

REALTEK

Kernel 2.6 SDK User Guide

Realtek Kenel 2.6 SDK User Guide

Date: 2010/12/01

Version: 1.12

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

Change History

Version	Date	Remarks
1.0	2010/04/06	Initial Release
1.1	2010/07/28	1) Change Jungle to kernel 2.6; 2) Add compile environment.
1.2	2010/08/03	Add Section 2.4 and add Chapter 4~6.
1.3	2010/08/20	1) Modify section 4.3; 2) Add section 4.4; 2) Add section 4.8~4.11; 3) Add chap 6; 4) Add section 7.3~7.5
1.4	2010/08/27	1) Modify section 4.11; 2) Add section 7.6~7.7
1.5	2010/09/21	Add section 4.12 and 6.16
1.6	2010/09/30	1) Modify section 6.8; 2) Add section 7.8~7.9
1.7	2010/10/22	Add section 4.12.4 and 4.13~4.14
1.8	2010/10/27	Add section 4.15~4.16
1.9	2010/11/05	Add section 4.17
1.10	2010/11/12	Add section 4.18 and 7.10
1.11	2010/11/16	Add section 4.19~4.22
1.12	2010/12/01	1) Modify section 4.18 for 92D; 2) Add section 4.23

1. Kenel 2.6 SDK introduction	1
2. Compile Kenel 2.6 SDK	2
2.1 <i>Compile Environment.....</i>	2
2.2 <i>Compile bootloader.....</i>	2
2.3 <i>Compile rxlinux.....</i>	3
2.4 <i>Compile rxlinux detailed</i>	5
3. Kenel 2.6 SDK images upload	7
3.1 <i>Topology to upload image</i>	7
3.2 <i>Upload bootloader image.....</i>	7
3.3 <i>Upload rxlinux image.....</i>	8
4. Features configure	10
4.1 <i>PCI support</i>	10
4.2 <i>USB support.....</i>	11
4.3 <i>Samba support.....</i>	12
4.4 <i>USB 3G support.....</i>	17
4.5 <i>WAPI support.....</i>	22
4.6 <i>Mesh support</i>	22
4.7 <i>VLAN support.....</i>	22
4.8 <i>I2C UART support.....</i>	23
4.9 <i>DLNA support.....</i>	25
4.10 <i>Pocket AP support.....</i>	29
4.11 <i>Wireless client mode 802.1x support.....</i>	30
4.12 <i>Hardware NAT.....</i>	31
4.13 <i>Iptables rule to ACL rule translation support.....</i>	34
4.14 <i>Hardware QoS support</i>	35
4.15 <i>IPv6 support.....</i>	35
4.16 <i>64K/sector SPI flash support</i>	39
4.17 <i>Two SPI flash support</i>	39
4.18 <i>92C/D support.....</i>	41
4.19 <i>GDB server support</i>	48
4.20 <i>HTTP file server support</i>	49
4.21 <i>Hostapd support.....</i>	57
4.22 <i>MP support.....</i>	60
4.23 <i>AP mode support</i>	63
5. FLASH tools	64
5.1 <i>flash.....</i>	64
5.2 <i>cvcfg-gw</i>	64

5.3	<i>compweb</i>	64
5.4	<i>cvimg</i>	65
5.5	<i>mgbin</i>	65
6.	Proc file format	67
6.1	<i>br_igmpProxy</i>	67
6.2	<i>br_igmpsnoop</i>	67
6.3	<i>custom_Passthru</i>	67
6.4	<i>enable_dos</i>	68
6.5	<i>fast_l2tp</i>	69
6.6	<i>fast_nat</i>	69
6.7	<i>fast_pptp</i>	70
6.8	<i>filter_table</i>	70
6.9	<i>gpio</i>	71
6.10	<i>pptp_conn_ck</i>	71
6.11	<i>qos</i>	72
6.12	<i>rf_switch</i>	72
6.13	<i>rtk_vlan_support</i>	72
6.14	<i>mib_vlan</i>	73
6.15	<i>sw_nat</i>	74
6.16	<i>hw_nat</i>	74
7.	FAQ.....	76
7.1	<i>How to modify the mappings of LAN/WAN port?</i>	76
7.2	<i>How to change the baudrate of console?</i>	76
7.3	<i>How to use vlan priority option of iptables?</i>	76
7.4	<i>Relationship of virtual AP and root AP?</i>	76
7.5	<i>How to customize icon of lld2d?</i>	77
7.6	<i>How to rename the AP when shown on Windows?</i>	77
7.7	<i>How to add MAC address filter based on ether driver API?</i>	77
7.8	<i>How to adjust CPU speed?</i>	80
7.9	<i>How to shutdown power of Ethernet ports?</i>	80
7.10	<i>How to disable Ethernet PHY?</i>	81

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 and RTL8198 for Kenel 2.6 SDK while rtl819x-sdk-v2.x/rtl819x stores applications and linux kernel.

2. Compile Kenel 2.6 SDK

We assume that Kenel 2.6 SDK is stored at root directory ('/').

2.1 Compile Environment

Red Hat Enterprise [Recommend]

Fedora 9

Ubuntu 8.10/9.10

2.2 Compile bootloader

Steps to compile bootloader are as follows:

Step 1, enter bootloader directory.

```
cd /RLXLinux/bootcode_rtl8196c_98
```

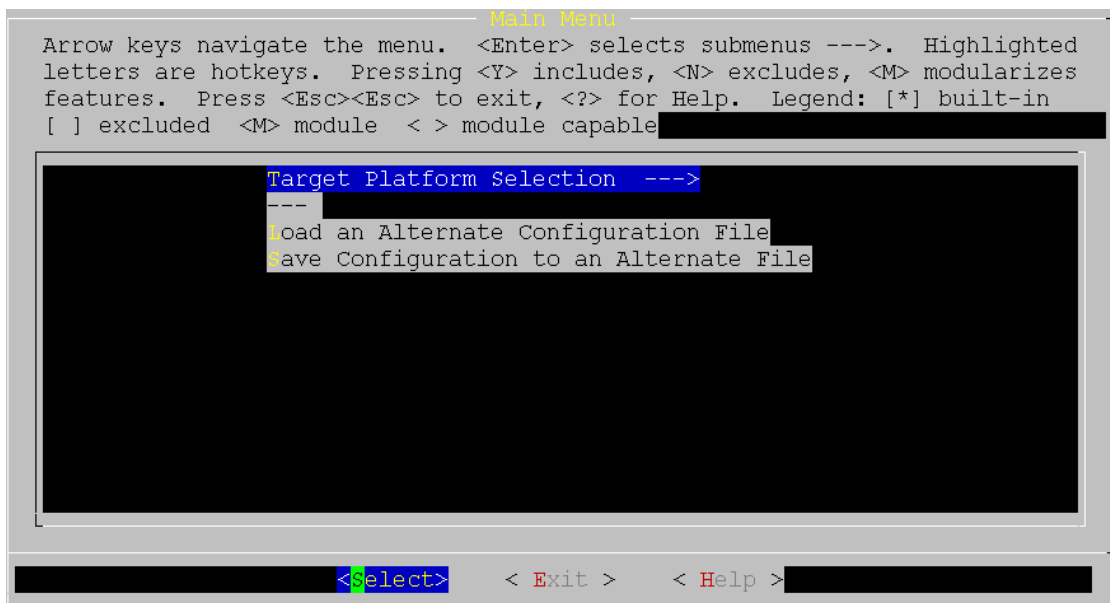
Step 2, choose default configure file.

For 8196C: `cp def-rtl8196c-config .config`

For 8198: `cp def-rtl8198-config .config`

Step 3, make default configure file choosed effective.

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



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

Note:

Above we just use default configure file for bootloader settings.

If we want to configure bootloader different from default configure file,
select 'Target Platform Selection' and click 'Enter' to customize settings of bootloader.



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

Step 4, compile bootloader.

make

2.3 Compile rlxlinux

Steps to compile applications and linux are as follows:

Step 1, enter rtl819x-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 or rtl8198 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 >

```

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'.

```
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 (rle0272_rlx5181) --->
Selected Kernel (linux-2.6.30) --->
Selected Busybox (busybox-1.12) --->
Selected toolchain (rsdk-1.3.6-4181-EB-2.6.30-0.9.30) --->
--- config components
[ ] Config kernel (NEW)
[ ] Config users (NEW)
[ ] Config busybox (NEW)
[*] Load default settings
[ ] Save default settings (NEW)
---
Load an Alternate Configuration File

+ (+)
<Select> < Exit > < Help >
```

```
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

2.4 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 or rtl819x-sdk-v2.x/rtl819x/boards/rtl8196c) 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

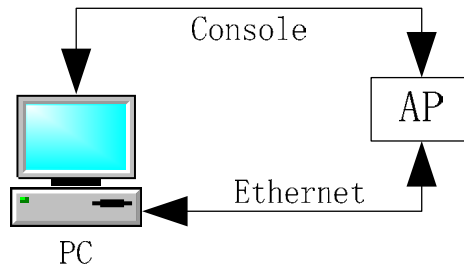
To make images, tools such as cvcfg-gw, compweb, cvimg, mgbin are stored at rtl819x-sdk-v2.x/rtl819x/users/goahead-2.1.1/LINUX.

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:



3.2 Upload bootloader image

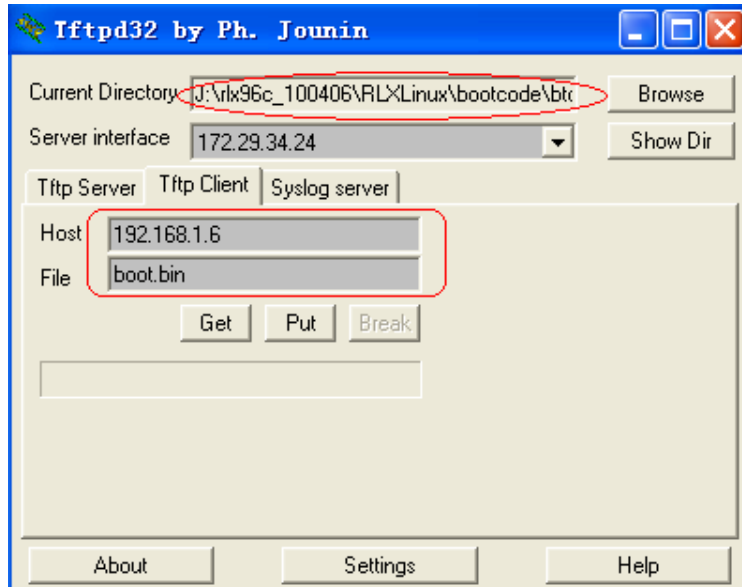
Via 2.1 bootloader compilation, bootloader image is generated at /RLXLinux/bootcode_rtl8196c_98/btcode/ which is boot.bin .

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:



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 and linux.bin .

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. Output of AP's console is as follows:

```

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

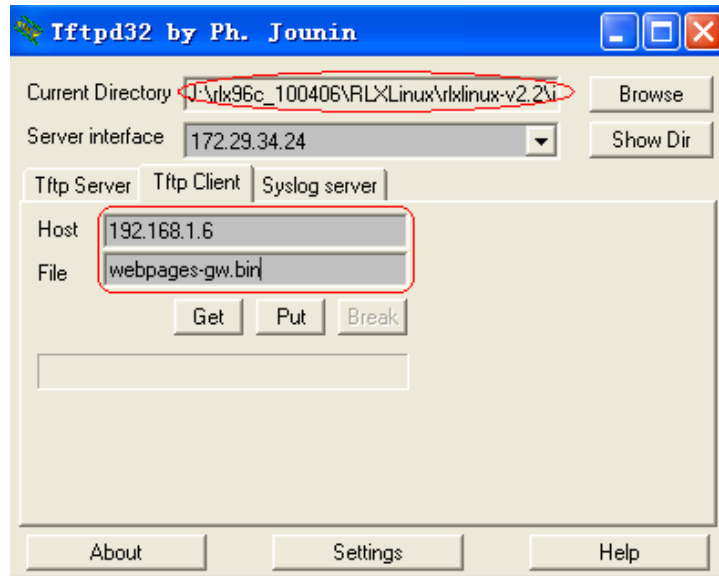
---RealTek(RTL8196C)at 2010.03.29-16:14+0800 version v1.1a [16bit](390MHz)
no sys signature at 00010000!
no sys signature at 00020000!
Set 8196C PHY Patch OK

---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 and linux.bin are the similar, just using root.bin and linux.bin as 'File' input at tftp UI instead.



Step 3, click 'Put' at tftp UI.

4. Features configure

4.1 PCI support

4.1.1 Linux kernel PCI not support

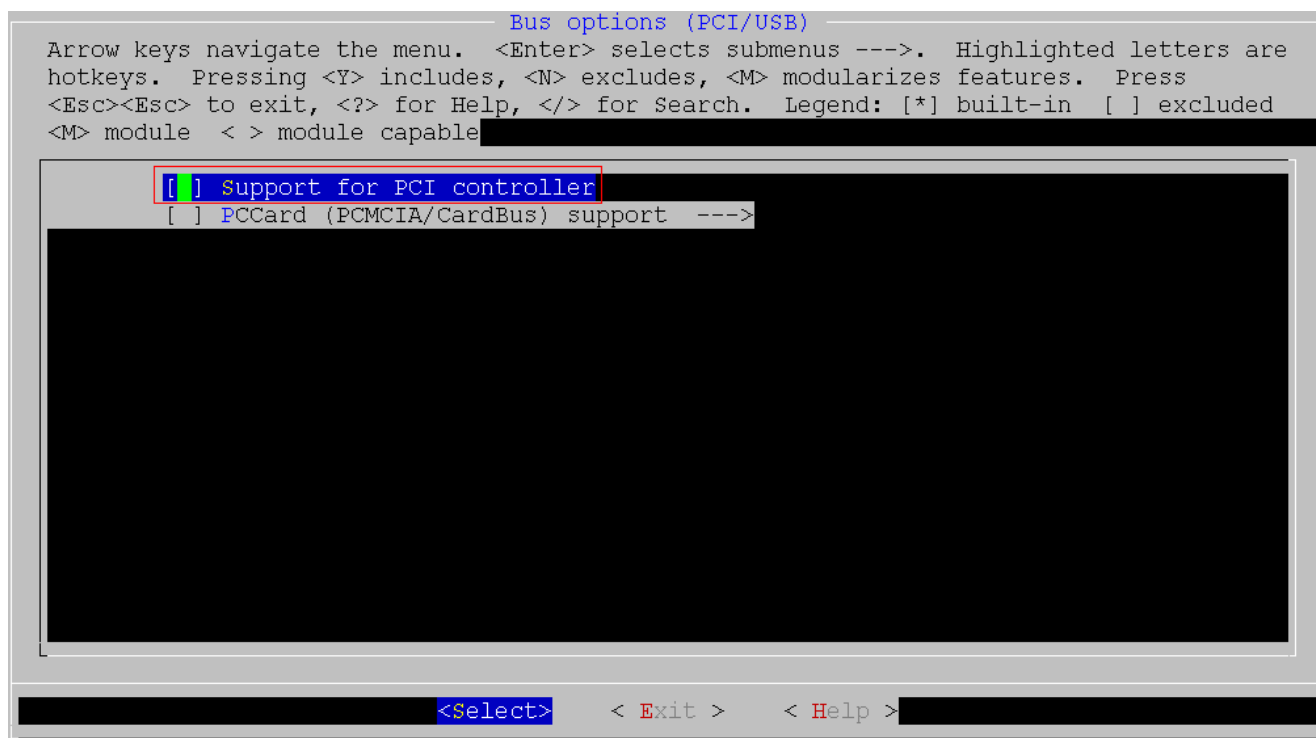
By kernel default settings, linux kernel PCI is not supported, because wireless driver use its own PCI driver. Linux kernel configure as follows.

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

Menuconfig:

Bus options (PCI/USB) --->

Support for PCI controller // Not selected



4.1.2 Linux kernel PCI support

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

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

Menuconfig:

Bus options (PCI/USB) --->

Support for PCI controller // Selected

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

