Advantech Co., LTD.

WISE-3610 LoRa Gateway SDK Build Procedure

Quick Start Guide

2017/11/1

ADVANTECH
Enabling an Intelligent Planet

Content

1. BUILD CODE ENVIRONMENT 1
2. PREPARE SDK SOURCE 1
3. TO MAKE IMAGE 1
4. SDK TREE 2
5. FIRMWARE UPGRADE VIA RS-232 CONSOLE 4
6. FIRMWARE UPGRADE VIA SSH 12
7. FIRMWARE UPGRADE VIA WEB GUI 19
8. TO ENABLE TASKSET IN BUSYBOX 20
9. TO INCLUDE A SAMPLE TASK INTO FIRMWARE 24
10. ANOTHER APPLICATION EXAMPLE 28
11. ADD A SAMPLE WEBPAGE 29
12. INTERFACE BETWEEN GATEWAY PLATFORM AND MODULE BOARD 31
13. CONFIGURATION MANAGEMENT 33
14. WIFI COMMAND INTERFACE 37
15. TO BUILD PROGRAM WITH TOOLCHAIN DIRECTLY 40
16. REPRODUCE SDK 41
17. VIRTUAL BOX + UBUNTU 16.04 42
18. BUILD SDK WITH ARM FEATURE REQUIREMENT 45

Revision History

Version	Date	Modification
2.0	2017/6/25	SDK Build Procedure
2.1	2017/7/10	Firmware Upgrade via RS-232 or SSH or WEB GUI
2.2	2017/7/10	Add RS-232 Console Board Picture
2.3	2017/7/27	Build Code Environment Support Ubuntu 16.04
2.4	2017/11/1	Add ARM Feature Requirement

ADVANTECH
Enabling an Intelligent Planet

1. Build Code Environment

Ubuntu 16.04 Server and install below packages

```
$ sudo apt-get install gcc g++ binutils patch autoconf libcurl4-openssl-dev bzip2 flex make gettext pkg-config unzip zlib1g-dev libc6-dev subversion libncurses5-dev gawk sharutils curl libxml-parser-perl python-yaml git ocaml-nox ocaml ocaml-findlib bison texinfo ncurses-term zlib1g-dev openssl libssl-dev u-boot-tools device-tree-compiler  git git-core curl phablet-tools
```

2. Prepare SDK Source

To un-tar code base, please introduce below command

```
$ tar jxvf Dakota.tar.bz2
…
```

Then, you will have Dakota directory show up

```
$ ls
Dakota  Dakota.tar.bz2
```

3. To Make Image

To build firmware image, just introduce make command

```
$ cd Dakota
Dakota$ make
Dakota$ ls
error  image  Makefile  meta-scripts  private  qsdk
```

After build process is completed, the single image will be generated in image directory

```
Dakota$ ls -al image/
total 21536
drwxr-xr-x 2 stephen stephen     4096 Mar 10 16:00 .
drwxrwxr-x 4 stephen stephen     4096 Mar 10 16:00 ..
-rw-rw-r-- 1 stephen stephen 22043584 Mar 10 16:00 nand-ipq40xx-single.img
```

4. SDK Tree

- ToolChains

Dakota/qsdk/staging_dir$ ls
target-arm_cortex-a7_uClibc-0.9.33.2_eabi
toolchain-mips_34kc_gcc-4.8-linaro_uClibc-0.9.33.2
toolchain-arm_cortex-a7_gcc-4.8-linaro_uClibc-0.9.33.2_eabi

- OpenWRT Packages (https://openwrt.org/)

Dakota/qsdk/package$ ls
base-files boot charlietalk devel feeds firmware kernel libs Makefile network system utils

Dakota/qsdk/package/network/utils$ ls -al
total 136
drwxrwxr-x 34 stephen stephen 4096 Mar 13 16:56 .
drwxrwxr-x 6 stephen stephen 4096 Jul 11 2016 ..
drwxrwxr-x 3 stephen stephen 4096 Jul 11 2016 arptables
drwxrwxr-x 4 stephen stephen 4096 Jul 11 2016 comgt
drwxrwxr-x 2 stephen stephen 4096 Jul 11 2016 conntrack-tools
drwxrwxr-x 3 stephen stephen 4096 Jul 11 2016 dante
drwxrwxr-x 2 stephen stephen 4096 Jul 11 2016 ebtables
drwxrwxr-x 4 stephen stephen 4096 Mar 13 16:40 housekeeper
drwxrwxr-x 2 stephen stephen 4096 Jul 11 2016 ifenslave
drwxrwxr-x 3 stephen stephen 4096 Jul 11 2016 iftop
drwxrwxr-x 3 stephen stephen 4096 Jul 11 2016 iperf
drwxrwxr-x 4 stephen stephen 4096 Jul 11 2016 iproute2
drwxrwxr-x 3 stephen stephen 4096 Jul 11 2016 ipset
drwxrwxr-x 4 stephen stephen 4096 Jul 11 2016 iptables
drwxrwxr-x 3 stephen stephen 4096 Jul 11 2016 iputils
drwxrwxr-x 3 stephen stephen 4096 Jul 11 2016 iw
drwxrwxr-x 3 stephen stephen 4096 Jul 11 2016 iwcap
drwxrwxr-x 3 stephen stephen 4096 Jul 11 2016 iwinfo
drwxrwxr-x 4 stephen stephen 4096 Jul 11 2016 linux-atm
drwxrwxr-x 3 stephen stephen 4096 Jul 11 2016 maccalc
drwxrwxr-x 4 stephen stephen 4096 Mar 13 16:40 mail-mod
drwxrwxr-x 4 stephen stephen 4096 Mar 13 16:40 mosquitto
drwxrwxr-x 4 stephen stephen 4096 Mar 13 16:40 net-snmp
drwxrwxr-x 3 stephen stephen 4096 Jul 11 2016 owipcalc
drwxrwxr-x 3 stephen stephen 4096 Mar 13 16:40 paho
drwxrwxr-x 4 stephen stephen 4096 Mar 13 16:40 redis

```
drwxrwxr-x  3 stephen stephen 4096 Jul 11  2016 resolveip
drwxrwxr-x  4 stephen stephen 4096 Jul 11  2016 rssileds
drwxrwxr-x  4 stephen stephen 4096 Mar 13 16:40 smtp-mod
drwxrwxr-x  4 stephen stephen 4096 Mar 13 16:40 switch-mod
drwxrwxr-x  4 stephen stephen 4096 Mar 13 16:40 syslog-mod
drwxrwxr-x  3 stephen stephen 4096 Jul 11  2016 tcpdump
drwxrwxr-x  3 stephen stephen 4096 Jul 11  2016 wireless-tools
drwxrwxr-x  4 stephen stephen 4096 Jul 11  2016 xtables-addons

Dakota/qsdk/dl$ ls
1.0.4.3.arm
argp-standalone-1.3.tar.gz
autoconf-2.68.tar.bz2
automake-1.11.6.tar.xz
avahi-0.6.31.tar.gz
…
```

- Qualcomm Packages

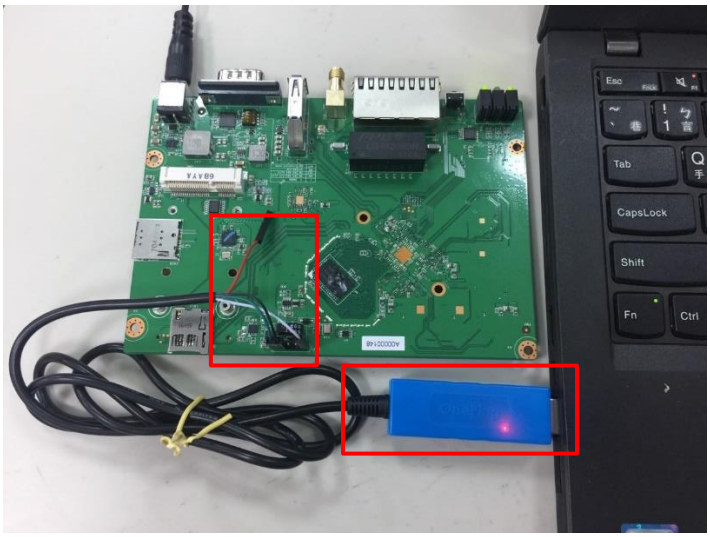
```
Dakota/qsdk/qca/src$ ls -al
total 44
drwxrwxr-x 11 stephen stephen 4096 Mar 13 16:56 .
drwxr-xr-x  5 stephen stephen 4096 Mar 13 16:40 ..
drwxrwxr-x  9 stephen stephen 4096 Mar 13 16:40 ath10k-firmware
drwxrwxr-x  2 stephen stephen 4096 Mar 13 16:40 btconfig
drwxrwxr-x  8 stephen stephen 4096 Mar 13 16:40 gobinet
drwxrwxr-x 25 stephen stephen 4096 Mar 13 16:55 linux
drwxrwxr-x  2 stephen stephen 4096 Mar 13 16:40 qca-ieee19051-dissector
drwxrwxr-x 27 stephen stephen 4096 Mar 13 16:40 qca-legacy-uboot
drwxrwxr-x  5 stephen stephen 4096 Mar 13 16:40 qca-romboot
drwxrwxr-x 21 stephen stephen 4096 Mar 13 16:40 u-boot
drwxrwxr-x 21 stephen stephen 4096 Mar 13 16:40 uboot-1.0
```

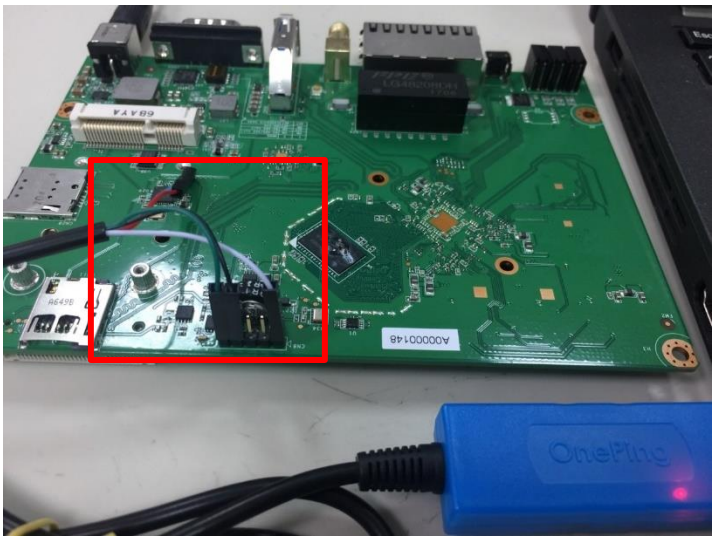
- LoRa and Qualcomm WIFI binaries

```
Dakota/qsdk/dl$ ls -al sdk-advanwise.tar.bz
-rw-rw-r-- 1 stephen stephen 58231258 Mar 13 16:56 sdk-advanwise.tar.bz

Dakota/qsdk/package/system/sdk$ ls
files  Makefile
```

5. Firmware Upgrade via RS-232 Console

(1) Firstly, connect console board to CN8 connector of WISE-3610, and the definition of CN8 is (TX, X, X, GND, RX). According to the console board we use, the White line is TX, the Black line is GND, and the Green Line is RX. (OP-1012-PBAM04D1)





(2) Commands to download firmware, and console logs as below

```
Format: Log Type - Time(microsec) - Message - Optional Info
Log Type: B - Since Boot(Power On Reset),  D - Delta,  S - Statistic
S - QC_IMAGE_VERSION_STRING=BOOT.BF.3.1.1-00096
S - IMAGE_VARIANT_STRING=DAACANAZA
S - OEM_IMAGE_VERSION_STRING=CRM
S - Boot Config, 0x00000025
S - Core 0 Frequency, 0 MHz
B -      261 - PBL, Start
B -     1339 - bootable_media_detect_entry, Start
B -     2877 - bootable_media_detect_success, Start
B -     2891 - elf_loader_entry, Start
B -     4297 - auth_hash_seg_entry, Start
B -     6448 - auth_hash_seg_exit, Start
B -    74008 - elf_segs_hash_verify_entry, Start
B -   194384 - PBL, End
B -   194408 - SBL1, Start
B -   283059 - pm_device_init, Start
D -        6 - pm_device_init, Delta
B -   284585 - boot_flash_init, Start
D -    84655 - boot_flash_init, Delta
```

```
B -    373287 - boot_config_data_table_init, Start
D -     13982 - boot_config_data_table_init, Delta - (419 Bytes)
B -    389964 - clock_init, Start
D -     7572 - clock_init, Delta
B -    400936 - CDT version:2,Platform ID:8,Major ID:1,Minor ID:0,Subtype:1
B -    404423 - sbl1_ddr_set_params, Start
B -    409408 - cpr_init, Start
D -       2 - cpr_init, Delta
B -    413897 - Pre_DDR_clock_init, Start
D -       5 - Pre_DDR_clock_init, Delta
D -     13141 - sbl1_ddr_set_params, Delta
B -    427177 - pm_driver_init, Start
D -       2 - pm_driver_init, Delta
B -    497720 - sbl1_wait_for_ddr_training, Start
D -      27 - sbl1_wait_for_ddr_training, Delta
B -    513246 - Image Load, Start
D -    140614 - QSEE Image Loaded, Delta - (262104 Bytes)
B -    654359 - Image Load, Start
D -     2118 - SEC Image Loaded, Delta - (2048 Bytes)
B -    664516 - Image Load, Start
D -    176286 - APPSBL Image Loaded, Delta - (417751 Bytes)
B -    841224 - QSEE Execution, Start
D -      56 - QSEE Execution, Delta
B -    847383 - SBL1, End
D -    655161 - SBL1, Delta
S - Flash Throughput, 2074 KB/s  (682322 Bytes,  328836 us)
S - DDR Frequency, 672 MHz


U-Boot 2012.07 [WISE-3610 R1.0.0,unknown] (Mar 10 2017 - 15:00:53)

smem ram ptable found: ver: 1 len: 3
DRAM:  256 MiB
machid : 0x8010001
NAND:  ONFI device found
ID = 9580f12c
Vendor = 2c
Device = f1
SF: Detected W25Q128 with page size 64 KiB, total 16 MiB
ipq_spi: page_size: 0x100, sector_size: 0x10000, size: 0x1000000
144 MiB
MMC:   qca_mmc: 0
*** Warning - bad CRC, using default environment
```

```
In:    serial
Out:   serial
Err:   serial
machid: 8010001
flash_type: 2
Net:   MAC0 addr:0:3:7f:ba:db:ad
PHY ID1: 0x4d
PHY ID2: 0xd0b1
ipq40xx_ess_sw_init done
eth0
Hit any key to stop autoboot:  0
(IPQ40xx) # set ipaddr 192.168.1.1
(IPQ40xx) # set serverip 192.168.1.100
(IPQ40xx) # tftpboot nand-ipq40xx-single.img
eth0 PHY0 Down Speed :10 Half duplex
eth0 PHY1 Down Speed :10 Half duplex
eth0 PHY2 up Speed :1000 Full duplex
eth0 PHY3 Down Speed :10 Half duplex
eth0 PHY4 Down Speed :10 Half duplex
Using eth0 device
TFTP from server 192.168.1.100; our IP address is 192.168.1.1
Filename 'nand-ipq40xx-single.img'.
Load address: 0x84000000
Loading: #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
```

```
         ##################################################################
         ##################################################################
         ##################################################################
         ##################################################################
         #######
done
Bytes transferred = 22043584 (1505bc0 hex)
(IPQ40xx) # imgaddr=0x84000000 && source $imgaddr:script && reset
## Executing script at 84000000
crc32+ Flashing mibib:                  ## Copying 'mibib-fb94cca75b16a5a04cae01227af254a0e9039bf8'
subimage from FIT image at 84000000 ...
crc32+
NAND erase: device 0 offset 0x100000, size 0x100000
Erasing at 0x1e0000 -- 100% complete.
OK

NAND write: device 0 offset 0x100000, size 0x40000
 262144 bytes written: OK
[ done ]
Flashing sbl1:                  ## Copying 'sbl1-73fb8022f5abb040c722a5d4674591b6463cfa1a' subimage from
FIT image at 84000000 ...
crc32+
NAND erase: device 0 offset 0x0, size 0x100000
Erasing at 0xe0000 -- 100% complete.
OK

NAND write: device 0 offset 0x0, size 0x25000
 151552 bytes written: OK
[ done ]
Flashing ddr-AP-DK04.1-C1:          ## Copying 'ddr-AP-DK04.1-C1-
44f7cf880531f125fc2394a28013813eb1a756e5' subimage from FIT image at 84000000 ...
crc32+
NAND erase: device 0 offset 0x500000, size 0x80000
Erasing at 0x560000 -- 100% complete.
OK

NAND write: device 0 offset 0x500000, size 0x800
 2048 bytes written: OK
[ done ]
Flashing tz:                    ## Copying 'tz-7fb7fc3700e39853414a46c5956c80067bd3af08' subimage from FIT
image at 84000000 ...
crc32+
NAND erase: device 0 offset 0x300000, size 0x100000
```

```
Erasing at 0x3e0000 -- 100% complete.
OK

NAND write: device 0 offset 0x300000, size 0x51800
 333824 bytes written: OK
[ done ]
Flashing u-boot:                 ## Copying 'u-boot-ee4297641e8ac05e0faa79f61de22344c4258284' subimage
from FIT image at 84000000 ...
crc32+
NAND erase: device 0 offset 0x700000, size 0x200000
Erasing at 0x8e0000 -- 100% complete.
OK

NAND write: device 0 offset 0x700000, size 0x6e000
 450560 bytes written: OK
[ done ]
Flashing ubi:                    ## Copying 'ubi-2113e3f3cc2a94e31f40d2c220669cca1b7e2845' subimage from
FIT image at 84000000 ...
crc32+
NAND erase: device 0 offset 0xb80000, size 0x4100000
Erasing at 0x4c60000 -- 100% complete.
OK

NAND write: device 0 offset 0xb80000, size 0x13c0000
 20709376 bytes written: OK
[ done ]
resetting ...

Format: Log Type - Time(microsec) - Message - Optional Info
Log Type: B - Since Boot(Power On Reset),  D - Delta,  S - Statistic
S - QC_IMAGE_VERSION_STRING=BOOT.BF.3.1.1-00096
S - IMAGE_VARIANT_STRING=DAACANAZA
S - OEM_IMAGE_VERSION_STRING=CRM
S - Boot Config, 0x00000025
S - Core 0 Frequency, 0 MHz
B -       261 - PBL, Start
B -      1339 - bootable_media_detect_entry, Start
B -      2609 - bootable_media_detect_success, Start
B -      2624 - elf_loader_entry, Start
B -      4030 - auth_hash_seg_entry, Start
B -      6181 - auth_hash_seg_exit, Start
B -     73755 - elf_segs_hash_verify_entry, Start
B -    194123 - PBL, End
```

