Driver All In One – MT6575 ICS

MSZ
20120109

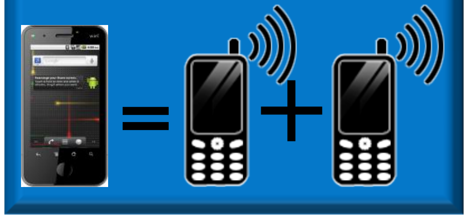
Outline

- Overview
- Build system
- EMI
- Download and Boot Up
- DCT
- Lights System
- Touch
- LCM
- Sensor System
- Connectivity
- Battery Manager
- Audio
- Misc.
- Appendix

Overview

Strength of MT6575 Smartphone Platform

Dual SIM



One-stop Service: Analog TV BT/FM/Wi-Fi/GPS

Wi Fi

GPS

Video Telephony

720p Video Playback

Video Streaming

youku 优酷 .com

YouTube

Full HTML Browser

Better Performance

Kaixin.com 开心网

人人网 renren.com

facebook

MT6575

CA9 1GHz AP

Rel. 6 HSPA Modem

Use External PMIC

8M Camera

FWVGA Video Record

Face Detection / Smile Shot

3D Acceleration

Launcher , Live Wallpaper

Gallery, Games

Tier 1 Performance

Audio/Speech/Modem

Low Power

Turnkey Solution

Android latest version

IOT, FTA, CTS 100% Pass

MT6575 Platform – Key Features

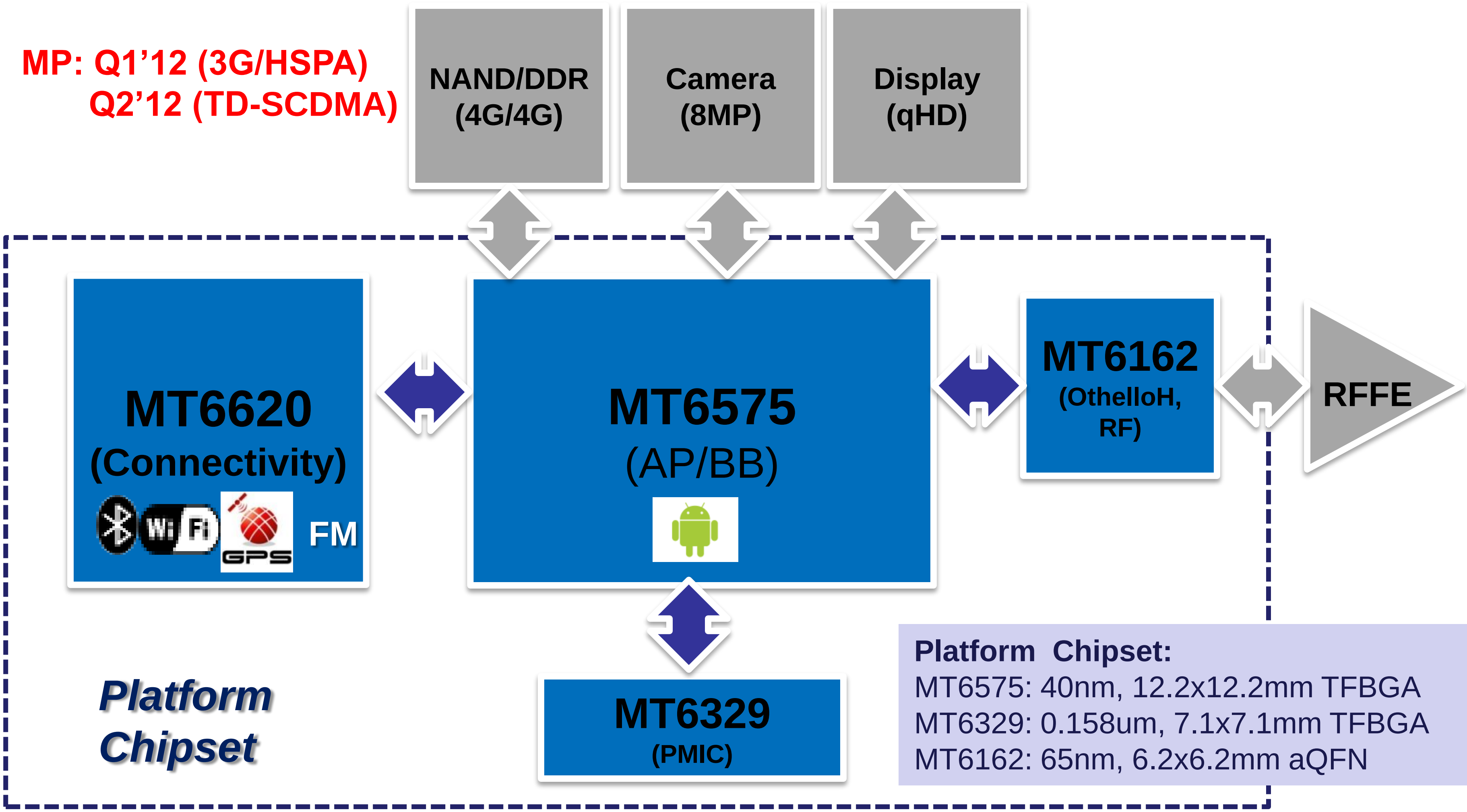
- **Highly-integrated 40nm AP/modem SoC for mainstream smartphones**
  - High performance **1GHz Cortex-A9 MCU** with
    - 32K/32K L1 Cache, 256 KB L2 Cache, NEON co-proc.
  - Rel. 6 HSPA modem (7.2/5.76 Mbps) integrated
  - Stand-alone advanced PMIC (MT6329)

wiFi
4:00 PM
Google
Rearrange your Home screen.
Touch & hold an item and when it vibrates, drag it where you want.

- **Advanced multi-media subsystem**
  - **High-definition 720p**, 30fps video encode/decode
  - **8 MP@13fps** camera with integrated ISP and rich features
  - Improved 3D Graphics (OpenGL ES 1.1/2.0) performance
  - **qHD (960x540)**/24-bit Color Display Controller with
    - **HDMI/MHL** support via external component
  - **Flash 10.3**, **Stereo 3D** image/video/display support
- **Multi-mode connectivity enabled by highly-integrated chipset**
  - 4-band 3G/4-band EDGE cellular modem (MT6162 OthelloH RF)
  - BT 3.0 HS + BT 4.0 LE (MT6620, 4-in-1 Combo)
  - WiFi 802.11 a/b/g/n (MT6620)
  - FM Rx/Tx (MT6620), GPS (MT6620)

MT6575 Platform Block Diagram

MP: Q1’12 (3G/HSPA)
Q2’12 (TD-SCDMA)



NAND/DDR (4G/4G)

Camera (8MP)

Display (qHD)

MT6620 (Connectivity)

FM

MT6575 (AP/BB)

MT6162 (OthelloH, RF)

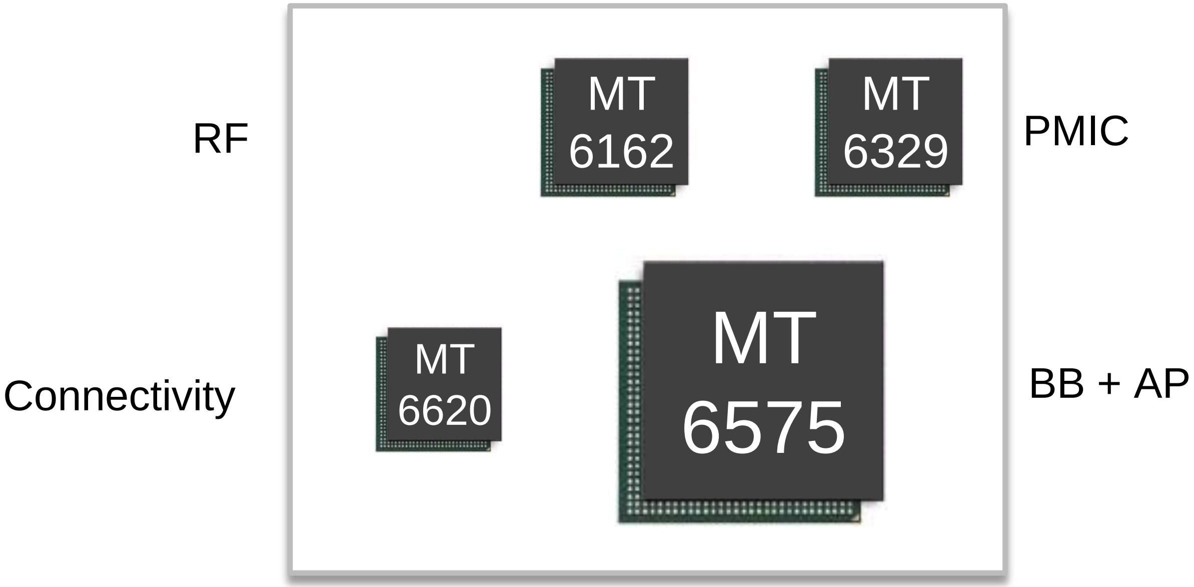
RFFE

MT6329 (PMIC)

Platform Chipset

Platform Chipset:
MT6575: 40nm, 12.2x12.2mm TFBGA
MT6329: 0.158um, 7.1x7.1mm TFBGA
MT6162: 65nm, 6.2x6.2mm aQFN

MT6575 Platform – Chipset



MT6575 Platform		Package Size	Package Type	Ball Pitch	Pin number
BB+AP	MT6575	12.2x12.2	TFBGA	0.4	537
RF	MT6162	6.2x6.2	aQFN	0.65	62
PM	MT6329	7.1x7.1	TFBGA	0.5	155
WCN	MT6620	5.3x5.7	WLCSP	0.4	149

MT6575 – Applications Sub-System

- **1GHz ARM Cortex-A9 + Neon (FPU)**
  - 32kB I-Cache, 32kB D-Cache
  - 256KB L2 Cache
  - Serial-Wire Debugger
  - Supports DVFS from 0.9V to 1.2V (1.1typ)
- **1x32-bit external memory interface**
  - mDDR/LPDDR2/PCDDR3
  - 200 / 266MHz
  - 2 CS (max 512MB DDR/1GB DDR2)
- **1x8/16-bit SLC NAND interface**
  - 12-bit HW ECC/EDC
  - 512/2k/4k page size
  - eMMC v4.41 support
- **ARM™ TrustZone® Security**

- **Rich Applications Interfaces**
  - USB2.0 high-speed OTG supporting 15 Tx and 15 Rx endpoints
  - 4 x UART to 3Mbps (EDR)
  - IrDA FIR/MIR/SIR
  - Dedicate 1 x SPI interface(up to 52M bps)
  - Dedicate 3 x I2C interfaces
  - Dedicate 2 x I2S interface
  - Dedicate 4 x SDIO interfaces (SD/MMC, eMMC v4.41,SD3.0)
  - 2 x SIM interfaces(IC-USB)
  - 7 x PWM channels
  - 7 x GP Timers
  - 5 x AuxADC
  - 8x8 QWERTY + 2 Key support
  - Touch panel I/F
  - Dedicate 2 pin for DVS Control

MT6575 – Multimedia Sub-System

- **Multi-format, 720p HD Video**
  - 30fps encode/decode (HW accel. + SW)
  - MPEG4, H.264, H.263, VP8 Codecs
  - **Other Codecs by Customer Request**
- **8 MP Camera Sub-system**
  - Integrated ISP with AF, Video/Image stabilization & many other features
  - 8MP@13fps
  - MIPI CSI-2 interface (2-lane, 1Gbps)
  - 10-bit, 96 MHz parallel I/F
  - JPEG decoder/encoder (35/75MP/s)
- **3D Graphics (OpenGL ES 2.0)**
  - 266MHz, 22MΔ/s, 700 MP/s
- **Display controller**
  - Main Display to qHD (960x540), S3D support
  - 2-lane MIPI-DSI (1GHz)
  - Command and video modes support
  - 24-bit Color, 6 blending layers
  - 1 Sub-Display to QCIF (serial)
- **TV-Out (NTSC/PAL), HDMI support**

- **HD Video Decode**
  - MPEG4, 720p 30fps SP/ASP profile
  - H.263, 720p 30fps
  - H.264, 720p 30fps BL, 24fps MP/HP, L3.1
  - VP8 720p 30fps
- **HD Video Encode**
  - MPEG4, 720p 30fps, SP profile
  - H.263, 720p 30fps
  - H.264/VP8, VGA 15fps
- **Video Streaming**
  - MPEG4/H.263/H.264 to D1, 30fps
- **3G-324M Video Telephony**
- **Audio Codecs**
  - **Decode**: MP3, MP2, AAC, AMR, WB-AMR, MIDI, Ogg Vorbis, WAV,
  - **Encode**: AMR-NB, AMR-WB, AAC
- **SW Audio Sound Effects**
  - Integrated SRS audio post-processing
  - HD Voice, Dual-Microphone noise reduction

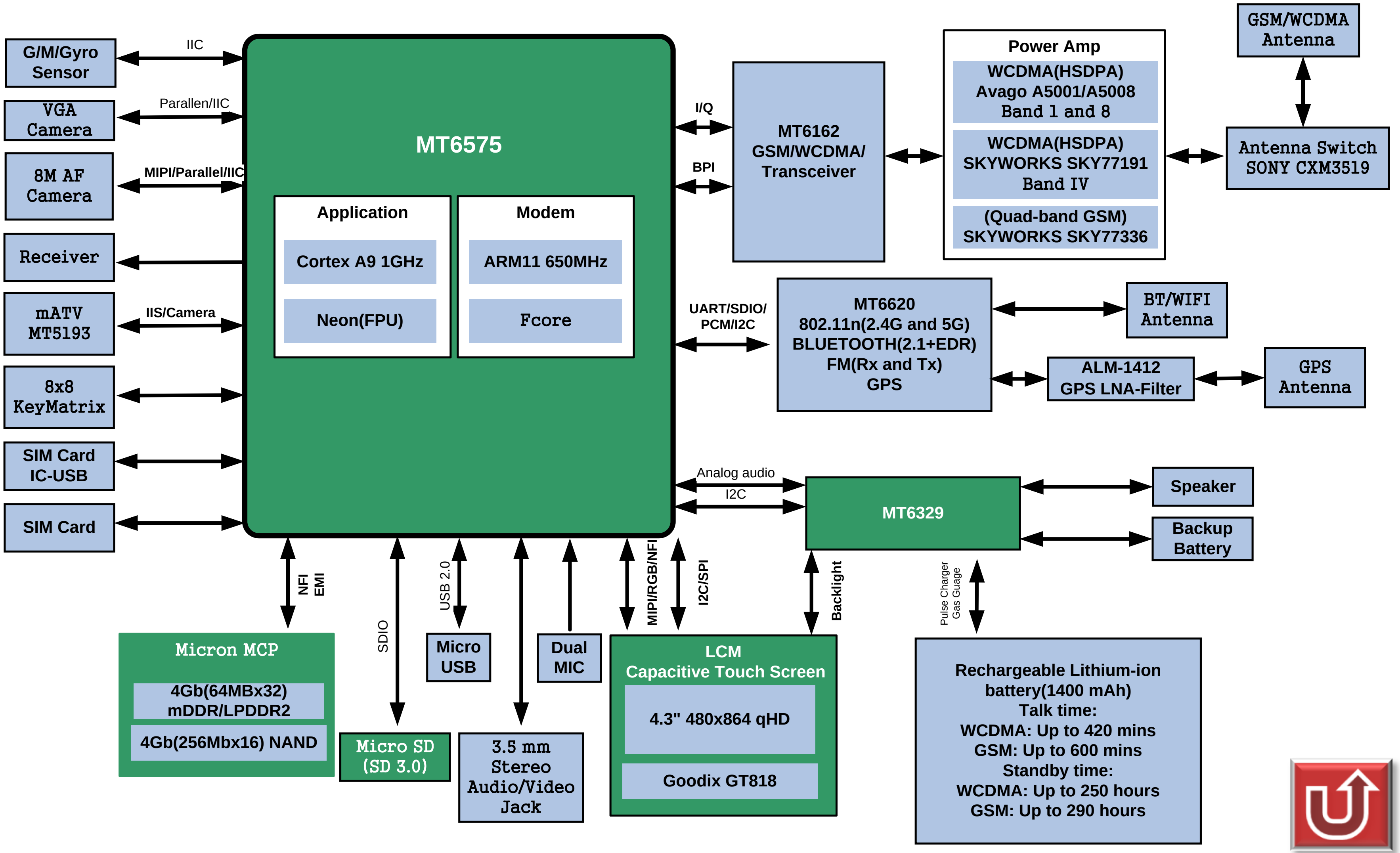
MT6575 – Modem Sub-System

- 3GPP Rel. 6 HSPA
  - Cat 8, 7.2 Mbps DL; Cat 6, 5.76 Mbps UL
  - Class 12 EDGE / GPRS
  - Dual SIM (Single Talk)
  - FR, HR, EFR, AMR, WB-AMR,SAIC
  - Dedicate 3 pin for PA Control
- 520MHz ARM1176JZ + 260MHz DSP modem core
  - Dedicated 32KB I-Cache and 32KB D-Cache
  - 64KB I- and 64KB D- tghtly couple memory
  - 96KB L2 tightly couple memory
  - Hardware-based 2G and 3G modems
  - 2xUART for Debug , Remove SATA
- Multi-mode OthelloH RF
  - 4-band 3G/4-band EGPRS
  - Band I, II, V, VIII on initial reference design for full-system qualification
- TD-SCDMA Capable (Addition of AST3001 + OT RF)

MT6573 / MT6575 Feature Comparison

	Mediatek MT6573	Mediatek MT6575
Package	12.6x12.6mm 519 balls, BGA, 0.4mm, 65nm	12.2x12.2mm 537 balls, BGA, 0.4mm, 40nm
Apps Processor	ARM1176JZFS@ 650 MHz w/ 32KB/32KB I/D cache 128KB L2, DVFS support	Cortex-A9@ 1GHz w/ 32KB/32KB I/D cache 256KB L2, DVFS support , Serial Debug Port
Modem Processor	ARM1176JFS@ 520 MHz, 280MHz DSP	ARM1176JFS@ 520 MHz, 260MHz DSP,96K L2
Modem	EDGE class12, HSDPA Cat8 7.2Mbps, HSUPA Cat6 5.76Mbps	EDGE class12, HSDPA Cat8 7.2Mbps, HSUPA Cat6 5.76Mbps
Memory	200MHz-mDDR, 256MB , NAND (NFI)	200MHz-mDDR, 256MB , NAND (NFI) , LPDDR2 ,PCDDR3
Camera / TV-out	8MP Bayer/YUV, 10bit parallel, MIPI CSI-2 CVBS TV out	8MP Bayer/YUV, 10bit parallel, MIPI CSI-2 CVBS TV out
Display	FWVGA, 24-bit color, MIPI DSI, NFI, CPU/RGB I/F	qHD, 24-bit color, MIPI DSI(video), NFI, CPU/RGB I/F,3D,HDMI
Audio	64-Poly, MP3,AAC,HE-AAC, WMA,G.711,G.723.1,G.729,AWB+,3D effect Dual mic, Digital mic support	64-Poly, MP3,AAC,HE-AAC, WMA,G.711,G.723.1,G.729,AWB+,3D effect Dual mic, Digital mic support
Video Decode	MPEG4/H.264: FWVGA @ 30fps	MPEG4/H.264: 720p @ 30fps
Video Encode	MPEG4: FWVGA @ 30fps, H.264 CIF @ 30fps	MPEG4: 720p @ 30fps, H.264 VGA @ 15fps
Video Telephony	3G-324M: QCIF 15 FPS, 64kbps	3G-324M: QCIF 15 FPS, 64kbps
Video Streaming	MPEG4/H.264 to D1, 30 fps	MPEG4/H.264 to D1, 30 fps
Peripherals	UARTx4, SIMx2, R-touch, PCMx1, I2Sx2, I2Cx2, SPIx1 SDIOx4 , Key Matrix 8x8 USB 2.0 HS OTG int. PHY x 1, USB FS host x 1	UARTx4, SIMx2, R-touch, PCMx1, I2Sx2, I2Cx3, SPIx1 SDIOx4 , Key Matrix 8x8 +2,MD UARTx2 USB 2.0 HS OTG int. PHY x 1, USB FS host x 1
Power Management	Integrate PMU, 5 bucks, 20 LDOs, LED driver, pulse charger, gas gauge	External PMU, 5 bucks, 21 LDOs, LED driver, pulse charger, gas gauge, Flash LED, Audio AMP, Analog SW, ISINK , 35V Boost controller, DVS Control, PA Control , Force Power Reset , Pre charge indicator , 2 Key support

MT6575 System Block Introduction



Confidential A

MEDIATEK

Build System

Outline

- Build System Environment
- MTK Wrapped Command
- Quick Build with Native Command
- Suggested Useful Command

Build System Environment

- Refer to document <Android 4.0 Build Environment on Ubuntu 10.04 64-bit Installation SOP> for environment setup details
  - DCC path:

3G Phone Data/Smart Phone/Software_Customer/Document Library/Build/V4.0

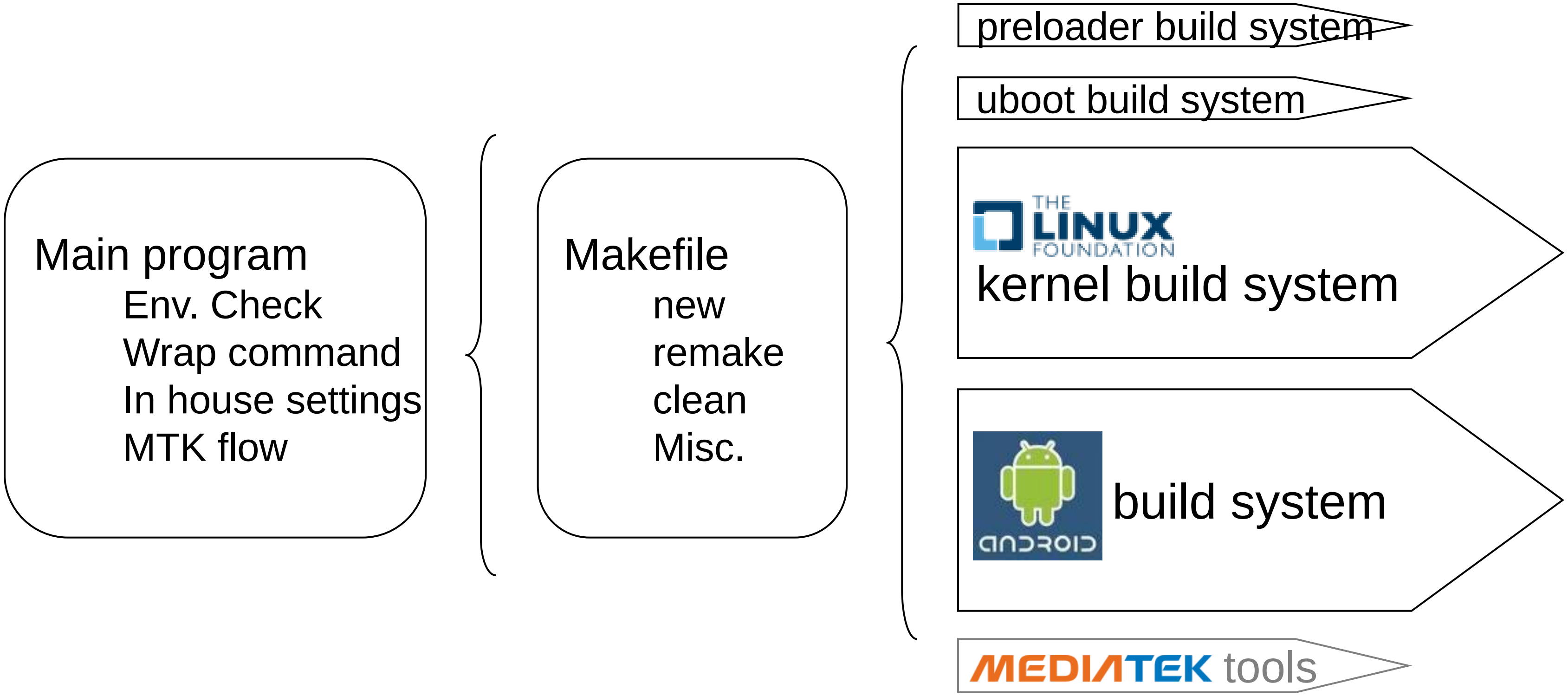
- Build procedure starts from environment checking, with check-env.log generated @alps folder.
  - Check check-env.log for the specific environment requirements of your project
- environment checking command: “./mk check-env”

build environment reference

	suggested	mtk
OS	Ubuntu 10.04	Ubuntu 10.04 (64-bit)
memory size	4G or above	12G
make	GNU make 3.81	3.81
perl	5.10.X	5.10.1
python	2.6.X	2.6.5
arm-linux-androideabi-gcc	4.4.x	4.4.3
gcc	4.4.3	4.4.3
jdk	1.6.X	16.0_23
wine	1.1 or above	1.1.33
bison	2.4.X	2.4.1
flex	2.5.X	2.5.35
gperf	3.0.X	3.0.3
mingw	Installed	Installed
unix2dos/tofrodos	Installed	Installed

MTK wrapped build command

- usage
  - (makeMtk | mk) [options] project actions [modules]



Main program
Env. Check
Wrap command
In house settings
MTK flow

Makefile
new
remake
clean
Misc.

preloader build system

uboot build system

THE LINUX FOUNDATION kernel build system

android build system

MEDIATEK tools

(makeMtk | mk) [options] project actions [modules]

- makeMtk | mk
  - mk → ./ makeMtk文件
- Abrrev mk, show help if without any arguments

- (makeMtk | mk) **[options]** project actions [modules]

  - options
    - -t, -tee : Print log information on the stand-out
    - -o,-opt=bypass_argument_to_make: pass extra arguments to make
    - -h, -help: Print usage message and exit.

(makeMtk | mk) [options] **project** actions [modules]

- project
  - one of available projects，such as ztemt73v3_2
  - absence is allowed:
    - 延续上次build的project
    - 保存在文件makeMtk.ini中

➢ 命令mk listp可以查看available projects

➢ 第一次build命令中写上project，后续可以省略

➢ 如果怀疑project不对，可查看makeMtk.ini文件

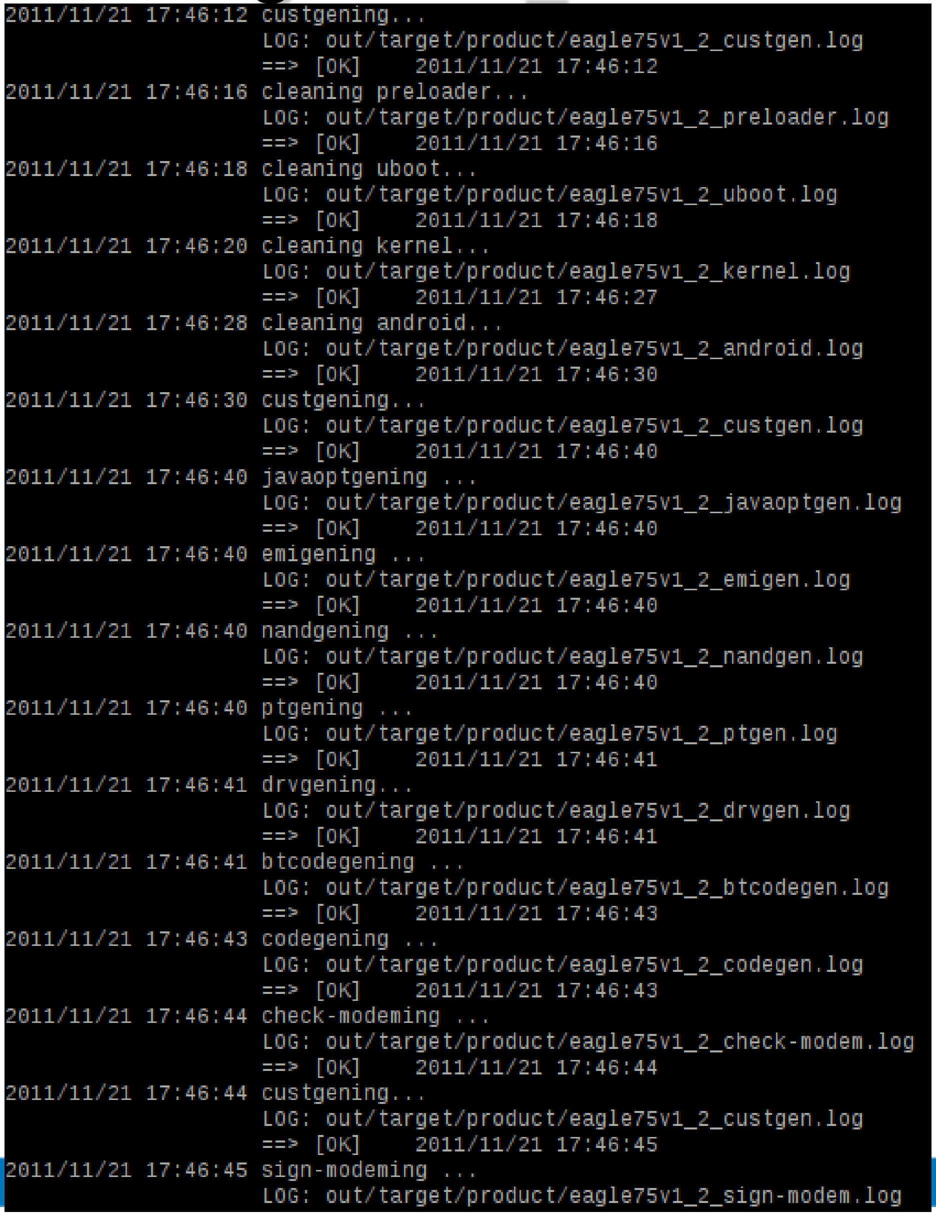
- (makeMtk | mk) [options] project **actions** [modules]

  - actions
    - new, clean, remake, bm_new, bm_remake, mm
    - emigen, nandgen, ptgen
    - bootimage, systemimage, userdataimage

- (makeMtk | mk) [options] project actions **[modules]**

argument	specify to build
pl, preloader	preloader
ub, uboot	uboot
k, kernel	kernel
dr, android	android
k <module path>	kernel module with the source path
dr <module name>	android module with module name
	all modules

./mk eagle75v1_2 new



```
2011/11/21 17:46:12 custgening...
                    LOG: out/target/product/eagle75v1_2_custgen.log
                    ==> [OK]    2011/11/21 17:46:12
2011/11/21 17:46:16 cleaning preloader...
                    LOG: out/target/product/eagle75v1_2_preloader.log
                    ==> [OK]    2011/11/21 17:46:16
2011/11/21 17:46:18 cleaning uboot...
                    LOG: out/target/product/eagle75v1_2_uboot.log
                    ==> [OK]    2011/11/21 17:46:18
2011/11/21 17:46:20 cleaning kernel...
                    LOG: out/target/product/eagle75v1_2_kernel.log
                    ==> [OK]    2011/11/21 17:46:27
2011/11/21 17:46:28 cleaning android...
                    LOG: out/target/product/eagle75v1_2_android.log
                    ==> [OK]    2011/11/21 17:46:30
2011/11/21 17:46:30 custgening...
                    LOG: out/target/product/eagle75v1_2_custgen.log
                    ==> [OK]    2011/11/21 17:46:40
2011/11/21 17:46:40 javaoptgening ...
                    LOG: out/target/product/eagle75v1_2_javaoptgen.log
                    ==> [OK]    2011/11/21 17:46:40
2011/11/21 17:46:40 emigening ...
                    LOG: out/target/product/eagle75v1_2_emigen.log
                    ==> [OK]    2011/11/21 17:46:40
2011/11/21 17:46:40 nandgening ...
                    LOG: out/target/product/eagle75v1_2_nandgen.log
                    ==> [OK]    2011/11/21 17:46:40
2011/11/21 17:46:40 ptgening ...
                    LOG: out/target/product/eagle75v1_2_ptgen.log
                    ==> [OK]    2011/11/21 17:46:41
2011/11/21 17:46:41 drvgening...
                    LOG: out/target/product/eagle75v1_2_drvgen.log
                    ==> [OK]    2011/11/21 17:46:41
2011/11/21 17:46:41 btcodegening ...
                    LOG: out/target/product/eagle75v1_2_btcodegen.log
                    ==> [OK]    2011/11/21 17:46:43
2011/11/21 17:46:43 codegening ...
                    LOG: out/target/product/eagle75v1_2_codegen.log
                    ==> [OK]    2011/11/21 17:46:43
2011/11/21 17:46:44 check-modeming ...
                    LOG: out/target/product/eagle75v1_2_check-modem.log
                    ==> [OK]    2011/11/21 17:46:44
2011/11/21 17:46:44 custgening...
                    LOG: out/target/product/eagle75v1_2_custgen.log
                    ==> [OK]    2011/11/21 17:46:45
2011/11/21 17:46:45 sign-modeming ...
                    LOG: out/target/product/eagle75v1_2_sign-modem.log
```

Code Path

- project source path: mediatek\custom\{$project}
- build log path: out\target\product
- After a successful build, many files will be generated @ out\target\product\{$project}
  - Use flash tool to open the scatter file “MT6575_Android_scatter.txt“ and then download

Name	File
PRELOADER	preloader_eagle75v1_2.bin
DSP_BL	DSP_BL
UBOOT	uboot_eagle75v1_2.bin
BOOTIMG	boot.img
RECOVERY	recovery.img
SEC_RO	secro.img
LOGO	logo.bin
ANDROID	system.img
USRDATA	userdata.img

Quick Build with Native Command

- 1. Native command is supported by wrapped MTK build command as follows:
  - ./mk <project> mm <module_path>

- 2. Extends the functionality by command “source ./build/envsetup”

1) mm ➔ build the single module at the sub-directory, as follows:

- – cd <module_path>
- – TARGET_PRODUCT=<project> mm

2) mmm ➔ build the single module at the root-directory, as follows:

- – TARGET_PRODUCT=<project> mmm <module_path>

3) m command

- – TARGET_PRODUCT=<project> m <module_name>
  - ➢ Will pack system.img
- – TARGET_PRODUCT=<project> m
  - ➢ Take long time > 10 min

Suggested Useful Command

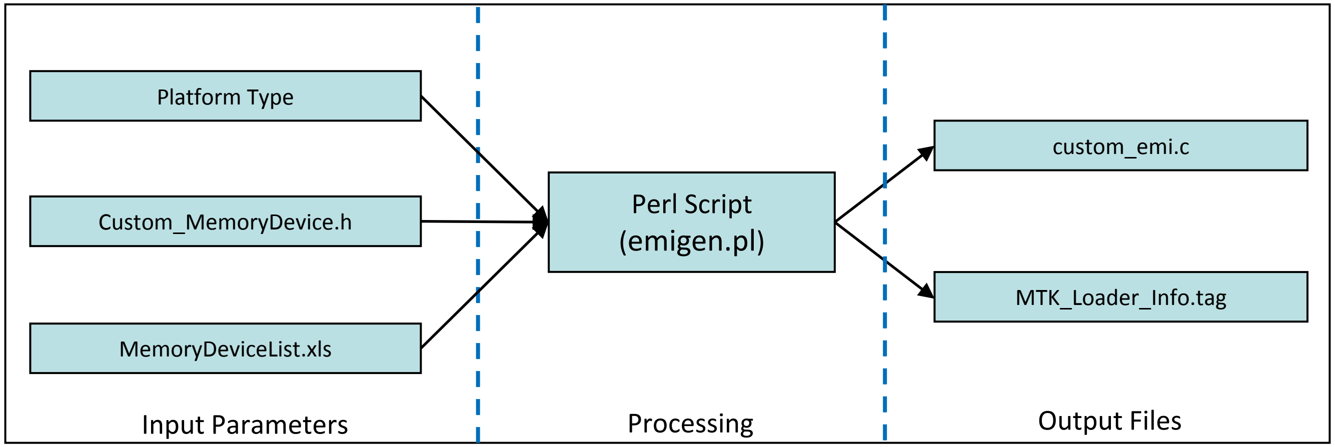
- Build Bootimage
  - 1) ./mk <project> remake kernel
  - 2) ./mk <project> bootimage
- Build logo.bin
  - ./mk <poject> remake uboot
- Generate DCT code
  - ./mk <project> codegen
- Synchronize MTK folder’s sourcecode
  - ./mk <project> custgen
- Build release version binary
  - ./mk -opt=TARGET_BUILD_VARIANT=user <project> new

MEDIATEK

EMI Customization

EMI Customization

- Introduction
  - Perl script is used to auto generate source file and header file of DDR initialization.
    - Location: alps/mediatek/build/tools/emigen/${platform}/emigen.pl
  - Memory DB file
    - Location: alps/mediatek/build/tools/emigen/${platform}/MemoryDeviceList.xls
    - Please update this file when more memory devices have been verified.



EMI Customization

- Change file list

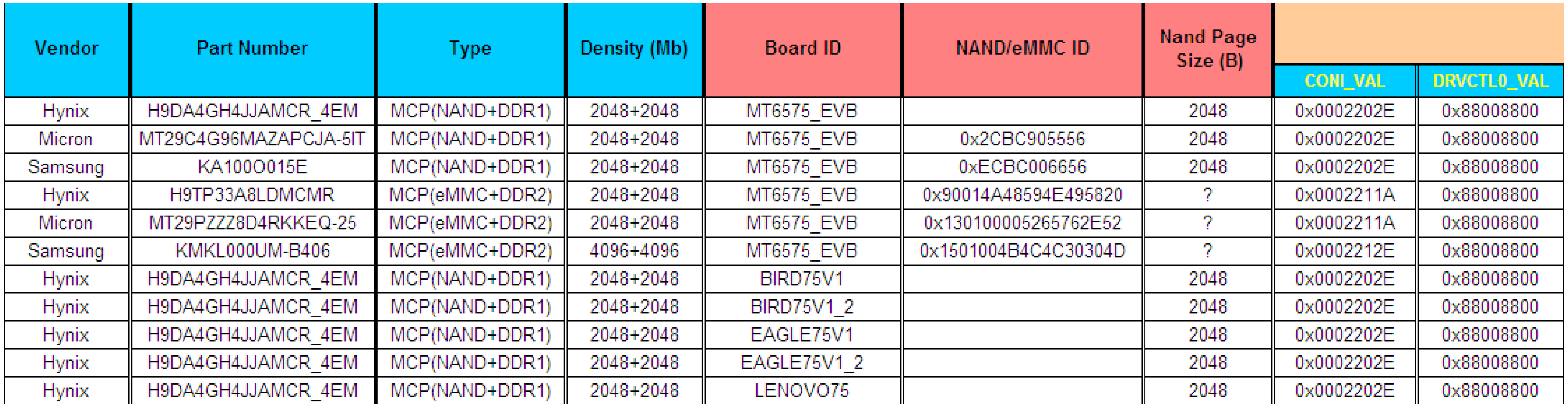
File	Description
Alps/mediatek/custom/${project_name}/preloader/inc/	
Custom_MemoryDevice.h	The customization file for EMI setting

- Customization item
  - Custom_MemoryDevice.h
  - Eg.,support 3 types MCP
    - #define BOARD_ID XXXX
    - #define CS_PART_NUMBER[0] <part number in excel>
    - #define CS_PART_NUMBER[1] <part number in excel>
    - #define CS_PART_NUMBER[2] <part number in excel>
  - Use ./makeMtk [project_name] emigen to generate emi files.
- **Remind: Must run ETT procedure**

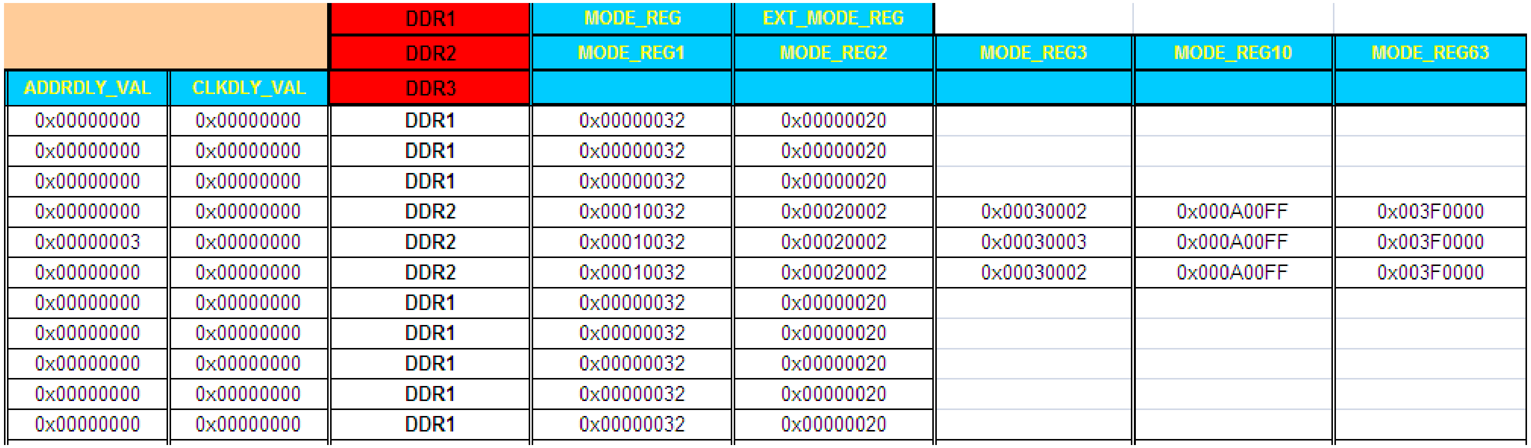
What is Combo MCP Feature

- Collect EMI settings of specified MCP devices into code base when compile time.
- Select correct EMI settings of one MCP device by detected NAND/eMMC ID in run time.
- NAND/eMMC ID should be unique between specified MCP devices.
- User can change MCP device without re-compiling / downloading pre-loader image if required MCP devices have already been specified in configure files.

MemoryDeviceList.xls example



Vendor	Part Number	Type	Density (Mb)	Board ID	NAND/eMMC ID	Nand Page Size (B)		
							CONI_VAL	DRVCTL0_VAL
Hynix	H9DA4GH4JJAMCR_4EM	MCP(NAND+DDR1)	2048+2048	MT6575_EVB		2048	0x0002202E	0x88008800
Micron	MT29C4G96MAZAPCJA-5IT	MCP(NAND+DDR1)	2048+2048	MT6575_EVB	0x2CBC905556	2048	0x0002202E	0x88008800
Samsung	KA100O015E	MCP(NAND+DDR1)	2048+2048	MT6575_EVB	0xECBC006656	2048	0x0002202E	0x88008800
Hynix	H9TP33A8LDMCMR	MCP(eMMC+DDR2)	2048+2048	MT6575_EVB	0x90014A48594E495820	?	0x0002211A	0x88008800
Micron	MT29PZZZ8D4RKKEQ-25	MCP(eMMC+DDR2)	2048+2048	MT6575_EVB	0x13010000526576 2E52	?	0x0002211A	0x88008800
Samsung	KMKL000UM-B406	MCP(eMMC+DDR2)	4096+4096	MT6575_EVB	0x1501004B4C4C30304D	?	0x0002212E	0x88008800
Hynix	H9DA4GH4JJAMCR_4EM	MCP(NAND+DDR1)	2048+2048	BIRD75V1		2048	0x0002202E	0x88008800
Hynix	H9DA4GH4JJAMCR_4EM	MCP(NAND+DDR1)	2048+2048	BIRD75V1_2		2048	0x0002202E	0x88008800
Hynix	H9DA4GH4JJAMCR_4EM	MCP(NAND+DDR1)	2048+2048	EAGLE75V1		2048	0x0002202E	0x88008800
Hynix	H9DA4GH4JJAMCR_4EM	MCP(NAND+DDR1)	2048+2048	EAGLE75V1_2		2048	0x0002202E	0x88008800
Hynix	H9DA4GH4JJAMCR_4EM	MCP(NAND+DDR1)	2048+2048	LENOVO75		2048	0x0002202E	0x88008800



		DDR1	MODE_REG	EXT_MODE_REG			
		DDR2	MODE_REG1	MODE_REG2	MODE_REG3	MODE_REG10	MODE_REG63
ADDRDLY_VAL	CLKDLY_VAL	DDR3					
0x00000000	0x00000000	DDR1	0x00000032	0x00000020			
0x00000000	0x00000000	DDR1	0x00000032	0x00000020			
0x00000000	0x00000000	DDR1	0x00000032	0x00000020			
0x00000000	0x00000000	DDR2	0x00010032	0x00020002	0x00030002	0x000A00FF	0x003F0000
0x00000003	0x00000000	DDR2	0x00010032	0x00020002	0x00030003	0x000A00FF	0x003F0000
0x00000000	0x00000000	DDR2	0x00010032	0x00020002	0x00030002	0x000A00FF	0x003F0000
0x00000000	0x00000000	DDR1	0x00000032	0x00000020			
0x00000000	0x00000000	DDR1	0x00000032	0x00000020			
0x00000000	0x00000000	DDR1	0x00000032	0x00000020			
0x00000000	0x00000000	DDR1	0x00000032	0x00000020			
0x00000000	0x00000000	DDR1	0x00000032	0x00000020			

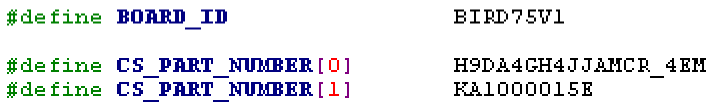
EMI Customization-with combo mcp

- Change file list

File	Description
alps/mediatek/custom/${BOARD}/preloader/inc/	
Custom_MemoryDevice.h	The customization file for EMI setting

- Customization item
  - custom_MemoryDevice.h

```
#define BOARD_ID                        XXXX

#define CS_PART_NUMBER [0]              PART_NUMBER0
#define CS_PART_NUMBER [1]              PART_NUMBER1
#define CS_PART_NUMBER [2]              PART_NUMBER2
#define CS_PART_NUMBER [3]              PART_NUMBER3
…
```



```
#define BOARD_ID                BIRD75V1

#define CS_PART_NUMBER[0]       H9DA4GH4JJAMCR_4EM
#define CS_PART_NUMBER[1]       KA1000015E
```

  - Use ./makeMtk [project] emigen to generate emi files.

- **Remind: Must run ETT procedure**

EMI Customization-Limitation

- NAND/eMMC ID must be unique, not support MCPs that NAND/eMMC IDs are the same but MCP part numbers are different.
- Combo MCP cannot involve eMMC MCP and NAND MCP together.
- If mounted memory device is not MCP (=> discrete)
  - It means DRAM part number must be specified, not support the auto-detect feature by NAND ID.
  - ID of corresponding record in MemoryDeviceList.xls must be empty.
  - There must be only one set of part number defined in custom_MemoryDevice.h to select EMI settings.
- Pre-loader size limitation
  - Pre-loader size will be checked in compile time.

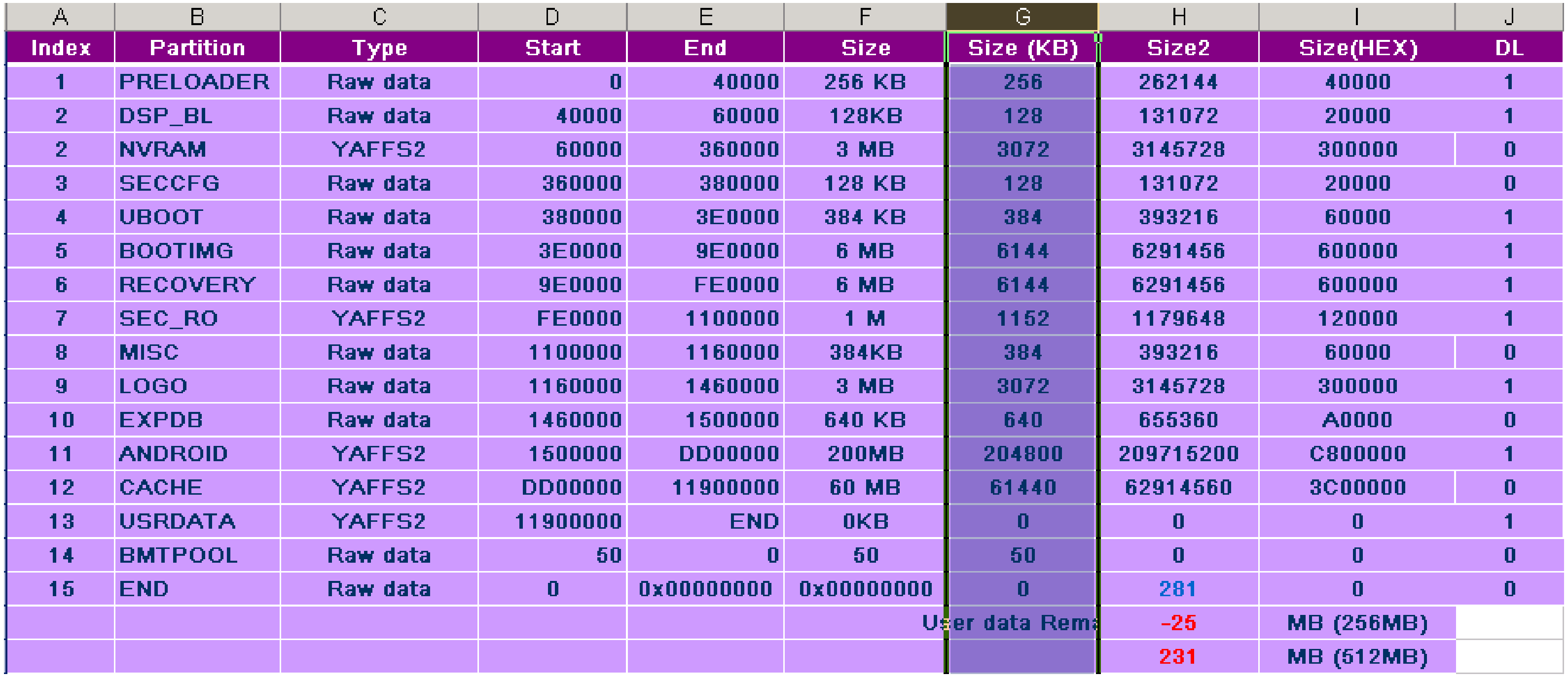
Confidential A

MEDIATEK

NAND Customization

NAND Partition Layout Customizations

- 修改alps/mediatek/build/tools/ptgen/partition_table.xls文件中的G列。



A	B	C	D	E	F	G	H	I	J
Index	Partition	Type	Start	End	Size	Size (KB)	Size2	Size(HEX)	DL
1	PRELOADER	Raw data	0	40000	256 KB	256	262144	40000	1
2	DSP_BL	Raw data	40000	60000	128KB	128	131072	20000	1
2	NVRAM	YAFFS2	60000	360000	3 MB	3072	3145728	300000	0
3	SECCFG	Raw data	360000	380000	128 KB	128	131072	20000	0
4	UBOOT	Raw data	380000	3E0000	384 KB	384	393216	60000	1
5	BOOTIMG	Raw data	3E0000	9E0000	6 MB	6144	6291456	600000	1
6	RECOVERY	Raw data	9E0000	FE0000	6 MB	6144	6291456	600000	1
7	SEC_RO	YAFFS2	FE0000	1100000	1 M	1152	1179648	120000	1
8	MISC	Raw data	1100000	1160000	384KB	384	393216	60000	0
9	LOGO	Raw data	1160000	1460000	3 MB	3072	3145728	300000	1
10	EXPDB	Raw data	1460000	1500000	640 KB	640	655360	A0000	0
11	ANDROID	YAFFS2	1500000	DD00000	200MB	204800	209715200	C800000	1
12	CACHE	YAFFS2	DD00000	11900000	60 MB	61440	62914560	3C00000	0
13	USRDATA	YAFFS2	11900000	END	0KB	0	0	0	1
14	BMTPOOL	Raw data	50	0	50	50	0	0	0
15	END	Raw data	0	0x00000000	0x00000000	0	281	0	0
					User data Rem		-25	MB (256MB)	
							231	MB (512MB)	

- 使用command： ./mk ptgen会自动生成出scatfile和partation table

NAND Partition Layout

- **Pre-loader**
  - Pre-loader image
  - Handles all the download and secure boot procedures
- **DSP_BL**
  - DSP boot loader
- **U-boot**
  - Second loader image
  - Handles most hardware initializations and bring-up entire Linux kernel
- **Boot**
  - Linux kernel image and it’s root file system
- **Recovery**
  - Recovery kernel image and it’s root file system
  - Handles all the system recovery and firmware update functionalities
- **System (Android)**
  - Android system image
- **Logo**
  - Boot-up logo image

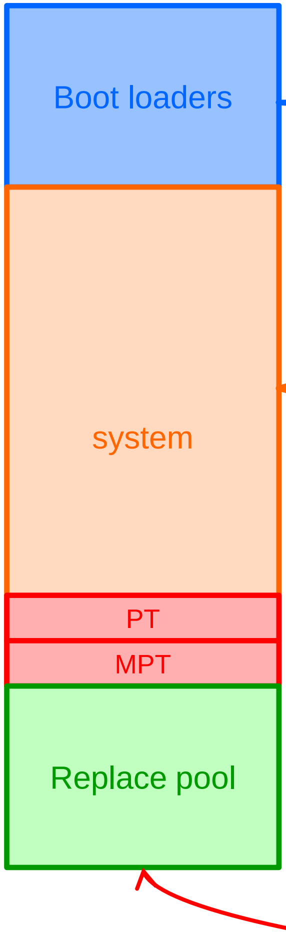
Index	Partition	Type
1	PRELOADER	RAW
2	DSP_BL	RAW
3	NVRAM	RAW
4	SECCFG	RAW
5	UBOOT	RAW
6	BOOTIMG	RAW
7	RECOVERY	RAW
8	SEC_RO	YAFF2
9	MISC	RAW
10	LOGO	RAW
11	EXPDB	RAW
12	ANDROID	YAFFS2
13	CACHE	YAFFS2
14	USRDATA	YAFFS2

NAND Partition Layout (cont.)

