

深圳易新泰科技有限公司 easitek



深圳易新泰科技有限公司 easitek

深圳易新泰科技有限公司 easitek

深圳易新泰科技有限公司 easitek

深圳易新泰科技

深圳易新泰科技有限公司 easitek

深圳易新泰科技有限公司 easitek

Dragonboard 使用指南

深圳易新泰科技有限公司 easitek

深圳易新泰科技有限公司 easitek

深圳易新泰科技

深圳易新泰科技有限公司 easitek

深圳易新泰科技有限公司 easitek

深圳易新泰科技有限公司 easitek

深圳易新泰科技有限公司 easitek

深圳易新泰科技

深圳易新泰科技有限公司 easitek

深圳易新泰科技有限公司 easitek

版本号: 1.0
发布日期: 2019.12.31

深圳易新泰科技有限公司 easitek

深圳易新泰科技有限公司 easitek

深圳易新泰科技

版本历史

版本号	日期	制/修订人	内容描述
1.0	2019.12.31	AWA1637	version 1.0



目 录

1 概述	1
1.1 编写目的	1
1.2 适用范围	1
1.3 适用人员	1
2 系统介绍	2
2.1 系统简介	2
2.2 源码结构介绍	2
2.3 系统配置以及编译	3
2.3.1 配置 dragonboard	3
2.3.2 编译 dragonboard	5
2.3.3 打包 dragonboard	5
2.3.4 烧写 drangonoard	6
2.3.5 卡启动注意事项	6
2.3.5.1 NAND 介质机器	6
2.3.5.2 eMMC 介质机器	6
3 测试流程	7
3.1 SD 卡测试	7
3.2 按键测试	7
3.3 WIFI 测试	8
3.4 蓝牙测试	8
3.5 触摸屏测试	8
3.6 摄像头测试	8
3.7 音频测试	8
3.8 USB 测试	9
3.9 重力传感器	9
3.10 U 盘	9
3.11 HDMI	9
4 配置文件	10
4.1 模块配置实例	10
4.2 DRAM 配置	11
4.3 RTC 配置	11
4.4 WIFI 配置	11
4.5 NAND 闪存配置	12
4.6 EMMC 闪存配置	12
4.7 蓝牙配置	13
4.8 MMC 配置	13
4.9 Udisk 配置	13
4.10 KEY 配置	14
4.11 TP 配置	14

4.12 摄像头测试	14
4.13 音频测试	15
4.14 重力传感器测试	15
4.15 U 盘测试	16
4.16 HDMI 测试	16
4.17 界面配置	16
5 FAQs	19
5.1 如何添加新的方案?	19
5.2 为什么录音测试没有声音打开	19
5.3 如何关闭内核打印	19
5.4 无法识别触摸屏	20



1 概述

1.1 编写目的

介绍全志平台上 DragonBoard 的使用以及调试方法，为 Dragonboard 系统的使用和开发提供参考

1.2 适用范围

使用于 DragonBoard 配套的 linux-4.9 内核平台。目前支持的平台有：

内核版本	IC 平台
linux-4.9	A100
linux-4.9	A133
linux-4.9	T509
linux-5.4	A100
linux-5.4	A133

1.3 适用人员

Dragonboard 的开发/使用/维护人员。

2 系统介绍

2.1 系统简介

DragonBoard 是一个基于 Linux BSP，集成了 DirectFB 的图形化板卡测试系统。该系统旨在检测板卡能否在特定的环境中正常工作。

DragonBoard 测试系统的固件可以直接烧录到板卡上，同时，也支持卡启动，减少对 PC 的依赖。

DragonBoard 测试系统的测试流程分为两个部分：自动测试和手动测试。自动测试包括内存、时钟、WIFI、重力传感器等；手动测试包括 SD 卡、U 盘、按键、录音耳机、蓝牙和等。系统上电运行后会自动加载、运行用户启用的测试用例，并将结果显示到界面，用户可从中看到哪些测试项通过，哪些失败。

DragonBoard 测试系统完成一张板卡测试所需的时间具有一定的浮动性，主要受以下几个方面影响：

- 主频和 DDR 频率；
- 测试项数量；
- 测试流水线 and 工人的熟练程度；

主频和 DDR 频率决定了系统的运行速度，越高的频率测试程序运行的越快，所需的时间就越短；测试项数量会影响系统负荷和工人操作时间；另外测试流水线设置是否合理以及工人的熟练程度也直接影响测试时间。

