IDO-EVB3562-V1 Android 开发手册

Industio
触 觉 智 能

IDO–EVB3562–V1

Android 开发手册

文档修订历史

版本	修订内容	修订	审核	日期
V1.0	创建文档	FYZ	HJT	2023/09/14
V1.1	文档优化	LZR	IDO	2024/04/16

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。

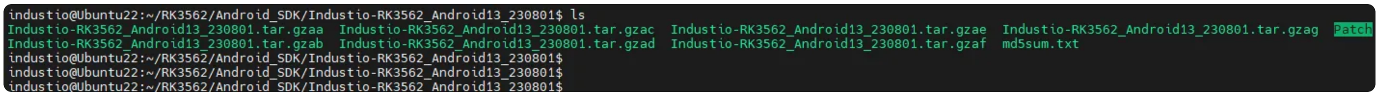
1 源码获取

本文档以Android13为例

链接：https://pan.baidu.com/s/1Y6LOR8yGEDGeWQ1wMM6nEw?pwd=1234

提取码：1234

从网盘下载SDK源码文件到PC端的Linux主机中，如下图所示：

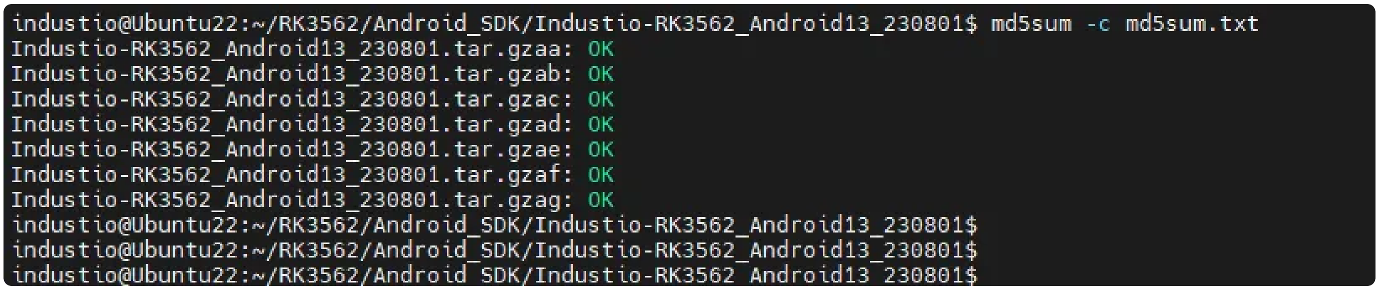


```
industio@Ubuntu22:~/RK3562/Android_SDK/Industio-RK3562_Android13_230801$ ls
Industio-RK3562_Android13_230801.tar.gzaa  Industio-RK3562_Android13_230801.tar.gzac  Industio-RK3562_Android13_230801.tar.gzae  Industio-RK3562_Android13_230801.tar.gzag  Patch
Industio-RK3562_Android13_230801.tar.gzab  Industio-RK3562_Android13_230801.tar.gzad  Industio-RK3562_Android13_230801.tar.gzaf  md5sum.txt
industio@Ubuntu22:~/RK3562/Android_SDK/Industio-RK3562_Android13_230801$
industio@Ubuntu22:~/RK3562/Android_SDK/Industio-RK3562_Android13_230801$
industio@Ubuntu22:~/RK3562/Android_SDK/Industio-RK3562_Android13_230801$
```

校验下载文件的完整性，命令如下：

```Shell
$ md5sum -c md5sum.txt
```

结果如下图所示：

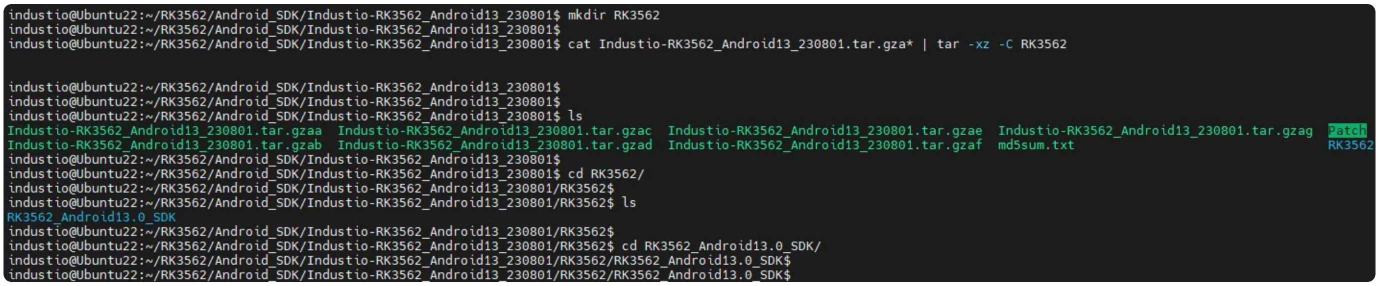


```
industio@Ubuntu22:~/RK3562/Android_SDK/Industio-RK3562_Android13_230801$ md5sum -c md5sum.txt
Industio-RK3562_Android13_230801.tar.gzaa: OK
Industio-RK3562_Android13_230801.tar.gzab: OK
Industio-RK3562_Android13_230801.tar.gzac: OK
Industio-RK3562_Android13_230801.tar.gzad: OK
Industio-RK3562_Android13_230801.tar.gzae: OK
Industio-RK3562_Android13_230801.tar.gzaf: OK
Industio-RK3562_Android13_230801.tar.gzag: OK
industio@Ubuntu22:~/RK3562/Android_SDK/Industio-RK3562_Android13_230801$
industio@Ubuntu22:~/RK3562/Android_SDK/Industio-RK3562_Android13_230801$
industio@Ubuntu22:~/RK3562/Android_SDK/Industio-RK3562_Android13_230801$
```

创建一个文件夹（名字自拟），将下载的文件解压到 RK3562 目录，命令如下：

```Shell
$ mkdir RK3562
$ cat Industio-RK3562_Android13_230801.tar.gza* | tar -xz -C RK3562
$ cd RK3562/RK3562_Android13.0_SDK
```

结果如下图所示：

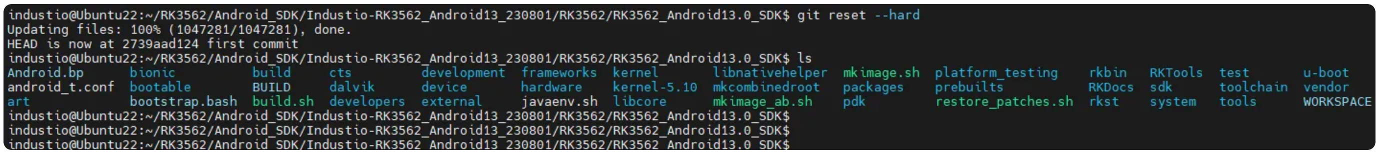


```
industio@Ubuntu22:~/RK3562/Android_SDK/Industio-RK3562_Android13_230801$ mkdir RK3562
industio@Ubuntu22:~/RK3562/Android_SDK/Industio-RK3562_Android13_230801$
industio@Ubuntu22:~/RK3562/Android_SDK/Industio-RK3562_Android13_230801$ cat Industio-RK3562_Android13_230801.tar.gza* | tar -xz -C RK3562


industio@Ubuntu22:~/RK3562/Android_SDK/Industio-RK3562_Android13_230801$
industio@Ubuntu22:~/RK3562/Android_SDK/Industio-RK3562_Android13_230801$
industio@Ubuntu22:~/RK3562/Android_SDK/Industio-RK3562_Android13_230801$ ls
Industio-RK3562_Android13_230801.tar.gzaa  Industio-RK3562_Android13_230801.tar.gzac  Industio-RK3562_Android13_230801.tar.gzae  Industio-RK3562_Android13_230801.tar.gzag  Patch
Industio-RK3562_Android13_230801.tar.gzab  Industio-RK3562_Android13_230801.tar.gzad  Industio-RK3562_Android13_230801.tar.gzaf  md5sum.txt                                 RK3562
industio@Ubuntu22:~/RK3562/Android_SDK/Industio-RK3562_Android13_230801$
industio@Ubuntu22:~/RK3562/Android_SDK/Industio-RK3562_Android13_230801$ cd RK3562/
industio@Ubuntu22:~/RK3562/Android_SDK/Industio-RK3562_Android13_230801/RK3562$
industio@Ubuntu22:~/RK3562/Android_SDK/Industio-RK3562_Android13_230801/RK3562$ ls
RK3562_Android13.0_SDK
industio@Ubuntu22:~/RK3562/Android_SDK/Industio-RK3562_Android13_230801/RK3562$
industio@Ubuntu22:~/RK3562/Android_SDK/Industio-RK3562_Android13_230801/RK3562$ cd RK3562_Android13.0_SDK/
industio@Ubuntu22:~/RK3562/Android_SDK/Industio-RK3562_Android13_230801/RK3562/RK3562_Android13.0_SDK$
industio@Ubuntu22:~/RK3562/Android_SDK/Industio-RK3562_Android13_230801/RK3562/RK3562_Android13.0_SDK$
```

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

```Shell
$ git reset --hard
```

结果如下图所示：



```
industio@Ubuntu22:~/RK3562/Android_SDK/Industio-RK3562_Android13_230801/RK3562/RK3562_Android13.0_SDK$ git reset --hard
Updating files: 100% (1047281/1047281), done.
HEAD is now at 2739aad124 first commit
industio@Ubuntu22:~/RK3562/Android_SDK/Industio-RK3562_Android13_230801/RK3562/RK3562_Android13.0_SDK$ ls
Android.bp      bionic          build     cts         development  frameworks  kernel       libnativehelper  mkimage.sh  platform_testing    rkbin   RKTools  test       u-boot
android_t.conf  bootable        BUILD     dalvik      device       hardware    kernel-5.10  mkcombinedroot   packages    prebuilts           RKDocs  sdk      toolchain  vendor
art             bootstrap.bash  build.sh  developers  external     javaenv.sh  libcore      mkimage_ab.sh    pdk         restore_patches.sh  rkst    system   tools      WORKSPACE
industio@Ubuntu22:~/RK3562/Android_SDK/Industio-RK3562_Android13_230801/RK3562/RK3562_Android13.0_SDK$
industio@Ubuntu22:~/RK3562/Android_SDK/Industio-RK3562_Android13_230801/RK3562/RK3562_Android13.0_SDK$
industio@Ubuntu22:~/RK3562/Android_SDK/Industio-RK3562_Android13_230801/RK3562/RK3562_Android13.0_SDK$
```

