

密级状态: 绝密() 秘密() 内部() 公开(√)

Security Class: Top-Secret() Secret() Internal() Public(√)

Rockchip PCBA测试工具开发指南

Rockchip PCBA Test Tool Developer Guide

| | | |
|---|-------------------------|---------------------|
| <p>文件状态 Status: <input type="checkbox"/> 草稿 Draft <input checked="" type="checkbox"/> 正式发布 Released <input type="checkbox"/> 正在修改 Modifying</p> | 文件标识 File No. | RK-KF-YF-264 |
| | 当前版本 Current Version | V1.3 |
| | 作者 Author | 卞金晨 Bian Jinchun |
| | 完成日期 Finish Date | 2020-07-09 |
| | 审核 Auditor | 陈海燕 Chen Haiyan |
| | 审核日期 Finish Date | 2020-07-09 |

免责声明 Disclaimer

本档按“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。
THIS DOCUMENT IS PROVIDED “AS IS”. ROCKCHIP ELECTRONICS CO., LTD.(“ROCKCHIP”)DOES NOT PROVIDE ANY WARRANTY OF ANY KIND, EXPRESSED, IMPLIED OR OTHERWISE, WITH RESPECT TO THE ACCURACY, RELIABILITY, COMPLETENESS, MERCHANTABILITY, FITNESS FOR ANY PARTICULAR PURPOSE OR NON-INFRINGEMENT OF ANY REPRESENTATION, INFORMATION AND CONTENT IN THIS DOCUMENT. THIS DOCUMENT IS FOR REFERENCE ONLY. THIS DOCUMENT MAY BE UPDATED OR CHANGED WITHOUT ANY NOTICE AT ANY TIME DUE TO THE UPGRADES OF THE PRODUCT OR ANY OTHER REASONS.

商标声明 Trademark Statement

“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，由其各自所有者所有。

“Rockchip”, “瑞芯微”, “瑞芯” shall be Rockchip’s registered trademarks and owned by Rockchip. All the other trademarks or registered trademarks mentioned in this document shall be owned by their respective owners.

版权所有 © 2020 瑞芯微电子股份有限公司 Copyright © 2020 Rockchip Electronics Co., Ltd.

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。

Beyond the scope of fair use, neither any entity nor individual shall extract, copy, or distribute this document in any form in whole or in part without the written approval of Rockchip.

| 版本号 Version no. | 作者 Author | 修改日期 Revision Date | 修改说明 Revision Description | 备注 Remark |
|-----------------------|------------------------|--------------------------|---|--------------|
| V1.0 | 李东强 Li Dongqiang | 2017-02- 15 | 发布初稿 Initial version release | |
| V1.1 | 刘益星 Liu Yixing | 2017-12- 22 | 修订部分描述错误 Correct some descriptions | |
| V1.2 | 林龙坚 Lin Longjian | 2018-05- 09 | 增加bt测试配置文件说明 Add the configuration file instruction for bt test | |
| V1.3 | 卞金晨 Bian Jinchen | 2020-07- 09 | 支持Android 10.0 Support Android 10.0 | |

文档问题反馈: kenjc.bian@rock-chips.com

If there is any question about the document, please send email to: kenjc.bian@rock-chips.com

概述 Overview

PCBA 测试工具用于帮助在量产的过程中快速地甄别产品功能的好坏，提高生产效率。目前包括屏幕（LCD）、无线（wifi）、蓝牙（bluetooth）、DDR/EMMC存储、SD卡（sdcard）、UST HOST、按键（KEY），喇叭耳机（Codec）测试项目。

PCBA test tool is used to help quickly identify product features failures during production to improve the production efficiency. Current test items include panel (LCD), wireless (Wi-Fi), Bluetooth, DDR/eMMC memory, SD card, USB HOST, KEY, speaker earphone (Codec).

这些测试项目包括自动测试项和手动测试项，无线网络、DDR/EMMC、以太网为自动测试项，按键、SD卡、USB HOST、Codec为手动测试项目。

These test items include automatic test items and manual test items. Wireless network, DDR/eMMC, Ethernet are automatic test items, while key, SD card, USB Host, Codec are manual test items.

工具通过配置文件 `test_config.cfg` 对测试项进行配置，并可根据需求增加新测试项，具体的配置说明请参考本文“配置说明”。

The tool configures the test items through the configuration file `test_config.cfg`, and can add

new test items according to the requirement. For the detailed configuration, please refer to "configuration instruction" in this document.

