## 安卓7.1移植过程

目录

[安卓7.1移植过程 1](#_Toc487041820)

[1. Jack server及jack错误处理 2](#_Toc487041821)

[1.1 Jack server优点 2](#_Toc487041822)

[1.2 Jack相关端口修改 2](#_Toc487041823)

[1.3 Jack编译遇到的问题 3](#_Toc487041824)

[1.3.1 build stopped: subcommand failed. 3](#_Toc487041825)

[1.3.2Communication error with Jack server (52) 4](#_Toc487041826)

[2. sdmc\_make.mk编译位置修改 4](#_Toc487041827)

[3. sdmc-libs修改 5](#_Toc487041828)

[4. 红外遥控适配 5](#_Toc487041829)

[5. 开机启动脚本 8](#_Toc487041830)

[6. 设置mac 属性 9](#_Toc487041831)

[7. 兼容S905D 1G 2G 9](#_Toc487041832)

[7.1修改device/amlogic/p230/Kernel.mk 9](#_Toc487041833)

[7.2修改board/amlogic/configs/gxm\_q200\_v1.h 9](#_Toc487041834)

[7.2修改board/amlogic/gxm\_q200\_v1/gxm\_q200\_v1.c 10](#_Toc487041835)

[7.4添加gxl\_p230\_1g.dts文件，直接拷贝gxl\_p230\_2g.dts 11](#_Toc487041836)

[8. S905D DVB移植 11](#_Toc487041837)

[8.1 Dvb相关遥控按键添加 11](#_Toc487041838)

[8.2 应用权限 12](#_Toc487041839)

[8.3 dtv中UI层遮挡视频层问题 13](#_Toc487041840)

[8.4 时移和录制回放问题 13](#_Toc487041841)

[8.5 32位与64位应用问题 14](#_Toc487041842)

## Jack server及jack错误处理

在编译的过程中，与以往系统不同的是Android7.0系统编译工具链使用了全新的Jack server来取代以往的编译器工具，所以在编译的过程中可能出现一些小问题。

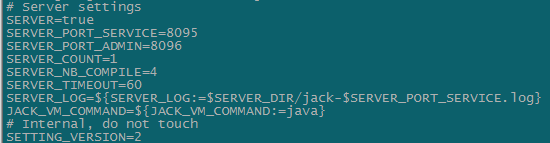
### ****1.1 Jack server优点****

1.适用于AOSP，意思就是说这套编译器是开源的，开发者也可以为其进行完善。   
2.Jack server能够减少编译的时间   
3.使用第三方包例如ProGuard将不在是必要的。

### ****1.2 Jack相关端口修改****

Jack拥有其自己的文件格式，即为：.jack。此文件包含了预编译dex代码的一些库文件，使得在预编译的过程中变的更快。

如图，我们打开$HOME路径下的.jack文件，看到的是如下参数：



其中：

SERVER=true 开启jack server.

SERVER\_PORT\_SERVICE=8072 设置TCP服务器端口号用来进行汇编编译

SERVER\_PORT\_ADMIN=8073 设置TCP服务器端口号用来管理员的编译管理

SERVER\_COUNT=1 当前未使用，一般为默认

SERVER\_NB\_COMPILE=4 同时编译最大线程

SERVER\_TIMEOUT=60 如果server没有在60s内没有反应，那么将会关掉此服务

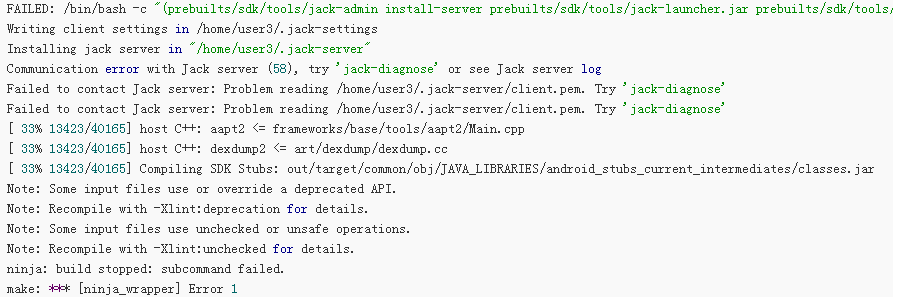
SERVER\_LOG=${SERVER\_LOG:=$SERVER\_DIR/jack-$SERVER\_PORT\_SERVICE.log} Log输出的位置

