

EVB3020 Android 系统使用手册

1 硬件资源概况

1.1 开发板照片

1.2 硬件资源及设备节点

二、Android界面功能使用说明

2.1 桌面与菜单栏界面

2.1.1 桌面区域介绍

2.1.2 状态栏隐藏/显示

2.1.3 导航栏隐藏/显示

2.2 鼠标

2.3 设置语言

2.4 调节亮度

2.5 WIFI

2.5.1 设置界面连接WIFI热点

2.5.2 软件中配置开启WIFI热点

2.5.3 APP中开启和关闭WIFI

2.6 4G

2.6.1 APP中开启和关闭4G数据

2.7 以太网口

2.7.1 界面设置以太网静态IP

2.7.2 代码中修改静态/动态IP地址

2.8 热点和网络共享

2.8.1 WLAN热点共享

2.8.2 USB网络共享

2.8.3 蓝牙网络共享

2.9 查看 IP

2.9.1 WIFI

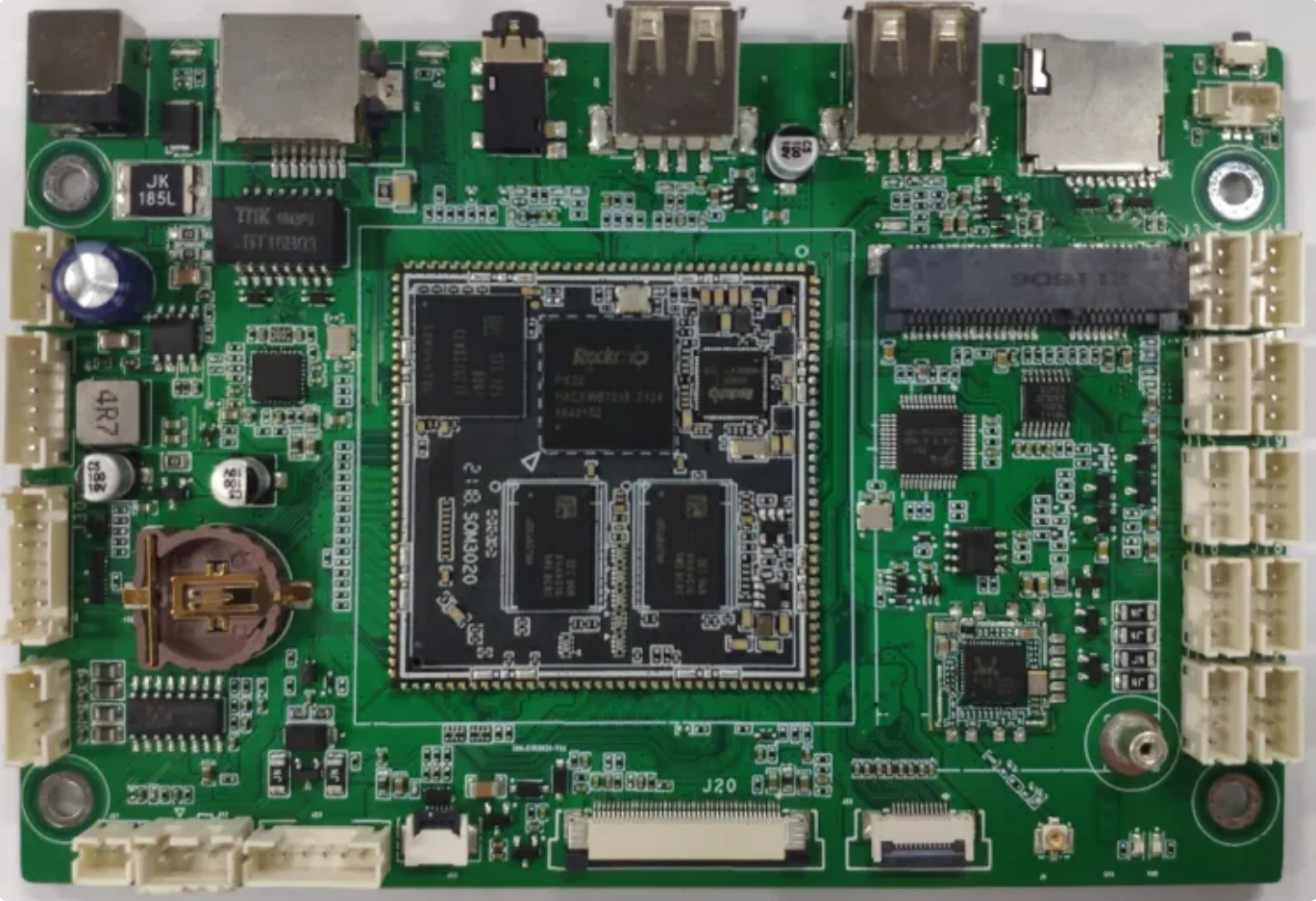
2.9.2 以太网

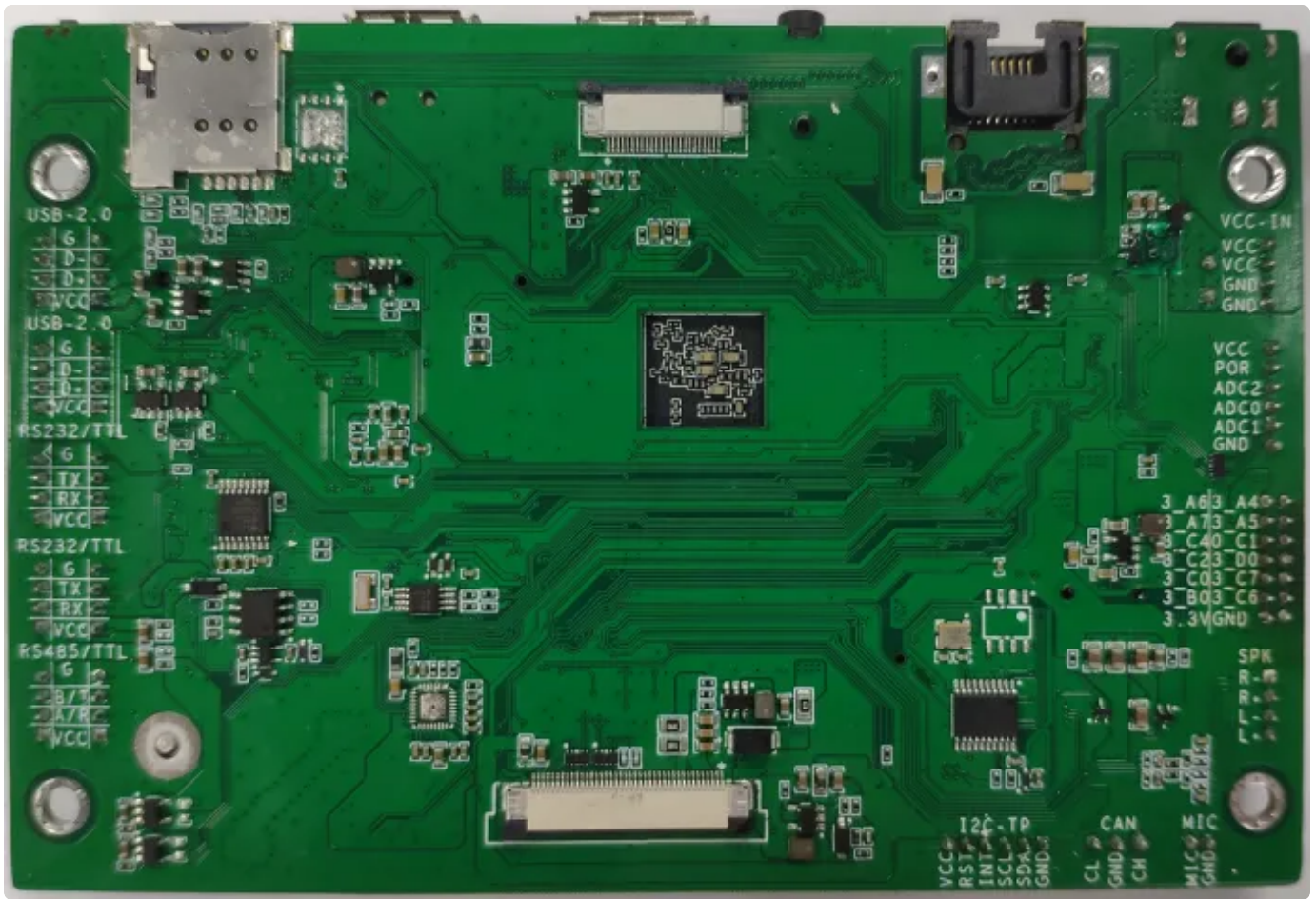
2.10 蓝牙（经典蓝牙协议）

- 2.11 SD卡和USB存储设备
- 2.12 APK安装与卸载
 - 2.12.1 通过U盘/SD卡安装APK
 - 2.12.2 ADB安装
- 2.13 查看图片与播放视频
- 2.14 时间设置
- 2.16 串口
 - 2.16.1 RS232
 - 2.16.2 RS485
- 2.17 录音
- 2.18 触摸划线
- 2.19 输入法切换
- 2.20 安兔兔跑分
- 2.21 恢复出厂设置
- 2.22 ZIP包升级
- 2.23 LCD显示方向旋转
- 2.24 LCD显示密度设置
- 2.25 APK静默安装
- 2.26 APP最前端运行检测
- 2.27 关机/重启
- 2.28 设置系统时间
- 2.29 设置默认Launcher
- 2.30 修改系统开机动画
- 2.31 网络ADB
- 2.32 设备背光亮度

1 硬件资源概况

1.1 开发板照片





1.2 硬件资源及设备节点

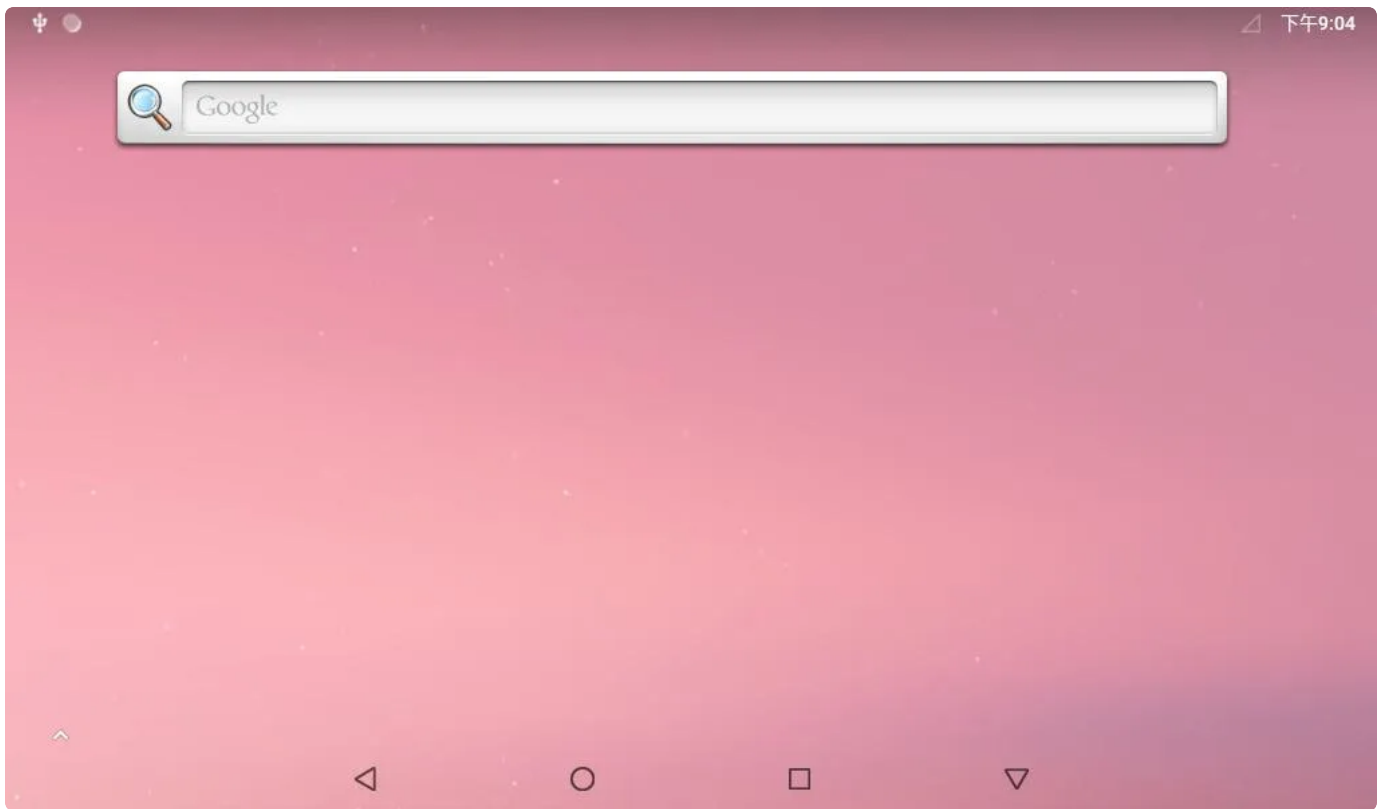
序号	名称	描述	设备节点
1	内核版本	4.4.138	
2	文件系统	Android8.1	
3	内存	DDR4 (1GB/2GB选配)	
4	存储	eMMC 5.1 (8GB/16GB/32GB/128GB 可选配)	/dev/block/mmcblk1
5	供电	12V DC @2A	
6	显示	MIPI、LVDS	
7	USB OTG	USB 2.0 OTG X 1	
8	USB HOST	USB 2.0 Type-A X 1 USB 2.0 PH-4 X 1	

9	TF Card	Micro SD 卡	/dev/block/mmcblk0
10	以太网	自适应10/100/100Mbps以太网	eth0
11	WiFi/BT	RT8723DU	wlan0、hci0
12	4G	支持MiniPCIE座子接口的4G模组, EC20	
13	扬声器	双声道 4Ω3W 扬声器	
14	耳机	3.5mm 耳机	
15	CAN	spi转CAN X 1	can0
16	摄像头	OV5648 mipi摄像头	
17	RTC	HYM8563	/dev/rtc0
18	RS232	RS232 X 4	/dev/ttyS0 /dev/ttyS1 /dev/ttyS4 /dev/ttyS5
19	RS485	RS485 X 2	/dev/ttyS2 (调试串口和RS485 2选1) /dev/ttyS3
20	调试串口	TTL X 1	/dev/ttyFIQ0
21	ADC按键	1路	/dev/input/event2
22	ADC IN	3路	/sys/bus/iio/devices/iio\:device0/in_voltage0_raw /sys/bus/iio/devices/iio\:device0/in_voltage1_raw /sys/bus/iio/devices/iio\:device0/in_voltage2_raw

二、Android界面功能使用说明

2.1 桌面与菜单栏界面

2.1.1 桌面区域介绍



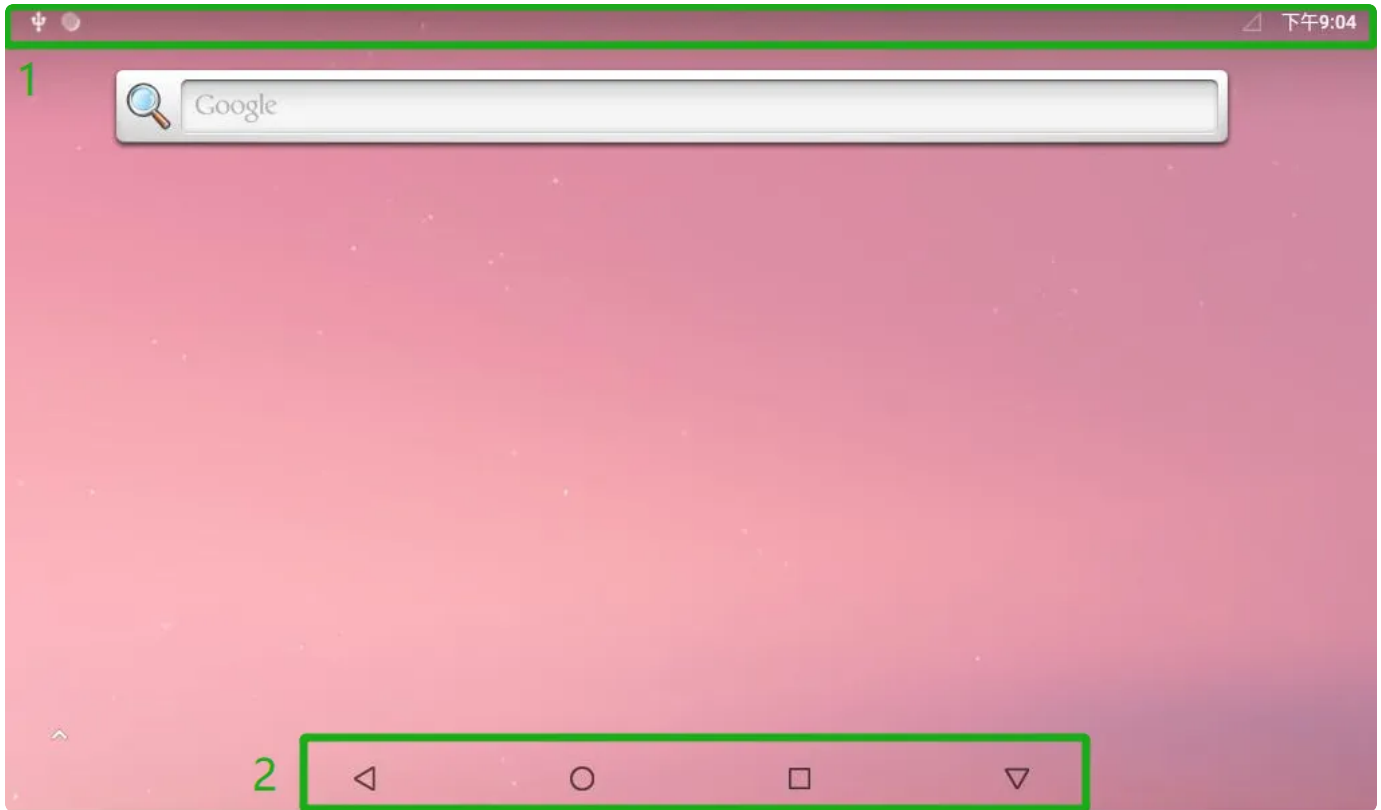
桌面



菜单栏界面

在菜单栏界面可看到系统自带的程序APP，可根据需要操作对应的APP

相关介绍：



1: 状态栏（可在此看到U盘SD卡挂载信息、WIFI连接信号等）

2: 导航栏

导航栏介绍：



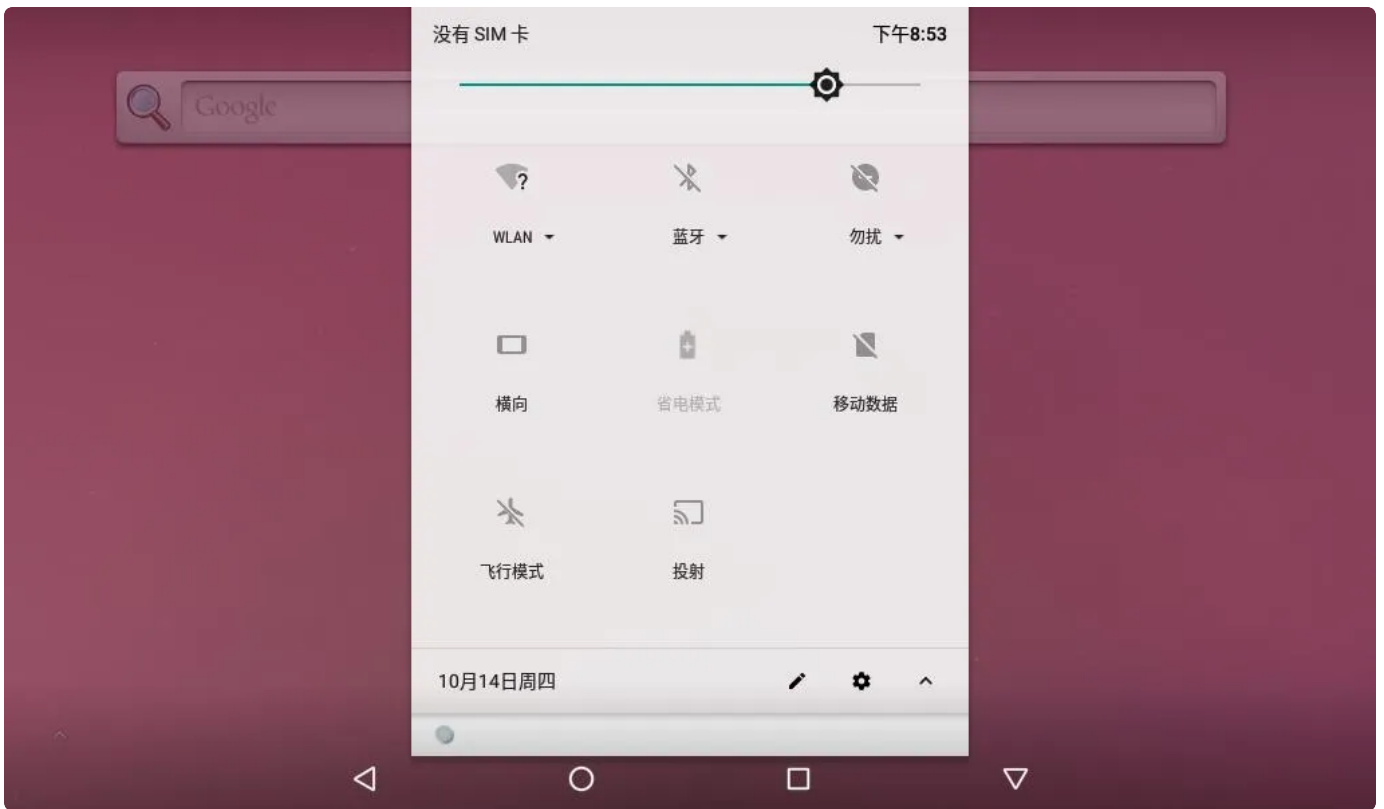
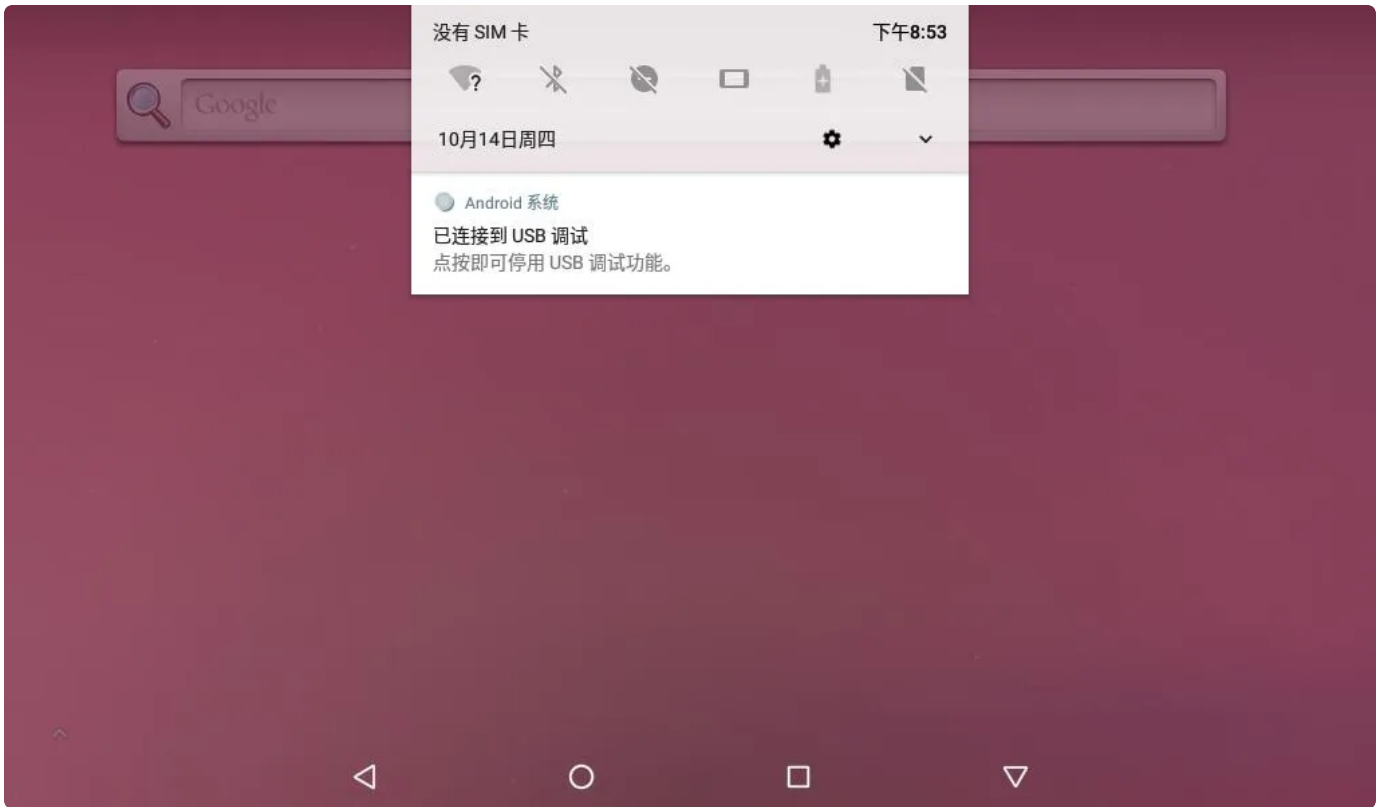
1: 返回

