



Application Notes

MBOX 现场分析指南

Revision 0.1

AMLOGIC, Inc.

2518 Mission College Blvd

Santa Clara, CA 95054

U.S.A.

www.amlogic.com

AMLOGIC reserves the right to change any information described herein at any time without notice.
AMLOGIC assumes no responsibility or liability from use of such information.

目录

1. 简介.....	5
2. ANDROID 系统层现场分析.....	6
2.1. ANDROID 重启.....	6
2.2. ANDROID 无法启动.....	6
2.3. 系统无响应.....	7
2.4. 系统慢.....	7
2.5. 开机速度慢.....	8
2.6. ANR 错误.....	8
2.7. APK 自动退出.....	8
2.8. 常用调试方法简介.....	8
3. CVBS 显示异常现场分析.....	10
3.1. 关于异常的原因和表现.....	10
3.2. 显示模式比对.....	10
3.3. 查看 CVBS 是否开启.....	10
3.4. 比对相关寄存器的值.....	10
3.5. 使用 ENCI 测试信号.....	11
3.6. 考虑线材因素.....	11
4. HDMI 显示异常现场分析.....	12
4.1. HDMI HDCP 异常分析.....	12
4.2. 显示模式比对.....	12
4.3. 查看电视机的 EDID 信息.....	13
4.4. 比对相关寄存器的值.....	13
4.5. 使用 VIDEO ENCODER 测试信号.....	14
4.6. 其它现象的观察和分析.....	14
5. 界面显示异常现场分析.....	15
5.1. 显示部分的异常表现.....	15
5.2. 显示部分相关节点简介.....	15
5.3. 显示部分相关节点分析.....	15
5.4. 比对 OSD 寄存器.....	16
5.5. 显示部分相关问题判定.....	16
6. 多媒体相关问题现场分析.....	17
6.1. 视频黑屏异常.....	17
6.2. 视频花屏有马赛克.....	17
6.3. 确认 DI 造成的花屏，黑屏等问题.....	18
6.4. 视频抖动异常.....	18
6.5. 视频画面卡住或位置比例异常.....	18
6.6. 音频解码异常.....	19
6.7. 音视频卡顿.....	19
6.8. 音视频不同步.....	19
6.9. 软件 DEMUX 设置.....	19
6.10. 抓取播放视频流和网络流.....	20

Amlogic Application Notes

7. 系统 SUSPEND, REBOOT 死机手册	20
7.1. 系统 SUSPEND 问题	20
7.2. 系统 REBOOT 问题	21
7.3. 关于 WATCHDOG 开关问题	21
8. 内存信息的查看和分析	22
8.1. 检查内存泄漏	22
8.2. 查看内存信息	22
9. M6 项目烧录注意事项及出现问题时需提供相关的信息汇总	23
9.1. 工位配置	23
9.2. 工具崩溃问题	24
9.3. 烧录意外停止问题	24
9.4. 其它失败或异常问题	24
附录 A: M6 烧录平台端常见错误排查	25
附录 B: M6 烧录 PC 端常见错误排查	27

Amlogic Application Notes

修改记录

版本	日期	作者	修改
0.1	2014/07/01	张健钦	收录、初稿

1. 简介

本文档提供 MOBX 相关的问题分析方法和软件开发过程中的注意事项---

包括:

- **Android 系统层现场分析**
- **CVBS 显示异常现场分析**
- **HDMI 显示异常现场分析**
- **界面显示异常现场分析**
- **多媒体相关问题现场分析**
- **系统 suspend, reboot 死机手册**
- **内存信息的查看和分析**
- **M6 项目烧录注意事项及出现问题时需提供相关的信息汇总**

附录:

- **M6 烧录平台端常见错误排查**
- **M6 烧录 PC 端常见错误排查**

2. Android 系统层现场分析

<术语>

- ANR(Application Not Responding)
- Vold (volume 守护进程)
- Netd (网络守护进程)

对所有判断为 Android 层的问题，请都用 logcat 抓取日志，可以根据内容，做个基本判断：

● 2.1 Android 重启：

- logcat 日志抓取，记录触发重启的操作，记录是否可以重启完成，回到 Home 界面。
 - 如果一直停留在动画界面，不能重启完成，输入 mount 命令，检查一下 data 分区是否只读，否则重启系统，看能否重现。mount 命令示例如下（红框内 rw 为正常，若为 ro 则为只读）：
- ```
255|root@k200:/ # mount
.....
/dev/block/system /system ext4 ro,seclabel,noatime,nodiratime,noauto_da_alloc,data=ordered 0 0
/dev/block/data /data ext4 rw,seclabel,nosuid,nodev,noatime,nodiratime,noauto_da_alloc,data=ordered 0 0
/dev/block/cache /cache ext4 rw,seclabel,nosuid,nodev,noatime,nodiratime,noauto_da_alloc,data=ordered 0 0
/dev/fuse /mnt/shell/emulated fuse rw,nosuid,nodev,relatime,user_id=1023,group_id=1023,default_permissions
```
- 如果重启可以完成，尝试重现一下，如果可以重现，记录概率情况。
  - 在多个日志中，搜索一下"backtrace:"，找到 crash 的堆栈，简单比较一下，如果每个日志的堆栈都一样，保存一个日志就可以了，如果不一样，多抓几个日志，一起保存起来。Crash 日志示例如下：

```
"main" prio=5 tid=1 NATIVE
 group="main" sCount=1 dsCount=0 obj=0x4154bca8 self=0x41484430
 sysTid=465 nice=-2 sched=0/0 cgrp=[fopen-error:2] handle=1073938772
 state=D schedstat=(18243170000 18029369000 20616) utm=706 stm=1118 core=0
 (native [backtrace] unavailable)
 at android.net.NetworkUtils.resetConnections(Native Method)
 at android.net.pppoe.PppoeStateTracker.stopInterface(PppoeStateTracker.java:117)
 at com.android.server.PppoeService$1.onReceive(PppoeService.java:184)
```

### ● 2.2 Android 无法启动：

- 使用 ps 命令检查一下当前都有哪些进程已经启动，如果 vold, netd, surfaceFlinger 等 core 进程都没有启动。
- 如果 vold, netd, surfaceflinger 等经常可以启动，记录启动日志和 logcat 日志。

```
root@k200:/ # ps
USER PID PPID VSIZE RSS WCHAN PC NAME
root 1 0 1136 588 c00cae60 0004bae8 S /init
root 2 0 0 0 c0044f7c 00000000 S kthreadd
root 115 1 556 272 c0038ad0 0002ff58 S /system/bin/sh
system 116 1 1016 340 c0384390 b6f697a4 S /system/bin/servicemanager
root 117 1 4724 936 ffffffff b6eb509c S /system/bin/vold
root 120 1 9728 1100 ffffffff b6f0a09c S /system/bin/netd
system 122 1 107052 9108 ffffffff b6ee5954 S /system/bin/surfaceflinger
root 123 1 627704 18968 ffffffff 401098f0 S zygote
```

- 输入 mount 命令，检查一下 system 分区的是否挂载。

```
tmpfs /mnt/asec tmpfs rw,seclabel,relatime,mode=755,gid=1000 0 0
tmpfs /mnt/obb tmpfs rw,seclabel,relatime,mode=755,gid=1000 0 0
tmpfs /storage/external_storage tmpfs rw,seclabel,relatime,mode=775,uid=1000,gid=1023 0 0
adb /dev/usb-ffs/adb_functionfs rw,relatime 0 0
/dev/block/system /system ext4 ro,seclabel,noatime,nodiratime,noauto_da_alloc,data=ordered 0 0
/dev/block/data /data ext4 rw,seclabel,nosuid,nodev,noatime,nodiratime,noauto_da_alloc,data=ordered 0 0
/dev/block/cache /cache ext4 rw,seclabel,nosuid,nodev,noatime,nodiratime,noauto_da_alloc,data=ordered 0 0
/dev/fuse /mnt/shell/emulated fuse rw,nosuid,nodev,relatime,user_id=1023,group_id=1023,default_permissions,allow_other 0 0
```

- 如果 system 分区未挂载，输入 ls /dev/block/system 查看 system 是否存在。
- 如果存在，手工 mount 试试（mount -t ext4 /dev/block/system /system），并记录这个过程的日志。

```
1|root@k200:/ # [ls /dev/block/system]
/dev/block/system
root@k200:/ # [mount -t ext4 /dev/block/system /system]
[5673.547614@0] blktrans_ioctl -cmd: 125d, 801c0204, 64
mount: Device or resource busy
```

