

Rockchip Linux 系统测试操作指南

文档标识: RK-SM-YF-352

发布版本: V2.0.2

日期: 2024-5-16

文件密级: ☐绝密 ☐秘密 ☐内部资料 ☒公开

免责声明

本文档按“现状”提供, 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。

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。

版权所有 © 2024 瑞芯微电子股份有限公司

超越合理使用范畴, 非经本公司书面许可, 任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部, 并不得以任何形式传播。

瑞芯微电子股份有限公司

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

前言

概述

本文档主要介绍Rockchip Linux SDK系统测试方法，旨在指导和帮助客户快速了解并上手Rockchip Linux产品量产封版前必要的软硬件稳定性测试。

读者对象

本文档（本指南）主要适用于以下工程师：

测试工程师

软件开发工程师

硬件开发工程师

修订记录

日期	版本	作者	修改说明
2017-01-15	V1.0.0	CQ	初始版本
2020-03-24	V1.1.0	CQ	更新测试项
2020-03-30	V1.1.1	Caesar Wang	修改格式
2020-08-02	V1.1.2	Ruby Zhang	格式修正
2020-12-18	V1.1.3	Ruby Zhang	语句修正
2022-12-01	V2.0.0	Nickey Yang	增加软硬件稳定性测试说明，修订部分描述和图片
2022-12-05	V2.0.1	Nickey Yang	修订部分描述
2024-5-16	V2.0.2	Ruby Zhang	修订部分描述

目录

Rockchip Linux 系统测试操作指南

1. 功能测试
 - 1.1 Buildroot
 - 1.1.1 录像
 - 1.1.2 录音
 - 1.1.3 Wi-Fi 连接
 - 1.1.4 音频播放
 - 1.1.5 系统时间
 - 1.1.6 RTC 时钟
 - 1.1.7 视频播放
 - 1.1.8 SD 卡升级、启动
 - 1.1.9 查看内存
 - 1.1.10 查看磁盘空间使用情况
 - 1.2 Debian
 - 1.2.1 终端位置
 - 1.2.2 Wi-Fi 连接
 - 1.2.3 双屏异显
 - 1.2.4 双屏异声
 - 1.2.5 显示屏旋转
2. 性能测试
 - 2.1 磁盘读写测试
 - 2.1.1 eMMC读写
 - 2.1.2 U盘读写
 - 2.2 设置性能模式
 - 2.3 查看CPU DDR频率
 - 2.4 glmark2 跑分
3. 硬件稳定性测试
 - 3.1 IO-DOMAINS
 - 3.2 电源
 - 3.3 信号
 - 3.4 高低温
4. 软件稳定性测试
 - 4.1 基础工具
 - 4.1.1 stressapptest
 - 4.1.1.1 Overview
 - 4.1.1.2 Usage
 - 4.1.2 memtester
 - 4.1.2.1 Overview
 - 4.1.2.2 Usage
 - 4.2 Rockchip test
 - 4.2.1 配置开启
 - 4.2.2 测试列表
 - 4.2.3 测试规模
 - 4.2.4 测试方法
 - 4.2.4.1 rockchip_test
 - 4.2.4.2 直接执行
 - 4.2.5 测试详请
 - 4.2.5.1 DDR测试
 - 4.2.5.1.1 ddr_freq_scaling
 - 4.2.5.1.2 stressapptest_test
 - 4.2.5.1.3 memtester_test
 - 4.2.5.2 CPU测试
 - 4.2.5.2.1 cpu_freq_scaling

- 4.2.5.3 Flash测试
- 4.2.5.4 待机唤醒测试
- 4.2.5.5 reboot测试
- 4.2.5.6 GPU测试
- 4.2.5.7 Camera测试
- 4.2.5.8 Video测试
- 4.2.5.9 Wi-Fi Bt测试

4.3 用户场景测试

5. TO DO

- 5.1 增加常见问题分析方法
- 5.2 添加npu, recovery, audio, 浏览器, 掉电测试说明

1. 功能测试

1.1 Buildroot

1.1.1 录像

录像：

```
rkisp_demo --device=/dev/video1 --output=/tmp/isp.yuv --  
iqfile=/etc/iqfiles/OV5695.xml
```

