Android Jelly Bean BSP Porting Guide

: MSM8960, Fusion Chipset

chunghan.yi@gmail.com, slowboot

목차

Jelly Bean File Download

from Code Aurora Forum

1. Jelly Bean Code Download & Build(1) – ***Android full source***

1) Jelly Bean code download

 # curl https://dl-ssl.google.com/dl/googlesource/git-repo/repo > ~/CAF/bin/repo

 # export PATH= ~/CAF/bin:$PATH

 # repo init -u git://codeaurora.org/platform/manifest.git -b jb -m M8960AAAAANLYA2205.xml --repo-url=git://codeaurora.org/tools/repo.git

 # repo sync -j4

(*) vendor 디렉토리를 제외하면, 상용 버전을 위한 코드와 동일함.

2) Jelly Bean build

 # source build/envsetup.sh

 # choosecombo

1 ← release

msm8960

eng

 # make –j4

➔ *out/target/product/msm8960/emmc_appsboot.mbn, boot.img, cache.img, persist.img, ramdisk.img, recovery.img, system.img, userdata.img*

For more information, see the following site.
➔ *https://www.codeaurora.org/xwiki/bin/QAEP/*

1. Jelly Bean Code Download & Build(2) – ***msm kernel code***

1) Kernel code download

```
# git clone git://codeaurora.org/kernel/msm.git
# cd msm
# git branch
  *android-msm-2.6.29
  jb_chocolate
  msm-3.4
# git checkout jb_chocolate
or
# git checkout msm-3.4
```

Kernel 만 download 할 경우의 예이며, 앞서 Android를 download 받았을 경우, 불필요한 작업임^^

2) Kernel build

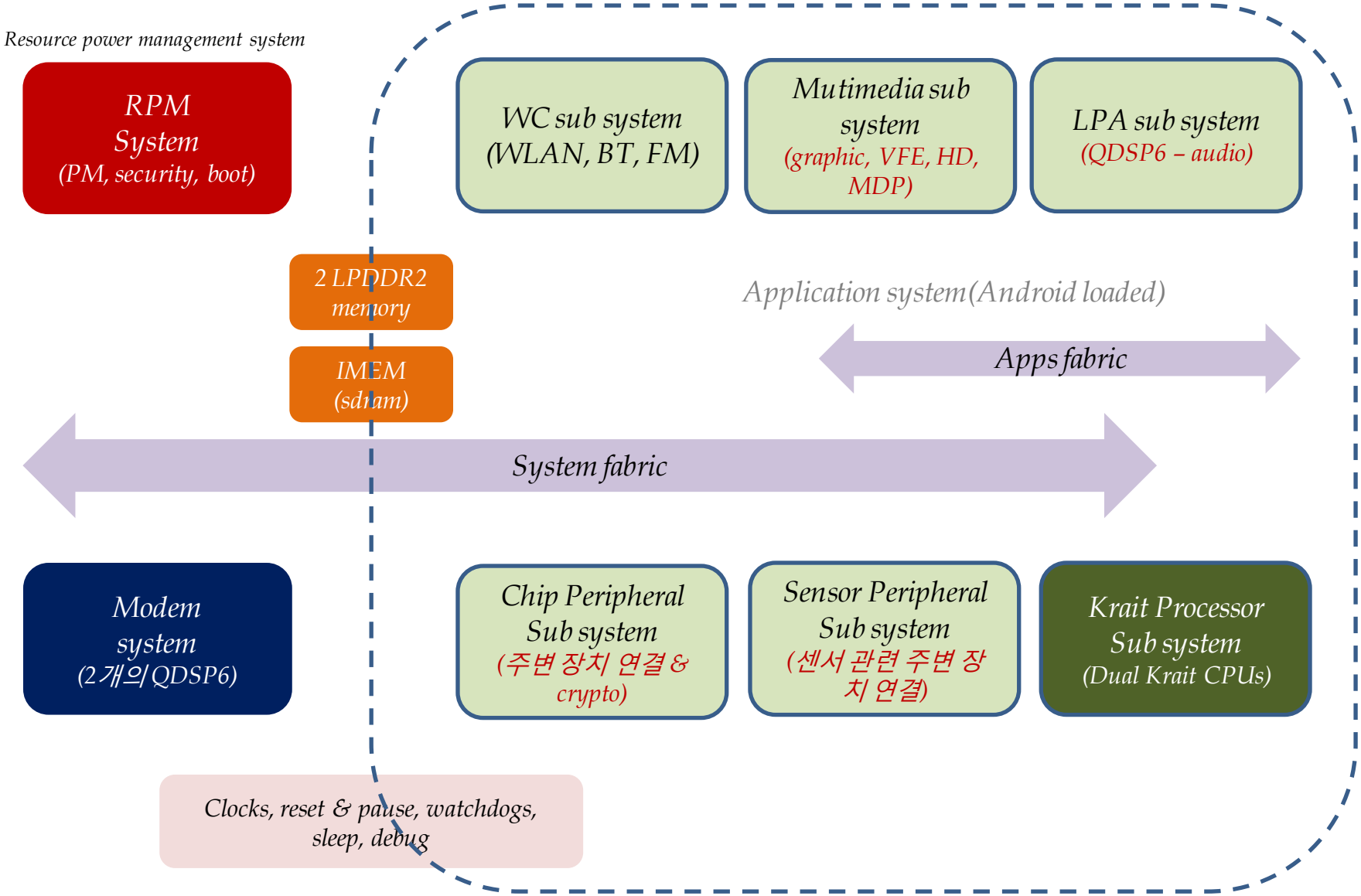
```
# export PATH=$(YOUR_HOME)/mydroid_jb/prebuilt/linux-x86/toolchain/arm-eabi-4.4.3/bin:$PATH
# make ARCH=arm CROSS_COMPILE=arm-eabi- msm8960-perf_defconfig
# make ARCH=arm CROSS_COMPILE=arm-eabi- menuconfig
# make ARCH=arm CROSS_COMPILE=arm-eabi- zImage
# make ARCH=arm CROSS_COMPILE=arm-eabi- modules
```

For more information, See the following site.

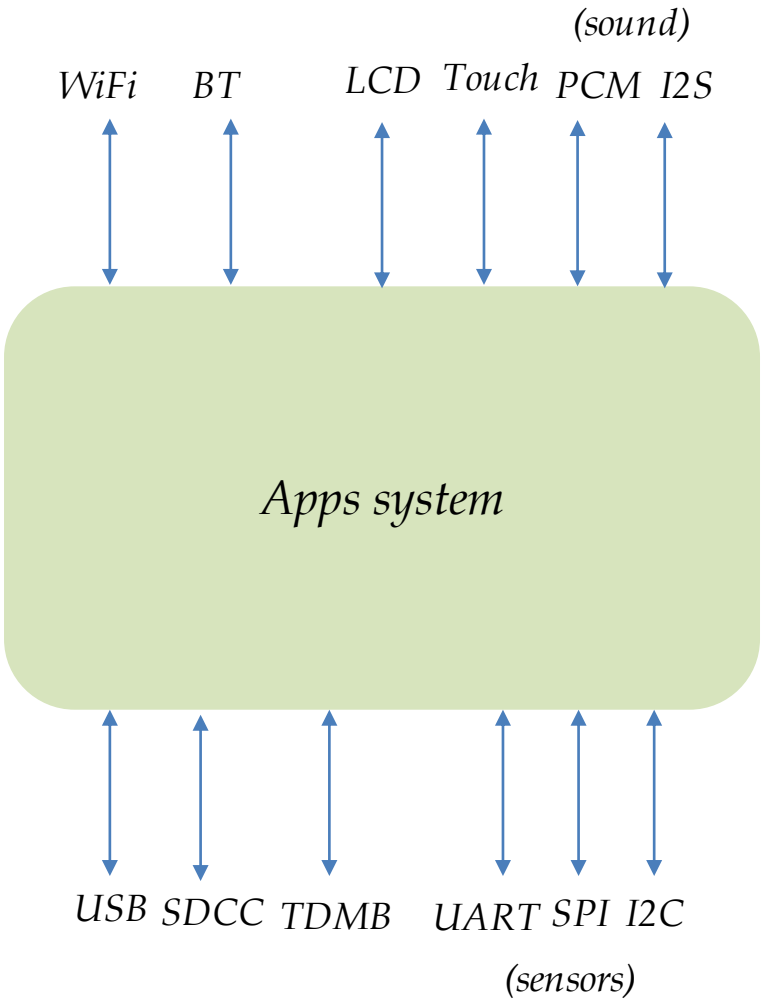
➔ *https://www.codeaurora.org/gitweb/quic/la/?p=kernel/msm.git;a=heads*

Case1: MSM8960

2. MSM8960 Chipset Overview(1)



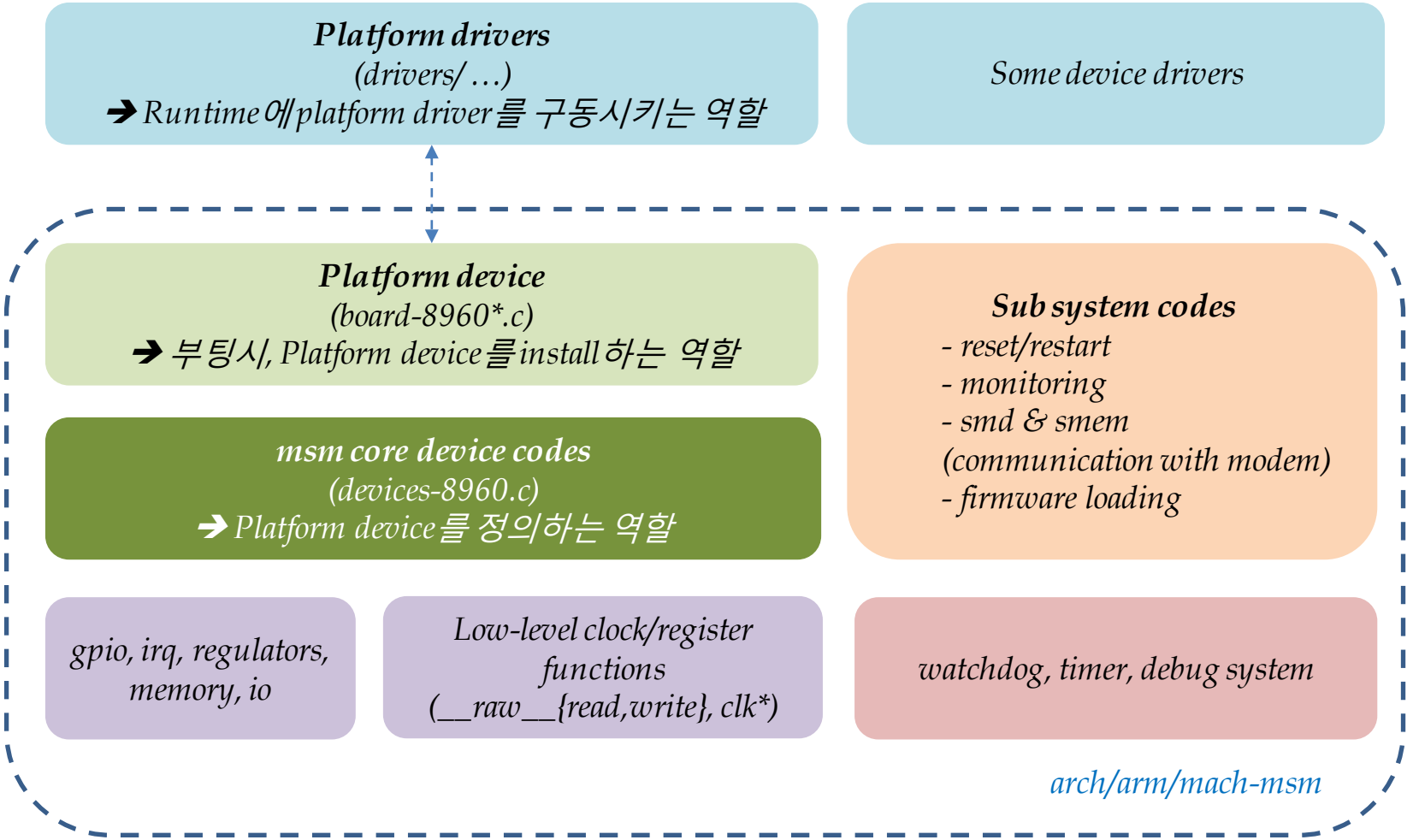
2. MSM8960 Chipset Overview(2)



(*) 옆의 그림에서는 실제 연결(통신) 방식과 기능을 편의상 서로 혼용해서 사용하였음.
➔ 즉, WiFi는 SDIO, BT는 UART를 사용하나…

(*) 옆의 그림을 그린 이유는, 실제로 kernel code에서 이를 기술해 주고 있기 때문임.

3. MSM8960 Machine Code Review(1)



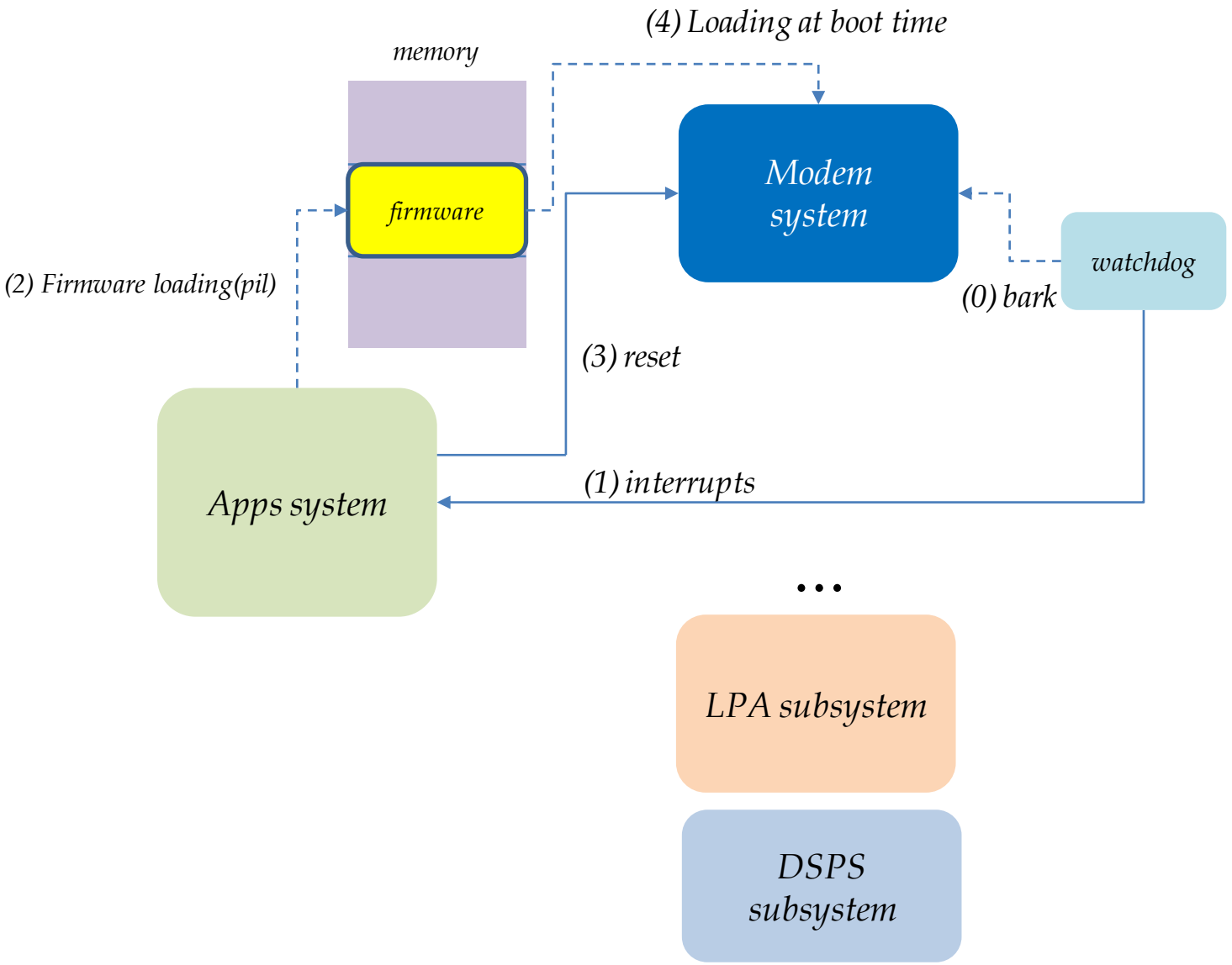
3. MSM8960 Machine Code Review(2)

