

# IDO-EVB3588S-V1 Android开发手册

---

1 源码获取

2 Android\_SDK编译环境配置

3 SDK编译

3.1 一键编译

3.2 单独编译

3.2.1 uboot编译步骤

3.2.2 kernel编译步骤

3.2.3 Android 编译

4 固件打包



IDO-EVB3588S-V1

Android 开发手册

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	PCBA版本	修订内容	修订	审核	日期
V1.0	V1A	创建文档	LJZ	IDO	2024/04/28
V1.1	V1C	优化文档格式	TBR	IDO	2024/08/26

# 1 源码获取

链接: <https://pan.baidu.com/s/1GbU1t0X9bRSMXS09PwlqzA?pwd=1234>

提取码: 1234

```
▼ 解压源码: Shell |
1 #合并tar压缩包文件
2 cat rk3588-Android12.tar.gz.0* > rk3588-Android12.tar.gz
3 #校验压缩包完整性
4 md5sum -c rk3588-Android12.tar.gz.md5
5 #解压SDK
6 tar -xzf rk3588-Android12.tar.gz
7 #进入SDK目录解压, checkout 源码
8 cd rk3588-Android12/
9 # 目录下存在 .git/ 的隐藏文件
10 git reset --hard
```

# 2 Android\_SDK编译环境配置

推荐编译主机配置如下：

1. Ubuntu22.04 操作系统64位
2. 64 位 CPU
3. 16GB 物理内存+交换内存
4. 250GB 空闲的磁盘空间

开发环境搭建，请参考SDK根目录下

RKDocs\android\RK3588\_Android12\_SDK\_Developer\_Guide\_CN.pdf，安装OpenJDK 8和一些编译依赖软件。

```
Bash |
1 $ sudo apt-get update
2 $ sudo apt-get install openjdk-8-jdk
3 $ sudo apt-get install git gnupg flex bison gperf libssl1.2-dev libbsd-jav
a \
4 squashfs-tools build-essential zip curl libncurses5 libncurses5-dev zlib1g
-dev \
5 pngcrush schedtool libxml2 libxml2-utils xsltproc lzip libncurses5-dev schedtool
g++-multilib \
6 lib32z1-dev lib32ncurses-dev lib32readline-dev gcc-multilib libswitch-perl
libssl-dev \
7 unzip zip device-tree-compiler liblz4-tool python2 python3-pyelftools -y
```

## 3 SDK编译

### 3.1 一键编译

```
Shell |
1 source build/envsetup.sh
2 lunch evb3588s-userdebug
3 ./build.sh -UKAu
4 # 编译完成的镜像在 rockdev/Image-evb3588s/ID0_EVB3588s-Android12-xxx.img
```

### 3.2 单独编译

#### 3.2.1 uboot编译步骤

进入 sdk 根目录执行命令

```
Shell |  
1  $ cd u-boot  
2  $ ./make.sh rk3588
```

### 3.2.2 kernel编译步骤

内核配置文件路径: kernel-5.10/arch/arm64/configs/rockchip\_defconfig

设备树文件路径: kernel-5.10/arch/arm64/boot/dts/rockchip/

IDO-EVB3588s 开发板 dts

```
1 #include "ido-evb3588s-typec0.dtsi" /* dp0 in vp1 */
2 #include "ido-evb3588s-gmac1.dtsi"
3
4 // M2 和 mSATA 互斥, 宏 M2 为 1,使能 M2.0;为 0,使能 mSATA
5 #define M2 1
6 #if M2
7 #include "ido-evb3588s-m2.dtsi"
8 #else
9 #include "ido-evb3588s-msata.dtsi"
10 #endif
11
12 // #include "ido-evb3588s-dcphy0-imx415.dtsi" /* J25 */
13 // #include "ido-evb3588s-dphy0-imx415.dtsi" /* J26 */
14 // #include "ido-evb3588s-dcphy1-ov13855.dtsi" /* J27 */
15 #include "ido-evb3588s-dcphy0-imx415-dcphy1-ov13855.dtsi" /* J25 and J27
*/
16
17 // HDMI 和 eDP 互斥, 宏 HDMI0 为 1,使能 HDMI;为 0,使能 eDP
18 #define HDMI0 1
19 #if HDMI0
20 #include "ido-evb3588s-hdmi0.dtsi" /* hdmi0 in vp0 */
21 #else
22 #include "ido-evb3588s-edp0.dtsi" /* edp0 in vp0 */
23 #endif
24
25 #include "ido-evb3588s-gpio.dtsi"
26
27 #include "ido-evb3588s-mipi1-1200-1920.dtsi" /* mipi1 in vp2 */
28 #include "ido-evb3588s-mipito8775-1920-1080.dtsi" /* mipi0 in vp3 */
29 // #include "ido-evb3588s-mipito8775-1280-800.dtsi" /* mipi0 in vp3 */
```

```
ido-evb3588s-v1b.dts C |
1 #include "ido-evb3588s-typec0.dtsi" /* dp0 in vp1 */
2 #include "ido-evb3588s-gmac1.dtsi"
3
4 // m2.0 and msata can't same enabled
5 #define M2 1
6 #if M2
7 #include "ido-evb3588s-m2.dtsi"
8 #else
9 #include "ido-evb3588s-msata.dtsi"
10 #endif
11
12 // #include "ido-evb3588s-dcphy0-imx415.dtsi" /* J25 */
13 // #include "ido-evb3588s-dcphy1-ov13855.dtsi" /* J27 */
14
15 #include "ido-evb3588s-dphy0-imx415.dtsi" /* J26 */
16 #include "ido-evb3588s-imx415-dcphy0-isp0-ov13855-dcphy1-isp1.dtsi" /* J2
17 5 and J27 */
18
19 // hdmi0 and edp0 can't same eneble
20 #define HDMI0 1
21 #if HDMI0
22 #include "ido-evb3588s-hdmi0.dtsi" /* hdmi0 in vp0 */
23 #else
24 #include "ido-evb3588s-edp0.dtsi" /* edp0 in vp0 */
25 #endif
26
27 #include "ido-evb3588s-gpio.dtsi"
28
29 #include "ido-evb3588s-mipi1-1200-1920.dtsi" /* mipi1 in vp2 */
30 #include "ido-evb3588s-mipito8775-1920-1080.dtsi" /* mipi0 in vp3 */
31 // #include "ido-evb3588s-mipito8775-1280-800.dtsi" /* mipi0 in vp3 */
```

编译方法如下：

```
Shell |
1 $ cd kernel-5.10
2 $ ./mkboot.sh # 编译内核
```

编译完成后，kernel-5.10 根目录，生成 boot.img 文件

**注意：** RK3588S 不用配置电压域

### 3.2.3 Android 编译

```
userdebug版本 Shell |
1 $ source build/envsetup.sh
2 $ lunch evb3588s-userdebug
3 $ make -jx # 其中x为使用线程数
```

- user版本选择: `lunch evb3588s-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将都 `rockdev/Image-rk3588_s/` 目录下

```
Shell |
1 $ ./mkimage.sh
```

将所有分区镜像合并成单个的镜像

```
Shell |
1 $ cd RKTools/linux/Linux_Pack_Firmware/rockdev
2 $ ./mkupdate_rk3588.sh
```

执行 `./mkupdate_rk3588.sh` 命令后会将各分区镜像合并成一个 `update.img` 的镜像文件。