

Rockchip

PCBA 测试开发指南

发布版本:**0.01**

日期:**2018.05**

前言

概述

本文档主要介绍 Rockchip 处理器内 PCBA 测试工具，使用该工具可以在量产阶段快速识别板端硬件中各个部件功能的好坏，文中通过实例介绍 PCBA 测试工具的使用方法和注意事项。

产品版本

芯片名称	软件版本
适用 RK3308、RK3229、PX3-SE	V1.6

读者对象

本文档（本说明）主要适用于以下工程师：

- 技术支持工程师
- 软件开发工程师

修订记录

日期	版本	作者	修改说明
2018.05	V0.01	Chad.ma	初稿

目录

1 PCBA TEST Tool.....	1
1.1 概述.....	1
1.2 工具安装.....	1
1.3 测试工具介绍.....	2
1.4 配置测试项.....	3
1.5 工具二次开发指引.....	7
2 PCBA 软件设计说明.....	9
2.1 设计框图.....	9
2.2 设计流程.....	10
2.3 PCBA 命令协议格式.....	11
3 PCBA 测试代码框架.....	14
3.1 代码结构.....	14
3.2 代码框架说明.....	15
3.3 代码编译.....	16

插图目录

图 1.1 PCBA Tool 安装.....	1
图 1.2 PCBA Tool 图形界面.....	2
图 1-3 PCBA 测试项配置窗口.....	3
图 1-4 PCBA 配置完成窗口.....	4
图 1-5 PCBA 测试工具配置文件.....	5
图 1-6 PCBA 启动测试.....	6
图 1-7 PCBA 按键测试.....	7
图 1-8 PCBA config.ini 变量说明.....	7
图 1-9 PCBA Chinese.ini 变量说明.....	8
图 2-1 PCBA 设计框图变量说明.....	9
图 2-2 PCBA 测试流程图变量说明.....	10
图 3-1 PCBA 设备端测试代码框架.....	15
图 3-2 PCBA 测试代码编译结果.....	16

表格目录

表 2-1 Command 命令说明..... 11

表 2-2 Response 命令说明..... 11

表 2-3 TYPE 命令参数取值与定义说明..... 11

表 2-4 TEST_ITEM 命令参数取值与定义说明..... 11

表 2-5 CMD/RES 命令参数取值与定义说明..... 12

表 2-6 MSG 命令参数取值与定义说明..... 12

表 2-7 STATUS 命令参数取值与定义说明..... 12

表 2-8 RESULT 命令参数取值与定义说明..... 12

表 2-9 ERR_CODE 命令参数取值与定义说明..... 12

1 PCBA TEST Tool

1.1 概述

PCBA 测试用于帮助在量产的过程中快速地甄别产品功能的好坏，即重点 FCT (Functional Test) 测试，进而提高生产效率。PCBA 测试工具在 Window 系统下开发，仅支持在 Window 系统下运行，配合设备端测试程序，即可以验证所需要重点关注的功能或器件的完整与好坏。

目前测试项包括：SD 卡测试、无线（wifi）测试、蓝牙测试、DDR 测试、环麦测试、USB host 测试、led 灯测试、放音测试、录音测试、按键测试、PDM Mic 测试、Audio Line in 测试、SPDIF IN/OUT 测试等，后续还可以添加扩展的测试项。

这些测试项目包括自动测试项和手动测试项，其中 SD 卡测试、无线（wifi）测试、蓝牙测试、DDR 测试、环麦测试、USB host 测试为自动测试项目；led 灯测试、放音测试、录音测试、按键测试、PDM Mic 测试、Audio Line in 测试、SPDIF IN/OUT 测试为手动测试项目。

自动测试项目无需人工干预测试结束后会直接上报测试结果并显示通过与否，人工测试项目需要人为判断测试项是否正确完成，并给出判断（通过或不通过）。

1.2 测试工具安装

运行 PCBA 工具的安装程序 PCBATool_Setup_xxxx.exe，成功安装后会在指定目录生成可执行程序及各种相关的配置文件和工具中用到的其他程序和库文件。

