User's Manual

飞凌嵌入式 FORLINX EMBEDDED

FCU1201 嵌入式控制单元

Embedded Control Unit

Product Manual_Android6.0 系统
Rev. 1.1
2021/04/09

Forlinx Embedded Technology Co. Ltd. www.forlinx.com

注意事项与维护

1、注意事项

- 请勿带电插拔核心板及外围模块！
- 请遵循所有标注在产品上的警示和指引信息。
- 请保持本产品干燥。如果不慎被任何液体泼溅或浸润，请立刻断电并充分晾干。
- 使用中注意本产品的通风散热，避免温度过高造成元器件损坏。
- 请勿在多尘、脏乱的环境中使用或存放本产品。
- 请勿将本产品应用在冷热交替环境中，避免结露损坏元器件。
- 请勿粗暴对待本产品，跌落、敲打或剧烈晃动都可能损坏线路及元器件。
- 请勿使用有机溶剂或腐蚀性液体清洗本产品。
- 请勿自行修理、拆卸本公司产品，如产品出现故障请及时联系本公司进行维修。
- 擅自修改或使用未经授权的配件可能损坏本产品，由此造成的损坏将不予以保修。

2、售后维修

如产品使用过程中出现硬件故障可根据售后服务政策进行维修

服务政策：参见官方网站 www.forlinx.com 售后服务说明；
地　　址：河北省保定市高开区向阳北大街 2699 号保定飞凌嵌入式新楼 5 层售后维修部
联 系 人：售后维修部
电　　话：0312-3102650-952/953　　邮编：071000
邮寄须知：建议使用顺丰、圆通或韵达，且不接收任何到付

技术支持与定制

1、技术支持方式

1.1 电话：0312-3119192

1.2 论坛：bbs.witech.com.cn

1.3 邮箱：

Linux 技术支持：	linux@forlinx.com
Android 技术支持：	android@forlinx.com
硬件技术支持：	hardware@forlinx.com

1.4 知识库：bbs.witech.com.cn/kb

2、技术支持时间

周一至周五: 上午 9:00—11:30，下午 13:30—17:00；

公司按照国家法定节假日安排休息，在此期间无法提供技术支持，请将问题发送至邮箱或论坛技术支持区，我们会在工作日尽快给您回复。

3、定制开发服务

我公司提供嵌入式操作系统底层驱动、硬件板卡的有偿定制开发服务，以缩短您的产品开发周期。

了解定制流程：http://www.forlinx.com/OEM.htm

填写需求文档：http://www.forlinx.com/docs/PR.docx

发至项目邮箱：project@forlinx.com

资料更新与获取

1、资料的更新

产品相关资料会不断的完善更新，本手册内容亦然如此；当您在使用这些内容时，请确保其为最新状态。

2、更新后如何通知

飞凌嵌入式产品资料更新通知采用微信公众号推送，敬请关注！

订阅号

3、资料如何获取

3.1 网络下载：

请注册并登陆“bbs.witech.com.cn”找到“开发板资料下载”选择对应平台下载；

下载前请阅读《资料下载说明》：http://bbs.witech.com.cn/thread-67932-1-1.html。

3.2 光盘：

请联系我公司销售人员购买。

版权声明

更新记录

日期	版本	更新内容
2020.03.02	V1.0	FCU1201 嵌入式控制单元 Android6.0 用户手册初版。
2021.04.09	V1.1	增加了看门狗支持； 增加了 Wifi 模块 8723DU1 的支持； 增加 4G 模块 EC20 的支持

目录

第一章 FCU1201 产品介绍

1.1 产品简介

FCU1201 嵌入式控制单元采用 NXP i.MX6Dual Lite 双核处理器（或者选配 NXP i.MX6Quad 四核处理器）开发设计，具有超高效、高性能、接口丰富等优势。主频高达 1GHZ，1GB/2GB DDR3，8GB eMMC，内部集成 RS485、CAN、ESAM、PSAM、USB、以太网口、4G、WiFi、LVDS 屏、HDMI、DI、DO、音频功能接口和模块，以满足不同场合的需求。

产品特点：

- 采用 NXP 的 i.MX6Dual Lite 处理器（或者选配 NXP i.MX6Quad 处理器），高性能、低功耗、高可靠性
- 核心模块所有元器件达到工业级-40 至 85℃温度范围
- 支持 ISO7816 协议，可直接与国家电网 ESAM/PSAM 模块通信
- 内置超级电容，断电后至少可维持系统正常运行 15 秒，确保信息不丢失
- 采用模块化设计，可迅速针对客户的个性化需求提供私人定制服务
- 4 路 DI、4 路 DO，2 路 485、2 路 CAN，均采用电气隔离和接口保护，安全可靠
- 可通过 TF 卡升级系统，无需拆卸外壳，简单方便
- 额定电压 12V，内置超级电容，安全可靠运行
- 采用铝合金型材外壳，带耳体积仅 100×147.5×41.8mm，体积小巧、外形美观、安装方便
- 采用 7 吋 LVDS 触摸屏（深圳拓普微提供），具备友好的人机交互界面
- 标准 MINI HDMI 接口，支持 1080P、720P 高清显示屏
- 全面的状态指示灯，令系统运行、网络通讯、接口连接……所有状况一目了然
- 通讯方式多样，板载千兆网口、Wi-Fi&蓝牙、4G 模块
- 标准 DB9 调试串口
- 标准 TF 卡插槽，方便扩充本地存储空间
- 标准 3.5mm 立体声耳机接口，内置话筒，亦可增设 1W×2 喇叭或 3.5mm 单声道话筒接口

1.2 应用领域

FCU1201 嵌入式控制单元适用于充电桩、广告牌、新零售、安防、车载、电力通讯等领域。

1.3 硬件参数

设备	描述	
CPU	NXP i.MX6Dual Lite ARM Cortex-A9 双核 1GHz	NXP i.MX6Quad ARM Cortex-A9 四核 1.0GHz
RAM	DDR3 1GB/2GB	
ROM	eMMC 8GB	
外扩存储	标准 TF 卡接口，最大支持 64GB（实测）	
移动通信	华为 ME909S 模块、移远 EC20 模块； 支持中国移动 4G/3G/2G、中国联通 4G/3G/2G； 采用标准 SIM 卡槽（卡槽上方标有 4G 标识）	
ESAM	支持 ESAM 芯片，ISO7816 协议； 飞凌提供读写驱动，用户自行购买板载芯片	
PSAM	支持 PSAM 卡； 采用抽屉式 Mini SIM 卡槽（卡槽下方有“SIM”标识） 飞凌提供读写驱动，用户自行买卡	
开关量输出	4 路，电磁继电器隔离； 触点容量：1A 30VDC / 0.5A 125VAC / 0.3A 60VDC 接口：3.81mm 间距绿端子	
开关量输入	4 路，光耦隔离，采用 3.81mm 间距绿端子； 默认配置为：直流电压输入，3V 至 24VDC 被判定为高电平，1VDC 以下被判定为低电平； 亦可配置为：内部提供隔离的 5V 电源，外部仅提供干接点。	
触摸彩屏接口	采用标准 DVI-I 插座； 默认 LMT070DICFWD-AKA 液晶显示器	
Mini HDMI接口	采用标准 mini HDMI 插座	
断电应对措施	CPU 有一路 GPIO 专门用于监测外部电源状态； 超级电容至少可维持系统运行 15 秒； 监测整机内部的 5V 主电源电压，当该电压跌落超过 10%时，整机断电，以免系统电压过低导致软件异常。	
串口（包含RS-485）	UART1：三线调试串口，DB9 插座，非隔离，机壳标识是 Console UART3：在机内转换为 RS-485-1，隔离电压 1.5KV，静电四级防护 UART4：在机内转换为 RS-485-2，隔离电压 1.5KV，静电四级防护 UART5：接读卡器，三线，绿端子引出，非隔离	
USB	1 个 USB OTG 接口，采用标准 Micro USB 插座 1 个 USB 主口，采用标准 USB A 型插座	
CAN BUS	CAN1：CAN2.0 B，1Mbps，隔离电压 1.5KV，静电四级防护 CAN2：CAN2.0 B，1Mbps，隔离电压 1.5KV，静电四级防护	
以太网路	标准 RJ-45 插座，10M/100/1000M 自适应	
蓝牙和Wi-Fi	采用 F23BUUM13-W2/BL-M8723DU1 Wifi 模块； 支持 IEEE 802.11b/g/n 1T1R WLAN and Bluetooth 2.1/3.0/4.0	
实时钟	采用专用 RTC 芯片 RX8010SJ； 板载 CR2032 电池，可至少维持 1 年走时	
音频	3.5mm 标准立体声耳机插座，内置单声道话筒； 可增设 1W×2 喇叭插座或 3.5mm 单声道话筒接口（PCB 有预留插座焊盘，但挡板未开孔）	
复位按键	1 个，用于系统复位	
Boot按键	1 个，与复位按键同时使用，用于系统固件更新	
电源与功耗	额定电压：12V，具备反接保护； 配拓普微 LMT070DICFWD-AKA 液晶显示器时的总功耗为 5.1W	

尺寸	100mm*147.5mm*41.8mm(长*宽*高)
安装	8 只Φ3mm 螺钉
工作环境	湿度：5%～95%，无凝露。 工作温度：-40℃～70℃（注：WiFi 模块工作温度为 0℃～70℃） 存储温度：-40℃～85℃（注：WiFi 模块存储温度为-40℃～80℃）

1.4 软件参数

软件支持	详细描述
操作系统	Android6.0
文件系统	Ext4
GCC	4.9.x
RS485	提供 485 测试示例
以太网	10M/100M/1000 自适应以太网，支持静态/动态分配 IP 地址，支持修改 MAC 地址
WiFi	支持 STA、AP 功能
4G 网络	ME909 模块、移远 EC20 模块，支持移动、联通 4G 上网功能

第二章 FCU1201 功能介绍

2.1 接口示意图

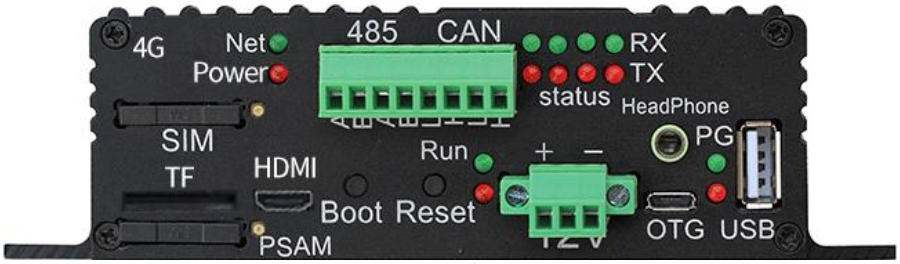


图 1 左侧接口图

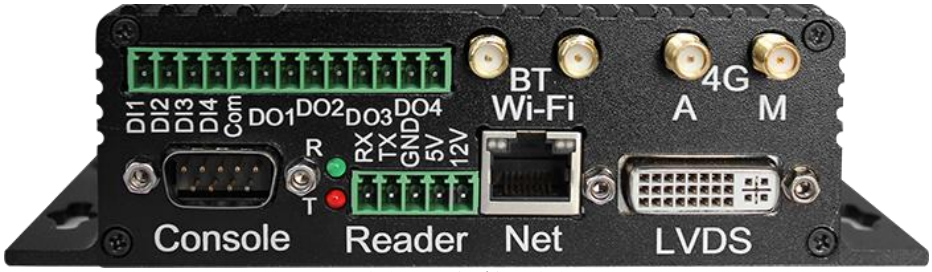
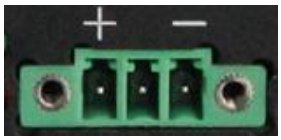


图 2 右侧接口图

2.2 电源供电

本产品支输入为 DC12V，飞凌提供适配器为 12V 2A。

实物图	说明	
	**引脚序号**	**引脚说明**
	1	电源正极，DC 12V
	2	保护地
	3	电源负极，GND

电源供电状态可通过电源指示灯显示，如下图：

CPU 有一路 GPIO 专门用于监测外部电源状态，当外部电源电压高于 8V 时，该 IO 为高电平，外壳上的 PG 绿灯点亮；反之为低电平，同时 PG 绿灯熄灭。

当外部电源中断后，系统自动切换为内置的超级电容供电，同时面板上的 PG 红灯点亮。超级电容至少可维持系统运行 15 秒，同时监测整机内部的 5V 主电源电压，当该电压跌落超过 10%时，整机断电，以免系统电压过低导致软件异常。

2.3 调试串口

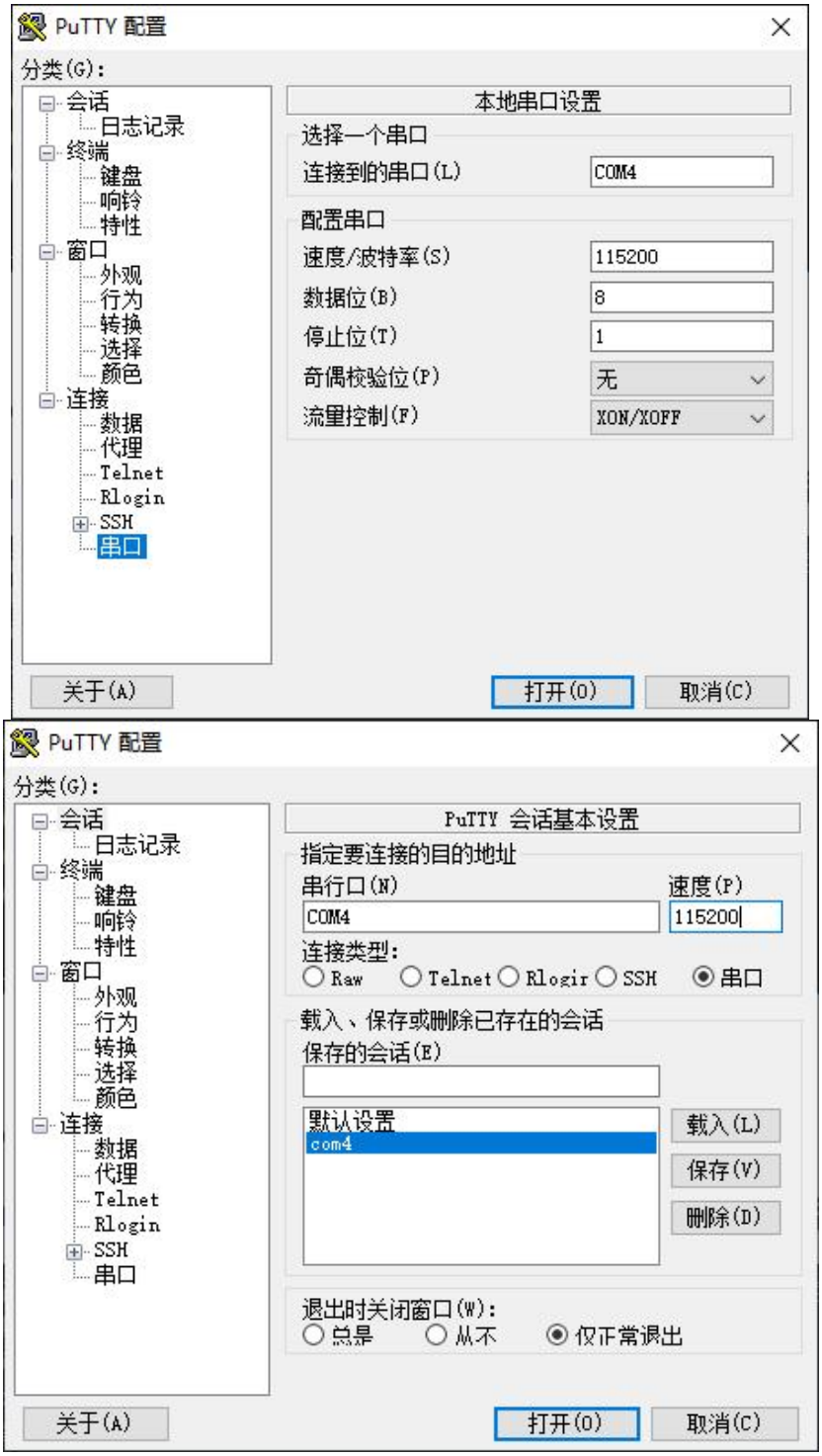
FCU1201 的调试串口在面板上标识符为 Console，如图所示：

调试串口 Console 用于查看板子运行信息，作为调试口，不做它用。

1、将 PC 和 FCU1201 控制单元的 Console 通过双母头交叉的串口线连接。

2、打开超级终端（Win7 系统的可以使用“用户资料\工具\putty.exe”）进行如下的设置：

根据您的串口连接进行选择串口号，并在左侧【串口】菜单设置相关内容。



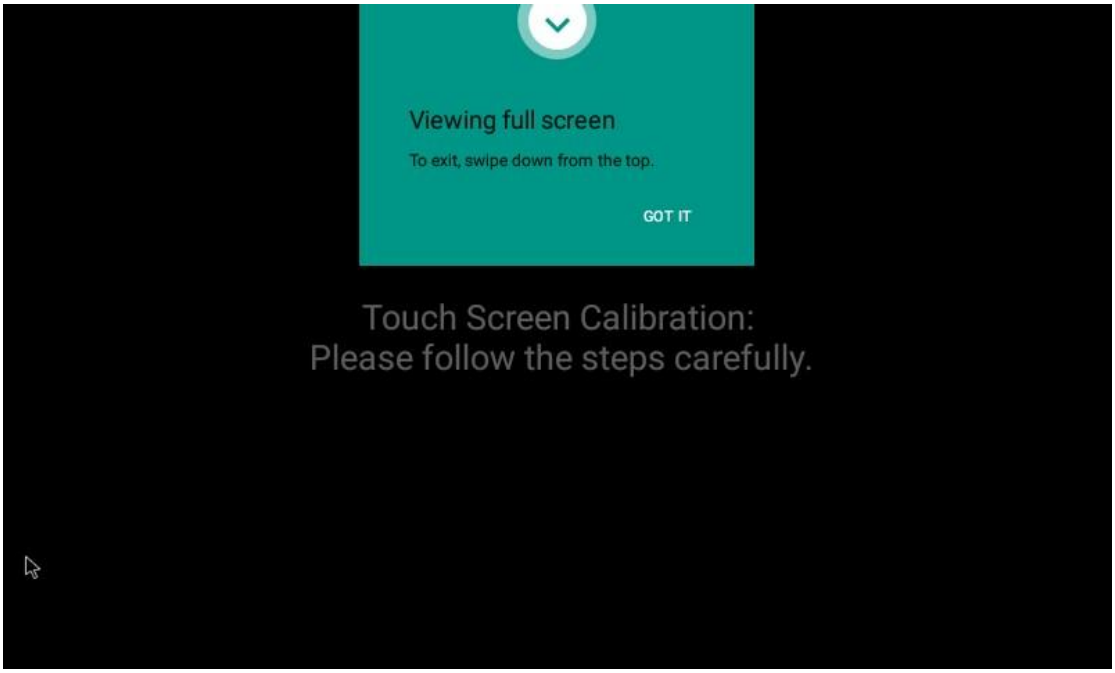
注意:此时采用的波特率为 115200，8 位数据位，无奇偶校验，1 位停止位，无数据流控制。设置完成后，启动 FCU1201 控制单元就可以看到调试信息了。

```
U-Boot 2016.03 (Feb 15 2020 - 16:48:24 +0800)

CPU:   Freescale i.MX6DL rev1.3 996 MHz (running at 792 MHz)
CPU:   Extended Commercial temperature grade (-20C to 105C) at 39C
Reset cause: POR
Board: MX6-SabreSD
I2C:   ready
DRAM:  1 GiB
PMIC:  PFUZE100 ID=0x10
MMC:   FSL_SDHC: 0, FSL_SDHC: 1, FSL_SDHC: 2
*** Warning - bad CRC, using default environment

Display: LDB-TOPWAY (800x480)
reading logo.bmp
1152054 bytes read in 45 ms (24.4 MiB/s)
In:    serial
Out:   serial
Err:   serial
check_and_clean: reg 0, flag_set 0
Fastboot: Normal
flash target is MMC:2
Net:   FEC [PRIME]
Error: FEC address not set.

Hit any key to stop autoboot:  0
boota mmc2
kernel   @ 14008000 (8478176)
ramdisk  @ 15000000 (1028988)
fdt      @ 14f00000 (51466)
## Booting Android Image at 0x12000000 ...
Kernel load addr 0x14008000 size 8280 KiB
Kernel command line: console=ttymxc0,115200 init=/init vmalloc=256M and
60,if=RGB24,bpp=32 ldb=sin1
## Flattened Device Tree blob at 14f00000
   Booting using the fdt blob at 0x14f00000
   Loading Kernel Image ... OK
   Using Device Tree in place at 14f00000, end 14f0f909
switch to ldo_bypass mode!

Starting kernel ...

Booting Linux on physical CPU 0x0
Initializing cgroup subsys cpu
Initializing cgroup subsys cpuacct
Linux version 4.1.15 (lixinguo@developer-RH2485-V2) (gcc version 4.9.x-
CPU: ARMv7 Processor [412fc09a] revision 10 (ARMv7), cr=10c53c7d
CPU: PIPT / VIPT nonaliasing data cache, VIPT aliasing instruction cach
Machine model: Freescale i.MX6 DualLite SABRE Smart Device Board
```

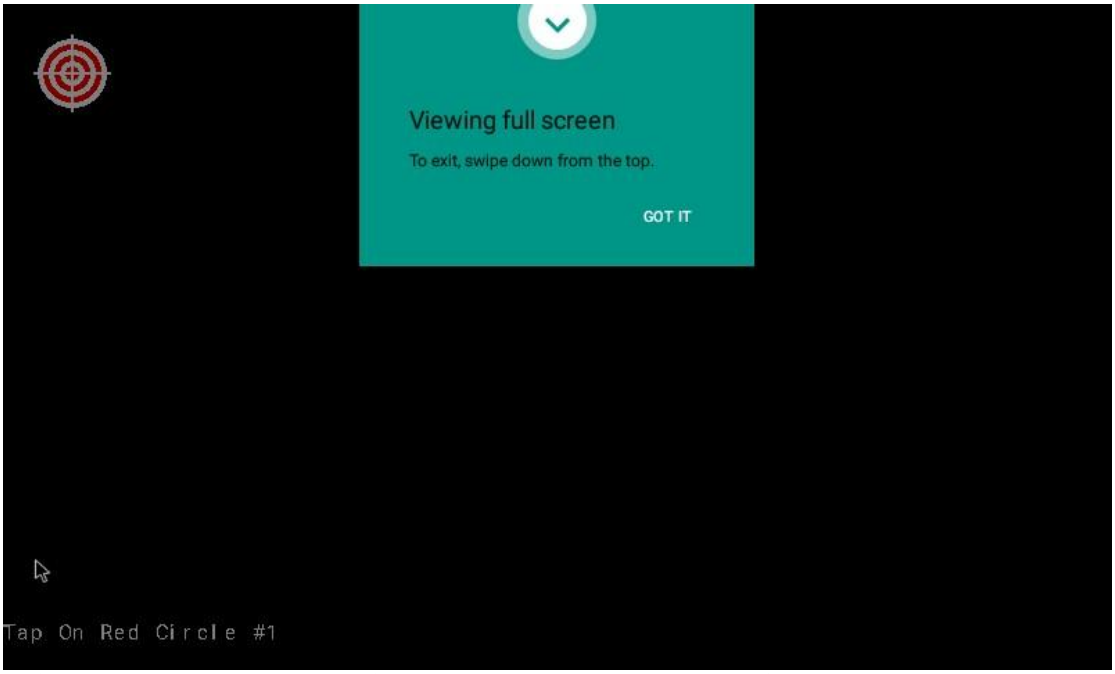
2.4 屏幕校准

LVDS 显示接口连接的电阻触摸屏需要校准后再使用，在 FCU1201 启动后，如果触摸未校准或检测到校准数据损坏，会启动校准程序，对触摸进行校准。

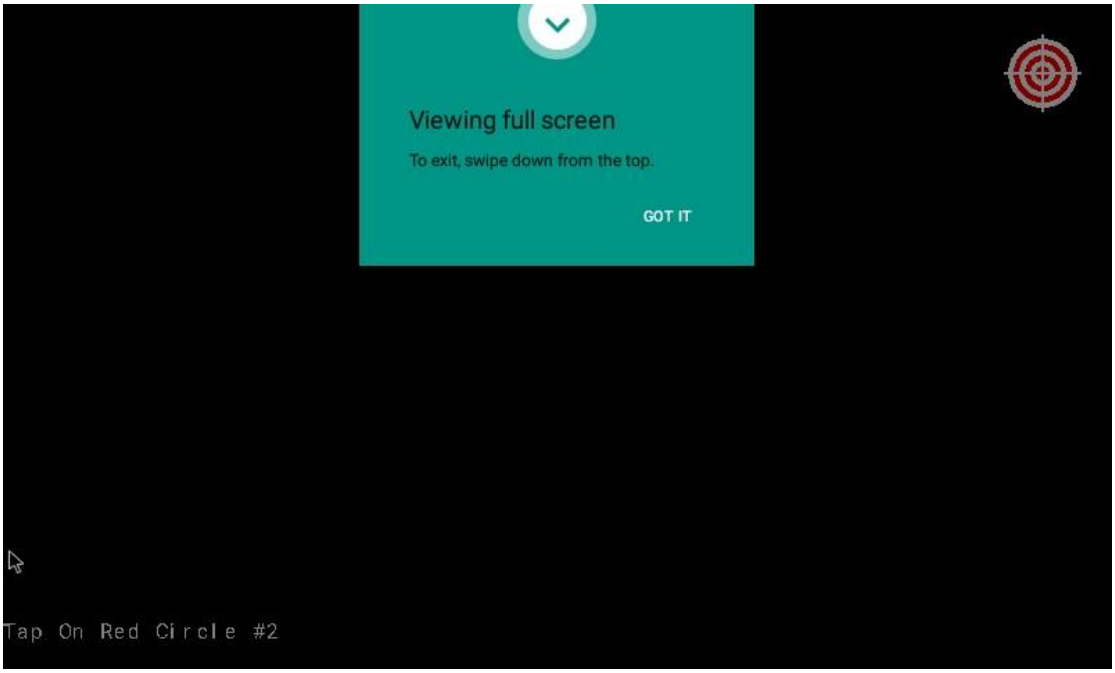
启动后的校准界面（如下图），会出现提示退出全屏从顶部下滑的界面（无需理会此提示，不影响校准），然后点击界面任意点，开始校准流程。



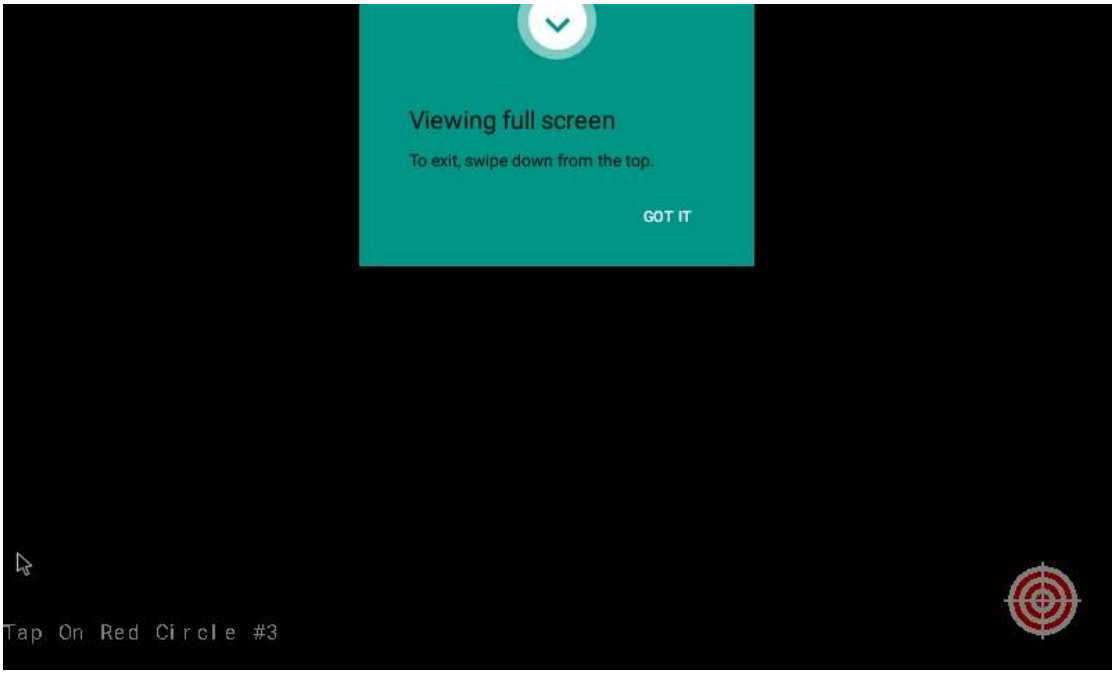
第一个校准点，点击校准点的正中心，（如下图）然后进入第二个校准点。



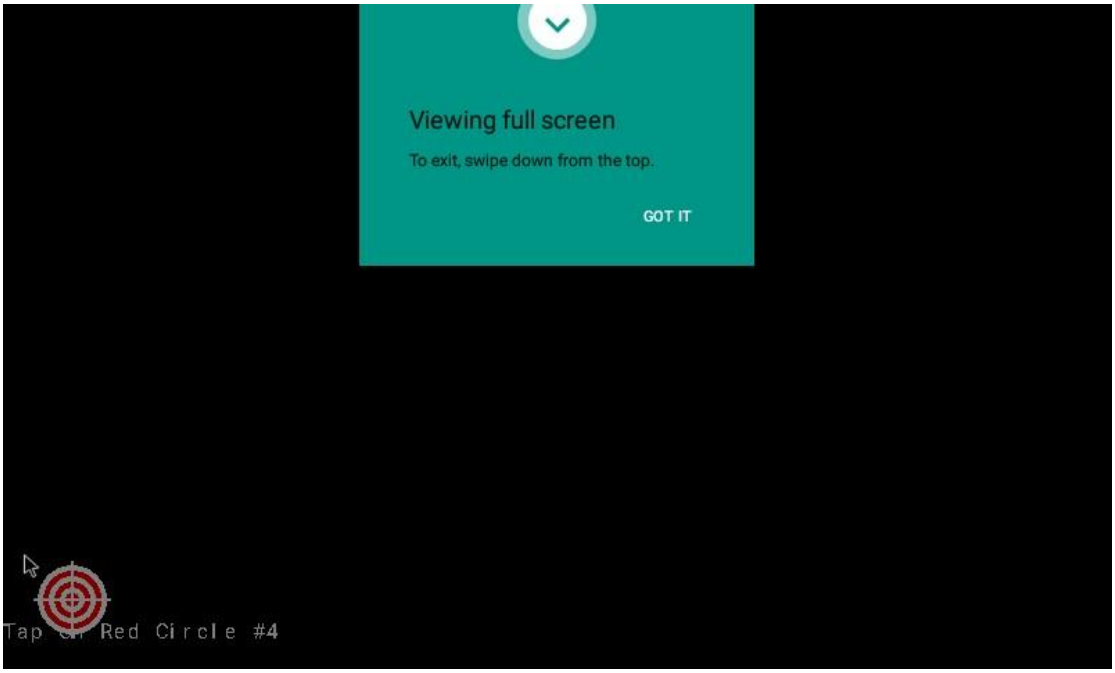
第二个校准点，点击校准点的正中心，（如下图）然后进入第三个校准点。



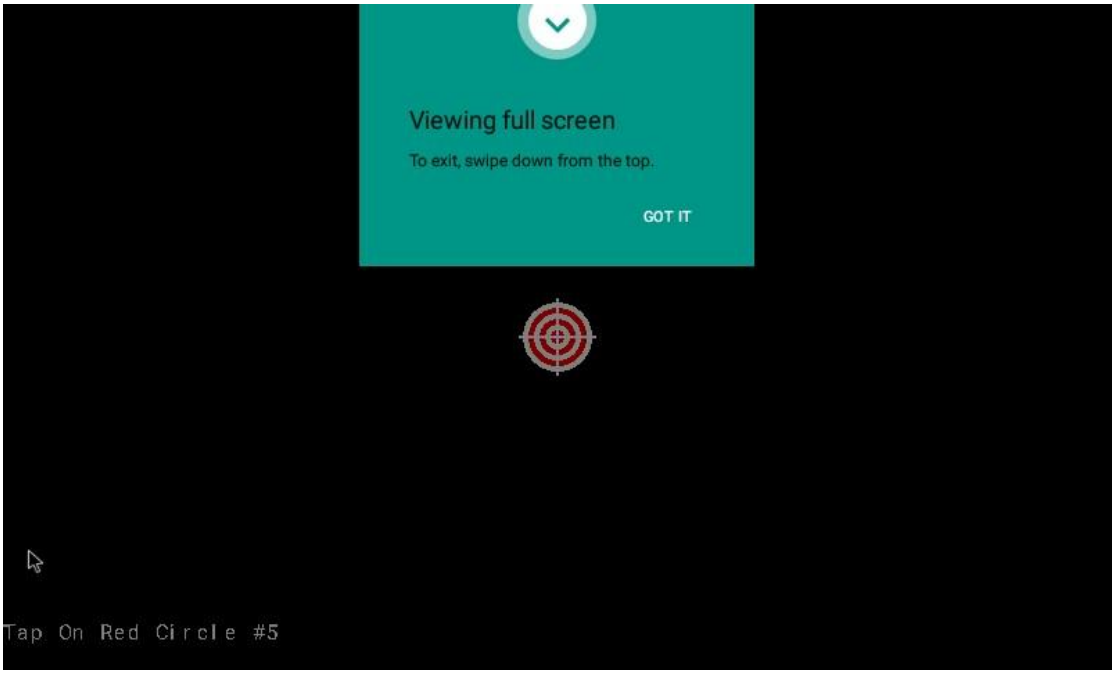
第三个校准点，点击校准点的正中心，（如下图）然后进入第四个校准点。



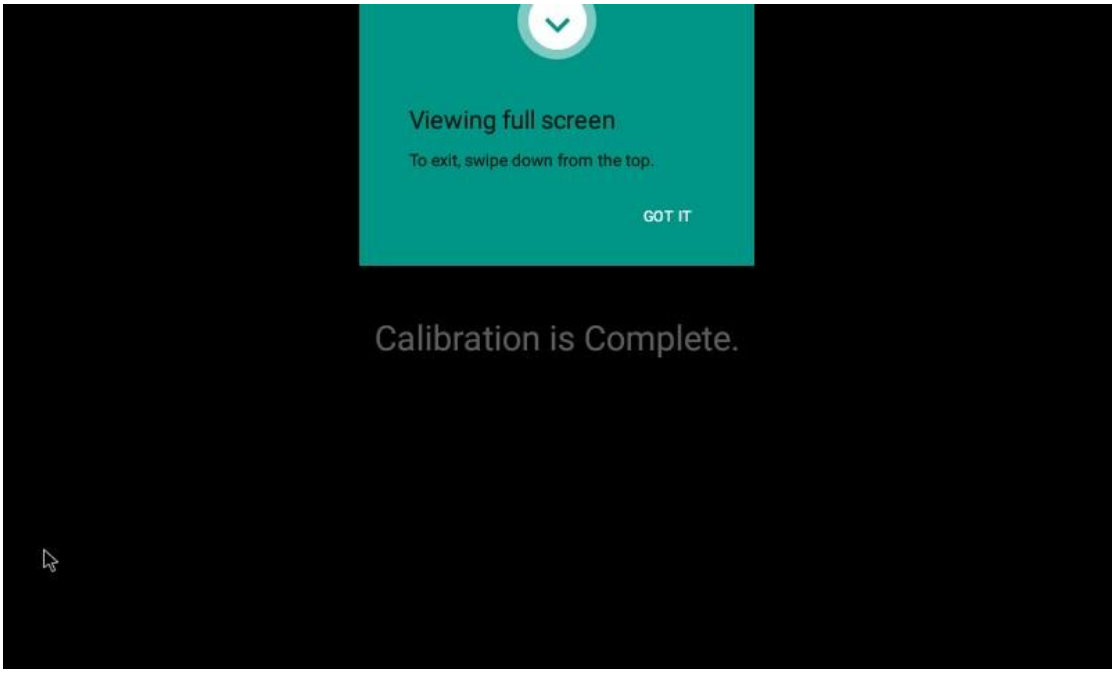
第四个校准点，点击校准点的正中心，（如下图）然后进入最后一个校准点。



最后一个校准点，点击校准点的正中心，（如下图）校准完成。



校准完成后，点击退成校准程序。（如下图）



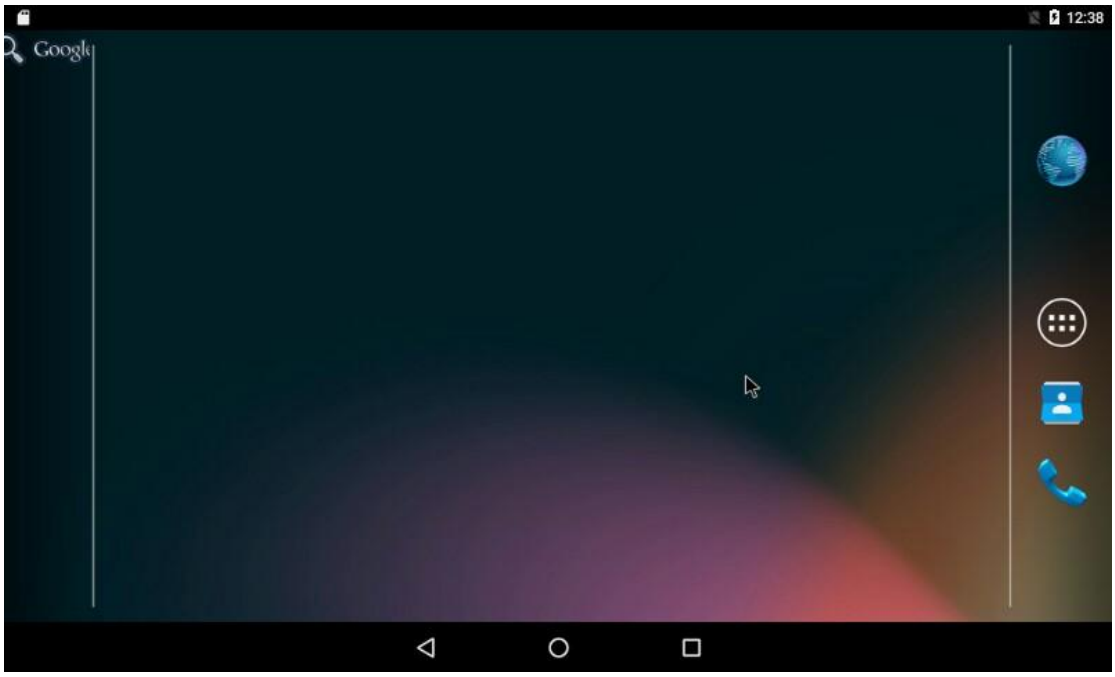
触摸校准完成。

注意：

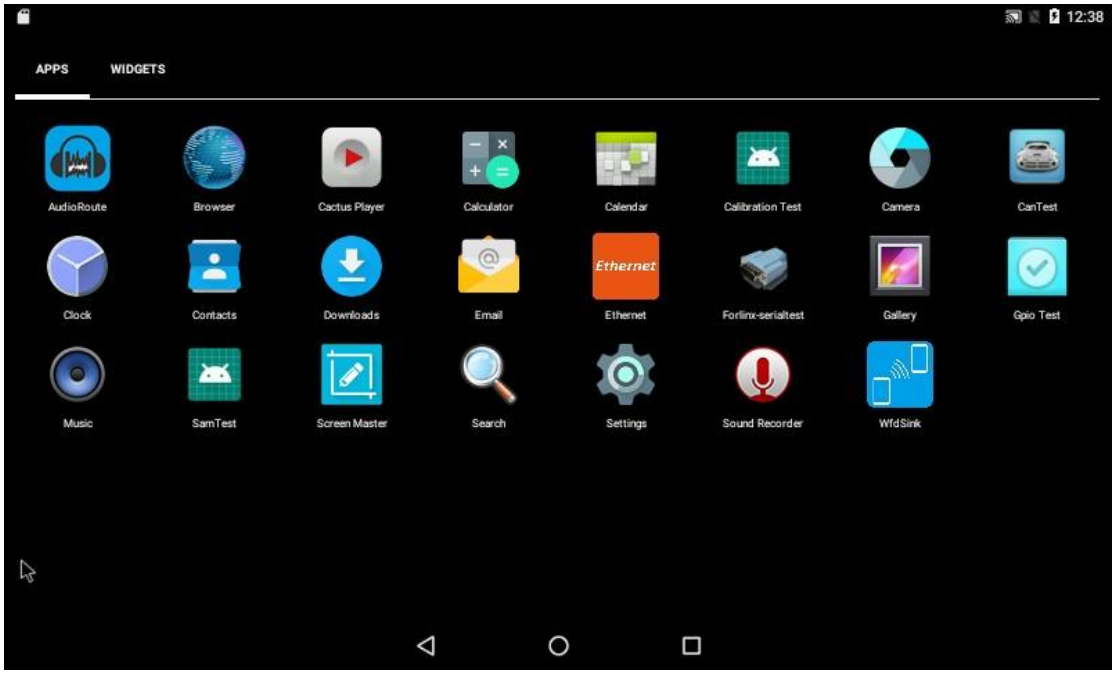
如果在校准后，更换显示屏后，触摸无法使用需要再次校准，此时校准无法使用，可以使用鼠标点击应用程序“Calibration Test”，进入校准程序按照上述步骤再次校准即可。

2.5 主界面

主界面如下图：



点击进入：



2.6 DI、DO

2.6.1 接口说明

DI、DO 接口如下图：

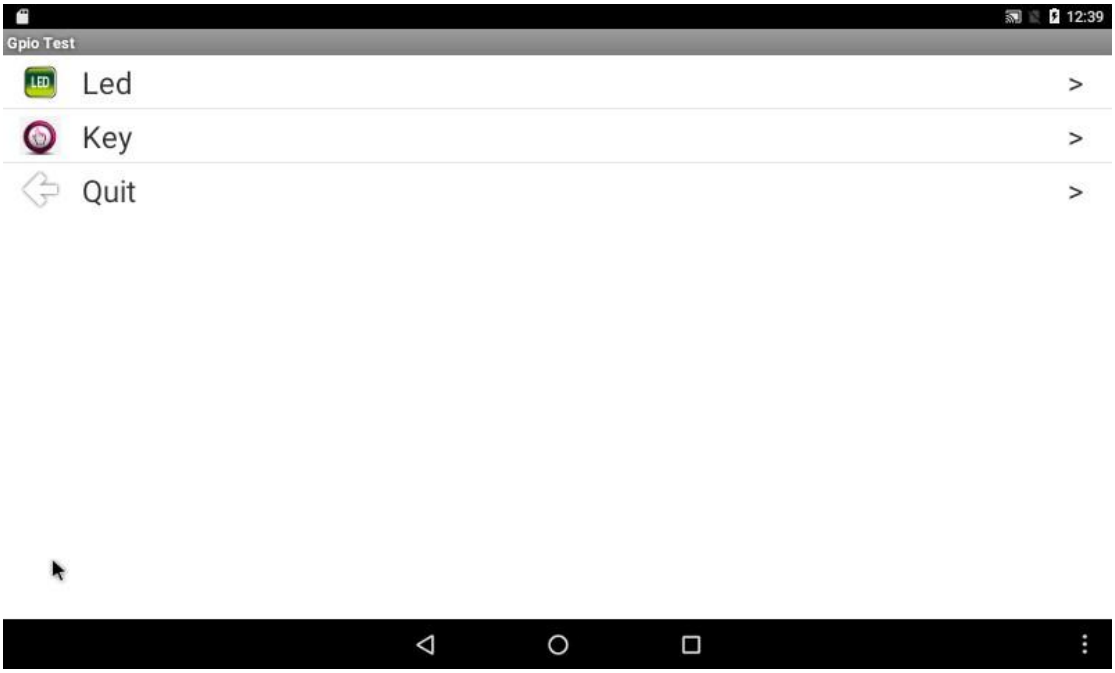
其中数字量输入接口能承受直流 24V 之内的开关量电压信号，3V 以上为高电平，1V 以下为低电平。这些输入口内部采用兼容设计，亦可根据用户需求，配置为机内 5V 供电，外部仅提供干接点的形式。