- mount 检查 data 分区是否正常挂载为 rw。

# Amlogic Application Notes

```
255|root@k200:/ #
255|root@k200:/ # mount -t ext4 /dev/block/data /data/
[5993.813389@0] blktrans_ioct1 -cmd: 125d,801c0204,64
mount: Device or resource busy
255|root@k200:/ # mount
.....
/dev/block/system /system ext4 ro,seclabel,noatime,nodiratime,noauto_da_alloc,data=ordered 0 0
/dev/block/data /data ext4 rw,seclabel,nosuid,nodev,noatime,nodiratime,noauto_da_alloc,data=ordered 0
/dev/block/cache /cache ext4 rw,seclabel,nosuid,nodev,noatime,nodiratime,noauto_da_alloc,data=ordered
/dev/fuse /mnt/shell/emulated fuse rw,nosuid,nodev,relatime,user_id=1023,group_id=1023,default_permiss
```

- 多次重启，看能够正常启动，如果每次都不能启动，进入 recovery 里边，wipe\_data 后再试一次。
- 如果 wipe\_data 之后还不能正常启动，cp -r /system/ /storage/external\_stroage/sdcard1/system，比较一下

拷贝出来的 system 分区文件和升级包中是否一致。

- 如果有时可以启动，有时不行，抓取多个 logcat。

## ● 2.3 系统无响应：

- 抓取 logcat。
- 输入 ps 命令，检查进程状态。
- 扭遥控器上的按键，看有没有类似以下的 logcat 输出：

```
[10740.719437@0] key 116 down
[10740.860024@0] key 116 up
```

## ● 2.4 系统慢：

- 抓取 logcat。
- 抓取现在系统占用，top -t -m 20。
- 抓取内存使用情况，cat /proc/meminfo，或者 dumsys meminfo。

```
130|root@k200:/ # top -t -m 20
User 1%, System 1%, IOW 0%, IRQ 0%
User 3 + Nice 0 + Sys 4 + Idle 224 + IOW 0 + IRQ 0 + SIRQ 0 = 231
PID TID PR CPU% S VSS RSS PCY UID Thread Proc
1694 1694 0 2% S 0K 0K fg root kworker/0:0
1904 1904 0 1% R 1400K 488K fg root top top
61 61 0 0% S 0K 0K fg root kthread_di
865 1265 0 0% S 643628K 18144K fg system SubTitleService com.amlogic.SubTitleService
464 523 0 0% S 719596K 42356K fg system Binder_3 system_server
1042 1042 0 0% S 936K 376K fg system usbttestpm /system/bin/usbttestpm
464 474 0 0% S 719596K 42356K fg system Binder_l system_server
.....
130|root@k200:/ # cat /proc/meminfo
MemTotal: 846544 kB
MemFree: 272708 kB
.....
root@k200:/ # dumsys meminfo
Total PSS by process:
34367 kB: com.android.systemui (pid 675)
31753 kB: com.mbx.settingsmbox (pid 981)
.....
```

## ● 2.5 开机速度慢：

- 抓取开机日志。
- logcat -v time。
- logcat -b events -v time。

```
1|root@k200:/ # logcat -v time
----- beginning of /dev/log/main
01-02 13:41:12.570 V/AudioFlinger(125): Audio hardware entering standby, mixer 0xb3617008, suspend count 0,mid 7
01-02 13:41:12.570 D/audio_hw_hdmi(125): out_standby(0xb8dba2d8)
.....
130|root@k200:/ # logcat -b events -v time
01-02 11:52:38.750 I/boot_progress_start(123): 14550
01-02 11:52:41.060 I/boot_progress_preload_start(123): 16865
.....
```



# Amlogic Application Notes

## ● 2.6 ANR 错误:

- 抓取 logcat, 并在 ANR 后, 马上抓取 /data/anr/traces.txt.
- 查看 LOG 中 CPU 占用率, 当 iowait 很高时说明当前系统在忙于 I/O:

```
.....
E/ActivityManager(465): 0% 2062/com.tcl.settings: 0% user + 0% kernel / faults: 12 minor
E/ActivityManager(465): 92% TOTAL: 8.4% user + 9% kernel + 73% iowait + 2.1% softirq
E/ActivityManager(465): CPU usage from 1411ms to 1921ms later:
E/ActivityManager(465): 14% 465/system_server: 0% user + 14% kernel
.....
```

- 通过 traces 分析 ANR 是否为死锁造成的, 示例 LOG 如下:

```
"PowerManagerService" prio=5 tid=24 MONITOR
.....
 at com.android.server.am.ActivityManagerService.broadcastIntent(ActivityManagerService.java:~13045)
 - waiting to lock <0x41a874a0> (a com.android.server.am.ActivityManagerService) held by tid=12 (android.server.ServerThread)
 at android.app.ContextImpl.sendBroadcast(ContextImpl.java:1144)
"Binder_B" prio=5 tid=85 MONITOR
.....
 at com.android.server.power.PowerManagerService$DisplayBlankerImpl.toString(PowerManagerService.java:~3449)
 - waiting to lock <0x41afe2e8> (a com.android.server.power.PowerManagerService$DisplayBlankerImpl) held by tid=24 (PowerManagerService)
 at java.lang.StringBuilder.append(StringBuilder.java:202)
 at android.server.ServerThread.prio=5 tid=12 MONITOR
.....
 at com.android.server.power.PowerManagerService.isScreenOnInternal(PowerManagerService.java:~2529)
 - waiting to lock <0x41afe2e8> (a java.lang.Object) held by tid=85 (Binder_B)
 at com.android.server.power.PowerManagerService.isScreenOn(PowerManagerService.java:2522)
```

从 trace 文件看, 是因为 TID 为 24 的线程等待一个 TID 为 12 的线程持有的锁, TID 为 12 的线程等待一个 TID 为 85 的线程持有的锁, 而 TID 为 85 的线程确等待一个 TID 为 24 的线程持有的锁。

## ● 2.7 APK 自动退出:

- 抓取 logcat 日志, 并记录一些关键信息。示例如下:

```
E/AndroidRuntime(1728): FATAL EXCEPTION: main
E/AndroidRuntime(1728): Process: com.android.settings, PID: 1728
E/AndroidRuntime(1728): java.lang.RuntimeException: Unable to start activity ComponentInfo{com.android.settings/com.android
java.lang.IllegalArgumentException: Invalid fragment for this activity: com.android.settings.ethernet.EthernetSettingsAML
E/AndroidRuntime(1728): at android.app.ActivityThread.performLaunchActivity(ActivityThread.java:2195)
E/AndroidRuntime(1728): at android.app.ActivityThread.handleLaunchActivity(ActivityThread.java:2245)
.....
E/AndroidRuntime(1728): Caused by: java.lang.IllegalArgumentException: Invalid fragment for this activity: com.android.sett
```

## ● 2.8 常用调试方法简介:

- Tombstones 简介

Tombstones 是一些进程 core dump (tombstone 是墓碑的意思), 里面放的是异常程序挂掉时的一些状态, 供开发人员找问题使用, /data/tombstones/目录下一般存在 tombstone\_XX 或 tombstoneNoCrash\_XX 的文件。tombstoneNoCrash\_XX 一般是 apk 错误, tombstone\_XX 是比较严重的系统错误, 里面有比较详细的信息。如果平台重启, 这个目录会被清掉。

- Dropbox 简介

Dropbox 在 /data/system/dropbox 目录下, 里面记录有 app crash、anr 这些 apk 错误和 system server crash、watchdog 这些是系统错误。信息示例:

```
root@k200:/data/system/dropbox # cat SYSTEM_BOOT@86407621.txt
Build: MBX/k200/k200:4.4.2/KOT49H/20140627:user/test-keys
Hardware: k200
Revision: 11
Bootloader: unknown
Radio: unknown
Kernel: Linux version 3.10.33 (mbx.autobuild@droid12-sz) (gcc version 4.7.3 201
```

- Dumpsys 命令简介

使用 `dumpsys list` 命令可以查看支持命令列表, 较常用的: meminfo, cpuinfo, power, display, SurfaceFlinger, activity, package 等, dumpsys 可以打印出相应模块的系统信息。以下是 `dumpsys cpuinfo` 的输出:

```
root@k200:/data/system/dropbox # dumpsys cpuinfo
Load: 0.04 / 0.13 / 0.13
CPU usage from 44967ms to 39484ms ago:
...
0.1% 752/com.amlogic.SubTitleService: 0.1% user + 0% kernel
10% TOTAL: 0.7% user + 0.4% kernel + 9.6% iowait
```



# Amlogic Application Notes

## ■ am/pm 命令简介

使用 **am/pm** 命令，可直接输入 **am** 或 **pm** 命令可以得到详细的帮助信息出来。以下是一小段 **am** 帮助信息示例：

```
root@k200:/data/system/dropbox # am
usage: am [subcommand] [options]
...
am display-size [reset|WxH]
am display-density [reset|DENSITY]
am to-uri [INTENT]
...
am start: start an Activity. Options are:
-D: enable debugging
-W: wait for launch to complete
```

