



PET_RK3576_P01 开发板 系统开发手册

一、安卓系统开发

1、Uboot 研发

详见 芯片原厂文档\common\UBOOT 目录下相关文件

Uboot 源码位于 u-boot 目录下

2、内核研发

内核设备树文件位置:

kernel-6.1/arch/arm64/boot/dts/rockchip/rk3576.dtsi

kernel-6.1/arch/arm64/boot/dts/rockchip/rk3576-gzpeite.dtsi

默认内核配置 kernel-6.1/arch/arm64/configs/gzpeite_rk3576_defconfig

```
./gzpeite_build_android.sh -m
```

修改相关配置并保存, 同时要手动将新的配置文件复制到 config 目录

```
cp -rf kernel-6.1/.config kernel-6.1/arch/arm64/configs/gzpeite_rk3576_defconfig
```

其他内核相关研发请参考 芯片原厂文档 目录下的相关文档

3、修改启动 logo

用新的 bmp 文件替换 kernel-6.1 目录下的 logo.bmp 和 logo_kernel.bmp 文件, 图片分辨率不要超过屏幕分辨率。

4、修改开机动画

将动画文件 bootanimation.zip 复制到 device/rockchip/common/bootshutdown 目录下

动画制作需要注意以下几个问题:

- 1、图片分辨率不要超过屏幕分辨率
- 2、压缩 bootanimation.zip 文件是需要选择“存储”方式
- 3、压缩后用 winrar 打开看一下, 不能有 bootanimation 这个目录

5、修改默认桌面背景

用新的桌面背景文件替换 device/rockchip/rk3576/overlay/frameworks/base/core/res/res 所有子目录内的 default_wallpaper.png 文件。

6、内置其他应用 APK

可参考 device\rockchip\rk3576\BaiduSpeechService 目录下的文件

7、开机自启动 Launcher(不显示系统桌面)

首先在开发应用 APK 时, 需要在应用程序 AndroidManifest.xml 的 Intent-filter 里添加下面几行

```
<intent-filter>
    <action android:name="android.intent.action.MAIN" />
    <category android:name="android.intent.category.HOME" />
    <category android:name="android.intent.category.DEFAULT"/>
</intent-filter>
```

可以参考源代码目录下的 OnlyLauncher.7z

8、修改系统默认参数配置

系统参数配置文件位置 `device/rockchip/rk3576/rk3576_u/rk3576_u.mk`

主显示屏默认显示方向

默认旋转 0 度，其他方向注意需要同时修改下面几个参数值

参数名	旋转 0 度	旋转 90 度	旋转 180 度	旋转 270 度
ro.surface_flinger.primary_display_orientation	ORIENTATION_0	ORIENTATION_90	ORIENTATION_180	ORIENTATION_270
ro.minui.default_rotation	ROTATION_NONE	ROTATION_RIGHT	ROTATION_DOWN	ROTATION_LEFT
ro.input_flinger.primary_touch.rotation	0	90	180	270

副显示屏默认显示方向

persist.sys.rotation.einit

默认值 0

可选值：0、1、2、3

0：不旋转 1：旋转 90 度 2：旋转 180 度 3：旋转 270 度

禁止屏幕旋转

persist.sys.forced_orient

当选择禁止屏幕旋转后，如果系统默认是横屏显示，即使启动竖屏应用，屏幕显示方向也不会改变
默认值 0

可选值：0、1

默认是否全屏显示（隐藏状态栏）

persist.sys.def_hidenavigation

persist.sys.def_hidestatusbar

默认值：0

可选值：0、1

主显示屏 DPI 值调整

ro.sf.lcd_density

默认值 160

可选值：120、160、240、320

副显示屏 DPI 值调整

persist.sys.eint_density

默认值 160

可选值：120、160、240、320

以太网默认设置

仅用于烧写固件或恢复出厂设置后的默认配置，不能在系统启动后用命令行或应用 app 修改参数值。
系统启动后需通过系统设置程序菜单修改或在应用 app 内使用安卓标准方式修改以太网参数。

