密级状态： 绝密( ) 秘密( ) 内部资料( ) 公开(√)
Security Class: Top-Secret ( ) Secret ( ) Internal ( ) Public (√)

Rockchip_Dr.G_User_Guide

（技术部，图形显示平台中心）

(Technical Department, Graphic Display Platform Center）

文件状态： **Status:** [ ] 草稿 [ ] Draft [ ] 正在修改 [ ] Modifying [√] 正式发布 [√] Released	**文件标识：** **File No.:**	RK-SM-YF-xxx
	当前版本： **Current Version:**	V1.5
	作 者： **Author:**	黄德胜、林志雄 Huang Desheng, Lin Zhixiong
	完成日期： **Finish Date:**	2019-11-03
	审 核： **Auditor:**	
	审核日期： **Finish Date:**	

福州瑞芯微电子股份有限公司
Fuzhou Rockchip Electronics Co., Ltd.

免责声明

本文档按“现状”提供，福州瑞芯微电子股份有限公司（“本公司”，下同）不对本文档的任何陈述、信息和内容的准确性、可靠性、完整性、适销性、特定目的性和非侵权性提供任何明示或暗示的声明或保证。本文档仅作为使用指导的参考。

由于产品版本升级或其他原因，本文档将可能在未经任何通知的情况下，不定期进行更新或修改。

Disclaimer

This document is provided “as is” and Fuzhou Rockchip Electronics Co. Ltd (“the company”) makes no express or implied statement or warranty as to the accuracy, reliability, completeness, merchantability, specific purpose and non-infringement of any statement, information and contents of the document. This document is for reference only.

This document may be updated without any notification due to product version upgrades or other reasons.

商标声明

“Rockchip”、“瑞芯微”、“瑞芯”均为本公司的注册商标，归本公司所有。

本文档可能提及的其他所有注册商标或商标，由其各自拥有者所有。

Brand Statement

Rockchip, Rockchip^{TM} icon, Rockchip and other Rockchip trademarks are trademarks of Fuzhou Rockchip Electronics Co., Ltd., and are owned by Fuzhou Rockchip Electronics Co., Ltd.

All other trademarks or registered trademarks mentioned in this document are owned by their respective owners.

福州瑞芯微电子股份有限公司
Fuzhou Rockchip Electronics Co., Ltd.
地址：　　　　福建省福州市铜盘路软件园 A 区 18 号
网址：　　　　www.rock-chips.com
客户服务电话：　+86-4007-700-590
客户服务传真：　+86-591-83951833
客户服务邮箱：　fae@rock-chips.com

Fuzhou Rockchip Electronics Co., Ltd.
Address: No. 18 Building, A District, No.89,software Boulevard Fuzhou,Fujian,PRC
Website:　　　　　　　www.rock-chips.com
Customer service tel.:　　+86-4007-700-590
Customer service fax:　　+86-591-83951833
Customer service e-mail:　fae@rock-chips.com

版 本 历 史 Revision History

版本号 Version no.	作者 Author	修改日期 Revision Date	修改说明 Revision description	备注 Remark
V1.4	黄德胜 Huang Desheng 林志雄 Lin Zhixiong	2019-07-18	初始版本 Initial version release	
V1.5	林志雄 Lin Zhixiong	2019-11-03	修改部分命令运行结果 Update some commands' execution results	

目 录 Contents

1 主要功能说明 Main functions

本工具为自动诊断工具，能够很好地帮助使用者 debug 显示相关的问题。

This tool is automatic diagnostic tool which can greatly help users debugging display related issues.

通过日常的归纳与总结，找到工作中效率低的方面，并通过脚本，应用程序等方式解决，最终整合成为 Dr.G 自动诊断工具，工具主要分为：

Find the low efficiency problems through daily summary and fix them by script, application or other methods, and finally generate Dr.G automatic diagnostic tool. The tool mainly includes:

1) 系统信息收集与 log 抓取 system information collection and log capture
2) 系统频率查询和设置 system frequency checking and setting
3) 总线优先级查询和设置 bus priority checking and setting
4) RGA 自动测试 RGA auto test
5) GPU 自动测试 GPU auto test
6) VOP 自动测试 VOP auto test
7) LOG 自动检测 LOG auto detect

此工具不仅限于开发人员使用，任何人在阅读完本文档后，也可以自行 debug 解决部分问题，此工具对收集 log 与系统状态信息也有极大的帮助，能够大大减少解决问题前，客户与开发人员的交互工作与代码同步工作，提升工作效率，节约大家时间。

This tool can not only be used by developers, anyone can use it to debug some issues by themselves after reading this document. This tool is also very useful for collecting log and system status information, which can greatly improve the efficiency and save the time on the co-work and code sync between customers and developers.

2 Dr.G 平台依赖 Dr.G platform dependency

2.1 Dr.G 适用平台 Dr.G applicable platforms

本工具主要支持 kernel 4.4 的所有平台，主要有：

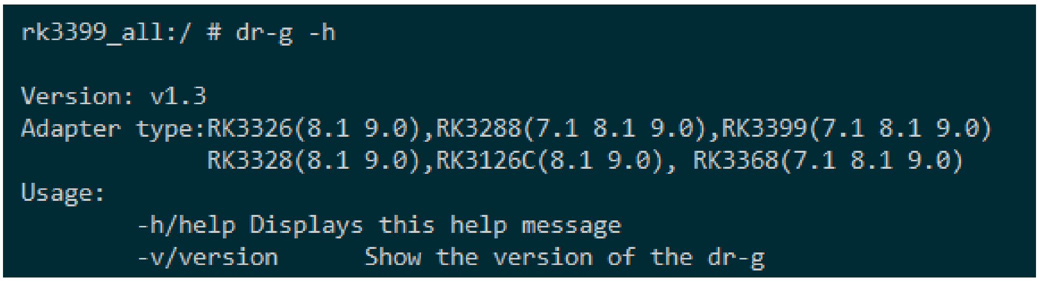
This tool mainly supports the platforms with kernel 4.4 including:

1) RK3368(7.1 8.1 9.0)
2) RK3288(7.1 8.1 9.0)
3) RK3399(7.1 8.1 9.0)
4) RK3328 (8.1 9.0)
5) RK3126C(8.1 9.0)
6) RK3326 (8.1 9.0)

可以通过如下命令查到：

Use the following command to check:

"dr-g -h"



```
rk3399_all:/ # dr-g -h

Version: v1.3
Adapter type:RK3326(8.1 9.0),RK3288(7.1 8.1 9.0),RK3399(7.1 8.1 9.0)
             RK3328(8.1 9.0),RK3126C(8.1 9.0), RK3368(7.1 8.1 9.0)
Usage:
        -h/help Displays this help message
        -v/version      Show the version of the dr-g
```

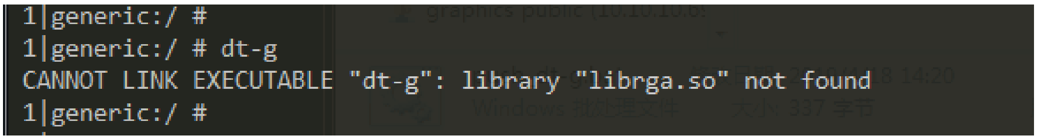
图 2-1 帮助命令

Picture 2-1 Help command

2.2 Android 平台依赖 Android platform dependency

考虑到多平台的兼容性，因此将 Dr.G 编成 32 位的 binary 来使用。应用建议放置在/system/bin 下，使用时如果发现找不到依赖库，如：librga.so not found

Considering the compatibility of multiple platforms, Dr.G is compiled as 32-bit binary for usage. Recommend to put the application in /system/bin. If the dependent lib cannot be found, for example, librga.so not found



```
1|generic:/ #
1|generic:/ # dt-g
CANNOT LINK EXECUTABLE "dt-g": library "librga.so" not found
1|generic:/ #
```

图 2-2 依赖 librga

Picture 2-2 Dependent librga

可将/vendor/lib/下的对应.so 库 拷贝至/system/lib/下，目前已经通过 push 脚本，自动完成此操作。

You can copy the corresponding .so library in /vendor/lib/ to /system/lib/. Currently this operation is already completed automatically by push script.

2.3 Dr.G 可执行程序压缩包说明 Dr.G executable program compressed package

各个平台的 Dr.G 可执行程序有 3 个压缩包，分别对应 android 平台如下表：

Each platform’s Dr.G executable program has three compressed packages, with corresponding android platforms shown as below table:

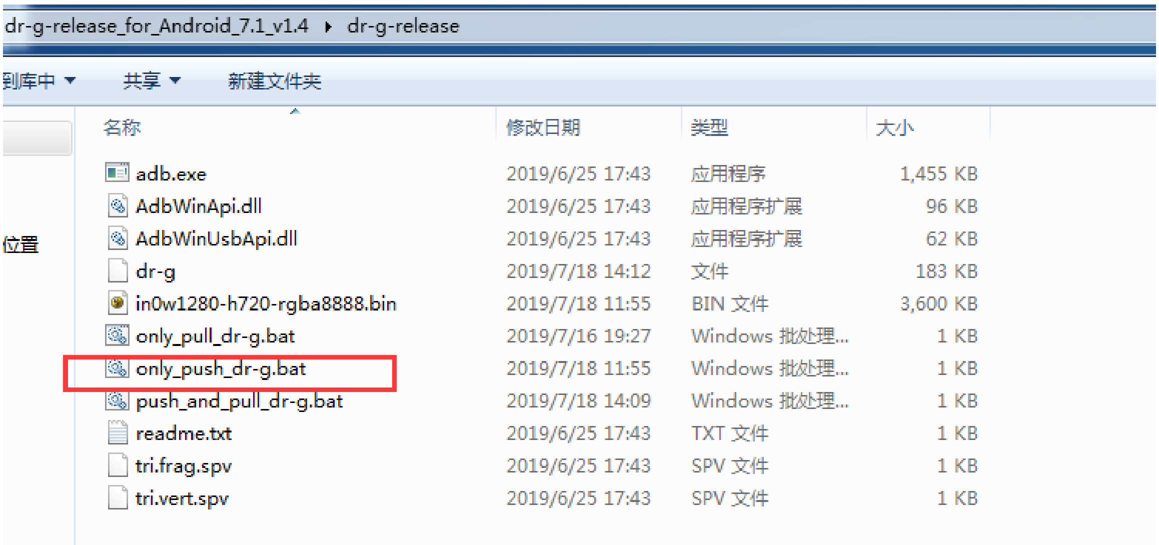
Dr.G 压缩包名称 Dr.G compressed package name	适用 android 版本 Applicable android version
dr-g-release_for_Android_7.1_v1.5.tar.gz	andorid 7.1
dr-g-release_for_Android_8.1_v1.5.tar.gz	andorid 8.1
dr-g-release_for_Android_9.0_v1.5.tar.gz	andorid 9.0

表 2-1 各版本发布

Table 2-1 Version release

使用者只需解压对应的压缩包，保证 adb 连接成功且能正常使用 adb root、adb remount、adb shell 等基本指令，运行 only_push_dr-g.bat 脚本就可以把 Dr.G 可执行程序安装到目标板子。然后在目标板子，就可以使用 Dr.G 进行排查问题。

Users only need to uncompress the corresponding package, make sure adb is connected successfully and adb root, adb remount, adb shell and other basic instructions can be used normally, executing only_push_dr-g.bat script can install Dr.G executable program into the target board. Then you can use Dr.G to debug the issues on the target board.