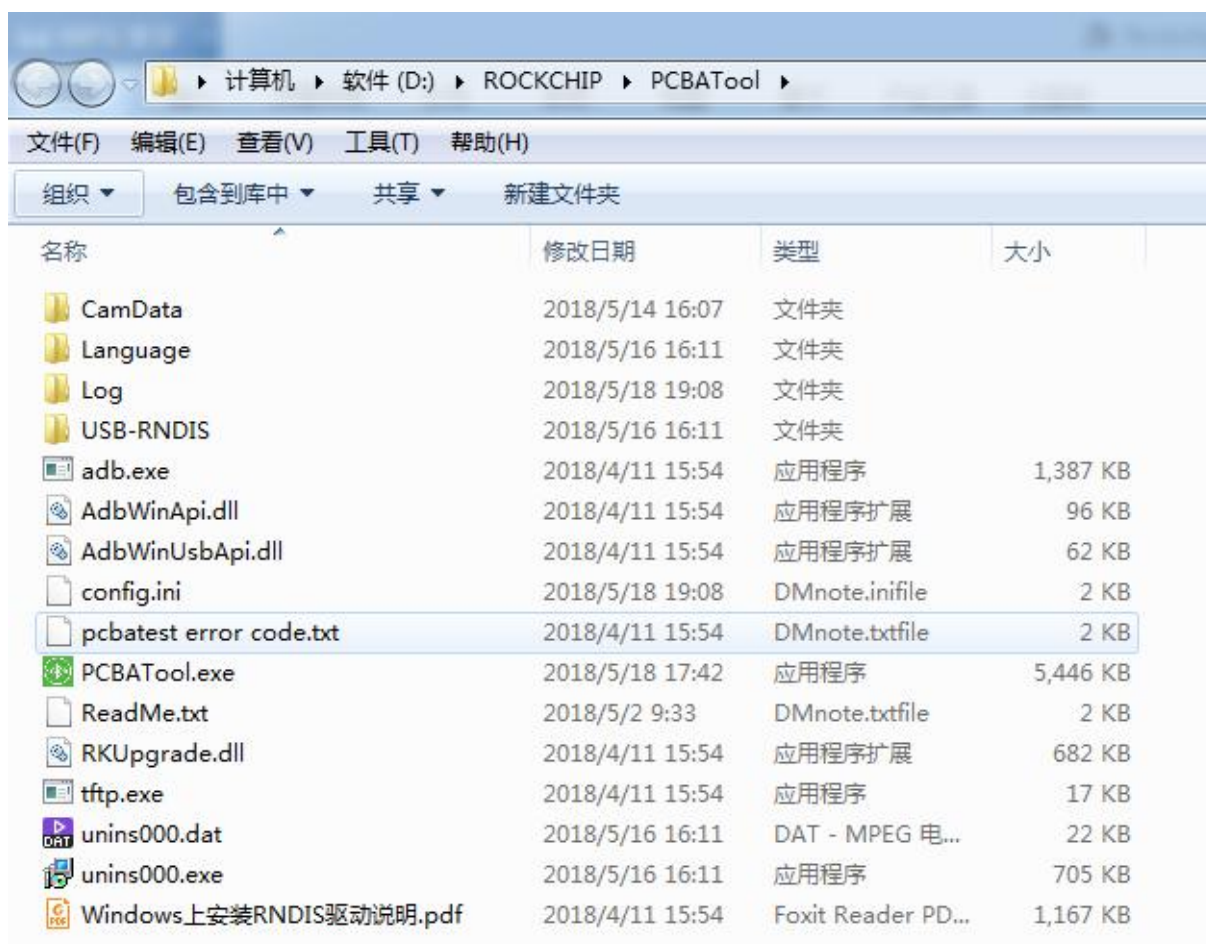


图 1.1 PCBA Tool 安装

其中：

CamData 目录用来保存 camera 测试中抓取的摄像头某一时刻的一帧图片。保存为 bmp 格式。RK3308

未使用。
Language 目录：存放工具语言配置文件。
Log 目录：存放工具测试过程中的各种 log 文件，方便后续查找问题原因。
Config.ini:工具配置文件，记录工具的配置项、语言配置与其他配置的信息。

注意：
如果指定自定义的安装目录，请使用全英文路径，避免工具中使用 adb 工具在非英文路径下执行失败的问题。

1.3 测试工具介绍

图 1-2 为 PCBA 测试工具打开后的主界面：

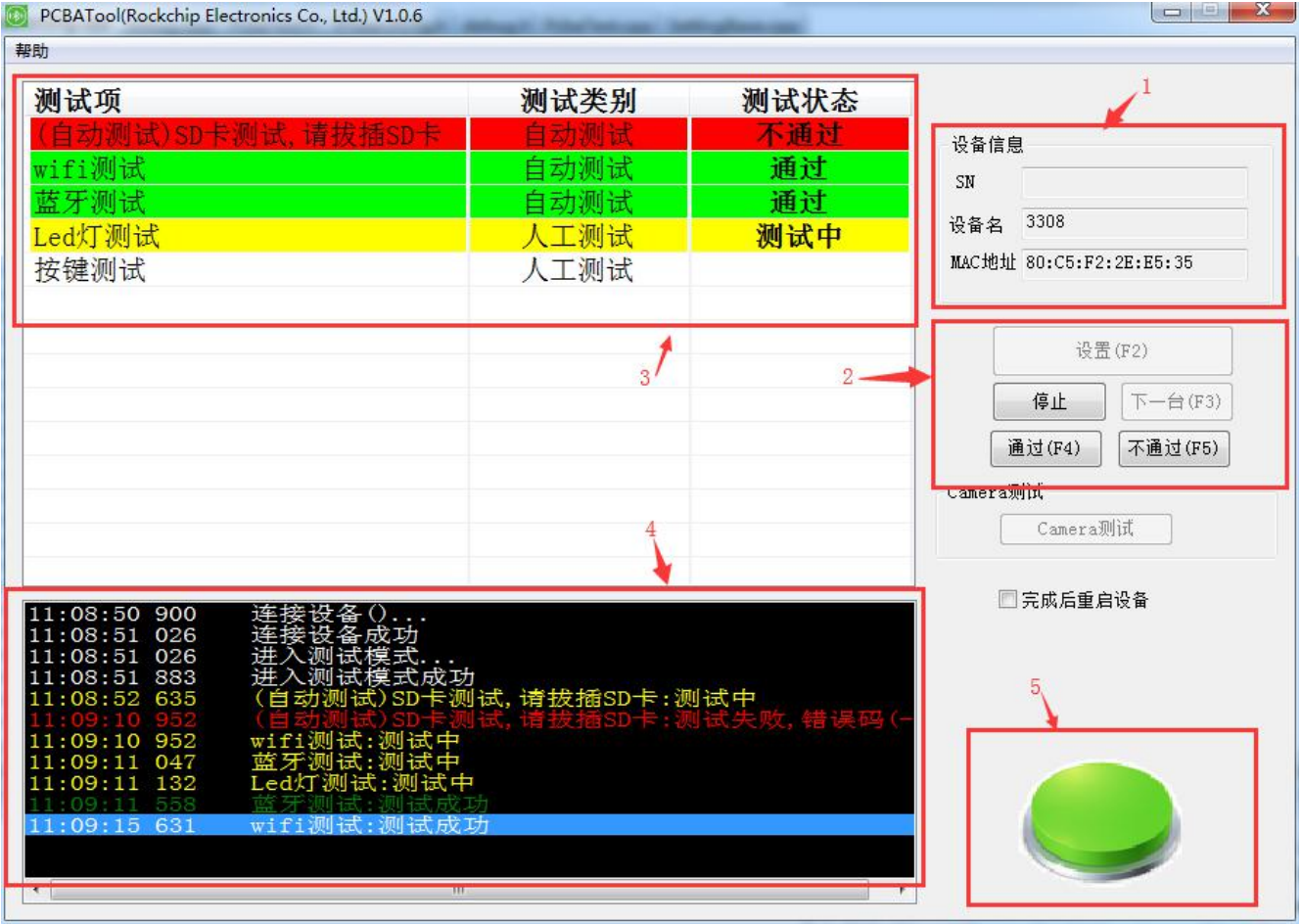


图 1.2 PCBA Tool 图形界面

- 如上图所示，PCBA Tool 主要分为五个部分：
- 第 1 部分为设备信息，测试开始后会获取设备名称及设备的 MAC 地址；
 - 第 2 部分为设置测试项及控制测试启动与停止，以及人工测试项判断通过与否的按钮；
 - 第 3 部分为测试项的显示列表框，将会显示测试类别与测试结果；
 - 第 4 部分为测试各个测试项过程中的信息显示列表框，将显示测试过程及结果；
 - 第 5 部分为设备连接状态，如果工具检测到设备可用（识别为 ADB 设备），显示一个绿色按钮图片，否则显示为橙色按钮图片。

1.4 配置测试项

点击工具主界面的设置按钮或者按 F2 快捷键，会弹出配置项设置对话框，如图 1-2 所示。选中某些测试项目后，点击确定按钮将会在上文介绍的第 3 部分测试项列表框中显示，并根据各个测试项定义好的测试类型（自动或人工测试）显示具体的测试类型。

目前 3308 平台支持的配置项为：SD 卡测试、无线（wifi）测试、蓝牙测试、DDR 测试、环麦测试、USB host 测试、led 灯测试、放音测试、录音测试、按键测试、PDM Mic 测试、Audio Line in 测试、SPDIF IN/OUT 测试等，后续还可以添加扩展的测试项。