2: 直接返回桌面

3: 近期任务展示

4: 隐藏导航栏

桌面/菜单栏界面向下划即可看到功能控制栏和通知消息栏界面，再划动一下即可看到更详细的控制台界面，在此界面可更简单进入相关功能，如WIFI、蓝牙等



2.1.2 状态栏隐藏/显示

```
1  /*
2  * 参数 show
3  *  -- true: 显示,
4  *  -- false: 隐藏
5  * 参数 saveToSystem
6  *  -- true:保存为系统默认状态
7  *  -- false:不修改系统默认状态
8  */
9  private void SetStatusBar(boolean show, boolean saveToSystem){
10     if(show){
11         Intent intent = new Intent("android.ido.intent.action.statusbar.SH
12         OW");
13         intent.putExtra("hasexpand", true); //true: 可下拉, false:禁止下拉
14         intent.putExtra("save", saveToSystem); //true: 将此状态设置为系统默认状
15         态
16         sendBroadcast(intent);
17     }else{
18         Intent intent = new Intent("android.ido.intent.action.statusbar.HI
19         DE");
20         intent.putExtra("save", saveToSystem); //true: 将此状态设置为系统默认
21         状态
22         sendBroadcast(intent);
23     }
24 }
```

2.1.3 导航栏隐藏/显示


```
1  /*
2  * 参数 show
3  *  -- true: 显示,
4  *  -- false: 隐藏
5  * 参数 saveToSystem
6  *  -- true:保存为系统默认状态
7  *  -- false:不修改系统默认状态
8  */
9  private void SetNavigation(boolean show, boolean saveToSystem){
10     if(show){
11         Intent intent = new Intent("android.ido.intent.action.navigation.S
HOW");
12         intent.putExtra("save",saveToSystem); //true: 将此状态设置为系统默认状
态
13         sendBroadcast(intent);
14     }else{
15         Intent intent = new Intent("android.ido.intent.action.navigation.H
IDE");
16         intent.putExtra("save", saveToSystem); //true: 将此状态设置为系统默认
状态
17         sendBroadcast(intent);
18     }
19 }
```

2.2 鼠标

鼠标是一种很常见的输入设备，它可以对当前屏幕上的游标进行定位，并通过按键和滚轮装置对游标所经过位置的屏幕元素进行操作。

鼠标功能介绍：

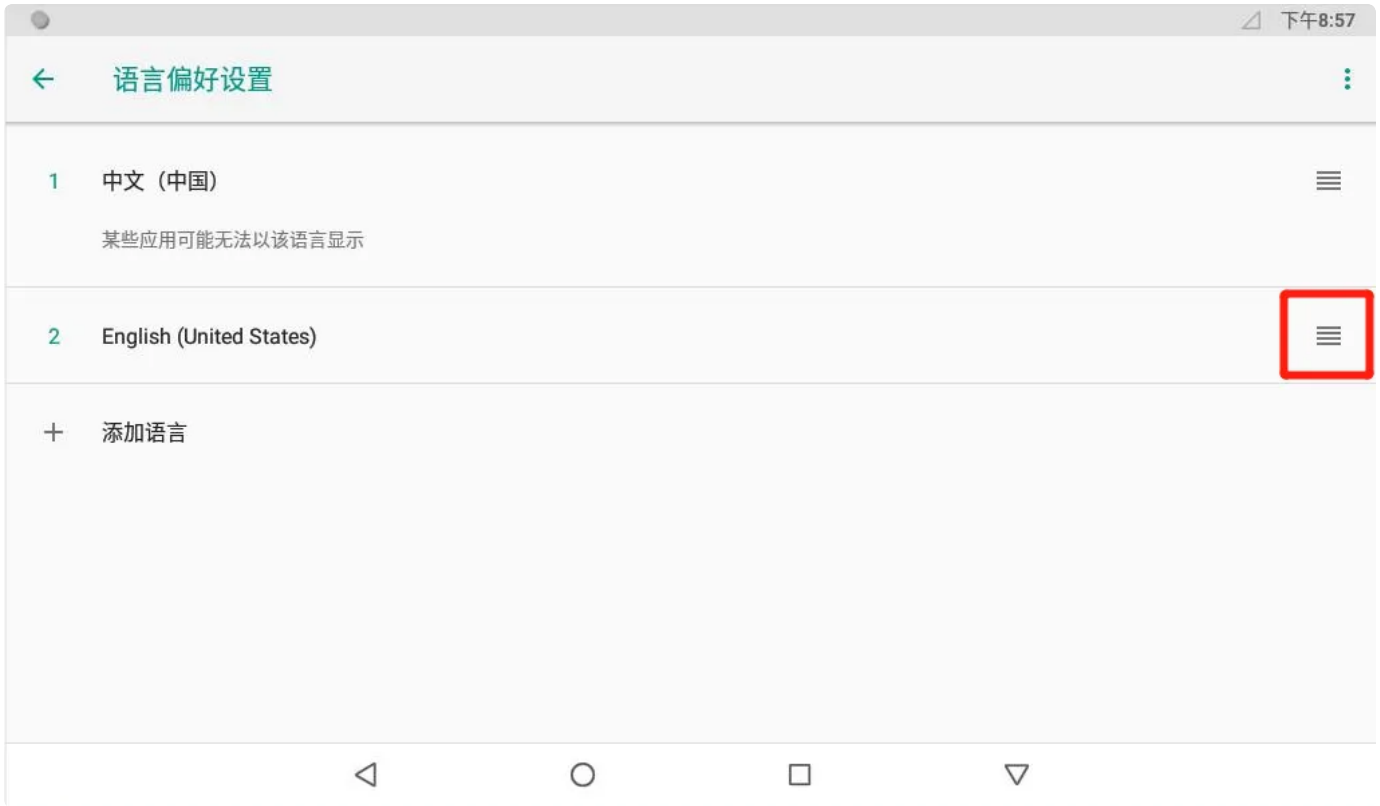


- 1: 左键，单击程序可打开程序，按住左键选定目标后拖拽鼠标可以移动目标
- 2: 中间的滚轴键，可在浏览网页时上下移动网页画面，单击程序也可打开程序
- 3: 右键，单击可返回

2.3 设置语言

菜单栏界面点击【设置】 -> 【系统】 -> 【语言和输入法】 -> 【语言】 -> 【添加语言】

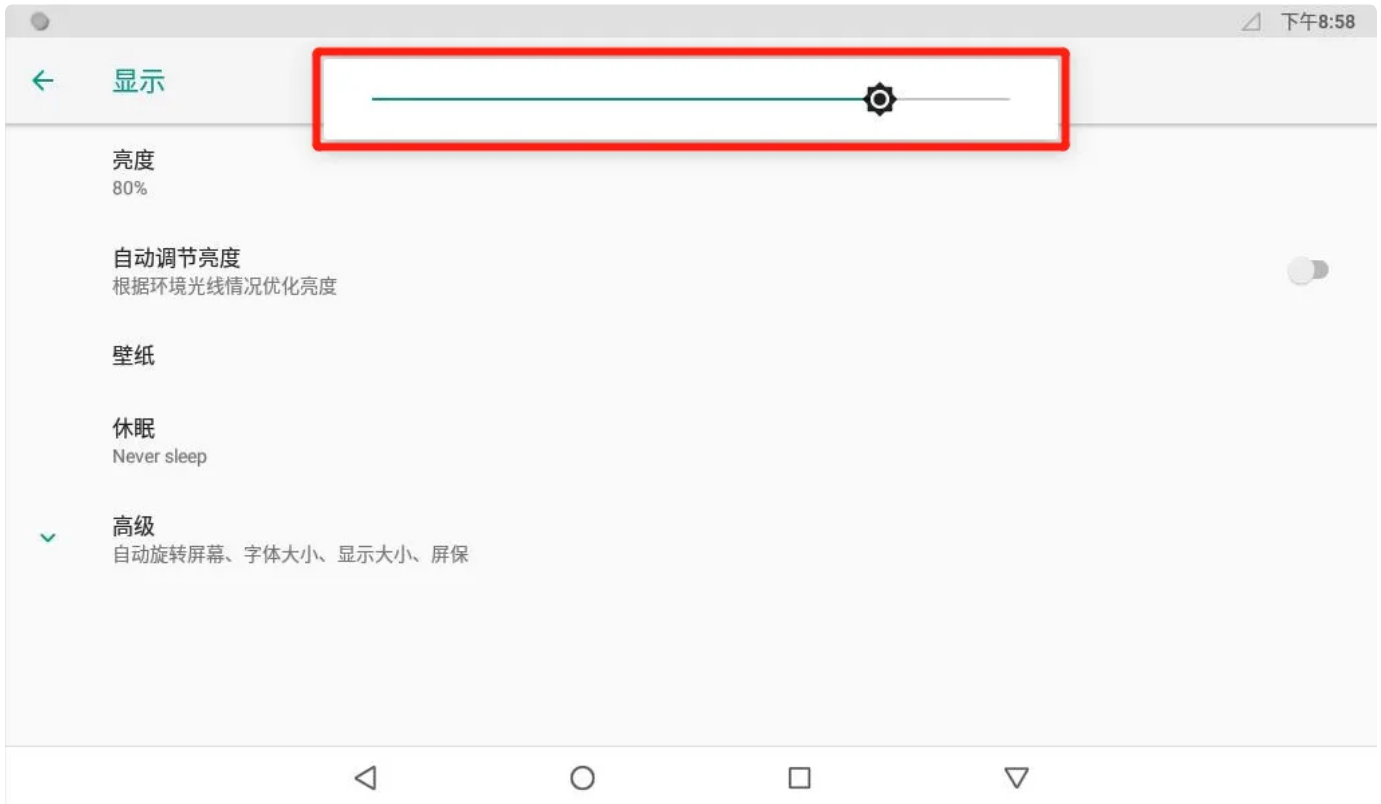
选择需要添加的语言，选择成功会返回【语言】界面，点击语言的右侧小图标往上拖即可设置成功



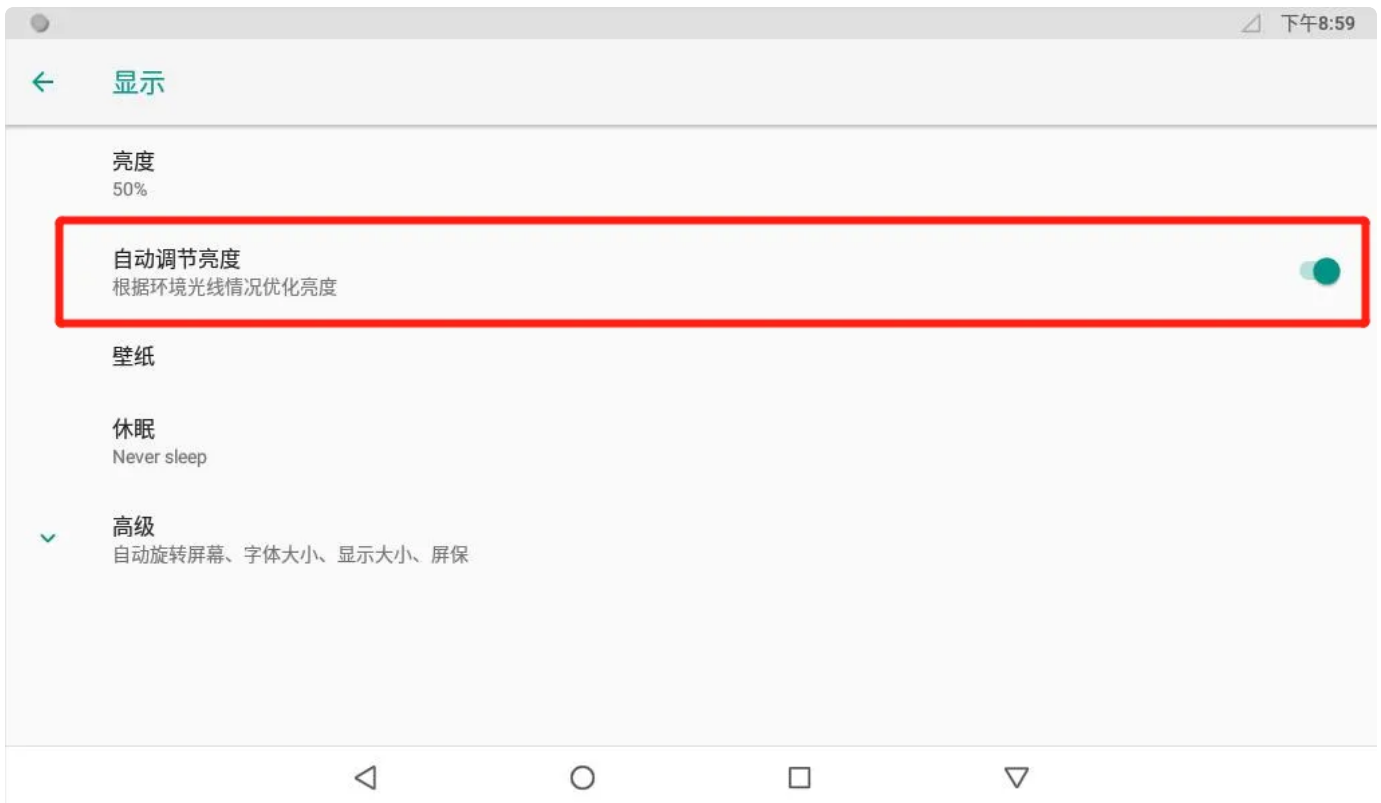
2.4 调节亮度

调节亮度有两种方法，可选择自己喜欢的方式进行选择

1、在菜单栏界面->设置->显示->亮度



可根据自己喜好选择开启【自动调节亮度】

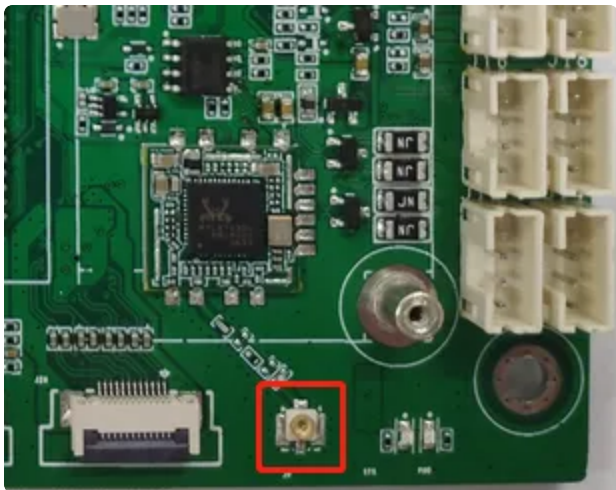


2、在桌面/菜单栏界面通过按住鼠标左键向下划两下可看到便捷功能栏界面，通过划动下面按键即可左右调节亮度



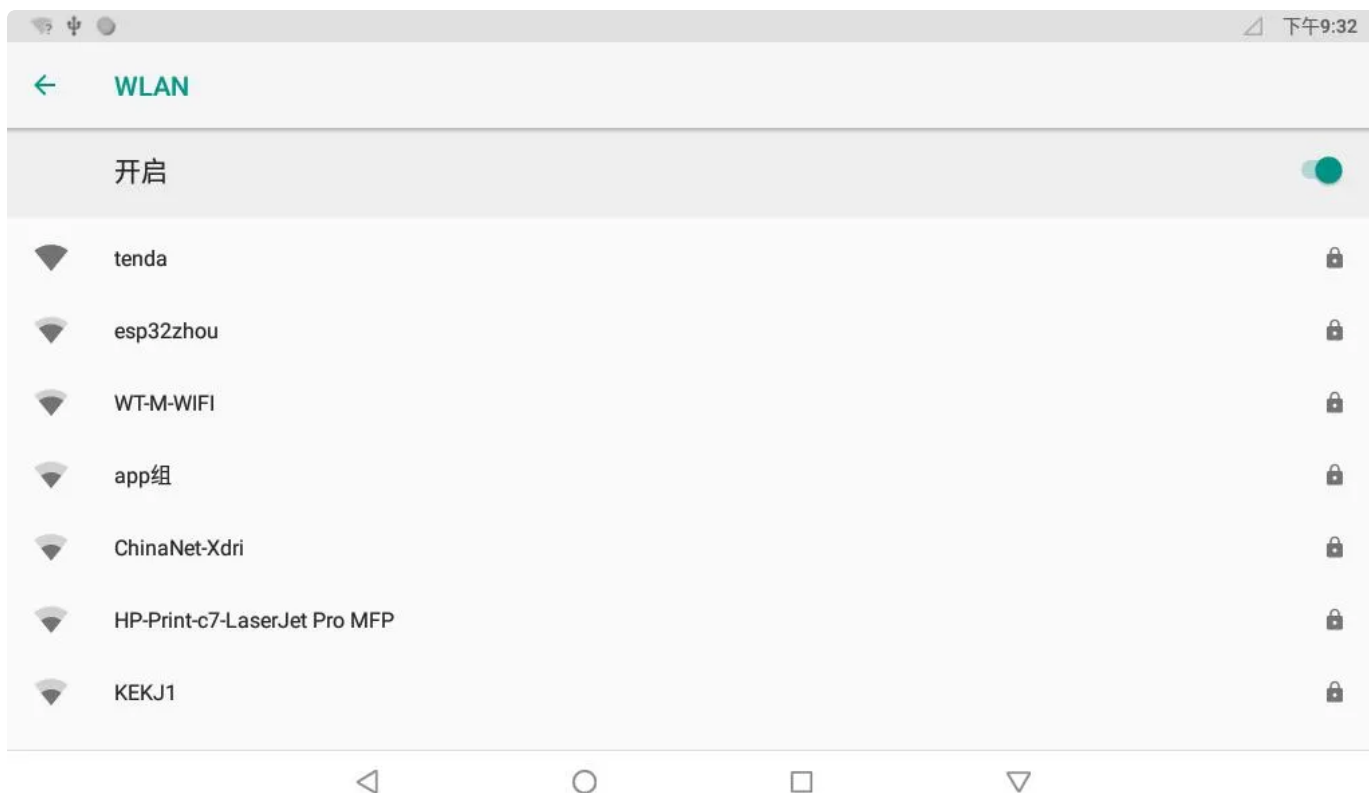
2.5 WIFI

测试时需要接上WIFI天线，以保证获得良好和稳定的信号。



2.5.1 设置界面连接WIFI热点

菜单栏界面点击【设置】 -> 【网络和互联网】 -> 【WLAN】

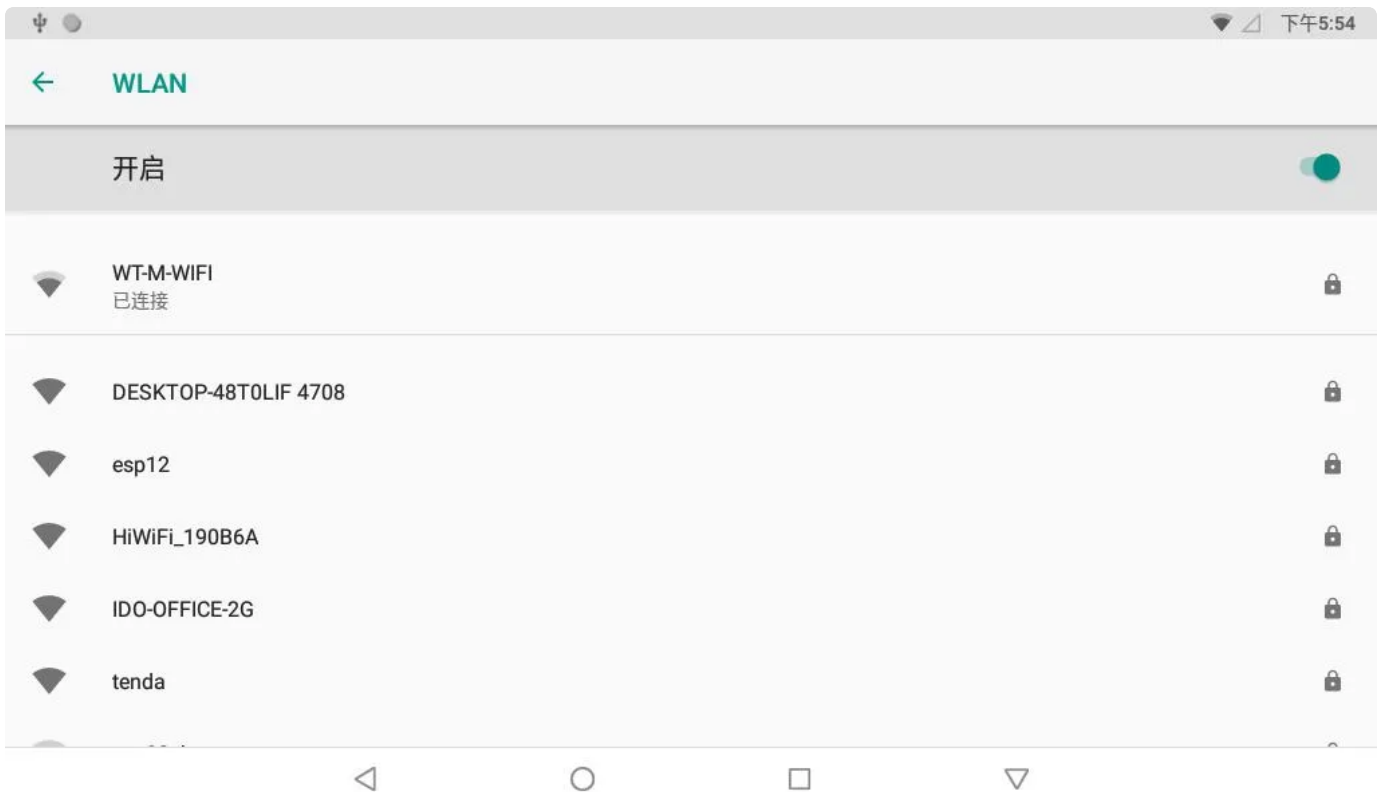


选择需连接的WIFI名称，输入对应密码即可连接成功



WIFI连接成功及连接成功标志（下图右上角所示）





2.5.2 软件中配置开启WIFI热点

```
Java |  
1 Intent mIntent1 = new Intent("android.ido.intent.action.wifiap");  
2 mIntent1.putExtra("enable",true);//true:开启AP, false:关闭  
3 mIntent1.putExtra("name","idoWifi");//无此参数热点默认名称为:wifiAp  
4 mIntent1.putExtra("password","12345678");//密码为不能小于8 个字符, 加密方式:WPA2  
   _PSK  
5 sendBroadcast(mIntent1);
```

可执行以下命令获取热点分配信息

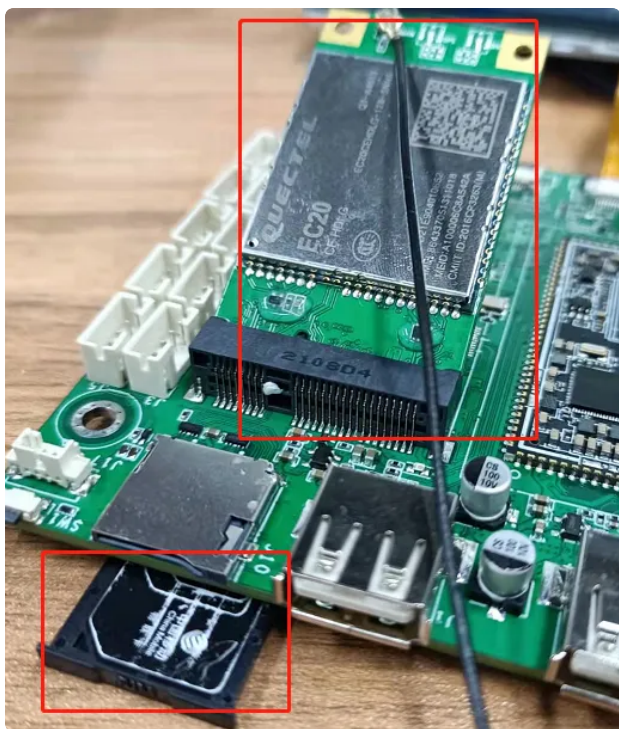
```
Java |  
1 cat /proc/net/arp
```

2.5.3 APP中开启和关闭WIFI

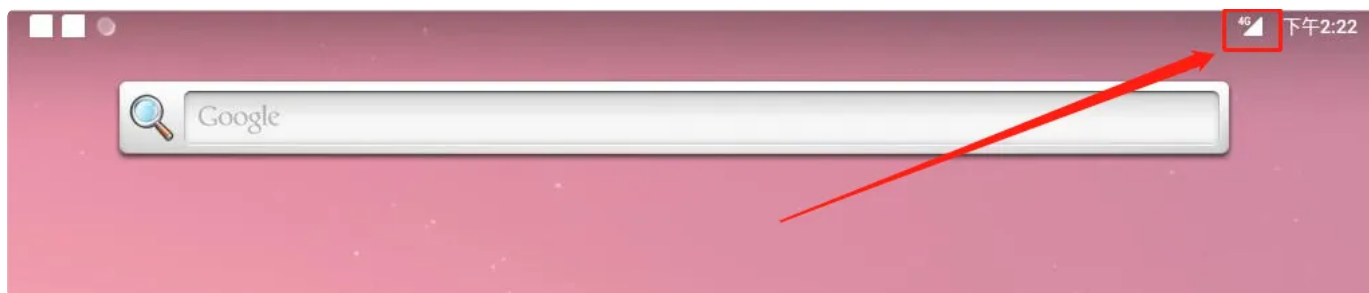
```
1 WifiManager wifimgr = (WifiManager) MainActivity.this.getSystemService(Context.WIFI_SERVICE);  
2 wifimgr.setWifiEnabled(true); //开启 true, 关闭 false
```