JACK\_VM\_COMMAND=${JACK\_VM\_COMMAND:=java} 启动JVM命令

### Jack编译遇到的问题

#### 1.3.1 build stopped: subcommand failed.

如不了解Jack编译的注意事项，经常会遇到以下问题

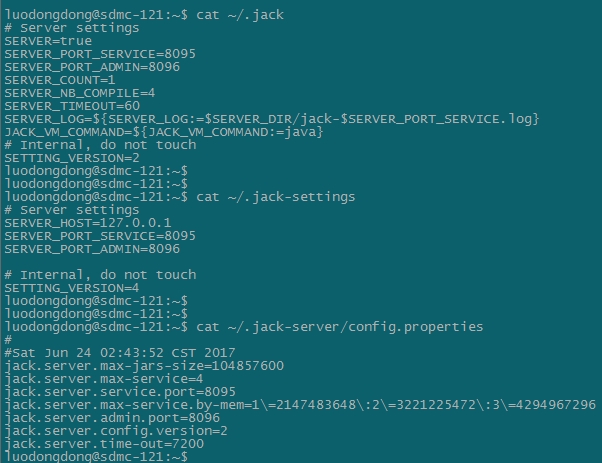


此问题是在于Android N新特性中，， 由于安卓7.1同一端口下不支持多用户同时编译，故在此分配一下jack的端口号，将同一个使用过的TCP端口将视为唯一，没编译过安卓7.1上需等编译一次后，再修改以下三个文件。

    如，我的端口号为PORT\_SERVICE：8095 和 PORT\_ADMIN：8096 设置如下：

    修改 cat ~/.jack  ~/.jack-settings  ~/.jack-server/config.properties 这三个地方对应的端口号：

    如图所示：



以上三个文件的端口一般只需要修改一次。

#### 1.3.2Communication error with Jack server (52)

如编译完7.1后需要编安卓6.0或5.1，需在编译安卓7.1终端的sdk根目录下执行jack-admin kill-server来杀掉jack服务，不然会报Communication error with Jack server (52)等的错误。

如编译7.1的终端关闭后，输入jack-admin kill-server会提示找不到jack-admin命令的错误，这个时候可以使用以下命令：

#cd prebuilts/sdk/tools

#jack-admin kill-server

## sdmc\_make.mk编译位置修改

在6.0和5.1上sdmc driver 以及预编译的sdmc\_make.mk的patch在

diff --git a/p212/AndroidBoard.mk b/p212/AndroidBoard.mk

index 508a3aa..31eb71d 100644

--- a/p212/AndroidBoard.mk

+++ b/p212/AndroidBoard.mk

@@ -16,3 +16,4 @@ LOCAL\_PATH := $(call my-dir)

include $(LOCAL\_PATH)/Kernel.mk

include device/amlogic/common/factory.mk

+include vendor/sdmc/drivers/sdmc\_make.mk

而在7.0以后移动到

diff --git a/common/factory.mk b/common/factory.mk

index 949a911..8750b6b 100644

--- a/common/factory.mk

+++ b/common/factory.mk

@@ -45,6 +45,7 @@ else# elif dts num == 1

cp -f $(KERNEL\_DEVICETREE\_BIN) $@

endif

$(hide) $(call aml-secureboot-sign-bin, $@)

+ $(MAKE) CROSS\_COMPILE=$(PREFIX\_CROSS\_COMPILE) -f vendor/sdmc/drivers/sdmc\_make.mk sdmcdrvs

@echo "Instaled $@"

.PHONY: dtbimage

## sdmc-libs修改

Android.mk

去除LOCAL\_STATIC\_LIBRARIES中的libstdc++项。

编译dm2016动态库时，使用预编译好的32位和64位的libdm2016\_d.so

include $(CLEAR\_VARS)