播放录像：

将/tmp/cif.yuv下的文件pull到电脑端：`adb pull /tmp/cif.yuv /tmp/cif.yuv`，通过 YUV 工具播放。

1.1.2 录音

```
arecord -c 通道 -r 采样频率 -f 采样位数 -d 录音时长 /录音存放路径/录音文件名
```

通道 `ch_tbl="2 4 6 8"`

采样频率 `fs_tbl="8000 11025 16000 22050 32000 44100 48000 64000 88200 96000 176400 192000"`

采样位数 `bits_tbl="S16_LE S24_LE S32_LE"`

封装格式 `wmv、wav、mp3` 等

例：

限时录音-录音10秒自动退出并保存：

```
arecord -c 2 -r 44100 -f S16_LE -d 10 /tmp/record.wav
```

不限时录音-ctrl+c退出即可保存：

```
arecord -c 2 -r 44100 -f S16_LE /tmp/record.wav
```

播放录音文件：

```
aplay /tmp/record.wav
```

1.1.3 Wi-Fi 连接

方法一：`wifi_start.sh`脚本

如果sdk集成了wifi_start.sh脚本，可以直接在脚本后面跟 `ssid`和`password`

```
wifi_start.sh WiFi-AP password
```

方法二：修改配置文件

```
$ vi /data/cfg/wpa_supplicant.conf
ctrl_interface=/var/run/wpa_supplicant  #默认不建议修改!
ap_scan=1
update_config=1
#这个配置使wpa_cli命令配置的热点保存到conf文件里面 (wpa_cli save_config)

#加密Wi-Fi
network={
    ssid="WiFi-AP"    # Wi-Fi名字
    psk="password"    # Wi-Fi密码
    key_mgmt=WPA-PSK  # 认证方法
}

#不加密Wi-Fi:
network={
    ssid="WiFi-AP"    # Wi-Fi名字
    key_mgmt=NONE     # 认证方法
}

#让wpa_supplicant进程重新读取上述配置:
wpa_cli -i wlan0 -p /var/run/wpa_supplicant reconfigure
#发起连接:
wpa_cli -i wlan0 -p /var/run/wpa_supplicant reconnect
```

方法三：wpa_cli工具

```
#加密Wi-Fi:
wpa_cli -i wlan0 -p /var/run/wpa_supplicant remove_network 0
wpa_cli -i wlan0 -p /var/run/wpa_supplicant ap_scan 1
wpa_cli -i wlan0 -p /var/run/wpa_supplicant add_network
wpa_cli -i wlan0 -p /var/run/wpa_supplicant set_network 0 ssid "WiFi-AP"
wpa_cli -i wlan0 -p /var/run/wpa_supplicant set_network 0 key_mgmt WPA-PSK
wpa_cli -i wlan0 -p /var/run/wpa_supplicant set_network 0 psk "password"
wpa_cli -i wlan0 -p /var/run/wpa_supplicant select_network 0
wpa_cli -i wlan0 -p /var/run/wpa_supplicant save_config #保存上述配置到conf文件

#不加密Wi-Fi:
wpa_cli -i wlan0 -p /var/run/wpa_supplicant remove_network 0
wpa_cli -i wlan0 -p /var/run/wpa_supplicant ap_scan 1
wpa_cli -i wlan0 -p /var/run/wpa_supplicant add_network
wpa_cli -i wlan0 -p /var/run/wpa_supplicant set_network 0 ssid "WiFi-AP"
wpa_cli -i wlan0 -p /var/run/wpa_supplicant set_network 0 key_mgmt NONE
wpa_cli -i wlan0 -p /var/run/wpa_supplicant select_network 0
wpa_cli -i wlan0 -p /var/run/wpa_supplicant save_config
```

查看连接状态:

```
wpa_cli -i wlan0 -p /var/run/wpa_supplicant status
```

详情可以查看 [Rockchip_Developer_Guide_Linux_WIFI_BT_CN.pdf](#)