```

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:

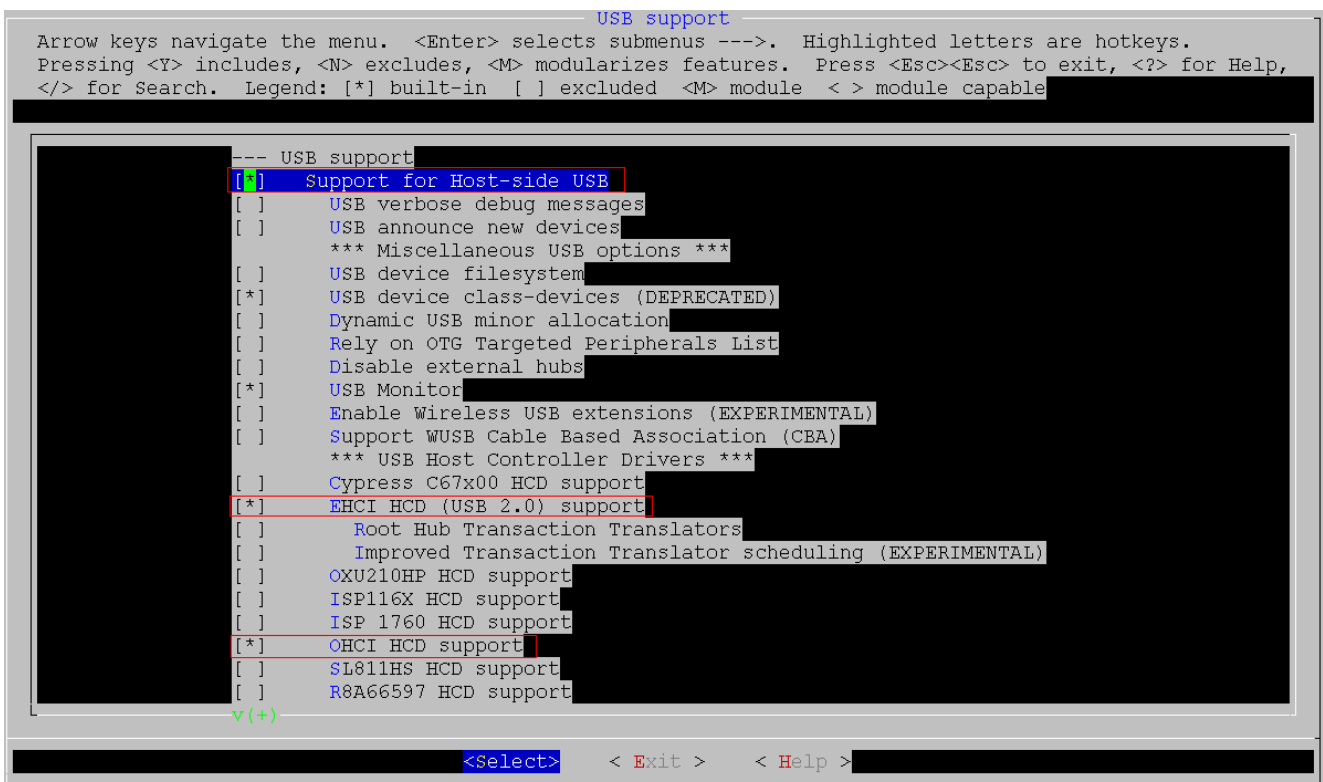
Device Drivers --->

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



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(+)
```

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
```

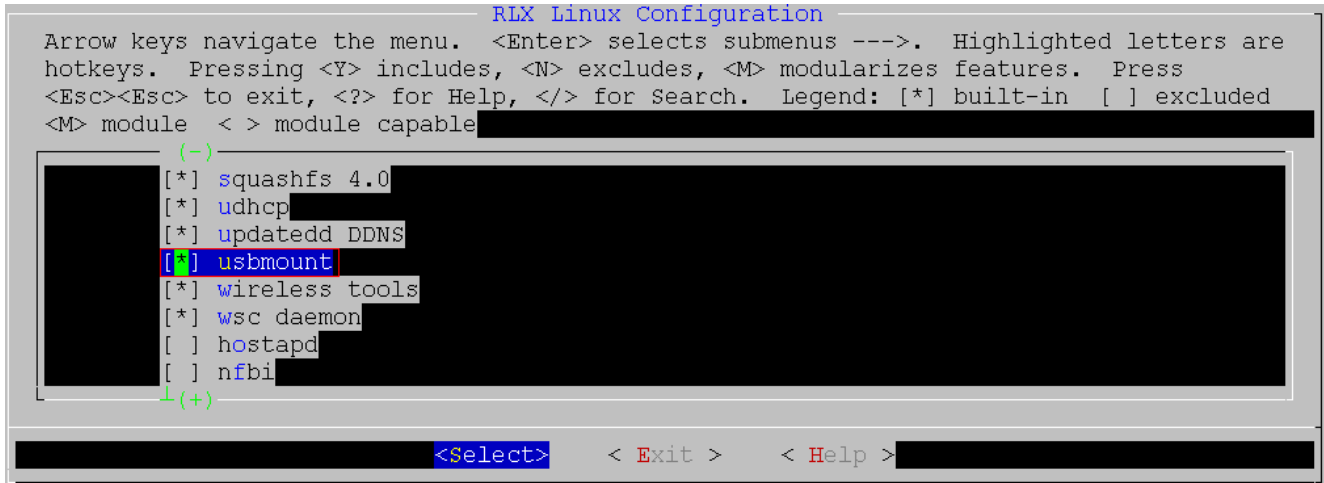
4.3.2 Samba application configure

4.3.2.1 enable usbmount

`make users_menuconfig` // To configure application settings

Menuconfig:

`usbmount` // selected



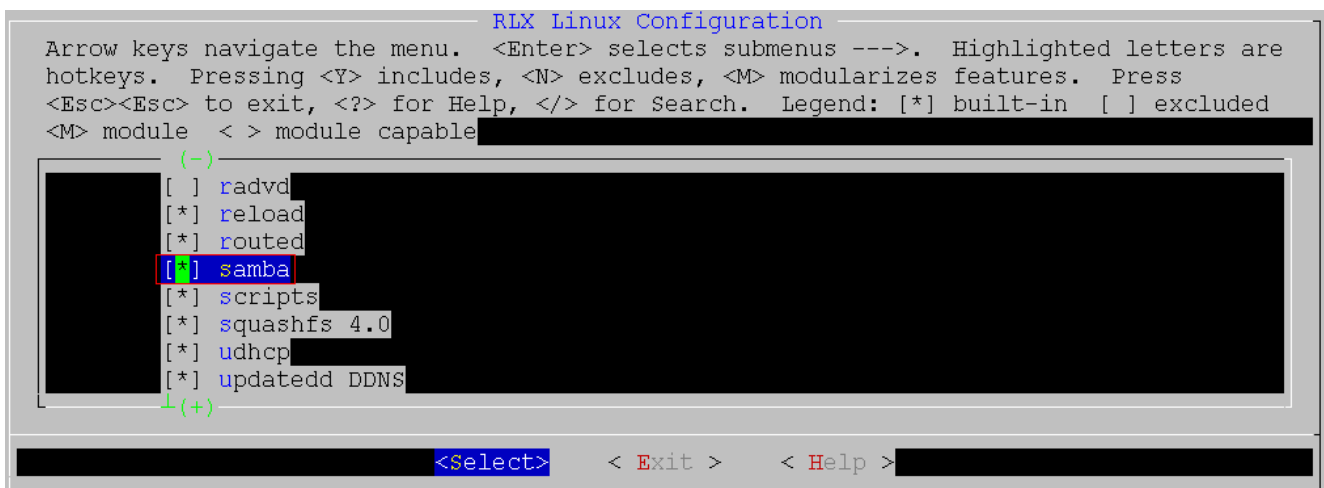
4.3.2.2 enable samba

`make users_menuconfig` // To configure application settings

Menuconfig:

`samba` // selected

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



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:

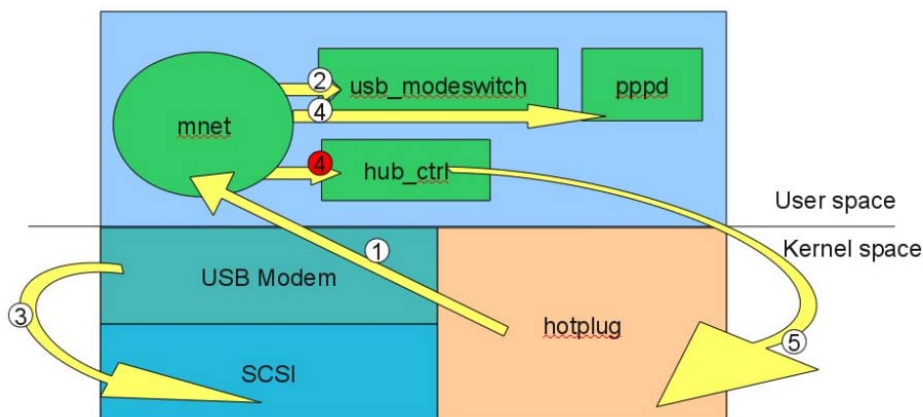
```
#echo "testuser:x:0:0:testuser:/:/dev/null" >> /var/passwd
#echo "[testuser]
comment = testuser's stuff
path = /var/log
valid users = testuser
public = no
writable = no
printable = no
create mask = 0765" > /var/smbuser.conf
# smbpasswd -a testuser
New SMB password:                // input new SMB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

Note: at present, only rtl8198 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) Kernel Stack Size Order Configuration
```

4.4.3 User configure

1) User space programs

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

2) Busybox command

Linux System Utilities --->

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

4.4.4 Add new USB 3G dongle support

4.4.4.1 Modify mnet

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

```
-detach
noauth
/dev/ttyUSB0

115200
debug
defaultroute
ipcp-accept-local
ipcp-accept-remote
usepeerdns
crtscts
lock
nocc
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

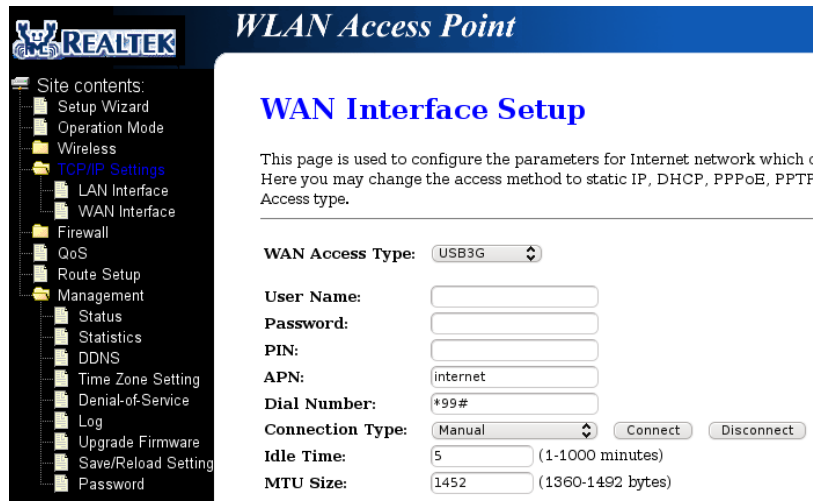
CustomMode="qisda"
CheckSuccess=20

```

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.

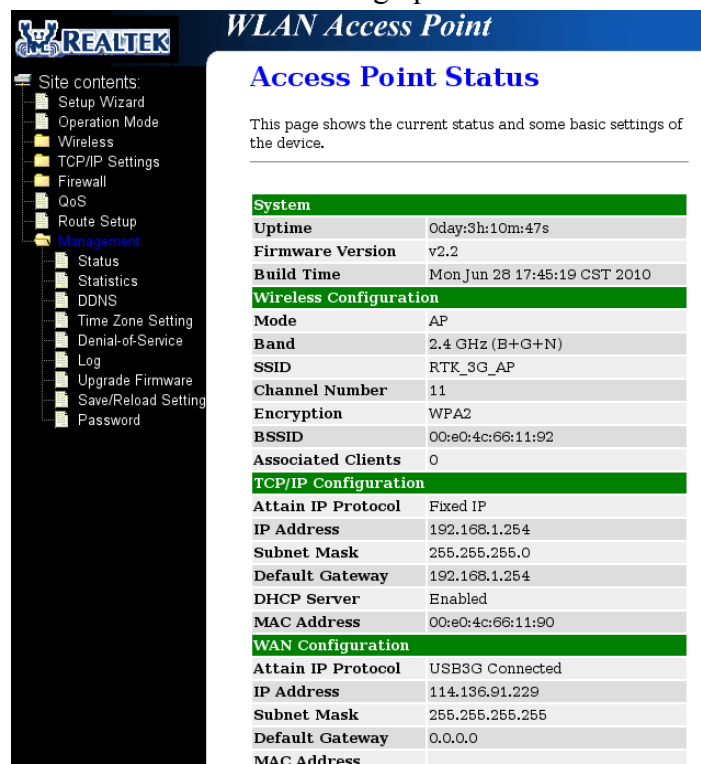


4.4.5.2 Connection status

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 mnet is starting eject process.
- ③ 3G Dialing...: ejected and attached as modem device. Trying to establish a PPP connection.
- ④ 3G Connected: PPP connection has been established.
- ⑤ 3G Disconnected: PPP connection hang up.



