

# IDO-Purple Pi OH RK3566-V1 Android开发手册

---

## 注意事项

### 1 源码获取

#### 1.1 整包源码获取

#### 1.2 补丁获取及使用方法

### 2 开发环境配置

### 3 SDK编译

#### 3.1 uboot编译步骤

#### 3.2 kernel编译步骤

#### 3.3 Android编译及固件生成步骤

#### 3.4 固件打包



# IDO-Purple Pi OH RK3566-V1 Android开发手册

## 文档修订历史

版本	PCBA版本	修订内容	修订	审核	日期
V1.0	V1B	创建文档	FYZ	IDO	2023/04/06
V1.1	V1B	添加Kernel单独编译说明	FYZ	IDO	2023/05/15
V1.2	V1B	添加解压和编译权限说明	FYZ	IDO	2023/07/22
V1.3	V1B	添加补丁使用说明	FYZ	IDO	2023/08/01
V1.4	V1B	优化文档	LZR	IDO	2024/05/10
V1.5	V1B	添加PCBA版本说明	FYZ	JJM、 ZYY	2024/08/07

## 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

### 1.1 整包源码获取

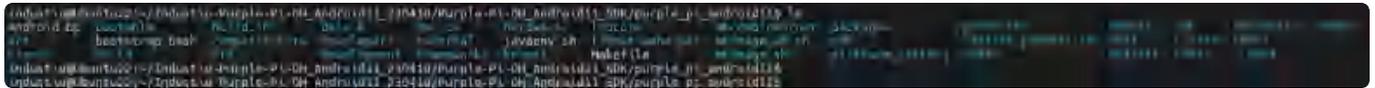
链接: <https://pan.baidu.com/s/1fjz9hWrnqzjs70zcSt5A9A?pwd=1234>

提取码: 1234

获取源码命令如下:

```
Bash |
1 cat purple_pi_android11.tar.gza* | tar -xz -C target_path
```

结果如下图所示:



## 1.2 补丁获取及使用方法

所有的补丁都是基于1.1中的sdk整包生成, 补丁位于源码下载连接当中。

补丁的目录结构如下表所示:

文件	说明
xxx.patch	补丁
files	按sdk目录结构放置xxx.patch补丁修改后的文件
readme.txt	打补丁的方法和编译说明

打补丁的方法:

- 方法一: 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, 请自行使用files中的文件对比修改解决冲突后提交, 命令如下:

```
Shell |
1 cd your/sdk/root/path/
2 git am --whitespace=fix xxx.patch
```

- 方法二: 将补丁中的files文件与sdk根目录对比替换原sdk中的内容。

## 2 开发环境配置

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

RKDocs\android\Rockchip\_Developer\_Guide\_Android11\_SDK\_V1.1.6\_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
1  $ sudo apt-get update
2  $ sudo apt-get install git gnupg flex bison gperf libssl1.2-dev libbsd-jav
   a \
3  squashfs-tools build-essential zip curl libncurses-dev zlib1g-dev \
4  pngcrush schedtool libxml2 libxml2-utils xsltproc lzip libc6-dev schedtool
   g++-multilib \
5  lib32ncurses5-dev lib32readline-dev gcc-multilib libswitch-perl libssl-dev

```

结果如下图所示：

