

IDO-EVB3588-V1C Android SDK 编译手册

一、源码获取

二、开发板环境配置

三、SDK编译

1. uboot编译步骤

2.kernel编译步骤

3.Android 编译及固件生成步骤

4. 固件打包



IDO-EVB3588-V1C Android SDK编译手册

深圳触觉智能科技有限公司

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	骆建志		2022/04/25
V1.1	修改文档为IDO-EVB3588-V1C Android SDK编译手册	huangjintao		2023/07/13
V1.2	更新EVB3588-V1C SDK	huangjintao		2023/12/19

一、源码获取

源码下载路径

链接: <https://pan.baidu.com/s/1fL3Y1K069Mfr0M2KuJqi9g?pwd=1234>

提取码: 1234

解压源码

```
▼ Shell |
1 # 校验文件是否完整
2 md5sum -c RK3588-Android12-20231212
3 # 解压缩SDK, -o 后接目标目录, 之间没有空格
4 7zr x rk3588-Android12-231211.7z.00* -otarget_patch
5 # 进入目录, checkout 源码
6 cd rk3588-Android12/ # 目录下存在 .git/ 的隐藏文件
7 git reset --hard
```

二、开发板环境配置

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。

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 rk3588
```

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/

3588开发板dts

```
ido-evb3588-v1c.dts

1 #include "dt-bindings/usb/pd.h"
2 #include "rk3588.dtsi"
3 #include "rk3588-android.dtsi"
4 #include "rk3588-rk806-single.dtsi"
5 #include "rk3588-ido-som3588.dtsi"//触觉移植适配
6 #include "ido-evb3588-imx415.dtsi"//摄像头 imx415
7 #include "ido-typec1.dtsi"
8
9 // #include "ido-hdmi0-hdmi1-mipi-1200-1920-mipi2lvds-1920-1080-dp0-dp1.dtsi"
10
11 #include "ido-hdmi0.dtsi"
12 #include "ido-hdmi1.dtsi"
13
14 // #include "ido-dp0.dtsi"//Type-C0 DP输出, 需要则取消注释
15 #include "ido-dp1.dtsi"//Type-C1 DP输出
16
17 #include "ido-mipito8775-1920-1080.dtsi"//双8 LVDS
18 // #include "ido-mipito8775-1280-800.dtsi"
19 // #include "ido-mipi1-1200-1920.dtsi"//mipi 屏显示
20 // #include "ido-mipi1-800-1280.dtsi"
21
22 // #include "ido-mipi1-800-1280-20230802.dtsi"
23
```

以编译 MIPI 屏内核为例，取消第13行MIPI dtsi注释。编译方法如下：

```
Shell

1 $ cd kernel-5.10
2 $ ./mkboot.sh config #生成内核编译配置
3 $ ./mkboot.sh # 编译内核
```

编译完成后，kernel-5.10 根目录，生成 boot.img 文件

注：RK3588 不用配置电压域

3.Android 编译及固件生成步骤

```
▼ Shell |
1 $ source build/envsetup.sh
2 $ lunch rk3588_s-userdebug
3 $ make -jn
```

- n : CPU内核数
- user版本选择: `lunch rk3588_s-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。