2.6 4G

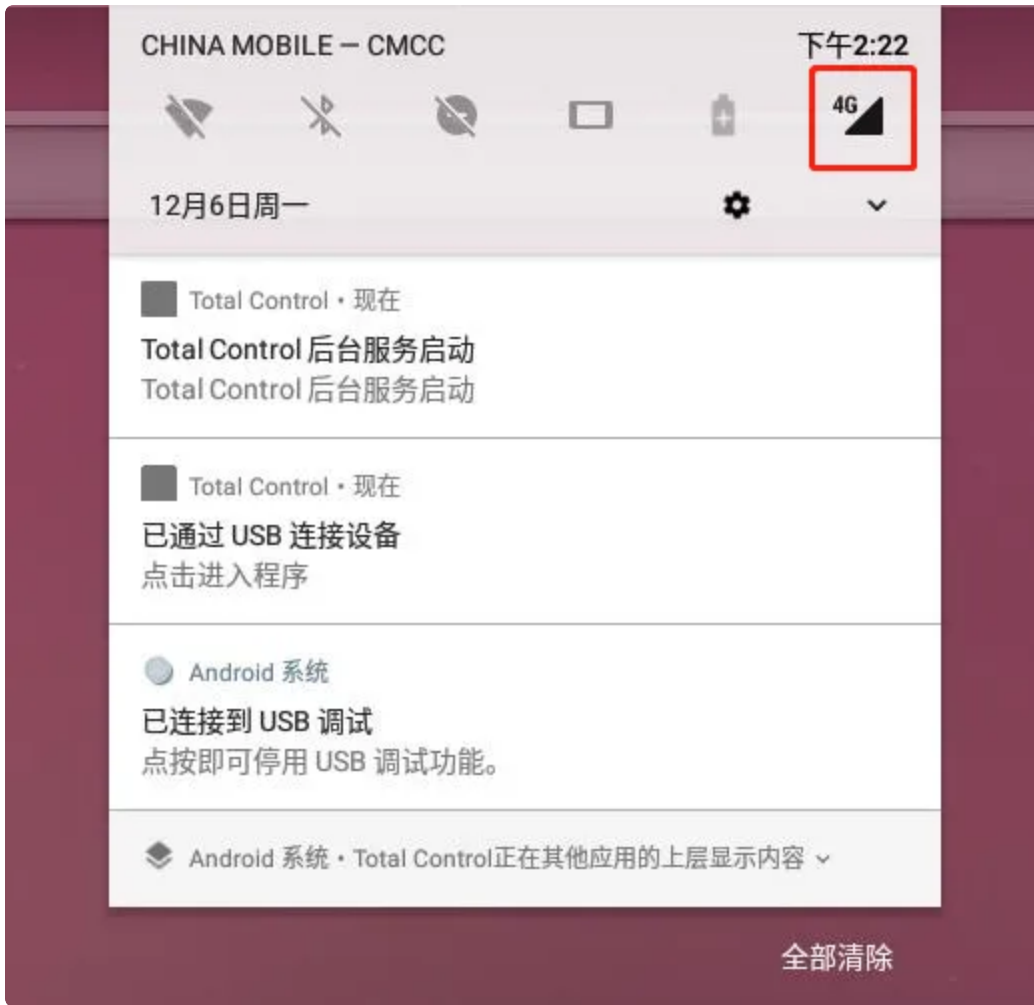
测试需要插入4G卡、EC20模块以及天线



会在桌面/菜单栏界面显示信号图标



也可以下拉便捷功能栏界面看到



然后再从菜单栏点击【闪电】在搜索栏输入一个网址即可测试4G网络



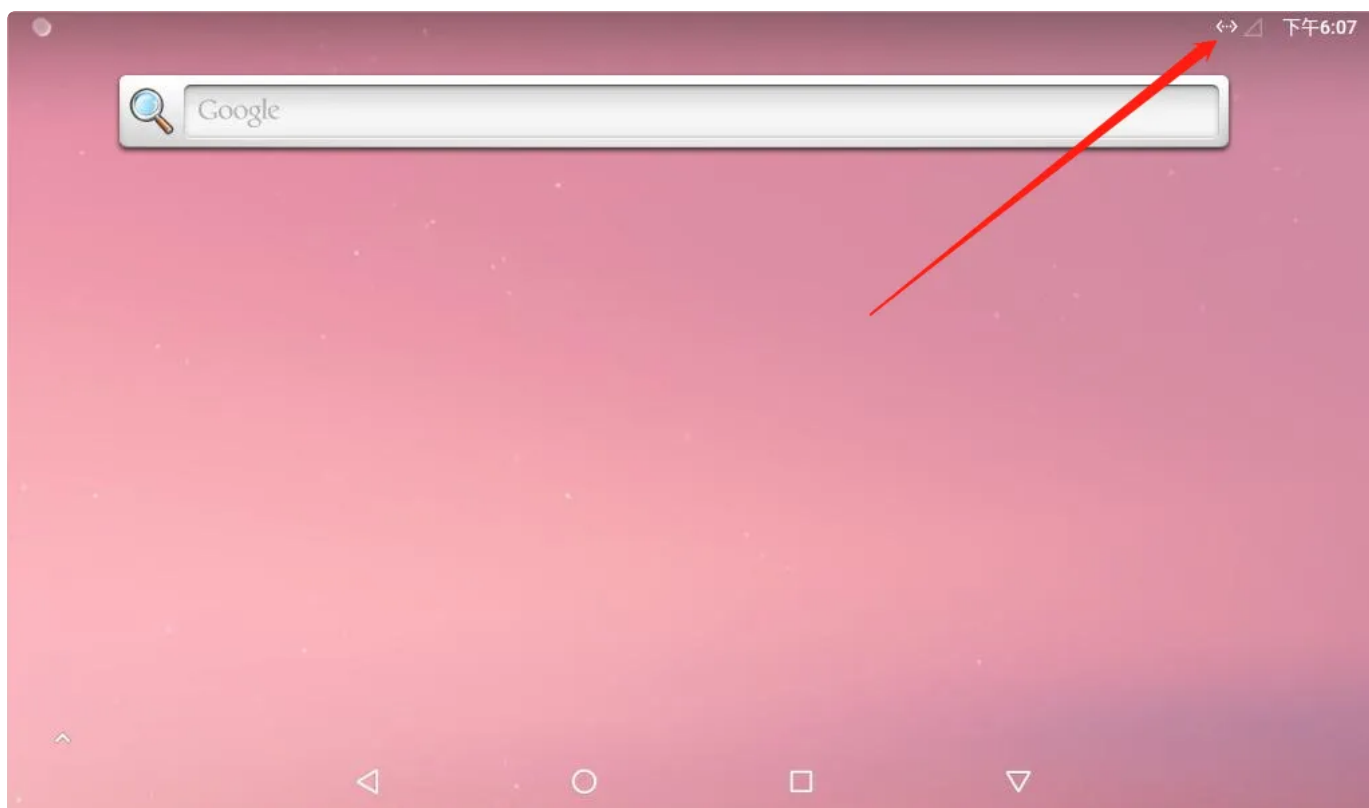
2.6.1 APP中开启和关闭4G数据

```
Java |  
1 public void setMobileDataState(boolean mobileDataEnabled) {  
2     try {  
3         TelephonyManager telephonyService = (TelephonyManager) getSystemService(Context.TELEPHONY_SERVICE);  
4  
5         Method setMobileDataEnabledMethod = telephonyService.getClass().getDeclaredMethod("setDataEnabled", boolean.class);  
6  
7         if (null != setMobileDataEnabledMethod) {  
8             setMobileDataEnabledMethod.invoke(telephonyService, mobileDataEnabled);  
9         }  
10    } catch (Exception ex) {  
11        Log.e(TAG, "Error setting mobile data state", ex);  
12    }  
13 }
```

2.7 以太网口

开发板有1路千兆以太网接口为 eth0。以太网接口默认支持DHCP，只需要将以太网接口连接路由器即可为开发板动态分配 IP 地址。

插上网线后，桌面显示如下图图标即可说明网线识别成功

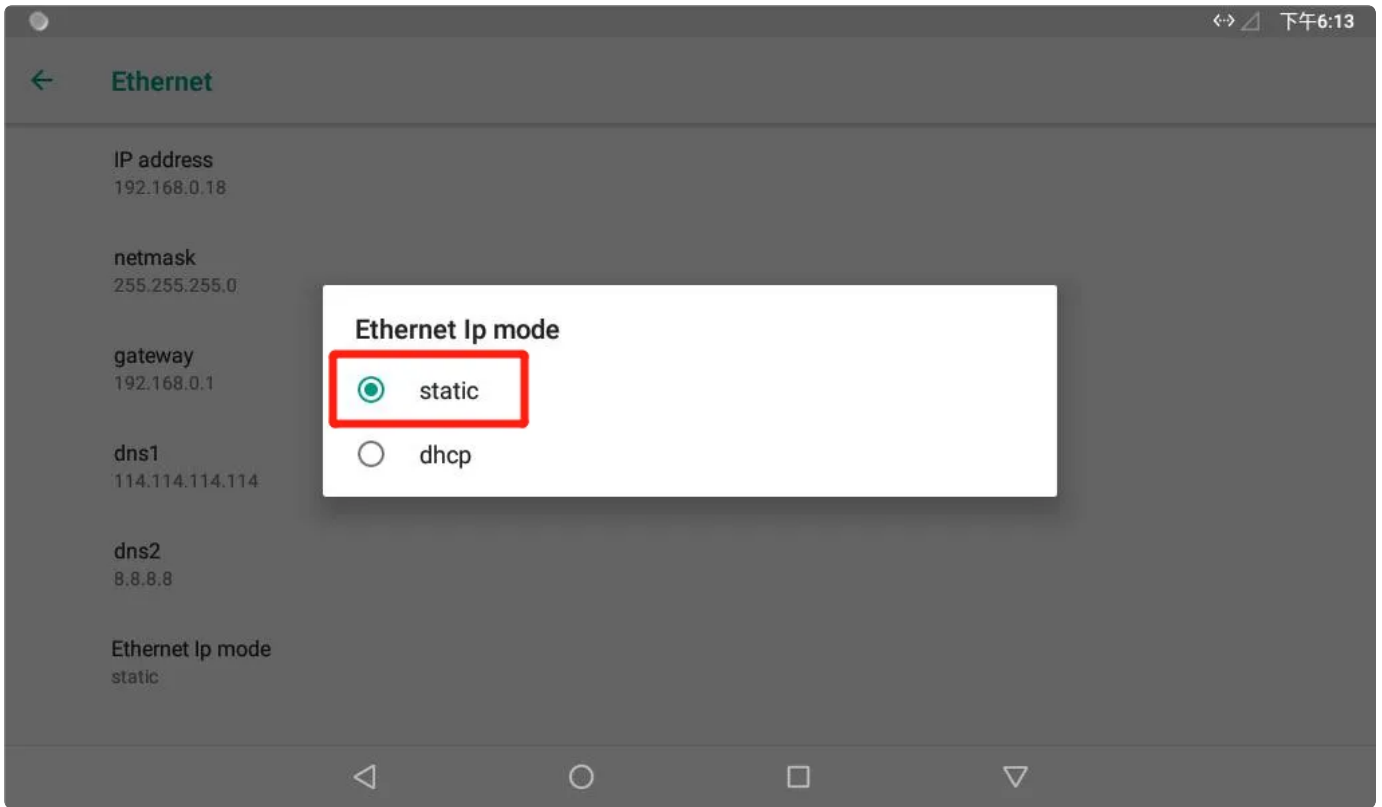


菜单栏界面点击【闪电】在搜索栏输入一个网址即可测试网线网络

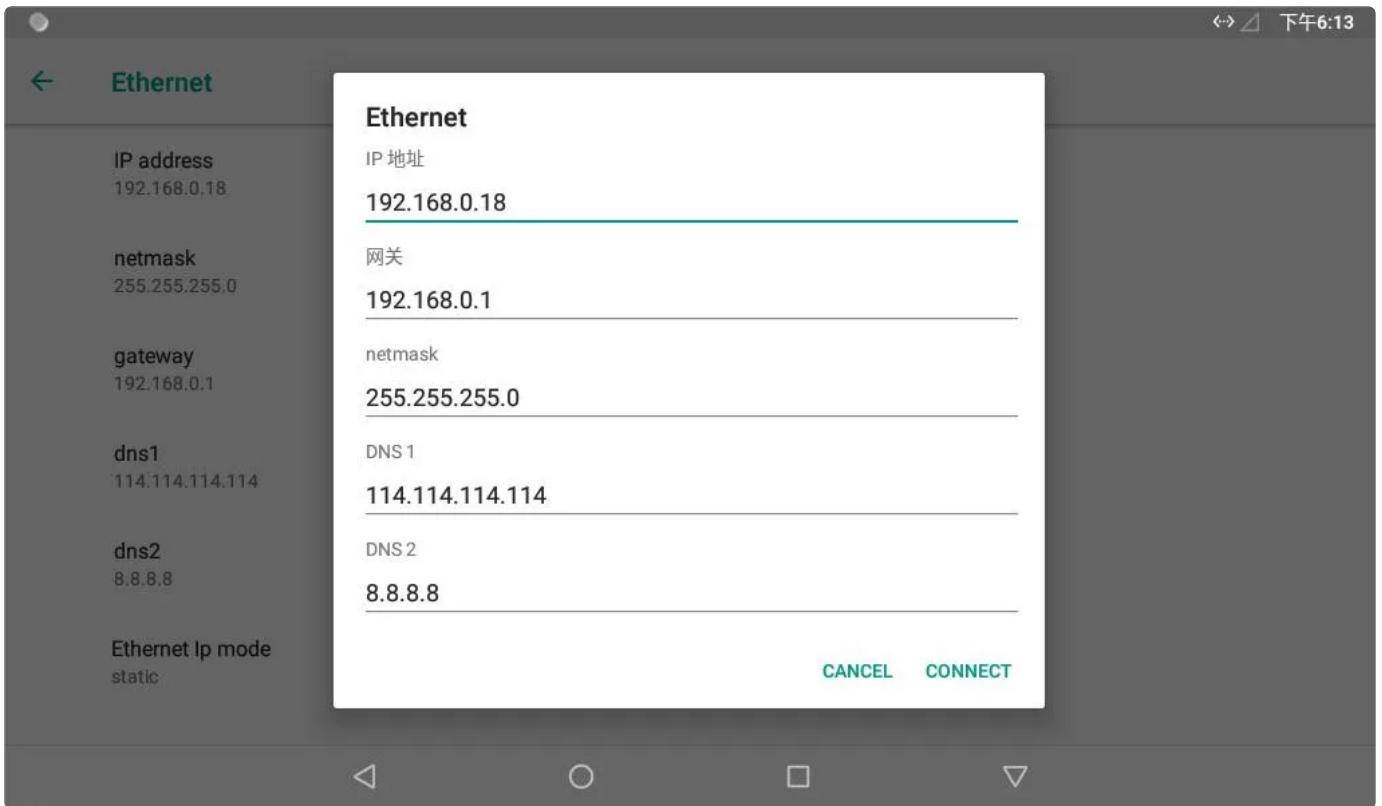


2.7.1界面设置以太网静态IP

菜单栏界面点击设置->网络和互联网->Ethernet->Ethernet Ip mode, 选择【static】即可设置, 如下图所示



点击static设置



2.7.2 代码中修改静态/动态IP地址

静态IP地址

```
1 Intent mIntent1 = new Intent("android.ido.intent.action.ethernet");
2 mIntent1.putExtra("mode","Static");
3 mIntent1.putExtra("ipAddr","192.168.0.103");
4 mIntent1.putExtra("netmask","255.255.255.0");
5 mIntent1.putExtra("gateway","192.168.0.1");
6 mIntent1.putExtra("dns1","114.114.114.114");
7 mIntent1.putExtra("dns2","8.8.8.8"); //可以省略
8 sendBroadcast(mIntent1);
```

动态IP地址

```
1 Intent mIntent1 = new Intent("android.ido.intent.action.ethernet");
2 mIntent1.putExtra("mode","DHCP");
3 sendBroadcast(mIntent1);
```

断开和重连

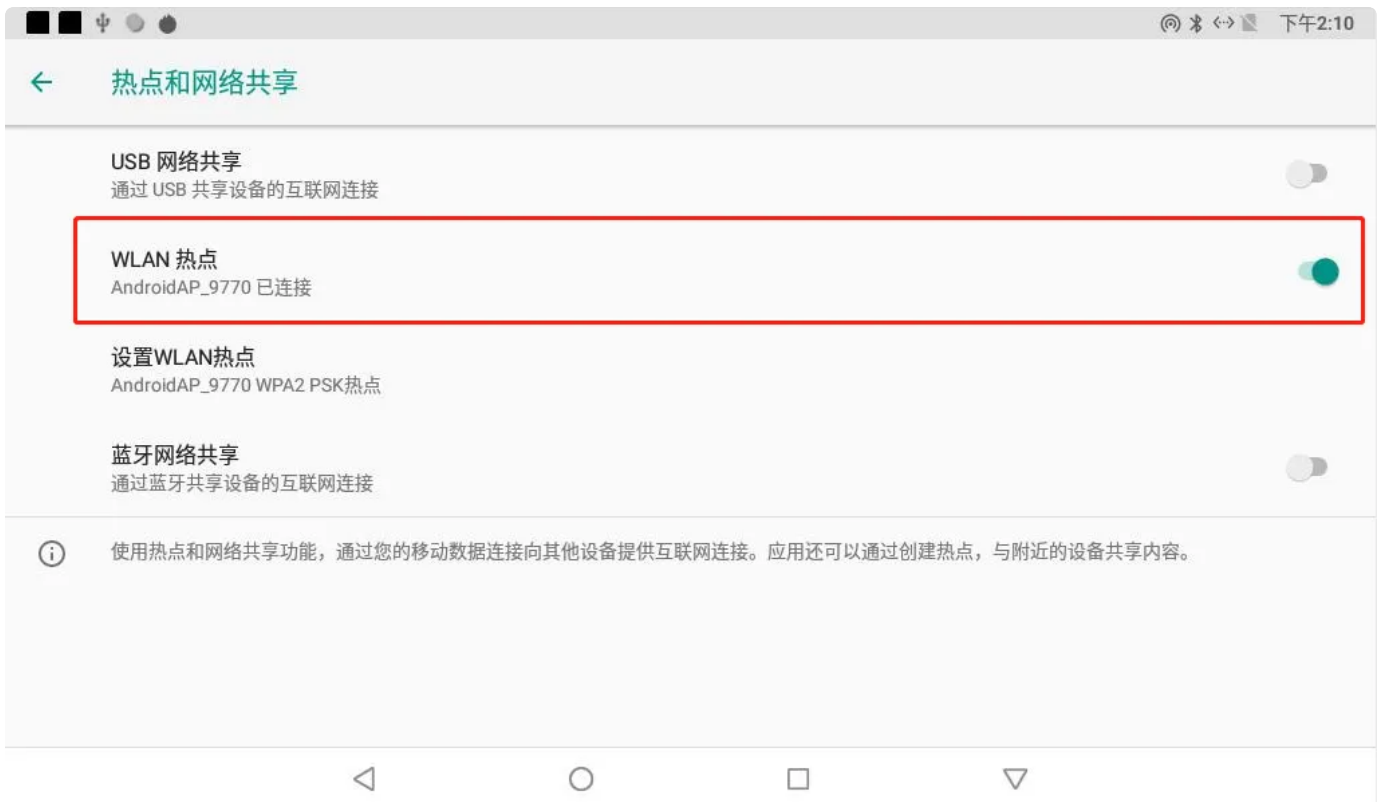
```
1 Intent mIntent1 = new Intent("android.ido.intent.action.ethernet");
2 mIntent1.putExtra("enable","true"); // "false":断开, "true":重新连接
3 sendBroadcast(mIntent1);
```

2.8 热点和网络共享

2.8.1 WLAN热点共享

WLAN热点共享需要有4G网络，插入4G卡以及4G模块这些外围设备后，菜单栏界面点击设置->网络和互联网->热点和网络共享

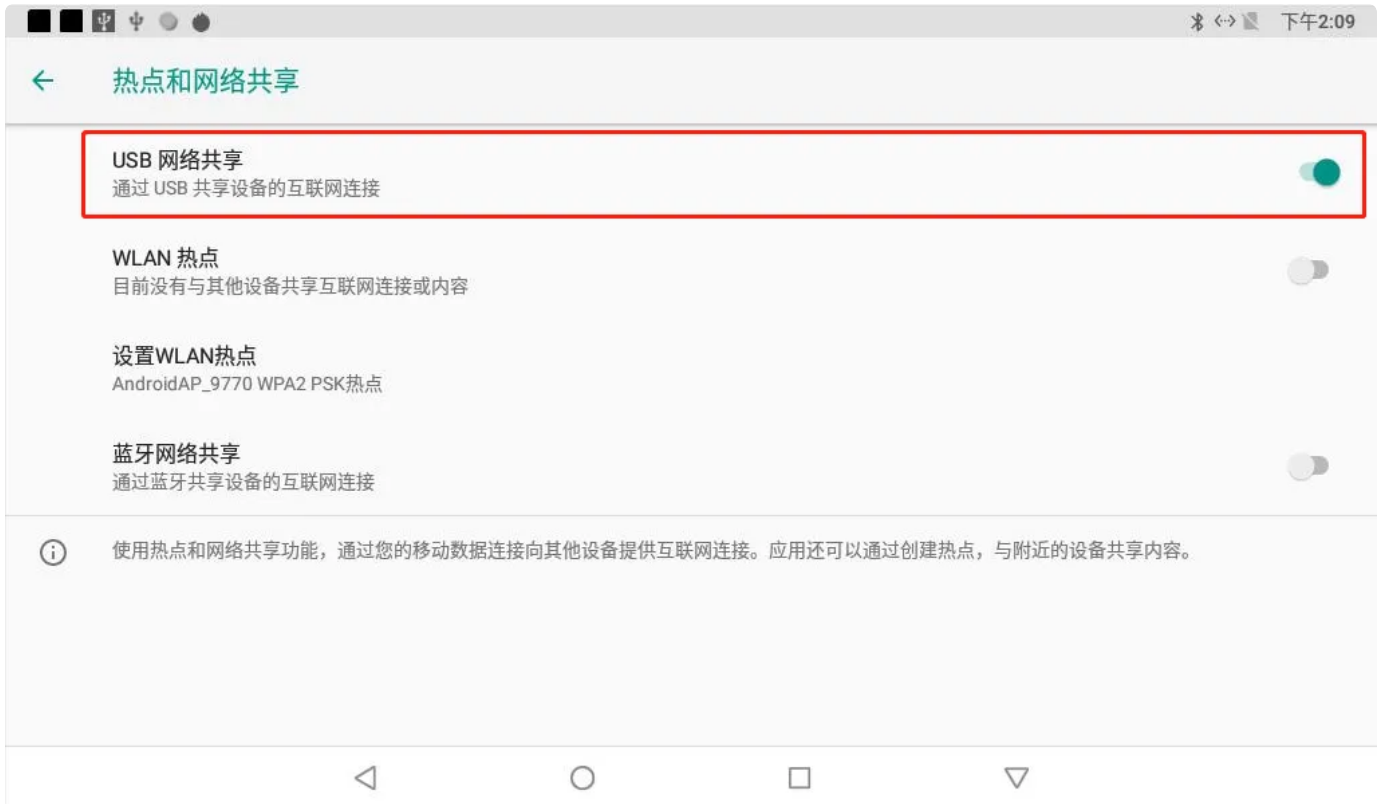
点击开启【WLAN 热点】，在此【设置WLAN热点】中可以自由设置【热点名称】、【热点密码】等



