

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

(技术部, 第二系统产品部)

(Technical Department, R & D Dept. II)

文件状态: Status: [ ] 正在修改 [ ] Modifying [ <input checked="" type="checkbox"/> ] 正式发布 [ <input checked="" type="checkbox"/> ] Released	当前版本: Current Version:	V1.2
	作 者: Author:	李东强、刘益星、林龙坚 Li Dongqiang, Liu Yixing, Lin Longjian
	完成日期: Finish Date:	2018-05-09
	审 核: Auditor:	
	完成日期: Finish Date:	

福州瑞芯微电子股份有限公司

Fuzhou Rockchip Electronics Co., Ltd

(版本所有, 翻版必究)

(All rights reserved)

## 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
V1.0	李东强 Li Dongqiang	2017-02-15	发布初始版本 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	

## 目 录 Contents

前 言 Preface.....	1
1 PCBA .....	2
1.1 概述 Overview .....	2
1.2 测试项说明 Test item introduction.....	3
1.2.1 自动测试项 Automatic test item .....	4
DDR/EMMC 容量检测 DDR/EMMC capability detection .....	4
无线网络（wifi 测试） Wireless network (Wi-Fi test) .....	4
以太网测试 Ethernet test .....	4
耳机喇叭（codec）测试 Earphone speaker (codec) test.....	5
蓝牙（bluetooth）测试 Bluetooth test.....	6
1.2.2 手动测试项 Manual test item .....	6
按键（KEY）测试 KEY test .....	6
屏幕坏点（LCD）测试 LCD defects test.....	6
SD 卡（sdcard）测试 SD card test .....	7
USB HOST 测试 USB HOST test .....	7
1.3 配置文件说明 Configuration file instruction .....	8
1.3.1 配置文件模板 Configuration file example.....	8
1.3.2 部分测试项配置说明 Some test item configuration instruction .....	10
1.3.3 配置脚本参数扩展 Configuration script parameter extension .....	13
1.4 测试项扩展 Test item extension .....	14
1.5 字体配置 Font configuration.....	15
1.6 屏幕旋转配置 Panel rotation configuration .....	16
1.7 固件编译配置打包 Image compiling configuration package.....	16

## 前言 Preface

### 概述 Overview

本文档主要介绍 Rockchip PCBA 测试工具的使用方法和开发指南。通过本文档可快速了解 PCBA 工具的使用，以及 PCBA 测试功能的扩展。

This document mainly introduces the usage and development guide for Rockchip PCBA test tool. Customers can quickly understand the usage of PCBA tool and the extension of PCBA test functions through this document.

### 产品版本 Product version

芯片名称 Chipset name	内核版本 Kernel version	Android 版本 Android version
RK3399	Linux4.4	Android7.1
RK3126C	Linux4.4	Android8.1
RK3368H	Linux4.4	Android8.1

### 读者对象 Object

本文档（本指南）主要适用于以下工程师：

This document (guide) is mainly suitable for below engineers:

- 技术支持工程师  
Field application engineers
- 软件开发工程师  
Software development engineers

# 1 PCBA

## 1.1 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）、USB 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 item and manual test item. 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.



图 1-1 PCBA 测试界面

Picture 1-1 PCBA test interface

## 1.2 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”，“绿”三种颜色表示不同的测试状态。

There are three colors “red”, “yellow” and “green” 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

Yellow: the item waiting for test or under testing

绿色：测试通过项

Green: test passed

红色：测试未通过项

Red: test failed

具体测试项说明如下。

The details of the test items are introduced as below.

### 1.2.1 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 automatically displays DDR and EMMC size for 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)

Wifi 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 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 detect whether the network card is started or not to judge. There will be a inet6 website if the network card is started, 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). 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.

### 1.2.2 手动测试项 **Manual test item**

- 按键 (KEY) 测试 **KEY test**

打开按键测试项后屏幕显示按键测试栏，手动点击设备按键后，屏幕会显示相应按键信息，如“ESC、POWER、MENU、HOME 等”。测试人员根据所按按键判断屏幕所显示按键信息是否正确判断按键是否正常。

After open the KEY test item, the screen displays the key test bar, manually click the device KEY, and the screen will display the corresponding KEY information, such as "ESC, POWER, MENU, HOME, etc.". Test engineer judges whether the KEY is normal or not by checking if the KEY information displayed on the screen is correct or not.