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

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

Menuconfig:

Device Drivers --->

Network device support --->

Wireless LAN --->

WAPI Support // selected if WAPI support

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

Note: 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.

4.6 Mesh support

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

Menuconfig:

Device Drivers --->

Network device support --->

Options for Realtek SoC --->

IEEE 802.11s mesh support // selected if mesh support

4.7 VLAN support

```
make linux_menuconfig // 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 menuconfig // To configure bootloader settings

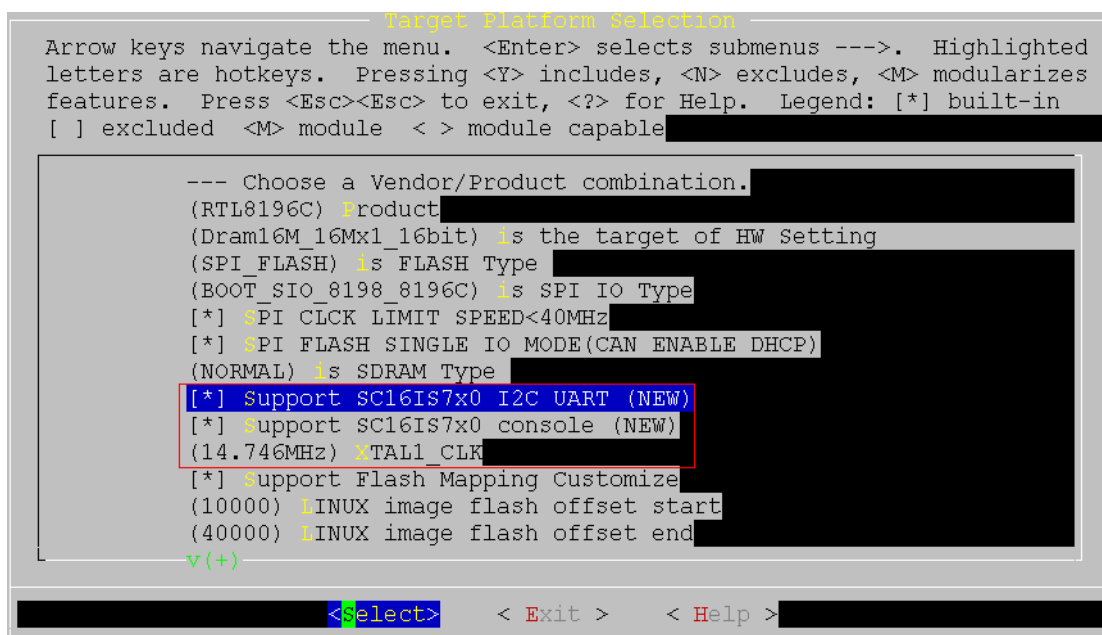
Menuconfig:

Target Platform Selection --->

Support SC16IS7x0 I2C UART // selected

Support SC16IS7x0 console // selected

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



4.8.2 Enable I2C UART in kernel

4.8.2.1 Modify code for GPIO pin in kernel

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

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

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

4.8.2.2 kernel menuconfig

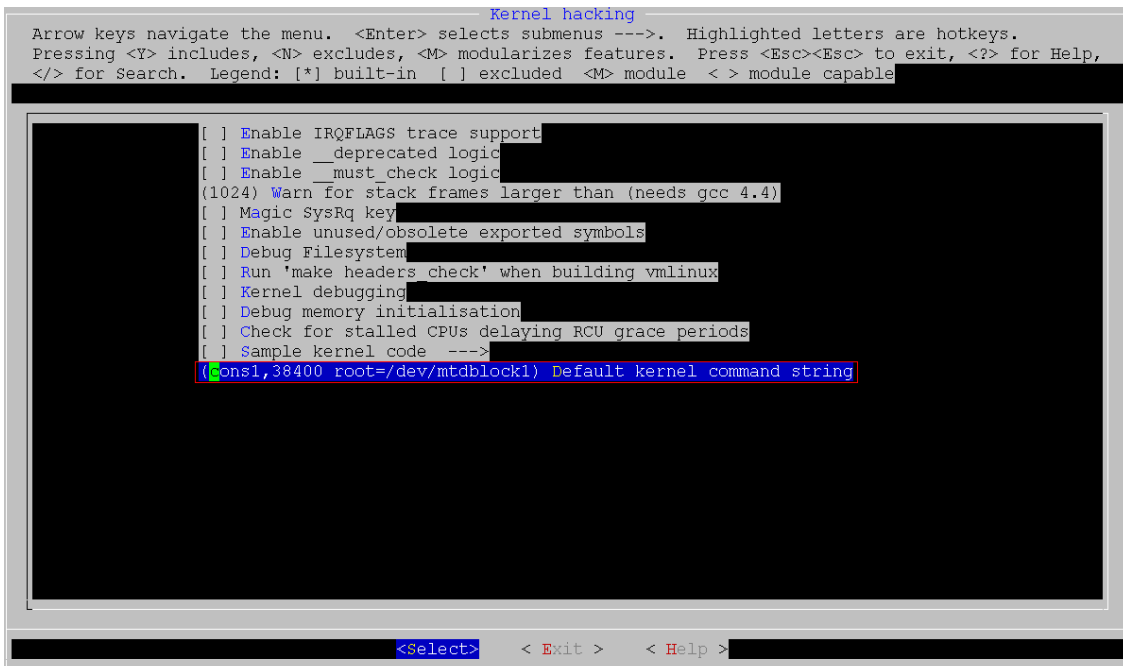
1) Set `ttys1` as console tty.

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

Menuconfig:

Kernel hacking --->

(console=ttys1,38400 root=/dev/mtdblock1) Default kernel command string



2) Modify the 8250/16500 serial port number.

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

Menuconfig:

Device Drivers --->

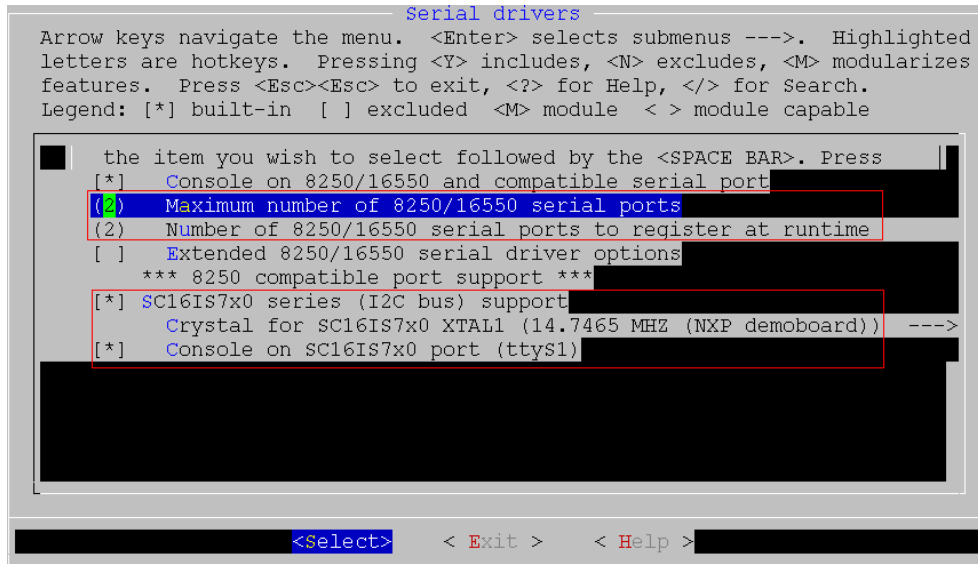
Character devices --->

Serial drivers --->

Maximum number of 8250/16550 serial ports = 2

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

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



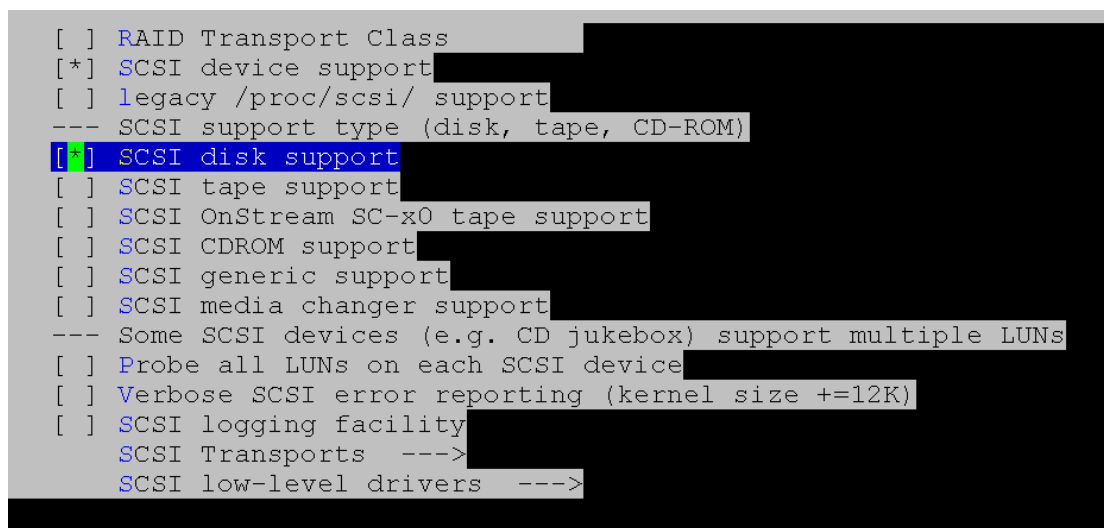
4.9 DLNA support

4.9.1 Linux kernel configure

4.9.1.1 SCSI support

Device Drivers --->

SCSI device support --->



4.9.1.2 File systems configure

File systems --->

DOS/FAT/NT Filesystems --->

```

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

```

File systems --->

```

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

```

Native Language Support --->

```

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

```

4.9.1.3 Enable USB support

Linux kernel configure for USB as follows.

make linux_menuconfig // To configure linux kernel settings

Menuconfig:

Device Drivers --->

USB support --->

Support for Host-side USB // selected

EHCI HCD (USB 2.0) support // selected

OHCI HCD support // selected

USB Mass Storage support // selected

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

4.9.1.4 General setup

General setup --->

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

4.9.2 Application configure

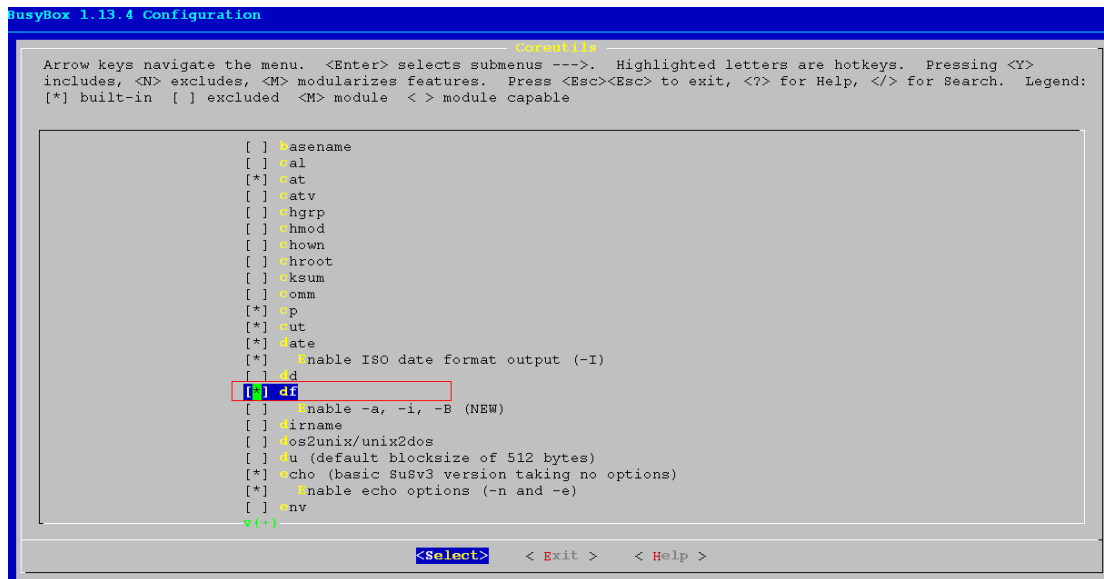
4.9.2.1 busybox configure

make menuconfig and select Config busybox

Menuconfig:

Coreutils--->

df support // selected

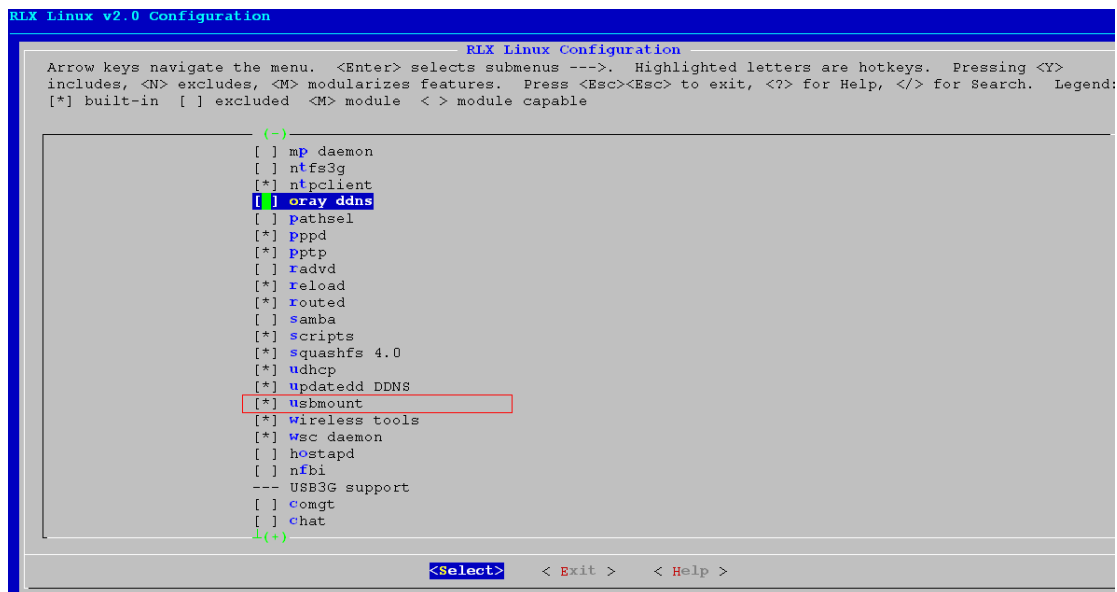


4.9.2.2 enable usbmount

`make users_menuconfig // To configure application settings`

Menuconfig:

`usbmount // selected`

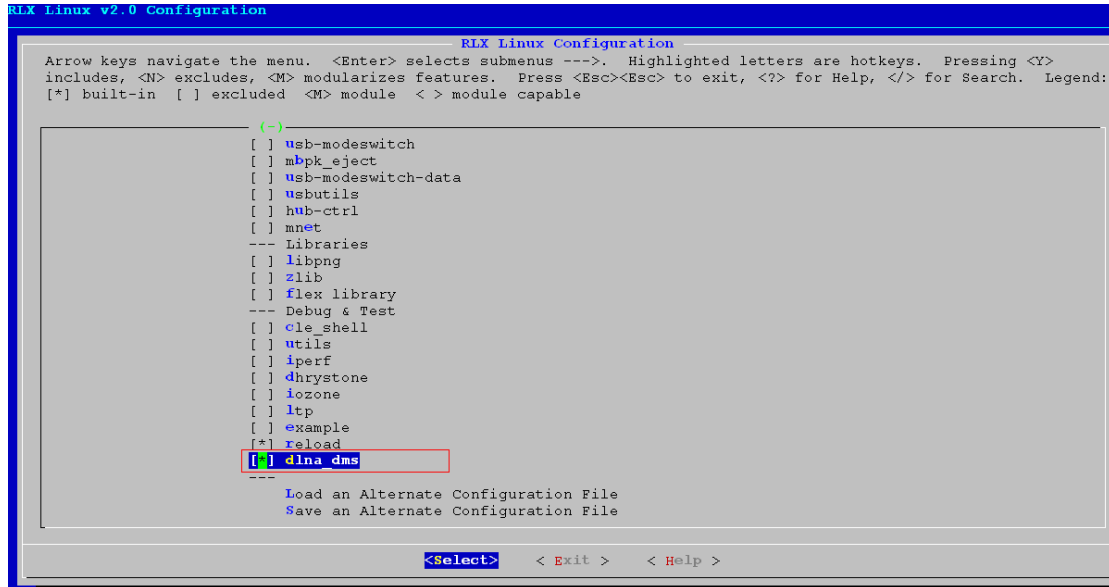


4.9.2.3 enable dlna

`make users_menuconfig // To configure application settings`

Menuconfig:

`Dlna_dms // selected`



4.9.3 Test dlina 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 dlina as a daemon.
- (4) Then you can enjoy the multimedia in your flashdisk with PS3 or XBOX and so on.

4.10 Pocket AP support

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

make menuconfig // To configure linux kernel settings

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