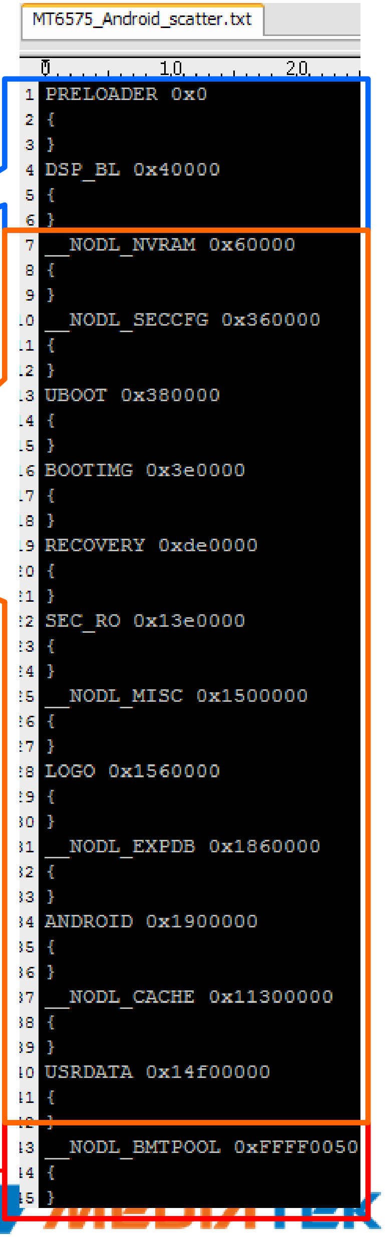
- **NVRAM**
  - Stores the hardware related information, such as calibration data, MAC address, IMEI … etc.
- **Cache**
  - For Android internal used
  - Store Android internal cache data or web cache data
- **Misc**
  - Used for the recovery procedure (power loss)
- **User**
  - Used for Android system to store user data such as user contacts, settings, installed applications … etc.
- **SECCFG and SEC_RO**
  - Reserved for the security platform used
- **EXPDB**
  - Used to store the kernel panic debug messages

Index	Partition	Type
1	PRELOADER	RAW
2	DSP_BL	RAW
3	NVRAM	RAW
4	SECCFG	RAW
5	UBOOT	RAW
6	BOOT	RAW
7	RECOVERY	RAW
8	SEC_RO	YAFF2
9	MISC	RAW
10	LOGO	RAW
11	EXPDB	RAW
12	ANDROID	YAFFS2
13	CACHE	YAFFS2
14	USRDATA	YAFFS2

NAND Device Layout



Index	Partition	Type
1	PRELOADER	RAW
2	DSP_BL	RAW
3	NVRAM	RAW
4	SECCFG	RAW
5	UBOOT	RAW
6	BOOT	RAW
7	RECOVERY	RAW
8	SEC_RO	YAFF2
9	MISC	RAW
10	LOGO	RAW
11	EXPDB	RAW
12	ANDROID	YAFFS2
13	CACHE	YAFFS2
14	USRDATA	YAFFS2

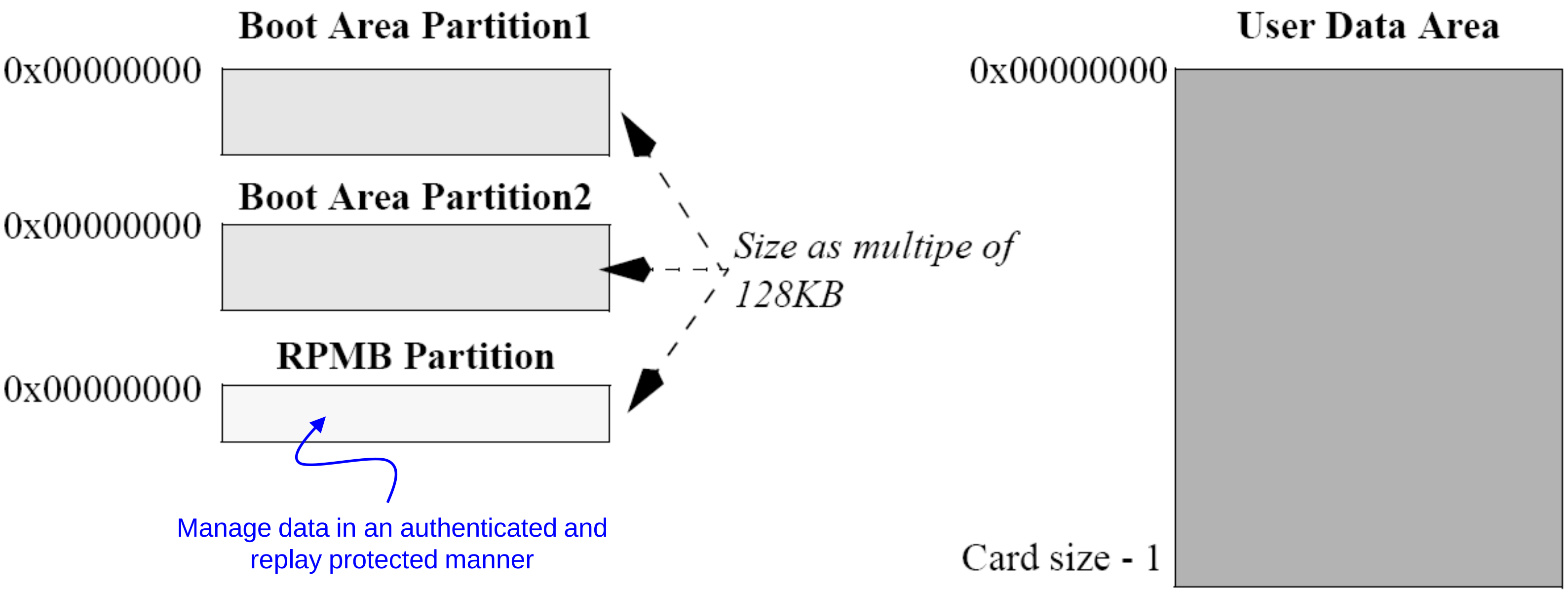


```
PRELOADER 0x0
{
}
DSP_BL 0x40000
{
}
__NODL_NVRAM 0x60000
{
}
__NODL_SECCFG 0x360000
{
}
UBOOT 0x380000
{
}
BOOTIMG 0x3e0000
{
}
RECOVERY 0xde0000
{
}
SEC_RO 0x13e0000
{
}
__NODL_MISC 0x1500000
{
}
LOGO 0x1560000
{
}
__NODL_EXPDB 0x1860000
{
}
ANDROID 0x1900000
{
}
__NODL_CACHE 0x11300000
{
}
USRDATA 0x14f00000
{
}
__NODL_BMTPOOL 0xFFFF0050
{
}
```

The size of replace pool is decided in scatter file

eMMC Partition Management

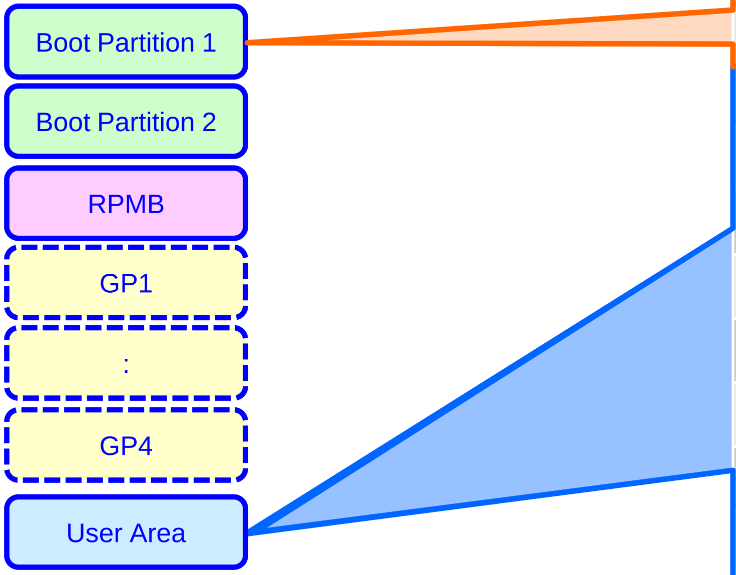
- **4 Default Areas of Memory Device**
  - 2 x Boot Area Partitions for Booting
  - 1 x Replay Protected Memory Block Area Partition
  - 1 x User Data Area



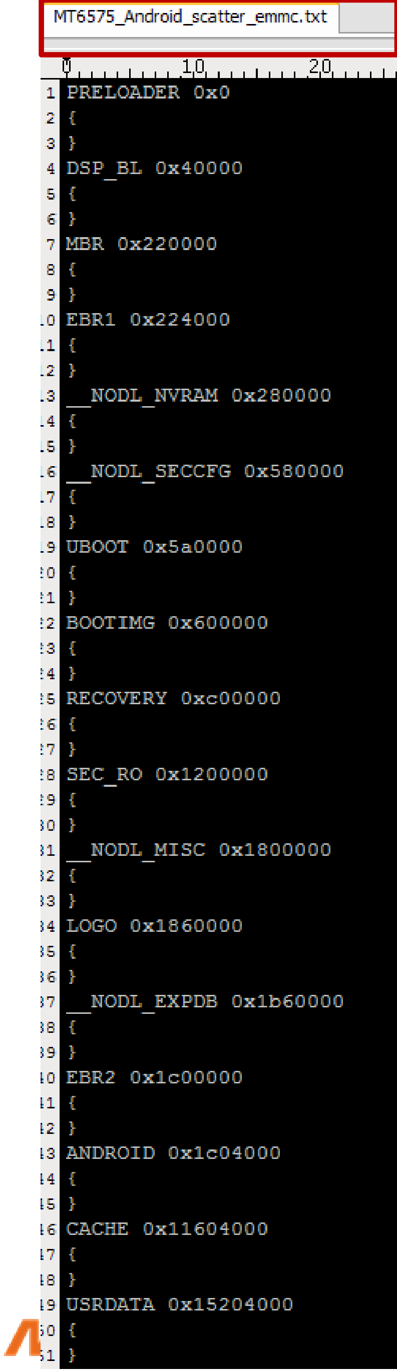
Manage data in an authenticated and replay protected manner

eMMC Device Layout (cont.)

- Scatter File Mapping



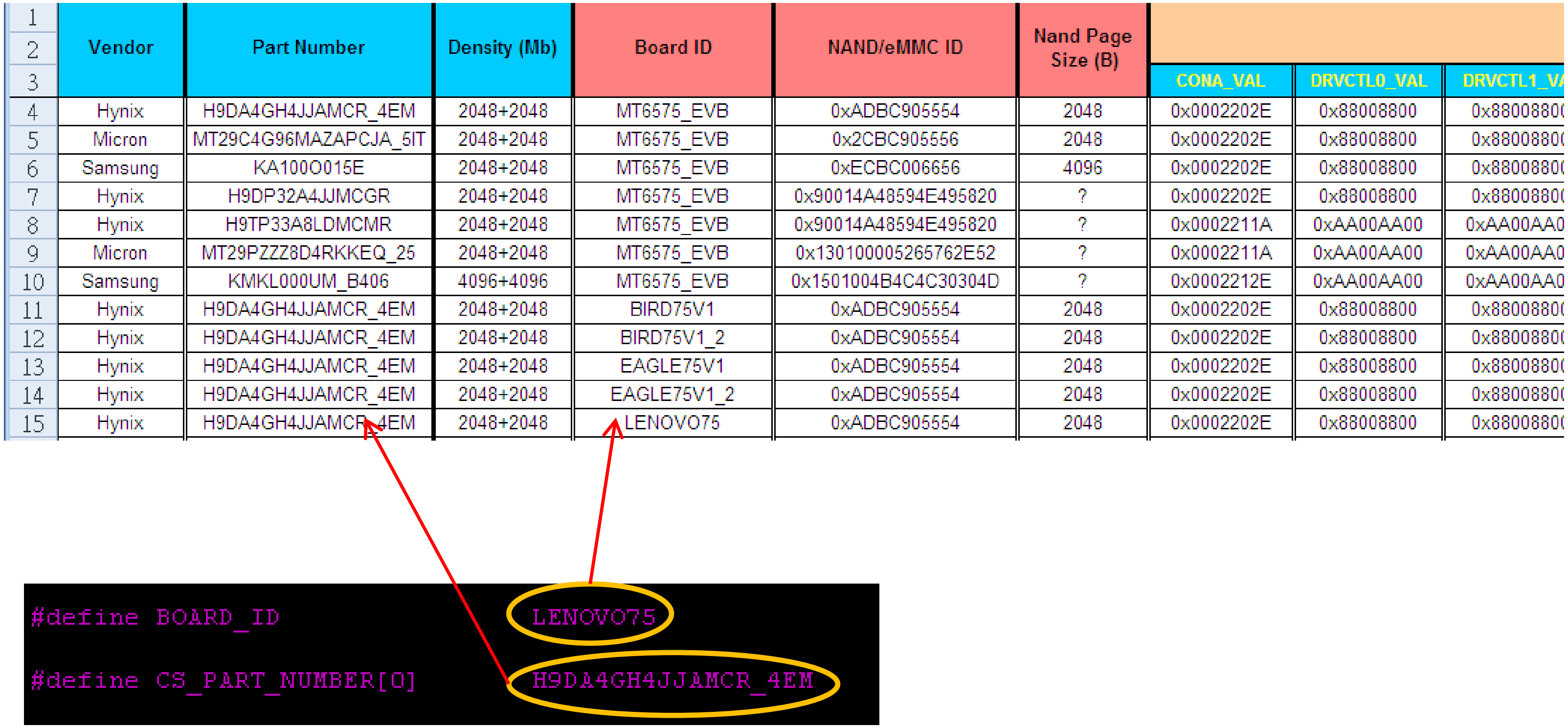
Index	Partition	Type
1	PRELOADER	RAW
2	DSP_BL	RAW
3	MBR	RAW
4	EBR1	RAW
5	NVRAM	RAW
6	SECCFG	RAW
7	UBOOT	RAW
8	BOOTIMG	RAW
9	RECOVERY	RAW
10	SECRO	RAW
11	MISC	RAW
12	LOGO	RAW
13	EXPDB	RAW
14	EBR2	RAW
15	ANDROID	EXT4
16	CACHE	EXT4
17	USRDATA	EXT4



```
PRELOADER 0x0
{
}
DSP_BL 0x40000
{
}
MBR 0x220000
{
}
EBR1 0x224000
{
}
__NODL_NVRAM 0x280000
{
}
__NODL_SECCFG 0x580000
{
}
UBOOT 0x5a0000
{
}
BOOTIMG 0x600000
{
}
RECOVERY 0xc00000
{
}
SEC_RO 0x1200000
{
}
__NODL_MISC 0x1800000
{
}
LOGO 0x1860000
{
}
__NODL_EXPDB 0x1b60000
{
}
EBR2 0x1c00000
{
}
ANDROID 0x1c04000
{
}
CACHE 0x11604000
{
}
USRDATA 0x15204000
{
}
```

select dram part number

mediatek/build/tools/emigen/MT6575/MemoryDeviceList_MT6575.xls



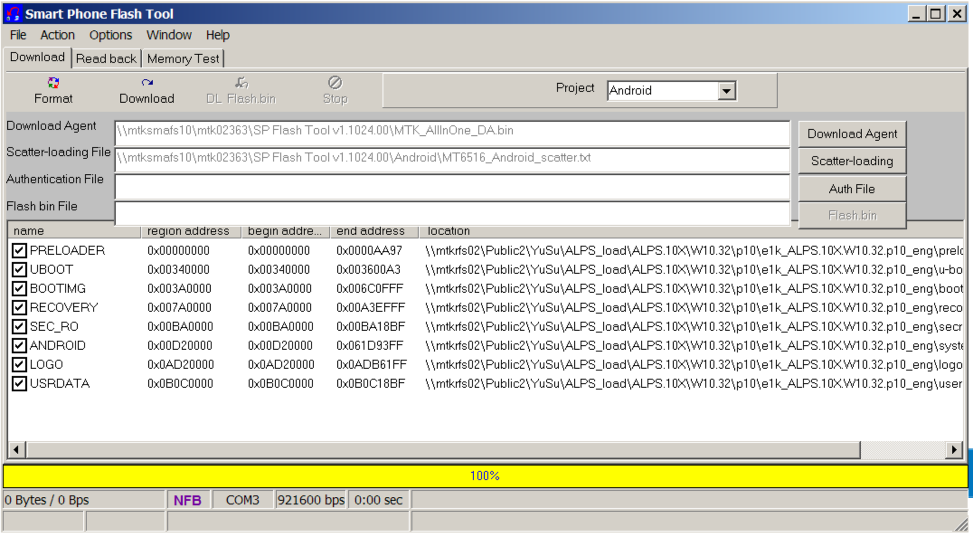
	Vendor	Part Number	Density (Mb)	Board ID	NAND/eMMC ID	Nand Page Size (B)			
1									
2									
3							CONA_VAL	DRVCTL0_VAL	DRVCTL1_VA
4	Hynix	H9DA4GH4JJAMCR_4EM	2048+2048	MT6575_EVB	0xADBC905554	2048	0x0002202E	0x88008800	0x8800880
5	Micron	MT29C4G96MAZAPCJA_5IT	2048+2048	MT6575_EVB	0x2CBC905556	2048	0x0002202E	0x88008800	0x8800880
6	Samsung	KA100O015E	2048+2048	MT6575_EVB	0xECBC006656	4096	0x0002202E	0x88008800	0x8800880
7	Hynix	H9DP32A4JJMCGR	2048+2048	MT6575_EVB	0x90014A48594E495820	?	0x0002202E	0x88008800	0x8800880
8	Hynix	H9TP33A8LDMCMR	2048+2048	MT6575_EVB	0x90014A48594E495820	?	0x0002211A	0xAA00AA00	0xAA00AA0
9	Micron	MT29PZZZ8D4RKKEQ_25	2048+2048	MT6575_EVB	0x130100005265762E52	?	0x0002211A	0xAA00AA00	0xAA00AA0
10	Samsung	KMKL000UM_B406	4096+4096	MT6575_EVB	0x1501004B4C4C30304D	?	0x0002212E	0xAA00AA00	0xAA00AA0
11	Hynix	H9DA4GH4JJAMCR_4EM	2048+2048	BIRD75V1	0xADBC905554	2048	0x0002202E	0x88008800	0x8800880
12	Hynix	H9DA4GH4JJAMCR_4EM	2048+2048	BIRD75V1_2	0xADBC905554	2048	0x0002202E	0x88008800	0x8800880
13	Hynix	H9DA4GH4JJAMCR_4EM	2048+2048	EAGLE75V1	0xADBC905554	2048	0x0002202E	0x88008800	0x8800880
14	Hynix	H9DA4GH4JJAMCR_4EM	2048+2048	EAGLE75V1_2	0xADBC905554	2048	0x0002202E	0x88008800	0x8800880
15	Hynix	H9DA4GH4JJAMCR_4EM	2048+2048	LENOVO75	0xADBC905554	2048	0x0002202E	0x88008800	0x8800880

```
#define BOARD_ID                LENOVO75

#define CS_PART_NUMBER[0]       H9DA4GH4JJAMCR_4EM
```

mediatek/custom/*PROJECT*/preloader/inc/custom_MemoryDevice.h

Software Package Download

- **Download Agent**
  - The agent on target to perform the download procedure upon tool request
- **Scatter-Loading File**
  - Describe the start address of each partition to download to
  - The storage type & chip is embedded into scatter file name
    - Tool will check if chip name matches devices while handshake
    - For downloading NAND images
      - MT6575_Android_scatter.txt
    - For downloading eMMC images
      - MT6575_Android_scatter_emmc.txt



Project: Android

Download Agent: \\mtksmafs10\mtk02363\SP Flash Tool v1.1024.00\MTK_AllInOne_DA.bin

Scatter-loading File: \\mtksmafs10\mtk02363\SP Flash Tool v1.1024.00\Android\MT6516_Android_scatter.txt

Authentication File:

Flash bin File:

name	region address	begin addre...	end address	location
☑ PRELOADER	0x00000000	0x00000000	0x0000AA97	\\mtkrfs02\Public2\YuSu\ALPS_load\ALPS.10X\W10.32\p10\e1k_ALPS.10X.W10.32.p10_eng\prelc
☑ UBOOT	0x00340000	0x00340000	0x003600A3	\\mtkrfs02\Public2\YuSu\ALPS_load\ALPS.10X\W10.32\p10\e1k_ALPS.10X.W10.32.p10_eng\u-bo
☑ BOOTIMG	0x003A0000	0x003A0000	0x006C0FFF	\\mtkrfs02\Public2\YuSu\ALPS_load\ALPS.10X\W10.32\p10\e1k_ALPS.10X.W10.32.p10_eng\boot
☑ RECOVERY	0x007A0000	0x007A0000	0x00A3EFFF	\\mtkrfs02\Public2\YuSu\ALPS_load\ALPS.10X\W10.32\p10\e1k_ALPS.10X.W10.32.p10_eng\reco
☑ SEC_RO	0x00BA0000	0x00BA0000	0x00BA18BF	\\mtkrfs02\Public2\YuSu\ALPS_load\ALPS.10X\W10.32\p10\e1k_ALPS.10X.W10.32.p10_eng\secr
☑ ANDROID	0x00D20000	0x00D20000	0x061D93FF	\\mtkrfs02\Public2\YuSu\ALPS_load\ALPS.10X\W10.32\p10\e1k_ALPS.10X.W10.32.p10_eng\syste
☑ LOGO	0x0AD20000	0x0AD20000	0x0ADB61FF	\\mtkrfs02\Public2\YuSu\ALPS_load\ALPS.10X\W10.32\p10\e1k_ALPS.10X.W10.32.p10_eng\logo
☑ USRDATA	0x0B0C0000	0x0B0C0000	0x0B0C18BF	\\mtkrfs02\Public2\YuSu\ALPS_load\ALPS.10X\W10.32\p10\e1k_ALPS.10X.W10.32.p10_eng\user

100%

0 Bytes / 0 Bps | NFB | COM3 | 921600 bps | 0:00 sec

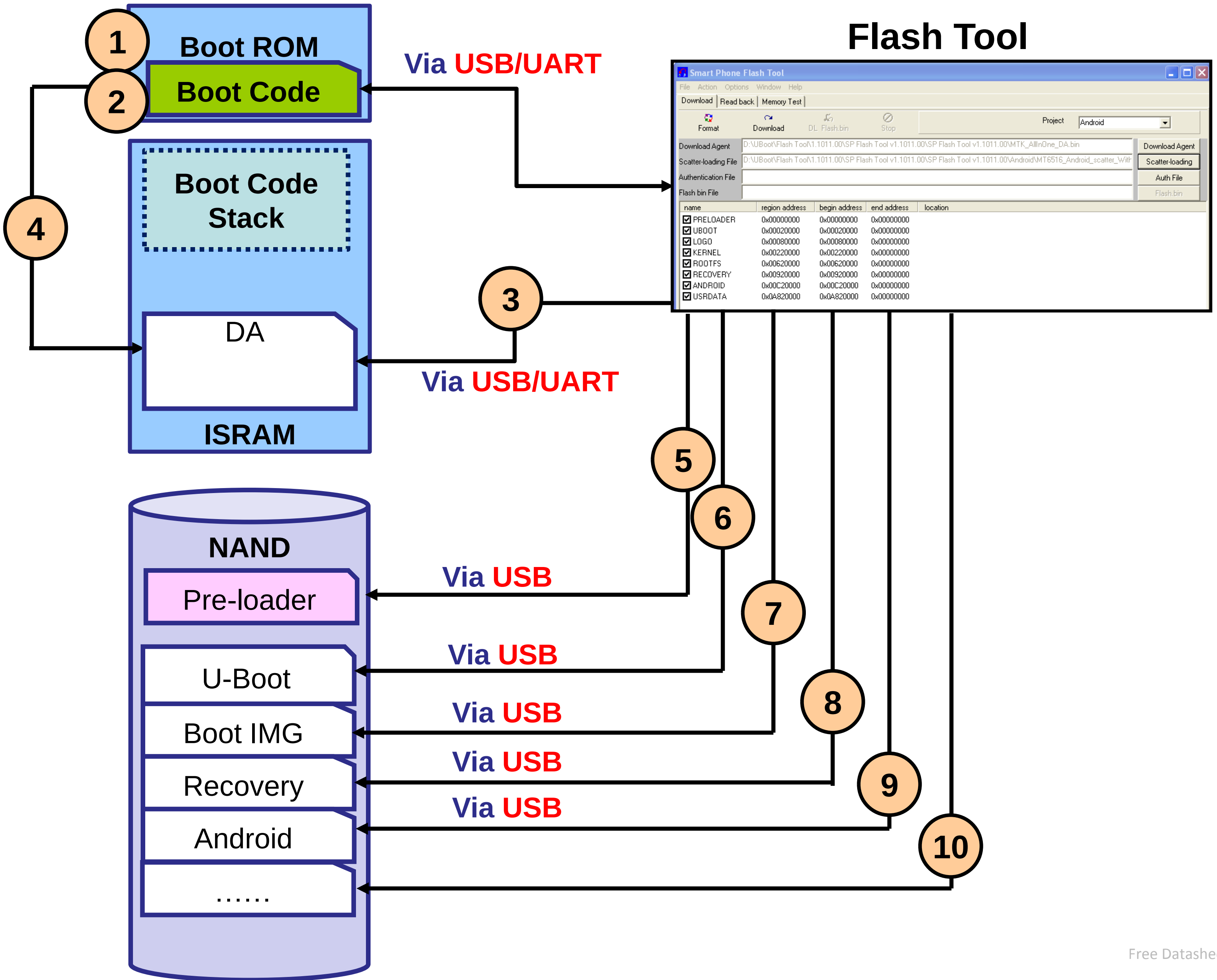
Confidential A

Download and Boot Up

Bootloader Overview

- **Bootloader contains Pre-loader (Initial program Loader) U-Boot (Secondary Loader)**
  - **Pre-loader (MTK in-house developed loader)**
    - takes charge of all the platform dependency work (including initializing EMI / PLL ..).
  - **U-Boot (GPL licensed loader)**
    - prepares the Linux compatible environment (e.g. Linux Kernel Parameter) before entering Linux Kernel.

System Download Process



1 Boot ROM

2 Boot Code

Via USB/UART

Flash Tool

4

Boot Code Stack

3

DA

Via USB/UART

ISRAM

5

6

NAND

Pre-loader

Via USB

U-Boot

Via USB

7

Boot IMG

Via USB

8

Recovery

Via USB

9

Android

Via USB

10

……

Project: Android

Download Agent: D:\UBoot\Flash Tool\1.1011.00\SP Flash Tool v1.1011.00\SP Flash Tool v1.1011.00\MTK_AllInOne_DA.bin

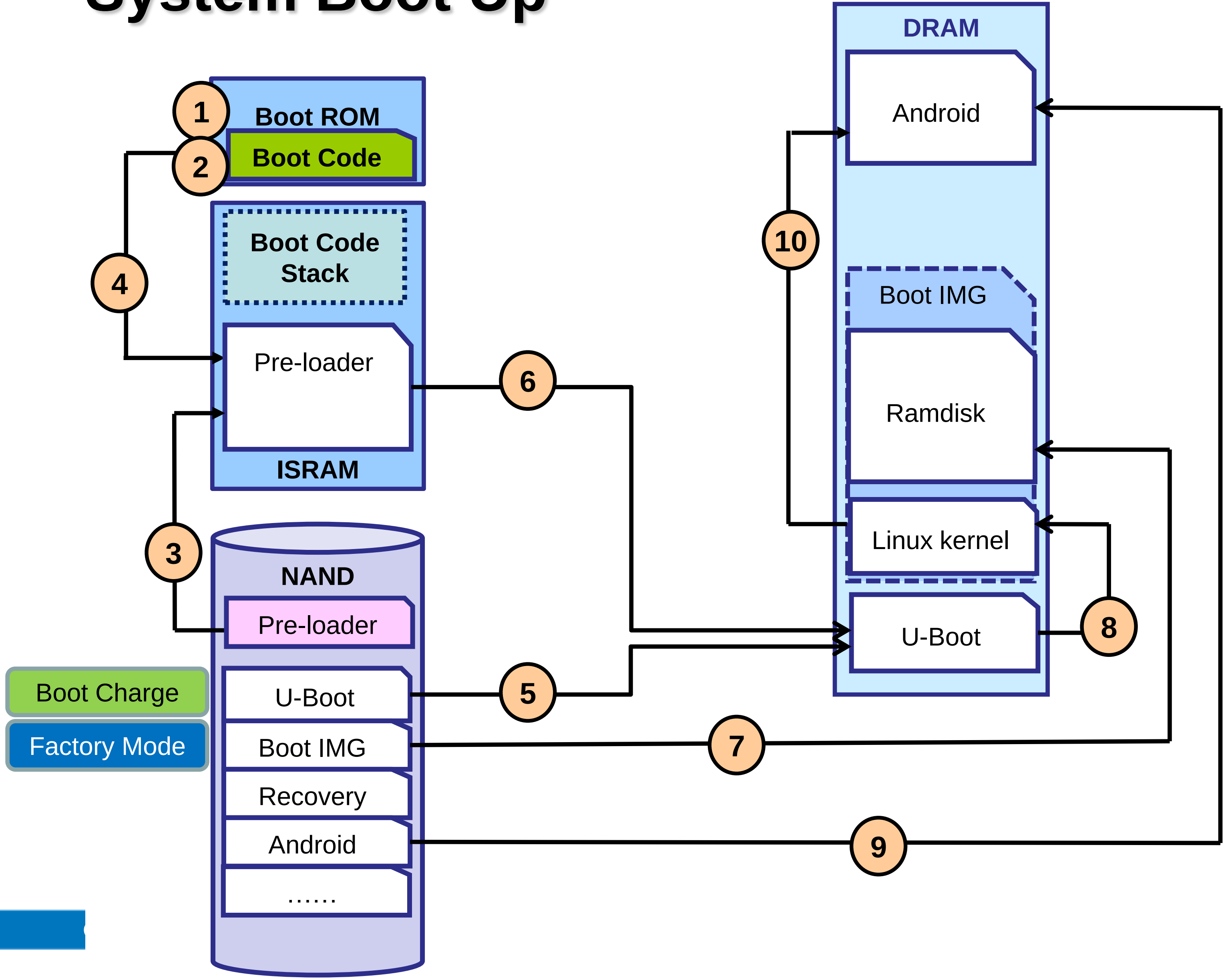
Scatter-loading File: D:\UBoot\Flash Tool\1.1011.00\SP Flash Tool v1.1011.00\SP Flash Tool v1.1011.00\Android\MT6516_Android_scatter_With

Authentication File:

Flash bin File:

name	region address	begin address	end address	location
☑ PRELOADER	0x00000000	0x00000000	0x00000000	
☑ UBOOT	0x00020000	0x00020000	0x00000000	
☑ LOGO	0x00080000	0x00080000	0x00000000	
☑ KERNEL	0x00220000	0x00220000	0x00000000	
☑ ROOTFS	0x00620000	0x00620000	0x00000000	
☑ RECOVERY	0x00920000	0x00920000	0x00000000	
☑ ANDROID	0x00C20000	0x00C20000	0x00000000	
☑ USRDATA	0x0A820000	0x0A820000	0x00000000	

System Boot Up



MEDIATEK

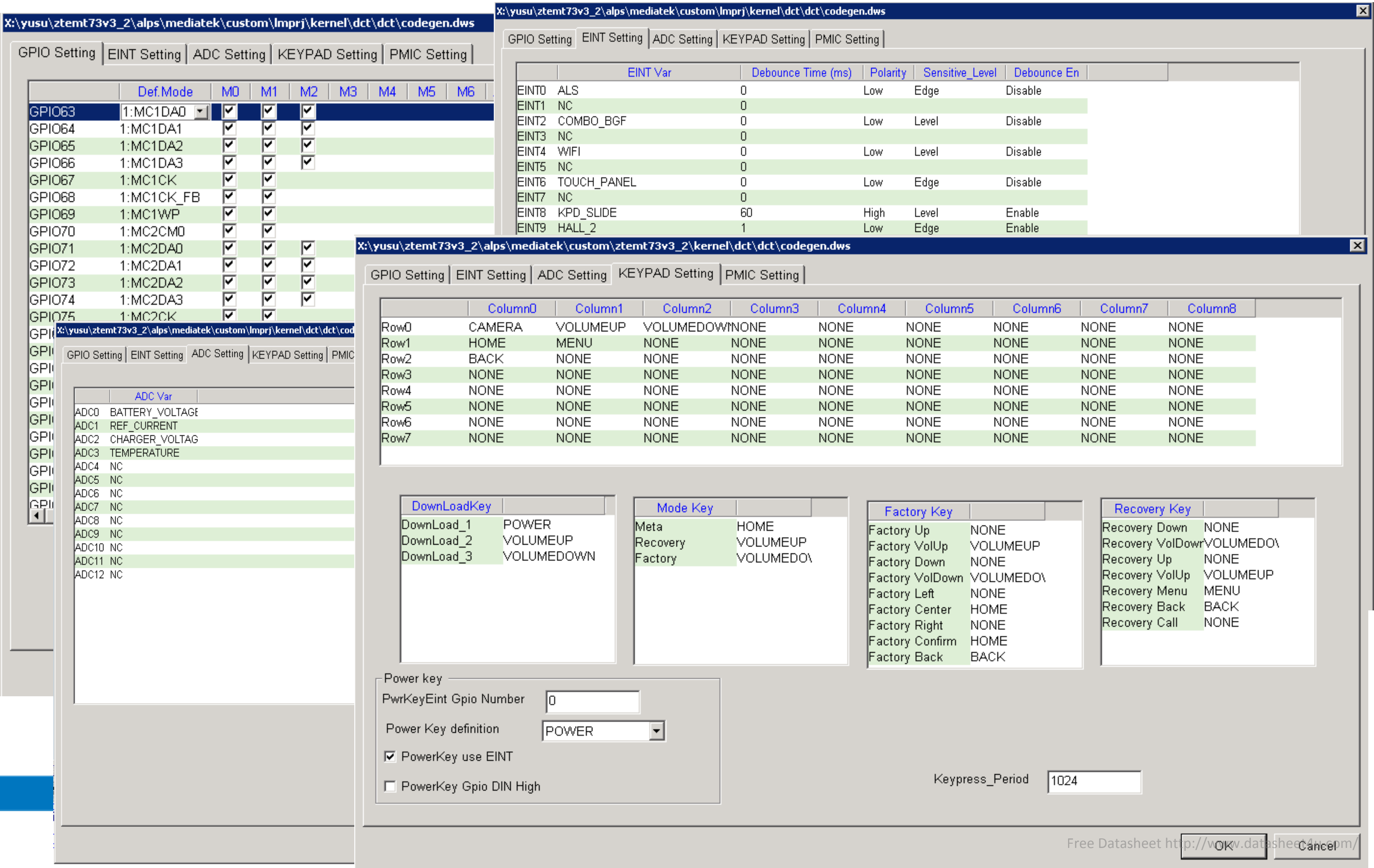
DCT

Outline

- **What is DCT?**
- Why use DCT?
- DCT Customization flow
- GPIO
- ADC
- EINT
- Keypad

What is DCT?

DCT(Device Customization Tool) is a GUI tool to auto-generate Source Code for Device customization, such as GPIO, EINT, Keypad...



X:\yusu\ztemt73v3_2\alps\mediatek\custom\lmprj\kernel\dct\dct\codegen.dws

GPIO Setting | EINT Setting | ADC Setting | KEYPAD Setting | PMIC Setting

	Def.Mode	M0	M1	M2	M3	M4	M5	M6
GPIO63	1:MC1DA0	☑	☑	☑				
GPIO64	1:MC1DA1	☑	☑	☑				
GPIO65	1:MC1DA2	☑	☑	☑				
GPIO66	1:MC1DA3	☑	☑	☑				
GPIO67	1:MC1CK	☑	☑					
GPIO68	1:MC1CK_FB	☑	☑					
GPIO69	1:MC1WP	☑	☑					
GPIO70	1:MC2CM0	☑	☑					
GPIO71	1:MC2DA0	☑	☑	☑				
GPIO72	1:MC2DA1	☑	☑	☑				
GPIO73	1:MC2DA2	☑	☑	☑				
GPIO74	1:MC2DA3	☑	☑	☑				
GPIO75	1:MC2CK	☑	☑					

X:\yusu\ztemt73v3_2\alps\mediatek\custom\lmprj\kernel\dct\dct\codegen.dws

GPIO Setting | EINT Setting | ADC Setting | KEYPAD Setting | PMIC Setting

	EINT Var	Debounce Time (ms)	Polarity	Sensitive_Level	Debounce En
EINT0	ALS	0	Low	Edge	Disable
EINT1	NC	0			
EINT2	COMBO_BGF	0	Low	Level	Disable
EINT3	NC	0			
EINT4	WIFI	0	Low	Level	Disable
EINT5	NC	0			
EINT6	TOUCH_PANEL	0	Low	Edge	Disable
EINT7	NC	0			
EINT8	KPD_SLIDE	60	High	Level	Enable
EINT9	HALL_2	1	Low	Edge	Enable

X:\yusu\ztemt73v3_2\alps\mediatek\custom\lmprj\kernel\dct\dct\cod

GPIO Setting | EINT Setting | ADC Setting | KEYPAD Setting | PMIC

	ADC Var
ADC0	BATTERY_VOLTAGE
ADC1	REF_CURRENT
ADC2	CHARGER_VOLTAG
ADC3	TEMPERATURE
ADC4	NC
ADC5	NC
ADC6	NC
ADC7	NC
ADC8	NC
ADC9	NC
ADC10	NC
ADC11	NC
ADC12	NC

X:\yusu\ztemt73v3_2\alps\mediatek\custom\ztemt73v3_2\kernel\dct\dct\codegen.dws

GPIO Setting | EINT Setting | ADC Setting | KEYPAD Setting | PMIC Setting

	Column0	Column1	Column2	Column3	Column4	Column5	Column6	Column7	Column8
Row0	CAMERA	VOLUMEUP	VOLUMEDOW	NONE	NONE	NONE	NONE	NONE	NONE
Row1	HOME	MENU	NONE	NONE	NONE	NONE	NONE	NONE	NONE
Row2	BACK	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE
Row3	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE
Row4	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE
Row5	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE
Row6	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE
Row7	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE

DownLoadKey	
DownLoad_1	POWER
DownLoad_2	VOLUMEUP
DownLoad_3	VOLUMEDOWN

Mode Key	
Meta	HOME
Recovery	VOLUMEUP
Factory	VOLUMEDO\

Factory Key	
Factory Up	NONE
Factory VolUp	VOLUMEUP
Factory Down	NONE
Factory VolDown	VOLUMEDO\
Factory Left	NONE
Factory Center	HOME
Factory Right	NONE
Factory Confirm	HOME
Factory Back	BACK

Recovery Key	
Recovery Down	NONE
Recovery VolDowr	VOLUMEDO\
Recovery Up	NONE
Recovery VolUp	VOLUMEUP
Recovery Menu	MENU
Recovery Back	BACK
Recovery Call	NONE

Power key

PwrKeyEint Gpio Number: 0

Power Key definition: POWER

☑ PowerKey use EINT

□ PowerKey Gpio DIN High

Keypress_Period: 1024

Outline

- What is DCT?
- **Why use DCT?**
- DCT Customization flow
- GPIO
- ADC
- EINT
- Keypad

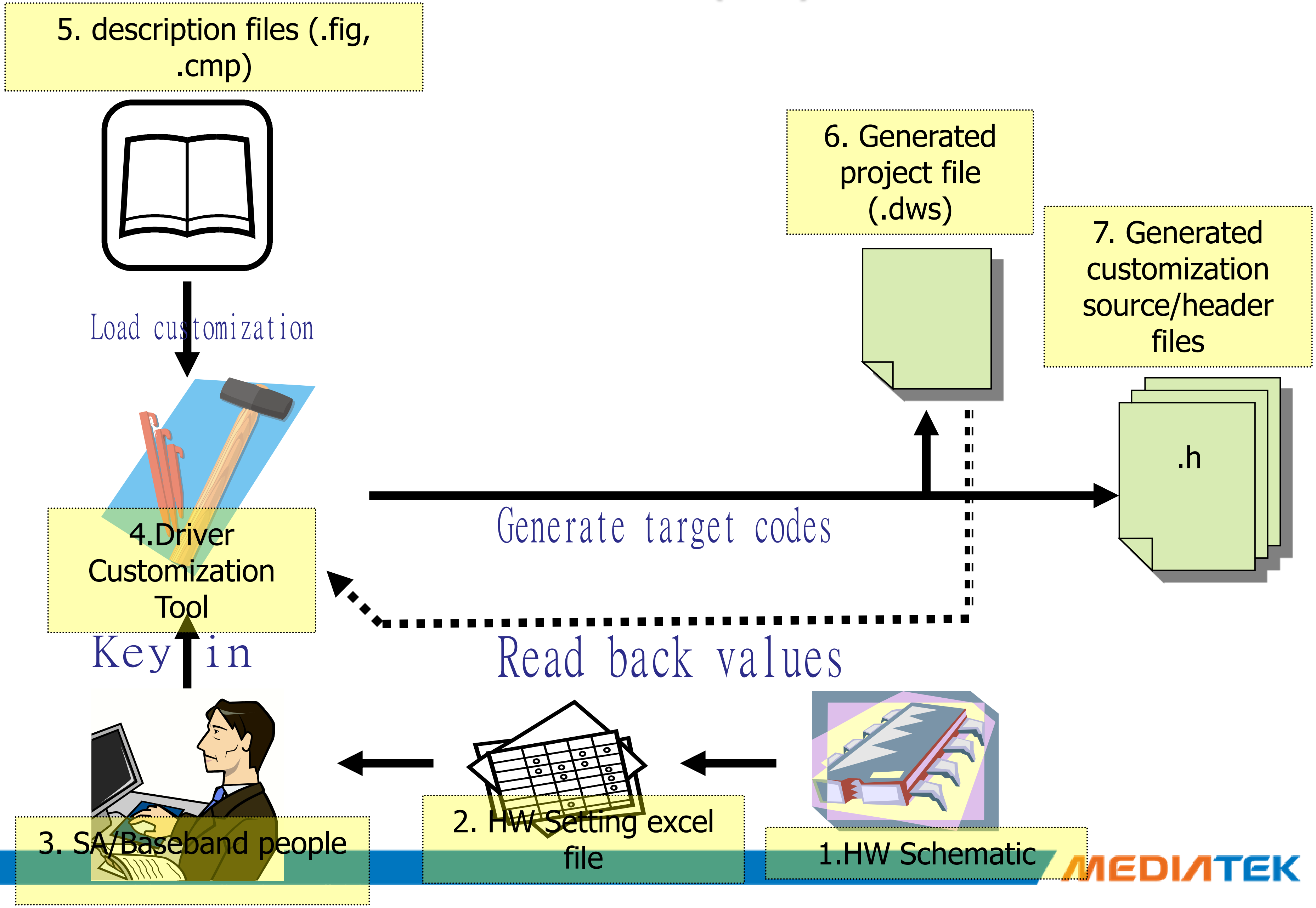
Why use DCT?

- MT6575 has:
  - 231 multi –function GPIO pins.
  - 20 IRQ Pins
  - 12 ADC Pins
  - 60+ keys
- Traditional device customization is trivial & error-prone.
- DCT can help to manage these customization easily, and try best to prevent from breaking rules.

Outline

- What is DCT?
- Why use DCT?
- **DCT Customization flow**
- GPIO
- ADC
- EINT
- Keypad

DCT Customization Flow (1/4)



5. description files (.fig, .cmp)

Load customization

6. Generated project file (.dws)

7. Generated customization source/header files

4.Driver Customization Tool

Generate target codes

Read back values

Key in

3. SA/Baseband people

2. HW Setting excel file

1.HW Schematic

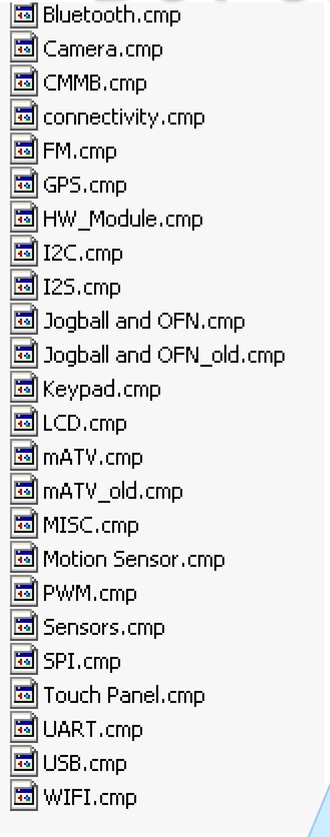
DCT Customization Flow (2/4)

- Chip customization files
  - A customization file (ex.mt6xxx.fig) will describe the hardware customization related to this chip.
  - For example, it will contain
    - GPIO’s pin count, available modes, pull up/down.
    - ADC channel count.
    - External interrupt (EINT) pin count.
    - Keypad scanner matrix size.

DCT Customization Flow (3/4)

- Component description files
  - For each component supported by the tool, for example: GPIO, EINT, ADC, keypad and UEM, there will be a component variable file (xxx.cmp)
  - .cmp file contains variable names

DCT Customization Flow (4/4)



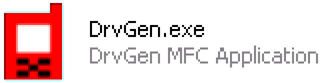
Bluetooth.cmp
Camera.cmp
CMMB.cmp
connectivity.cmp
FM.cmp
GPS.cmp
HW_Module.cmp
I2C.cmp
I2S.cmp
Jogball and OFN.cmp
Jogball and OFN_old.cmp
Keypad.cmp
LCD.cmp
mATV.cmp
mATV_old.cmp
MISC.cmp
Motion Sensor.cmp
PWM.cmp
Sensors.cmp
SPI.cmp
Touch Panel.cmp
UART.cmp
USB.cmp
WIFI.cmp

MT6573_MP.fig

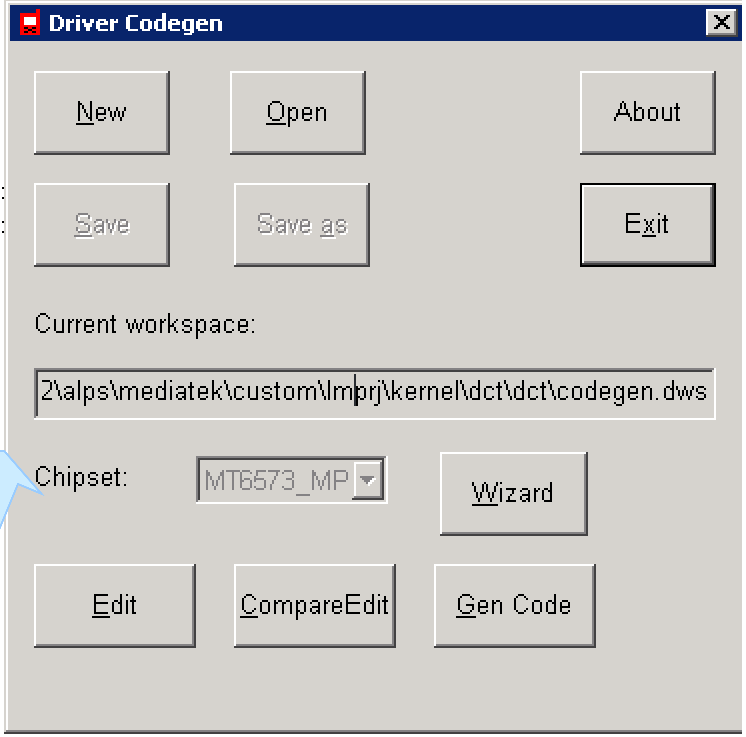
or

MT6575_MP.fig

Load



DrvGen.exe
DrvGen MFC Application



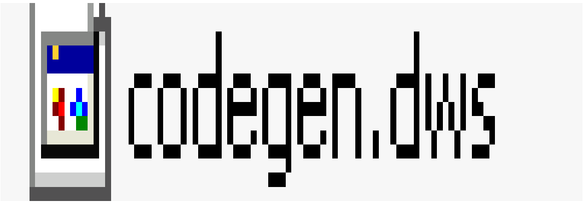
Current workspace:

2\alps\mediatek\custom\lmprj\kernel\dct\dct\codegen.dws

Chipset: MT6573_MP

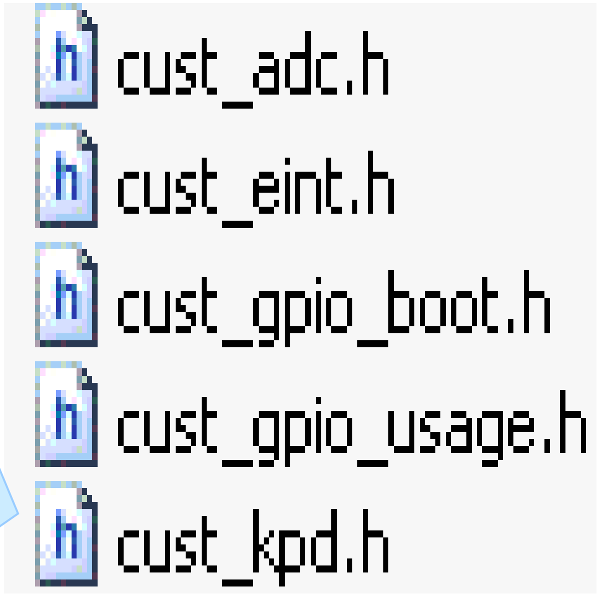
Save

Read



codegen.dws

Gen Code

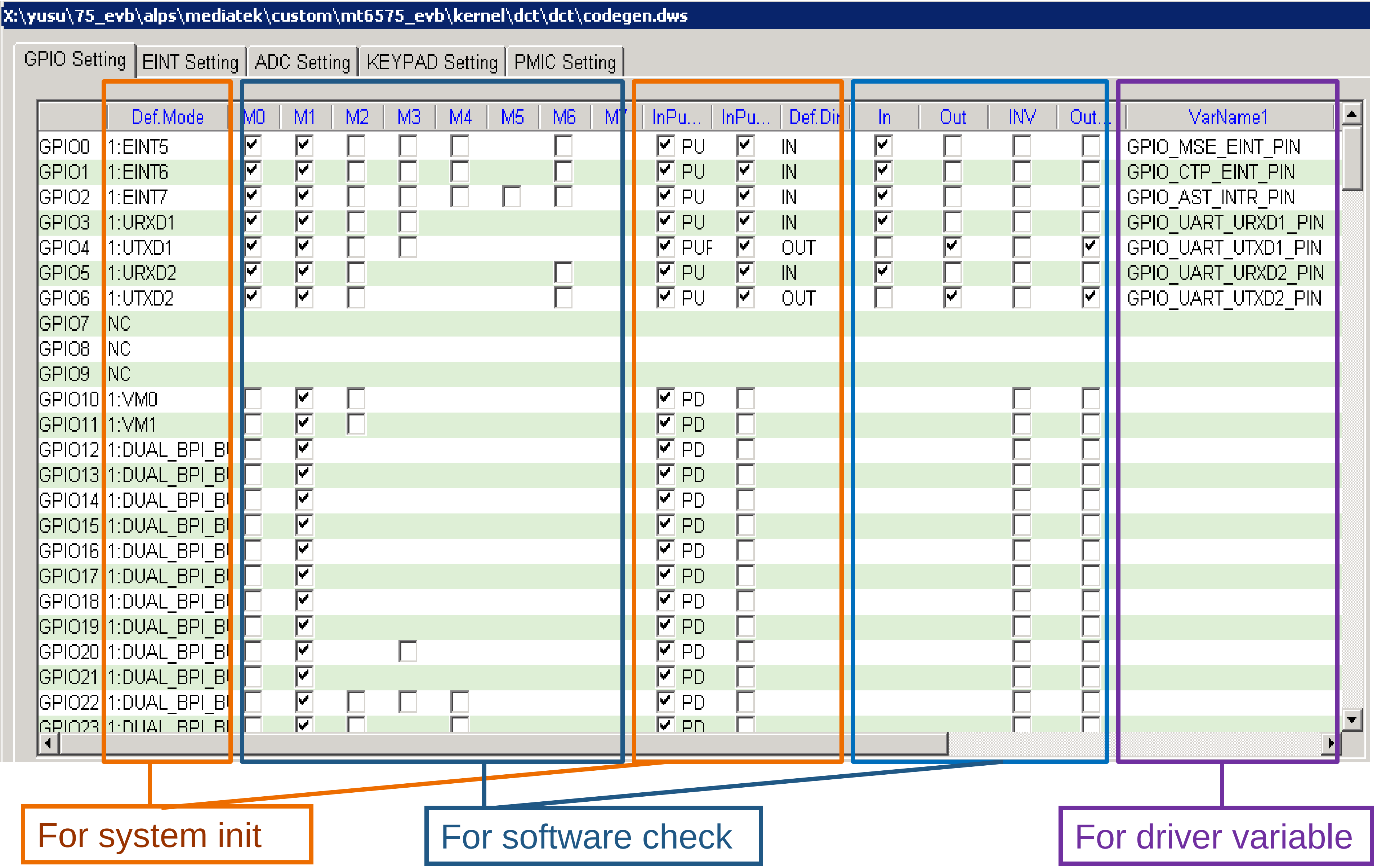


cust_adc.h
cust_eint.h
cust_gpio_boot.h
cust_gpio_usage.h
cust_kpd.h

Outline

- What is DCT?
- Why use DCT?
- DCT Customization flow
- **GPIO**
- ADC
- EINT
- Keypad

GPIO Customize GUI



X:\yusu\75_evb\alps\mediatek\custom\mt6575_evb\kernel\dct\dct\codegen.dws

	Def.Mode	M0	M1	M2	M3	M4	M5	M6	M7	InPu...	InPu...	Def.Dir	In	Out	INV	Out..	VarName1
GPIO0	1:EINT5	☑	☑	□	□	□		□		☑ PU	☑	IN	☑	□	□	□	GPIO_MSE_EINT_PIN
GPIO1	1:EINT6	☑	☑	□	□	□		□		☑ PU	☑	IN	☑	□	□	□	GPIO_CTP_EINT_PIN
GPIO2	1:EINT7	☑	☑	□	□	□	□	□		☑ PU	☑	IN	☑	□	□	□	GPIO_AST_INTR_PIN
GPIO3	1:URXD1	☑	☑	□	□					☑ PU	☑	IN	☑	□	□	□	GPIO_UART_URXD1_PIN
GPIO4	1:UTXD1	☑	☑	□	□					☑ PUF	☑	OUT	□	☑	□	☑	GPIO_UART_UTXD1_PIN
GPIO5	1:URXD2	☑	☑	□				□		☑ PU	☑	IN	☑	□	□	□	GPIO_UART_URXD2_PIN
GPIO6	1:UTXD2	☑	☑	□				□		☑ PU	☑	OUT	□	☑	□	☑	GPIO_UART_UTXD2_PIN
GPIO7	NC																
GPIO8	NC																
GPIO9	NC																
GPIO10	1:VM0	□	☑	□						☑ PD	□				□	□	
GPIO11	1:VM1	□	☑	□						☑ PD	□				□	□	
GPIO12	1:DUAL_BPI_BI	□	☑							☑ PD	□				□	□	
GPIO13	1:DUAL_BPI_BI	□	☑							☑ PD	□				□	□	
GPIO14	1:DUAL_BPI_BI	□	☑							☑ PD	□				□	□	
GPIO15	1:DUAL_BPI_BI	□	☑							☑ PD	□				□	□	
GPIO16	1:DUAL_BPI_BI	□	☑							☑ PD	□				□	□	
GPIO17	1:DUAL_BPI_BI	□	☑							☑ PD	□				□	□	
GPIO18	1:DUAL_BPI_BI	□	☑							☑ PD	□				□	□	
GPIO19	1:DUAL_BPI_BI	□	☑							☑ PD	□				□	□	
GPIO20	1:DUAL_BPI_BI	□	☑		□					☑ PD	□				□	□	
GPIO21	1:DUAL_BPI_BI	□	☑							☑ PD	□				□	□	
GPIO22	1:DUAL_BPI_BI	□	☑	□	□	□				☑ PD	□				□	□	
GPIO23	1:DUAL_BPI_BI	□	☑	□		□				☑ PD	□				□	□	

