MTK-OpenWrt-3.10.14-SDK
Release Notes

Version:	3.3
Release date:	2016-01-04

Specifications are subject to change without notice.

Document Revision History

Revision	Date	Author	Description
3.0	2014.11.10	Hua Shao	OpenWrt 3.0, with up-to-date kernel & drivers from MTK/Ralink Linux SDK.
3.1	2014.11.27	Dragon Xiong	OpenWrt3.1, support 7628A,7688A, besides update to fix some issue base on 3.0
3.2	2015.03.09	Dragon Xiong	OpenWrt3.2, add support 7620A,7610E, besides update to fix some issue base on 3.1
3.3	2015.12.31	Ziqiang Yu	OpenWrt3.3 Release for MT7615

Table of Contents

1 Introduction

1.1 About OpenWrt

OpenWrt (http://www.openwrt.org/) is a linux distribution primarily used on embedded devices to route network traffic. The main components are the Linux kernel, uClibc, busybox, and OpenWrt framework utilities. All components have been optimized for size, to be small enough for fitting into the limited storage and memory available in the routers.

1.2 About this SDK

This SDK is a MTK customized OpenWrt project.

To provide better compatibility and better stability, some OpenWrt drivers were replaced with MTK drivers, such as Ethernet, USB, WiFi, SD Card, etc.

Brief Summary about this SDK:

- OpenWrt framework: Barrier Breaker
- Linux Kernel: 3.10.14
- Toolchain: toolchain-mipsel_24kec+dsp_gcc-4.8-linaro_uClibc-0.9.33.2
- MTK Linux SDK base: linux-3.10.14
- Supported SoC Chips: MT7620, MT7621, MT7628, MT7688
- Supported WiFi Chips: MT7603e, MT7602e, MT7612e, MT7628, MT7620, MT7610e

2 Change History

2.1 V3.0, 20141110

Feature:

- SoC chip support: MT7621, MT7620, MT7628
- WiFi chip support: MT7603e. MT7602e, MT7612e, MT7620, MT7610e, MT7628
- Ethernet driver Ready
- Flash driver ready
- PCI-e driver ready
- USB driver ready
- SDXC driver ready
- Ralink apps (8021xd, ated, btnd, gpio, nvram, mii_mgr, reg, switch, uci2dat, watchdog) ready
- Support luci-mtk web UI.

2.2 V3.1, 20141127

Feature:

- Add support for MT7628A,7688A SoC chip
- Add support for MT7628A, 7688A WiFi Chip
- Support hardware NAT for MT7621
- Support ntfs/extfat fs auto mount

Update:

- Update 7603 driver to fix WMM and RTS/CTS issue
- Update 7612e package to support SingleSKU
- Init 802.1xd when WPA/WPA2 enterprise encryption
- Fix some web issues

Note: Compile 7688a image, please select Subtarget (MT7628 based boards).

2.3 V3.2, 20150310

Feature:

- SoC chip support: MT7621, MT7620, MT7628, MT7688
- WiFi chip support: MT7603e. MT7602e, MT7612e, MT7620, MT7610e, MT7628
- Add source code for MTK APSoC U-boot
- Add 76x2e led support.

Update:

- WiFi: Bug fixes and performance tunning.
- Rewrite nvram and libnvram.
- Adjust default configuration to reduce CPU usage.
- Mt7620: support wifi hwnat.
- Update 7628 driver to lsdk V4.0.1.3_DPA_20150216

- Update Nand flash driver to P4 115775 for solving some nand flash issues
- Update 76x2e default config to improve througput and so on

2.4 V3.3, 20151231

Feature:

- SoC chip support: MT7621, MT7620, MT7628, MT7688
- WiFi chip support: MT7603e. MT7602e, MT7612e, MT7620, MT7610e, MT7628,MT7615
- Add mpstat for MT7621

.

Update:

- WiFi: fine tune performance on multi-core processor.
- Fine Tune samba performance

3 SDK Files

- ✓ **MTK-OpenWrt-3.10.14-SDK-Release Notes.docx**
  - This document.
- ✓ **mtksdk-openwrt-3.10.14-{version}-{date}-{tag}.tar.bz2**
  - SDK
- ✓ **openwrt-ramips-{chip-id}-squashfs-sysupgrade.bin**
  - Pre-build software

4 Build the SDK

4.1 Setup Build Environment

To build this SDK, you should have a linux server (linux 2.6.x or later) as the build host.
The default build will take up to 6 GB disk space. Make sure you have enough space to hold it.

Prepare the source project:

```
tar xjvf mtksdk-openwrt-3.10.14-{date}-{tag}.mini.tar.bz2 -C /path/to/your/workspace
```

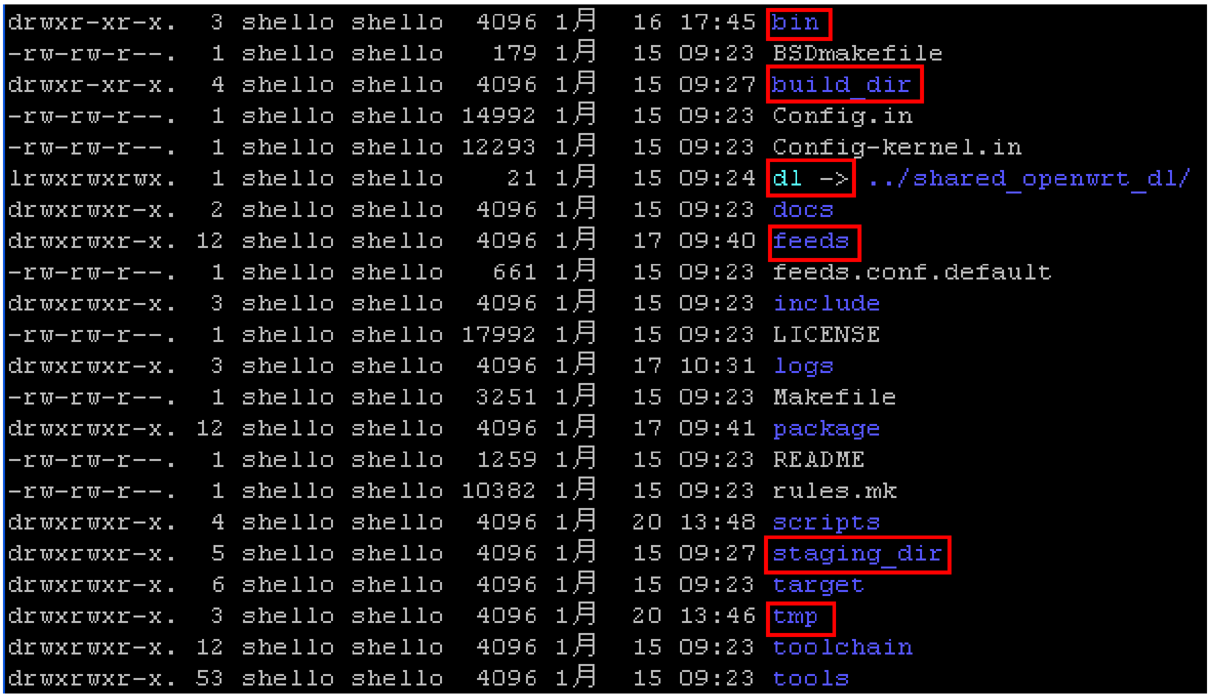
4.2 Check Build Dependency

In the first build, OpenWrt will check your build environment. If it complains that some library or software is missing, you should install them first, eg:

```
yum install svn
yum install wget
yum install ncurses-devel
yum install zlib-devel
…..
```

4.3 SDK root folder

This is what the SDK root folder looks like (Those folder names surrounded with red line are auto generated during build).



```
drwxr-xr-x.  3 shello shello  4096 1月  16 17:45 bin
-rw-rw-r--.  1 shello shello   179 1月  15 09:23 BSDmakefile
drwxr-xr-x.  4 shello shello  4096 1月  15 09:27 build_dir
-rw-rw-r--.  1 shello shello 14992 1月  15 09:23 Config.in
-rw-rw-r--.  1 shello shello 12293 1月  15 09:23 Config-kernel.in
lrwxrwxrwx.  1 shello shello    21 1月  15 09:24 dl -> ../shared_openwrt_dl/
drwxrwxr-x.  2 shello shello  4096 1月  15 09:23 docs
drwxrwxr-x. 12 shello shello  4096 1月  17 09:40 feeds
-rw-rw-r--.  1 shello shello   661 1月  15 09:23 feeds.conf.default
drwxrwxr-x.  3 shello shello  4096 1月  15 09:23 include
-rw-rw-r--.  1 shello shello 17992 1月  15 09:23 LICENSE
drwxrwxr-x.  3 shello shello  4096 1月  17 10:31 logs
-rw-rw-r--.  1 shello shello  3251 1月  15 09:23 Makefile
drwxrwxr-x. 12 shello shello  4096 1月  17 09:41 package
-rw-rw-r--.  1 shello shello  1259 1月  15 09:23 README
-rw-rw-r--.  1 shello shello 10382 1月  15 09:23 rules.mk
drwxrwxr-x.  4 shello shello  4096 1月  20 13:48 scripts
drwxrwxr-x.  5 shello shello  4096 1月  15 09:27 staging_dir
drwxrwxr-x.  6 shello shello  4096 1月  15 09:23 target
drwxrwxr-x.  3 shello shello  4096 1月  20 13:46 tmp
drwxrwxr-x. 12 shello shello  4096 1月  15 09:23 toolchain
drwxrwxr-x. 53 shello shello  4096 1月  15 09:23 tools
```

4.4 Config

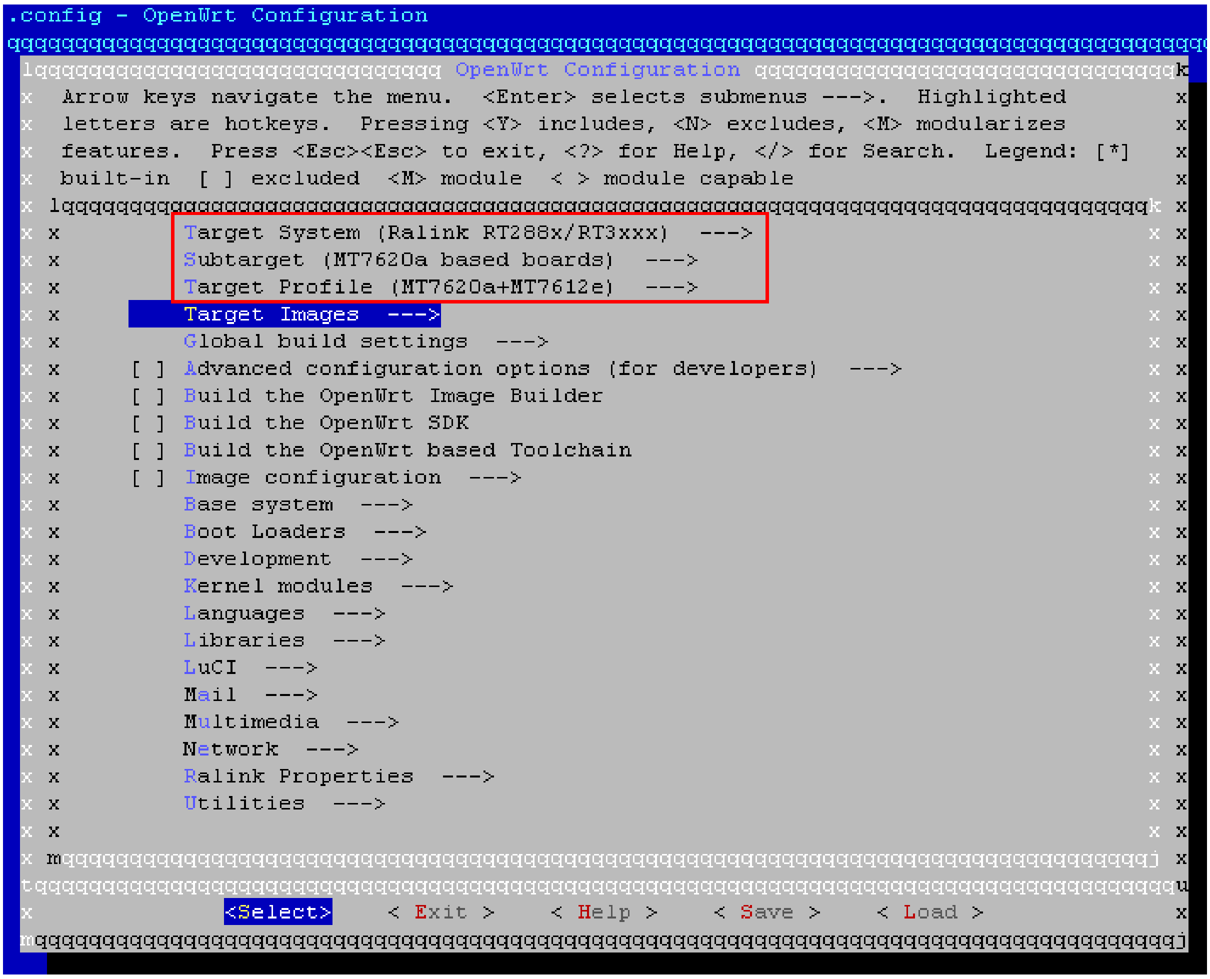
4.4.1 Config OpenWrt

Under SDK root folder, call:

```
make menuconfig
```

Then specify you configuration. For a default build, you need at least 3 items:

- Target System (Ralink Platform)
- Subtarget (Ralink SoC chip series)
- Target Profile (A specific model name)



```
.config - OpenWrt Configuration
qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq
 lqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq OpenWrt Configuration qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk
 x  Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted        x
 x  letters are hotkeys.  Pressing <Y> includes, <N> excludes, <M> modularizes        x
 x  features.  Press <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend: [*]  x
 x  built-in  [ ] excluded  <M> module  < > module capable                            x
 x lqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk x
 x x          Target System (Ralink RT288x/RT3xxx)  --->                            x x
 x x          Subtarget (MT7620a based boards)  --->                                x x
 x x          Target Profile (MT7620a+MT7612e)  --->                                x x
 x x          Target Images  --->                                                   x x
 x x          Global build settings  --->                                           x x
 x x      [ ] Advanced configuration options (for developers)  --->                 x x
 x x      [ ] Build the OpenWrt Image Builder                                       x x
 x x      [ ] Build the OpenWrt SDK                                                 x x
 x x      [ ] Build the OpenWrt based Toolchain                                     x x
 x x      [ ] Image configuration  --->                                             x x
 x x          Base system  --->                                                     x x
 x x          Boot Loaders  --->                                                    x x
 x x          Development  --->                                                     x x
 x x          Kernel modules  --->                                                  x x
 x x          Languages  --->                                                       x x
 x x          Libraries  --->                                                       x x
 x x          LuCI  --->                                                            x x
 x x          Mail  --->                                                            x x
 x x          Multimedia  --->                                                      x x
 x x          Network  --->                                                         x x
 x x          Ralink Properties  --->                                               x x
 x x          Utilities  --->                                                       x x
 x x                                                                                x x
 x mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj x
 tqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqu
 x              <Select>    < Exit >    < Help >    < Save >    < Load >               x
 mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj
```

After menuconfig done, you configuration will be saved in /SDK root/.config

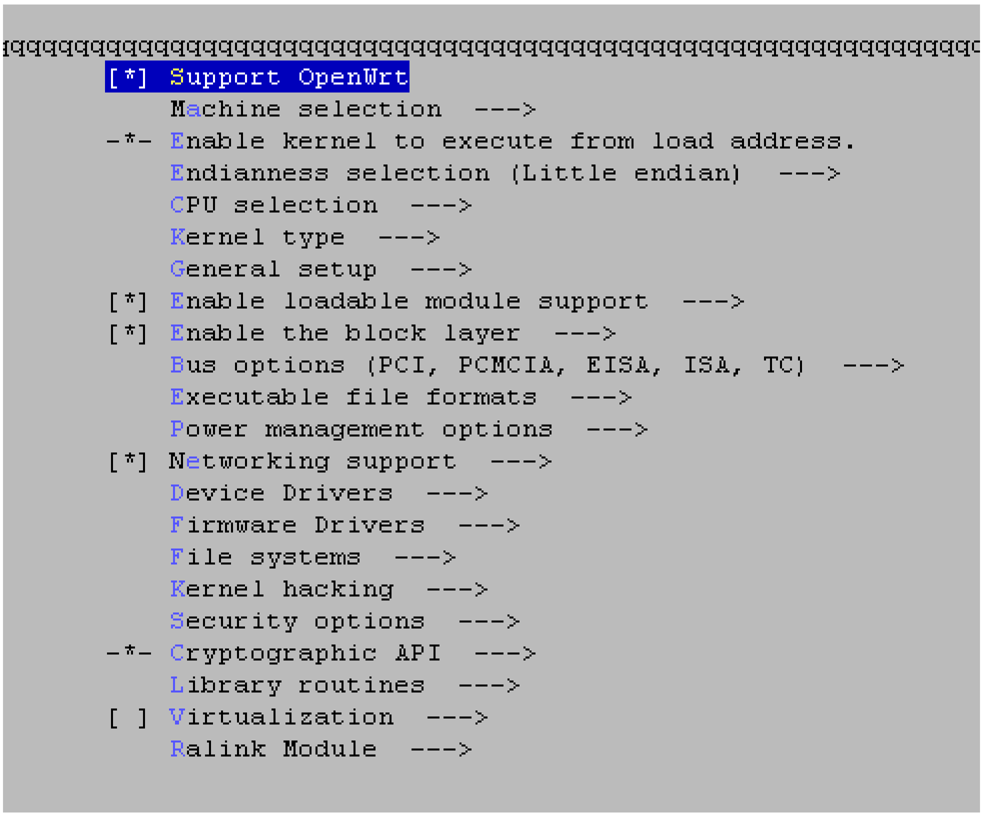
Note: In OpenWrt3.2, WiFi chips is not defined in profile. You can choose wifi drivers by yourself based on your HW.

4.4.2 Config Linux Kernel.

We provide default kernel configuration. you can find it at target/linux/ramips/mt76xx/config-3.10.14.
If that does not meet your needs, you can configure the kernel by yourself.
Under SDK root folder, call:

make kernel_menuconfig

Then you will see the classic kernel configuration menu like this:



```
qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq
      [*] Support OpenWrt
          Machine selection  --->
      -*- Enable kernel to execute from load address.
          Endianness selection (Little endian)  --->
          CPU selection  --->
          Kernel type  --->
          General setup  --->
      [*] Enable loadable module support  --->
      [*] Enable the block layer  --->
          Bus options (PCI, PCMCIA, EISA, ISA, TC)  --->
          Executable file formats  --->
          Power management options  --->
      [*] Networking support  --->
          Device Drivers  --->
          Firmware Drivers  --->
          File systems  --->
          Kernel hacking  --->
          Security options  --->
      -*- Cryptographic API  --->
          Library routines  --->
      [ ] Virtualization  --->
          Ralink Module  --->
```

4.5 Build

Under SDK root folder, call:

make

Or

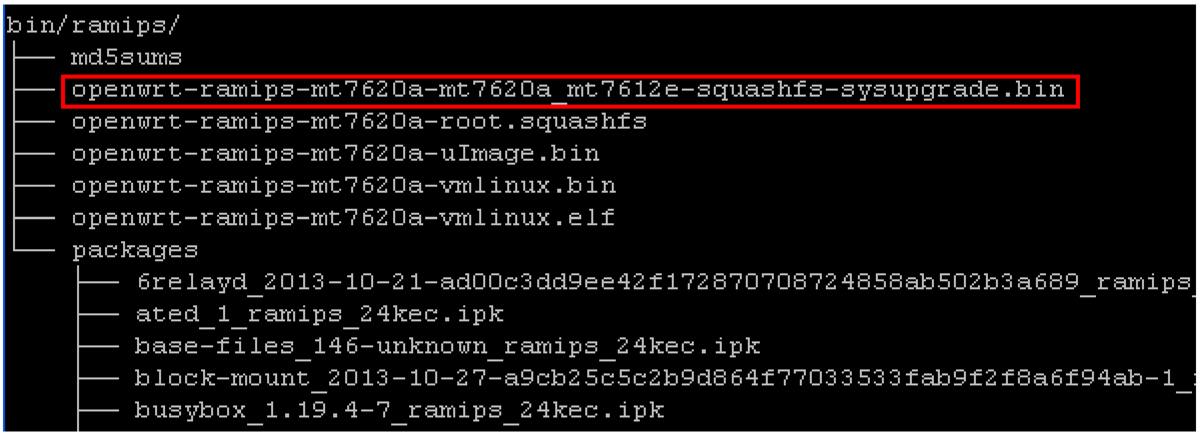
make V=s # this will produce verbose log

During build, the SDK will download many source packages from Internet. So, make sure your build host can access the open Internet.

The first build will take hours, please be patient. After first build, your build will be ready in minutes.

If anything goes wrong during building, use “make V=s” to see what happened.

If everything is OK, the target image will be generated under “bin/ramips”.



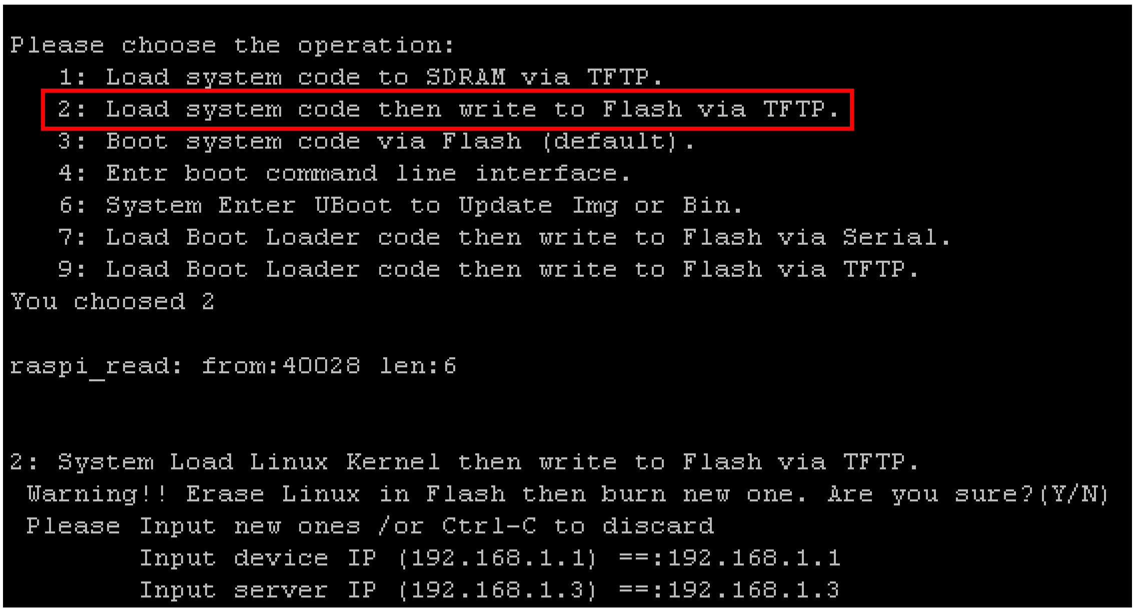
```
bin/ramips/
├── md5sums
├── openwrt-ramips-mt7620a-mt7620a_mt7612e-squashfs-sysupgrade.bin
├── openwrt-ramips-mt7620a-root.squashfs
├── openwrt-ramips-mt7620a-uImage.bin
├── openwrt-ramips-mt7620a-vmlinux.bin
├── openwrt-ramips-mt7620a-vmlinux.elf
└── packages
    ├── 6relayd_2013-10-21-ad00c3dd9ee42f172870708724858ab502b3a689_ramips_
    ├── ated_1_ramips_24kec.ipk
    ├── base-files_146-unknown_ramips_24kec.ipk
    ├── block-mount_2013-10-27-a9cb25c5c2b9d864f77033533fab9f2f8a6f94ab-1_
    ├── busybox_1.19.4-7_ramips_24kec.ipk
```

Note: 3.10.14 kernel should use MTK's kernel, not the original linux kernel.

4.6 Install Firmware

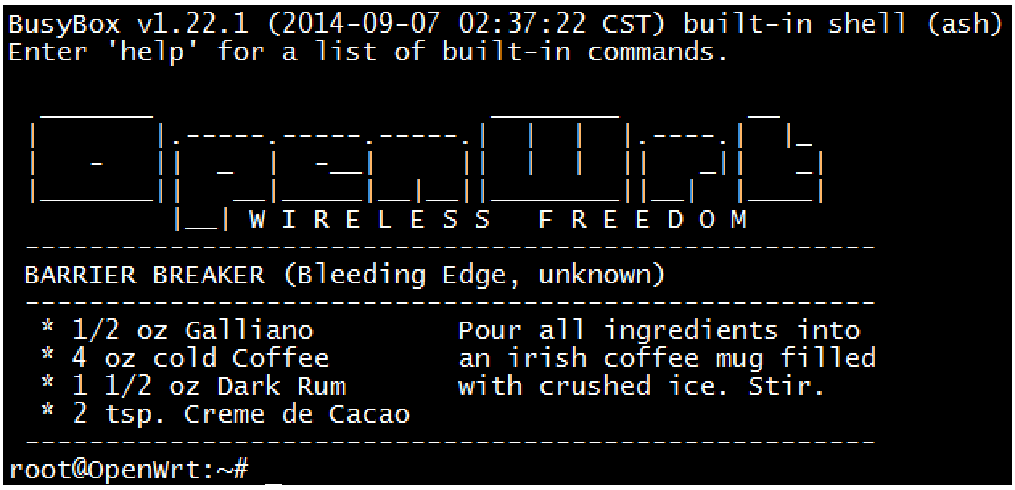
OpenWrt firmware can be flashed into the target board using MTK bootloader option 2.

