

A64

DragonBoard 使用说明书

目录

A64.....	1
DragonBoard 使用说明书.....	1
1. 前言.....	4
1.1 简介.....	4
1.2 测试时间.....	4
1.3 目的.....	4
1.4 名词解释.....	4
2. 使用入门.....	5
2.1 编译内核.....	5
2.2 配置测试项目.....	6
2.3 打包固件.....	6
2.4 烧写固件.....	6
2.5 启动系统.....	7
2.5.1 安全固件的卡启动注意事项.....	7
2.5.2 非安全固件的卡启动注意事项.....	7
3. 测试流程.....	8
4. 配置文件.....	10
4.1 配置文件格式限制.....	10
4.2 模块配置示例.....	10
4.3 DRAM 配置.....	11
4.4 RTC 配置.....	11
4.5 WIFI 配置.....	11
4.6 Nand 配置.....	12
4.7 Emmc 闪存配置.....	12
4.8 蓝牙配置.....	12
4.9 MMC 配置.....	13
4.10 UDisk 配置.....	13
4.11 Usb_HOST1 配置（可选）.....	13
4.12 Usb_HOST2 配置（可选）.....	13
4.13 Usb_OTG 配置.....	13
4.14 HDMI 配置.....	14
4.15 Key 配置.....	14
4.16 MIC 配置.....	14
4.17 TP 配置.....	15
4.18 Camera 配置.....	15
4.19 界面配置.....	16
5. FAQs.....	19
5.1 为系统启动后会进入 TP 校准界面？.....	19
5.2 如何添加新的方案？.....	19
5.3 为什么录音测试没有声音输出？.....	19
5.4 如何关闭内核的打印？.....	19
6. 参考资料.....	20

7. Declaration.....21

Confidential

1. 前言

1.1 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、HDMI、按键、红外、录音耳机、蓝牙和触摸等。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。

1.2 测试时间

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；
- 测试流水线和工人的熟练程度；

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。

1.3 目的

本文档主要向用户阐述在 A64 平台上 DragonBoard 测试系统的使用方法。

1.4 名词解释

2.使用入门

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 buildroot/target/dragonboard 目录下，A64 SDK 默认包含了 dragonboard，编译内核时会自动编译这些源码，并生成所需的测试用例。

2.1 编译内核

在 lichee 根目录下执行下面的命令，编译一个供 DragonBoard 使用的内核，这个内核和 Android 的 Linux 内核使用相同的配置：

打包使用下面命令：

```
cd lichee
./build.sh config
```

1. 选择芯片平台

```
Welcome to mkscript setup progress
```

```
All available chips:
```

- 0. sun50iw1p1
- 1. sun8iw10p1
- 2. sun8iw1p1
- 3. sun8iw3p1
- 4. sun8iw5p1
- 5. sun8iw6p1
- 6. sun8iw7p1
- 7. sun8iw8p1
- 8. sun8iw9p1
- 9. sun9iw1p1

```
Choice: 0
```

2. 选择 dragonboard

```
All available platforms:
```

- 0. android
- 1. dragonboard
- 2. linux
- 3. camdroid
- 4. secureandroid

```
Choice: 1
```

3. 选择内核

```
All available kernel:
```

- 0. linux-3.10
- 1. linux-3.4

```
Choice: 0
```

4. 选择方案

```
All available boards:
```

0. fpga
1. p1
2. perf1_v1_0
3. perf2_v1_0
4. perf3_v1_0
5. t1
6. t1_v1

```
Choice: 1
```

编译完内核后自动编译 DragonBoard 测试系统，并生成 rootfs.ext4，位于 out/sun50iw1p1/dragonboard/common 目录下，输入命令得其目录树如下：

```
tree -L 1
```

```
.
├── arisc
├── boot.img
├── buildroot
├── rootfs.ext4
└── vmlinux.tar.bz2
```

其中，rootfs.ext4 包含了一个根文件系统，里面有驱动模块文件，位于 /lib/modules/3.10.65 目录下，还有 dragonboard 测试用例和数据，位于 /dragonboard 目录下。

注意：

虽然在量产阶段，卡启动是最好的选择，但是在开发阶段使用 flash 启动开发更加方便、快捷。如果客户使用 flash 启动来开发，建议关闭 flash 的读写测试。因为 flash 的读写测试是针对 nanda 这个分区，这个分区默认做为了 bootloader 分区，这个分区中存放了 boot 相关的代码，flash 测试的代码会破坏这个分区中的文件，从而导致下次开机的时候不能正常启动。