- *1) CPU 기술(description) ➔ clock…*
- *2) bus 기술 ➔ system fabric, apps fabric, clocks …*
- *3) memory 기술 & 사용 관련 정의 ➔ pmem, ion mem 등 할당 …*
- *4) CPU와 주변 sub system과의 관계 기술 ➔ interrupt, reset, communication*
- *5) 주변 장치 기술/초기화(i2c, spi, uart, uim, sdcc, usb, …., gpio, irq, regulators) ➔ GSBI & GPIOMux*
- *6) key internal functions 기술 ➔ watchdog, sleep…*
- *7) linux driver model 적용(platform device & driver)*

3. MSM8960 Machine Code Review(3) – ***MACHINE_START macro***

```
MACHINE_START(MSM8960_CDP, "QCT MSM8960 CDP")
    .map_io = msm8960_map_io,
    .reserve = msm8960_reserve,
    .init_irq = msm8960_init_irq,
    .handle_irq = gic_handle_irq,
    .timer = &msm_timer,
    .init_machine = msm8960_cdp_init,
    .init_early = msm8960_allocate_memory_regions,
    .init_very_early = msm8960_early_memory,
    .restart = msm_restart,
MACHINE_END
```

3. MSM8960 Machine Code Review(4) – ***restart subsystem***



3. MSM8960 Machine Code Review(5) – *gpiomux/1*

<gpio operation 정의>

- *drivers/gpio/gpio-msm-common.c*
  - ➔ *msm_gpio_probe( ) 함수에서 gpiochip_add(&msm_gpio.gpio_chip)을 호출하여 msm_gpio.gpio_chip을 등록했으므로, 이후 kernel에서 사용하는 gpio 관련 함수 즉, request, free, set, get, direction_input, direction_output 등은 모두 msm_gpio에서 새로 정의한 함수를 사용하게 될 것임.*
- *drivers/gpio/gpio-msm-v2.c*
- *drivers/gpio/gpiolib.c*
- *drivers/gpio/pm8xxx-gpio.c*

<gpio pin 배치 정의>

- *arch/arm/mach-msm/gpiomux.c*
- *arch/arm/mach-msm/gpiomux-v2.c*
- *arch/arm/mach-msm/board-8960-gpiomux.c*
  - ➔ *여기에서 gpiomux pin 배치 코드 나열함.*

=> *See Documentation/arm/msm/gpiomux.txt*

3. MSM8960 Machine Code Review(6) – *gpiomux/2*

<gpiomux setting example>

```
struct msm_gpiomux_config  gpio123_config __initdata = {
   .gpio = 123,
   .settings = {
      [GPIOMUX_ACTIVE] = {
         .func = GPIOMUX_FUNC_GPIO,
         .drv  = GPIOMUX_DRV_2MA,
         .pull = GPIOMUX_PULL_NONE,
         .dir  = GPIOMUX_OUT_HIGH,
      },
      [GPIOMUX_SUSPENDED] = {
         .func = GPIOMUX_FUNC_3,
         .drv  = GPIOMUX_DRV_8MA,
         .pull = GPIOMUX_PULL_DOWN,
      },
   },
};
```

<gpiomux setting 의미>

- *When the system boots, gpio 123 will be put into the SUSPENDED setting.*
- *When the reference count for gpio 123 rises above 0, the ACTIVE setting will be applied.*
- *When the reference count falls back to 0, the SUSPENDED setting will be reapplied.*

➔ *이렇게 하는(runtime gpio configuration) 이유는 전류 소모를 최대한 줄이기 위함이다 !*

<초기화 - msm8960_init_gpiomux( ) 에서 호출함>

- *msm_gpiomux_init() -> memory 할당*
- *msm_gpiomux_install() -> 해당 gpio pin 주소에 값(drv, func, pull 등의 참조)을 쓰고 있음(초기화 정도)*

3. MSM8960 Machine Code Review(7) – *GSBI 주변 장치 추가 절차*

1) Set the clocks

➔ *arch/arm/mach-msm/clock-8960.c*

2) Create a platform device

➔ *arch/arm/mach-msm/devices.h*

➔ *arch/arm/mach-msm/devices-8960.c*

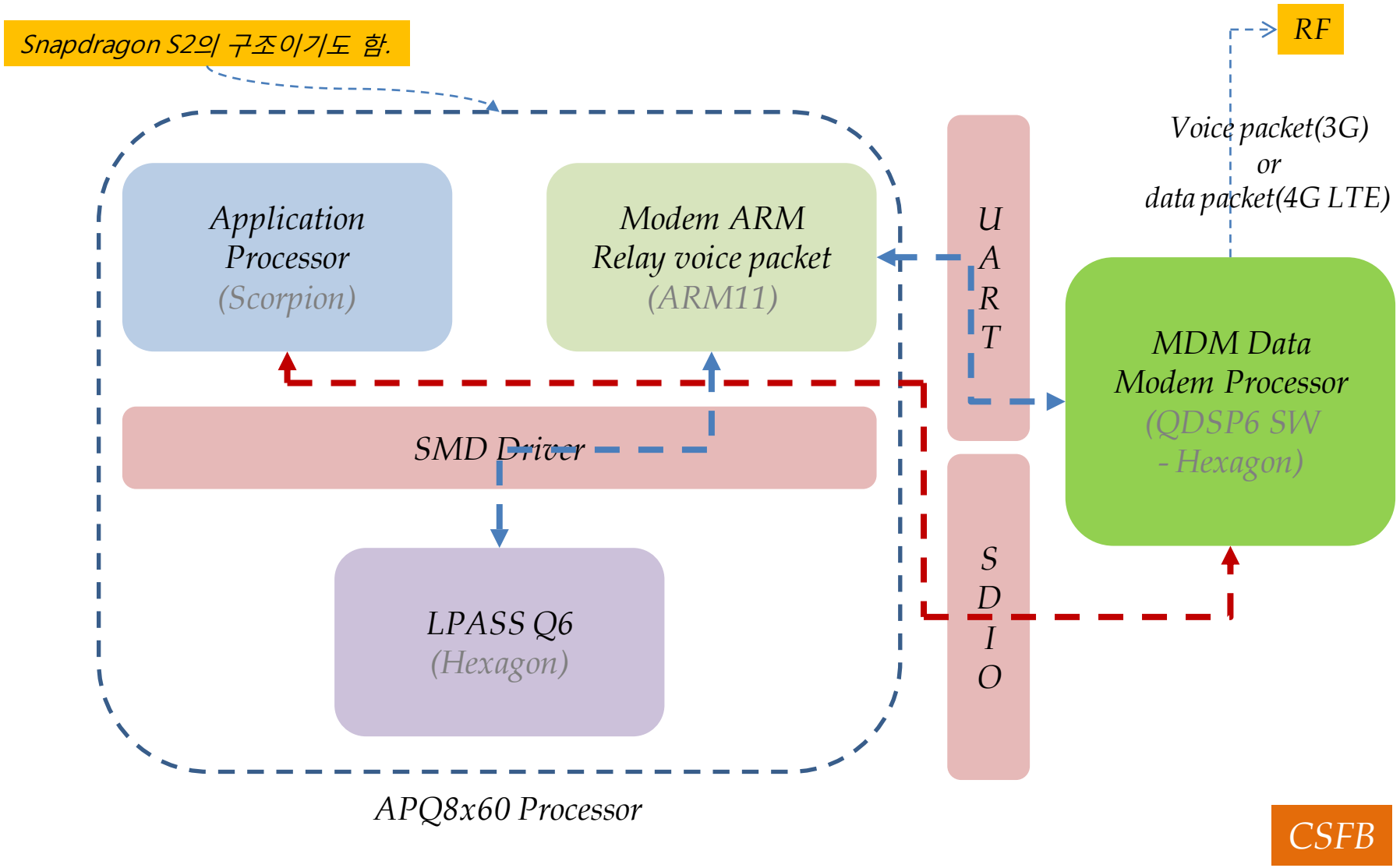
➔ *arch/arm/mach-msm/board-8960.c*

3) Configure GPIO

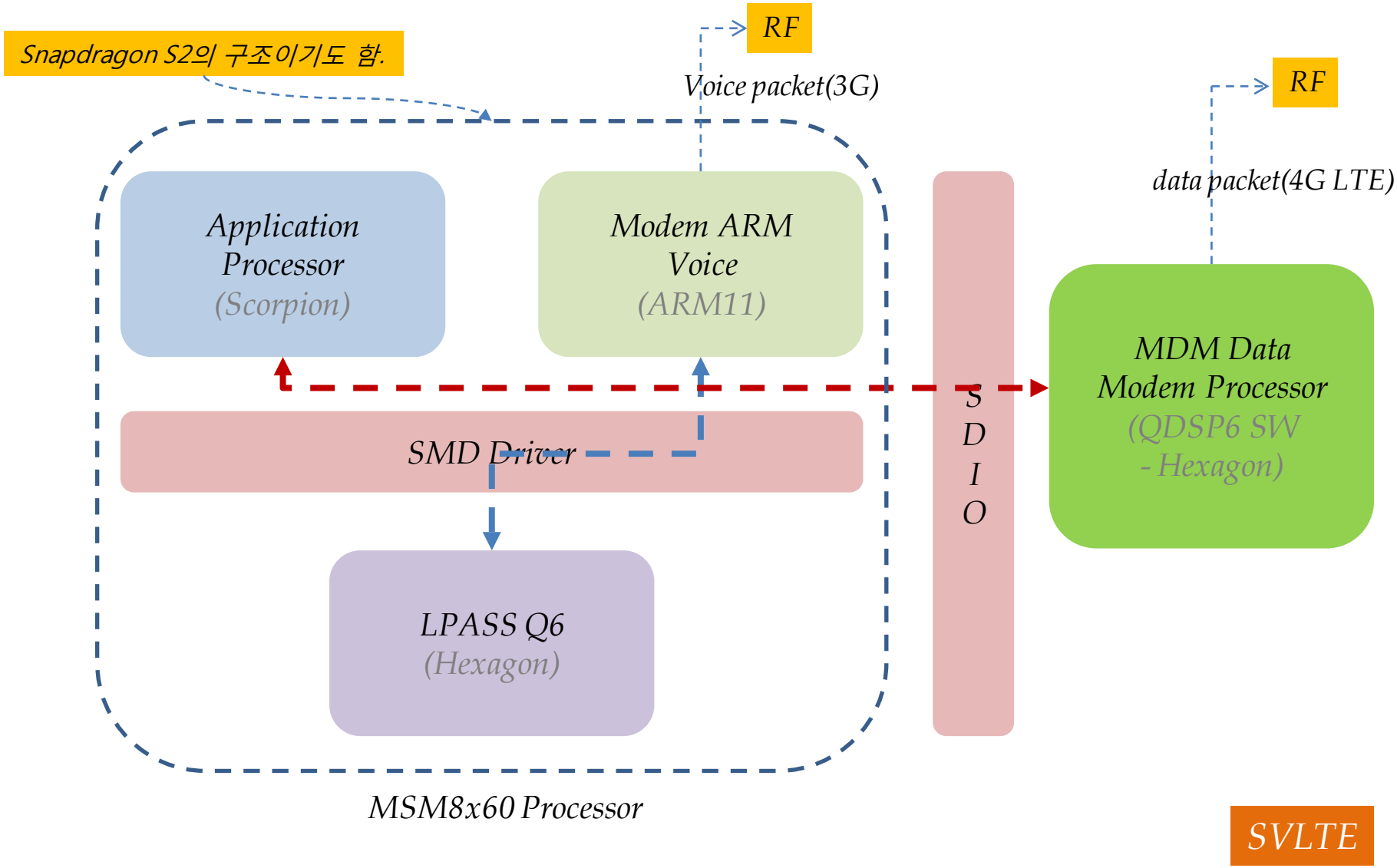
➔ *arch/arm/mach-msm/board-8960-gpiomux.c*