名称	修改日期	类型	大小
adb.exe	2019/6/25 17:43	应用程序	1,455 KB
AdbWinApi.dll	2019/6/25 17:43	应用程序扩展	96 KB
AdbWinUsbApi.dll	2019/6/25 17:43	应用程序扩展	62 KB
dr-g	2019/7/18 14:12	文件	183 KB
in0w1280-h720-rgba8888.bin	2019/7/18 11:55	BIN 文件	3,600 KB
only_pull_dr-g.bat	2019/7/16 19:27	Windows 批处理...	1 KB
only_push_dr-g.bat	2019/7/18 11:55	Windows 批处理...	1 KB
push_and_pull_dr-g.bat	2019/7/18 14:09	Windows 批处理...	1 KB
readme.txt	2019/6/25 17:43	TXT 文件	1 KB
tri.frag.spv	2019/6/25 17:43	SPV 文件	1 KB
tri.vert.spv	2019/6/25 17:43	SPV 文件	1 KB

图 2-3 使用脚本

Picture 2-3 Use the script

3 Dr.G 使用说明 Dr.G usage

3.1 系统信息收集与 log 抓取 System information collection and log capture

3.1.1 命令说明 Command description

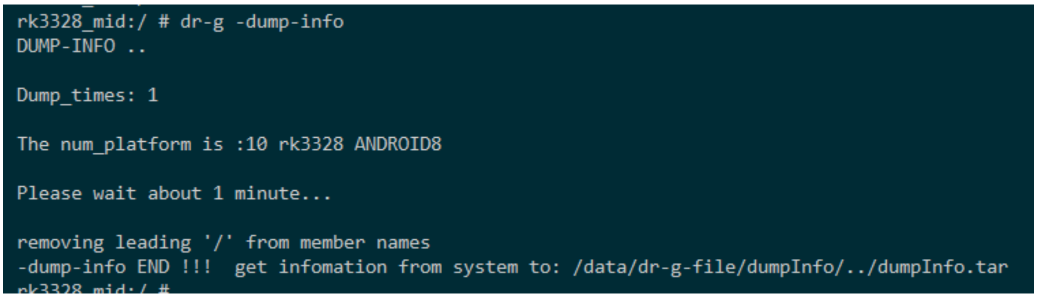
"dr-g -dump-info"

该命令运行后，会从系统中获取平台与各组件版本信息，主要有：manifest.xml、commit_id.xml、build.prop、anr_trace 文件、logcat、dmesg、io 优先级配置、CPU+GPU+DDR 当前频率与电压、CPU+GPU+DDR 最大频率、vop_clk_summary 信息、当前的 fence 状态信息、dumpsys SurfaceFlinger、ps 当前进程。

After executing this command, it will acquire the platform and component version information from the system, which mainly includes manifest.xml, commit_id.xml, build.prop, anr_trace file, logcat, dmesg, io priority configuration, CPU+GPU+DDR current frequency and voltage, CPU+GPU+DDR maximum frequency, vop_clk_summary information, current fence status information, dumpsys SurfaceFlinger, and ps current process.

运行结果： 会在/data/dr-g-file/下生成 dumpInfo.tar

Execution result: it will generate dumpInfo.tar under /data/dr-g-file/



```
rk3328_mid:/ # dr-g -dump-info
DUMP-INFO ..

Dump_times: 1

The num_platform is :10 rk3328 ANDROID8

Please wait about 1 minute...

removing leading '/' from member names
-dump-info END !!!  get infomation from system to: /data/dr-g-file/dumpInfo/../dumpInfo.tar
rk3328_mid:/ #
```

图 3-1 获取系统信息

Picture 3-1 Acquire the system information

3.1.2 脚本说明 Script description

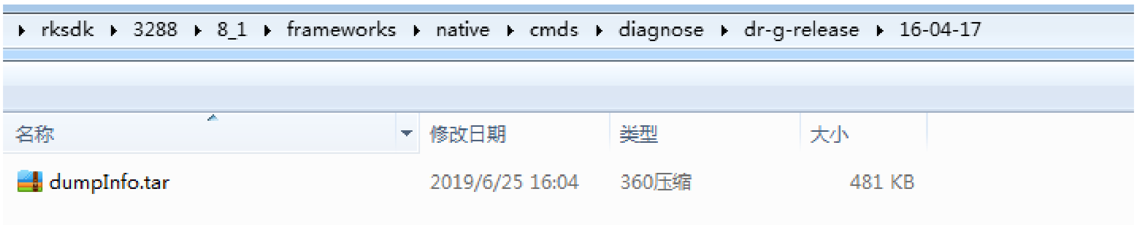
"push_and_pull_dr-g.bat"

当需要收集信息时，只需双击一个脚本，就可以轻松收集信息，并在当前路径生成一个对应时间的文件夹存放 dumpInfo.tar。

When need to collect the information, you just need to double click one script, and it will generate a folder with the corresponding time in current path to save dumpInfo.tar.

运行结果：

Execution result:



▸ rksdk ▸ 3288 ▸ 8_1 ▸ frameworks ▸ native ▸ cmds ▸ diagnose ▸ dr-g-release ▸ 16-04-17

名称	修改日期	类型	大小
dumpInfo.tar	2019/6/25 16:04	360压缩	481 KB

图 3-2 一键获取信息

Picture 3-2 One key to acquire the information

如果并不需要获取 dumpInfo.tar，想要将 dr-g 推至设备上，使用其他功能，只需双击：

If you don’t need to acquire dumpInfo.tar and want to push dr-g to the device to use other functions, only need to double click:

"only_push_dr-g.bat"

同理，单纯将 dumpInfo.tar 从设备中 pull 出来，可以双击：

Similarly, if you want to purely pull dumpInfo.tar from the device, you can double click:

"only_pull_dr-g.bat"

3.2 系统频率查询和设置 System frequency checking and setting

3.2.1 设置最大性能模式 Set the max performance mode

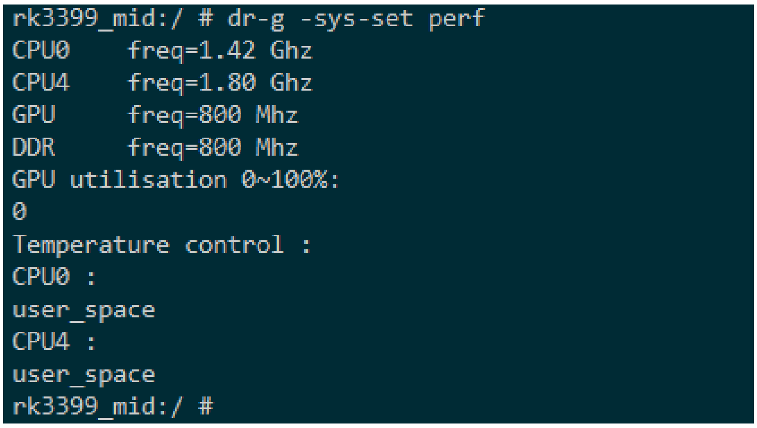
"dr-g -sys-set perf"

设置性能模式，该命令执行后，CPU，GPU，DDR 设置最高频率，温控关闭。回显打印设置后的各频率。可用于调试 APP 性能问题。

Set the performance mode. After this command is executed, CPU, GPU, DDR are set to the highest frequency, and the thermal control is disabled. The display will print the new frequencies. It can be used to debug APP performance issues.

运行结果：

Execution result:



```
rk3399_mid:/ # dr-g -sys-set perf
CPU0    freq=1.42 Ghz
CPU4    freq=1.80 Ghz
GPU     freq=800 Mhz
DDR     freq=800 Mhz
GPU utilisation 0~100%:
0
Temperature control :
CPU0 :
user_space
CPU4 :
user_space
rk3399_mid:/ #
```

图 3-3 设置最大频率

Picture 3-3 Set the max frequency

3.2.2 获取当前频率信息 Acquire current frequency information

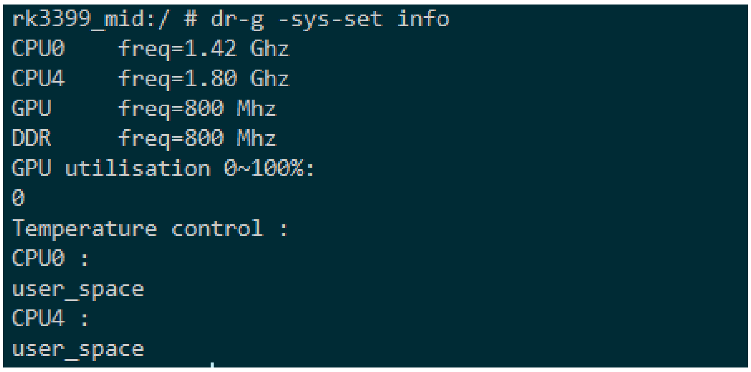
"dr-g -sys-set info"

获取当前状态，该命令执行后，回显打印当前 CPU，GPU，DDR 当前频率，温控状态。可用于调试 APP 性能问题。

Acquire current status. After this command is executed, the display will print current CPU, GPU, DDR current frequency and thermal status. It can be used to debug APP performance issues.