1.1.4 音频播放

```
aplay /media/usb0/musicdemo.wmv
```

1.1.5 系统时间

```
date                                #查看系统时间
date --set='2018-12-24 15:17:42'  #设置系统时间
hwclock --show                     #查看硬件时间
hwclock --systohc                 #硬件时间同步显示系统时间
```

1.1.6 RTC 时钟

查看当前状态下或重启后时间是否有变化。

```
cat /sys/class/rtc/rtc0/time
```

1.1.7 视频播放

单窗口视频播放:

```
gst-play-1.0 /oem/SampleVideo_1280x720_5mb.mp4
```

多窗口视频播放:

```
sh /rockchip-test/video/test_gst_multivideo.sh
```

1.1.8 SD 卡升级、启动

- SD 卡插入 PC 端，在 PC 端执行SD_Firmware_Tool.exe，选择固件升级/SD启动，选择固件-update.img，开始创建。
- 将 SDK 进入maskrom擦除flash后，断电。
- 插入制作好的 SD 卡，将 SDK 上电开机，会自动烧写固件。



1.1.9 查看内存

```
cat /proc/meminfo 或 free -h
```

1.1.10 查看磁盘空间使用情况

```
df -h
```

1.2 Debian

1.2.1 终端位置

开始-> System Tools -> LXTerminal

1.2.2 Wi-Fi 连接

在串口输入如下命令：


```
nmcli r wifi on #第1步: 开启Wi-Fi
nmcli dev wifi #第2步: 扫描附近AP
#第3步: 连接AP
nmcli dev wifi connect "DIR-803" password "839919060" ifname wlan0
nmcli r wifi off #第4步: 关闭Wi-Fi
```

1.2.3 双屏异显

使用 xrandr 来确定有几个显示设备，比如下面检测到系统当前连接着 HDMI-1 和 DSI-1 两个设备：

```
root@linaro-alip:/# xrandr
Screen 0: minimum 320 x 200, current 1920 x 1920, maximum 16384 x 16384
HDMI-1 connected primary 1920x1080+0+0 (normal left inverted right x axis y axis) 0mm x 0mm
    1920x1080    60.00 + 60.00* 50.00 30.00 24.00
    1920x1080i    60.00 50.00 50.00
    1280x1024    60.02
    1280x720     60.00 50.00 50.00
    720x576      50.00 50.00
    720x576i     50.00
    720x480      59.94 59.94
    720x480i     59.94
HDMI-2 disconnected (normal left inverted right x axis y axis)
DSI-1 connected 1080x1920+0+0 (normal left inverted right x axis y axis) 0mm x 0mm
    1080x1920    59.90*+
DP-1 disconnected (normal left inverted right x axis y axis)
DP-2 disconnected (normal left inverted right x axis y axis)
```

