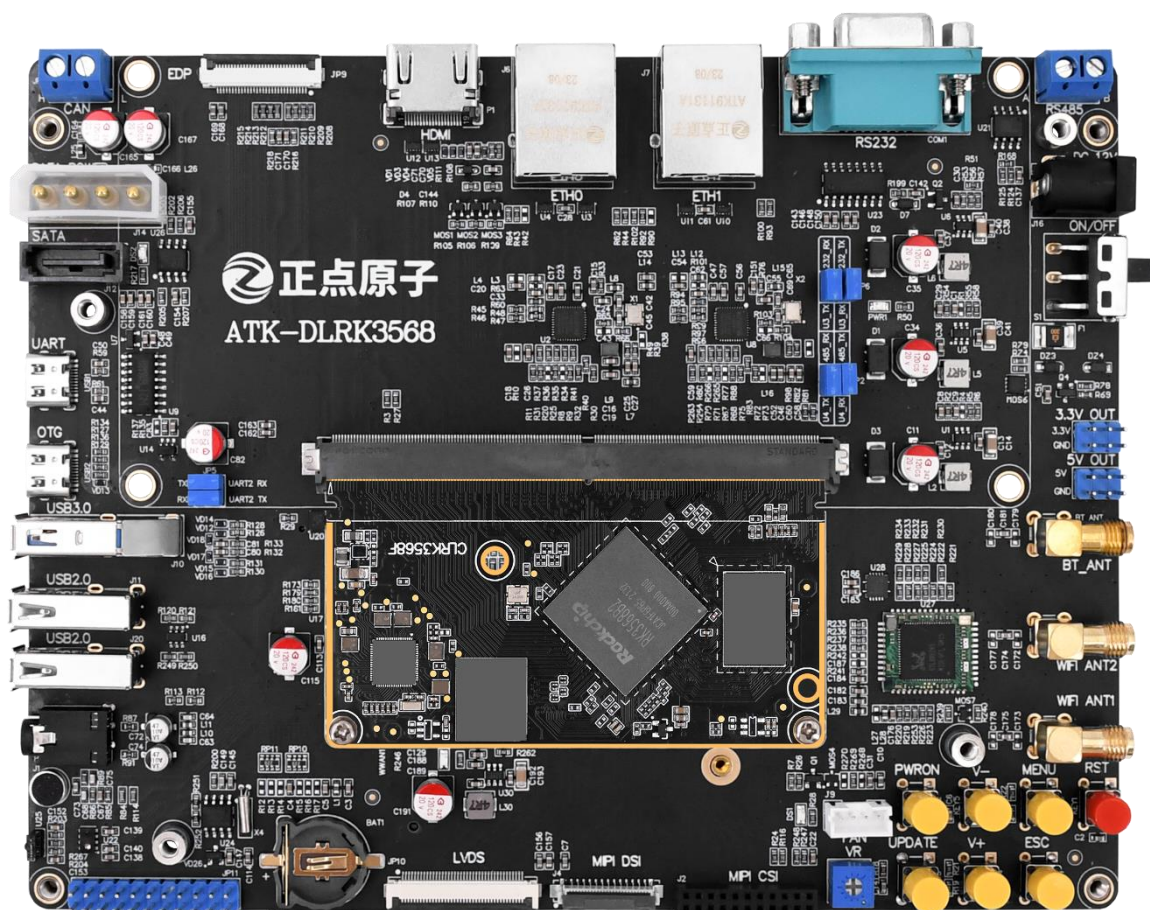


# ATK-DLRK3568

## Android13 SDK 编译

### 说明 V1.0

-正点原子 ATK-DLRK3568 开发板文档



## 修订历史:

版本	日期	修改内容
V1.0	2024/07/10	第一次发布



正点原子公司名称 : 广州市星翼电子科技有限公司

原子哥在线教学平台 : [www.yuanzige.com](http://www.yuanzige.com)

开源电子网 / 论坛 : <http://www.openedv.com/forum.php>

正点原子淘宝店铺 : <https://openedv.taobao.com>

正点原子官方网站 : [www.alientek.com](http://www.alientek.com)