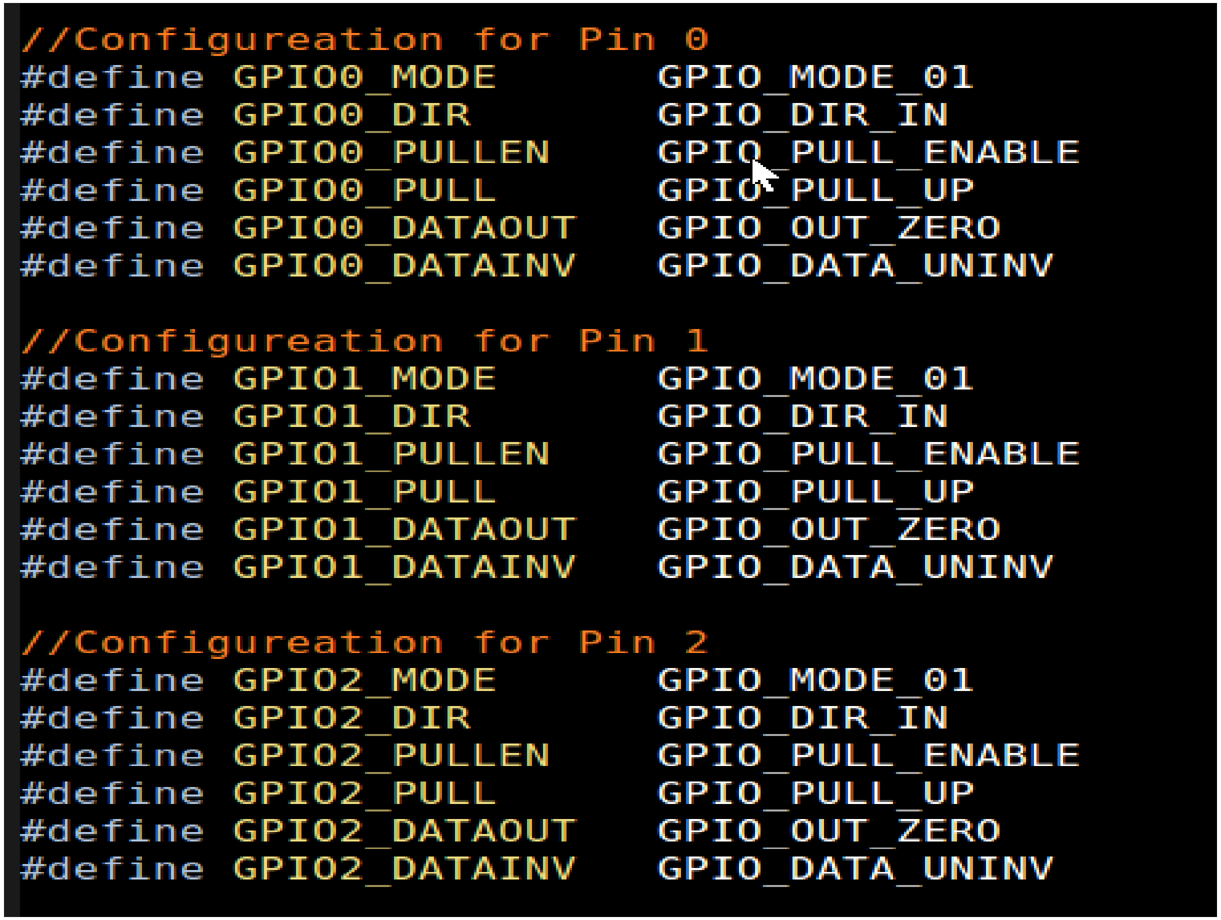
For system init

For software check

For driver variable

Generate code

System init: cust_gpio_boot.h

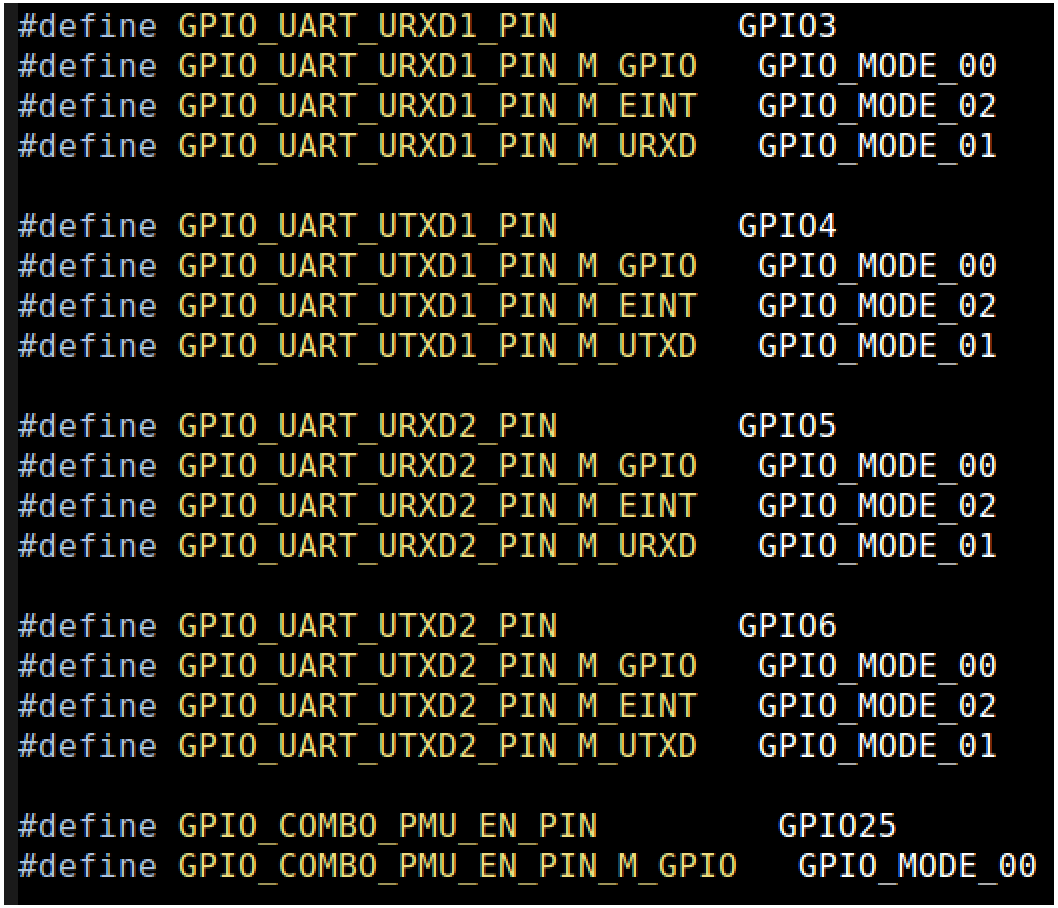


```
//Configureation for Pin 0
#define GPIO0_MODE          GPIO_MODE_01
#define GPIO0_DIR           GPIO_DIR_IN
#define GPIO0_PULLEN        GPIO_PULL_ENABLE
#define GPIO0_PULL          GPIO_PULL_UP
#define GPIO0_DATAOUT       GPIO_OUT_ZERO
#define GPIO0_DATAINV       GPIO_DATA_UNINV

//Configureation for Pin 1
#define GPIO1_MODE          GPIO_MODE_01
#define GPIO1_DIR           GPIO_DIR_IN
#define GPIO1_PULLEN        GPIO_PULL_ENABLE
#define GPIO1_PULL          GPIO_PULL_UP
#define GPIO1_DATAOUT       GPIO_OUT_ZERO
#define GPIO1_DATAINV       GPIO_DATA_UNINV

//Configureation for Pin 2
#define GPIO2_MODE          GPIO_MODE_01
#define GPIO2_DIR           GPIO_DIR_IN
#define GPIO2_PULLEN        GPIO_PULL_ENABLE
#define GPIO2_PULL          GPIO_PULL_UP
#define GPIO2_DATAOUT       GPIO_OUT_ZERO
#define GPIO2_DATAINV       GPIO_DATA_UNINV
```

- All GPIO default settings are in cust_gpio_boot.h
  - GPIO mode , Pull setting , gpio dir

Generate code

Driver use: cust_gpio_usage.h

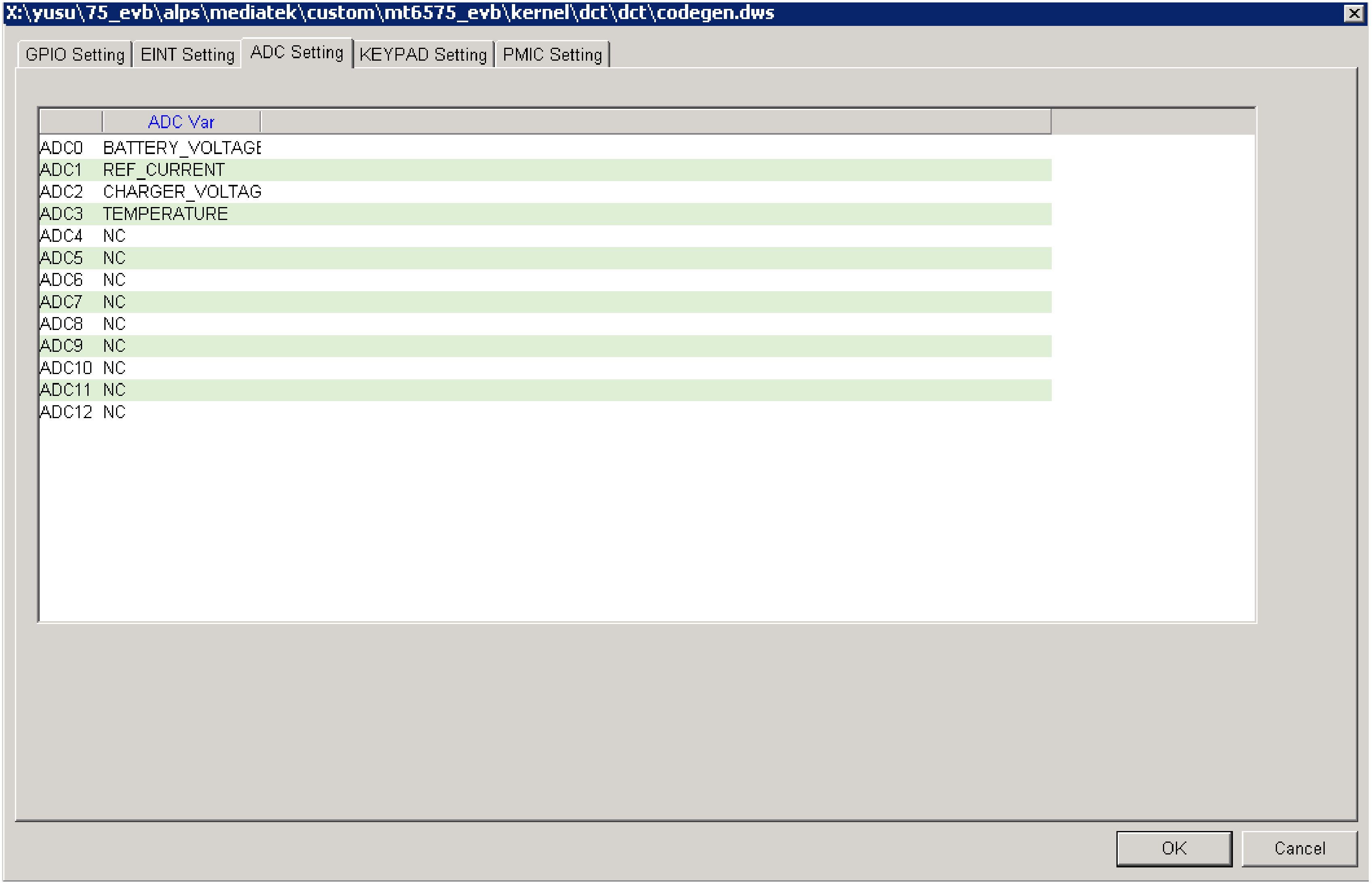


```
#define GPIO_UART_URXD1_PIN         GPIO3
#define GPIO_UART_URXD1_PIN_M_GPIO   GPIO_MODE_00
#define GPIO_UART_URXD1_PIN_M_EINT   GPIO_MODE_02
#define GPIO_UART_URXD1_PIN_M_URXD   GPIO_MODE_01

#define GPIO_UART_UTXD1_PIN         GPIO4
#define GPIO_UART_UTXD1_PIN_M_GPIO   GPIO_MODE_00
#define GPIO_UART_UTXD1_PIN_M_EINT   GPIO_MODE_02
#define GPIO_UART_UTXD1_PIN_M_UTXD   GPIO_MODE_01

#define GPIO_UART_URXD2_PIN         GPIO5
#define GPIO_UART_URXD2_PIN_M_GPIO   GPIO_MODE_00
#define GPIO_UART_URXD2_PIN_M_EINT   GPIO_MODE_02
#define GPIO_UART_URXD2_PIN_M_URXD   GPIO_MODE_01

#define GPIO_UART_UTXD2_PIN         GPIO6
#define GPIO_UART_UTXD2_PIN_M_GPIO   GPIO_MODE_00
#define GPIO_UART_UTXD2_PIN_M_EINT   GPIO_MODE_02
#define GPIO_UART_UTXD2_PIN_M_UTXD   GPIO_MODE_01

#define GPIO_COMBO_PMU_EN_PIN         GPIO25
#define GPIO_COMBO_PMU_EN_PIN_M_GPIO   GPIO_MODE_00
```

Outline

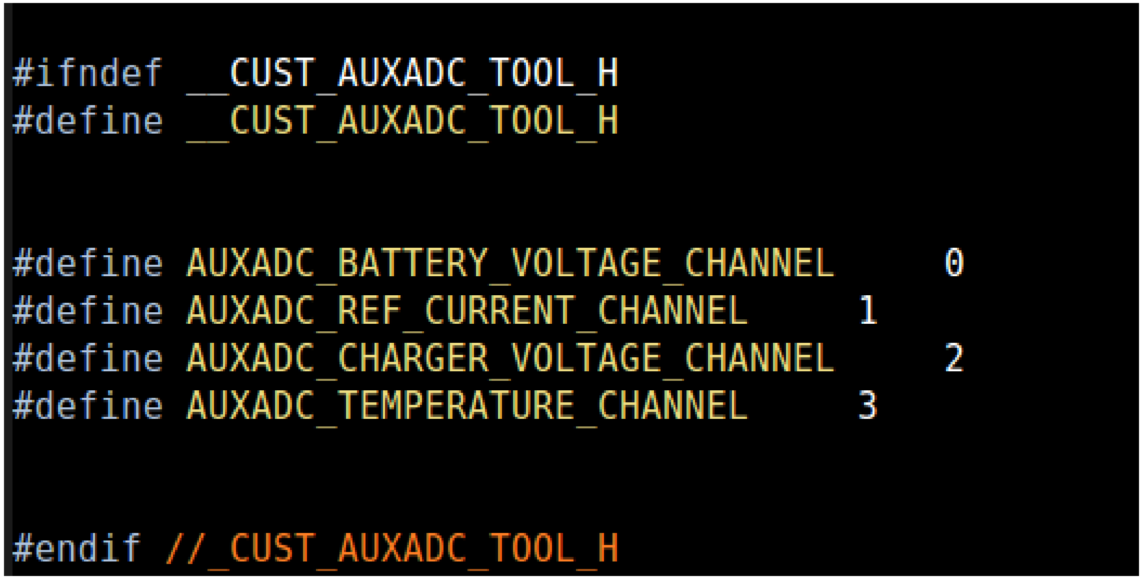
- What is DCT?
- Why use DCT?
- DCT Customization flow
- GPIO
- **ADC**
- EINT
- Keypad

ADC


X:\yusu\75_evb\alps\mediatek\custom\mt6575_evb\kernel\dct\dct\codegen.dws
GPIO Setting
EINT Setting
ADC Setting
KEYPAD Setting
PMIC Setting
OK
Cancel

	ADC Var
ADC0	BATTERY_VOLTAGE
ADC1	REF_CURRENT
ADC2	CHARGER_VOLTAG
ADC3	TEMPERATURE
ADC4	NC
ADC5	NC
ADC6	NC
ADC7	NC
ADC8	NC
ADC9	NC
ADC10	NC
ADC11	NC
ADC12	NC

ADC – cust_adc.h



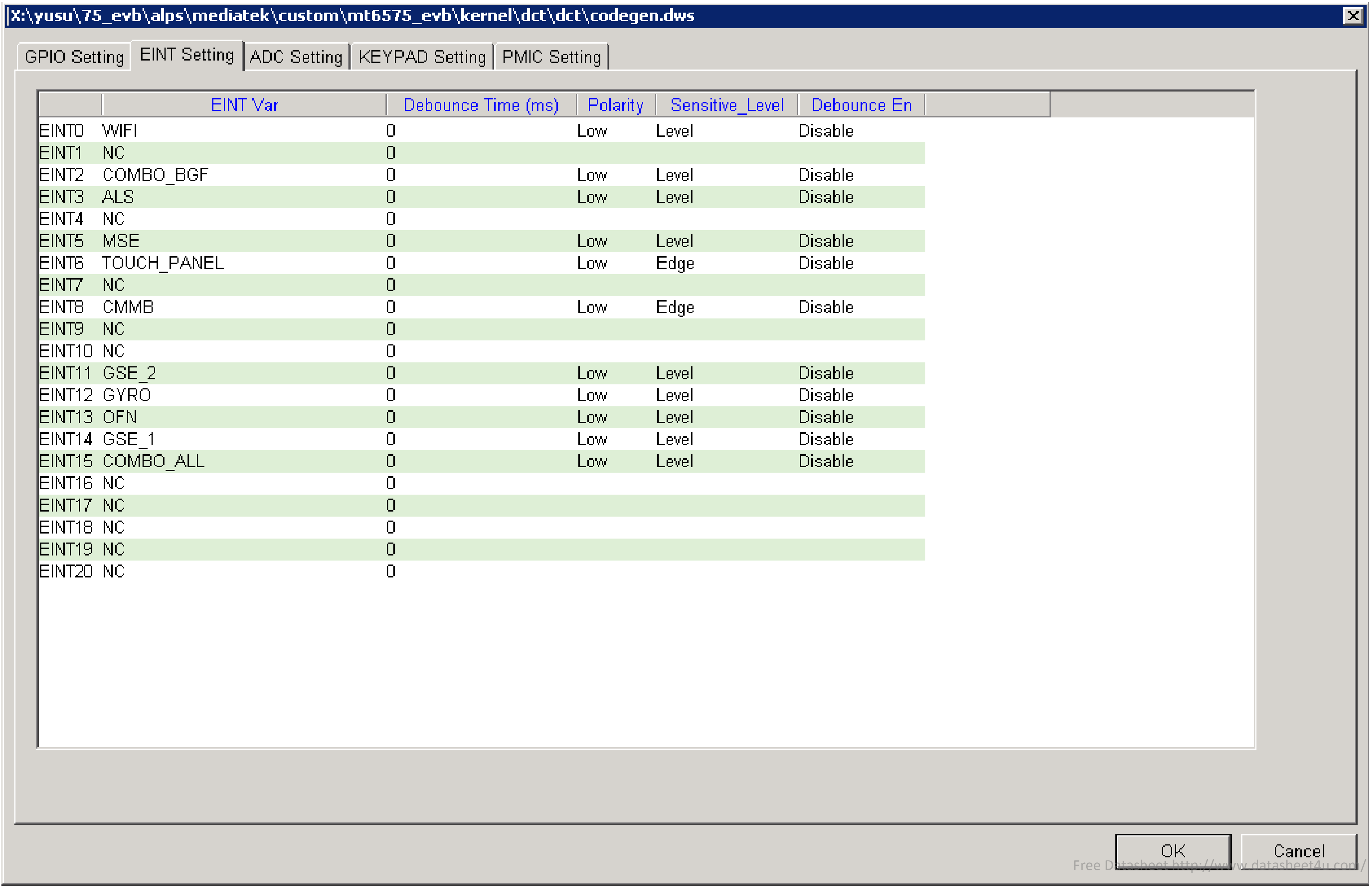
```
#ifndef __CUST_AUXADC_TOOL_H
#define __CUST_AUXADC_TOOL_H


#define AUXADC_BATTERY_VOLTAGE_CHANNEL     0
#define AUXADC_REF_CURRENT_CHANNEL     1
#define AUXADC_CHARGER_VOLTAGE_CHANNEL     2
#define AUXADC_TEMPERATURE_CHANNEL     3


#endif //_CUST_AUXADC_TOOL_H
```

DCT provide an external ADC channel to an ADC Variable

Outline

- What is DCT?
- Why use DCT?
- DCT Customization flow
- GPIO
- ADC
- **EINT**
- Keypad

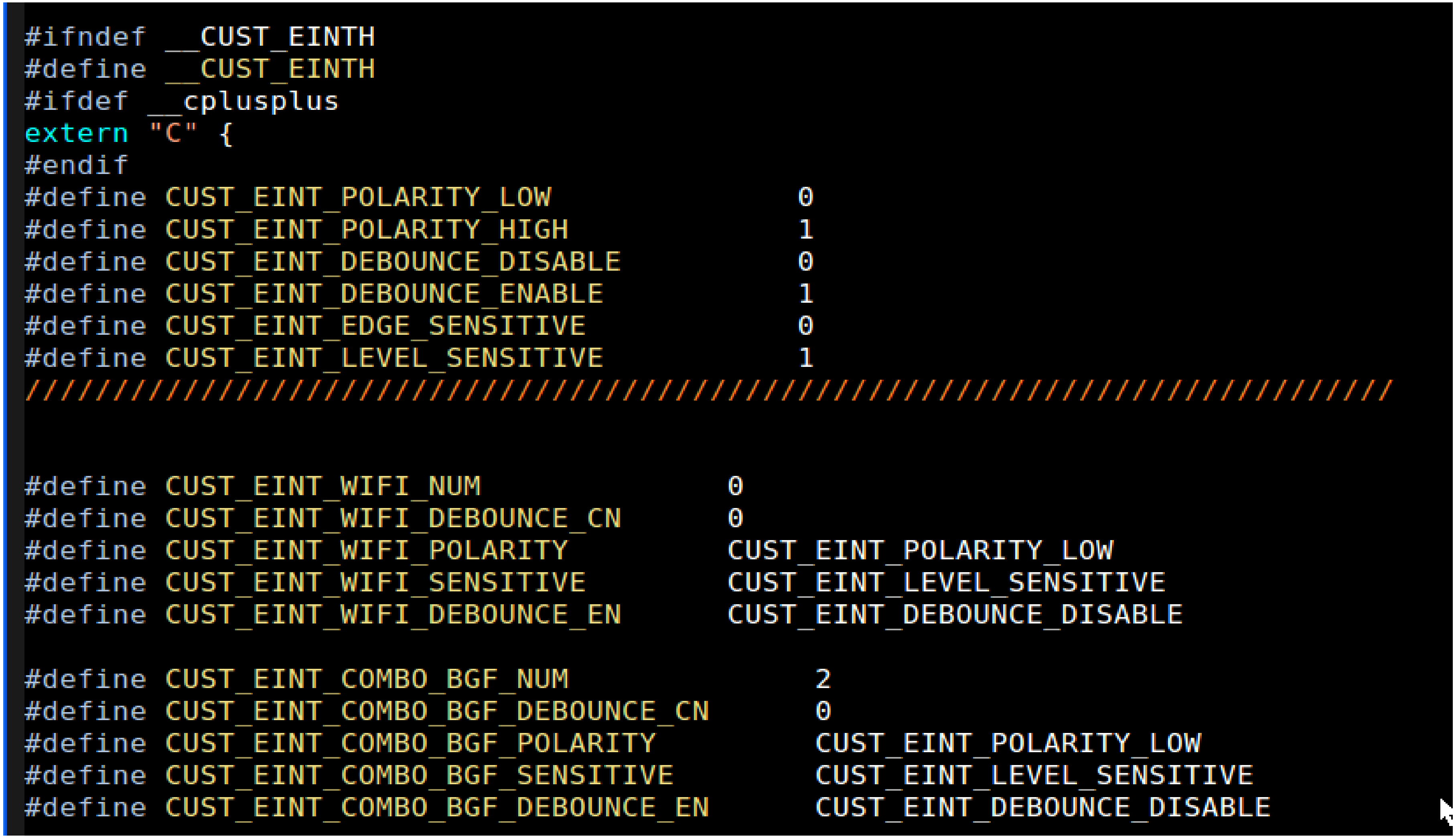
EINT



X:\yusu\75_evb\alps\mediatek\custom\mt6575_evb\kernel\dct\dct\codegen.dws
GPIO Setting
EINT Setting
ADC Setting
KEYPAD Setting
PMIC Setting
OK
Cancel

	EINT Var	Debounce Time (ms)	Polarity	Sensitive_Level	Debounce En
EINT0	WIFI	0	Low	Level	Disable
EINT1	NC	0			
EINT2	COMBO_BGF	0	Low	Level	Disable
EINT3	ALS	0	Low	Level	Disable
EINT4	NC	0			
EINT5	MSE	0	Low	Level	Disable
EINT6	TOUCH_PANEL	0	Low	Edge	Disable
EINT7	NC	0			
EINT8	CMMB	0	Low	Edge	Disable
EINT9	NC	0			
EINT10	NC	0			
EINT11	GSE_2	0	Low	Level	Disable
EINT12	GYRO	0	Low	Level	Disable
EINT13	OFN	0	Low	Level	Disable
EINT14	GSE_1	0	Low	Level	Disable
EINT15	COMBO_ALL	0	Low	Level	Disable
EINT16	NC	0			
EINT17	NC	0			
EINT18	NC	0			
EINT19	NC	0			
EINT20	NC	0			

EINT – cust_eint.h

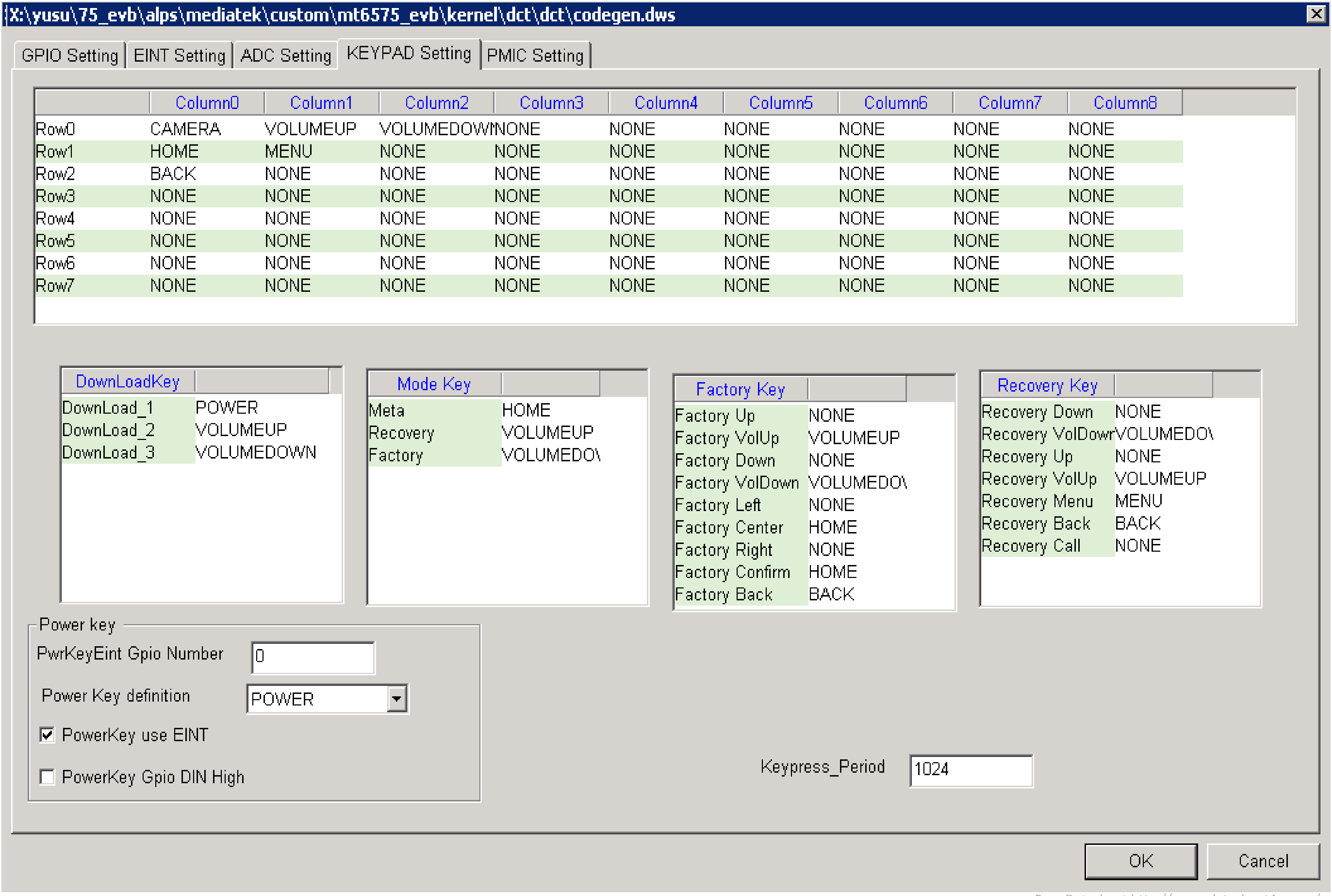


```
#ifndef __CUST_EINTH
#define __CUST_EINTH
#ifdef __cplusplus
extern "C" {
#endif
#define CUST_EINT_POLARITY_LOW              0
#define CUST_EINT_POLARITY_HIGH             1
#define CUST_EINT_DEBOUNCE_DISABLE          0
#define CUST_EINT_DEBOUNCE_ENABLE           1
#define CUST_EINT_EDGE_SENSITIVE            0
#define CUST_EINT_LEVEL_SENSITIVE           1
//////////////////////////////////////////////////////////////////////////////


#define CUST_EINT_WIFI_NUM              0
#define CUST_EINT_WIFI_DEBOUNCE_CN      0
#define CUST_EINT_WIFI_POLARITY         CUST_EINT_POLARITY_LOW
#define CUST_EINT_WIFI_SENSITIVE        CUST_EINT_LEVEL_SENSITIVE
#define CUST_EINT_WIFI_DEBOUNCE_EN      CUST_EINT_DEBOUNCE_DISABLE

#define CUST_EINT_COMBO_BGF_NUM              2
#define CUST_EINT_COMBO_BGF_DEBOUNCE_CN      0
#define CUST_EINT_COMBO_BGF_POLARITY         CUST_EINT_POLARITY_LOW
#define CUST_EINT_COMBO_BGF_SENSITIVE        CUST_EINT_LEVEL_SENSITIVE
#define CUST_EINT_COMBO_BGF_DEBOUNCE_EN      CUST_EINT_DEBOUNCE_DISABLE
```

Outline

- What is DCT?
- Why use DCT?
- DCT Customization flow
- GPIO
- ADC
- EINT
- **Keypad**

Keypad



X:\yusu\75_evb\alps\mediatek\custom\mt6575_evb\kernel\dct\dct\codegen.dws

	Column0	Column1	Column2	Column3	Column4	Column5	Column6	Column7	Column8
Row0	CAMERA	VOLUMEUP	VOLUMEDOWI	NONE	NONE	NONE	NONE	NONE	NONE
Row1	HOME	MENU	NONE	NONE	NONE	NONE	NONE	NONE	NONE
Row2	BACK	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE
Row3	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE
Row4	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE
Row5	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE
Row6	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE
Row7	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE

DownLoadKey	
DownLoad_1	POWER
DownLoad_2	VOLUMEUP
DownLoad_3	VOLUMEDOWN

Mode Key	
Meta	HOME
Recovery	VOLUMEUP
Factory	VOLUMEDO\

Factory Key	
Factory Up	NONE
Factory VolUp	VOLUMEUP
Factory Down	NONE
Factory VolDown	VOLUMEDO\
Factory Left	NONE
Factory Center	HOME
Factory Right	NONE
Factory Confirm	HOME
Factory Back	BACK

Recovery Key	
Recovery Down	NONE
Recovery VolDowr	VOLUMEDO\
Recovery Up	NONE
Recovery VolUp	VOLUMEUP
Recovery Menu	MENU
Recovery Back	BACK
Recovery Call	NONE

Power key

PwrKeyEint Gpio Number: 0

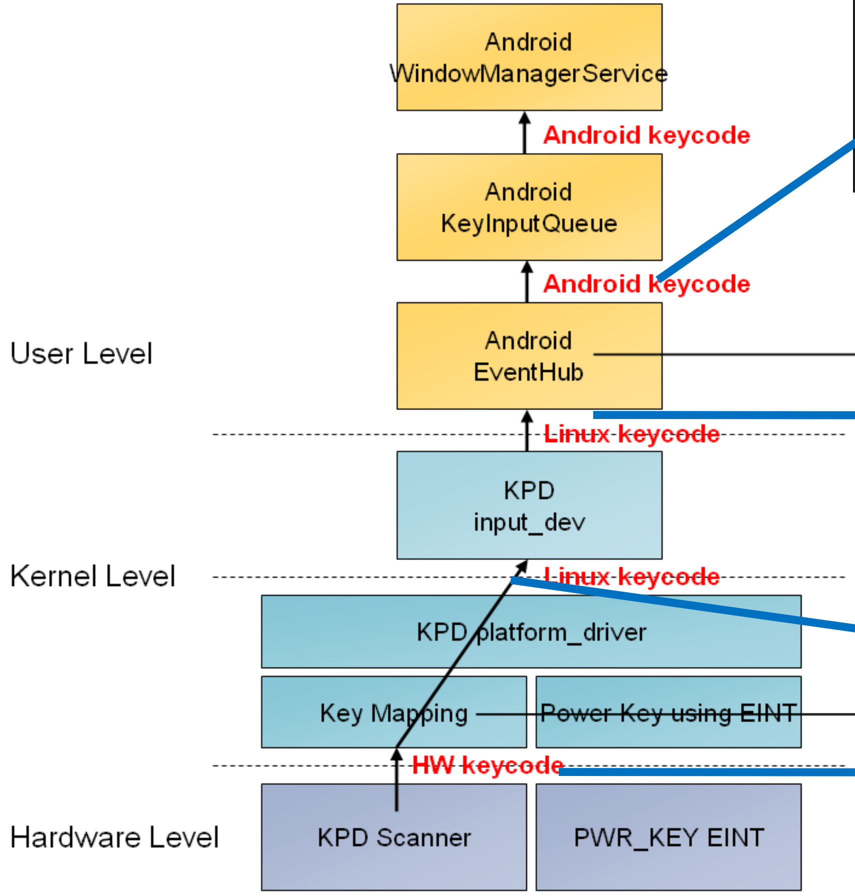
Power Key definition: POWER

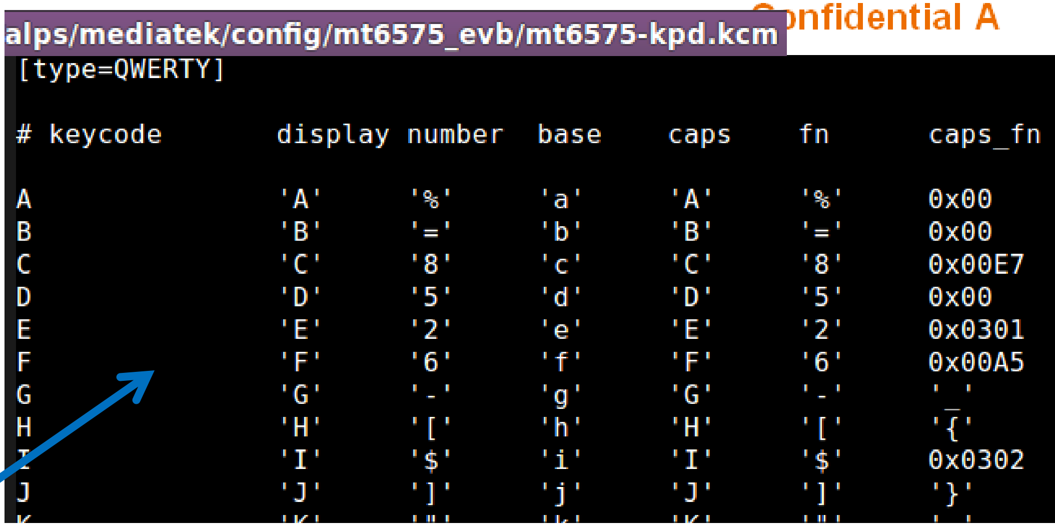
☑ PowerKey use EINT

□ PowerKey Gpio DIN High

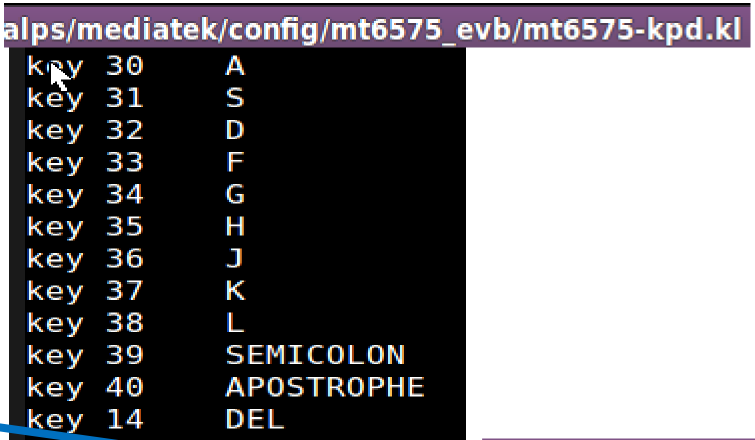
Keypress_Period: 1024

Key Report flow

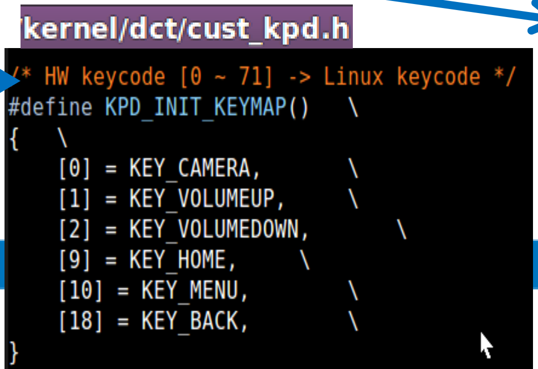



alps/mediatek/config/mt6575_evb/mt6575-kpd.kcm

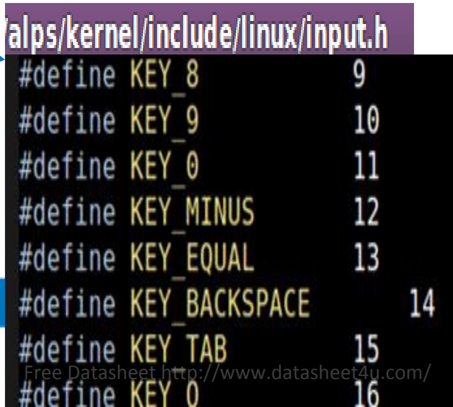
```
[type=QWERTY]

# keycode       display number  base    caps    fn      caps_fn

A               'A'     '%'     'a'     'A'     '%'     0x00
B               'B'     '='     'b'     'B'     '='     0x00
C               'C'     '8'     'c'     'C'     '8'     0x00E7
D               'D'     '5'     'd'     'D'     '5'     0x00
E               'E'     '2'     'e'     'E'     '2'     0x0301
F               'F'     '6'     'f'     'F'     '6'     0x00A5
G               'G'     '-'     'g'     'G'     '-'     '_'
H               'H'     '['     'h'     'H'     '['     '{'
I               'I'     '$'     'i'     'I'     '$'     0x0302
J               'J'     ']'     'j'     'J'     ']'     '}'
```


alps/mediatek/config/mt6575_evb/mt6575-kpd.kl

```
key 30     A
key 31     S
key 32     D
key 33     F
key 34     G
key 35     H
key 36     J
key 37     K
key 38     L
key 39     SEMICOLON
key 40     APOSTROPHE
key 14     DEL
```


kernel/dct/cust_kpd.h

```
/* HW keycode [0 ~ 71] -> Linux keycode */
#define KPD_INIT_KEYMAP()    \
{    \
      [0] = KEY_CAMERA,       \
      [1] = KEY_VOLUMEUP,     \
      [2] = KEY_VOLUMEDOWN,       \
      [9] = KEY_HOME,     \
      [10] = KEY_MENU,        \
      [18] = KEY_BACK,        \
}
```

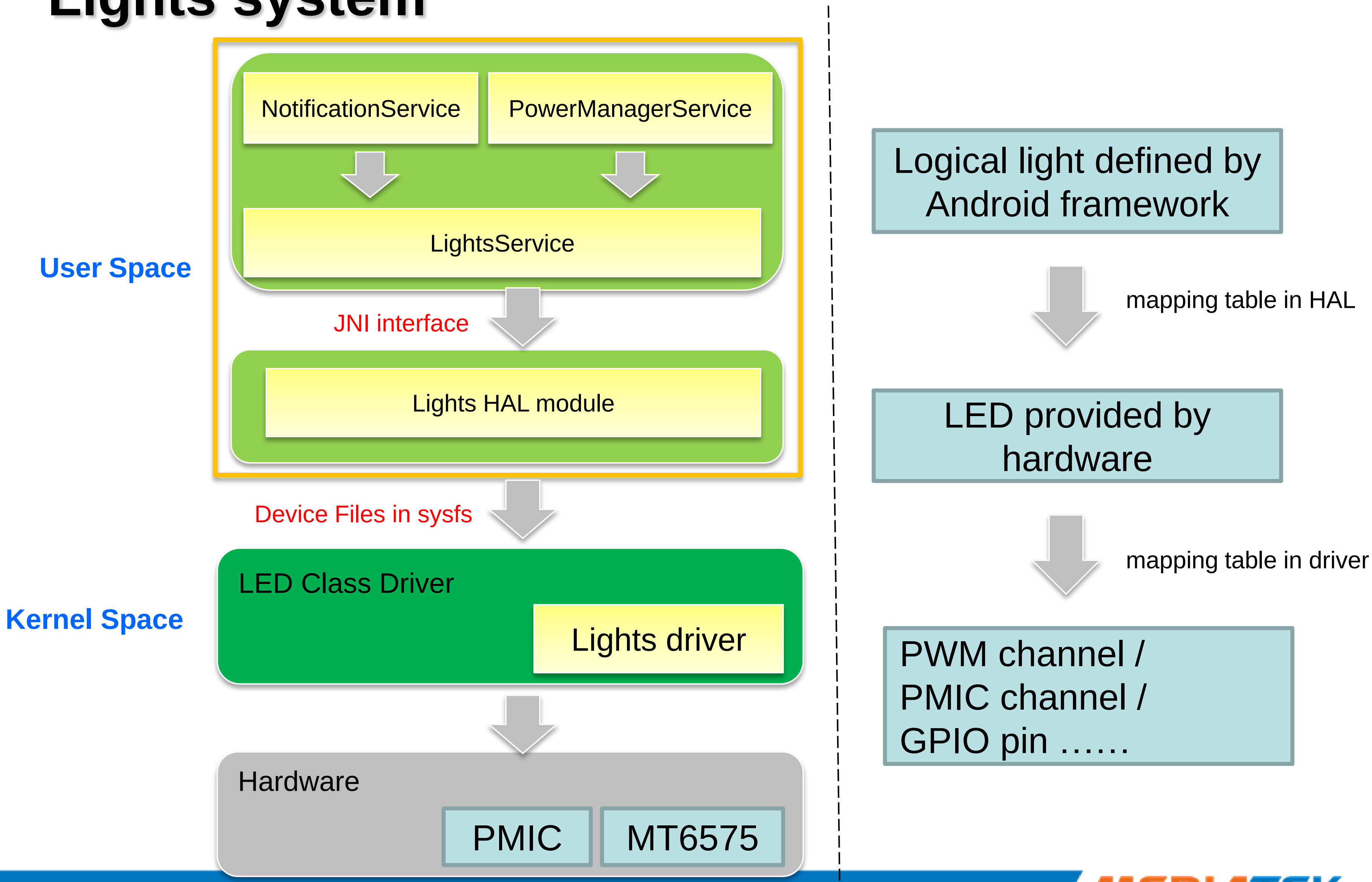

alps/kernel/include/linux/input.h

```
#define KEY_8           9
#define KEY_9           10
#define KEY_0           11
#define KEY_MINUS       12
#define KEY_EQUAL       13
#define KEY_BACKSPACE       14
#define KEY_TAB         15
#define KEY_Q           16
```

Confidential A

Lights System

Lights system



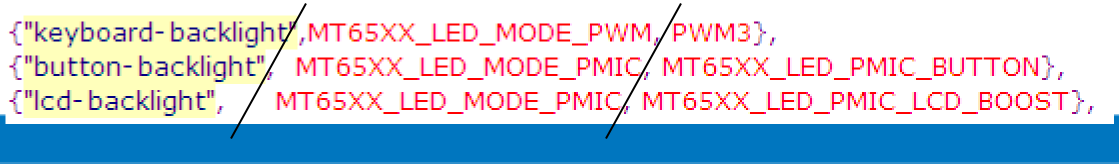
Lights HAL

- Lights.c(./mediatek/source/hardware/liblights/)
  - leds.c(./mediatek/source/kernel/drivers/leds/)
  - Led-class.c(kernel/driver/leds/)
  - Sysfiles: /sys/class/leds/
  - Provide a mechanism for communication between kernel space and user space

```
/
  |-- sys
      |--class
          |--leds
              |--red
              |--green
              |--blue
              |--jogball-backlight
              |--keyboard-backlight
              |--button-backlight
              |--lcd-baclight
```

How to define a light

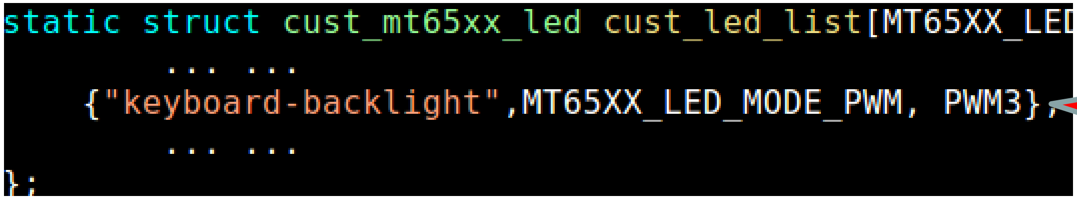
- Name
  - predefined strings, used by both HAL and driver
  - will be the name of sysfs file for each light
    - see */sys/class/leds/*, can be *red*, *green*, *blue*, *lcd-backlight, etc*
- Mode
  - PWM / GPIO / PMIC / CUST
  - used to locate the specific function for each mode, like brightness_set_pwm() or brightness_set_pmic()
- Data
  - indicates which PWM or PMIC channel is used for this light
  - in CUST mode, it's a pointer to customer's self-implemented light control function
- Example



```
{"keyboard-backlight",MT65XX_LED_MODE_PWM, PWM3},
{"button-backlight",  MT65XX_LED_MODE_PMIC, MT65XX_LED_PMIC_BUTTON},
{"lcd-backlight",     MT65XX_LED_MODE_PMIC, MT65XX_LED_PMIC_LCD_BOOST},
```

Changes on MT6575

- The architecture does NOT change from MT6573 to MT6575
- MT6575 add a PMIC chip MT6329 to perform backlight and NLED control
- Differences
  - MT6329 add a pre-charge LED
    - Turn on when charger plug in, turn off when UBOOT set a register bit
    - fixed clock and duty, only on/off can be controlled by software
    - NOT included in kernel’s lights system, due to it can NOT perform hardware accelerated blink
  - PMIC MT6329’s PWM is simpler than MT6573

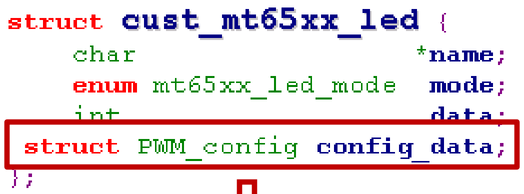


```
static struct cust_mt65xx_led cust_led_list[MT65XX_LED
        ... ...
    {"keyboard-backlight",MT65XX_LED_MODE_PWM, PWM3},
        ... ...
};
```

！！此处PWM3对应于DCT中GPIO配置成PWM2， 其它PWM以此类推

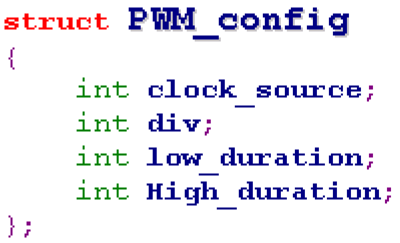
Changes on MT6575

- On MT6575,PWM parameters can be configured
- Detail Differences
  - add a config_data for cust_mt65xx_led ⇨
    - Using PWM , set this config_data for PWM
    - If using PWM default value, set this para {0}

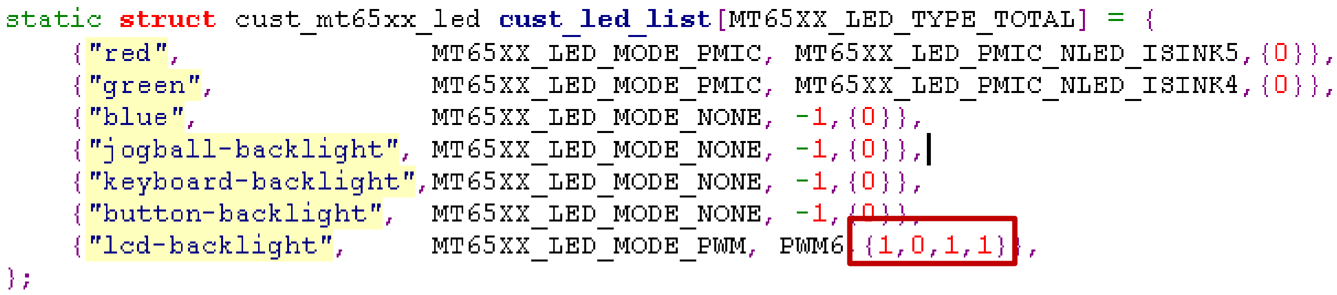


```
struct cust_mt65xx_led {
    char                    *name;
    enum mt65xx_led_mode    mode;
    int                     data;
 struct PWM_config config_data;
};
```

⇩



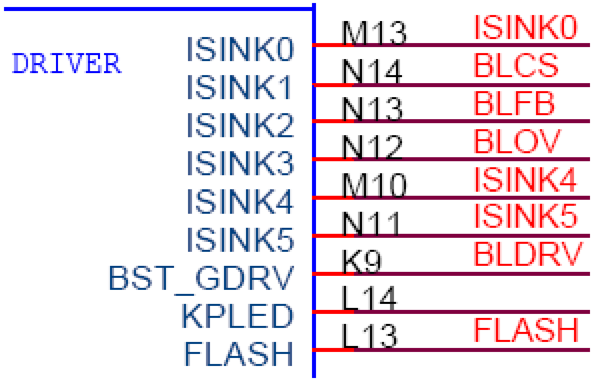
```
struct PWM_config
{
    int clock_source;
    int div;
    int low_duration;
    int High_duration;
};
```



```
static struct cust_mt65xx_led cust_led_list[MT65XX_LED_TYPE_TOTAL] = {
    {"red",                MT65XX_LED_MODE_PMIC, MT65XX_LED_PMIC_NLED_ISINK5,{0}},
    {"green",              MT65XX_LED_MODE_PMIC, MT65XX_LED_PMIC_NLED_ISINK4,{0}},
    {"blue",               MT65XX_LED_MODE_NONE, -1,{0}},
    {"jogball-backlight",  MT65XX_LED_MODE_NONE, -1,{0}},
    {"keyboard-backlight", MT65XX_LED_MODE_NONE, -1,{0}},
    {"button-backlight",   MT65XX_LED_MODE_NONE, -1,{0}},
    {"lcd-backlight",      MT65XX_LED_MODE_PWM, PWM6,{1,0,1,1}},
};
```

MT6329 resource for lights

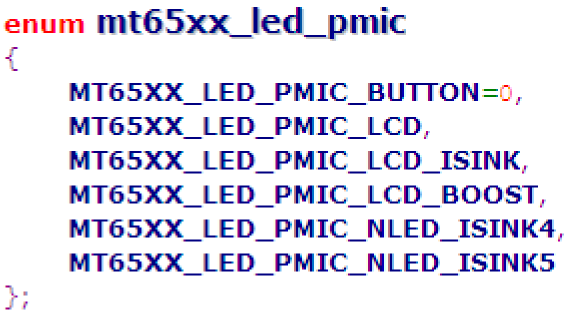
MT6329 pins



- Output channel
  - 1 BOOST drive channel
  - 6 ISINK channels: ISINK0-ISINK5
    - ISINK1, 2, 3 can work for boost mode
  - 1 dedicated button LED control
    - use a fixed 1m Hz clock, div rate and duty are adjustable
- Internal control
  - 3 PWMs: PWM0, PWM1 and PWM2
    - PWM1 and PWM2 has more available frequencies and can work in sleep mode
    - If you have more LEDs, use MT6575’s PWM
    - 2 attributes adjustable for each PWM
      - frequency & duty
- So there are several combinations for output channels and internal controls

PMIC modes

- MT65XX_LED_PMIC_BUTTON
  - use the button led control channel
- MT65XX_LED_PMIC_LCD
  - NOT implemented
- MT65XX_LED_PMIC_LCD_ISINK
  - use ISINK1, 2 and 3 on ISINK mode, PWM0 for control
- MT65XX_LED_PMIC_LCD_BOOST
  - use BOOST output and ISINK1, 2, 3 on BOOST mode
  - PWM0 for control
- MT65XX_LED_PMIC_NLED_ISINK4
  - use ISINK4, PWM1 for control
- MT65XX_LED_PMIC_NLED_ISINK5
  - use ISINK5, PWM2 for control