Note: Option 1 won't work, because the image does not support initram mechanism.



```
Please choose the operation:
   1: Load system code to SDRAM via TFTP.
   2: Load system code then write to Flash via TFTP.
   3: Boot system code via Flash (default).
   4: Entr boot command line interface.
   6: System Enter UBoot to Update Img or Bin.
   7: Load Boot Loader code then write to Flash via Serial.
   9: Load Boot Loader code then write to Flash via TFTP.
You choosed 2

raspi_read: from:40028 len:6


2: System Load Linux Kernel then write to Flash via TFTP.
 Warning!! Erase Linux in Flash then burn new one. Are you sure?(Y/N)
 Please Input new ones /or Ctrl-C to discard
        Input device IP (192.168.1.1) ==:192.168.1.1
        Input server IP (192.168.1.3) ==:192.168.1.3
```

After system reboot, you will see OpenWrt running.



```
BusyBox v1.22.1 (2014-09-07 02:37:22 CST) built-in shell (ash)
Enter 'help' for a list of built-in commands.

  _______                     ________        __
 |       |.-----.-----.-----.|  |  |  |.----.|  |_
 |   -   ||  _  |  -__|     ||  |  |  ||   _||   _|
 |_______||   __|_____|__|__||________||__|  |____|
          |__| W I R E L E S S   F R E E D O M
 -----------------------------------------------------
 BARRIER BREAKER (Bleeding Edge, unknown)
 -----------------------------------------------------
  * 1/2 oz Galliano         Pour all ingredients into
  * 4 oz cold Coffee        an irish coffee mug filled
  * 1 1/2 oz Dark Rum       with crushed ice. Stir.
  * 2 tsp. Creme de Cacao
 -----------------------------------------------------
root@OpenWrt:~# _
```

5 Web Interface

OpenWrt does not build the web interface by default. Web interface is provided as a 3rd party package. Such as LuCI and XWRT.

5.1 LuCI

5.1.1 Install

Under SDK root folder, call:

```
scrips/feeds update -a
scripts/feeds install luci
```

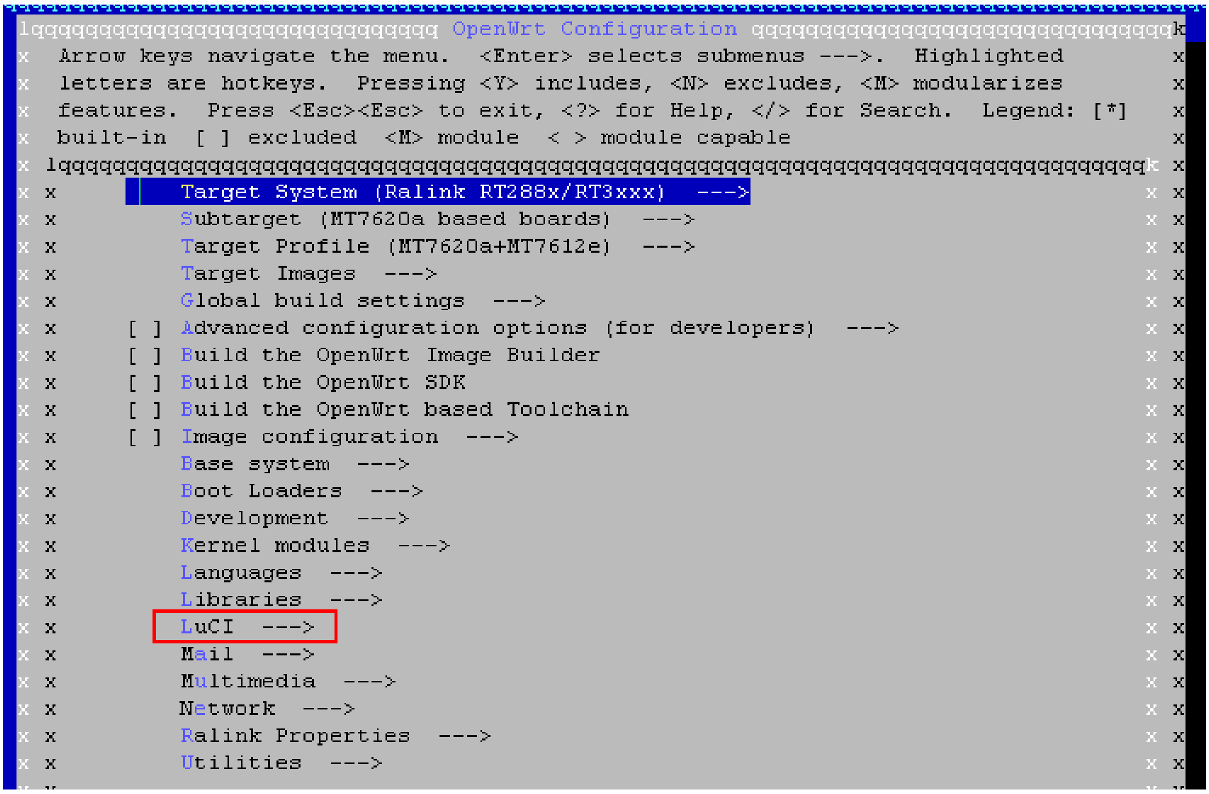
The LuCI package will be installed into SDK.

5.1.2 Config & Build

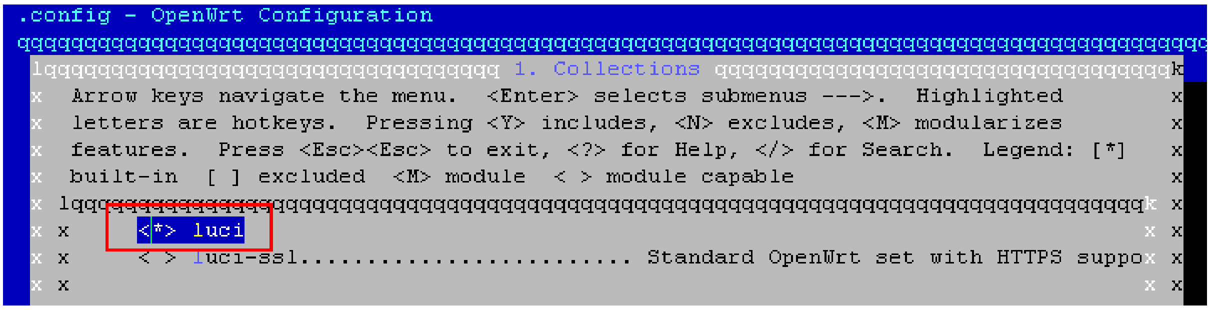
After installing LuCI, a submenu called “LuCI” will show up in “menuconfig”.
LuCI is not selected by default, choose “*” in “LuCI”->”Collection”->”luci" to enable LuCI by default.Then:

make V=s

You will see that LuCI get build along with the SDK.



```
lqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq OpenWrt Configuration qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk
x  Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted      x
x  letters are hotkeys.  Pressing <Y> includes, <N> excludes, <M> modularizes      x
x  features.  Press <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend: [*]  x
x  built-in  [ ] excluded  <M> module  < > module capable                          x
x lqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk x
x x          Target System (Ralink RT288x/RT3xxx)  --->                           x x
x x          Subtarget (MT7620a based boards)  --->                               x x
x x          Target Profile (MT7620a+MT7612e)  --->                               x x
x x          Target Images  --->                                                  x x
x x          Global build settings  --->                                          x x
x x      [ ] Advanced configuration options (for developers)  --->                x x
x x      [ ] Build the OpenWrt Image Builder                                      x x
x x      [ ] Build the OpenWrt SDK                                                x x
x x      [ ] Build the OpenWrt based Toolchain                                    x x
x x      [ ] Image configuration  --->                                            x x
x x          Base system  --->                                                    x x
x x          Boot Loaders  --->                                                   x x
x x          Development  --->                                                    x x
x x          Kernel modules  --->                                                 x x
x x          Languages  --->                                                      x x
x x          Libraries  --->                                                      x x
x x          LuCI  --->                                                           x x
x x          Mail  --->                                                           x x
x x          Multimedia  --->                                                     x x
x x          Network  --->                                                        x x
x x          Ralink Properties  --->                                              x x
x x          Utilities  --->                                                      x x
```



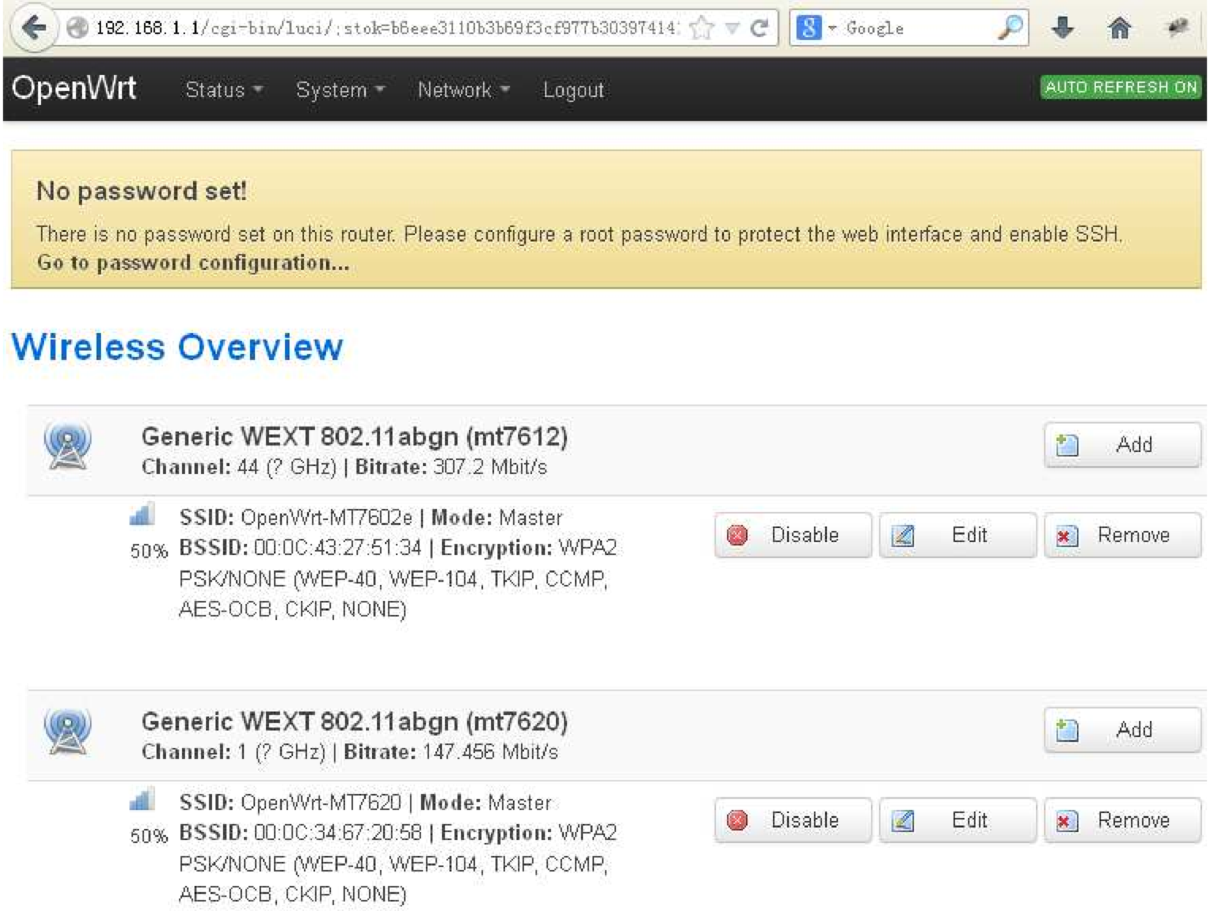
```
.config - OpenWrt Configuration
qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq
 lqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq 1. Collections qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk
 x  Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted        x
 x  letters are hotkeys.  Pressing <Y> includes, <N> excludes, <M> modularizes        x
 x  features.  Press <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend: [*]    x
 x  built-in  [ ] excluded  <M> module  < > module capable                             x
 x lqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk x
 x x      <*> luci                                                                 x x
 x x      < > luci-ssl......................... Standard OpenWrt set with HTTPS suppo x x
 x x                                                                               x x
```

5.1.3 Access

By default, You can access the web interface at http://192.168.1.1/.

Account = “admin”

Password = “admin”



192.168.1.1/cgi-bin/luci/;stok=b6eee3110b3b69f3cf977b30397414
Google
OpenWrt
Status
System
Network
Logout
AUTO REFRESH ON

No password set!

There is no password set on this router. Please configure a root password to protect the web interface and enable SSH.
Go to password configuration...