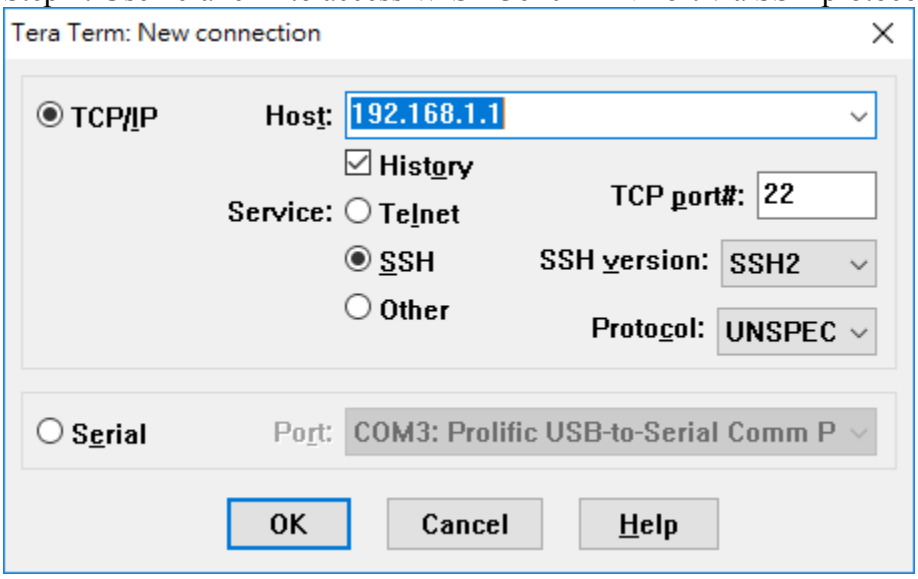
```
B -   194147 - SBL1, Start
B -   282787 - pm_device_init, Start
D -       6 - pm_device_init, Delta
B -   284313 - boot_flash_init, Start
D -    84653 - boot_flash_init, Delta
B -   373016 - boot_config_data_table_init, Start
D -    13981 - boot_config_data_table_init, Delta - (419 Bytes)
B -   389694 - clock_init, Start
D -     7562 - clock_init, Delta
B -   400659 - CDT version:2,Platform ID:8,Major ID:1,Minor ID:0,Subtype:1
B -   404146 - sbl1_ddr_set_params, Start
B -   409131 - cpr_init, Start
D -       2 - cpr_init, Delta
B -   413621 - Pre_DDR_clock_init, Start
D -       5 - Pre_DDR_clock_init, Delta
D -    13142 - sbl1_ddr_set_params, Delta
B -   426902 - pm_driver_init, Start
D -       2 - pm_driver_init, Delta
B -   497502 - sbl1_wait_for_ddr_training, Start
D -      27 - sbl1_wait_for_ddr_training, Delta
B -   513031 - Image Load, Start
D -   140616 - QSEE Image Loaded, Delta - (262104 Bytes)
B -   654145 - Image Load, Start
D -     2118 - SEC Image Loaded, Delta - (2048 Bytes)
B -   664307 - Image Load, Start
D -   176297 - APPSBL Image Loaded, Delta - (417751 Bytes)
B -   841026 - QSEE Execution, Start
D -      56 - QSEE Execution, Delta
B -   847183 - SBL1, End
D -   655222 - SBL1, Delta
S - Flash Throughput, 2074 KB/s  (682322 Bytes,  328845 us)
S - DDR Frequency, 672 MHz


U-Boot 2012.07 [WISE-3610 R1.0.0,unknown] (Mar 10 2017 - 15:48:16)

smem ram ptable found: ver: 1 len: 3
DRAM:  256 MiB
machid : 0x8010001
NAND:  ONFI device found
ID = 9580f12c
Vendor = 2c
Device = f1
```

```
SF: Detected W25Q128 with page size 64 KiB, total 16 MiB
ipq_spi: page_size: 0x100, sector_size: 0x10000, size: 0x1000000
144 MiB
MMC:   qca_mmc: 0
*** Warning - bad CRC, using default environment

In:    serial
Out:   serial
Err:   serial
machid: 8010001
flash_type: 2
Net:   MAC0 addr:0:3:7f:ba:db:ad
PHY ID1: 0x4d
PHY ID2: 0xd0b1
ipq40xx_ess_sw_init done
eth0
Hit any key to stop autoboot:  0
Creating 1 MTD partitions on "nand0":
0x000000b80000-0x000004c80000 : "mtd=0"
UBI: attaching mtd2 to ubi0
UBI: physical eraseblock size:   131072 bytes (128 KiB)
UBI: logical eraseblock size:    126976 bytes
UBI: smallest flash I/O unit:    2048
UBI: VID header offset:          2048 (aligned 2048)
UBI: data offset:                4096
UBI: volume 2 ("rootfs_data") re-sized from 1 to 355 LEBs
UBI: attached mtd2 to ubi0
UBI: MTD device name:            "mtd=0"
UBI: MTD device size:            65 MiB
UBI: number of good PEBs:        520
UBI: number of bad PEBs:         0
UBI: max. allowed volumes:       128
UBI: wear-leveling threshold:    4096
UBI: number of internal volumes: 1
UBI: number of user volumes:     3
UBI: available PEBs:             0
UBI: total number of reserved PEBs: 520
UBI: number of PEBs reserved for bad PEB handling: 5
UBI: max/mean erase counter: 1/0
Read 0 bytes from volume kernel to 84000000
No size specified -> Using max size (3809280)
## Booting kernel from FIT Image at 84000000 ...
   Using 'config@1' configuration
```

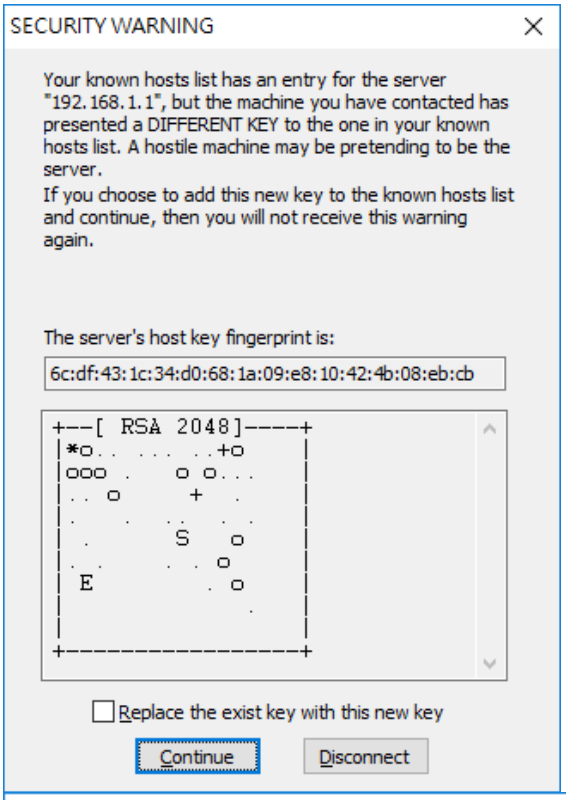
```
   Trying 'kernel@1' kernel subimage
     Description:  ARM OpenWrt Linux-3.14.43
     Type:         Kernel Image
     Compression:  gzip compressed
     Data Start:   0x840000e4
     Data Size:    3300641 Bytes = 3.1 MiB
     Architecture: ARM
     OS:           Linux
     Load Address: 0x80208000
     Entry Point:  0x80208000
     Hash algo:    crc32
     Hash value:   788850e2
     Hash algo:    sha1
     Hash value:   5d50c93c633a3792ba6b20331b563eba661f07ba
   Verifying Hash Integrity ... crc32+ sha1+ OK
## Flattened Device Tree from FIT Image at 84000000
   Using 'config@1' configuration
   Trying 'fdt@1' FDT blob subimage
     Description:  ARM OpenWrt qcom-ipq40xx-ap.dkxx device tree blob
     Type:         Flat Device Tree
     Compression:  uncompressed
     Data Start:   0x84325f48
     Data Size:    36585 Bytes = 35.7 KiB
     Architecture: ARM
     Hash algo:    crc32
     Hash value:   edde39eb
     Hash algo:    sha1
     Hash value:   9d2fcc96070cd8a5d5216860059a180b2e2e6df9
   Verifying Hash Integrity ... crc32+ sha1+ OK
   Booting using the fdt blob at 0x84325f48
   Uncompressing Kernel Image ... OK
   Loading Device Tree to 86ff4000, end 86fffee8 ... OK
eth0 MAC Address from ART is not valid
eth1 MAC Address from ART is not valid
Using machid 0x8010001 from environment

Starting kernel ...

[    0.000000] Booting Linux on physical CPU 0x0
[    0.000000] Linux version 3.14.43 (stephen@AdvanWISE-YG-409) (gcc version 4.8.3 (OpenWrt/Linaro GCC
4.8-2014.01 unknown) ) #3 SMP PREEMPT Fri Mar 10 15:59:12 CST 2017
```

