Telechips TCCxxxx Android Quick Boot

Kitkat_QuickBoot

Rev. 1.03.6

07 Mar, 2014

Telechips

DISCLAIMER

All information and data contained in this material are without any commitment, are not to be considered as an offer for conclusion of a contract, nor shall they be construed as to create any liability. Any new issue of this material invalidates previous issues. Product availability and delivery are exclusively subject to our respective order confirmation form; the same applies to orders based on development samples delivered. By this publication, Telechips, Inc. does not assume responsibility for patent infringements or other rights of third parties that may result from its use.

Further, Telechips, Inc. reserves the right to revise this publication and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes.
No part of this publication may be reproduced, photocopied, stored on a retrieval system, or transmitted without the express written consent of Telechips, Inc.
This product is designed for general purpose, and accordingly customer be responsible for all or any of intellectual property licenses required for actual application. Telechips, Inc. does not provide any indemnification for any intellectual properties owned by third party.
Telechips, Inc. can not ensure that this application is the proper and sufficient one for any other purposes but the one explicitly expressed herein. Telechips, Inc. is not responsible for any special, indirect, incidental or consequential damage or loss whatsoever resulting from the use of this application for other purposes.

COPYRIGHT STATEMENT

Copyright in the material provided by Telechips, Inc. is owned by Telechips unless otherwise noted.
For reproduction or use of Telechips' copyright material, permission should be sought from Telechips. That permission, if given, will be subject to conditions that Telechips' name should be included and interest in the material should be acknowledged when the material is reproduced or quoted, either in whole or in part. You must not copy, adapt, publish, distribute or commercialize any contents contained in the material in any manner without the written permission of Telechips.
Trade marks used in Telechips' copyright material are the property of Telechips.

Important Notice

This product may include technology owned by Microsoft Corporation and in this case it cannot be used or distributed without a license from Microsoft Licensing, GP.

For customers who use licensed Codec ICs and/or licensed codec firmware of mp3:
"Supply of this product does not convey a license nor imply any right to distribute content created with this product in revenue-generating broadcast systems (terrestrial. Satellite, cable and/or other distribution channels), streaming applications(via internet, intranets and/or other networks), other content distribution systems(pay-audio or audio-on-demand applications and the like) or on physical media(compact discs, digital versatile discs, semiconductor chips, hard drives, memory cards and the like). An independent license for such use is required. For details, please visit http://mp3licensing.com".

For customers who use other firmware of mp3:
"Supply of this product does not convey a license under the relevant intellectual property of Thomson and/or Fraunhofer Gesellschaft nor imply any right to use this product in any finished end user or ready-to-use final product. An independent license for such use is required. For details, please visit http://mp3licensing.com".

For customers who use Digital Wave DRA solution:
"Supply of this implementation of DRA technology does not convey a license nor imply any right to this implementation in any finished end-user or ready-to-use terminal product. An independent license for such use is required."

For customers who use DTS technology:
"Supply of this implementation of DTS technology does not convey a license, exhaust DTS' rights in the implementation, or imply a right under any patent, or any other industrial or intellectual property right of DTS to use, offer for sale, sell, or import such implementation in any finished end-user or ready-to-use final product. Notice is hereby provided that a license from DTS is required prior to such use."
"This product made under license to U.S. Patents 5,451,942; 5,956,674; 5,974,380; 5,978,762; 6,487,535; 6,226,616 and/or foreign counterparts."
"© 1996 – 2010 DTS, Inc."

For customers who use Dolby technology:
"Supply of this Implementation of Dolby technology does not convey a license nor imply a right under any patent, or any other industrial or intellectual property right of Dolby Laboratories, to use this Implementation in any finished end-user or ready-to-use final product. It is hereby notified that a license for such use is required from Dolby Laboratories."

Revision History

Date	Version	Description
2013-03-19	0.01	This document is a guide to the Quick Boot of jellybean. Initial release.
2013-06-21	0.02	Add “How to register new service”
2013-07-04	0.03	Modify “How to register new service”
2013-08-05	0.04	Add display last frame buffer image when boot with snapshot
2013-08-13	1.02	Add “How to prebuilt “apk” file on system application” Add [*]Watchdog Timer Support & [*]TCC Watchdog in kernel configuration
2013-09-23	1.02.1	Modify “How to register new service” and “Quickboot logo change”
2013-10-11	1.02.2	Modify “6.6 Displaying an image during making snapshot image” and “6.7 Displaying an image during quickboot” Add “How to use boot chart”
2013-10-29	1.02.3	Add “How to built-in services on quick boot image”
2013-11-04	1.02.4	Modify diagrams to indicate sequence of SMP Enable & Disable
2013-11-29	1.03	Modify “How to built-in services on quick boot image” Quick boot v1.03
2013-11-29	1.03.1	Quickboot pre-built guide is added
2013-12-03	1.03.2	Partition configuration guide of quickboot pre-built is updated
2014-01-21	1.03.3	Increase snapshot partition size.
2014-02-10	1.03.4	Add “How to get QuickBoot Time"
2014-02-17	1.03.5	Update “QuickBoot” for Kitkat.
2014-03-07	1.03.6	Modify “BeforeQBSystem Callback” & “Device Driver” for QuickBoot 1.03.

TABLE OF CONTENTS

Contents

1 Introduction

1.1 Overview

Telechips quick-boot solution is base on snapshot boot. Snapshot boot is a resume-from-disk operation, which is a system resume from semi permanent snapshot image that restores the machine to a known running state. Snapshot boot is based on the current software suspend technology in Linux kernel.

There are 3 suspend states in the linux kernel

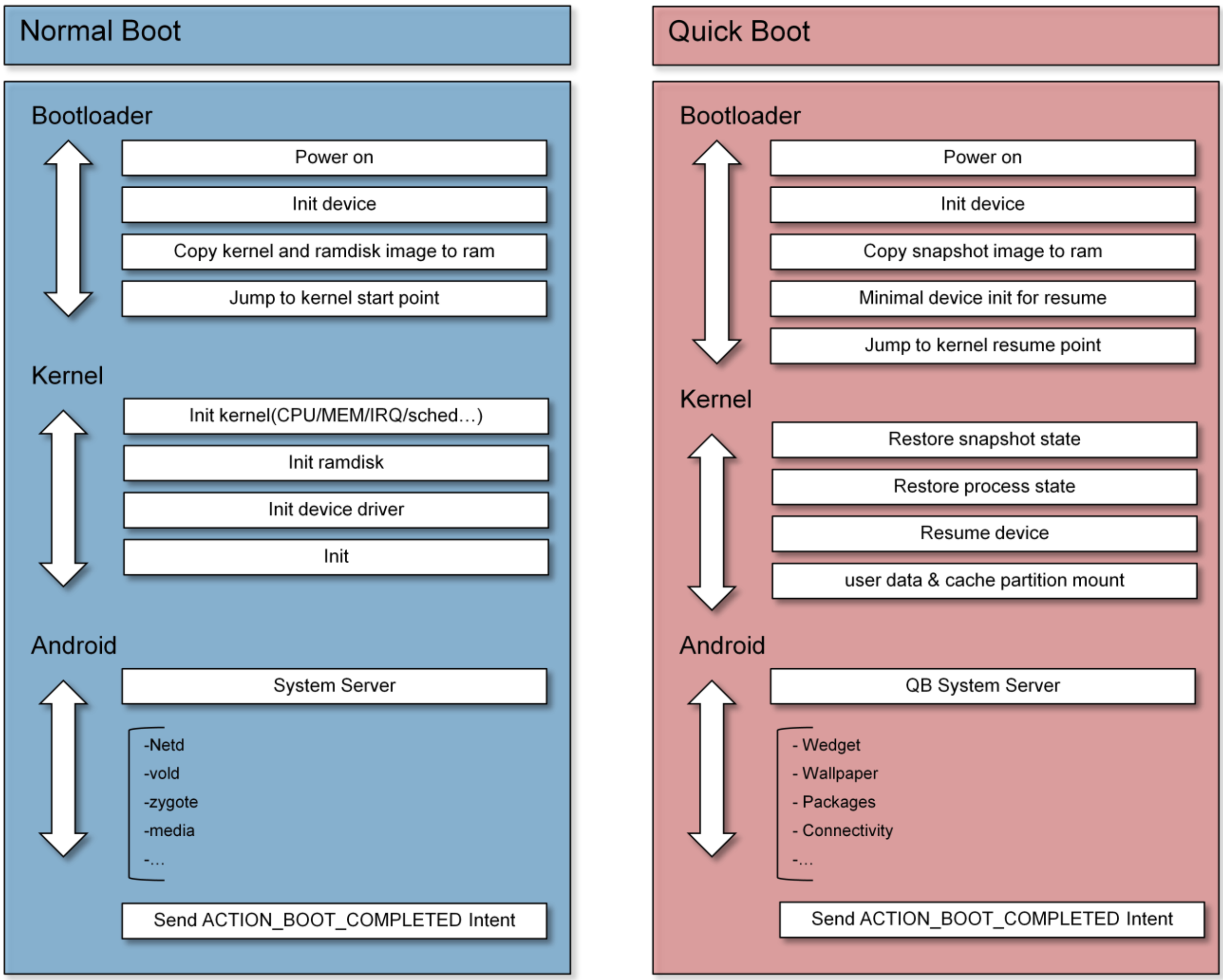
1) Standby State
2) Suspend to RAM State
3) Suspend to Disk State(Hibernation)

Snapshot boot uses hibernation.

To improve start-up time, snapshot image is created only once, started on flash memory, and the same image is used repeatedly.

1.2 Difference between snapshot boot and normal boot

You can understand the differences between normal boot and snapshot boot by checking following two figures.
These figures show the start-up process of normal boot and snapshot boot



2 How to include quick boot feature

2.1 Patch kk-enable-quickboot.patch in order to enable quick boot

The quick boot is base on kitkat official release

The lunch supporting snapshot boot is as follows.
full_tcc8920st-eng
full_tcc8920-eng
full_tcc8930st-eng
full_tcc893x-eng

The quick boot requires about 200MB(200000KB) in order to save snapshot image.

The following shows how to enable the quick boot.

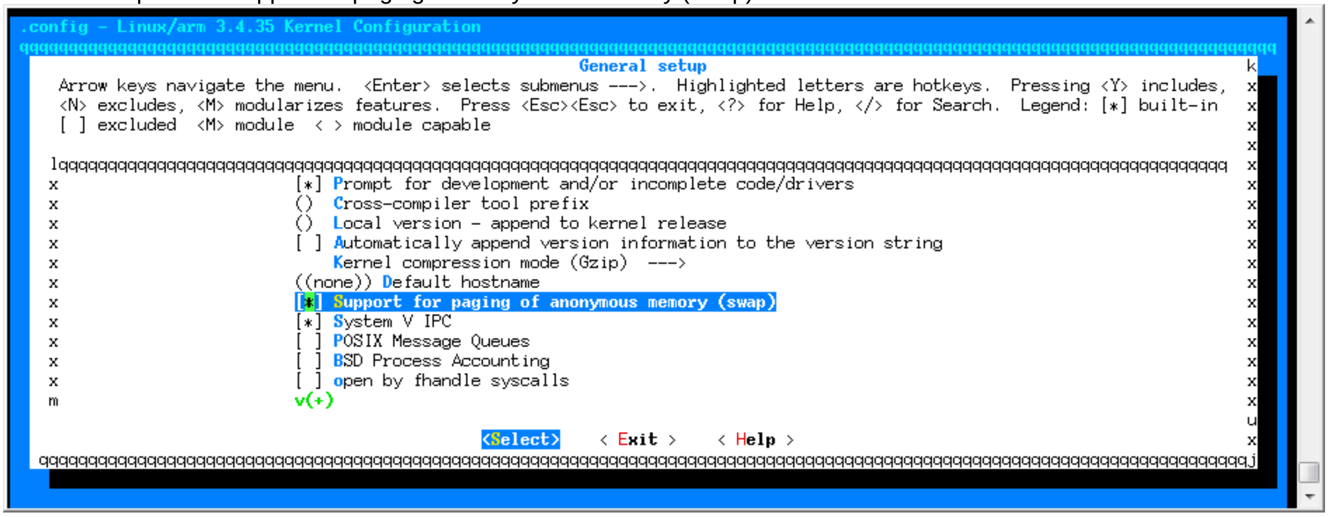
1) cd your kitkat project
2) Execute “patch –p1 < device/telechips/common/kk-enable-quickboot.patch” in shell

2.2 Enable quick boot feature in kernel configuration

The followings are how to set kernel configuration in order to use the snapshot boot.

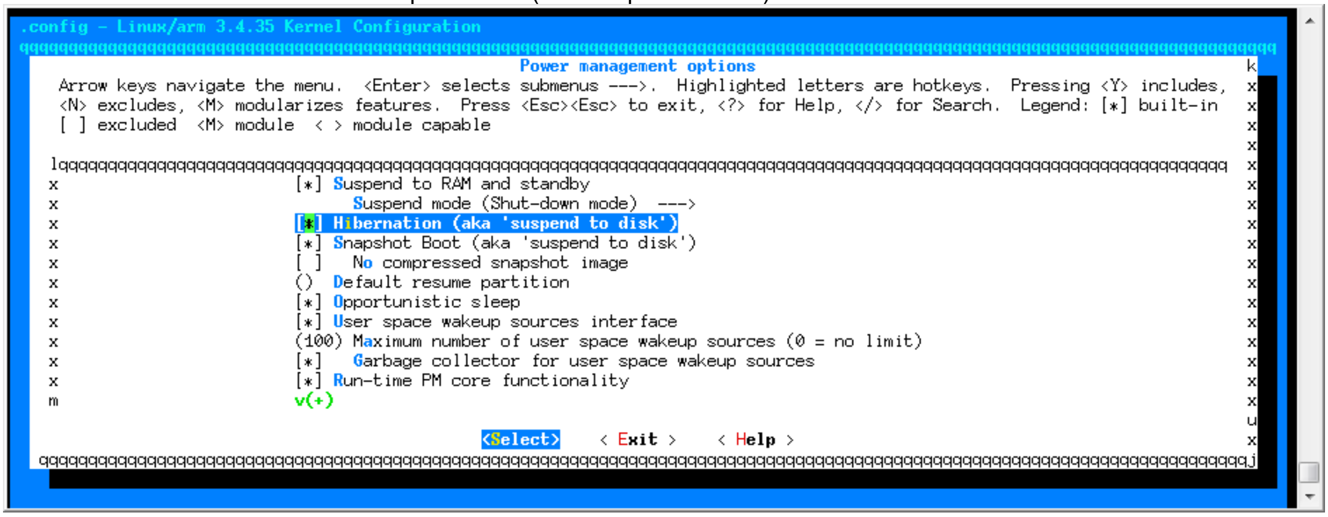
$cd kernel
$make menuconfig

General setup → <*> support for paging of anonymous memory (swap)



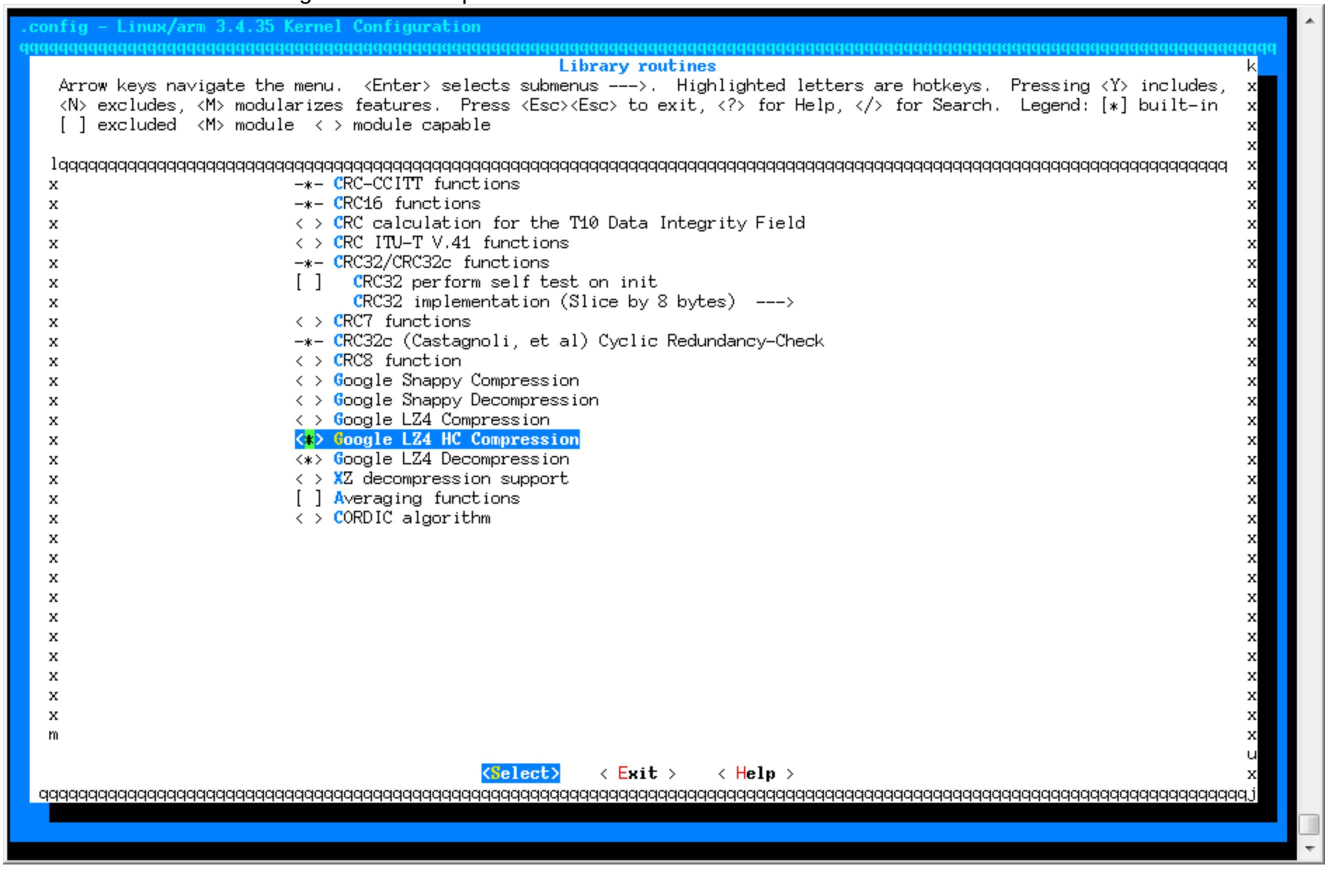
```
.config - Linux/arm 3.4.35 Kernel Configuration
qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq
                                                       General setup                                                      k
   Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted letters are hotkeys.  Pressing <Y> includes,  x
   <N> excludes, <M> modularizes features.  Press <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend: [*] built-in   x
   [ ] excluded  <M> module  < > module capable                                                                             x
                                                                                                                            x
  lqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq  x
  x                        [*] Prompt for development and/or incomplete code/drivers                                        x
  x                        ()  Cross-compiler tool prefix                                                                   x
  x                        ()  Local version - append to kernel release                                                     x
  x                        [ ] Automatically append version information to the version string                               x
  x                            Kernel compression mode (Gzip)  --->                                                         x
  x                        ((none)) Default hostname                                                                        x
  x                        [*] Support for paging of anonymous memory (swap)                                                x
  x                        [*] System V IPC                                                                                 x
  x                        [ ] POSIX Message Queues                                                                         x
  x                        [ ] BSD Process Accounting                                                                       x
  x                        [ ] open by fhandle syscalls                                                                     x
  m                        v(+)                                                                                             x
                                                                                                                            u
                                                 <Select>    < Exit >    < Help >                                           x
 qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj
```

Power management options → <*>Hibernation (aka ‘suspend to disk’)
→ <*>Snapshot Boot (aka ‘suspend to disk’)



