

IDO-SBC3968-V1D Android11系统使用手册

1、硬件资源概况

1.1 主板照片

1.2 硬件资源及设备节点

2、功能测试及接口使用方法

2.1 Ethernet

2.2 WIFI

2.3 Bluetooth

2.4 4G

2.5 串口

2.6 喇叭/麦克风

2.7 耳机

2.8 RTC

2.9 USB

2.10 TF Card

2.11 MIPI CSI Camera

2.14 网络ADB

2.14.1 通过命令行打开网络ADB

2.14.2 网络ADB使用

2.15 屏幕背光

2.16 zip包升级

U盘/TF卡升级

命令升级

3、自定义API功能使用

3.1 关机和重启

3.1.1 关机

3.1.2 重启

3.2 定时开关机

3.3 以太网

3.3.1 以太网口设置静态IP地址

3.3.1 以太网口设置动态获取IP地址

3.4 WIFI

3.5 飞行模式

3.6 APP前台运行守护

3.7 截屏

3.8 设置系统时间

3.9 APK静默安装

3.10 状态栏显示和隐藏

3.10.1 隐藏状态栏

3.10.2 显示状态栏

3.11 导航栏显示和隐藏

3.11.1 隐藏导航栏

3.11.2 显示导航栏

4、APP Launcher配置方法

5、APP签名



IDO-SBC3968-V1D

Android11系统使用手册

深圳触觉智能科技有限公司

www.industio.cn

文档修订历史

版本	修订内容	修订	审核	日期
V1.0	创建文档	Huangjintao		2023/08/02

1、硬件资源概况

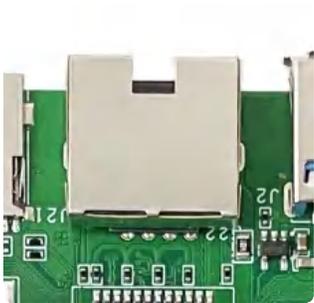
1.1 主板照片

序号	名称	描述	设备节点
1	内核版本	Linux 4.19.193	
2	系统版本	Android11	
3	内存	LPDDR4 (2G/4G选配)	
4	存储	eMMC5.1 (16GB / 32GB / 64GB / 128GB选配)	
5	供电	DC接口12V@2A	
6	显示	HDM LVDS eDP MIPI	
7	USB OTG	USB OTG Type-C	
8	USB HOST	USB3.0 HOST(Type-A) X 1 USB2.0 HOST(Type-A) X 2 USB2.0 HOST(PH2.0) X 7	
9	TF Card	TF Card x 1	
10	以太网	千兆以太网 x 2	eth0、eth1
11	WIFI/BT	AM-NM371SM 2.4G	wlan0 、hci0
12	扬声器		
13	耳机	3.5mm 美标	
14	Camera	OV5648、OV8858	
15	串口	UART x 2 RS485 x 1 RS232 x 2	
16	调试串口	TTL x 1	
17	RTC	HYM8563 x 1	

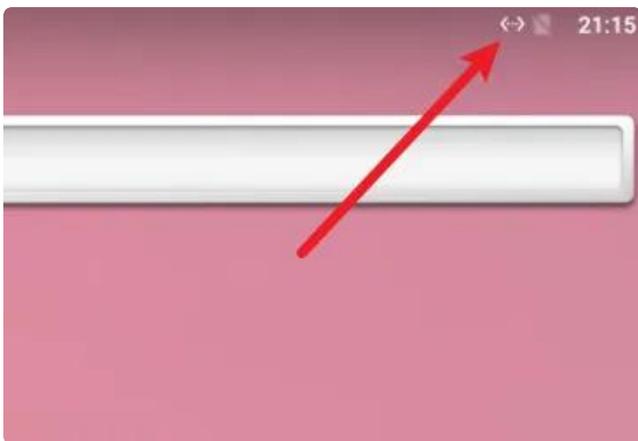
18	系统指示灯	x1	
19	ADC按键	1路	
20	4G	1路支持USB2.0 和USB2.0 MIPI PCIE 接口4G模块	
21	POWER ON	x1	

2、功能测试及接口使用方法

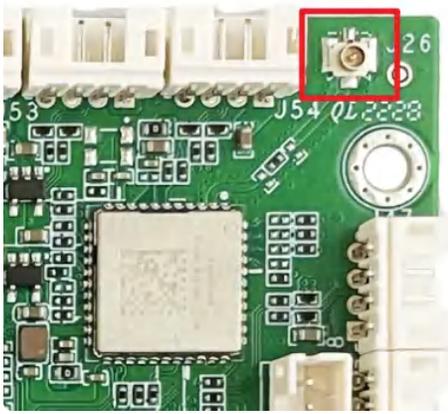
2.1 Ethernet



主板有1路千兆以太网接口，设备节点及位置如上图所示，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主板动态分配 IP 地址。如下图所示即为成功分配到ip



