**如何在源码添加产品和lunch菜单**

**Author：linhui@yongyida.com**

目 录

[**一、** **产品菜单显示流程** 2](#_Toc457560481)

[**1.1** **build/envsetup.sh文件** 2](#_Toc457560482)

[**1.2** **添加菜单的方法** 2](#_Toc457560483)

[**二、** **注释无用的产品菜单（可选）** 3](#_Toc457560484)

[**三、** **仿照源码，添加device下的公司产品** 3](#_Toc457560485)

[**3.1 新建一个公司目录** 3](#_Toc457560486)

[**3.2 修改部分产品信息文件** 4](#_Toc457560487)

[**四、** **添加bootable中preLoader及lk相关目录** 7](#_Toc457560488)

[**4.1 preLoader** 7](#_Toc457560489)

[**4.2 lk** 7](#_Toc457560490)

[**五、** **添加vendor/mediate/proprietary下的目录** 9](#_Toc457560491)

[**5.1 custom目录** 9](#_Toc457560492)

[**5.2 trustzone目录** 10](#_Toc457560493)

[**5.3 bootanim目录** 10](#_Toc457560494)

[**六、** **添加vendor产品libs目录** 12](#_Toc457560495)

[**七、** **添加不同产品的应用目录** 12](#_Toc457560496)

[**八、** **使用宏控制Kernel选择编译** 12](#_Toc457560497)

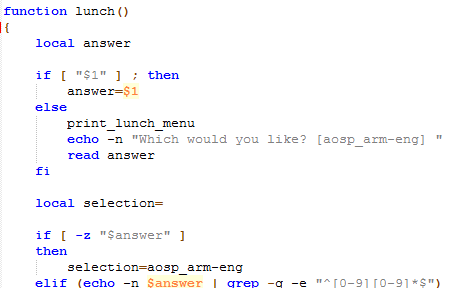
[**附：** 12](#_Toc457560498)

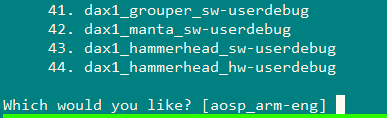
[**1.** **常用调试方法** 12](#_Toc457560499)

1. **产品菜单显示流程**
   1. **build/envsetup.sh文件**

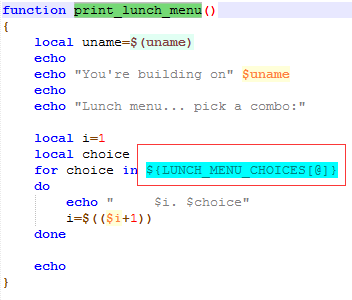
编译源码之前，我们会先执行source build/envsetup.sh，然后才有lunch命令和其他编译命令。

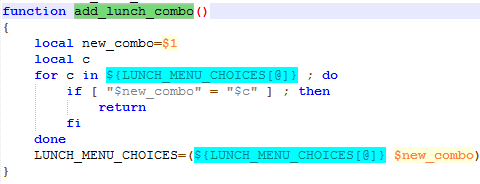
看看envsetup.sh这个文件里有什么。

从下图看到，有lunch命令的定义。有个熟悉的提示 “Which would you like? [aosp\_arm-eng]”，这是默认的选项，我们可以改成自己的产品



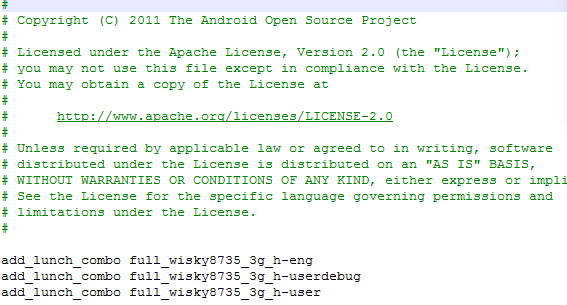
输入lunch命令后，可以看到类似右图

菜单是由print\_lunch\_menu打印出来的，菜单又是由数组变量LUNCH\_MENU\_CHOICES获得。这个变量在add\_lunch\_combo函数中赋值。



* 1. **添加菜单的方法**

添加菜单需要用到上面那个命令add\_lunch\_combo，脚本会搜索各目录，查找vendorsetup.sh文件，文件内容如下图，正如我们看到的lunch一样，其中有菜单名称。用同样的方法可以添加lunch菜单。后面将会演示如何添加菜单。