persist.net.eth0.mode

默认值：0

可选值：0（DHCP）、1（静态）

persist.net.eth0.ip

默认值：192.168.1.200/24 格式为<IP>/24

persist.net.eth0.gateway

默认值: 192.168.1.1 格式为<IP>

persist.net.eth0.dns

默认值: 192.168.1.1 格式为<IP>

persist.net.eth1.mode

默认值: 0 可选值: 0 (DHCP)、1 (静态)

persist.net.eth1.ip

默认值: 192.168.2.200/24 格式为<IP>/24

persist.net.eth1.gateway

默认值: 192.168.2.1 格式为<IP>

persist.net.eth1.dns

默认值: 192.168.2.1 格式为<IP>

以太网共享默认设置

是否默认启用以太网共享

persist.net.eth.Tether

默认值: 0

可选值: 0、1

persist.net.eth.TetherIP=192.168.1.1/24

默认值: 192.168.2.200/24 格式为<IP>/24

是否关闭北斗/GPS 功能

config.disable_gps

默认值: false

可选值: false、true

是否关闭蓝牙功能

config.disable_bluetooth

默认值: false

可选值: false、true

是否打开开发者选项

sys.def_develop_enable

默认值: 1

可选值: 0、1

长按电源键功能

sys.def_powerkey_long

默认值: 1

可选值: 0 (无效)、1 (显示关机菜单)、2 (直接关机需确认)、3 (直接关机无需确认)

默认 WIFI 自动连接 SSID 和密码

persist.net.wifi.ssid

默认值: PEITE-WIFI-WORK

persist.net.wifi.pass

默认值: peite-13579

默认是否打开 WIFI**sys.def_wifi_on**

默认值: 1

可选值: 0、1

默认是否打开蓝牙**sys.def_bluetooth_on**

默认值: 0

可选值: 0、1

自动休眠时间**sys.def_screen_off_timeout**

默认值: 0

可选值: 0 永不休眠

1800000 30 分钟

600000 10 分钟

300000 5 分钟

120000 2 分钟

60000 1 分钟

30000 30 秒

15000 15 秒

默认背光亮度**sys.def_screen_brightness**

默认值: 255

可选值: 0 ~ 255

系统默认音量

sys.def_volume_music=15 范围 0 ~ 15

sys.def_volume_ring=7 范围 0 ~ 7

sys.def_volume_system=7 范围 0 ~ 7

sys.def_volume_voicecall=5 范围 0 ~ 5

sys.def_volume_alarm=7 范围 0 ~ 7

sys.def_volume_notification=7 范围 0 ~ 7

sys.def_volume_bluetoothsoc=15 范围 0 ~ 15

是否禁用深度休眠**sys.def_no_deepsleep**

默认值: 1

可选值: 0 (启用深度休眠)、1 (禁用深度休眠)

默认 NTP 服务器地址**sys.def_ntp_server**

默认值: ntp.aliyun.com

默认 NTP 超时时间

sys.def_ntp_timeout

默认值: 10000

默认界面模式

sys.def_nightmode

默认值: 2

可选值: 0 (自动模式)、1 (普通模式)、2 (黑夜暗黑模式)

是否禁用 USB 音频

sys.def_usbaudio_disable

默认值: 1

可选值: 0 (启用)、1 (禁用)

是否显示虚拟音量按键

persist.sys.show_volumebutton

默认值: 1

可选值: 0、1

是否显示虚拟电源按键

persist.sys.show_powerbutton

默认值: 1

可选值: 0、1

定时关机

persist.sys.poweroff_time

默认值: 0

可选值: 0 (不启用)、23:00(格式为 24 小时制 HH:MM)

定时开机

persist.sys.poweron_time

默认值: 0

可选值: 0 (不启用)、07:00(格式为 24 小时制 HH:MM)