```

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_menuconfig // To configure linux kernel settings
```

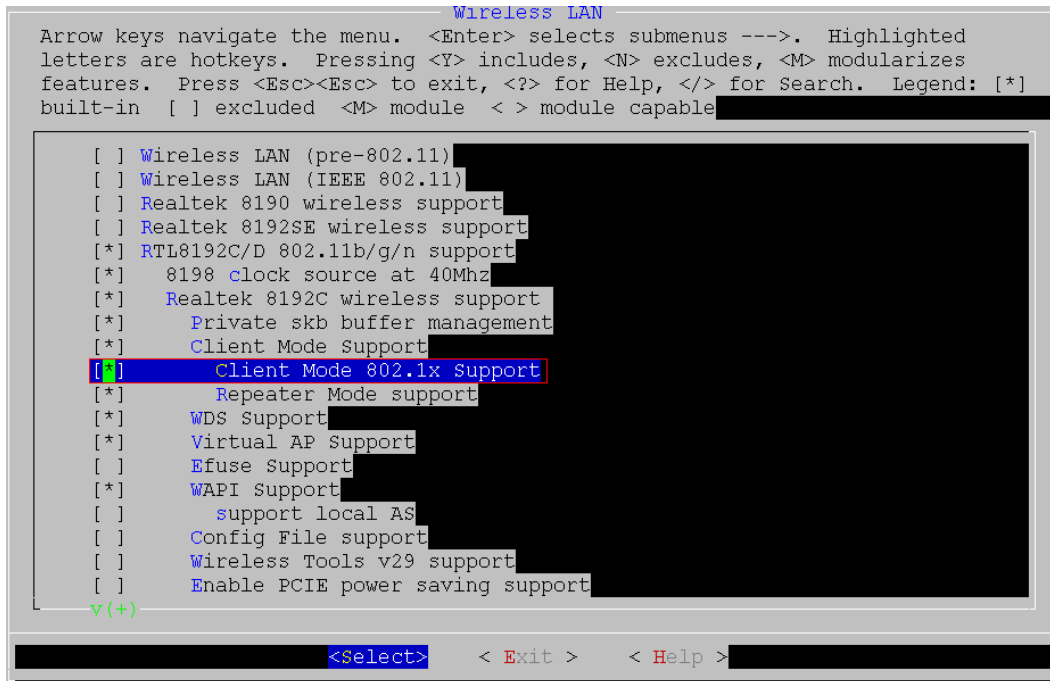
Menuconfig:

```
Device Drivers --->
```

```
Network device support --->
```

```
Wireless LAN --->
```

```
Client Mode 802.1x Support // selected
```



4.12 Hardware NAT

4.12.1 Hardware NAT description

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

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

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

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

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

```

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

```

4.12.2 Proc files related to hardware NAT

1) Name: fast_nat

Path: /proc/fast_nat

Description: Flags for fast path control

Input Format:

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

Input Para.:

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

Output Format:

```
$FLAG
```

Output Para.:

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

2) hw_nat

Path: /proc/hw_nat

Description: Flags for hardware NAT control

Input Format:

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

Input Para.:

- * the unit is non-zero: hardware NAT enabled
- * the unit is 0: hardware NAT disabled
- * "echo 0 > /proc/hw_nat" : hardware NAT disabled, change to gateway mode.
- * "echo 1 > /proc/hw_nat" : hardware NAT enabled, change to gateway mode.
- * "echo 2 > /proc/hw_nat" : Change to bridge mode.
- * "echo 3 > /proc/hw_nat" : Change to WISP mode.
- * "echo 4 > /proc/hw_nat" : simply disabled the hardware NAT.
- * "echo 5 > /proc/hw_nat" : simply disabled the hardware NAT.
- * "echo 8 > /proc/hw_nat" : simply disabled the hardware NAT.
- * "echo 9 > /proc/hw_nat" : init hardware NAT parameters. (Must init before hardware

NAT works)

Output Format:

```
$FLAG
```

Output Para.:

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

3) Name: nf_contrack

Path: /proc/net/nf_contrack

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

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 contracks 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 fragmentation will be automatically processed, no other settings involved.

4.12.4 Enable/Disable hardware NAT support

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

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

Menuconfig:

Device Drivers --->

Network device support --->

Options for Realtek SoC --->

Config for Layered Driver Features --->

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

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

Enable RTL Hardware NAPT

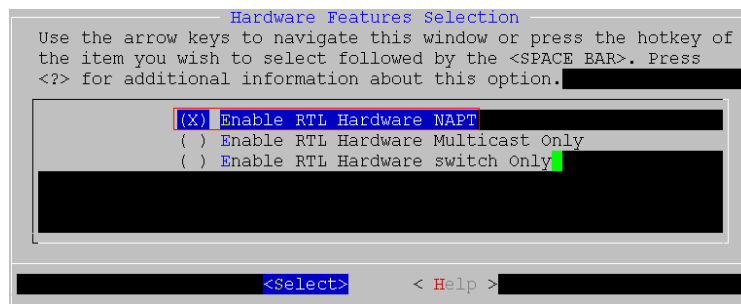
/* If selected, enable hardware L2 and multicast function */

Enable RTL Hardware Multicast Only

/* If selected, enable hardware L2 function */

Enable RTL Hardware switch Only

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



4.13 Iptables rule to ACL rule translation support

In order to offload cpu when firewall is enabled by iptables rules, the feature of iptables rule to ACL rule translation can be enabled.

Enable iptables rule to ACL rule translation support as follows:

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

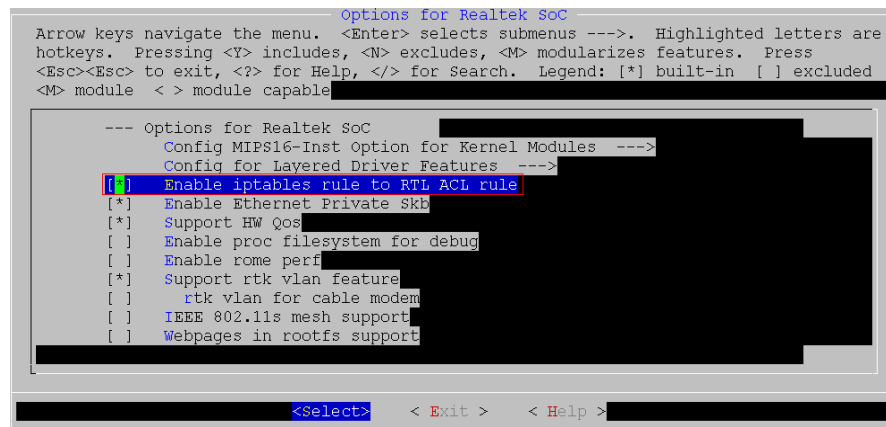
Menuconfig:

Device Drivers --->

Network device support --->

Options for Realtek SoC --->

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



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_menuconfig // To configure linux kernel settings
```

Menuconfig:

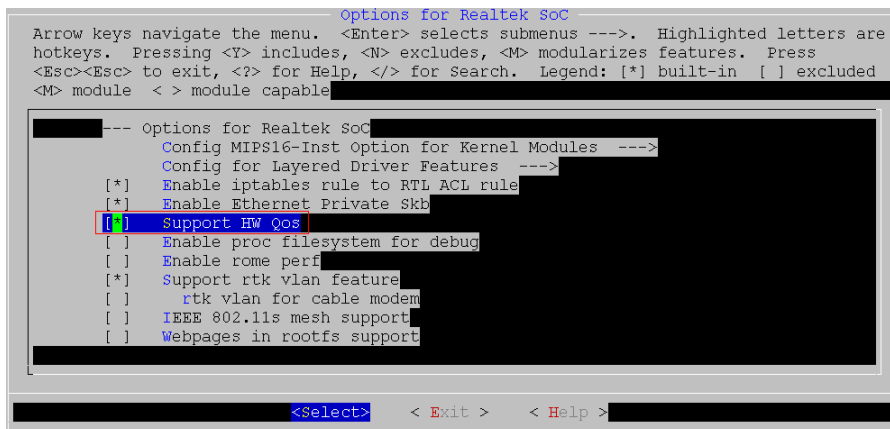
Device Drivers --->

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.



4.15 IPv6 support

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

4.15.1 Linux kernel for IPv6 configure

Configure Linux kernel for IPv6 as follows:

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

Menuconfig:

1) Networking support --->

Networking options--->

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

```

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

