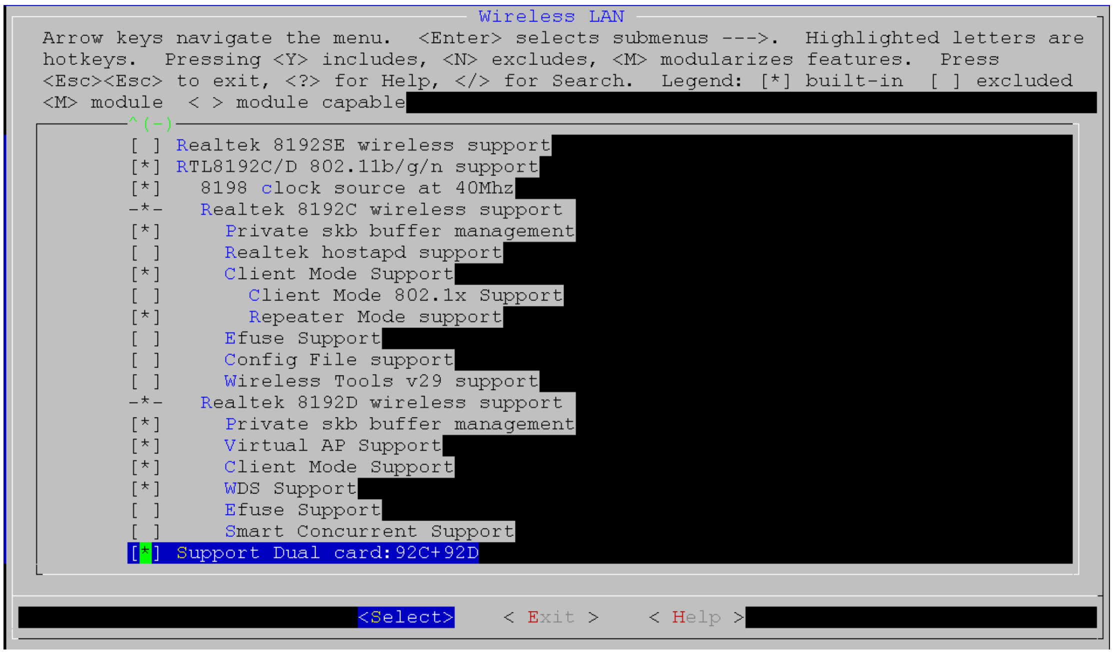
Config kernel --->

  Device Drivers --->

    Network device support --->

      Wireless LAN --->

Select “Support Dual card:92C+92D”



```
                              Wireless LAN
 Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted letters are
 hotkeys.  Pressing <Y> includes, <N> excludes, <M> modularizes features.  Press
 <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend: [*] built-in  [ ] excluded
 <M> module  < > module capable
        ^(-)
        [ ] Realtek 8192SE wireless support
        [*] RTL8192C/D 802.11b/g/n support
        [*]   8198 clock source at 40Mhz
        -*-   Realtek 8192C wireless support
        [*]     Private skb buffer management
        [ ]     Realtek hostapd support
        [*]     Client Mode Support
        [ ]       Client Mode 802.1x Support
        [*]       Repeater Mode support
        [ ]     Efuse Support
        [ ]     Config File support
        [ ]     Wireless Tools v29 support
        -*-   Realtek 8192D wireless support
        [*]     Private skb buffer management
        [*]     Virtual AP Support
        [*]     Client Mode Support
        [*]     WDS Support
        [ ]     Efuse Support
        [ ]     Smart Concurrent Support
        [*] Support Dual card:92C+92D

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

This configuration can support both 92Dvs and 92Dvc.

Save and make.

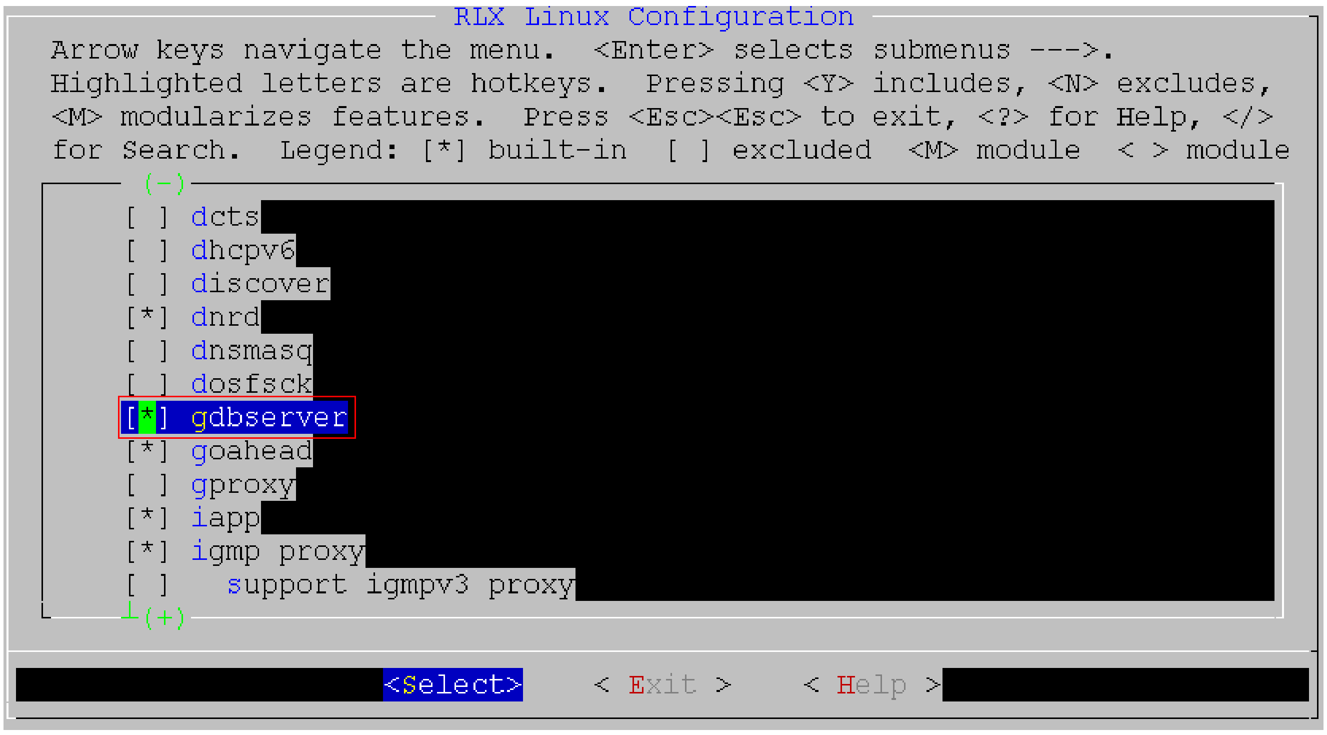
4.19 GDB server support

4.19.1 Enable GDB server

make users_menuconfig // To configure application settings

Menuconfig:

  [*] gdbserver // selected



```
                     RLX Linux Configuration
 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
          (-)
          [ ] dcts
          [ ] dhcpv6
          [ ] discover
          [*] dnrd
          [ ] dnsmasq
          [ ] dosfsck
          [*] gdbserver
          [*] goahead
          [ ] gproxy
          [*] iapp
          [*] igmp proxy
          [ ]   support igmpv3 proxy
          (+)

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

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 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 target remote <remote ip:port>

4.19.2.3 Note for GDB server usage

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

Host should ping remote succeed.

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

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

At present gdbserver6.8 is used.

4.20 HTTP file server support

4.20.1 HTTP File Server Introduction

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

4.20.2 Configuration for HTTP File Server

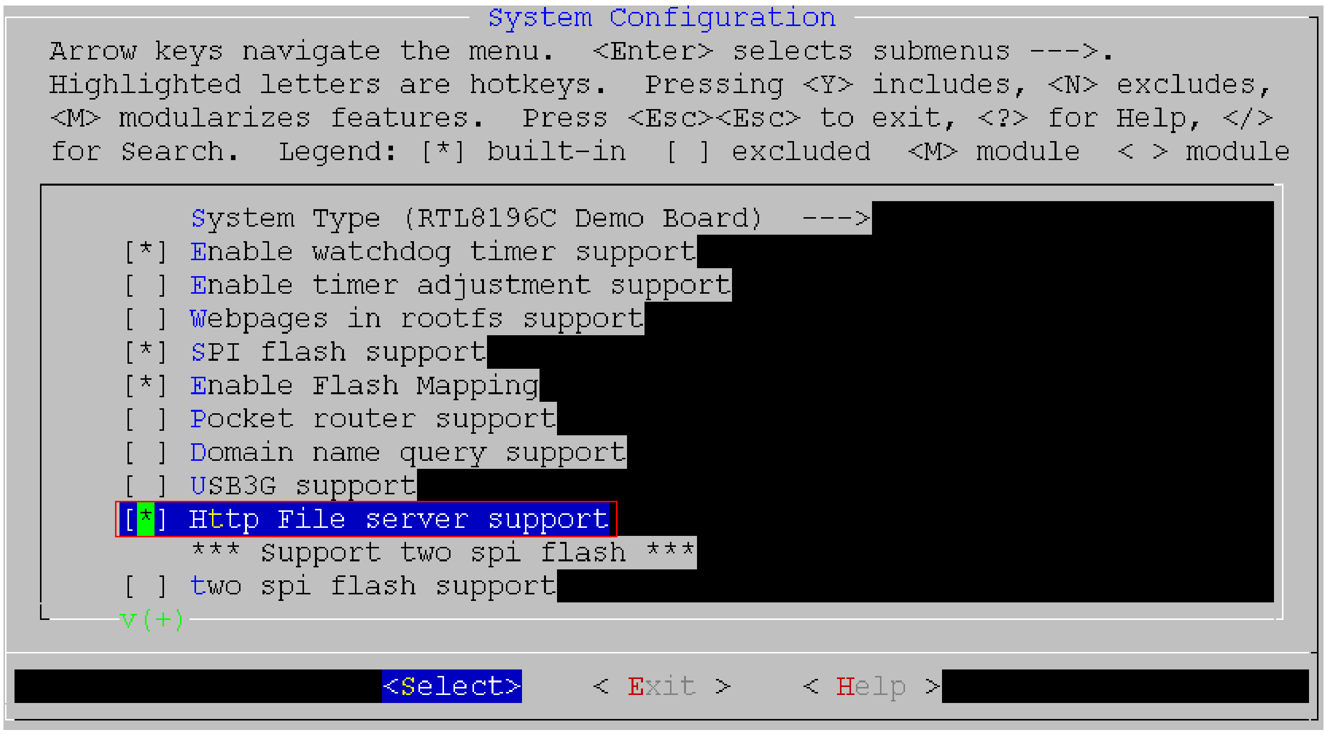
1) Kernel configure for HTTP File Server

make linux_menuconfig   // To configure kernel settings

Menuconfig:

System Configuration   --->

[*] Http File server support   // Selected



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

         System Type (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
     [*] Http File server support
         *** Support two spi flash ***
     [ ] two spi flash support
    v(+)