USB 摄像头图像旋转

persist.sys.uvc_orientation

默认值: 0

可选值: 0、90、180、270

二、安卓应用开发

1、GPIO 编程参考

通过 sysfs 方式控制 GPIO, GPIO 的操作接口包括 direction 和 value 等, direction 控制 GPIO 输入和输入模式, 而 value 可控制 GPIO 输出或获得 GPIO 输入。

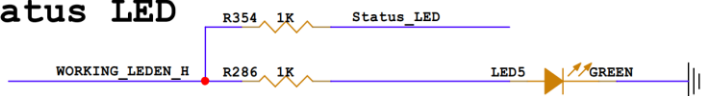
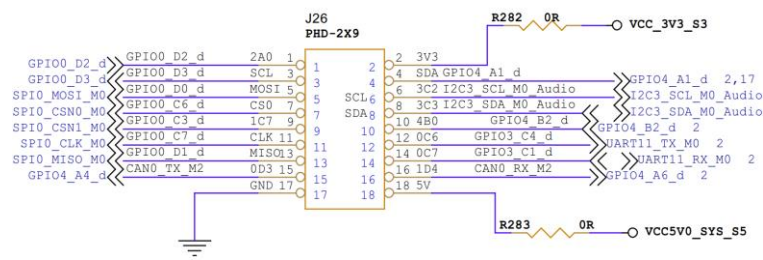
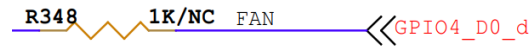
例如控制调试灯 GPIO 操作如下 (串口终端命令行方式):

调试灯 GPIO 设置为输出 echo out > /sys/class/gpio/gpio151/direction

调试灯 GPIO 输出高电平 `echo 1 > /sys/class/gpio/gpio151/value`
 调试灯 GPIO 输出高低平 `echo 0 > /sys/class/gpio/gpio151/value`
 调试灯 GPIO 设置为输入 `echo in > /sys/class/gpio/gpio151/direction`
 读取调试灯 GPIO 输出输入电平 `cat /sys/class/gpio/gpio151/value`

当 GPIO 处于输出和输入模式时都可以读取，当设置为输入模式时读取的是 GPIO 实际电平，当设置为输出模式时读取的是设置的值（如果设置为高电平输出，外部将引脚电平拉低后，读取的值依然是 1）。

应用程序控制请参考源码下的 demo 程序源码

GPIO 对应控制目录列表			
丝印	接口	脚位	目录
LED5	Status LED 		/sys/class/gpio/gpio151
J26		5 脚 7 脚 9 脚 10 脚 11 脚 12 脚 13 脚 14 脚 15 脚 16 脚	/sys/class/gpio/gpio24 /sys/class/gpio/gpio22 /sys/class/gpio/gpio19 /sys/class/gpio/gpio138 /sys/class/gpio/gpio23 /sys/class/gpio/gpio116 /sys/class/gpio/gpio25 /sys/class/gpio/gpio113 /sys/class/gpio/gpio132 /sys/class/gpio/gpio134
4G_RST			/sys/class/gpio/gpio126
4G_PWR			/sys/class/gpio/gpio74
FAN 供电			/sys/class/gpio/gpio152

2、串口 UART 编程参考

J29	串口/dev/ttyS0	XH2.54 3Pin	标配	调试串口，波特率 115200，8 n 1
J32	串口/dev/ttyS1	PH2.0 4Pin	标配	默认为 RS232，可修改电阻配置为 TTL 串口
J35	串口/dev/ttyS2	PH2.0 4Pin	标配	默认为 RS232，可修改电阻配置为 TTL 串口
J30	串口/dev/ttyS3	PH2.0 4Pin	标配	TTL 串口
J31	串口/dev/ttyS5	PH2.0 4Pin	标配	TTL 串口
J33	串口/dev/ttyS6	PH2.0 4Pin	标配	TTL 串口
J34	串口/dev/ttyS7	PH2.0 4Pin	标配	TTL 串口
J36	RS485 /dev/ttyS8	PH2.0 4Pin	标配	RS485
J38	RS485 /dev/ttyS10	PH2.0 4Pin	标配	RS485
	串口/dev/ttyS4			蓝牙功能使用