```

2) Networking support --->

Networking options--->

The ipv6 protocol--->

IPv6: ready logo patch // Selected

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

```

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_menuconfig // 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
[ ] dnssd
[ ] 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
[ ] net
[ ] 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, Linux kernel configure as follows (here we take a 64K/sector 8M SPI flash as an example) :

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

Menuconfig:

System Configuration --->

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



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:

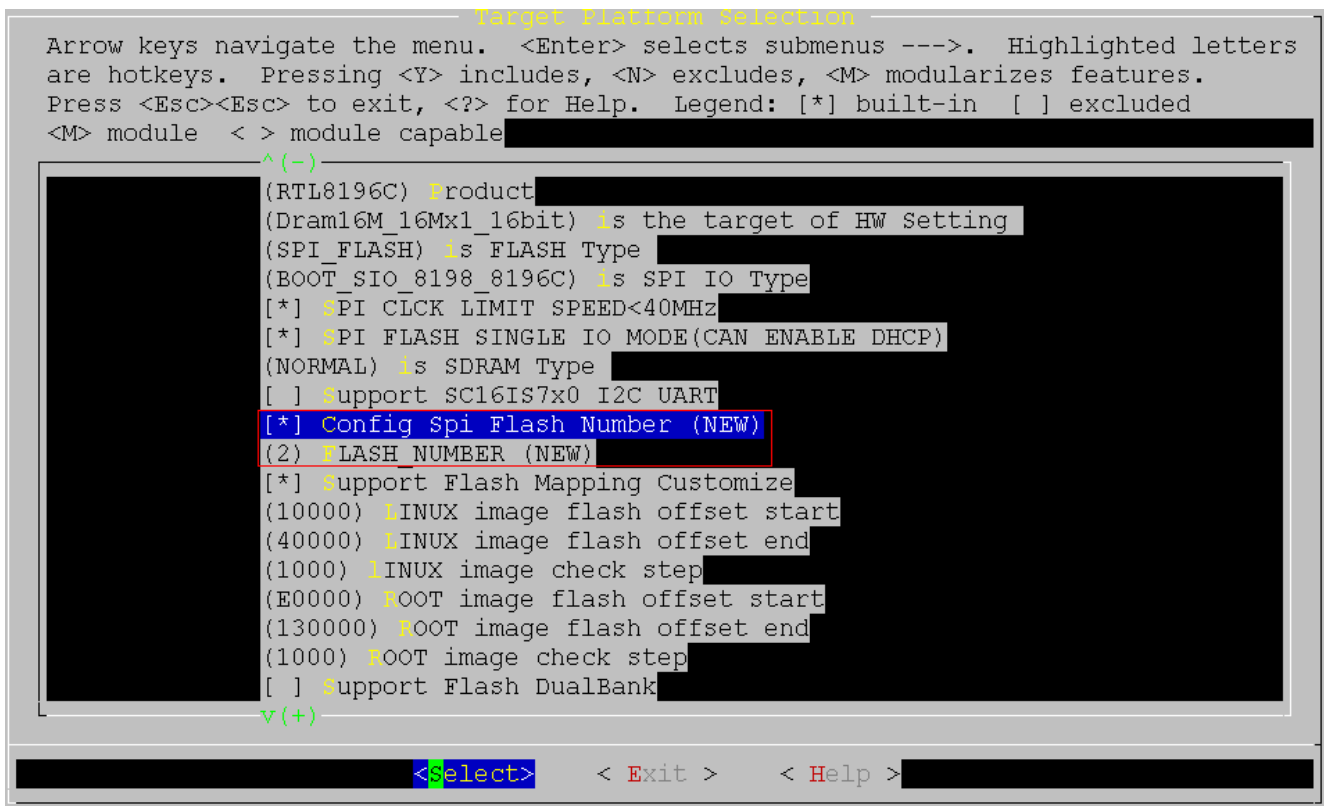
Target Platform Selection --->

//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:

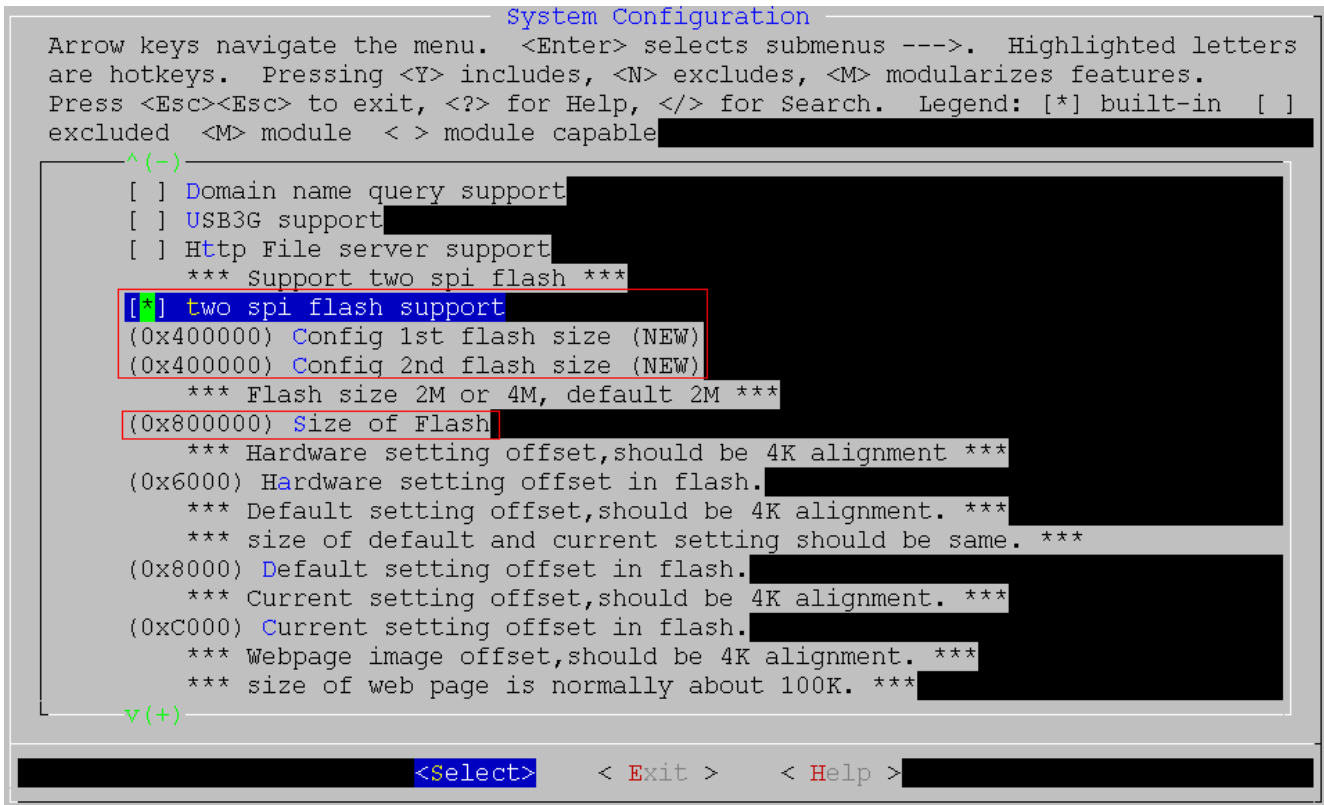
System Configuration --->

[*] 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



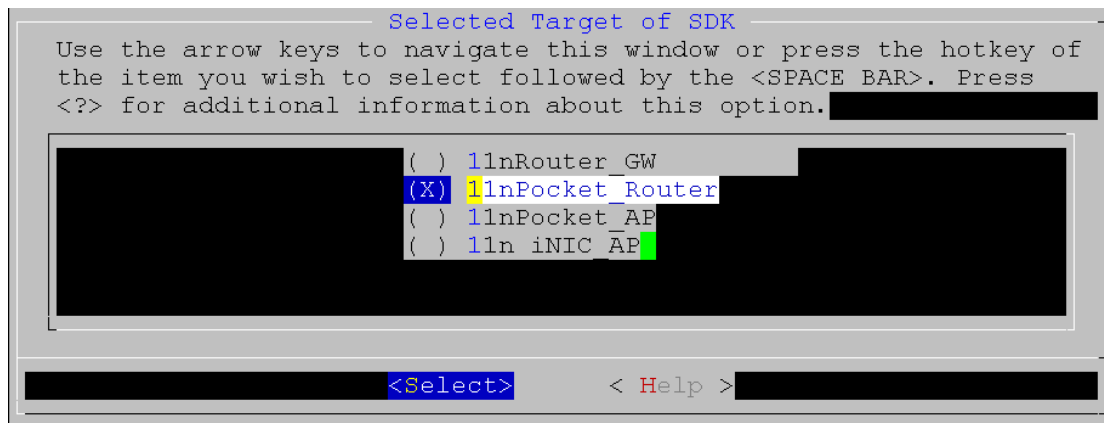
4.18 92C/D support

4.18.1 92C support for Pocket AP SDK

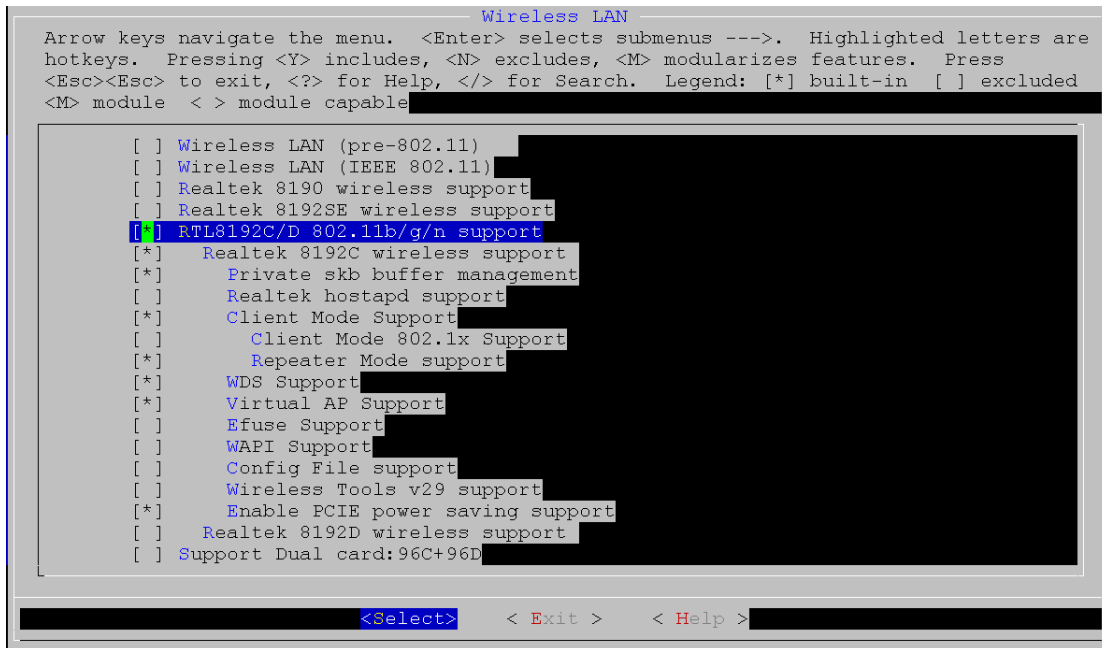
1) Select SDK for Pocket AP 92C:

Selected Target of SDK --->

And selected "11nPocket_Router"



2) Enable 92C Driver

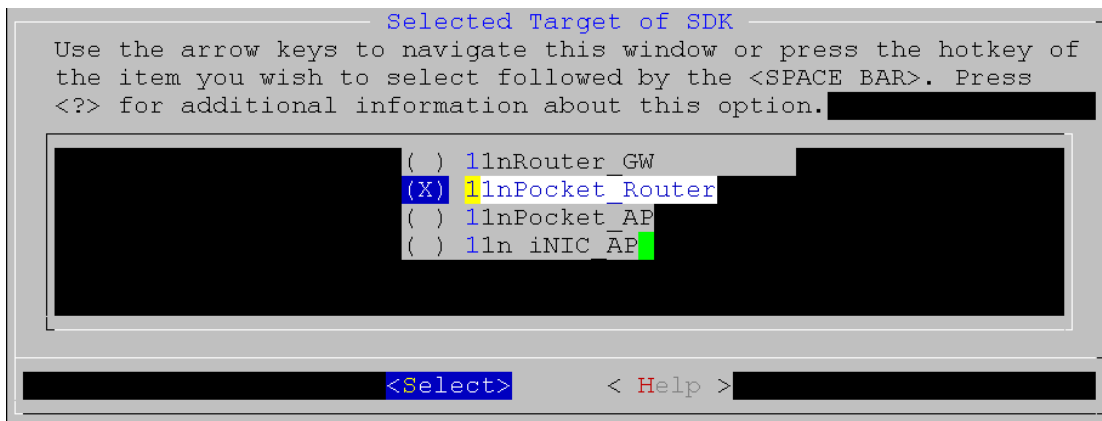


4.18.2 92D support for Pocket AP SDK

1) Select SDK for Pocket AP 92D:

Selected Target of SDK --->

And selected "11nPocket_Router"



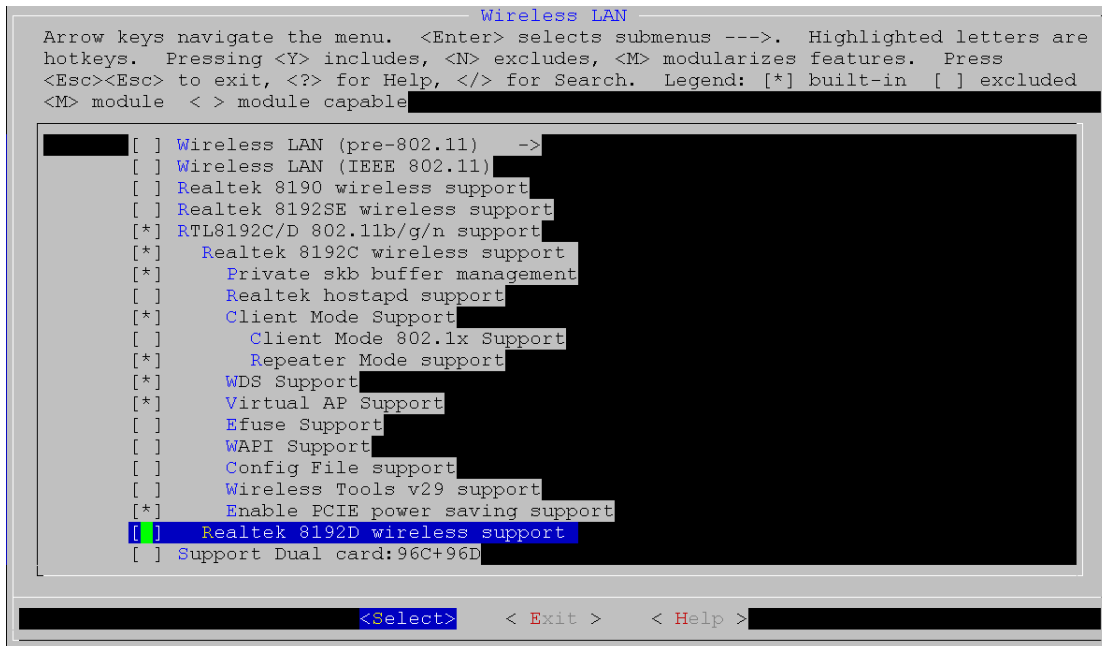
2) Enable 92D Driver

Config kernel --->

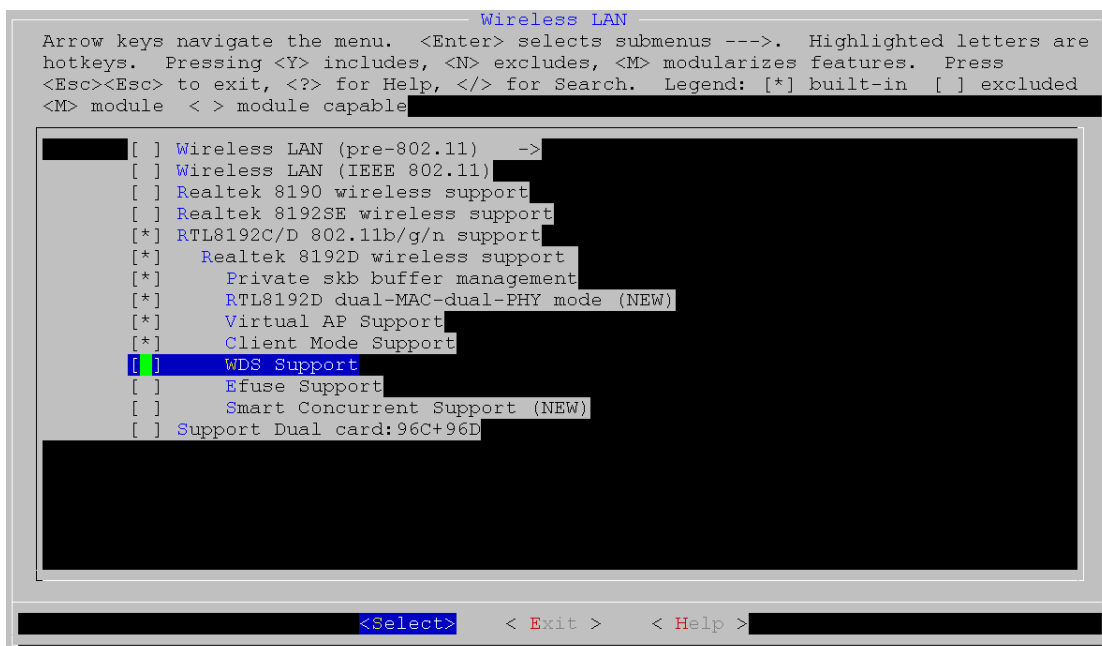
Device Drivers --->

Network device support --->

Wireless LAN --->