运行结果：

Execution result:



```
rk3399_mid:/ # dr-g -sys-set info
CPU0    freq=1.42 Ghz
CPU4    freq=1.80 Ghz
GPU     freq=800 Mhz
DDR     freq=800 Mhz
GPU utilisation 0~100%:
0
Temperature control :
CPU0 :
user_space
CPU4 :
user_space
```

图 3-4 获取频率

Picture 3-4 Acquire the frequency

3.2.3 设置 gpu 频率档 Set GPU frequency level

"dr-g -sys-set gpu low"

"dr-g -sys-set gpu mid"

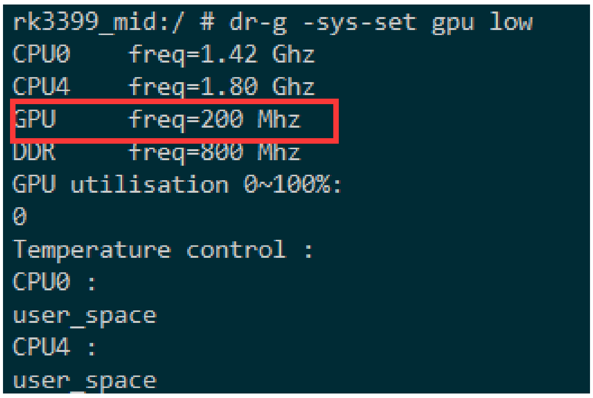
"dr-g -sys-set gpu high"

设置 GPU 三档频率，常用于 GPU 瓶颈的应用调试。上述 3 条命令运行后，CPU ，DDR 设置最高频，分别设置 GPU 频率在频率表里面低档，中档，高档。用于调试 APP 在 3 种 GPU 频率下的表现。

Set three level frequencies of GPU, which is usually used for debugging the application with GPU bottle neck. After executing above three commands, CPU, DDR are set to the highest frequency, and GPU frequency is respectively set to the low, middle, high level in the frequency table. It is used to debug the APP performance with the three GPU frequencies.

运行结果：

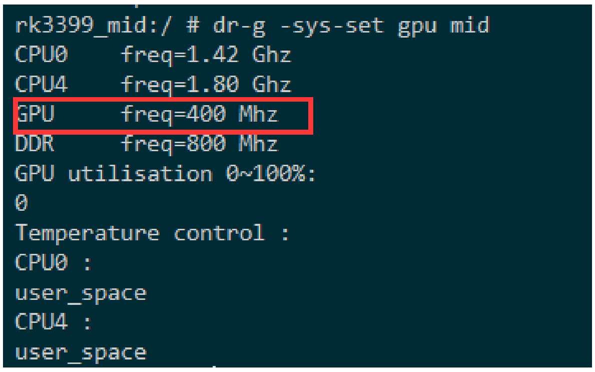
Execution result:



```
rk3399_mid:/ # dr-g -sys-set gpu low
CPU0     freq=1.42 Ghz
CPU4     freq=1.80 Ghz
GPU      freq=200 Mhz
DDR      freq=800 Mhz
GPU utilisation 0~100%:
0
Temperature control :
CPU0 :
user_space
CPU4 :
user_space
```

图 3-5 gpu 定低频

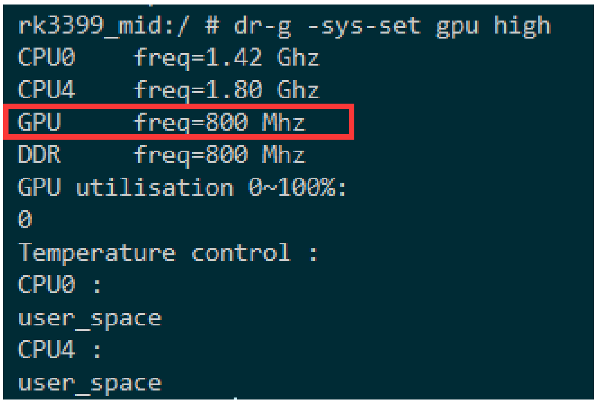
Picture 3-5 Set GPU to low frequency



```
rk3399_mid:/ # dr-g -sys-set gpu mid
CPU0     freq=1.42 Ghz
CPU4     freq=1.80 Ghz
GPU      freq=400 Mhz
DDR      freq=800 Mhz
GPU utilisation 0~100%:
0
Temperature control :
CPU0 :
user_space
CPU4 :
user_space
```

图 3-6 gpu 定中频

Picture 3-6 Set GPU to middle frequency



```
rk3399_mid:/ # dr-g -sys-set gpu high
CPU0     freq=1.42 Ghz
CPU4     freq=1.80 Ghz
GPU      freq=800 Mhz
DDR      freq=800 Mhz
GPU utilisation 0~100%:
0
Temperature control :
CPU0 :
user_space
CPU4 :
user_space
```

图 3-7 gpu 定高频

Picture 3-7 Set GPU to high frequency

3.2.4 设置 cpu 频率档 Set CPU frequency level

"dr-g -sys-set cpu low"

"dr-g -sys-set cpu mid"

"dr-g -sys-set cpu high"

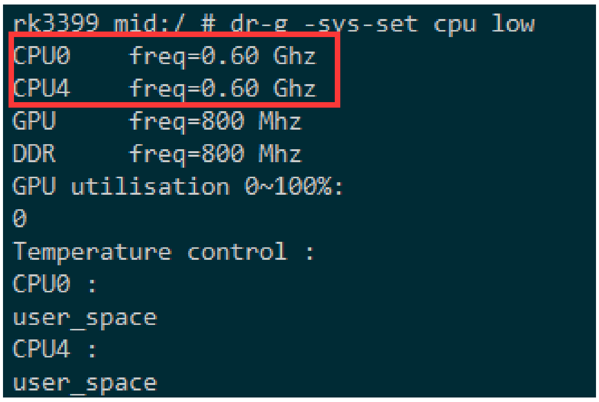
设置 CPU 三档频率，常用于 CPU 瓶颈的应用调试。上述 3 条命令运行后，GPU ，DDR 设置最高频，分别设置 CPU 频率在频率表里面低档，中档，高档。用于调试 APP 在 3 种 CPU 频

率下的表现。

Set three level frequencies of CPU, which is usually used for debugging the application with CPU bottle neck. After executing above three commands, GPU, DDR are set to the highest frequency, and CPU frequency is respectively set to the low, middle, high level in the frequency table. It is used to debug the APP performance with the three CPU frequencies.

运行结果：

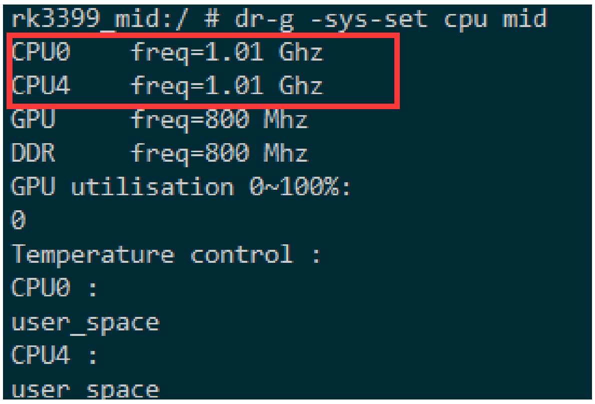
Execution result:



```
rk3399_mid:/ # dr-g -sys-set cpu low
CPU0     freq=0.60 Ghz
CPU4     freq=0.60 Ghz
GPU      freq=800 Mhz
DDR      freq=800 Mhz
GPU utilisation 0~100%:
0
Temperature control :
CPU0 :
user_space
CPU4 :
user_space
```

图 3-8 cpu 定低频

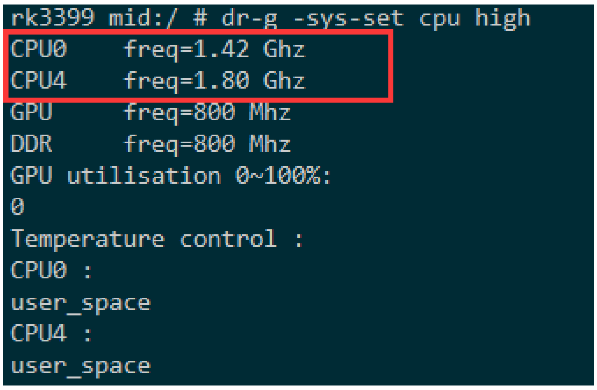
Picture 3-8 Set CPU to low frequency



```
rk3399_mid:/ # dr-g -sys-set cpu mid
CPU0     freq=1.01 Ghz
CPU4     freq=1.01 Ghz
GPU      freq=800 Mhz
DDR      freq=800 Mhz
GPU utilisation 0~100%:
0
Temperature control :
CPU0 :
user_space
CPU4 :
user_space
```

图 3-9 cpu 定中频

Picture 3-9 Set CPU to middle frequency



```
rk3399_mid:/ # dr-g -sys-set cpu high
CPU0     freq=1.42 Ghz
CPU4     freq=1.80 Ghz
GPU      freq=800 Mhz
DDR      freq=800 Mhz
GPU utilisation 0~100%:
0
Temperature control :
CPU0 :
user_space
CPU4 :
user_space
```

图 3-10 cpu 定高频

Picture 3-10 Set CPU to high frequency

3.2.5 设置 ddr 频率档 Set DDR frequency level

"dr-g -sys-set ddr low"

"dr-g -sys-set ddr mid"

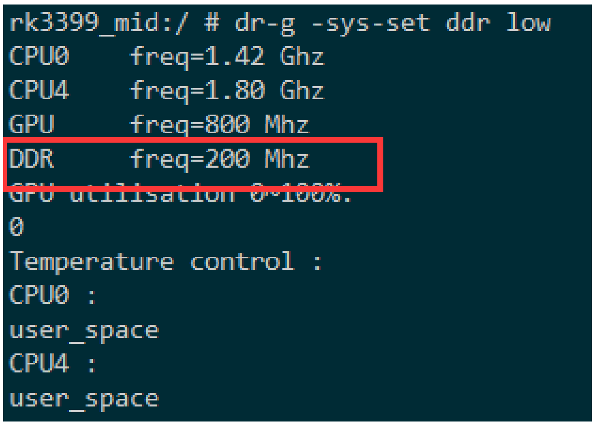
"dr-g -sys-set ddr high"

设置 DDR 三档频率，常用于 DDR 瓶颈的应用调试。上述 3 条命令运行后，CPU，GPU 设置最高频，分别设置 DDR 频率在频率表里面低档，中档，高档。用于调试 APP 在 3 种 DDR 频率下的表现。

Set three level frequencies of DDR, which is usually used for debugging the application with DDR bottle neck. After executing above three commands, CPU, GPU are set to the highest frequency, and DDR frequency is respectively set to the low, middle, high level in the frequency table. It is used to debug the APP performance with the three DDR frequencies.