```

industio@ubuntu22:~/Purple-Pi-0H/Industio-Purple-Pi-0H_Android11_230410$ sudo apt-get update
获取:1 https://packages.mirrors.ustc.edu.cn/repos/codes stable InRelease [3,990 B]
命中:2 https://security.ubuntu.com/ubuntu jammy-security InRelease
命中:3 http://mirrors.tuna.tsinghua.edu.cn/ubuntu jammy InRelease
命中:4 http://mirrors.tuna.tsinghua.edu.cn/ubuntu jammy-updates InRelease
命中:5 http://mirrors.tuna.tsinghua.edu.cn/ubuntu jammy-backports InRelease
已下载 3,530 B，耗时 1秒 (4,009 B/s)
正在读取软件包列表... 完成
industio@ubuntu22:~/Purple-Pi-0H/Industio-Purple-Pi-0H_Android11_230410$ sudo apt-get install git gnupg flex bison gperf libssl1.2-dev libbsd java \
> squashfs-tools build-essential zip curl libncurses-dev zlib1g-dev \
> pngcrush schedtool libxml2 libxml2-utils xsltproc lzip libc6-dev schedtool g++-multilib \
> lib32ncurses5-dev lib32readline-dev gcc-multilib libswitch-perl libssl-dev \
> unzip zip device-tree-compiler libl4-tool 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
正在读取变更信息... 完成
注意：选中 'lib32ncurses-dev' 而非 'lib32ncurses5-dev'
bison 已经是最新版 (2:3.8.2+dfsg-1build1)。
build-essential 已经是最新版 (12.0ubuntu3)。
device-tree-compiler 已经是最新版 (1.0.1-1)。
flex 已经是最新版 (2.6.4-8build2)。
lib32readline-dev 已经是最新版 (8.1.2-1)。
lzip 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)。
libbsd-java 已经是最新版 (0.0.7-5)。
libl4-tool 已经是最新版 (1:9.1-2build2)。
libssl1.2-dev 已经是最新版 (1.2.15+dfsg-2)。
pngcrush 已经是最新版 (1.8.13-0.1)。
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)。
libc6-dev 已经是最新版 (2.35-0ubuntu3.7)。
libncurses-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)。
下列软件包是自动安装的并且现在不需要了：
binutils-arm-linux-gnueabi cpp-arm-linux-gnueabi gcc-11-arm-linux-gnueabi-base gcc-12-cross-base gcc-12-cross-base libasan6-armel-cross
libatomic1-armel-cross libc6-armel-cross libc6-dev-armel-cross libgcc-11-dev-armel-cross libgcc-11-dev-armel-cross libgomp1-armel-cross libstdc++6-armel-cross
libubsan1-armel-cross linux-libc-dev-armel-cross
使用 'sudo apt autoremove' 来卸载它(它们)。
下列软件包将被【卸载】：
gcc-11-arm-linux-gnueabi gcc-arm-linux-gnueabi
下列【新】软件包将被安装：
g++-multilib gcc-multilib libswitch-perl
升级了 8 个软件包，新安装了 3 个软件包，要卸载 2 个软件包，有 8 个软件包未被升级。
需要下载 18.3 kB/20.6 kB 的归档。
解压后将会空出 53.6 MB 的空间。
获取:1 http://mirrors.tuna.tsinghua.edu.cn/ubuntu jammy/universe amd64 libswitch-perl all 2.17-2.1 [10.3 kB]

```

## 3 SDK编译

## 3.1 uboot编译步骤

进入 sdk 根目录命令如下：

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

结果如下图所示：

```
industio@ubuntu22:~/Industio-Purple-Pi-0H_Android11_230410/Purple-Pi-0H_Android11_SDK/purple_pi_android11$ cd u-boot/
industio@ubuntu22:~/Industio-Purple-Pi-0H_Android11_230410/Purple-Pi-0H_Android11_SDK/purple_pi_android11/u-boot$ ./make.sh rk3566
grep: .config: No such file or directory
## make rk3566_defconfig rk3566.config -j14
HOSTCC scripts/basic/fixdep
HOSTCC scripts/kconfig/conf.o
SHIPPED scripts/kconfig/zconf.tab.c
SHIPPED scripts/kconfig/zconf.lex.c
SHIPPED scripts/kconfig/zconf.hash.c
HOSTCC scripts/kconfig/zconf.tab.o
HOSTLD scripts/kconfig/conf
#
# configuration written to .config
#
Using .config as base
Merging ./arch/./configs/rk3566.config
Value of CONFIG_BASE_DEFCONFIG is redefined by fragment ./arch/./configs/rk3566.config:
Previous value: CONFIG_BASE_DEFCONFIG=""
New value: CONFIG_BASE_DEFCONFIG="rk3566_defconfig"
Value of CONFIG_LOADER_INI is redefined by fragment ./arch/./configs/rk3566.config:
Previous value: CONFIG_LOADER_INI=""
New value: CONFIG_LOADER_INI="RK3566MINIMAL.ini"
```

uboot编译成功结果如下图所示：