Selected “Realtek 8192D wireless support”



Cancel “WDS Support” item.

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

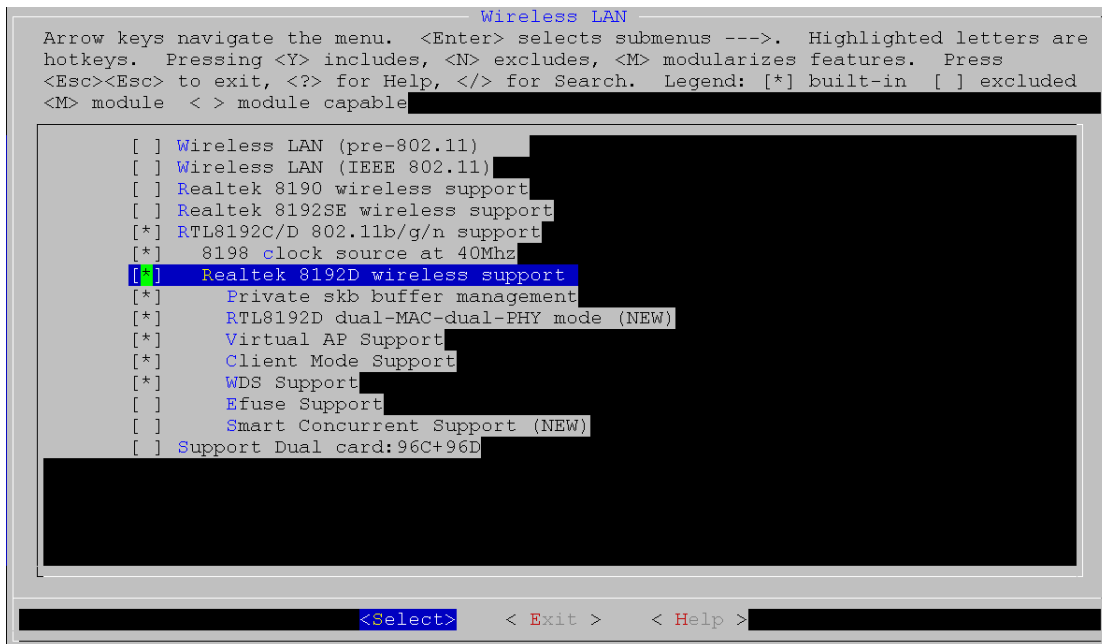
Config kernel --->

Device Drivers --->

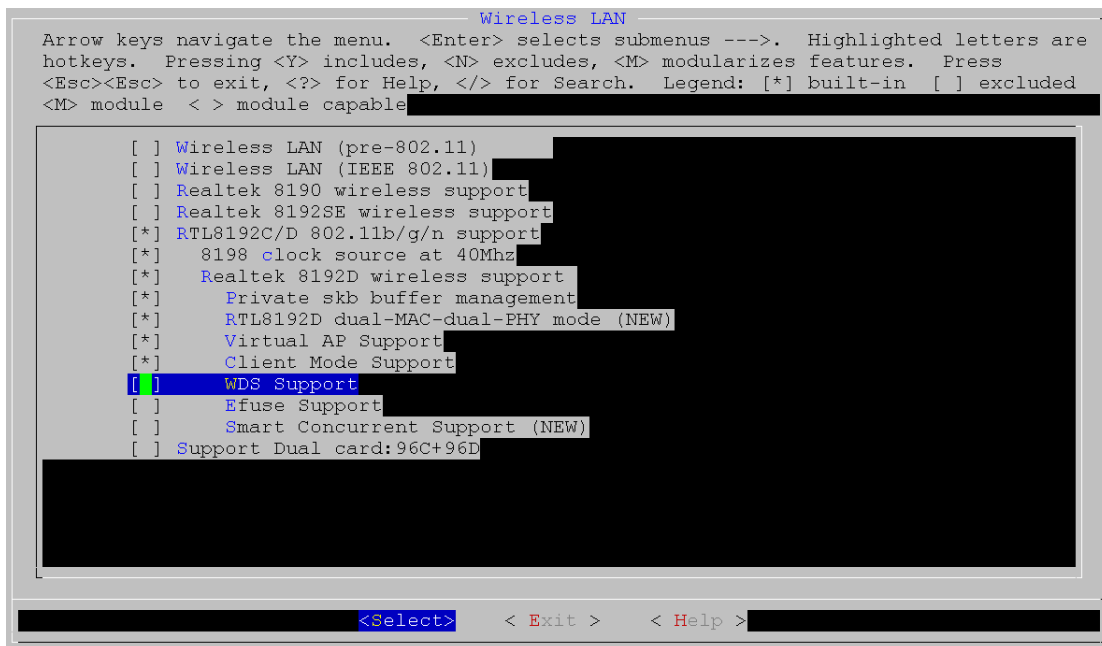
Network device support --->

Wireless LAN --->

Enable 92D:



Cancel Wds:



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

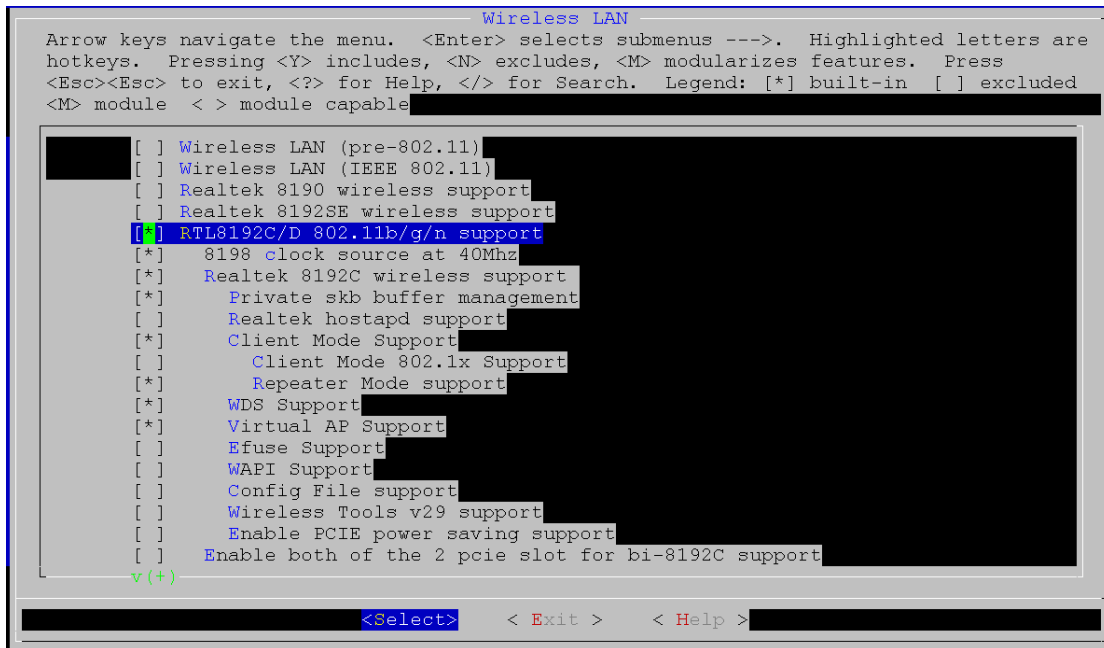
Config kernel --->

Device Drivers --->

Network device support --->

Wireless LAN --->

Enable 92C:



Save and make.

4.18.5 92C and 92D support for RTL8198 SDK

Enable 92D Driver and 92C Driver

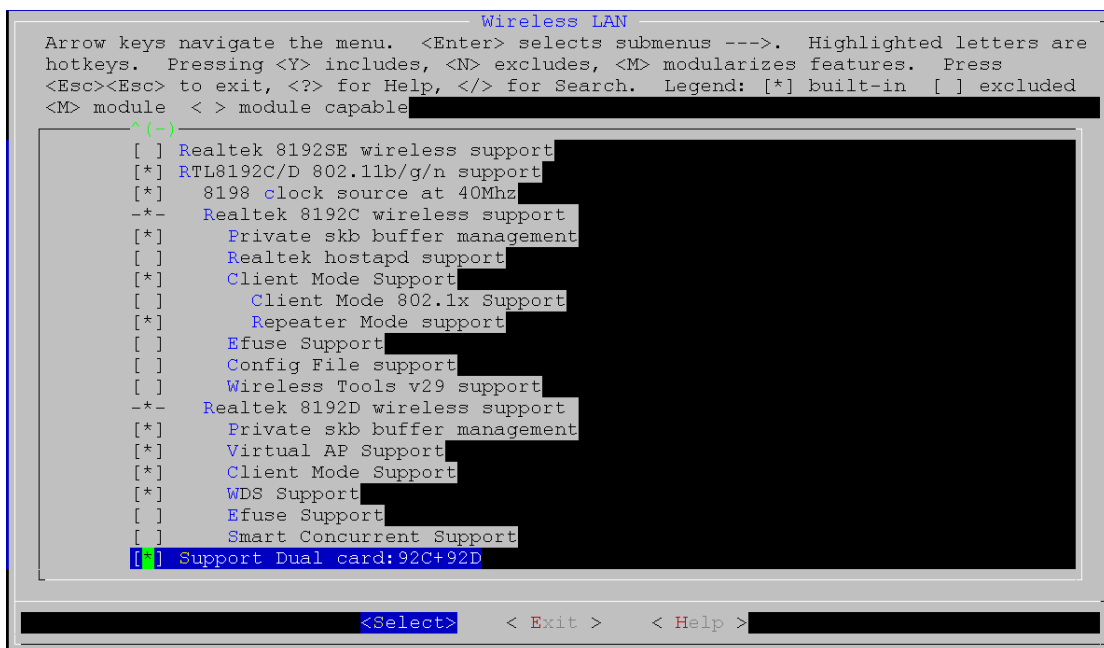
Config kernel --->

Device Drivers --->

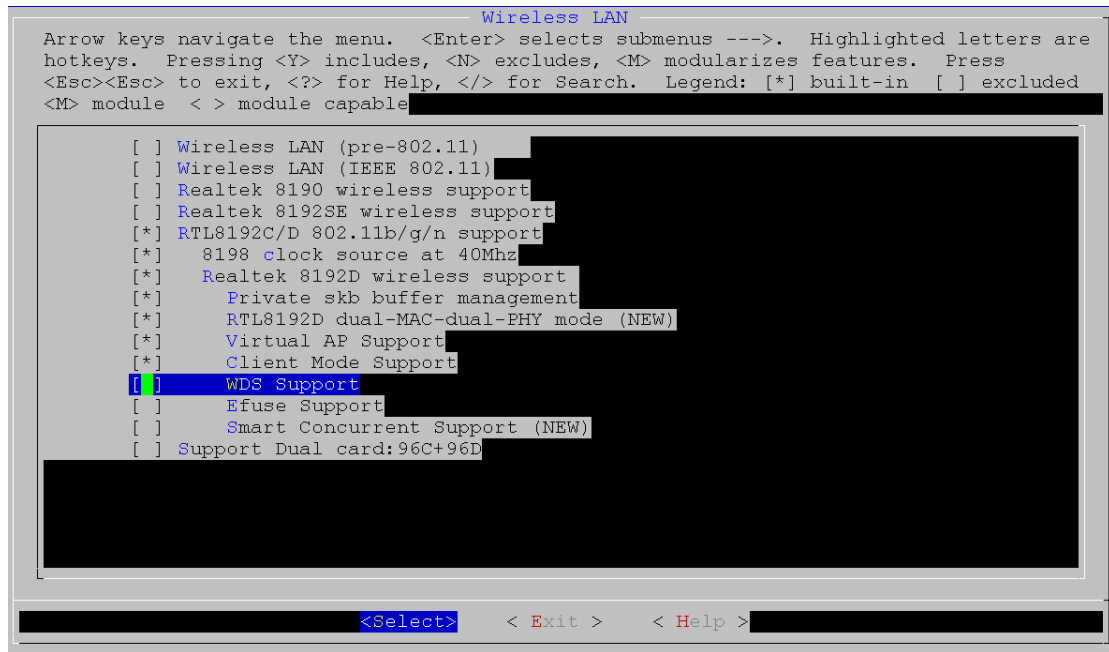
Network device support --->

Wireless LAN --->

Select "Support Dual card:92C+92D"



Cancel 92D Wds:



This configuration can support both 92Dvs and 92Dvc.

Save and make.

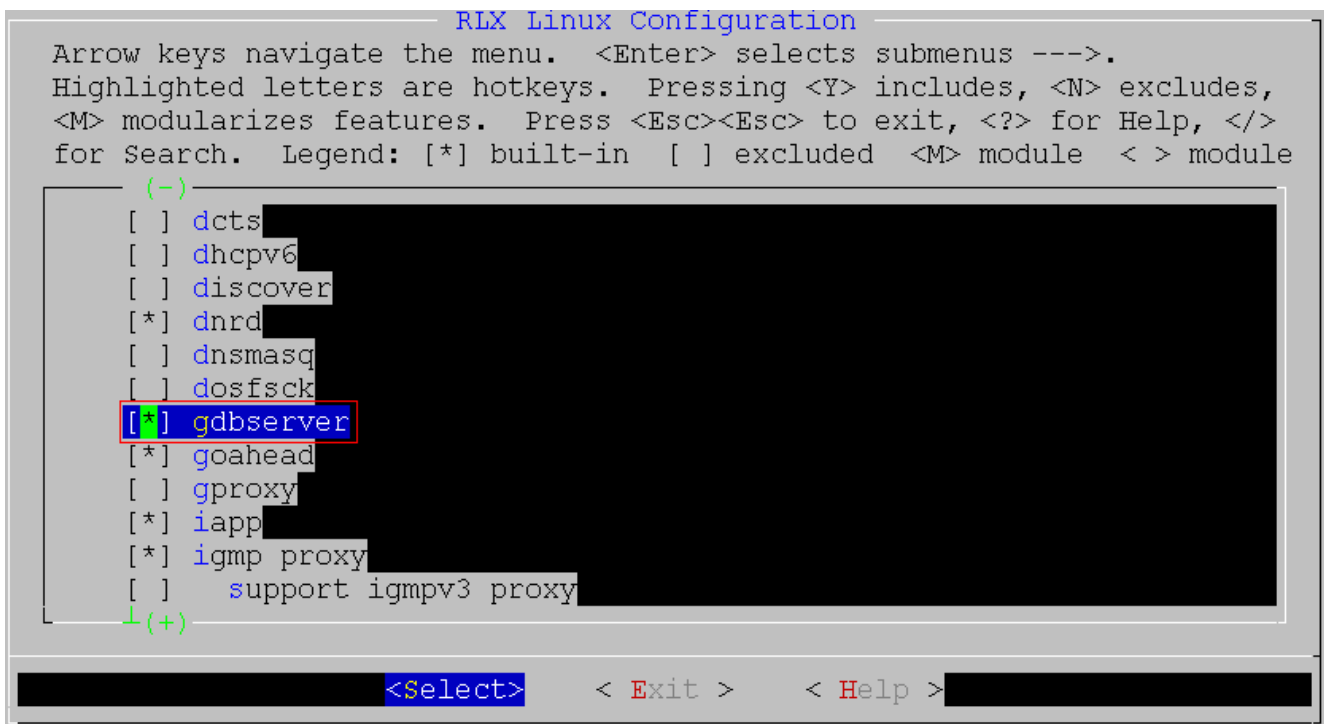
4.19 GDB server support

4.19.1 Enable GDB server

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

Menuconfig:

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



4.19.2 Use GDB server

4.19.2.1 Debugging an Already-running Process(over network)

1) remote step

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

2) host step

- ① export PATH=YOUR_PATH/rlxlinux-v2.2/users/gdb/gdb-host/bin/:\$PATH
- ② cd <source file dir of debugged program>
- ③ mips-linux-gdb <debugged program> /**enter gdb**/
- ④ 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

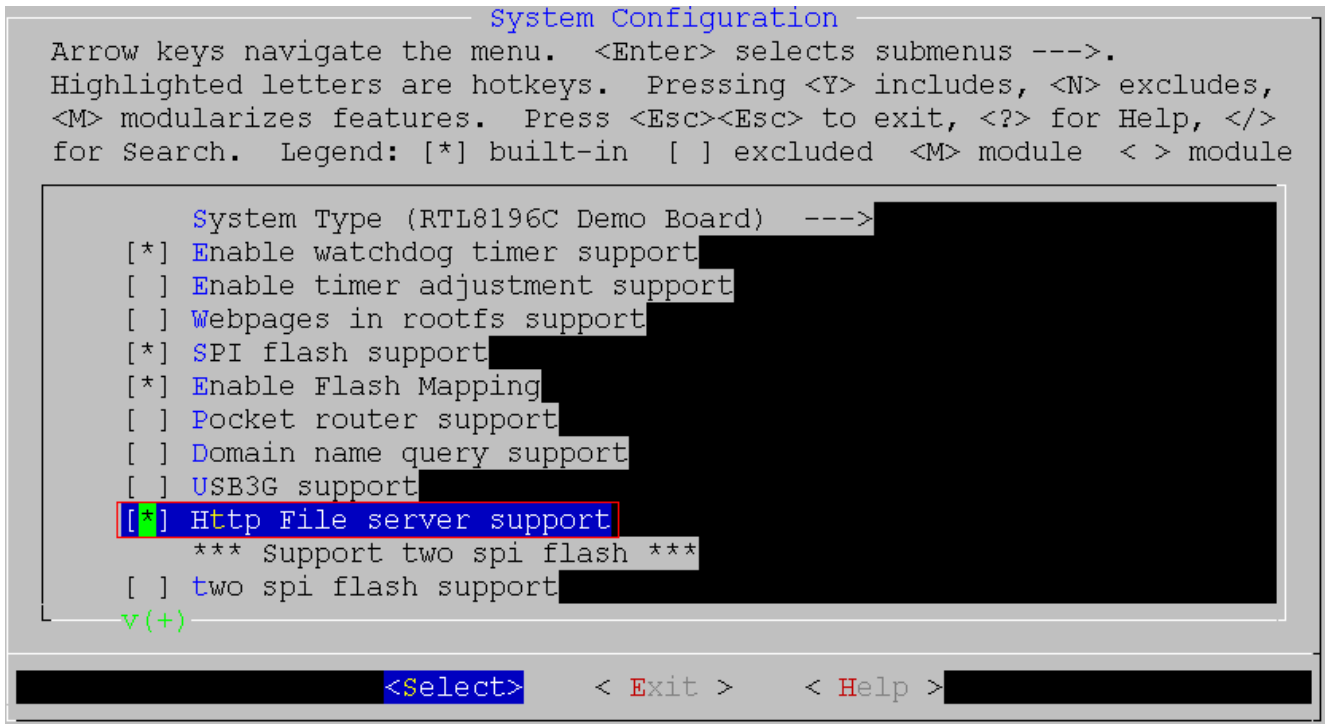
1) Kernel configure for HTTP File Server

