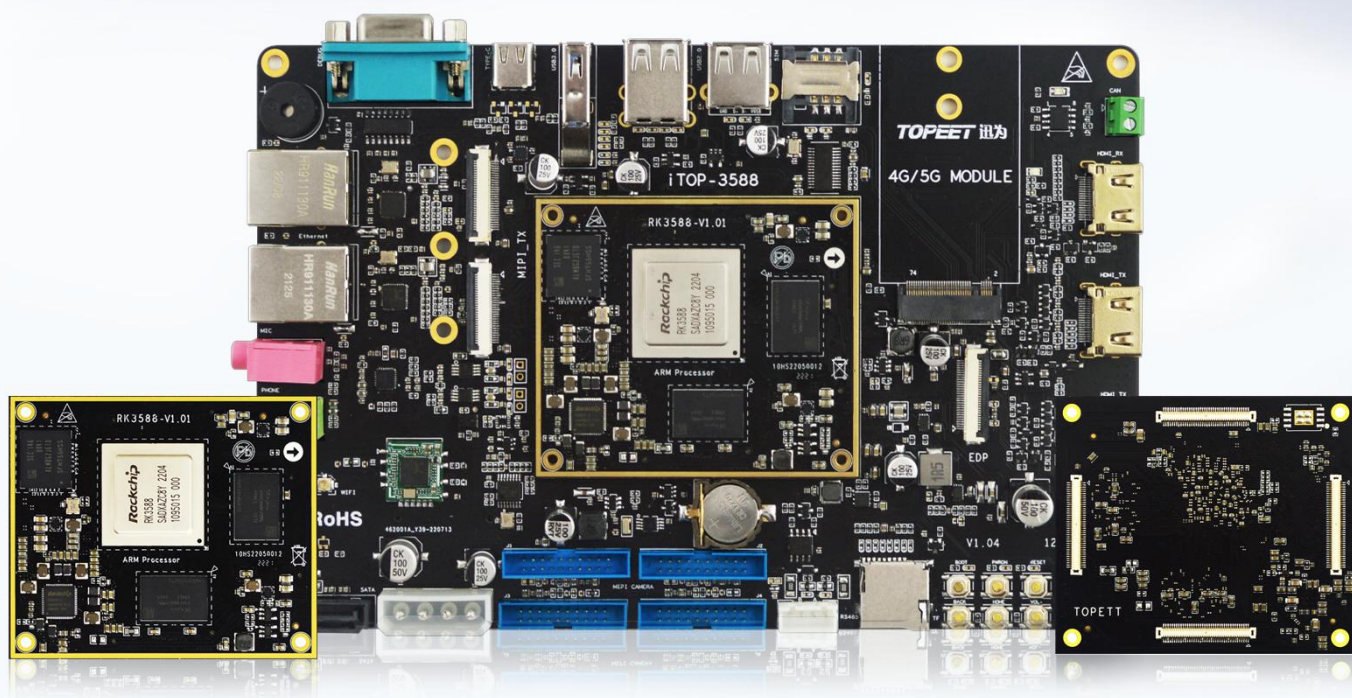


强大的 AI 能力 更快更强

超长供货周期 | 7X24 小时稳定运行 | 8K 视频编解码



iTOP-RK3588 开发板使用手册

八核 64 位 CPU | 主频 2.4GHz | NPU 算力 6T | 4800 安防级别 ISP

更新记录

更新版本	修改内容
V1.0	初版
V1.1	更正一些细节

目录

更新记录	2
目录	3
版权声明	4
更多帮助	5
第 1 章 安装 adb 工具	6
1.1 adb 简介	6
1.2 Windows 下安装 ADB 工具	6
1.3 Ubuntu 下安装 ADB 工具	9
第 2 章 使用 adb 工具	10
2.1 Android 系统允许 usb 调试	10
2.2 安装 adb 驱动	15
2.3 查看帮助信息	17
2.4 查看当前连接设备	18
2.5 连接设备	19
2.5.1 usb 连接 adb 方式	19
2.5.2 网络 adb 方式	20
2.6 运行 shell	23
2.7 文件传输	24
2.8 应用相关	25
2.9 获取 root 权限	25
2.10 重新挂载 remount	25
2.11 其他命令	25