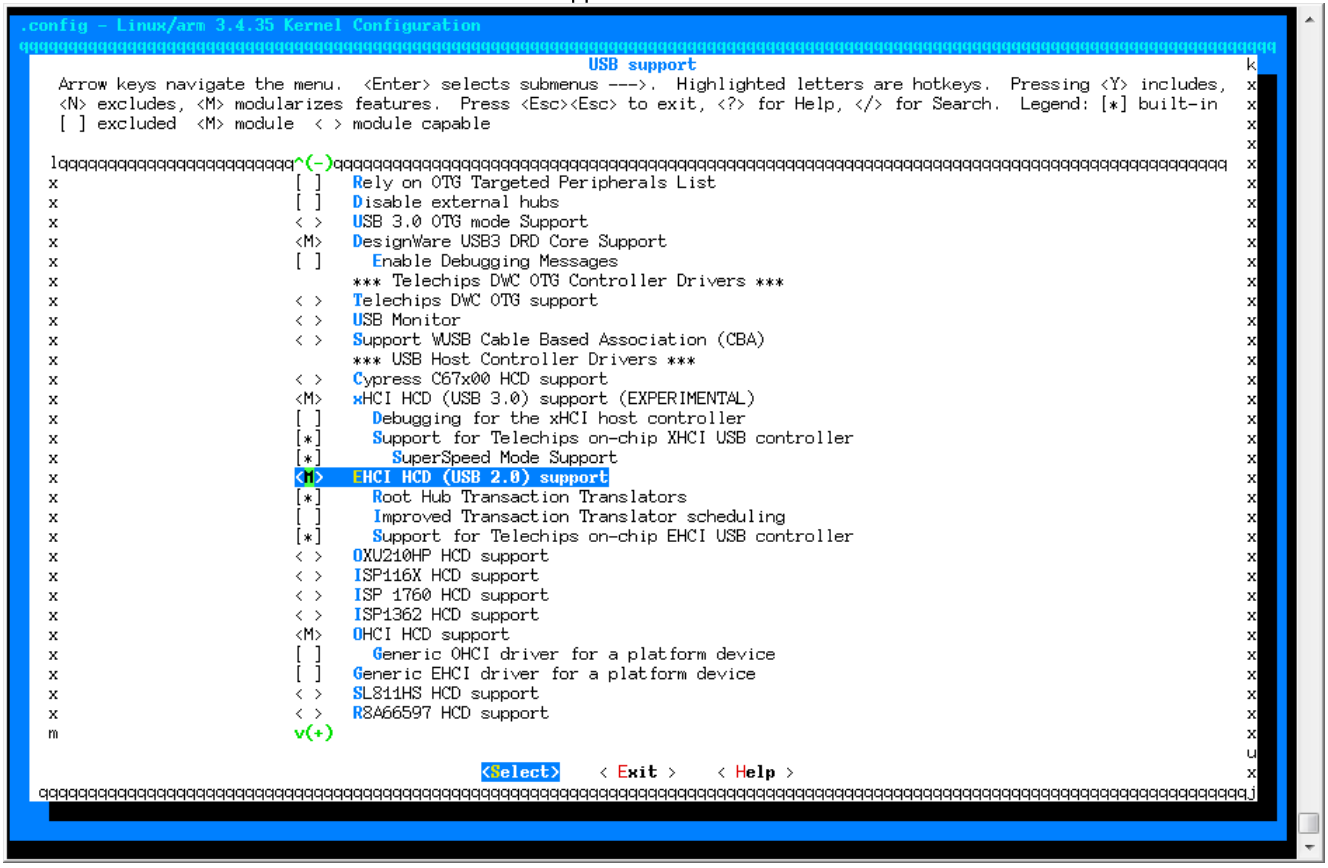
```
.config - Linux/arm 3.4.35 Kernel Configuration
qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq
                                                 Power management options                                                  k
   Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted letters are hotkeys.  Pressing <Y> includes,  x
   <N> excludes, <M> modularizes features.  Press <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend: [*] built-in   x
   [ ] excluded  <M> module  < > module capable                                                                             x
                                                                                                                            x
  lqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq  x
  x                        [*] Suspend to RAM and standby                                                                   x
  x                              Suspend mode (Shut-down mode)  --->                                                        x
  x                        [*] Hibernation (aka 'suspend to disk')                                                          x
  x                        [*] Snapshot Boot (aka 'suspend to disk')                                                        x
  x                        [ ]   No compressed snapshot image                                                               x
  x                        ()  Default resume partition                                                                     x
  x                        [*] Opportunistic sleep                                                                          x
  x                        [*] User space wakeup sources interface                                                          x
  x                        (100) Maximum number of user space wakeup sources (0 = no limit)                                 x
  x                        [*]   Garbage collector for user space wakeup sources                                            x
  x                        [*] Run-time PM core functionality                                                               x
  m                        v(+)                                                                                             x
                                                                                                                            u
                                                 <Select>    < Exit >    < Help >                                           x
 qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj
```

Library routines → <*> Google LZ4 HC Compression
→ <*> Google LZ4 Decompression



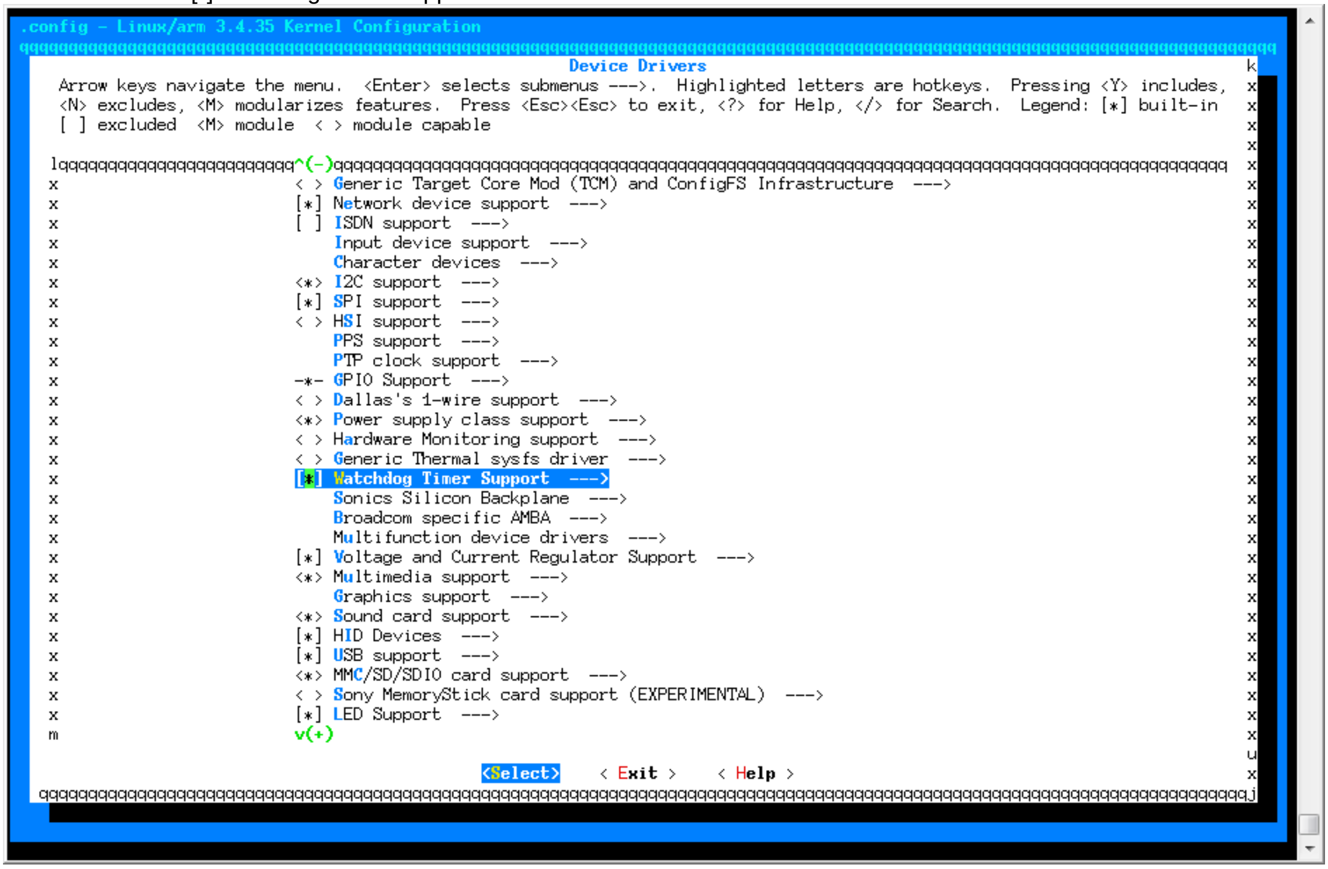
```
.config - Linux/arm 3.4.35 Kernel Configuration
                                  Library routines
  Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted letters are hotkeys.  Pressing <Y> includes,
  <N> excludes, <M> modularizes features.  Press <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend: [*] built-in
  [ ] excluded  <M> module  < > module capable

                        -*- CRC-CCITT functions
                        -*- CRC16 functions
                        < > CRC calculation for the T10 Data Integrity Field
                        < > CRC ITU-T V.41 functions
                        -*- CRC32/CRC32c functions
                        [ ]   CRC32 perform self test on init
                              CRC32 implementation (Slice by 8 bytes)  --->
                        < > CRC7 functions
                        -*- CRC32c (Castagnoli, et al) Cyclic Redundancy-Check
                        < > CRC8 function
                        < > Google Snappy Compression
                        < > Google Snappy Decompression
                        < > Google LZ4 Compression
                        <*> Google LZ4 HC Compression
                        <*> Google LZ4 Decompression
                        < > XZ decompression support
                        [ ] Averaging functions
                        < > CORDIC algorithm

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

Device Drivers → USB support → <M> EHCI HCD (USB 2.0) support
→ <M> OHCI HCD support



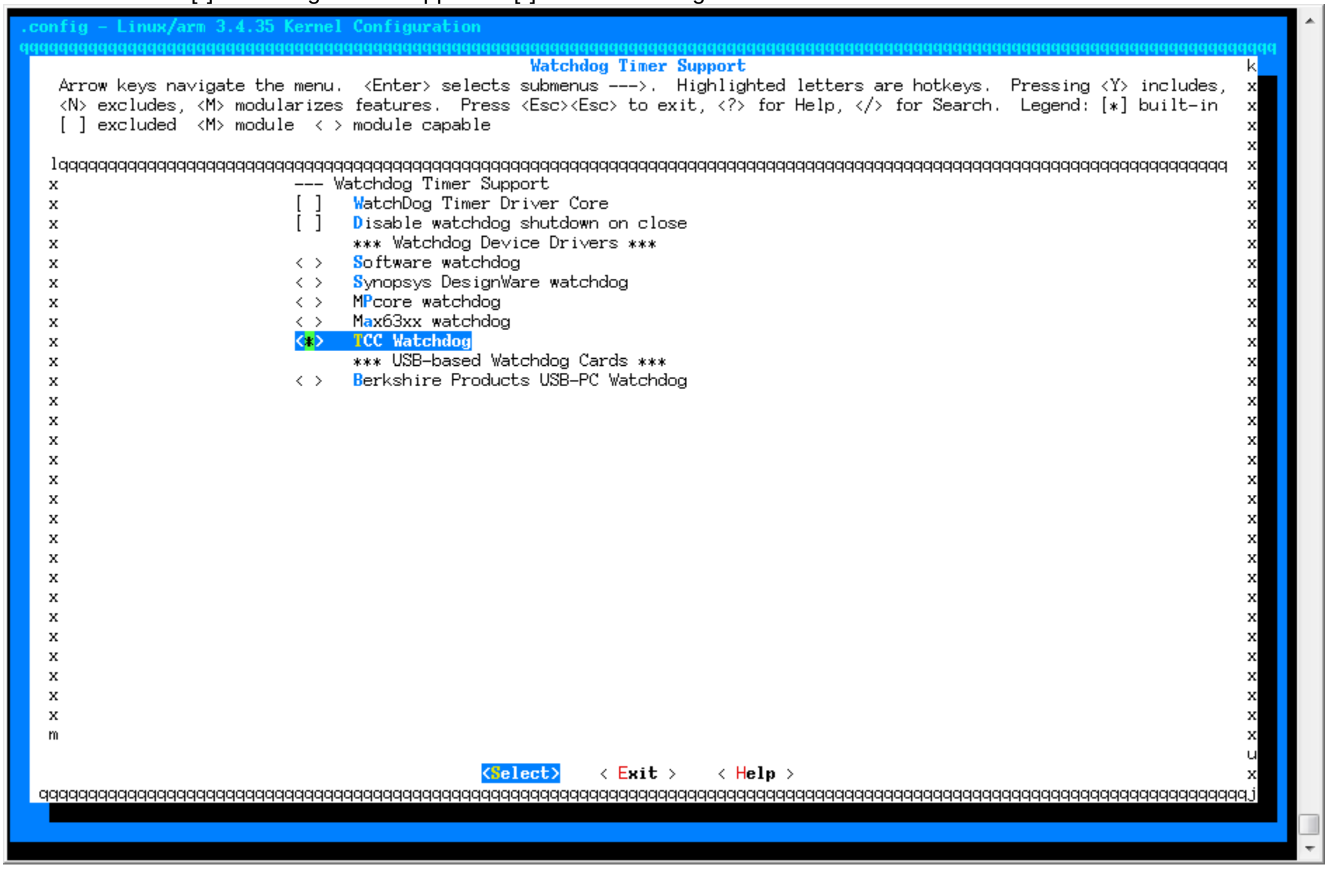
```
.config - Linux/arm 3.4.35 Kernel Configuration
                                  USB support
  Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted letters are hotkeys.  Pressing <Y> includes,
  <N> excludes, <M> modularizes features.  Press <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend: [*] built-in
  [ ] excluded  <M> module  < > module capable

                        ^(-)
                        [ ]   Rely on OTG Targeted Peripherals List
                        [ ]   Disable external hubs
                        < >   USB 3.0 OTG mode Support
                        <M>   DesignWare USB3 DRD Core Support
                        [ ]     Enable Debugging Messages
                              *** Telechips DWC OTG Controller Drivers ***
                        < >   Telechips DWC OTG support
                        < >   USB Monitor
                        < >   Support WUSB Cable Based Association (CBA)
                              *** USB Host Controller Drivers ***
                        < >   Cypress C67x00 HCD support
                        <M>   xHCI HCD (USB 3.0) support (EXPERIMENTAL)
                        [ ]     Debugging for the xHCI host controller
                        [*]     Support for Telechips on-chip XHCI USB controller
                        [*]       SuperSpeed Mode Support
                        <M>   EHCI HCD (USB 2.0) support
                        [*]     Root Hub Transaction Translators
                        [ ]     Improved Transaction Translator scheduling
                        [*]     Support for Telechips on-chip EHCI USB controller
                        < >   OXU210HP HCD support
                        < >   ISP116X HCD support
                        < >   ISP 1760 HCD support
                        < >   ISP1362 HCD support
                        <M>   OHCI HCD support
                        [ ]     Generic OHCI driver for a platform device
                        [ ]   Generic EHCI driver for a platform device
                        < >   SL811HS HCD support
                        < >   R8A66597 HCD support
                        v(+)

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

Device Drivers -> [*]Watchdog Timer Support



```
.config - Linux/arm 3.4.35 Kernel Configuration
qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq
                                              Device Drivers                                                         k
   Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted letters are hotkeys.  Pressing <Y> includes,  x
   <N> excludes, <M> modularizes features.  Press <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend: [*] built-in   x
   [ ] excluded  <M> module  < > module capable                                                                          x
                                                                                                                         x
  lqqqqqqqqqqqqqqqqqqqqqqqqq^(-)qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq  x
  x                         < > Generic Target Core Mod (TCM) and ConfigFS Infrastructure  --->                       x
  x                         [*] Network device support  --->                                                         x
  x                         [ ] ISDN support  --->                                                                   x
  x                             Input device support  --->                                                           x
  x                             Character devices  --->                                                              x
  x                         <*> I2C support  --->                                                                    x
  x                         [*] SPI support  --->                                                                    x
  x                         < > HSI support  --->                                                                    x
  x                             PPS support  --->                                                                    x
  x                             PTP clock support  --->                                                              x
  x                         -*- GPIO Support  --->                                                                   x
  x                         < > Dallas's 1-wire support  --->                                                        x
  x                         <*> Power supply class support  --->                                                     x
  x                         < > Hardware Monitoring support  --->                                                    x
  x                         < > Generic Thermal sysfs driver  --->                                                   x
  x                         [*] Watchdog Timer Support  --->                                                         x
  x                             Sonics Silicon Backplane  --->                                                       x
  x                             Broadcom specific AMBA  --->                                                         x
  x                             Multifunction device drivers  --->                                                   x
  x                         [*] Voltage and Current Regulator Support  --->                                          x
  x                         <*> Multimedia support  --->                                                             x
  x                             Graphics support  --->                                                               x
  x                         <*> Sound card support  --->                                                             x
  x                         [*] HID Devices  --->                                                                    x
  x                         [*] USB support  --->                                                                    x
  x                         <*> MMC/SD/SDIO card support  --->                                                       x
  x                         < > Sony MemoryStick card support (EXPERIMENTAL)  --->                                   x
  x                         [*] LED Support  --->                                                                    x
  m                         v(+)                                                                                     x
                                                                                                                     u
                                         <Select>    < Exit >    < Help >                                            x
 qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj
```

Device Drivers -> [*]Watchdog Timer Support -> [*] TCC Watchdog



```
.config - Linux/arm 3.4.35 Kernel Configuration
qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq
                                          Watchdog Timer Support                                                     k
   Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted letters are hotkeys.  Pressing <Y> includes,  x
   <N> excludes, <M> modularizes features.  Press <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend: [*] built-in   x
   [ ] excluded  <M> module  < > module capable                                                                          x
                                                                                                                         x
  lqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk  x
  x                         --- Watchdog Timer Support                                                               x
  x                         [ ]   WatchDog Timer Driver Core                                                         x
  x                         [ ]   Disable watchdog shutdown on close                                                 x
  x                               *** Watchdog Device Drivers ***                                                    x
  x                         < >   Software watchdog                                                                  x
  x                         < >   Synopsys DesignWare watchdog                                                       x
  x                         < >   MPcore watchdog                                                                    x
  x                         < >   Max63xx watchdog                                                                   x
  x                         <*>   TCC Watchdog                                                                       x
  x                               *** USB-based Watchdog Cards ***                                                   x
  x                         < >   Berkshire Products USB-PC Watchdog                                                 x
  x                                                                                                                  x
  x                                                                                                                  x
  x                                                                                                                  x
  x                                                                                                                  x
  x                                                                                                                  x
  x                                                                                                                  x
  x                                                                                                                  x
  x                                                                                                                  x
  x                                                                                                                  x
  x                                                                                                                  x
  x                                                                                                                  x
  x                                                                                                                  x
  x                                                                                                                  x
  x                                                                                                                  x
  x                                                                                                                  x
  x                                                                                                                  x
  x                                                                                                                  x
  m                                                                                                                  x
                                                                                                                     u
                                         <Select>    < Exit >    < Help >                                            x
 qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj
```

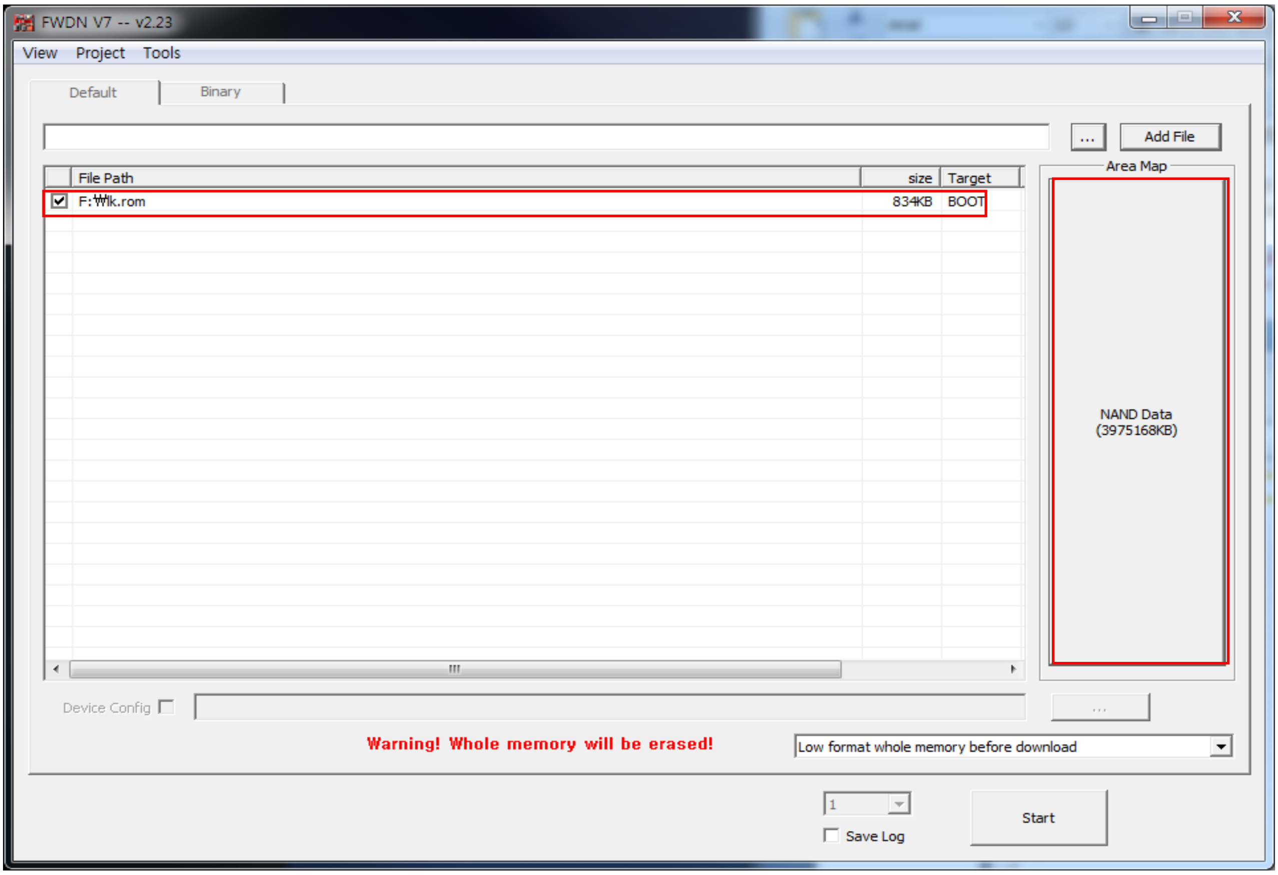
2.3 Creating snapshot partition

Telechips quick boot solution support two storage type(NAND/eMMC(SD)).