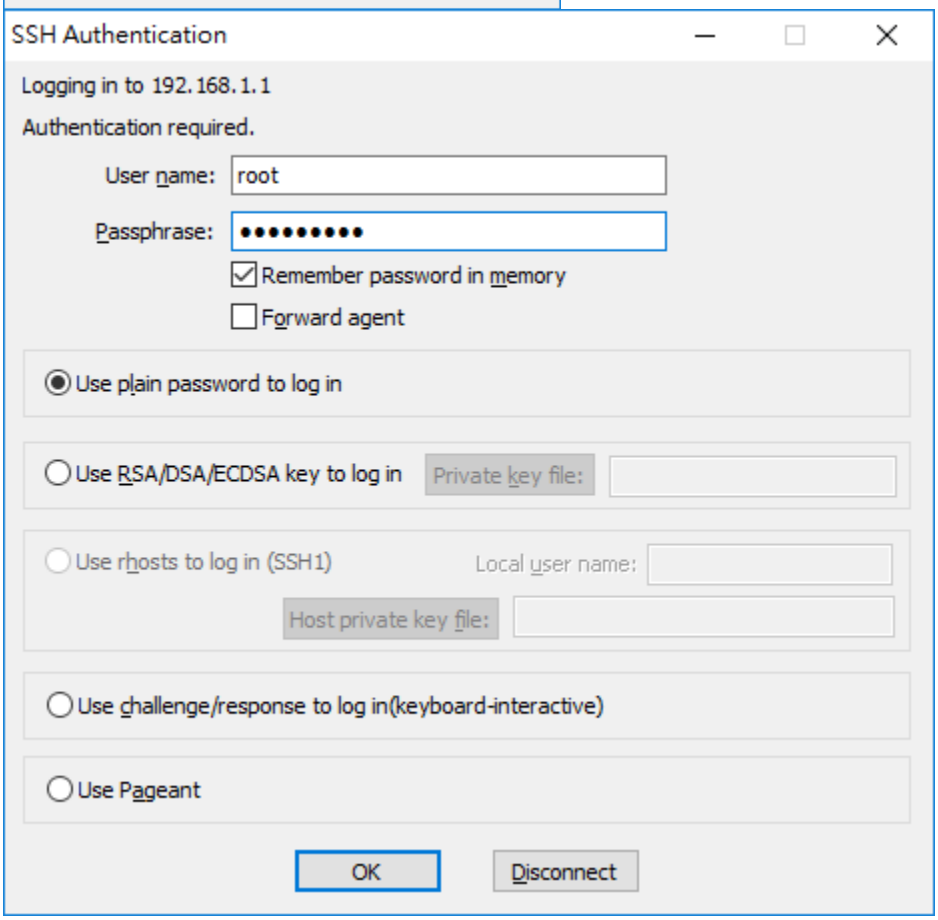
6. Firmware Upgrade via SSH

Step 1: Use TeraTerm to access WISE-3610 LAN Port via SSH protocol

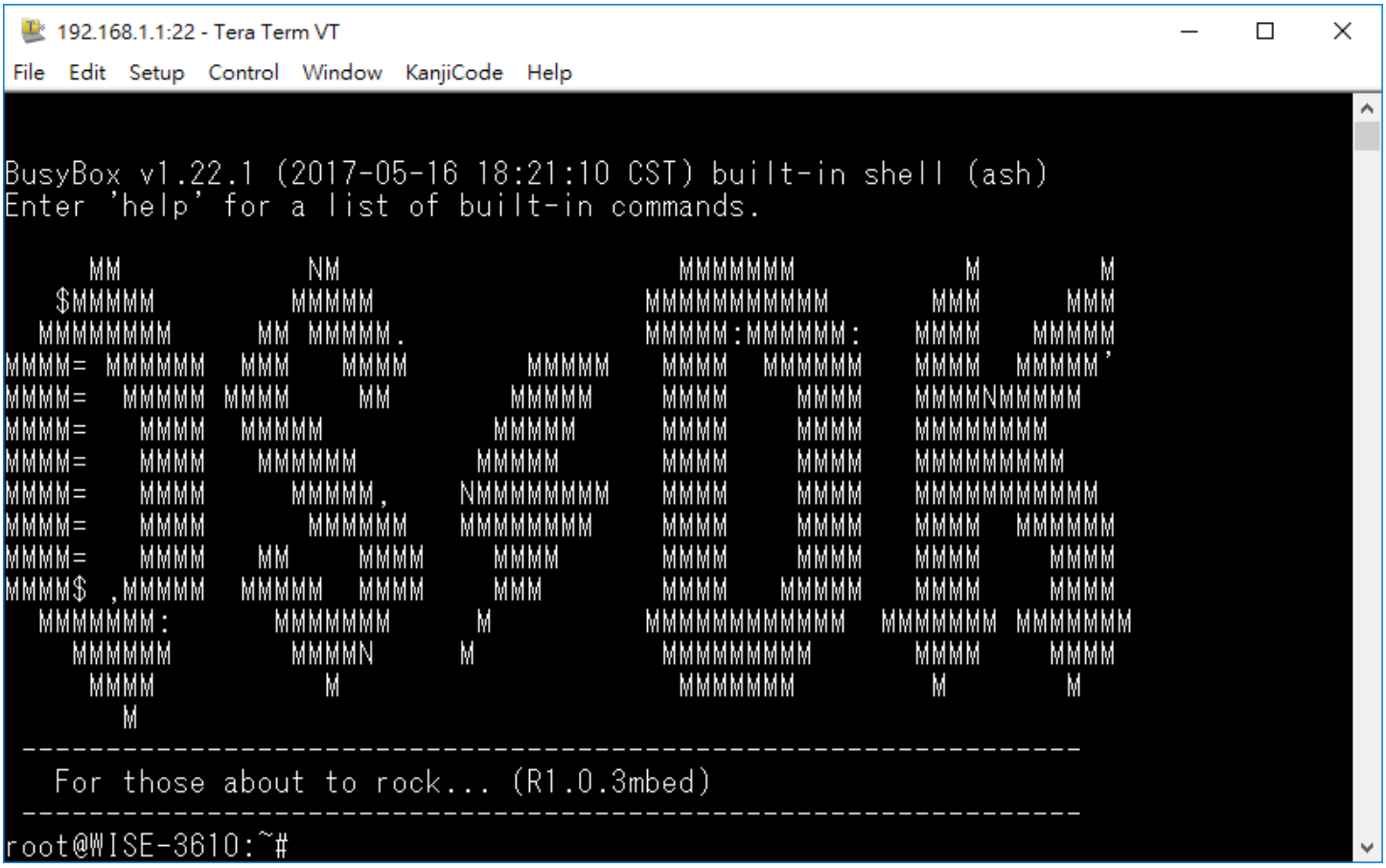


Step 2: User have to accept Certificate when login to shell, and use root/advantech account information to login

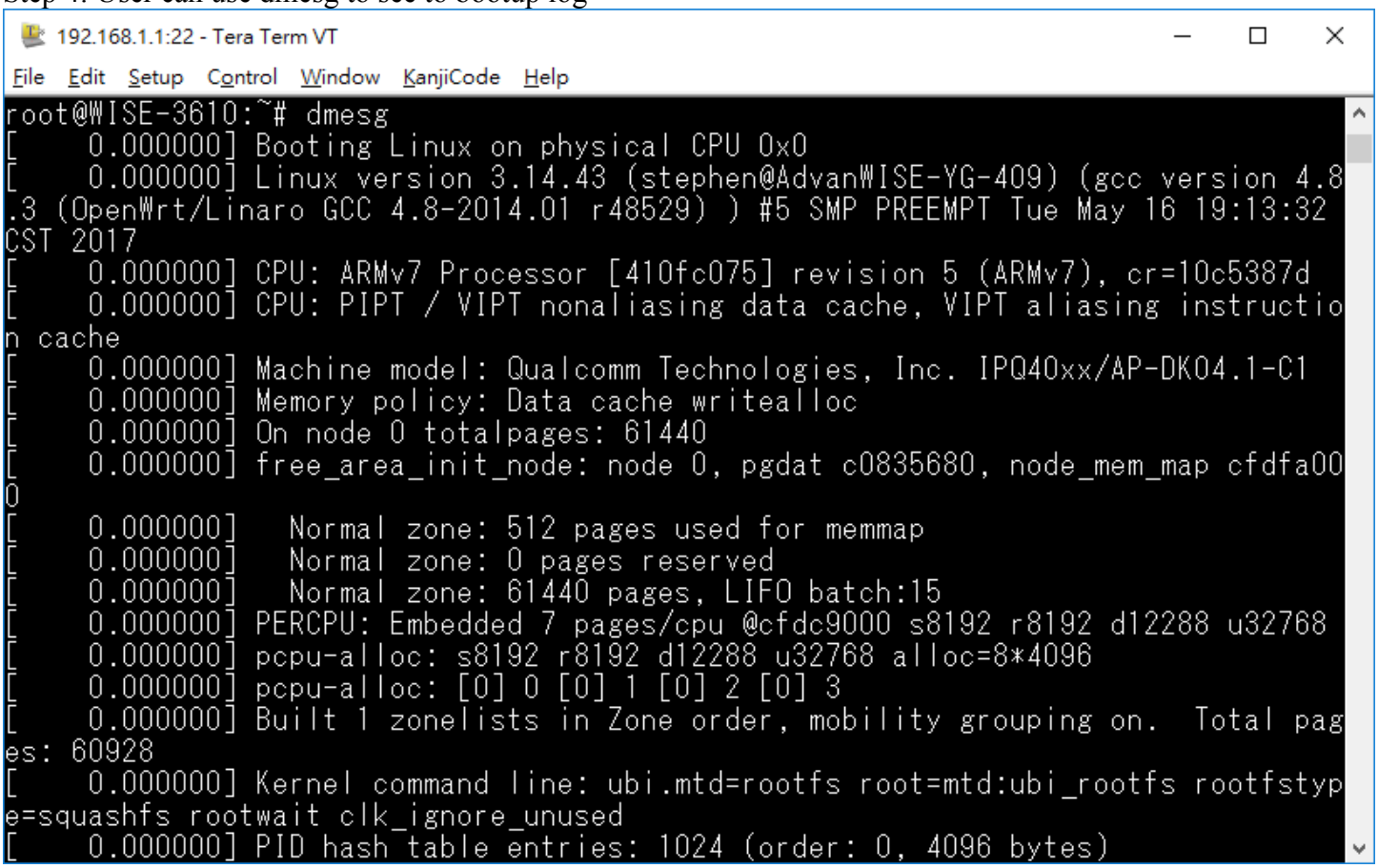

SECURITY WARNING
Your known hosts list has an entry for the server "192.168.1.1", but the machine you have contacted has presented a DIFFERENT KEY to the one in your known hosts list. A hostile machine may be pretending to be the server.
If you choose to add this new key to the known hosts list and continue, then you will not receive this warning again.
The server's host key fingerprint is:
6c:df:43:1c:34:d0:68:1a:09:e8:10:42:4b:08:eb:cb
+--[RSA 2048]----+
Replace the exist key with this new key
Continue
Disconnect


SSH Authentication
Logging in to 192.168.1.1
Authentication required.
User name: root
Passphrase:
Remember password in memory
Forward agent
Use plain password to log in
Use RSA/DSA/ECDSA key to log in
Private key file:
Use rhosts to log in (SSH1)
Local user name:
Host private key file:
Use challenge/response to log in(keyboard-interactive)
Use Pageant
OK
Disconnect

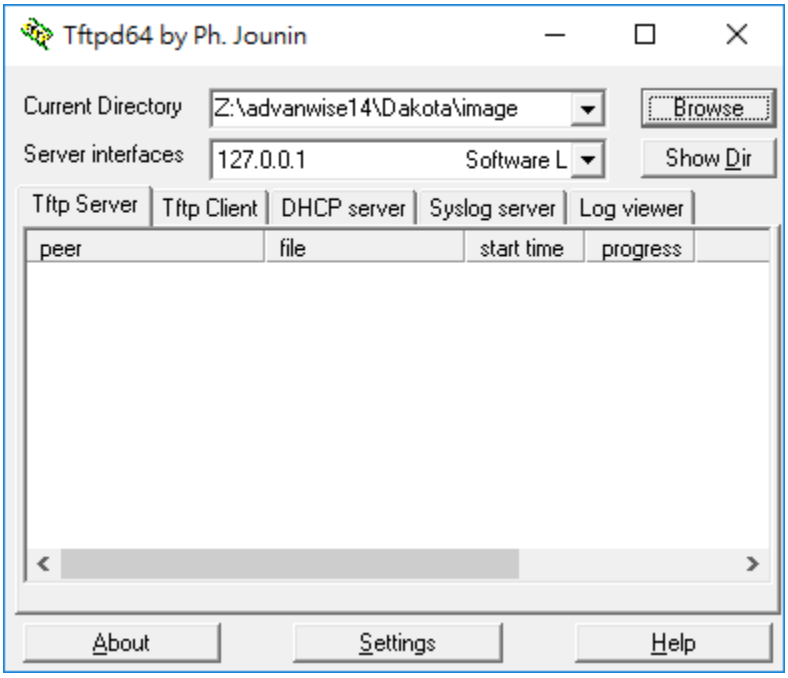
Step 3: After authentication pass, the engineer shell is as below


192.168.1.1:22 - Tera Term VT
File Edit Setup Control Window KanjiCode Help
BusyBox v1.22.1 (2017-05-16 18:21:10 CST) built-in shell (ash)
Enter 'help' for a list of built-in commands.
For those about to rock... (R1.0.3mbed)
root@WISE-3610:~#

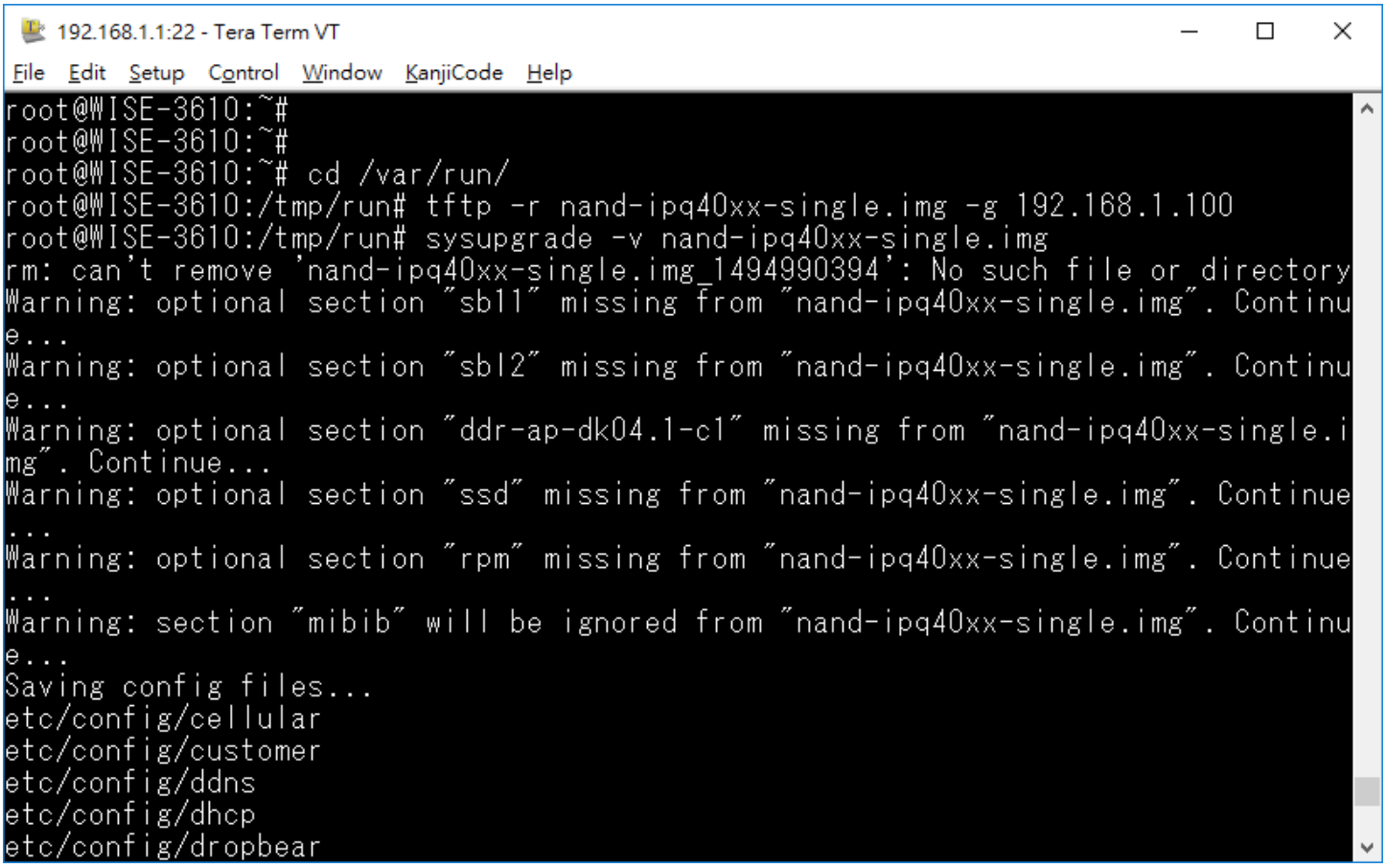
Step 4: User can use dmesg to see to bootup log



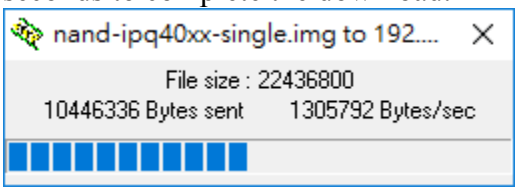
Step 5 : Configure tftp server on PC to point to the directory of WISE-3610 Firmware (e.g. Current Directory of Tftpd64)



Step 6: Introduce ‘tftp -r nand-ipq40xx-single.img -g 192.168.1.100’ command to download firmware from NB/PC to WISE-3610. Finally, introduce ‘sysupgrade -v nand-ipq40xx-single.img’ to burn firmware into flash.

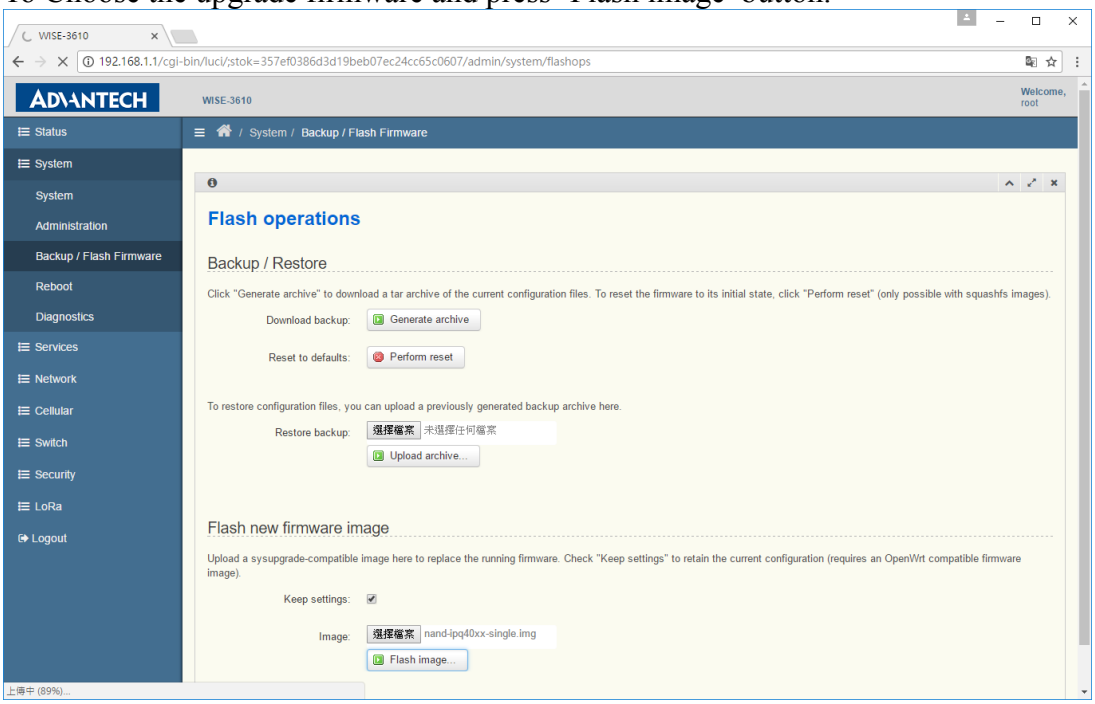


note : When tftp is downloading firmware, user can see the progress status on NB/PC and it take about 30~60 seconds to complete the download.