安卓系统串口编程请参考源码下的 demo 程序源码或以下链接：

<https://github.com/yutils/YSerialPort>

<https://github.com/Accord/AndroidSerialPort>

<https://github.com/Geek8ug/Android-SerialPort>

3、WatchDog 看门狗编程参考

进入内核后默认会启动看门狗，内核崩溃等情况出现，会在 15 秒内自动复位主板。
上层应用程序打开看门狗后，内核将看门狗控制权交由上层应用程序控制，上层应用程序的喂狗间隔建议不大于 3 秒。

看门狗的使用流程为 打开看门狗→循环喂狗→停止喂狗→关闭看门狗

喂狗之前必须先打开看门狗，关闭看门狗之前需停止喂狗操作。

打开看门狗后如果 15 秒内没有喂狗或关闭看门狗，系统会自动复位。

命令行测试：

打开看门狗：echo 1 >/sys/class/gzpeite/user/watch_dog

喂狗：echo 2 >/sys/class/gzpeite/user/watch_dog

关闭看门狗：echo 0 >/sys/class/gzpeite/user/watch_dog

应用程序控制请参考源码下的 demo 程序源码

4、获取 root 权限

系统默认已开启 root 权限，上层应用 app 可直接获取 root 权限并进行相关操作，可以参考源码目录下的 demo 程序

5、系统签名

系统签名文件位于源代码目录下，使用对应的文件对 APK 进行签名即可。

6、动态隐藏/显示系统状态栏和导航栏

隐藏状态栏和导航栏在应用 app 里面向系统发送广播

gzpeite.intent.systemui.hidenavigation 和 gzpeite.intent.systemui.hidestatusbar

显示状态栏和导航栏在应用 app 里面向系统发送广播

gzpeite.intent.systemui.shownavigation 和 gzpeite.intent.systemui.showstatusbar

测试命令如下：

```
am broadcast -a "gzpeite.intent.systemui.hidenavigation"
am broadcast -a "gzpeite.intent.systemui.hidestatusbar"

am broadcast -a "gzpeite.intent.systemui.shownavigation"
am broadcast -a "gzpeite.intent.systemui.showstatusbar"
```

请参考源码下的 demo 程序源码

7、静默安装/卸载应用

静默安装 APK/启动时，向系统发送 gzpeite.intent.action.install_apk 广播

静默卸载 APK 时，向系统发送 gzpeite.intent.action.uninstall_apk 广播

测试命令如下：

```
静默安装：
am broadcast -a "gzpeite.intent.action.install_apk" --es apk_path "/mnt/media_rw/0000-4823/GPSTest.apk"

静默安装并启动：
am broadcast -a "gzpeite.intent.action.install_apk" --es apk_path "/mnt/sdcard/GPSTest-gzpeite.apk" --ez is_start "true"

静默卸载：
am broadcast -a "gzpeite.intent.action.uninstall_apk" --es pkg_name "com.android.gptest"
```

8、重启、关机操作

重启：向系统发送 `gzpeite.intent.action.reboot` 广播

关机向系统发送 `gzpeite.intent.action.shutdown` 广播

测试命令如下：

```
重启(有确认提示): am broadcast -a "gzpeite.intent.action.reboot" --ez confirm true
重启(无确认提示): am broadcast -a "gzpeite.intent.action.reboot" --ez confirm false
关机(有确认提示): am broadcast -a "gzpeite.intent.action.shutdown" --ez confirm true
关机(无确认提示): am broadcast -a "gzpeite.intent.action.shutdown" --ez confirm false
```

应用程序控制请参考源码下的 demo 程序源码

9、获取 MAC 地址

原生 Android12 系统默认禁止应用获取 MAC 地址，为了兼容更早期的应用程序，我司已对系统代码进行优化允许应用 app 获取 WIFI 及以太网的 MAC 地址，详见源代码目录下的 demo 程序源码。