```
oack loader ok (rk3566 spl loader v1.16.112.bin)(0.04)
oack loader okay! Input: /home/industio/Industio-Purple-Pi-0H_Android11_230410/Purple-Pi-0H_Android11_SDK/purple_pi_android11/rkbin/RKBOOT/RK3566MINIMAL.ini
/home/industio/Industio-Purple-Pi-0H_Android11_230410/Purple-Pi-0H_Android11_SDK/purple_pi_android11/u-boot
Image is signed, verify ok! uboot img (FIT with uboot, fdt, ...) is ready
Image is signed! rk3566 spl loader v1.16.112.bin (with spl, ddr, ubiplug) is ready
oack abort img okay! Input: /home/industio/Industio-Purple-Pi-0H_Android11_230410/purple_pi_0H_android11_sdk/purple_pi_android11/rkbin/RKTRUST/RK3566TRUST.VBI
Platform rk3566 is build OK with new config make rk3566_defconfig rk3566.config -j14
/home/industio/Industio-Purple-Pi-0H_Android11_230410/Purple-Pi-0H_Android11_SDK/purple_pi_android11/prebuilt/gcc/1/usr-x86_64-linux4/gcc-11/usr-bin.3.1-2017.05-x86_64-linux4-11/usr-gnulib
./arch/arm64/Makefile:54: *** 目标没有生成。
industio@ubuntu22:~/Industio-Purple-Pi-0H_Android11_230410/Purple-Pi-0H_Android11_SDK/purple_pi_android11/u-boot$
```

## 3.2 kernel编译步骤

内核配置文件路径：kernel/arch/arm64/configs/rockchip\_purple\_pi\_3566\_defconfig

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

内核配置及设备树文件名，如下表所示：

序号	文件名	功能说明
1	ido-pi-oh3566-v1.dts	显示为hdmi
2	ido-pi-oh3566-v1-dsi0-mipi.dts	显示为MIPI

以编译HDMI屏内核为例，编译命令如下：

```

1  $ cd kernel
2  $ make ARCH=arm64 rockchip_purple_pi_3566_defconfig rk356x_evb.config android-11.config
3  $ make ARCH=arm64 ido-pi-oh3566-v1.img -j10

```

编译结果如下图所示：

```

Indust1a@ubuntu22:~/Indust1a-Purple-Pi-OH_Android11_230410/Purple-Pi-OH_Android11_SDK/purple_pi_android11/kernel$
Indust1a@ubuntu22:~/Indust1a-Purple-Pi-OH_Android11_230410/Purple-Pi-OH_Android11_SDK/purple_pi_android11/kernel$ make ARCH=arm64 rockchip_purple_pi_3566_defconfig rk356x_evb.config android-11.config
HOSTCC scripts/basic/fixdep
HOSTCC scripts/kconfig/conf.o
YACC  scripts/kconfig/zconf.tab.c
LEX   scripts/kconfig/zconf.lex.c
HOSTCC scripts/kconfig/zconf.tab.o
HOSTLD scripts/kconfig/conf
# configuration written to .config
#
Using .config as base
Merging ./kernel/configs/rk356x_evb.config
# merged configuration written to .config (needs make)
#
scripts/kconfig/conf --oldconfig Kconfig
# configuration written to .config
#
Using .config as base
Merging ./kernel/configs/android-11.config
# merged configuration written to .config (needs make)
#
scripts/kconfig/conf --oldconfig Kconfig
# configuration written to .config
#
Indust1a@ubuntu22:~/Indust1a-Purple-Pi-OH_Android11_230410/Purple-Pi-OH_Android11_SDK/purple_pi_android11/kernel$ make ARCH=arm64 ido-pi-oh3566-v1.img -j10
scripts/kconfig/conf --syncconfig Kconfig
WRAP arch/arm64/include/generated/uapi/asm/arm64.h
WRAP arch/arm64/include/generated/uapi/asm/ioctls.h
WRAP arch/arm64/include/generated/uapi/asm/ipcbuf.h
WRAP arch/arm64/include/generated/uapi/asm/kvm_para.h
WRAP arch/arm64/include/generated/uapi/asm/mman.h
WRAP arch/arm64/include/generated/uapi/asm/msgbuf.h
WRAP arch/arm64/include/generated/uapi/asm/poll.h
WRAP arch/arm64/include/generated/uapi/asm/resource.h
WRAP arch/arm64/include/generated/uapi/asm/smbuf.h
WRAP arch/arm64/include/generated/uapi/asm/shmbuf.h
WRAP arch/arm64/include/generated/uapi/asm/socket.h
WRAP arch/arm64/include/generated/uapi/asm/socket.h
WRAP arch/arm64/include/generated/uapi/asm/socket.h
WRAP arch/arm64/include/generated/uapi/asm/swab.h
WRAP arch/arm64/include/generated/uapi/asm/termios.h
WRAP arch/arm64/include/generated/uapi/asm/types.h
WRAP arch/arm64/include/generated/uapi/asm/types.h
UPD  include/config/kernel.release
UPD  include/generated/uapi/linux/version.h

```

**注意：**如果编译提示选择io\_domain电压，vccio4和vccio5选择1.8v，其它选择3.3v。以上方式编译完成后，kernel目录生成的boot.img文件不能直接烧录。需要使用build.sh -K 命令来编译kernel，编译后烧写rockdev/Image-rk3566\_r/boot.img。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-rk3566\_r/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：