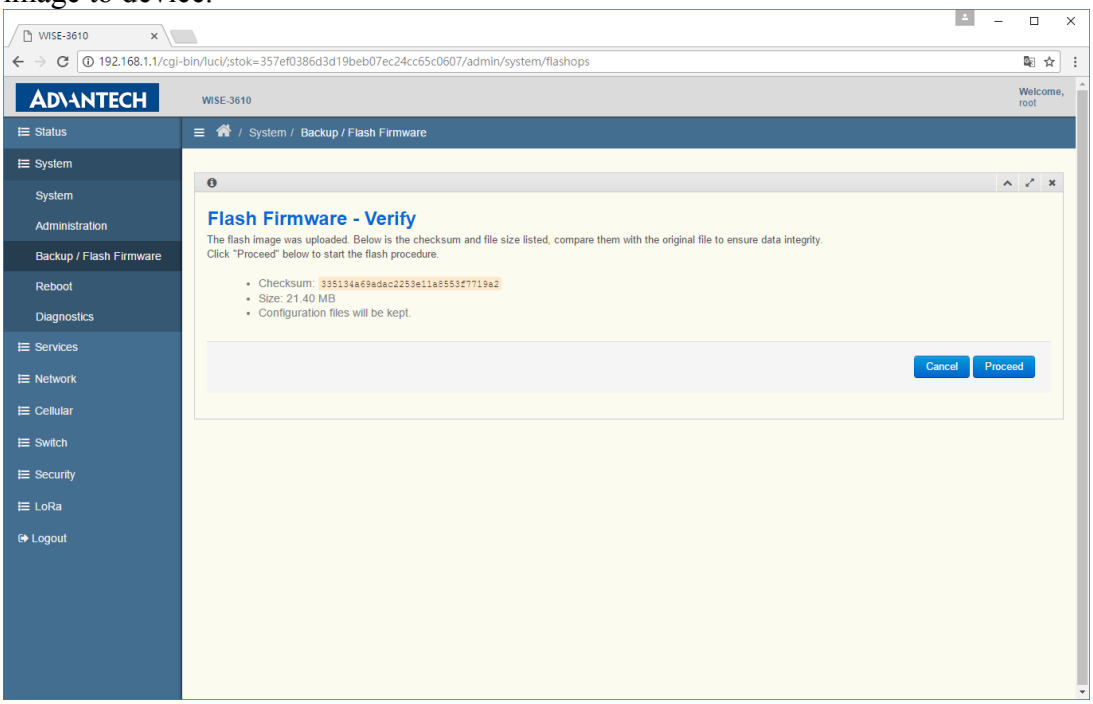


7. Firmware Upgrade via WEB GUI

Step 1: Click ‘Backup / Flash Firmware link, and go to ‘Flash new firmware image’ paragraph.
To Choose the upgrade firmware and press ‘Flash image’ button.

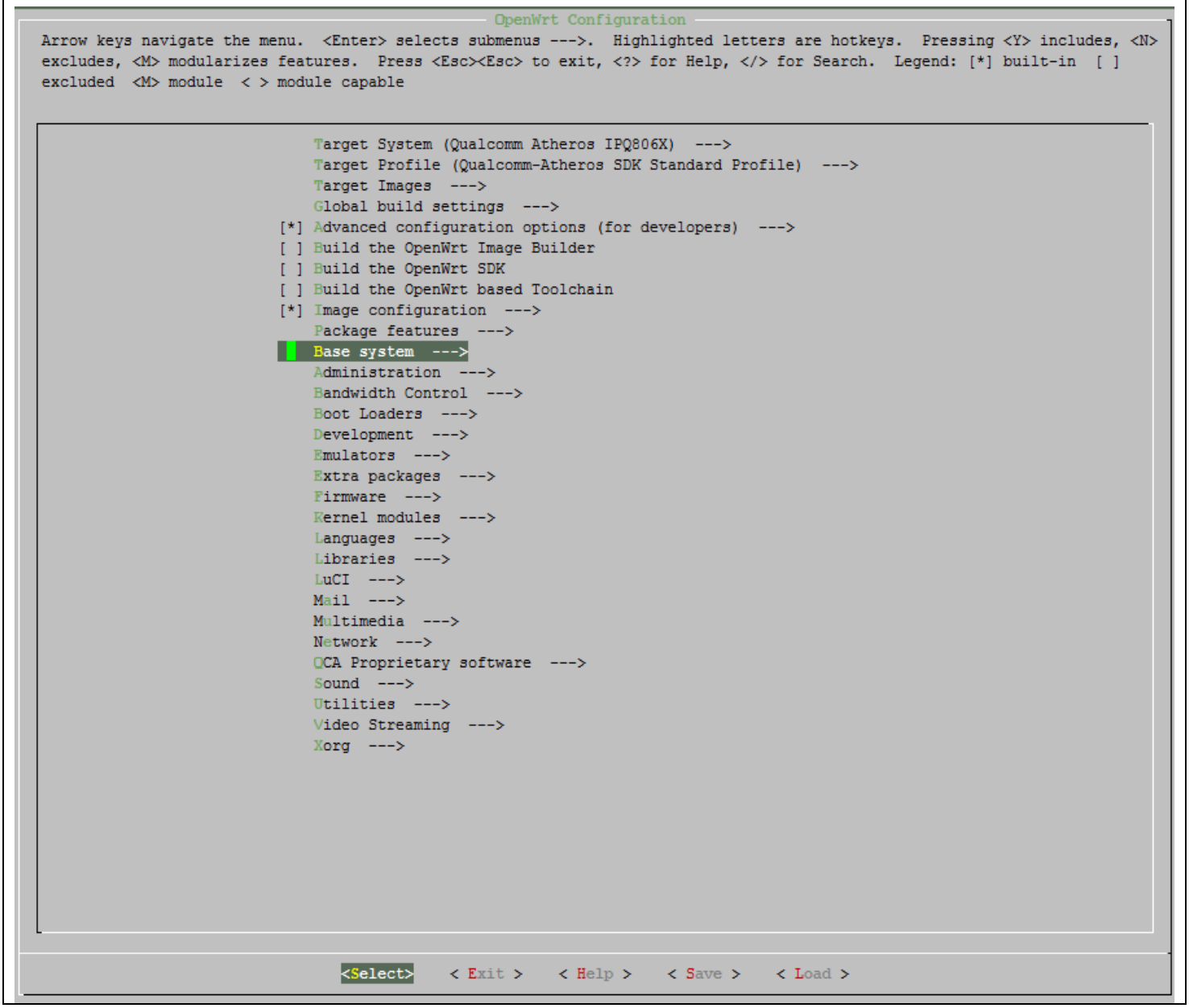


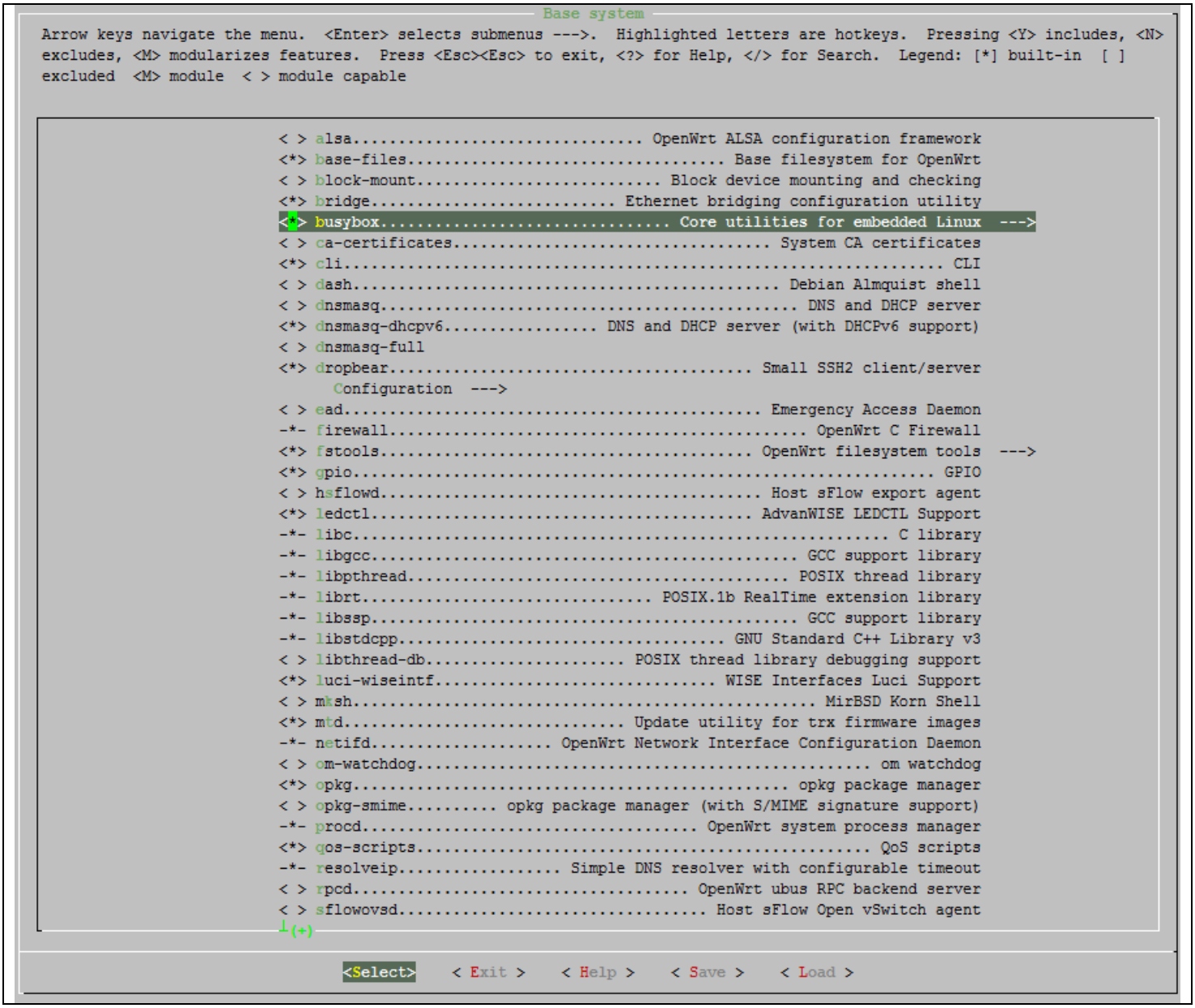
Step 2: The Web GUI will show up checksum for your double confirm. Please press ‘Proceed’ button to flash image to device.