10、 开机自启动 Launcher(不显示系统桌面)

首先在开发应用 APK 时，需要在应用程序 `AndroidManifest.xml` 的 `Intent-filter` 里添加下面几行

```
<intent-filter>
    <action android:name="android.intent.action.MAIN" />
    <category android:name="android.intent.category.HOME" />
    <category android:name="android.intent.category.DEFAULT"/>
</intent-filter>
```

可以参考资料目录下的 `OnlyLauncher.7z`

三、Linux 系统开发

1、Uboot 研发

详见 芯片原厂文档\common\UBOOT 目录下相关文件

Uboot 源码位于 `u-boot` 目录下

2、内核研发

内核设备树文件位置：

`kernel-6.1/arch/arm64/boot/dts/rockchip/rk3576.dtsi`

`kernel-6.1/arch/arm64/boot/dts/rockchip/rk3576-gzpeite.dtsi`

默认内核配置 `kernel/arch/arm64/configs/gzpeite_rk3576_linux_defconfig`

```
./build.sh kernel-config
```

其他内核相关研发请参考 芯片原厂文档 目录下的相关文档

3、修改启动 logo

用新的 bmp 文件替换 `kernel-6.1` 目录下的 `logo.bmp` 和 `logo_kernel.bmp` 文件，图片分辨率不要超过屏幕分辨率。

4、buildroot 配置选项修改

默认配置文件是 `buildroot/configs/rockchip_rk3576_defconfig`

```
cd buildroot
source envsetup.sh rockchip_rk3576
```

```
make menuconfig
make savedefconfig
```

5、buildroot 及 debian12 文件系统修改

buildroot 及 debian 系统由瑞芯微原厂制作，相关文档已经比较完善，可以参考 芯片原厂文档\Linux 目录下的相关文档进行相关修改。

对于 buildroot 跟文件系统修改：

将需要修改的文件放到 buildroot\board\rockchip\rk3576\fs-overlay 目录下的相应目录里面，在编译 buildroot 系统的时候会自动将文件复制到根文件系统内。

对于 debian 根文件系统的修改，可以参考下面的方式：

将需要修改的文件放到 debian\overlay 目录下的相应目录里面，在编译 debian 系统的时候会自动将文件复制到根文件系统内。

6、Ubuntu 24.04 系统修改

修改根文件系统

将需要修改的文件放到 ubuntu\overlay 目录下的相应目录里面，在编译 ubuntu 系统的时候会自动复制到根文件系统内。

修改 WIFI 连接的 SSID 及密码

ubuntu\overlay\usr\local\sbin\boot_run.sh （SSID 是 GZPEITE-WIFI，密码是 1357924680）

```
nmcli connection add type wifi con-name "wlan0" ifname wlan0 ssid "GZPEITE-WIFI"
nmcli connection modify "wlan0" wifi-sec.key-mgmt wpa-psk
nmcli connection modify "wlan0" wifi-sec.psk "1357924680"
```

修改以太网连接参数

ubuntu\overlay\usr\local\sbin\boot_run.sh （默认是 DHCP 方式）

```
nmcli connection delete "Wired connection 1"
nmcli connection add type ethernet con-name "eth0" ifname eth0 （DHCP 方式）
# nmcli connection add type ethernet con-name "eth0" ifname eth0 ipv4.method manual
ipv4.address 192.168.1.68/24 gw4 192.168.1.1 ipv4.dns "223.5.5.5,8.8.8.8" （静态方式）
```