****

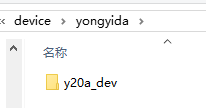
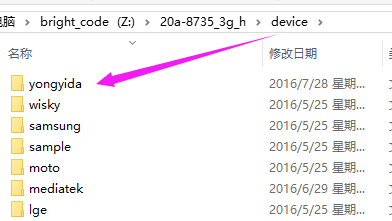
1. **注释无用的产品菜单（可选）**

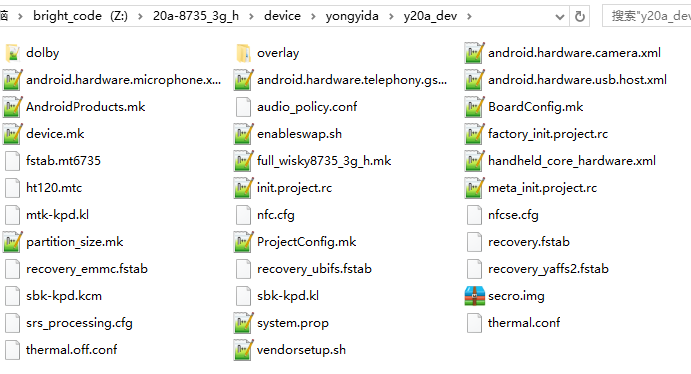
搜索不需要的lunch菜单，找到vendorsetup.sh，打开后，注释里面的add\_lunch\_combo，用“#”注释

1. **仿照源码，添加device下的公司产品**

**3.1 新建一个公司目录**

在里面继续新建产品目录。

产品信息参考device下其他产品，我们一直使用的是device\wisky\wisky8735\_3g\_h，可以把这个目录里的所有文件拷贝到上面我们新建的产品目录里。



**3.2 修改部分产品信息文件**

1. AndroidProducts.mk
2. BoardConfig.mk
3. device.mk，
4. full\_wisky8735\_3g\_h.mk，
5. vendorsetup.sh

* **AndroidProducts.mk**

将内容改成相应产品mk文件





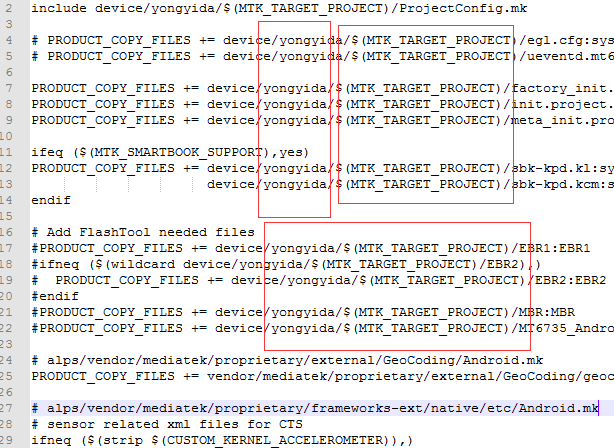
* **BoardConfig.mk**

同样也是替换产品公司和产品名称

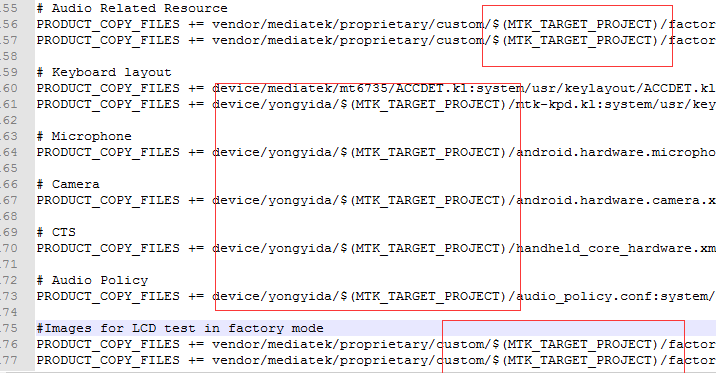


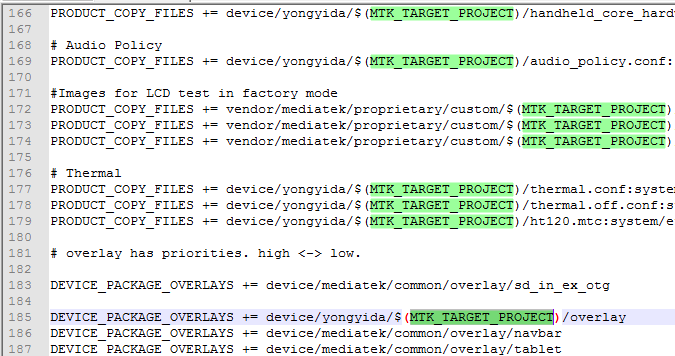
* **device.mk**

这里面很多关于配置文件，指定目录相关路径的，改成我们产品目录就好了。搜索device/ wisky, 替换成device/yongyida; 搜索wisky8735\_3g\_h，替换成通用的变量$(MTK\_TARGET\_PROJECT)。需要注意的是不能替换全部，容易替换错。