2.2 源码结构介绍

dragonboard 系统的源代码在 longan/test/dragonboard 目录下，具体的源码结构如下所示：



```
├── rootfs //根目录，目录里面的是进入内核里面显示的目录
│   ├── autorun.sh
│   ├── bin
│   ├── dev
│   ├── dragonboard
│   ├── etc
│   ├── home
│   ├── init -> bin/busybox
│   ├── lib
│   ├── linuxrc -> bin/busybox
│   ├── mnt
│   ├── opt
│   ├── proc
│   ├── root
│   ├── sbin
│   ├── sys
│   ├── tmp
│   ├── usr
│   └── var
├── rootfs.ext4
├── src //源码目录
│   ├── build.sh //drangonboard的编译代码，编译drangonboard时会调用该文件
│   ├── core //核心代码，包括测试用例的解析/加载，显示的加载
│   ├── include //头文件
│   ├── lib //链接文件
│   ├── Makefile
│   ├── rule.mk
│   ├── testcases //测试用例目录，里面包含了各个模块的测试用例代码
│   └── view //视图加载目录
├── sysroot
│   ├── bin
│   ├── cedar-dev-pkg-test
│   ├── dev
│   ├── etc
│   ├── include
│   ├── lib
│   ├── sbin
│   ├── usr
│   └── var
```

2.3 系统配置以及编译

由于 DragonBoard 测试系统基于 Linux BSP，因此需要待测试平台的 Linux BSP，即 Linux 内核。系统默认使用 Android 的 Linux 内核。DragonBoard 测试系统的源码位于 longan/test/dragonboard 目录下面，由于 drangonboard 是一个单独的板卡测试系统，所以需要单独编译。具体的配置编译过程如下：

2.3.1 配置 dragonboard

以 A100 为例。

- 步骤 1：进入 longan 目录执行./build.sh config。选择 linux 平台：

```
longan$ ./build.sh config

Welcome to mkscrip setup progress
All available platform:
    0. android
    1. linux
Choice [linux]: 1
```

- 步骤 2：选择 dragonboard：

```
All available linux_dev:
    0. bsp_
    1. dragonboard
    2. sata
    3. longan
    4. tinyos
Choice [bsp]: 1
```

- 步骤 3：选择 IC 为 a100：

```
All available ic:
    0. a100
    1. a133
    2. a50
    3. a63
    4. a64
    5. f113
    6. f133
    7. h3
    8. h6
    9. h616
   10. r328-s2
   11. r328-s3
   12. r328
   13. r329
   14. r528
   15. t3
   16. t507
   17. t7
   18. t8
   19. tv303
   20. v316
   21. v5
   22. v533
   23. v536
   24. v831
   25. v833
Choice [v459]: 0 //此处如果是其他平台选择其他平台
```

- 步骤 4：选择 b3 板子：

```
All available board:
0. b1
1. b3
2. fpga
3. perf1
4. perf2
5. qa
6. ver
Choice [b1]:1
```

- 步骤 5: 选择 flash, 默认选择 default 则可。

```
All available flash:
0. default
1. not
Choice [default]: 0
```

- 步骤 5: 选择内核版本:

```
All available kern_ver:
0. linux-3.10
1. linux-3.4
2. linux-4.4
3. linux-4.9
4. linux-5.4
Choice [linux-4.9]:
```

2.3.2 编译 dragonboard

执行完配置步骤后, 就可以编译 dragonboard 了, 编译 dragonboard 相当于编译 linux+linux 应用程序。进入 longan 目录执行 ./build.sh 则可。

```
longan$ ./build.sh
```

2.3.3 打包 dragonboard

执行完编译步骤后, 最后一个步骤就是打包 dragonboard 固件。

```
longan$ ./build.sh pack //uart0固件, 可测TF卡
longan$ ./build.sh pack_debug //card0固件, 可查看串口log, 但不能测TF卡
longan$ ./build.sh pack_ecure //安全uart0固件, 能测TF卡
longan$ ./build.sh pack_debug_secure //安全card0固件, 可查看串口log, 但不能测TF卡
```

