

MEDIATEK

INTERNAL USE

MTK AOSP Build

MT2601

Android 4.4



Prerequisites

- MTK release requires the same build environment as AOSP (For Android 4.4.x, KitKat)
 - Ubuntu 10.04 64-bit
 - Sun JDK 1.6.0_45
 - Other packages listed in <http://source.android.com/source/initializing.html>

Code tree

Turnkey AOSP Code tree

Turnkey ABS	Turnkey AOSP
alps/ mediatek /	alps/ vendor / mediatek / proprietary /
alps/ mediatek / build /	alps/ device / mediatek / build / build /
alps/ mediatek /config/common/ ProjectConfig.mk alps/ mediatek /config/\${Platform}/ ProjectConfig.mk alps/ mediatek /config/\${Project}/ ProjectConfig.mk alps/ mediatek /config/ out /\${Project}/ ProjectConfig.mk	alps/ device / mediatek / build /config/\${Project}/ ProjectConfig.mk Example: alps/ device / mediatek / build /config/ flora01v1_phone / ProjectConfig.mk [Hint] No hierarchy in Turnkey AOSP's ProjectConfig.mk
alps/ mediatek / preloader /	alps/ bootable / bootloader / preloader /
alps/ kernel / alps/ mediatek / kernel / alps/ mediatek /platform/\${Platform}/ kernel / alps/ mediatek /custom/common/ kernel /	alps/ kernel-3.10 /

Turnkey AOSP Code tree (Cont.)

Turnkey ABS	Turnkey AOSP
<code>alps/vendor/mediatek/\${Project}/artifacts/</code>	<code>alps/vendor/\${Company}/libs/\${Project}/</code>
<code>alps/mediatek/config/common/*.rc</code>	<code>alps/device/mediatek/\${Platform}/*.rc</code>
<code>alps/mediatek/config/\${Platform}/*.rc</code>	<code>alps/device/\${Company}/\${Project}/*.rc</code>
<code>alps/mediatek/config/\${Project}/*.rc</code>	

Folder structure

- All MTK's **build-environment** files are located under device/mediatek/**build/**
- All Android **project configurations** are located under device/**\$Company/\$Project/**
 - **No custgen** to combine custom folders and ProjectConfig.mk in build time.
- All MTK **customization files** are located under *vendor/mediatek/proprietary/custom/\$Project/*
 - **MODEM** files are located under *vendor/mediatek/proprietary/custom/\$Project/modem/\$CUSTOM_MODEM/*

Folder structure (Cont.)

- All MTK **proprietary** is located under **vendor**/mediatek/**proprietary**/.
- All **kernel** files were located under kernel-3.10, not under mediatek/kernel/ folder.

Folder structure (Cont.)

- Configuration related

Turnkey ABS	Turnkey AOSP
<p>alps/build/target/product/common.mk alps/build/target/product/\$Project.mk - Define <i>PRODUCT_PACKAGES</i>, <i>PRODUCT_COPY_FILES</i>, <i>PRODUCT_PROPERTY_OVERRIDES</i>, etc.</p>	<p>alps/device/mediatek/common/device.mk alps/device/mediatek/\$Platform/device.mk alps/device/\$Company/\$Project/device.mk - Define <i>PRODUCT_PACKAGES</i>, <i>PRODUCT_COPY_FILES</i>, <i>PRODUCT_PROPERTY_OVERRIDES</i>, etc.</p>

Ex. call inherit-product要放在\$Project/device.mk和\$Platform/device.mk的最後面,才會有
Project > Platform > Common的效果

Folder structure (Cont.)

- All MTK **released libraries** will be located under vendor/\$Company/**libs**/\$Project/

Turnkey AOSP build

Turnkey AOSP build

- *Step 1: source build/envsetup.sh*

```
$ source build/envsetup.sh
including device/asus/tilapia/vendorsetup.sh
including device/asus/grouper/vendorsetup.sh
including device/asus/deb/vendorsetup.sh
including device/asus/flo/vendorsetup.sh
including device/mediatek/flora01v2/vendorsetup.sh
including device/mediatek/flora01v1/vendorsetup.sh
including device/mediatek/pandora01v1/vendorsetup.sh
including device/samsung/manta/vendorsetup.sh
```

