

# Tina Linux

dragonMAT 说明文档 v1.0

# 文档履历

版本号	日期	制/修订人	制/修订记录
V1.0	2018/2/24		初始版本



# 目 录

1. 概述.....	4
1.1. 编写目的.....	4
1.2. 适用范围.....	4
2. 配置.....	5
2.1. PC 端配置.....	5
2.2. 设备端配置.....	6
3. 用例编写.....	8
3.1. C 接口.....	8
3.2. shell 接口.....	10
4. 程序示例.....	11
4.1. shell.....	11
4.2. C.....	12
5. dragonMAT 界面.....	13
6. Declaration.....	16



## 1. 概述

文档主要描述量产测试工具 dragonMAT 的配置、接口使用方法、及界面使用方法。

### 1.1. 编写目的

简要介绍 dragonMAT，方便开发者配置 dragonMAT & 使用接口编写测试用例

### 1.2. 适用范围

适用于 Tina SDK



## 2. 配置

### 2.1. PC 端配置

dragonMAT 的配置文件在其安装目录下，名为“global.ini”。

开发者在打开 dragonMAT 软件前，编辑此文件，能够对 dragonMAT 进行配置。

一般只需更改常用的几个配置，其他配置选项不要更改，以免发生不可预料的错误。

#### 常用配置项：

;最大测试样机数。dragonMAT 最多支持同时测试 4 台设备。dut<=4。

dut=3

;手动启动测试。设为 0：自动测试，无需手动 1：手动测试，需手动点击开始测试按钮。

manual=0

;循环测试。

circle=0

;设备断开时是否清空界面，否则到下一个设备进来时才清空

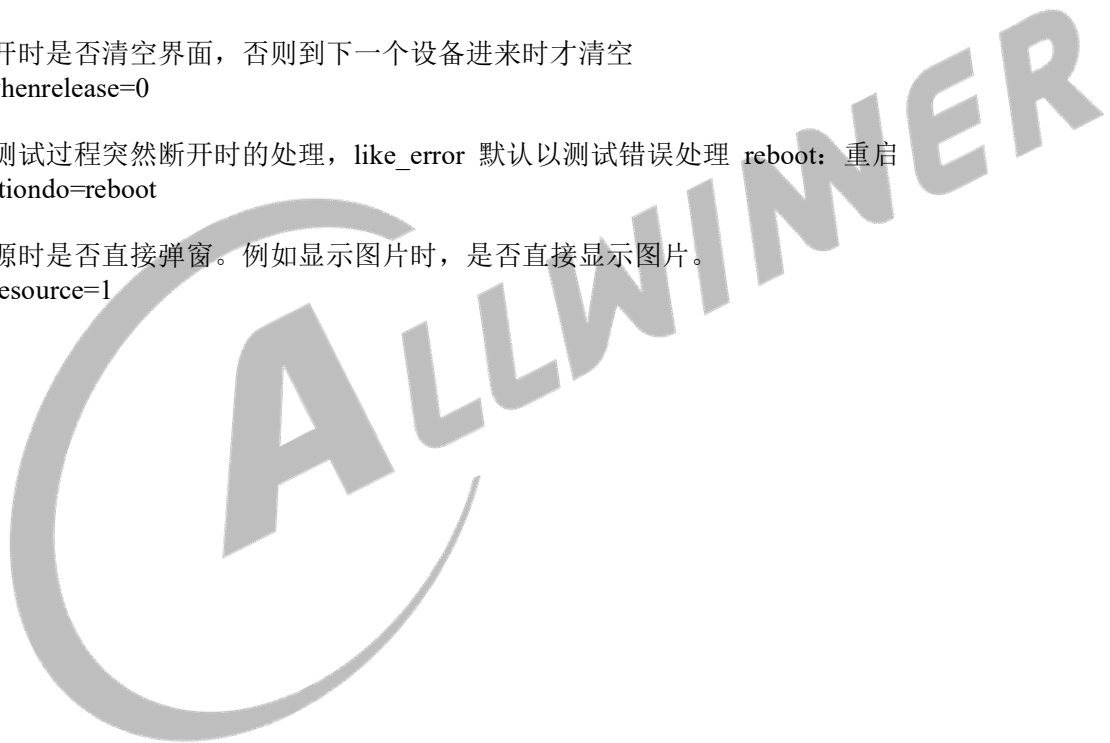