四、Linux 系统 OpenGL 测试

在调试串口终端输入以下命令

```
glmark2-es2-wayland （buildroot 系统）
glmark2-es2 （debian、ubuntu 系统）
```

```

root@rk3576-buildroot:/# glmark2-es2-wayland
arm_release_ver: gl3p0-01eac0, rk_so_ver: 11
=====
glmark2 2023.01
=====
OpenGL Information
GL_VENDOR: ARM
GL_RENDERER: Mali-G52
GL_VERSION: OpenGL ES 3.2 v1.gl3p0-01eac0.98c5dad4e3309b873e3189000b74ea36
Surface Config: buf=32 r=8 g=8 b=8 a=8 depth=24 stencil=0 samples=0
Surface Size: 800x600 windowed
=====
[build] use-vbo=false: FPS: 1758 FrameTime: 0.569 ms
[build] use-vbo=true: FPS: 1638 FrameTime: 0.611 ms
[texture] texture-filter=nearest: FPS: 1574 FrameTime: 0.636 ms
[texture] texture-filter=linear: FPS: 1379 FrameTime: 0.726 ms
[texture] texture-filter=mipmap: FPS: 1394 FrameTime: 0.718 ms
[shading] shading=gouraud: FPS: 1398 FrameTime: 0.716 ms
[shading] shading=blinn-phong-inf: FPS: 1450 FrameTime: 0.690 ms
[shading] shading=phong: FPS: 1350 FrameTime: 0.741 ms
[shading] shading=cel: FPS: 1376 FrameTime: 0.727 ms
[bump] bump-render=high-poly: FPS: 818 FrameTime: 1.223 ms
[bump] bump-render=normals: FPS: 1524 FrameTime: 0.656 ms
[bump] bump-render=height: FPS: 1565 FrameTime: 0.639 ms
[effect2d] kernel=0,1,0,1,-4,1,0,1,0,: FPS: 1125 FrameTime: 0.889 ms
[effect2d] kernel=1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,: FPS: 646 FrameTime: 1.548 ms
[pulsar] light=false:quads=5:texture=false: FPS: 1571 FrameTime: 0.637 ms
[desktop] blur-radius=5:effect=blur:passes=1:separable=true:windows=4: FPS: 819 FrameTime: 1.221 ms
[desktop] effect=shadow:windows=4: FPS: 2065 FrameTime: 0.484 ms
[buffer] columns=200:interleave=false:update-dispersion=0.9:update-fraction=0.5:update-method=map: FPS: 233 FrameTime: 4.294 ms
[buffer] columns=200:interleave=false:update-dispersion=0.9:update-fraction=0.5:update-method=subdata: FPS: 235 FrameTime: 4.259 ms
[buffer] columns=200:interleave=true:update-dispersion=0.9:update-fraction=0.5:update-method=map: FPS: 284 FrameTime: 3.521 ms
[ideas] speed=duration: FPS: 924 FrameTime: 1.082 ms
[jellyfish] <default>: FPS: 1081 FrameTime: 0.926 ms
[terrain] <default>: FPS: 106 FrameTime: 9.457 ms
[shadow] <default>: FPS: 814 FrameTime: 1.230 ms
[refract] <default>: FPS: 217 FrameTime: 4.609 ms
[conditionals] fragment-steps=0:vertex-steps=0: FPS: 1613 FrameTime: 0.620 ms
[conditionals] fragment-steps=5:vertex-steps=0: FPS: 1523 FrameTime: 0.657 ms
[conditionals] fragment-steps=0:vertex-steps=5: FPS: 1389 FrameTime: 0.720 ms
[function] fragment-complexity=low:fragment-steps=5: FPS: 1381 FrameTime: 0.724 ms
[function] fragment-complexity=medium:fragment-steps=5: FPS: 1305 FrameTime: 0.767 ms
[loop] fragment-loop=false:fragment-steps=5:vertex-steps=5: FPS: 1360 FrameTime: 0.735 ms
[loop] fragment-steps=5:fragment-uniform=false:vertex-steps=5: FPS: 1355 FrameTime: 0.738 ms
[loop] fragment-steps=5:fragment-uniform=true:vertex-steps=5: FPS: 1357 FrameTime: 0.737 ms
=====
glmark2 Score: 1169
=====
root@rk3576-buildroot:/#