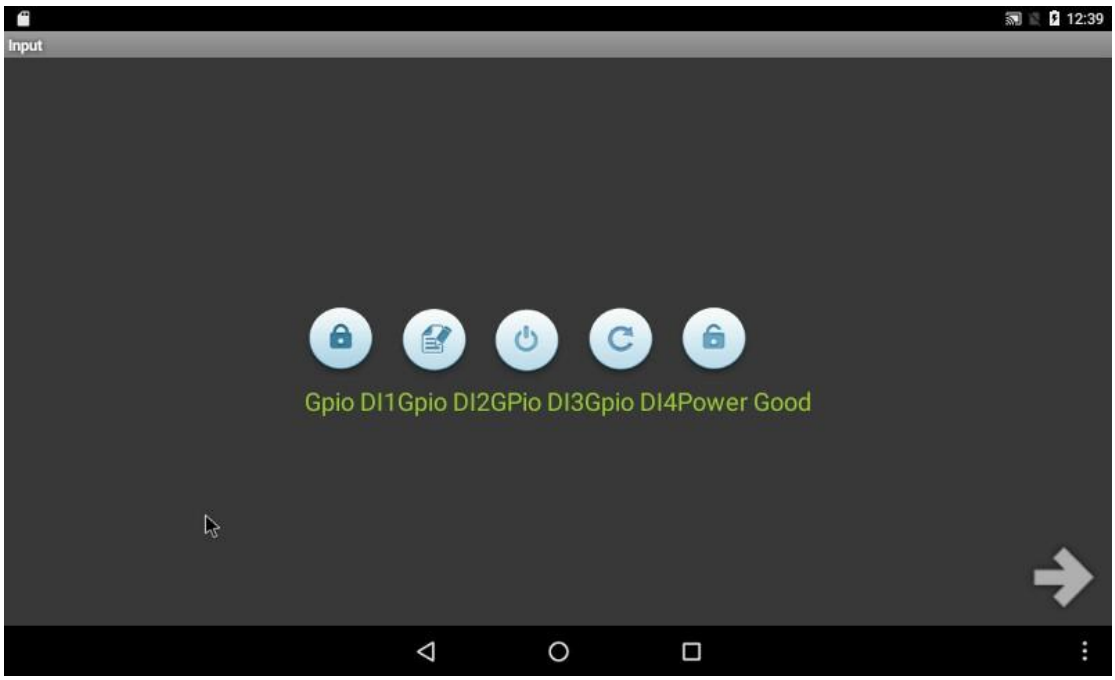
2.6.2 软硬件对应关系

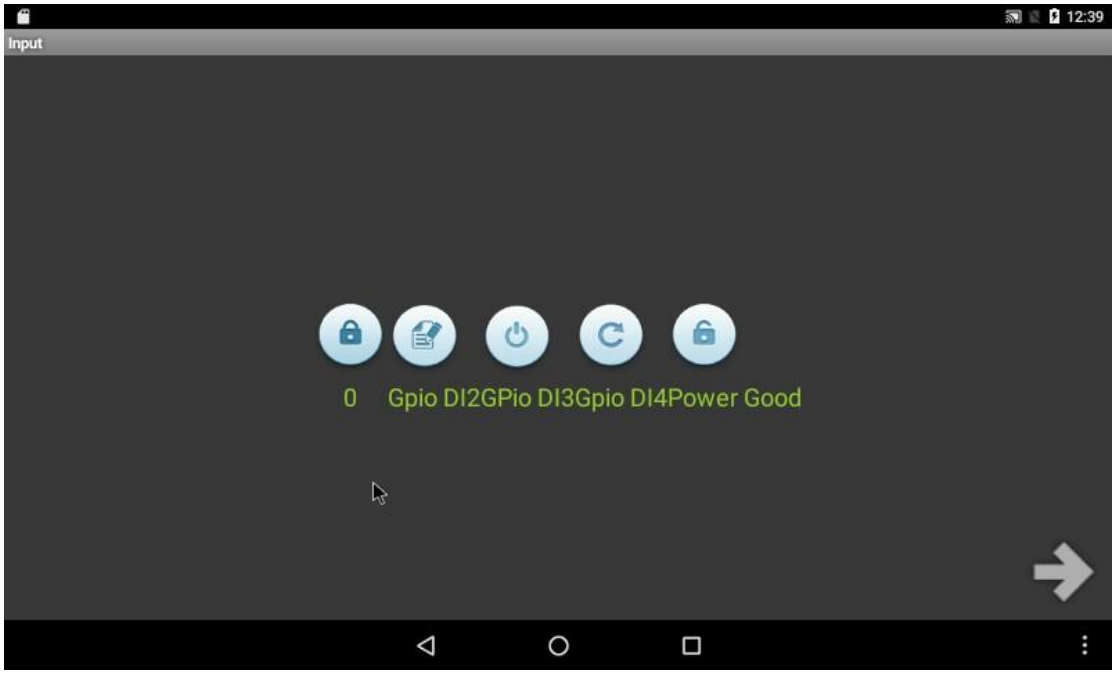
设备节点名称	硬件标识名称
/dev/gpio_input_0	DI1
/dev/gpio_input_1	DI2
/dev/gpio_input_2	DI3
/dev/gpio_input_3	DI4
/dev/gpio_input_3	POWER GOOD（掉电检测）
/sys/class/leds/do1/brightness	DO1
/sys/class/leds/do2/brightness	DO2
/sys/class/leds/do3/brightness	DO3
/sys/class/leds/do4/brightness	DO4
/sys/class/leds/run-red/brightness	RUN RED（红色LED指示灯）
/sys/class/leds/run-green/brightness	RUN GREEN（绿色LED指示灯）
/sys/class/leds/wifi/brightness	Wifi电源控制
/sys/class/leds/gprs/brightness	4G电源控制

2.6.3 DI 测试

在主界面点击“Gpio Test”进入 DI、DO 测试程序，点击“KEY”选项，进入 DI 测试。

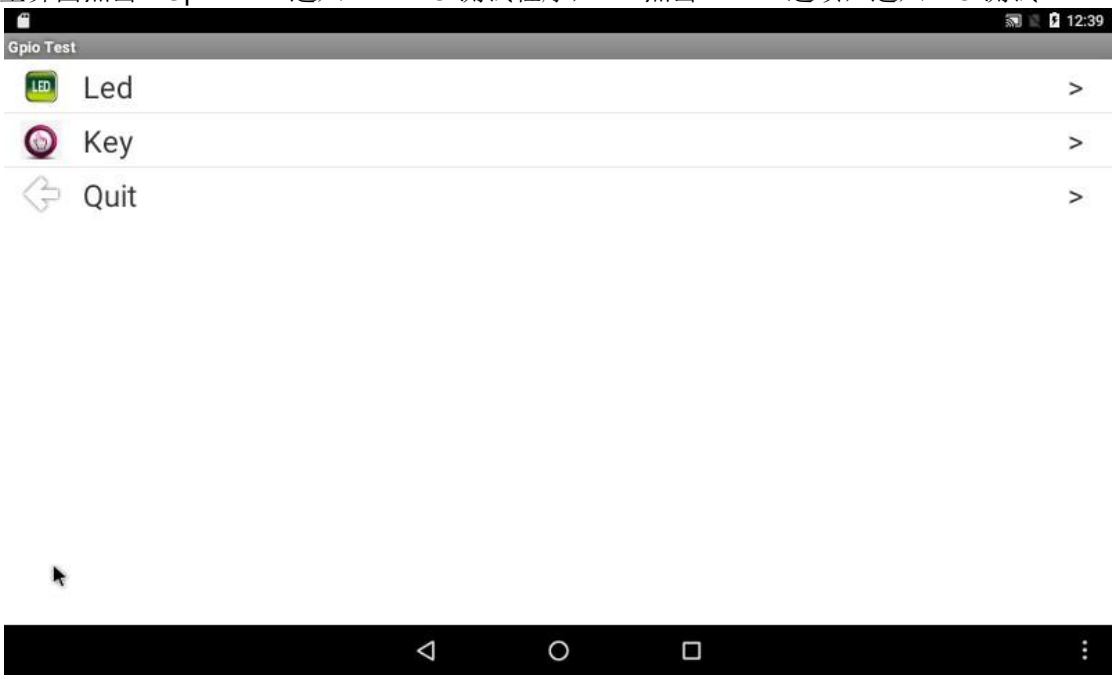

12:39
Gpio Test
Led
Key
Quit

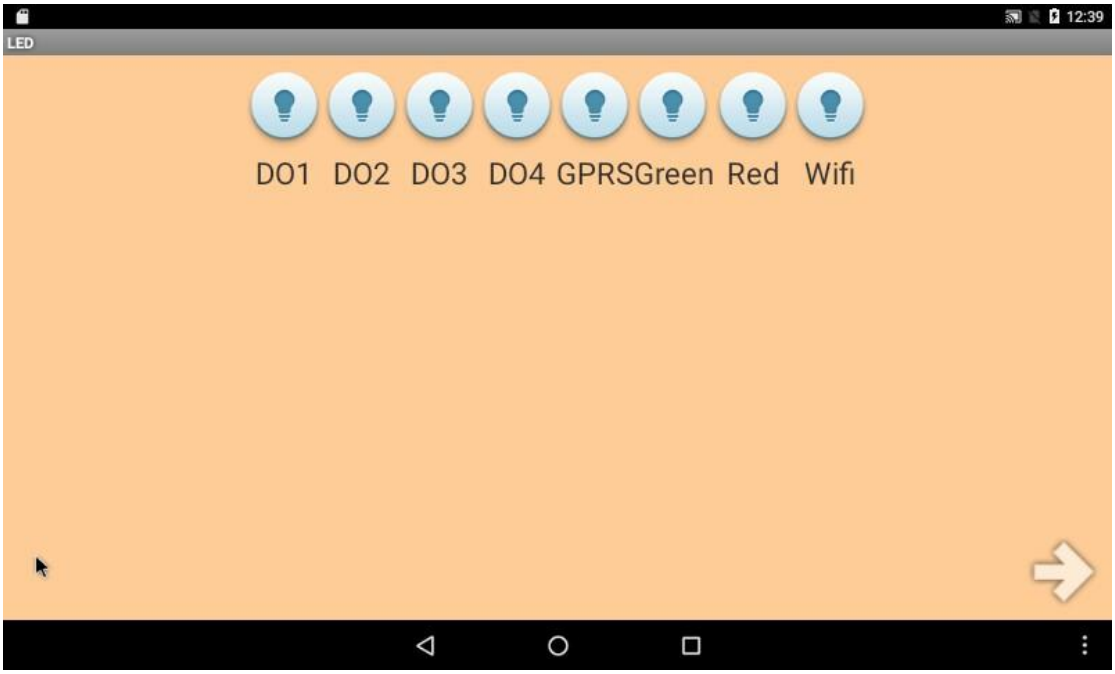

12:39
Input
Gpio DI1Gpio DI2GPio DI3Gpio DI4Power Good



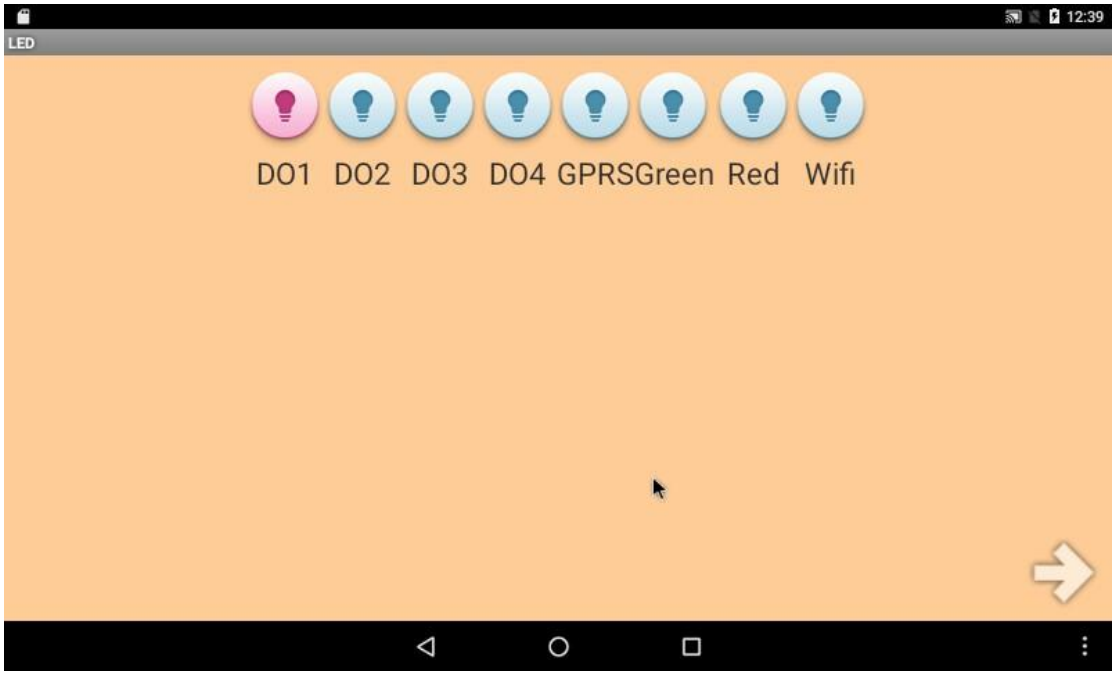
2.6.4 DO 测试

在主界面点击“Gpio Test”进入 DI DO 测试程序， 点击“LED”选项，进入 DO 测试。



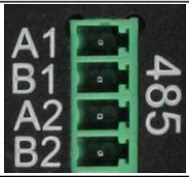


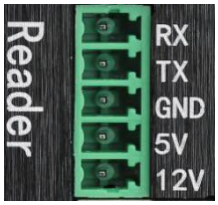
将 DO 对应的图标点亮则输出置使能，熄灭输出为禁止。（使能表示 led 点亮、DO 对应节点导通、电源控制导通）。



2.7 串口

2.7.1 线序说明

实物图	说明	
	引脚	引脚说明
	A1	RS485_1 Data+
	B1	RS485_1 Data-
	A2	RS485_2 Data+
	B2	RS485_2 Data-



引脚	引脚说明
RX	串口数据接收
TX	串口数据发送
GND	数字地
5V	5V 电源输出
12V	12V 电源输出

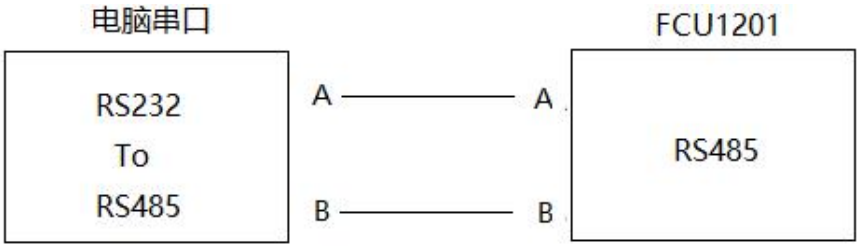
2.7.2 软硬件对应关系

设备节点名称	硬件标识名称
/dev/ttymxc2	RS485_1
/dev/ttymxc3	RS485_2
/dev/ttymxc4	Reader

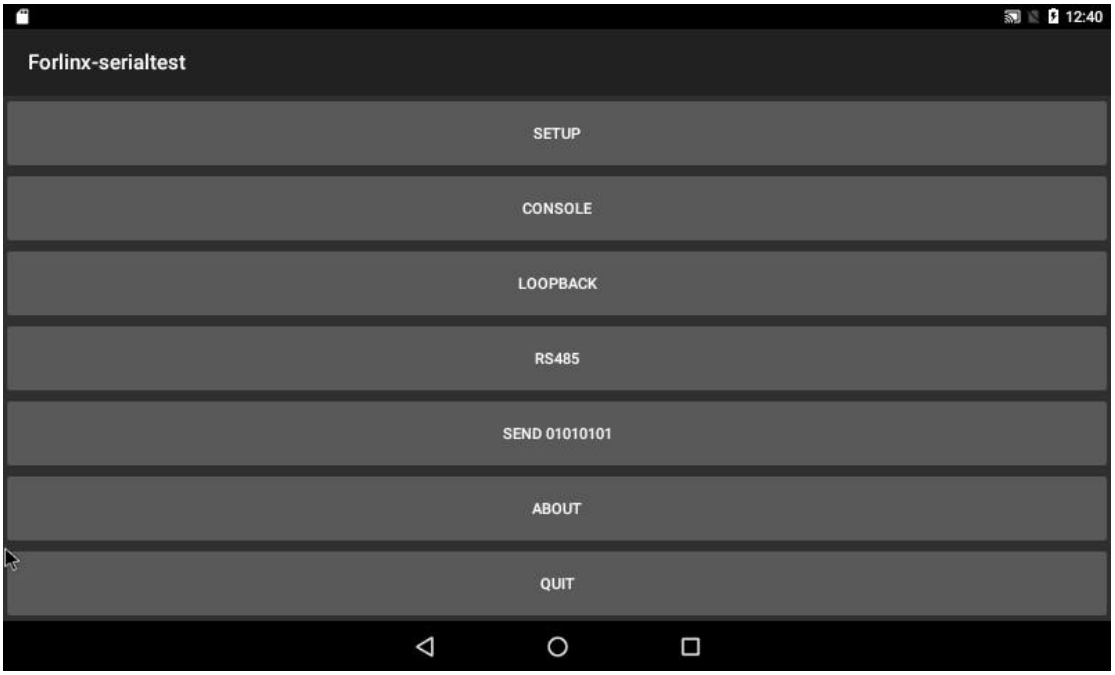
2.7.3 RS485 测试

以 RS485_1 为例，如需测试其他 RS485 接口，只需修改对应接口的设备节点名称即可；

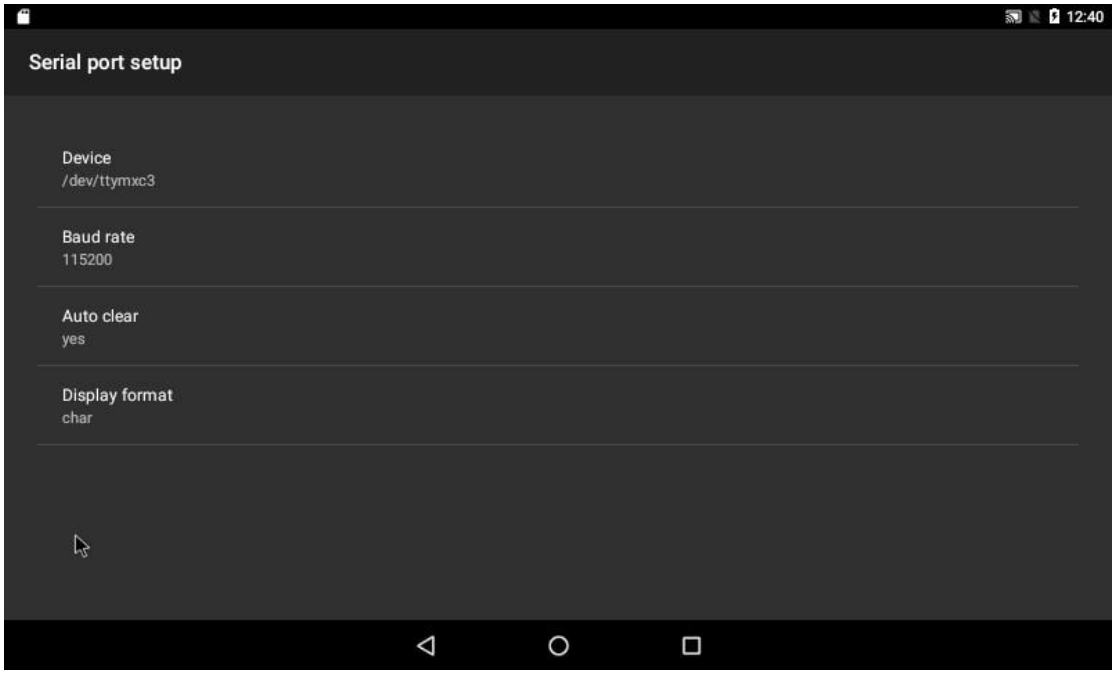
本测试方法以与电脑通讯为例，将 RS232 转 RS485 模块和 FCU1201 的 RS485_1 接口连接，连接方式如下所示：



点击主界面 Forlinx-serialtest 进入串口测试程序



点击 SETUP 选择测试串口并设置串口参数后，返回上一级，

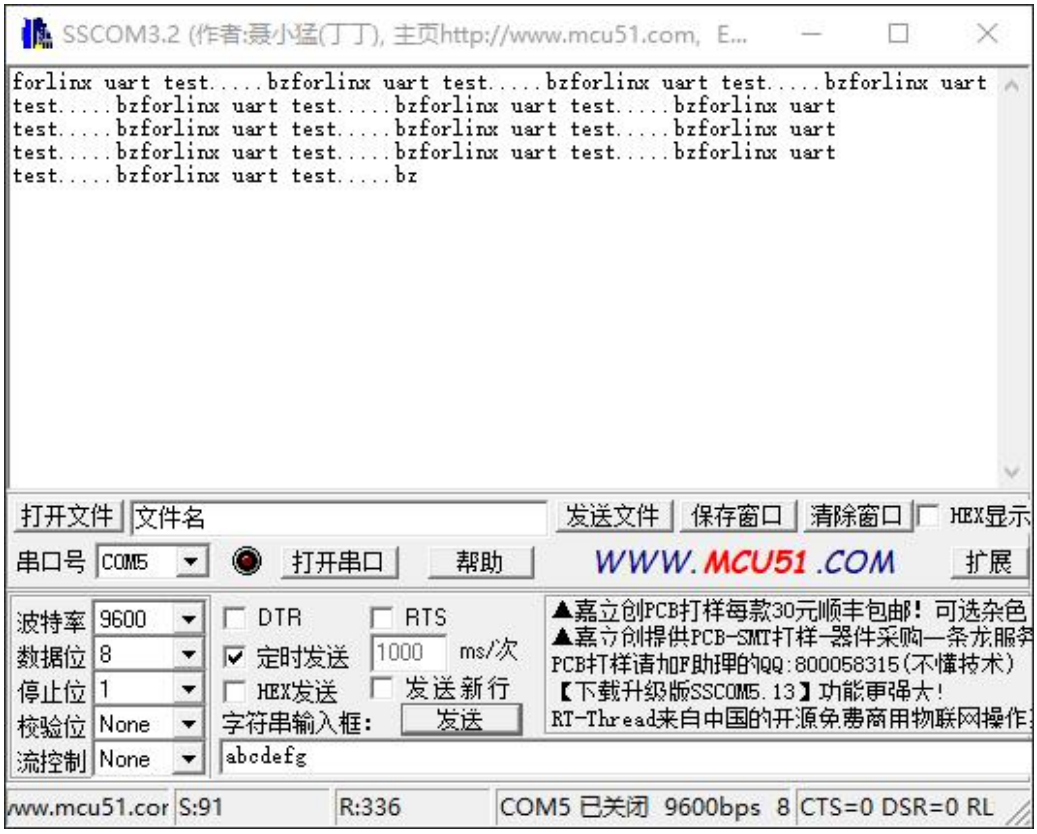


选择 CONSOLE 进行测试，在发送框输入 forlinx uart test…..bz 后点击 SEND 按钮向 PC 端发送数据，PC 端串口会收到此数据。

PC 端向 FCU1201 发送数据 abcdefg，FCU1201 也能正常接收数据。

FCU1201 串口测试程序接收和发送数据如下：

PC 端的串口助手软件（用户资料/工具/sscom32.exe），发送与接收数据如下：



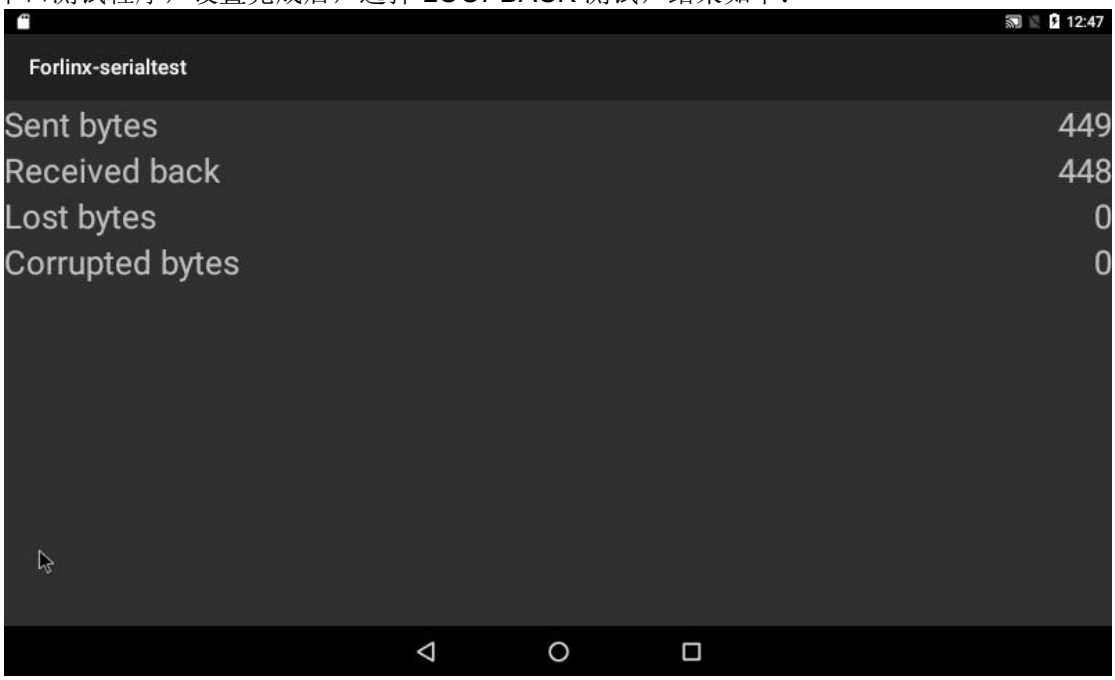
2.7.4 RS232 接口测试

FCU1201 嵌入式控制单元中除了调试串口 Console 之外还有一个串口，在 FCU1201 嵌入式控制单元面板上的标识符为 Reader，并提供对外 5V 与 12V 供电接口。

Reader 外设接口测试方法如下：

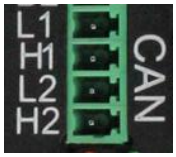
将 RS232 电平的 Reader 外设接口的 RX 和 TX 短接；

打开串口测试程序，设置完成后，选择 LOOPBACK 测试，结果如下：



2.8 FlexCAN 测试

2.8.1 FlexCAN 线序说明

实物图	说明	
	**引脚**	**引脚说明**
	L1	CAN1_L
	H1	CAN1_H
	L2	CAN2_L
	H2	CAN2_H

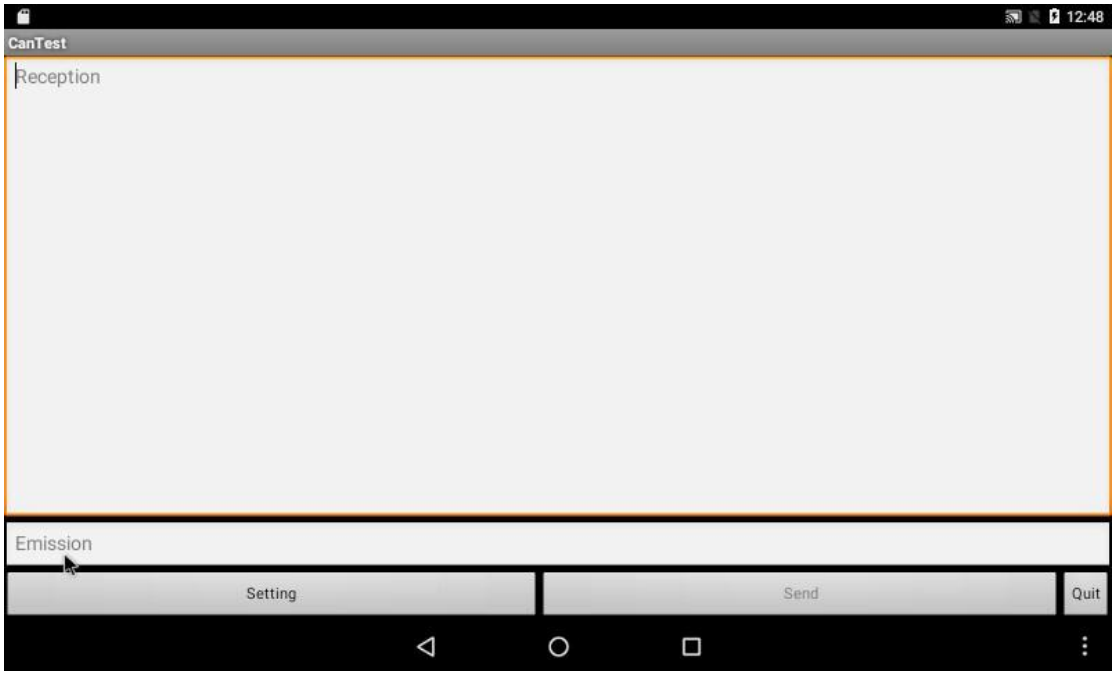
2.8.2 软硬件对应关系

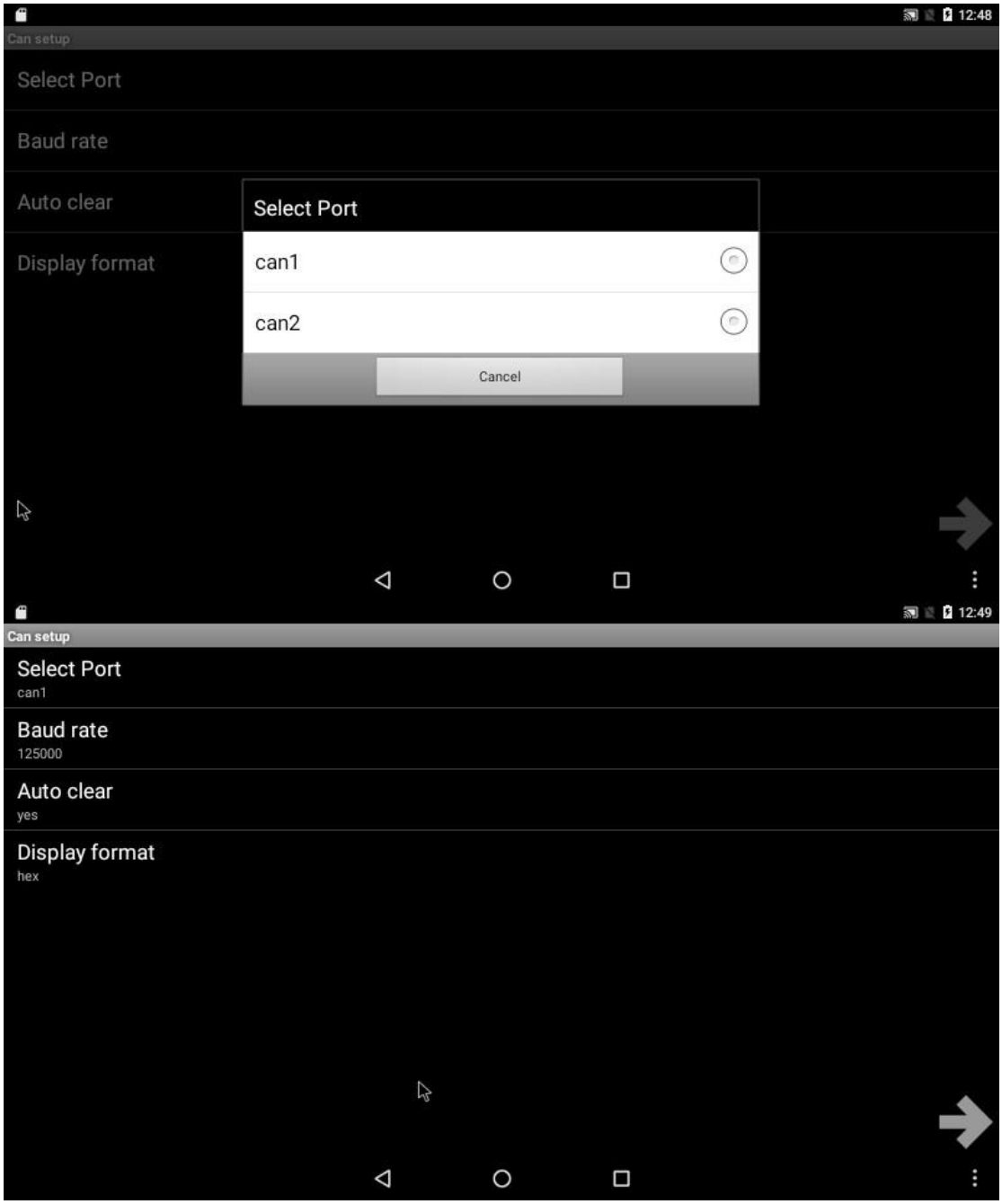
备节点名称	硬件标识名称
can0	CAN0
can1	CAN1

2.8.3 测试

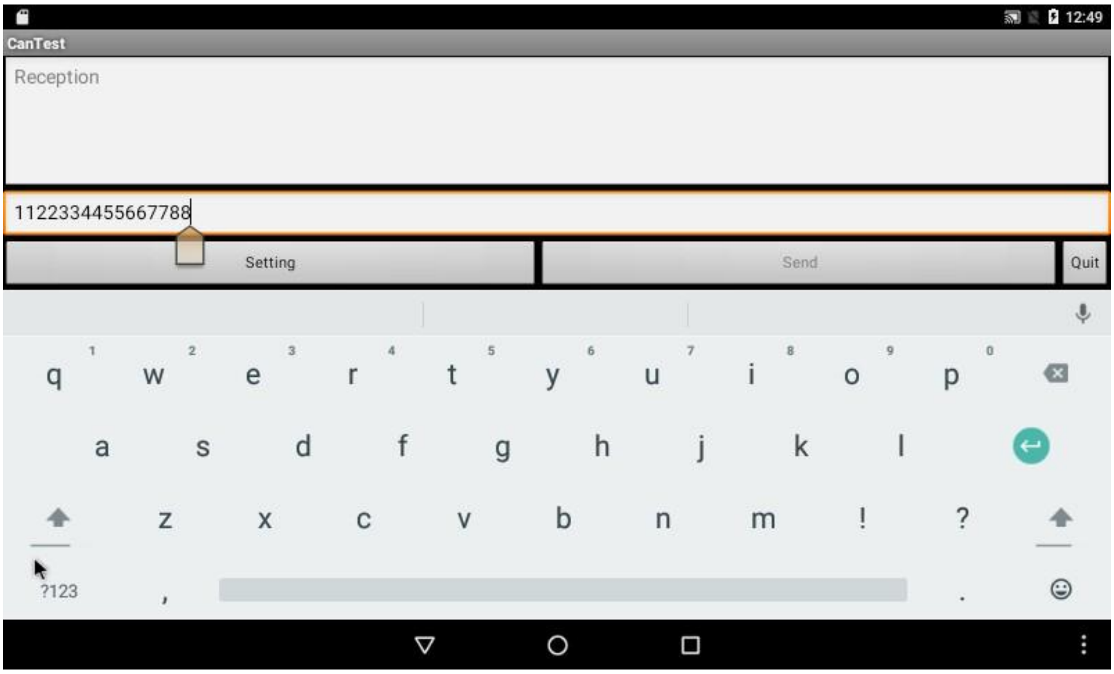
将测试的 CAN 接口与 CAN 测试工具的 CANL 和 CANH 连接好后开始测试。

打开 Can 测试程序 CanTest，点击 Setting 进入设置界面，选择测试的 Can 接口，并设置你需要的波特率，选择 HEX 格式或字符格式发送数据，设置完成后返回主界面。

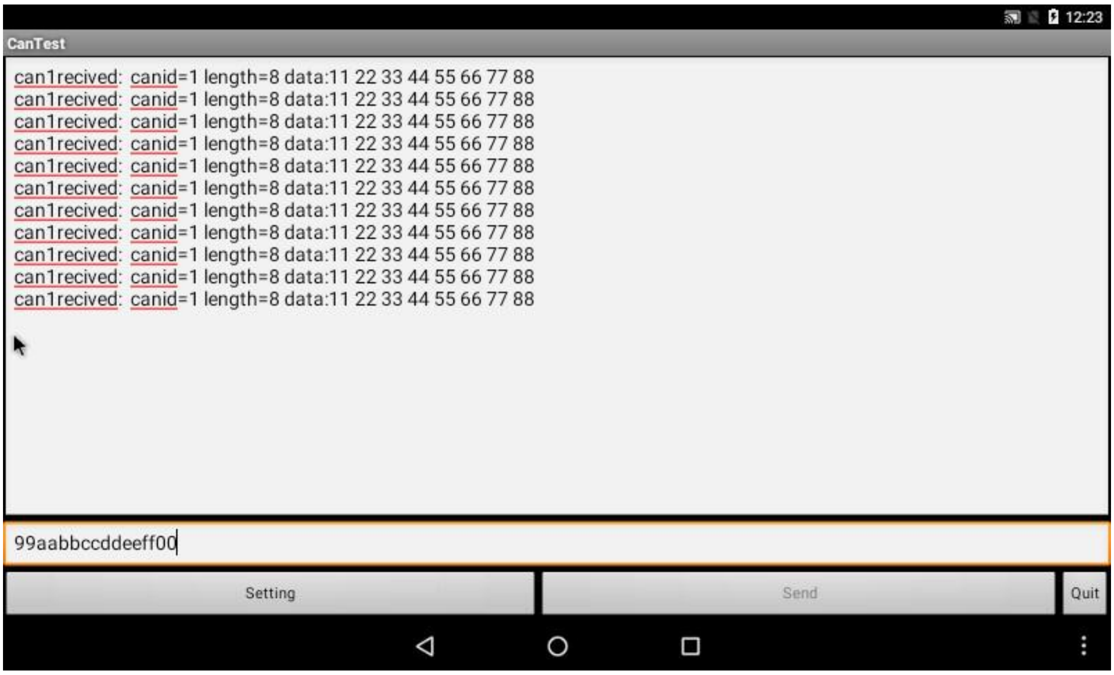




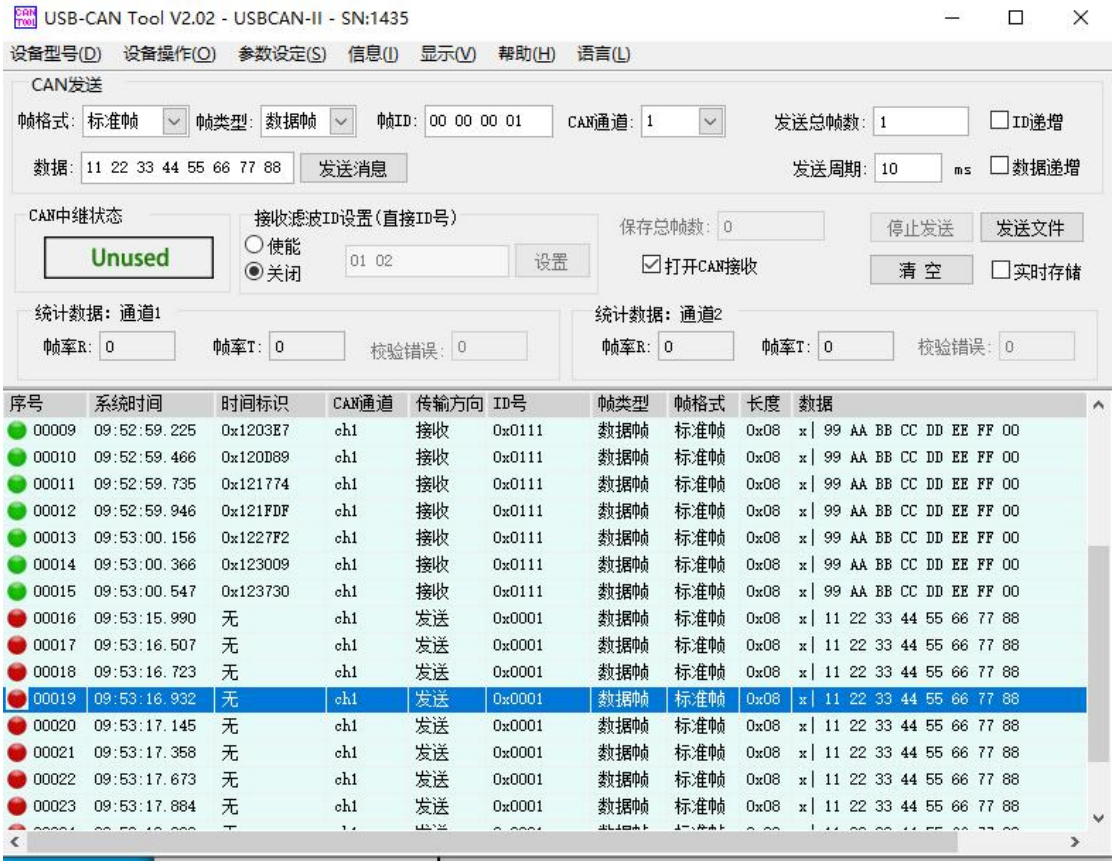
在发送文本框内输入你要发送的数据，HEX 格式下每个字节 2 位且无空格，发送的数据不能超过 8 个字节。



FCU1201 发送接受数据如下：



在 PC 端其数据发送接收如下：



2.9 播放音乐

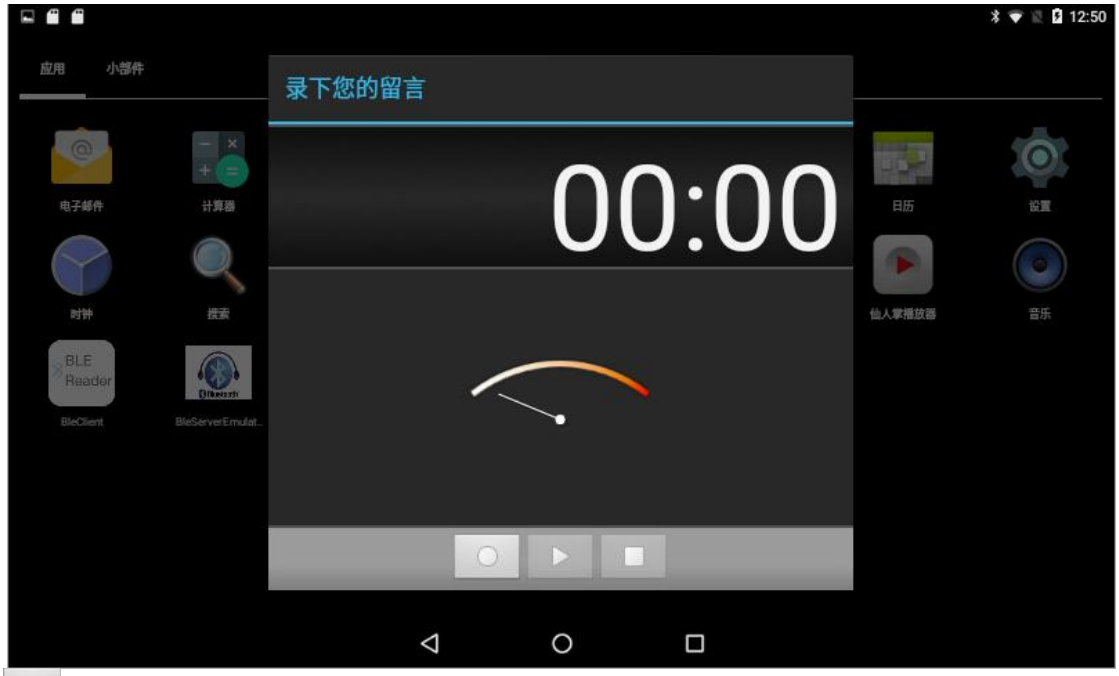
此功能需要将所播放的音乐放到 TF 卡或 U 盘中，选择“”->“”，然后选择想要播放的歌曲，点击播放。