测试项说明 Test item introduction

测试项分为 自动测试项 和 手动测试项。

Test items are divided into `automatic test item` and `manual test item`.

自动测试项由系统自动进行测试并判断测试结果，如：网络、存储等；手动测试项需要由人工配合完成或者配合判断测试结果，如：录音，按键，U盘，SD卡等。

The system automatically tests and judges the test result for the automatic test items, such as network, memory and so on. While it requires people to help to test or judge the test result for the manual test items, such as audio recording, KEY, U disk, SD card and so on.

测试项分别有“红”，“黄”两种颜色表示不同的测试状态。

There are two colors "red" and "yellow" to separately display the different test status for the test items.

黄色：未测试项或者正在测试的项，通过后显示[PASS].

Yellow: the item waiting for test or under testing, it will display [PASS] after testing successfully.

红色：测试未通过项

Red: test failed

自动测试项 Automatic test item

- DDR/EMMC容量检测 DDR/EMMC capability detection

DDR/EMMC容量检测为自动测试项，自动显示当前设备DDR容量和EMMC容量大小，显示单位为GB，常见DDR容量为1GB、2GB、4GB等，常见EMMC容量为2GB、4GB、8GB、16GB、32GB、64GB、128GB等。测试结果示例如下：“系统存储：[通过] { DDR: 2GB, EMMC: 16GB }”

DDR/EMMC capability detection is the automatic test item, which will automatically display DDR and EMMC size of current device with the unit GB. Generally DDR capability is 1GB, 2GB, 4GB, etc., while EMMC capability is 2GB, 4GB, 8GB, 16GB, 32GB, 64GB, 128GB, etc. Here is the example of the test result: "system memory: [PASS] { DDR: 2GB, EMMC: 16GB }"

- 无线网络 (wifi测试) Wireless network (Wi-Fi test)

Wi-Fi为自动测试项，会自动扫描周边的AP，并显示信号最强的那个AP名字及信号强度。信号强度根据AP强度显示0到4格。测试结果如下：`网络：[通过] { "AP WIFI" 信号强度 4 格 }`

Wi-Fi is the automatic test item. It will automatically scan the surrounding AP and display the name and the signal strength of the AP which has the strongest signal. The signal strength displays from 0 to 4 bars based on the AP strength. The test result is: `network: [PASS] { "AP WIFI" Signal strength 4 bars }`

- 以太网测试 Ethernet test

有两种测试方式，默认以ping的方式测试以太网的通信功能，发5个包收5个包，0包丢失。本机网址和ping的目标网址在external/rk-pcba-test/res/test_cong.cfg中修改。

There are two test methods. The default is to test Ethernet communication function by ping. Send 5 packages, receive 5 packages, 0 package is lost. The local website and the target ping website are modified in external/rk-pcba-test/res/test_cong.cfg.

另一种测试方法，是以检测网卡是否启动来判断，网卡启动时会会有一个inet6的网址，如：“inet6 addr: fe80::c8eb:eaff:fe6d:730d/64 Scope:Link”。

The other test method is to judge by detecting whether the network card is enabled or not. There will be an inet6 website if the network card is enabled, such as "inet6 addr: fe80::c8eb:eaff:fe6d:730d/64 Scope:Link".

检测到有inet6网址，则认为以太网是正常的。（这种方式不能确定网络的通信是否正常）。如需使用此方式测试以太网须修改external/rk-pcba-test/lan_test.c，将c文件中的“use_ping = 1;”注释

掉即可。

If inet6 website is detected, it is considered the Ethernet is normal. (this method cannot confirm whether the network communication is normal or not). If need to use this method to test Ethernet, you need to modify external/rk-pcba-test/lan_test.c. Just comment out "use_ping = 1;" in the c file.

- 耳机喇叭 (codec) 测试 Earphone speaker (codec) test

Codec测试有两种模式：边录边放、先放后录。先放后录模式，测试效率相对低，使用喇叭时不会有啸叫，可在使用喇叭时选择此模式。边录边放模式，测试效率高，使用喇叭时会有啸叫，可在使用耳机时选择此模式。测试后的录音音量根据实际输入变化，范围从0-100%：“录音音量：[25%]”（模式配置见本文配置说明）。

Codec test has two modes: playing while recording, playing before recording. The test efficiency of playing before recording mode is relatively low, and there is no noise when using speaker. You can select this mode when using speaker. The test efficiency of playing while recording mode is high, but there is noise when using speaker. You can select this mode when using earphone. The recording volume after testing changes according to the actual input, with range from 0 to 100%: "recording volume: [25%]" (mode configuration refers to the configuration instruction in this document).