**am/pm** 命令功能强大，以下我们以 **am broadcast** 和 **pm list packages** 为例，展示一下 **am/pm** 命令的使用

1) **adb shell am broadcast -a android.intent.action.MASTER\_CLEAR**，该命令会发出一个恢复出厂设置的广播，直接启动 **android** 的出厂设置功能，示例如下：

```
root@k200:/ # am broadcast -a android.intent.action.MASTER_CLEAR
Broadcasting: Intent { act=android.intent.action.MASTER_CLEAR }
Broadcast completed: result=0
BT : 10:00:58 Jun 25 2014
DCDC01 set to 1100, register from 0x001e to 0x0028, addr:0x0036
```

2) **pm list packages com.android.setting**，该命令可以为你查找到系统中包名为 **com.android.setting** 的应用程序所在的绝对路径，示例如下：

```
root@k200:/ # pm list packages com.android.setting
package:/system/priv-app/Settings.apk=com.android.settings
root@k200:/ #
```

## ■ Core dump 简介

**core dump** 是软件错误无法恢复的产物（**core** 为内存的意思）。进程 **coredump** 的时候，操作系统会将进程终止并释放其占用的资源，正常情况下，应用进程 **coredump** 不会对系统本身的运行造成危害。当然如果系统中存在与此进程相关的其他进程，则这些进程会受到影响，至于后果则视其对此异常的具体处理而定。

在 **Android** 平台上，能够正确生成 **core dump** 的方法是修改 **init.rc** 文件。在 **init.rc** 中 **setrlimit 13 40 40** 下增加一条记录: **setrlimit 4 -1 -1**。具体修改方式如下所示（以下文件在 **system/core** 下）：

```
diff --git rootdir/init.rc rootdir/init.rc
index ad77923..f0759a7 100755
--- rootdir/init.rc
+++ rootdir/init.rc
@@ -267,6 +267,8 @@ on boot

 # set RLIMIT_NICE to allow priorities from 19 to -20
 setrlimit 13 40 40
+ setrlimit 4 -1 -1
+ write /proc/sys/kernel/core_pattern "/data/core-%e.%p.%u"
```

增加修改后，当进程 **coredump** 的时候，会在 **/data** 下面生成 **core-XXXXXX** 的文件，其中

%e: 出 **Core** 进程对应的可执行文件名

%p: 出 **Core** 进程的 PID

%u: 出 **Core** 进程的 UID

发生 **core dump** 的主要原因有以下几个方面：

- 1) 内存访问越界
- 2) 多线程程序使用了线程不安全的函数
- 3) 多线程读写的数据未加锁保护
- 4) 非法指针
- 5) 堆栈溢出

## 3. CVBS 显示异常现场分析

- 3.1 关于异常的原因和表现:

就当前使用中的视频接口来看, cvbs 是模拟信号中最差的一种格式, 既是模拟信号又是亮色复合, 这两方面本身就会造成显示效果不理想。所以关于显示模糊或抖动现象, 如果用数字的 hdmi 信号、模拟的分量或 s 端子信号, 比 cvbs 表现好, 都是正常的。

关于模糊和抖动下列表现, 是异常显示:

- a. 相同 sourcecode 的不同单板, 在同一电视上表现明显有差异;
- b. 相同 sourcecode 不同版本软件, 在同一电视上表现有差异;
- c. 相比其他解决方案, 在同一电视上表现明显有差异;
- d. 特定操作后, 导致输出明显变得更加模糊或抖动;
- e. 单独出现在 osd 或者 video 的模糊或抖动现象;

- 3.2 显示模式比对:

使用命令 `cat /sys/class/display/mode` 查看当前的显示模式, 如果非预期模式, 则需要向上层追查。

- 3.3 查看 CVBS 是否开启:

使用命令 `echo r v 0x1b30 > /sys/class/amlogic/debug` 查看寄存器[0x1b30]的值, 输出:

VCBUS[0x1b30]=0x00000000: cvbs 已开启

VCBUS[0x1b30]=0x00000001: cvbs 已关闭

- 3.4 比对相关寄存器的值:

在同一软件相同输出模式下, 记录正常表现时和非正常表现时以下各寄存器的值, 进行对比。

- CLOCK

```
echo dump c 0x10c8 0x10cc > /sys/class/amlogic/debug
echo r c 0x1066 > /sys/class/amlogic/debug
echo r c 0x105f > /sys/class/amlogic/debug
echo r c 0x1059 > /sys/class/amlogic/debug
echo r c 0x104a > /sys/class/amlogic/debug
echo clkmsr > /sys/class/amlogic/debug
```

- VDAC (对于 m3/m6, 不需要 dump 这两个寄存器; 对于 m8 系列, 需要 dump 这两个寄存器)

```
echo dump c 0x10bd 0x10be > /sys/class/amlogic/debug
```

- VENC

```
echo dump v 0x1b60 0x1b66 > /sys/class/amlogic/debug
echo dump v 0x1b68 0x1b7f > /sys/class/amlogic/debug
echo dump v 0x1bf0 0x1bfc > /sys/class/amlogic/debug
echo r v 0x1c51 > /sys/class/amlogic/debug
echo dump v 0x1c54 0x1c5f > /sys/class/amlogic/debug
```

- ENCI (对于 M8 之前的 M3/M6, 需要将上面的 v 改为 c)

```
echo dump v 0x1b00 0x1b13 > /sys/class/amlogic/debug
echo dump v 0x1b16 0x1b1b > /sys/class/amlogic/debug
echo dump v 0x1b20 0x1b46 > /sys/class/amlogic/debug
echo dump v 0x1b48 0x1b4d > /sys/class/amlogic/debug
echo dump v 0x1b50 0x1b57 > /sys/class/amlogic/debug
echo dump v 0x1c00 0x1c0c > /sys/class/amlogic/debug
echo dump v 0x1c10 0x1c1f > /sys/class/amlogic/debug
echo dump v 0x1c40 0x1c43 > /sys/class/amlogic/debug
```

# Amlogic Application Notes

---

```
echo r v 0x271c > /sys/class/amlogic/debug
```

## ■ GATE

```
echo dump c 1050 1054 > /sys/class/amlogic/debug
```

```
echo dump c 1040 1044 > /sys/class/amlogic/debug
```

## ● 3.5 使用 ENCI 测试信号:

### ■ 彩条（色彩分辨为白、黄、青、绿、紫、红、蓝）

```
echo w 2 v 0x1b01 > /sys/class/amlogic/debug
```

```
echo w 1 v 0x1c11 > /sys/class/amlogic/debug
```

```
echo w 112 v 0x1c15 > /sys/class/amlogic/debug
```

```
echo w b4 v 0x1c16 > /sys/class/amlogic/debug
```

```
echo w 1 v 0x1c10 > /sys/class/amlogic/debug
```

### ■ 点阵

```
echo w 2 v 0x1b01 > /sys/class/amlogic/debug
```

```
echo w 3 v 0x1c11 > /sys/class/amlogic/debug
```

```
echo w 112 v 0x1c15 > /sys/class/amlogic/debug
```

```
echo w b4 v 0x1c16 > /sys/class/amlogic/debug
```

```
echo w 1 v 0x1c10 > /sys/class/amlogic/debug
```

### ■ 方格

```
echo w 2 v 0x1b01 > /sys/class/amlogic/debug
```

```
echo w 2 v 0x1c11 > /sys/class/amlogic/debug
```

```
echo w 112 v 0x1c15 > /sys/class/amlogic/debug
```

```
echo w b4 v 0x1c16 > /sys/class/amlogic/debug
```

```
echo w 1 v 0x1c10 > /sys/class/amlogic/debug
```

### ■ 恢复

```
echo w 0x26 v 0x1b01 > /sys/class/amlogic/debug
```

```
echo w 0 v 0x1c10 > /sys/class/amlogic/debug
```

## ● 3.6 考虑线材因素

普通的 cvbs 线，尤其是耳机插孔那种一转三的线材，大部分为不符合要求的线材；可以使用好的线材进行对比。

## 4. HDMI 显示异常现场分析

### ● 4.1 HDMI HDCP 异常分析:

#### ■ HDCP 简介

HDCP 的全称是 High-bandwidth Digital Content Protection，也就是“高带宽数字内容保护”。简单的说，HDCP 就是要将通过 DVI 接口传递的数字信号进行加密，多媒体内容的发出端（电脑、DVD、机顶盒等）与接受端（显示器、电视机、投影机等）之间加上一道保护。这样一层保护主要并不是用来防止通过数字信号进行不合法的复制，而是将数字信号内容进行加密，使得不合法的复制无法得到准确的内容、满意的效果。

#### ■ 检查 HDCP 交互

输入命令 `cat /sys/module/hdmitx/parameters/hdmi_authenticated`，输出大于 0 为成功，小于等于 0 为失败

#### ■ 检查 ksv(视频加密密钥)

输入命令 `cat /sys/class/amhdmitx/amhdmitx0/hdcp_ksv_info`，输出如下：

AKSV: 663fa14267 Valid

BKSV: 27c2d1b6ca Valid

如果出现 AKSV 为 invalid 的情况，请检查 HDCP key 烧录是否有误。

如果出现 BKSV 为 invalid 的情况，请检查接收端是否为合法的带有 HDCP key 的设备。

#### ■ HDCP 的内部检测

输入命令 `cat /sys/class/amhdmitx/amhdmitx0/hdcp_ksv_info`，出现以下打印表示检测成功，否则为检测失败

[ 8279.851320@0] HDMI: Monitor HDCP start

[ 8279.853715@0] Aksv\_shw: 663fa14267

... ..