Case2: APQ/MSM8x60 + MDM9x00 fusion chipset

4. Fusion Chipset Overview(1)

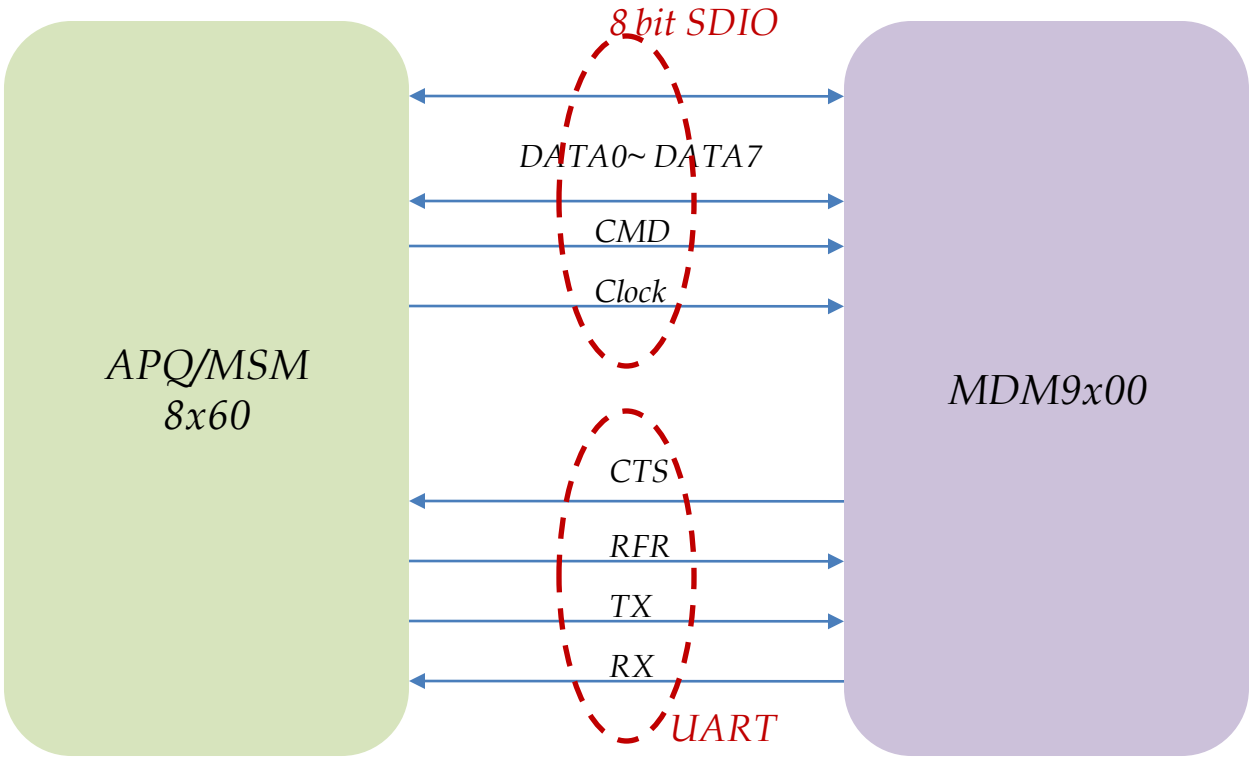


4. Fusion Chipset Overview(2)

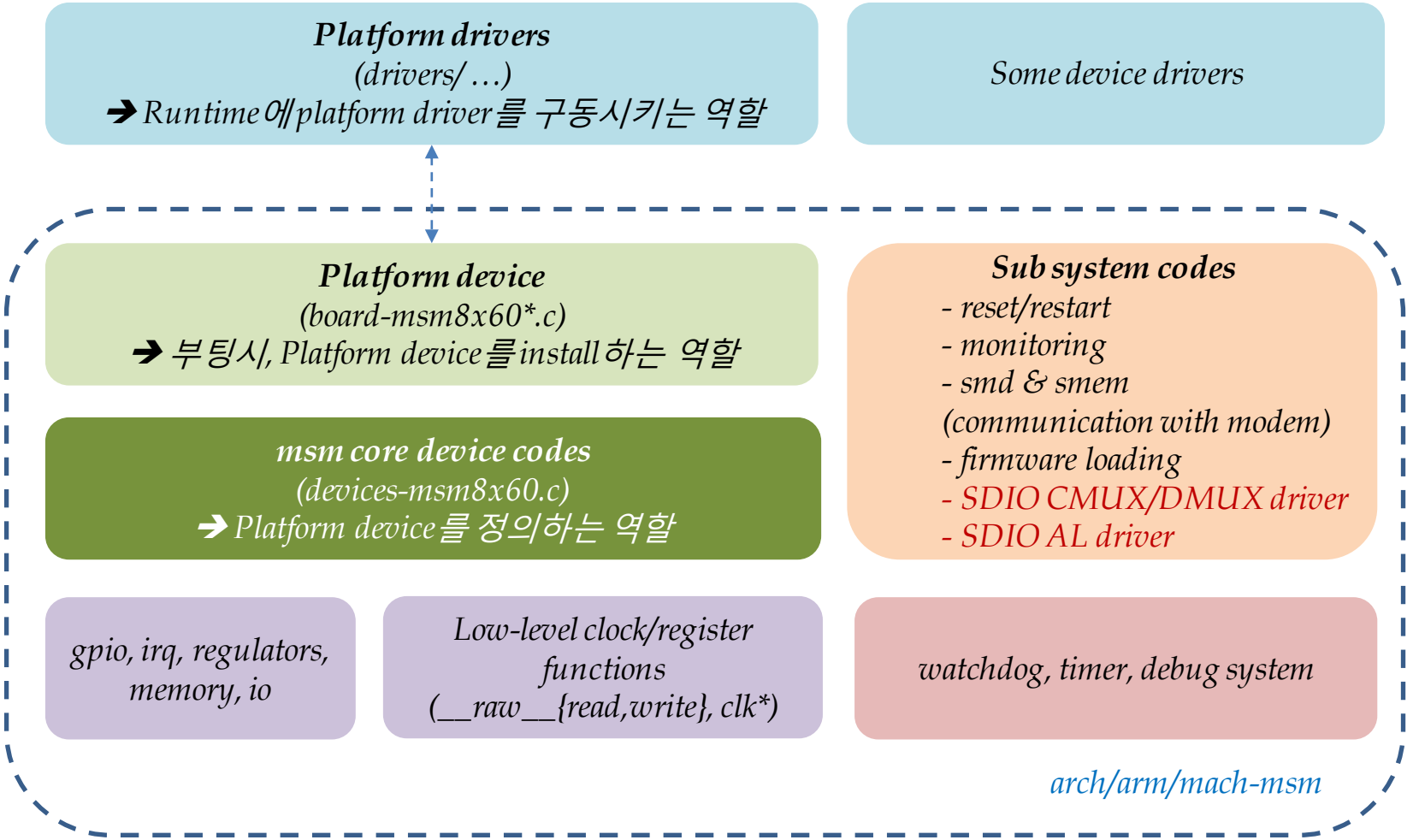


4. Fusion Chipset Overview(3)

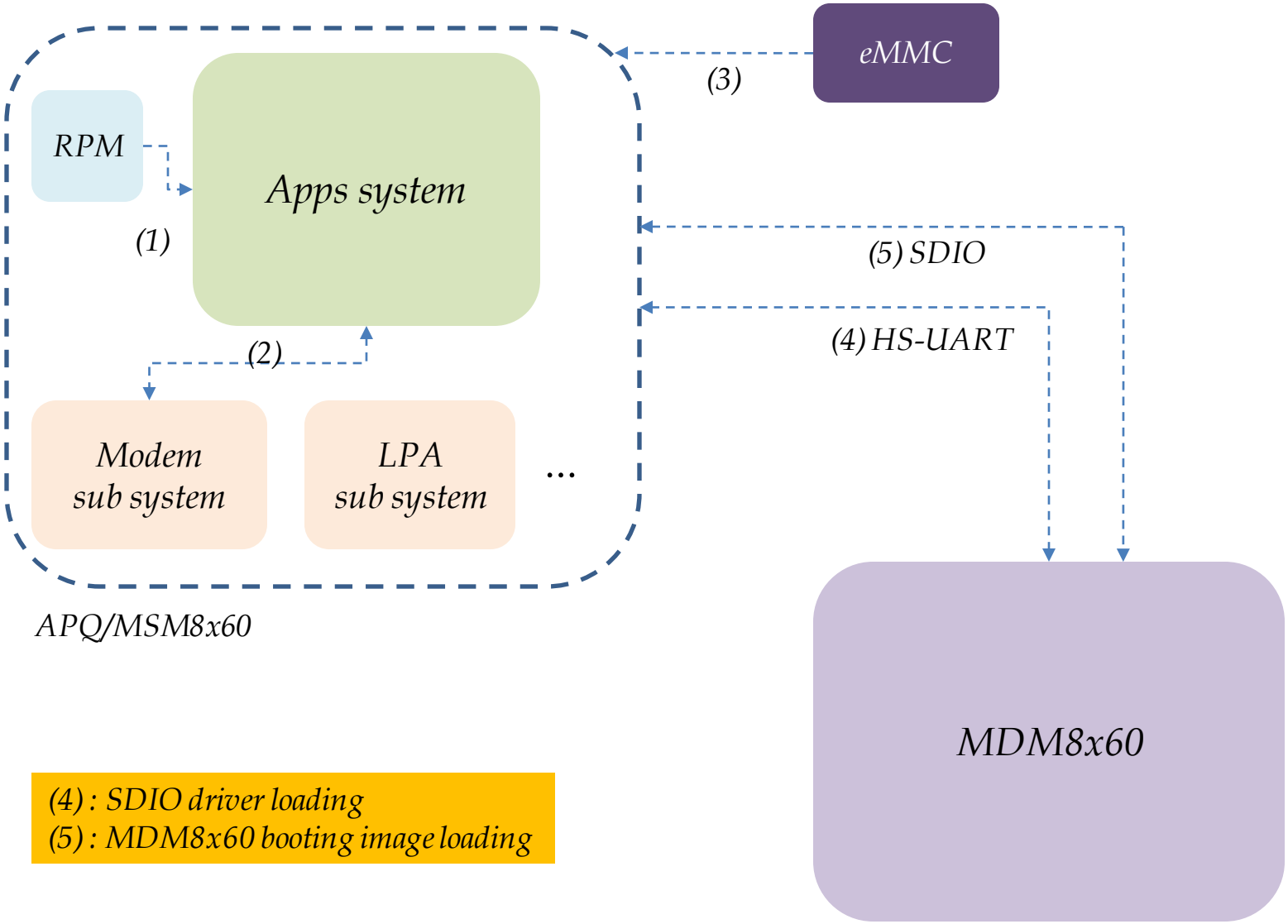
- ***SDIO***: *QMI, RmNet, RPC, Diag*
  - ➔ *주로 LTE data 전송을 위해 사용됨(고속 data 전송).*
- ***HS-UART***: *QMI control, voice, boot*
  - ➔ *주로 voice data를 전송하기 위해 사용됨.*



5. APQ/MSM8x60 Machine Code Review

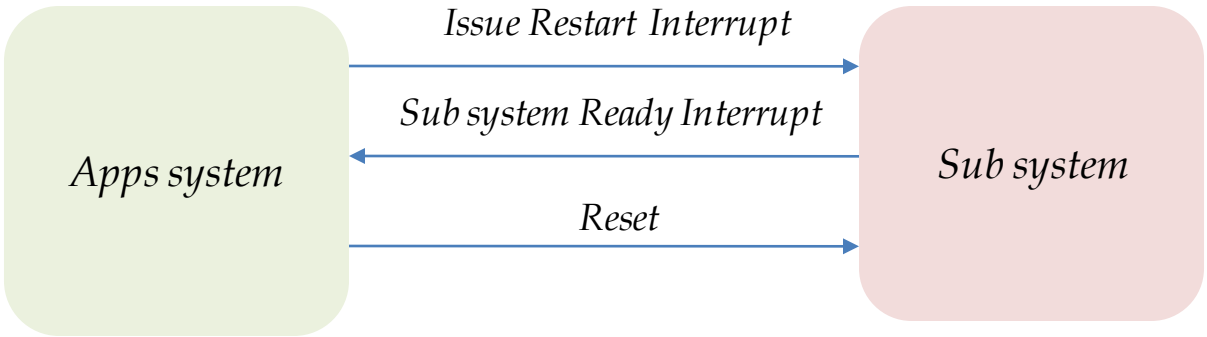


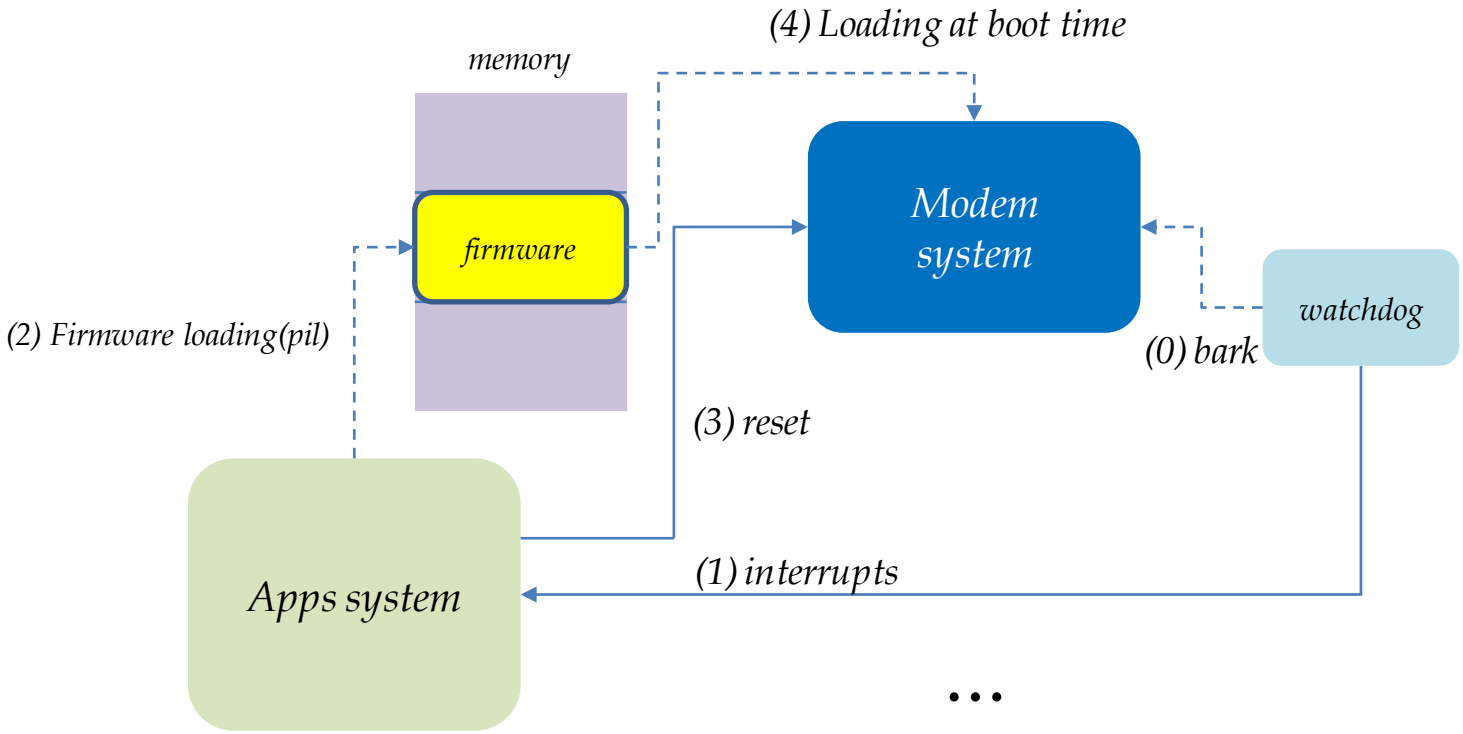
5. APQ/MSM8x60 Machine Code Review – ***Boot Flow***



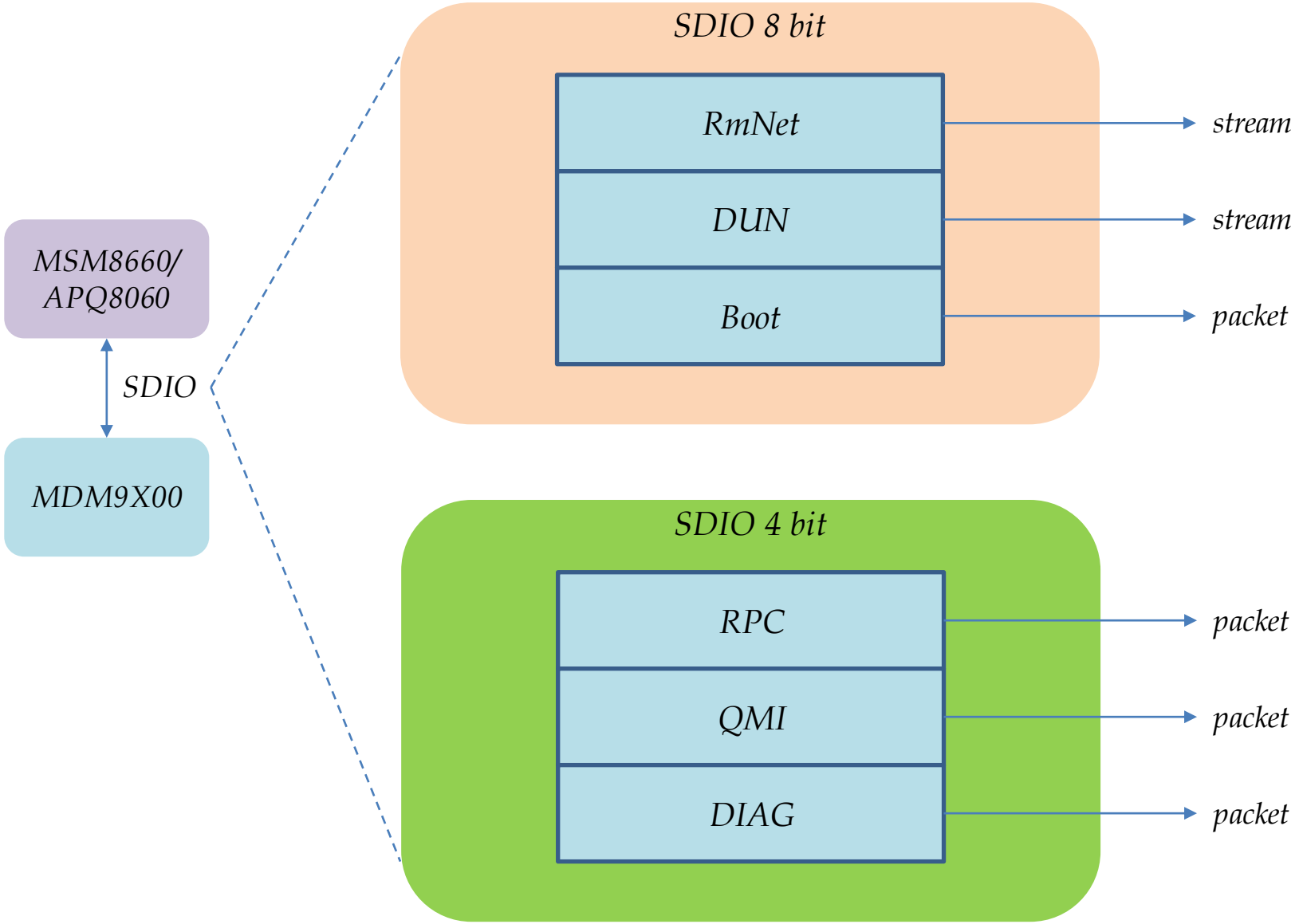
(4) : SDIO driver loading
(5) : MDM8x60 booting image loading