1.1 补丁获取及使用方法

补丁位于SDK下载链接的 Android_SDK/patch 目录下。所有的补丁都是基于源码获取的sdk整包生成，需要按顺序逐个合并至sdk源码中。

打补丁的方法：使用git am命令将指定的patch合并到原始sdk中；如果存在冲突的情况，请使用编辑器打开patch文件对比修改，命令如下：

```shell
# 切换到SDK的根目录
cd your_target_path/RK3562_Android13.0_SDK

# 检查是否有冲突
git apply --check xxx.patch

# 打补丁
git am --whitespace=fix xxx.patch
```

2 Android_SDK编译环境配置

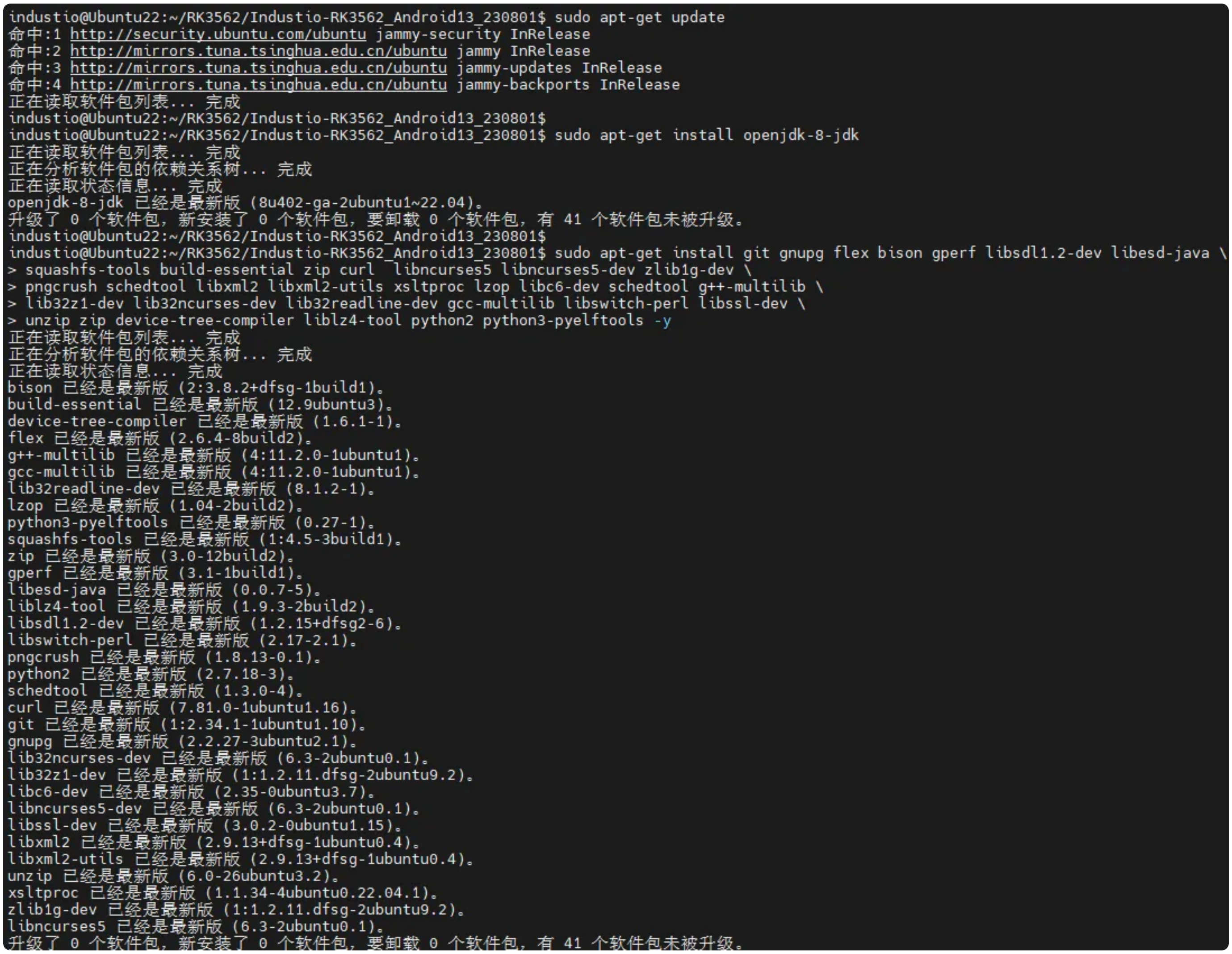
推荐编译主机配置如下：

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RKDocs\android\Rockchip_Android13_SDK_Developer_Guide_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，Ubuntu22.04通用软件安装包，命令如下：

```bash
$ sudo apt-get update
$ sudo apt-get install openjdk-8-jdk
$ sudo apt-get install git gnupg flex bison gperf libsdl1.2-dev libesd-java \
squashfs-tools build-essential zip curl  libncurses5 libncurses5-dev zlib1g-dev \
pngcrush schedtool libxml2 libxml2-utils xsltproc lzop libc6-dev schedtool g++-multilib \
```

结果如下图所示：



```
industio@Ubuntu22:~/RK3562/Industio-RK3562_Android13_230801$ sudo apt-get update
命中:1 http://security.ubuntu.com/ubuntu jammy-security InRelease
命中:2 http://mirrors.tuna.tsinghua.edu.cn/ubuntu jammy InRelease
命中:3 http://mirrors.tuna.tsinghua.edu.cn/ubuntu jammy-updates InRelease
命中:4 http://mirrors.tuna.tsinghua.edu.cn/ubuntu jammy-backports InRelease
正在读取软件包列表... 完成
industio@Ubuntu22:~/RK3562/Industio-RK3562_Android13_230801$
industio@Ubuntu22:~/RK3562/Industio-RK3562_Android13_230801$ sudo apt-get install openjdk-8-jdk
正在读取软件包列表... 完成
正在分析软件包的依赖关系树... 完成
正在读取状态信息... 完成
openjdk-8-jdk 已经是最新版 (8u402-ga-2ubuntu1~22.04)。
升级了 0 个软件包，新安装了 0 个软件包，要卸载 0 个软件包，有 41 个软件包未被升级。
industio@Ubuntu22:~/RK3562/Industio-RK3562_Android13_230801$
industio@Ubuntu22:~/RK3562/Industio-RK3562_Android13_230801$ sudo apt-get install git gnupg flex bison gperf libsdl1.2-dev libesd-java \
> squashfs-tools build-essential zip curl  libncurses5 libncurses5-dev zlib1g-dev \
> pngcrush schedtool libxml2 libxml2-utils xsltproc lzop libc6-dev schedtool g++-multilib \
> lib32z1-dev lib32ncurses-dev lib32readline-dev gcc-multilib libswitch-perl libssl-dev \
> unzip zip device-tree-compiler liblz4-tool python2 python3-pyelftools -y
正在读取软件包列表... 完成
正在分析软件包的依赖关系树... 完成
正在读取状态信息... 完成
bison 已经是最新版 (2:3.8.2+dfsg-1build1)。
build-essential 已经是最新版 (12.9ubuntu3)。
device-tree-compiler 已经是最新版 (1.6.1-1)。
flex 已经是最新版 (2.6.4-8build2)。
g++-multilib 已经是最新版 (4:11.2.0-1ubuntu1)。
gcc-multilib 已经是最新版 (4:11.2.0-1ubuntu1)。
lib32readline-dev 已经是最新版 (8.1.2-1)。
lzop 已经是最新版 (1.04-2build2)。
python3-pyelftools 已经是最新版 (0.27-1)。
squashfs-tools 已经是最新版 (1:4.5-3build1)。
zip 已经是最新版 (3.0-12build2)。
gperf 已经是最新版 (3.1-1build1)。
libesd-java 已经是最新版 (0.0.7-5)。
liblz4-tool 已经是最新版 (1.9.3-2build2)。
libsdl1.2-dev 已经是最新版 (1.2.15+dfsg2-6)。
libswitch-perl 已经是最新版 (2.17-2.1)。
pngcrush 已经是最新版 (1.8.13-0.1)。
python2 已经是最新版 (2.7.18-3)。
schedtool 已经是最新版 (1.3.0-4)。
curl 已经是最新版 (7.81.0-1ubuntu1.16)。
git 已经是最新版 (1:2.34.1-1ubuntu1.10)。
gnupg 已经是最新版 (2.2.27-3ubuntu2.1)。
lib32ncurses-dev 已经是最新版 (6.3-2ubuntu0.1)。
lib32z1-dev 已经是最新版 (1:1.2.11.dfsg-2ubuntu9.2)。
libc6-dev 已经是最新版 (2.35-0ubuntu3.7)。
libncurses5-dev 已经是最新版 (6.3-2ubuntu0.1)。
libssl-dev 已经是最新版 (3.0.2-0ubuntu1.15)。
libxml2 已经是最新版 (2.9.13+dfsg-1ubuntu0.4)。
libxml2-utils 已经是最新版 (2.9.13+dfsg-1ubuntu0.4)。
unzip 已经是最新版 (6.0-26ubuntu3.2)。
xsltproc 已经是最新版 (1.1.34-4ubuntu0.22.04.1)。
zlib1g-dev 已经是最新版 (1:1.2.11.dfsg-2ubuntu9.2)。
libncurses5 已经是最新版 (6.3-2ubuntu0.1)。
升级了 0 个软件包，新安装了 0 个软件包，要卸载 0 个软件包，有 41 个软件包未被升级。
```

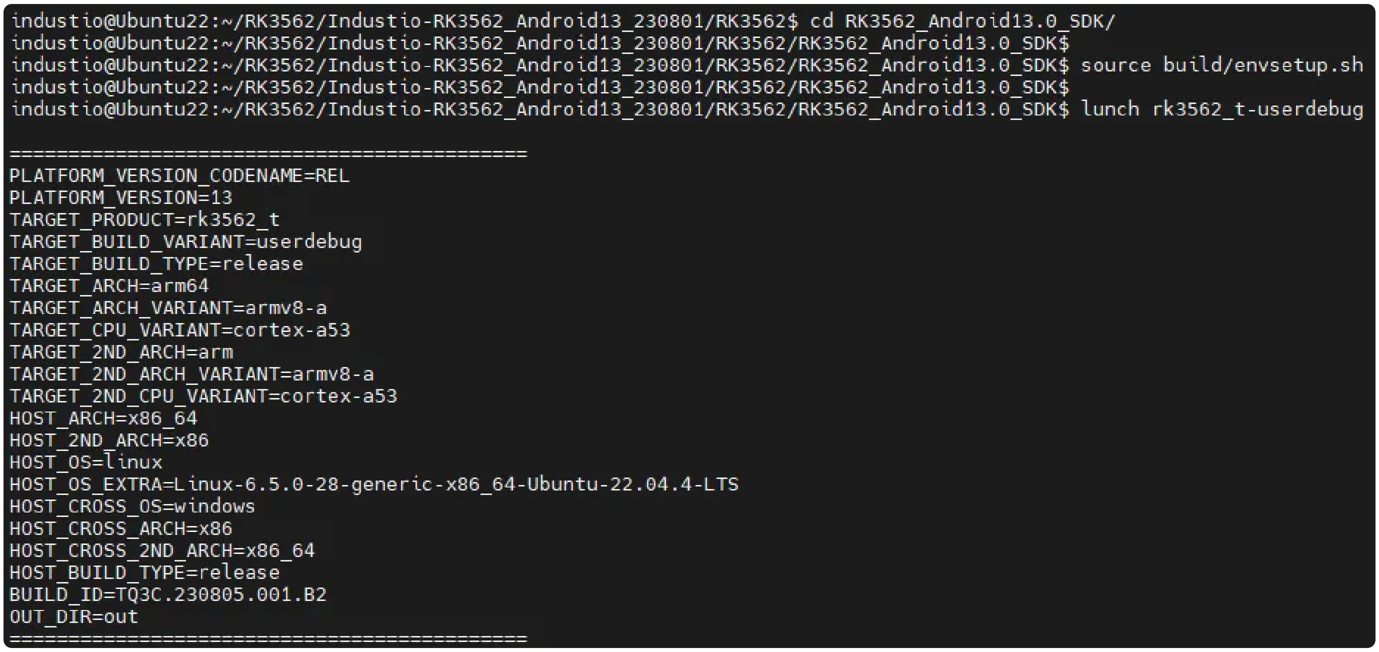
3 SDK编译

3.1 一键编译

进入 sdk 根目录执行命令，可以使用编译脚本，一次编译uboot、kernel、system等分区，命令如下：

```Shell
$ cd RK3562_Android13.0_SDK/
$ source build/envsetup.sh
$ lunch rk3562_t-userdebug
```