****

****

****

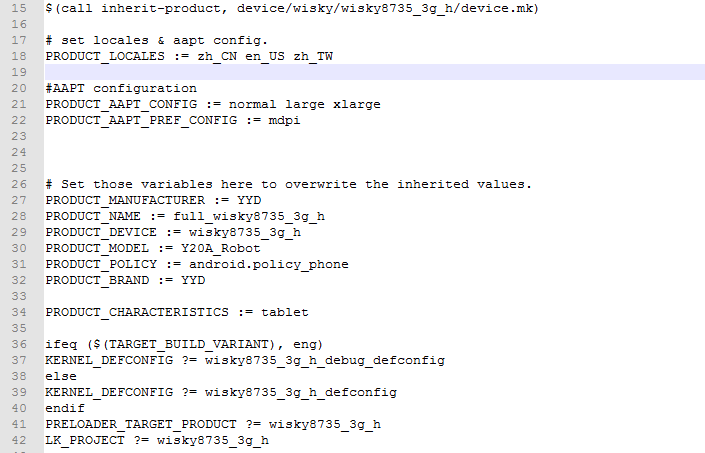
****

****

最后一条，227行的，需要在vendor/下新建一个yongyida目录，后面会讲到

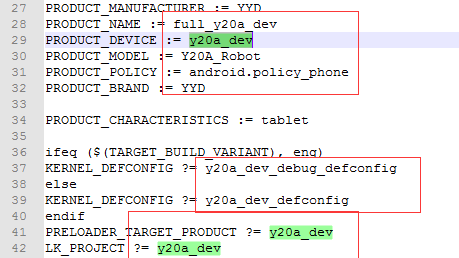
* **full\_wisky8735\_3g\_h.mk**

这个文件需要改文件名，如**full\_y20a\_dev.mk**。需要修改文件内容，对应路径和产品信息。KERNEL\_DEFCONFIG可以使用其他产品的文件，LK和PRELOADER因为需要定制，就修改为自己的产品。



修改为：



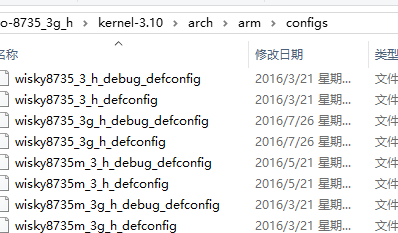


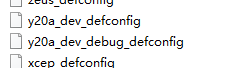
* **添加驱动defconfig文件**

打开路径：kernel-3.10/arch/arm/configs/

复制wisky8735\_3g\_h\_debug\_defconfig 和 wisky8735\_3g\_h\_defconfig

副本改名为y20a\_dev\_debug\_defconfig 和 y20a\_dev\_defconfig





* **vendorsetup.sh**

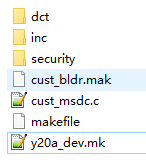
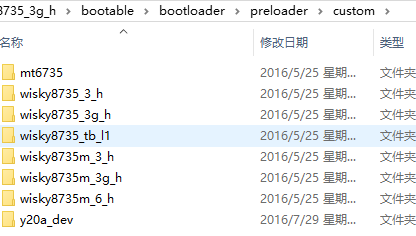
添加lunch菜单，就写到这里。



1. **添加bootable中preLoader及lk相关目录**

**4.1 preLoader**

目录位置：bootable/bootloader/preloader/custom，添加对应产品的custom目录。

****

从中复制文件到下，修改一个文件y20a\_dev.mk，修改文件内容TARGET=y20a\_dev

**4.2 lk**

lk主要是kernel的logo的地方需要更改几个地方：

**bootable/bootloader/lk/dev/logo/rules.mk**

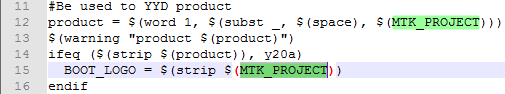
**bootable/bootloader/lk/dev/logo/y20a\_dev/**