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

Device Drivers --->

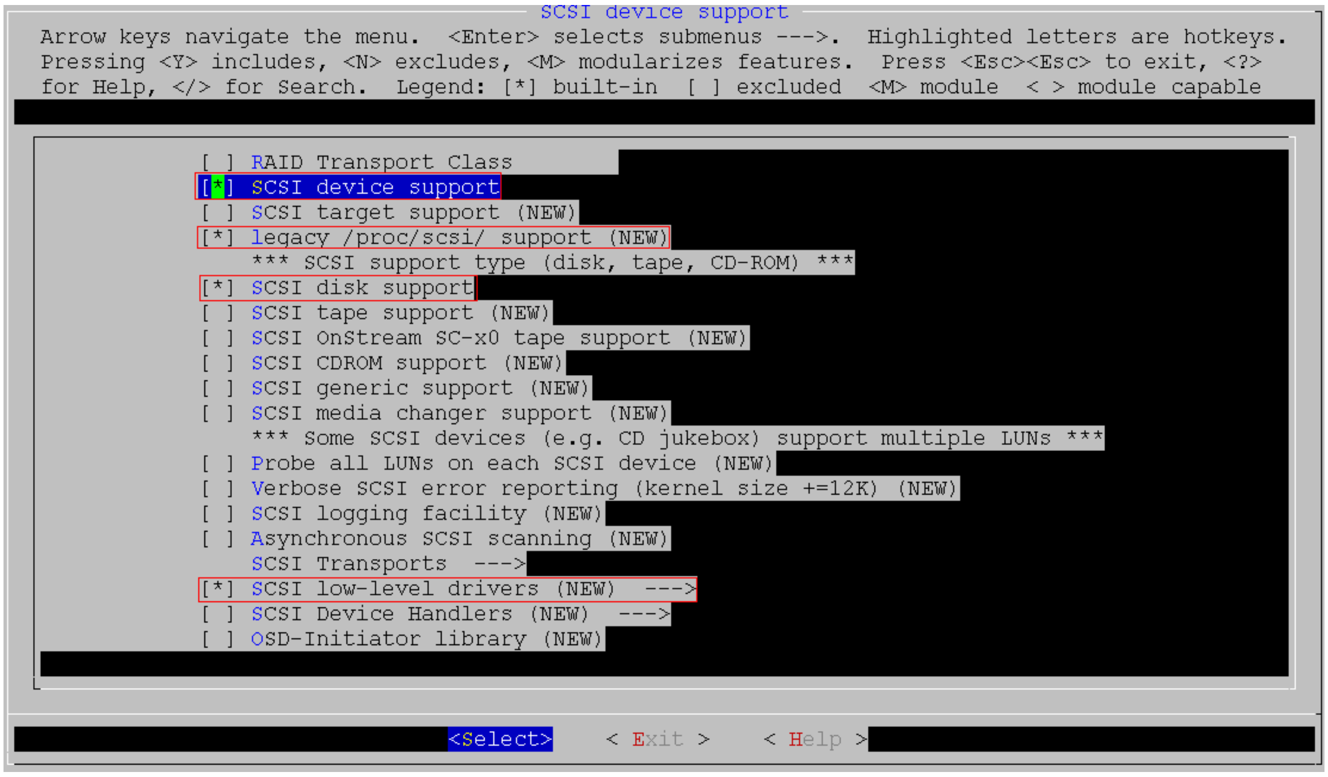
SCSI device support --->

[*] SCSI device support // Selected

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

[*] SCSI disk support // Selected

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

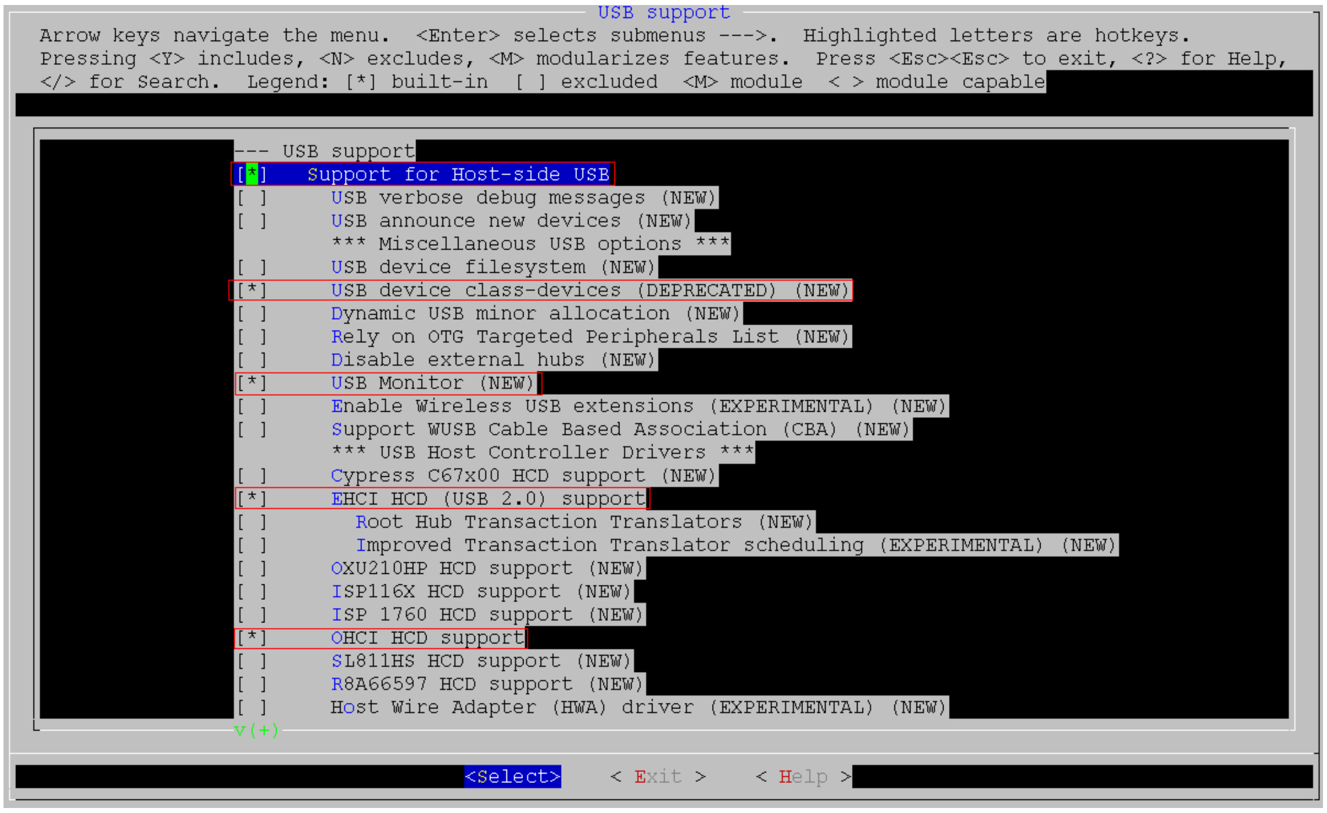


```
                              SCSI device support
  Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted letters are hotkeys.
  Pressing <Y> includes, <N> excludes, <M> modularizes features.  Press <Esc><Esc> to exit, <?>
  for Help, </> for Search.  Legend: [*] built-in  [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

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

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

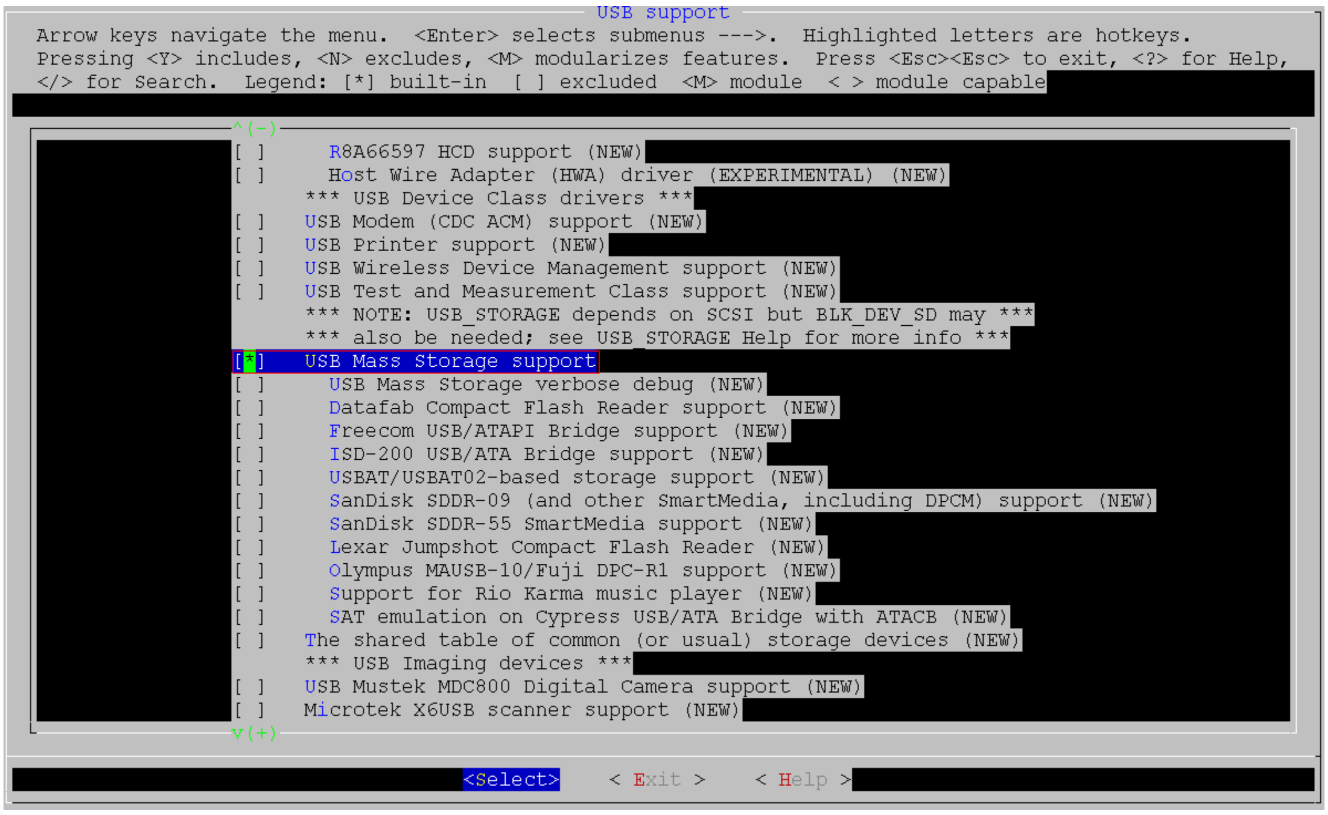
Device Drivers --->

[*] USB support --->

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



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

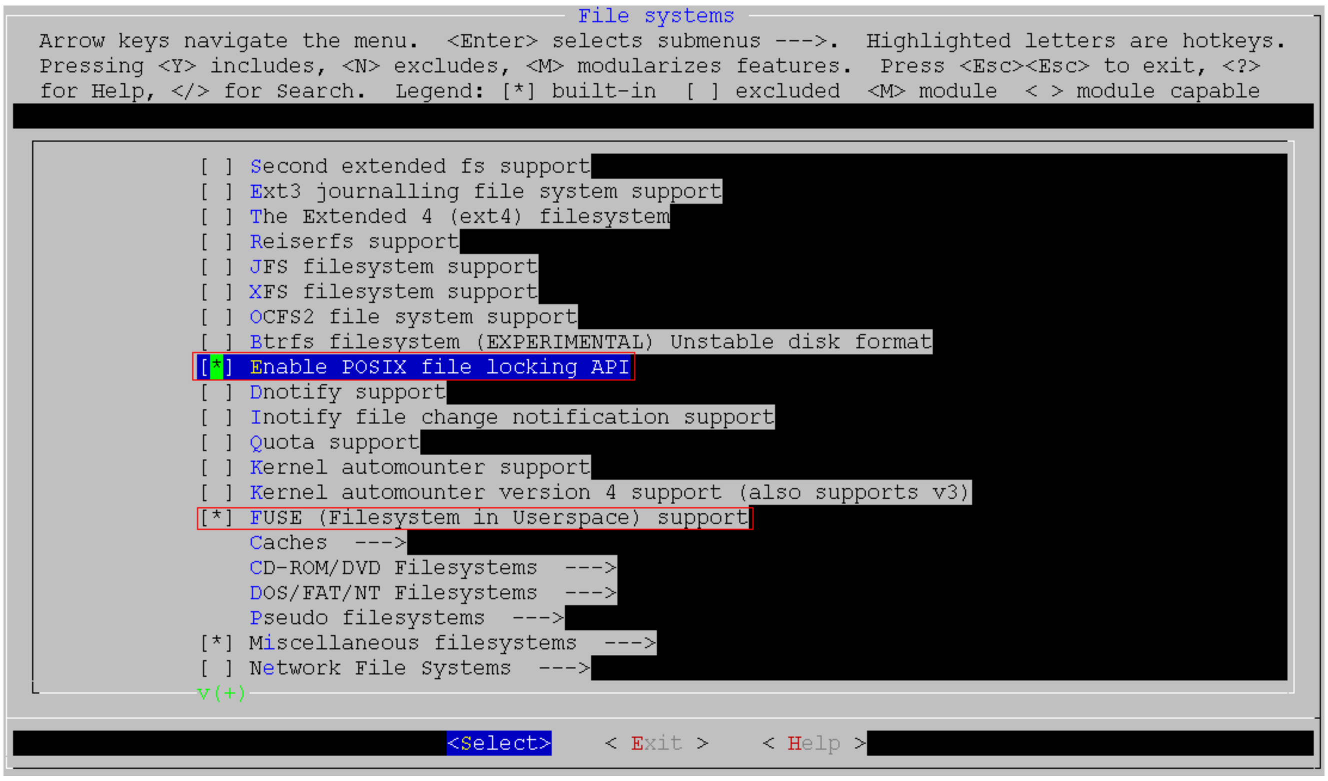


```
                              USB support
 Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted letters are hotkeys.
 Pressing <Y> includes, <N> excludes, <M> modularizes features.  Press <Esc><Esc> to exit, <?> for Help,
 </> for Search.  Legend: [*] built-in  [ ] excluded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

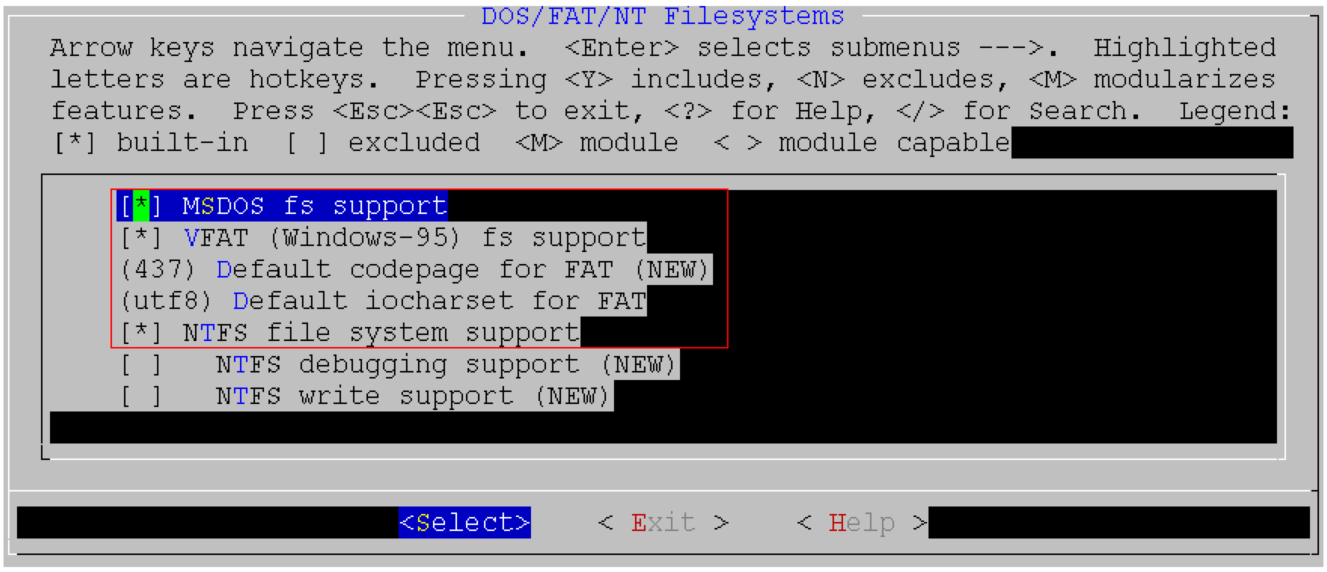
[*] MSDOS fs support // Selected

[*] 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

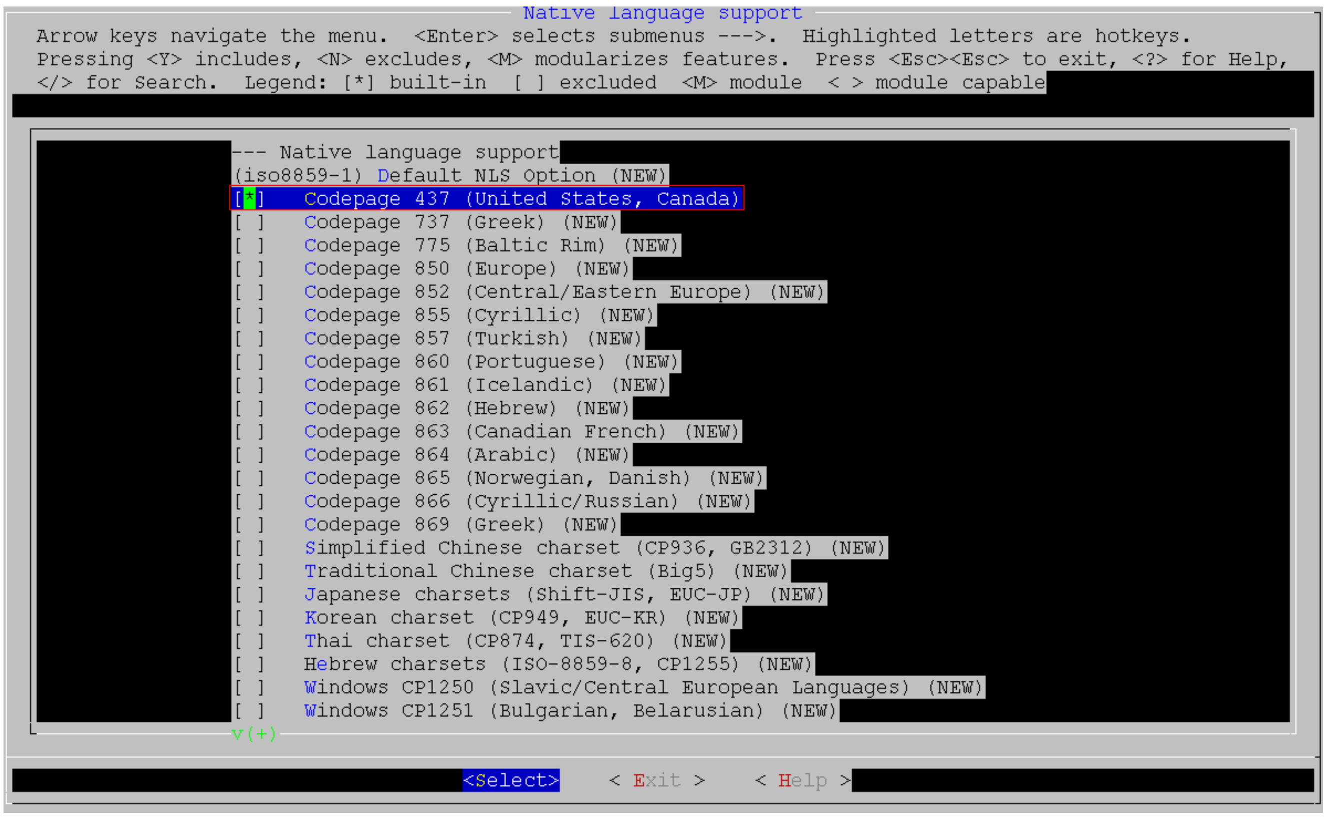


```
                      DOS/FAT/NT Filesystems
 Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted
 letters are hotkeys.  Pressing <Y> includes, <N> excludes, <M> modularizes
 features.  Press <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend:
 [*] built-in  [ ] excluded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

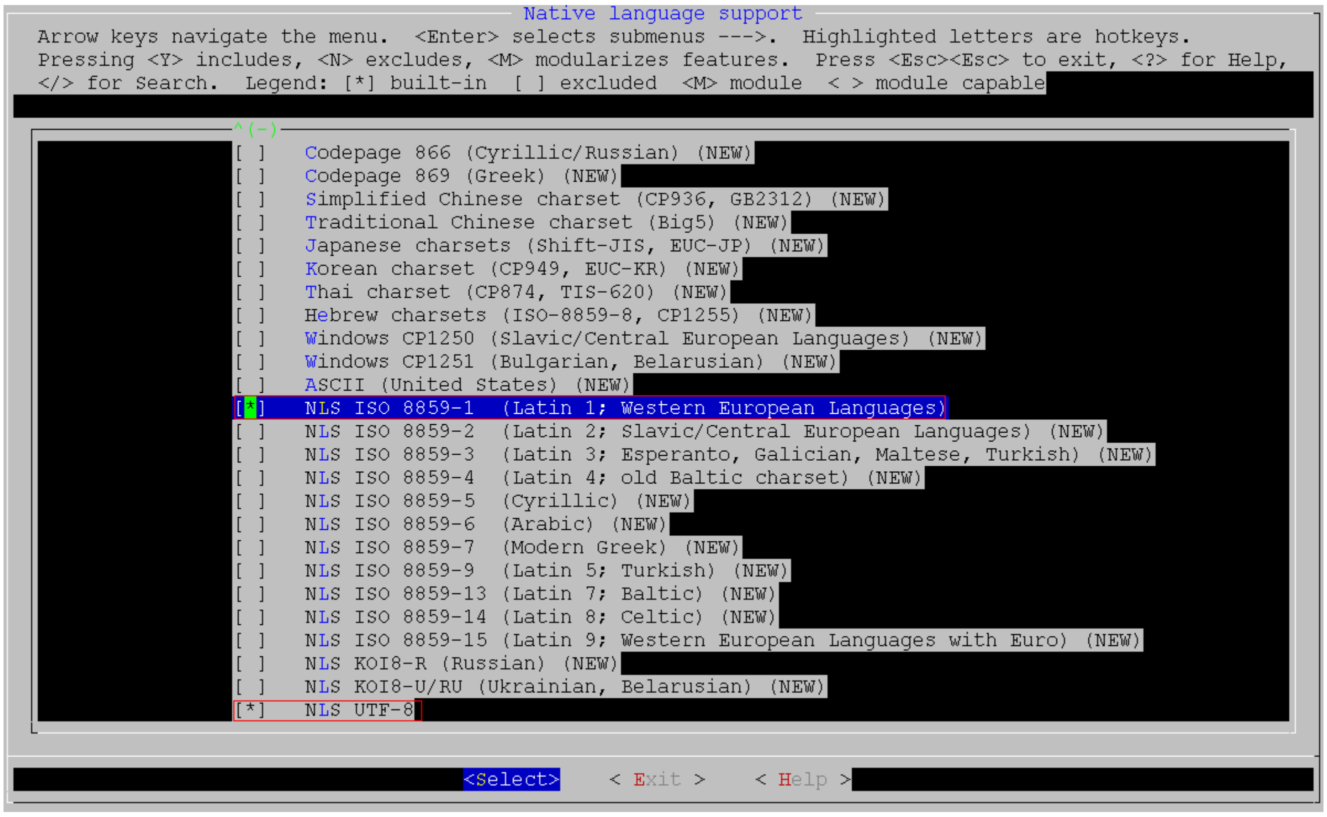
File systems --->

[*] Native language support --->

[*]	Codepage 437 (United States, Canada)	// Selected
[*]	NLS ISO 8859-1 (Latin 1; Western European Languages)	// Selected
[*]	NLS UTF-8	// Selected



```
                              Native language 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

                --- 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)
                v(+)