2.2 WIFI



使用WIFI/蓝牙时，需要连接天线以获得良好的信号，上图为WIFI/蓝牙天线接口
菜单栏界面点击【设置】->【WLAN】



选择需连接的WIFI名称，输入对应密码即可连接成功

2.3 Bluetooth

菜单栏界面点击【设置】->【蓝牙】

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



配对成功后主板即可通过蓝牙与手机相互传输文件

2.4 4G

序号	模块名称	说明
1	EC20	4G LTE

测试需要插入SIM卡、模组以及连接好天线



4G模组

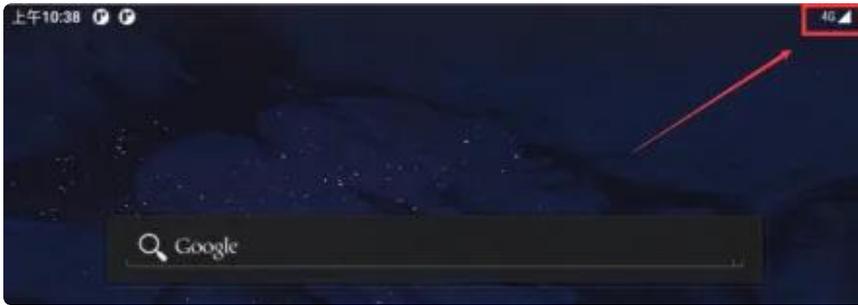


SIM卡接口

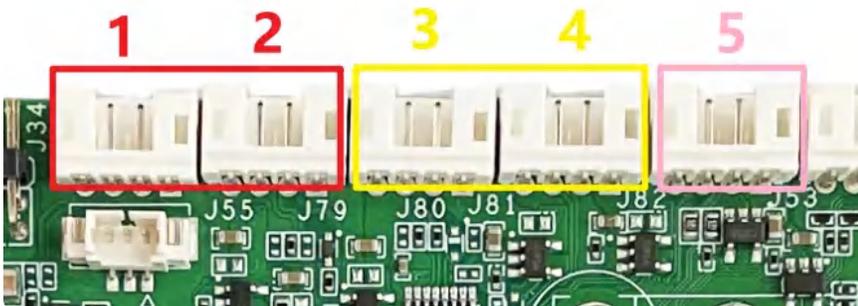
使用标准的尺寸SIM卡，如下图所示



桌面/菜单栏界面显示信号图标后，在菜单栏点击【闪电】在搜索栏输入一个网址即可测试4G网络



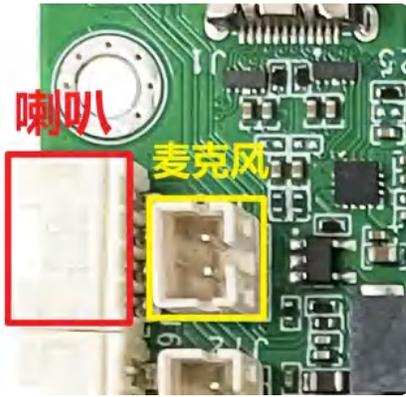
2.5 串口



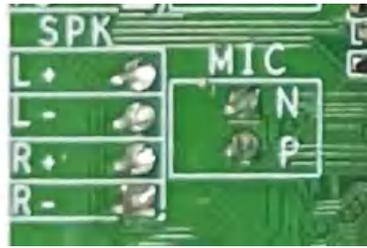
串口接口位置及引脚定义如上图所示，设备节点列表如下：

序号	电平类型	设备节点	驱动原生节点
1	TTL	/dev/ttyS5	/dev/ttyXRUSB0
2	TTL	/dev/ttyS6	/dev/ttyXRUSB1
3	RS232	/dev/ttyS7	/dev/ttyXRUSB2
4	RS232	/dev/ttyS4	
5	RS485	/dev/ttyS8	/dev/ttyXRUSB3