打包完毕后, 会在 longan 的 out 目录下生成固件, 固件的位置如下所示, 以 A100 为例:

```
a100_dragonboard_b3_card0_secure_v0.img //卡打印固件  
或a100_dragonboard_b3_uart0_secure_v0.img //串口打印固件
```

2.3.4 烧写 drangonboard

有两种方式可以烧写 drangonboard 固件，分别是：

1. Nand/eMMC 启动。使用 PhoenixSuit 通过串口直接烧写固件。
2. SD 卡启动。使用 PhoenixCard4.1.1 烧写固件, 用于在 PC 制作“启动卡”。

2.3.5 卡启动注意事项

编译 dragonboard 卡启动固件需要注意以下事项

2.3.5.1 NAND 介质机器

1. 需要将 sys_config.fex，具体路径为：longan/device/config/chips/{IC}/configs/{BOARD}。配置文件的 [nand0_para] 主项下的 nand0_dragonboard = 0 改为 nand0_dragonboard = 1
2. 并且将 sys_config.fex 配置文件的 [target] 主项下的 dragonboard_test = 0 改为 dragonboard_test = 1

2.3.5.2 eMMC 介质机器

1. 需要将 sys_config.fex 配置文件的 [target] 主项下的 dragonboard_test = 0 改为 dragonboard_test = 1
2. 修改后，打包前，必须重新再编译一次内核，即打包前，必须先再一次执行./build.sh

3 测试流程

板卡上电之后系统自动启动，测试程序依照 test_config.fex 配置依次加载，并显示主界面。根据配置的不同，主界面的布局也将不同。当进入主界面之后，用户即看到各个测试项目的状态。这些测试项目被分为两组：自动测试项和手动测试项。自动测试项整个测试过程自动完成，无需用户干预，测试通过测试项目描成蓝色，测试失败测试项目描成红色。手动测试项需要用户参与。其中带 wifi 测试与 mic 测试与 camera 测试的主界面如下图所示，如果没有相关测试选项不会显示相应的选项，如 T507 平台上没有摄像头，不会显示右上角的图像边框。