To include quick boot feature, you have to create snapshot partition using FWDN.

The procedure to create snapshot partition is as follow

1) Run FWDN



	File Path	size	Target
☑	F:₩lk.rom	834KB	BOOT

NAND Data
(3975168KB)

Device Config □

Warning! Whole memory will be erased!

Low format whole memory before download

1

□ Save Log

2) Choose EMMC Data and following menu is displayed.

Prepare ‘snapshot partition’ as following.

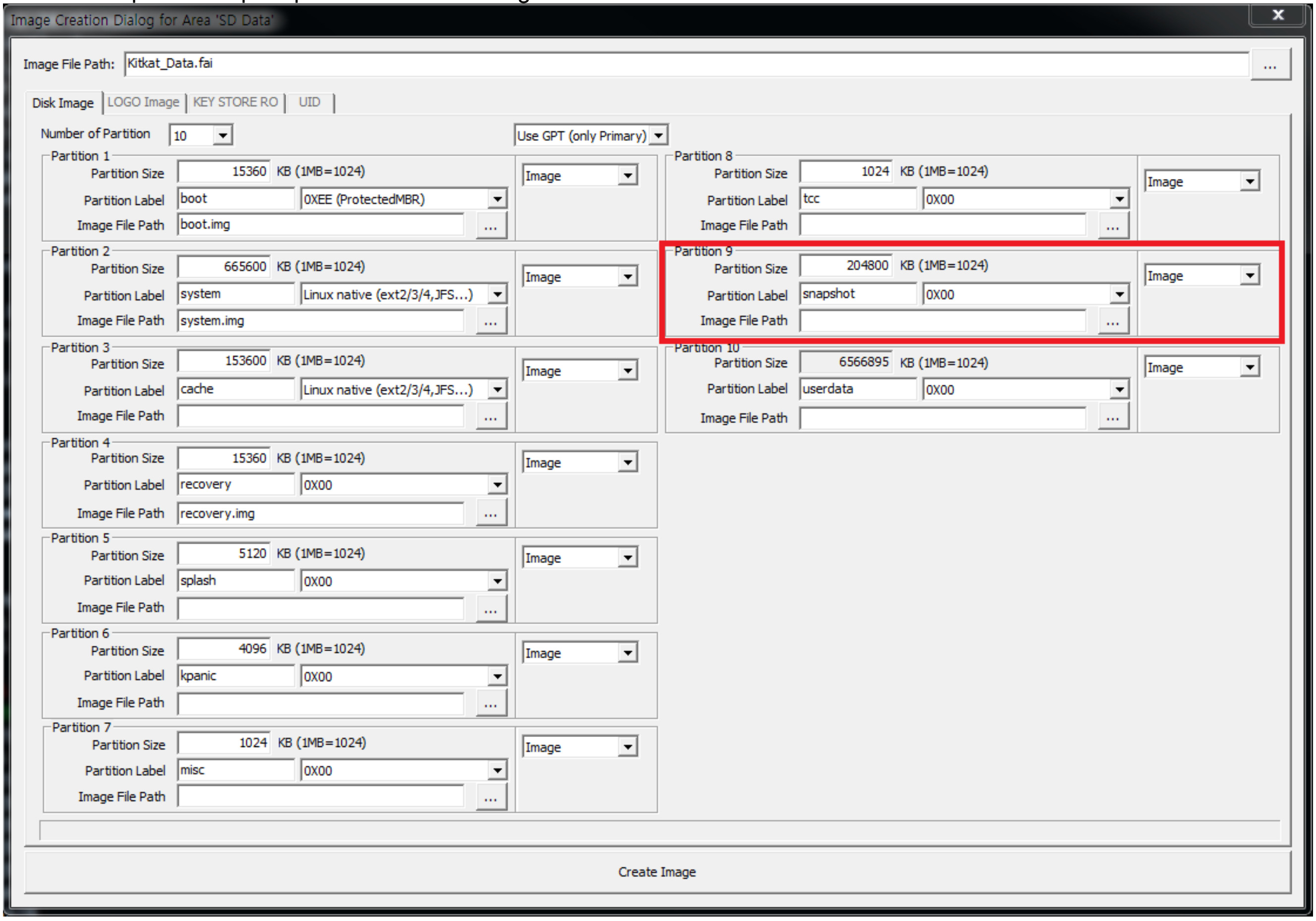


Image Creation Dialog for Area 'SD Data'

Image File Path: Kitkat_Data.fai

Number of Partition: 10

Use GPT (only Primary)

Partition 1
Partition Size: 15360 KB (1MB=1024)
Partition Label: boot — 0XEE (ProtectedMBR)
Image File Path: boot.img
Image

Partition 2
Partition Size: 665600 KB (1MB=1024)
Partition Label: system — Linux native (ext2/3/4,JFS...)
Image File Path: system.img
Image

Partition 3
Partition Size: 153600 KB (1MB=1024)
Partition Label: cache — Linux native (ext2/3/4,JFS...)
Image File Path:
Image

Partition 4
Partition Size: 15360 KB (1MB=1024)
Partition Label: recovery — 0X00
Image File Path: recovery.img
Image

Partition 5
Partition Size: 5120 KB (1MB=1024)
Partition Label: splash — 0X00
Image File Path:
Image

Partition 6
Partition Size: 4096 KB (1MB=1024)
Partition Label: kpanic — 0X00
Image File Path:
Image

Partition 7
Partition Size: 1024 KB (1MB=1024)
Partition Label: misc — 0X00
Image File Path:
Image

Partition 8
Partition Size: 1024 KB (1MB=1024)
Partition Label: tcc — 0X00
Image File Path:
Image

Partition 9
Partition Size: 204800 KB (1MB=1024)
Partition Label: snapshot — 0X00
Image File Path:
Image

Partition 10
Partition Size: 6566895 KB (1MB=1024)
Partition Label: userdata — 0X00
Image File Path:
Image

SNAPSHOT PARTITION SIZE : About 200MB (204800 KB)

We need to download a new image to the device frequently. However, we have to be carefully if we enable quick-boot feature. To erase snapshot boot image, you have to choose image filled with 0x00.

How to make image that fill with 0.
Execute “dd if=/dev/zero of=empty_snapshot.img bs=4096 count=10” in linux shell

- We recommend to use 200MB for snapshot partition when developing the solution.
  If The customer use prebuilt mode, they can optimize it’s size through enabling below log just before mass-production.

  Path: kernel/kernel/power/swap.c, enough_swap() function.
  printk("PM: Free swap pages: %u required swap pages %u %u \n", free_swap, required, nr_pages);

  Needed snapshot partition size will be about (required * 4096)/1024 KB.

3) And then click “create image” button

Note

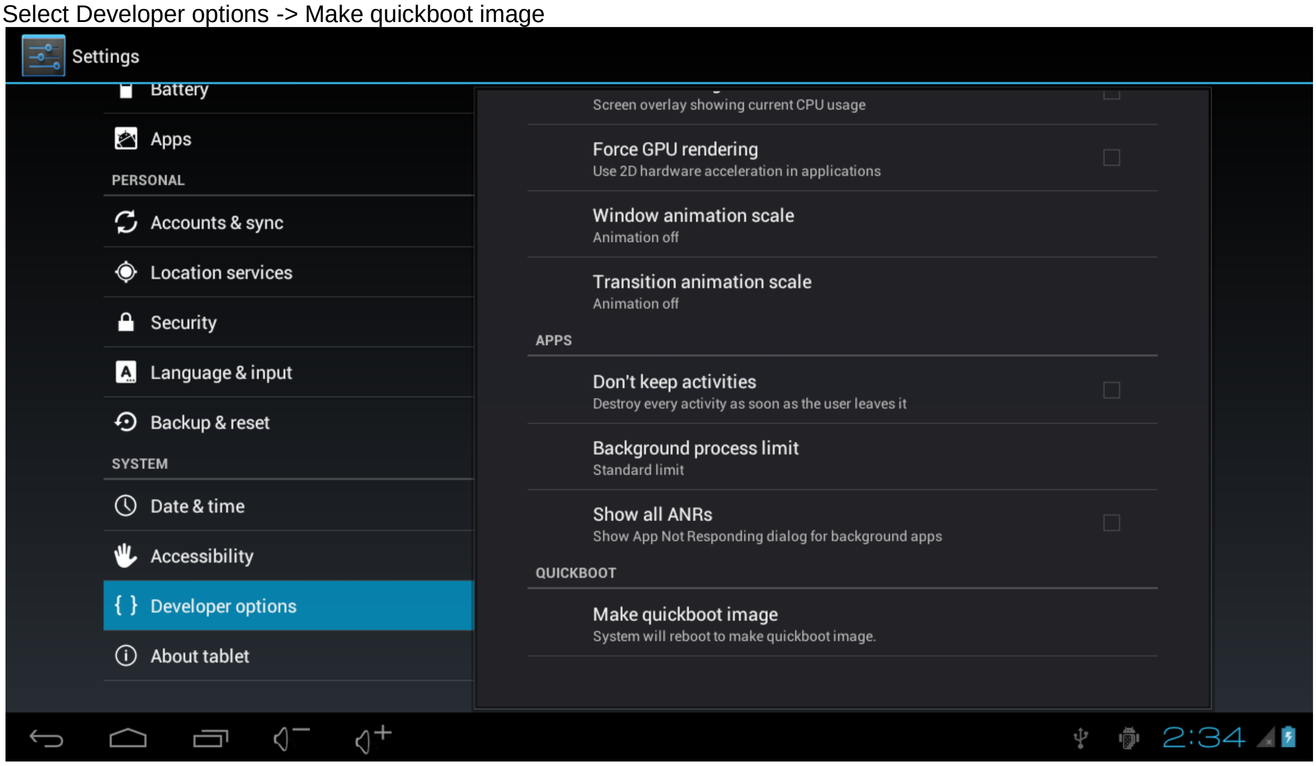
Usage of FWDN refers to
“TCCxxxx-Android-ALL-V1.0-Partition Layout Guide.pdf”
“TCC892X-Android_4.4(Kitkat)-V0.01E-Quick Start Guide.pdf”
“TCC893X-Android_4.4(Kitkat)-V0.01E-Quick Start Guide.pdf”
“TCC892X-Android_4.4(Kitkat)-V1.01E-SDMMC Boot Guide.pdf”
“TCC893X-Android_4.4(Kitkat)-V1.01E-SDMMC Boot Guide.pdf”

3 How to make snapshot boot Image

The snapshot boot image is generated once at the initial stage.

The followings are how to generate the snapshot boot images.

1. Execute Settings Application.
2. Select Developer options -> Make quickboot image



Settings

Battery
Apps
PERSONAL
Accounts & sync
Location services
Security
Language & input
Backup & reset
SYSTEM
Date & time
Accessibility
{ } Developer options
About tablet

Screen overlay showing current CPU usage ☐

Force GPU rendering
Use 2D hardware acceleration in applications ☐

Window animation scale
Animation off

Transition animation scale
Animation off

APPS

Don't keep activities
Destroy every activity as soon as the user leaves it ☐

Background process limit
Standard limit

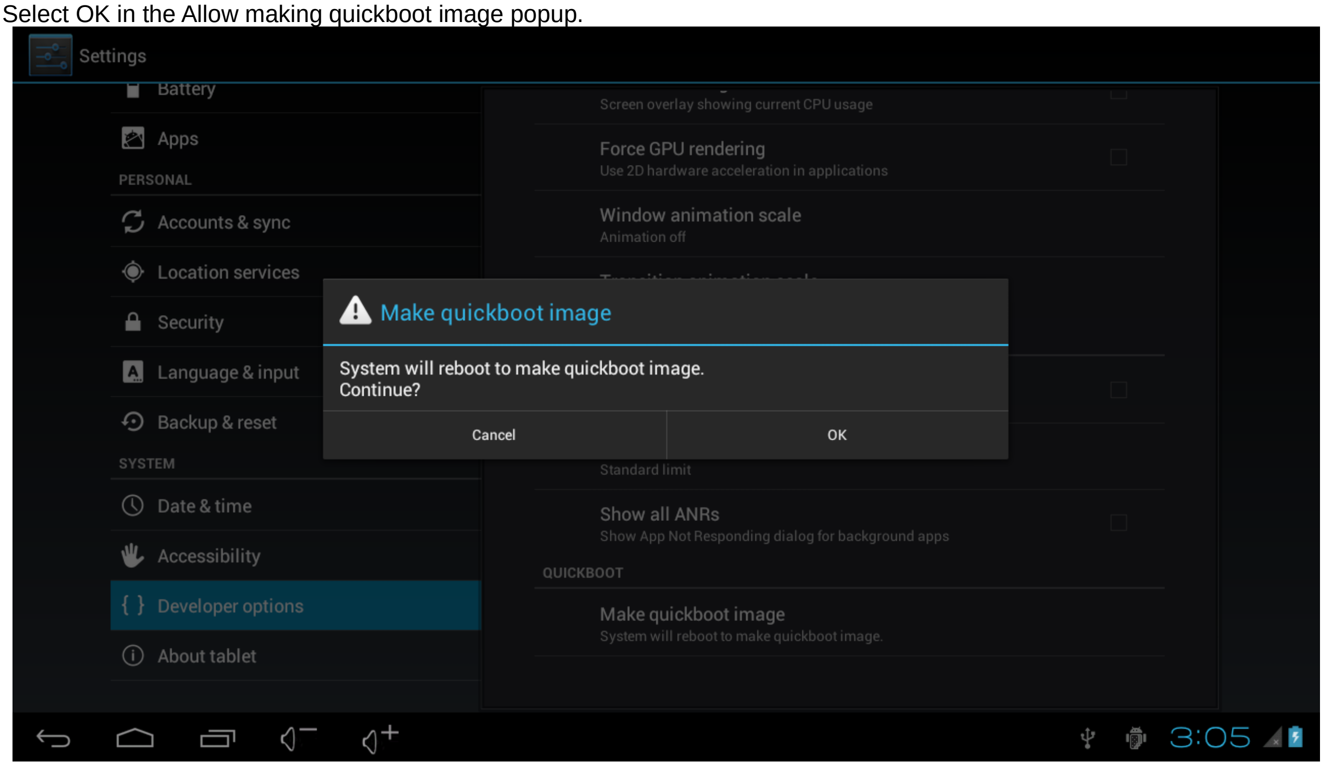
Show all ANRs
Show App Not Responding dialog for background apps ☐

QUICKBOOT

Make quickboot image
System will reboot to make quickboot image.

2:34

3. Select OK in the Allow making quickboot image popup.



Settings

Battery
Apps
PERSONAL
Accounts & sync
Location services
Security
Language & input
Backup & reset
SYSTEM
Date & time
Accessibility
{ } Developer options
About tablet

Screen overlay showing current CPU usage ☐

Force GPU rendering
Use 2D hardware acceleration in applications ☐

Window animation scale
Animation off

⚠ Make quickboot image

System will reboot to make quickboot image.
Continue?

Cancel | OK

Standard limit

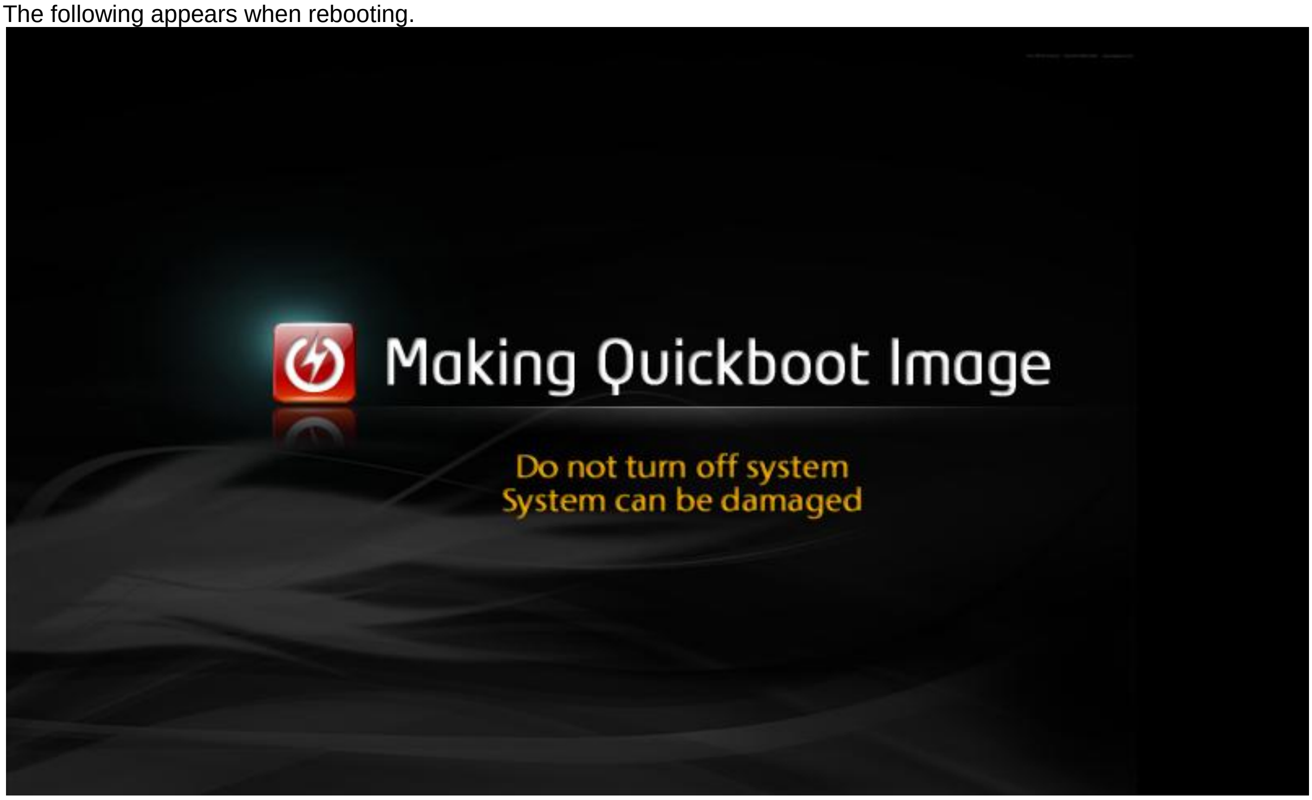
Show all ANRs
Show App Not Responding dialog for background apps ☐

QUICKBOOT

Make quickboot image
System will reboot to make quickboot image.