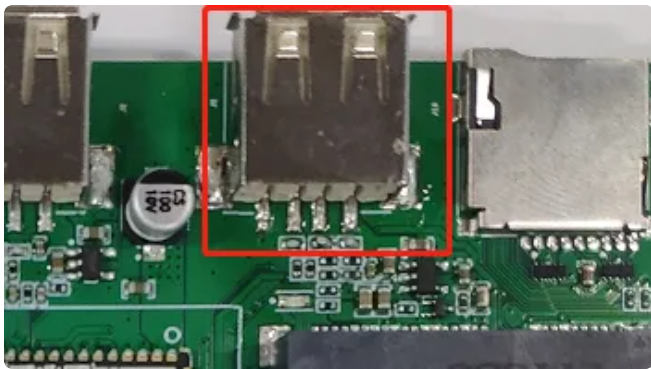
2.8.2 USB Network Sharing

Menu bar interface click Settings -> Network and Internet -> Hotspots and Network Sharing

开启【USB网络共享】

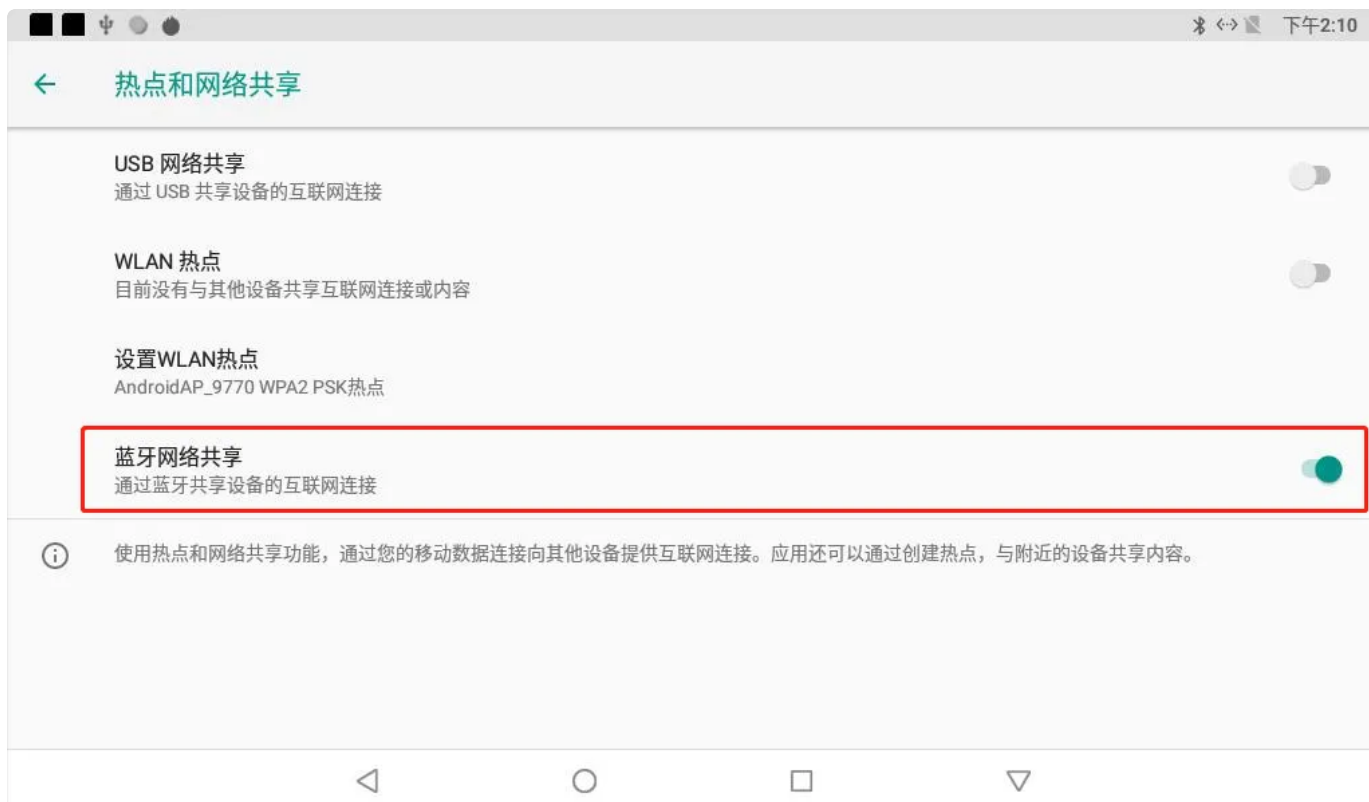


开发板插入可上网的网线后，USB数据线一头插入到板子USB OTG接口(如下图所示)，另一头插入电脑，电脑即可共享板子的互联网连接



2.8.3 蓝牙网络共享

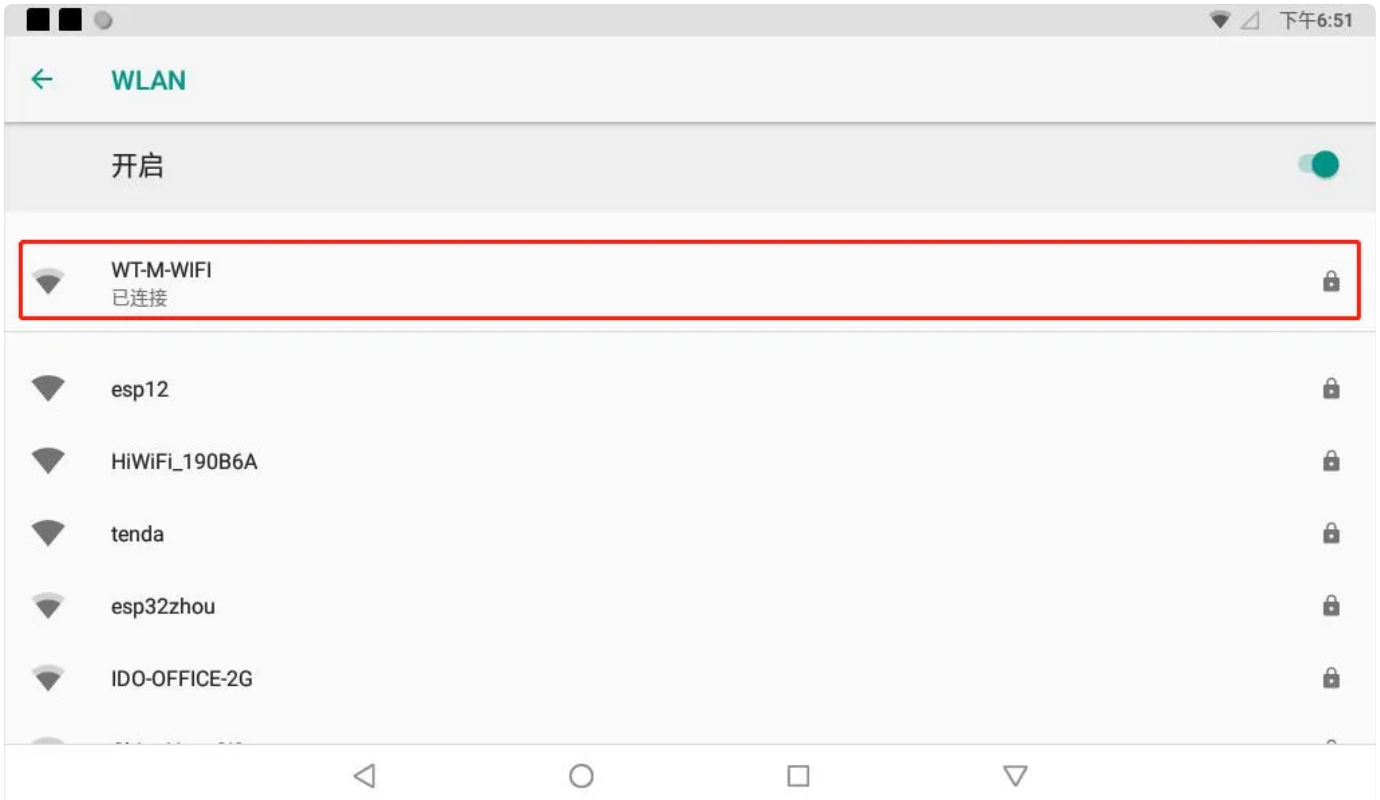
蓝牙网络共享开启并开发板与手机/电脑连接上蓝牙后，手机/电脑即可使用当前板的网络数据菜单栏界面点击设置->网络和互联网->热点和网络共享，开启【蓝牙网络连接】



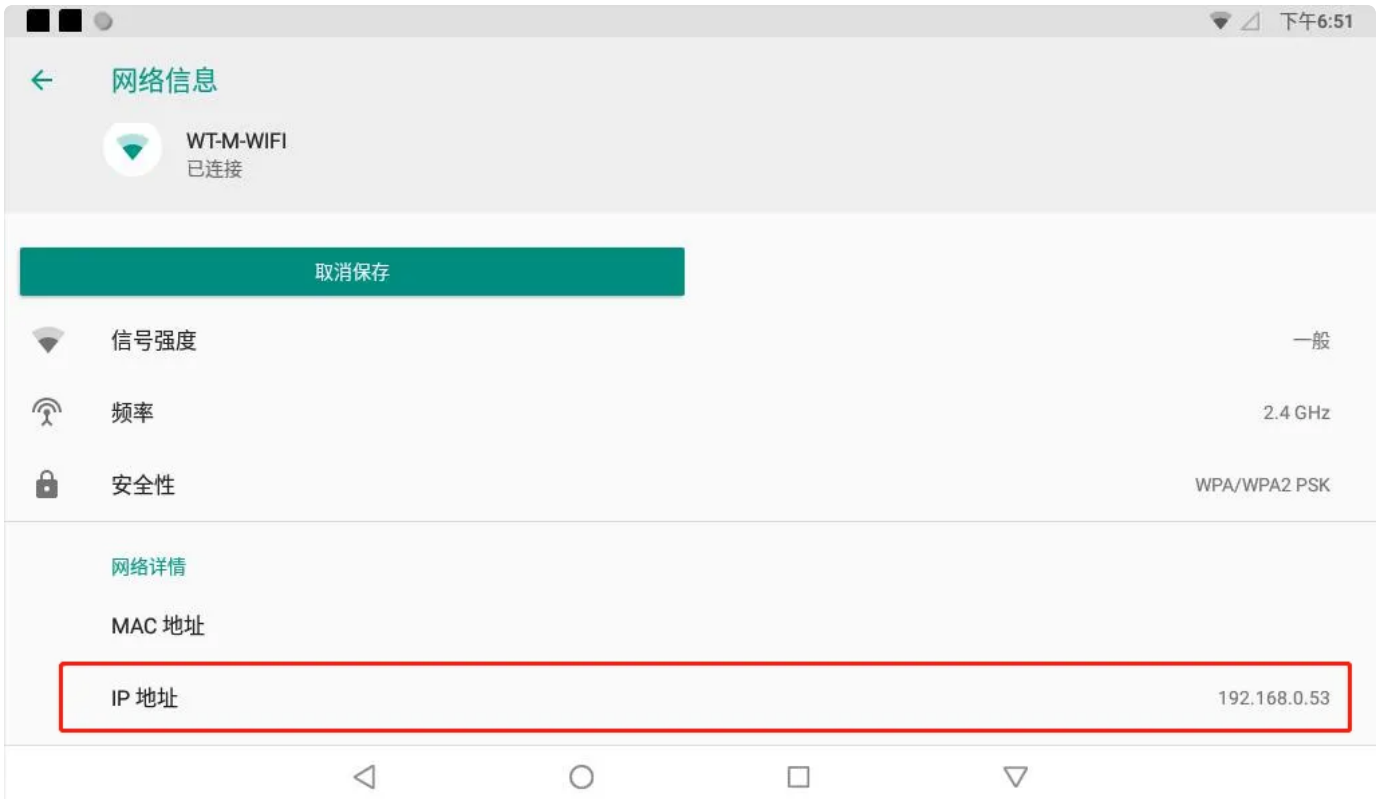
2.9 查看 IP

2.9.1 WIFI

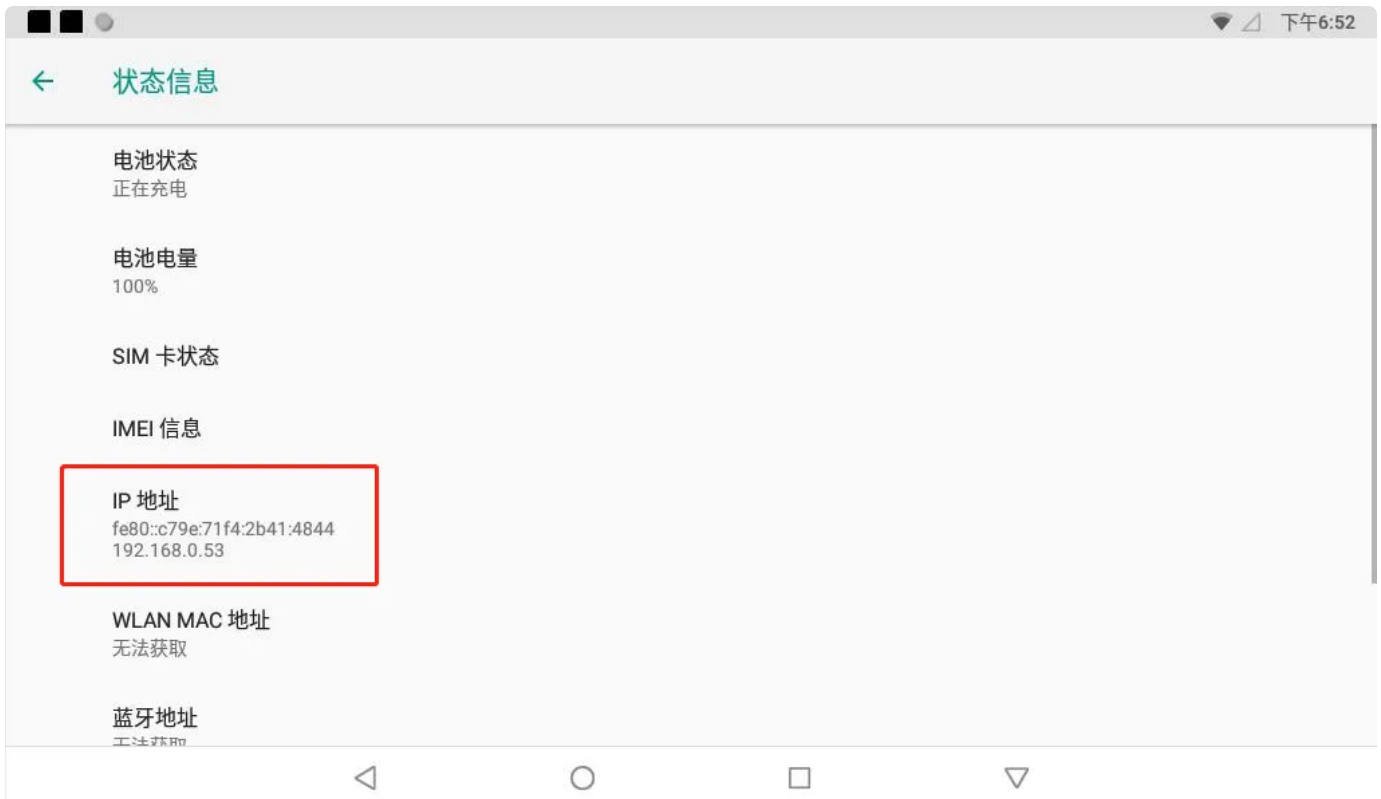
设备连接WIFI后，菜单栏界面点击设置->网络和互联网->WLAN，点击已连接WIFI



进去后点击【高级】，即可查看WIFI的IP及网关信息等

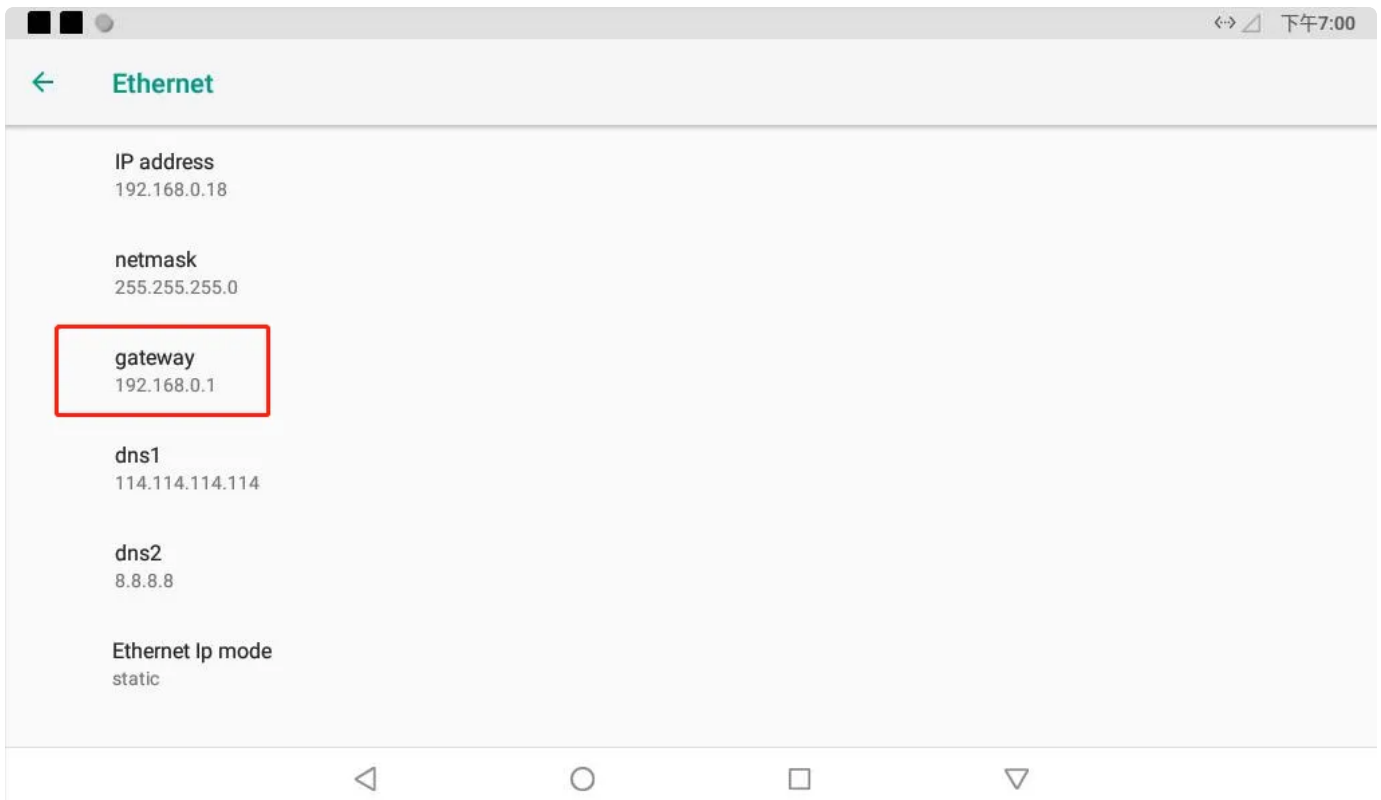


也可点击设置->系统->关于设备->状态信息，在这里也可查看IP



2.9.2 以太网

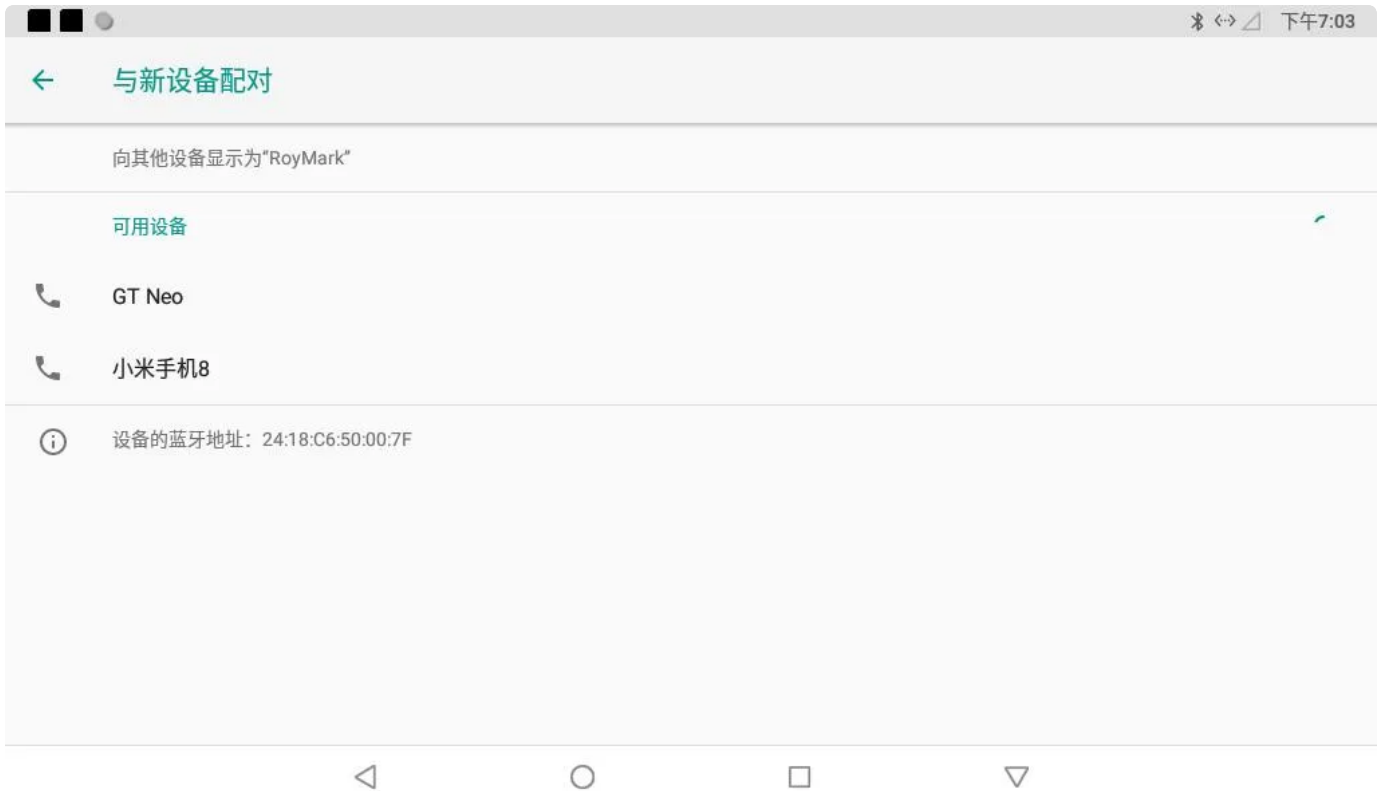
插上网线后，菜单栏界面点击设置->网络和互联网->Ethernet，即可查看以太网IP