- 屏幕坏点 (LCD) 测试 **LCD defects test**

LCD 坏点测试为手动测试项，打开测试选项后屏幕的右下方显示红、绿、蓝三原色的方块。测试人员连续“N 次”按“启动按键”后进入坏点测试模式，坏点测试整屏依次显示“红、绿、蓝、黑、白”画面并通过“切换按键”进行切换。测试人员根据显示画面判断 LCD 显示是否正常、是否有坏

点。

LCD defects test is manual test item. After open the test item, the red, green and blue squares will be displayed in the lower right corner of the screen. After test engineer continuously presses "Start button" "N" times, it will enter the defect test mode. The defect test uses "red, green, blue, black, white" patterns to test the whole screen and switch the pattern through "switch button". Test engineer judges whether LCD display is normal and there is defect point according to the display.

"启动按键"及"N 次"以及"切换按键"的设定见"配置文件"中的 lcd 配置。

The settings of "Start button" , "N times" and "Switch button" refer to LCD configuration in the configuration file.

#### ● **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. Do not support other format! The card can have only 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. Do not support other format! The whole card can have only one partition. If it doesn't meet the requirement, please format it to the standard format by formatting.

### 1.3 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 src”/external/rk-pcba-test/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 src”/external/rk-pcba-test/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 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, 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
```

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

#### 屏幕测试 **LCD test**

[Lcd]

```
display_name= "lcd"

activated    = 1           //测试该项 test this item

program      = "lcd test"

category     = 1           //手动测试 manual test

run_type     = 1

start_key    = "KEY_BACK" //启动测试的按键 the key to start testing

key_times    = 3           //连续按启动键的次数 the times of continuously pressing
the start key

all_key_change = 1         //进入测试后是否全部按键可切换画面 whether all the keys
can change the screen after entering the test
```

连续按 key\_times 次 start\_key 按键将进入测试模式，进入测试模式后通过 start\_key 进行画面切换，若需要通过任意键进行画面切换则令 all\_key\_change=1，否则 all\_key\_change 设为 0；测试结束后回到主界面，显示“屏幕：[已测试]”。

Continuously press key\_times times start\_key will enter the test mode. After entering test mode, change the screen through start\_key. If need to change the screen through any key, set all\_key\_change=1, otherwise set all\_key\_change as 0. After testing, it will return to the main interface and display “LCD: [tested]”.

start\_key 常用按键： "KEY\_BACK" ， "KEY\_VOLUMEUP" ， "KEY\_VOLUMEDOWN""KEY\_HOME""KEY\_MENU" ， "KEY\_ENTER" ， "KEY\_ALL" 等。

(start\_key 设为 KEY\_ALL 或无效键值则任一键连续按 key\_times 次进入测试模式)

start\_key usually uses "KEY\_BACK", "KEY\_VOLUMEUP", "KEY\_VOLUMEDOWN", "KEY\_HOME", "KEY\_MENU", "KEY\_ENTER", "KEY\_ALL" and so on. (set start\_key as KEY\_ALL or invalid key value, and then continuously press any key key\_times times will enter test mode)

## 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**

[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 the selected 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.

### 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 corresponding 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 camera 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 yourself, please refer to the document 《RK\_PCBA\_Camera\_移植说明\_v1.0.doc》.

### 1.3.3 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 below:

[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);
}
```

## 1.4 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 project requirement, you can add test program by yourself and then integrate into the test framework.

集成方法如下:

The integration method is as below:

- (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) 格式的接口。

Prepare your own test program and head file first. The test program should be packaged as the interface with void \* xxxx\_test(void \*argv) 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) 在 pretest.c 中注册自己的测试代码。

Register your own test code in pretest.c.

以 Lcd 为例:

Take LCD as example:

Pretest.c 头部包含头文件并定义线程:

The header of Pretest.c includes the header file and defines the thread:

```
#include "screen_test.h"

pthread_t screen_tid;

Pretest.c 文件中 start_test_pthread()函数新增创建线程代码:

start_test_pthread() function in Pretest.c file adds newly created thread code:

if (!strcmp(tc_info->base_info->name, "Lcd")) {

    err = pthread_create(&screen_tid, NULL, screen_test, tc_info);

    if (err != 0) {

        printf("create screen test thread error: %s/n",

               strerror(err));

        return -1;

    }

}
```

其中 screen\_test 为模块中的测试函数。

screen\_test is the test function of the module.