LOCAL\_MODULE := libdm2016\_d

LOCAL\_MULTILIB := both

LOCAL\_MODULE\_SUFFIX := .so

LOCAL\_MODULE\_TAGS := optional

LOCAL\_MODULE\_CLASS := SHARED\_LIBRARIES

LOCAL\_MODULE\_PATH\_32 := $(TARGET\_OUT)/lib/

LOCAL\_MODULE\_PATH\_64 := $(TARGET\_OUT)/lib64/

LOCAL\_SRC\_FILES\_arm := libdm2016\_s\_intermediates\_32/$(LOCAL\_MODULE)$(LOCAL\_MODULE\_SUFFIX)

LOCAL\_SRC\_FILES\_arm64 := libdm2016\_s\_intermediates\_64/$(LOCAL\_MODULE)$(LOCAL\_MODULE\_SUFFIX)

include $(BUILD\_PREBUILT)

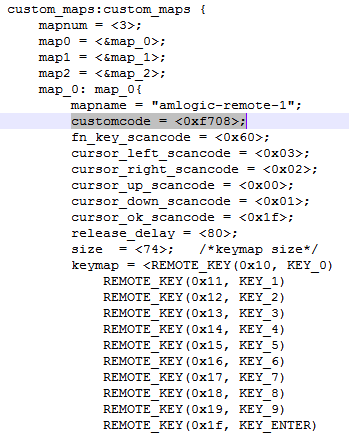
## 4. 红外遥控适配

安卓7.0以后不再走remote.conf这一套遥控器配制，故在安卓7.0上不再需要修改remote.conf文件，甚至可以直接删除。

而之前的remote.conf中物理码值和键值的对应关系移到了内核dts中

主要是修改mesongxl.dtsi文件，s912则修改mesongxm.dtsi文件

修改的地方如下图所示：



customcode = <0xf708>;对应的是遥控器机码

customcode = <0xf708>;

fn\_key\_scancode = <0x60>;

cursor\_left\_scancode = <0x03>;

cursor\_right\_scancode = <0x02>;

cursor\_up\_scancode = <0x00>;

cursor\_down\_scancode = <0x01>;

cursor\_ok\_scancode = <0x1f>;

这部份是鼠标键

size = <74>;这个数值很关键，其值是REMOTE\_KEY的数量+2，如它的值小于REMOTE\_KEY的数量的值，则剩下那部分的遥控器按键等于没适配。

例如OK键的映射过程如下：

其对应关系依次为

mesongxl.dtsi对应input.h

mesongxl.dtsi

REMOTE\_KEY(0x1f, KEY\_ENTER)

input.h

#define KEY\_ENTER97

input.h对应Generic.kl

key 28 DPAD\_CENTER

Generic.kl对应 KeyEvent.java

public static final int KEYCODE\_DPAD\_CENTER = 23;

接着修改framework层的res文件，把键值1，2....添加到全局资源中，这样，所有app应用都能识别这个键值了。

frameworks/base/core/res/res/values/attrs.xml

<enum name="KEYCODE\_SETTINGS" value="176" />

如荷兰客户需添加Setting按键，当按下Settting按键时，弹出高级设备界面

其适配如下：

mesongxl.dtsi对应input.h

sdk\common\arch\arm64\boot\dts\amlogic\mesongxl.dtsi中添加

REMOTE\_KEY(0X61, KEY\_SETTINGS)

sdk\common\include\dt-bindings\input.h

#define KEY\_SETTINGS 249

sdk\device\amlogic\p212\files\Generic.kl

key 249 SETTINGS

或者添加到Vendor\_0001\_Product\_0001.kl中

key 249 SETTINGS

sdk\frameworks\base\core\java\android\ KeyEvent.java

public static final int KEYCODE\_SETTINGS = 176;