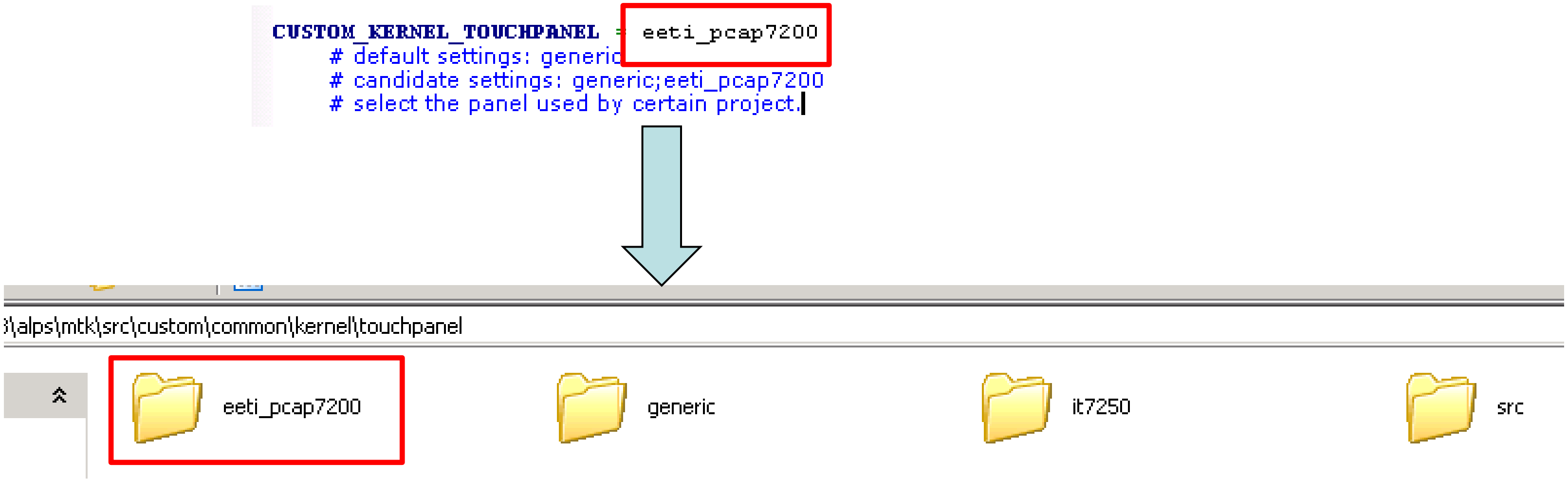


```
enum mt65xx_led_pmic
{
        MT65XX_LED_PMIC_BUTTON=0,
        MT65XX_LED_PMIC_LCD,
        MT65XX_LED_PMIC_LCD_ISINK,
        MT65XX_LED_PMIC_LCD_BOOST,
        MT65XX_LED_PMIC_NLED_ISINK4,
        MT65XX_LED_PMIC_NLED_ISINK5
};
```

Touch

Touch Panel Driver in ALPS

- **Select TP Driver's type**
  - makefile option in mediatek\config\$(project)\ProjectConfig.mk



```
CUSTOM_KERNEL_TOUCHPANEL = eeti_pcap7200
     # default settings: generic
     # candidate settings: generic;eeti_pcap7200
     # select the panel used by certain project.
```

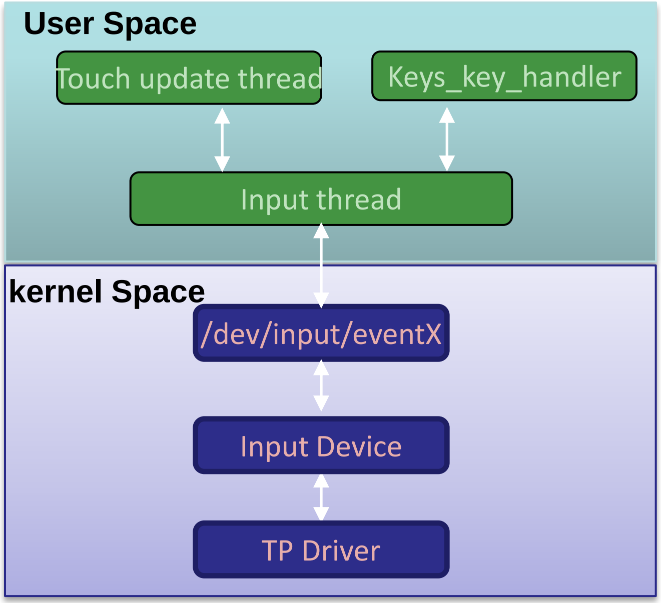
3\alps\mtk\src\custom\common\kernel\touchpanel

Generic	---- >	R touch
Src	---- >	R and C touch Common part
Others	---- >	C touch

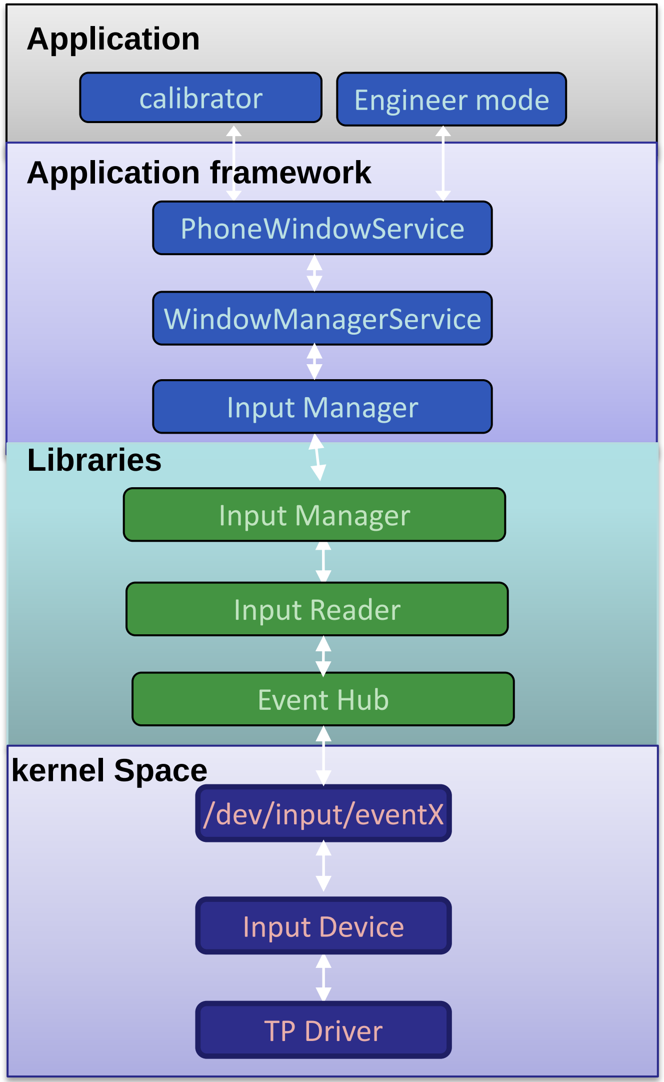
Touch Panel Driver in ALPS

- **Architecture**

Factory Mode

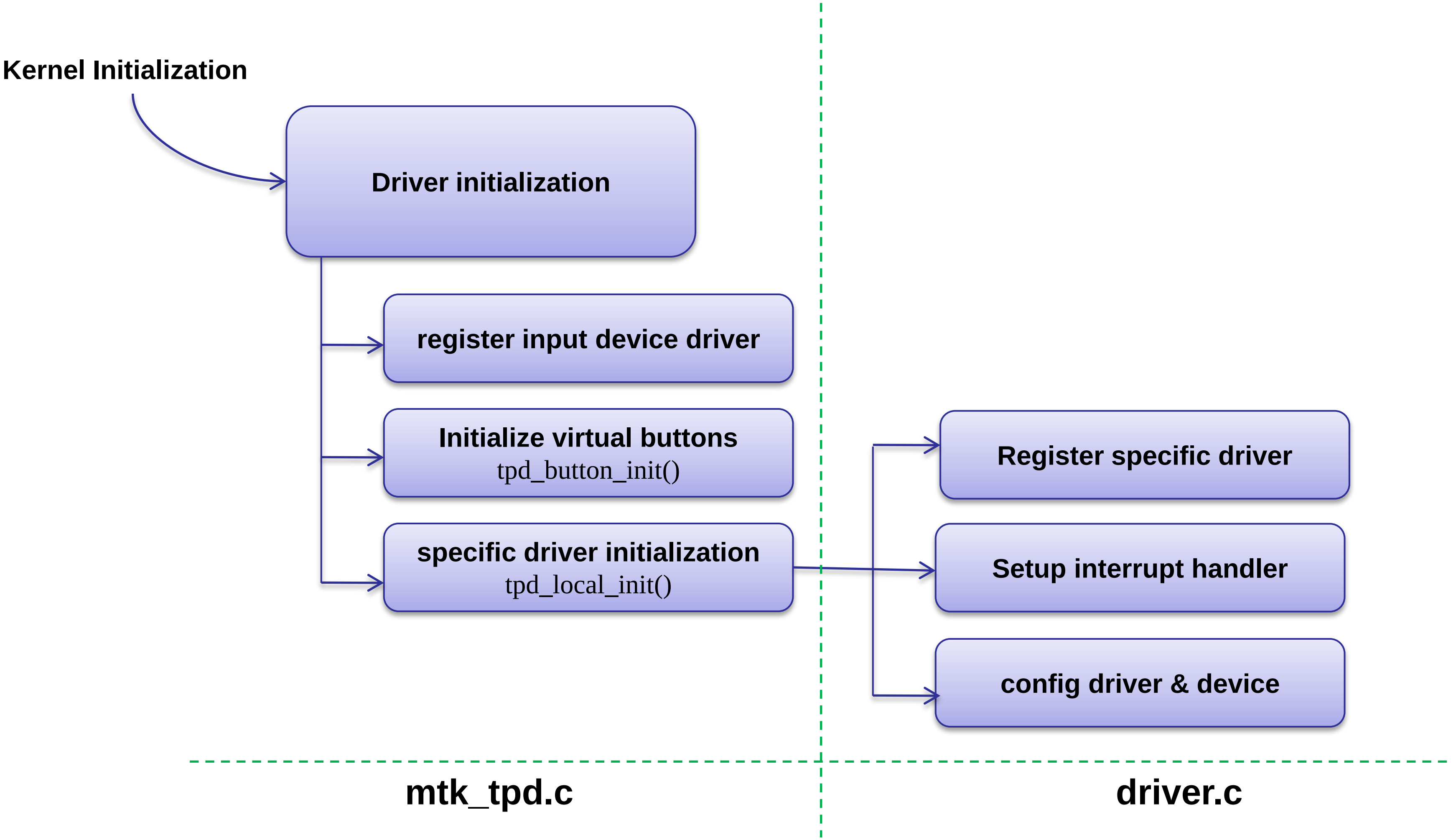


Normal Mode



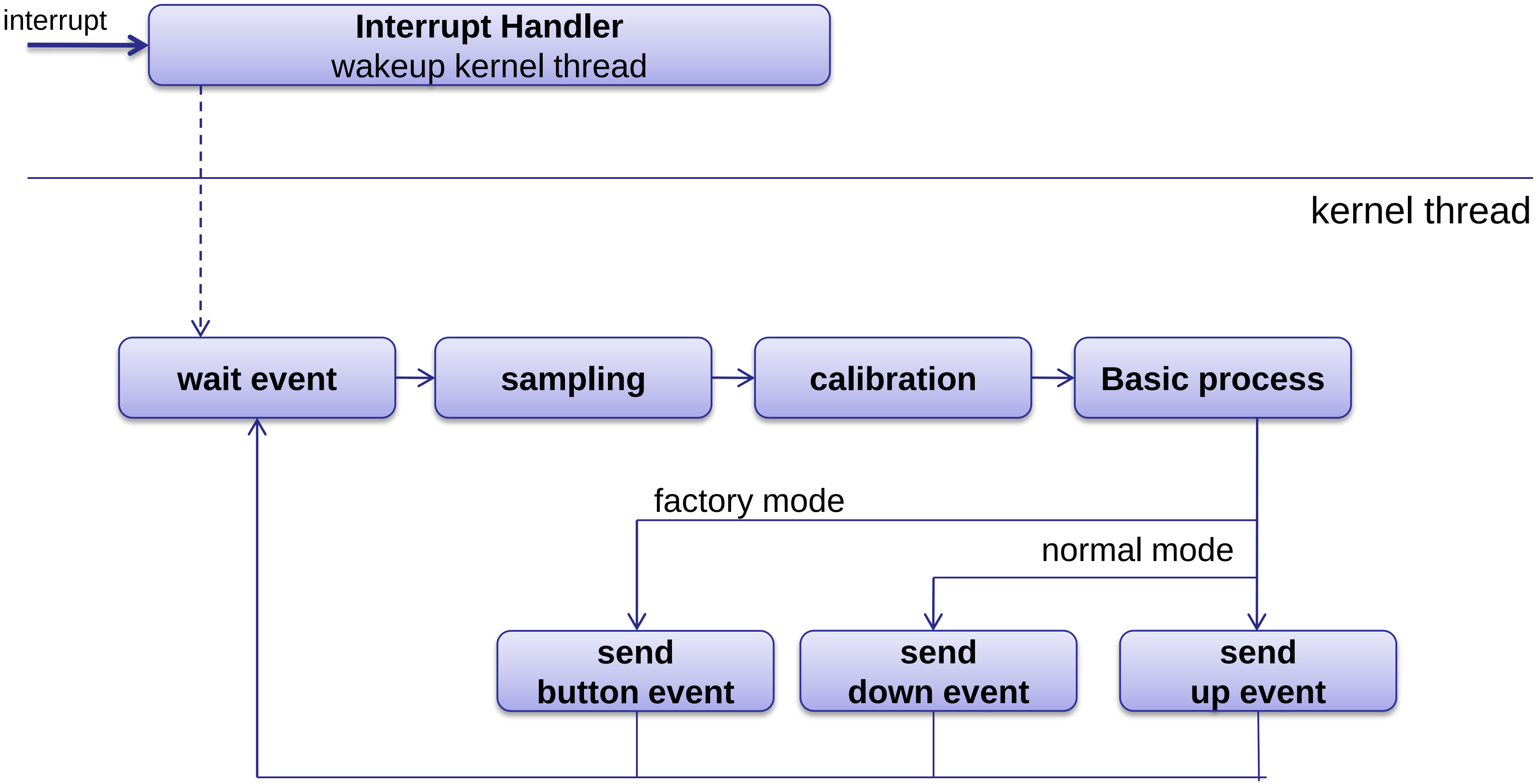
Touch Panel Driver Flow

- Initialization



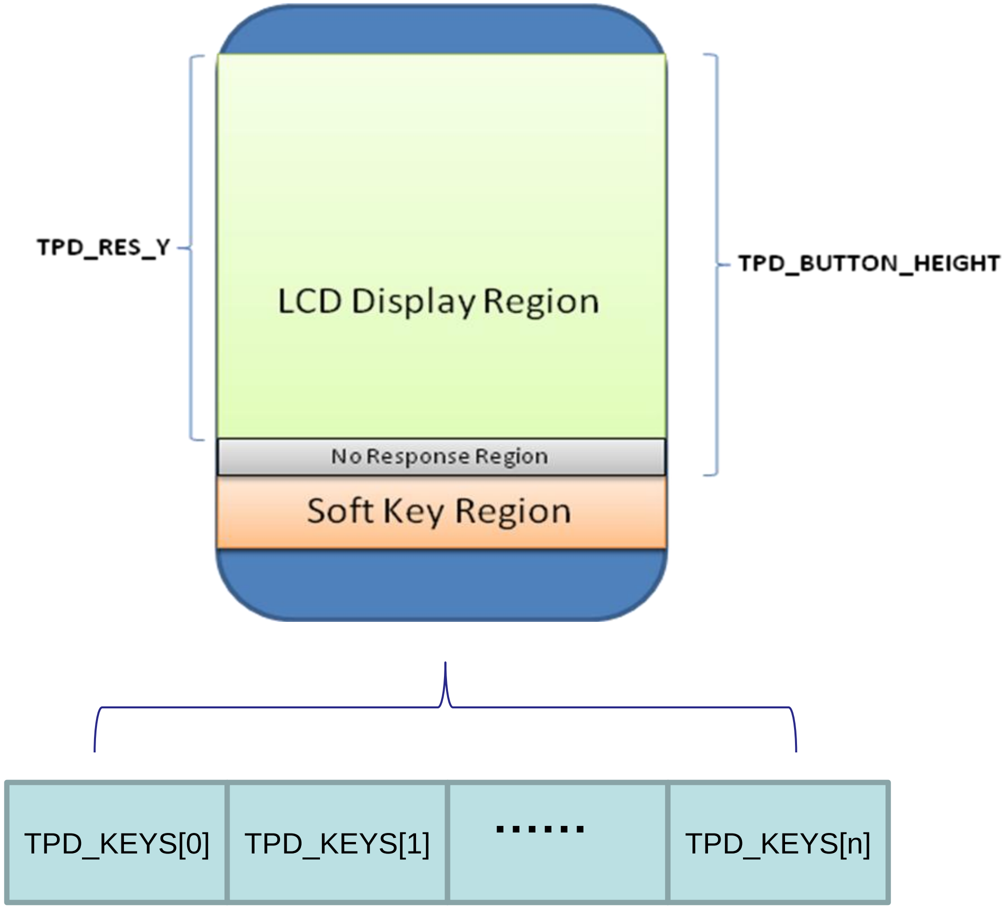
Touch Panel Driver Flow

- Event Handling



Button Related Customization

- Tpd Button



TPD_KEYS[0]	TPD_KEYS[1]	……	TPD_KEYS[n]

Button Related Customization

category	name	type	description
Button Related	TPD_HAVE_BUTTON	Any	If virtual button is needed to be implemented by touch panel driver, define this macro.
	TPD_CUSTOM_BUTTON	Any	If button layout is different with predefined ones, this macro should be defined and the function **tpd_button** should be implemented
	TPD_BUTTON_HEIGHT	int	It defines the actual y coordinate of touch panel where soft key should be recognized.
	TPD_KEY_COUNT	int	Defines the number of soft key
	TPD_KEYS	int array	Defines the key code of each soft key.

File Name	Location
Tpd_custom_xx.h	Alps\mediatek\custom\$(project)\kernel\touchpanel\${touch folder}\

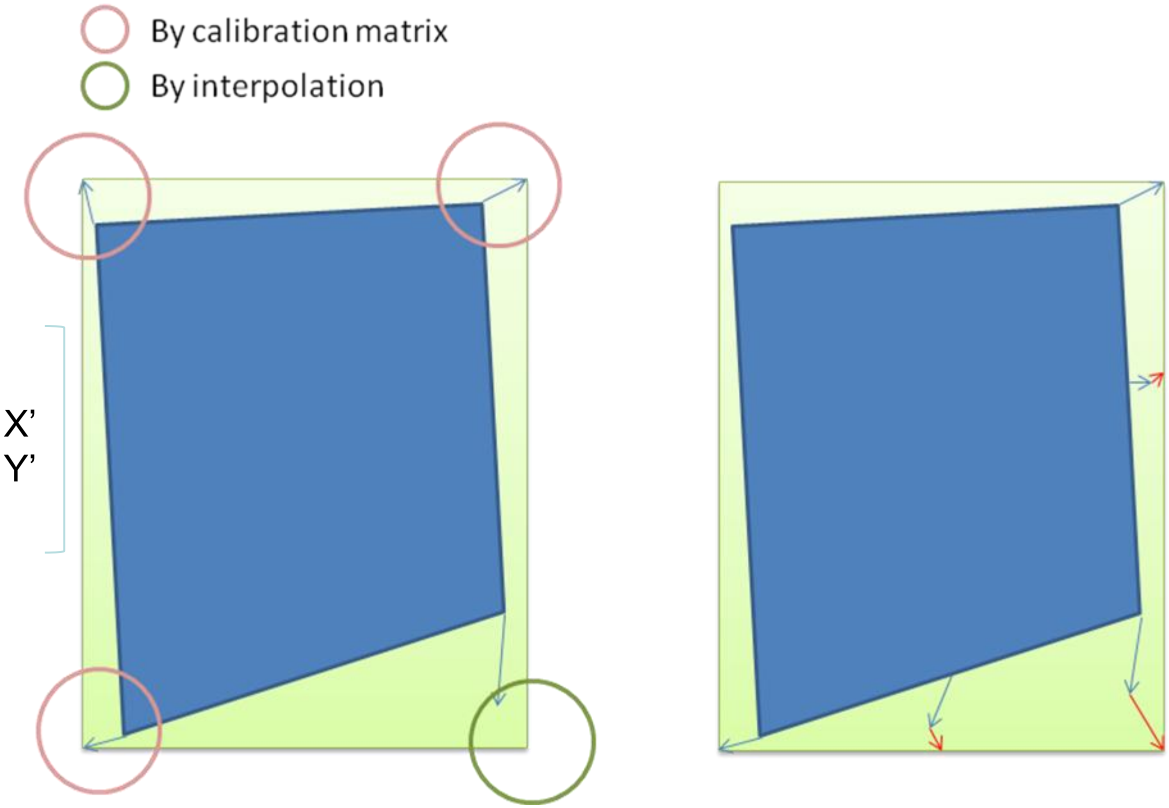
Calibration Related Customization

- **Calibration**

$$\begin{bmatrix} X' \\ Y' \end{bmatrix} = \frac{\begin{bmatrix} A & B & C \\ D & E & F \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}}{4096}$$

$$\begin{bmatrix} X'' \\ Y'' \end{bmatrix} = \begin{bmatrix} 1 & \frac{(V_x)X'}{TPD\_RES\_x*TPD\_RES\_y} \\ \frac{(V_y)Y'}{TPD\_RES\_x*TPD\_RES\_y} & 1 \end{bmatrix} \begin{bmatrix} X' \\ Y' \end{bmatrix}$$

TPD_CALIBRATION_MATRIX:
{ A, B, C, D, E, F, Vx, Vy }



Calibration Related Customization

category	name	type	description
Calibration	TPD_HAVE_CALIBRATION	any	If it is defined, touch panel calibration functionality will be turned on
	TPD_CALIBRATION_MATRIX	int array	It’s an 8 elements integer array. It defined the default calibration matrix for touch panel driver.
	TPD_CUSTOM_CALIBRATION	any	If it’s needed to implement customized calibration function, define this macro and implement tpd_calibrate() function.
	TPD_WARP_START	int array	These two macros should be defined as an integer array with 4 elements. They should be both defined to enable calibration warp around edge. When they are defined, warp algorithm will be applied to defined edge region.
	TPD_WARP_END	int array	

R-type Touch Panel Calibrator matrix in tpd_custom_xx.h

{ TPD_RES_X*4,0,0, 0, TPD_RES_Y*4,0 } -- >MT6516
{ TPD_RES_X,0,0, 0, TPD_RES_Y,0 } -- >MT6573

R-type Touch Panel Customization

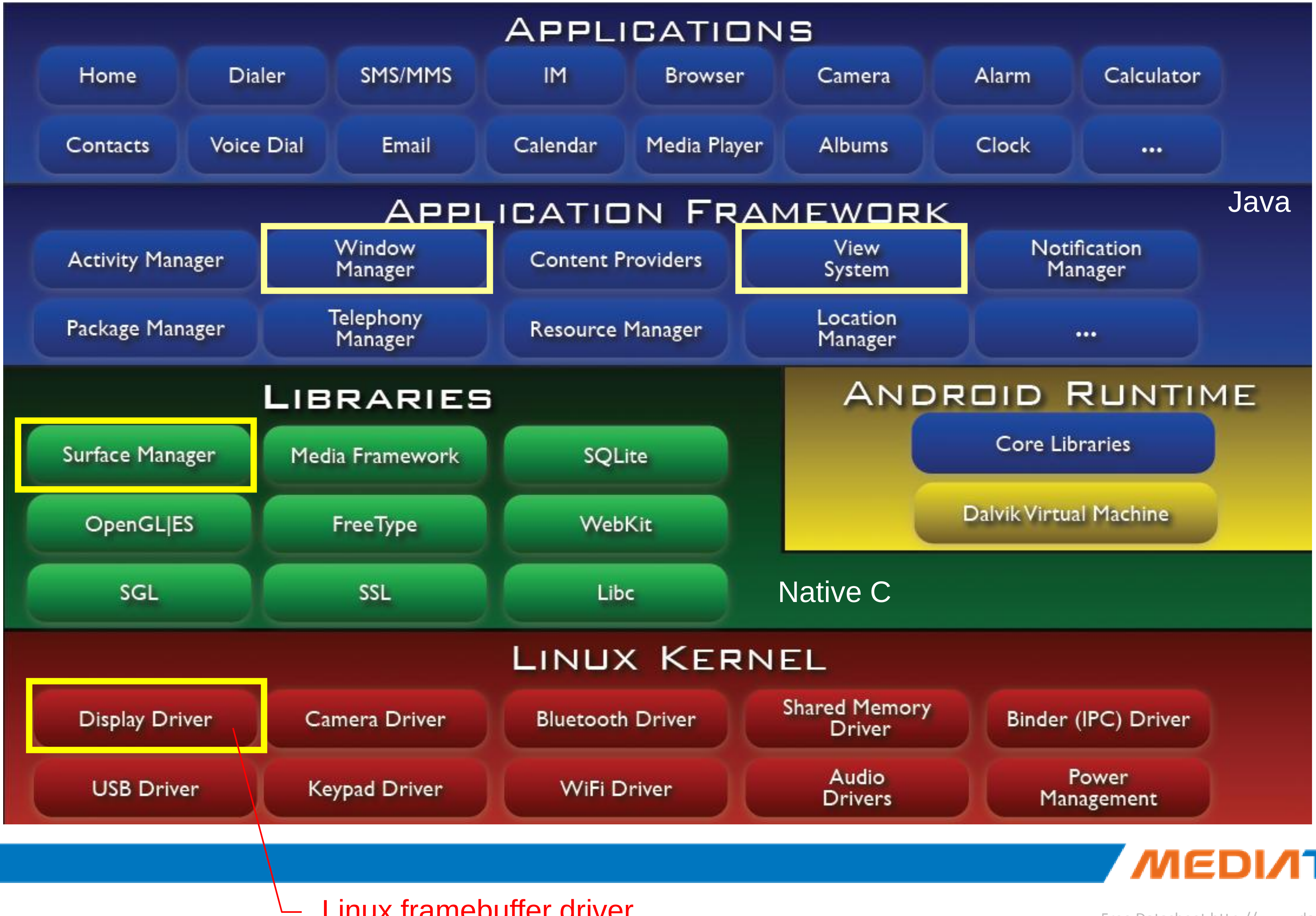
category	name	type	description
Pressure Related	TPD_DELAY	int	in jiffies, next timeout value for tasklet. It controls event rate; faster event rate with smaller TPD_DELAY
	TPD_PRESSURE_MAX	int	Defines the maximum pressure that can be generated by touch panel.
	TPD_PRESSURE_MIN	Int	Defines the minimum pressure that can be generated by touch panel.
	TPD_PRESSURE_NICE	Int	Defines the “nice” pressure of event. If event has larger pressure value than TPD_PRESSURE_NICE, it will be queued and judged whether it is a valid event by following event.

C-type Touch Panel Customization

category	name	type	description
	TPD_POWER_SOURCE	int	Define power source of touch panel component. Refer to mt6573_pll.h for detail power source list.
	TPD_I2C_NUMBER	int	I2c controller number touch panel is on. It should be 0, 1.

LCM

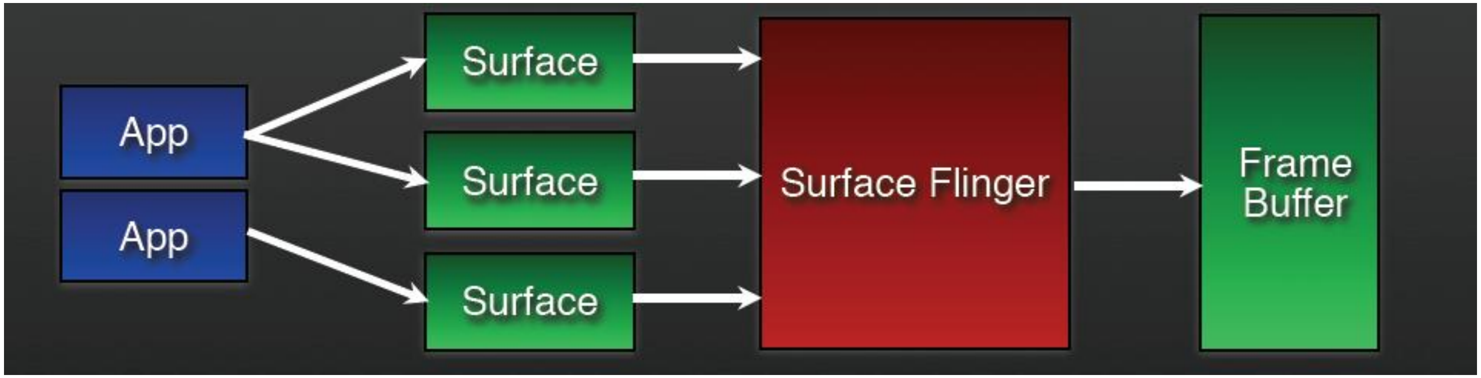
Android SW Stack Overview



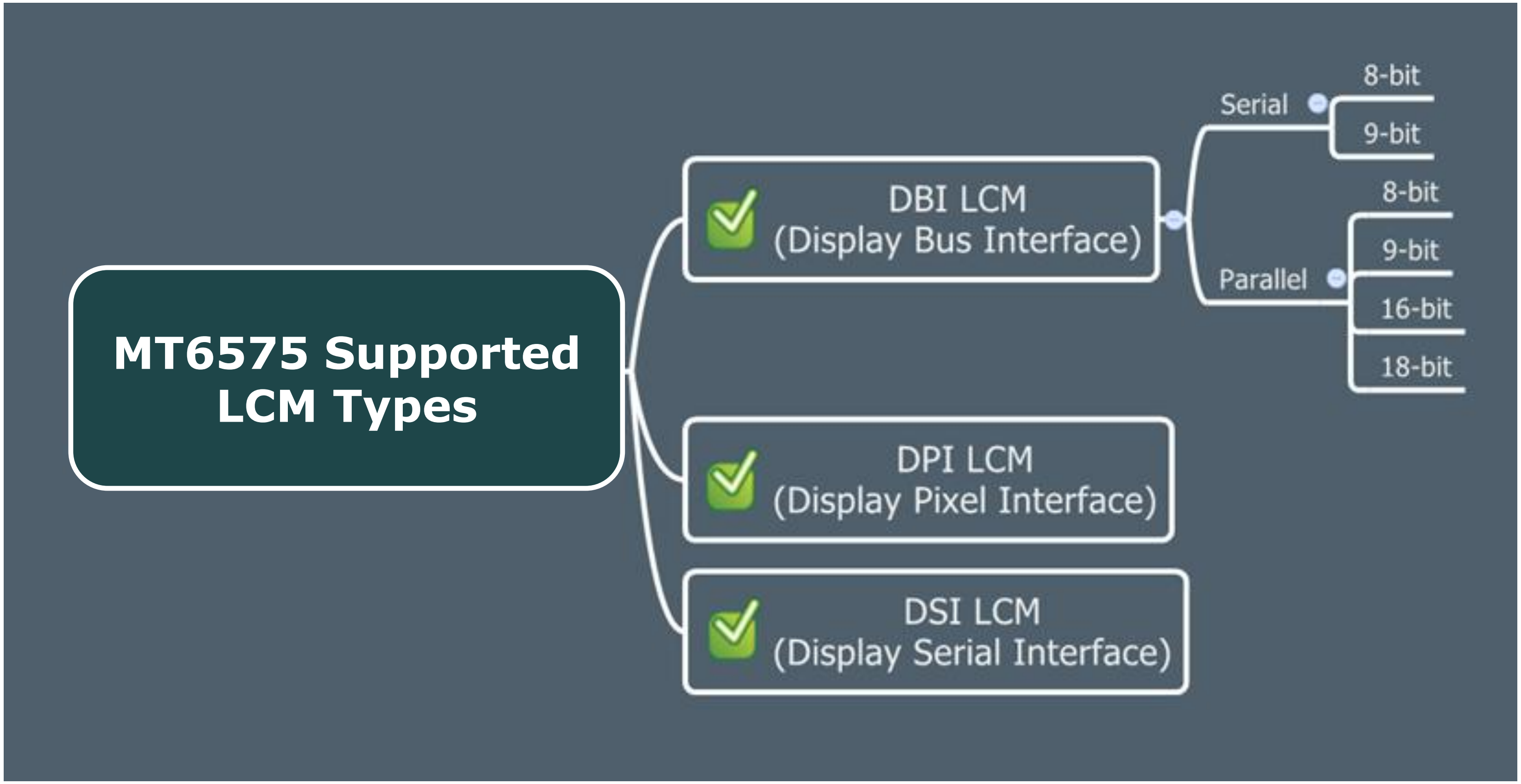
Linux framebuffer driver

SurfaceFlinger

- Handling all surface rendering to frame buffer device
- Can combine 2D and 3D surfaces and surfaces from multiple applications
- Can use OpenGL ES and 2D hardware accelerator for its compositions



MT6575 Supported LCM Types



MT6575 Supported LCM Types

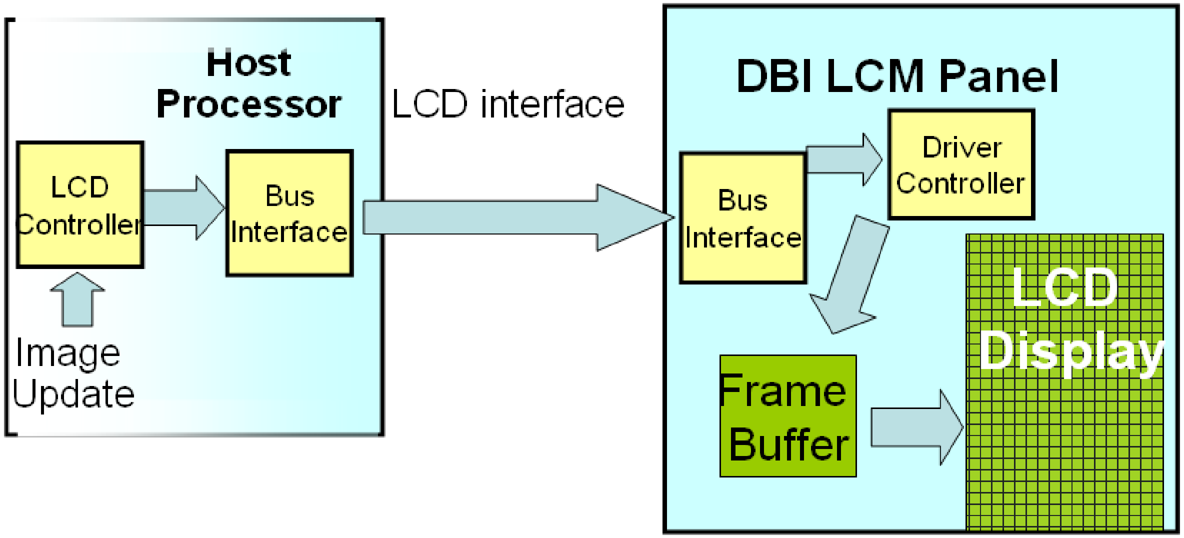
✓ DBI LCM (Display Bus Interface)

- Serial
  - 8-bit
  - 9-bit
- Parallel
  - 8-bit
  - 9-bit
  - 16-bit
  - 18-bit

✓ DPI LCM (Display Pixel Interface)

✓ DSI LCM (Display Serial Interface)

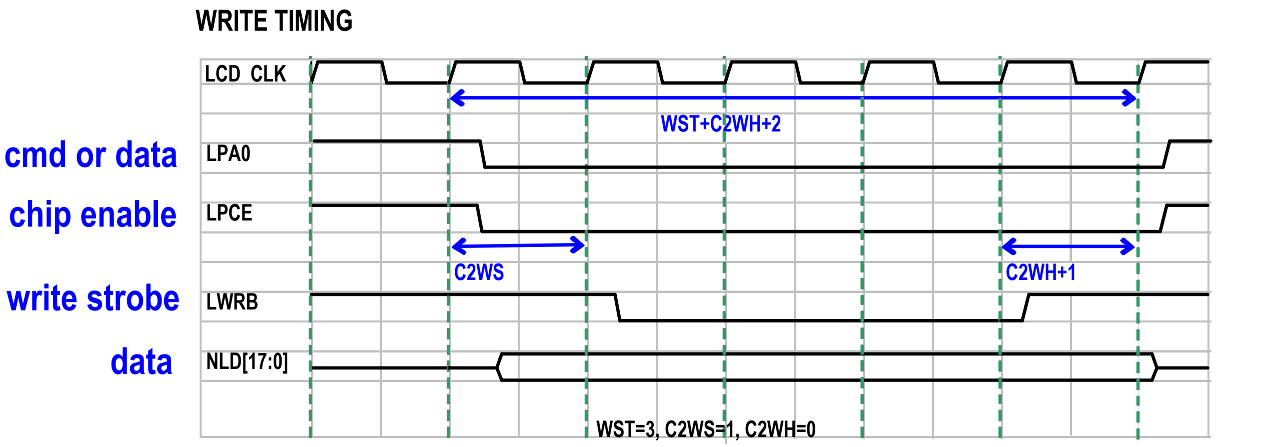
DBI (Display Bus Interface) LCM (1/3)

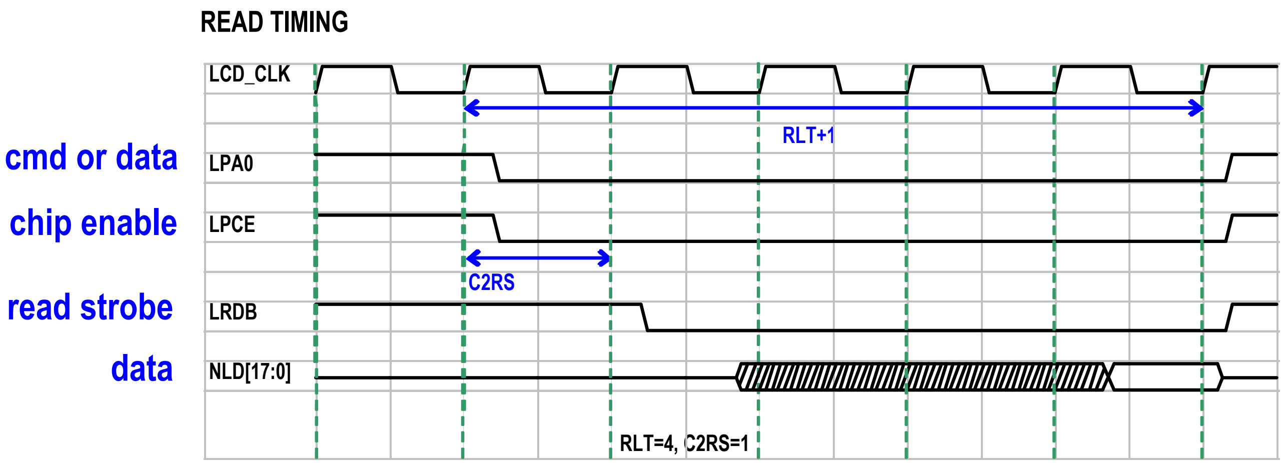


LCM equips with its own RAM

DBI (Display Bus Interface) LCM (2/3)

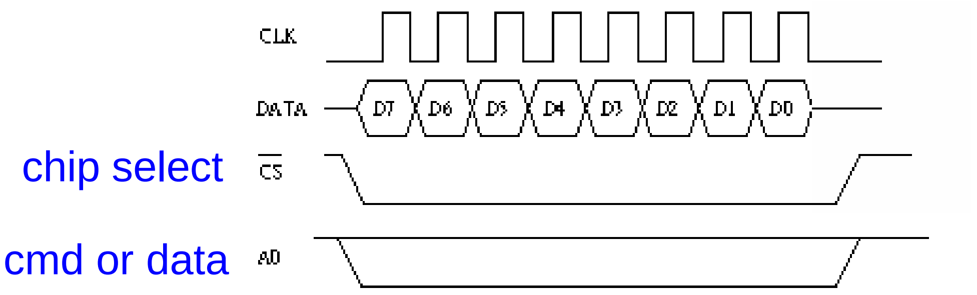
- DBI interface timing (parallel I/F)





DBI (Display Bus Interface) LCM (3/3)

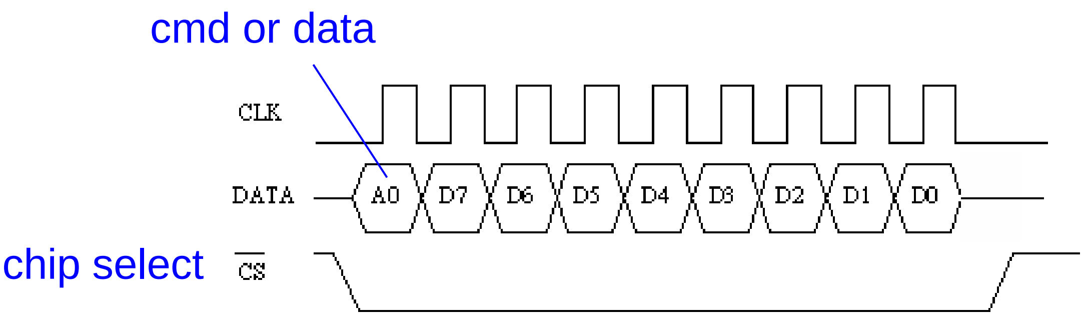
- DBI interface timing (8-bit / 9-bit serial I/F)



chip select

cmd or data

8 bits

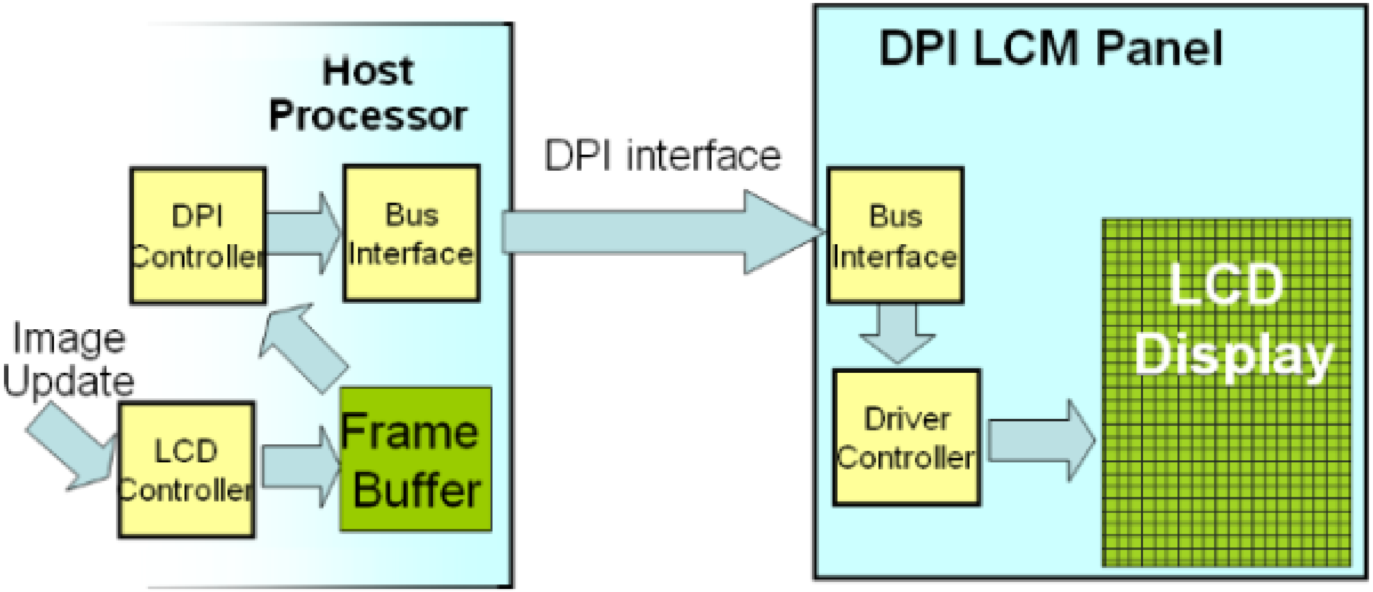


cmd or data

chip select

9 bits

DPI (Display Pixel Interface) LCM (1/2)



DPI (Display Pixel Interface) LCM (2/2)

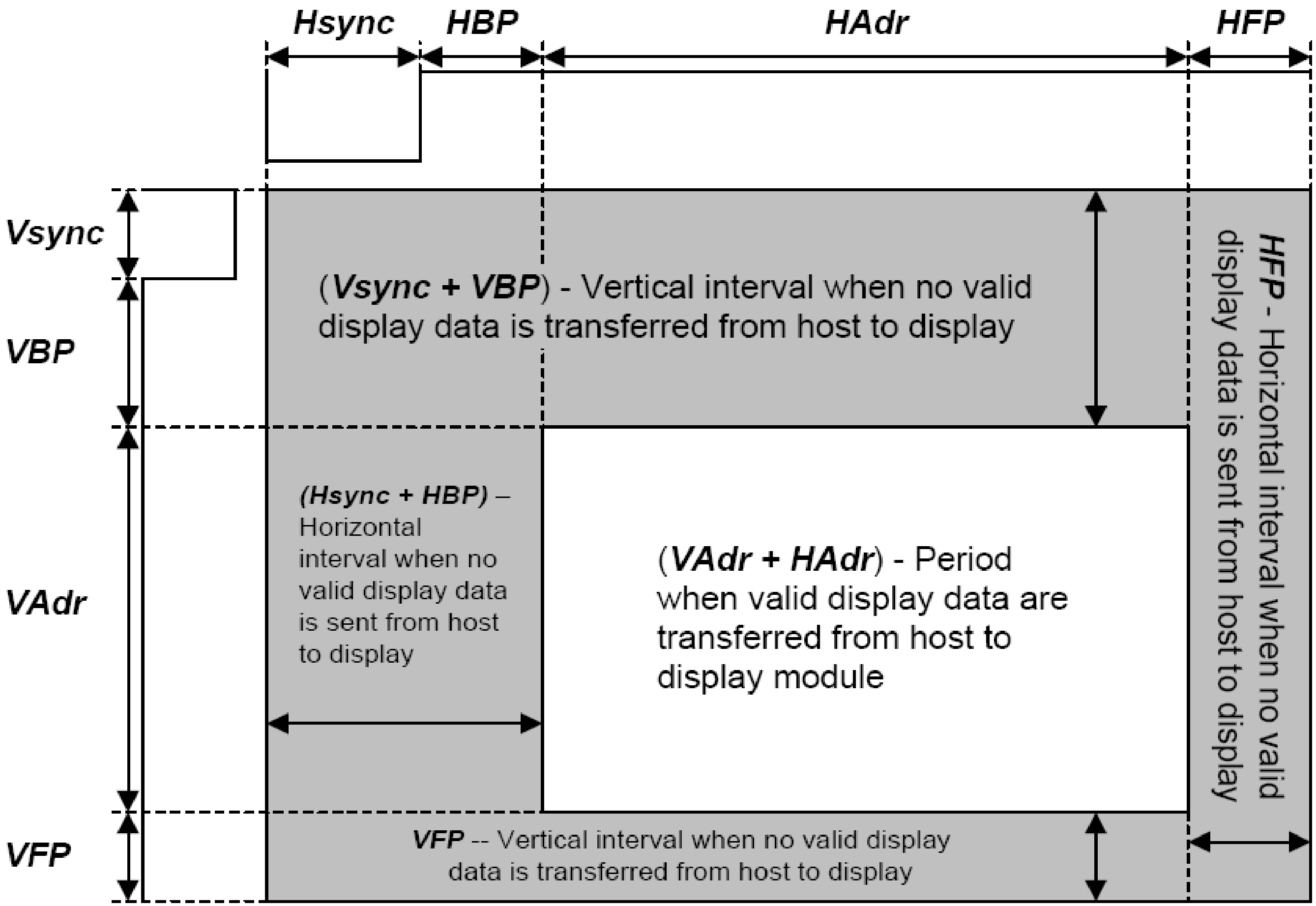
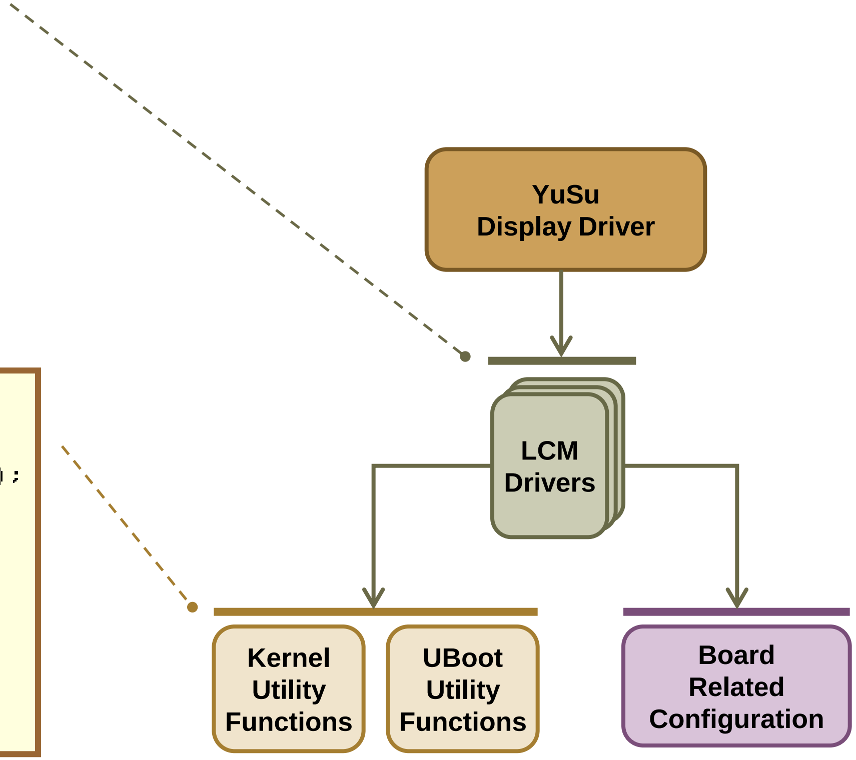

Hsync
HBP
HAdr
HFP
Vsync
VBP
VAdr
VFP
(Vsync + VBP) - Vertical interval when no valid display data is transferred from host to display
(Hsync + HBP) – Horizontal interval when no valid display data is sent from host to display
(VAdr + HAdr) - Period when valid display data are transferred from host to display module
HFP - Horizontal interval when no valid display data is sent from host to display
VFP -- Vertical interval when no valid display data is transferred from host to display

Figure 6 DPI Timing Parameters

LCM Driver Interface

```
typedef struct
{
    void (*set_util_funcs)(const LCM_UTIL_FUNCS *util);
    void (*get_params)(LCM_PARAMS *params);

    void (*init)(void);
    void (*suspend)(void);
    void (*resume)(void);

    void (*update)(unsigned int x, unsigned int y,
                   unsigned int width, unsigned int height);
} LCM_DRIVER;
```

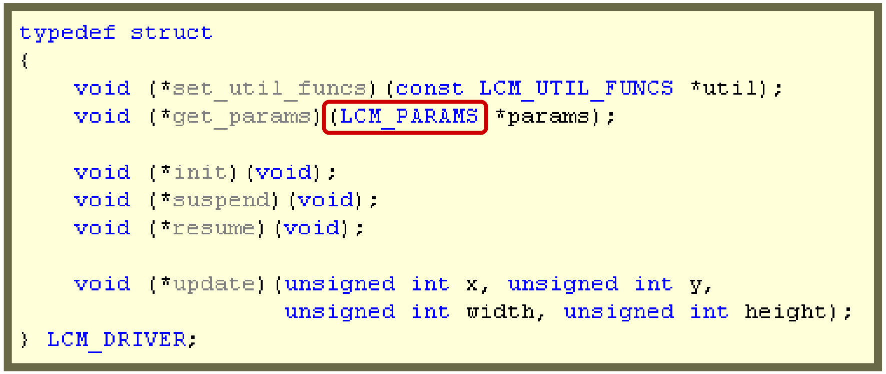
```
typedef struct
{
    void (*set_reset_pin)(unsigned int value);
    int  (*set_gpio_out)(unsigned int gpio, unsigned int value);

    void (*udelay)(unsigned int us);
    void (*mdelay)(unsigned int ms);

    void (*send_cmd)(unsigned int cmd);
    void (*send_data)(unsigned int data);
    unsigned int (*read_cmd)(void);
    unsigned int (*read_data)(void);
} LCM_UTIL_FUNCS;
```



- Reset pin control function
- GPIO control function
- Delay function
- ...