5. APQ/MSM8x60 Machine Code Review – ***Sub system Restart***

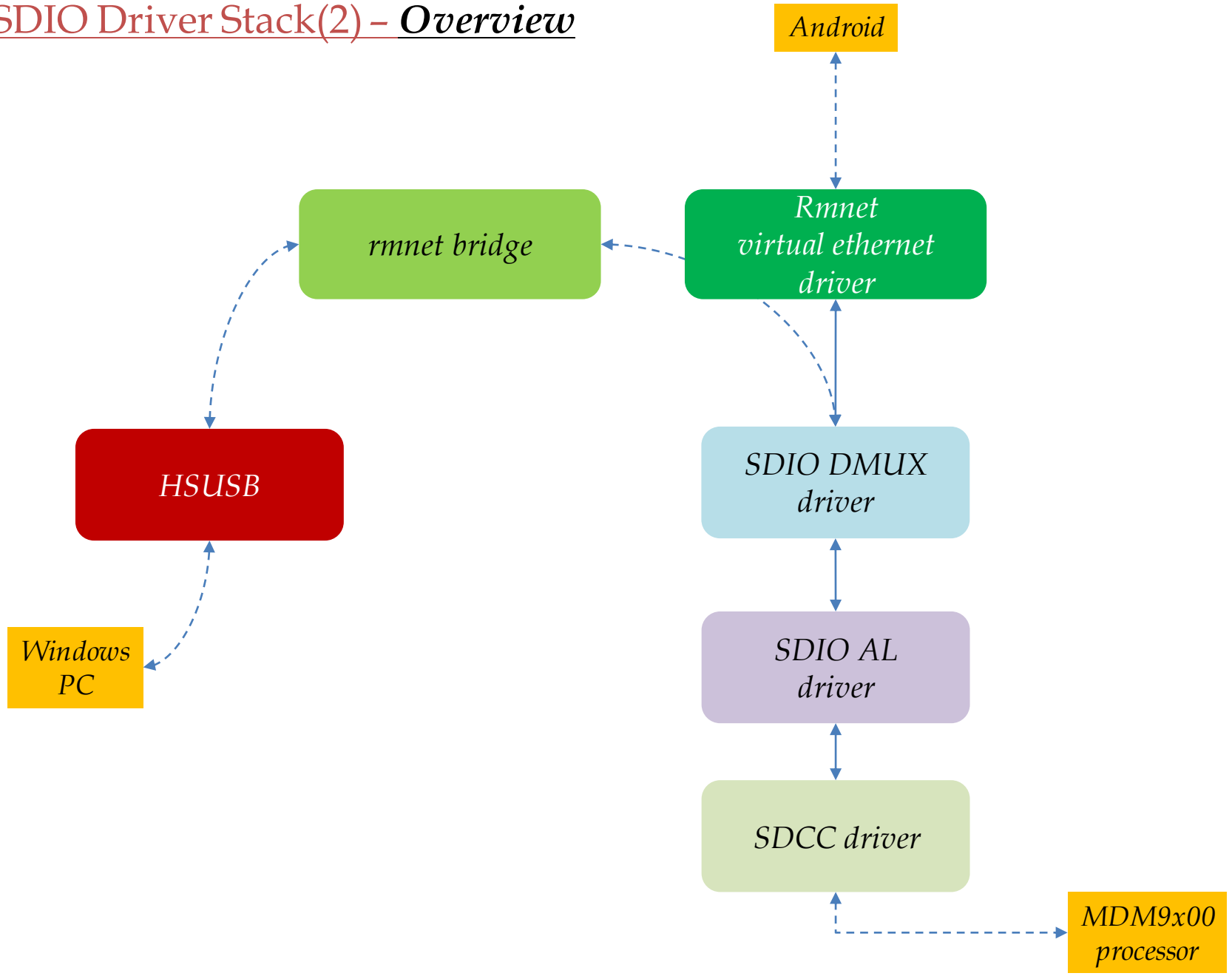




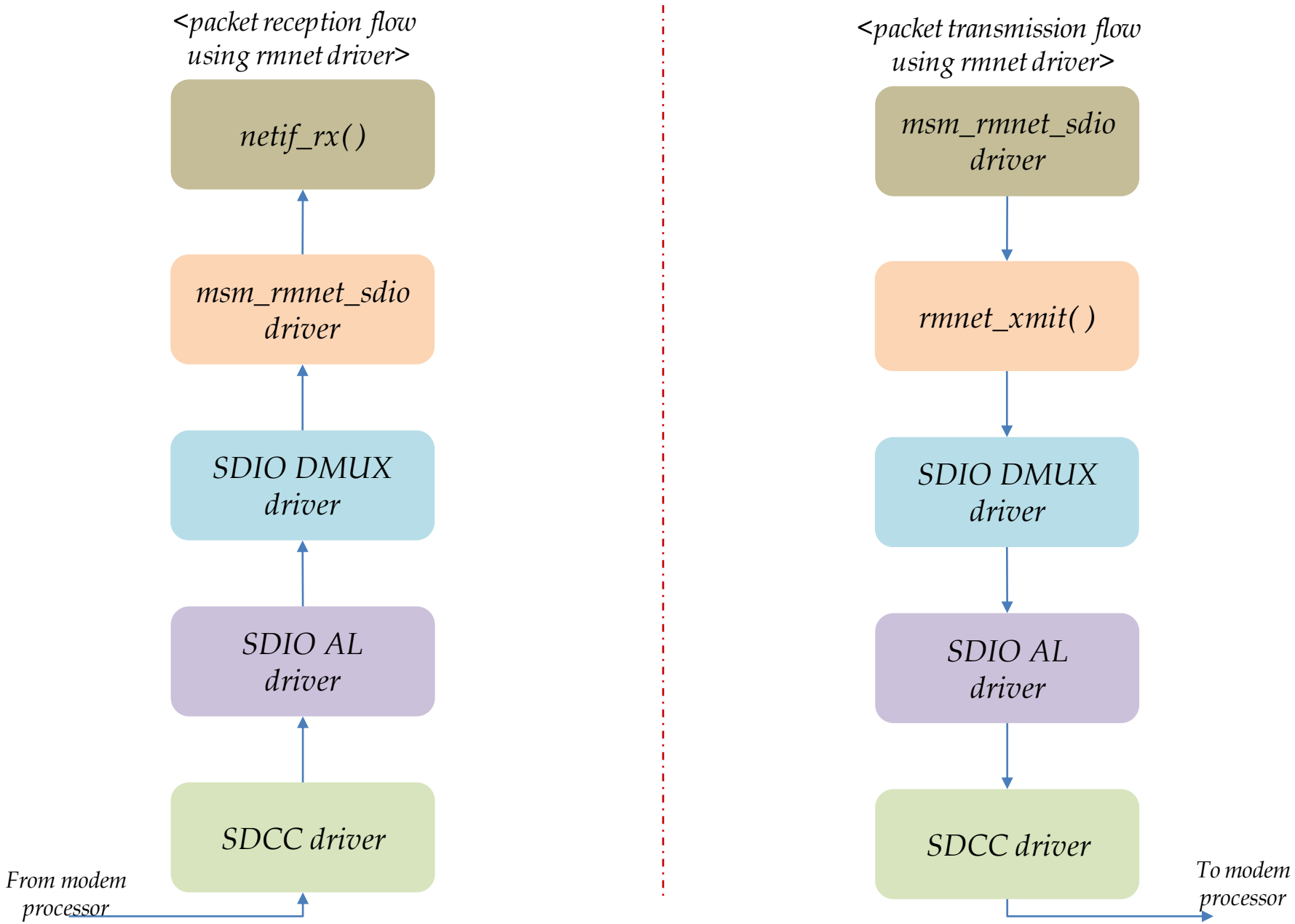
6. SDIO Driver Stack(1) – ***SDIO 8/4 pins***



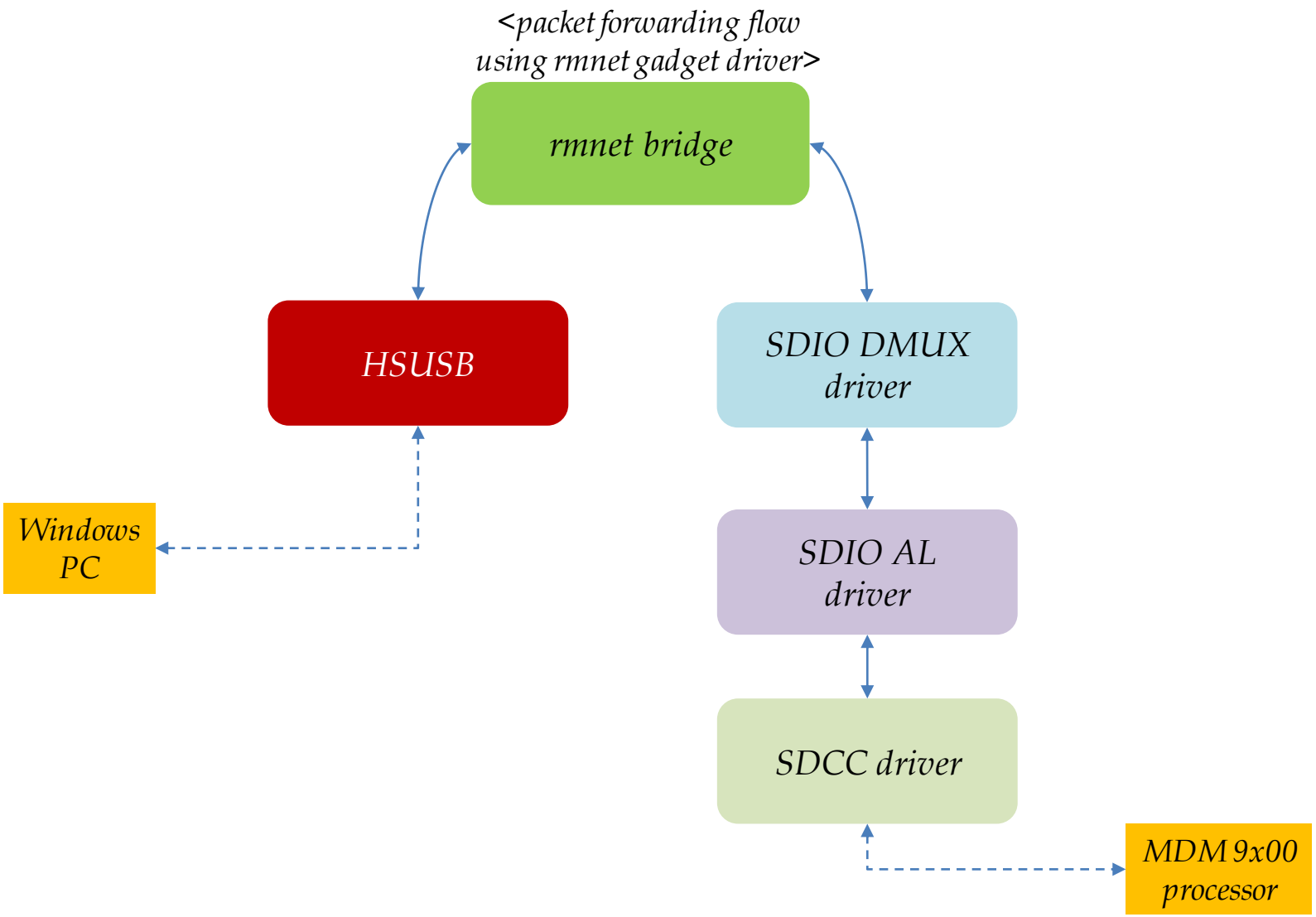
6. SDIO Driver Stack(2) – ***Overview***



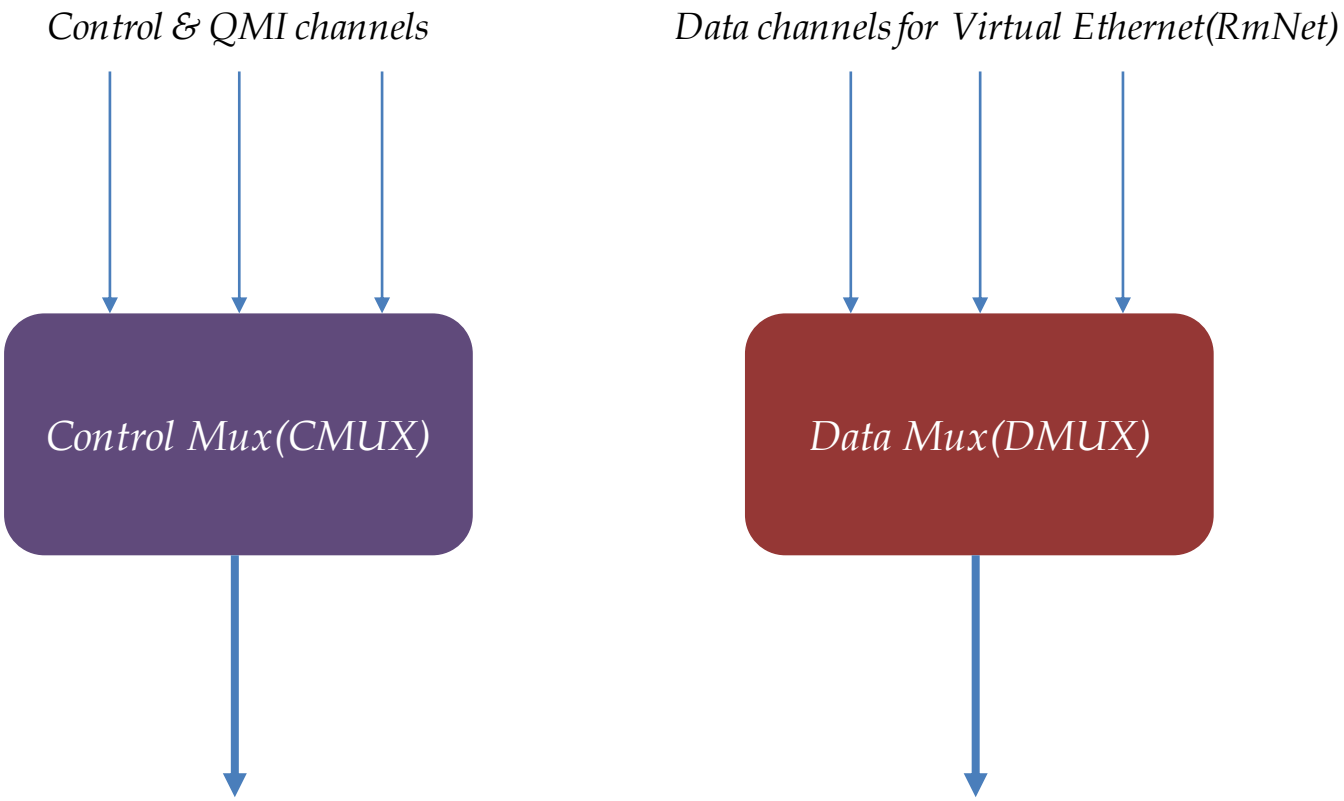
6. SDIO Driver Stack(3) – ***LTE data packet flow***



6. SDIO Driver Stack(4) - ***LTE data packet flow under Tethering Env.***



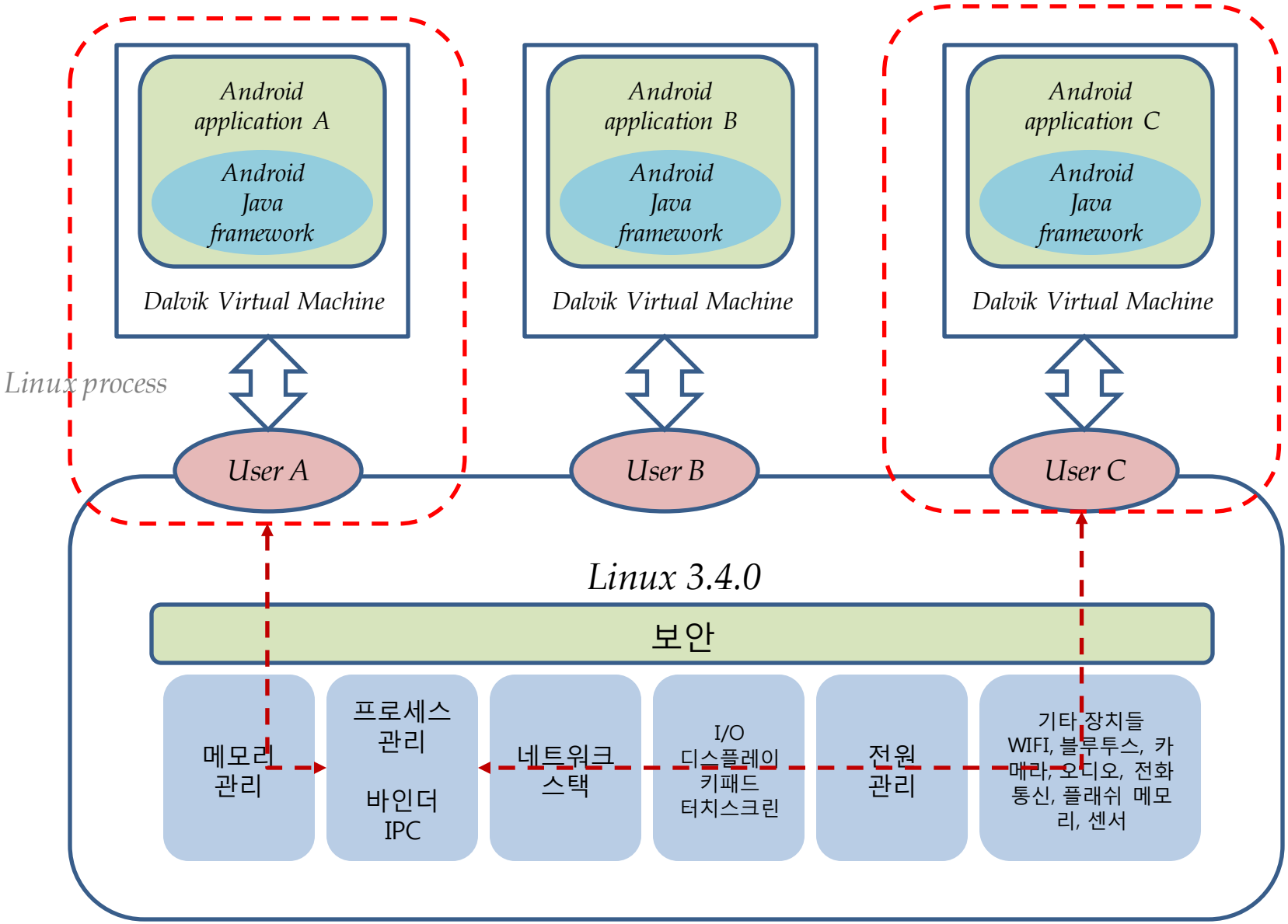
6. SDIO Driver Stack(5) – ***Mux Driver***