注：FCU1201 未将 speaker 接口引出，需要插入耳机或连接带音频的 HDMI 设备。(耳机和 HDMI 的输出顺序为：连接耳机时，耳机为输出；不连接耳机时，连接 HDMI 设备时，HDMI 设备输出；)

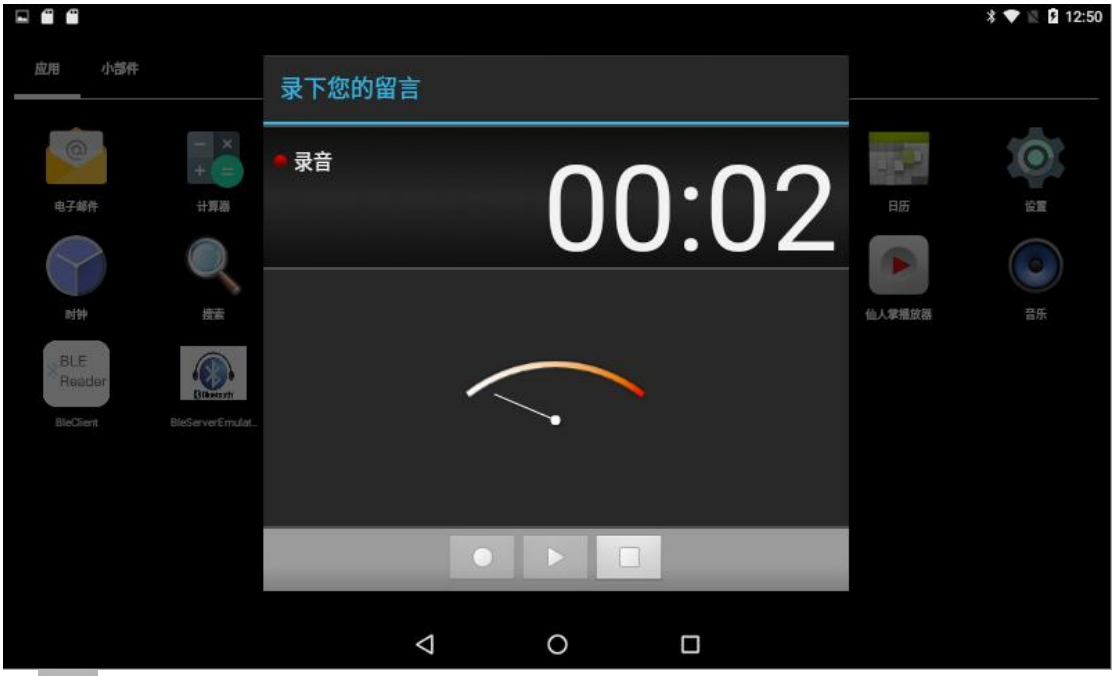
2.10 录音（板载 Mic 输入）

选择“”->“”进入录音，首次使用时会询问权限：

选择允许后进入录音界面：



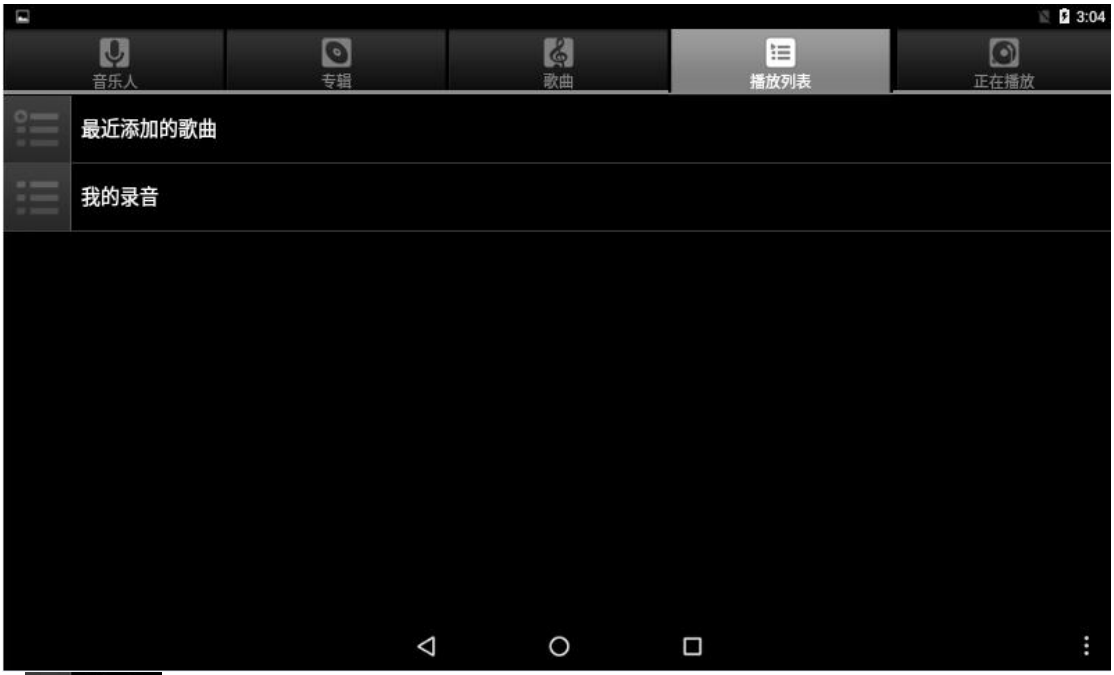
点击“ ”开始录音：



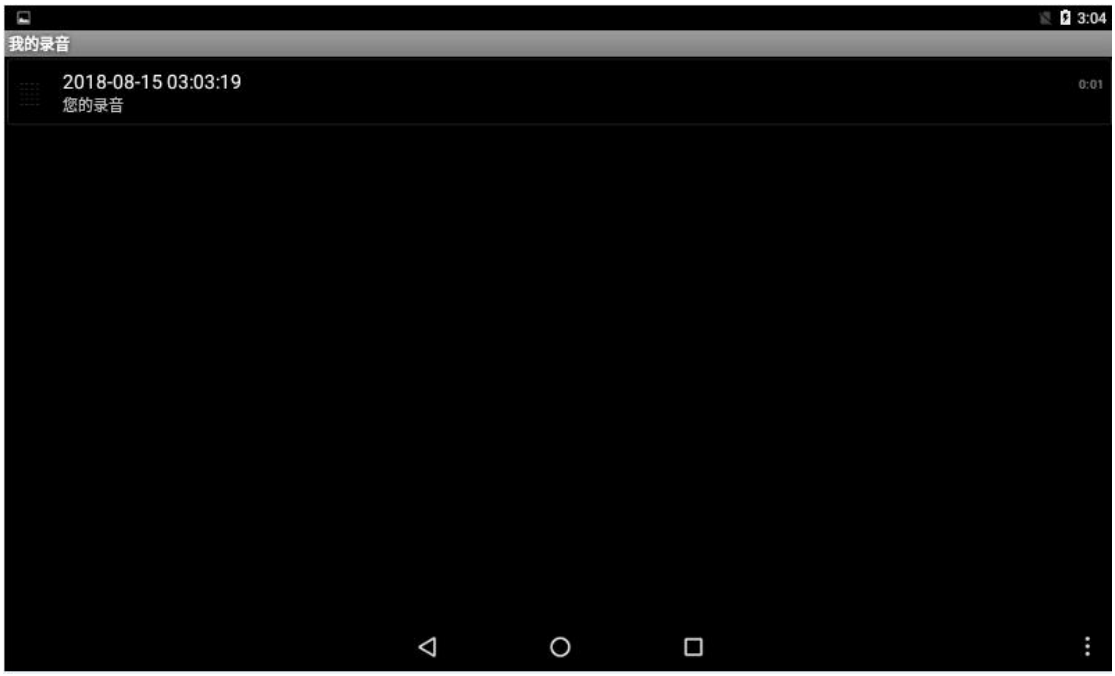
按下“ ”结束录音：

选择 “完成”完成录音，选择“放弃”放弃录音。

录音完成后在音乐播放程序内播放录音内容：



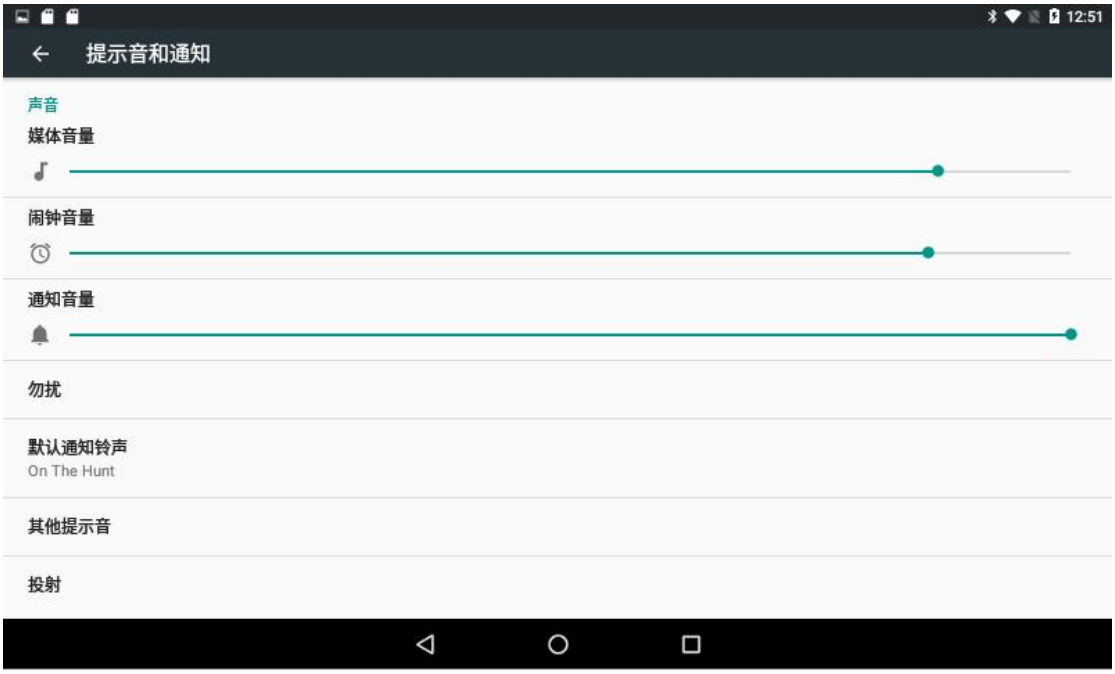
选择“我的录音”查看已完成录音。



根据时间选择您的录音进行播放。

2.11 调节音量

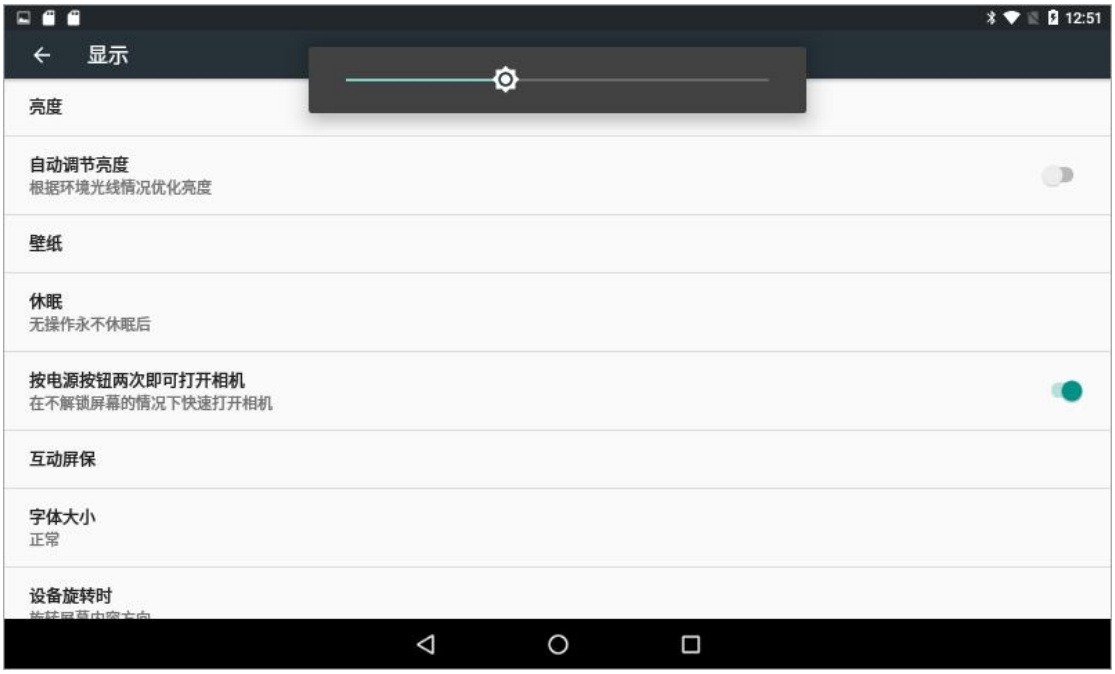
选择“ 提示音和通知 ”进入音量设置界面：



2.12 背光控制

选择设置应用，进入设置界面：

选择“ 显示 ”进入显示设置，选择“ 亮度 ”，出现亮度调节滑块，调节亮度。由于飞凌提供的开发板没有光感，所以这里的自动亮度调节没有起到作用。



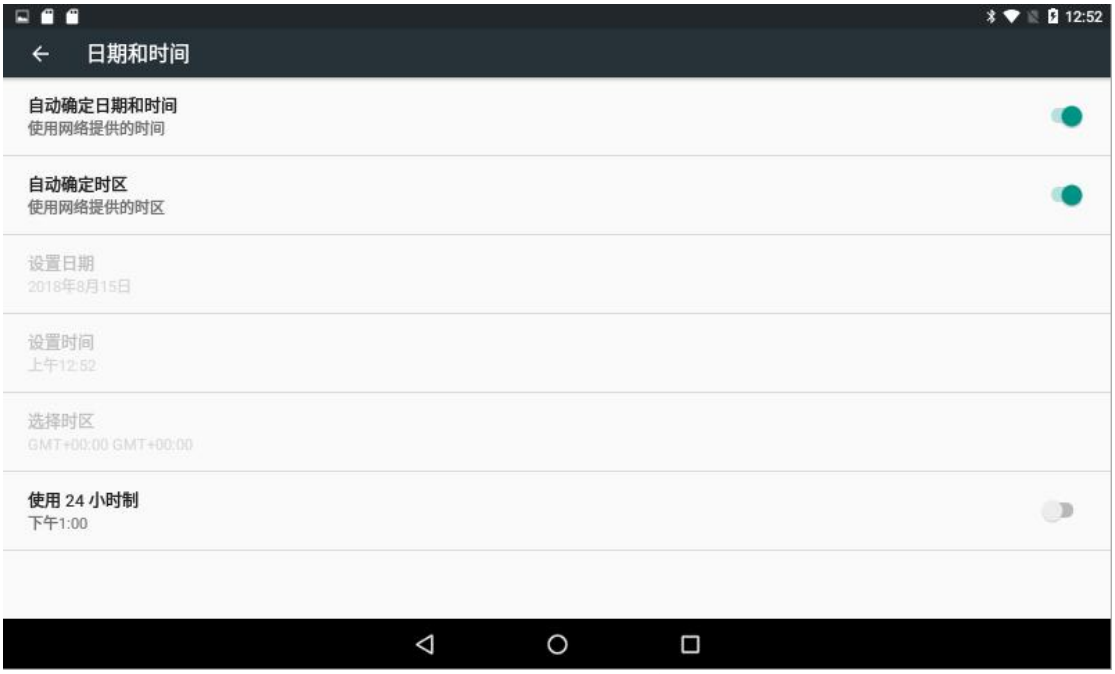
2.13 设置时间（RTC）

选择设置应用，进入设置界面：

选择“ 日期和时间 ”，在这里可以更改日期和时间，并且在您断电之后时间仍可同步更新。

注意：

此步测试一定要先去掉“ 自动确定日期和时间 使用网络提供的时间 ”以及“ 自动确定时区 使用网络提供的时区 ”两部分的勾选，否则无法准确测试 RTC 功能



点击“设置日期”和“设置时间”设置好之后，就可以给板子断电再上电，再次进入时间设置界面，就会看到时间已经同步更新了。

2.14 以太网测试

注：

1.开发板支持千兆以太网；

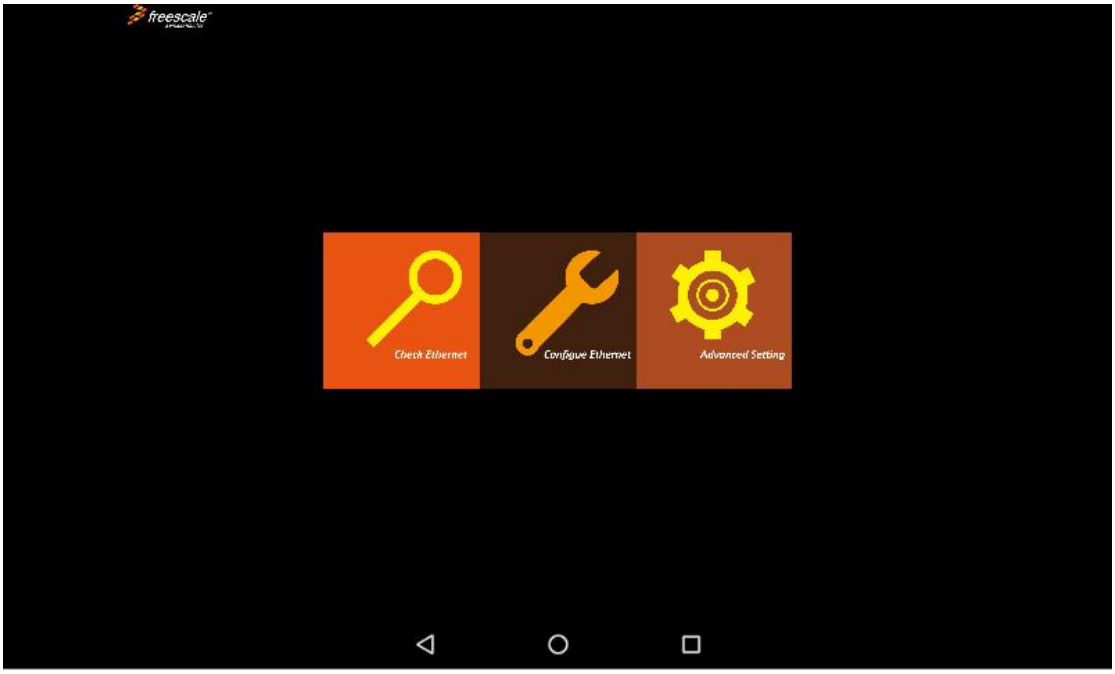
2.打开以太网之后，wifi 功能将不能使用。

以太网测试方法如下：

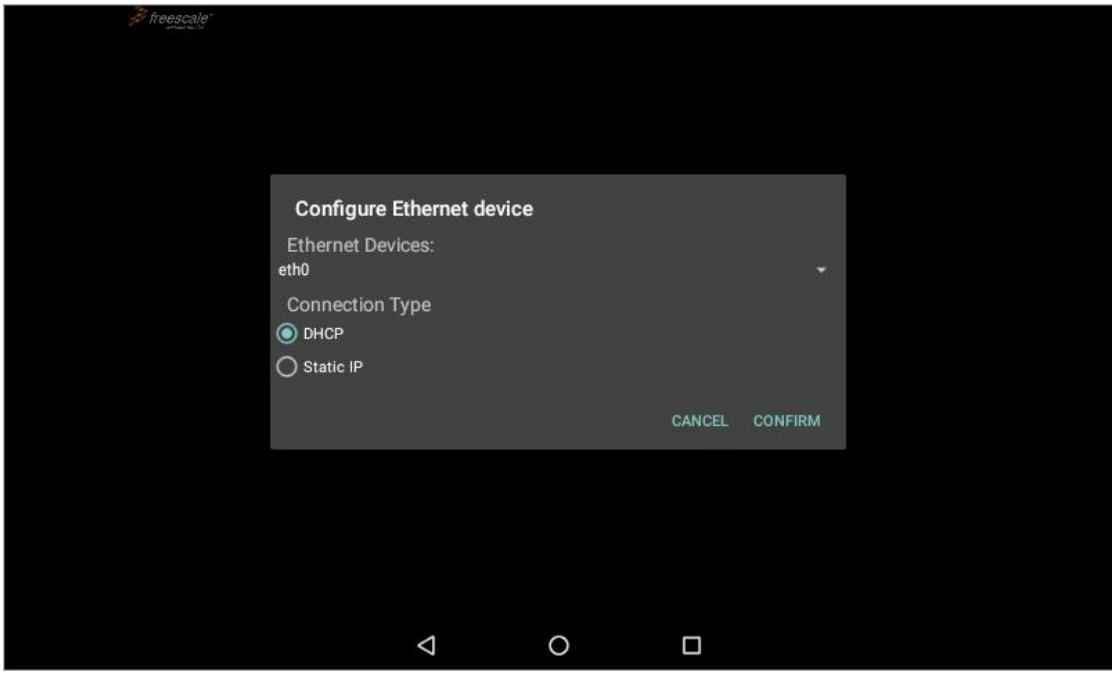
首先确保没有插着 4G 模块且关闭 wifi 网络。

准备路由器一个，网线一根，可连接外网的网线接口。

1. 插入网线后，选择 “ ”,如下图：



这里点击 Configue Ethernet，可以弹出以太网配置界面：



该菜单可以选择以太网设备 eth0，连接方式有 DHCP、Static IP，如果选择为 Static IP ,那么 IP 地址，子网掩码，DNS 等需要配置为用户所处环境以太网参数,如:

IP：192.168.1.2

子网掩码：255.255.255.0

DNS 地址：8.8.8.8，网关地址：192.168.1.1

配置完成后点击 confirm，保存设置。

2. 如果需要代理上网，点击 Advanced Setting,可弹出代理配置界面：

2.15 WiFi

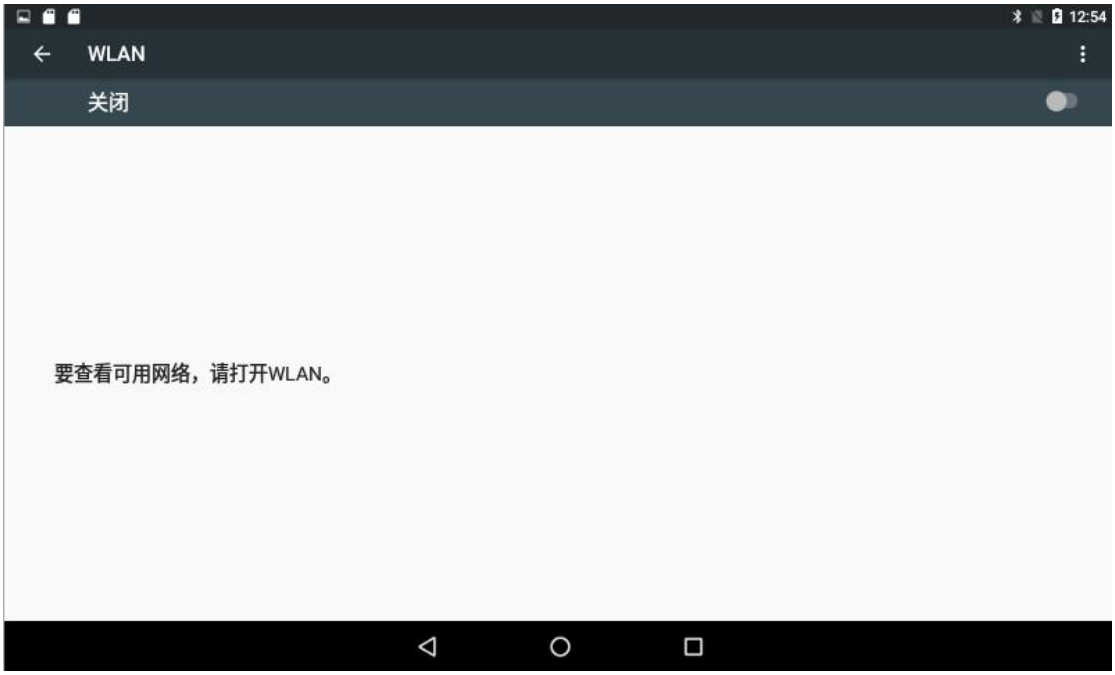
2.15.1 WiFi 功能测试

WiFi 天线如下图：

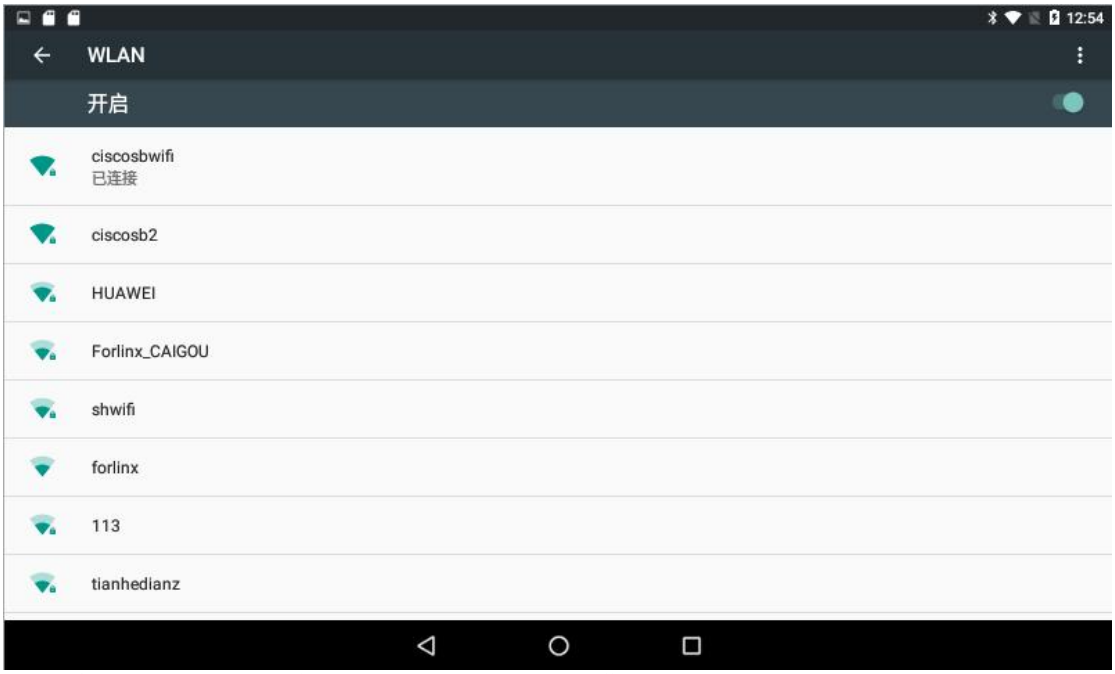
注意：测试 WiFi 时，将有线网络拔掉。

WiFi 测试使用 wifi&Bluetooth 一体模块，选择设置，界面如下：

选择“ WLAN ” 进入 WiFi 配置界面：



点击“ 关闭 ”使 wifi 进入开启状态：



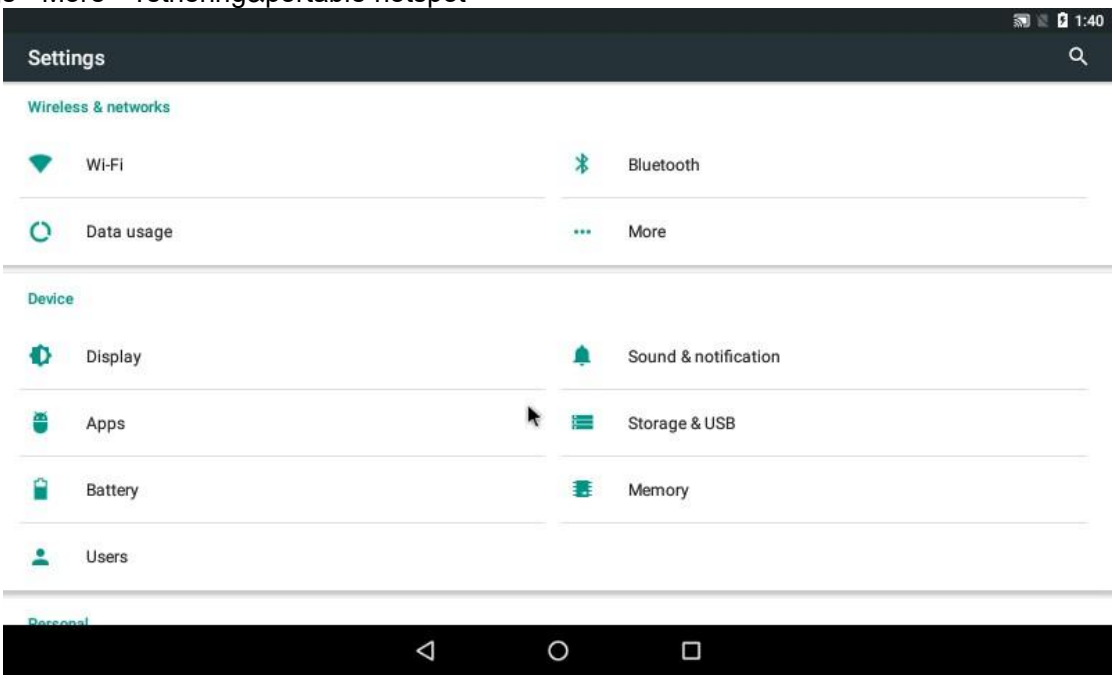
当 WiFi 搜索到可用 AP 后，会以列表的形式显示到屏幕。

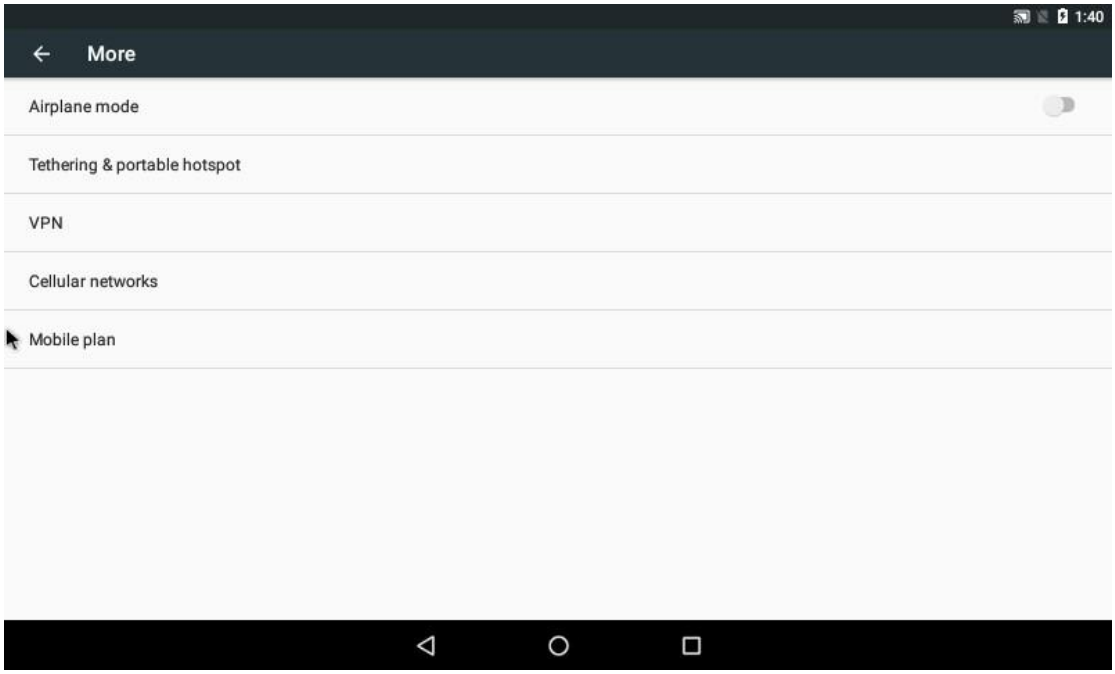
点击已知的 WiFi AP 进行连接，输入密码进行连接，连接成功后，就可以使用浏览器等网络应用。

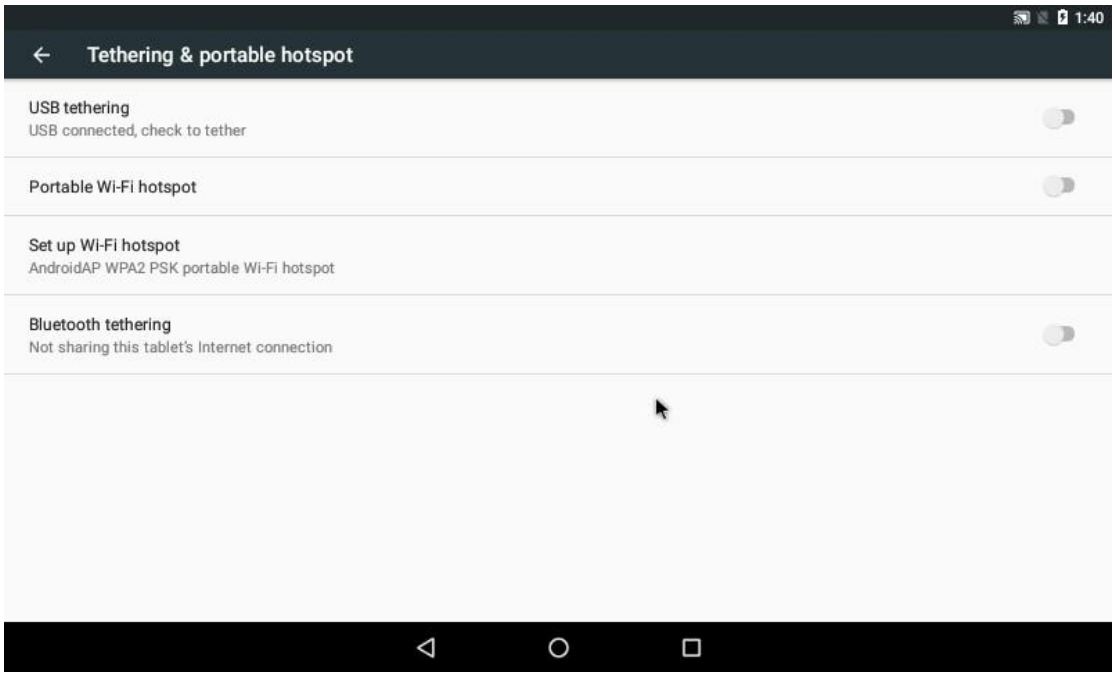
对于已经连接过的 WiFi AP，密码默认已经记录，在 Settings 界面可以直接打开“打开/关闭”开关，Android 会自动进行连接。

2.15.2 WiFi 热点测试

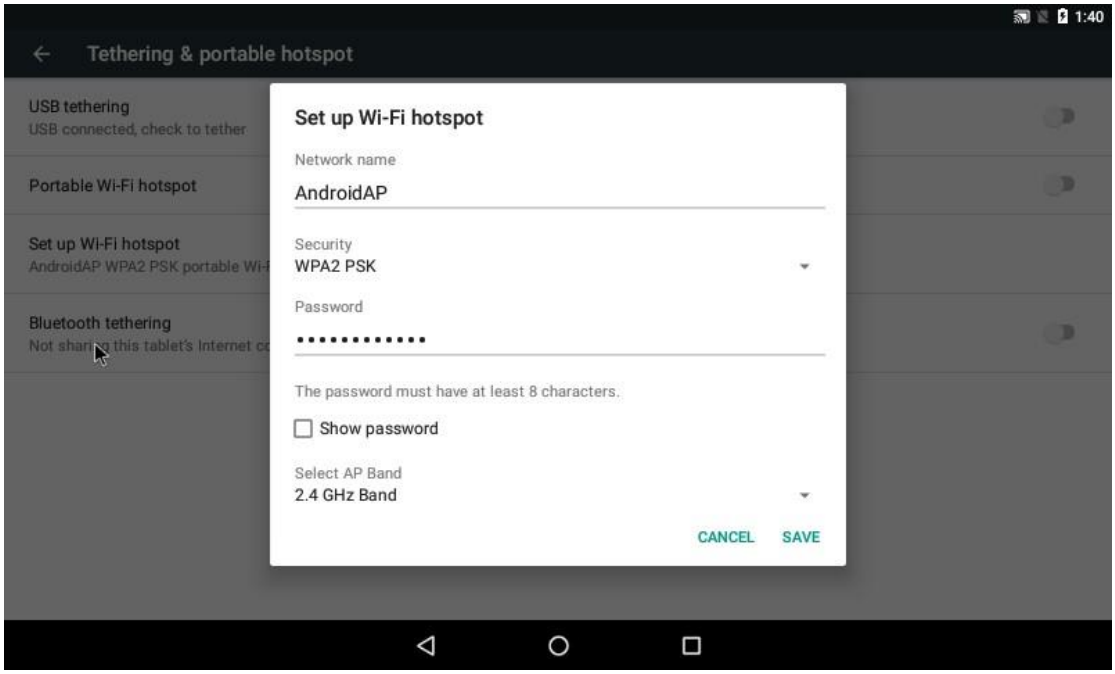
FCU 通过 4G 或以太网连接网络时，可通过 wifi 热点分享网络，可以通过 Settings->More->Tethering&portable hotspot



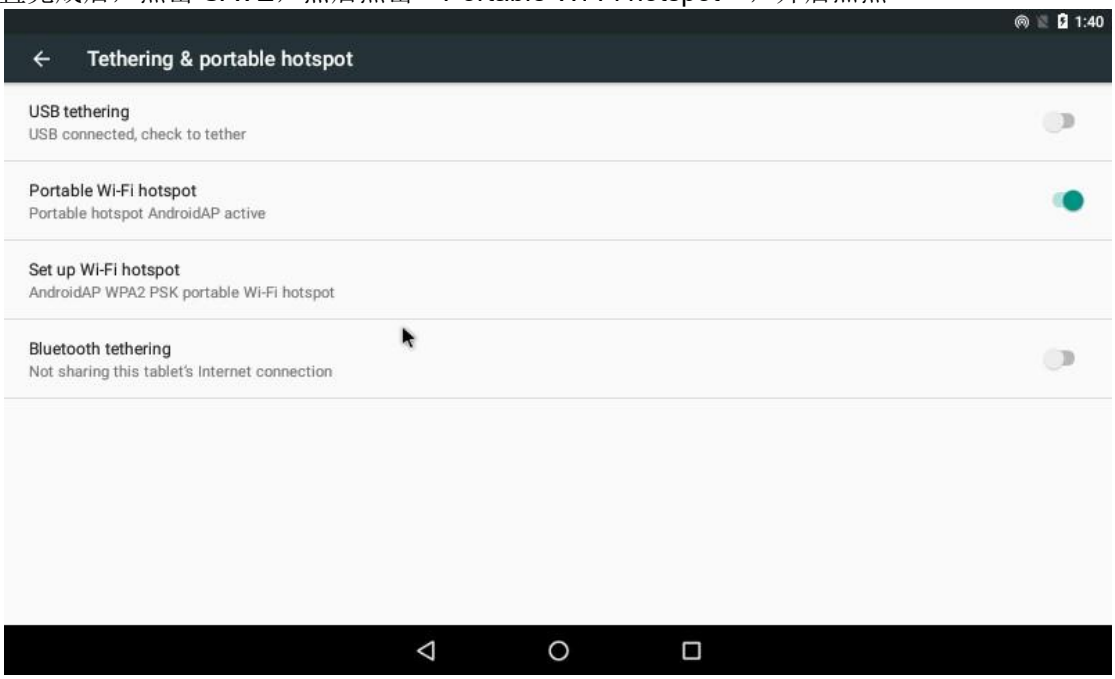




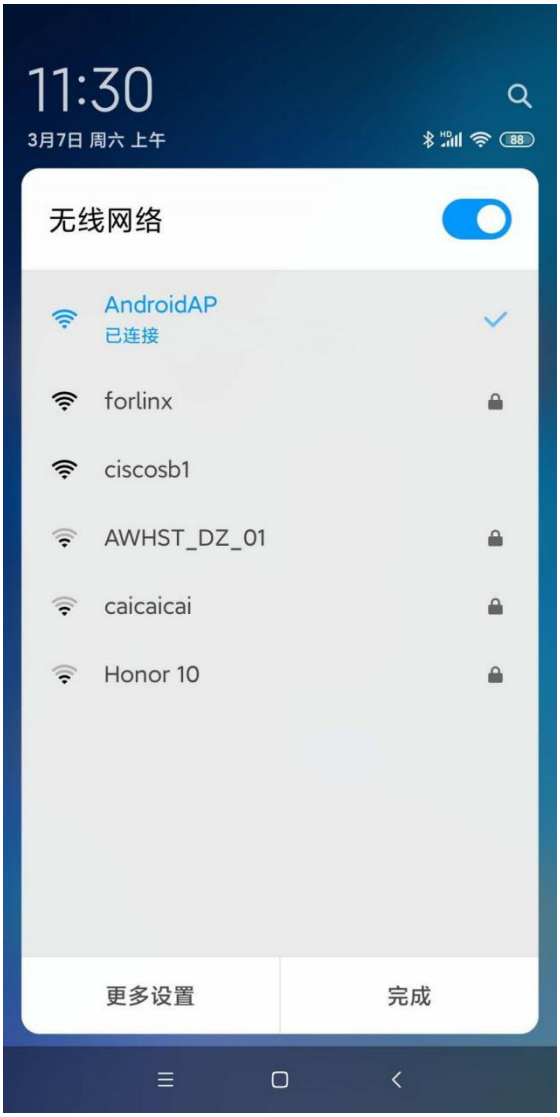
然后设置热点“Set up Wi-Fi hotspot”