Wireless Overview

Generic WEXT 802.11abgn (mt7612)
Channel: 44 (? GHz) | Bitrate: 307.2 Mbit/s

50% SSID: OpenWrt-MT7602e | Mode: Master
BSSID: 00:0C:43:27:51:34 | Encryption: WPA2 PSK/NONE (WEP-40, WEP-104, TKIP, CCMP, AES-OCB, CKIP, NONE)

Generic WEXT 802.11abgn (mt7620)
Channel: 1 (? GHz) | Bitrate: 147.456 Mbit/s

50% SSID: OpenWrt-MT7620 | Mode: Master
BSSID: 00:0C:34:67:20:58 | Encryption: WPA2 PSK/NONE (WEP-40, WEP-104, TKIP, CCMP, AES-OCB, CKIP, NONE)

5.2 LuCI-mtk

5.2.1 Install

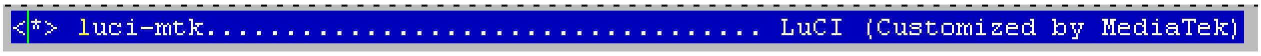
We provide a customized LuCI UI, called luci-mtk.
To install luci-mtk, you should remove the official luci ui first.
If you haven’t installed official luci yet, you can skip this step.

```
make package/luci/clean
scripts/feeds uninstall luci
rm -rf tmp
rm .config*
```

Then,

```
make menuconfig
```

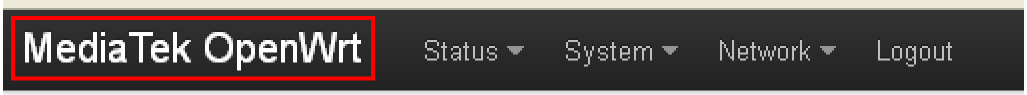
In “ Ralink Properties” ->” UI” ->” luci-mtk”, check it as “*”.



```
<*> luci-mtk................................. LuCI (Customized by MediaTek)
```

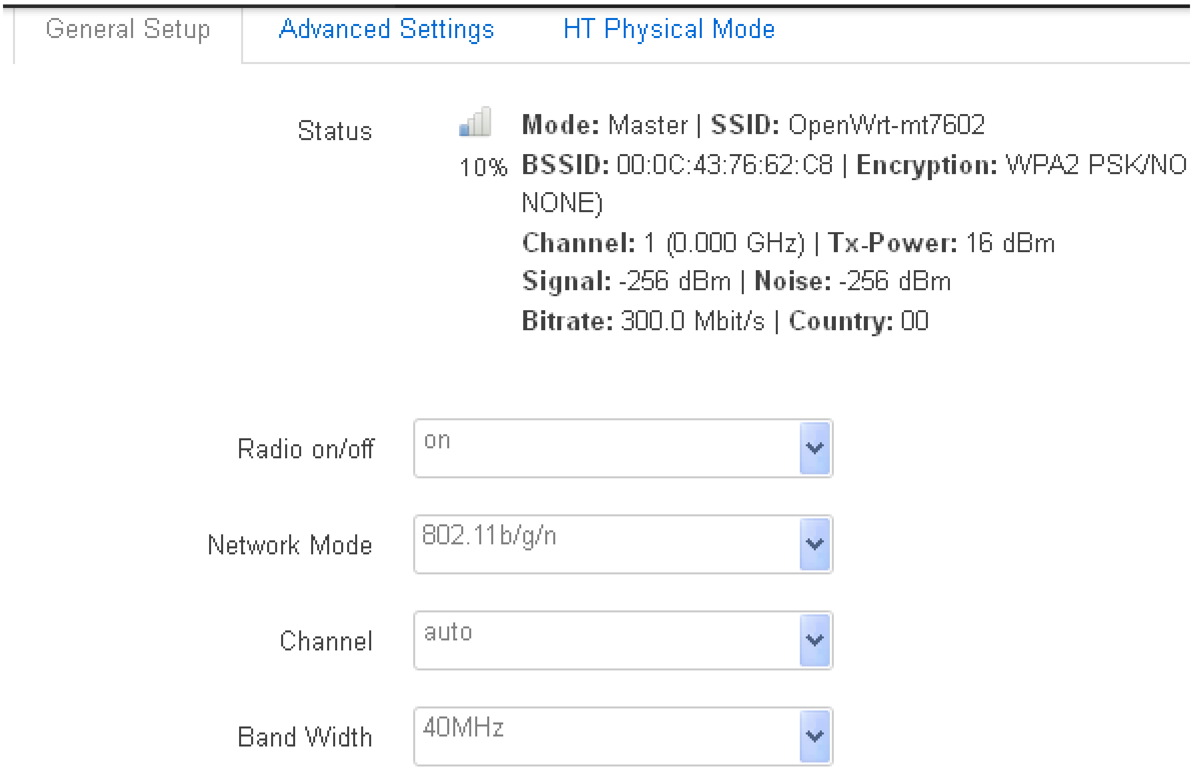
Then luci-mtk will be installed into your image.

We changed the OpenWrt title to make sure you installed the right luci ui.



5.2.2 Configure wifi via luci-mtk

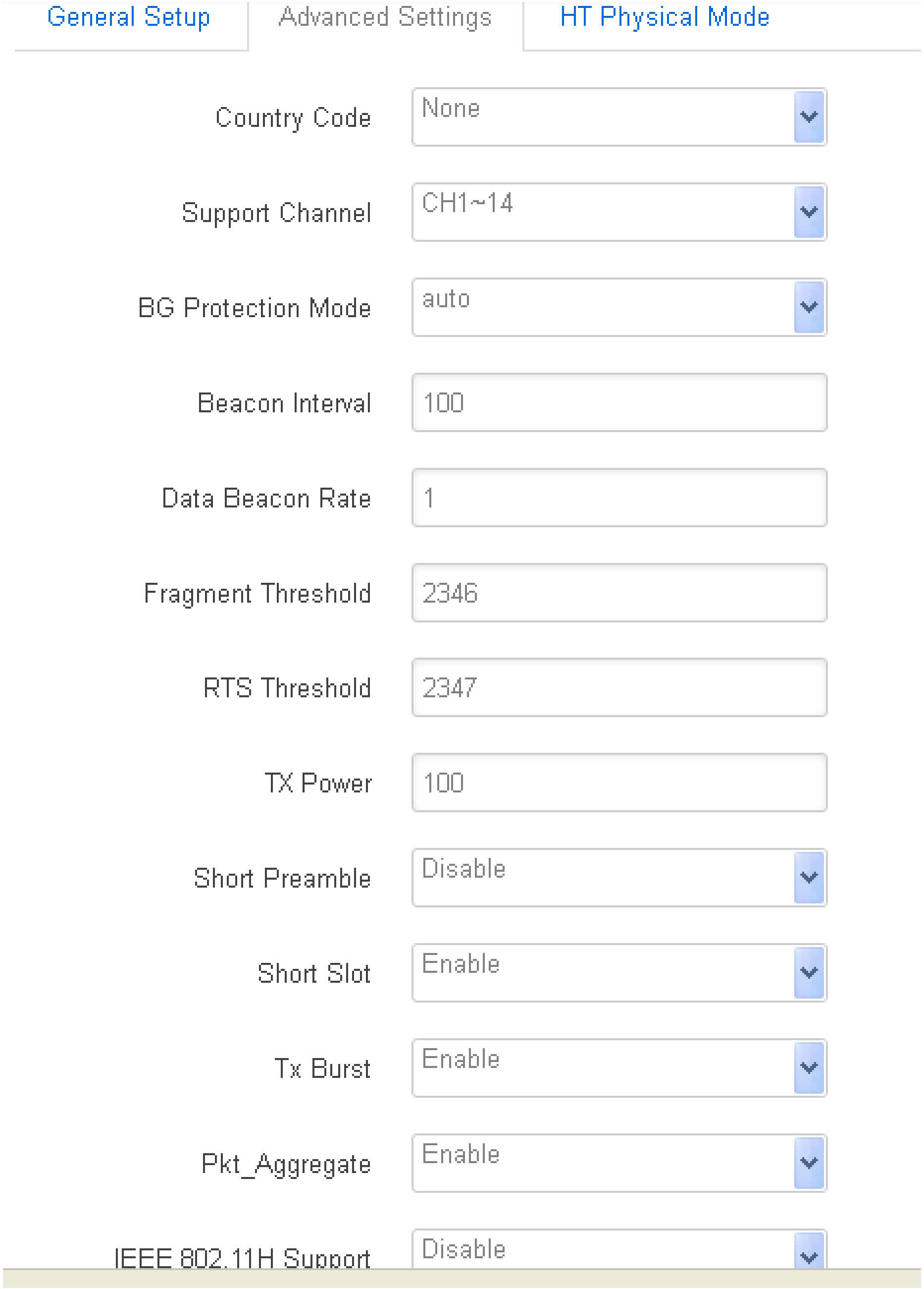
General setup



Status 10% **Mode:** Master | **SSID:** OpenWrt-mt7602
BSSID: 00:0C:43:76:62:C8 | **Encryption:** WPA2 PSK/NO NONE)
Channel: 1 (0.000 GHz) | **Tx-Power:** 16 dBm
Signal: -256 dBm | **Noise:** -256 dBm
Bitrate: 300.0 Mbit/s | **Country:** 00