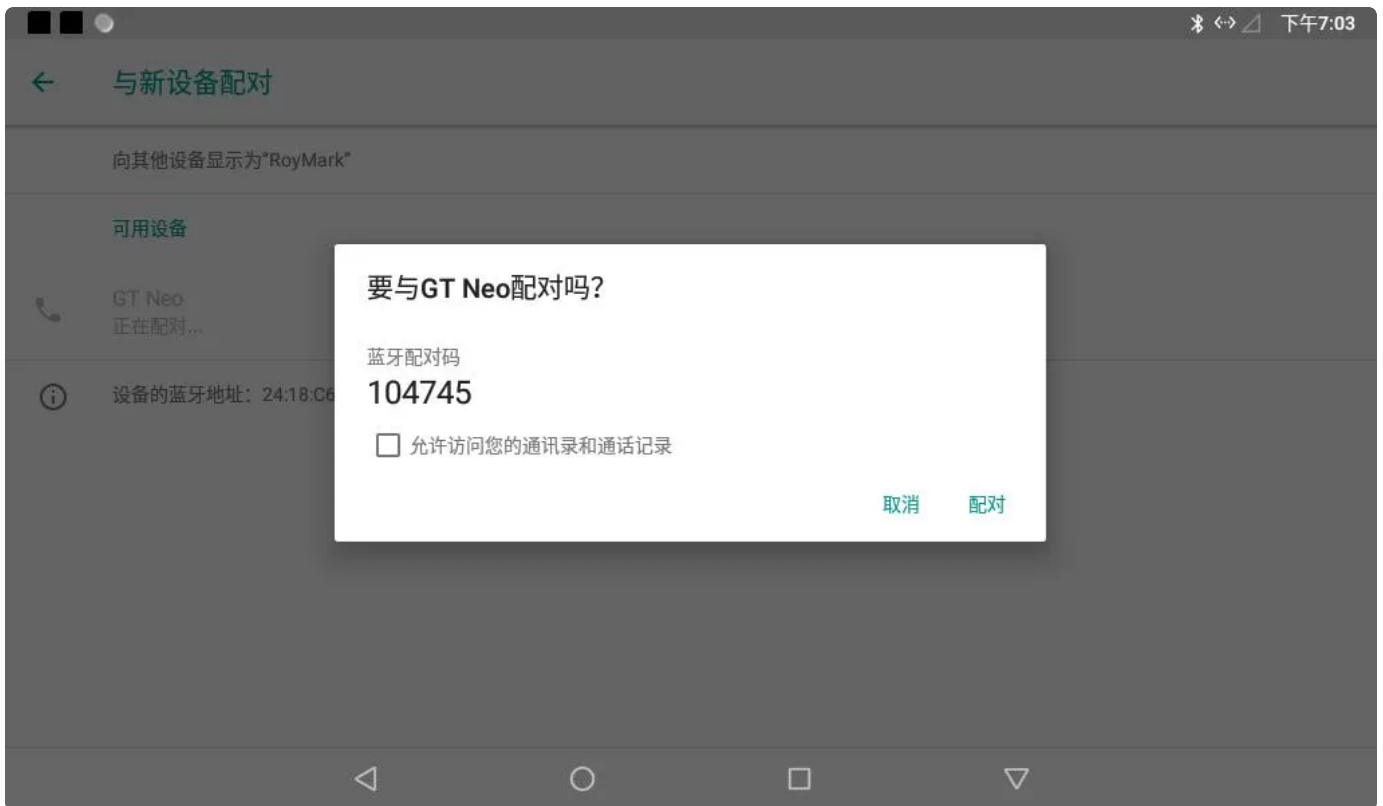


2.10 蓝牙（经典蓝牙协议）

菜单栏界面点击【已连接的设备】 -> 【与新设备配对】

即可扫描到附近的蓝牙设备，选择需要连接的设备即可根据配对信息进行连接

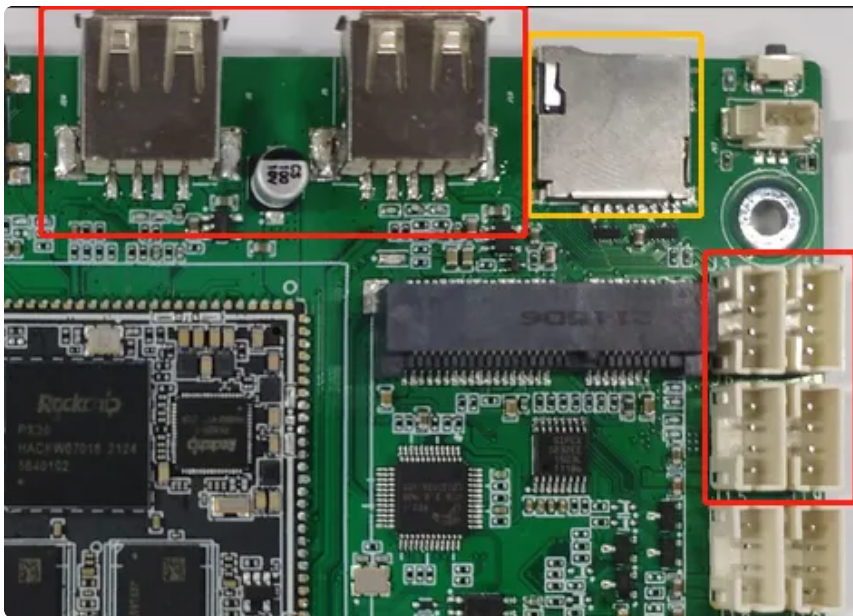




配对成功后开发板即可通过蓝牙与手机相互传输文件

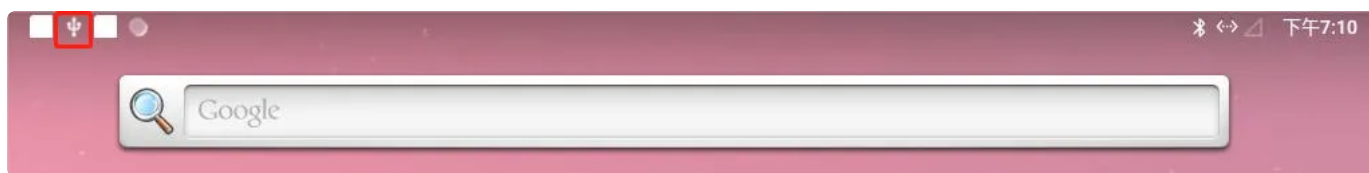
2.11 SD卡和USB存储设备

开发板可通过插入U盘和SD卡进行文件传输

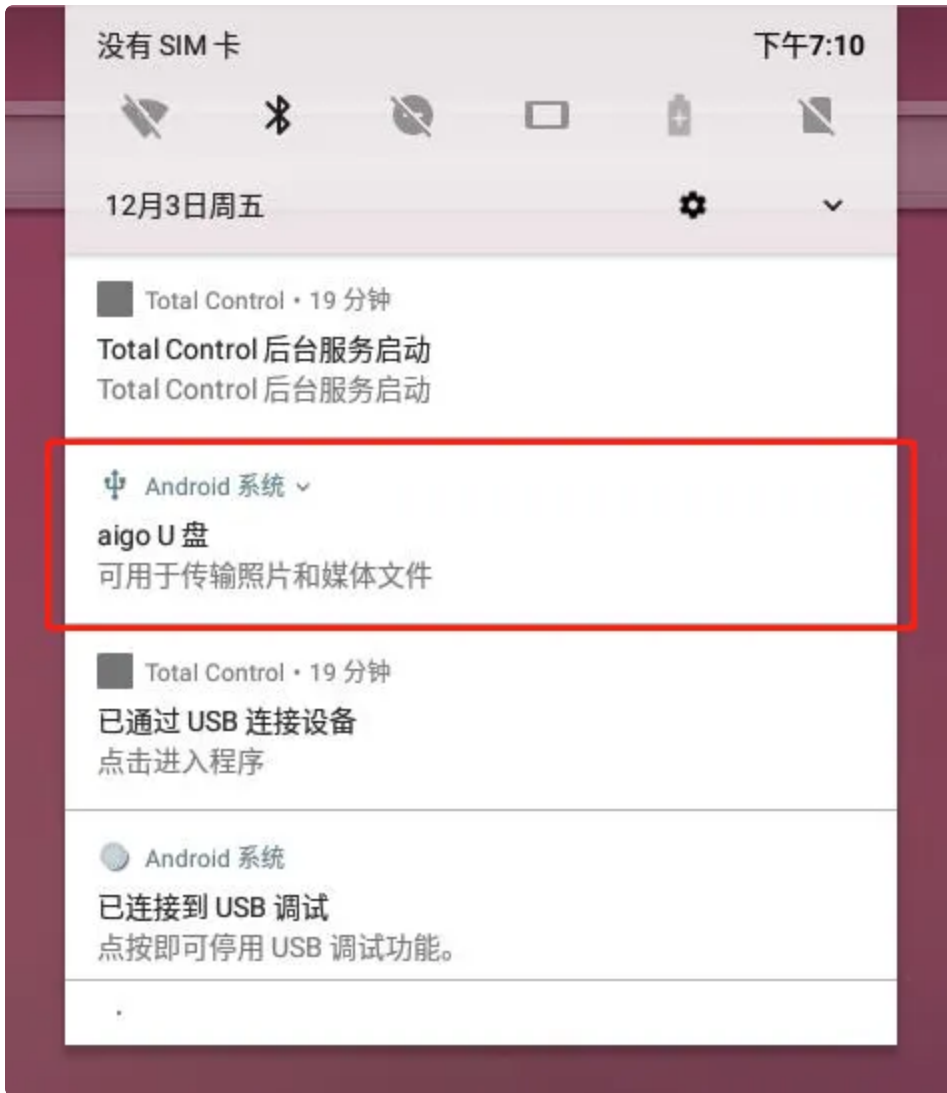


红框是USB存储接口，黄框是SD卡接口

插入U盘后开发板即可识别到USB设备（SD卡方法一样）



U盘成功识别标志（左上角）



2.12 APK安装与卸载

2.12.1 通过U盘/SD卡安装APK

将APK文件拷贝到U盘或者SD卡，然后将其插入到板子，识别到设备后通过【菜单栏】找到【资源管理器】，进入到APK文件放置目录，双击该文件进行安装



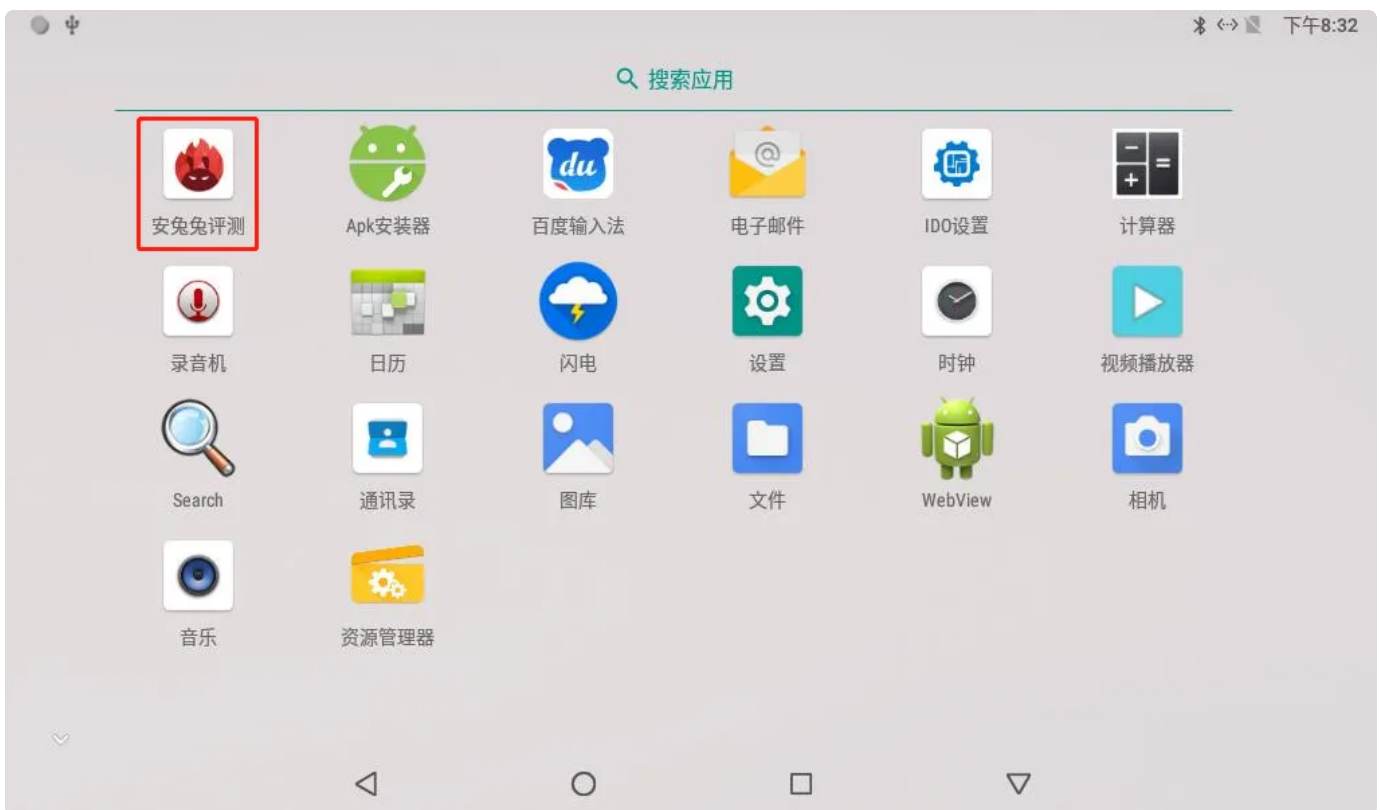
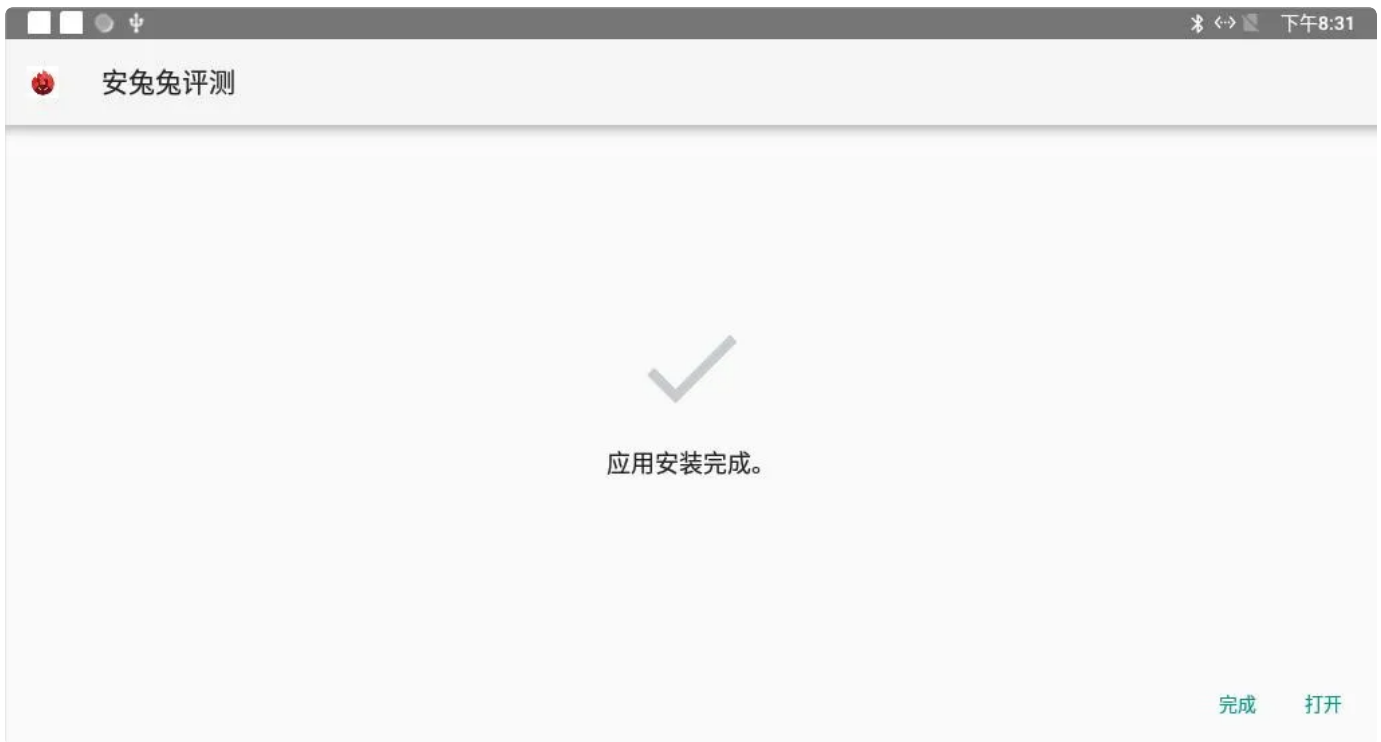
注意：首次安装会弹出您是否选择允许【安装未知应用】，选择允许来自此来源的应用即可，否则将无法安装