- *Step 2: lunch & select \${Project}*

```
$ lunch
```

```
You're building on Linux
```

```
Lunch menu... pick a combo:
```

```
12. full_flora01v1-eng
13. full_flora01v1-userdebug
14. full_flora01v1-user
15. full_pandora01v1-eng
16. full_pandora01v1-userdebug
17. full_pandora01v1-user
18. aosp_manta-userdebug
19. mini_mips-userdebug
20. mini_x86-userdebug
21. mini_armv7a_neon-userdebug
22. aosp_mako-userdebug
23. aosp_hammerhead-userdebug
```

```
Which would you like? [aosp_arm-eng] 12
```

```
=====
PLATFORM_VERSION_CODENAME=REL
PLATFORM_VERSION=4.4.2
TARGET_PRODUCT=full_flora01v1
TARGET_BUILD_VARIANT=eng
TARGET_BUILD_TYPE=release
TARGET_BUILD_APPS=
TARGET_ARCH=arm
TARGET_ARCH_VARIANT=armv7-a-neon
TARGET_CPU_VARIANT=cortex-a7
HOST_ARCH=x86
HOST_OS=linux
HOST_OS_EXTRA=Linux-2.6.38-10-server-x86_64-with-Ubuntu-10.04-lucid
HOST_BUILD_TYPE=release
BUILD_ID=KOT49H
OUT_DIR=out
=====
```

Turnkey AOSP build (Cont.)

- Step 2: *lunch* `${Project}` also works

```
$lunch full flora01v1-eng
```



```
=====
PLATFORM_VERSION_CODENAME=REL
PLATFORM_VERSION=4.4.2
TARGET_PRODUCT=full_flora01v1
TARGET_BUILD_VARIANT=eng
TARGET_BUILD_TYPE=release
TARGET_BUILD_APPS=
TARGET_ARCH=arm
TARGET_ARCH_VARIANT=armv7-a-neon
TARGET_CPU_VARIANT=cortex-a7
HOST_ARCH=x86
HOST_OS=linux
HOST_OS_EXTRA=Linux-2.6.38-10-server-x86_64-with-Ubuntu-10.04-lucid
HOST_BUILD_TYPE=release
BUILD_ID=KOT49H
OUT_DIR=out
=====
```

- Step 3: *source* `./mbldenv.sh`
 - To setup MTK internal build environment parameters.

```
$ source mbldenv.sh
```

Customers need to modify this file according to customers' build environments.

Step 3': ***PATH=/mtkoss/jdk/jdk1.6.0_45/bin:\$PATH***

◆ If you cannot find `alps/mbldenv.sh`, please execute Step 3' instead.

Turnkey AOSP build (Cont.)

- **Step 4:** *make -j24 2>&1 | tee build.log*
 - -j24: depend on CPU numbers on your build machine.
 - Execute “*cat /proc/cpuinfo | grep processor | wc -l*” to get CPU numbers on your build machine.
 - If your machine has 8 CPUs, the suggested argument is “-j8”.
- preloader, lk and kernel will be built in the above command.

Turnkey AOSP build (Cont.)

- [Difference from ABS] NOT specify the project name in command line.
 - Project name was specified when executing **lunch** or **choosecombo**.

ABS	AOSP
... ./mk flora01v1 new	... lunch full_flora01v1-eng make -j36

- How to get build project?
 - get_build_var TARGET_DEVICE

```
[mtk80869@mbjswgbm543 KKW]$get_build_var TARGET_DEVICE  
flora01v1
```

Turnkey AOSP build (Cont.)

- [Difference from ABS] NOT use `./mk` or `./makeMtk` to wrap the build command.
 - Use the native build command with MTK's `pregen` and MTK's special build flows (Ex. `customimage`).
- Enable a feature option in build time
 - **ABS**: `./mk -o=MTK_AUTO_TEST=yes`
 - **AOSP**: `make MTK_AUTO_TEST=yes -j24`

All build systems can be built independently

- preloader, lk, kernel and Android can be built **independently**.
 - They can be built **without others**.
 - Example:
 - They **cannot share** the same device/mediatek/build/config/\$Project/**ProjectConfig.mk**

Build system	Project Configuration
preloader	bootable/bootloader/preloader/custom/\$Project/cust_bldr.mk
lk	bootable/bootloader/lk/project/\$Project.mk
kernel	kernel-3.10/arch/arm/configs/\$Project_debug_defconfig
Android	device/mediatek/build/config/\$Project/ProjectConfig.mk

All build systems can be built independently (Cont.)