运行结果:

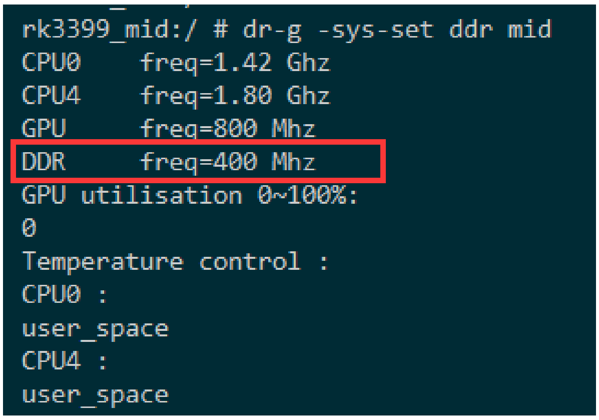
Execution result:



```
rk3399_mid:/ # dr-g -sys-set ddr low
CPU0     freq=1.42 Ghz
CPU4     freq=1.80 Ghz
GPU      freq=800 Mhz
DDR      freq=200 Mhz
GPU utilisation 0~100%:
0
Temperature control :
CPU0 :
user_space
CPU4 :
user_space
```

图 3-11 ddr 定低频

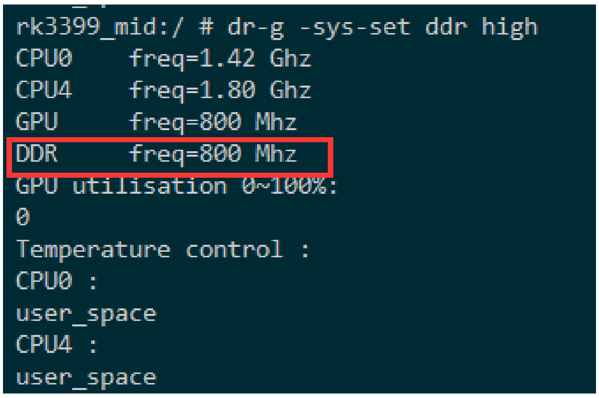
Picture 3-11 Set DDR to low frequency



```
rk3399_mid:/ # dr-g -sys-set ddr mid
CPU0     freq=1.42 Ghz
CPU4     freq=1.80 Ghz
GPU      freq=800 Mhz
DDR      freq=400 Mhz
GPU utilisation 0~100%:
0
Temperature control :
CPU0 :
user_space
CPU4 :
user_space
```

图 3-12 ddr 定中频

Picture 3-12 Set DDR to middle frequency



```
rk3399_mid:/ # dr-g -sys-set ddr high
CPU0     freq=1.42 Ghz
CPU4     freq=1.80 Ghz
GPU      freq=800 Mhz
DDR      freq=800 Mhz
GPU utilisation 0~100%:
0
Temperature control :
CPU0 :
user_space
CPU4 :
user_space
```

图 3-13 ddr 定高频

Picture 3-13 Set DDR to high frequency

3.2.6 重置为调频状态 Reset frequency scaling status

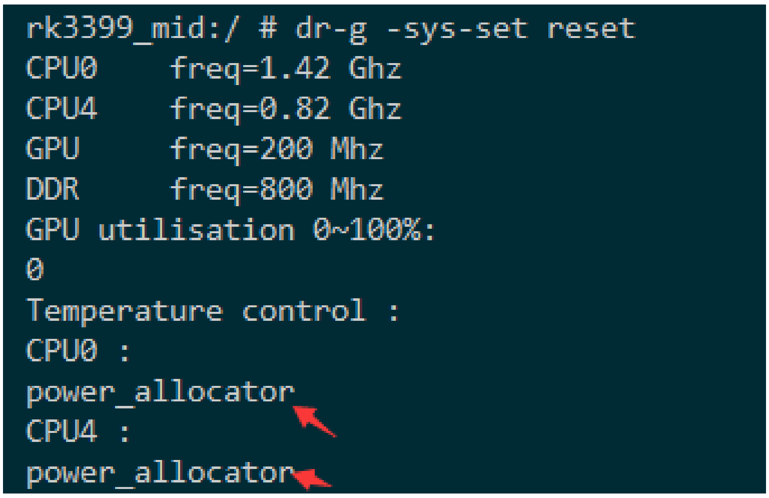
"dr-g -sys-set reset"

该命令运行后，CPU，DDR，GPU 恢复系统调频模式，温控恢复。

After executing this command, CPU, DDR, GPU are reset to system frequency scaling mode, and the thermal control is enabled.

运行结果：

Execution result:



```
rk3399_mid:/ # dr-g -sys-set reset
CPU0     freq=1.42 Ghz
CPU4     freq=0.82 Ghz
GPU      freq=200 Mhz
DDR      freq=800 Mhz
GPU utilisation 0~100%:
0
Temperature control :
CPU0 :
power_allocator
CPU4 :
power_allocator
```

图 3-14 重置调频状态

Picture 3-14 Reset frequency scaling status

3.3 总线优先级查询和设置 Bus priority checking and setting

3.3.1 查询总线优先级 Inquire bus priority

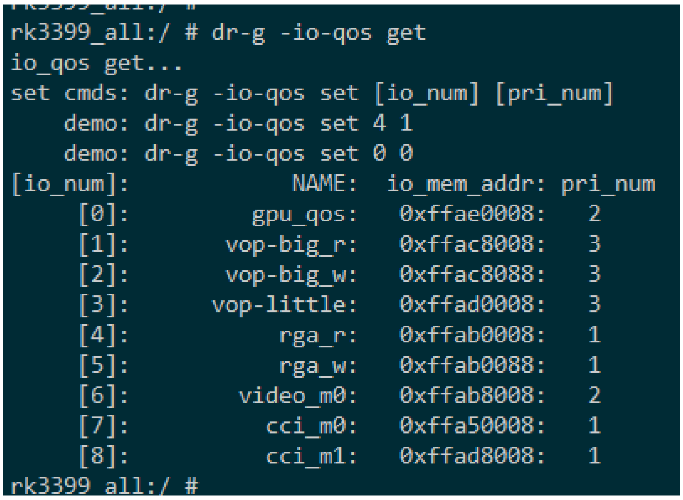
"dr-g -io-qos get"

获取总线优先级，并打印出对应的 name 表格，带有编号，后面可根据此编号进行 set。

Acquire the bus priority, and print out the corresponding name table with the number. Then you can perform set according to the number in the future.

运行结果：

Execution result:



```
rk3399_all:/ #
rk3399_all:/ # dr-g -io-qos get
io_qos get...
set cmds: dr-g -io-qos set [io_num] [pri_num]
    demo: dr-g -io-qos set 4 1
    demo: dr-g -io-qos set 0 0
[io_num]:           NAME:  io_mem_addr: pri_num
     [0]:        gpu_qos:   0xffae0008:    2
     [1]:      vop-big_r:   0xffac8008:    3
     [2]:      vop-big_w:   0xffac8088:    3
     [3]:     vop-little:   0xffad0008:    3
     [4]:          rga_r:   0xffab0008:    1
     [5]:          rga_w:   0xffab0088:    1
     [6]:       video_m0:   0xffab8008:    2
     [7]:         cci_m0:   0xffa50008:    1
     [8]:         cci_m1:   0xffad8008:    1
rk3399_all:/ #
```

图 3-15 io 优先级查询

Picture 3-15 Inquire io priority

3.3.2 设置总线优先级 Set bus priority

"dr-g -io-qos set"

设置总线优先级，按照提示内容，查找表格对应项设置，pri_num 可选值为 0~3，设置成功后会提示：" io_qos set SUCCESS !!! "。

Set bus priority. Look up the corresponding item setting in the table according to the hint. The optional value of pri_num is 0~3. After setting successfully, it will prompt " io_qos set SUCCESS !!! ".

命令 demo:

Command demo:

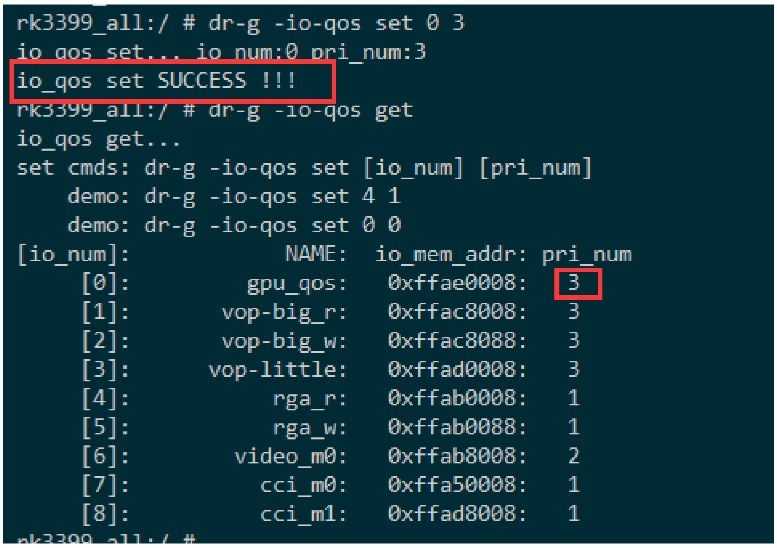
dr-g -io-qos set 4 1 //设置 get 提示中序号为[4]的优先级为 1 。 Set the priority of [4] in the get as 1

dr-g -io-qos set 1 2 //设置 get 提示中序号为[1]的优先级为 2 。 Set the priority of [1] in the get as 2

dr-g -io-qos set 0 1 //设置 get 提示中序号为[0]的优先级为 0 。 Set the priority of [0] in the get as 0