其他 Rockchip 平台的芯片可以按照需求及硬件开发板情况去配置相应的测试项目。



图 1-3 PCBA 测试项配置窗口

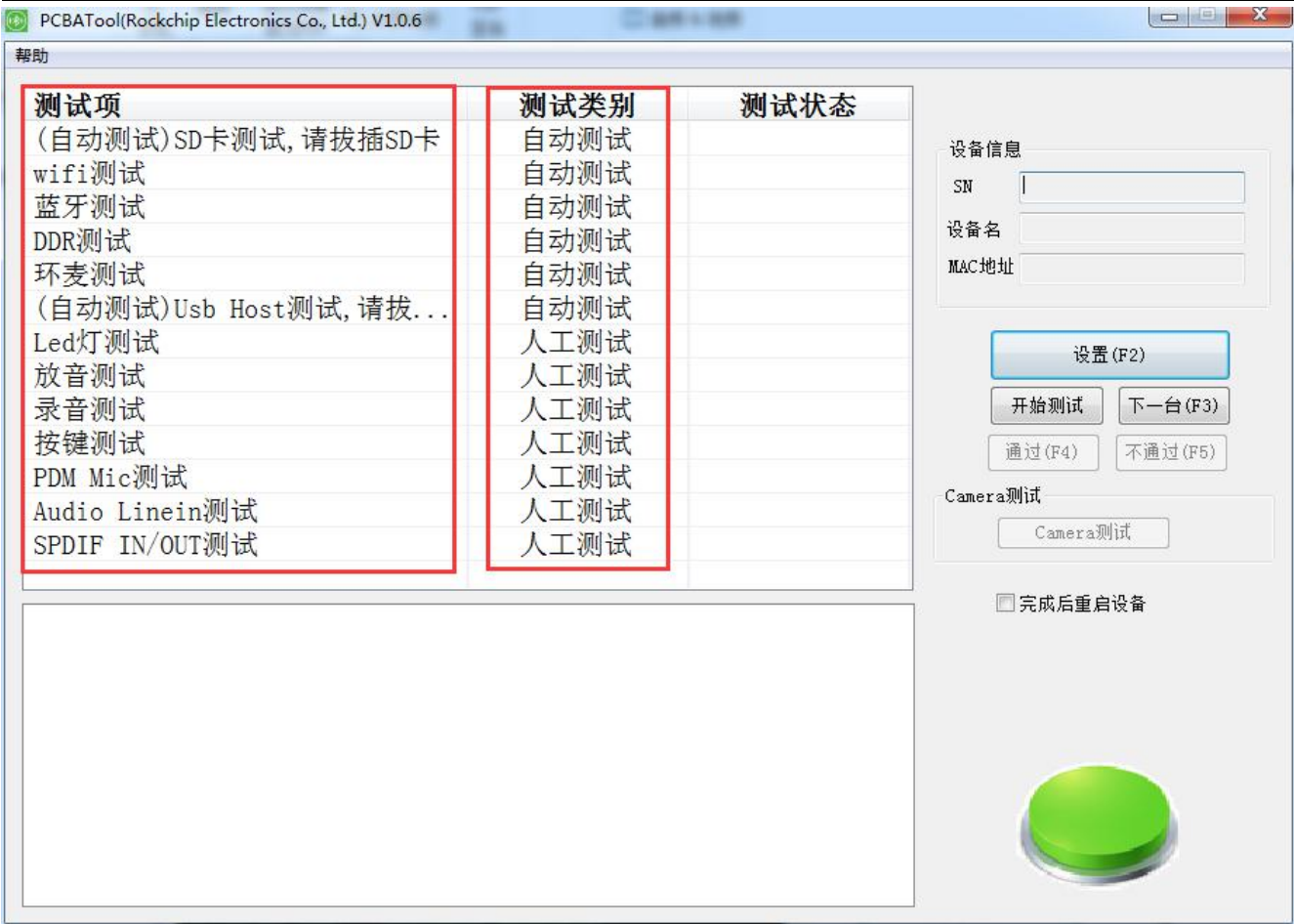


图 1- 4 PCBA 配置完成窗口

1.4.1 测试项说明

- 1、 WiFi 测试：自动测试项，测试 WiFi 时设备会自动扫描周围的 WiFi 热点，根据扫描的结果判定成功与否。
- 2、 蓝牙测试：自动测试项，测试蓝牙时设备会先自动关闭蓝牙设备，然后重新开启蓝牙，加载固件后激活蓝牙，根据是否成功开启蓝牙控制接口 hci0，是否有蓝牙硬件地址，判定成功与否。
- 3、 ddr 测试：自动测试项，测试 ddr 时，会自动读取设备的 ddr 容量，根据是否读取正确判定成功与否。
- 4、 SD 卡测试：自动项测试，测试 SD 卡时会运行测试脚本，插入 SD 卡，将测试设备节点是否存在，若存在，自动挂载 SD 卡到指定文件目录下，SD 卡容量也会记录下来。60s 超时未检测到设备节点将自动报错退出。
- 5、 USB Host 测试：自动项测试，测试 USB Host 功能时会运行测试脚本，插入 USB host 设备，测试设备节点是否存在，若存在，自动挂载 host 设备到指定文件目录下，设备容量也会记录下来。60s 超时未检测到设备节点将自动报错退出。
- 6、 LED 测试：人工测试项，测试 LED 时，设备第一轮亮红灯，第二轮亮绿灯，第三轮亮蓝灯，第四轮，三种颜色合成白灯；需要人工根据每种灯是否正常亮起，决定测试成功或者失败，点击通过或者不通过按键跳入下一项测试。
- 7、 放音测试：人工测试项，测试放音功能时，设备会播放存放在设备指定位置的音频数据。需要人工根据声音情况来决定测试成功或者失败，点击通过或者不通过按键跳入下一项测试。
- 8、 录音测试：人工测试项，测试录音功能时，设备会循环录制当前麦克风采集的一段音频，再把录到的声音通过设备播放出来。需要人工根据播放出来的声音情况决定测试成功或者失败，点击通过或者不通过按键跳入下一项测试。

9、 按键测试：人工测试项，测试按键时，需要人工去按下按键，如果设备检测到按键按下，工具会提示某一名称的按键按下，如果所有板级定义的可测按键均已按下，工具会弹窗提醒，通过后即为测试成功，否则测试失败。测试结果需要人工判断，点击通过或者不通过按键跳入下一项测试。

10、 PDM Mic 测试：人工测试项，测试 PDM Mic 功能是否正常。需要人工根据播放出来的声音情况决定测试成功或者失败，点击通过或者不通过按键跳入下一项测试。

11、 Audio Line in 测试：人工测试项，板级插入 audio line in 设备后，播放音乐出来，需要人工根据喇叭播放的声音来决定测试成功或失败，点击通过或者不通过按键跳入下一项测试。

12、 SPDIF IN/OUT 测试：人工测试项

1.4.2 测试流程说明

Step1: 正确安装 PCBA 测试工具后，启动工具，如果工具右下角显示绿色按钮图标，表示已经正确扫描到已连接到 PC 的 ADB 设备。如果设备连接不成功，工具右下角将显示橙色按钮图标，表示当前设备不可用。