结果如下图所示：

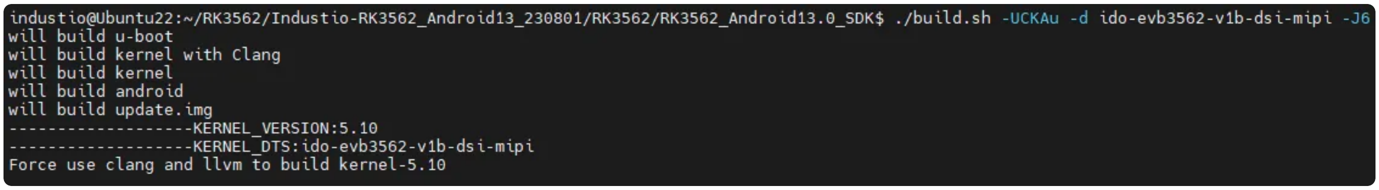


```
industio@Ubuntu22:~/RK3562/Industio-RK3562_Android13_230801/RK3562$ cd RK3562_Android13.0_SDK/
industio@Ubuntu22:~/RK3562/Industio-RK3562_Android13_230801/RK3562/RK3562_Android13.0_SDK$
industio@Ubuntu22:~/RK3562/Industio-RK3562_Android13_230801/RK3562/RK3562_Android13.0_SDK$ source build/envsetup.sh
industio@Ubuntu22:~/RK3562/Industio-RK3562_Android13_230801/RK3562/RK3562_Android13.0_SDK$
industio@Ubuntu22:~/RK3562/Industio-RK3562_Android13_230801/RK3562/RK3562_Android13.0_SDK$ lunch rk3562_t-userdebug

============================================
PLATFORM_VERSION_CODENAME=REL
PLATFORM_VERSION=13
TARGET_PRODUCT=rk3562_t
TARGET_BUILD_VARIANT=userdebug
TARGET_BUILD_TYPE=release
TARGET_ARCH=arm64
TARGET_ARCH_VARIANT=armv8-a
TARGET_CPU_VARIANT=cortex-a53
TARGET_2ND_ARCH=arm
TARGET_2ND_ARCH_VARIANT=armv8-a
TARGET_2ND_CPU_VARIANT=cortex-a53
HOST_ARCH=x86_64
HOST_2ND_ARCH=x86
HOST_OS=linux
HOST_OS_EXTRA=Linux-6.5.0-28-generic-x86_64-Ubuntu-22.04.4-LTS
HOST_CROSS_OS=windows
HOST_CROSS_ARCH=x86
HOST_CROSS_2ND_ARCH=x86_64
HOST_BUILD_TYPE=release
BUILD_ID=TQ3C.230805.001.B2
OUT_DIR=out
============================================
```

编译命令如下：

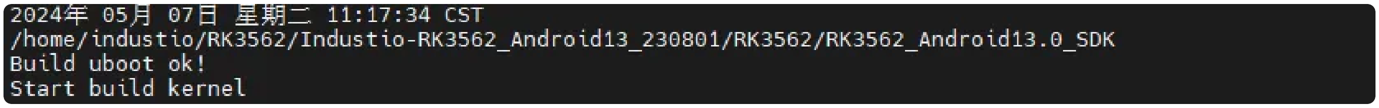
```Plain Text
$ ./build.sh -UCKAu -d ido-evb3562-v1b-dsi-mipi -J6
```

结果如下图所示：



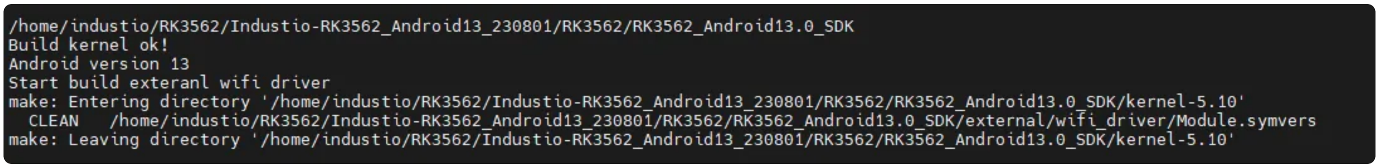
```
industio@Ubuntu22:~/RK3562/Industio-RK3562_Android13_230801/RK3562/RK3562_Android13.0_SDK$ ./build.sh -UCKAu -d ido-evb3562-v1b-dsi-mipi -J6
will build u-boot
will build kernel with Clang
will build kernel
will build android
will build update.img
-------------------KERNEL_VERSION:5.10
-------------------KERNEL_DTS:ido-evb3562-v1b-dsi-mipi
Force use clang and llvm to build kernel-5.10
```

uboot编译成功，如下图所示：



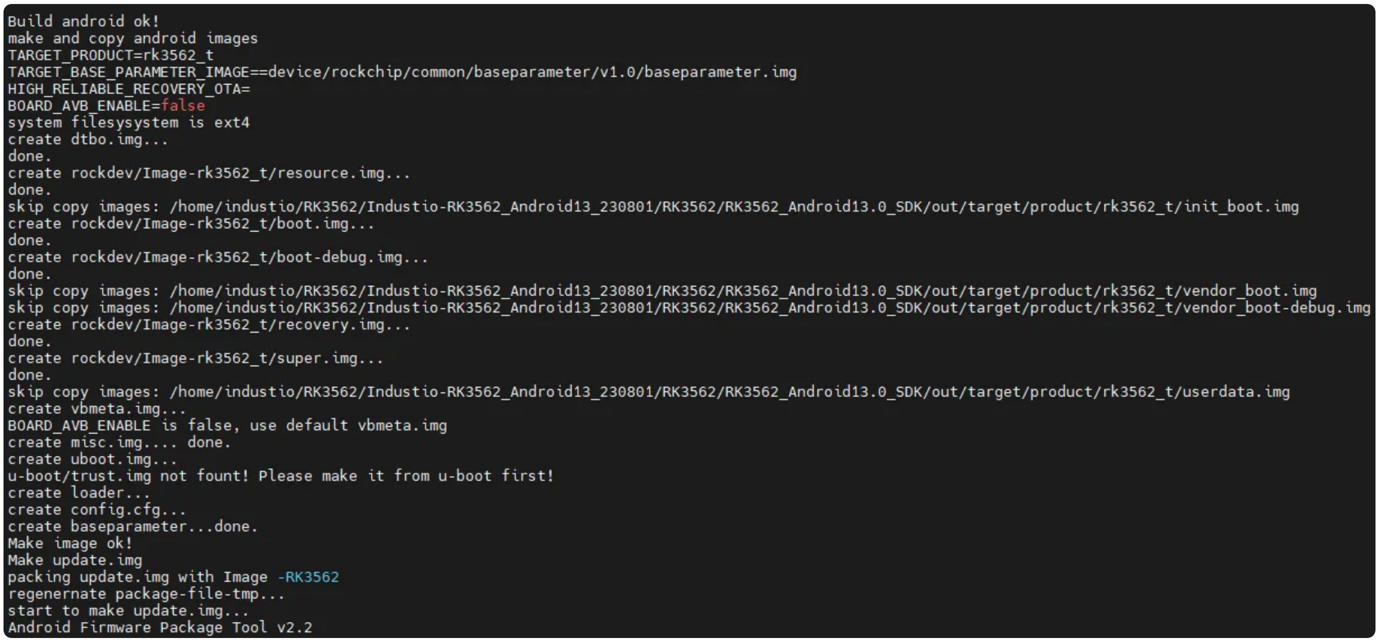
```
2024年 05月 07日 星期二 11:17:34 CST
/home/industio/RK3562/Industio-RK3562_Android13_230801/RK3562/RK3562_Android13.0_SDK
Build uboot ok!
Start build kernel
```

编译kernel成功，如下图所示：



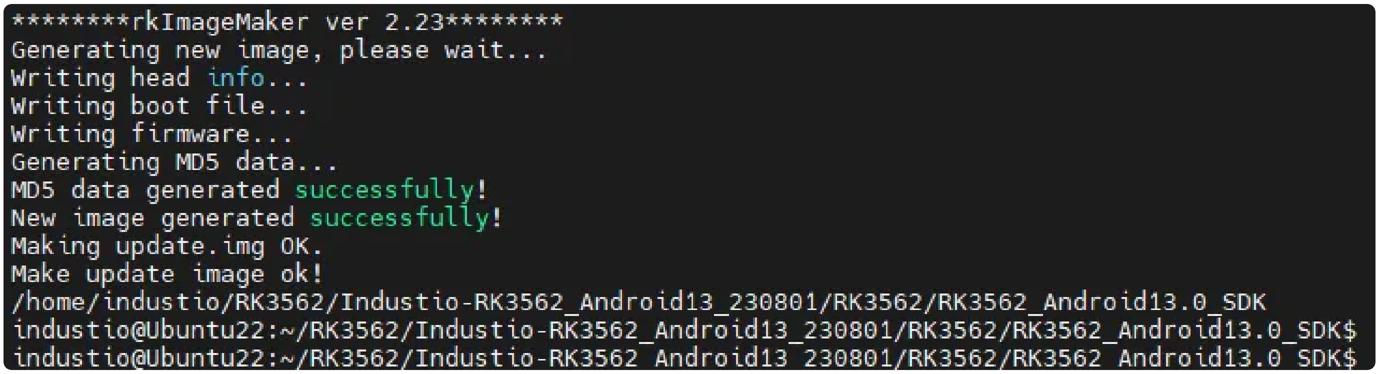
```
/home/industio/RK3562/Industio-RK3562_Android13_230801/RK3562/RK3562_Android13.0_SDK
Build kernel ok!
Android version 13
Start build exteranl wifi driver
make: Entering directory '/home/industio/RK3562/Industio-RK3562_Android13_230801/RK3562/RK3562_Android13.0_SDK/kernel-5.10'
  CLEAN   /home/industio/RK3562/Industio-RK3562_Android13_230801/RK3562/RK3562_Android13.0_SDK/external/wifi_driver/Module.symvers
make: Leaving directory '/home/industio/RK3562/Industio-RK3562_Android13_230801/RK3562/RK3562_Android13.0_SDK/kernel-5.10'
```