cleanupwhenrelease=0

;设备在测试过程突然断开时的处理，like\_error 默认以测试错误处理 reboot：重启

netexceptiondo=reboot

;显示资源时是否直接弹窗。例如显示图片时，是否直接显示图片。

autopopresource=1



## 2.2. 设备端配置

在 tina 目录下执行”make menuconfig”，如图所示，进行输出方式的选择：

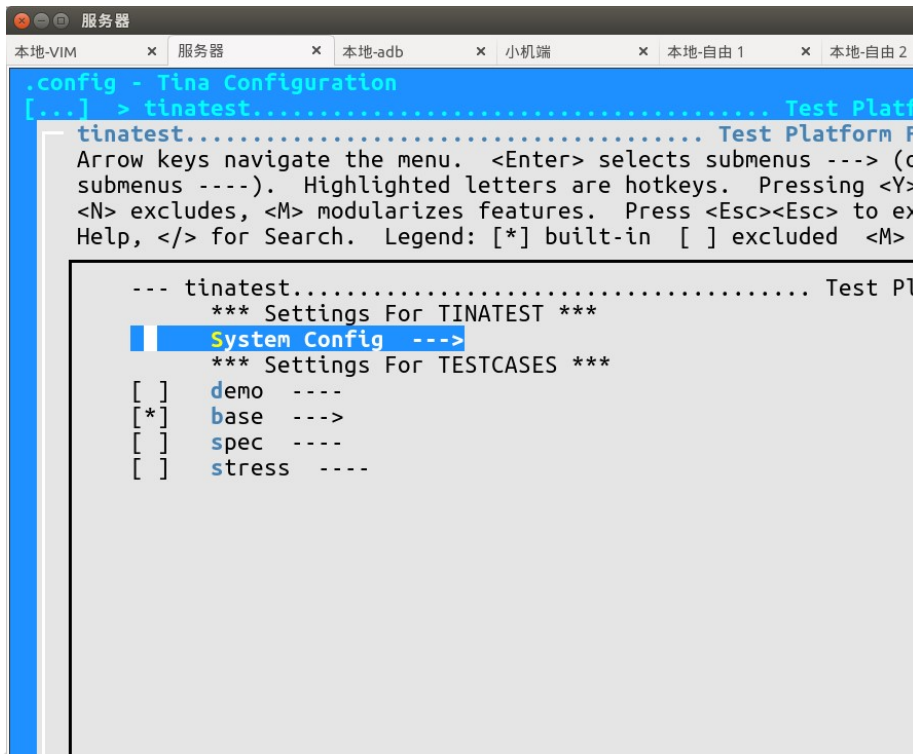


图 2-1 TestTools-->tinatool-->System Config

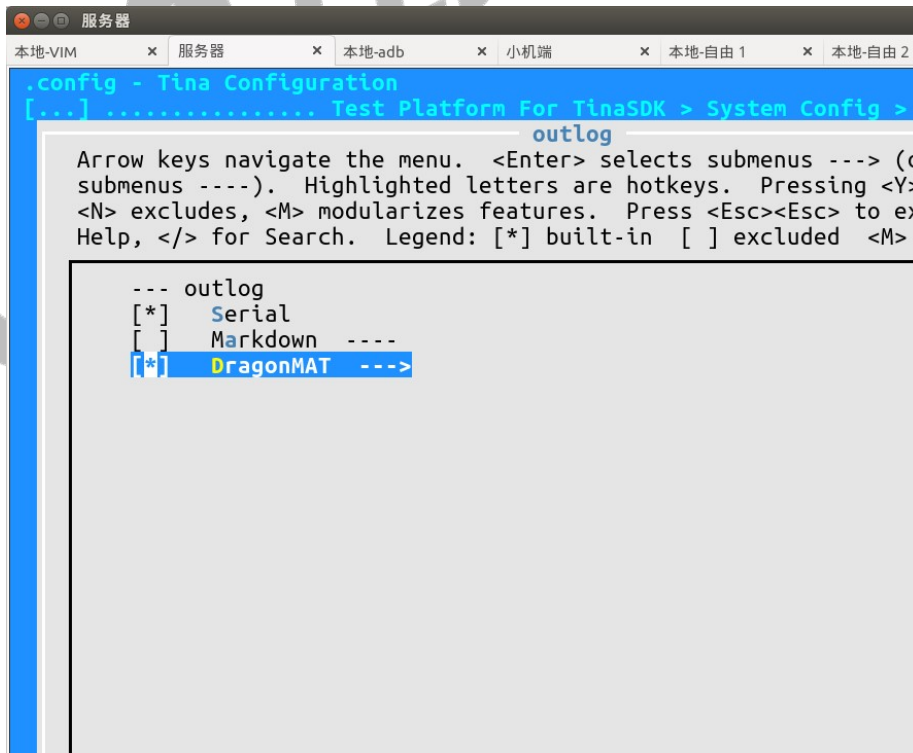


图 2-2 System Config-->global-->outlog

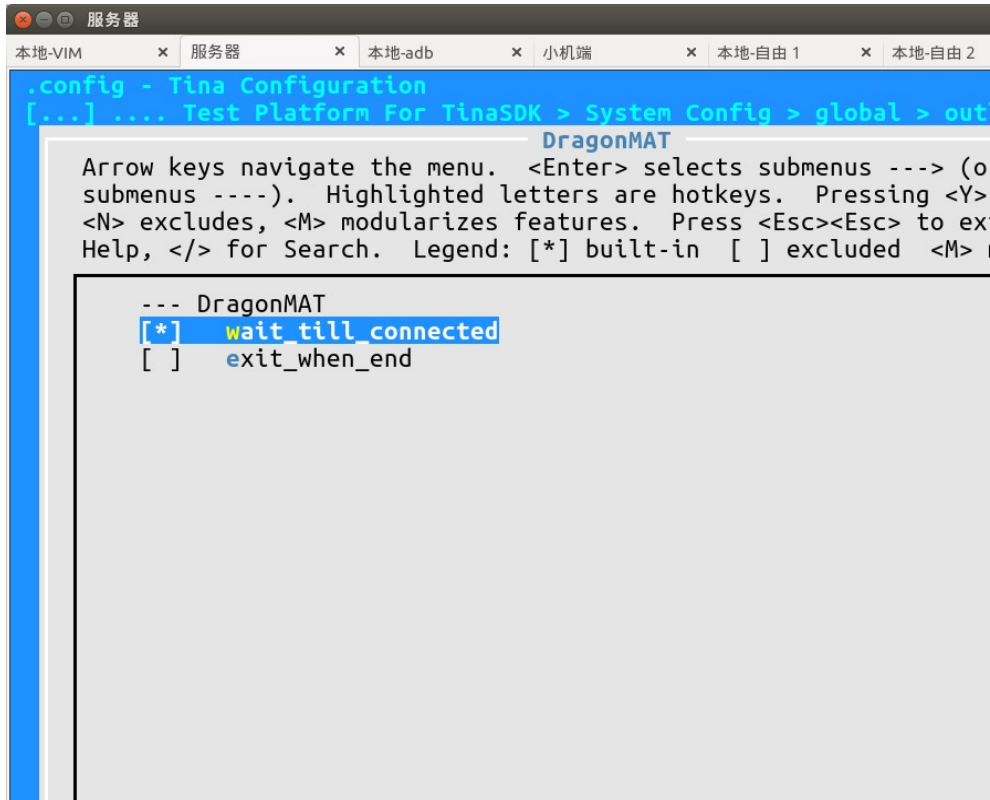


图 2-3 配置 dragonMAT

其中，输出方式根据需要，可以选择 Serial 或者 DragonMAT，或者二者都选均可。

dragonMAT 配置项说明：

wait\_till\_connected: 等待设备端和 PC 端连接后，再进行测试

exit\_when\_end: 当测试完成时退出

### 3. 用例编写

#### 3.1. C 接口

使用前提：测试用例需包含头文件： #include “interact.h”

函数原型	int task(const char *ask, char *reply, int len)
参数说明	<p><b>ask</b>: 该测试用例向用户提出的问题。 eg1: ask = “Please enter the WiFi password:”</p> <p><b>reply</b>: 指向一块内存空间，用于存放用户输入的回答</p> <p><b>len</b>: reply 指向的内存空间的大小</p>