- They can be build from Android.
 - preloader
 - `make -j24 pl 2>&1 | tee build.log`
 - lk
 - `make -j24 lk 2>&1 | tee build.log`
 - kernel
 - `make -j24 kernel 2>&1 | tee build.log`

All build systems can be built independently (Cont.)

- They can be build **independently**.
 - preloader
 - `cd bootable/bootloader/preloader`
 - `TARGET_PRODUCT=$project ./build.sh 2>&1 | tee build.log`
 - lk
 - `cd bootable/bootloader/lk`
 - `make -j24 $project 2>&1 | tee build.log`
 - kernel
 - `cd kernel-3.10`
 - `mkdir out`
 - `make ARCH=arm O=out $project_defconfig`
 - `make -j24 -k ARCH=arm O=out`

Android partial build command

- **Build android module**
 - **mmm** <directory>
 - example: **mmm** frameworks/base
- **Rebuild android module**
 - **mmm -B** <directory>
 - example: **mmm -B** frameworks/base
- **Build android module by name**
 - **make -j24** <module name>
 - example: **make -j24 libjpeg**

Clean commands

- Clean all
 - make **clean**
- Clean preloader
 - make **clean-pl**
- Clean lk
 - make **clean-lk**
- Clean kernel
 - make **clean-kernel**

Project Configuration

Project Configuration

- MTK's ***ProjectConfig.mk*** was kept and located at device/mediatek/build/config/**\$Project/**
 - \$Project is **unique** under different \$Company folders.
 - **Example**
 - device/mediatek/build/config/**flora01v1/ProjectConfig.mk**
 - ProjectConfig.mk's contexts **are the same with ABS's.**
 - ProjectConfig.mk **didn't support hierarchy.**
 - **No custgen** to generate the combined ProjectConfig.mk.

Project Configuration (Cont.)

- MTK's global compile options were kept.
- **ProjectConfig.mk** will be included by device/\$Company/\$Project/**full_\$Project.mk**
- **BoardConfig.mk**
 - Export compile options from **ProjectConfig.mk**

```
include_device/mediatek/${MTK_TARGET_PROJECT}/ProjectConfig.mk

MTK_INTERNAL_CDEFS ← $(foreach t, $(AUTO_ADD_GLOBAL_DEFINE_BY_NAME), $(if $(filter-out no NO none NONE false FALSE, ${$(t)}),
MTK_INTERNAL_CDEFS += $(foreach t, $(AUTO_ADD_GLOBAL_DEFINE_BY_VALUE), $(if $(filter-out no NO none NONE false FALSE, ${$(t)}),
MTK_INTERNAL_CDEFS += $(foreach t, $(AUTO_ADD_GLOBAL_DEFINE_BY_NAME_VALUE), $(if $(filter-out no NO none NONE false FALSE, ${$(t)}),

COMMON_GLOBAL_CFLAGS += $(MTK_INTERNAL_CDEFS)
COMMON_GLOBAL_CPPFLAGS += $(MTK_INTERNAL_CDEFS)
```

Project Configuration (Cont.)

- **Global compile options** may be **removed** in the future because they didn't follow AOSP rules.
- These compile options should be defined in module **Android.mk** individually.

Project Configuration (Cont.)

- device/**\$Company**/\$Project/device.mk,
device/**mediatek**/\$platform/device.mk,
device/**mediatek**/common/device.mk
 - You can use **ProjectConfig.mk's** feature options to wrap PRODUCT_COPY_FILES, PRODUCT_PROPERTY_OVERRIDES, PRODUCT_PACKAGES, etc.

```
85 #_GPS_relative_file
86 ifeq ($(MTK_GPS_SUPPORT),yes)
87     PRODUCT_COPY_FILES += frameworks/native
88 endif
```
 - Platform and common **device.mk** were located under **device/mediatek** folder and **shared** for **all customers**.

Pregen

Keep pregen with the build process in Turnkey AOSP (Under construction)

- Goal
 - To integrate MTK's **pregen** processes into AOSP build flows.
 - Follow **AOSP rules**.
 - Modify **less** native build scripts.
 - To generate files under **out** folder.