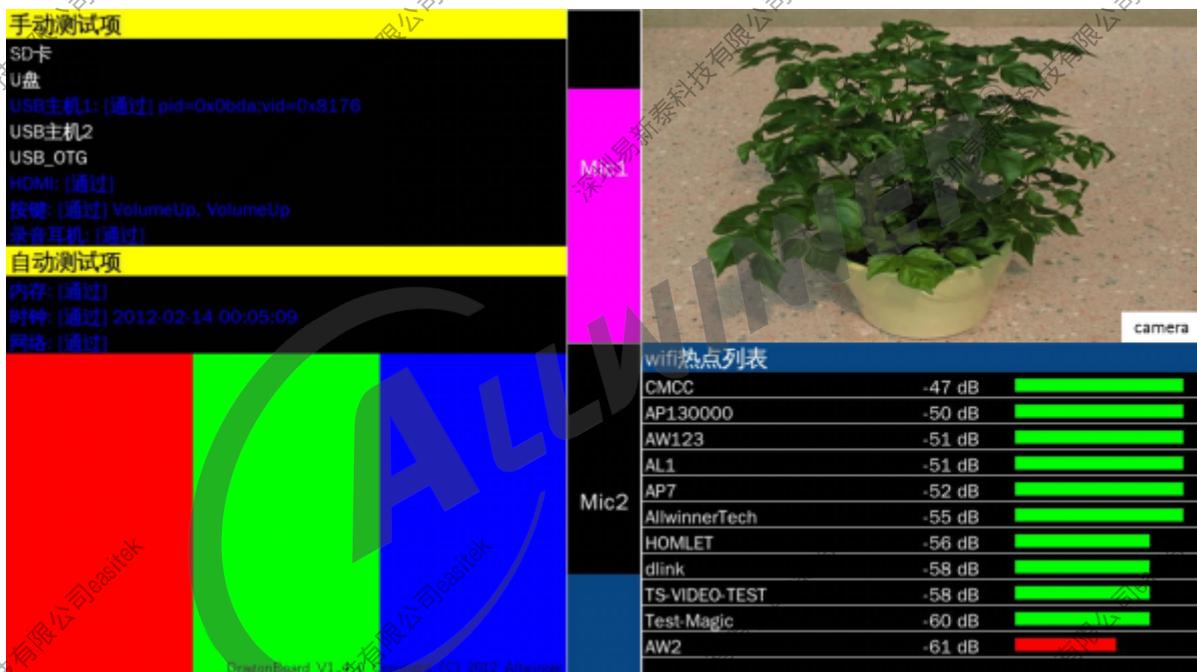


图 3-1: dragonboard

3.1 SD 卡测试

用户插入一张 SD 卡等待 3 秒，如果成功会在界面显示容量并且测试项目变成蓝色，否则测试项目变成红色。

3.2 按键测试

界面会显示用户按下的按键。如果成功测试项目变成蓝色，否则测试项目变成红色。

3.3 WIFI 测试

Dragonboard 会扫描 wifi 热点，并把扫描到的热点显示到 UI 上。而且所有的热点按信号强度依次显示。同时也会显示出信号的 DB 值和信号强度能量条。信号强度大于 60db 的能量条为绿色，表示信号强度较好。信号强度小于 60db 的能量条为红色，表示信号强度较差。

3.4 蓝牙测试

要确保测试蓝牙测试准确，需要特别注意两点：第一，确保附近有蓝牙设备打开。第二，确保该的状态是：让附近所有的蓝牙设备均可检测。

Dragonboard 则会扫描附件可配对蓝牙，配对成功，在串口答应配对蓝牙名称，测试通过，否则失败。

3.5 触摸屏测试

用户可以通过触摸屏幕来检测触摸屏的好坏，界面会实时显示当前坐标值，测试通过。否则失败。

3.6 摄像头测试

用户可以在右上角的方框里面查看到摄像头拍出来的预览照片，并且点击屏幕左边的切换按钮切换前后置摄像头。

3.7 音频测试

可选音频测试方式一：1. 系统开机启动后会立刻播放一段测试音频，测试板载喇叭输出以及外接耳机输出状态，此时可以通过外听喇叭输出确认喇叭输出是否正常或可以插上耳机接听确认耳机输出是否正常。2. 在开机测试音频播放完毕后会进入板载麦克风录音状态，录制外部环境声音，而屏幕中上方会有一个 MIC1 录音音量指示条用于检测确认板载麦克风是否能正常录音，如果 MIC1 音量指示条有幅度变化则表示板载麦克风录音正常，反之 MIC1 音量指示条没有幅度变化的话则表示板载麦克风没接或异常。3. 当 2. 检测到 MIC1 录音强度达到一定等级的时候 ($\geq 1/3$)，此时可以插入带麦耳机进行耳机麦克风录音测试，录制外部环境声音，而屏幕中下方会有一个 MIC2 录音音量指示条用于检测确认插入带麦耳机的耳机麦克风是否录音正常，如果 MIC2 音量指示条有幅度变化则表示耳机麦克风录音正常，反之 MIC2 音量指示条没有幅度变化的话则

表示插入的不是带麦耳机或耳机麦克风异常。4. 当 3. 检测到 MIC2 录音强度达到一定等级的时候 ($\geq 1/3$)，音频测试项“音频基本功能测试”则会显示绿色的“PASS”表示通过。

可选音频测试方式二：1. 系统开机启动后会立刻播放一段测试音频，测试板载喇叭输出以及外接耳机输出状态，此时可以通过外听喇叭输出确认喇叭输出是否正常或可以插上耳机接听确认耳机输出是否正常。2. 开机测试音频播放完后，会自动进入录音并播放测试循环。未插入耳机时，会先通过喇叭播放音频”录音测试开始“，随后立即通过板载麦克风进行录制固定时长的外部环境声音，录制完成后会继续通过喇叭播放音频”播放测试开始“，接着继续播放刚通过板载麦克风录制的音频，以此循环；插入耳机后，会先通过耳机播放音频”录音测试开始“，随后立即通过耳机麦克风进行录制固定时长的外部环境声音，录制完成后会继续通过耳机播放音频”播放测试开始“，接着继续播放刚通过耳机麦克风录制的音频，以此循环。