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



```
                              Native language 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

                 ^(-)
               [ ]   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)
               [ ]   NLS ISO 8859-4  (Latin 4; old Baltic charset)  (NEW)
               [ ]   NLS ISO 8859-5  (Cyrillic)  (NEW)
               [ ]   NLS ISO 8859-6  (Arabic)  (NEW)
               [ ]   NLS ISO 8859-7  (Modern Greek)  (NEW)
               [ ]   NLS ISO 8859-9  (Latin 5; Turkish)  (NEW)
               [ ]   NLS ISO 8859-13 (Latin 7; Baltic)  (NEW)
               [ ]   NLS ISO 8859-14 (Latin 8; Celtic)  (NEW)
               [ ]   NLS ISO 8859-15 (Latin 9; Western European Languages with Euro)  (NEW)
               [ ]   NLS KOI8-R (Russian)  (NEW)
               [ ]   NLS KOI8-U/RU (Ukrainian, Belarusian)  (NEW)
               [*]   NLS UTF-8

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

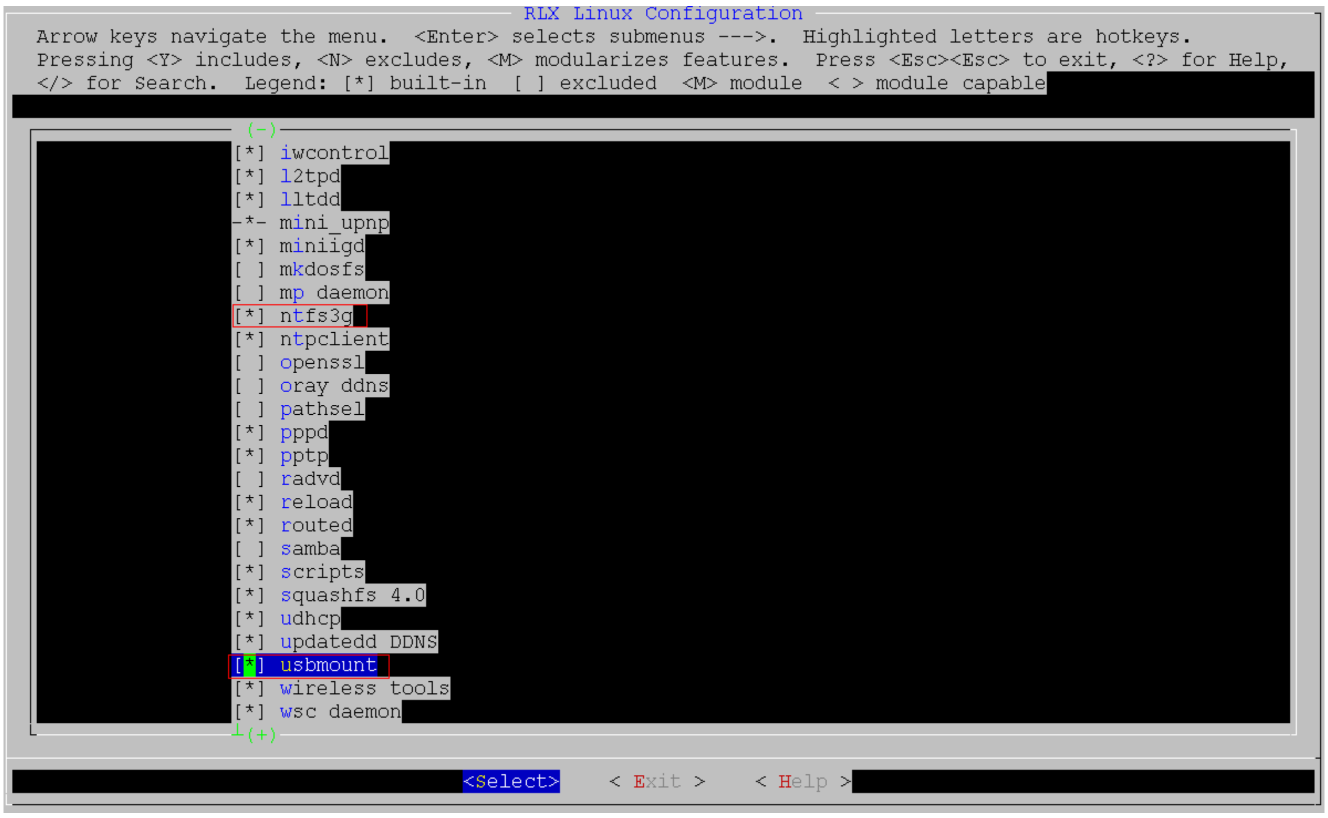
2) Application configure for HTTP File Server

make users_menuconfig // To configure application settings

Menuconfig:

[*] ntfs3g // selected

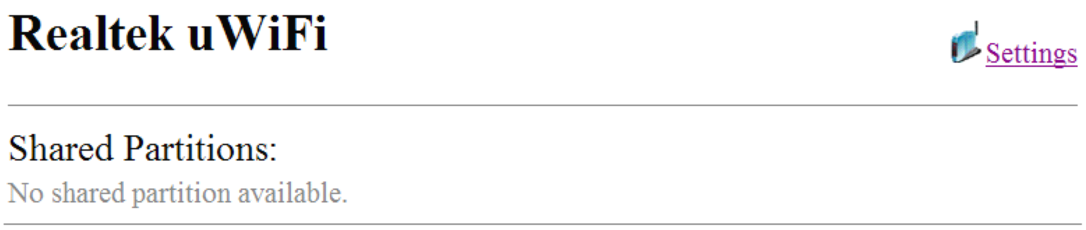
[*] usbmount // selected



```
                                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

                         (-)
                         [*] iwcontrol
                         [*] l2tpd
                         [*] lltdd
                         -*- mini_upnp
                         [*] miniigd
                         [ ] mkdosfs
                         [ ] mp daemon
                         [*] ntfs3g
                         [*] ntpclient
                         [ ] openssl
                         [ ] oray ddns
                         [ ] pathsel
                         [*] pppd
                         [*] pptp
                         [ ] radvd
                         [*] reload
                         [*] routed
                         [ ] samba
                         [*] scripts
                         [*] squashfs 4.0
                         [*] udhcp
                         [*] updatedd DDNS
                         [*] usbmount
                         [*] wireless tools
                         [*] wsc daemon
                         (+)

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

4.20.3 WEB GUI Manual

Home page:



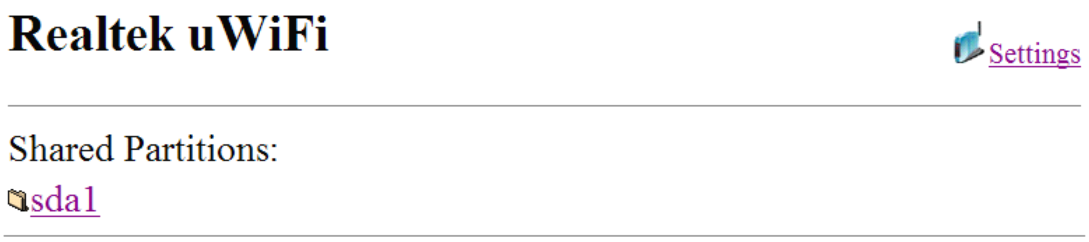
Realtek uWiFi

Settings

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

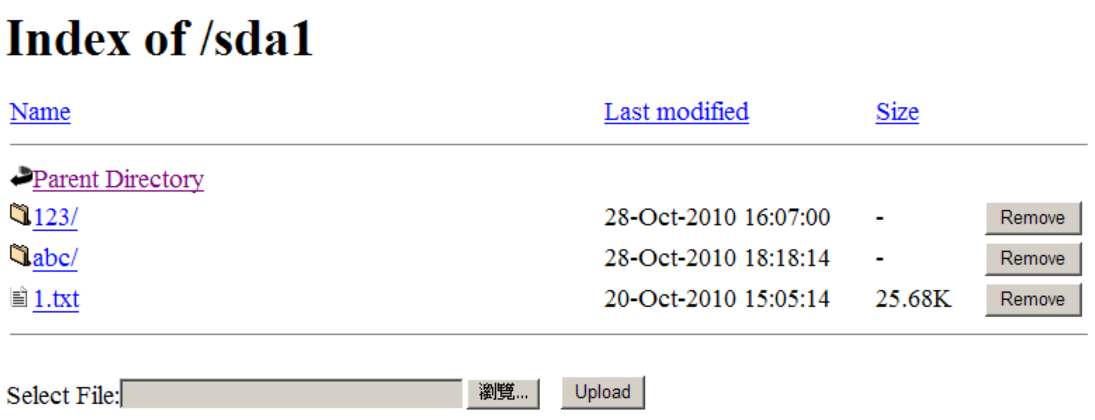
Settings

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:

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

ifconfig eth4 up

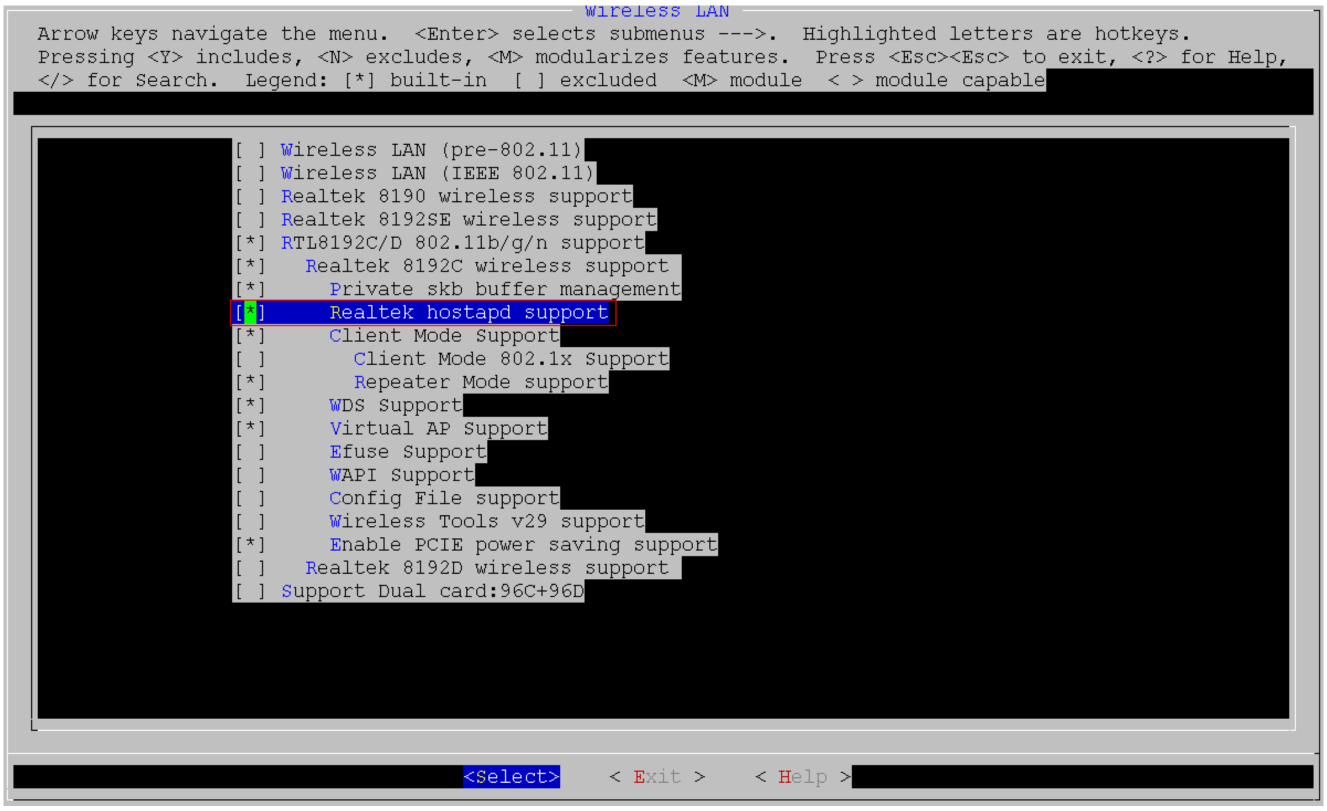
ifconfig wlan0 hw ether 00:44:55:66:98:81

cp /etc/hostapd. /tmp*

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

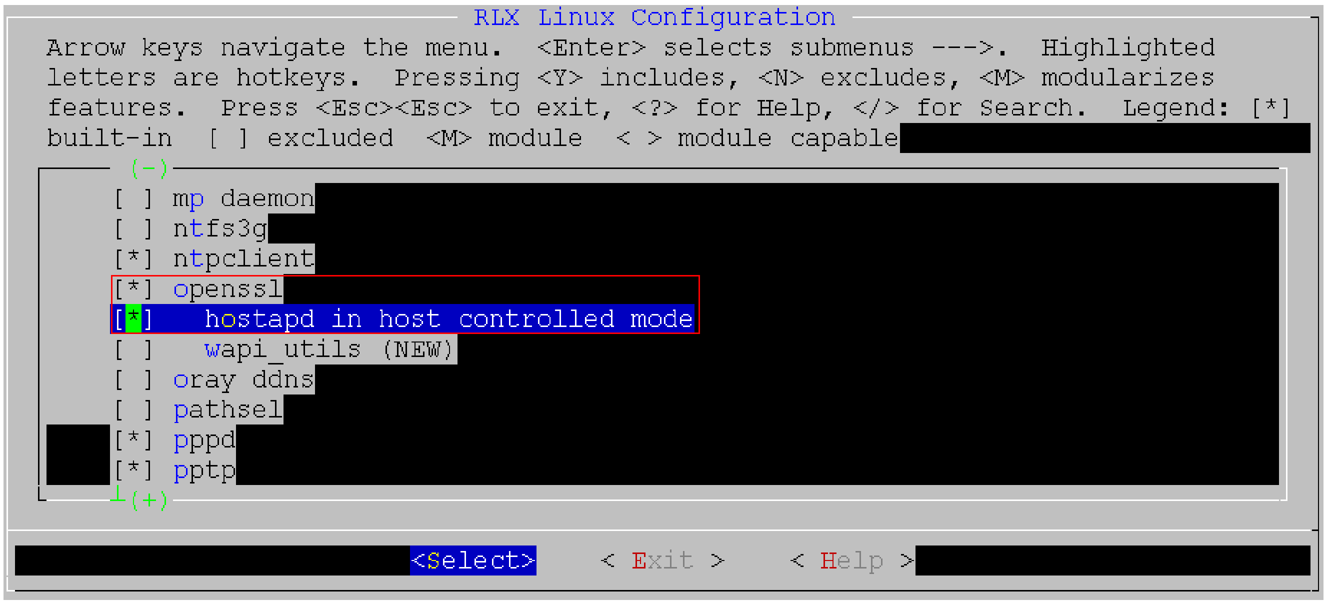
Step 2, configure kernel and application.

① Select "Realtek hostapd support" at kernel menuconfig.