版权声明

本文档版权归北京迅为电子有限公司所有。未经本公司书面许可，任何单位和个人无权以任何形式复制、传播、转载本文档的任何内容，违者将被追究法律责任。

更多帮助

注意事项与维护

- ❖ 请注意和遵循标注在产品上的所有警示和指引信息；
- ❖ 请勿带电插拔核心板及外围模块；
- ❖ 使用产品之前，请仔细阅读本手册，并妥善保管，以备将来参考；
- ❖ 请使用配套电源适配器，以保证电压、电流的稳定；
- ❖ 请勿在冷热交替环境中使用本产品，避免结露损坏元器件；
- ❖ 请保持产品干燥，如果不慎被任何液体泼溅或浸润，请立刻断电并充分晾干；
- ❖ 请勿使用有机溶剂或腐蚀性液体清洗本产品；
- ❖ 请勿在多尘、脏乱的环境中使用本产品，如果长期不使用，请包装好本产品；
- ❖ 如果在震动场景使用，请做好核心板与底板的固定，避免核心板跌落损坏；
- ❖ 请勿在通电情况下，插拔核心板及外围模块(特别是串口模块)；
- ❖ 请勿自行维修、拆解本产品，如产品出现故障应及时联系本公司进行维修；
- ❖ 请勿自行修改或使用未经授权的配件，由此造成的损坏将不予保修；

资料的更新

为了确保您的资料是最新状态，请密切关注我们的动态，我们将会通过微信公众号和 QQ 群推送。

关注“迅为电子”微信公众号，不定期分享教程、资料和行业干货及产品一线资料。

迅为新媒体账号

官网: <https://www.topeetboard.com>

知乎 <https://www.zhihu.com/people/topeetabc123>

CSDN: <https://blog.csdn.net/BeiJingXunWei>



售后服务政策

1. 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
2. 服务政策：参见官方网售后服务说明
<https://www.topeetboard.com/sydyml/Service/bx.html>

送修地址：

1. 地址：北京市海淀区永翔北路 9 号中国航发大厦三层
2. 联系人：迅为开发板售后服务部
3. 电话：010-85270716
4. 邮编：100094
5. 邮寄须知：建议使用顺丰、圆通或韵达，且不接受任何到付

技术支持范围

1. 了解产品的软、硬件资源提供情况咨询
2. 产品的软、硬件手册使用过程中遇到的问题
3. 下载和烧写更新系统过程中遇到的问题
4. 产品用户的资料丢失、更新后重新获取
5. 产品的故障判断及售后维修服务。

PS：（由于嵌入式系统知识范围广泛，我们无法保证对各种问题都能一一解答，部分内容无法供技术支持，只能提供建议。）

技术支持

1. 周一至周五：（法定节假日除外）
上午 9:00 ~ 11:30 / 下午 13:30 ~ 17:30
2. QQ 技术交流群：
824412014
822183461
95631883
861311530

第 1 章 安装 adb 工具

本章节将向用户介绍 adb 工具在 Windows 系统以及 Ubuntu 系统的安装。

1.1 adb 简介

ADB 英文名叫 Android debug bridge，翻译过来就是 Android 调试桥，是 Android SDK 里面的一个通用的命令行工具。Adb 是 Android 设备和电脑端连接的一个桥梁，它可以让开发者通过网络或 USB 和 Android 设备进行通信，所以说 ADB 是 Android 设备的调试工具。

虽然 ADB 是 Android 设备的调试工具，但是随着 adb 的普及，很多 Linux 设备也支持了 ADB 调试，比如瑞芯微平台的 Linux 设备。注！iTOP-RK3568 开发板的 Android 和 Linux 系统都支持 ADB。

这个工具可以操作管理 Android 模拟器或者真实的 Android 设备，主要的功能如下所示：

- 在 Android 设备上运行 shell 终端，用命令行操作
- 管理模拟器或设备的端口映射
- 电脑和设备之间相互拷贝文件
- APK 调试和安装

1.2 Windows 下安装 ADB 工具

开发板资料包中已经 adb 工具的安装包，在网盘资料“**iTOP-3588 开发板\02_【iTOP-RK3588 开发板】开发资料\07_Android 系统开发配套资料\01_ADB 工具配套资料**”路径下。将 ADB 工具拷贝到 Windows 的某个目录。**注意！！！ adb 工具在 Windows 上的路径必须是英文路径，不能含有中文。**

打开 adb 工具下的 cmd.exe 即可打开 adb 工具，adb 命令都是在 cmd.exe 中输入的。如下图所示：



SuperSU-v2.79-201612051815	2022/5/18 18:05	文件夹	
adb.exe	2015/4/25 12:16	应用程序	988 KB
AdbWinApi.dll	2015/4/25 12:16	应用程序扩展	94 KB
AdbWinUsbApi.dll	2015/4/25 12:16	应用程序扩展	60 KB
cmd.exe	2021/8/19 12:19	应用程序	274 KB
rootchecker.paid-2.1.apk	2017/3/28 11:19	APK 文件	266 KB

用户也可以通过 <https://developer.android.google.cn/studio/releases/platform-tools> 地址自行下载 ADB 工具，如下图所示：