```

1  $ cd kernel
2  $ make ARCH=arm64 rockchip_purple_pi_3566_defconfig rk356x_evb.config android-11.config
3  $ make ARCH=arm64 BOOT_IMG=./rockdev/Image-rk3566_r/boot.img ido-pi-oh3566-v1.img -j10

```

kernel编译结果，如下图所示：

```
Building modules, stage 2.
MODPOST 4 modules
CC      drivers/media/usb/gspca/gspca_main.mod.o
CC      drivers/net/wireless/rockchip_wlan/rkwifi/bcmdhd/bcmdhd.mod.o
CC      drivers/net/wireless/rockchip_wlan/rtl8723ds/8723ds.mod.o
CC      drivers/net/wireless/rockchip_wlan/rtl8821cs/8821cs.mod.o
LD [M]  drivers/net/wireless/rockchip_wlan/rtl8821cs/8821cs.ko
LD [M]  drivers/net/wireless/rockchip_wlan/rkwifi/bcmdhd/bcmdhd.ko
LD [M]  drivers/media/usb/gspca/gspca_main.ko
LD [M]  drivers/net/wireless/rockchip_wlan/rtl8723ds/8723ds.ko
found ./arch/arm64/boot/dts/rockchip/.ido-pi-oh3566-v1.dtb.dts.tmp
Image: resource.img (with ido-pi-oh3566-v1.dtb logo.bmp logo_kernel.bmp) is ready
Image: boot.img (with Image resource.img) is ready
Image: zboot.img (with Image.lz4 resource.img) is ready
industio@ubuntu22:~/Industio-Purple-Pi-0H_Android11_230410/Purple-Pi-0H_Android11_SDK/purple_pi_android11/kernel$
```

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。

### 3.3 Android编译及固件生成步骤

```
Shell |
1  $ source build/envsetup.sh
2  $ lunch rk3566_r-userdebug
3  $ make -j10
```

结果如下图所示：

```
industio@ubuntu22:~/Industio-Purple-Pi-0H_Android11_230410/Purple-Pi-0H_Android11_SDK/purple_pi_android11$ source build/envsetup.sh
industio@ubuntu22:~/Industio-Purple-Pi-0H_Android11_230410/Purple-Pi-0H_Android11_SDK/purple_pi_android11$
industio@ubuntu22:~/Industio-Purple-Pi-0H_Android11_230410/Purple-Pi-0H_Android11_SDK/purple_pi_android11$ lunch rk3566_r-userdebug

=====
PLATFORM_VERSION_CODENAME=REL
PLATFORM_VERSION=11
TARGET_PRODUCT=rk3566_r
TARGET_BUILD_VARIANT=userdebug
TARGET_BUILD_TYPE=release
TARGET_ARCH=arm64
TARGET_ARCH_VARIANT=armv8-a
TARGET_CPU_VARIANT=cortex-a55
TARGET_2ND_ARCH=arm
TARGET_2ND_ARCH_VARIANT=armv8-2a
TARGET_2ND_CPU_VARIANT=cortex-a55
HOST_ARCH=x86_64
HOST_2ND_ARCH=x86
HOST_OS=linux
HOST_OS_EXTRA=Linux-6.5.0-35-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=RQ3A.210705.001
OUT_DIR=out
=====
industio@ubuntu22:~/Industio-Purple-Pi-0H_Android11_230410/Purple-Pi-0H_Android11_SDK/purple_pi_android11$ make -j10
```

Android编译成功结果如下图所示：