正点原子 B 站视频 : <https://space.bilibili.com/394620890>

电话: 020-38271790 传真: 020-36773971

请关注正点原子公众号, 资料发布更新我们会通知。

请下载原子哥 APP, 数千讲视频免费学习, 更快更流畅。



扫码关注正点原子公众号



扫码下载“原子哥”APP

免责声明 .....	1
前言 .....	2
第 1 章 Android13 SDK 编译 .....	3
1.1 下载 SDK .....	3
1.2 拷贝 SDK 到 Ubuntu 系统.....	4
1.3 选择屏幕 .....	5
1.4 编译 SDK .....	7
1.5 打包成 update.img.....	7

## 免责声明

本文档所提及的产品规格和使用说明仅供参考，如有内容更新，恕不另行通知；除非有特殊约定，本文档仅作为产品指导，所作陈述均不构成任何形式的担保。本文档版权归广州市星翼电子科技有限公司所有，未经公司的书面许可，任何单位和个人不得以营利为目的进行任何方式的传播。

为了得到最新版本的产品信息，请用户定时访问正点原子资料下载中心或者与淘宝正点原子旗舰店客服联系索取。感谢您的包容与支持。

正点原子资料下载中心: <http://www.openedv.com/docs/index.html>

## 前言

本文档为 ATK-DLRK3568 **Android13** SDK 编译说明文档。阅读本文档之前, 请先参考文档: <开发板光盘 A 盘-基础资料→10、用户手册→02、开发文档→【正点原子】ATK-DLRK3568\_Android 系统开发手册.pdf>中的第一章和第二章内容安装 Ubuntu 系统、搭建开发环境。并按照该文档中的 4.1.1 小节所示安装依赖软件包。

## 第 1 章 Android13 SDK 编译

ATK-DLRK3568 Android13 SDK 编译说明。

### 1.1 下载 SDK

由于正点原子的 RK3568 和 RK3588 开发板使用的是同一个 Android13 SDK, RK3568 开发板的资料网盘中并未单独提供, 只在 RK3588 开发板的网盘中有提供, 所以大家需要到 RK3588 开发板的网盘中下载。

首先进入 <http://www.openedv.com/docs/boards/arm-linux/RK3588Linux.html> 此链接, 按照下面的指示下载 Android13 SDK:

### 正点原子RK3588开发板

#### 开发板介绍

- Android12系统演示 B站哔哩哔哩链接: <https://www.bilibili.com/video/BV1Wf421B7wp>
- buildroot+QT系统演示 B站哔哩哔哩链接: <https://www.bilibili.com/video/BV1AT421Y74p>
- AI功能演示 B站哔哩哔哩链接: <https://www.bilibili.com/video/BV16s421T7jK>
- 多屏显示功能演示 B站哔哩哔哩链接: <https://www.bilibili.com/video/BV1Rs421T7NN>
- 8K视频解码及显示功能演示 B站哔哩哔哩链接: 视频待上线...

#### 资料下载链接

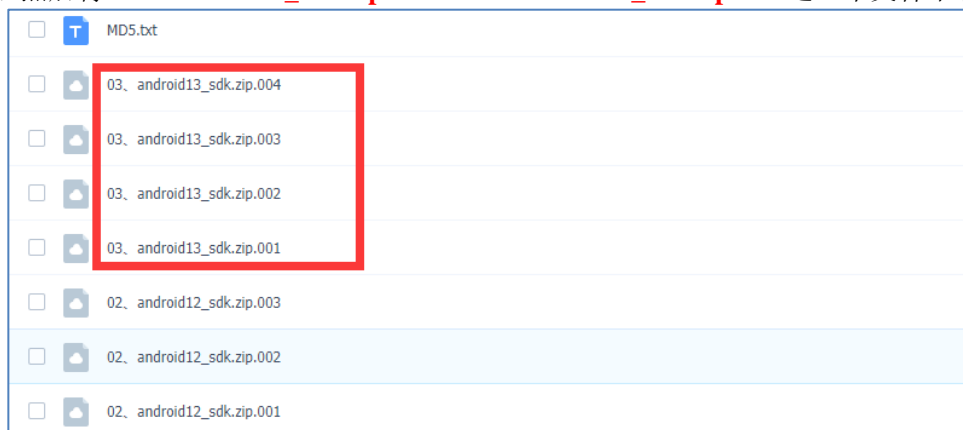
##### 基础资料下载 (A盘)