SDK 平台工具版本说明

Android SDK Platform-Tools 是 Android SDK 的一个组件。它包含与 Android 平台进行交互的工具，主要是 `adb` 和 `fastboot`。虽然 `adb` 是 Android 应用开发所必需的，但应用开发者通常仅使用 Studio 安装的副本。如果您想直接从命令行使用 `adb` 并且未安装 Studio，此项下载非常有用。（如果您安装了 Studio，建议您仅使用 Studio 安装的副本，因为 Studio 会自动对其进行更新。）如果您想要解锁设备的引导加载程序并为其刷入新的系统映像，则需要 `fastboot`。该软件包曾经包含 `systrace`，但已由 Studio 性能分析器、`gpuinspector.dev` 或 `Perfetto` 取代。

虽然 `adb` 和 `fastboot` 中的某些新功能仅适用于较新的 Android 版本，但它们是向后兼容的，因此您只需使用最新版本的 SDK 平台工具即可。如果您发现异常，请提交 bug 报告。

下载

如果您是 Android 开发者，则应从 Android Studio 的 [SDK 管理器](#)或通过 `sdkmanager` 命令行工具获取最新的 SDK Platform-Tools。这样可确保这些工具能够与其他 Android SDK 工具一起保存到正确的位置，并可轻松地进行更新。

但是，如果您只想使用这些命令行工具，请访问以下链接：[点击对应的链接，即可进行下载](#)

- [下载适用于 Windows 的 SDK Platform-Tools](#) **Windows系统的ADB工具**
- [下载适用于 Mac 的 SDK Platform-Tools](#) **Mac系统的ADB工具**
- [下载适用于 Linux 的 SDK Platform-Tools](#) **Linux系统的ADB工具**

虽然这些链接不会发生变化，但它们始终指向最新版本的工具。

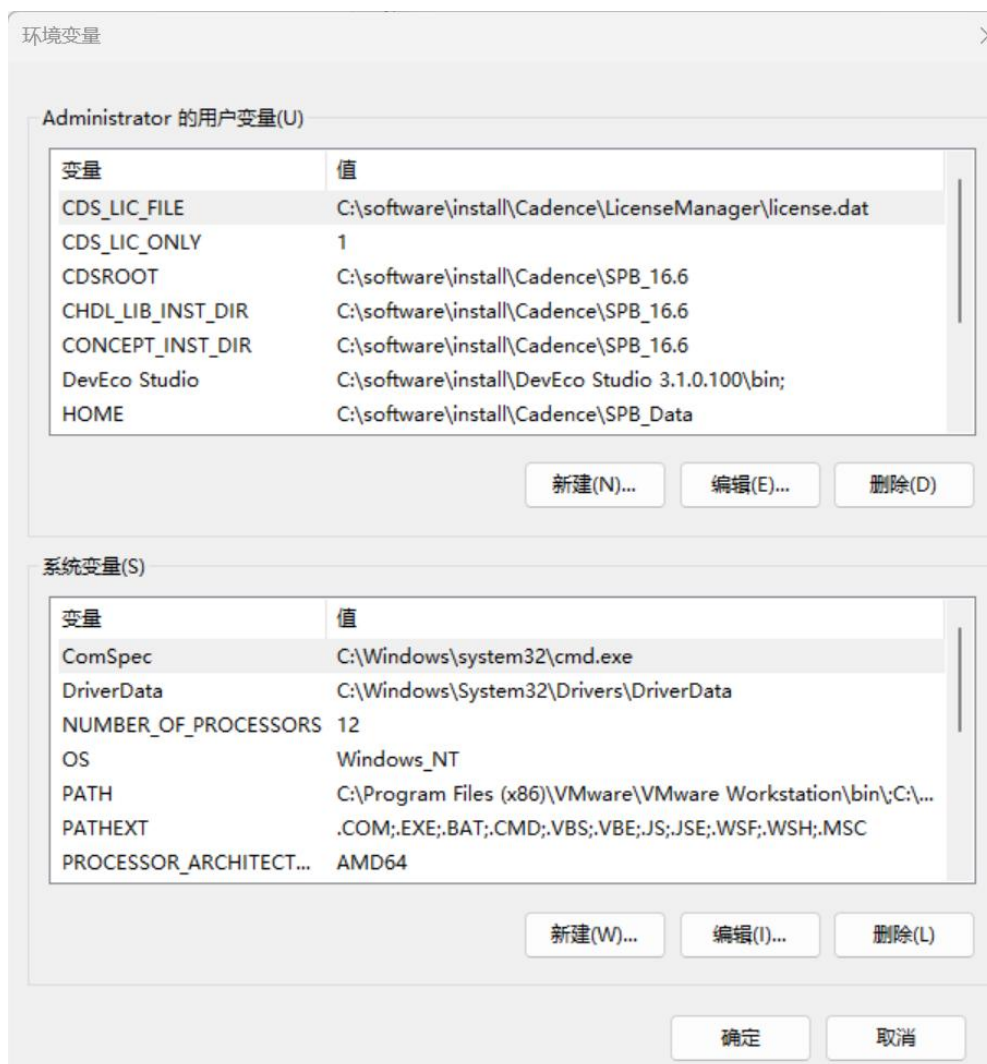
接下来需要将 ADB 工具的安装目录添加到 Windows 系统的环境变量中。在 Windows 11 中，你可以通过以下步骤设置环境变量：