运行结果：

Execution result:



```
rk3399_all:/ # dr-g -io-qos set 0 3
io_qos set... io_num:0 pri_num:3
io_qos set SUCCESS !!!
rk3399_all:/ # dr-g -io-qos get
io_qos get...
set cmds: dr-g -io-qos set [io_num] [pri_num]
    demo: dr-g -io-qos set 4 1
    demo: dr-g -io-qos set 0 0
[io_num]:          NAME:  io_mem_addr: pri_num
     [0]:       gpu_qos:   0xffae0008:   3
     [1]:     vop-big_r:   0xffac8008:   3
     [2]:     vop-big_w:   0xffac8088:   3
     [3]:    vop-little:   0xffad0008:   3
     [4]:         rga_r:   0xffab0008:   1
     [5]:         rga_w:   0xffab0088:   1
     [6]:      video_m0:   0xffab8008:   2
     [7]:        cci_m0:   0xffa50008:   1
     [8]:        cci_m1:   0xffad8008:   1
rk3399_all:/ #
```

图 3-16 io 优先级设置

Picture 3-16 Set io priority

3.4 RGA 自动测试 RGA auto test

3.4.1 RGA 信息打印 RGA information print

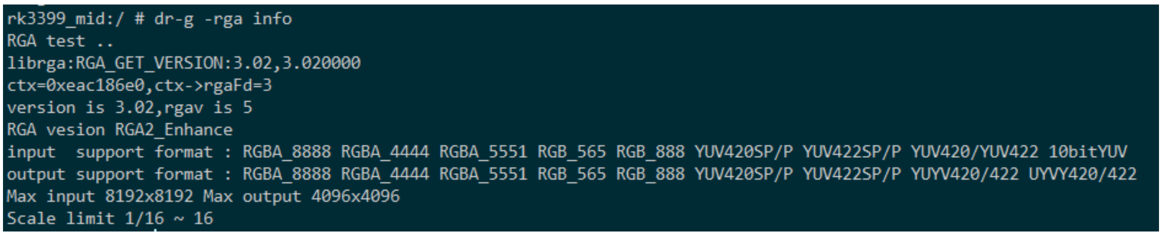
"dr-g -rga info"

打印 RGA 版本，硬件支持的数据格式，缩放倍数。最大输入输出分辨率信息。

Print RGA version, data format supported by hardware, zoom level. Maximum input/output resolution information.

运行结果:

Execution result:



```
rk3399_mid:/ # dr-g -rga info
RGA test ..
librga:RGA_GET_VERSION:3.02,3.020000
ctx=0xeac186e0,ctx->rgaFd=3
version is 3.02,rgav is 5
RGA vesion RGA2_Enhance
input  support format : RGBA_8888 RGBA_4444 RGBA_5551 RGB_565 RGB_888 YUV420SP/P YUV422SP/P YUV420/YUV422 10bitYUV
output support format : RGBA_8888 RGBA_4444 RGBA_5551 RGB_565 RGB_888 YUV420SP/P YUV422SP/P YUYV420/422 UYVY420/422
Max input 8192x8192 Max output 4096x4096
Scale limit 1/16 ~ 16
```

图 3-17 rga 信息

Picture 3-17 rga information

3.4.2 RGA 自检测试 RGA self-detection test

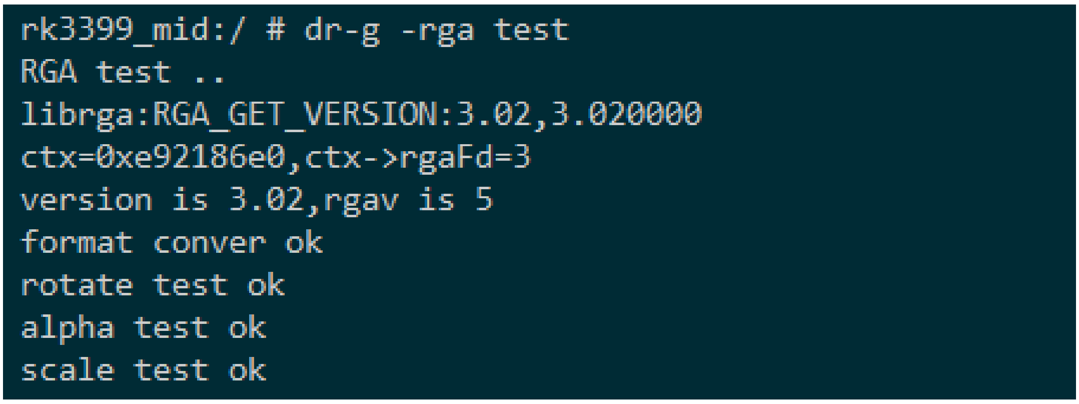
"dr-g -rga test"

根据 RGA 支持的格式做格式转换，缩放，旋转，合成的测试。所有转换的结果生成 crc 数据和预先保存的 crc 数据做比较。如果机器 RGA 本身有问题，那么生成结果与预先 crc 数据不一致，那么认为该系统 RGA 本身有问题。

Perform the format transformation, zoom, rotation and integration test based on the format RGA supported. All the transformation result will generate crc data and compare with the pre-saved crc data. If the device RGA itself has problem, the generation result will be inconsistent with the pre-saved crc data, so we can think there is problem with the system RGA itself.

运行结果:

Execution result:



```
rk3399_mid:/ # dr-g -rga test
RGA test ..
librga:RGA_GET_VERSION:3.02,3.020000
ctx=0xe92186e0,ctx->rgaFd=3
version is 3.02,rgav is 5
format conver ok
rotate test ok
alpha test ok
scale test ok
```

图 3-18 rga 测试

Picture 3-18 rga test

3.4.3 RGA 效率测试 RGA efficiency test

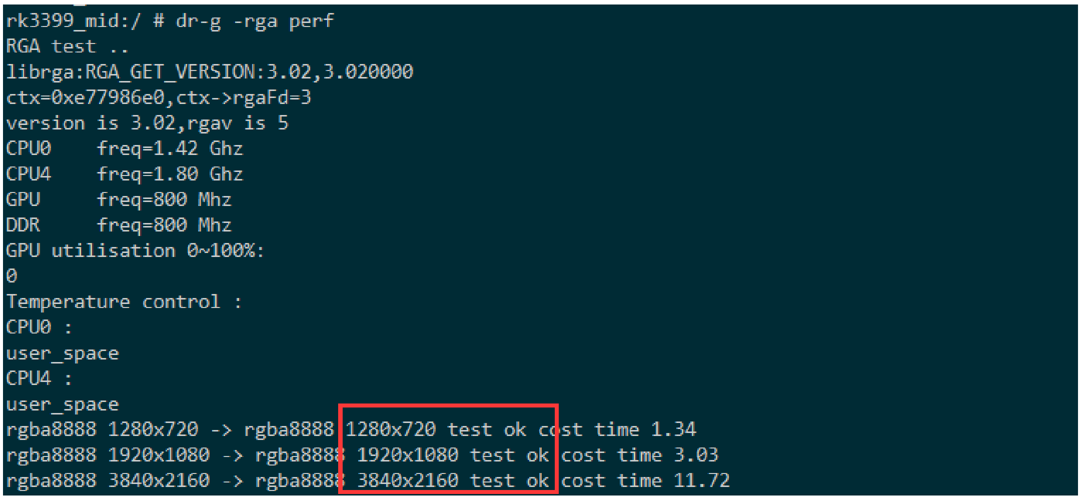
"dr-g -rga perf"

RGA 效率测试。根据 获取到的 ddr 频率以及 rga 的 aclk 计算出 720p 1080p 4k 数据理论下的耗时与当前测试耗时做比较，从而得出当前运行的 RGA 性能是否符合预期。

RGA efficiency test. Calculate the theoretical time cost under the condition of 720p 1080p 4k according to the acauired ddr frequency and rga aclk, and compare with current test data, and then make the conclusion whether current working efficiency of RGA performance meets with expectation or not.

运行结果:

Execution result:



```
rk3399_mid:/ # dr-g -rga perf
RGA test ..
librga:RGA_GET_VERSION:3.02,3.020000
ctx=0xe77986e0,ctx->rgaFd=3
version is 3.02,rgav is 5
CPU0    freq=1.42 Ghz
CPU4    freq=1.80 Ghz
GPU     freq=800 Mhz
DDR     freq=800 Mhz
GPU utilisation 0~100%:
0
Temperature control :
CPU0 :
user_space
CPU4 :
user_space
rgba8888 1280x720 -> rgba8888 1280x720 test ok cost time 1.34
rgba8888 1920x1080 -> rgba8888 1920x1080 test ok cost time 3.03
rgba8888 3840x2160 -> rgba8888 3840x2160 test ok cost time 11.72
```

图 3-19 rga 效率测试

Picture 3-19 rga efficiency test

3.4.4 RGA demo 测试 RGA demo test

"dr-g -rga demo"

通过命令参数让 rga 执行对应的功能。输出的源数据和目标数据放在 data 目录下。

Use the command parameter to execute the corresponding function of rga. Put the output source data and target data in data directory.

命令格式： dr-g -rga demo sw1280sh720sf0dw1280dh720df0rt2sx0sy0dx0dy0

The command format: dr-g -rga demo sw1280sh720sf0dw1280dh720df0rt2sx0sy0dx0dy0

注：-rga demo 后的参数串需要连续不能有空格

Note: the parameter string behind -rga demo should be continuous and without space.