xrandr 来设置两个屏幕的关系：

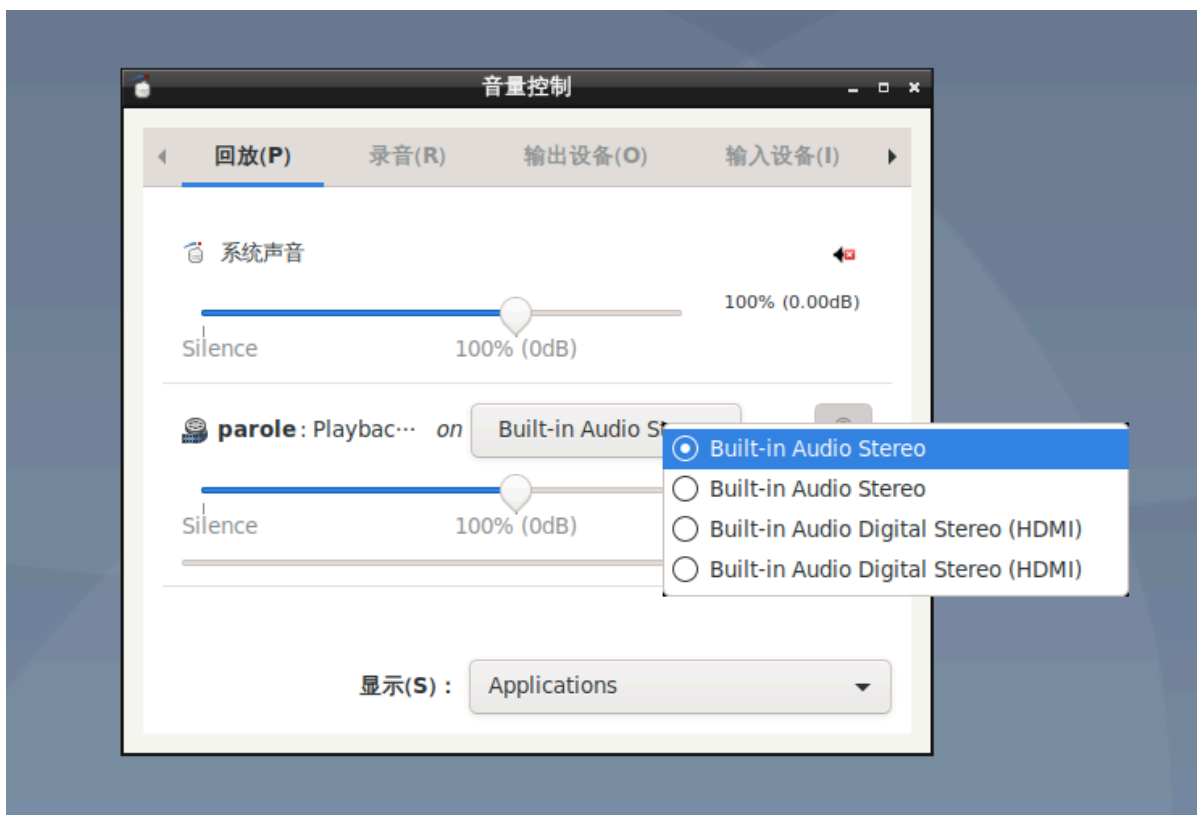
```
su linaro-c "DISPLAY=:0 xrandr--output HDMI-1--above DSI-1"
```

其中 --above 可以代换成 right-of, left-of, below, same-as, preferred, off 等。

更多详情可查阅 [Rockchip_Developer_Guide_Linux_Graphics_CN.pdf](#)

1.2.4 双屏异声

方法一：打开左下角的 Sound&Video -> PulseAudio Volume Control -> Playback



方法二：使用 aplay 来确认声卡和选择声卡播放：

```
root@linaro-alip:/# aplay -l
**** List of PLAYBACK Hardware Devices ****
card 0: rockchipdp0 [rockchip,dp0], device 0: rockchip,dp0 spdif-hifi-0
[rockchip,dp0 spdif-hifi-0]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 1: rockchipes8388 [rockchip-es8388], device 0: dailink-multicodecs ES8323.7-
0011-0 [dailink-multicodecs ES8323.7-0011-0]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 2: rockchiphdmi0 [rockchip-hdmi0], device 0: rockchip-hdmi0 i2s-hifi-0
[rockchip-hdmi0 i2s-hifi-0]
  Subdevices: 1/1
  Subdevice #0: subdevice #0

dp0: aplay-Dplughw:0,0/dev/urandom
es8388: aplay-Dplughw:1,0/dev/urandom
hdmi0: aplay-Dplughw:2,0/dev/urandom
```

方法三：打开一个音乐歌曲从主屏拖到副屏，然后在主屏中同样方式选择一个声卡来播放，完成双屏异声功能。

更多详情可查阅 [Rockchip_Developer_Guide_Linux_Graphics_CN.pdf](#)

1.2.5 显示屏旋转

旋转 normal/left/right:

```
vi /etc/X11/xorg.conf.d/20-modesetting.conf
```

把normal改为left/right/, reboot后生效。

2. 性能测试

2.1 磁盘读写测试

测试前先查一下节点：`fdisk -l`

```
root@linaro-alip:~# fdisk -l
Disk /dev/mmcblk0: 29.1 GiB, 31268536320 bytes, 61071360 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: 1124E418-9008-41F8-9E3D-8D872216C8A1
```