2.6 喇叭/麦克风



正面



背面

喇叭：每个声道支持4ohm 3W输出

麦克风：支持驻极体麦克风输入

连接喇叭后，菜单栏界面打开【设置】->【声音】即可测试喇叭

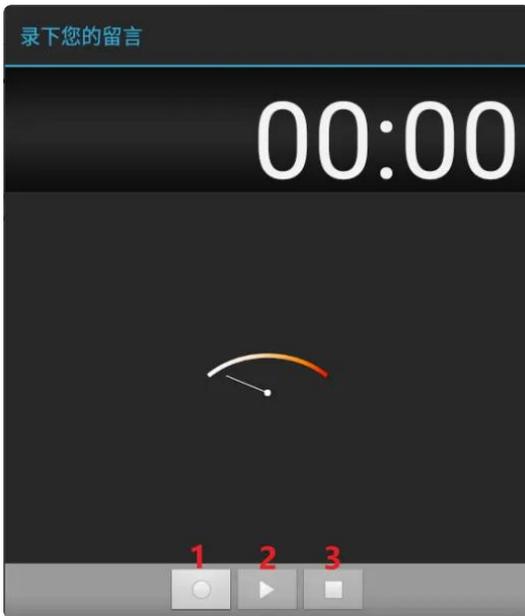


2.7 耳机

支持一路OTMP标准四节耳机座



连接mic后，菜单栏界面点击【录音机】进行录音功能



序号1: 按下录音

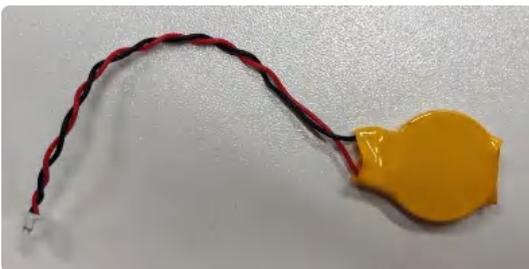
序号2: 按下播放录音

序号3: 按下暂停录音

2.8 RTC



外部RTC HYM8563 电池座位于J37，规格为 MX1.25-2P 立式，可连接3V 纽扣电池，RTC电池参考如下

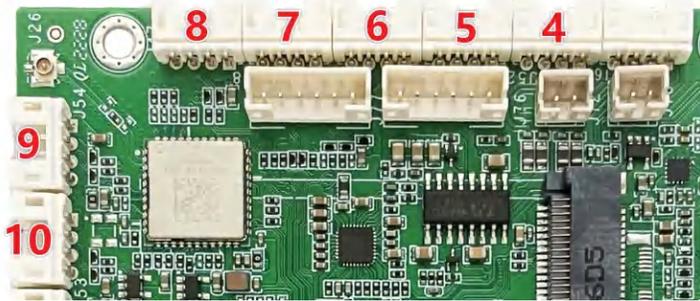
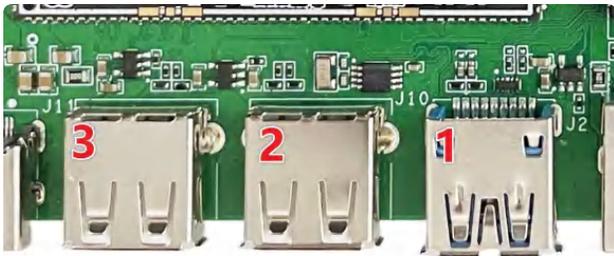


设备节点: /dev/rtc0

系统默认使用HYM8563作为系统时钟, 时间设置方法如下:

```
▼ Plain Text |
1 #设置时间
2 console:/ $ su date "2022-11-26 14: 00"
3
4 #将rtc时钟调整为与目前的系统时钟一致
5 console:/ $ su hwclock -w
6
7 #获取硬件rtc当前时间
8 console:/ $ su hwclock
9 Sat Nov 26 17:27:34 2022 0.000000 seconds
```

2.9 USB



USB接口如上图所示, 功能说明如下

序号	功能	控电节点
1	USB 3.0 HOST	/sys/class/leds/usb3_pwr/brightness
2	USB 2.0 HOST	/sys/class/leds/usb2_pwr/brightness
3	USB 2.0 HOST	/sys/class/leds/usb1_pwr/brightness
4	USB 2.0 HOST	/sys/class/leds/usb4_pwr/brightness

5	USB 2.0 HOST	/sys/class/leds/usb6_pwr/brightness
6	USB 2.0 HOST	/sys/class/leds/usb5_pwr/brightness
7	USB 2.0 HOST	/sys/class/leds/usb7_pwr/brightness
8	USB 2.0 HOST	/sys/class/leds/usb8_pwr/brightness
9	USB 2.0 HOST	/sys/class/leds/usb9_pwr/brightness
10	USB 2.0 HOST	/sys/class/leds/usb10_pwr/brightness

供电控制说明，序号1写"1"关闭电源，写"0"开启电源；序号2-序号6设备节点写"0"关闭电源，写"1"开启电源

命令行控制方法如下，以序号1为例

```

1 #关闭
2 echo 0 > /sys/class/leds/usb3_pwr/brightness
3 #开启 (默认状态)
4 echo 1 > /sys/class/leds/usb3_pwr/brightness

```

USB OTG 切换命令

USB OTG 支持host 和device 模式的切换，软件切换方法如下

```

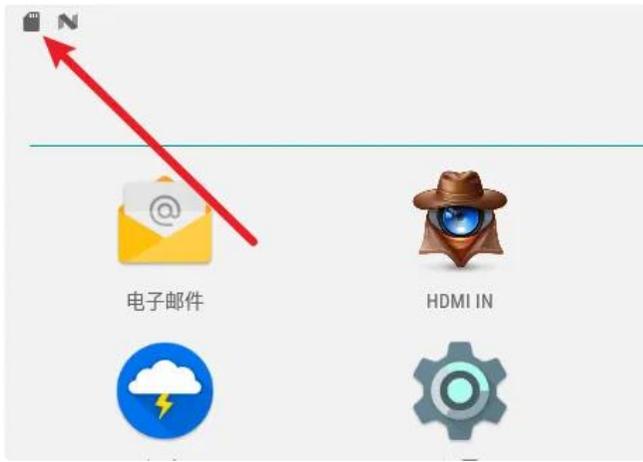
1 ## host
2 echo host > /sys/devices/platform/usb@fe800000/dwc3_mode
3 ## device
4 echo peripheral > /sys/devices/platform/usb@fe800000/dwc3_mode