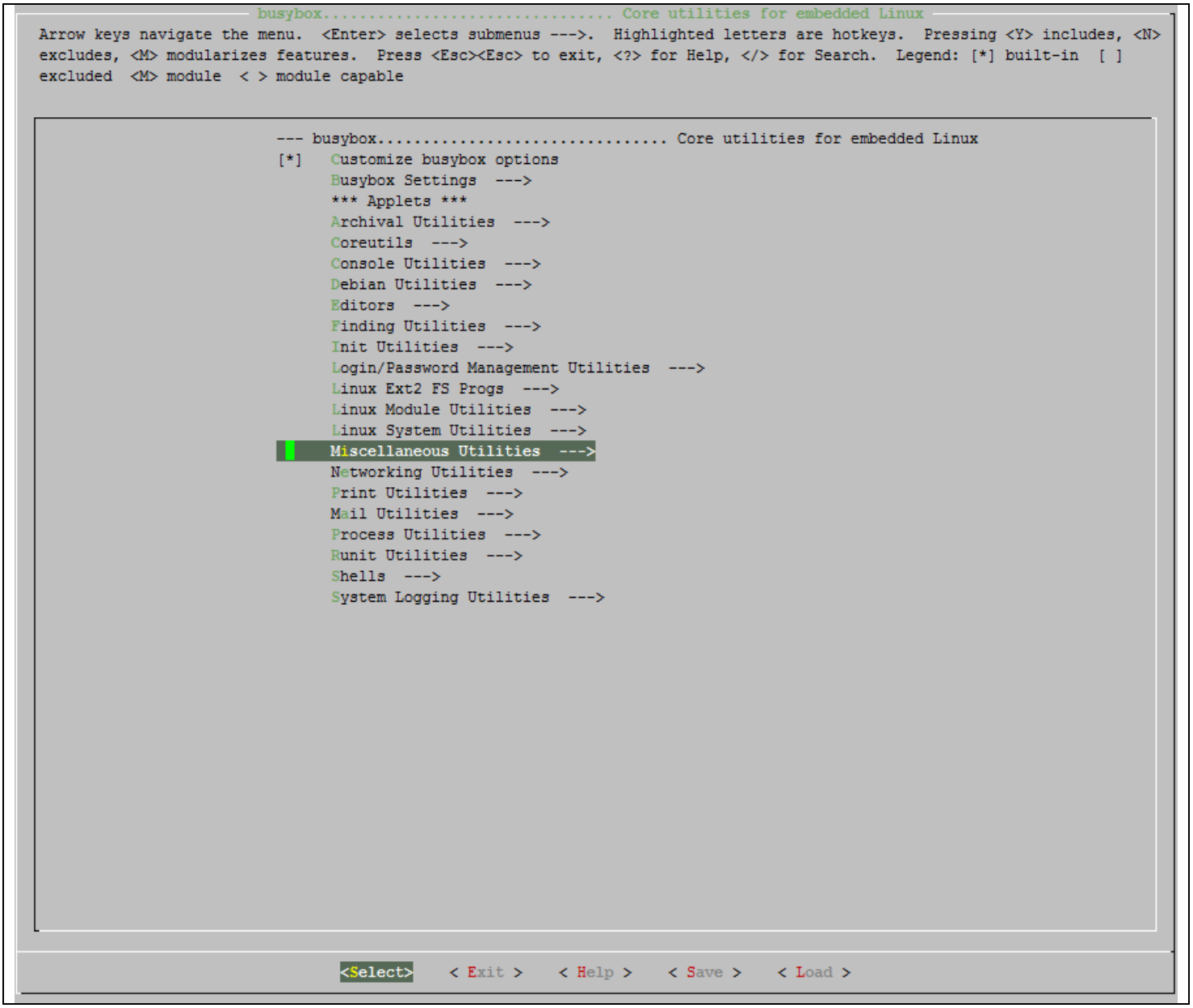


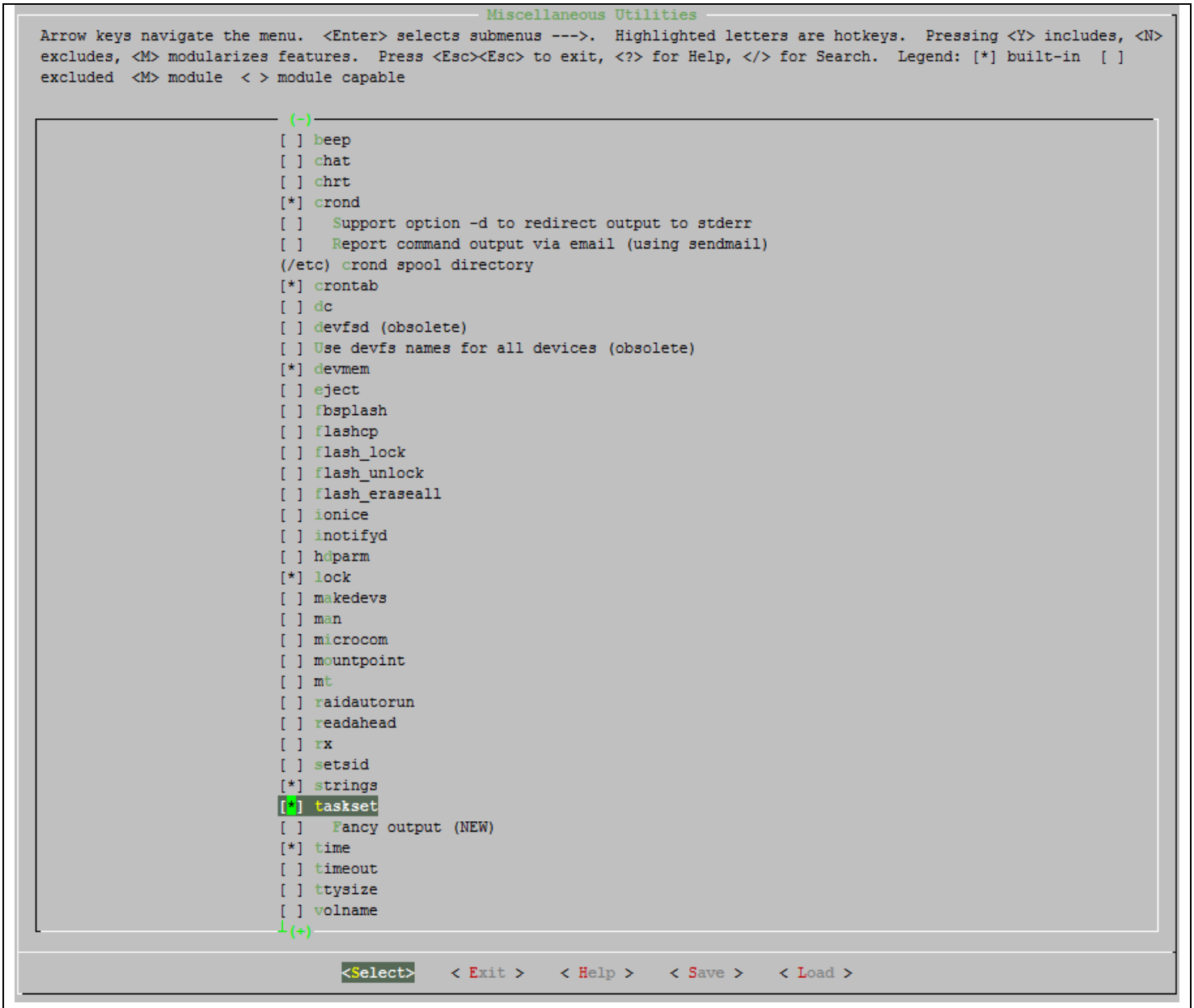
8. To enable taskset in busybox

Dakota$ make menuconfig




Base system
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
< > alsa..... OpenWrt ALSA configuration framework
<*> base-files..... Base filesystem for OpenWrt
< > block-mount..... Block device mounting and checking
<*> bridge..... Ethernet bridging configuration utility
<*> busybox..... Core utilities for embedded Linux --->
< > ca-certificates..... System CA certificates
<*> cli..... CLI
< > dash..... Debian Almquist shell
< > dnsmasq..... DNS and DHCP server
<*> dnsmasq-dhcpv6..... DNS and DHCP server (with DHCPv6 support)
< > dnsmasq-full
<*> dropbear..... Small SSH2 client/server
Configuration --->
< > ead..... Emergency Access Daemon
-*- firewall..... OpenWrt C Firewall
<*> fstools..... OpenWrt filesystem tools --->
<*> gpio..... GPIO
< > hsflowd..... Host sFlow export agent
<*> ledctl..... AdvanWISE LEDCTL Support
-*- libc..... C library
-*- libgcc..... GCC support library
-*- libpthread..... POSIX thread library
-*- librt..... POSIX.1b RealTime extension library
-*- libssp..... GCC support library
-*- libstdcpp..... GNU Standard C++ Library v3
< > libthread-db..... POSIX thread library debugging support
<*> luci-wiseintf..... WISE Interfaces Luci Support
< > mksh..... MirBSD Korn Shell
<*> mtd..... Update utility for trx firmware images
-*- netifd..... OpenWrt Network Interface Configuration Daemon
< > om-watchdog..... om watchdog
<*> opkg..... opkg package manager
< > opkg-smime..... opkg package manager (with S/MIME signature support)
-*- procd..... OpenWrt system process manager
<*> qos-scripts..... QoS scripts
-*- resolveip..... Simple DNS resolver with configurable timeout
< > rpcd..... OpenWrt ubus RPC backend server
< > sflowovsd..... Host sFlow Open vSwitch agent
(+)
<Select> < Exit > < Help > < Save > < Load >


busybox..... Core utilities for embedded Linux
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
--- busybox..... Core utilities for embedded Linux
[*] Customize busybox options
Busybox Settings --->
*** Applets ***
Archival Utilities --->
Coreutils --->
Console Utilities --->
Debian Utilities --->
Editors --->
Finding Utilities --->
Init Utilities --->
Login/Password Management Utilities --->
Linux Ext2 FS Progs --->
Linux Module Utilities --->
Linux System Utilities --->
Miscellaneous Utilities --->
Networking Utilities --->
Print Utilities --->
Mail Utilities --->
Process Utilities --->
Runit Utilities --->
Shells --->
System Logging Utilities --->
<Select> < Exit > < Help > < Save > < Load >



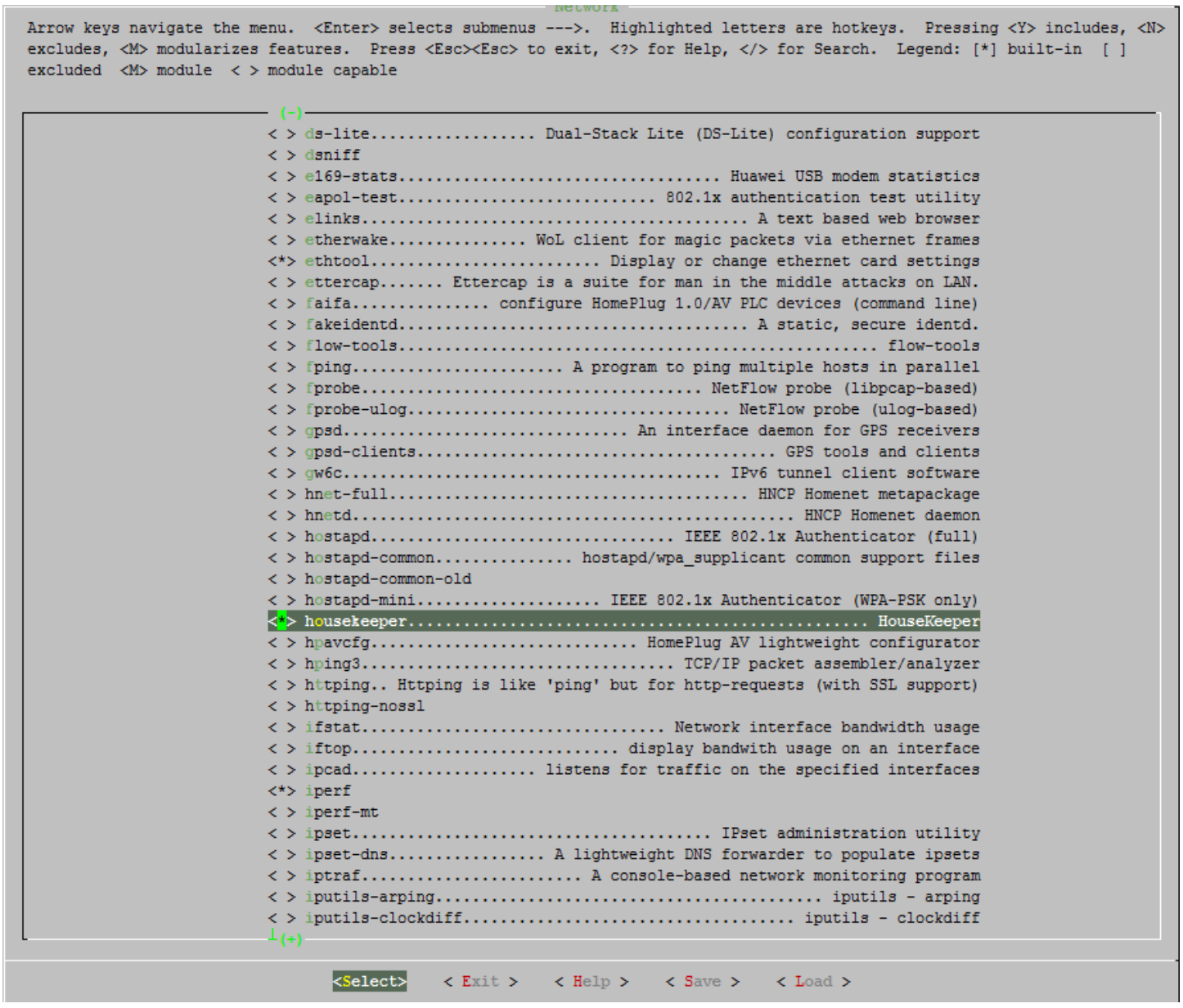
Exit, Save config, and make again to generate new firmware image

Dakota$ make

9. To include a sample task into firmware

To enable housekeeper sample task into firmware, please select the checkbox in Network/housekeeper, and then press exit to save config.

Dakota$ make menuconfig



Exit, Save config, and make again to generate new firmware image

Dakota$ make

……

```
Dakota$ ls -al image/
total 21280
drwxr-xr-x 2 stephen stephen     4096 Mar 10 18:34 .
drwxrwxr-x 4 stephen stephen     4096 Mar 10 18:34 ..
-rw-rw-r-- 1 stephen stephen 21781440 Mar 10 18:34 nand-ipq40xx-single.img
```

- Then, the housekeeper in network/utils will be built into image.

```
Dakota/qsdk/package/network/utils/housekeeper$ ls
files  Makefile  src
```

- OpenWRT description Makefile for housekeeper task

```
Dakota/qsdk/package/network/utils/housekeeper$ cat Makefile
include $(TOPDIR)/rules.mk

PKG_NAME:=housekeeper
PKG_VERSION:=advanwise
PKG_RELEASE=$(PKG_SOURCE_VERSION)

# PKG_SOURCE_PROTO:=git
# PKG_SOURCE_URL:=git://git.kernel.org/pub/scm/linux/kernel/git/shemminger/rstp.git
PKG_SOURCE_SUBDIR:=$(PKG_NAME)-$(PKG_VERSION)

# PKG_SOURCE_VERSION:=434d24bae108dbb21461a13a4abcf014afa8b029
PKG_SOURCE:=$(PKG_NAME)-$(PKG_VERSION).tar.gz
PKG_MAINTAINER:=AdvanWISE
# PKG_MIRROR_MD5SUM:=
# CMAKE_INSTALL:=1

PKG_LICENSE:=GPLv2
PKG_LICENSE_FILES:=

PKG_BUILD_PARALLEL:=1

include $(INCLUDE_DIR)/package.mk

define Package/housekeeper
  SECTION:=net
  CATEGORY:=Network
  MAINTAINER:=AdvanWISE
  URL:=http://www.advantech.com.tw
  TITLE:=HouseKeeper
endef
```

```
# TARGET_CFLAGS += \
#	-I$(STAGING_DIR)/usr/include

define Build/Prepare
	mkdir -p $(PKG_BUILD_DIR)
	$(CP) ./src $(PKG_BUILD_DIR)/
	$(CP) ./files/Makefile $(PKG_BUILD_DIR)/Makefile
endef

define Package/housekeeper/install/SDKDefault
	$(INSTALL_DIR) $(1)/usr/sbin/
	$(INSTALL_BIN) $(PKG_BUILD_DIR)/src/captain $(1)/usr/sbin/

#	$(INSTALL_DIR) $(1)/etc/init.d
#	$(INSTALL_BIN) ./files/housekeeper.init $(1)/etc/init.d/housekeeper
endef

define Package/housekeeper/install
	$(INSTALL_DIR) $(1)/usr/sbin/
	$(INSTALL_BIN) $(PKG_BUILD_DIR)/src/captain $(1)/usr/sbin/

#	$(INSTALL_DIR) $(1)/etc/init.d
#	$(INSTALL_BIN) ./files/housekeeper.init $(1)/etc/init.d/housekeeper

	$(call Package/housekeeper/install/SDKDefault, "$(TOPDIR)/../sdklib/sdkrootfs")
endef

$(eval $(call BuildPackage,housekeeper))
```

- .c/.h Makefile for housekeeper task

Dakota/qsdk/package/network/utils/housekeeper$ cat files/Makefile

```
all:
	echo -e "\033[32m Make HouseKeeper ...\033[0m"
	make -C src
	echo -e "\033[32m Make HouseKeeper Done ...\033[0m"


clean:
	echo -e "\033[32mCleaning HouseKeeper ...\033[0m"
	rm -f src/captain
	rm -f src/*.o
```

- .c/.h for housekeeper task

```
Dakota/qsdk/package/network/utils/housekeeper$ ls -al src/
total 20
drwxrwxr-x 3 stephen stephen 4096 Mar 10 17:43 .
drwxrwxr-x 4 stephen stephen 4096 Mar 10 17:43 ..
drwxrwxr-x 2 stephen stephen 4096 Mar 10 17:43 bin
-rwxrwxr-x 1 stephen stephen  956 Mar 10 17:43 captain.c
-rwxrwxr-x 1 stephen stephen  534 Mar 10 17:43 Makefile
```

- The real built copy is at build_dir

```
Dakota/qsdk/build_dir/target-arm_cortex-a7_uClibc-0.9.33.2_eabi/housekeeper-advanwise$ ls
ipkg-ipq806x  Makefile  src
```

10. Another Application Example

To add an app into QSDK, you can also refer CLI module.