[ 8279.854491@0] HDCP OK

[8279.940312@0] HDMI: Monitor HDCP end

### ● 4.2 显示模式比对:

#### ■ 查看是否已连接电视机

输入命令 `cat /sys/class/amhdmitx/amhdmitx0/hpd_state`

0: 未连接电视

1: 已连接电视

如果输出为 0 且 hdmi 线已连接电视，那么需要测量 hdmi 座子及芯片引脚上相应电压

#### ■ 查看当前显示模式，是否是预期的显示模式，如果非预期模式，则需要向上层追查

输入命令 `cat /sys/class/display/mode`，输出当前分辨率（如：720p50hz）

可尝试其它分辨率效果，（如：输入 `echo 1080p > /sys/class/display/mode`）

输入命令 `cat /sys/class/amhdmitx/amhdmitx0/disp_mode`，也可查看分辨率，显示的是 vic 编号，其对应关系：

VIC: 0 --> HDMI\_Unkown

VIC: 2 --> 480p

VIC: 4 --> 720p

VIC: 5 --> 1080i

VIC: 6 --> 480i

VIC: 16 --> 1080p

VIC: 17 --> 576p

VIC: 19 --> 720p50

VIC: 20 --> 1080i50

VIC: 21 --> 576i

VIC: 31 --> 1080p50

VIC: 93 --> 4k2k24hz

VIC: 94 --> 4k2k25hz

VIC: 95 --> 4k2k30hz

VIC: 98 --> smpte, 24hz

# Amlogic Application Notes

## ● 4.3 查看电视机的 edid 信息:

### ■ 查看电视机的 edid 信息

命令 1: `cat /sys/class/amhdmix/amhdmix0/edid`

命令 2: `echo d0 > /sys/class/amhdmix/amhdmix0/edid`

命令 3: `echo d1 > /sys/class/amhdmix/amhdmix0/edid`

### ■ 查看电视机支持的最佳分辨率

命令: `cat /sys/class/amhdmix/amhdmix0/disp_cap`, 以下为输出 (其中带星号的为最佳分辨率):

480i

576i

576p

720p

1080i

1080p

720p50hz\*

1080i50hz

1080p50hz

## ● 4.4 比对相关寄存器的值:

### ■ 查看电视机支持的最佳分辨率

首先关闭 kernel 打印信息, 使用命令 `echo 0 > /sys/module/printk/parameters/time`

在同一软件相同输出模式下, 记录正常表现时和非正常表现时以下各寄存器的值, 进行对比。

### ■ CLOCK

`echo dump c 0x10c8 0x10cc > /sys/class/amlogic/debug`

`echo r c 0x1066 > /sys/class/amlogic/debug`

`echo r c 0x105f > /sys/class/amlogic/debug`

`echo r c 0x1059 > /sys/class/amlogic/debug`

`echo r c 0x104a > /sys/class/amlogic/debug`

`echo r c 0x1073 > /sys/class/amlogic/debug`

`echo clkmsr > /sys/class/amlogic/debug`

### ■ VENC

`echo dump v 0x1b60 0x1b66 > /sys/class/amlogic/debug`

`echo dump v 0x1b68 0x1b7f > /sys/class/amlogic/debug`

`echo dump v 0x1bf0 0x1bfc > /sys/class/amlogic/debug`

`echo r v 0x1c51 > /sys/class/amlogic/debug`

`echo dump v 0x1c54 0x1c5f > /sys/class/amlogic/debug`

### ■ ENCP

`echo dump v 0x1b58 0x1b5e > /sys/class/amlogic/debug`

`echo dump v 0x1b80 0x1b8e > /sys/class/amlogic/debug`

`echo dump v 0x1b90 0x1ba4 > /sys/class/amlogic/debug`

`echo dump v 0x1ba6 0x1bbd > /sys/class/amlogic/debug`

`echo dump v 0x1bc0 0x1bce > /sys/class/amlogic/debug`

`echo dump v 0x1bd0 0x1bd6 > /sys/class/amlogic/debug`

`echo dump v 0x1c0d 0x1c0f > /sys/class/amlogic/debug`

`echo dump v 0x1c30 0x1c3f > /sys/class/amlogic/debug`

`echo dump v 0x1c44 0x1c47 > /sys/class/amlogic/debug`

`echo r v 0x1c50 > /sys/class/amlogic/debug`

`echo r v 0x271d > /sys/class/amlogic/debug`

### ■ ENCI(对于 M8 之前的 M3/M6, 需要将上面的 v 改为 c)

# Amlogic Application Notes

```
echo dump v 0x1b00 0x1b13 > /sys/class/amlogic/debug
echo dump v 0x1b16 0x1b1b > /sys/class/amlogic/debug
echo dump v 0x1b20 0x1b46 > /sys/class/amlogic/debug
echo dump v 0x1b48 0x1b4d > /sys/class/amlogic/debug
echo dump v 0x1b50 0x1b57 > /sys/class/amlogic/debug
echo dump v 0x1c00 0x1c0c > /sys/class/amlogic/debug
echo dump v 0x1c10 0x1c1f > /sys/class/amlogic/debug
echo dump v 0x1c40 0x1c43 > /sys/class/amlogic/debug
echo r v 0x271c > /sys/class/amlogic/debug
```

## ■ GATE

```
echo dump c 1050 1054 > /sys/class/amlogic/debug
echo dump c 1040 1044 > /sys/class/amlogic/debug
```

## ■ HDMI

```
echo dump hdmireg > /sys/class/amhdmity/amhdmity0/debug
```

### ● 4.5 使用 video encoder 测试信号:

■ 不同分辨率模式下, 彩条宽度和位置有所不同, 通过其他板子在其他电视上测试信号, 来对比问题板子, 通过色彩、图像大小、抖动等, 判断是否 video encoder 及 hdmi 后端出问题, 还是 osd+video 前端有问题。

#### ■ 彩条 (色彩分辨为白、黄、青、绿、紫、红、蓝)

```
echo w 0 v 0x1b8e > /sys/class/amlogic/debug
echo w 1 v 0x1b71 > /sys/class/amlogic/debug
echo w 1 v 0x1b70 > /sys/class/amlogic/debug
```

#### ■ 方格

```
echo w 0 v 0x1b8e > /sys/class/amlogic/debug
echo w 2 v 0x1b71 > /sys/class/amlogic/debug
echo w 1 v 0x1b70 > /sys/class/amlogic/debug
```

#### ■ 点阵

```
echo w 0 v 0x1b8e > /sys/class/amlogic/debug
echo w 3 v 0x1b71 > /sys/class/amlogic/debug
echo w 1 v 0x1b70 > /sys/class/amlogic/debug
```

#### ■ 另外对 hdmi, 可用基本色场查看纯色画面

红屏: `echo w 0xaff00000 v 2727 > /sys/class/amlogic/debug`

绿屏: `echo w 0x800ffc00 v 2727 > /sys/class/amlogic/debug`

蓝屏: `echo w 0x800003ff v 2727 > /sys/class/amlogic/debug`

关闭: `echo w 0 v 0x2727 > /sys/class/amlogic/debug`

### ● 4.6 其它现象的观察和分析:

■ 屏蔽 hpd(`echo hpd_lock1 > /sys/class/amhdmity/amhdmity0/debu`), 拔插 hdmi 线、尝试同一电视不同 hdmi 端口或电视机, 观察输出是否有变化, 打开 hpd 响应: `echo hpd_lock0 > /sys/class/amhdmity/amhdmity0/debug`

■ 观察电视机是黑屏, 还是提示无信号? 电视机是否有提示分辨率信息? 拔插 hdmi 线, 观察现象表现。

对比同一软件不同硬件单板表现、对比公版软硬件表现、对比不同电视机表现、对比是否与某一分辨率有关; 如果可能与电视机有关, 可使用 hdmi splitter 信号分支器, 同时接多台电视, 进行对比, 是否某一电视上表现异常时, 另外一个电视表现正常。

## 5. 界面显示异常现场分析

### ● 5.1 显示部分的异常表现:

- 在 home 界面或者其他 APK 界面出现花屏, 斜屏, 黑屏, 显示区域缩小或者超出屏幕的情况。
- 在视频播放界面中, 控制栏出现花屏, 斜屏, 或者无法调出控制栏的情况。