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

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

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

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



```
                         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
        (-)
     [ ] mp daemon
     [ ] ntfs3g
     [*] ntpclient
     [*] openssl
     [*]   hostapd in host controlled mode
     [ ]   wapi_utils (NEW)
     [ ] oray ddns
     [ ] pathsel
     [*] pppd
     [*] pptp
        (+)

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

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

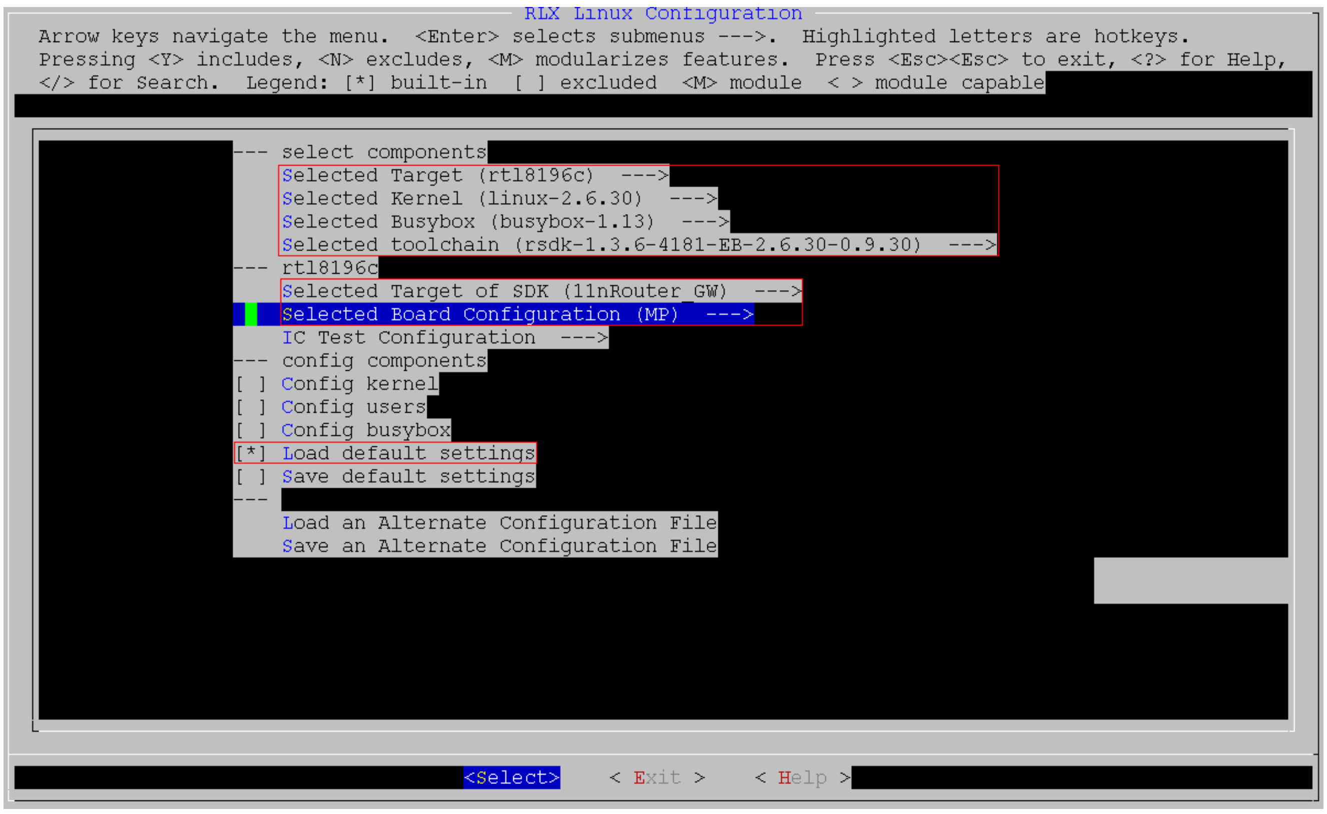
Step 4, customize hostapd.

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.

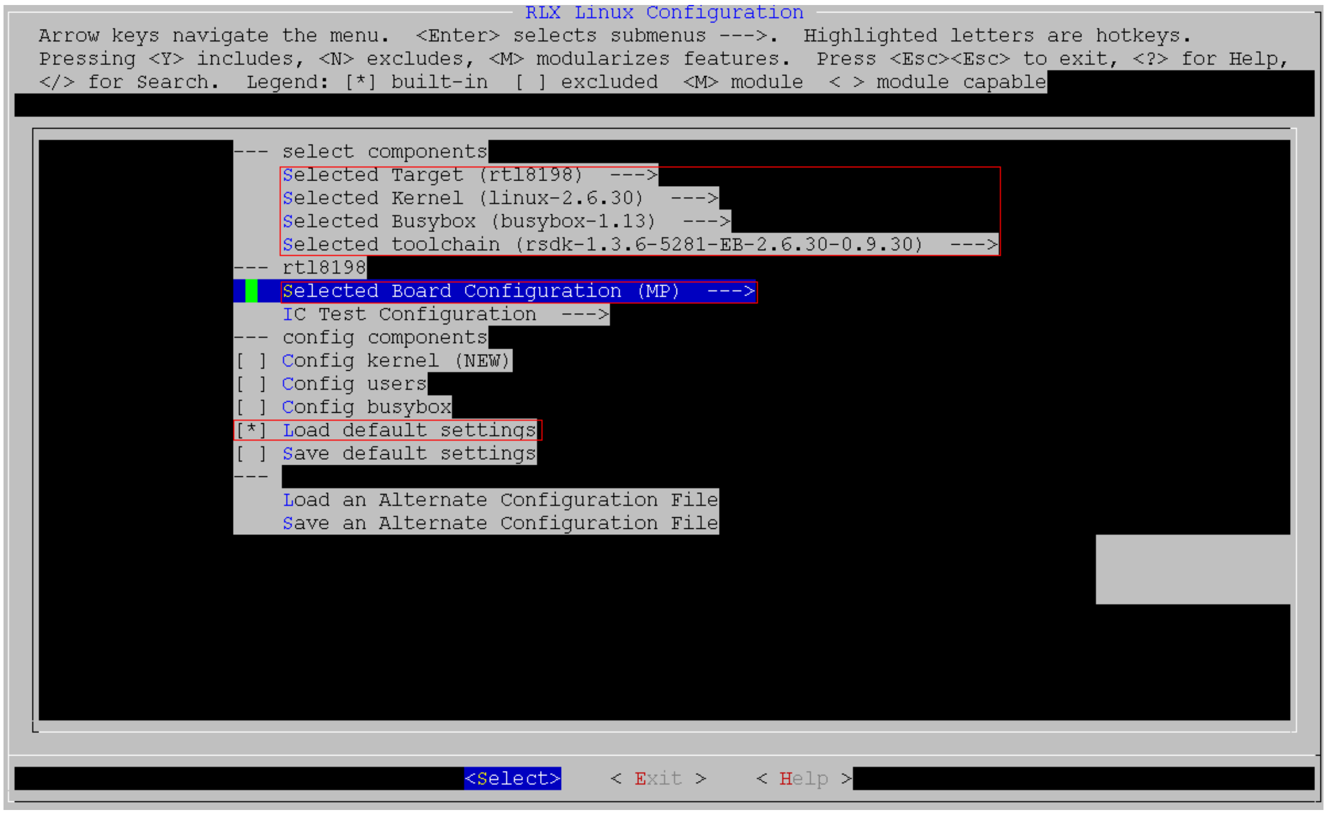


```
                              RLX Linux Configuration
 Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted letters are hotkeys.
 Pressing <Y> includes, <N> excludes, <M> modularizes features.  Press <Esc><Esc> to exit, <?> for Help,
 </> for Search.  Legend: [*] built-in  [ ] 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 (11nRouter_GW)  --->
                    Selected Board Configuration (MP)  --->
                    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 >
```

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.



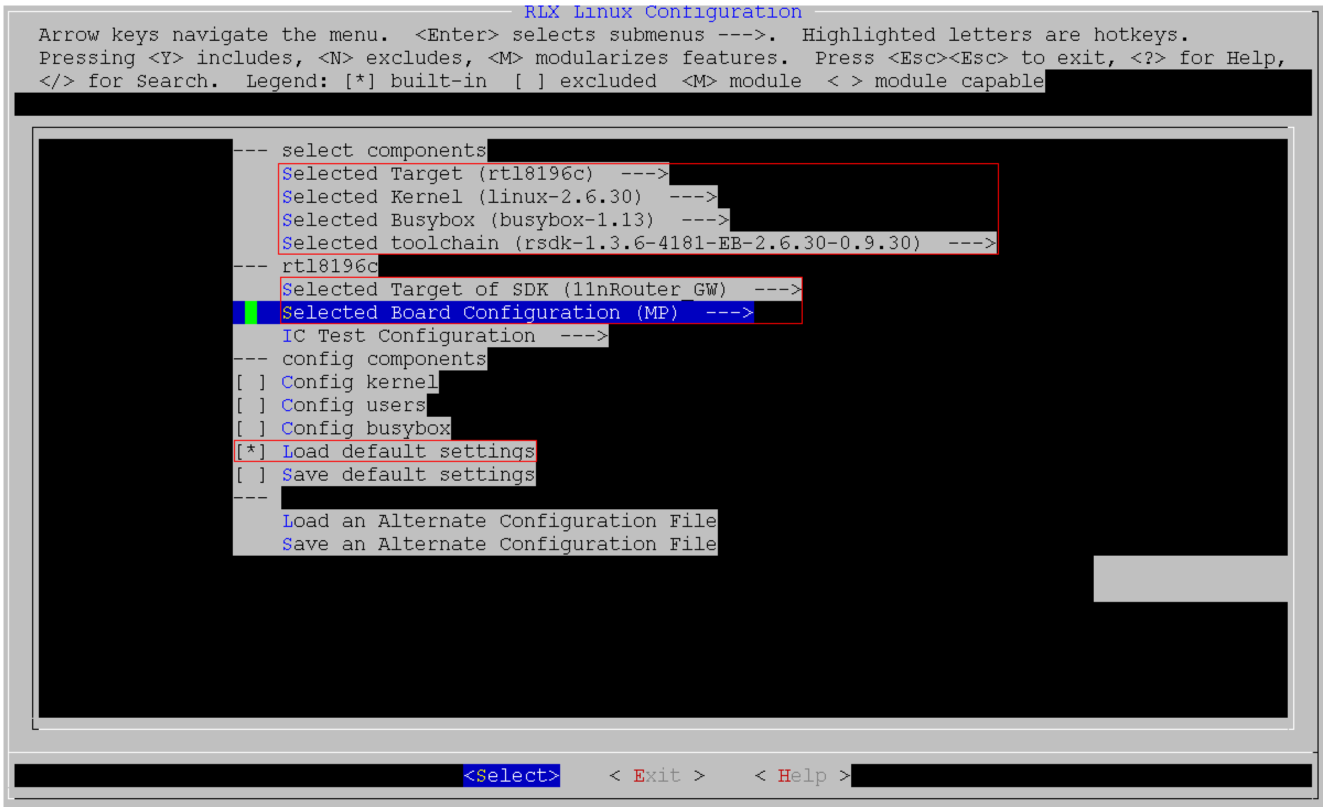
```
                              RLX Linux Configuration
  Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted letters are hotkeys.
  Pressing <Y> includes, <N> excludes, <M> modularizes features.  Press <Esc><Esc> to exit, <?> for Help,
  </> for Search.  Legend: [*] built-in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

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

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

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

4.22.3 POCKET AP MP support

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

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



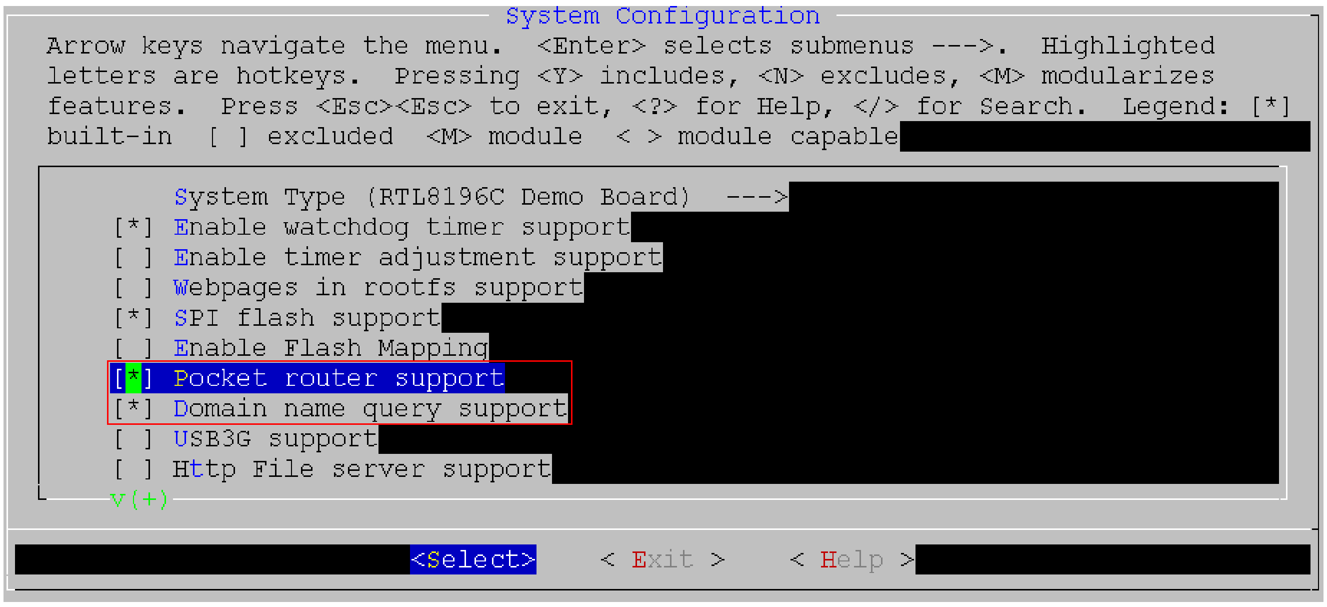
```
                              RLX Linux Configuration
  Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted letters are hotkeys.
  Pressing <Y> includes, <N> excludes, <M> modularizes features.  Press <Esc><Esc> to exit, <?> for Help,
  </> for Search.  Legend: [*] built-in  [ ] 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 (11nRouter_GW)  --->
                    Selected Board Configuration (MP)  --->
                    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 >
```

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

System Configuration --->

[*] Pocket router support // Selected

[*] Domain name query support // Selected

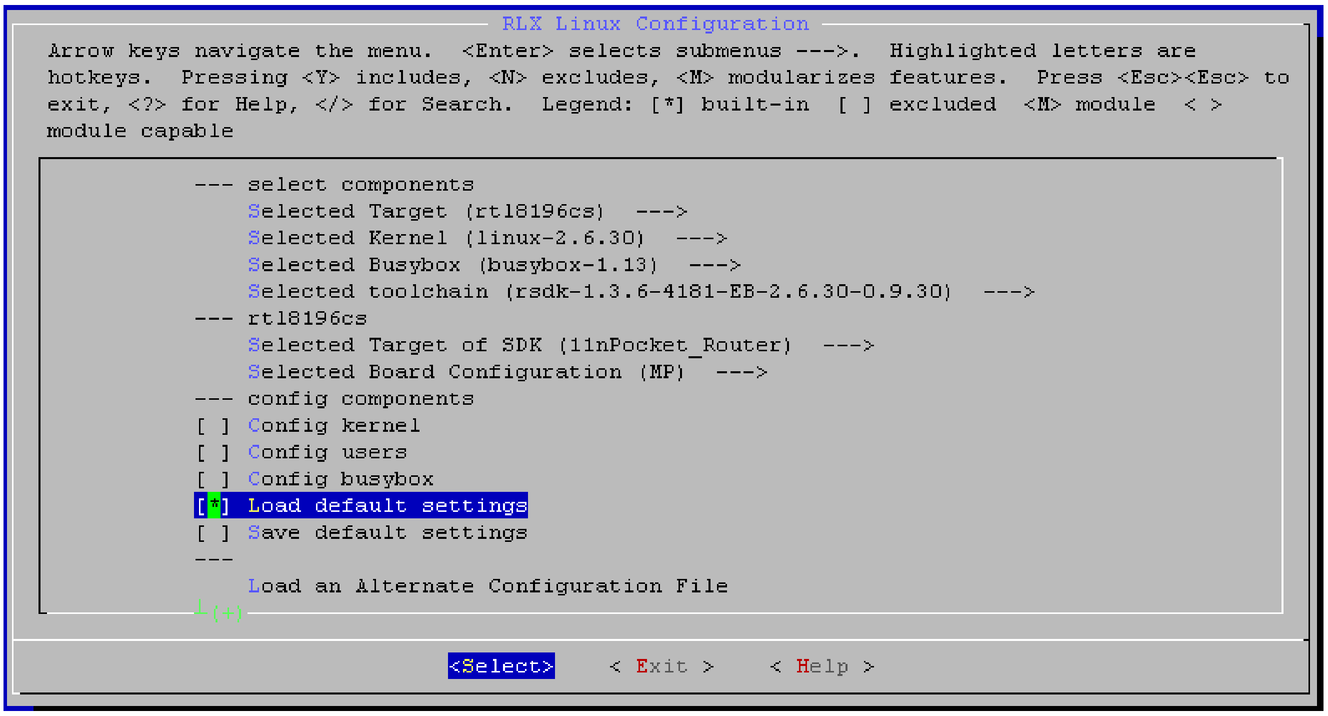


```
                         System Configuration
  Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted
  letters are hotkeys.  Pressing <Y> includes, <N> excludes, <M> modularizes
  features.  Press <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend: [*]
  built-in  [ ] 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
      [*] Domain name query support
      [ ] USB3G support
      [ ] Http File server support
      v(+)

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

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.



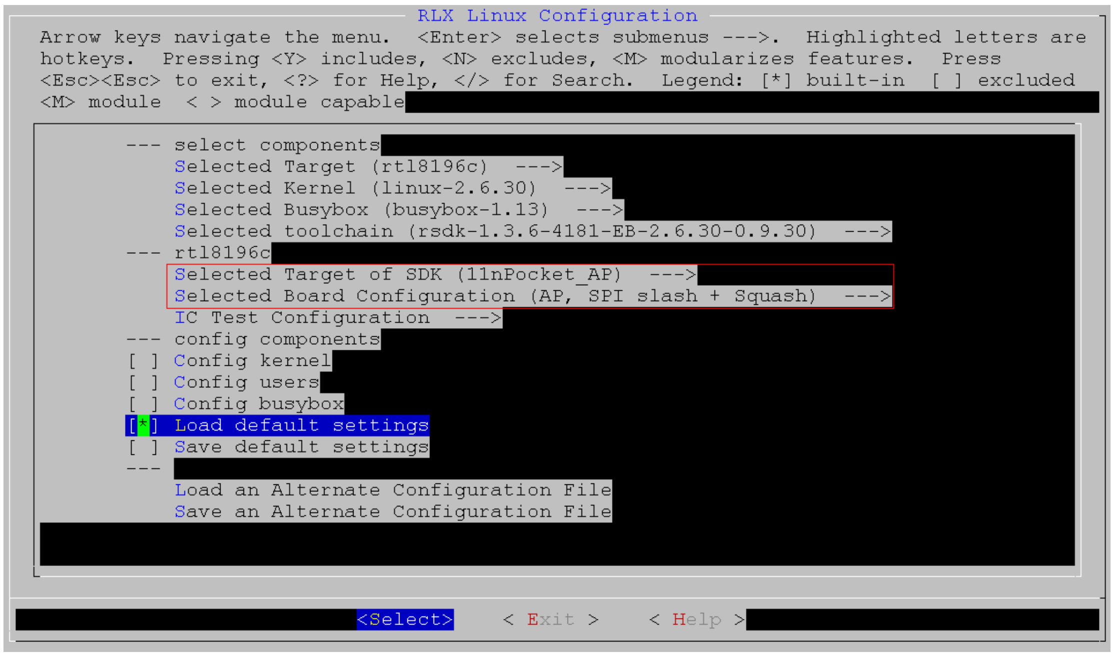
```
                         RLX Linux Configuration
 Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted letters are
 hotkeys.  Pressing <Y> includes, <N> excludes, <M> modularizes features.  Press <Esc><Esc> to
 exit, <?> for Help, </> for Search.  Legend: [*] built-in  [ ] excluded  <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 (11nPocket_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