- 百度网盘 开发板资料链接: <https://pan.baidu.com/s/1M5BEFhBA16YlFqR8rKEw6A> 提取码: bxzv

##### RK3588 SDK与虚拟机链接 (B盘)

- 为方便用户编译RK3588开发板SDK包, 减少因环境搭建问题而导致的SDK包编译错误, 正点原子团队特地打包了一个基础的SDK源码编译使用。如果编译其他程序需要对应的库和工具, 请自行安装和适配。
- 百度网盘 链接: <https://pan.baidu.com/s/1vQEGoP6QYdSGPKDZx8yXCA> 提取码: gq0n

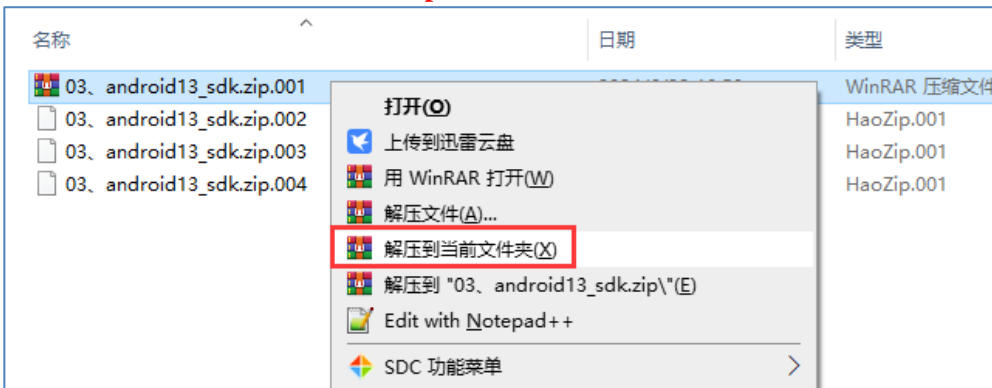
进入网盘后, 依次进入“**开发板光盘 B 盘-开发环境及 SDK→02、ATK-DLRK3588 开发板 SDK**”, 然后将 **03、android13\_sdk.zip.001~03、android13\_sdk.zip.004** 这 4 个文件下载下来:



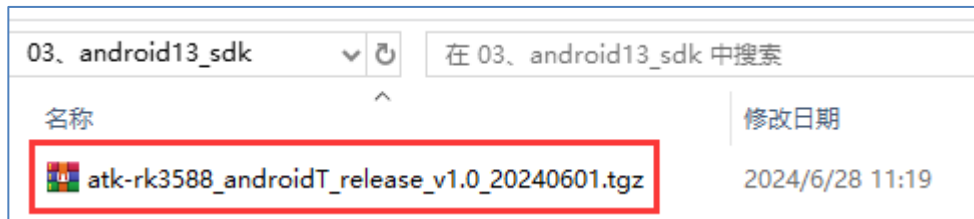
下载完成后如下所示 (确保是完整下载, 可自行对比 MD5 值):

名称	日期
03、android13_sdk.zip.001	2024/6/28 10:53
03、android13_sdk.zip.002	2024/6/28 10:53
03、android13_sdk.zip.003	2024/6/28 10:53
03、android13_sdk.zip.004	2024/6/28 10:53

接着右键单击 **03、android13\_sdk.zip.001** 进行解压:



解压完成后得到一个 **03、android13\_sdk** 文件夹, 进入该文件夹, 如下:



此为 Android13 SDK 的压缩包文件。随着版本更新, SDK 压缩包文件的名字也将会发生改变, 但均以 **atk-rk3588\_androidT\_release\_版本\_发布日期.tgz** 方式进行命名!