Device	Start	End	Sectors	Size	Type
/dev/mmcblk0p1	16384	24575	8192	4M	unknown
/dev/mmcblk0p2	24576	32767	8192	4M	unknown
/dev/mmcblk0p3	32768	163839	131072	64M	unknown
/dev/mmcblk0p4	163840	425983	262144	128M	unknown
/dev/mmcblk0p5	425984	491519	65536	32M	unknown
/dev/mmcblk0p6	491520	29851647	29360128	14G	unknown
/dev/mmcblk0p7	29851648	30113791	262144	128M	unknown
/dev/mmcblk0p8	30507008	61071295	30564288	14.6G	unknown

查看分区可读写的是mmcblk0p8，这个分区容量是14.6G，一般我们会用最后这个分区作为可读写的userdata分区，所以可拿来作为eMMC读写性能测试。

2.1.1 eMMC读写

写：

```
dd if=/dev/zero of=/dev/mmcblk0p8 bs=1M count=2000 oflag=direct,nonblock
```

读：

```
dd if=/dev/mmcblk0p8 of=/dev/null bs=1M count=2000 iflag=direct,nonblock
```

2.1.2 U盘读写

写：

```
dd if=/dev/zero of=/dev/sda1 bs=1M count=2000 oflag=direct,nonblock
```

读：

```
dd if=/dev/sda1 of=/dev/null bs=1M count=2000 iflag=direct,nonblock
```

2.2 设置性能模式

```
echo performance | tee $(find /sys/ -name *governor)
```

2.3 查看CPU DDR频率

```
cat /sys/kernel/debug/clk/clk_summary |grep -E "cpu|arm|ddr"
```

2.4 glmark2 跑分

显示屏幕跑分：

```
sh /rockchip-test/gpu/test_fullscreen_glmark2.sh
```

屏幕不显示跑分：

```
sh /rockchip-test/gpu/test_offscreen_glmark2.sh
```

3. 硬件稳定性测试

3.1 IO-DOMAINS

要求所有GPIO电源域的硬件供电电压与软件配置相匹配，所以请在量产前再次重新审核项目电压和电源域配置的正确性，特别是Wi-Fi，FLASH，以太网这几路IO电源的配置。否则会产生以下不良：

- 软件配置为1.8V，硬件供电3.3V，会使得低耐压电路暴露在高压环境下，长期工作会损坏电路
- 软件配置为3.3V，硬件供电1.8V，电路工作会异常

配置检查的方法可参考对应芯片平台下的文档，如RK3568可参考：

Rockchip_RK356X_Introduction_IO_Power_Domains_Configuration_CN.pdf

3.2 电源

建议项目硬件的电源设计使用我们的参考设计，如果电源方案 and 我们的参考设计有改动，或者器件选型不一样，特别是当采用分立电源而不是搭载我们匹配的PMIC电源方案时，需要对涉及芯片启动的各路核心电源如：VDD_LOGIC，VDD_ARM，VCC_PMU，VCC_DDR供电等进行必要的测试，包括冷启动、重启时的上电时序，电压、高负载下电源的纹波等。

3.3 信号

项目硬件图纸在完成设计后，建议通过redmine系统提交我们的工程师帮忙review，特别是DDR部分需要严格按照我们的模板完成设计。同时DDR、EMMC、FLASH颗粒的选型需要在我们的AVL中，AVL列表可以通过redmine获取。

3.4 高低温

除了常温下的稳定性测试，建议客户在实际产品工作温度的高、低温环境下完成CPU压力，DDR压力，reboot压力以及冷启动的测试，保证产品在高低温工作环境下的稳定性。