设置完成后，点击 SAVE，然后点击“Portable Wi-Fi hotspot”，开启热点



手机搜索热点，连接后能正常使用。



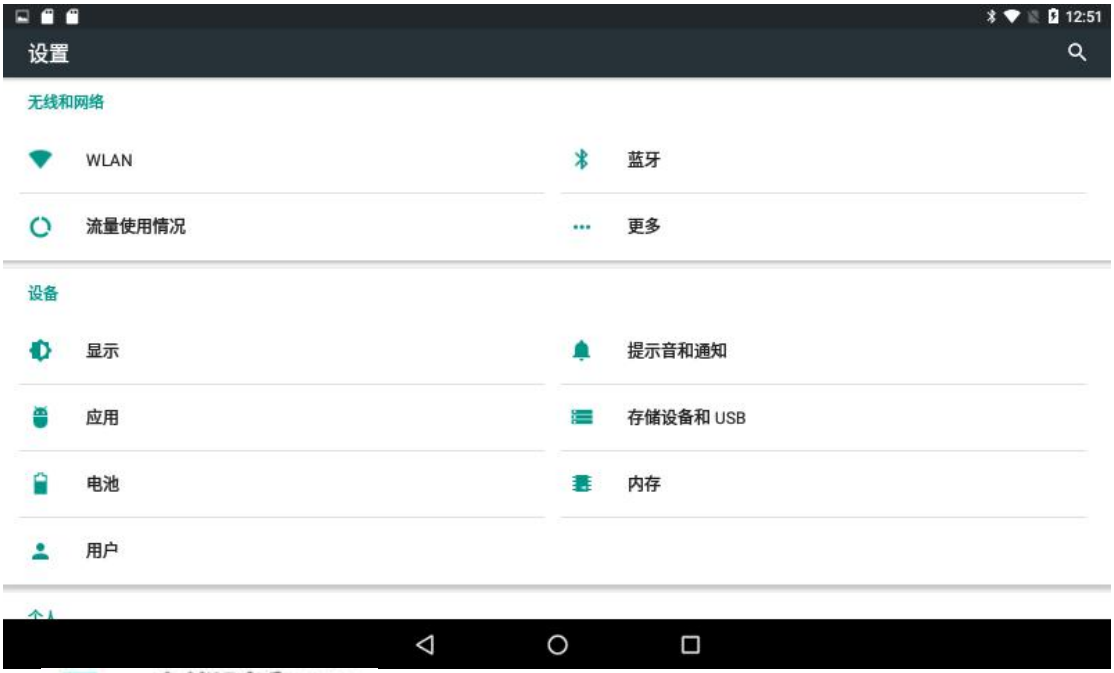
2.16 Android USB 设备测试

系统运行之后，在 USB host 上插入 USB 鼠标，您就会在界面内看到鼠标光标“  ”，您可以通过鼠标操作 Android 系统。

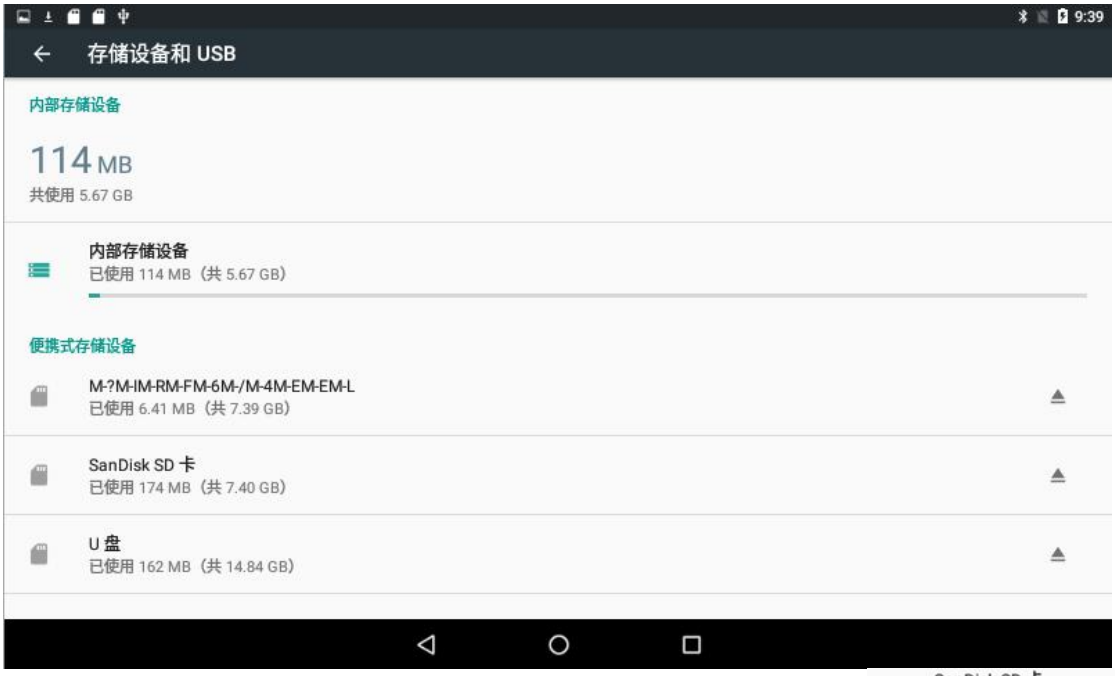
2.17 TF 卡/USB 存储测试

将 TF 卡或 USB 存储设备插入。

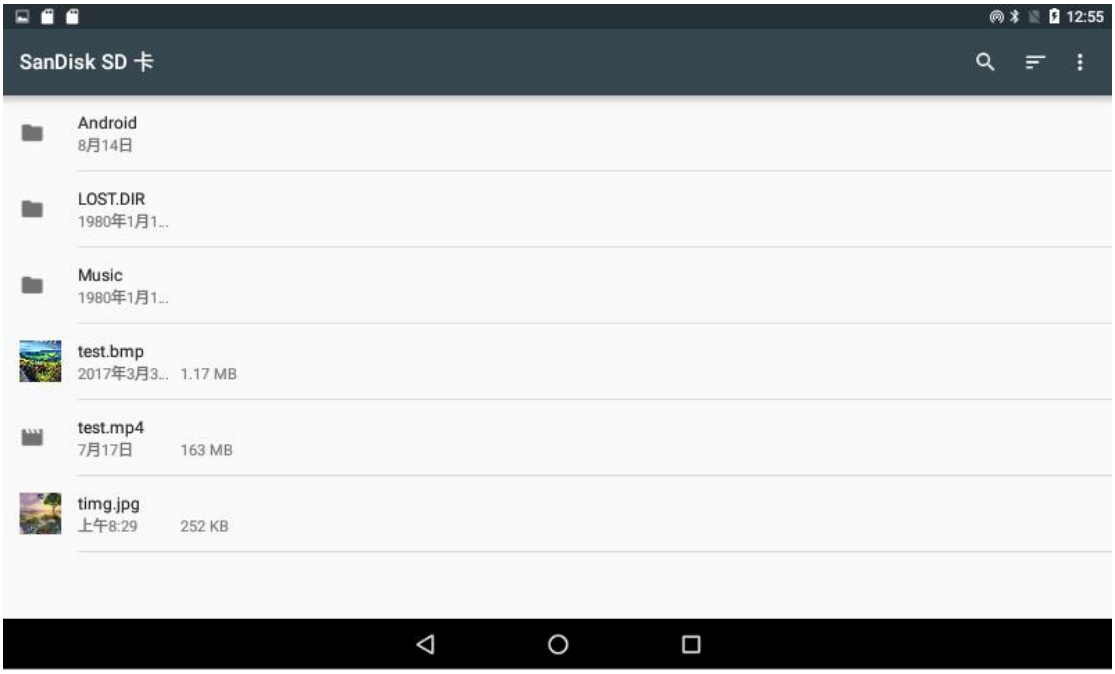
选择设置，出现如下画面：



选择“ 存储设备和 USB ”，查看存储设备：



该界面会将已连接的存储设备显示到当前界面，选择要查看的设备，如“ SanDisk SD 卡 已使用 174 MB（共 7.40 GB）”

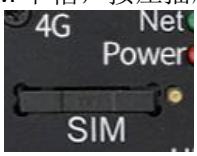


2.18 Android 4G 拨号上网测试

开机之后，默认 4G 模块已供电，驱动已经加载。

4G 模块配备两支天线，其中 M 为主天线，A 为副天线，主天线是必须的，副天线是用来增强接收效果的，并非必须，但仍建议接上。如下图：

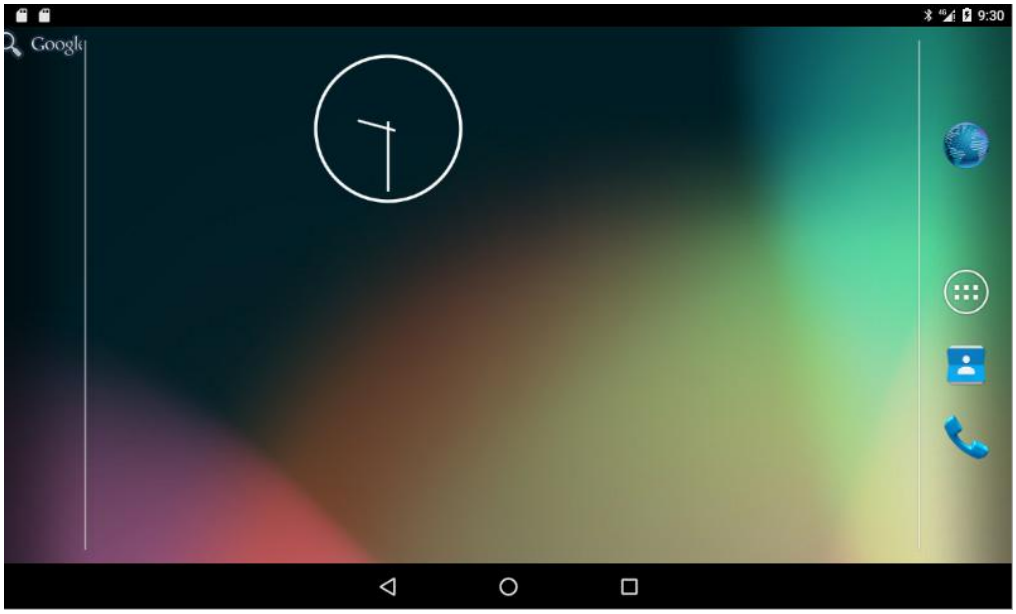
4G 模块 SIM 卡使用标准抽屉式 mini SIM 卡槽，按压抽屉右边的黄色按钮可弹出抽屉，如下图：



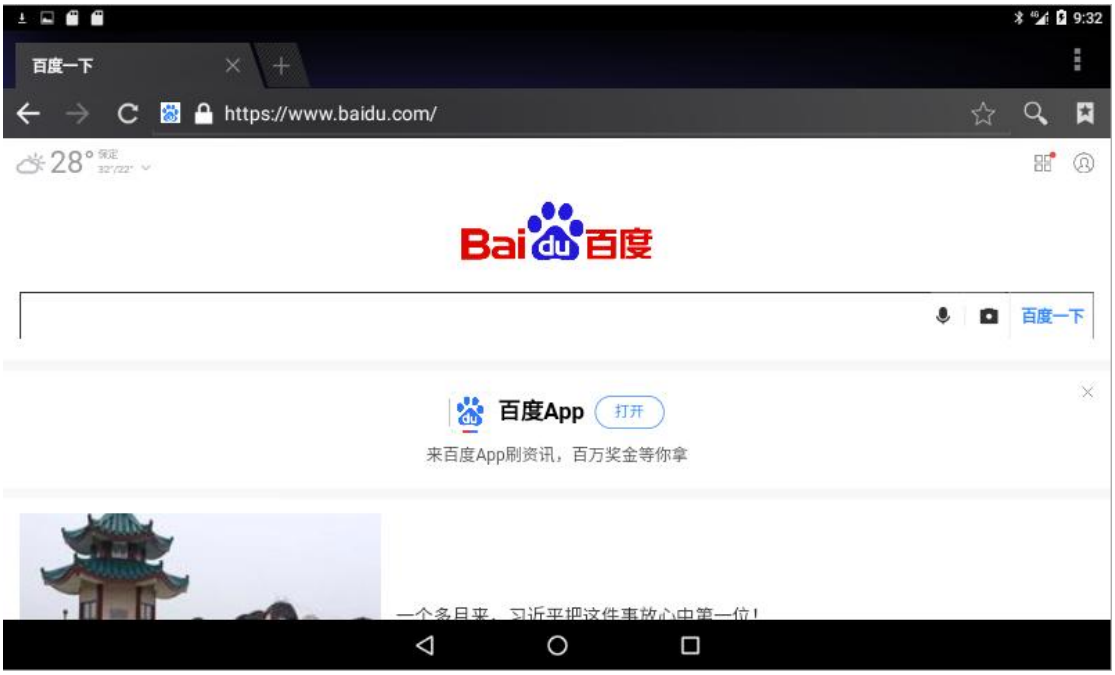
FCU1201 使用 ME909S4G 模块上网，支持中国移动与中国联通 4G、3G、2G。

系统启动后，可以看到已经有 4G 的信号 

。



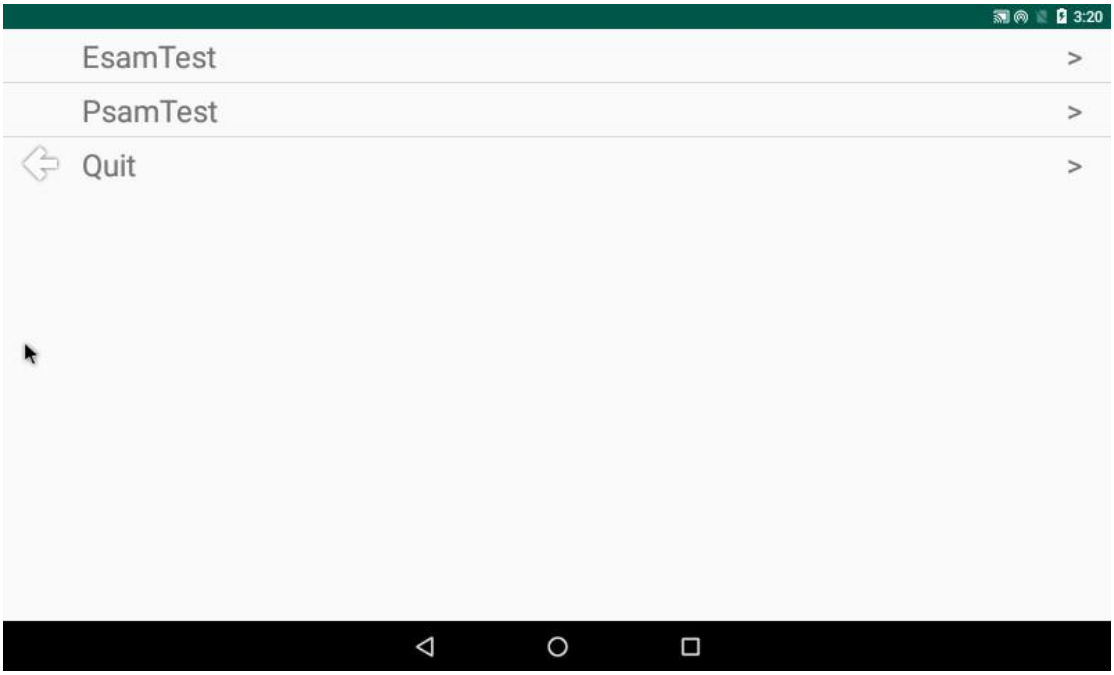
打开浏览器，能够浏览网页。



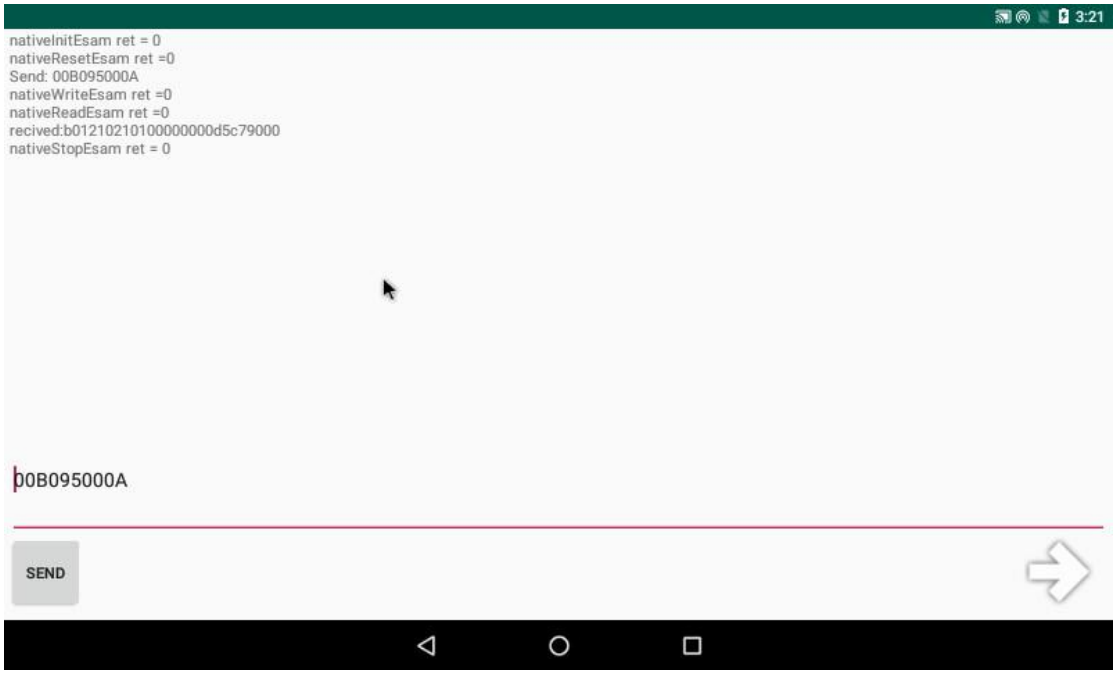
2.19 ESAM、PSAM 测试

ESAM 模块在机壳内，采用 DIP-8 封装座子，PSAM 采用标准抽屉式 mini SIM 卡槽，按压抽屉右边的黄色按钮可弹出抽屉，如下图：

在图形界面中，选择 ESAM PSAM Test 页面



选择 ESAM 或 PSAM， 点击 Test，在文本框内输出读取的结果。



2.20 显示

2.20.1 LVDS 接口测试

关闭电源，将 LCD 屏幕接至 LVDS 接口，上电即可正常显示，触摸屏可以使用。现阶段支持 LMT070DICFWD-AKA 液晶显示器。

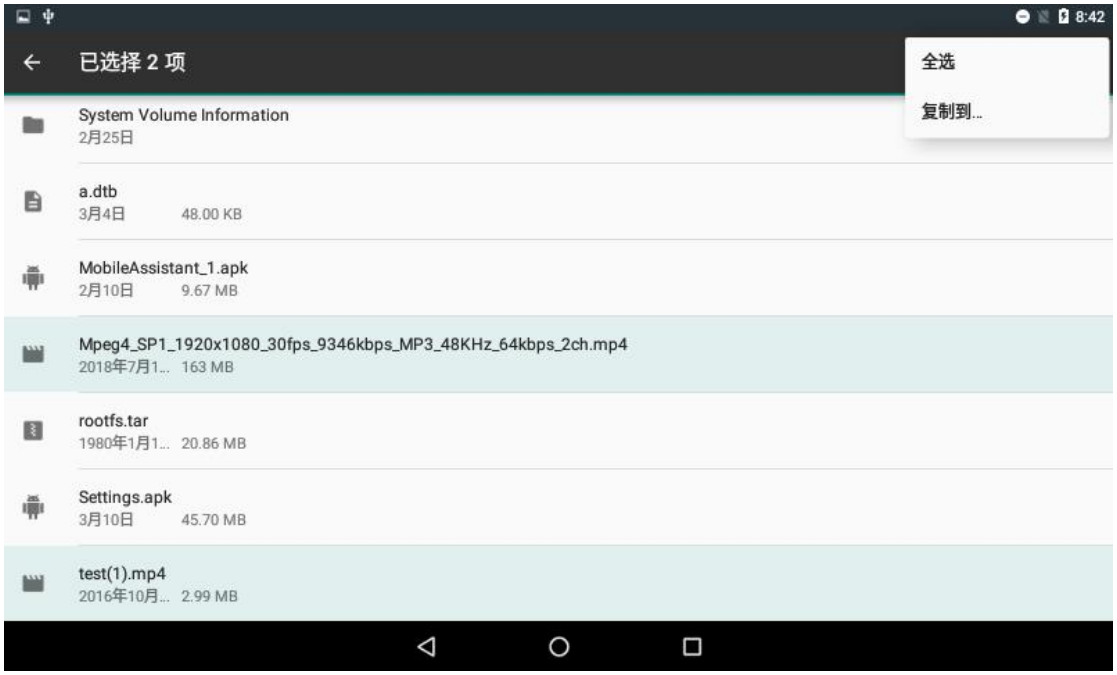
2.20.2 HDMI 接口测试

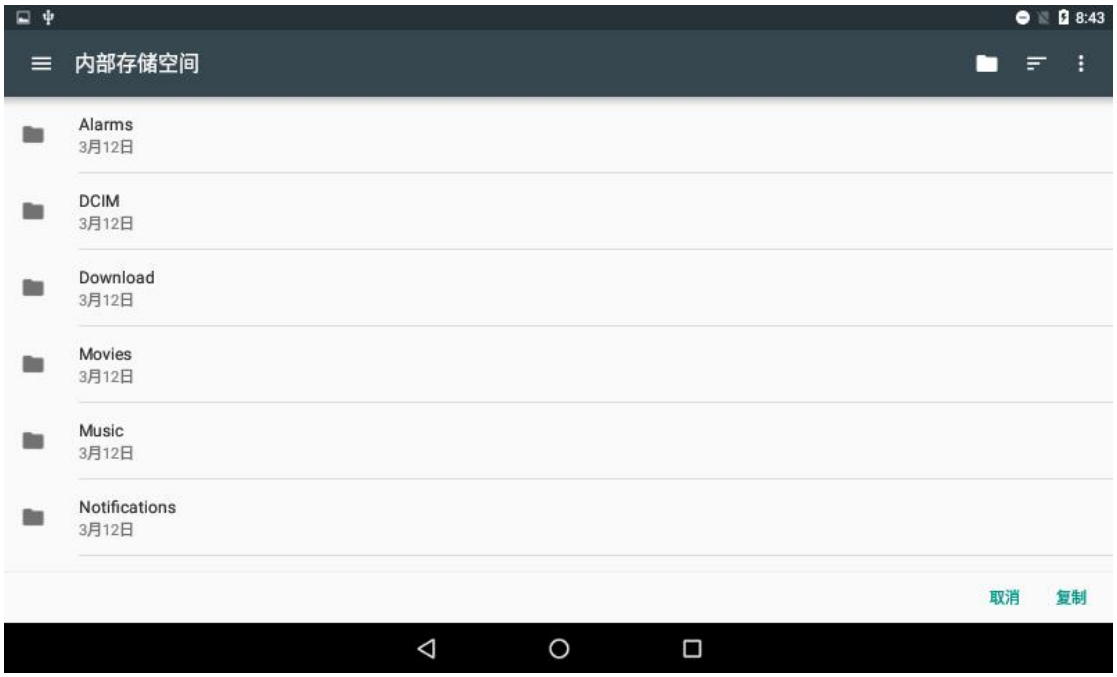
本机采用标准 Mini HDMI 接口座，如下图：

2.20.3 双屏显示测试

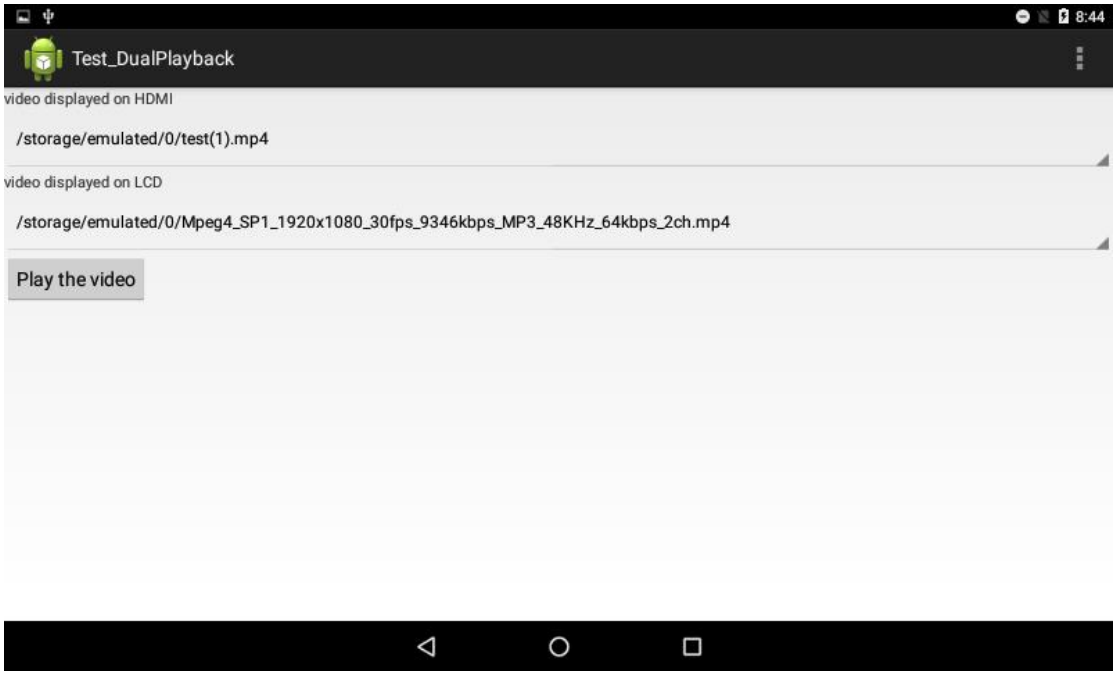
该测试在 2 个显示屏上播放不同的视频，测试前需要将 2 个 MP4 格式的视频复制到内部存储的根目录下。

在设置->存储设备和 USB 选择你出入的 U 盘或者 SD 卡，选择 mp4 文件复制到内部存储





然后运行测试程序 Test_DualPlayback，点击按钮“Play the video”（如果没放置视频到内部存储此按钮会变不可用且文字变为 have no video）， 此时在 2 个屏上播放不同的视频。



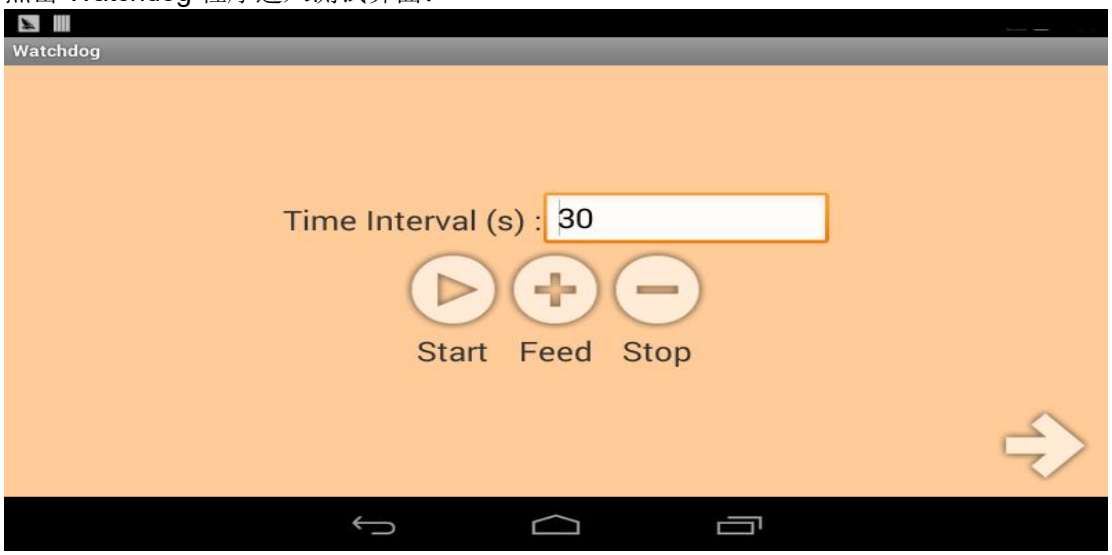
注：第一个视频对应 LVDS 显示，第二个视频对应 HDMI 显示。

2.21 系统复位

按 Reset 按钮，系统立即重启。同时调试串口将打印系统启动时的信息。
Reset 按钮如下图：

2. 22 Watchdog 测试

点击 Watchdog 程序进入测试界面：



首先 timeout interval(s)里写入看门狗周期，单位为秒，可输入范围为 1-30；
点击 Start 开始倒计时，当计时到 0 后重启；
点击 Feed 会执行一次喂狗，倒计时重新开始计时，点击 Stop 停止看门狗。

第三章 Android 编译环境的搭建

3.1 安装 Ubuntu 14.04.5 x64bit 及编译环境

在这里建议用户使用 Ubuntu 虚拟机进行编译，ubuntu 虚拟机的安装与配置见附录 1，我们提供 Android6.0 的代码在 Ubuntu14.04 64 位系统下编译测试通过。

另外，本公司的网盘中提供了装有库文件及编译器的虚拟机，可作为参考。

注：推荐电脑配置处理器：Core(TM) i7　　内存：8G 以上;

3.2 安装编译 Android 系统所需要的库

Android 系统的编译需要安装一些工具包。本节操作前必须确保您的计算机或虚拟机能正常连接互联网，如您在安装中出现网络断开连接请再按照以下步骤进行安装。

1. 安装编译 Android 必要的包

```
# sudo apt-get install bison g++-multilib lzop libxml2-utils
```

2. 安装过程中出现如下提示需作出对应操作

```
Note! This command requi
on your host.
Press return to continue
```

此提示按下回车

```
Do you want to continue [Y/n]? Y
```

此提示输入“Y”后按下回车

3. 安装 JDK：

JDK 是整个 java 开发的核心，它包含了 JAVA 的运行环境，JAVA 工具和 JAVA 基础的类库。

```
#sudo apt-get install openjdk-7-jdk
```

```
The following packages will be upgraded:
  tzdata
1 upgraded, 26 newly installed, 0 to remove and 408 not upgraded.
Need to get 61.8 MB of archives.
After this operation, 104 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
WARNING: The following packages cannot be authenticated!
  tzdata tzdata-java openjdk-7-jre-headless openjdk-7-jre x11proto-core-dev
  xtrans-dev openjdk-7-jdk
Install these packages without verification? [y/N] y
```

查看 java 版本(需要 1.7 版本)

```
#java –version
```

```
forlinx@ubuntu:~$ java -version
java version "1.7.0_181"
OpenJDK Runtime Environment (IcedTea 2.6.14) (7u181-2.6.14-0ubuntu0.2)
OpenJDK 64-Bit Server VM (build 24.181-b01, mixed mode)
```

释：通过此步可以查看到当前的 java 版本，若出现无 java 命令的情况，或 java 版本显示不正常，请检查命令是否输入错误，是否是在 root 下安装的 JDK 后在普通用户下查看的 java 版本。此步未成功请勿进行编译操作。

3.3 Android 系统的编译

3.3.1 编译前的准备

请确认当前系统 swap 分区大小，若 swap 分区不足会造成编译 Android 源码失败，推荐 4G。

查看 swap 分区情况：

```
#cat /proc/swaps
```

在此提供一种通过创建 swap 文件的方式来增加 swap 分区大小的操作方法，可作为参考：

```
#sudo fallocate –l 4G /swapfile
#sudo chmod 600 /swapfile
#sudo mkswap /swapfile
#sudo swapon /swapfile
#sudo vim /etc/fstab
```

在/etc/fstab 文件最后添加如下内容：/swapfile none swap sw 0 0

android_6.0.1_2.1.0 系**统的源码包** android_6.0.1_2.1.0.tar.gz 位于 FCU1201 嵌入式控制单元_Android6.0_用户资料\Android\源码\。将它拷贝到 ubuntu 文件夹/work/forlinx 下;

注：

防止编译出现不必要的错误，请您也把代码解压到上面提到的目录下。

首先解压 Android 源码，解压命令如下：

```
#cat  android_6.0.1_2.1.0.tar.bz2xa*  >  android_6.0.1_2.1.0.tar.bz2
#cd  /work/forlinx
#tar  xvf  android_6.0.1_2.1.0.tar.bz2
```

说明:

Android 文件系统位置 ：android_6.0.1_2.1.0

Kernel 位置：android_6.0.1_2.1.0/kernel_imx

Uboot 位置（Uboot-2016）：android_6.0.1_2.1.0/bootable/bootloader/uboot-imx

3.3.2 编译 Android 文件系统

注： 此节的整体编译文件系统源码会生成 uboot，kernel 及文件系统映像，不推荐用户单独编译内核及 uboot，不提供单独编译的方法。

编译 android6.0 进入该目录，执行以下命令：

```
#cd  /work/forlinx/android_6.0.1_2.1.0
# source build/envsetup.sh
#lunch sabresd_6dq-user
make -j8
```

注：

1. Android 的编译过程需要耗费几个小时的时间，时间长短视电脑配置而定。

2. 编译生成的镜像全部在 android_6.0.1_2.1.0/out/target/product/sabresd_6dq/下生成，分别是 boot-imx6q-c.img（四核）、boot-imx6dl-c.img（双核）、recovery-imx6q-c.img（四核）、recovery-imx6dl-c.img（双核）、system.img、u-boot-imx6q-c.imx（四核 1GDDR）、u-boot-imx6dl-c.imx（双核 1GDDR）u-boot-imx6q-c-2g.imx（四核 2GDDR）、u-boot-imx6dl-c-2g.imx（双核 2GDDR）。

3.4 eMMC 存储器分区表

下面表格是Android操作系统的eMMC存储器 分区信息：

分区类型	名称	偏移	大小	文件系统	内容
Boot0	Bootloader（启动引导分区）	1KB	1MB	N/A	bootloader
N/A	Boot args	768K	8K	N/A	Boot args
主分区 1	Boot（启动分区）	8MB	16MB	boot.img (kernel+ramdisk)	boot.img
主分区 2	Recovery（恢复分区）	Follow Boot	16MB	Boot.img (kernel+ramdisk)	recovery.img
主分区 4	Data（数据分区）	Follow Misc	剩余所有容量	ext4.挂载在/data	用来存储系统应用和内部媒体分区（目录/mnt/sdcard/）
逻辑分区 5	System（系统分区）	Follow Recovery	800MB	ext4.挂载在 /system	Android 系统文件在 /system 目录下
逻辑分区 6	Cache（缓存分区）	Follow System	512MB	ext4.挂载在/cache	Android 用来放 OTA 升级 image 的分区
逻辑分区 7	Device（设备分区）	Follow Cache	8MB	ext4.挂载在 /vendor	用来存储 MAC 地址文件
逻辑分区 8	Misc	Follow Device	6M	N/A	为了恢复保存 bootloader 信息，保留
逻辑分区 9	Datafooter	Follow Misc	2M	N/A	
逻辑分区 10	Logo	Follow Datafooter	10M	vFat	保存 log 图片

第四章 系统固件更新

本产品更新固件方法有：OTG 更新固件，TF 卡更新固件。

4.1 烧写 Android6.0 镜像

➢ **烧写前的准备**

1、把 micro usb 数据线连接到开发板的 otg 接口，数据线的另一端连接到 pc 的 usb 接口。

2、 打开 Android\工具\USB 烧写工具\mfgtools-linux4.1.15&android6.0 打开 mfgtool2-qt-OKMX6-C-emmc.vbs 。

🗁 Android\工具\USB 烧写工具\mfgtools-linux4.1.15&android6.0。

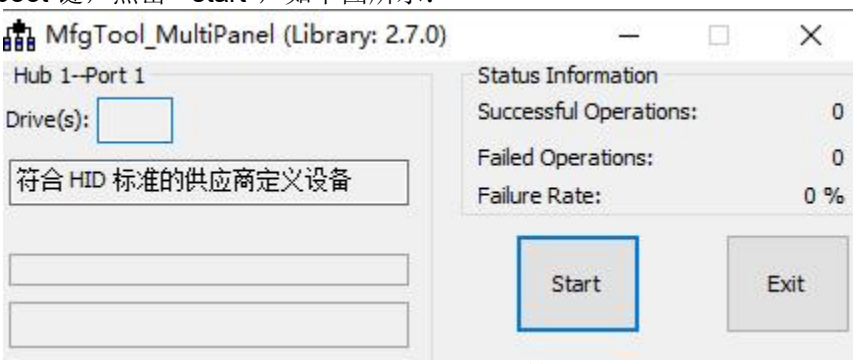
➢ **烧写 Android6.0**

1、 烧写时，确保其烧写工具目录\Profiles\Linux\OS Firmware\files\android\okmx6 下有 u-boot-imx6dl-c.imx, u-boot-imx6q-c.imx, u-boot-imx6dl-c-2g.imx, u-boot-imx6q-c-2g.imx, boot-imx6dl-c.img, boot-imx6q-c.img, logo.bmp, recovery-imx6dl-c.img, recovery-imx6q-c.img, system.img 文件。

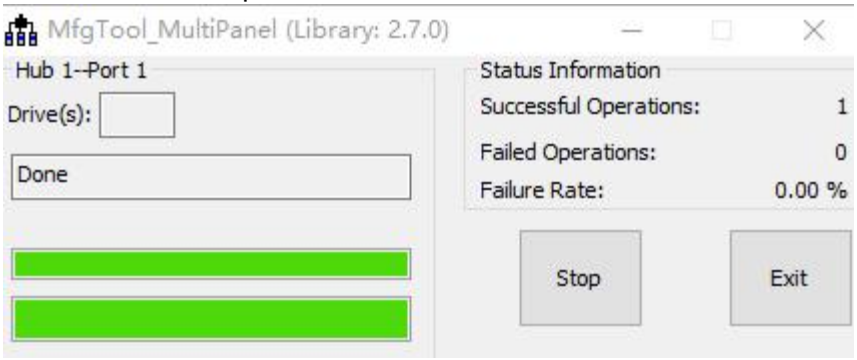
2、1GDDR 使用 mfgtool2-android-mx6x-c-sabresd-emmc.vbs 进行烧写，
2GDDR 使用 mfgtool2-android-mx6x-c-2g-sabresd-emmc.vbs 进行烧写。

3、给设备上电，按住 Boot 键的同时按下 reset 键，先松开 reset 按键后等烧写工具识别 HID 设备后可以松开 Boot 键。

4、首次升级过程中，会通过网络自动安装驱动，等安装完成后，烧写工具里出现设备的名称(HID-compliant device)后，抬起 boot 键，点击 "start"，如下图所示：



5、中间弹出格式化对话框，点击“取消”格式化选项，或者不管它，直到烧写完成，看见 DONE 之后烧写完成。如要退出烧写程序，先点击 stop，再点击 exit 退出即可。



6、烧写完成后，复位或者重新上电启动即可。

注意：

1. 多次烧写时，遇到烧写失败，需要关闭烧写软件再重新打开。
2. 调试串口是 DEBUG（UART1）口。

4.2 TF 卡更新固件

4.2.1 制作 TF 卡

如果处理器使用 i.MX6Quad/Dual Lite 则使用 TF 卡烧写工具 sdmaker.tar.bz2，将 sdmaker.tar.bz2

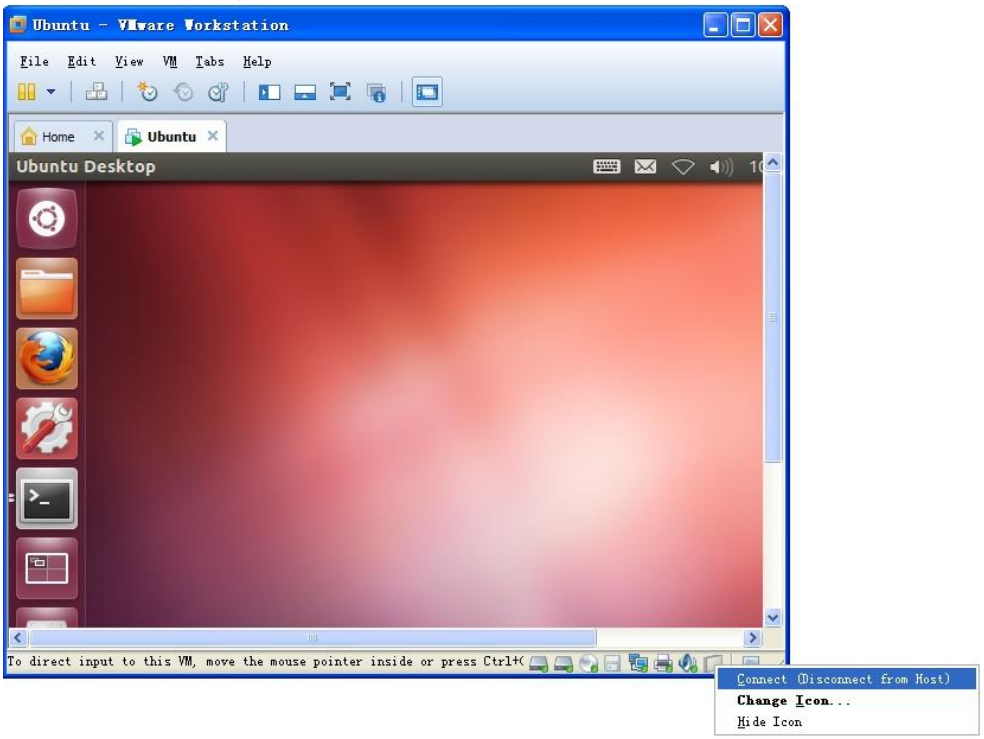
拷贝到到 ubuntu 系统的任一目录，假设为/home/forlinx/work；

🗁 Andriod 工具\TF 烧写工具。

步骤 1：解压 sdmaker.tar.bz2，命令如下。

```
cd /home/forlinx/work/
tar xvf sdmaker.tar.bz2
```

步骤2：使用 USB 读卡器把TF卡插入到电脑的USB 端口（VMware **虚拟机用户如果U盘没有被虚拟机识别，可以使用如下方式将U盘连接到虚拟机**）。



步骤3：进入imx6dl-sd-linux-tools目录

```
cd /home/forlinx/work/sdmaker
```

createSdcard.sh添加可执行权限：

```
chmod u+x createSdcard.sh
```

执行制作TF卡脚本，脚本后跟参数imx6q或imx6dl，通过参数来区分是制作双核（imx6dl）还是四核（imx6q）烧写的TF卡，制作TF卡需要root权限，非root用户需要通过sudo来执行此命令。

如四核：

```
sudo ./createSdcard.sh imx6q
```

双核：

```
sudo ./createSdcard.sh imx6dl
```

执行上述命令后，终端会列出电脑的硬盘或优盘，对应选择自己的TF 卡，然后回车。

注意：判定自己的U盘是 sda/sdb/sdc 可以根据容量进行判断，比如自己的TF卡容量为16G，则其size为15558144 字节≈ 16G，建议用户执行此操作时不要同时插入多个优盘，以免混淆。

这里以我们的操作为例：

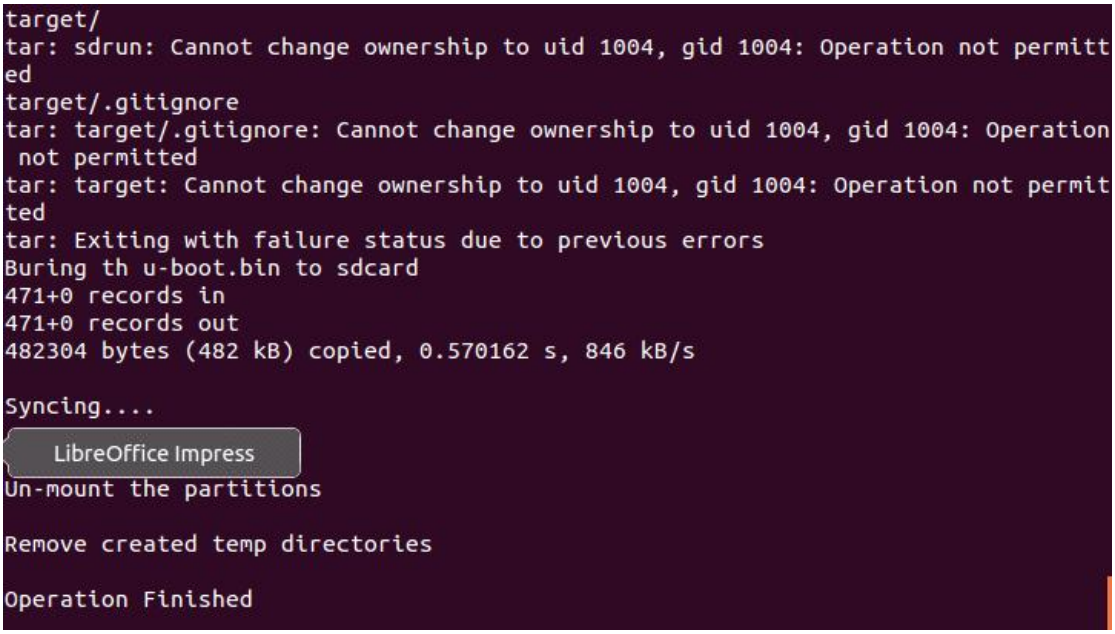
选择1，回车

```
##############################################################################

This script will create a bootable SD card from custom or pre-built binaries.

The script must be run with root permissions and from the bin directory of
the SDK

Example:
 $ sudo ./create-sdcard.sh

Formatting can be skipped if the SD card is already formatted and
partitioned properly.

##############################################################################

LibreOffice Impress                ite images to:

#  major   minor    size   name
1:   8       16   15558144 sdb

Enter Device Number: 1
```