源数据:

Source data:

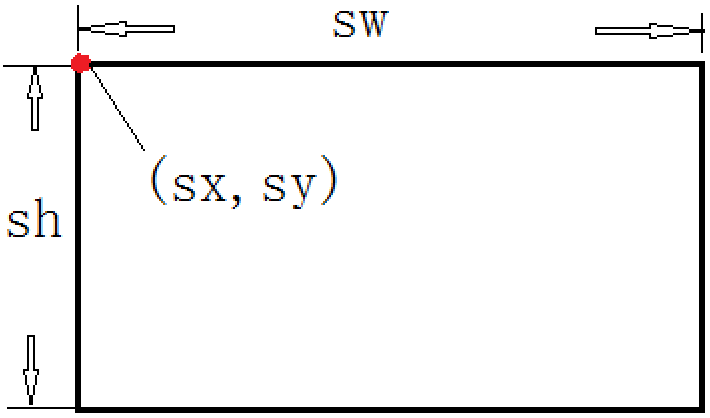


图 3-20 rga 源数据

Picture 3-20 rga source data

目标数据:

Target data:

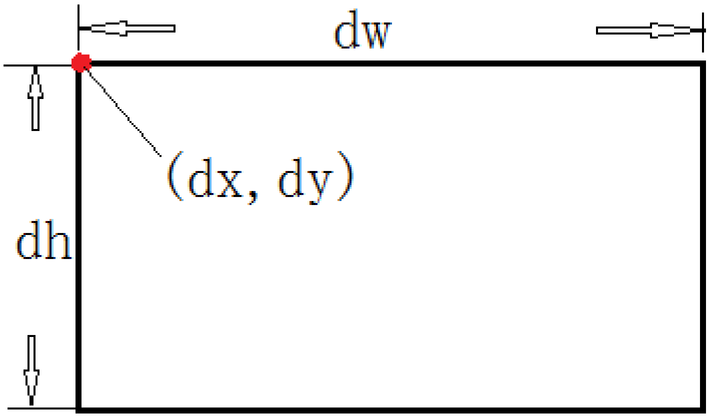


图 3-21 rga 目标数据

Picture 3-21 rga target data

参数说明:

Parameter description:

alpah 模式	blit 类型	混合公式
bd1	105	src + dst*(1-alpha)
bd2	405	src*alpha + dst*(1-alpha)

表 3-1 rga 混合公式

Table 3-1 rga mixed formula

旋转参数	rt1	rt2	rt3	rt4	rt5
对应旋转方向	rotate 90	rotate180	rotate270	mirrorH	mirrorV

表 3-2 rga 旋转参数

Table 3-2 rga rotation parameter

格式参数	sf0	sf1	sf2	sf3	sf4	sf5
对应格式	RGBA8888	RGBX8888	BGRA8888	RGB888	RGB565	NV12

表 3-3 rga 格式参数 1

Table 3-3 rga format parameter 1

格式参数	sf6	sf7	sf8	sf9	sf10	sf11	sf12
对应格式	NV21	YV12	YV21	NV16	NV61	YV16	YV61

表 3-4 rga 格式参数 2

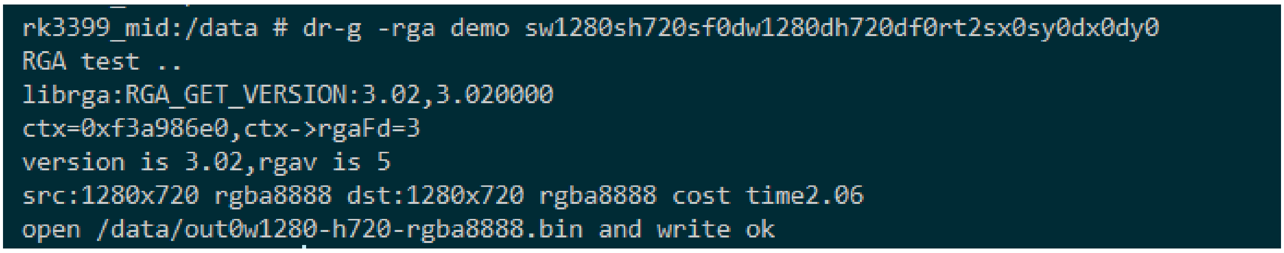
Table 3-4 rga format parameter 2

运行结果：

Execution result:

源文件 /data/in0w1280-h720-rgba8888.bin，对该图像进行 180 旋转，把旋转后的数据写到 /data/out0w1280-h720-rgba8888.bin 。可以把此 raw 数据拉出来，用工具查看。

Source file is /data/in0w1280-h720-rgba8888.bin. Rotate this image with 180 degrees, and write the data into /data/out0w1280-h720-rgba8888.bin after rotation. You can pull out the raw data and use the tool to check.



```
rk3399_mid:/data # dr-g -rga demo sw1280sh720sf0dw1280dh720df0rt2sx0sy0dx0dy0
RGA test ..
librga:RGA_GET_VERSION:3.02,3.020000
ctx=0xf3a986e0,ctx->rgaFd=3
version is 3.02,rgav is 5
src:1280x720 rgba8888 dst:1280x720 rgba8888 cost time2.06
open /data/out0w1280-h720-rgba8888.bin and write ok
```

图 3-21 rga 目标数据

Picture 3-21 rga target data

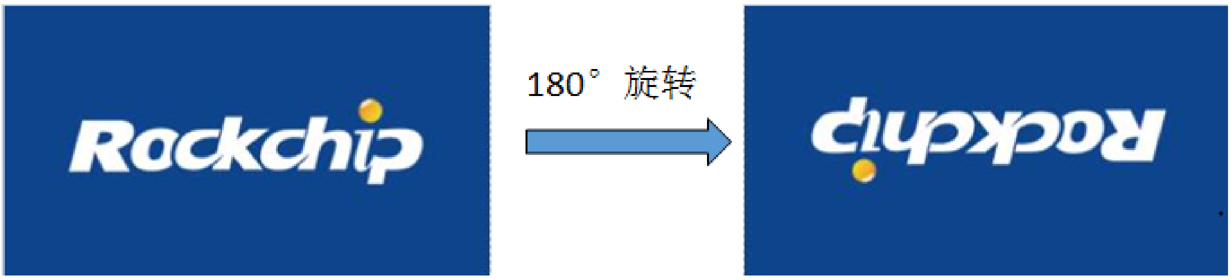


图 3-22 rga 旋转结果

Picture 3-22 rga rotation result

3.5 GPU 自动测试 GPU auto test

3.5.1 GPU 最大负载测试 GPU max load test

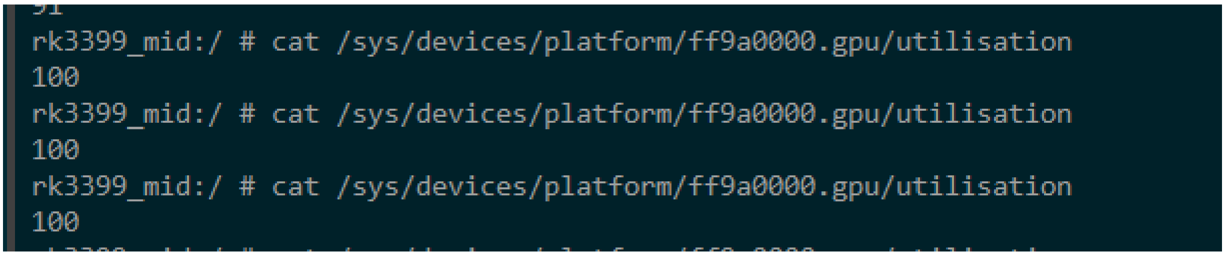
"dr-g -gpu max-load"

最大负载测试，同时可以给定 run 的时间,后面直接加时间（单位秒）。如 dr-g -gpu max-load 100, 最大负载运行 100 秒。该测试运行后，GPU 负载接近 100%。可用于老化测试，增加系统负载等测试。

Test the max load, with the specific run time, directly adding the time at the end (unit: second). For example, dr-g -gpu max-load 100, means running 100s with the max load. After this testing, GPU load is close to 100%. It can be used for stress test to add the system load.

运行结果:

Execution result:



```
rk3399_mid:/ # cat /sys/devices/platform/ff9a0000.gpu/utilisation
100
rk3399_mid:/ # cat /sys/devices/platform/ff9a0000.gpu/utilisation
100
rk3399_mid:/ # cat /sys/devices/platform/ff9a0000.gpu/utilisation
100
```

图 3-23 gpu 最大负载

Picture 3-23 GPU max load

"dr-g -gpu max-load-bg"

后台运行最大负载测试 ，不显示在前端，不影响当前显示内容。其他功能与 max-load 一致。

Test the max load with background running, which doesn’t display in front and will not affect current display content. Other functions are the same as max-load.

3.5.2 GPU 最高帧率测试 GPU max frame rate test

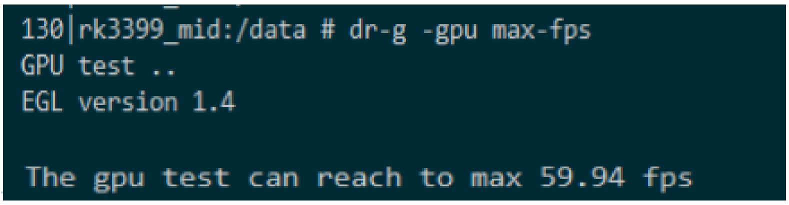
"dr-g -gpu max-fps"

运行成功后，shell 打印 GPU 应用在当前系统下可以达到的最高帧率。最高帧率就是屏的刷新率，如果打印的值与 60fps 相差较大，那么需要排除屏的刷新率配置。

After running successfully, shell will print the max frame rate of GPU application in current system. The max frame rate is the refresh rate of the panel. If the print value is much different from 60fps, need to check the refresh rate configuration of the panel.

运行结果:

Execution result:



```
130|rk3399_mid:/data # dr-g -gpu max-fps
GPU test ..
EGL version 1.4

 The gpu test can reach to max 59.94 fps
```

图 3-24 gpu 最大帧率

Picture 3-24 GPU max frequency

3.5.3 GPU 性能测试 GPU performance test

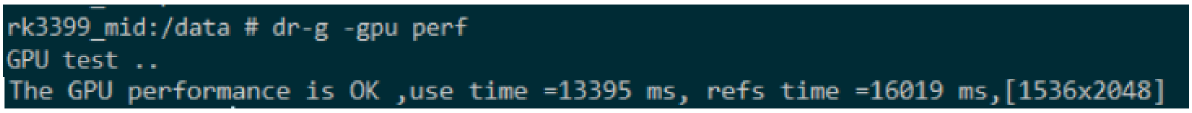
"dr-g -gpu perf"

性能测试，对当前系统 GPU 性能是否满足预期测试，测试完成会打印本次测试消耗的时间，当前平台完成的参考时间。测试前需要把系统 CPU，DDR，GPU 频率定最高（ dr-g -sys-set perf ）。如果测试消耗时间比参考时间大 10% 以上，那么认为该机器 GPU 性能不能充分发挥。

Performance test, judges whether GPU performance of current system meets with expectation or not. After testing, it will print the time consumed during this test, the reference completion time of current platform. Before test, need to set CPU, DDR, GPU frequencies to the highest (dr-g -sys-set perf). If the time consumed is larger than reference time over 10%, GPU performance of the device is considered not to be fully used.