返回说明	0：成功 -1：发送失败
功能描述	该测试用例向用户提出问题（ask），并获取用户的回答（reply）。表现为在执行测试用例时，打印出 ask 字符串，用户在输入回答后，测试用例将回答字符串保存在 reply 指向的内存空间。
交互形式	<p><b>串口</b>:</p> <pre>=====Testcase Information ===== &lt;该测试用例路径&gt;: 该测试用例提出的问题 ask =====END=====</pre> <p><b>dragonMAT</b>:</p> <p>弹出对话框，用户输入数据后，点击提交</p>

函数原型	int ttips(const char *tips)
参数说明	<b>tips</b> : 该测试用例向用户提示的信息
返回说明	0：成功 -1：发送失败
功能描述	该测试用例向用户提示信息。
交互形式	<p><b>串口</b>:</p> <pre>=====Testcase Information ===== &lt;该测试用例路径&gt;: 该测试用例的提示信息 tips =====END=====</pre> <p><b>dragonMAT</b>:</p> <p>dragonMAT 界面中，该测试用例对应区域显示 tips 字符串</p>



函数原型	int ttrue(const char *tips)
参数说明	<b>tips:</b> 该测试用例向用户提出的问题 eg1: tips = "Could you see this picture?" eg2: tips = "Could you sound the music clearly?"
返回说明	1: 用户选择“是” 0: 用户选择“否” -1: 发送失败
功能描述	该测试用例向用户提问, 用户根据问题, 选择 (是 / 否)