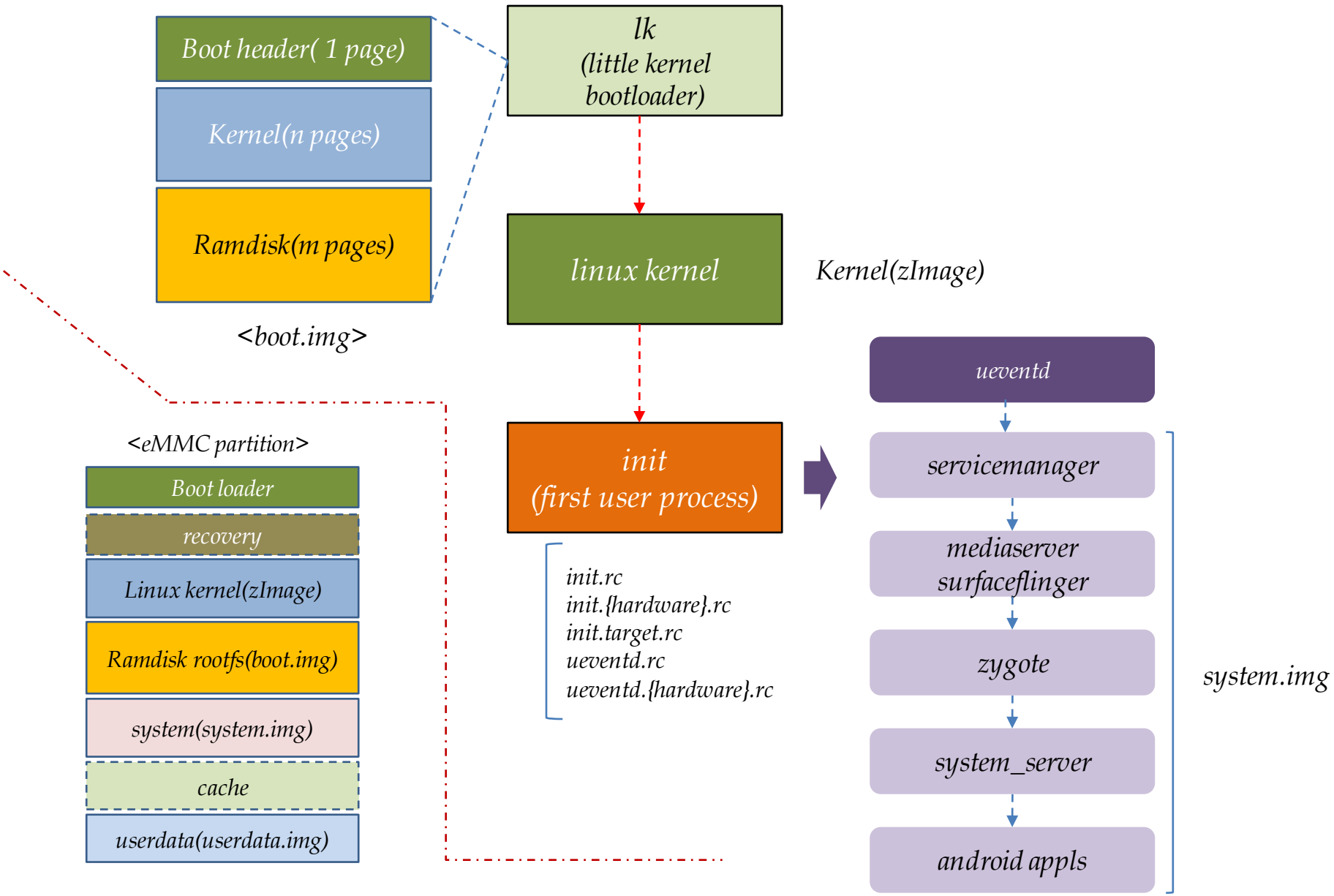
(*) 위의 그림은 MUX를 통해 control 및 data packet이 하나로 묶여 전달되는 것을 그림으로 표현한 것으로, 패킷 수신 시에는 반대의 과정을 거치게 된다.
➔ framing/deframing, multiplexing of multiple logical SDIO ports over a single SDIO function

Jelly Bean BSP Porting Guide

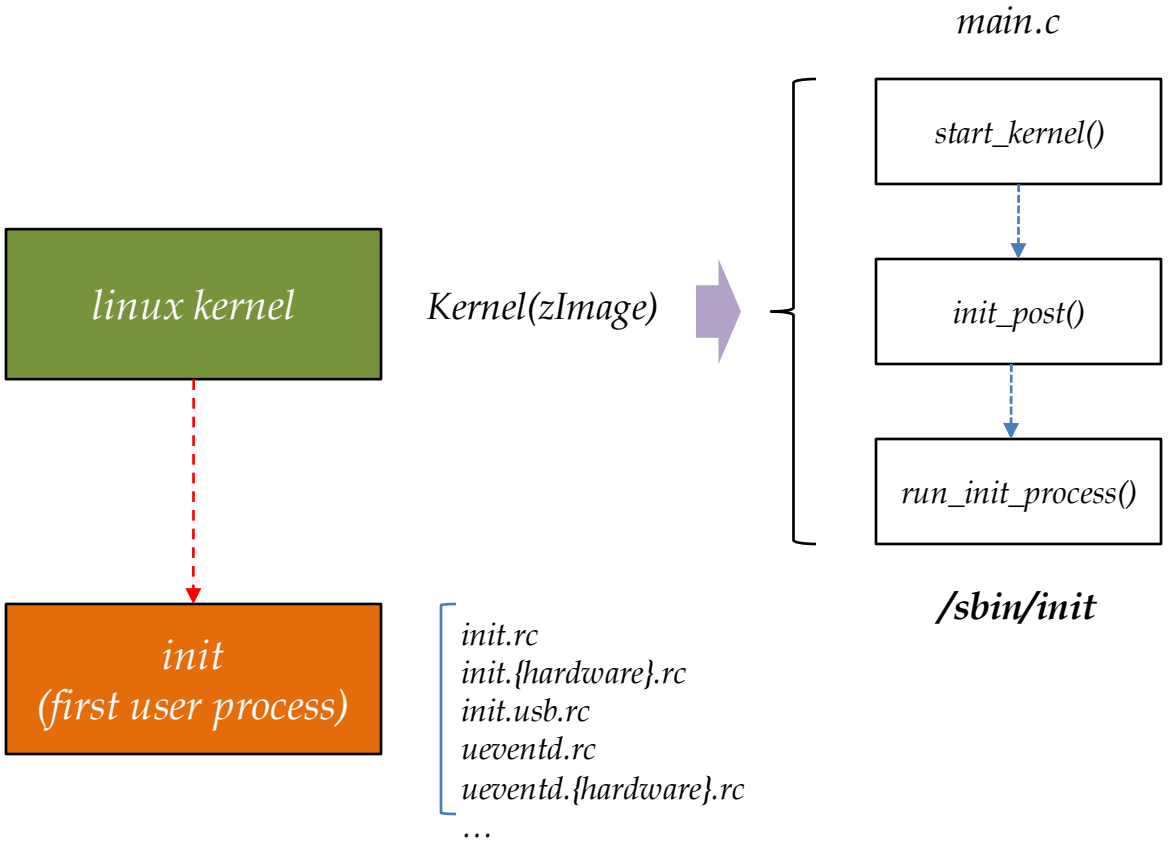
7. Jelly Bean Android Boot Flow(1)



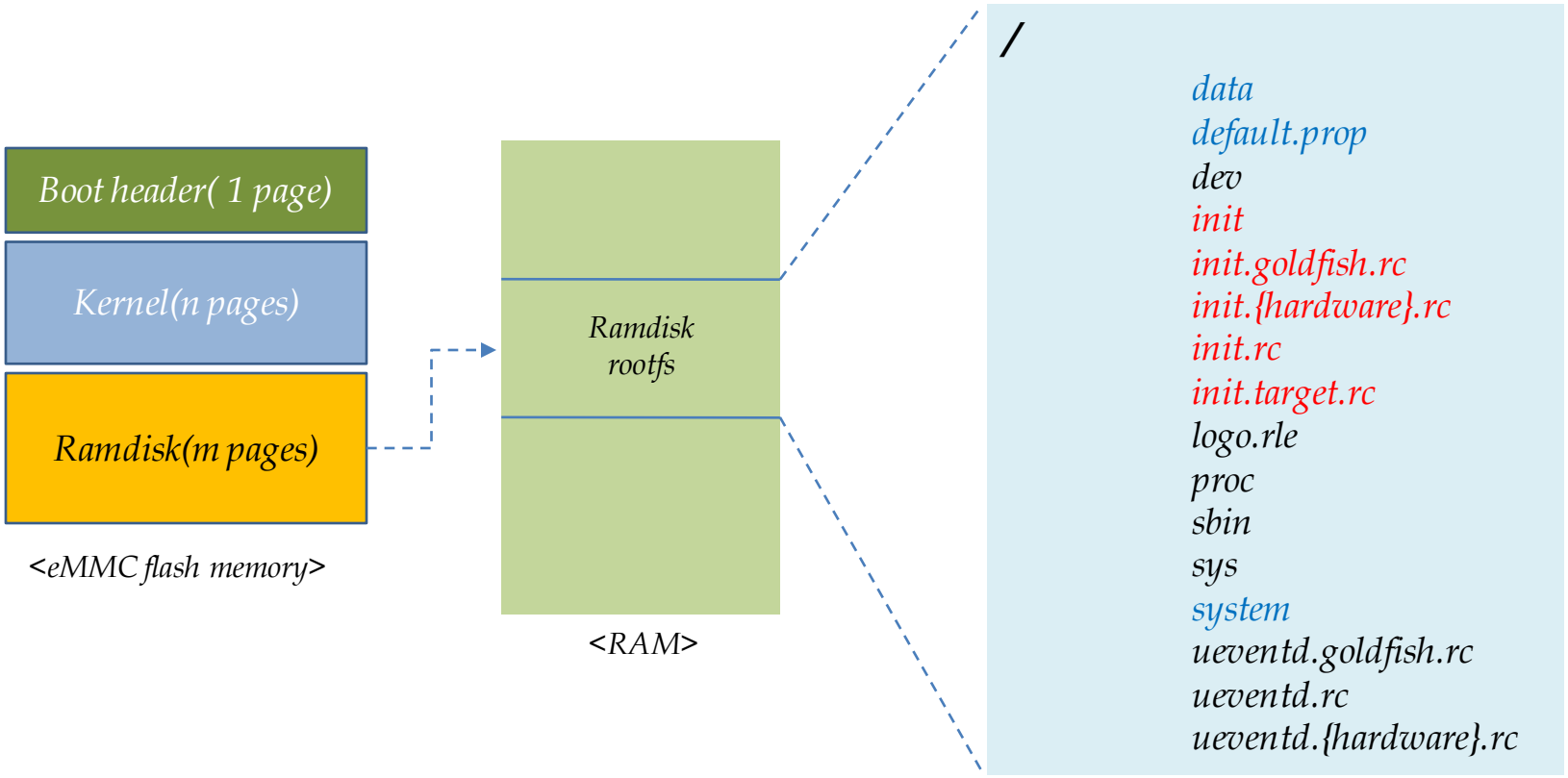
7. Jelly Bean Android Boot Flow(2)



7. Jelly Bean Android Boot Flow(3)



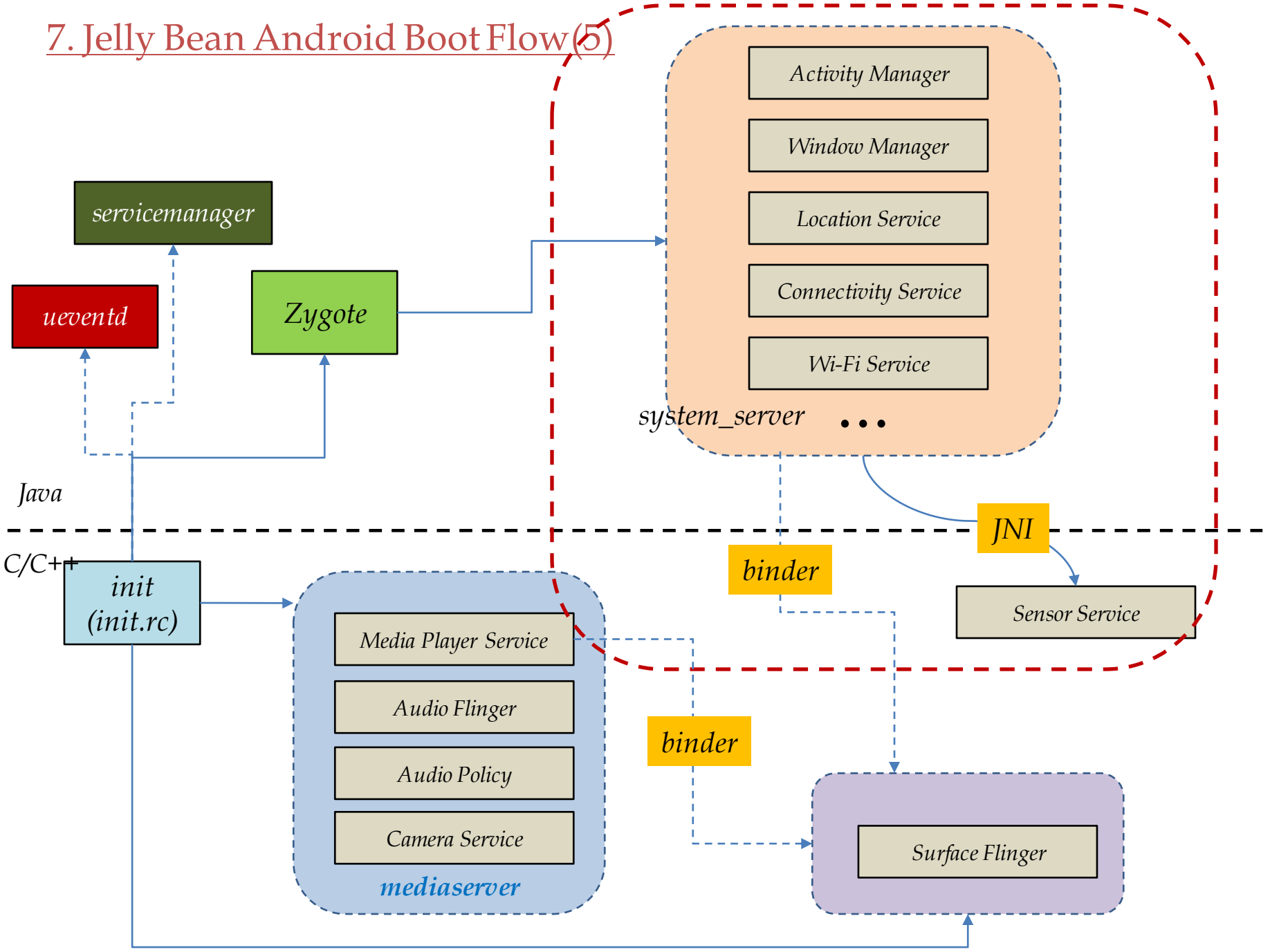
7. Jelly Bean Android Boot Flow(4)



