

# AndroidP kernel board 创建及 Dtbo 介绍

---

版本：V1.0

日期：2018-12-27

Unisoc Confidential For hiar

# 目 录

版本历史 .....	1
1. Introduction .....	3
2. Board-config 创建 .....	3
2.1. kernel 相关修改 .....	3
2.1.1. 创建 board 相关属性文件 .....	3
2.2. Android Board 相关修改 .....	4
2.2.1. 删除 AndroidKernel.mk 文件 .....	4
2.2.2. 在 device 下 include sprd-board-config .....	5
3. System as root .....	5
3.1. 把 system 分区从 early mount 中删除 .....	5
3.2. 在 boot_args 中指定 root 分区为 system .....	6
4. Google Treble .....	6
4.1. DTBO .....	6
4.1.1. Overlay 文件创建 .....	6
4.1.2. Dtbo 文件编译 .....	7
5. Reference .....	8

## 版本历史

版本	日期	作者	备注
V1.0	2018.12.27	Ruifeng.Zhang	

Unisoc Confidential For hiar

## 声明

本文件所含数据和信息都属于紫光展锐所有的机密信息，紫光展锐保留所有相关权利。本文件仅为信息参考之目的提供，不包含任何明示或默示的知识产权许可，也不表示有任何明示或默示的保证，包括但不限于满足任何特殊目的、不侵权或性能。当您接受这份文件时，即表示您同意本文件中内容和信息属于紫光展锐机密信息，且同意在未获得紫光展锐书面同意前，不使用或复制本文件的整体或部分，也不向任何其他方披露本文件内容。紫光展锐有权在未经事先通知的情况下，在任何时候对本文件做任何修改。紫光展锐对本文件所含数据和信息不做任何保证，在任何情况下，紫光展锐均不负任何与本文件相关的直接或间接的、任何伤害或损失。

Unisoc Confidential For hiar

# 1. Introduction

在基于 AndroidP 分支上，Kernel 开始支持独立编译及 DTBO 功能。加入这两个功能后，与之前相比编译流程上有些区别，本文档主要提供在 AndroidP 下新建 kernel board 介绍。

## 2. Board-config 创建

### 2.1. kernel 相关修改

#### 2.1.1. 创建 board 相关属性文件

创建一个新的 board，必然有一些 board 独有的 config，这些 config 通过 diff-config 文件控制。product 加载哪些 diff-config，就需要与 board 相关的属性宏，这些属性宏原来定义在 device/sprd/project\_name 目录下，现在统一把这部分属性宏移动到 kernel 下，sprd-board-config 目录下。

操作步骤如下：

1) 创建 board 相关属性文件(以 sharkle gofu 为例)

```
sprd-board-config/sharkle/sp9832e_1h10_go/sp9832e_1h10_go_base  
sprd-board-config/sharkle/sp9832e_1h10_go/sp9832e_1h10_gofu
```

黄色：大系统名

绿色：product 名称，按原 device 下类别区分

红色：每个 product 属性文件必须创建出一个 base 文件，其余具体项目再基于此进行添加

蓝色：具体的 product 名称

2) 完善属性文件内容

sp9832e\_1h10\_go\_base 文件内容如下，其中红色部分需要根据具体 board 进行更改。

14~18 行，因是 base 文件，请确保通用性。若是修改只改:=后面的内容，不要删除前面的定义

```
1 TARGET_KERNEL_ARCH=arm  
  
2 ifdef BUILD_FROM_KERNEL  
3 KERNEL_OUT=.  
4 TARGET_ROOT_OUT=.  
5 KERNEL_DEFCONFIG=sprd_sharkle_defconfig  
6 KERNEL_DIFF_CONFIG_ARCH := sprd-diffconfig/sharkle/$(TARGET_KERNEL_ARCH)  
7 KERNEL_DIFF_CONFIG_COMMON := sprd-diffconfig/sharkle/common  
8 else  
9 KERNEL_OUT := $(TARGET_OUT_INTERMEDIATES)/KERNEL
```

```
10 KERNEL_DIFF_CONFIG_ARCH := kernel/sprd-diffconfig/sharkle/$(TARGET_KERNEL_ARCH)
11 KERNEL_DIFF_CONFIG_COMMON := kernel/sprd-diffconfig/sharkle/common
12 endif
13
14 BOARD_WCN_CONFIG :=
15 BOARD_EXT_PMIC_CONFIG :=
16 BOARD_TEE_CONFIG :=
17 PRODUCT_GO_DEVICE := true
18 BOARD_FEATUREPHONE_CONFIG :=
19
20 ifeq ($(strip $(BOARD_FEATUREPHONE_CONFIG)),true)
21 export BOARD_TEE_64BIT :=
22 else
23 export BOARD_TEE_64BIT ?= true
24 endif
```