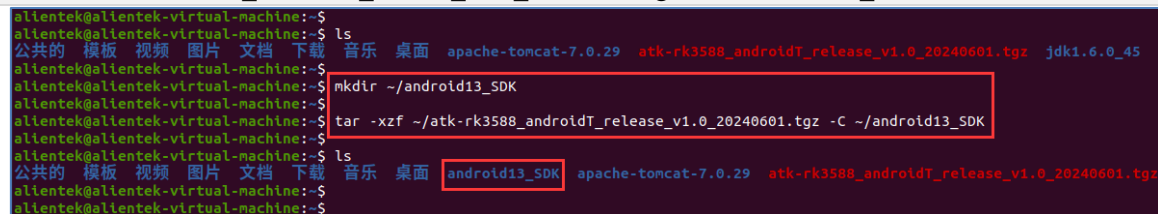
## 1.2 拷贝 SDK 到 Ubuntu 系统

**注意: 如果当前终端编译过其它 SDK, 则需要重新打开一个新的终端, 然后在这个新的终端去编译 Android13 SDK, 否则可能会导致编译出错!!!**

将 **atk-rk3588\_androidT\_release\_v1.0\_20240601.tgz** 压缩文件拷贝到 Ubuntu 系统的用户家目录下, 并执行如下命令进行解压:

```
mkdir ~/android13_SDK
```

```
tar -xzf ~/atk-rk3588_androidT_release_v1.0_20240601.tgz -C ~/android13_SDK
```



解压时间会比较长, 请耐心等待!

解压完成后执行如下命令检出源码:

```
cd ~/android13_SDK/  
.repo/repo/repo sync -l -j10
```



```

allientek@allientek-virtual-machine:~$
allientek@allientek-virtual-machine:~$ cd ~/android13_SDK/
allientek@allientek-virtual-machine:~/android13_SDK$
allientek@allientek-virtual-machine:~/android13_SDK$ .repo/repo/repo sync -l -j10
正在更新文件: 100% (1222/1222), 完成.
正在更新文件: 100% (1133/1133), 完成.
正在更新文件: 100% (9543/9543), 完成.
正在更新文件: 100% (344/344), 完成.
正在更新文件: 100% (1558/1558), 完成.
正在更新文件: 100% (737/737), 完成.
正在更新文件: 100% (2695/2695), 完成.
正在更新文件: 100% (16/16), 完成.
正在更新文件: 100% (2794/2794), 完成.
正在更新文件: 100% (4595/4595), 完成.
正在更新文件: 100% (3440/3440), 完成.
正在更新文件: 100% (78/78), 完成.
正在更新文件: 100% (87/87), 完成.
正在更新文件: 100% (7408/7408), 完成.
正在更新文件: 100% (818/818), 完成.
正在更新文件: 100% (1787/1787), 完成.
正在更新文件: 100% (708/708), 完成.
正在更新文件: 100% (11975/11975), 完成.
正在更新文件: 100% (590/590), 完成.
正在更新文件: 100% (9218/9218), 完成.
正在更新文件: 100% (9260/9260), 完成.
正在更新文件: 100% (34798/34798), 完成.
正在更新文件: 100% (8370/8370), 完成.
正在更新文件: 100% (1523/1523), 完成.
正在更新文件: 100% (1366/1366), 完成.
正在更新文件: 100% (8703/8703), 完成.
正在更新文件: 100% (1904/1904), 完成.
Checking out: 100% (1169/1169), done in 1h50m51.820s
repo sync has finished successfully.
allientek@allientek-virtual-machine:~/android13_SDK$