```

[100% 52940/52940] Target Super Ye image for debug: out/target/product/rk3566_r/super.img
2024-05-17 10:46:25 build_super_image.py - INFO : Building super image from 'lunch' dict...
2024-05-17 10:46:25 sparse_img.py - INFO : Total of 239360 4096-byte output blocks in 18 input chunks.
2024-05-17 10:46:25 sparse_img.py - INFO : Total of 30160 4096-byte output blocks in 3 input chunks.
2024-05-17 10:46:25 sparse_img.py - INFO : Total of 74647 4096-byte output blocks in 8 input chunks.
2024-05-17 10:46:25 sparse_img.py - INFO : Total of 52097 4096-byte output blocks in 5 input chunks.
2024-05-17 10:46:25 sparse_img.py - INFO : Total of 453 4096-byte output blocks in 3 input chunks.
2024-05-17 10:46:25 common.py - INFO : Running: 'lsmake -w -fdata=1-rs 65235 --split-name super --m4data=5161 2 --device super rk3566r3566 --lunch flash011
258574288 --partition system:readonly:99F418560:rockchip:dynamic:partitions --image system=out/target/product/rk3566_r/system.img --partition system:readonly:124
6_187116016 --image system_ext=out/target/product/rk3566_r/system_ext.img --partition vendor:readonly:395754152:rockchip:dynamic:partitions --lunch vendor-out:target
vendor_img --partition product:readonly:211348352:rockchip:dynamic:partitions --image product=out/target/product/rk3566_r/product.img --lunch l00_ada:readonly:626608:
t10ns --image hda=out/target/product/rk3566_r/hda.img --sparse --lunch hnt/target/product/rk3566_r/super_img'
2024-05-17 10:46:25 common.py - INFO : lsmake I 05-17 10:46:25 156728 156220 builder.cpp:10113 [libio]Partition system will resize from 0 bytes to 99F418560 by
lsmake I 05-17 10:46:25 156728 156326 builder.cpp:10113 [libio]Partition system_ext will resize from 0 bytes to 124672208 bytes
lsmake I 05-17 10:46:25 156728 156326 builder.cpp:10113 [libio]Partition vendor will resize from 0 bytes to 395754152 bytes
lsmake I 05-17 10:46:25 156728 156228 builder.cpp:10113 [libio]Partition product will resize from 0 bytes to 211348352 bytes
lsmake I 05-17 10:46:25 156728 156328 builder.cpp:10113 [libio]Partition hda will resize from 0 bytes to 626608 bytes
2024-05-17 10:46:32 build_super_image.py - INFO : Done writing image out/target/product/rk3566_r/super.img

```

版本选择：lunch rk3566\_r-user

需要编译内核选择的屏幕修改Android主显和系统方向，如下表所示：

屏幕	系统配置参数修改
HDMI	<ul style="list-style-type: none"> <li>device/rockchip/rk356x/rk3566_r/rk3566_r.mk PRODUCT_PROPERTY_OVERRIDES += vendor.hwc.device.primary=HDMI-A-1</li> <li>device/rockchip/rk356x/BoardConfig.mk SF_PRIMARY_DISPLAY_ORIENTATION := 0</li> </ul>
MIPI	<ul style="list-style-type: none"> <li>device/rockchip/rk356x/rk3566_r/rk3566_r.mk PRODUCT_PROPERTY_OVERRIDES += vendor.hwc.device.primary=DSI</li> <li>device/rockchip/rk356x/BoardConfig.mk SF_PRIMARY_DISPLAY_ORIENTATION := 270</li> </ul>

### 3.4 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-rk3566\_r/目录下，命令如下：

```

▼ Shell |
1 $ ./mkimage.sh

```

结果如下图所示：

```

industio@ubuntu22:~/Industio-Purple-Pi-0H_Android11_230410/Purple-Pi-0H_Android11_SDK/purple_pi_android11$ ls
Android.bp  bootstrap.bash  cts  device  javaenv.sh  Makefile  out  prebuilts  rkat  test  vendor
art  build  dalvik  external  kernel  mkncombinedroot  packages  restore_patches.sh  RKTools  toolchain
binutils  build.sh  developers  frameworks  libcore  mkimage_ab.sh  pdk  rkbin  sdk  tools
bootable  compatibility  development  hardware  libnativehelper  mkimage.sh  platform_testing  RKDocs  system  u-boot
industio@ubuntu22:~/Industio-Purple-Pi-0H_Android11_230410/Purple-Pi-0H_Android11_SDK/purple_pi_android11$
industio@ubuntu22:~/Industio-Purple-Pi-0H_Android11_230410/Purple-Pi-0H_Android11_SDK/purple_pi_android11$ ./mkimage.sh
TARGET_PRODUCT=rk3566_r
TARGET_BASE_PARAMETER_IMAGE=device/rockchip/common/baseparameter/v2.0/baseparameter.img
HIGH_RELIABLE_RECOVERY_DTA=
BOARD_AVB_ENABLE=false
system filesystem is ext4
create dtbo.img...
done.
create resource.img...
done.
create boot.img...
done.
create boot-debug.img...
done.
skip copy images: vendor_boot.img
skip copy images: vendor_boot-debug.img
create recovery.img...
done.
create super.img...
done.
create vbmeta.img...
BOARD_AVB_ENABLE is false, use default vbmeta.img
create misc.img... done.
create uboot.img...
u-boot/trust.img not found! Please make it from u-boot first!
create loader...
create config.cfg...
create baseparameter... done.

```

将所有分区镜像合并成单个的镜像，命令如下：