1. Create a directory in qsdk/package/system/
   Dakota/qsdk/package/system/cli

2. Add a Makefile to describe the OpenWRT package
   Dakota/qsdk/package/system/cli$ ls Makefile
   Makefile

   define Package/$(PKG_NAME)
   endef

   define Build/Prepare
   endef

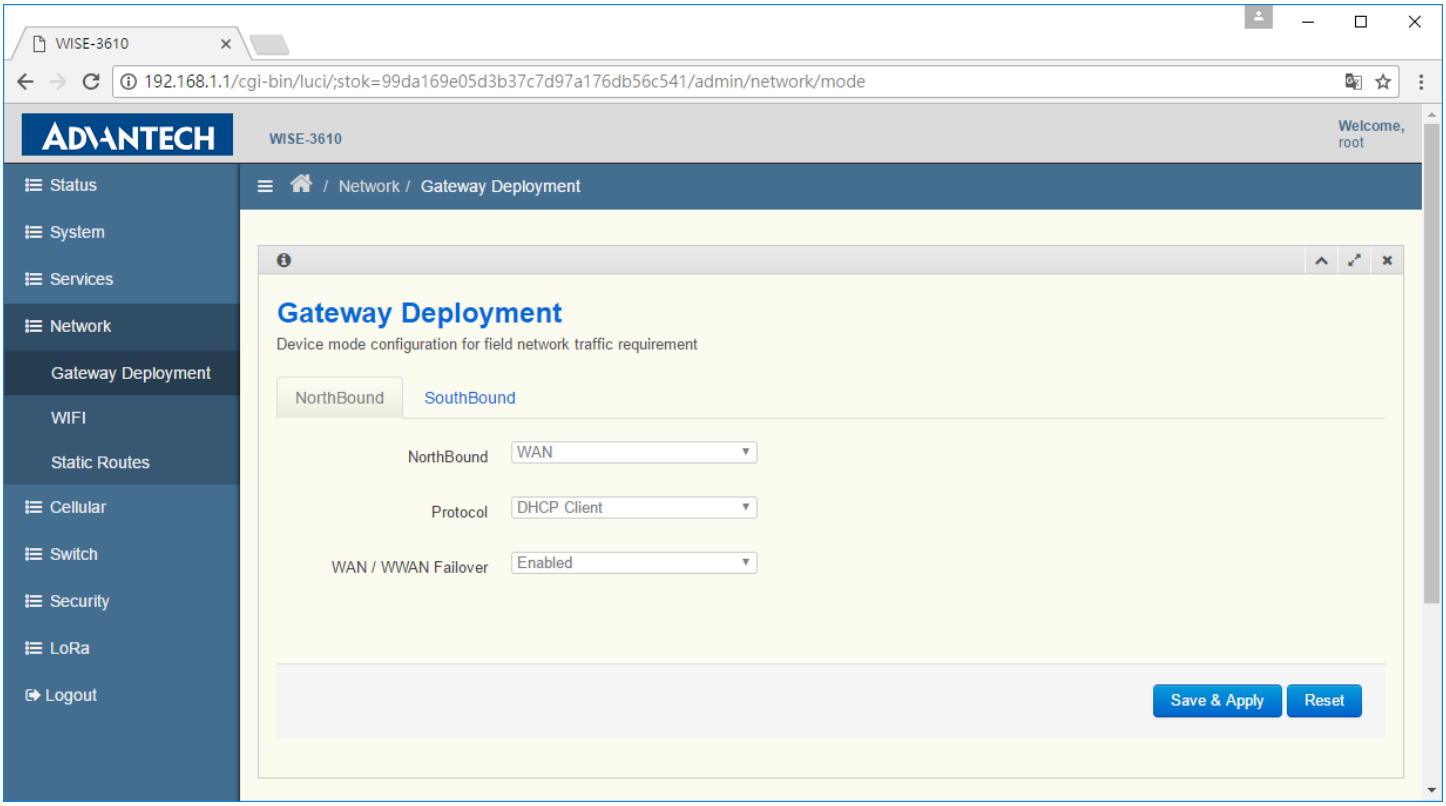
   define Package/$(PKG_NAME)/install
   endef

3. In source directory, we can see all .c, .h and Makefile
   Dakota/qsdk/package/system/cli/cli-advanwise$ ls
   Makefile source

   Dakota/qsdk/package/system/cli/cli-advanwise/source/src$ ls
   awcli.c cmd.c wificmd.c

11. Add a sample Webpage

- Device Mode Web GUI



- Controller File

Dakota/qsdk/package/system/luci-wiseintf$ cat luasrc/controller/wiseintf.lua

```
module("luci.controller.wiseintf", package.seeall)

function index()
        if not nixio.fs.access("/etc/config/network") then
                return
        end

        local page0 = entry({"admin", "network", "mode"}, cbi("wiseintf/devicemode"), _("Device Mode"), 0)
        page0.dependent = true
……
```

```
end
```

- Lua Webpage

Dakota/qsdk/package/system/luci-wiseintf$ cat luasrc/model/cbi/wiseintf/devicemode.lua

```
-- organization
s = m:section(NamedSection, "specify", "scenario")
s.anonymous = true

s:tab("backhaul", translate("NorthBound"))
s:tab("lan", translate("SouthBound"))
s:tab("wifi", translate("WIFI"))

-------------------
------- wan -------
-------------------
-- device mode
local mode = s:taboption("backhaul", ListValue, "mode", translate("NorthBound"))
mode.default = "1"
mode:value("1", translate("WAN"))
mode:value("2", translate("WWAN"))
mode:value("3", translate("2.4GHz WIFI"))
mode:value("4", translate("5GHz WIFI"))
……
```

12. Interface between Gateway Platform and Module Board

- ttyACM0~ACM5 for Telit module and ttyACM6 for LoRa module

```
root@WISE-3610:/# ls -al /dev/ttyACM
ttyACM0  ttyACM1  ttyACM2  ttyACM3  ttyACM4  ttyACM5  ttyACM6
```

- Config Baud Rate

```
……
        case   300:  baud = B300;
                  break;
        case  1200:  baud = B1200;
                  break;
        case  2400:  baud = B2400;
                  break;
        case  4800:  baud = B4800;
                  break;
        case  9600:  baud = B9600;
                  break;
        case 19200:  baud = B19200;
                  break;
        case 38400:  baud = B38400;
                  break;
        case 115200: baud = B115200;
                  break;
……
   if (baud > 0)
   {
     term.c_cflag &= ~CBAUD;
     term.c_cflag |= baud;
   }
……
ioctl(fd, TCSETA, &term);
……
```

- Read/Write tty Device

```
……
   if ((fdr = fopen(argv[1], "r")) == NULL )
   {
     perror(argv[1]);
     exit(errno);
   }
   if ((fdw = fopen(argv[1], "w")) == NULL )
   {
     perror(argv[1]);
     exit(errno);
   }
……
write(fileno(fdw), buffer, strlen(buffer));
……
num = read(fileno(fdr), buffer, sizeof(buffer));
```

13. Configuration Management

- Configuration Files

```
root@WISE-3610:/etc/config# ls
cellular      loraserver    php5-fastcgi   rstp          thermal
customer      luci          polipo         samba         ubootenv
ddns          mcproxy       pppoe          skb_recycler  ucitrack
dhcp          mcsd          pptpd          snmpd         uhttpd
dropbear      mosquitto     pure-ftpd      ssid-steering wireless
firewall      multiwan      qos            switch        wsplcd
ipsec         network       radvd          syslog
lbd           nss           repacd         system
```

- File Content

```
root@WISE-3610:/etc/config# cat network
.....
config interface 'lan'
        option ifname 'eth1'
        option type 'bridge'
        option proto 'static'
        option ipaddr '192.168.1.1'
        option netmask '255.255.255.0'

config interface 'wan'
        option ifname 'eth0'
        option proto 'dhcp'
......
```

- Management Tool

```
root@WISE-3610:/etc/config# uci
Usage: uci [<options>] <command> [<arguments>]

Commands:
        batch
        export     [<config>]
        import     [<config>]
        changes    [<config>]
        commit     [<config>]
        add        <config> <section-type>
```

```
        add_list   <config>.<section>.<option>=<string>
        del_list   <config>.<section>.<option>=<string>
        show       [<config>[.<section>[.<option>]]]
        get        <config>.<section>[.<option>]
        set        <config>.<section>[.<option>]=<value>
        delete     <config>[.<section>[[.<option>][=<id>]]]
        rename     <config>.<section>[.<option>]=<name>
        revert     <config>[.<section>[.<option>]]
        reorder    <config>.<section>=<position>

Options:
        -c <path>  set the search path for config files (default: /etc/config)
        -d <str>   set the delimiter for list values in uci show
        -f <file>  use <file> as input instead of stdin
        -m         when importing, merge data into an existing package
        -n         name unnamed sections on export (default)
        -N         don't name unnamed sections
        -p <path>  add a search path for config change files
        -P <path>  add a search path for config change files and use as default
        -q         quiet mode (don't print error messages)
        -s         force strict mode (stop on parser errors, default)
        -S         disable strict mode
        -X         do not use extended syntax on 'show'

root@WISE-3610:/etc/config# uci show network
network.loopback=interface
network.loopback.ifname='lo'
network.loopback.proto='static'
network.loopback.ipaddr='127.0.0.1'
network.loopback.netmask='255.0.0.0'
network.lan=interface
network.lan.ifname='eth1'
network.lan.type='bridge'
network.lan.proto='static'
network.lan.ipaddr='192.168.1.1'
network.lan.netmask='255.255.255.0'
network.wan=interface
network.wan.ifname='eth0'
network.wan.proto='dhcp'
network.@switch[0]=switch
network.@switch[0].name='switch0'
network.@switch[0].reset='1'
network.@switch[0].enable_vlan='1'
```

```
network.@switch_vlan[0]=switch_vlan
network.@switch_vlan[0].device='switch0'
network.@switch_vlan[0].vlan='1'
network.@switch_vlan[0].ports='0t 1 2 3'
network.@switch_vlan[1]=switch_vlan
network.@switch_vlan[1].device='switch0'
network.@switch_vlan[1].vlan='2'
network.@switch_vlan[1].ports='0t 4'
network.@switch_ext[0]=switch_ext
network.@switch_ext[0].device='switch0'
network.@switch_ext[0].name='QosPtMode'
network.@switch_ext[0].port_id='1'
network.@switch_ext[0].mode='dscp'
network.@switch_ext[0].status='enable'
network.@switch_ext[1]=switch_ext
network.@switch_ext[1].device='switch0'
network.@switch_ext[1].name='QosPtMode'
network.@switch_ext[1].port_id='2'
network.@switch_ext[1].mode='dscp'
network.@switch_ext[1].status='enable'
network.@switch_ext[2]=switch_ext
network.@switch_ext[2].device='switch0'
network.@switch_ext[2].name='QosPtMode'
network.@switch_ext[2].port_id='3'
network.@switch_ext[2].mode='dscp'
network.@switch_ext[2].status='enable'
network.@switch_ext[3]=switch_ext
network.@switch_ext[3].device='switch0'
network.@switch_ext[3].name='QosPtMode'
network.@switch_ext[3].port_id='4'
network.@switch_ext[3].mode='dscp'
network.@switch_ext[3].status='enable'
network.@switch_ext[4]=switch_ext
network.@switch_ext[4].device='switch0'
network.@switch_ext[4].name='QosPtMode'
network.@switch_ext[4].port_id='5'
network.@switch_ext[4].mode='dscp'
network.@switch_ext[4].status='enable'
network.cellular=interface
network.cellular.ifname='wwan0'
network.cellular.proto='dhcp'
network.wwan11ng=interface
network.wwan11ng.ifname='ath0'
```

```
network.wwan11ng.proto='dhcp'
network.wwan11ac=interface
network.wwan11ac.ifname='ath1'
network.wwan11ac.proto='dhcp'
network.vpn=interface
network.vpn.proto='l2tp'
network.vpn.server='192.168.107.192'
network.vpn.username='username'
network.vpn.password='password'
root@WISE-3610:/etc/config#
root@WISE-3610:/etc/config# uci show
……
```

- Bring up scripts

```
root@WISE-3610:/etc/init.d# ls
avahi-daemon      loraserver       qca-nss-drv      sysfixtime
boot              luci_dhcp_migrate  qcmbr          sysinfo
cellular          luci_fixtime     qos              syslog
cellular_led      mcproxy          qrfs             syslog-ng
cellular_loop     mcsd             radvd            sysntpd
cron              mosquitto        repacd           sysstat
ddns              multiwan         rngd             system
dnsmasq           network          rstp             telnet
done              odhcpd           samba            thermal
dropbear          php5-fastcgi     shortcut-fe      uhttpd
firewall          polipo           skb_recycler     umount
gpioInit          powerctl         snmpd            wise-mgmt
ipsec             pppoe-relay      ssid_steering    wise-mgmt.bak
lbd               pptpd            storage-mgmt     wise_snail_agent
led               pure-ftpd        switch           wsplcd
log               qca-edma         sysctl           xl2tpd
```

14. WIFI Command Interface

- WIFI Command Interface

```
root@WISE-3610:/# iwconfig --help
Usage: iwconfig [interface]
          interface essid {NNN|any|on|off}
          interface mode {managed|ad-hoc|master|...}
          interface freq N.NNN[k|M|G]
          interface channel N
          interface bit {N[k|M|G]|auto|fixed}
          interface rate {N[k|M|G]|auto|fixed}
          interface enc {NNNN-NNNN|off}
          interface key {NNNN-NNNN|off}
          interface power {period N|timeout N|saving N|off}
          interface ap {N|off|auto}
          interface txpower {NmW|NdBm|off|auto}
          interface rts {N|auto|fixed|off}
          interface frag {N|auto|fixed|off}
          interface commit
     Check man pages for more details.
root@WISE-3610:/# iwpriv --help
Usage: iwpriv interface [private-command [private-arguments]]
root@WISE-3610:/# wlanconfig
usage: wlanconfig athX create wlandev wifiX
           wlanmode
[sta|adhoc|ap|monitor|wrap|p2pgo|p2pcli|p2pdev|specialvap|mesh|smart_monitor|lp_iot_mode]
           [wlanaddr <mac_addr>] [mataddr <mac_addr>] [bssid|-bssid] [nosbeacon]
usage: wlanconfig athX destroy
usage: wlanconfig athX nawds mode (0-4)
usage: wlanconfig athX nawds defcaps CAPS
usage: wlanconfig athX nawds override (0-1)
usage: wlanconfig athX nawds add-repeater MAC (0-1)
usage: wlanconfig athX nawds del-repeater MAC
usage: wlanconfig athX nawds list
usage: wlanconfig athX hmwds add-addr wds_ni_macaddr wds_macaddr
usage: wlanconfig athX hmwds reset-addr macaddr
```

```
usage: wlanconfig athX hmwds reset-table
usage: wlanconfig athX hmwds read-addr wds_ni_macaddr
usage: wlanconfig athX hmwds read-table
usage: wlanconfig athX ald sta-enable <sta_mac_addr> <0/1>
usage: wlanconfig athX hmmc add ip mask
usage: wlanconfig athX hmmc del ip mask
usage: wlanconfig athX hmmc dump
usage: wlanconfig athX wnm setbssmax <idle period in seconds> [<idle option>]
usage: wlanconfig athX wnm getbssmax
usage: wlanconfig athX wnm tfsreq <filename>
usage: wlanconfig athX wnm deltfs
usage: wlanconfig athX wnm timintvl <Interval>
usage: wlanconfig athX wnm gettimparams
usage: wlanconfig athX wnm timrate <highrateEnable> <lowRateEnable>
usage: wlanconfig athX wnm bssterm <delay in TBTT> [<duration in minutes>]
usage: wlanconfig athX addssid ssidname per_value(0--100)
usage: wlanconfig athX addsta  macaddr(example:112233445566) per_value(0--100)
usage: wlanconfig athX delssid ssidname
usage: wlanconfig athX delsta  macaddr
usage: wlanconfig athX showatftable
usage: wlanconfig athX showairtime
usage: wlanconfig athX flushatftable
usage: wlanconfig athX addatfgroup groupname ssid
usage: wlanconfig athX configatfgroup groupname value (0 - 100))
usage: wlanconfig athX delatfgroup groupname
usage: wlanconfig athX showatfgroup
usage: wlanconfig athX addtputsta macaddr tput airtime(opt)
usage: wlanconfig athX deltputsta macaddr
usage: wlanconfig athX showtputtbl
usage: wlanconfig athX vendorie add len <oui+pcap_data in bytes> oui <eg:xxxxxx> pcap_data <eg:xxxxxxxx>
ftype_map <eg:xx>
usage: wlanconfig athX vendorie update len <oui+pcap_data in bytes> oui <eg:xxxxxx> pcap_data
<eg:xxxxxxxx> ftype_map <eg:xx>
usage: wlanconfig athX vendorie remove len <oui+pcap_data in bytes> oui <eg:xxxxxx> pcap_data <eg:xx>
usage: wlanconfig athX vendorie list
usage: wlanconfig athX vendorie list len <oui in bytes> oui <eg:xxxxxx>
usage: wlanconfig athX nac add/del bssid <ad1 eg: xx:xx:xx:xx:xx:xx> <ad2> <ad3>
usage: wlanconfig athX nac add/del client <ad1 eg: xx:xx:xx:xx:xx:xx> <ad2> <ad3> <ad4> <ad5>  <ad6>
<ad7>  <ad8>
usage: wlanconfig athX nac list bssid/client
root@WISE-3610:/#
```

- WIFI Bring Up Scripts

```
root@WISE-3610:/# ls -al /lib/wifi/
drwxr-xr-x    2 root     root           283 Mar 10 07:35 .
drwxr-xr-x   14 root     root           957 Mar 10 07:05 ..
-rw-r--r--    1 root     root         25530 Mar 10 07:15 hostapd.sh
-rwxr-xr-x    1 root     root          1041 Mar 10 07:14 icm.sh
-rw-r--r--    1 root     root         24286 Mar 10 07:35 qcawifi.sh
-rw-r--r--    1 root     root          1641 Mar 10 07:35 qcawifi_countrycode.txt
-rw-r--r--    1 root     root          5427 Mar 10 07:15 qwrap.sh
-rw-r--r--    1 root     root         12448 Mar 10 07:35 wifidevice.sh
-rw-r--r--    1 root     root         24873 Mar 10 07:35 wifiinterface.sh
-rw-r--r--    1 root     root          1294 Mar 10 07:35 wifimacfilter.sh
-rw-r--r--    1 root     root          7646 Mar 10 07:15 wpa_supplicant.sh
-rwxr-xr-x    1 root     root           518 Mar 10 07:14 wpc.sh
-rwxr-xr-x    1 root     root          2606 Mar 10 07:15 wps-hostapd-update-uci
-rwxr-xr-x    1 root     root          4779 Mar 10 07:15 wps-supplicant-update-uci
```