Step2: 点击【设置】按钮，打开测试项的选择窗口，选择需要测试的测试项。【确定】后，工具主窗口测试列表框会显示即将要测试的项目列表。该次选择的配置也会保存在 PCBA 测试工具的安装目录下 config.ini 文件中，测试项的配置文件如下图 1-5 所示。红框标识出的标识该测试项被选中，下次启动工具后将默认加载。

```
1 [CONFIG]
2 AllPassReboot=0
3 AudioLineinName=echo_audio_linein_test
4 AudioLineinTest=0
5 BtName=echo_bt_test
6 BtTest=1
7 CameraName=echo_camera_test
8 CameraTest=0
9 DdrName=echo_ddr_test
10 DdrTest=1
11 Debug=0
12 Dev_Plateform=GVA
13 EmmcName=echo_emmc_test
14 EmmcTest=0
15 HdmiName=hdmi_test
16 HdmiTest=0
17 InterphoneName=echo_audio_play_test
18 InterphoneTest=1
19 IrcutName=ircut_test
20 IrcutTest=0
21 KeyName=echo_key_test
22 KeyRGBLightName=echo_key_rgblight_test
23 KeyRGBLightTest=0
24 KeyTest=1
```

图 1-5 PCBA 测试工具配置文件

Step3: 点击【开始测试】按钮后，首先同时测试所有的自动测试项以及第一个人工测试项；自动测试项无需点击【通过】或者【不通过】。如下图 1-6 所示。

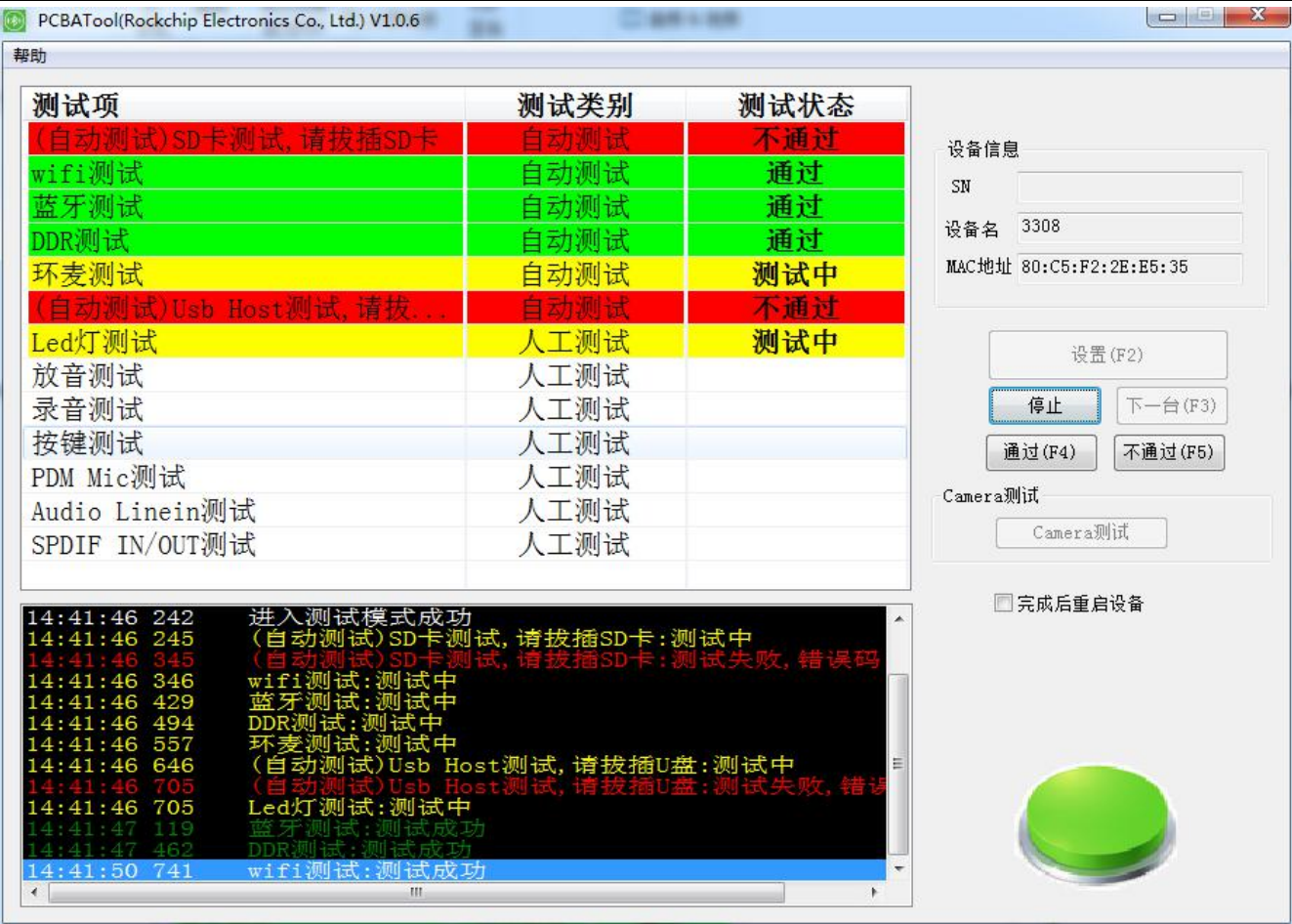


图 1-6 PCBA 启动测试

Step4: 人工测试项都是单项测试，每一项测试完成后都必须点击通过或者不通过来结束当前测试项，然后进入下一个测试项测试。除按键测试会在信息提示框中显示出来按键具体键名（需设备端程序指定），测试完成后会弹窗提示外，其他的测试项均需要人工判断通过或者不通过。