4. 软件稳定性测试

本章节主要介绍rockchip test包使用的基础工具，以及如何使用rockchip test包完成产品的软件稳定性测试。

4.1 基础工具

4.1.1 stressapptest

4.1.1.1 Overview

stressapptest 是Stressful Application Test的简称。

stressapptest会生产线程来拷贝和直接对磁盘进行读取和写入。目标则是尽可能的加大系统内存负载，以便更加有效地进行测试。并且让来自处理器和I/O到内存的数据尽量随机化，以创造出模拟现实的环境来测试现在的硬件设备是否稳定，Google就在使用它，现在是Apache 2.0许可。stressapptest 有以下几种用途：

- 压力测试
- 硬件检查和调试
- 内存接口测试：参考后面章节的内容。
- 磁盘测试。

4.1.1.2 Usage

典型的执行命令：

```
stressapptest -s 20 -M 256 -m 8 -W      # Test 256MB, running 8 "warm copy"
threads. Exit after 20 seconds.
stressapptest --help                    # list the available arguments.
```

常用参数：

- -M MB：要测试的 RAM 大小（自动检测所有可用内存）
- -s 秒：运行的秒数 (20)
- -m 线程：要运行的内存复制线程数（自动检测 CPU 数）
- -W：使用更多 CPU 压力内存复制（错误的）
- -n ipaddr：添加一个连接到系统“ipaddr”的网络线程。（无）
- --listen：运行一个线程来侦听并响应网络线程。（0）
- -f filename：添加带有临时文件“文件名”的磁盘线程（无）
- -F：不要检查每个事务的结果，而是使用 libc memcpy。（错误的）

错误处理

- -l logfile：输出log到文件“logfile”（无）
- -v level：详细程度 (0-20)（默认值：8）

4.1.2 memtester

4.1.2.1 Overview

一款在用户空间下，用于测试内存正确性的实用工具，适用于大多数平台。可以在32位或64位的类Unix操作系统编译和运行(当然也包括Unices和MacOs)。这款工具主要面向嵌入式开发人员，从4.1.0版本开始，memtester可以指定起始物理内存地址进行测试。

4.1.2.2 Usage

```
memtester [-p physaddrbase [-d device]] <mem>[B|K|M|G] [loops]
```

loop中具体的测试项：

```
Stuck Address
Random Value
Compare XOR
Compare SUB
Compare MUL
Compare DIV
Compare OR
Compare AND
Sequential Increment
Solid Bits
Block Sequential
Checkerboard
Bit Spread
Bit Flip
Walking Ones
Walking Zeroes
8-bit Writes
16-bit Writes
```

4.2 Rockchip test

- Rockchip test压力测试的日志输出路径为 /userdata/rockchip-test，所以测试前请保证该目录具有可读写的权限
- 完整的串口log有助于更好的分析和定位问题。所以测试时产品需要接出串口，且串口上位机开启保存所有log的功能

4.2.1 配置开启

在buildtoot中使能rockchip-test包需要开启如下配置：

```
# - stress test on cpu and memory
BR2_PACKAGE_STRESSAPPTST=y
# - memtest used to testing the memory subsystem for faults
BR2_PACKAGE_MEMTESTER=y
# - audio, reboot, bluetooth, ddr, dvfs, flash, suspend/resume, wifi
BR2_PACKAGE_ROCKCHIP_TEST=y
```

4.2.2 测试列表

目前已支持的压力测试列表如下：

```
# bash /rockchip-test/rockchip_test.sh
"*****"
"***"
"***"
"***"
"***"
"***"
"***"
"*****"
"*****"
"ddr test:          1 (ddr stress test)"
"cpu test:         2 (cpu stress test)"
"gpu test:         3 (gpu stress test)"
"npu test:         4 (npu stress test)"
"suspend_resume test: 5 (suspend resume)"
"reboot test:      6 (auto reboot test)"
"power lost test:  7 (power lost test)"
"flash stress test: 8 (flash stress test)"
"recovery test:    9 (recovery wipe all test)"
"audio test:      10 (audio test)"
"camera test:     11 (camera test)"
"video test:      12 (video test)"
"bluetooth test:  13 (bluetooth on off test)"
"wifi test:       14 (wifi on off test)"
"chromium test:   15 (chromium with video test)"
"*****"
```