```
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(+)
```

```
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)

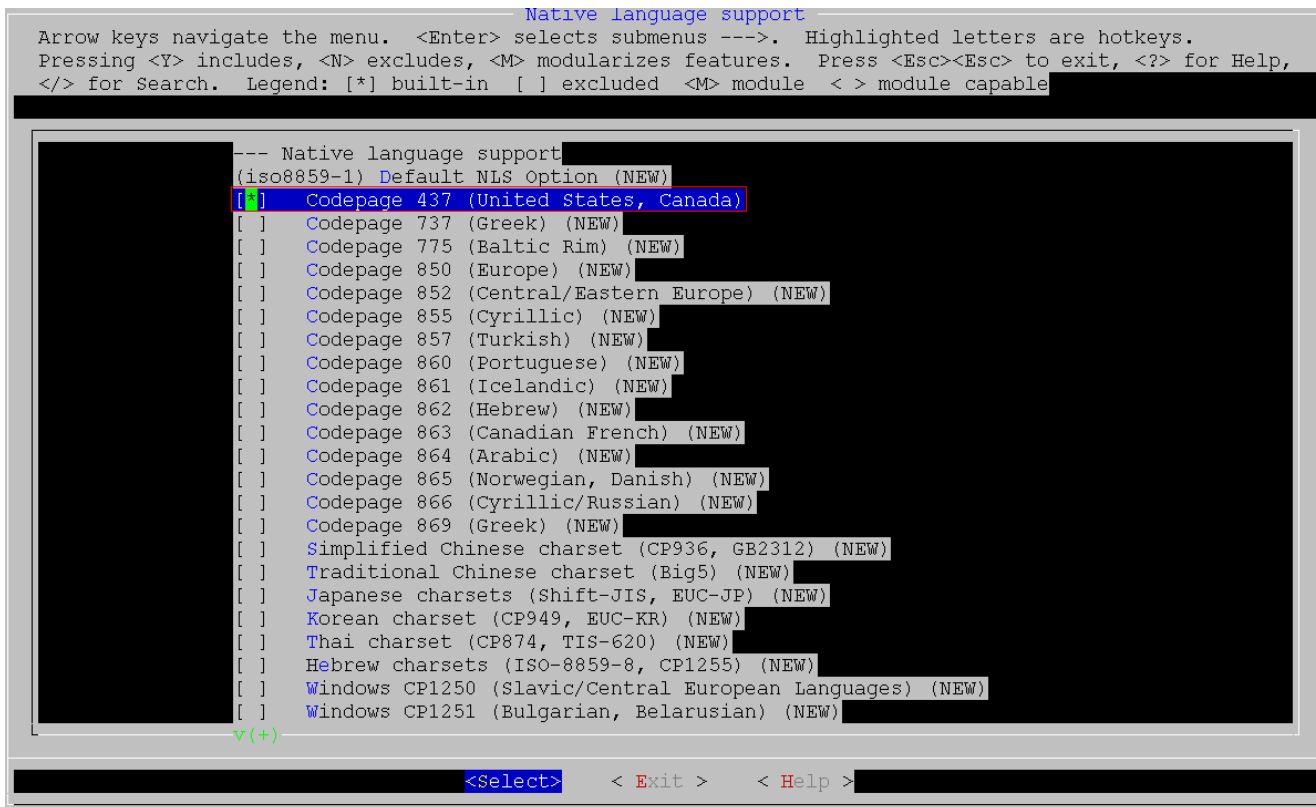
<Select> < Exit > < Help >

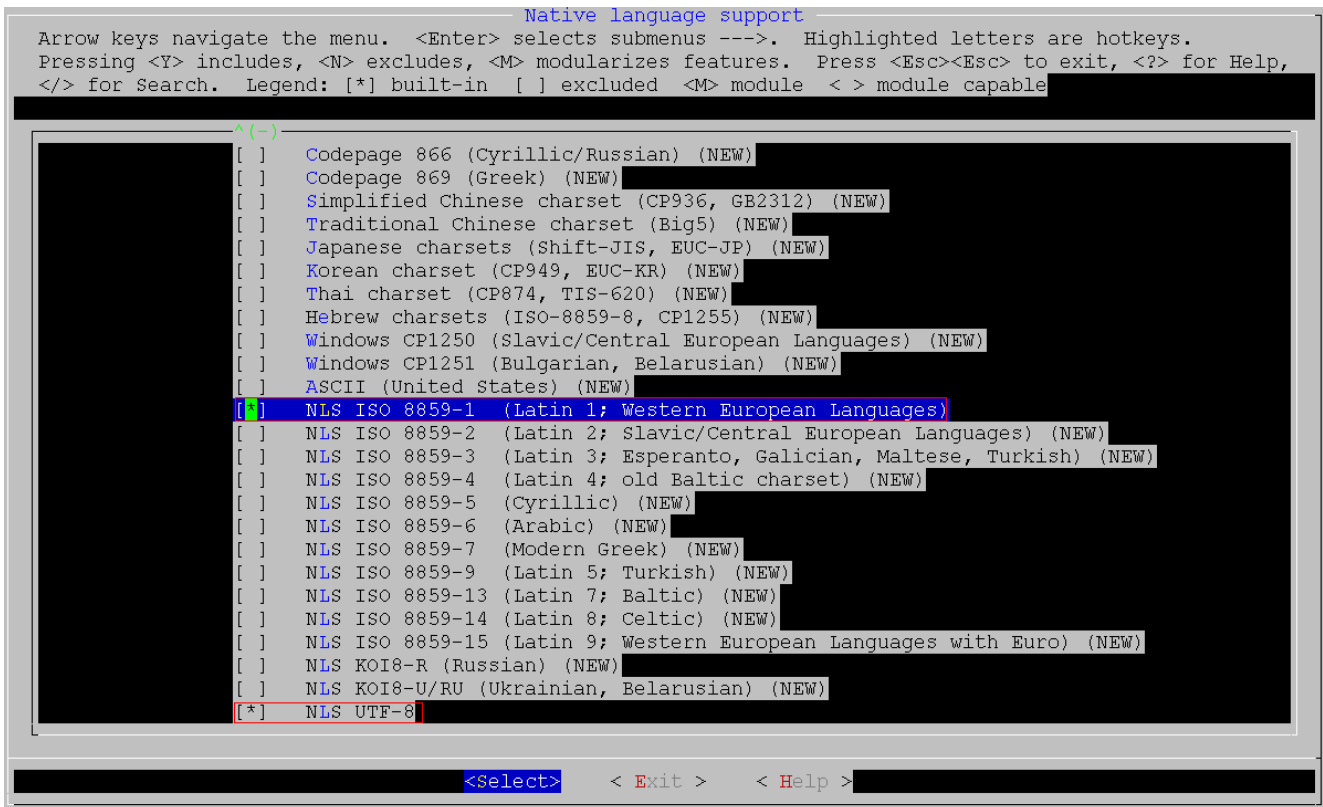
```

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





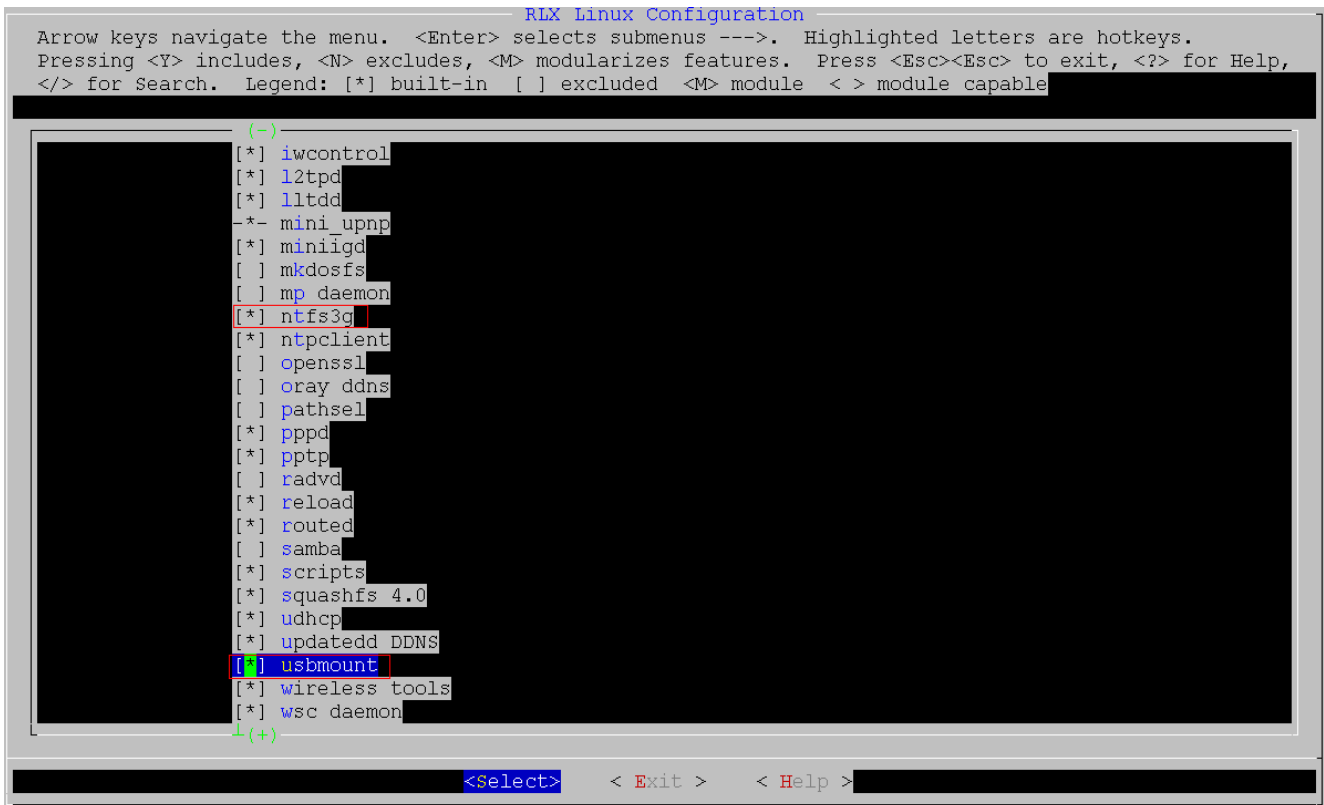
2) Application configure for HTTP File Server

make users_menuconfig // To configure application settings

Menuconfig:

[*] ntfs3g // selected

[*] usbmount // selected



4.20.3 WEB GUI Manual

Home page:

Realtek uWiFi

 [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:

 [sdal](#)

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
```