15. To build program with ToolChain directly

The toolchain is located at staging_dir, and you can copy it out for future development

qsdk$ ls -al staging_dir/
host/ target-arm_cortex-a7_uClibc-0.9.33.2_eabi/ toolchain-arm_cortex-a7_gcc-4.8-linaro_uClibc-0.9.33.2_eabi/

At host Linux PC, we can create a hello world sample program

```
stephen@AdvanWISE-YG-409:~$ mkdir testToolChain
stephen@AdvanWISE-YG-409:~$ cd testToolChain/
stephen@AdvanWISE-YG-409:~/testToolChain$ ls
stephen@AdvanWISE-YG-409:~/testToolChain$ vim main.c
stephen@AdvanWISE-YG-409:~/testToolChain$
stephen@AdvanWISE-YG-409:~/testToolChain$ /opt/toolchainsNew/toolchain-arm_cortex-a7_gcc-4.8-linaro_uClibc-0.9.33.2_eabi/bin/arm-openwrt-linux-gcc main.c -o main.o
arm-openwrt-linux-gcc: warning: environment variable 'STAGING_DIR' not defined
arm-openwrt-linux-gcc: warning: environment variable 'STAGING_DIR' not defined
arm-openwrt-linux-gcc: warning: environment variable 'STAGING_DIR' not defined
stephen@AdvanWISE-YG-409:~/testToolChain$ ls -al
total 20
drwxrwxr-x  2 stephen stephen 4096 Apr  7 11:32 .
drwxr-xr-x 80 stephen stephen 4096 Apr  7 11:31 ..
-rw-rw-r--  1 stephen stephen   73 Apr  7 11:31 main.c
-rwxrwxr-x  1 stephen stephen 5809 Apr  7 11:32 main.o
```

At WISE-3610 board, we can run this program directly.

```
root@WISE-3610:/# tftp -r main.o -g 192.168.1.100
root@WISE-3610:/# chmod +x main.o
```

```
root@WISE-3610:/# ./main.o
hello world !
root@WISE-3610:/#
```

16. Reproduce SDK

After user add their own module and want to release SDK, please introduce following commands

Dakota$ make clean
……

To remove un-necessary files
Dakota$ rm -rf error image

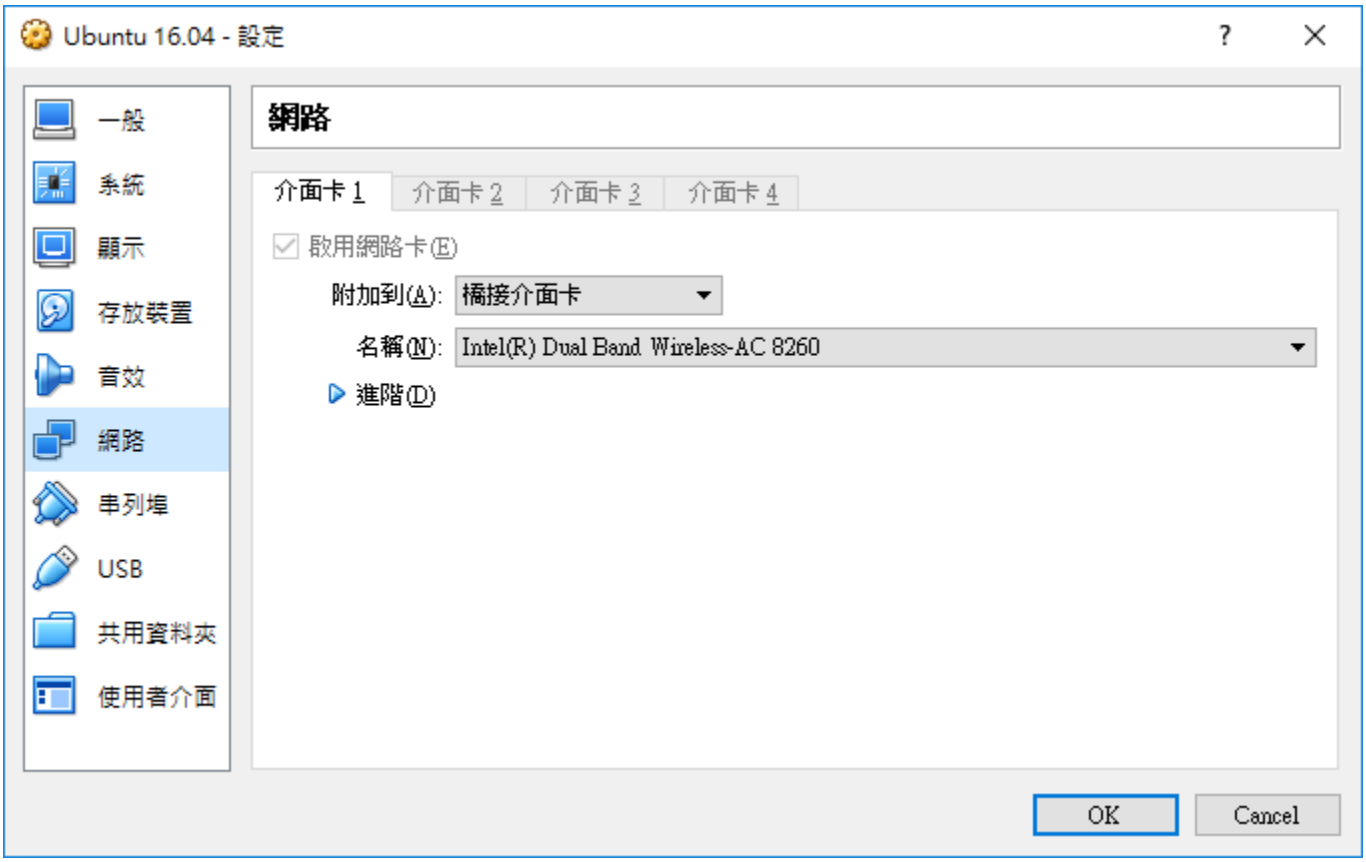
And then tar the SDK source files into Dakota.tar.bz2
Dakota$ cd ..
$ tar jcvf Dakota.tar.bz2 Dakota/

17. Virtual Box + Ubuntu 16.04

The hint to setup compiler server using Virtual Box + Ubuntu 16.04

1. Download Ubuntu 16.04 Server Version :

2. To let Ubuntu 16.04 Linux Guest System be able to connect to outside network



3. Reference Commands

steven@steven-VirtualBox:~$ sudo apt-get install openssh-server

steven@steven-VirtualBox:~$ sudo apt-get install samba

steven@steven-VirtualBox:~$ sudo apt-get install vim

steven@steven-VirtualBox:~$ sudo vim /etc/samba/smb.conf

```
[homes]
  comment = Home Directories
  browseable = no

# By default, the home directories are exported read-only. Change the
# next parameter to 'no' if you want to be able to write to them.
  read only = yes
```

steven@steven-VirtualBox:~$ sudo service smbd restart

steven@steven-VirtualBox:~$ sudo smbpasswd -a steven
New SMB password:
Retype new SMB password:
Added user steven.
steven@steven-VirtualBox:~$

steven@steven-VirtualBox:~$ sudo apt-get install gcc g++ binutils patch autoconf libcurl4-openssl-dev bzip2 flex make gettext pkg-config unzip zlib1g-dev libc6-dev subversion libncurses5-dev gawk sharutils curl libxml-parser-perl python-yaml git ocaml-nox ocaml ocaml-findlib bison texinfo ncurses-term zlib1g-dev openssl libssl-dev u-boot-tools device-tree-compiler git git-core curl phablet-tools

18. Build SDK with ARM Feature Requirement

Step 1: Find specific SDK codebase with ARM keyword in file name, such as WISE-3610_ARM_SDK_20171031_d207f81.tar.bz2

Step 2: tar jxvf WISE-3610_ARM_SDK_20171031_d207f81.tar.bz2

Step 3: cd Dakota

Step 4: make

Step 5: Refer Section 5 in this document to download firmware image to device. Because the ARM partition requirement are different, user has to follow Section 5 when changing firmware between different partition layout. (WEB GUI cannot be used to load firmware with different partition format)

Note: Normal Partition Layout

```
==Boot Log==
[    0.942930] Creating 13 MTD partitions on "7980000.qcom,nand":
[    0.948768] 0x000000000000-0x000000100000 : "0:SBL1"
[    0.955523] 0x000000100000-0x000000200000 : "0:MIBIB"
[    0.961276] 0x000000200000-0x000000300000 : "0:BOOTCONFIG"
[    0.967509] 0x000000300000-0x000000400000 : "0:QSEE"
[    0.973217] 0x000000400000-0x000000500000 : "0:QSEE_ALT"
[    0.979172] 0x000000500000-0x000000580000 : "0:CDT"
[    0.984348] 0x000000580000-0x000000600000 : "0:CDT_ALT"
[    0.989857] 0x000000600000-0x000000680000 : "0:DDRPARAMS"
[    0.995585] 0x000000680000-0x000000700000 : "0:APPSBLENV"
[    1.001149] 0x000000700000-0x000000900000 : "0:APPSBL"
[    1.007563] 0x000000900000-0x000000b00000 : "0:APPSBL_ALT"
[    1.014363] 0x000000b00000-0x000000b80000 : "0:ART"
[    1.019473] 0x000000b80000-0x000004c80000 : "rootfs"

==Linux Runtime==
root@WISE-3610:/# cat /proc/mtd
dev:    size   erasesize  name
mtd0: 00100000 00020000 "0:SBL1"
mtd1: 00100000 00020000 "0:MIBIB"
mtd2: 00100000 00020000 "0:BOOTCONFIG"
mtd3: 00100000 00020000 "0:QSEE"
mtd4: 00100000 00020000 "0:QSEE_ALT"
```

```
mtd5: 00080000 00020000 "0:CDT"
mtd6: 00080000 00020000 "0:CDT_ALT"
mtd7: 00080000 00020000 "0:DDRPARAMS"
mtd8: 00080000 00020000 "0:APPSBLENV"
mtd9: 00200000 00020000 "0:APPSBL"
mtd10: 00200000 00020000 "0:APPSBL_ALT"
mtd11: 00080000 00020000 "0:ART"
mtd12: 04100000 00020000 "rootfs"
mtd13: 003a2000 0001f000 "kernel"
mtd14: 01059000 0001f000 "ubi_rootfs"
mtd15: 02815000 0001f000 "rootfs_data"
root@WISE-3610:/#
```

Note: ARM Required Partition Layout

```
==Boot Log==
[    0.937002] 19 ofpart partitions found on MTD device 7980000.qcom,nand
[    0.943411] Creating 19 MTD partitions on "7980000.qcom,nand":
[    0.949218] 0x000000000000-0x000000100000 : "0:SBL1"
[    0.956003] 0x000000100000-0x000000200000 : "0:MIBIB"
[    0.961754] 0x000000200000-0x000000300000 : "0:BOOTCONFIG"
[    0.968002] 0x000000300000-0x000000400000 : "0:QSEE"
[    0.973719] 0x000000400000-0x000000500000 : "0:QSEE_ALT"
[    0.979651] 0x000000500000-0x000000580000 : "0:CDT"
[    0.984833] 0x000000580000-0x000000600000 : "0:CDT_ALT"
[    0.990342] 0x000000600000-0x000000680000 : "0:DDRPARAMS"
[    0.996069] 0x000000680000-0x000000700000 : "0:APPSBLENV"
[    1.001633] 0x000000700000-0x000000900000 : "0:APPSBL"
[    1.008058] 0x000000900000-0x000000b00000 : "0:APPSBL_ALT"
[    1.014839] 0x000000b00000-0x000000b80000 : "0:ART"
[    1.019961] 0x000000b80000-0x000000bc0000 : "FLAGS_0"
[    1.025111] 0x000000bc0000-0x000000c00000 : "FLAGS_1"
[    1.030232] 0x000000c00000-0x000003c00000 : "rootfs"
[    1.083769] 0x000003c00000-0x000004400000 : "empty_0"
[    1.095326] 0x000004400000-0x000007400000 : "rootfs_1"
[    1.133640] 0x000007400000-0x000007c00000 : "empty_1"
[    1.144147] 0x000007c00000-0x000008000000 : "KCM"

==Linux Runtime==
root@WISE-3610:/# cat /proc/mtd
dev:    size   erasesize  name
mtd0: 00100000 00020000 "0:SBL1"
mtd1: 00100000 00020000 "0:MIBIB"
mtd2: 00100000 00020000 "0:BOOTCONFIG"
```

```
mtd3: 00100000 00020000 "0:QSEE"
mtd4: 00100000 00020000 "0:QSEE_ALT"
mtd5: 00080000 00020000 "0:CDT"
mtd6: 00080000 00020000 "0:CDT_ALT"
mtd7: 00080000 00020000 "0:DDRPARAMS"
mtd8: 00080000 00020000 "0:APPSBLENV"
mtd9: 00200000 00020000 "0:APPSBL"
mtd10: 00200000 00020000 "0:APPSBL_ALT"
mtd11: 00080000 00020000 "0:ART"
mtd12: 00040000 00020000 "FLAGS_0"
mtd13: 00040000 00020000 "FLAGS_1"
mtd14: 03000000 00020000 "rootfs"
mtd15: 00800000 00020000 "empty_0"
mtd16: 03000000 00020000 "rootfs_1"
mtd17: 00800000 00020000 "empty_1"
mtd18: 00400000 00020000 "KCM"
mtd19: 003a2000 0001f000 "kernel"
mtd20: 0101b000 0001f000 "ubi_rootfs"
mtd21: 019ea000 0001f000 "rootfs_data"
root@WISE-3610:/#
```