1 打开“控制面板”：点击任务栏上的“开始”按钮，然后在弹出的菜单中选择“控制面板”。

2 进入“系统”：在控制面板窗口中，找到并点击“系统”选项。在“系统”页面中，点击“高级系统设置”，如下图所示：



3 在“系统属性”对话框中，点击“环境变量”按钮。如下图所示：

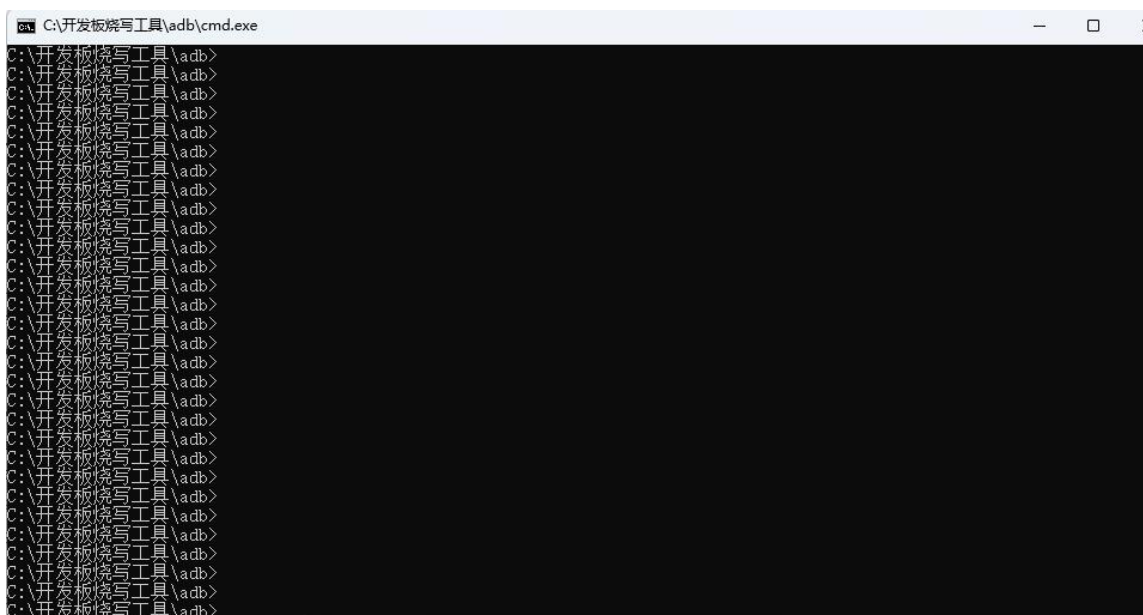


4 在“环境变量”对话框中，你会看到两个部分：用户变量和系统变量。如果想为当前用户设置环境变量，选择“用户变量”部分；如果想为整个系统设置环境变量，选择“系统变量”部分。

5 在选择用户变量或系统变量后，可以点击“新建”按钮来添加一个新的环境变量，需要提供环境变量的名称和值，我们可以新建一个环境变量，然后输入 `adb` 的安装目录，并点击“确定”按钮来保存。

6 完成设置：完成添加或编辑环境变量后，你可以点击“确定”按钮来关闭对话框。

至此，`adb` 安装完成，我们按住“Win+R”组合键打开 Windows 运行窗口，输入“`cmd`”按回车键打开 Windows 的命令窗口，然后会打开如下所示：



1.3 Ubuntu 下安装 ADB 工具

在虚拟机 Ubuntu 使用以下命令安装 ADB，如下图所示：

```
sudo apt install adb
```

```
topeet@ubuntu:~/Linux/rk356x_linux_module$ sudo apt-get install adb
[sudo] topeet 的密码:
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
将会同时安装下列软件:
  android-libadb android-libboringssl android-libcrypto-utils
  android-sdk-platform-tools-common
下列【新】软件包将被安装:
  adb android-libadb android-libboringssl android-libcrypto-utils
  android-sdk-platform-tools-common
升级了 0 个软件包, 新安装了 5 个软件包, 要卸载 0 个软件包, 有 53 个软件包未被升级。
需要下载 764 kB 的归档。
解压缩后会消耗 2,366 kB 的额外空间。
您希望继续执行吗? [Y/n]
```