3.8 USB 测试

用户用 USB 线连接到电脑上的时候，如果 USB 能够正常识别，USB 主机测试项目会变成绿色，否则变成红色。

3.9 重力传感器

开机后如果重力传感器可以正常使用，那么重力传感器测试项目选项会变成绿色，否则变成红色。

3.10 U 盘

用户可以在开机后插入 U 盘，如果识别正常，U 盘测试选项会变成绿色，否则变成红色

3.11 HDMI

用户可以在开机后用 HDMI 线连接到显示屏，显示屏会显示图片，并且 HDMI 测试选项会变成绿色，否则变成红色

4 配置文件

DragonBoard 板卡测试系统提供了一个灵活的配置脚本 `test_config.fex`，位于 `longan/device/config/chips/{IC}/configs/{BOARD}/dragonboard`，方便用户定制自己的测试项目，从而提高系统的效率。

配置文件可以完成以下工作：

1. 修改界面的语言和颜色
2. 启用或者禁用某个测试项目
3. 修改测试项目的参数

该脚本使用 ini 文件格式，由段、键和值三者组成，通常一个段表示一个模块配置。目前要求该配置文件使用 UTF-8 编码。

4.1 模块配置实例

测试模块配置示例：

```
[example]
display_name= "Example"
activated    = 1
program     = "example.sh"
category    = 0
run_type    = 1
```

- `example`: 表示一个模块配置 `example`。
- `display_name`: 当前测试模块显示到界面的名称，字符串类型，最多可容纳 64 个字节。如果为空，测试程序不会运行。如需显示双引号，使用下面语法：

```
display_name= string:"Example"
```

如果模块显示的名称需要不同的语言支持，修改 `display_name` 字段即可。

- `activated`: 0: 不测试该模块，1: 测试该模块。如果用户的方案不需要测试当前模块，请将该项置 0；否则，请将该项置 1。

- **program**: 模块的测试用例代码, 可在 drangonboard 下的 testcase 目录找到, 字符串类型, 最多可容纳 16 个字节。
- **category**: 0: 自动测试模块; 1: 手动测试模块
- **run_type**: 0: 等待当前模块的测试程序执行完毕再运行下一个模块的测试程序; 1: 不等待当前模块的测试程序执行完毕; 一般为了提高整体的测试速度, 对于耗时较长的测试程序建议填 0, 反之填 1。注意, 当 category = 1, 即手动测试模块时, 该项无效。因为手动测试模块会阻止其他测试模块加载, 系统可能无法继续加载其他测试模块。

4.2 DRAM 配置

内存测试配置。配置示例如下:

```
[dram]
display_name= "内存"
activated    = 1
program     = "memtester.sh"
category    = 0
run_type    = 1
dram_size   = 2048
test_size   = 8
```

- **dram_size**: 板卡上使用了多大的 dram, 以 MB 为单位, 在 dram 测试之初, 程序会去检测板卡上 dram 的实际容量, 如果小于配置容量, 则说明贴片存在问题, 测试不通过。
- **test_size**: 使用多大的容量测试 DRAM 的性能, 单位是 MB, 默认使用 8M。实际如果需要测出 DRAM 的性能, 需要设定较大的容量, 但是这样耗时较长, 因此需要用户折中选择一个合适的值。8M 大概可以在 30 秒以内完成 DRAM 性能测试。

4.3 RTC 配置

时钟测试配置。配置示例如下所示:

```
[rtc]
display_name= "时钟"
activated    = 1
program     = "rtctester.sh"
category    = 0
run_type    = 1
```

4.4 WIFI 配置

WIFI 测试配置。配置示例如下所示:

```
display_name= "网络"
activated    = 1
program     = "wifitester.sh"
category    = 0
run_type    = 1
module_count= 1
module_path = "/system/vendor/modules/8723cs.ko"
module_args =
```

- `module_path`: 需要加载的模块全路径, DragonBoard 测试系统将模块文件放在 `/lib/modules/4.9.17/` 目录下, 并且创建了一个 `/system/vendor/modules`, 解决部分 wifi 驱动下载固件失败的问题。
- `module_args`: 模块的参数 (可选)。

4.5 NAND 闪存配置

闪存测试配置, 针对闪存是 nand 方案, 否则请关闭该项测试, 即设 `activated = 0`。

```
[nand]
display_name= "闪存"
activated    = 0
program     = "nandtester.sh"
category    = 0
run_type    = 1
module_path = "/system/vendor/modules/nand.ko"
test_size   = 16
```