Radio on/off	on
Network Mode	802.11b/g/n
Channel	auto
Band Width	40MHz

Advanced Settings



Country Code: None

Support Channel: CH1~14

BG Protection Mode: auto

Beacon Interval: 100

Data Beacon Rate: 1

Fragment Threshold: 2346

RTS Threshold: 2347

TX Power: 100

Short Preamble: Disable

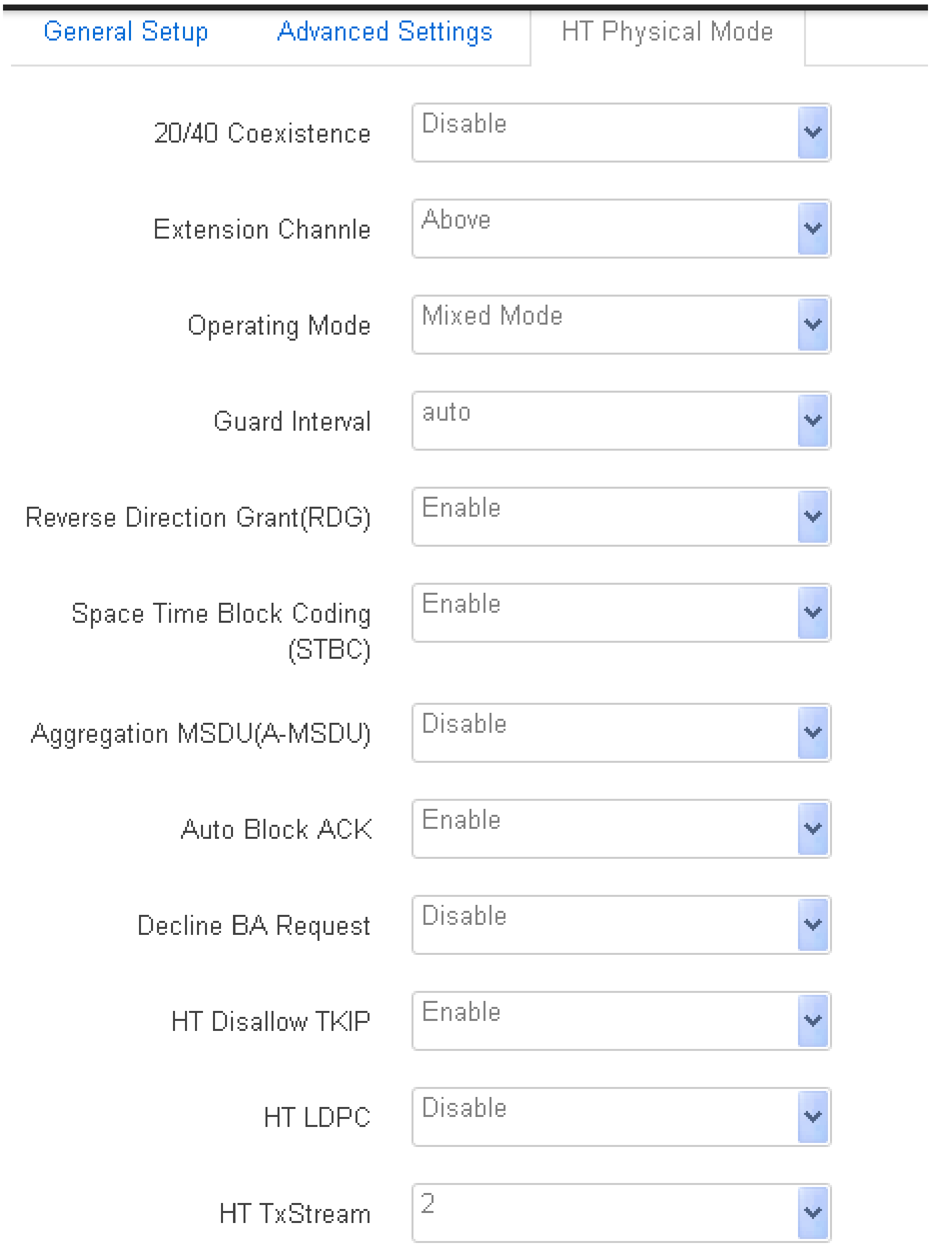
Short Slot: Enable

Tx Burst: Enable

Pkt_Aggregate: Enable

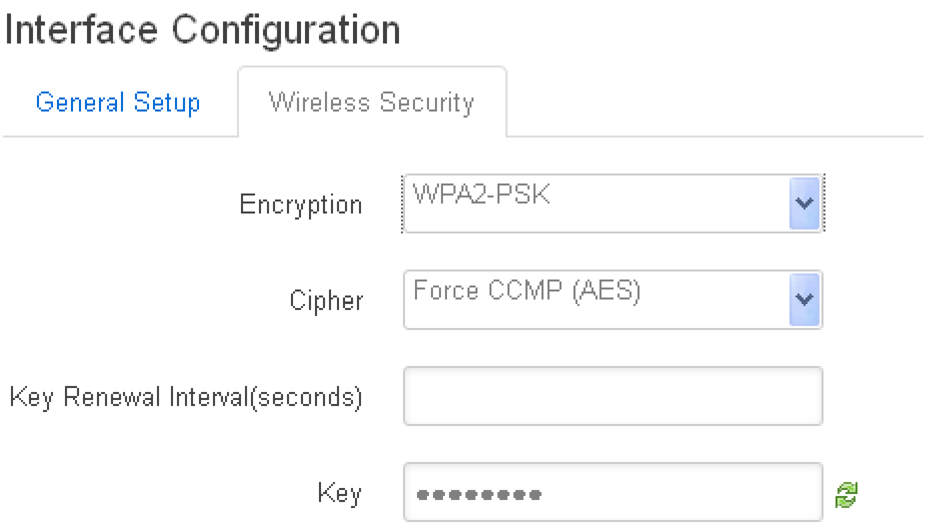
IEEE 802.11H Support: Disable

HT Physical Mode



20/40 Coexistence	Disable
Extension Channle	Above
Operating Mode	Mixed Mode
Guard Interval	auto
Reverse Direction Grant(RDG)	Enable
Space Time Block Coding (STBC)	Enable
Aggregation MSDU(A-MSDU)	Disable
Auto Block ACK	Enable
Decline BA Request	Disable
HT Disallow TKIP	Enable
HT LDPC	Disable
HT TxStream	2

Security settings



Interface Configuration

Encryption: WPA2-PSK

Cipher: Force CCMP (AES)

Key Renewal Interval(seconds):

Key: ●●●●●●●●●

1 Wireless configuration via UCI

1.1 Basic idea

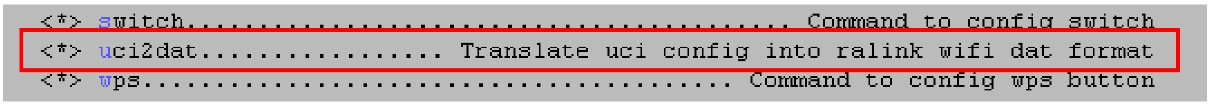
There are already 2 ways to configure MTK wireless drivers.

1) Using iwpriv command. (eg: *iwpriv ra0 set SSID=myrouter*)
2) Edit the profile of the driver. (located at */etc/Wireless/chipname/chipname.dat*)

Here we introduce the 3rd way, which is for the convenience of LuCI development.

To use this feature, you should enable **uci2dat** first. You can find this application at:

Menuconfig -> Ralink Properties -> Applications -> uci2dat



```
<*> switch..................................... Command to config switch
<*> uci2dat................. Translate uci config into ralink wifi dat format
<*> wps......................................... Command to config wps button
```

Then you can configure MTK wireless drivers via uci commands, like this:

```
uci set wireless.chipname.option1=value1
uci set wireless.chipname.option2=value2
……
uci commit
wifi down
wifi up
```

1.2 Examples

In your script, you should replace “chipname” with the right name of your wireless chip, (like mt7620, mt7612, mt7602, mt7610 etc).

1.2.1 SSID

This is a little tricky, read the example carefully.

```
uci set wireless.@wifi-iface[n].ssid=newssid
```

N is the index of the interface you want to change. You can check the interface index by :

```
uci show wireless
```

You may see something like this:

```
[root@OpenWrt]uci show wireless
wireless.mt7612=wifi-device
wireless.mt7612.type=mt7612
wireless.mt7612.vendor=ralink
wireless.mt7612.channel=0
wireless.mt7612.autoch=2
wireless.@wifi-iface[0]=wifi-iface
wireless.@wifi-iface[0].device=mt7612
wireless.@wifi-iface[0].ifname=rai0
wireless.@wifi-iface[0].network=lan
wireless.@wifi-iface[0].mode=ap
wireless.@wifi-iface[0].ssid=OpenWrt-mt7612
wireless.@wifi-iface[0].encryption=psk2
wireless.@wifi-iface[0].key=12345678
wireless.mt7620=wifi-device
wireless.mt7620.type=mt7620
wireless.mt7620.vendor=ralink
wireless.mt7620.channel=0
wireless.mt7620.autoch=2
wireless.@wifi-iface[1]=wifi-iface
wireless.@wifi-iface[1].device=mt7620
wireless.@wifi-iface[1].ifname=ra0
wireless.@wifi-iface[1].network=lan
wireless.@wifi-iface[1].mode=ap
wireless.@wifi-iface[1].ssid=OpenWrt-mt7620
wireless.@wifi-iface[1].encryption=psk2
wireless.@wifi-iface[1].key=12345678
```

Then you have 2 WiFi interfaces, one has SSID “OpenWrt-mt7620” and the other has SSID “OpenWrt-mt7612”.

If you want to change “OpenWrt-mt7620” to “MyNew7620”, you should call:

```
uci set wireless.@wifi-iface[1].ssid=MyNew7620 # “1” is the index of “OpenWrt-mt7620”
```

1.2.2 Encryption

Read the example in “SSID” section first, then you can:

```
uci set wireless.@wifi-iface[n].encryption=x
```

x could be:

x	encryption
psk-mixed	WPAPSKWPA2PSK
psk2	WPA2PSK
psk	WPAPSK
wpa-mixed	WPAWPA2
wpa2	WPA2

wpa	WPA
wep-open	OPEN
wep-shared	SHARED
open	OPEN

1.2.3 Key

Read the example in “SSID” section first, then you can:

```
uci set wireless.@wifi-iface[n].key=x
```

1.2.4 Add new SSID (When multi-SSID is enabled)

Read the example in “SSID” section first, then you can:

```
uci add wireless wifi-iface
uci set wireless.@wifi-iface[n].device=chipname
uci set wireless.@wifi-iface[n].ifname=ra1
uci set wireless.@wifi-iface[n].network=lan
uci set wireless.@wifi-iface[n].mode=ap
uci set wireless.@wifi-iface[n].ssid=newssid
uci set wireless.@wifi-iface[n].encryption=psk2
uci set wireless.@wifi-iface[n].key=11111111
```

N is the new index, which is current index increased by 1.

1.2.5 Remove SSID

Read the example in “SSID” section first, then you can:

```
uci del wireless.@wifi-iface[n]
```

N is the index which you want to remove.

1.2.6 Wireless Mode

```
uci set wireless.chipname.wifimode=n
```

n could be:

0: Legacy 11b/g mixed
1: Legacy 11B only
2: Legacy 11A only
3: Legacy 11a/b/g mixed
4: Legacy 11G only
5: 11ABGN mixed
6: 11N only
7: 11GN mixed
8: 11AN mixed
9: 11BGN mixed
10: 11AGN mixed
11: 11N only in 5G band only

14: 11A/AN/AC mixed 5G band only (Only 11AC chipset support)
15:11 AN/AC mixed 5G band only (Only 11AC chipset support)

1.2.7 Radio On/Off

```
uci set wireless.chipname.radio=n
```

n could be:

0: Disable
1: Enable

1.2.8 Channel

1) Fixed channel:

```
uci set wireless.chipname.autoch=0
uci set wireless.chipname.channel=n   # n is the channel number.
```

2) Auto channel:

```
uci set wireless.chipname.channel=0
uci set wireless.chipname.autoch=1 # or 2, different algorithm.
```

3) Auto channel with channels skipped

```
uci set wireless.chipname.channel=0
uci set wireless.chipname.autoch=1 # or 2, different algorithm.
uci set wireless.chipname.autoch_skip=”1;2;3;4”
```

1.2.9 Operating Mode

```
uci set wireless.chipname.ht_opmode=n
```

n could be:
0: Mixed Mode
1: Green Mode

1.2.10 Channel Band Width

```
uci set wireless.chipname.bw=n
```

n could be:
0: 20MHz
1: 40MHz
2: 80MHz (5G only)

1.2.11 Guard Interval

```
uci set wireless.chipname.ht_gi=n          # 2.4G
```

n could be:
0: long GI
1: Auto GI

```
uci set wireless.chipname.vht_sgi=n       # 5G
```

n could be:

0: Disable
1: Enable

1.2.12 MCS

```
uci set wireless.chipname.ht_mcs=0;....;1;0;
```

bitwise data, bit index equals to the MCS index.

1.2.13 Reverse Direction Grant

```
uci set wireless.chipname.ht_rdg=n
```

n could be:
0: Disable
1: Enable

1.2.14 Space Time Block Coding (STBC)

```
uci set wireless.chipname.ht_stbc=n #2.4G
```

n could be:
0: Disable
1: Enable

```
uci set wireless.chipname.vht_stbc=n #5G
```

n could be:
0: Disable
1: Enable

1.2.15 Aggregation MSDU (A-MSDU)

```
uci set wireless.chipname.ht_amsdu=n
```

n could be:
0: Disable
1: Enable

1.2.16 AP Power Saving (APSDCapable)

```
uci set wireless.chipname.apsd=n
```

n could be:
0: Disable
1: Enable

1.2.17 WMM

```
uci set wireless.chipname.wmm=n
```

n could be:
0: Disable
1: Enable

1.2.18 Decline BA Request

```
uci set wireless.chipname.ht_badec=n
```

n could be:

0: Disable

1: Enable

1.2.19 HT LDPC

```
uci set wireless.chipname.ht_ldpc=n  # 2.4G
```

n could be:

0: Disable

1: Enable

```
uci set wireless.chipname.vht_ldpc=n  # 5G
```

n could be:

0: Disable

1: Enable

1.2.20 BG Protection Mode

```
uci set wireless.chipname.bgprotect=n
```

n could be:

0: Auto

1: On

2: Off

1.2.21 VHT BW Signaling

```
uci set wireless.chipname.vht_bw_sig=n
```

n could be:

0: Disable

1: Static

2: Dynamic

1.2.22 Beacon Interval

```
uci set wireless.chipname.beacon=n
```

n could be: 20~999

1.2.23 Data Beacon Rate (DTIM)

```
uci set wireless.chipname.dtim=n
```

n could be: 1~255

1.2.24 Fragment Threshold

```
uci set wireless.chipname.fragthres=n
```

n could be: 255~2346

1.2.25 RTS Threshold

```
uci set wireless.chipname.rtsthres=n
```

n could be: 1~2347

1.2.26 TX Power

```
uci set wireless.chipname.txpower=n
```

n could be: 1~100

1.2.27 Short Preamble

```
uci set wireless.chipname.txpreamble=n
```

n could be:
0: Disable
1: Enable

1.2.28 Short Slot

```
uci set wireless.chipname.shortslot=n
```

n could be:
0: Disable
1: Enable

1.2.29 Tx Burst

```
uci set wireless.chipname.txburst=n
```

n could be:
0: Disable
1: Enable

1.2.30 Pkt_Aggregate

```
uci set wireless.chipname.pktaggre=n
```

n could be:
0: Disable
1: Enable

1.2.31 IEEE 802.11H Support

```
uci set wireless.chipname.pktaggre=n
```

n could be:
0: Disable
1: Enable

1.2.32 Country Code

```
uci set wireless.chipname.country=str
```

str could be: **”US”, ”JP” ,”FR” ,”TW”, ”IE” ,”HK” ,”NONE”**

1.2.33 2.4G Country Region

```
uci set wireless.chipname.region=n
```

Region	Channels
0	1-11
1	1-13
2	10-11
3	10-13
4	14
5	1-14
6	3-9
7	5-13
31	1-14
32	1-11 active scan, 12 and 13 passive scan
33	1-14 all active scan, 14 b mode only

1.2.34 5G Country Region

```
uci set wireless.chipname.aregion=n
```

Region	Channels
0	36, 40, 44, 48, 52, 56, 60, 64, 149, 153, 157, 161, 165
1	36, 40, 44, 48, 52, 56, 60, 64, 100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140
2	36, 40, 44, 48, 52, 56, 60, 64
3	52, 56, 60, 64, 149, 153, 157, 161
4	149, 153, 157, 161, 165
5	149, 153, 157, 161
6	36, 40, 44, 48
7	36, 40, 44, 48, 52, 56, 60, 64, 100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140, 149, 153, 157, 161, 165
8	52, 56, 60, 64
9	36, 40, 44, 48, 52, 56, 60, 64, 100, 104, 108, 112, 116, 132, 136, 140, 149, 153, 157, 161, 165
10	36, 40, 44, 48, 149, 153, 157, 161, 165
11	36, 40, 44, 48, 52, 56, 60, 64, 100, 104, 108, 112, 116, 120, 149, 153, 157, 161

1.2.35 E2pAccessMode for MT7615

```
uci set wireless. chipname. e2paccmode=1 /2
```

1:effuse mode,2 flash mode.