3:05

4. The following appears when rebooting.



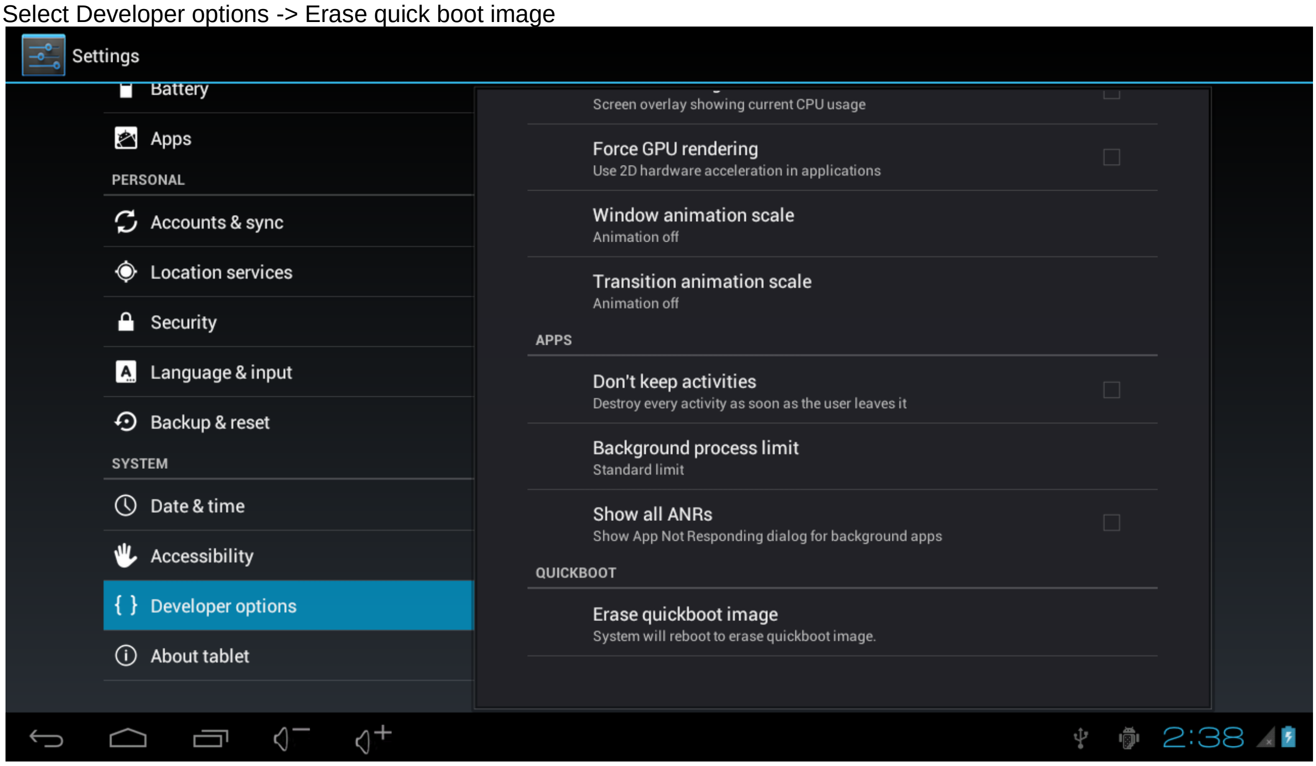
Making Quickboot Image

Do not turn off system
System can be damaged

4 How to erase snapshot boot image

4.1 Delete the snapshot boot image from the setting ui

1. Execute Settings Application.
2. Select Developer options -> Erase quick boot image



Settings

Battery
Apps
PERSONAL
Accounts & sync
Location services
Security
Language & input
Backup & reset
SYSTEM
Date & time
Accessibility
Developer options
About tablet

Screen overlay showing current CPU usage □

Force GPU rendering
Use 2D hardware acceleration in applications □

Window animation scale
Animation off

Transition animation scale
Animation off

APPS

Don't keep activities
Destroy every activity as soon as the user leaves it □

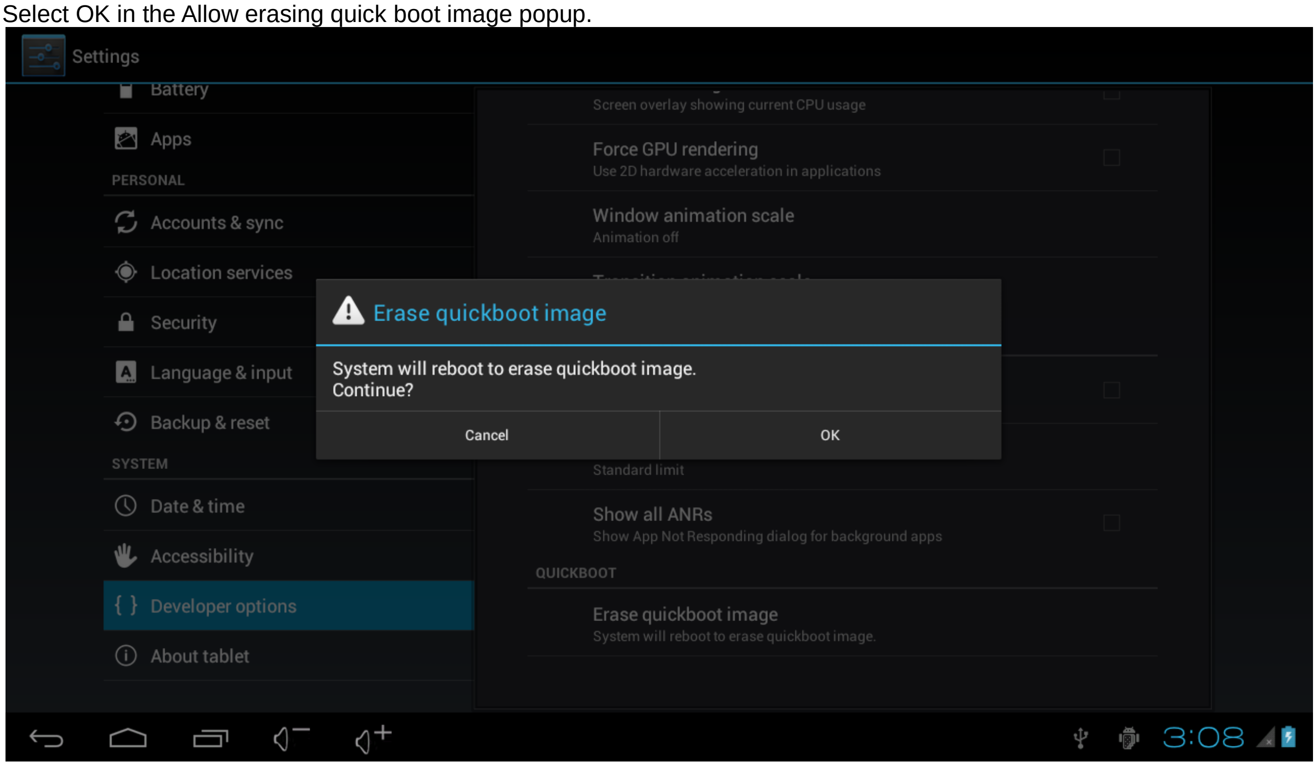
Background process limit
Standard limit

Show all ANRs
Show App Not Responding dialog for background apps □

QUICKBOOT

Erase quickboot image
System will reboot to erase quickboot image.

3. Select OK in the Allow erasing quick boot image popup.



Erase quickboot image

System will reboot to erase quickboot image.
Continue?

Cancel OK

4.2 Delete the snapshot boot image with the fastboot

1. In the fastboot mode, connect USB.
2. Execute “$fastboot erase snapshot” in the shell.

5 How to disable quickboot feature

5.1 Patch kk-disable-quickboot.patch in order to disable quick boot

The following shows how to enable the quick boot.

1) cd your kitkat project
2) Execute “patch –p1 < device/telechips/common/kk-disable-quickboot.patch” in shell

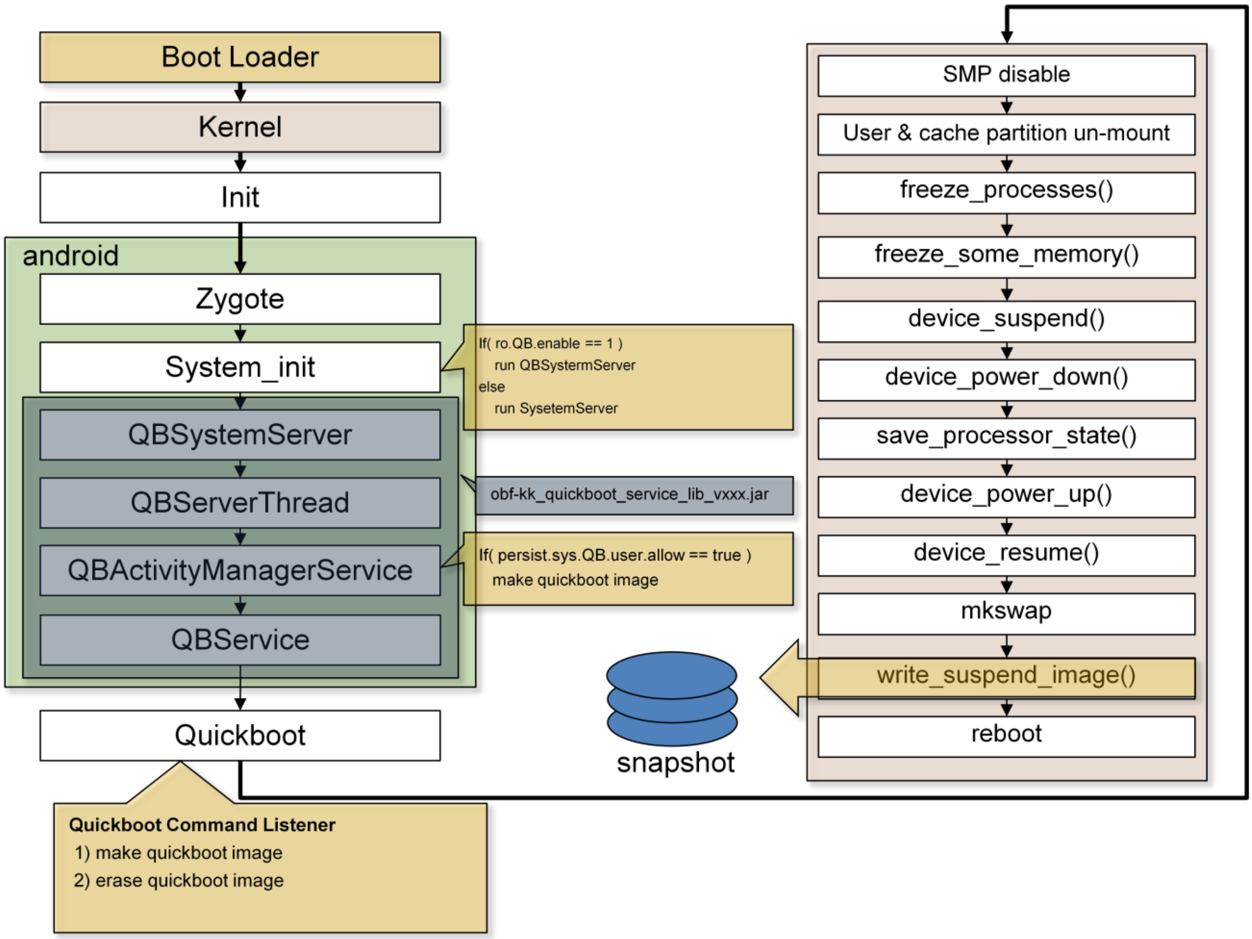
5.2 Disable quickboot feature in kernel configuration

```
below kernel configuration unset.

General setup -> support for paging of anonymous memory (swap)
Power management options -> Hibernation (aka ‘suspend to disk’)
                            -> Snapshot Boot (aka ‘suspend to disk’)
Library routines -> Google LZ4 HC Compression
                 -> Google LZ4 Decompression
 Device Drivers → USB support → <*> EHCI HCD (USB 2.0) support
                              → <*> OHCI HCD support
```

6 Appendix

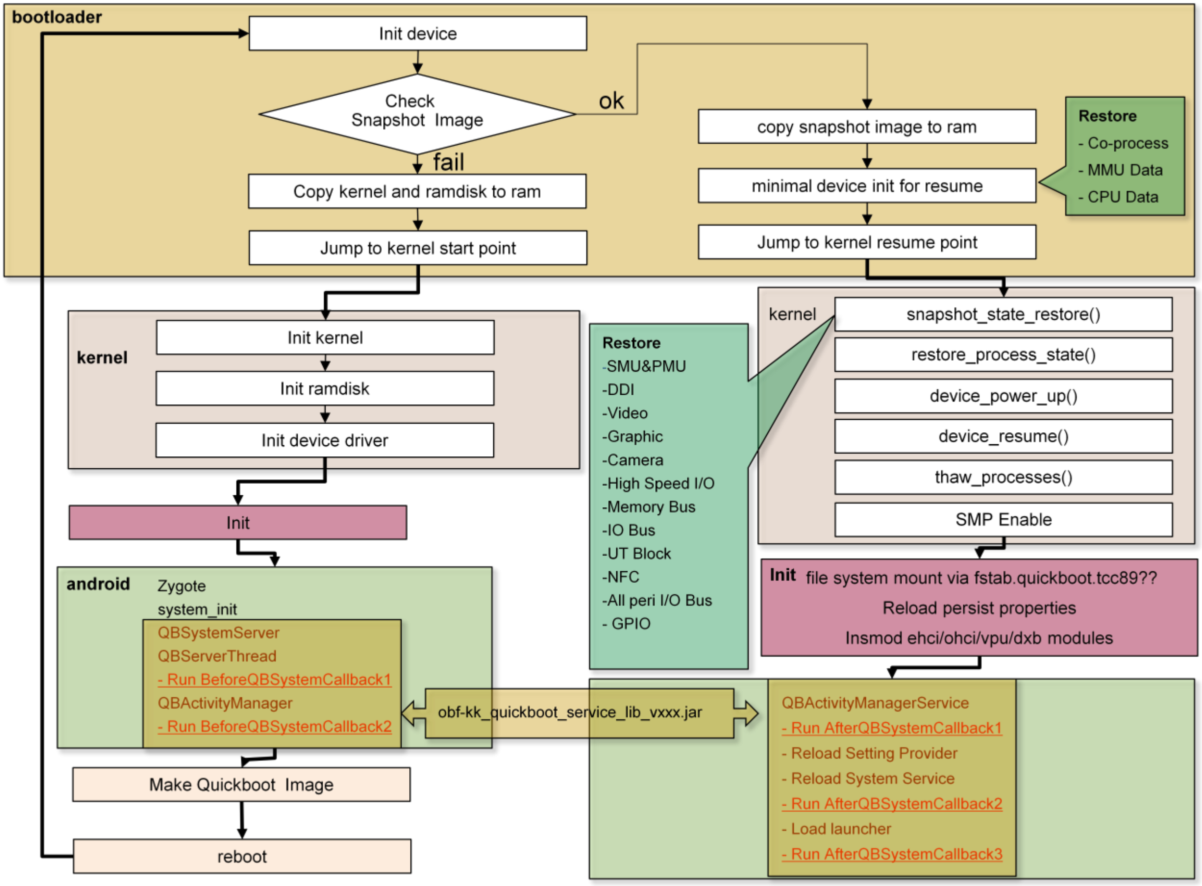
6.1 When snapshot boot image is created

The following shows procedure for making snapshot image



6.2 Modify device driver for quick boot

The following show procedure for booting with snapshot image.



During snapshot boot, probe function of platform_driver is not executed and resume function will be called same as wake up from deep sleep. The suspend / resume functions are different wheather it use dev_pm_ops or not.

6.2.1 In case of not using CONFIG_PM_RUNTIME & dev_pm_ops

The proto type of platform_driver structure is as follows. It is the same with JellyBean.

```
struct platform_driver {
        int (*probe)(struct platform_device *);
        int (*remove)(struct platform_device *);
        void (*shutdown)(struct platform_device *);
        int (*suspend)(struct platform_device *, pm_message_t state);
        int (*resume)(struct platform_device *);
        struct device_driver driver;
        const struct platform_device_id *id_table;
};
```

6.2.2 In case of using CONFIG_PM_RUNTIME & dev_pm_ops

If the platform_driver->driver use CONFIG_PM_RUNTIME & dev_pm_ops, QuickBoot does not call platform_driver->suspend/resume functions. Instead of platform_driver->suspend/resume, QuickBoot call dev_pm_ops->freeze/thaw/restore functions.

```
struct platform_driver {
        int (*probe)(struct platform_device *);
        int (*remove)(struct platform_device *);
        void (*shutdown)(struct platform_device *);
        int (*suspend)(struct platform_device *, pm_message_t state);
        int (*resume)(struct platform_device *);
        struct device_driver driver;
        const struct platform_device_id *id_table;
};
```

```
struct device_driver {
        const char          *name;
        struct bus_type        *bus;

        struct module           *owner;
        const char          *mod_name;    /* used for built-in modules */

        bool suppress_bind_attrs;     /* disables bind/unbind via sysfs */

        const struct of_device_id     *of_match_table;

        int (*probe) (struct device *dev);
        int (*remove) (struct device *dev);
        void (*shutdown) (struct device *dev);
        int (*suspend) (struct device *dev, pm_message_t state);
        int (*resume) (struct device *dev);
        const struct attribute_group **groups;

        const struct dev_pm_ops *pm;

        struct driver_private *p;
};
```

If the Device Driver defines **platform_driver -> driver -> dev_pm_ops (.pm) ,** QuickBoot Call **dev_pm_ops ->freeze/thaw/restore.** So the Device Driver is Needed to define freeze / thaw / restore functions in dev_pm_ops.

```
struct dev_pm_ops {
        int (*prepare)(struct device *dev);
        void (*complete)(struct device *dev);
        int (*suspend)(struct device *dev);
        int (*resume)(struct device *dev);
        int (*freeze)(struct device *dev);
        int (*thaw)(struct device *dev);
        int (*poweroff)(struct device *dev);
        int (*restore)(struct device *dev);
        int (*suspend_late)(struct device *dev);
        int (*resume_early)(struct device *dev);
        int (*freeze_late)(struct device *dev);
        int (*thaw_early)(struct device *dev);
        int (*poweroff_late)(struct device *dev);
        int (*restore_early)(struct device *dev);
        int (*suspend_noirq)(struct device *dev);
        int (*resume_noirq)(struct device *dev);
        int (*freeze_noirq)(struct device *dev);
        int (*thaw_noirq)(struct device *dev);
        int (*poweroff_noirq)(struct device *dev);
        int (*restore_noirq)(struct device *dev);
        int (*runtime_suspend)(struct device *dev);
        int (*runtime_resume)(struct device *dev);
        int (*runtime_idle)(struct device *dev);
};
```

So, You need to define freeze / thaw / restore functions in dev_pm_ops.

Additionally, dev_pm_ops -> suspend / resume will be called when the system state goes to Sleep Mode.

6.2.3 How to know the system is boot in QuickBoot or normal boot sequence.

When snapshot booting is done, do_hibernate_boot variable will become 1.
So, you can use this variable to decide whether resume function should be executed or not.
Vaiable "do_hibernate_boot" is defined as follows.

```
        unsigned int do_hibernate_boot;
```

Normally, Telechips Android platform supports deep sleep mode and it means default device drivers includes resume function and don't need to implement it.
However, you have to make resume function if you want to use new device.

6.3 Why start up time becomes slower

6.3.1 Rescanning package

The launcher is executed after executing package rescan. For this reason, booting time may be increased depending on the numbers of the installed applications in the user area. Rescanning package is executed in the System Server(partial init).

6.4 Need to compile android platform twice

In order to increase the stability of file system, we need to un-mount the user and cache partition before creating a snapshot image. For this reason, we need to use “WITH_DEXPREOPT=true” in build/envsetup.sh.

If you use “WITH_DEXPREOPT=true” option, you see a dex synchronization error. It is shown below.

```
dalvikvm( 1754): Zip is good, but no classes.dex inside, and no valid .odex file in the same directory
W/PackageManager( 1754): StaleDexCacheError when reading apk: /system/app/SettingsProvider.apk
W/PackageManager( 1754): dalvik.system.StaleDexCacheError: /system/app/SettingsProvider.apk
W/PackageManager( 1754):        at dalvik.system.DexFile.isDexOptNeeded(Native Method)
W/PackageManager(               1754):                                                        at
com.android.server.pm.PackageManagerService.performDexOptLI(PackageManagerService.java:3536)
W/PackageManager(               1754):                                                        at
com.android.server.pm.PackageManagerService.performDexOpt(PackageManagerService.java:3521)
W/PackageManager(               1754):                                                        at
com.android.server.am.ActivityManagerService.ensurePackageDexOpt(ActivityManagerService.java:1926)
W/PackageManager(               1754):                                                        at
com.android.server.am.ActivityManagerService.generateApplicationProvidersLocked(ActivityManagerService.java:6252)
W/PackageManager(               1754):                                                        at
com.android.server.am.QBActivityManagerService.installSystemProviders(QBActivityManagerService.java:64)
W/PackageManager( 1754):        at com.android.server.QBServerThread.run(QBServerThread.java:291)
E/System   ( 1754): ******************************************
E/System   ( 1754): ************ Failure starting core service
E/System   ( 1754): java.lang.RuntimeException: Unable to get provider com.android.providers.settings.SettingsProvider:
java.lang.ClassNotFoundException: Didn't find class "com.android.providers.settings.SettingsProvider" on path:
/system/app/SettingsProvider.apk
E/System   ( 1754):        at android.app.ActivityThread.installProvider(ActivityThread.java:4822)
E/System   ( 1754):        at android.app.ActivityThread.installContentProviders(ActivityThread.java:4432)
E/System   ( 1754):        at android.app.ActivityThread.installSystemProviders(ActivityThread.java:4970)
E/System   ( 1754):        at com.android.server.am.QBActivityManagerService.installSystemProviders(QBActivityManagerService.java:77)
E/System   ( 1754):        at com.android.server.QBServerThread.run(QBServerThread.java:291)
E/System   ( 1754): Caused by: java.lang.ClassNotFoundException: Didn't find class "com.android.providers.settings.SettingsProvider" on
path: /system/app/SettingsProvider.apk
E/System   ( 1754):        at dalvik.system.BaseDexClassLoader.findClass(BaseDexClassLoader.java:65)
E/System   ( 1754):        at java.lang.ClassLoader.loadClass(ClassLoader.java:501)
E/System   ( 1754):        at java.lang.ClassLoader.loadClass(ClassLoader.java:461)
E/System   ( 1754):        at android.app.ActivityThread.installProvider(ActivityThread.java:4807)
E/System   ( 1754):        ... 4 more
I/QBSystemServer( 1754): Input Method Service
```

If you meet this error, please compile android platform once more.