- `test_size`: 测试 nand 读写的大小。

注意: 1、请务必保证待测 nand 闪存为“白片”, 否则因为 mbr 不对应导致测试失败。2、打开 nand 闪存测试, 则务必保证关闭 emmc 闪存测试项目

4.6 EMMC 闪存配置

闪存测试配置, 针对闪存是 emmc 方案, 否则请关闭该项测试, 即设 `activated = 0`。

```
[emmc]
display_name= "emmc闪存"
activated    = 1
program     = "emmctester.sh"
category    = 0
run_type    = 1
test_size   = 128
```

- `test_size`: 测试 emmc 闪存读写的大小。

注意：1、请务必保证待测 emmc 闪存为“白片”，否则因为 mbr 不对应导致测试失败。2、打开 emmc 闪存测试，则务必保证关闭 nand 闪存测试项目。

4.7 蓝牙配置

蓝牙测试配置，具体配置如下：

```
[bluetooth]
display_name= "蓝牙"
activated    = 1
program     = "bttester.sh"
test_time   = 3
category    = 0
dst_bt      = ":"
run_type    = 1
module_path =
device_node = "/dev/ttyS1"
baud_rate   =
bt_vnd      = "realtek"
rtk_bt_chip = "rtl8723cs_xx"
```

4.8 MMC 配置

SD 卡测试配置，具体配置如下：

```
[mmc]
display_name= "SD卡"
activated    = 1
program     = "mmctester.sh"
category    = 1
run_type    = 1
```

4.9 Udisk 配置

U 盘测试配置，具体配置如下所示：

```
[udisk]
display_name= "U盘"
activated    = 1
program     = "udisktester.sh"
category    = 1
run_type    = 1
```

4.10 KEY 配置

按键测试配置，具体配置如下所示：

```
[key]
display_name= "按键"
activated    = 1
program     = "keytester"
category    = 1
run_type    = 1
```

4.11 TP 配置

触摸屏测试配置，具体配置如下所示：

```
[tp]
display_name= "触摸"
type        = 1
module_path = "/system/vendor/modules/gslX680new.ko"
device_name = "gslX680"
```

- type：触摸屏的类型，0 表示电阻屏，1 表示电容屏。电阻屏在系统启动后会运行 TP 校准的程序，只有通过调屏程序才会进入 dragonboard 界面。
- module_path：需要加载的模块全路径

4.12 摄像头测试

摄像头测试配置，具体配置如下所示：

```
[camera]
activated    = 1
module_count= 8
module1_path= "/system/vendor/modules/videobuf2-core.ko"
module2_path= "/system/vendor/modules/videobuf2-memops.ko"
module3_path= "/system/vendor/modules/videobuf2-dma-contig.ko"
module4_path= "/system/vendor/modules/videobuf2-v4l2.ko"
module5_path= "/system/vendor/modules/vin_io.ko"
module6_path= "/system/vendor/modules/gc2385_mipi.ko"
module7_path= "/system/vendor/modules/gc030a_mipi.ko"
module8_path= "/system/vendor/modules/vin_v4l2.ko"
dev_cnt     = 2
csi_cnt     = 2
fps         = 10
```

4.13 音频测试

音频测试配置，具体配置如下所示：

```
[mic]
display_name= "音频基本功能测试"
activated    = 1
program     = "mictester"
category    = 1
run_type    = 1
delay       = 3
volume      = 0
music_playtime = 20
music_file  = "/dragonboard/data/test48000.pcm"
samplerate  = 48000
mic1_used   = 1
mic2_used   = 1
mic1_gain   = 1
mic2_gain   = 1
mic1_threshold = 4096
mic2_threshold = 4096

[soundcard]
card = 0;
device = 0;
mic_spk_test = 1;
test_wav_time = 5;
```

其中，- mic_spk_test：用于选择当前音频测试方式，当值为 0 时，则选用音频测试方式一；当值为 1 时，则选用音频测试方式二，当前默认选用音频测试方式二；- test_wav_time：当选用音频测试方式二时有效，用于控制进入录音并播放测试循环后录音与播放的音频时长，当前默认时长为 5s。