### ● 5.2 显示部分相关节点简介 (以下以 K200 为例):

- K200 目前用的是一套真实的 1080P UI, 即使用的是 1920x1080 大小的 framebuffer, 因此它的原始输出大小 (free\_scale\_axis) 是固定的, android 层的 display-size 和 density 也是固定的。

#### ■ 节点简介

1) /sys/class/display/mode: 该节点用于查看或设置当前的输出模式, 其值随设置而改变。

查看: `cat /sys/class/display/mode`(输出示例: 720p50hz)

设置: `echo 720p > /sys/class/display/mode` (设置后应使用查看命令确认设置是否生效)

2) /sys/class/graphics/fb0/free\_scale\_axis: 查看和设置 OSD 及 video 的源输出尺寸, 每次开机会设回默认值。

查看: `cat /sys/class/graphics/fb0/free_scale_axis`(输出示例: 0 0 1919 1079)

3) /sys/class/graphics/fb0/freescale\_mode: 设置缩放模式, 每次开机会设回默认值 (默认 M6 为 0, M8 为 1)

查看: `cat /sys/class/graphics/fb0/freescale_mode`(输出示例: free\_scale\_mode:new 【new 表示 1, 即 M8 缩放模式】)

4) /sys/class/graphics/fb0/free\_scale: 设置是否开启缩放, (1080P UI 只有在 1080p 制式下不需要缩放)

查看: `cat /sys/class/graphics/fb0/free_scale`(输出示例: free\_scale\_enable:[0x1] 【缩放被开启】)

设置: `echo 0 > /sys/class/graphics/fb0/free_scale` (关闭缩放功能)

5) /sys/class/graphics/fb0/window\_axis: 设置 osd 重显率, 即系统显示界面的位置及大小。

查看: `cat /sys/class/graphics/fb0/window_axis`(输出示例: fwindow axis is [0 0 1919 1078])

注: M6 上该节点应为: /sys/class/ppmgr/ppsclaer\_rect

6) /sys/class/video/axis: 设置 video 的重显率, 即视频界面的位置及大小

查看: `cat /sys/class/video/axis`(输出示例: 0 0 1919 1079)

7) display-size 和 density (android 系统显示尺寸和像素密度), 要和 free\_scale\_axis 对应起来。

1080P native UI 下: display-size=1920x1080,density=240;

720P native UI 下: display-size=1280x720,density=160;

查看: `dumpsys display`(输出示例: mBaseDisplayInfo=DisplayInfo{"Built-in Screen", app 1920 x 1080, real 1920 x 1080, largest app 1920 x 1080, smallest app 1920 x 1080, 60.000004 fps, rotation0, density 240 (159.89508 x 160.42105) dpi, layerStack 0, type BUILT\_IN, FLAG\_SECURE, FLAG\_SUPPORTS\_PROTECTED\_BUFFERS})

### ● 5.3 显示部分相关节点分析:

#### ■ M8 或 M8 baby 使用 1080P native UI 的情况

1) /sys/class/graphics/fb0/free\_scale\_axis 的值必须为 "0 0 1919 1079", 否则为异常值。

2) /sys/class/graphics/fb0/freescale\_mode 的值必须为 "free\_scale\_mode:new", 否则为异常值。

3) /sys/class/graphics/fb0/free\_scale 在非 1080p 制式下必须为 0x1; 1080p 制式下为 0x0 或 0x1。

4) display-size 必须为 1920x1080, density 必须为 240, 否则为异常值。

5) /sys/class/graphics/fb0/window\_axis 和 /sys/class/video/axis 这两个节点的值必须要小于或等于当前分辨率, 但是也不能小到超过 20%。譬如 HDMI 为 1080P 模式, 则 HDMI 的宽和高分别为 1920,1080, 因此查看这两个节点的值 W\*H 应为:1536x864<W\*H<1920x1080, 其他 HDMI 模式, 依次类推。

#### ■ M8 或 M8 baby 使用 720P native UI 的情况

1) /sys/class/graphics/fb0/free\_scale\_axis 的值必须为 "0 0 1279 719", 否则为异常值。

2) /sys/class/graphics/fb0/freescale\_mode 的值必须为 "free\_scale\_mode:new", 否则为异常值。

3) /sys/class/graphics/fb0/free\_scale 在非 720p 制式下必须为 0x1; 720p 制式下为 0x0 或 0x1。

4) display-size 必须为 1280x720, density 必须为 160, 否则为异常值。



## Amlogic Application Notes

5) `/sys/class/graphics/fb0/window_axis` 和 `/sys/class/video/axis` 这两个节点的值必须要小于或等于当前分辨率，但是也不能小到超过 20%。譬如 HDMI 为 720P 模式，则 HDMI 的宽和高分别为 1280,720，因此查看这两个节点的值 W\*H 应为:1024x576<W\*H<1280x720，其他 HDMI 模式，依次类推。

### ■ M6 使用的是 720P native UI

1) `/sys/class/graphics/fb0/free_scale_axis` 的值必须为 “0 0 1279 719”，否则为异常值。

2) `/sys/class/graphics/fb0/freescale_mode` 的值必须为 “free\_scale\_mode:normal”，否则为异常值。

3) `/sys/class/graphics/fb0/free_scale` 在非 720p 制式下必须为 0x1；720p 制式下为 0x0 或 0x1(如果打开了点对点播放视频时，则 free\_scale 的值必须为 0)。

4) `display-size` 必须为 1280x720，`density` 必须为 160，否则为异常值。

5) `/sys/class/ppmgr/ppscaler_rect` 和 `/sys/class/video/axis` 这两个节点的值必须要小于或等于当前分辨率，但是也不能小到超过 20%。譬如 HDMI 为 720P 模式，则 HDMI 的宽和高分别为 1280,720，因此查看这两个节点的值 W\*H 应为:1024x576<W\*H<1280x720，其他 HDMI 模式，依次类推。

### ● 5.4 比对 osd 寄存器：

获取 OSD 寄存器信息（比对正常情况下和异常情况下的输出差异）

#### ■ 输入命令：echo reg > /sys/class/graphics/fb0/osd\_info\_msg，输出：

```
[4320.178367@1] [0x1a1b]: *0x408588*
[4320.178367@1] [0x1a1c]: *0x77f0000*
[4320.178367@1] [0x1a1d]: *0x4370000*
[4320.178367@1] [0x1a1e]: *0x4ff0000*
[4320.178367@1] [0x1a13]: *0x2cf0000*
[4320.191132@1] [0x1a3b]: *0x438490*
[4320.191132@1] [0x1a3c]: *0x4ff0000*
[4320.191132@1] [0x1a3d]: *0x2cf0000*
[4320.191132@1] [0x1a3e]: *0x4ff0000*
[4320.191132@1] [0x1a64]: *0x2cf0000*
```

#### ■ 输入命令：echo r c 1d26 > /sys/class/amlogic/debug，输出：

CBUS[0x1d26]=0x00000000

### ● 5.5 显示部分相关问题判定：

#### ■ 确认 framebuffer 的数据是否正常

使用 fb2png 工具，将该工具拷贝至 /system/bin 下，并修改其权限（`chmod 777 /system/bin/fb2png`）

输入命令：fb2png /storage/external\_storage/sdcard1/test.png，输出：

```
 bpp : 4
 width : 1920
 height : 1080
 pixel : 8294400
 r_offset : 0
 g_offset : 8
 b_offset : 16
 a_offset : 24
 r_length : 8
 g_length : 8
 b_length : 8
 a_length : 8
```

完成后在 SD 卡中查看图片是否正常。

#### ■ 确认 OSD 是否开启

输入命令：dd if=/system/xbin/busybox of=/dev/graphics/fb0 bs=7680 seek=1080，输出

```
65+1 records in
65+1 records out
501176 bytes transferred in 0.016 secs (31323500 bytes/sec)
```

完成后看屏幕是否有反应，如果有反应，证明 OSD 已开启。

## 6. 多媒体相关问题现场分析

### ● 6.1 视频黑屏异常：

- 确定视频显示在 video layer 上还是 OSD 层上

输入命令：[cat /sys/class/vfm/map](#)，输出：

```
default_osd { osd(0) amvideo4osd}
default_ext { vdin0(0) vm}
default_amlvideo2 { vdin1(0) amlvideo2}
default { decoder(0) ppmgr(0) deinterlace(0) amvideo}
provider list:
receiver list:
... ..
```

在播放视频过程中，如果输出为：

default { decoder(1) ppmgr(1) deinterlace(1) amvideo}则为 video layer 显示。

default { decoder(0) ionvideo}则为 osd 显示。

default { decoder(0) ppmgr(0) deinterlace(0) amvideo}则是 APK 自己解码，显示在 OSD 层。

- 显示在 OSD 层上

显示在 OSD 层出现黑屏现象，需要分析 ion 模块，或者是上层窗口和图层模块。

- 显示在 video layer 层上

1) 确认 video layer 是否打开：

在播放视频时输入命令 [cat /sys/class/video/disable\\_video](#)

输出【0】表示 video layer 已开启

输出【2】表示 video layer 已关闭

2) 若 video layer 已开启，则确认 OSD 是否把 video layer 挡住：

尝试关闭 OSD 层，输入命令 [echo 1 > /sys/class/graphics/fb0/blank](#) 观察输出是否有变化。

打开 OSD 层命令 [echo 0 > /sys/class/graphics/fb0/blank](#)

3) 确定是不是 ppmgr 和 deinterlace 造成的：

输入以下两条指令后，再播放视频能显示，则说明是 ppmgr 或者 deinterlace 有问题

[echo rm default > /sys/class/vfm/map](#)

[echo add default decoder amvideo > /sys/class/vfm/map](#)

4) 解码器状态是否正常：

在播放视频时输入命令 [cat /sys/class/vdec/amrisc\\_regs](#)

输出【amrisc is power off】表示解码器未打开

输出【amrisc registers show: ... ..】表示解码器正常

5) 比对寄存器(取正常情况与异常情况的寄存器值进行比对)：

在播放视频时输入命令 [echo dump v 0x1d01 0x1d40 > /sys/class/amlogic/debug](#)

[ 1040.637164@0] VCBUS[0x1d01]=0x00000780

... ..