- 蓝牙 (bluetooth) 测试 Bluetooth test

蓝牙测试为自动测试项，开启检测后系统自动进行检测，根据检测结果显示成功或失败。

The Bluetooth test is the automatic test item. When the detection is enabled, the system will automatically detect and display PASS or FAIL according to the test result.

蓝牙测试默认使用的端口是/dev/ttyS0,如果实际使用的端口不是ttyS0,需要修改配置文件: hardware/broadcom/libbt/conf/rockchip/rksdk/bt_vendor.conf,将bt_vendor.conf中的UartPort修改成实际使用的端口。

The Bluetooth test uses the port /dev/ttyS0 by default. If the actually used port is not ttyS0, need to modify the configuration file:

hardware/broadcom/libbt/conf/rockchip/rksdk/bt_vendor.conf, changing UartPort in bt_vendor.conf to the actually used port.

手动测试项 Manual test item

- SD卡 (sdcard) 测试 SD card test

插入SD卡，系统会进行自动检测，检测后屏幕显示SD卡是否通过，如通过，则会显示SD卡容量信息。

After SD card is inserted, the system will execute automatic detection. After detection the screen will display whether SD card is passed or not. If passed, it will display the capacity information of SD card.

SD card必须为FAT32格式，不支持其他格式！整个卡只能包含一个分区。如果不符合要求，请通过格式化来格式成标准格式。

SD card must be FAT32 format. Other format is not supported! The whole card can only have one partition. If the card doesn't meet the requirement, please format it to the standard format by formatting.

- USB HOST测试 USB HOST test

USB HOST常用U盘进行测试，插入U盘后会自动检测，检测结束后显示是否通过，并显示U盘容量信息。

USB HOST usually uses U disk to test. After inserting U disk, it will automatically detect. After detection it will display whether pass or not, and display the capacity information of U disk.

与SD卡类似，U盘必须为FAT32格式，不支持其他格式！整个卡只能包含一个分区。如果不符合要求，请通过格式化来格式成标准格式。

Similar as SD card, U disk must be FAT32 format. Other format is not supported! The whole card can only have one partition. If it doesn't meet the requirement, please format it to the standard format by formatting.

配置文件说明 Configuration file instruction

PCBA所有的测试项目通过配置脚本 `test_config.cfg` 来配置，位于 `"Android SDK"/bootable/recovery/pcba_core/res/test_config.cfg`，用户可以根据项目的硬件配置来配置 `test_config.cfg` 文件，决定要对哪些模块进行测试，以及给自己的测试程序传递相关的参数。

PCBA all test items are configured through the configuration script `test_config.cfg`, which is in `"Android SDK"/bootable/recovery/pcba_core/res/test_config.cfg`. Users can configure `test_config.cfg` file according to the project hardware configuration, determine which modules to test and transfer relative parameters for the test program.

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，其他编译格式可能会导致未知错误。

The script uses ini file format, which consists of segment, key and value. Generally a segment represents a module configuration. Currently the configuration file is required to use UTF-8 encoding, and other compiling format may cause unknown error.

1.3.1 配置文件模板 Configuration file example

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

- Example
Example表示一个配置模块的名称，如果是cfg文件中自带的模块名称，则不能改动，否则会导致某个测试项不被测试系统启动。
Example represents a configuration module name. If it is the module name in the cfg file, it cannot be changed, otherwise it will cause some test item not to be started by test system.
 - display_name
display_name表示该测试模块在屏幕上显示的名称，可以根据自己的需要修改，该名称最长为64字节，如果为空，则测试程序不会运行。
display_name represents the name of the test module displayed on the screen. It can be changed according to your requirement. The name can be up to 64 bytes, and if it is null, the test program will not run.
 - activated
activated表示是否测试该模块，0：不测试该模块，1：测试该模块。
Activated represents whether to test this module or not, 0: not to test this module, 1: test this module.
 - program
该键值目前没用到，可以不用配置。
Currently it is not used. No need to configure.
 - category
category 表示测试方式，0：自动测试，1：手动测试。
Category represents test method, 0: automatic test, 1: manual test.
- 示例：
Example:

```
[Key] //按键测试 key test
display_name= "key"
activated = 1 //测试该项目 test this item
program = "keytester"
category = 1 //手动测试 manual test
```

部分测试项配置说明 Some test item configuration instruction

- 音频测试 Audio test

```
[Codec]
display_name= "Codec"
activated = 1 //测试该项目 test this item
program = "case1" //case1, case2
category = 1 //手动测试 manual test
run_type = 1
delay = 5
volume = 40
```

Case1 :

先放后录模式，测试效率相对低，使用喇叭时不会有啸叫，可在使用喇叭时选择此模式。

Playing before recording mode, the test efficiency is relatively low, there is no noise when using speaker, you can select this mode when using speaker.