编译Android成功，如下图所示：



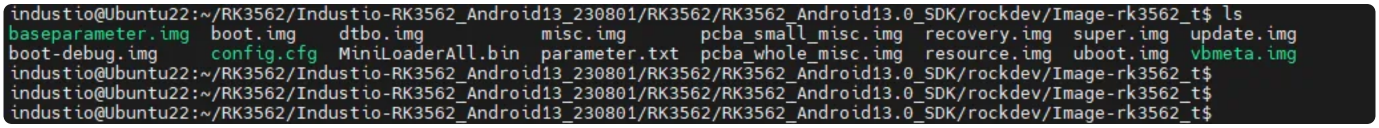
```
Build android ok!
make and copy android images
TARGET_PRODUCT=rk3562_t
TARGET_BASE_PARAMETER_IMAGE==device/rockchip/common/baseparameter/v1.0/baseparameter.img
HIGH_RELIABLE_RECOVERY_OTA=
BOARD_AVB_ENABLE=false
system filesysystem is ext4
create dtbo.img...
done.
create rockdev/Image-rk3562_t/resource.img...
done.
skip copy images: /home/industio/RK3562/Industio-RK3562_Android13_230801/RK3562/RK3562_Android13.0_SDK/out/target/product/rk3562_t/init_boot.img
create rockdev/Image-rk3562_t/boot.img...
done.
create rockdev/Image-rk3562_t/boot-debug.img...
done.
skip copy images: /home/industio/RK3562/Industio-RK3562_Android13_230801/RK3562/RK3562_Android13.0_SDK/out/target/product/rk3562_t/vendor_boot.img
skip copy images: /home/industio/RK3562/Industio-RK3562_Android13_230801/RK3562/RK3562_Android13.0_SDK/out/target/product/rk3562_t/vendor_boot-debug.img
create rockdev/Image-rk3562_t/recovery.img...
done.
create rockdev/Image-rk3562_t/super.img...
done.
skip copy images: /home/industio/RK3562/Industio-RK3562_Android13_230801/RK3562/RK3562_Android13.0_SDK/out/target/product/rk3562_t/userdata.img
create vbmeta.img...
BOARD_AVB_ENABLE is false, use default vbmeta.img
create misc.img.... done.
create uboot.img...
u-boot/trust.img not fount! Please make it from u-boot first!
create loader...
create config.cfg...
create baseparameter...done.
Make image ok!
Make update.img
packing update.img with Image -RK3562
regenernate package-file-tmp...
start to make update.img...
Android Firmware Package Tool v2.2
```

编译完成后结果，如下图所示：



```
********rkImageMaker ver 2.23********
Generating new image, please wait...
Writing head info...
Writing boot file...
Writing firmware...
Generating MD5 data...
MD5 data generated successfully!
New image generated successfully!
Making update.img OK.
Make update image ok!
/home/industio/RK3562/Industio-RK3562_Android13_230801/RK3562/RK3562_Android13.0_SDK
industio@Ubuntu22:~/RK3562/Industio-RK3562_Android13_230801/RK3562/RK3562_Android13.0_SDK$
industio@Ubuntu22:~/RK3562/Industio-RK3562_Android13_230801/RK3562/RK3562_Android13.0_SDK$
```

固件所在目录为/rockdev/Image–rk3562_t/，update.img为整包固件，结果如下图所示：



```
industio@Ubuntu22:~/RK3562/Industio-RK3562_Android13_230801/RK3562/RK3562_Android13.0_SDK/rockdev/Image-rk3562_t$ ls
baseparameter.img  boot.img    dtbo.img           misc.img       pcba_small_misc.img  recovery.img  super.img  update.img
boot-debug.img     config.cfg  MiniLoaderAll.bin  parameter.txt  pcba_whole_misc.img  resource.img  uboot.img  vbmeta.img
industio@Ubuntu22:~/RK3562/Industio-RK3562_Android13_230801/RK3562/RK3562_Android13.0_SDK/rockdev/Image-rk3562_t$
industio@Ubuntu22:~/RK3562/Industio-RK3562_Android13_230801/RK3562/RK3562_Android13.0_SDK/rockdev/Image-rk3562_t$
industio@Ubuntu22:~/RK3562/Industio-RK3562_Android13_230801/RK3562/RK3562_Android13.0_SDK/rockdev/Image-rk3562_t$
```

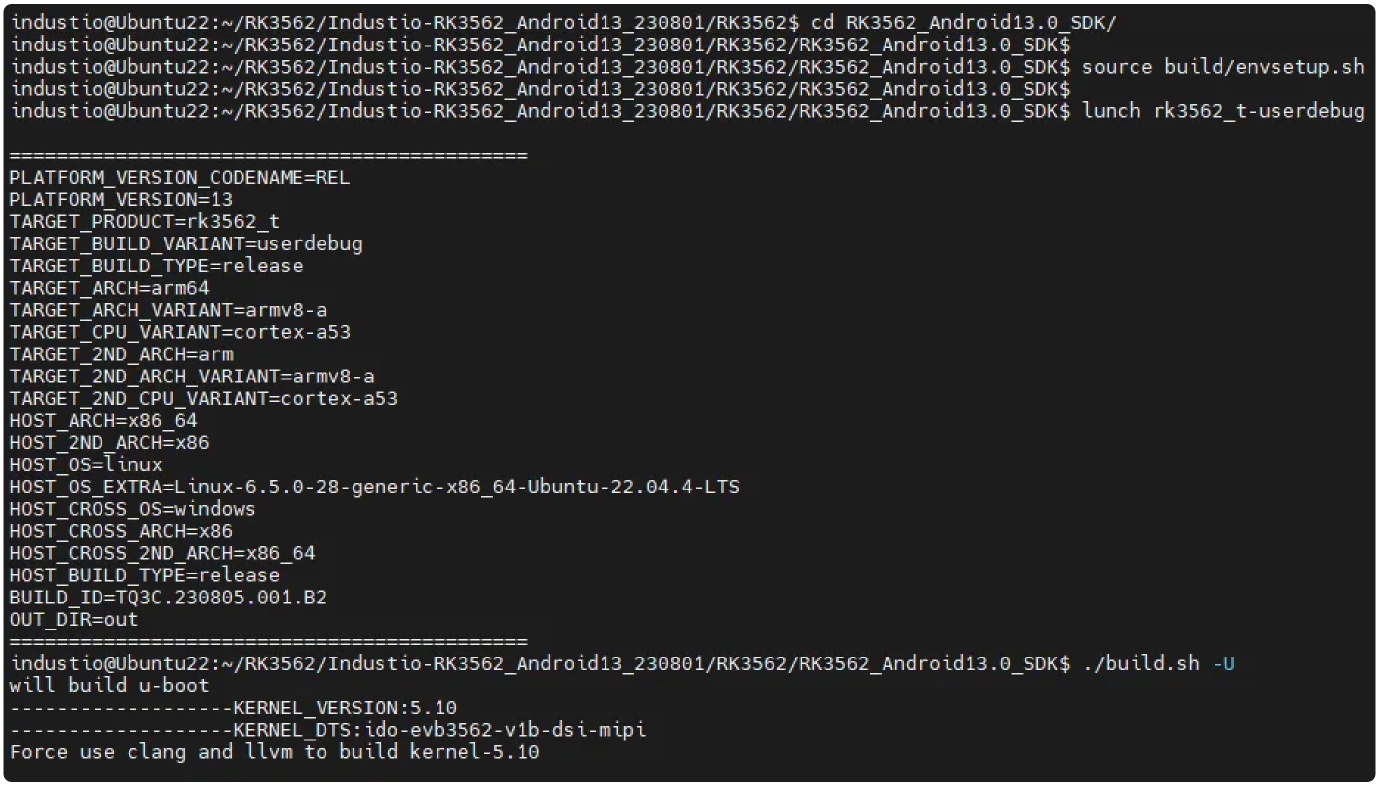
3.2 单独编译

3.2.1 uboot编译步骤

进入 sdk 根目录执行命令。使用编译脚本编译，命令如下：

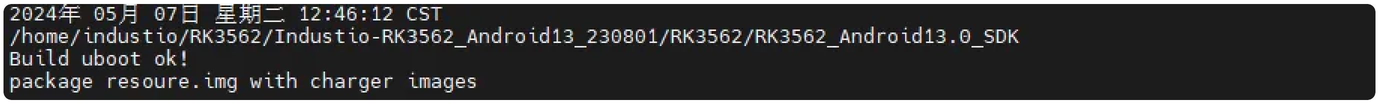
```bash
$ cd RK3562_Android13.0_SDK/
$ source build/envsetup.sh
$ lunch rk3562_t-userdebug
$ ./build.sh -U
```

结果如下图所示：



```
industio@Ubuntu22:~/RK3562/Industio-RK3562_Android13_230801/RK3562$ cd RK3562_Android13.0_SDK/
industio@Ubuntu22:~/RK3562/Industio-RK3562_Android13_230801/RK3562/RK3562_Android13.0_SDK$
industio@Ubuntu22:~/RK3562/Industio-RK3562_Android13_230801/RK3562/RK3562_Android13.0_SDK$ source build/envsetup.sh
industio@Ubuntu22:~/RK3562/Industio-RK3562_Android13_230801/RK3562/RK3562_Android13.0_SDK$
industio@Ubuntu22:~/RK3562/Industio-RK3562_Android13_230801/RK3562/RK3562_Android13.0_SDK$ lunch rk3562_t-userdebug

============================================
PLATFORM_VERSION_CODENAME=REL
PLATFORM_VERSION=13
TARGET_PRODUCT=rk3562_t
TARGET_BUILD_VARIANT=userdebug
TARGET_BUILD_TYPE=release
TARGET_ARCH=arm64
TARGET_ARCH_VARIANT=armv8-a
TARGET_CPU_VARIANT=cortex-a53
TARGET_2ND_ARCH=arm
TARGET_2ND_ARCH_VARIANT=armv8-a
TARGET_2ND_CPU_VARIANT=cortex-a53
HOST_ARCH=x86_64
HOST_2ND_ARCH=x86
HOST_OS=linux
HOST_OS_EXTRA=Linux-6.5.0-28-generic-x86_64-Ubuntu-22.04.4-LTS
HOST_CROSS_OS=windows
HOST_CROSS_ARCH=x86
HOST_CROSS_2ND_ARCH=x86_64
HOST_BUILD_TYPE=release
BUILD_ID=TQ3C.230805.001.B2
OUT_DIR=out
============================================
industio@Ubuntu22:~/RK3562/Industio-RK3562_Android13_230801/RK3562/RK3562_Android13.0_SDK$ ./build.sh -U
will build u-boot
-------------------KERNEL_VERSION:5.10
-------------------KERNEL_DTS:ido-evb3562-v1b-dsi-mipi
Force use clang and llvm to build kernel-5.10
```