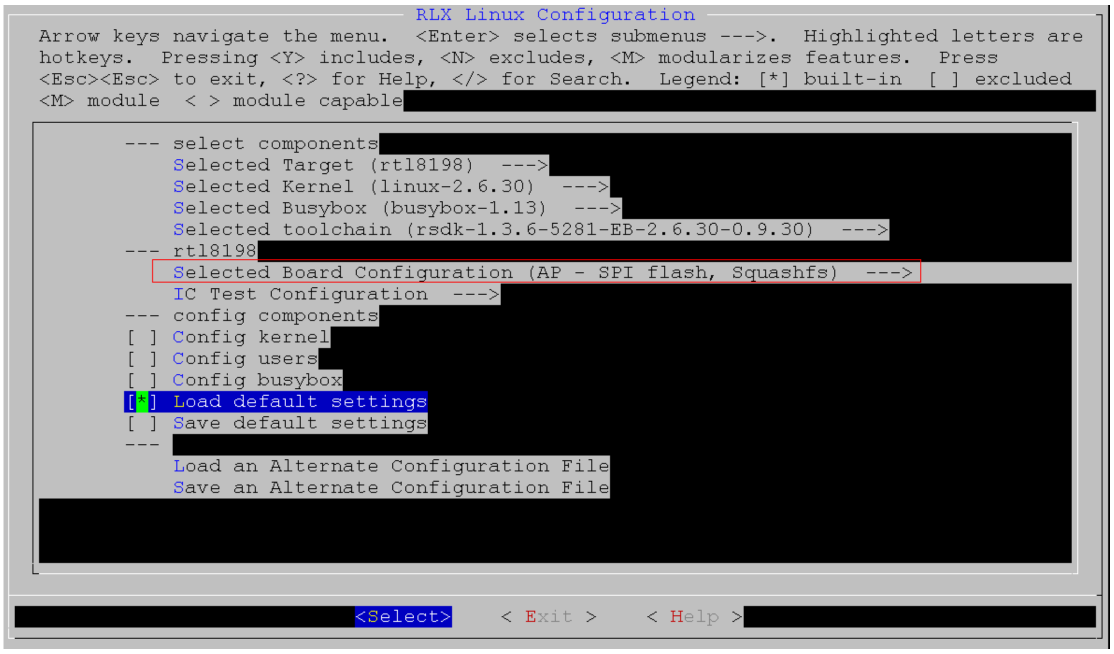
To configure AP mode for Pocket AP SDK, please use "*make menuconfig*" to configure SDK settings as follows.



```
                              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 (11nPocket_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.



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

        --- select components
            Selected Target (rtl8198)  --->
            Selected Kernel (linux-2.6.30)  --->
            Selected Busybox (busybox-1.13)  --->
            Selected toolchain (rsdk-1.3.6-5281-EB-2.6.30-0.9.30)  --->
        --- rtl8198
            Selected Board Configuration (AP - 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

                          <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.

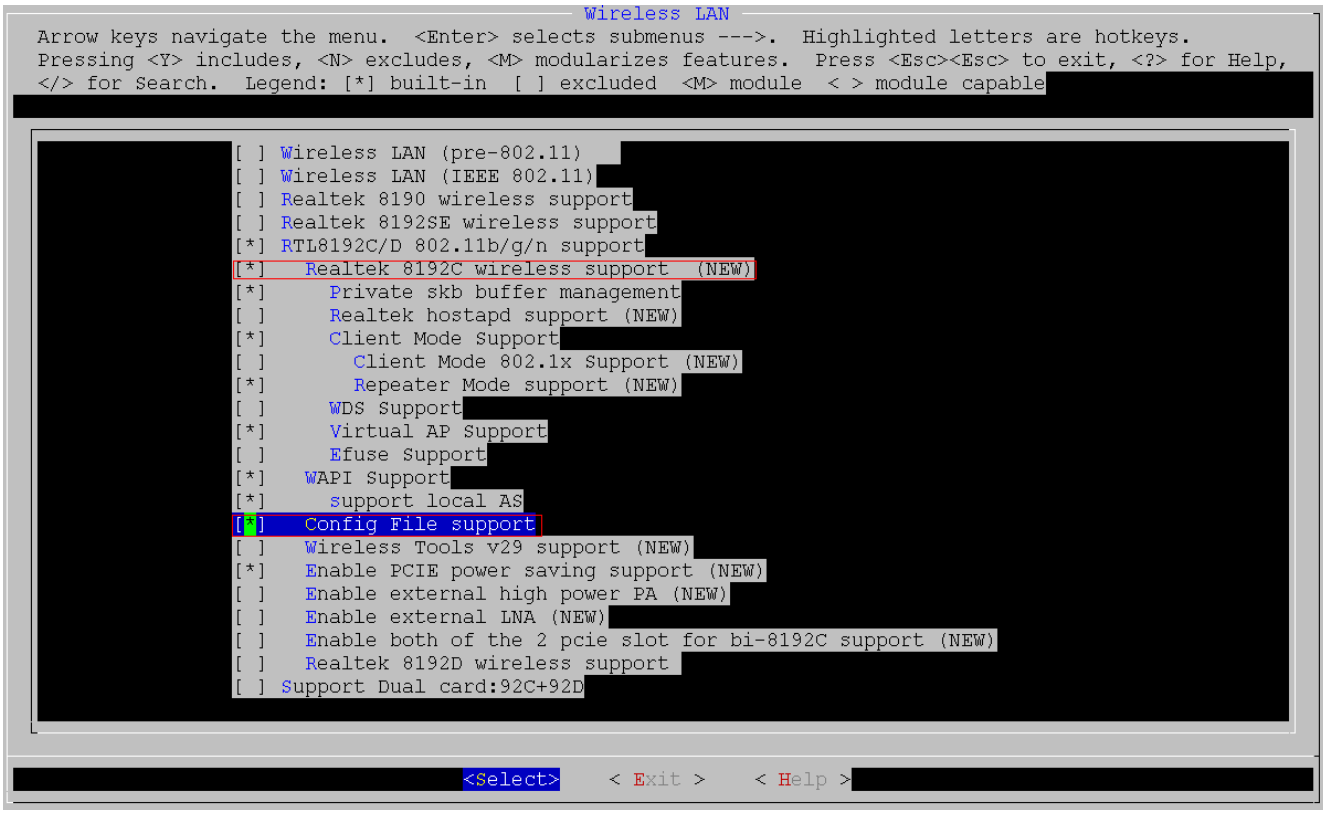
Device Drivers --->

[*] 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

Sytax: '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="#XXXXX@##$$%%"

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:

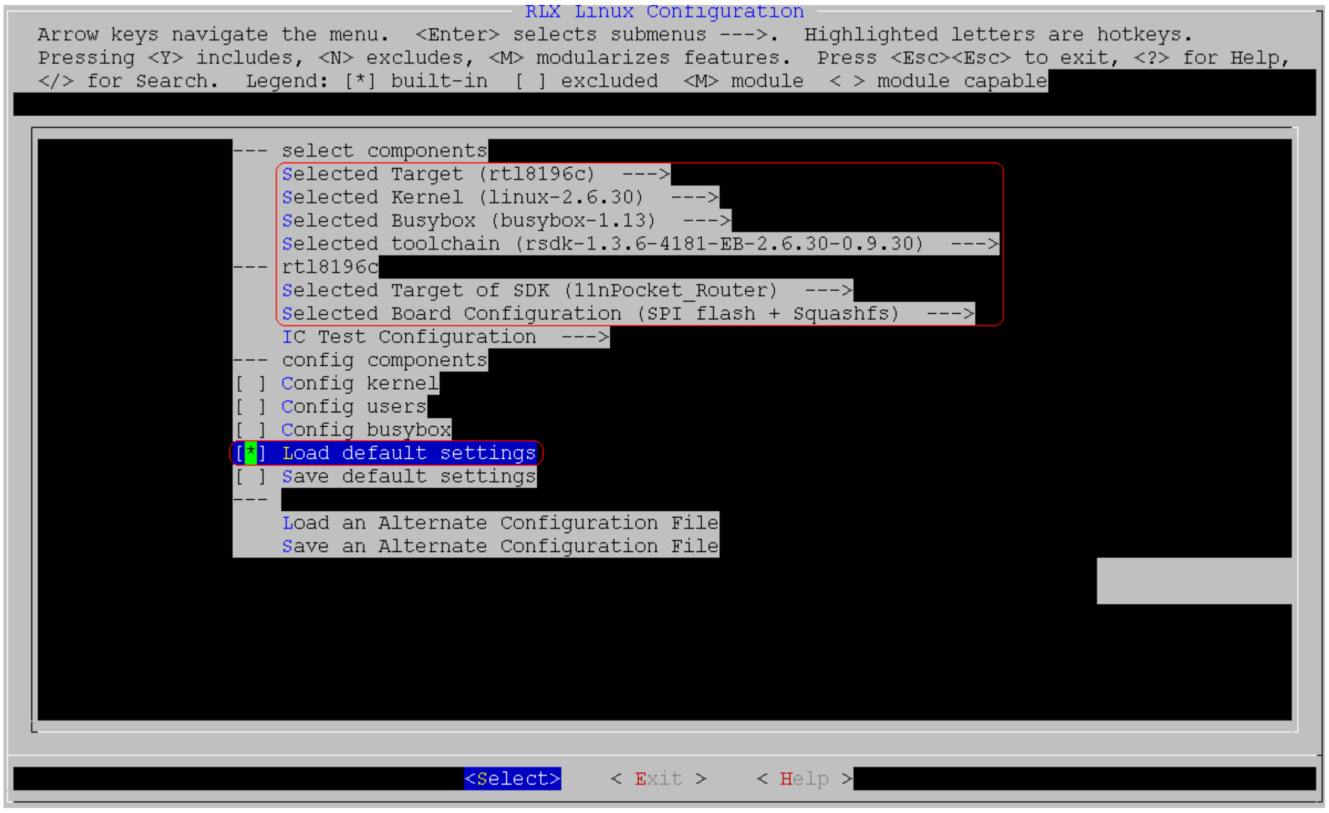
The iwcontrol tool is also needed for WPS. Please make sure the FIFO is created and opened in wscd, and iwcontrol use the same FIFO, otherwise wscd will hang.

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.



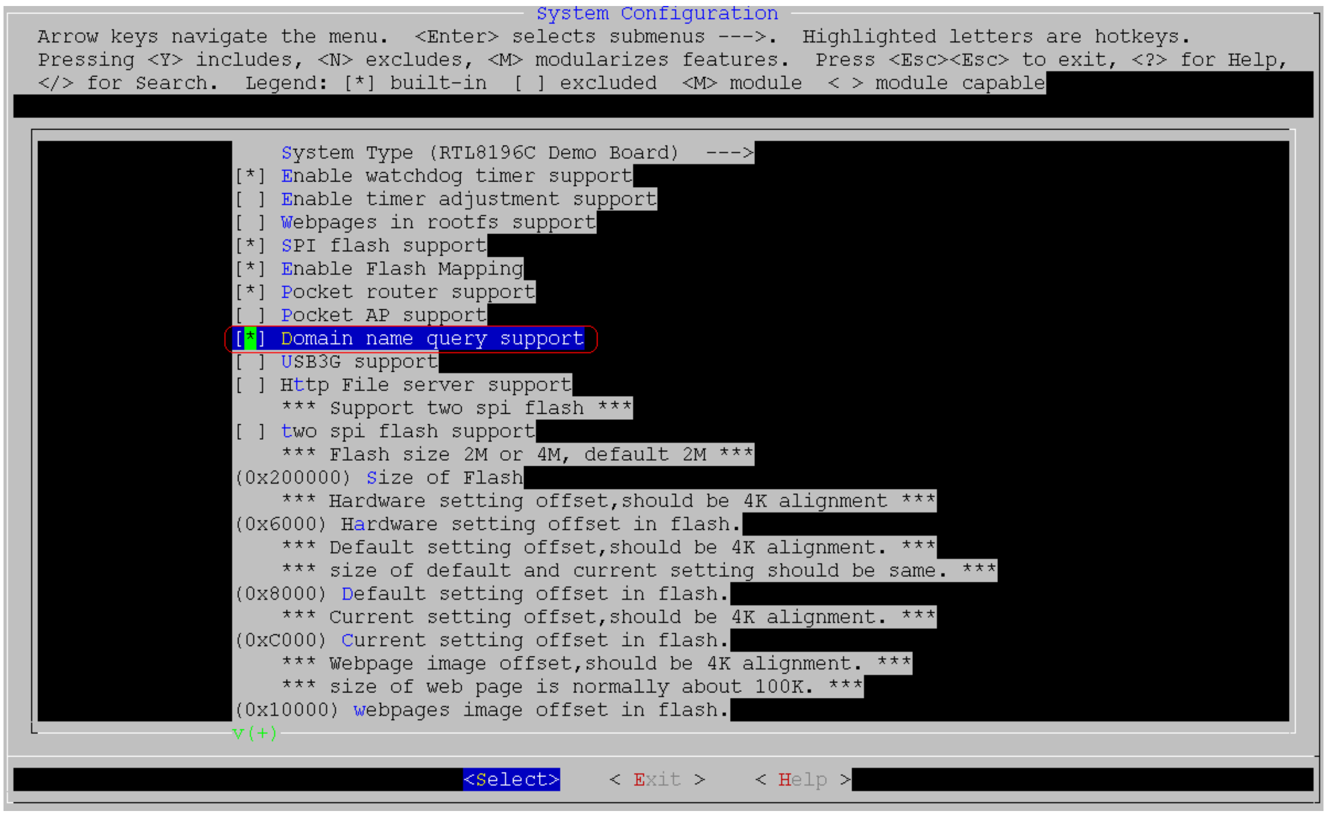
```
                              RLX Linux Configuration
 Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted letters are hotkeys.
 Pressing <Y> includes, <N> excludes, <M> modularizes features.  Press <Esc><Esc> to exit, <?> for Help,
 </> for Search.  Legend: [*] built-in  [ ] 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 (11nPocket_Router)  --->
                    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

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

Step 2, choose domain name query support.

make linux_menuconfig as follows:

System Configuration   --->

[*] Domain name query support



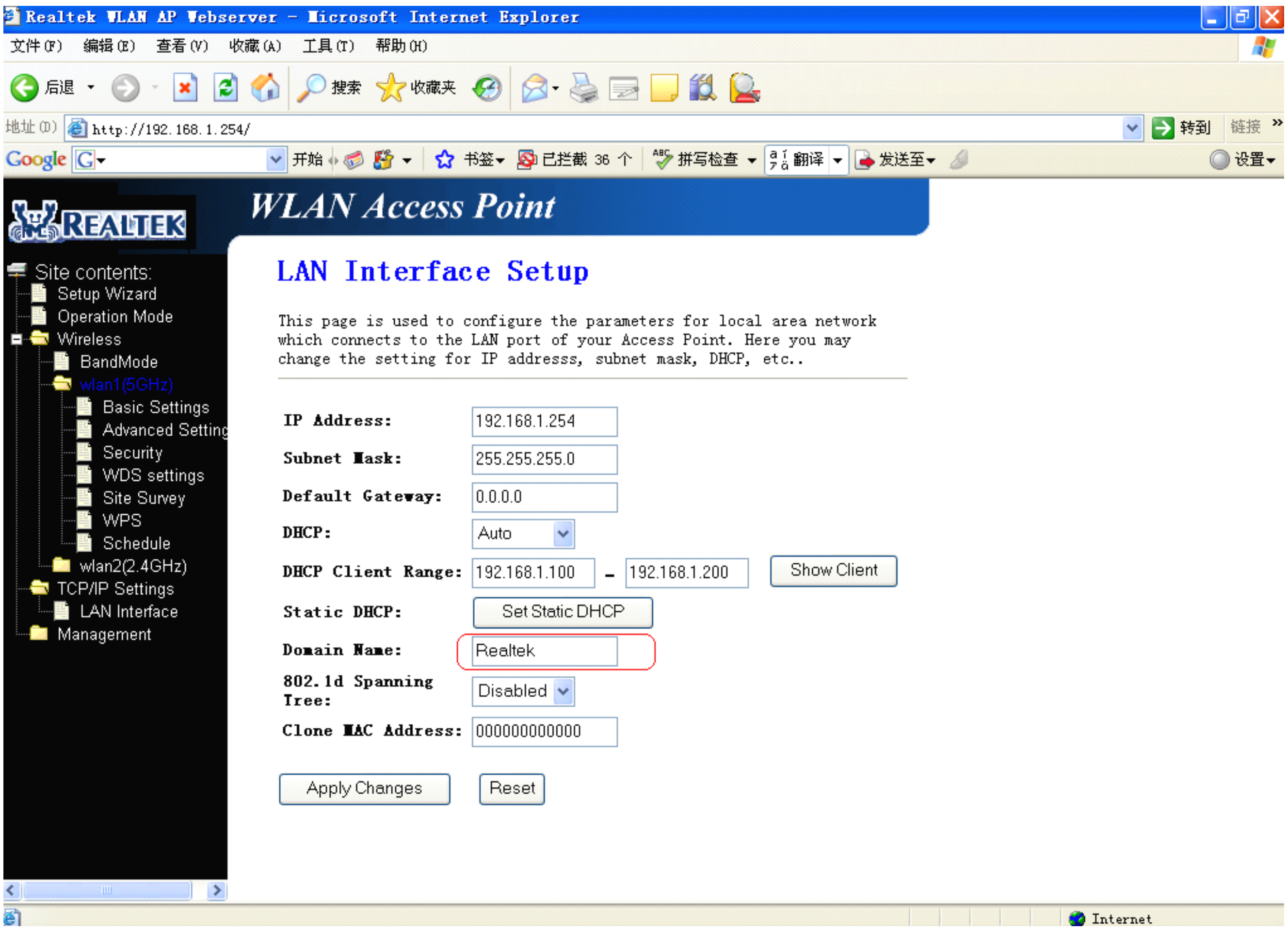
```
                                 System Configuration
  Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted letters are hotkeys.
  Pressing <Y> includes, <N> excludes, <M> modularizes features.  Press <Esc><Esc> to exit, <?> for Help,
  </> for Search.  Legend: [*] built-in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

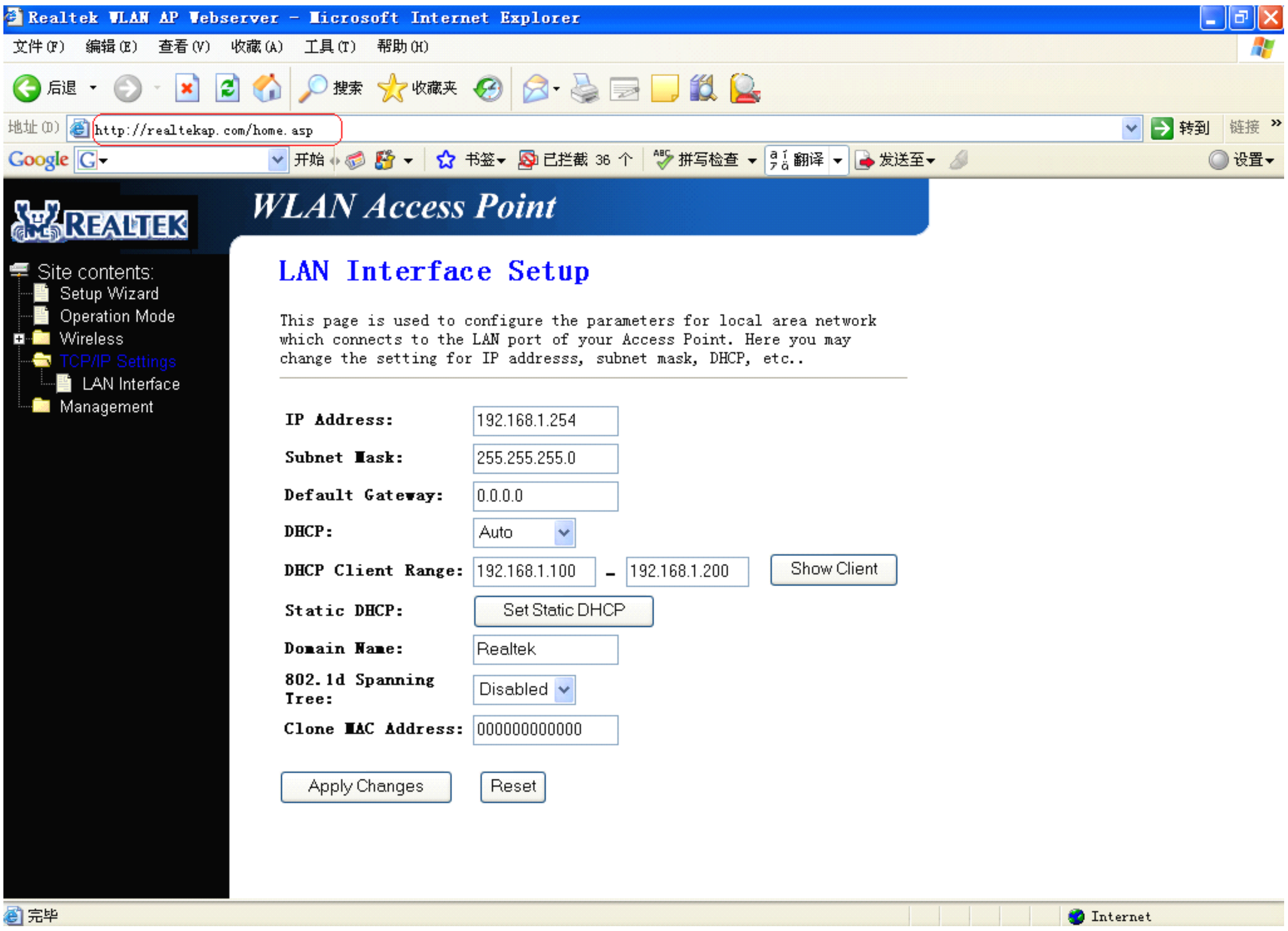
3) If AP works at client mode, AP webpage can be accessed via http://realtekcl.com/home.asp



LAN Interface Setup

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

IP Address:	192.168.1.254
Subnet Mask:	255.255.255.0
Default Gateway:	0.0.0.0
DHCP:	Auto
DHCP Client Range:	192.168.1.100 – 192.168.1.200
Static DHCP:	Set Static DHCP
Domain Name:	Realtek
802.1d Spanning Tree:	Disabled
Clone MAC Address:	000000000000

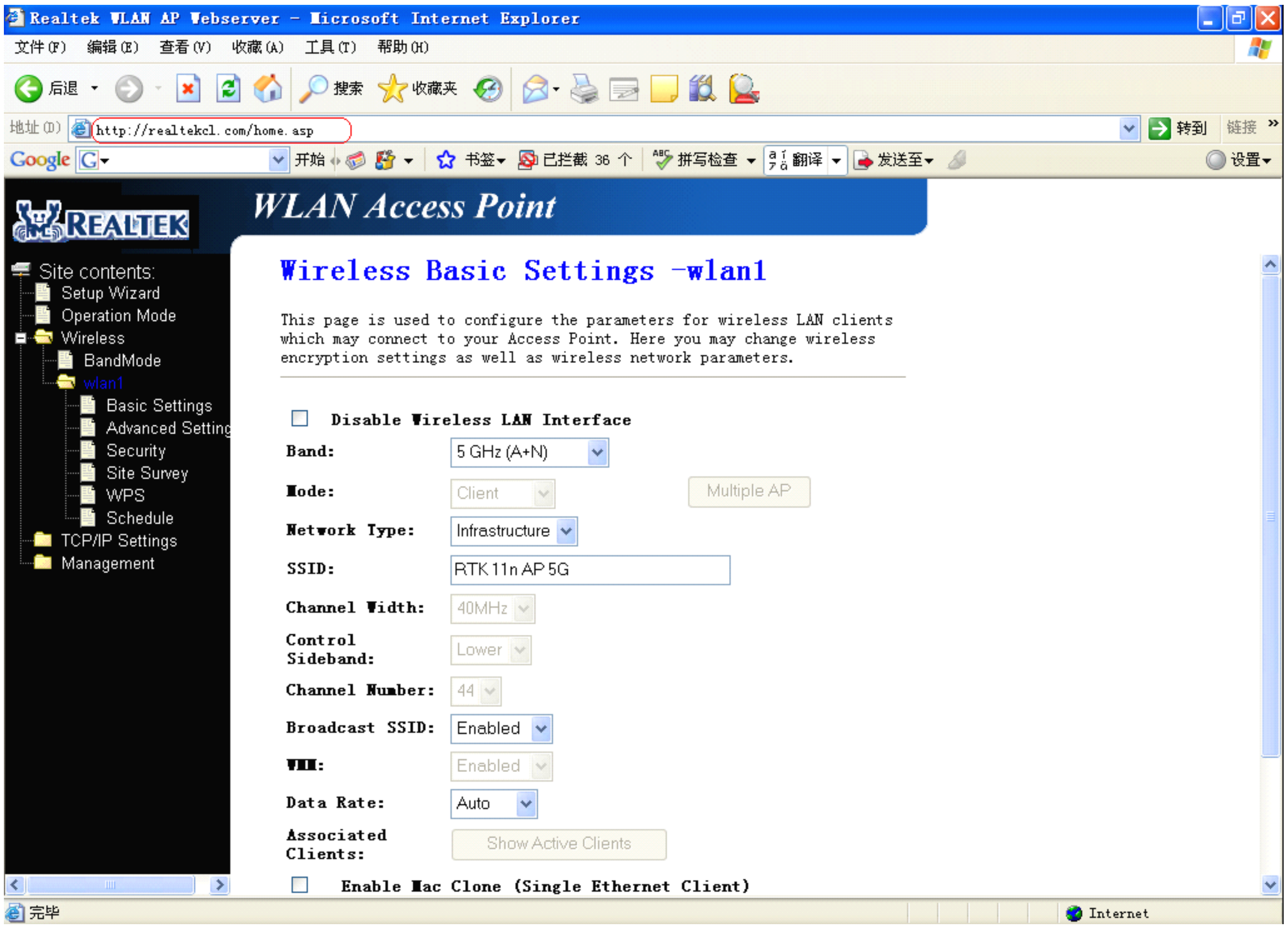


LAN Interface Setup

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 Show Client
Static DHCP:	Set Static DHCP
Domain Name:	Realtek
802.1d Spanning Tree:	Disabled
Clone MAC Address:	000000000000

Apply Changes Reset



Wireless Basic Settings -wlan1

This page is used to configure the parameters for wireless LAN clients which may connect to your Access Point. Here you may change wireless encryption settings as well as wireless network parameters.

☐ Disable Wireless LAN Interface

Band:	5 GHz (A+N)
Mode:	Client
Network Type:	Infrastructure
SSID:	RTK 11n AP 5G
Channel Width:	40MHz
Control Sideband:	Lower
Channel Number:	44
Broadcast SSID:	Enabled
WMM:	Enabled
Data Rate:	Auto
Associated Clients:	Show Active Clients

☐ Enable Mac Clone (Single Ethernet Client)

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,

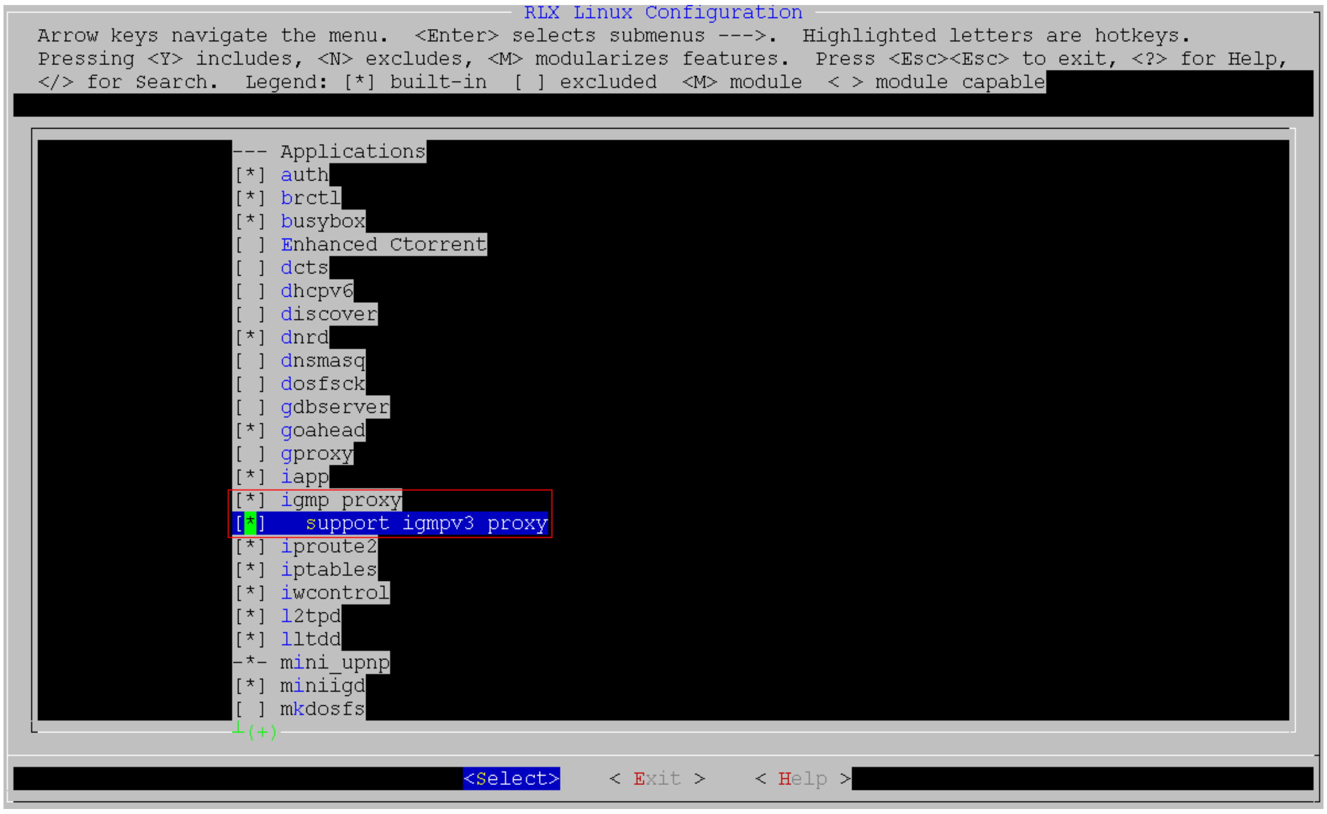
Support 128 hardware multicast entries.

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.

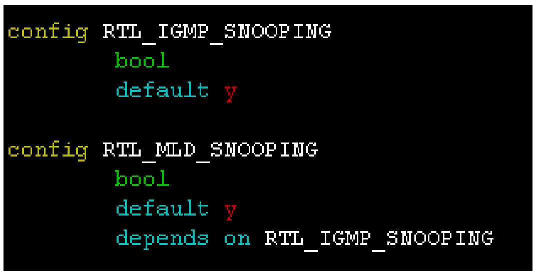


```
                              RLX Linux Configuration
  Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted letters are hotkeys.
  Pressing <Y> includes, <N> excludes, <M> modularizes features.  Press <Esc><Esc> to exit, <?> for Help,
  </> for Search.  Legend: [*] built-in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

                  --- Applications
                  [*] auth
                  [*] brctl
                  [*] busybox
                  [ ] Enhanced Ctorrent
                  [ ] dcts
                  [ ] dhcpv6
                  [ ] discover
                  [*] dnrd
                  [ ] dnsmasq
                  [ ] dosfsck
                  [ ] gdbserver
                  [*] goahead
                  [ ] gproxy
                  [*] iapp
                  [*] igmp proxy
                  [*]   support igmpv3 proxy
                  [*] iproute2
                  [*] iptables
                  [*] iwcontrol
                  [*] l2tpd
                  [*] lltdd
                  -*- mini_upnp
                  [*] miniigd
                  [ ] mkdosfs
                  ↓(+)

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

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 --->

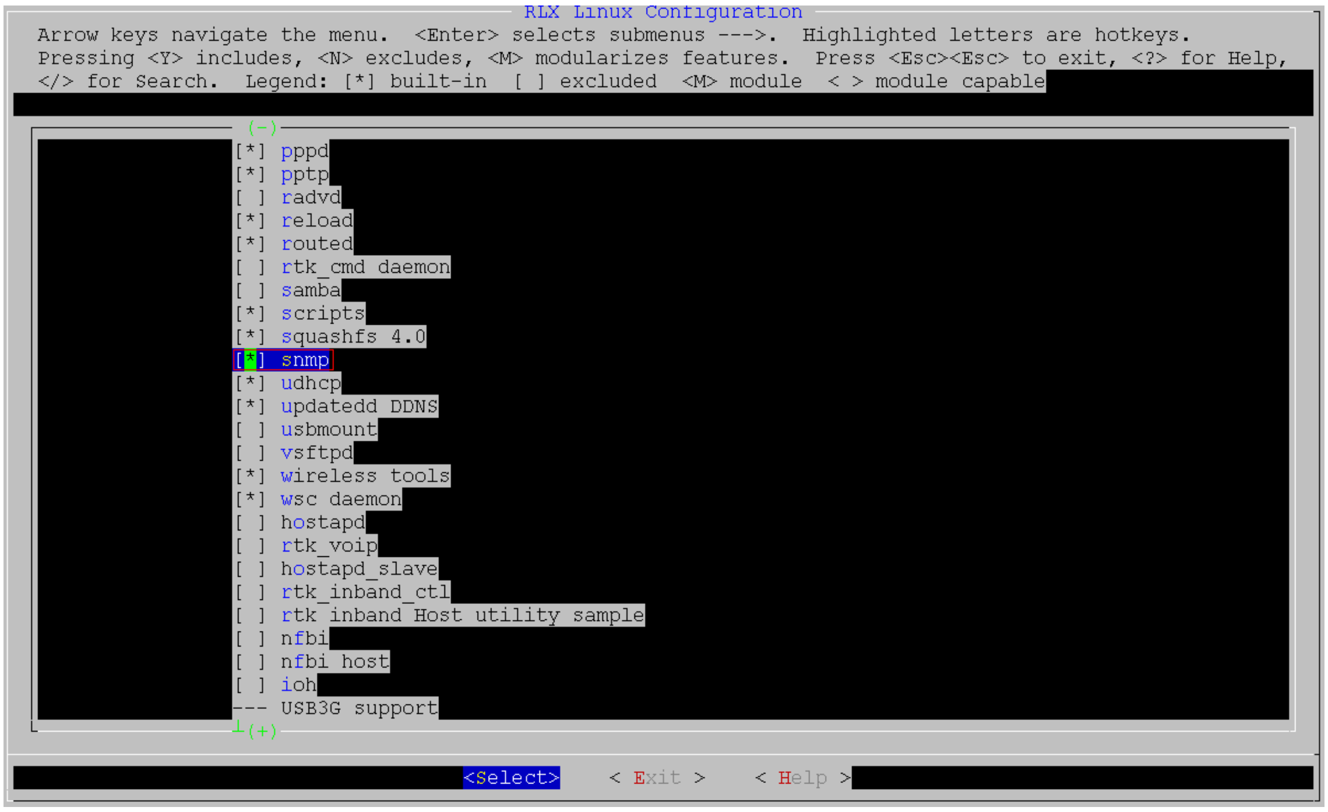
Hardware Features Selection (Enable RTL Hardware Multicast Only)

Or Hardware Features Selection (Enable RTL Hardware NAPT)