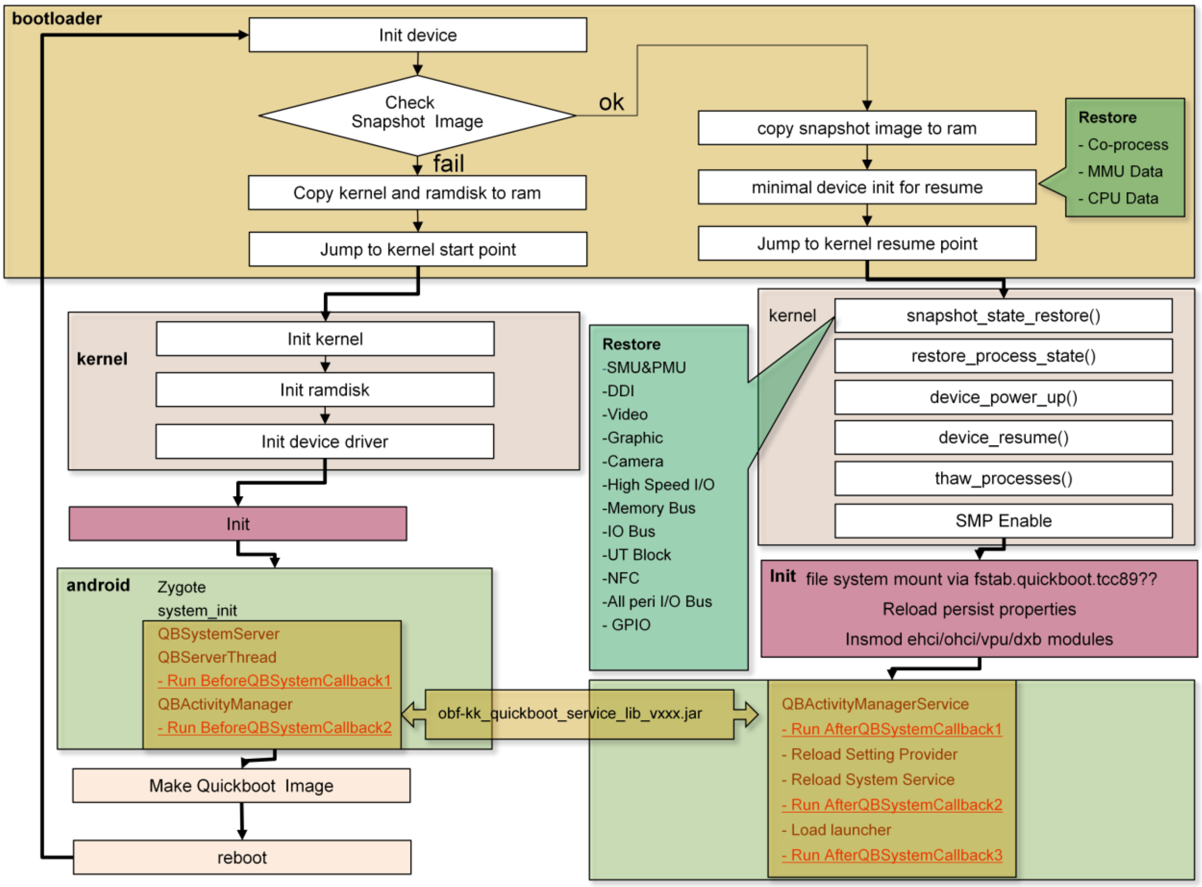
6.5 How to register new service

If ro.QB.enable(It declared in device.mk) property is true, the QBSystemServer will be run instead of the SystemServer. QBSystemServer is included to frameworks/base/service/java/obf-kk_quickboot_service_lib_v1.xxx,.jar.

If you want to register new service, please register a new service in the following Callback.

1. frameworks/base/service/java/com/android/server/BeforeQBSystemCallback1.java
2. frameworks/base/service/java/com/android/server/BeforeQBSystemCallback2.java
3. frameworks/base/service/java/com/android/server/AfterQBSystemCallback1.java
4. frameworks/base/service/java/com/android/server/ AfterQBSystemCallback2.java
5. frameworks/base/service/java/com/android/server/ AfterQBSystemCallback3.java

The sequences of calling a callback refer to red color text in the diagram below.



In order to run check disk for user/cache partition we have to un-mount these partition. So file handle have to release on user and cache partition. If you open file handle in your application or service before making quick-boot image, you have to release file handle.

6.6 Displaying an image during making snapshot image

During making snapshot image for quick boot, an image (Quickboot logo) can be display.

6.6.1 Preparing an image (*.rle)

※ Image format for Quick boot logo is 24bit bmp.

In order to make an image(*.rle) for Quickboot logo, move to the following path.

```
$ cd ~/mydroid/android/
$ cd device/telechips/common/
```

After making 1024x600 (depending on LCD size, default 1024x600) bmp image file, convert it to *.img(raw) format and then convert to *.rle file as follows.

```
$ convert -depth 8 ci_quickboot.bmp rgb:quickboot_1024x600.raw
$ rgb2565 -rle <quickboot_1024x600.raw> Quickboot_1024x600.rle
```

6.6.2 Setting path for Quickboot Logo

You can set path for Quickboot logo by modifying `device/telechips/tcc893x/device.mk` as follows.

```
$ cd ~/mydroid/android/
$ vi device/telechips/tcc893x/device.mk
```

`device/telechips/tcc893x/device.mk`.

```
…
PRODUCT_COPY_FILES += \
        device/telechips/common/initlogo1024x600.rle:root/initlogo.rle  \
        device/telechips/common/Quickboot_1024x600.rle:root/QuickBoot_logo.rle
…
```

6.7 Displaying an image during quickboot

After kernel starts, an image can be displayed and will be maintained until android starts. This image is called as agree image and it can be used for various purposes. For example, it can be used to ask something to end user.

Two types of image can be used for agree image as follows. You have to choose which one will be used before enabling this feature.

6.7.1 Using last framebuffer

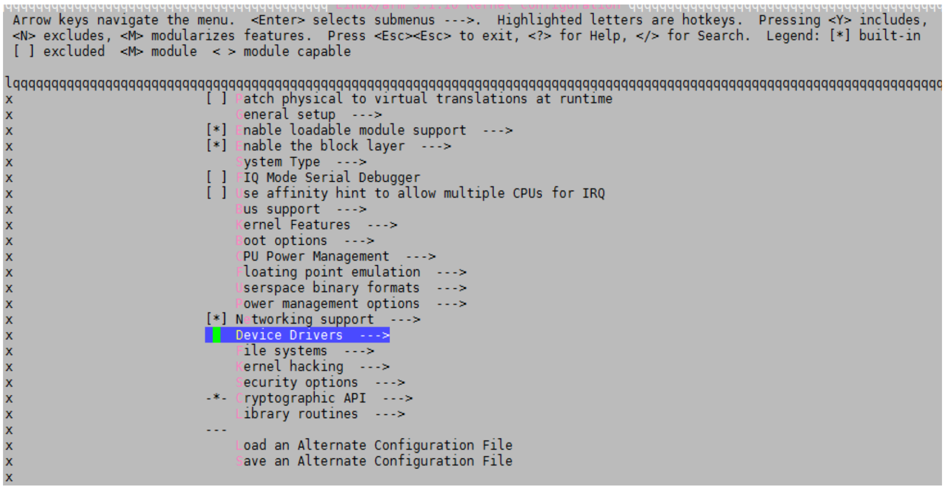
You can use an image in framebuffer just before starting making a snapshot boot image

※ Quickboot logo is output when creating a snapshot boot image(refer to 6.6.2)

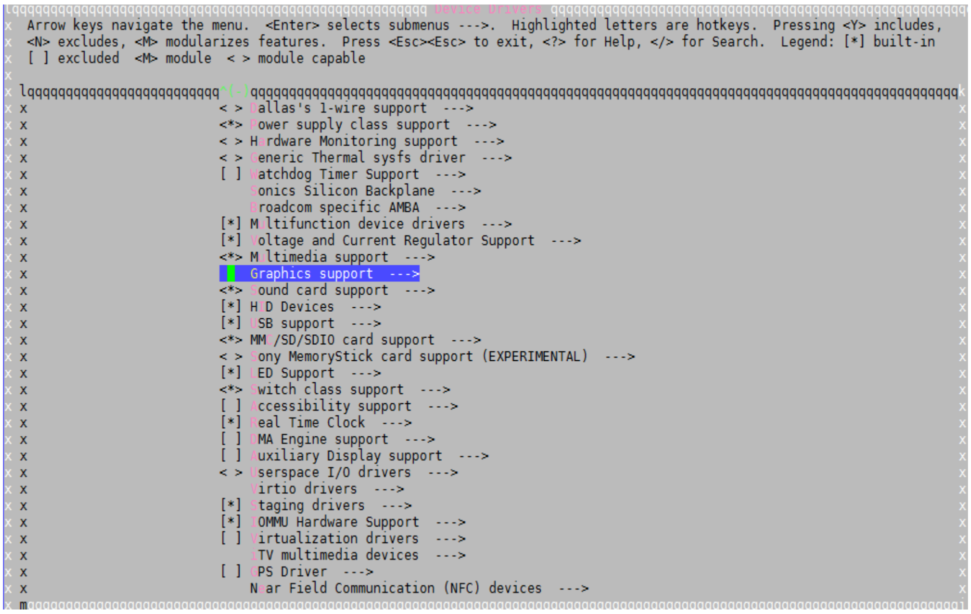
The fb driver is resumed,with rebooting(refer to 6.2). Then, It displays the image which was saved framebuffer image just before output Quickboot logo (refer to 6.6).

6.7.2 Enable Using last framebuffer feature in kernel

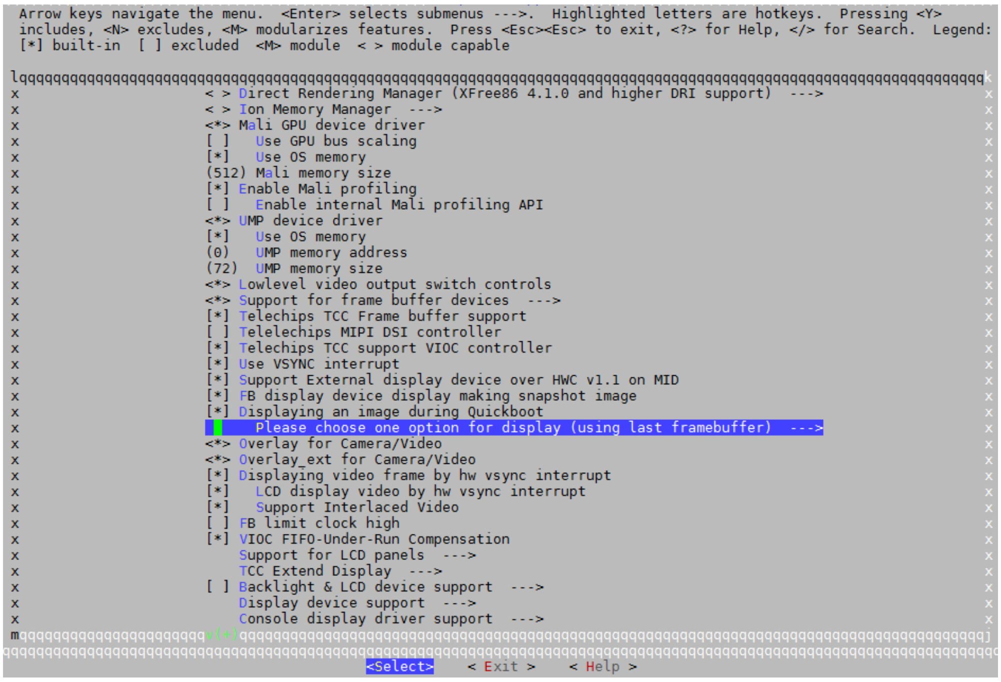
Device drivers → Graphics support → [*] Displaying an image during Quickboot
→ [*] using last framebuffer



```
 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

x                          [ ] Patch physical to virtual translations at runtime
x                              General setup  --->
x                          [*] Enable loadable module support  --->
x                          [*] Enable the block layer  --->
x                              System Type  --->
x                          [ ] FIQ Mode Serial Debugger
x                          [ ] Use affinity hint to allow multiple CPUs for IRQ
x                              Bus support  --->
x                              Kernel Features  --->
x                              Boot options  --->
x                              CPU Power Management  --->
x                              Floating point emulation  --->
x                              Userspace binary formats  --->
x                              Power management options  --->
x                          [*] Networking support  --->
x                              Device Drivers  --->
x                              File systems  --->
x                              Kernel hacking  --->
x                              Security options  --->
x                          -*- Cryptographic API  --->
x                              Library routines  --->
x                          ---
x                              Load an Alternate Configuration File
x                              Save an Alternate Configuration File
x
```



```
                                         Device Drivers
x  Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted letters are hotkeys.  Pressing <Y> includes,
x  <N> excludes, <M> modularizes features.  Press <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend: [*] built-in
x  [ ] excluded  <M> module  < > module capable
x
x x                          < > Dallas's 1-wire support  --->                                                        x
x x                          <*> Power supply class support  --->                                                     x
x x                          < > Hardware Monitoring support  --->                                                    x
x x                          < > Generic Thermal sysfs driver  --->                                                   x
x x                          [ ] Watchdog Timer Support  --->                                                         x
x x                              Sonics Silicon Backplane  --->                                                       x
x x                              Broadcom specific AMBA  --->                                                         x
x x                          [*] Multifunction device drivers  --->                                                   x
x x                          [*] Voltage and Current Regulator Support  --->                                          x
x x                          <*> Multimedia support  --->                                                             x
x x                              Graphics support  --->                                                               x
x x                          <*> Sound card support  --->                                                             x
x x                          [*] HID Devices  --->                                                                    x
x x                          [*] USB support  --->                                                                    x
x x                          <*> MMC/SD/SDIO card support  --->                                                       x
x x                          < > Sony MemoryStick card support (EXPERIMENTAL)  --->                                   x
x x                          [*] LED Support  --->                                                                    x
x x                          <*> Switch class support  --->                                                           x
x x                          [ ] Accessibility support  --->                                                          x
x x                          [*] Real Time Clock  --->                                                                x
x x                          [ ] DMA Engine support  --->                                                             x
x x                          [ ] Auxiliary Display support  --->                                                      x
x x                          < > Userspace I/O drivers  --->                                                          x
x x                              Virtio drivers  --->                                                                 x
x x                          [*] Staging drivers  --->                                                                x
x x                          [*] IOMMU Hardware Support  --->                                                         x
x x                          [ ] Virtualization drivers  --->                                                         x
x x                              iTV multimedia devices  --->                                                         x
x x                          [ ] GPS Driver  --->                                                                     x
x x                              Near Field Communication (NFC) devices  --->                                         x
```



```
 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

                       < > Direct Rendering Manager (XFree86 4.1.0 and higher DRI support)  --->
                       < > Ion Memory Manager  --->
                       <*> Mali GPU device driver
                       [ ]   Use GPU bus scaling
                       [*]   Use OS memory
                       (512) Mali memory size
                       [*] Enable Mali profiling
                       [ ]   Enable internal Mali profiling API
                       <*> UMP device driver
                       [*]   Use OS memory
                       (0)   UMP memory address
                       (72)  UMP memory size
                       <*> Lowlevel video output switch controls
                       <*> Support for frame buffer devices  --->
                       [*] Telechips TCC Frame buffer support
                       [ ] Telelechips MIPI DSI controller
                       [*] Telechips TCC support VIOC controller
                       [*] Use VSYNC interrupt
                       [*] Support External display device over HWC v1.1 on MID
                       [*] FB display device display making snapshot image
                       [*] Displaying an image during Quickboot
                             Please choose one option for display (using last framebuffer)  --->
                       <*> Overlay for Camera/Video
                       <*> Overlay_ext for Camera/Video
                       [*] Displaying video frame by hw vsync interrupt
                       [*]   LCD display video by hw vsync interrupt
                       [*]   Support Interlaced Video
                       [ ] FB limit clock high
                       [*] VIOC FIFO-Under-Run Compensation
                           Support for LCD panels  --->
                           TCC Extend Display  --->
                       [ ] Backlight & LCD device support  --->
                           Display device support  --->
                           Console display driver support  --->
                       v(+)

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

6.7.3 Using an image what you want to use

You can choose any image for agree image

6.7.3.1 How to make an image

You can prepare an image as same way with quick boot logo.

```
$ cd ~/mydroid/android/
$ vi device/telechips/tcc893x/device.mk
```

```
$ convert -depth 8 user1.bmp rgb:user1_1024x600.raw
$ rgb2565 -rle <user1_1024x600.raw> user1_1024x600.rle
```

Copy the above image into `android/device/tetelchips/tcc893x`
And you have to modify `device.mk to display this image as agree image.`

```
PRODUCT_COPY_FILES += \
       device/telechips/common/initlogo1024x600.rle:root/initlogo.rle  \
       device/telechips/common/Quickboot_1024x600.rle:root/QuickBoot_logo.rle \
       device/telechips/common/user1_1024x600.rle:root/user1.rle \ //(just example)
       device/telechips/common/user2_1024x600.rle:root/user2.rle   //(just example)
```

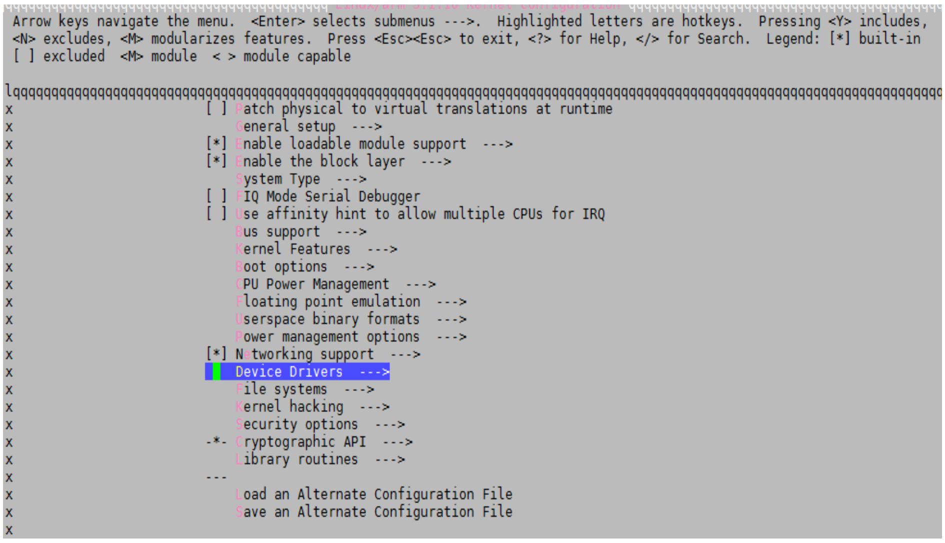
Image file name is defined at If `android/kernel/drivers/video/tcc/tcc_qb_fb.c`
If agree logo is not displayed, please check the correct file name as follows.