Keep pregen with the build process in Turnkey AOSP (Cont.) - ptgen

- **ptgen** was integrated with Android build flow already.
- Other build systems (preloader/lk/kernel) have not been integrated with ptgen yet.
 - **Input**
 - device/mediatek/build/build/tools/ptgen/\$Platform/partition_table_\$Platform.xls

Keep pregen with the build process in Turnkey AOSP (Cont.) - ptgen

- Output

Build system	ptgen output
preloader	out/target/product/\$Project/obj/BOOTLOADER_OBJ/inc/mt_partition.h
lk	out/target/product/\$Project/obj/BOOTLOADER_OBJ/inc/part_private.h out/target/product/\$Project/obj/BOOTLOADER_OBJ/inc/partition_define.h out/target/product/\$Project/obj/BOOTLOADER_OBJ/inc/pmt.h
kernel	out/target/product/\$Project/KERNEL_OBJ/PTGEN/inc/partition_define.h out/target/product/\$Project/obj/KERNEL_OBJ/PTGEN/inc/partition_dumchar.h out/target/product/\$Project/obj/KERNEL_OBJ/PTGEN/inc/pmt.h
Android	out/target/product/\$Project/\$Platform_Android_scatter.txt out/target/product/\$project/obj/PTGEN/partition_size.mk

Keep pregen with the build process in Turnkey AOSP (Cont.) - emigen

- **emigen** was a standalone tool temporally.
 - **Command**
 - *make PLATFORM=\$Platform PROJECT=\$Project -f device/mediatek/build/build/tools/emigen/emigen.mk*
 - Ex. *make PLATFORM=MT2601 PROJECT=flora01v1 -f device/mediatek/build/build/tools/emigen/emigen.mk*
 - **Input**
 - bootable/bootloader/preloader/custom/\${Project}/inc/custom_MemoryDevice.h
 - device/mediatek/build/build/tools/emigen/MT2601/MemoryDeviceList_MT2601.xls

Keep pregen with the build process in Turnkey AOSP (Cont.) - emigen

- **Output**

- bootable/bootloader/preloader/custom/\${Project}/custom_emi.c
- bootable/bootloader/preloader/custom/\${Project}/MTK_Loader_Info.tag
- bootable/bootloader/preloader/custom/\${Project}/inc/custom_emi.h

Keep pregen with the build process in Turnkey AOSP (Cont.) - drvgen

- **drvgen** was a standalone tool temporally.
 - **Command**
 - *make PROJECT=\$Project PLATFORM=\$Platform -f mediatek/build/build/tools/drvgen/drvgen.mk*
 - Ex. *make PROJECT=flora01v1 PLATFORM=mt2601 -f mediatek/build/build/tools/drvgen/drvgen.mk*
 - **Input**
 - kernel-3.10/arch/arm/mach-<PLATFORM>/<PROJECT>/dct/<CUSTOM_KERNEL_DCT>/codegen.dws
 - CUSTOM_KERNEL_DCT is from ProjectConfig.mk

Keep pregen with the build process in Turnkey AOSP (Cont.) - drvgen

- **Output**

Build System	Output files
preloader	bootable/bootloader/preloader/custom/<project>/inc/cust_kpd.h bootable/bootloader/preloader/custom/<project>/inc/cust_eint.h bootable/bootloader/preloader/custom/<project>/inc/custo_gpio_boot.h bootable/bootloader/preloader/custom/<project>/inc/cust_gpio_usage.h
lk	bootable/bootloader/lk/target/<project>/inc/cust_eint/h bootable/bootloader/lk/target/<project>/inc/cust_kpd.h bootable/bootloader/lk/target/<project>/inc/cust_gpio_usage.h
kernel	kernel-3.10/arch/arm/mach- <platform>/<project>/dct/<CUSTOM_KERNEL_DCT>/*
Android	vendor/mediatek/proprietary/custom/<project>/kernel/dct/*

Keep pregen with the build process in Turnkey AOSP (Cont.) - codegen

- **codegen** was integrated with Android build flow already.
 - **Standalone command**
 - *make -f device/mediatek/build/build/libs/codegen.mk TARGET_DEVICE=\${TARGET_DEVICE} **btcodegen cgen***
 - *Eg. make -f device/mediatek/build/build/libs/codegen.mk TARGET_DEVICE=flora01v1 **btcodegen cgen***
 - **Output**
 - *out/target/product/\${TARGET_DEVICE}/obj/CGEN/*
 - *Ex. out/target/product/flora01v1/obj/CGEN/*

MTK Flash Tool

MTK Flash Tool

Smart Phone Flash Tool(Runtime Trace Mode)