```

▼ | Bash |
1 $ cd RKTools/linux/Linux_Pack_Firmware/rockdev/
2 $ ./mkupdate_rk356x.sh

```

结果如下图所示：

```

industio@ubuntu22:~/Industio-Purple-Pi-0H_Android11_230410/Purple-Pi-0H_Android11_SDK/purple_pi_android11/RKTools/linux/Linux_Pack_Firmware/rockdev$ ls
ifptool  mkupdate_rk3226_ab.sh  mkupdate_rk3226.sh  mkupdate_rk3226p.sh  pack_boot_file_rk356x_box.sh  pack_boot_file_rk356x_box_ab
gen_package_file.sh  mkupdate_rk3226p_ab.sh  mkupdate_rk3226p.sh  mkupdate_rk356x_ab.sh  package-file-ab  paramo.txt
Image  mkupdate_rk3226.sh  mkupdate_rk3226_ab.sh  mkupdate_rk356x_box_32.sh  package-file-box  rkImageMaker
mkupdate_ab_rk356x_box_32.sh  mkupdate_rk3226p_ab.sh  mkupdate_rk3226p.sh  mkupdate_rk356x_box.sh  package-file-rk356x  update.sh
mkupdate_ab_rk356x_box.sh  mkupdate_rk3226p.sh  mkupdate_rk3226p.sh  mkupdate_rk356x_box.sh  package-file-rk356x-ab
mkupdate_pp3a.sh  mkupdate_rk3226p_ab.sh  mkupdate_rk3226p.sh  mkupdate_rk356x_box.sh  package-file-rk356x-box
industio@ubuntu22:~/Industio-Purple-Pi-0H_Android11_230410/Purple-Pi-0H_Android11_SDK/purple_pi_android11/RKTools/linux/Linux_Pack_Firmware/rockdev$ ./mkupdate_rk356
x.sh
start to make update.img...
Android Firmware Package Tool v2.0
----- PACKAGE
Add file: ./package-file
package-file,Add file: ./package-file done,offset=0x000,size=0x2ba,userspace=0x1
Add file: ./Image/MiniloaderAll.bin
bootloader,Add file: ./Image/MiniloaderAll.bin done,offset=0x1000,size=0x6f9c0,userspace=0xe0
Add file: ./Image/parameter.txt
parameter,Add file: ./Image/parameter.txt done,offset=0x71000,size=0x262,userspace=0x1
Add file: ./Image/uboot.img
uboot,Add file: ./Image/uboot.img done,offset=0x71000,size=0x40000,userspace=0x000
Add file: ./Image/misc.img
misc,Add file: ./Image/misc.img done,offset=0x471000,size=0xc900,userspace=0x18
Add file: ./Image/boot.img
boot,Add file: ./Image/boot.img done,offset=0x47d000,size=0x1f0700,userspace=0x30e0
Add file: ./Image/dtbo.img
dtbo,Add file: ./Image/dtbo.img done,offset=0x2384000,size=0x26f,userspace=0x1
Add file: ./Image/vbmeta.img
vbmeta,Add file: ./Image/vbmeta.img done,offset=0x2385000,size=0x100,userspace=0x2
Add file: ./Image/recovery.img
recovery,Add file: ./Image/recovery.img done,offset=0x2380000,size=0x41a000,userspace=0x8354
Add file: ./Image/baseparameter.img
baseparameter,Add file: ./Image/baseparameter.img done,offset=0x6530000,size=0x100000,userspace=0x200
Add file: ./Image/super.img
super,Add file: ./Image/super.img done,offset=0x6530000,size=0x5fe52cf4,userspace=0xbfc6
Add CRC...
Make firmware OK!
----- OK
*****rkImageMaker ver 2.0*****
Generating new image, please wait...
Writing head info...
Writing boot file...
Writing firmware...
Generating MDS data...
MDS data generated successfully!
New image generated successfully!
Making update.img OK

```

执行mkupdate\_rk356x.sh 命令后会将各分区镜像合并成一个update.img 的镜像文件，如下图所示：