4.2.3 测试规模

我们建议每个测试项目部署 **10台 * 24小时** 以上，可以根据项目对于稳定性要求的程度进行适当调整。

4.2.4 测试方法

下面以DDR压力测试中的 `memtester` 为例，介绍一下各单元测试的方法。单元测试支持从 `rockchip_test` 脚本执行，也可以直接执行对应的脚本。

4.2.4.1 rockchip_test

```
bash rockchip-test/rockchip_test.sh
```

- 输入压力测试列表内 `ddr test` 对应的序号 1

```
"*****"
"***                                     ***"
"***          DDR TEST                  ***"
"***                                     ***"
"*****"
"*****"
"memtester:                             1"
"stressapptest:                          2"
"stressapptest + memtester:              3"
"ddr auto scaling:                       4"
"stressapptest + memtester + ddr auto scaling: 5"
"*****"
```

- 再输入测试项目 `memtester` 对应的序号 1，即可开始测试。

4.2.4.2 直接执行

```
bash /rockchip-test/ddr/memtester_test.sh
```

4.2.5 测试详请

4.2.5.1 DDR测试

DDR压力测试支持以下五种测试场景：

```
"memtester:                             1"
"stressapptest:                          2"
"stressapptest + memtester:              3"
"ddr auto scaling:                       4"
"stressapptest + memtester + ddr auto scaling: 5"
```


- 产品不支持变频，可以选择 3，即使用stressapptest + memtester的场景进行测试
- 产品支持变频，可以选择 5，即使用stressapptest + memtester + ddr自动变频的场景进行测试

CPU压力测试由入口脚本 `ddr_test.sh` 和以下3个脚本组成。

```
/rockchip-test/ddr/ddr_freq_scaling.sh
/rockchip-test/ddr/memtester_test.sh
/rockchip-test/ddr/stressapptest_test.sh
```

4.2.5.1.1 ddr_freq_scaling

DDR变频脚本

- 脚本不带参数的时候，会执行自动变频

```
bash /rockchip-test/ddr/ddr_freq_scaling.sh
cnt: 0, ddr freq: success change to 528000000 Hz
..
cnt:100, ddr freq: success change to 324000000 Hz
```

- 脚本带频率点的参数，且频率点在支持频率中，会变频到对应的频率点

```
bash /rockchip-test/ddr/ddr_freq_scaling.sh 528000000
ddr freq: success change to 528000000 Hz
```

- 脚本带频率点的参数，但频率点不在支持频率中，会打印支持的频率点，以及当前频率

```
bash /rockchip-test/ddr/ddr_freq_scaling.sh 1
ddr freq: 1 is not in available frequencies: 324000000 528000000
ddr freq: now 528000000 Hz
```

- 关闭测试

```
killall bash
```

4.2.5.1.2 stressapptest_test

stressapptest脚本，默认使用系统可用内存的一半容量测试，24小时后自动停止

- 关闭测试

```
killall stressapptest
```

4.2.5.1.3 memtester_test

memtester脚本，默认使用系统可用内存的一半容量测试。

- 关闭测试

```
killall memtester
```

- 注：memtester 出错不会自动停止测试，需要观测程序运行过程的打印

4.2.5.2 CPU测试

CPU压力测试支持以下三种测试场景：

```
"stressapptest test: 1"
"cpu auto scaling: 2"
"stressapptest + cpu auto scaling: 3"
```

- 产品不支持变频，可以选择 2，即使用stressapptest的场景进行测试
- 产品支持变频，可以选择 3，即使用stressapptest + cpu自动变频的场景进行测试

CPU压力测试由入口脚本 `cpu_test.sh` 和 `cpu_freq_scaling.sh` 组成。

4.2.5.2.1 cpu_freq_scaling

- 脚本不带参数的时候，会执行自动变频

```
bash /rockchip-test/cpu/cpu_freq_scaling.sh
cnt: 0, cpu freq policy:0 success change to 816000 KHz
...
cnt: 100, cpu freq policy:0 success change to 1104000 KHz
```