- GPIO settings
- …

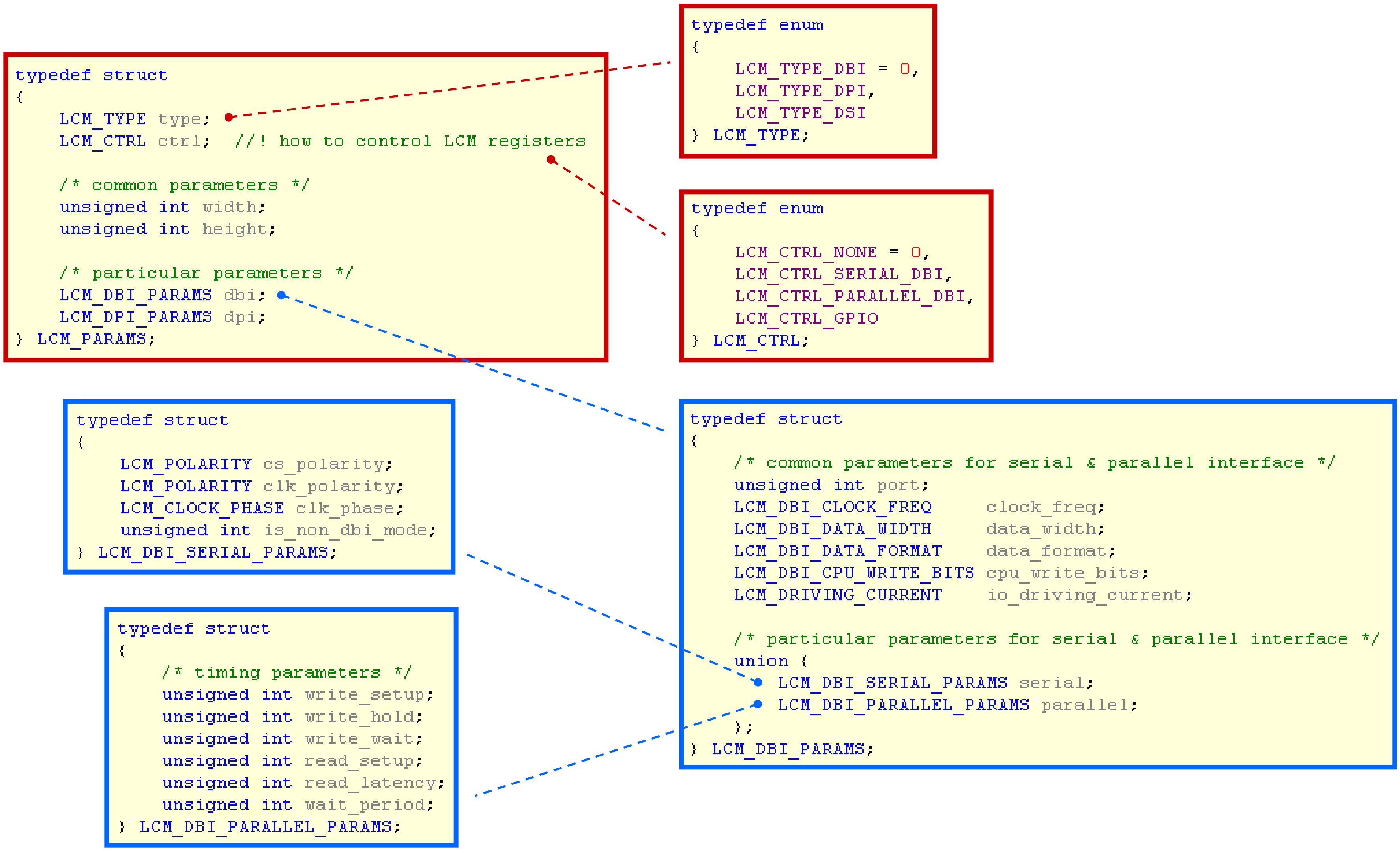
LCM Driver Interface



```
typedef struct
{
    void (*set_util_funcs)(const LCM_UTIL_FUNCS *util);
    void (*get_params)(LCM_PARAMS *params);

    void (*init)(void);
    void (*suspend)(void);
    void (*resume)(void);

    void (*update)(unsigned int x, unsigned int y,
                   unsigned int width, unsigned int height);
} LCM_DRIVER;
```

Name	Description
set_util_funcs	Set LCM utility function interface to LCM driver
get_params	Return LCM parameters for display driver to initialize related HW controllers
init	Initialize the LCM
suspend	Suspend the LCM
resume	Resume the LCM
update	Send the block update commands to LCM

LCM Parameters (Common and DBI)



```
typedef struct
{
    LCM_TYPE type;
    LCM_CTRL ctrl;  //! how to control LCM registers

    /* common parameters */
    unsigned int width;
    unsigned int height;

    /* particular parameters */
    LCM_DBI_PARAMS dbi;
    LCM_DPI_PARAMS dpi;
} LCM_PARAMS;
```

```
typedef enum
{
    LCM_TYPE_DBI = 0,
    LCM_TYPE_DPI,
    LCM_TYPE_DSI
} LCM_TYPE;
```

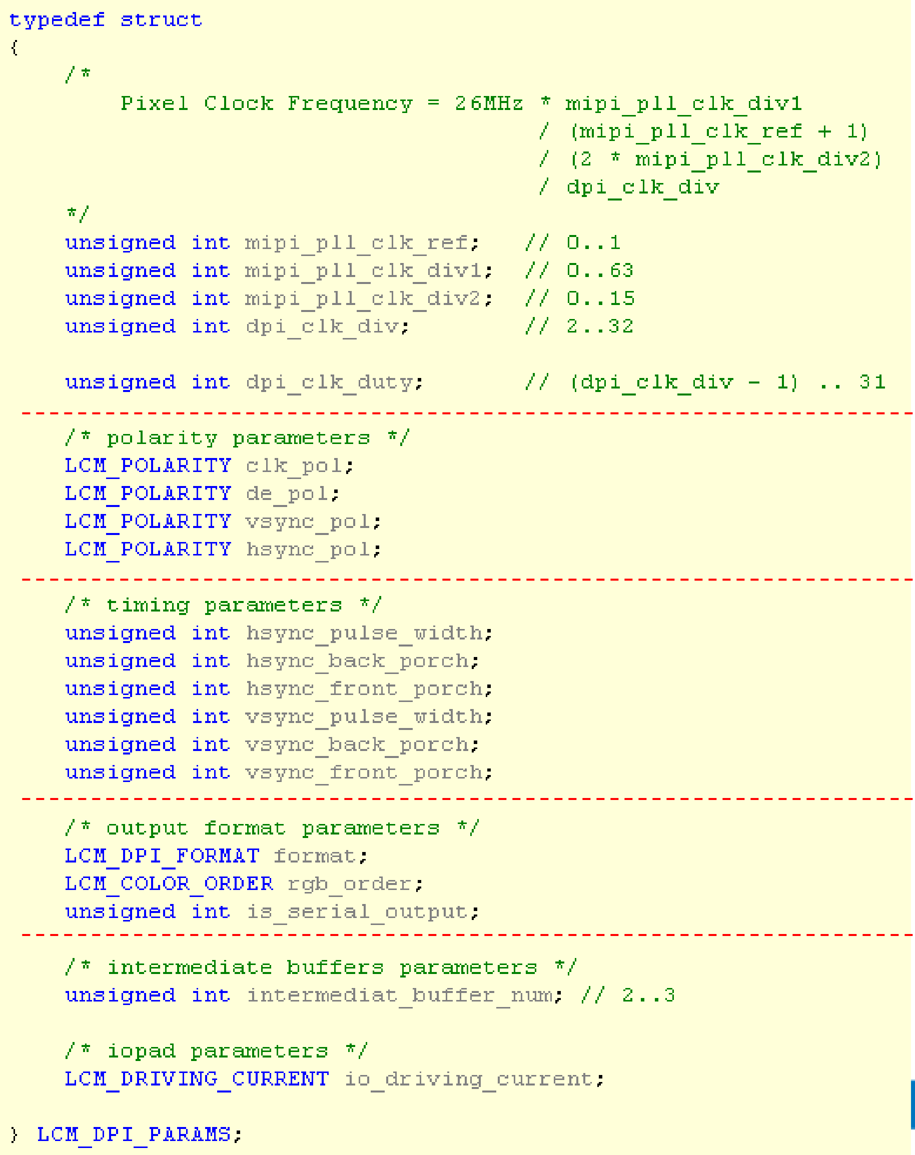
```
typedef enum
{
    LCM_CTRL_NONE = 0,
    LCM_CTRL_SERIAL_DBI,
    LCM_CTRL_PARALLEL_DBI,
    LCM_CTRL_GPIO
} LCM_CTRL;
```

```
typedef struct
{
    /* common parameters for serial & parallel interface */
    unsigned int port;
    LCM_DBI_CLOCK_FREQ     clock_freq;
    LCM_DBI_DATA_WIDTH     data_width;
    LCM_DBI_DATA_FORMAT    data_format;
    LCM_DBI_CPU_WRITE_BITS cpu_write_bits;
    LCM_DRIVING_CURRENT    io_driving_current;

    /* particular parameters for serial & parallel interface */
    union {
        LCM_DBI_SERIAL_PARAMS serial;
        LCM_DBI_PARALLEL_PARAMS parallel;
    };
} LCM_DBI_PARAMS;
```

```
typedef struct
{
    LCM_POLARITY cs_polarity;
    LCM_POLARITY clk_polarity;
    LCM_CLOCK_PHASE clk_phase;
    unsigned int is_non_dbi_mode;
} LCM_DBI_SERIAL_PARAMS;
```

```
typedef struct
{
    /* timing parameters */
    unsigned int write_setup;
    unsigned int write_hold;
    unsigned int write_wait;
    unsigned int read_setup;
    unsigned int read_latency;
    unsigned int wait_period;
} LCM_DBI_PARALLEL_PARAMS;
```

LCM Parameters (DPI)



```
typedef struct
{
    /*
        Pixel Clock Frequency = 26MHz * mipi_pll_clk_div1
                                      / (mipi_pll_clk_ref + 1)
                                      / (2 * mipi_pll_clk_div2)
                                      / dpi_clk_div
    */
    unsigned int mipi_pll_clk_ref;   // 0..1
    unsigned int mipi_pll_clk_div1;  // 0..63
    unsigned int mipi_pll_clk_div2;  // 0..15
    unsigned int dpi_clk_div;        // 2..32

    unsigned int dpi_clk_duty;       // (dpi_clk_div - 1) .. 31

    /* polarity parameters */
    LCM_POLARITY clk_pol;
    LCM_POLARITY de_pol;
    LCM_POLARITY vsync_pol;
    LCM_POLARITY hsync_pol;

    /* timing parameters */
    unsigned int hsync_pulse_width;
    unsigned int hsync_back_porch;
    unsigned int hsync_front_porch;
    unsigned int vsync_pulse_width;
    unsigned int vsync_back_porch;
    unsigned int vsync_front_porch;

    /* output format parameters */
    LCM_DPI_FORMAT format;
    LCM_COLOR_ORDER rgb_order;
    unsigned int is_serial_output;

    /* intermediate buffers parameters */
    unsigned int intermediat_buffer_num; // 2..3

    /* iopad parameters */
    LCM_DRIVING_CURRENT io_driving_current;

} LCM_DPI_PARAMS;
```

pixel clock frequency

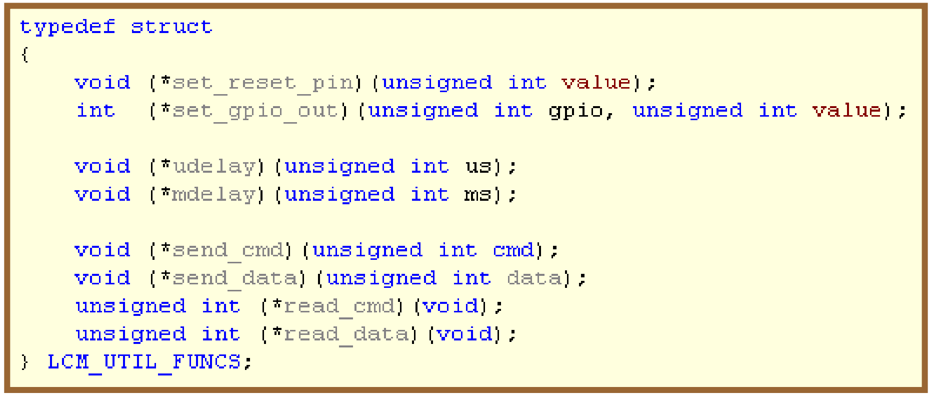
polarity

blanking timing

output color format

misc.

LCM Utility Function Interface

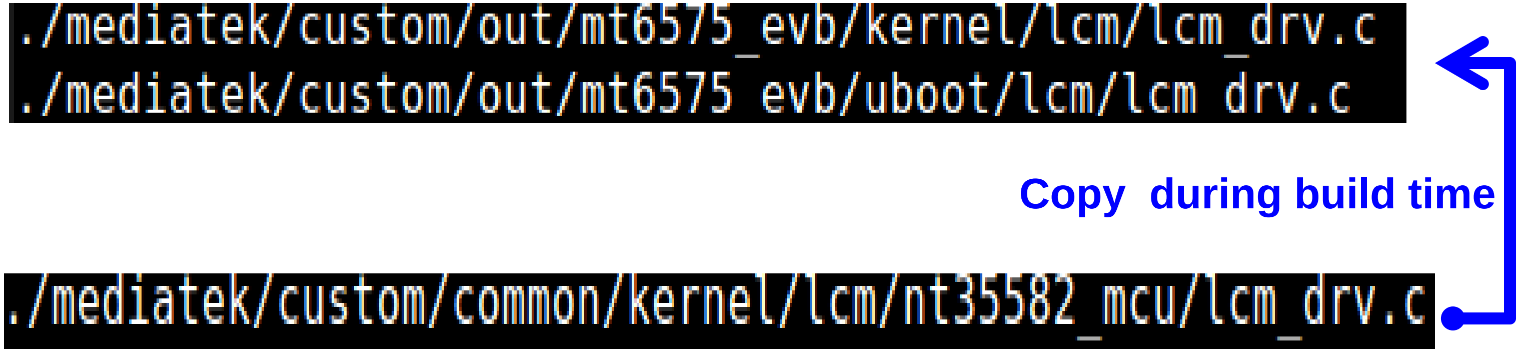


```
typedef struct
{
    void (*set_reset_pin)(unsigned int value);
    int  (*set_gpio_out)(unsigned int gpio, unsigned int value);

    void (*udelay)(unsigned int us);
    void (*mdelay)(unsigned int ms);

    void (*send_cmd)(unsigned int cmd);
    void (*send_data)(unsigned int data);
    unsigned int (*read_cmd)(void);
    unsigned int (*read_data)(void);
} LCM_UTIL_FUNCS;
```

Name	Description
set_reset_pin	Output value to the LCM reset pin
set_gpio_out	Output value to the specified GPIO pin
udelay	Delay several microseconds
mdelay	Delay several milliseconds
send_cmd	Write command to the LCM
send_data	Write data to the LCM
read_cmd	Read command from the LCM
read_data	Read data from the LCM

LCM Customer Folder

- Put all LCM drivers in the custom common kernel folder
- Select LCM by modifying project make file
  - e.g. ./mediatek/config/mt6575_evb/ProjectConfig.mk
    CUSTOM_KERNEL_LCM = nt35582_mcu



```
./mediatek/custom/out/mt6575_evb/kernel/lcm/lcm_drv.c
./mediatek/custom/out/mt6575 evb/uboot/lcm/lcm drv.c
```

Copy during build time

```
./mediatek/custom/common/kernel/lcm/nt35582_mcu/lcm_drv.c
```

MEDIATEK

Sensor System

Outline

- Sensor Hal
- Sensor Driver Customization

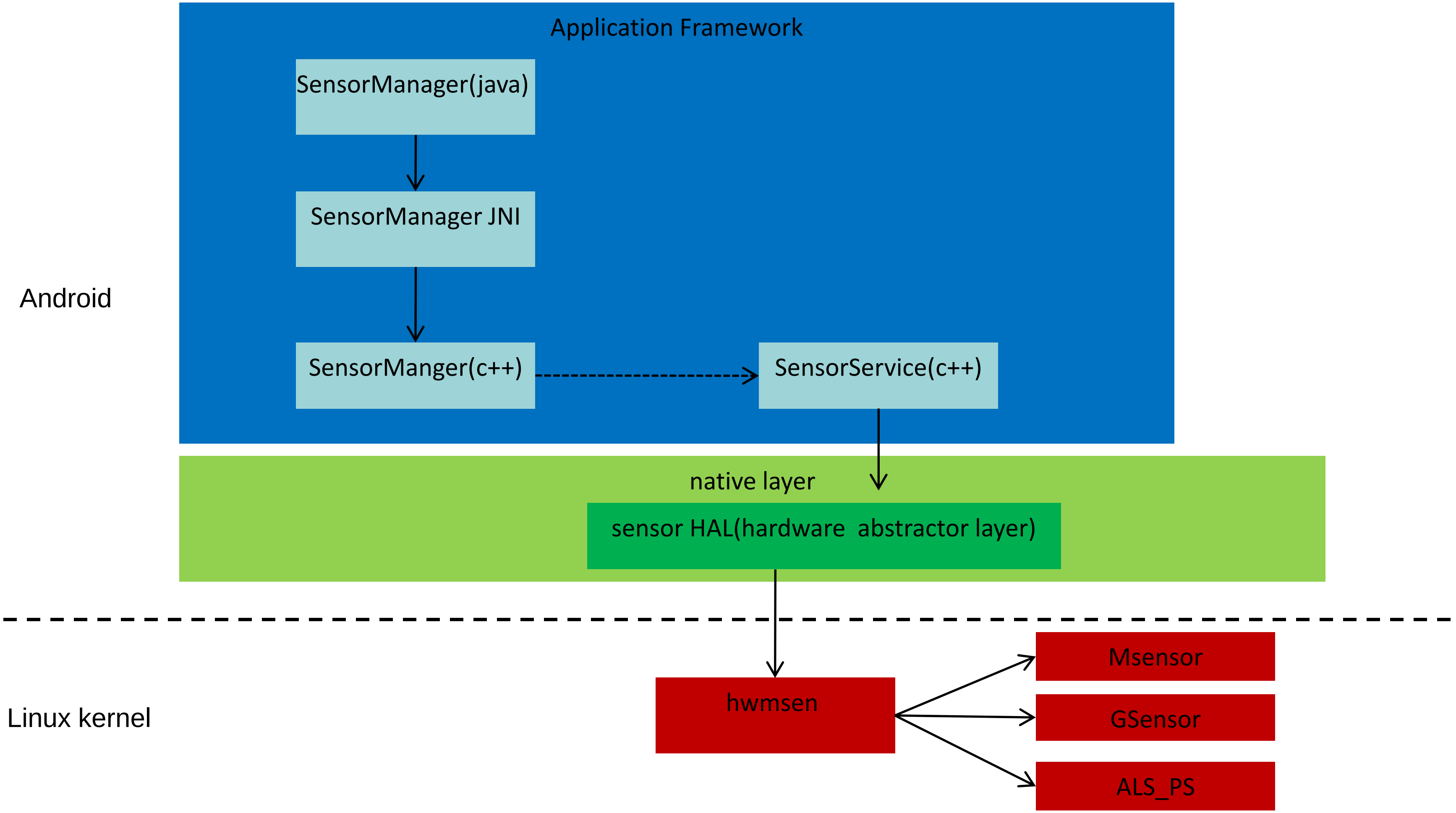
Android sensor support types

- Now Android support 8 types sensors.

Sensor types	Service define	Driver define
Accelerometer	TYPE_ACCELEROMETER	SENSOR_TYPE_ACCELEROMETER
Magnetic	TYPE_MAGNETIC_FIELD	SENSOR_TYPE_MAGNETIC_FIELD
Orientation	TYPE_ORIENTATION	SENSOR_TYPE_ORIENTATION
Gyroscope	TYPE_GYROSCOPE	SENSOR_TYPE_GYROSCOPE
Light	TYPE_LIGHT	SENSOR_TYPE_LIGHT
Pressure	TYPE_PRESSURE	SENSOR_TYPE_PRESSURE
Temperature	TYPE_TEMPERATURE	SENSOR_TYPE_TEMPERATURE
Proximity	TYPE_PROXIMITY	SENSOR_TYPE_PROXIMITY

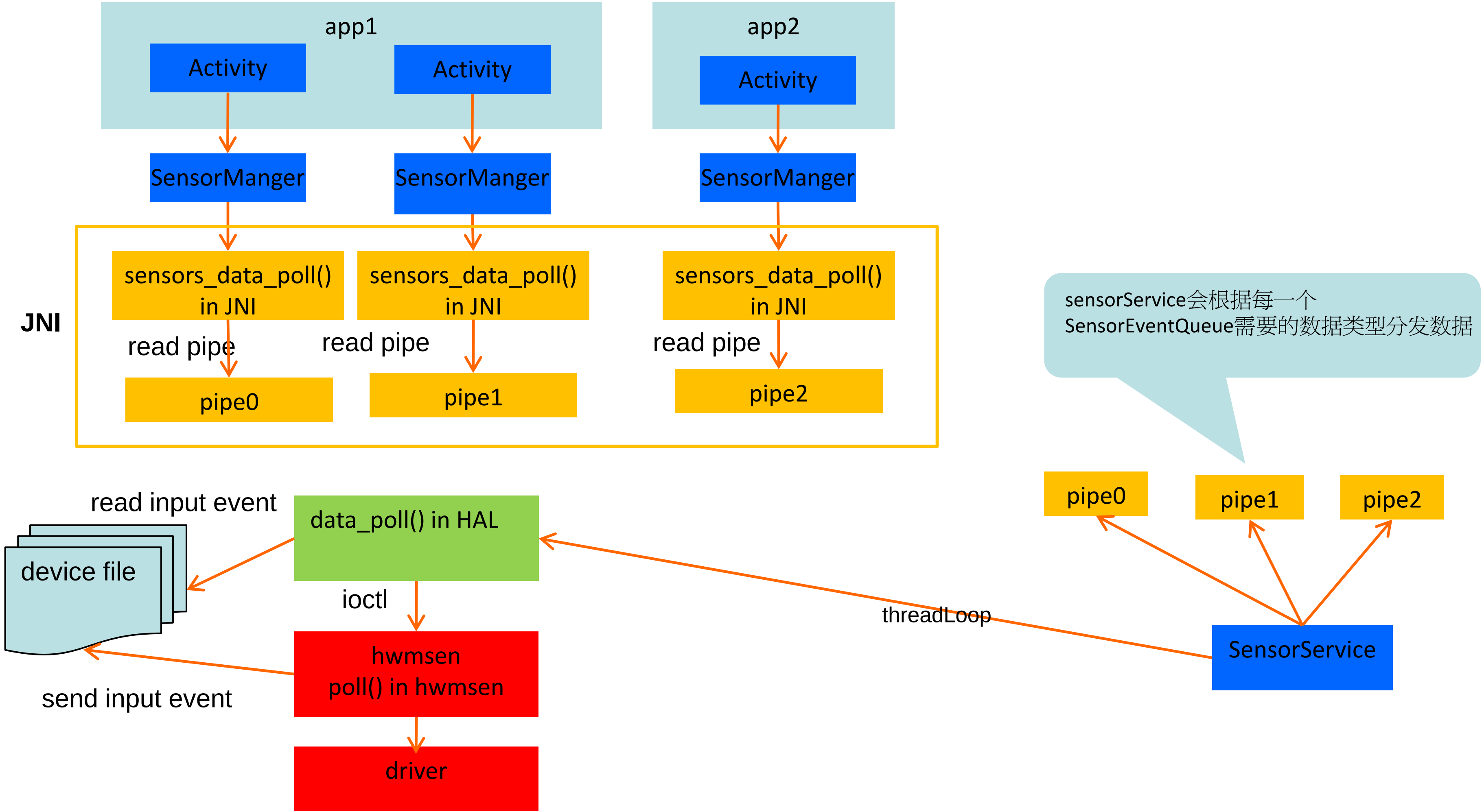
Sensor system Architecture

- sensor Architecture (android 2.3)



Sensor Manager

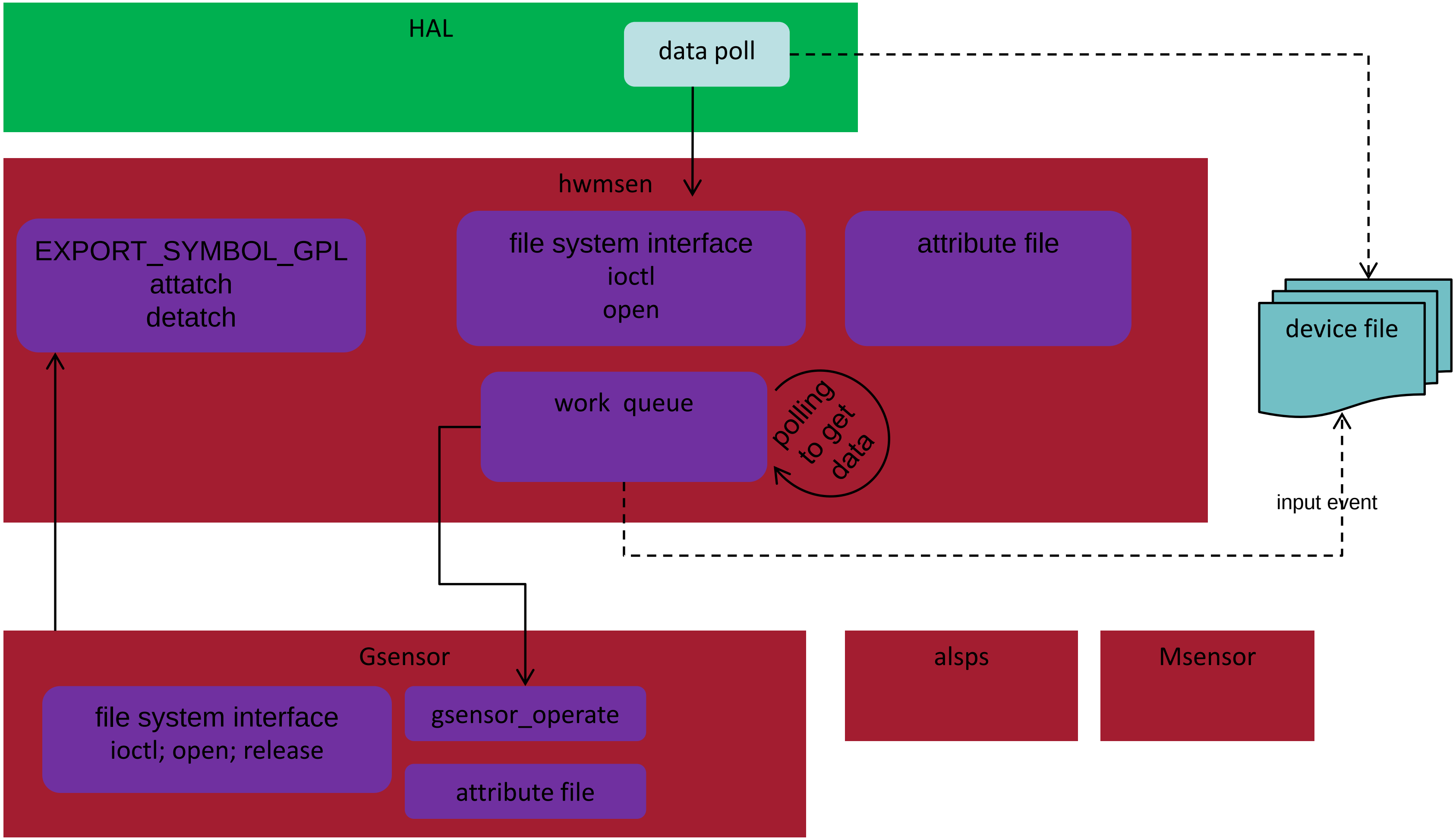
- JNI(android2.3)



sensorService会根据每一个SensorEventQueue需要的数据类型分发数据

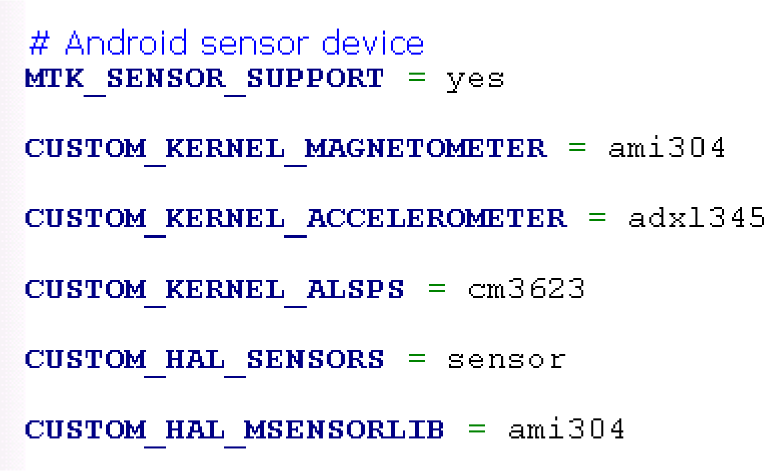
Hwmsen driver(1/5)

- architecture



Sensor HAL Customization

- Makefile Customization
  – Alps/mediatek/config/$(project)/ProjectConfig.mk file set the sensors' configure



```
# Android sensor device
MTK_SENSOR_SUPPORT = yes

CUSTOM_KERNEL_MAGNETOMETER = ami304

CUSTOM_KERNEL_ACCELEROMETER = adxl345

CUSTOM_KERNEL_ALSPS = cm3623

CUSTOM_HAL_SENSORS = sensor

CUSTOM_HAL_MSENSORLIB = ami304
```

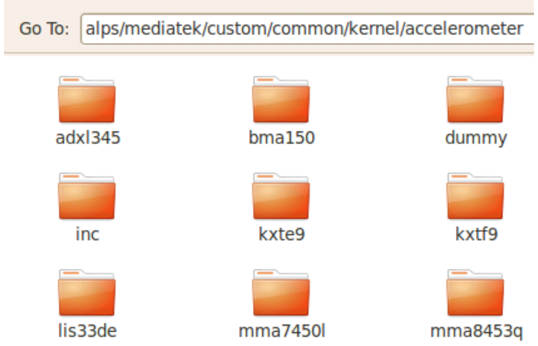
  – If you want to support sensor in your project, please always set

    MTK_SENSOR_SUPPORT = yes

    CUSTOM_HAL_SENSORS = sensor

Sensor HAL Customization

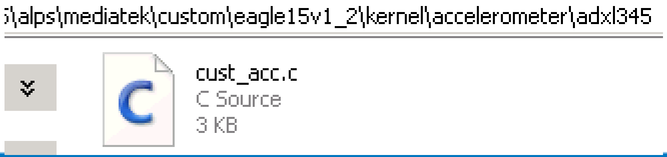
- G sensor driver Customization
  - If project use g sensor adxl345, please set
    - CUSTOM_KERNEL_ACCELEROMETER = adxl345
  - If have no g sensor, set as follow
    - CUSTOM_KERNEL_ACCELEROMETER =
  - G sensor driver is location at
    - G sensor driver



Go To: alps/mediatek/custom/common/kernel/accelerometer

adxl345 bma150 dummy
inc kxte9 kxtf9
lis33de mma7450l mma8453q

    - Customization file

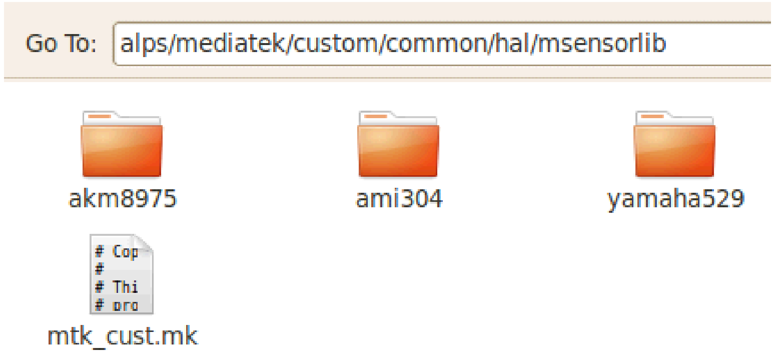


5\alps\mediatek\custom\eagle15v1_2\kernel\accelerometer\adxl345

cust_acc.c
C Source
3 KB

Sensor HAL Customization

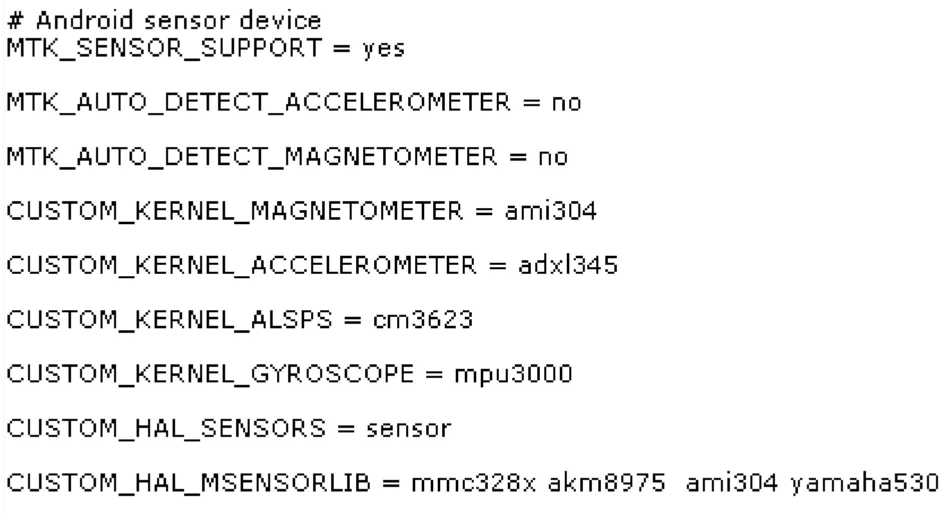
- M sensor driver Customization
  - If project use m sensor ami304, please set
    - CUSTOM_KERNEL_MAGNETOMETER = ami304
    - CUSTOM_HAL_MSENSORLIB = ami304
  - If have no g sensor, set as follow
    - CUSTOM_KERNEL_MAGNETOMETER =
    - CUSTOM_HAL_MSENSORLIB =
  - M-sensor daemon source code



Go To: alps/mediatek/custom/common/hal/msensorlib

Sensor HAL Customization

- Makefile Customization (auto-detect)
  - alps/mediatek/config/$(project)/ProjectConfig.mk file set the sensors' configure

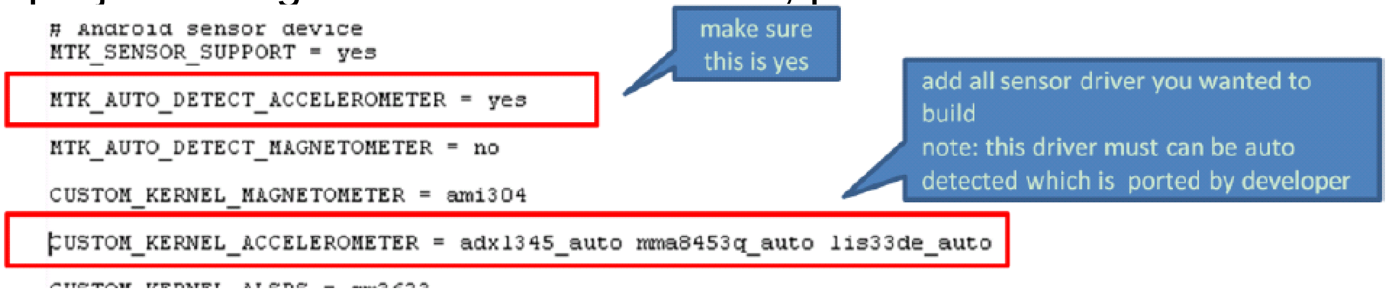


```
# Android sensor device
MTK_SENSOR_SUPPORT = yes

MTK_AUTO_DETECT_ACCELEROMETER = no

MTK_AUTO_DETECT_MAGNETOMETER = no

CUSTOM_KERNEL_MAGNETOMETER = ami304

CUSTOM_KERNEL_ACCELEROMETER = adxl345

CUSTOM_KERNEL_ALSPS = cm3623

CUSTOM_KERNEL_GYROSCOPE = mpu3000

CUSTOM_HAL_SENSORS = sensor

CUSTOM_HAL_MSENSORLIB = mmc328x akm8975  ami304 yamaha530
```

  - If you want to support sensor in your project, please always set
    MTK_SENSOR_SUPPORT = yes
    CUSTOM_HAL_SENSORS = sensor

Sensor HAL Customization

- G sensor driver Customization (auto-detect)
  - If project use g sensor more than one, please set

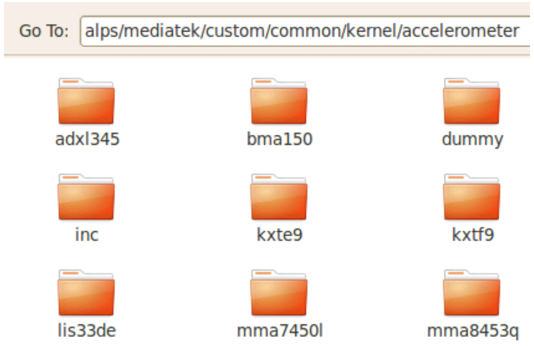


```
# Android sensor device
MTK_SENSOR_SUPPORT = yes
MTK_AUTO_DETECT_ACCELEROMETER = yes
MTK_AUTO_DETECT_MAGNETOMETER = no
CUSTOM_KERNEL_MAGNETOMETER = ami304
CUSTOM_KERNEL_ACCELEROMETER = adxl345_auto mma8453q_auto lis33de_auto
CUSTOM_KERNEL_ALSPS = cm3623
```

make sure this is yes

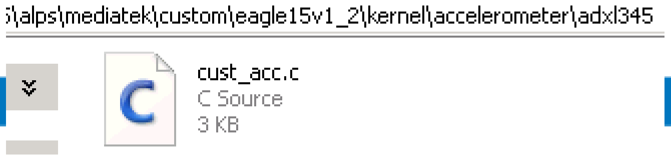
add all sensor driver you wanted to build
note: this driver must can be auto detected which is ported by developer

  - If have no g sensor, set as follow
    - CUSTOM_KERNEL_ACCELEROMETER =
  - G sensor driver is location at
    - G sensor driver



Go To: alps/mediatek/custom/common/kernel/accelerometer

    - Customization file

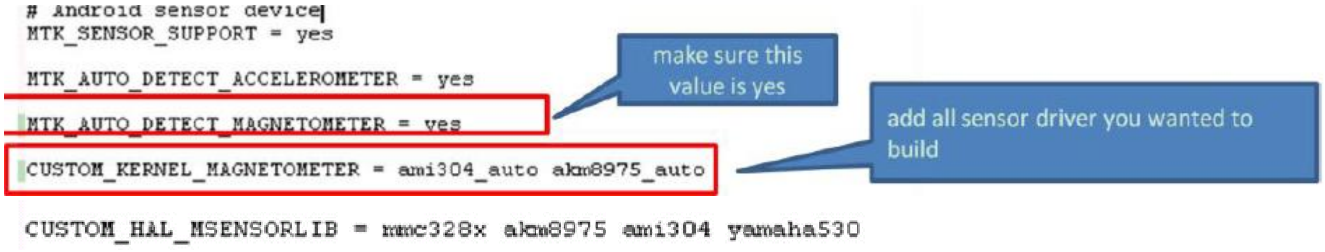


5\alps\mediatek\custom\eagle15v1_2\kernel\accelerometer\adxl345

cust_acc.c
C Source
3 KB

Sensor HAL Customization

- M sensor driver Customization (auto-detect)
  - If project use more than one m-sensor, please set

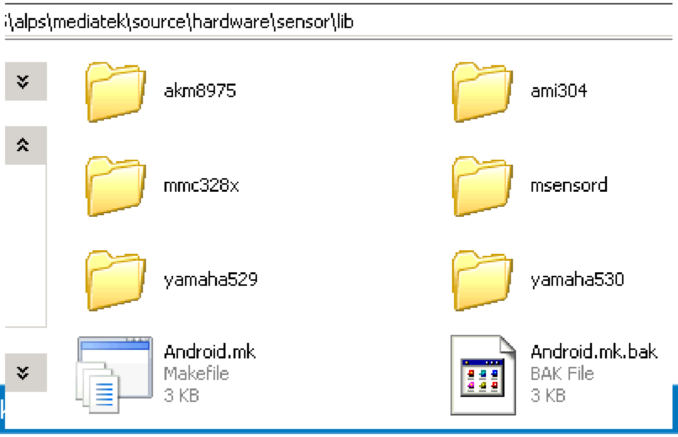


```
# Android sensor device
MTK_SENSOR_SUPPORT = yes

MTK_AUTO_DETECT_ACCELEROMETER = yes

MTK_AUTO_DETECT_MAGNETOMETER = yes

CUSTOM_KERNEL_MAGNETOMETER = ami304_auto akm8975_auto

CUSTOM_HAL_MSENSORLIB = mmc328x akm8975 ami304 yamaha530
```

make sure this value is yes

add all sensor driver you wanted to build

  - If have no g sensor, set as follow
    - CUSTOM_KERNEL_MAGNETOMETER =
    - CUSTOM_HAL_MSENSORLIB =
  - M-sensor daemon source code

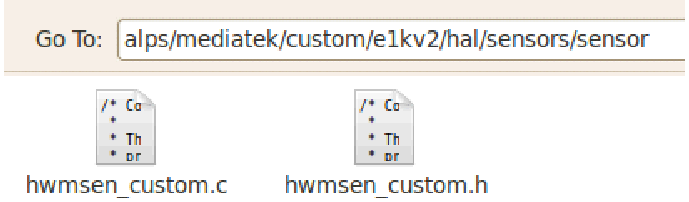


```
:\alps\mediatek\source\hardware\sensor\lib
```

akm8975
ami304
mmc328x
msensord
yamaha529
yamaha530
Android.mk
Makefile
3 KB
Android.mk.bak
BAK File
3 KB

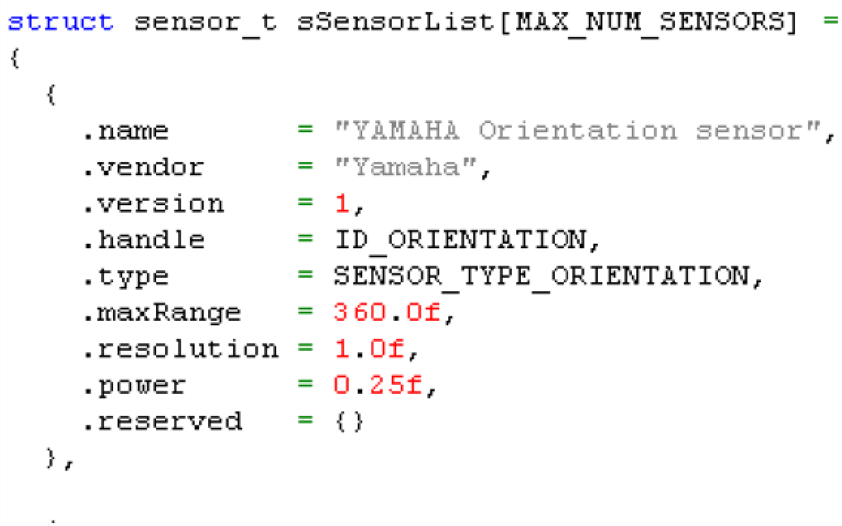
Sensor HAL Customization

- Sensor Hal Customization
  - Alps/mediatek/custom/$(project)/hal/sensors/sensor folder have project customization configure file



Go To: alps/mediatek/custom/e1kv2/hal/sensors/sensor

  - Customization the detail information in hwmsens_custom.c, for example



```
struct sensor_t sSensorList[MAX_NUM_SENSORS] =
{
  {
    .name       = "YAMAHA Orientation sensor",
    .vendor     = "Yamaha",
    .version    = 1,
    .handle     = ID_ORIENTATION,
    .type       = SENSOR_TYPE_ORIENTATION,
    .maxRange   = 360.0f,
    .resolution = 1.0f,
    .power      = 0.25f,
    .reserved   = {}
  },
```

Outline

- Sensor Hal
- **Sensor Driver Customization**

G-Sensor Customization (1/3)

- Change file list

File Name	Location
cust_acc.h	alps\mediatek\custom\common\kernel\accelerometer\inc
cust_acc.c	alps\mediatek\custom\${BOARD}\kernel\accelerometer\${MODULE}
(G sensor driver)	alps\mediatek\custom\common\kernel\accelerometer\${sensor_name}

- Customization item
  - Overview

```
struct acc_hw {
    int i2c_num;
    int direction;
    int power_id;
    int power_vol;
    int firlen;
};
```

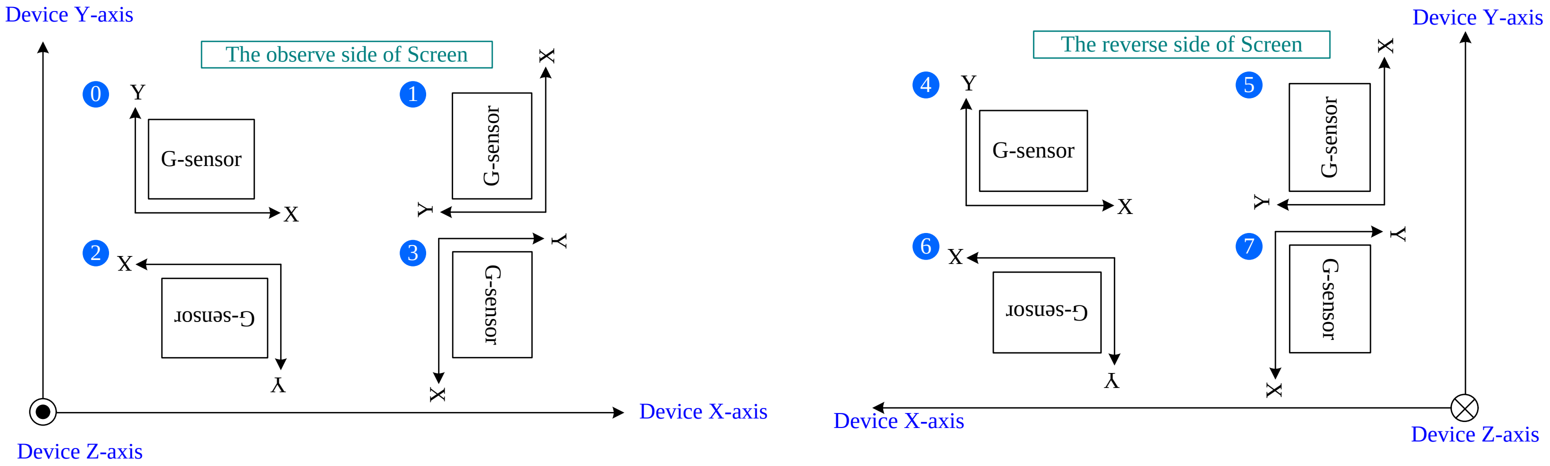
  - i2c_num
    - Customer can define the I2C number used by sensor
    - The value could be defined as 0 ~ 2
  - firlen
    - Customer can define the filter length of SW low pass filter.
    - The value could be defined as 0 ~ 32. 0 will disable the functionality.

G-Sensor Customization (2/3)

– direction

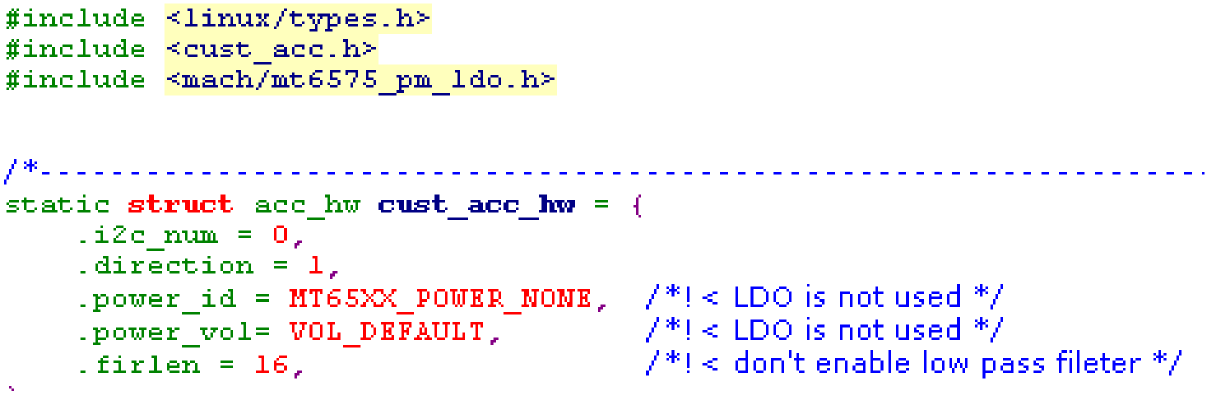
- Customer can define the device direction of g-sensor in device.
- The value could be defined as 0 ~ 7

Value	Description
0	{x, y, z} => { x, y, z}
1	{x, y, z} => {-y, x, z}
2	{x, y, z} => {-x, -y, z}
3	{x, y, z} => { y, -x, z}
4	{x, y, z} => {-x, y, -z}
5	{x, y, z} => { y, x, -z}
6	{x, y, z} => { x, -y, -z}
7	{x, y, z} => {-y, -x, -z}



G-Sensor Customization (3/3)

- power_id / power_vol
  - Customer could define power source of device according to layout
  - Please refer to the following file for power id and voltage
    - alps\mediatek\platform\mt6575\kernel\core\include\mach\mt6575_pm_ldo.h
  - If the power source can’t be shutdown, please set the power_id as MT65XX_POWER_NONE



```
#include <linux/types.h>
#include <cust_acc.h>
#include <mach/mt6575_pm_ldo.h>


/*-----------------------------------------------------------------
static struct acc_hw cust_acc_hw = {
    .i2c_num = 0,
    .direction = 1,
    .power_id = MT65XX_POWER_NONE,  /*!< LDO is not used */
    .power_vol= VOL_DEFAULT,        /*!< LDO is not used */
    .firlen = 16,                   /*!< don't enable low pass fileter */
```

M-Sensor Customization (1/3)

- Change file list

File Name	Location
cust_mag.h	alps\mediatek\custom\common\kernel\magnetometer\inc
cust_mag.c	alps\mediatek\custom\${BOARD}\kernel\magnetometer\${MODULE}

- Customization item
  - Overview

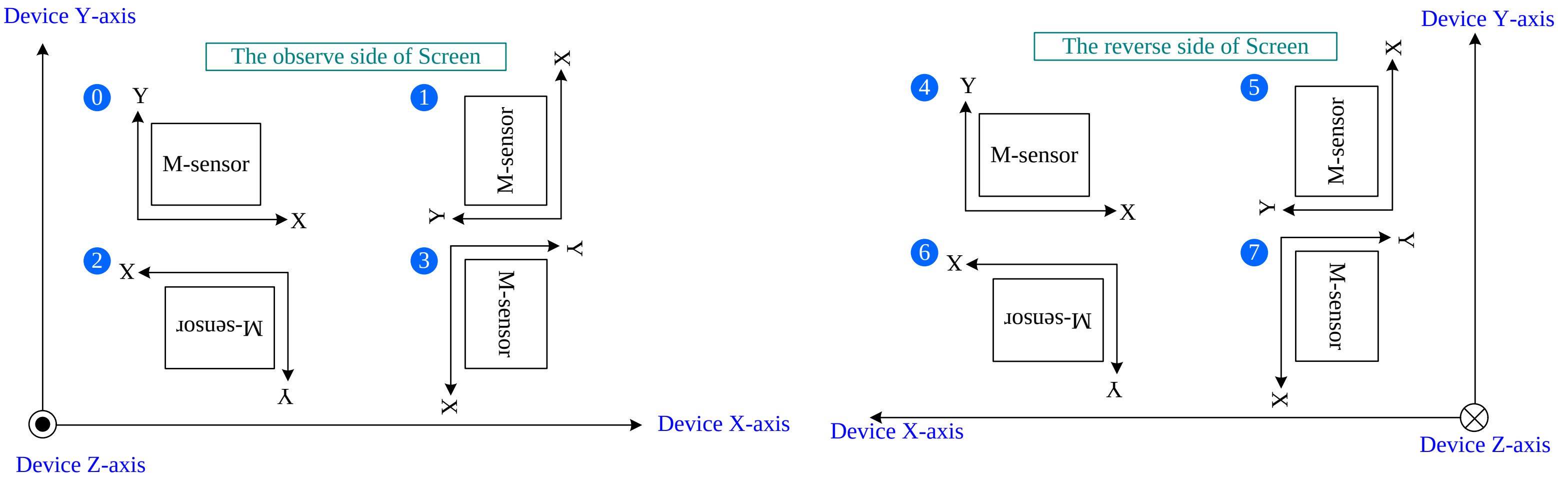
```
struct mag_hw {
    int i2c_num;
    int direction;
    int power_id;
    int power_vol;
};
```

  - i2c_num
    - Customer can define the I2C number used by sensor
    - The value could be defined as 0 ~ 2

M-Sensor Customization (2/3)

– direction
- Customer can define the device direction of sensor in device.
- The value could be defined as 0 ~ 7

Value	Description
0	{x, y, z} => { x, y, z}
1	{x, y, z} => {-y, x, z}
2	{x, y, z} => {-x, -y, z}
3	{x, y, z} => { y, -x, z}
4	{x, y, z} => {-x, y, -z}
5	{x, y, z} => { y, x, -z}
6	{x, y, z} => { x, -y, -z}
7	{x, y, z} => {-y, -x, -z}



M-Sensor Customization (3/3)

- power_id / power_vol
  - Customer could define power source of device according to layout
  - Please refer to the following file for power id and voltage
    - alps\mediatek\platform\mt6575\kernel\core\include\mach\mt6575_pm_ldo.h
  - If the power source can’t be shutdown, please set the power_id as MT65XX_POWER_NONE

ALS/PS Customization (1/4)

- Change file list

File Name	Location
cust_alsps.h	alps\mediatek\custom\common\kernel\alsps\inc
cust_alsps.c	alps\mediatek\custom\${BOARD}\kernel\alsps\${MODULE}

- Customization item
  - Overview

```
#define C_CUST_ALS_LEVEL    16
#define C_CUST_I2C_ADDR_NUM 4

struct alsps_hw {
    int i2c_num;
    int power_id;
    int power_vol;
    unsigned char   i2c_addr[C_CUST_I2C_ADDR_NUM];
    unsigned int    als_level[C_CUST_ALS_LEVEL-1];
    unsigned int    als_value[C_CUST_ALS_LEVEL];
    unsigned int    ps_threshold;
};
```

  - i2c_num
    - Customer can define the I2C number used by sensor
    - The value could be defined as 0 ~ 2

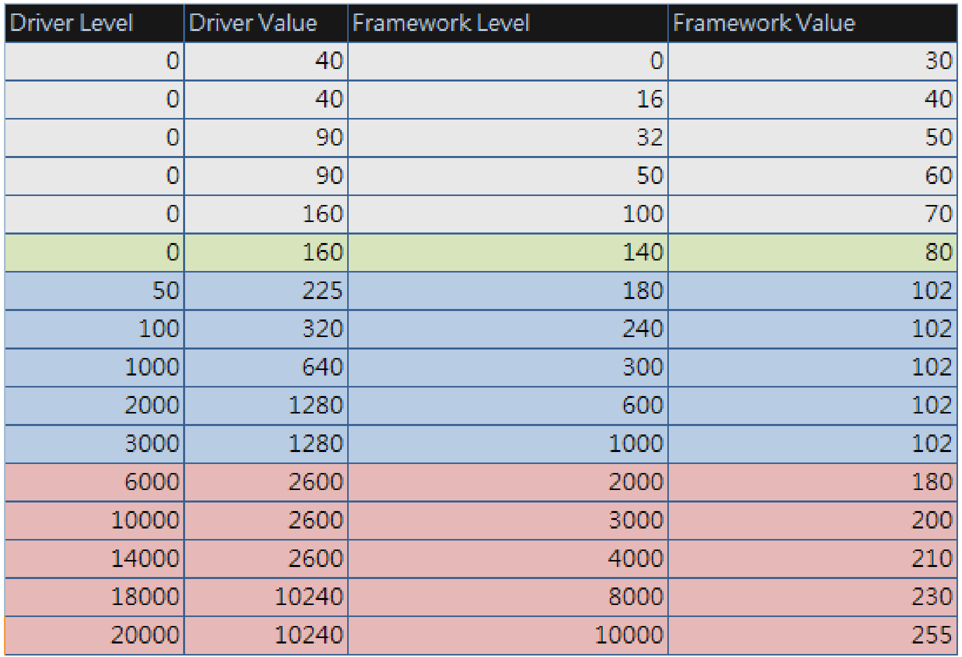
ALS/PS Customization (2/4)

- power_id / power_vol
  - Customer could define power source of device according to layout
  - Please refer to the following file for power id and voltage
    - alps\mediatek\platform\mt6575\kernel\core\include\mach\mt6575_pm_ldo.h
  - If the power source can’t be shutdown, please set the power_id as MT65XX_POWER_NONE
- i2c_addr
  - This is an array of i2c address used in ALS+PS sensor.
    - Some component (CM3623) owns more than one i2c address
- ps_threshold
  - The threshold is used to judge if object is close or not.
  - If the value reported by proximity sensor is larger than **ps_threshold**, it means the object is close. Otherwise, the object is far away.
  - The actual value range depends on each sensor

ALS/PS Customization (3/4)

– als_level & als_value

- The two items will remap the raw data to range 0.0 ~ 10240.0.
- The (C_CUST_ALS_LEVEL-1) values in als_level will divide [0.0 10240.0] into C_CUST_ALS_LEVEL zones. The values in als_value will be reported if the raw data falls into the corresponding zones.
- The framework will use the remapped value to adjust screen backlight/keypad/button backlight



Driver Level	Driver Value	Framework Level	Framework Value
0	40	0	30
0	40	16	40
0	90	32	50
0	90	50	60
0	160	100	70
0	160	140	80
50	225	180	102
100	320	240	102
1000	640	300	102
2000	1280	600	102
3000	1280	1000	102
6000	2600	2000	180
10000	2600	3000	200
14000	2600	4000	210
18000	10240	8000	230
20000	10240	10000	255