[ 1040.866440@0] VCBUS[0x1d40]=0x0000000f

### ● 6.2 视频花屏有马赛克：

- 视频码流异常（以 H264 片源为例）

查看 [error\\_recovery\\_mode](#) 节点，输入命令 [cat /sys/module/amvdec\\_h264/parameters/error\\_recovery\\_mode](#)

去除马赛克，输入命令：[echo 2 > /sys/module/amvdec\\_h264/parameters/error\\_recovery\\_mode](#)

查看解码内容，输入命令：[echo 1 > /sys/module/amvdec\\_h264/parameters/error\\_recovery\\_mode](#)

查看错误帧，输入命令：[echo 0 > /sys/module/amvdec\\_h264/parameters/error\\_recovery\\_mode](#)

# Amlogic Application Notes

- 排除 ppmgr 和 deinterlace 模块的问题

在播放之前输入如下指令：

```
echo rm default > /sys/class/vfm/map
```

```
echo add default decoder amvideo > /sys/class/vfm/map
```

如果还出现花屏，就可以排除 ppmgr 和 deinterlace 模块的问题，否则与这两个模块相关。

- 确认是否微码问题，查找以下两个打印

**fatal error happend** 微码发生不知名错误

**Over decoder supported size** 超出解码器支持的宽高尺寸

- 6.3 确认 DI 造成的花屏，黑屏等问题：

- 输入命令：**echo 1 > /sys/module/di/parameters/bypass\_all**。确认还是有花屏和黑屏等现象。

- 输入命令：**echo state > /sys/class/deinterlace/di0/debug**，查看 di buffer 状态。输出：

```
[1999.055367@0] version 2014-06-18a, provider vframe level 0, init_flag 0, is_bypass 0, receiver_is_amvideo 1
...
[1999.218973@0] vf_peek()=>0x (null)
[1999.222441@0] di_process_cnt = 399064, video_peek_cnt = 0, force_trig_cnt = 199532
```

- 6.4 视频抖动异常：

- 确认 DI 是否被 bypass

播放过程中，输入命令：**cat /sys/module/di/parameters/bypass\_state**，输出【1】为已被 bypass。

若被 bypass，可进入目录 **cd /sys/module/di/parameters/**，查看 bypass 相关节点：

```
bypass_1080p
bypass_3d
bypass_all
bypass_dynamic
bypass_get_buf_threshold
bypass_hd
bypass_hd_prog
bypass_interlace_output
bypass_post
bypass_post_state
bypass_prog
bypass_state
bypass_superd
bypass_trick_mode
```

看是哪个结点造成 di bypass 的。把它设成 0 之后，然后再播放，看是否还抖动。

- RDMA 是否被关闭

输入命令：**cat /sys/module/rdma/parameters/enable**

输出【1】已开启

输出【0】已关闭

若 RDMA 被关闭，尝试将其打开 **echo 1 > /sys/module/rdma/parameters/enable** 看是否还抖动现象

- 6.5 视频画面卡住或位置比例异常：

- 视频画面卡住

输入命令：**logcat -s amplaye**，查找以下打印：

```
l/amplayer(119): **[update_state]abuflevel=0.00000000 vbublevel=0.99946618 abufrp=0 vbufrp=1c5b7f00 read_end=0
```

如果 vbublevel 是满的 0.99 以上一直不变，画面卡住或者黑屏，请连续多次 **cat /sys/class/vdec/amrisc\_regs** 后保存 log 给多媒体组分析。

- 视频位置不对，或者是视频比例和尺寸不对

**cat /sys/class/video/axis** 视频的位置（0 0 -1 -1 为全屏）

**cat /sys/class/video/screen\_mode** 视频宽高比 0:normal;1:full stretch;2:4-3;3:16-9

**echo dump v 0x1d01 0x1d40 > /sys/class/amlogic/debug** 查看相关寄存器，与正常状态进行对比。

# Amlogic Application Notes

- 6.6 音频解码异常:

- 查看解码器是否解码出充足的数据

使用命令: `cat /sys/class/audiodsp/dsp_working_status`, 输出:

```
dsp status 0x0
....
dsp pcm rp 0x0
dsp pcm buffer level 0x0
dsp pcm buffered size 0x0
dsp es read offset 0x0
```

- 查看 `audioflinger` 是否经常发生 `underun`

使用命令: `dumpsys media.audio_flinger` (输出中搜索 `underun`)

- 6.7 音视频卡顿:

- 确定是 `video` 问题还是 `audio` 问题

只播放视频: `setprop media.amplayer.noaudio true`

只播放音频: `setprop media.amplayer.novideo true`

- 查看 `buffer` 状态

输入命令: `logcat -s amplaye`, 查找以下打印:

```
I/amplayer(119): **[update_state]pid:1 status=PLAYING(last:PLAYING) err=0x0 curtime=0 (ms:800) fulltime=4077
lstime=0
```

```
I/amplayer(119): **[update_state]abuflevel=0.00000000 vbublevel=0.99946618 abufrp=0 vbufrp=1c5b7f00 read_end=0
```

查看 `abuflevel` `vbublevel` 是否数据量足够, 是不是比较空。

- 看是否有 `pts` 跳变, 命令如下:

|                                                              |                                             |
|--------------------------------------------------------------|---------------------------------------------|
| <code>echo 1 &gt; /sys/class/tsync/debug_audio_pts</code>    | 查看 <code>audio pts</code>                   |
| <code>echo 1 &gt; /sys/class/tsync/debug_pts_checkin</code>  | 查看 <code>checkin</code> 的 <code>pts</code>  |
| <code>echo 1 &gt; /sys/class/tsync/debug_pts_checkout</code> | 查看 <code>checkout</code> 的 <code>pts</code> |
| <code>echo 1 &gt; /sys/class/tsync/debug_video_pts</code>    | 查看 <code>video pts</code>                   |

- 6.8 音视频不同步:

- 是否已开启音视频同步, 输入命令 `cat /sys/class/tsync/enable`, 输出:

【1: enabled】已开启

【0: disabled】未开启

- 查看同步的模式, 输入命令 `cat /sys/class/tsync/mode`

- 查看 `PTS` 的变化

|                                                              |                                             |
|--------------------------------------------------------------|---------------------------------------------|
| <code>echo 1 &gt; /sys/class/tsync/debug_audio_pts</code>    | 查看 <code>audio pts</code>                   |
| <code>echo 1 &gt; /sys/class/tsync/debug_pts_checkin</code>  | 查看 <code>checkin</code> 的 <code>pts</code>  |
| <code>echo 1 &gt; /sys/class/tsync/debug_pts_checkout</code> | 查看 <code>checkout</code> 的 <code>pts</code> |
| <code>echo 1 &gt; /sys/class/tsync/debug_video_pts</code>    | 查看 <code>video pts</code>                   |

- `Seek` 或者切台时声音先出来, 画面再动

查找有无 `sys.amplayer.drop_pcm` 属性, 如果有, 将其设为 1, 使用命令 `setprop media.amplayer.novideo 1`

- 6.9 软件 `demux` 设置:

- 网络点播 `ts` 设为软件 `demux`

`setprop libplayer.netts.softdemux 1`

- 网络直播 `ts` 设为软件 `demux`

`setprop libplayer.livets.softdemux 1`

- 本地 ts 文件播放设为软件 demux  
`setprop libplayer.ts.softdemux 1`

- 6.10 抓取播放视频流和网络流:

- 播放在线视频 dump 数据

- 1) 创建一个用于保存 dump 数据的文件夹, 输入命令: `mkdir /data/tmp`
- 2) 修改文件夹权限, 输入命令: `chmod 777 /data/tmp`
- 3) 设置 dump 数据类型, 输入命令: `setprop media.libplayer.dumpmode x`  
x 的值:

【1】 ts,ps,rm 几种流的 raw data 从文件或网络读到的数据 dump

【2】 ts,ps,rm 几种流的 raw data 写入解码 buffer 的数据 dump

【4】 es 码流 video 数据, 读数据 dump

【8】 es 码流 video 数据, 写数据 dump

【16】 es 码流 audio 数据, 读数据 dump

【32】 es 码流 audio 数据, 写数据 dump

dump 到的数据存放在 `/data/temp/pidn_dump_xx.dat` 以 pidn 开头, 可以区分是哪个线程 dump 出来的数据。

播放 ts 如果开 softdemux 后 dump 不到 ts 流, 需要按 es 流 dump

- 通过 tcpdump 工具抓包 (以 k200 为例)

网络播放碰到一些问题时, 可以通过 tcpdump 命令抓包

默认升级包里面没有这个命令, 可以从下面编译路径拷贝 android 中 `/system/bin` 目录, 并修改权限。

`out/target/product/k200/symbols/system/xbin`

用法: `tcpdump -s 0 -w /data/test.log`

- 查看媒体文件的基本信息

视频文件播放有问题, 可以下载 mediainfo 工具软件, 该软件可查看媒体文件基本信息。

## 7. 系统 suspend, reboot 死机手册

对于系统出现死机现象, 可以打开 boot monitor 查看一些启动信息, 只需将 menuconfig 中 `CONFIG_BOOT_MONITOR=y` 即可

- 7.1 系统 SUSPEND 问题:

- 不断在后台自动休眠/唤醒, 而屏幕又不亮

该问题是 android 上层的 alarm list 问题, 偶尔出现这个现象是正常的。但是如果频繁出现则有问题, 找 android engineer check 下 alarm list。

- 休眠后, 按唤醒源没有任何反应, 一句 log 都没有输出

这个问题往往是唤醒源不对, 即新增的唤醒源没有加入唤醒源中, 致使系统无法唤醒。还有就是检查 32k 晶振是否起振。如果起振, 各路电是否按预期恢复了。

- 待机唤醒缓慢

首先查看下 kernel log, 查看哪个 driver 在 resume 的时候用的时间过多。其次, 关闭 kernel log 可以省掉 0.5s resume 时间。

- suspend 过程中死机, 并且连 ICE 无法 stop

这个原因往往是某个模块在 suspend 时候 gate off 后, 其它模块由于中断或者什么原因又去访问该模块资源, 造成死机, 这个一般发生在多媒体部分, 如 VPU 这类的很多模块在共用资源的时候发生。

- resume 唤醒后在亮屏后或者接近亮屏后死机

这个很多时候和 wifi 有关。请关闭 wifi 进行排除煲机。

■ Resume 的 log 停在“restart arm”后就没了  
往往也是 ddr 的问题，需要检查 ddr1.5v 和 ddr 自身的稳定性。

## ● 7.2 系统 REBOOT 问题:

■ Kernel 启动过程中出现随机死机问题

- 1) 分析异常 log 看是否能找出原因，如果没有异常 log 打出，则连上 ICE 去 dump log: info os-log。
- 2) 测量死机现场的各路电是否正常，包括 vcck, AO\_EE, ddr1.5, 3.3v。防止发生 PMU 引起的跌落问题。如果是瞬间跌落又恢复了当然看不到。
- 3) 排除 vcck 问题，通过 vcck 的动态调压表格把 vcck 定在最高值，并进行 reboot 煲机，如果不死后，则需要重新调整 vcck 表格。
- 4) 排除 ddr 稳定性，可以降频 DDR 试试，并查看 DDR 纹波。
- 5) 特定平台出现随机死机，且以上条目都排除后，则需要排查 pll patch 问题（针对 M8/M8baby）。确保该 patch 已经打入。

■ 偶尔死在 uboot 中，出现 ucl 解压失败  
该现象一般是 ddr 问题。

■ Reboot 准备关机的时候死机

- 1) 这个问题往往和一些 driver 的 shut\_off 的 call back 有关，需要注意 log 的位置或者连 ICE，一般能看到我们想要的信息。
- 2) 我们需要#define CONFIG\_ARM\_FLUSH\_CONSOLE\_ON\_RESTART 这个 config 来看更多 reboot 前的 log 信息。但是打开这个 config 也会有一定的风险，会出现 console 的输出清理死锁问题。#define CONFIG\_ARM\_FLUSH\_CONSOLE\_ON\_RESTART 这个 config 是否打开，包括 defconfig 的和 code 中的两个地方。

## ● 7.3 关于 WATCHDOG 开关问题:

■ 目前 aml watchdog 属于硬件 watchdog 覆盖的死机种类(kernel 喂狗的方式)

- 1) Unable to handle kernel NULL pointer dereference at virtual address 00000000

测试方法:

```
int *p=NULL;
```

```
*p=2;
```

- 2) 硬件死机，例如 bus 上的死机

软件死机 硬件 watch dog 无法覆盖，可以打开 kernel 的相关宏，可以解决部分软件死机的状况:

1 spin\_lock 死锁:

```
CONFIG_LOCKUP_DETECTOR=y
CONFIG_BOOTPARAM_HARDLOCKUP_PANIC=y
CONFIG_BOOTPARAM_SOFTLOCKUP_PANIC=y
CONFIG_PANIC_TIMEOUT=2
```

如果发生 spin\_lock 死锁，系统 20s 后 dump stack 2 秒后重启系统。

2 mutex 死锁，kernel 空间的 while(1) 死循环

```
CONFIG_DETECT_HUNG_TASK=y
CONFIG_BOOTPARAM_HUNG_TASK_PANIC=y
CONFIG_PANIC_TIMEOUT=2
```

120 秒后 dump stack 2 秒后重启系统

对于 kernel，只是多开了几个线程，安兔兔 实测 对跑分没有影响。



## ■ 源码的定义

```
amlogic-watchdog{
 compatible = "amlogic,aml-wdt";
 status = "okay";
 one-second=<0x100000>;
 min_timeout=<1>;
 max_timeout=<41>;
 default_timeout=<5>;
 reset_watchdog_method=<1>;//0:sysfs,1:kernel
 reset_watchdog_time=<2>;
 shutdown_timeout=<10>;
};
status = "okay"; // "okay":enable this driver,"disable":disable this driver 默认 disable
one-second=<0x100000>; //watch dog 计时 1s 的 count 数目。
min_timeout=<1>; //watch dog 的最小超时时间
max_timeout=<41>; //watch dog 的最大超时时间
default_timeout=<5>; //watch dog 默认超时时间
reset_watchdog_method=<1>; //0:sysfs,1:kernel 选择 通过上层喂狗还是 kernel 喂狗
reset_watchdog_time=<2>; //kernel 喂狗的时间间隔
shutdown_timeout=<10>; //关机的时候 设置 watch dog 超时时间
```

## 8. 内存信息的查看和分析

### ● 8.1 检查内存泄漏:

■ 通过 `./device/amlogic/g24ref/quick_build_kernel.sh menuconfig` 分别打开

1) `DEBUG_KMEMLEAK`: 它有依赖的选项, 不过依赖默认设置好了。

可以再 `menuconfig` 里面查找 `DEBUG_KMEMLEAK`, 看到它的依赖条件的打开情况;

`menuconfig` 的查找方法: 在界面直接输入 `/"` 打开查找界面, 输入 `DEBUG_KMEMLEAK` 即可。

这个选项在文件 `commonlib/Kconfig.debug`。

2) `SLUB_DEBUG`: 默认已经打开。目前我们使用的是 `slub` 而不是 `slab`。这个选项在 `init/Kconfig`。

3) `debugfs /sys/kernel/debug`: 此项已经 `mount`; 命令: `mount -t debugfs nodev /sys/kernel/debug`

■ 检查方法:

1) 查看命令: `cat /sys/kernel/debug/kmemleak`, 缺省 10 分钟扫描一次。

2) 扫描命令: `echo scan > /sys/kernel/debug/kmemleak`, 可以立即触发扫描。

3) 清除命令: `echo clear > /sys/kernel/debug/kmemleak`

4) 设置扫描间隔: `echo scan=60 > /sys/kernel/debug/kmemleak`, 该命令以秒为单位 (示例中为 60s)

### ● 8.2 查看内存信息:

内存上的跟踪重点查看两个信息

■ 查看 `meminfo`:

命令: `cat /proc/meminfo`

■ 导出内存分配信息:

命令: `echo m > /proc/sysrq-trigger`

注: 对于 `low mem`, 也需要查看: `cat /proc/sys/vm/mem_management_thresh` 的值, 可以再需要的地方添加函数:

`show_mem(0)` 查看当前的内存分布;



