

# IDO-EVB3566-V1 Android开发手册

---



## IDO-EVB3566-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 | V1C    | 创建文档 | HJT | IDO | 2024/05/08 |

# 1 源码获取

## 1.1 源码下载路径

链接：<https://pan.baidu.com/s/1WGSk4WukhVchpeVanClelg?pwd=1234>

提取码：1234

解压源码命令如下：

```
1 tar -xzf ido_evb3566-v1b_android11_sdk_240605.tar.tar.gz -C your/target/path
```

## 1.2 补丁获取及使用方法

补丁文件：

[0001-add-rga3-and-Fixed-the-green-screen-problem.patch](#)

使用方法：

将补丁文件拷贝到源码跟目录，使用git am命令合并到原始sdk中；如果存在冲突的情况，请自行使用files中的文件对比修改解决冲突后提交，命令如下：

```
1 git am 0001-add-rga3-and-Fixed-the-green-screen-problem.patch
```

```
commit 18c590b589358a14f4736d0da4922cd53067d245 (HEAD -> evb3566-v1a)
Author: huangjintao <huangjintao@industio.com>
Date: Thu Mar 27 14:43:15 2025 +0800

    add rga3 and Fixed the green screen problem
```

## 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
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

```
1 $ sudo apt-get update
2 $ sudo apt-get install git gnupg flex bison gperf libssl1.2-dev libbsd-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
```

## 3 SDK编译

### 3.1 uboot编译步骤

进入 sdk 根目录执行命令如下：

```
1 $ cd u-boot
2 $ ./make.sh rk3566
```

### 3.2 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/

EVB3566-V1B开发板dts：ido-evb3566-v1a.dts

```

1  /dts-v1/;
2
3  #include <dt-bindings/gpio/gpio.h>
4  #include <dt-bindings/display/media-bus-format.h>
5  #include <dt-bindings/pinctrl/rockchip.h>
6  #include "rk3566.dtsi"
7  #include "rk3568-android.dtsi"
8  #include "ido-sbc3566-core.dtsi"
9  // #include "ido-sbc3566-v1a-light-sensor.dtsi"
10
11 /* lcd */
12 // #include "ido-evb3566-v1a-lvds-1280-800.dtsi" //单lvds屏
13 // #include "ido-evb3566-v1a-edp-1920-1080.dtsi" //edp屏
14 #include "ido-evb3566-v1a-mipi-1200-1920.dtsi" //mipi屏

```

### 3.2.1 完整编译kernel

以编译mipi屏内核为例（默认为hdmi），取消dts第14行注释，编译命令如下：

```

1  $ cd kernel
2  $ make ARCH=arm64 rockchip-evb3566_defconfig
3  $ make ARCH=arm64 ido-evb3566-v1a.img -j10

```

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-rk3566\_r/boot.img文件。

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

### 3.2.2 单独编译kernel

此处的编译方法的前提是，已存在rockdev/Image-rk3566\_r/boot.img文件（即Android代码已经完全编译过一次）。

编译原理：在kernel目录下将编译生成的 kernel.img 和 resource.img 替换到旧的 boot.img 中，可以直接单独烧录。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。

```

1  $ cd kernel
2  $ make ARCH=arm64 rockchip-evb3566_defconfig
3  $ ./mk_kernel.sh ido-evb3566-v1a.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。

**注意：** make ARCH=arm64 rockchip\_defconfig，正常没有修改rockchip\_defconfig配置文件的情况下只需要执行一次即可。

### 3.3 Android 编译及固件生成步骤

```
1  $ source build/envsetup.sh
2  $ lunch rk3566_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 rk3566\_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-rk3566\_r/目录，命令如下：

```
1  $ ./mkimage.sh
```

将所有分区镜像合并成单个的镜像，命令如下：

```
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。