格式化，选择y，回车，完成格式化。

```
#  major   minor    size   name
1:   8       16   15558144 sdb

Enter Device Number: 1


sdb was selected

Checking the device is unmounted
 unmounted /dev/sdb1

sdb1  sdb2   sdb3
15506432


##############################################################################

   Detected device has 1 partitions already

Ubuntu Software Center             llow the choice of 1 partitions

##############################################################################

Would you like to re-partition the drive anyways [y/n] : y
```

格式化完成，如图：



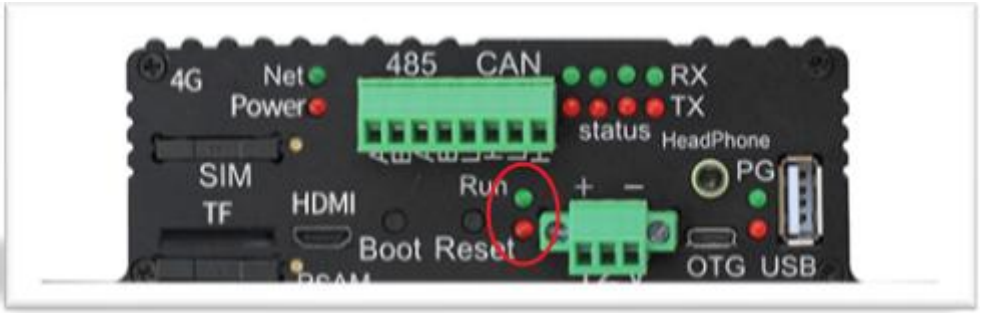
TF 卡制作完成后可以看到 boot 分区包含 sdrun 和 target 两个目录。sdrun 文件夹内容用于引导系统烧写，无需修改；target 目录内容会烧写到 flash 芯片。将需要烧写的镜像文件 u-boot-imx6dl-c.imx,u-boot-imx6q-c.imx, boot-imx6dl-c.img, boot-imx6q-c.img, logo.bmp, recovery-imx6dl-c.img, recovery-imx6q-c.img, system.img，复制到 target 目录，烧写 2GDDR 镜像需要将 u-boot-imx6dl-c-2g.imx, u-boot-imx6q-c-2g.imx 替换 u-boot-imx6q-c.imx, u-boot-imx6dl-c.imx 之后进行系统烧写。

4.2.2 TF 卡更新系统

将上一节中制作好的 TF 卡插入 FCU1201 的 TF 卡槽中，给设备上电，按住 boot 键的同时按下 reset 键，松开 reset 后再抬起 boot 键。TF 卡中新的固件会自动更新到 FCU1201 中。更新时间较长，可以从屏幕上看到更新过程。

FCU1201 重新上电或者复位。即可正常使用。

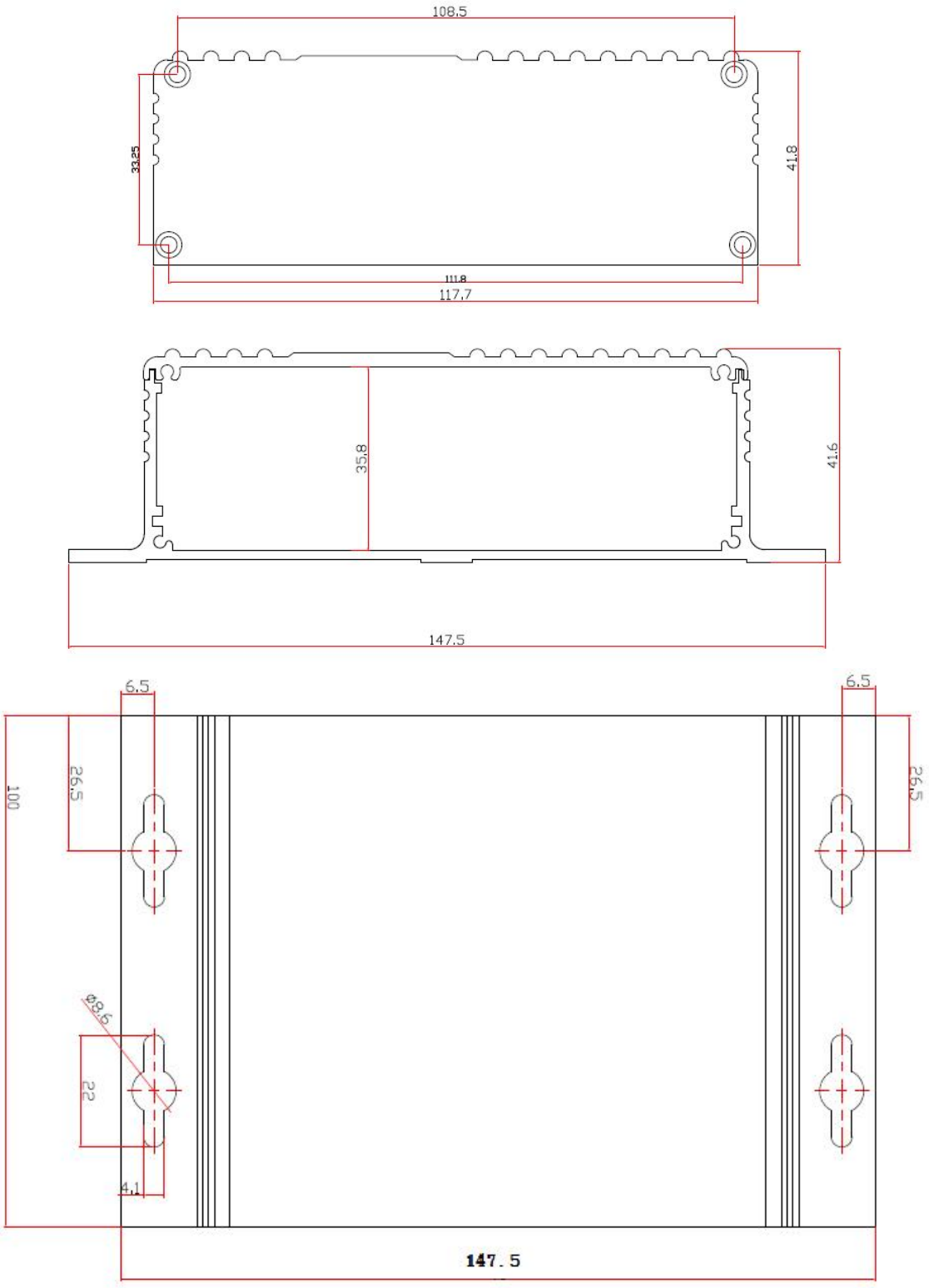
如果没有连接显示屏，也可以通过观察下图中的 2 个 led 判断。



烧写异常：2 个灯常亮。
烧写完成：2 个灯闪烁。

附录一 外壳尺寸图

下图为 FCU1201 外壳尺寸图单位 mm：

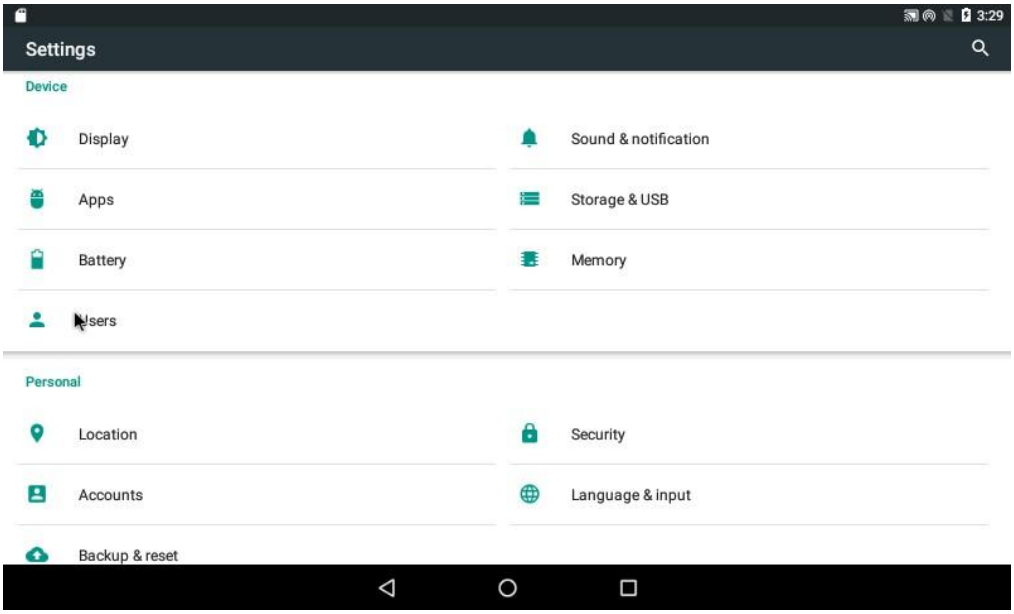


附录二 APK 安装

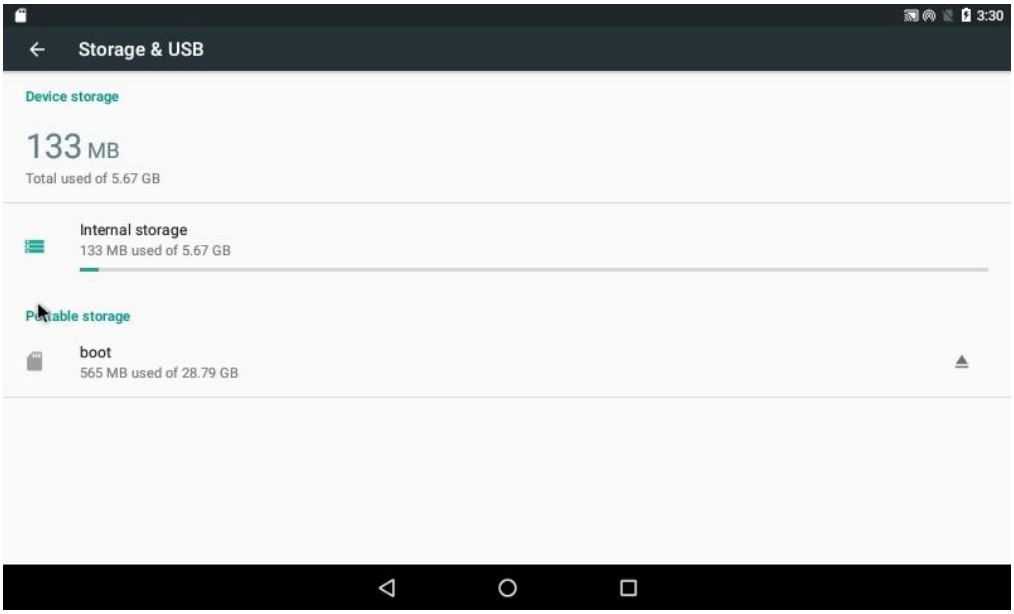
通过外部存储安装 APK 到系统内，可以通过 TF 卡安装，也可以通过 USB 存储（需要先将 APK 复制到内部存储，否则无权限安装）。

2.1 TF 卡安装：

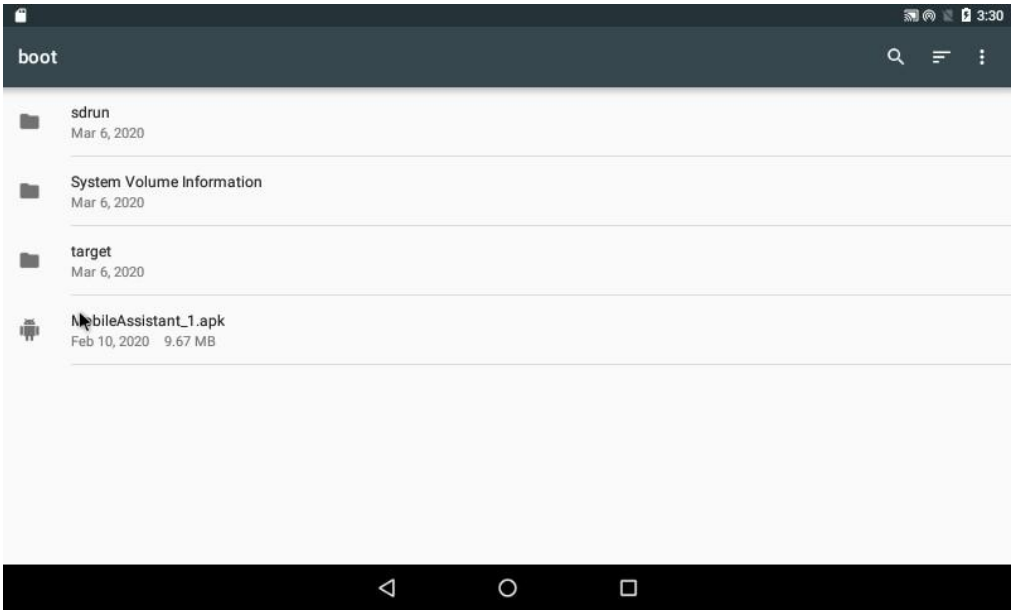
打开设置：



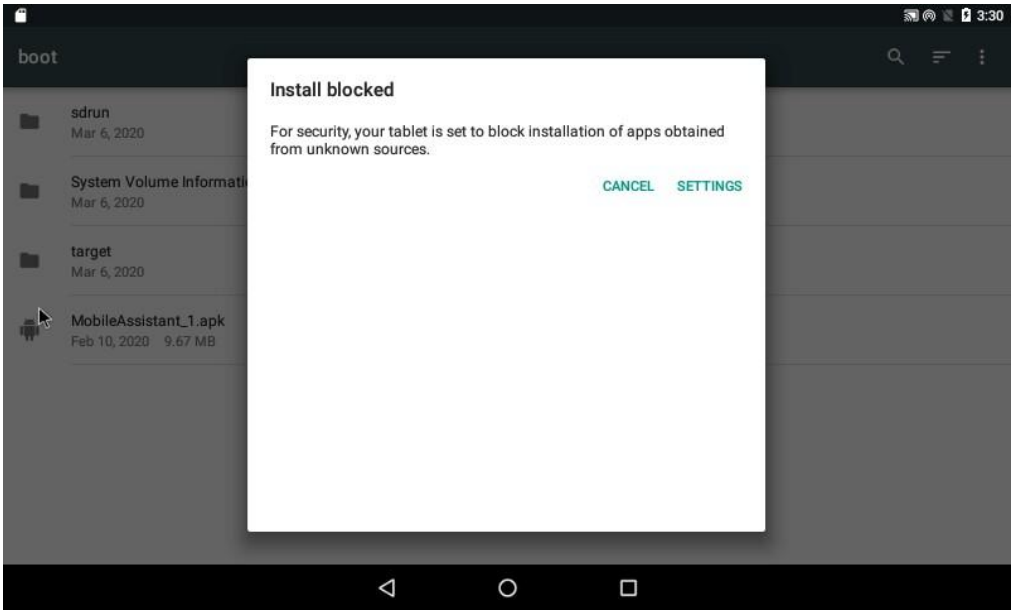
进入 Store&USB：



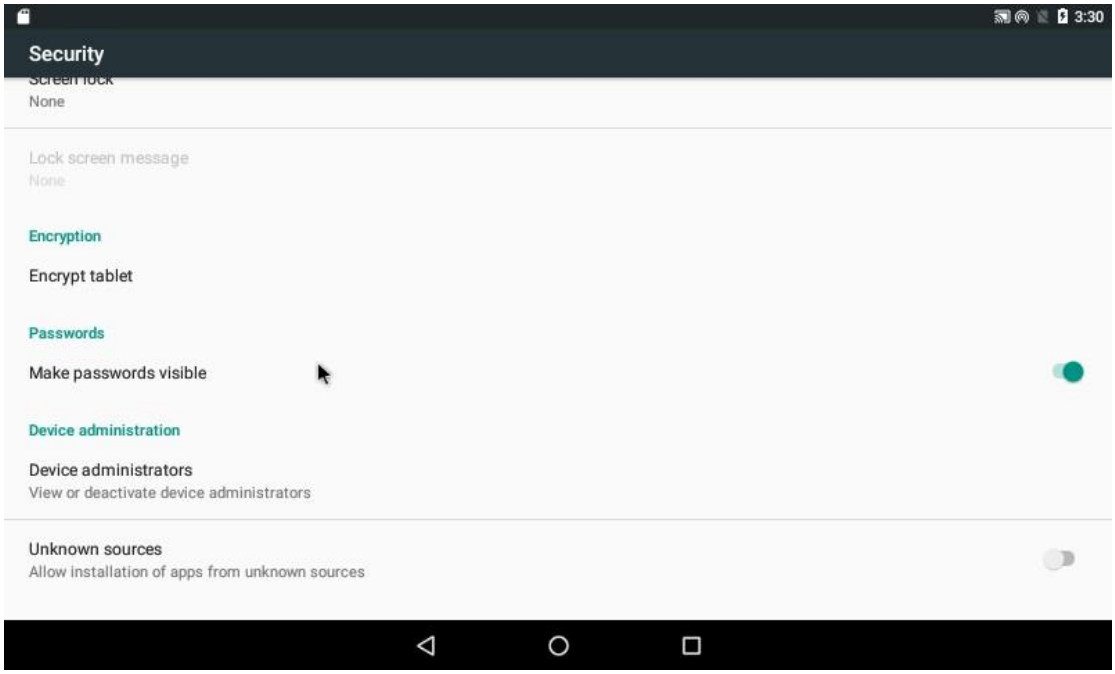
点击 TF 卡的标签“boot”，进入 TF 卡：



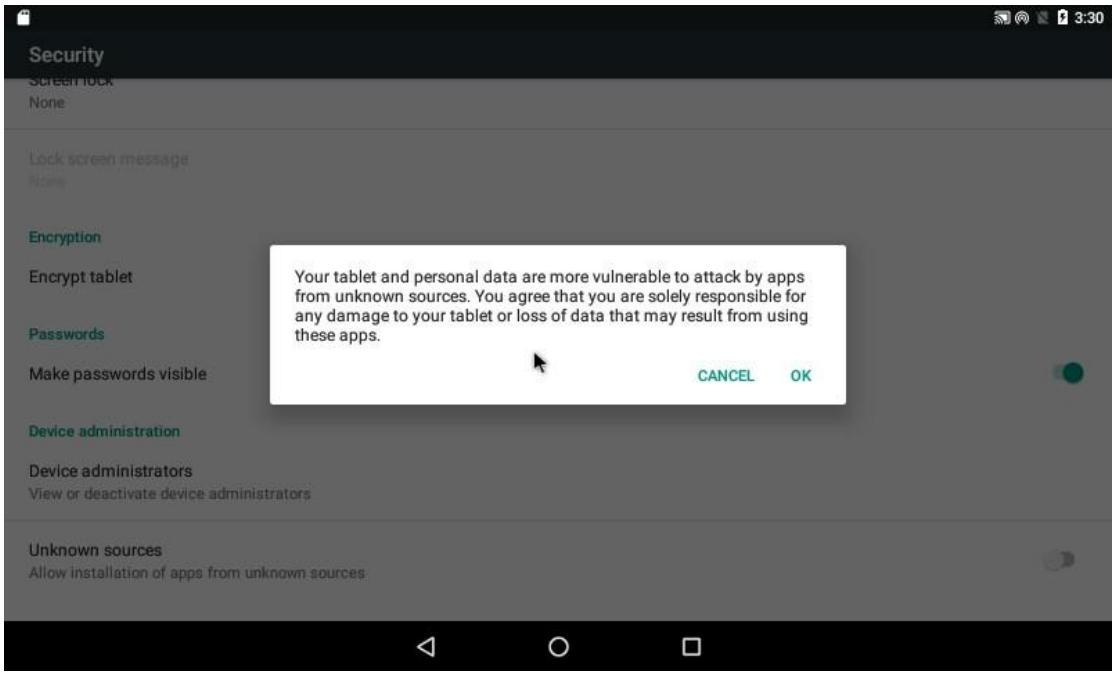
打开 APK，提示未知来源，需要打开允许未知来源应用，选择 SETTING：



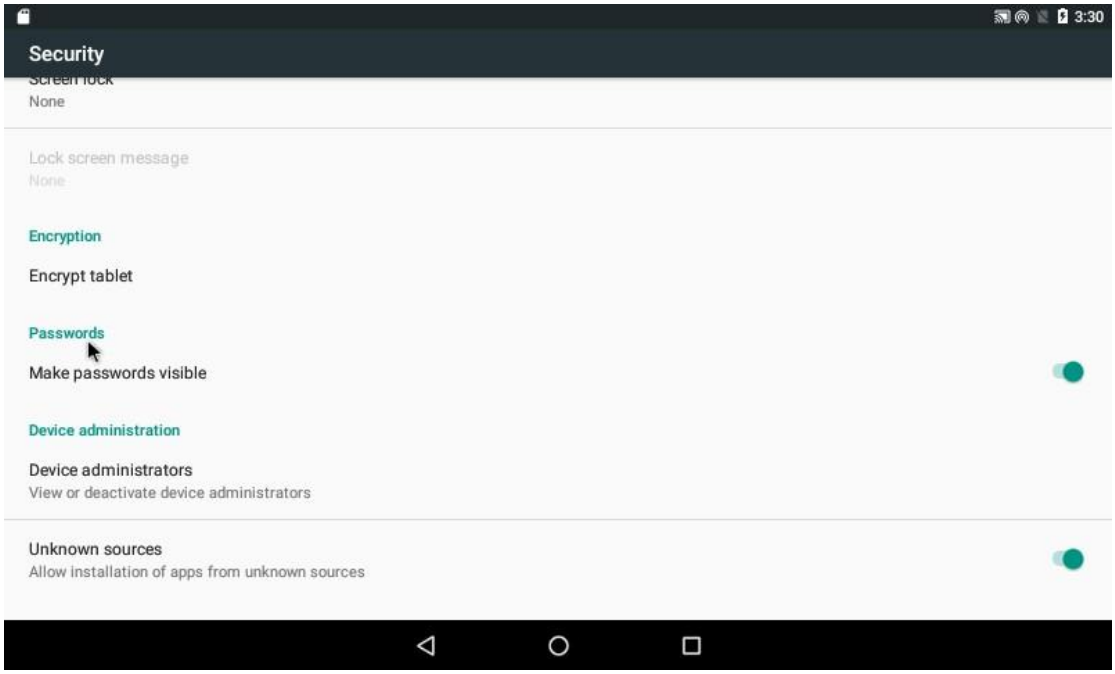
打开允许安装未知来源应用：



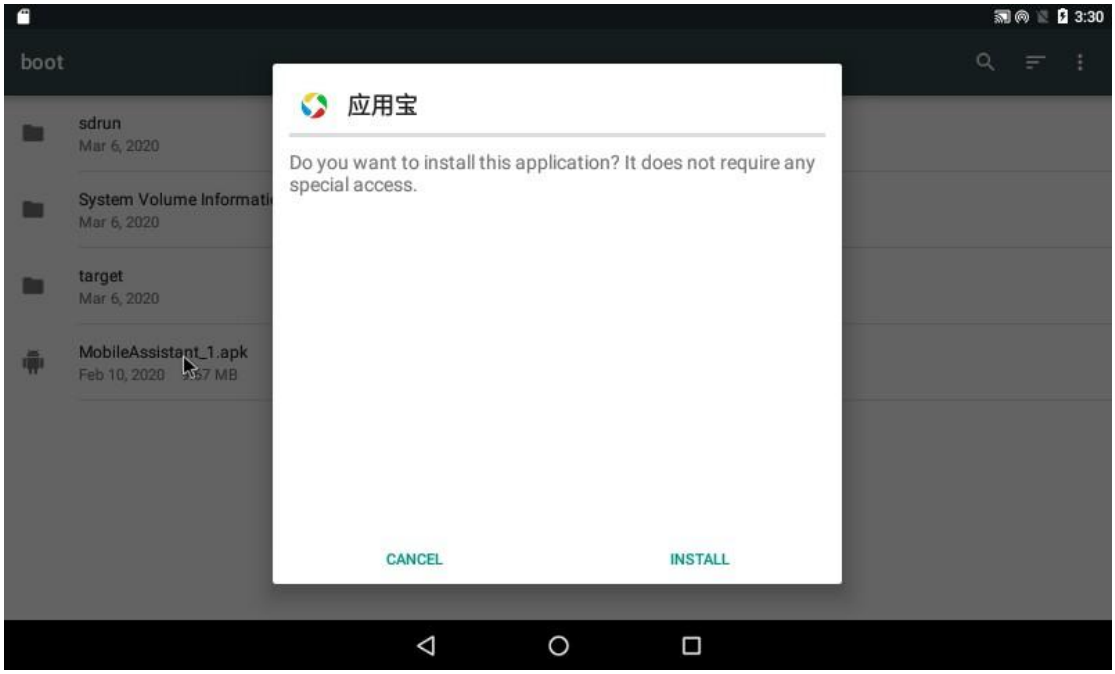
OK 确认：



允许安装未知来源应用已打开：



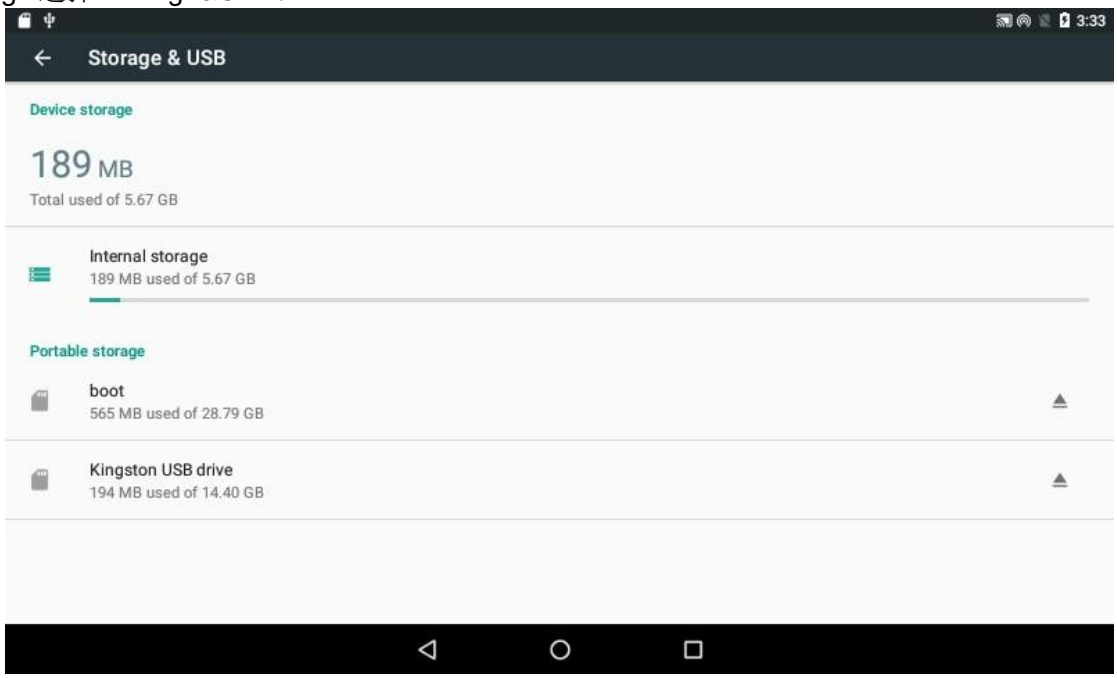
再次打开 APK，选择 INSTALL 进行安装：



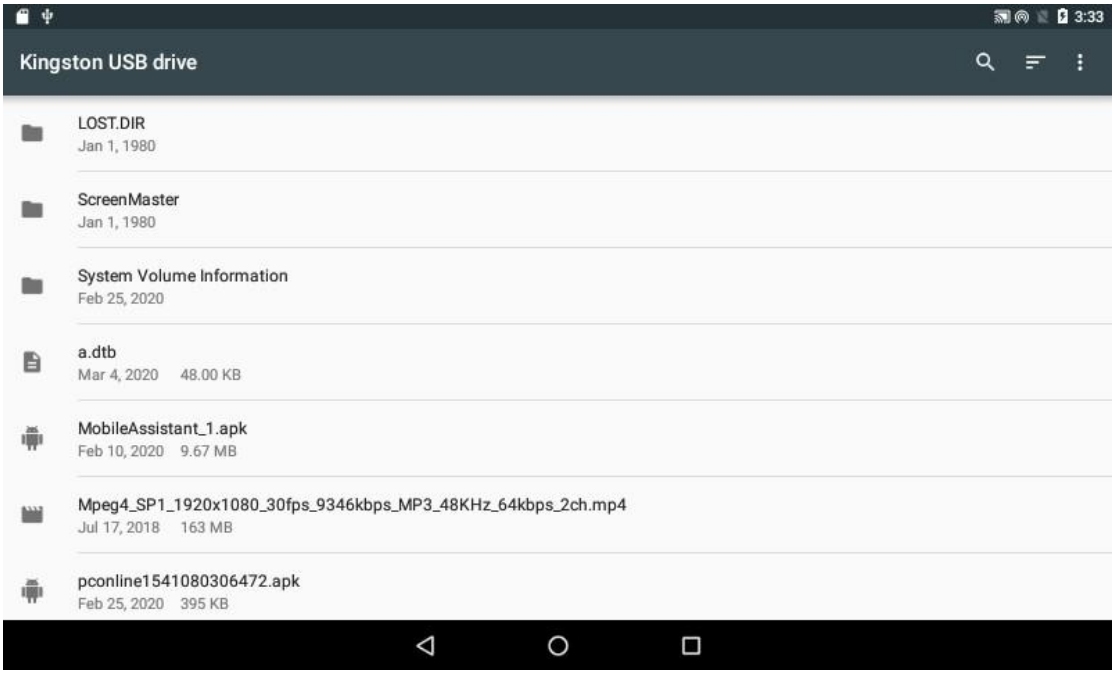
安装完成。

2.2 USB 安装

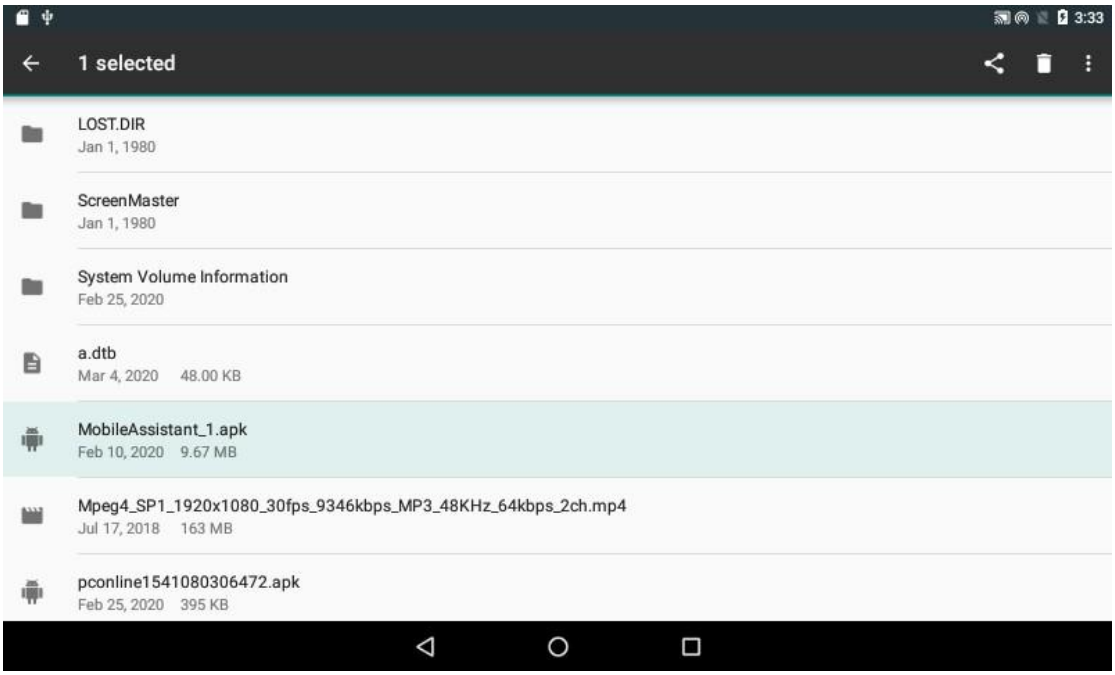
Settings 选择 Storage&USB：



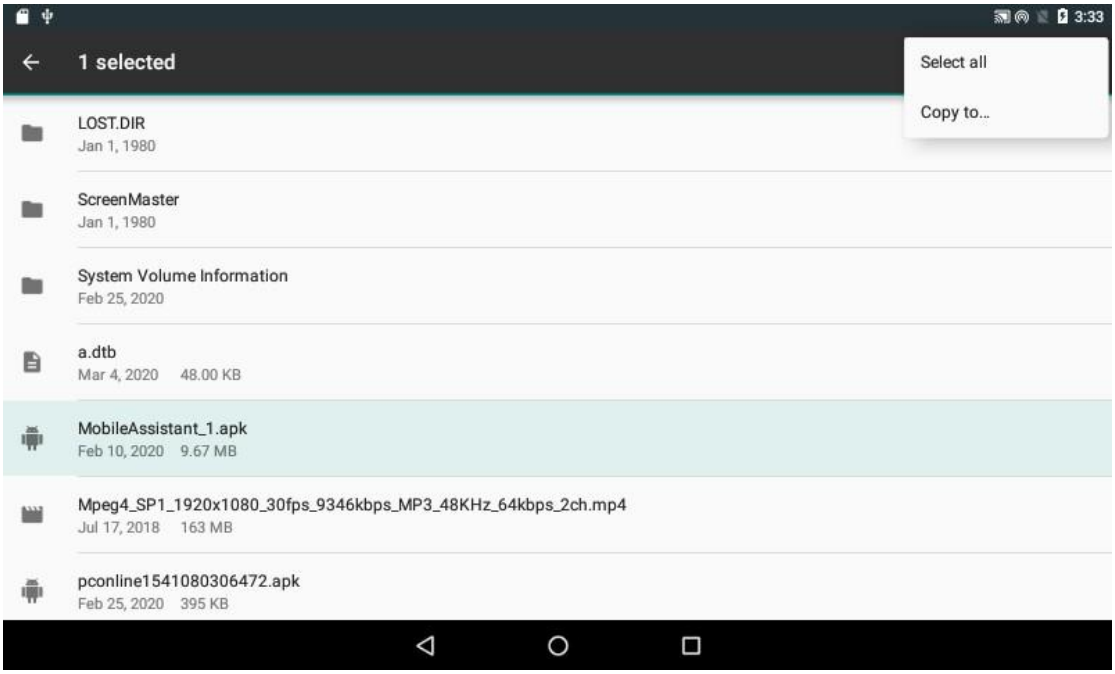
打开 U 盘（Kingston USB drive）：



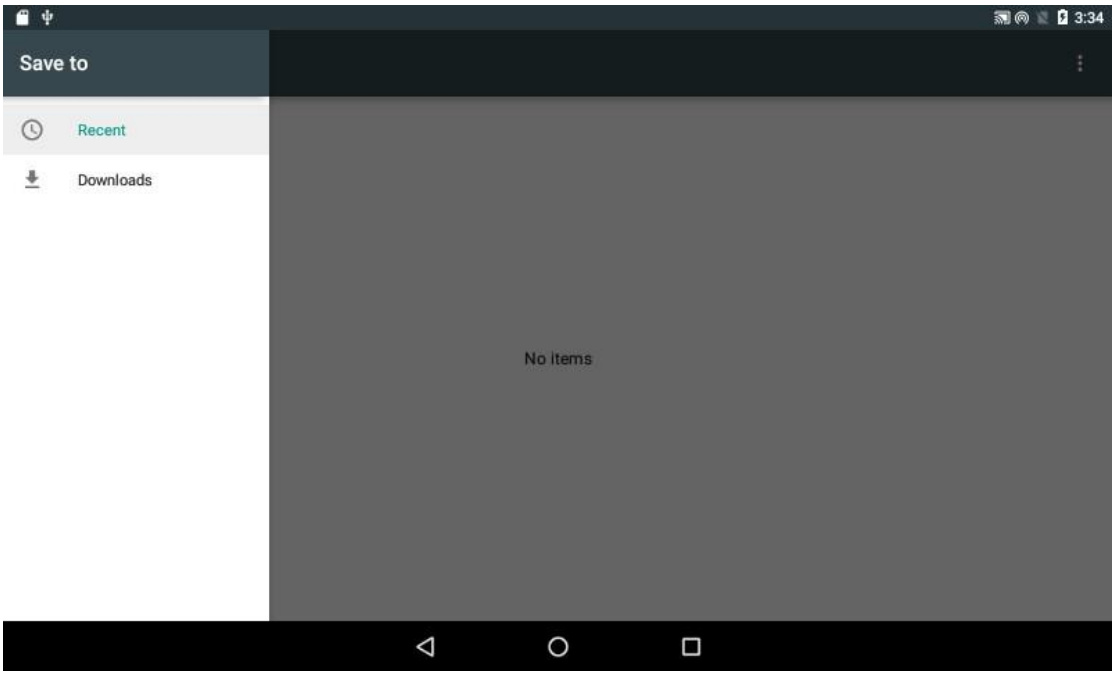
选择要安装的 APK：



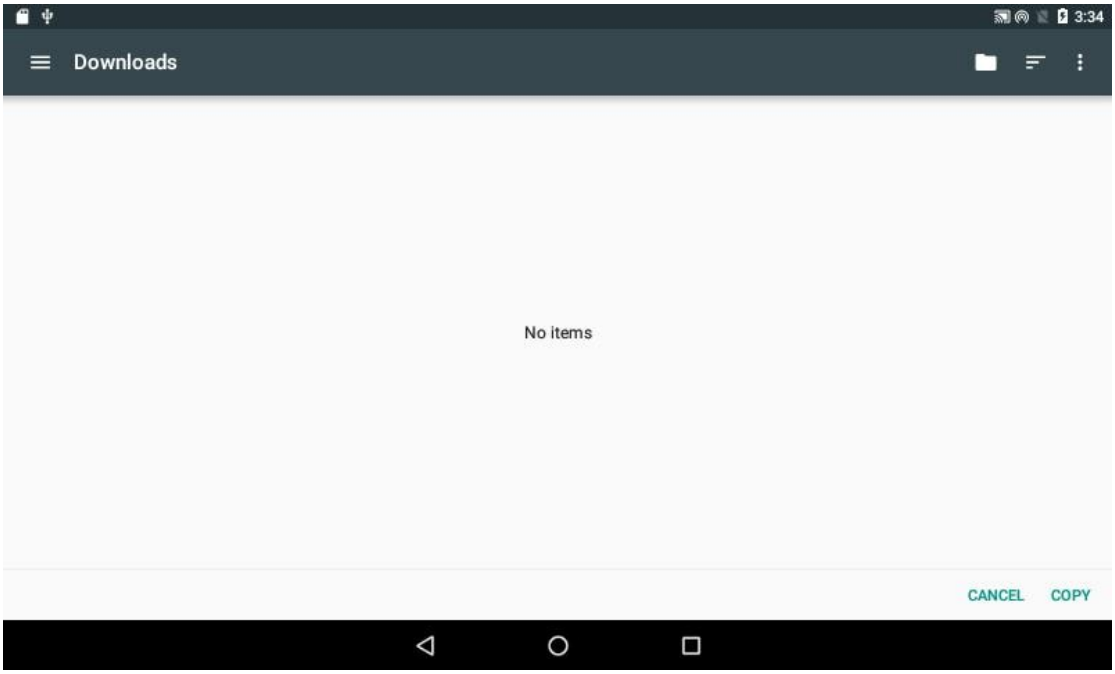
点击  选择 Copy to：



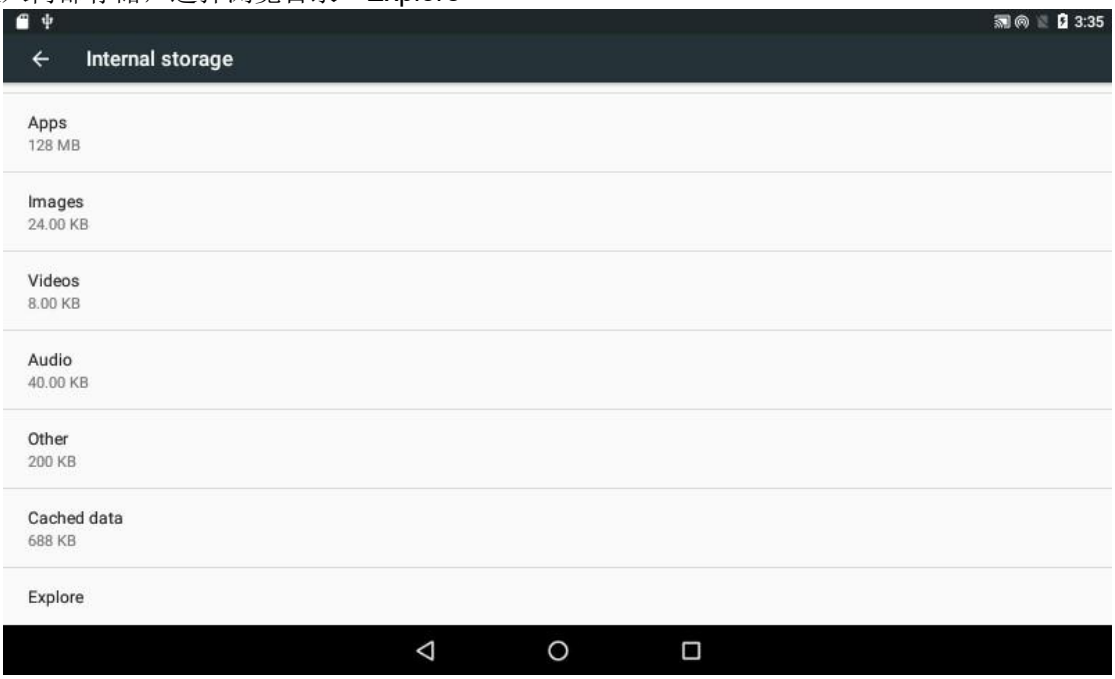
选择 Downloads 目录：

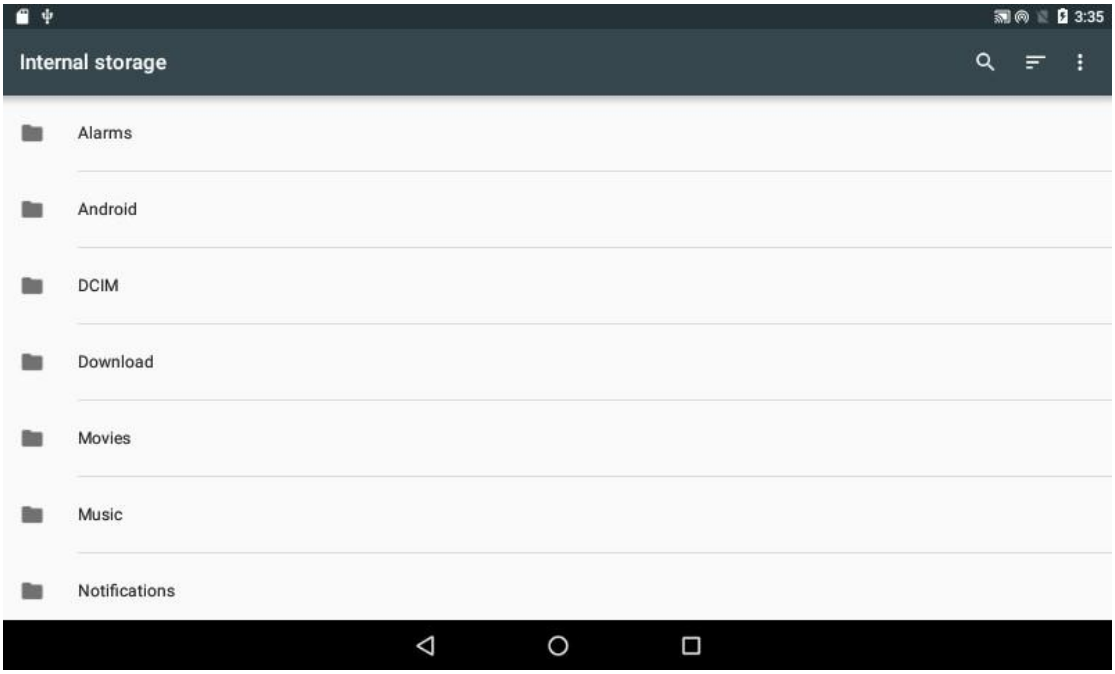


进入 downloads 目录，选择 COPY,将 APK 复制到当前目录：

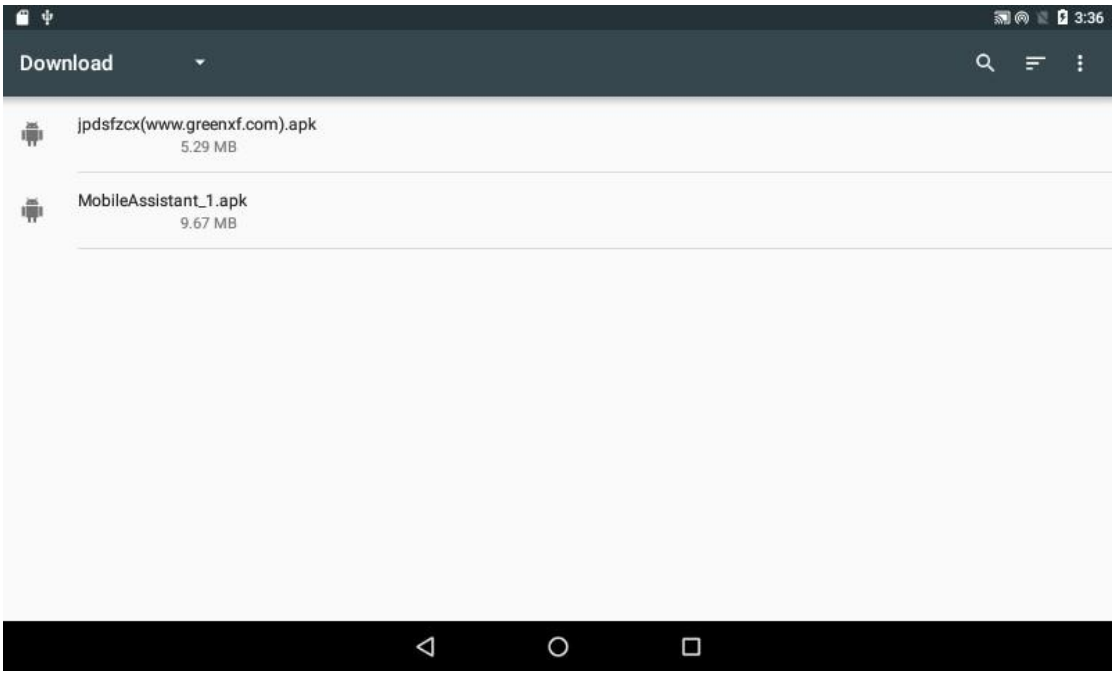


进入内部存储，选择浏览目录“Explore”