点击安装

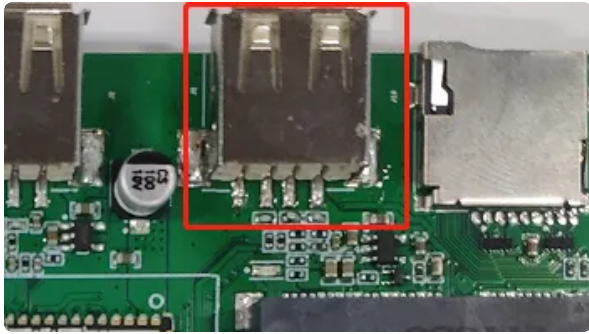


安装成功后即可在菜单栏找到【安兔兔评测】APP

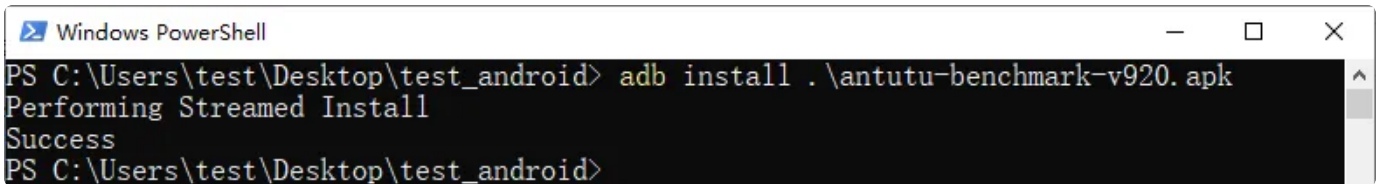


2.12.2 ADB安装

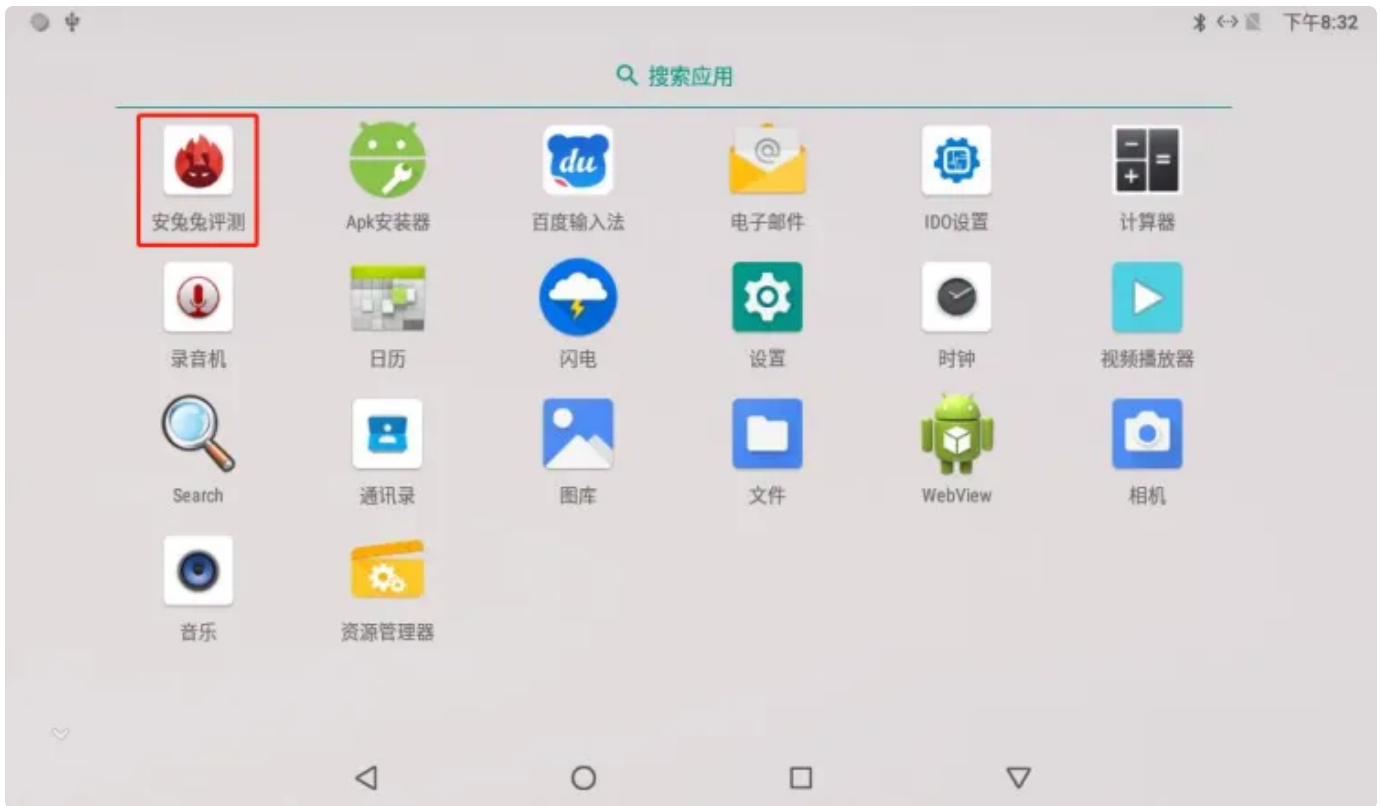
使用USB-A双公头数据线连接电脑与开发板以下USB OTG接口



打开windows的控制台，输入adb install +"APK路径"，按enter键安装，如下图所示



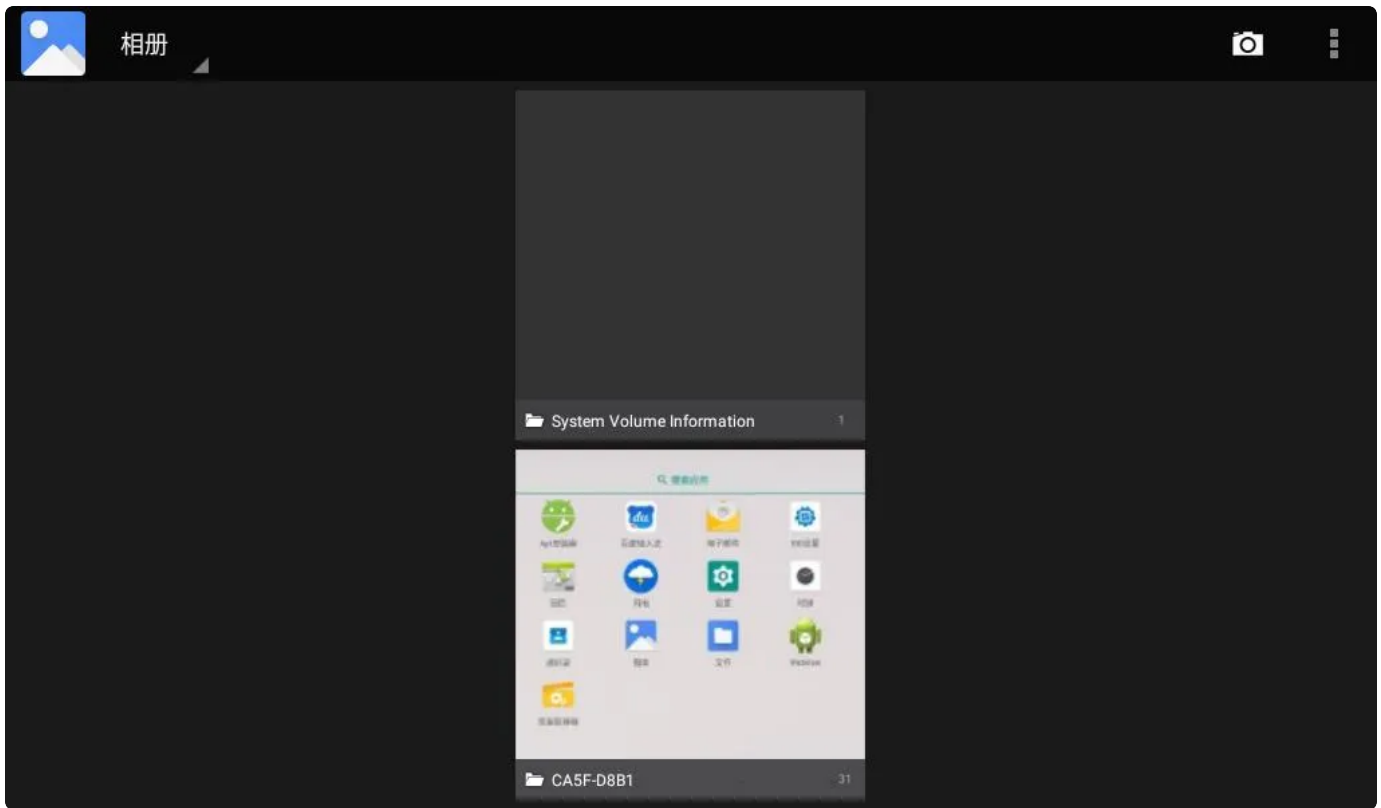
安装成功后即可在【菜单栏界面】找到刚安装的APP



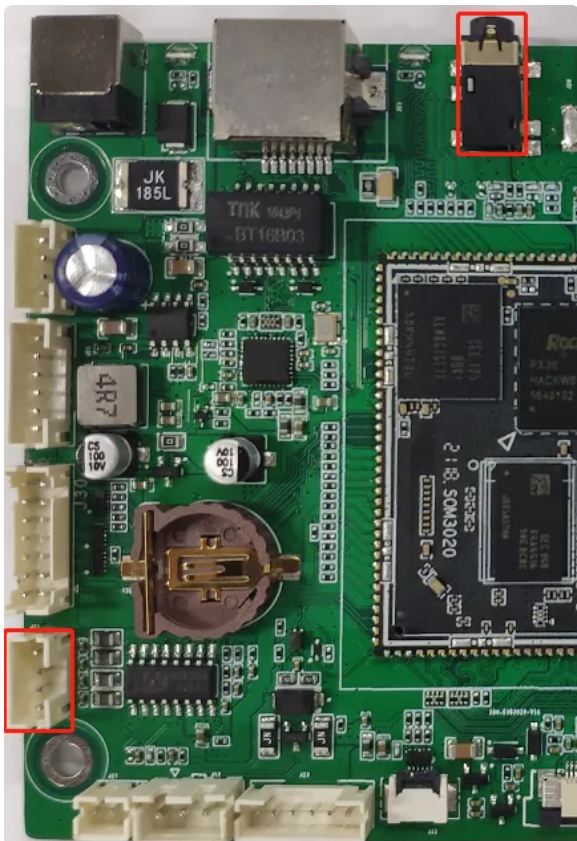
2.13 查看图片与播放视频

提前将图片和视频文件通过U盘或者SD卡拷贝到开发板

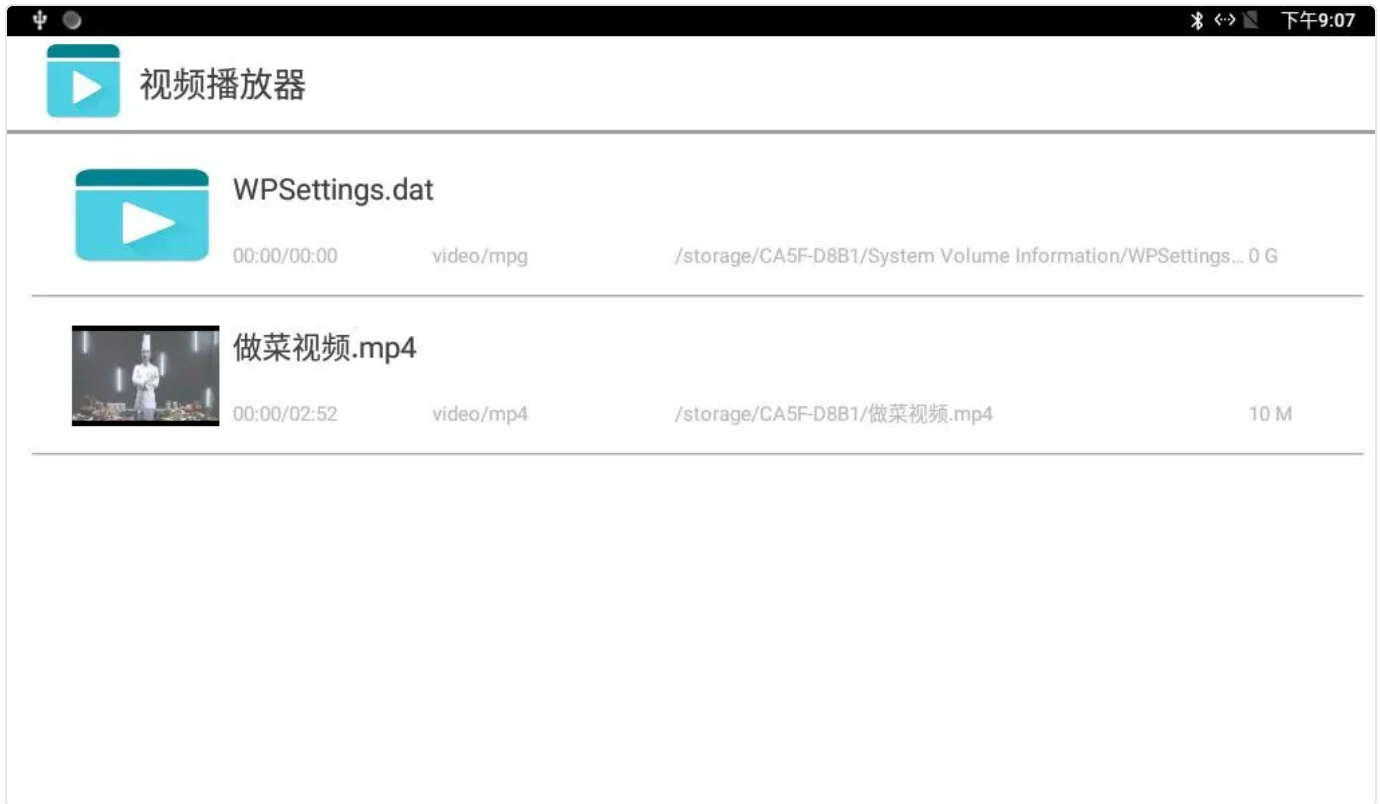
菜单栏点击【图库】即可查看系统内部或者U盘或者SD卡的图片信息



如需听到视频声音，需插入耳机或者喇叭即可听到视频播放声音，接口为下图红框处

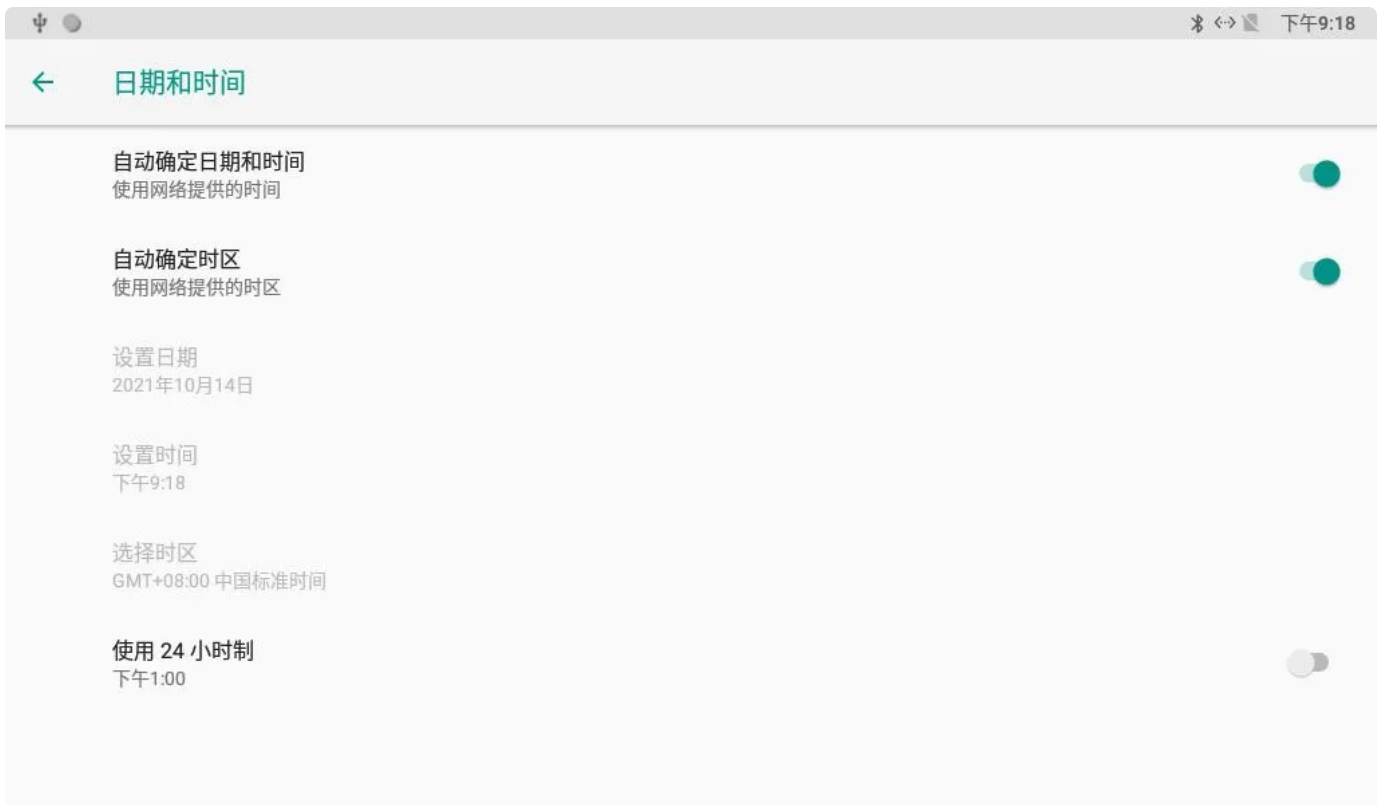


菜单栏点击【视频播放器】即可查看相关的视频文件

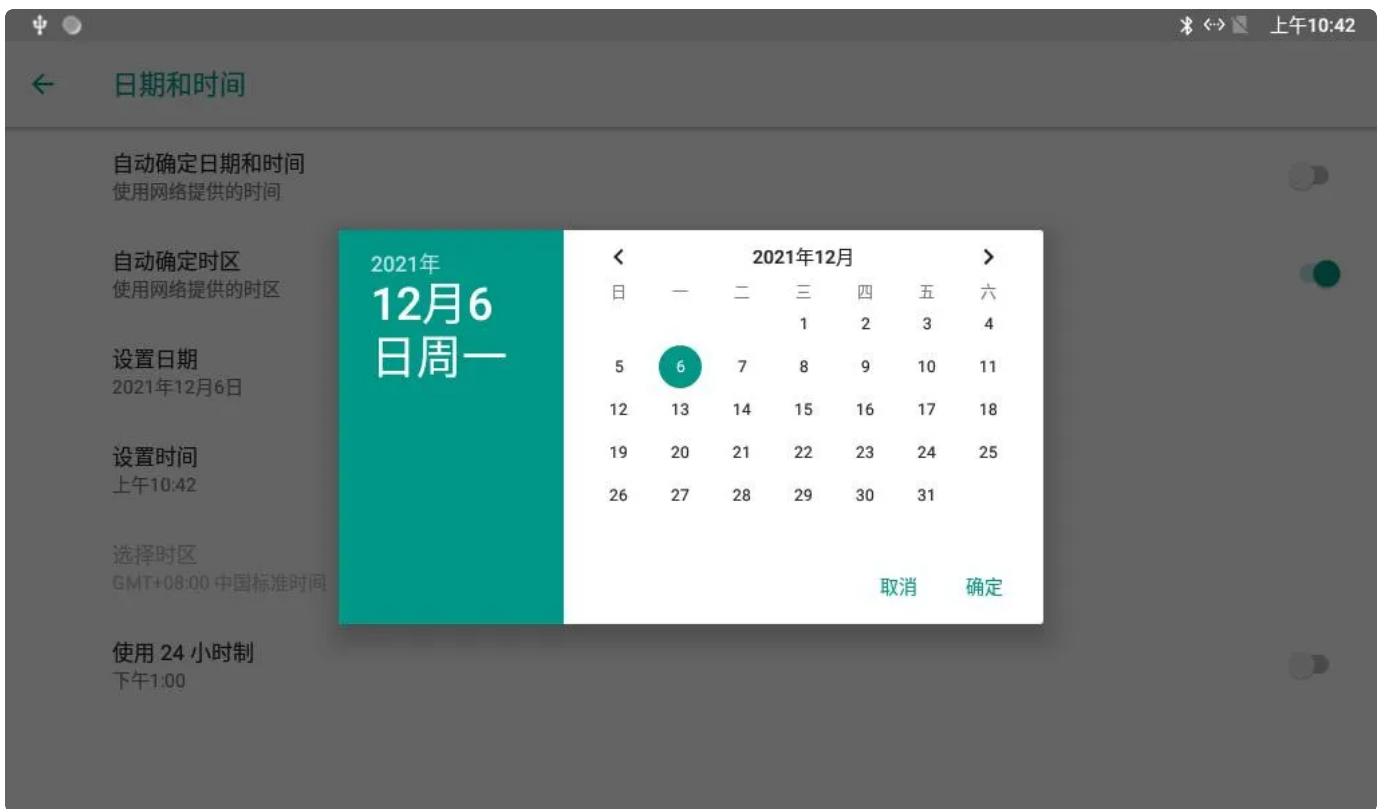
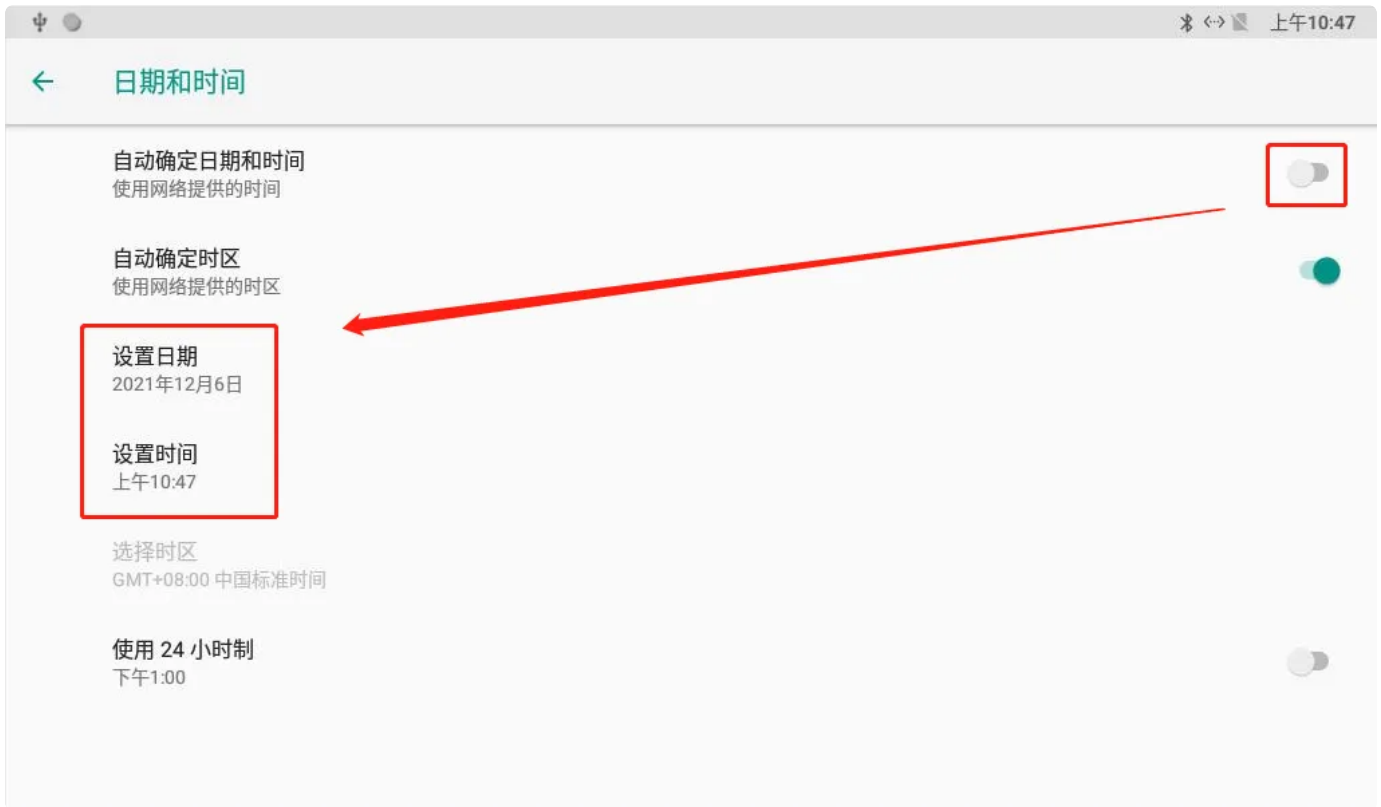


2.14 时间设置

菜单栏界面点击【设置】->【系统】->【日期和时间】
可默认选择【自动确定日期和时间】和【自动确认时区】



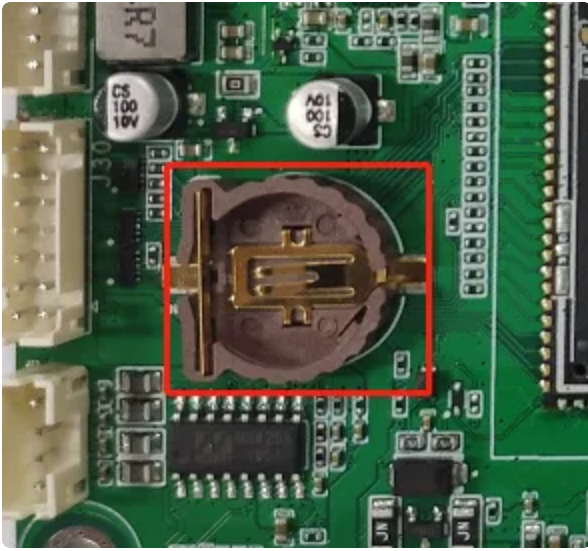
如需自行设置日期和时间只需取消选择【自动确定日期和时间】即可，再点击【设置日期】和【设置时间】自行设置



【时区】的自行设置也是同等道理

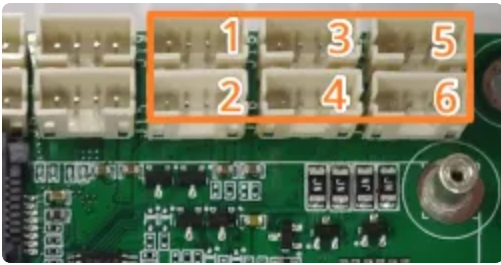
2.15 RTC同步

RTC有独立的时钟芯片，插入rtc电池后，可通过以太网/4G网络自动同步时间



RTC纽扣电池

2.16 串口



串口接口位置及引脚定义如上图所示，设备节点列表如下图所示

序号	接口位置	电平	串口设备节点
1	J19	RS232	/dev/ttyS0
2	J15	RS232	/dev/ttyS1
3	J18	RS232	/dev/ttyS4
4	J16	RS232	/dev/ttyS5
5	J14	RS485	/dev/ttyS3
6	J13	RS485	/dev/ttyS2 (默认此接口不开启，配置为调试串口)