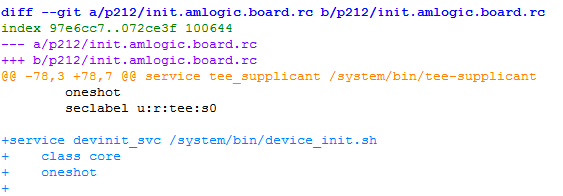
sdk/frameworks/base/services/core/java/com/android/server/policy/PhoneWindowManager.java

按settint键后具体实现请看如下patch

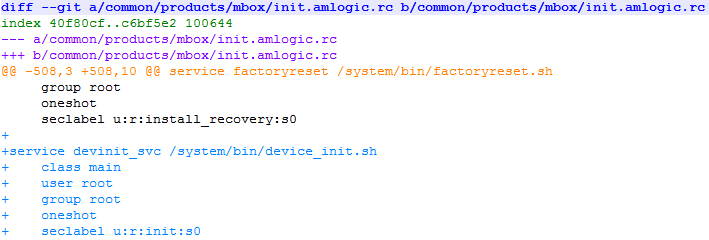
1. **diff --git a/services/core/java/com/android/server/policy/PhoneWindowManager.java b/services/core/java/com/android/server/policy/PhoneWindowManager.java**
2. **index 74e720f..6ee87a9 100644**
3. **--- a/services/core/java/com/android/server/policy/PhoneWindowManager.java**
4. **+++ b/services/core/java/com/android/server/policy/PhoneWindowManager.java**
5. **@@ -5941,6 +5941,10 @@ public class PhoneWindowManager implements WindowManagerPolicy {**
6. **}**
7. **break;**
8. **}**
9. **+ case KeyEvent.KEYCODE\_SETTINGS: {**
10. **+ startSystemSetting();**
11. **+ break;**
12. **+ }**
13. **}**
15. **if (useHapticFeedback) {**
16. **@@ -5953,6 +5957,16 @@ public class PhoneWindowManager implements WindowManagerPolicy {**
18. **return result;**
19. **}**
20. **+**
21. **+ private static final String TVSETTING\_PKG = "com.android.tv.settings";**
22. **+ private static final String TVSETTING\_CLS = "com.android.tv.settings.MainSettings";**
23. **+**
24. **+ private void startSystemSetting() {**
25. **+ Intent intent = new Intent();**
26. **+ intent.setClassName(TVSETTING\_PKG,TVSETTING\_CLS);**
27. **+ intent.addFlags(Intent.FLAG\_ACTIVITY\_NEW\_TASK);**
28. **+ mContext.startActivity(intent);**
29. **+ }**
31. **/\*\***
32. **\* Handle statusbar expansion events.**

## 5. 开机启动脚本

在android 6.0上开机启动脚本的服务添加如下图：



在android 7.0上需修改为：



如按6.0的patch，开机不会启动脚本。

同理，预安装脚本preinstall.sh也要如此处理。

## 6. 设置mac 属性

6.0中设置mac属性是在system/core中修改，在init进程中设置的

如果直接移到7.0上会一址报以下错误，卡在开机界面，进入不了launcher

request\_suspend\_state: wakeup (0->0) at 78392064577 (2015-01-01 00:01:21.597120118 UTC)

request\_suspend\_state,164,old\_sleep=0,new\_state=0

解决办法：

通过log分析到Init进程启动出错。

将mac属性的设置移到systemcontrol这个后台服务中

具体的patch见

\amlogic\_n\_gxx\0014-set-mac-sn-propperty\001-system-core-set-mac-sn-propperty.patch

\amlogic\_n\_gxx\0014-set-mac-sn-propperty\002-vendor-amlogic-frameworks-bootenv-set-mac-property.patch