运行结果:

Execution result:



```
rk3399_mid:/data # dr-g -gpu perf
GPU test ..
The GPU performance is OK ,use time =13395 ms, refs time =16019 ms,[1536x2048]
```

图 3-25 gpu 性能测试

Picture 3-25 GPU performance test

3.5.4 GLES 版本支持测试 GLES version support test

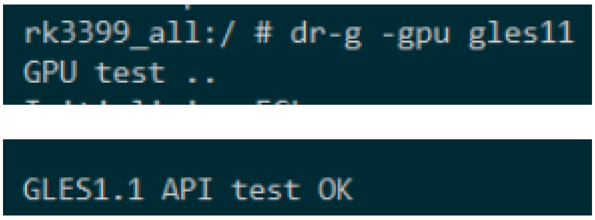
"dr-g -gpu gles11"

对 GLES 1.1 测试，运行后，shell 打印 egl 信息表示 GLES1.0 运行成功。

Test GLES 1.1, after executing, shell prints egl information means GLES1.0 is executed successfully.

运行结果:

Execution result:



```
rk3399_all:/ # dr-g -gpu gles11
GPU test ..

GLES1.1 API test OK
```

图 3-26 gles 1.1 测试

Picture 3-26 gles 1.1 test

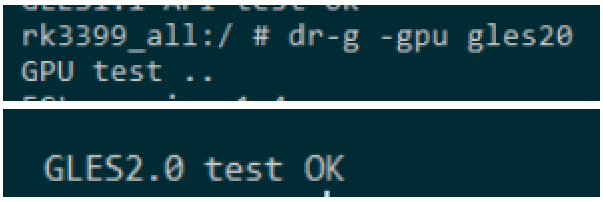
"dr-g -gpu gles20"

对 GLES 2.0 测试，运行后，shell 打印 egl 信息表示 GLES2.0 运行成功。

Test GLES 2.0, after executing, shell prints egl information means GLES2.0 is executed successfully.

运行结果:

Execution result:



```
rk3399_all:/ # dr-g -gpu gles20
GPU test ..

 GLES2.0 test OK
```

图 3-27 gles 2.0 测试

Picture 3-27 gles 2.0 test

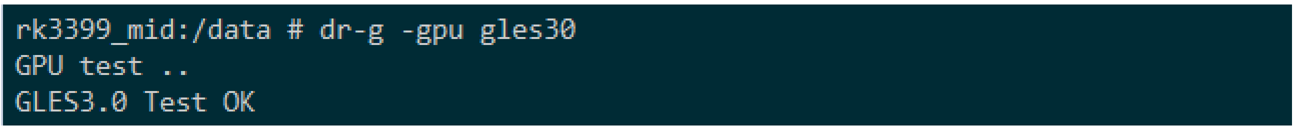
"dr-g -gpu gles30"

对 GLES 3.0 测试，运行后，shell 打印 egl 信息表示 GLES3.0 运行成功。

Test GLES 3.0, after executing, shell prints egl information means GLES3.0 is executed successfully.

运行结果:

Execution result:



```
rk3399_mid:/data # dr-g -gpu gles30
GPU test ..
GLES3.0 Test OK
```

图 3-27 gles 3.0 测试

Picture 3-27 gles 3.0 test

3.5.5 OpenCL 支持测试 OpenCL support test

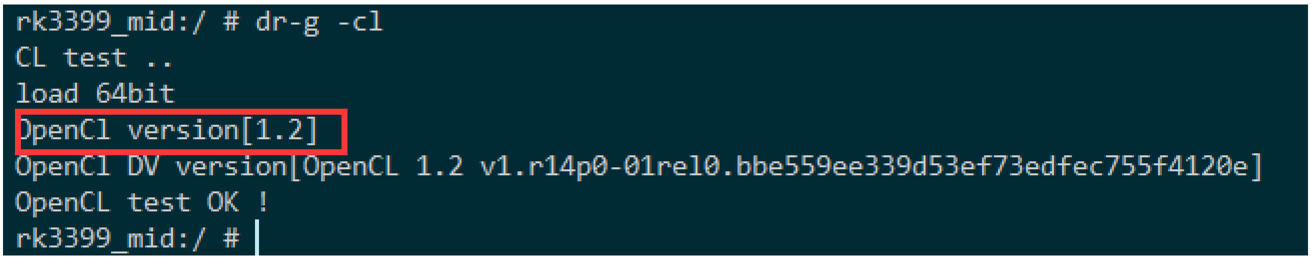
"dr-g –cl"

系统是否支持 openCL，运行后如果提示"OpenCL test OK !" 表示该平台支持 opencl。

Judge whether the system supports openCL. After executing, if it prompts "OpenCL test OK !", it means this platform supports opencl.

运行结果:

Execution result:



```
rk3399_mid:/ # dr-g -cl
CL test ..
load 64bit
OpenCl version[1.2]
OpenCl DV version[OpenCL 1.2 v1.r14p0-01rel0.bbe559ee339d53ef73edfec755f4120e]
OpenCL test OK !
rk3399_mid:/ #
```

图 3-28 OpenCL 支持测试

Picture 3-28 OpenCL support test

3.5.6 Vulkan 支持测试 Vulkan support test

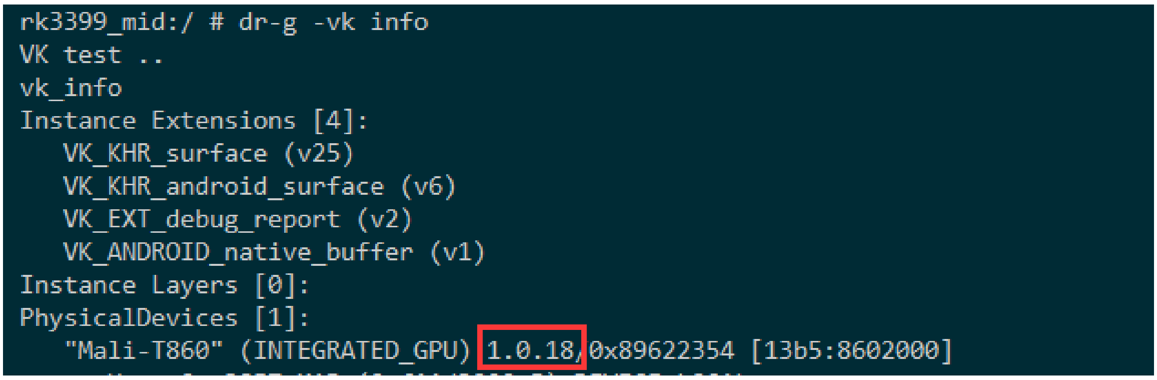
"dr-g -vk info"

打印 vulkan 版本信息，检测支持的 vk 扩展，检测支持的 vk 功能。

Print vulkan version information, detect the supported vk extension and vk functions.

运行结果:

Execution result:



```
rk3399_mid:/ # dr-g -vk info
VK test ..
vk_info
Instance Extensions [4]:
   VK_KHR_surface (v25)
   VK_KHR_android_surface (v6)
   VK_EXT_debug_report (v2)
   VK_ANDROID_native_buffer (v1)
Instance Layers [0]:
PhysicalDevices [1]:
   "Mali-T860" (INTEGRATED_GPU) 1.0.18/0x89622354 [13b5:8602000]
```

图 3-29 Vulkan 支持测试

Picture 3-29 Vulkan support test

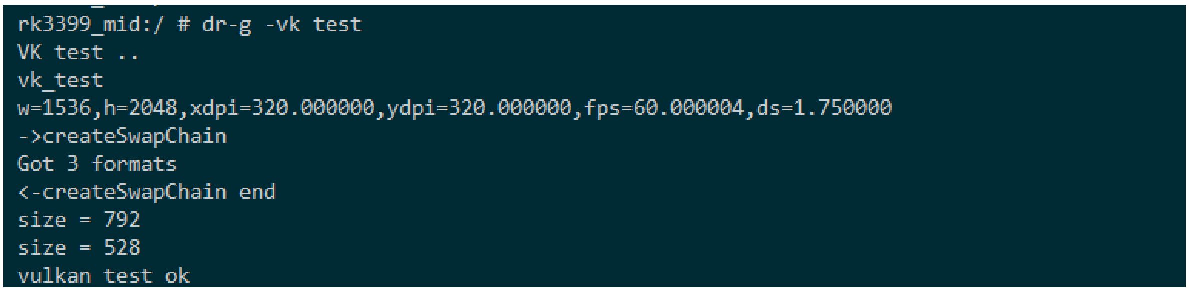
"dr-g -vk test"

vulkan sample，绘制一个简单的三角形。

vulkan sample, draw a simple triangle.

运行结果:

Execution result:



```
rk3399_mid:/ # dr-g -vk test
VK test ..
vk_test
w=1536,h=2048,xdpi=320.000000,ydpi=320.000000,fps=60.000004,ds=1.750000
->createSwapChain
Got 3 formats
<-createSwapChain end
size = 792
size = 528
vulkan test ok
```

图 3-30 Vulkan 测试

Picture 3-30 Vulkan test

运行后屏幕显示一个三角形，命令行打印 vulkan test ok。表示系统 vulkan 驱动环境是 OK 的。

After executing, the screen will display a triangle, with the command line printing vulkan test ok. It means the system vulkan driver environment is OK.