编译成功结果，如下图所示：

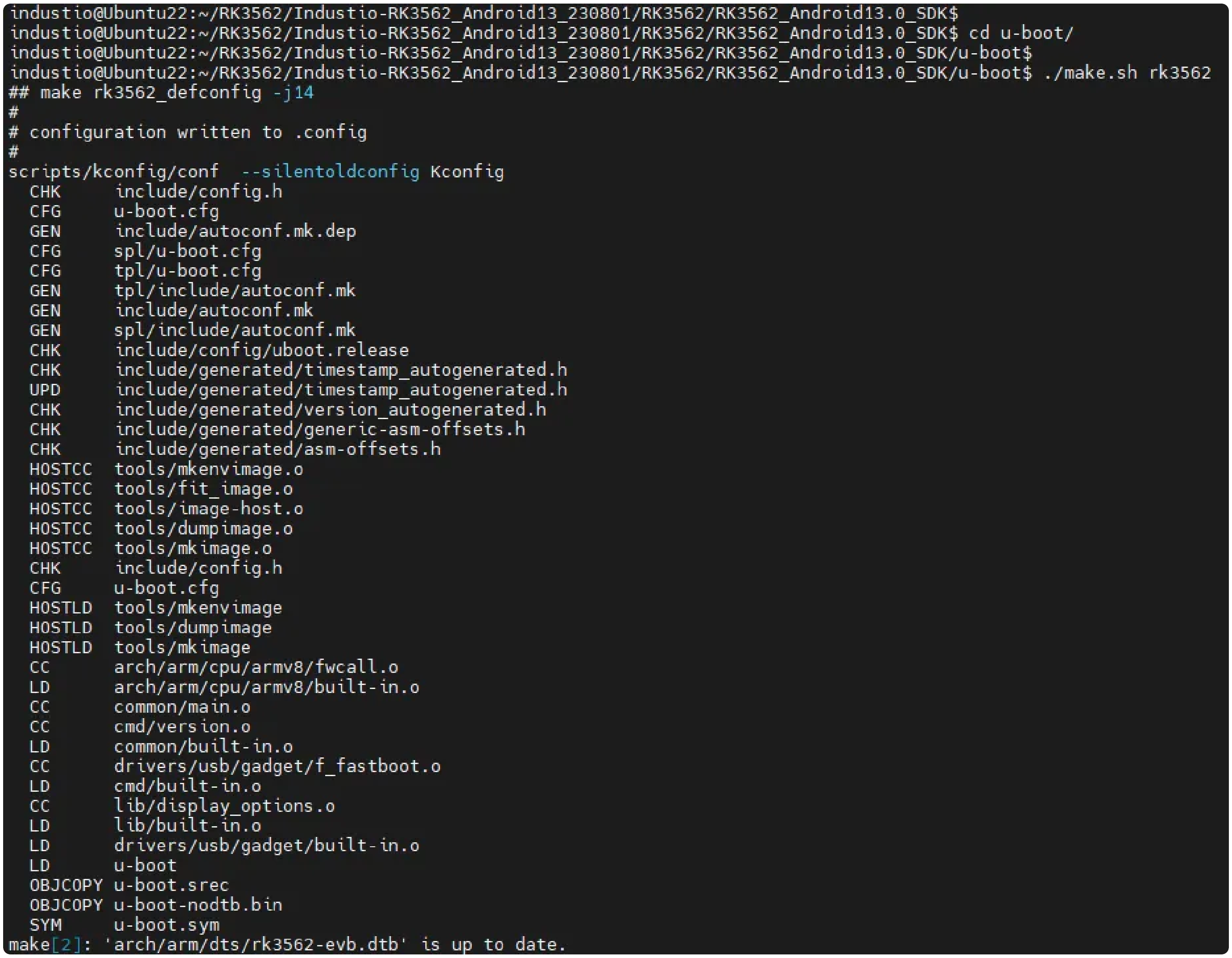


```
2024年 05月 07日 星期二 12:46:12 CST
/home/industio/RK3562/Industio-RK3562_Android13_230801/RK3562/RK3562_Android13.0_SDK
Build uboot ok!
package resoure.img with charger images
```

手动执行命令编译，命令如下：

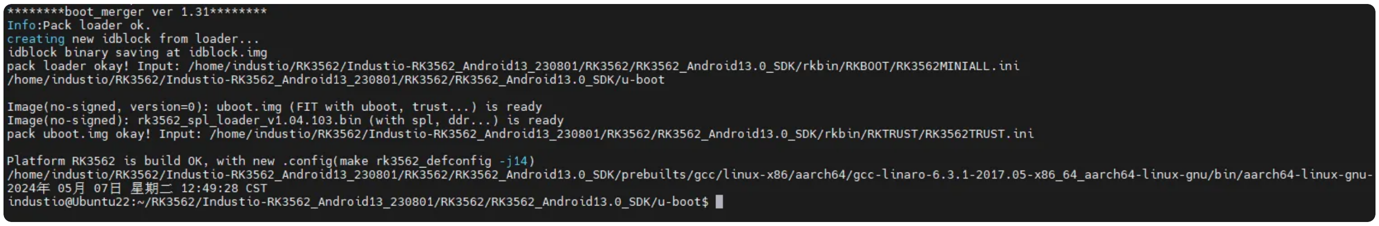
```shell
$ cd u-boot
$ ./make.sh rk3562
```

结果如下图所示：



```
industio@Ubuntu22:~/RK3562/Industio-RK3562_Android13_230801/RK3562/RK3562_Android13.0_SDK$
industio@Ubuntu22:~/RK3562/Industio-RK3562_Android13_230801/RK3562/RK3562_Android13.0_SDK$ cd u-boot/
industio@Ubuntu22:~/RK3562/Industio-RK3562_Android13_230801/RK3562/RK3562_Android13.0_SDK/u-boot$
industio@Ubuntu22:~/RK3562/Industio-RK3562_Android13_230801/RK3562/RK3562_Android13.0_SDK/u-boot$ ./make.sh rk3562
## make rk3562_defconfig -j14
#
# configuration written to .config
#
scripts/kconfig/conf  --silentoldconfig Kconfig
  CHK     include/config.h
  CFG     u-boot.cfg
  GEN     include/autoconf.mk.dep
  CFG     spl/u-boot.cfg
  CFG     tpl/u-boot.cfg
  GEN     tpl/include/autoconf.mk
  GEN     include/autoconf.mk
  GEN     spl/include/autoconf.mk
  CHK     include/config/uboot.release
  CHK     include/generated/timestamp_autogenerated.h
  UPD     include/generated/timestamp_autogenerated.h
  CHK     include/generated/version_autogenerated.h
  CHK     include/generated/generic-asm-offsets.h
  CHK     include/generated/asm-offsets.h
  HOSTCC  tools/mkenvimage.o
  HOSTCC  tools/fit_image.o
  HOSTCC  tools/image-host.o
  HOSTCC  tools/dumpimage.o
  HOSTCC  tools/mkimage.o
  CHK     include/config.h
  CFG     u-boot.cfg
  HOSTLD  tools/mkenvimage
  HOSTLD  tools/dumpimage
  HOSTLD  tools/mkimage
  CC      arch/arm/cpu/armv8/fwcall.o
  LD      arch/arm/cpu/armv8/built-in.o
  CC      common/main.o
  CC      cmd/version.o
  LD      common/built-in.o
  CC      drivers/usb/gadget/f_fastboot.o
  LD      cmd/built-in.o
  CC      lib/display_options.o
  LD      lib/built-in.o
  LD      drivers/usb/gadget/built-in.o
  LD      u-boot
  OBJCOPY u-boot.srec
  OBJCOPY u-boot-nodtb.bin
  SYM     u-boot.sym
make[2]: 'arch/arm/dts/rk3562-evb.dtb' is up to date.
```

编译结果，如下图所示：



```
********boot_merger ver 1.31********
Info:Pack loader ok.
creating new idblock from loader...
idblock binary saving at idblock.img
pack loader okay! Input: /home/industio/RK3562/Industio-RK3562_Android13_230801/RK3562/RK3562_Android13.0_SDK/rkbin/RKBOOT/RK3562MINIALL.ini
/home/industio/RK3562/Industio-RK3562_Android13_230801/RK3562/RK3562_Android13.0_SDK/u-boot

Image(no-signed, version=0): uboot.img (FIT with uboot, trust...) is ready
Image(no-signed): rk3562_spl_loader_v1.04.103.bin (with spl, ddr...) is ready
pack uboot.img okay! Input: /home/industio/RK3562/Industio-RK3562_Android13_230801/RK3562/RK3562_Android13.0_SDK/rkbin/RKTRUST/RK3562TRUST.ini

Platform RK3562 is build OK, with new .config(make rk3562_defconfig -j14)
/home/industio/RK3562/Industio-RK3562_Android13_230801/RK3562/RK3562_Android13.0_SDK/prebuilts/gcc/linux-x86/aarch64/gcc-linaro-6.3.1-2017.05-x86_64_aarch64-linux-gnu/bin/aarch64-linux-gnu-
2024年 05月 07日 星期二 12:49:28 CST
industio@Ubuntu22:~/RK3562/Industio-RK3562_Android13_230801/RK3562/RK3562_Android13.0_SDK/u-boot$ █
```