```

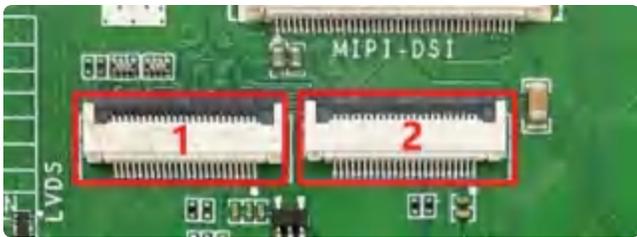
2.10 TF Card



TF Card支持FAT32和NTFS格式分区自动挂载。插入TF卡后，桌面/菜单栏界面界面会显示TF卡标识



2.11 MIPI CSI Camera



MIPI CSI 接口如上图所示，支持OV5648 和OV8858 摄像头模组



OV8858



OV5648

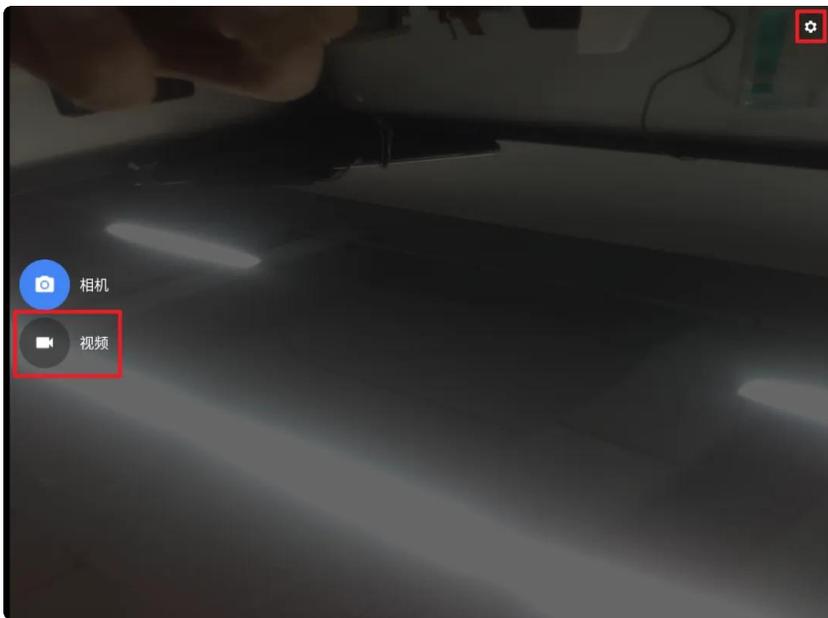
连接方法如下



菜单栏界面点击  相机 软件后，点击右边相机图标即可拍照



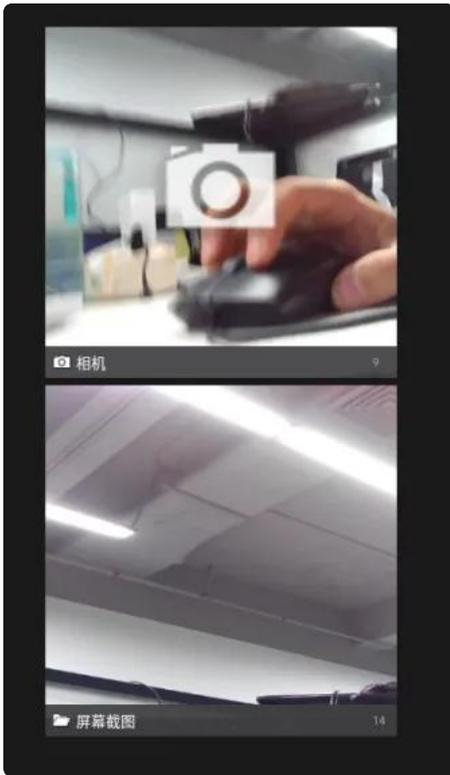
相机拍照界面向右划可切换录视频模式，点击右上角设置图标即可设置分辨率及画质等



设置界面



拍好的照片及视频可在菜单栏界面点击【图库】即可找到



2.14 网络ADB

使用网络ADB的条件：

1. 主板上首先要有网口，或者能通过 WiFi 连接网络
2. 需要主板和研发机（PC 机）已经接入局域网，并且设备设有局域网的 IP 地址。
3. 确保研发机和主板能够相互 ping 得通
4. 主板已经安装了ADB

2.14.1 通过命令行打开网络ADB

可通过调试串口或者adb进入主板系统，并执行以下命令即可开启网络adb

```
1 # setprop persist.internet_adb_enable 1
```