2.2 配置测试项目

DragonBoard 测试系统提供了一个灵活的配置脚本 test_config.fex，方便用户定制自己的测试项目，从而提高系统的效率。test_config.fex 能够完成以下基本工作：

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

测试项目的各项配置说明详见第 4 节配置文件。

2.3 打包固件

打包使用下面命令：

```
$ ./build.sh pack_secure    【安全固件的打包命令】
$ ./build.sh pack          【非安全固件的打包命令】
```

2.4 烧写固件

1. Nand 启动

使用 PhoenixSuit 烧写固件。

2. SD 卡启动

使用 PhoenixCard3.09 或更新版本烧写固件，用于在 PC 制作“启动卡”。

2.5 启动系统

1. Nand/eMMC 启动

2. SD 卡启动 (上电前请先将 SD 卡插入板卡卡槽)

2.5.1 安全固件的卡启动注意事项

编译 A64 dragonboard 安全卡固件需要注意以下事项

一、 NAND 介质机器

1. 需要将 sys_config.fex 配置文件的 [nand0_para] 主项下的 nand0_dragonboard = 0 改为 nand0_dragonboard = 1
2. 打包固件的命令必须是 ./build.sh pack_secure
3. 打包固件前 tools 目录下必须有用于签名证书的 keys

二、 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. 打包固件的命令必须是 ./build.sh pack_secure
4. 打包固件前 tools 目录下必须有用于签名证书的 keys

2.5.2 非安全固件的卡启动注意事项

编译 A64 dragonboard 非安全卡固件需要注意以下事项

三、 NAND 介质机器

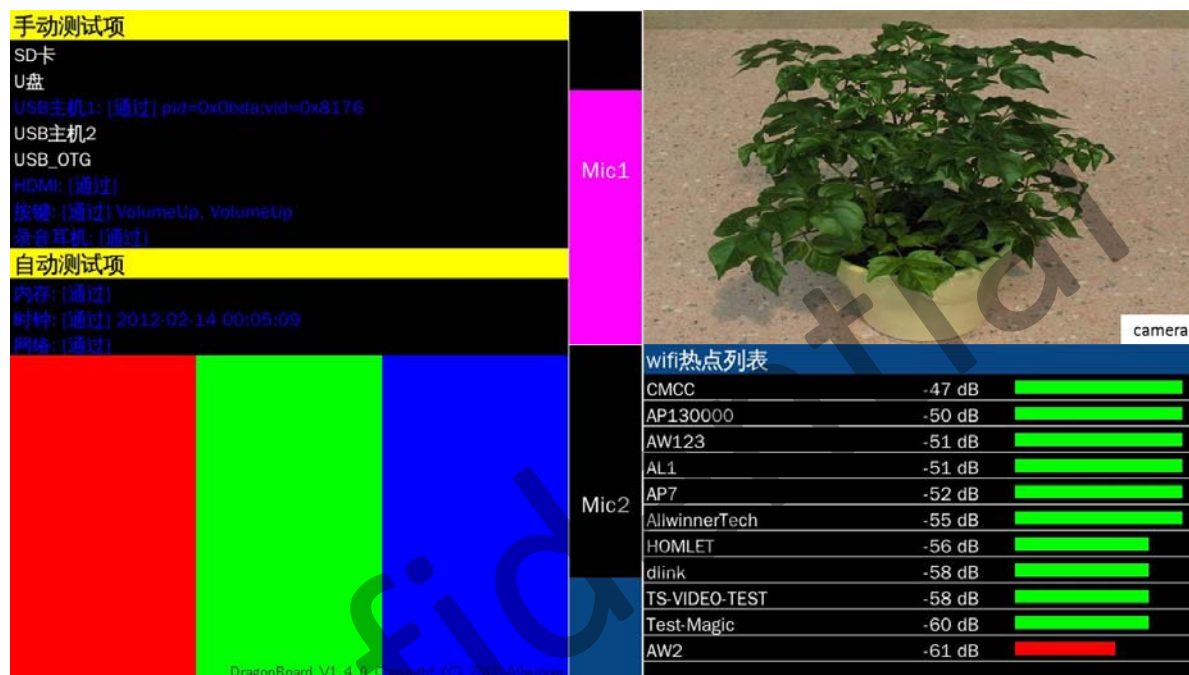
1. 需要将 sys_config.fex 配置文件的 [nand0_para] 主项下的 nand0_dragonboard = 0 改为 nand0_dragonboard = 1
2. 打包固件的命令必须是 ./build.sh pack