sp9832e\_1h10\_gofu 文件内容如下，其中红色部分需要根据具体 board 进行更改。

7~10 行，为 gofu product 区别于其它 go 系列 product 的属性，此处为 overlay 方式的配置，会覆盖掉 base 下的原有定义，若有则增添，若无则删除

```
1 ifdef BUILD_FROM_KERNEL
2 include sprd-board-config/sharkle/sp9832e_1h10_go/sp9832e_1h10_go_base
3 else
4 include kernel/sprd-board-config/sharkle/sp9832e_1h10_go/sp9832e_1h10_go_base
5 endif
6
7
8 BOARD_TEE_CONFIG := trusty
```

## 2.2. Android Board 相关修改

### 2.2.1. 删除 AndroidKernel.mk 文件

已经把该文件移至 kernel 目录下，需要删除该文件，若已删除则跳过此项。

路径：device/sprd/sharkle/common/AndroidKernel.mk

删除此项后，还需要修改从 kernel 中 include 该文件，位于 AndroidBoard.mk 中

路径：device/sprd/sharkle/sp9832e\_1h10\_go/AndroidBoard.mk

```
-include device/sprd/sharkle/common/AndroidKernel.mk
+include kernel/AndroidKernel.mk
```

### 2.2.2. 在 device 下 include sprd-board-config

修改 device/sprd/sharkle/sp9832e\_1h10\_go/sp9832e\_1h10\_gofu.mk，在该文件首行添加

```
BOARD_PATH := kernel/sprd-board-config/sharkle/sp9832e_1h10_go/sp9832e_1h10_gofu
include $(BOARD_PATH)
```

## 3. System as root

### 3.1. 把 system 分区从 early mount 中删除

较之 AndroidO，AndroidP 将 system 分区从 early mount 中删除，相关代码如下：

arch/arm64/boot/dts/sprd/sharkle-board-common.dtsi

```
firmware {
    android {
        compatible = "android,firmware";

        vbmeta {
            compatible = "android,vbmeta";
            parts = "vbmeta,boot,recovery,system,vendor";
        };
        fstab {
            compatible = "android,fstab";
-           system {
-               compatible = "android,system";
-               dev = "/dev/block/platform/soc/soc:ap-ahb/20600000.sdio/by-name/system";
-               type = "ext4";
-               mnt_flags = "ro,barrier=1";
-               fsmgr_flags = "wait,avb";
-           };
            vendor {
                compatible = "android,vendor";
                dev = "/dev/block/platform/soc/soc:ap-ahb/20600000.sdio/by-name/vendor";
                type = "ext4";
                mnt_flags = "ro,barrier=1";
                fsmgr_flags = "wait,avb";
            };
        };
    };
};
```

## 3.2. 在 boot\_args 中指定 root 分区为 system

arch/arm64/boot/dts/sprd/sp9832e-1h10-native.dts

```
- root=/dev/ram0 rw
+ root=/dev/mmcblk0p30 rootfstype=ext4 rw rootwait
```

## 4. Google Treble

### 4.1. DTBO

Google 为了解决 Android 升级困难的问题，成立了 Google Treble 项目，它的主要目的是减少芯片厂商与 OEM/ODM 厂商的耦合。DTBO 就是为了实现 Google Treble 原理，在 kernel 中的实现。DTBO 把 OEM/ODM 厂商相关的外设放入一个单独的分区中，即 dtbo 分区。