```
$ cd ~/mydroid/android/
$ vi kernel/drivers/video/tcc/tcc_qb_fb.c
```

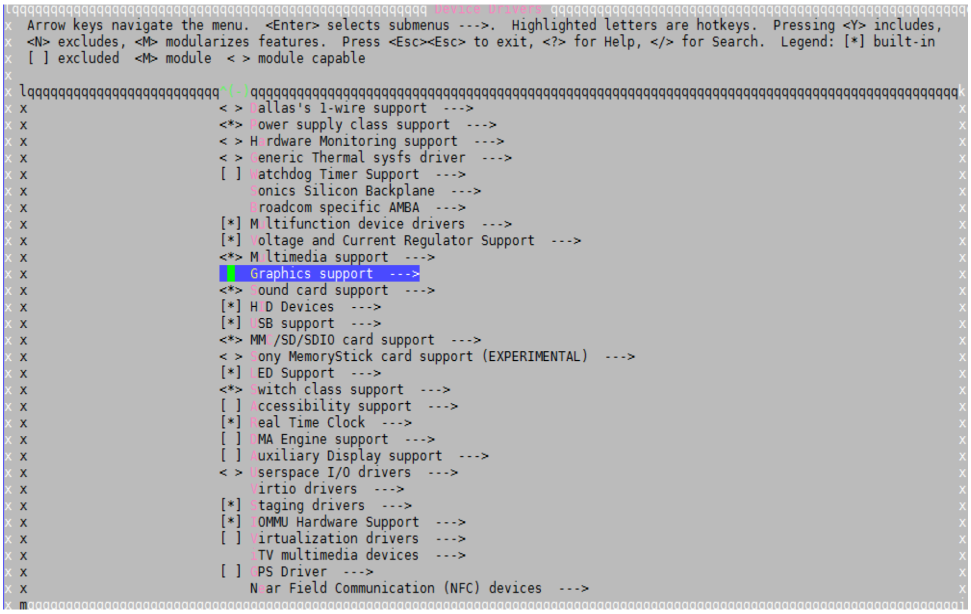
```
#if defined(CONFIG_QUICKBOOT_DISPLAY_LOGO)
#define    QUICKBOOT_USER_LOGO    "user1.rle"
#endif
```

6.7.3.2 Enable Quickboot display user image feature in kernel

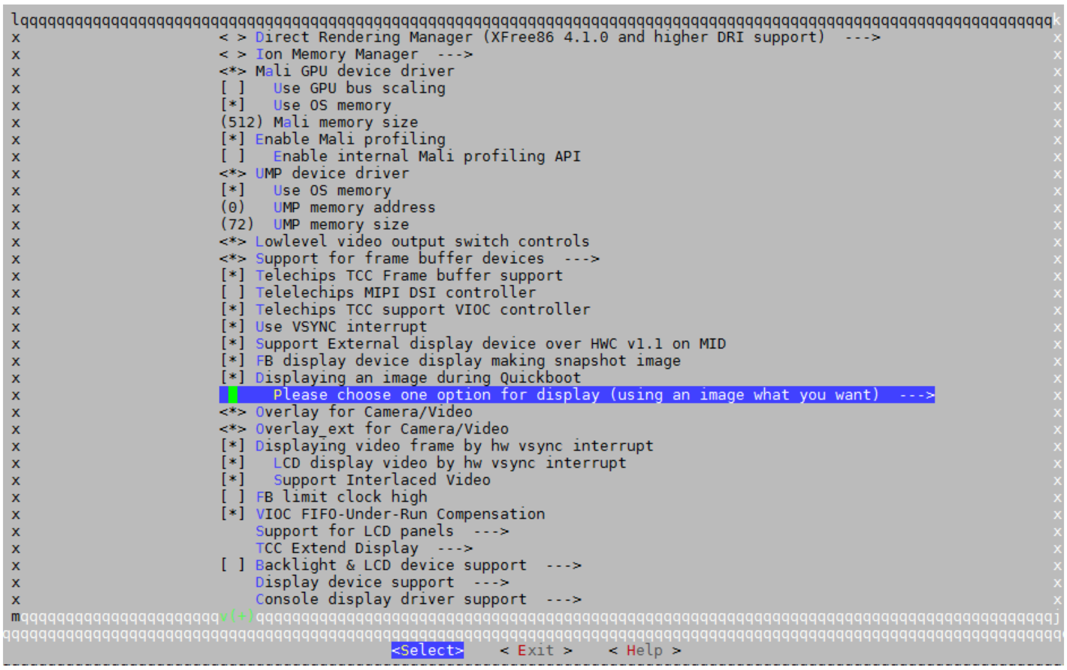
Device drivers → Graphics support → [*] Displaying an image during Quickboot
→ [*] using an image what you want



```
 Linux/arm 3.4.10 Kernel Configuration
 Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted letters are hotkeys.  Pressing <Y> includes,
 <N> excludes, <M> modularizes features.  Press <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend: [*] built-in
 [ ] excluded  <M> module  < > module capable

                          [ ] Patch physical to virtual translations at runtime
                              General setup  --->
                          [*] Enable loadable module support  --->
                          [*] Enable the block layer  --->
                              System Type  --->
                          [ ] FIQ Mode Serial Debugger
                          [ ] Use affinity hint to allow multiple CPUs for IRQ
                              Bus support  --->
                              Kernel Features  --->
                              Boot options  --->
                              CPU Power Management  --->
                              Floating point emulation  --->
                              Userspace binary formats  --->
                              Power management options  --->
                          [*] Networking support  --->
                              Device Drivers  --->
                              File systems  --->
                              Kernel hacking  --->
                              Security options  --->
                          -*- Cryptographic API  --->
                              Library routines  --->
                          ---
                              Load an Alternate Configuration File
                              Save an Alternate Configuration File
```



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

                          ^(-)
                          < > Dallas's 1-wire support  --->
                          <*> Power supply class support  --->
                          < > Hardware Monitoring support  --->
                          < > Generic Thermal sysfs driver  --->
                          [ ] Watchdog Timer Support  --->
                              Sonics Silicon Backplane  --->
                              Broadcom specific AMBA  --->
                          [*] Multifunction device drivers  --->
                          [*] Voltage and Current Regulator Support  --->
                          <*> Multimedia support  --->
                              Graphics support  --->
                          <*> Sound card support  --->
                          [*] HID Devices  --->
                          [*] USB support  --->
                          <*> MMC/SD/SDIO card support  --->
                          < > Sony MemoryStick card support (EXPERIMENTAL)  --->
                          [*] LED Support  --->
                          <*> Switch class support  --->
                          [ ] Accessibility support  --->
                          [*] Real Time Clock  --->
                          [ ] DMA Engine support  --->
                          [ ] Auxiliary Display support  --->
                          < > Userspace I/O drivers  --->
                              Virtio drivers  --->
                          [*] Staging drivers  --->
                          [*] IOMMU Hardware Support  --->
                          [ ] Virtualization drivers  --->
                              iTV multimedia devices  --->
                          [ ] GPS Driver  --->
                              Near Field Communication (NFC) devices  --->
```



```
lqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk
x                      < > Direct Rendering Manager (XFree86 4.1.0 and higher DRI support)  --->
x                      < > Ion Memory Manager  --->
x                      <*> Mali GPU device driver
x                      [ ]   Use GPU bus scaling
x                      [*]   Use OS memory
x                      (512) Mali memory size
x                      [*] Enable Mali profiling
x                      [ ]   Enable internal Mali profiling API
x                      <*> UMP device driver
x                      [*]   Use OS memory
x                      (0)   UMP memory address
x                      (72)  UMP memory size
x                      <*> Lowlevel video output switch controls
x                      <*> Support for frame buffer devices  --->
x                      [*] Telechips TCC Frame buffer support
x                      [ ] Telelechips MIPI DSI controller
x                      [*] Telechips TCC support VIOC controller
x                      [*] Use VSYNC interrupt
x                      [*] Support External display device over HWC v1.1 on MID
x                      [*] FB display device display making snapshot image
x                      [*] Displaying an image during Quickboot
x                            Please choose one option for display (using an image what you want)  --->
x                      <*> Overlay for Camera/Video
x                      <*> Overlay_ext for Camera/Video
x                      [*] Displaying video frame by hw vsync interrupt
x                      [*]   LCD display video by hw vsync interrupt
x                      [*]   Support Interlaced Video
x                      [ ] FB limit clock high
x                      [*] VIOC FIFO-Under-Run Compensation
x                          Support for LCD panels  --->
x                          TCC Extend Display  --->
x                      [ ] Backlight & LCD device support  --->
x                          Display device support  --->
x                          Console display driver support  --->
mqqqqqqqqqqqqqqqqqqqqqqqv(+)qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj
qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq
                                          <Select>    < Exit >    < Help >
```

6.8 How to add “apk” only file with system application

In order to run check disk, we need to un-mount user data & cache partition.
If just add “apk” file only in /system/app folder, used data un-mount issue will be happen.
“apk” file and “odex” file should exist together.
That means, there is no problem, if you use Android Build System.

To avoid this problem, we create /system/app2 folder. The system will scan for the “/system/app2” folder when system boot up with snapshot image.

You will need to insert a “LOCAL_MODULE_APP2 := ENABLE” option in order to copy “apk” only file into the “/system/app2”.

```
LOCAL_PATH := $(call my-dir)
include $(CLEAR_VARS)

# Module name should match apk name to be installed.
LOCAL_MODULE := LocalModuleName
LOCAL_SRC_FILES := $(LOCAL_MODULE).apk
LOCAL_MODULE_CLASS := APPS
LOCAL_MODULE_APP2 := ENABLE
LOCAL_MODULE_SUFFIX := $(COMMON_ANDROID_PACKAGE_SUFFIX)

include $(BUILD_PREBUILT)
```

6.9 How to use Bootchart

# Bootchart On Android : http://www.elinux.org/Using_Bootchart_on_Android

6.9.1 Bootchart on Android QuickBoot

Bootchart isn’t support Quickboot. So, we need to change bootchart source.

1. Add the “bootchart_init” in the system/core/init/init.c in order to work after File System Mount.
2. Modify system/core/init/bootchart.c – bootchart_init() to SKIP bootchart if the system boots on Making QuickBoot Image Mode. Boot Modes depend on “persist.sys.QB.user.allow” & “tcc.QB.boot.with”.
3. Add bootchart.c - get_log_init_jiffies( ) to save the time when Quickboot starts. Then, do_log_uptime( ) uses that time to get how long it takes to Quickboot. So, bootchart can make correct boot log.

6.9.2 How to Install Bootchart on Android

1. Open system/core/init/bootchart.h
2. Change # **define BOOTCHART 0** to **# define BOOTCHART 1**
3. Open system/core/init.c

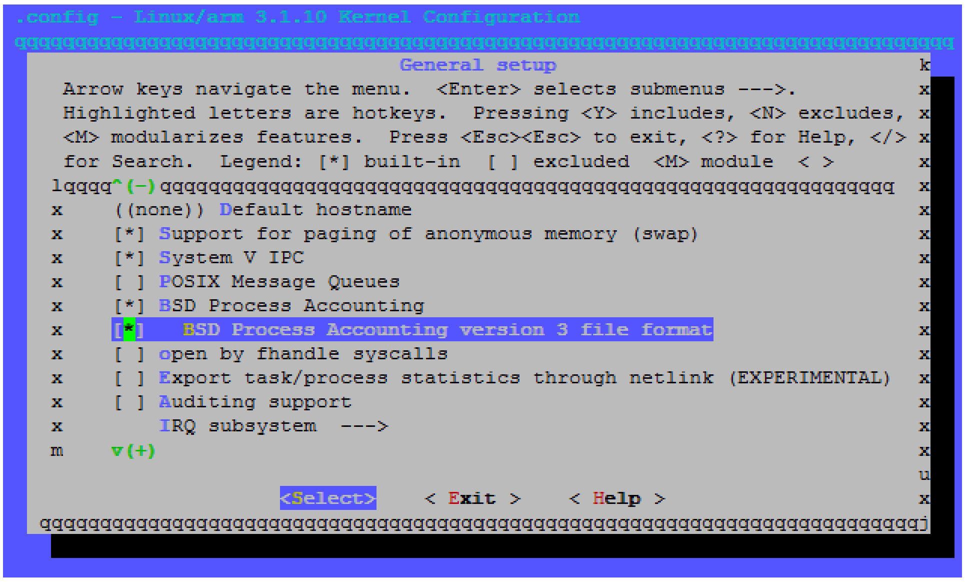
```
#if BOOTCHART
static int bootchart_init_action(int nargs, char **args)
{
   bootchart_count = bootchart_init();
   if (bootchart_count < 0) {
      ERROR("bootcharting init failure\n");
   } else if (bootchart_count > 0) {
      NOTICE("bootcharting started (period=%d ms)\n", bootchart_count*BOOTCHART_POLLING_MS);
   } else {
      NOTICE("bootcharting ignored\n");
   }
return 0;
}
#endif
```

Check **return 0;** If it isn’t, insert it.

4. Kernel Setting

< It needs to change nothing. But if bootchart isn’t work correctly, you need to do follow. >

General Setup → [*] BSD Process Accounting → [*] BSD Process Accounting version 3 file format



```
.config - Linux/arm 3.1.10 Kernel Configuration
qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq
                              General setup                                  k
   Arrow keys navigate the menu.  <Enter> selects submenus --->.              x
   Highlighted letters are hotkeys.  Pressing <Y> includes, <N> excludes,    x
   <M> modularizes features.  Press <Esc><Esc> to exit, <?> for Help, </>    x
   for Search.  Legend: [*] built-in  [ ] excluded  <M> module  < >          x
  lqqqq^(-)qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq    x
  x     ((none)) Default hostname                                            x
  x     [*] Support for paging of anonymous memory (swap)                    x
  x     [*] System V IPC                                                     x
  x     [ ] POSIX Message Queues                                             x
  x     [*] BSD Process Accounting                                           x
  x     [*]   BSD Process Accounting version 3 file format                   x
  x     [ ] open by fhandle syscalls                                         x
  x     [ ] Export task/process statistics through netlink (EXPERIMENTAL)    x
  x     [ ] Auditing support                                                 x
  x         IRQ subsystem  --->                                              x
  m     v(+)                                                                 x
                                                                             u
                     <Select>    < Exit >    < Help >                        x
 qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj
```

Then, build kernel.

5. Move to android source root.
6. $ export INIT_BOOTCHART=true
7. Build Android Source.

6.9.3 How to get log files

1. Turn on Target Board.
2. $ mkdir /data/bootchart
3. $ echo 40 > /data/bootchart-start
   (The number 40 is a variable how long records booting log files.)
4. Reboot
5. Copy all files in /data/bootchart/ to Host PC and compress it to bootchart.tgz

6.9.4 How to change log files to a graph – Ubuntu 12.04 LTS 32bit

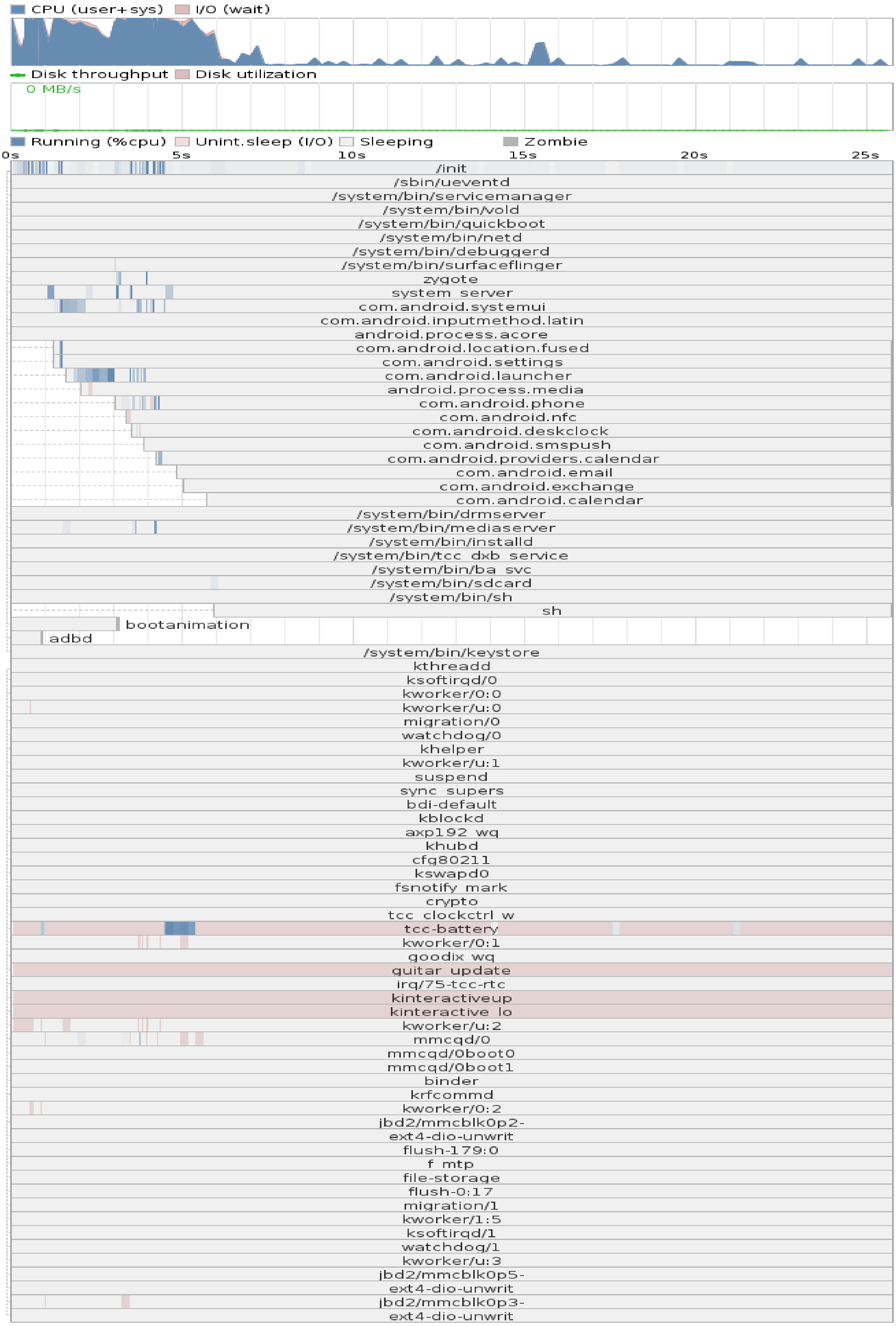
1. Download bootchart Source. **< bootchart-0.9.tar.bz2 >** http://www.bootchart.org/download.html
   < Don't use **apt-get install bootchart** >
2. $ apt-get install ant1.7
3. Install JDK 6 which is distributed by Oracle.
   < If you use openJDK, it can make errors >

```
$ sudo add-apt-repository ppa:webupd8team/java
$ sudo apt-get update
$ sudo apt-get install oracle-java6-installer
```

4. Move to bootchart Source
   Ex) bootchart-0.9 $
5. Compile bootchart-0.9 using ant.
   $ bootchart-0.9 $ ant
6. **bootchart.jar** file is created.
   < Instead of compile bootchart.jar, you can use vender/telechips/tools/bootchart/bootchart-32.jar or bootchart-64.jar >
7. Move bootchart.tgz to bootchart source directory.
8. Create a graph using bootchart.jar and bootchart.tgz
   $ bootchart-0.9 $ java –jar bootchart.jar ./bootchart.tgz
9. You can find bootchart.png in the directory.

6.9.5 QuickBoot Results

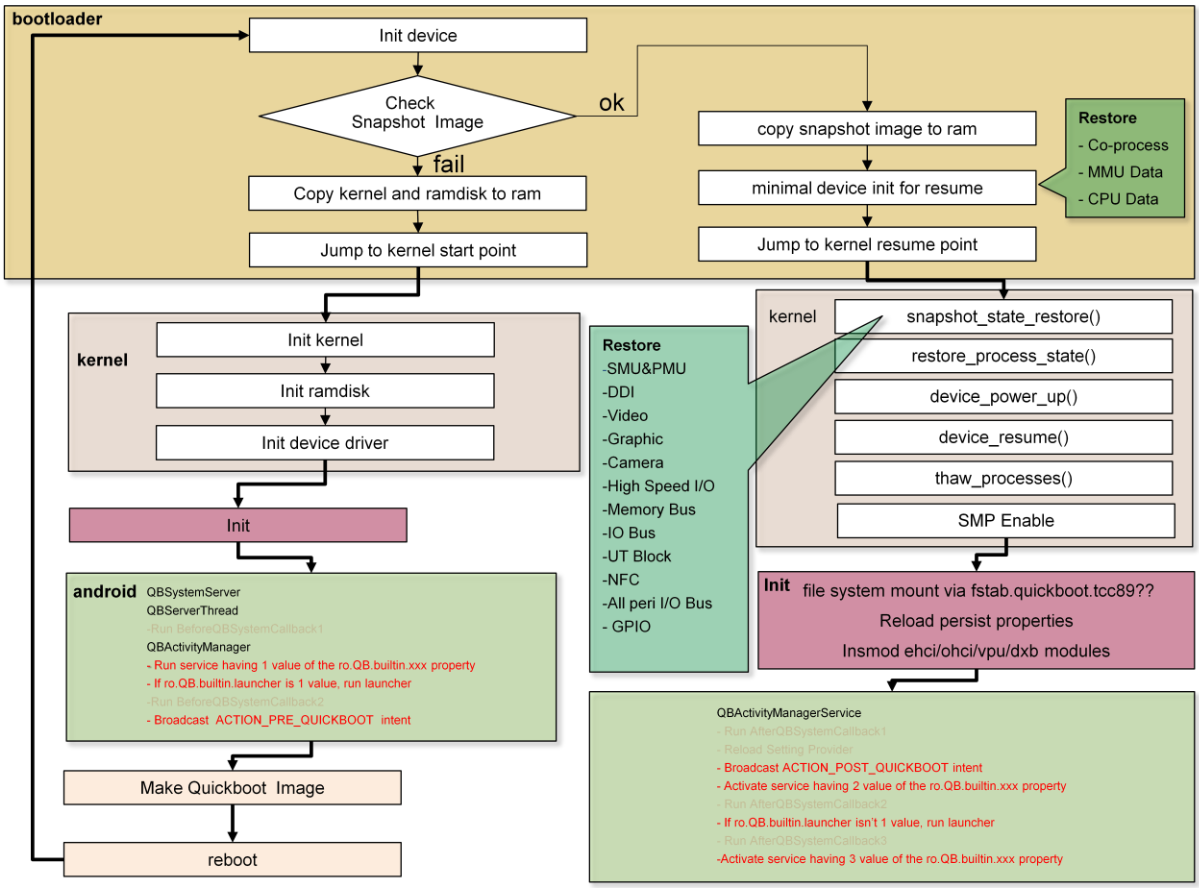
< It doesn't include bootloader's time & snapshot image loading time. >



6.10 How to built-in services on snapshot image

We support ro.QB.builtin.xxx properties in order to built-in default services in quick boot image.