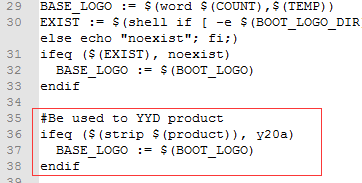
**bootable/bootloader/lk/project/y20a\_dev.mk**

**bootable/bootloader/lk/target/y20a\_dev/**

* **bootable/bootloader/lk/dev/logo/rules.mk**

识别当前产品，选择对应的目录进行编译。添加判断当前产品是y20a前缀的，并要在logo目录下建立与$(MTK\_TARGET\_PROJECT)对应的文件夹，里面的logo图片前缀也是$(MTK\_TARGET\_PROJECT)开头的。

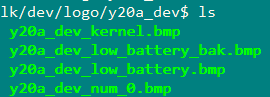
****

****

****

BASE\_LOGO也要判断是否是y20a的产品，如果是，BASE\_LOGO为产品名，对应查找以BASE\_LOGO为前缀的图片资源

* **bootable/bootloader/lk/dev/logo/y50b\_dev/**

****

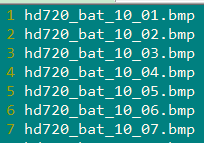
产品目录下是kernel中的开机logo，充电图标等。

从hd720或fwvga文件夹中拷贝过来，文件数量多达80多个，如何快速重命名这些文件？

* **编写重命名脚本。**

在控制台中输入命令 ls > ls.txt ，可以将ls的列表保存到ls.txt中，然后通过脚本执行重命名操作。



 ls.txt

**#!/bin/bash**

**for line in $(cat ls.txt)**

**do**

**old=$line**

**new=y20a\_dev\_"${line#\*hd720\_}"**

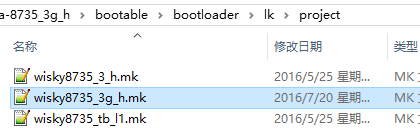
**mv $old $new**

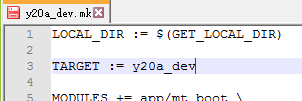
**done**

说明：将脚本保存到通过for循环，依次读取ls.txt的行字符串，通过${line#\*hd720\_}截取”hd720\_”后面的字符串。然后与”y20a\_dev\_”组合成新字符串给new变量，通过mv命令，将文件$old重命名为$new。

* **bootable/bootloader/lk/project/y20a\_dev.mk**

改一个地方：TARGET := y20a \_dev，对应产品名

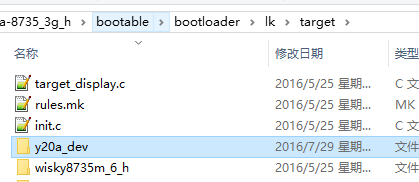




* **bootable/bootloader/lk/target/y50b\_dev/**

复制wisky8735\_3g\_h的文件到此，内容不需要改



1. **添加vendor/mediate/proprietary下的目录**

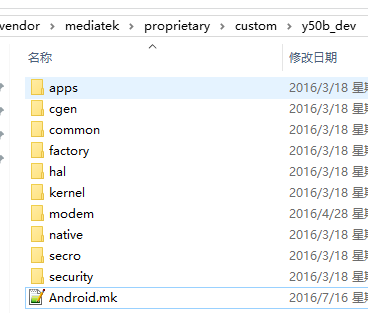
**5.1 custom目录**

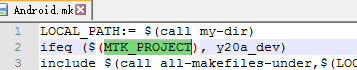
* 路径： **vendor/mediate/proprietary/custom/**

主要是底层的一些数据。

添加产品目录，拷贝wisky8735\_3g\_h的文件到此，修改目录中的Android.mk

ifeq ($(MTK\_PROJECT), y20a\_dev)



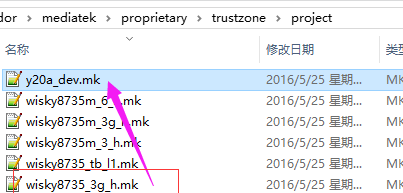


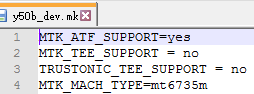
**5.2 trustzone目录**

* 路径： **vendor/mediatek/proprietary/trustzone/project/**

这个工程编译后，刷机的时候，对应的是tee1.img

添加mk文件





**5.3 bootanim目录**

* 路径：

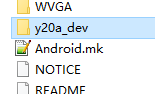
**vendor/mediatek/proprietary/operator/OP02/prebuilt/bootanim/bootaudio/**