接下来进行测试, 执行如下命令, 确认 adb 安装是否成功。

```
adb version
```

```
topeet@ubuntu:~/Linux/rk356x_linux_module$ adb version
Android Debug Bridge version 1.0.39
Version 1:8.1.0+r23-5ubuntu2
Installed as /usr/lib/android-sdk/platform-tools/adb
topeet@ubuntu:~/Linux/rk356x_linux_module$
```

如上图所示, 执行命令能成功获取到 adb 版本信息, 说明 adb 安装成功。

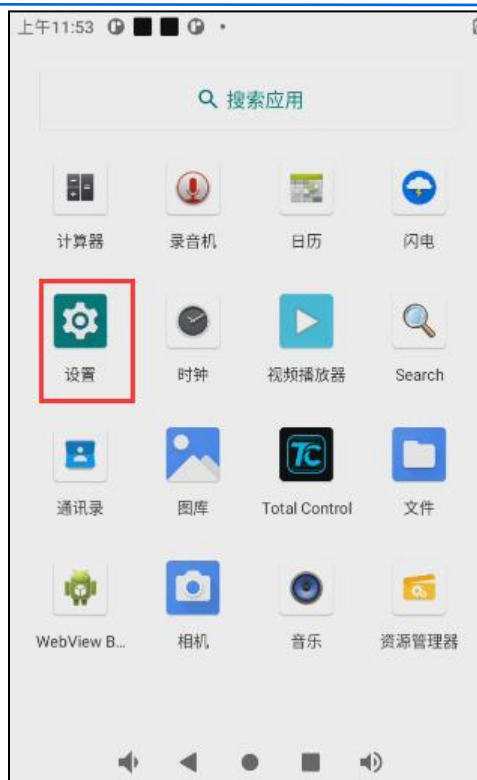
第 2 章 使用 adb 工具

2.1 Android 系统允许 usb 调试

如果开发板烧写的是 Android 系统, 要设置允许 USB 调试。如果开发板烧写的是 Linux 系统, 不需要操作本章节。

首先开发板烧写 Android11 系统, 连接屏幕, 烧写线, 串口线, 电源线。连接完毕后, 接着我们打开 usb 调试。步骤如下:

1. 在屏幕上打开设置 app, 如下图所示:



2 滑到最底部，点击“关于平板电脑”，如下图所示：



3 滑到最底部，找到版本号，连续点击版本号 7 次，便进入了开发者模式。如下图所示：



4 返回，点击“系统”，如下图所示：



5 点击“高级”-》“开发者选项”，如下图所示：



6 然后打开 usb 调试，如下图所示：



7 设置默认 usb 配置，点击“usb 默认配置”，选择“文件传输模式”如下图所示：



8 打开以上设置之后，便可以开始安装 adb 驱动。

2.2 安装 adb 驱动

因为离线安装 adb 驱动失败率很高。这里作者选择在线安装，在线安装 ADB 驱动软件推荐使用“驱动精灵”，使用驱动精灵安装的成功率非常高，但也有概率失败，如果安装失败可以更换其他软件。安装驱动精灵软件的时候，记得把附带的垃圾软件安装选项去掉。

保证电脑和开发板是连接的状态，Android 系统启动成功后并允许 USB 调试，然后用驱动精灵扫描驱动，注意：如果硬件没有连接允许 USB 调试没有打开，驱动精灵是扫描不到的，如下图所示：



检测完成后可看到驱动异常提示，点击安装，会自动安装 adb 驱动。



等待安装，成功后可看到安装成功：



打开设备管理器，如果出现如下图情况，则表明 ADB 安装成功。



2.3 查看帮助信息

在第 1 章内容中,我们讲解了如何在 Windows 和 Ubuntu 系统下安装 adb 工具。在 Windows 和 Ubuntu 系统下,adb 的使用方法是相同的,都是通过执行 adb 命令来使用 adb 工具的。在 Windows 系统中,需要打开 cmd 命令行窗口来执行 adb 命令。