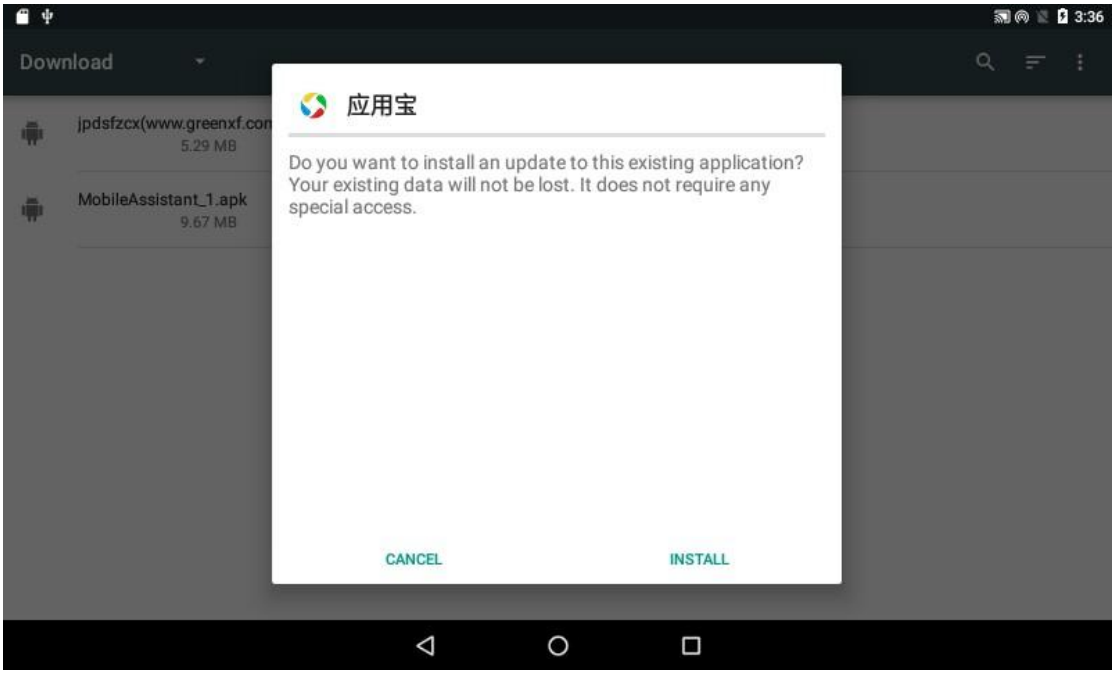




进入 Download 目录



打开 apk，选择 INSTALL 安装

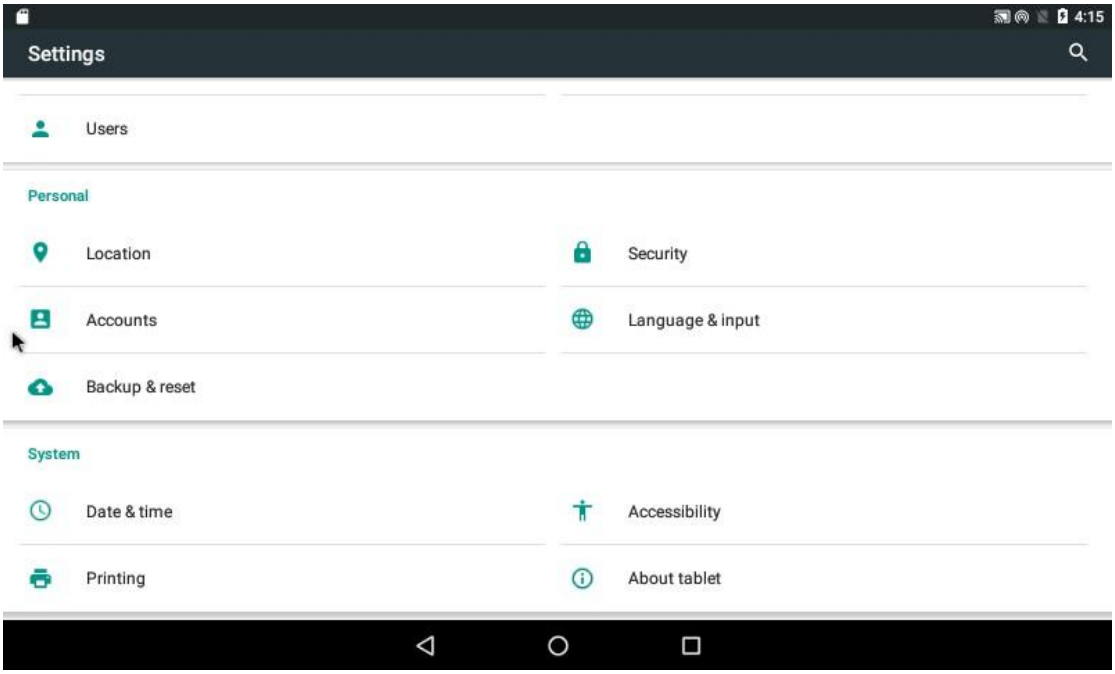

3:36
Download
jpdsfzcx(www.greenxf.com
5.29 MB
MobileAssistant_1.apk
9.67 MB
应用宝
Do you want to install an update to this existing application? Your existing data will not be lost. It does not require any special access.
CANCEL
INSTALL

安装完成。

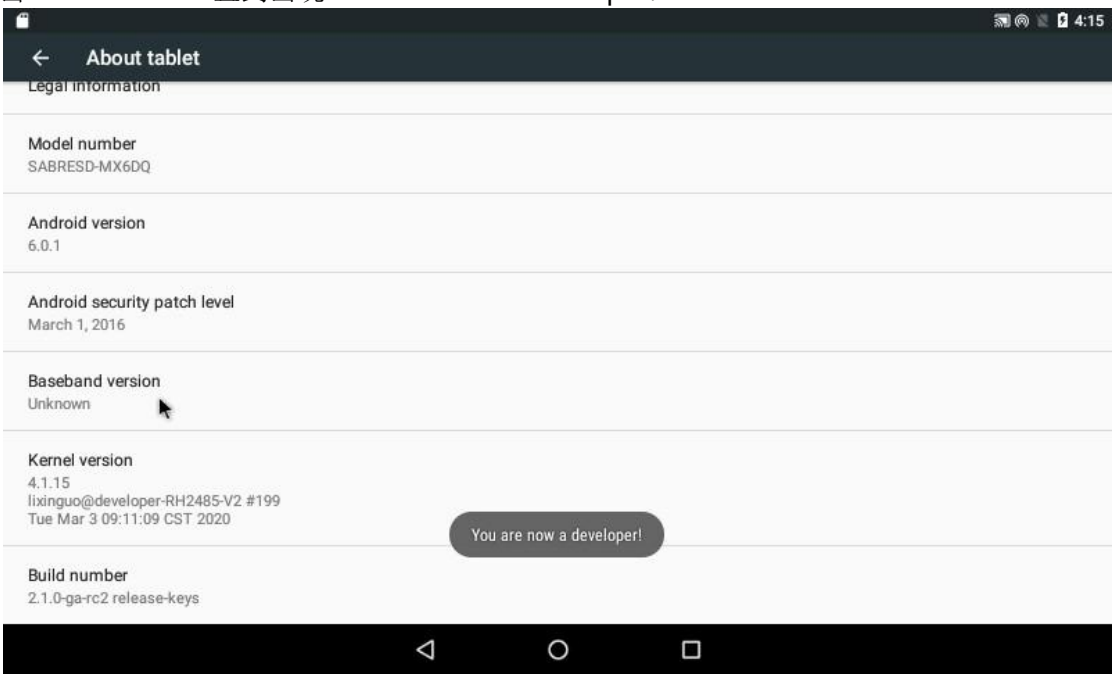

3:37
Download
jpdsfzcx(www.greenxf.com
5.29 MB
MobileAssistant_1.apk
9.67 MB
应用宝
App installed.
DONE
OPEN

附录三 打开 USB 调试

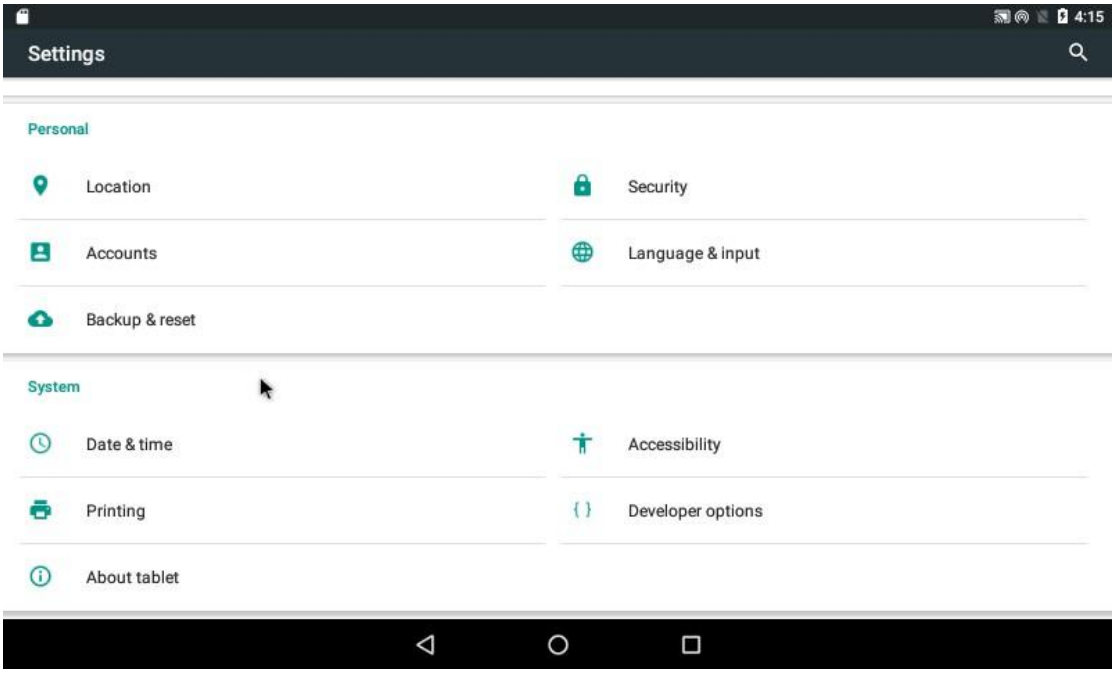
进入开发者模式需要先打开开发者选项，选择 About tablet ：



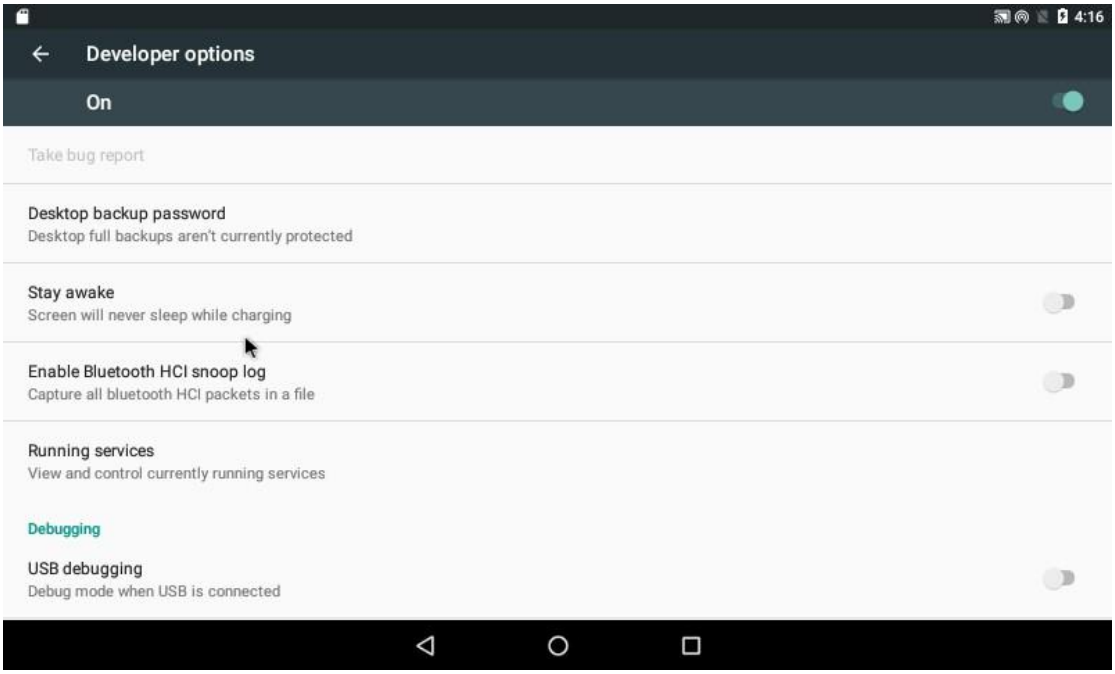
点击 Build number 直到出现 You are now a developer：

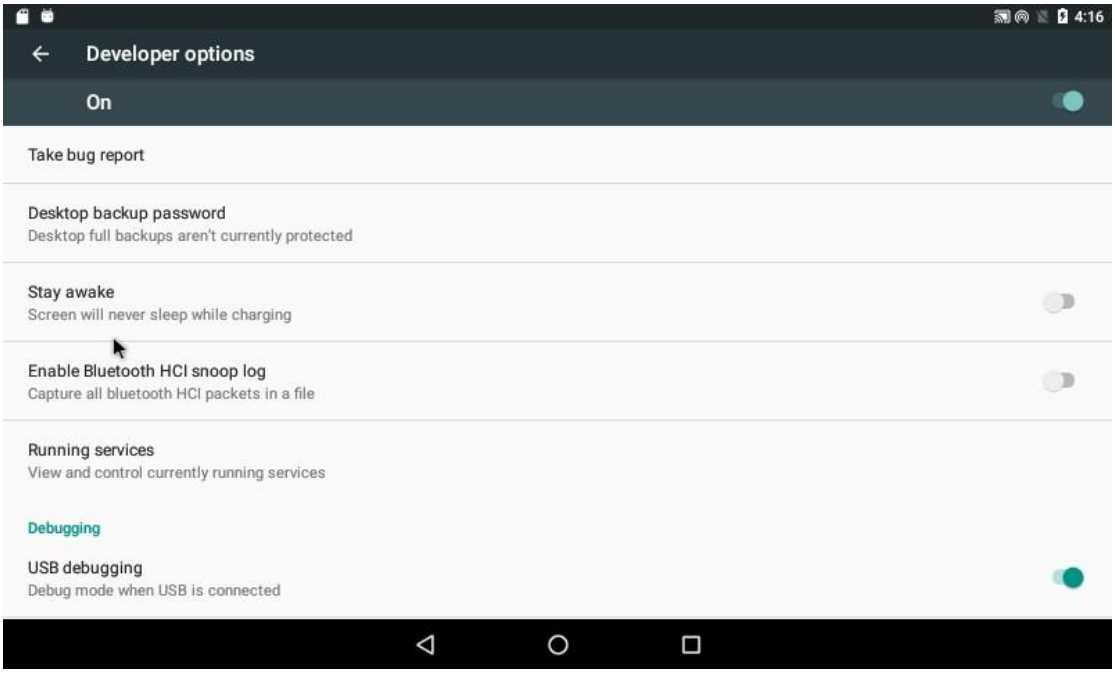


进入开发者选项：


4:15
Settings
Personal
Location
Security
Accounts
Language & input
Backup & reset
System
Date & time
Accessibility
Printing
Developer options
About tablet

打开 USB 调试：


4:16
Developer options
On
Take bug report
Desktop backup password
Desktop full backups aren't currently protected
Stay awake
Screen will never sleep while charging
Enable Bluetooth HCI snoop log
Capture all bluetooth HCI packets in a file
Running services
View and control currently running services
Debugging
USB debugging
Debug mode when USB is connected


4:16
Developer options
On
Take bug report
Desktop backup password
Desktop full backups aren't currently protected
Stay awake
Screen will never sleep while charging
Enable Bluetooth HCI snoop log
Capture all bluetooth HCI packets in a file
Running services
View and control currently running services
Debugging
USB debugging
Debug mode when USB is connected

附录四 Android 应用程序开发

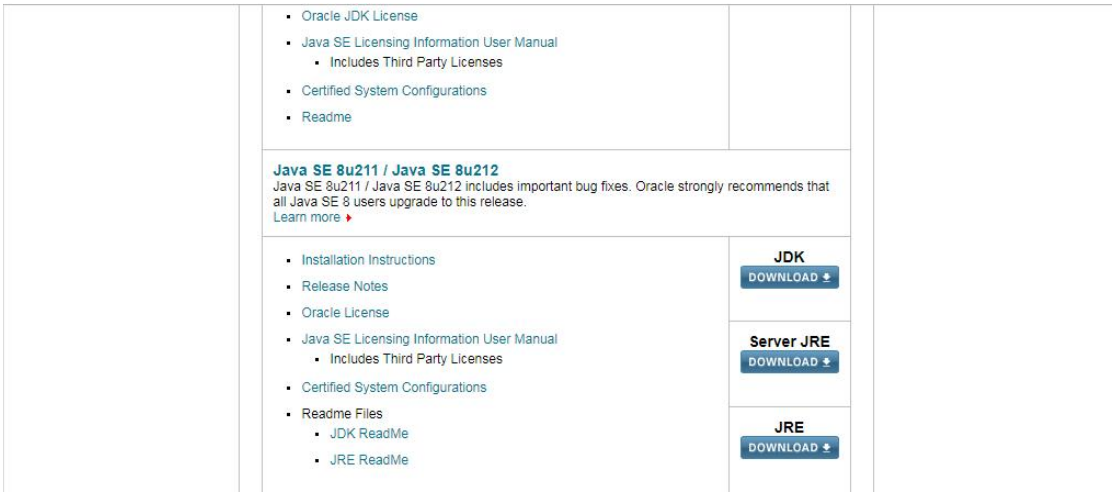
本章节讲解如何建立 Android 应用开发环境，包括 Android SDK 和 Android studio 集成开发环境的下载及安装，以及如何使用 OKMX6 开发板作为真机调试程序，非常适合 Android 初学者学习和参考。

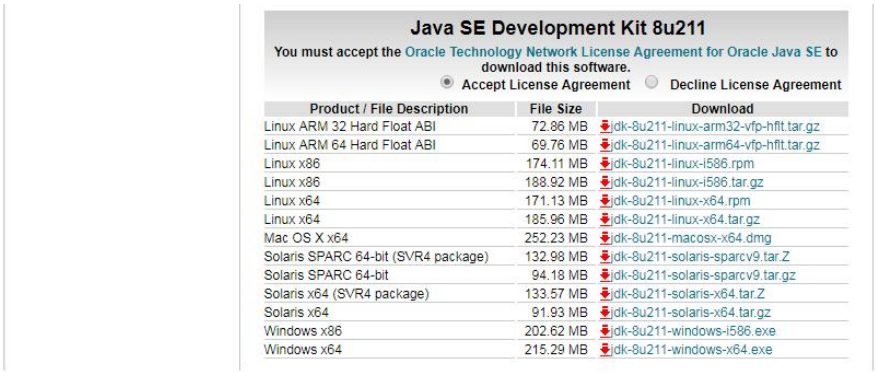
4.1 建立 Android 应用开发环境

4.1.1 下载并安装 JDK（Java SE Development Kit）

由于 Android 应用代码都是用 Java 编写的，因此需要先在 Windows 上安装 JDK，JDK 可按以下方法下载：

访问网站 http://www.oracle.com/technetwork/java/javase/downloads/index.html，在页面中点击 JDK，推荐安装 Java8。





Java SE Development Kit 8u211

You must accept the Oracle Technology Network License Agreement for Oracle Java SE to download this software.

Accept License Agreement　Decline License Agreement

Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	72.86 MB	jdk-8u211-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	69.76 MB	jdk-8u211-linux-arm64-vfp-hflt.tar.gz
Linux x86	174.11 MB	jdk-8u211-linux-i586.rpm
Linux x86	188.92 MB	jdk-8u211-linux-i586.tar.gz
Linux x64	171.13 MB	jdk-8u211-linux-x64.rpm
Linux x64	185.96 MB	jdk-8u211-linux-x64.tar.gz
Mac OS X x64	252.23 MB	jdk-8u211-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	132.98 MB	jdk-8u211-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	94.18 MB	jdk-8u211-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	133.57 MB	jdk-8u211-solaris-x64.tar.Z
Solaris x64	91.93 MB	jdk-8u211-solaris-x64.tar.gz
Windows x86	202.62 MB	jdk-8u211-windows-i586.exe
Windows x64	215.29 MB	jdk-8u211-windows-x64.exe

点击“Accept License Agreement”根据 windows 版本选择 exe 安装程序。您也可以在光盘资料工具目录找到飞凌提供文件。

下载完成后，双击安装程序，根据向导的提示完成安装即可。

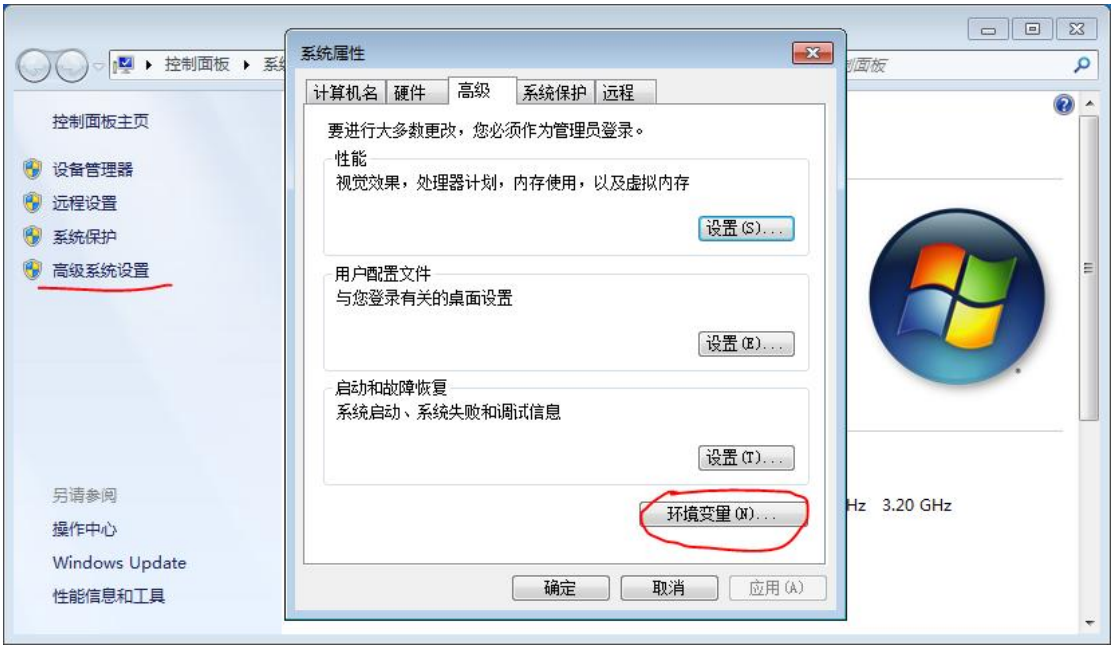
安装完成后，需要将 JDK 命令添加到 Path 环境变量中，通过下面的方法将 JDK 命令所在的路径添加到 Path 环境变量中：

1) 右击“我的电脑”- > 属性，再选择左边导航的“高级系统设置”选项。

2) 点击右下角的“环境变量”选项。

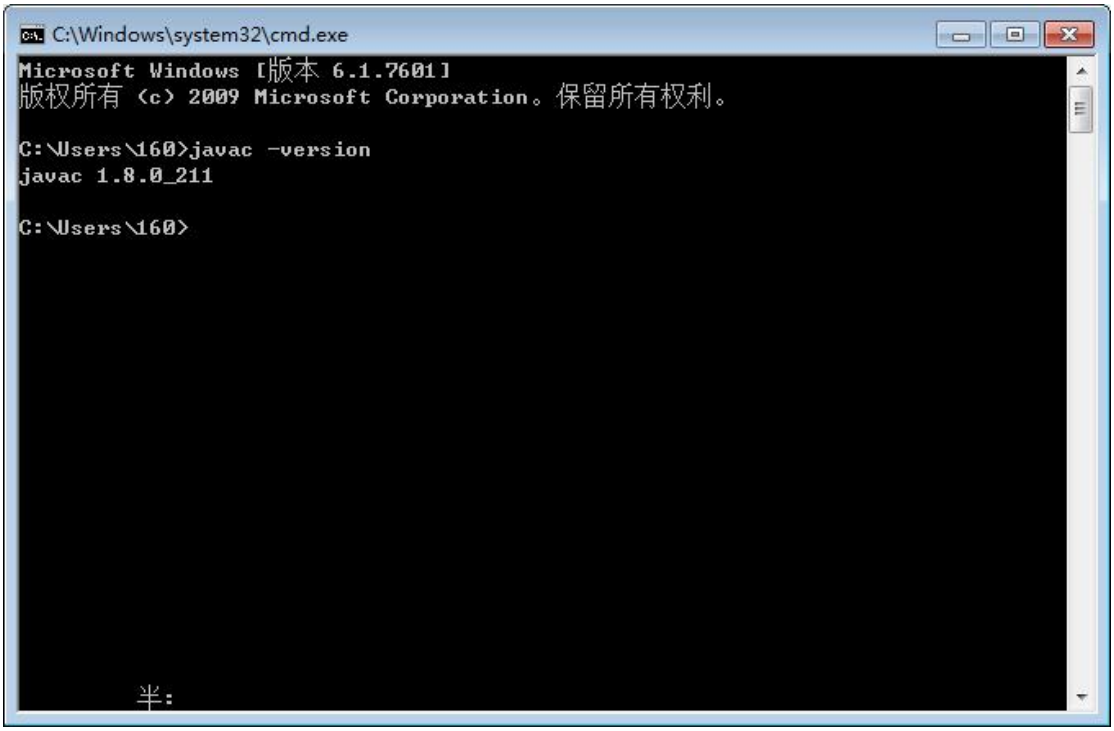
3) 在“系统变量”中，找到 Path 环境变量，双击它，根据实际安装路径设置 java 环境变量，默认安装时追加以下内容“C:\Program Files\Java\jdk1.8.0_211\bin”

4) 点击“确定”完成环境变量设置。



5) 检查安装是否成功

打开命令提示符工具，输入 javac -version



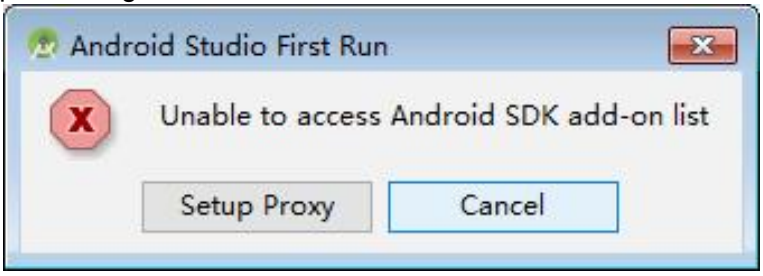
正确显示 Java 版本即表示安装成功。

4.1.2 安装 Android studio

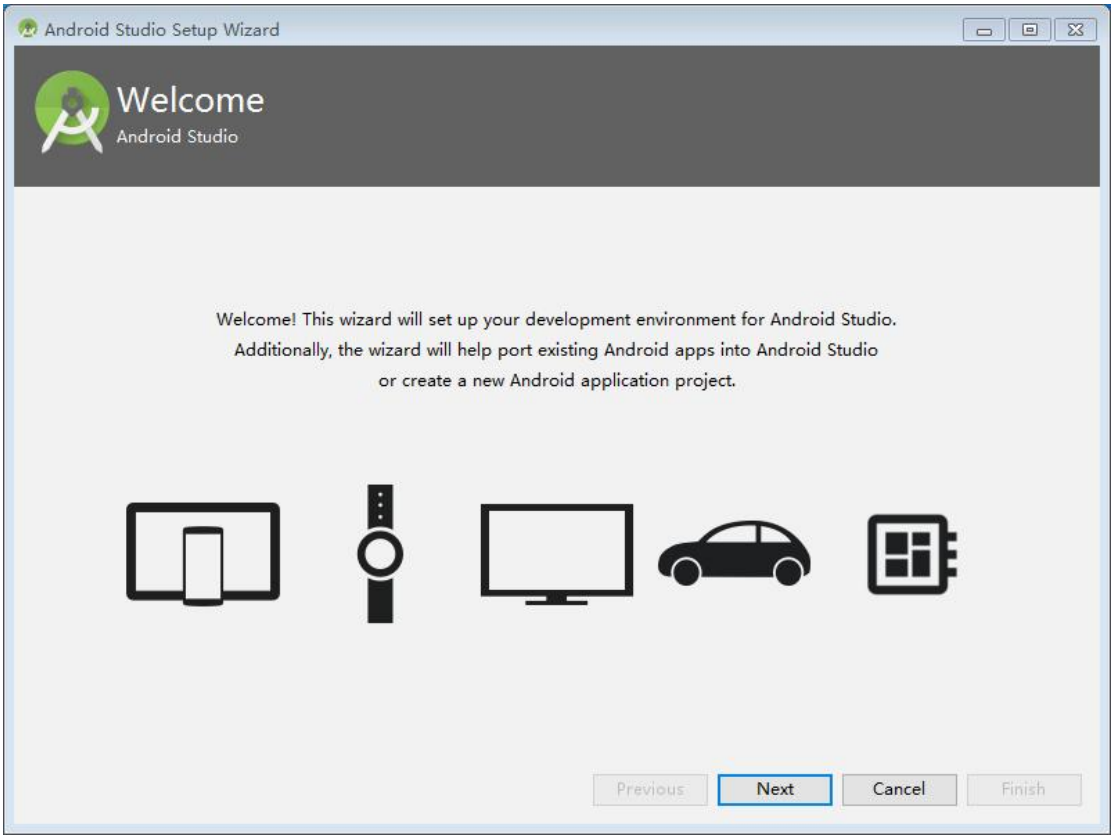
Android Studio 是 Google 于 2013 I/O 大会针对 Android 开发推出的新的开发工具,国内可在 http://www.android-studio.org/进行下载安装。

下载完成后按照提示进行安装即可。安装完成后将出现下图所示：

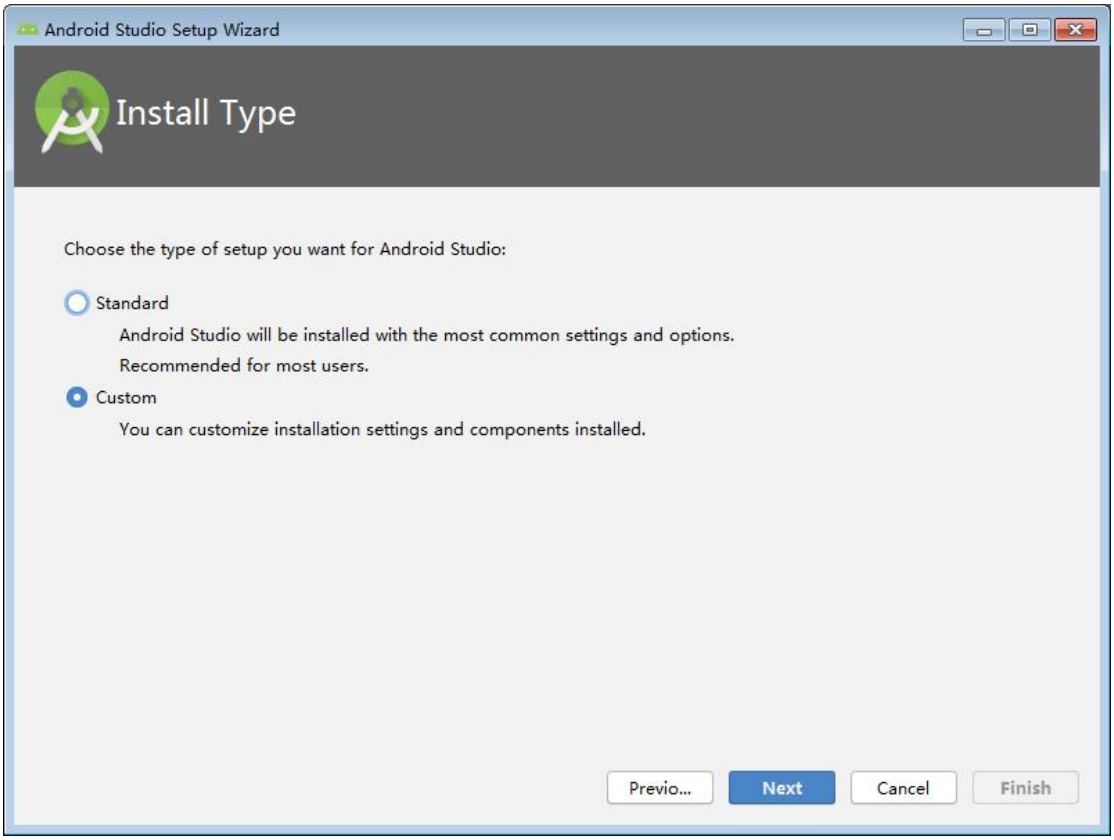
选择“Do not import settings”



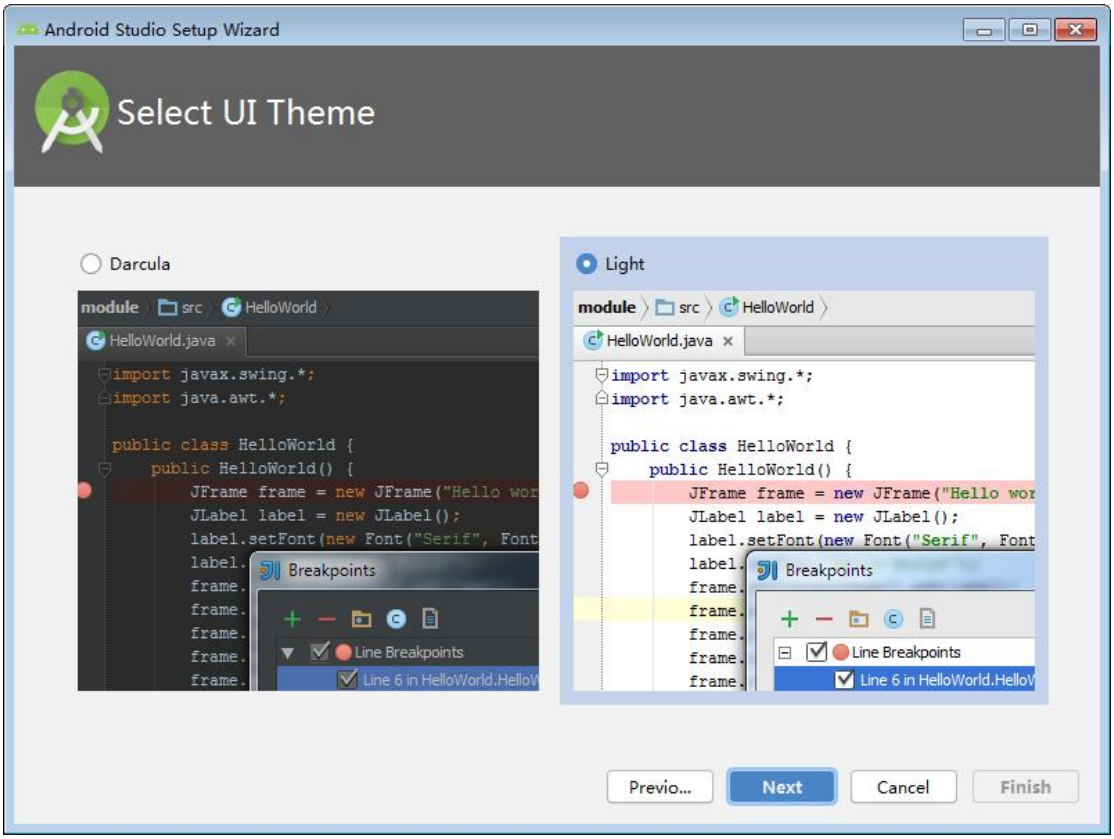
选择“Cancel”取消。



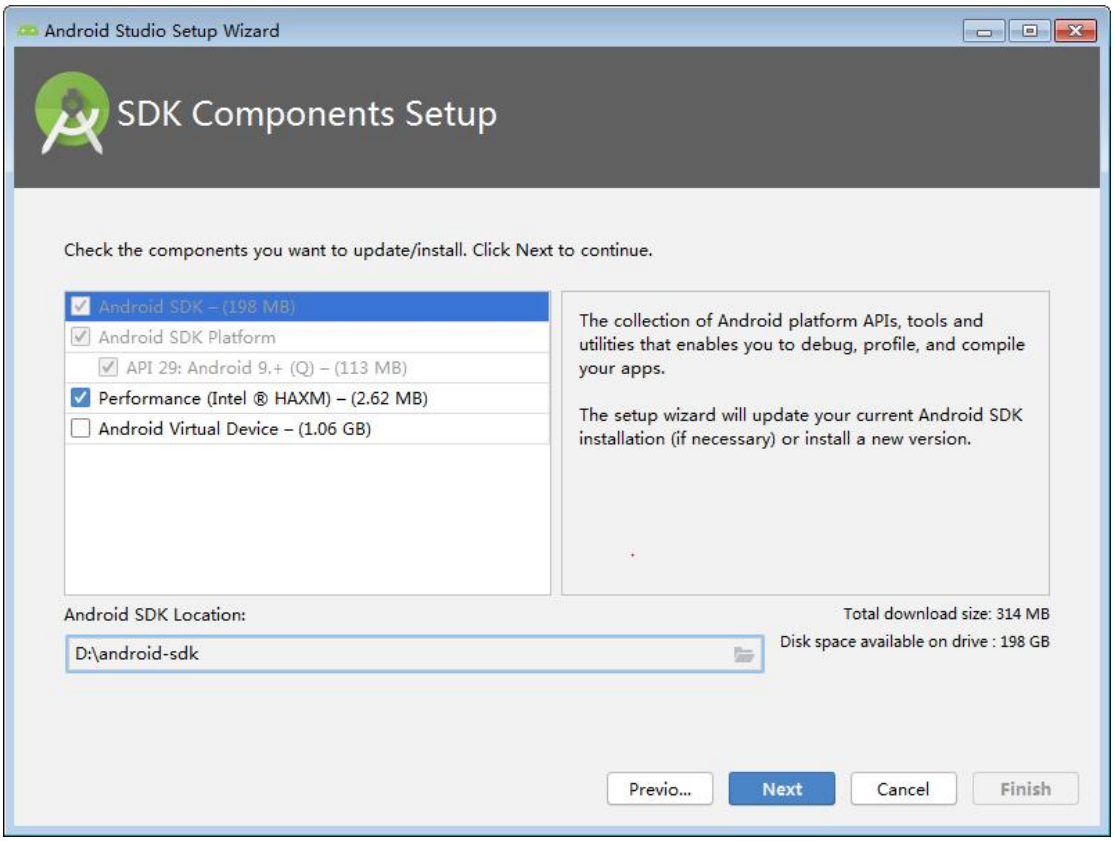
点击“next”进行下一步



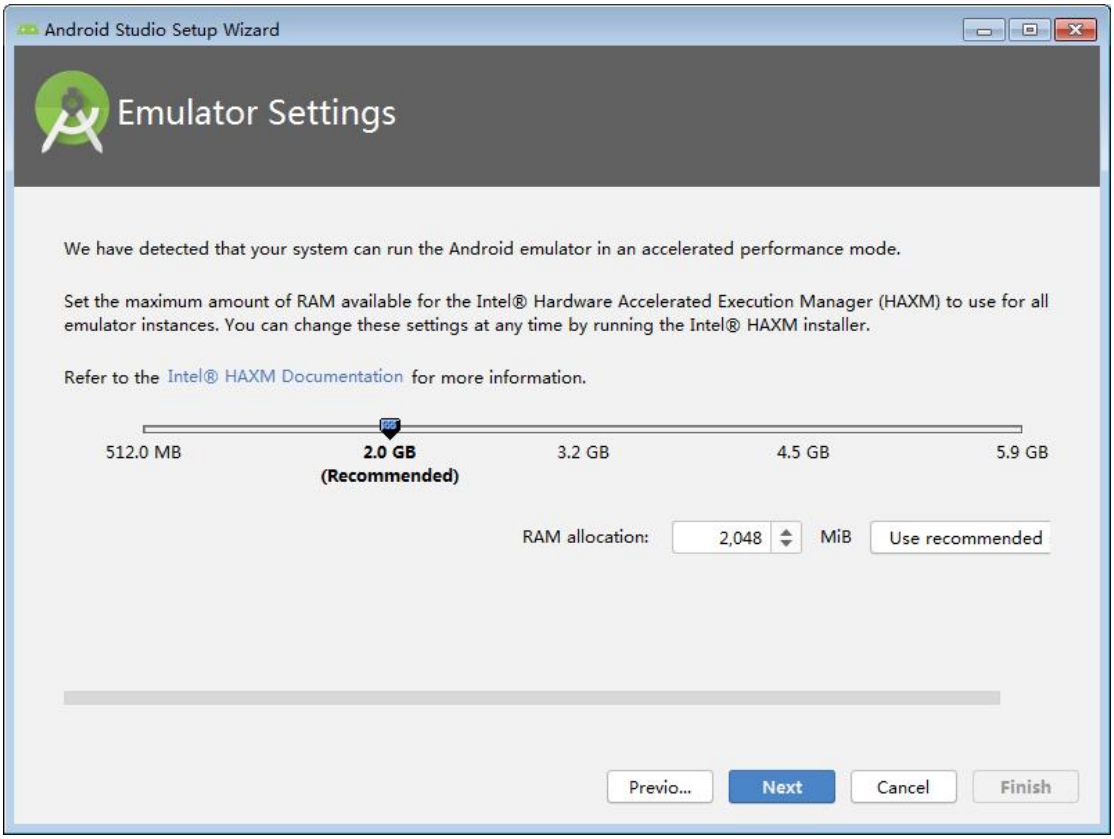
这里选择“custom”