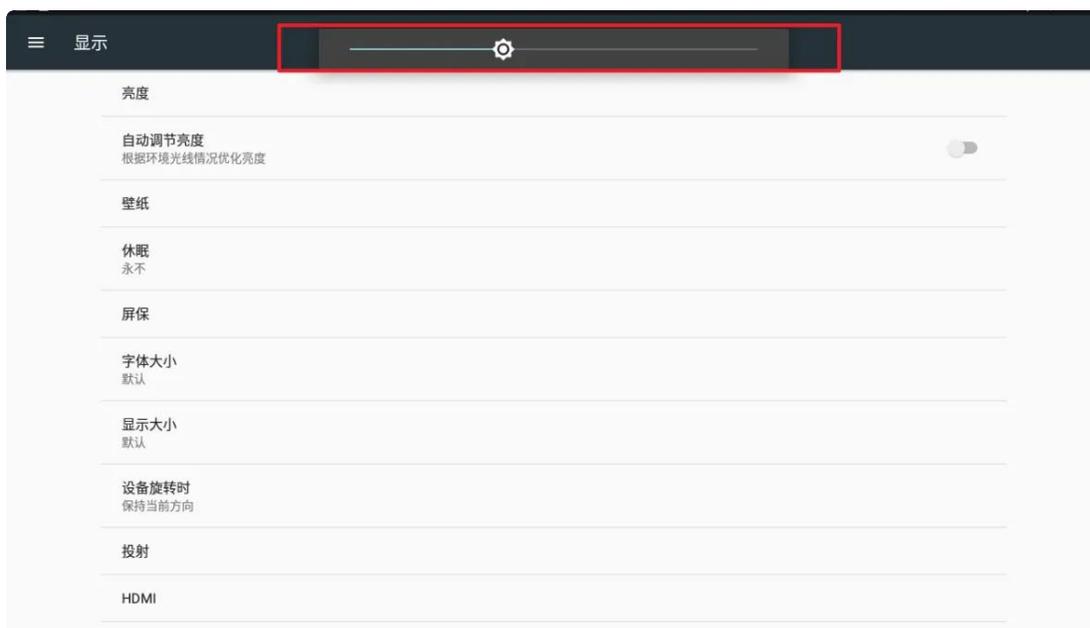
2.14.2 网络ADB使用

假设主板IP为：192.168.0.7，在终端执行以下命令

```
1 # 连接到设备
2 C:\Users\aston> adb connect 192.168.0.7:5555
3 connected to 192.168.0.7:5555
4
5 #查看是否连接成功
6 C:\Users\aston> adb devices
7 List of devices attached
8 192.168.0.7:5555      device (识别出设备主板ip表示连接成功)
9
10 #执行命令进入到主板系统
11 C:\Users\aston> adb shell
12 rk3399_Android11:/ $
13 rk3399_Android11:/ $ ifconfig eth0
14 eth0      Link encap:Ethernet  HWaddr 42:97:62:34:b7:4d  Driver rk_gmac-dw
mac
15          inet addr:192.168.0.7  Bcast:192.168.0.255  Mask:255.255.255.0
16          inet6 addr: fe80::295e:e9d0:9c00:d28e/64 Scope: Link
17          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
18          RX packets:387 errors:0 dropped:0 overruns:0 frame:0
19          TX packets:129 errors:0 dropped:0 overruns:0 carrier:0
20          collisions:0 txqueuelen:1000
21          RX bytes:45303 TX bytes:14353
22          Interrupt:40
```

2.15 屏幕背光

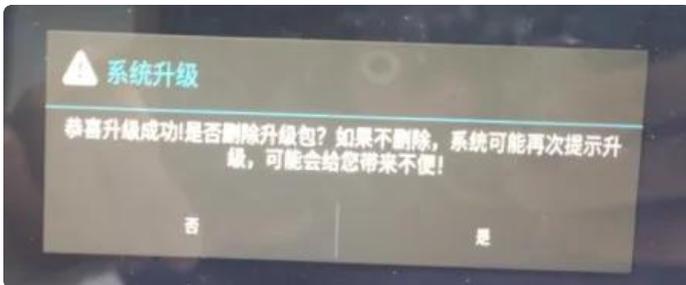
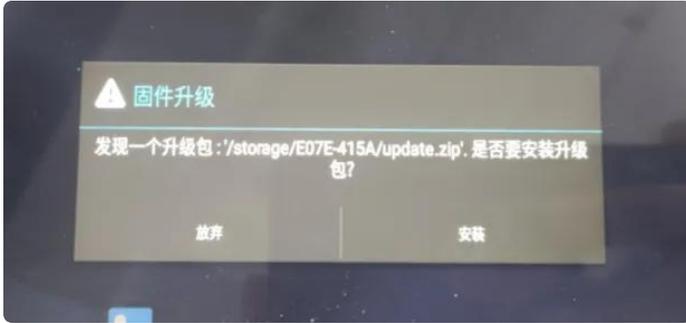
菜单栏界面打开【设置】->【显示】->【亮度】，即可调节屏幕背光



2.16 zip包升级

U盘/TF卡升级

将需要升级的update.zip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【是否需要安装升级】框，点击【安装】即可开始升级系统，升级结束会重新启动系统，如下图所示：