For 2.4G :

```
uci set wireless.mt7615e2. e2paccmode=1
uci commit wireless
/etc/init.d/network restart
```

For 5G :

```
uci set wireless.mt7615e5. e2paccmode=1
uci commit wireless
```

```
/etc/init.d/network restart
```

1.2.36 TxBF For MT7615

```
uci set wireless.chipname. txbf =n
n=3 iBF &eBF
n=2 eBF
n=1 iBF
n=0 No BF
```

2 MTK/Ralink Property Packages

Here are packages located under **{SDKRoot}/package/ralink**.
Here is a brief introduction to them.

2.1 Applications

2.1.1 ated

ATE daemon. For factory test.
You should enable the "ATE/QA" option in the driver configuration. (Check "make menuocnfig" -> "ralink properties" -> "drivers" -> "mtxxxx").
Usage:

```
ated -i ifname          # run as a daemon
```

2.1.2 btnd

A daemon program to handle GPIO button event.
Once it detects a button event (click, hold), it call corresponding shell scripts defined under /etc/btnd/ .
Usage:

```
btnd <button-name> <gpio-number> &          # run as a daemon
```

Assume that your product has 2 gpio buttons, one for "WiFi WPS", uses gpio 2, the other for "restore factory settings", uses gpio 1. Then you can:

```
btnd wps 2 &   // "wps" is a given name chosen by you.
btnd reset 1 & // just make sure the given name matches the script name under /etc/btnd/.
```

When the user clicked wps button (gpio 2), btnd would call "sh /etc/btnd/wps_click.sh".
When the user held reset button (gpio 1), btnd would call "sh /etc/btnd/reset_hold.sh".
You can put your own task in <btn-name>_hold.sh or <btn-name>_click.sh.

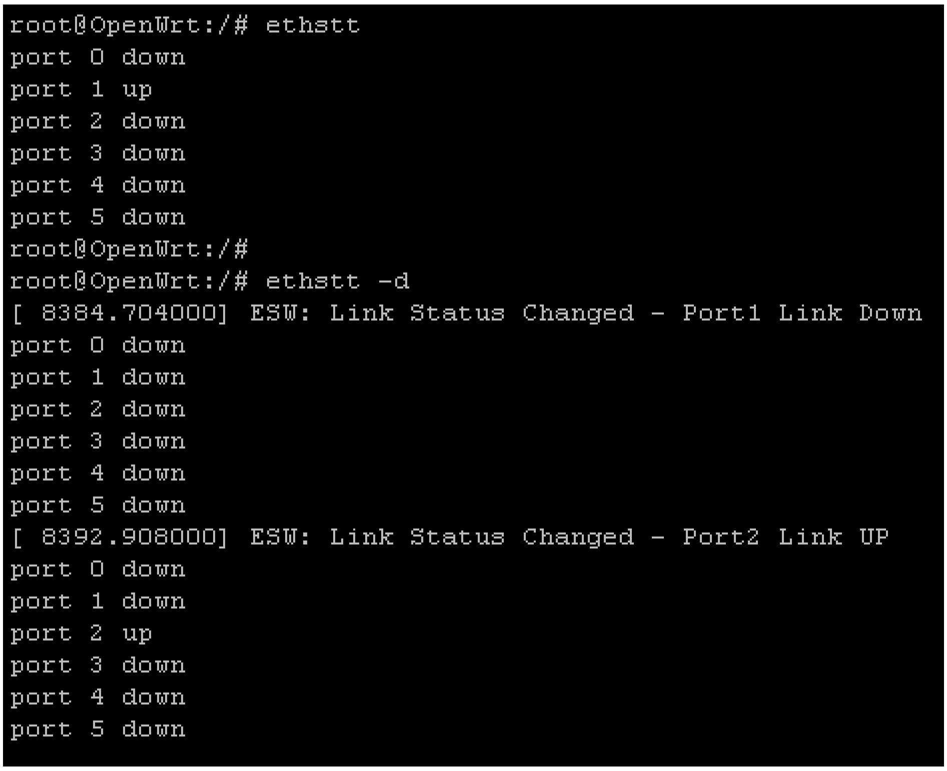
2.1.3 eth_mac

Change the Ethernet MAC address in EEPROM.

2.1.4 ethstt

A user tool to query switch port status.
Usage:

```
ethstt          # print switch port status
ethstat –d      # run as a daemon
```

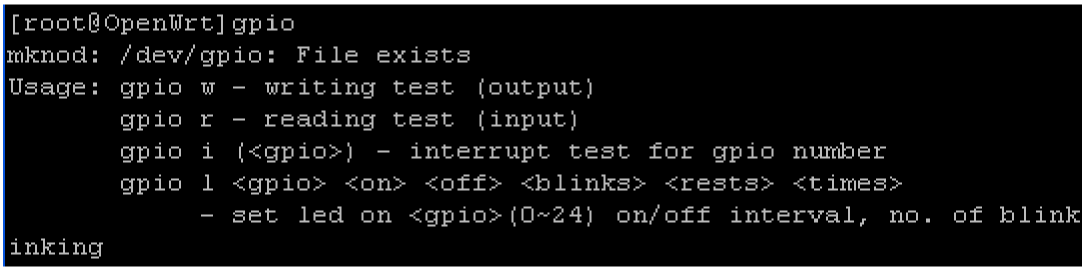


```
root@OpenWrt:/# ethstt
port 0 down
port 1 up
port 2 down
port 3 down
port 4 down
port 5 down
root@OpenWrt:/#
root@OpenWrt:/# ethstt -d
[ 8384.704000] ESW: Link Status Changed - Port1 Link Down
port 0 down
port 1 down
port 2 down
port 3 down
port 4 down
port 5 down
[ 8392.908000] ESW: Link Status Changed - Port2 Link UP
port 0 down
port 1 down
port 2 up
port 3 down
port 4 down
port 5 down
```

2.1.5 gpio

A user tool to query switch port status.
Usage:

gpio w - writing test (output)
gpio r - reading test (input)
gpio i (<gpio>) - interrupt test for gpio number
gpio l <gpio> <on> <off> <blinks> <rests> <times>



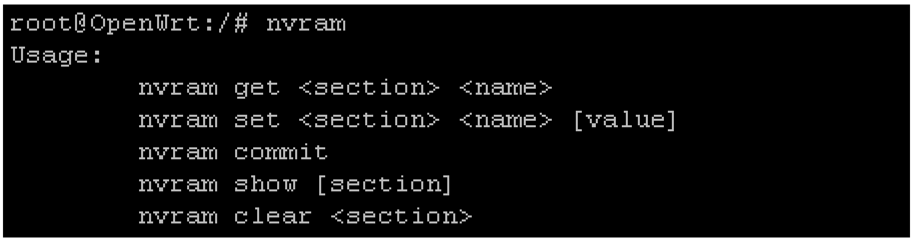
```
[root@OpenWrt]gpio
mknod: /dev/gpio: File exists
Usage: gpio w - writing test (output)
       gpio r - reading test (input)
       gpio i (<gpio>) - interrupt test for gpio number
       gpio l <gpio> <on> <off> <blinks> <rests> <times>
            - set led on <gpio>(0~24) on/off interval, no. of blink
inking
```

2.1.6 nvram

A user tool to manage nvram data.

Usage:

Please check the command help message.

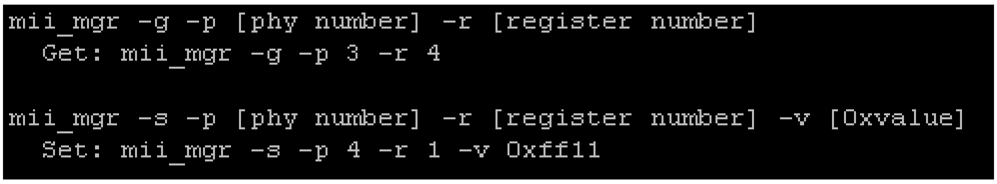


```
root@OpenWrt:/# nvram
Usage:
        nvram get <section> <name>
        nvram set <section> <name> [value]
        nvram commit
        nvram show [section]
        nvram clear <section>
```

2.1.7 mii_mgr

mii register read/wirte test program.

Usage:



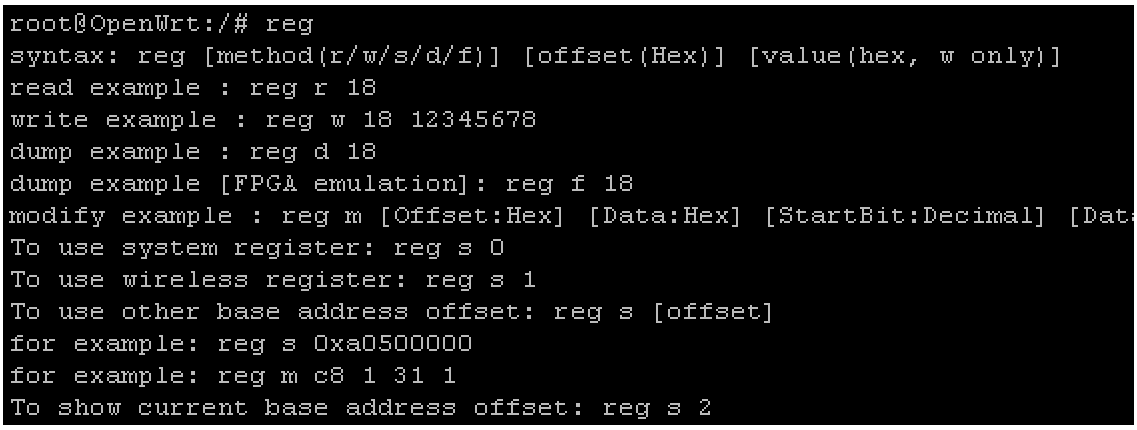
```
mii_mgr -g -p [phy number] -r [register number]
  Get: mii_mgr -g -p 3 -r 4

mii_mgr -s -p [phy number] -r [register number] -v [0xvalue]
  Set: mii_mgr -s -p 4 -r 1 -v 0xff11
```

2.1.8 reg

A user tool to debug system register

Usage:

Please check the command help message.