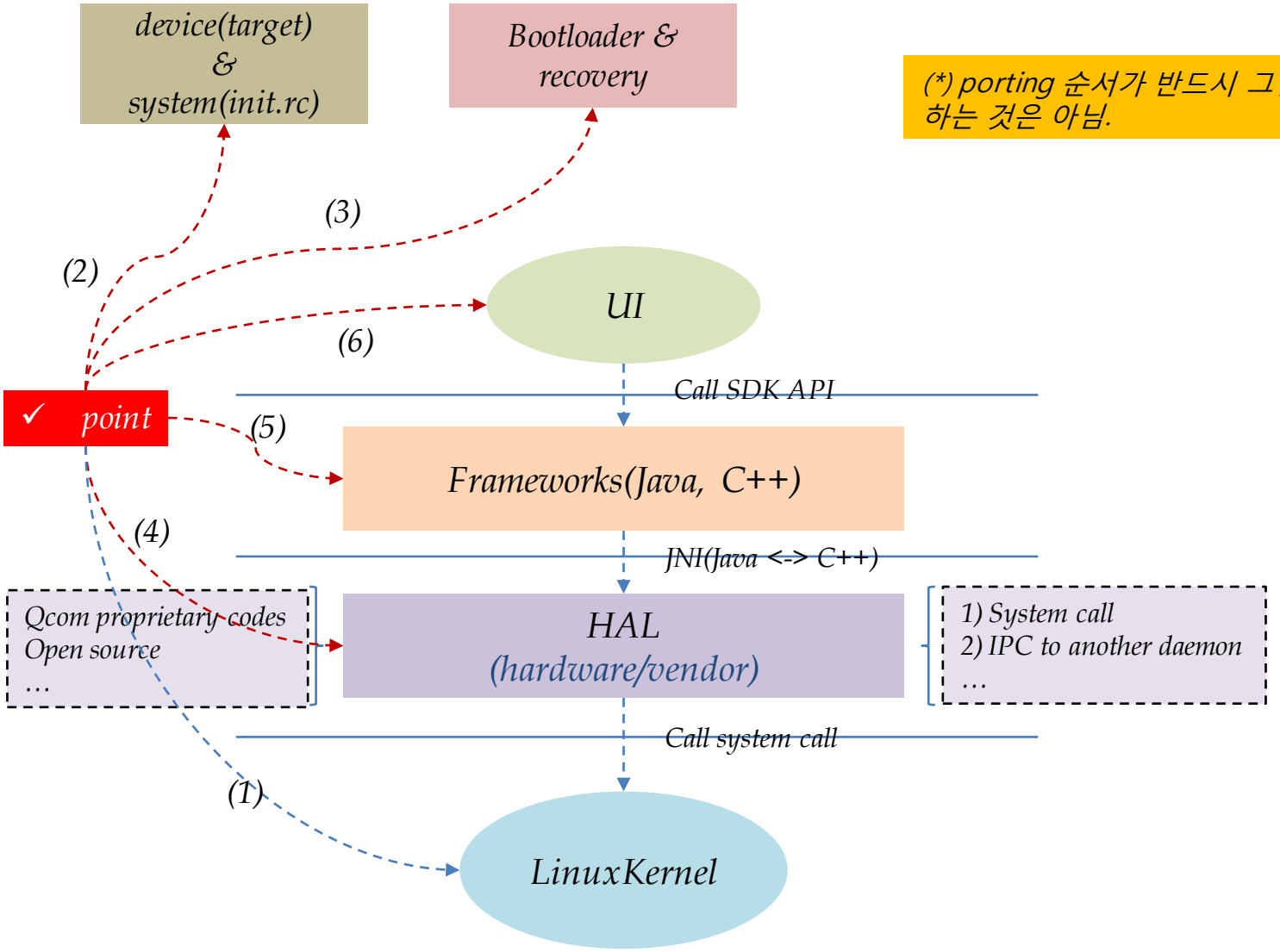

```
# mount –o remount,rw –t ext4 /dev/block/mtdblockX  /system
# mount –o remount,rw –t ext4 /dev/block/mtdblockX  /data
```

(*) rootfs 이외의 file system은 위와 같은 형태로 mount 가능

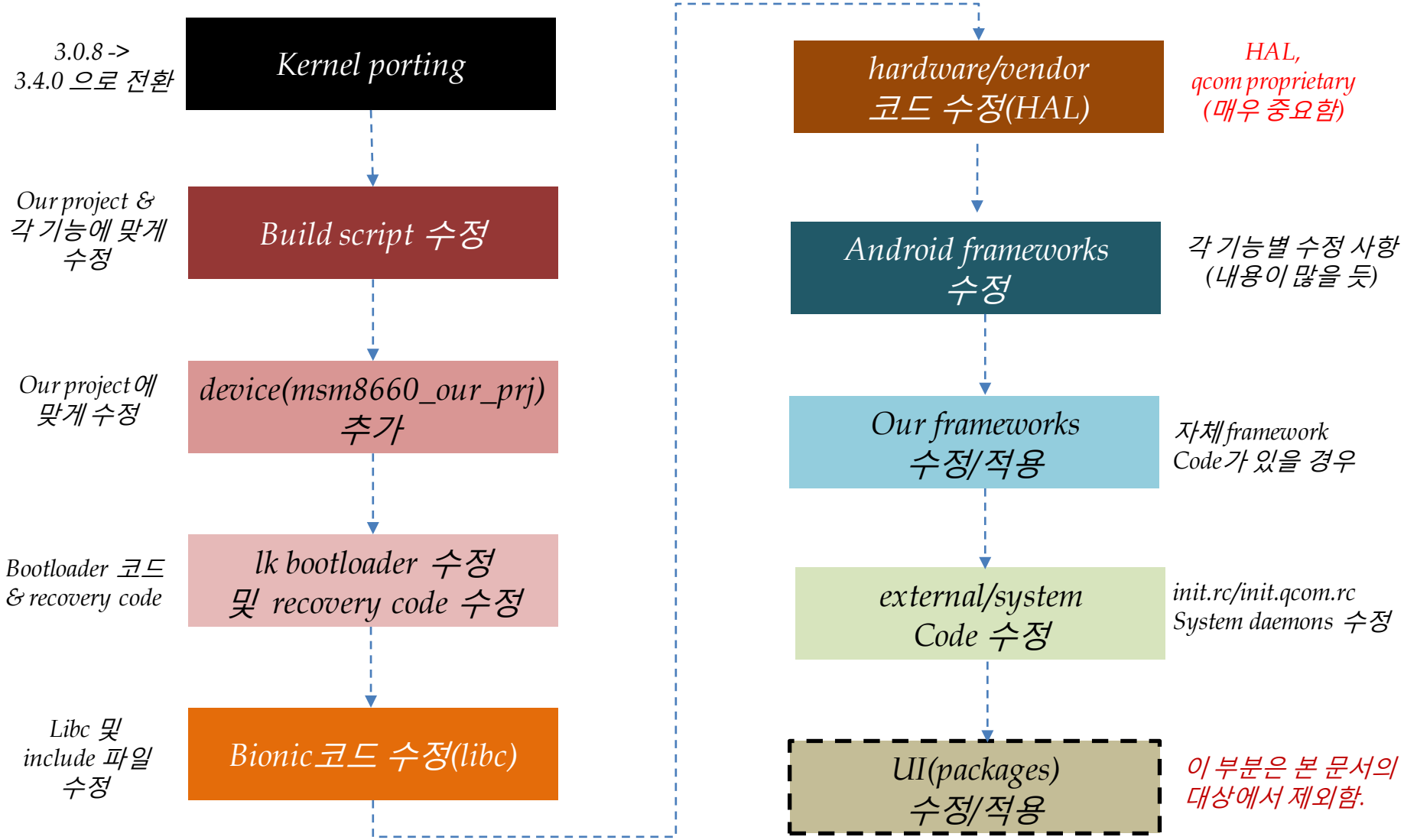
7. Jelly Bean Android Boot Flow(5)



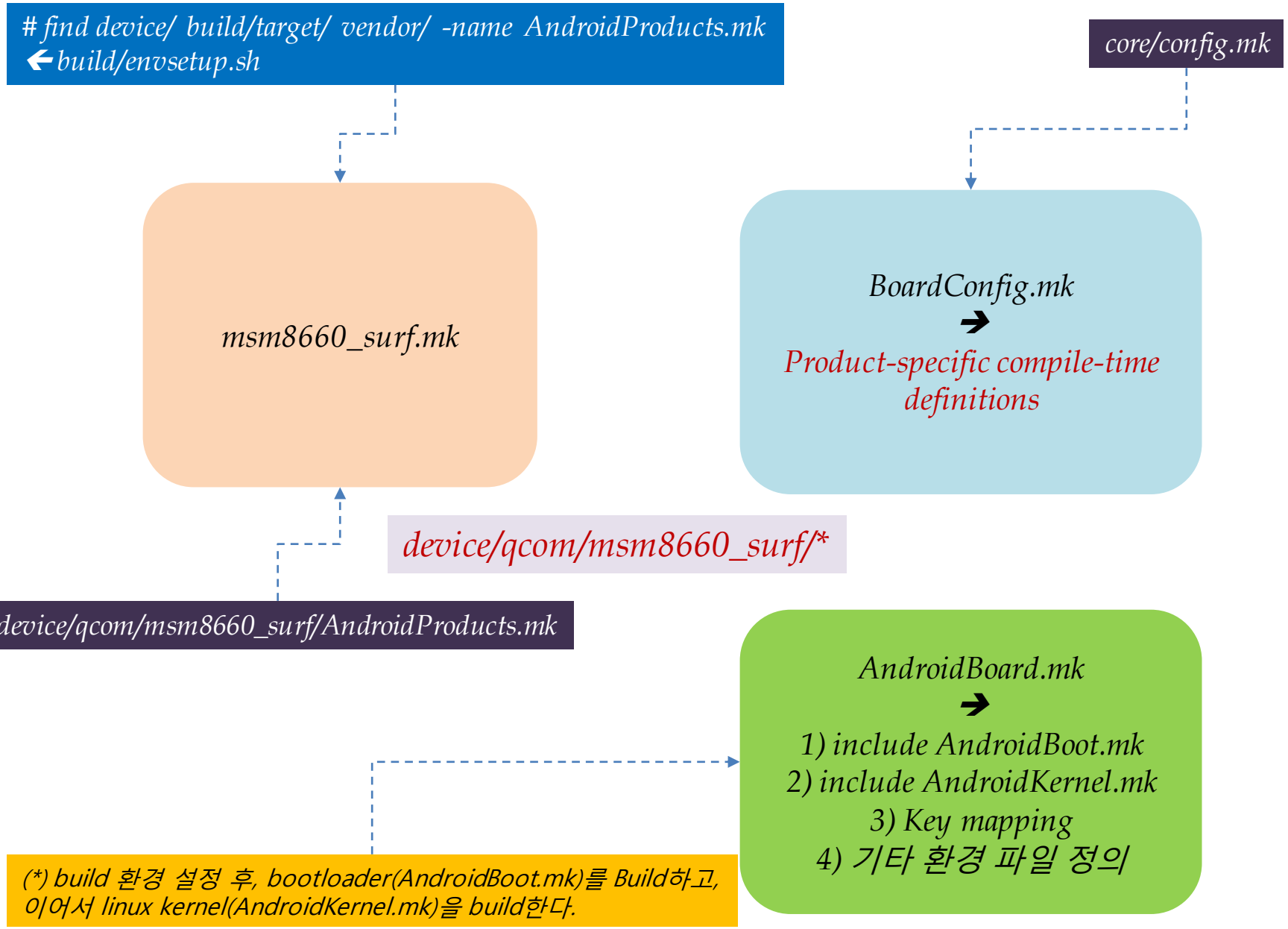
8. Jelly Bean BSP Porting Scope(1) – ***Android/1***



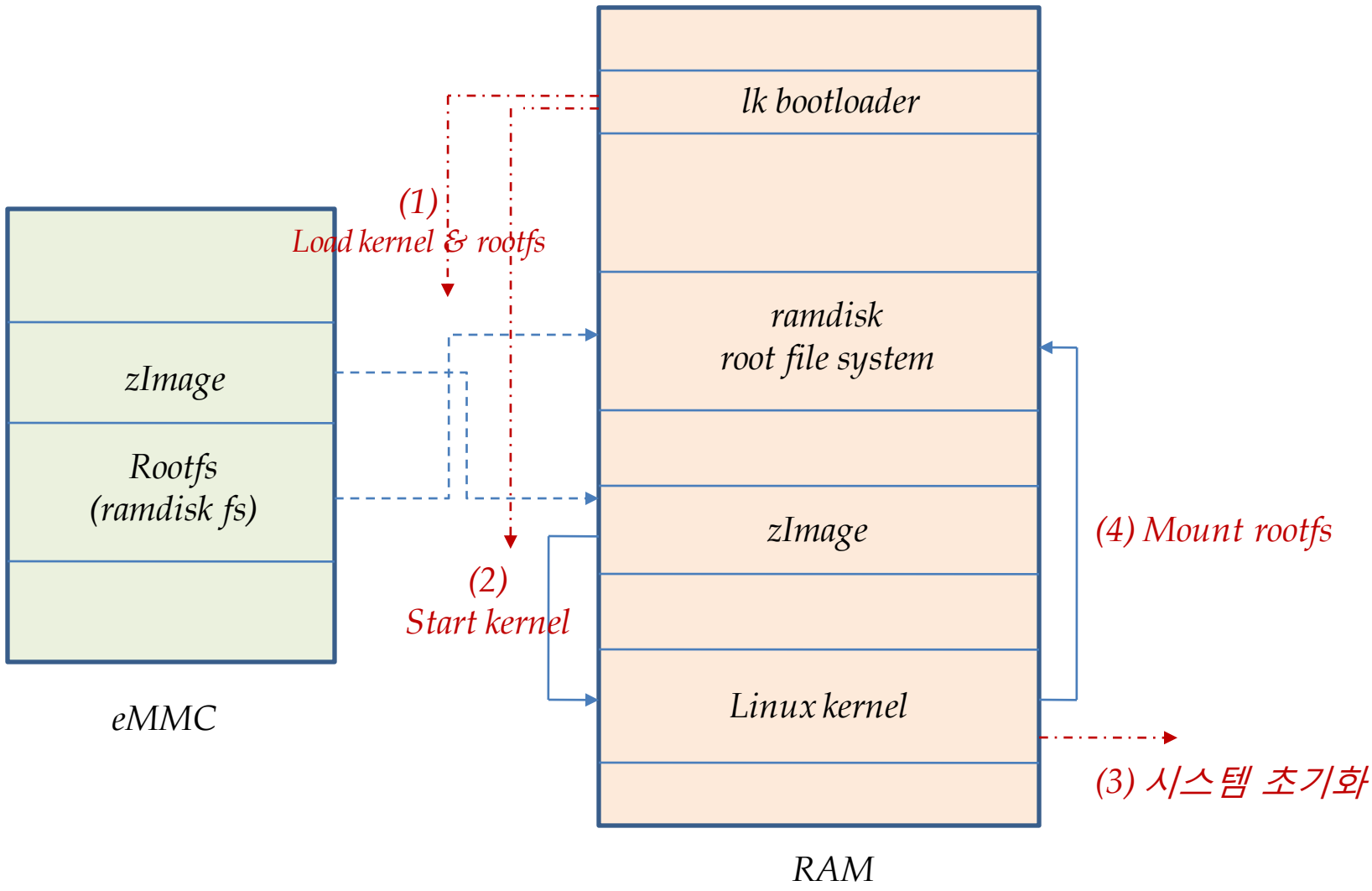
8. Jelly Bean BSP Porting Scope(1) – ***Android/2***