注意：升级用到的U盘和TF卡类型必须为FAT32

命令升级

如果无法通过U盘升级，可通过adb命令push升级包进系统/sdcard目录下升级系统，命令如下：

```
Shell |  
1  adb root  
2  adb remount  
3  adb push your/update.zip/path /sdcard
```

push成功后执行adb reboot 重启系统

3、自定义API功能使用

3.1 关机和重启

通过广播的方式设置系统关机和重启

3.1.1 关机

功能说明：

发送广播的方式关闭设备。

参数说明：

名称	值	说明
confirm	true	发送广播后，会弹出是否关机的确认窗口
	false	发送广播后，无弹窗，直接关机

调用示例：

```
▼ Java |  
1 Intent intent = new Intent("android.intent.action.set.shutdown");  
2 intent.putExtra("confirm", true);  
3 sendBroadcast(intent);
```

3.1.2 重启

功能说明：

发送广播的方式重启设备。

参数说明：

名称	值	说明
confirm	true	发送广播后，会弹出是否重启的确认窗口
	false	发送广播后，无弹窗，直接重启

调用示例：

```

1 Intent intent = new Intent("android.ido.intent.action.set.reboot");
2 intent.putExtra("confirm", true);
3 sendBroadcast(intent);

```

3.2 定时开关机

功能说明：

发送广播，设置定时开关机的时间，设置的时间需要是未来时间。

参数说明：

名称	值
timeon	整数类型时间数组 例：int[] mOnTime = {2022,9,26,9,35};
timeoff	整数类型时间数组 例：int[] mOffTime = {2022,9,26,9,34};

调用示例：

```

1 int[] mOffTime = {2022,9,26,9,34}; //{年,月,日,时,分}
2 int[] mOnTime = {2022,9,26,9,35}; //{年,月,日,时,分}
3 Intent mIntent = new Intent("android.intent.action.set.rtc.poweronoff");
4 mIntent.putExtra("timeon",mOnTime); //开机时间,见上面数组定义
5 mIntent.putExtra("timeoff",mOffTime); //关机时间,见上面数组定义
6 mIntent.putExtra("enable",true); //true开启,false关闭
7 sendBroadcast(mIntent);

```

3.3 以太网

3.3.1 以太网口设置静态IP地址

功能说明：

发送广播配置以太网口为静态IP地址。

参数说明：

名称	值	说明
mode	Static	设置为IP地址为静态模式
iface	eth0	以太网卡名称
ipAddr	IP地址	点分十进制IPV4地址
netmask	掩码	点分十进制IPV4掩码
gateway	网关地址	点分十进制IPV4地址
dns1	DNS	第一个点分十进制DNS地址
dns2	DNS	第二个点分十进制DNS地址，非必须

调用示例：

```
▼ Java |  
1 Intent intent = new Intent("android.ido.intent.action.ethernet");  
2 intent.putExtra("mode", "Static");  
3 intent.putExtra("iface", "eth0");  
4 intent.putExtra("ipAddr", "192.168.0.7");  
5 intent.putExtra("netmask", "255.255.255.0");  
6 intent.putExtra("gateway", "192.168.0.1");  
7 intent.putExtra("dns1", "114.114.114.114");  
8 intent.putExtra("dns2", "8.8.8.8"); //如果只有一个dns，可将dns2设置为和dns1一样  
9 sendBroadcast(intent);
```

3.3.1 以太网口设置动态获取IP地址

功能说明：

发送广播配置以太网口IP地址获取方式为DHCP。

参数说明：

名称	值	说明
mode	DHCP	设置为IP地址为动态获取
iface	eth0	以太网卡名称

调用示例：

```
Java |  
1 Intent intent = new Intent("android.ido.intent.action.ethernet");  
2 intent.putExtra("mode", "DHCP");  
3 intent.putExtra("iface", "eth0");  
4 sendBroadcast(intent);
```

3.4 WIFI

功能说明：

发送广播开启和关闭WIFI。

参数说明：

名称	值	说明
enable	true	开启WIFI
	false	关闭WIFI

调用示例：

```
Java |  
1 Intent intent = new Intent( "android.ido.intent.action.wifionoff");  
2 intent.putExtra("enable", true); //true, 开启; false, 关闭  
3 sendBroadcast(intent);
```

3.5 飞行模式

功能说明：

此功能用于配置系统的飞行模式开启和关闭。

参数说明：

名称	值	说明
enable	true	开启飞行模式
	false	关闭飞行模式

调用示例：

```
Java |  
1 Intent intent = new Intent("android.ido.intent.action.set.airPlaneMode");  
2 intent.putExtra("enable", true);  
3 sendBroadcast(intent);
```

3.6 APP前台运行守护

功能说明：

此功能是设置循环检测APP是否在最前端运行（在界面上显示）。

参数说明：

名称	值	说明
enable	true	开启检测功能
	false	关闭检测功能
packageName	包名	需要保持前台运行的APK的包名
className	类名	需要保持前台运行的APK的类名
checkCnt	检测时间	单位2秒，例：设置为3，则6秒未检测到APP在最前端，则重启拉起