An example of als_value & als_level

ALS/PS Customization (4/4)

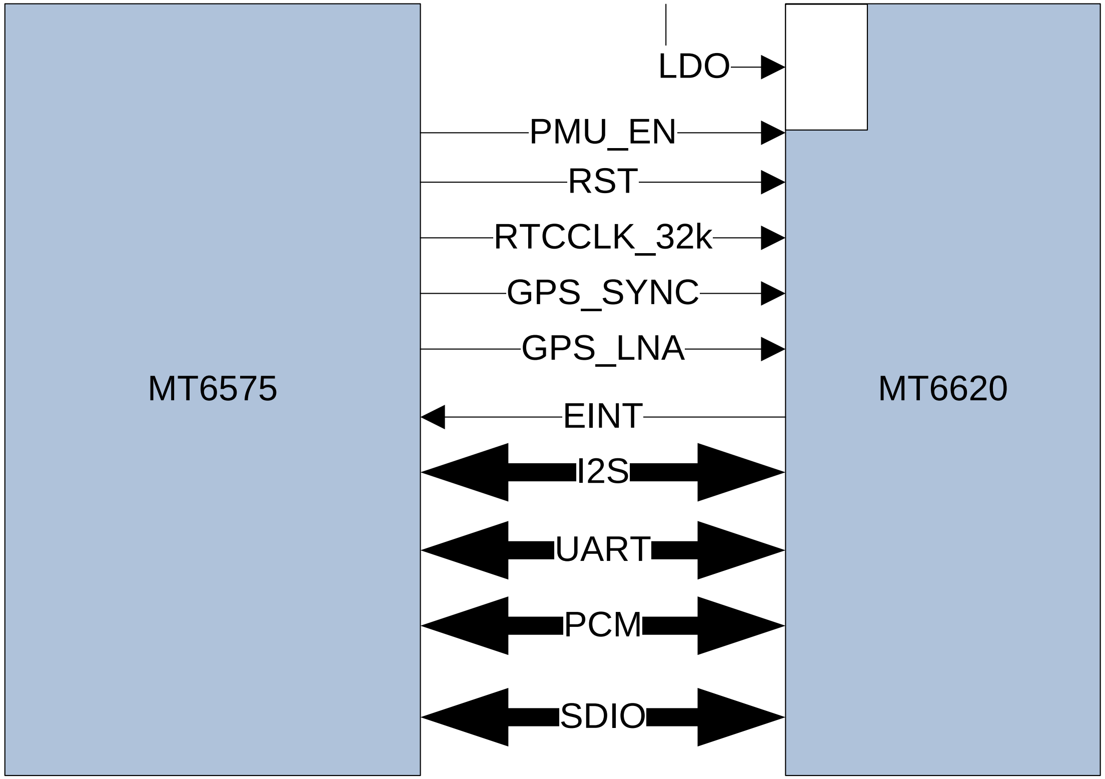
- DCT Customization

DCT definition	Description
GPIO_ALS_EINT_PIN	The GPIO pin for ALS EINT (external interrupt)
CUST_EINT_ALS_NUM	The ID of ALS EINT
CUST_EINT_ALS_DEBOUNCE_CN	The debounce count of ALS. It's set as **0x00** for CM3623
CUST_EINT_ALS_POLARITY	The polarity of ALS. It's set as **low leve**l for CM3623
CUST_EINT_ALS_SENSITIVE	The sensitivity of ALS. It's set as **level sensitive** for CM3623
CUST_EINT_ALS_DEBOUNCE_EN	Enable / disable the debounce. It's set as **disable** for CM3623

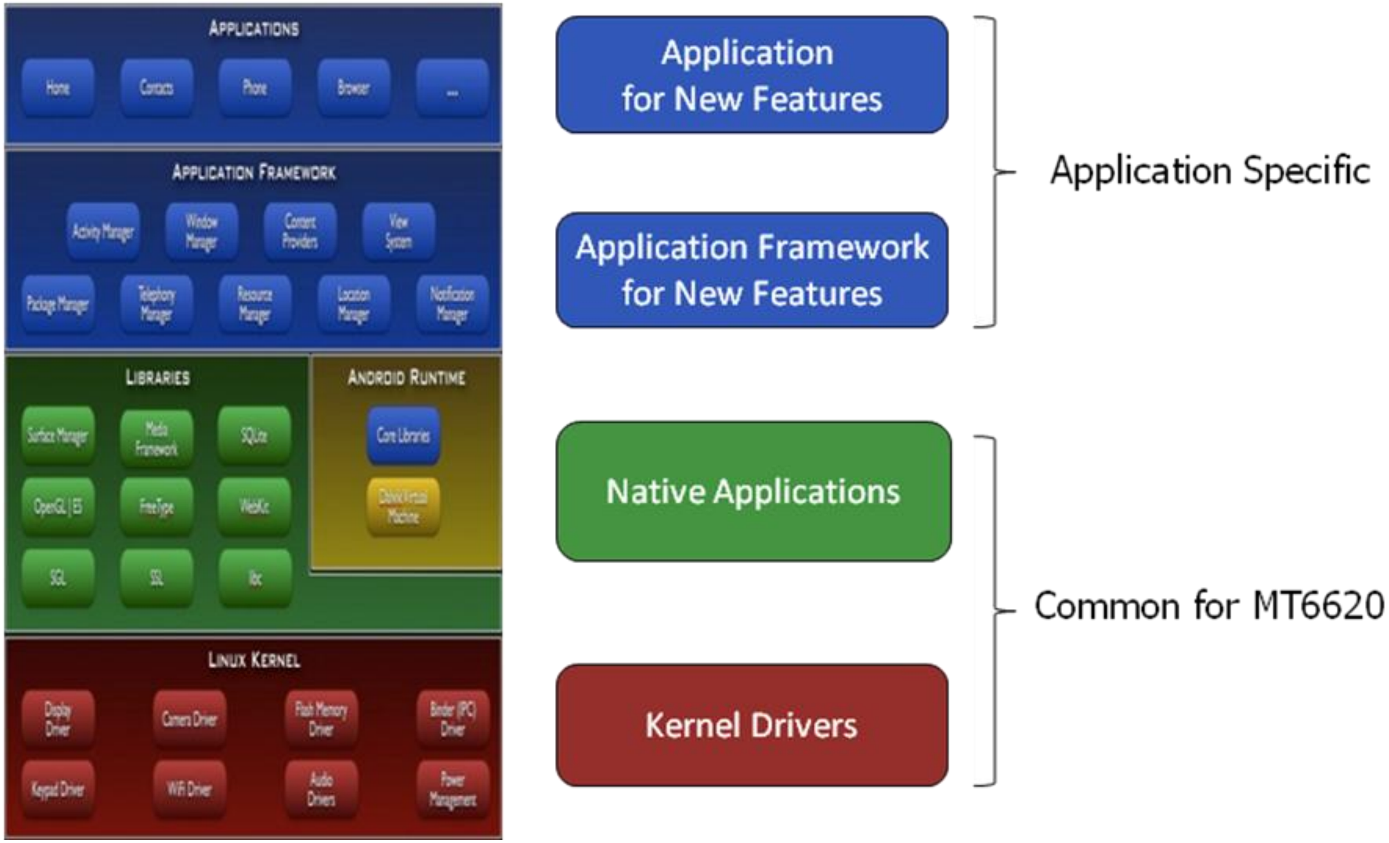
Connectivity

MT6620 Hardware Environment

- MT6620= BT+WIFI+GPS+FM



MT6620 SW Architecture



Application for New Features

Application Framework for New Features

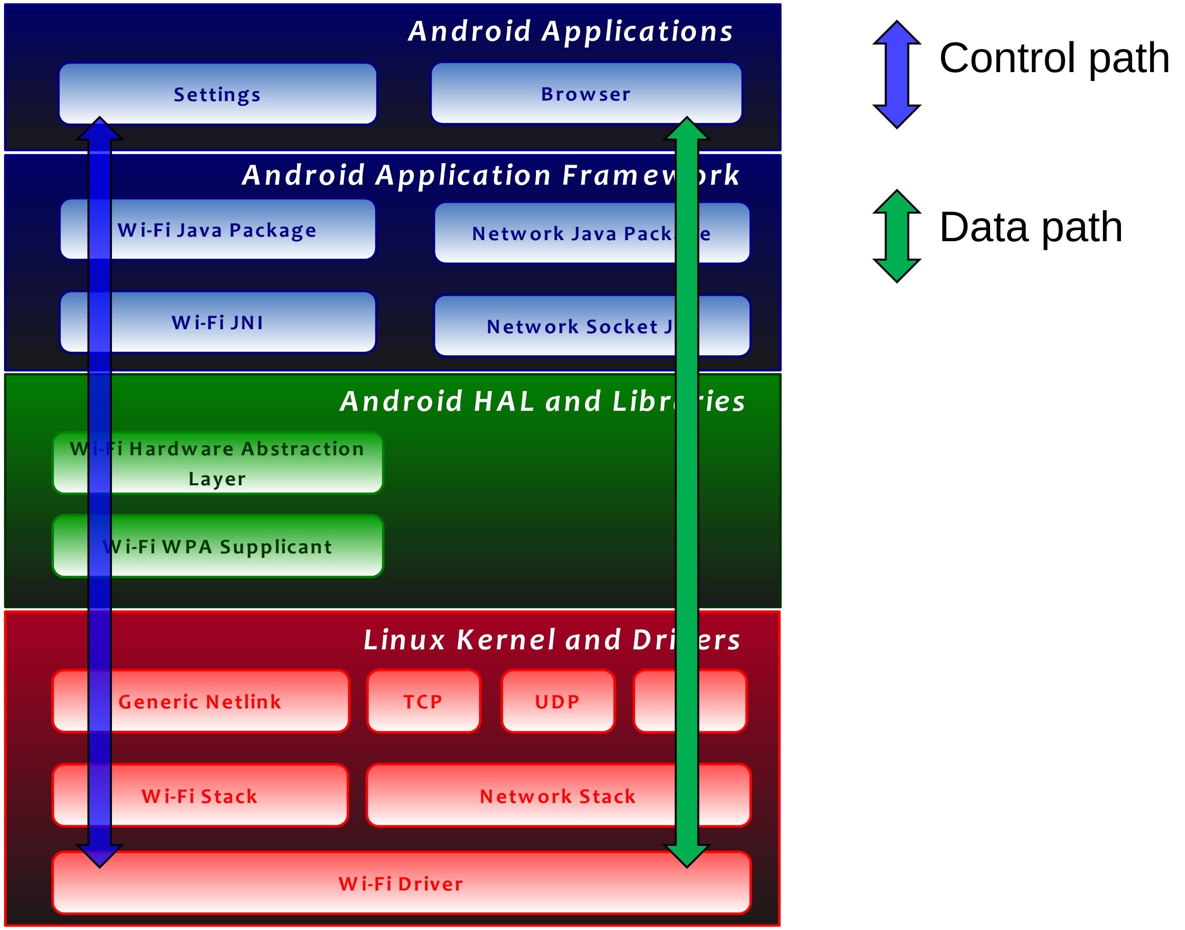
Application Specific

Native Applications

Kernel Drivers

Common for MT6620

Wi-Fi Architecture



Android Applications

Settings

Browser

Android Application Framew ork

Wi-Fi Java Package

Network Java Pack e

Wi-Fi JNI

Network Socket J

Android HAL and Libr ries

Wi-Fi Hardware Abstraction Layer

Wi-Fi WPA Supplicant

Linux Kernel and Dri ers

Generic Netlink

TCP

UDP

Wi-Fi Stack

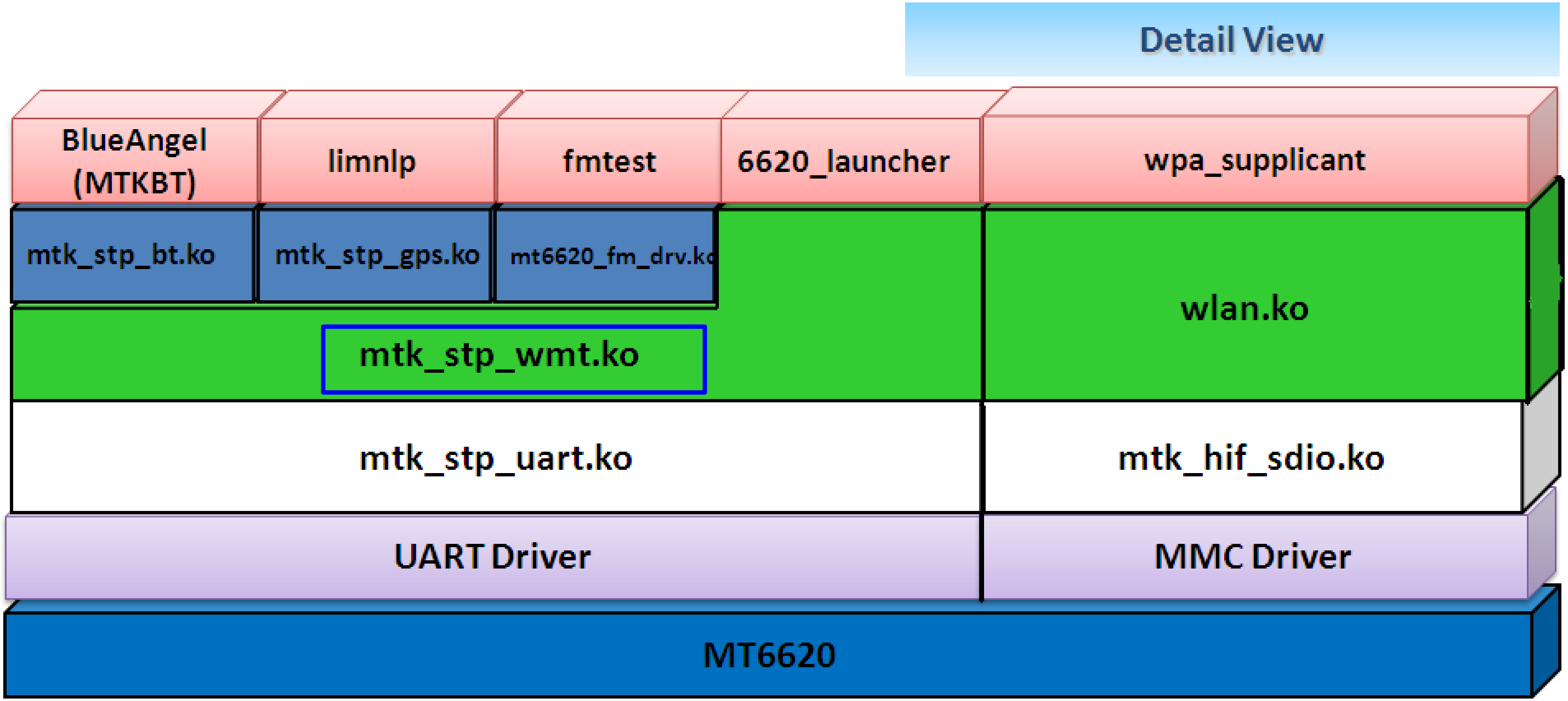
Network Stack

Wi-Fi Driver

Control path

Data path

Detail Architecture



GPIO pins customization

- LDO control pin
  - GPIO_COMBO_6620_LDO_EN_PIN
- UART
  - GPIO_UART_UTXD3_PIN
  - GPIO_UART_URXD3_PIN
- PCM: (for BT)
  - GPIO_PCM_DAICLK_PIN
  - GPIO_PCM_DAIPCMOUT_PIN
  - GPIO_PCM_DAIPCMIN_PIN
  - GPIO_PCM_DAISYNC_PIN
- External interrupt
  - GPIO_COMBO_BGF_EINT_PIN
  - GPIO_WIFI_EINT_PIN
- Power enable pin and reset pin
  - GPIO_COMBO_PMU_EN_PIN
  - GPIO_COMBO_RST_PIN
- GPS SYNC PIN
  - GPIO_GPS_SYNC_PIN
- GPS LNA PIN
  - GPIO_GPS_LNA_PIN

Bluetooth

- BT firmware configurations
  - stored in NVRAM. Those settings will be set to MT6620 by vendor-specific HCI commands. Those settings store in AP_CFG_RDEB_FILE_BT_ADDR_LID.
- **Bluetooth Address**
- **Bluetooth Voice Configuration**
- **Bluetooth PCM configuration**
- **Bluetooth RF configuration**
- **Bluetooth Sleep mode Configuration**
- **TX Power Channel Offset Compensation**

Compile Option

alps/mediatek/config/<project-name>/ProjectConfig.mk:

```
MTK_BT_SUPPORT=yes
MTK_BT_CHIP = MTK_MT6620
```

GPS (1)

- GPS hardware related settings as follows can be got through NVRAM.
- the default value be stored in alps/mediatek/custom/$(project)/cgen/cfgdefault/CFG_GPS_Default.h

Configuration	Description	Value and comment	
gps_tcxo_type	TCXO clock type	0x00	16.368M integer
		0x01	16.369M integer, generated freq has bias
		0x02	Reserved
		0x03	26M integer, generated freq has bias
		0xFE	Crystal clock, GPS chip utilizes a wide range of clock frequency architecture.
		0xFF	TCXO clock, GPS chip utilizes a wide range of clock frequency architecture, with more power consumption and no frequency bias.
gps_tcxo_hz	TCXO frequency in Hz	26000000	26MHz
gps_tcxo_ppb	TCXO clock drift in ppb	500	0.5ppm
gps_lna_mode	GPS LNA type	0	Mixer in
		1	Internal LNA

GPS (2)

- TCXO settings
  - gps_tcxo_type : TCXO clock type, default is 0xFF
  - gps_tcxo_hz : TCXO frequency, in HZ unit, default is 26000000
  - gps_tcxo_ppb : TCXO drift, in ppb unit, default is 500
  - GPS SW (MNL) could support 0.5ppm and 2.0ppm TCXO. GPS positioning accuracy is the same between 0.5ppm and 2.0 ppm TCXO. But the TTFF of 2.0ppm may be longer than 0.5ppm TCXO under some conditions:
    - Temperature has severe change (over tens of Celsius degree). Or
    - GPS power on after a very long time (several months).
  - If customers decide to use 2.0ppm TCXO, We suggest customers to set gps_tcxo_ppb = 2000 ppb. So that GPS SW can improve the TTFF may be longer problem, and keep GPS has better performance while using 2.0ppm TCXO.
- LNA settings
  - The LNA setting gps_lna_mode is invalid on MT6620 currently. For internal LNA, it needs no change. For external LNA, use MT6620 GPIO to enable LNA on MT6573 platform and use host GPIO on MT6575.

GPS (3)

- Launch MNLD through Property Service
  - MNLD will be launched when the system boot up if you have the following settings.

File	alps/mediatek/config/$(project)/init.rc
Setting	service mnld /system/xbin/mnld socket mnld stream 666 system system disabled

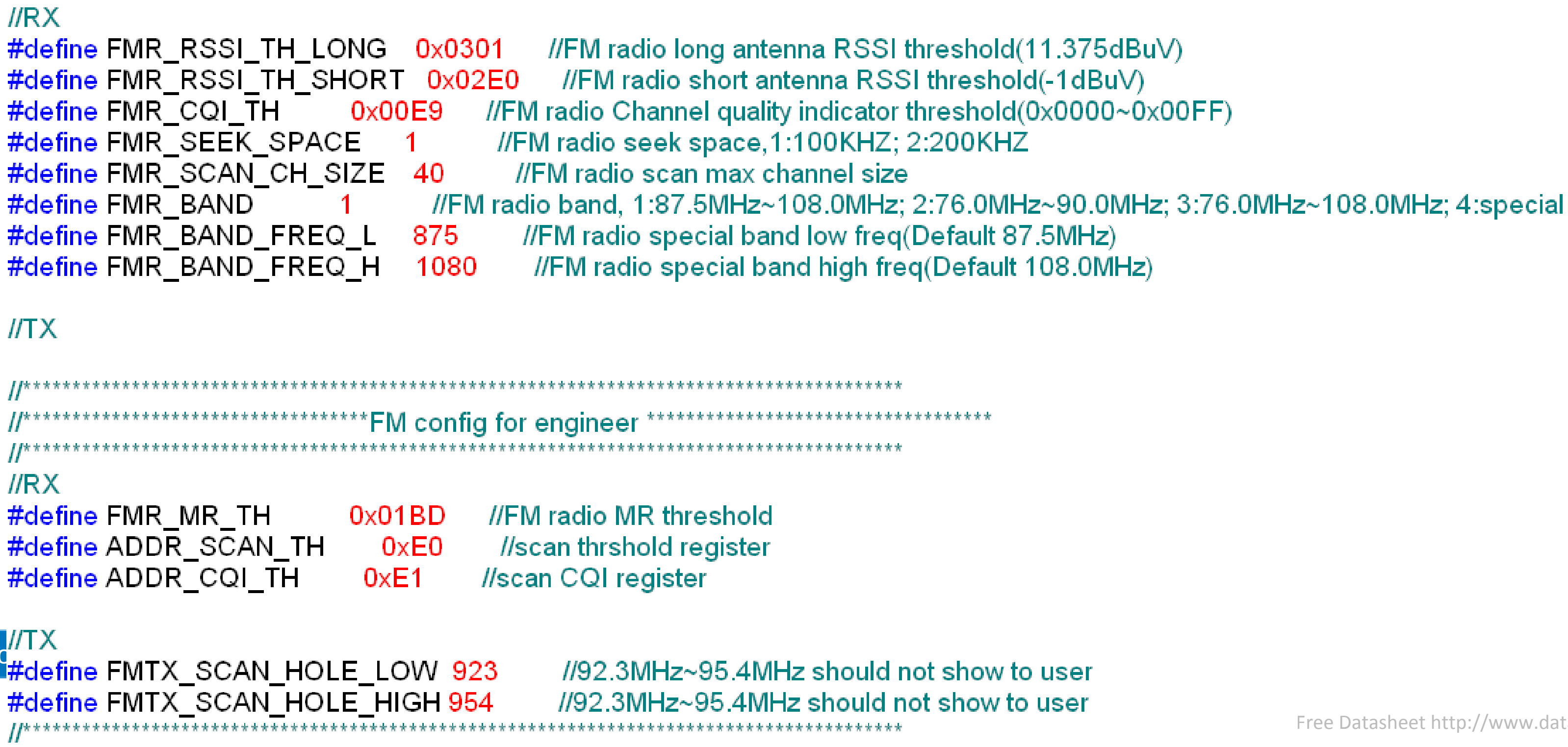
- If you want to open or close GPS on your device, you can change the following settings to achieve this purpose.

File	alps/mediatek/config/$(project)/PoardConfig.mk	
Setting	Open	MTK_GPS_SUPPORT =yes

	Close	MTK_GPS_SUPPORT=no
		MTK_AGPS_APP=no

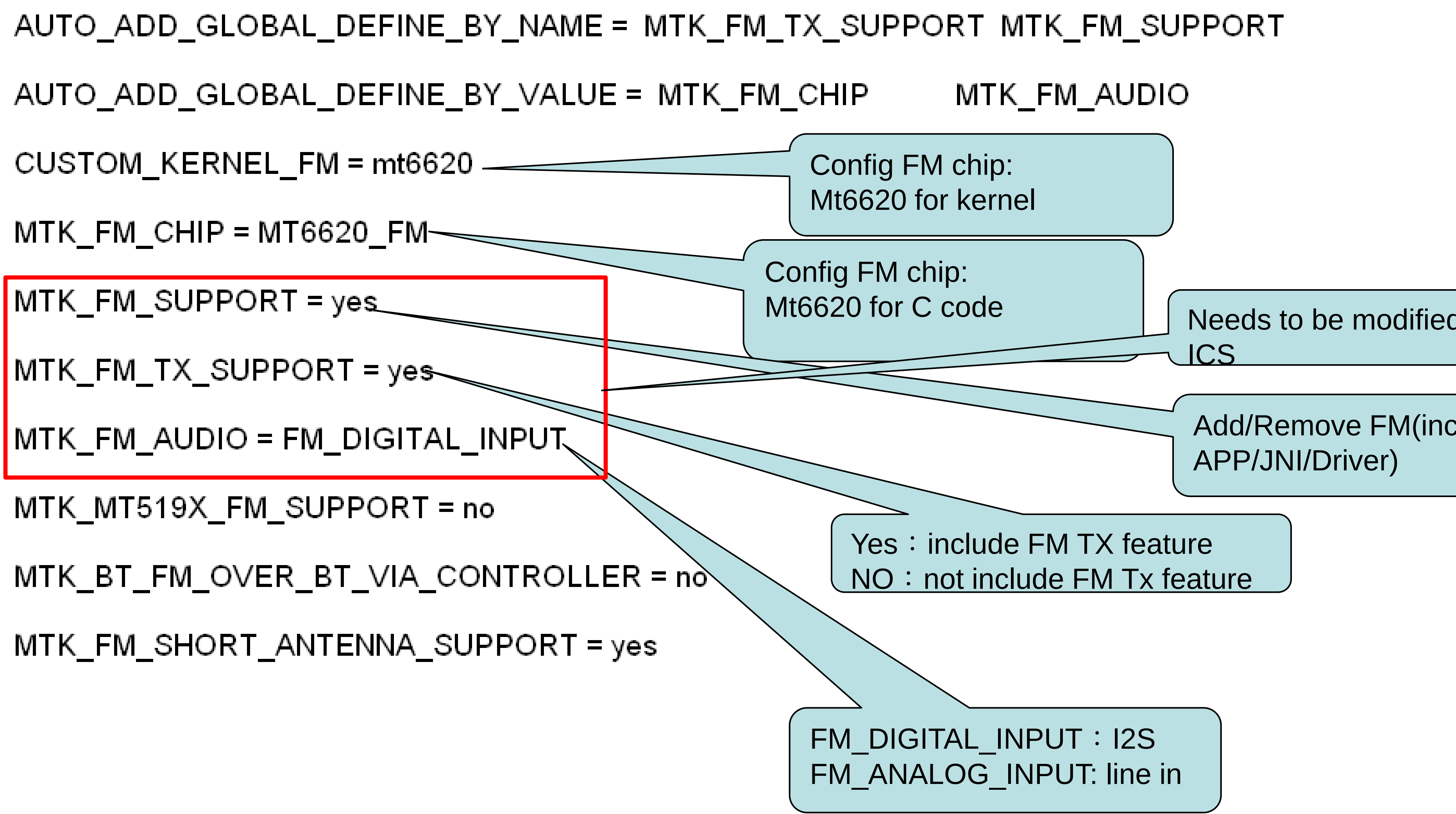
FM (2)

- Customers could fine tune FM performance
  - RSSI
  - PAMD(CQI)
  - SCAN channel size: default 40
  - FM band: default 1(UAS)
  - defined in “bionic/libc/common/linux/fm.h”



```
//RX
#define FMR_RSSI_TH_LONG   0x0301     //FM radio long antenna RSSI threshold(11.375dBuV)
#define FMR_RSSI_TH_SHORT   0x02E0     //FM radio short antenna RSSI threshold(-1dBuV)
#define FMR_CQI_TH        0x00E9     //FM radio Channel quality indicator threshold(0x0000~0x00FF)
#define FMR_SEEK_SPACE     1         //FM radio seek space,1:100KHZ; 2:200KHZ
#define FMR_SCAN_CH_SIZE   40        //FM radio scan max channel size
#define FMR_BAND           1        //FM radio band, 1:87.5MHz~108.0MHz; 2:76.0MHz~90.0MHz; 3:76.0MHz~108.0MHz; 4:special
#define FMR_BAND_FREQ_L    875       //FM radio special band low freq(Default 87.5MHz)
#define FMR_BAND_FREQ_H    1080       //FM radio special band high freq(Default 108.0MHz)

//TX

//******************************************************************************************
//***********************************FM config for engineer ***********************************
//******************************************************************************************
//RX
#define FMR_MR_TH         0x01BD     //FM radio MR threshold
#define ADDR_SCAN_TH         0xE0      //scan thrshold register
#define ADDR_CQI_TH        0xE1      //scan CQI register

//TX
#define FMTX_SCAN_HOLE_LOW  923       //92.3MHz~95.4MHz should not show to user
#define FMTX_SCAN_HOLE_HIGH 954       //92.3MHz~95.4MHz should not show to user
//******************************************************************************************
```

FM (2)



```
AUTO_ADD_GLOBAL_DEFINE_BY_NAME = MTK_FM_TX_SUPPORT MTK_FM_SUPPORT

AUTO_ADD_GLOBAL_DEFINE_BY_VALUE = MTK_FM_CHIP       MTK_FM_AUDIO

CUSTOM_KERNEL_FM = mt6620

MTK_FM_CHIP = MT6620_FM

MTK_FM_SUPPORT = yes

MTK_FM_TX_SUPPORT = yes

MTK_FM_AUDIO = FM_DIGITAL_INPUT

MTK_MT519X_FM_SUPPORT = no

MTK_BT_FM_OVER_BT_VIA_CONTROLLER = no

MTK_FM_SHORT_ANTENNA_SUPPORT = yes
```

Config FM chip:
Mt6620 for kernel

Config FM chip:
Mt6620 for C code

Needs to be modified
ICS

Add/Remove FM(inc
APP/JNI/Driver)

Yes：include FM TX feature
NO：not include FM Tx feature

FM_DIGITAL_INPUT：I2S
FM_ANALOG_INPUT: line in

FM (3)Compile option for ICS

- In ICS codebase， some of the compile options have been changed
- **Original**
  - MTK_FM_SUPPORT = yes
  - MTK_FM_TX_SUPPORT = yes
  - MTK_FM_AUDIO = FM_DIGITAL_INPUT
- **Now**
  - MTK_FM_SUPPORT = yes → FM feature switch control
  - MTK_FM_Rx_SUPPORT = yes
  - MTK_FM_TX_SUPPORT = yes
  - MTK_FM_Rx_AUDIO = FM_ANALOG_INPUT/ FM_DIGITAL_INPUT
  - MTK_FM_Tx_AUDIO = FM_ANALOG_OUTPUT/FM_DIGITAL_OUTPUT

Wi-Fi NVRAM settings(1)

Byte Offset	Content	Description	Default Value
0x000	u2Part1OwnVersion	Own version of the 1st 256-bytes of NVRAM content. This field indicates the version of the created content and might be identified by driver for compatibility checking.	0x0103
0x002	u2Part1PeerVersion	Required version for software component, usually driver, which parses the 1st 256 bytes of NVRAM content.	0x0000
0x004	aucMacAddress	MAC address	
0x	aucCountryCode	Country code for regulatory domain	0x0000
0x	rTxPwr	TX Power Control	
0x034	aucEFUSE	Mirrored content of EFUSE for overriding EFUSE values.	
0x0c4	ucTxPwrValid	Zero: rTxPwr is not valid Nonzero: Use values from rTxPwr for overriding default TX power	
0x0c5	ucSupport5GBand	Zero: Not supporting 5GHz band Nonzero: 5GHz band is supported	0x00
0x0c6	fg2G4BandEdgePwrUsed	Zero: Do not apply extra band edge power control Nonzero: Apply band edge TX power control	0x00
0x0c7	cBandEdgeMaxPwrCCK	Max. Band Edge TX Power for CCK rates	

Wi-Fi NVRAM settings (1)

0x0c8	cBandEdgeMaxPwrOFDM20	Max. Band Edge TX Power for OFDM rates within 20MHz bandwidth	
0x0c9	cBandEdgeMaxPwrOFDM40	Max. Band Edge TX Power for OFDM rates within 40MHz bandwidth	
0x0ca	ucRegChannelListMap	0: By aucCountryCode 1: By ucRegChannelListIndex 2: By aucRegSubBandInfo field	0x00
0x0cb	ucRegChannelListIndex	Channel list is defined based on channel list index in the mapping table of country channels	0x00
0x0cc	aucRegSubbandInfo	There are 6 regulation channel sub-bands and each sub-band has 6 bytes data. Please refer to the following regulation domain section for detailed description.	0x00, …, 0x00
0x0f0	aucReserved2	Reserved fields	
0x100	u2Part2OwnVersion	Own version of the 2nd 256-bytes of NVRAM content. This field indicates the version of the created content and might be identified by driver for compatibility checking.	0x0000
0x102	u2Part2PeerVersion	Required version for software component, usually driver, which parses the 2nd 256 bytes of NVRAM content.	0x0000
0x104			0x00
0x105			0x00
0x106	ucEnable5GBand	Zero: Disable 5GHz band support Nonzero: Enable 5GHz band support	0x00
0x107	aucTailReserved	Reserved fields	

Confidential A

MEDIATEK

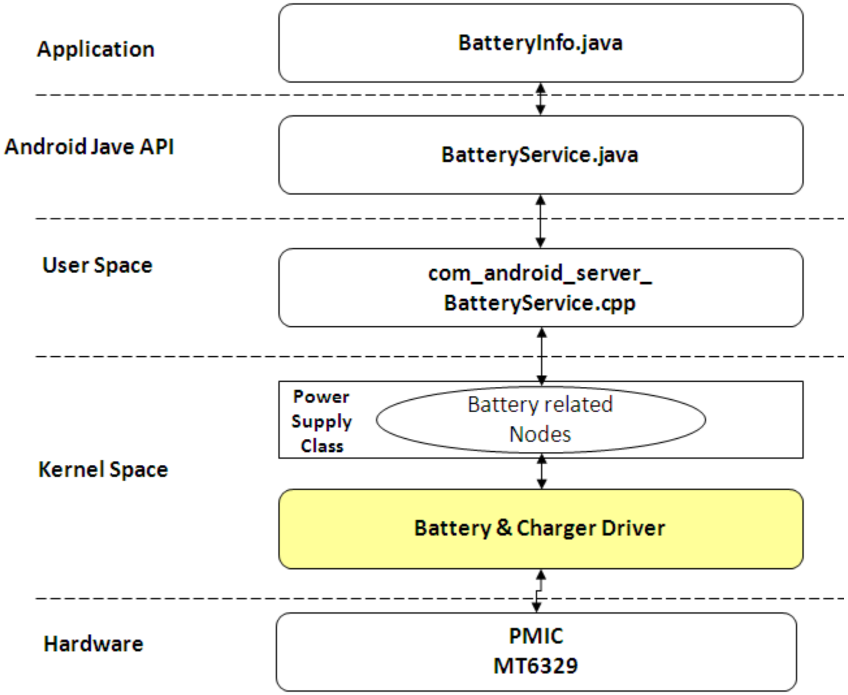
Battery Manager

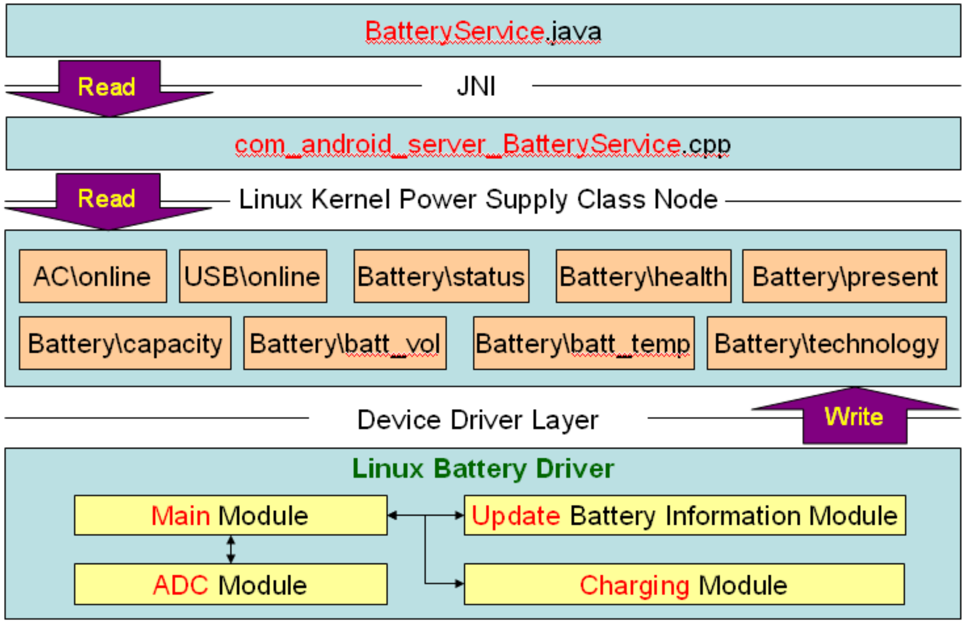
Outline

- **Battery Service**
- Battery Charging Overview
- Power Off Charging
- Fuel Gauge

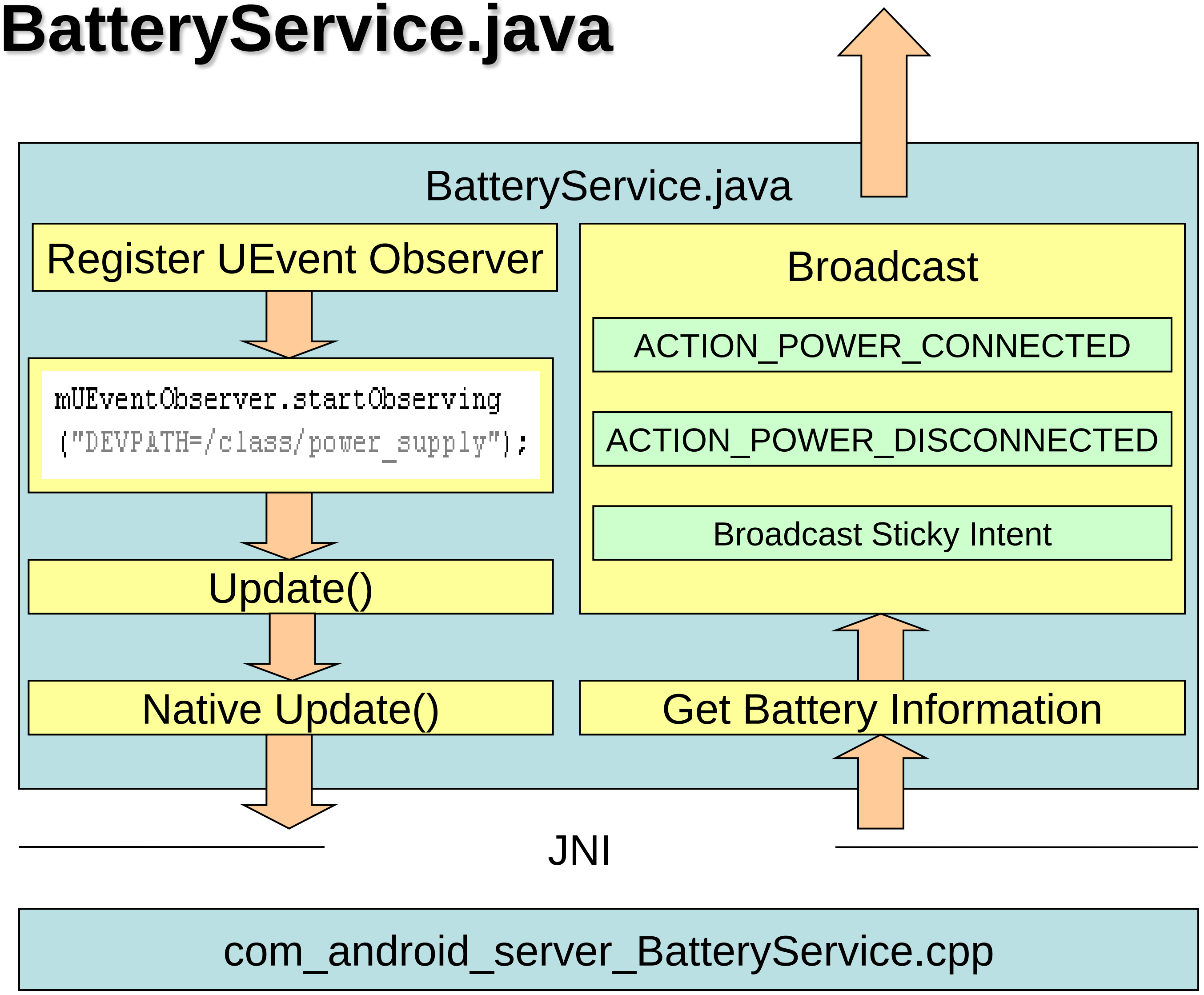
Battery Introduction

- Introduction



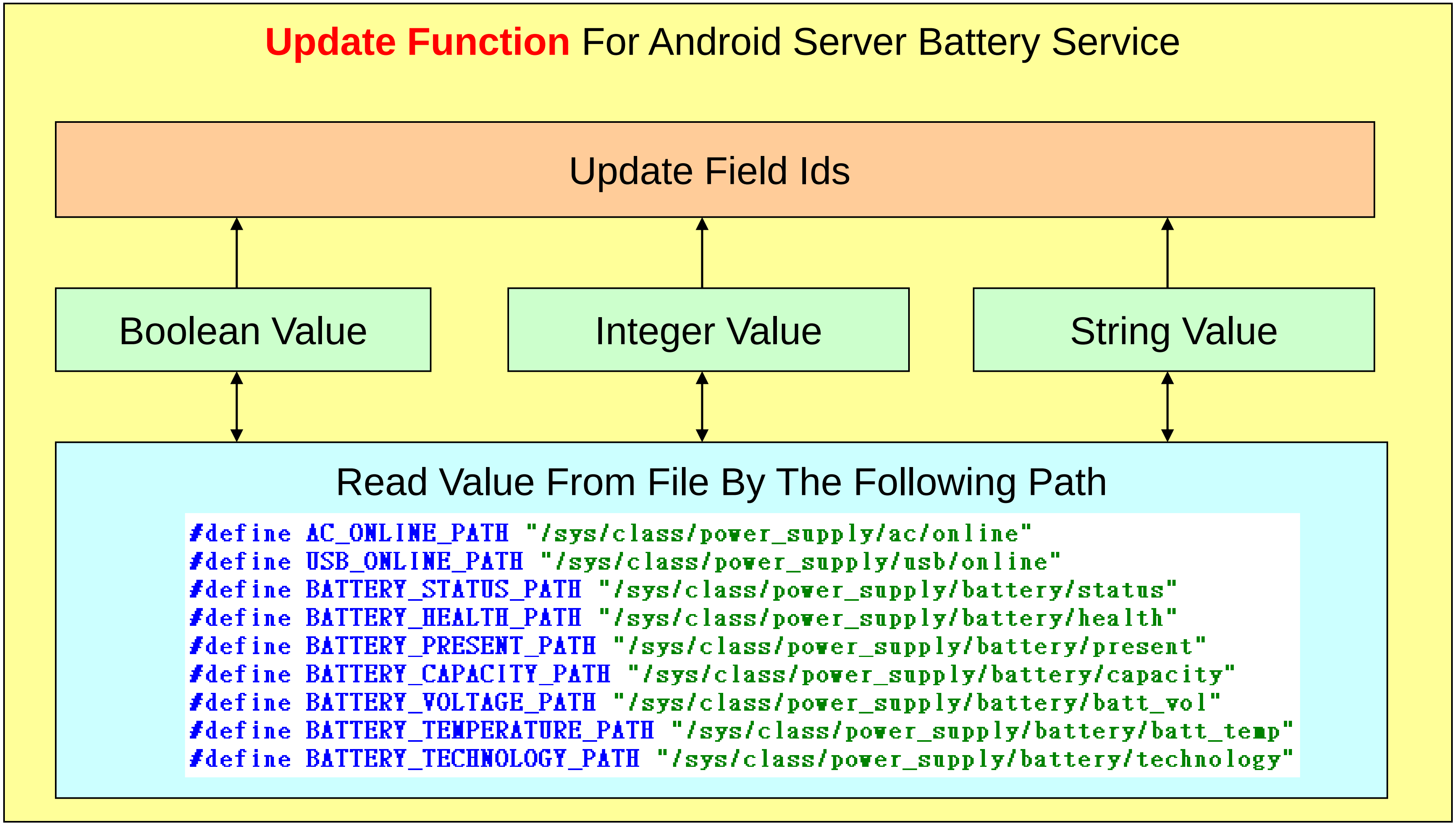


BatteryService.java



```
mUEventObserver.startObserving
("DEVPATH=/class/power_supply");
```

Battery Information Update Function



Update Function For Android Server Battery Service

Update Field Ids

Boolean Value

Integer Value

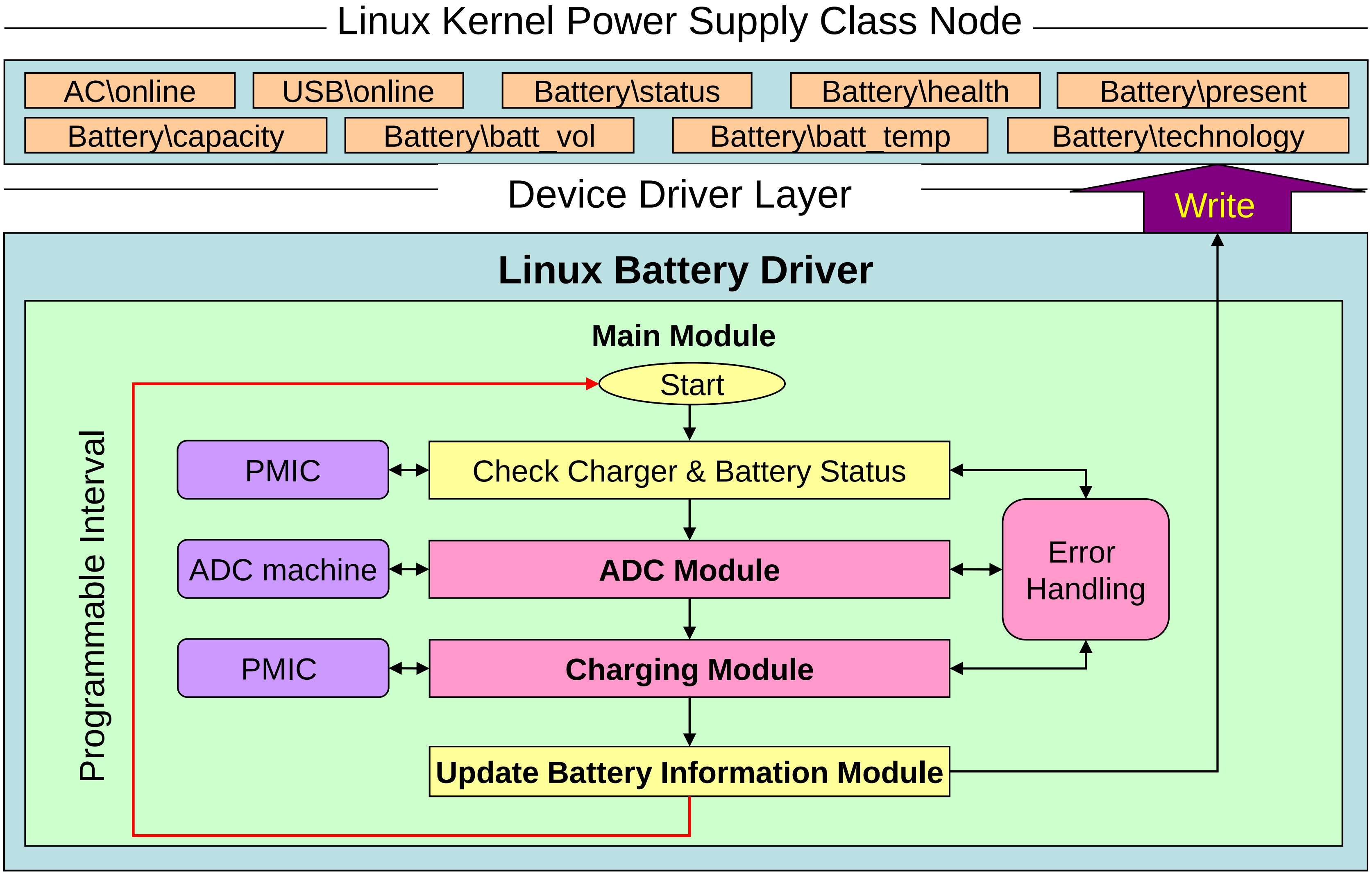
String Value

Read Value From File By The Following Path

```
#define AC_ONLINE_PATH "/sys/class/power_supply/ac/online"
#define USB_ONLINE_PATH "/sys/class/power_supply/usb/online"
#define BATTERY_STATUS_PATH "/sys/class/power_supply/battery/status"
#define BATTERY_HEALTH_PATH "/sys/class/power_supply/battery/health"
#define BATTERY_PRESENT_PATH "/sys/class/power_supply/battery/present"
#define BATTERY_CAPACITY_PATH "/sys/class/power_supply/battery/capacity"
#define BATTERY_VOLTAGE_PATH "/sys/class/power_supply/battery/batt_vol"
#define BATTERY_TEMPERATURE_PATH "/sys/class/power_supply/battery/batt_temp"
#define BATTERY_TECHNOLOGY_PATH "/sys/class/power_supply/battery/technology"
```

Working Module

Hareware / Register
Unique module
Common module



Linux Kernel Power Supply Class Node

AC\online
USB\online
Battery\status
Battery\health
Battery\present
Battery\capacity
Battery\batt_vol
Battery\batt_temp
Battery\technology

Device Driver Layer

Write

Linux Battery Driver

Main Module

Programmable Interval

Start

PMIC

Check Charger & Battery Status

ADC machine

ADC Module

Error Handling

PMIC

Charging Module

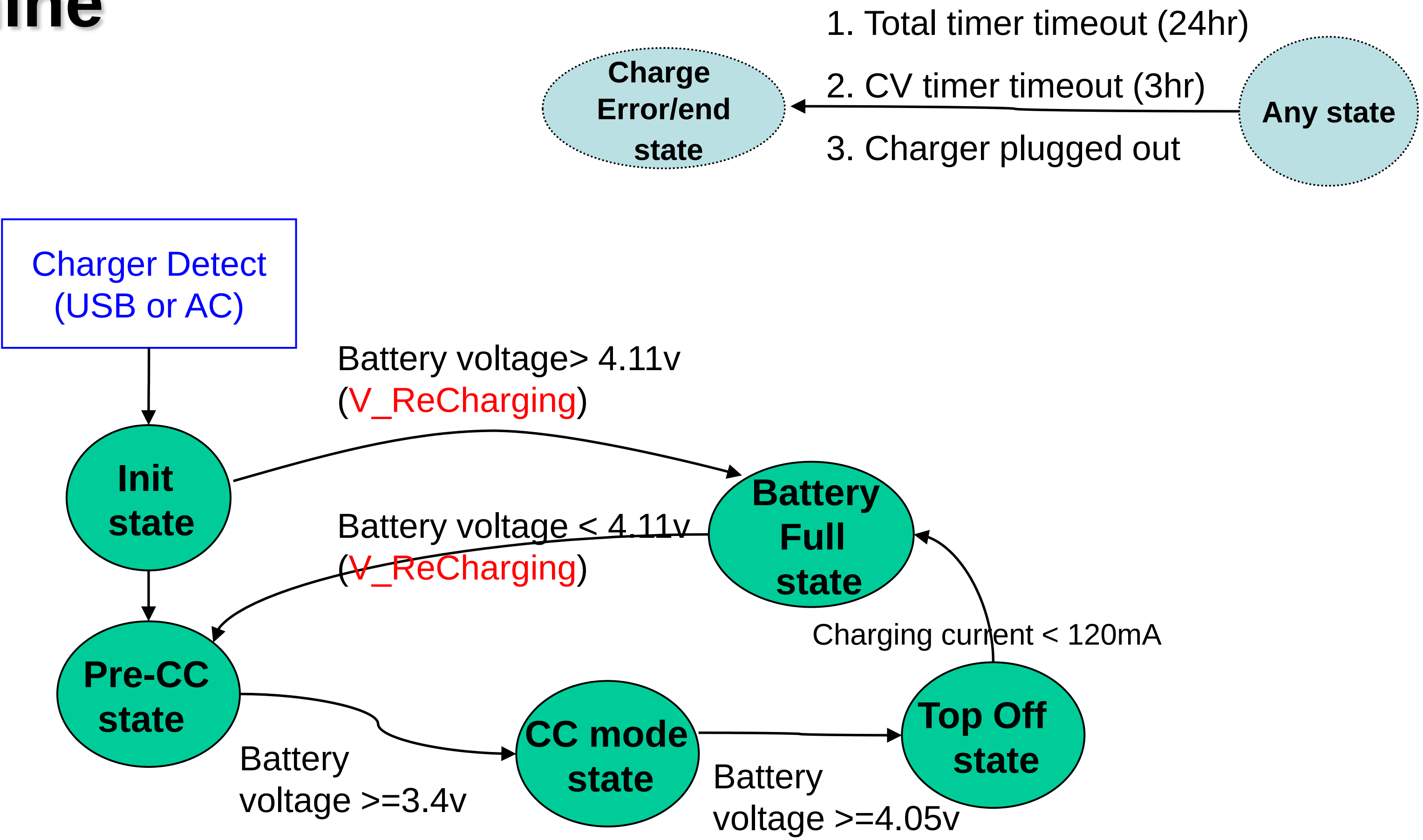
Update Battery Information Module

Reason : This is a simple, common and extendable working module for all BU.

Outline

- Battery Service
- **Battery Charging Overview**
- Power Off Charging
- Fuel Gauge

Battery Charging State Machine



Battery Customization

- Change file list

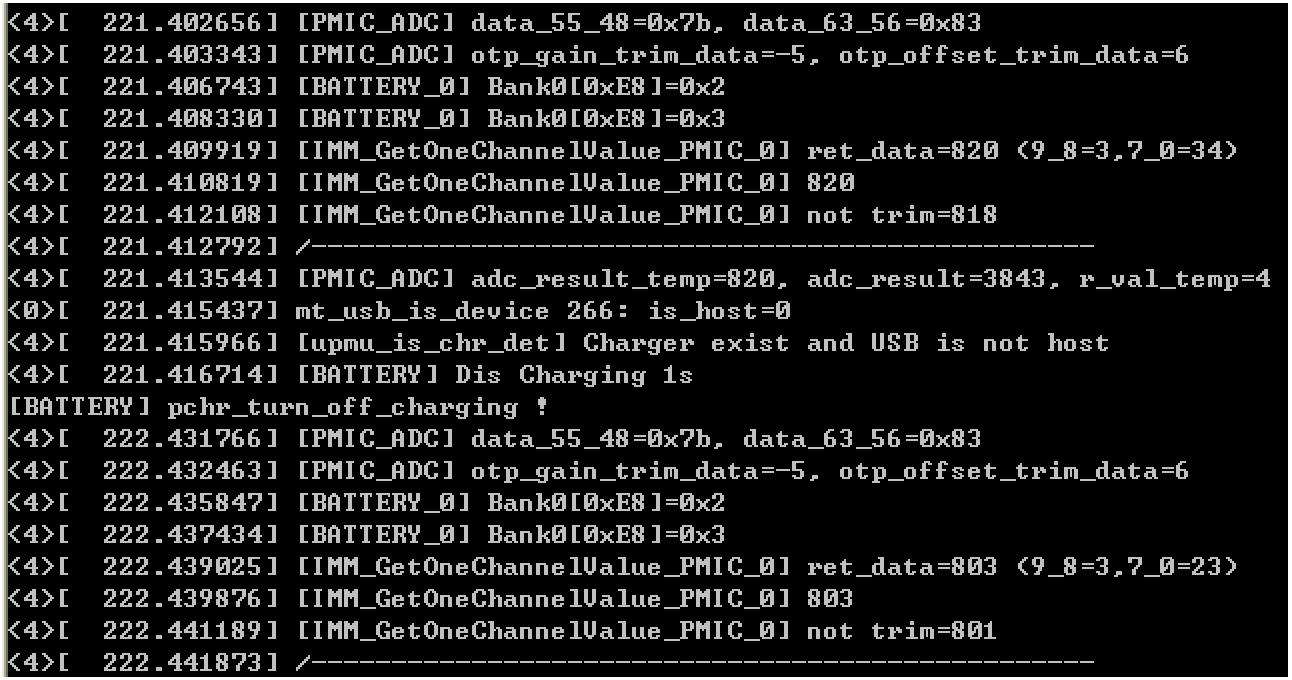
File	Description
alps\mediatek\platform\mt6575\kernel\drivers\power	
mt6575_battery.c	The implementation of battery charging related APIs.
mt6575_fuel_gauge.c	The implementation of fuel gauge related APIs.
alps\mediatek\platform\mt6575\kernel\drivers\power	
mt6575_battery.h	The battery charging related settings (internal).
alps\mediatek\custom\${*project_name*}\kernel\battery\battery	
cust_battery.h	The battery charging related settings (customer).
cust_fuel_gauge.h	The fuel gauge related settings (customer).