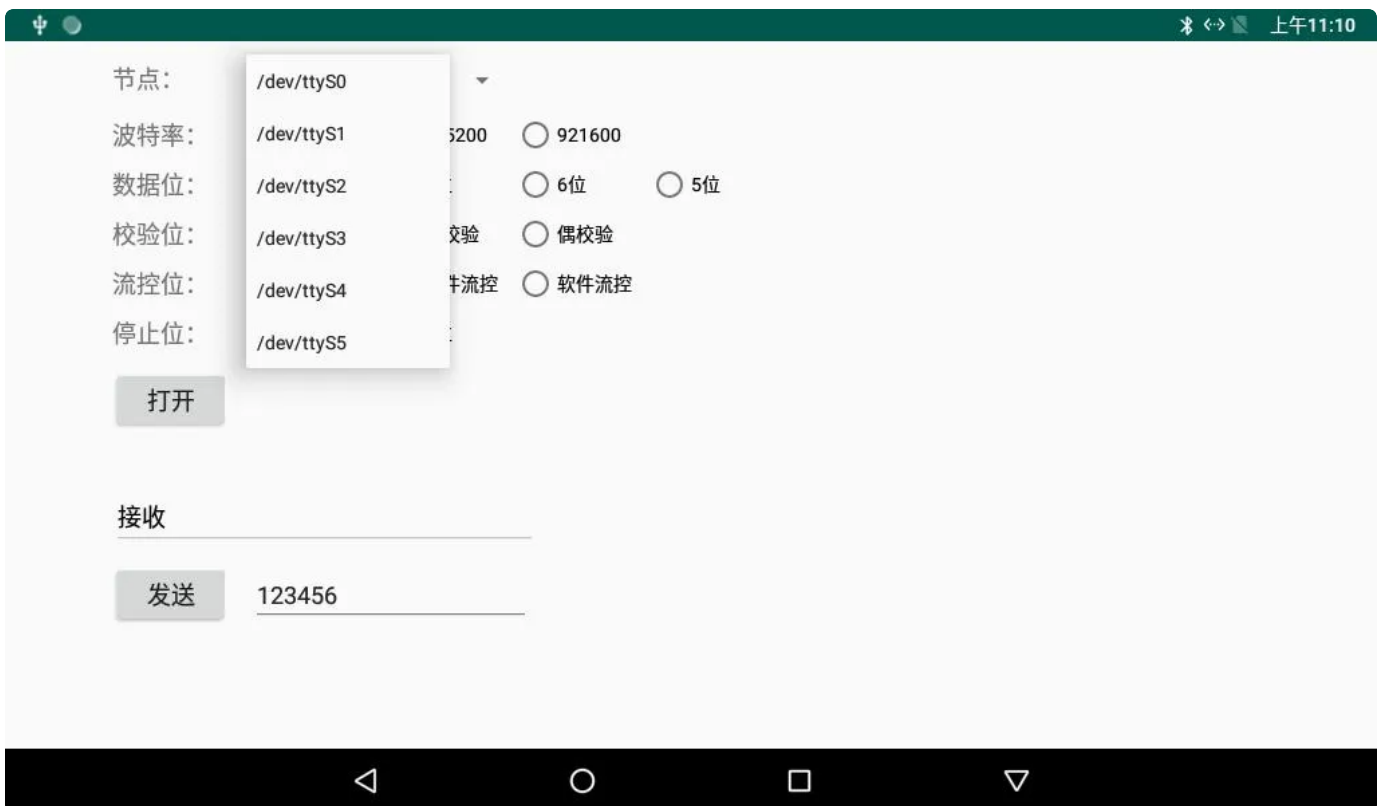
测试需自行安装IDO设置APK测试串口



菜单栏点击【ID0设置】->【串口测试】

2.16.1 RS232

选择对应的串口节点，提示：【打开成功：ttyS*】串口后，短接对应串口tx-rx引脚，点击发送数据123456，串口测试界面即可收到数据123456，如下图所示



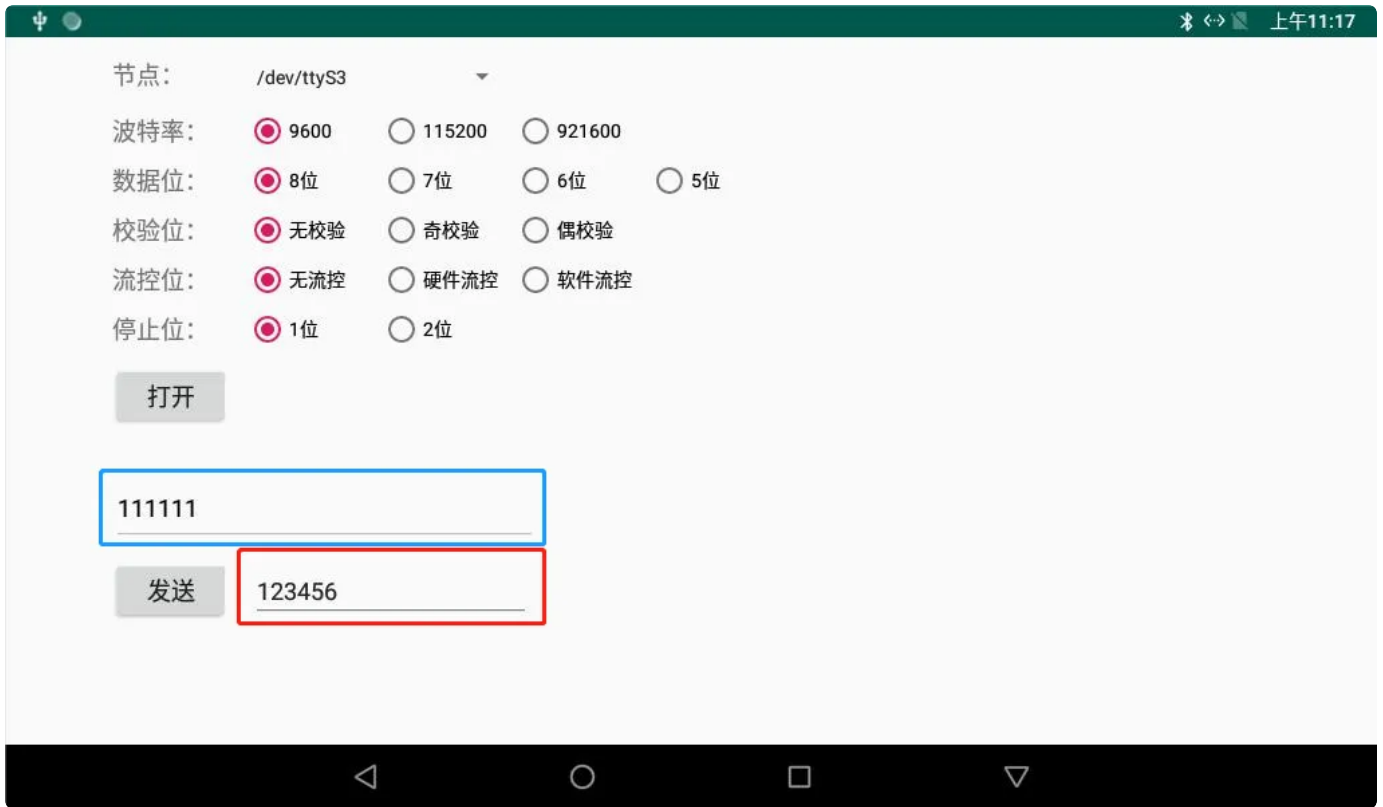


注：TTL串口测试可参考RS232

2.16.2 RS485

选择对应的串口节点，提示：【打开成功：ttyS*】串口后，使用485串口工具连接板子引脚：A-A1 B-B1

开发板串口界面



PC端串口调试助手

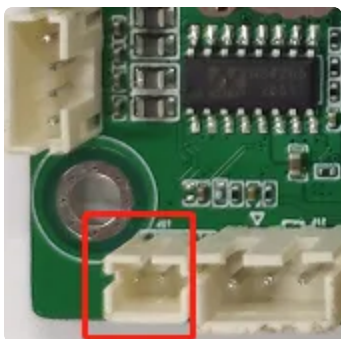


红框：开发板发送到PC串口调试助手的数据

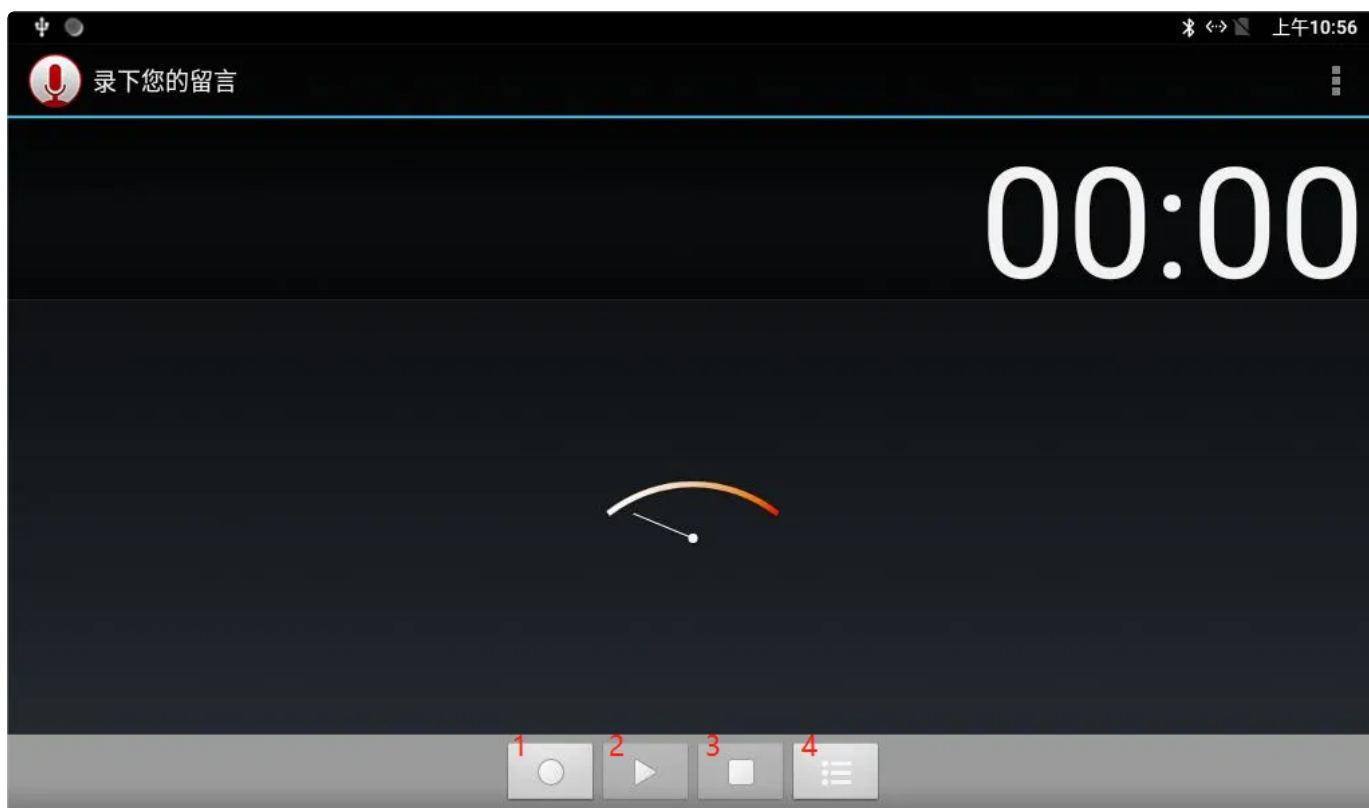
蓝框：PC串口调试助手发送到开发板的数据

2.17 录音

录音前需要接上麦，麦克风接口如下图红框所示

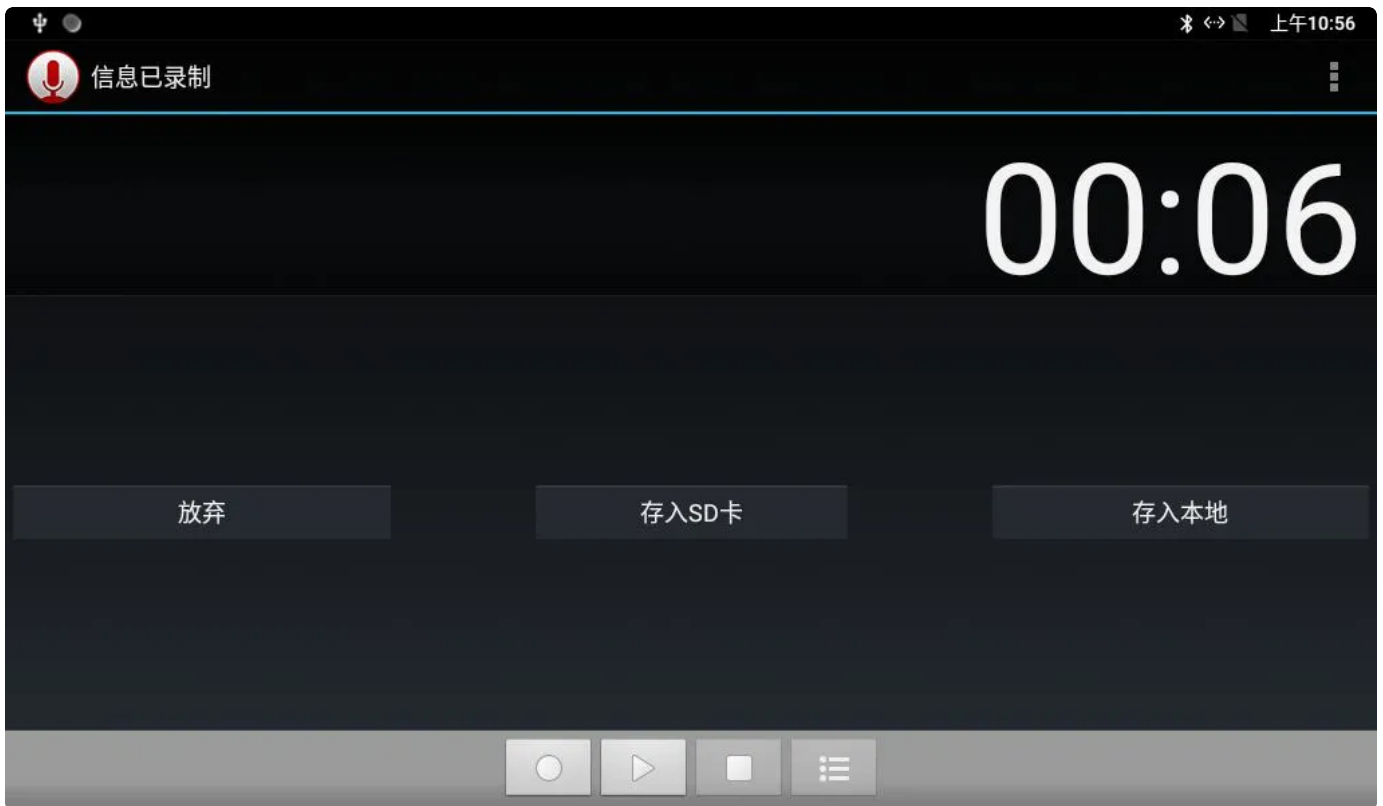


菜单栏界面点击【录音机】

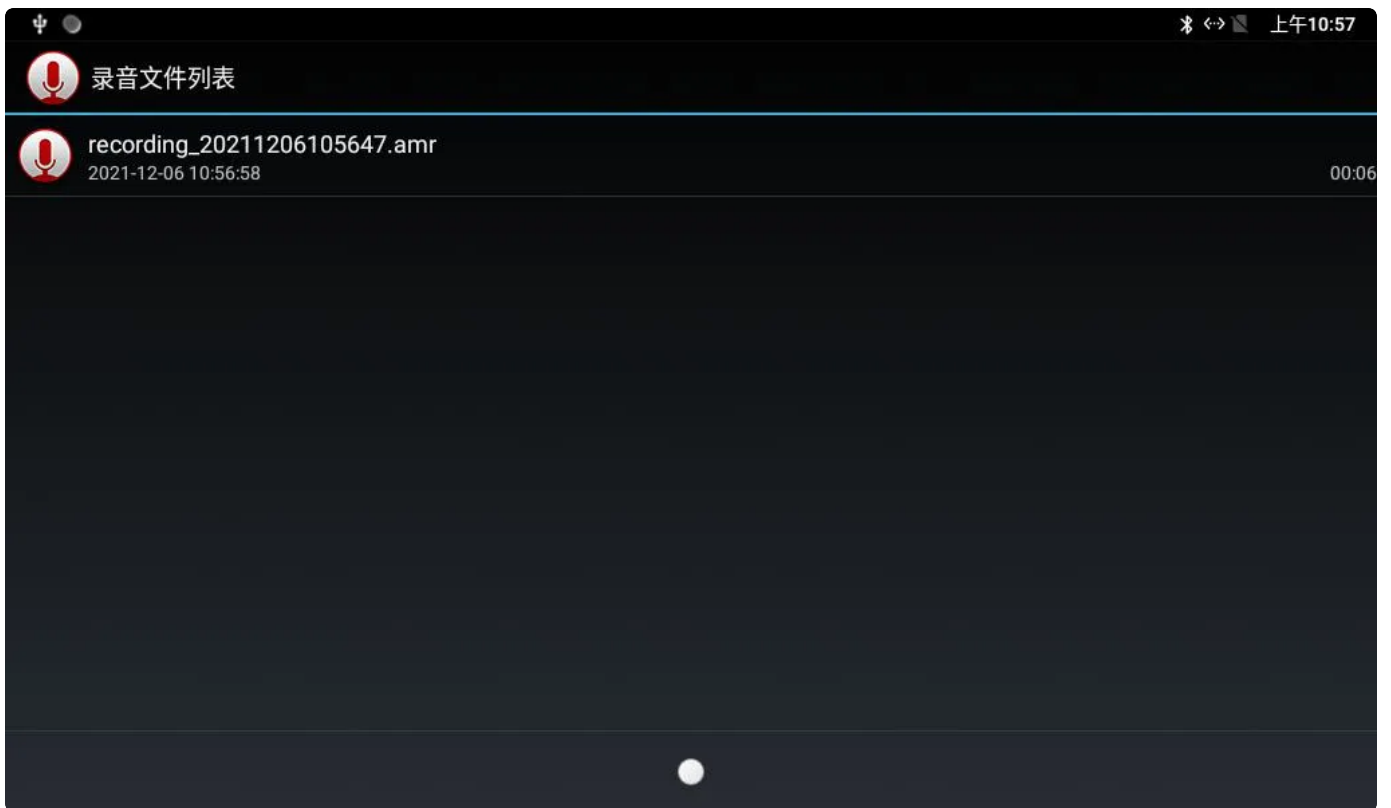


- 1: 开始录音
- 2: 播放录制完成的音频
- 3: 结束录音，完成录制
- 4: 查看录音文件列表

录制完成时



保存后，去查看录音文件列表



如需听到录音声音，需插入耳机或者喇叭即可听到声音，接口详见2.13

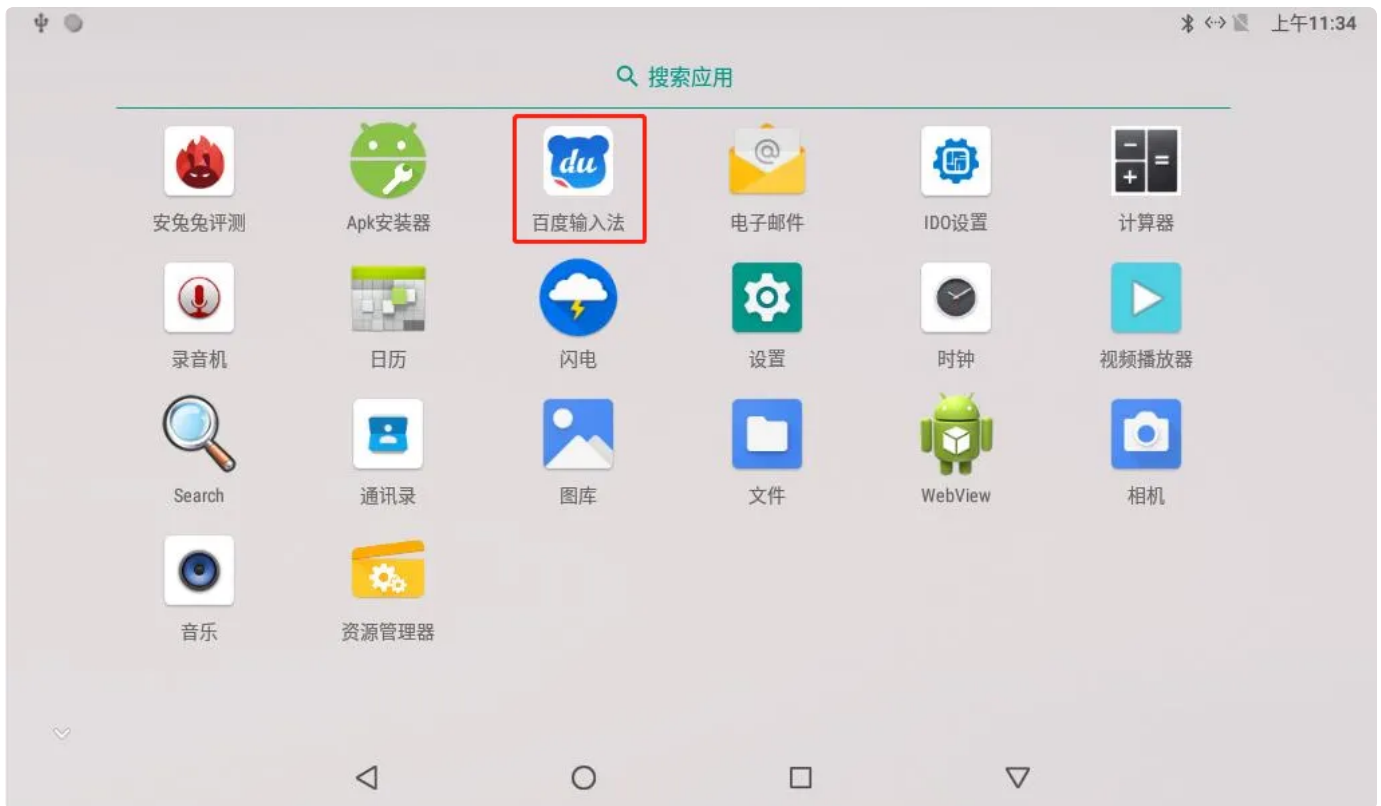
2.18 触摸划线

使用触摸划线同意需要安装IDO设置APK，进入方法如下：
在菜单栏界面->IDO设置->打开触摸测试即可，效果如下



2.19 输入法切换

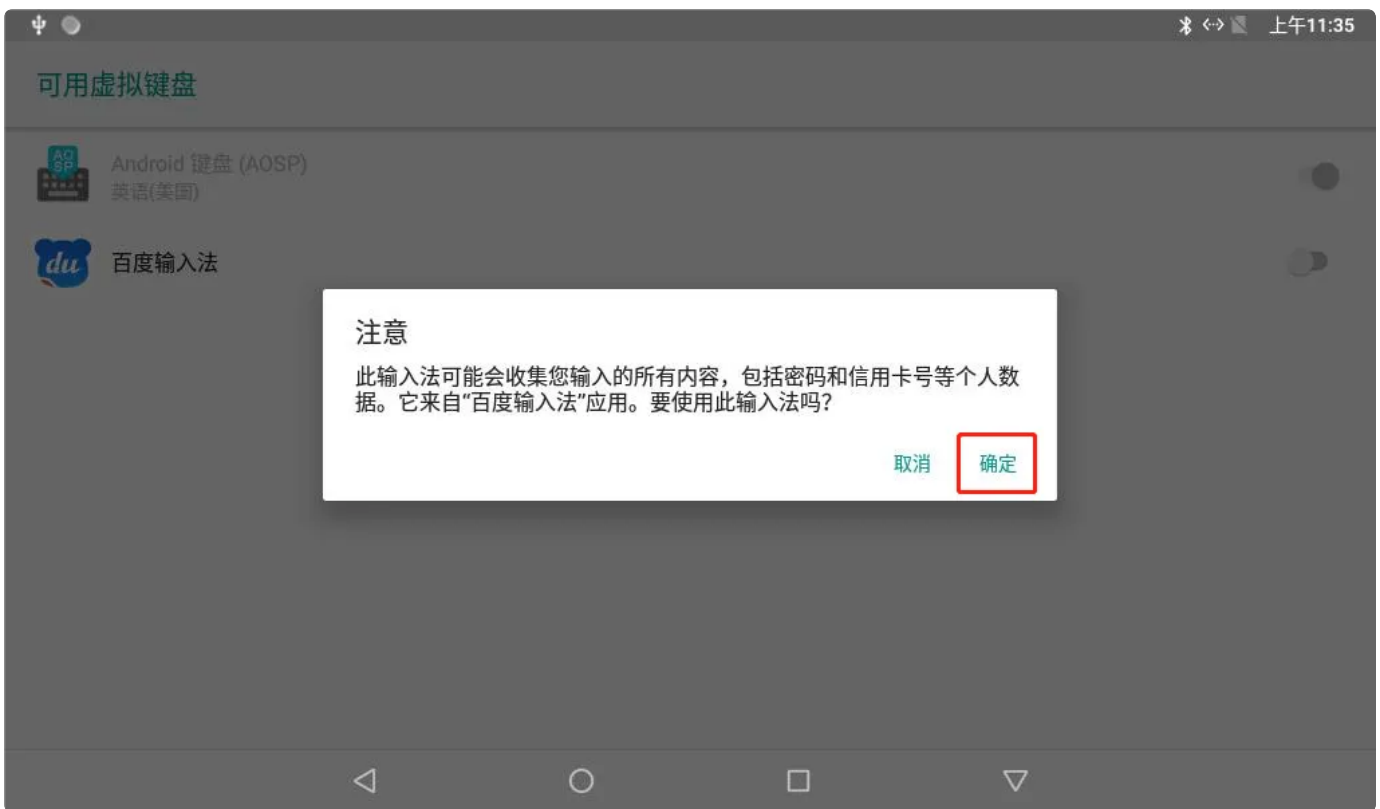
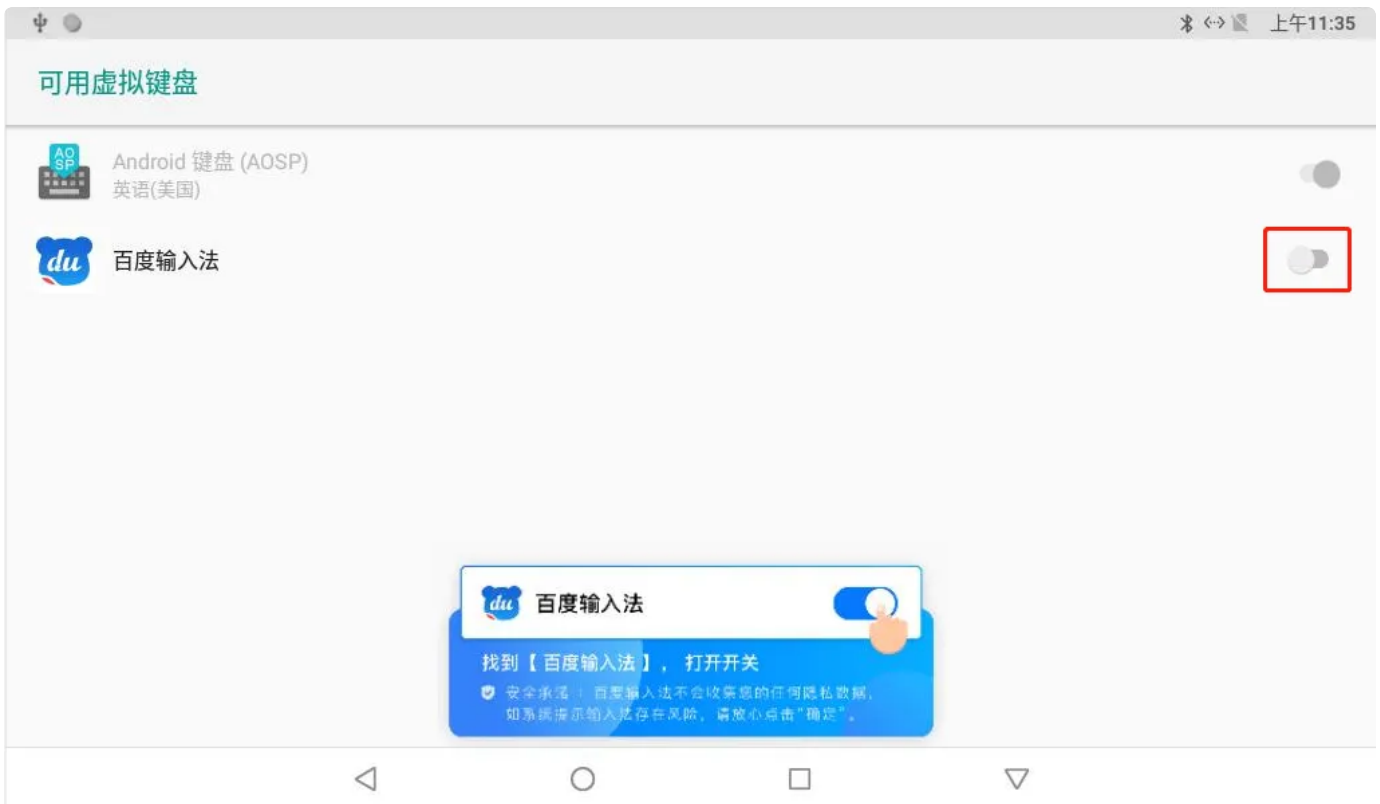
自行下载一个输入法APK文件，下面以百度输入法为例，安装完百度输入法后，启用百度输入法，在桌面找到并点击百度输入法程序