调用示例：

```

1 private void AppTopCheck(String packageName, String className, boolean enable){
2     Intent intent = new Intent( "android.ido.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, 开机会自动运行
6     intent.putExtra("enable", enable);
7
8     //检测最前端运行的APP包名
9     intent.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    intent.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    intent.putExtra("checkCnt", 3);
16    sendBroadcast(intent);
17 }

```

3.7 截屏

功能说明:

此功能用于截取当前屏幕内容。

参数说明:

名称	值	说明
path	保存路径	例: /sdcard/screenshot.png
id	屏幕ID	此参数缺省为0,当有多个屏幕可选择0、1

调用示例:

```

1 Intent intent = new Intent( "android.ido.intent.action.screenshot");
2 intent.putExtra("path", "/sdcard/screenshot.png"); //保存png文件的绝对路径
3 intent.putExtra("id", "0"); //屏幕ID, 此参数缺省为0, 当有多个屏幕可选择0、1、2...
4 sendBroadcast(intent);

```

3.8 设置系统时间

功能说明：

此功能用于设置系统RTC时间。

参数说明：

名称	值	说明
time	整数类型时间数组	整数类型时间数组 {年, 月, 日, 时, 分, 秒} 例: <code>int[] time = {2022,9,26,9,35,0};</code>

调用示例：

```
Java |  
1  int[] time = {2022, 9, 30, 18, 0, 0};  
2  Intent intent = new Intent("android.ido.intent.action.settime");  
3  intent.putExtra("time", time);  
4  sendBroadcast(intent);
```

3.9 APK静默安装

功能说明：

此功能用于静默安装指定路径的apk。

参数说明：

名称	值	说明
apkFilePath	apk文件路径	绝对路径
autostart	true/false	安装完后是否自动运行

调用示例：

```

1 Intent intent = new Intent("android.intent.action.SILENT_INSTALL_PACKAGE");
2 intent.putExtra("apkFilePath", fileName); //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);

```

3.10 状态栏显示和隐藏

功能说明：此功能用于设置状态栏的隐藏和显示。

3.10.1 隐藏状态栏

参数说明：

名称	值	说明
save	true/false	true,断电保存； false, 断电不保存。

调用示例：

```

1 Intent intent = new Intent("android.ido.intent.action.statusbar.HIDE");
2 intent.putExtra("save", true); //true,断电保存； false, 断电不保存
3 sendBroadcast(intent);

```

3.10.2 显示状态栏

参数说明：

名称	值	说明
save	true/false	true,断电保存； false, 断电不保存。

调用示例：

```

1 Intent intent = new Intent("android.ido.intent.action.statusbar.SHOW");
2 intent.putExtra("save", true); //true,断电保存； false, 断电不保存
3 sendBroadcast(intent);

```

3.11 导航栏显示和隐藏

功能说明：此功能用于设置导航栏的隐藏和显示。

3.11.1 隐藏导航栏

参数说明：

名称	值	说明
save	true/false	true,断电保存; false, 断电不保存。

调用示例：

```
▼ Java |  
1 Intent intent = new Intent("android.ido.intent.action.navigation.HIDE");  
2 intent.putExtra("save", true); //true,断电保存; false, 断电不保存  
3 sendBroadcast(intent);
```