## 7. 兼容S905D 1G 2G

### 7.1修改device/amlogic/p230/Kernel.mk

修改 KERNEL\_DEVICETREE := gxl\_p230\_2g为

KERNEL\_DEVICETREE := gxl\_p230\_1g gxl\_p230\_2g

### 7.2修改board/amlogic/configs/gxm\_q200\_v1.h

-#define CONFIG\_DDR\_CHANNEL\_SET CONFIG\_DDR0\_RANK01

+#define CONFIG\_DDR\_CHANNEL\_SET CONFIG\_DDR\_CHL\_AUTO

+#define CONFIG\_MULTI\_DTB 1

### 7.2修改board/amlogic/gxm\_q200\_v1/gxm\_q200\_v1.c

中添加dtb文件加载判断

1. **+#ifdef CONFIG\_MULTI\_DTB**
2. **+int checkhw(char \* name)**
3. **+{**
4. **+ unsigned int ddr\_size=0;**
5. **+ char loc\_name[64] = {0};**
6. **+ int i;**
7. **+ for (i=0; i<CONFIG\_NR\_DRAM\_BANKS; i++) {**
8. **+ ddr\_size += gd->bd->bi\_dram[i].size;**
9. **+ }**
10. **+#if defined(CONFIG\_SYS\_MEM\_TOP\_HIDE)**
11. **+ ddr\_size += CONFIG\_SYS\_MEM\_TOP\_HIDE;**
12. **+#endif**
13. **+ switch (ddr\_size) {**
14. **+ case 0x80000000:**
15. **+ strcpy(loc\_name, "gxl\_p230\_2g\0");**
16. **+ break;**
17. **+ case 0x40000000:**
18. **+ strcpy(loc\_name, "gxl\_p230\_1g\0");**
19. **+ break;**
20. **+ case 0x2000000:**
21. **+ strcpy(loc\_name, "gxl\_p230\_512m\0");**
22. **+ break;**
23. **+ default:**
24. **+ //printf("DDR size: 0x%x, multi-dt doesn't support\n", ddr\_size);**
25. **+ strcpy(loc\_name, "gxl\_p230\_unsupport");**
26. **+ break;**
27. **+ }**
28. **+ strcpy(name, loc\_name);**
29. **+ setenv("aml\_dt", loc\_name);**
30. **+ return 0;**
31. **+}**
32. **+#endif**

### 7.4添加gxl\_p230\_1g.dts文件，直接拷贝gxl\_p230\_2g.dts

注意拷贝后要修改以下这几个地方

\* arch/arm64/boot/dts/amlogic/gxl\_p230\_1g.dts 修改为

\* arch/arm64/boot/dts/amlogic/gxl\_p230\_2g.dts

amlogic-dt-id = "gxl\_p230\_1g"; ->

amlogic-dt-id = "gxl\_p230\_2g";

linux,usable-memory = <0x0 0x100000 0x0 0x3ff00000>; 修改为

linux,usable-memory = <0x0 0x100000 0x0 0x7ff00000>;

reg = <0x0 0x3e000000 0x0 0x2000000>;修改为

reg = <0x0 0x7e000000 0x0 0x2000000>;

logo\_addr = "0x3f851000";修改为

logo\_addr = "0x7f851000";