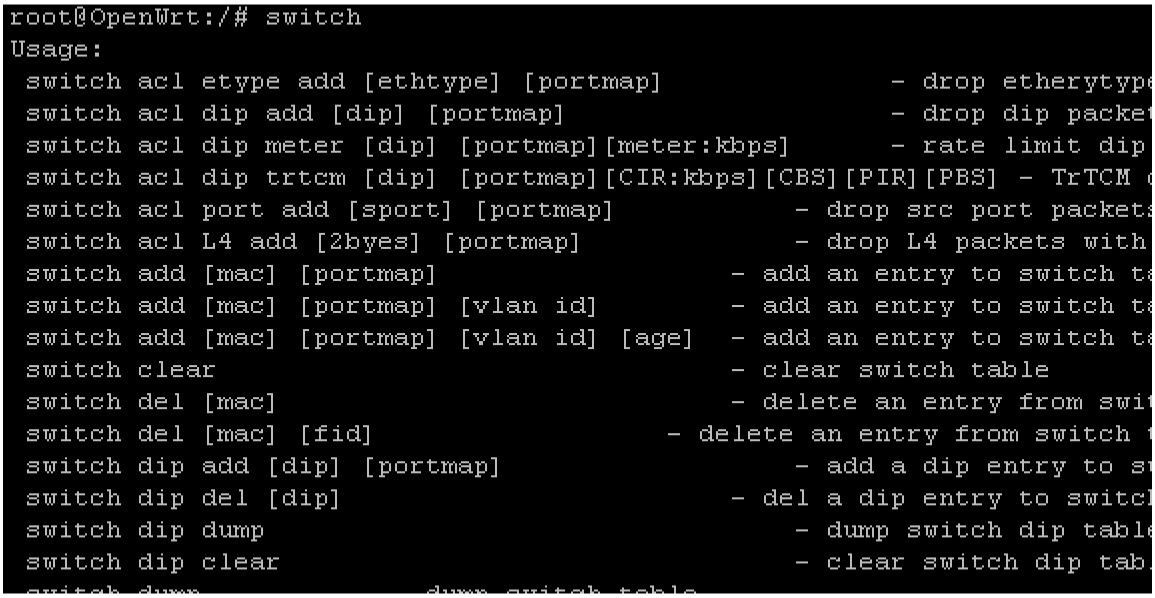
```
root@OpenWrt:/# reg
syntax: reg [method(r/w/s/d/f)] [offset(Hex)] [value(hex, w only)]
read example : reg r 18
write example : reg w 18 12345678
dump example : reg d 18
dump example [FPGA emulation]: reg f 18
modify example : reg m [Offset:Hex] [Data:Hex] [StartBit:Decimal] [Dat
To use system register: reg s 0
To use wireless register: reg s 1
To use other base address offset: reg s [offset]
for example: reg s 0xa0500000
for example: reg m c8 1 31 1
To show current base address offset: reg s 2
```

2.1.9 switch

A user tool to configure Ethernet switch

Usage:

Check the command help message.



```
root@OpenWrt:/# switch
Usage:
 switch acl etype add [ethtype] [portmap]              - drop etherytype
 switch acl dip add [dip] [portmap]                    - drop dip packet
 switch acl dip meter [dip] [portmap][meter:kbps]      - rate limit dip
 switch acl dip trtcm [dip] [portmap][CIR:kbps][CBS][PIR][PBS] - TrTCM
 switch acl port add [sport] [portmap]           - drop src port packets
 switch acl L4 add [2byes] [portmap]             - drop L4 packets with
 switch add [mac] [portmap]                  - add an entry to switch ta
 switch add [mac] [portmap] [vlan id]        - add an entry to switch ta
 switch add [mac] [portmap] [vlan id] [age]  - add an entry to switch ta
 switch clear                                - clear switch table
 switch del [mac]                            - delete an entry from swit
 switch del [mac] [fid]                  - delete an entry from switch t
 switch dip add [dip] [portmap]                  - add a dip entry to sw
 switch dip del [dip]                        - del a dip entry to switch
 switch dip dump                                 - dump switch dip table
 switch dip clear                                - clear switch dip tab
 switch dump            - dump switch table
```

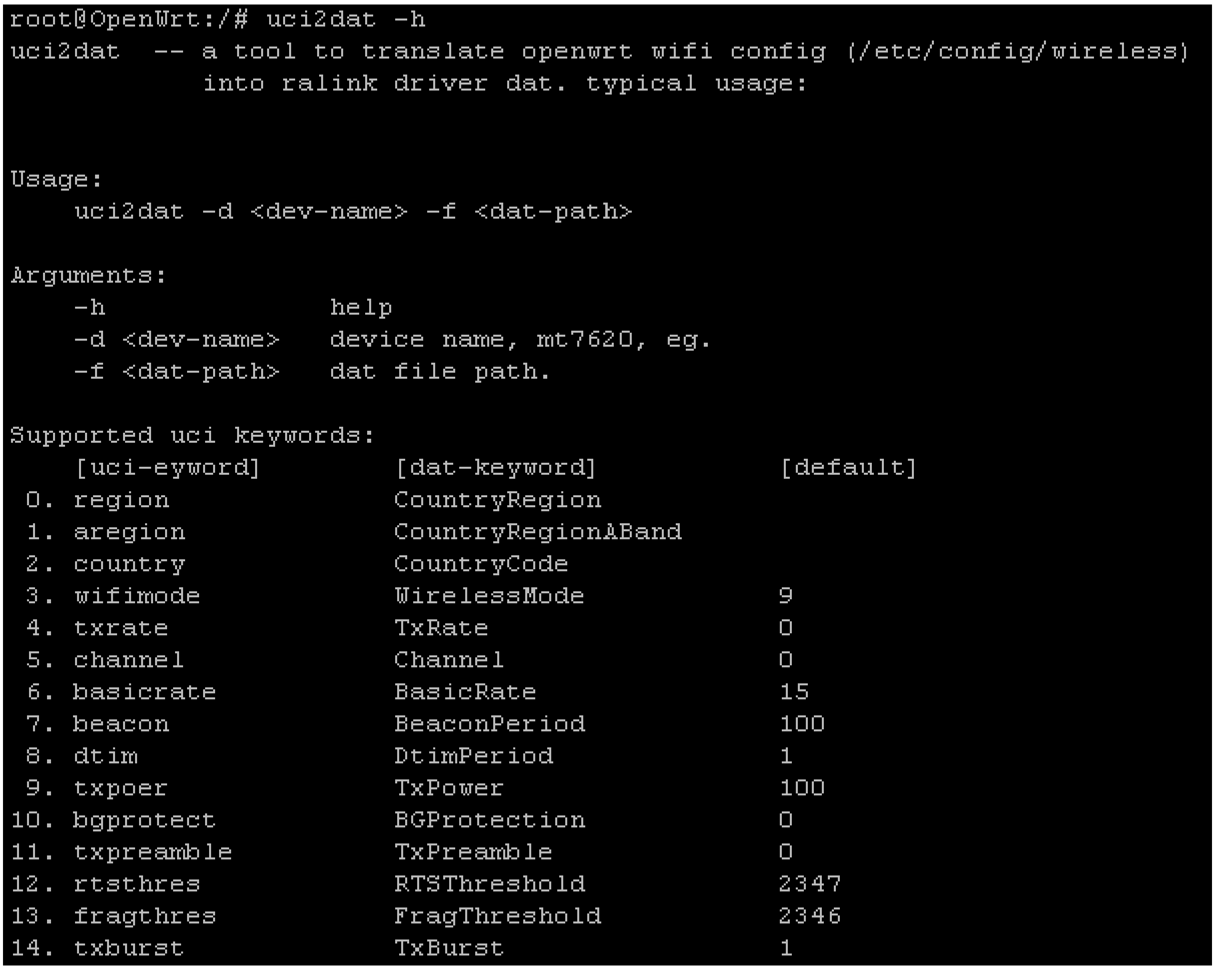
2.1.10 uci2dat

A user tool to translate OpenWrt uci configuration files into DAT files which can be loaded by Ralink wireless drivers.

Usage:

uci2dat -h # print help info

uci2dat -d devname -f dat filepath # devname is the “device“ name in uci file.



```
root@OpenWrt:/# uci2dat -h
uci2dat  -- a tool to translate openwrt wifi config (/etc/config/wireless)
            into ralink driver dat. typical usage:


Usage:
    uci2dat -d <dev-name> -f <dat-path>

Arguments:
    -h                help
    -d <dev-name>     device name, mt7620, eg.
    -f <dat-path>     dat file path.

Supported uci keywords:
    [uci-eyword]           [dat-keyword]            [default]
 0. region                 CountryRegion
 1. aregion                CountryRegionABand
 2. country                CountryCode
 3. wifimode               WirelessMode             9
 4. txrate                 TxRate                   0
 5. channel                Channel                  0
 6. basicrate              BasicRate                15
 7. beacon                 BeaconPeriod             100
 8. dtim                   DtimPeriod               1
 9. txpoer                 TxPower                  100
10. bgprotect              BGProtection             0
11. txpreamble             TxPreamble               0
12. rtsthres               RTSThreshold             2347
13. fragthres              FragThreshold            2346
14. txburst                TxBurst                  1
```

2.1.11 watchdog

An user space daemon to co-work with ralink watchdog hardware.
When this program started, it will wake up the watchdog hardware, and it feed the hardware every second. If watchdog program failed to function under some situations (like kernel oops, panic) , the system will be reset by watchdog hardware.

Usage:

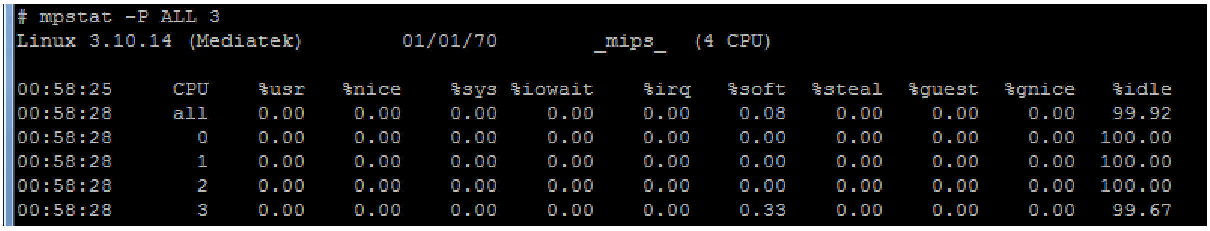
watchdog & **# silent daemon**

2.1.12 mpstat

A user tool to show cpu loading of multi- core processor

Usage:

mpstat [options] [<interval> [<count>]]



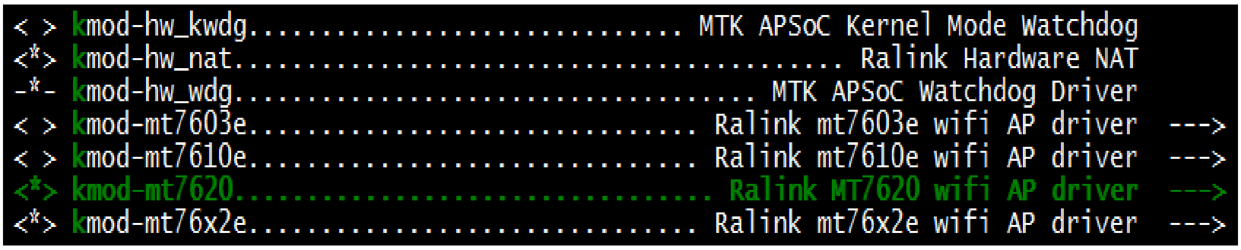
```
# mpstat -P ALL 3
Linux 3.10.14 (Mediatek)        01/01/70        _mips_  (4 CPU)

00:58:25     CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
00:58:28     all    0.00    0.00    0.00    0.00    0.00    0.08    0.00    0.00    0.00   99.92
00:58:28       0    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  100.00
00:58:28       1    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  100.00
00:58:28       2    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  100.00
00:58:28       3    0.00    0.00    0.00    0.00    0.00    0.33    0.00    0.00    0.00   99.67
```

2.2 Drivers

All drivers are put under **{SDKRoot}/dl/** folder.
You can choose the right driver for you device by:

"make menuconfig" -> "ralink properties" -> "drivers"

Each driver has its own configuration menu, make sure you didn't miss the features you want.



```
< > kmod-hw_kwdg............................. MTK APSoC Kernel Mode Watchdog
<*> kmod-hw_nat.......................................... Ralink Hardware NAT
-*- kmod-hw_wdg................................... MTK APSoC Watchdog Driver
< > kmod-mt7603e............................ Ralink mt7603e wifi AP driver  --->
< > kmod-mt7610e............................ Ralink mt7610e wifi AP driver  --->
<*> kmod-mt7620............................... Ralink MT7620 wifi AP driver  --->
<*> kmod-mt76x2e............................ Ralink mt76x2e wifi AP driver  --->
```

These drivers are all built as an OpenWrt packages, you can find the package definition under:

{SDKRoot}/package/ralink/drivers/mtxxxx/

```
├── config.in                 # the configuration script, can be seen in menuconfig
├── files
│   ├── mt76xx.dat                # default driver profile
│   └── mt76xx.sh                 # startup script, which co-work with OpenWrt framework
├── Makefile                      # OpenWrt package makefile
└── patches                       # patches which will be taken in during compiling
    ├── 001-xxxxxxx.patch
    …….
    ├── xxx-xxxxxxx.patch
```

2.2.1 MT7620

WiFi Driver for MT7620.
Driver configuration can be found at package/ramips/driver/mt7620/config.in

2.2.2 MT7610e

Driver for MT7610e
Driver configuration can be found at package/ramips/driver/mt7610e/config.in

2.2.3 MT7628/MT7688

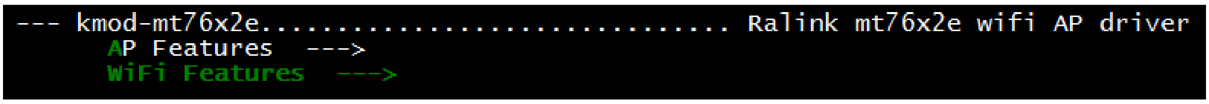
Driver for MT7628 and MT7688.
Driver configuration can be found at package/ramips/driver/mt7628/config.in