```

五、Linux 系统 OpenCL 测试

在调试串口终端输入以下命令

```

clinfo -a (debian、ubuntu 系统)

root@gzpeite:~# clinfo -a
arm_release_ver: gl3p0-01eac0, rk_so_ver: 11
Number of platforms                                1
Platform Name                                       ARM Platform
Platform Vendor                                    ARM
Platform Version                                   OpenCL 3.0 v1.gl3p0-01eac0.0fd2effaec483a5f4c440d2ffa
25eb7a
Platform Profile                                    FULL_PROFILE
Platform Extensions                                cl_khr_global_int32_base_atomics cl_khr_global_int32_
extended_atomics cl_khr_local_int32_base_atomics cl_khr_local_int32_extended_atomics cl_khr_byte_adres
sable_store cl_khr_3d_image_writes cl_khr_int64_base_atomics cl_khr_int64_extended_atomics cl_khr_fp16
cl_khr_icd cl_khr_egl_image cl_khr_image2d_from_buffer cl_khr_depth_images cl_khr_subgroups cl_khr_subg
roup_extended_types cl_khr_subgroup_non_uniform_vote cl_khr_subgroup_ballot cl_khr_subgroup_non_uniform
_arithmetic cl_khr_subgroup_shuffle cl_khr_subgroup_shuffle_relative cl_khr_subgroup_clustered_reduce c
l_khr_il_program cl_khr_priority_hints cl_khr_create_command_queue cl_khr_spirv_no_integer_wrap_decorat
ion cl_khr_extended_versioning cl_khr_device_uuid cl_khr_suggested_local_work_size cl_khr_extended_bit
ops cl_khr_integer_dot_product cl_khr_semaphore cl_khr_external_semaphore cl_khr_external_semaphore_syn
c_fd cl_khr_command_buffer cl_arm_core_id cl_arm_printf cl_arm_non_uniform_work_group_size cl_arm_impor
t_memory cl_arm_import_memory_dma_buf cl_arm_import_memory_host cl_arm_integer_dot_product_int8 cl_arm_
job_slot_selection cl_arm_scheduling_controls cl_arm_controlled_kernel_termination cl_ext_cxx_for_openc
l cl_ext_image_tiling_control cl_ext_image_requirements_info cl_ext_image_from_buffer
Platform Extensions with Version                   cl_khr_global_int32_base_atomics
0x4000000 (1.0.0)
                                                    cl_khr_global_int32_extended_atomics
0x4000000 (1.0.0)
                                                    cl_khr_local_int32_base_atomics
0x4000000 (1.0.0)
                                                    cl_khr_local_int32_extended_atomics

```

六、Linux 系统 OpenCV 测试

在调试串口终端输入以下命令

```
python3 -c "import cv2; print(cv2.__version__)" (debian、ubuntu 系统)
```

```
root@gzpeite:~#
root@gzpeite:~# python3 -c "import cv2; print(cv2.__version__)"
4.6.0
root@gzpeite:~#
```

