

IDO-EVB7608-V1 Android开发手册



IDO-EVB7608-V1 开发手册

深圳触觉智能科技有限公司

www.industio.cn

版本	PCBA版本	修订内容	修订	审核	日期
V1.0	V1B	创建文档	TBR	IDO	2024/11/26

注意事项

源码的解压和编译使用普通用户即可，无需使用sudo或root用户。

1. 源码获取

百度网盘:EVB7608-Android/4.软件资料/SDK

校验下载文件的完整性，命令如下：

▼XML

```
1 # 校验压缩包完整性
2 md5sum -c evb7608_android14.md5
```

结果如下图所示：

```
evb7608_android14_v1.0.tar.gzaa: 成功
evb7608_android14_v1.0.tar.gzab: 成功
evb7608_android14_v1.0.tar.gzac: 成功
evb7608_android14_v1.0.tar.gzad: 成功
evb7608_android14_v1.0.tar.gzae: 成功
evb7608_android14_v1.0.tar.gzaf: 成功
```

创建一个文件夹（名字自拟），将下载的文件解压到 evb7608 目录，命令如下：

▼XML

```
1 # mkdir evb7608
2 # cat evb7608_android14_v1.0.tar.gza* | tar -xz -C evb7608
3 #进入SDK目录解压, checkout 源码
4 cd evb7608/rk3576_android14_v1.0/
```

解压后的进到目录下有.git隐藏文件，使用以下命令从.git中恢复SDK源码，命令如下：

```
1 # git reset --hard
```

```
Android.bp  bootstrap.bash  Copyright_Statement.pdf  development  git_mv.sh  kernel-6.1  mkimage_ab.sh  pdk  platform_testing  rkbin  rockdev  toolchain  WORKSPACE
art         build           cts                 device       hardware    libcore     mkimage.sh     out          prebuilts     RKDocs     sdk          tools
bionic     BUILD          dalvik             external     javaenv.sh  libnativehelper  out           prebuilts     rkst         system    u-boot      vendor
bootable   build.sh       developers         frameworks   kernel      mkcombinedroot  packages      restore_patches.sh  RKTools    test      vendor
```

2. SDK编译

2.1. 一键编译:

进入 sdk 根目录执行命令，可以使用编译脚本，一键编译uboot、kernel、system等所有分区并生成固件。

```
1 source build/envsetup.sh
2 lunch rk3576_u-userdebug
```

结果如下图所示：

```
tbr@dell-PowerEdge-R430:~/work/rk3576_android14_v1.0$
tbr@dell-PowerEdge-R430:~/work/rk3576_android14_v1.0$ source build/envsetup.sh
tbr@dell-PowerEdge-R430:~/work/rk3576_android14_v1.0$ lunch rk3576_u-userdebug

=====
PLATFORM_VERSION_CODENAME=REL
PLATFORM_VERSION=14
PRODUCT_INCLUDE_TAGS=com.android.mainline
TARGET_PRODUCT=rk3576_u
TARGET_BUILD_VARIANT=userdebug
TARGET_ARCH=arm64
TARGET_ARCH_VARIANT=armv8-a
TARGET_CPU_VARIANT=generic
TARGET_2ND_ARCH=arm
TARGET_2ND_ARCH_VARIANT=armv8-a
TARGET_2ND_CPU_VARIANT=generic
HOST_OS=linux
HOST_OS_EXTRA=Linux-5.15.0-122-generic-x86_64-Ubuntu-20.04.5-LTS
HOST_CROSS_OS=windows
BUILD_ID=UQ1A.240205.004.B1
OUT_DIR=out
=====
tbr@dell-PowerEdge-R430:~/work/rk3576_android14_v1.0$
```

```
一键编译命令

1 ./build.sh -UKAu
```

2.2. 分步编译

单独编译前，也需要先配置一次环境

▼ Shell |

```
1 source build/envsetup.sh
2 lunch rk3576_u-userdebug
```

2.2.1. uboot编译

进入 sdk 根目录执行命令

▼ uboot编译 Shell |

```
1 ./build.sh -U
2 # 或
3 cd u-boot && make clean && make mrproper && make distclean && make rk3576_defconfig && ./make.sh && cd -
```

编译完成后，镜像生成在 u-boot/uboot.img

2.2.2. kernel编译

内核配置文件路径：kernel/arch/arm64/configs/rockchip_linux_rk3576_defconfig

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

▼ kernel编译 Shell |

```
1 ./build.sh -K
2 # 或
3 cd kernel-6.1 && ./mkboot.sh && cd -
```

编译完成后，镜像生成在 kernel-6.1/boot.img

注：RK3576 不用配置电压域

2.2.3. recovery编译

▼ recovery编译 Shell |

```
1 make recoveryimage -j16
```

编译完成后，镜像生成在 out/target/product/rk3576_u/recovery.img

2.2.4. Android编译

Android编译命令如下：

Android编译Shell

```
1  ./build.sh -A
2  # 或
3  make -j16 && ./mkimage.sh
```

编译完成后，镜像生成在 out/target/product/rk3576_u/super.img

2.2.5. 固件打包

编译完成后，将所有分区镜像合并成单个的镜像：

固件打包Shell

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

打包完成后，固件生成在：

rockdev/Image-rk3576_u/IDO_EVB7608_V1B_Android14_HDMI4K_DP4K_20241126-XXXX.img

3. 软件开发

3.1. 屏幕配置

kernel-6.1\arch\arm64\boot\dtb\rockchip\ido-evb7608-v1b.dtsi

默认MIPI 1200x1920C

```
1  // #include "ido-evb7608-v1b-mipi2lvds.dtsi" //Dual-LVDS 1920x1080 vp1
2  #include "ido-evb7608-v1b-mipi.dtsi" //MIPI 1200x1920 vp1
3  // #include "ido-evb7608-v1a-edp.dtsi" //eDP 1920x1080 vp0
```

修改后，需要重新编译kernel分区

3.2. LOGO旋转

修改 kernel-6.1/logo.bmp 和 logo_kernel.bmp，LOGO旋转0/90/1180/270度。

修改后，需要重新编译kernel分区

3.3. 系统旋转

系统旋转代码修改如下：

```
device/rockchip/rk3576/BoardConfig.mk  Java |
1  -SF_PRIMARY_DISPLAY_ORIENTATION := 0
2  +SF_PRIMARY_DISPLAY_ORIENTATION := 270 #0/90/180/270
```

修改后，需要重新编译安卓分区

3.4. J87配置成SPI接口

出厂J87的【1、4、6、8】脚默认不做配置，作为预留GPIO口使用，如需配置成SPI接口，修改：

```
kernel-6.1\arch\arm64\boot\dts\rockchip\ido-evb7608-v1b.dtsi  C |
1  &spi1 {
2  -   status = "disabled";
3  +   status = "okay";
4     pinctrl-names = "default";
5     pinctrl-0 = <&spi1m2_csn0 &spi1m2_pins>;
6  };
7
```