Case2 :

边录边放模式，测试效率高，使用喇叭时会有啸叫，可在使用耳机时选择此模式。

Recording while playing mode, the test efficiency is high, there is noise when using speaker, you can select this mode when using earphone.

- 蓝牙测试 Bluetooth test

```
[bluetooth]
display_name= "bluetooth"
activated = 1
program =
category =
run_type = 1
chip_type = "" ; rk903, mt6622, rda587x, rda5990, rtk8723as
chip_type 为选择相应的BT芯片型号，默认为空，也就是不测试BT，Android 5.0后不需要选择，系统会自动识别。
chip_type is to select the corresponding BT chipset type. It is null by default, that is, not to test BT. There is no need to select after Android5.0, and the system will automatically identify it.
```

- 传感器测试 Sensor test

```
[allsensor]
display_name= "allsensor"
activated = 0
program = ""
category = 0
run_type = 1
sensor_type = 39
```

sensor_type 会选择相应的传感器种类。1: GSENSOR, 2: GYROSCOPE, 4: COMPASS, 8: LSENSOR, 16: PSENSOR, 32: GSENSOR_CALIBRATE。如果需要测试多种传感器, 只需把传感器种类对应的数值相加即可。默认的sensor_type是39, 即 GSENSOR_CALIBRATE(32)+COMPASS(4)+GYROSCOPE(2)+GSENSOR(1)。

sensor_type is the selected sensor type. 1: GSENSOR,2: GYROSCOPE, 4: COMPASS, 8: LSENSOR, 16: PSENSOR, 32: GSENSOR_CALIBRATE. If need to test multiple sensors, only need to calculate the sum of the values corresponding to the sensor types. The default value of sensor_type is 39, that is, GSENSOR_CALIBRATE(32)+COMPASS(4)+GYROSCOPE(2)+GSENSOR(1).

- Camera测试 Camera test

```
[camera]
display_name = "camera"
activated    = 1
category     = 0
program      = ""
number       = 2
```

number表示测试的camera个数, 最大支持测试2个camera.

Number represents the quantity of the camera to be tested. It supports 2 cameras at most.

如需自行添加测试模组, 请参考文档《RK_PCBA_Camera移植说明v1.0.doc》。

If need to add test module by yourself, please refer to the document 《RK_PCBA_Camera移植说明v1.0.doc》.

配置脚本参数扩展 Configuration script parameter extension

test_config.cfg 配置脚本中各测试项的参数可以进一步扩展, 如果某个模块需要通过配置脚本传递相关参数, 可进行如下扩展:

The test item parameters in the configuration script test_config.cfg can be extended further. If some module needs to transfer relative parameter through configuration script, you can do the extension as follows:

```
[example]
display_name= "Example"
activated   = 1
program    = "example.sh"
module_args = "xxx"
```

在具体的测试程序中, 可以通过 script_fetch api获得设置的相关参数值:

In the specific test program, you can obtain the relative set parameter values through script_fetch api:

```
int script_fetch(char *main_name, char *sub_name, int value[], int count);
```

main_name: 测试模块的名称, 在test_config.cfg文件中测试项, 示例中的“example”

main_name: the name of the test module, test item in the file of test_config.cfg, “example” in the example

sub_name:键值, 比如activated、display_name、module_args等等。

sub_name: key value, such as activated, display_name, module_args and so on.

取值示例:

Example to fetch the value:

```

if(script_fetch("example", "module_args", (int *)des, 8) == 0)
{
    printf("module_args value is: %s\n",des);
}

```

测试项扩展 Test item extension

该PCBA程序允许用户扩展自己的测试样例，如果因为项目需要，用到了该测试程序中目前还未支持到的模块，可以自己添加测试程序，然后集成到测试框架中。

This PCBA program allows users to extend their own test example. If need to use the module currently unsupported due to the project requirement, you can add the test program by yourself and then integrate into the test framework.