七、Linux 系统 NPU 测试

```
cat /sys/kernel/debug/rknpu/version (查询内核驱动版本)
strings /usr/bin/rknn_server | grep -i "rknn_server version" (查询 rknn server 版本)
strings /usr/lib/librknnrt.so | grep -i "librknnrt version" (查询 rknn 库文件版本)
restart_rknn.sh
rknn_common_test /usr/share/model/RK3576/mobilenet_v1.rknn /usr/share/model/dog_224x224.jpg
```

```
root@rk3576-buildroot:/#
root@rk3576-buildroot:/# cat /sys/kernel/debug/rknpu/version
RKMPU driver: v0.9.8
root@rk3576-buildroot:/#
root@rk3576-buildroot:/# strings /usr/bin/rknn_server | grep -i "rknn_server version"
rknn_server version: 2.3.2 (1842325 build@2025-03-30T09:54:34)
root@rk3576-buildroot:/#
root@rk3576-buildroot:/# strings /usr/lib/librknnrt.so | grep -i "librknnrt version"
librknnrt version: 2.3.2 (429f97ae6b@2025-04-09T09:09:27)
root@rk3576-buildroot:/#
root@rk3576-buildroot:/# restart_rknn.sh
root@rk3576-buildroot:/# start rknn server, version:2.3.2 (1842325 build@2025-03-30T09:54:34)
I NPUTransfer(3537): Starting NPU Transfer Server, Transfer version 2.2.2 (2024-06-18T03:50:51)
root@rk3576-buildroot:/#
root@rk3576-buildroot:/# rknn_common_test /usr/share/model/RK3576/mobilenet_v1.rknn /usr/share/model/dog_224x224.jpg
rknn_api/rknnrt version: 2.3.2 (429f97ae6b@2025-04-09T09:09:27), driver version: 0.9.8
model input num: 1, output num: 1
input tensors:
  index=0, name=input, n_dims=4, dims=[1, 224, 224, 3], n_elems=150528, size=150528, fnt=NHWC, type=INT8, qnt_type=AFINE, zp=0, scale=0.007812
output tensors:
  index=0, name=MobilenetV1/Predictions/Reshape_1, n_dims=2, dims=[1, 1001, 0, 0], n_elems=1001, size=2002, fnt=UNDEFINED, type=FP16, qnt_type=AFINE,
custom string:
Begin perf ...
  0: Elapse Time = 5.86ms, FPS = 170.62
----- Top5 -----
0.935059 - 156
0.057037 - 155
0.003881 - 205
0.003119 - 284
0.000172 - 285
root@rk3576-buildroot:/#
```

八、显示屏配置

在源代码目录下有 U 盘或 TF 卡更新显示参数.7z 压缩包，里面有常见显示屏的参考配置，在调试显示屏时根据显示屏的规格书修改内核设备树文件，重新编译源码、烧写测试。

为了提高调试效率，可以先用 U 盘或 TF 更新显示屏参数的方式把显示屏调试好，然后再修改源码编译。

SDK 内的系统源代码默认配置为 HDMI 输出，购买 7 寸或 10.1 寸贴合屏的用户如果重新烧写系统固件后显示屏无显示，用压缩包内下面文件更新显示屏参数后就正常了。

rk3576-gzpeite_p01_lvs_1024x600_7 寸.dts

rk3576-gzpeite_p01_mipi_10.1 寸(40P).dts

九、动态修改开机 logo 和动画

- 1、将 TF 卡或 U 盘格式化为 FAT32(可以在安卓系统的设置->存储菜单里面格式化 U 盘或 TF 卡)
- 2、将制作好的 logo.bmp、logo_kernel.bmp、bootanimation.zip 文件复制到 U 盘或 TF 卡根目录下，注意图片分辨率不能超过显示屏分辨率。可同时更新以上三个或单独某一个文件。
- 3、开机进入安卓或 Linux 系统后，插入 U 盘或 TF 卡

- 4、等待系统自动重启，正确加载新的文件
- 5、拔出 U 盘或 TF 卡

两个 logo 文件必须为 bmp 文件格式，一个是在 uboot 阶段加载显示，一个是在内核阶段加载显示，文件名不可修改，文件内容可以完全一样。

开机动画 bootanimation.zip 仅支持安卓系统，制作方式可以通过搜索引擎查询相关教程。

十、联系方式

总公司：广州佩特电子科技有限公司

总公司地址：广州市天河区大观中路新塘大街鑫盛工业园 A1 栋 201

总公司网站：<http://www.gzpeite.net>

SMT 子公司：广州佩特精密电子科技有限公司（全资子公司）

子公司地址：广州市白云区人和镇大巷村顺景路 11 号

SMT 网站：<http://www.gzptjm.com>

官方淘宝店：<https://shop149045251.taobao.com>

微信扫描维码联系支持人员：



广州佩特电子科技有限公司

2025 年 10 月