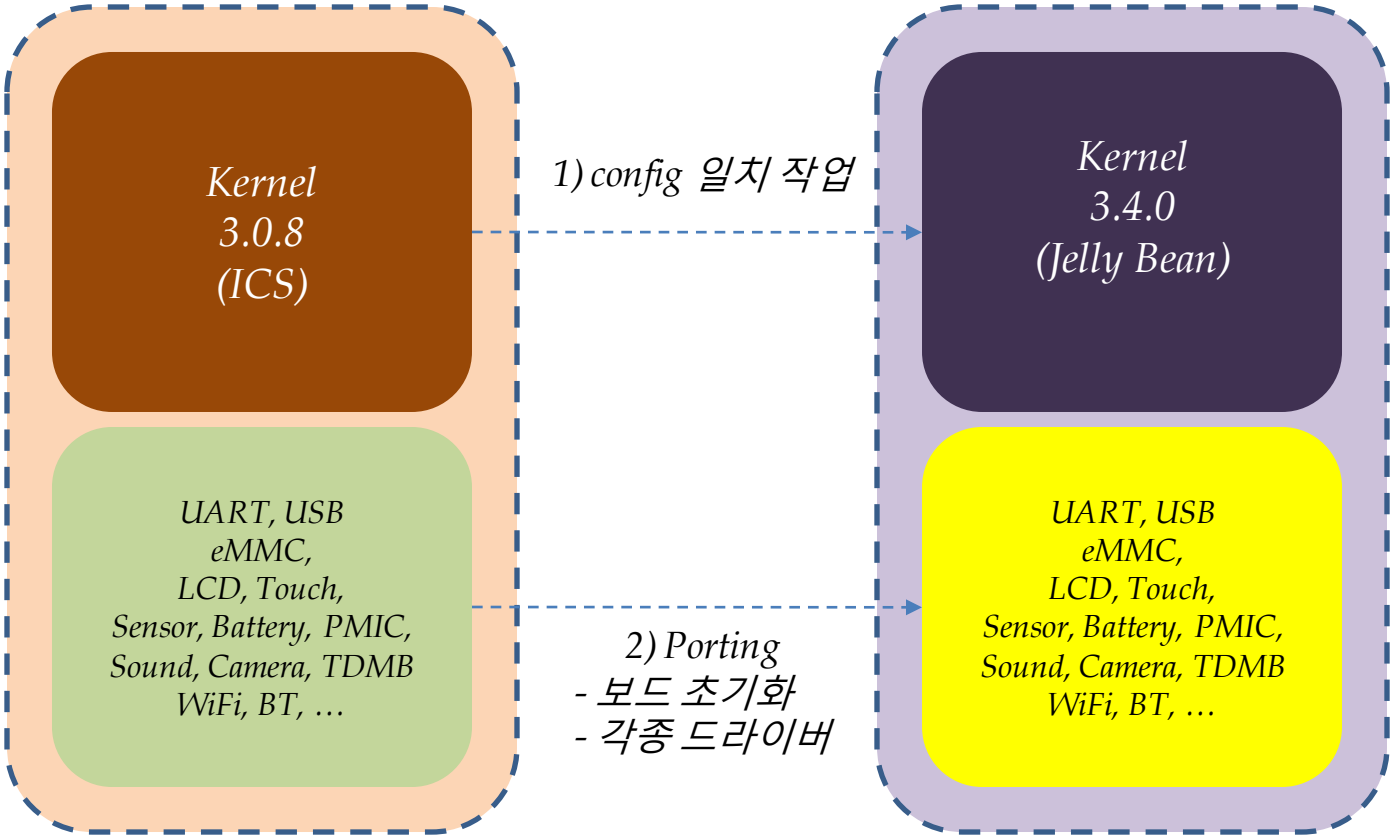
8. Jelly Bean BSP Porting Scope(2) - *Target Board 파일 분석*



8. Jelly Bean BSP Porting Scope(3) - ***lk bootloader***

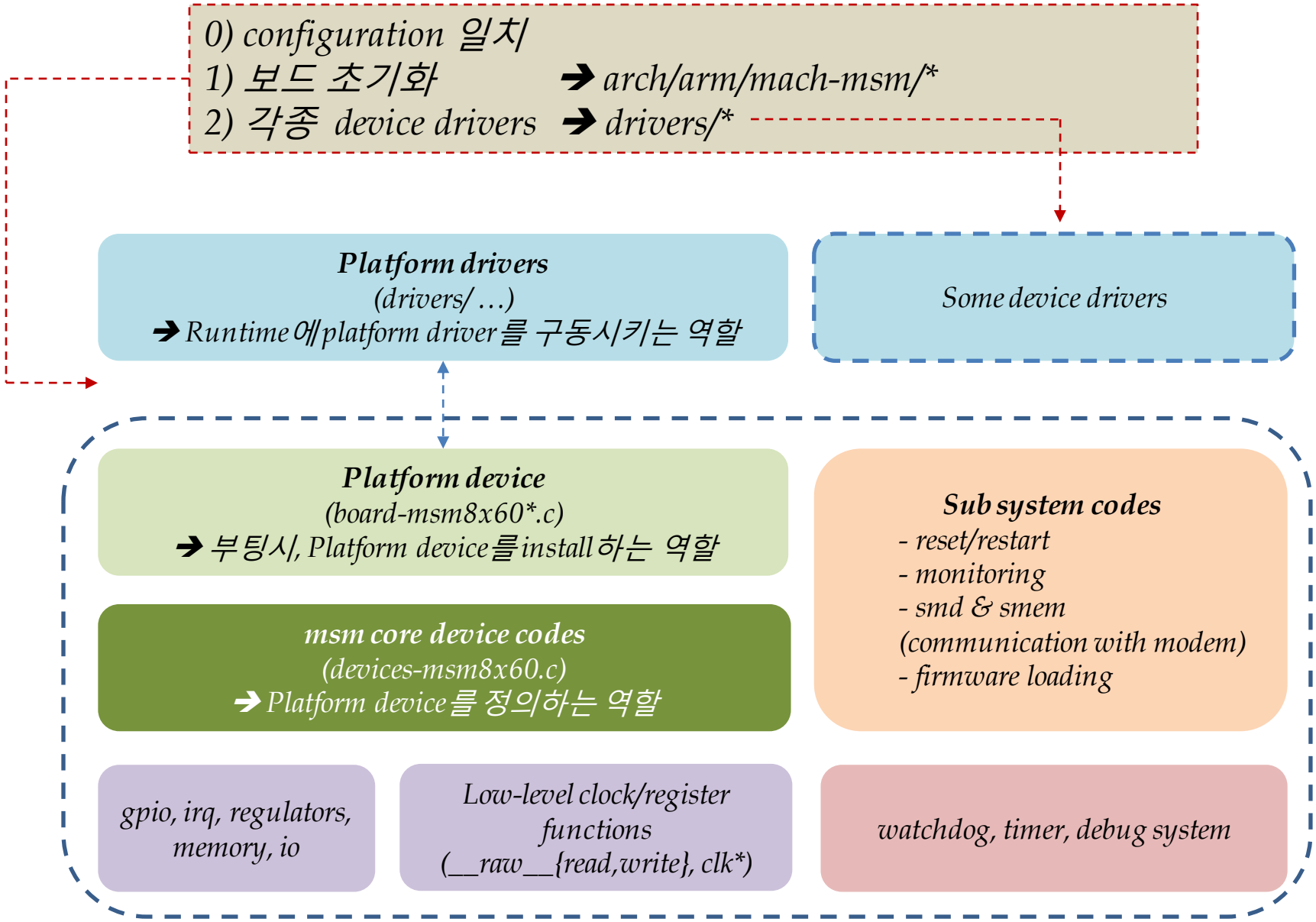


8. Jelly Bean BSP Porting Scope(4) – ***Linux Kernel/1***

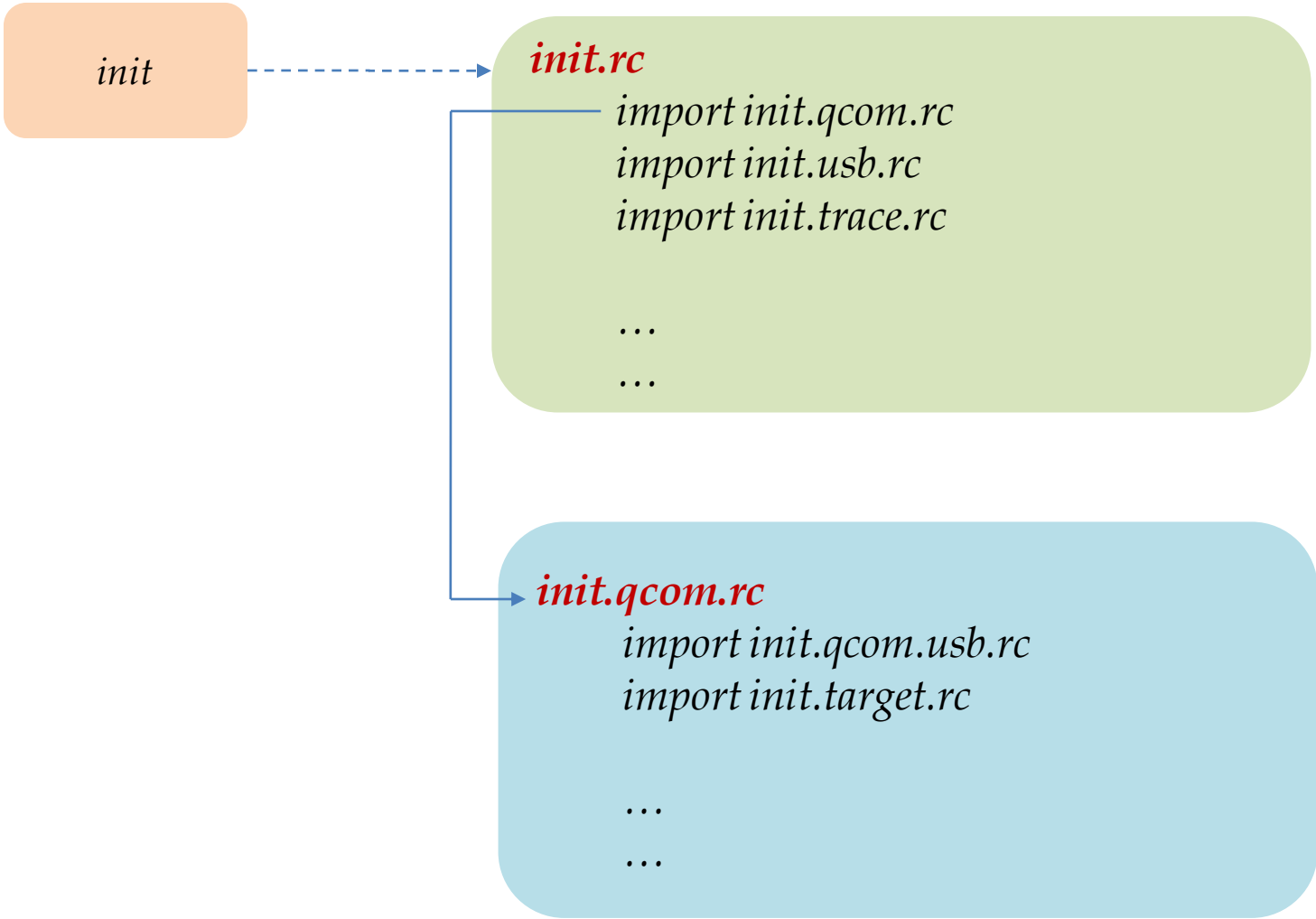


(*) 여기서 말하는 포팅은 ICS kernel에 대해 신규로 추가한 모든 내용(디바이스 드라이버 등)을 Jelly Bean kernel 기반으로 옮기는 작업을 뜻한다.
(*) 따라서, 1차적으로 kernel configuration을 일치시키는 작업을 진행해야 하며, 이후 신규로 추가한 코드 부분을 Jelly Bean kernel로 옮기는 작업을 진행해야 한다.

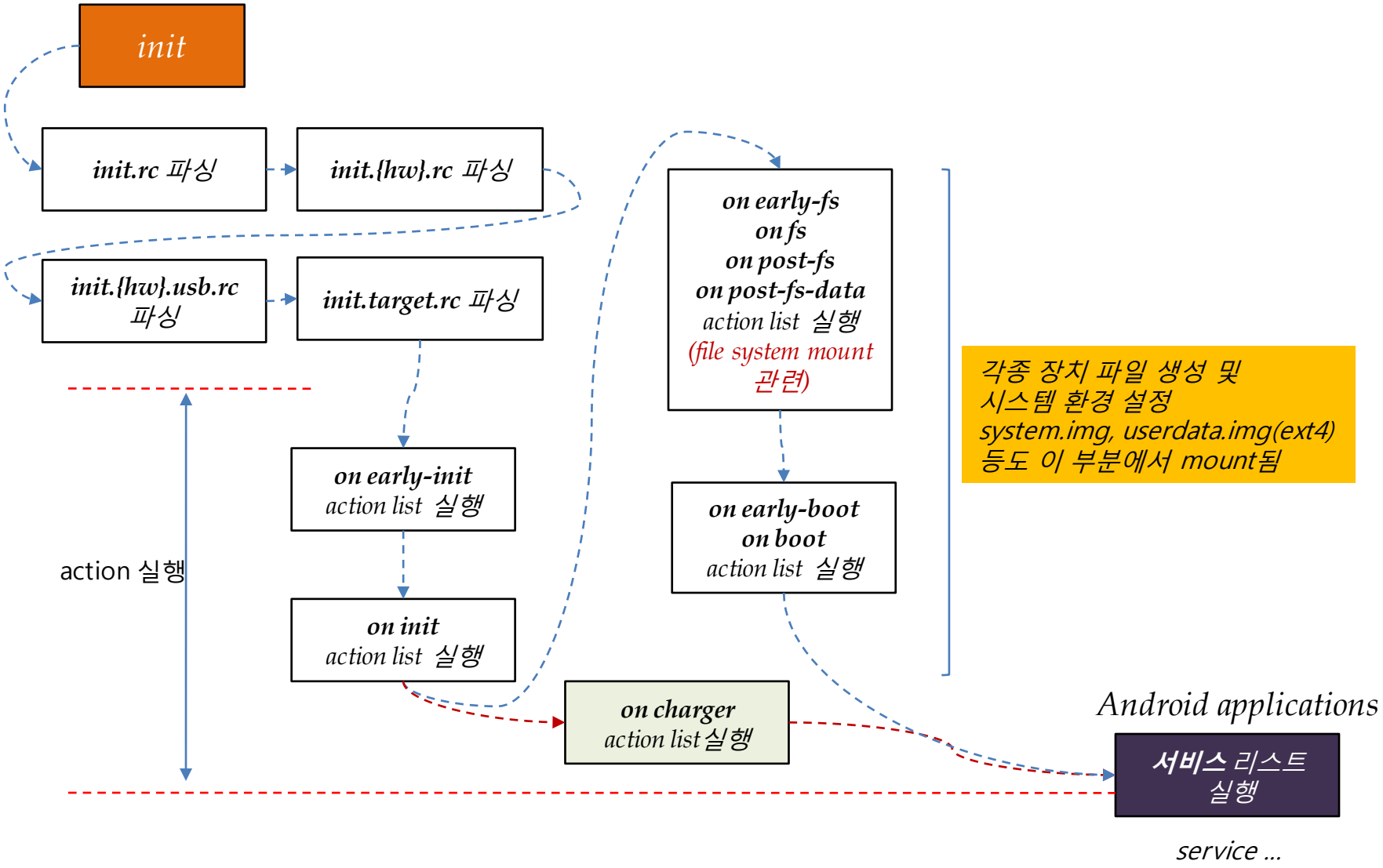
8. Jelly Bean BSP Porting Scope(4) – ***Linux Kernel/2***