集成方法如下：

The integration method is as follows:

(1) 先写好自己的测试程序和头文件。测试程序要封装成 `void * xxxx_test(void *argv, display_callback *hook)` 格式的接口。

Prepare your own test program and header file first. The test program should be packaged as the interface with `void * xxxx_test(void *argv, display_callback *hook)` format.

(2) 确定该测试项为手动测试项或者是自动测试项，并在 `test_config.cfg` 里面加入想要的配置。Confirm this test item is manual test item or automatic test item, and add the needed configuration in `test_config.cfg`.

(3) `rkfactory_test.cpp` 中注册自己的测试代码。

Register your own test code in `rkfactory_test.cpp`.

```

/* new_test.h */
#ifndef _NEW_TEST_H_
#define _NEW_TEST_H_
#include "display_callback.h"
void *new_test(void *argv, display_callback *hook);
#endif

/* new_test.cpp */
void *new_test(void *argv, display_callback *hook) {
    struct testcase_info *tc_info = (struct testcase_info *)argv;
    char msg[50];
    snprintf(msg, sizeof(msg), "%s:[%s..]", "New Test", PCBA_TESTING);
    hook->handle_refresh_screen(tc_info->y, msg);
    if (/* pass or fail */) {
        hook->handle_refresh_screen_hl(tc_info->y, msg, highlight);
    }
}

/* rkfactory_test.cpp */
#include "new_test.h"
int start_test_pthread(struct testcase_info *tc_info) {
    .....
    if (!strcmp(tc_info->base_info->name, "new")) {
        init_title_lines_for_testcase(tc_info->base_info->name, tc_info);
        std::thread *temp = new std::thread(&new_test, tc_info,
get_display_hook());
        if (!temp) {
            printf("create %s test thread error/n", tc_info->base_info->name);
        }
    }
}

```



```
.....  
}
```

接口说明 Interfac introduction

```
/**  
 * msg_index: 显示词条的索引，通常将创建时当前的位置传入即可；The index of display  
 message in screen,  
 * please set it with the location in message array.  
 * msg: 显示词条的内容，std::string类型，内容自己填写；The message you want to  
 display in the screen,  
 * set this message with std::string type.  
 */  
handle_refresh_screen(int msg_index, string msg);  
  
/**  
 * highlight: 显示词条是否高亮，默认高亮为红色，通常在测试fail时传入true即可；  
 * whether to highlight the message or not, with red by default, you can set it  
 with true when the test fails.  
 */  
handle_refresh_screen_hl(int msg_index, string msg, boolean highlight);
```

其中 `new_test` 为模块中的测试函数。

`new_test` is the test function of the module.

屏幕旋转配置 Panel rotation configuration

PCBA测试支持屏幕旋转功能，根据需求旋转0°、90°、180°、270°以支持不同摆放的横/竖屏显示，其中0°、180°用于支持竖屏，90°、270°用于支持横屏，默认为0°。

PCBA test supports panel rotation function. Rotate 0°, 90°, 180° or 270° based on the requirement to support different landscape/portrait display. 0°, and 180° are used to support portrait, while 90° and 270° are used to support landscape. The default is 0°.

旋转屏幕的配置在 `device/rockchip/common/Boardconfig.mk` 中

The configuration of panel rotation is in `device/rockchip/common/Boardconfig.mk`

```
#rotate screen to 0, 90, 180, 270  
#0: ROTATION_NONE ORIENTATION_0 : 0  
#90: ROTATION_RIGHT ORIENTATION_90 : 90  
#180: ROTATION_DOWN ORIENTATION_180: 180  
#270: ROTATION_LEFT ORIENTATION_270: 270  
# For Recovery Rotation  
TARGET_RECOVERY_DEFAULT_ROTATION ?= ROTATION_NONE
```

固件编译配置打包 Image compiling configuration package

PCBA测试程序位于 `"Android SDK"/bootable/recovery/pcba_core` 目录下，编译会静态链接到 `recovery` 中。 `pcba_core/res` 下的相关文件在编译的时候会被自动拷贝到 `recovery` 的 `pcba` 目录下。

PCBA test program is in the directory of `"Android SDK"/bootable/recovery/pcba_core`.

Compiling will static linked `pcba_core` into `recovery`. Relative files in `pcba_core/res` will be automatically copied to `pcba` directory of `recovery` during compiling.

PCBA程序运行于 `Recovery` 系统中，具体测试流程为：开机进入 `Recovery`，启动 `PCBA` 测试程序进行各项功能测试。

PCBA program is run in `Recovery` system. The detailed test process is: enter `Recovery` after power up, start `PCBA` test program to do the function tests.