4.14 重力传感器测试

重力传感器配置，具体配置如下所示：

```
-----
; TODO: 重力传感器
;
-----
[gsensor]
display_name= "重力传感器"
activated    = 1
program     = "gsensortester.sh"
category    = 0
run_type    = 1
module_count= 1
module1_path= "/system/vendor/modules/sc7a20.ko"
device_count= 1
device1_name = "sc7a20"
```

4.15 U 盘测试

U 盘测试配置，具体配置如下所示：

```
[udisk]
display_name= "U盘"
activated    = 1
program     = "udisktester.sh"
category    = 1
run_type    = 1
```

4.16 HDMI 测试

HDMI 测试配置，具体配置如下所示：

```
[hdmi]
display_name= "HDMI"
activated    = 1
program     = "hdmittester.sh"
category    = 1
run_type    = 1
sound_file  = "/dragonboard/data/test48000.pcm"
samplerate  = 48000
```

4.17 界面配置

配置显示界面，具体配置如下所示：

```
[df_view]
tv_scale_factor = 95
manual_menu_name = "手动测试项"
auto_menu_name  = "自动测试项"
clear_button_name = "清屏"
wifi_menu_name  = "wifi热点列表"
font_size       = 48
menu_bgcolor    = 1
menu_fgcolor    = 7

item_init_bgcolor = 7
item_init_fgcolor = 0
item_ok_bgcolor   = 7
item_ok_fgcolor   = 2
item_fail_bgcolor = 7
item_fail_fgcolor = 5
tp_draw_color    = 0

;item_init_bgcolor = 0
;item_init_fgcolor = 7
;item_ok_bgcolor   = 0
```

```

;item_ok_fgcolor    = 6
;item_fail_bgcolor  = 0
;item_fail_fgcolor  = 5
;tp_draw_color     = 7

pass_str           = "通过"
fail_str           = "失败"

```

- tv_scale_factor: HDMI 输出缩放因子。现在市场上的很多电视都会对输入的视频裁边，导致视频内容不能完全的显示或者是电视四周有黑边。针对这种情况，dragonboard 会根据这个缩放因子对输出视频进行缩放。如: tv_scale-factor=80. 配置输出为 1280*720 分辨率输出时。实际的可显示的图像区域为: (1280*80%)*(720*80%)，也即是 1024*576。此缩放因子有效范围为 50~100，超出此范围的将自动认为不缩放，也就是 tv_scale-factor=100。
- manual_menu_name: 手动测试项的菜单显示内容。
- auto_menu_name: 自动测试项的菜单显示内容。
- clear_button_name: 清屏按键显示内容。
- font_size: 测试用例名称的字体大小。
- menu_bgcolor: 菜单的背景色，从 Color Index 中选择。
- menu_fgcolor: 菜单的前景色，从 Color Index 中选择。
- item_init_bgcolor: 测试项初始背景色，从 Color Index 中选择。
- item_init_fgcolor: 测试项初始前景色，从 Color Index 中选择。
- item_ok_bgcolor: 测试项通过背景色，从 Color Index 中选择。
- item_ok_fgcolor: 测试项通过前景色，从 Color Index 中选择。
- item_fail_bgcolor: 测试项失败背景色，从 Color Index 中选择。
- item_fail_fgcolor: 测试项失败前景色，从 Color Index 中选择。
- tp_draw_color: 触摸轨迹前景色，从 Color Index 中选择。为了能够清楚看到触摸轨迹，请选择和大部分区域的背景色相反的颜色。
- pass_str: 测试项通过提示语。
- fail_str: 测试项失败提示语。

Color Index 的选择如下所示：

Color	Index
白色	0
黄色	1
绿色	2
青色	3
洋红	4
红色	5

Color	Index
-------	-------

蓝色	6
----	---

黑色	7
----	---



5 FAQs

5.1 如何添加新的方案？

在 `longan/device/config/chips/{IC}/configs/{方案目录}/dragonboard` 目录下添加新的方案目录，可以先从其他板级的 `dragonboard` 拷贝 `test_config.fex` 到新的方案目录，将对应 android 方案的 `sys_config.fex` 拷贝到新的方案目录。针对新的方案，修改 `test_config.fex` 文件。一般需要修改启用或者禁用（activated）某些测试项目，修改模块文件的全路径，或者修改测试项目在界面上显示的语言（display_name）。下面列举几个经常修改的模块：