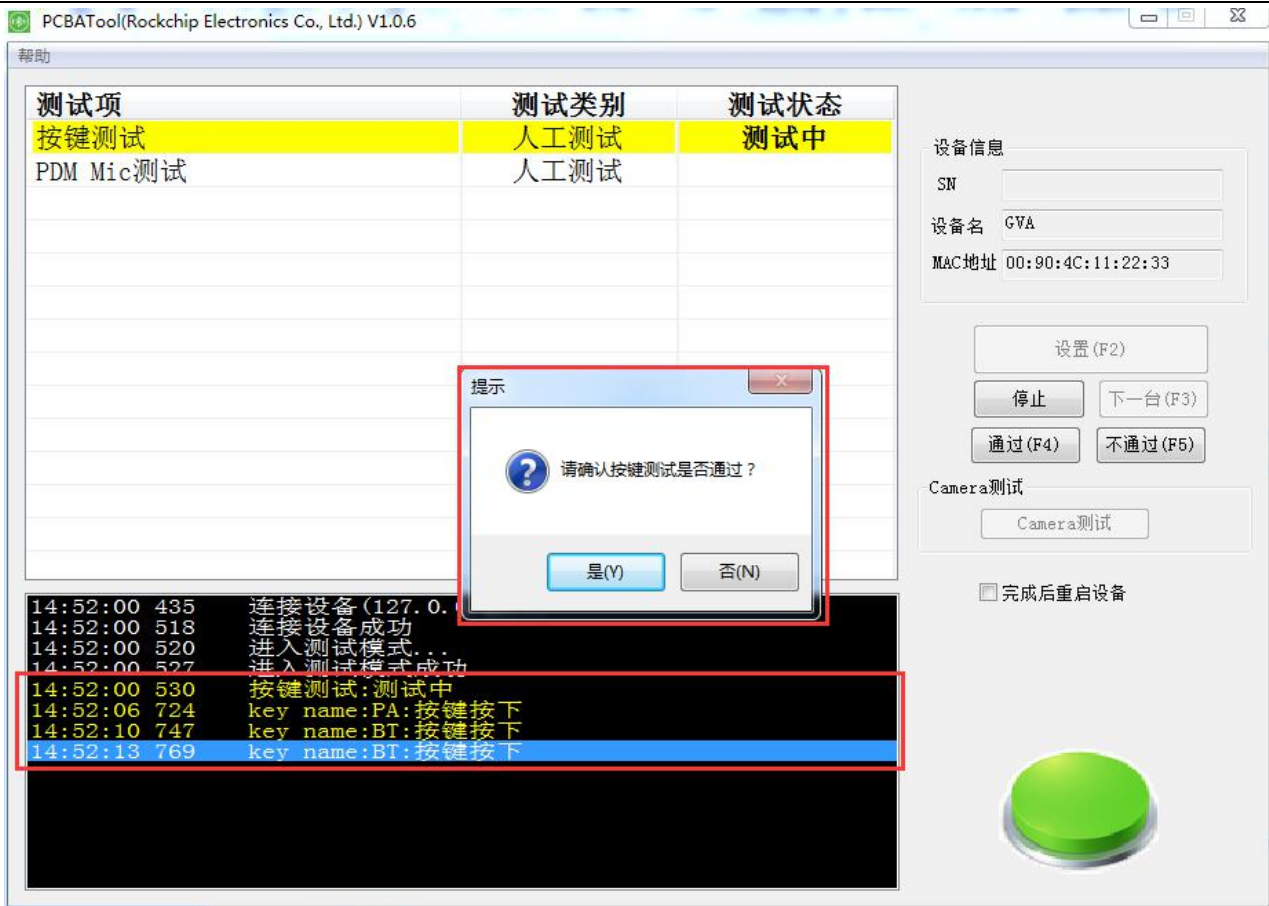


图 1-7 PCBA 按键测试

1.5 工具二次开发指引

PCBA 测试工具如需进行二次开发，扩展相关测试项目，相对比较简单，只需要根据以下步骤即可在 PCBA 测试工具上添加自己的测试项。

- 安装软件开发环境 VS2008（打上 SP1 补丁）。
- 打开源码工程，资源视图中 IDD_SETTING_DIALOG 对话框中添加新的测试项，添加一个 CheckBox 控件，重命名 ID 及 caption。
- SettingBase.h 头文件中添加测试项字符串变量及是否选择布尔型变量。
- SettingBase.cpp 源码文件中 LoadToolSetting 接口中添加对这两个变量的初始化赋值。可参考其他测试项变量的赋值。
- Config.ini 文件中添加测试项名称及是否默认选择设置。

以 SD 卡测试项为例，如下图 1-8。SdcardName 为 SD 卡测试项变量名，echo_sdcard_test 为具体板端运行的 bin 程序名称。SdcardTest 变量表示默认选中状态，0 表示默认未被选中，1 表示默认被选中。代码中将解析这两个变量的值。

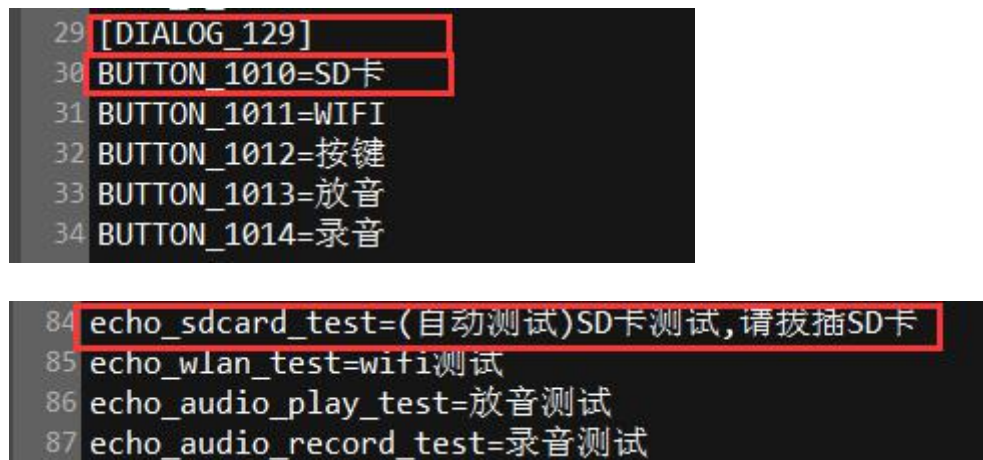
```
45 SdcardName=echo_sdcard_test
46 SdcardTest=0
```

图 1-8 PCBA config.ini 变量说明

- 语言配置文件 Chinese.ini/English.ini 中添加相应的测试项中英文字符，用来在软件界面中显示。同样以 SD 卡测试项为例：根据新添加的测试项的 ID 号，添加测试项中文显示。

[DIALOG_129]: 129 表示 IDD_SETTING_DIALOG 对话框 ID。

BUTTON_1010: 1010 表示 SD 卡测试项的 ID，等号后面表示将在软件界面上显示的名称“SD 卡”。echo_sdcard_test 这里跟 config.ini 中 SdcardName 变量值保持一致。等号后的内容为软件界面测试项列表框中显示的字符串。



```
29 [DIALOG_129]
30 BUTTON_1010=SD卡
31 BUTTON_1011=WIFI
32 BUTTON_1012=按键
33 BUTTON_1013=放音
34 BUTTON_1014=录音

84 echo_sdcard_test=(自动测试)SD卡测试, 请拔插SD卡
85 echo_wlan_test=wifi测试
86 echo_audio_play_test=放音测试
87 echo_audio_record_test=录音测试
```

图 1-9 PCBA Chinese.ini 变量说明

- Pcbatest.cpp 源文件及 Pcbatest.h 中：添加新加测试项的测试接口。
- IPSearchDlg.cpp 源文件中：
 - initTestCase 接口添加选中该新加测试项的代码，可参考其他测试项代码。
 - DoTestItem 接口添加选中执行新加测试项的代码，可参考其他测试项代码。