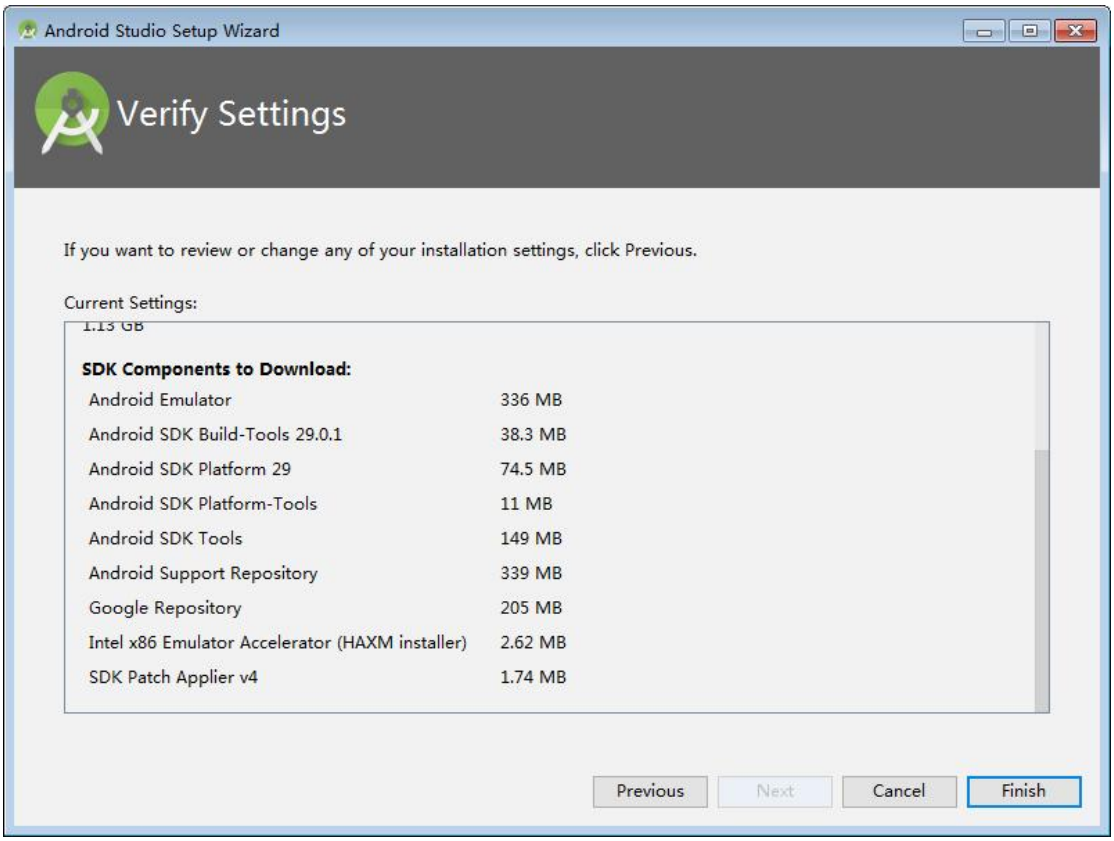
根据个人喜好选择 UI 主题



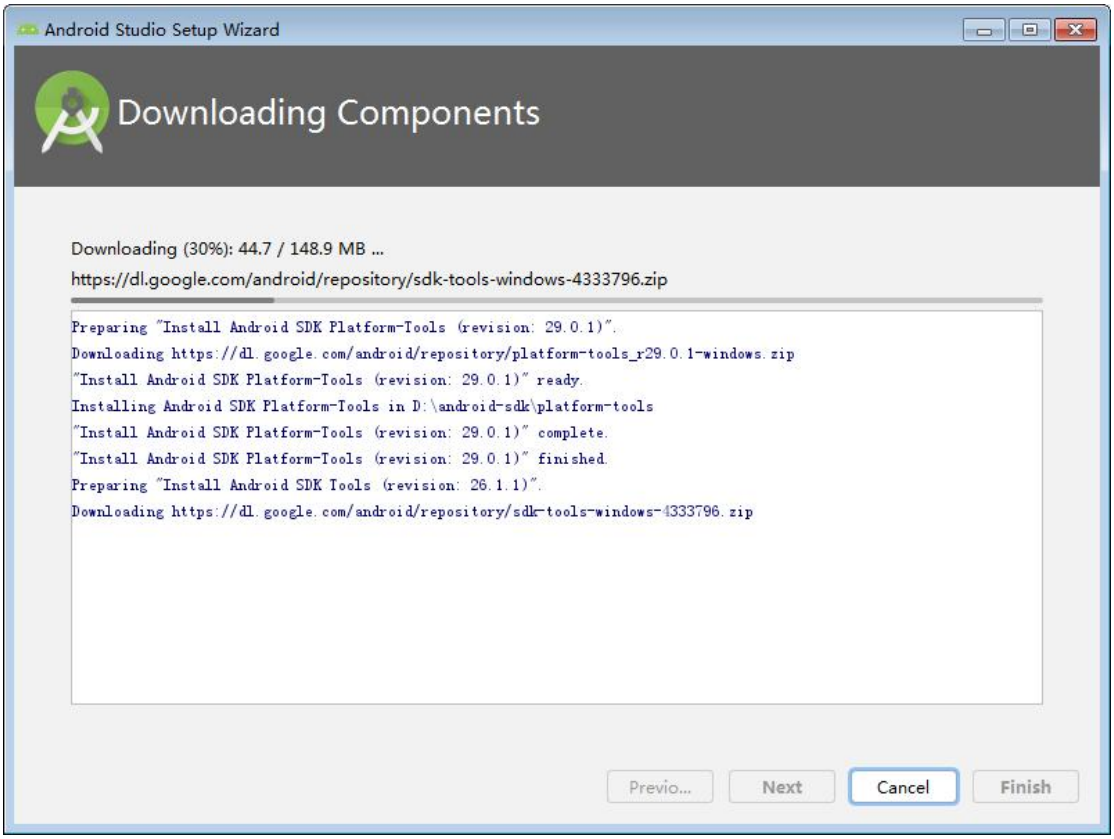
根据实际情况选择 Android SDK 的安装路径



选择模拟器的内存大小



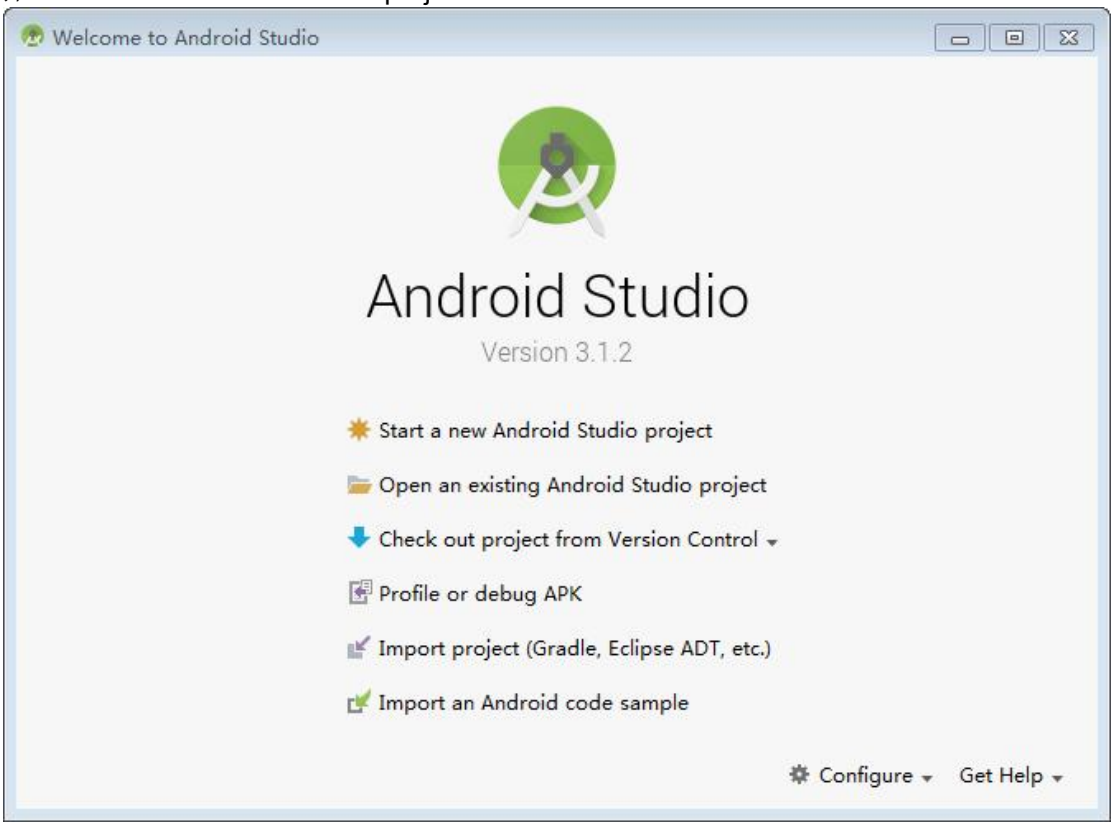
点击“Finish”



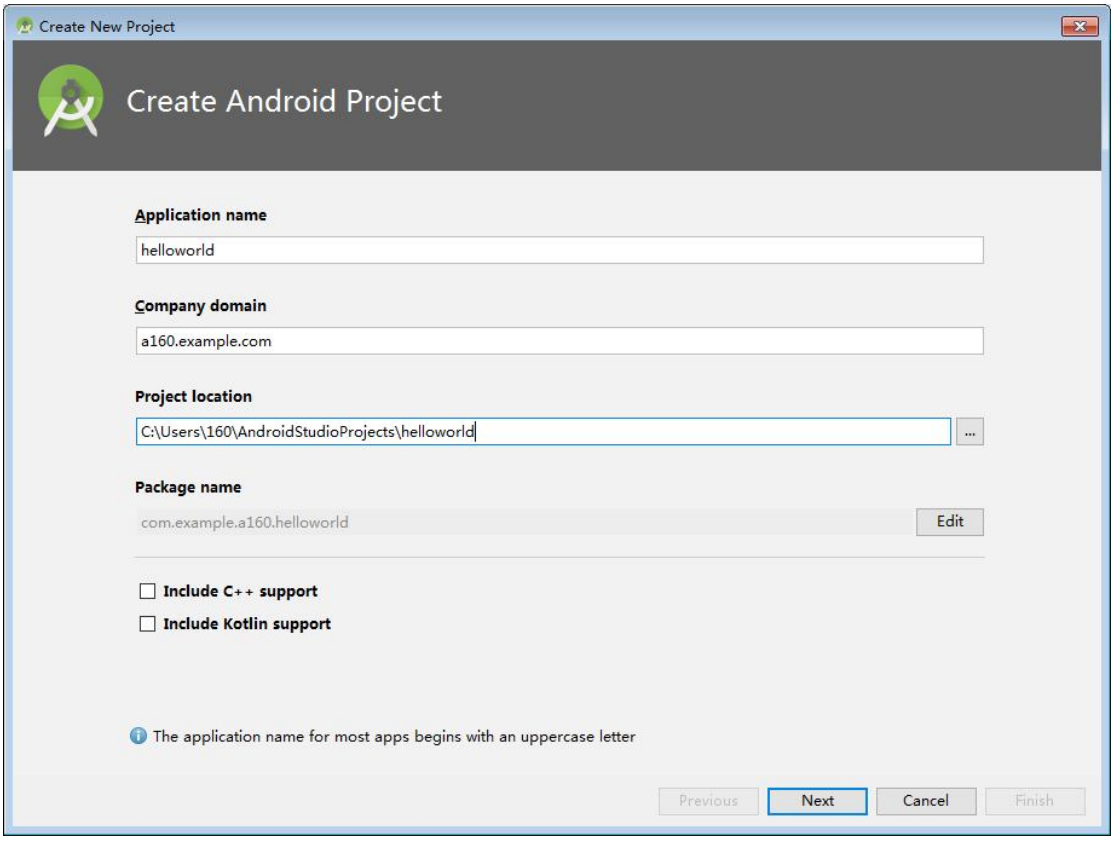
等待安装完成，点击“Finish”

4.1.3 创建 Helloworld 工程

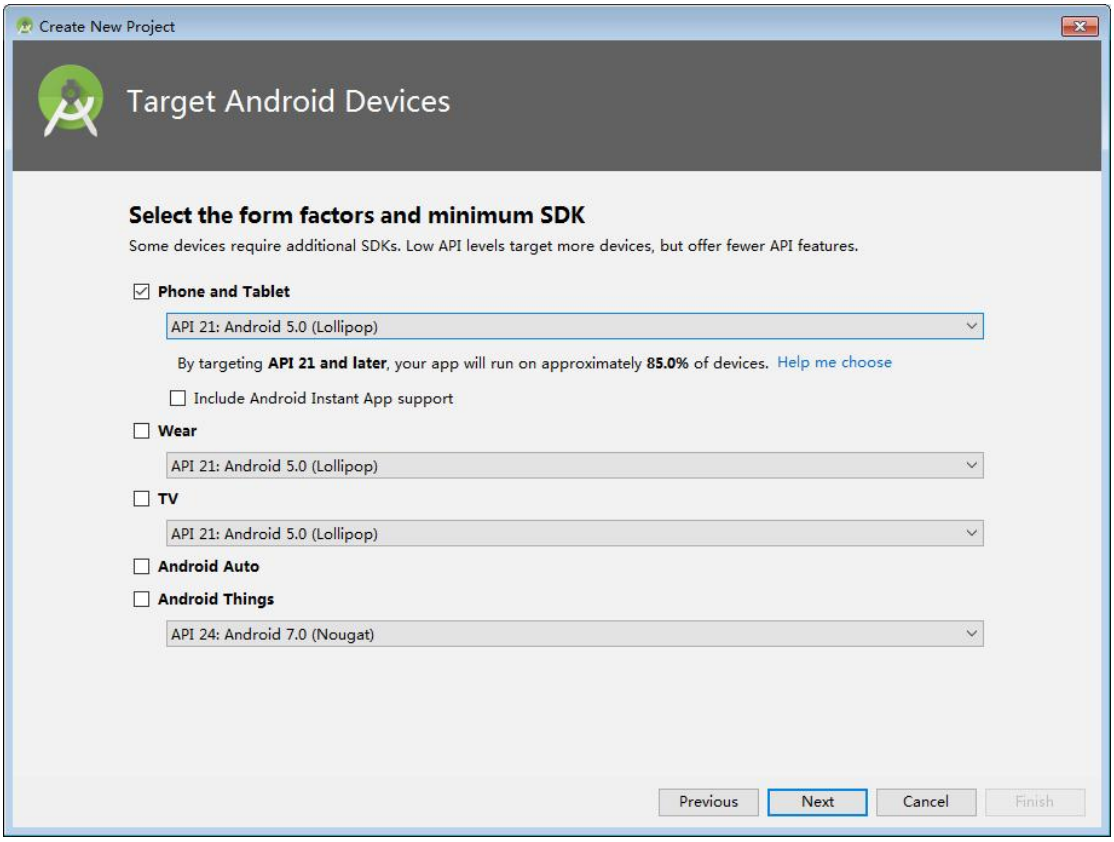
1. 选择“start a new android studio project”



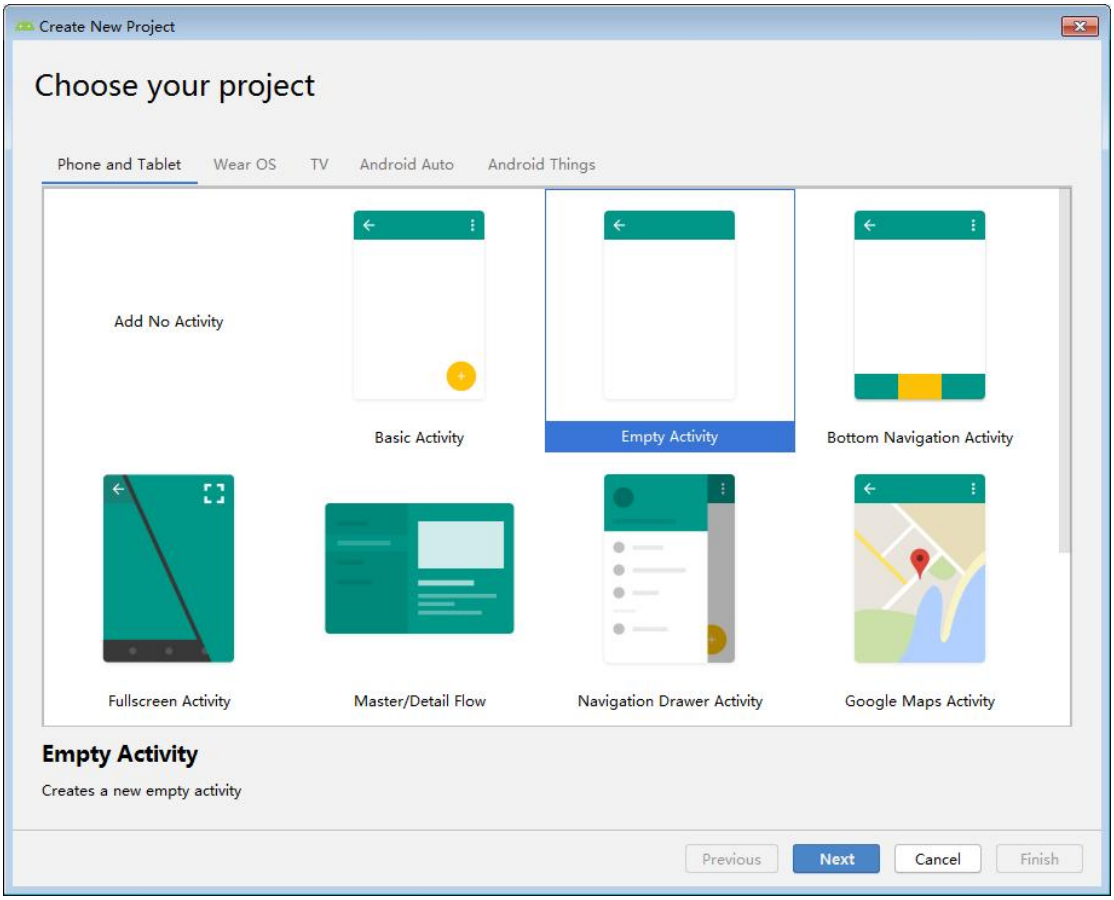
2. 修改项目名称



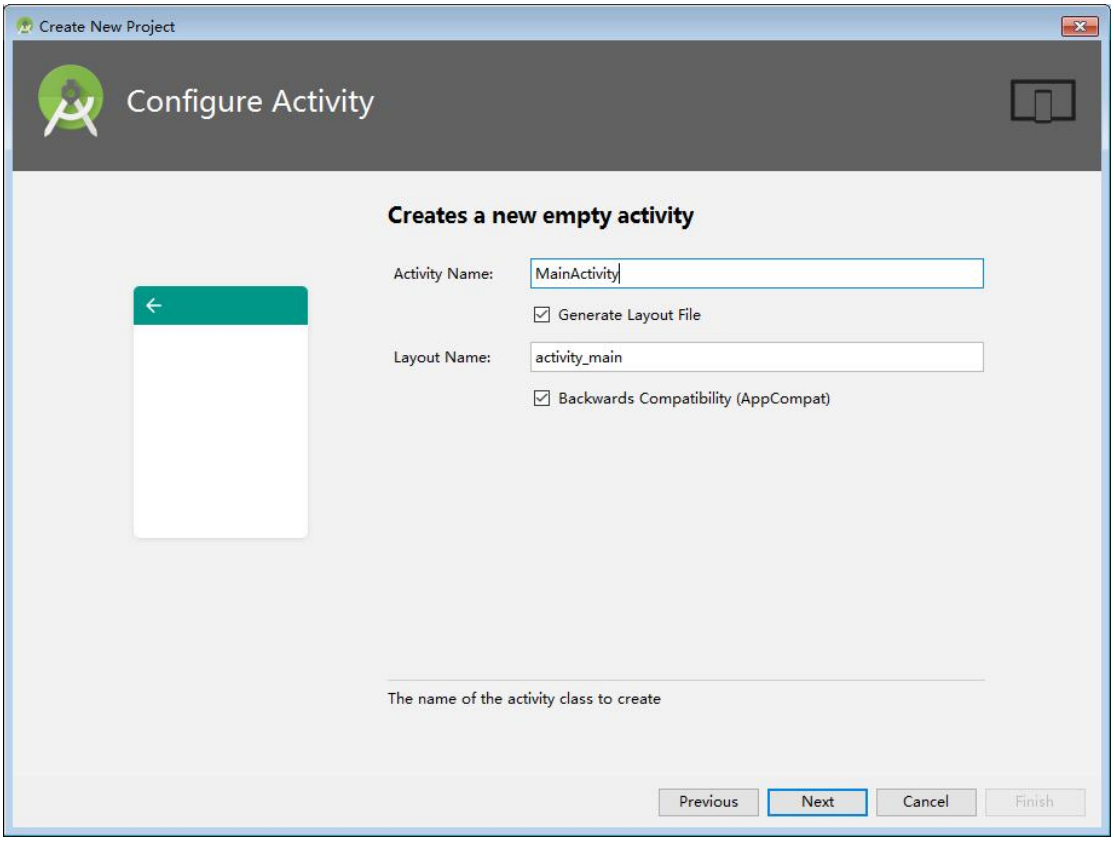
3. 选择最低支持的 SDK 版本

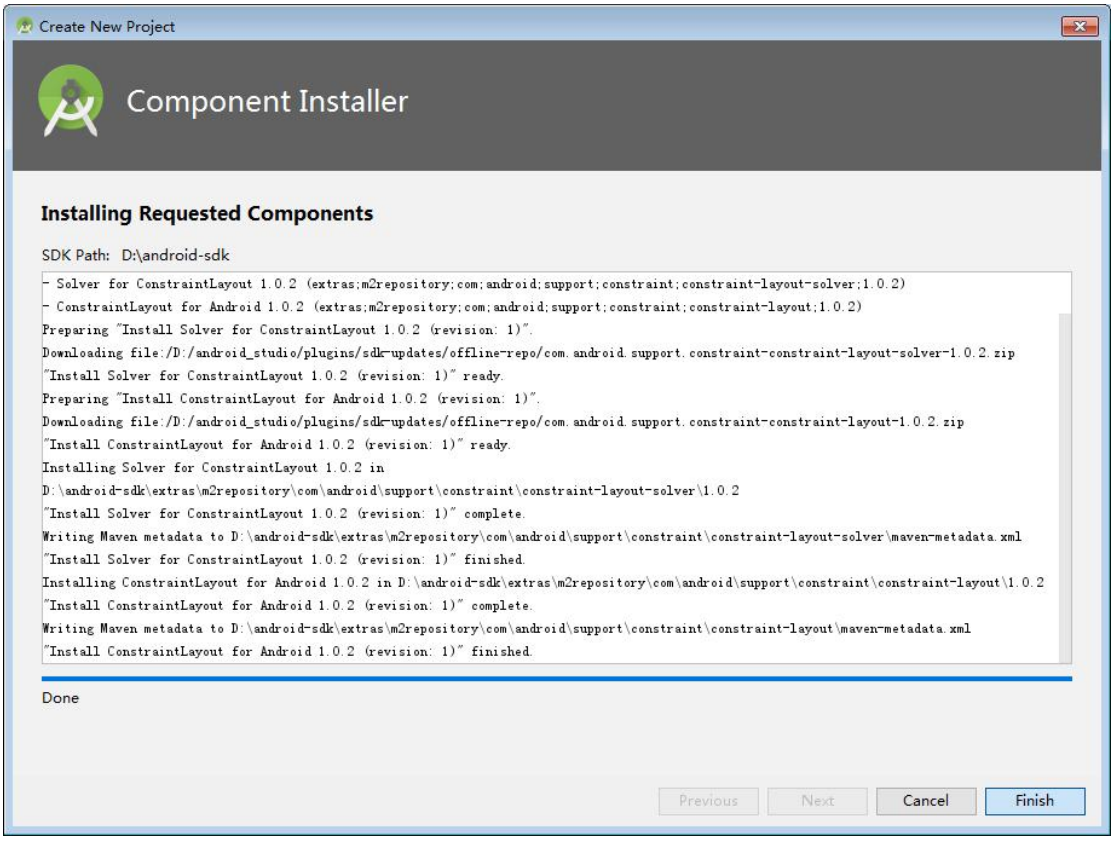


4. 选择 Empty Activity

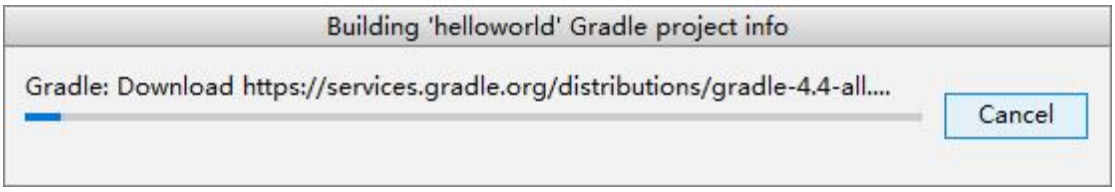

Create New Project
Choose your project
Phone and Tablet
Wear OS
TV
Android Auto
Android Things
Add No Activity
Basic Activity
Empty Activity
Bottom Navigation Activity
Fullscreen Activity
Master/Detail Flow
Navigation Drawer Activity
Google Maps Activity
Empty Activity
Creates a new empty activity
Previous
Next
Cancel
Finish

点击“Next”


Create New Project
Configure Activity
Creates a new empty activity
Activity Name:
MainActivity
Generate Layout File
Layout Name:
activity_main
Backwards Compatibility (AppCompat)
The name of the activity class to create
Previous
Next
Cancel
Finish



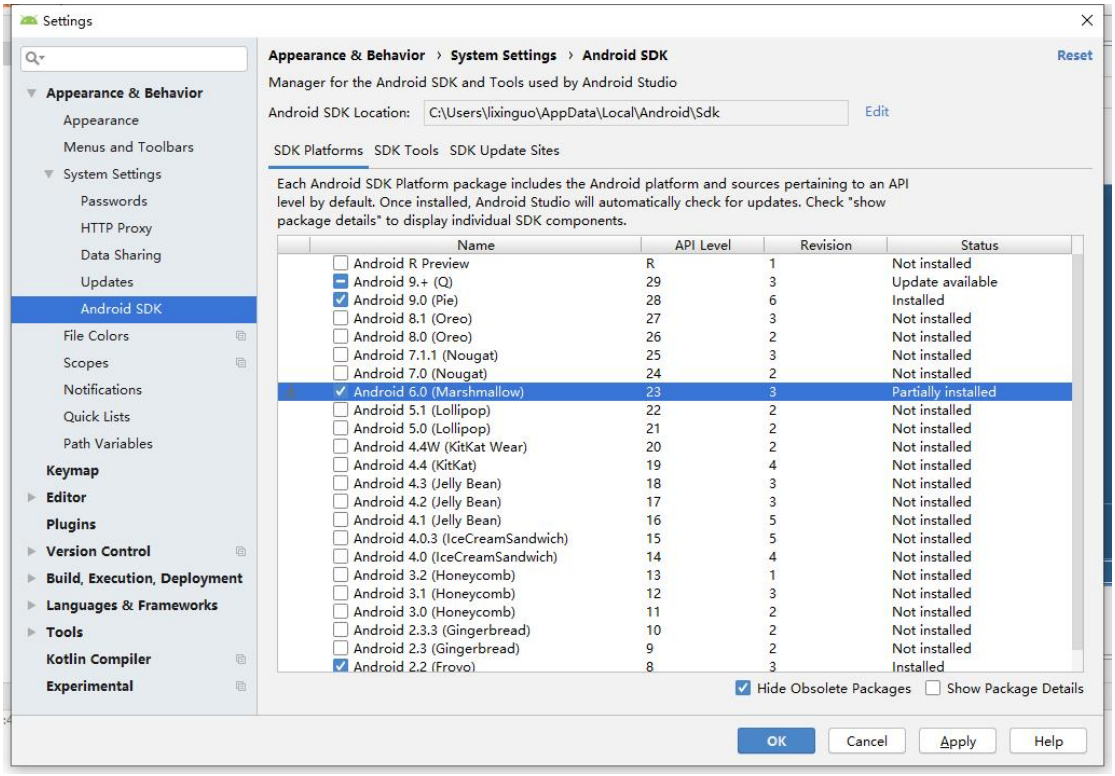
点击“Finish”



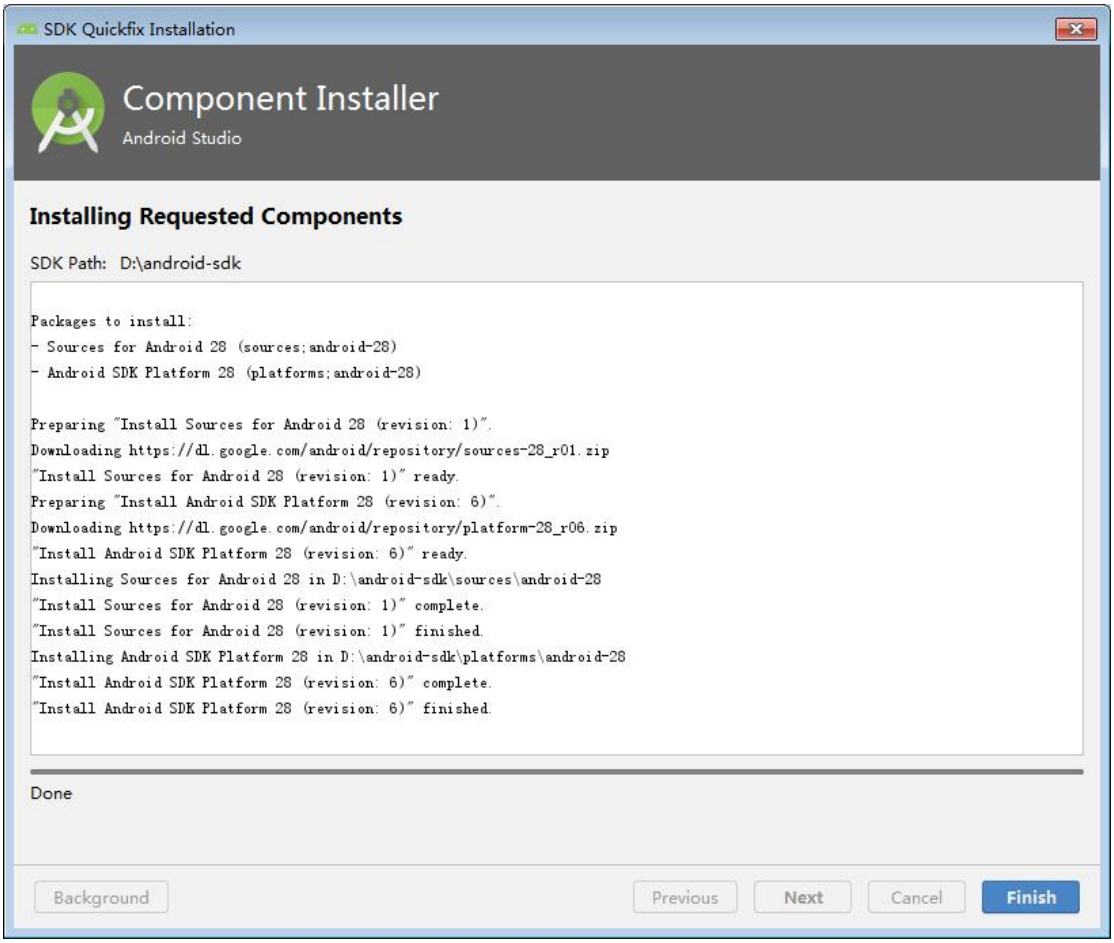
初次使用时会下载 Gradle 等工具，请耐心等待。

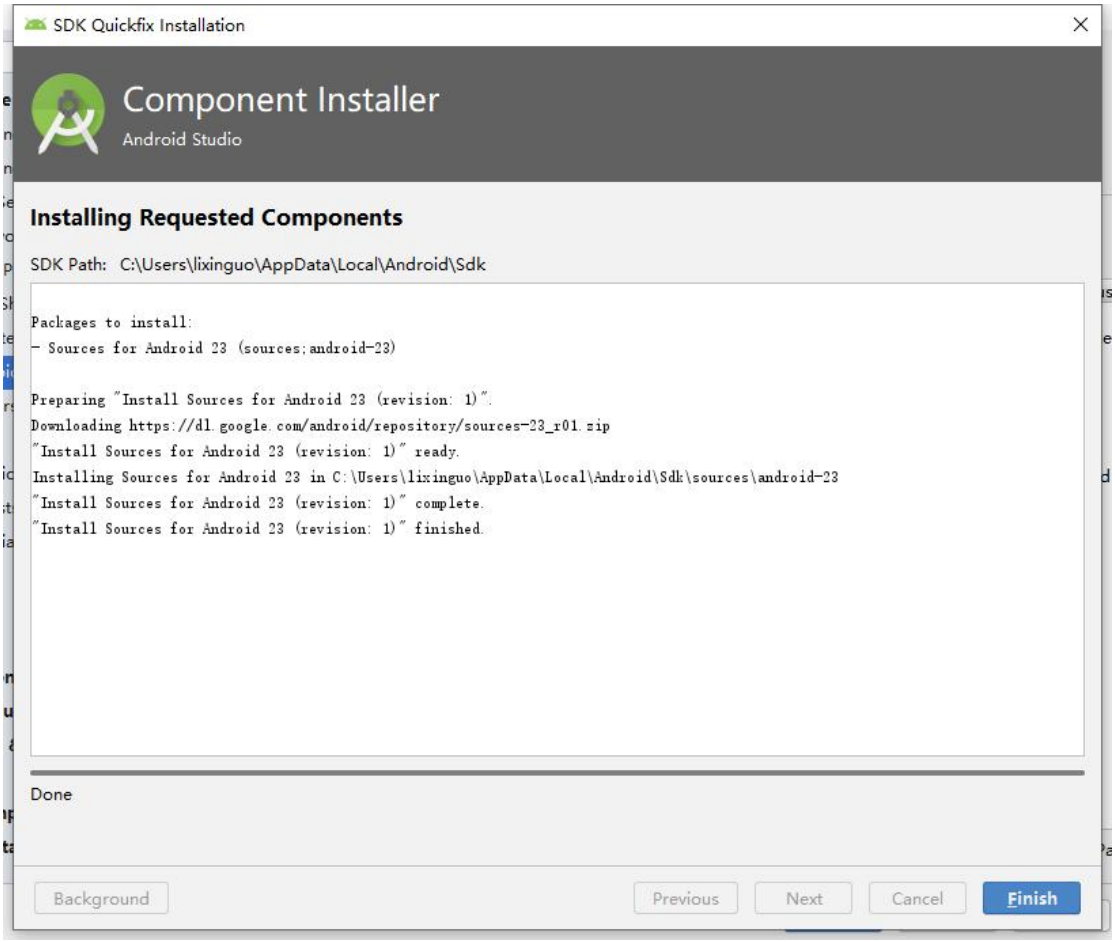
5. 安装 android 9.0 SDK

点击“File”->“settings” 搜索 SDK 打开下图所示界面：

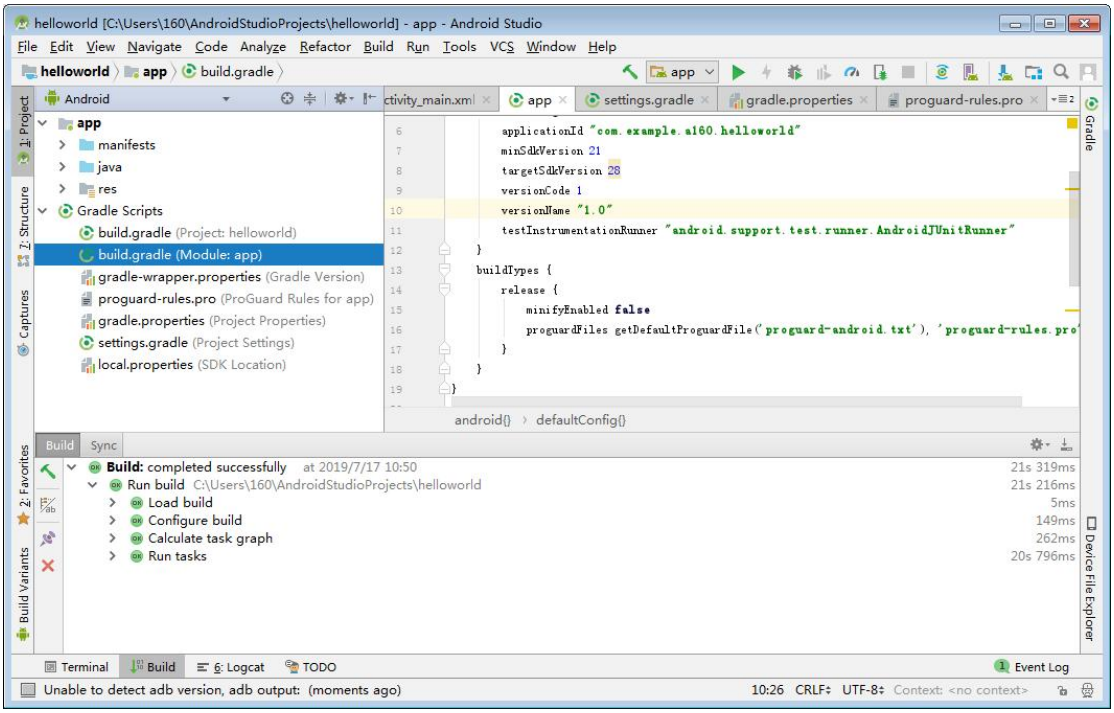


勾选“API23”点击“OK”进行安装。





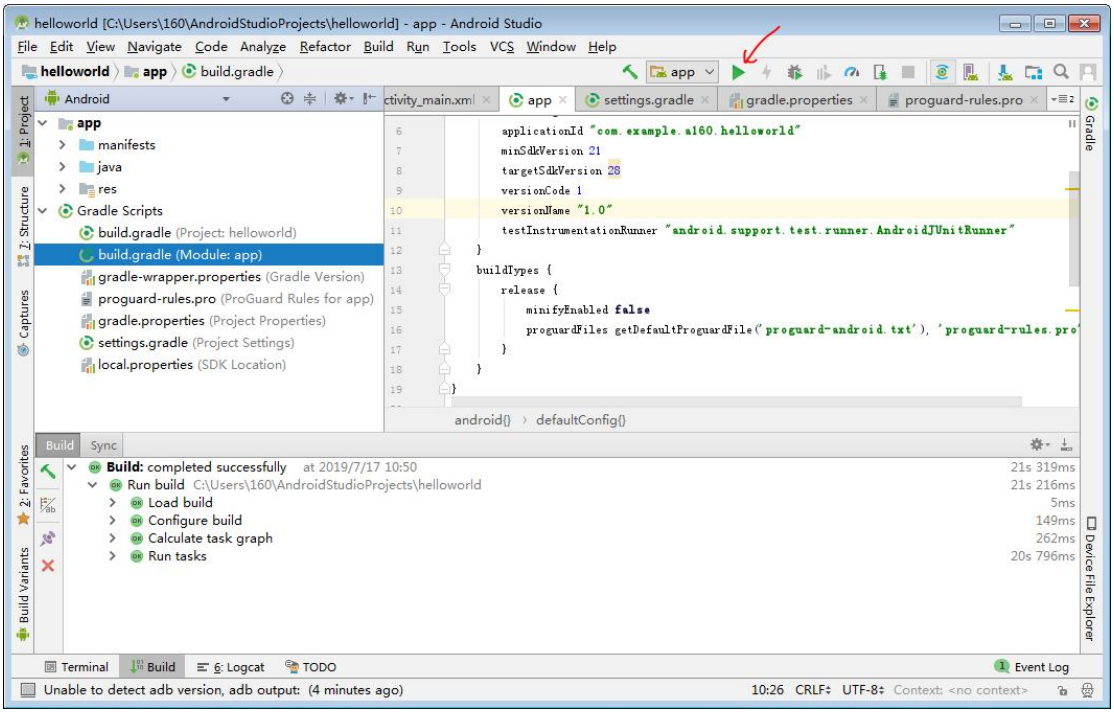
6. 编译

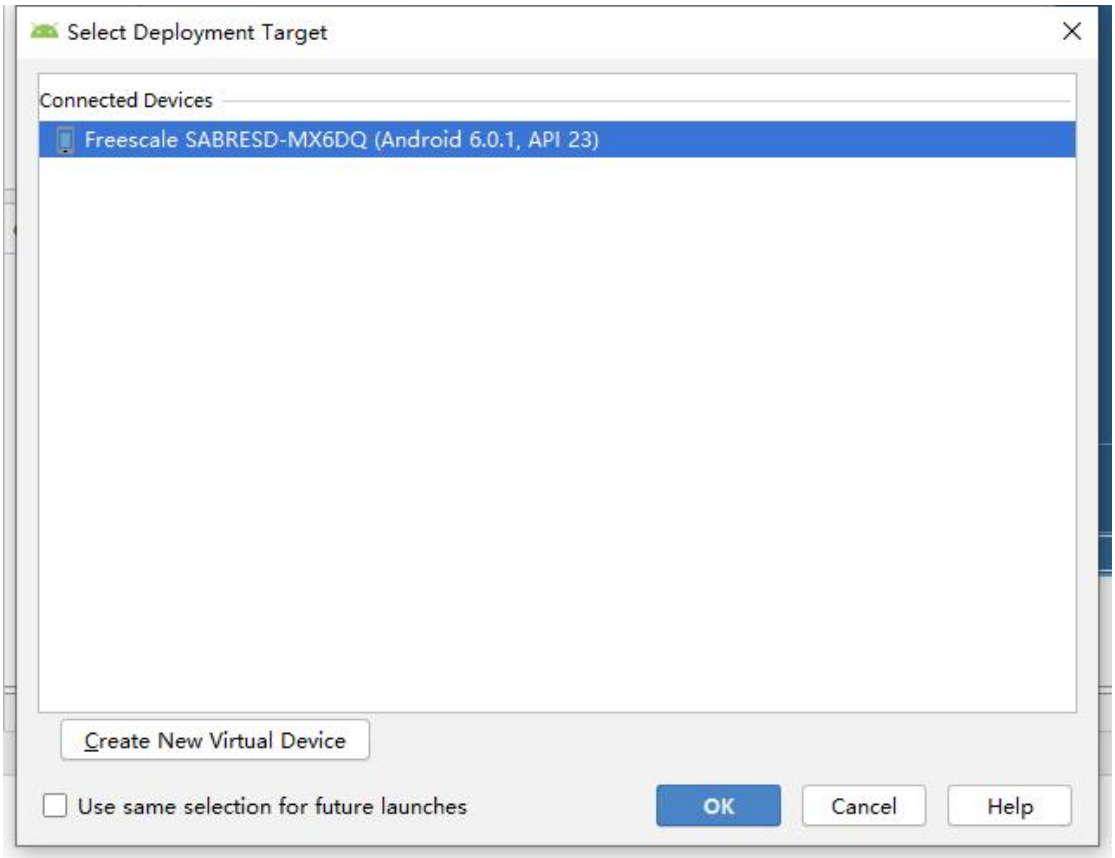


点击“build”->“Make Project”重新编译。

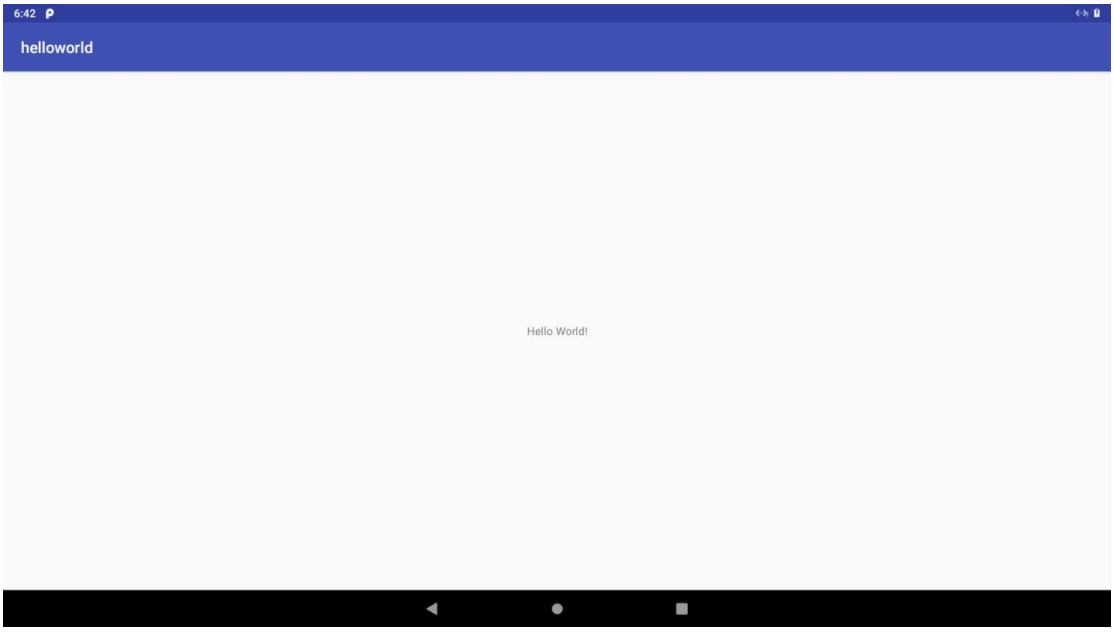
7. 运行

编译完成后，将 OTG 线链接到 PC 的 USB 接口中，点击菜单栏中的绿色三角形图标。





点击“OK”，稍等一会程序即在开发板上运行起来。



4.2 Apk platform 签名

Android 平台中 SELinux 将 App 划分为三种,包括没有 platform 签名和系统权限的 untrusted_app，拥有 platform 签名没有系统权限的 platform_app,和拥有 platform 签名和系统权限的 system_app。本节，将介绍如何给 apk 签名获得 system 权限。