四、 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. 打包固件的命令必须是 ./build.sh pack

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：



1. SD 卡

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。

2. U 盘

用户插入一个 U 盘等待 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. 按键

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。

4. 红外

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。

5. 录音耳机

根据用户配置，如果用户需要在 mic 录音前播放一段音乐到耳机口，则会先播放音乐，到用户配置的规定时间后音乐会停止。录音程序会在延时 5 秒后（默认值）开始录音，对于 pad 产品，用户可以插入耳机测试耳机输出是否正常，同时也可判断 Speaker 是否正常。对于 HDMI 输出的产品，录音会出从 HDMI 输出。在录音的同时，UI 上会动态直观的显示录音的能量条。

6. 触摸

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。

8. 网络

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 的为能量条为绿色，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。

9. 蓝牙

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。

Confidential

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 `/tools/pack/chips/sun50iw1p1/configs/dragonboard/{方案目录}`，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 配置文件格式限制

4.2 模块配置示例

测试模块配置示例：

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

1. [example]

[example]表示一个模块配置 example。

2. 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。

3. 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。

4. program

模块的测试程序，字符串类型，最多可容纳 16 个字节。

5. category

0：自动测试模块

1：手动测试模块

6. 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。

4.3 DRAM 配置

内存测试配置。

```
[dram]
display_name= "内存"
activated    = 1
program     = "memtester.sh"
category    = 0
run_type    = 1
dram_size   = 1024
test_size   = 8
platform    = "a64"
```

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。

4.4 RTC 配置

时钟测试配置。

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

4.5 WIFI 配置

WIFI 测试配置。

```
[wifi]
display_name= "网络"
activated    = 1
program     = "rt_wifitester.sh"
category    = 0
run_type    = 1
module_count= 1
module_path = "/system/vendor/modules/8723bs.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/3.10.65/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）。

4.6 Nand 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

4.7 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。

4.8 蓝牙配置

蓝牙测试配置

```
[bluetooth]
display_name= "蓝牙"
activated    = 1
program     = "rt_bttester.sh"
test_time   = 3
category    = 0
dst_bt      = ":"
run_type    = 1
device_node = "/dev/ttyS1"
```

4.9 MMC 配置

SD 卡测试配置。

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

4.10 UDisk 配置

U 盘测试配置。

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

4.11 Usb_HOST1 配置（可选）

读出连接到 host1 的设备的 PID,VID 到 UI。

```
[Usb_HOST1]
display_name= "USB 主机 1"
activated    = 0
program     = "host1tester.sh"
category    = 1
run_type    = 1
```

4.12 Usb_HOST2 配置（可选）

读出连接到 host2 的设备的 PID,VID 到 UI。

```
[Usb_HOST2]
display_name= "USB 主机 2"
activated    = 0
program     = "host2tester.sh"
category    = 1
run_type    = 1
```

4.13 Usb_OTG 配置

读出连接到 otg 的设备的 PID,VID 到 UI。

```
[Usb_OTG]
display_name= "USB_OTG"
activated    = 0
```

```

program    = "otgtester.sh"
category   = 1
run_type   = 1

```

4.14 HDMI 配置

HDMI 测试配置

```

[hDMI]
display_name= "HDMI"
activated    = 1
program     = "hdmittester.sh"
category    = 1
run_type    = 1
sound_file  = "/dragonboard/data/test48000.pcm"
samplerate  = 48000
support_mode= 2
module_count= 1
module1_path= "/system/vendor/modules/hdmi.ko"

```

测试步骤：

1、设置 test_config.fex

[hDMI]

activated = 1 使能测试项

module1_path= "/system/vendor/modules/hdmi.ko" 配置正确的驱动

2、连接电视，看到电视端有彩色条纹，平板端显示 HDMI 测试 pass

4.15 Key 配置

按键测试配置。

```

[key]
display_name= "按键"
activated    = 1
program     = "keytester"
category    = 1
run_type    = 1
module_path = "/system/vendor/modules/sunxi-keyboard.ko"

```

4.16 MIC 配置

录音耳机测试配置。

```

[mic]
display_name= "录音耳机"
activated    = 1
program     = "mictester"
category    = 1
run_type    = 1
delay       = 3
music_playtime =10

```