至此，编译无错后新加测试项到此就结束了，运行工具软件，DEBUG 模式下可以断点调试跟踪，快速定位自己新加测试项代码是否运行正常。

2 PCBA 软件设计说明

2.1 设计框图

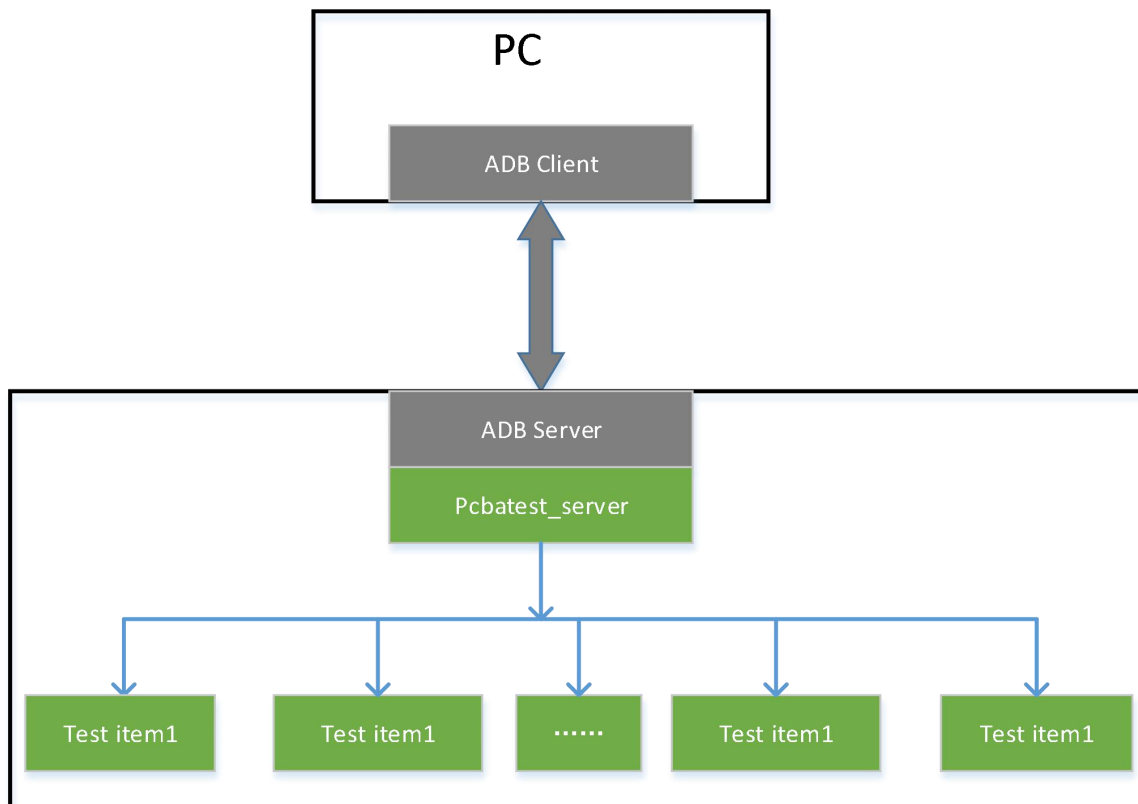


图 2-1 PCBA 设计框图变量说明

- 1、测试工具通过 **adb** 协议与设备端作为通讯手段。
- 2、每个测试程序都是一个独立的进程，互不影响。

2.2 设计流程

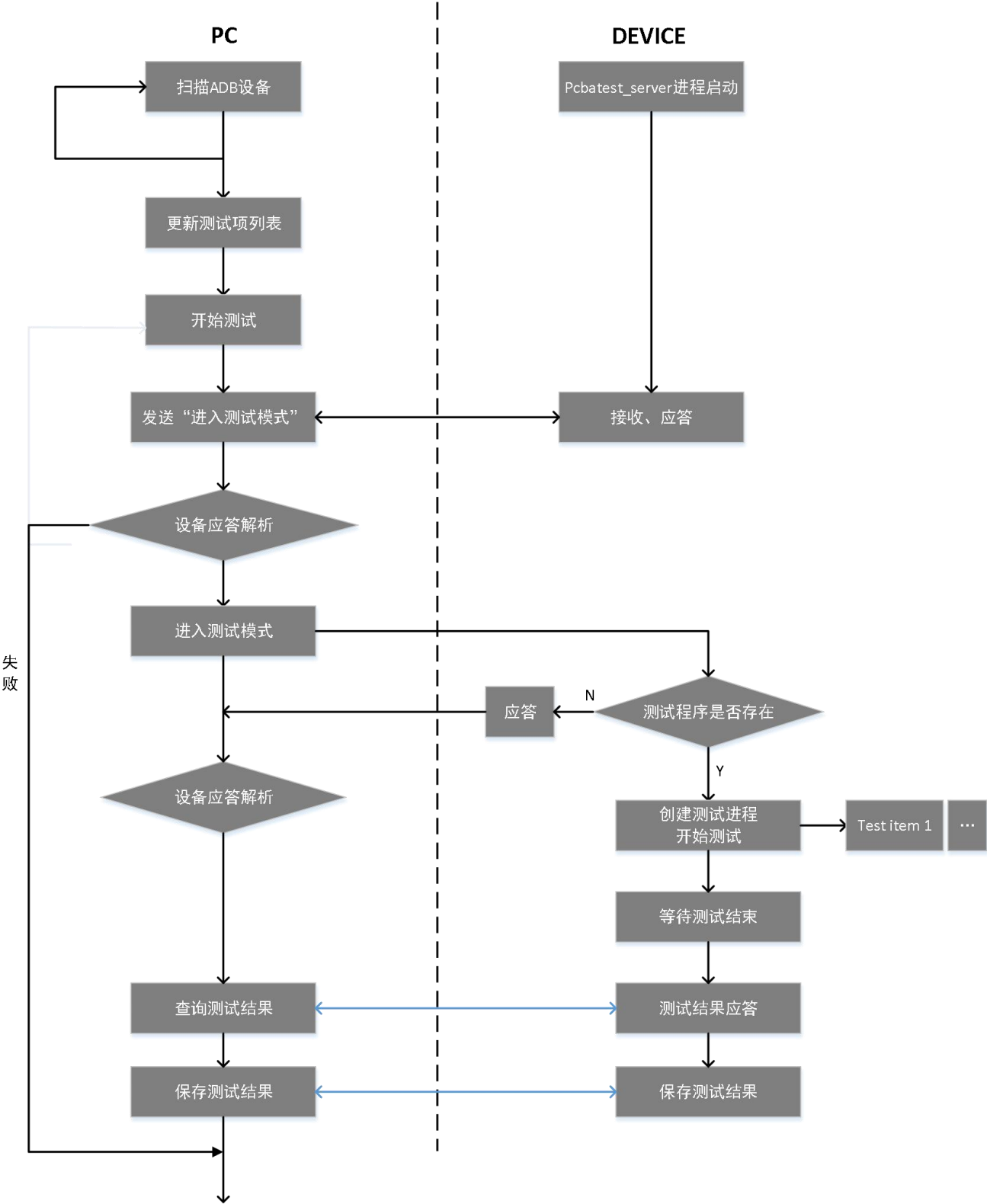


图 2-2 PCBA 测试流程图变量说明

2.3 PCBA 命令协议格式

PCBA 测试工具与设备端通过 adb 协议，使用 socket 套接字实现数据的交换。
下文将介绍 PCBA 测试工具与设备端的命令交互协议格式。

PC 端向设备端发送 command，设备端向 PC 端响应 response:

其中 command 命令的格式如下：

Command: <TYPE><TEST_ITEM><CMD><MSG>

各参数含义如表 2-1 所示。

命令格式	TPYE	TEST_ITEM	CMD	MSG
格式说明	类型	测试项名称	具体执行的命令	附带信息

表 2-1 Command 命令说明

Response 命令的格式如下：

Response: <TYPE><TEST_ITEM><RES><MSG><STATUS><RESULT><ERR_CODE>

命令格式	TPYE	TEST_ITEM	RES	MSG	STATUS	RESULT	ERR_CODE
格式说明	类型	测试项名称	响应	附带信息	状态	执行结果	错误码

表 2-2 Response 命令说明

下面详细给出了各参数取值及含义：

TYPE:

TYPE	CMD	RES				
CMD:PC 端向设备端发送命令						
RES:设备端向 PC 响应命令						

表 2-3 TYPE 命令参数取值与定义说明

TEST_ITEM:

TEST_ITEM	KEY_test SDCard_test AUDIO_test HDMI_test CAMERA_test USB_HOST_test LED_test
1.“TEST ITEM”为具体的测试项，测试项名称要与测试程序名相同	
2.要考虑支持测试项扩展	

表 2-4 TEST_ITEM 命令参数取值与定义说明

CMD/RES:

CMD/RES	ENTER	EXIT	START	SAVE	QUERY	
ENTER: 进入测试模式（不带测试项） EXIT:退出测试模式（不带测试项） START: 开始测试 SAVE: 保存测试结果 QUERY: 查询测试状态						

表 2-5 CMD/RES 命令参数取值与定义说明

MSG:

MSG	msg					
command 下发的数据或者 response 上传的数据，如果没有数据，不发送该选项						

表 2-6 MSG 命令参数取值与定义说明

STATUS:

STATUS	ACK	NAK				
PC 发送每一个命令设备端都需要应答 ACK: 应答成功 NAK: 应答失败						

表 2-7 STATUS 命令参数取值与定义说明

RESULT:

RESULT	TESTING	PASS	FAIL	VERIFY		
PC 下发"QUERY"命令查询测试结果，设备端需要返回测试结果给 PC TESTING:测试中 PASS: 测试成功 FAIL: 测试失败 VERIFY: 测试完成，确认测试结果 PRESS:只有按键测试用到						

表 2-8 RESULT 命令参数取值与定义说明

ERR_CODE:

ERR_CODE	err_code					
测试失败原因，测试成功不发送改选项 0x01-0xFF 错误码						

表 2-9 ERR_CODE 命令参数取值与定义说明

下面将举例说明上面命令的用法:

1.enter pcba test mode

进入测试模式

{ "TYPE": "CMD", "CMD": "ENTER" }

进入测试模式应答

正确: {"TYPE":"RES", "RES":"ENTER", "STATUS":"ACK"}

错误: {"TYPE":"RES", "RES":"ENTER", "STATUS":"NAK", "ERR_CODE":"err_code"}

2.exit pcba test mode

退出测试模式

{"TYPE":"CMD", "CMD":"EXIT"}

退出测试模式应答

{"TYPE":"RES", "CMD":"EXIT", "STATUS":"ACK"}

{"TYPE":"RES", "RES":"EXIT", "STATUS":"NAK", "ERR_CODE":"err_code"}

3.以按键测试项为例：，其他测试项与此类似。

开始测试

{"TYPE":"CMD", "TEST_ITEM":"KEY-TEST", "CMD":"START"}

开始测试应答

{"TYPE":"RES", "TEST_ITEM":"KEY-TEST", "RES":"START", "STATUS":"ACK"}

{"TYPE":"RES", "TEST_ITEM":"KEY-TEST", "RES":"START", "STATUS":"NAK",
"ERR_CODE":"err_code"}

4.查询测试状态

{"TYPE":"CMD", "TEST_ITEM":"test_item", "CMD":"QUERY" }

查询状态应答

出错应答: {"TYPE":"RES", "TEST_ITEM":"test_item", "RES":"QUERY", "MSG":"msg",
"STATUS":"NAK", "ERR_CODE":"err_code"}

正在测试: {"TYPE":"RES", "TEST_ITEM":"test_item", "RES":"QUERY", "MSG":"msg",
"STATUS":"ACK", "RESULT":"TESTING"}

测试通过: {"TYPE":"RES", "TEST_ITEM":"test_item", "RES":"QUERY", "MSG":"msg",
"STATUS":"ACK", "RESULT":"PASS"}

测试结果确认: {"TYPE":"RES", "TEST_ITEM":"test_item", "RES":"QUERY", "MSG":"msg",
"STATUS":"ACK", "RESULT":"VERIFY"}

测试失败: {"TYPE":"RES", "TEST_ITEM":"test_item", "RES":"QUERY", "MSG":"msg",
"STATUS":"ACK", "RESULT":"FAIL", "ERR_CODE":"err_code"}

某个按键按下: {"TYPE":"RES", "TEST_ITEM":"key_test", "RES":"QUERY", "MSG":"msg",
"STATUS":"ACK", "RESULT":"PRESS"}(此应答仅针对按键测试)

3 PCBA 测试代码框架

3.1 代码结构

PCBA 测试代码是设备端执行的具体测试项的二进制程序，可以单独运行。测试程序位于工程源码 /external/rk-pcba-test 目录， 结构如下：

```
.
|-- audio_test.h
|-- bootinfo.c
|-- bootinfo.h
|-- bt_test.h
|-- build_all.sh
|-- cJSON
|   |-- cJSON.c
|   `-- cJSON.h
|-- CMakeLists.txt
|-- common.h
|-- cpu_test.h
|-- ddr_test.h
|-- echo_audio_play_test.c
|-- echo_audio_record_test.c
|-- echo_audio_test.c
|-- echo_auto_test.c
|-- echo_bt_test.c
|-- echo_cpu_test.c
|-- echo_ddr_test.c
|-- echo_discovery.c
|-- echo_emmc_test.c
|-- echo_key_test.c
|-- echo_led_test.c
|-- echo_pcbatest_server
|-- echo_pcbatest_server.c
|-- echo_rotary_test.c
|-- echo_rtc_test.c
|-- echo_wlan_test.c
|-- emmc_test.h
|-- key_test.h
|-- led_test.h
|-- pcbatest_server.h
|-- rk_pcba_test
|   |-- audio_test_start.wav
|   |-- sdcard_test.sh
|   |-- usbhost_test.sh
|   |-- vibration.wav
|   `-- wifi.sh
|-- rk_pcba_test_led.h
```

```
|-- rtc_test.h
|-- tinyalsa
|   |-- asoundlib.h
|   |-- pcm.c
|   |-- tinycap.c
|   `-- tinyplay.c
```