3.11.2 显示导航栏

参数说明：

名称	值	说明
save	true/false	true,断电保存; false, 断电不保存。

调用示例：

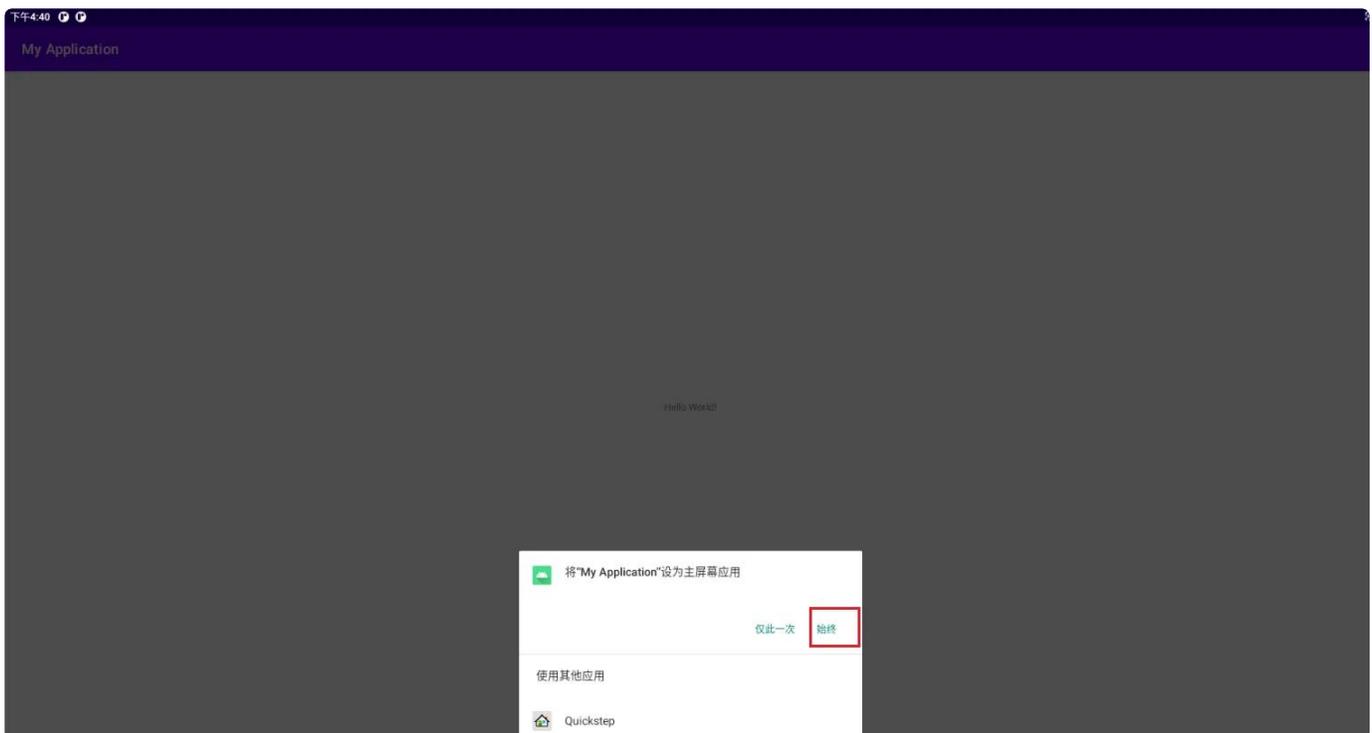
```
▼ Java |  
1 Intent intent = new Intent("android.ido.intent.action.navigation.SHOW");  
2 intent.putExtra("save", true); //true,断电保存; false, 断电不保存  
3 sendBroadcast(intent);
```

4、APP Launcher配置方法

修改APP源码的AndroidManifest.xml，添加以下内容

```
1 <application
2     ...
3     <activity android:name=".MainActivity">
4         <intent-filter>
5             <action android:name="android.intent.action.MAIN" />
6             <category android:name="android.intent.category.HOME" />
7             <category android:name="android.intent.category.DEFAULT" />
8             <category android:name="android.intent.category.LAUNCHER" />
9         </intent-filter>
10    </activity>
11    ...
12 </application>
```

将修改后的apk文件安装至主板中，主板界面上打开安装的应用程序，之后退出到主页时，会弹出提示主屏幕应用选择框，选择安装的应用并设置为“始终”，之后此应用程序将会作为默认的Launcher。



5、APP签名

1. 签名文件获取

链接：<https://pan.baidu.com/s/1D9WqftMfKbUSRnFLRLVmUw?pwd=1234>

提取码：1234

参数	说明
android11.jks	签名文件
123456	密码
ido	签名文件别名

2. Android Studio工程配置签名

a. 修改AndroidManifest.xml，添加android.uid.system 如下

```
HTML |  
1 <manifest xmlns:android="http://schemas.android.com/apk/res/android"  
2     xmlns:tools="http://schemas.android.com/tools"  
3     package="com.example.myapplication"  
4     android:sharedUserId="android.uid.system">
```

b. 在工程根目录新建一个signature文件夹，并将android11.jks 文件放入该文件夹下；

c. 修改工程根目录的 app/build.gradle 文件，添加signingConfigs和配置buildTypes如下：

```
1  android {
2      compileSdk 32
3
4      defaultConfig {
5          applicationId "com.example.myapplication"
6          minSdk 21
7          targetSdk 32
8          versionCode 1
9          versionName "1.0"
10
11         testInstrumentationRunner "androidx.test.runner.AndroidJUnitRu
12         nner"
13     }
14     signingConfigs {
15         release {
16             storeFile file("../signature/android11.jks")
17             storePassword '123456'
18             keyAlias 'ido'
19             keyPassword '123456'
20         }
21
22         debug {
23             storeFile file("../signature/android11.jks")
24             storePassword '123456'
25             keyAlias 'ido'
26             keyPassword '123456'
27         }
28     }
29
30     buildTypes {
31         release {
32             minifyEnabled false
33             proguardFiles getDefaultProguardFile('proguard-android-optimiz
34             e.txt'), 'proguard-rules.pro'
35             signingConfig signingConfigs.release
36         }
37         debug {
38             minifyEnabled false
39             proguardFiles getDefaultProguardFile('proguard-android-optimiz
40             e.txt'), 'proguard-rules.pro'
41             signingConfig signingConfigs.release
42         }
43     }
44 }
```

```
42     }
43     compileOptions {
44         sourceCompatibility JavaVersion.VERSION_1_8
45         targetCompatibility JavaVersion.VERSION_1_8
46     }
47 }
```