文档以 Windows 系统为例,介绍如何使用 adb 工具。

我们执行如下命令查看 adb 命令的使用帮助信息。

```
adb help
```

```

C:\开发板烧写工具\adb>adb.exe help
Android Debug Bridge version 1.0.29

-d                - directs command to the only connected USB device
                  returns an error if more than one USB device is present.
-e                - directs command to the only running emulator.
                  returns an error if more than one emulator is running.
-s <serial number> - directs command to the USB device or emulator with
                  the given serial number. Overrides ANDROID_SERIAL
                  environment variable.
-p <product name or path> - simple product name like 'sooner', or
                  a relative/absolute path to a product
                  out directory like 'out/target/product/sooner'.
                  If -p is not specified, the ANDROID_PRODUCT_OUT
                  environment variable is used, which must
                  be an absolute path.
devices           - list all connected devices
connect <host>[:<port>] - connect to a device via TCP/IP
                  Port 5555 is used by default if no port number is specified.
disconnect [<host>[:<port>]] - disconnect from a TCP/IP device.
                  Port 5555 is used by default if no port number is specified.
                  Using this command with no additional arguments
                  will disconnect from all connected TCP/IP devices.

device commands:
adb push <local> <remote> - copy file/dir to device
adb pull <remote> [<local>] - copy file/dir from device
adb sync [ <directory> ] - copy host->device only if changed
                        (-l means list but don't copy)
                        (see 'adb help all')
adb shell               - run remote shell interactively
adb shell <command>     - run remote shell command
adb emu <command>       - run emulator console command
adb logcat [ <filter-spec> ] - View device log
adb forward <local> <remote> - forward socket connections
                        forward specs are one of:
                        tcp:<port>
                        localabstract:<unix domain socket name>
                        localreserved:<unix domain socket name>
                        localfilesystem:<unix domain socket name>
                        dev:<character device name>
                        jdwp:<process pid> (remote only)
adb jdwp                - list PIDs of processes hosting a JDWP transport
adb install [-l] [-r] [-s] <file> - push this package file to the device and install it
                        ('-l' means forward-lock the app)
                        ('-r' means reinstall the app, keeping its data)
                        ('-s' means install on SD card instead of internal storage)
adb uninstall [-k] <package> - remove this app package from the device

```

adb 命令支持多种参数搭配来实现不同的 adb 操作。

2.4 查看当前连接设备

输入命令 adb devices 查找 adb 设备，如下图所示：

```

F:\Desktop\000\3588\adb>adb devices
* daemon not running. starting it now on port 5037 *
* daemon started successfully *
List of devices attached
57ab0b0e05056b00    device

F:\Desktop\000\3588\adb>

```

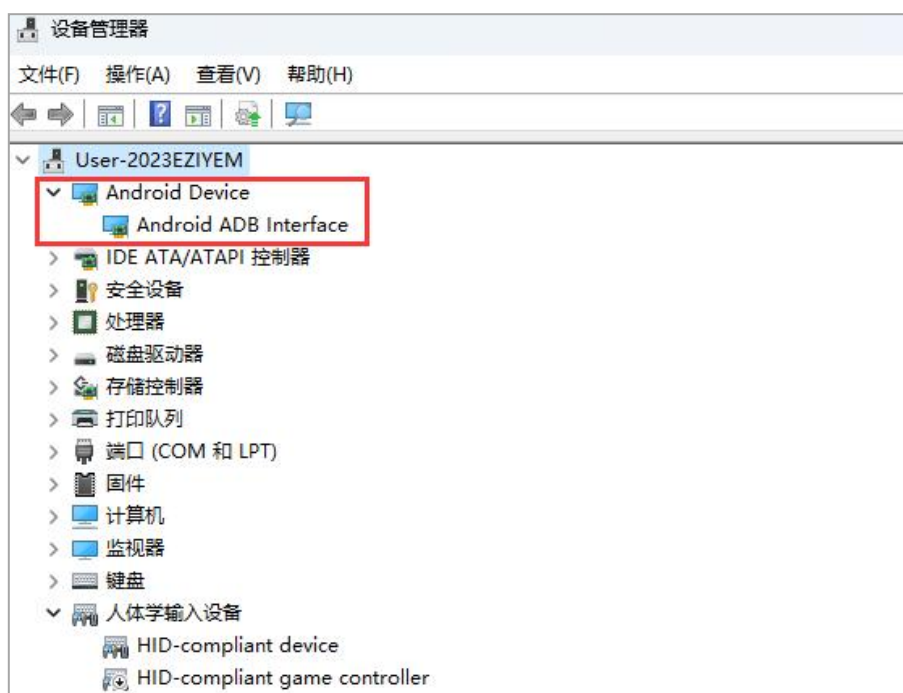
“adb devices”命令用于列举出当前连接的设备有哪些，并将其显示在“List of devices attached”下方，上图中显示有设备连接。