3.2.2 kernel编译步骤

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

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

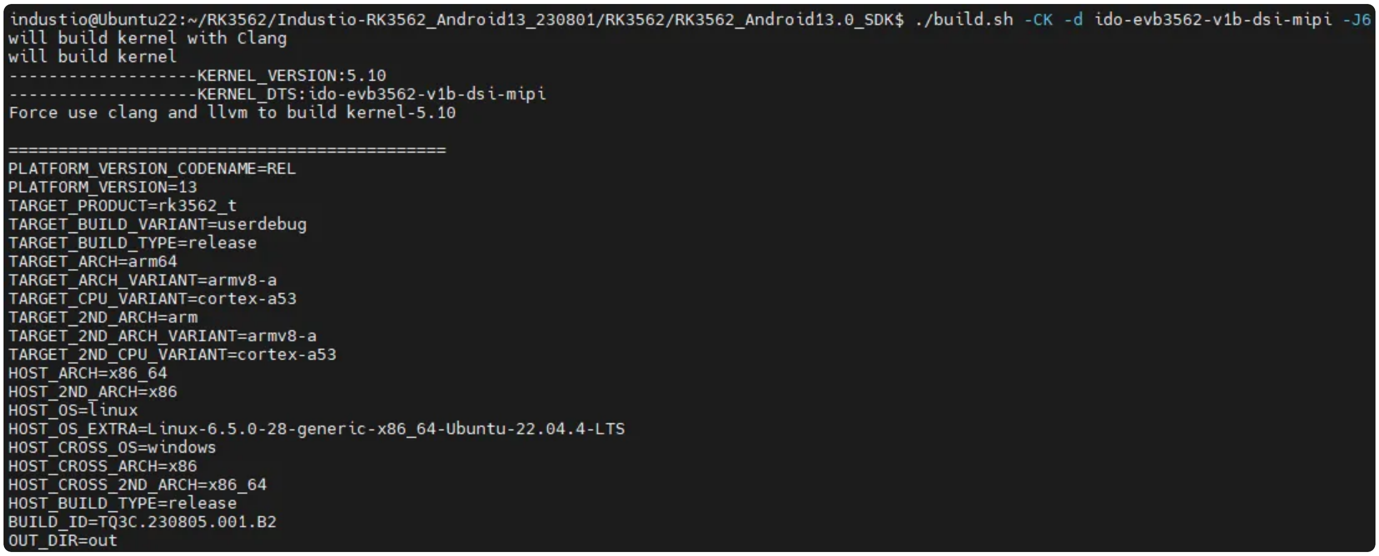
内核配置及设备树文件名，列表如下：

序号	文件名	功能说明

1	ido-evb3562-v1b-lvds.dts	显示为LVDS屏
2	ido-evb3562-v1b-dsi-mipi.dts	显示为MIPI屏

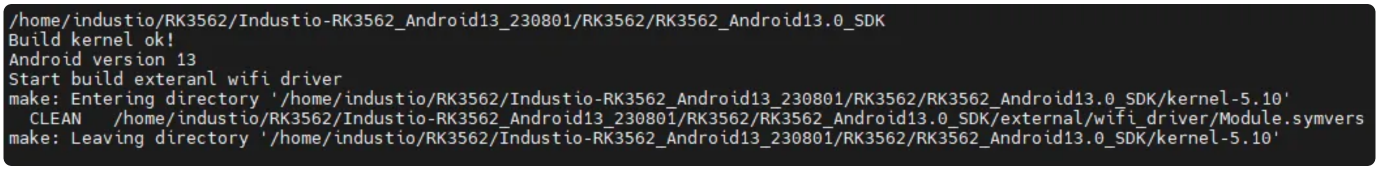
以编译MIPI屏dts为例，编译命令如下：

```Shell
./build.sh -CK -d ido-evb3562-v1b-dsi-mipi -J6
```



```
industio@Ubuntu22:~/RK3562/Industio-RK3562_Android13_230801/RK3562/RK3562_Android13.0_SDK$ ./build.sh -CK -d ido-evb3562-v1b-dsi-mipi -J6
will build kernel with Clang
will build kernel
-------------------KERNEL_VERSION:5.10
-------------------KERNEL_DTS:ido-evb3562-v1b-dsi-mipi
Force use clang and llvm to build kernel-5.10

============================================
PLATFORM_VERSION_CODENAME=REL
PLATFORM_VERSION=13
TARGET_PRODUCT=rk3562_t
TARGET_BUILD_VARIANT=userdebug
TARGET_BUILD_TYPE=release
TARGET_ARCH=arm64
TARGET_ARCH_VARIANT=armv8-a
TARGET_CPU_VARIANT=cortex-a53
TARGET_2ND_ARCH=arm
TARGET_2ND_ARCH_VARIANT=armv8-a
TARGET_2ND_CPU_VARIANT=cortex-a53
HOST_ARCH=x86_64
HOST_2ND_ARCH=x86
HOST_OS=linux
HOST_OS_EXTRA=Linux-6.5.0-28-generic-x86_64-Ubuntu-22.04.4-LTS
HOST_CROSS_OS=windows
HOST_CROSS_ARCH=x86
HOST_CROSS_2ND_ARCH=x86_64
HOST_BUILD_TYPE=release
BUILD_ID=TQ3C.230805.001.B2
OUT_DIR=out
```

编译成功结果，如下图所示：



```
/home/industio/RK3562/Industio-RK3562_Android13_230801/RK3562/RK3562_Android13.0_SDK
Build kernel ok!
Android version 13
Start build exteranl wifi driver
make: Entering directory '/home/industio/RK3562/Industio-RK3562_Android13_230801/RK3562/RK3562_Android13.0_SDK/kernel-5.10'
  CLEAN   /home/industio/RK3562/Industio-RK3562_Android13_230801/RK3562/RK3562_Android13.0_SDK/external/wifi_driver/Module.symvers
make: Leaving directory '/home/industio/RK3562/Industio-RK3562_Android13_230801/RK3562/RK3562_Android13.0_SDK/kernel-5.10'
```

单独编译kernel生成可直接烧录的boot.img

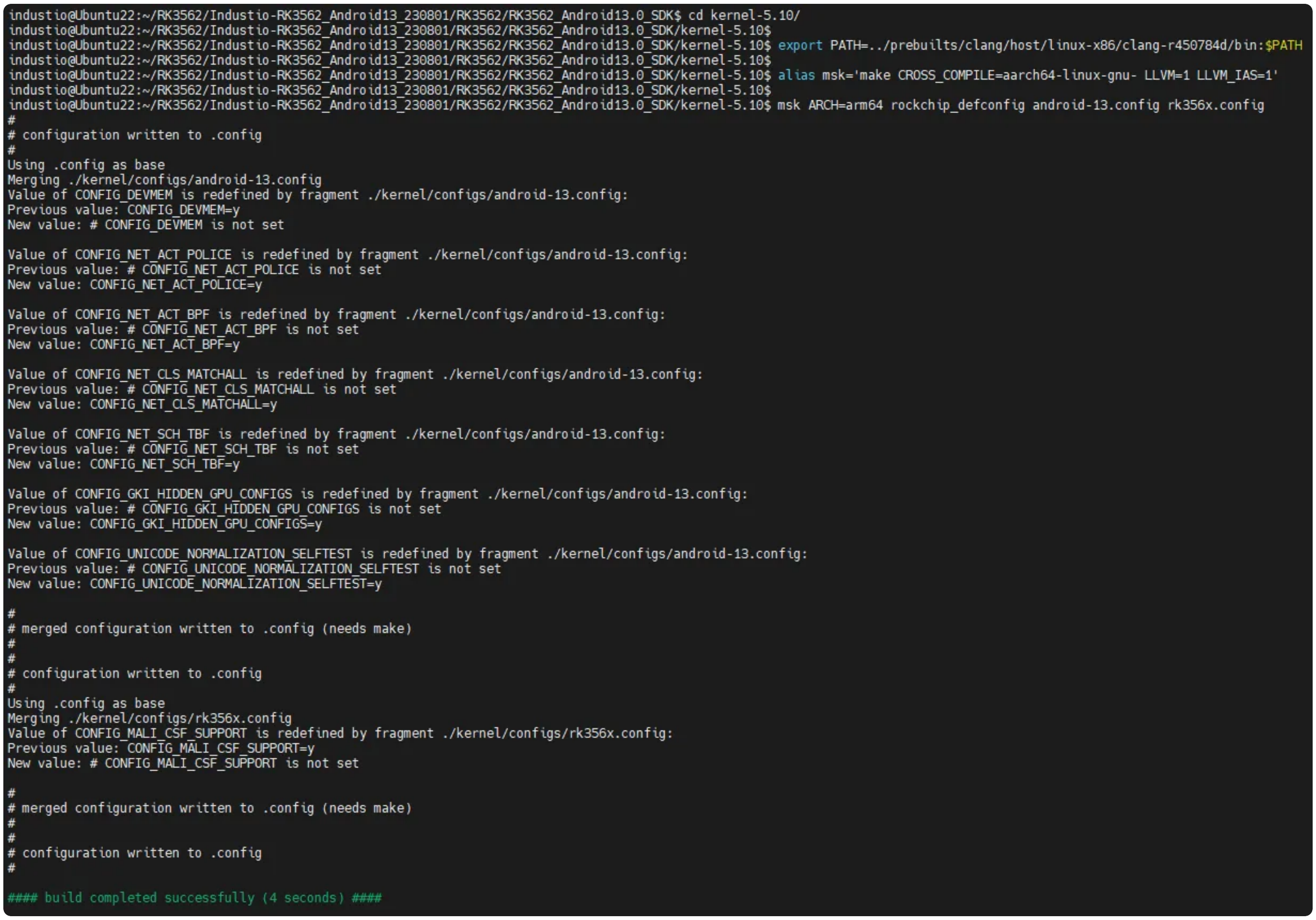
此处的编译方法的前提已存在rockdev/Image-rk3562_t/boot.img文件（即Android代码已经完全编译过一次或者执行过build.sh -K）。

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，命令如下：

Shell

```
$ cd kernel-5.10
$ export PATH=../prebuilts/clang/host/linux-x86/clang-r450784d/bin:$PATH
$ alias msk='make CROSS_COMPILE=aarch64-linux-gnu- LLVM=1 LLVM_IAS=1'
$ msk ARCH=arm64 rockchip_defconfig android-13.config rk356x.config
```