- 脚本带频率点的参数，且频率点在支持频率中，会变频到对应的频率点

```
bash /rockchip-test/cpu/cpu_freq_scaling.sh 1104000
cpu freq policy:0 success change to 1104000 KHz
```

- 脚本带频率点的参数，但频率点不在支持频率中，会打印支持的频率点，以及当前频率

```
bash /rockchip-test/cpu/cpu_freq_scaling.sh 1
cpu freq: 1 is not in available frequencies: 600000 816000 1008000 1104000
1416000
cpu freq: now 1104000 Hz
```

- 关闭测试

```
killall bash
```

4.2.5.3 Flash测试

- 在 `$test_dir/src_test_data` 生成7个5MB以内的随机源数据，拷贝到 `$test_dir/des_test_data` 的五个子目录下，并对比md5值，循环200次

4.2.5.4 待机唤醒测试

- 使能自动待机唤醒需要系统存在RTC，且正常工作
- 默认会执行执行待机唤醒10000次后退出，每次待机唤醒的时间是3-6秒之间的随机数

4.2.5.5 reboot测试

- 默认重启10000次后停止测试，每次启动后8秒自动关机
- 提前关闭测试

```
echo off > /userdata/rockchip_test/reboot_cnt
```

- 如果系统使能CONFIG_PSTORE_RAM，脚本会在每次启动后检查上次关机过程中的log是否存在奔溃等异常，异常则会自动停止该测试

4.2.5.6 GPU测试

- GPU压力测试是基于 glmark-es2 的，选择的 `test_stress_glmark2.sh` 脚本，主要是以 `run-forever` 参数来执行 `glmark-es2`
- `glmark-es2` is a benchmark for OpenGL ES 2.0. It only uses the subset of the OpenGL 2.0 API that is compatible with OpenGL ES 2.0

4.2.5.7 Camera测试

- Camera压力测试可以通过以下两种方法进行：

1. `camera_stresstest_rkisp_demo.sh`

需保证buildtoot已使能如下配置：

```
BR2_PACKAGE_CAMERA_ENGINE_RKAIQ_RKISP_DEMO=y
```

使用`rkisp_demo`无需通过 `media-ctl` 配置isp相关的pipeline，这种方法不会对usb camera 进行压力测试

该测试流程是循环抓取 ISP CIF 接口的camera数据各100帧，但不校验文件大小

2. `camera_stresstest_v4l2.sh`

需保证buildtoot已使能如下配置：

```
BR2_PACKAGE_LIBV4L_UTILS=y
```

使用 `v4l2-ctl` 可能需要通过 `media-ctl` 配置isp相关的pipeline，这种方法会对usb camera 进行压力测试

该测试流程是以640x480的分辨率循环抓取 USB ISP CIF 接口的camera数据各5帧，并校验文件大小

- 根据实际应用场景选择以上任一方法即可

4.2.5.8 Video测试

- Video压力测试是循环播放 `/mnt/udisk/videos` 或 `/userdata/videos` 目录下的视频文件
- 测试前把待测试的视频文件放在上述任意目录下，优先查找的是 `/userdata/videos` 目录，如果上述两个目录都不存在会提示并退出该测试

4.2.5.9 Wi-Fi Bt测试

- Wi-Fi Bt的测试为自动开关测试

4.3 用户场景测试

除了上述针对系统功能的单元测试，我们还可以根据实际产品功能去模拟这种真实场景进行压力测试，作为对系统稳定性测试的一种补充。

比如针对词典笔的产品形态，典型的用户使用场景：

开机 -> 扫描识别 -> 播放翻译结果 -> 放置直到系统进入二级待机 -> 唤醒后再次使用，同时可能随机性的伴随有Wi-Fi，BT的开关等

我们可以使用上面基础脚本中的命令，去构建出这种真实场景：

开机 -> loop (v4l2采图 -> aplay播放 -> pm-suspend -> rtc 唤醒) ，同时在唤醒的时间段内，随机去执行Wi-Fi，Bt的开关等

5. TO DO

5.1 增加常见问题分析方法

5.2 添加npu，recovery，audio，浏览器，掉电测试说明