Customization item

Customization Item		Description	Default Value	Range
Battery Voltage-to-Percentage Table		11 level battery percentage (0,10,20,30,40,50,60,70,80,90,100%)	By battery SPEC.	3400mV (0%) ~ 4200mV (100%)
Normal Charging Current	USB	CC mode USB charging current	Cust_CC_450MA	Cust_CC_1600MA, Cust_CC_1500MA, Cust_CC_1400MA, Cust_CC_1300MA, Cust_CC_1200MA, Cust_CC_1100MA, Cust_CC_1000MA, Cust_CC_900MA, Cust_CC_800MA, Cust_CC_700MA, Cust_CC_650MA, Cust_CC_550MA, Cust_CC_450MA, Cust_CC_400MA, Cust_CC_200MA, Cust_CC_70MA , Cust_CC_0MA
	AC	CC mode AC charging current	Cust_CC_650MA	
USB-IF Charging Current	USB	CC mode USB charging current (USB suspended)	Cust_CC_0MA	
		CC mode USB charging current (USB un-configured)	Cust_CC_70MA	
		CC mode USB charging current (USB configured)	Cust_CC_450MA	
	AC	CC mode AC charging current	Cust_CC_650MA	
Recharging Battery Voltage		Recharging for keeping the battery capacity	4110mV	3900 ~ 4150mV
Battery Temperature Charging Protection	Higher	Above the hi-temperature ➔ disable charging	50C	- 20 ~ 60 C
	Lower	Below the low-temperature ➔ disable charging	0C	
Charging Resister		The resister for measuring the charging current	2 (=0.2Ohm)	
Battery Sense Resister		The resister for measuring the battery sense voltage	4	
I Sense Resister		The resister for measuring the I sense voltage	4	
Charger Sense Resister		The resister for measuring the charger sense voltage	((R_CHARGER_1+R_CHARGER_2)/R_CHARGER_2) R_CHARGER_1 = 330, R_CHARGER_2 = 39	
V_CHARGER_MAX		The max value of charger voltage	6000 (mV)	> V_CHARGER_MIN
V_CHARGER_MIN		The min value of charger voltage	4400 (mV)	< V_CHARGER_MAX
V_CHARGER_ENABLE		Enable/disable the charger voltage protection	0	(1:ON, 0:OFF)
RBAT_PULL_UP_R		The pull up resister for measuring battery temperature	24000 (Ohm)	
RBAT_PULL_UP_VOLT		The pull up voltage for measuring battery temperature	1200 (mV)	
TBAT_OVER_CRITICAL_LOW		The extreme value for calculating resister	483954	
BAT_TEMP_PROTECT_ENABLE		Enable/disable the battery temperature protection	0	(1:ON, 0:OFF)

Runtime Battery Logging Entry

- Path
  - cd /proc
- Enable logging
  - echo 1 > batdrv_log
- Disable logging
  - echo 0 > batdrv_log
- Demo



```
<4>[  221.402656] [PMIC_ADC] data_55_48=0x7b, data_63_56=0x83
<4>[  221.403343] [PMIC_ADC] otp_gain_trim_data=-5, otp_offset_trim_data=6
<4>[  221.406743] [BATTERY_0] Bank0[0xE8]=0x2
<4>[  221.408330] [BATTERY_0] Bank0[0xE8]=0x3
<4>[  221.409919] [IMM_GetOneChannelValue_PMIC_0] ret_data=820 (9_8=3,7_0=34)
<4>[  221.410819] [IMM_GetOneChannelValue_PMIC_0] 820
<4>[  221.412108] [IMM_GetOneChannelValue_PMIC_0] not trim=818
<4>[  221.412792] /--------------------------------------------------
<4>[  221.413544] [PMIC_ADC] adc_result_temp=820, adc_result=3843, r_val_temp=4
<0>[  221.415437] mt_usb_is_device 266: is_host=0
<4>[  221.415966] [upmu_is_chr_det] Charger exist and USB is not host
<4>[  221.416714] [BATTERY] Dis Charging 1s
[BATTERY] pchr_turn_off_charging !
<4>[  222.431766] [PMIC_ADC] data_55_48=0x7b, data_63_56=0x83
<4>[  222.432463] [PMIC_ADC] otp_gain_trim_data=-5, otp_offset_trim_data=6
<4>[  222.435847] [BATTERY_0] Bank0[0xE8]=0x2
<4>[  222.437434] [BATTERY_0] Bank0[0xE8]=0x3
<4>[  222.439025] [IMM_GetOneChannelValue_PMIC_0] ret_data=803 (9_8=3,7_0=23)
<4>[  222.439876] [IMM_GetOneChannelValue_PMIC_0] 803
<4>[  222.441189] [IMM_GetOneChannelValue_PMIC_0] not trim=801
<4>[  222.441873] /--------------------------------------------------
```

LOG Analysis

- BATTERY:ADC
  - VCHR, BAT_SENCE, I_SENCE
    - Real-time value from ADC channel
  - Current
    - Conversion value from BAT_SENCE, I_SENCE

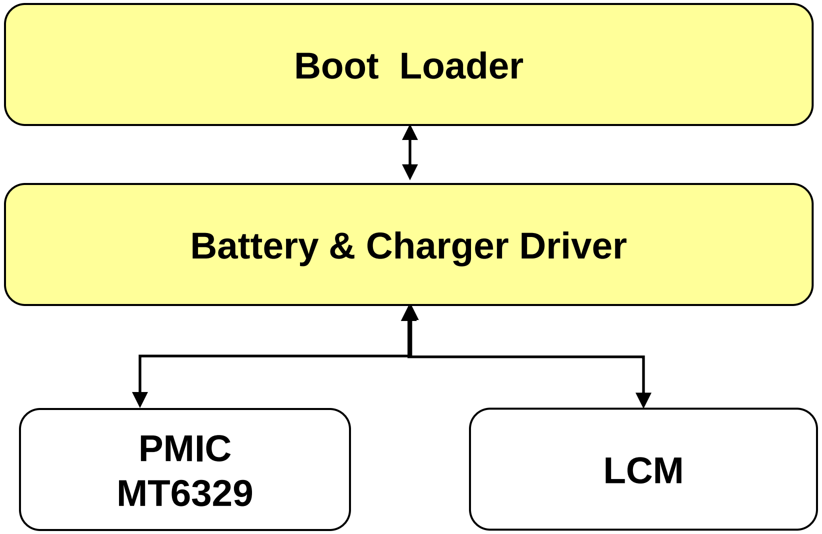
- BATTERY:AVG
  - vbat, Icharging, SOC (battery percentage)
    - vbat : 5 minutes BAT_SENCE average value
    - Icharging : 5 minutes I_SENCE average value
    - SOC : 5 minutes battery percentage average value
  - state
    - CHR_CC (0x1002), CHR_CV (0x1003), CHR_BATFULL (0x1004)
    - CHR_ERROR (0x1005)
  - chrtype
    - STANDARD_HOST (1), CHARGING_HOST (2),NONSTANDARD_CHARGER (3), STANDARD_CHARGER(4)

Outline

- Battery Service
- Battery Charging Overview
- **Power Off Charging**
- Fuel Gauge

Power Off Charging Introduction

- Introduction



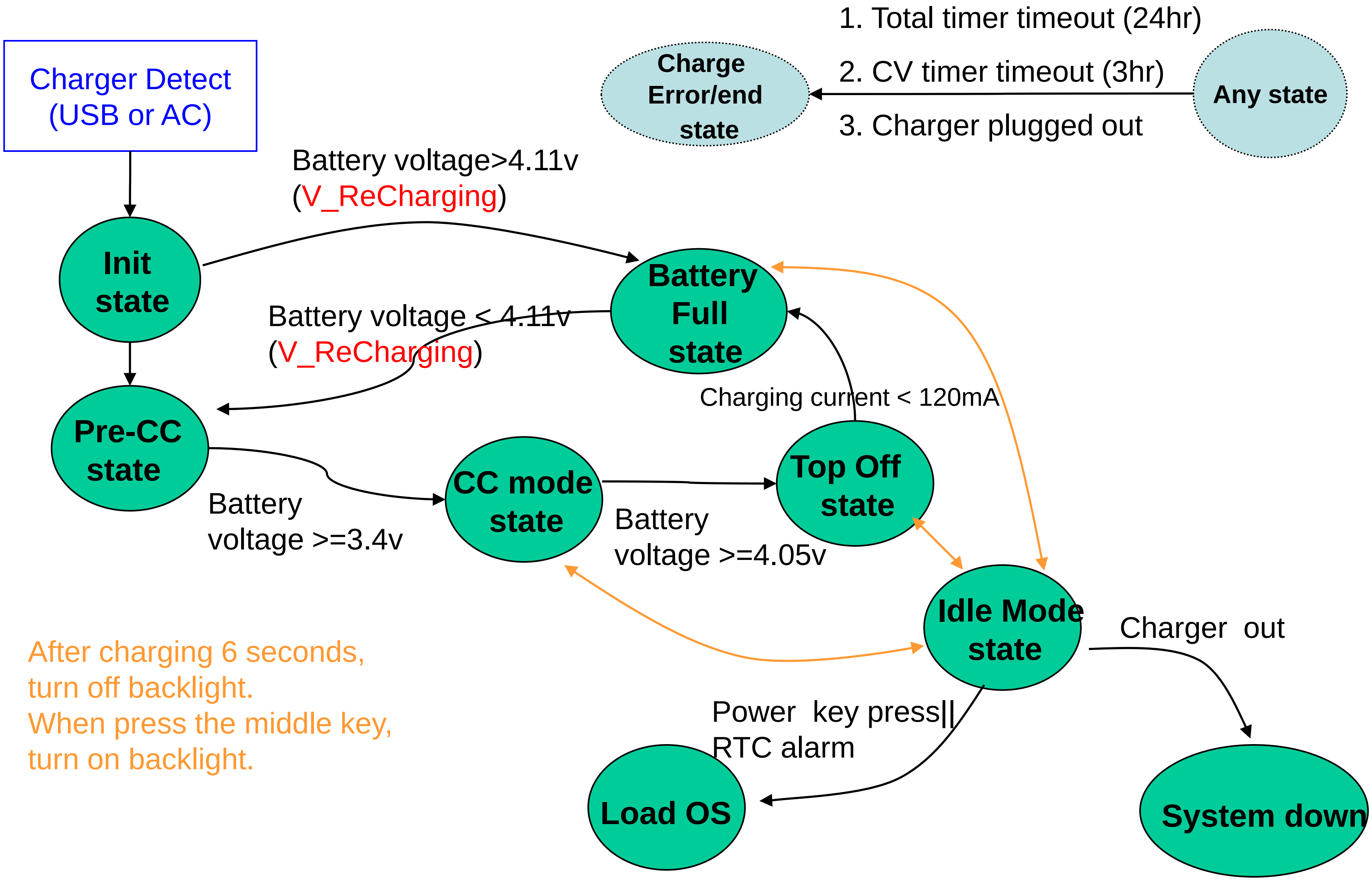
Boot Loader

Battery & Charger Driver

PMIC
MT6329

LCM

Charging State Machine



After charging 6 seconds, turn off backlight.
When press the middle key, turn on backlight.

Power Off Charging Customization

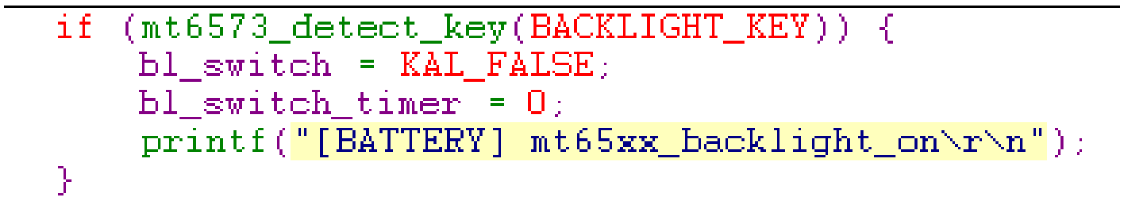
- Change file list

File	Description
alps\mediatek\platform\mt6573\uboot	
mt6575_bat.c	The implementation of battery charging related APIs.
alps\mediatek\custom\${project_name}\uboot\inc	
cust_battery.h	The battery charging related settings.

- Customization item(same with kernel)

Wake up the backscreen

- Set CHARGING_IDLE_MODE to 1
  - alps\mediatek\custom\${project_name}\uboot\inc\cust_battery.h
  - Uboot will turn off the backlight after some seconds in power off charging
- BL_SWITCH_TIMEOUT
  - Define the timeout time, default is 6 seconds
  - check BAT_CheckBatteryStatus () in alps\mediatek\platform\mt6575\uboot\mt6575_bat.c
  - Press the BACKLIGHT_KEY to wake up the back screen



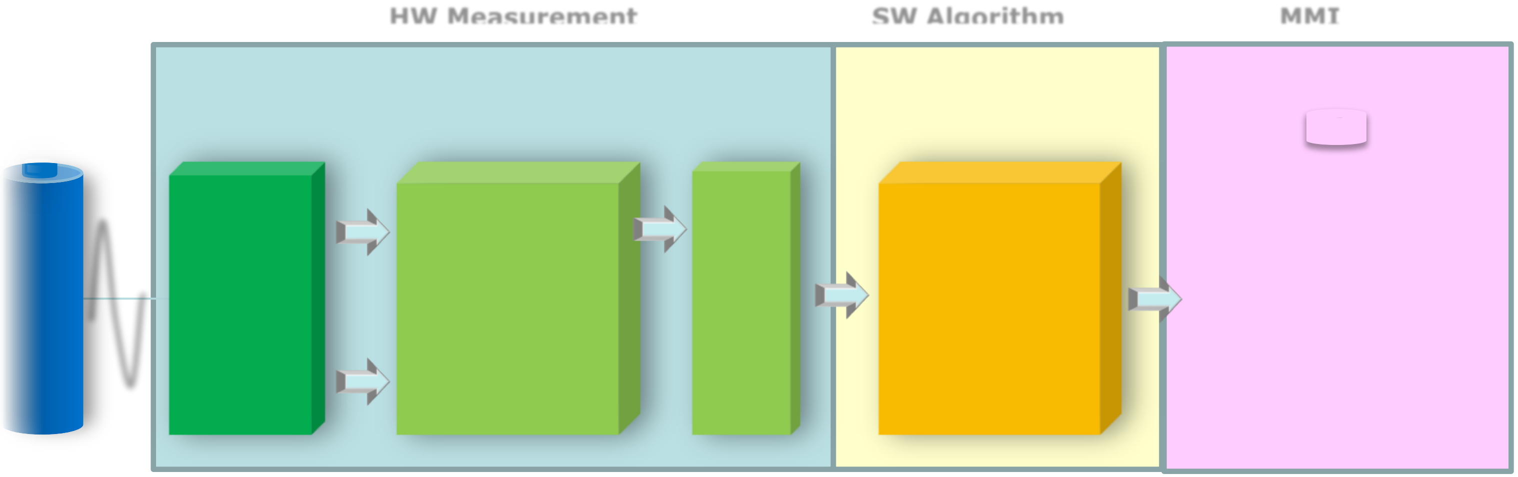
```
if (mt6573_detect_key(BACKLIGHT_KEY)) {
    bl_switch = KAL_FALSE;
    bl_switch_timer = 0;
    printf("[BATTERY] mt65xx_backlight_on\r\n");
}
```

Outline

- Battery Service
- Battery Charging Overview
- Power Off Charging
- **Fuel Gauge**

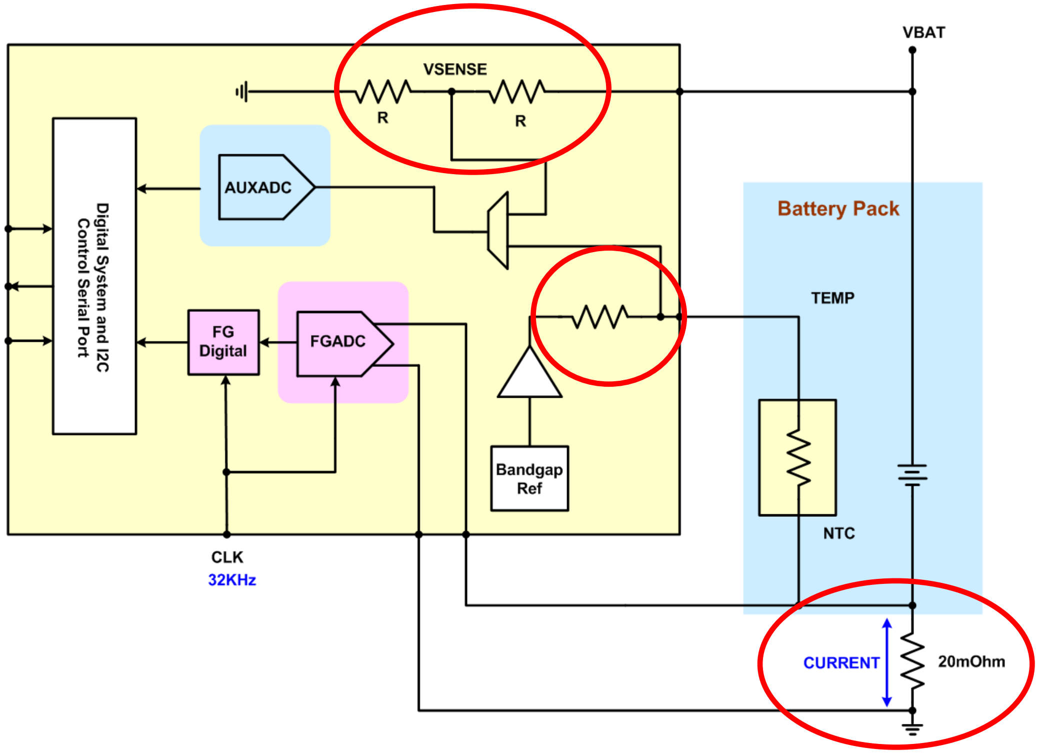
MTK Fuel Gauge System

- System-side Li-Ion battery fuel gauge S%
  - Precise Battery Fuel Gauging
  - Battery current measurement



Fuel Gauge

- Introduction
  - The fuel gauging system includes a dedicated ADC for Li-Ion battery current measurements and utilizes the measurement ADC (AUXADC) for battery voltage and temperature measurements. The battery state-of-charge (SOC) estimation is performed by software using these three measurements and the accumulated current measurement.
  - The application diagram of the fuel gauging system is shown in below, where an external resistor is used to converter the current drawn from the battery into a voltage which is then measured by the FG ADC.
    - The value of the external resistor must be chosen such that the maximum current during charging or discharging does not cause the ADC to exceed its input voltage range.



Fuel Gauge Customer Support

Case of Customer Support		Customers	Pros	Cons	Effort
No Use Fuel Gauge		1. For Cost down (remove Rfg) 2. Do not case the battery percentage	Remove Rfg (< 0.01US)	Battery percentage error rate = 30%~50%	None
Use MTK Fuel Gauge	**Use default ZCV Table**	1. Need precise battery percentage 2. Can not get the battery ZCV table	1. Battery percentage error rate < 20% 2. Cost is cheaper than the Fuel Gauge IC (0.6~0.9US)	Need Use default ZCV Table	Need Rfg (< 0.01US)
	MTK SA measure ZCV Table for each customer	**1.Need precise battery percentage 2. Can get the battery ZCV table**	**1. Battery percentage error rate <10% 2. Cost is cheaper than the Fuel Gauge IC (0.6~0.9US)**	**Need 3 weeks for creating the ZCV table**	**1. Need Rfg (< 0.01US) 2. Need provide the battery packet and SPEC to MTK SA for creating the ZCV table. (same as the flow of Gas Gauge IC vender)**

MEDIATEK

Audio

MT6575 Audio System

- Audio SubSystem is almost the same between 6573 and 6575.

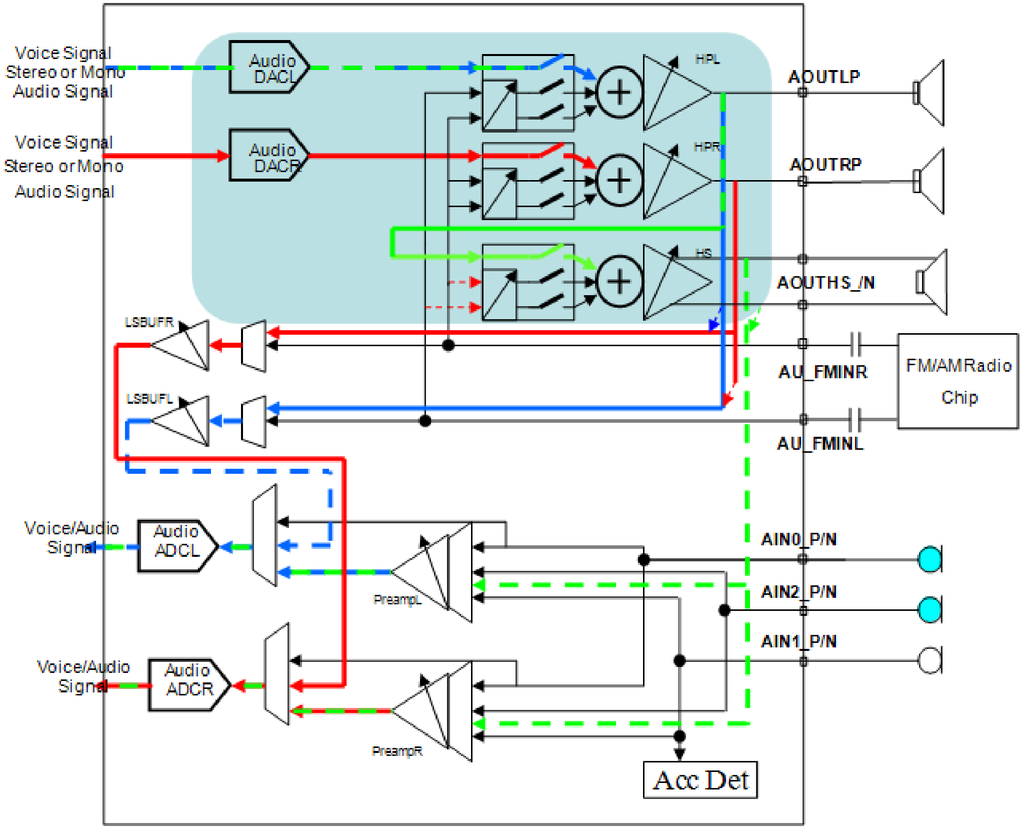
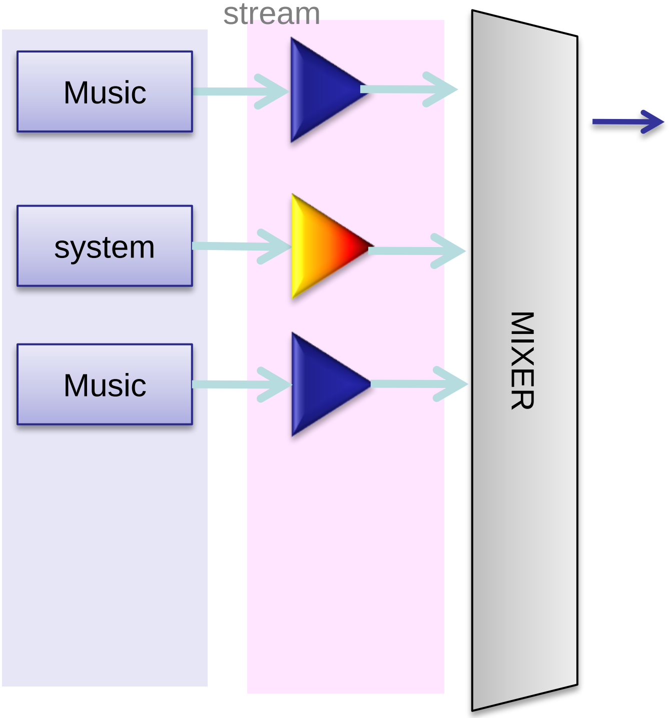


Figure 1-6. Block Diagram of Audio Mixed-signal Blocks

Audio gain control

- Digital gain – by android stream type
- Analog gain – audio PGA gain.

Digital gain by source by stream



Analog gain

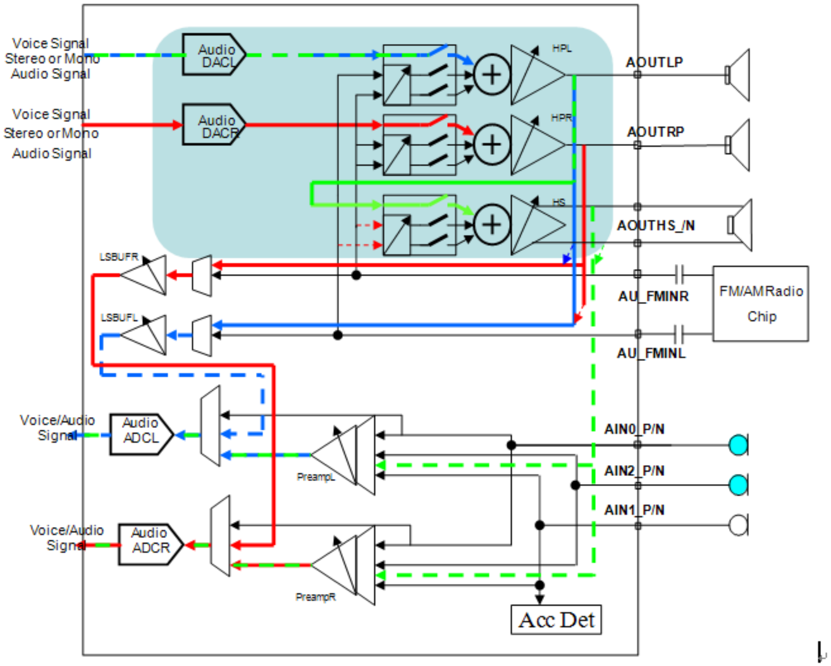


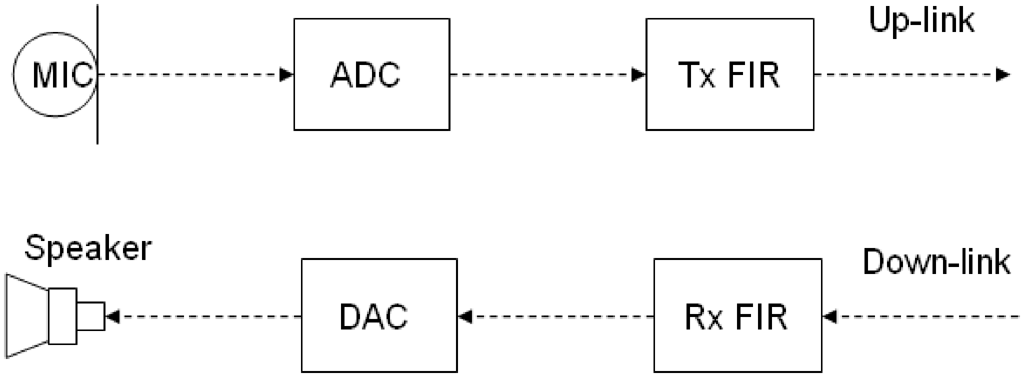
Figure 1-6. Block Diagram of Audio Mixed-signal Blocks

Different part

- Audio
  - High Sample rate recording.
    - Now AP support for 32kHz and 48kHz recording.
  - Headset compensation filter
    - Supporting compensate for headset and headphone.
    - Tuning tool is the same.
- Speech part
  - Add wide-band dual Mic capability.

Speech setting Tx/Rx FIR

- Currently, Tx/Rx FIR are stored in AP side. The data structure is the same as feature phone.
- Please use quick tuning tool to adjust speech relate parameters.



Define in audio_custom_exp.h

ENABLE_AUDIO_TRAK_LOUDNESS	Define this will enable loudness in audio track, this will apply loudness on ringtone steam. We suggest if it is opened, do not configure ACF with DRC Otherwise ringtone stream will pass 2 DRC and will be very loud.
ENABLE_AUDIO_COMPENSATION_FILTER	Define this will enable audio compensation filter with speaker mode It can be “pure ACF” or ACF with DRC. Please reference ACF document for more detail.
AUDIO_COMPENSATION_FLT_MODE	Define the mode of compensation filter, please reference ACF document.
HEADPHONE_COMPENSATION_FLT_MODE	Define the mode of compensation filter, please reference ACF document.
ENABLE_HEADPHONE_COMPENSATION_FILTER	Define this will enable audio compensation filter for headset and headphone mode.
ENABLE_STEREO_SPEAKER	if define Stereo speaker , speaker output will not do stero to mono, keep in stereo format because stereo output can apply on more than 1 speaker.
ENABLE_HIGH_SAMPLERATE_RECORD	When enable this feature , recording can support for 32K and 48KHz.

Document

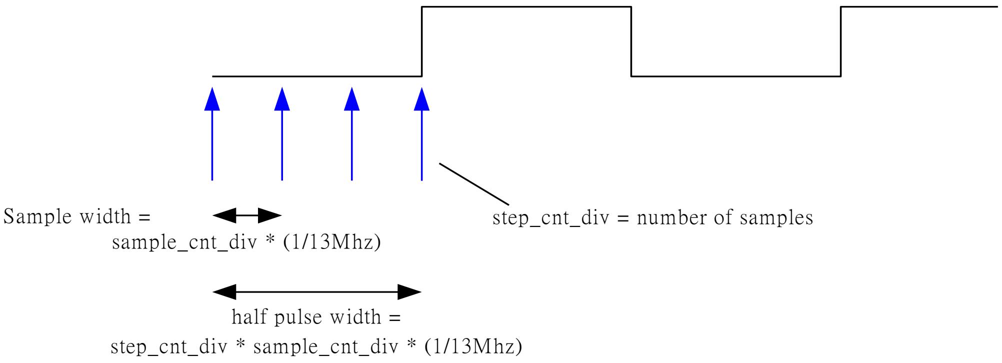
- Audio volume customization
  - Please reference to
    - Android_MT657x_SmartPhone_Audio_Customization.doc
    - Android_657x_Audio_Customer.pptx
- Meta and Engineering mode tuning
  - Audio_MT657x_Audio_EM&Meta_Tunning.pptx.
- Compensation filter
  - Other teammate will do training course.
- Speech tuning
  - Other teammate will do training course

MEDIATEK

I2C

Programmable Timing Parameters of I2C

- The system sets default parameters for I2C.
- Some I2C slave devices need to meet their timing requirements.



Sample width =
sample_cnt_div * (1/13Mhz)

step_cnt_div = number of samples

half pulse width =
step_cnt_div * sample_cnt_div * (1/13Mhz)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name						**SAMPLE_CNT_DIV**					**STEP_CNT_DIV**					
Type						R/W					R/W					
Reset						‘h3					‘h3					

Customize APIs

- Two series APIs in the I2C driver perform master send/receive transactions.
  - i2c_master_send(), i2c_master_recv()
  - i2c_transfer()
- Customization of I2C APIs is **ONE-SHOT only**.
  - I2C_A_FILTER_MSG : filter out error message
  - I2C_WR_FLAG: enable write and read transaction
  - I2C_RS_FLAG: enable repeat start transaction

Difference between MT6575 and MT6573

- MT6575 has 3 I2C controllers
- ALL don’t support DMA.

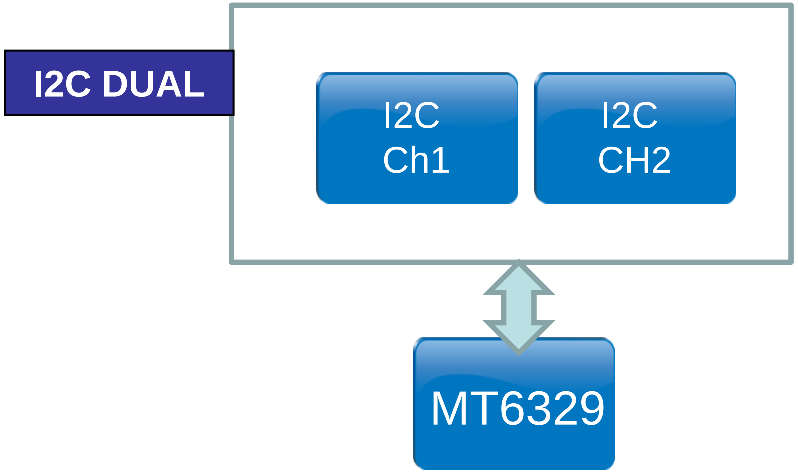
Controller #	Pin Name	GPIO pin	Feature
I2C0	SCL0, SDA0	GPIO87, GPIO88	NO DMA
I2C1	SCL1, SDA1	GPIO222, GPIO224	NO DMA
I2C2(DUAL)	SCL2, SDA2	GPIO145, GPIO142	DUAL/NO DMA

Difference between MT6575 and MT6573

- I2C polling burst write mode(new in MT6575).
  - optimization for sums of data sending once a time.
  - performance like DMA in MT6573
  - automatically start when there are over 8 bytes data sending

I2C dual introduction

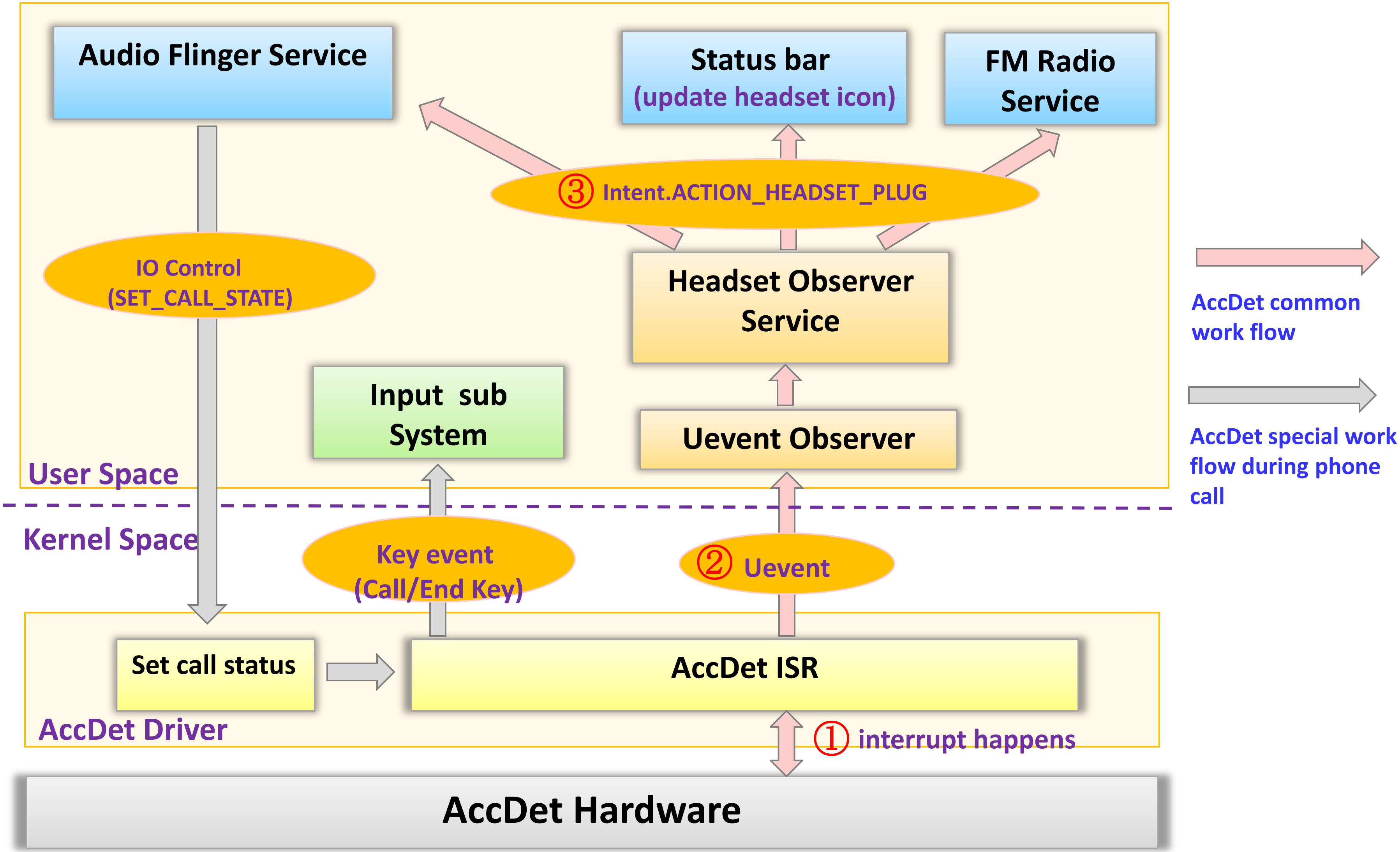
- I2C dual is a controller who contains two host but shares the same pin to connect client.



- Only for MT6329 use.

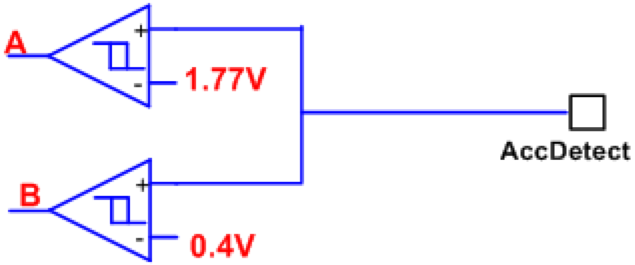
HeadSet

Headset Introduction



HW Architecture

- AccDet Hardware Design:
  - Accessory detecting depends on the voltage when 3-pole or 4 pole headset plug in/out, and it uses internal 2-bit comparator to separate what kinds of external components are.



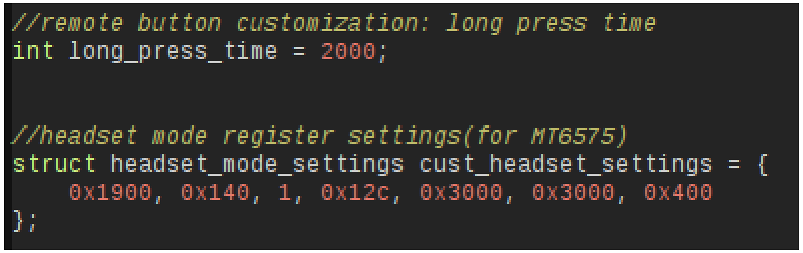
- If the voltage of AccDet is higher than 1.77V, A=1; or else, A=0;
- If the voltage of AccDet is higher than 0.4V, B=1; or else, B=0.
- So AccDet is divided into 3 headset state according to the voltage range:
  - Plug out state: 1.77V ≤ Voltage ≤1.9V (A=1, B=1);
  - Mic Bias state: 0.4V ≤ Voltage<1.77V (A=0, B=1);
  - Hook Switch state: 0V ≤ Voltage < 0.4V (A=0, B=0).

Headset Customization

- Change file list

File	Description
mediatek\platform\mt6575\kernel\drivers\accdet\	
accdet.c	The implementation of headset related APIs.
mediatek\custom\$(project)\headset\accdet\	
accdet_custom.h	The headset related settings.

- Customization item
  - long_press_time
    - Headset_custom.h
  - cust_headset_settings
    - Headset_custom.h



```
//remote button customization: long press time
int long_press_time = 2000;


//headset mode register settings(for MT6575)
struct headset_mode_settings cust_headset_settings = {
    0x1900, 0x140, 1, 0x12c, 0x3000, 0x3000, 0x400
};
```

Confidential A

MEDIATEK

USB OTG

Outline

- Introduction
- Code Architecture
- OTG Customization
- Android ICS enhancement
- Limitations
- How to disable OTG feature

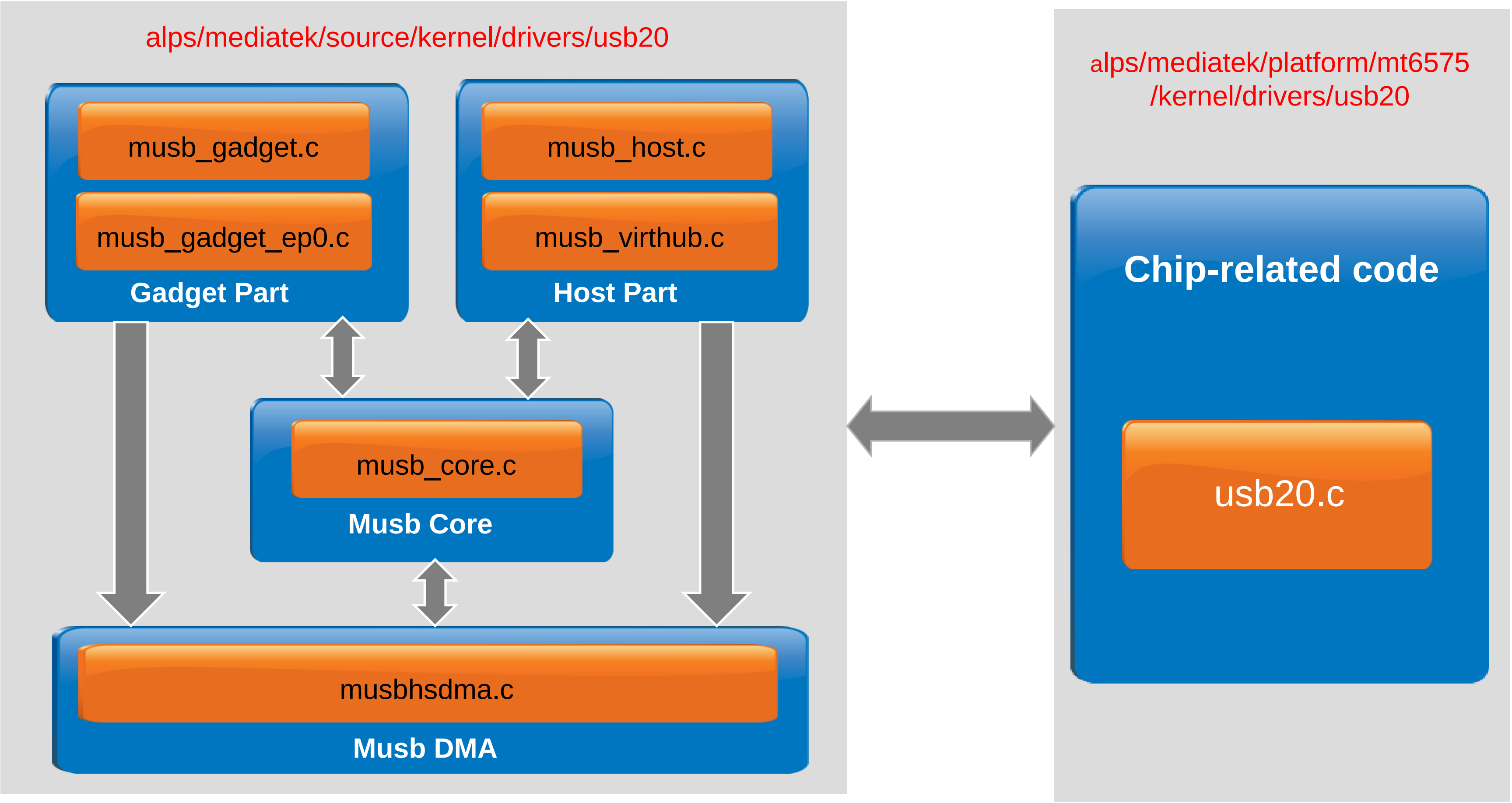
OTG Introduction

- An OTG product is a portable device that uses a single Micro-AB receptacle to operate at times as a USB Targeted Host and at times as a USB peripheral. OTG devices must always operate as a standard peripheral when connected to a standard USB host.

  – Plug in A-cable, phone can be used as host.

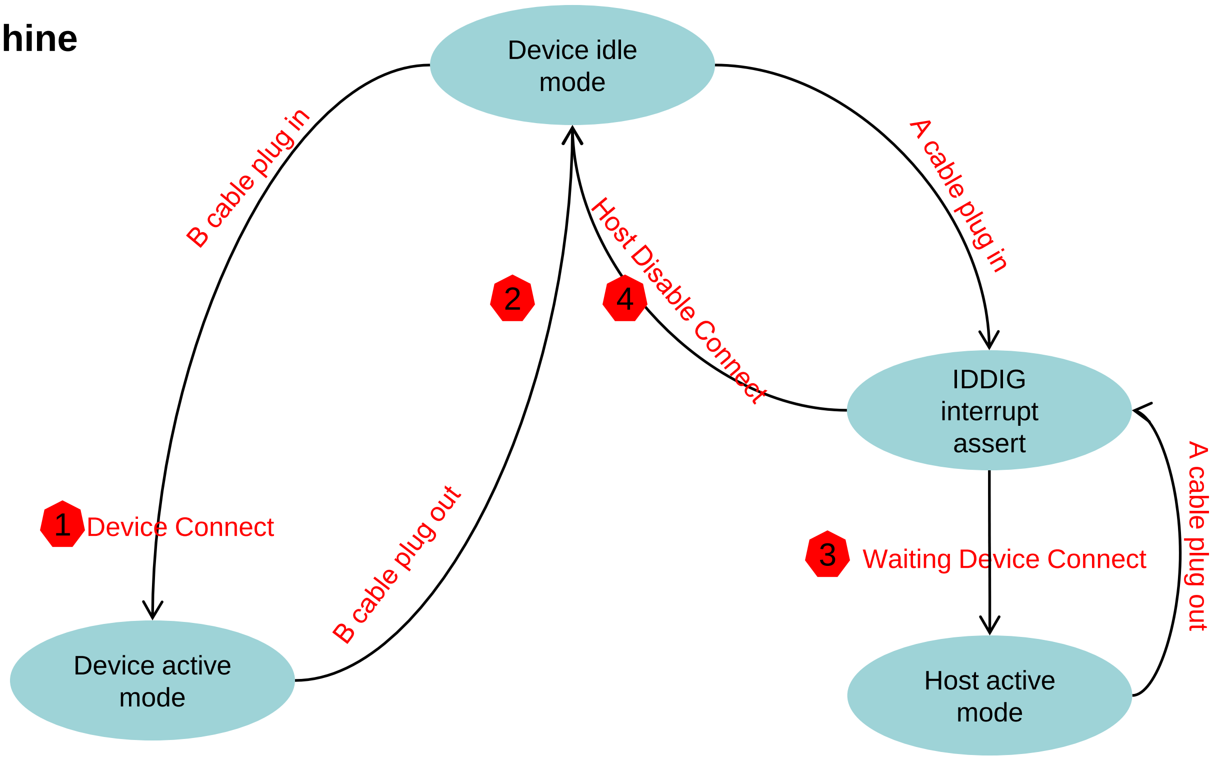
  – Plug in B-cable, phone can be used as device.

MT6575 MUSB Code Architecture



For different chip, we just need modify the usb20.c file

OTG State Machine



1 Will enable soft connect and wait USB host to enumerate-- musb_start()

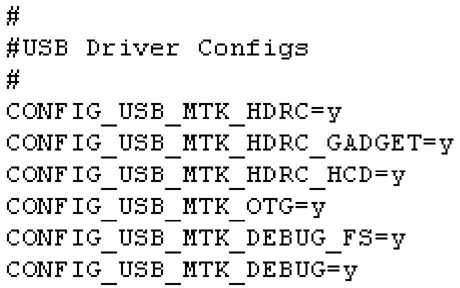
2 Will disable all EPs, flush all EPs FIFO and clear the soft connect – musb_stop()

3 Will set the session and vbus, enable all interrupt, wait for usb device connect – musb_start()

4 Will clear the session and vbus, disable all EPs, flush all EPs FIFO – musb_stop()

Some Configs

- There are some configs about MUSB defined in alps\mediatek\config\mt6575\autoconfig\kconfig\platform
  - CONFIG_USB_MTK_HDRC
    - If use MUSB, it must be defined.
  - CONFIG_USB_MTK_HDRC_GADGET
    - If support usb gadget, it must be defined.
  - CONFIG_USB_MTK_HDRC_HCD
    - If support usb host, it must be defined.
  - CONFIG_USB_MTK_OTG
    - If support usb OTG, it must be defined. If this config is defined, CONFIG_USB_MTK_HDRC_HCD also must be defined.
  - CONFIG_USB_MTK_DEBUG_FS
    - If use MUSB debug fs, it must be defined.
  - CONFIG_USB_MTK_DEBUG
    - If want define DEBUG macro, it must be defiend.



```
#
#USB Driver Configs
#
CONFIG_USB_MTK_HDRC=y
CONFIG_USB_MTK_HDRC_GADGET=y
CONFIG_USB_MTK_HDRC_HCD=y
CONFIG_USB_MTK_OTG=y
CONFIG_USB_MTK_DEBUG_FS=y
CONFIG_USB_MTK_DEBUG=y
```

- Also we need implement the kconfig file in alps\mediatek\platform\mt6575\kernel\kconfig\drivers

Chip relative head file

- In file alps/mediatek/platform/mt6575/kernel/core/include/mach/mtk_musb.h, we define some chip relative macros and declarations :
  - Some USB register address which are different in different chip. Such as SWRST register.
  - USB phy register access macro.
  - Some chip relative function declarations which define in other kernel modules.
  - If the num of the logical endpoints MUSB driver support is not 8, we need define MUSB_C_NUM_EPS macro. Its value equals the num of the logical endpoints + 1(endpoint 0) .

OTG Factory Mode Implement

- We implement a switch device to tell the usb state to user space, the factory mode app will polling the file /sys/class/switch/otg_state/state to get the usb state.
- There are two factory test cases for usb and both of them is implement by using a switch device to tell the status to user space. factory mode app will polling the following files to check the status:
  - OTG test: used to test the USB mode
    - /sys/class/switch/otg_state/state
    - state = 0 means USB is in device mode.
    - state = 1 means USB is in host mode.
  - USB configuration test: used to test the USB configuration status
    - /sys/class/switch/usb_configuration /state
    - state = 0 means USB is not configured.
    - state =1 means USB is configured.
  - Relative codes is located in /alps/mediatek/source/factory/src/test/ftm_usb.c and /alps/mediatek/source/factory/src/test/ftm_otg.c

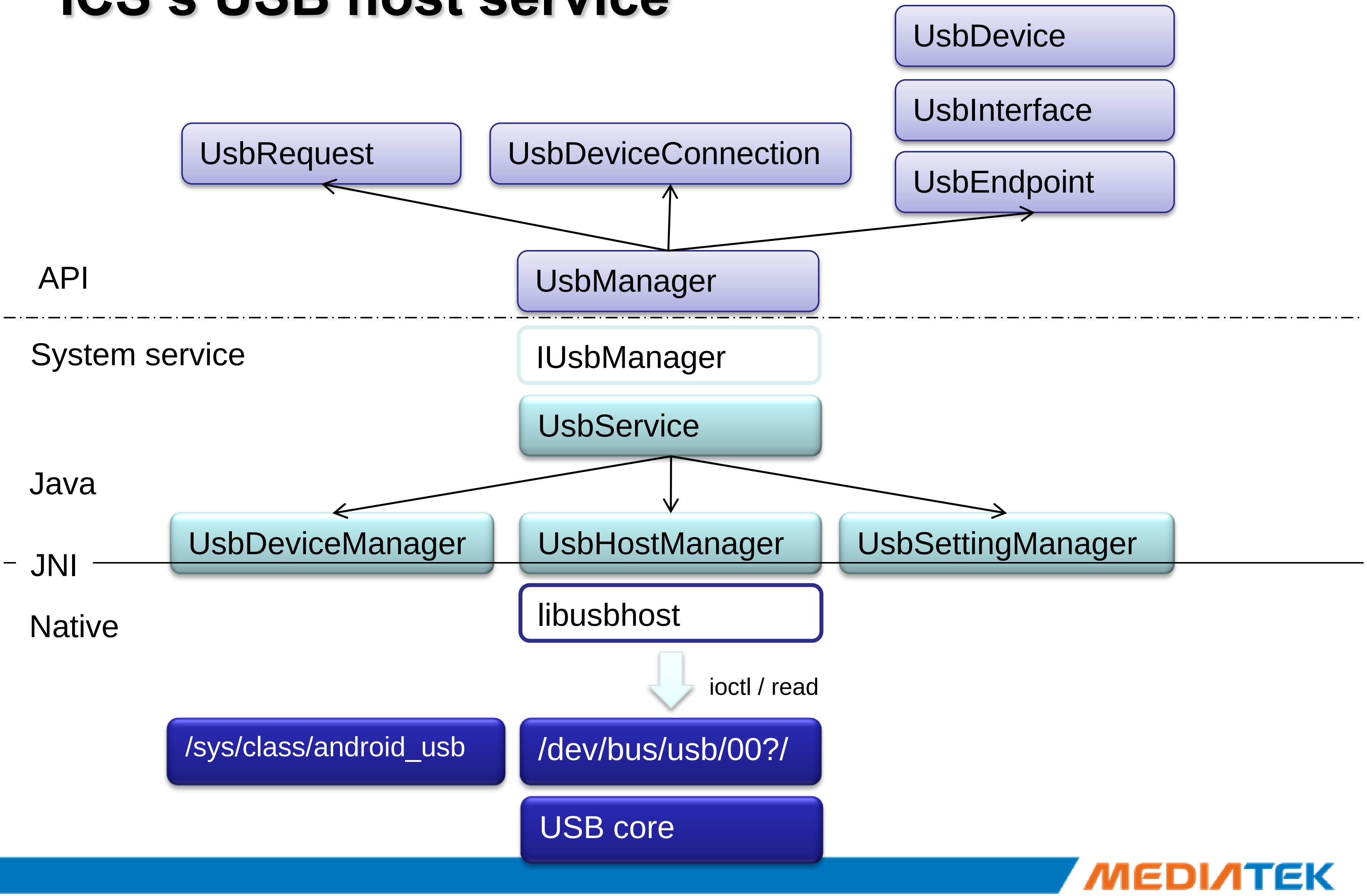
Customization

- There are two custom files in folder /alps/mediatek/custome/mt6575/kernel/usb/src/
  - mtk_usb_custom.c
    - USB phy operation and clock control.
  - mtk_usb_custom.h
    - VID/PID and usb strings customization.