交互形式	<b>串口:</b> =====Testcase Information ===== <该测试用例路径>: 该测试用例提出的问题 tips [Y  n]: =====END ===== <b>dragonMAT:</b> dragonMAT 界面中, 该测试用例对应区域显示 tips 字符串, 并出现是和否按钮, 让用户点击选择

函数原型	int tupfile(const char *filepath, const char *tips)
参数说明	<b>filepath:</b> 设备端要上传的文件路径 <b>tips:</b> 该测试用例向用户提示的信息
返回说明	0: 成功 -1: 失败
功能描述	将设备端的文件上传到 PC 端。PC 端文件保存目录: dragonmat_XXX/result_dir/0/
交互形式	<b>串口:</b> 暂不支持该接口 <b>dragonMAT:</b> dragonMAT 界面中, 底端显示信息“文件上传:<filename>”, 并上传文件到 PC 端 dragonmat_XXX/result_dir/0/目录

函数原型	int tshowimg(const char *filepath, const char *tips);
参数说明	<b>filepath:</b> 设备端要上传的图片路径 <b>tips:</b> 该测试用例向用户提出的问题
返回说明	0: 用户选择“是” 1: 用户选择“否” -1: 发送失败
功能描述	设备端上传图片到 PC 端, PC 端显示图片并询问用户 tips 问题(比如: 图片是否清晰正确?)
交互形式	<b>串口:</b> 暂不支持该接口 <b>dragonMAT:</b> 1. dragonMAT 界面中, 底端显示信息“文件上传:<filename>”, 上传图片到 PC 端 dragonmat_XXX/result_dir/0/目录。 2. 自动显示图片 3. 询问用户, 且出现是和否按钮, 用户点击选择