3.6 VOP 自动测试 VOP auto test

3.6.1 VOP 最大 ddr 负载测试 VOP max ddr load test

```
"dr-g  -vop  -c  chip-type"
```

该命令设置后，dr-g 会配置 vop，并增加系统 ddr 负载，用以测试在预设模式下 ddr 条件是否可以正确输出图像。 chip-type 可设置的参数有：

After setting this command, dr-g will configure vop, and add the system ddr load, to test whether it can output the image correctly with the prebuilt ddr condition.

"rk3399","rk3326","rk3368","rk3328","rk3288","rk3128","rk3188"。

运行结果:

Execution result:

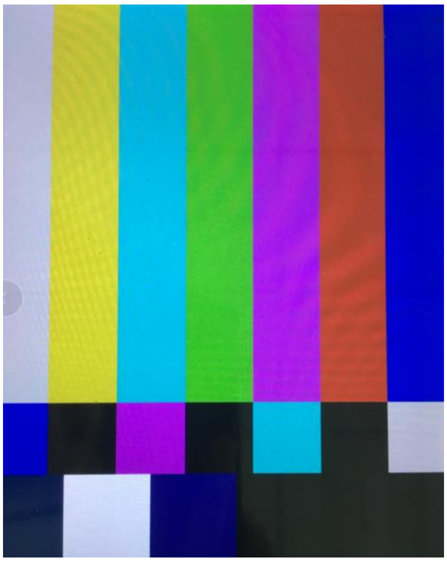


图 3-31 VOP 测试结果

Picture 3-31 VOP test result

3.6.2 VOP demo 测试 VOP demo test

```
"dr-g  -vop  -w  <win_id>@<crtc_id>:<src_left>,<src_top>,<src_w>,<src_h>:<dst_left>,<dst_top>,<dst_w>,<dst_h>[#zpos][@<format>]"
```

通过命令参数配置 vop 的对应 win_id 输出图像，提高调屏时的工作效率。

Use the command parameter to configure the corresponding win_id of vop to output the image, to improve the efficiency of panel debugging.

命令 demo： dr-g -vop -w 0@0:0,0,1536,2048:0,0,1536,2048#1@AR24 -f 1536x2048@AR24

Command demo: dr-g -vop -w 0@0:0,0,1536,2048:0,0,1536,2048#1@AR24 -f 1536x2048@AR24

运行结果:

Execution result:

源数据：

Source data:

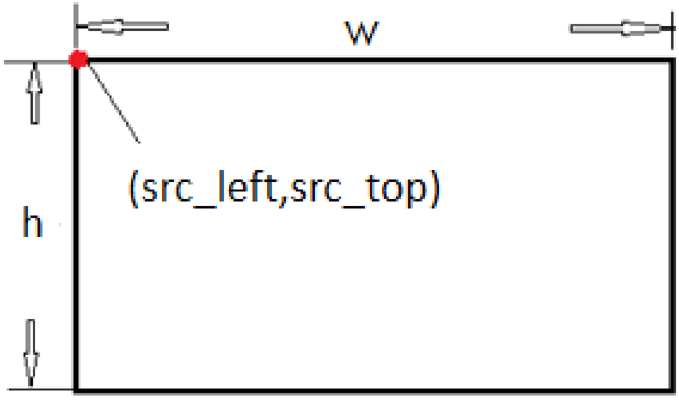


图 3-32 VOP 源数据

Picture 3-32 VOP source data

目标数据：

Target data:

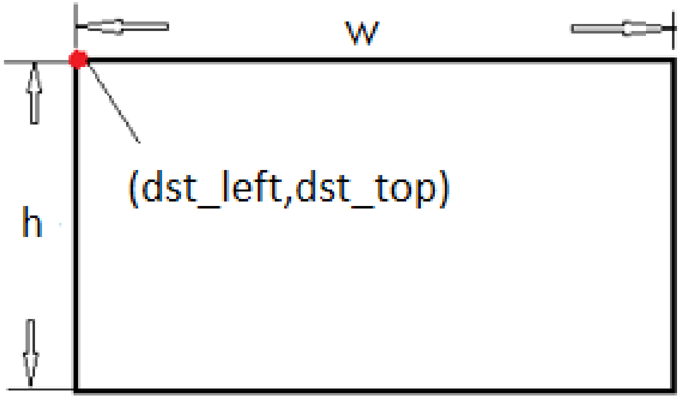


图 3-33 VOP 目标数据

Picture 3-33 VOP target data

3.7 LOG 自动检测功能 LOG auto detect function

3.7.1 当前 log 分析 Current log analysis

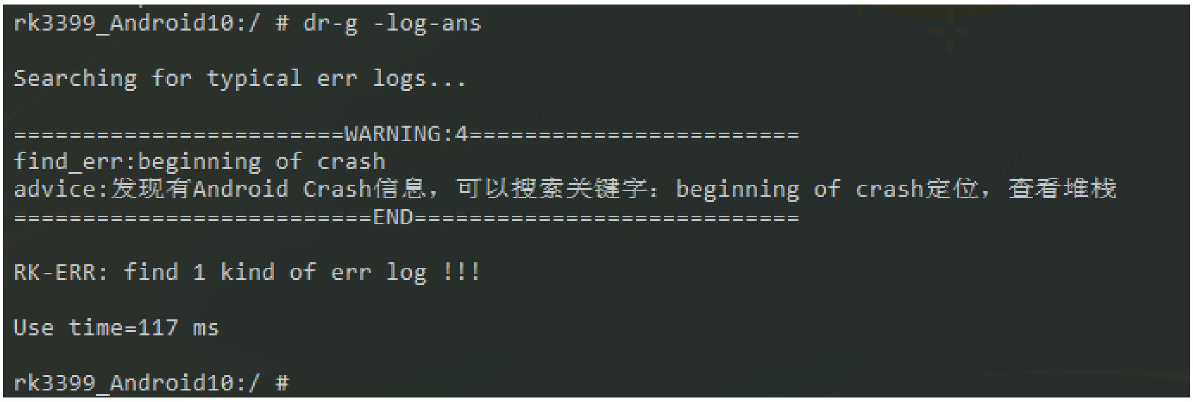
```
"dr-g  -log-ans"
```

对当前 logcat 存储的 log 与预设的报错日志规则进行匹配，若检测到对映 log，则报出警告，并给出相映的修改意见。前期收集与整理报错日志多花时间，在后期使用中能大大提高工作效率。用机器提前匹配一次 log，并打印出某行，直接定位，无需人工在复杂的 log 中定位报错。

Compare the log saved by current logcat with the prebuilt error log rule, if detecting the corresponding log, it will prompt alarm and give the corresponding modification comment. Take more time to collect and summary the error log in advance and it will greatly improve the working efficiency in the future. Use the device to match log once in advance, and print out some line to directly locate, no need to manually locate the error within the complex log.

运行结果：

Execution result:



```
rk3399_Android10:/ # dr-g -log-ans

Searching for typical err logs...

========================WARNING:4========================
find_err:beginning of crash
advice:发现有Android Crash信息，可以搜索关键字：beginning of crash定位，查看堆栈
==========================END============================

RK-ERR: find 1 kind of err log !!!

Use time=117 ms

rk3399_Android10:/ #
```

图 3-34 log 分析结果

Picture 3-34 log analysis result

3.7.2 XML 规则文件 XML rule file

log_id：规则 log 的 id 号

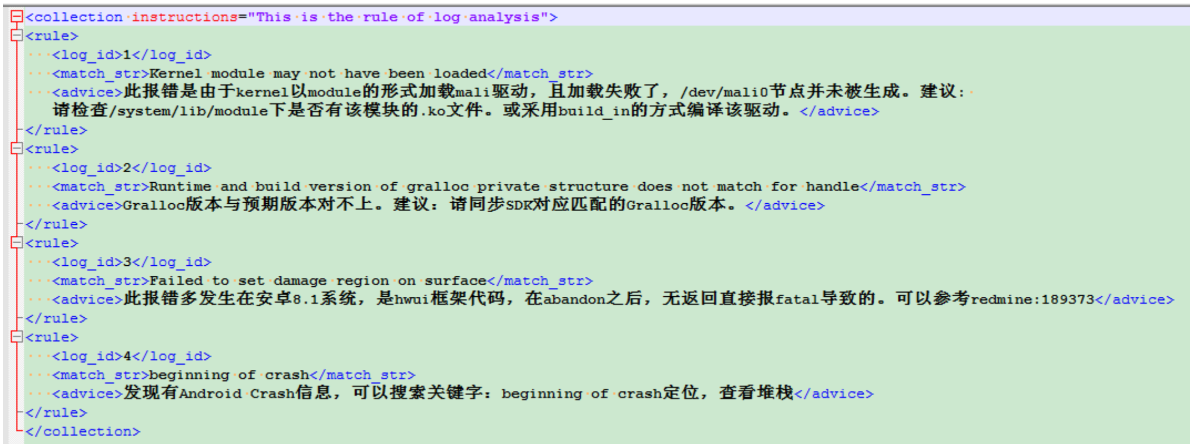
log_id: the id number of log in the rule

match_str：规则匹配字符串

match_str: matched string in the rule

advice：给出相应的建议

advice: the corresponding suggestion provided



```
<collection instructions="This is the rule of log analysis">
<rule>
    <log_id>1</log_id>
    <match_str>Kernel module may not have been loaded</match_str>
    <advice>此报错是由于kernel以module的形式加载mali驱动，且加载失败了，/dev/mali0节点并未被生成。建议：
    请检查/system/lib/module下是否有该模块的.ko文件。或采用build_in的方式编译该驱动。</advice>
</rule>
<rule>
    <log_id>2</log_id>
    <match_str>Runtime and build version of gralloc private structure does not match for handle</match_str>
    <advice>Gralloc版本与预期版本对不上。建议：请同步SDK对应匹配的Gralloc版本。</advice>
</rule>
<rule>
    <log_id>3</log_id>
    <match_str>Failed to set damage region on surface</match_str>
    <advice>此报错多发生在安卓8.1系统，是hwui框架代码，在abandon之后，无返回直接报fatal导致的。可以参考redmine:189373</advice>
</rule>
<rule>
    <log_id>4</log_id>
    <match_str>beginning of crash</match_str>
    <advice>发现有Android Crash信息，可以搜索关键字：beginning of crash定位，查看堆栈</advice>
</rule>
</collection>
```

图 3-35 xml 规则

Picture 3-35 xml rule