**vendor/mediatek/proprietary/operator/OP02/prebuilt/bootanim/bootanimation/**

**vendor/mediatek/proprietary/operator/OP02/prebuilt/bootanim/shutaudio/**

**vendor/mediatek/proprietary/operator/OP02/prebuilt/bootanim/shutanimation/**

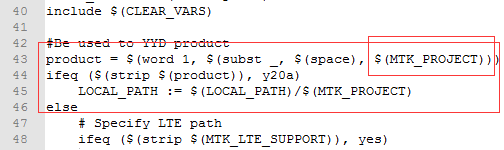
开关机动画，新建产品目录，区分每个产品不同的开关机动画



需要修改Android.mk文件，

* **vendor/mediatek/proprietary/operator/OP02/prebuilt/bootanim/bootanimation/Android.mk**
* **vendor/mediatek/proprietary/operator/OP02/prebuilt/bootanim/** **shutanimation /Android.mk**

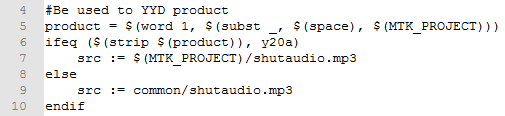
在原有的基础上，加个判断，如果是y20a的产品，跳转到对应的产品目录下选择编译。

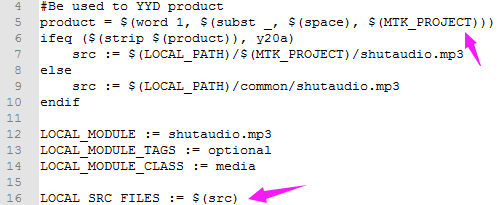




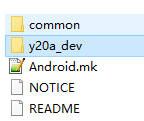
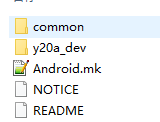
* **vendor/mediatek/proprietary/operator/OP02/prebuilt/bootanim/bootaudio/Android.mk**
* **vendor/mediatek/proprietary/operator/OP02/prebuilt/bootanim/shutaudio/Android.mk**

修改开关机共2个Android.mk，判断产品名称MTK\_TARGET\_PROJECT



****

开关机铃声也一样，建产品目录，里面放bootaudio.mp3文件

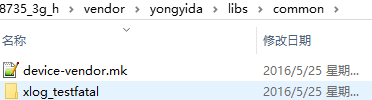
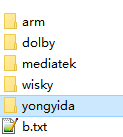
1. **添加vendor产品libs目录**

在vendor下模仿wisky新建一个目录，将wisky中的libs拷贝到yongyida目录。Libs目录很大，约1Gb，没有特别的需求，可以不同产品共享一个文件夹，需要修改device下的配置文件。

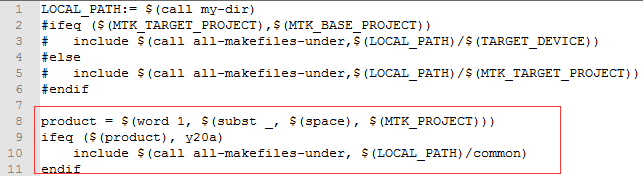
每个产品的device.mk: 最后一行修改为：

$(call inherit-product-if-exists, **vendor/yongyida/libs/common**/device-vendor.mk)

* 如果已经存在了，就不需要添加了。除非是特别定制的libs，可以多建一个目录。



另外在vendor/yongyida/libs目录也要拷贝wisky/libs/Android.mk到这里。这个文件是根据lunch选择产品判断是否需要编译这部分的libs的，如果没有这个mk文件，lunch其他产品也会编译这部分libs，这是不正确的。



1. **添加不同产品的应用目录**

* **路径：packages/apps/**

1. **使用宏控制Kernel选择编译**

**附：**

1. **常用调试方法**

* 在mk中打印调试信息：$(warning $(你的变量名)) 或 $(warning “调试信息：$(你的变量名)”)
* 获取当前产品 $(MTK\_PROJECT) 或 $(MTK\_TARGET\_PROJECT)
* 在源码命令行中查看变量值：**get\_build\_var** MTK\_PROJECT