### 3.2. shell 接口

使用方法与 C 接口类似。其交互形式与 3.2 中 C 接口交互形式相同。

命令	task <ask>
参数说明	<b>ask:</b> 该测试用例向用户提出的问题。 eg1: task "Please enter the WiFi password!"
返回说明	0 : 成功 1 : 失败
功能描述	该测试用例向用户提出问题 (ask)，并打印出用户的回答。

命令	ttips <tips>
参数说明	<b>tips:</b> 该测试用例向用户提示的信息
返回说明	0 : 成功 1 : 失败
功能描述	该测试用例向用户提示信息。

命令	ttrue <tips>
参数说明	<b>tips:</b> 该测试用例向用户提出的问题
返回说明	0 : 用户选择“是” 1 : 用户选择“否”
功能描述	该测试用例向用户提问，用户根据问题，选择 (是 / 否)

命令	tupfile <filepath> <tips>
参数说明	<b>filepath:</b> 设备端要上传的文件路径 <b>tips:</b> 该测试用例向用户提示的信息
返回说明	0 : 成功 1 : 失败
功能描述	将设备端的文件上传到 PC 端。PC 端文件保存目录: dragonmat_XXX/result_dir/0/

命令	tshowimg <filepath> <tips>
参数说明	<b>filepath:</b> 设备端要上传的图片路径 <b>tips:</b> 该测试用例向用户提示的信息
返回说明	0 : 用户选择“是” 1 : 用户选择“否”
功能描述	设备端上传图片到 PC 端, PC 端显示图片并询问用户 tips 问题(比如: 图片是否清晰正确? )

## 4. 程序示例

### 4.1. shell

```
#!/bin/sh
# 提示用户： This is a example test
tips "This is a example test."

# 提问用户： Please enter the WiFi password 并将字符串保存在 password 变量里
password=$(task "Please enter the WiFi password")
echo "WiFi password: $password"

# 提问用户： Could you sound the music?用户会选择（是 / 否）
# 通过 $? 获取返回值，得知用户选择
true "Could you sound the music?"
if [ $? = 0 ]
then
    echo "GOOD, audio run success."
else
    echo "ERROR, audio run failed."
fi

# 上传设备端文件 /etc/text.txt 到 PC 端，并显示提示信息  upload file test.txt */
tupfile "/etc/test.txt" "upload file test.txt"

# 上传设备端图片/etc/demo.bmp 并显示在 PC 端，提示信息 Could you see this picture clearly?
# 用户会选择（是 / 否）,获取其返回值，得知用户选择
tshowimg "/etc/demo.bmp" "Could you see this picture clearly?"
if [ $? = 0 ]
then
    echo "GOOD, picture show success."
else
    echo "ERROR, picture show failed."
fi
```

## 4.2. C

```
#include <stdio.h>
#include <stdlib.h>
#include "interact.h"

int main(int argc, char **argv)
{
    int ret = -1;
    char *reply=(char *)malloc(sizeof(char)*1024);
    /* 提示用户： This is a example test. */
    tips("This is a example test.");

    /*
     * 提问用户： Please enter the password:
     * 用户输入的密码保存在 reply 指向的内存空间中
     */
    task("Please enter the password:", reply, 1024);
    printf("Password: %s\n", reply);

    /*
     * 提问用户： Could you sound the music?
     * 用户会选择（是 / 否）
     * 通过获取返回值，得知用户选择
     */
    ret = ttrue("Could you sound the music?");
    if (ret == 0) {
        printf("GOOD, audio run success.\n");
    } else {
        printf("ERROR, audio run failed.\n");
        goto err;
    }
    /* 上传设备端文件 /etc/text.txt 到 PC 端，并显示提示信息 upload file test.txt */
    tupfile("/etc/text.txt", "upload file test.txt");

    /*
     * 上传设备端图片/etc/demo.bmp 并显示在 PC 端，提示信息 Could you see this picture clearly?
     * 用户会选择（是 / 否）
     * 获取其返回值，得知用户选择
     */
    ret = tshowimg("/etc/demo.bmp", "Could you see this picture clearly?");
    if (ret == 0) {
        printf("GOOD, picture show success.\n");
    } else {
        printf("ERROR, picture show failed.\n");
        goto err;
    }
    ret = 0;
err:
    free(reply);
    return ret;
}
```