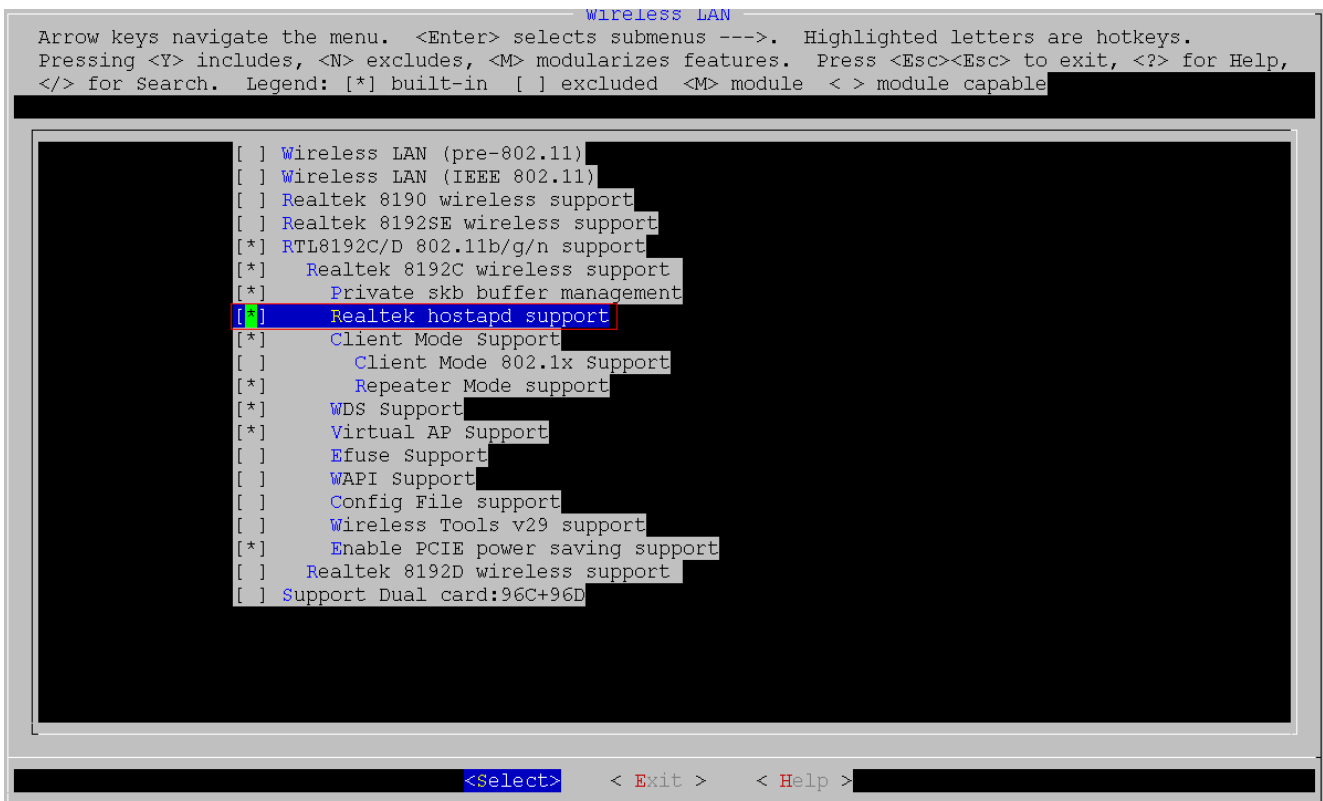
```
ifconfig wlan0 hw ether 00:44:55:66:98:81
```

```
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.



- 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.

4.23 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 (1lnPocket_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 >
```

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 conversion tool.

Usage:

```
cvcfg-gw [-r] [-no_hw] <input> <output>
```

Options:

- r: generate raw configuration 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 control and bit 1 for PPPoE pass through control.
- * 0: Disable both IPv6 and PPPoE pass through.

- * 1: Enable IPv6 pass through and disable PPPoE pass through.
- * 2: Enable PPPoE pass through and disable PPPoE 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: 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.

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 brcode/start.h and boot/init/utility.h, define BAUD_RATE as needed.

7.3 How to use vlan priority option of iptables?

Firstly, Linux kernel configure as follows:

- ① Networking support --->
 - Networking options --->
 - Network packet filtering framework (Netfilter) --->
 - Core Netfilter Configuration --->
 - "VLAN" match support //Selected
- ② Device Drivers --->
 - Network device support --->
 - Options for Realtek SoC --->
 - Support HW Qos //Selected
 - Config for Layered Driver Features --->
 - HW Qos support vlan priority //Selected

Secondly, one example of iptables rule using vlan priority option as follows:

```
// Mapping all the packets with vlan priority 3 to skb MARK 13
iptables -A PREROUTING -t mangle -m vlanpriority --prio-value 3 -j MARK --set-mark 13
```

7.4 Relationship of virtual AP and root AP?

Relationship of virtual AP and root AP is as follows:

- ① Since there is only one wireless IC, virtual AP and root AP use the same hardware.
- ② Boot sequence of our AP is that root AP is initialized firstly and virtual AP is initialized then.
- ③ Hardware settings of virtual AP such as band and channel should be subset of root AP hardware settings. Since our hardware support multiple data rate at the same, data rate of

virtual AP can be set different from root AP.

- ④ Since virtual AP is also the software AP of our AP, the software settings of virtual AP such as SSID, WMM, ACCESS, ENCRYPTION etc can be set different from root AP.

7.5 How to customize icon of lld2d?

Steps to customize icon of lld2d as follows:

- ① Add Icon, and install to /etc/
- ② Modify lld2d.conf, set "jumbo-icon" value, and install lld2d.conf to /etc/

One example to customize icon of lld2d as follows:

- ① Add Icon: realsil_gw.ico, Install to /etc/realsil_gw.ico
- ② Modify lld2d.conf: jumbo-icon = /etc/realsil_gw.ico

And install lld2d.conf to /etc/lld2d.conf

7.6 How to rename the AP when shown on Windows?

Please input command “ flash set DEVICE_NAME ‘name’ ” in console and then reboot AP.

One example as follows: If you want AP show the name ‘BARLX48’ on Windows, do command “flash set DEVICE_NAME BARLX48” in console, and then reboot AP.

7.7 How to add MAC address filter based on ether driver API?

1) Solution 1: Use ACL rule to do source or destination MAC address filter.

Program APIs as follows:

```
int32 rtl865x_add_acl(rtl865x_AclRule_t *rule, char *netifName,int32 priority);
int32 rtl865x_del_acl(rtl865x_AclRule_t *rule, char *netifName,int32 priority);
```

Example code for solution 1 as follows:

```
int32 retval;
rtl865x_AclRule_t rule;

// To add acl rule for smac(00:1e:c9:3b:b3:44) filter
bzero((void*)&rule,sizeof(rule));
rule.ruleType_ = RTL865X_ACL_MAC;
rule.actionType_ = RTL865X_ACL_DROP;
rule.pktOpApp_ = RTL865X_ACL_ALL_LAYER;
rule.srcMac_.octet[0]=0x00;
rule.srcMac_.octet[1]=0x1e;
```

```

rule.srcMac_.octet[2]=0xc9;
rule.srcMac_.octet[3]=0x3b;
rule.srcMac_.octet[4]=0xb3;
rule.srcMac_.octet[5]=0x44;

rule.srcMacMask_.octet[0]=0xFF;
rule.srcMacMask_.octet[1]=0xFF;
rule.srcMacMask_.octet[2]=0xFF;
rule.srcMacMask_.octet[3]=0xFF;
rule.srcMacMask_.octet[4]=0xFF;
rule.srcMacMask_.octet[5]=0xFF;

retval= rtl865x_add_acl(&rule, "br0", RTL865X_ACL_SYSTEM_USED);

// To del acl rule for smac(00:1e:c9:3b:b3:44) filter
bzero((void*)&rule, sizeof(rule));
rule.ruleType_ = RTL865X_ACL_MAC;
rule.actionType_ = RTL865X_ACL_DROP;
rule.pktOpApp_ = RTL865X_ACL_ALL_LAYER;
rule.srcMac_.octet[0]=0x00;
rule.srcMac_.octet[1]=0x1e;
rule.srcMac_.octet[2]=0xc9;
rule.srcMac_.octet[3]=0x3b;
rule.srcMac_.octet[4]=0xb3;
rule.srcMac_.octet[5]=0x44;

rule.srcMacMask_.octet[0]=0xFF;
rule.srcMacMask_.octet[1]=0xFF;
rule.srcMacMask_.octet[2]=0xFF;
rule.srcMacMask_.octet[3]=0xFF;
rule.srcMacMask_.octet[4]=0xFF;
rule.srcMacMask_.octet[5]=0xFF;

retval= rtl865x_del_acl(&rule, "br0", RTL865X_ACL_SYSTEM_USED);

```

2) Solution 2: Use source block via hardware L2 table to do source MAC address filter.

Program APIs as follows:

```
int32 rtl865x_addFilterDatabaseEntryExtension( uint16 fid, rtl865x_filterDbTableEntry_t * L2entry);
```

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

Example code for solution 2 as follows:

```
int32 retval;
uint32 fid;
rtl865x_filterDbTableEntry_t    l2temp_entry;
ether_addr_t macAddr;

// To add hardware L2 table entry for smac(00:1e:c9:3b:b3:44) filter
fid=RTL_LAN_FID;
memset((void *)&l2temp_entry, 0, sizeof(l2temp_entry));
l2temp_entry.l2type = RTL865x_L2_TYPEII;
l2temp_entry.process = FDB_TYPE_SRCBLK;
l2temp_entry.memberPortMask = RTL_LANPORT_MASK;
l2temp_entry.auth = FALSE;
l2temp_entry.SrcBlk = TRUE;

l2temp_entry.macAddr.octet[0]=0x00;
l2temp_entry.macAddr.octet[1]=0x1e;
l2temp_entry.macAddr.octet[2]=0xc9;
l2temp_entry.macAddr.octet[3]=0x3b;
l2temp_entry.macAddr.octet[4]=0xb3;
l2temp_entry.macAddr.octet[5]=0x44;

retval=rtl865x_addFilterDatabaseEntryExtension(fid, &l2temp_entry);

// To del hardware L2 table entry for smac(00:1e:c9:3b:b3:44) filter
fid=RTL_LAN_FID;

macAddr.octet[0]=0x00;
macAddr.octet[1]=0x1e;
macAddr.octet[2]=0xc9;
macAddr.octet[3]=0x3b;
macAddr.octet[4]=0xb3;
macAddr.octet[5]=0x44;

retval = rtl865x_delFilterDatabaseEntry(RTL865x_L2_TYPEII, fid, &macAddr);
```

7.8 How to adjust CPU speed?

For hardware designer, please refer to Fig 7.8.1 and adjust signal MA11~9 (CPUclkSel[2:0]) to select CPU clock from 250MHz, 270 MHz, 290 MHz, 310 MHz, 330 MHz, 350 MHz, 370 MHz and 390 MHz.

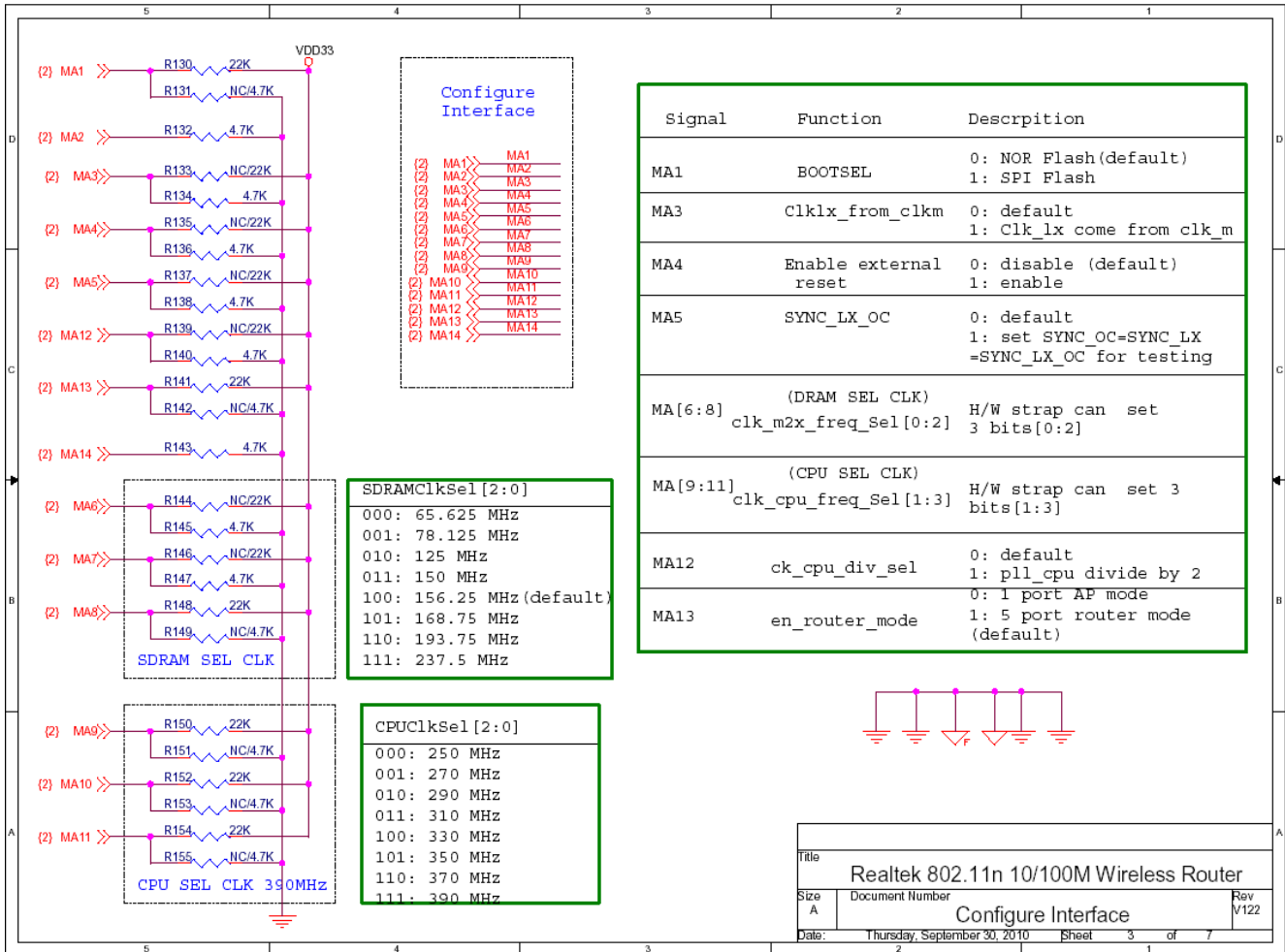


Fig 7.8.1 Realtek 802.11n 10/100M Wireless Router Interface Circuit

7.9 How to shutdown power of Ethernet ports?

1) For hardware designer, please refer to Fig 7.8.1 above and modify as follows:

Set MA13=0: port0~3 closed and only port4 opened.

Set MA13=1: port0~4 opened.

2) For software designer, “#define CONFIG_RTL_PHY_POWER_CTRL” in linux-2.6.30/include/net/rtl/rtl_nic.h, rebuild image and update it to AP.

After AP boots up, control the 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`