2.2.4 MT7603e

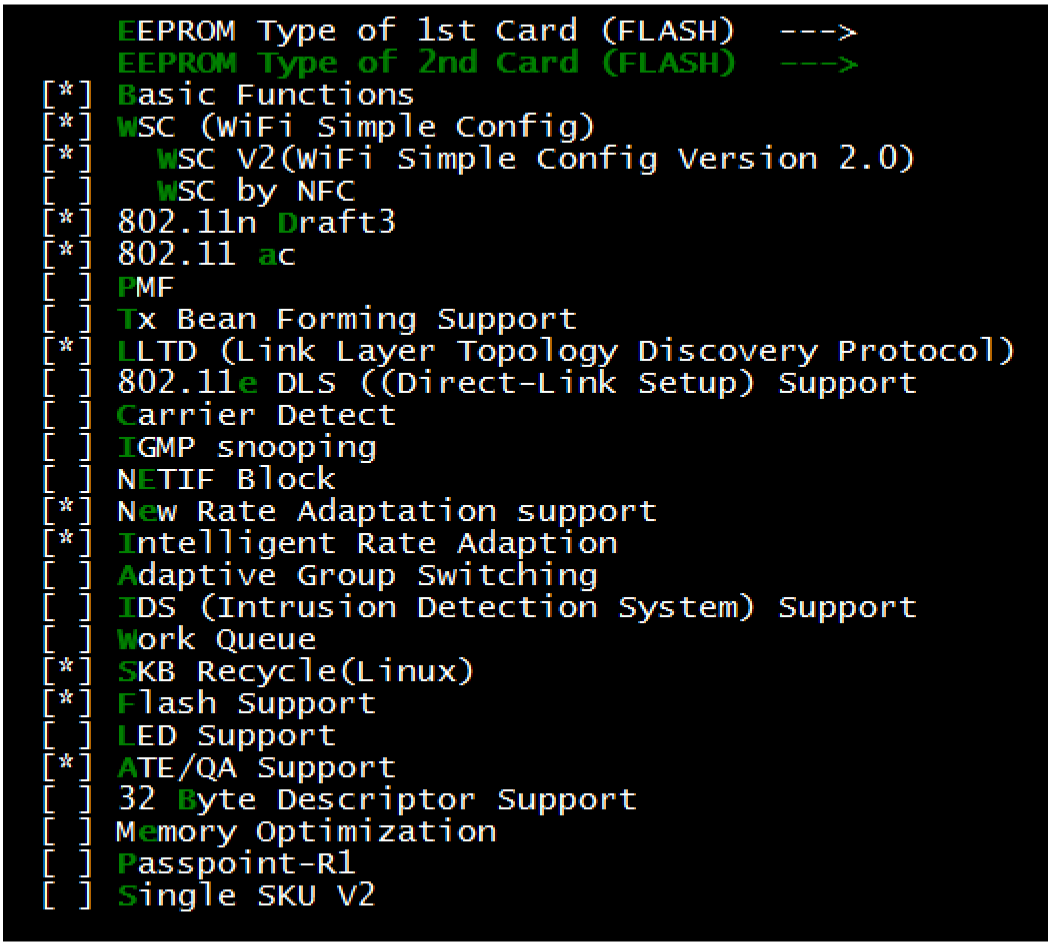
Driver for MT7603e
Driver configuration can be found at package/ramips/driver/mt7603e/config.in

2.2.5 MT76x2e

All in one driver for MT7602e (2.4G) and MT7612e (5G).
Driver configuration can be found at package/ramips/driver/mt76x2e / config.in
Note: If your MT7612e is onboard, you shoud config EEPROM Type of 2nd Card reading MT7612e's caldata from flash, if not select efuse.



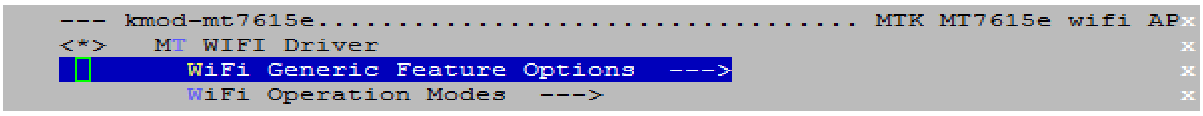
```
--- kmod-mt76x2e.............................. Ralink mt76x2e wifi AP driver
      AP Features   --->
      WiFi Features   --->
```



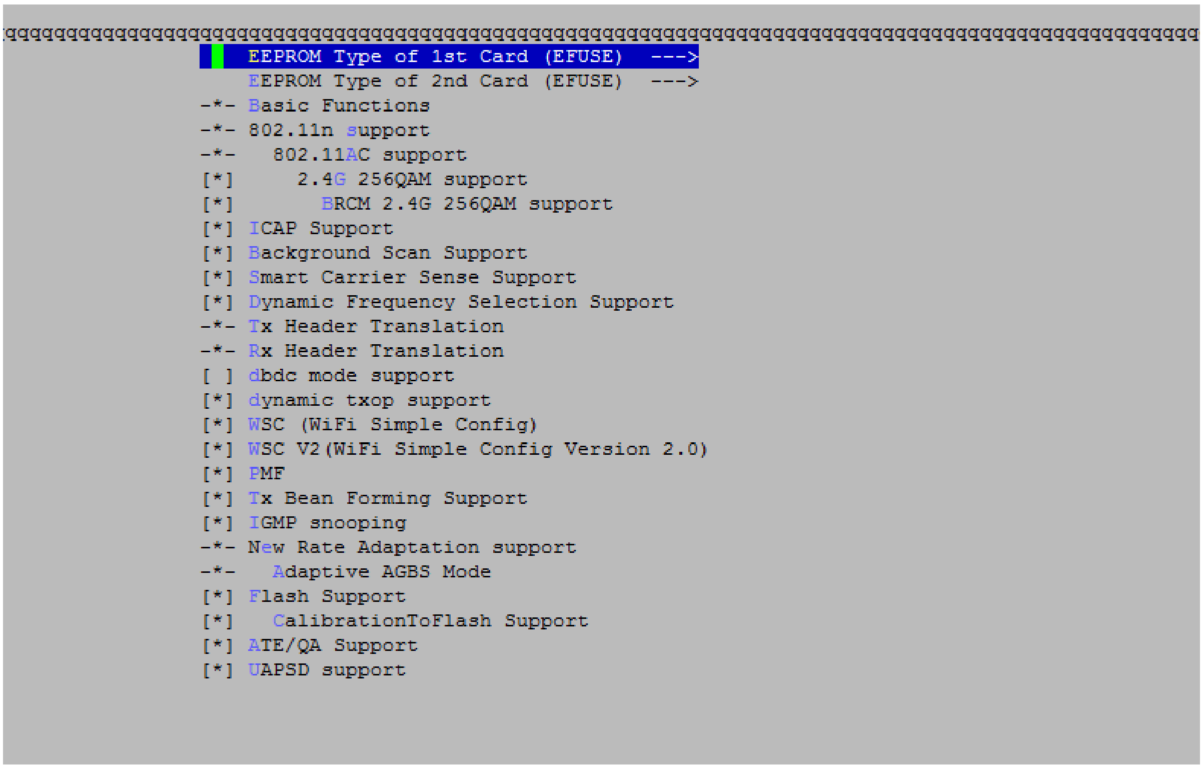
```
     EEPROM Type of 1st Card (FLASH)  --->
     EEPROM Type of 2nd Card (FLASH)  --->
 [*] Basic Functions
 [*] WSC (WiFi Simple Config)
 [*]   WSC V2(WiFi Simple Config Version 2.0)
 [ ]   WSC by NFC
 [*] 802.11n Draft3
 [*] 802.11 ac
 [ ] PMF
 [ ] Tx Bean Forming Support
 [*] LLTD (Link Layer Topology Discovery Protocol)
 [ ] 802.11e DLS ((Direct-Link Setup) Support
 [ ] Carrier Detect
 [ ] IGMP snooping
 [ ] NETIF Block
 [*] New Rate Adaptation support
 [*] Intelligent Rate Adaption
 [ ] Adaptive Group Switching
 [ ] IDS (Intrusion Detection System) Support
 [ ] Work Queue
 [*] SKB Recycle(Linux)
 [*] Flash Support
 [ ] LED Support
 [*] ATE/QA Support
 [ ] 32 Byte Descriptor Support
 [ ] Memory Optimization
 [ ] Passpoint-R1
 [ ] Single SKU V2
```

2.2.6 MT7615e

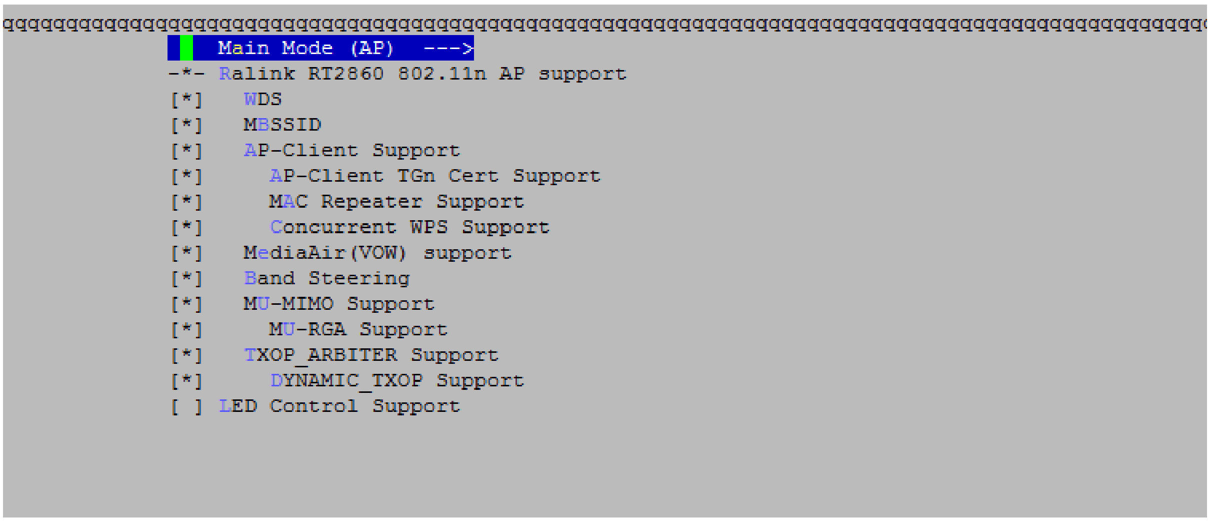
All in one driver for MT7615 2.4G and 5G.
Driver configuration can be found at package/ramips/driver/mt7615e / config.in
Note: for MT7615, e2p mode is config by .dat, so NO need to care about EEPROM Mode.



```
    --- kmod-mt7615e................................... MTK MT7615e wifi APx
    <*>   MT WIFI Driver                                                    x
              WiFi Generic Feature Options   --->                           x
              WiFi Operation Modes   --->                                   x
```



```
qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq
                    EEPROM Type of 1st Card (EFUSE)  --->
                    EEPROM Type of 2nd Card (EFUSE)  --->
                -*- Basic Functions
                -*- 802.11n support
                -*-   802.11AC support
                [*]     2.4G 256QAM support
                [*]       BRCM 2.4G 256QAM support
                [*] ICAP Support
                [*] Background Scan Support
                [*] Smart Carrier Sense Support
                [*] Dynamic Frequency Selection Support
                -*- Tx Header Translation
                -*- Rx Header Translation
                [ ] dbdc mode support
                [*] dynamic txop support
                [*] WSC (WiFi Simple Config)
                [*] WSC V2(WiFi Simple Config Version 2.0)
                [*] PMF
                [*] Tx Bean Forming Support
                [*] IGMP snooping
                -*- New Rate Adaptation support
                -*-   Adaptive AGBS Mode
                [*] Flash Support
                [*]   CalibrationToFlash Support
                [*] ATE/QA Support
                [*] UAPSD support
```



```
qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq
                 Main Mode (AP)  --->
             -*- Ralink RT2860 802.11n AP support
             [*]   WDS
             [*]   MBSSID
             [*]   AP-Client Support
             [*]     AP-Client TGn Cert Support
             [*]     MAC Repeater Support
             [*]     Concurrent WPS Support
             [*]   MediaAir(VOW) support
             [*]   Band Steering
             [*]   MU-MIMO Support
             [*]     MU-RGA Support
             [*]   TXOP_ARBITER Support
             [*]     DYNAMIC_TXOP Support
             [ ] LED Control Support
```

3 Patches

Target	Patch Name	Description
mt7603e	001-softirq-warning.patch	Fix a kernel softirq warning
mt7620a	001-fix-kernel-warning.patch	Fix a kernel warning
mt7620a	002-fix-insmod-fail.patch	Insmod fail with 3.10.x kernel
mt7620a	003-support-hwnat.patch	Support hwnat
kernel	0201-firmware-size.patch	firmware size based on flash size
kernel	0301-ramips-profile.patch	7621 irq profile
kernel	0302-rt-timer.patch	Rt_timer compile flag
kernel	0305-sdk4.3.0.5_20141205_ MT7621_HW_NAT.patch	Sync 4300 lsdk 7621 HNAT patch
kernel	0306-sdk4.3.0.10_20150209_ 6RD_issue.patch	Sync 4300 lsdk HNAT 6rd issue patch
kernel	0307-to-115775-nand-error.patch	Nand flash squashfs error pacth
mt76x2e	001-led.pacth	76x2e led support