2.5 连接设备

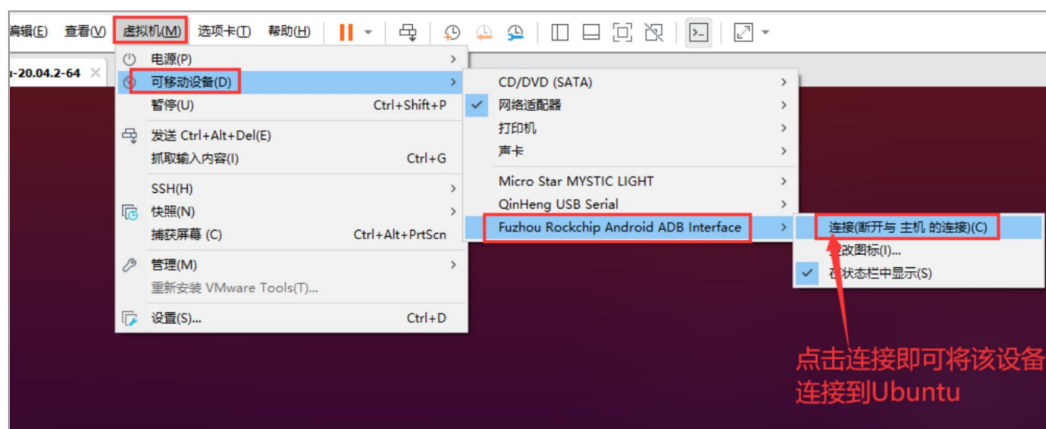
本小节将介绍如何连接设备，设备包括 Android 模拟器，Android 真机（RK3588Android 设备），Linux 设备（RK3588 Linux buildroot 设备、RK3588 Linux debian 设备、RK3588 Linux Ubuntu 设备）。如果是 Android 系统，首先要查看 2.1 小节设置允许 USB 调试。我们可以通过 Type-C 线或者网络（有线网络，无线网络）。

2.5.1 usb 连接 adb 方式

iTOP-RK3588 开发板运行 Linux 系统/Android 系统，（注意 Android 系统需要根据 2.1 小节打开 USB 调试，然后进行本节操作），通过 USB 线将开发板的烧写口连接到电脑上，默认情况下会连接到 Windows 系统上，此时打开设备管理器，会看到“Android ADB Interface”的设备，如下图所示，表示有 adb 设备接入。



如果要在 Ubuntu 系统下使用 adb,那么需要将开发板连接到 Ubuntu 系统，点击虚拟机->可移动设备下面也可以找到该设备，如下图所示，点击连接，便可将开发板连接到 Ubuntu。




```
F:\adb\adb>adb devices
adb server is out of date.  killing...
* daemon started successfully *
List of devices attached
sigma-adb-server      sigma-2.0

F:\adb\adb>
```

6 使用 pull 命令把 Android 设备中的 build.prop 属性配置文件拷贝出来，如下图所示：

adb pull /system/build.prop .

```
F:\adb\adb>adb pull /system/build.prop .
adb server is out of date.  killing...
* daemon started successfully *
failed to copy '/system/build.prop' to './build.prop': open failed: Permission denied
F:\adb\adb>
```

7 如果提示没有权限，输入 adb root,如下图所示：

```
F:\adb\adb>
F:\adb\adb>adb root
adb server is out of date.  killing...
* daemon started successfully *
restarting adbd as root

F:\adb\adb>
```

8 然后重新输入以下命令拷贝，此命令会将 build.prop 文件拷贝到 adb 工具所在的目录。

adb pull /system/build.prop .

```
F:\adb\adb>adb pull /system/build.prop .
adb server is out of date.  killing...
* daemon started successfully *
2356 KB/s (2408 bytes in 0.000s)

F:\adb\adb>
```

9 拷贝出来后，打开 build.prop 文件，在文件的最后添加网络 adb 属性，5555 为端口号。
如下图所示：

service.adb.tcp.port=5555

```
#
# from device/rockchip/rk356x/rk356x.prop
#
ro.vendor.rk_sdk=1
# end of device/rockchip/rk356x/rk356x.prop

#
# ADDITIONAL_BUILD_PROPERTIES
#
ro.treble.enabled=true
persist.debug.dalvik.vm.core_platform_api_policy=just-warn
dalvik.vm.lockprof.threshold=500
net.bt.name=Android
service.adb.tcp.port=5555
```

10 修改完后，使用 `adb push` 命令将配置文件放在 Android 设备的 `/system` 目录中。如下图所示：

```
F:\adb\adb>
F:\adb\adb>adb push build.prop /system/build.prop
adb server is out of date. killing...
* daemon started successfully *
failed to copy 'build.prop' to '/system/build.prop': couldn't create file: Read-only file system
F:\adb\adb>
```

11 如上图可能会提示系统是只读系统，这时需要使用以下命令重新挂载系统为可读可写。如下图所示：

adb remount

```
F:\adb\adb>adb remount
adb server is out of date. killing...
* daemon started successfully *
remount succeeded
F:\adb\adb>
```

12 然后重新 push,如下图所示：

adb push build.prop /system/build.prop

```
F:\adb\adb>adb push build.prop /system/build.prop
adb server is out of date. killing...
* daemon started successfully *
58 KB/s (2408 bytes in 0.039s)
F:\adb\adb>
```

13 输入以下命令，重启下系统

adb reboot

```
F:\3568开发板开发\adb\adb>
F:\3568开发板开发\adb\adb>adb reboot
adb server is out of date. killing...
* daemon started successfully *
F:\3568开发板开发\adb\adb>_
```