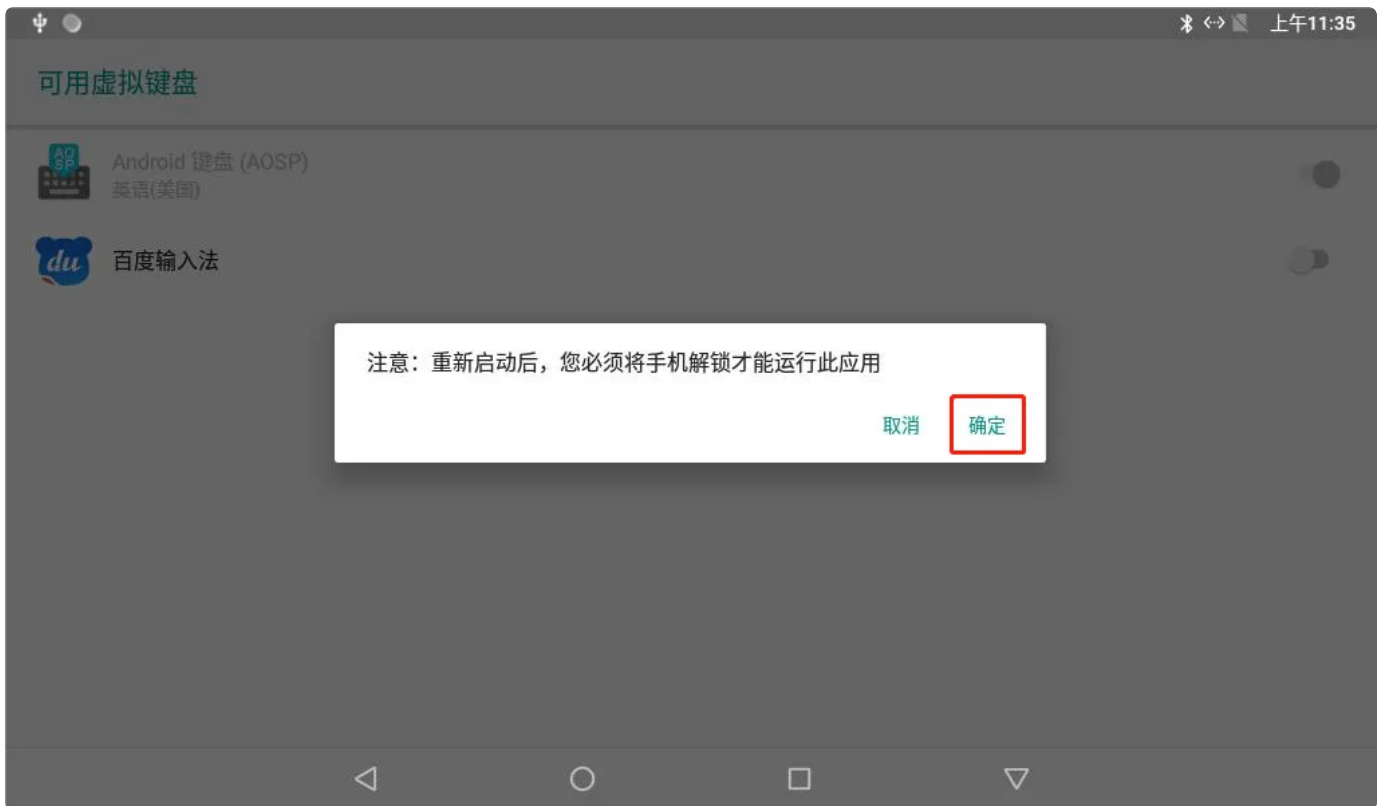


点击【步骤一 勾选百度输入法】



打开百度输入法开关

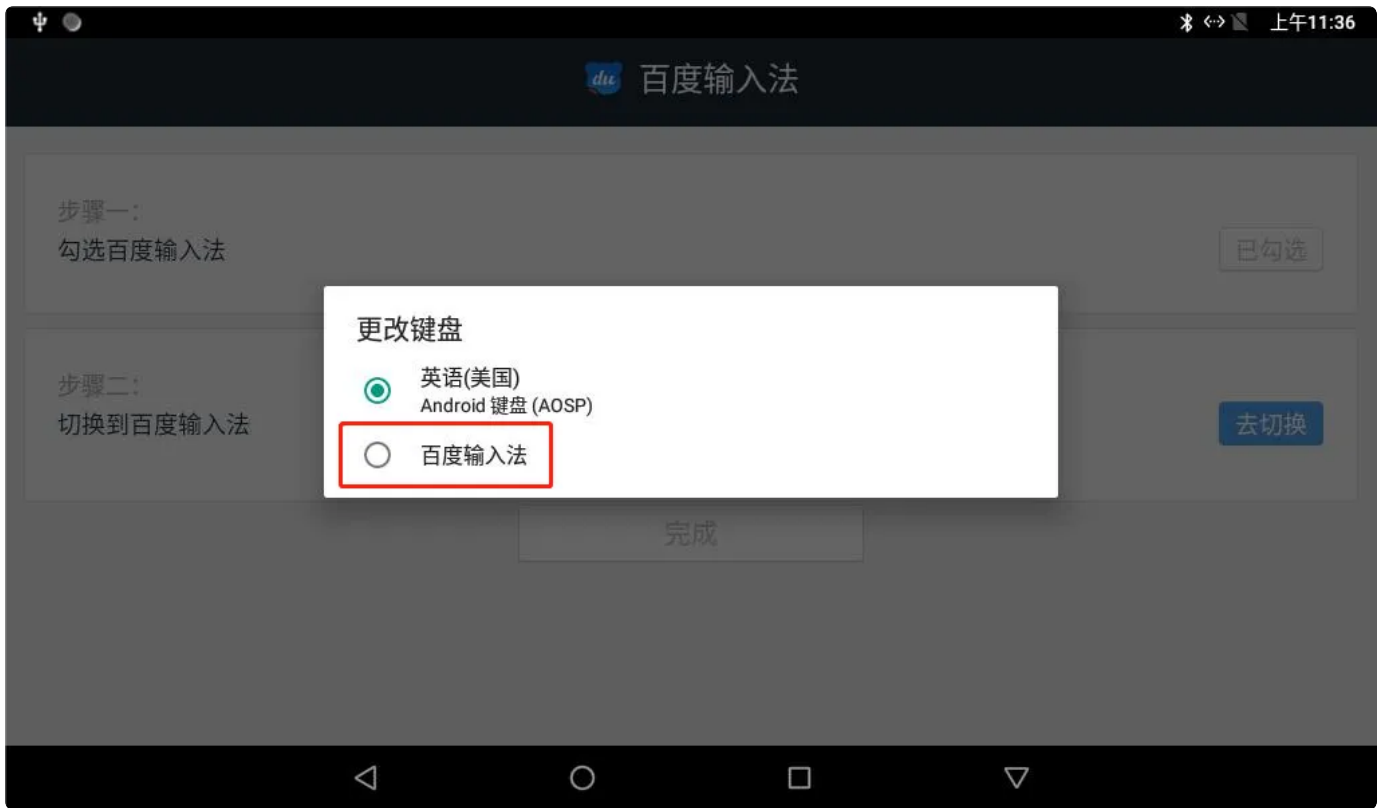




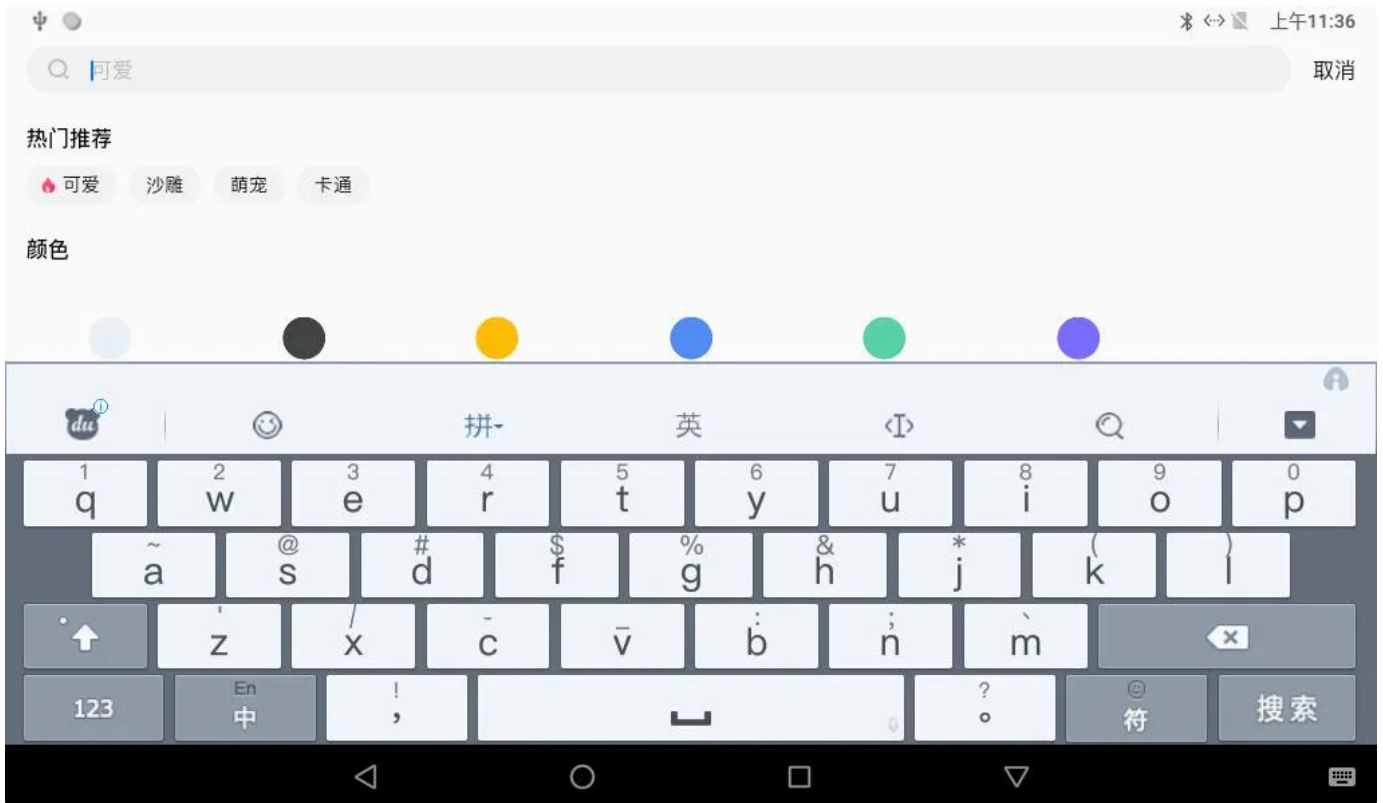
点击【步骤二 切换到百度输入法】



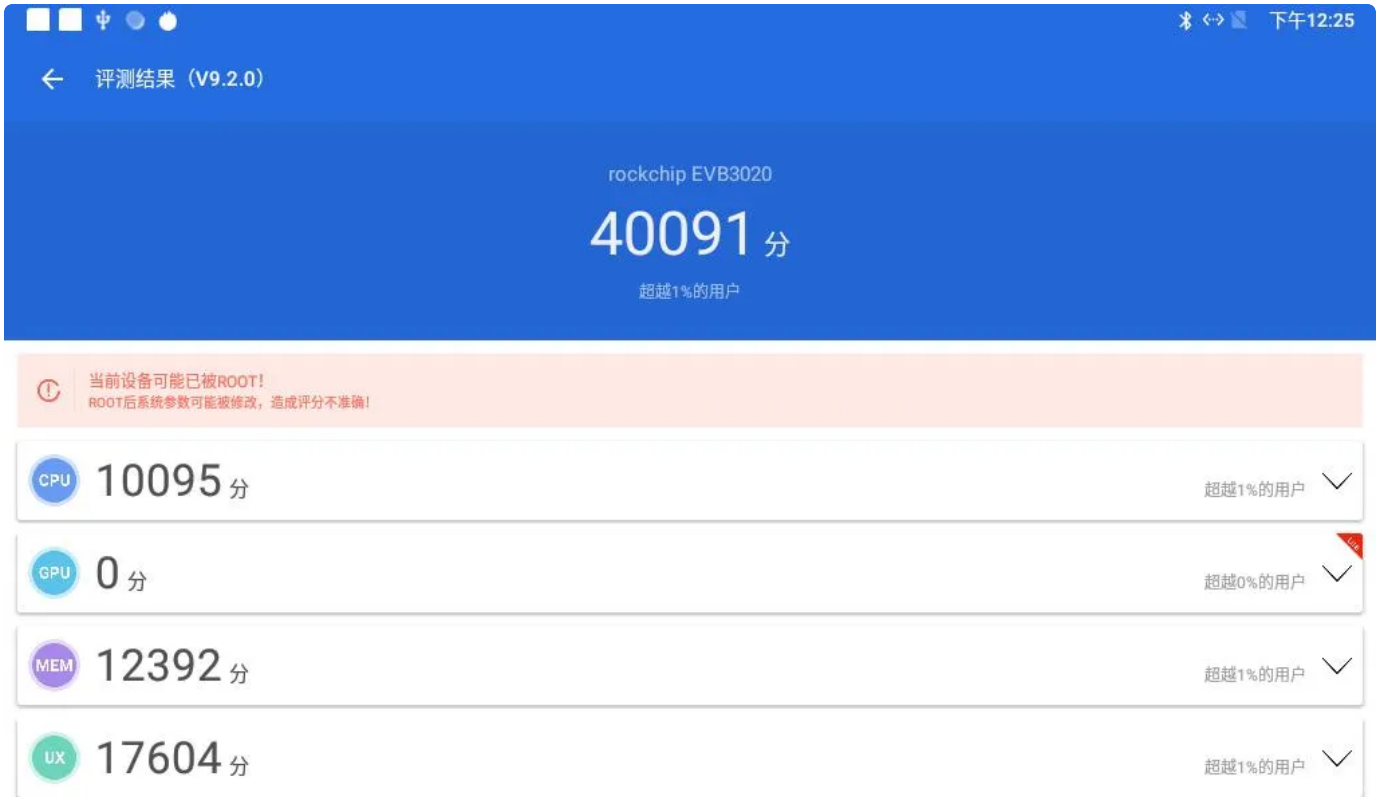
选择【百度输入法】



设置成功



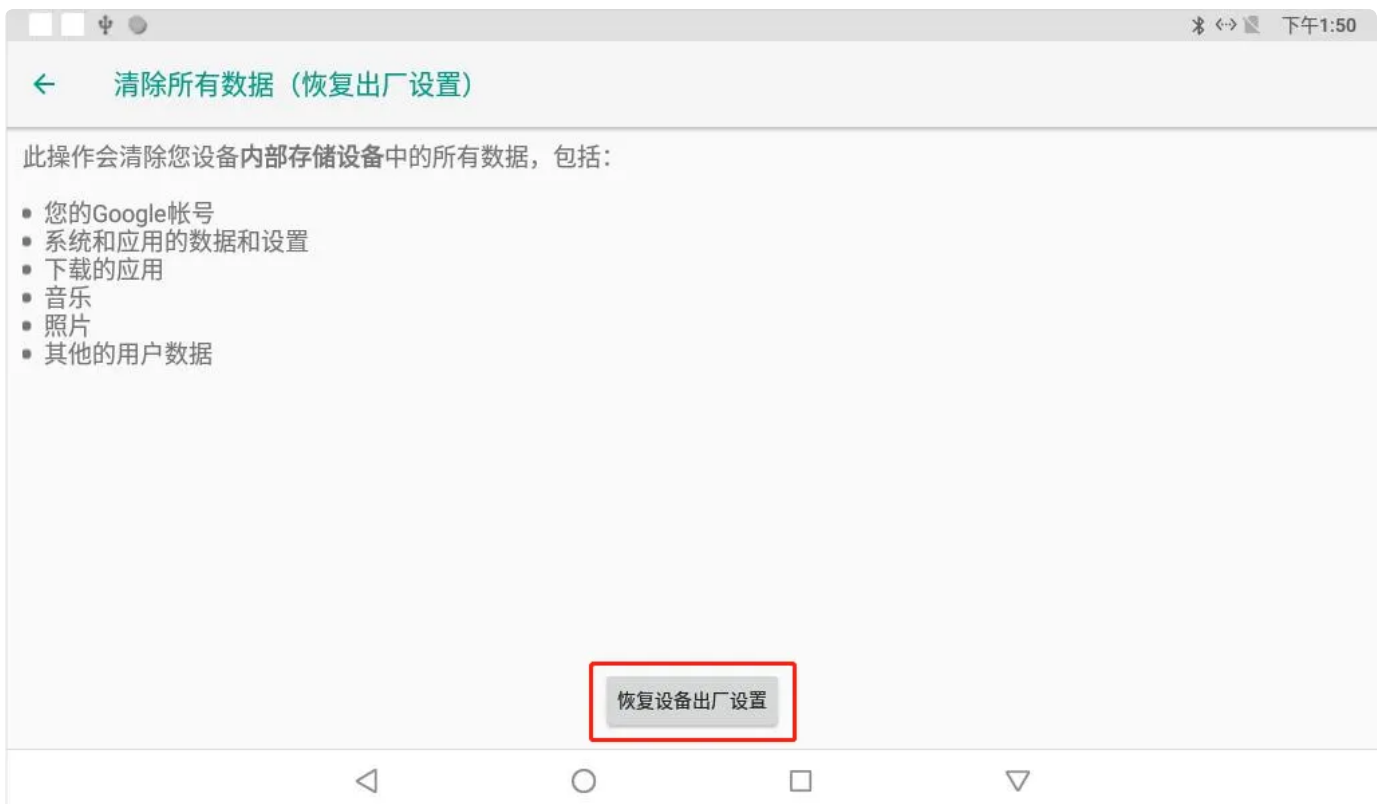
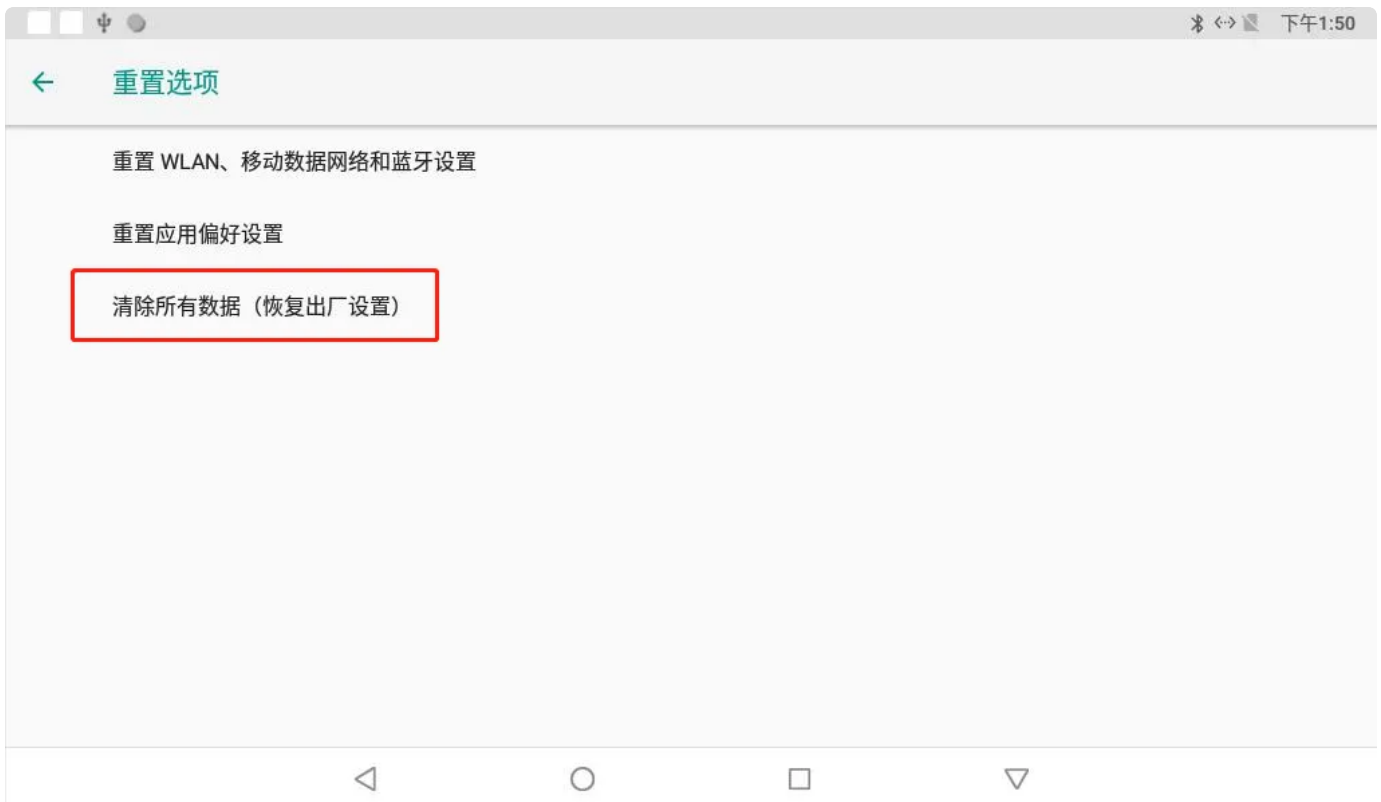
2.20 安兔兔跑分



2.21 恢复出厂设置

设备在使用较长的时间后会产生很多所谓的垃圾文件, 会导致设备变卡, 还有当用户将设备设置改变时想还原的时候, 只需要恢复出厂设置能解决这些问题

方法: 菜单栏界面->设置->系统->重置选项, 点击【清除所有数据(恢复出厂设置)】



此外，还能【重置WLAN、移动数据网络和蓝牙设置】和【重置应用偏好设置】

注意：恢复出厂设置会删除data下全部数据

2.22 ZIP包升级

将需要升级的固件文件拷贝到U盘或者TF卡，插到板子上后，上电过一段时间会弹出提示信息，点击【升级】即可升级

注意：升级用到的U盘和TF卡类型必须为FAT32

2.23 LCD显示方向旋转

```
▼ Java |  
1 Intent mIntent = new Intent("android.ido.intent.action.lcdrotation");  
2 mIntent.putExtra("rotation", "90");//0,90,180,270  
3 sendBroadcast(mIntent);
```

2.24 LCD显示密度设置

默认值未160，调用广播设置后，需要重启才会生效。

```
▼ Java |  
1 Intent mIntent = new Intent("android.ido.intent.action.lcddensity");  
2 mIntent.putExtra("density", "160");//160, 240, 260, 320  
3 sendBroadcast(mIntent);
```

2.25 APK静默安装

```
▼ Java |  
1 Intent intent = new Intent("android.intent.action.SILENT_INSTALL_PACKAGE");  
2 intent.putExtra("apkFilePath", "/path/apkName");//安装apk 绝对路径  
3 intent.putExtra("autostart", true);//true:安装完成后自动运行  
4 sendBroadcast(intent);
```

2.26 APP最前端运行检测

此功能用于设置循环检测APP是否在界面的最前端运行。

/注：如果开启了此功能，在安装APP 的时需要关闭此app check 功能，防止出现错误，在安装完成后再次开启此功能。

```
1 private void AppTopCheck(String packageName, String className, boolean enable){
2     Intent intent1 = new Intent( "android.intent.action.set.appcheck")
3     ;
4     //开启、关闭app check 功能, 开启后, 一直检测当前最前端显示
5     //是否为设置的app, 此设置断电会保存, 开机自动运行
6     intent1.putExtra("enable", enable);
7
8     //检测最前端运行的APP包名
9     intent1.putExtra("packageName", packageName);
10    //如果检测到最前端运行的APP不为参数里面设置的包名, 则重启APP
11    intent1.putExtra("className", className);
12
13    //此参数为设置多少次未检测到APP 在前端运行则重启app, 默认循环检测APP 的时间为2
    秒,
14    //这里设置为3 次, 如果APP 未在前面运行, 则6 秒后则重启app
15    intent1.putExtra("checkCnt", 3);
16    sendBroadcast(intent1);
17 }
```

调用示例:

```
1 AppTopCheck("com.example.myapplication", "com.example.myapplication.MainActivity", true);
```

2.27 关机/重启


```

1 //confirm:ture-会弹出是否关机的确认窗口, false-无弹框, 直接关机
2 public void systemShutdown(boolean confirm) {
3     Intent intent = new Intent("android.intent.action.set.shutdown");
4     intent.putExtra("confirm", confirm);
5     sendBroadcast(intent);
6 }
7
8 //confirm:ture-会弹出是否关机的确认窗口, false-无弹框, 直接重启
9 public void systemReboot(boolean confirm) {
10    Intent intent = new Intent("android.intent.action.set.reboot");
11    intent.putExtra("confirm", confirm);
12    sendBroadcast(intent);
13 }

```

2.28 设置系统时间

```

1 int[] time = {2020,6,12,12,20,0}; //时间{年,月,日,时,分,秒}
2 Intent mIntent1 = new Intent("android.intent.action.settime");
3 mIntent1.putExtra("time",time);
4 sendBroadcast(mIntent1);

```

2.29 设置默认Launcher

系统默认的 Launcher

包名: com.android.launcher3

类名: com.android.launcher3.Launcher

代码中调用广播设置包名和类名

```

1 Intent mIntent1 = new Intent("android.intent.action.launcher");
2 mIntent1.putExtra("packageName", pakName); //pakName:Launcher 包名字符串
3 mIntent1.putExtra("className", clsName); // // clsName:Launcher 类名字符串
4 sendBroadcast(mIntent1);

```

AndroidManifest.xml 添加如下配置

```

1 <action android:name="android.intent.action.MAIN" />
2 <category android:name="android.intent.category.HOME" />
3 <category android:name="android.intent.category.DEFAULT" />
4 <category android:name="android.intent.category.LAUNCHER" />

```

2.30 修改系统开机动画

bootanimation.zip制作方法参考：<https://mp.weixin.qq.com/s/b-5gVQEvHX1SGWeGPPHsgw>

设置开机动画

将制作好的 bootanimation.zip 放到 /sdcard/bootanimation.zip，系统将在2秒内自动执行开机动画替换，并在2秒后删除 /sdcard/bootanimation.zip 文件，当重启系统，即可看到开机动画已变更。

取消添加的开机动画

在/sdcard/目录放一个 delbootanimation 文件（文件内容可以为空），系统将在2秒识别，并删除/sdcard/delbootanimation文件，执行完成后，开机动画将会恢复为默认。

2.31 网络ADB

可通过广播的方式开启和关闭网络ADB功能

```

1 Intent mIntent = new Intent("android.ido.intent.action.netadb");
2 mIntent.putExtra("enable", true); //true: 开启网络adb调试, false: 禁止网络adb
   调试
3 sendBroadcast(mIntent);

```

2.32 设备背光亮度

```

1 Intent mIntent = new Intent("android.ido.action.screen.brightness");
2 mIntent.putExtra("value", 200); //背光值 0-255
3 sendBroadcast(mIntent);

```