Android ICS enhancement

- In Android ICS, USB device is supported in 2 ways
  - HID input device (such as keyboard, mouse, joystick) and USB mass storage device (known as U-disk) are supported by dedicated function drivers inside Linux kernel. All protocol is handled inside the driver.
    - upper level applications only see standard/abstract input device and storage volume.
  - Other devices (for example, a camera works as PTP device) is supported via applications (for example, Gallery). Kernel space USB drivers only send/receive raw data through USB bus, just as a pipe, all logical processing is done by user space program.
    - this is based on ICS's USB host service and related APIs
- So HID input devices and USB mass storage devices are "native" supported by default. But for other devices, you need to install proper application first.

ICS's USB host service



Implemented Functions

- Detect A-cable and B-cable plug in/out .
- Detect other devices connected, such as keyboard, u disk and so on.
- Fully support keyboard, mouse, and U-disk.
- By default Android ICS support PTP device such as Camera.
- If user installs the corresponding application, Android ICS will support any standard USB device.

Limitations

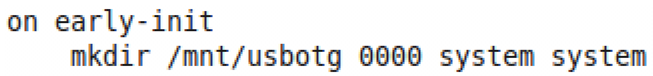
- PM has not implemented. That means if you plug in A-cable, phone will not suspend until A-cable is plugged out. And on the other hand, after you plug in a USB device, phone will not send suspend/resume signal on USB bus anyway.
- HNP(Host Negotiation Protocol) is not implemented in our product driver. So phone will only work as host after A-cable is plugged in. Beware this makes our product NOT fully compatible with USB OTG specification, but it will not impact daily use.

How to disable OTG feature 1

- Compile options
  - mediatek/config/$project/autoconfig/kconfig/project
    - change the following macros
      - # CONFIG_USB_MTK_OTG is not set
      - #CONFIG_USB_MTK_HDRC_HCD is not set
  - These 2 macros will disable USB controller driver’s OTG feature, but if you want to get a clean kernel, the following feature should also be disabled
    - HID → CONFIG_USB_HID
    - USB Mass Storage → CONFIG_USB_STORAGE
      - SCSI → CONFIG_SCSI
    - CDC-ACM → CONFIG_USB_ACM

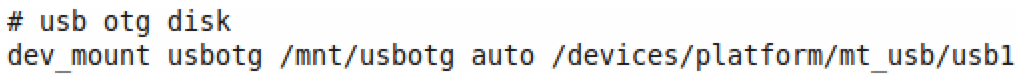
How to disable OTG feature 2

- VOLD rule item
  - mediatek/config/$project/init.project.rc
    - remove the following lines



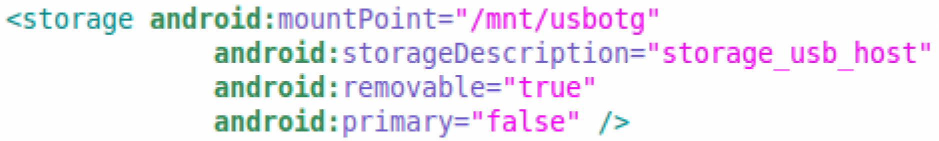
```
on early-init
    mkdir /mnt/usbotg 0000 system system
```

  - mediatek/config/$project/vold.fstab
    - remove the following lines



```
# usb otg disk
dev_mount usbotg /mnt/usbotg auto /devices/platform/mt_usb/usb1
```

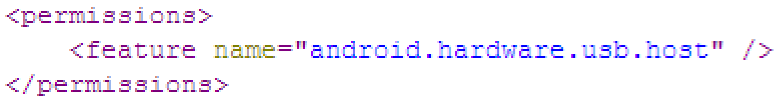
- storage_list.xml item
  - frameworks/base/core/res/res/xml/storage_list.xml
    - remove the following lines



```
<storage android:mountPoint="/mnt/usbotg"
             android:storageDescription="storage_usb_host"
             android:removable="true"
             android:primary="false" />
```

How to disable OTG feature 3

- feature xml
  - remove android.hardware.usb.host.xml under mediatek/config/$project/ folder
    - this will prevent Android start USB host service



```
<permissions>
    <feature name="android.hardware.usb.host" />
</permissions>
```

- If you want to enable OTG feature, just follow the reverse steps. But please make sure your hardware supports OTG (ex. VBUS supply and ID pin).

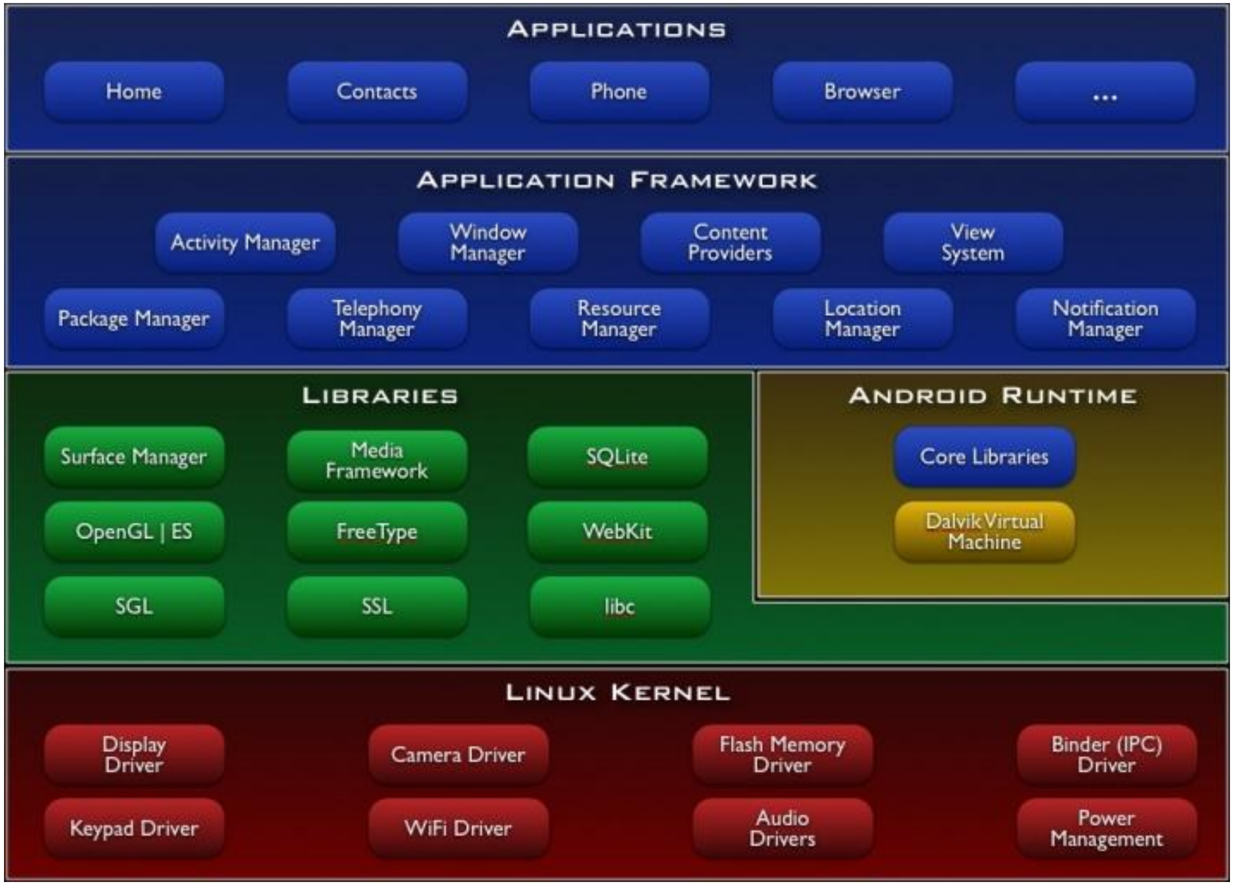
MEDIATEK

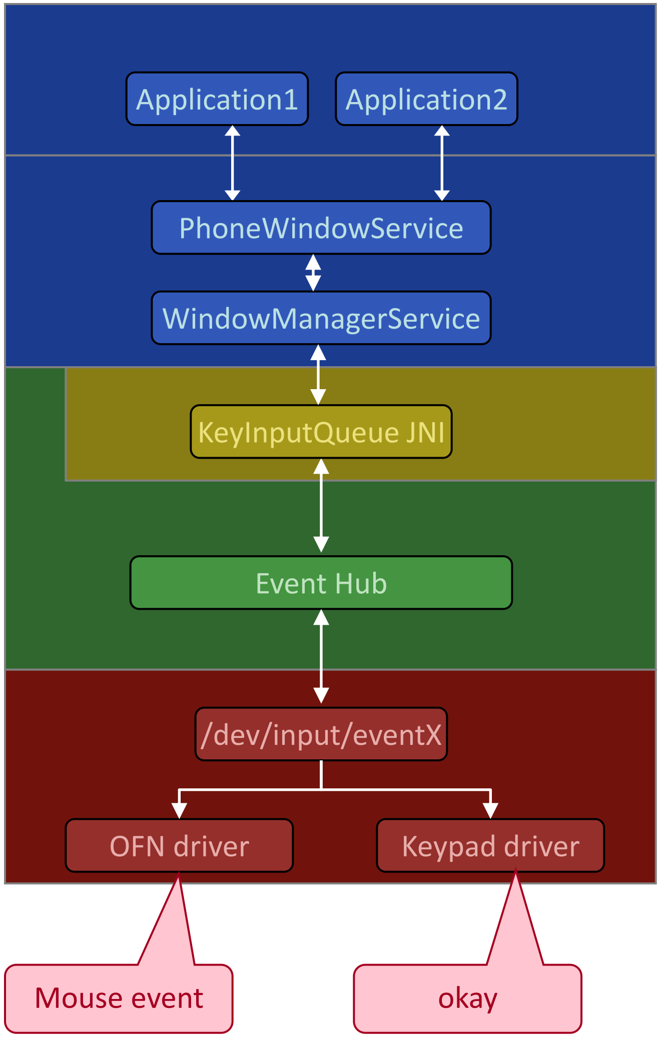
Appendix A

Outline

- OFN
- Jogball
- USB
- RTC

OFN Introduction





OFN Customization (1/4)

- Change file list

File Name	Location
cust_ofn.h	alps\mediatek\custom\common\kernel\ofn\inc
cust_ofn.c	alps\mediatek\custom\${BOARD}\kernel\ofn\${MODULE}

- Customization item
  - Overview

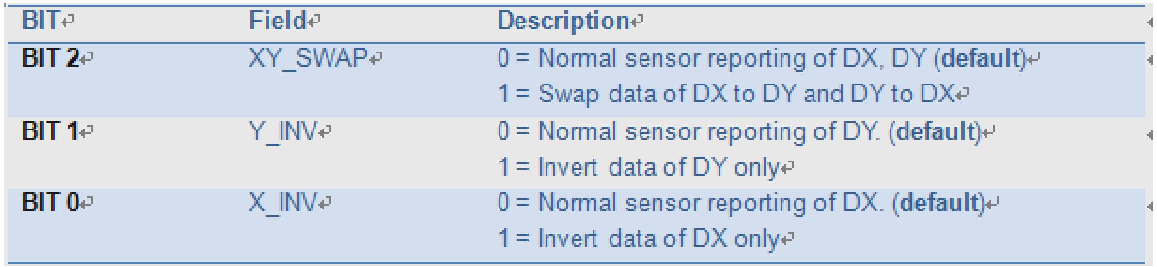
```
struct ofn_hw {
    int                power_id;
    int                power_vol;

    int                report_cls;
    OFN_ID             chip_id;
    int                slave_addr;
    int                i2c_num;
    unsigned int       layout;

    /*trackball class*/
    int                quan_x;
    int                quan_y;
    int                accu_max;

    /*keyboard class*/
                :
};
```

OFN Customization (2/4)

- power_id / power_vol
  - Customer could define power source of device according to layout
  - Please refer to the following file for power id and voltage
    - arch\arm\mach\include\mach-mt6516\include\mach\mt6516_pll.h
  - If the power source can’t be shutdown, please set the power_id as MT6516_POWER_NONE
- report_cls
  - Since Android expects OFN acts as a mouse, please set it as OFN_CLASS_TRACKBALL
- chip_id
  - For different model, the initialization sequence differs. Please choose the correct chip id, or the OFN could not be correctly enabled
- slave_addr
  - The i2c slave address depends on layout. Please fill the correct i2c slave address in different platform

OFN Customization (3/4)

- i2c_num
  - Customer can define the I2C number used by OFN
  - The value could be defined as 0 ~ 2
- layout
  - The data reported from register will vary depends on device layout. The field is a three bit binary. Please see the following definition:



BIT	Field	Description
BIT 2	XY_SWAP	0 = Normal sensor reporting of DX, DY (**default**) 1 = Swap data of DX to DY and DY to DX
BIT 1	Y_INV	0 = Normal sensor reporting of DY. (**default**) 1 = Invert data of DY only
BIT 0	X_INV	0 = Normal sensor reporting of DX. (**default**) 1 = Invert data of DX only

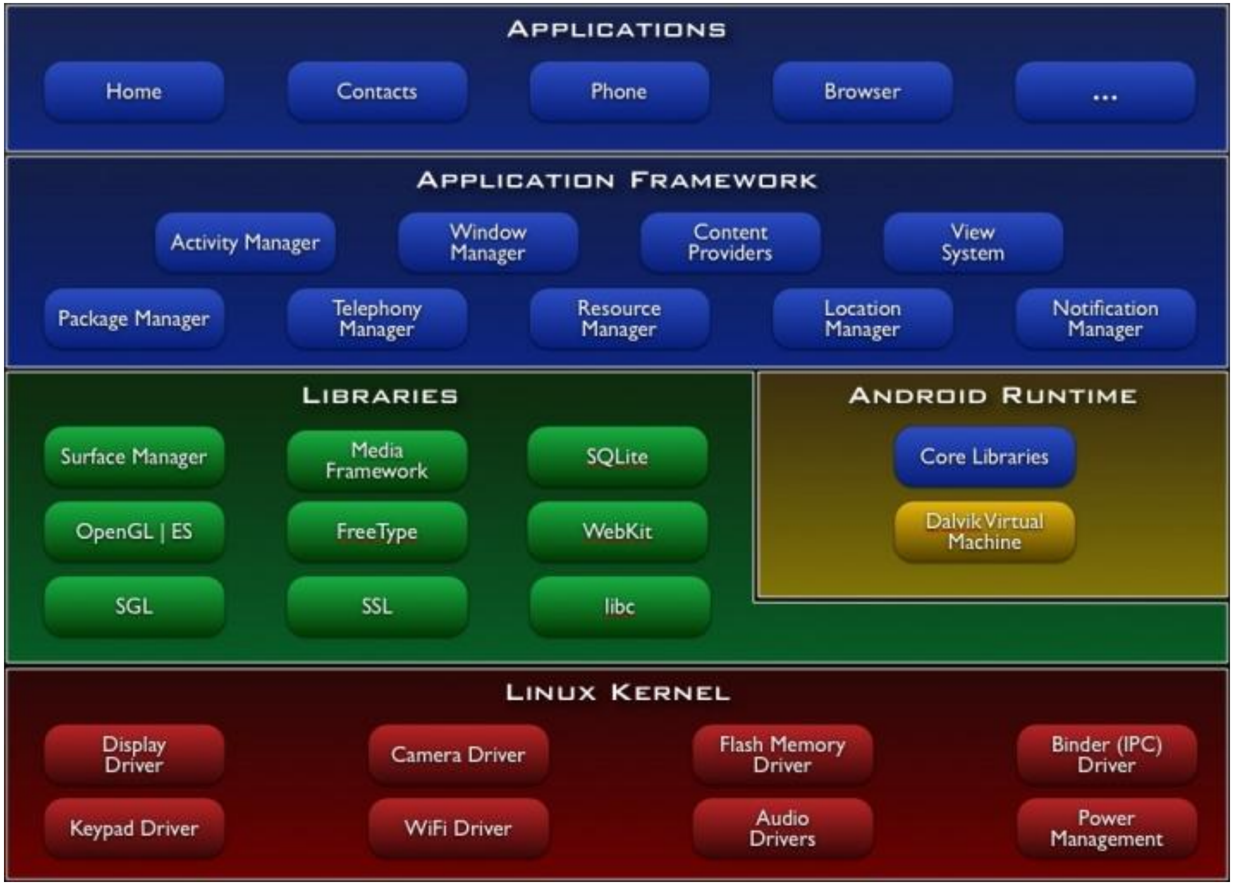
- quan_x / quan_y
  - The quatized step for x/y axis movement
  - To adjust the sensitivity
- accu_max
  - The maximum accumulated count in each motion interrupt
  - To suppress large motion

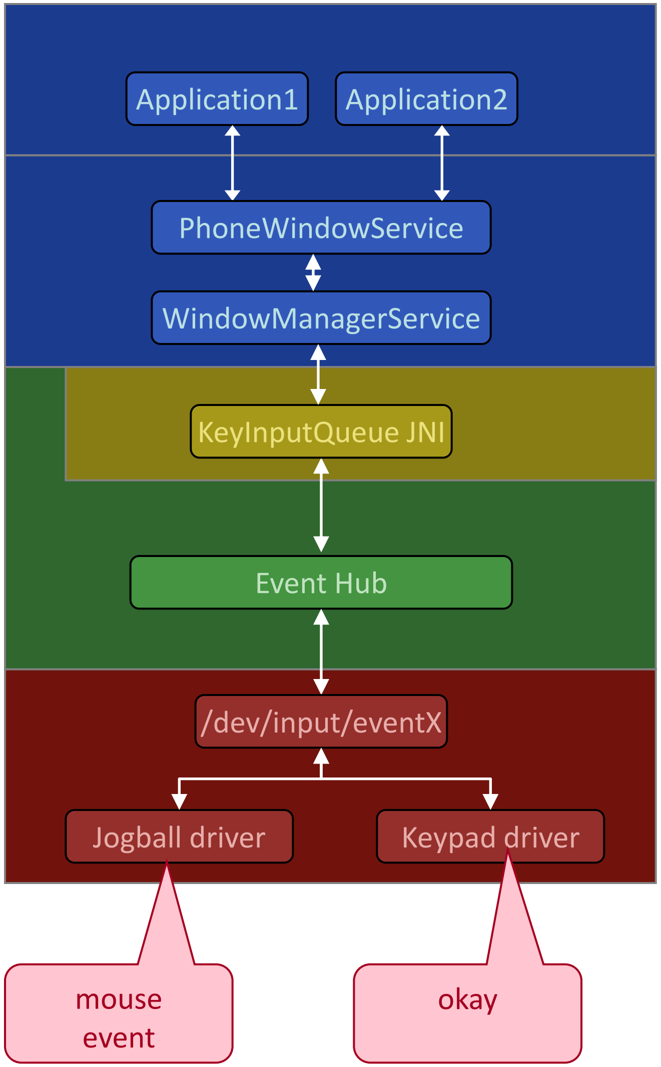
OFN Customization (4/4)

- DCT Customization

DCT definition	Description
GPIO_OFN_DWN_PIN	Shutdown pin. It’s important in power on / shutdown sequence
GPIO_OFN_RST_PIN	Reset pin. It’s important for power on sequence
GPIO_OFN_EINT_PIN	The external interrupt pin for detecting motion
CUST_EINT_OFN_NUM	The ID of external interrupt used for OFN
CUST_EINT_OFN_DEBOUNCE_CN	The debounce count of EINT pin. It's set as zero by default.
CUST_EINT_OFN_POLARITY	The polarity of EINT pin. It’s set as low by default.
CUST_EINT_OFN_SENSITIVE	The sensitivity of EINT pin. It’s set as level sensitive by default
CUST_EINT_OFN_DEBOUNCE_EN	The debounce enable of EINT pin. It’s set as disable by default

Jogball Introduction





Jogball Customization (1/3)

- Change file list

File Name	Location
cust_jogball.h	alps\mediatek\custom\common\kernel\jogball\inc
cust_jogball.c	alps\mediatek\custom\${BOARD}\kernel\jogball\${MODULE}

- Customization item
  - Overview

```
struct jogball_hw {
    int report_cls; /*refer to JBD_CLASS*/

    /*trackball class*/
    int gain_x;
    int gain_y;

    /*keyboard class*/
            :
};
```

  - report_cls
    - Since Android expects jogball acts as a mouse, please set it as JBD_CLASS_TRACKBALL.

Jogball Customization (2/3)

- gain_x / gain_y
  - They are the gain of x-axis and y-axis when detecting one movement
  - The value is currently set as 1 to keep the highest sensitivity

DCT Customization

DCT definition	Description
GPIO_JBD_INPUT_UP_PIN	The GPIO pin corresponding to up EINT.
GPIO_JBD_INPUT_LEFT_PIN	The GPIO pin corresponding to left EINT.
GPIO_JBD_INPUT_RIGHT_PIN	The GPIO pin corresponding to right EINT
GPIO_JBD_INPUT_DOWN_PIN	The GPIO pin corresponding to down EINT
CUST_EINT_HALL_1_NUM	The ID of up EINT
CUST_EINT_HALL_1_DEBOUNCE_CN	The debounce count, it will generally set as **0x01**
CUST_EINT_HALL_1_POLARITY	The polarity of external pin. It will be set as low level initially. Since any change between 0 and 1 means movement in jogball device, the polarity will be changed during runtime.
CUST_EINT_HALL_1_SENSITIVE	The sensitivity external pin. It will generally set as **edge sensitive** to detect the change between 0 and 1
CUST_EINT_HALL_1_DEBOUNCE_EN	Enable or Disable debounce. It will generally set as **enable** (1)

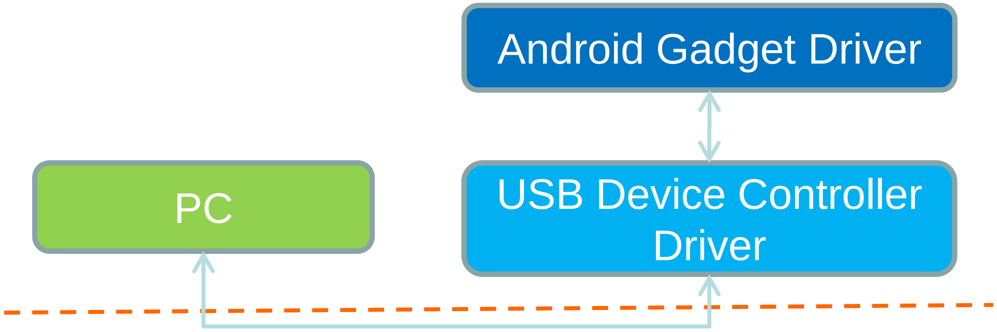
Jogball Customization (3/3)

- DCT Customization (cont.)

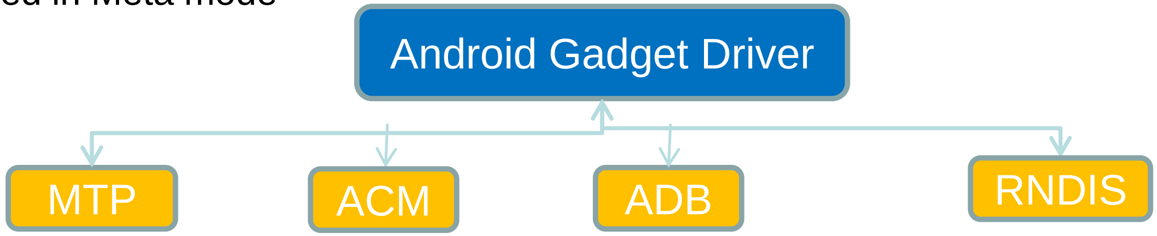
DCT definition	Description
CUST_EINT_HALL_2_NUM	The ID of left EINT
CUST_EINT_HALL_2_DEBOUNCE_CN	The debounce count, it will generally set as **0x01**
CUST_EINT_HALL_2_POLARITY	The polarity of external pin. It will be set as low level initially. Since any change between 0 and 1 means movement in jogball device, the polarity will be changed during runtime.
CUST_EINT_HALL_2_SENSITIVE	The sensitivity external pin. It will generally set as **edge sensitive** to detect the change between 0 and 1
CUST_EINT_HALL_2_DEBOUNCE_EN	Enable or Disable debounce. It will generally set as **enable** (1)
CUST_EINT_HALL_3_NUM	The ID of right EINT
CUST_EINT_HALL_3_DEBOUNCE_CN	The debounce count, it will generally set as **0x01**
CUST_EINT_HALL_3_POLARITY	The polarity of external pin. It will be set as low level initially. Since any change between 0 and 1 means movement in jogball device, the polarity will be changed during runtime.
CUST_EINT_HALL_3_SENSITIVE	The sensitivity external pin. It will generally set as **edge sensitive** to detect the change between 0 and 1
CUST_EINT_HALL_3_DEBOUNCE_EN	Enable or Disable debounce. It will generally set as **enable** (1)
CUST_EINT_HALL_4_NUM	The ID of down EINT
CUST_EINT_HALL_4_DEBOUNCE_CN	The debounce count, it will generally set as **0x01**
CUST_EINT_HALL_4_POLARITY	The polarity of external pin. It will be set as low level initially. Since any change between 0 and 1 means movement in jogball device, the polarity will be changed during runtime.
CUST_EINT_HALL_4_SENSITIVE	The sensitivity external pin. It will generally set as **edge sensitive** to detect the change between 0 and 1
CUST_EINT_HALL_4_DEBOUNCE_EN	Enable or Disable debounce. It will generally set as **enable** (1)

Overview

- Android USB device mode architecture



- Inside Android gadget driver, there are four interfaces
  - **MTP Interface**
    - No more Mass Storage Interface.
  - Android Debug Bridge Interface
    - Driver installation is needed for this function to work properly
  - RNDIS
  - ACM
    - A virtual COM port used in Meta mode



MTP (Media Transfer Protocol)

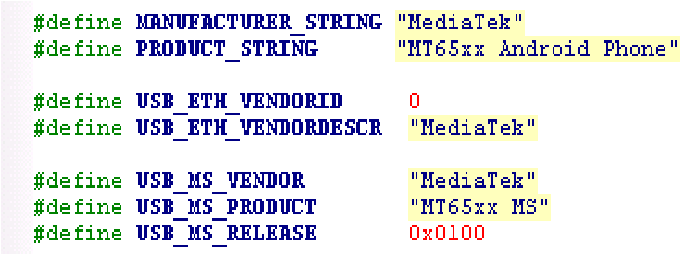
- MTP/PTP are introduced from Android 3.0.
- What does MTP support
  - Device
    - The MountService maintains a list of Volumn which is defined by stoage_list.xml
    - MTP will get the list from MountService via StorageManager and get the state of every partition
      - MTP add the partition with “Mounted” state into its list, no-matter the format of the partition
  - WindowsXP Scenario
    - Content Transfer to/from Devices.
    - Browsing Device Contents Using Windows Explorer.
    - Music and Video synchronzation with MediaPlayer.
    - Pictures files transfer.
  - Multi Storage
    - If there is multi-storage of the ALPS4.0 device. The MTP could have multi-storage with “Mounted Partition”, too.

Customization Items

- Change file list

File Name	Location
mtk_usb_custom.h	alps\mediatek\custom\mt6575\kernel\usb\src

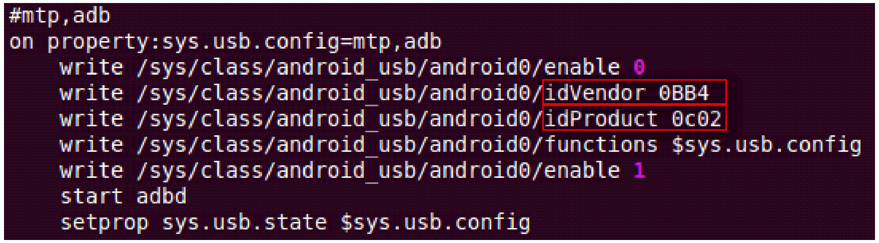
- Vendor Description



```
#define MANUFACTURER_STRING "MediaTek"
#define PRODUCT_STRING      "MT65xx Android Phone"

#define USB_ETH_VENDORID     0
#define USB_ETH_VENDORDESCR  "MediaTek"

#define USB_MS_VENDOR        "MediaTek"
#define USB_MS_PRODUCT       "MT65xx MS"
#define USB_MS_RELEASE       0x0100
```

File Name	Location
init.rc	alps\mediatek\config\mt6575

- PID/VID

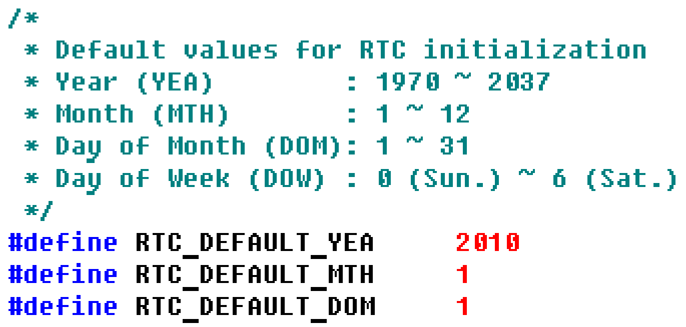


```
#mtp,adb
on property:sys.usb.config=mtp,adb
    write /sys/class/android_usb/android0/enable 0
    write /sys/class/android_usb/android0/idVendor 0BB4
    write /sys/class/android_usb/android0/idProduct 0c02
    write /sys/class/android_usb/android0/functions $sys.usb.config
    write /sys/class/android_usb/android0/enable 1
    start adbd
    setprop sys.usb.state $sys.usb.config
```

RTC in Preloader

- When initializing RTC HW, we will initialize RTC time counters by using a set of time

File Name	Location
Cust_rtc.h	Alps\mediatek\custom\[project]\preloader\inc



```
/*
 * Default values for RTC initialization
 * Year (YEA)        : 1970 ~ 2037
 * Month (MTH)       : 1 ~ 12
 * Day of Month (DOM): 1 ~ 31
 * Day of Week (DOW) : 0 (Sun.) ~ 6 (Sat.)
 */
#define RTC_DEFAULT_YEA     2010
#define RTC_DEFAULT_MTH     1
#define RTC_DEFAULT_DOM     1
```

RTC_DEFAULT_YEA	Year	1970 ~ 2037
RTC_DEFAULT_MTH	Month	1 ~ 12
RTC_DEFAULT_DOM	Day of Month	1 ~ 31

RTC in Kernel

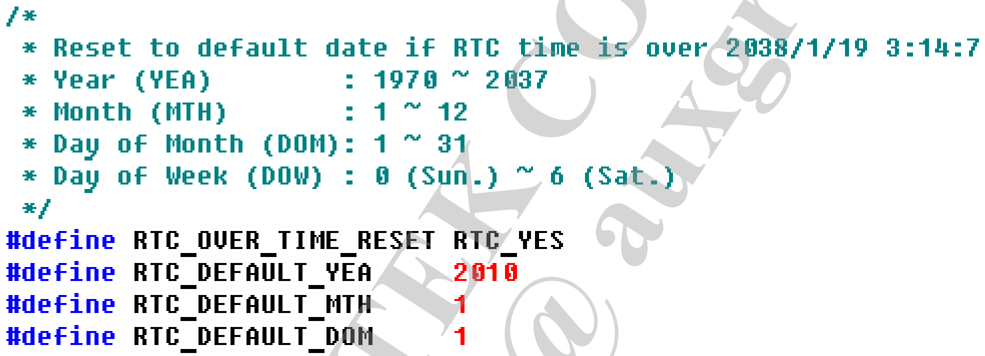
- The time in 32-bit Linux will overflow at 2038/01/19 03:14:07 but RTC time counters still keep running
  - If the user does not reset the system time, there may be abnormal exceptions occurred, especially after rebooting the system

- You can enable RTC_OVER_TIME_RESET which will reset RTC time counters when Kernel reads RTC time and RTC time is over 2038/01/19 03:14:07
  - If the user reboots the system, the system time will be the normal state, not overflow state

RTC in Kernel

- RTC_OVER_TIME_RESET (default: Yes)

File Name	Location
Rtc-mt6573.h	Alps\mediatek\custom\[project]\kernel\rtc\rtc



```
/*
 * Reset to default date if RTC time is over 2038/1/19 3:14:7
 * Year (YEA)        : 1970 ~ 2037
 * Month (MTH)       : 1 ~ 12
 * Day of Month (DOM): 1 ~ 31
 * Day of Week (DOW) : 0 (Sun.) ~ 6 (Sat.)
 */
#define RTC_OVER_TIME_RESET RTC_YES
#define RTC_DEFAULT_YEA     2010
#define RTC_DEFAULT_MTH     1
#define RTC_DEFAULT_DOM     1
```

RTC_DEFAULT_YEA	Year	1970 ~ 2037
RTC_DEFAULT_MTH	Month	1 ~ 12
RTC_DEFAULT_DOM	Day of Month	1 ~ 31

Confidential A

MEDIATEK

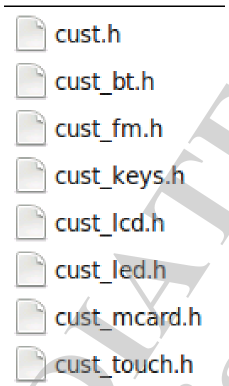
Appendix B

Outline

- Factory Key Mapping
- Recovery Key Mapping
- Modem Customization

Key mapping

- KEY define
  - Use Power key + Factory key to enter factory mode
    - Please use DCT – KEYPAD – MODE KEY for configuring Factory key.
  - Customization Files
    - Location: mediatek\custom\$(project)*factory\inc*\

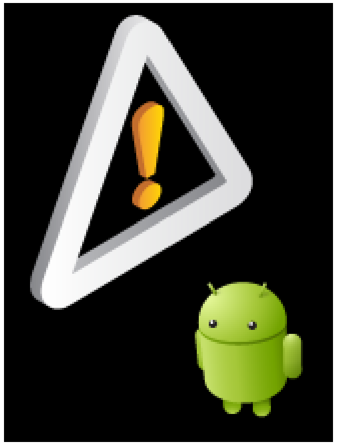


cust.h
cust_bt.h
cust_fm.h
cust_keys.h
cust_lcd.h
cust_led.h
cust_mcard.h
cust_touch.h

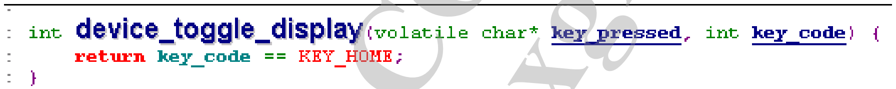
Key mapping

Press toggle key to display menu

- KEY define
  - Use Power key + Recovery keyt to enter recovery mode
    - Please use DCT – KEYPAD – MODE KEY for configuring Factory key.

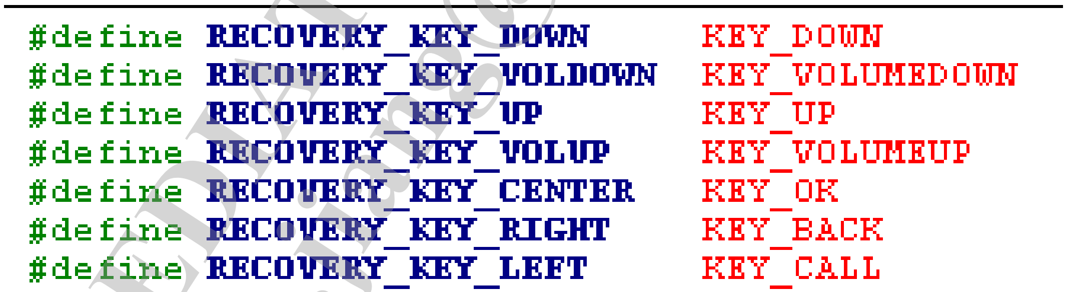


  - Change the toggle display key in the function
    - Location: bootable\recovery\default_recovery_ui.c



```
int device_toggle_display(volatile char* key_pressed, int key_code) {
    return key_code == KEY_HOME;
}
```

  - Define the keys in Recovery Mode
    - Location: alps\mediatek\custom\$(project)\recovery\inc\cust_keys.h



```
#define RECOVERY_KEY_DOWN       KEY_DOWN
#define RECOVERY_KEY_VOLDOWN    KEY_VOLUMEDOWN
#define RECOVERY_KEY_UP         KEY_UP
#define RECOVERY_KEY_VOLUP      KEY_VOLUMEUP
#define RECOVERY_KEY_CENTER     KEY_OK
#define RECOVERY_KEY_RIGHT      KEY_BACK
#define RECOVERY_KEY_LEFT       KEY_CALL
```

Recovery Process

- **Build update image step by step**
  - ./makeMtk <project> otapackage
  - Copy out /target/product/<project> /< project >-ota-<mode>.<user_id>.zip to root of the directory of SD card and rename it to update.zip
- **Upgrade Procedure**
  - Press volume up and powerkey to enter recovery mode
  - Press home key to enter main menu
  - Use volume down key to select apply sdcard:update.zip
  - Press menu key to execute to upgrade procedure

Modem Customization(1/2)

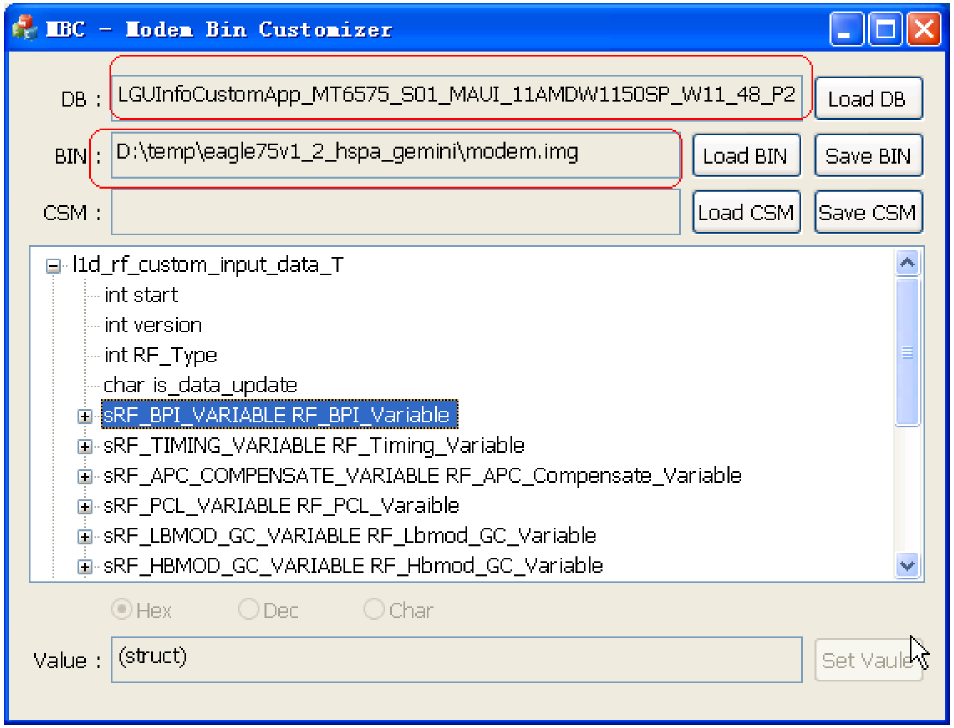
- Makefile Option (mediatek/config/${project}/projectConfig.mk)
  - CUSTOM_MODEM = eagle75v1_2_hspa_gemini (replace with your folder name)

- The modem image is placed in the below path:
  - alps/mediatek/custom/common/modem

Modem Customization(2/2)

- Use MBC to Customization

File	Description
mediatek/custom/common/modem/${CUSTOM_COMMON_MODEM}	
modem.img	The customization image file



DB : LGUInfoCustomApp_MT6575_S01_MAUI_11AMDW1150SP_W11_48_P2

BIN : D:\temp\eagle75v1_2_hspa_gemini\modem.img

CSM :

```
⊟ l1d_rf_custom_input_data_T
    int start
    int version
    int RF_Type
    char is_data_update
  ⊞ sRF_BPI_VARIABLE RF_BPI_Variable
  ⊞ sRF_TIMING_VARIABLE RF_Timing_Variable
  ⊞ sRF_APC_COMPENSATE_VARIABLE RF_APC_Compensate_Variable
  ⊞ sRF_PCL_VARIABLE RF_PCL_Varaible
  ⊞ sRF_LBMOD_GC_VARIABLE RF_Lbmod_GC_Variable
  ⊞ sRF_HBMOD_GC_VARIABLE RF_Hbmod_GC_Variable
```

● Hex ○ Dec ○ Char

Value : (struct)

MEDIATEK

Appendix C

Outline

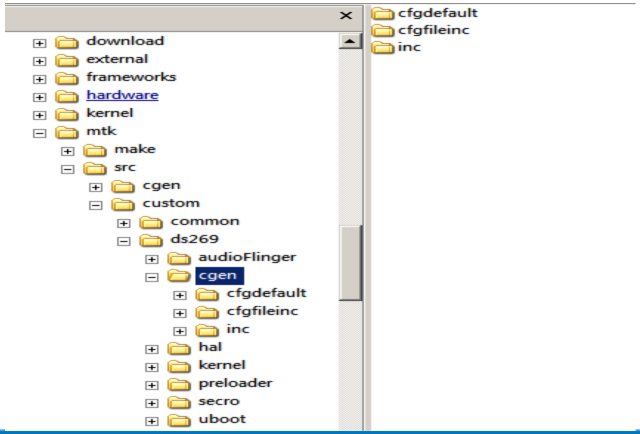
- NVRAM customization

Customization in NvRam

- For the different requirements of projects, NvRam modules also need to provide the supports of customization configurations, including default value and record data structure of NvRam files.
- There are two parts of NvRAM data
  - Common
    - For MTK platform NvRAM used
    - Customer can see the definition of related NVRAM record structure
    - But should not modify them
  - Customized for different projects
    - For customer NvRAM used
    - Customer can see the definition of related NVRAM record structure
    - Can modify them according to the requirements

Customization in NvRam

- The folder of NvRam customization is located in the path
  - *mediatek\custom\$(PROJECT)\cgen*
- There are three folders in this customization folder
  - Cfgdefault
    - Used to define the default value of NvRam files
  - Cfgfileinc
    - Used to define the record data structure of NvRam file
  - Inc
    - Used to support general NvRam module functionalities

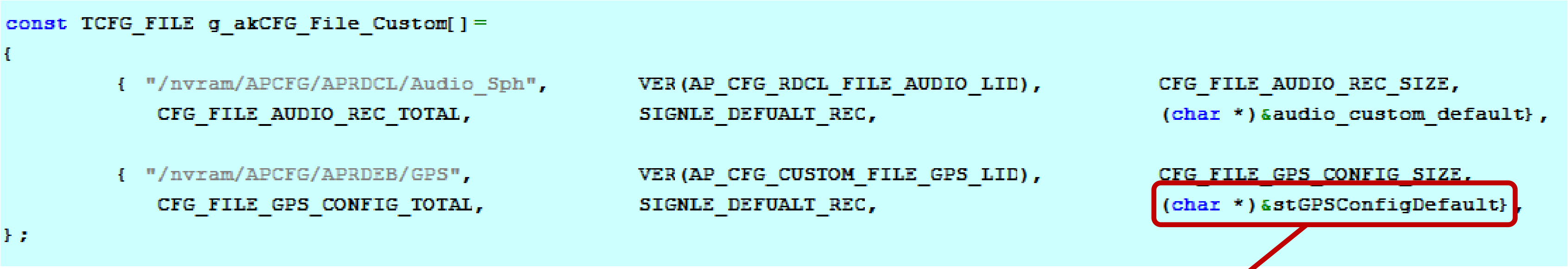


Customization in NvRam

- Should modify the file
  - *mediatek\custom\$(PROJECT)\cgen\inc\CFG_file_info_custom.h*
  - Data structure of *g_akCFG_File_Custom*
- The information of NvRAM file
  - File path
    - The file path that the NvRAM files should be store
  - File version
  - Record size
  - Record numbers
  - The type of the default value
  - The default value

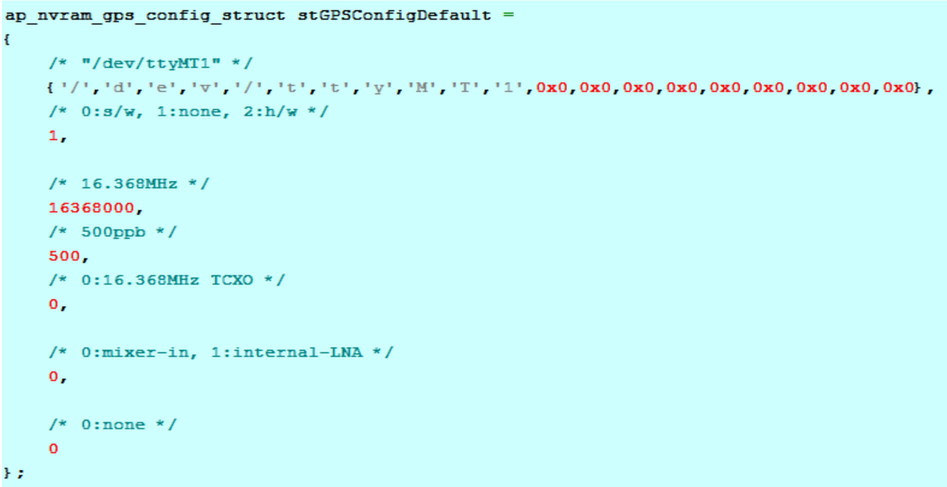
Customization in NvRam

- The data structure of *g_akCFG_File_Custom*



```
const TCFG_FILE g_akCFG_File_Custom[]=
{
        { "/nvram/APCFG/APRDCL/Audio_Sph",        VER(AP_CFG_RDCL_FILE_AUDIO_LID),        CFG_FILE_AUDIO_REC_SIZE,
           CFG_FILE_AUDIO_REC_TOTAL,               SIGNLE_DEFUALT_REC,                     (char *)&audio_custom_default},

        { "/nvram/APCFG/APRDEB/GPS",              VER(AP_CFG_CUSTOM_FILE_GPS_LID),        CFG_FILE_GPS_CONFIG_SIZE,
           CFG_FILE_GPS_CONFIG_TOTAL,              SIGNLE_DEFUALT_REC,                     (char *)&stGPSConfigDefault},
};
```

- The default value of *stGPSConfigDefault*



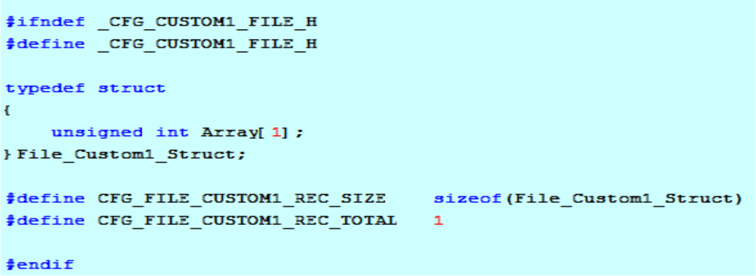
```
ap_nvram_gps_config_struct stGPSConfigDefault =
{
    /* "/dev/ttyMT1" */
    {'/','d','e','v','/','t','t','y','M','T','1',0x0,0x0,0x0,0x0,0x0,0x0,0x0,0x0,0x0},
    /* 0:s/w, 1:none, 2:h/w */
    1,

    /* 16.368MHz */
    16368000,
    /* 500ppb */
    500,
    /* 0:16.368MHz TCXO */
    0,

    /* 0:mixer-in, 1:internal-LNA */
    0,

    /* 0:none */
    0
};
```

Reset to Default

Type	Descriptions
SINGLE_DEFAULT_REC	If multiple records have same default value, this type should be used to minimize the Ram size. It only need define the default value of one record, NvRam module will use the default value of this record to initialize all of records
MULTIPLE_DEFAULT_REC	If NvRam has different default value for different records, this type should be used. It will use default value which is define in the cfg_file, then writes to NvRam file
DEFAULT_ZERO	The default value is 0, the property of default value will not be cared
DEFAULT_FF	The default value is 0xff, the property of default value will not be cared

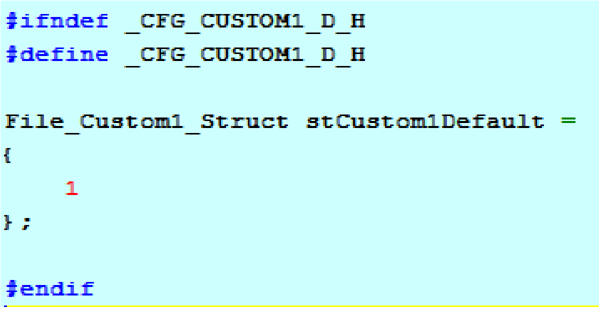
Step by Step to Add NvRAM Data

1. Add one header file which describes the definition of its record data structure, record size and record numbers
   - In the path of *mediatek\custom\$(PROJECT)\\cgen\cfgfileinc*



```
#ifndef _CFG_CUSTOM1_FILE_H
#define _CFG_CUSTOM1_FILE_H

typedef struct
{
    unsigned int Array[ 1] ;
}File_Custom1_Struct;

#define CFG_FILE_CUSTOM1_REC_SIZE     sizeof(File_Custom1_Struct)
#define CFG_FILE_CUSTOM1_REC_TOTAL    1

#endif
```

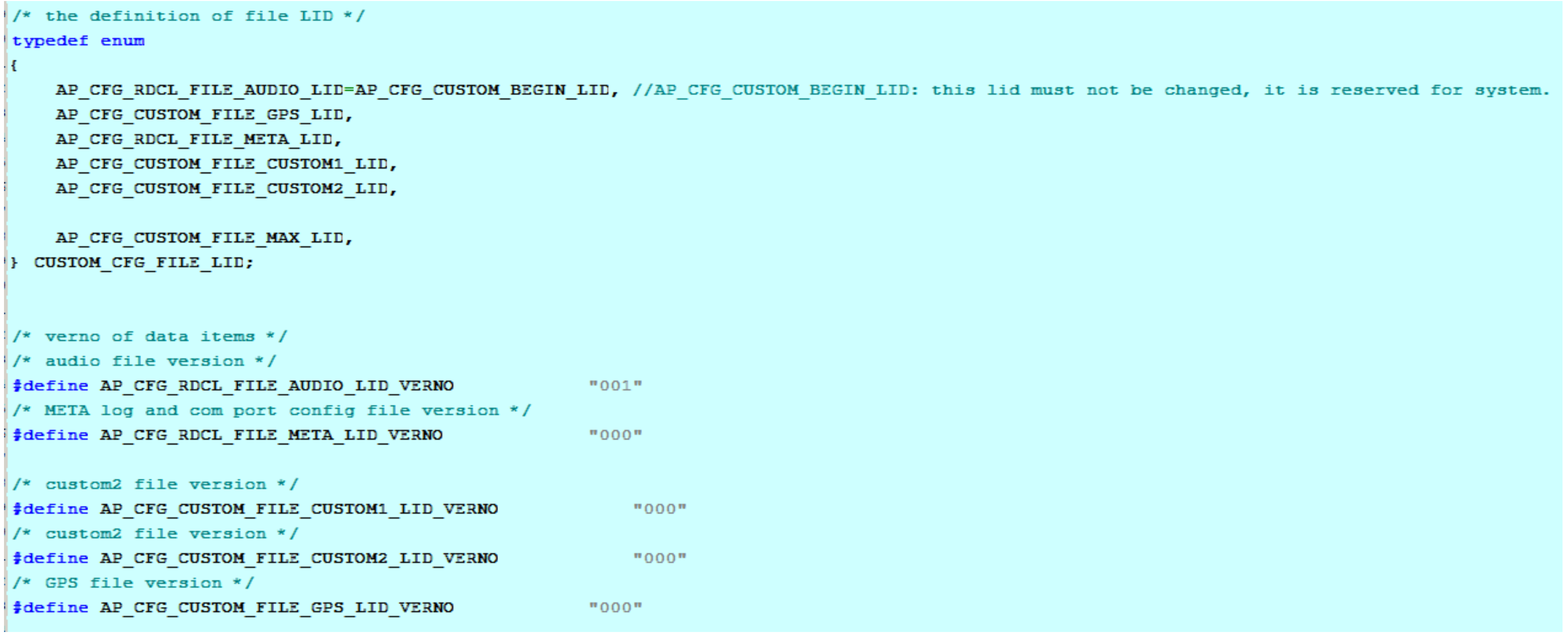
2. Add header file which define its default value of NvRam file
   - In the path of *mediatek\custom\$(PROJECT)\\cgen\cfgdefault*



```
#ifndef _CFG_CUSTOM1_D_H
#define _CFG_CUSTOM1_D_H

File_Custom1_Struct stCustom1Default =
{
     1
};

#endif
```

Step by Step to Add NvRAM Data

3. Add one lid in the enum defination of “*CUSTOM_CFG_FILE_LID*” and define the version number of NvRam file
   - In the path of *mediatek\custom\$(PROJECT)\cgen\inc\Custom_NvRam_LID.h*



```
/* the definition of file LID */
typedef enum
{
    AP_CFG_RDCL_FILE_AUDIO_LID=AP_CFG_CUSTOM_BEGIN_LID, //AP_CFG_CUSTOM_BEGIN_LID: this lid must not be changed, it is reserved for system.
    AP_CFG_CUSTOM_FILE_GPS_LID,
    AP_CFG_RDCL_FILE_META_LID,
    AP_CFG_CUSTOM_FILE_CUSTOM1_LID,
    AP_CFG_CUSTOM_FILE_CUSTOM2_LID,

    AP_CFG_CUSTOM_FILE_MAX_LID,
} CUSTOM_CFG_FILE_LID;


/* verno of data items */
/* audio file version */
#define AP_CFG_RDCL_FILE_AUDIO_LID_VERNO            "001"
/* META log and com port config file version */
#define AP_CFG_RDCL_FILE_META_LID_VERNO             "000"

/* custom2 file version */
#define AP_CFG_CUSTOM_FILE_CUSTOM1_LID_VERNO            "000"
/* custom2 file version */
#define AP_CFG_CUSTOM_FILE_CUSTOM2_LID_VERNO            "000"
/* GPS file version */
#define AP_CFG_CUSTOM_FILE_GPS_LID_VERNO            "000"
```

Step by Step to Add NvRAM Data

4. Add one include path which added in the step 1
   - In the path of *mediatek\custom\$(PROJECT)\cgen\inc\custom_cfg_module_file.h*
5. Add one include path which added in the step 2
   - In the path of

   mediatek\custom\$(PROJECT)\cgen\inc\custom_cfg_module_default.h
6. Add the related information of NvRam file into the definition of "*g_akCFG_File_Custom*"
   - In the path of *mediatek\custom\$(PROJECT)\cgen\inc\CFG_file_info_custom.h*
7. Add its related information, including record structure, NvRam lid, and record number
   - *In the path of mediatek\custom\$(PROJECT)\cgen\inc\Custom_NvRam_data_item.h*

Confidential A

MEDIATEK

Thank You!