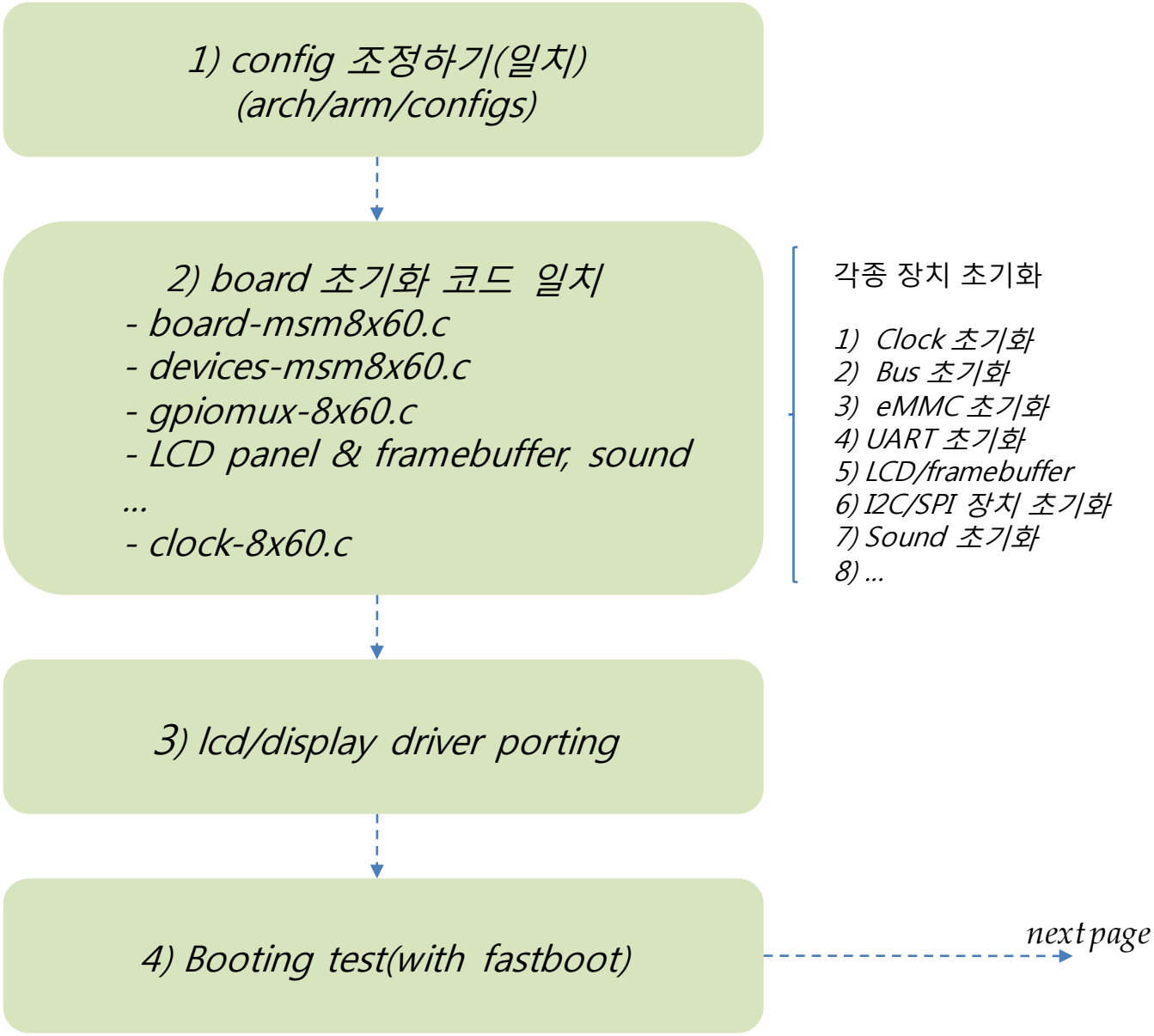
8. Jelly Bean BSP Porting Scope(5) – *init & init.rc/1*



8. Jelly Bean BSP Porting Scope(5) – *init & init.rc/2*

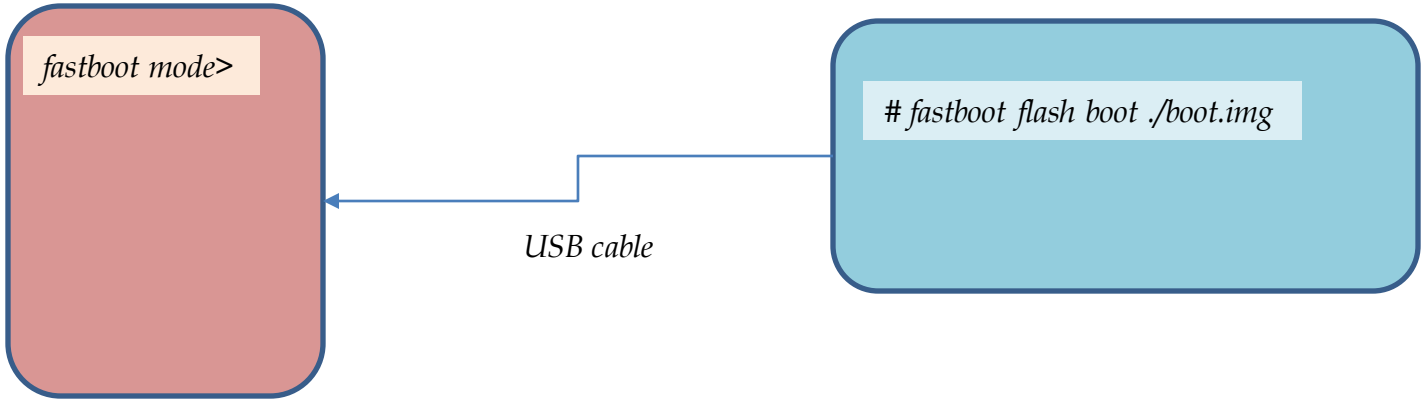


9. BSP Porting(1) – 1단계



9. BSP Porting(2) – 2단계 : fastboot로 image write 하기

- # adb reboot-bootloader ← fastboot mode로 전환(혹은 시스템에서 정의한 key 조합으로도 진입 가능)
- # fastboot flash boot ./boot.img
- # fastboot flash system ./system.img
- # fastboot erase userdata
- # fastboot flash userdata ./userdata.img
- # fastboot flash persist ./persist.img
- # fastboot flash cache ./cache.img



9. BSP Porting(3) – ***3 단계***

5) Debugging
➔ kernel panic 등 다양한 문제 해결 과정 필요

6) USB gadget driver porting
➔ adb가 동작하도록 …

7) mach-msm/ 나머지 부분 porting
- mdm
- sub system restart
- Reset
➔ Power off 시 정상 동작하도록 …

8) Battery/Charger driver porting
➔ 충전이 안되어 booting이 안될 경우 대비

9. BSP Porting(4) – ***4 단계: android target porting***

- *device/qcom/msm8660_surf/AndroidBoard.mk*
- *device/qcom/msm8660_surf/BoardConfig.mk*
- *device/qcom/msm8660_surf/init.qcom.mdm_links.sh*
- *device/qcom/msm8660_surf/init.target.rc*
- *device/qcom/msm8660_surf/msm8660_surf.mk*
- *device/qcom/msm8660_surf/recovery.fstab*
- *device/qcom/msm8660_surf/*.kl*
- *device/qcom/msm8660_surf/**

- *device/qcom/common/rootdir/etc/init.qcom.rc*
- *device/qcom/common/rootdir/etc/init.qcom.usb.rc*
- *device/qcom/common/rootdir/etc/ueventd.qcom.rc*
- *device/qcom/common/rootdir/etc/**

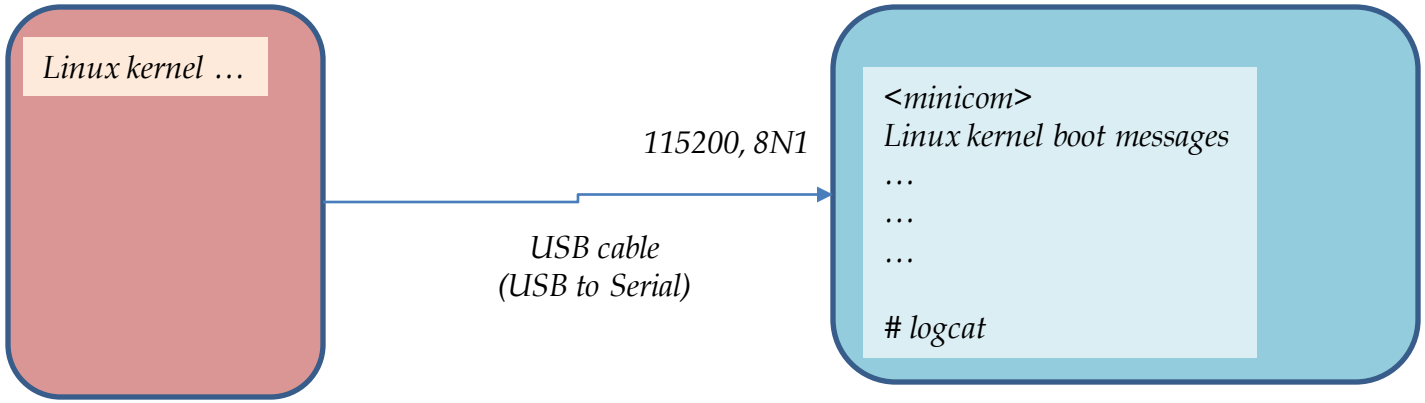
- *system/core/rootdir/init.rc*
- *system/core/rootdir/ueventd.rc*

9. BSP Porting(5) – *5 단계: debugging*

- 1) console
  - minicom – 115200, 8N1

- 2) adb
  - adb shell
  - logcat 으로 문제점 debugging

➔ *Board bring-up 단계에서는 모든 기능을 porting 한 것이 아니므로, 의외의 곳에서 시스템이 죽을 수 있음(주요 대상: surfaceflinger, system_server, mediaserver 등)*



9. BSP Porting(6) – *Kernel Boot Messages*

```
<6>[    0.000000] Booting Linux on physical CPU 0
<6>[    0.000000] Initializing cgroup subsys cpu
<5>[    0.000000] Linux version 3.4.0-perf-g49f91a2-dirty (chunghan.yi@gmail.com) (gcc version 4.6.x-google 20120106
 (prerelease) (GCC) ) #26 SMP PREEMPT Fri Oct 5 10:52:50 KST 2012
<4>[    0.000000] CPU: ARMv7 Processor [510f02d2] revision 2 (ARMv7), cr=10c5387d
<4>[    0.000000] CPU: PIPT / VIPT nonaliasing data cache, VIVT ASID tagged instruction cache
<4>[    0.000000] Machine: SLOWBOOT SLOWBOOT BOARD MSM8X60
<6>[    0.000000] memory pool 0 (start 38000000 size 600000) initialized
<6>[    0.000000] memory pool 1 (start 38600000 size 3802000) initialized
<6>[    0.000000] memory pool 3 (start 79400000 size 6c00000) initialized
<4>[    0.000000] Memory policy: ECC disabled, Data cache writealloc
<6>[    0.000000] socinfo_init: v6, id=71, ver=2.0, raw_id=1329, raw_ver=1, hw_plat=0,  hw_plat_ver=65537
<6>[    0.000000]  accessory_chip=58 hw_plat_subtype=0
<7>[    0.000000] On node 0 totalpages: 212992
<7>[    0.000000] free_area_init_node: node 0, pgdat c0cc7b80, node_mem_map c18eb000
<7>[    0.000000]   Normal zone: 992 pages used for memmap
<7>[    0.000000]   Normal zone: 0 pages reserved
<7>[    0.000000]   Normal zone: 104992 pages, LIFO batch:31
<7>[    0.000000]   HighMem zone: 1052 pages used for memmap
<7>[    0.000000]   HighMem zone: 105956 pages, LIFO batch:31
<6>[    0.000000] allocating 30175232 bytes at c7e00000 (48000000 physical) for fb
<6>[    0.000000] PERCPU: Embedded 9 pages/cpu @c9aca000 s15296 r8192 d13376 u36864
<7>[    0.000000] pcpu-alloc: s15296 r8192 d13376 u36864 alloc=9*4096
<7>[    0.000000] pcpu-alloc: [0] 0 [0] 1
<4>[    0.000000] Built 1 zonelists in Zone order, mobility grouping on.  Total pages: 210948
<5>[    0.000000] Kernel command line: console=ttyDCC0,115200,n8 console=ttyHSL0,115200,n8 androidboot.hardware=qcom
 kgsl.mmutype=gpummu vmalloc=512M uart_console=disable slowboot.rev=rev_b slowboot.hreset=off slowboot.reboot=pwroff
 slowboot.lcd=off slowboot.batt_info=<null> slowboot.pwron=usb_chg slowboot.reset=rst_etc slowboot.usb_cable=normal
kcal= 0 0 0 0 0 0 mdm_force_dump_enabled idv2 androidboot.emmc=true androidboot.serialno=64005dc3 slowboot.signed_im
age=true androidboot.baseband=svlte2a
<6>[    0.000000] BOARD : SLOWBOOT rev_b
<6>[    0.000000] BOARD : SLOWBOOT target 2
<6>[    0.000000] usb_cable_info : normal (8)
<6>[    0.000000] ####kcal=, r=0, g=0, b=0  sign=0,0,0
<6>[    0.000000] PID hash table entries: 2048 (order: 1, 8192 bytes)
<6>[    0.000000] Dentry cache hash table entries: 65536 (order: 6, 262144 bytes)
<6>[    0.000000] Inode-cache hash table entries: 32768 (order: 5, 131072 bytes)
<6>[    0.000000] Memory: 44MB 788MB = 832MB total
<5>[    0.000000] Memory: 794416k/794416k available, 168144k reserved, 428032K highmem
<5>[    0.000000] Virtual kernel memory layout:
<5>[    0.000000]     vector  : 0xffff0000 - 0xffff1000   (   4 kB)
<5>[    0.000000]     fixmap  : 0xfff00000 - 0xfffe0000   ( 896 kB)
<5>[    0.000000]     vmalloc : 0xdf800000 - 0xff000000   ( 504 MB)
<5>[    0.000000]     lowmem  : 0xc0000000 - 0xdf000000   ( 496 MB)
<5>[    0.000000]     pkmap   : 0xbfe00000 - 0xc0000000   (   2 MB)
                                                                                40,1          꼭대기
```

9. BSP Porting(7) – *ps result(busybox ps)*

```
  100 0             0 SW<  [krfcommd]
  101 0             0 SW<  [msm-cpufreq]
  102 0             0 SW<  [rq_stats]
  103 0             0 SW<  [deferwq]
  104 0             0 SW<  [fsa8008]
  105 0             0 SW   [irq/639-fsa8008]
  106 0             0 SW   [irq/640-fsa8008]
  107 0           376 S    /sbin/ueventd
  108 0             0 SW   [jbd2/mmcblk0p26]
  109 0             0 SW<  [ext4-dio-unwrit]
  110 0             0 SW   [jbd2/mmcblk0p20]
  111 0             0 SW<  [ext4-dio-unwrit]
  112 0             0 SW   [jbd2/mmcblk0p29]
  113 0             0 SW<  [ext4-dio-unwrit]
  114 0             0 SW   [jbd2/mmcblk0p28]
  115 0             0 SW<  [ext4-dio-unwrit]
  116 0             0 SW   [jbd2/mmcblk0p12]
  117 0             0 SW<  [ext4-dio-unwrit]
  118 0             0 SW   [jbd2/mmcblk0p27]
  119 0             0 SW<  [ext4-dio-unwrit]
  120 0             0 SW   [jbd2/mmcblk0p23]
  121 0             0 SW<  [ext4-dio-unwrit]
  123 1000        916 S    /system/bin/servicemanager
  125 0          4184 S    /system/bin/vold
  128 0         10040 S    /system/bin/netd
  129 0          1212 S    /system/bin/debuggerd
  133 1019       8884 S    /system/bin/drmserver
  134 1013      29740 S    /system/bin/mediaserver
  135 1002       1428 S    /system/bin/dbus-daemon --system --nofork
  136 0           928 S    /system/bin/installd
  138 1017       1944 S    /system/bin/keystore /data/misc/keystore
  144 2000        844 S    /system/bin/sh
  145 9999        908 S    /system/bin/rmt_storage
  146 0          8892 S    /system/bin/time_daemon
  147 0          5504 S    /sbin/adbd
  170 1001       4048 S    /system/bin/qmuxd
  173 1001       5204 S    /system/bin/qmiproxy
  176 0          4148 S    /system/bin/netmgrd
  903 0          5264 S <  /system/bin/mpdecision --no_sleep --avg_comp
  983 1000       6096 S    /system/bin/mm-pp-daemon
 1238 0          440m S    zygote /bin/app_process -Xzygote /system/bin --zygot
 1239 1000      74312 S    /system/bin/surfaceflinger
 1279 1000       554m S    system_server
 1430 10042      471m S    {ndroid.systemui} com.android.systemui
 1487 10012      453m S    {d.process.media} android.process.media
 1492 10019      452m S    {putmethod.latin} com.android.inputmethod.latin
 1519 1001       460m S    {m.android.phone} com.android.phone
 1530 10020      482m S    {ndroid.launcher} com.android.launcher
```

Thanks a lot !

SlowBoot