结果如下图所示：



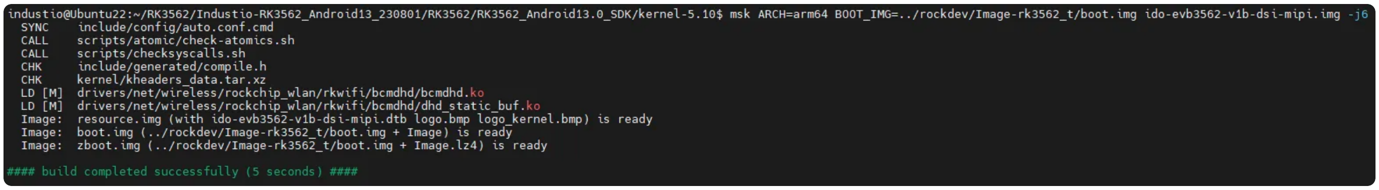
```
industio@Ubuntu22:~/RK3562/Industio-RK3562_Android13_230801/RK3562/RK3562_Android13.0_SDK$ cd kernel-5.10/
industio@Ubuntu22:~/RK3562/Industio-RK3562_Android13_230801/RK3562/RK3562_Android13.0_SDK/kernel-5.10$
industio@Ubuntu22:~/RK3562/Industio-RK3562_Android13_230801/RK3562/RK3562_Android13.0_SDK/kernel-5.10$ export PATH=../prebuilts/clang/host/linux-x86/clang-r450784d/bin:$PATH
industio@Ubuntu22:~/RK3562/Industio-RK3562_Android13_230801/RK3562/RK3562_Android13.0_SDK/kernel-5.10$
industio@Ubuntu22:~/RK3562/Industio-RK3562_Android13_230801/RK3562/RK3562_Android13.0_SDK/kernel-5.10$ alias msk='make CROSS_COMPILE=aarch64-linux-gnu- LLVM=1 LLVM_IAS=1'
industio@Ubuntu22:~/RK3562/Industio-RK3562_Android13_230801/RK3562/RK3562_Android13.0_SDK/kernel-5.10$
industio@Ubuntu22:~/RK3562/Industio-RK3562_Android13_230801/RK3562/RK3562_Android13.0_SDK/kernel-5.10$ msk ARCH=arm64 rockchip_defconfig android-13.config rk356x.config
#
# configuration written to .config
#
Using .config as base
Merging ./kernel/configs/android-13.config
Value of CONFIG_DEVMEM is redefined by fragment ./kernel/configs/android-13.config:
Previous value: CONFIG_DEVMEM=y
New value: # CONFIG_DEVMEM is not set

Value of CONFIG_NET_ACT_POLICE is redefined by fragment ./kernel/configs/android-13.config:
Previous value: # CONFIG_NET_ACT_POLICE is not set
New value: CONFIG_NET_ACT_POLICE=y

Value of CONFIG_NET_ACT_BPF is redefined by fragment ./kernel/configs/android-13.config:
Previous value: # CONFIG_NET_ACT_BPF is not set
New value: CONFIG_NET_ACT_BPF=y

Value of CONFIG_NET_CLS_MATCHALL is redefined by fragment ./kernel/configs/android-13.config:
Previous value: # CONFIG_NET_CLS_MATCHALL is not set
New value: CONFIG_NET_CLS_MATCHALL=y

Value of CONFIG_NET_SCH_TBF is redefined by fragment ./kernel/configs/android-13.config:
Previous value: # CONFIG_NET_SCH_TBF is not set
New value: CONFIG_NET_SCH_TBF=y

Value of CONFIG_GKI_HIDDEN_GPU_CONFIGS is redefined by fragment ./kernel/configs/android-13.config:
Previous value: # CONFIG_GKI_HIDDEN_GPU_CONFIGS is not set
New value: CONFIG_GKI_HIDDEN_GPU_CONFIGS=y

Value of CONFIG_UNICODE_NORMALIZATION_SELFTEST is redefined by fragment ./kernel/configs/android-13.config:
Previous value: # CONFIG_UNICODE_NORMALIZATION_SELFTEST is not set
New value: CONFIG_UNICODE_NORMALIZATION_SELFTEST=y

#
# merged configuration written to .config (needs make)
#
#
# configuration written to .config
#
Using .config as base
Merging ./kernel/configs/rk356x.config
Value of CONFIG_MALI_CSF_SUPPORT is redefined by fragment ./kernel/configs/rk356x.config:
Previous value: CONFIG_MALI_CSF_SUPPORT=y
New value: # CONFIG_MALI_CSF_SUPPORT is not set

#
# merged configuration written to .config (needs make)
#
#
# configuration written to .config
#

#### build completed successfully (4 seconds) ####
```

命令如下：

Shell

```
$ msk ARCH=arm64 BOOT_IMG=../rockdev/Image-rk3562_t/boot.img ido-evb3562-v1b-dsi-mipi.img -j6
```

结果如下图所示：



```
industio@Ubuntu22:~/RK3562/Industio-RK3562_Android13_230801/RK3562/RK3562_Android13.0_SDK/kernel-5.10$ msk ARCH=arm64 BOOT_IMG=../rockdev/Image-rk3562_t/boot.img ido-evb3562-v1b-dsi-mipi.img -j6
  SYNC    include/config/auto.conf.cmd
  CALL    scripts/atomic/check-atomics.sh
  CALL    scripts/checksyscalls.sh
  CHK     include/generated/compile.h
  CHK     kernel/kheaders_data.tar.xz
  LD [M]  drivers/net/wireless/rockchip_wlan/rkwifi/bcmdhd/bcmdhd.ko
  LD [M]  drivers/net/wireless/rockchip_wlan/rkwifi/bcmdhd/dhd_static_buf.ko
  Image:  resource.img (with ido-evb3562-v1b-dsi-mipi.dtb logo.bmp logo_kernel.bmp) is ready
  Image:  boot.img (../rockdev/Image-rk3562_t/boot.img + Image) is ready
  Image:  zboot.img (../rockdev/Image-rk3562_t/boot.img + Image.lz4) is ready

#### build completed successfully (5 seconds) ####
```

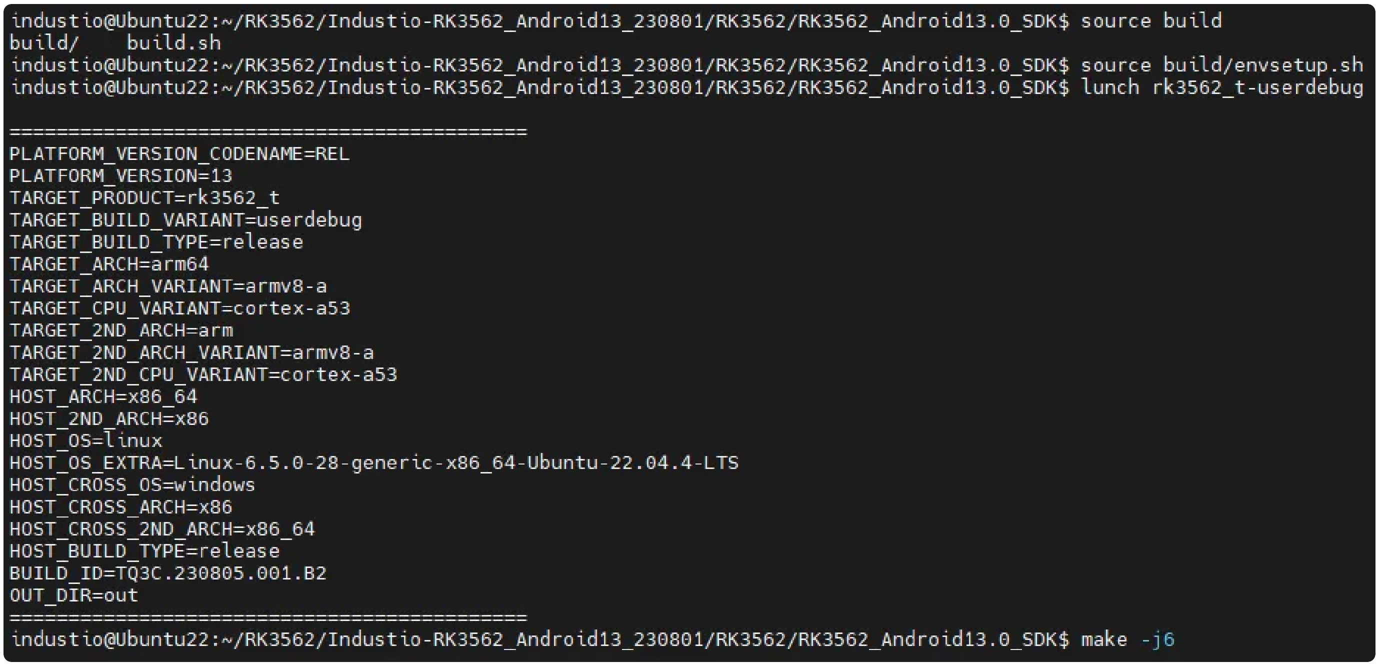
使用此方法编译出kernel-5.10/boot.img文件可以直接用于烧录至boot分区。

3.2.3 Android编译步骤

Android编译命令如下：

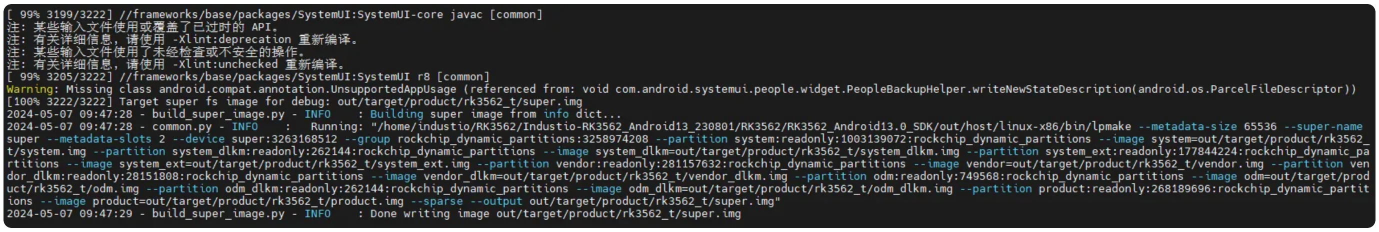
```Shell
$ source build/envsetup.sh
$ lunch rk3562_t-userdebug
$ make -jx
```

结果如下图所示



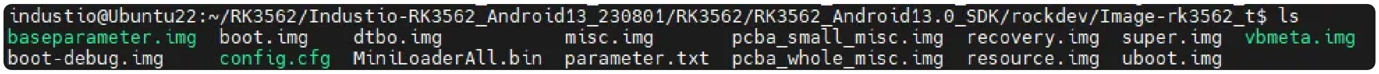
```
industio@Ubuntu22:~/RK3562/Industio-RK3562_Android13_230801/RK3562/RK3562_Android13.0_SDK$ source build
build/    build.sh
industio@Ubuntu22:~/RK3562/Industio-RK3562_Android13_230801/RK3562/RK3562_Android13.0_SDK$ source build/envsetup.sh
industio@Ubuntu22:~/RK3562/Industio-RK3562_Android13_230801/RK3562/RK3562_Android13.0_SDK$ lunch rk3562_t-userdebug

============================================
PLATFORM_VERSION_CODENAME=REL
PLATFORM_VERSION=13
TARGET_PRODUCT=rk3562_t
TARGET_BUILD_VARIANT=userdebug
TARGET_BUILD_TYPE=release
TARGET_ARCH=arm64
TARGET_ARCH_VARIANT=armv8-a
TARGET_CPU_VARIANT=cortex-a53
TARGET_2ND_ARCH=arm
TARGET_2ND_ARCH_VARIANT=armv8-a
TARGET_2ND_CPU_VARIANT=cortex-a53
HOST_ARCH=x86_64
HOST_2ND_ARCH=x86
HOST_OS=linux
HOST_OS_EXTRA=Linux-6.5.0-28-generic-x86_64-Ubuntu-22.04.4-LTS
HOST_CROSS_OS=windows
HOST_CROSS_ARCH=x86
HOST_CROSS_2ND_ARCH=x86_64
HOST_BUILD_TYPE=release
BUILD_ID=TQ3C.230805.001.B2
OUT_DIR=out
============================================
industio@Ubuntu22:~/RK3562/Industio-RK3562_Android13_230801/RK3562/RK3562_Android13.0_SDK$ make -j6
```