4.31 SNMP support

4.31.1 applications configure

make users_menuconfig //To configure applications



```
                              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

                        (-)
                        [*] pppd
                        [*] pptp
                        [ ] radvd
                        [*] reload
                        [*] routed
                        [ ] rtk_cmd daemon
                        [ ] samba
                        [*] scripts
                        [*] squashfs 4.0
                        [*] snmp
                        [*] udhcp
                        [*] updatedd DDNS
                        [ ] usbmount
                        [ ] vsftpd
                        [*] wireless tools
                        [*] wsc daemon
                        [ ] hostapd
                        [ ] rtk_voip
                        [ ] hostapd_slave
                        [ ] rtk_inband_ctl
                        [ ] rtk inband Host utility sample
                        [ ] nfbi
                        [ ] nfbi host
                        [ ] ioh
                        --- USB3G support
                        (+)

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

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 SNMP test

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

4.32 UVC support

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

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

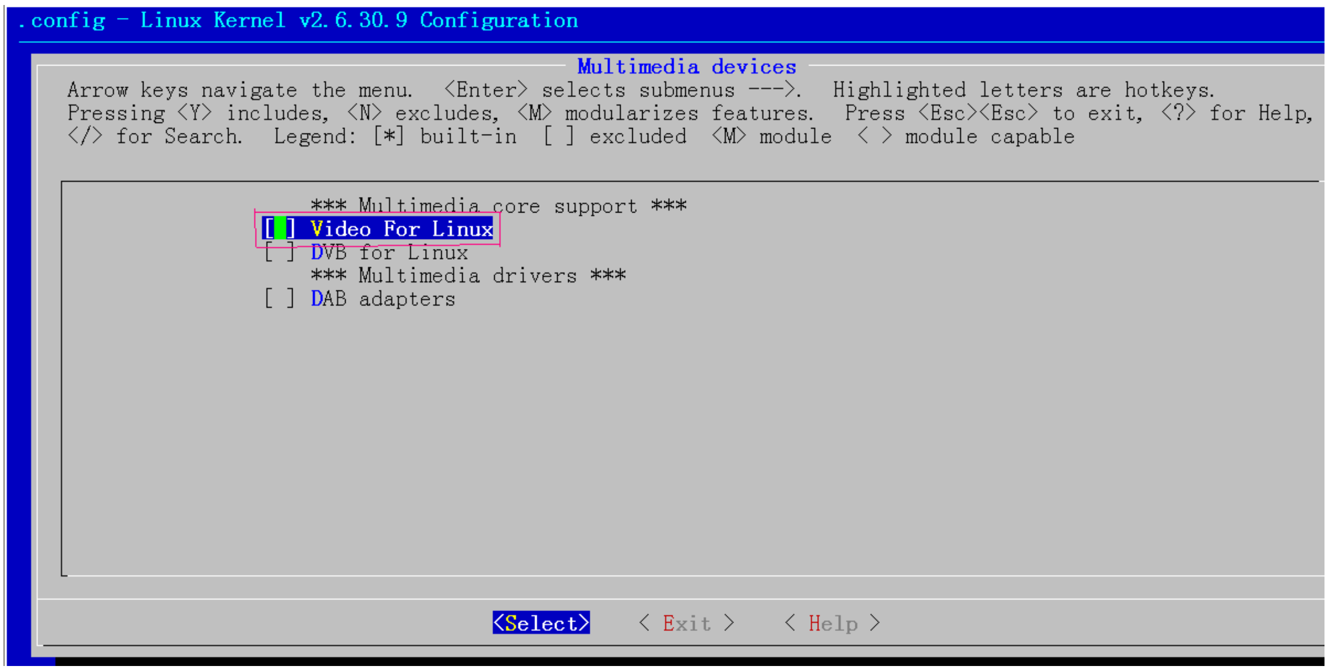
make linux_menuconfig   // To configure linux kernel settings

Menuconfig:

Device Drivers --->

Multimedia devices   --->

[ ] Video For Linux   // Not selected



```
.config - Linux Kernel v2.6.30.9 Configuration
                              Multimedia devices
  Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted letters are hotkeys.
  Pressing <Y> includes, <N> excludes, <M> modularizes features.  Press <Esc><Esc> to exit, <?> for Help,
  </> for Search.  Legend: [*] built-in  [ ] excluded  <M> module  < > module capable

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

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

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

make linux_menuconfig   // To configure linux kernel settings

Menuconfig:

Device Drivers --->

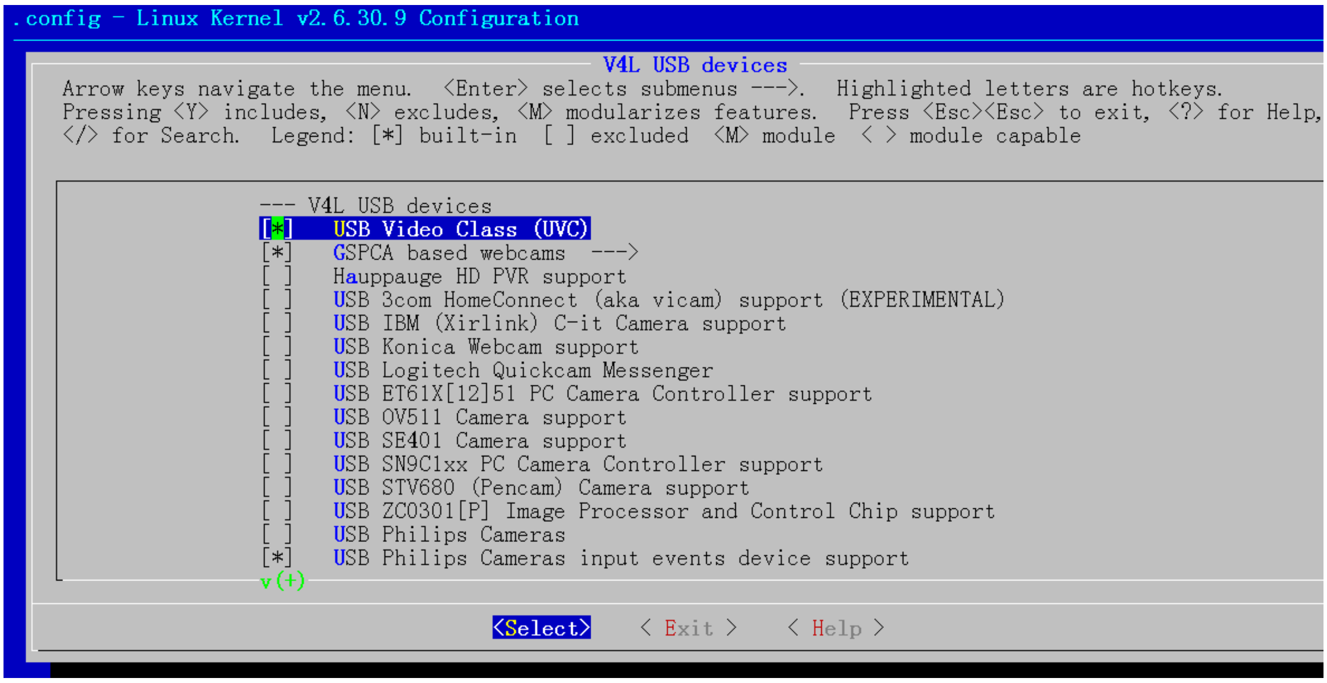
Multimedia devices   --->

[*] Video For Linux   // selected

[*] Video capture adapters   --->//selected and enter

[*]   V4L USB devices   --->//selected and enter

[*]   USB Video Class (UVC) //selected



```
.config - Linux Kernel v2.6.30.9 Configuration
                                   V4L USB devices
  Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted letters are hotkeys.
  Pressing <Y> includes, <N> excludes, <M> modularizes features.  Press <Esc><Esc> to exit, <?> for Help,
  </> for Search.  Legend: [*] built-in  [ ] excluded  <M> module  < > module capable

                 --- V4L USB devices
                 [*]   USB Video Class (UVC)
                 [*]   GSPCA based webcams  --->
                 [ ]   Hauppauge HD PVR support
                 [ ]   USB 3com HomeConnect (aka vicam) support (EXPERIMENTAL)
                 [ ]   USB IBM (Xirlink) C-it Camera support
                 [ ]   USB Konica Webcam support
                 [ ]   USB Logitech Quickcam Messenger
                 [ ]   USB ET61X[12]51 PC Camera Controller support
                 [ ]   USB OV511 Camera support
                 [ ]   USB SE401 Camera support
                 [ ]   USB SN9C1xx PC Camera Controller support
                 [ ]   USB STV680 (Pencam) Camera support
                 [ ]   USB ZC0301[P] Image Processor and Control Chip support
                 [ ]   USB Philips Cameras
                 [*]   USB Philips Cameras input events device support
                 v(+)

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

5. FLASH tools

FLASH tools such as flash, cvcfg-gw, compweb, cvimg, mgbin are stored at rtl819x-sdk-v2.x/rtl819x/users/goahead-2.1.1/LINUX.

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> <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 convertion tool.

Usage:

```
cvcfg-gw [-r] [-no_hw] <input> <ouput>
Options:
    -r: generate raw configutation with padding 0
    -no_hw: do not to generate hw setting
```

Examples:

```
cvcfg-gw config-gw-96c.txt config-gw-96c.dat
cvcfg-gw –r –no_hw config-gw-96c.txt config.bin
```

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.
           mgbin –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
mgbin –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 contol 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.

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.

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

7. FAQ

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

Modify linux-2.6.30/include/net/rtl/rtl_nic.h, define RTL_WANPORT_MASK and RTL_LANPORT_MASK as needed.

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);
```

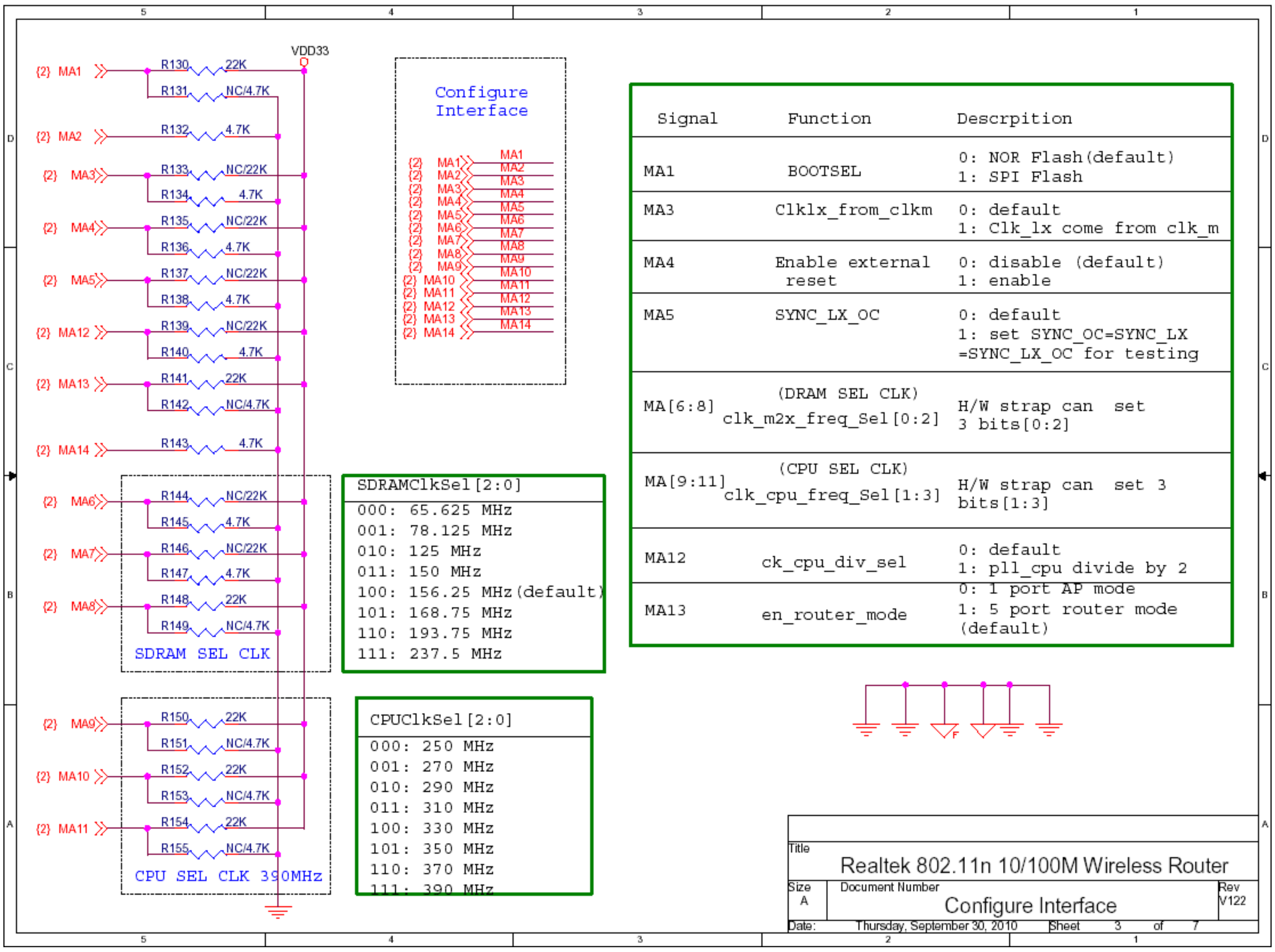
```
int32 rtl865x_delFilterDatabaseEntry(uint16 l2Type, uint16 fid, ether_addr_t * macAddr);
```

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.



Signal	Function	Descrpition
MA1	BOOTSEL	0: NOR Flash(default) 1: SPI Flash
MA3	Clklx_from_clkm	0: default 1: Clk_lx come from clk_m
MA4	Enable external reset	0: disable (default) 1: enable
MA5	SYNC_LX_OC	0: default 1: set SYNC_OC=SYNC_LX =SYNC_LX_OC for testing
MA[6:8]	(DRAM SEL CLK) clk_m2x_freq_Sel[0:2]	H/W strap can set 3 bits[0:2]
MA[9:11]	(CPU SEL CLK) clk_cpu_freq_Sel[1:3]	H/W strap can set 3 bits[1:3]
MA12	ck_cpu_div_sel	0: default 1: pll_cpu divide by 2
MA13	en_router_mode	0: 1 port AP mode 1: 5 port router mode (default)

```
SDRAMClkSel[2:0]
000: 65.625 MHz
001: 78.125 MHz
010: 125 MHz
011: 150 MHz
100: 156.25 MHz(default)
101: 168.75 MHz
110: 193.75 MHz
111: 237.5 MHz
```

```
CPUClkSel[2:0]
000: 250 MHz
001: 270 MHz
010: 290 MHz
011: 310 MHz
100: 330 MHz
101: 350 MHz
110: 370 MHz
111: 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" > /proc/phyPower   // 0: disable; 1: enable.

For example:

echo "0x00" > /proc/phyPower   // all Ethernet ports power off

```
echo "0x03" > /proc/phyPower     // Ethernet port0 and port1 power on
```

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_setAsicEthernetPHYAdvCapality(uint32 port, uint32 capality);
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_setAsicEthernetPHYAdvCapality(port,advCapability);
rtl8651_restartAsicEthernetPHYNway(port);
```

7.13 How to dial up pppoe for test when wan interface is changed?

Answer:

One example, interface usb_3g is used as wan interface to dial up pppoe.

For test, pppoe script (named pppoe_test.sh for example) can be used as follows:

```
#!/bin/sh
WAN=$2
OPTIONS=/etc/ppp/options
PAPFILE=/etc/ppp/pap-secrets
CHAPFILE=/etc/ppp/chap-secrets
RESOLV=/etc/ppp/resolv.conf
LINKFILE=/etc/ppp/link
PPPFILE=/var/run/ppp
FIRSTFILE=/etc/ppp/first
FIRSTDEMAND=/etc/ppp/firstdemand
CONNECTFILE=/etc/ppp/connectfile
DNRDPIDFILE=/var/run/dnrd.pid

# Customized here for test
eval `flash set WAN_DHCP 3`
eval `flash set PPP_USER_NAME "zj2"`
eval `flash set PPP_PASSWORD "123"`
eval `flash set PPP_IDLE_TIME 300`
eval `flash set PPP_CONNECT_TYPE 0`
eval `flash set PPP_MTU_SIZE 1460`
eval `flash set DNS_MODE 0`
eval `flash set DNS1 0.0.0.0`
eval `flash set DNS2 0.0.0.0`
eval `flash set DNS3 0.0.0.0`

eval `flash get WAN_DHCP`
eval `flash get PPP_USER_NAME`
eval `flash get PPP_PASSWORD`
eval `flash get PPP_IDLE_TIME`
eval `flash get PPP_CONNECT_TYPE`
eval `flash get PPP_MTU_SIZE`
eval `flash get DNS_MODE`
eval `flash get DNS1`
eval `flash get DNS2`
eval `flash get DNS3`

ifconfig $WAN 0.0.0.0
if [ $1 = 'connect' ]; then
  ENABLE_CONNECT=1
else
 ENABLE_CONNECT=0
fi
if [ -n "$PPP_USER_NAME" ] ; then
   echo "name \"$PPP_USER_NAME\"" > $OPTIONS
   echo "#################################################" > $PAPFILE
   echo "\"$PPP_USER_NAME\"        *         \"$PPP_PASSWORD\"" >> $PAPFILE
   echo "#################################################" > $CHAPFILE
   echo "\"$PPP_USER_NAME\"        *         \"$PPP_PASSWORD\"" >> $CHAPFILE
fi

echo "noauth" >>$OPTIONS
echo "nomppc" >>$OPTIONS
echo "noipdefault" >> $OPTIONS
echo "hide-password" >> $OPTIONS
echo "defaultroute" >> $OPTIONS
echo "persist" >> $OPTIONS
```

```
        echo "ipcp-accept-remote" >> $OPTIONS
        echo "ipcp-accept-local" >> $OPTIONS
        echo "nodetach" >>$OPTIONS
        echo "usepeerdns" >> $OPTIONS
        echo "mtu $PPP_MTU_SIZE" >> $OPTIONS
        echo "mru $PPP_MTU_SIZE" >> $OPTIONS
        echo "lcp-echo-interval 20" >> $OPTIONS
        echo "lcp-echo-failure 3" >> $OPTIONS
        echo "wantype $WAN_DHCP" >> $OPTIONS
        echo "holdoff 10" >> $OPTIONS

        if [ -n "$PPP_SERVICE_NAME" ]; then
          echo "plugin /etc/ppp/plugins/libplugin.a rp_pppoe_service $PPP_SERVICE_NAME $WAN" >>
$OPTIONS
        else
          echo "plugin /etc/ppp/plugins/libplugin.a $WAN" >> $OPTIONS
        fi

        PID_FILE=/var/run/ppp0.pid
        DNRD_PID=/var/run/dnrd.pid

        if [ ! -f $DNRD_PID ]; then
             DNS="--cache=off"
             if [ $DNS_MODE != 1 ]; then
                  dnrd $DNS -s 168.95.1.1
             fi
             if [ $DNS_MODE = 1 ]; then
                  if [ "$DNS1" != '0.0.0.0' ]; then
                       DNS="$DNS -s $DNS1"
                  fi
                  if [ "$DNS2" != '0.0.0.0' ]; then
                       DNS="$DNS -s $DNS2"
                  fi
                if [ "$DNS3" != '0.0.0.0' ]; then
                     DNS="$DNS -s $DNS3"
                fi
              dnrd $DNS
          fi
        fi

        if [ -r "$PPPFILE" ]; then
           rm $PPPFILE
        fi

        killall -15 pppd 2> /dev/null

        if [ -r "$CONNECTFILE" ]; then
           rm -f $CONNECTFILE
        fi

        if [ $PPP_CONNECT_TYPE = 0 ] ; then
        {
           while [ true ]; do

           if [ $WAN_DHCP != 3 ] || [ $PPP_CONNECT_TYPE != 0 ]; then
                break
           fi
```

```
    if [ ! -r "$CONNECTFILE" ]    && [ $PPP_CONNECT_TYPE = 0 ]; then
       echo "pass" > $CONNECTFILE
       if [ ! -f $FIRSTFILE ]; then
          echo "pass" > $FIRSTFILE
       fi
       pppd
    fi
    sleep 5
    done
    rm -f $FIRSTFILE
    } &
 fi

 if [ $PPP_CONNECT_TYPE = 1 ] ; then
 {
    echo "demand" >> $OPTIONS
    echo "idle $PPP_IDLE_TIME" >> $OPTIONS

    while [ true ]; do

    if [ $WAN_DHCP != 3 ] || [ $PPP_CONNECT_TYPE != 1 ]; then
       break
    fi
    if [ ! -r "$CONNECTFILE" ]    && [ $PPP_CONNECT_TYPE = 1 ]; then
       echo "pass" > $CONNECTFILE
       if [ ! -f $FIRSTDEMAND ]; then
          echo "pass" > $FIRSTDEMAND
       fi
       pppd
         if [ -f $DNRDPIDFILE ]; then
             PID=`cat $DNRDPIDFILE`
             kill -9 $PID
             rm -f $DNRDPIDFILE
       fi
       dnrd --cache=off -s 168.95.1.1
    fi
    sleep 5
    done
    rm -f $FIRSTDEMAND
 } &
 fi

 if [ $PPP_CONNECT_TYPE = 2 ]; then
     if [ $ENABLE_CONNECT = 1 ]; then
     pppd &
     fi
 fi
```

Test steps as follows:

Step 1): add pppoe_test.sh as above in users/script/cinit/.