1. WIFI 模块：主要修改模块文件的全路径和参数。
2. G-Sensor 模块：主要修改模块文件的全路径。
3. TP 模块：主要修改模块文件的全路径和触摸屏类型。
4. Camera 模块：主要修改模块文件的全路径。注意模块数量改变后修改 `module_count`。

5.2 为什么录音测试没有声音打开

检查 `sys_config.fex` 中以下的配置项有没有打开

```
[s_i2s1]
daudio_used          = 1
[audio0]
audio_pa_ctrl        = port:PA16<1><default><default><0>
```

检查板级目录下的 `board.dts` 的 `audio` 配置项有没有打开

检查内核下的设备树里面的 `audio` 配置项有没有打开

同时，`dragonboard` 的声音只可以从耳机或者外放输出，两者不可以同时输出，当外放没有声音时，需确认耳机是否有插入。

5.3 如何关闭内核打印

修改 `longan/test/dragonboard/default/env.cf` 文件，找到下面这行：

```
loglevel=7
```

改成

```
loglevel=4
```

5.4 无法识别触摸屏

具体现象为系统启动后会显示一会测试界面，后面测试界面消失，显示灰屏界面，查看串口 log 信息，有以下：

```
11.554022] android_work: sent uevent USB_STATE=CONNECTED
dragonboard(error): Can't not open /dev/video0
dragonboard(msg): buildin_tc_init...
view/df_view_pad.c <2071>:
    (#) DirectFBError [dfb->GetInputDevice(dfb, 1, &tp_dev)]: Requested ID not
    found!
(!!!) *** WARNING [Application exited without deinitialization of DirectFB!] *** [core.c
:881 in dfb_core_deinit_check()]
(!!!) *** WARNING [still objects in 'Window Pool'] *** [object.c:241 in
fusion_object_pool_destroy()]
(!!!) *** WARNING [still objects in 'Layer Region Pool'] *** [object.c:241 in
fusion_object_pool_destroy()]
(!!!) *** WARNING [still objects in 'Layer Context Pool'] *** [object.c:241 in
fusion_object_pool_destroy()]
(!!!) *** WARNING [still objects in 'Surface Pool'] *** [object.c:241 in
fusion_object_pool_destroy()]
```

可以看到输入设备没找到，找到板子对应的触摸屏驱动，到内核目录下 (linux) 执行 make ARCH=arm64(arm) menuconfig 选上重新编译打包即可。

著作权声明

版权所有 © 2021 珠海全志科技股份有限公司。保留一切权利。

本档及内容受著作权法保护，其著作权由珠海全志科技股份有限公司（“全志”）拥有并保留一切权利。

本档是全志的原创作品和版权财产，未经全志书面许可，任何单位和个人不得擅自摘抄、复制、修改、发表或传播本档内容的部分或全部，且不得以任何形式传播。

商标声明

、 **全志科技** （不完全列举）均为珠海全志科技股份有限公司的商标或者注册商标。在本档描述的产品中出现的其它商标，产品名称，和服务名称，均由其各自所有人拥有。

免责声明

您购买的产品、服务或特性应受您与珠海全志科技股份有限公司（“全志”）之间签署的商业合同和条款的约束。本档中描述的全部或部分产品、服务或特性可能不在您所购买或使用的范围内。使用前请认真阅读合同条款和相关说明，并严格遵循本档的使用说明。您将自行承担任何不当使用行为（包括但不限于如超压，超频，超温使用）造成的不利后果，全志概不负责。

本档作为使用指导仅供参考。由于产品版本升级或其他原因，本档内容有可能修改，如有变更，恕不另行通知。全志尽全力在本档中提供准确的信息，但并不确保内容完全没有错误，因使用本档而发生损害（包括但不限于间接的、偶然的、特殊的损失）或发生侵犯第三方权利事件，全志概不负责。本档中的所有陈述、信息和建议并不构成任何明示或暗示的保证或承诺。

本档未以明示或暗示或其他方式授予全志的任何专利或知识产权。在您实施方案或使用产品的过程中，可能需要获得第三方的权利许可。请您自行向第三方权利人获取相关的许可。全志不承担也不代为支付任何关于获取第三方许可的许可费或版税（专利税）。全志不对您所使用的第三方许可技术做出任何保证、赔偿或承担其他义务。