```

这个过程也将会持续很长一段时间，请耐心等待！

检出完成后就得到了整个 Android13 SDK 的源码目录，如下所示：

```

allientek@allientek-virtual-machine:~/android13_SDK$
allientek@allientek-virtual-machine:~/android13_SDK$ ls
Android.bp      BUILD          device        kernel-5.10   packages      RKDocs        toolchain
art             build.sh      external     libcore       pdk           rkst          tools
bionic         cts          frameworks   libnativehelper platform_testing RKTools       u-boot
bootable       dalvik       hardware     mkcombinedroot prebuilts     sdk           vendor
bootstrap.bash developers   javaenv.sh   mkimage_ab.sh restore_patches.sh system        WORKSPACE
build          development  kernel       mkimage.sh    rkbin         test
allientek@allientek-virtual-machine:~/android13_SDK$

```

### 1.3 选择屏幕

正点原子 ATK-DLRK3568 开发板支持以下种类屏幕：

- 正点原子 5.5 寸 1080x1920（竖屏）MIPI 屏
- 正点原子 5.5 寸 720x1280（竖屏）MIPI 屏
- 正点原子 10.1 寸 800x1280（竖屏）MIPI 屏
- 正点原子 10.1 寸 1280x800（横屏）LVDS 屏
- 标准 HDMI 显示屏（通过 HDMI 线连接）
- eDP 屏（需自己做转接板连接 eDP 屏，正点原子不做 eDP 屏，用户可在网上购买）

ATK-DLRK3568 开发板上有 1 个 MIPI 屏接口（MIPI DSI），可连接正点原子 5.5 寸或 10.1 寸 MIPI 屏；1 个 LVDS 屏接口，可连接正点原子 10.1 寸 1280x800 LVDS 屏；1 个 HDMI 输出接口，可连接标准 HDMI 显示器；1 个 eDP 屏接口，可连接 eDP 屏幕；用户可以在板子上找到这些显示接口，这里就不贴图了！

用户需要根据自己购买的屏幕、配置使能对应的显示接口。

打开 `kernel-5.10/arch/arm64/boot/dts/rockchip/rk3568-screen_choose.dtsi` 设备树文件：

```
vi kernel-5.10/arch/arm64/boot/dts/rockchip/rk3568-screen_choose.dtsi
```

```

/*
 * 屏幕选择
 * ATK_LCD_TYPE_MIPI_720P: 正点原子5.5寸 720*1280 MIPI屏
 * ATK_LCD_TYPE_MIPI_1080P: 正点原子5.5寸 1080*1920 MIPI屏
 * ATK_LCD_TYPE_MIPI_10P1_800X1280: 正点原子10.1寸 800*1280 MIPI屏
 * ATK_LCD_TYPE_LVDS: 正点原子10.1寸 1280*800 LVDS屏
 * ATK_LCD_TYPE_HDMI: HDMI显示器
 * ATK_LCD_TYPE_EDP_VGA: eDP屏或者VGA显示器 (硬件默认使能的是VGA接口, 若用户需要使用eDP屏, 则需修改硬件
 * 具体情况可以看正点原子RK3568底板原理图!)
 */
/*
 * RK3568可支持三屏显示, 也就是三路显示 VP0 VP1 VP2
 * 但是三屏显示需要注意一些问题, 具体情况可以看正点原子提供的文档<RK3568三屏显示参考手册> !
 */
/*
 * ATK_LCD_TYPE_MIPI_720P/ATK_LCD_TYPE_MIPI_1080P/ATK_LCD_TYPE_MIPI_10P1_800X1280 三选一
 */
#define ATK_LCD_TYPE_MIPI_720P // 从VP1输入
// #define ATK_LCD_TYPE_MIPI_1080P // 从VP1输入
// #define ATK_LCD_TYPE_MIPI_10P1_800X1280 // 从VP1输入
/*
 * ATK_LCD_TYPE_HDMI 和 ATK_LCD_TYPE_EDP_VGA 二选一
 */
#define ATK_LCD_TYPE_HDMI // 从VP0输入
// #define ATK_LCD_TYPE_EDP_VGA // 从VP0输入
// #define ATK_LCD_TYPE_LVDS // 从VP2输入