3.2 代码框架说明

设备进行 pcba 测试时，设备端是作为 tcp 中的服务器端，PC 端作为客户端；设备与 PC 的交互主要体现在 echo_pcbatest_server.c 代码中。echo_pcbatest_server.c 中设备和 PC 的交互流程如下：

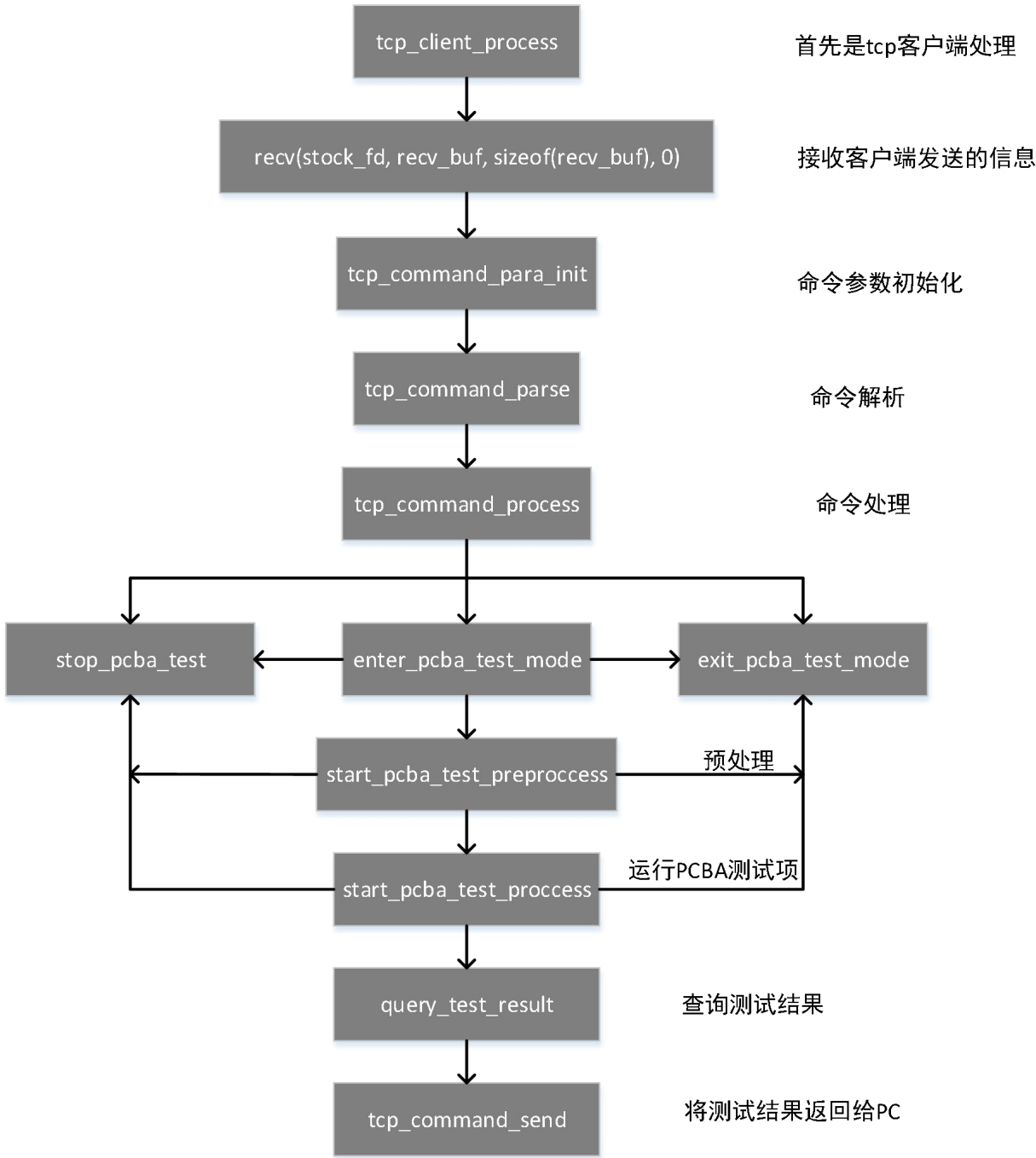


图 3-1 PCBA 设备端测试代码框架

3.3 代码编译

RK3308 的 EVB 板的 PCBA 测试程序编译如下：

根目录下执行：

```
source buildroot/build/envsetup.sh
```

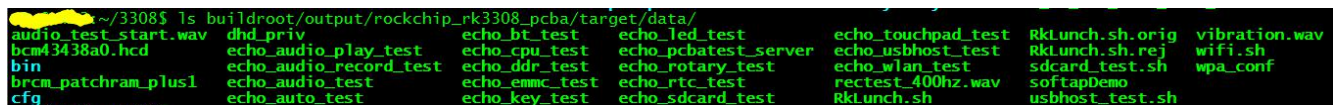
```
You're building on Linux
Lunch menu...pick a combo:
1. rockchip_rk3308_release
2. rockchip_rk3308_debug
3. rockchip_rk3308_robot_release
4. rockchip_rk3308_robot_debug
5. rockchip_rk3308_mini_release
6. rockchip_rk3326_release
7. rockchip_rk3326_debug
8. rockchip_rk3308_pcba
```

```
Which would you like? [1] 8
```

这里选择第 8 个选项： rockchip_rk3308_pcba，
之后执行编译并生成 PCBA 的固件，命令如下：

```
make pcba
make
```

编译后的执行文件如下图所示：



```
~/3308$ ls buildroot/output/rockchip_rk3308_pcba/target/data/
audio_test_start.wav  dhdt_priv  echo_bt_test  echo_led_test  echo_touchpad_test  Rklunch.sh.orig  vibration.wav
bcm43438a0.hcd        echo_audio_play_test  echo_cpu_test  echo_pcbatest_server  echo_usbhost_test  Rklunch.sh.rej  wifi.sh
bin                   echo_audio_record_test  echo_ddr_test  echo_rotary_test  echo_wlan_test  sdcard_test.sh  wpa_conf
brcm_patchram_plus1  echo_audio_test        echo_emmc_test  echo_rtc_test  rectest_400hz.wav  softapDemo
cfg                   echo_auto_test         echo_key_test   echo_sdcard_test  Rklunch.sh        usbhost_test.sh
```

图 3-2 PCBA 测试代码编译结果

修改代码后，请在 SDK 工程根目录执行

```
make pcba-rebuild
```

然后再

```
make
```

执行 make 命令后，pcba 的固件将会在

./buildroot/output/rockchip_rk3308_pcba/images/目录生成。

将固件文件 rootfs.squashfs 烧写进 rootfs 分区即可开机进行测试。