

# IDO-EVB3568-V1 - Android SDK 编译说明

---

一、源码获取

二、开发主机环境配置

三、SDK编译

1. uboot编译步骤

2. kernel编译步骤

单独编译kernel生成可直接烧录的boot.img

3. Android 编译及固件生成步骤

4. 固件打包



## IDO-EVB3568-V1 Android SDK编译说明

深圳触觉智能科技有限公司

[www.industio.cn](http://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	Lyle		2022/03/17
V1.1	修改开发主机环境配置说明	FuYingzhe		2023/08/01

## 一、源码获取

### 源码下载路径

链接: [https://pan.baidu.com/s/1cc91gGOY87aLb9\\_DwbD0pQ](https://pan.baidu.com/s/1cc91gGOY87aLb9_DwbD0pQ)

提取码: 1234

md5sum:747f4dd47a58a5b965ba3aac4735074d

解压源码

```
▼ Bash |  
1 cat ido_evb3568_android11_211025_sdk.tgz.xa* > ido_evb3568_android11_211025_sdk.tgz  
2 tar xzvf ido_evb3568_android11_211025_sdk.tgz -C .
```

注: ido\_evb3568\_android11\_211025\_sdk目录下有[Readme.txt](#)请必读

## 二、开发主机环境配置

推荐编译主机配置如下:

- Ubuntu18.04 操作系统
- 64 位 CPU
- 16GB 物理内存+交换内存
- 250GB 空闲的磁盘空间

开发环境搭建, 请参考SDK根目录下

RKDocs\android\Rockchip\_Developer\_Guide\_Android11\_SDK\_V1.1.5\_CN.pdf文档中的 附录 A 编译开发环境搭建, 安装OpenJDK 8和一些编译依赖软件, Ubuntu18.04通用软件安装包

```
Shell |
1 $ sudo apt-get update
2 $ sudo apt-get install git gnupg flex bison gperf libsdl1.2-dev libesd-jav
  a \
3 libwxgtk3.0-dev squashfs-tools build-essential zip curl libncurses5-dev zli
  b1g-dev \
4 pngcrush schedtool libxml2 libxml2-utils xsltproc lzop libc6-dev schedtool
  g++-multilib \
5 lib32z1-dev lib32ncurses5-dev lib32readline-dev gcc-multilib libswitch-per
  l libssl-dev \
6 unzip zip device-tree-compiler liblz4-tool python-pyelftools python3-pyelft
  ools -y
```

### 三、SDK编译

#### 1. uboot编译步骤

进入 sdk 根目录执行命令

```
Shell |
1 $ cd u-boot
2 $ ./make.sh rk3568
```

#### 2.kernel编译步骤

内核配置文件路径: kernel/arch/arm64/configs/rockchip\_defconfig

设备树文件路径: kernel/arch/arm64/boot/dts/rockchip/

EVB3568-V1开发板dts说明如下:

序号	文件名	功能说明
----	-----	------

1	ido-evb3568-v1b-hdmi.dts	显示为hdmi
2	ido-evb3568-v1b-dsi1-1200-1920.dts	MIPI 1200x1920
3	ido-rk3568-evb3568-v1b-dlvds-1080p-hdmi.dts	双通道LVDS 1920x1080
4	ido-rk3568-evb3568-v1b-edp-1080p-hdmi.dts	eDP 2lane 1920x1080
5	ido-rk3568-evb3568-v1b-lvds-1024-600-hdmi.dts	LVDS 1024x600

以编译HDMI屏内核为例，编译方法如下：

```

Shell |
1  $ cd kernel
2  $ make ARCH=arm64 rockchip_defconfig rk356x_evb.config android-11.config
3  $ make ARCH=arm64 ido-evb3568-v1b-hdmi.img -j10

```

**注：**编译如果提示选择io\_domain电压，我司核心板vccio4，vccio6是1.8v，其它为3.3v。

以上方式编译完成后，kernel目录生成的boot.img文件不能直接用于烧录。内核分区烧录的文件是编译完Android所产生的rockdev/Image-rk3568\_r/boot.img文件。

### 单独编译kernel生成可直接烧录的boot.img

此处的编译方法的前提已存在rockdev/Image-rk3568\_r/boot.img文件（即Android代码已经完全编译过一次）

编译的原理：在kernel目录下将编译生成的 kernel.img 和 resource.img 替换到旧的 boot.img 中。这样就能在调试驱动时，快速编译获得可直接烧录的内核镜像。

```

Shell |
1  $ cd kernel
2  $ make ARCH=arm64 rockchip_defconfig rk356x_evb.config android-11.config
3  $ ./mk_kernel.sh ido-evb3568-v1b-hdmi.img

```

使用此方法编译出kernel/boot.img文件可以直接用于烧录至boot分区。

## 3.Android 编译及固件生成步骤

```
▼ Shell |
1 $ source build/envsetup.sh
2 $ lunch rk3568_r-userdebug
3 $ make -jn
```

n: CPU内核数（按最大数编译一般为CPU数\*2），user版本选择：lunch rk3568\_r-user

## 4. 固件打包

编译完成后，执行 SDK 根目录下的 mkimage.sh 脚本生成固件，所有烧写所需的各分区镜像会被拷贝至rockdev/Image-rk3568\_r/目录

```
▼ Shell |
1 $ ./mkimage.sh
```

将所有分区镜像合并成单个的镜像

```
▼ Bash |
1 $ cd RKTools/linux/Linux_Pack_Firmware/rockdev/
2 $ ./mkupdate_rk356x.sh
```

执行mkupdate\_rk356x.sh 命令后，会将各分区镜像合并成一个整包镜像文件 RKTools/linux/Linux\_Pack\_Firmware/rockdev/update.img。