```

默认使能了 5.5 寸 720x1280 MIPI 屏和 HDMI 屏; 如果用户不需要在板子上接 MIPI 屏, 只使用 HDMI 显示屏, 定义如下:

```

// #define ATK_LCD_TYPE_MIPI_720P // 从VP1输入
// #define ATK_LCD_TYPE_MIPI_1080P // 从VP1输入
// #define ATK_LCD_TYPE_MIPI_10P1_800X1280 // 从VP1输入

/*
 * ATK_LCD_TYPE_HDMI 和 ATK_LCD_TYPE_EDP_VGA 二选一
 */
#define ATK_LCD_TYPE_HDMI // 从VP0输入
// #define ATK_LCD_TYPE_EDP_VGA // 从VP0输入

// #define ATK_LCD_TYPE_LVDS // 从VP2输入

```

如果用户使用的是 10.1 寸 800x1280 MIPI 屏, 则定义如下:

```

// #define ATK_LCD_TYPE_MIPI_720P // 从VP1输入
// #define ATK_LCD_TYPE_MIPI_1080P // 从VP1输入
#define ATK_LCD_TYPE_MIPI_10P1_800X1280 // 从VP1输入

/*
 * ATK_LCD_TYPE_HDMI 和 ATK_LCD_TYPE_EDP_VGA 二选一
 */
#define ATK_LCD_TYPE_HDMI // 从VP0输入
// #define ATK_LCD_TYPE_EDP_VGA // 从VP0输入

// #define ATK_LCD_TYPE_LVDS // 从VP2输入

```

如果用户使用的是 10.1 寸 LVDS 屏, 则定义如下:

```

// #define ATK_LCD_TYPE_MIPI_720P // 从VP1输入
// #define ATK_LCD_TYPE_MIPI_1080P // 从VP1输入
// #define ATK_LCD_TYPE_MIPI_10P1_800X1280 // 从VP1输入

/*
 * ATK_LCD_TYPE_HDMI 和 ATK_LCD_TYPE_EDP_VGA 二选一
 */
#define ATK_LCD_TYPE_HDMI // 从VP0输入
// #define ATK_LCD_TYPE_EDP_VGA // 从VP0输入

#define ATK_LCD_TYPE_LVDS // 从VP2输入

```

总之, 用户需根据自己所使用的屏幕来配置, 配置完成后保存退出!

## 1.4 编译 SDK

在 android13\_SDK 目录下, 执行如下命令编译:

```
source build/envsetup.sh
lunch ATK_DLRK3568-userdebug
./build.sh -UCKA -J12
```

-U: 表示编译 U-Boot  
-K: 表示编译 Linux Kernel  
-C: 使用 clang 编译器编译  
-A: 表示编译 Android  
-J12: 指定编译线程数量为 12

整个编译过程将会持续很长一段时间, 大概需要 3~4 小时左右, 或者更长时间, 具体情况取决于个人电脑的配置 (CPU 和内存)。

如果没有出现意外, 将会编译成功。编译完成后, 生成的镜像文件将会打包到 **rockdev/Image-ATK\_DLRK3568/** 目录下, 如下所示:

```
Android13_SDK$ cd rockdev/Image-ATK_DLRK3568/
Android13_SDK/rockdev/Image-ATK_DLRK3568$
Android13_SDK/rockdev/Image-ATK_DLRK3568$ ls
baseparameter.img  boot.img  dtbo.img  misc.img  pcba_small_misc.img  recovery.img  super.img  vbmeta.img
boot-debug.img     config.cfg  MiniLoaderAll.bin  parameter.txt  pcba_whole_misc.img  resource.img  uboot.img
Android13_SDK/rockdev/Image-ATK_DLRK3568$
```

## 1.5 打包成 update.img

在 SDK 根目录下执行如下命令可将 **rockdev/Image-ATK\_DLRK3568/** 目录下的分立镜像打包成一个 **update.img** 固件, 方便用户烧录:

```
./build.sh -u
```

生成的 **update.img** 固件也存放在 **rockdev/Image-ATK\_DLRK3568/** 目录。