The follow figure is shown how to activate services depending on ro.QB.builtin.xxx. of properties.


bootloader
Init device
Check Snapshot Image
ok
fail
Copy kernel and ramdisk to ram
Jump to kernel start point
copy snapshot image to ram
minimal device init for resume
Jump to kernel resume point
Restore
- Co-process
- MMU Data
- CPU Data
kernel
Init kernel
Init ramdisk
Init device driver
Init
android
QBSystemServer
QBServerThread
-Run BeforeQBSystemCallback1
QBActivityManager
- Run service having 1 value of the ro.QB.builtin.xxx property
- If ro.QB.builtin.launcher is 1 value, run launcher
-Run BeforeQBSystemCallback2
- Broadcast ACTION_PRE_QUICKBOOT intent
Make Quickboot Image
reboot
Restore
-SMU&PMU
-DDI
-Video
-Graphic
-Camera
-High Speed I/O
-Memory Bus
-IO Bus
-UT Block
-NFC
-All peri I/O Bus
- GPIO
kernel
snapshot_state_restore()
restore_process_state()
device_power_up()
device_resume()
thaw_processes()
SMP Enable
Init
file system mount via fstab.quickboot.tcc89??
Reload persist properties
Insmod ehci/ohci/vpu/dxb modules
QBActivityManagerService
- Run AfterQBSystemCallback1
- Reload Setting Provider
- Broadcast ACTION_POST_QUICKBOOT intent
- Activate service having 2 value of the ro.QB.builtin.xxx property
- Run AfterQBSystemCallback2
- If ro.QB.builtin.launcher isn't 1 value, run launcher
- Run AfterQBSystemCallback3
-Activate service having 3 value of the ro.QB.builtin.xxx property

If you want to add default services in quickboot, enable and modify ro.QB.builtin.xxxx property in the device/telechips/tcc893x/device.mk.

Note >>
If you change ro.QB.builtin.xxx properties, we cannot support reloading services.
Please note.

device/telechips/common/common.mk

```
…

PRODUCT_PROPERTY_OVERRIDES += ro.QB.builtin.windowManager = 2

#PRODUCT_PROPERTY_OVERRIDES += ro.QB.builtin.widget=1
#PRODUCT_PROPERTY_OVERRIDES += ro.QB.builtin.wallpaper=1
#PRODUCT_PROPERTY_OVERRIDES += ro.QB.builtin.launcher=1

…
```

You can choose value of ro.QB.builtin.xxx from 0 to 3.

0 means that a service is deactivated.
1 means that a service is activated before making quick boot image.
2 mean that a service is activated after boot with snapshot image.
3 mean that a service is activated after running launcher.

Supported ro.QB.builtin.xxxx properties

name	range	Description
ro.QB.builtin.widget	1 ~ 2	Activate widget (2 is default value, if corresponding property isn't set any value)
ro.QB.builtin.wallpaper	1 ~ 2	Activate wallpaper (2 is default value, if corresponding property isn't set any value) If this value is set into '1', following issues related to reloading service will be occured. 1) Changed Wallpaper setting is restored after booting.
ro.QB.builtin.launcher	1 ~ 2	Activate default home before making snapshot image or after booting with snapshot image. - 2 mean activating default home after booting with snapshot image. (2 is default value, if corresponding property isn't set any value) - 1 means activating default home before making snapshot image. If this value is set into '1', following issues related to reloading service will be occured. 1) The icon of setting widget in idle is disappeared after booting. 2) The icon of installed apk is disappeared after booting. 3) The widget added in idle is disappeared after booting. 4) The Guide-screen on first booting is appeared repeatedly whenever booting. (if quickboot image was created with the Guide-screen)
ro.QB.builtin.windowManager	1 ~ 2	Activate windowManager before making snapshot image or after booting with snapshot image. - 1 means activating windowManager before making snapshot image. (1 is default value, if corresponding property isn't set any value) if this value is set into '1', should create the snapshot image with setting lock-screen into NONE. - 2 mean activating windowManager after booting with snapshot image.

6.11 Quickboot Pre-built Usage

Quickboot pre-built solution is devised to reduce time of making quickboot image one by one for every devices. The principle of this is to extract several partitions(userdata, cache, snapshot) needed quickboot, and then write the extracted partition files to new devices in fwdn tool.

6.11.1 Make Quickboot Image

1. Partition Configuration

   In this quickboot pre-built solution, partition number 11 is newly added. The role of the added partition is to save extracted image associated with quickboot(userdata, cache, snapshot). The added partition data is used in factory reset to initialize device.

   Depending on storage device size, the size of partition number 11 is changed. In other word, the added partition size depends on size of extracted quickboot image. In this guide, storage device is used to 4GByte device with about 2.4GByte userdata partition which is the biggest partition.

   a) Partition Configuration in FWDN Tool

   Add partition number in “Number of Partition”, and then set the added partition configuration like following figure.

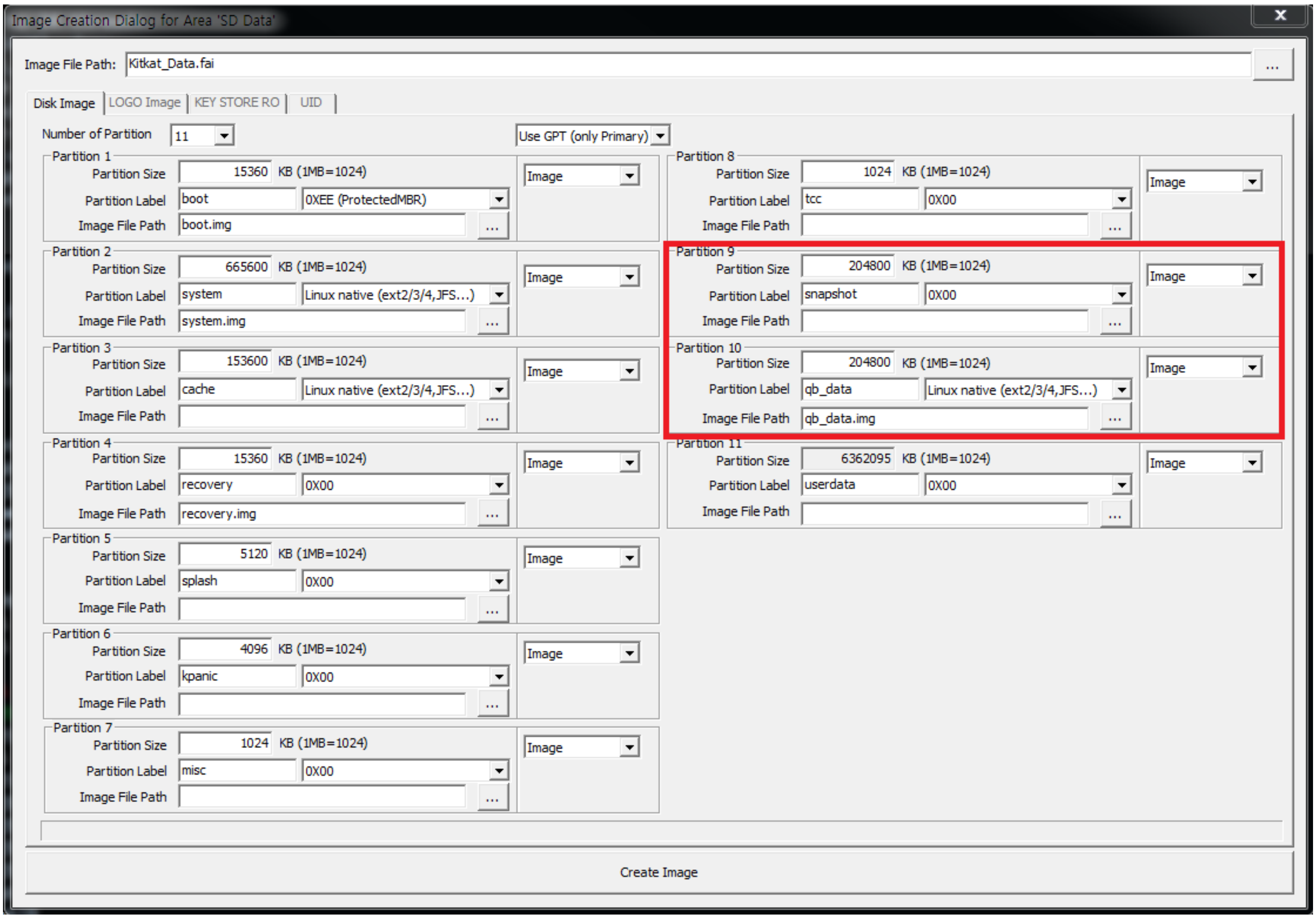


Image File Path: Kitkat_Data.fai

Number of Partition: 11

Use GPT (only Primary)

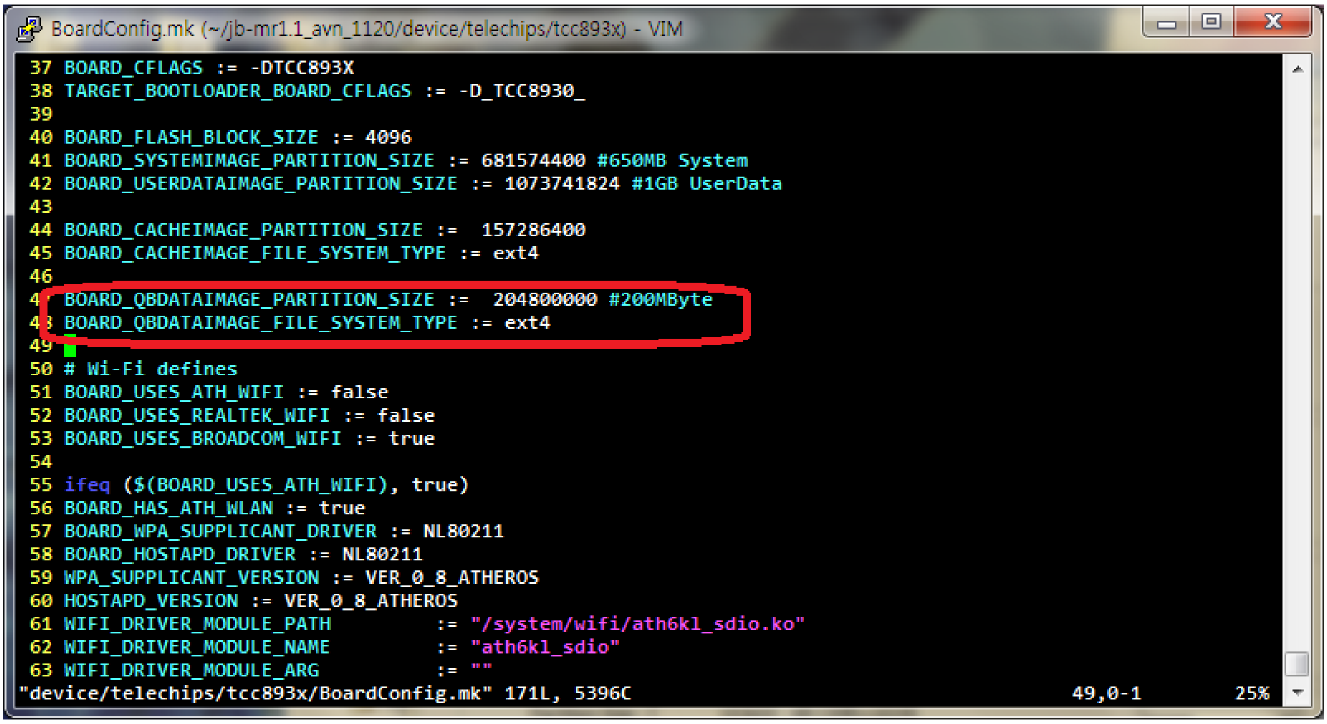
	Partition Size		Partition Label		Image File Path	
Partition 1	15360	KB (1MB=1024)	boot	0XEE (ProtectedMBR)	boot.img	Image
Partition 2	665600	KB (1MB=1024)	system	Linux native (ext2/3/4,JFS...)	system.img	Image
Partition 3	153600	KB (1MB=1024)	cache	Linux native (ext2/3/4,JFS...)		Image
Partition 4	15360	KB (1MB=1024)	recovery	0X00	recovery.img	Image
Partition 5	5120	KB (1MB=1024)	splash	0X00		Image
Partition 6	4096	KB (1MB=1024)	kpanic	0X00		Image
Partition 7	1024	KB (1MB=1024)	misc	0X00		Image
Partition 8	1024	KB (1MB=1024)	tcc	0X00		Image
Partition 9	204800	KB (1MB=1024)	snapshot	0X00		Image
Partition 10	204800	KB (1MB=1024)	qb_data	Linux native (ext2/3/4,JFS...)	qb_data.img	Image
Partition 11	6362095	KB (1MB=1024)	userdata	0X00		Image

   b) Make “qb_data.img”

   Before to make quickboot image to be extracted, partition 11 must be formatted as ext4 type. To make “qb_data.img” automatically, set following parameters:

   >> Path : device/telechips/<device>/BoardConfig.mk
   >> Parameter :

   + BOARD_QBDATAIMAGE_PARTITION_SIZE := 204800000
   + BOARD_QBDATAIMAGE_FILE_SYSTEM_TYPE := ext4

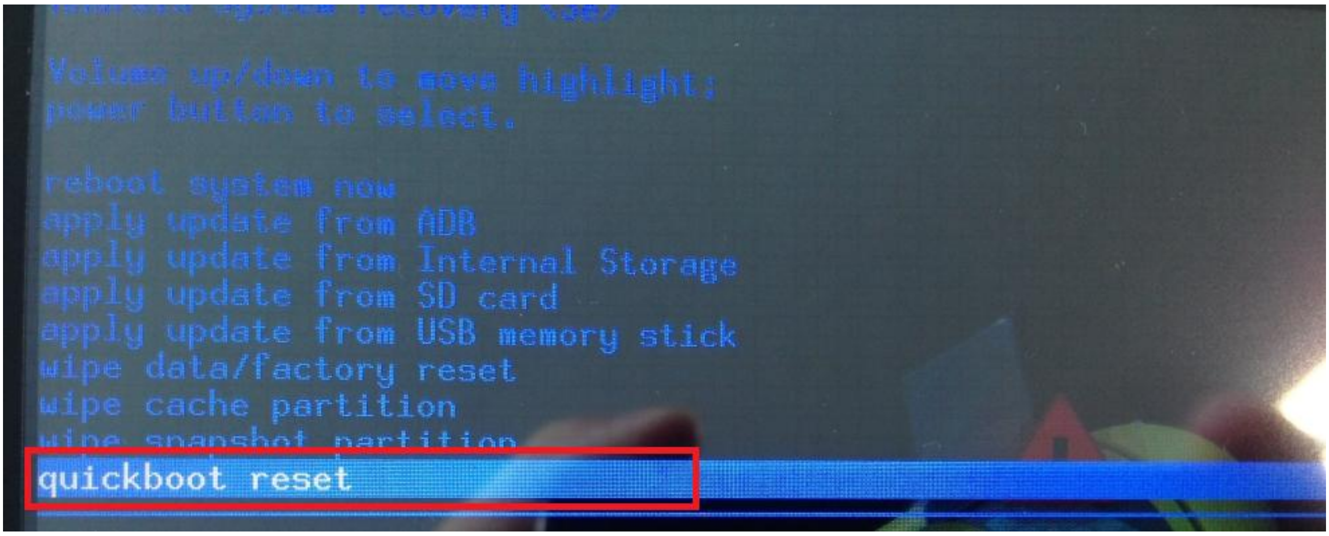

BoardConfig.mk (~/jb-mr1.1_avn_1120/device/telechips/tcc893x) - VIM

```
BOARD_CFLAGS := -DTCC893X
TARGET_BOOTLOADER_BOARD_CFLAGS := -D_TCC8930_

BOARD_FLASH_BLOCK_SIZE := 4096
BOARD_SYSTEMIMAGE_PARTITION_SIZE := 681574400 #650MB System
BOARD_USERDATAIMAGE_PARTITION_SIZE := 1073741824 #1GB UserData

BOARD_CACHEIMAGE_PARTITION_SIZE :=  157286400
BOARD_CACHEIMAGE_FILE_SYSTEM_TYPE := ext4

BOARD_QBDATAIMAGE_PARTITION_SIZE :=  204800000 #200MByte
BOARD_QBDATAIMAGE_FILE_SYSTEM_TYPE := ext4

# Wi-Fi defines
BOARD_USES_ATH_WIFI := false
BOARD_USES_REALTEK_WIFI := false
BOARD_USES_BROADCOM_WIFI := true

ifeq ($(BOARD_USES_ATH_WIFI), true)
BOARD_HAS_ATH_WLAN := true
BOARD_WPA_SUPPLICANT_DRIVER := NL80211
BOARD_HOSTAPD_DRIVER := NL80211
WPA_SUPPLICANT_VERSION := VER_0_8_ATHEROS
HOSTAPD_VERSION := VER_0_8_ATHEROS
WIFI_DRIVER_MODULE_PATH         := "/system/wifi/ath6kl_sdio.ko"
WIFI_DRIVER_MODULE_NAME         := "ath6kl_sdio"
WIFI_DRIVER_MODULE_ARG          := ""
"device/telechips/tcc893x/BoardConfig.mk" 171L, 5396C                         49,0-1        25%
```

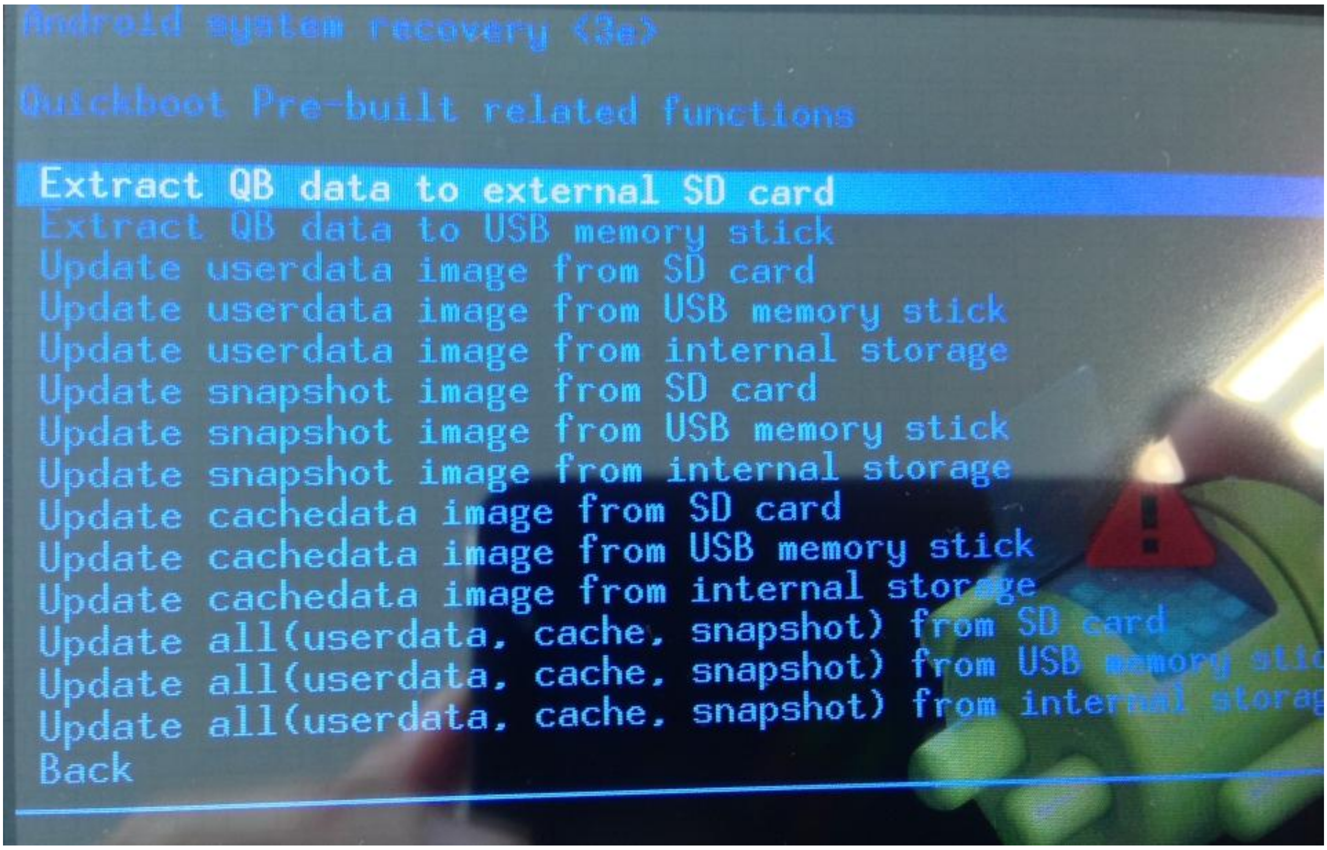
6.11.2 Excute Quickboot Pre-built Solution

1. Extract Quickboot Image

   a) Enter Recovery Mode
      To extract quickboot image, go to recovery mode, and then select “quickboot reset” menu.