File Options Window Help

Welcome Format Download Readback MemoryTest

Download

Download-Agent C:\Users\mtk80869\Desktop\SP_Flash_Tool_v5.1505.02\MTK_AllInOne_DA.bin Download Agent

Scatter-loading File C:\Users\mtk80869\Desktop\flora01v2_ALPS_CAKKAW.p25_eng\MT2601_Android_scatter.tx Scatter-loading

Format All + Download

Name	Begin Address	End Address	Region	Location
preloader	0x0000000000000000	0x0000000000018533	EMMC_BOOT_1	C:\Users\mtk80869\Desktop\flora01v2_ALPS.C...
lk	0x0000000001ca0000	0x0000000001ce6947	EMMC_USER	C:\Users\mtk80869\Desktop\flora01v2_ALPS.C...
boot	0x0000000001d00000	0x0000000002198fff	EMMC_USER	C:\Users\mtk80869\Desktop\flora01v2_ALPS.C...
recovery	0x0000000002400000	0x00000000029517ff	EMMC_USER	C:\Users\mtk80869\Desktop\flora01v2_ALPS.C...
secro	0x0000000002b00000	0x0000000002b20fff	EMMC_USER	C:\Users\mtk80869\Desktop\flora01v2_ALPS.C...
logo	0x0000000002bc0000	0x0000000002bd5f45	EMMC_USER	C:\Users\mtk80869\Desktop\flora01v2_ALPS.C...
system	0x0000000004000000	0x00000000013dd2773	EMMC_USER	C:\Users\mtk80869\Desktop\flora01v2_ALPS.C...
cache	0x0000000003600000	0x0000000003660e093	EMMC_USER	C:\Users\mtk80869\Desktop\flora01v2_ALPS.C...
userdata	0x0000000003e000000	0x00000000040180333	EMMC_USER	C:\Users\mtk80869\Desktop\flora01v2_ALPS.C...

MT2601

0 B/s 0 Bytes EMMC High Speed 0:00 USB: DA Download All (high speed, auto detect)

④ Unplug&plug phone battery

⑤ connect phone with USB

Use DDMS to catch log (the same as AOSP)

The screenshot shows the Dalvik Debug Monitor (DDMS) interface. The top window displays a list of processes, with the selected process being `com.mec 1337` (PID 8601 / 8700). The bottom window shows a log of messages, including system events and application logs.

Process List:

Name	State	Version
0123456789	Online	4.4, debug
com.mec 881		8600
com.mec 1337		8601 / 8700
com.mec 1033		8602
com.and 1155		8603
com.and 1373		8604
com.and 854		8605
com.and 1431		8606
com.and 718		8607
com.and 1261		8608
system_ 641		8609
com.and 1393		8610
android.i 946		8611
com.mec 894		8612
com.and 911		8613

Log Messages:

L...	Time	PID	TID	Application	Tag	Text
D	01-01 00:46:45...	175	175		ADB_SERVICES	max_read_time=0.000072 max_send_time=0.000045
D	01-01 00:46:45...	175	175		ADB_SERVICES	adb local_socket_ready list (63) (36) (40)
D	01-01 00:46:46...	1373	1391	com.android.br...	dalvikvm	threadid=14: exiting
D	01-01 00:46:46...	175	175		ADB_SERVICES	adb fdevent_process list (15) (63) (40) (36) (44)
D	01-01 00:46:46...	1373	1391	com.android.br...	dalvikvm	threadid=14: bye!
I	01-01 00:46:50...	415	415		MUXD	[gsm0710muxd] 3518:main(): Frames received/droppe
D	01-01 00:46:50...	175	175		ADB_SERVICES	adb socket read list (63) (44) (36)
D	01-01 00:46:50...	175	175		ADB_SERVICES	read_data=947
D	01-01 00:46:50...	175	175		ADB_SERVICES	max_read_time=0.000071 max_send_time=0.000035
D	01-01 00:46:50...	175	175		ADB_SERVICES	adb local_socket_ready list (63) (44) (36)
I	01-01 00:46:50...	171	171		sn	Retry 9
I	01-01 00:46:50...	171	171		sn	Fail to access err=No such file or directory
I	01-01 00:46:50...	171	171		sn	Fail to access err=No such file or directory
I	01-01 00:46:50...	171	171		sn	Fail to access err=No such file or directory
I	01-01 00:46:50...	171	171		sn	Check all possible paths

MEDIATEK

everyday genius