14 系统重新启动后要保证开发板可以 ping 同电脑主机，也就是要保证开发板和电脑在一个网段下。

15 输入以下命令连接网络 adb 这里的 ip 地址是开发板的地址，5555 是端口号，重启开发板后，ip 地址可能会发生改变，要输入 ifconfig 命令查看确定开发板的 ip 地址。

```
adb connect 192.168.1.17:5555
```

```
F:\adb\adb>adb connect 192.168.1.17:5555
adb server is out of date. killing...
* daemon started successfully *
connected to 192.168.1.17:5555
F:\adb\adb>_
```

上图显示连接成功，说明网络 adb 已经配置成功。这样就可以拔掉 usb 数据线了。

16 断开网络 adb,输入以下命令：

```
adb disconnect 192.168.1.17:5555
```

2.6 运行 shell

输入 adb shell 命令进入设备 shell 终端，如下图所示：

```
topeet@ubuntu:~$ adb connect 192.168.1.202:5555
connected to 192.168.1.202:5555
topeet@ubuntu:~$ adb shell
root@topeet:/$
root@topeet:/$ ls
bin  dev  lib          mnt  proc  sbin  srv  udisk  var
boot etc  lost+found  oem  root  sdcard sys  userdata
data home media      opt  run  snap  tmp  usr
root@topeet:/$
```

可执行 exit 退出 shell。如果连接了多台设备，那么可以使用-s 选项来指定运行哪一台设备的 shell(如果只连接了一个设备，那么不需要加-s 选项，默认就是该设备)

```
topeet@ubuntu:~$ adb devices
List of devices attached
394e7c3efbc10849    device
192.168.1.202:5555    device

topeet@ubuntu:~$
topeet@ubuntu:~$ adb -s 394e7c3efbc10849 shell
topeet_rk3588s:/ $
topeet_rk3588s:/ $ ls
acct      debug_ramdisk    odm           sys
apex      dev              odm_dkkm      system
bin       etc              oem           system_dkkm
bugreports  init            postinstall   system_ext
cache     init.environ.rc  proc          vendor
config    linkerconfig     product       vendor_dkkm
d         lost+found       sdcard
data      metadata         second_stage_resources
data_mirror  mnt             storage
topeet_rk3588s:/ $
topeet_rk3588s:/ $
```

我们也可以通过以下命令来让设备执行一个 shell 命令:

adb shell command

Command 表示需要让设备执行的命令, 比如以下命令

adb shell ls

adb shell ifconfig

adb shell pwd

```
topeet@ubuntu:~$ adb -s 394e7c3efbc10849 shell pwd
/
topeet@ubuntu:~$
topeet@ubuntu:~$ adb -s 394e7c3efbc10849 shell ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 9a:88:11:7d:fa:ff  Driver rk_gmac-dwmac
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 TX bytes:0
          Interrupt:93

topeet@ubuntu:~$ adb -s 394e7c3efbc10849 shell ls vendor
app
bin
build.prop
etc
firmware
lib
lib64
lost+found
odm
overlay
topeet@ubuntu:~$
```

2.7 文件传输

通过“adb push”命令将本地文件拷贝到设备中。

adb push local remote

local 表示本地文件, remote 表示远端设备的某个目录。将本地文件拷贝至远端设备的某个目录下。

通过“adb pull”命令可以将设备中的文件拷贝到本地。

adb pull remote local

remote 表示远端设备的某个文件, local 表示本地某目录。将远端设备的某个文件拷贝到本地

2.8 应用相关

输入以下命令可以在 Android 设备或者 Android 模拟器上安装 APK

```
adb install path_apk
```

path_apk 表示 APK 的路径。

2.9 获取 root 权限

输入以下命令可以获取 root 权限，使得 adb 有更高的权限。

```
adb root
```

2.10 重新挂载 remount

使用“adb push”命令将本地文件拷贝到设备端的时候，很可能会失败，如下图所示：

```
F:\adb\adb>
F:\adb\adb>adb push build.prop /system/build.prop
adb server is out of date.  killing...
* daemon started successfully *
failed to copy 'build.prop' to '/system/build.prop': couldn't create file: Read-only file system
F:\adb\adb>
```

如上图可能会提示系统是只读系统，这时需要使用以下命令重新挂载系统为可读可写。如下图所示：

```
adb remount
```

```
F:\adb\adb>adb remount
adb server is out of date.  killing...
* daemon started successfully *
remount succeeded
F:\adb\adb>
```

注意！执行“adb remount”命令之前，需要先执行“adb root”命令获取到设备的 root 用户权限，重新挂载之后，再次 push 时就不会出现问题了。

2.11 其他命令

更多使用方法请参考以下网址：

<https://developer.android.google.cn/studio/command-line/adb?hl=zh-cn>