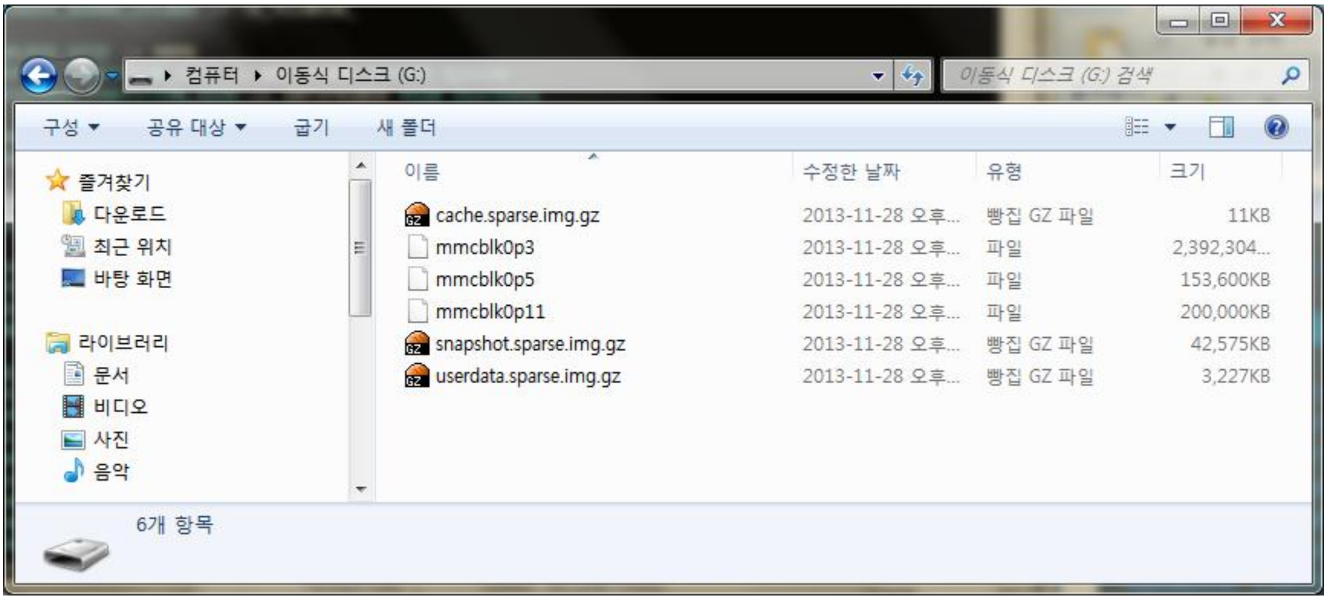


b) Extract Quickboot Image
Extract quickboot image to external SD card or USB memory stick.



```
Android system recovery <3e>

Quickboot Pre-built related functions

Extract QB data to external SD card
Extract QB data to USB memory stick
Update userdata image from SD card
Update userdata image from USB memory stick
Update userdata image from internal storage
Update snapshot image from SD card
Update snapshot image from USB memory stick
Update snapshot image from internal storage
Update cachedata image from SD card
Update cachedata image from USB memory stick
Update cachedata image from internal storage
Update all(userdata, cache, snapshot) from SD card
Update all(userdata, cache, snapshot) from USB memory stic
Update all(userdata, cache, snapshot) from internal storag
Back
```

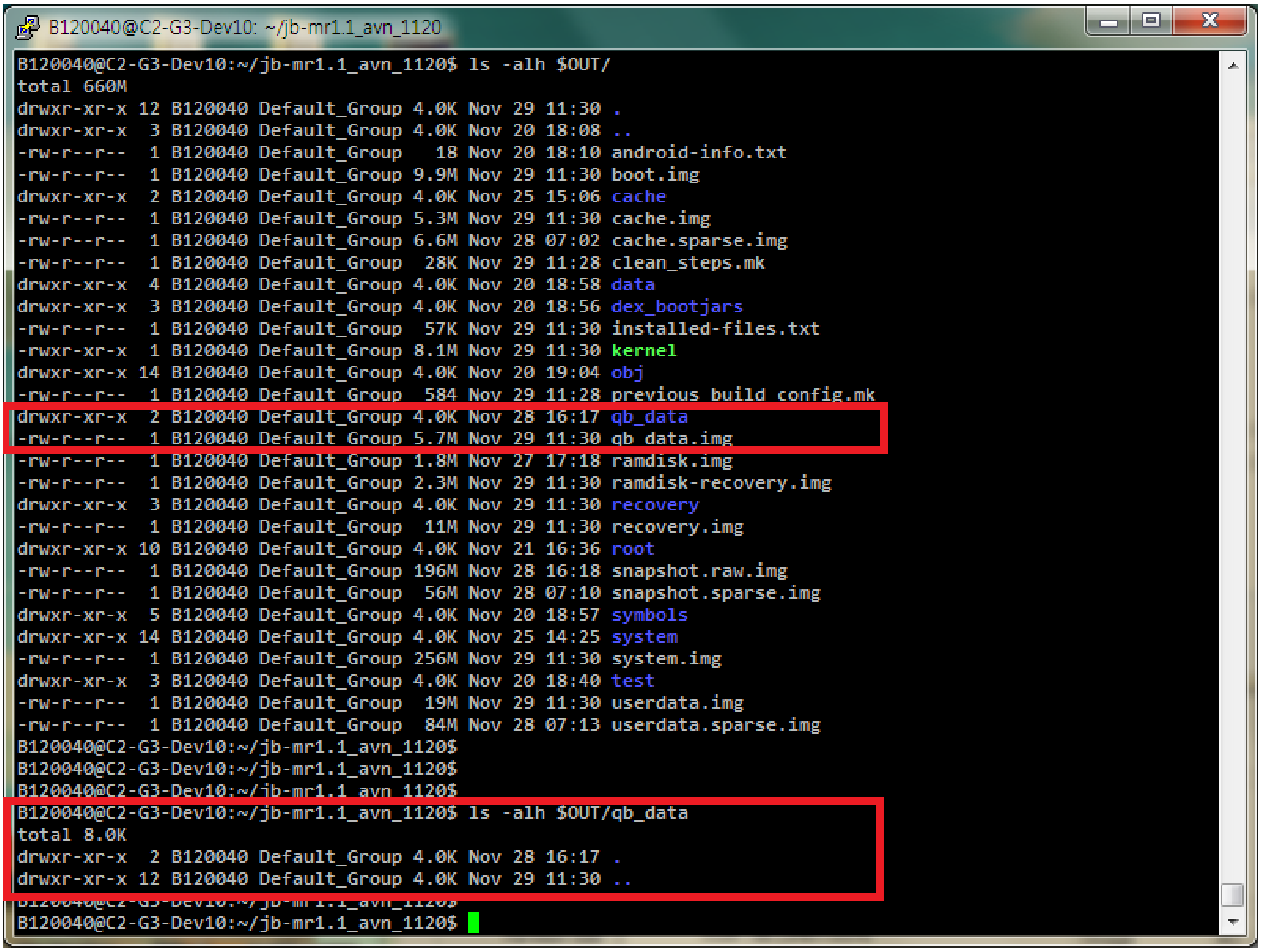
c) Convert Extracted Quickboot Image
After quickboot images are extracted, following files are shown.
In this list, “userdata.sparse.img.gz”, “cache.sparse.img.gz”, and “snapshot.sparse.img.gz” files are used.



이름	수정한 날짜	유형	크기
cache.sparse.img.gz	2013-11-28 오후...	빵집 GZ 파일	11KB
mmcblk0p3	2013-11-28 오후...	파일	2,392,304...
mmcblk0p5	2013-11-28 오후...	파일	153,600KB
mmcblk0p11	2013-11-28 오후...	파일	200,000KB
snapshot.sparse.img.gz	2013-11-28 오후...	빵집 GZ 파일	42,575KB
userdata.sparse.img.gz	2013-11-28 오후...	빵집 GZ 파일	3,227KB

6개 항목

① Re-make “qb_data.img” with Extracted Files
In ANDROID_OUT directory, “qb_data.img” and “qb_data” directory are shown like following figure.
There is no any data inside “qb_data” directory.
To Make “qb_data.img” with extracted quickboot images, copy “userdata.sparse.img.gz”, “cache.sparse.img.gz”, and “snapshot.sparse.img.gz” to “qb_data” directory, and then make it again with make command or do following command.
>> command : make_ext4fs –s –l <size:204800000> -a qb_data <output:qb_data.img> <target directory:qb_data>



```
B120040@C2-G3-Dev10:~/jb-mr1.1_avn_1120$ ls -alh $OUT/
total 660M
drwxr-xr-x 12 B120040 Default_Group 4.0K Nov 29 11:30 .
drwxr-xr-x  3 B120040 Default_Group 4.0K Nov 20 18:08 ..
-rw-r--r--  1 B120040 Default_Group   18 Nov 20 18:10 android-info.txt
-rw-r--r--  1 B120040 Default_Group 9.9M Nov 29 11:30 boot.img
drwxr-xr-x  2 B120040 Default_Group 4.0K Nov 25 15:06 cache
-rw-r--r--  1 B120040 Default_Group 5.3M Nov 29 11:30 cache.img
-rw-r--r--  1 B120040 Default_Group 6.6M Nov 28 07:02 cache.sparse.img
-rw-r--r--  1 B120040 Default_Group  28K Nov 29 11:28 clean_steps.mk
drwxr-xr-x  4 B120040 Default_Group 4.0K Nov 20 18:58 data
drwxr-xr-x  3 B120040 Default_Group 4.0K Nov 20 18:56 dex_bootjars
-rw-r--r--  1 B120040 Default_Group  57K Nov 29 11:30 installed-files.txt
-rwxr-xr-x  1 B120040 Default_Group 8.1M Nov 29 11:30 kernel
drwxr-xr-x 14 B120040 Default_Group 4.0K Nov 20 19:04 obj
-rw-r--r--  1 B120040 Default_Group  584 Nov 29 11:28 previous_build_config.mk
drwxr-xr-x  2 B120040 Default_Group 4.0K Nov 28 16:17 qb_data
-rw-r--r--  1 B120040 Default_Group 5.7M Nov 29 11:30 qb_data.img
-rw-r--r--  1 B120040 Default_Group 1.8M Nov 27 17:18 ramdisk.img
-rw-r--r--  1 B120040 Default_Group 2.3M Nov 29 11:30 ramdisk-recovery.img
drwxr-xr-x  3 B120040 Default_Group 4.0K Nov 29 11:30 recovery
-rw-r--r--  1 B120040 Default_Group  11M Nov 29 11:30 recovery.img
drwxr-xr-x 10 B120040 Default_Group 4.0K Nov 21 16:36 root
-rw-r--r--  1 B120040 Default_Group 196M Nov 28 16:18 snapshot.raw.img
-rw-r--r--  1 B120040 Default_Group  56M Nov 28 07:10 snapshot.sparse.img
drwxr-xr-x  5 B120040 Default_Group 4.0K Nov 20 18:57 symbols
drwxr-xr-x 14 B120040 Default_Group 4.0K Nov 25 14:25 system
-rw-r--r--  1 B120040 Default_Group 256M Nov 29 11:30 system.img
drwxr-xr-x  3 B120040 Default_Group 4.0K Nov 20 18:40 test
-rw-r--r--  1 B120040 Default_Group  19M Nov 29 11:30 userdata.img
-rw-r--r--  1 B120040 Default_Group  84M Nov 28 07:13 userdata.sparse.img
B120040@C2-G3-Dev10:~/jb-mr1.1_avn_1120$
B120040@C2-G3-Dev10:~/jb-mr1.1_avn_1120$
B120040@C2-G3-Dev10:~/jb-mr1.1_avn_1120$
B120040@C2-G3-Dev10:~/jb-mr1.1_avn_1120$ ls -alh $OUT/qb_data
total 8.0K
drwxr-xr-x  2 B120040 Default_Group 4.0K Nov 28 16:17 .
drwxr-xr-x 12 B120040 Default_Group 4.0K Nov 29 11:30 ..
B120040@C2-G3-Dev10:~/jb-mr1.1_avn_1120$
B120040@C2-G3-Dev10:~/jb-mr1.1_avn_1120$
```

② Decompress Quickboot Pre-built Image
Decompress extracted quickboot images(“userdata.spase.img.gz”, “cache.sparse.img.gz”, “snapshot.sparse.img.gz”) with “gunzip” process in host PC.
Specially, snapshot image has to be coverted from sparse image to raw image after decompression. To do this, execute following command one more.
>> command : simg2img <in:snapshot.sparse.img> <out:snapshot.raw.img>

6.11.3 Execute FWDN with Quickboot Pre-built Image

Choose extracted quickboot image in FWDN tool like following figure. This example is based on 4GByte eMMC storage. Generally, depending on each system configuration and storage capacity, partition size of number 3, 10, and 11 varies. But, each partition size of number 3, 4, 10, and 11 has to be identically set on every device when quickboot pre-built solution is used.

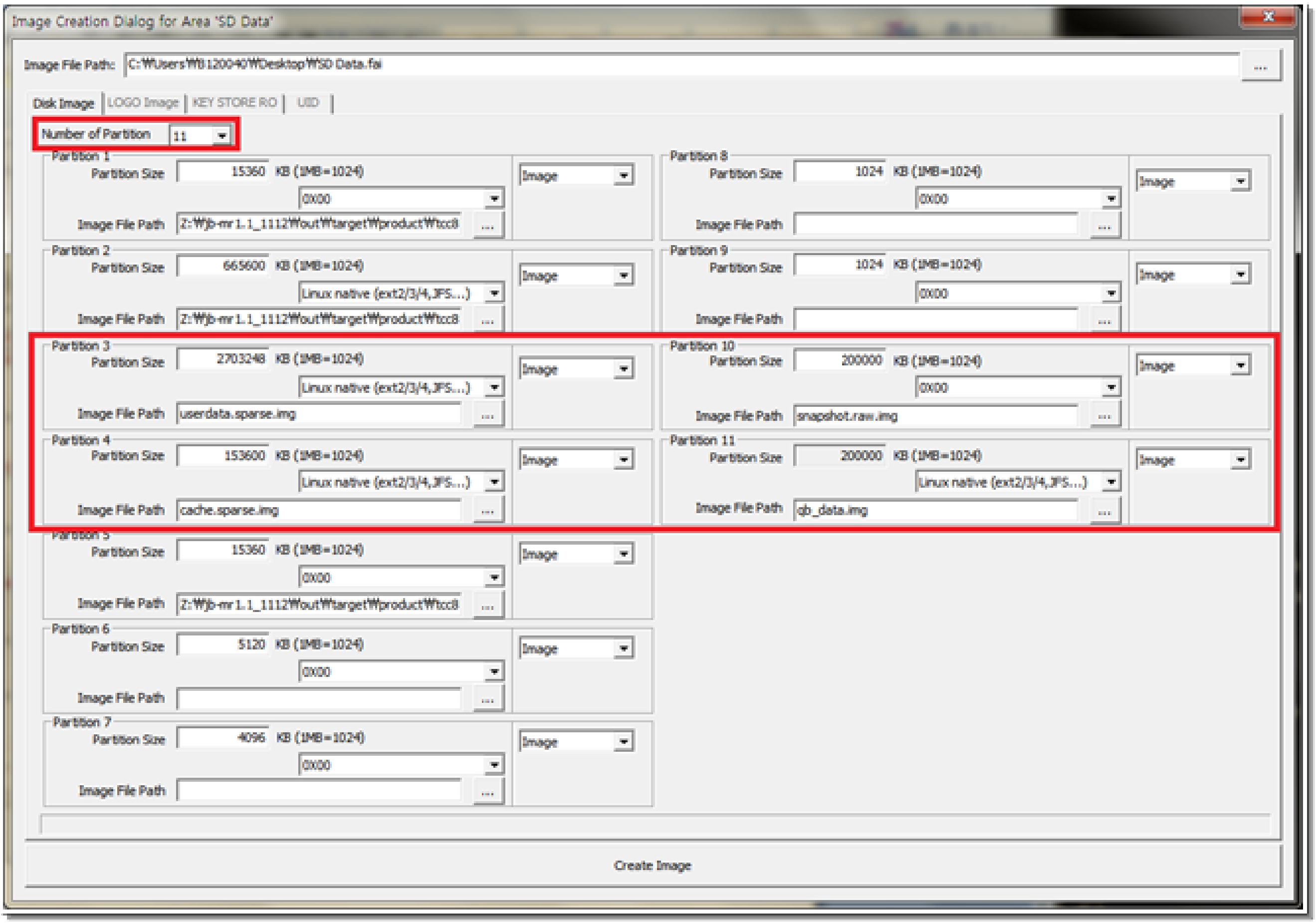


Image File Path: C:₩Users₩B120040₩Desktop₩SD Data.fai

Number of Partition: 11

Partition 1
Partition Size: 15360 KB (1MB=1024)
0X00
Image File Path: Z:₩jb-mr1.1_1112₩out₩target₩product₩tcc8
Image

Partition 2
Partition Size: 665600 KB (1MB=1024)
Linux native (ext2/3/4,JFS...)
Image File Path: Z:₩jb-mr1.1_1112₩out₩target₩product₩tcc8
Image

Partition 3
Partition Size: 2703248 KB (1MB=1024)
Linux native (ext2/3/4,JFS...)
Image File Path: userdata.sparse.img
Image

Partition 4
Partition Size: 153600 KB (1MB=1024)
Linux native (ext2/3/4,JFS...)
Image File Path: cache.sparse.img
Image

Partition 5
Partition Size: 15360 KB (1MB=1024)
0X00
Image File Path: Z:₩jb-mr1.1_1112₩out₩target₩product₩tcc8
Image

Partition 6
Partition Size: 5120 KB (1MB=1024)
0X00
Image File Path:
Image

Partition 7
Partition Size: 4096 KB (1MB=1024)
0X00
Image File Path:
Image

Partition 8
Partition Size: 1024 KB (1MB=1024)
0X00
Image File Path:
Image

Partition 9
Partition Size: 1024 KB (1MB=1024)
0X00
Image File Path:
Image

Partition 10
Partition Size: 200000 KB (1MB=1024)
0X00
Image File Path: snapshot.raw.img
Image

Partition 11
Partition Size: 200000 KB (1MB=1024)
Linux native (ext2/3/4,JFS...)
Image File Path: qb_data.img
Image

6.12 How to get QuickBoot Time

You can check QuickBoot Time from bootloader and android logs.

6.12.1 Bootloader QuickBoot Time.

You can see the log below by UART.

```
[1627]load_image_from_emmc.... meta[      27] copy[ 27585] index[27580]
[1633]time(ms)... totoal[1103]   memcpy[ 122] lz4[ 463] read snapshot image[ 505][45MBps] device_init[ 530]
```

You need to add the time in bootloader log.

> **read snapshot image [   ]**

> **memcpy [   ]**

> **lz4 decompress [   ]**

> **device init [   ]**

> **chip boot** : It’s the time of excuting Chip Boot Code. It cannot show log but it’s almost 901ms.

So, [read snapshot image] + [memcpy] + [lz4 decompress] + [device init] + [chip boot] = [QuickBoot Time of Bootloader]

6.12.2 Android + Kernel QuickBoot Time.

To see the log of QuickBoot Time, you need to type this command after android booting.

>> command : logcat -v time | grep quickboot

Then, the QuickBoot log is shown by UART.

```
01-28 04:01:05.650 I/log       ( 2179): quickboot profile kernel - total [800ms], resume [600ms] chkdsk [30ms] etc [170ms]
dmp resume[570ms]
01-28 04:01:05.840 D/ActivityManager( 1768): quickboot profile hibernated proc :: ProcessRecord{410ad9c0
1768:system/1000}
01-28 04:01:05.840 I/ActivityManager( 1768): @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ quickboot
service version : v1.021, booting count :2############################
01-28 04:01:05.840 I/ActivityManager( 1768): quickboot goingAfterCallback1   : 0 ms
01-28 04:01:06.200 I/ActivityManager( 1768): quickboot reloading setting provider   : 359 ms
01-28 04:01:06.680 I/QBSystemServerCallback( 1768): quickboot profile pm 476ms
01-28 04:01:07.620 I/QBSystemServerCallback( 1768): quickboot profile app widget 879ms
01-28 04:01:07.930 I/ActivityManager( 1768): quickboot goingCallback1   : 1729 ms
01-28 04:01:08.380 I/ActivityManager( 1768): quickboot goingAfterCallback2   : 454 ms
01-28 04:01:08.380 I/ActivityManager( 1768): quickboot reload services   : 2183 ms
01-28 04:01:11.250 I/ActivityManager( 1768): quickboot profile launcher & bootanim   : 2716 ms
01-28 04:01:11.820 I/ActivityManager( 1768): quickboot goingAfterCallback3   : 556 ms
01-28 04:01:11.820 I/ActivityManager( 1768): quickboot framework   : 5975 ms
01-28 04:01:12.070 I/ActivityManager( 1768): quickboot gogingCallback2   : 195 ms
```

You need to add the time in the first line of Android log for **kernel QuickBoot Time**.

> **resume : Restore CPU + Coprocessor + Cache + Device + etc….**

> **chkdsk : e2fsck (Check Disk)**

> **etc : Thaw_processes + rtc + Enable CPUs + late_resume + etc…**

So, [resume] + [chkdsk] + [etc] = [QuickBoot Time of kernel]

You need to add the time in Android log for **Android Frameworks QuickBoot Time**.

> **quickboot reloading setting provider**

> **quickboot reload services**

> **launcher & bootanim**

So, [quickboot reloading setting provider] + [quickboot reload services] + [launcher & bootanim]
= [QuickBoot Time of Android Frameworks]

Finally, [Total QuickBoot Time] is
[bootloader QuickBoot Time] + [Kernel QuickBoot Time] + [Android QuickBoot Time]

※ QuickBoot Time can be different by “Lock Screen Option” and “Setting target board horizontality or verticality"