## 5. dragonMAT 界面

demo-true 测试用例，调用了 task 接口，其问题为“Please enter password:”，用户会输入相应的回答，点击“提交”即可。如图 5-1:



图 5-1 task 接口交互

demo-true 测试用例，调用了 ttrue 接口，其问题为“demo-true: ttrue. see?”，用户会点击是 / 否进行选择。如图 5-2:

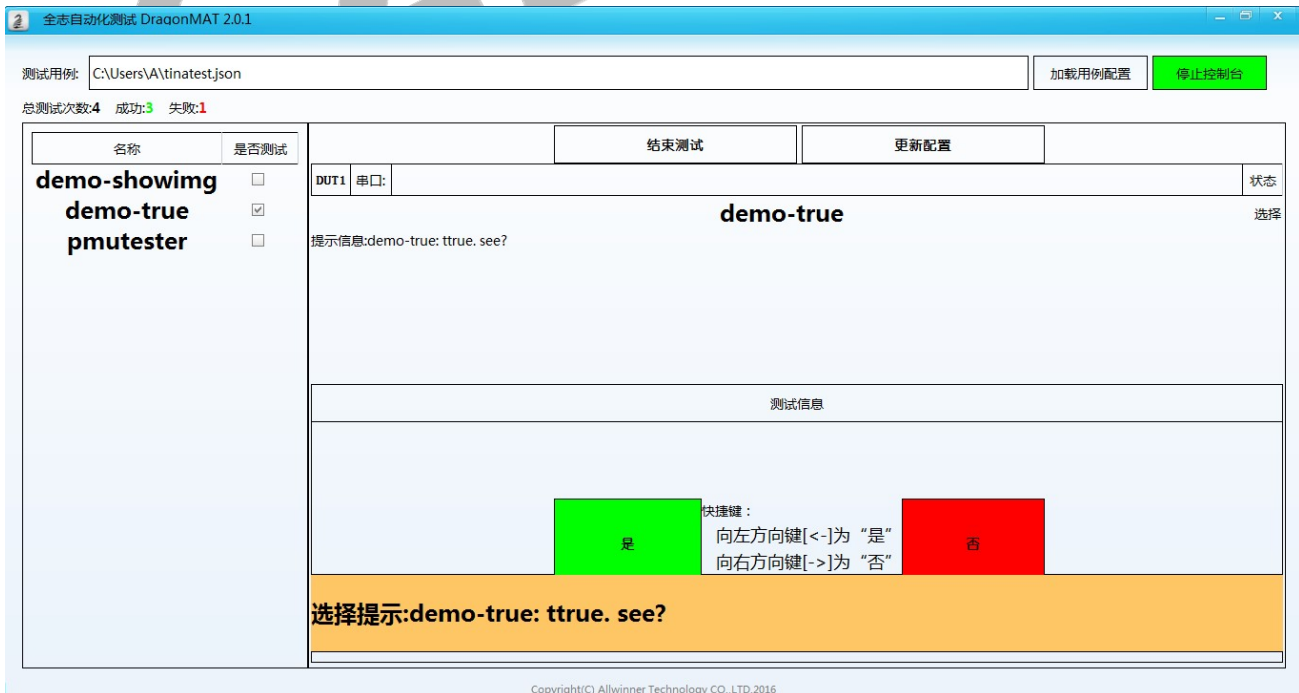


图 5-2 ttrue 接口交互

demo-true 测试用例，调用了 ttips 接口，其提示信息为“demo-true:tips message.” 如图 5-3:

		开始测试	更新配置	
DUT1	串口:			状态
<b>demo-true</b>				✔
提示信息:demo-true: tips message.				
<b>pmutester</b>				✔

图 5-3 ttips 接口提示信息

demo-showimg 测试用例，调用了 tupfile 接口，上传 tinatest.json 文件。如图 5-4:

		开始测试	更新配置	
DUT1	串口:			状态
<b>demo-showing</b>				✔
<b>pmutester</b>				✔
<b>文件上传:tinatest.json</b>				

图 5-4 tupfile 接口交互

demo-showimg 测试用例，调用了 tshowimg 接口，上传并显示 demo.bmp 图片，并进行提问：“demo-showimg OK?” 如图 5-5:

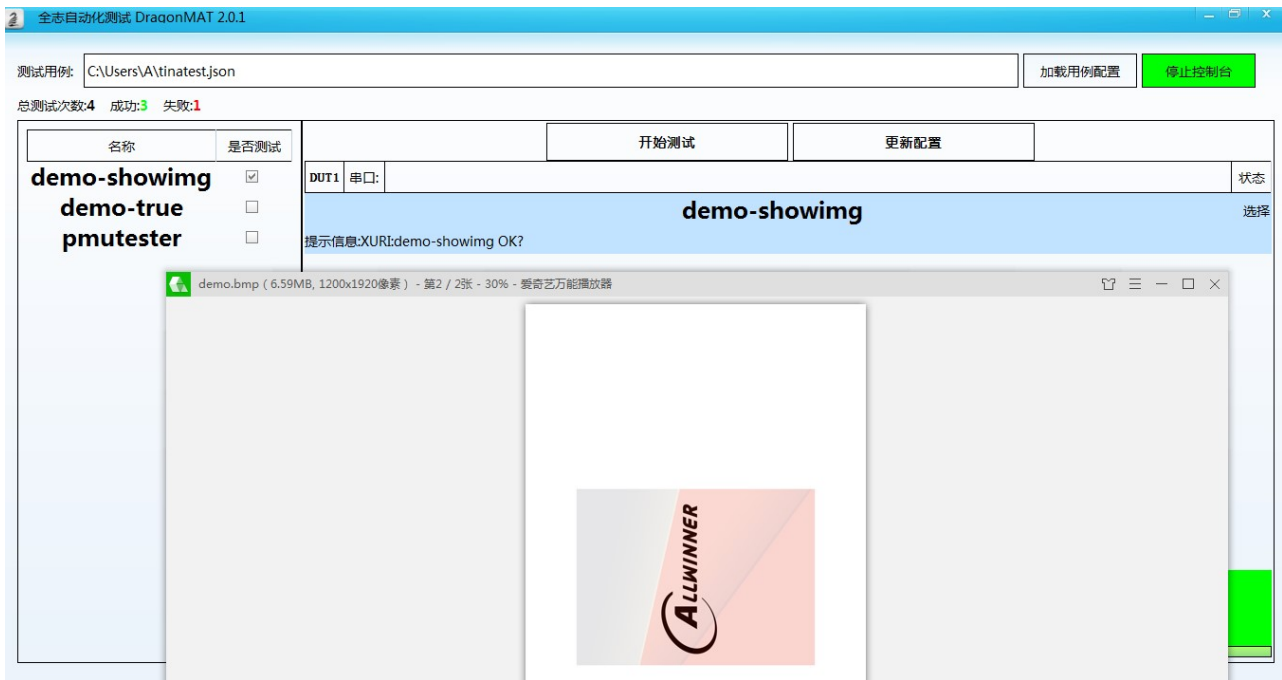


图 5-5 tshowing 接口交互



## 6. Declaration

This document 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 acknowledgment 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.