编译成功结果，如下图所示：



```
[ 99% 3199/3222] //frameworks/base/packages/SystemUI:SystemUI-core javac [common]
注: 某些输入文件使用或覆盖了已过时的 API。
注: 有关详细信息, 请使用 -Xlint:deprecation 重新编译。
注: 某些输入文件使用了未经检查或不安全的操作。
注: 有关详细信息, 请使用 -Xlint:unchecked 重新编译。
[ 99% 3205/3222] //frameworks/base/packages/SystemUI:SystemUI r8 [common]
Warning: Missing class android.compat.annotation.UnsupportedAppUsage (referenced from: void com.android.systemui.people.widget.PeopleBackupHelper.writeNewStateDescription(android.os.ParcelFileDescriptor))
[100% 3222/3222] Target super fs image for debug: out/target/product/rk3562_t/super.img
2024-05-07 09:47:28 - build_super_image.py - INFO    : Building super image from info dict...
2024-05-07 09:47:28 - common.py - INFO    :   Running: "/home/industio/RK3562/Industio-RK3562_Android13_230801/RK3562/RK3562_Android13.0_SDK/out/host/linux-x86/bin/lpmake --metadata-size 65536 --super-name
super --metadata-slots 2 --device super:3263168512 --group rockchip_dynamic_partitions:3258974208 --partition system:readonly:1003139072:rockchip_dynamic_partitions --image system=out/target/product/rk3562_
t/system.img --partition system_dlkm:readonly:262144:rockchip_dynamic_partitions --image system_dlkm=out/target/product/rk3562_t/system_dlkm.img --partition system_ext:readonly:177844224:rockchip_dynamic_pa
rtitions --image system_ext=out/target/product/rk3562_t/system_ext.img --partition vendor:readonly:281157632:rockchip_dynamic_partitions --image vendor=out/target/product/rk3562_t/vendor.img --partition ven
dor_dlkm:readonly:28151808:rockchip_dynamic_partitions --image vendor_dlkm=out/target/product/rk3562_t/vendor_dlkm.img --partition odm:readonly:749568:rockchip_dynamic_partitions --image odm=out/target/prod
uct/rk3562_t/odm.img --partition odm_dlkm:readonly:262144:rockchip_dynamic_partitions --image odm_dlkm=out/target/product/rk3562_t/odm_dlkm.img --partition product:readonly:268189696:rockchip_dynamic_partit
ions --image product=out/target/product/rk3562_t/product.img --sparse --output out/target/product/rk3562_t/super.img"
2024-05-07 09:47:29 - build_super_image.py - INFO    : Done writing image out/target/product/rk3562_t/super.img
```

固件所在目录为rockdev/Image-rk3562_t/，结果如下图所示：



```
industio@Ubuntu22:~/RK3562/Industio-RK3562_Android13_230801/RK3562/RK3562_Android13.0_SDK/rockdev/Image-rk3562_t$ ls
baseparameter.img  boot.img     dtbo.img          misc.img       pcba_small_misc.img  recovery.img  super.img  vbmeta.img
boot-debug.img     config.cfg   MiniLoaderAll.bin parameter.txt  pcba_whole_misc.img  resource.img  uboot.img
```

4 驱动开发

4.1 LOGO旋转

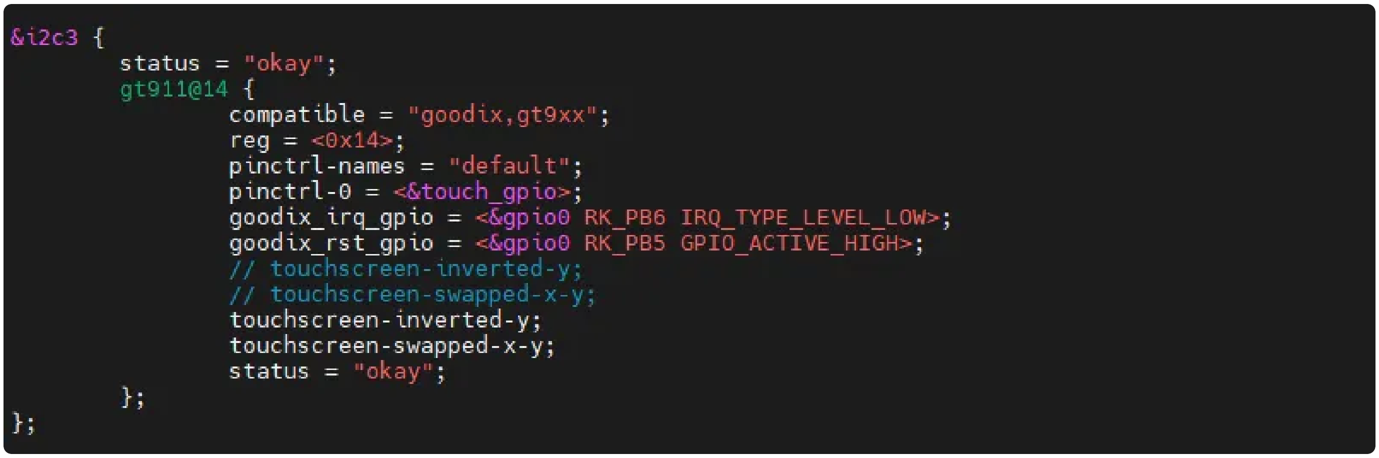
修改kernel-5.10/logo.bmp 和kernel-5.10/logo_kernel.bmp，LOGO旋转270度。

4.2 触摸旋转

触摸旋转代码修改如下：

```
--- a/kernel-5.10/arch/arm64/boot/dts/rockchip/ido-evb3562-v1b-dsi-mipi.dts
+++ b/kernel-5.10/arch/arm64/boot/dts/rockchip/ido-evb3562-v1b-dsi-mipi.dts
@@ -360,8 +360,8 @@ gt911@14 {
                pinctrl-0 = <&touch_gpio>;
                goodix_irq_gpio = <&gpio0 RK_PB6 IRQ_TYPE_LEVEL_LOW>;
                goodix_rst_gpio = <&gpio0 RK_PB5 GPIO_ACTIVE_HIGH>;
-               // touchscreen-inverted-y;
-               // touchscreen-swapped-x-y;
+               touchscreen-inverted-y;
+               touchscreen-swapped-x-y;
                status = "okay";
        };
 };
```

结果如下图所示：



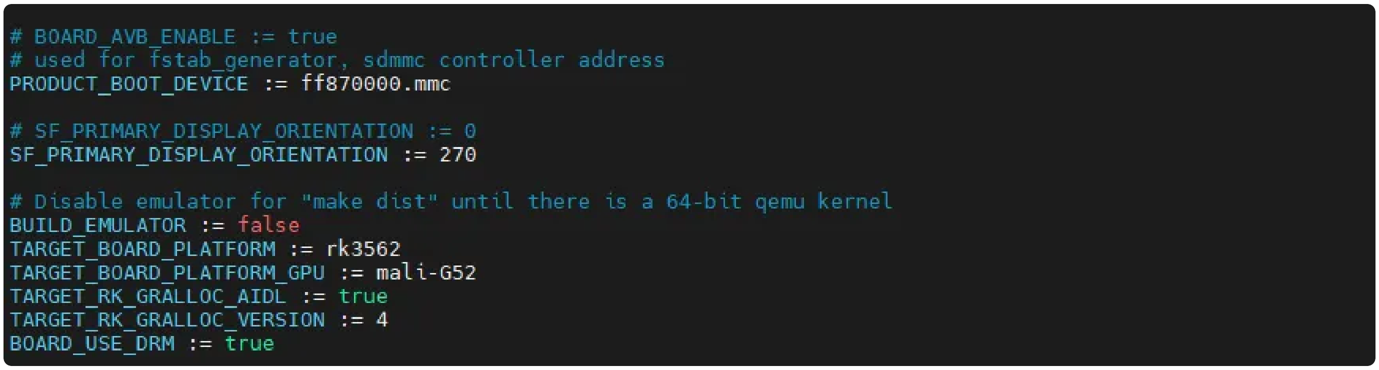
```
&i2c3 {
        status = "okay";
        gt911@14 {
                compatible = "goodix,gt9xx";
                reg = <0x14>;
                pinctrl-names = "default";
                pinctrl-0 = <&touch_gpio>;
                goodix_irq_gpio = <&gpio0 RK_PB6 IRQ_TYPE_LEVEL_LOW>;
                goodix_rst_gpio = <&gpio0 RK_PB5 GPIO_ACTIVE_HIGH>;
                // touchscreen-inverted-y;
                // touchscreen-swapped-x-y;
                touchscreen-inverted-y;
                touchscreen-swapped-x-y;
                status = "okay";
        };
};
```

4.3 系统旋转

系统旋转代码修改如下：

```java
--- a/device/rockchip/rk3562/BoardConfig.mk
+++ b/device/rockchip/rk3562/BoardConfig.mk
@@ -26,7 +26,7 @@ PRODUCT_KERNEL_CONFIG += rk356x.config
 # used for fstab_generator, sdmmc controller address
 PRODUCT_BOOT_DEVICE := ff870000.mmc

-SF_PRIMARY_DISPLAY_ORIENTATION := 0
+SF_PRIMARY_DISPLAY_ORIENTATION := 270

 # Disable emulator for "make dist" until there is a 64-bit qemu kernel
 BUILD_EMULATOR := false

```

结果如下图所示：



```
# BOARD_AVB_ENABLE := true
# used for fstab_generator, sdmmc controller address
PRODUCT_BOOT_DEVICE := ff870000.mmc

# SF_PRIMARY_DISPLAY_ORIENTATION := 0
SF_PRIMARY_DISPLAY_ORIENTATION := 270

# Disable emulator for "make dist" until there is a 64-bit qemu kernel
BUILD_EMULATOR := false
TARGET_BOARD_PLATFORM := rk3562
TARGET_BOARD_PLATFORM_GPU := mali-G52
TARGET_RK_GRALLOC_AIDL := true
TARGET_RK_GRALLOC_VERSION := 4
BOARD_USE_DRM := true
```

以上三个位置修改后，重新编译，编译命令如下：

```shell
$ source build/envsetup.sh
$ lunch rk3562_t-userdebug
$ ./build.sh -CKAu -d ido-evb3562-v1b-dsi-mipi -J6
```