Note: Success Changing Log

```
MM   MMMM    MMMM
 MMMMMMM:     MMMMMMM     M       MMMMMMMMMMMM  MMMMMMM MMMMMMM
   MMMMMM      MMMMN     M         MMMMMMMMM      MMMM    MMMM
    MMMM        M               MMMMMMM       M      M
     M
 ---------------------------------------------------------------
  For those about to rock... (R1.0.16)
 ---------------------------------------------------------------
root@WISE-3610:/#
root@WISE-3610:/# cat /proc/mtd
dev:    size   erasesize  name
mtd0: 00100000 00020000 "0:SBL1"
mtd1: 00100000 00020000 "0:MIBIB"
mtd2: 00100000 00020000 "0:BOOTCONFIG"
mtd3: 00100000 00020000 "0:QSEE"
mtd4: 00100000 00020000 "0:QSEE_ALT"
mtd5: 00080000 00020000 "0:CDT"
mtd6: 00080000 00020000 "0:CDT_ALT"
mtd7: 00080000 00020000 "0:DDRPARAMS"
mtd8: 00080000 00020000 "0:APPSBLENV"
mtd9: 00200000 00020000 "0:APPSBL"
```

```
mtd10: 00200000 00020000 "0:APPSBL_ALT"
mtd11: 00080000 00020000 "0:ART"
mtd12: 04100000 00020000 "rootfs"
mtd13: 003a2000 0001f000 "kernel"
mtd14: 01059000 0001f000 "ubi_rootfs"
mtd15: 02815000 0001f000 "rootfs_data"
root@WISE-3610:/#
root@WISE-3610:/# reboot
procd: - reboot -
root@WISE-3610:/# [ 3225.971709] reboot: Restarting system

Format: Log Type - Time(microsec) - Message - Optional Info
Log Type: B - Since Boot(Power On Reset),  D - Delta,  S - Statistic
S - QC_IMAGE_VERSION_STRING=BOOT.BF.3.1.1-00096
S - IMAGE_VARIANT_STRING=DAACANAZA
S - OEM_IMAGE_VERSION_STRING=CRM
S - Boot Config, 0x00000025
S - Core 0 Frequency, 0 MHz

••••••

U-Boot 2012.07 [WISE-3610 R1.0.16,unknown] (Nov 01 2017 - 17:07:44)

smem ram ptable found: ver: 1 len: 3
DRAM:  256 MiB
machid : 0x8010001
NAND:  ONFI device found
ID = 9580f12c
Vendor = 2c
Device = f1
SF NAND unsupported id:ff:ff:ff:ffSF: Unsupported manufacturer ff
ipq_spi: SPI Flash not found (bus/cs/speed/mode) = (0/0/48000000/0)
128 MiB
MMC:   qca_mmc: 0
*** Warning - bad CRC, using default environment

In:    serial
Out:   serial
Err:   serial
machid: 8010001
flash_type: 2
Net:   MAC0 addr:0:3:7f:ba:db:ad
PHY ID1: 0x4d
PHY ID2: 0xd0b1
```

```
ipq40xx_ess_sw_init done
eth0
Hit any key to stop autoboot:  0
(IPQ40xx) # set ipaddr 192.168.1.1
(IPQ40xx) # set serverip 192.168.1.100
(IPQ40xx) # tftpboot nand-ipq40xx-single.img
eth0 PHY0 Down Speed :10 Half duplex
eth0 PHY1 Down Speed :10 Half duplex
eth0 PHY2 up Speed :1000 Full duplex
eth0 PHY3 Down Speed :10 Half duplex
eth0 PHY4 Down Speed :10 Half duplex
Using eth0 device
TFTP from server 192.168.1.100; our IP address is 192.168.1.1
Filename 'nand-ipq40xx-single.img'.
Load address: 0x84000000
Loading: ###################################################################
         ###################################################################

••••••
         ##################################################
done
Bytes transferred = 23223232 (1625bc0 hex)
(IPQ40xx) # imgaddr=0x84000000 && source $imgaddr:script && reset
## Executing script at 84000000
crc32+ Flashing mibib:                  ## Copying 'mibib-fb94cca75b16a5a04cae01227af254a0e9039bf8'
subimage from FIT image at 84000000 ...
crc32+
NAND erase: device 0 offset 0x100000, size 0x100000
Erasing at 0x1e0000 -- 100% complete.
OK

NAND write: device 0 offset 0x100000, size 0x40000
 262144 bytes written: OK
[ done ]
Flashing sbl1:                  ## Copying 'sbl1-73fb8022f5abb040c722a5d4674591b6463cfa1a' subimage from
FIT image at 84000000 ...
crc32+
NAND erase: device 0 offset 0x0, size 0x100000
Erasing at 0xe0000 -- 100% complete.
OK

NAND write: device 0 offset 0x0, size 0x25000
 151552 bytes written: OK
[ done ]
```

```
Flashing ddr-AP-DK04.1-C1:          ## Copying 'ddr-AP-DK04.1-C1-
44f7cf880531f125fc2394a28013813eb1a756e5' subimage from FIT image at 84000000 ...
crc32+
NAND erase: device 0 offset 0x500000, size 0x80000
Erasing at 0x560000 -- 100% complete.
OK

NAND write: device 0 offset 0x500000, size 0x800
 2048 bytes written: OK
[ done ]
Flashing tz:                    ## Copying 'tz-7fb7fc3700e39853414a46c5956c80067bd3af08' subimage from FIT
image at 84000000 ...
crc32+
NAND erase: device 0 offset 0x300000, size 0x100000
Erasing at 0x3e0000 -- 100% complete.
OK

NAND write: device 0 offset 0x300000, size 0x51800
 333824 bytes written: OK
[ done ]
Flashing u-boot:                ## Copying 'u-boot-ee4297641e8ac05e0faa79f61de22344c4258284' subimage
from FIT image at 84000000 ...
crc32+
NAND erase: device 0 offset 0x700000, size 0x200000
Erasing at 0x8e0000 -- 100% complete.
OK

NAND write: device 0 offset 0x700000, size 0x6e000
 450560 bytes written: OK
[ done ]
Flashing ubi:                   ## Copying 'ubi-2113e3f3cc2a94e31f40d2c220669cca1b7e2845' subimage from
FIT image at 84000000 ...
crc32+
NAND erase: device 0 offset 0xc00000, size 0x3000000
Erasing at 0x3be0000 -- 100% complete.
OK

NAND write: device 0 offset 0xc00000, size 0x14e0000
 21889024 bytes written: OK
[ done ]
resetting ...

Format: Log Type - Time(microsec) - Message - Optional Info
```

```
Log Type: B - Since Boot(Power On Reset),  D - Delta,  S - Statistic
S - QC_IMAGE_VERSION_STRING=BOOT.BF.3.1.1-00096
S - IMAGE_VARIANT_STRING=DAACANAZA
S - OEM_IMAGE_VERSION_STRING=CRM
S - Boot Config, 0x00000025
S - Core 0 Frequency, 0 MHz
B -       261 - PBL, Start
B -      1338 - bootable_media_detect_entry, Start
B -      2609 - bootable_media_detect_success, Start
B -      2623 - elf_loader_entry, Start
B -      4029 - auth_hash_seg_entry, Start
B -      6180 - auth_hash_seg_exit, Start
B -     73748 - elf_segs_hash_verify_entry, Start
B -    194116 - PBL, End
B -    194140 - SBL1, Start
B -    282775 - pm_device_init, Start
D -         6 - pm_device_init, Delta
B -    284300 - boot_flash_init, Start
D -     84651 - boot_flash_init, Delta
B -    372999 - boot_config_data_table_init, Start
D -     13976 - boot_config_data_table_init, Delta - (419 Bytes)
B -    389670 - clock_init, Start
D -      7583 - clock_init, Delta
B -    400657 - CDT version:2,Platform ID:8,Major ID:1,Minor ID:0,Subtype:1
B -    404142 - sbl1_ddr_set_params, Start
B -    409127 - cpr_init, Start
D -         2 - cpr_init, Delta
B -    413616 - Pre_DDR_clock_init, Start
D -         5 - Pre_DDR_clock_init, Delta
D -     13140 - sbl1_ddr_set_params, Delta
B -    426895 - pm_driver_init, Start
D -         2 - pm_driver_init, Delta
B -    497361 - sbl1_wait_for_ddr_training, Start
D -        27 - sbl1_wait_for_ddr_training, Delta
B -    512876 - Image Load, Start
D -    140632 - QSEE Image Loaded, Delta - (262104 Bytes)
B -    654004 - Image Load, Start
D -      2115 - SEC Image Loaded, Delta - (2048 Bytes)
B -    664157 - Image Load, Start
D -    176366 - APPSBL Image Loaded, Delta - (417791 Bytes)
B -    840944 - QSEE Execution, Start
D -        56 - QSEE Execution, Delta
B -    847206 - SBL1, End
```

```
D -    655037 - SBL1, Delta
S - Flash Throughput, 2074 KB/s  (682362 Bytes,  328930 us)
S - DDR Frequency, 672 MHz


U-Boot 2012.07 [WISE-3610 R1.0.16,unknown] (Nov 01 2017 - 17:57:11)

smem ram ptable found: ver: 1 len: 3
DRAM:  256 MiB
machid : 0x8010001
NAND:  ONFI device found
ID = 9580f12c
Vendor = 2c
Device = f1
SF NAND unsupported id:ff:ff:ff:ffSF: Unsupported manufacturer ff
ipq_spi: SPI Flash not found (bus/cs/speed/mode) = (0/0/48000000/0)
128 MiB
MMC:   qca_mmc: 0
*** Warning - bad CRC, using default environment

In:    serial
Out:   serial
Err:   serial
machid: 8010001
flash_type: 2
Net:   MAC0 addr:0:3:7f:ba:db:ad
PHY ID1: 0x4d
PHY ID2: 0xd0b1
ipq40xx_ess_sw_init done
eth0
Hit any key to stop autoboot:  0
Creating 1 MTD partitions on "nand0":
0x000000c00000-0x000003c00000 : "mtd=0"
UBI: attaching mtd1 to ubi0
UBI: physical eraseblock size:   131072 bytes (128 KiB)
UBI: logical eraseblock size:    126976 bytes
UBI: smallest flash I/O unit:    2048
UBI: VID header offset:          2048 (aligned 2048)
UBI: data offset:                4096
UBI: volume 2 ("rootfs_data") re-sized from 1 to 212 LEBs
UBI: attached mtd1 to ubi0
UBI: MTD device name:            "mtd=0"
UBI: MTD device size:            48 MiB
```

```
UBI: number of good PEBs:       384
UBI: number of bad PEBs:        0
UBI: max. allowed volumes:      128
UBI: wear-leveling threshold:   4096
UBI: number of internal volumes: 1
UBI: number of user volumes:    3
UBI: available PEBs:            0
UBI: total number of reserved PEBs: 384
UBI: number of PEBs reserved for bad PEB handling: 3
UBI: max/mean erase counter: 1/0
Read 0 bytes from volume kernel to 84000000
No size specified -> Using max size (3809280)
## Booting kernel from FIT Image at 84000000 ...
   Using 'config@1' configuration
   Trying 'kernel@1' kernel subimage
     Description:  ARM OpenWrt Linux-3.14.43
     Type:         Kernel Image
     Compression:  gzip compressed
     Data Start:   0x840000e4
     Data Size:    3310771 Bytes = 3.2 MiB
     Architecture: ARM
     OS:           Linux
     Load Address: 0x80208000
     Entry Point:  0x80208000
     Hash algo:    crc32
     Hash value:   63106529
     Hash algo:    sha1
     Hash value:   147a66a19c9338cc3c4f6d2efce97e199391a25b
   Verifying Hash Integrity ... crc32+ sha1+ OK
## Flattened Device Tree from FIT Image at 84000000
   Using 'config@1' configuration
   Trying 'fdt@1' FDT blob subimage
     Description:  ARM OpenWrt qcom-ipq40xx-ap.dkxx device tree blob
     Type:         Flat Device Tree
     Compression:  uncompressed
     Data Start:   0x843286d8
     Data Size:    36585 Bytes = 35.7 KiB
     Architecture: ARM
     Hash algo:    crc32
     Hash value:   416ea2a0
     Hash algo:    sha1
     Hash value:   241e87160dca86ed44b97c25906db08fb06f7bdc
   Verifying Hash Integrity ... crc32+ sha1+ OK
```

```
   Booting using the fdt blob at 0x843286d8
   Uncompressing Kernel Image ... OK
   Loading Device Tree to 86ff4000, end 86fffee8 ... OK
Device nand2 not found!
eth0 MAC Address from ART is not valid
eth1 MAC Address from ART is not valid
Using machid 0x8010001 from environment

Starting kernel ...

[    0.000000] Booting Linux on physical CPU 0x0

••••••
 [    0.936748] 19 ofpart partitions found on MTD device 7980000.qcom,nand
[    0.943143] Creating 19 MTD partitions on "7980000.qcom,nand":
[    0.948983] 0x000000000000-0x000000100000 : "0:SBL1"
[    0.955749] 0x000000100000-0x000000200000 : "0:MIBIB"
[    0.961504] 0x000000200000-0x000000300000 : "0:BOOTCONFIG"
[    0.967742] 0x000000300000-0x000000400000 : "0:QSEE"
[    0.973448] 0x000000400000-0x000000500000 : "0:QSEE_ALT"
[    0.979391] 0x000000500000-0x000000580000 : "0:CDT"
[    0.984568] 0x000000580000-0x000000600000 : "0:CDT_ALT"
[    0.990079] 0x000000600000-0x000000680000 : "0:DDRPARAMS"
[    0.995805] 0x000000680000-0x000000700000 : "0:APPSBLENV"
[    1.001369] 0x000000700000-0x000000900000 : "0:APPSBL"
[    1.007781] 0x000000900000-0x000000b00000 : "0:APPSBL_ALT"
[    1.014574] 0x000000b00000-0x000000b80000 : "0:ART"
[    1.019690] 0x000000b80000-0x000000bc0000 : "FLAGS_0"
[    1.024844] 0x000000bc0000-0x000000c00000 : "FLAGS_1"
[    1.029971] 0x000000c00000-0x000003c00000 : "rootfs"
[    1.067993] mtd: device 14 (rootfs) set to be root filesystem
[    1.072938] mtdsplit: no squashfs found in "rootfs"
[    1.077811] mtdsplit: no squashfs found in "7980000.qcom,nand"
[    1.083405] 0x000003c00000-0x000004400000 : "empty_0"
[    1.094965] 0x000004400000-0x000007400000 : "rootfs_1"
[    1.133147] 0x000007400000-0x000007c00000 : "empty_1"
[    1.143676] 0x000007c00000-0x000008000000 : "KCM"

••••••
BusyBox v1.22.1 (2017-11-01 18:52:27 CST) built-in shell (ash)
Enter 'help' for a list of built-in commands.

   MM         NM                MMMMMMMM        M      M
  $MMMMMM      MMMMMM            MMMMMMMMMMMMM     MMM     MMM
```

```
 MMMMMMMMM    MM MMMMM.          MMMMM:MMMMMM:  MMMM  MMMMM
MMMM= MMMMMM  MMM  MMMM     MMMMM  MMMM MMMMMM  MMMM MMMMM'
MMMM=  MMMMM MMMM   MM     MMMMM   MMMM   MMMM  MMMMNMMMMM
MMMM=  MMMM  MMMMM       MMMMM   MMMM   MMMM  MMMMMMMM
MMMM=  MMMM  MMMMMM     MMMMM    MMMM   MMMM  MMMMMMMMM
MMMM=  MMMM   MMMMM,  NMMMMMMMM  MMMM   MMMM  MMMMMMMMMM
MMMM=  MMMM    MMMMMM MMMMMMMM   MMMM   MMMM  MMMMMM
MMMM=  MMMM  MM   MMMM   MMMM    MMMM   MMMM  MMMM
MMMM$ ,MMMMM MMMMM MMMM   MMM     MMMM  MMMM  MMMM   MMMM
 MMMMMMM:    MMMMMMM   M      MMMMMMMMMMMM MMMMMMM MMMMMMM
  MMMMMM     MMMMN   M       MMMMMMMMM    MMMM   MMMM
   MMMM       M             MMMMMMM      M     M
    M
 ------------------------------------------------------------
  For those about to rock... (R1.0.16)
 ------------------------------------------------------------
root@WISE-3610:/# cat /proc/mtd
dev:    size   erasesize  name
mtd0: 00100000 00020000 "0:SBL1"
mtd1: 00100000 00020000 "0:MIBIB"
mtd2: 00100000 00020000 "0:BOOTCONFIG"
mtd3: 00100000 00020000 "0:QSEE"
mtd4: 00100000 00020000 "0:QSEE_ALT"
mtd5: 00080000 00020000 "0:CDT"
mtd6: 00080000 00020000 "0:CDT_ALT"
mtd7: 00080000 00020000 "0:DDRPARAMS"
mtd8: 00080000 00020000 "0:APPSBLENV"
mtd9: 00200000 00020000 "0:APPSBL"
mtd10: 00200000 00020000 "0:APPSBL_ALT"
mtd11: 00080000 00020000 "0:ART"
mtd12: 00040000 00020000 "FLAGS_0"
mtd13: 00040000 00020000 "FLAGS_1"
mtd14: 03000000 00020000 "rootfs"
mtd15: 00800000 00020000 "empty_0"
mtd16: 03000000 00020000 "rootfs_1"
mtd17: 00800000 00020000 "empty_1"
mtd18: 00400000 00020000 "KCM"
mtd19: 003a2000 0001f000 "kernel"
mtd20: 01059000 0001f000 "ubi_rootfs"
mtd21: 019ac000 0001f000 "rootfs_data"
root@WISE-3610:/# WLAN 2 interfaces not ready,
```

Step 6: The Web Account is **admin/mbedcloud**