## 9. M6 项目烧录注意事项及出现问题时需提供相关的信息汇总

### 9.1 工位配置：

#### ■ PC 硬件要求

2G 以上主频

1G 内存

主板比较重要，请选择品牌主板，如华硕等，必须支持 USB2.0 高速模式

<注>：不是所有 USB2.0 都支持高速模式

#### ■ 操作系统要求

XP 或 WIN7

关闭杀毒类监控软件

#### ■ HUB 及连接线

“SSK” 魔王 USB HUB--SHU023(7 口)

7 根烧录所用的连接线最好也用“SSK”连接线，避免使用较长或质量较差的连接线。若平台到 HUB 的连接线较长，可能会导致 PC 机端到 HUB 端的电压降低，导致连接失败；若连接线质量较差，可能会导致平台失去连接，或者变成不可识别的 USB 设备。

#### ■ PC 软件配置

1) HUB 端的电压如果超过了 PC 机 USB 口端的电压，PC 会自动将切断供电给 HUB，为了保险起见，不让 PC 机切断电源，需要连接 HUB 后在 PC 机上进行如下设置：右键【我的电脑】选择[管理]选项->【设备管理器】->右键【USB Root Hub】选择[属性]选项->点击【电源管理】->将【允许计算机关闭此设备以节约电源】选项中的勾去掉。【允许计算机关闭此设备以节约电源】默认是打开的。如下图所示。

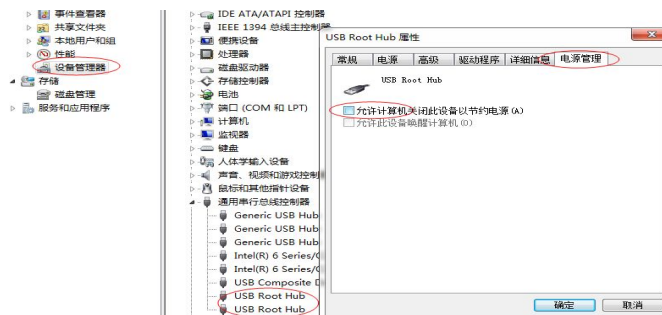


图 - USB 不断电设置

2) 如果电脑系统是 win7 系统，需要对 PC 机进行相应的设置以保证烧录工具正常运行，具体设置为：点击【控制面板】->【所有控制面板项】->【用户帐户】->【用户账户控制设置】，更改成下图的设置。



图 - 用户账户控制设置

<注>：设置后，以后在打开 USB 烧录工具时，再以右键管理员方式运行，这样可以确保在 win7 系统下烧录工具 的正常运行。

## Amlogic Application Notes

- 9.2 工具崩溃问题:

- 获取 **dump** 文件

若出现如下图所示的工具崩溃现象时，提供位于工具目录下的 **crash.dmp** 的 **dump** 文件



图 - 工具崩溃后弹出的对话框

- 9.3 烧录意外停止问题:

- 若使用过程中出现任何意外导致没有完成烧录就停止，可通过重新打开烧录工具来重新烧录。

- 9.4 其它失败或异常问题:

- 提供最近一次烧录的 **PC** 工具端的 **log**。具体文件在工具目录的 **log** 目录下。
- 提供所使用的烧录包、平台型号、**PC** 操作系统等信息
- 提供从 **uboot** 启动到 **kernel** 的 **code log** 信息。
- 提供 **recovery.log** 信息，可接上串口输入 `busybox cat /tmp/recovery.log`。

# 附录 A: M6 烧录平台端常见错误排查

### 实例分析一

问题：烧录进度停在 20%左右的错误，准确定位是在发完“运行 uboot”的命令 `<command sn="?">run 0x8f800000</command>` 后失败。

分析如下：

- 烧录用的 u-boot-orig.bin 跑起来后，首先会重置平台端的 usb 端口为高速 USB 设备，重置过程会断开与 PC 的连接，然后重新连上。
- 如果断开后无法重新连上，首先排查 PC 问题，更换烧录所用的 PC。
- 其次排查平台端 USB 口是否异常：相同 HUB 端口，更换平台排查。
- 然后排查 USB 烧录线，更换成较短较粗的数据线重新烧录。

### 实例分析二

问题：烧录进度停在 78%左右的错误，准确定位是在运行 recovery.img 时 `<command sn="?">bootm 0x82000000</command>` 失败。

分析如下：

- 首先需要确定 kernel 是否能正常启动  
如果 RTC、DDR 或 AXP 等有硬件问题，kernel 无法正常启动，烧录工具就会在这一步后失败。  
确定的方式是：用 SD 卡对相关平台进行升级。
- kernel 正常启动后会进入 usb\_recovery 模式，这种模式下会运行脚本把 media 或 cache 分区格式化成大容量存储设备，可被 PC 端的操作系统枚举成可移动存储设备，然后 PC 端会把烧录包中的 update.zip 包拷贝到该分区。所有这些操作都是在 bootm 这一条命令里完成的。如果在此过程中失败，排查流程如下：
  - 首先，排查是否因为超时问题，该命令有个配置属性是等待“可移动存储设备”出现的最长时间（count="xxx"），如果此处的值设置的比较小（默认为 30 秒），可能会导致超时报错。排查方法为：  
工具报错后，再等一段时间，查看“我的电脑”里是否有新的可移动磁盘出现，如果报错后出现新的可移动磁盘，则说明是超时问题，增大该处属性的值即可，如 count="500"。
  - 其次，排除杀毒、拦截软件的原因：win7 系统和装有杀毒、拦截软件的系统会使可移动磁盘弹出的时间增长，甚至无法弹出。
  - 然后，排除 PC 原因，实践证明：有些平台在某个 PC 上无法正常弹出 disk，但换到其他 PC 就没有问题，问题的根源与 PC 操作系统的枚举机制有关，虽不能探其根源，但也可以用来尝试解决产线的紧急问题。
  - 最后，更换 PC 也无法解决问题，可考虑对目标平台接上串口查看 recovery 的运行状态，以及提供详细的 Log(PC 端和平台端)带回公司分析。

### 实例分析三

问题：烧录进度停在 87%左右的错误`<command sn="? " count="400">update:ota-package.zip</command>`。

分析如下：

- 到此步以后，正式进入 android 的经典升级模式，与卡升级的原理、流程完全相同。升级的时间与升级包的大小有关，一般的升级包为 200M 左右，参考时间为 250 秒。如果此步出错，排查过程如下：
  - 首先，排查超时问题，原理方法与平台端实例分析二相同。
  - 其次，在不接串口的情况下，可以打开 PC 端该平台对应的 disk，打开后会发现有个 recovery.log，其中记录了该平台此次的整个升级流程。可以现场做出分析判断，也可联系同事寻求远程支援。
  - 最后，如果发现是校验升级包时失败，首先请硬件专业人员查看 CPU 对 DDR 的引脚是否有短路、断接、虚焊，然后请工厂协助更换 DDR 尝试解决。拆卸，焊上过程要小心。换完后再升级，来验证是否解决。
  - 若以上还不能解决，考虑是否本身的升级包是否有问题。可用 sdcard 进行升级一次。

# 附录 B: M6 烧录 PC 端常见错误排查

## 实例分析一

问题：PC 端无法识别平台设备。

分析如下：

- 排除平台设备本身的问题。判断方法是，设备连入到可以正常烧录的 PC 上，可以烧录成功。
- PC 系统没有安装过 libusb 驱动。安装与 PC 系统版本匹配的 libusb 驱动。驱动路径为 USB 烧录 PC 工具所在目录\AmlogicusbBurningdriver。
- PC 端安装与其版本不匹配的 libusb 驱动。问题现象是，PC 接入烧录设备后显示找到 libusb 驱动，但是打开设备管理器，找不到 libusb-win32 devices 节点。卸载原来驱动，安装正确版本驱动可以解决这类问题。

## 实例分析二

问题：在一些客户 PC 上，接入休眠状态的 USB 设备，会造成 PC 死机。

分析如下：

- 问题原因是客户 PC 不支持 USB 设备休眠。禁止设备休眠可以解决这类问题。

## 实例分析三

问题：客户错误导入了升级包，如直接导入了 SD 卡升级包（OTA 包）失败问题。

分析如下：

- USB 烧录工具所使用的烧录包不仅包含了 OTA 包，还包括其他辅助烧录的程序和文件，这些组合在一起才是烧录包。

## 实例分析四

问题：升级包数据损坏问题。

分析如下：

- 压缩包数据损坏后，解压出的文件都有损坏的可能。
- update.zip 文件最大，出现损坏概率最大。
- 一些客户产线发生这类问题，因 U 盘质量差，复制到烧录 PC 的压缩包数据损坏，造成解压出不完整 update.zip，进而出现烧录 update.zip 失败的问题。
- 压缩包的数据一旦被损坏，解压缩工具解压过程报告错误信息。
- 老版本工具后面推出改进版本，检测到解压错误信息，提示用户烧录包数据损坏。