1. 制作签名文件

将 Android 系统中 device/fsl/common/security/platform.x509.pem 以及 device/fsl/common/security/platform.pk8 拷贝到 windows 中。

打开命令行窗口执行：

```
openssl pkcs8 -in platform.pk8 -inform DER -outform PEM -out shared.priv.pem
-nocrypt

openssl pkcs12 -export -in platform.x509.pem -inkey shared.priv.pem \
-out shared.pk12 -name androiddebugkey

keytool -importkeystore -deststorepass android -destkeypass android \
-destkeystore debug.keystore -srckeystore shared.pk12 -srcstoretype PKCS12 \
-srcstorepass android -alias androiddebugkey
```

其中 key-alias 以及 password 您可以根据需求修改为其它内容。将签名文件 debug.keystore 文件保存到您的常用目录中。

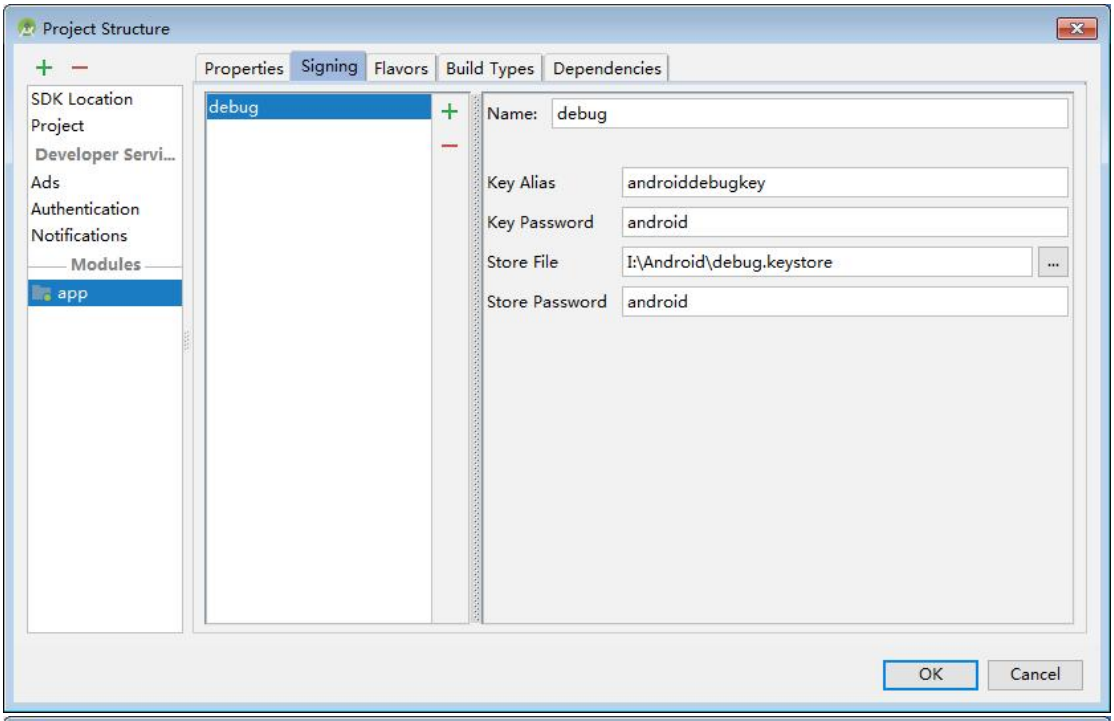
注意：如果您的 windows 系统中没有 openssl 命令，请前往 http://slproweb.com/products/Win32OpenSSL.html 下载安装，并设置环境变量。

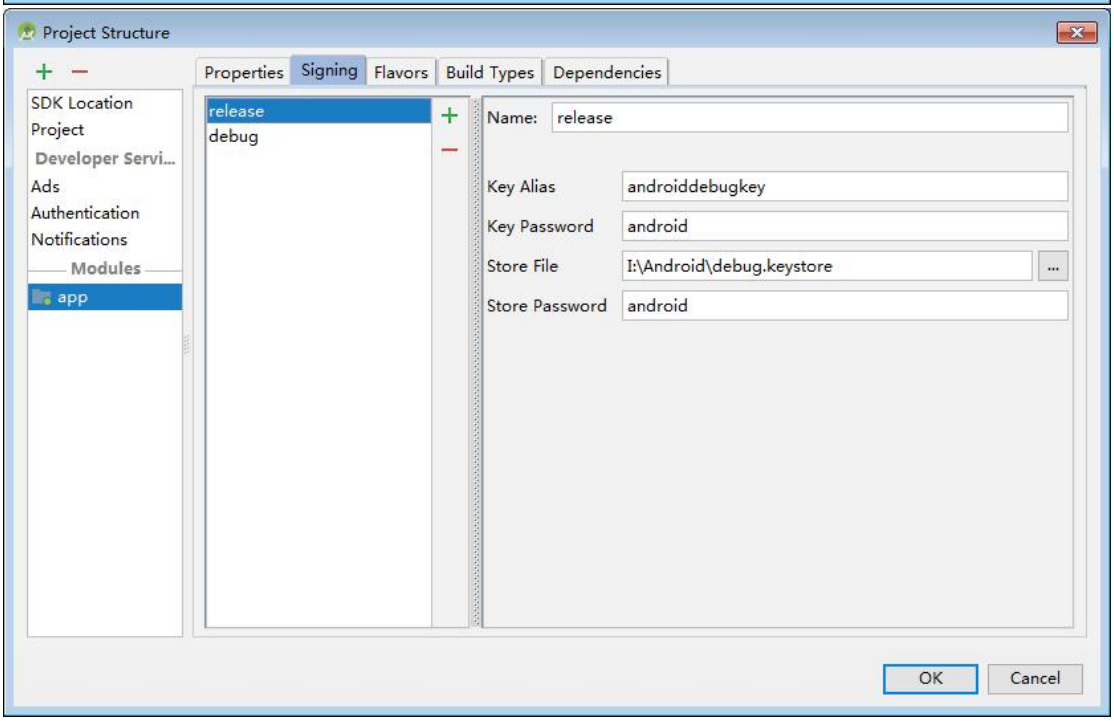
2. 设置 andorid studio

打开任意 android studio 工程，在 AndroidManifest.xml 中添加共享 UID 例如：

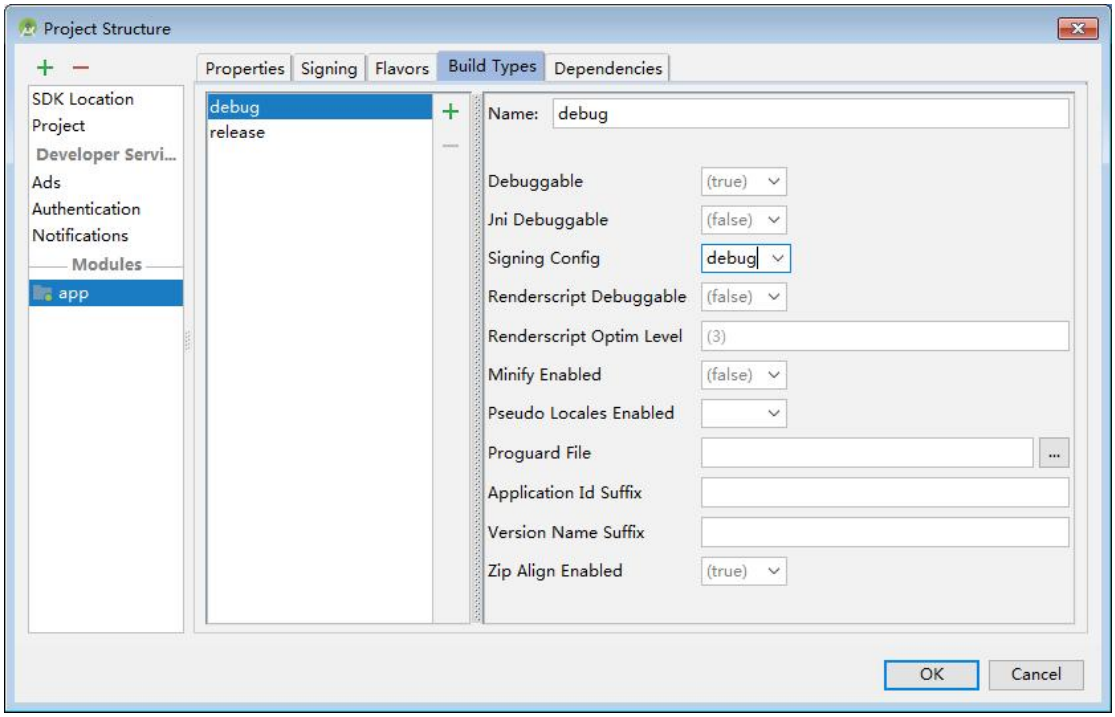
```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.forlinx.serialporttest"
    android:sharedUserId="android.uid.system" >
```

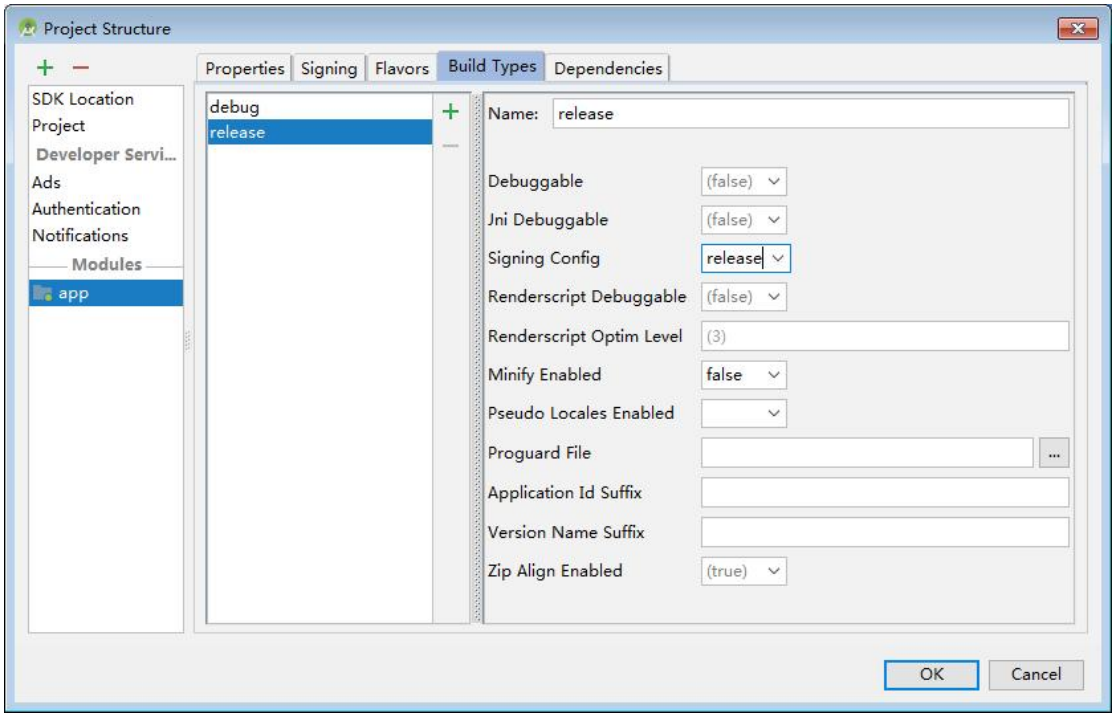
点击“File”->“project structure”：





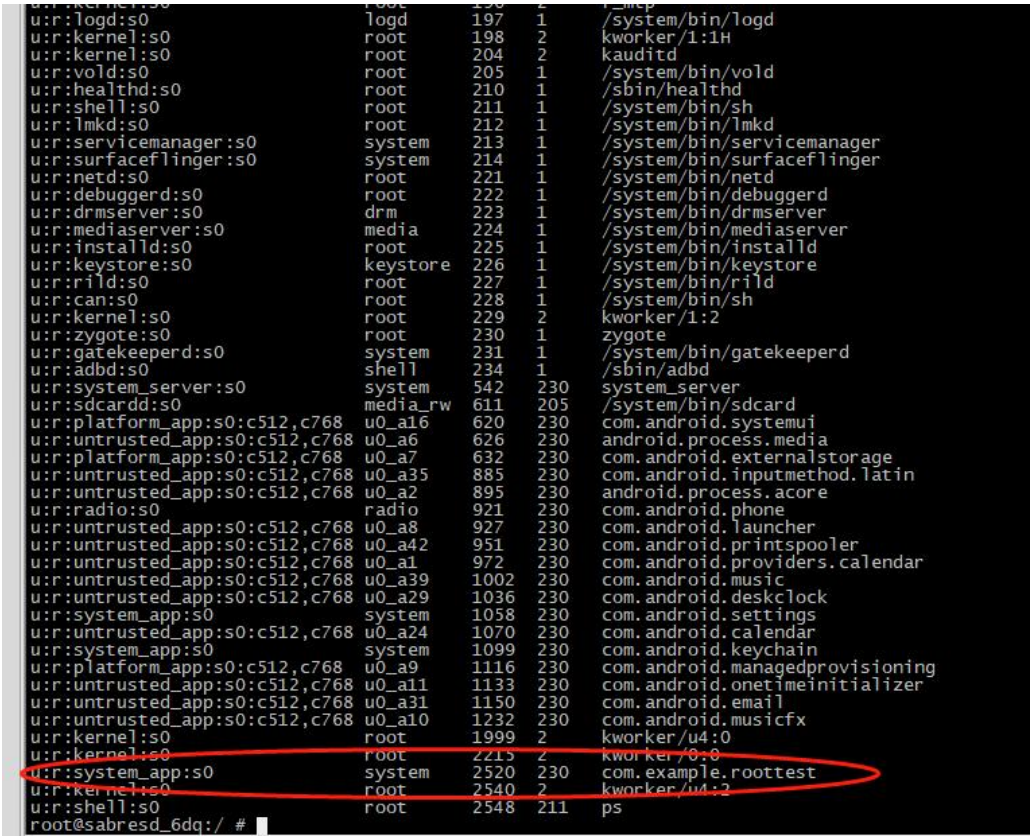
添加 debug 版本和 release 版本的签名配置，并点击“build types”。





如图设置对应版本的签名配置。

点击 android studio 的运行按钮，启动 app。在串口中输入 ps -Z



确认你的 app 是否已经成为 system_app。

4.3 系统预装 Apk 的方法

1. 在 android 系统中新建目录：

```
mkdir packages/apps/serialporttest
```

将需要预装的 apk(无需签名)拷贝到目录中以 serialporttest.apk 为例：

```
cp serialporttest.apk packages/apps/serialporttest
```

2. 在 packages/apps/serialporttest 新建 Android.mk

```
LOCAL_PATH := $(call my-dir)
include $(CLEAR_VARS)
LOCAL_MODULE := serialporttest
LOCAL_SRC_FILES := serialporttest.apk
LOCAL_MODULE_CLASS := APPS
LOCAL_MODULE_SUFFIX := .apk
LOCAL_BUILT_MODULE_STEM := package.apk
LOCAL_CERTIFICATE := platform
LOCAL_DEX_PREOPT := false
LOCAL_PRIVILEGED_MODULE := true
include $(BUILD_PREBUILT)
```

3. 同时修改 device/fsl/imx8m/evk_8mm/evk_8mm.mk 添加

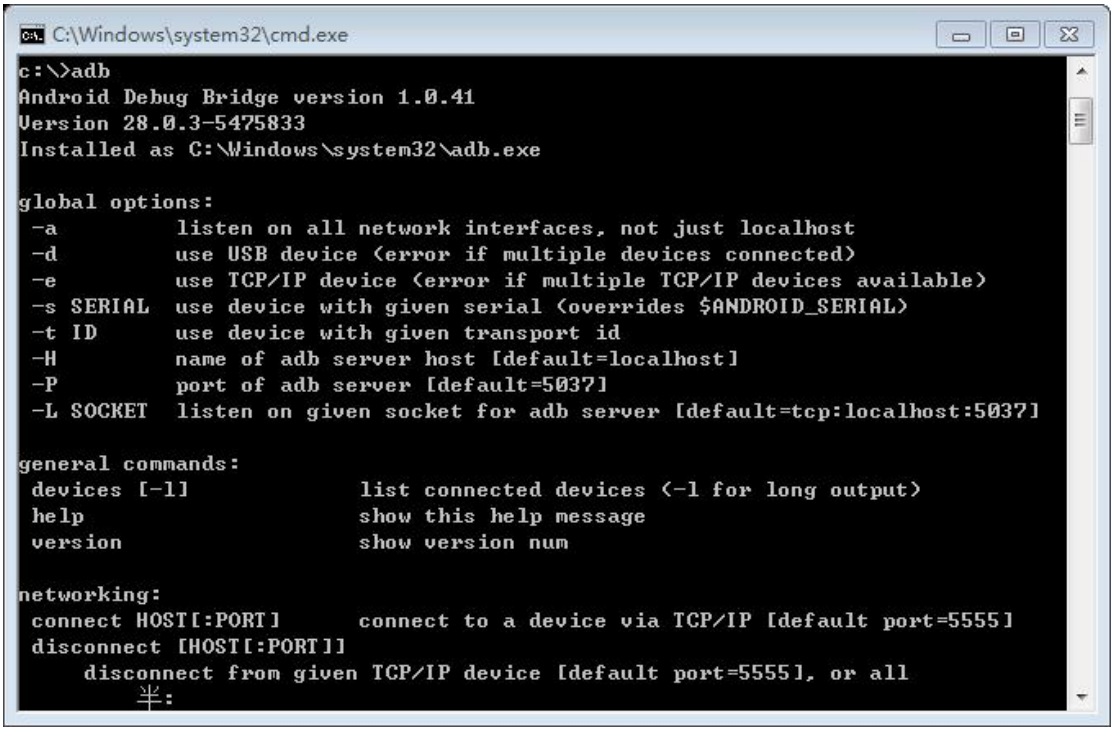
```
PRODUCT_PACKAGES += \
Serialporttest
```

4. 重新编译镜像。

4.4 ADB 安装

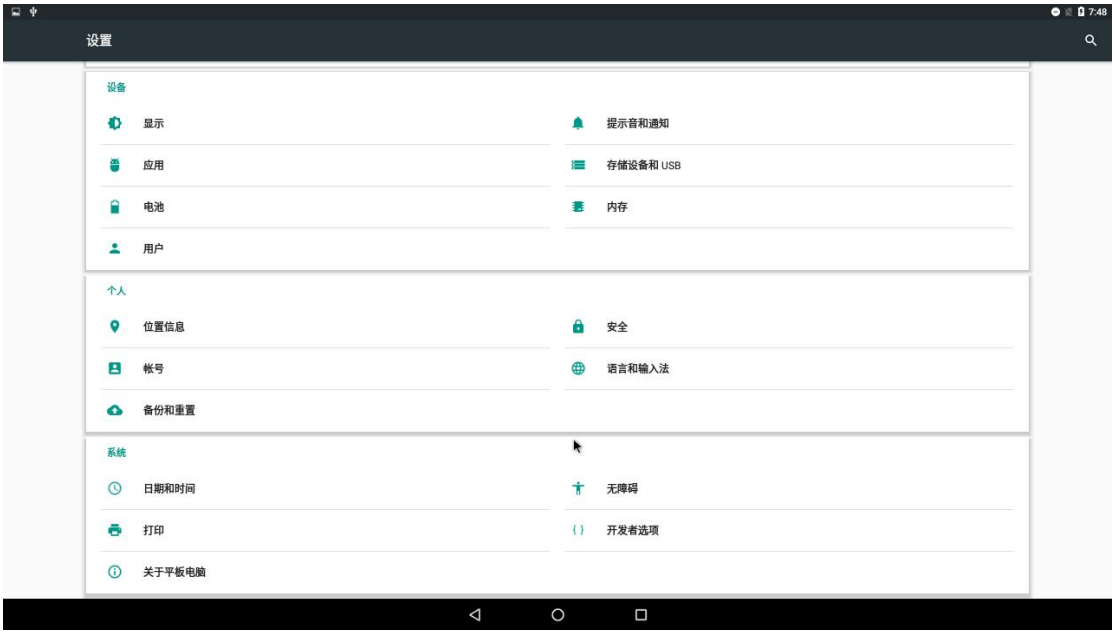
将用户资料中工具目录的 platform-tools_r28.0.3-windows 中的文件解压到 C:\Windows\System32 目录，如果是 64 位系统请解压到 C:\Windows\SysWOW64。

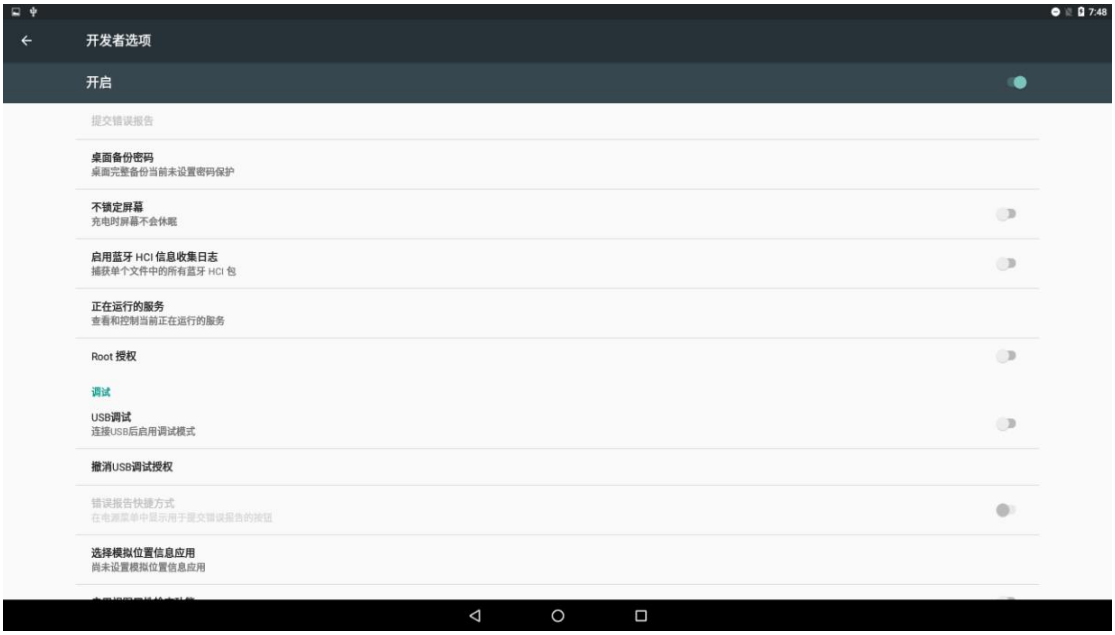
通过点击开始菜单，在开始菜单下方的搜索框中输入 cmd ，在 cmd.exe 上按回车来启动 DOS 窗口，在 DOS 窗口中输入 “adb” 确认是否安装成功。

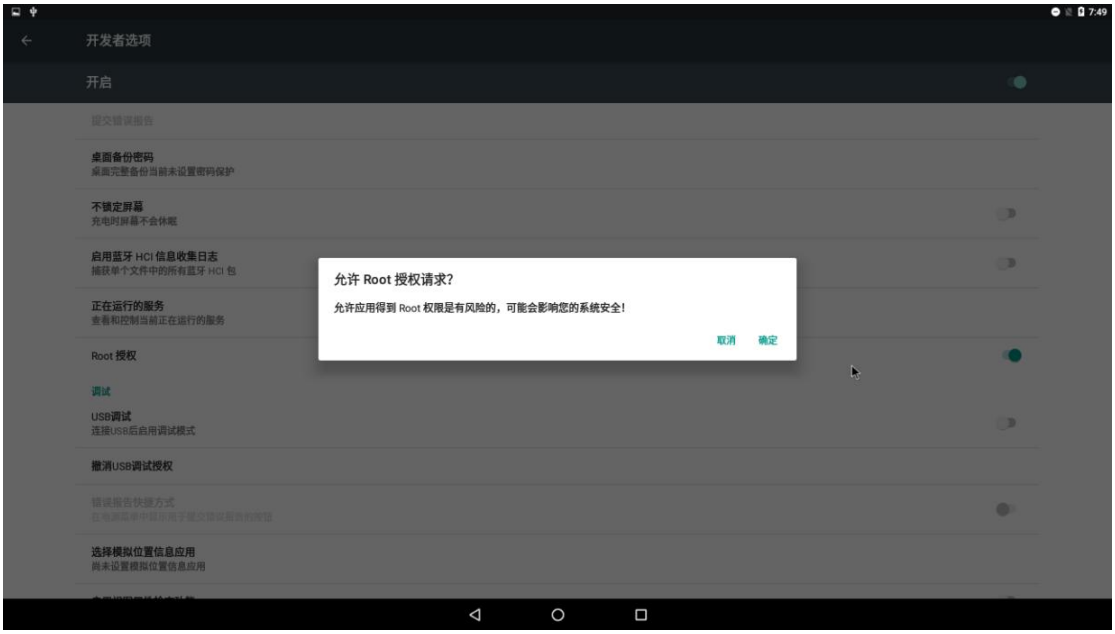

C:\Windows\system32\cmd.exe
c:\>adb
Android Debug Bridge version 1.0.41
Version 28.0.3-5475833
Installed as C:\Windows\system32\adb.exe
global options:
-a listen on all network interfaces, not just localhost
-d use USB device (error if multiple devices connected)
-e use TCP/IP device (error if multiple TCP/IP devices available)
-s SERIAL use device with given serial (overrides $ANDROID_SERIAL)
-t ID use device with given transport id
-H name of adb server host [default=localhost]
-P port of adb server [default=5037]
-L SOCKET listen on given socket for adb server [default=tcp:localhost:5037]
general commands:
devices [-l] list connected devices (-l for long output)
help show this help message
version show version num
networking:
connect HOST[:PORT] connect to a device via TCP/IP [default port=5555]
disconnect [HOST[:PORT]]
disconnect from given TCP/IP device [default port=5555], or all
半:

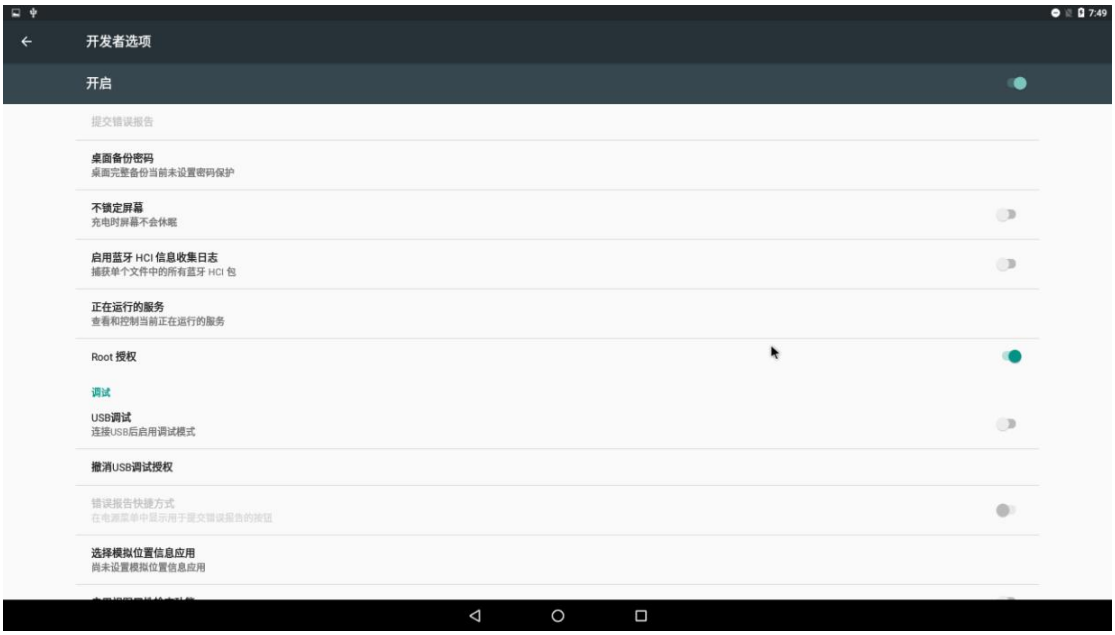
附录五 Root 授权

1. 进入开发者选项，打开 root 授权，打开开发者模式，参考附录三





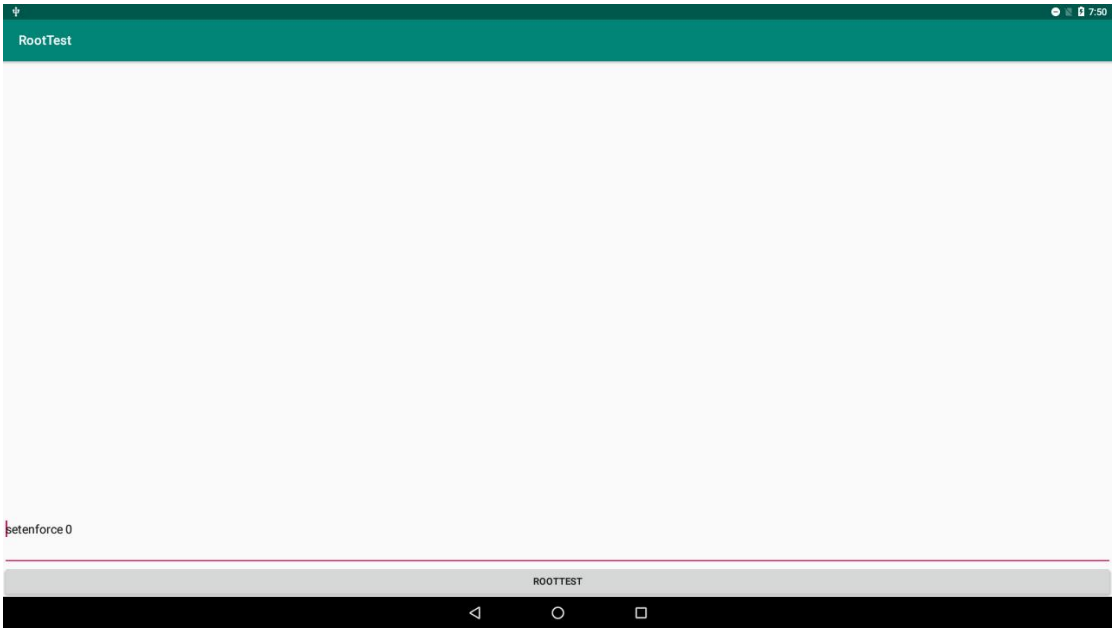


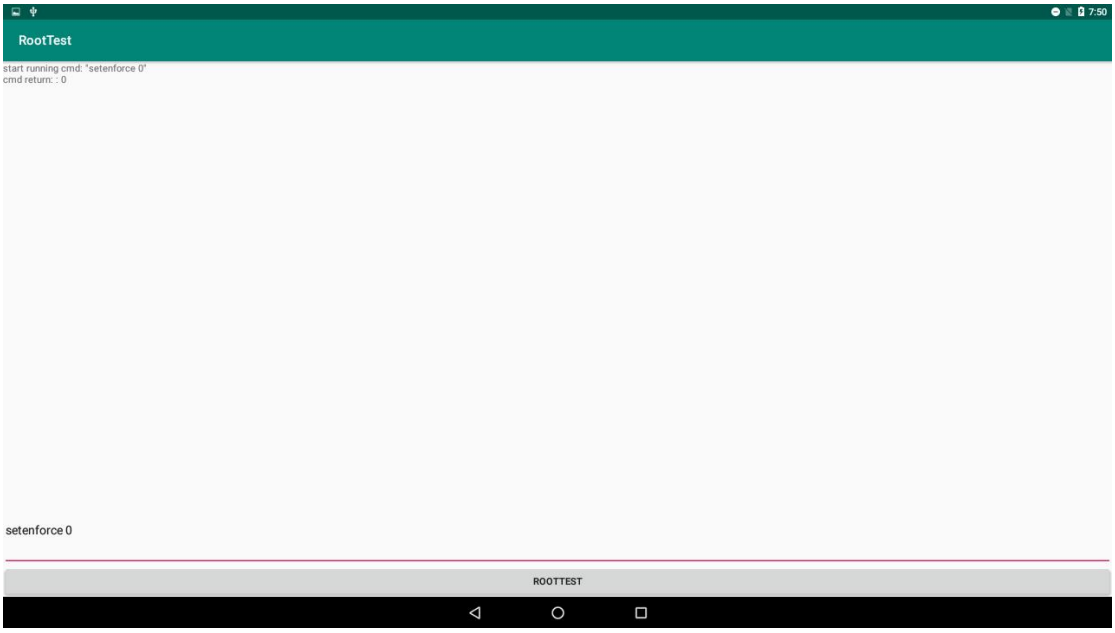


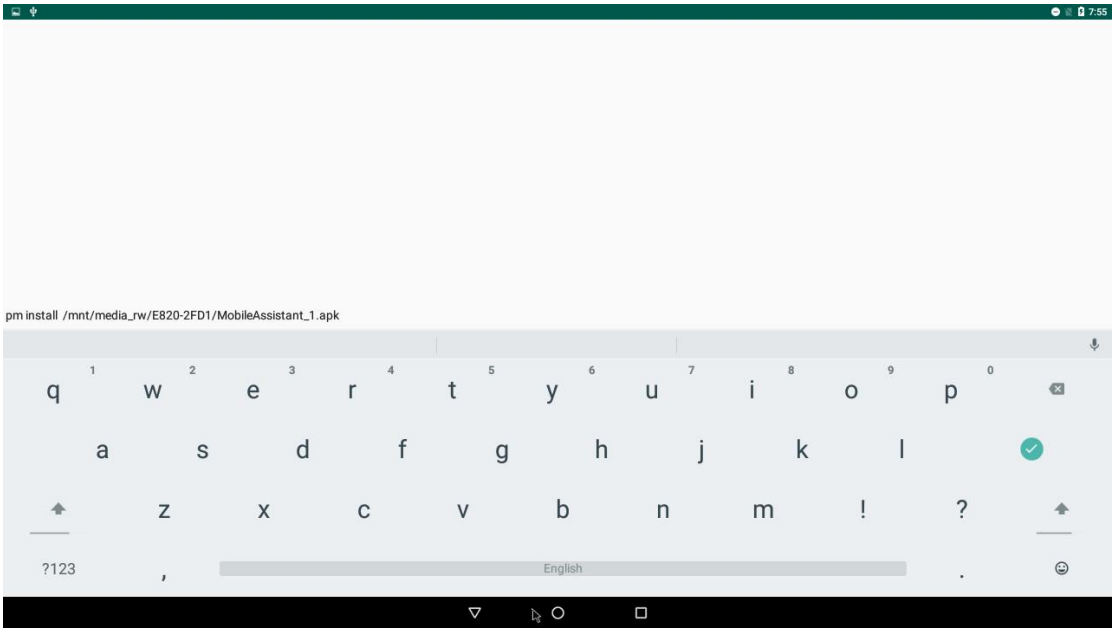
2. 测试，修改完成配置后需要重启才能生效。

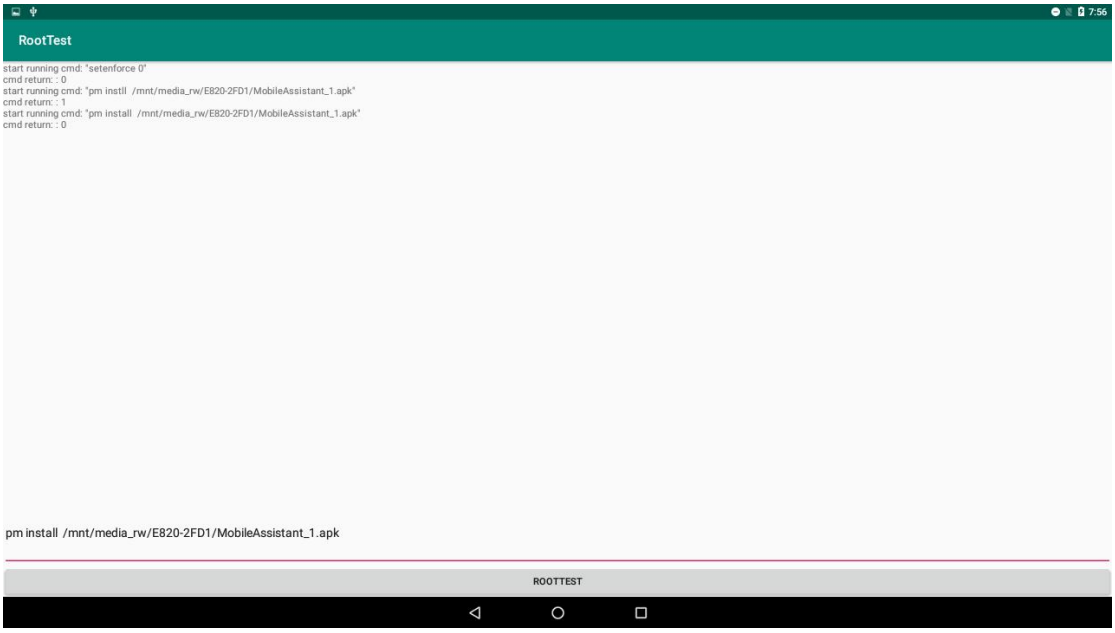
获取 root 权限时，只获取了 root 权限，执行命令时，还是会受到 selinux 权限限制，在因为 selinux 导致命令不能执行时，可以先暂时关闭 selinux 限制，执行完后在开启。setenforce 0 关闭和 setenfoce 1 打开。

打开测试程序“RootTest”，在文本框输入测试命令，点击“ROOTTEST”执行命令，等待返回结果。返回结果为该指令的运行结果，一般 0 为执行成功。


7:50
RootTest
setenforce 0
ROOTTEST


7:50
RootTest
start running cmd: "setenforce 0"
cmd return: : 0
setenforce 0
ROOTTEST


7:55
pm install /mnt/media_rw/E820-2FD1/MobileAssistant_1.apk
English


7:56
RootTest
start running cmd: "setenforce 0"
cmd return: : 0
start running cmd: "pm instll /mnt/media_rw/E820-2FD1/MobileAssistant_1.apk"
cmd return: : 1
start running cmd: "pm install /mnt/media_rw/E820-2FD1/MobileAssistant_1.apk"
cmd return: : 0
pm install /mnt/media_rw/E820-2FD1/MobileAssistant_1.apk
ROOTTEST