## 1.5 字体配置 Font configuration

说明: PCBA 2.0 以后的版本增加了对中文的支持, 并可以支持多种字体大小的配置, 包括 18\*18, 20\*20, 24\*24, 28\*28, 32\*32, 36\*36, 可以通过修改 minuitwrp/graphics.c 包含

的头文件来修改使用不同大小的字库。(输出到屏幕的中文必须是 UTF-8 编码格式)

Description: PCBA 2.0 higher versions add Chinese support, and can support the configuration of multiple font size, including 18\*18, 20\*20, 24\*24, 28\*28, 32\*32 and 36\*36. You can change to use the font library with different sizes by modifying the header file contained in minuitwrp/graphics.c. (Chinese output to the screen must be UTF-8 encoding format)

## 1.6 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 rk-pcba-test 根目录下的 Android.mk 中，打开相应角度的配置后进行编译，如已经编译过需强制重编。mmm external/rk-pcba-test/ -B

The configuration of panel rotation is in Android.mk under the root directory of rk-pcba-test. Compile after opening the configuration of the corresponding angle. Forcedly re-compile if it is already compiled. mmm external/rk-pcba-test/ -B

```
#===== rotate screen 0, 90, 180, 270 degree=====
# warning: If changed please force rebuild the target -B
#ROTATE_SCREEN := 0
#ROTATE_SCREEN := 90
#ROTATE_SCREEN := 180
ROTATE_SCREEN := 270
```

## 1.7 固件编译配置打包 Image compiling configuration package

PCBA 测试程序位于 Android 源码/external/rk-pcba-test 目录下，编译会生成 pcba\_core 可执行文件。pcba\_core 和 rk-pcab-test/res 下的相关文件在编译的时候会被自动拷贝到 recovery 的 sbin 目录下。

PCBA test program is in the directory of Android 源码/external/rk-pcba-test.

Compiling will generate pcba\_core executable file. Relative files in pcba\_core and rk-pcab-test/res will be automatically copied to sbin 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 test.

## 1、内核配置(7.1 系统跳过此步骤) Kernel configuration(skip this step for 7.1)

PCBA 固件无需 selinux，请关闭，对于正常固件请打开：

PCBA image doesn't need selinux. Please close it. For normal image, please open it:

```
----- NSA SELinux Support -----
CONFIG_SECURITY_SELINUX:
  This selects NSA Security-Enhanced Linux (SELinux).
  You will also need a policy configuration and a labeled filesystem.
  If you are unsure how to answer this question, answer N.

  Symbol: SECURITY_SELINUX [=n]
  Type : boolean
  Prompt: NSA SELinux Support
  Location:
    -> Security options
  Defined at security/selinux/Kconfig:1
  Depends on: SECURITY_NETWORK [=y] && AUDIT [=y] && NET [=y] && INET [=y]
  Selects: NETWORK_SECMARK [=y]
```

重新编译 kernel，然后更新 recovery 的 kernel。执行下面命令即可：cp kernel/arch/arm64/boot/Image out/target/product/rk3399/kernel （红色的目录需要换成你自己的编译路径。）

Re-compile kernel, and then update recovery kernel. Just need to execute the following command: cp kernel/arch/arm64/boot/Image out/target/product/rk3399/kernel (the red directory should be replaced with your own compiling path).

## 2、打开 device/rockchip/common/BoardConfig.mk 文件中的：Open device/rockchip/common/BoardConfig.mk file:

修改 BoardConfig.mk 文件中的 TARGET\_ROCKCHIP\_PCBATEST?=true

Modify TARGET\_ROCKCHIP\_PCBATEST?=true in BoardConfig.mk file

make installclean

make

rm -rf out/target/product/rk3399/recovery 删除 out 下的 recovery 目录 delete

the recovery directory under out

make recoveryimage

./mkimage

### 3、编译 Compiling

执行 make 命令，编译完后请确保下面目录是否有以下文件(注意:红色目录是你们自己的目录):

Execute make command, and confirm if there is the following file in below directory after compiling (note: the red directory is your own directory):

ls out/target/product/**rk3399**/recovery/root/sbin/目录:

```
adbd      e2fsck    healthd   iwlist    pcba_core  resize2fs  sdtool    ueventd
busybox   eio       insmod    mkdosfs   recovery   rmmmod     sh         watchdogd
```

ls out/target/product/**rk3399**/recovery/root/system/lib

```
hw          libcrypto.so  libgccdemangle.so  libm.so      libstdc++.so  libusbhost.so  modules
libbacktrace.so  libc.so       libhardware_legacy.so  libnetutils.so  libstlport.so  libutils.so
libbt-vendor-rtl8723bu.so  libcutils.so  libhardware.so      libpower.so     libunwind-pttrace.so  libwpa_client.so
libbt-vendor.so   libdl.so      liblog.so           libselinux.so  libunwind.so    libz.so
```

(如果没有的话请手动在 **external/rk-pcba-test/** 目录执行 **mm -B** ,并重新 make。)

(If no, please manually execute **mm -B** in the directory of **external/rk-pcba-test/**, and make again.)