```
music_file = "/dragonboard/data/test48000.pcm"
samplerate = 48000
mic1_used = 1
mic2_used = 0
mic1_threshold = 8191
mic2_threshold = 8191
```

delay: 录音前的延时。(单位: 秒)

music_playtime : 录音前音乐播放时间。(单位: 秒)

music_file: 要播放音乐文件的路径。

samplerate : 要播放音乐文件的采样率。

mic1_used: 是否测试 mic1

mic2_used: 是否测试 mic2

mic1_threshold: mic1 测试通过振幅阈值, 最大值为 32767

mic2_threshold: mic2 测试通过振幅阈值, 最大值为 32767

4.17 TP 配置

触摸测试配置。

```
[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。

4.18 Camera 配置

摄像头测试配置。

```
[camera]
activated = 1
module_count= 7
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/vfe_io.ko"
module5_path= "/system/vendor/modules/gc0328c.ko"
module6_path= "/system/vendor/modules/gc2145.ko"
module7_path= "/system/vendor/modules/vfe_v4l2.ko"
dev_cnt = 2
csi_cnt = 1
fps = 10
```

module_count: 需要加载的模块总数。

module_x_path: “x” 表示第几个需要加载的模块, 从 1 开始。

dev_cnt: 设置摄像头个数。

csi_cnt: 设置 csi 通路, 一般情况下是一路 csi 拖两个摄像头; 如果两路 csi 各拖一个摄像头, 则设成 2。

fps: 帧率。

4.19 界面配置

```
;
; manual_menu_name
;   manual test case menu name, 32bytes.
; auto_menu_name
;   auto test case menu name, 32bytes.
; clear_button_name
;   clear screen button name, 8bytes.
; font_size
;   test case font display size, valid value: 20pixel(default), 24pixel.
; height_adjust
;   adjust height of manual test case and auto test case.
;
; Color Index
;   0: White
;   1: Yellow
;   2: Green
;   3: Cyan
;   4: Magenta
;   5: Red
;   6: Blue
;   7: Black
;   8: Beauty
; menu_bgcolor
;   The background color index of test case category menu.
; menu_fgcolor
;   The foreground color index of test case category menu.
; item_init_bgcolor
;   The background color index of test case item init status.
; item_init_fgcolor
;   The foreground color index of test case item init status.
; item_ok_bgcolor
;   The background color index of test case item OK status.
; item_ok_fgcolor
;   The foreground color index of test case item OK status.
; item_fail_bgcolor
;   The background color index of test case item fail status.
; item_fail_fgcolor
;   The foreground color index of test case item fail status.
;
```

```

; pass_str
; The string display after test case display_name when test OK.
; fail_str
; The string display after test case display_name when test Fail.
;-----
[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"
font_size            = 20
menu_bgcolor         = 1
menu_fgcolor         = 7

item_init_bgcolor    = 7
item_init_fgcolor    = 0
item_ok_bgcolor      = 7
item_ok_fgcolor      = 6
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

Color	Index
白色	0
黄色	1
绿色	2
青色	3
洋红	4
红色	5
蓝色	6
黑色	7

5.FAQs

5.1 为系统启动后会进入 TP 校准界面?

电阻屏需要 TP 校准，如果你使用的是电容屏，请在 `test_config.fex` 文件的 TP 配置里把 `type` 置成 1，这样系统启动后就不会进入 TP 校准界面。详细参考 4.13 TP 配置。

5.2 如何添加新的方案?

在 `tools\pack\chips\sun50iw1p1\configs\dragonboard` 目录下添加新的方案目录，从标案 t1 上拷贝 `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. IR 模块
主要修改模块文件的全路径。
4. TP 模块
主要修改模块文件的全路径和触摸屏类型。
5. 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`。

5.3 为什么录音测试没有声音输出?

检查 `sys_config1.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>
```

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。

5.4 如何关闭内核的打印?

修改 `\tools\pack\chips\sun50iw1p1\configs\dragonboard\default\env.cf` 文件，找到下面这行：

```
loglevel=7
```

改成：

```
loglevel=4
```

6. 参考资料

《A83T DragonBoard 使用说明书.doc》

Confidential

7. Declaration

This(全志科技 A64 DragonBoard 使用说明书) is the original work and copyrighted property of Allwinner Technology (“Allwinner”). Reproduction in whole or in part must obtain the written approval of Allwinner and give clear acknowledgement to the copyright owner.

The information furnished by Allwinner is believed to be accurate and reliable. Allwinner reserves the right to make changes in circuit design and/or specifications at any time without notice. Allwinner does not assume any responsibility and liability for its use. Nor for any infringements of patents or other rights of the third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Allwinner. This datasheet neither states nor implies warranty of any kind, including fitness for any particular application.

Confidential