Step 2): make sdk and update image.

Step 3): after AP boots up, at AP console, input command as follows:

/bin/sh -x /bin/pppoe_test2.sh all usb_3g &

Test result: pppoe should dial up via wan interface usb_3g.

Debug:

At AP console, cd /etc/ppp/ and check settings for pppoe as follows:

1) chap-secrets and pap-secrets is secret settings, one example as follows:

```
# cat chap-secrets
####################################################
"zj2"      *            "123"
# cat pap-secrets
####################################################
"zj2"      *            "123"
```

2) options is config settings for pppoe, one example as follows:

```
# cat options
name "zj2"
noauth
nomppc
noipdefault
hide-password
defaultroute
persist
ipcp-accept-remote
ipcp-accept-local
nodetach
usepeerdns
mtu 1460
mru 1460
lcp-echo-interval 20
lcp-echo-failure 3
wantype 3
holdoff 10
plugin /etc/ppp/plugins/libplugin.a eth1
```

7.14 How to enable/disable ipv6 passthru?

Answer:

1.1) At AP mode,enable ipv6 passthru as follows:

```
ifconfig peth0 up
brctl addif br0 peth0
echo "1">/proc/custom_Passthru
```

1.2) At AP 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 goahead?

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, 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, 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, add program-default value for new MIB entry at function writeDefault().

Step 4, in users/goahead-2.1.1/LINUX /upmib.h, 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, 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, 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, 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 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 goahead?

Answer:

The steps to modify webpages at the web-server goahead as follows:

Step 1, in users/goahead-2.1.1/web-gw (where are webpages for router SDK) or users/goahead-2.1.1/web-ap (where are webpages for AP SDK)

a. Add new page.

b. Add new link at code.asp

c. Create new page as *.asp

Step 2, in users/goahead-2.1.1/web-gw /*.asp or users/goahead-2.1.1/web-ap/*.asp, dynamic values are generated and MUST be within "<% getInfo('value_name'); %>".

And in users/goahead-2.1.1/LINUX/fmget.c, add the return value by apmib_get() in "getInfo" function.

Step 3, in users/goahead-2.1.1/web-gw /*.asp or users/goahead-2.1.1/web-ap/*.asp, add goform action which post the settings of webpages to c code.

Ex: in AP\goahead-2.1.1\web-gw\tcpipwan.asp, add source code as follows,

```
<form action=/goform/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. 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
    #if defined(CONFIG_RTL865X_BICOLOR_LED)
    #ifdef BICOLOR_LED_VENDOR_BXXX
    REG32(LEDCR) |= (1 << 19); // 5 ledmode set to 1 for bi-color LED
    REG32(PABCNR) &= ~0x001f0000; /* set port port b-4/3/2/1/0 to gpio */
    REG32(PABDIR) |=    0x001f0000; /* set port port b-4/3/2/1/0 gpio direction-output */
    #else
```

```
        //8650B demo board default: Bi-color 5 LED
        WRITE_MEM32(LEDCR, READ_MEM32(LEDCR) | 0x01180000 ); // bi-color LED
        #endif
        /* config LED mode */
        WRITE_MEM32(SWTAA, PORT5_PHY_CONTROL);
        WRITE_MEM32(TCR0, 0x000002C2); //8651 demo board default: 15 LED boards
        WRITE_MEM32(SWTACR, CMD_FORCE | ACTION_START); // force add
        #else /* CONFIG_RTL865X_BICOLOR_LED */
        /* config LED mode */
        WRITE_MEM32(LEDCR, 0x00000000 ); // 15 LED
        WRITE_MEM32(SWTAA, PORT5_PHY_CONTROL);
        WRITE_MEM32(TCR0, 0x000002C7); //8651 demo board default: 15 LED boards
        WRITE_MEM32(SWTACR, CMD_FORCE | ACTION_START); // force add
        #endif /* CONFIG_RTL865X_BICOLOR_LED */
    #endif
```

Step 3, modify ARCH_CPU_SLEEP default value to 'N' in boards/rtl8196c/config.in

```
config ARCH_CPU_SLEEP
bool
default n
```

make menuconfig and load default value.

Step 4, close watchdog at linux kernel menuconfig

```
System Configuration    --->
    Enable watchdog timer support[]
```

Step 5, re-build image.

Step 6, load linux-2.6.30/vmlinux into ICE and run it.

7.20 The image size and free memory info for RTL8198 + RTL8192C SDK.

Answer:

RTL8198 + RTL8192C demo board v630 run SDK image (2001-01-18), the test result as follows.

Configure \ Test entry	fw.bin (KB)	MemFree (KB)
Default	1928	18008
Enable samba	2784	17688
Enable samba and DLNA	2949	15588

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 speicifid forward port mask

3) examples

example 1:

flood special address 224.1.2.3 if no client join

```
echo "add all ipv4 224.1.2.3 0xffffffff" > /proc/br_igmpDb
```

example 2:

block special address 224.1.2.3 if no client join

```
echo "add all ipv4 224.1.2.3 0" > /proc/br_igmpDb
```

example 3:

del special address 224.1.2.3 record

```
echo "del all ipv4 224.1.2.3 0" > /proc/br_igmpDb
```

example 4:

default to flood ipv4 unknown multicast

```
echo "add all ipv4 0.0.0.0 0xffffffff" > /proc/br_igmpDb
```

example 5:

default to block ipv4 unknown multicast

```
echo "add all ipv4 0.0.0.0 0x0" > /proc/br_igmpDb
```

example 6:

default to flood ipv6 unknown multicast

```
echo "add all ipv6 0x0-0-0-0 0xFFFFFFFF" > /proc/br_igmpDb
```

example 7:

default to block ipv6 unknown multicast

```
echo "add all ipv6 0x0-0-0-0 0x0" > /proc/br_igmpDb
```

Notice ipv6 address must use the following format:

0xAAAAAAAA-BBBBBBBB-CCCCCCCC-DDDDDDDD

4) Note

If user dosen't spcify 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

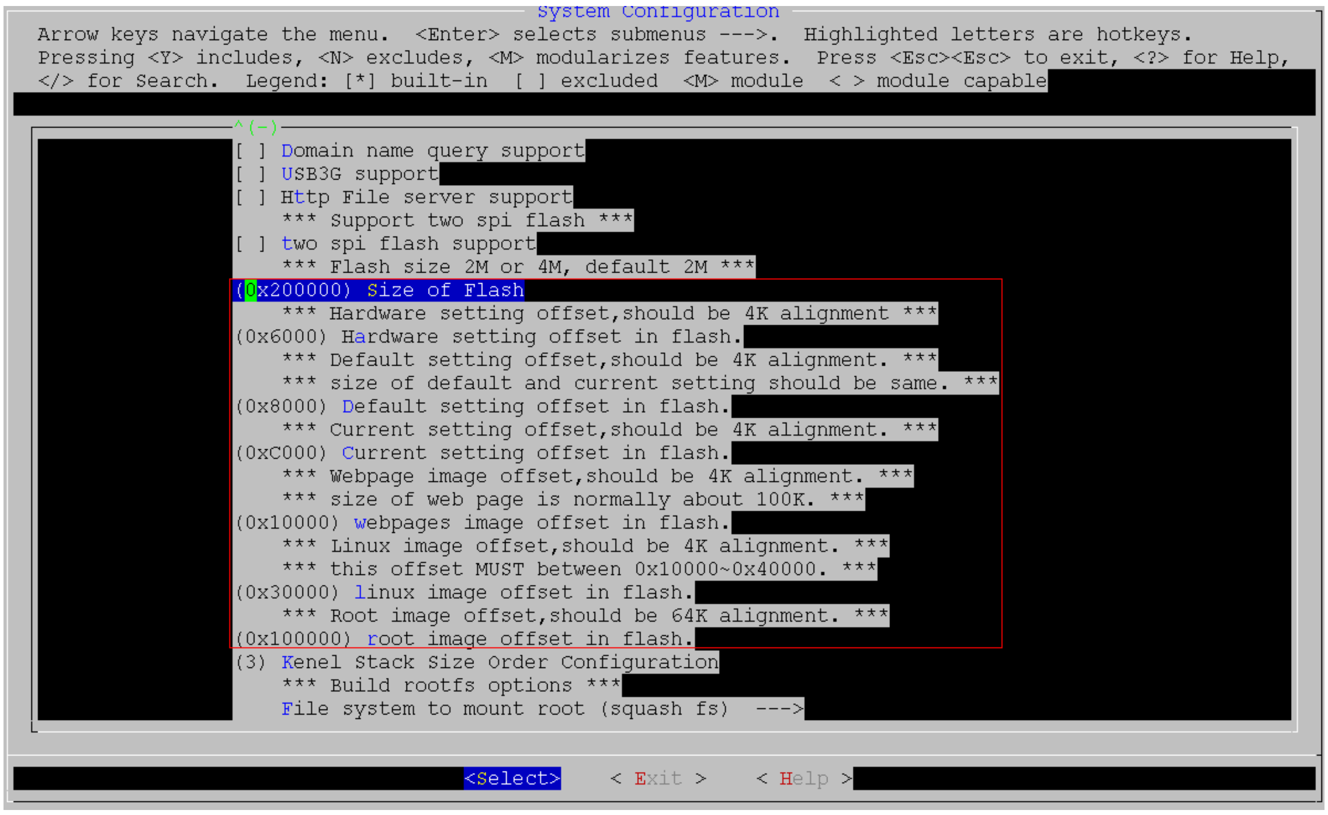
(0x8000) Default setting offset in flash. //customized, should not overwrite current settings in flash

(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) Kenel 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:

	*HW_**WLANX**_11N_TRSWPAPE_C9*	*HW_**WLANX**_11N_TRSWPAPE_CC*
Type 1: 5G TRSW + Ext. PA, 2G TR co-matched (Default)	0	0
Type 2 : 5G TRSW + Int. PA, 2G TR co-matched	170 (=0xAA)	160 (=0xA0)
Type 3: 5G SP3TSW + Int.PA, 2G TR co-matched	170 (=0xAA)	175 (=0xAF)
Type 4: 5G TRSW + Int PA, 2G TRSW + Int PA	0	160 (=0xA0)

If the wlan driver can not find the matching settings, the RTL8192D driver will be initialized as the Type 1: External PA.

7.27 How to configure RTL8192C External PA?

Answer:

RTL8192C use internal PA as the default setting. If the external PA is wanted to be used, the linux kernel configure need to be configured as follows.

make linux_menuconfig //To configure linux menuconfig

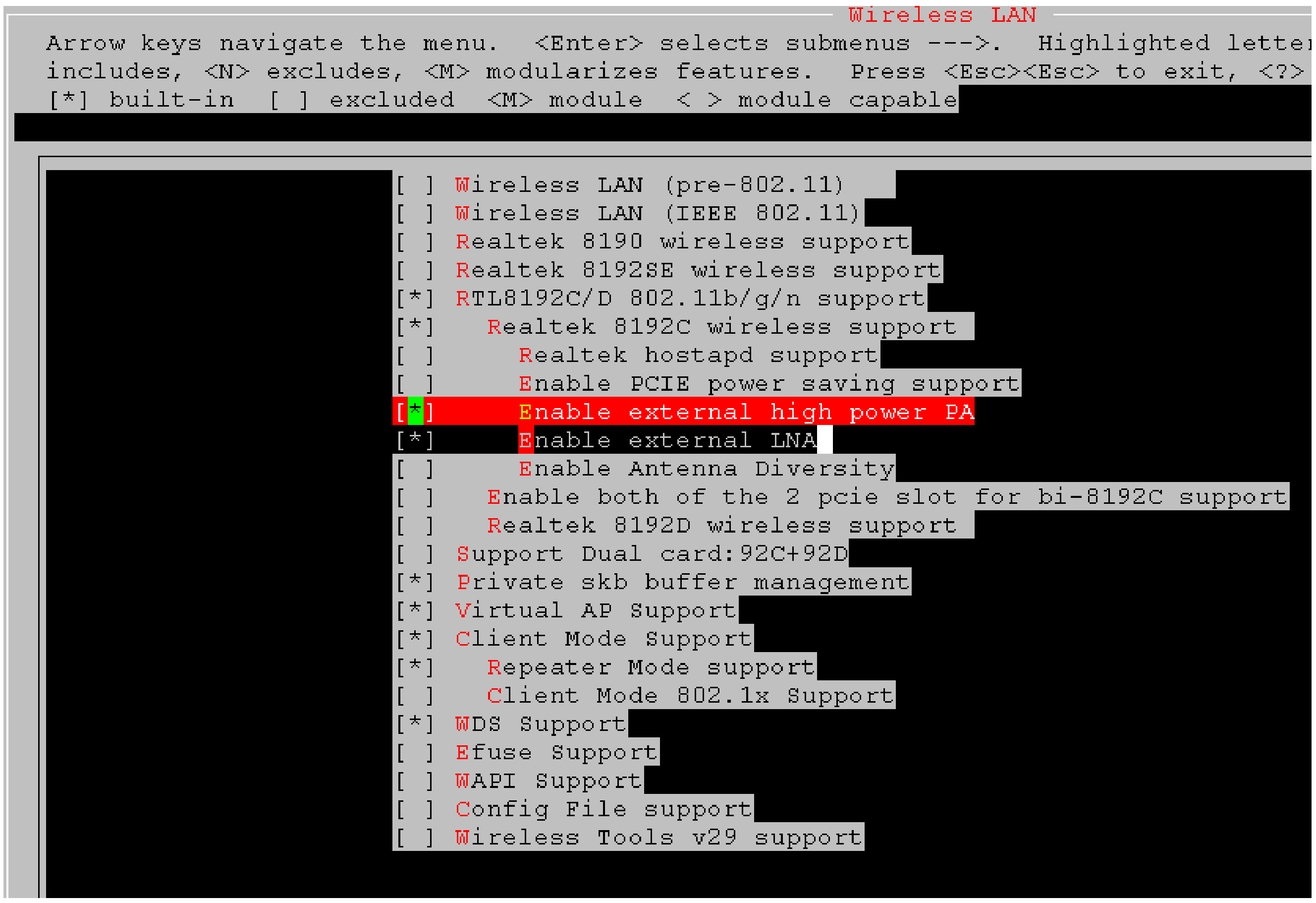
Device Drivers --->

[*] Network device support --->

Wireless LAN --->

[*] Enable external high power PA

[*] Enable external LNA



```
                                 Wireless LAN
  Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted lette
  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 date(without HW setting)?

Answer:

First, build CONFIG-DATA with no HW setting. Modify the file : users/goahead-2.1.1/LINUX/Makefile

./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 date to firmware image(fw.bin). Modify the file board/rtl8198(or rtl8196c ,rtl8196ct etc. Depend on CPU)/Makefile

```
        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/rtl8198t/image; \
                cd -; \
        fi; \
```

Move the sentence back and add the argument *image/*.dat*

```
        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/rtl8198t/image; \
                cd -; \
                $(MGBIN) -c -o $(FW_BIN) image/*.dat $(ROOT_BIN) $(WEBPAGE_BIN)
$(LINUX_BIN); \
        fi; \
```

At last, rebuild, and the image with firmware and config date(without HW setting) fw.bin will be generated.

ATTENTION: Do not upload the image fw.bin with config date via tftp and console(section 3.3), upload it via webpage!(section 3.5)