#### 4.1.1. Overlay 文件创建

创建 overlay 模板文件并添加到 Makefile 中

Overlay dts 文件：arch/arm64/boot/dts/sprd/sp9832e-1h10-overlay.dts

```
/dts-v1/;
/plugin/;

/ {
    fragment {
        target-path = "/";
        __overlay__ {
            gpio-keys {
                compatible = "gpio-keys";

                key-volumedown {
                    label = "Volume Down Key";
                    linux,code = <KEY_VOLUMEDOWN>;
                    gpios = <&ap_gpio 124 GPIO_ACTIVE_LOW>;
                    debounce-interval = <2>;
                    wakeup-source;
                };
            };
        };
    };
};
```



```
&reserved_memory {
    fd_reserved: face-mem@fd1c0000{
        reg = <0x0 0xfd1c0000 0x0 0x01e00000>;
    };
};
```

1) 红色部分——编译器标识符

/dts-v1/; 表明编译时使用的 dts 版本

/plugin/; 表明使用 overlay 编译语法(允许在无节点时进行 overlay 操作)

这两个都是 overlay 的 “key”，相当于给编译器的一个标识

2) 绿色部分——新增节点

新增节点需要保留原有格式，fragment / target-path / \_\_overlay\_\_几个关键字格式必须保存，否则节点将不被识别。

3) 蓝色部分——已有节点 overlay

对于定义在 dts 文件中，若是 OEM/ODM 厂商需要修改其节点内容则使用这种方式。

其中 reserved\_memory 为节点 label，&为对节点进行重写的关键字。fd\_reserved 为子节点 label，

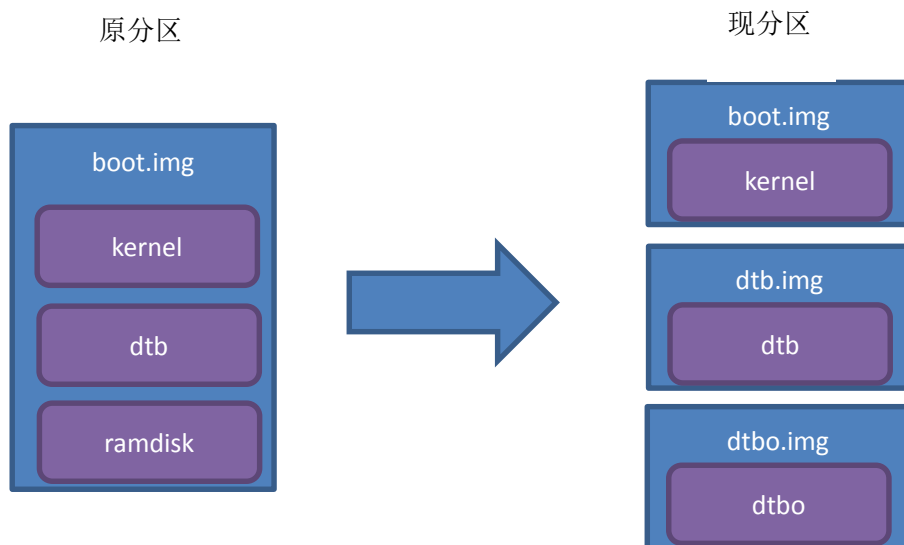
face-mem 为要修改节点的 name，reg 为要修改节点的具体属性。

将 overlay 添加到 Makefile 中，arch/arm64/boot/dts/sprd/Makefile

```
dtbo-$(CONFIG_ARCH_SPRD) += \
sp9832e-1h10-overlay.dtbo
```

## 4.1.2. Dtbo 文件编译

根据 Google Treble 原则，对原有的 boot.img 进行修改。



编译命令：在 Android 根目录下执行命令

boot.img	make bootimage
dtb.img	make dtbimage
dtbo.img	make dtboimage

## 5. Reference

- <https://developer.android.com/about/versions/pie/power#battery-saver>
- <https://developer.android.com/topic/performance/power/power-details?hl=zh-cn>

Unisoc Confidential For hiar