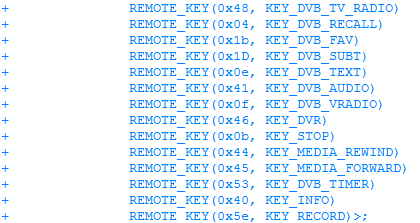
以上改动缺一不可。

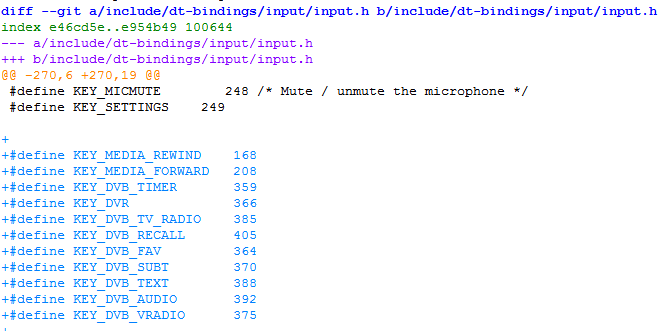
## 8. S905D DVB移植

### 8.1 Dvb相关遥控按键添加

将6.0 s905d dvb的patch移植到7.0后

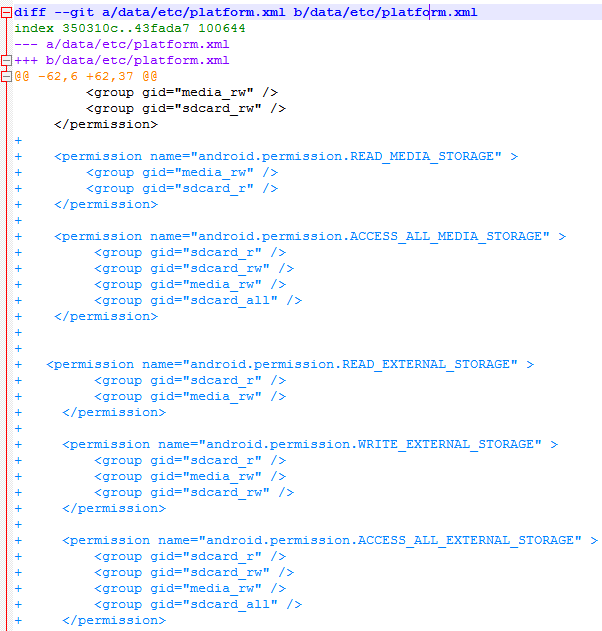
需在添加





### 8.2 应用权限

另外，之前data/etc/platform.xml的相关应用权限的这部份去掉，否则导致开机进不去launcher



### 8.3 dtv中UI层遮挡视频层问题

解决办法，在dtv.apk中在打开dtv播放和pvr播放的接口中添加设置透明度，如下：



### 8.4 时移和录制回放问题

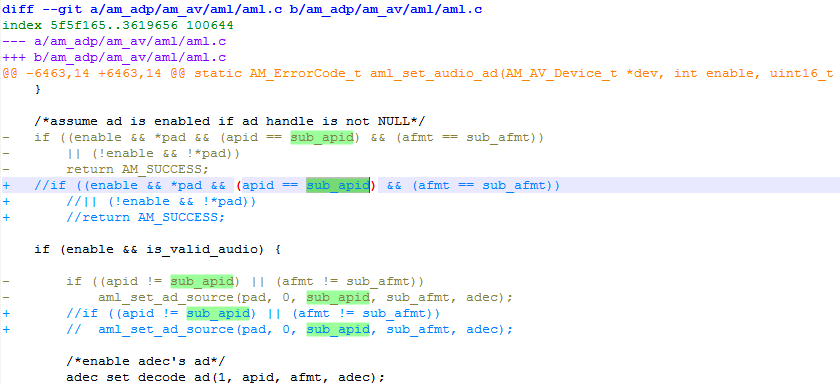
时移过程中或者录制回放过程中，按停止键，dtv崩溃。

解决办法：

修改libam\_adp.so源码

在时移启动接口中添加设置音频ID和fmt的接口，并屏蔽音频ID ，fmt比对出错的接口。

如下patch所示



详细见

amlogic\_n\_gxx\0044-adapt-dvb\011-vendor-amlogic-dvb-fix-am\_adp.patch

### 8.5 32位与64位应用问题

由于7.0 上跑的是64位系统 ，而我们Dtv跑的是32位，故需要求中间件组提供一个可以在64位系统上跑32位应用的dtv.apk

替换这个apk后，盒子上一些应用由于未限定是64位还是32位，重启后会由64位启动方式变为32位启动方式，从而开机启动的时候apk报错。

解决办法是，指定这些应用为64位启动方法：

修改的办法是，这里在/system/priv-app/TvSettings/中添加lib64文件夹或者在相应apk的Andriod.mk中添加LOCAL\_JNI\_SHARED\_LIBRARIES := liblog

刚在64位的sdk中编译时自己在apk目录下生成lib64文件夹（因apk启动时，会优先判断/system/priv-app/TvSettings/目录下时否有lib或者lib64文件夹来加载相应位数的库，如存在lib文件夹，则以32位的方式启动apk，如存在lib64文件夹，则64位的方式启动apk）。

其中需修改的apk有如下：

TvSettings.apk

KeyChain.apk

MboxLauncher2.apk

PPPoE.apk

SubTitle.apk

Dig.apk

ReadLog.apk